

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□□

NX1P2-11□□□□

NX1P2-10□□□□

NX1P2-90□□□□

NJ501-□5□□

NJ501-□4□□

NJ501-□3□□

NJ301-12□□

NJ301-11□□

NJ101-10□□

CPU单元



## — 预告 —

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

## — 商标 —

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

本手册中记载的其它系统名称、产品名称为各公司的商标或注册商标。

## — 著作权 —

屏幕截图的使用已获得微软的许可。

# 前言

非常感谢您购买 NJ/NX 系列 CPU 单元。

本手册记载了使用 NJ/NX 系列 CPU 单元的运动控制功能模块所必需的信息。使用前请仔细阅读本手册，充分理解其功能和性能，并用于系统的构建。

此外，阅读后请将本手册妥善保管于易取处。

## 阅读对象

本手册提供给下列阅读对象：

具有电工专业知识的人员（合格的电气工程师或具有同等知识的人员）；

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

此外，编程语言的阅读对象为理解国际标准规格 IEC61131-3 或国内标准规格 JIS B3503 的规定内容的人员。

## 对象商品

本手册以下列产品为对象。

- NX 系列 CPU 单元
  - NX701-17□□
  - NX701-16□□
  - NX1P2-11□□□□
  - NX1P2-11□□□□1
  - NX1P2-10□□□□
  - NX1P2-10□□□□1
  - NX1P2-90□□□□
  - NX1P2-90□□□□1
- NJ 系列 CPU 单元
  - NJ501-□5□□
  - NJ501-□4□□
  - NJ501-□3□□
  - NJ301-12□□
  - NJ301-11□□
  - NJ101-10□□

各产品的部分规格或限制事项可能记载在其他手册中。请确认 □ “分册构成 (P.2)” 及 □ “相关手册 (P.23)”。

# 分册构成

本产品的手册分为下表所示各分册。请根据目的阅读，充分应用本产品。

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

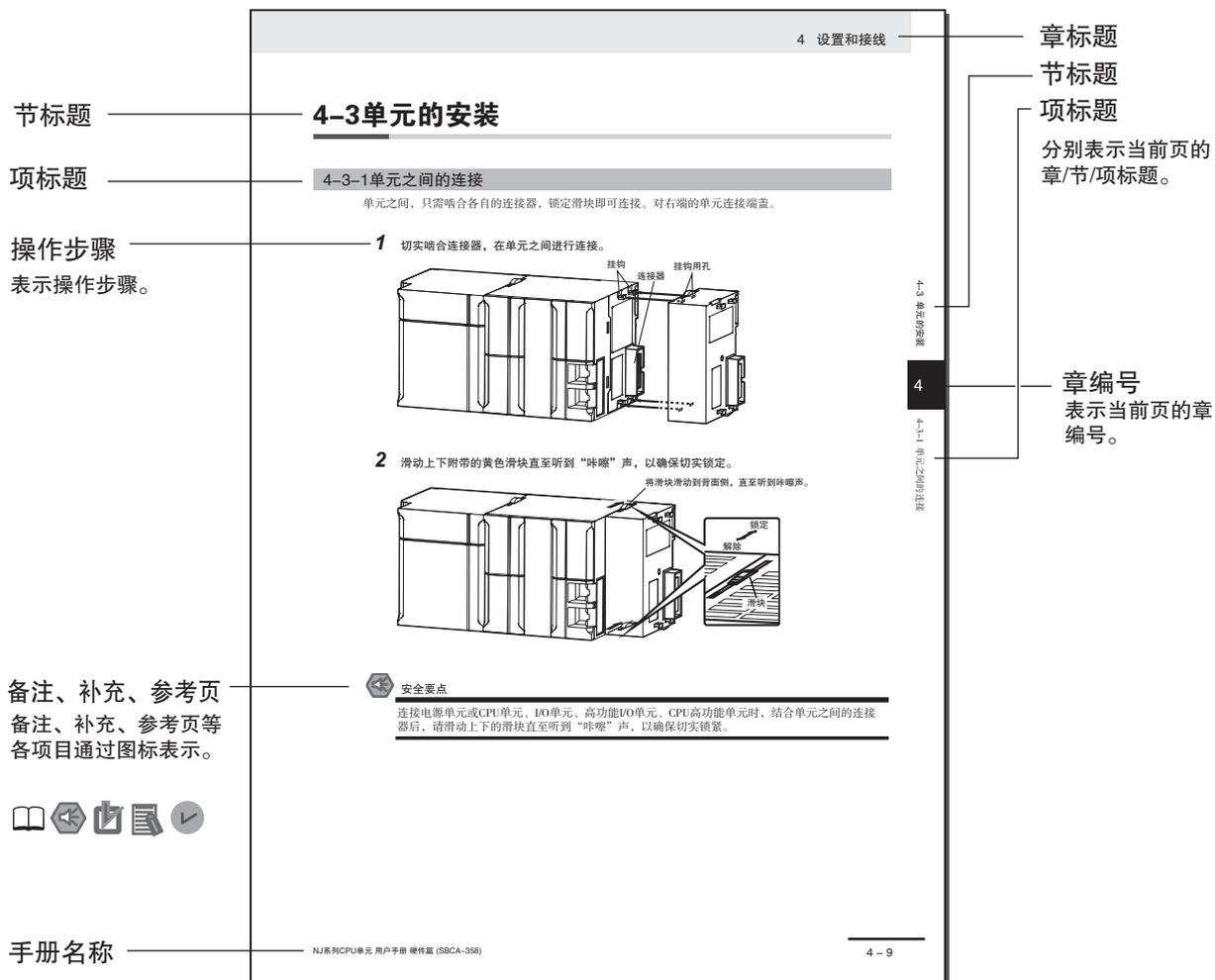
使用目的	基本信息											
	手册											NJ/NX系列 故障诊断手册
	NX系列 CPU 单元 用户手册 硬件篇	NX系列 NX1P2 CPU 单元 用户手册 硬件篇	NJ系列 CPU 单元 用户手册 硬件篇	NJ/NX系列 CPU 单元 用户手册 软件篇	NX系列 NX1P2 CPU 单元 用户手册 内置 I/O、扩展板功能篇	NJ/NX系列 指令基准手册 基本篇	NJ/NX系列 CPU 单元 用户手册 运动控制篇	NJ/NX系列 指令基准手册 运动篇	NJ/NX系列 内置 EtherCAT 端口用户手册	NJ/NX系列 CPU 单元 内置 EtherNet/IP 端口用户手册	NJ系列 数据库连接 CPU 单元 用户手册	
了解NX701的概要	●											
了解NX1P2的概要		●										
了解NJ系列的概要			●									
进行安装、设置、硬件设定												
进行运动控制时							●					
使用EtherCAT时	●	●	●						●			
使用EtherNet/IP时										●		
进行软件设定												
进行运动控制时							●					
使用EtherCAT时									●			
使用EtherNet/IP时										●		
使用数据库连接服务时				●							●	
使用GEM 服务时												●
进行机器人控制时												●
使用NX1P2的功能时					●							
编写用户程序												
进行运动控制时							●	●				
使用EtherCAT时									●			
使用EtherNet/IP时										●		
使用数据库连接服务时				●		●					●	
使用GEM 服务时												●
进行机器人控制时												●
进行异常处理时												●
使用NX1P2的功能时					●							
进行动作确认和调试												
进行运动控制时							●					
使用EtherCAT时									●			
使用EtherNet/IP时										●		
使用数据库连接服务时				●							●	
使用GEM 服务时												●
进行机器人控制时												●
使用NX1P2的功能时					●							
了解异常管理功能和故障发生时的处理方法*	△	△	△	△	△		△		△	△	△	△
了解维护作业												
进行运动控制时	●	●	●				●					
使用EtherCAT时									●			
使用EtherNet/IP时										●		

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

# 手册的阅读方法

## 页面构成

本手册的各页面构成如下所示。



本页为用于说明的范例页。与实际内容有所差异。

## 图标

本资料中使用的图标，含义如下。



### 安全要点

表示为了产品的安全使用而应当实施或避免的事项。



### 使用注意事项

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



### 参考

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



### 版本相关信息

对 CPU 单元、Sysmac Studio 不同版本的不同性能和功能进行说明。



表示详细信息、相关信息的所在页。

## 关于标记的注意事项

在本手册中，将数据从 Sysmac Studio 传送到实机控制器被称为“下载”，从实机控制器传送到 Sysmac Studio 被称为“上传”。

在 Sysmac Studio 中，“下载”及“上传”都使用“同步”功能。Sysmac Studio 的“同步”功能是指自动核对电脑上的数据与实机控制器的数据，由用户选择向哪个方向传送数据的功能。

# 目录构成

<b>1</b>	<b>运动控制功能模块概要</b>	<b>10</b>	<b>示例程序</b>	<b>1</b>	<b>10</b>
<b>2</b>	<b>运动控制系统的构成与原理</b>	<b>11</b>	<b>异常的确认和处理</b>	<b>2</b>	<b>11</b>
<b>3</b>	<b>轴和轴组的设定</b>	<b>A</b>	<b>附录</b>	<b>3</b>	<b>A</b>
<b>4</b>	<b>通过Sysmac Studio 确认接线</b>	<b>I</b>	<b>索引</b>	<b>4</b>	<b>I</b>
<b>5</b>	<b>运动控制参数</b>			<b>5</b>	
<b>6</b>	<b>运动控制程序</b>			<b>6</b>	
<b>7</b>	<b>手动运行</b>			<b>7</b>	
<b>8</b>	<b>确定原点</b>			<b>8</b>	
<b>9</b>	<b>运动控制功能</b>			<b>9</b>	

# 目录

前言 .....	1
阅读对象 .....	1
对象商品 .....	1
分册构成 .....	2
手册的阅读方法 .....	3
页面构成 .....	3
图标 .....	4
关于标记的注意事项 .....	4
目录构成 .....	5
承诺事项 .....	12
安全注意事项 .....	14
安全要点 .....	15
使用注意事项 .....	16
法规与标准 .....	17
日本国外的使用 .....	17
符合 EC 指令 .....	17
符合船级标准 .....	18
软件许可证与著作权 .....	18
版本 .....	19
版本确认方法 .....	19
CPU 单元的单元版本和 Sysmac Studio 版本 .....	22
相关手册 .....	23
手册修订履历 .....	26

## 第 1 章 运动控制功能模块概要

1-1 特长 .....	1-2
1-2 系统构成 .....	1-3
1-3 使用步骤 .....	1-5
1-4 规格 .....	1-7
1-4-1 一般规格 .....	1-7
1-4-2 性能规格 .....	1-7
1-4-3 功能规格 .....	1-10

## 第 2 章 运动控制系统的构成与原理

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 .....	2-5
2-3-1 CPU 单元的任务 .....	2-5
2-3-2 运动控制的任务动作示例 .....	2-11

2-4 与 EtherCAT 通信之间的关系 .....	2-18
2-4-1 CAN application protocol over EtherCAT (CoE) .....	2-18
2-4-2 EtherCAT 主站功能与运动控制功能模块之间的关系 .....	2-19
2-4-3 过程通信周期与运动控制周期之间的关系 .....	2-21

## 第 3 章 轴和轴组的设定

3-1 轴 .....	3-2
3-1-1 轴概要 .....	3-2
3-1-2 轴参数概要 .....	3-3
3-1-3 轴变量概要 .....	3-6
3-1-4 轴变量的同步 .....	3-8
3-1-5 用户程序中轴的指定方法 .....	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 轴组概要 .....	3-19
3-3-2 轴组参数概要 .....	3-20
3-3-3 轴组变量概要 .....	3-21
3-3-4 用户程序中轴组的指定方法 .....	3-22
3-4 轴组的设定步骤 .....	3-23
3-4-1 轴组的设定流程 .....	3-23
3-4-2 设定步骤 .....	3-23

## 第 4 章 通过 Sysmac Studio 确认接线

4-1 Sysmac Studio 的功能 .....	4-2
4-1-1 MC 试运行功能一览 .....	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 试运行功能的启动步骤 .....	4-5
4-2 传感器信号的监控 .....	4-7
4-3 电机动作的确认 .....	4-8
4-3-1 伺服 ON .....	4-8
4-3-2 微动移动 .....	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 轴参数一览表 .....	5-4
5-2-2 轴基本设定 .....	5-6
5-2-3 单位转换设定 .....	5-11
5-2-4 动作设定 .....	5-17
5-2-5 扩展动作设定 .....	5-20
5-2-6 限制设定 .....	5-21
5-2-7 位置计数设定 .....	5-21
5-2-8 伺服驱动器设定 .....	5-23
5-2-9 原点复位设定 .....	5-24
5-2-10 轴参数的设定示例 .....	5-25

5-3 轴组参数 .....	5-27
5-3-1 轴组参数一览表 .....	5-27
5-3-2 轴组基本设定 .....	5-28
5-3-3 轴组动作设定 .....	5-30
5-3-4 启用轴组 .....	5-30

## 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 启动相关的通用规则 .....	6-8
6-4-2 启动的时序图 .....	6-10
6-4-3 重启运动控制指令时的时序图 .....	6-12
6-4-4 多重启动运动控制指令时的时序图 .....	6-13
6-5 位置的处理 .....	6-14
6-5-1 位置的种类 .....	6-14
6-5-2 各种轴可使用的位置的种类 .....	6-15
6-6 运动控制系统变量 .....	6-16
6-6-1 运动控制系统变量概要 .....	6-16
6-6-2 运动控制系统变量的结构 .....	6-17
6-6-3 运动控制系统变量一览 .....	6-19
6-7 凸轮表与凸轮数据变量 .....	6-29
6-8 运动控制程序的编写方法 .....	6-32
6-9 凸轮表的创建方法 .....	6-34

## 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 轴参数的设定 .....	7-4
7-2-3 程序示例 .....	7-4
7-3 微动移动 .....	7-5
7-3-1 微动移动的步骤 .....	7-5
7-3-2 轴参数的设定 .....	7-6
7-3-3 输入变量的设定示例 .....	7-6
7-3-4 程序示例 .....	7-7

## 第 8 章 确定原点

8-1 概要 .....	8-2
8-2 原点复位的步骤 .....	8-4
8-2-1 原点复位参数的设定 .....	8-5
8-2-2 原点复位的监控 .....	8-11
8-3 原点复位的动作 .....	8-12

8-4 绝对值编码器原点设定 .....	8-13
8-4-1 功能概要 .....	8-14
8-4-2 设定步骤 .....	8-15
8-5 高速原点复位 .....	8-16

## 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 中断标准定位 .....	9-5
9-1-5 周期同步定位 .....	9-6
9-1-6 停止 .....	9-6
9-1-7 超调 .....	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 .....	9-22
9-2-7 加减运算定位 .....	9-23
9-2-8 主轴相位补偿 .....	9-24
9-2-9 从轴位置补偿 .....	9-25
9-2-10 多任务运动中实现同步控制的方法 .....	9-25
9-3 单轴速度控制 .....	9-27
9-3-1 速度控制 .....	9-27
9-3-2 周期同步速度控制 .....	9-28
9-4 单轴转矩控制 .....	9-29
9-5 单轴控制的通用功能 .....	9-30
9-5-1 位置 .....	9-30
9-5-2 速度 .....	9-32
9-5-3 加速与减速 .....	9-33
9-5-4 跃度 .....	9-35
9-5-5 指定动作方向 .....	9-36
9-5-6 运动控制指令的重启 .....	9-39
9-5-7 运动控制指令的多重启动 (缓存模式) .....	9-44
9-6 多轴协调控制 .....	9-49
9-6-1 动作概要 .....	9-49
9-6-2 直线插补 .....	9-52
9-6-3 圆弧插补 .....	9-53
9-6-4 轴组周期同步位置控制 .....	9-53
9-6-5 多轴协调控制中的停止 .....	9-54
9-6-6 多轴协调控制中的超调 .....	9-56
9-7 多轴协调控制的通用功能 .....	9-57
9-7-1 多轴协调控制中的速度 .....	9-57
9-7-2 多轴协调控制中的加速和减速 .....	9-58
9-7-3 多轴协调控制中的跃度 .....	9-59
9-7-4 多轴协调控制的运动控制指令的重启 .....	9-60
9-7-5 多轴协调控制的运动控制指令的多重启动 (缓存模式) .....	9-61
9-8 其他功能 .....	9-68
9-8-1 当前位置变更功能 .....	9-68
9-8-2 转矩限制功能 .....	9-69
9-8-3 锁定功能 .....	9-69
9-8-4 区域功能 .....	9-70
9-8-5 软件限制功能 .....	9-71
9-8-6 位置偏差监控功能 .....	9-72

9-8-7 偏差计数器复位 .....	9-73
9-8-8 轴间偏差监控功能 .....	9-73
9-8-9 到位检查 .....	9-74
9-8-10 轴使用变更功能 .....	9-76
9-8-11 数字凸轮开关有效功能 .....	9-76
9-8-12 任意坐标系的 3D 运动监视显示 .....	9-77

## 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 设置和接线 .....	10-2
10-1-3 设定 .....	10-2
10-2 基本程序示例 .....	10-3
10-2-1 EtherCAT 通信的监控和伺服 ON .....	10-3
10-2-2 通过主站控制指令进行轴动作的联锁 .....	10-4
10-2-3 异常的监控和异常解除 (单轴动作、同步动作) .....	10-6
10-2-4 异常的监控和异常解除 (多轴协调动作) .....	10-8
10-2-5 指令异常的监控方法 .....	10-14
10-2-6 异常解除的监控方法 .....	10-16
10-2-7 单轴动作过程中的轴停止 .....	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 通过重启指令变更目标位置 .....	10-33
10-2-11 中断标准定位 .....	10-38
10-2-12 通过重启指令切换凸轮表 .....	10-42
10-2-13 使用凸轮曲线补偿同步位置 .....	10-51
10-2-14 凸轮动作时补偿主轴的相位 .....	10-60
10-2-15 速度控制中变更当前位置 .....	10-68
10-2-16 变更凸轮数据变量和保存凸轮表 .....	10-74
10-2-17 临时改写轴参数 .....	10-83
10-2-18 更新凸轮表的终点索引 .....	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 .....	11-3
11-1-2 MC 功能模块相关的异常 .....	11-6
11-2 故障诊断 .....	11-10
11-2-1 异常一览表 .....	11-10
11-2-2 异常内容 .....	11-16
11-2-3 异常现象的推测原因和处理 .....	11-44

## 附录

A-1 与伺服驱动器 1S 系列之间的连接 .....	A-2
A-1-1 与伺服驱动器之间的接线 .....	A-2
A-1-2 伺服驱动器的设定 .....	A-2
A-2 与伺服驱动器 G5 系列之间的连接 .....	A-10
A-2-1 与伺服驱动器之间的接线 .....	A-10
A-2-2 伺服驱动器的设定 .....	A-10
A-3 与编码器输入终端之间的连接 .....	A-19
A-3-1 与编码器输入终端之间的接线 .....	A-19
A-3-2 编码器输入终端的设定 .....	A-19
A-4 与 NX 单元之间的连接 .....	A-25

A-5 PDS 状态切换.....	A-26
A-5-1 PDS 状态控制方式.....	A-27
A-5-2 主电路电源 OFF 检测 .....	A-28
A-6 用语说明 .....	A-29
A-6-1 NJ/NX 系列通用 .....	A-29
A-6-2 运动控制功能.....	A-30
A-6-3 EtherCAT 通信.....	A-31
A-7 版本相关信息.....	A-32

## 索引

---

# 承诺事项

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

## ● 定义

本承诺事项中用语的定义如下所示。

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

## ● 记载内容的注意事项

关于“产品样本等”中的内容，请注意以下几点。

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

## ● 使用注意事项

使用时，请注意以下几点。

- 使用时请符合额定值、性能以及“使用条件等”。
- 请用户自行确认“适用性等”，判断是否可使用“本公司产品”。  
“本公司”对“适用性等”不作任何保证。
- 用户将“本公司产品”用于整个系统时，请务必事先自行确认配电、设置是否恰当。
- 使用“本公司产品”时，请注意以下各事项。(i)使用“本公司产品”时，应在额定值和性能方面留有余量，采用冗余设计等安全设计，(ii)采用安全设计，即使“本公司产品”发生故障，也可将“用户用途”造成的危险降至最低程度，(iii)对整个系统采取安全措施，以便向使用者告知危险，(iv)定期维护“本公司产品”及“用户用途”。
- “本公司产品”是本公司设计并制造面向一般工业产品的通用产品。但是，不可用于以下用途。如果用户将“本公司产品”用于以下用途，则“本公司”不对“本公司产品”作任何保证。但经“本公司”许可后用于以下用途或与“本公司”签订特殊协议的情况除外。
  - (a) 需高安全性的用途（例：原子能控制设备、燃烧设备、航空航天设备、铁路设备、起重设备、娱乐设备、医疗设备、安全装置以及其他危及生命、健康的用途）
  - (b) 需高可靠性的用途（例：煤气、自来水、电力等供应系统、24小时连续运行的系统、支付系统等涉及权利、财产的用途等）
  - (c) 用于严格条件或环境下（例：需设置在室外的设备、会受化学污染的设备、会受电磁波干扰的设备、会受振动和冲击影响的设备等）
  - (d) 在“产品样本等”中未记载的条件或环境下使用
- 上述(a)~(d)以及“本产品样本等中记载的产品”不可用于汽车（含两轮车。下同）。请勿装入汽车进行使用。关于可装入汽车的产品，请咨询本公司销售负责人。

## ● 保修条件

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

## ● 责任免除

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

## ● 出口管理

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

# 安全注意事项

---

关于安全注意的内容，请参阅如下手册。

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

# 安全要点

---

关于安全要点的内容，请参阅如下手册。

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

# 使用注意事项

---

关于使用注意事项的内容，请参阅如下手册。

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

# 法规与标准

## 日本国外的使用

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

## 符合 EC 指令

### 符合指令

- EMC 指令
- 低电压指令

### 适用途径

#### ● EMC 指令

欧姆龙的产品为装入各种机械、制造装置使用的电气设备，为使装入的机械、装置更容易符合 EMC 标准，产品自身需符合相关 EMC 标准 (\*1)。

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

- \*1 EMC (Electro-Magnetic Compatibility: 电磁环境兼容性) 相关标准中，  
与 EMS (Electro-Magnetic Susceptibility: 电磁敏感性) 相关的为 EN61131-2；  
与 EMI (Electro-Magnetic Interference: 电磁干扰) 相关的为 EN61131-2。  
此外，Radiated emission 依照 10m 法。

#### ● 低电压指令

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

#### ● 符合 EU 指令

NJ/NX 系列符合 EU 指令。要使客户的机械、装置符合 EU 指令，需注意以下事项。

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

## 符合船级标准

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

## 各船级标准的使用条件 (NK、LR)

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

### 噪声滤波器

厂家	型号
COSEL 公司	TAH-06-683

## 软件许可证与著作权

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

# 版本

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

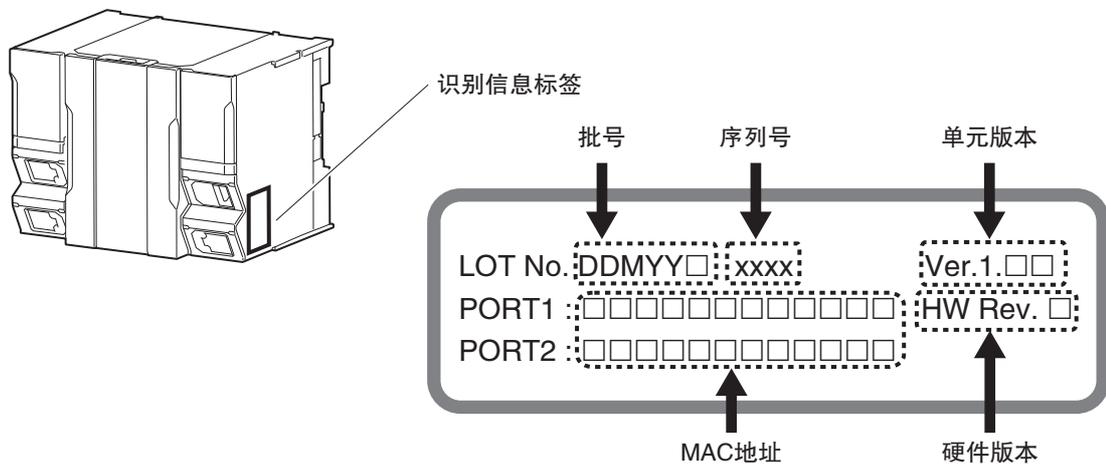
## 版本确认方法

版本可通过识别信息标签或 Sysmac Studio 进行确认。

### 通过识别信息标签确认

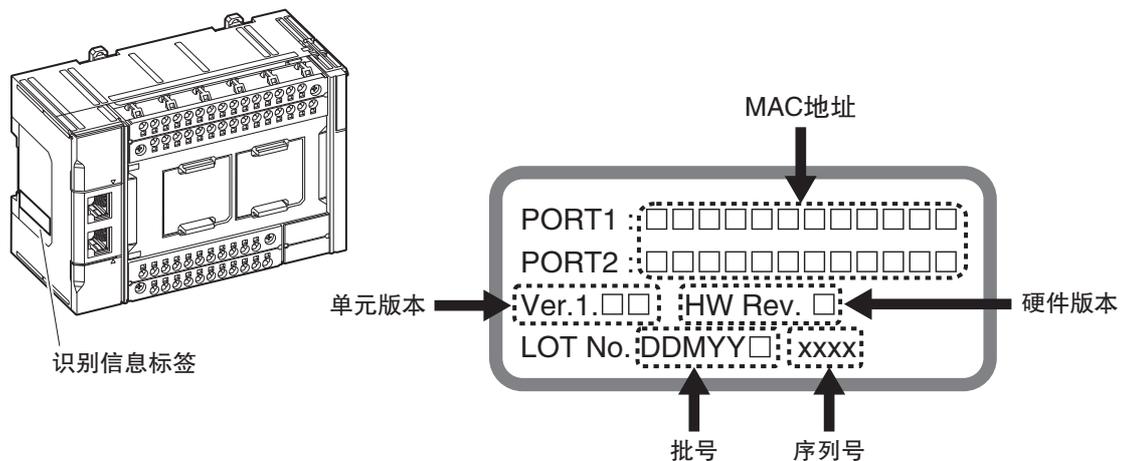
版本可通过产品侧面的识别信息标签进行确认。

NX 系列 CPU 单元 NX701-□□□□ 的识别信息标签如下图所示。



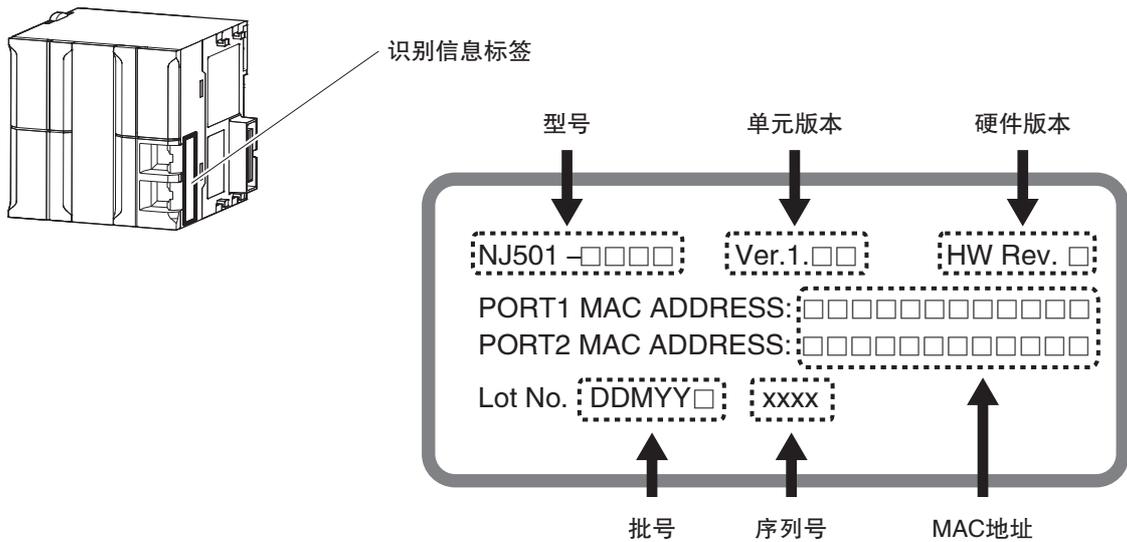
(注) 硬件版本为“无”的单元不显示硬件版本。

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



(注) 硬件版本为“无”的单元不显示硬件版本。

NJ 系列 CPU 单元 NJ501-□□□□ 的识别信息标签如下图所示。



(注) 硬件版本为“无”的单元不显示硬件版本。

## 基于 Sysmac Studio 的确认方法

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

### ● NX 系列 CPU 单元的版本确认方法

单元版本可通过在线状态下的 [ 生产信息 ] 确认。可确认版本的单元为 CPU 单元、CPU 机架上的 NX 单元及扩展板。

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

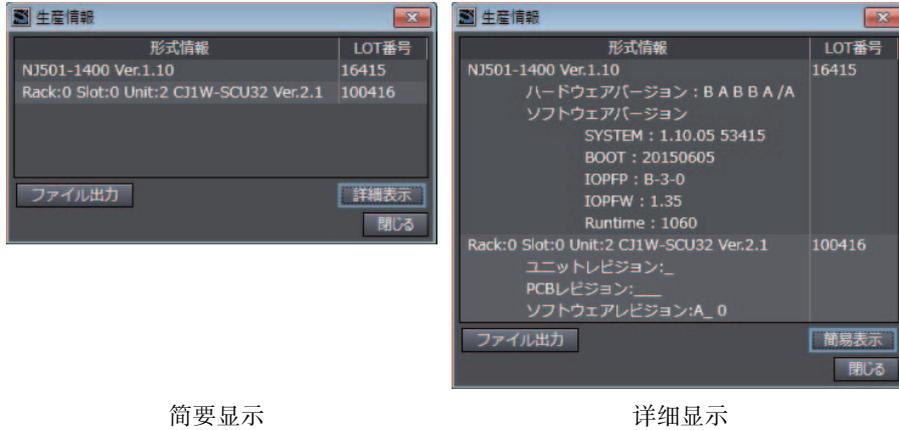
### ● NJ 系列 CPU 单元的版本确认方法

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

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

● 生产信息显示内容的切换

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



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

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

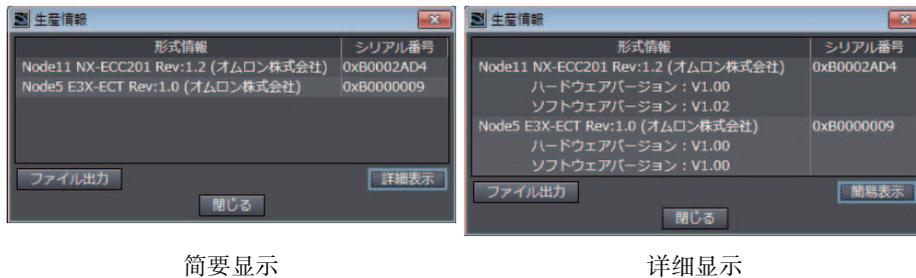
● EtherCAT 从站版本确认方法

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

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

● 生产信息显示内容的切换

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



## CPU 单元的单元版本和 Sysmac Studio 版本

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

CPU 单元的单元版本的种类和 Sysmac Studio 版本之间的关系，请参阅 □ “NJ/NX 系列 CPU 单元 用户手册 软件篇（SBCA-359）”。

关于不同单元版本支持功能一览，请参阅 □ “A-7 版本相关信息 (P.A-32)”。

# 相关手册

与本手册相关的手册如下表所述。请同时参阅。

手册名称	Man.No.	型号	用途	内容
NX 系列 CPU 单元 用户手册 硬件篇	SBCA-418	NX701- □□□□	希望了解 NX701 CPU 单元的概要 / 设计 / 安装 / 保养等基本规格时。 与硬件相关的信息为主。	对 NX701 的系统整体概要和 CPU 单元进行以下内容的说明。 · 特长和系统构成 · 概要 · 各部分的名称和功能 · 一般规格 · 安装与接线 · 维护检查
NX 系列 NX1P2 CPU 单元 用户手册 硬件篇	SBCA-448	NX1P2- □□□□	希望了解 NX1P2 CPU 单元的概要 / 设计 / 安装 / 保养等基本规格时。 与硬件相关的信息为主。	对 NX1P2 的系统整体概要和 CPU 单元进行以下内容的说明。 · 特长和系统构成 · 概要 · 各部分的名称和功能 · 一般规格 · 安装与接线 · 维护检查
NJ 系列 CPU 单元 用户手册 硬件篇	SBCA-358	NJ501- □□□□ NJ301- □□□□ NJ101- □□□□	希望了解 NJ 系列 CPU 单元的概要 / 设计 / 安装 / 保养等基本规格时。 与硬件相关的信息为主。	对 NJ 系列的系统整体概要和 CPU 单元进行以下内容的说明。 · 特长和系统构成 · 概要 · 各部分的名称和功能 · 一般规格 · 安装与接线 · 维护检查
NJ/NX 系列 CPU 单元 用户手册 软件篇	SBCA-359	NX701- □□□□ NX1P2- □□□□ NJ501- □□□□ NJ301- □□□□ NJ101- □□□□	NJ/NX 系列 希望了解 CPU 单元的编程 / 系统调试时。 与软件相关的信息为主。	对 NJ/NX 系列的 CPU 单元进行以下内容的说明。 · CPU 单元的动作 · CPU 单元的功能 · 初始设定 · 符合 IEC 61131-3 标准的语言规格和编程
NX 系列 NX1P2 CPU 单元 用户手册 内置 I/O、扩展板 功能篇	SBCA-449	NX1P2- □□□□	NX 系列 希望了解 NX1P2 CPU 单元独有功能的详情和 NJ/NX 系列功能的概要时。	对 NX1P2 CPU 单元功能中的以下内容进行说明。 · 内置 I/O · 串行通信扩展板 · 对模拟输入输出用扩展板以及 NJ/NX 系列 CPU 单元的以下功能概要进行说明。 · 运动控制功能 · EtherNet/IP 通信功能 · EtherCAT 通信功能
NJ/NX 系列 指令基准 手册 基本篇	SBCA-360	NX701- □□□□ NX1P2- □□□□ NJ501- □□□□ NJ301- □□□□ NJ101- □□□□	希望了解 NJ/NX 系列的基本指令规格的详情时。	对各指令（IEC 61131-3 标准）的详情进行说明。
NJ/NX 系列 CPU 单元 用户手册 运动控制篇	SBCE-363	NX701- □□□□ NX1P2- □□□□ NJ501- □□□□ NJ301- □□□□ NJ101- □□□□	希望了解运动控制的设定和编程思路时。	对用于运动控制的 CPU 单元的设定、动作及编程思路进行说明。
NJ/NX 系列 指令基准 手册 运动篇	SBCE-364	NX701- □□□□ NX1P2- □□□□ NJ501- □□□□ NJ301- □□□□ NJ101- □□□□	希望了解运动指令规格的详情时。	对各运动指令的详情进行说明。
NJ/NX 系列 CPU 单元 内置 EtherCAT® 端口 用户手册	SBCD-358	NX701- □□□□ NX1P2- □□□□ NJ501- □□□□ NJ301- □□□□ NJ101- □□□□	NJ/NX 系列 使用 CPU 单元的内置 EtherCAT 端口时。	对内置 EtherCAT 端口进行说明。 对概要、构成、功能、安装进行描述。

手册名称	Man.No.	型号	用途	内容
NJ/NX 系列 CPU 单元 内置 EtherNet/IP™ 端口 用户手册	SBCD-359	NX701- □□□□ NX1P2- □□□□ NJ501- □□□□ NJ301- □□□□ NJ101- □□□□	NJ/NX 系列 使用 CPU 单元的内置 EtherNet/IP 端口时。	对内置 EtherNet/IP 端口进行说明。对基本设定、标签数据链接及其他功能进行描述。
NJ 系列 数据库连接 CPU 单元 用户手册	SBCA-411	NJ501-1 □ 20 NJ101- □□ 20	在 NJ 系列中使用数据库连接服务功能时。	对数据库连接服务功能进行说明。
NJ 系列 配备 SECS/GEM 的 CPU 单元 用户手册	SBCA-412	NJ501-1340	在 NJ 系列中使用 GEM 服务功能时。	对 GEM 服务功能进行说明。
NJ 系列 NJ Robotics CPU 单元 用户手册	SBCA-421	NJ501-4 □□□	在 NJ 系列中进行机器人控制时。	对机器人控制功能进行说明。
NJ/NX 系列 故障诊断 手册	SBCA-361	NX701- □□□□ NX1P2- □□□□ NJ501- □□□□ NJ301- □□□□ NJ101- □□□□	希望了解通过 NJ/NX 系列检测异常的详情时。	对通过 NJ/NX 系列系统检测的异常管理的途径和各异常项目进行说明。
Sysmac Studio Version 1 操作手册	SBCA-362	SYSMAC-SE2 □□□	希望了解 Sysmac Studio 的操作方法、功能时。	对 Sysmac Studio 的操作方法进行说明。
NX 系列 EtherCAT® 耦合器单元 用户手册	SBCD-361	NX-ECC □□□	NX 系列希望了解 EtherCAT 耦合器单元和 EtherCAT 从站终端的使用方法时。	对由 NX 系列 EtherCAT 耦合器单元和 NX 单元构成的 EtherCAT 从站终端的系统概要和构成方法，以及经由 EtherCAT 对 NX 单元进行设定、控制、监控的 EtherCAT 耦合器单元的硬件、设定方法及功能进行说明。
NX 系列 数据基准 手册	SBCA-410	NX- □□□□□□	希望通过一览表查看 NX 系列各单元的系统构成所需的数据时。	汇总了 NX 系列各单元的“消耗功率”、“重量”等系统构建所需的数据。
NX 系列 NX 单元 用户手册	SBCA-407	NX-ID □□□□ NX-IA □□□□ NX-OC □□□□ NX-OD □□□□ NX-MD □□□□	希望了解 NX 单元的使用方法时。	对 NX 单元的硬件、设定方法及功能进行说明。备有以下单元的手册。 数字 I/O 单元、模拟 I/O 单元、系统单元、位置接口单元、通信接口单元、负载传感器输入单元、IO-Link 主站单元
	SBCA-408	NX-AD □□□□ NX-DA □□□□		
	SBCA-440	NX-TS □□□□ NX-HB □□□□		
	SBCA-409	NX-PD1 □□□ NX-PF0 □□□ NX-PC0 □□□ NX-TBX01		
	SBCE-374	NX-ECO □□□ NX-ECS □□□ NX-PG0 □□□		
	SBCA-422	NX-CIF □□□		
	SBCA-439	NX-RS □□□□		
	SBCD-370	NX-ILM □□□		
EtherCAT 远程 I/O 终端 GX 系列 EtherCAT 从站 用户手册	SBCD-350	GX-ID □□□□ GX-OD □□□□ GX-OC □□□□ GX-MD □□□□ GX-AD □□□□ GX-DA □□□□ GX-EC □□□□ XWT-ID □□ XWT-OD □□	希望了解 EtherCAT 远程 I/O 终端的使用方法时。	对 EtherCAT 远程 I/O 终端的硬件、设定方法及功能进行说明。
AC 伺服电机 / 驱动器 IS 系列 EtherCAT® 通信内置型 用户手册	SBCE-377	R88M-1 □ R88D-1SN □ -ECT	希望了解 EtherCAT 通信内置型伺服电机 / 驱动器的使用方法时。	对 EtherCAT 通信内置型伺服电机 / 驱动器的硬件和设定方法、功能进行说明。

手册名称	Man.No.	型号	用途	内容
AC 伺服电机 / 驱动器 G5 系列 (EtherCAT® 通信内置型) 用户手册	SBCE-360	R88M-K □ R88D-KN □ -ECT-R	希望了解 EtherCAT 通信内置型伺服电机 / 驱动器的使用方法时。	对 EtherCAT 通信内置型伺服电机 / 驱动器的硬件和设定方法、功能进行说明。 G5 系列有直线电机型和位置控制限定型。
	SBCE-365	R88M-K □ R88D-KN □ -ECT		
	SBCE-366	R88L-EC- □ R88D-KN □ -ECT-L		

# 手册修订履历

手册的修订记号附加在封面和封底的 Man.No. 的末尾。

**Man.No. SBCE-CN5-363R**

↑  
修订记号

修订记号	修订年月	修订理由、修订页
A	2011年7月	初版
B	2012年1月	NJ301- □□□□的追加 CPU 单元升级为 Ver.1.01 所伴随的修订
C	2012年5月	CPU 单元升级为 Ver.1.02 所伴随的修订 错误修正
D	2012年8月	CPU 单元升级为 Ver.1.03 所伴随的修订
E	2013年2月	CPU 单元升级为 Ver.1.04 所伴随的修订
F	2013年4月	CPU 单元升级为 Ver.1.05 所伴随的修订 勘误
G	2013年6月	CPU 单元升级为 Ver.1.06 所伴随的修订 勘误
H	2013年12月	CPU 单元升级为 Ver.1.08 所伴随的修订 勘误
J	2014年7月	CPU 单元升级为 Ver.1.09 所伴随的修订 勘误
K	2015年1月	CPU 单元升级为 Ver.1.10 所伴随的修订 勘误
L	2015年4月	追加 NX 系列 CPU 单元 NX701- □□□及 NJ 系列 CPU 单元 NJ101-10 □□ 所伴随的修订 勘误
M	2016年4月	CPU 单元升级为 Ver.1.11 所伴随的修订 勘误
N	2016年7月	CPU 单元升级为 Ver.1.12 所伴随的修订 勘误
P	2016年10月	追加 NX 系列 NX1P2 CPU 单元所伴随的修订 CPU 单元升级为 Ver.1.13 所伴随的修订 勘误
R	2017年1月	勘误

# 1

## 运动控制功能模块概要

下面对运动控制功能模块的特长和系统构成、使用流程进行说明。

---

1-1 特长 .....	1-2
1-2 系统构成 .....	1-3
1-3 使用步骤 .....	1-5
1-4 规格 .....	1-7
1-4-1 一般规格 .....	1-7
1-4-2 性能规格 .....	1-7
1-4-3 功能规格 .....	1-10

## 1-1 特长

运动控制功能模块(以下有时简称“MC 功能模块”)是指, CPU 单元内置软件的功能模块。MC 功能模块通过 CPU 单元的内置 EtherCAT 端口, 最多可执行 256 轴的运动控制。通过与 EtherCAT 端口连接的伺服驱动器建立周期通信, 可实现要求高速及高精度的机器控制。

### 依据 PLCopen® 的运动控制命令

MC 功能模块的运动控制指令以通过 PLCopen® 实现标准化的运动控制用功能块为基础。除了单轴 PTP 定位、插补控制、电子凸轮等同步控制, 通过依据标准的指令还可对速度控制及转矩控制进行编程。另外, 还可对各运动控制指令的启动设定速度、加速度、减速度及跃度, 因此可实现基于应用的灵活动作控制。



#### 参考

- 何谓 PLCopen®

PLCopen® 是总部在欧洲的 IEC 61131-3 推广团体, 是一个全球性的会员组织。PLCopen® 对运动控制用功能块进行了标准化, 制定了 IEC 61131-3(JISB 3503) 规格语言程序接口的定义。

- 何谓跃度

跃度是指加速度 / 减速度的变化比率, 也称为“跃动”、“跃度”或“加加速度”。如果指定跃度, 则加减速时的速度波形变为 S 型。

### 通过 EtherCAT 通信传送数据

通过与欧姆龙制 1S 系列或 G5 系列伺服驱动器 EtherCAT 通信内置型组合, 可通过高速数据通信交换所有控制信息。使用数据通信传输各种控制指令, 从而不受编码器反馈脉冲响应频率等接口规格的限制, 将伺服电机的性能发挥至最大极限。另外, 在高位控制器侧可处理伺服驱动器的各种控制参数及监控信息, 从而实现系统信息管理的一元化。



#### 参考

#### 何谓 EtherCAT

EtherCAT 是指, 基于 Ethernet(IEEE802.3) 的打开实现超高速产业用网络系统。各节点以高速传送以太网帧, 因此可实现较短通信周期。另外, 通过共享时钟信息的结构, 可实现较低通信抖动的高精度同步控制。

## 1-2 系统构成

MC 功能模块接收来自装置和控制柜的传感器信号状态和来自用户程序的运动控制指令的指令，执行伺服驱动器、编码器输入终端及 NX 系列位置接口单元的运动控制。

### EtherCAT 网络中的运动控制的结构

MC 功能模块使用 EtherCAT 网络结构、EtherCAT 耦合器单元的从站终端结构，以及 Sysmac Studio。

#### ● EtherCAT 网络构成

MC 功能模块使用 CPU 单元内置的 EtherCAT 通信主站功能端口，控制伺服驱动器、编码器输入终端。使用该构成，可在固定周期内实现波动较少的高精度运动控制。

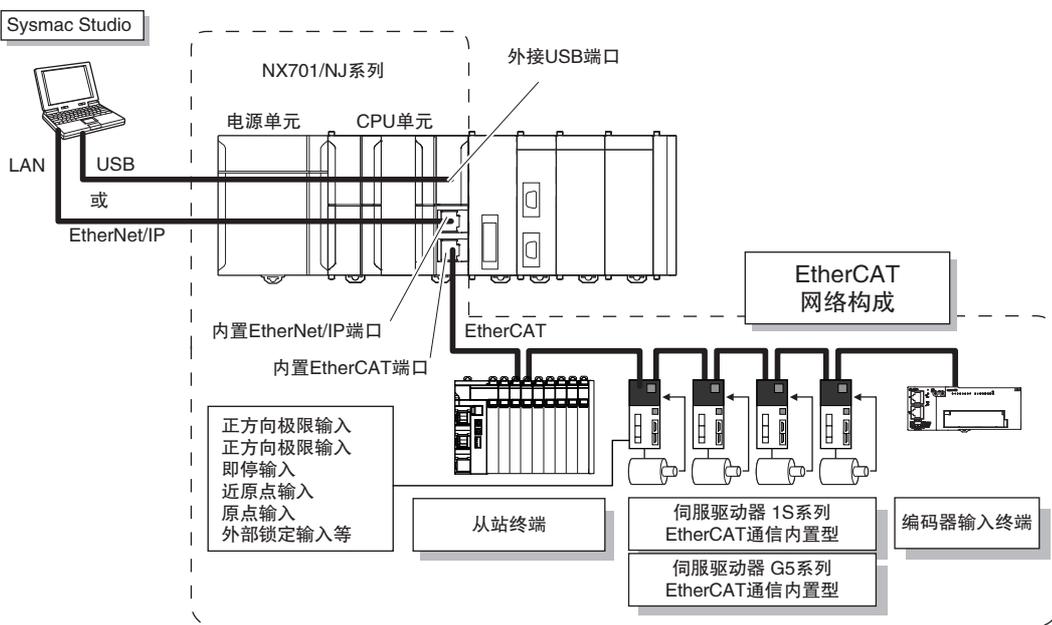
#### ● EtherCAT 耦合器单元的从站终端结构

MC 功能模块使用 EtherCAT 耦合器单元上安装的位置接口单元，进行电机控制用脉冲输出及编码器输入的导入。

使用该构成，可在固定周期内实现波动较少的高精度运动控制。

#### ● Sysmac Studio

通过市售 USB 电缆连接至 CPU 单元的外接 USB 端口。或通过 Ethernet 电缆连接至 CPU 单元内置的 EtherNet/IP 端口。



#### 使用注意事项

使用 NX 系列位置接口单元时，运动控制功能的部分功能存在差别。详情请参阅 □□ “NX 系列位置接口单元 用户手册 (SBCE-374)”。



#### 版本相关信息

NX 系列位置接口单元可组合使用 CPU 单元 Ver.1.05 以上版本与 Sysmac Studio Ver.1.06 以上版本。

## CPU 单元上的运动控制结构

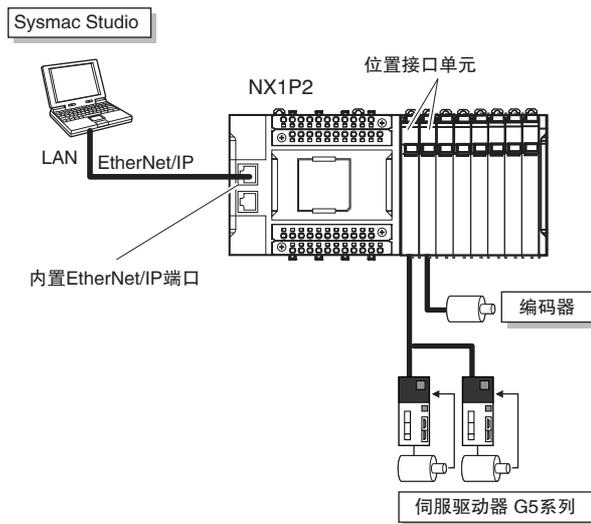
MC 功能模块中使用位置接口单元及 Sysmac Studio。

- 位置接口单元

使用位置接口单元，进行电机控制用脉冲输出及编码器输入的导入。

- Sysmac Studio

通过 Ethernet 电缆连接至 CPU 单元内置的 EtherNet/IP 端口。

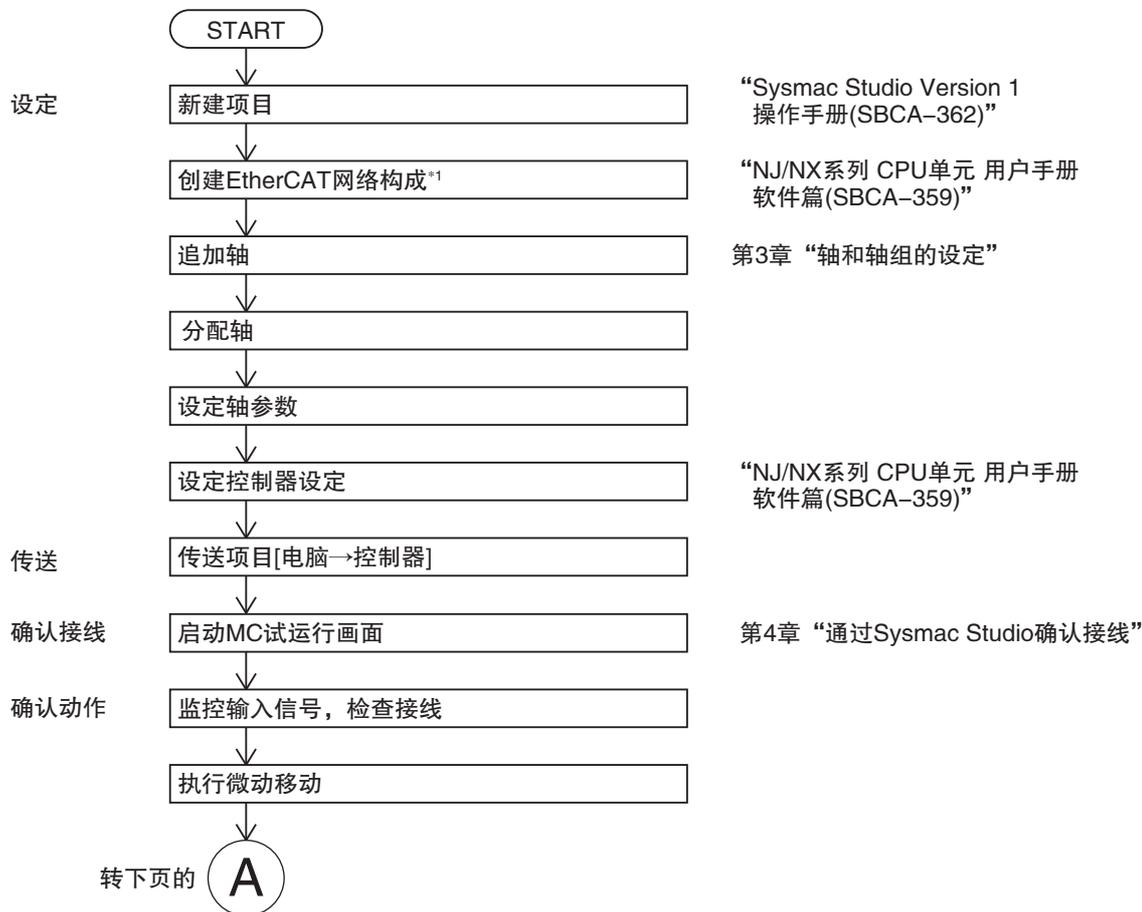


### 使用注意事项

使用 NX 系列位置接口单元时，运动控制功能的部分功能存在差别。详情请参阅 □ “NX 系列位置接口单元 用户手册 (SBCE-374)”。

## 1-3 使用步骤

下面对使用 MC 功能模块执行运动控制的基本使用步骤进行说明。

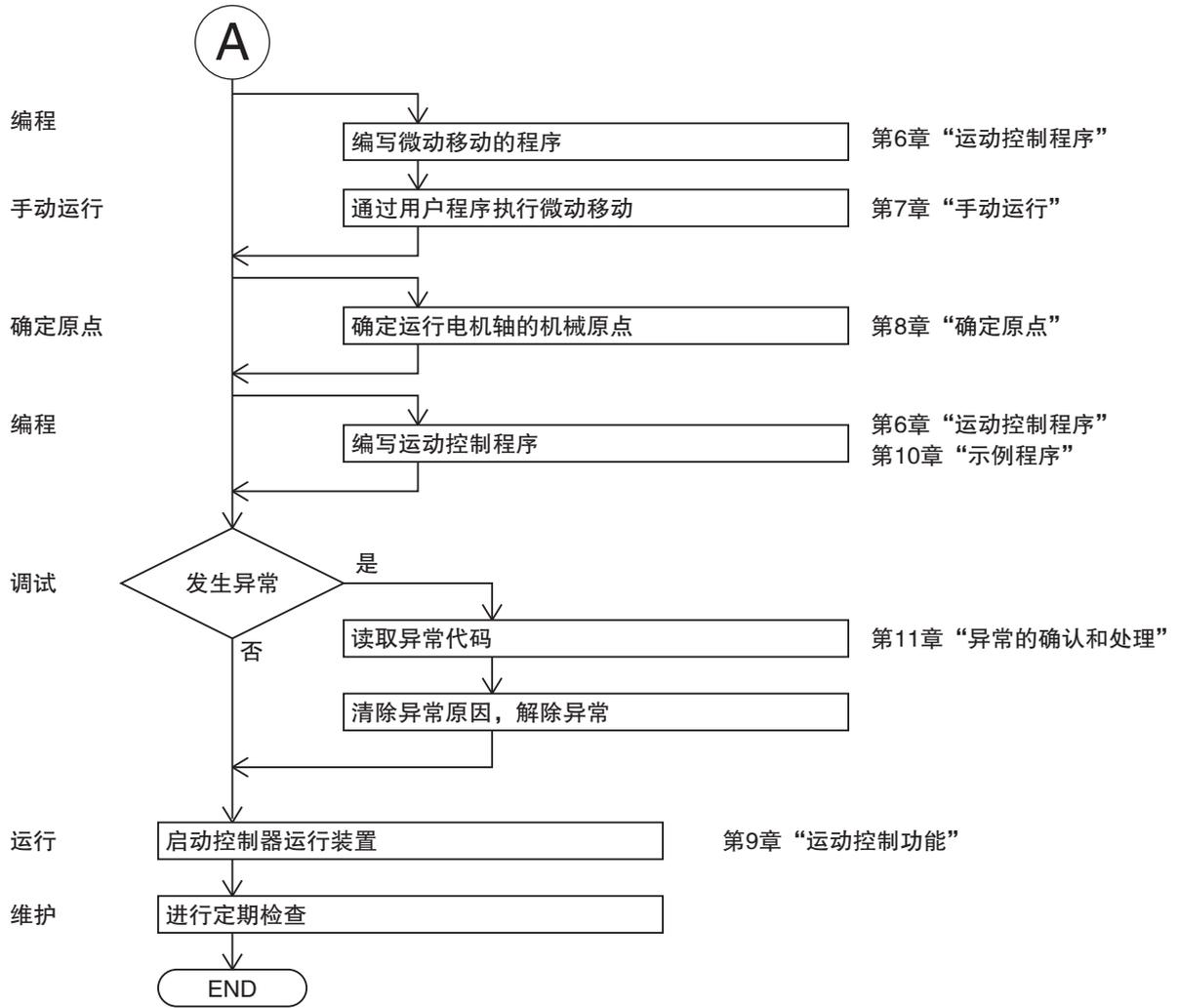


\*1 连接实机时，可在线设定实机构成。  
无实机时，可在线选择并设定所用 EtherCAT 从站。



### 参考

使用 NX 系列位置接口单元时的使用步骤请参阅 “NX 系列位置接口单元 用户手册 (SBCE-374)”。



## 1-4 规格

下面对 MC 功能模块的规格进行说明。



### 使用注意事项

NJ101-90 □□没有运动控制功能模块。

### 1-4-1 一般规格

一般规格依据 CPU 单元的一般规格。

详情请参阅 □□ “NX 系列 CPU 单元 用户手册 硬件篇 (SBCA-418)” 或 “NJ 系列 CPU 单元 用户手册 硬件篇 (SBCA-358)”。

### 1-4-2 性能规格

对各 CPU 单元的性能规格进行说明。

规格项目		规格 (NX701-)		
		17 □□	16 □□	
控制轴数	控制轴最大数量 *1	256 轴	128 轴	
		运动控制轴 *2	256 轴	128 轴
		单轴位置控制轴 *3	-	
	使用实轴最大数量 *4	256 轴	128 轴	
		使用运动控制伺服轴	256 轴	128 轴
		使用单轴位置控制伺服轴	-	
直线插补控制最大轴数		每轴组 4 轴		
圆弧插补控制轴数		每轴组 2 轴		
最大轴组数		64 组		
超调		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		
多任务运动		可		
凸轮	凸轮数据点数	每个凸轮表的最大点数	65,535 点	
		所有凸轮表的最大点数	1,048,560 点	
	凸轮表的最大表数	640 个表		

\*1. 所有轴种类的合计。轴种类的详情请参阅 □□ “轴种类 (P.5-8)”。

\*2. 可使用运动控制功能的所有功能的轴。

\*3. 运动控制功能中，仅可使用“单轴位置控制”的轴。单轴位置控制轴的详情请参阅 □□ “控制功能 (P.5-9)”。

\*4. 轴种类为 [ 伺服轴 ] 或 [ 编码器轴 ]，且轴使用设为 [ 使用轴 ] 的轴的合计。

# 1 运动控制功能模块概要

规格项目		规格 (NX1P2-)			
		11 □□	10 □□	90 □□	
控制轴数	控制轴最大数量 *1	12 轴	10 轴	4 轴	
		运动控制轴 *2	8 轴	6 轴	-
		单轴位置控制轴 *3	4 轴		
	使用实轴最大数量 *4	8 轴	6 轴	4 轴	
		使用运动控制伺服轴	4 轴	2 轴	-
		使用单轴位置控制伺服轴	4 轴		
	直线插补控制最大轴数	每轴组 4 轴		-	
	圆弧插补控制轴数	每轴组 2 轴		-	
最大轴组数	8 组		-		
超调	0.00、0.01 ~ 500.00%				
运动控制周期	与原始恒定周期任务的周期相同				
多任务运动	不可		-		
凸轮	凸轮数据点数	每个凸轮表的最大点数	65,535 点	-	
		所有凸轮表的最大点数	262,140 点	-	
	凸轮表的最大表数	80 个表		-	

\*1. 所有轴种类的合计。轴种类的详情请参阅 □□ “轴种类 (P.5-8)”。

\*2. 可使用运动控制功能的所有功能的轴。

\*3. 运动控制功能中，仅可使用“单轴位置控制”的轴。单轴位置控制轴的详情请参阅 □□ “控制功能 (P.5-9)”。

\*4. 轴种类为 [ 伺服轴 ] 或 [ 编码器轴 ]，且轴使用设为 [ 使用轴 ] 的轴的合计。

规格项目		规格 (NJ501-)			
		□ 5 □□	□ 4 □□	□ 3 □□	
控制轴数	控制轴最大数量 *1	64 轴	32 轴	16 轴	
		运动控制轴 *2	64 轴	32 轴	16 轴
		单轴位置控制轴 *3	-		
	使用实轴最大数量 *4	64 轴	32 轴	16 轴	
		使用运动控制伺服轴	64 轴	32 轴	16 轴
		使用单轴位置控制伺服轴	-		
	直线插补控制最大轴数	每轴组 4 轴		-	
	圆弧插补控制轴数	每轴组 2 轴		-	
最大轴组数	32 组		-		
超调	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				
多任务运动	不可		-		
凸轮	凸轮数据点数	每个凸轮表的最大点数	65,535 点	-	
		所有凸轮表的最大点数	1,048,560 点	-	
	凸轮表的最大表数	640 个表		-	

\*1. 所有轴种类的合计。轴种类的详情请参阅 □□ “轴种类 (P.5-8)”。

\*2. 可使用运动控制功能的所有功能的轴。

\*3. 运动控制功能中，仅可使用“单轴位置控制”的轴。单轴位置控制轴的详情请参阅 □□ “控制功能 (P.5-9)”。

\*4. 轴种类为 [ 伺服轴 ] 或 [ 编码器轴 ]，且轴使用设为 [ 使用轴 ] 的轴的合计。

规格项目		规格 (NJ301-)		规格 (NJ101-)
		12 □□	11 □□	10 □□
控制轴数	控制轴最大数量 *1	15 轴 *2*3	15 轴 *2*4	6 轴
		运动控制轴 *5	15 轴 *2*3	6 轴
		单轴位置控制轴 *6	-	
	使用实轴最大数量 *7	8 轴	4 轴	2 轴
		使用运动控制伺服轴	8 轴	2 轴
		使用单轴位置控制伺服轴	-	
	直线插补控制最大轴数	每轴组 4 轴		
圆弧插补控制轴数	每轴组 2 轴			
最大轴组数	32 组			
超调	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			
多任务运动	不可			
凸轮	凸轮数据 点数	每个凸轮表的最大点数	65,535 点	
		所有凸轮表的最大点数	262,140 点	
	凸轮表的最大表数	160 个表		

\*1. 所有轴种类的合计。轴种类的详情请参阅 □□ “轴种类 (P.5-8)”。

\*2. CPU 单元版本升级后追加的功能。版本升级的相关信息请参阅 □□ “A-7 版本相关信息 (P.A-32)”。

\*3. Ver.1.06 以上的 CPU 单元和 Ver.1.07 以上的 Sysmac Studio 组合时的轴数。其它情况下为 8 轴。

\*4. Ver.1.06 以上的 CPU 单元和 Ver.1.07 以上的 Sysmac Studio 组合时的轴数。其它情况下为 4 轴。

\*5. 可使用运动控制功能的所有功能的轴。

\*6. 运动控制功能中，仅可使用“单轴位置控制”的轴。单轴位置控制轴的详情请参阅 □□ “控制功能 (P.5-9)”。

\*7. 轴种类为 [ 伺服轴 ] 或 [ 编码器轴 ]，且轴使用设为 [ 使用轴 ] 的轴的合计。

## 1-4-3 功能规格

下面介绍与欧姆龙制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		欧姆龙制伺服驱动器 1S 系列或 G5 系列 (EtherCAT 通信内置型) <sup>*1*2</sup>	
控制对象编码器输入终端		欧姆龙制 EtherCAT 远程 I/O 终端 GX 系列 GX-EC0211/EC0241 <sup>*3</sup>	
控制对象位置接口单元		欧姆龙制增量编码器输入单元 NX-ECO □□□ 欧姆龙制 SSI 输入单元 NX-ECS □□□ 欧姆龙制脉冲输出单元 NX-PG0 □□□	
控制方式		发出基于 EtherCAT 通信的控制指令	
控制模式		位置控制 (cyclic synchronous position) 速度控制 (cyclic synchronous velocity) 转矩控制 (cyclic synchronous torque)	
单位转换	表示单位	脉冲、mm、 $\mu\text{m}$ 、nm、degree、inch	
	电子齿轮比	电机转 1 圈的脉冲数 / 电机转 1 圈的移动量或 (电机转 1 圈的脉冲数 × 电机侧齿轮比) / (工件侧 1 圈的移动量 × 工件侧齿轮比)	
可管理的位置		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 <sup>*5</sup> )	
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 <sup>2</sup> )	
跃度指令		双精度实数型 (LREAL 型) 的正数、0 (指令单位 /s <sup>3</sup> )	
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	指定自指令当前位置起的移动距离, 进行定位的功能
		中断标准定位	指定自外部输入的中断输入发生位置起的移动距离, 进行定位的功能
		周期同步绝对定位控制 <sup>*2</sup>	位置控制模式下, 每个控制周期输出指令位置的功能
	单轴速度控制	速度控制	位置控制模式下进行速度控制的功能
		周期同步速度控制	速度控制模式下, 每个控制周期输出速度指令的功能
	单轴转矩控制	转矩控制	对电机进行转矩控制的功能
	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	结束通过输入参数指定的轴的凸轮动作的功能
		齿轮动作开始	设定主轴和从轴间的齿轮比, 进行齿轮动作的功能
		位置指定齿轮动作	设定主轴和从轴间齿轮比的同步位置, 进行齿轮动作的功能
		齿轮动作解除	中止正在执行的齿轮动作、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	执行同步控制中主轴的相位补偿的功能
		加减运算定位	将 2 个轴的指令位置相加、或相减的值作为指令位置输出的功能
	单轴手动操作	可运行	将伺服驱动器的状态切换为伺服 ON 状态, 使轴可执行动作的功能
		微动移动	根据指定目标速度执行微动移动的功能
	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 <sup>*2</sup>	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”设为目标位置进行定位, 返回原点的功能

功能项目		内容	
单轴	单轴控制辅助	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
		超调值设定	变更轴的目标速度的功能
		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值的功能
		启用外部锁定	根据触发的发生记录轴的位置的功能
		不启用外部锁定	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在指定范围(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	监控指定轴的指令位置或反馈位置的差异量是否超过了容许值的 功能
		偏差计数器复位	使指令当前位置和反馈当前位置的偏差归零的功能
		转矩限制	通过伺服驱动器的转矩限制功能的有效 / 无效切换和转矩限制值的 设定, 限制输出转矩的功能
		轴使用变更 *2	暂时切换轴参数的 [ 轴使用 ] 的功能
		启动速度 *2	设定开始轴动作时的初速度的功能
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	进行两轴的圆弧插补的功能
		轴组周期同步绝对位置控制 *2	位置控制模式下, 每个控制周期输出指令位置的功能
轴组 (续)	多轴协调控制 辅助	轴组错误复位	解除轴组和轴异常的功能
		启用轴组	启用轴组的功能
		不启用轴组	不启用轴组的功能
		写入轴组构成轴 *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
		轴组立即停止	使插补动作中的所有轴立即停止的功能
		轴组超调值设定	变更插补动作中的合成目标速度的功能
轴组位置获取 *2	获取轴组的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	更新通过输入参数指定的凸轮表的终点索引的功能
		保存凸轮表	将通过输入参数指定的凸轮表保存至 CPU 单元内置非易失性存 储器的功能
		生成凸轮表 *2	通过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	写入 MC 设定	暂时改写部分轴或轴组参数的功能
		轴参数的变更 *2	通过用户程序查看、变更轴参数的功能
辅助功能	计数模式		可选择线性模式(有限长度)或旋转模式(无限长度)
	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	通过跃度设定轴及轴组动作时的加减速曲线的功能(梯形曲 线、S 形曲线)
		变更加减速度	加减速动作中也可变更加减速度的功能
	到位检查		设定到位宽度和到位检查时间以检查定位是否完成的功能
	停止方法选择		设定立即停止输入信号或极限输入信号有效时的停止方法的功 能
	重启运动控制指令		通过变更正在执行的运动控制指令的输入变量后重启, 在动作 中变更目标值的功能
运动控制指令的多重启动 (缓存模式)		指定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 速度连接方法的功能	

功能项目		内容	
辅助功能	监控功能	软件限位	监控轴的动作范围的功能
		位置偏差	监控各轴的指令当前值与反馈当前值之间的位置偏差的功能
		速度 / 加减速速度 / 转矩 / 插补速度 / 插补加减速速度	按各轴或轴组设定警告值并进行监控的功能
	支持绝对值编码器	通过使用欧姆龙制 1S 系列或 G5 系列带绝对值编码器的电机, 接通电源时无需进行原点复位的功能 <sup>*7</sup>	
	输入信号的逻辑反转 <sup>*2</sup>	将立即停止输入信号、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进行反转的功能	
外部 I/F 信号		伺服驱动器可使用以下输入信号: 原点信号、近原点信号、正方向极限信号、 负方向极限信号、立即停止信号、中断输入信号	

\*1. 推荐的伺服驱动器 G5 系列的版本在电机为气缸型时为 Ver.2.1 以上, 在电机为直线型时为 Ver.1.1 以上。

\*2. CPU 单元版本升级后追加的功能。版本升级的相关信息请参阅 □□ “A-7 版本相关信息 (P.A-32)”。

\*3. 推荐单元版本为 Ver.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存在差别。详情请参阅 □□ “NX 系列 位置接口单元 用户手册 (SBCE-374)”。

\*5. 可在带符号整数型 40 位范围内设定转换为脉冲单位的数值。

\*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进行创建。指定主轴相和从轴位移。可按区间分别变更相位间隔宽度。可在用户程序内改写凸轮数据。

\*7. 欧姆龙制伺服驱动器 G5 系列 EtherCAT 通信内置直线电机型可使用绝对值型外部标尺。

# 2

## 运动控制系统的构成与原理

本章对 CPU 单元内部构成的概要和运动控制功能模块的构成、原理进行说明。

---

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 .....	2-5
2-3-1 CPU 单元的任务 .....	2-5
2-3-2 运动控制的任务动作示例 .....	2-11
2-4 与 EtherCAT 通信之间的关系 .....	2-18
2-4-1 CAN application protocol over EtherCAT (CoE) .....	2-18
2-4-2 EtherCAT 主站功能与运动控制功能模块之间的关系 .....	2-19
2-4-3 过程通信周期与运动控制周期之间的关系 .....	2-21

## 2-1 CPU 单元内部构成

下面，对 NJ/NX 系列 CPU 单元内部结构的概要进行说明。

CPU 单元的软件结构如下所示。

运动控制功能模块是指执行运动控制的软件。



\*1. 其它功能模块请参阅 □□ “NJ/NX 系列 CPU 单元 用户手册 软件篇 (SBCA-359)”。

OS 中含有 PLC 功能模块，在该 OS 中运行各功能模块。

各功能模块的内容如下所示。

功能模块名	简称	内容
PLC 功能模块	PLC	管理整体日程，执行用户程序，向运动控制功能模块发送指示，连接 USB、SD 存储卡。
运动控制功能模块	MC	根据用户程序的运动控制指令执行运动控制，向 EtherCAT 主站功能模块发送数据。
EtherCAT 主站功能模块	ECAT	作为 EtherCAT 主站，与 EtherCAT 从站进行通信。

(注) 其它功能模块请参阅 □□ “NJ/NX 系列 CPU 单元 用户手册 软件篇 (SBCA-359)”。

本手册对运动控制功能模块 (以下有时简称 “MC 功能模块”) 的规格和使用进行说明。

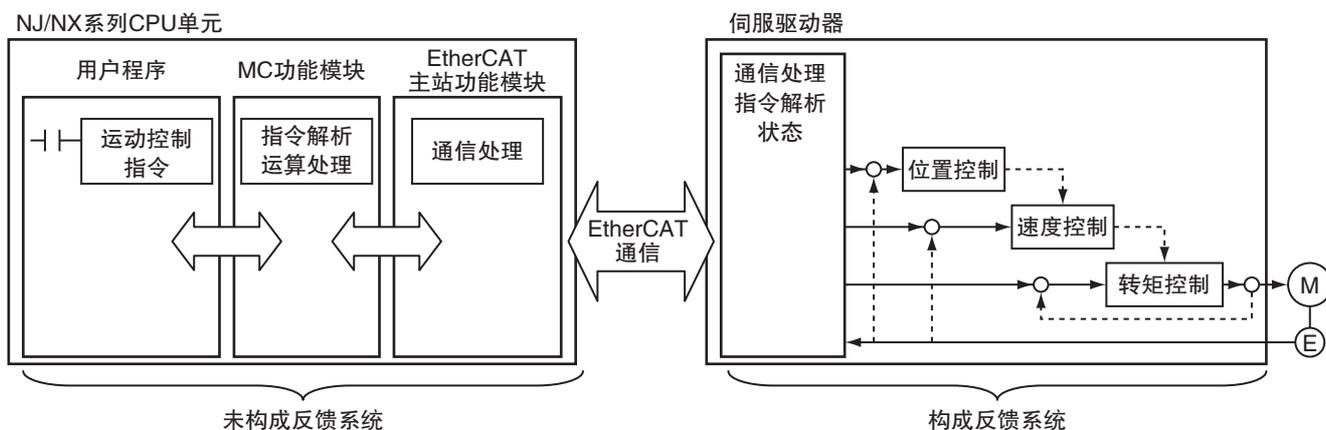
使用 MC 功能模块时，请同时参阅 NJ/NX 系列的相关用户手册。

## 2-2 运动控制系统的构成

通过伺服构建的控制系统一般采用半闭环方式控制电机动作。采用半闭环方式时，通过电机上安装的编码器检测指令值相应的电机旋转量。将其反馈为机械的移动量。计算指令值与电机实际旋转量之间的偏差并使其变为“0”的控制方法。

在使用了 MC 功能模块的设备构成中，不对 CPU 单元用户程序发送的指令使用反馈信息。在伺服驱动器内构建反馈系统。

### ● EtherCAT 网络构成时



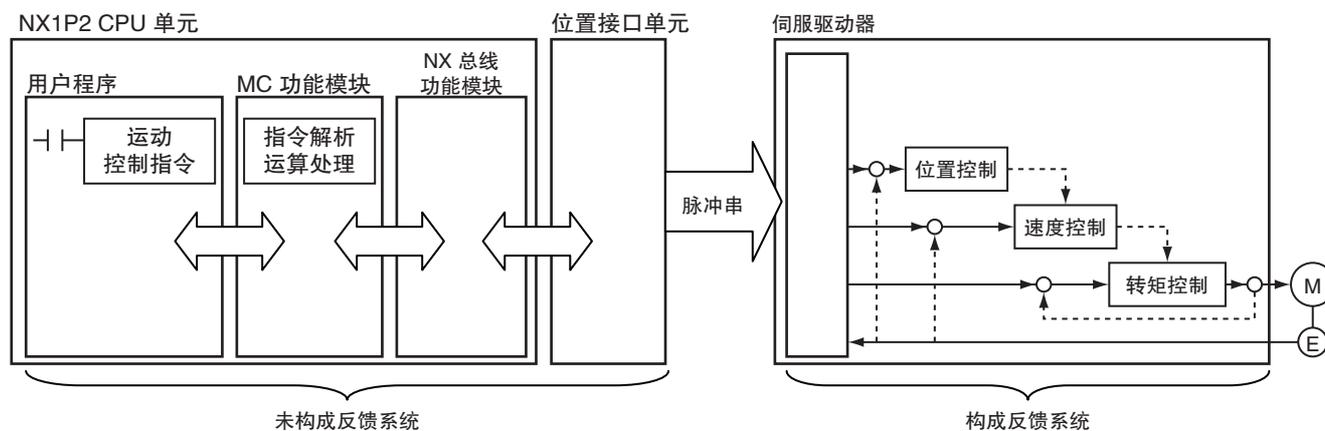
- 通过用户程序启动运动控制指令时，由 MC 功能模块解析指令。
- MC 功能模块根据指令的解析结果以固定周期执行运动运算，向伺服驱动器生成指令值。生成的指令值为目标位置、目标速度、目标转矩。
- 通过 PDO 通信，在每个 EtherCAT 通信的过程数据通信周期内发送生成的指令值。
- 伺服驱动器根据各 EtherCAT 通信的过程数据通信周期的指令值，执行位置循环、速度循环、转矩循环控制。
- 在每个 EtherCAT 通信的过程数据通信周期内向 CPU 单元发送编码器的当前值和伺服驱动器的状态。



### 参考

- 运动运算和 EtherCAT 通信的过程数据通信的周期相同。
- MC 功能模块以内置位置控制循环、速度控制循环、转矩控制循环的伺服驱动器为对象。
- 使用 NX 系列位置接口单元时的构成，请参阅 □□ “NX 系列位置接口单元用户手册 (SBCE-374)”。

### ● CPU 单元上的构成时



- 通过用户程序启动运动控制指令时，由 MC 功能模块解析指令。
- MC 功能模块根据指令的分析结果按一定周期执行运动运算，生成针对位置接口单元（脉冲输出单元）的指令值。生成的指令值为目标位置、目标速度。
- 将生成的指令值作为脉冲串输出至伺服驱动器。
- 伺服驱动器根据脉冲串的指令值执行位置环控制、速度环控制。



#### 参考

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

## 2-3 运动控制系统的原理

下面，对 CPU 单元的任务与运动控制之间的关系进行说明。

### 2-3-1 CPU 单元的任务

任务是指处理用户程序等时的执行条件和顺序的属性。

NJ/NX 系列 CPU 单元中含有以下任务。

任务的种类	任务名称
以固定周期执行的任务	原始恒定周期任务
	固定周期任务 (执行优先级 5 <sup>*1</sup> 、16 <sup>*2</sup> 、17、18)
仅在事先指定的执行条件成立时执行 1 次的任务。	事件任务 (执行优先级 8、48) <sup>*3</sup>

\*1. 固定周期任务 (执行优先级 5) 仅 NX701 CPU 单元可使用。

\*2. 固定周期任务 (执行优先级 16) 不能在 NX1P2 CPU 单元中使用。

\*3. Ver.1.03 以上的 CPU 单元和 Ver.1.04 以上的 Sysmac Studio 的组合中可使用事件任务。

用户程序和任务的详情以及设定方法请参阅 □□ “NJ/NX 系列 CPU 单元 用户手册 软件篇 (SBCE-359)”。

### 任务的种类与优先级

NX701 CPU 单元通过搭载多芯处理器，可并行执行原始恒定周期任务和固定周期任务 (执行优先级 5)。

NX1P2 CPU 单元及 NJ 系列 CPU 单元无法同时执行多个任务。

任务具有执行优先级，优先执行优先级较高的任务。执行某一任务时，如果其他执行优先级较高的任务的执行条件成立，则优先执行优先级较高的任务。

下表是 NJ/NX 系列 CPU 单元中可使用运动控制指令的任务种类和优先级。事件任务无法使用运动控制指令。

任务的种类	任务数	优先级	动作
原始恒定周期任务	1 个	4	在每个设定的任务周期内执行 I/O 刷新、用户程序、运动控制。任务的执行优先级最高，可高速、高精度执行任务，因此适用于需同步控制或高速响应的控制。以单任务执行所有控制时，使用原始恒定周期任务。
固定周期任务	0 ~ 1 个	5 <sup>*1</sup>	在每个设定的任务周期内执行 I/O 刷新、用户程序、运动控制。任务的执行优先级仅次于原始恒定周期任务，可高速、高精度执行任务。以原始恒定周期任务控制构成装置的功能中以最高速控制的功能，并将其他功能分割为固定周期任务 (执行优先级 5) 进行控制时使用。以多任务执行运动控制时，使用原始恒定周期任务和固定周期任务 (执行优先级 5)。*2
固定周期任务	0 ~ 1 个	16 <sup>*3</sup>	在每个设定的任务周期内执行 I/O 刷新和用户程序。在慢于原始恒定周期任务和固定周期任务 (执行优先级 5) 的周期内执行用户程序时，使用固定周期任务 (执行优先级 16)。固定周期任务 (执行优先级 16) 无法对以原始恒定周期任务执行 I/O 刷新的部分从站和单元描述用户程序。例如，以原始恒定周期任务和固定周期任务 (执行优先级 5) 执行同步控制和需要高速响应的控制，而整个装置的控制则分割为固定周期任务 (执行优先级 16) 进行控制。

\*1. 固定周期任务 (执行优先级 5) 仅 NX701 CPU 单元可使用。

\*2. 使用原始恒定周期任务和固定周期任务 (执行优先级 5) 两种任务并行执行运动控制被称为“多任务运动”。

\*3. 固定周期任务 (执行优先级 16) 不能在 NX1P2 CPU 单元中使用。

(注) CPU 单元中含有执行优先级为 17 和 18 的恒定周期任务。但在此类任务中无法使用运动控制指令。也无法执行 I/O 刷新。



### 使用注意事项

- 运动控制指令可在原始恒定周期任务和执行优先级为 5 和 16 的固定周期任务中使用。
- 如果在上述以外的任务中使用运动控制指令，则在 Sysmac Studio 中编连时会发生异常。



### 参考

NX701 CPU 单元不能以原始恒定周期任务和固定周期任务 (执行优先级 5) 执行运动控制。需要区分 2 种运动控制时，将原始恒定周期任务的运动控制称为“运动控制 1”，将固定周期任务 (执行优先级 5) 的运动控制称为“运动控制 2”。

## 任务的分配

- 轴及轴组可分配给原始恒定周期任务或固定周期任务 (执行优先级 5) 其中之一。分配给轴的 I/O 设备的任务必须与被分配轴的任务相同。

可分配



不可分配



- 对于分配给原始恒定周期任务的轴及轴组，可通过在固定周期任务 (执行优先级 16) 中运行的用户程序执行运动控制指令。



### 使用注意事项

- 对于分配给原始恒定周期任务的轴及轴组，无法通过在固定周期任务 (执行优先级 5) 中运行的用户程序执行运动控制指令。执行后，将发生“执行 ID 设定范围外 (事件代码: 57490000 Hex)”。
- 同样，对于分配给固定周期任务 (执行优先级 5) 的轴及轴组，无法通过在原始恒定周期任务中运行的用户程序执行运动控制指令。
- 对于分配给固定周期任务 (执行优先级 5) 的轴及轴组，无法通过在固定周期任务 (执行优先级 16) 中运行的用户程序执行运动控制指令。执行后，将发生“执行 ID 设定范围外 (事件代码: 57490000 Hex)”。



### 参考

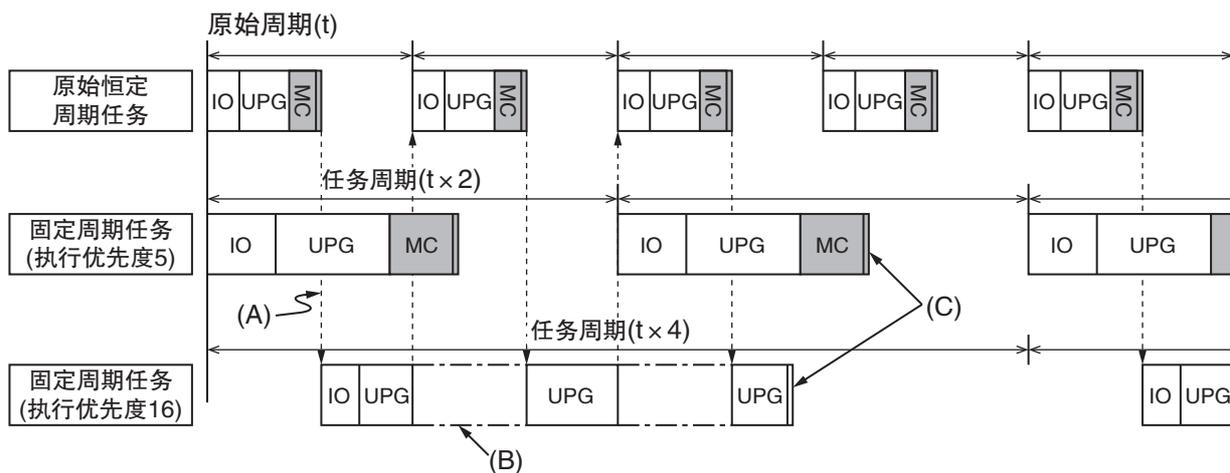
轴及轴组请参阅 “第 3 章 轴和轴组的设定”。

## 任务的基本动作

### ● 任务整体的动作

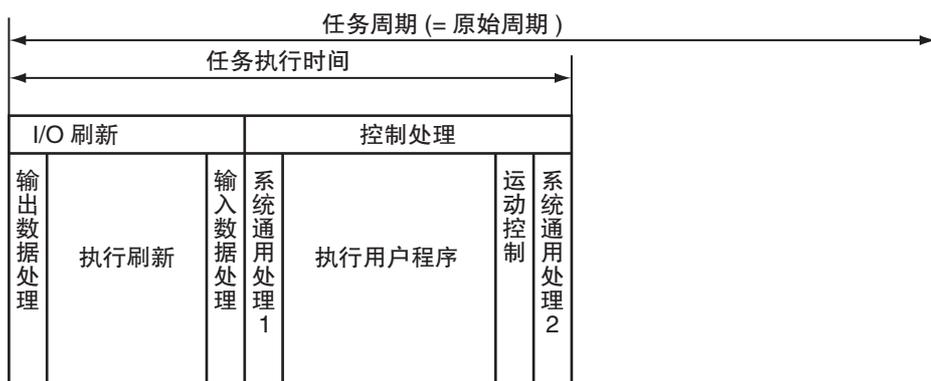
原始恒定周期任务和固定周期任务根据原始恒定周期任务的任务周期(以下称为“原始周期”)运行。原始恒定周期任务及固定周期任务(执行优先级5)中除了 I/O 刷新、执行用户程序外,还包括系统通用处理、运动控制。

执行该任务的 END 指令后,通过用户程序中描述的运动控制指令在下一运动控制(MC)的时间内执行运算。



记号	内容
IO	I/O 刷新
UPG	执行用户程序
MC	运动控制
(A)	虚线表示任务切换。
(B)	点划线表示任务处理中断状态。
(C)	双划线表示任务内处理执行完成。

● 原始恒定周期任务的动作



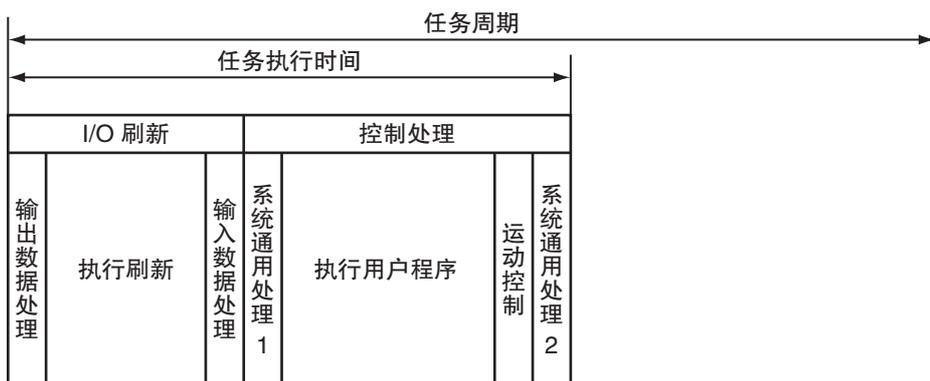
处理	处理内容
输出数据处理	<ul style="list-style-type: none"> <li>生成执行 I/O 刷新的 I/O(输出)的 OUT 刷新数据。</li> <li>设定了强制值刷新时, 会将强制值(输出)反映至 OUT 刷新数据。</li> </ul>
执行刷新	<ul style="list-style-type: none"> <li>执行与 I/O 之间的数据交换。</li> </ul>
输入数据处理	<ul style="list-style-type: none"> <li>执行事件任务的条件表达式一致判定。</li> <li>导入执行了 I/O 刷新的 I/O(输入)单元的 IN 刷新数据。</li> <li>设定了强制值刷新时, 会将强制值(输入)反映至导入的 IN 刷新数据。</li> </ul>
系统通用处理 1	<ul style="list-style-type: none"> <li>变量的任务间互斥控制处理(查看任务时)</li> <li>运动输入处理<sup>*1</sup></li> <li>数据跟踪处理(采样、触发判定)</li> </ul>
执行用户程序	<ul style="list-style-type: none"> <li>按分配顺序执行分配至任务的程序。</li> </ul>
运动控制 1 <sup>*2</sup>	<ul style="list-style-type: none"> <li>通过原始恒定周期任务及固定周期任务(执行优先级 16)的用户程序的运动控制指令执行运动控制指令。</li> <li>运动输出处理<sup>*3</sup></li> </ul>
系统通用处理 2	<ul style="list-style-type: none"> <li>变量的任务间互斥控制处理(更新任务时)</li> <li>用于确保与执行任务之间的同时性的控制器外部变量访问处理(通过[任务设定]设定的“变量访问时间”)</li> <li>处理 EtherNet/IP 标签数据链接时, 如果设定标签(网络公布属性的变量)相应的更新任务, 则作为以上变量访问处理执行。</li> </ul>

\*1. 根据从伺服驱动器等接收的数据, 更新运动控制系统变量的伺服驱动器状态和轴当前值等。

\*2. 分配给运动控制 1 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阅 □□ “3-1-3 轴变量概要 (P.3-6)”, 轴组的系统定义变量请参阅 □□ “3-3-3 轴组变量概要 (P.3-21)”。

\*3. 通过下一原始恒定周期任务的 “I/O 刷新” 向伺服驱动器等发送。

● 执行优先级 5 的固定周期任务动作

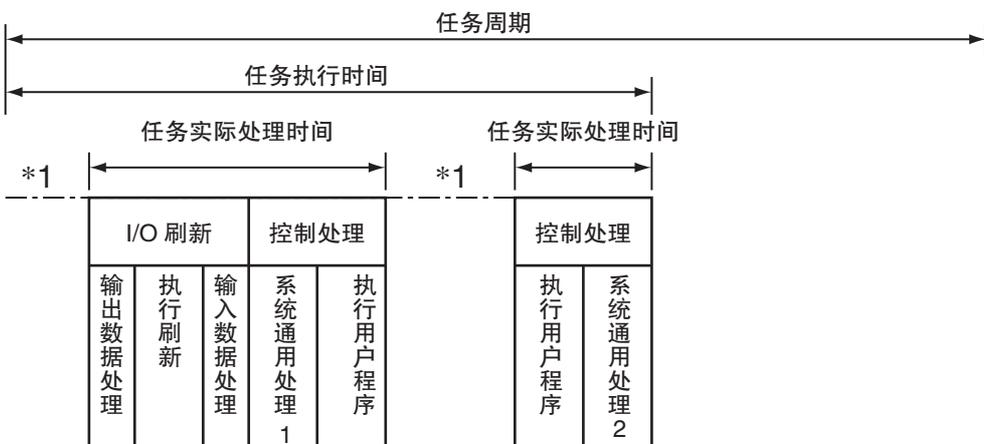


处理	处理内容
输出数据处理	<ul style="list-style-type: none"> <li>生成执行 I/O 刷新的 I/O(输出)的 OUT 刷新数据。</li> <li>设定了强制值刷新时,会将强制值(输出)反映至 OUT 刷新数据。</li> </ul>
执行刷新	执行与 I/O 之间的数据交换。
输入数据处理	<ul style="list-style-type: none"> <li>导入执行了 I/O 刷新的 I/O(输入)单元的 IN 刷新数据。</li> <li>设定了强制值刷新时,会将强制值(输入)反映至导入的 IN 刷新数据。</li> </ul>
系统通用处理 1	<ul style="list-style-type: none"> <li>变量的任务间互斥控制处理(查看任务时)</li> <li>运动输入处理 *1</li> <li>数据跟踪处理(采样、触发判定)</li> </ul>
执行用户程序	按分配顺序执行分配至任务的用户程序。
运动控制 2 *2	<ul style="list-style-type: none"> <li>通过固定周期任务(执行优先级 5)的用户程序的运动控制指令执行运动控制指令。</li> <li>运动输出处理 *3</li> </ul>
系统通用处理 2	<ul style="list-style-type: none"> <li>变量的任务间互斥控制处理(更新任务时)</li> <li>用于确保与执行任务之间的同时性的控制器外部变量访问处理(通过[任务设定]设定的“变量访问时间”)</li> </ul>

- \*1. 根据从伺服驱动器等接收的数据,更新运动控制系统变量的伺服驱动器状态和轴当前值等。
- \*2. 分配给运动控制 2 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阅 □ “3-1-3 轴变量概要(P.3-6)”,轴组的系统定义变量请参阅 □ “3-3-3 轴组变量概要(P.3-21)”。
- \*3. 通过下一固定周期任务(执行优先级 5)的“I/O 刷新”向伺服驱动器等发送。

● 执行优先级 16 的固定周期任务动作

执行优先级 16 的固定周期任务可执行 I/O 刷新。



- \*1 通过执行优先级较高的任务,暂时中断执行任务。

## 任务周期

单任务时，在原始恒定周期任务的任务周期即原始周期内进行动作。

单任务时，原始周期自动变为运动控制周期 (=EtherCAT 通信的过程数据通信周期)。

NX1P2 CPU 单元及 NJ 系列 CPU 单元仅限于单任务的控制。

多任务运动时，在原始周期和固定周期任务 (执行优先度 5) 的任务周期内进行动作。

多任务运动时，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各个任务周期。

在原始周期内同步执行固定周期任务。请将固定周期任务的任务周期设定为原始周期的整数倍。

例如，将原始周期设为 1ms、固定周期任务 (执行优先度 5) 的任务周期设为 2ms、固定周期任务 (执行优先度 16) 的任务周期设为 4ms 时，原始恒定周期任务和固定周期任务 (执行优先度 5) 的周期起点每 2 个原始周期排列 1 次。同样，固定周期任务 (执行优先度 16) 和原始恒定周期任务的周期起点每 4 个原始周期排列 1 次。

任务周期详情请参阅 □□ “NJ/NX 系列 CPU 单元 用户手册 软件篇 (SBCA-359)”。



### 参考

需要区分 2 种过程数据通信周期时，将原始恒定周期任务的通信周期称为 “过程数据通信周期 1”，将固定周期任务 (执行优先度 5) 的通信周期称为 “过程数据通信周期 2”。

### ● NX701 CPU 单元的可设定任务周期

根据任务种类，NX701 CPU 单元的可设定任务周期如下所示。

任务	可设定任务周期
原始恒定周期任务	125 μs, 250 μs ~ 8ms(可以 250 μs 为单位进行指定)
固定周期任务 (执行优先度 5)	125 μs, 250 μs ~ 100ms(可以 250 μs 为单位进行指定)
固定周期任务 (执行优先度 16)	1ms ~ 100ms(可以 250 μs 为单位进行指定)

(注) 原始恒定周期任务和固定周期任务的任务周期设定条件如下所示。

请将固定周期任务的任务周期设定为原始恒定周期任务的整数倍。

各任务的任务周期的最小公倍数请控制在 600ms 以下。

### ● NX1P2 CPU 单元的可设定任务周期

NX1P2 CPU 单元的任务种类仅限于原始恒定周期任务。可设定任务周期如下所示。

任务	可设定任务周期
原始恒定周期任务	2ms ~ 8ms(可以 250 μs 为单位进行设定)

### ● NJ 系列 CPU 单元的可设定任务周期

NJ 系列 CPU 单元的原始恒定周期任务与固定周期任务 (执行优先度 16) 的任务周期的可能组合如下表所示。

原始周期	可设定为固定周期任务的任务周期
500 μs <sup>*1</sup>	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 CPU 单元中可用于 Ver.1.03 以上的单元版本。

NJ101-10 □□ 不适用。

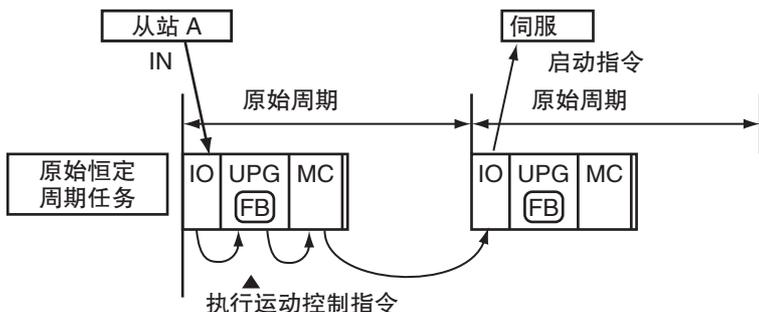
## 2-3-2 运动控制的任务动作示例

可在原始恒定周期任务或执行优先级 5 和 16 的固定周期任务中描述运动控制指令。

下面，对各种情况下的任务动作示例进行说明。

### 在原始恒定周期任务中使用运动控制指令时

需高速运动控制时，在原始恒定周期任务中描述运动控制指令 (FB)。



#### 1

##### 数据的导入

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

#### 2

##### 指令的启动

执行用户程序 (UPG) 时，根据导入的输入数据启动运动控制指令 (FB)。

此时，更新运动控制指令的输出变量。

#### 3

##### 指令的生成

根据启动的运动控制指令 (FB)，通过该原始恒定周期任务中之后的运动控制 (MC) 执行运动运算，向伺服驱动器生成启动指令。

#### 4

##### 指令的发送

通过下一周期的 I/O 刷新 (IO)，向伺服驱动器发送生成的启动指令。



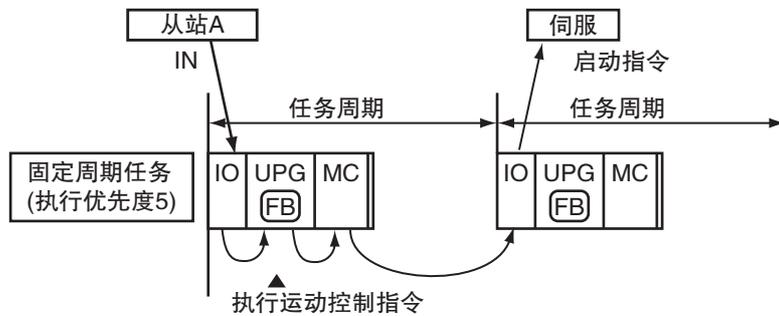
#### 参考

高速处理从输入到向伺服驱动器输出启动指令的整个过程，因此建议在原始恒定周期任务中描述运动控制指令。

## 在执行优先级 5 的固定周期任务中使用运动控制指令时

要求仅次于原始恒定周期任务的高速运动控制时，在固定周期任务（执行优先级 5）中描述运动控制指令（FB）。基本动作与原始恒定周期任务相同。

### ● 处理时间



#### 1 数据的导入

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

#### 2 指令的启动

执行用户程序 (UPG) 时，根据导入的输入数据启动运动控制指令 (FB)。此时，更新运动控制指令的输出变量。

#### 3 指令的生成

根据启动的运动控制指令 (FB)，通过该原始恒定周期任务中之后的运动控制 (MC) 执行运动运算，向伺服驱动器等生成启动指令。

#### 4 指令的发送

通过下一周期的 I/O 刷新 (IO)，向伺服驱动器发送生成的启动指令。



#### 参考

- 固定周期任务 (执行优先级 5) 仅 NX701 CPU 单元可使用。
- 将构成装置的功能中希望以最高速控制的功能分割为原始恒定周期任务，并将其他功能分割为固定周期任务 (执行优先级 5) 进行并行控制时，使用固定周期任务 (执行优先级 5)。

### ● 多任务运动中的轴变量的反映时间

“多任务运动”是指使用原始恒定周期任务和固定周期任务(执行优先度5)两种任务进行并行控制。在多任务运动中,以原始恒定周期任务控制的轴的轴变量值可通过固定周期任务(执行优先度5)的用户程序查看。与之相反,以固定周期任务(执行优先度5)控制的轴的轴变量值可通过原始恒定周期任务的用户程序查看。



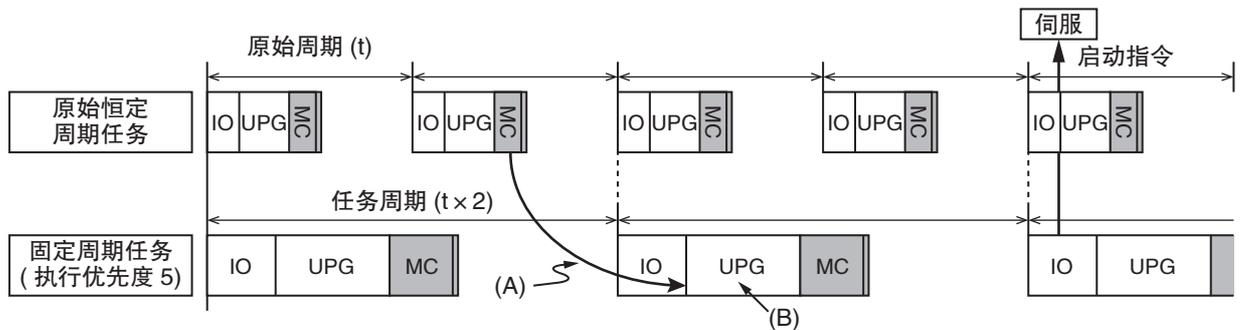
#### 参考

- 关于轴变量的同步请参阅 □ “3-1-4 轴变量的同步 (P.3-8)”。
- 以原始恒定周期任务控制的轴的轴变量值可通过固定周期任务(执行优先度16)的用户程序查看。详情请参阅 □ “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.2-15)” 及 □ “3-1-4 轴变量的同步 (P.3-8)”。

轴变量值与查看侧的任务周期同步执行。

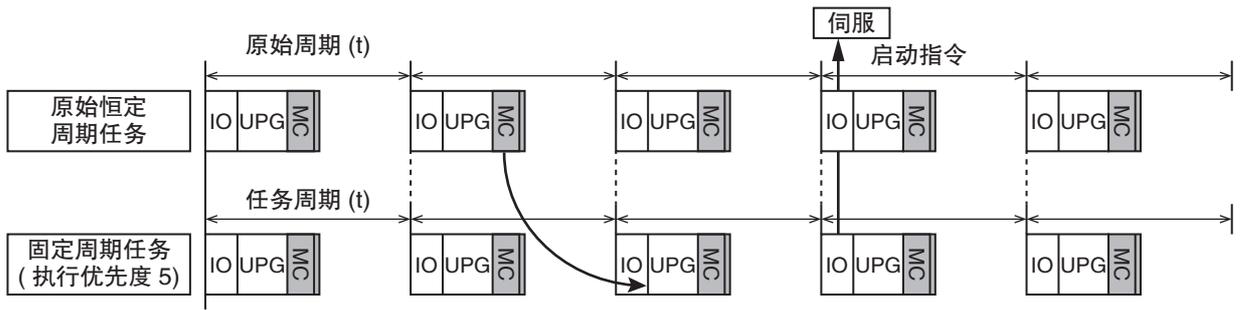
在查看轴变量值侧的任务的用户程序的执行过程中,所查看的轴变量值不会改写。

将原始恒定周期任务的轴变量值反映至固定周期任务(执行优先度5)的时间如下所示。



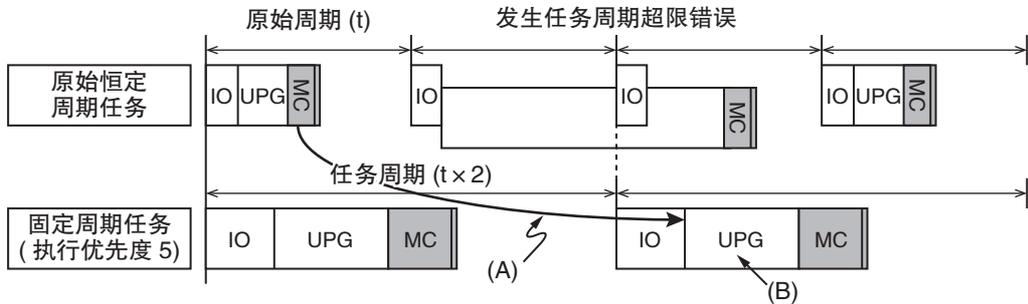
记号	说明
(A)	轴变量的更新。 不论固定周期任务(执行优先度5)的用户程序执行开始位置如何,都会反映任务周期起点重合的周期即将开始之前的原始恒定周期任务的执行结果。
(B)	执行用户程序的过程中,轴变量值不会改写。

任务周期相同时也一样。



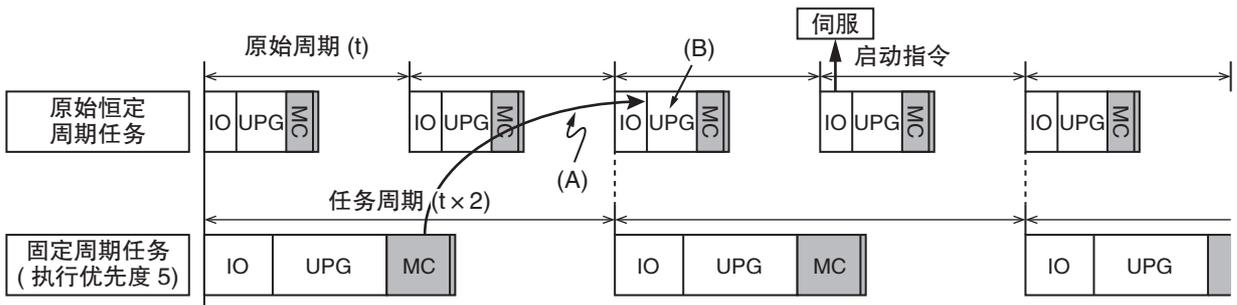
### 参考

发生任务周期超限时，不是反映任务周期起点重合的周期即将开始之前的任务周期的执行结果，而是反映再前一个周期的执行结果。



记号	说明
(A)	轴变量的更新。 发生任务周期超限时，反映再前一个周期的值。
(B)	执行用户程序的过程中，轴变量值不会改写。

将固定周期任务 (执行优先级 5) 的轴变量值反映至原始恒定周期任务的时间如下所示。



记号	说明
(A)	轴变量的更新。 反映任务周期起点重合的周期即将开始之前的固定周期任务 (执行优先级 5) 的执行结果。
(B)	执行用户程序的过程中，轴变量值不会改写。

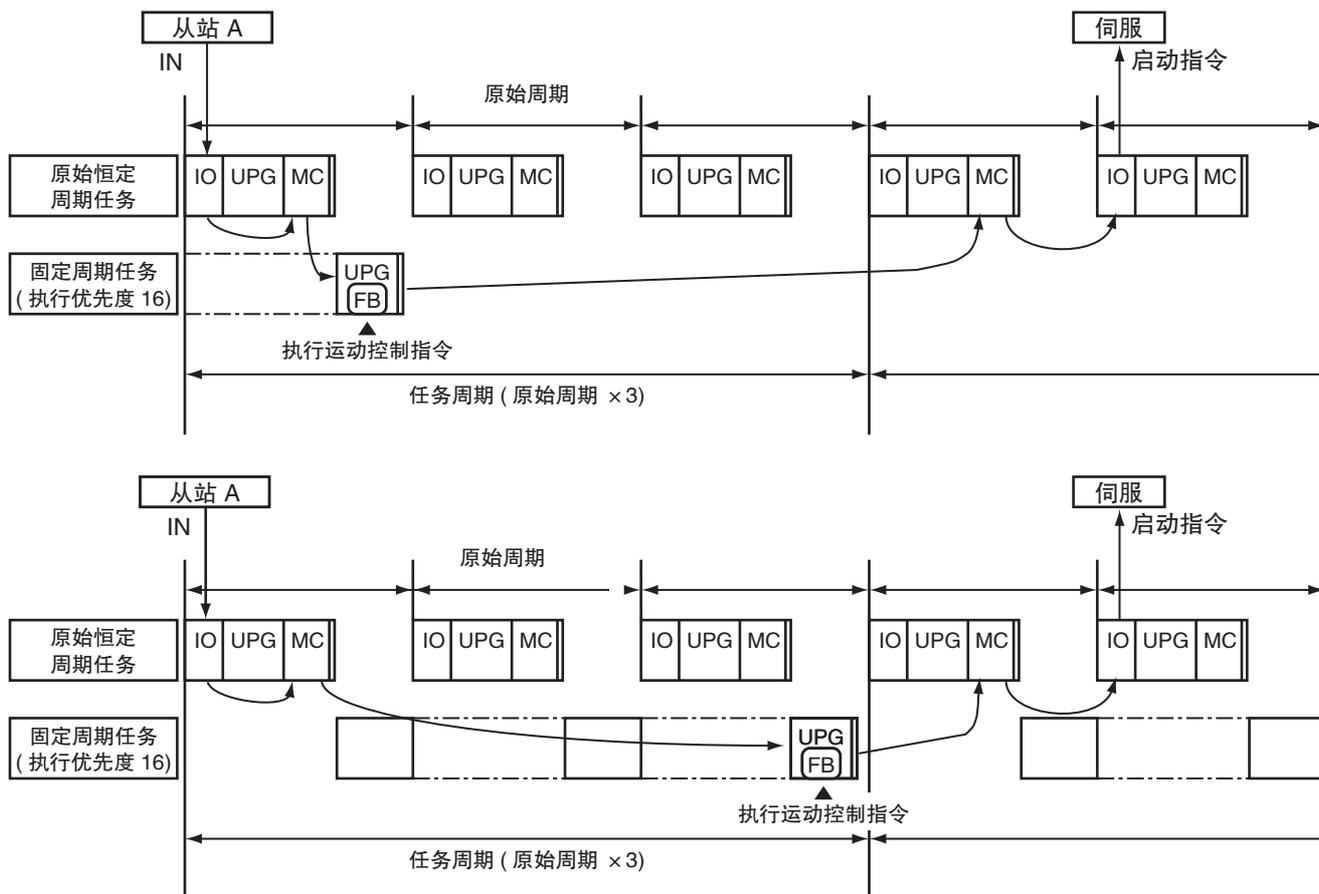
## 在执行优先级 16 的固定周期任务中使用运动控制指令时

无需高速运动控制且用户程序较大等时，在固定周期任务（执行优先级 16）中描述运动控制指令。

### ● 处理时间

在与固定周期任务（执行优先级 16）相同的任务周期内执行的运动控制指令（FB）在相同时间内执行运动控制（MC）。

因此，可在多轴中同时启动或中断。



### 1 数据的导入

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

### 2 指令的启动

执行固定周期任务 (执行优先级 16) 的用户程序 (UPG) 时，根据导入的输入数据启动运动控制指令 (FB)。

此时，更新运动控制指令的输出变量。

### 3 指令的生成

根据启动的运动控制指令 (FB)，在固定周期任务 (执行优先级 16) 的任务周期内执行运动控制指令 (FB)。

通过该固定周期任务的下一原始恒定周期任务中的运动控制 (MC) 执行运动运算，向伺服驱动器等生成指令。

### 4 指令的发送

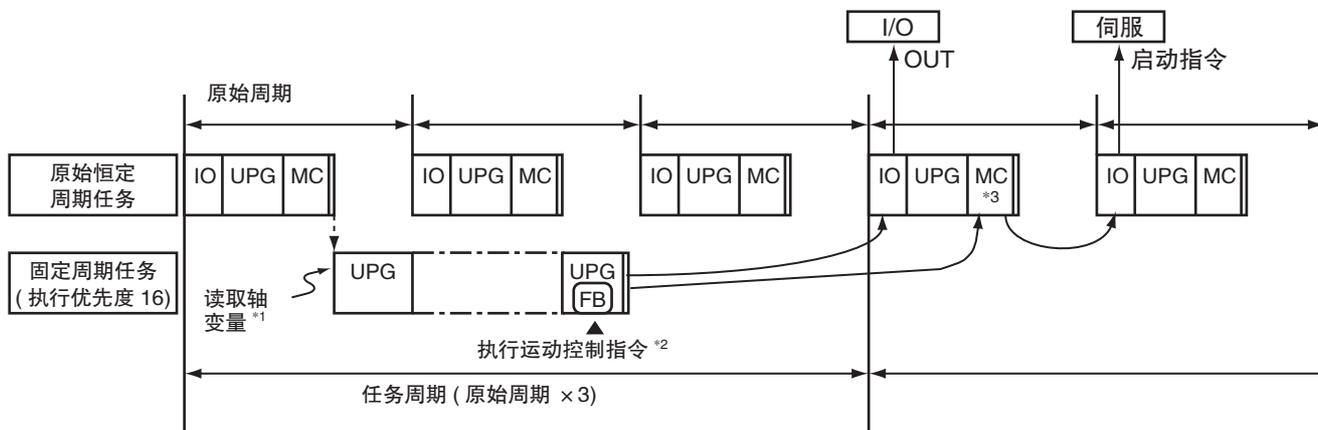
通过下一周期的 I/O 刷新 (IO)，向伺服驱动器等发送生成的启动指令。

### ● 轴变量的反映时间

轴变量是由部分 MC 功能模块的轴参数、当前位置及异常信息等监控信息组成的系统定义变量。

通过固定周期任务 (执行优先级 16) 查看原始固定周期任务的轴变量时, 读取固定周期任务 (执行优先级 16) 开始执行时的值。

执行运动控制指令 (FB) 时, 不改写轴变量的值。通过下一固定周期任务 (执行优先级 16) 中首次处理的运动控制 (MC) 进行改写。



\*1 读取固定周期任务 (执行优先级 16) 的用户程序执行开始时的原始恒定周期任务的轴变量值。

\*2 执行固定周期任务 (执行优先级 16) 的运动控制指令 (FB) 时, 不改写轴变量值。

\*3 通过该运动控制 (MC) 进行改写。



### 使用注意事项

- 在固定周期任务 (执行优先级 16) 中描述运动控制指令时, 如果固定周期任务 (执行优先级 16) 的任务周期变长, 则伺服驱动器等的响应时间也会变长。
- 从外部执行紧急停止时, 请事先确认可否安全停止。
- 创建程序时请注意, 执行优先级 16 的固定周期任务中描述的运动控制指令的执行时间和 I/O 控制的执行时间有所不同。
- 固定周期任务 (执行优先级 16) 不能在 NX1P2 CPU 单元中使用。



### 参考

关于轴变量, 请参阅 “3-1-3 轴变量概要 (P.3-6)”。

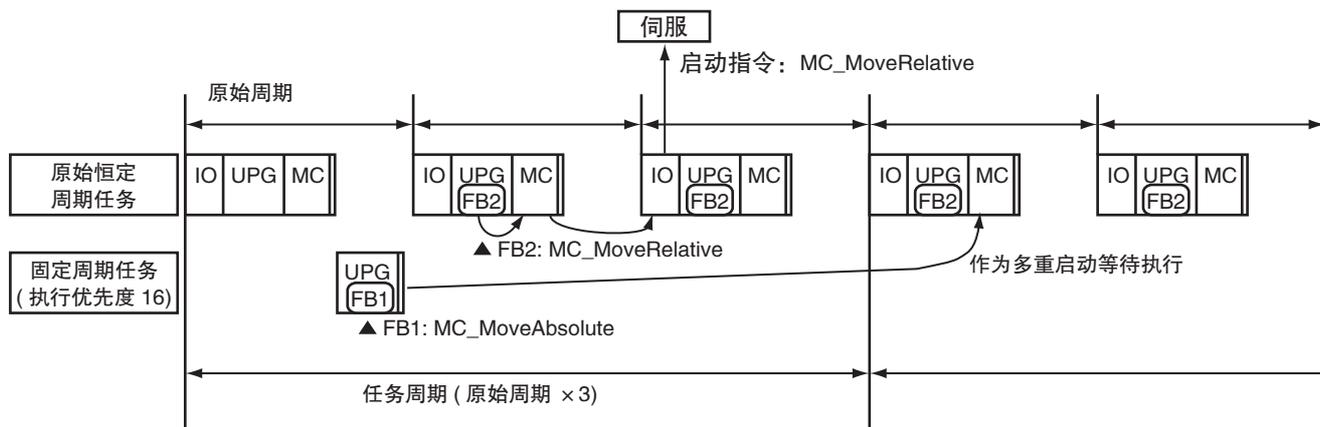
## 在 2 种任务中使用运动控制指令时

在相同轴中存在需或无需高速运动控制的处理时，在原始恒定周期任务和执行优先级 16 的固定周期任务中描述运动控制指令 (FB)。

在执行优先级 16 的周期内，启动上述 2 种任务的运动控制指令 (FB) 时，MC 功能模块先执行原始恒定周期任务的指令的运动运算。

例如，在执行优先级 16 的固定周期内启动 MC\_MoveAbsolute。然后，在原始恒定周期任务中对相同轴启动 MC\_MoveRelative 时，动作如下所示。

- MC 功能模块先执行 MC\_MoveRelative，作为指令的多重启动执行 MC\_MoveAbsolute。



在启动了该指令的任务的 I/O 刷新时间内，运动控制指令的输出变量值和运动控制系统变量值会发生变化。因此，在多个任务中对相同轴使用运动控制指令时，状态可能因任务而异。请充分了解各任务的处理后创建程序。



### 使用注意事项

- 在原始恒定周期任务和执行优先级 16 的固定周期任务中描述相同轴的运动控制指令时，创建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。
- 如果在多个任务中使用相同轴的运动控制系统变量，则编程时请注意运动控制系统变量的更新时间的不同。
- 固定周期任务 (执行优先级 16) 不能在 NX1P2 CPU 单元中使用。



### 参考

关于多重启动，请参阅 □ “9-5-7 运动控制指令的多重启动 (缓存模式)(P.9-44)”。

## 2-4 与 EtherCAT 通信之间的关系

MC 功能模块通过 CPU 单元内置的 EtherCAT 主功能模块的 PDO 通信控制伺服驱动器、计数器及 NX 系列位置接口单元。

下面，对 EtherCAT 通信时 MC 功能模块的相关内容进行说明。

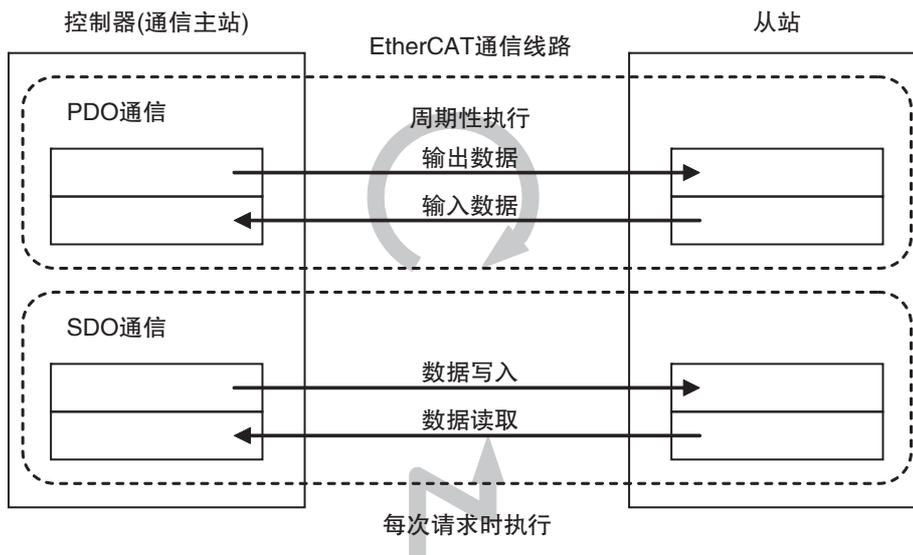
### 2-4-1 CAN application protocol over EtherCAT (CoE)

MC 功能模块使用 CAN application protocol over EtherCAT(CoE) 协议，与 EtherCAT 从站之间进行信息交换。在 CoE 中通过对象字典 (OD) 的数据规格规定各种从站带有的参数和控制信息。

在控制器/通信主站与从站之间传输此类数据的方法分为在固定周期内进行实时信息交换的过程数据对象 (Process Data Objects: PDO) 和在任意时间内传递信息的服务数据对象 (Service Data Objects: SDO) 的 2 种。

在 MC 功能模块中，对伺服电机的位置控制等以一定的控制周期进行输入输出数据更新的指令使用 PDO 通信。

与其不同，在参数传送等指定时间内读写数据的指令中使用 SDO 通信。



## 2-4-2 EtherCAT 主站功能与运动控制功能模块之间的关系

在 NJ/NX 系列 CPU 单元中，可连接 EtherCAT 从站执行时序控制和运动控制。

### 时序控制

- 在 Sysmac Studio 的 EtherCAT 编辑画面中创建 EtherCAT 构成时，自动生成构成从站的“I/O 端口”。
- 通过 Sysmac Studio 的 I/O 映射分配设备变量。
- 通过运动控制指令以外的指令执行时序控制。

### 运动控制

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

可分配给轴变量的设备为 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

- 运动控制指令以外的指令不能通过 PDO 通信直接向分配给轴变量的 EtherCAT 从站和 NX 系列位置接口单元发出指令。但可通过轴变量间接查看状态。但可通过轴变量间接查看状态。
- 使用 SDO 通信，也可读取分配给轴变量的 EtherCAT 从站和 NX 系列位置接口单元的对象。请勿通过 SDO 通信写入 PDO 通信中映射的对象。写入后的动作取决于从站的规格。如果是欧姆龙制从站，则会发生 SDO 通信异常。
- 未将 EtherCAT 从站的伺服驱动器和编码器输入终端及 NX 系列 位置接口单元分配给轴变量时，请按照与通用 EtherCAT 从站相同的方式执行时序控制。

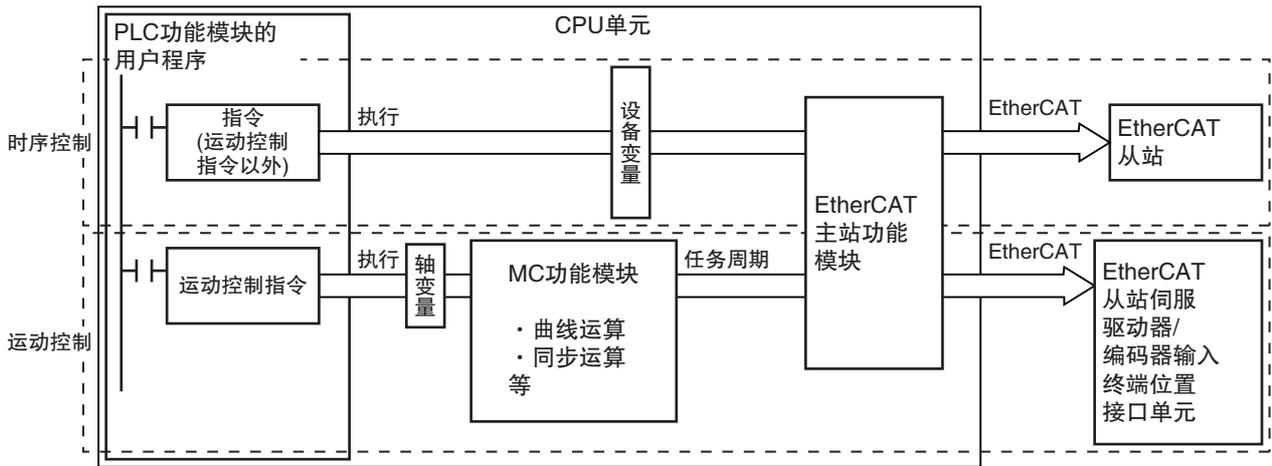


#### 版本相关信息

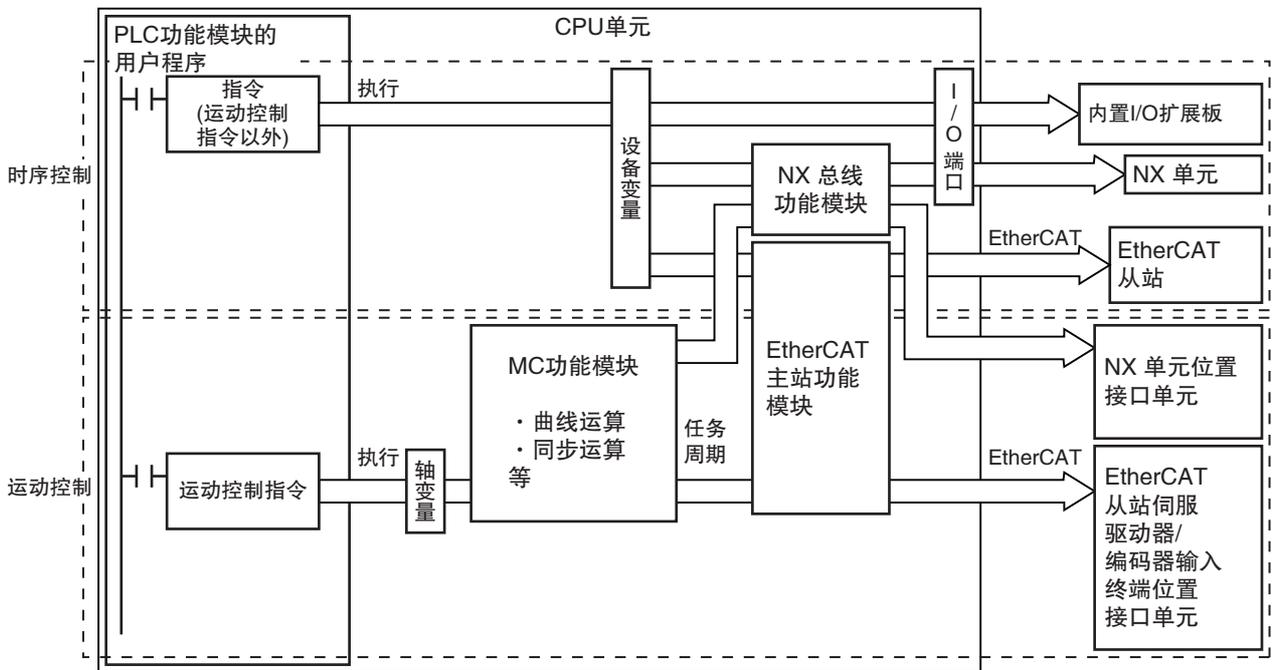
使用 Sysmac Studio Ver.1.09 以上版本，可对分配给轴变量的从站、单元的 I/O 端口分配设备变量。可分配的 I/O 端口的条件如下。

- 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), Sysmac Studio 的 [ 轴基本设定 ] 画面的 [ 详细设定 ] 中将过程数据设为 [ < 不分配 > ] 的端口 在 Ver.1.08 以下版本的 Sysmac Studio 中打开将设备变量分配给轴变量的项目，若使用 Ver.1.09 以上版本的 Sysmac Studio，则设备变量的分配将被解除。

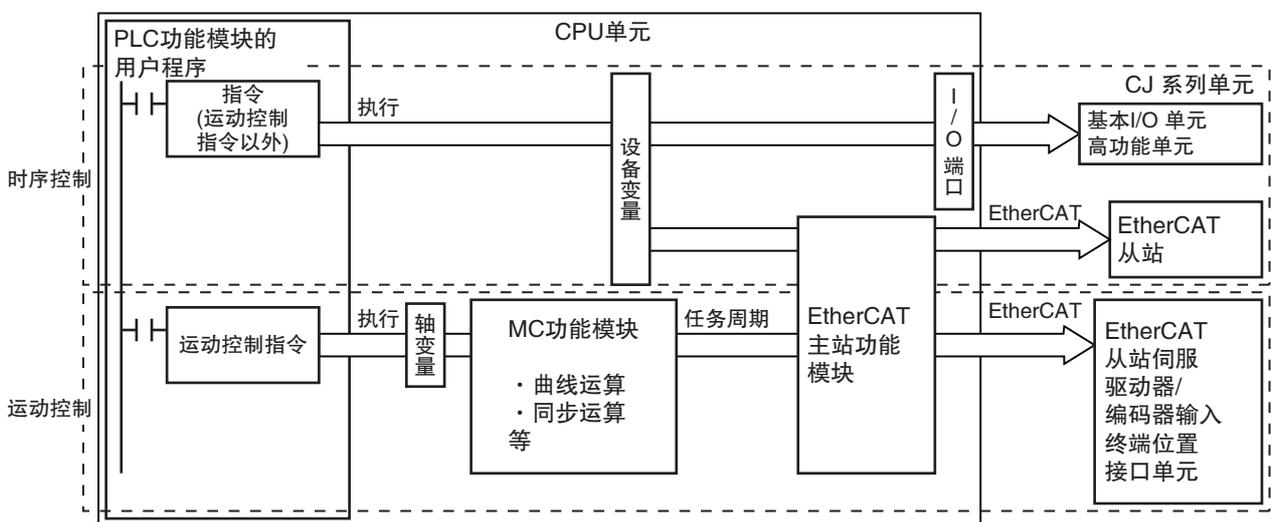
● NX701 CPU 单元



● NX1P2 CPU 单元



● NJ 系列 CPU 单元



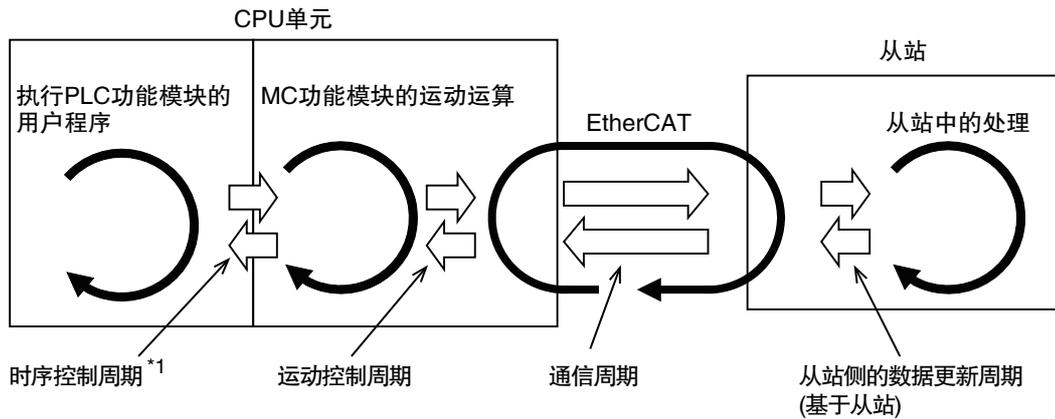
### 2-4-3 过程通信周期与运动控制周期之间的关系

PLC 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送运动控制的指令。通过以上操作，运动控制功能模块会执行运动运算，向 EtherCAT 的伺服驱动器发送运算结果（指令）。

此类数据交换按以下周期更新。

#### ● NX701 CPU 单元

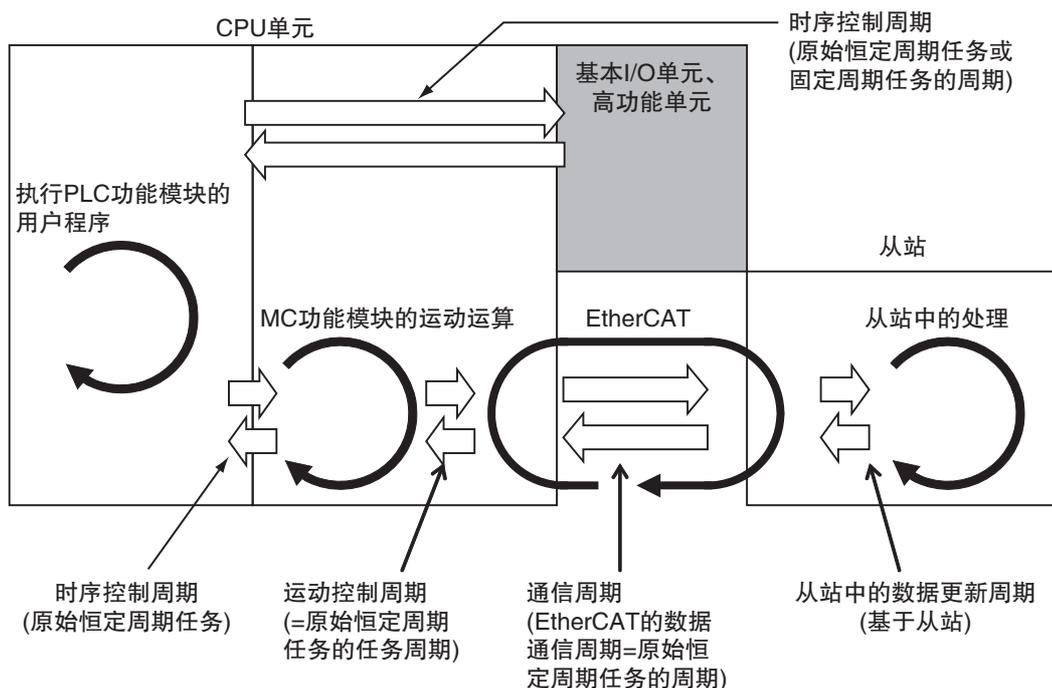
- “原始周期” = “运动控制 1 的周期” = “EtherCAT 通信的过程数据通信周期 1”
- “固定周期任务（执行优先级 5）的任务周期” = “运动控制 2 的周期” = “EtherCAT 通信的过程数据通信周期 2”



- \*1 时序控制周期为原始周期时，运动控制周期及通信周期变为原始周期。  
时序控制周期为固定周期任务（执行优先级 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（执行优先级 5）的任务周期。

#### ● NX1P2 CPU 单元及 NJ 系列 CPU 单元

- “原始周期” = “运动控制周期” = “EtherCAT 通信的过程数据通信周期”





# 3

## 轴和轴组的设定

本章对轴和轴组的概要、通过 Sysmac Studio 的 MC 试运行功能运行时所需的轴的设定项目及通过 Sysmac Studio 进行创建和设定的步骤进行说明。

---

<b>3-1 轴</b> .....	<b>3-2</b>
3-1-1 轴概要 .....	3-2
3-1-2 轴参数概要 .....	3-3
3-1-3 轴变量概要 .....	3-6
3-1-4 轴变量的同步 .....	3-8
3-1-5 用户程序中轴的指定方法 .....	3-8
<b>3-2 轴的设定步骤</b> .....	<b>3-9</b>
3-2-1 轴的设定流程 .....	3-9
3-2-2 设定步骤 .....	3-9
<b>3-3 轴组</b> .....	<b>3-19</b>
3-3-1 轴组概要 .....	3-19
3-3-2 轴组参数概要 .....	3-20
3-3-3 轴组变量概要 .....	3-21
3-3-4 用户程序中轴组的指定方法 .....	3-22
<b>3-4 轴组的设定步骤</b> .....	<b>3-23</b>
3-4-1 轴组的设定流程 .....	3-23
3-4-2 设定步骤 .....	3-23

## 3-1 轴

下面，对 MC 功能模块的轴进行说明。

### 3-1-1 轴概要

在运动控制系统中，将运动控制的对象称为“轴”。

轴中含有通过 EtherCAT 连接的实际伺服驱动器、编码器和 MC 功能模块内部的虚拟伺服驱动器、编码器。

此外，NX701 CPU 单元和 NJ 系列 CPU 单元的所有“轴”为运动控制轴，而 NX1P2 CPU 单元除运动控制轴以外，还增加了“单轴位置控制轴”。通过轴基本设定的[控制功能]切换到“单轴位置控制轴”。[控制功能]的详情请参阅 □□ “控制功能 (P.5-9)”。

MC 功能模块的轴中包括下表中的种类。

轴种类	内容
伺服轴	使用 EtherCAT 从站伺服驱动器*1 的轴。分配为实际伺服驱动器加以使用。 将 1 台伺服电机作为 1 根轴使用。 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给 1 轴。
虚拟伺服轴	MC 功能模块内的虚拟轴。不使用实际伺服驱动器。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和 NX 系列 位置接口单元*1 的轴。分配为实际编码器输入终端加以使用。 1 台编码器输入终端有 2 个计数器时，将各个计数器作为 1 根轴使用。
虚拟编码器轴	虚拟执行编码器动作的轴。没有编码器实物时，作为编码器轴的替代品临时使用。*2

\*1. 控制对象设备请参阅 □□ “1-4-3 功能规格 (P.1-10)”。

\*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。在支持该指令的版本的 MC 功能模块中，无法使用计数器功能。

MC 功能模块的轴中含有以下相关要素。

这些要素对应各型号的最大控制轴数。最大控制轴数因型号而异。详情请参阅 □□ “1-4-2 性能规格 (P.1-7)”。

构成要素	内容	参考页
轴参数	为 MC 功能模块控制的各轴设定最高速度、微动移动、原点复位等轴动作。 设定轴参数时，使用 Sysmac Studio。	P.3-3
轴变量	轴变量是由 MC 功能模块控制的各轴的当前位置和异常信息等监控信息组成的系统定义变量。 通过 Sysmac Studio 的多视图浏览器添加轴时，创建轴变量。可通过该视图设定轴变量的名称（以下称为轴变量名称）。	P.3-6
用户程序中轴的指定方法	在用户程序中，使用运动控制指令执行运动控制。将执行单轴控制的运动控制指令称为“轴指令”。 在指令的输入输出变量“Axis(轴)”中，通过系统定义变量的轴变量名称或 Sysmac Studio 设定的轴变量名称指定轴指令控制哪个轴。	P.3-8

## 3-1-2 轴参数概要

## ● 轴参数一览表

分类	参数名称
轴基本设定	轴号
	运动控制 *1
	轴使用
	轴种类
	控制功能 *2
	输入设备 / 输出设备
单位转换设定	显示单位
	电机转 1 圈的脉冲数
	电机转 1 圈的移动量
	减速机使用 *3
	工件侧转 1 圈的移动量 *3
	工件侧齿轮比 *3
	电机侧齿轮比 *3
动作设定	最高速度
	启动速度 *4
	微动最高速度
	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
	正方向转矩警告值 *5
	负方向转矩警告值 *5
	反馈速度滤波时间常数
	到位宽度
	到位检查时间
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
	驱动器错误复位监视时间
	正方向转矩限制上限值
	负方向转矩限制上限值
	立即停止输入逻辑反转 *4
	正方向极限输入逻辑反转 *4
	负方向极限输入逻辑反转 *4
	原点附近输入逻辑反转 *4
限制设定	软件限制功能
	正方向软件限制
	负方向软件限制
	位置偏差超过值
	位置偏差警告值
位置计数设定	计数模式
	环计数器上限设定值
	环计数器下限设定值
	编码器种类
伺服驱动器设定	环计数器上限设定值
	环计数器下限设定值
	PDS 状态控制方式

分类	参数名称
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
	原点输入屏蔽距离
	原点位置偏置
	原点复位接触时间
原点复位补偿值	
原点复位补偿速度	

\*1. NX701 CPU 单元时设定。

\*2. NX1P2 CPU 单元时设定。

\*3. Ver.1.11 以上的 CPU 单元和 Ver.1.15 以上的 Sysmac Studio 组合时可使用。

\*4. Ver.1.05 以上的 CPU 单元和 Ver.1.06 以上的 Sysmac Studio 组合时可使用。

\*5. 仅转矩控制时有效。

轴参数的详情请参阅 □ “5-2 轴参数 (P.5-4)”。

### ● 使用轴时须设定的项目

为通过 Sysmac Studio 创建的轴设定以下项目后可使用轴。

分类	参数名称	设定内容	参考
轴基本设定	轴号	按创建轴的顺序自动设定。	P.5-6
	运动控制 *1	选择“原始恒定周期任务”。	
	轴使用	选择 [ 轴使用 ]。	
	轴种类	选择需控制的轴的种类。	
	控制功能 *2	选择 [ 全部 ]。*3	
输入设备 / 输出设备	选择轴中分配的 EtherCAT 从站设备的节点地址。 轴种类为 [ 虚拟轴 ] 时无法选择。		

\*1. NX701 CPU 单元时设定。

\*2. NX1P2 CPU 单元时设定。

\*3. 用作运动控制轴时，选择“全部”。用作单轴位置控制轴时，请选择“仅单轴位置控制”。

### ● 通过 Sysmac Studio 试运行伺服驱动器时须设定的项目

通过 Sysmac Studio 的 MC 试运行功能运行 EtherCAT 连接的伺服驱动器时，设定以下项目。

分类	参数名称	设定内容	参考
轴基本设定	轴号	可按创建轴的顺序进行前移设定。	P.5-6
	运动控制 *1	选择“原始恒定周期任务”。	
	轴使用	选择 [ 轴使用 ]。	
	轴种类	选择 [ 伺服轴 ]。	
	控制功能 *2	选择 [ 全部 ]。*3	
输入设备 / 输出设备	选择轴中分配的 EtherCAT 从站设备的节点地址。 轴种类为 [ 虚拟轴 ] 时无法选择。		

分类	参数名称	设定内容	参考
单位转换设定	显示单位	选择 [mm]、[degree] 等需显示的单位。	P.5-11
	电机转 1 圈的脉冲数	根据编码器的分辨率, 设定电机转 1 圈的脉冲数。*4	
	电机转 1 圈的移动量	根据机械的规格, 设定电机转 1 圈的工件移动量。	
	减速机使用 *5	指定是否使用减速机设定。	
	工件侧转 1 圈的移动量 *5	设定工件侧转 1 圈时的移动量。	
	工件侧齿轮比 *5	设定工件侧的齿轮比。	
	电机侧齿轮比 *5	设定电机侧的齿轮比。	
位置计数设定	计数模式	根据机械的规格进行选择。	P.5-21
限制设定	软件限制功能	根据装置的规格进行选择。	P.5-21

\*1. NX701 CPU 单元时设定。

\*2. NX1P2 CPU 单元时设定。

\*3. 用作运动控制轴时, 选择“全部”。用作单轴位置控制轴时, 请选择“仅单轴位置控制”。

\*4. 例如, 编码器分辨率为 10000 脉冲/圈时, 设为 10000。

\*5. Ver.1.11 以上的 CPU 单元和 Ver.1.15 以上的 Sysmac Studio 组合时可使用。



#### 使用注意事项

- 实际运行电机时, 请根据机械的动作条件, 为最高速度、最大加减速度及停止设定等设定恰当的值。
- 欧姆龙制伺服驱动器 1S 系列及 G5 系列可通过驱动器正面的旋转开关对节点地址设定固定值。将该旋转开关设定为“00”时, 节点地址遵从 Sysmac Studio 的 EtherCAT 编辑的设定值。连接的所有驱动器将旋转开关设定为“00”时, 即使变更驱动器的连接位置也不会发生异常。为装置的各控制对象分配特定的伺服驱动器时, 请通过旋转开关设定驱动器的节点地址。

### 3-1-3 轴变量概要

轴变量是由部分 MC 功能模块的轴参数、当前位置及异常信息等监控信息组成的系统定义变量。通过 Sysmac Studio 创建轴时，按创建的顺序将轴变量登录至变量表。轴变量的数据类型为结构体型，数据名称为 “\_sAXIS\_REF”。

#### 轴变量名称

MC 功能模块的轴变量名称分为系统定义变量的轴变量名称和通过 Sysmac Studio 添加轴时的轴变量名的 2 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]”。

通过 Sysmac Studio 添加轴时，为 “\_MC\_AX[0] ~ \_MC\_AX[255]” 设定相应的默认名称，即 “MC\_Axis000 ~ MC\_Axis255”。编号按添加的顺序前移。可通过 Sysmac Studio 将各轴的轴变量名称变更为任意的变量名称。用户程序中可使用系统定义变量的轴变量名称或通过 Sysmac Studio 添加的轴变量名称。

#### ● 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对象 *1)	通过 Sysmac Studio 添加的 轴变量的默认名称	轴号示例
_MC_AX[0]	MC_Axis000	轴 0
_MC_AX[1]	MC_Axis001	轴 1
:	:	:
_MC_AX[255]	MC_Axis255	轴 255

\*1. 变更 Sysmac Studio 的全局变量视图中的分配对象栏名称时会发生异常。



#### 参考

- 对于 NX701 CPU 单元，可使用 “\_MC\_AX[0-255]” 或 “\_MC1\_AX[0-255]”，以及 “\_MC2\_AX[0-255]”。
- 对于 NX1P2 CPU 单元，只能使用 “\_MC\_AX[0-11]”。
- 对于 NJ 系列 CPU 单元，只能使用 “\_MC\_AX[0-63]”。
- 对于 NX701 CPU 单元，“\_MC\_AX[0-255]” 和 “\_MC1\_AX[0-255]” 的轴号相同时查看相同值。任意轴变量皆可使用，也可混合使用。
- 分配给原始恒定周期任务的轴变量为 “\_MC\_AX[0-255]” 或 “\_MC1\_AX[0-255]”。
- 分配给恒定周期任务 (执行优先级 5) 的轴变量为 “\_MC2\_AX[0-255]”。

## 轴变量的层次和轴变量名称的变更示例

下面以 `_MC_AX[0]` 为例进行说明。其他轴变量也一样。

<code>_MC_AX[0]</code>	: 轴变量
<code>_MC_AX[0].Status</code>	: 表示“轴状态”的层次
<code>_MC_AX[0].Status.Ready</code>	: 表示“轴启动准备完成”的变量
<code>_MC_AX[0].Status.Disabled</code>	: 表示“轴无效”的变量
:	
<code>_MC_AX[0].Details</code>	: 表示“轴控制状态”的层次
<code>_MC_AX[0].Details.Idle</code>	: 表示“停止中”的变量
<code>_MC_AX[0].Details.InPosWaiting</code>	: 表示“到位等待”的变量
:	
<code>_MC_AX[0].Cmd</code>	: 表示“轴指令值”的层次
<code>_MC_AX[0].Cmd.Pos</code>	: 表示“指令当前位置”的变量
<code>_MC_AX[0].Cmd.Vel</code>	: 表示“指令当前速度”的变量
<code>_MC_AX[0].Cmd.AccDec</code>	: 表示监控轴的“指令当前加减速度”的变量
:	
<code>_MC_AX[0].Act</code>	: 表示“轴当前值”的层次
<code>_MC_AX[0].Act.Pos</code>	: 表示“反馈当前位置”的变量
<code>_MC_AX[0].Act.Vel</code>	: 表示“反馈当前速度”的变量
:	
<code>_MC_AX[0].Cfg</code>	: 表示“轴基本设定”的层次
<code>_MC_AX[0].Cfg.AxNo</code>	: 表示“轴号”的变量
<code>_MC_AX[0].Cfg.AxEnable</code>	: 表示“轴使用”的变量
<code>_MC_AX[0].Cfg.AxType</code>	: 表示“轴种类”的变量
:	
<code>_MC_AX[0].Scale.Units</code>	: 表示“显示单位”的变量
<code>_MC_AX[1]</code>	: 轴变量
:	

### 例

将“`MC_Axis000`”变更为“`MyAxis1`”时，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。

轴变量的详情请参阅 □□ “轴变量 (P.6-20)”。

## 3-1-4 轴变量的同步

以原始恒定周期任务控制的轴的轴变量值可通过固定周期任务 (执行优先度 5) 的用户程序查看。与之相反, 以固定周期任务 (执行优先度 5) 控制的轴的轴变量值可通过原始恒定周期任务的用户程序查看。另外, 以原始恒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) 的用户程序查看。查看任务的种类及轴变量的关系如下所示。

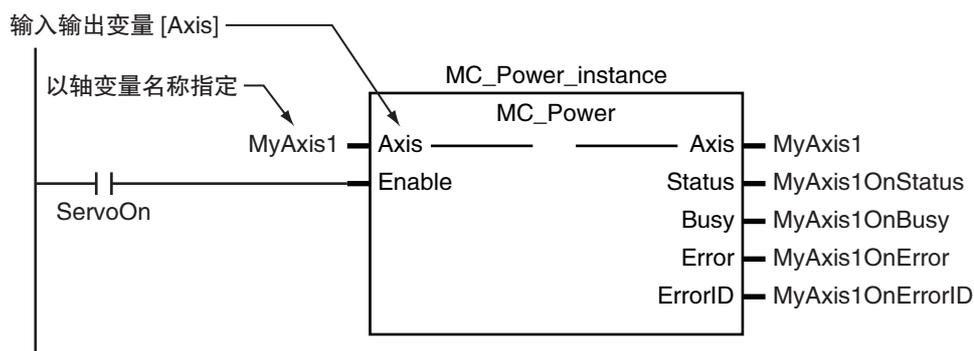
○: 可查看 x: 不可查看

任务的种类	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
原始恒定周期任务	○	○	○ *4	x
固定周期任务 (执行优先度 5)	○ *5	x	○	○
固定周期任务 (执行优先度 16)	○ *5	○ *6	○ *4	x
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. 通过 Sysmac Studio 创建“轴”时自动生成的用户定义的轴变量“MC\_Axis000”和“Loader1”等。建议使用用户定义的轴变量。
- \*2. “\_MC\_AX[\*]”和“\_MC1\_AX[\*]”。
- \*3. “\_MC2\_AX[\*]”。
- \*4. 在固定周期任务 (执行优先度 5) 的每个任务周期内处理单个轴的轴变量。
- \*5. 在每个原始周期内处理单个轴的轴变量。
- \*6. 在每个原始周期内处理最大轴数的轴变量。原始恒定周期任务的任务执行时间会增加, 因此不推荐。
- \*7. 在每个原始周期内处理最大轴数的轴变量。
- \*8. 在固定周期任务 (执行优先度 5) 的每个任务周期内处理最大轴数的轴变量。

## 3-1-5 用户程序中轴的指定方法

在用户程序中, 将轴变量名称指定为运动控制指令的输入输出变量“Axis”。如下图所示, 对于系统定义变量的轴变量名称为“\_MC\_AX[0]”的轴, 将通过 Sysmac Studio 添加的轴变量名称变更为“MyAxis1”。



也可使用系统定义变量“\_MC\_AX[0]”替代“MyAxis1”。

运动控制指令的详情请参阅 □ “6-2 运动控制指令 (P.6-4)”。

各运动控制指令的详细功能 □ 请参阅“NJ/NX系列 指令基准手册 运动篇(SBCE-364)”中各指令的说明。

## 3-2 轴的设定步骤

下面，对使用 Sysmac Studio 首次设定伺服轴的步骤进行说明。

### 3-2-1 轴的设定流程

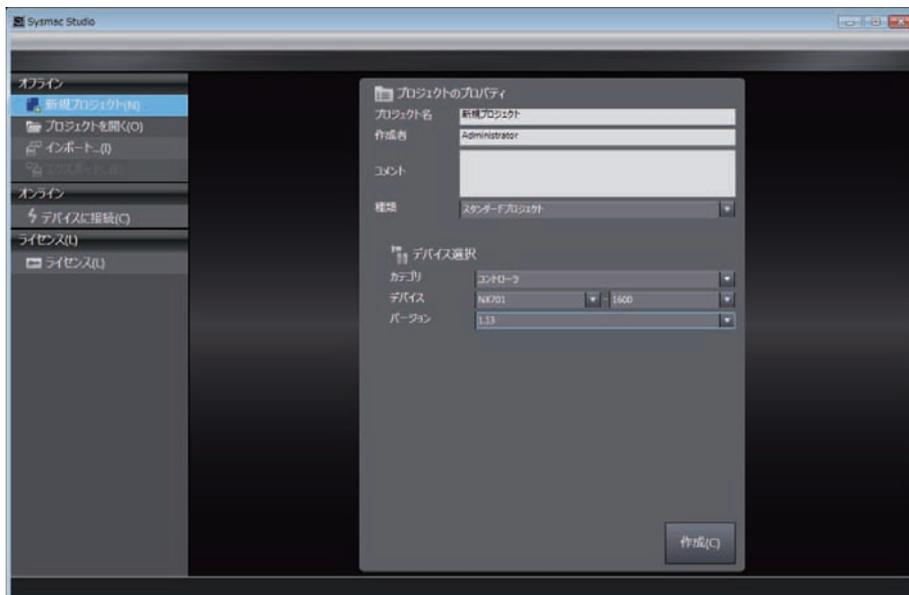


### 3-2-2 设定步骤

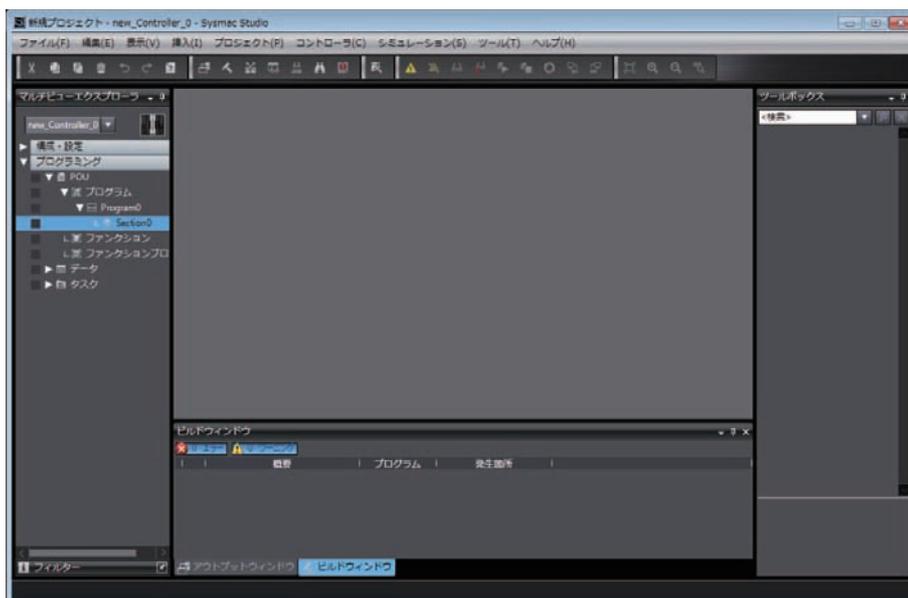
下面，对轴的设定方法进行说明。

#### Sysmac Studio 的启动

- 1 启动 Sysmac Studio，点击 [ 新建项目 ] 按钮。
- 2 设定项目的属性和设备的选择，点击 [ 创建 ] 按钮。



显示新建项目画面。

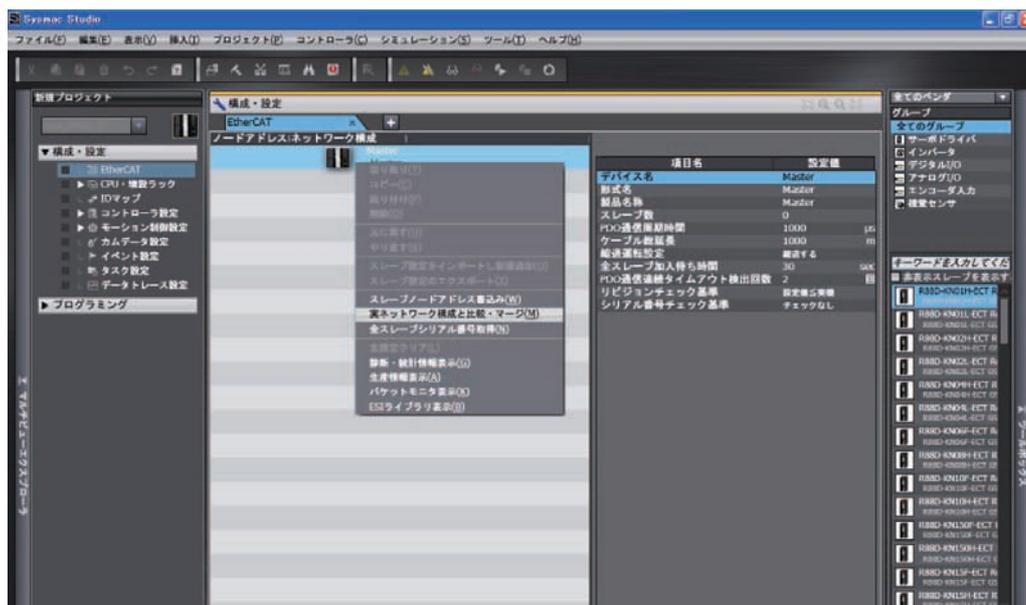


## EtherCAT 网络构成的创建

创建 EtherCAT 网络构成的方法分为在线创建和离线创建的 2 种方法。

### ● 在线创建的方法

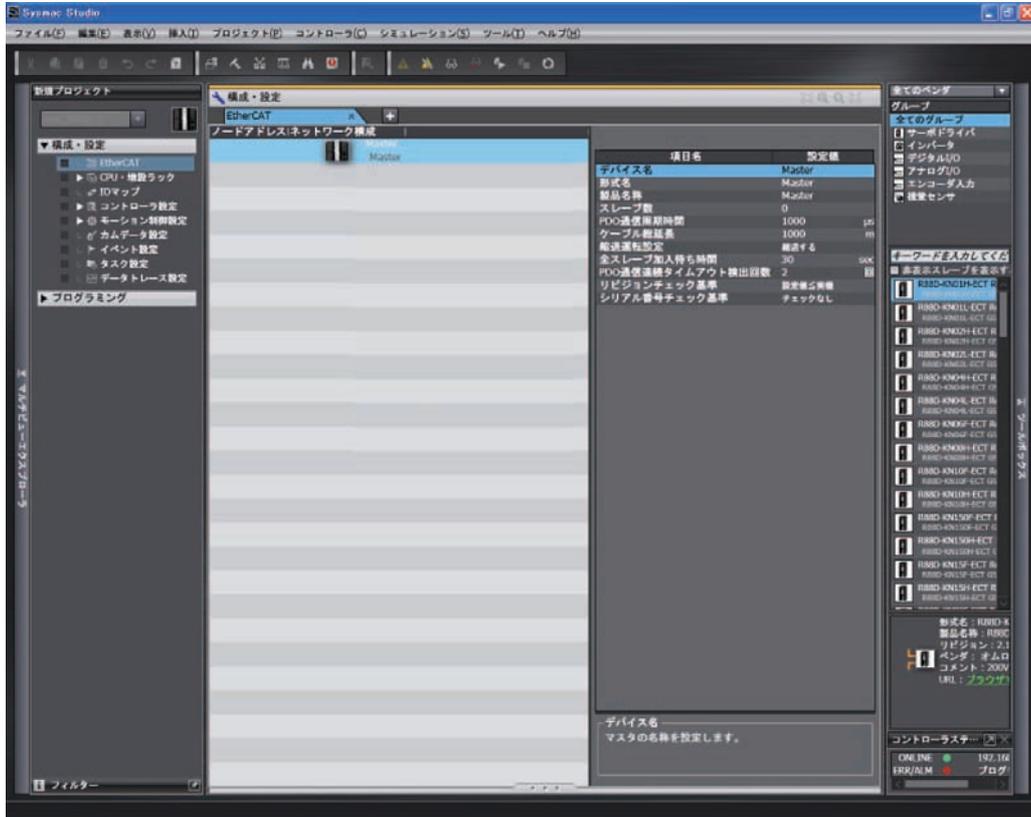
- 1 在 [ 多视图浏览器 ] 中双击 [ EtherCAT ]。  
显示 EtherCAT 编辑画面。
- 2 点击 [ 控制器 ] 菜单中的 [ 在线 ]，设为在线。
- 3 右击画面中的 [ Master ]，选择 [ 实际网络构成以及比较和合并 ]。



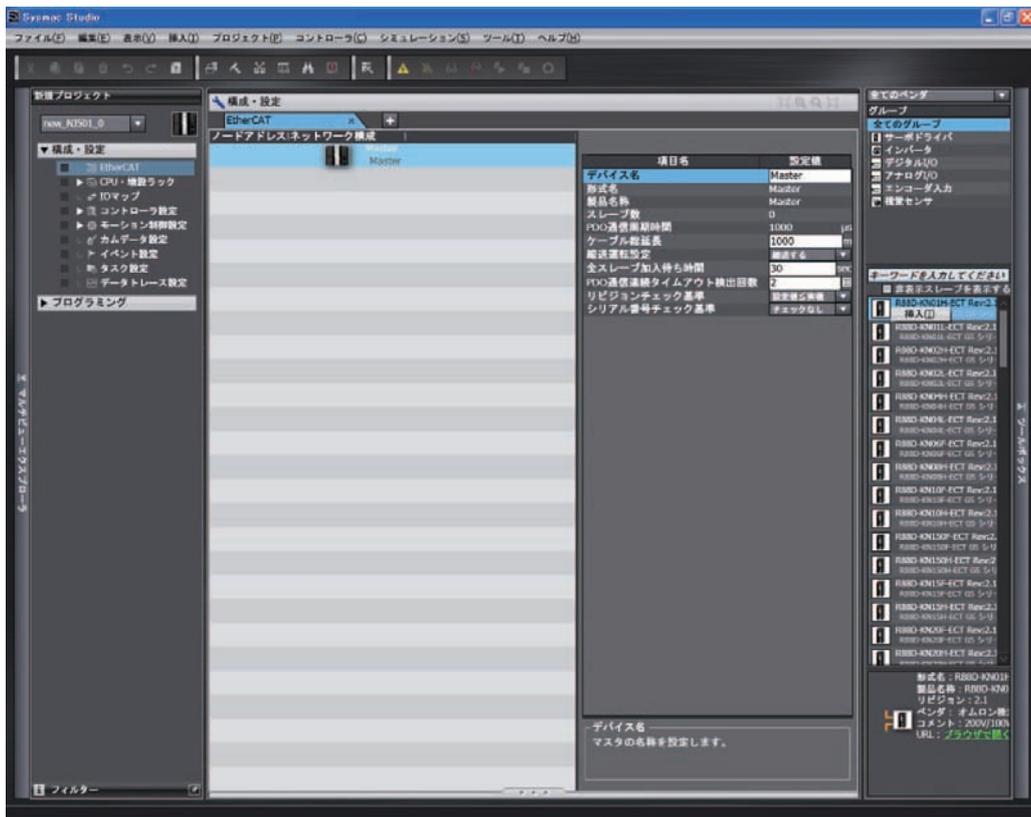
获取信息结束后，显示 EtherCAT 从站的实际设备构成。  
右击显示的实际设备构成，再点击 [ 使用实际网络构成 ]。

- 在线创建时

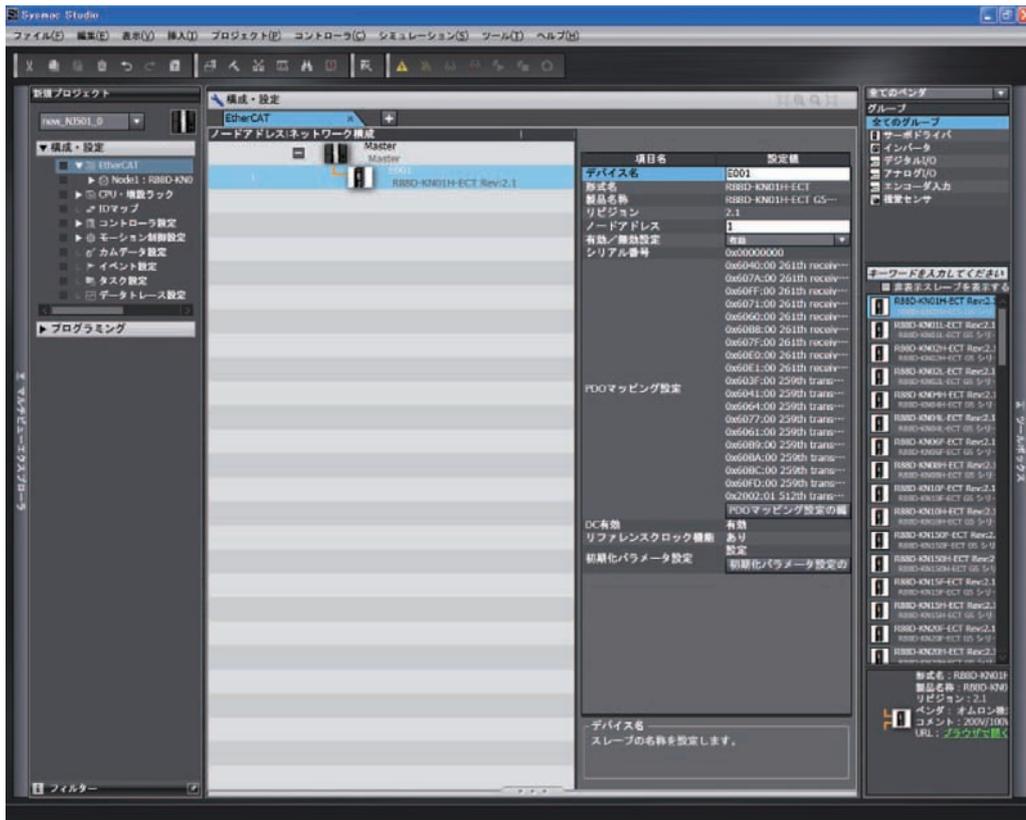
- 1 在 [多视图浏览器] 中双击 [EtherCAT]。  
显示 EtherCAT 编辑画面。



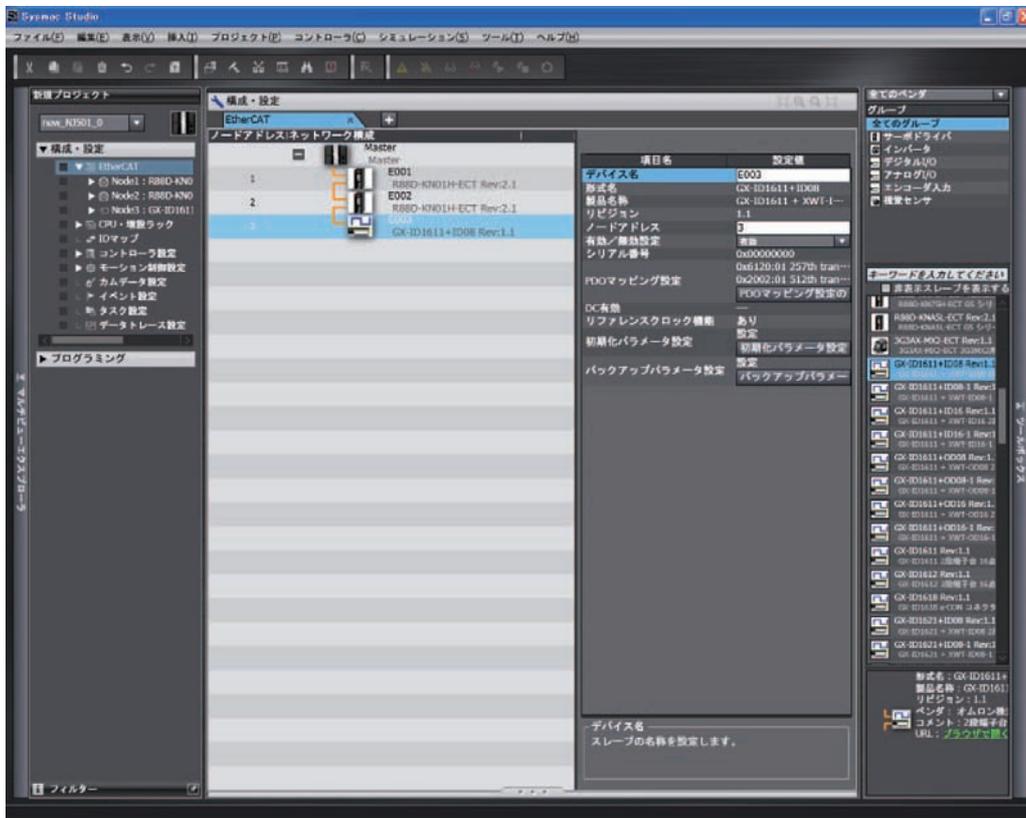
- 2 右击需连接的从站，再点击 [插入]。



显示插入的从站。



### 3 插入需使用的所有从站。



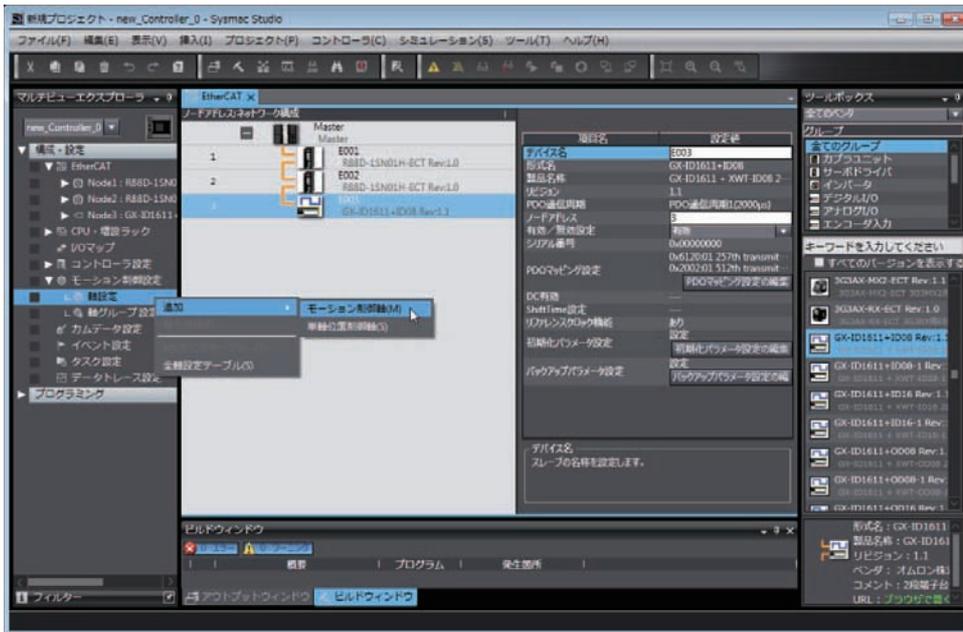
## 轴的添加

- 1 在多视图浏览器中右击 [ 轴设定 ], 点击 [ 添加 ]-[ 运动控制轴 ] 或 [ 单轴位置控制轴 ]。

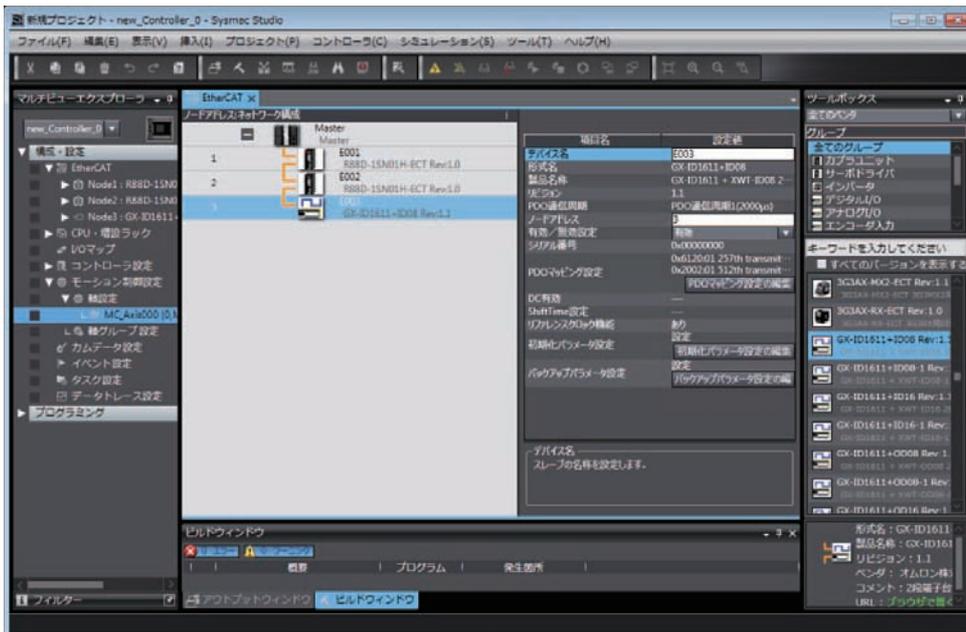


参考

NX1P2 CPU 单元显示 [ 单轴位置控制轴 ]。



在多视图浏览器中添加轴。  
添加的轴变量的默认名称为 “MC\_Axis000”。

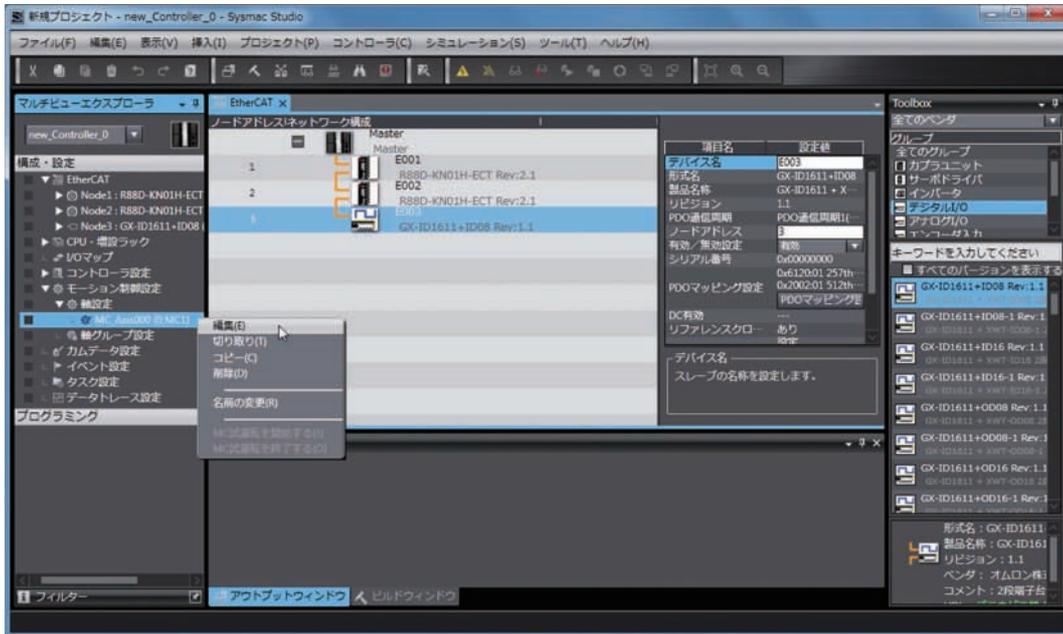


- 通过复制添加轴

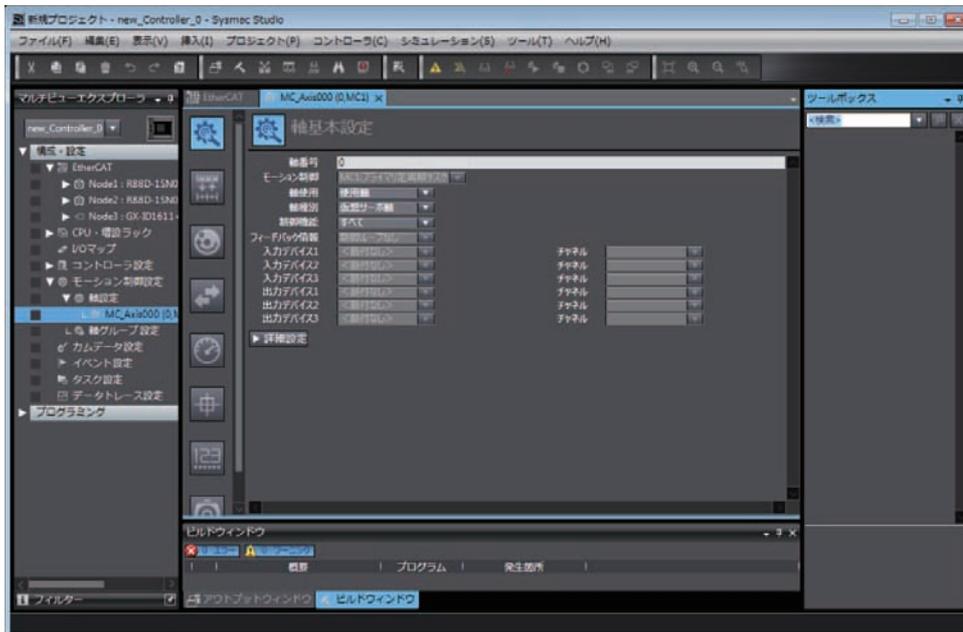
通过复制粘贴现有的轴设定, 也可添加轴。

## 轴的分配

- 1 右击多视图浏览器中添加的轴，再点击 [ 编辑 ]。



显示轴参数设定画面中的 [ 轴基本设定 ] 画面。



参考

NX1P2 CPU 单元显示 [ 控制功能 ]。

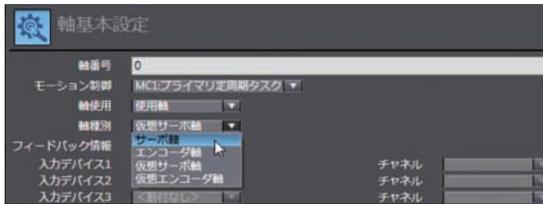
## 2 选择 [ 运动控制 ] 中的 [ 原始恒定周期任务 ] 或 [ 固定周期任务 ( 优先级 5) ]。



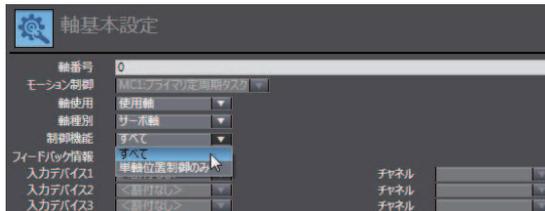
### 参考

NX701 CPU 单元时设定。

## 3 选择 [ 轴种类 ] 中的 [ 伺服轴 ]。



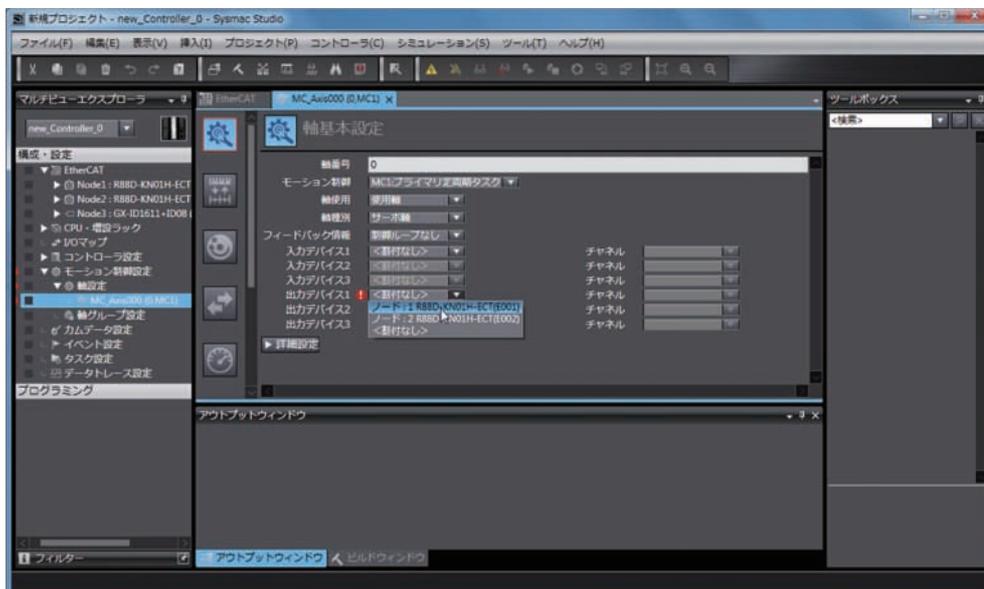
## 4 选择 [ 控制功能 ] 中的 [ 全部 ]。



### 参考

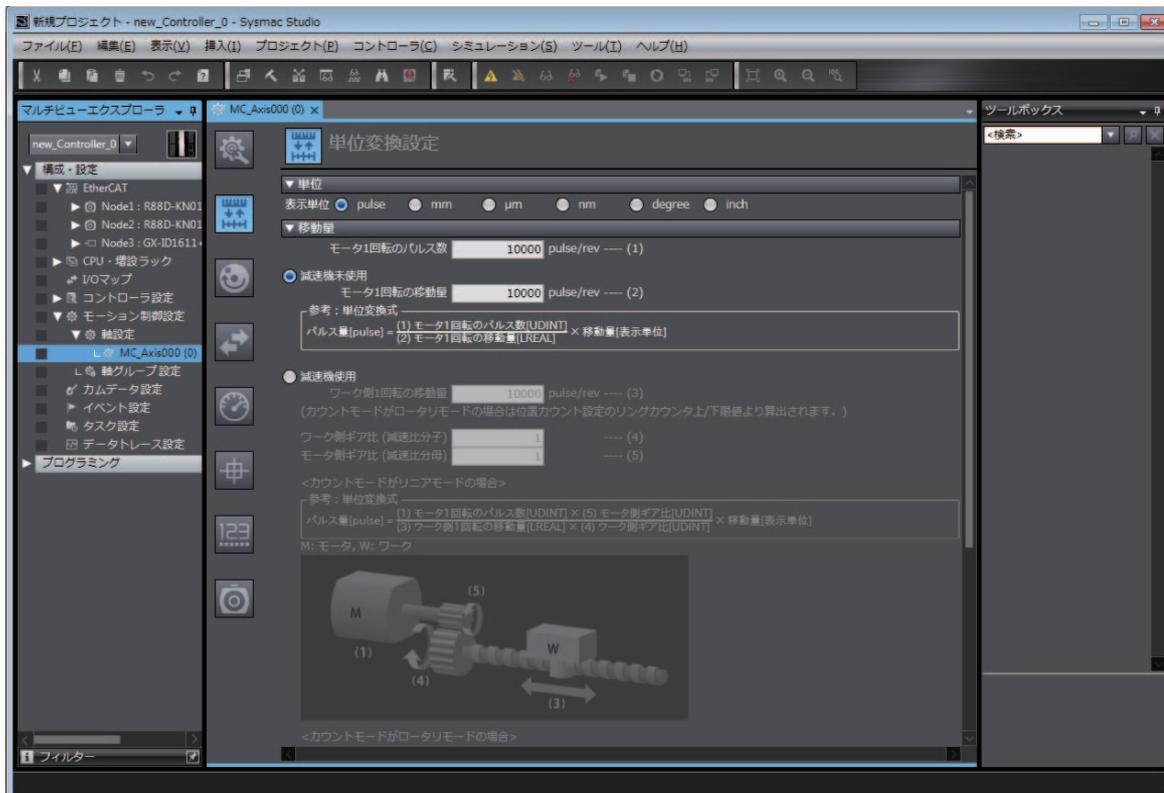
- NX1P2 CPU 单元时设定。
- 用作运动控制轴时，选择 “全部”。用作单轴位置控制轴时，请选择 “仅单轴位置控制”。

- 5** 选择要使用的伺服驱动器。  
通过该操作，可将 EtherCAT 从站的伺服驱动器用作轴。

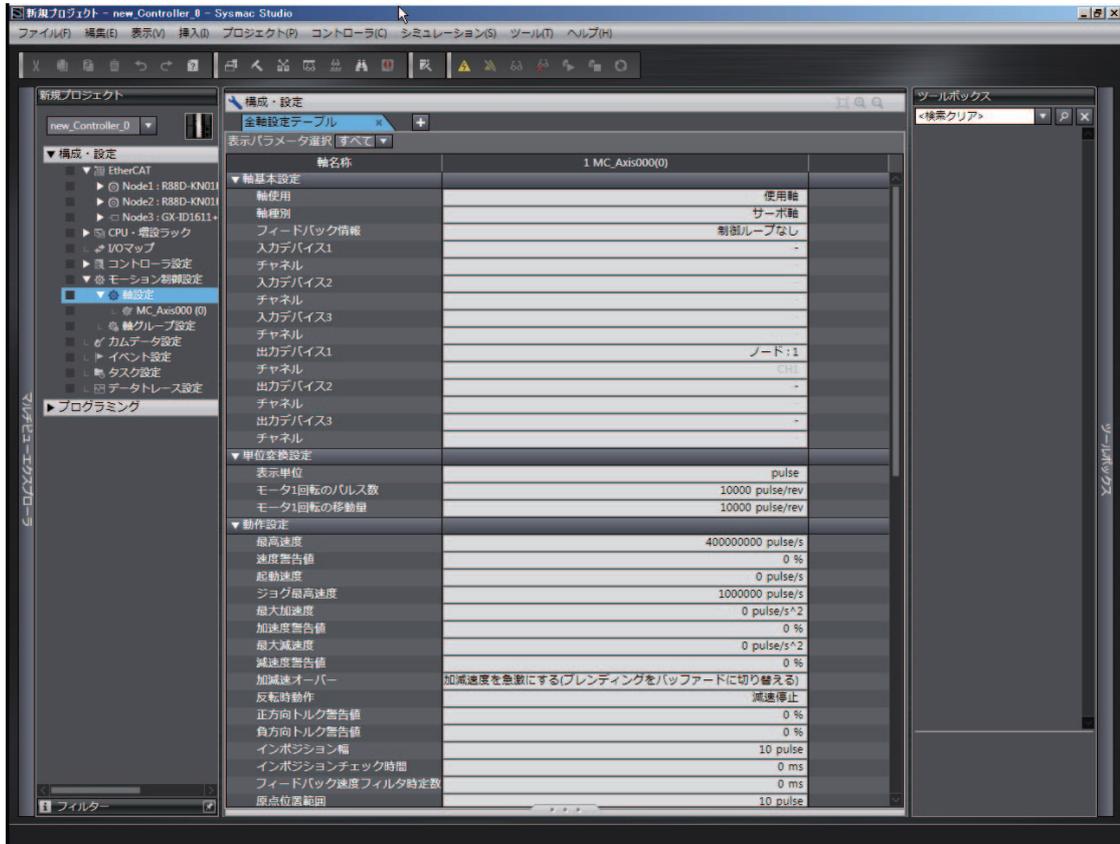


## 轴参数的设定

点击轴参数设定画面中的图标。  
显示按图标分类的轴参数设定画面。



通过右击多视图浏览器中的 [ 轴设定 ] 并点击 [ 所有轴设定表 ], 可在一览画面中设定所有轴的轴参数。



### 使用注意事项

请根据装置的动作条件, 为显示单位、电子齿轮 (单位转换公式)、最高速度及最大加减速速度等动作设定恰当的值。



### 参考

#### 用户程序中使用的轴变量名称的变更

变更已使用的轴变量名称时, 请执行以下 2 个操作。

- 声明变量的变量表中的轴变量名称的变更。
- 用户程序中轴变量名称的变更。

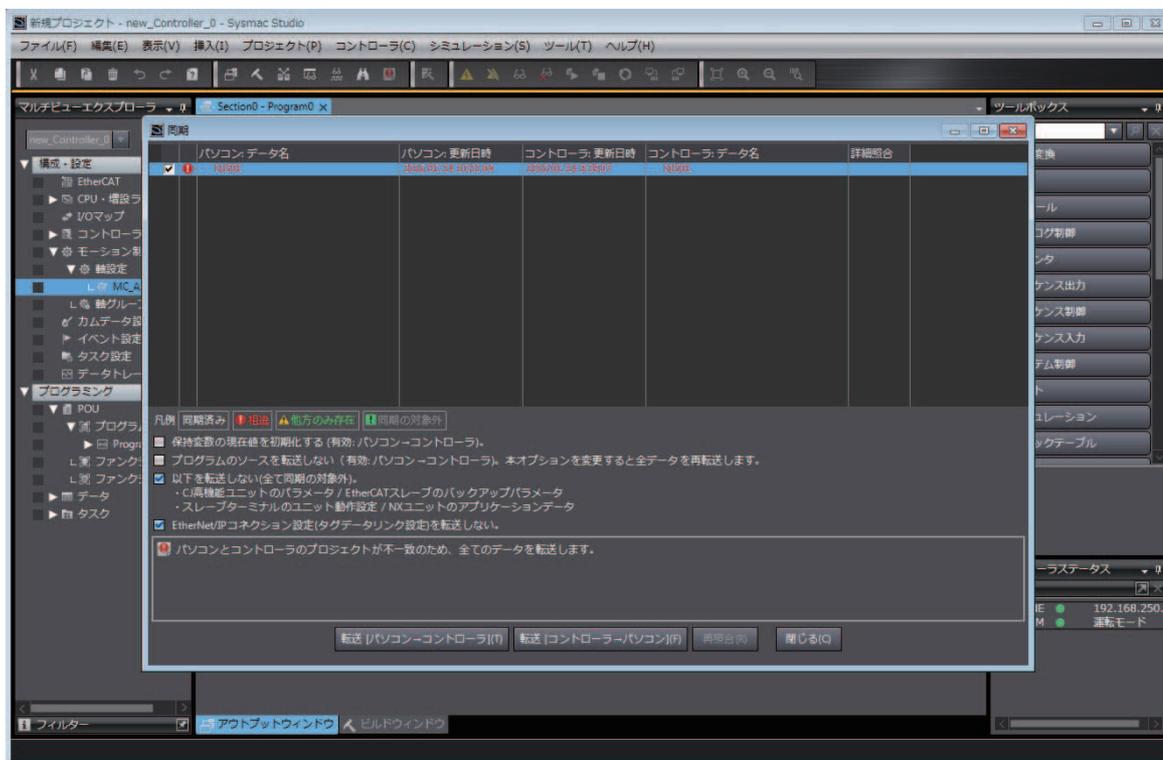
变更变量表中的轴变量名称后仍不反映至用户程序中的轴变量名称。

将未在变量表中声明的变量名称用于用户程序时, 会发生异常。请务必变更上述 2 项内容。

## 下载至 CPU 单元

下载至 CPU 单元时，使用 Sysmac Studio 的“同步”功能。

- 1 点击 [ 控制器 ] 菜单中的 [ 在线 ]，设为在线。
- 2 选择 [ 控制器 ] 菜单中的 [ 同步 ]，点击 [ 传送 [ 计算机→控制器 ] ]。



### 参考

MC 功能模块以欧姆龙制伺服驱动器 1S 系列 EtherCAT 通信内置型，以及 G5 系列 EtherCAT 通信内置型、NX 系列脉冲输出单元为连接对象。

#### 伺服驱动器的连接对象机型

- 伺服驱动器的对象机型的型号为 R88D-1SN □□□ -ECT，以及 R88D-KN □□□ -ECT、R88D-KN □□□ -ECT-L。
- R88D-KN □□□ -ECT-R 是仅适用于位置控制 (cyclic synchronous position) 模式的伺服驱动器。因此，使用速度控制 (cyclic synchronous velocity) 模式或转矩控制 (cyclic synchronous torque) 模式的功能无法使用。

#### 伺服驱动器的设定

- MC 功能模块使用伺服驱动器的输入信号和部分功能。为了正确使用 MC 功能模块的功能，需设定伺服驱动器的信号接线和对象。

关于设定内容，请参阅“A-1 与伺服驱动器 1S 系列之间的连接 (P.A-2)”或 □□ “A-2 与伺服驱动器 G5 系列之间的连接 (P.A-10)”。

使用 NX 系列脉冲输出单元时的设定请参阅 □□ “NX 系列位置接口单元用户手册 (SBCE-374)”。

## 3-3 轴组

下面，对 MC 功能模块的轴组进行说明。

### 3-3-1 轴组概要

同时运行直线插补和圆弧插补等多个轴时，使用轴组。

因此，轴组由多个轴构成。

使用 Sysmac Studio 设定轴组变量，可启动轴组的运动控制指令或获取轴组的状态。

MC 功能模块中最多可处理 64 组。

轴组规格的概要如下所示。

项目	规格		
	NX701	NX1P2	NJ 系列
轴组数	最多 64 组	最多 8 组	最多 32 组
构成轴数	最多 4 轴 / 轴组		

MC 功能模块的轴组中含有以下相关要素。

构成要素	内容	参考页
轴组参数	为 MC 功能模块控制的各轴组设定最高插补速度和最大插补加减速度等轴组动作。 设定轴组参数时，使用 Sysmac Studio。	P.3-20
轴组变量	轴组变量是由 MC 功能模块控制的各轴组的部分轴组参数、指令插补速度及异常信息等监控信息组成的系统定义变量。 通过 Sysmac Studio 的多视窗浏览器追加轴组时，创建轴组变量。可通过该视图设定轴组变量的名称（以下称为轴组变量名称）。	P.3-21
用户程序中轴组的指定方法	在用户程序中，使用运动控制指令执行运动控制。将执行多轴协调控制的运动控制指令称为“轴组指令”。 在指令的输入输出变量“AxesGroup(轴组)”中，通过系统定义变量的轴变量名称或 Sysmac Studio 设定的轴组变量名称指定轴组指令控制哪个轴组。	P.3-22



#### 使用注意事项

请勿通过 EtherCAT 网络上的轴和 CPU 单元上的轴构成轴组。

由于网络之间不进行同步，因此可能会导致指令时间发生偏差，引发意外动作。

## 3-3-2 轴组参数概要

## ● 轴组参数一览表

分类	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
中心点补偿容许率	

\*1. NX701 CPU 单元时设定。

轴组参数的详情请参阅 □ “5-3 轴组参数 (P.5-27)”。

## ● 使用轴组时须设定的项目

为通过 Sysmac Studio 创建的轴组设定以下项目后可使用轴组。

分类	参数名称	设定内容	参考页
轴组基本设定	轴组编号	按创建轴组的顺序自动设定。	P.5-28
	运动控制 *1	选择 [ 原始恒定周期任务 ]。	
	轴组使用	选择 [ 使用 ]。	
	轴构成	选择需控制的轴构成。	
	构成轴	选择分配至轴组的轴。	

\*1. NX701 CPU 单元时设定。



## 使用注意事项

- 请根据动作条件等为最高插补速度和停止方法等设定恰当的值。
- [ 控制功能 ] 的设定为 [ 仅单轴位置控制 ] 的轴，无法分配为轴组的构成轴。

### 3-3-3 轴组变量概要

轴组变量是由 MC 功能模块的各轴组的设定信息、当前位置及异常信息等监控信息组成的系统定义变量。通过 Sysmac Studio 创建轴组时，按创建的顺序将轴组变量登录至变量表。轴组变量的数据类型为结构体类型，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分为系统定义变量的轴组变量名称和通过 Sysmac Studio 添加轴组时的轴组变量名称的 2 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]”。

通过 Sysmac Studio 添加轴组时，为“\_MC\_GRP[0] ~ \_MC\_GRP[63]”设定相应的默认名称，即“MC\_Group000 ~ MC\_Group063”。编号按添加的顺序前移。可通过 Sysmac Studio 将各轴组的轴组变量名称变更为任意的变量名称。

用户程序中可使用系统定义变量的轴组变量名称或通过 Sysmac Studio 添加的轴组变量名称。

#### ● 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对象 <sup>*1</sup> )	通过 Sysmac Studio 添加的 轴组变量的默认名称	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. 变更 Sysmac Studio 的全局变量视图中的分配对象栏名称时会发生异常。



#### 参考

- 对于 NX701 CPU 单元，可使用“\_MC\_GRP[0-63]”或“\_MC1\_GRP[0-63]”，以及“\_MC2\_GRP[0-63]”。  
对于 NX1P2 CPU 单元，只能使用“\_MC\_GRP[0-7]”。  
对于 NJ 系列 CPU 单元，只能使用“\_MC\_GRP[0-31]”。
- 对于 NX701 CPU 单元，“\_MC\_GRP[0-63]”和“\_MC1\_GRP[0-63]”的轴组号相同时查看相同值。任意轴组变量皆可使用，也可混合使用。
- 分配给原始恒定周期任务的轴组变量为“\_MC\_GRP[0-63]”或“\_MC1\_GRP[0-63]”。  
分配给固定周期任务（执行优先级 5）的轴组变量为“\_MC2\_GRP[0-63]”。

#### 轴组变量的层次和轴组变量名称的变更示例

下面以 \_MC\_GRP[0] 为例进行说明。其他轴组变量也一样。

_MC_GRP[0]	: 轴组变量
_MC_GRP[0].Status	: 表示“轴组状态”的层次
:	:
_MC_GRP[0].Cmd	: 表示“轴组指令值”的层次
_MC_GRP[0].Cmd.Vel	: 表示“指令插补速度”的变量
_MC_GRP[0].Cmd.AccDec	: 表示“指令插补加减速速度”的变量
:	:
_MC_GRP[0].Cfg	: 表示“轴组基本设定”的层次
_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” 的层次
_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)” 轴号的变量
:	
_MC_GRP[0].Kinematics.Axis[3]	: 表示 “构成轴 (轴 A3)” 轴号的变量
_MC_GRP[1]	: 轴组变量
:	

#### ● 例

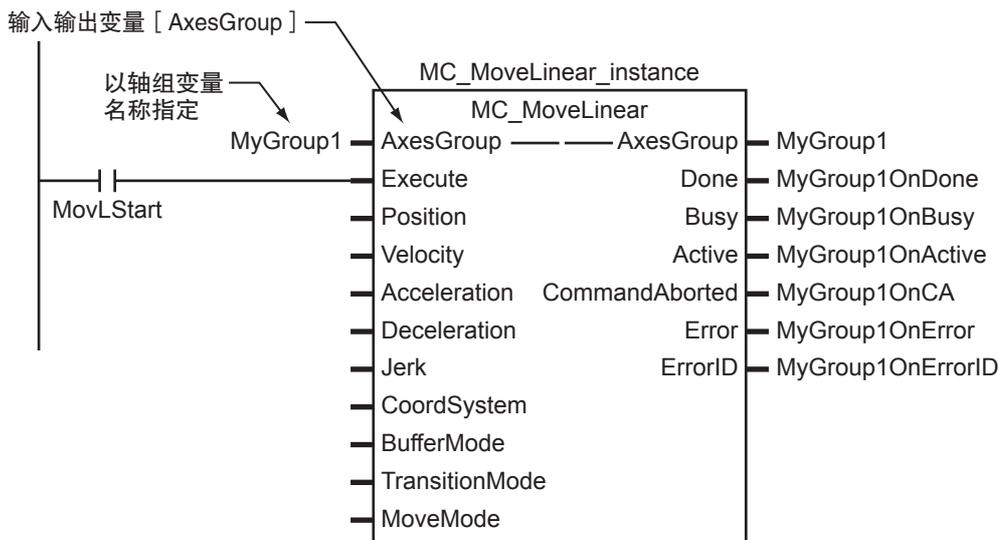
将 “MC\_Group000” 变更为 “MyGroup1” 时, 表示 “指令插补速度” 的变量可使用 “MyGroup1.Cmd.Vel” 和 “\_MC\_GRP[0].Cmd.Vel” 中的任意一个。

轴组变量的详情请参阅 □ “轴组变量 (P.6-27)”。

### 3-3-4 用户程序中轴组的指定方法

在用户程序中, 将轴组变量名称指定为运动控制指令的输入输出变量 “AxesGroup”。

如下图所示, 对于系统定义变量的轴组名称为 “\_MC\_GRP[0]” 的轴组, 将通过 Sysmac Studio 添加时的轴组变量名称变更为 “MyGroup1”。



也可用系统定义变量 “\_MC\_GRP[0]” 替代 “MyGroup1”。

运动控制指令的详情请参阅 □ “6-2 运动控制指令 (P.6-4)”。

各运动控制指令的详细功能 □ 请参阅 “NJ/NX系列 指令基准手册 运动篇(SBCE-364)” 中各指令的说明。

## 3-4 轴组的设定步骤

下面，对使用 Sysmac Studio 设定轴组的步骤进行说明。  
不使用直线插补或圆弧插补等轴组指令时，无需进行设定。

### 3-4-1 轴组的设定流程

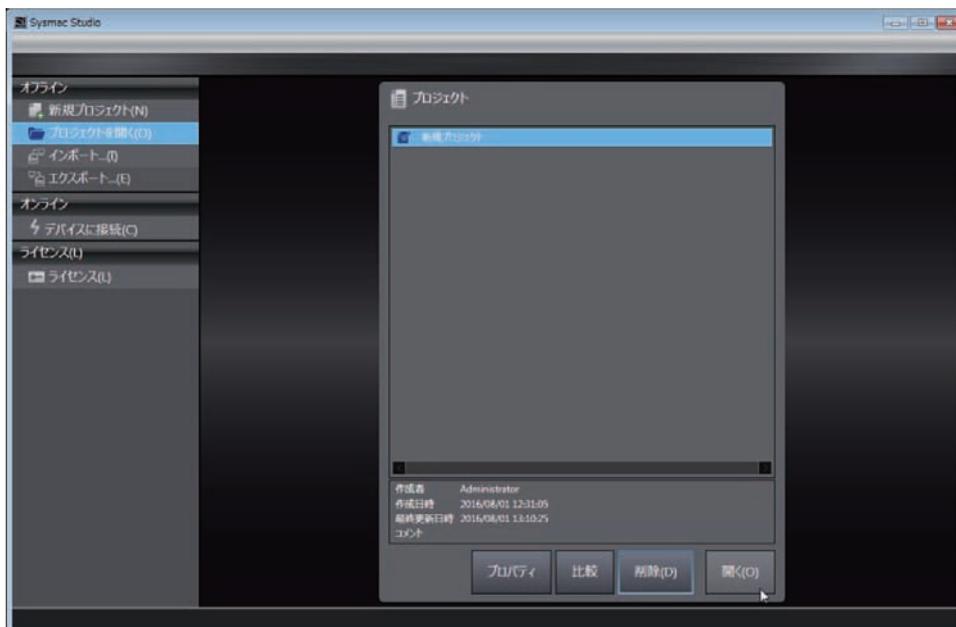


### 3-4-2 设定步骤

下面，对在已添加轴的现有项目中设定轴组的方法进行说明。

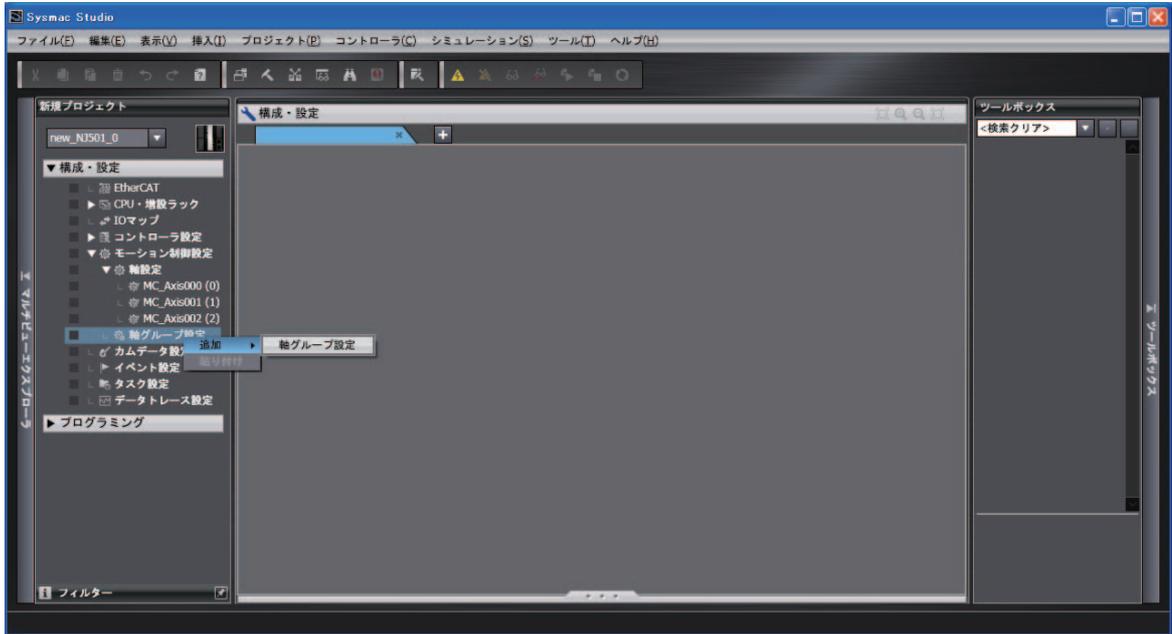
#### Sysmac Studio 的启动

启动 Sysmac Studio，打开现有项目。



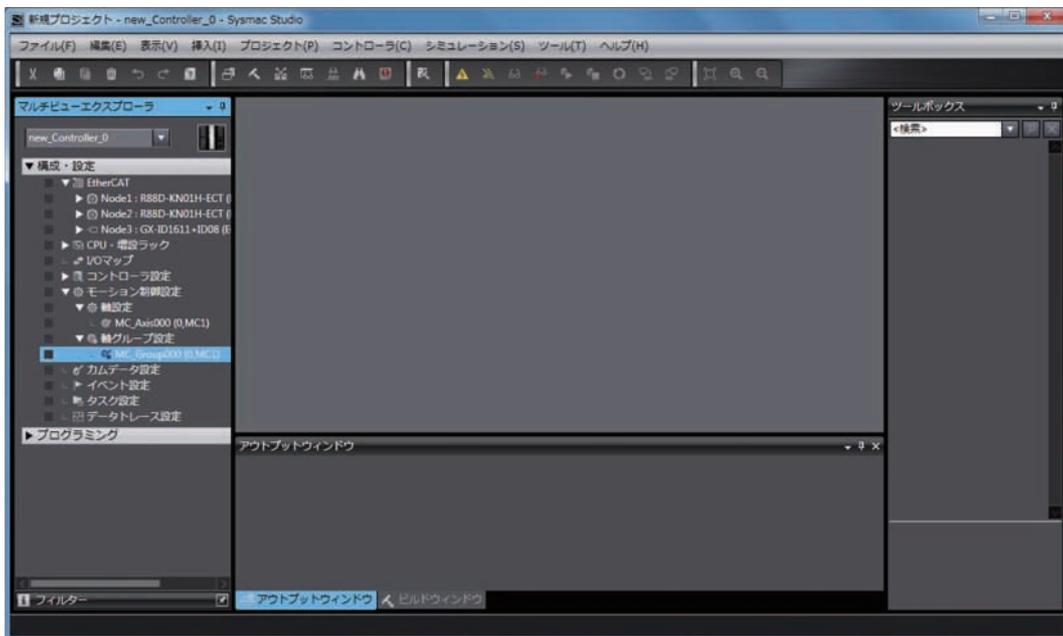
## 轴组的添加

右击多视图浏览器中的 [ 轴组设定 ], 再点击 [ 添加 ] 中的 [ 轴组设定 ]。



在多视图浏览器中添加轴组。

添加的轴组变量的默认名称为 “MC\_Group000”。

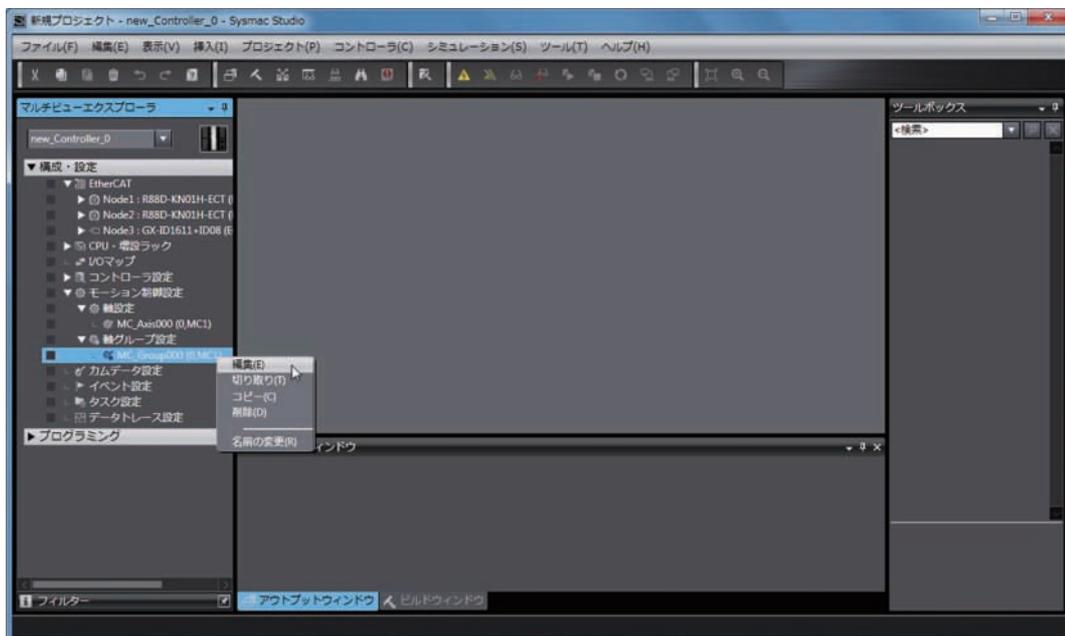


### ● 通过复制创建轴

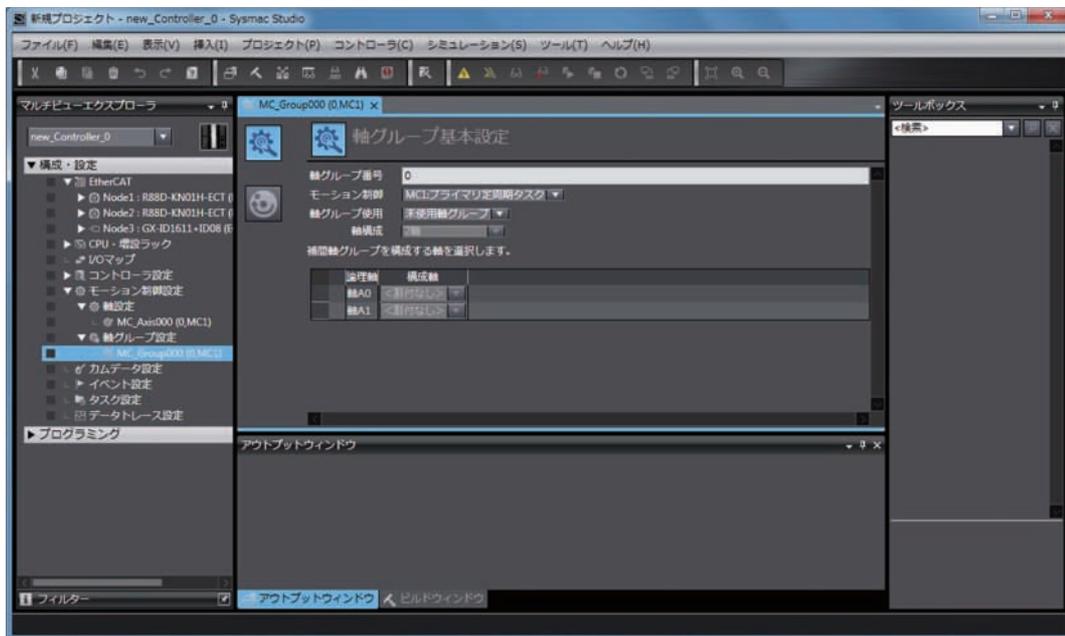
也可通过复制现有项目的轴组新建轴组。

## 轴组参数的设定

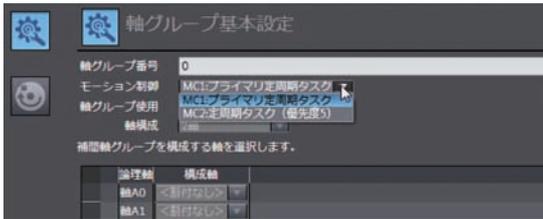
- 1 右击多视图浏览器中添加的轴组，再点击 [ 编辑 ]。



显示轴组参数设定画面中的 [ 轴组基本设定 ] 画面。



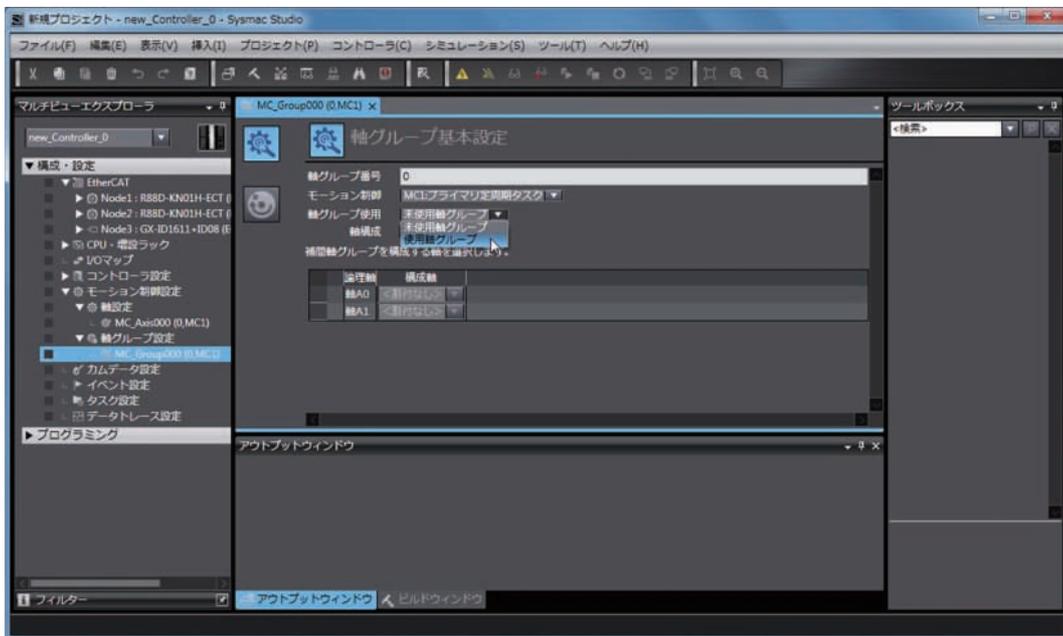
**2** 选择 [ 运动控制 ] 中的 [ 原始恒定周期任务 ] 或 [ 固定周期任务 ( 优先级 5) ]。



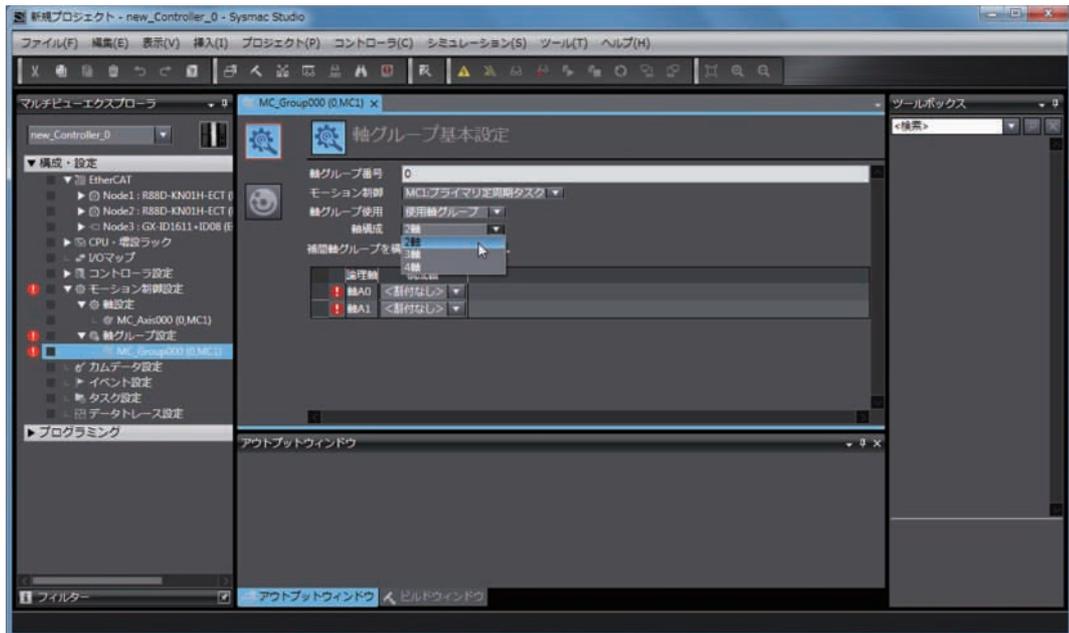
参考

NX701 CPU 单元时设定。

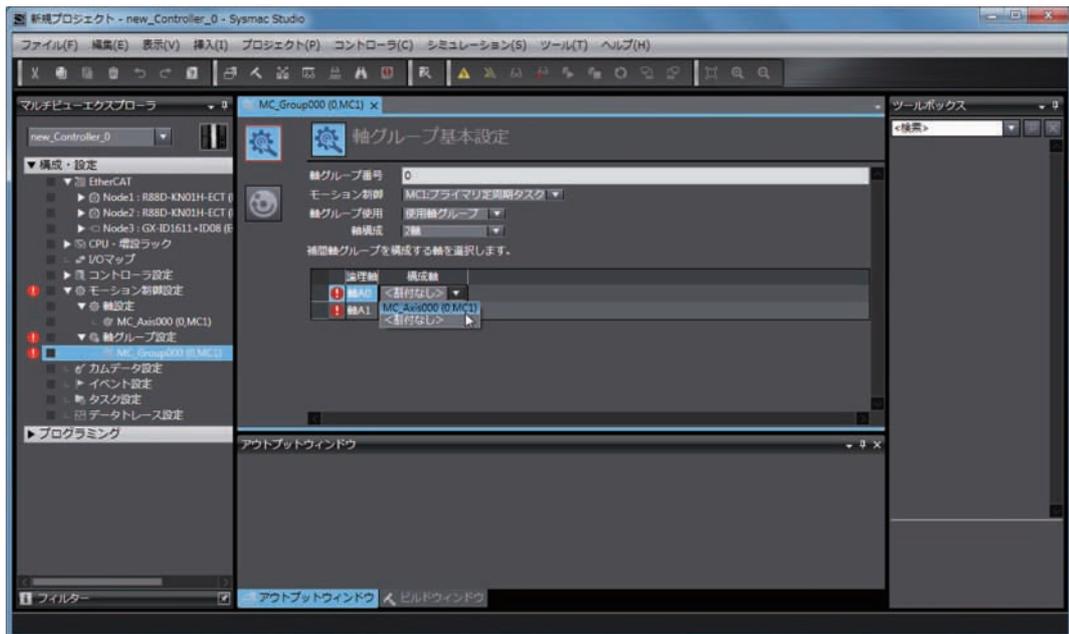
**3** 通过 [ 轴组使用 ], 选择 [ 使用轴组 ]。



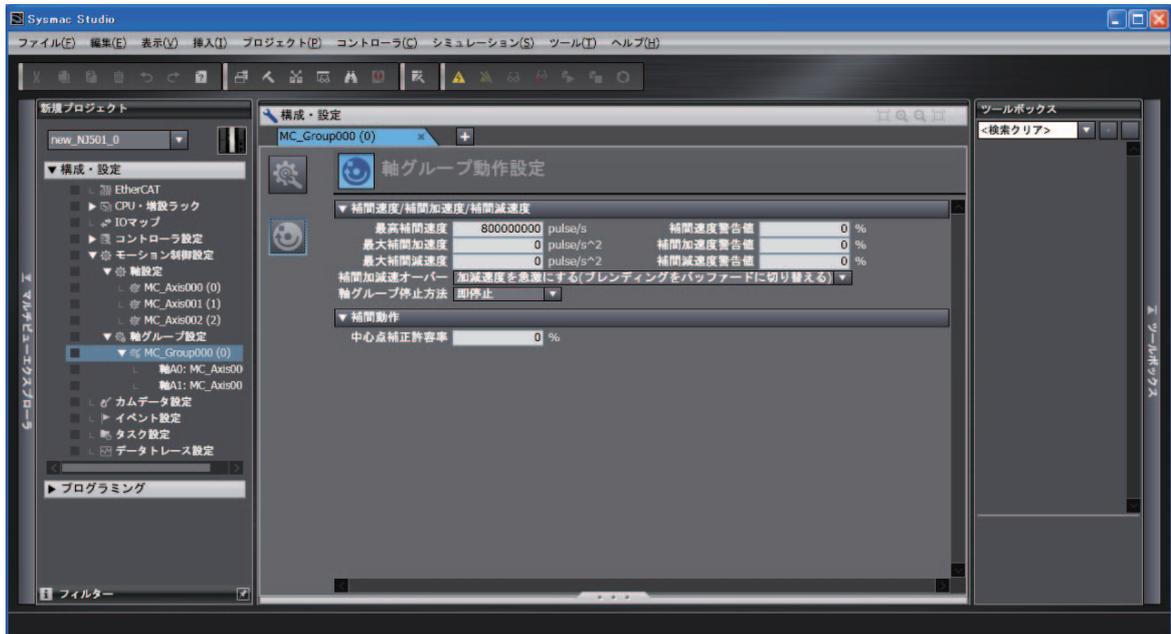
#### 4 通过 [ 轴构成 ] 选择轴组的构成。以下为选择 2 轴构成时的情形。



#### 5 在 [ 逻辑轴 ] 中分配需使用的轴。



- 6 点击下方的图标，显示 [ 轴组动作设定 ] 画面。  
请根据装置的动作条件分别设定恰当的值。



#### 参考

##### 用户程序中使用的轴组变量名称的变更

变更已使用的轴组变量名称时，请执行以下 2 个操作。

- 声明变量的变量表中的轴组变量名称的变更。
- 用户程序中轴组变量名称的变更。

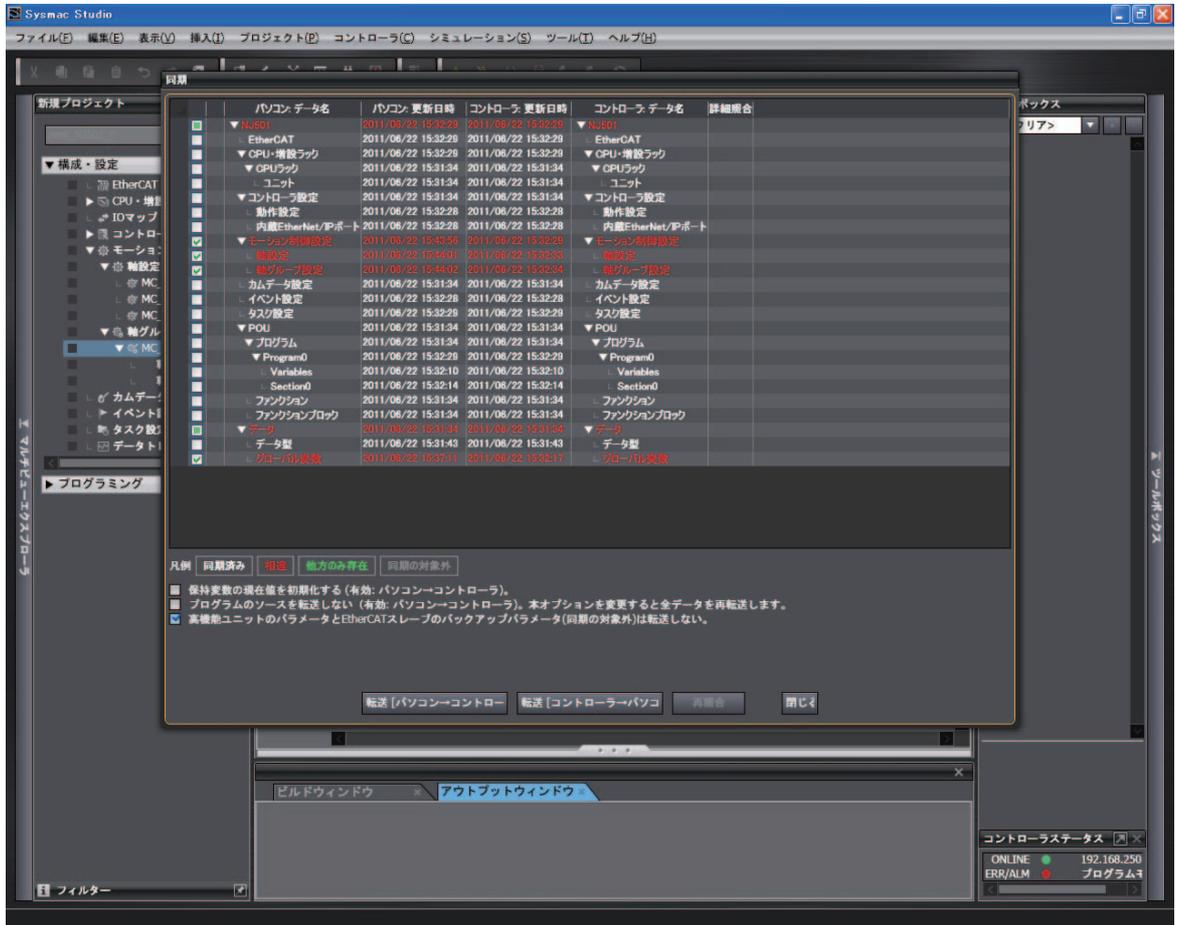
变更变量表中的轴组变量名称后仍不反映至用户程序中的轴组变量名称。

将未在变量表中声明的变量名称用于用户程序时，会发生异常。请务必变更上述 2 项内容。

## 下载至 CPU 单元

下载至 CPU 单元时，使用 Sysmac Studio 的“同步”功能。

- 1 点击 [ 控制器 ] 菜单中的 [ 在线 ]，设为在线。
- 2 选择 [ 控制器 ] 菜单中的 [ 同步 ]，点击 [ 传送 [ 计算机→控制器 ] ]。





# 4

## 通过 Sysmac Studio 确认接线

下面，对 Sysmac Studio 中的 MC 试运行功能进行说明。通过 MC 试运行功能，无需程序即可确认传感器信号的监控情况和伺服电机的接线情况。

4

4-1 Sysmac Studio 的功能 .....	4-2
4-1-1 MC 试运行功能一览 .....	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 试运行功能的启动步骤 .....	4-5
4-2 传感器信号的监控 .....	4-7
4-3 电机动作的确认 .....	4-8
4-3-1 伺服 ON .....	4-8
4-3-2 微动移动 .....	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

下面使用 MC 试运行功能，对接线和基本设定的确认进行说明。  
通过 Sysmac Studio 中的 MC 试运行功能，无需程序即可确认接线情况。

### 4-1-1 MC 试运行功能一览

MC 试运行功能含有以下功能。

分类	功能	说明	设定项目 / 监控项目
轴动作	减速停止	MC 试运行时减速停止	-
	伺服 ON/OFF	伺服 ON/OFF	-
	异常解除	解除 MC 功能模块发生的异常	-
	微动移动	向正方向或负方向微动移动	目标速度 加速度 / 减速度
	绝对值定位	执行绝对值定位 *1	目标位置 目标速度 加速度 / 减速度 跃度
	相对值定位	执行相对值定位	移动距离 目标速度 加速度 / 减速度 跃度
	原点复位	设定原点复位参数后执行原点复位	原点复位参数
监控	异常一览表	监控 MC 功能模块发生的异常	MC 通用异常 轴异常 轴组异常
	轴状态	监控轴的状态	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
	监控当前位置	监控当前位置	当前位置 (指令 / 反馈)
	监控当前速度	监控当前速度	当前速度 (指令 / 反馈)
	伺服驱动器 状态	监控伺服驱动器的状态	伺服 ON/OFF 伺服待机 主电路电源
	输入信号	监控输入信号的状态	正方向极限输入 / 负方向极限输入 即停输入 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定位。

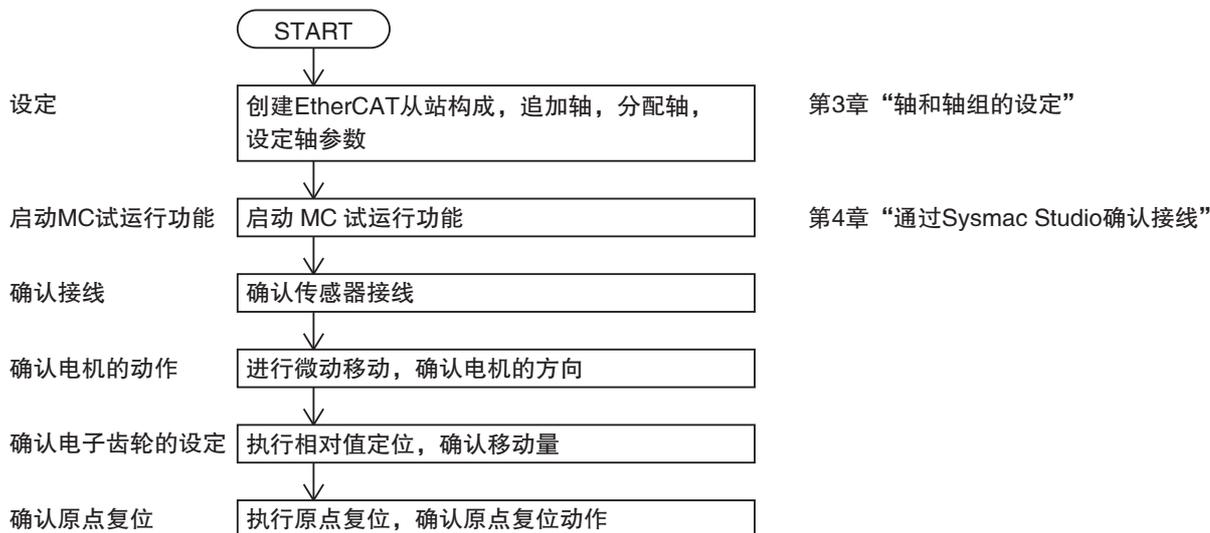
详情请参阅  “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 的 MC\_MoveAbsolute(绝对值定位) 指令。  
(注) MC 试运行功能以欧姆龙制伺服驱动器 1S 系列及 G5 系列、NX 系列脉冲输出单元为对象。请勿用于其它公司生产的驱动器。

## 4-1-2 使用步骤

执行 MC 试运行时，请务必确认以下 2 点。

- 是否在线连接了 Sysmac Studio 和控制器主体。
- 是否通过其它 Sysmac Studio 启动了 MC 试运行。

请在确认结束后按以下流程进行操作。



### 使用注意事项

- 通过 Sysmac Studio 发送的指令执行伺服 ON、微动移动、相对值定位、绝对值定位、原点复位后，以设定的速度运行电机。  
请充分确认不会因运行电机而导致危险后再执行。
- 通过 Sysmac Studio 操作时，请在外部设置紧急停止电路，以确保可安全停止电机。计算机发生异常等时，可能无法从 Sysmac Studio 发送指令。
- 通过 Sysmac Studio 操作时，请设定 EtherCAT 通信并建立通信后再执行。
- 使用 NX 系列 位置接口单元时的步骤，请参阅 □ “NX 系列 位置接口单元 用户手册 (SBCE-374)”。



### 参考

通过以下操作，可随时结束 MC 试运行。

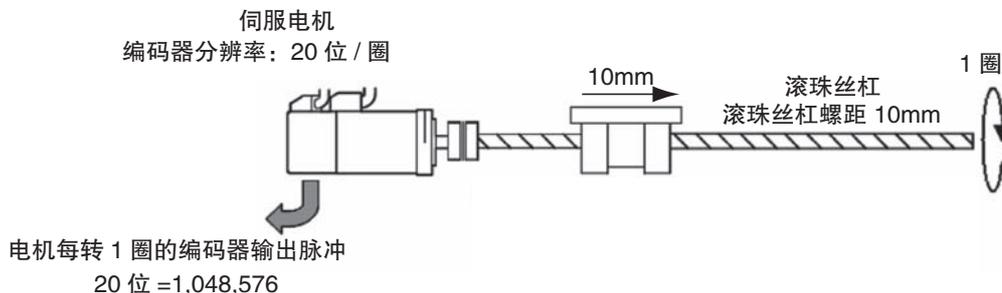
- 在 Sysmac Studio 的 [ 控制器 ] 菜单中的 [ MC 试运行 ] 中选择 [ 结束 ]。
- 右击 Sysmac Studio 的多视图浏览器的对称轴，选择 [ 结束 MC 试运行 ]。
- 关闭 Sysmac Studio 的 MC 试运行画面。
- 关闭 Sysmac Studio。

操作方法的详情请参阅 □ “Sysmac Studio Version 1 操作手册 (SBCA-362)”。

## 4-1-3 轴参数的设定示例

启动 Sysmac Studio 的 MC 试运行功能时，轴参数的设定如下。

下面以 1 轴装置为例对设定例进行说明。



设定项目	设定内容
轴变量名称	Axis1 *1
轴号	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	$\mu\text{m}$
电机转 1 圈的脉冲数	1,048,576 *4
电机转 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
软件限制功能	对指令位置有效，执行立即停止
正方向软件限制	500,000 *8
负方向软件限制	0 *8
计数模式	线性模式

\*1. 如果是多个轴，则按轴设定不同的变量名称。

\*2. 如果是多个轴，则按轴设定不同的值。

\*3. 设定与伺服驱动器的节点地址相同的值。如果是多个轴，则按轴设定不同的值。

\*4. 位置的指令单位为  $1(\mu\text{m})$ 。

\*5. 最高速度为  $3,000\text{r}/\text{min} = 30\text{m}/\text{min} = 0.5\text{m}/\text{s} = 500,000 \mu\text{m}/\text{s}$ 。

\*6. 微动最高速度为最高速度的 10% 即  $0.05\text{m}/\text{s} = 50,000 \mu\text{m}/\text{s}$ 。

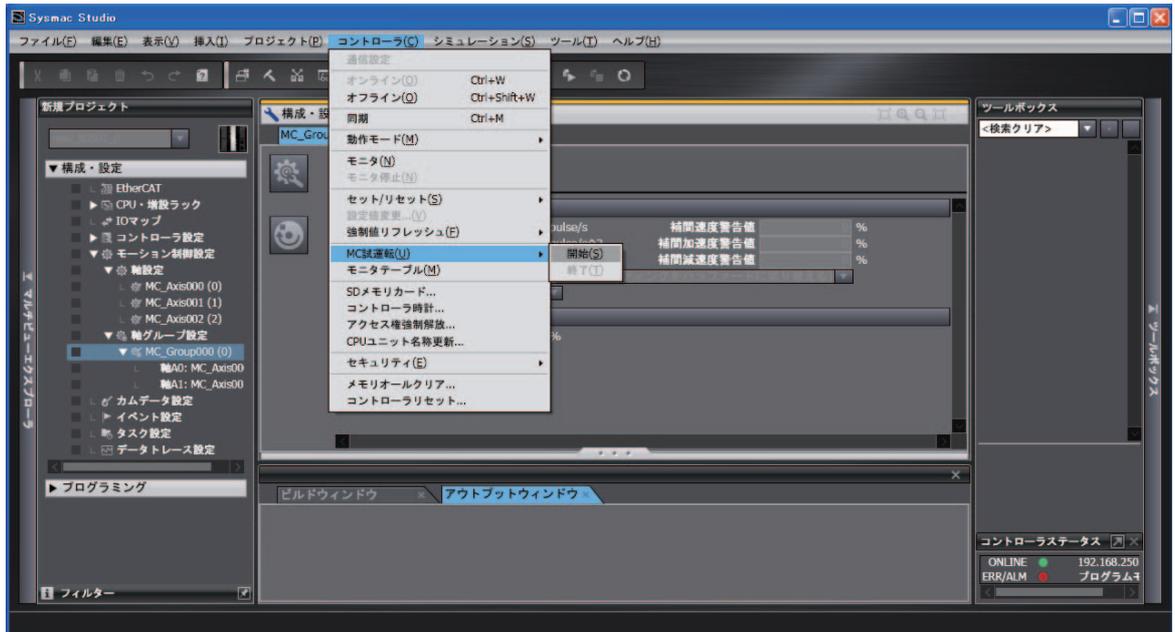
\*7. 最大加速度和最大减速度为  $5\text{m}/\text{s}^2$ 。达到最高速度  $3,000\text{r}/\text{min}$  的加速时间设为 0.1s。

\*8. 设定装置可动范围内侧的位置。正方向软件限制的设定值为  $50\text{cm} = 500,000 \mu\text{m}$ 。

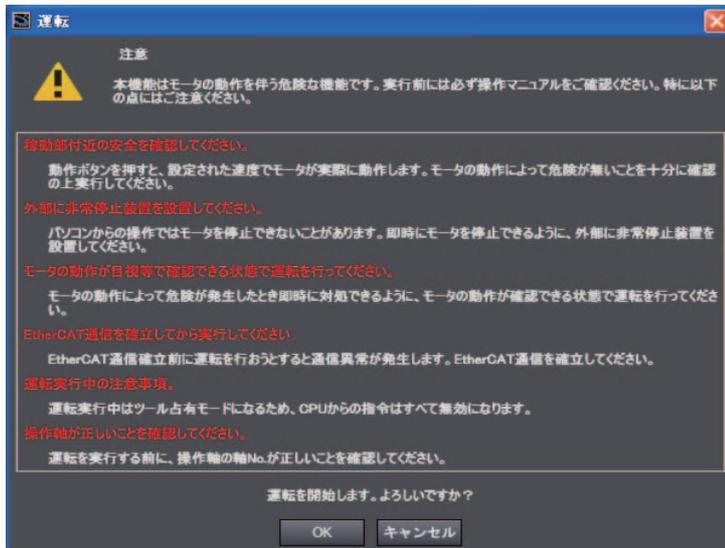
### 4-1-4 MC 试运行功能的启动步骤

通过 Sysmac Studio 启动 MC 试运行功能。

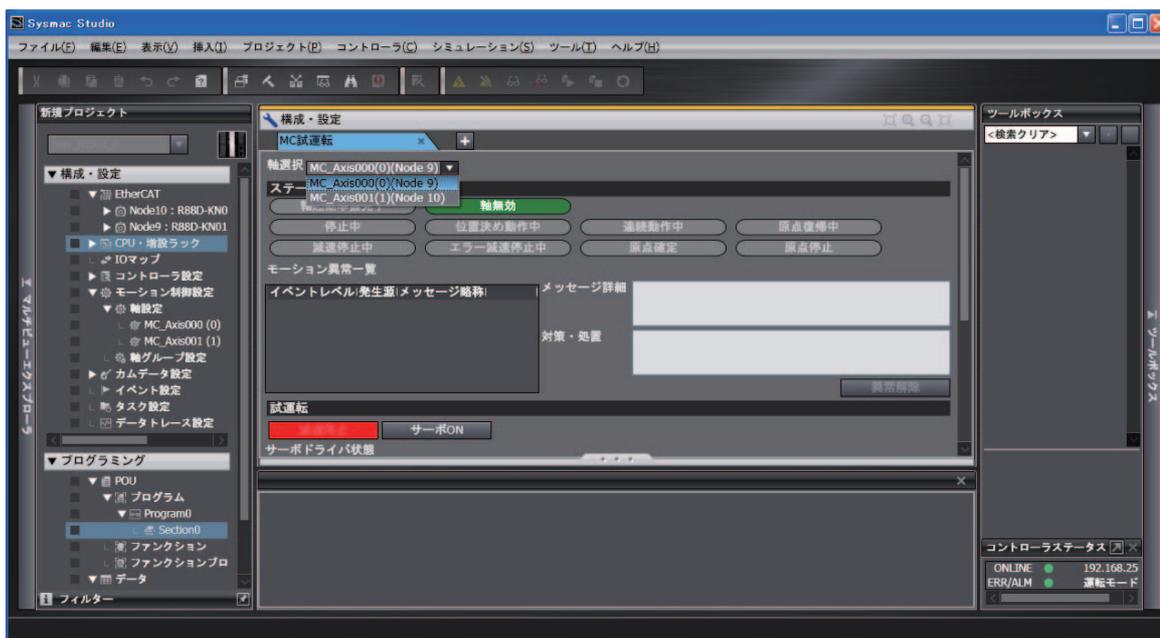
- 1 启动 Sysmac Studio，打开完成伺服轴设定的现有项目。
- 2 点击 [ 控制器 ] 菜单中的 [ 在线 ]，设为在线。
- 3 在 [ 控制器 ] 菜单中的 [ MC 试运行 ] 中点击 [ 开始 ]。



显示以下注意对话框，请仔细阅读内容并确认安全后点击 [OK]。



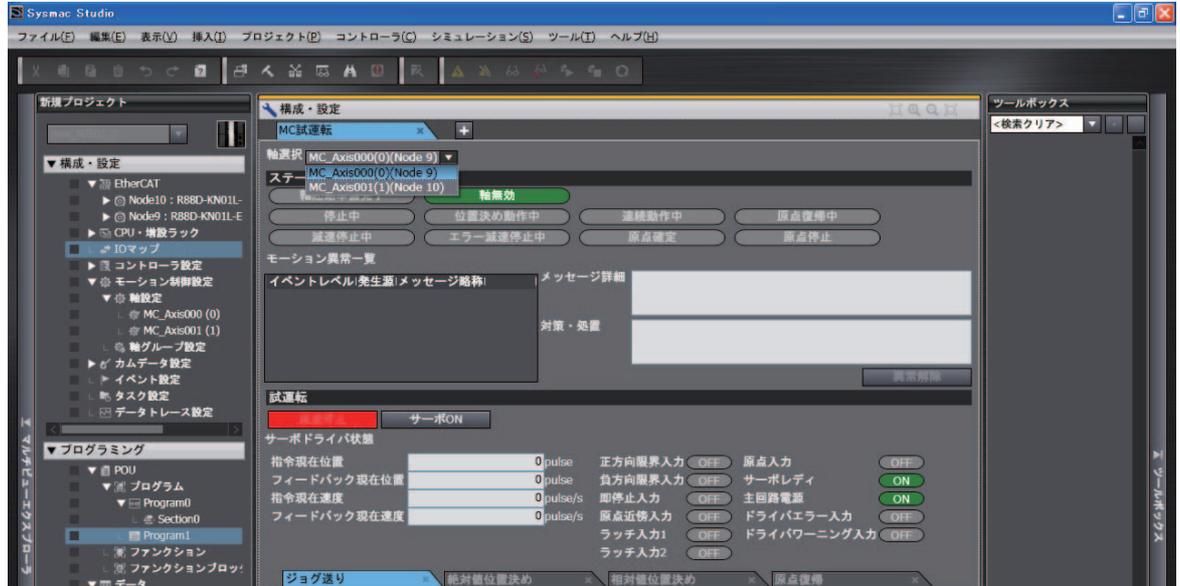
显示 MC 试运行画面。



## 4-2 传感器信号的监控

确认传感器信号的接线情况时，使用各输入信号的显示。

### 1 选择需通过 MC 试运行画面进行确认的轴。



### 2 将各输入信号连接的传感器 ON/OFF，确认监控画面中的相应信号是否正确 ON/OFF。

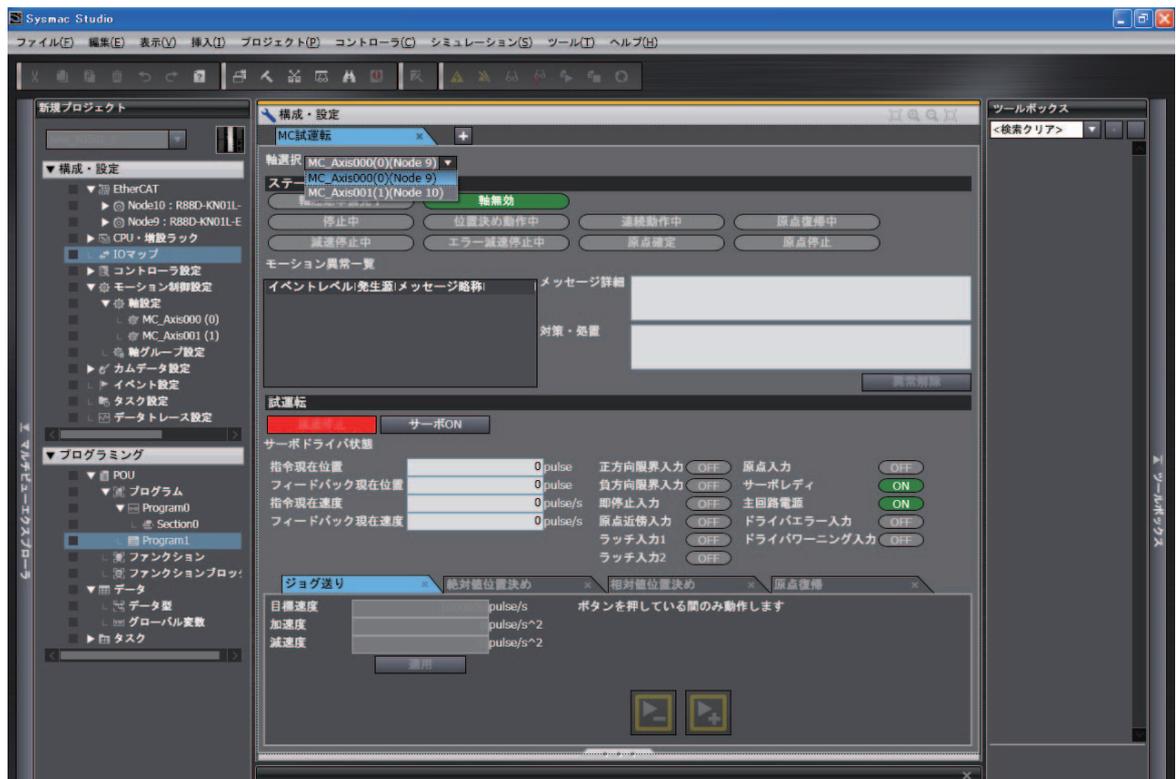
## 4-3 电机动作的确认

确认电机动作时，使用 MC 试运行的各功能。

### 4-3-1 伺服 ON

按下 [伺服 ON] 按钮，可在伺服 ON/OFF 之间进行切换。

#### 1 选择需伺服 ON 的轴。



2 按下 [伺服 ON] 按钮，设为伺服 ON 状态。

3 按下 [伺服 OFF] 按钮，返回至伺服 OFF 状态。



#### 使用注意事项

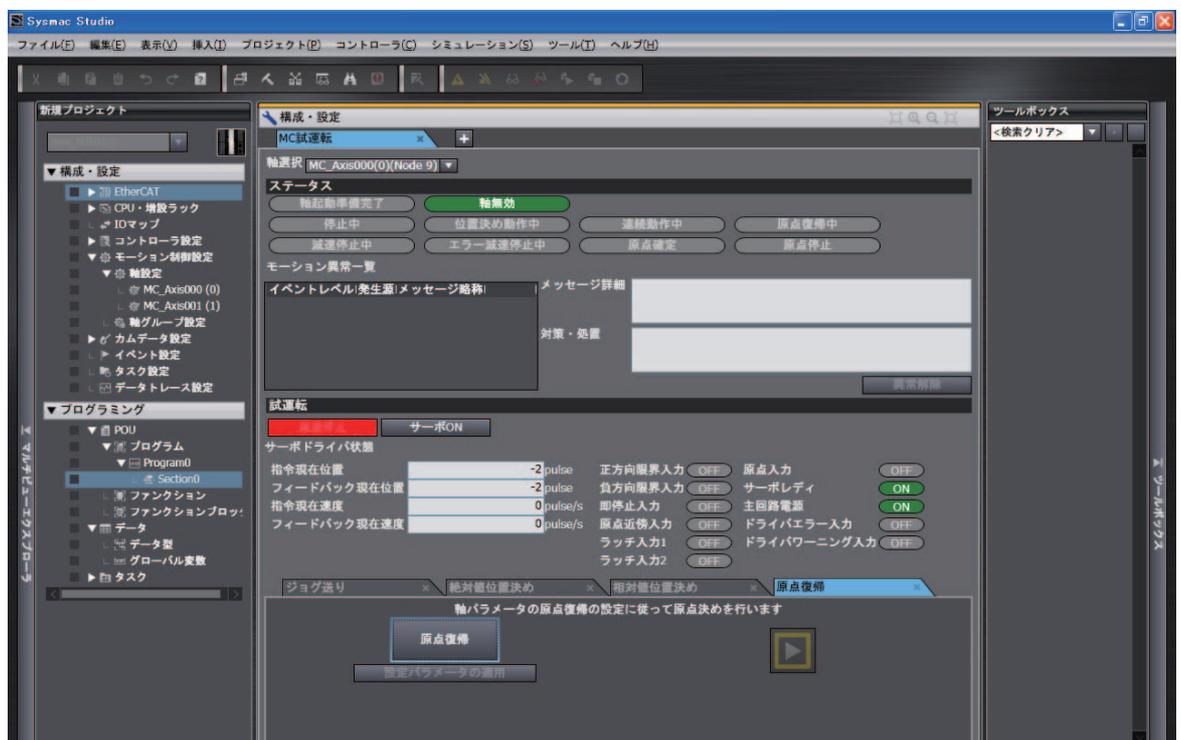
- 通过 Sysmac Studio 发送的指令执行伺服 ON、微动移动、相对值定位、绝对值定位、原点复位后，以设定的速度运行电机。  
请充分确认不会因运行电机而导致危险后再执行。
- 通过 Sysmac Studio 操作时，请在外部设置紧急停止电路，以确保可安全停止电机。计算机发生异常等时，可能无法从 Sysmac Studio 发送指令。
- 通过 Sysmac Studio 操作时，请设定 EtherCAT 通信并建立通信后再执行。
- 使用 NX 系列脉冲输出单元，需另行对电机驱动器进行通电 ON/OFF。详情请参阅 [“NX 系列位置接口单元用户手册 \(SBCE-374\)”](#)。

## 4-3-2 微动移动

- 1 通过 MC 试运行画面中的 [微动移动] 标签，选择需微动移动的轴。
- 2 按下 [伺服 ON] 按钮，设为伺服 ON 状态。
- 3 输入目标速度、加速度、减速度的值，按下 [应用] 按钮。
- 4 按下  按钮或  按钮。  
按住按钮时，向正方向或负方向移动。  
请确认是否向设定的方向移动。

## 4-3-3 原点复位

- 1 请事先通过轴参数设定画面中的 [原点复位设定] 来设定原点复位参数。
- 2 选择试运行画面中的 [原点复位] 标签。  
显示以下画面。



- 3 选择需执行原点复位的轴。
- 4 按下 [伺服 ON] 按钮，设为伺服 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与设定一致。

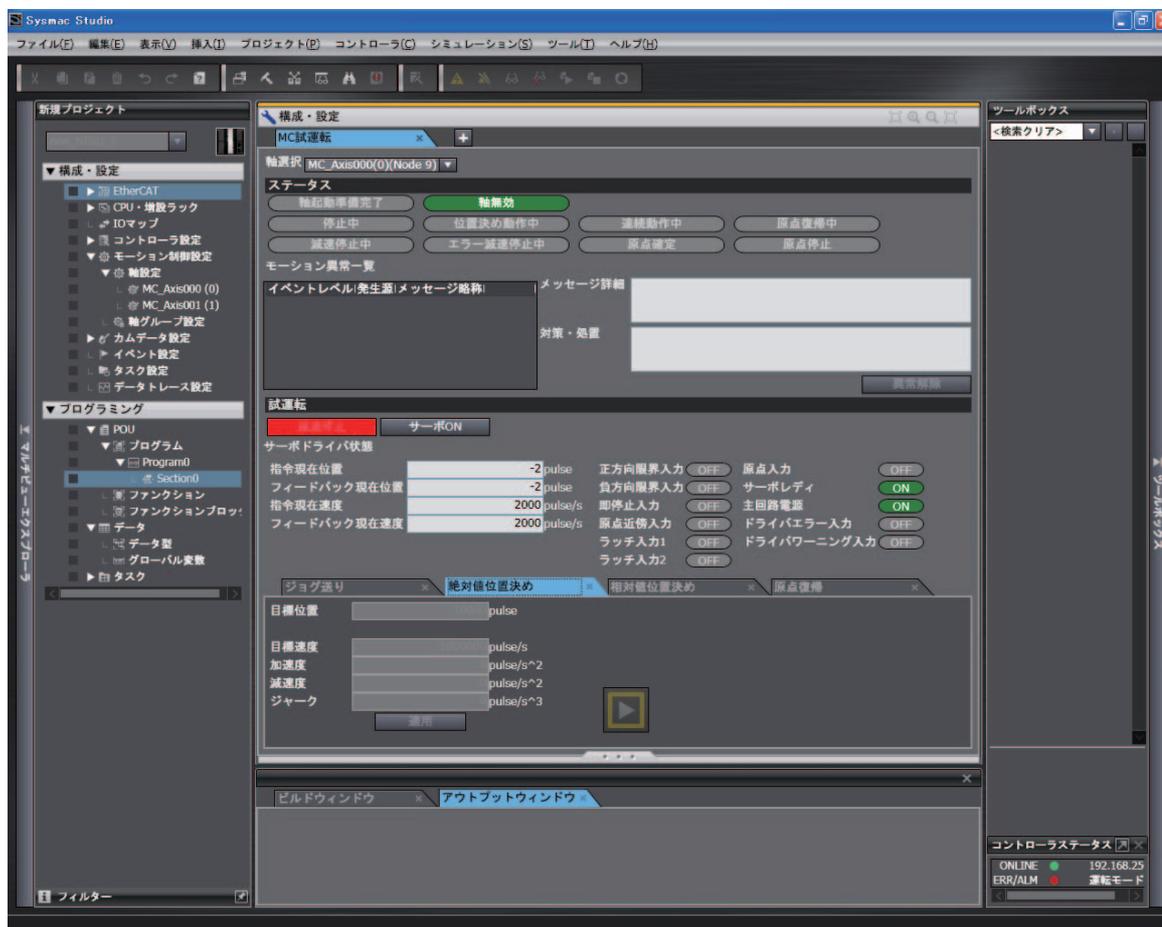


## 参考

- 按下 [原点复位设定] 按钮后会显示 [轴参数设定] 中的 [原点复位设定] 画面，请设定 [原点复位参数]。
- 事先设定 [原点复位参数]，按下 [应用设定参数] 按钮后应用该设定。

## 4-3-4 绝对值定位

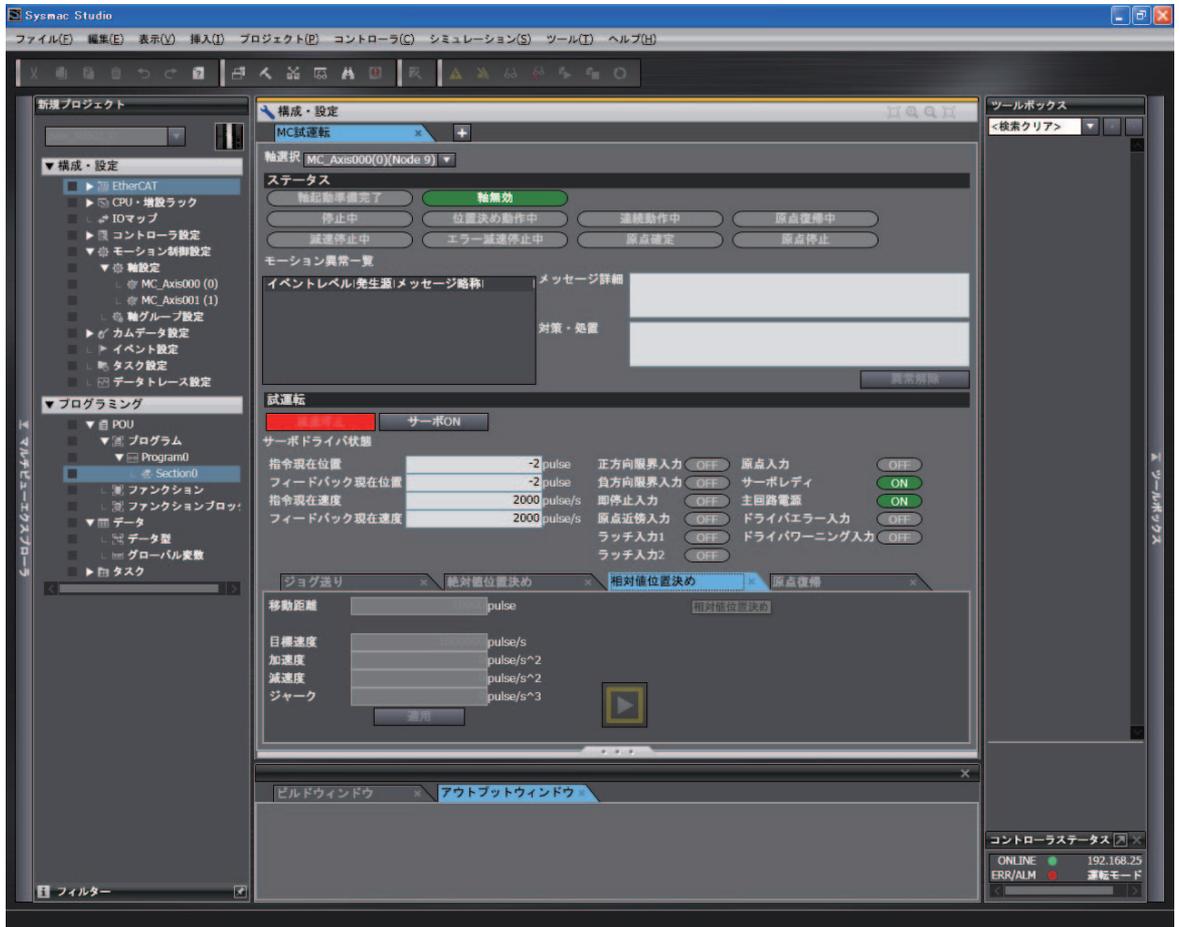
- 1 在 MC 试运行画面中选择 [ 绝对值定位 ] 标签。  
显示以下画面。



- 2 选择需执行绝对值定位的轴。
- 3 按下 [ 伺服 ON ] 按钮，设为伺服 ON 状态。
- 4 输入目标位置、目标速度、加速度、减速度、跃度值，按下 [ 应用 ] 按钮。
- 5 按下  按钮，开始执行绝对值定位。  
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 试运行画面中选择 [ 相对值定位 ] 标签。  
显示以下画面。



- 2 选择需执行相对值定位的轴。
- 3 按下 [ 伺服 ON ] 按钮，设为伺服 ON 状态。
- 4 输入目标移动距离、目标速度、加速度、减速度、跃度值，按下 [ 应用 ] 按钮。
- 5 按下  按钮，开始执行相对值定位。  
请确认移动距离是否与设定一致。



# 5

## 运动控制参数

本章对运动控制使用的轴参数和轴组参数进行说明。

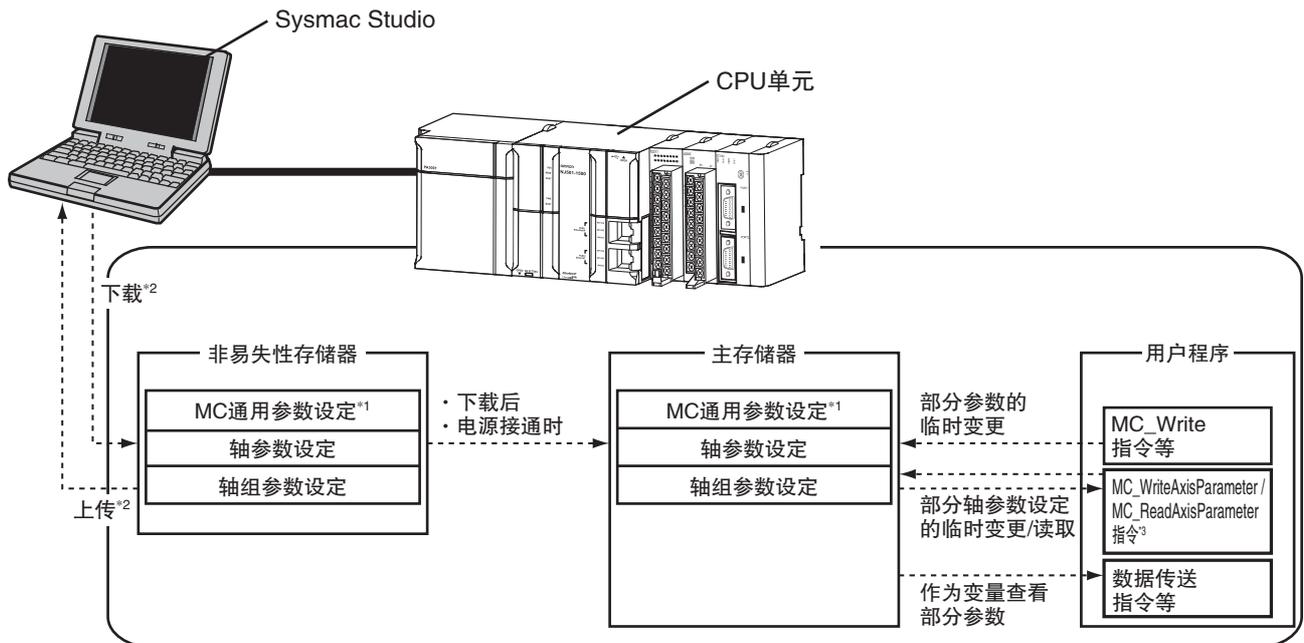
5-1 概要 .....	5-2
5-2 轴参数 .....	5-4
5-2-1 轴参数一览表 .....	5-4
5-2-2 轴基本设定 .....	5-6
5-2-3 单位转换设定 .....	5-11
5-2-4 动作设定 .....	5-17
5-2-5 扩展动作设定 .....	5-20
5-2-6 限制设定 .....	5-21
5-2-7 位置计数设定 .....	5-21
5-2-8 伺服驱动器设定 .....	5-23
5-2-9 原点复位设定 .....	5-24
5-2-10 轴参数的设定示例 .....	5-25
5-3 轴组参数 .....	5-27
5-3-1 轴组参数一览表 .....	5-27
5-3-2 轴组基本设定 .....	5-28
5-3-3 轴组动作设定 .....	5-30
5-3-4 启用轴组 .....	5-30

## 5-1 概要

NJ/NX 系列 CPU 单元的 MC 功能模块中，可通过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。

轴参数需要时常设定，但是，不使用轴组的多轴动作时，无需设定轴组参数。

这些参数称为运动控制参数设定（简称：MC 参数设定）。



\*1 当前版本的 MC 功能模块中，没有“MC 通用参数设定”设定项目。

\*2 “下载”和“上传”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)指令可组合 Ver.1.08 以上版本的 CPU 单元和 Ver.1.09 以上版本的 Sysmac Studio 使用。

### MC 参数设定的数据流

- 将通过 Sysmac Studio 设定的 MC 参数设定下载到 CPU 单元中时，保存到 CPU 单元内置的非易失性存储器中。将 MC 参数设定上传到 Sysmac Studio 中时，非易失性存储器中保存的设定被上传。
- 非易失性存储器中保存的设定在下载后或接通电源时，将反映到主存储器中。
- 如果设定内容正确，则 MC 功能模块以主存储器的设定内容执行控制。
- 部分参数作为运动控制系统变量，可查看设定内容。
- 无论 CPU 单元的模式和 MC 功能模块的状态如何，均可执行 MC 参数设定的上传和下载。
- 开始下载时，动作中的轴立即停止。



### 版本相关信息

组合 Ver.1.13 以上版本的 CPU 单元和 Ver.1.17 以上版本的 Sysmac Studio，即使在下载期间仍可持续向 I/O 设备发送指令。

Ver.1.12 以下版本的 CPU 单元，则在下载期间停止向 I/O 设备发送指令。

开始下载时停止 / 持续向 I/O 设备发送指令的设定，请参阅“NJ/NX 系列 CPU 单元用户手册 软件篇 (SBCA-359)”。

#### 停止向 I/O 设备发送指令

下载期间，伺服 OFF、轴状态为“Disable(轴无效)”。

#### 持续向 I/O 设备发送指令

下载期间，将持续下载前通过 MC\_Power 指令实现的伺服 ON 状态、及通过 MC\_SetTorqueLimit 指令进行的输出转矩限制。

即使通过下载更新用户程序、删除 MC\_Power 指令及 MC\_SetTorqueLimit 指令，仍将持续伺服 ON 状态及转矩限制。

## 通过 MC 参数设定的指令改写

- 利用运动控制指令的 MC\_Write(写入 MC 设定) 指令、MC\_ChangeAxesInGroup(写入轴组构成轴) 指令、以及 MC\_WriteAxisParameter(写入轴参数) 指令等，可在执行用户程序的过程中对主存储器中的部分 MC 参数设定内容进行变更。
- 如果指定的设定值超出范围，则指令的输出变量“Error(错误)”变为 TRUE，MC 参数设定不变。
- 变更后的 MC 参数设定内容生效的时间为下列任一时间。
  - a) 在轴或轴组停止状态下，启动轴指令、轴组指令时。
  - b) 将指令的缓存模式选择设为中断，多重启动时。
- 关于 MC\_Write(写入 MC 设定) 指令、MC\_ChangeAxesInGroup(写入轴组构成轴) 指令、以及 MC\_WriteAxisParameter(写入轴参数) 指令等的详情，请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)”。



### 使用注意事项

- 通过 MC 参数设定的 MC\_Write(写入 MC 设定) 指令等进行的变更将保存到 CPU 单元内的主存储器中。不会保存到 CPU 单元内置的非易失性存储器中。因此，必须重新接通电源或从 Sysmac Studio 下载，才能恢复为非易失性存储器的参数设定内容。此外，无法通过 Sysmac Studio 上传这些数据。必须保存到非易失性存储器中时，请通过 Sysmac Studio 变更参数设定，下载到 CPU 单元中。
- 要在下载后也保持 MC\_Power(可运行) 指令及 MC\_SetTorqueLimit(转矩限制) 指令，请将指令输入接点的保持属性设为保持。
- 可变更 MC 参数设定的指令如下所示。
  - MC\_Write(写入 MC 设定) 指令
  - MC\_ChangeAxesInGroup(写入轴组构成轴) 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，请参阅 □ “NX 系列 位置接口单元 用户手册 (SBCE-374)”。

## 5-2 轴参数

关于通过 MC 功能模块控制的轴，轴参数执行最高速度、跃度、原点复位等轴动作的设定。  
轴参数对应各型号的最大控制轴数。最大控制轴数因型号而异。详情请参阅 □ “1-4-2 性能规格 (P.1-7)”。

各轴参数设定相同，下面对单轴进行说明。

### 5-2-1 轴参数一览表

使用 Sysmac Studio，按轴分别设定轴参数。

分类	参数名称	临时变更 *1		变量读取 *2	参考页
		可否	使用指令		
轴基本设定	轴号			可	P.5-6
	运动控制 *3			可 *4	
	轴使用	可 *5	MC_ChangeAxisUse	可	
	轴种类			可	
	控制功能 *6				
	输入 / 输出设备			可	
单位转换设定	显示单位	可 *7	MC_WriteAxisParameter *8	可	P.5-11
	电机转 1 圈的脉冲数			可	
	电机转 1 圈的移动量			可	
	减速机使用 *9				
	工件侧转 1 圈的移动量 *9				
	工件侧齿轮比 *9				
	电机侧齿轮比 *9				
动作设定	最高速度	可 *7	MC_WriteAxisParameter *8		P.5-17
	启动速度 *10				
	微动最高速度				
	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 *8		
	加速度警告值				
	减速度警告值				
	正方向转矩警告值 *11				
	负方向转矩警告值 *11				
	到位宽度	可 *12			
	到位检查时间	可			
	反馈速度滤波时间常数	可 *7	MC_WriteAxisParameter *8		
原点位置范围					
扩展动作设定	立即停止输入停止方法	可 *7	MC_WriteAxisParameter *8		P.5-20
	极限输入停止方法				
	驱动器错误复位监视时间				
	正方向转矩限制上限值				
	负方向转矩限制上限值				
	立即停止输入逻辑反转 *10				
	正方向极限输入逻辑反转 *10				

分类	参数名称	临时变更 *1		变量读取 *2	参考页	
		可否	使用指令			
扩展动作设定	负方向极限输入逻辑反转 *10	可 *7	MC_WriteAxisParameter *8		P.5-20	
	近原点输入逻辑反转 *10					
限制设定	软件限制功能	可	MC_Write MC_WriteAxisParameter *8		P.5-21	
	正方向软件限制					
	负方向软件限制					
	位置偏差超过值					
	位置偏差警告值					
位置计数设定	计数模式	可 *7	MC_WriteAxisParameter *8		P.5-21	
	环计数器上限设定值					可 *4
	环计数器下限设定值					可 *4
	编码器种类					
伺服驱动器设定	环计数器上限设定值				P.5-23	
	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 *7	MC_WriteAxisParameter *8		P.5-24	
	原点输入信号					
	原点复位开始方向					
	原点检测方向					
	正方向极限输入时动作					
	负方向极限输入时动作					
	原点复位速度					
	原点复位接近速度					
	原点复位加速度					
	原点复位减速度					
	原点复位跃度					
	原点输入屏蔽距离					
	原点位置偏置					
	原点复位接触时间					
	原点复位补偿值					
原点复位补偿速度						

\*1. 表示可否通过指令临时变更。

\*2. 表示可否从用户程序读取运动控制系统变量。

\*3. NX701 系列 CPU 单元时设定。

\*4. Ver.1.10 以上的 CPU 单元和 Ver.1.12 以上的 Sysmac Studio 组合时可使用。

\*5. Ver1.04 以上的 CPU 单元和 Ver1.05 以上的 Sysmac Studio 组合时可临时变更。

\*6 NX1P2 CPU 单元时设定。

\*7. Ver1.08 以上的 CPU 单元和 Ver1.09 以上的 Sysmac Studio 组合时可临时变更。

\*8. MC\_WriteAxisParameter 指令在 Ver.1.08 以上的 CPU 单元和 Ver.1.09 以上的 Sysmac Studio 组合时可使用。

可通过 MC\_WriteAxisParameter 指令临时变更的参数可通过 MC\_ReadAxisParameter 指令读取。

\*9. Ver.1.11 以上的 CPU 单元和 Ver.1.15 以上的 Sysmac Studio 组合时可使用。

\*10.Ver.1.05 以上的 CPU 单元和 Ver.1.06 以上的 Sysmac Studio 组合时可使用。

\*11.仅转矩控制时有效。

\*12.通过 MC\_Write 指令可在 Ver.1.01 以上的 CPU 单元和 Ver.1.02 以上的 Sysmac Studio 组合时进行临时变更。

通过 MC\_WriteAxisParameter 指令可在 Ver.1.08 以上的 CPU 单元和 Ver.1.09 以上的 Sysmac Studio 组合时进行临时变更。

关于轴参数的设定方法，请参阅 □ “3-2 轴的设定步骤 (P.3-9)”。

关于 MC\_Write(写入 MC 设定)指令的详情，请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)”。

关于运动控制系统变量，请参阅 □ “6-6 运动控制系统变量 (P.6-16)”。

## 5-2-2 轴基本设定

设定使用 / 不使用轴、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]”的编号为此处设定的编号。	-	-
运动控制 *1	分配至原始恒定周期任务或固定周期任务 (执行优先级 5) 中的某一任务。 1: 原始恒定周期任务 2: 固定周期任务 (执行优先级 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接线。 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
控制功能 *4	选择要控制轴的功能。*5 0: 全部 1: 仅单轴位置控制	0, 1	0
输入设备 / 输出设备	选择轴中分配的 EtherCAT 从站设备的节点地址 *6。轴种类选择为 [ 虚拟轴 ] 时无法选择。	0 ~ 65535	-

\*1. NX701 CPU 单元时设定。

\*2. 通过未使用轴和未创建轴执行运动控制指令时，Busy(执行中)变为 TRUE。

将 Execute(启动)或 Enable(有效)设为 FALSE，则 Busy(执行中)也变为 FALSE。

设为“未使用轴”时，使轴构成不同的装置共用 1 个用户程序，因此无需删除未使用轴的程序。

\*3. Ver.1.04 以上的 CPU 单元和 Ver.1.05 以上的 Sysmac Studio 组合时，在 Sysmac Studio 的设定画面中，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)]。设为 [ 未使用轴 (可切换为使用轴)] 时，可设定轴参数，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，请参阅  “9-8-10 轴使用变更功能 (P.9-76)”。

\*4. NX1P2 CPU 单元时设定。

\*5. 用作运动控制轴时，选择“全部”。用作单轴位置控制轴时，请选择“仅单轴位置控制”。

\*6. 对于 NX 系列位置接口单元，选择 EtherCAT 耦合器单元的节点地址及位置接口单元的 NX 单元号。



## 使用注意事项

## 使用绝对值编码器时

[ 绝对值编码器原点位置偏置 ] 与轴号建立关联，按轴分别保存到电池备份存储器中。变更轴号时，保存的偏置量将消失。变更轴号时，请再次执行原点复位。

## 轴号

可设定的最大轴号为最大控制轴数。可切换为使用轴的最大实轴数为最大使用实轴数。

规格项目	NX701-17 □□	NX701-16 □□
可设定的轴号	0 ~ 255	0 ~ 127
使用实轴最大数量	256 轴	128 轴

规格项目	NX1P2-11 □□□□	NX1P2-10 □□□□	NX1P2-90 □□□□
可设定的轴号	0 ~ 11	0 ~ 9	0 ~ 3
使用实轴最大数量	8 轴	6 轴	4 轴
使用运动控制伺服轴	4 轴	2 轴	-
使用单轴位置控制伺服轴	4 轴	4 轴	4 轴

规格项目	NJ501- □ 5 □□	NJ501- □ 4 □□	NJ501- □ 3 □□
可设定的轴号	0 ~ 63	0 ~ 31	0 ~ 15
使用实轴最大数量	64 轴	32 轴	16 轴

规格项目	NJ301-12 □□	NJ301-11 □□	NJ301-10 □□
可设定的轴号	0 ~ 14*1	0 ~ 14*2	0 ~ 5
使用实轴最大数量	8 轴	128 轴	2 轴

\*1. Ver.1.05 以下版本的 CPU 单元的轴号为“0 ~ 7”。

\*2. Ver.1.05 以下版本的 CPU 单元的轴号为“0 ~ 3”。

## 运动控制

对于 NX701 CPU 单元，将使用的轴分配至原始恒定周期任务或固定周期任务 (执行优先级 5) 中的某一任务。



### 参考

NX1P2 CPU 单元及 NJ 系列 CPU 单元只有原始恒定周期任务，因此没有“运动控制”设定。

## 轴种类

利用轴种类可选择的轴种类的内容如下表所述。

轴种类	内容
伺服轴	使用 EtherCAT 从站伺服驱动器和 NX 系列 位置接口单元 *1 的轴。分配为实际伺服驱动器等加以使用。 将 1 台伺服电机作为 1 根轴使用。 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给 1 轴。
虚拟伺服轴	MC 功能模块内的虚拟轴。不使用实际伺服驱动器。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和 NX 系列 位置接口单元 *1 的轴。分配为实际编码器输入终端加以使用。 1 台编码器输入终端有 2 个计数器时，将各个计数器作为 1 根轴使用。
虚拟编码器轴	虚拟执行编码器动作的轴。没有编码器实物时，作为编码器轴的替代品临时使用。*2

\*1. 控制对象设备请参阅 □□ “1-4-3 功能规格 (P.1-10)”。

\*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。在与该指令不对应的版本的 MC 功能模块中，不能作为编码器轴的替代品使用。

### ● 虚拟伺服轴

虚拟伺服轴不存在物理性编码器和外部输入输出信号。

因此，虚拟伺服轴与实际伺服轴存在如下不同。

- 作为常时伺服 ON 状态进行处理。
- 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
- 无法使用外部输入信号。
- 通过 MC\_Home(原点复位) 指令或 MC\_HomeWithParameter(参数指定原点复位) 指令执行原点复位时，作为 [原点预设] 进行处理。与轴参数的原点复位动作的设定无关。
- 需要通过使用锁定功能的运动控制指令，将触发输入条件的模式设定为控制器模式。即使设定为驱动器模式也不会发生异常，但是，由于不发生锁定触发，因此无法完成指令的执行。  
使用锁定功能的指令包括，MC\_TouchProbe(启用外部锁定) 指令、MC\_MoveFeed(中断标准定位) 指令、以及 MC\_MoveLink(梯形模式凸轮) 指令等。
- 由于不存在即停输入和极限输入信号，因此，不会发生即停输入、正方向 / 负方向极限输入检测的异常。

### ● 编码器轴 / 虚拟编码器轴

编码器轴 / 虚拟编码器轴与伺服轴 / 虚拟伺服轴存在如下不同。

- 没有指令位置。仅有反馈位置。
- 无法使用动作类的运动控制指令。



#### 使用注意事项

单轴位置控制轴的轴种类无法指定为编码器轴 / 虚拟编码器轴。

\*1 由于在 MC 功能模块内部是以双精度实数型进行处理，因此可能包括运算误差。

## 控制功能

选择要控制轴的功能。

仅使用 NX1P2 CPU 单元时可选择。

控制功能	内容
全部	设定为可使用所有功能的轴的“运动控制轴”。
仅单轴位置控制	设定为仅可使用单轴位置控制的“单轴位置控制轴”。

通过选择控制功能，可使用的轴种类如下所示。

轴种类	控制功能	
	[全部]	[仅单轴位置控制]
伺服轴	<input type="radio"/>	<input type="radio"/>
虚拟伺服轴	<input type="radio"/>	<input type="radio"/>
编码器轴	<input type="radio"/>	<input checked="" type="checkbox"/>
虚拟编码器轴	<input type="radio"/>	<input checked="" type="checkbox"/>

另外，可动作的功能如下所示。

运动控制功能	控制功能	
	[全部]	[仅单轴位置控制]
单轴位置控制	<input type="radio"/>	<input type="radio"/>
单轴同步控制	<input type="radio"/>	<input checked="" type="checkbox"/>
单轴速度控制	<input type="radio"/>	<input type="radio"/> *1
单轴转矩控制	<input type="radio"/>	<input checked="" type="checkbox"/>
多轴协调控制	<input type="radio"/>	<input checked="" type="checkbox"/>

\*1. 仅可使用 MC\_MoveVelocity(速度控制) 指令。

运动控制功能的详情请参阅“第9章 运动控制功能”。



### 参考

使用 NX1P2-90 □□□□时，请选择 [仅单轴位置控制]。

## 输入 / 输出设备

轴种类为伺服轴或编码器轴时，选择轴中分配的 EtherCAT 设备的节点地址。对于 NX 系列位置接口单元，选择 EtherCAT 耦合器单元的节点地址及位置接口单元的 NX 单元号。轴种类选择为 [ 虚拟轴 ] 时无法选择。



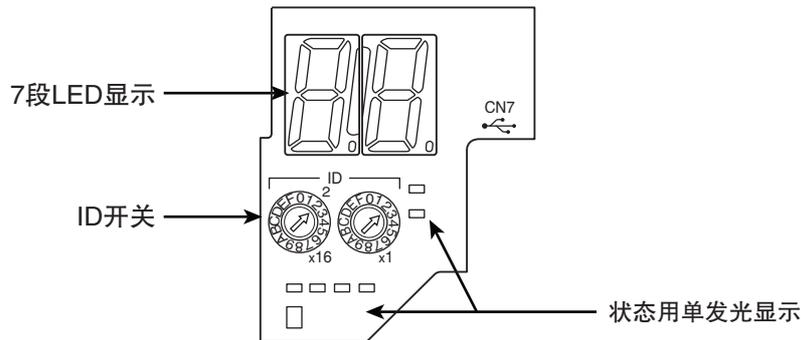
### 使用注意事项

- 欧姆龙制 1S 系列及 G5 系列伺服驱动器可以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。将该节点地址开关设为“00”时，节点地址遵从 Sysmac Studio 的 EtherCAT 编辑的设定值。  
相连的所有驱动器将节点地址开关设为“00”时，即使变更驱动器的连接位置也不会发生异常。对装置的各控制对象分配特定的伺服驱动器时，驱动器的节点地址请通过节点地址开关加以设定。
- 接通伺服驱动器控制电源时，仅读取一次伺服驱动器节点地址开关的设定值。  
接通电源后即使变更也不会反映到控制中，将在下次接通电源时生效。  
接通电源后请勿变更节点地址开关的数值。
- 节点地址重复时将发生异常。



### 参考

- 作为 EtherCAT 设备节点地址的设定示例，欧姆龙制 1S 系列伺服驱动器 EtherCAT 通信内置型如下所示。



- 使用伺服驱动器显示部的旋转开关，设定 EtherCAT 的节点地址。

旋转开关的设定值	节点地址的设定值	
	欧姆龙制从站	其他公司制从站 <sup>*1</sup>
00	Sysmac Studio 的设定值 (1 ~ 512) <sup>*2</sup>	Sysmac Studio 的设定值 (1 ~ 512) <sup>*2</sup>
01 ~ FF	节点地址开关的设定值	

\*1. 其他公司制从站不管有无从站侧的设定，Sysmac Studio 的设定值均生效。

\*2. NJ 系列 CPU 单元时为“1 ~ 192”。

## 5-2-3 单位转换设定

设定位置的单位系。

参数名称	功能	设定范围	初始值
表示单位	设定指令位置的单位。 pulse mm μm nm degree inch	0 ~ 5	0
电机转 1 圈的脉冲数 *1	根据编码器分辨率, 设定指令位置的电机转 1 圈的脉冲数。*2 根据电子齿轮比将指令值转换为脉冲量。	1 ~ 4294967295	10000
电机转 1 圈的移动量 *3	设定指令位置电机转 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 转 1 圈的移动量 *4*6*7	设定工件侧转 1 圈时的移动量。	双精度实数型正数	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. 是电子齿轮比 (单位转换的表达式) 的分子。

\*2. 例如, 编码器分辨率为 10000 脉冲 / 圈时, 设为 10000。

\*3. 将减速机使用设定为“不使用”时的电子齿轮比 (单位转换的表达式) 的分母。  
将减速机使用设定为“使用”时无效。

\*4. Ver.1.11 以上的 CPU 单元和 Ver.1.15 以上的 Sysmac Studio 组合时可使用。

\*5. 将减速机使用设定为“使用”时, 单位转换设定的设定有条件。条件请参阅  “减速机使用的条件(P.5-12)”。

\*6. 将减速机使用设定为“使用”时有效。

\*7. 仅计数模式为线性模式时可设定。为旋转模式时, 根据环计数器上限设定值及环计数器下限设定值计算得出。



#### 使用注意事项

- 在旋转模式下使用计数模式时, 请将减速机使用设定为“使用”。  
将减速机使用设定为“不使用”时, 由于将环计数器的 1 圈转换为脉冲单位时的运算误差, 环计数器每 1 圈的脉冲数可能无法变为预期的整数。这种情况会导致位置偏移。
- 除显示单位的转换外, 变更单位转换设定时, 物理机械位置和 MC 功能模块的指令当前位置之间的关系会发生偏差。因此, 变更单位转换设定时, 请执行原点复位指令, 再次确定原点。

在伺服驱动器、编码器输入终端和 MC 功能模块之间，原则上使用脉冲单位的位置。为了轻松把握实际动作，运动控制指令使用的位置使用 mm、degree 等表示单位。



可根据表示单位和电子齿轮 (单位转换的表达式) 的设定，将脉冲单位的表示转换为 mm、degree。



### 参考

对于虚拟伺服轴，以脉冲为单位转换指令当前值，并将该值再次按表示单位转换后的值作为反馈当前值。



## 表示单位

表示单位为 Sysmac Studio 中表示单位的设定。显示位置的表示单位。下面对可设定的单位进行说明。

可设定单位	说明
pulse	以脉冲单位表示时使用。
mm	直线运动较长距离时使用。
$\mu\text{m}$	高精度直线运动时使用。
nm	比 $\mu\text{m}$ 更高精度的直线运动时使用。
degree	用于转台等的旋转轴。
inch	直线运动等时使用。

## 减速机使用的条件

将减速机使用设定为“使用”时，单位转换设定的设定需要满足以下条件。

- 工件侧转 1 圈的移动量  $\times$  工件侧齿轮比  $\div$  电机侧齿轮比的值应大于 0.000000001、小于 4294967295。

计数模式为旋转模式时，还需要满足下列条件。

- 电机转 1 圈的脉冲数  $\times$  电机侧齿轮比应小于 1099511627775。

但是，这些条件的工件侧齿轮比和电机侧齿轮比的值是作为除以两者的最大公约数后的商进行计算的。因此，即使按设定的值计算不满足条件，实际也有可能满足条件。

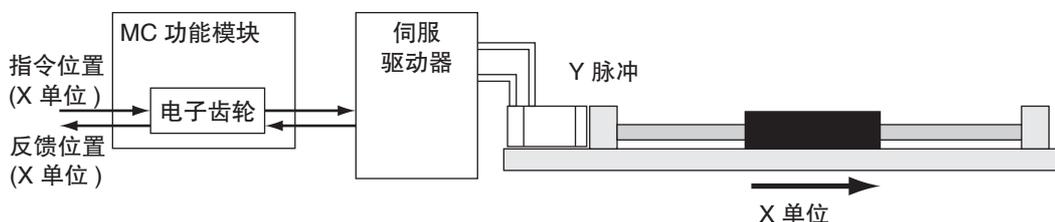
### ● 例

工件侧齿轮比为“4”、电机侧齿轮比为“6”时，分别将工件侧齿轮比视为“2”、电机侧齿轮比视为“3”进行计算。

## 电子齿轮比 (单位转换的表达式)

MC 功能模块中, 为了设定表示单位和脉冲单位的关系, 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

### ● 将减速机使用设定为“不使用”时

将减速机使用设定为“不使用”时, 电子齿轮比通过以下表达式表示。

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

\*1 采用编码器轴时, 为编码器转 1 圈的脉冲数。

\*2 采用编码器轴时, 为编码器转 1 圈的移动量。



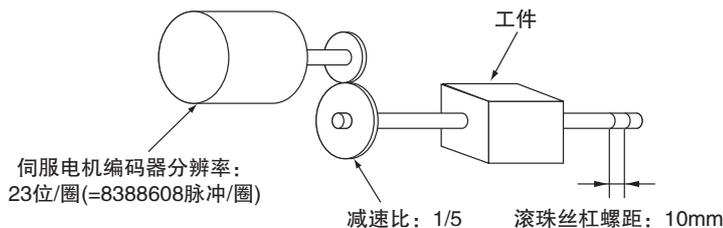
### 使用注意事项

MC 功能模块、运动控制指令使用的位置, 使用电子齿轮单位转换后的值。

通过运动控制指令将目标位置指定为 LREAL 型。但是, 经电子齿轮转换为脉冲单位后的指令位置超过 40 位时, 指令将发生异常。

例如, 使用电机带欧姆龙制 1S 系列的 23 位绝对值编码器。

下面对减速机的减速比为 1/5, 滚珠丝杠转 1 圈时, 工件移动 10mm 的机械进行说明。



表示单位为 mm。在电机转 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, 伺服电机转 5 圈时, 滚珠丝杠转动 1 圈。此时, 工件移动 2mm (= 10mm × 1/5), 因此, 将电机转 1 圈的移动量设为“2”。

设定项目	设定内容
表示单位	mm
电机转 1 圈的脉冲数	8388608
电机转 1 圈的移动量	2

根据该设定, 用户程序的位置指令单位为 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)”设为“100.5”。



## 参考

与设定示例相同的机械使用减速比为 1/9 的减速机时

伺服电机转 1 圈时的工件移动量为  $10\text{mm} \times 1/9 = 1.111 \dots \text{mm}$ ，除不尽。

遇到这样除不尽的数字时，请分别设定对 [ 电机转 1 圈的脉冲数 ] 和 [ 电机转 1 圈的移动量 ] 乘以相同系数后所得的数值。

减速比为 1/9 时，设系数为 “9”。

- [ 电机转 1 圈的脉冲数 ] : 75497472 (= 8388608 × 9)
- [ 电机转 1 圈的移动量 ] : 10 (= 10 × 1/9 × 9)

### ● 将减速机使用设定为 “使用”、计数模式为线性模式时

将减速机使用设定为 “使用” 且计数模式为线性模式时，电子齿轮比通过以下表达式表示。

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



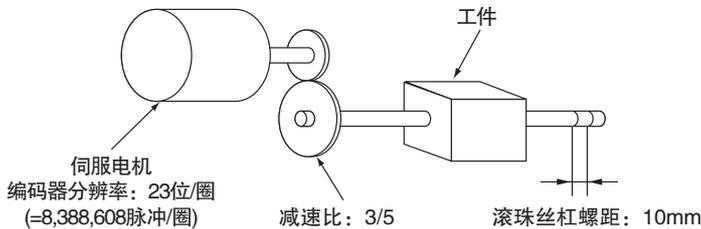
## 使用注意事项

MC 功能模块、运动控制指令使用的位置，使用电子齿轮单位转换后的值。

通过运动控制指令将目标位置指定为 LREAL 型。但是，经电子齿轮转换为脉冲单位后的指令位置超过 40 位时，指令将发生异常。

以使用欧姆龙制伺服电机 1S 系列带 23 位绝对值编码器伺服电机为例进行说明。

机械系统中，减速机的减速比为 3/5，滚珠丝杠转 1 圈工件移动 10mm。



表示单位为 mm。在电机转 1 圈的脉冲数中，设定伺服电机的编码器分辨率。

在工件转 1 圈的移动量中，设定滚珠丝杠螺距 10mm。

由于使用减速比为 3/5 的减速机，因此，伺服电机转 5 圈时，滚珠丝杠转 3 圈。此时减速比的设定如下：工件侧齿轮比设为 “3”，电机侧齿轮比设为 “5”。

设定项目	设定内容
显示单位	mm
电机转 1 圈的脉冲数	8,388,608
工件转 1 圈的移动量	10
工件侧齿轮比	3
电机侧齿轮比	5

根据该设定，用户程序的位置指令单位为 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 )” 设为 “100.5”。

### ● 将减速机使用设定为“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，电子齿轮比通过以下表达式表示。

为旋转模式时，工件转 1 圈的移动量根据“| 环计数器上限值 - 环计数器下限值 |”自动设定。

$$\begin{aligned} \text{电子齿轮比} &= \frac{\text{电机转 1 圈的脉冲数} \times \text{电机侧齿轮比}}{\text{工件转 1 圈的移动量} \times \text{工件侧齿轮比}} \\ &= \frac{\text{电机转 1 圈的脉冲数} \times \text{电机侧齿轮比}}{|\text{环计数器上限值} - \text{环计数器下限值}| \times \text{工件侧齿轮比}} \end{aligned}$$



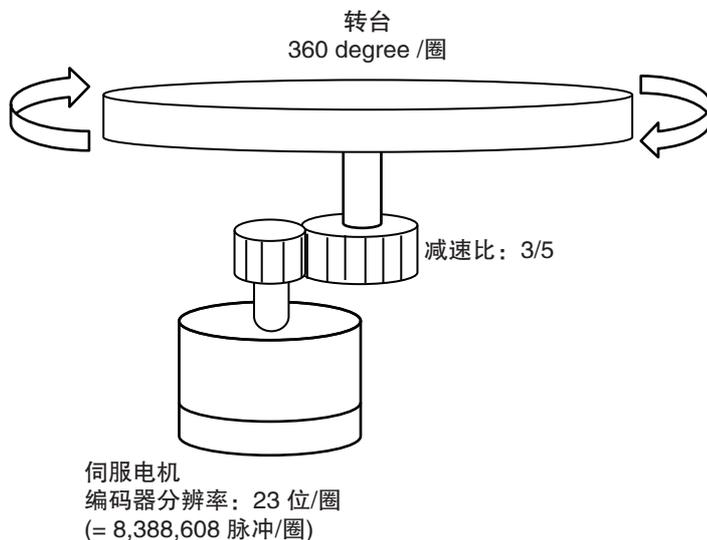
#### 使用注意事项

MC 功能模块、运动控制指令使用的位置，使用电子齿轮单位转换后的值。

通过运动控制指令将目标位置指定为 LREAL 型。但是，经电子齿轮转换为脉冲单位后的指令位置超过 40 位时，指令将发生异常。

例 1 以使用欧姆龙制伺服电机 1S 系列带 23 位绝对值编码器伺服电机为例进行说明。

机械系统中，减速机的减速比为 3/5，转台转 1 圈工件移动 360degree。



表示单位为 degree。在电机转 1 圈的脉冲数中，设定伺服电机的编码器分辨率。

工件转 1 圈的移动量根据“| 环计数器上限值 - 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，伺服电机转 5 圈时，转台 (工件) 转 3 圈。此时减速比的设定如下：工件侧齿轮比设为“3”，电机侧齿轮比设为“5”。

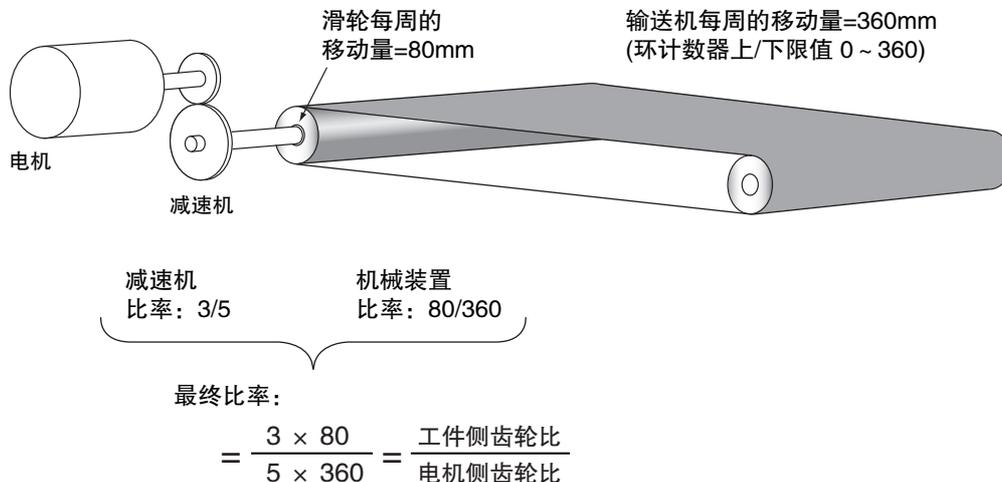
设定项目	设定内容
显示单位	degree
电机转 1 圈的脉冲数	8,388,608
工件转 1 圈的移动量	环计数器上限值 - 环计数器下限值
工件侧齿轮比	3
电机侧齿轮比	5

根据该设定，用户程序的位置指令单位为 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)”设为“100.5”。

例 2 以使用欧姆龙制伺服电机 1S 系列带 23 位绝对值编码器伺服电机为例进行说明。  
机械系统中，减速机的减速比为 3/5，滑轮转 1 圈输送机移动 80mm。输送机转 1 周为 360mm。

伺服电机  
编码器分辨率：23位/圈  
(=8,388,608脉冲/圈)



表示单位为 mm。在电机转 1 圈的脉冲数中，设定伺服电机的编码器分辨率。  
输送机转 1 圈的移动量根据“| 环计数器上限值 - 环计数器下限值 |”自动设定。

设定项目	设定内容
环计数器上限值	360
环计数器下限值	0

由于使用减速比为 3/5 的减速机，因此，伺服电机转 5 圈时，滑轮转 3 圈。并且，由于滑轮每转 1 周输送机的移动量为 80mm，输送机转 1 周为 360mm，因此减速比的设定如下：工件侧齿轮比设为“3×80”即“240”，电机侧齿轮比设为“5×360”即“1800”。

设定项目	设定内容
显示单位	mm
电机转 1 圈的脉冲数	8,388,608
工件转 1 圈的移动量	环计数器上限值 - 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 圈输送机的移动量为 80.1mm 等非整数时，请使用乘以整数系数“10”后得到的比率，计算出工件侧齿轮比和电机侧齿轮比。以上述例为基础，工件侧齿轮的比率为“3×80.1×10”即“2403”，电机侧齿轮的比率为“5×360×10”即“18000”。
- 为多级减速机时也 and 上述例 2 一样，将乘以多个比率的最终比率设定为工件侧齿轮比和电机侧齿轮比。

## 5-2-4 动作设定

根据控制装置的规格，如最高速度、最大加减速度等，对轴动作进行设定。

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。 <sup>*1</sup> 设定时请勿超出使用电机的最大转速。 (单位: 指令单位 /s)	双精度实数型的 正数 <sup>*2</sup>	40000000
启动速度 <sup>*3</sup>	设定各轴的启动速度。 请勿超过最高速度。(单位: 指令单位 /s)	双精度实数型正数	0
微动最高速度	设定各轴的微动最高速度。 <sup>*4</sup> 设定时请勿超出使用电机的最大转速。 (单位: 指令单位 /s)	双精度实数型的 正数	1000000
最大加速度	设定轴动作指令时各轴的最大加速度。 设为“0”时, 无加速度限制。 (单位: 指令单位 /s <sup>2</sup> )	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。 设为“0”时, 无减速度限制。 (单位: 指令单位 /s <sup>2</sup> )	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, 优先向目标位置停止后, 产生减速超限。指定超过最大加减速度时的动作。 0: 提高加减速速度(将合并切换为等待) <sup>*5</sup> 1: 提高加减速速度 2: 异常停止 <sup>*6</sup>	0 ~ 2	0
反转时动作	指定多重启动指令时、重启指令时、中断标准定位时的反转时动作。 <sup>*7</sup> 0: 减速停止 1: 立即停止	0 ~ 1	0
速度警告值	设定对最高速度(用于输出各轴的速度警告)的比例。 设为“0”时, 不输出速度警告。(单位: %)	0 ~ 100	0
加速度警告值	设定对最高加速度(用于输出各轴的加速度警告)的比例。 设为“0”时, 不输出加速度警告。(单位: %)	0 ~ 100	0
减速度警告值	设定对最高减速度(用于输出各轴的减速度警告)的比例。 设为“0”时, 不输出减速度警告。(单位: %)	0 ~ 100	0
正方向转矩警告值 <sup>*8</sup>	设定转矩指令值(用于输出各轴的正方向转矩警告)。 设为“0”时, 不输出正方向转矩警告。(单位: %)	0 ~ 1000	0
负方向转矩警告值 <sup>*8</sup>	设定转矩指令值(用于输出各轴的负方向转矩警告)。 设为“0”时, 不输出负方向转矩警告。(单位: %)	0 ~ 1000	0
反馈速度 滤波时间常数	以毫秒为单位, 设定反馈速度移动平均值的计算时间。 设为“0”时, 不计算移动平均值。 (单位: ms) 需要抑制轴的速度较慢时反馈当前速度的偏差时使用。	0 ~ 100	0
到位宽度 <sup>*9</sup>	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

参数名称	功能	设定范围	初始值
到位检查时间*9	以毫秒为单位，设定定位完成的检查时间。设为“0”时，仅原点复位时的原点确定时，持续定位完成的检查。非原点复位时，不作定位完成检查。(单位: ms)	0 ~ 10000	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

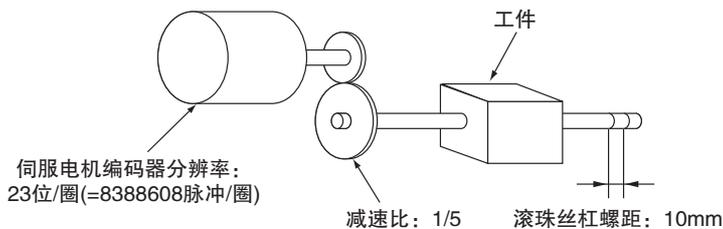
- \*1. 指定超过最高速度的速度指令值时，以最高速度发出指令。插补控制动作时也生效。请勿设定超过与轴连接的伺服电机 / 驱动器的可输出电机最高速度（最高转速）的值。请在确认所连接的伺服电机 / 驱动器的规格的基础上进行设定。
- \*2. 可设定的最大值为转换为脉冲单位的值，如下所示。  
 对于 Ver.1.02 以下版本的 CPU 单元，为 400,000,000 [pulse/s]，  
 对于 Ver.1.03 以上版本的 CPU 单元，为 500,000,000 [pulse/s]，  
 对于 Ver.1.11 以上版本的 CPU 单元，为 2,147,483,647 [pulse/s]。
- \*3. Ver.1.05 以上的 CPU 单元和 Ver.1.06 以上的 Sysmac Studio 组合时可使用。
- \*4. 指定超过微动最高速度的速度指令值时，以微动最高速度发出指令。
- \*5. Ver.1.10 以上的 CPU 单元，无法将合并切换为等待。详情请参阅 [□ “9-5-7 运动控制指令的多重启动 \(缓存模式\) \(P.9-44\)”](#)。
- \*6. Ver.1.10 以上的 CPU 单元，合并动作时不会异常停止。详情请参阅 [□ “9-5-7 运动控制指令的多重启动 \(缓存模式\) \(P.9-44\)”](#)。
- \*7. 关于反转时动作的详情，请参阅 [□ “9-5-6 运动控制指令的重启 \(P.9-39\)”](#) 和 [□ “9-5-7 运动控制指令的多重启动 \(缓存模式\) \(P.9-44\)”](#)。
- \*8. 仅转矩控制时有效。
- \*9. 到位检查在 MC 功能模块侧处理。未使用伺服驱动器侧的功能。

## 最高速度

对最高速度的设定示例进行说明。

### ● 最高速度的设定示例

下面对在与其所述的“将减速机使用设定为“不使用”时 (P.5-13)”相同的机械中，伺服电机的最大转速为 6,000r/min 时的情形进行说明。



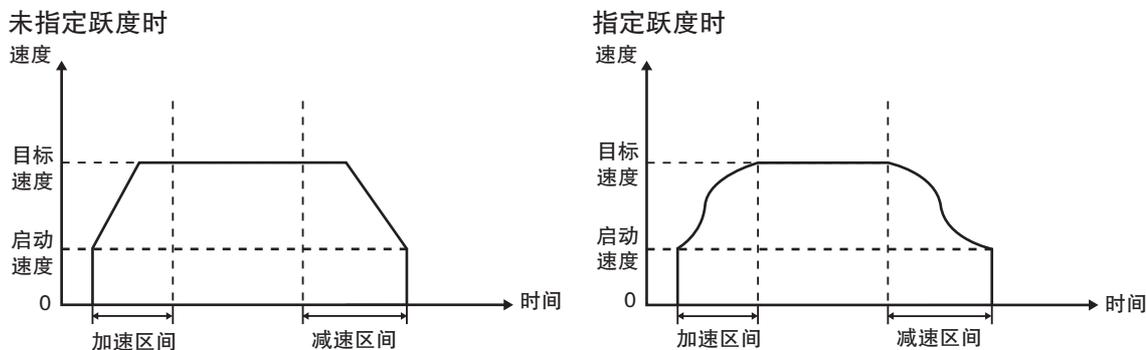
根据最大转速: 6,000r/min、减速比: 1/5、滚珠丝杠螺距: 10mm 等各条件计算可得， $6,000\text{r/min} \times 1/5 \times 10\text{mm} = 12,000\text{mm/min} = 200\text{mm/s}$ 。因此将最高速度设为“200”。  
 初始值 400,000,000 超过电机的最大转速，因此请务必变更设定值。

设定项目	设定内容
表示单位	mm
电机转 1 圈的脉冲数	8388608
电机转 1 圈的移动量	2
最高速度	200

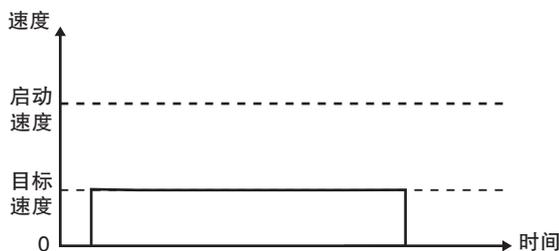
## 启动速度

使用伺服电机时，请设为“0”。

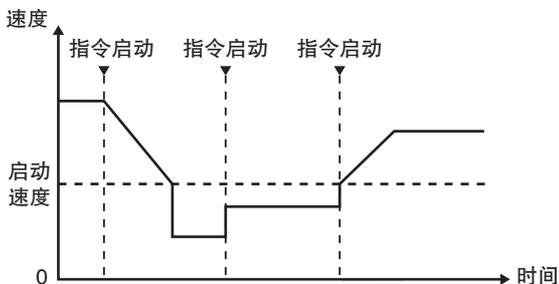
使用步进电机时，为了防止启动时失步，通常设定为最大自启动频率的 1/10 ~ 1/2 的值。会根据负荷情况变化，详情请确认步进电机的手册。



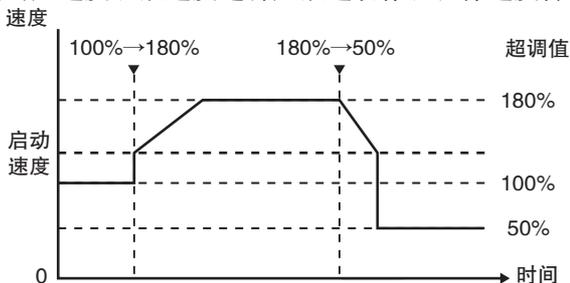
目标速度低于启动速度时，不作加减速而以目标速度动作。



动作中根据运动控制指令的重启和多重启动，在变更目标速度后也适用启动速度。目标速度超过启动速度时，以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，不作加减速而动作。



启动速度也适用于通过 MC\_SetOverride(超调值设定) 指令变更速度时。目标速度超过启动速度时，以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，不作加减速而动作。



以下功能不适用启动速度。

- 转矩控制模式
- 周期同步速度控制
- 周期同步定位
- 同步控制

但是，MC\_GearOut(齿轮动作解除)、MC\_CamOut(凸轮动作解除)适用启动速度。

- 多轴协调控制

但是，以“减速停止各轴”减速时，适用各轴的启动速度。

## 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转矩限制值等。

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的 MC 功能模块侧的停止方法。 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的 MC 功能模块侧的停止方法。 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
驱动器错误复位监视时间	设定执行驱动器错误复位时的监视时间。 (单位: ms) 超过监视时间时, 即使驱动器异常未复位, 也会结束复位处理。	1 ~ 1000	200
正方向转矩限制上限值	设定正方向转矩限制值的上限。 <sup>*1</sup> (单位: %)	0.0 ~ 1000.0	300.0
负方向转矩限制上限值	设定负方向转矩限制值的上限。 <sup>*1</sup> (单位: %)	0.0 ~ 1000.0	300.0
立即停止输入逻辑反转 <sup>*2</sup>	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 <sup>*3</sup>
正方向极限输入逻辑反转 <sup>*2</sup>	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 <sup>*3</sup>
负方向极限输入逻辑反转 <sup>*2</sup>	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 <sup>*3</sup>
原点附近输入逻辑反转 <sup>*2</sup>	设定原点附近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“正转侧转矩限制 (60E0h)”和“反转侧转矩限制 (60E1h)”时, 这些设定值将通过 EtherCAT 的过程数据通信发送。通过 MC\_SetTorqueLimit(转矩限制) 指令将转矩限制设为有效时, 指令输入变量中的指定值将被发送。

\*2. Ver.1.05 以上的 CPU 单元和 Ver.1.06 以上的 Sysmac Studio 组合时可使用。Ver.1.04 以下的 CPU 单元不进行逻辑反转。分配了 NX 系列数字输入单元等不能设定输入信号逻辑的设备时设定。欧姆龙制伺服驱动器 1S 系列等可设定输入信号逻辑的设备通常设定为“不反转”。

\*3. 分配了 NX 系列脉冲输出单元的轴的初始值为“TRUE”。

### 5-2-6 限制设定

选择位置偏差相关的限制功能和软件的限制功能。

参数名称	功能	设定范围	初始值
软件限制功能 *1	选择软件限制的功能。 0: 无效 1: 对指令位置有效, 执行减速停止 2: 对指令位置有效, 执行立即停止 3: 对反馈位置有效, 执行减速停止 4: 对反馈位置有效, 执行立即停止	0 ~ 4	0
正方向软件限制	设定正方向侧的软件限制。(单位: 指令单位)	双精度实数型的 负数、正数、0	2147483647
负方向软件限制	设定负方向侧的软件限制。(单位: 指令单位)	双精度实数型的 负数、正数、0	-2147483648
位置偏差超过值	设定超过位置偏差的检查值。 设为“0”时, 超过位置偏差的检查无效。 (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检查值。 设为“0”时, 位置偏差警告检查无效。 (单位: 指令单位)	双精度实数型的 正数、0 (位置偏差超过 值以下)	0

- \*1. 该功能仅在计数模式为线性模式、原点已确定时启动。  
关于软件限制功能的详情, 请参阅 □ “9-8-5 软件限制功能 (P.9-71)”。

### 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, 请参阅 □ “NX 系列 位置接口单元 用户手册 (SBCE-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: 线性模式 (有限长) 1: 旋转模式 (无限长)	0 ~ 1	0
环计数器 上限设定值	设定将计数模式设为旋转模式时的环计数器上 限值。 (单位: 指令单位)	双精度实数型的 负数、正数、0	2147483647
环计数器 下限设定值	设定将计数模式设为旋转模式时的环计数器下 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: 绝对值编码器 (ABS)	0 ~ 1	0

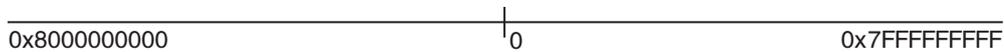
- \*1. 使用以下设备时, 请在编码器种类中设定 [1: 绝对值编码器 (ABS)]。  
使用欧姆龙制伺服电机 / 驱动器 1S 系列时  
欧姆龙制伺服电机 / 驱动器 G5 系列使用带绝对值编码器的伺服电机时  
欧姆龙制伺服电机 / 驱动器 G5 系列使用全闭环控制用绝对值型外部标尺时  
欧姆龙制伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型使用绝对值型外部标尺时
- \*2. 欧姆龙制伺服电机 / 驱动器 G5 系列使用全闭环控制用外部标尺时, 或者使用欧姆龙制伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型时, 请分别如下进行改读。  
0: 增量型外部标尺  
1: 绝对值型外部标尺

## 计数模式

是轴移动的模式。按轴分别选择指令位置的计数模式。分为轴移动范围为有限长度的线性模式和为无限长度的旋转模式。

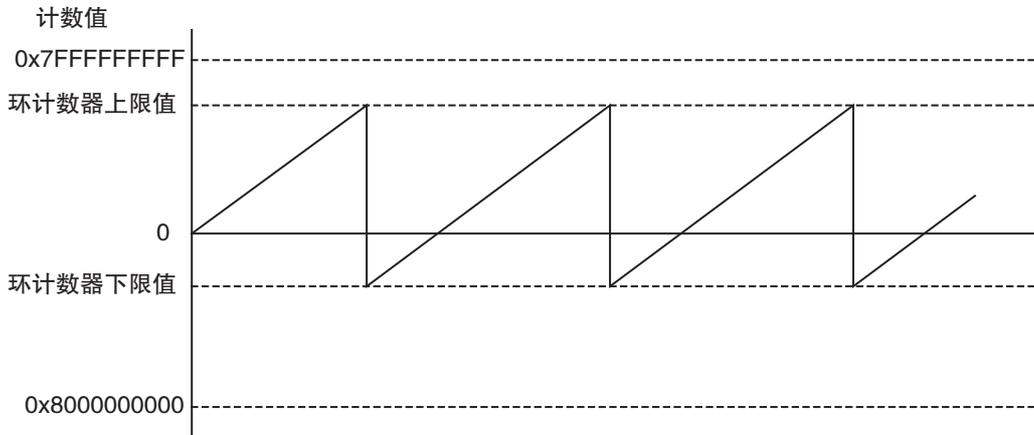
### ● 线性模式 (有限长轴)

- 是以“0”为中心的直线设定。  
在 X-Y 轴滑台等限定机械动作范围的装置中使用。
- 设定范围为脉冲单位转换后的带符号整数型 40 位 (0x8000000000 ~ 7FFFFFFF)。
- 在相对值移动或绝对值移动等指定目标位置的定位中，无法发出超范围目标位置的指令。
- 在速度控制、原点复位以及转矩控制等不指定目标位置的动作中，如果超出动作范围，则出现监控信息 (指令位置溢出 / 下溢)。指令位置的输出持续，但不更新当前位置，以上限值或下限值固定。
- 当前位置固定的状态下，在不指定目标位置的动作中沿着进入范围内的方向发出指令或停止时，可以启动。沿着远离范围的方向发出指令时，在启动指令时将发生异常。
- 溢出 / 下溢状态得不到解除，当前位置就不会更新。



### ● 旋转模式 (无限长轴)

- 是在设定范围内重复无限计数的环计数器形式的模式。  
在转台或卷轴等中使用。
- 关于环计数器的范围，通过 Sysmac Studio 设定环计数器上限值和下限值。
- 设定范围为脉冲单位转换后的带符号整数型 40 位 (0x8000000000 ~ 7FFFFFFF)。



- 由于将环计数器的 1 圈转换为脉冲单位时的运算误差，环计数器每 1 圈的脉冲数可能无法变为预期的整数。这种情况会导致位置偏移。



### 版本相关信息

使用 V1.11 以上的 CPU 单元时，请将减速机使用设定为“使用”。  
通过设定为“使用”，可消除上述位置偏差的原因。  
关于减速机使用，请参阅 □ “5-2-3 单位转换设定 (P.5-11)”

## 环计数器上限设定值和环计数器下限设定值

将计数模式设为旋转模式后生效的设定值。

设定环计数器的上限值和下限值。



### 使用注意事项

- 环计数器的上限值和下限值之间不含“0”时，如果执行 MC\_MoveZeroPosition(高速原点复位)指令，将发生异常。
- 此外，通过 MC\_MoveAbsolute(绝对值定位)指令、MC\_Move(定位)指令执行绝对值定位时，请在环计数器的范围内设定目标位置。超出范围时会发生异常。但是，在指令的输入变量 Direction(方向选择)中指定“4:无方向指定”时，超出范围也可以设定为目标位置。此时的动作表现为，将目标位置和指令当前位置的差值作为目标距离的相对值定位。
- 变更了位置计数设定后，物理机械位置与 MC 功能模块的指令当前位置会产生相对偏移。因此，变更了位置计数设定后，请执行原点复位指令，再次确定原点。

## 编码器种类

选择要反馈输入的编码器种类。

使用以下设备时，请在编码器种类中设定 [1:绝对值编码器 (ABS)]。

- 使用欧姆龙制伺服电机 / 驱动器 1S 系列时
- 欧姆龙制伺服电机 / 驱动器 G5 系列使用带绝对值编码器伺服电机时
- 欧姆龙制伺服电机 / 驱动器 G5 系列使用全闭环控制用绝对值型外部标尺时
- 欧姆龙制伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型使用绝对值型外部标尺时

### 5-2-8 伺服驱动器设定

设定相连伺服驱动器或编码器输入终端设定的数值。

参数名称	功能	设定范围	初始值*1
环计数器 上限设定值	设定伺服驱动器或编码器输入终端设定的环计数器上限值。*2	$-2^{63} \sim 2^{63}-1$ (单位:脉冲)	2147483647
环计数器 下限设定值	设定伺服驱动器或编码器输入终端设定的环计数器下限值。*2	$-2^{63} \sim 2^{63}-1$ (单位:脉冲)	-2147483648
PDS 状态控制方式*3	设定因 MC_Power 指令伺服 OFF 时 PDS 状态的切换目标。*4 0: 伺服 OFF 时 Switched on 1: 伺服 OFF 时 Ready to switch on	0 ~ 1	0

\*1. 初始值为整数型 (DINT) 的所有范围。由于是整数型 (DINT) 的所有范围，欧姆龙制 1S 系列及 G5 系列伺服驱动器可以直接使用初始值。

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

\*2. 使用欧姆龙制 GX 系列 EtherCAT 从站编码器输入终端时，必须使编码器输入终端的环计数器最大值 (对象索引 0x4003) 与环计数器上限设定值一致。并且，必须将环计数器下限设定值设为 0。

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

\*4. 使用欧姆龙制伺服驱动器 G5 系列时，如设为“1”，则无法使用伺服驱动器的伺服待机 (Switched on)。要使用伺服准备就绪 (Switched on) 时，请设为“0”。关于 PDS 状态的切换，请参阅 “A-5 PDS 状态切换 (P.A-26)”。

## 5-2-9 原点复位设定

对确定机械原点的电机动作进行设定。

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: 指定为附近避让、近原点输入 OFF 1: 指定为附近避让、近原点输入 ON 4: 指定为近原点输入 OFF 5: 指定为近原点输入 ON 8: 指定为极限输入 OFF 9: 指定为附近避让、原点输入屏蔽距离 11: 仅极限输入 12: 指定为附近避让、接触时间 13: 指定为无近原点输入、接触原点输入 14: 原点预设	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	设定启动原点复位时的开始方向。 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	设定原点复位中正方向极限输入“ON”时的停止方法。 0: 不反转 / 轴异常而停止 (遵照极限输入停止方法) 1: 反转 / 立即停止 2: 反转 / 减速停止	0 ~ 2	1
负方向极限输入时动作	设定原点复位中负方向极限输入“ON”时的停止方法。 0: 不反转 / 轴异常而停止 (遵照极限输入停止方法) 1: 反转 / 立即停止 2: 反转 / 减速停止	0 ~ 2	1
原点复位速度	设定原点复位中的速度。(单位: 指令单位 /s)	双精度实数型的正数	10000
原点复位接近速度	设定近原点输入 ON 后的速度。(单位: 指令单位 /s)	双精度实数型的正数	1000
原点复位加速度	指定原点复位时的加速度。 设为“0”时, 不加速达到原点复位速度等目标速度。 (单位: 指令单位 /s <sup>2</sup> )	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。 设为“0”时, 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 <sup>2</sup> )	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。 设定为“0”时无跃度。(单位: 指令单位 /s <sup>3</sup> )	双精度实数型的正数、0	0
原点输入屏蔽距离	对原点复位动作模式指定为附近避让、原点输入屏蔽距离时原点输入的屏蔽距离进行设定。 (单位: 指令单位)	双精度实数型的正数、0	10000
原点位置偏置	原点复位完成时, 按照设定值预设当前位置。 (单位: 指令单位)	双精度实数型的负数、正数、0	0
原点复位接触时间	以毫秒为单位, 对原点复位动作模式指定为附近避让、接触时间时的接触时间进行设定。(单位: ms)	0 ~ 10000	100
原点复位补偿值	设定原点复位的原点确定后的原点复位补偿量。 (单位: 指令单位)	双精度实数型的负数、正数、0	0
原点复位补偿速度	设定原点复位补偿时的速度。(单位: 指令单位 /s)	双精度实数型的正数	1000

\*1. 为原点复位动作的设定。详情请参阅“第8章 确定原点”。

\*2. 欧姆龙制伺服驱动器 G5 系列中, 直线电机型无法将 Z 相输入映射到 PDO 中。因此, 要使用原点复位动作的“指定为无近原点输入、接触原点输入”时, 请勿在原点输入信号中选择 Z 相输入。注: 该原点复位动作使用 PDO 中映射的 Z 相输入。

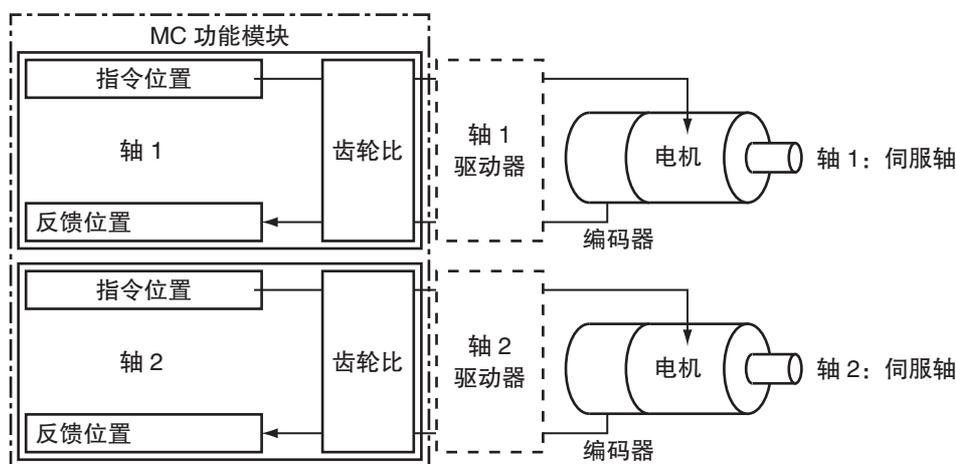
- \*3. 使用欧姆龙制伺服驱动器 1S 系列及 G5 系列时可指定。将分配给伺服驱动器的锁定功能 1 的输入用作外部原点输入。1S 系列及 G5 系列的初始状态下，将外部锁定输入 1 分配给锁定功能 1。  
 详情请参阅 □ “AC 伺服电机 / 驱动器 1S 系列 EtherCAT 通信内置型 用户手册 (SBCE-377)”、□ “AC 伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置型 用户手册 (SBCE-365)” 及 □ “AC 伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型 用户手册 (SBCE-366)”。  
 使用 NX 系列位置接口单元时，请参阅 □ “NX 系列 位置接口单元 用户手册 (SBCE-374)”。

## 5-2-10 轴参数的设定示例

下面对轴参数中间位置相关项目的设定示例进行说明。

### 单轴定位

下面以使用各个单轴对双轴进行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
表示单位	$\mu\text{m}^*$	$\mu\text{m}$
电机转 1 圈的脉冲数	1,048,576	1,048,576
电机转 1 圈的移动量	10,000	10,000
最高速度	500,000 <sup>*2</sup>	500,000 <sup>*2</sup>
微动最高速度	50,000 <sup>*3</sup>	50,000 <sup>*3</sup>
最大加速度	5,000,000 <sup>*4</sup>	5,000,000 <sup>*4</sup>
最大减速度	5,000,000 <sup>*4</sup>	5,000,000 <sup>*4</sup>
软件限制功能	对指令位置有效，执行立即停止	对指令位置有效，执行立即停止
正方向软件限制	500,000 <sup>*5</sup>	500,000 <sup>*5</sup>
负方向软件限制	0 <sup>*5</sup>	0 <sup>*5</sup>
计数模式	线性模式	线性模式

\*1. 位置的指令单位为  $1(\mu\text{m})$ 。

\*2. 最高速度为  $3,000\text{r}/\text{min} = 30\text{m}/\text{min} = 0.5\text{m}/\text{s} = 500,000\mu\text{m}/\text{s}$ 。

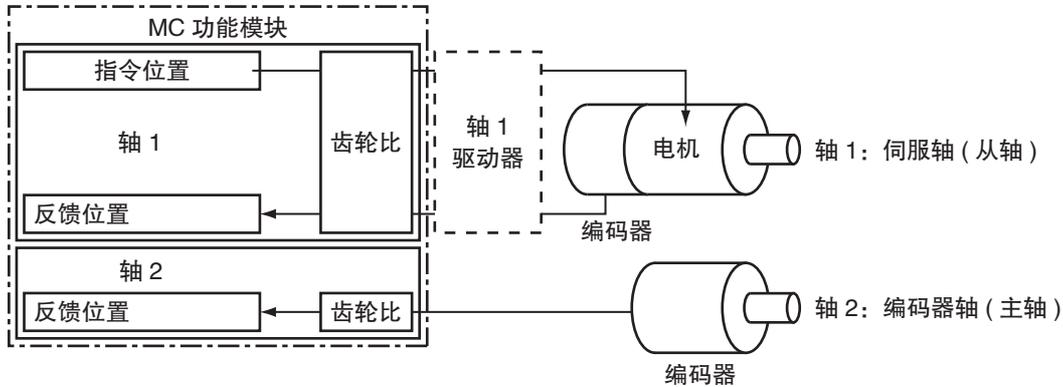
\*3. 微动最高速度为最高速度的 10% 即  $0.05\text{m}/\text{s} = 50,000\mu\text{m}/\text{s}$ 。

\*4. 最大加速度和最大减速度为  $5\text{m}/\text{s}^2$ 。达到最高速度  $3,000\text{r}/\text{min}$  的加速时间设为  $0.1\text{s}$ 。

\*5. 设定装置可动范围内侧的位置。正方向软件限制的设定值为  $50\text{cm} = 500,000\mu\text{m}$ 。

## 编码器为主轴的同步控制

下面，将安装在输送机上的轴 2(编码器)的反馈位置作为主轴，以轴 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
表示单位	$\mu\text{m}^{*1}$	$\mu\text{m}^{*1}$
电机转 1 圈的脉冲数	1,048,576	1,048,576
电机转 1 圈的移动量	10,000	10,000
最高速度	500,000 <sup>*2</sup>	-
微动最高速度	50,000 <sup>*3</sup>	-
最大加速度	5,000,000 <sup>*4</sup>	-
最大减速度	5,000,000 <sup>*4</sup>	-
软件限制功能	对指令位置有效，执行立即停止	无效
正方向软件限制	500,000 <sup>*5</sup>	-
负方向软件限制	0 <sup>*5</sup>	-
计数模式	线性模式	旋转模式
环计数器上限值	-	1,000,000 <sup>*6</sup>
环计数器下限值	-	0 <sup>*6</sup>

\*1. 位置的指令单位为  $1(\mu\text{m})$ 。

\*2. 最高速度为  $3,000\text{r}/\text{min} = 30\text{m}/\text{min} = 0.5\text{m}/\text{s} = 500,000 \mu\text{m}/\text{s}$ 。

\*3. 微动最高速度为最高速度的 10% 即  $0.05\text{m}/\text{s} = 50,000 \mu\text{m}/\text{s}$ 。

\*4. 最大加速度和最大减速度为  $5\text{m}/\text{s}^2$ 。达到最高速度  $3,000\text{r}/\text{min}$  的加速时间设为  $0.1\text{s}$ 。

\*5. 设定装置可动范围内侧的位置。

正方向软件限制的设定值为  $50\text{cm} = 500,000 \mu\text{m}$ 。

\*6. 位置的重复范围为  $0 \sim 1\text{m}(1,000,000 \mu\text{m})$ 。



### 参考

主轴按照装置构成的不同，可以选择轴种类。轴种类分为伺服轴、虚拟伺服轴、编码器轴和虚拟编码器轴 4 种。

在本例中，主轴的轴种类为编码器轴，运动控制指令的输入变量“ReferenceType(位置类型选择)”请指定反馈位置。

## 5-3 轴组参数

关于通过 MC 功能模块控制的轴组，轴组参数执行轴构成、最高插补速度、轴组停止方法之轴组动作的设定。

轴组参数对应轴组数。轴组数因型号而异。

详情请参阅 □ “1-4-2 性能规格 (P.1-7)”。

各轴组参数设定相同，下面对单组进行说明。

### 5-3-1 轴组参数一览表

使用 Sysmac Studio，按轴组分别设定轴组参数。

分类	参数名称	临时变更*1		变量读取*2	参考页
		可否	使用指令		
轴组基本设定	轴组编号			可	P.5-28
	运动控制*3			可*4	
	轴组使用			可	
	轴构成			可	
	构成轴	可*5	MC_Change AxesInGroup	可	
轴组动作设定	最高插补速度				P.5-30
	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				
	中心点补偿容许率				

\*1. 表示可否通过指令临时变更。

\*2. 表示可否从用户程序读取变量。

\*3. NX701 CPU 单元时设定。

\*4. Ver.1.10 以上的 CPU 单元和 Ver.1.12 以上的 Sysmac Studio 组合时可使用。

\*5. Ver.1.01 以上的 CPU 单元和 Ver.1.02 以上的 Sysmac Studio 组合时可使用。

关于轴组参数的设定方法，请参阅 □ “3-4 轴组的设定步骤 (P.3-23)”。

关于 MC\_Write( 写入 MC 设定 ) 指令和 MC\_ChangeAxesInGroup( 写入轴组构成轴 ) 指令的详情，请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)”。

关于运动控制系统变量，请参阅 □ “6-6 运动控制系统变量 (P.6-16)”。

## 5-3-2 轴组基本设定

设定使用 / 不使用轴组、使用时的轴构成以及使用轴。

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]”的编号为此处设定的编号。	-	-
运动控制 <sup>*1</sup>	分配至原始恒定周期任务或固定周期任务(执行优先级 5)中的某一任务。 1: 原始恒定周期任务 2: 固定周期任务(执行优先级 5)	1 ~ 2	1
轴组使用	设定是否使用轴组。对未创建轴组和未使用轴组执行运动控制指令时, 将会发生异常。 <sup>*1</sup> 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	设定分配给轴组的轴号。 在 Sysmac Studio 中, 对轴 A0 ~ A3 选择轴变量名称。	2 ~ 4 轴	0

\*1. NX701 CPU 单元时设定。

\*2. 对含未使用轴的轴组执行 MC\_GroupEnable(启用轴组)指令时, 将会发生异常。

## 轴构成

下表所示为 MC 功能模块可使用的轴构成。

利用 Sysmac Studio, 根据实际装置设定轴构成。

轴构成	说明
2 轴	双轴的轴构成。例如, 双轴直交坐标系的机械。
3 轴	3 轴的轴构成。例如, 3 轴直交坐标系的机械。
4 轴	4 轴的轴构成。例如, 具有 3 轴直交坐标系和前端工具的转轴的机械。

## 构成轴

属于轴组的轴称为构成轴。此外，采用轴组指令的插补指令时，为了提高程序的重复利用性，使用逻辑轴（轴 A0 ~ 轴 255）替代轴号（轴 A0 ~ 轴 A3）。

构成轴设定轴组所属轴的轴号和逻辑轴编号的分配。

逻辑轴可从伺服轴、虚拟伺服轴中选择。

利用 Sysmac Studio，从轴 A0 开始，依次设定轴构成选择轴数的轴。

创建多个轴组时，也是按轴组分别从轴 A0 开始依次设定轴号。

此外，也可以在多个轴组中设定相同的轴号。

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（轴号）。
3 轴	设定轴 A0、轴 A1、轴 A2 的轴变量名称（轴号）。
4 轴	设定轴 A0、轴 A1、轴 A2、轴 A3 的轴变量名称（轴号）。



### 使用注意事项

- 轴组的构成轴请全部分配至同一任务。
- 多个轴组中设定相同的轴号时，设定了相同轴号的轴组无法同时激活。
- [ 控制功能 ] 的设定为 [ 仅单轴位置控制 ] 的轴，无法分配为轴组的构成轴。



### 版本相关信息

Ver1.04 以上的 CPU 单元和 Ver1.05 以上的 Sysmac Studio 组合时，如果是伺服轴或虚拟伺服轴，则在轴组中既能设定使用轴，也能设定未使用轴（可切换为使用轴）。

### ● 构成轴的设定示例

- 对轴组分配轴号 1、2、5、8 等 4 轴时，如下所示

逻辑轴	轴号
轴 A0	轴 1
轴 A1	轴 2
轴 A2	轴 5
轴 A3	轴 8

- 对轴组分配轴号 1、8、2 等 3 轴时，如下所示

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

### 5-3-3 轴组动作设定

根据控制装置的规格，如最高插补速度、轴组停止方法等，对轴组动作进行设定。  
使用 NX 系列位置接口单元时，请参阅 □□ “NX 系列 位置接口单元 用户手册 (SBCE-374)”。

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。 设为“0”时，无插补速度限制。 通过轴组动作指令指定了超过最高插补速度的插补目标速度时，以最高插补速度进行动作。 (单位：指令单位 /s)	双精度实数型的正数、0	80000000
最大插补加速度	设定轨迹的最大插补加速度。 设为“0”时，无插补加速度限制。 (单位：指令单位 /s <sup>2</sup> )	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。 设为“0”时，无插补减速度限制。 (单位：指令单位 /s <sup>2</sup> )	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，优先向目标位置停止后，产生减速超限。指定超过最大插补加减速速度时的动作。 0: 提高加减速速度 (将合并切换为等待)* <sup>1</sup> 1: 提高加减速速度 2: 异常停止* <sup>2</sup>	0 ~ 2	0
插补速度警告值	设定对最高插补速度 (用于输出插补速度警告) 的比例。 设为“0”时，不输出插补速度警告。(单位：%)	0 ~ 100	0
插补加速度警告值	设定对最高插补加速度 (用于输出插补加速度警告) 的比例。 设为“0”时，不输出插补加速度警告。 (单位：%)	0 ~ 100	0
插补减速度警告值	设定对最高插补减速度 (用于输出插补减速度警告) 的比例。 设为“0”时，不输出插补减速度警告。 (单位：%)	0 ~ 100	0
轴组停止方法	多轴协调动作中的轴发生即停异常时，设定未发生异常的构成轴的停止方法。 0: 立即停止 1: 使各轴减速停止 (各轴的最大减速度) 3: 立即停止，同时执行伺服 OFF	0、1、3	0
中心点补偿容许率	圆弧插补指令的中心点指定方式中，起点与中心点、终点与中心点的距离不同时，进行补偿，使两者相等。按照对半径的比率设定该补偿的容许范围。将比率设定在 0.1% 以上。 设为“0”时，不执行误差检查。	单精度浮点型 0 ~ 100	0

\*1. Ver.1.10 以上的 CPU 单元，无法将合并切换为等待。详情请参阅 □□ “9-5-7 运动控制指令的多重启动 (缓存模式) (P.9-44)”。

\*2. Ver.1.10 以上的 CPU 单元，合并动作时不会异常停止。详情请参阅 □□ “9-5-7 运动控制指令的多重启动 (缓存模式) (P.9-44)”。

### 5-3-4 启用轴组

为了使用户程序中对轴组的动作指令生效，通过 MC\_GroupEnable(启用轴组) 指令指定生效的轴组编号。  
对无效的轴组启动运动控制指令时，指令发生异常。  
并且，可以同时使多个轴组生效，但是，如果使包含相同轴的轴组生效，指令将发生异常。

在各个作业工序中，要以不同轴组使某个轴动作时，事先创建包含该轴的多个轴组。然后，依次切换 MC\_GroupEnable(启用轴组) 指令和 MC\_GroupDisable(不启用轴组) 指令，可以实现。

在多轴动作中，启动 MC\_GroupDisable(不启用轴组) 指令时执行减速停止。

# 6

## 运动控制程序

本章对运动控制程序的规格和编写程序等的操作步骤进行说明。

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 启动相关的通用规则	6-8
6-4-2 启动的时序图	6-10
6-4-3 重启运动控制指令时的时序图	6-12
6-4-4 多重启动运动控制指令时的时序图	6-13
6-5 位置的处理	6-14
6-5-1 位置的种类	6-14
6-5-2 各种轴可使用的位置的种类	6-15
6-6 运动控制系统变量	6-16
6-6-1 运动控制系统变量概要	6-16
6-6-2 运动控制系统变量的结构	6-17
6-6-3 运动控制系统变量一览	6-19
6-7 凸轮表与凸轮数据变量	6-29
6-8 运动控制程序的编写方法	6-32
6-9 凸轮表的创建方法	6-34

## 6-1 概要

NJ/NX 系列 CPU 单元可执行时序控制和运动控制。

在用户程序中记述运动控制指令，以 EtherCAT 从站伺服驱动器或 NX 系列位置接口单元等为对象，执行运动控制。

本手册中将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分配轴变量。

不分配轴变量时，分配与通用从站相同的 I/O 设备变量。

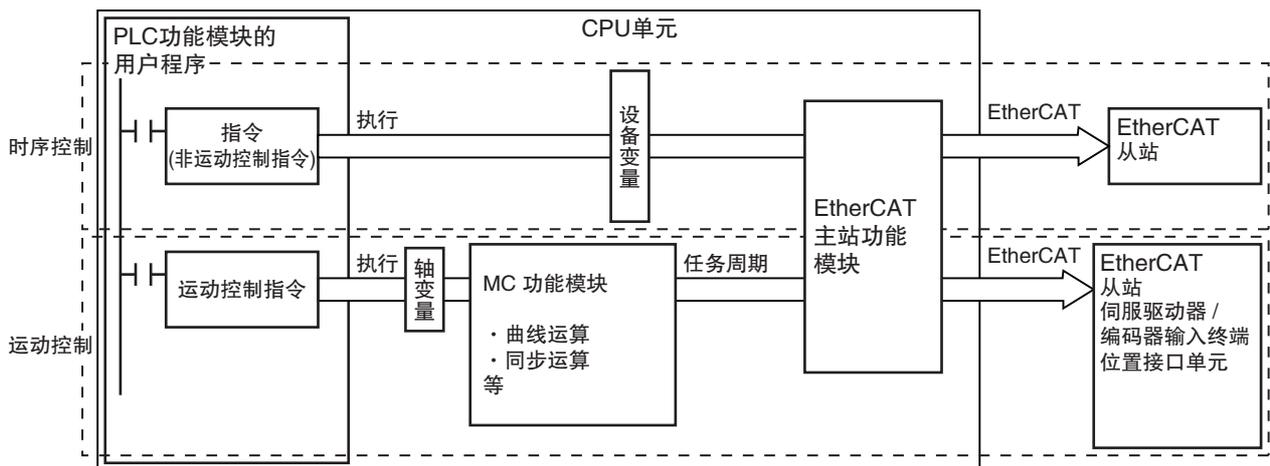
运动控制指令可以在原始恒定周期任务或固定周期任务（执行优先级 5、16）中使用。



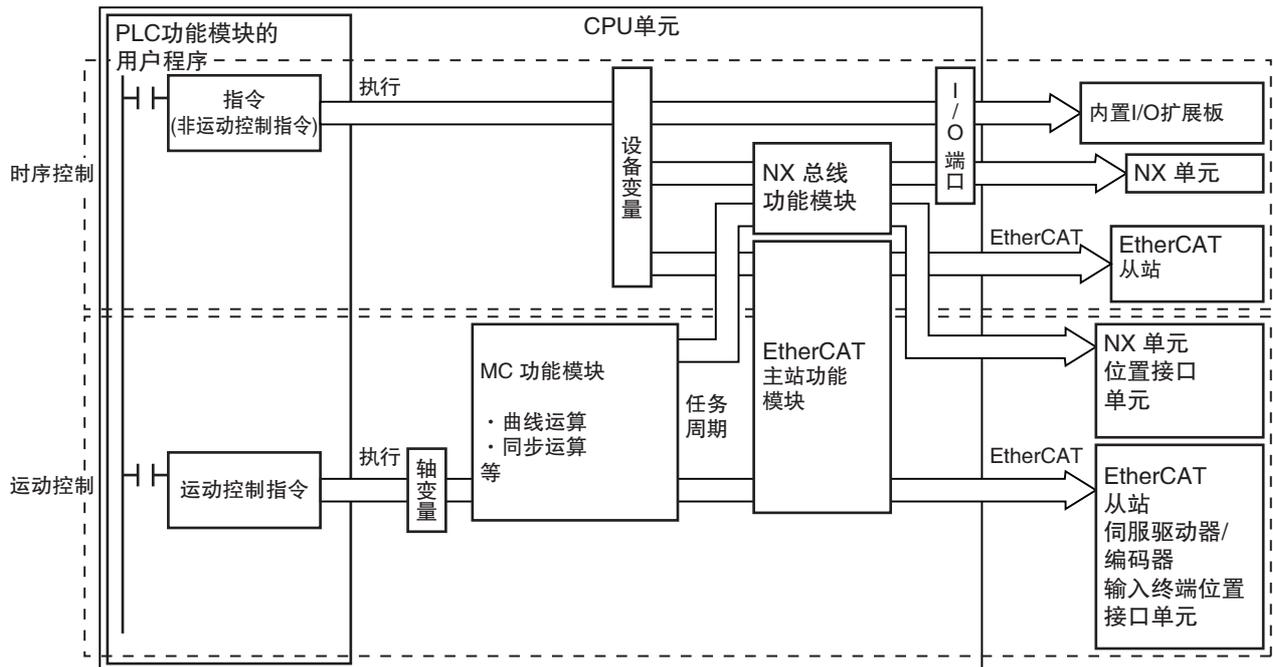
### 版本相关信息

使用 Sysmac Studio Ver.1.09 以上版本，可对分配给轴变量的从站、单元的 I/O 端口分配设备变量。  
详情请参阅 □ “2-4-2 EtherCAT 主站功能与运动控制功能模块之间的关系 (P.2-19)”。

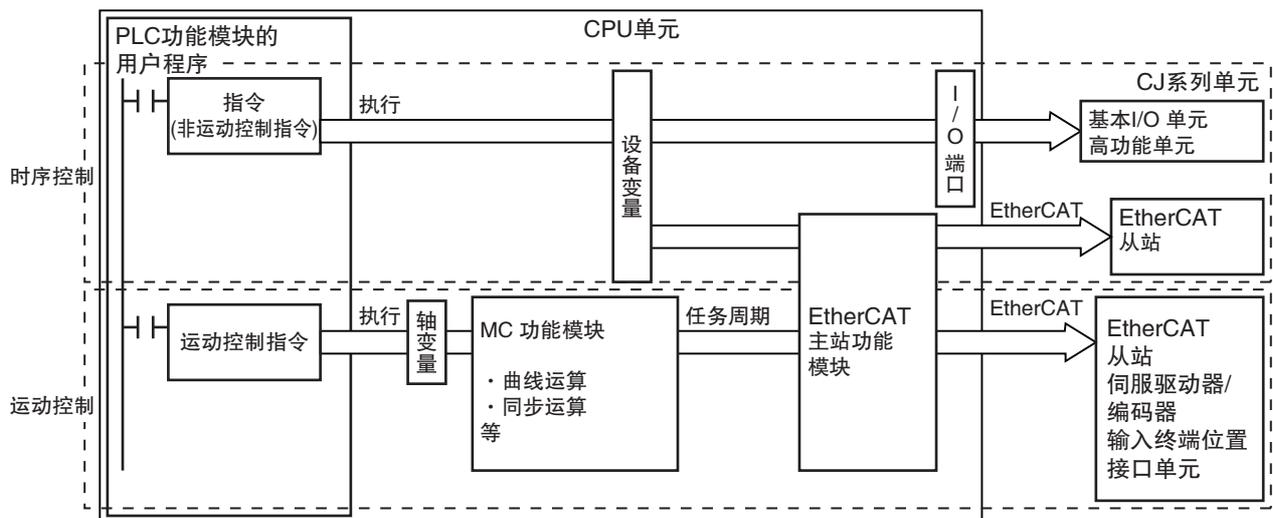
### ● NX701 CPU 单元



## ● NX1P2 CPU 单元



## ● NJ系列 CPU 单元



## 6-2 运动控制指令

要从 NJ/NX 系列的用户程序执行运动控制功能，需使用作为功能块(以下简称为 FB)定义的运动控制指令。MC 功能模块的运动控制指令以 PLCopen® 的运动控制用功能块的技术规格为基础。运动控制指令有 PLCopen® 定义的指令和 MC 功能模块独有的指令两种。

下面对 PLCopen® 运动控制用 FB 的概要和 MC 功能模块的规格进行说明。

NJ/NX 系列的功能块 (FB) 的基本事项请参阅 □ “NJ/NX 系列 CPU 单元 用户手册 软件篇 (SBCA-359)”。Ver.1.05 以上版本的 CPU 单元和 Ver.1.06 以上版本的 Sysmac Studio 组合时，可使用 NX 系列 位置接口单元。使用 NX 系列 位置接口单元时，请参阅 □ “NX 系列 位置接口单元 用户手册 (SBCE-374)”。

### 6-2-1 PLCopen® 运动控制用功能块

PLCopen® 对运动控制用功能块进行了标准化，制定了 IEC 61131-3(JISB 3503) 规格语言程序接口的定义。除了单轴定位、电子凸轮功能、多轴插补控制等功能，还对启动指令的基本步骤进行了定义。

利用 PLCopen® 运动控制用功能块，可减少对硬件的依赖，提高用户程序的可再利用性。还能降低培训、支持等的成本。



#### 参考

##### 何谓 PLCopen®

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

IEC 61131-3 是 PLC 编程的国际标准规格。

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

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

### 6-2-2 运动控制功能模块的运动控制指令

运动控制指令分为以下 3 种。

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

MC 功能模块可使用的指令的一览和详情请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)”。

## 6-3 状态变化

MC 功能模块的轴和轴组的状态变化以及启动指令后的状态变化取决于以 PLCopen® 的运动控制用功能块的技术规格。

下面对包括 MC 功能模块整体在内的各状态和状态变化进行说明。

### 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	启用运动控制指令的状态。 分析用户程序的运动控制指令，执行运动控制。 与 CPU 单元的动作模式无关，可设为“MC 功能模块运行中”。
MC 功能模块试运行中 *2	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. 可通过 MC 通用变量 “\_MC\_COM.Status.RunMode” 进行监控。

\*2. 可通过 MC 通用变量 “\_MC\_COM.Status.TestMode” 进行监控。

\*3. 可通过 MC 通用变量 “\_MC\_COM.Status.CamTableBusy” 进行监控。

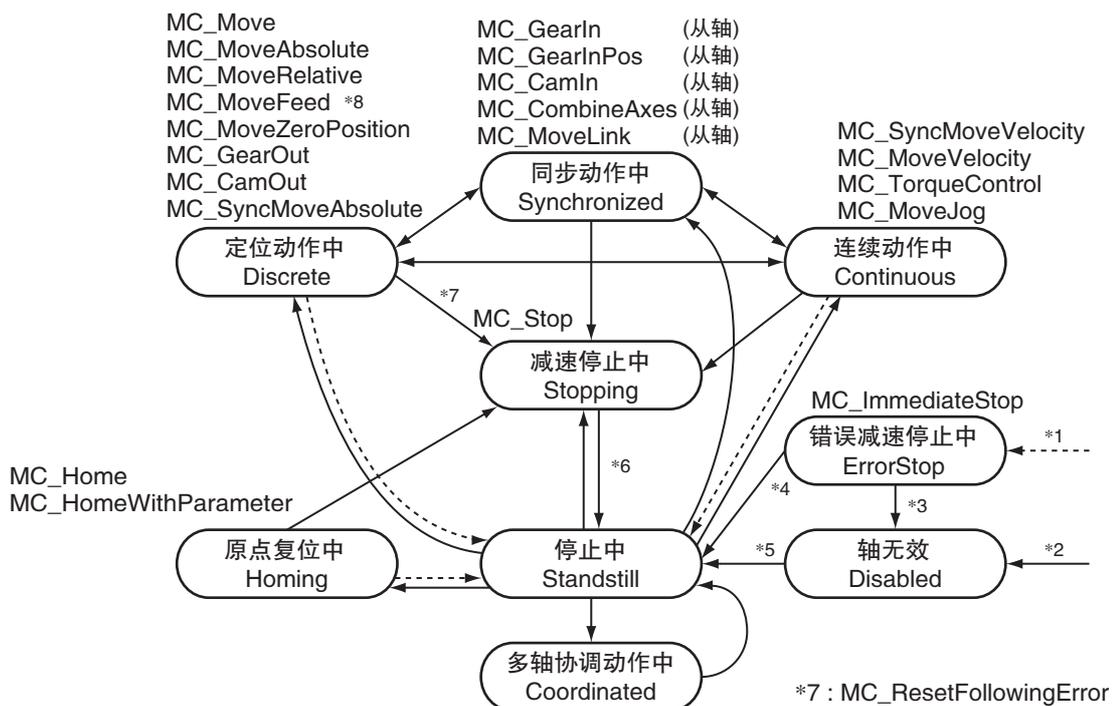
\*4. 可通过 MC 通用变量 “\_MC\_COM.Status.GenerateCamBus” 进行监控。

\*5. 要关闭 CPU 单元的电源时，请确认并非凸轮表生成执行中。在凸轮表生成执行中，如关闭 CPU 单元的电源，将无法正确生成凸轮表。

### 6-3-2 轴的状态

启动轴相应的运动控制指令后轴的动作如下图所示。

原则上，依次执行运动控制指令，轴处于下表中的任一状态。



- \*1 发生轴异常时，“多轴协调动作中”以外的任一状态均会发生变化。
- \*2 MC\_Power指令的“Status(可运行)”输出为“FALSE”且未发生轴异常时，任一状态均会发生变化。(伺服OFF)
- \*3 伺服OFF时通过MC\_Reset指令或ResetMCErrror指令解除异常后，状态发生变化。
- \*4 伺服ON时通过MC\_Reset指令或ResetMCErrror指令解除异常后，状态发生变化。
- \*5 将MC\_Power指令的“Enable(启用)”输入设为“TRUE”，MC\_Power指令的“Status(可运行)”输出为“TRUE”时，状态发生变化。(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，状态发生变化。
- \*7 通过MC\_ResetFollowingError指令，切换至减速停止中。
- \*8 在MC\_MoveFeed指令的输入变量“MoveMode(选择移动方法)”中设定“2: \_mcVelocity(速度控制)”时，会变为“连续动作中”，直至检测到触发输入。

状态名称	定义
伺服 OFF	轴处于伺服 OFF 的状态。 切换至该状态时，解除指令多重启动的待机状态。
轴无效	轴处于伺服 OFF、停止中、启动准备就绪的状态。
错误减速停止中 *1	轴处于伺服 OFF、停止中，发生轴异常的状态。
伺服 ON	轴处于伺服 ON 的状态。
停止中	轴处于伺服 ON、停止的状态。
定位动作中	正在执行指定了目标位置的定位的状态。 也包括等待定位完成的状态和定位动作中将超调值设定为“0”而使速度为“0”的状态。
连续动作中	正在执行未指定目标位置的连续动作的状态。 速度控制或转矩控制时变为此状态。 也包括将目标速度设为“0”而使速度为“0”的状态、以及连续动作中将超调值设定为“0”而使速度为“0”的状态。
同步动作中	通过同步控制指令同步控制轴的状态。 也包括同步控制指令切换后的同步等待状态。
减速停止中	通过 MC_Stop(强制停止)指令或 MC_TouchProbe(启用外部锁定)指令直至轴停止的状态。 也包括通过 MC_Stop(强制停止)指令停止后，Execute(启动)为 TRUE 的状态。 该状态下无法通过轴指令执行动作。 执行后指令的 CommandAborted(执行中断)变为 TRUE。
错误减速停止中 *1	轴处于伺服 ON、发生轴异常的状态。 也包括执行了 MC_ImmediateStop(立即停止)指令的状态和轴因发生异常而减速的状态。 该状态下无法执行轴动作指令。执行后指令处于中断(CommandAborted = 1)状态。
原点复位中	通过 MC_Home(原点复位)指令或 MC_HomeWithParameter(参数指定原点复位)指令搜索原点的状态。
多轴协调动作中	轴处于通过轴组指令启用轴组的状态。 轴组在“组动作中”、“减速停止中”或“错误停止中”的情况下动作的状态。

\*1. “错误减速停止中”含有轴处于伺服 OFF 或伺服 ON 的 2 种情况。

(注) 可通过轴变量“\_MC\_AX[0 ~ 255].Status”的各结构要素变量监控轴的状态。



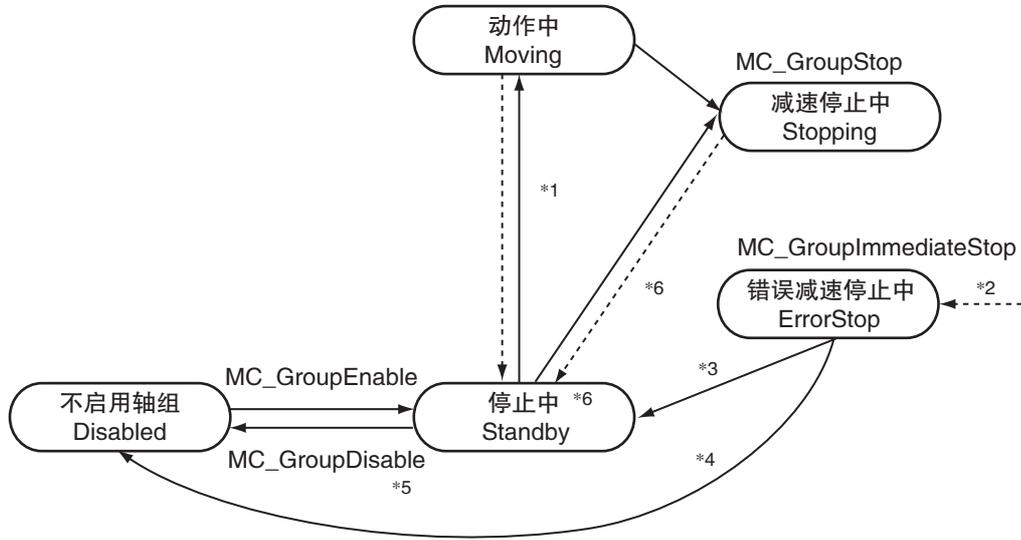
#### 版本相关信息

对于 NX701 CPU 单元，以 \_MC\_AX[\*] 开头的变量名可能为 \_MC1\_AX[\*]、\_MC2\_AX[\*]。

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

## 6-3-3 轴组状态

启动轴组相应的运动控制指令后轴组的动作如下图所示。



- \*1 通过所有的轴组动作指令切换状态。
- \*2 所有状态发生变化。不启用轴组时，发生异常后状态仍会发生变化。
- \*3 启用轴组时，通过 MC\_GroupReset 指令或 ResetMCErrror 指令切换状态。
- \*4 不启用轴组时，通过 MC\_GroupReset 指令或 ResetMCErrror 指令切换状态。
- \*5 在错误减速停止中的状态下，启动 MC\_GroupDisable 指令后，仍保持相同状态不变。
- \*6 MC\_GroupStop 指令的输出变量 “Done” = TRUE，输入变量 “Execute” = FALSE 时，切换状态。
- \*7 在停止中 (Standby) 的状态下，满足以下所有条件后变为启动准备完成 (Ready)。
  - 所有构成轴处于伺服 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
不启用轴组	不启用轴组的状态。 切换至该状态时，解除指令多重启动的待机状态。
错误减速停止中 *1	不启用轴组时发生轴组异常的状态。
启用轴组	启用轴组的状态。
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	通过轴组指令的动作指令执行指定了目标位置的定位的状态。 也包括等待到位的状态和组动作中将超调值设定为“0”而使速度为“0”的状态。
减速停止中	执行了 MC_GroupStop(轴组强制停止)指令的状态。 也包括通过 MC_GroupStop(轴组强制停止)指令停止后，Execute(启动)为 TRUE 的状态。 该状态下无法通过轴组指令执行动作。 执行后指令的 CommandAborted(执行中断)变为 TRUE。
错误减速停止中 *1	轴组发生异常的状态。 本状态也包括执行了 MC_GroupImmediateStop(轴组立即停止)指令的状态和轴组因发生异常而减速的状态。 该状态下禁止执行多轴协调指令。 执行后 CommandAborted(执行中断)变为 TRUE。

- \*1. “错误减速停止中”中含有不启用轴组和启用轴组的 2 种情况。  
(注) 可通过轴组变量 “\_MC\_GRP[0 ~ 63].Status”、“\_MC1\_GRP[0 ~ 63].Status”、“\_MC2\_GRP[0 ~ 63].Status”的各结构要素变量监控轴组的状态。

## 6-4 运动控制指令的启动和状态

将启动运动控制指令的变量和表示指令执行状态的变量作为 MC 功能模块的通用规则进行定义。

启动运动控制指令功能的输入变量分为“Execute”和“Enable”的 2 种。表示指令执行状态的输出变量包括“Busy”、“Done”、“CommandAborted”、“Error”等。

### 6-4-1 启动相关的通用规则

MC 功能模块的通用规则如下表所示。

□ “6-4-2 启动的时序图 (P.6-10)” 中记载有启动示例。

请对照参阅。

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, 不会多个同时变为 TRUE。 同样地, 输出变量 Active(控制中)、Done(完成)、Error(错误)、CommandAborted(执行中断) 也不会多个同时变为 TRUE。 Busy(执行中) 和 Active(控制中) 有时会同时变为 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) 的下降沿变为 FALSE。 在 Execute(启动) 的下降沿不会停止执行运动控制指令。 指令实例执行完成前, 如果指令实例的状态发生变化, 则即使 Execute(启动) 变为 FALSE, 至少在一个周期内相应的输出变量仍变为 TRUE。 执行 MC_Reset(轴错误复位) 指令、MC_GroupReset(轴组错误复位) 指令或 ResetMCError(所有错误复位) 指令前, 输出变量 Error(错误) 和 ErrorID(错误代码) 不会清零。*1 执行运动控制指令时, 将相同指令实例的 Execute(启动) 再次从 FALSE 变为 TRUE 进行启动(重启指令)后, CommandAborted(执行中断) 仍不会变为 TRUE。
输入参数	通过输入变量 Execute(启动) 启动的运动控制指令的输入参数使用 Execute(启动) 处于上升沿时的值。 通过输入变量 Enable(启用) 启动的运动控制指令的输入参数使用 Enable(启用) 为 TRUE 时的各周期的值。
输入参数的省略	省略指令实例的输入参数时, 使用初始值。*2
“Position(目标位置)”与 “Distance(移动距离)”	输入变量 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, 表示 2 个位置之间的差异。
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) 可为正值、负值或“0”。
异常处理动作	表示执行指令实例时发生的异常的输出变量分为 2 种。 此类输出的定义如下。 · Error(错误): 输出变量“Error”的上升沿表示执行指令实例时发生异常。 · ErrorID(错误代码): 表示异常原因的代码。 输出变量 Done(完成)、InVelocity(达到目标速度)、InGear(达到齿轮比)、InSync(同步中) 表示正常完成或正常动作, 与输出变量 Error(错误) 互相排斥。 异常种类: · 指令实例异常(例: 超过参数范围、违反状态变化的条件) · 轴异常(例: 位置偏差超限、伺服驱动器异常) 指令实例发生异常时, 有时不会发生轴异常便停止轴。
输出变量“Done(完成)”的动作	指令动作正常完成或到达指令指定的状态时, 输出变量 Done(完成)、InGear(达到齿轮比)、InSync(同步中) 变为 TRUE。对于需指定目标位置的指令, 变为 TRUE 的时间因轴参数的“到位检查时间”设定而异。 · 如果“不为 0”, 则在完成定位的下一周期内变为 TRUE。 · 如果为“0”, 则在指令结束(输出完成)的下一周期内变为 TRUE。 相同轴运行的多个指令因其它动作指令而中断时, 最初的指令输出变量“Done(完成)”不会变为 TRUE, 直至轴动作到达最终目标位置。

项 目	原 则
输出变量 “CommandAborted(执行中断)”的动作	指令指定的动作因其它动作指令而中断时，输出变量 CommandAborted(执行中断) 变为 TRUE。 本 MC 功能模块中，启动运动控制指令时，对象轴和轴组发生异常或减速停止过程中变为 TRUE。 CommandAborted(执行中断) 变为 TRUE 时，其它输出变量变为 FALSE。
超过有效范围的输入	启动时如果输入变量的值超过了有效范围，则指令实例会发生异常。
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。 在输入变量 Execute(启动) 的上升沿，Busy(执行中) 变为 TRUE；在输出变量 Done(完成)、CommandAborted(执行中断)、Error(错误) 变为 TRUE 时，Busy(执行中) 变为 FALSE。 由于不确定上述输出变量何时发生变化，因此编程时请确保在每个周期内均执行 *3 指令实例，以便在 Busy(执行中) 为 TRUE 时监控输出变量的变化情况。 在一个轴或一个轴组中，多个指令实例的 Busy(执行中) 可变为 TRUE，而 1 个指令实例的输出变量 Active(控制中) 仅 1 次变为 TRUE。 MC_Phasing(主轴相对值相位补偿) 指令除外。
输出变量 “Active(控制中)”	指令实例获取相应轴的控制权后，输出变量 Active(控制中) 变为 TRUE。 输出变量 Active(控制中) 可能迟于 Busy(执行中) 发生变化。*4

- \*1. 此规格与 PLCopen® 不同。PLCopen® 标准下，Error(错误) 和 ErrorID(错误代码) 在 Execute(启动) 的下降沿变为 “0”。Error(错误) 为 TRUE 时也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，因此也不启动指令。解除异常后，Enable(启用) 为 TRUE 时启动 Enable 型运动控制指令。
- \*2. 用梯形图语言编制程序时，请将触点配置在输入变量的 Execute(启动) 或 Enable(启用) 和左母线之间。如果不配置触点，直接与左母线连接，则执行编连时会发生异常。  
对于保留 (Reserved) 的输入变量，请省略输入参数或设定初始值。
- \*3. ST 结构指令中条件表达式或选择值不一致时，不执行其语句中记述的指令。详情请参阅  “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)”。
- \*4. 输出变量的时序图请参阅  “NJ/NX 系列 指令基准手册 运动篇 (SBCE-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处于正常执行中的状态。  
详情请参阅 “10-2-1 EtherCAT 通信的监控和伺服 ON (P.10-3)”。
- 编写用户程序时，请确保在执行用户程序的首个周期内 Execute(启动) 为 “FALSE”。

## 6-4-2 启动的时序图

MC 功能模块的运动控制指令为始终执行型 FB。

下面对用于启动运动控制指令的输入变量“Execute(启动)”的指令为 Execute 型，“Enable(启用)”的指令为 Enable 型进行说明。

启动条件	内容
Execute 型	<p>在运动控制指令的输入变量“Execute(启动)”的上升沿启动的指令。 继续执行指令，直至变为下一状态。</p> <ul style="list-style-type: none"> <li>· 指令指定的动作的完成</li> <li>· 通过其它运动控制指令启动的动作的中断</li> <li>· Execute(启动)再次处于上升沿时重启</li> </ul> <p>在 Execute(启动)的上升沿，导入其它输入变量的值。</p>
Enable 型	<p>运动控制指令的输入变量“Enable(启用)”为 TRUE 时，在每个周期内执行的指令。 “Enable(启用)”为 TRUE 时，也在每个周期内导入其他输入变量。 但以下情况属于例外，即在“PositiveEnable(启用正方向移动)”或 “NegativeEnable(启用负方向移动)”的上升沿导入 MC_MoveJog(微动移动)的输入变量 “Velocity(目标速度)”、“Acceleration(加速度)”、“Deceleration(减速度)”。</p>



### 使用注意事项

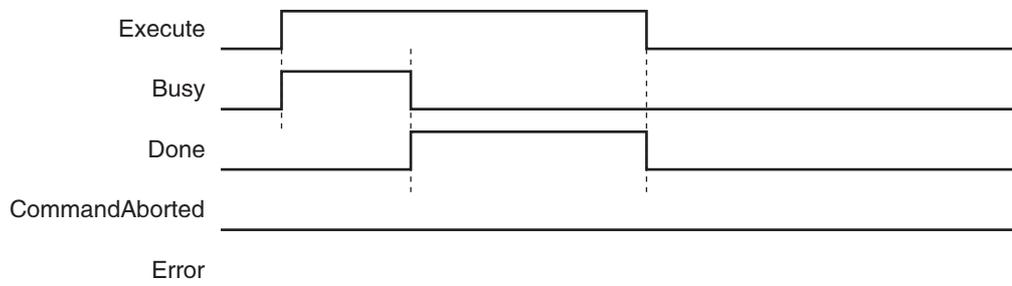
手册中记载的时序图可能与 Sysmac Studio 的数据跟踪中显示的时序不同。

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

## Execute 型的指令执行时序图

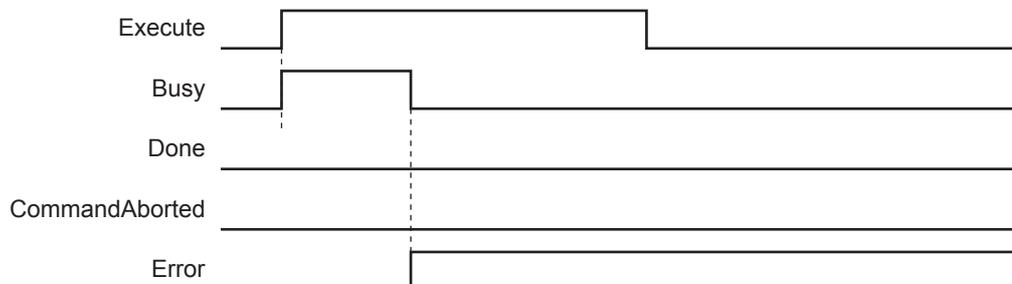
- 输入变量 Execute(启动) 为 TRUE 时，指令的动作完成。

如下图所示，未发生异常，直至将 Execute(启动) 设为 FALSE。

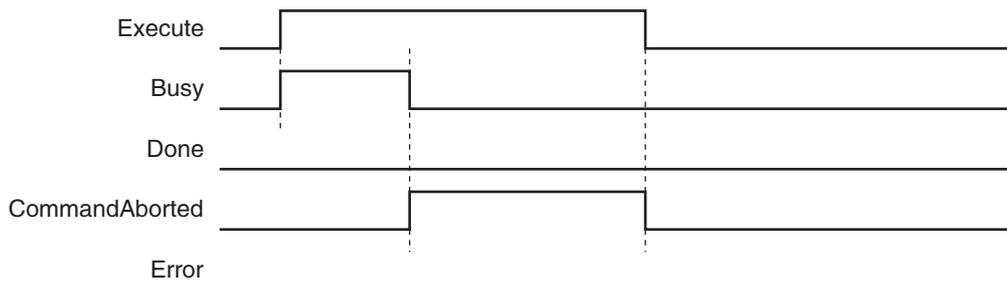


- 如下图所示，输入变量 Execute(启动) 为 TRUE 时，发生异常。

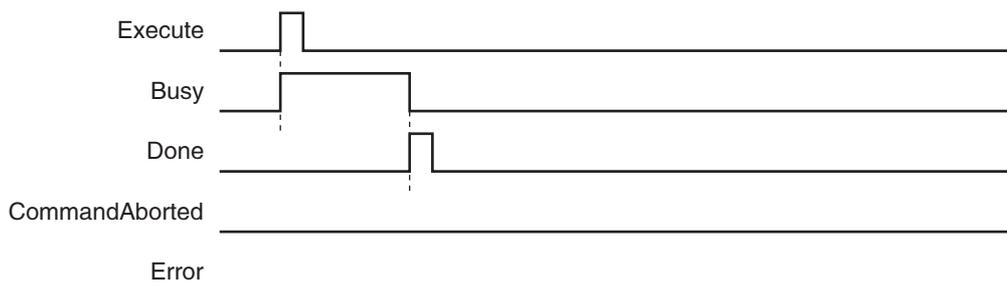
将 Execute(启动) 设为 FALSE 后，输出变量 Error(错误) 仍保持 TRUE 不变。



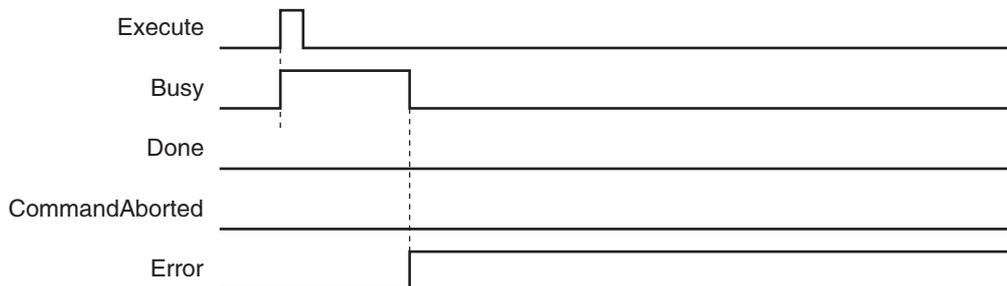
- 如下图所示，输入变量 Execute(启动) 为 TRUE 时，中途中断执行指令。



- 如下图所示，仅在 1 个周期内将输入变量 Execute(启动) 设为 TRUE，指令未发生异常。指令动作完成后，输出变量 Done(完成) 仅在 1 个周期内变为 TRUE。

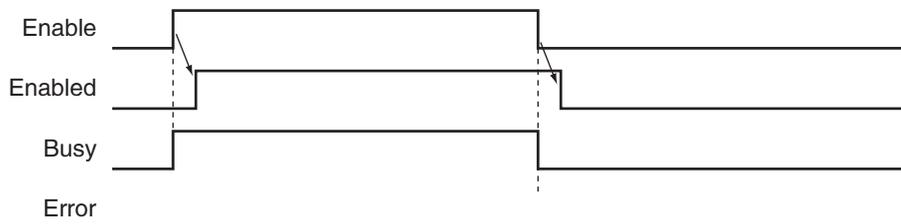


- 如下图所示，仅在 1 个周期内将输入变量 Execute(启动) 设为 TRUE，指令发生异常。输出变量 Error(错误) 保持 TRUE 不变。

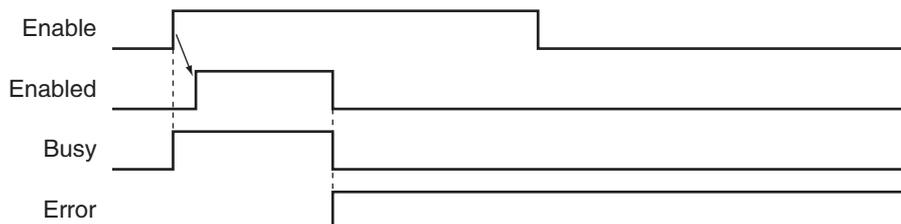


## Enable 型的指令执行时序图

- 如下图所示，将输入变量 “Enable(启用)” 设为 TRUE，指令未发生异常。



- 如下图所示，将输入变量 “Enable(启用)” 设为 TRUE，指令发生异常。





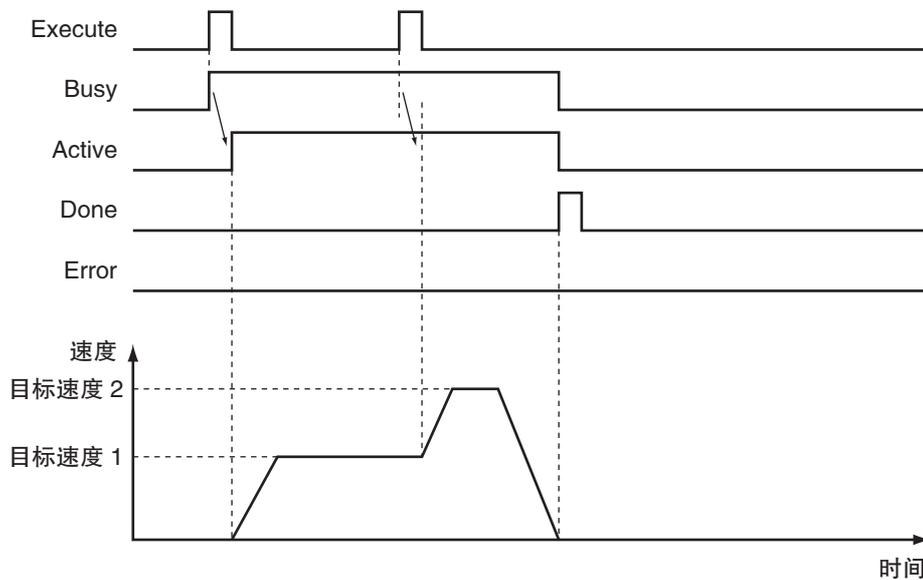
## 参考

还包括 MC\_ZoneSwitch( 监控区域 ) 指令、MC\_AxesObserve( 监控轴间偏差 ) 指令等 Enable 和 Enabled 的时间相同的指令。各指令的时间的详情请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)”。

### 6-4-3 重启运动控制指令时的时序图

执行运动控制指令过程中变更相同实例的输入变量值, 使输入变量 “Execute( 启动 )” 按 TRUE → FALSE → TRUE 的顺序变化时, 会按变更的输入变量值运行。

如下图所示, 通过 MC\_MoveAbsolute( 绝对值定位 ) 变更速度。



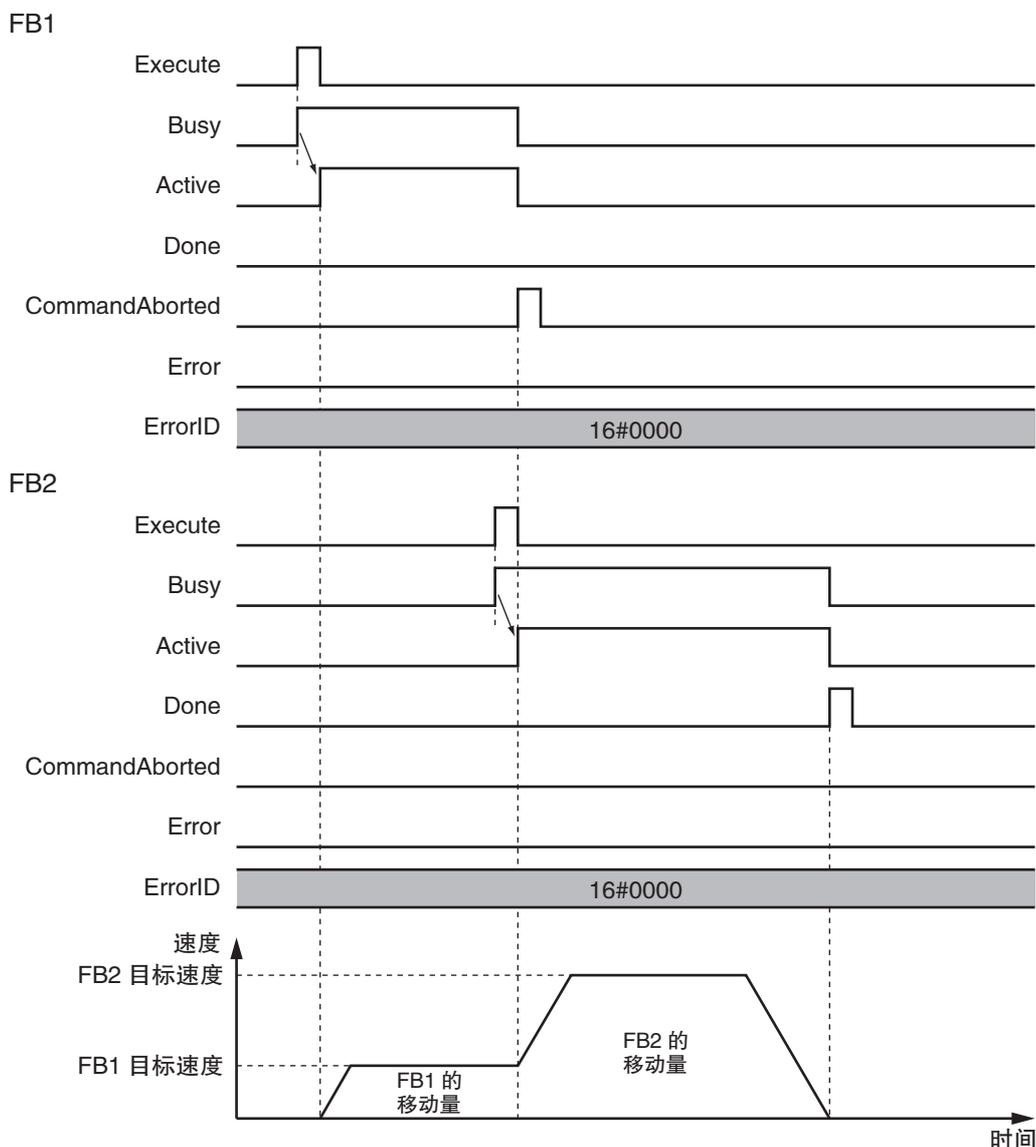
通过 MC 功能模块重启指令的详情请参阅 □ “9-5-6 运动控制指令的重启 (P.9-39)”、□ “9-7-4 多轴协调控制的运动控制指令的重启 (P.9-60)”。

## 6-4-4 多重启动运动控制指令时的时序图

可对动作中的轴启动其它实例。

通过设定输入变量 BufferMode(选择缓存模式),可指定动作开始的时间。

如下图所示,将 BufferMode(选择缓存模式)设定为中断,多重启动2个 MC\_MoveAbsolute(绝对值定位)指令。  
图中的“FB1”和“FB2”为指令的实例名称。



在 MC 功能模块中多重启动指令请参阅 □ “9-5-7 运动控制指令的多重启动 (缓存模式) (P.9-44)”、  
□ “9-7-5 多轴协调控制的运动控制指令的多重启动 (缓存模式) (P.9-61)”。

## 6-5 位置的处理

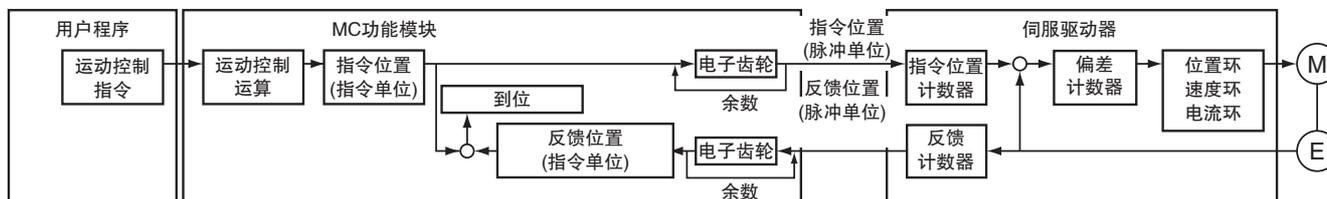
下面对运动控制程序处理的位置进行说明。

### 6-5-1 位置的种类

MC 功能模块中含有以下 2 种位置。

位置类型	定义
指令位置	用于控制轴的由 MC 功能模块输出的位置。
反馈位置	从伺服驱动器或计数器输入的实际位置。

指令位置与反馈位置的关系如下图所示。



指令位置与反馈位置的以下项目相同。

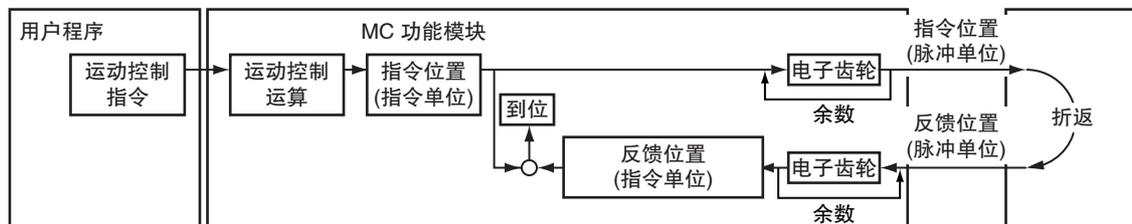
项 目	指令位置	反馈位置
计数模式	设定线性模式或旋转模式。	与指令位置的计数模式相同。
位置的单位	从 mm、 $\mu\text{m}$ 、nm、inch、degree、pulse 中设定任意一个。	与指令位置的单位相同。
软件限位	设定软件可操作的范围。	与指令位置的范围相同。
当前位置变更	将当前位置变更为任意位置。	同时设定与指令位置相同的值。 <sup>*1</sup>
原点确定	具有原点确定或未确定的状态。	与指令位置的原点状态相同。

\*1. 如果变更前存在位置偏差，则将位置偏差值的保持值设定至反馈位置。



#### 参考

- 对于虚拟伺服轴，以脉冲为单位转换指令当前值，并将该值再次按表示单位转换后的值作为反馈当前值。



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

## 6-5-2 各种轴可使用的位置的种类

各种轴可使用的位置的种类如下所示。

轴种类	位置的种类	
	指令位置	反馈位置
伺服轴	可以使用	可以使用
虚拟伺服轴	可以使用	可以使用 <sup>*1</sup>
编码器轴	不可使用	可以使用
虚拟编码器轴	不可使用	可以使用 <sup>*2</sup>

\*1. 虚拟伺服轴时，“反馈位置 = 指令位置”。

由于在 MC 功能模块内部是以双精度实数型进行处理，因此可能包括运算误差。

\*2. 用于不存在实际编码器的场合等。

## 6-6 运动控制系统变量

下面对 MC 功能模块的所属变量进行说明。

### 6-6-1 运动控制系统变量概要

NJ/NX 系列为依照 IEC 61131-3 标准的控制器。

在 NJ/NX 系列的程序中，将参数设定和状态信息等数据作为变量处理。

此类变量中，将 MC 功能模块所属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的一览。

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

#### ● MC 通用变量

监控 MC 功能模块的通用状态的变量。变量名称为“\_MC\_COM”。

#### ● 轴变量

用于处理 EtherCAT 从站的伺服驱动器、编码器输入终端、NX 系列位置接口单元以及虚拟伺服驱动器、编码器输入的变量。

用户程序中可使用系统定义变量的变量名称或通过 Sysmac Studio 创建的变量名称。

可通过 Sysmac Studio 创建的变量名称按轴变更为任意的变量名称。

a) 系统定义变量的变量名称

- \_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

- MC\_Axis000 ~ MC\_Axis255 (默认)

#### ● 轴组变量

用于处理集中了多个轴的轴组的变量。

用户程序中可使用系统定义变量的变量名称或通过 Sysmac Studio 创建的变量名称。

可通过 Sysmac Studio 创建的变量名称按轴组变更为任意的变量名称。

a) 系统定义变量的变量名称

- \_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

- MC\_Group000 ~ MC\_Group063 (默认)

NJ/NX 系列中处理的变量请参阅 □ “NJ/NX 系列 CPU 单元 用户手册 软件篇 (SBCA-359)”。

## 通过运动控制系统变量处理的数据类型

运动控制系统变量包括基本数据类型和派生数据类型。

### ● 基本数据类型

分类	数据类型	大小	数值范围	描述方法
布尔	BOOL	2*1	FALSE、TRUE	FALSE、TRUE
整数	UINT	2	0 ~ +65535	2 进制格式：开头加上“2#” <sup>*2</sup> 。 8 进制格式：开头加上“8#” <sup>*3</sup> 。 10 进制格式：开头加上“10#” <sup>*4</sup> 。 16 进制格式：开头加上“16#” <sup>*5</sup> 。 如果开头不添加任何内容，则为 10 进制。
	UDINT	4	0 ~ +4294967295	
浮点	LREAL	8	-1.79769313486231e+308 ~ -2.22507385850721e-308、 0、 2.22507385850721e-308 ~ 1.79769313486231e+308、 + ∞ /- ∞	以 ( 符号)+ 整数部分 +(小数点)+(小 数部分)+( 指数部分) 的形式进行描 述。 <sup>*6</sup> 带 ( ) 的可省略

\*1. BOOL 型的数据大小为 1 位，占用 2 个字节的存储器空间。

\*2. 2 进制格式例：2#1111\_1111, 2#1110\_0000

\*3. 8 进制格式例：8#377, 8#340

\*4. 10 进制格式例：-12, 0, 123\_456, +986, 10#1234

\*5. 16 进制格式例：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

### ● 衍生数据类型

种类	内容
枚举体型	将预设的名称列表内的一个名称设为值的数据类型。 数据类型的开头为“_e”。
结构体型	将多个数据类型按层次进行统一的数据类型。 数据类型的开头为“_s”。

NJ/NX 系列中处理的其它数据类型请参阅 □□ “NJ/NX 系列 CPU 单元 用户手册 软件篇 (SBCE-359)”。

## 运动控制系统变量的属性

运动控制系统变量具有相同的以下属性。

属性	运动控制系统变量的属性
全局 / 局部	全局变量
R/W 访问	只读
保持	不保持
网络公布	公布 <sup>*1</sup>

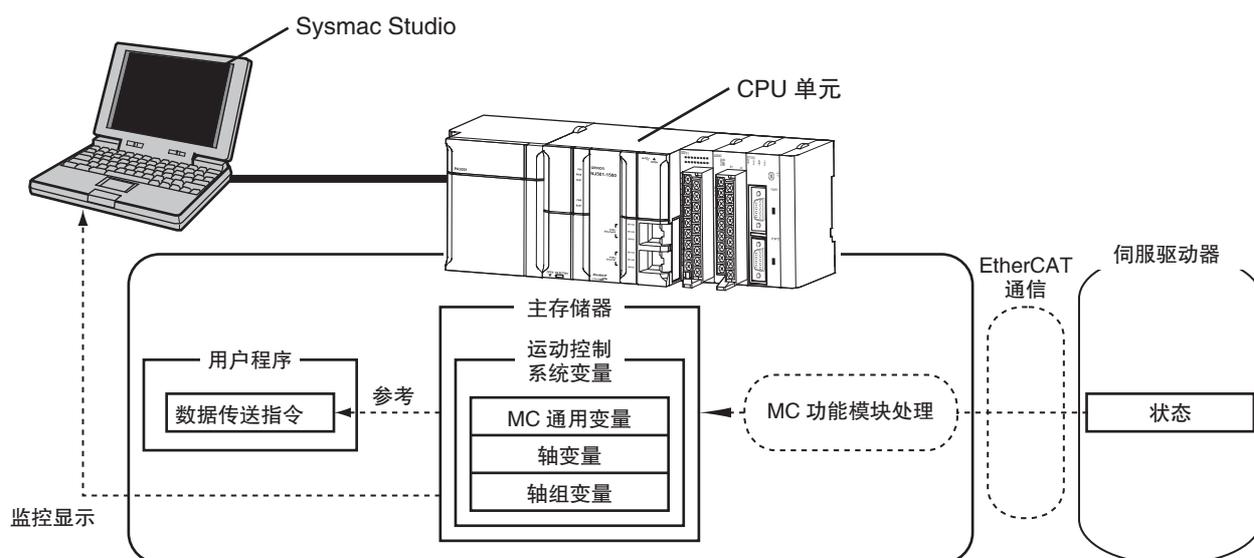
\*1. 以系统定义变量的变量名称在网络上公开。通过 Sysmac Studio 创建轴或轴组时的变量名称不在网络上公开。

### 6-6-2 运动控制系统变量的结构

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和 CPU 单元上的 NX 单元的状态信息及部分用于执行运动控制的 MC 参数设定构成。

运动控制系统变量可通过用户程序查看，也可通过 Sysmac Studio 监控显示。

运动控制系统变量按原始周期进行更新。



## 运动控制系统变量的更新时间

运动控制系统变量的更新时间在 NX701 CPU 单元、NX1P2 CPU 单元及 NJ 系列 CPU 单元中有所不同，如下所示。

### ● NX701 CPU 单元

将轴分配给原始恒定周期任务和固定周期任务 (执行优先级 5) 时，如下所示。

- 仅按原始周期更新分配给原始恒定周期任务的区域和固定周期任务 (执行优先级 5) 的基本设定区域。
- 固定周期任务 (执行优先级 5) 的其他区域，将不按初始值更新，而是按照固定周期任务 (执行优先级 5) 的任务周期更新。

更新对象的变量区域	原始恒定周期任务	固定周期任务 (执行优先级 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]		○

### ● NX1P2 CPU 单元及 NJ 系列 CPU 单元

运动控制系统变量按原始周期进行更新。

### 6-6-3 运动控制系统变量一览

下面按运动控制系统变量的种类，以一览表的形式进行说明。

#### MC 通用变量

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

下面对 MC 通用变量的构成及其要素进行详细说明。

变量名称	数据类型	名称	功能
<code>_MC_COM</code>	<code>_sCOMMON_REF</code>	MC 通用变量	
Status	<code>_sCOMMON_REF_STA</code>	MC 通用状态	
RunMode	BOOL	MC 运行中	MC 功能模块在运行中变为 TRUE。
TestMode	BOOL	MC 试运行中	通过 Sysmac Studio 试运行时变为 TRUE。
CamTableBusy	BOOL	凸轮表文件保存执行中 / 待机中	在执行或等待执行凸轮表文件保存时变为 TRUE。
GenerateCamBusy <sup>*1</sup>	BOOL	凸轮表生成执行中	执行凸轮表生成时变为 TRUE。
PFaultLvl	<code>_sMC_REF_EVENT</code>	MC 通用 部分停止故障	
Active	BOOL	正在发生 MC 通用部分停止故障	发生 MC 通用部分停止故障时变为 TRUE。
Code	WORD	MC 通用 部分停止故障代码	输出 MC 通用部分停止故障的代码。与事件代码前四位的值相同。
MFaultLvl	<code>_sMC_REF_EVENT</code>	MC 通用 轻度故障	
Active	BOOL	正在发生 MC 通用轻度故障	发生 MC 通用轻度故障时变为 TRUE。
Code	WORD	MC 通用 轻度故障代码	MC 通用 输出轻度故障的代码。与事件代码前四位的值相同。
Obsr	<code>_sMC_REF_EVENT</code>	MC 通用 监控信息	
Active	BOOL	正在产生 MC 通用监控信息	产生 MC 通用监控信息时变为 TRUE。
Code	WORD	MC 通用 监控信息代码	MC 通用 输出监视信息的代码。与事件代码前四位的值相同。

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

## 轴变量

轴变量是系统定义变量的名称为 `_MC_AX[0-255]` 或 `_MC1_AX[0-255]`、`_MC2_AX[0-255]`，数据类型为 `_sAXIS_REF` 型的结构体变量。

下面以 `_MC_AX[0-255]` 为例对轴变量的构成及其要素进行详细说明。`_MC1_AX[0-255]`、`_MC2_AX[0-255]` 也是如此。

变量名称	数据类型	名称	功能
<code>_MC_AX[0-255]</code>	<code>_sAXIS_REF</code>	轴变量	
Status	<code>_sAXIS_REF_STA</code>	轴状态	
Ready	BOOL	轴启动准备完成	轴启动准备就绪并在停止中的状态下变为 TRUE。 <code>_MC_AX[*].Status.Standstill</code> 表示与“TRUE: 停止中”的状态相同。
Disabled	BOOL	轴无效	在轴由于伺服 OFF 而停止的状态下变为 TRUE。 以下的轴状态互相排斥，任意一个单独变为 TRUE。 Disabled/ Standstill/ Discrete/ Continuous/ Synchronized/ Homing/ Stopping/ ErrorStop/ Coordinated
Standstill	BOOL	停止中	在轴由于伺服 ON 而停止的状态下变为 TRUE。
Discrete	BOOL	定位动作中	在向目标位置执行定位控制时变为 TRUE。 也包括定位动作中将超调值设定为“0”而使速度为“0”的状态。
Continuous	BOOL	连续动作中	没有目标位置连续动作时变为 TRUE。 速度控制或转矩控制时变为此状态。 也包括将目标速度设为“0”而使速度为“0”的状态、以及连续动作中将超调值设定为“0”而使速度为“0”的状态。
Synchronized	BOOL	同步动作中	执行同步控制时变为 TRUE。 也包括同步控制指令切换后的同步等待状态。
Homing	BOOL	原点复位中	通过 <code>MC_Home</code> 指令或 <code>MC_HomeWithParameter</code> 指令执行原点复位时变为 TRUE。
Stopping	BOOL	减速停止中	在通过 <code>MC_Stop</code> 指令、 <code>MC_TouchProbe</code> 指令停止轴动作之前的状态下变为 TRUE。 也包括通过 <code>MC_Stop</code> 指令停止轴动作后 <code>Execute</code> 变为 TRUE 时。 在减速停止中的状态下无法启动轴动作指令。 ( <code>CommandAborted = TRUE</code> )
ErrorStop	BOOL	错误减速停止中	执行 <code>MC_ImmediateStop</code> 指令或正在发生轻度故障 ( <code>_MC_AX[*].MFAultLvl.Active</code> 为“TRUE: 正在发生轴轻度故障”)时，轴停止前和停止后的状态。 该状态下无法启动轴动作指令。 ( <code>CommandAborted = TRUE</code> )
Coordinated	BOOL	多轴协调动作中	通过多轴协调指令启用轴组时变为 TRUE。

变量名称	数据类型	名称	功能
Details	_sAXIS_REF_DET	轴控制状态	表示指令的控制状态。
Idle	BOOL	停止中	除到位等待以外，未进行指令值运算时也变为 TRUE。 <sup>*1</sup> Idle 与 InPosWaiting 相互排斥，不会同时变为 TRUE。
InPosWaiting	BOOL	到位等待	到位等待时变为 TRUE。 通过需执行到位检查的定位来执行到位检查的状态。
Homed	BOOL	原点确定	确定原点时变为 TRUE。 <sup>*2</sup> 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变为 TRUE。 以下是 AND 条件。 · 原点确定状态 · 反馈当前位置以原点为中心进入“原点位置范围”时 在指令状态下轴动作过程中通过原点位置时也会变为 TRUE。
VelLimit <sup>*3</sup>	BOOL	指令速度饱和	同步控制中，指令速度以最高速度进行限制时变为 TRUE。
Dir	_sAXIS_REF_DIR	指令方向状态	表示指令指定的移动方向。
Posi	BOOL	正方向指令指定中	指令指定为正方向时变为 TRUE。
Nega	BOOL	负方向指令指定中	指令指定为负方向时变为 TRUE。
DrvStatus	_sAXIS_REF_STA_DRV	伺服驱动器状态	表示驱动器的状态。
ServoOn	BOOL	伺服 ON	伺服电机通电时变为 TRUE。
Ready	BOOL	伺服准备就绪	伺服准备就绪时变为 TRUE。 <sup>*4</sup>
MainPower	BOOL	主电路电源	伺服驱动器的主电路电源通电时变为 TRUE。
P_OT	BOOL	正方向极限输入	启用正方向极限输入时变为 TRUE。
N_OT	BOOL	负方向极限输入	启用负方向极限输入时变为 TRUE。
HomeSw	BOOL	近原点输入	启用近原点输入时变为 TRUE。
Home	BOOL	原点输入	启用原点输入时变为 TRUE。 <sup>*5*6</sup>
ImdStop	BOOL	立即停止输入	启用立即停止输入时变为 TRUE。
Latch1	BOOL	外部锁定输入 1	启用锁定输入 1 时变为 TRUE。
Latch2	BOOL	外部锁定输入 2	启用锁定输入 2 时变为 TRUE。
DrvAlarm	BOOL	驱动器错误输入	发生驱动器错误时变为 TRUE。
DrvWarning	BOOL	驱动器警告输入	发生驱动器警告时变为 TRUE。
ILA	BOOL	驱动器内部功能限制中	通过伺服驱动器的限制功能进行实际限制时变为 TRUE。 <sup>*7</sup>
CSP	BOOL	周期同步位置 (CSP) 模式中	伺服驱动器的伺服 ON，在 CSP 模式下变为 TRUE。 <sup>*8</sup>
CSV	BOOL	周期同步速度 (CSV) 模式中	伺服驱动器的伺服 ON，在 CSV 模式下变为 TRUE。 <sup>*6</sup>
CST	BOOL	周期同步转矩 (CST) 模式中	伺服驱动器的伺服 ON，在 CST 模式下变为 TRUE。 <sup>*6</sup>

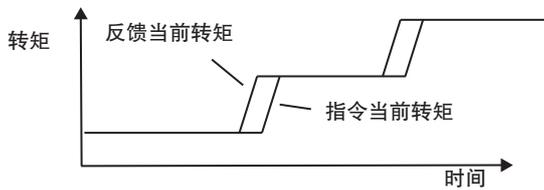
变量名称	数据类型	名称	功能
Cmd	_sAXIS_REF_CMD_DATA	轴指令值	
Pos	LREAL	指令当前位置	输出指令位置的当前值。(单位: 指令单位) 伺服 OFF 或非位置控制模式时, 输出反馈当前位置。 <sup>*9</sup>
Vel	LREAL	指令当前速度	输出指令速度的当前值。(单位: 指令单位/s) 正方向移动时附带正号、负方向移动时附带负号。 根据指令当前位置之差算出的速度。伺服 OFF 或非位置控制模式时, 根据反馈当前位置计算出的速度。
AccDec	LREAL	指令当前加减速速度	输出指令加减速度的当前值。(单位: 指令单位/s <sup>2</sup> ) 根据指令当前速度之差算出的加减速速度。加速时附带正号, 减速时附带负号。 执行中的指令的加减速速度为“0”时变为“0”。
Jerk	LREAL	指令当前跃度	输出指令跃度的当前值。(指令单位/s <sup>3</sup> ) 加减速度的绝对值增大时附带正号, 减小时附带负号。 执行中的指令的加减速速度或指令跃度为“0”时变为“0”。
Trq	LREAL	指令当前转矩	输出指令转矩的当前值。(单位: %) 正方向移动时附带正号、负方向移动时附带负号。 非转矩控制模式时, 输出与反馈当前转矩相同的值。 <sup>*10</sup>
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) <sup>*9</sup>
Vel	LREAL	反馈当前速度	输出反馈速度的当前值。(单位: 指令单位/s) 正方向移动时附带正号、负方向移动时附带负号。
Trq	LREAL	反馈当前转矩	输出反馈转矩的当前值。(单位: %) 正方向移动时附带正号、负方向移动时附带负号。
TimeStamp <sup>*11</sup>	ULINT	时间戳	输出轴当前位置的更新时刻。运行变化时刻获取功能的轴有效。(单位: ns)
MFaultLvl	_sMC_REF_EVENT	轴轻度故障	
Active	BOOL	正在发生轴轻度故障	正在发生轴轻度故障时变为 TRUE。
Code	WORD	轴轻度故障代码	输出轴轻度故障的代码。 与事件代码前四位值相同。
Obsr	_sMC_REF_EVENT	轴监控信息	
Active	BOOL	正在产生轴监控信息	产生轴监控信息时变为 TRUE。
Code	WORD	轴监控信息代码	输出轴监视信息的代码。 与事件代码前四位值相同。

变量名称	数据类型	名称	功能
Cfg	_sAXIS_REF_CFG	轴基本设定	表示轴基本设定内容。
AxNo	UINT	轴号	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。 虚拟轴无需输入输出接线。 0: _mcServo(伺服轴) 1: _mcEnclr(编码器轴) 2: _mcVirServo(虚拟伺服轴) 3: _mcVirEnclr(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: 分配至原始恒定周期任务 2: 分配至固定周期任务 (执行优先级 5)
Scale	_sAXIS_REF_SCALE	单位转换	表示电子齿轮比的设定值。
Num	UDINT	电机转 1 圈的脉冲数	表示指令位置电机转 1 圈的脉冲数。 根据电子齿轮比将指令值转换为脉冲量。
Den	LREAL	电机转 1 圈的移动量	表示指令位置电机转 1 圈的工件移动量。*14
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( $\mu$ m) 3: _mcNm(nm) 4: _mcDeg(degree) 5: _mcInch(inch)
CountMode*13	_eMC_UNITS	计数模式	表示计数模式。 0: _mcCountModeLinear(线性模式) 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、同步控制中以及多轴协调动作中也属于正在运算的状态。
- \*2. 即使为 TRUE，以下情况仍需重新确定原点。  
变更位置计数设定或单位转换设定时。  
在伺服驱动器侧发生异常或进行操作导致丢失绝对位置信息时。例如，编码器电缆断线或绝对值编码器数据清零等。
- \*3. VelLimit 请仅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”其中之一，且主电路电源 (voltage enabled) ON 时，为 TRUE。但是 [主电路电源 OFF 检测] 设定为 [不检测] 时，将无视主电路电源的 ON/OFF。PDS 状态、主电路电源 OFF 检测的详情请参阅  “A-5 PDS 状态切换 (P.A-26)”。
- \*5. 通过 Sysmac Studio 的 [轴基本设置] 画面中的 [高级设置] 设定为 [数字输入] 的 [编码器 Z 相检测] 的信号的状态。使用其他公司生产的驱动器可能无法将此信号映射至 PDO。请确认所连接的驱动器的手册。
- \*6. 欧姆龙制伺服驱动器 G5 系列 EtherCAT 通信内置直线电机型无法将该信号映射至 PDO。
- \*7. 表示映射至 PDO 的“状态字 (6041H)”的位 11(启用内部限制功能) 的状态。变为 TRUE 的条件取决于伺服驱动器的规格。请参阅所连接的伺服驱动器的使用手册。欧姆龙制伺服驱动器 1S 系列及 G5 系列为转矩限制 / 速度限制 / 驱动禁止输入 / 软件限制的其中之一。
- \*8. 根据映射至 PDO 的“状态字 (6061h)”的值进行显示。CSP、CSV、CST 变为 TRUE 的条件取决于伺服驱动器的规格。请参阅所连接的伺服驱动器的使用手册。  
如“操作模式显示 (6061Hex)”未映射至 PDO，不同版本 CPU 单元的显示如下所示。  
对于 Ver.1.09 以下版本的 CPU 单元，始终为 FALSE。  
对于 Ver.1.10 以上版本的 CPU 单元，如映射至 PDO 的“状态字 (6041Hex)”的状态为 Operation Enabled，则为 TRUE。
- \*9. 如分配给轴的 EtherCAT 从站或 NX 单元与 CPU 单元之间尚未确立过程数据通信，不同版本 CPU 单元的显示如下所示。  
对于 Ver.1.09 以下版本的 CPU 单元，轴变量的反馈当前位置及指令当前位置为“0”或下限值。下限值的条件是计数模式为旋转模式时位置范围中不含“0”。  
对于 Ver.1.10 以上版本的 CPU 单元，轴变量的反馈当前位置及指令当前位置为尚未确立过程数据通信前的反馈当前位置。

- \*10.通过数据跟踪对指令当前转矩和反馈当前转矩进行比较显示时，在非转矩控制模式下，显示为指令当前转矩在 1 个任务周期后发生变化。这是由于受到了数据跟踪处理时间的影响。数据跟踪的处理时间请参阅 □ “NJ/NX 系列 CPU 单元用户手册 软件篇 (SBCA-359)” 中数据跟踪的记述。

数据跟踪画面示例



如图所示，指令当前转矩的变化迟于反馈当前转矩。

- \*11.Ver.1.06 以上的 CPU 单元和 Ver.1.07 以上的 Sysmac Studio 组合时可使用。  
 \*12.对于 NX 系列 位置接口单元，表示安装了位置接口单元的 EtherCAT 耦合器单元的节点地址。  
 \*13.Ver.1.10 以上的 CPU 单元和 Ver.1.12 以上的 Sysmac Studio 组合时可使用。  
 \*14.Ver.1.11 以上版本的 CPU 单元新增的单位转换设定中，减速机使用设定为“使用”时，该参数无效。确认反之有效的参数“工件侧 1 圈的移动量”、“工件侧齿轮比”、“电机侧齿轮比”的值时，请使用 MC\_ReadAxisParameter(轴参数读取)指令。  
 \*15.计数模式为线性模式时，表示溢出前的位置。为旋转模式时，表示环计数器上限值。  
 \*16.计数模式为线性模式时，表示下溢出前的位置。为旋转模式时，表示环计数器下限值。

### ● 轴变量与轴种类的关系

启用或不启用的轴变量的结构要素因轴种类而异。查看不启用的结构要素时，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]` 也是如此。此外，下表为按轴种类记载启用或禁用各结构要素的一览表。

○：启用； -：不启用

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
<code>_MC_AX[0-255]</code>	<code>_sAXIS_REF</code>	轴变量				
Status	<code>_sAXIS_REF_STA</code>	轴状态				
Ready	BOOL	轴启动准备完成	○	○	-	-
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	○	○	-	-

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
Details	_sAXIS_REF_DET	轴控制状态				
Idle	BOOL	停止中	○	○	○	○
InPosWaiting	BOOL	到位等待	○	○	-	-
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	正方向指令指定中	○	○	-	-
Nega	BOOL	负方向指令指定中	○	○	-	-
DrvStatus	_sAXIS_REF_STA_DRV	伺服驱动器状态				
ServoOn	BOOL	伺服 ON	○	-	-	-
Ready	BOOL	伺服准备就绪	○	-	-	-
MainPower	BOOL	主电路电源	○	-	-	-
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	即停输入	○	-	-	-
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	○	-	-	-
DrvWarning	BOOL	驱动器警告输入	○	-	-	-
ILA	BOOL	驱动器内部功能限制中	○	-	-	-
CSP	BOOL	周期同步位置 (CSP) 模式中	○	-	-	-
CSV	BOOL	周期同步速度 (CSV) 模式中	○	-	-	-
CST	BOOL	周期同步转矩 (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	○	○	-	-
Jerk	LREAL	指令当前跃度	○	○	-	-
Trq	LREAL	指令当前转矩	○	○	-	-
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	反馈当前转矩	○	○	-	-
TimeStamp	ULINT	时间戳	○	-	○	-
MFaultLvl	_sMC_REF_EVENT	轴轻度故障				
Active	BOOL	正在发生轴轻度故障	○	○	○	○
Code	WORD	轴轻度故障代码	○	○	○	○
Obsr	_sMC_REF_EVENT	轴监控信息				
Active	BOOL	正在产生轴监控信息	○	○	○	○
Code	WORD	轴监控信息代码	○	○	○	○

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
Cfg	_sAXIS_REF_CFG	轴基本设定				
AxNo	UINT	轴号	○	○	○	○
AxEnable	_eMC_AXIS_USE	轴使用	○	○	○	○
AxType	_eMC_AXIS_TYPE	轴种类	○	○	○	○
NodeAddress	UINT	节点地址	○	-	○	-
ExecID <sup>*1</sup>	UINT	执行 ID	○	○	○	○
Scale	_sAXIS_REF_SCALE	单位转换				
Num	UDINT	电机转 1 圈的脉冲数	○	○	○	○
Den <sup>*2</sup>	LREAL	电机转 1 圈的移动量	○	○	○	○
Units	_eMC_UNITS	显示单位	○	○	○	○
CountMode <sup>*1</sup>	_eMC_COUNT_MODE	计数模式	○	○	○	○
MaxPos <sup>*1</sup>	LREAL	当前位置上限值	○	○	○	○
MinPos <sup>*1</sup>	LREAL	当前位置下限值	○	○	○	○

\*1. Ver.1.10 以上的 CPU 单元和 Ver.1.12 以上的 Sysmac Studio 组合时可使用。

\*2. Ver.1.11 以上版本的 CPU 单元新增的单位转换设定中，减速机使用设定为“使用”时，该参数无效。

## 轴组变量

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

下面以 `_MC_GRP[0-63]` 为例对轴组变量的构成及其要素进行详细说明。

`_MC1_GRP[0-63]`、`_MC2_GRP[0-63]` 也是如此。

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

变量名称	数据类型	名称	功能
<code>_MC_GRP[0-63]</code>	<code>_sGROUP_REF</code>	轴组变量	
Status	<code>_sGROUP_REF_STA</code>	轴组状态	
Ready	BOOL	启动准备完成	轴组停止、启动准备已经完成时变为 TRUE。 以下的 AND 条件为启动准备完成的条件。 · 构成轴未执行 <code>MC_Stop</code> 指令。 · <code>_MC_GRP[*].Status.Standby</code> 为 TRUE (停止中) · 构成轴为伺服 ON 状态 · 构成轴的 <code>_MC_AX[*].Details.Homed</code> 为 TRUE (原点确定)
Disabled	BOOL	不启用轴组	因轴组无效而停止时变为 TRUE。 以下的轴组状态互相排斥，任意一个单独变为 TRUE。 <code>Disabled/ Standby/ Moving/ Stopping/ ErrorStop</code>
Standby	BOOL	停止中	轴组动作指令停止时变为 TRUE。 与轴组构成轴的伺服 ON/OFF 状态无关。
Moving	BOOL	动作中	在向目标位置执行轴组动作指令时变为 TRUE。 也包括到位等待的状态、以及因超调而使速度为“0”的状态。
Stopping	BOOL	减速停止中	在通过 <code>MC_GroupStop</code> 指令停止轴组动作之前的状态下变为 TRUE。 也包括通过 <code>MC_GroupStop</code> 指令停止轴动作后 <code>Execute</code> 变为 TRUE 时。 该状态下无法启动轴组动作指令。 ( <code>CommandAborted=TRUE</code> )
ErrorStop	BOOL	错误减速停止中	执行 <code>MC_GroupImmediateStop</code> 指令或发生轴组轻度故障 ( <code>_MC_GRP[*].MFAultLvl.Active=TRUE</code> ) 时，轴组停止前或停止后变为 TRUE。该状态下无法启动轴组动作指令。 ( <code>CommandAborted=TRUE</code> )
Details	<code>_sGROUP_REF_DET</code>	轴组控制状态	表示指令的控制状态。
Idle	BOOL	停止中	除了到位等待，在未进行指令值运算时变为 TRUE。 <sup>*1</sup> Idle 与 <code>InPosWaiting</code> 相互排斥，不会同时变为 TRUE。
InPosWaiting	BOOL	到位等待	任意一个构成轴到位等待时变为 TRUE。 通过需执行到位检查的定位来执行到位检查的状态。 <sup>*2</sup>

变量名称	数据类型	名称	功能
Cmd	_sGROUP_REF_CMD_D ATA	轴组指令值	
	Vel	LREAL	指令插补速度 输出指令插补速度的当前值。 根据插补指令当前位置之差算出的插补速度。 正方向移动时附带正号、负方向移动时附带负号。 不启用轴组时变为“0”。
	AccDec	LREAL	指令插补加减速度 输出指令插补加减速度的当前值。 根据指令插补速度之差算出的插补加减速度。 加速时附带正号，减速时附带负号。 不启用轴组或正在执行的轴组动作指令的指令 加减速度指定为“0”时变为“0”。
MFAultLvl	_sMC_REF_EVENT	轴组 轻度故障	
	Active	BOOL	正在发生轴组 轻度故障 正在发生轴组轻度故障时变为 TRUE。
	Code	WORD	轴组 轻度故障代码 输出轴组轻度故障的代码。 与事件代码前四位的值相同。
Obsr	_sMC_REF_EVENT	轴组 监控信息	
	Active	BOOL	正在产生轴组监控信息 发生轴组监控信息时变为 TRUE。
	Code	WORD	轴组 监控信息代码 输出轴组监控信息的代码。 与事件代码前四位的值相同。
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 表示各分配任务的执行 ID。 0: 未分配任务 (未创建组) 1: 分配至原始恒定周期任务 2: 分配至固定周期任务 (执行优先度 5)
Kinematics	_sGROUP_REF_KIM	运动学转换设定	表示轴组的运动学转换定义。
	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) 表示为轴 A0 分配的轴号。
	Axis[1]	UINT	构成轴 (轴 A1) 表示为轴 A1 分配的轴号。
	Axis[2]	UINT	构成轴 (轴 A2) 表示为轴 A2 分配的轴号。
	Axis[3]	UINT	构成轴 (轴 A3) 表示为轴 A3 分配的轴号。

\*1. 以速度“0”动作也属于正在运算的状态。

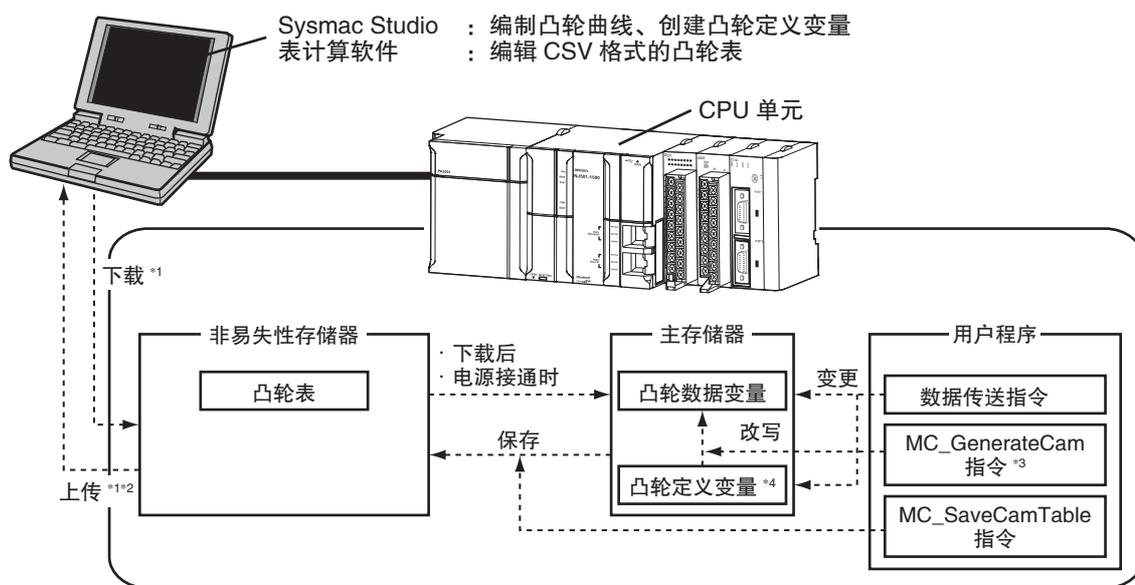
\*2. 轴组内的所有构成轴控制在各轴参数的到位宽度的范围内时，变为 FALSE。

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

## 6-7 凸轮表与凸轮数据变量

MC 功能模块中可通过 Sysmac Studio 的凸轮编辑器功能绘制的凸轮曲线用作凸轮表。  
NJ/NX 系列的程序中将凸轮表的数据作为凸轮数据变量进行处理。

### 凸轮表的创建和保存的结构



- \*1 “下载”和“上传”使用 Sysmac Studio 的“同步”功能。
- \*2 无法通过凸轮编辑器功能编辑上传的凸轮数据变量。使用计算机重新编辑由用户程序编辑的凸轮数据变量的方法请参阅下一页。  
通过 Sysmac Studio 重新编连、编辑凸轮曲线时，以凸轮曲线的数据覆盖上传的凸轮数据变量。
- \*3 Ver.1.08 以上版本的 CPU 单元和 Ver.1.09 以上版本的 Sysmac Studio 组合时，可使用 MC\_GenerateCamTable(凸轮表生成)指令。
- \*4 凸轮定义变量是凸轮属性变量和凸轮节点变量的总称。

### ● 凸轮表的数据流

- 将通过 Sysmac Studio 绘制的凸轮曲线下载至 CPU 单元时，在 CPU 单元内置的非易失性存储器中保存为凸轮表。将凸轮表上传至 Sysmac Studio 时，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在下载后或接通电源时，作为凸轮数据变量反映至主存储器。
- 可通过用户程序编辑主存储器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，请参阅 □ “9-2-5 凸轮表 (P.9-15)”。
- 用户程序中的 MC\_GenerateCamTable(凸轮表生成)指令可根据凸轮定义变量的值改写主存储器的凸轮数据变量。
- 运动控制指令的 MC\_SaveCamTable(保存凸轮表)指令将主存储器的凸轮数据变量保存至非易失性存储器。
- 在 Sysmac Studio 的凸轮数据设定中创建的凸轮定义变量，即使在用户程序中进行了更改仍可上传。作为用户定义变量创建的凸轮定义变量，如在用户程序中进行了更改将无法上传。
- 关于通过 Sysmac Studio 创建和传送凸轮定义变量，请参阅 □ “Sysmac Studio Version 1 操作手册 (SBCA-362K 以上)”。
- 关于 MC\_GenerateCamTable(凸轮表生成)指令及 MC\_SaveCamTable(凸轮表保存)指令的详情，请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)”。
- 无论 CPU 单元的模式和 MC 功能模块的状态如何，均可执行凸轮表的上传和下载。但保存凸轮表时，无法上传或下载凸轮数据，也无法执行在线开始操作、在线编辑及数据跟踪开始的操作等。执行在线编辑等时，也无法使用 MC\_SaveCamTable(保存凸轮表)指令。
- 开始下载时，动作中的轴以最大减速度减速。



### 使用注意事项

- 重新接通电源或从 Sysmac Studio 下载，可将通过用户程序变更的凸轮数据恢复为非易失性存储器中的凸轮表的内容。无法通过 Sysmac Studio 上传此类数据。



### 版本相关信息

组合 Ver.1.13 以上版本的 CPU 单元和 Ver.1.17 以上版本的 Sysmac Studio，即使在下载期间仍可持续向 I/O 设备发送指令。

Ver.1.12 以下版本的 CPU 单元，则在下载期间停止向 I/O 设备发送指令。

开始下载时停止 / 持续向 I/O 设备发送指令的设定，请参阅“NJ/NX 系列 CPU 单元用户手册 软件篇 (SBCA-359)”。

#### 停止向 I/O 设备发送指令

下载期间，伺服 OFF、轴状态为“Disable(轴无效)”。

#### 持续向 I/O 设备发送指令

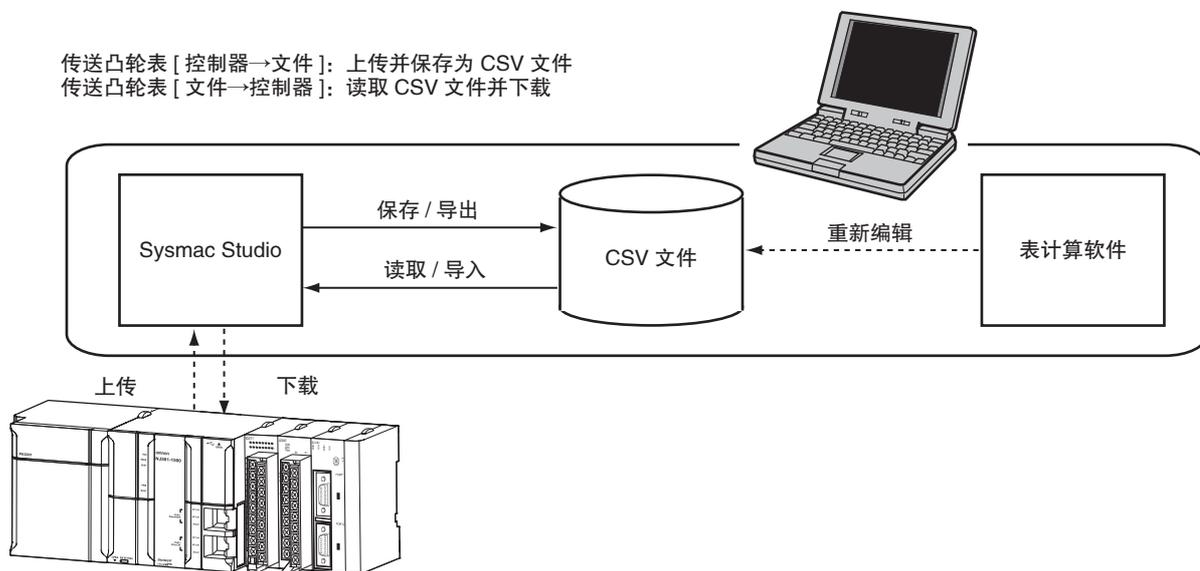
MC\_Power(可运行)指令及 MC\_SetTorqueLimit(转矩限制)指令在下载期间仍可运行。

下载期间，将持续下载前通过 MC\_Power 指令实现的伺服 ON 状态、及通过 MC\_SetTorqueLimit 指令进行的输出转矩限制。

即使通过下载更新用户程序、删除 MC\_Power 指令及 MC\_SetTorqueLimit 指令，仍将持续伺服 ON 状态及转矩限制。

## 使用计算机重新编辑由用户程序编辑的凸轮数据变量

通过用户程序编辑凸轮数据变量，再通过 MC\_SaveCamTable(保存凸轮表)指令保存至非易失性存储器的凸轮表无法通过 Sysmac Studio 的凸轮编辑器功能重新编辑数据。下面对通过表计算软件等重新编辑数据并用作凸轮表的方法进行说明。



\*1 “下载”和“上传”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 ] 中右击需重新编辑的凸轮曲线，从菜单中选择 [ 传送凸轮表 [ 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，点击 [ 保存 ] 按钮。

### ● 重新编辑 CSV 文件

- 使用表计算软件等编辑数据，以 CSV 文件格式保存。

## ● 将 CSV 文件传送至 CPU 单元

- 右击需下载的凸轮曲线，从菜单中选择 [ 传送凸轮表 [ 文件→控制器 ] ]。
- 在 [ 打开文件 ] 对话框中，指定需传送的文件名，点击 [OK] 按钮。
- 请在传送完成后执行控制器复位或重新接通控制器电源，以启用传送的凸轮表。



### 使用注意事项

- 执行传送凸轮表 [ 文件→控制器 ] 前，请与控制器同步。
- 通过传送凸轮表 [ 文件→控制器 ] 传送凸轮表后，通过同步功能将凸轮数据设定传送至控制器时，变更为凸轮数据设定的数据。  
此时，请再次通过传送凸轮表 [ 文件→控制器 ] 进行传送或将凸轮数据设定设为非同步对象。

也可将通过凸轮编辑器编制的凸轮数据设定导出为 CSV 文件。

凸轮数据设定和操作方法的详情请参阅 “Sysmac Studio Version 1 操作手册 (SBCA-362)”。

## 凸轮曲线的名称

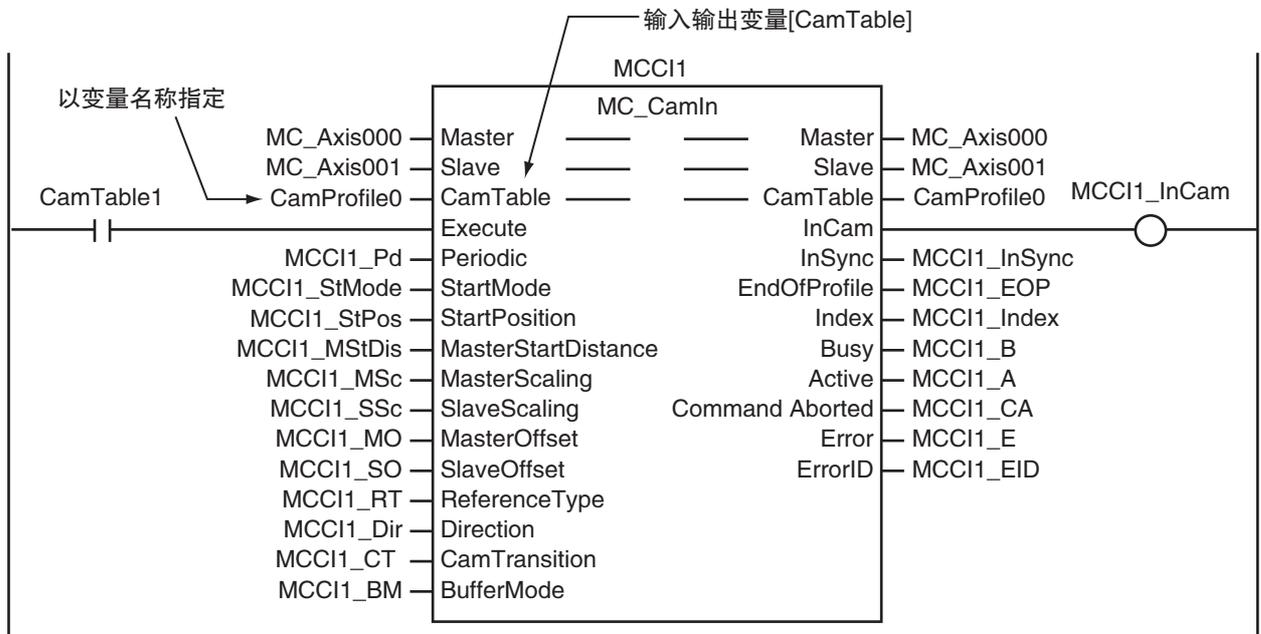
通过 Sysmac Studio 添加凸轮曲线时，默认名称设定为 “CamProfile0”。添加凸轮曲线时，需在末尾数字中加上后缀。

可通过 Sysmac Studio 将各凸轮曲线的名称变更为任意的名称。

此处设定的凸轮曲线名称作为凸轮表名称进行处理。

## 用户程序的指定方法

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



## 6-8 运动控制程序的编写方法

可通过在 NJ/NX 系列的用户程序中记述运动控制指令来执行运动控制。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，任意形式的 CPU 单元均可执行最多 256 轴的设定和编程。但无法将超过 CPU 形式最多控制轴数的项目下载至 CPU 单元。
- 沿用项目等时，请确认 CPU 单元的形式，确保不超过最多控制轴数。
- 轴使用为 [ 未使用轴 ] 的轴也是控制轴数的对象。

编程的详情请参阅 □ “NJ/NX 系列 CPU 单元 用户手册 软件篇 (SBCA-359)”。

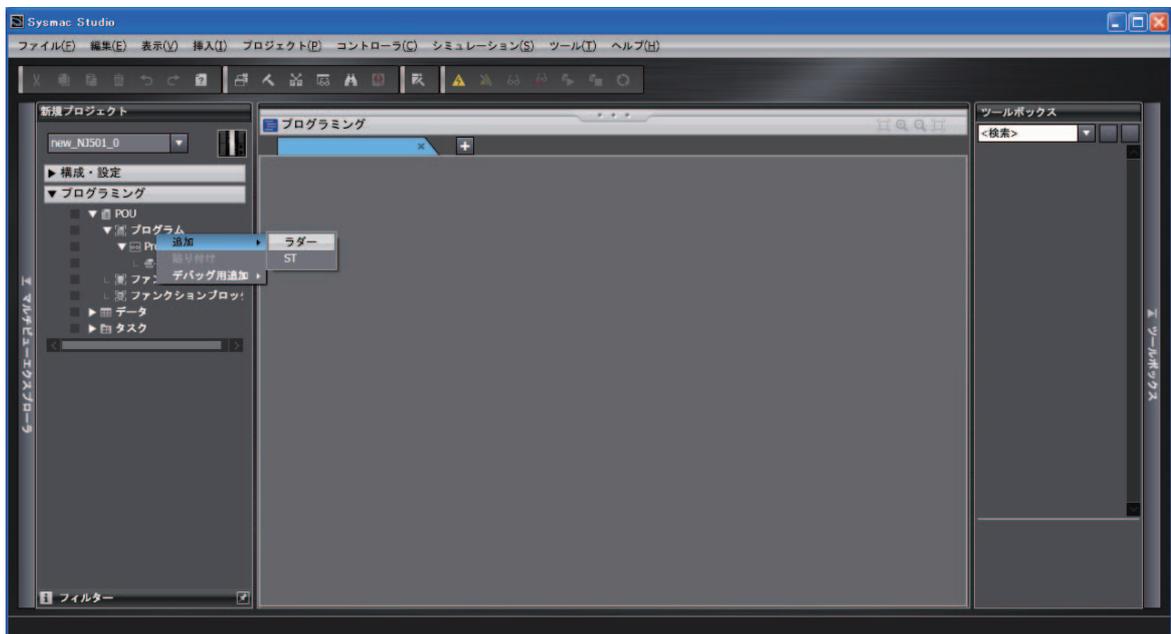
下面对使用 Sysmac Studio 并向现有项目添加程序的步骤进行说明。

### 1 Sysmac Studio 的启动

启动 Sysmac Studio，打开现有项目。

### 2 程序的添加

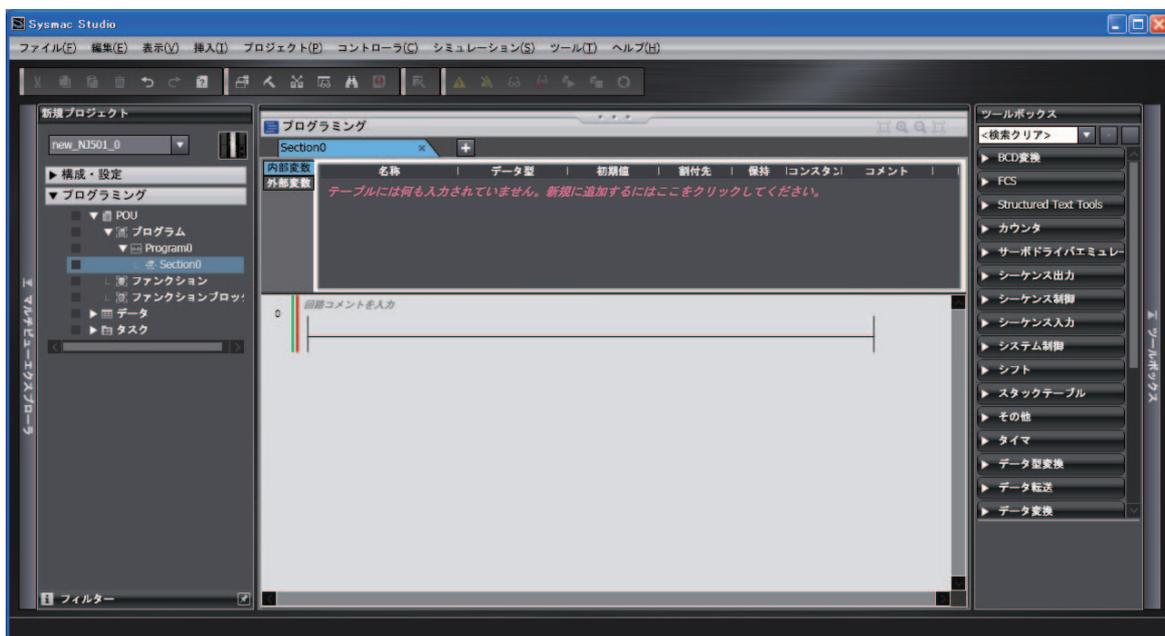
右击多视图浏览器中的 [ 程序 ]，再点击 [ 添加 ] 中的 [ 梯形图 ] 或 [ ST ]。



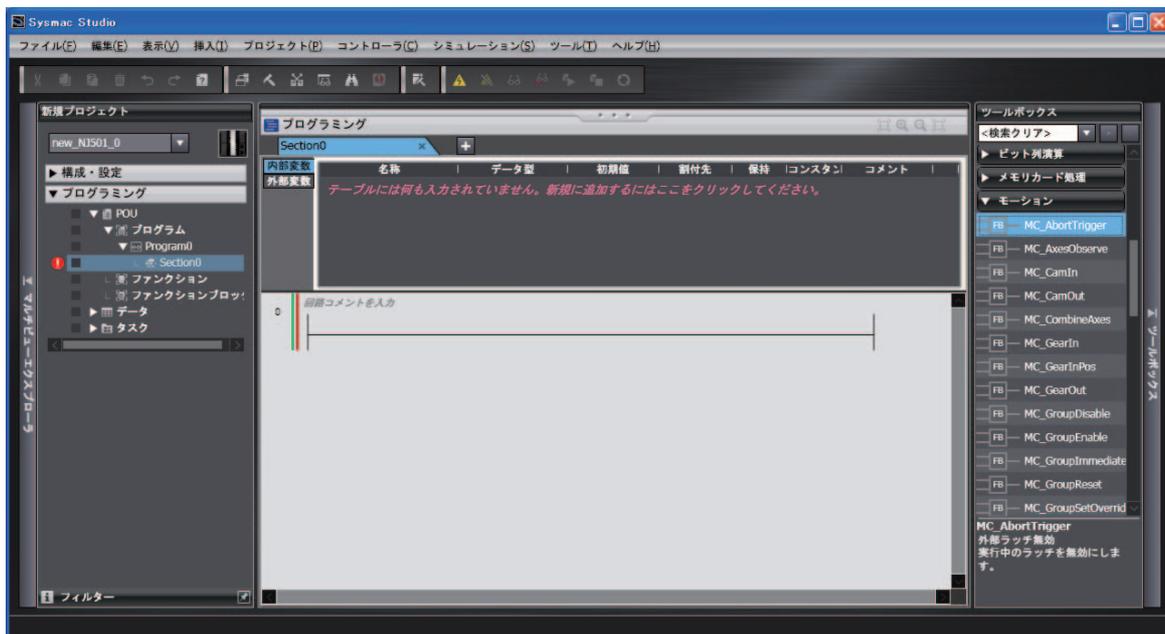
在多视图浏览器中添加程序。

### 3 程序的编辑

右击添加的程序下方的区域，再点击编辑，打开程序编辑画面。



请从工具箱选择指令，编写所需的程序。



编程的详情请参阅 [□](#) “NJ/NX 系列 CPU 单元 用户手册 软件篇 (SBCA-359)”。

操作方法的详情请参阅 [□](#) “Sysmac Studio Version 1 操作手册 (SBCA-362)”。

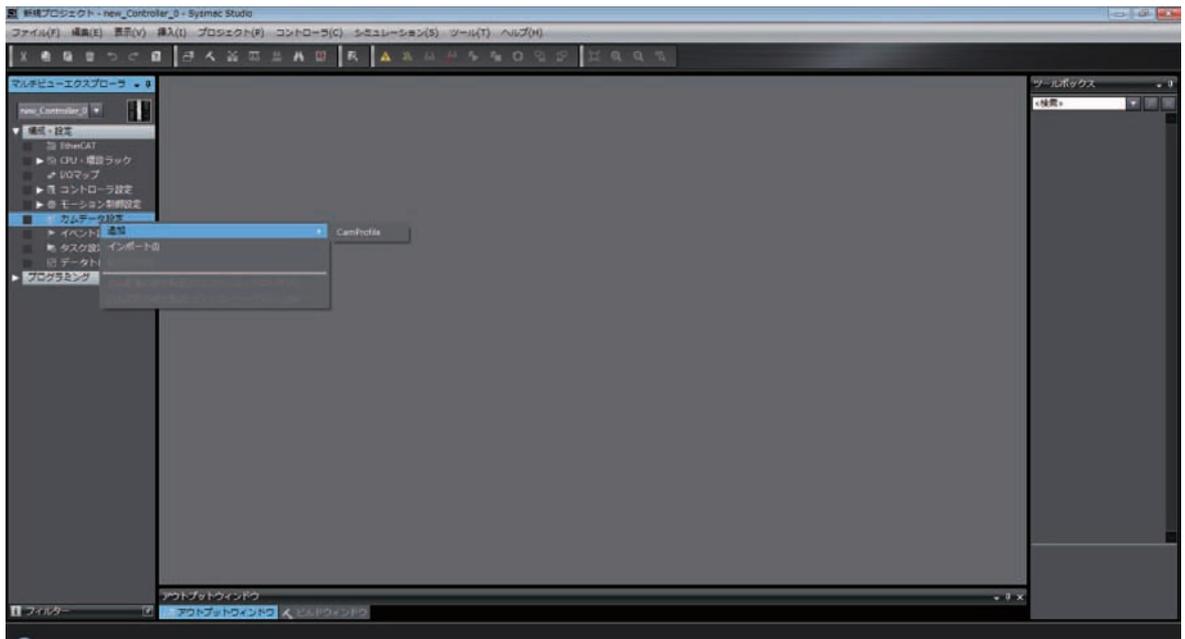
## 6-9 凸轮表的创建方法

下面对使用 Sysmac Studio 中的凸轮编辑器功能来创建凸轮表的步骤进行说明。

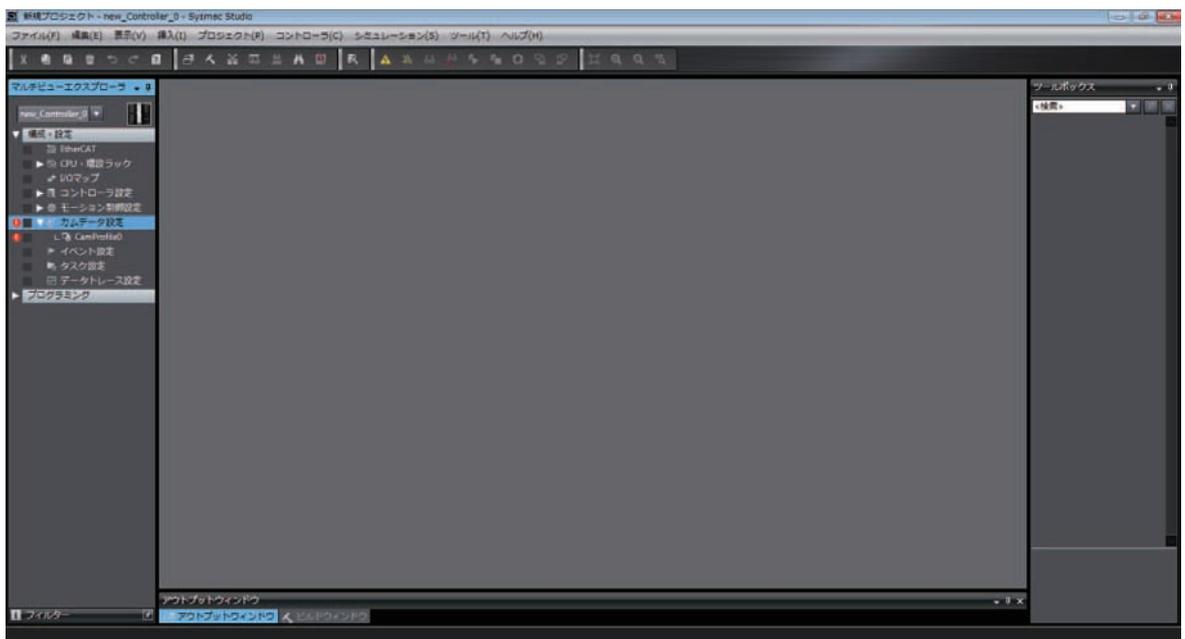
凸轮编辑器功能的详情请参阅  “Sysmac Studio Version 1 操作手册 (SBCA-362)”。

### 1 凸轮曲线的添加

右击多视图浏览器中的 [ 凸轮数据设定 ], 再点击 [ 追加 ] 中的 [ CamProfile ]。

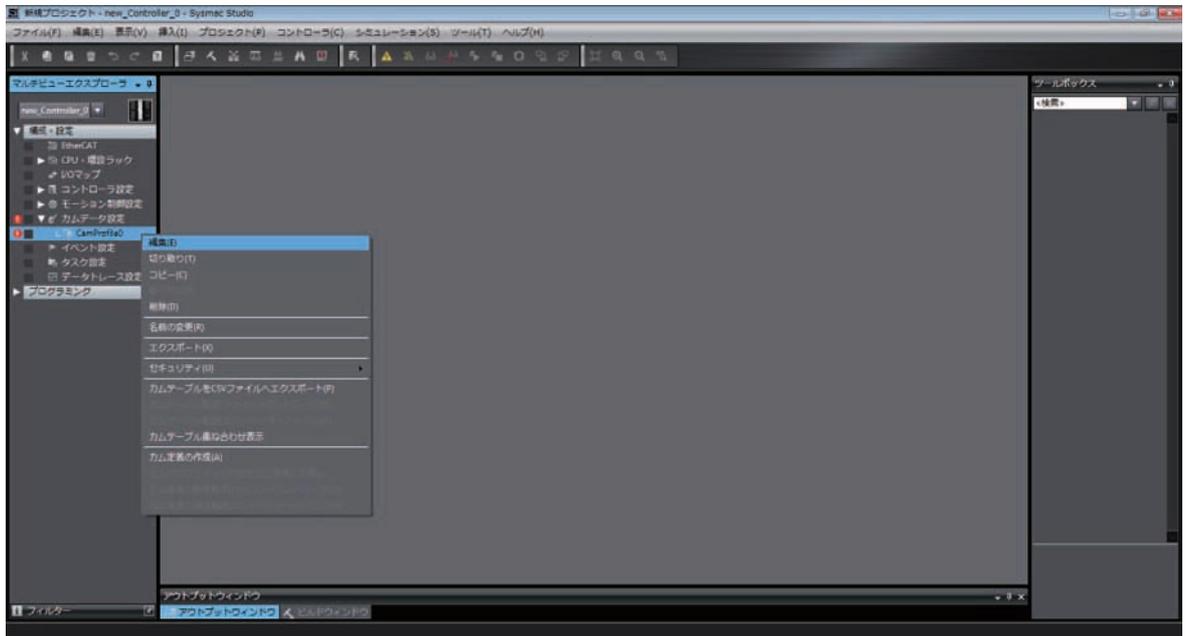


在多视图浏览器中添加凸轮曲线。可将添加的凸轮曲线的名称从默认的 “CamProfile0” 变更为任意的名称。

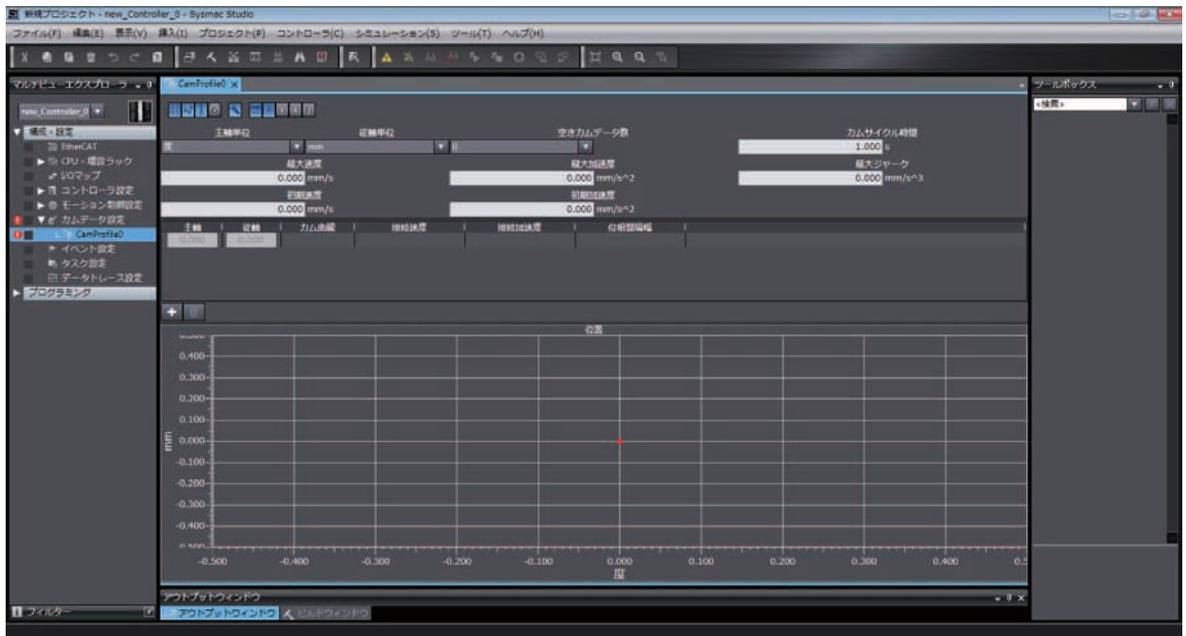


## 2 凸轮曲线的编辑

右击多视图浏览器中添加的凸轮曲线，再点击 [ 编辑 ]。



显示凸轮曲线的编辑画面。



请设定各项目，编辑凸轮曲线。

操作方法的详情请参阅 □ “Sysmac Studio Version 1 操作手册 (SBCA-362)”。



# 7

## 手动运行

下面对与欧姆龙制 1S 系列伺服驱动器组合使用时用于手动操作的功能进行说明。

---

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 轴参数的设定 .....	7-4
7-2-3 程序示例 .....	7-4
7-3 微动移动 .....	7-5
7-3-1 微动移动的步骤 .....	7-5
7-3-2 轴参数的设定 .....	7-6
7-3-3 输入变量的设定示例 .....	7-6
7-3-4 程序示例 .....	7-7

## 7-1 概要

---

下面对组合使用 MC 功能模块和欧姆龙制 1S 系列伺服驱动器时，通过用户程序使用运动控制指令执行手动运行的情况进行说明。

手动运行的运动控制指令分为将伺服驱动器设为伺服 ON 状态的“MC\_Power(可运行)”指令和执行微动移动的“MC\_MoveJog(微动移动)”指令。



### 使用注意事项

---

执行手动运行时需设定轴。轴的设定方法请参阅 □□ “第 3 章 轴和轴组的设定”。

---



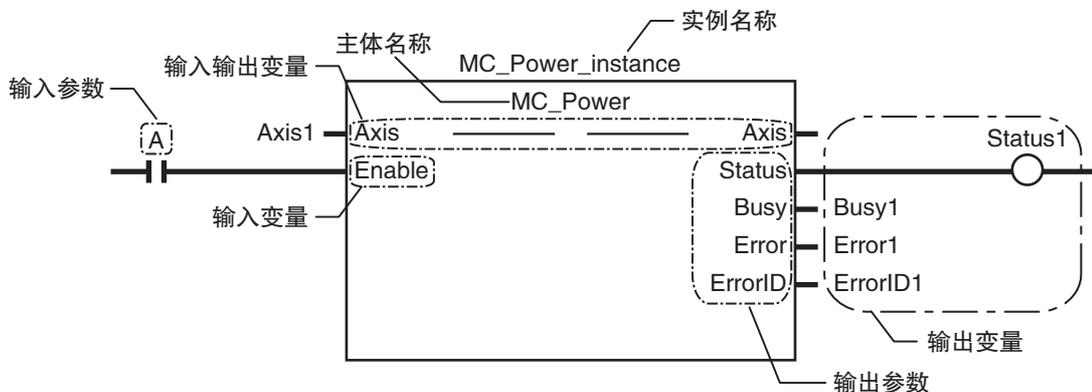
### 参考

---

- 需不编写程序即执行手动运行时，请使用 Sysmac Studio。  
使用 Sysmac Studio 执行手动运行请参阅 □□ “4-3 电机动作的确认 (P.4-8)”。
  - 用户程序的编写方法请参阅 □□ “第 6 章 运动控制程序”。
  - 使用欧姆龙制伺服驱动器 G5 系列的步骤和操作相同。
-

## 7-2 伺服 ON

将伺服驱动器相应的运行指令 ON/OFF 的功能。  
使用运动控制指令的 MC\_Power(可运行) 指令。



通过输入输出变量 Axis(轴) 指定需运行的轴。  
将 MC\_Power(可运行) 指令的输入变量 Enable(启用) 设为 TRUE 后, 变为 ON 状态。将 Enable(启用) 设为 FALSE 后, 变为伺服 OFF 状态。



### 使用注意事项

- 轴动作时将 Enable(启用) 设为 FALSE 后, 立即停止指令, 不会启用该轴相应的运动控制指令。
- 使用 NX 系列脉冲输出单元, 需另行对电机驱动器进行通电 ON/OFF。详情请参阅 □ “NX 系列位置接口单元用户手册 (SBCE-374)”。



### 参考

如果在欧姆龙制伺服电机 1S 系列及 G5 系列带绝对值编码器伺服电机或欧姆龙制电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型中使用绝对值型外部标尺, 则将 MC\_Power(可运行) 指令的输入变量 Enable(启用) 设为 TRUE 即可确定原点。



### 版本相关信息

对于 Ver.1.10 以上版本的 CPU 单元, 如果在欧姆龙制伺服电机 1S 系列及 G5 系列带绝对值编码器伺服电机或欧姆龙制电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型中使用绝对值型外部标尺, 则除上述“参考”外, 在 EtherCAT 过程数据通信从未确立状态切换至确立状态时, 也可确定原点。

## 7-2-1 伺服 ON 的步骤

- 1 轴的添加与设定**  
通过 Sysmac Studio 添加和设定轴。  
详情请参阅 □ “3-2-2 设定步骤 (P.3-9)”。
- 2 轴参数的设定**  
通过 Sysmac Studio 设定轴参数。  
详情请参阅 □ “3-2-2 设定步骤 (P.3-9)”。
- 3 用户程序的编写**  
通过 Sysmac Studio 编写程序。  
详情请参阅 □ “6-8 运动控制程序的编写方法 (P.6-32)”。
- 4 轴参数和用户程序的下载**  
将设定的轴参数和用户程序下载至 CPU 单元。  
下载至 CPU 单元时，使用 Sysmac Studio 的“同步”功能。  
详情请参阅 □ “3-2-2 设定步骤 (P.3-9)”。
- 5 用户程序的启动**  
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 轴参数的设定

仅设为伺服 ON 状态时，需设定的轴参数仅为以下设定项目。  
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

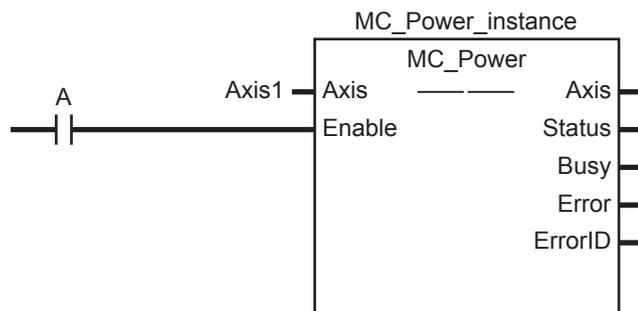
\*1 如果是多个轴，则按轴设定不同的变量名称。

\*2 如果是多个轴，则按轴设定不同的值。

\*3 请设定与伺服驱动器的节点地址相同的值。如果是多个轴，则按轴设定不同的值。

## 7-2-3 程序示例

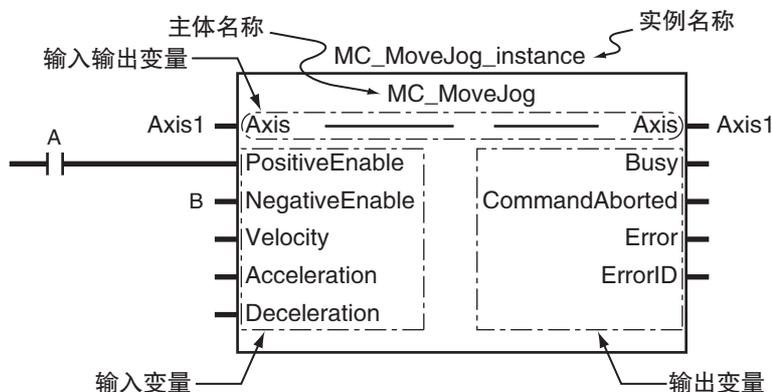
通过触点 A 切换轴 Axis1 的伺服 ON/OFF 的示例程序如下所示。



MC\_Power(可运行)指令的详情请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)”。

## 7-3 微动移动

微动移动使用运动控制指令中的 MC\_MoveJog(微动移动) 指令。



通过输入输出变量 Axis(轴) 指定需微动移动的轴。

将输入变量的 PositiveEnable 设为 TRUE 时, 以指定的 Velocity(目标速度)、Acceleration(加速度) 向正方向启动轴。将 PositiveEnable 设为 FALSE 时, 以指定的 Deceleration(减速度) 减速、停止。

同样地, 将输入变量 NegativeEnable 设为 TRUE 时, 向负方向启动, 变为 FALSE 时停止。

未确定原点的状态下也可执行微动移动。

### 7-3-1 微动移动的步骤

#### 1 轴的添加与设定

通过 Sysmac Studio 添加和设定轴。

详情请参阅 □□ “3-2-2 设定步骤 (P.3-9)”。

#### 2 轴参数的设定

通过 Sysmac Studio 设定轴参数。

详情请参阅 □□ “3-2-2 设定步骤 (P.3-9)”。

#### 3 用户程序的编写

通过 Sysmac Studio 编写程序。

详情请参阅 □□ “6-8 运动控制程序的编写方法 (P.6-32)”。

#### 4 轴参数和用户程序的下载

将设定的轴参数和用户程序下载至 CPU 单元。

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

详情请参阅 □□ “3-2-2 设定步骤 (P.3-9)”。

#### 5 用户程序的启动

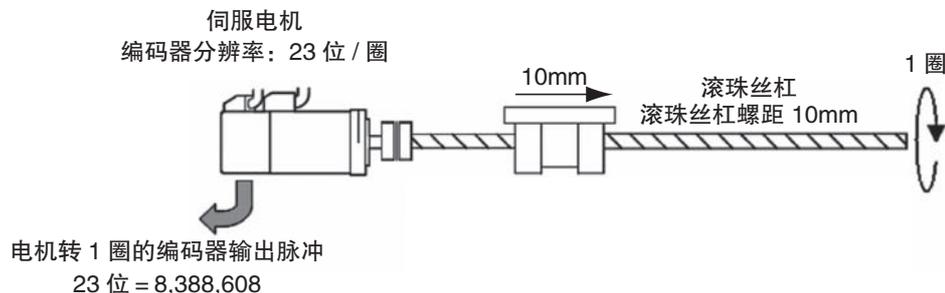
启动用户程序, 将 MC\_Power(可运行) 指令的输入变量“Enable”设为 TRUE, 将伺服驱动器设为伺服 ON 状态。

将 MC\_MoveJog(微动移动) 指令的输入变量“PositiveEnable”或“NegativeEnable”设为 TRUE, 执行微动移动。

### 7-3-2 轴参数的设定

在未确定原点的状态下执行微动移动时，请设定以下轴参数。

下面以 1 轴装置为例对设定例进行说明。



设定项目	设定内容
轴变量名称	Axis1 *1
轴号	1 *2
轴使用	使用轴
轴种类	伺服轴
输入设备 / 输出设备	1 *3
电机转 1 圈的脉冲数	8,388,608 *4
电机转 1 圈的移动量	10,000 *4
软件限制功能	对反馈位置有效
显示单位	$\mu\text{m}$
计数模式	线性模式
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. 如果是多个轴，则按轴设定不同的变量名称。

\*2. 如果是多个轴，则按轴设定不同的值。

\*3. 请设定与伺服驱动器的节点地址相同的值。如果是多个轴，则按轴设定不同的值。

\*4. 位置的指令单位为  $1(\mu\text{m})$ 。

\*5. 最高速度为  $3,000\text{r}/\text{min} = 30\text{m}/\text{min} = 0.5\text{m}/\text{s} = 500,000 \mu\text{m}/\text{s}$ 。

\*6. 微动最高速度为最高速度的 10% 即  $0.05\text{m}/\text{s} = 50,000 \mu\text{m}/\text{s}$ 。

\*7. 最大加速度和最大减速度为  $5\text{m}/\text{s}^2$ 。达到最高速度  $3,000\text{r}/\text{min}$  的加速时间设为 0.1s。

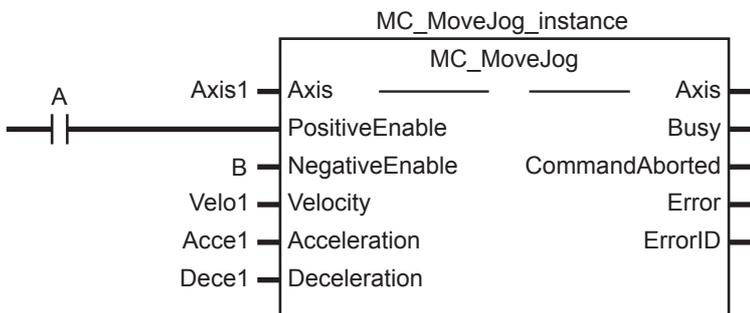
### 7-3-3 输入变量的设定示例

下面对“MC\_MoveJog(微动移动)”指令的输入变量“Velocity(目标速度)”、“Acceleration(加速度)”、“Deceleration(减速度)”的设定进行说明。

- 例如，以  $0.03\text{m}/\text{s}(30,000 \mu\text{m}/\text{s})$  的速度微动移动时，将“Velocity”指定为 30,000。
- 或以  $3\text{m}/\text{s}^2(3,000,000 \mu\text{m}/\text{s}^2)$  加减速时，将“Acceleration”和“Deceleration”指定为 3,000,000。

### 7-3-4 程序示例

通过触点 A 将轴 Axis1 向正方向微动移动和通过触点 B 向负方向微动移动时的示例程序如下所示。



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

以输入变量设定示例的条件运行时,请事先在各程序中设定各变量的值。

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

MC\_MoveJog(微动移动)指令的详情请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)”。



# 8

## 确定原点

本章对原点的确定进行说明。

---

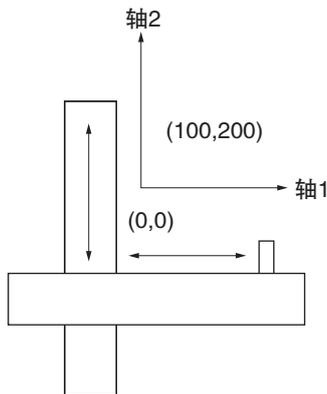
8-1 概要 .....	8-2
8-2 原点复位的步骤 .....	8-4
8-2-1 原点复位参数的设定 .....	8-5
8-2-2 原点复位的监控 .....	8-11
8-3 原点复位的动作 .....	8-12
8-4 绝对值编码器原点设定 .....	8-13
8-4-1 功能概要 .....	8-14
8-4-2 设定步骤 .....	8-15
8-5 高速原点复位 .....	8-16

## 8-1 概要

下面对组合使用 MC 功能模块和欧姆龙制 G5 伺服驱动器时的操作进行说明。

在需执行定位的系统中，对系统的绝对位置进行定位时，需先确定原点。

例如，在下图所示的 XY 滑台中，需定位至 (轴 1、轴 2) = (100mm, 200mm) 位置时，须确定机械原点 (0, 0)。确定该机械原点的过程称为确定原点。



在 MC 功能模块中，将运动控制指令中的“MC\_Home(原点复位)”指令或 MC\_HomeWithParameter(参数指定原点复位)指令用作确定机械原点的方法。

名称	内容
原点复位	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。 将接近传感器和编码器的 Z 相等用作原点输入信号。



### 使用注意事项

- 以下情况下，为未确定原点的状态。
  - MC\_SetPosition(变更当前位置)指令的执行
  - 线性模式下的溢出与下溢
  - 原点复位开始时
  - EtherCAT 通信控制未运行
- 使用 NX 系列位置接口单元时，部分功能受限。详情请参阅  “NX 系列位置接口单元用户手册 (SBCE-374)”。



### 参考

如果在欧姆龙制伺服电机 1S 系列及 G5 系列带绝对值编码器伺服电机或欧姆龙制电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型中使用绝对值型外部标尺，则将 MC\_Power(可运行)指令的输入变量 Enable(启用)设为 TRUE 即可确定原点。



### 版本相关信息

对于 Ver.1.10 以上版本的 CPU 单元，如果在欧姆龙制伺服电机 1S 系列及 G5 系列带绝对值编码器伺服电机或 G5 系列 EtherCAT 通信内置直线电机型中使用绝对值型外部标尺，则除前述“参考”外，在 EtherCAT 过程数据通信从未确立状态切换至确立状态时，也可确定原点。

定位至通过上述方法确定的原点的功能还含有以下 MC\_MoveZeroPosition( 高速原点复位 ) 指令。

名称	内容
高速原点复位	将绝对坐标“0”设为目标位置进行定位，返回原点。

在未确定原点的状态下，MC 功能模块也可驱动电机 ( 高速原点复位除外 )。

功能	动作
微动移动、速度控制	未确定原点时，接通电源后原点变为“0”。
高速原点复位	无法使用。 使用时发生指令异常。
定位	未确定原点时，接通电源后原点变为“0”。
中断标准定位	
凸轮动作开始	
齿轮动作开始	
梯形模式凸轮	
加减运算定位	
转矩控制	
区域监视	
直线插补	无法使用。
圆弧插补	使用时发生指令异常。



#### 参考

未确定原点时，不启用软件限制功能。

各操作或事件相应的原点确定状态的变化如下所示。

操作或事件	变化条件	原点确定状态的变化
伺服 ON 启用轴时	增量编码器	无变化
	通过绝对值编码器正常读取绝对值数据时	原点确定
	通过绝对值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

- 无论轴参数的 [ 原点复位动作 ] 的设定如何，均将 [ 预设原点 ] 作为原虚拟轴的原点。
- MC 功能模块将伺服驱动器的正转驱动禁止输入 (POT)、反转驱动禁止输入 (NOT)、近原点输入 (DEC) 用作正方向极限输入、负方向极限输入、近原点输入。  
对于此类输入信号，请确保伺服驱动器可检测的信号宽度并使其超过 MC 功能模块的控制周期。  
输入信号的宽度小于控制周期时，MC 功能模块无法检测输入信号且无法执行正常的动作。
- 需在伺服驱动器中设定各输入信号的参数。请参阅需连接的伺服驱动器的手册和附录，正确进行设定。

## 8-2 原点复位的步骤

---

下面对执行原点复位时的步骤进行说明。

### 1 轴的添加与设定

通过 Sysmac Studio 添加和设定轴。

### 2 轴参数的设定

通过原点复位参数设定原点复位方法。

### 3 用户程序的编写

通过 Sysmac Studio 编写用户程序。

详情请参阅 □ “6-8 运动控制程序的编写方法 (P.6-32)”。

### 4 轴参数和用户程序的下载

将设定的轴参数和用户程序下载至 CPU 单元。

下载至 CPU 单元时，使用 Sysmac Studio 的“同步”功能。

详情请参阅 □ “3-2-2 设定步骤 (P.3-9)”。

### 5 用户程序的启动

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。

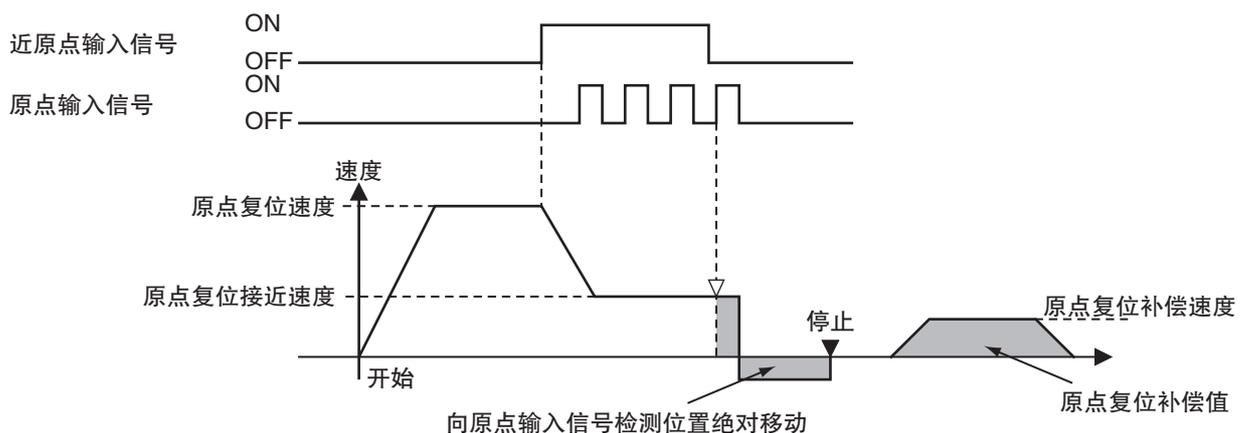
将 MC\_Home(原点复位)指令或 MC\_HomeWithParameter(参数指定原点复位)指令的输入参数“Execute”设为 TRUE，执行原点复位。

## 8-2-1 原点复位参数的设定

通过原点复位参数的设定值指定原点复位的方法。

通过 Sysmac Studio 设定原点复位参数。

设定	概要
原点复位动作	选择原点复位的移动方法。
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	设定启动原点复位时的开始方向。
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	设定原点复位中正方向极限输入“ON”时的停止方法。
负方向极限输入时动作	设定原点复位中负方向极限输入“ON”时的停止方法。
原点复位速度	设定原点复位中的速度。(单位: 指令单位/s)
原点复位接近速度	设定近原点输入 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。 设定为“0”时, 无需加速即可达到原点复位速度。 (单位: 指令单位/s <sup>2</sup> )
原点复位减速度	指定原点复位时的减速度。 设定为“0”时, 无需减速即可达到原点速度。 (单位: 指令单位/s <sup>2</sup> )
原点复位跃度	指定原点复位时的跃度。 设定为“0”时无跃度。(单位: 指令单位/s <sup>3</sup> )
原点输入屏蔽距离	对原点复位动作模式指定为附近避让、原点输入屏蔽距离时原点输入的屏蔽距离进行设定。(单位: 指令单位)
原点位置偏置	原点复位完成时, 按照设定值预设当前位置。 (单位: 指令单位)
原点复位接触时间	以毫秒为单位, 对原点复位动作模式指定为附近避让、接触时间时的接触时间进行设定。(单位: ms)
原点复位补偿值	设定原点复位的原点确定后的原点复位补偿量。(单位: 指令单位)
原点复位补偿速度	设定原点复位补偿时的速度。(单位: 指令单位/s)



下面对各原点复位参数进行说明。

## 原点复位动作

作为确定机械原点的动作，可以从 10 种动作中选择。

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

各原点复位动作使用的原点复位参数如下表所示。

(○：使用参数、×：不使用参数)

原点复位动作	原点复位参数														
	原点输入信号	原点复位开始方向	原点检测方向	正方向极限输入时动作	负方向极限输入时动作	原点复位速度	原点复位接近速度	原点复位加速度	原点复位减速度	原点复位跃度	原点输入屏蔽距离	原点位置偏置	原点复位接触时间	原点复位补偿值	原点复位补偿速度
指定为附近避让、近原点输入 OFF	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
指定为附近避让、近原点输入 ON	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
指定为近原点输入 OFF	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
指定为近原点输入 ON	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
指定为极限输入 OFF	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
指定为附近避让、原点输入屏蔽距离	○	○	○	○	○	○	○	○	○	○	○	○	×	○	○
仅极限输入	×	○	○	○	○	○	○	○	○	○	×	○	×	○	○
指定为附近避让、接触时间	×	○	○	○	○	○	○	○	○	○	×	○	○	○	○
指定为无近原点输入、接触原点输入	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
原点预设	×	×	×	×	×	×	×	×	×	×	×	○	×	×	×

各原点复位动作的详情请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)”。



### 使用注意事项

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

## 原点输入信号

在需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用于确定原点的信号。



### 使用注意事项

仅连接欧姆龙制 1S 系列或 G5 系列时可通过该设定选择原点输入信号。

## 原点复位开始方向

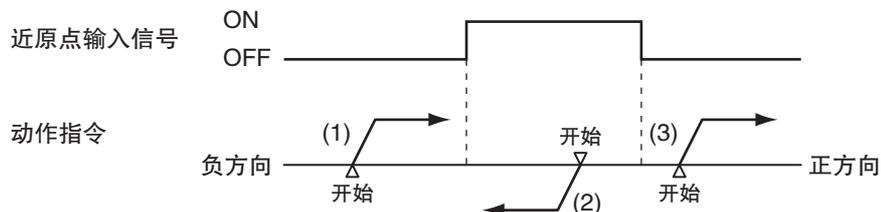
从正方向和负方向中选择启动原点复位时开始移动的方向。

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

包含基于附近避让的反转动作的原点复位动作模式有如下 4 种。

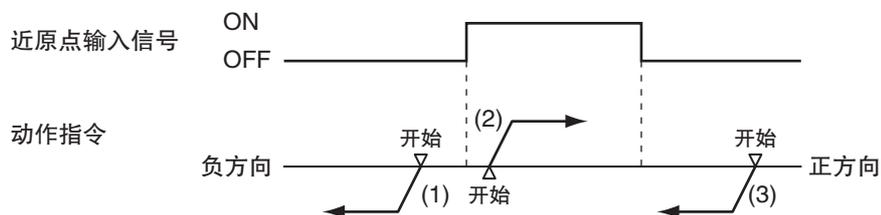
- 0: 指定为附近避让、近原点输入 OFF
- 1: 指定为附近避让、近原点输入 ON
- 9: 指定为附近避让、原点输入屏蔽距离
- 12: 指定为附近避让、接触时间

原点复位开始方向：正  
原点检测方向：正



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

原点复位开始方向：负  
原点检测方向：负



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

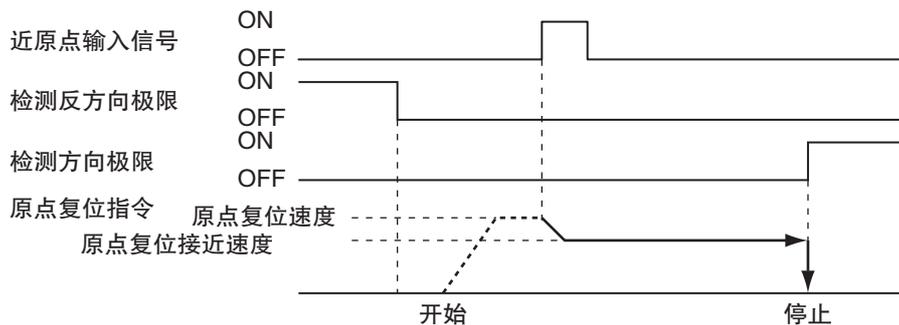
## 原点检测方向

从正方向和负方向中选择检测原点的方向。

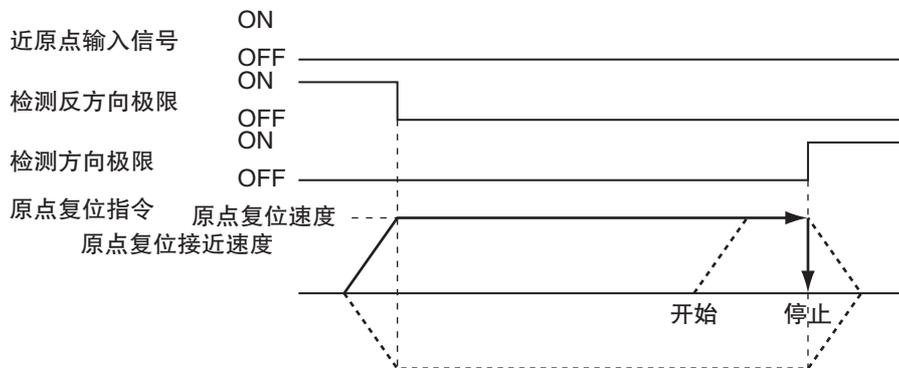
原点检测方法与启动原点复位时开始移动的方向之间的关系请参阅 □ “原点复位开始方向 (P.8-7)”。

## 正方向极限输入时动作 / 负方向极限输入时动作

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



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



## 原点复位速度

以指令单位 /s 设定原点复位中的速度。

## 原点复位接近速度

以指令单位 /s 设定近原点输入 “ON” 后的速度。

## 原点复位加速度

以指令单位 /s<sup>2</sup> 设定原点复位时的速度。

设为 “0” 时，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<sup>2</sup> 设定原点复位时的减速度。

设为 “0” 时，不减速达到原点复位接近速度等目标速度。

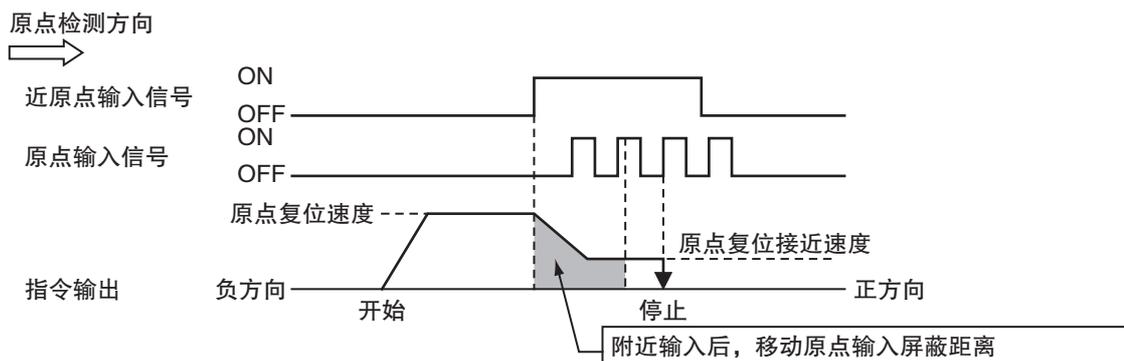
## 原点复位跃度

以指令单位 /s<sup>3</sup> 设定原点复位时的跃度。

设定为 “0” 时，变为无跃度的加减速。

## 原点输入屏蔽距离

原点复位动作为 [9: 指定为附近避让、原点输入屏蔽距离] 时，按指令单位设定原点输入的屏蔽距离。近原点输入信号从 “OFF” 变为 “ON” 后减速开始时的距离。



## 原点位置偏置

确定机械原点并设定原点复位补偿值后，原点复位补偿运行完成时，将设定的值预设为当前值。

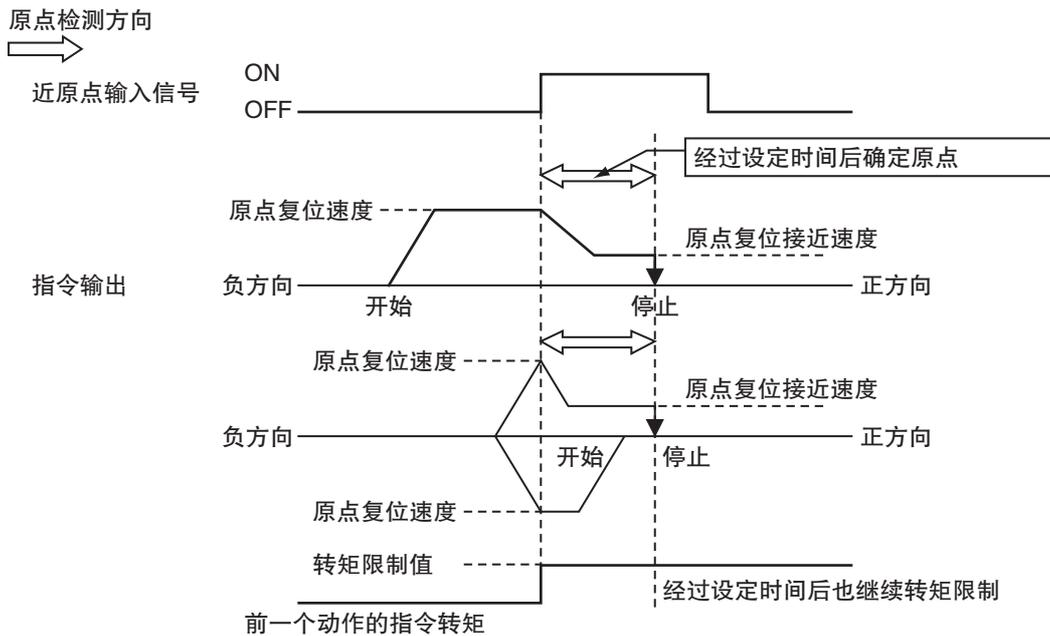
不将机械原点设为“0”而将其设为任意值进行使用的功能。

对于绝对值编码器系统，同时自动计算绝对值编码器原点位置偏置值，保存至CPU单元的电池备份存储器。

## 原点复位接触时间

以毫秒为单位，设定原点复位动作“指定为12：附近避让、接触时间”时的接触时间。

近原点输入信号从“OFF”变为“ON”后减速开始时的时间。



## 原点复位补偿值

确定机械原点后，按设定的值执行相对值定位，补偿机械原点。此时的移动速度为原点复位补偿速度。

确定机械原点后难以调整工件。如需微调，请将首次确定的机械原点的偏移量设为补偿数据。

用于难以重新设置近原点传感器或更换电机后机械原点错位的场合时，便于操作。

## 原点复位补偿速度

对设定了原点复位补偿值的速度，以指令单位/s进行设定。

## 8-2-2 原点复位的监控

通过用户程序读取轴变量后，可监控原点复位状态和输入信号状态。

另外，变量名称以 `_MC_AX[*]` 为例进行说明，`_MC1_AX[*]`、`_MC2_AX[*]` 同理。

变量名称	数据类型	名称	功能
<code>_MC_AX[0-255].Status.Homing</code>	BOOL	原点复位中	通过 <code>MC_Home</code> 指令或 <code>MC_HomeWithParameter</code> 指令执行原点复位时变为 TRUE。
<code>_MC_AX[0-255].Dtails.Homed</code>	BOOL	原点确定	确定原点时变为 TRUE。 FALSE: 原点未确定 TRUE: 原点确定
<code>_MC_AX[0-255].Dtails.InHome</code>	BOOL	原点停止	在 原点位置的范围内时变为 TRUE。 以下是 AND 条件。 · 原点确定状态 · 反馈当前位置以原点为中心进入“原点位置范围”时 在指令状态下轴动作过程中通过原点位置时也会变为 TRUE。
<code>_MC_AX[0-255].DrvStatus.P_OT</code>	BOOL	正方向极限输入	启用正方向极限输入时变为 TRUE。
<code>_MC_AX[0-255].DrvStatus.N_OT</code>	BOOL	负方向极限输入	启用负方向极限输入时变为 TRUE。
<code>_MC_AX[0-255].DrvStatus.HomeSw</code>	BOOL	近原点输入	启用近原点输入时变为 TRUE。
<code>_MC_AX[0-255].DrvStatus.Home</code>	BOOL	原点输入	启用原点输入时变为 TRUE。

## 8-3 原点复位的动作

执行原点复位时，需根据定位系统的构成和用途选择机械原点的确定方法。

MC 功能模块中含有 10 种原点复位动作模式。

确定原点后，还可通过原点复位补偿值进行微调。



### 参考

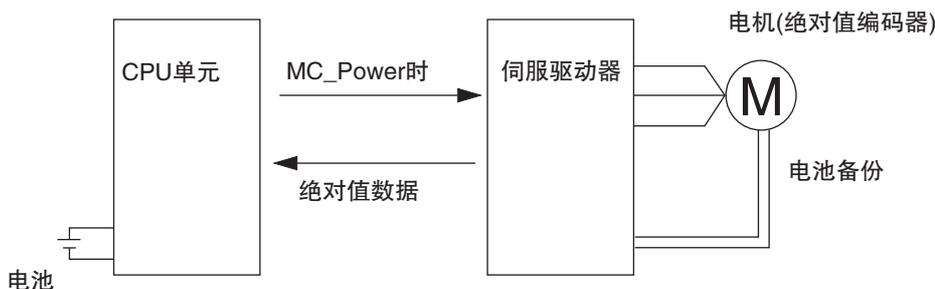
- 模式是否合适取决于定位系统的构成和用途。  
线性模式（有限长度）下，在具备近原点传感器、正方向极限输入、负方向极限输入的机械系统中，一般选择“指定为附近避让、近原点输入 OFF 的动作”。
- 执行原点复位时不启用超调。
- 仅确定原点或原点复位补偿移动完成时，根据到位检查的设定执行到位检查。
- 执行原点复位过程中执行使用了其它运动控制指令的多重启动时，无法执行等待和合并。

原点复位的详情请参阅 “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)”。

## 8-4 绝对值编码器原点设定

下面对使用 G5 系列伺服驱动器 EtherCAT 通信内置型的绝对值编码器时的情形进行说明。

切断 CPU 单元电源后，绝对值编码器仍会通过编码器内的电池备份保存绝对值数据。因此，执行 MC\_Power(可运行)指令时，通过绝对值编码器读取当前位置，以确定位置。确定原点后，无需像增量编码器那样进行原点复位操作。



### 使用注意事项

- 使用欧姆龙制伺服驱动器 1S 系列时，请连接 CPU 单元的电池。
- 欧姆龙制伺服驱动器 G5 系列使用绝对值编码器时，请连接 CPU 单元的电池和伺服驱动器的绝对值编码器备份用电池。
- 因初次使用绝对值编码器、更换电机、使用欧姆龙制伺服驱动器 G5 系列时绝对值编码器的电池耗尽等而丢失绝对值数据时，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。之后，将 MC\_Power(可运行)指令的输入变量 Enable(启用)设为 TRUE，即可确定原点。对于 Ver.1.10 以上版本的 CPU 单元，除前述外，在 EtherCAT 过程数据通信从未确立状态切换至确立状态时，也可确定原点。
- 如果在 CPU 单元的电池异常时接通电源，会发生“绝对值编码器原点位置偏置读取异常(事件代码：1460000Hex)。此时，如果通过“ResetMCErrror(MC 所有错误复位)”指令解除异常，则可设为伺服 ON。

Ver.1.09 以下版本的 CPU 单元，在异常解除后伺服 ON 时，[绝对值编码器原点位置偏置]的值为“0”，确定为原点。

Ver.1.10 以上版本的 CPU 单元，在异常解除后直至执行“MC\_Home(原点复位)”指令或“MC\_HomeWithParameter(参数指定原点复位)”指令确定原点前，保持未确定原点的状态。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，以便在正确的位置执行动作。

切断 CPU 单元的电源后会变为未确定原点的状态。



### 参考

使用欧姆龙制伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型时也可设定绝对值编码器的原点。使用直线电机时，请注意以下几点并阅读本节的说明。

- 直线电机不使用“编码器”，而使用具有相同功能的“外部标尺”。
- 本节说明中的“绝对值编码器”可改称为“绝对值型外部标尺”。
- 绝对值型外部标尺不含绝对值编码器中的旋转量数据。因此，绝对值编码器中无需进行安装。也不使用保存旋转量数据的电池。
- 直线电机型规格的详情请参阅 □ “AC 伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型用户手册(SBCE-366)”。

### 8-4-1 功能概要

绝对值编码器系统的原点设定如下，即执行 MC\_Power(可运行)指令时执行绝对值编码器的偏置补偿，确定原点。

对于 Ver.1.10 以上版本的 CPU 单元，除前述外，在 EtherCAT 过程数据通信从未确立状态切换至确立状态时，也可确定原点。

也可与增量编码器相同，通过原点复位确定原点。此时，将确定原点后的指令位置与通过绝对值编码器读取的绝对值数据之差值作为“绝对值编码器原点位置偏置”导入 CPU 单元的电池备份存储器。

通过 MC\_Home(原点复位)指令或 MC\_HomeWithParameter(参数指定原点复位)指令执行当前位置预设后，也可将确定原点后的指令位置与绝对值数据之差作为偏置导入 [绝对值编码器原点位置偏置]。在 MC 功能模块中，已将 [绝对值编码器原点位置偏置] 自动保存至 CPU 单元内的电池备份存储器，因此无需在 Sysmac Studio 等中进行保存。



#### 使用注意事项

- [绝对值编码器原点位置偏置]与轴号建立关联，按轴分别保存到电池备份存储器中。变更轴号时，保存的偏置量将消失。变更轴号时，请再次执行原点复位。
- 更换 CPU 单元或 CPU 单元的电池时，请在更换前先确定原点，再备份电池备份存储器。通过以上操作备份 [绝对值编码器原点位置偏置]。
- 更换后执行恢复，可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阅 □□ “Sysmac Studio Version 1 操作手册 (SBCA-362)”。

## 对象伺服电机

可使用绝对值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	IS 系列	R88D-ISN □□□ -ECT	R88M-1 □□□□□ S R88M-1 □□□□□ T R88M-1 □□□□□ C
	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 系列脉冲输出单元无法使用绝对值编码器。

## 与伺服驱动器之间的连接

□□ 请根据 “NJ/NX 系列 CPU 单元内置 EtherCAT 端口 用户手册 (SBCD-358)” 正确进行连接。

## 8-4-2 设定步骤

下面对绝对值编码器系统的原点设定步骤进行说明。

### 1 绝对值编码器的安装

安装方法请参阅  “Sysmac Studio Version 1 操作手册 (SBCA-362)”。

### 2 轴参数的设定

请在 MC 功能模块的轴参数中的 [ 位置计数设定 ] 的 [ 编码器种类 ] 中选择 [1: 绝对值编码器 (ABS)]。详情请参阅 “5-2-7 位置计数设定 (P.5-21)”。

### 3 原点复位的执行

执行原点复位。

可通过 MC 功能模块的轴参数中的 [ 原点复位设定 ] 的 [ 原点复位动作 ] 选择原点确定方法。

确定原点后，将指令位置与通过绝对值编码器读取的绝对值数据之差作为 [ 绝对值编码器原点位置偏置 ] 自动保存至电池备份存储器。

## 绝对值编码器的安装

初次使用绝对值编码器、需将旋转量数据初始化为“0”或不连接电池和绝对值编码器并长期放置等时，需安装绝对值编码器。

安装方法的详情请参阅  “Sysmac Studio Version 1 操作手册 (SBCA-362)”。



### 使用注意事项

安装绝对值编码器后，需重新接通欧姆龙制 1S 系列及 G5 系列伺服驱动器的电源。完成绝对值编码器的安装处理时，伺服驱动器中会发生绝对值清除错误 (A27.1)。关闭伺服驱动器的控制电源后，重新接通电源，解除该异常，完成绝对值编码器的安装步骤。

## 旋转模式下绝对值编码器的使用

在轴参数的 [ 选择计数模式 ] 中设定为 [ 旋转模式 ] 时，当前位置在 [ 环形计数器上限设定值 ] 和 [ 环形计数器下限设定值 ] 的范围内变为环形计数器。

使用绝对值编码器并设定为旋转模式时，MC 功能模块会在每个运动控制周期内自动计算并更新 [ 绝对值编码器原点位置偏置 ]。

更新的 [ 绝对值编码器原点位置偏置 ] 在 CPU 单元的电源 OFF 时会自动保存至 CPU 单元内置的电池备份存储器。

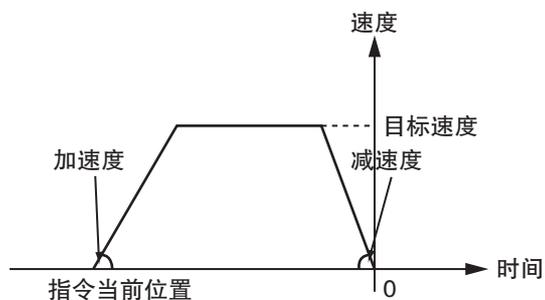
通过以上操作，CPU 单元和伺服驱动器的电源 OFF 后，在下次接通电源时仍可通过绝对值编码器复原旋转轴中的当前位置。

## 8-5 高速原点复位

在确定原点的状态下高速定位至原点位置的功能。

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。

如果在未确定原点的状态下执行, 则指令会发生异常。



### 参考

MC\_MoveZeroPosition( 高速原点复位 ) 指令的详情请参阅 “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)”。

# 9

## 运动控制功能

本章对连接欧姆龙制 1S 系列伺服驱动器 (EtherCAT 通信内置型) 时的运动控制功能进行说明。

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 中断标准定位	9-5
9-1-5 周期同步定位	9-6
9-1-6 停止	9-6
9-1-7 超调	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	9-22
9-2-7 加减运算定位	9-23
9-2-8 主轴相位补偿	9-24
9-2-9 从轴位置补偿	9-25
9-2-10 多任务运动中实现同步控制的方法	9-25
9-3 单轴速度控制	9-27
9-3-1 速度控制	9-27
9-3-2 周期同步速度控制	9-28
9-4 单轴转矩控制	9-29
9-5 单轴控制的通用功能	9-30
9-5-1 位置	9-30
9-5-2 速度	9-32
9-5-3 加速与减速	9-33
9-5-4 跃度	9-35
9-5-5 指定动作方向	9-36
9-5-6 运动控制指令的重启	9-39
9-5-7 运动控制指令的多重启动 (缓存模式)	9-44

9-6 多轴协调控制 .....	9-49
9-6-1 动作概要 .....	9-49
9-6-2 直线插补 .....	9-52
9-6-3 圆弧插补 .....	9-53
9-6-4 轴组周期同步位置控制 .....	9-53
9-6-5 多轴协调控制中的停止 .....	9-54
9-6-6 多轴协调控制中的超调 .....	9-56
9-7 多轴协调控制的通用功能 .....	9-57
9-7-1 多轴协调控制中的速度 .....	9-57
9-7-2 多轴协调控制中的加速和减速 .....	9-58
9-7-3 多轴协调控制中的跃度 .....	9-59
9-7-4 多轴协调控制的运动控制指令的重启 .....	9-60
9-7-5 多轴协调控制的运动控制指令的多重启动(缓存模式) .....	9-61
9-8 其他功能 .....	9-68
9-8-1 当前位置变更功能 .....	9-68
9-8-2 转矩限制功能 .....	9-69
9-8-3 锁定功能 .....	9-69
9-8-4 区域功能 .....	9-70
9-8-5 软件限制功能 .....	9-71
9-8-6 位置偏差监控功能 .....	9-72
9-8-7 偏差计数器复位 .....	9-73
9-8-8 轴间偏差监控功能 .....	9-73
9-8-9 到位检查 .....	9-74
9-8-10 轴使用变更功能 .....	9-76
9-8-11 数字凸轮开关有效功能 .....	9-76
9-8-12 任意坐标系的 3D 运动监视显示 .....	9-77

## 9-1 单轴位置控制

MC 功能模块与欧姆龙制 1S 系列及 G5 系列 EtherCAT 通信内置型伺服驱动器连接后，可进行位置控制、速度控制、转矩控制。

下面对单轴的定位动作进行说明。

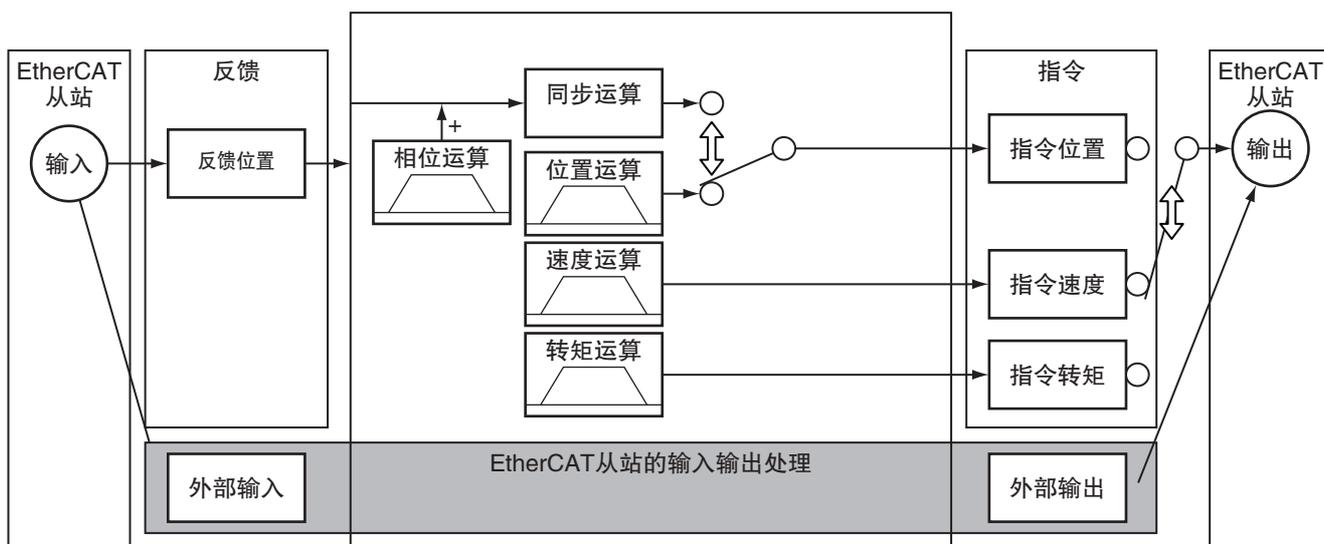
使用 NX 系列 脉冲输出单元时，运动控制功能中的部分功能存在差别。详情请参阅 □ “NX 系列 位置接口单元 用户手册 (SBCE-374)”。

### 9-1-1 动作概要

MC 功能模块的单轴控制功能包括指定动作曲线的控制和同步控制。

在需指定动作曲线的控制中，可通过位置控制、速度控制、转矩控制的 3 种控制模式进行指定。

在同步控制中，控制对象的从轴与主轴之间具有同步关系，以凸轮曲线和齿轮比等表示，同时进行动作。还支持微动移动等手动运行和原点复位。



(注) 同步运算中可将指令位置或反馈位置作为输入进行使用。

### 轴错误复位

轴发生异常时，排除异常原因后通过 MC\_Reset(轴错误复位) 指令解除异常。

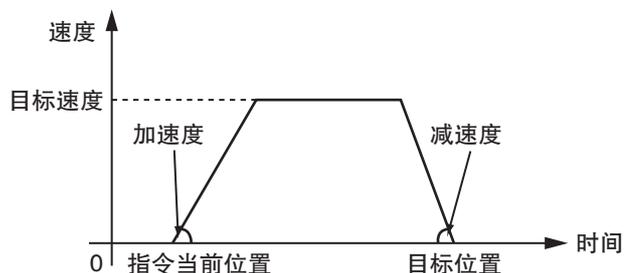
轴错误复位的详情请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 中的 MC\_Reset(轴错误复位) 指令。

关于使用 NX 系列 脉冲输出单元时的差异，请参阅 “NX 系列 位置接口单元 用户手册 (SBCE-374)”。

### 9-1-2 绝对值定位

指定以原点为基准的绝对坐标目标位置的定位。

通过将计数模式设为旋转模式后指定动作方向，可进行转台附近等的定位。

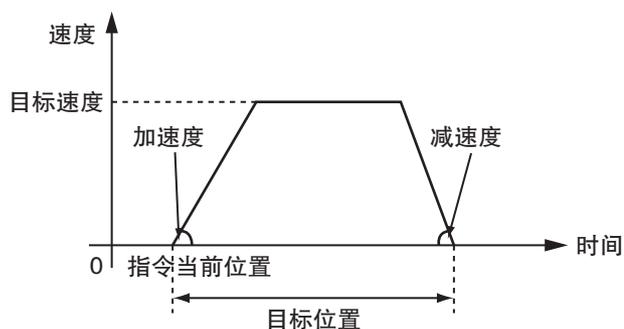


功能的详情请参阅 □□ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 的 MC\_MoveAbsolute(绝对值定位) 指令和 MC\_Move(定位) 指令。

### 9-1-3 相对值定位

指定与当前位置之间的距离的定位。

通过将计数模式设定为旋转模式，可指定超过环形计数器范围的移动距离。



功能的详情请参阅 □□ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 的 MC\_MoveRelative(相对值定位) 指令和 MC\_Move(定位) 指令。

### 9-1-4 中断标准定位

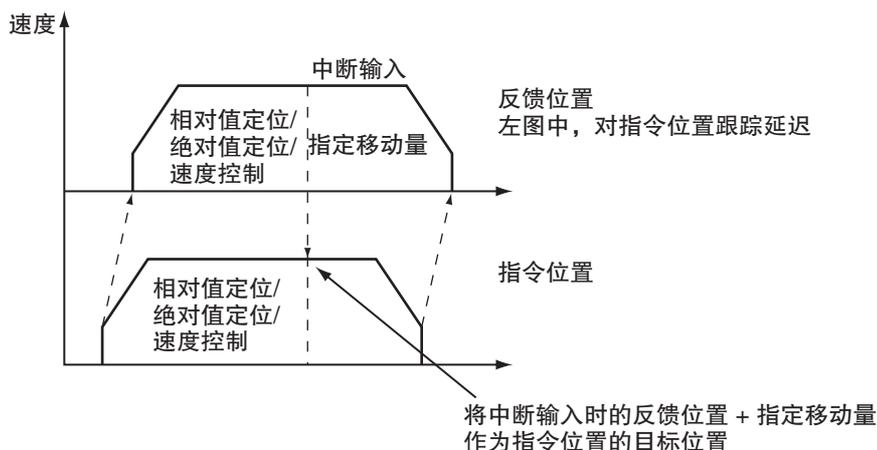
以指定的速度移动轴，从发生触发信号的反馈位置开始移动指定距离的定位。

对于绝对值移动或相对值移动，还可选择在指定移动量以内未发生触发信号时输出异常。

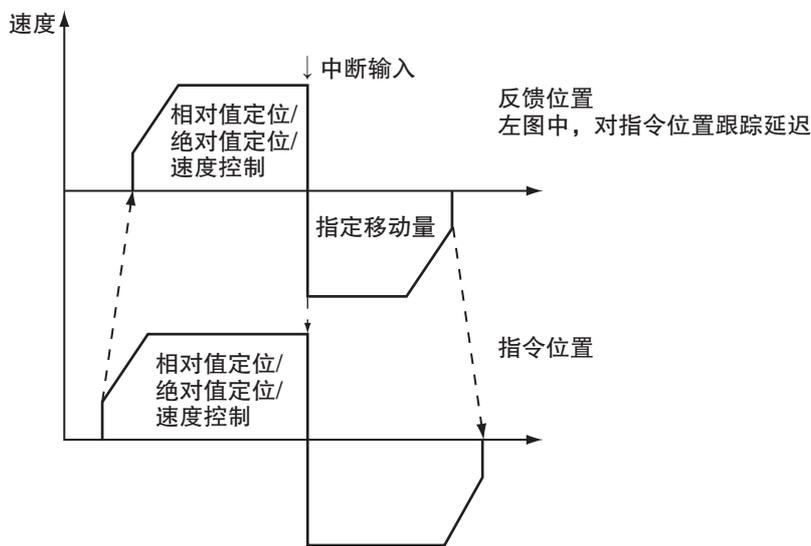
发生触发信号时的反馈位置使用了伺服驱动器的锁定功能，因此可实现不受位置偏差影响的标准定位。通过禁用超过指定位置范围的触发信号发生的窗口功能，还可消除包装机等非薄膜标志印刷的影响。

#### 以反馈位置为基准

##### ● 向行进方向移动指定量



##### ● 向行进方向的反方向移动指定量



如果在轴参数的取反时动作中选择 [ 反转时减速停止 ]，则以加减速曲线进行取反。

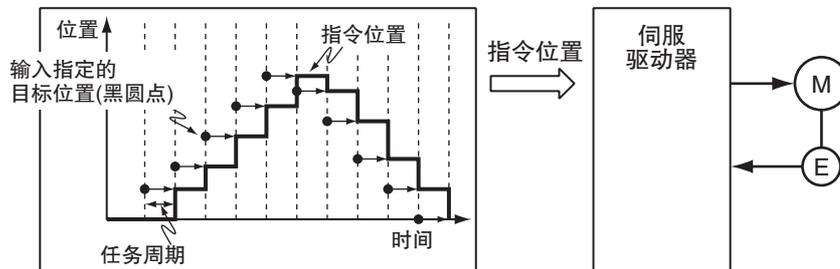
功能的详情请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 的 MC\_MoveFeed( 中断标准定位 ) 指令。

关于使用 NX 系列 脉冲输出单元时的差异，请参阅 □ “NX 系列 位置接口单元 用户手册 (SBCE-374)”。

### 9-1-5 周期同步定位

周期同步定位是按原始周期或固定周期任务的任务周期向轴输出指定目标位置的功能。目标位置以绝对位置指定。

用于需按用户创建的任意轨迹动作的场合等。



功能的详情请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 的 MC\_SyncMoveAbsolute(周期同步绝对定位) 指令。



#### 版本相关信息

周期同步定位功能可用于 Ver.1.03 以上的 CPU 单元和 Ver.1.04 以上的 Sysmac Studio 的组合。

### 9-1-6 停止

停止轴动作的功能包括通过连接至伺服驱动器的立即停止输入信号、极限输入信号和用户程序的运动控制指令停止的功能以及发生异常等后停止。

关于使用 NX 系列 脉冲输出单元时的差异, 请参阅 □ “NX 系列 位置接口单元 用户手册 (SBCE-374)”。

#### 通过伺服驱动器的输入信号停止

通过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

根据伺服驱动器的输入信号状态, 在 MC 功能模块中执行停止处理。

MC 功能模块的停止方法可从以下几种方法中进行选择。

- 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

欧姆龙制 1S 系列及 G5 系列的立即停止输入在伺服驱动器中也会发生异常, 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 )

根据伺服驱动器的输入信号状态，在 MC 功能模块中执行停止处理。  
MC 功能模块的停止方法可从以下几种方法中进行选择。

- 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

- 如果极限输入信号 ON，则启动轴指令的命令时请勿与处于 ON 状态的极限输入信号的方向相同。
- 轴组中任意轴的极限输入 ON 时，请勿启动轴组指令的命令。
- 执行输入变量中带有“Deceleration( 减速度 )”的同步动作指令时，如果输入减速停止信号，则以“Deceleration( 减速度 )”减速停止。
- 执行输入变量中无“Deceleration( 减速度 )”的同步动作指令时，如果输入减速停止信号，则以轴参数的最大减速度减速停止。



### 参考

- 使用伺服驱动器的输入信号时，需设定伺服驱动器。在欧姆龙制 1S 系列及 G5 系列 EtherCAT 内置型中，为伺服驱动器的立即停止输入和极限输入分配初始值。
- 连接欧姆龙制伺服驱动器 1S 系列的设定示例，请参阅 □ “A-1 与伺服驱动器 1S 系列之间的连接 (P.A-2)”。
- 连接欧姆龙制 G5 系列时的设定示例请参阅 □ “A-2 与伺服驱动器 G5 系列之间的连接 (P.A-10)”。

## 通过运动控制指令停止

需停止单轴动作时，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

可对单轴控制和同步控制功能指定减速度和跃度并减速停止。

如果减速度指定为“0”，则将立即停止的指令发送至伺服驱动器。

通过本指令执行减速停止时或输入变量的 Execute( 启动 ) 为 TRUE 时，无法接受其他动作指令。

### ● MC\_ImmediateStop( 立即停止 ) 指令

可对单轴控制和同步控制功能执行立即停止。

也可对启用轴组的轴执行。

功能的详情请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 的 MC\_Stop( 强制停止 ) 指令、MC\_ImmediateStop( 立即停止 ) 指令。



### 参考

如果将 MC\_Power( 可运行 ) 指令的输入变量 Enable( 启用 ) 设为 FALSE，MC 功能模块会立即停止指令值，执行伺服 OFF。  
伺服 OFF 时伺服驱动器的动作取决于伺服驱动器的设定。

## 发生异常后停止

### ● 单轴动作中发生异常后停止

单轴动作中发生异常时，根据异常内容立即停止或减速停止。  
各异常的停止方法请参阅 □ “11-2-2 异常内容 (P.11-16)”。

### ● 通过软件限制停止

在轴参数设定的“软件限制功能”中设定为通过软件限制停止。  
通过软件限制停止的方法可从以下几种方法中进行选择。

- 对指令位置启用。执行减速停止。
- 对指令位置启用。执行立即停止。
- 对反馈位置启用。执行减速停止。
- 对反馈位置启用。执行立即停止。

软件限制功能请参阅 □ “9-8-5 软件限制功能 (P.9-71)”。

### ● 超过运动控制周期后停止

运动控制处理未在 2 个周期内完成且超过控制周期时，立即停止。



#### 使用注意事项

使用 NX701 CPU 单元执行多运动动作，当任意一方的任务超过运动控制周期时，两个任务的所有轴立即停止。

多任务运动请参阅 □ “A-6-2 运动控制功能 (P.A-30)”。

### ● 发生导致伺服 OFF 的异常

发生导致伺服 OFF 的异常时立即停止。伺服 OFF 时伺服驱动器的动作取决于伺服驱动器的设定。

### ● 启动 MC 试运行功能后停止

启动 Sysmac Studio 的 MC 试运行功能后，各轴以“最大减速度”减速停止。

### ● 关闭 MC 试运行功能后停止

关闭 Sysmac Studio 的 MC 试运行功能后，各轴以“最大减速度”减速停止。

- 选择 Sysmac Studio 的 MC 试运行画面中的 [MC 结束试运行]。
- 关闭 Sysmac Studio 的 MC 试运行画面。
- 关闭 Sysmac Studio。

### ● 切换 CPU 单元的模式后停止

切换 CPU 单元的模式后，各轴以“最大减速度”减速停止。



#### 使用注意事项

- 执行输入变量中带有“Deceleration(减速度)”的同步动作指令时，如果输入减速停止信号，则以“Deceleration(减速度)”减速停止。
- 执行输入变量中无“Deceleration(减速度)”的同步动作指令时，如果发生导致减速停止的异常，则以轴参数的最大减速度减速停止。

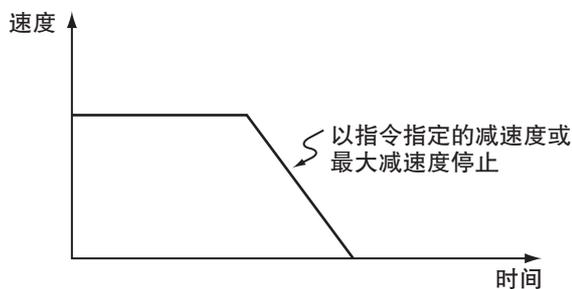


## 参考

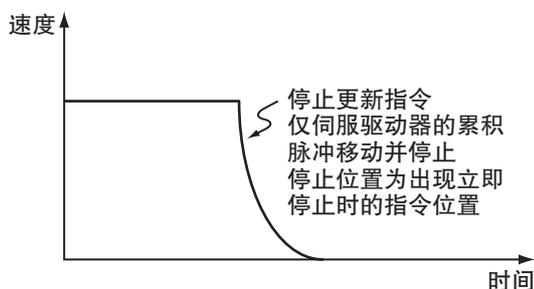
- 从运行模式切换至程序模式后，中断动作中的运动控制指令。此时，指令的输出变量“Command Aborted(执行中断)”保持 FALSE 不变。切换至程序模式后仍保持伺服 ON/OFF 状态。
- 从运行模式切换至程序模式后减速停止时，如果返回运行模式，则会清除运动控制指令的输出变量，因此运动控制指令的输出变量“CommandAborted(执行中断)”保持 FALSE 不变。
- 通过 MC\_SaveCamTable(保存凸轮表)指令正在保存时，会继续执行保存处理。
- 如果正在通过 MC\_GenerateCamTable(凸轮表生成)指令执行凸轮表生成时，会继续执行生成处理。

## 停止方法

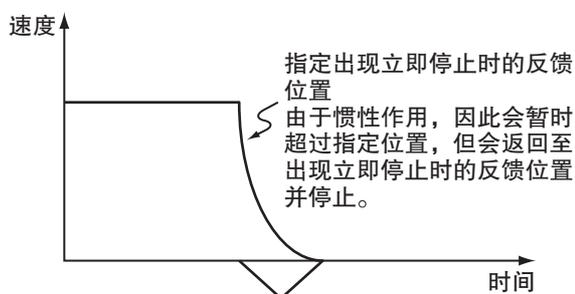
### ● 减速停止指令值



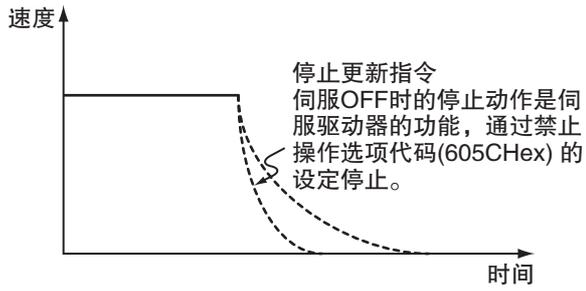
### ● 立即停止指令值



### ● 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



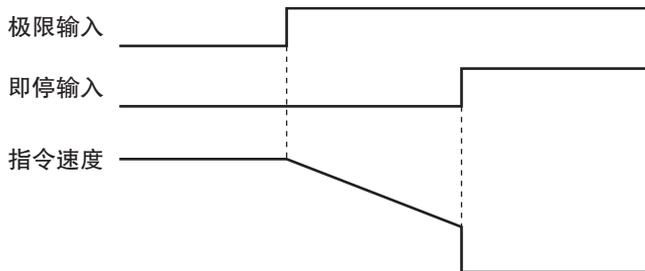
## 停止的优先顺序

停止的优先顺序如下表所示。停止过程中如果发生优先顺序较高的停止，则切换至优先顺序较高的停止方法。

停止方法	优先顺序 (数字越大优先度越高)
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

下图为极限输入信号 ON 时在减速停止中使立即停止输入 ON，执行立即停止时的示例。



### 9-1-7 超调

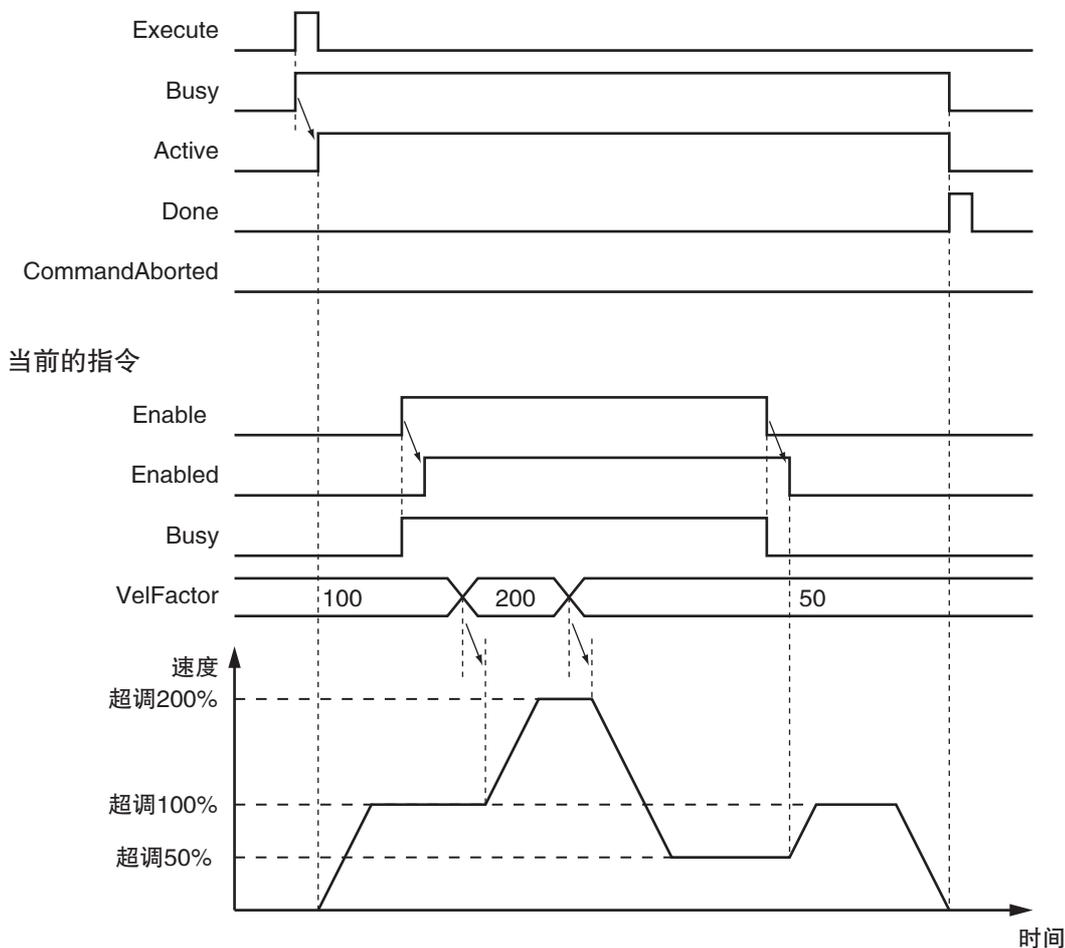
通过 MC\_SetOverride( 设定超调值 ) 指令，设定启动中的轴动作相应的超调值的功能。  
超调值是目标速度相应的比率，可在 0 ~ 500% 的范围内设定。  
如果相对于目标速度设定为 0%，则以速度 “0” 停止，保持动作状态。

启用超调时，导入设定的超调值。  
如果禁用超调，则超调值恢复为 100%。  
变更后的超调值超过最高速度时，则变为各轴的 “最高速度”。

#### ● 对 MC\_MoveAbsolute( 绝对值定位 ) 指令的超调

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

之前的指令(MC\_MoveAbsolute)



功能的详情请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 的 MC\_SetOverride( 设定超调值 ) 指令。

## 9-2 单轴同步控制

下面对单轴的同步控制的动作进行说明。

### 9-2-1 同步控制的概要

与主轴的位置同步，控制从轴位置的功能。

可为主轴指定各轴的指令位置或反馈位置。

从轴的指定速度超过轴参数的 [ 最高速度 ] 时，以最高速度进行指定。此时，不足的移动量会在下一周期后分配输出。



#### 使用注意事项

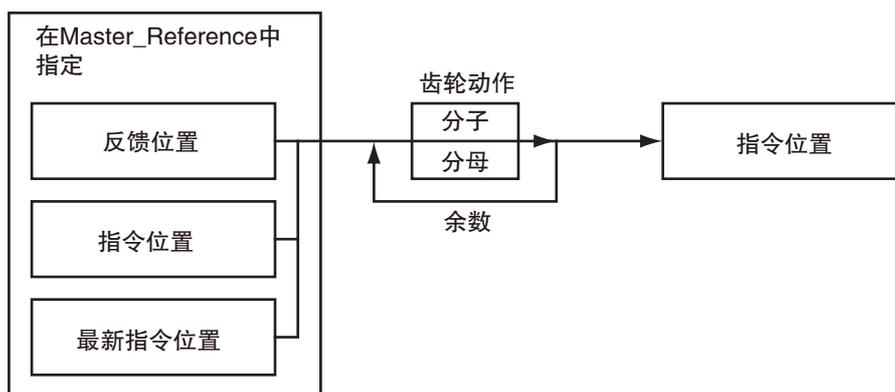
- 编码器轴、虚拟编码器轴、单轴位置控制轴不可指定为从轴。
- 使用 NX701 CPU 单元执行多运动动作，在同步控制时，请将构成的主轴和从轴设为同一任务。

通过 MC\_GearIn( 齿轮动作开始 ) 指令和 MC\_CamIn( 凸轮动作开始 ) 指令等的同步控制指令，将任务各异的轴指定为主轴、从轴时，将会发生“主轴指定错误(事件代码: 54620000 Hex)”。要将不同任务的轴指定为主轴、从轴时，请参阅 □ “9-2-10 多任务运动中实现同步控制的方法 (P.9-25)”。

### 9-2-2 齿轮动作

设定主轴和从轴间的齿轮比，进行齿轮动作的功能。

通过 MC\_GearIn( 齿轮动作开始 ) 指令开始齿轮动作，通过 MC\_GearOut( 齿轮动作解除 ) 指令或 MC\_Stop( 强制停止 ) 指令解除同步。



可为动作对象从轴指定齿轮比分子、齿轮比分母、位置类型、加速度、减速度。还可为主轴指定指令位置、反馈位置、最新的指令位置中的任意一个。

开始动作后，从轴以主轴速度乘以齿轮比得到的速度为目标速度，进行加减速动作。

达到目标速度之前称为 Catching phase( 追赶中 )，达到后称为 InGear phase( 齿轮同步中 )。

齿轮比为正数时，Slave( 从轴 ) 沿 Master( 主轴 ) 的同方向移动；为负数时，Slave( 从轴 ) 沿 Master( 主轴 ) 的反方向移动。

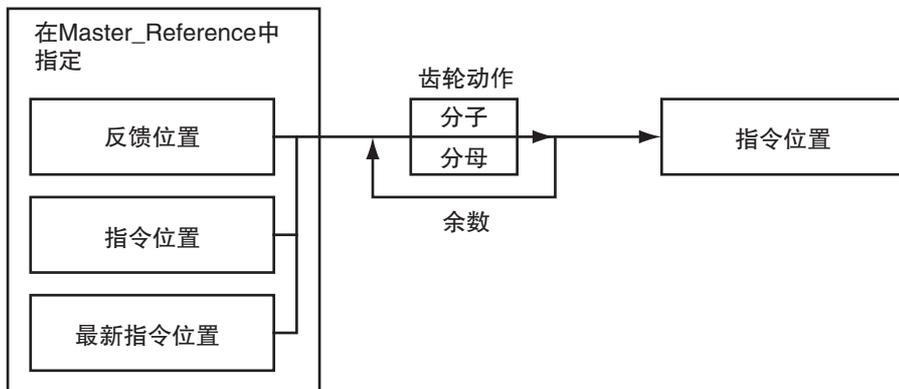
齿轮动作的详情请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 中的 MC\_GearIn( 齿轮动作开始 ) 指令、MC\_GearOut( 齿轮动作解除 ) 指令及 MC\_Stop( 强制停止 ) 指令。

### 9-2-3 位置指定齿轮动作

设定主轴和从轴间的齿轮比，进行齿轮动作的功能。

位置指定齿轮动作中可指定需开始同步的主轴位置和从轴位置。

通过 MC\_GearInPos( 位置指定齿轮动作 ) 指令开始位置指定齿轮动作，通过 MC\_GearOut( 齿轮动作解除 ) 指令或 MC\_Stop( 强制停止 ) 指令结束同步。

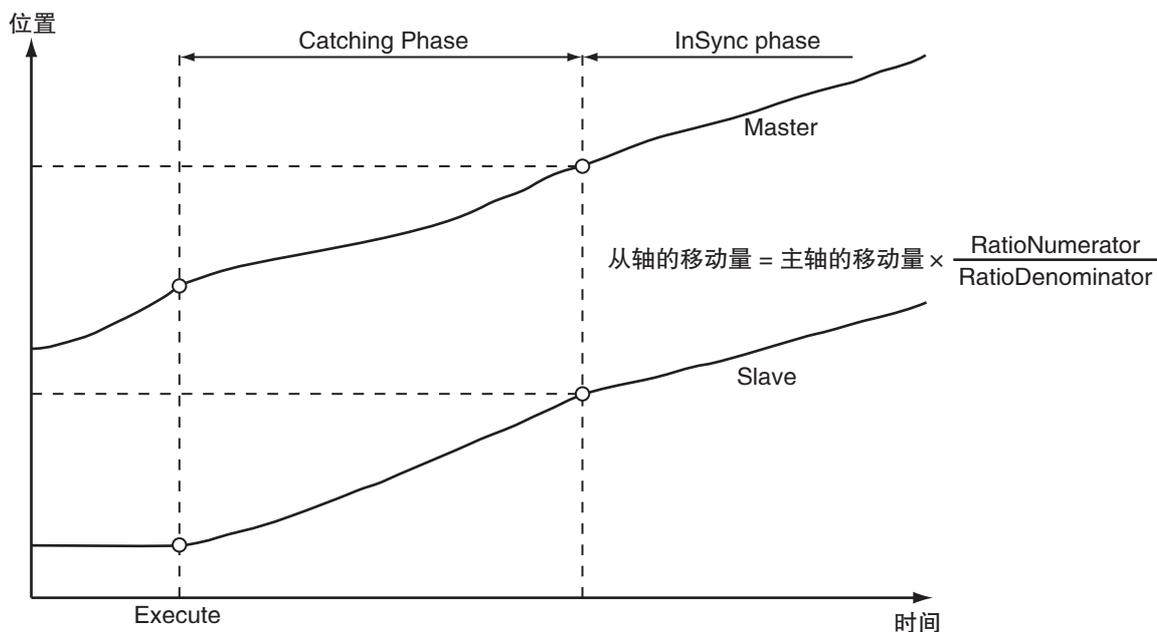


可为动作对象从轴指定齿轮比分子、齿轮比分母、位置类型、加速度、减速度。还可为主轴指定指令位置、反馈位置、最新的指令位置中的任意一个。

开始动作后，从轴以主轴速度乘以齿轮比得到的速度为目标速度，进行加减速动作。

从轴到达从轴同步位置之前为 Catching phase( 追赶中 )，从轴到达从轴同步开始位置之后为 InSync( 齿轮同步中 )。从轴在任何区间内均与主轴的位置同步。

齿轮比为正数时，Slave( 从轴 ) 沿 Master( 主轴 ) 的同方向移动；为负数时，Slave( 从轴 ) 沿 Master( 主轴 ) 的反方向移动。下图为齿轮比为正数时的动作。



位置指定齿轮动作的详情请参阅 “NJ/NX 系列 指令基准手册 运动篇(SBCE-364)” 中的 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。

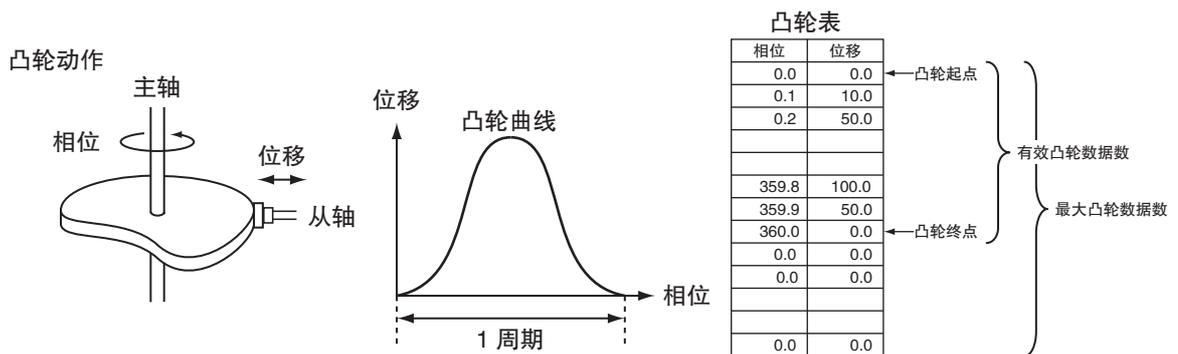
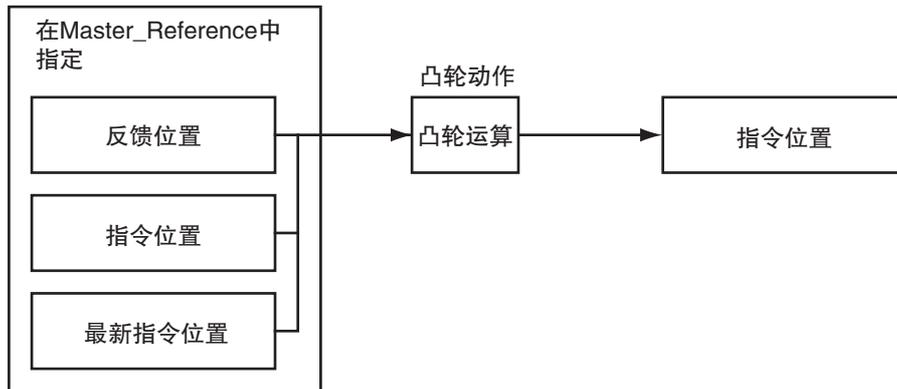
## 9-2-4 凸轮动作

从轴根据凸轮表与主轴位置同步进行动作的功能。

通过 MC\_CamIn(凸轮动作开始)指令开始凸轮动作,通过 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。

使用 Sysmac Studio 的凸轮编辑功能创建凸轮表,下载至 CPU 单元主体。

下载至 CPU 单元时,使用 Sysmac Studio 的“同步”功能。



Ver.1.06 以上版本的 CPU 单元和 Ver.1.07 以上版本的 Sysmac Studio 组合时,在执行 MC\_CamIn(凸轮动作开始)指令期间,通过指定合并,多重启动其他 MC\_CamIn(凸轮动作开始)指令,从轴不会停止,可在切换凸轮表的同时保持动作。

凸轮动作的详情请参阅 [□](#) “NJ/NX 系列 指令基准手册 运动篇(SBCE-364)”中的 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。

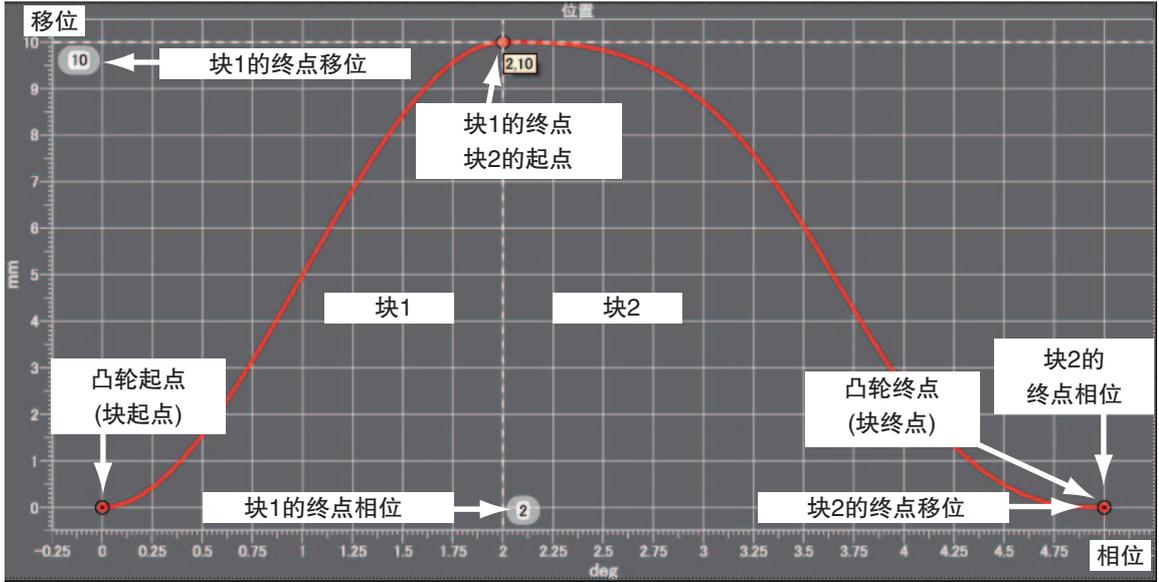
凸轮编辑功能的详情请参阅 [□](#) “Sysmac Studio Version 1 操作手册(SBCA-362)”。

## 9-2-5 凸轮表

下面对凸轮动作使用的凸轮表进行说明。

## 凸轮表用语

用 语	说 明
凸轮动作	具有 1 个主轴和 1 个从轴，根据凸轮曲线通过主轴相位计算从轴的位移进行动作的功能。
凸轮曲线	用于实现凸轮动作的相位与位移之间关系的曲线。 通过 Sysmac Studio 的凸轮编辑器创建。将凸轮曲线下载至 CPU 单元后，可通过用户程序将其作为凸轮数据变量处理。 下载至 CPU 单元时，使用 Sysmac Studio 的“同步”功能。
凸轮块	可选择凸轮曲线的区间。 表示前一个凸轮块的终点与本凸轮块的终点之间。
凸轮曲线	表示凸轮特性的曲线。可按凸轮块选择凸轮曲线。 通过指定点使用由 Sysmac Studio 计算出的相位宽度、位移宽度，由 Sysmac Studio 创建实际的凸轮曲线。 包括直线、恒定加速度、Trapeclloid 等。
凸轮数据	由实现凸轮动作的相位(主轴)和位移(从轴)构成的数据。
凸轮数据变量	对凸轮数据进行结构体数组化的变量。
凸轮表	表示凸轮数据的数据表。 如果相位为升序，则视为错误的凸轮表。
凸轮起点	表示凸轮数据的起始数据。
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，凸轮终点后的相位和位移为“0”。
凸轮块起点	凸轮块的起始点。在凸轮动作的开始部分，与“凸轮起点”相同的点。凸轮曲线连续时，与“凸轮块终点”相同的点。
凸轮块终点	凸轮块的终止点。在凸轮动作的终止部分，与“凸轮终点”相同的点。凸轮曲线连续时，与“凸轮块起点”相同的点。 凸轮块终点以(横轴、纵轴)=(相位终点、位移终点)进行定义。
原始凸轮数据	通过凸轮编辑器分割凸轮曲线后创建的凸轮数据。
程序变更凸轮数据	运行 CPU 时，通过用户程序变更的凸轮数据。
主轴	在凸轮动作的输入轴中可同时指定线性模式或旋转模式
从轴	在凸轮动作的输出轴中可同时指定线性模式或旋转模式。
相位	距离凸轮表起点位置的主轴的相对量。
位移	距离主轴跟踪开始位置的从轴的相对量。
有效凸轮数据	凸轮起点以外，相位为非“0”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	可带有凸轮表的凸轮数据的最大组数。
凸轮数据索引	执行凸轮数据的编号。
凸轮数据开始位置	凸轮起点(相位=0)相应的主轴的绝对位置。
主轴跟踪起始位置	在从轴开始凸轮动作的主轴位置中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。
起始位置指定方式	指定以绝对位置或相对位置表示主轴跟踪起始位置。
空白凸轮数据	可设定在终点后的相位和位移为“0”的凸轮数据。
连接速度	在凸轮曲线之间连接时的速度。部分曲线无法指定。
连接加速度	在凸轮曲线之间连接时的加速度。部分曲线无法指定。
相位间隔宽度	以相位(横轴)分割凸轮曲线时的宽度。以相位间隔宽度分割曲线时的各点为凸轮表中的凸轮数据。须按凸轮块进行指定。



### 何谓凸轮表

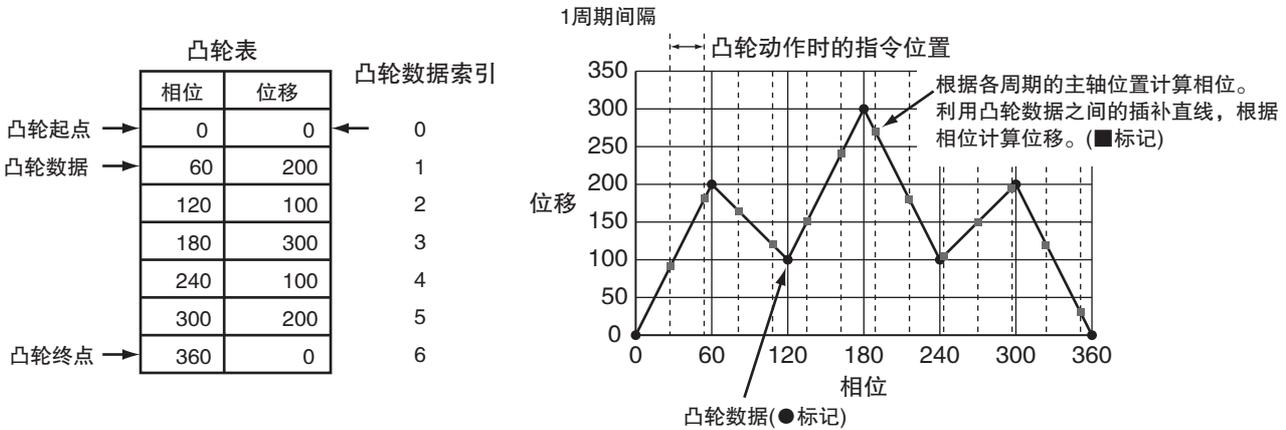
在 MC 功能模块中，将由主轴相位和从轴位移组成的 1 对数据定义为凸轮数据，将多个凸轮数据的组合定义为凸轮表。

通过 Sysmac Studio 的凸轮编辑器创建凸轮表。还可通过用户程序变更凸轮表内的凸轮数据。

凸轮表中的凸轮数据的相位和位移值表示为从起点“0.0”开始的相对量。

凸轮动作中，对主轴相位相邻的 2 个凸轮数据进行直线插补后的位移到达向从轴发送指令的位置。

凸轮表中的凸轮数据数越多，轨迹精度越高，凸轮曲线越平滑。



### 使用注意事项

- 凸轮表中的凸轮数据按相位值升序创建。相位非升序时，执行凸轮动作指令时会因指令而发生异常。
- 凸轮数据变量为全局变量，可以从多个任务中查看或变更凸轮数据变量值。从多个任务中变更凸轮数据变量值时，请编写通过多个任务进行改写处理时相互不会发生冲突的程序。
- 使用“全局变量的任务间排他性控制”功能进行凸轮数据变量的排他性控制时，请不要使用排他对象的凸轮数据变量启动运动指令。否则会发生轻度故障等级的异常“凸轮表指定错误(错误代码: 54390000 Hex)”。

## 凸轮表规格

项 目	说 明
每个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 个表 *2
凸轮动作的切换	可通过运动控制指令切换至其他凸轮动作
凸轮数据的改写	可通过用户程序改写 可通过凸轮表生成指令改写 *3
凸轮数据的保存	可通过凸轮数据保存指令保存至非易失性存储器
凸轮数据的附带信息	可下载、上传凸轮编辑器显示用信息 *4
凸轮数据反映至主存储器的时间	· 从 Sysmac Studio 下载时 · 电源接通时

\*1. 1 个凸轮表为 65,535 点时，最多为 16 个凸轮。分辨率为 0.1deg 时，1 个凸轮表为 3600 点，最多为 291 个凸轮。

\*2. 合计 10MB 以下。

\*3. Ver.1.08 以上版本的 CPU 单元和 Ver.1.09 以上版本的 Sysmac Studio 组合时，可使用凸轮表生成指令。

\*4. “下载”和“上传”使用 Sysmac Studio 的“同步”功能。

## 凸轮表的数据类型

对凸轮数据结构体进行数组声明后再使用凸轮表。凸轮数据结构体事先进行的类型宣言如下所示。

```

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

```

须通过 Sysmac Studio 的凸轮编辑器创建凸轮数据，指定凸轮表名称和凸轮数据数（数组大小）。

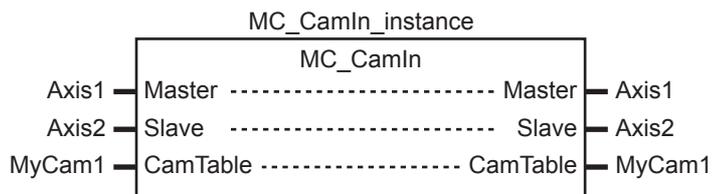
例如，凸轮名称为“MyCam1”，凸轮数据数为 1000 个时，声明如下。

```

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

```

通过凸轮动作指令指定“MyCam1”时，记述如下。其中，将主轴设为 Axis1、从轴设为 Axis2。



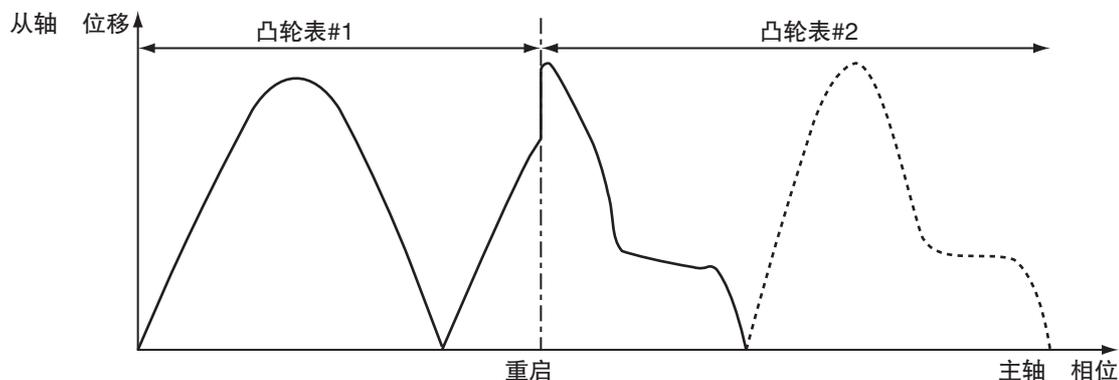
控制器中不存在指定凸轮表时，检测到异常。此外，多根轴可指定同一个凸轮表。

## 凸轮表的切换

通过在凸轮动作中重启凸轮动作指令，可切换凸轮表。

切换后，重启时以指定的凸轮表执行凸轮动作。

按切换后的凸轮表输出 MC\_CamIn(凸轮动作开始)指令的输出变量“EndOfProfile(终点执行)”和“Index(索引)”。



### 使用注意事项

- 切换目标凸轮表需事先保存至非易失性存储器。
- 如果在凸轮动作中切换凸轮，则速度会不连续。请调整切换时间进行使用，避免速度过度不连续。

## 凸轮数据的读写与凸轮表的保存

与其他变量相同，可通过用户程序读写凸轮数据。

例如，凸轮表名称为“MyCam1”的第1个数组元素，相位可通过“MyCam1[0].Phase”指定，位移可通过“MyCam1[0].Distance”指定。

通过用户程序改写的凸轮数据可使用 MC\_SaveCamTable(保存凸轮表)指令作为凸轮表保存至 CPU 单元的非易失性存储器。



### 使用注意事项

- 启动保存凸轮表指令前，如果 CPU 单元的电 OFF、通过本指令执行的保存未正常完成或通过 Sysmac Studio 执行下载，则改写后的凸轮数据会丢失。
- 通过 CPU 单元中的用户程序改写凸轮数据时，请注意避免丢失改写的的数据。
- 非易失性存储器中保存的凸轮数据可通过 Sysmac Studio 的上传功能导入。
- “下载”和“上传”使用 Sysmac Studio 的“同步”功能。

数组指定请参阅 “NJ/NX 系列 CPU 单元用户手册 软件篇 (SBCA-359)”。

保存凸轮表指令的详情请参阅 “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 的 MC\_SaveCamTable(保存凸轮表)指令。

## 凸轮表的属性更新

MC 功能模块中需识别凸轮表的凸轮终点。

如果在凸轮动作中进行了改写(通过用户程序变更了有效凸轮数据数),则需将有效凸轮数据数更新为最新。因此,需使用 MC\_SetCamTableProperty(更新凸轮表属性)指令。

终点是指从起点开始按顺序搜索凸轮表时,排列在相位最初为“0”的凸轮数据前一位的数据。检测到相位“0”之后的凸轮数据无效。

例如,对于以下凸轮表,MC\_SetCamTableProperty(更新凸轮表属性)指令的输出变量“EndPointIndex(终点索引)”为“999”、“MaxDataNumber(最大数据数)”为“5000”。

凸轮数据结构体数组	相位	位移	
MyCam1 [0]	0	0	凸轮起点
.	.	.	
.	.	.	
.	.	.	
MyCam1 [997]	359.8	2	有效数据
MyCam1 [998]	359.9	1	
MyCam1 [999]	360.0	0	凸轮终点
MyCam1 [1000]	0	0	
.	.	.	无效数据
.	.	.	
.	.	.	
MyCam1 [4999]	0	0	

最大数据数: 5000



### 使用注意事项

- 最大凸轮数据数不能通过用户程序变更。
- 请改写(变更有效数据数)后再执行本指令。  
如果不更新有效凸轮数据数,则凸轮和MC\_CamIn的EndOfProfile(凸轮周期完成)不会正常动作。

凸轮表属性更新的详情请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 中的 MC\_SetCamTableProperty(更新凸轮表属性)指令。

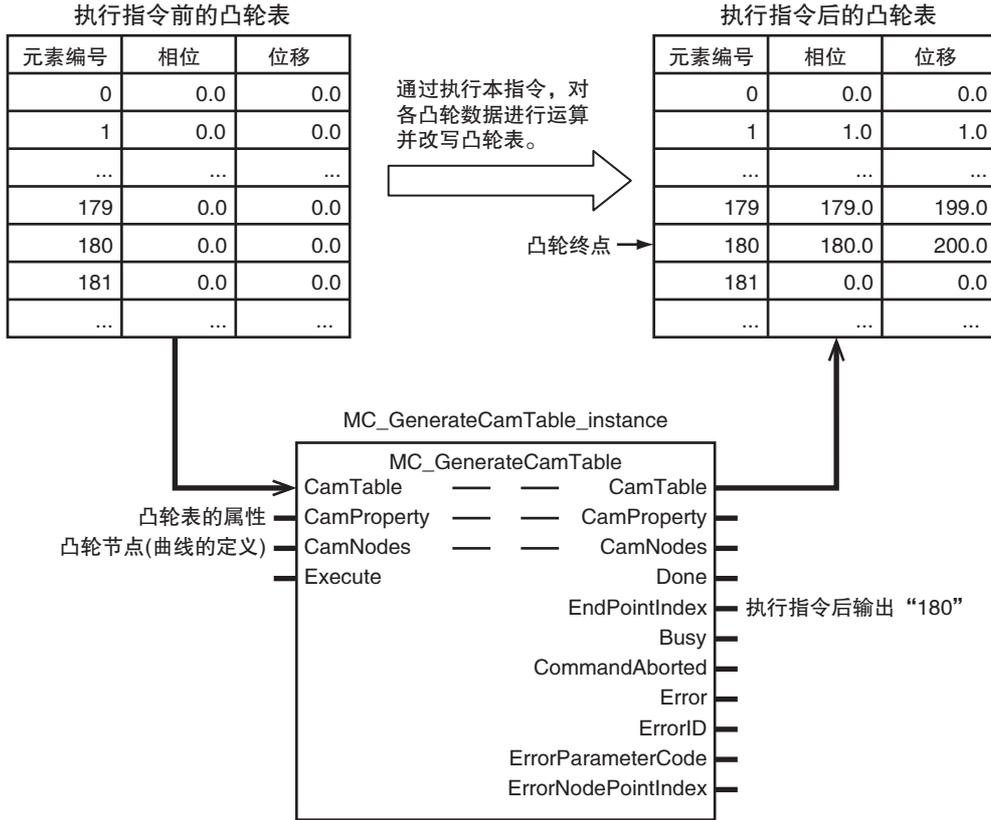
## 凸轮表生成

Ver.1.08 以上的 CPU 单元和 Ver.1.09 以上的 Sysmac Studio 组合时，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）的指定值计算凸轮数据的值，改写指定为指令的输入输出变量 CamTable（凸轮表）的凸轮数据变量。

完成改写后更新凸轮表的终点索引，将凸轮终点的元素编号输出到 EndPointIndex（终点索引）。

MC\_GenerateCamTable（凸轮表生成）指令结束后，无需执行 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` 型的数组变量，通过 Sysmac Studio 的凸轮编辑器创建。CamProperty（凸轮属性）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，是通过 Sysmac Studio 的全局变量表作为用户定义变量创建，或通过 Sysmac Studio 的凸轮数据设定创建。

CamNodes（凸轮节点）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` 型的数组变量，是通过 Sysmac Studio 的全局变量表作为用户定义变量创建，或通过 Sysmac Studio 的凸轮数据设定创建。

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，变量保存属性的初始值为“不保存”。若要在变更变量值，切换至程序模式或重新接通电源后再次使用，请将变量的保存属性设定为“保存”。若每次都通过显示器等进行设定，也可维持“不保存”。

通过 Sysmac Studio 的凸轮数据设定创建凸轮定义变量时，变量的保存属性固定为“保存”。

通过从显示器等设定 MC\_GenerateCamTable(凸轮表生成)指令的指定值，无需使用 Sysmac Studio 即可创建凸轮数据变量、调整凸轮动作。

下面对调整凸轮动作的步骤的概要进行说明。

- 1** 事先通过用户程序创建以下处理。
  - 将通过显示器设定的凸轮定义变量值指定到凸轮表生成指令的处理
  - 在显示器中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
  - 在显示器中显示 EndPointIndex(终点索引)的值的处理
- 2** 通过显示器设定凸轮定义变量值。
- 3** 执行凸轮表生成指令。
- 4** 通过显示器确认生成的凸轮表的形状和终点索引的值。
- 5** 如果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，研究是否变更凸轮定义变量值。
- 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阅 □□ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 的 MC\_GenerateCamTable(凸轮表生成)指令。

关于通过 Sysmac Studio 创建和传送凸轮定义变量，请参阅 □□ “Sysmac Studio Version 1 操作手册 (SBCA-362K 以上)”。

### 9-2-6 梯形模式凸轮动作

指定的从轴与指定的主轴同步，同时按梯形曲线执行定位的功能。

一种电子凸轮，不使用通过凸轮编辑器创建的凸轮表。

启动 MC\_MoveLink( 梯形模式凸轮 ) 指令后开始动作。

需停止动作中的轴时，应使用 MC\_Stop( 强制停止 ) 指令。

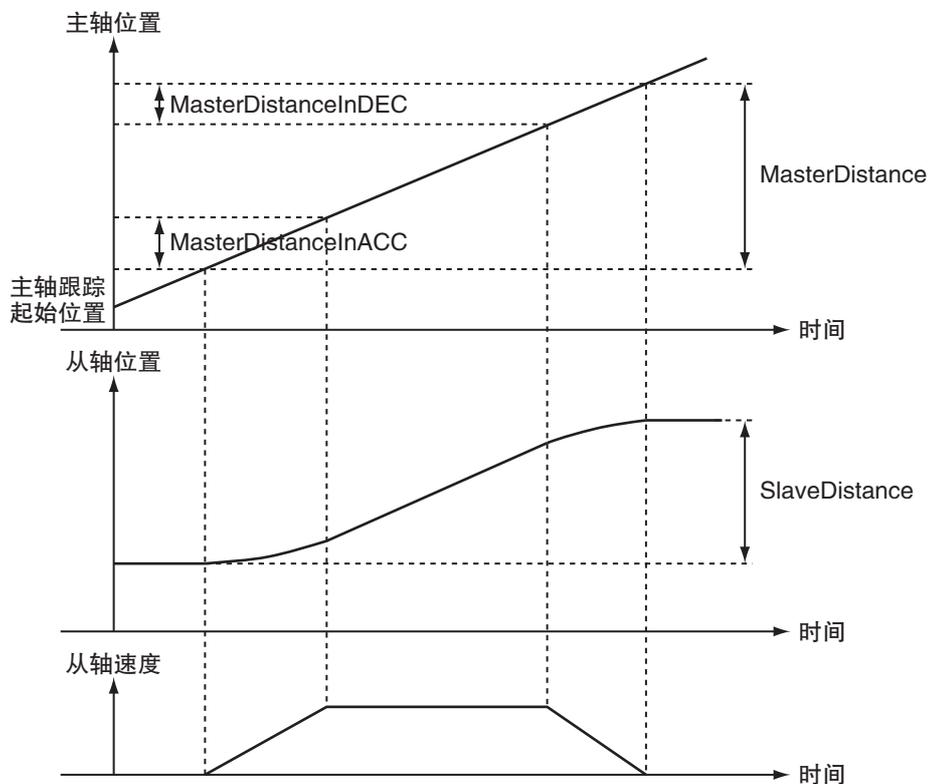
以动作对象轴为 “Slave( 从轴 )”，指定 “Master( 主轴 )”、“MasterDistance( 主轴移动距离 )”、“MasterDistanceInACC( 主轴加速移动距离 )”、“MasterDistanceInDEC( 主轴减速移动距离 )”、“SlaveDistance( 从轴移动距离 )”、“MasterStartDistance( 主轴跟踪距离 )”。

可为主轴指定指令位置或反馈位置。

可从 “指令开始时”、“触发检测时”、“主轴到达主轴跟踪距离时” 中指定任意一个作为同步动作的开始条件。

如下图所示，从轴的速度和位置由主轴与从轴的移动量之比决定。

下图的同步开始位置表示同步开始条件有效时的位置。



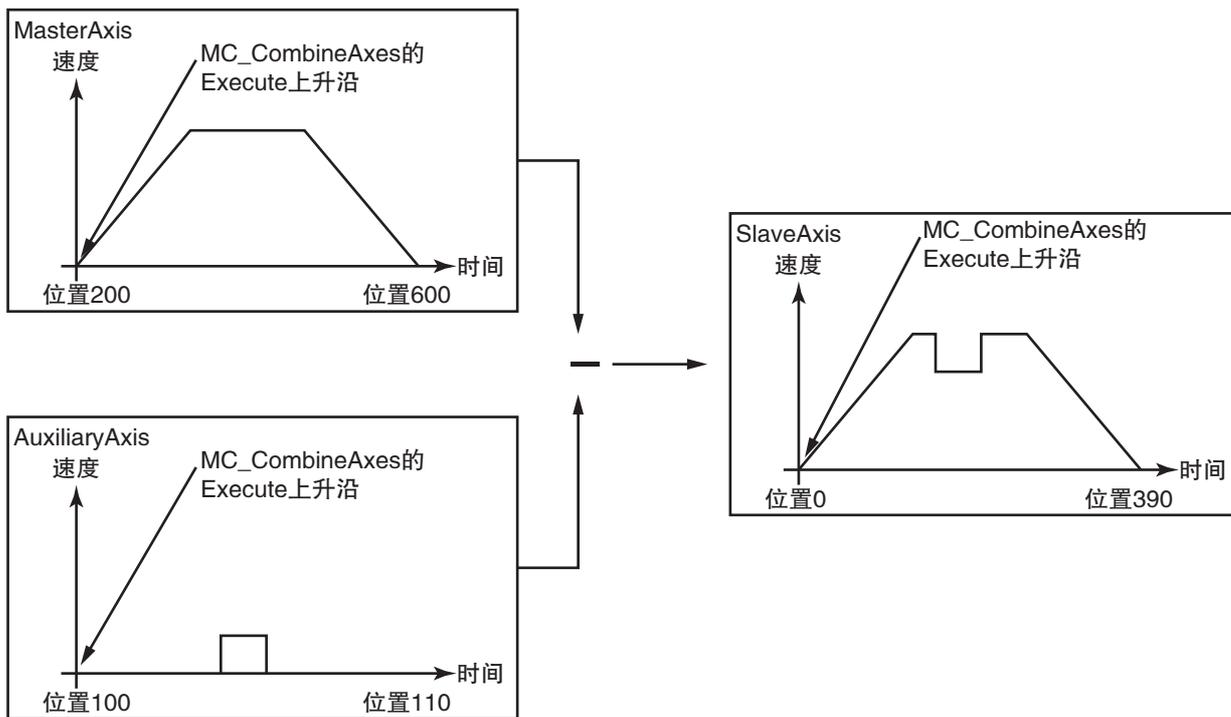
梯形曲线凸轮动作的详情请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 的 MC\_MoveLink ( 梯形模式凸轮 ) 指令、MC\_Stop( 强制停止 ) 指令。

### 9-2-7 加减运算定位

将 2 个位置相加或相减的值作为从轴指令位置的功能。  
启动 MC\_CombineAxes(加减运算定位) 指令后开始动作。  
需停止动作中的轴时, 应使用 MC\_Stop(强制停止) 指令。

下图为相减时的动作示例。

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



加减法运算定位动作的详情请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 中的 MC\_Cobine Axes(加减法运算定位) 指令和 MC\_Stop(强制停止) 指令。

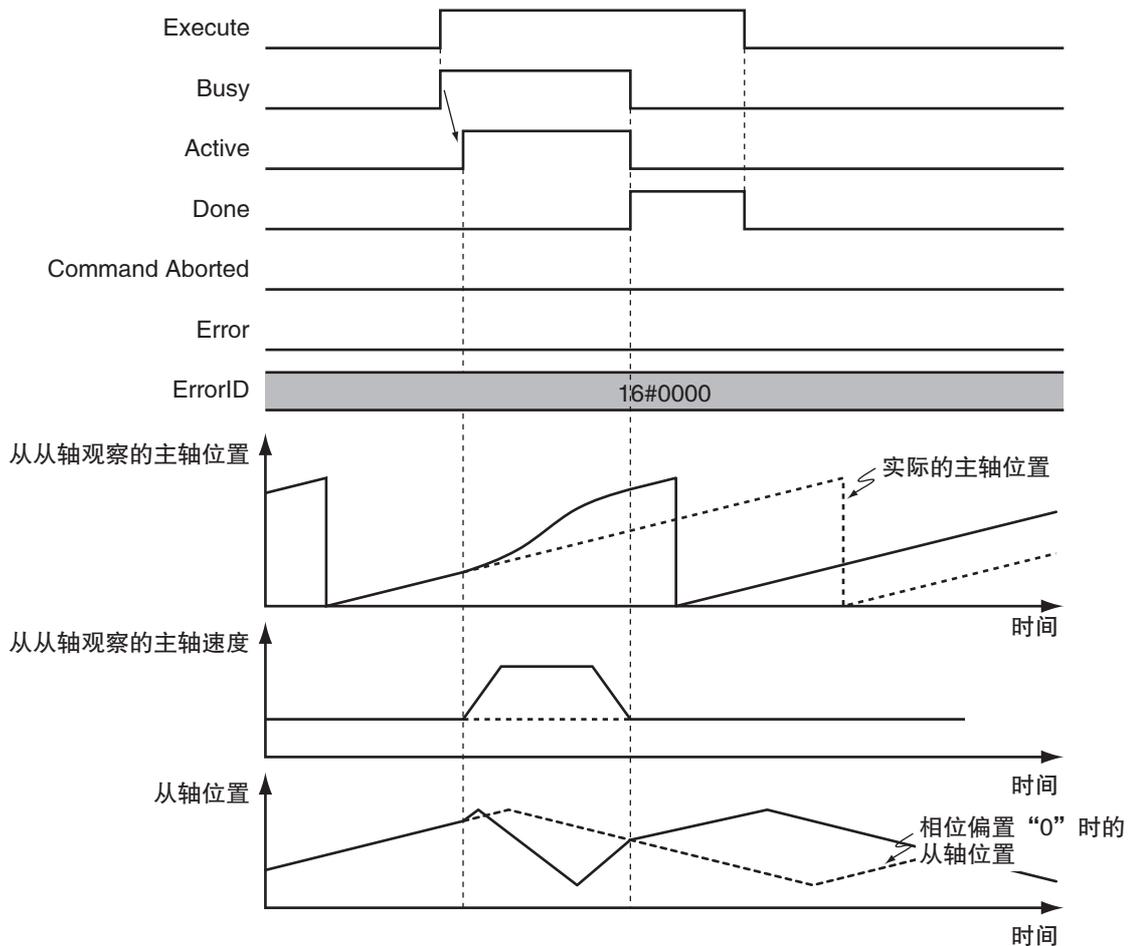
## 9-2-8 主轴相位补偿

对动作中的指令执行主轴(从从轴观察)的相位补偿的功能。

从从轴观察的补偿量为相对量,在同步状态下,从轴对主轴的相对量同步。

通过启动 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)指令可指定“相位补偿量”、“目标速度”、“加速度”、“减速度”、“跃度”。



可执行主轴相位补偿的同步控制指令和主轴相位补偿功能的详情请参阅 “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 中的 MC\_Phasing(主轴相对值相位补偿) 指令。

### 9-2-9 从轴位置补偿

对同步控制中的从轴进行位置补偿的功能。

对于同步控制中的从轴，在指令当前位置上加上根据输入变量值计算出的偏置量，输出至伺服驱动器。即使 MC 功能模块对 2 轴发出移动量相同的指令，但因机械变形等影响，移动量出现差异时，可以进行补偿。

通过启动 MC\_SyncOffsetPosition(周期性同步位置偏置补偿)指令，可以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阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 的 MC\_SyncOffsetPosition(周期性同步位置偏置补偿) 指令。

### 9-2-10 多任务运动中实现同步控制的方法

使用 MC 功能模块的标准功能时，若执行多运动且分配至不同任务的轴间同步控制指令，将会发生“主轴指定错误 (事件代码：54620000 Hex)”。

但通过使用 MC\_PeriodicSyncVariables(轴变量任务间固定周期同步)指令，可以对原始恒定周期任务所控制的主轴与固定周期任务 (执行优先度 5) 所控制的从轴进行同步控制。

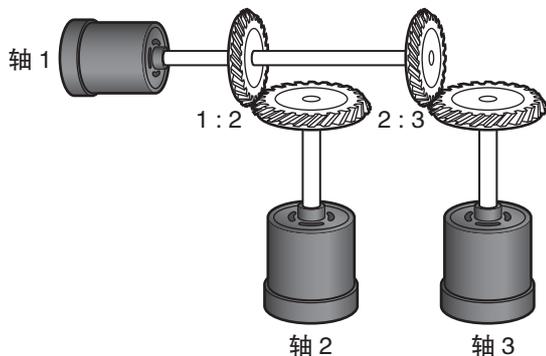
主轴与从轴同步的凸轮动作和齿轮动作可用于以下组合。

主轴的任务	从轴的任务	
	原始恒定周期任务	固定周期任务 (执行优先度 5)
原始恒定周期任务	通过运动控制指令同步	在原始恒定周期任务中执行 MC_PeriodicSyncVariables(轴变量任务间固定周期同步) 指令，通过在固定周期任务 (执行优先度 5) 中使用“虚拟主轴”可以实现。
固定周期任务 (执行优先度 5)	无法使用	通过运动控制指令同步

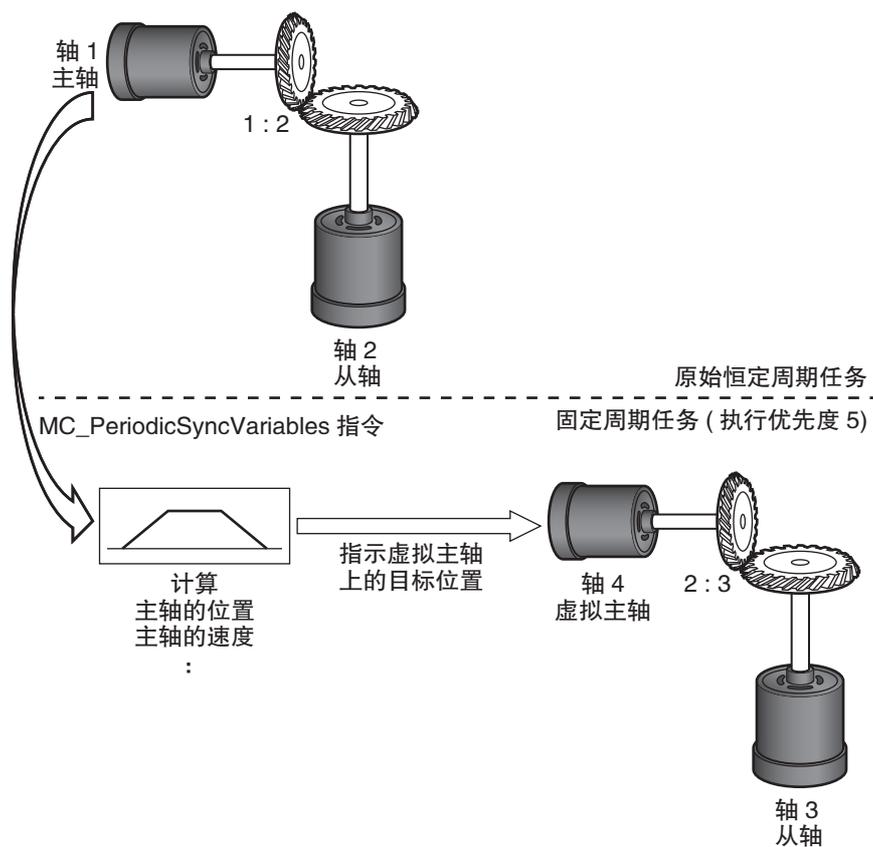
### 动作示例的轴构成

如下图所示，相对于主轴 (轴 1)，将高速且需要高精度的从轴 (轴 2) 分配至原始恒定周期任务，将不需要精度的从轴 (轴 3) 分配至固定周期任务 (执行优先度 5) 进行控制。并将主轴 (轴 1) 分配至原始恒定周期任务。

#### ● 物理轴构成



● 逻辑轴构成



通过为原始恒定周期任务和固定周期任务 (执行优先度 5) 分别编写程序, 实现上述应用的动作。

MC\_PeriodicSyncVariables(轴变量任务间固定周期同步) 指令的详情请参阅  “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)”。

## 9-3 单轴速度控制

下面对单轴速度控制的动作进行说明。

关于使用 NX 系列 脉冲输出单元时的差异, 请参阅  “NX 系列 位置接口单元 用户手册 (SBCE-374)”。

### 9-3-1 速度控制

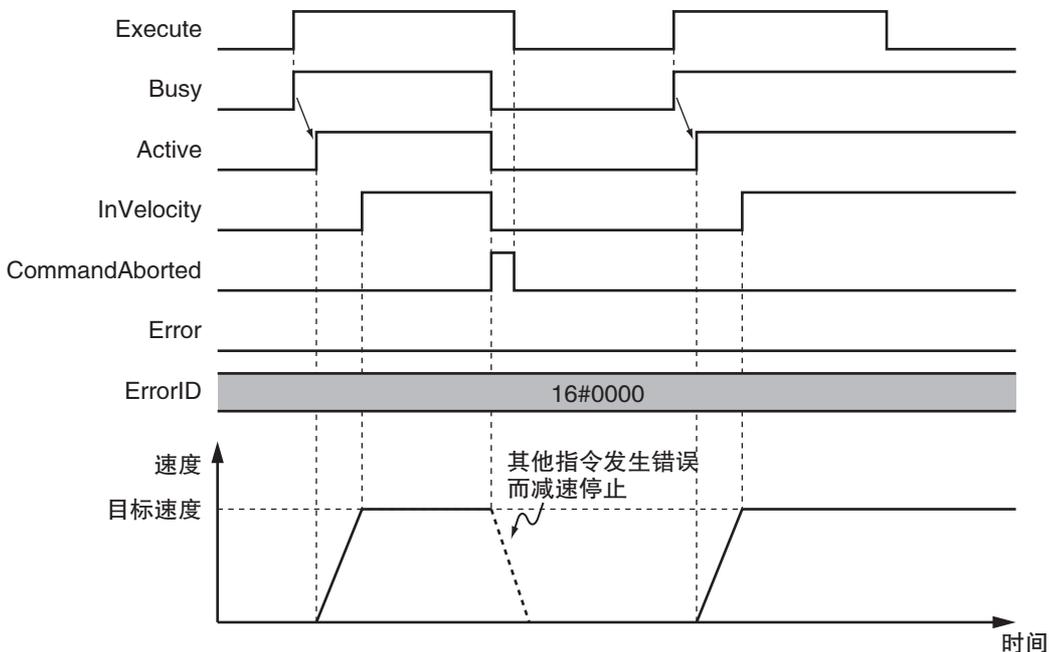
速度控制是指以指定速度连续移动轴的功能。

也可指定加速度、减速度及跃度。

需停止轴时, 应使用 MC\_Stop(强制停止) 指令或启动其他动作指令。

目标速度指定为“0”时, 轴不会移动, 但轴处于正在动作的状态。

速度控制过程中, 如果多重启动其他运动控制指令后发生中断, 则达到目标速度后切换动作。



在 MC 功能模块中指定目标位置, 以达到将伺服驱动器指定为位置控制模式的目标速度。

伺服驱动器中启用位置控制循环, 因此因外部干扰等导致比指令速度慢, 位置偏差变大时, 会发生速度变动以消除位置偏差。

功能的详情请参阅  “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 中的 MC\_MoveVelocity(速度控制) 指令。

## 9-3-2 周期同步速度控制

将伺服驱动器的控制模式作为速度控制模式，按控制周期输出指令速度的功能。



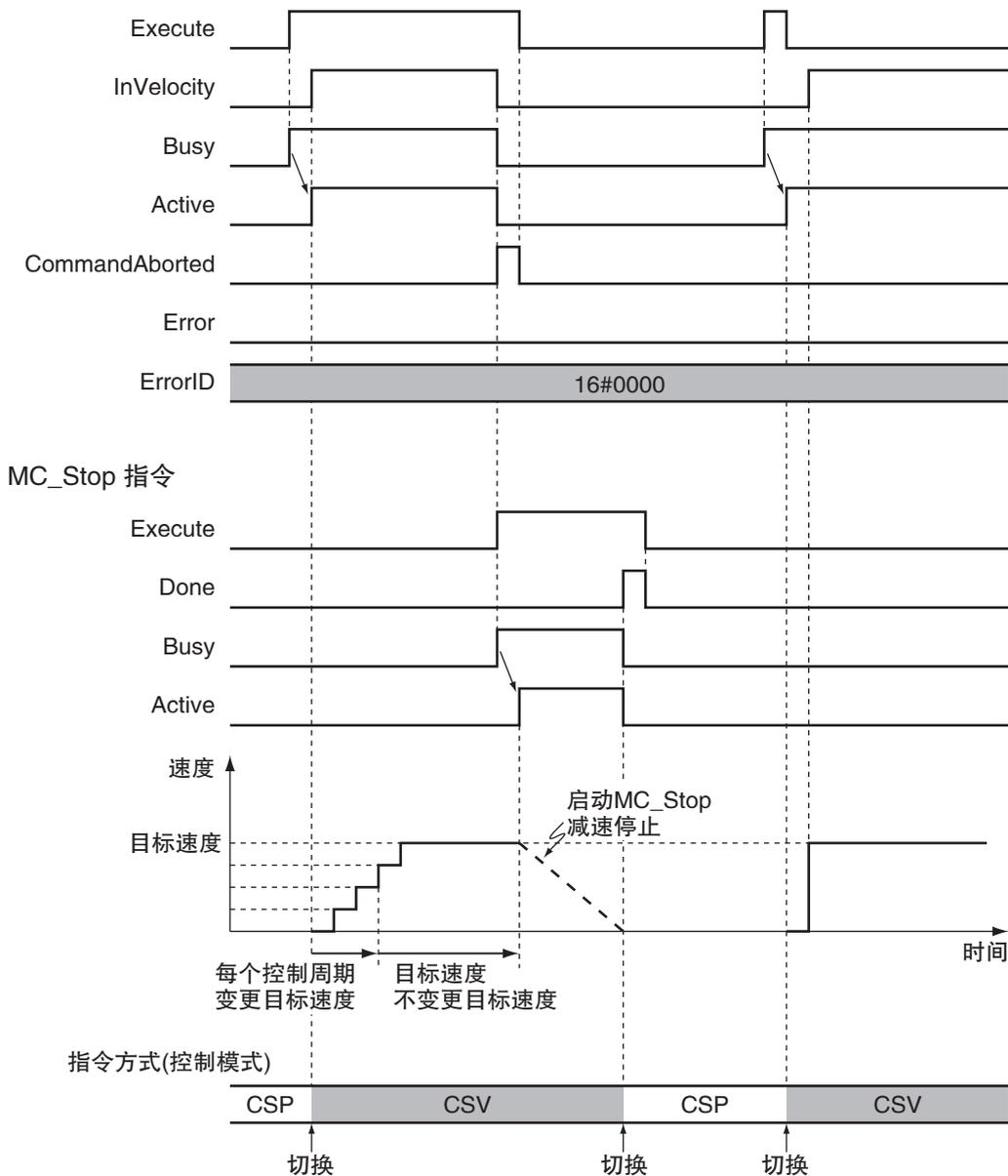
## 使用注意事项

使用 NX 系列脉冲输出单元时，无法使用该功能。

需停止轴时，应使用 MC\_Stop(强制停止) 指令或启动其他动作指令。

目标速度指定为“0”时，轴不会移动，但轴处于正在动作的状态。

## MC\_SyncMoveVelocity命令



伺服驱动器中执行速度控制循环以接受指令，因此即使因外部干扰等导致比指令速度慢，也不会发生速度变动以消除位置偏差。

功能的详情请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 中的 MC\_SyncMoveVelocity(周期同步速度控制) 指令。

## 9-4 单轴转矩控制

转矩控制是指持续产生指定转矩的功能。

可通过 TorqueRamp( 转矩斜度 ) 指定达到 Torque( 目标转矩 ) 前的转矩变化率。



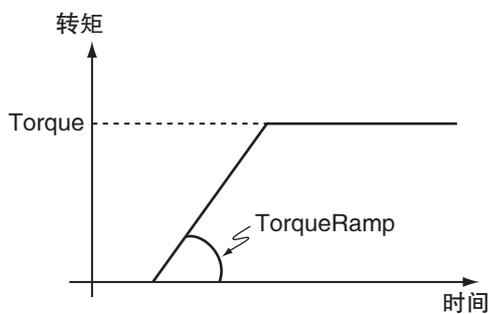
### 使用注意事项

- 进行转矩控制时，为安全起见请务必设定速度限制值。
- 使用 NX 系列脉冲输出单元时，无法使用该功能。

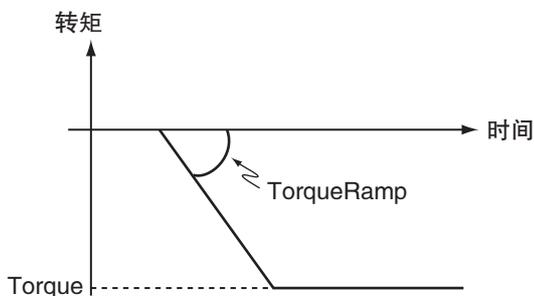
需停止轴时，应使用 MC\_Stop( 强制停止 ) 指令或启动其他动作指令。

Torque( 目标转矩 ) 指定为 “0” 时，轴不会移动，但轴处于正在动作的状态。

#### ● 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转矩控制模式。

伺服驱动器中通过转矩控制循环接收 MC 功能模块发送的转矩指令值进行控制。

可通过运动控制指令的输入变量 “Velocity( 速度限制 )” 指定伺服驱动器的 “速度限制值”。通过以上操作，可通过转矩控制模式在电机负载变小等时限制电机的高速旋转。

功能的详情请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 中的 MC\_TorqueControl( 转矩控制 ) 指令。

## 9-5 单轴控制的通用功能

下面对单轴控制的通用功能进行说明。

关于使用 NX 系列 脉冲输出单元时的差异, 请参阅 □ “NX 系列 位置接口单元 用户手册 (SBCE-374)”。

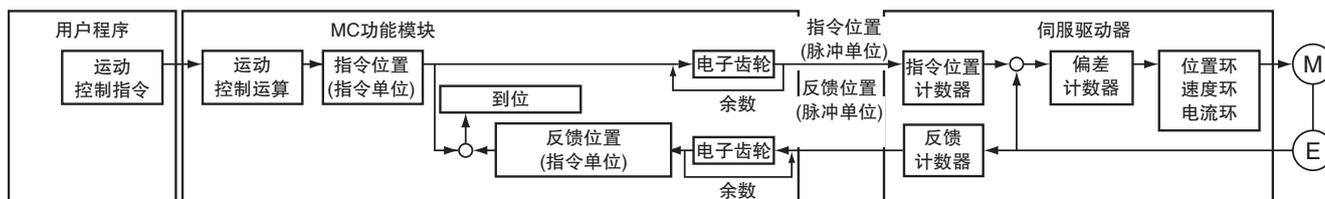
### 9-5-1 位置

#### 位置的种类

MC 功能模块中含有以下 2 种位置。

位置类型	定义
指令位置	用于控制轴的由 MC 功能模块输出的位置。
反馈位置	从伺服驱动器或计数器输入的实际位置。

对于 EtherCAT 从站伺服驱动器, 指令位置与反馈位置的关系如下图所示。



指令位置与反馈位置的以下项目相同。

项 目	指令位置	反馈位置
计数模式	设定线性模式或旋转模式。	与指令位置的计数模式相同。
位置的单位	从 mm、 $\mu\text{m}$ 、nm、inch、degree、pulse 中设定任意一个。	与指令位置的单位相同。
软件限制	设定软件可操作的范围。	与指令位置的范围相同。
当前位置变更	将当前位置变更为任意位置。	同时设定与指令位置相同的值。 <sup>*1</sup>
原点确定	具有原点确定或未确定的状态。	与指令位置的原点状态相同。

\*1. 如果变更前存在位置偏差, 则将位置偏差值的保持值设定至反馈位置。



#### 参考

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

## 位置相关的轴参数

参数名称	功能	设定范围	初始值
到位宽度	设定定位完成宽度。 (单位: 指令单位)	双精度实数型 正数、0	10
到位检查时间	以毫秒为单位, 设定定位完成的检查时间。 设定为“0”时, 仅原点复位时的原点确定时, 持续定位完成的检查。非原点复位时, 不作定位完成检查。 (单位: ms)	0 ~ 10000	0
软件限制功能	选择软件限制的功能。 0: 禁用 1: 对指令位置启用, 减速停止 2: 对指令位置启用, 立即停止 3: 对反馈位置启用, 减速停止 4: 对反馈位置启用, 立即停止	0 ~ 4	0
正方向软件限制	设定正方向侧的软件限制值。(单位: 指令单位)	双精度实数型 负数、正数、0	2147483647
负方向软件限制	设定负方向侧的软件限制值。(单位: 指令单位)	双精度实数型 负数、正数、0	-2147483648
位置偏差超过值	设定超过位置偏差的检查值。 设定为“0”时, 超过位置偏差的检查无效。 (单位: 指令单位)	双精度实数型 正数、0	0
位置偏差警告值	设定位置偏差警告的检查值。 设定为“0”时, 位置偏差警告检查无效。 (单位: 指令单位)	双精度实数型 正数、0 (位置偏差超过 值以下)	0

## 轴动作的目标位置指定

通过运动控制指令的输入变量“Position(目标位置)”、“Distance(移动距离)”指定实际定位动作的位置和距离。

## 位置的监控

可通过用户程序读取轴变量, 监控位置。

另外, 变量名称以 `_MC_AX[*]` 为例进行说明, `_MC1_AX[*]`、`_MC2_AX[*]` 同理。

变量名称	数据类型	名称	功能
<code>_MC_AX[0-255].Cmd.Pos</code>	LREAL	指令当前位置	指令位置的当前值。 伺服 OFF 或非位置控制模式时, 输出反馈当前位置。
<code>_MC_AX[0-255].Act.Pos</code>	LREAL	反馈当前位置	反馈位置的当前值。

## 9-5-2 速度

### 速度的种类

MC 功能模块使用的轴的速度分为以下 2 种。

速度种类	定义
指令速度	用于控制轴的由 MC 功能模块输出的速度。
反馈速度	根据通过伺服驱动器和计数器输入的反馈位置，在 MC 功能模块内计算出的速度。 <sup>*1</sup>

\*1. 将反馈速度 (606CHex) 映射至 PDO 映射，分配至“反馈当前速度”功能时显示该值。

### 速度的单位

速度的单位为“指令单位 /s”。

指令单位是通过位置的显示单位和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。 通过轴动作指令指定的目标速度超过最高速度时，以最高速度动作。	双精度实数型 正数	40000000
启动速度 <sup>*1</sup>	设定各轴的启动速度。 请勿超过最高速度。 (单位：指令单位 /s)	双精度实数型 正数	
微动最高速度	设定各轴的微动最高速度。 <sup>*2</sup> 请勿超过最高速度。 (单位：指令单位 /s)	双精度实数型 正数	1000000
速度警告值	设定对最高速度 (用于输出各轴的速度警告) 的比例。 设定为“0”时，不输出速度警告。(单位：%)	0 ~ 100	0
反馈速度 滤波器时间常数	以毫秒为单位，设定反馈速度移动平均值的计算时间。 设定为“0”时，不计算移动平均值。(单位：ms) 需要抑制轴的速度较慢时反馈当前速度的偏差时使用。	0 ~ 100	0

\*1. Ver.1.05 以上的 CPU 单元和 Ver.1.06 以上的 Sysmac Studio 组合时可使用。

\*2. 指定超过微动最高速度的速度指令值时，以微动最高速度发出指令。

### 轴动作的目标速度指定

实际的定位动作的速度通过运动控制指令的输入变量“Velocity(目标速度)”指定。

## 速度的监控

可通过用户程序读取轴变量，监控速度。

另外，变量名称以 `_MC_AX[*]` 为例进行说明，`_MC1_AX[*]`、`_MC2_AX[*]` 同理。

变量名称	数据类型	名称	功能
<code>_MC_AX[0-255].Cmd.Vel</code>	LREAL	指令当前速度	指令速度的当前值。 向正方向移动带正号，向负方向移动带负号。
<code>_MC_AX[0-255].Act.Vel</code>	LREAL	反馈当前速度	反馈速度的当前值。 向正方向移动带正号，向负方向移动带负号。

### 9-5-3 加速与减速

#### 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<sup>2</sup>”。

指令单位是通过位置的显示单位和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。 设定为“0”时，无加速度限制。(单位：指令单位/s <sup>2</sup> )	双精度实数型 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。 设定为“0”时，无减速度限制。(单位：指令单位/s <sup>2</sup> )	双精度实数型 正数、0	0
加减速超限	在轴的加减速控制中，优先向目标位置停止后，产生减速超限。指定超过最大加减速度的动作。 0：提高加减速速度(将合并切换为等待)*1 1：提高加减速速度 2：异常停止*2	0 ~ 2	0
加速度警告值	设定对最高加速度(用于输出各轴的加速度警告)的比例。 设定为“0”时，不输出加速度警告。(单位：%)	0 ~ 100	0
减速度警告值	设定对最高减速度(用于输出各轴的减速度警告)的比例。 设定为“0”时，不输出减速度警告。(单位：%)	0 ~ 100	0

\*1. Ver.1.10 以上的 CPU 单元，无法将合并切换为等待。详情请参阅 [□ “9-5-7 运动控制指令的多重启动\(缓存模式\)\(P.9-44\)”](#)。

\*2. Ver.1.10 以上的 CPU 单元，合并动作时不会异常停止。详情请参阅 [□ “9-5-7 运动控制指令的多重启动\(缓存模式\)\(P.9-44\)”](#)。

## 轴动作的加速度和减速度的指定

通过运动控制指令的输入变量“Acceleration(加速度)”、“Deceleration(减速度)”指定实际定位动作的加速度和减速度。

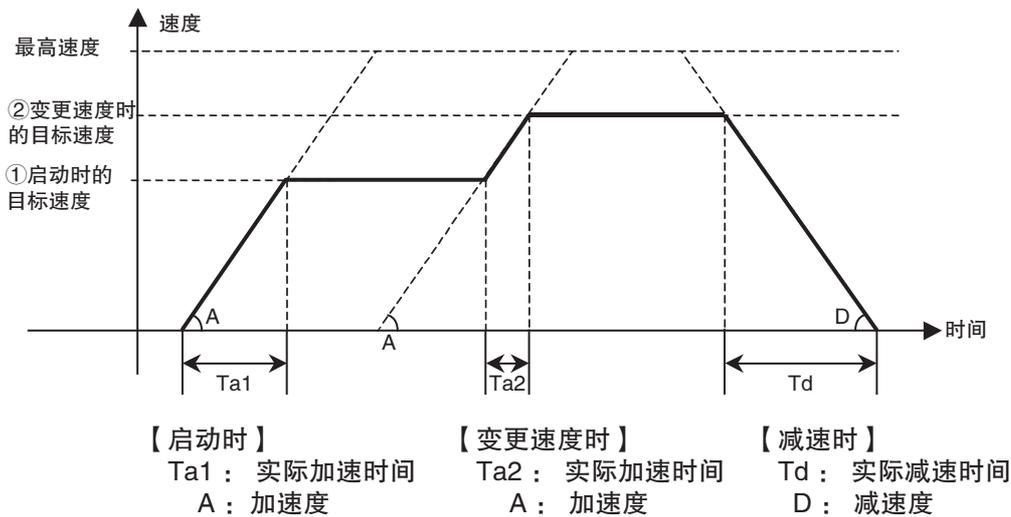
## 加速度和减速度的监控

通过用户程序读取轴变量后，可监控加速度和减速度。

另外，变量名称以 `_MC_AX[*]` 为例进行说明，`_MC1_AX[*]`、`_MC2_AX[*]` 同理。

变量名称	数据类型	名称	功能
<code>_MC_AX[0~255].Cmd.AccDec</code>	LREAL	指令当前加减速速度	指令加减速度的当前值。 加速时附带正号，减速时附带负号。

## 加减速动作示例



指定的移动量较小或加减速速度较低时，可能无法达到目标速度。

如果重启运动控制指令后以变更的新加速度(减速度)动作，则超过目标位置时，以可在目标位置停止的加速度或减速度执行定位。

### 9-5-4 跃度

跃度指定加速度 / 减速度的变化比率。指定跃度后，加减速时的速度波形为 S 形，可减小机械冲击。



#### 参考

跃度也称为“跃动”或“加加速度”。

### 跃度的单位

跃度的单位为“指令单位 / s<sup>3</sup>”。

指令单位是通过位置的显示单位和电子齿轮进行单位转换后的值。

### 轴动作的跃度的指定

实际的定位动作的跃度通过运动控制指令的输入变量“Jerk(跃度)”指定。

加速和减速的值相同。

将按下式计算得出的值设定为跃度。

公式：跃度 = 加速度 / (加速时间 × 加速时间中适用于跃度的时间的比例 / 2)

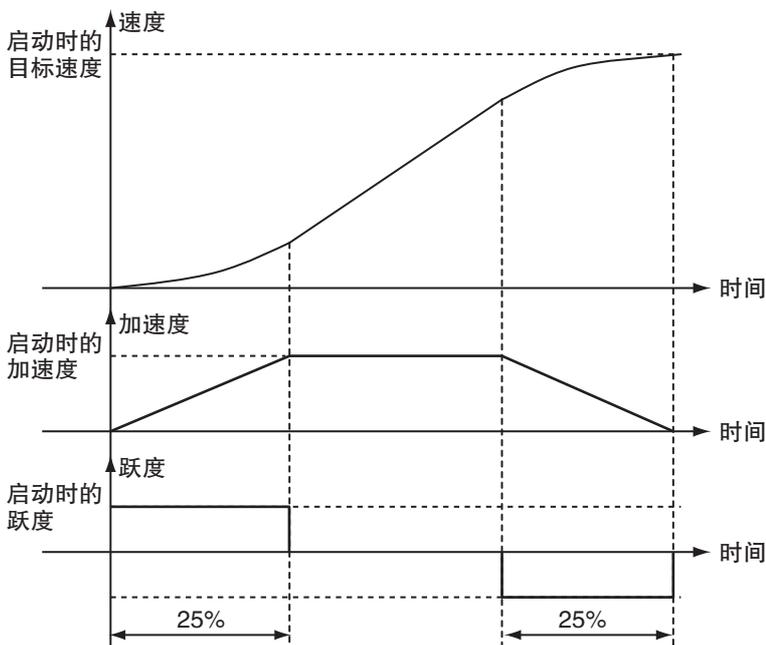
适用于跃度的时间分为自加速开始起的时间和到加速结束为止的时间 2 等分。因此，适用于跃度的时间的比例需除以 2。

#### ● 指定跃度后的加速控制示例

跃度适用的区间的加速度以固定比率变化，因此指令速度为平滑的 S 形。跃度为“0”的区间的加速度固定，因此指令速度为直线形。

例：加速度为 25,000mm/s<sup>2</sup>，加速时间为 0.1s，跃度适用的时间比例为 50% 时

$$\text{跃度} = 25000 / (0.1 \times 0.5 / 2) = 1,000,000 \text{ (mm/s}^3\text{)}$$



## 跃度的监控

可通过用户程序读取轴变量，监控跃度。

另外，变量名称以 `_MC_AX[*]` 为例进行说明，`_MC1_AX[*]`、`_MC2_AX[*]` 同理。

变量名称	数据类型	名称	功能
<code>_MC_AX[0-255].Cmd.Jerk</code>	LREAL	指令当前跃度	指令跃度的当前值。

### 9-5-5 指定动作方向

需通过索引表等指定附近等的旋转方向执行动作时，应将计数模式设定为旋转模式。还应通过需指定绝对位置的运动控制指令的输入变量“Direction(选择方向)”指定动作方向。

方向选择为“指定为附近”、“指定为正方向”、“指定为负方向”、“指定为当前方向”时，可将不足环计数器1圈、环计数器上下限值以内的位置指定为目标位置。

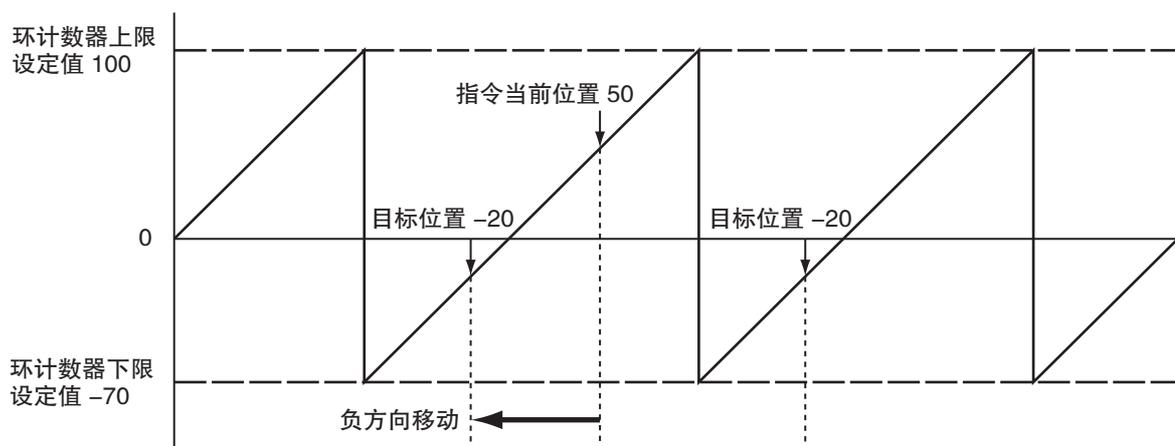
计数模式为线性时，无论方向如何选择，直接向目标位置进行定位。

MC 功能模块可指定的方向如下所示。

方向选择	动作
指定为附近	向指令当前位置与目标位置之间的距离较短的方向开始动作。
指定为正方向	向正方向开始动作。
指定为负方向	向负方向开始动作。
指定为当前方向	向与上次动作相同的方向开始动作。
无方向指定	向不通过环形计数器上下限值的方向开始动作。 选择方向时可指定超过环形计数器上下限值的目标位置。此时的动作表现为以目标位置与指令当前位置之差为目标距离的相对值定位。通过以上操作，可向环形计数器的多圈位置进行定位。

## 指定为附近时的示例

以下是指令当前位置为“50”时向目标位置“-20”进行定位的示例。

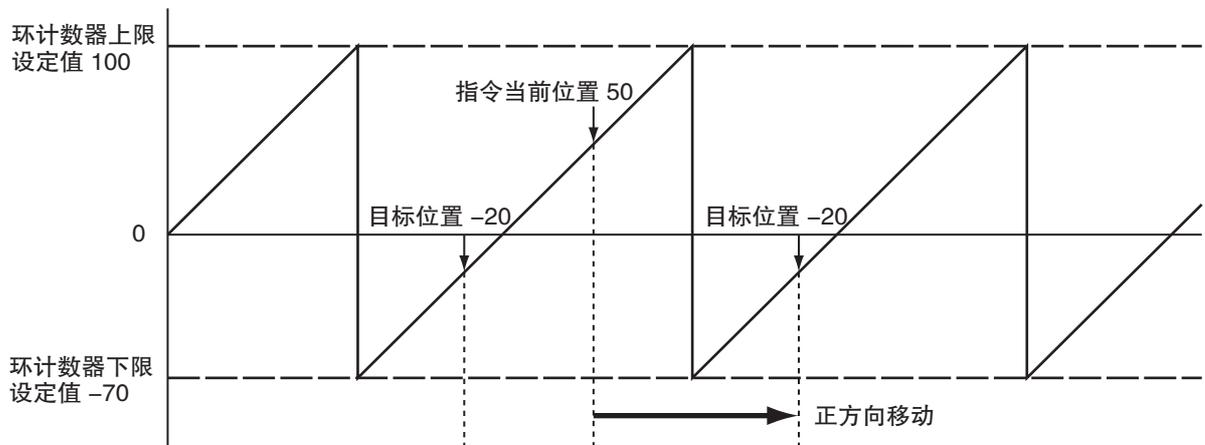


#### 参考

正方向、负方向移动距离都相同时，动作与“指定为当前方向”时相同。

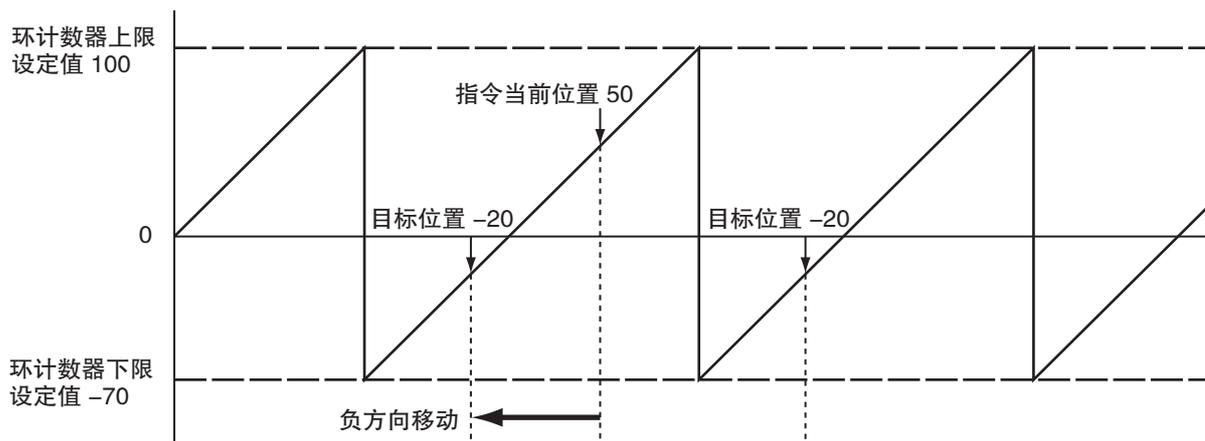
### 指定为正方向时的示例

以下是指令当前位置为“50”时向目标位置“-20”进行定位的示例。



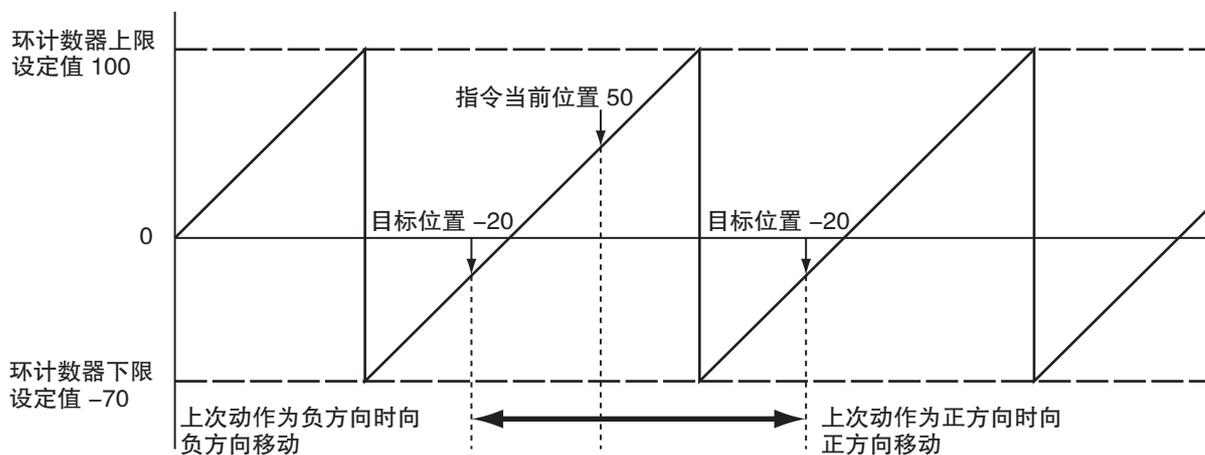
### 指定为负方向时的示例

以下是指令当前位置为“50”时向目标位置“-20”进行定位的示例。



### 指定为当前方向时的示例

以下是指令当前位置为“50”时向目标位置“-20”进行定位的示例。



上次的动作方向可通过轴变量的“指令方向状态”进行判定。



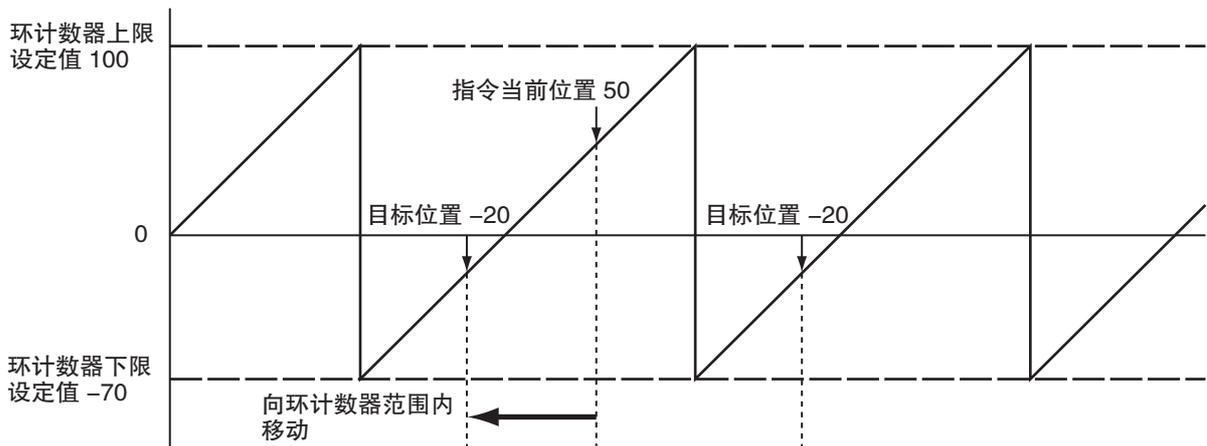
使用注意事项

上次的动作属于以下情况时，请注意动作方向。

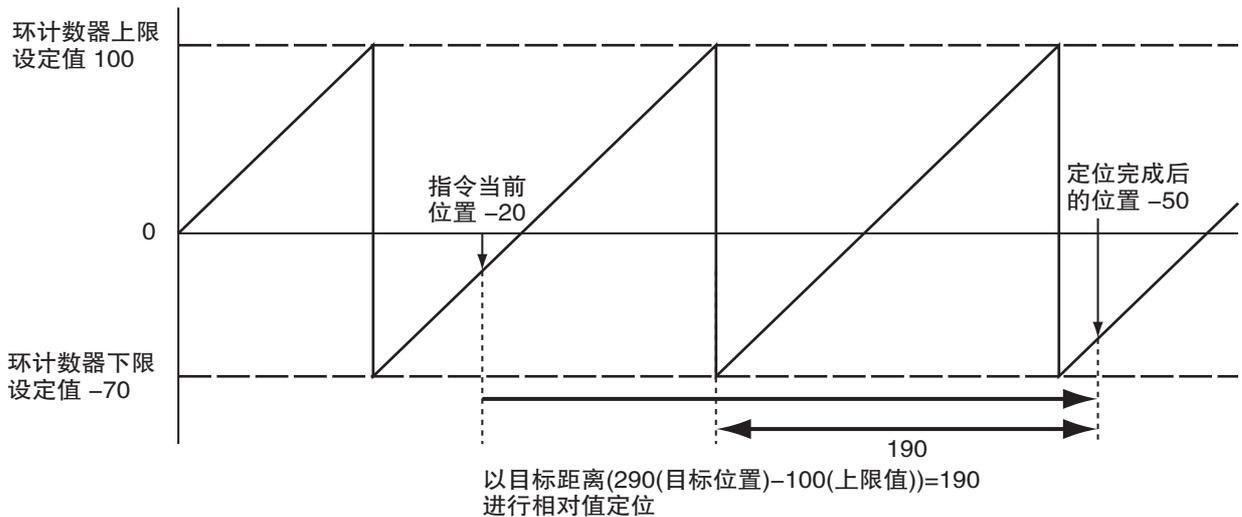
- 通过 MC\_Home(原点复位)指令或 MC\_HomeWithParameter(参数指定原点复位)指令超过检测到原点输入的位置并执行取反动作时，动作方向与原点检测方向相反。
- 通过 MC\_Home(原点复位)指令或 MC\_HomeWithParameter(参数指定原点复位)指令设定原点复位补偿值后，动作方向为向补偿值动作的方向。
- 通过 MC\_TouchProbe(启用外部锁定)指令指定立即停止后，超过锁定位置并执行取反动作时。
- 通过 MC\_MoveFeed(中断标准定位)指令执行取反动作时。
- 执行 MC\_ResetFollowingError(偏差计数器复位)指令后，动作方向为使偏差为“0”的指令位置的方向。
- 通过外部输入信号的立即停止和极限输入停止指定偏差计数器复位后，向发生外部输入信号的位置执行取反动作时。

无方向指定时的示例

以下是指令当前位置为“50”时向目标位置“-20”进行定位的示例。



以下是环形计数器上限值为“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

下面对单轴动作中变更与运动控制指令相同的实例的输入变量并重启进行说明。

重启运动指令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( 目标转矩 )。

如果变更无法变更的输入变量后重启，则指令会发生异常。

多重启动运动控制指令后重启缓存的指令，缓存中的指令的输入变量会发生变化。

可变更的输入变量的详情请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 中的各指令。

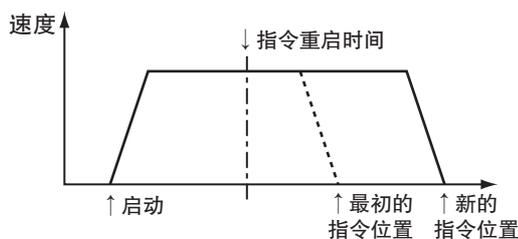
### 目标位置的变更

重启后变更目标位置时，动作因变更的目标位置和时间而异。

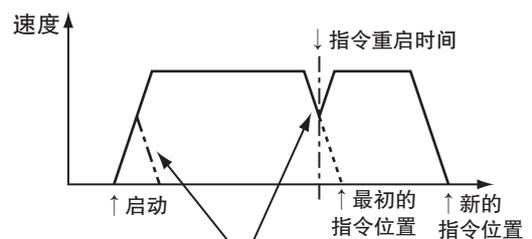
变更目标位置后动作方向取反时的动作可在轴参数的 [ 取反时动作 ] 中选择 [ 取反时减速停止 ] 或 [ 取反时立即停止 ]。

#### ● 不因新的指令值而取反时

等速动作中的重启



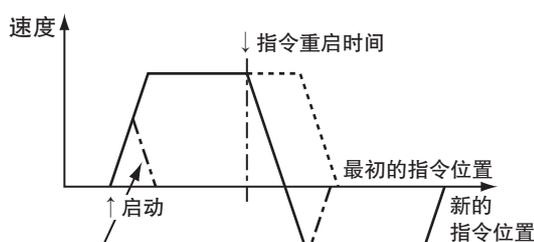
加减速动作中的重启



三角控制或减速中重启时，再次加速至目标速度。此时，也可能不加速至目标速度。

#### ● 因新的指令值而取反时

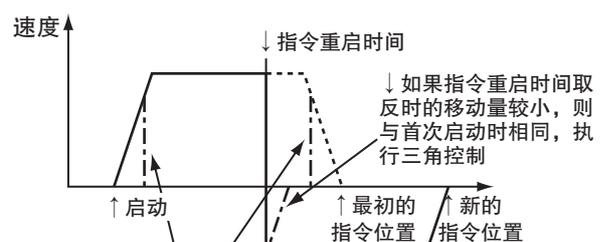
取反时减速停止设定



如果在加速中重启，则在重启时开始减速

如果取反时的移动量较小，则与首次启动时相同，执行三角控制

取反时立即停止设定

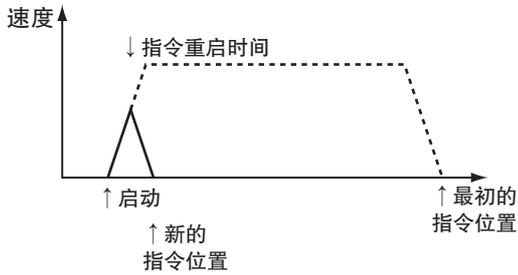


如果在加减速中重启，则在重启时立即停止。减速中也相同

● 三角控制的模式

移动距离因目标位置的变更而变短时，可能会变为如下图所示的三角控制。

不取反时



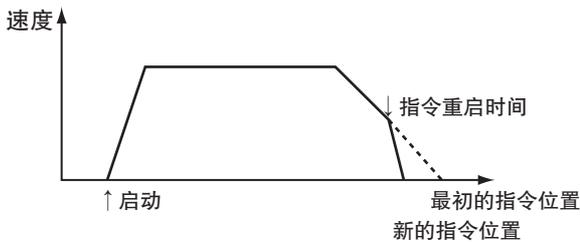
● 减速超限的模式

以下情况下会优先在目标位置停止，因此减速度会超过指定的减速度

减速度超过轴参数的“最大减速度”时，执行通过轴参数设定的“加减速超限”设定的动作。

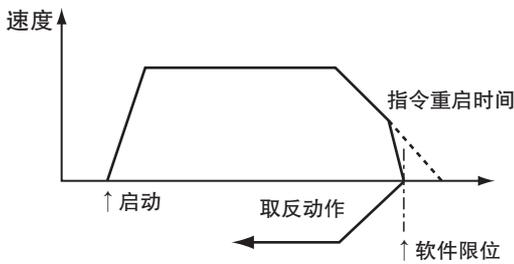
- 动作不取反，因指定的减速度而超过目标位置时

不取反时



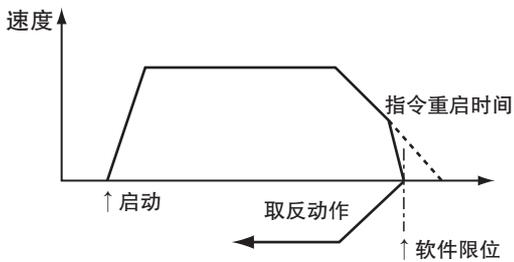
- 动作取反，因减速停止而超过软件限制时

不取反时



- 动作取反，因减速停止而使指令当前位置发生溢出或下溢时

不取反时

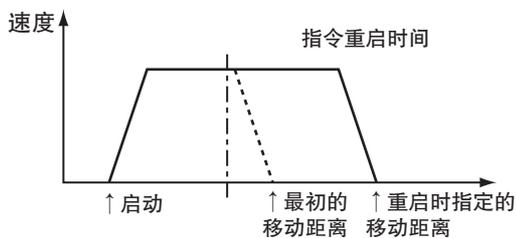


## 移动距离的变更

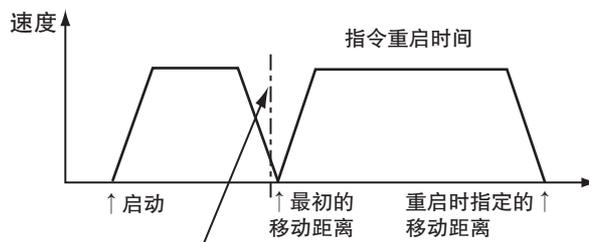
通过 MC\_MoveRelative(相对值定位)指令变更移动距离并重启后仍以根据开始移动的位置变更的移动距离进行定位。

如果在定位即将完成前重启，则可能作为新指令的启动进行移动。

动作中的重启



定位即将完成前的重启



如果在定位即将完成前重启，则有时会根据以最初的移动距离定位的位置，在重启时移动指定的移动距离。



### 使用注意事项

请勿在定位即将完成前重启以变更移动距离。

## 目标速度的变更

加速、等速或三角控制下的加速时，目标速度的动作会发生变化。

减速时也可进行变更，但不影响实际动作。

## 加速度的变更

加速、等速或三角控制下的加速时，加速度的动作会发生变化。

如果在等速时进行变更，则会因超调等而加速。

减速时也可进行变更，但不影响实际动作。

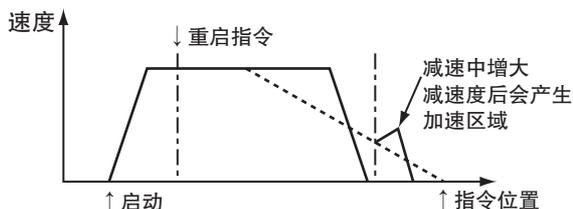
## 减速度的变更

加速、等速、减速、三角控制或减速超限控制时，减速度会发生变化。

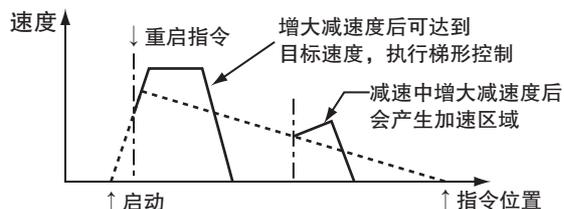
因变更后的减速度而超过目标位置时，优先在目标位置停止，因此减速度会超过指定的减速度。

### ● 减速度增大模式

梯形控制

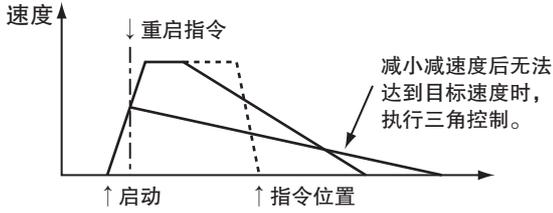


三角控制→梯形控制

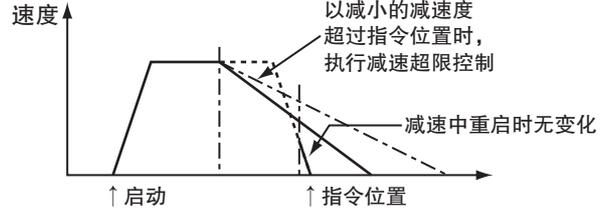


● 减速度减小模式

梯形控制 · 三角控制



减速超限控制



转矩指令的变更

重启运动控制指令后，根据转矩斜度的指定变更转矩指令值。

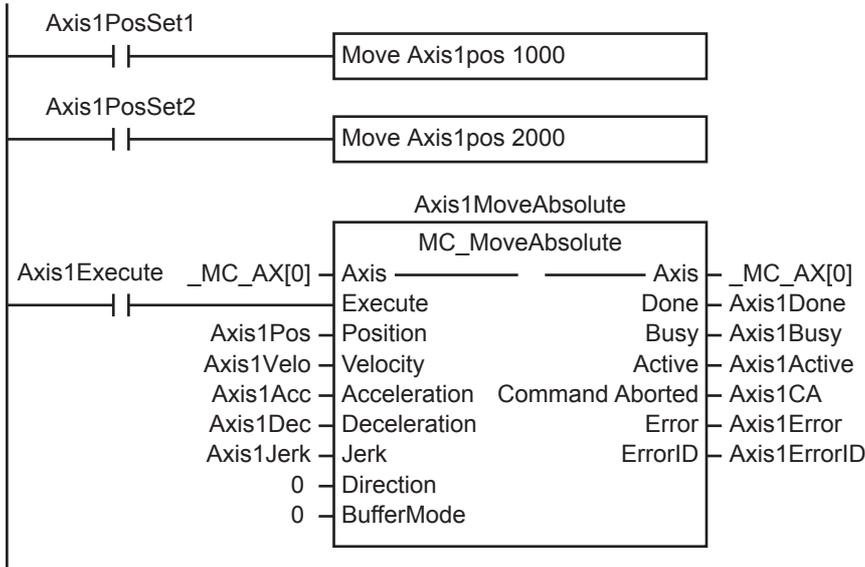
重启的程序示例

以下为通过绝对值定位将目标位置从“1000”变更为“2000”的示例。

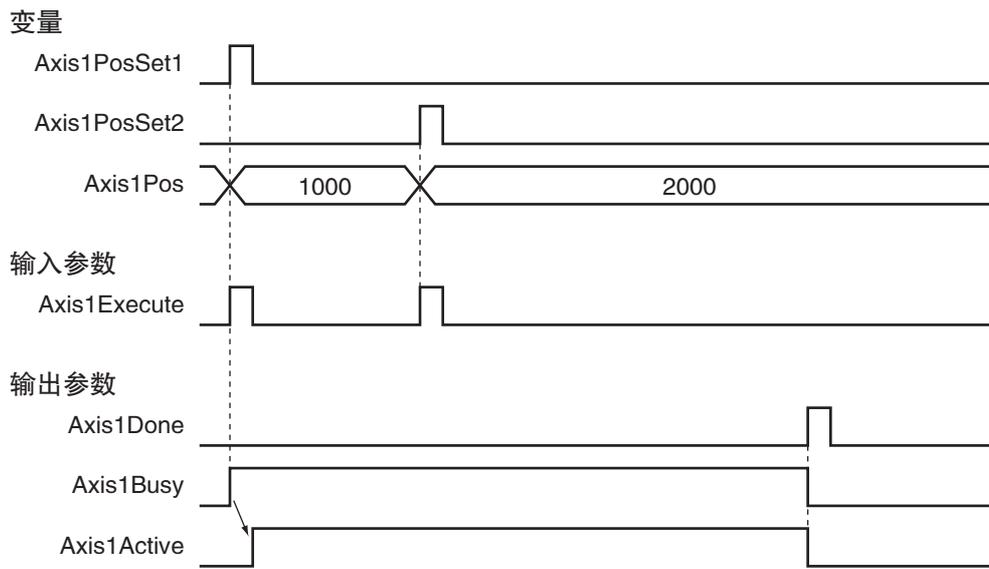
目标位置的输入参数使用变量“Axis1Pos”。

通过 MOV 指令将目标位置设定为“1000”，将 Axis1Execute 从 FALSE 设为 TRUE 时，开始定位。

动作中将目标位置设定为“2000”，将 Axis1Execute 再次从 FALSE 设为 TRUE，变更为向目标位置“2000”定位的动作。



## ● 时序图



### 使用注意事项

不变更的输入变更请保持重启前的值。

### 9-5-7 运动控制指令的多重启动 (缓存模式)

轴动作中可启动其他运动控制指令。

在 PLCopen® 技术规格书中定义为缓存模式，在 MC 功能模块中作为“多重启动”进行说明。使用多重启动后不停止多个运动控制指令的动作即可连续执行动作。

在 MC 功能模块中，多重启动使用以下用语。

用 语		含 义
本手册	PLCopen®	
当前指令	以前的 FB	执行多重启动前动作的运动控制指令
缓存指令	下一个 FB	轴动作中启动并处于执行等待状态的运动控制指令
中继速度	合并	指定合并时向当前指令指定的目标位置动作时使用的指令速度

指定运动控制指令的输入变量“BufferMode(选择缓存模式)”后可选择以下多重启动模式。上述模式的主要差异在于多重启动指令的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
等待	当前指令的动作正常结束后执行缓存指令。
合并	到达当前指令的目标位置后执行缓存指令。 该模式下，在当前指令与缓存指令之间不会停止。 当前指令到达目标位置时的中继速度可从以下几种进行选择。
以低速合并 (低速)	将当前指令和缓存指令的目标速度中较低的速度作为中继速度
以前一个速度合并 (前一个速度)	以当前指令的目标速度为中继速度。
以后一个速度合并 (后一个速度)	以缓存指令的目标速度为中继速度。
以高速合并 (高速)	将当前指令和缓存指令的目标速度中较高的速度作为中继速度

等待和合并时缓存在 MC 功能模块内，以指定的多重启动模式的时间和中继速度执行多重启动的指令。各轴均带有 1 个该缓存。

中断时立即执行多重启动的指令，因此不缓存指令。



#### 使用注意事项

- 1 个轴只能缓存 1 个多重启动。如果多重启动 2 个以上的指令，则指令会发生异常。
- 无法对单轴控制下动作的轴多重启动多轴协调控制指令 (轴组指令)。无法对多轴协调控制 (轴组指令) 下动作的轴多重启动单轴控制指令 (轴指令)。如果在上述情况下执行多重启动，则指令会发生异常。

## 中断

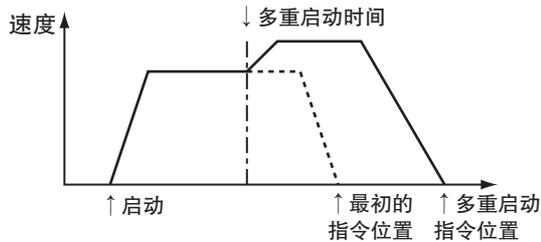
不缓存的默认模式。

立即中断执行中的指令，执行多重启动的新指令。

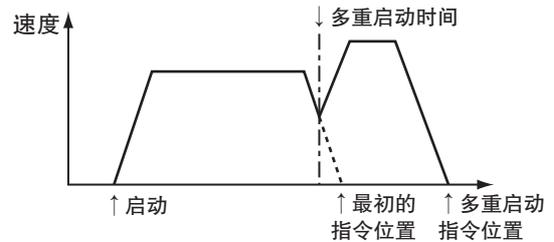
可通过单轴控制和同步控制的运动控制指令多重启动中断。

### ● 不对多重启动指令位置执行取反时

等速动作中的多重启动



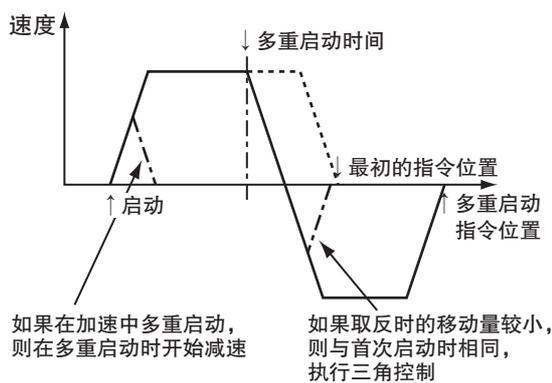
加减速动作中的多重启动



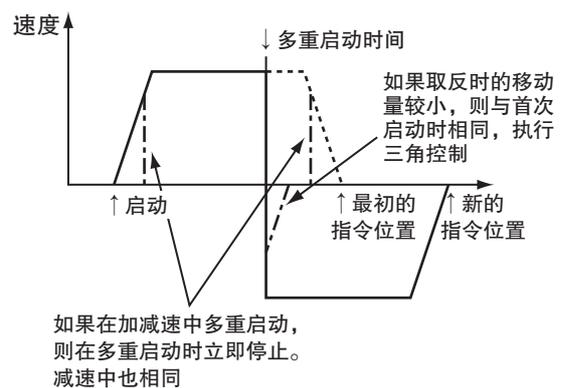
三角控制或减速中执行多重启动后会加速至缓存指令的目标速度。此时，也可能不加速至目标速度。

### ● 对多重启动指令值执行取反时

取反时减速停止设定



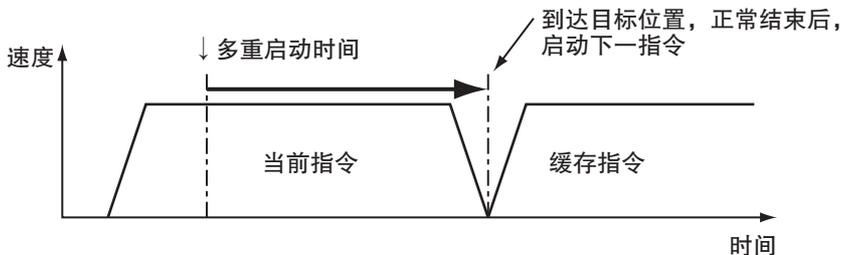
取反时立即停止设定



## 等待

执行中的当前指令的动作完成前，等待执行多重启动的指令。

当前指令的动作正常结束后执行缓存指令。



## 合并

到达当前指令的目标位置前，等待执行多重启动的指令。

到达当前指令的目标位置后执行缓存指令，此时不会停止，会以输入变量“BufferMode(缓存模式选择)”指定的速度连续执行2个动作。

执行相对值移动时，最后的到达位置为2个指令的合计值。

执行绝对值移动时，为第2个指令的目标位置。

因轴参数的“最大加速度”或“最大减速度”而超过目标位置时的动作可在轴参数的“加减速超限”中设定为以下几种。

- 提高加减速速度(将合并切换为等待)
- 提高加减速速度
- 异常停止



### 使用注意事项

在单轴控制和同步控制组合中无法执行合并。



### 版本相关信息

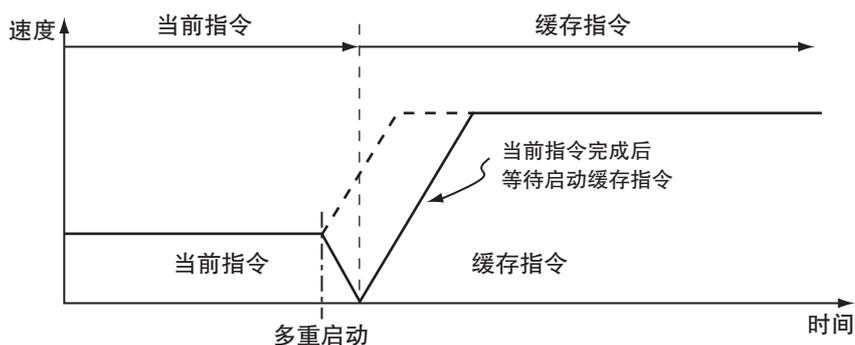
Ver.1.10 以上版本的 CPU 单元，即使设定 [ 提高加减速速度 ( 将合并切换为等待 ) ]，也不将合并切换为等待。此时以最大加减速速度加减速，继续进行合并。

此外，即使设定 [ 异常停止 ]，也不作为异常停止。此时也以最大加减速速度加减速，继续进行合并。

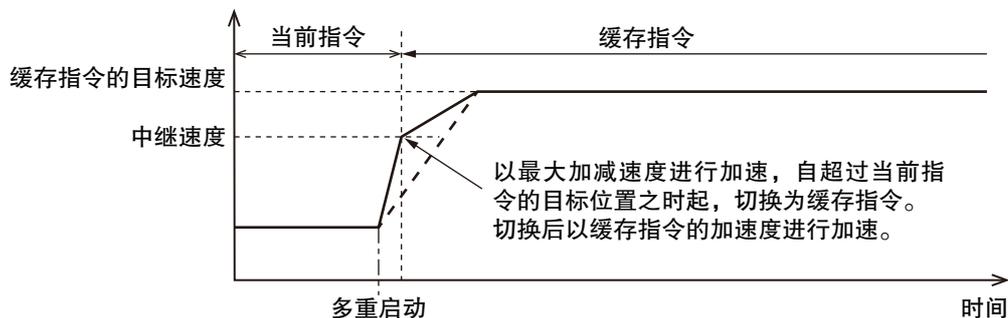
“加减速超限”的动作示例如下所示。

#### ● 提高加减速速度 ( 将合并切换为等待 )

- Ver.1.09 以下版本的 CPU 单元时执行以下动作。

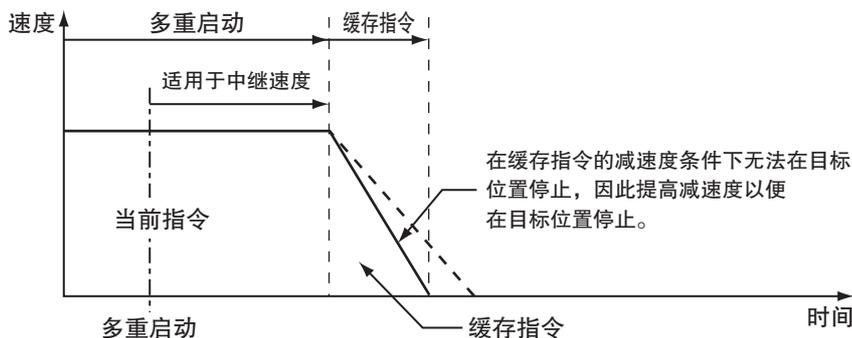


- Ver.1.10 以上版本的 CPU 单元时执行以下动作。  
此外，即使设定为 [ 异常停止 ]，也执行相同的动作。  
在此例中，BufferMode 设定为以后一个速度合并。

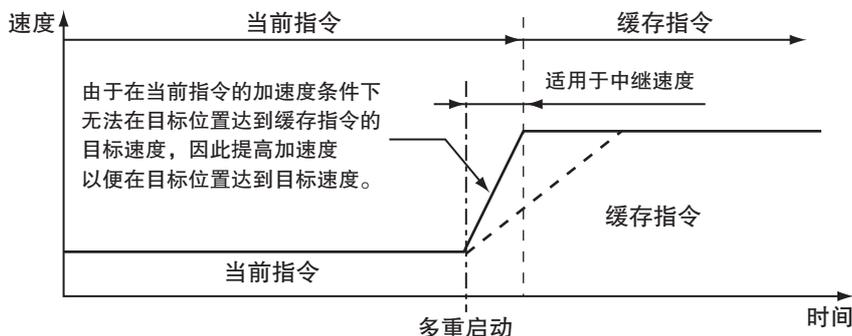


### ● 提高加减速速度

- “BufferMode(缓冲模式选择)” 设定为以前一个速度合并时, 执行以下动作。



- “BufferMode(缓冲模式选择)” 设定为以后一个速度合并时, 执行以下动作。



## 以低速合并 (低速)

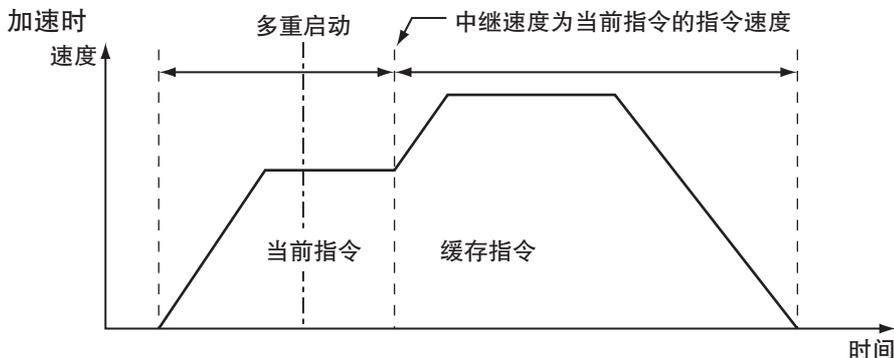
在当前指令的目标位置动作, 以达到当前指令和缓存指令中较低的目标速度。

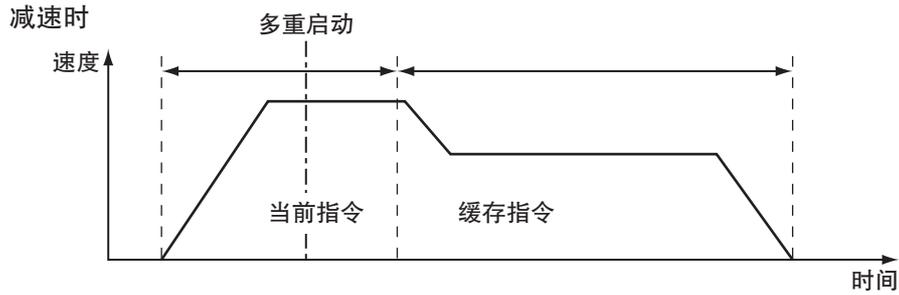
## 以前一个速度合并 (前一个速度)

到达当前指令的目标位置前, 以当前指令的目标速度动作。

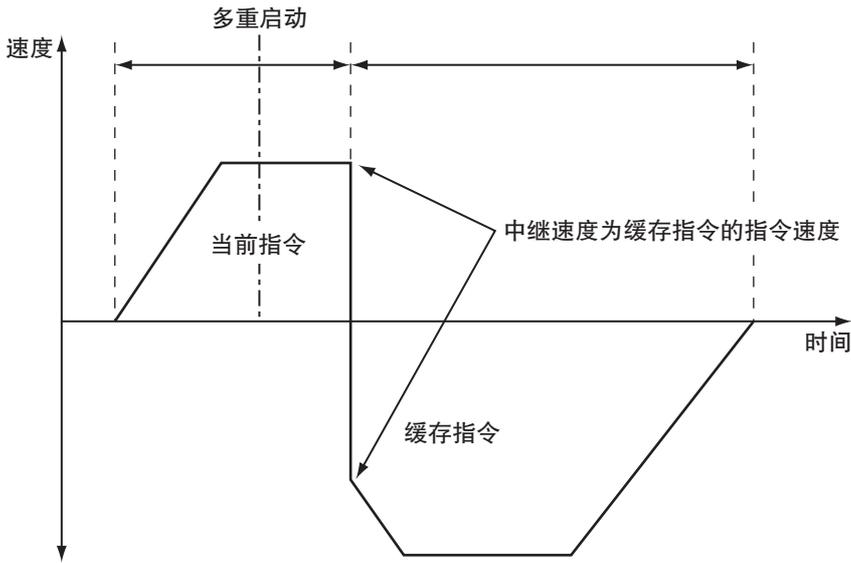
到达目标位置后, 动作时应执行加减速以达到缓存指令的目标速度。

### ● 动作方向不取反时



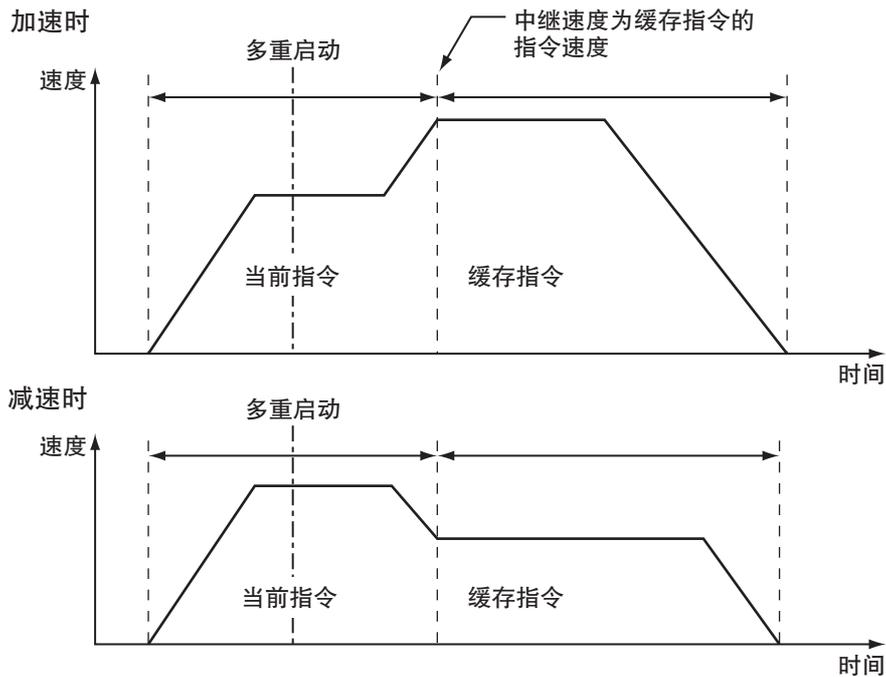


● 动作方向取反时



以后一个速度合并 (后一个速度)

在当前指令的目标位置动作，以达到缓存指令的目标速度。



以高速合并 (高速)

在当前指令的目标位置动作，以达到当前指令和缓存指令中较高的目标速度。

## 9-6 多轴协调控制

下面对多轴的协调控制动作进行说明。

在 MC 功能模块中，事先通过 Sysmac Studio 设定轴组，可进行多轴的插补控制。

### 9-6-1 动作概要

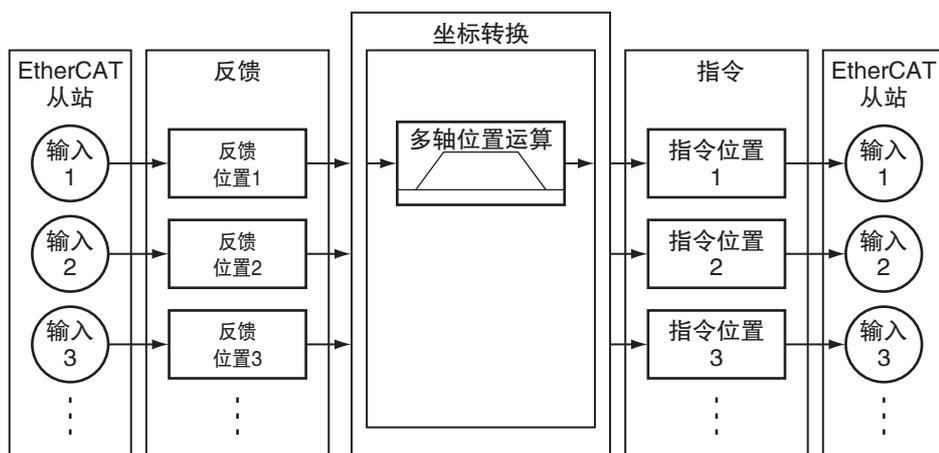
多轴协调控制是为了控制工具前端等控制对象的轨迹，将多个相关轴分组进行协调动作的功能。

在 MC 功能模块中，通过 Sysmac Studio 将执行协调动作的轴的组合设定为轴组。

通过用户程序将各轴设为伺服 ON 后，启用实际执行多轴协调控制的轴组。

多轴协调控制以轴组所属的轴的协调动作为目的，因此无法对有效轴组中的轴启动单轴运动的运动控制指令。轴组所属的轴发生异常时，轴组所属的其他轴根据轴组参数中的 [ 轴组停止方法 ] 的设定停止。

MC 功能模块中可执行 2 轴 ~ 4 轴的直线插补和 2 轴的圆弧插补。



#### 参考

对于需在动作中变更轴的组合并执行插补控制的装置，应事先通过 Sysmac Studio 创建包含需变更组合的轴的多个轴组。动作时在用户程序中指定需启用的轴组后即可实现。

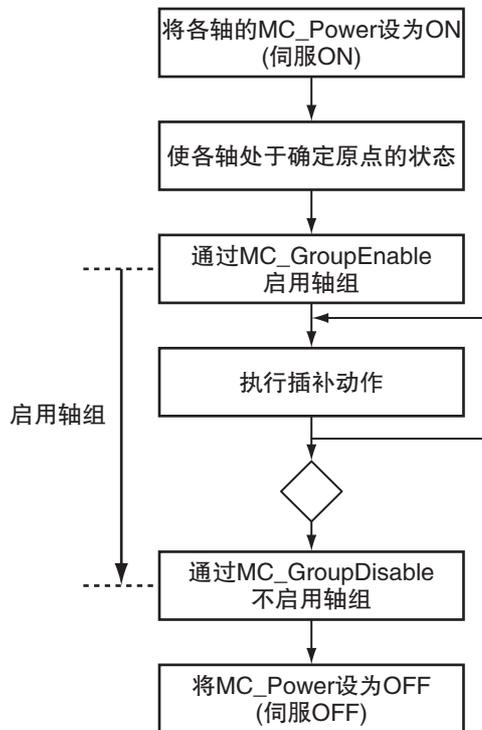
在 Ver.1.01 以上的 CPU 单元和 Sysmac Studio Ver.1.02 以上的 Sysmac Studio 的组合中，使用 MC\_ChangeAxesInGroup(写入轴组构成轴)后可改写已禁用的轴组的构成轴。

轴组的详情请参阅 □ “3-3 轴组 (P.3-19)”。

## 启用 / 不启用轴组

需启用轴组时，应通过 MC\_GroupEnable( 启用轴组 ) 指令指定需启用的轴组。如果在禁用状态下启动轴组指令，则指令会发生异常并无法启动。

需禁用处于启用状态下的轴组时，应通过 MC\_GroupDisable( 禁用轴组 ) 指令指定需禁用的轴组。如果在轴组动作中启动禁用轴组指令，则轴组所属的轴会以轴参数设定的最大减速度减速停止。



启用 / 不启用轴组的功能的详情请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 中的 MC\_Group Enable( 启用轴组 ) 指令和 MC\_GroupDisable( 不启用轴组 ) 指令。

## 写入轴组构成轴

通过 MC\_ChangeAxesInGroup( 写入轴组构成轴 ) 指令暂时改写已禁用的轴组的构成轴的功能。启用轴组时，请通过 MC\_GroupDisable( 不启用轴组 ) 指令禁用后再改写构成轴。

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



### 使用注意事项

- 通过 MC\_ChangeAxesInGroup( 写入轴组构成轴 ) 进行的变更不保存至 CPU 单元内置的非易失性存储器。重新接通电源或从 Sysmac Studio 下载后恢复非易失性存储器的参数设定内容。

写入轴组构成轴的详情请参阅 “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 中的 MC\_ChangeAxesInGroup( 写入轴组构成轴 ) 指令。

## 轴组位置获取

通过 MC\_GroupReadPosition( 获取轴组位置 ) 指令获取轴组的指令当前位置和反馈当前位置的功能。  
Ver.1.01 以上的 CPU 单元和 Ver.1.02 以上的 Sysmac Studio 组合时可使用。

轴组位置获取功能的详情请参阅 “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 中的 MC\_GroupRead Position( 获取轴组位置 ) 指令。

## 轴组错误复位

轴组发生异常时，排除异常原因后通过 MC\_GroupReset( 轴组错误复位 ) 指令解除异常。

轴组错误复位的详情请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 中的 MC\_GroupReset ( 轴组错误复位 ) 指令。

关于使用 NX 系列 脉冲输出单元时的差异，请参阅 □ “NX 系列 位置接口单元 用户手册 (SBCE-374)”。

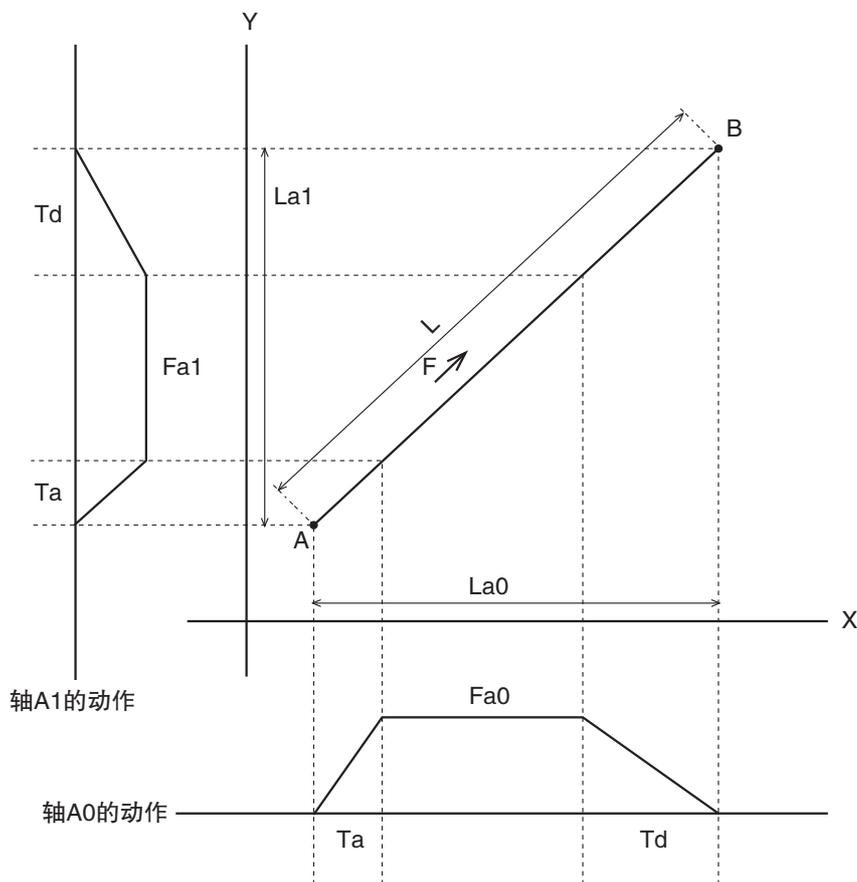
## 9-6-2 直线插补

直线插补使用轴组中轴 A0 ~ A3 的逻辑轴中的 2 轴 ~ 4 轴，定位直线的起点和重点。可执行绝对值定位和相对值定位，可指定插补速度、插补加速度、插补减速度及跃度。

MC 功能模块包括以下 3 种直线插补指令。

- MC\_MoveLinear( 直线插补 )  
指定输入变量 “MoveMode( 选择移动方法 )” 后可选择绝对值或相对值直线插补。它是 MC 功能模块独有的指令。
- MC\_MoveLinearAbsolute( 绝对值直线插补 )  
执行绝对值直线插补。它是 PLCopen® 技术规格书定义的指令。
- MC\_MoveLinearRelative( 相对值直线插补 )  
执行相对值直线插补。它是 PLCopen® 技术规格书定义的指令。

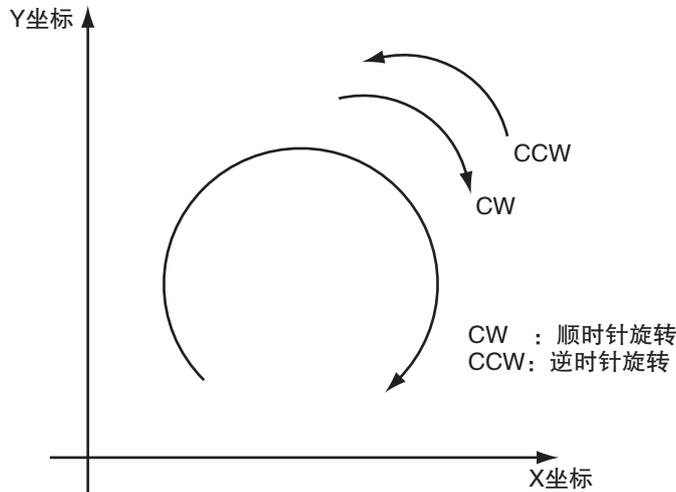
使用 2 轴的 A 点至 B 点的直线插补如下图所示。



直线插补的详情请参阅 “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 中的 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

圆弧插补使用轴组中轴 A0 ~ A3 的逻辑轴中的 2 轴，在二维平面中绘制圆弧进行定位。可执行绝对值定位和相对值定位，可指定圆弧插补模式、路径的方向、插补速度、插补加速度、插补减速度及 2 轴合成值的跃度。



MC 功能模块中可通过输入变量 “CircMode( 圆弧插补模式 )”，从以下 3 种圆弧插补方式中指进行定。

- 通过点指定
- 中心点指定
- 半径指定



#### 使用注意事项

圆弧插补使用的轴的计数模式请设为“线性模式”。  
如果以“旋转模式”启动，则指令会发生异常。

### 9-6-4 轴组周期同步位置控制

该功能周期性输出为轴组中各轴指定的目标位置。  
需以任意轨迹动作等时，应以绝对位置指定通过用户程序计算出的目标位置。  
Ver.1.01 以上的 CPU 单元和 Ver.1.02 以上的 Sysmac Studio 组合时可使用。

轴组周期同步位置控制功能的详情请参阅“NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)”中的 MC\_Group SyncMoveAbsolute( 轴组周期同步绝对位置控制 ) 指令。

### 9-6-5 多轴协调控制中的停止

停止轴组动作的功能包括执行用户程序的运动控制指令后停止的功能和发生异常等后停止。

#### 通过运动控制指令停止

需停止轴组的动作时，应使用 MC\_GroupStop(轴组强制停止) 指令或 MC\_GroupImmediateStop(轴组立即停止) 指令。

##### ● MC\_GroupStop(轴组强制停止) 指令

可对轴组中执行的直线插补和圆弧插补指定减速度和跃度，使其在指定轨迹上减速停止。

如果减速度指定为“0”，则将立即停止的指令发送至伺服驱动器。

通过本指令执行减速停止时或输入变量的 Execute(启动) 为 TRUE 时，无法接受其他动作指令。

##### ● MC\_GroupImmediateStop(轴组立即停止) 指令

可立即停止轴组中的所有轴。立即停止的方法取决于各轴的轴参数“立即停止输入停止方法”。

也可对执行 MC\_GroupStop(轴组强制停止) 指令后处于减速停止状态的轴组执行。

功能的详情请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 中的 MC\_GroupStop(轴组强制停止) 指令和 MC\_GroupImmediateStop(轴组立即停止) 指令。

#### 发生异常等后停止

##### ● 轴组动作中发生异常后停止

轴组动作时，轴组中任意的构成轴发生减速停止异常后会在插补轨迹上执行插补减速停止。此时的插补减速度取决于控制中的指令的减速度的指定。

任意一个构成轴发生立即停止异常时，轴组中其他构成轴会根据轴组参数设定中“轴组停止方法”的设定而停止。

轴组的停止方法可从以下几种方法中进行选择。

- 立即停止
- 使各轴减速停止 (各轴的最大减速度)
- 立即停止，同时执行伺服 OFF

##### ● 超过控制周期后停止

运动控制处理未在 2 个周期内完成且超过控制周期时，立即停止。



#### 使用注意事项

使用 NX701 CPU 单元执行多运动动作，当任意一方的任务超过运动控制周期时，两个任务的所有轴立即停止。

多任务运动请参阅 □ “A-6-2 运动控制功能 (P.A-30)”。

### ● 启动 MC 试运行功能后停止

启动 Sysmac Studio 的 MC 试运行功能后，各轴以“最大减速度”减速停止。

### ● 切换 CPU 单元的模式后停止

切换 CPU 单元的模式后，各轴以“最大减速度”减速停止。



#### 参考

- 如果在轴组动作中启动 MC\_GroupDisable(不启用轴组)指令，则各轴以最大减速度减速停止。
- 如果在轴组动作中启动 MC\_Stop(强制停止)指令，则轴和轴组会发生异常，对轴组动作执行插补减速停止。此时的插补减速度取决于控制中的指令的减速度的指定。
- 如果在轴组动作中将 MC\_Power(可运行)指令的输入变量 Enable(启用)设为 FALSE，MC 功能模块会立即停止该轴的指令值，执行伺服 OFF。伺服 OFF 时伺服驱动器的动作取决于伺服驱动器的设定。  
轴组中其他轴根据轴组参数设定中“轴组停止方法”的设定停止。轴组发生异常。
- 从运行模式切换至程序模式后，中断动作中的运动控制指令。此时，指令的输出变量“CommandAborted(执行中断)”保持 FALSE 不变。切换至程序模式后仍保持伺服 ON/OFF 状态。
- 如果在从运行模式切换至程序模式后减速停止时返回运行模式，则运动控制指令的输出变量“CommandAborted(执行中断)”变为 TRUE。
- 通过 MC\_SaveCamTable(保存凸轮表)指令正在保存时，会继续执行保存处理。
- 如果正在通过 MC\_GenerateCamTable(凸轮表生成)指令执行凸轮表生成时，会继续执行生成处理。

### 9-6-6 多轴协调控制中的超调

在轴组中执行多轴协调控制时，可通过 MC\_GroupSetOverride( 设定轴组超调值 ) 指令对启动中的插补动作设定超调值。

超调值是插补目标速度相应的比率，可在 0 ~ 500% 的范围内设定。

如果相对于插补目标速度设定为 0%，则以速度 “0” 停止，保持动作状态。

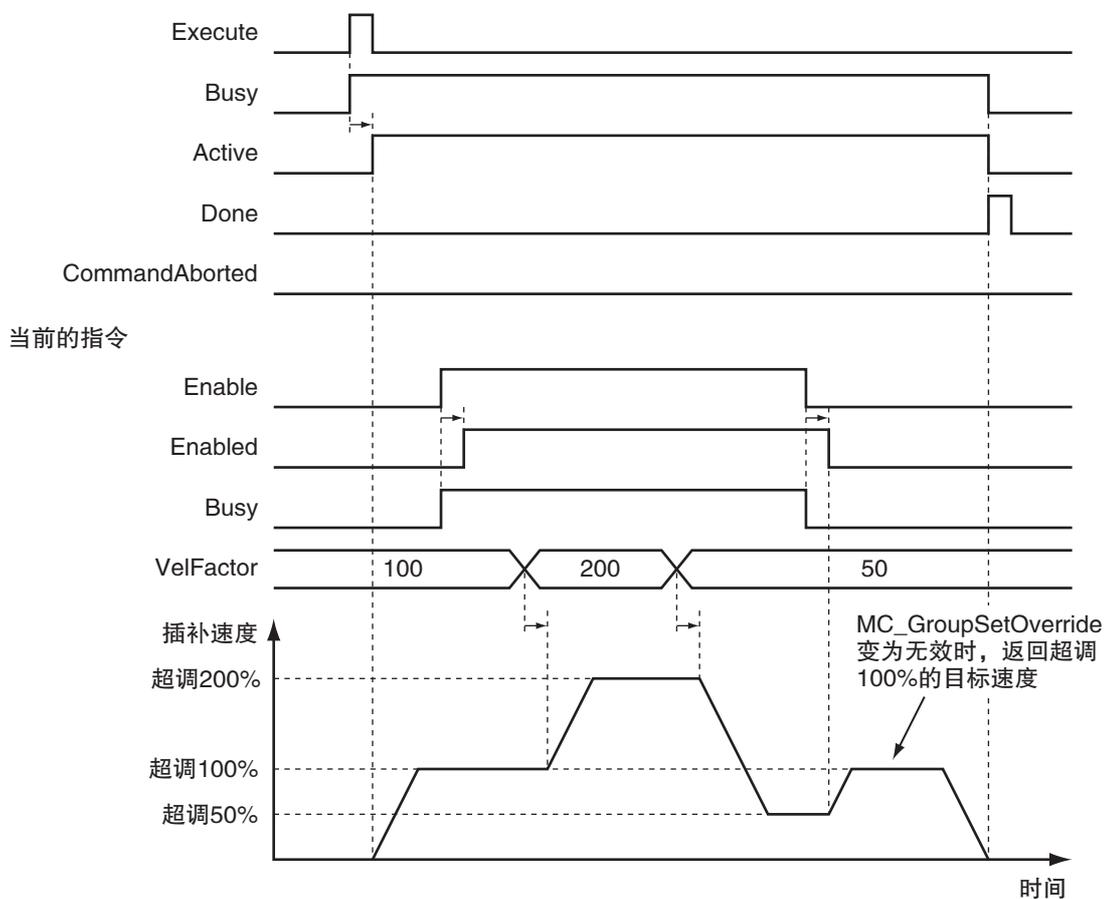
启用超调时，导入设定的超调值。如果禁用超调，则超调值恢复为 100%。

变更后的超调值超过最高插补速度时，则变为轴组参数的 “最高插补速度”。

#### ● 对 MC\_MoveLinear( 直线插补 ) 指令的超调

在 MC\_MoveLinear( 直线插补 ) 指令中使用超调指令时的时序图的示例如下所示。

之前的指令(MC\_MoveLinear)



功能的详情请参阅 “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 中的 MC\_GroupSetOverride( 设定轴组超调值 ) 指令。

## 9-7 多轴协调控制的通用功能

下面对多轴协调控制的通用功能进行说明。

### 9-7-1 多轴协调控制中的速度

多轴协调控制时以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”。

#### 速度的种类

MC 功能模块使用的轴组的插补速度仅有以下 1 种。

速度种类	定义
指令插补速度	MC 功能模块输出的用于控制轴组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。 设定为“0”时，无插补速度限制。 通过轴组动作指令指定了超过最高插补速度的插补目标速度时，以最高插补速度进行动作。	双精度实数型 正数、0	800000000
插补速度警告值	设定对最高插补速度（用于输出插补速度警告）的比例。 设定为“0”时，不输出插补速度警告。（单位：%）	0 ~ 100	0

#### 轴动作的目标速度指定

实际的定位动作的插补速度通过运动控制指令的输入变量“Velocity(目标速度)”指定。

#### 速度的监控

可通过用户程序读取轴组变量，监控插补速度。

另外，变量名称以 `_MC_GRP[*]` 为例进行说明，`_MC1_GRP[*]`、`_MC2_GRP[*]` 同理。

变量名称	数据类型	名称	功能
<code>_MC_GRP[0-63].Cmd.Vel</code>	LREAL	指令插补速度	指令插补速度的当前值。 向正方向移动带正号，向负方向移动带负号。

### 9-7-2 多轴协调控制中的加速和减速

多轴协调控制时以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<sup>2</sup>”。

#### 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。 设定为“0”时，无插补加速度限制。(单位：指令单位 /s <sup>2</sup> )	双精度实数型 正数、0	0
最大插补减速度	设定轨迹的最大插补减速度。 设定为“0”时，无插补减速度限制。(单位：指令单位 /s <sup>2</sup> )	双精度实数型 正数、0	0
插补加减速超限	在轴组的加减速控制中，优先向目标位置停止后，产生减速超限。指定超过最大插补加减速度的动作。 0: 提高加减速速度(将合并切换为等待)*1 1: 提高加减速速度 2: 异常停止*2	0 ~ 2	0
插补加速度警告值	设定对最高插补加速度(用于输出插补加速度警告)的比例。 设定为“0”时，不输出插补加速度警告。(单位：%)	0 ~ 100	0
插补减速度警告值	设定对最高插补减速度(用于输出插补减速度警告)的比例。 设定为“0”时，不输出插补减速度警告。(单位：%)	0 ~ 100	0

\*1. Ver.1.10 以上的 CPU 单元，无法将合并切换为等待。详情请参阅 [□](#) “9-5-7 运动控制指令的多重启动(缓存模式)(P.9-44)”。

\*2. Ver.1.10 以上的 CPU 单元，合并动作时不会异常停止。详情请参阅 [□](#) “9-5-7 运动控制指令的多重启动(缓存模式)(P.9-44)”。

#### 轴组动作的插补加速度和插补减速度的指定

通过运动控制指令的输入变量“Acceleration(加速度)”、“Deceleration(减速度)”指定实际定位动作的插补加速度和插补减速度。

#### 插补加速度和插补减速度的监控

可通过用户程序读取轴组变量，监控插补加速度和插补减速度。

另外，变量名称以 `_MC_GRP[*]` 为例进行说明，`_MC1_GRP[*]`、`_MC2_GRP[*]` 同理。

变量名称	数据类型	名称	功能
<code>_MC_GRP[0-63].Cmd.AccDec</code>	LREAL	指令插补加减速速度	指令插补加速度的当前值。 加速时附带正号，减速时附带负号。

### 9-7-3 多轴协调控制中的跃度

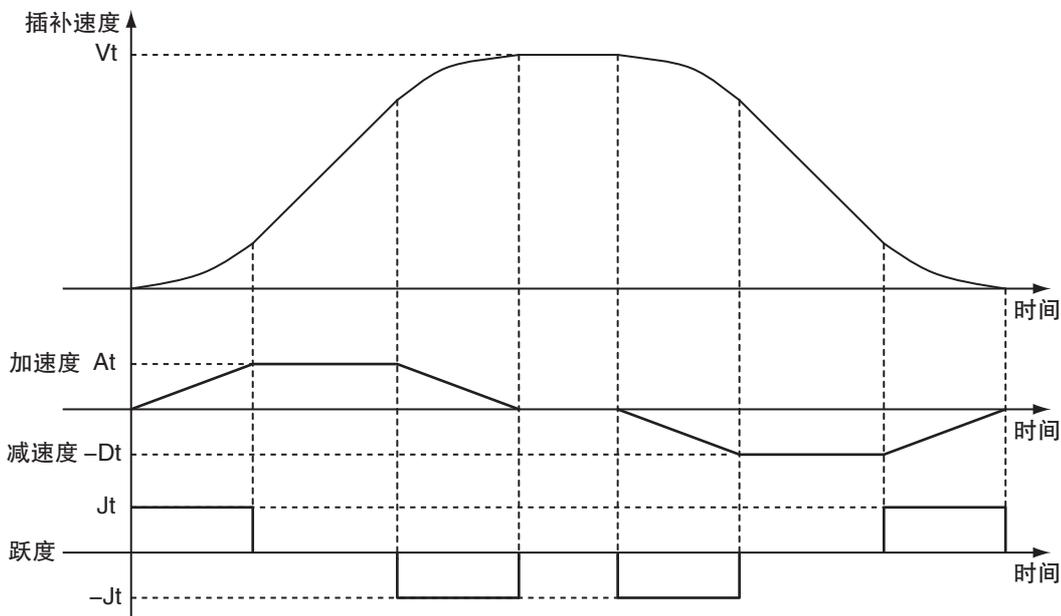
多轴协调控制中的 Jerk(跃度) 是使插补轨迹上的插补加减速呈平滑的 S 形, 以减小机械冲击的功能。单位与单轴时相同, 使用 “指令单位 /s<sup>3</sup>”。

#### 轴组动作的跃度的指定

实际的插补动作的跃度通过运动控制指令的输入变量 “Jerk(跃度)” 指定。

#### Jerk(跃度) 为非 “0” 时

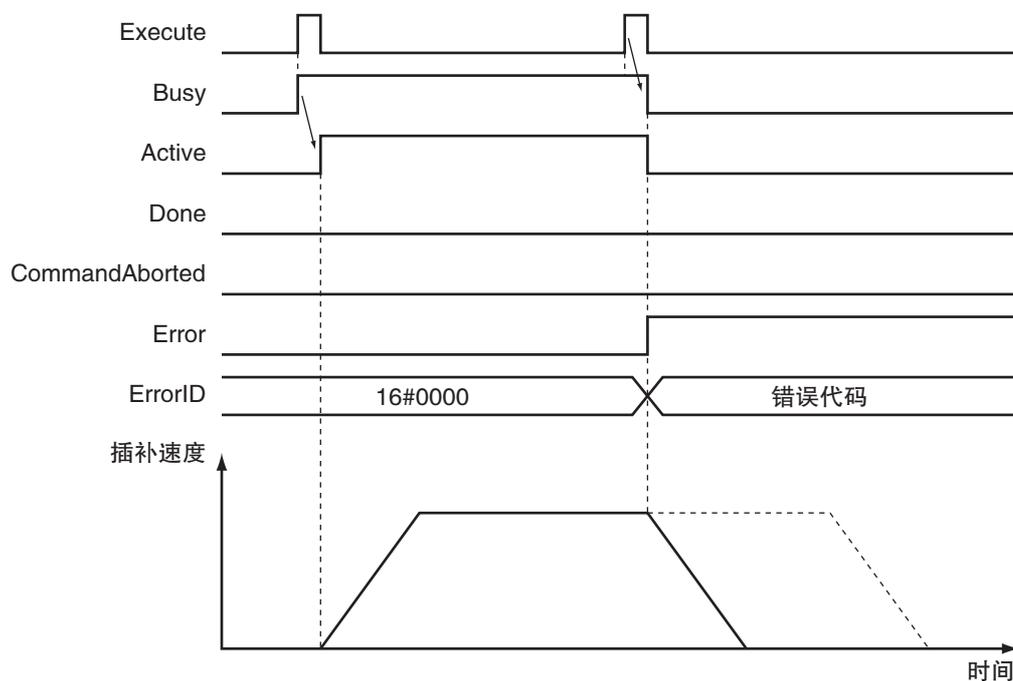
跃度适用的区间的加减速以固定比率变化, 因此指令插补速度为平滑的 S 形。跃度为 “0” 的区间的插补加速度固定, 因此指令插补速度为直线形。



$V_t$ : 插补速度的指定值、 $A_t$ : 加速度的指定值、 $D_t$ : 减速度的指定值、 $J_t$ : 跃度的指定值

## 9-7-4 多轴协调控制的运动控制指令的重启

重启直线插补或圆弧插补指令后，指令会发生异常。



重启 MC\_GroupStop(轴组强制停止) 指令后可变更减速度，但无法变更跃度。

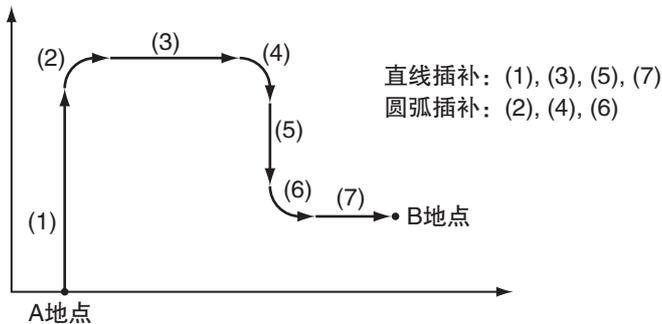
重启 MC\_GroupReset(轴组错误复位) 指令后，会忽略重启的指令并继续执行异常解除处理。

重启各运动控制指令的详情请参阅 “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 中的各指令。

### 9-7-5 多轴协调控制的运动控制指令的多重启动（缓存模式）

与轴动作相同，在轴组中也可多重启动多轴协调控制。

如果在多轴协调控制中使用多重启动，则可执行多个直线和圆弧连续的轨迹控制。



指定运动控制指令的输入变量“BufferMode(选择缓存模式)”后可选择与单轴时相同的多重启动模式。

各轴组均有 8 个指令缓存，包括轴组动作时的 1 个和用于多重启动的 7 个。

轴动作的指令和轴组动作的指令无法相互多重启动。



#### 使用注意事项

- 1 个轴组最多能缓存 7 个多重启动。如果执行 8 个以上指令的多重启动，则指令会发生异常。
- 无法对单轴控制下动作的轴多重启动多轴协调控制指令（轴组指令）。无法对多轴协调控制（轴组指令）下动作的轴多重启动单轴控制指令（轴指令）。如果在上述情况下执行多重启动，则指令会发生异常。

## 中断

不缓存的默认模式。

立即中断执行中的指令，执行多重启动的新指令。

多重启动无输入变量“BufferMode(选择缓存模式)”的运动控制指令时，中断动作。

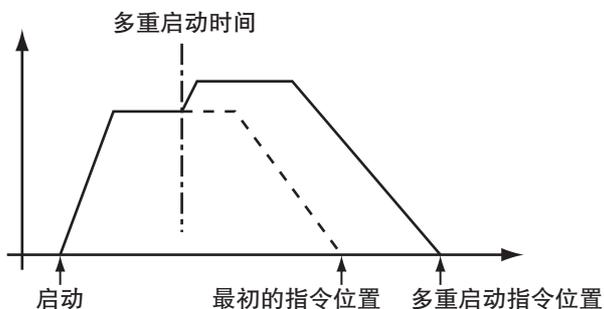
以多重启动时的插补速度为起点开始多重启动的指令的动作。

在包含同步控制的单轴控制和轴组控制的组合中无法执行中断。

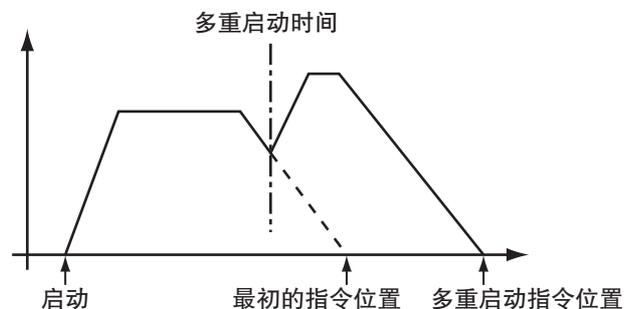
如果对单轴动作中的轴启动轴组动作，则多重启动时指令会发生异常。

如果在轴组动作中启动单轴动作，轴和轴组均发生异常并停止。

#### 等速动作中的多重启动

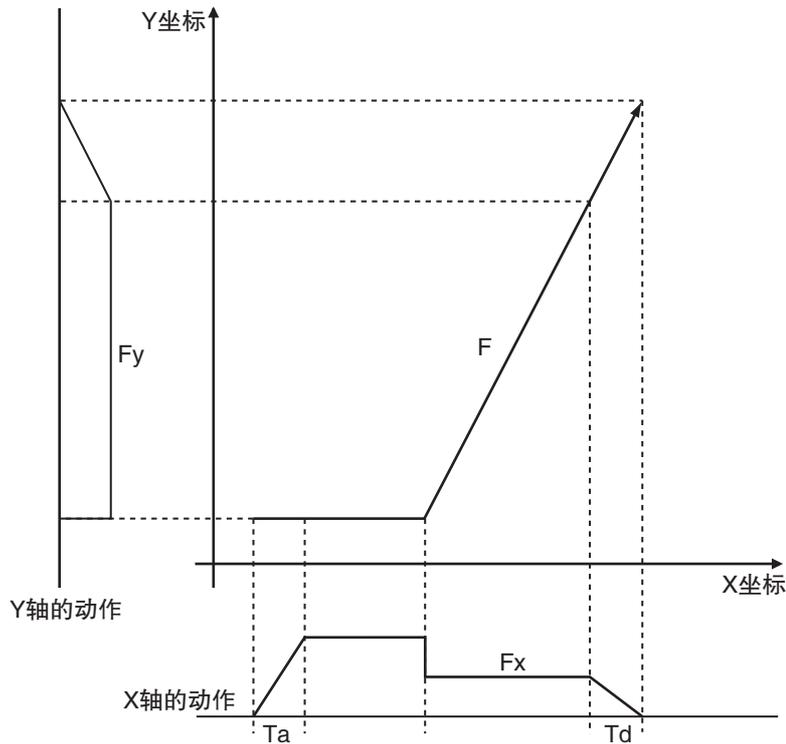


#### 加减速动作中的多重启动



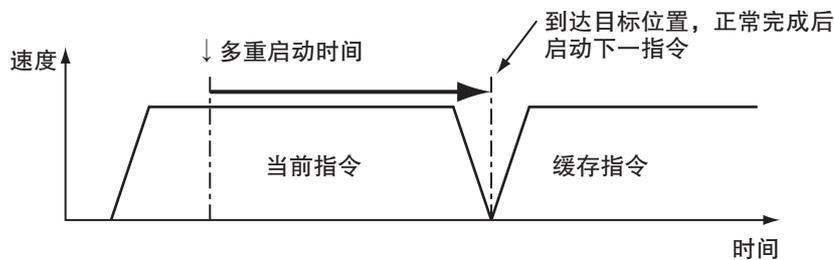
需在轴组中执行多重启动时，应使插补速度连续。与包含移动量为“0”的轴的指令连续时，各轴的速度变化会不连续。

### ● 2轴直角坐标中的插补速度和各轴的速度



## 等待

执行中的动作完成前，等待执行多重启动的指令。  
当前指令的动作正常结束后执行缓存指令。



## 合并

轴组中合并的动作与单轴动作中合并的动作相同。  
到达当前指令的目标位置前，等待执行多重启动的指令。  
到达当前指令的目标位置后执行缓存指令。此时不在到达位置停止，以输入变量“BufferMode(选择缓存模式)”指定的插补速度连续执行2个动作。  
因缓存指令的加速度或减速度而超过目标位置时的动作可通过轴组参数中的[插补加减速超限]从以下几种中进行选择。

- 提高加减速度(将合并切换为等待)
- 提高加减速度
- 异常停止的等待动作

### 版本相关信息

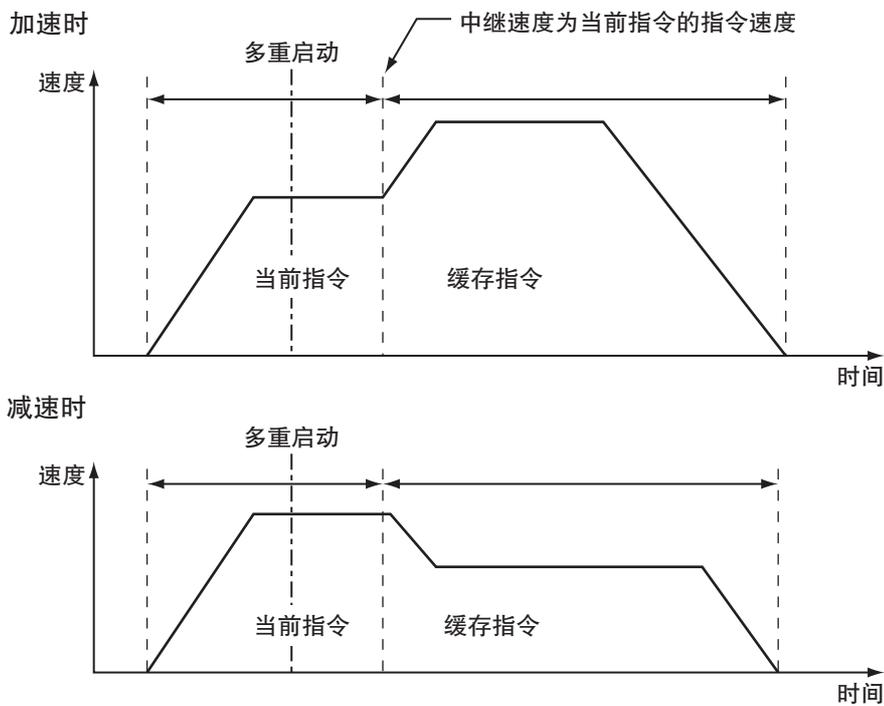
- Ver.1.10 以上的 CPU 单元，即使设定为 [ 提高加减速速度 ( 将合并切换为等待 ) ]，也无法将合并切换为等待。此时以最大加减速速度加减速，继续进行合并。  
此外，即使设定 [ 异常停止 ]，也不作为异常停止。此时也以最大加减速速度加减速，继续进行合并。
- 使用 Ver.1.09 以下版本的 CPU 单元时，请注意以下几点。  
多轴协调控制下的合并中，曲线运算的结果、当前指令的执行时间不足 4 个控制周期时，执行等待动作。此时，发出监视信息的“到达合并中继速度之前的移动量不足通知”。  
使用 Ver.1.10 以上版本的 CPU 单元时，无上述限制。

#### ● 以低速合并 ( 低速 )

在当前指令的目标位置动作，以达到当前指令和缓存指令中较低的目标速度。

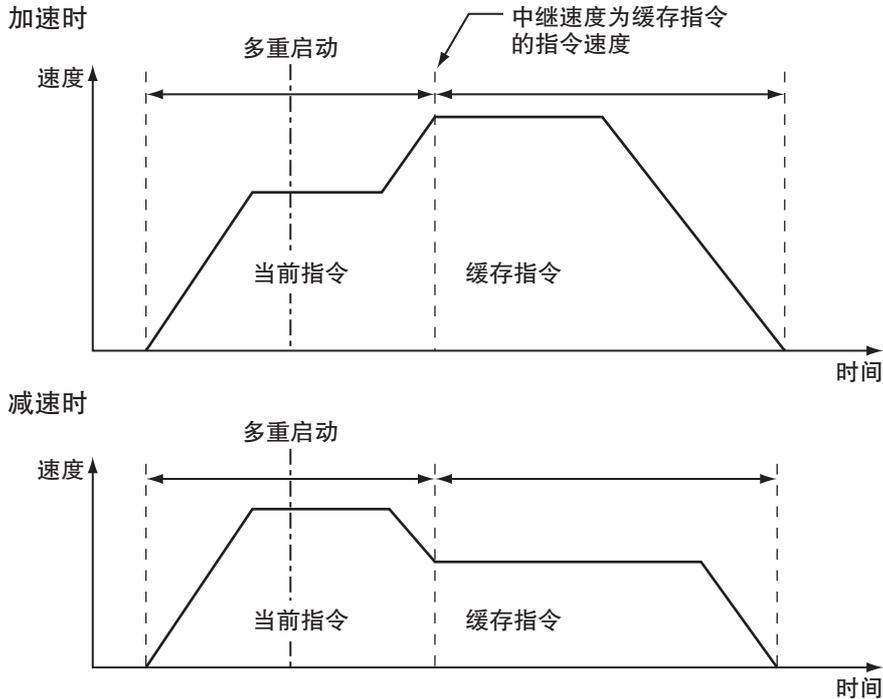
#### ● 以前一个速度合并 ( 前一个速度 )

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，动作时应执行加减速以达到缓存指令的目标速度。



### ● 以后一个速度合并 ( 后一个速度 )

在当前指令的目标位置动作，以达到缓存指令的目标速度。



### ● 以高速合并 ( 高速 )

在当前指令的目标位置动作，以达到当前指令和缓存指令中较高的目标速度。

## 过渡模式

多重启动轴组时，改变插补轨迹的方向后可能会对装置和工件产生冲击。可通过运动控制指令的输入变量“TransitionMode(过渡模式)”指定指令间的连接动作方法以减小上述冲击。

MC 功能模块中可选择的过渡模式如下所示。

No.	过渡模式	说明
0	禁用过渡 (_mcTMNone)	不执行过渡处理(默认)。 不减小冲击，动作时间变短。
10	附加角 (_mcTMCornerSuperimposed)	叠加当前指令的减速和缓存指令的加速。 可固定插补轨迹的线速度。



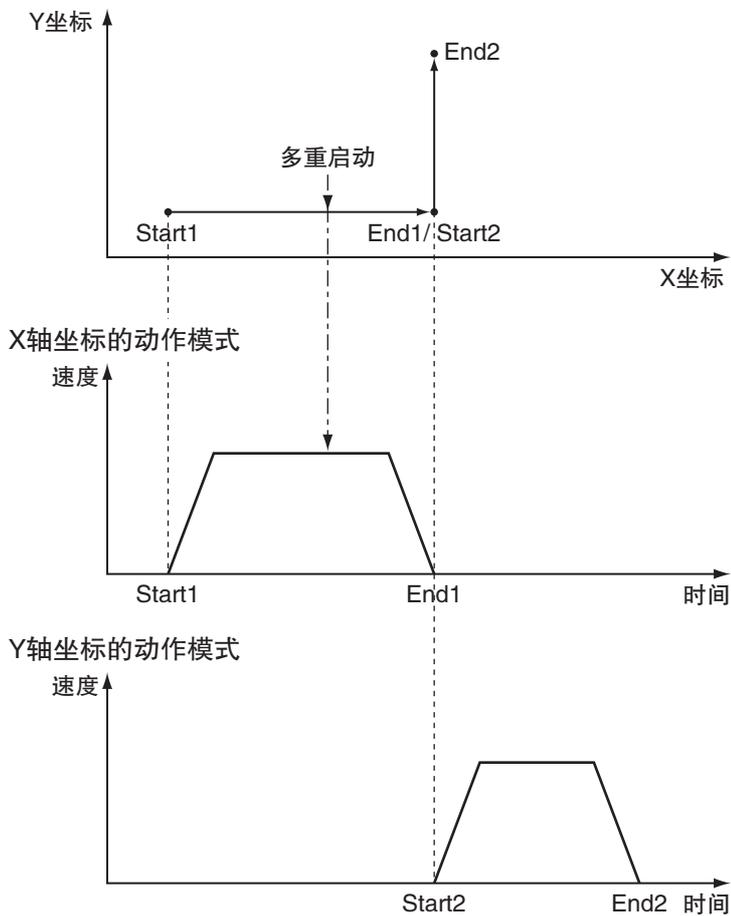
### 参考

PLCopen® 的技术规格中定义为 No.0 ~ 9。No.10 是 MC 功能模块独有的规格。

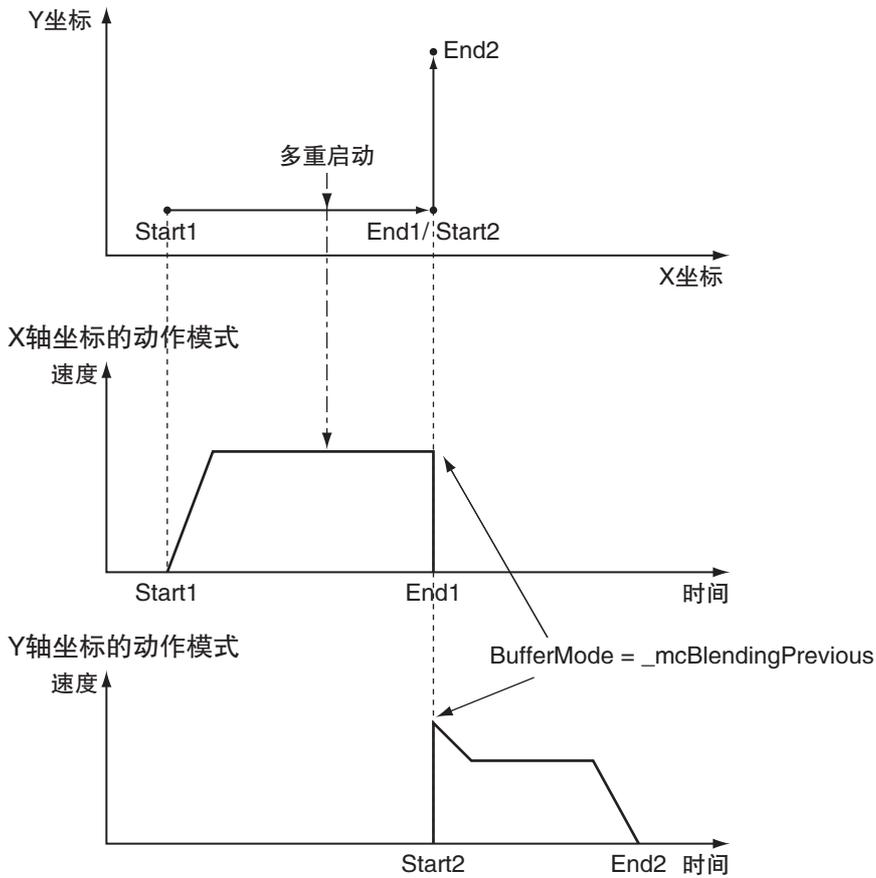
### ● 禁用过渡 (0:\_mcTMNone)

不执行连接 2 个位置的处理

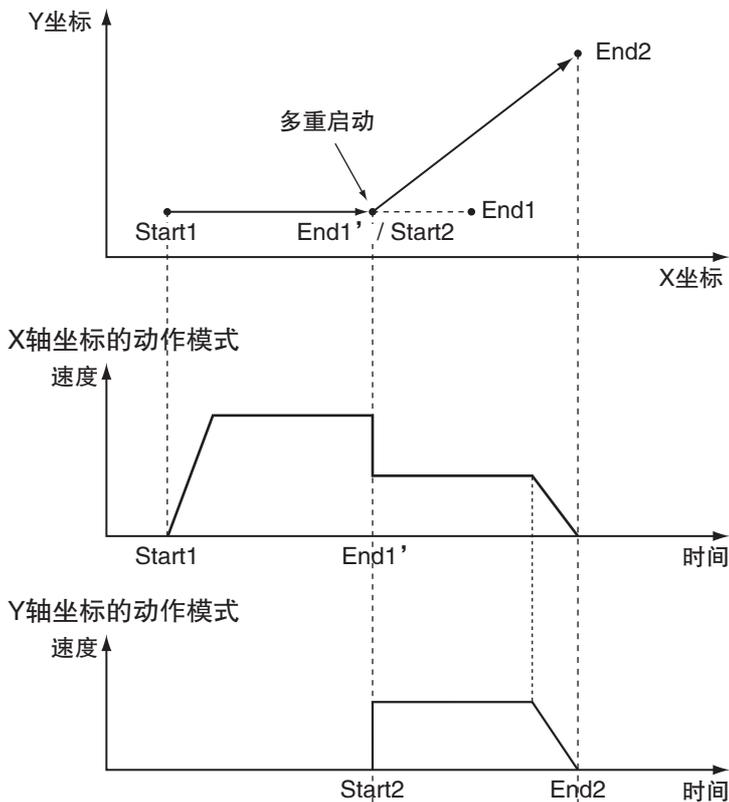
- BufferMode(缓冲模式选择) 为 [1:\_mcBuffered] 时, 如下所示, 在移动至位置 End1 并停止后, 再移动至位置 End2。



- “BufferMode(缓冲模式选择)”为合并时，如下所示，在移动至位置 End1 后，再移动至位置 End2。



- “BufferMode(缓冲模式选择)”为 [0:\_mcAborting] 时，如下所示，从多重启动位置 End1 移动至位置 End2。

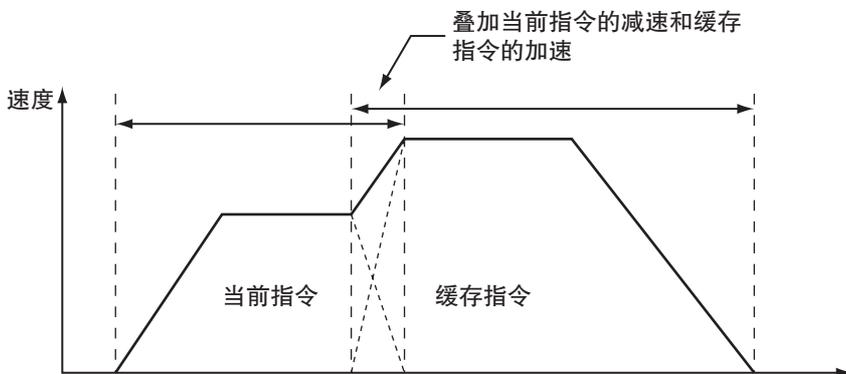


### ● 附加角 (10:\_mcTMCornerSuperimposed)

叠加当前指令的减速和缓存指令的加速。

与缓存指令的加速度的指定无关，以与当前指令减速度的减速时间相同的时间动作。

与跃度的指定无关，叠加的区间无跃度。



执行 \_mcTMCornerSuperimposed 时，表示运动控制指令完成的输出变量 “Done(完成)” 在叠加区间结束时变为 TRUE。



#### 参考

同时满足以下 2 个条件时，轨迹的线速度固定不变。

- 将当前指令和缓存指令的目标速度设为相同的值。
- 将当前指令的减速度和缓存指令的加速度设为相同的值。

## 过渡模式和多重启动的组合

TransitionMode(过渡模式) 和 BufferMode(选择缓存模式) 下多重启动指令的组合。

○：可动作 -：发生异常并停止

过渡模式	缓存模式	中断	等待	以低速合并	以前一个速度合并	以后一个速度合并	以高速合并
禁用过渡 (_mcTMNone)		○	○	○	○	○	○
附加角*1 (_mcTMCornerSuperimposed)		-	-	○	○	○	○

\*1. 对于附加角，与合并的种类无关，叠加当前指令的减速和缓存指令的加速。

## 9-8 其他功能

下面对 MC 功能模块中的其他功能进行说明。

关于使用 NX 系列 脉冲输出单元时的差异, 请参阅 □ “NX 系列 位置接口单元 用户手册 (SBCE-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与指令当前位置保持位置偏差的值的功  
能。

对于编码器轴, 变更反馈当前位置。

使用 MC\_SetPosition( 变更当前位置 ) 指令, 指定需变更的当前位置。

可在轴动作中变更当前位置。

执行绝对值定位时, 向变更后的绝对坐标中的目标位置定位。

执行相对值定位时, 不变更需移动的距离。

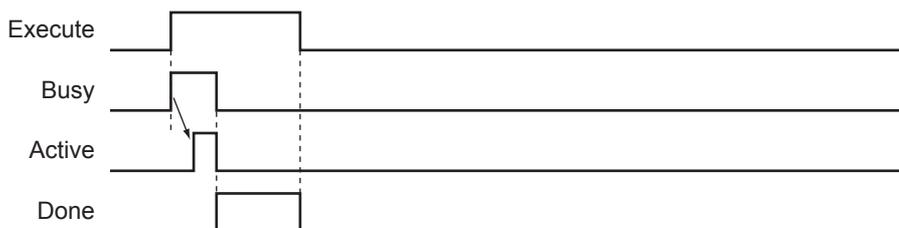


#### 使用注意事项

- 计数模式为旋转模式时, 如果指定的位置超过环形计数器的范围, 则指令会发生异常。
- 当前位置变更结束后处于未确定原点的状态, 因此无法使用以下功能和指令。

软件限制功能  
高速原点复位  
插补指令 ( 直线插补和圆弧插补 )

#### ● 轴停止状态下执行时的时序图



#### 参考

如果通过 MC\_Home( 原点复位 ) 指令或 MC\_HomeWithParameter( 参数指定原点复位 ) 指令指定 “原点预设”, 可在确定原点的状态下变更当前位置。

MC\_SetPosition( 变更当前位置 ) 指令的详情 □ 请参阅 “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)”。

## 9-8-2 转矩限制功能

通过伺服驱动器的转矩限制功能的启用 / 禁用切换和转矩限制值的设定, 限制输出转矩的功能。  
可单独指定正方向的转矩限制和负方向的转矩限制。

功能的详情请参阅 □□ “NJ/NX系列 指令基准手册 运动篇(SBCE-364)” 中的MC\_SetTorqueLimit(限制转矩)指令。



### 使用注意事项

使用 NX 系列脉冲输出单元时, 无法使用该功能。

## 9-8-3 锁定功能

为了实现以传感器输入等发生触发信号为基点进行位置控制, 发生触发信号时记录 (锁定) 轴位置。  
可同时为每个轴设定 2 点触发信号。

可使用 MC\_TouchProbe(启用外部锁定) 指令, 对需锁定的轴指定 “触发输入条件”、“启用窗口”、“选择停止方法”。

触发除可指定连接伺服驱动器的信号以外, 还可指定用户程序可使用的变量。

需中断锁定功能时, 应使用 MC\_AbortTrigger(禁用外部锁定) 指令。

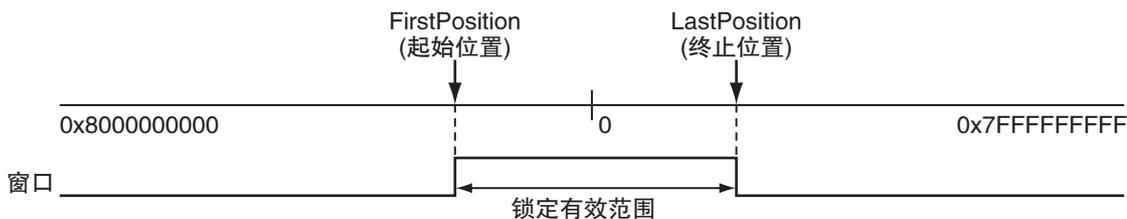
锁定功能可用于 1S 系列等支持锁定功能 (Touch probe) 的伺服驱动器和编码器输入终端 GX-EC0211/EC0241。

使用 WindowOnly(启用窗口) 后, 仅在起点和终点的范围内检测触发信号。

不同计数模式的范围如下所示。

### ● 线性模式

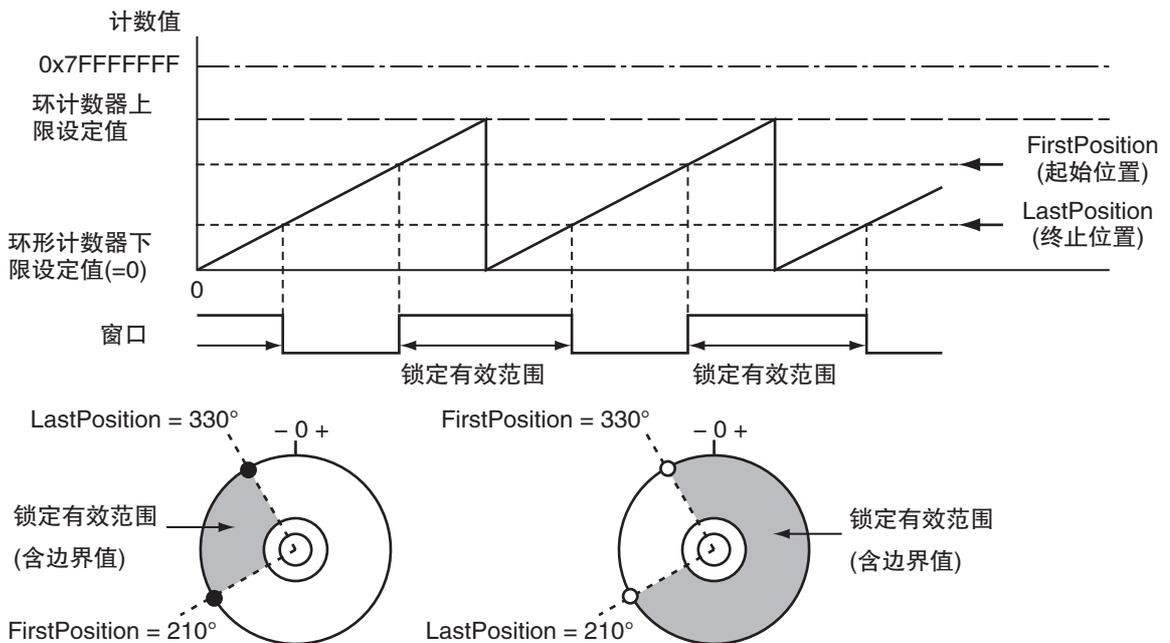
- 仅 FirstPosition(起始位置)  $\leq$  LastPosition(终止位置) 时可检测。
- 指定 FirstPosition(起始位置)  $>$  LastPosition(终止位置) 时, 指令会发生异常。
- 超过线性模式的位置范围进行指定时, 指令会发生异常。



● 旋转模式

- FirstPosition( 起始位置 ) ≤ LastPosition( 终止位置 )  
和 FirstPosition( 起始位置 ) > LastPosition( 终止位置 ) 两者均可指定。指定后者时，设定为跨越环形计数器下限设定值。
- 超过环形计数器上下限范围进行指定时，指令会发生异常。

	First Position ≤ Last Position	First Position > Last Position
有效范围	FirstPosition ~ LastPosition	LastPosition ~ FirstPosition



锁定功能的详情请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 中的 MC\_TouchProbe( 启用外部锁定 ) 指令和 MC\_AbortTrigger( 禁用外部锁定 ) 指令。

关于使用 NX 系列 脉冲输出单元时的差异，请参阅 □ “NX 系列 位置接口单元 用户手册 (SBCE-374)”。

9-8-4 区域功能

判定指定范围 ( 区域 ) 内是否存在轴的指令位置或反馈位置的功能。

使用 MC\_ZoneSwitch( 监控区域 ) 指令指定需判定的范围的起始位置和结束位置。

进入轴位置指定的区域内时，监控区域指令的输出变量 “InZone( 区域内 )” 变为 TRUE。

1 个轴可指定多个区域，可重复区域。

区域功能的详情请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 中的 MC\_ZoneSwitch( 监控区域 ) 指令。

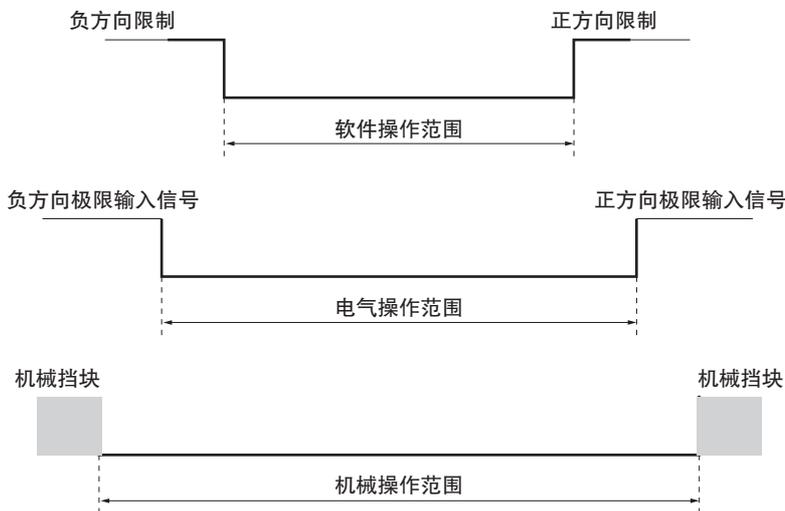
### 9-8-5 软件限制功能

除硬件的极限输入信号输入以外，还通过 MC 功能模块的软件监控当前位置的功能。

软件限制的监控范围需设定轴参数的 [ 正方向软件限制 ] 和 [ 负方向软件限制 ]。

一般定位功能可在该软件限制的范围内动作。

设定软件限制后可防止用户程序的错误和因误操作而导致的设备损坏。



#### ● 软件限制功能相关的轴参数

参数名称	功能	设定范围	初始值
软件限制功能	选择软件限制的功能。 0: 禁用 1: 对指令位置启用, 减速停止 *1 2: 对指令位置启用, 立即停止 3: 对反馈位置启用, 减速停止 *1 4: 对反馈位置启用, 立即停止	0 ~ 4	0
正方向软件限制	设定正方向的软件限制值。 单位为指令单位。	双精度实数型 负数、正数、0 *2	2147483647
负方向软件限制	设定负方向的软件限制值。 单位为指令单位。		-2147483648

\*1. 执行输入变量中带有“Deceleration(减速度)”的指令时，如果当前位置超过软件限制范围，则以“Deceleration(减速度)”减速停止。执行输入变量中无“Deceleration(减速度)”的指令时，如果当前位置超过软件限制范围，则以轴参数的最大减速度减速停止。

\*2. 可在带符号整数型 40 位范围内设定转换为脉冲单位的数值。

可通过 Sysmac Studio、MC\_Write(写入 MC 设定) 指令或 MC\_WriteAxisParameter(写入轴参数) 指令设定上述轴参数。

如果在轴或轴组动作中变更设定值，则变更后的设定会在下一个动作开始时生效。

根据轴的动作状态和运动控制指令，软件限制在以下 2 种情况下启动。

### ● 启动运动指令时

- 当前位置在软件限制范围内时  
目标位置超过软件限制范围时指令会发生异常。
- 当前位置超过软件限制范围时  
可向软件限制范围内移动。此时的目标位置无需在软件限制范围内。



#### 使用注意事项

启动轴指令的命令时请避免目标位置超过软件限制的范围。

### ● 轴动作时

定位动作中、同步动作中、连续动作中、多轴协调动作中的轴动作时

- [对指令位置启用] 软件限制功能时  
指令位置从在范围内变为超过范围时轴会发生异常。
- [对反馈位置启用] 软件限制功能时  
反馈位置从在范围内变为超过范围时轴会发生异常。



#### 参考

计数模式为线性模式，处于确定原点的状态下启用软件限制功能。

以下情况下，无论轴参数如何设定，均禁用软件限制功能。

- 计数模式为旋转模式时
- 未确定原点的状态下
- 原点复位动作中

写入 MC 设定的指令的详情请参阅 “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 中的 MC\_Write (写入 MC 设定) 指令及 MC\_WriteAxisParameter(写入轴参数) 指令。

## 9-8-6 位置偏差监控功能

轴的指令位置与反馈位置之差为位置偏差。MC 功能模块按运动控制周期监控位置偏差。

位置偏差值超过轴参数的“位置偏差超限值”时，会发生轻度故障等级的“位置偏差超限。”超过“位置偏差警告值”时，监控信息中会显示“位置偏差警告”。通过原点复位执行接触动作时，禁用位置偏差监控。

### ● 位置偏差监控功能相关的轴参数

可通过设定轴参数设定位置偏差监控的检查值。设定时请将“位置偏差警告值”控制在“位置偏差超限值”以下。

通过 Sysmac Studio 设定上述轴参数。

参数名称	功能	设定范围	初始值
位置偏差超过值	设定超过位置偏差的检查值。 设定为“0”时，超过位置偏差的检查无效。 (单位：指令单位)	双精度实数型 正数、0	0
位置偏差警告值	设定位置偏差警告的检查值。 设定为“0”时，位置偏差警告检查无效。(单位：指令单位)	双精度实数型 正数、0 (位置偏差超过值以下)	0

关于使用 NX 系列 脉冲输出单元时的差异，请参阅 “NX 系列 位置接口单元 用户手册 (SBCE-374)”。

### 9-8-7 偏差计数器复位

偏差计数器复位是使位置偏差为“0”的功能。

需通过用户程序复位偏差计数器时，应使用“MC\_ResetFollowingError(偏差计数器复位)”指令。

偏差计数器复位指令可用于单轴定位和原点复位。

如果在轴动作中执行偏差计数器复位，动作中的运动控制指令会中断动作，将指令位置变更为与反馈位置相同的值。

执行偏差计数器复位后仍不会改变原点的确定状态。

偏差计数器复位指令的详情请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 中的 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、极限输入停止时或原点复位完成时复位偏差计数器。

通过 Sysmac Studio 设定上述轴参数。

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的 MC 功能模块侧的停止方法。 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的 MC 功能模块侧的停止方法。 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

关于使用 NX 系列 脉冲输出单元时的差异, 请参阅 □ “NX 系列 位置接口单元 用户手册 (SBCE-374)”。

### 9-8-8 轴间偏差监控功能

监控 2 轴之间的指令位置或反馈位置的偏差量的功能。

使用 MC\_AxesObserve(轴间偏差监控) 指令指定监控对象的 2 轴和轴间偏差容许值。

超过容许值时, 轴间偏差监控指令的输出变量 “Invalid(偏差超限)” 变为 TRUE。

在龙门控制等 2 轴动作相同的装置中使用轴间监控功能后可对轴间偏差过大时的处理进行自由编程。



#### 使用注意事项

超过轴间偏差容许值后 MC 功能模块仍不会发生异常。

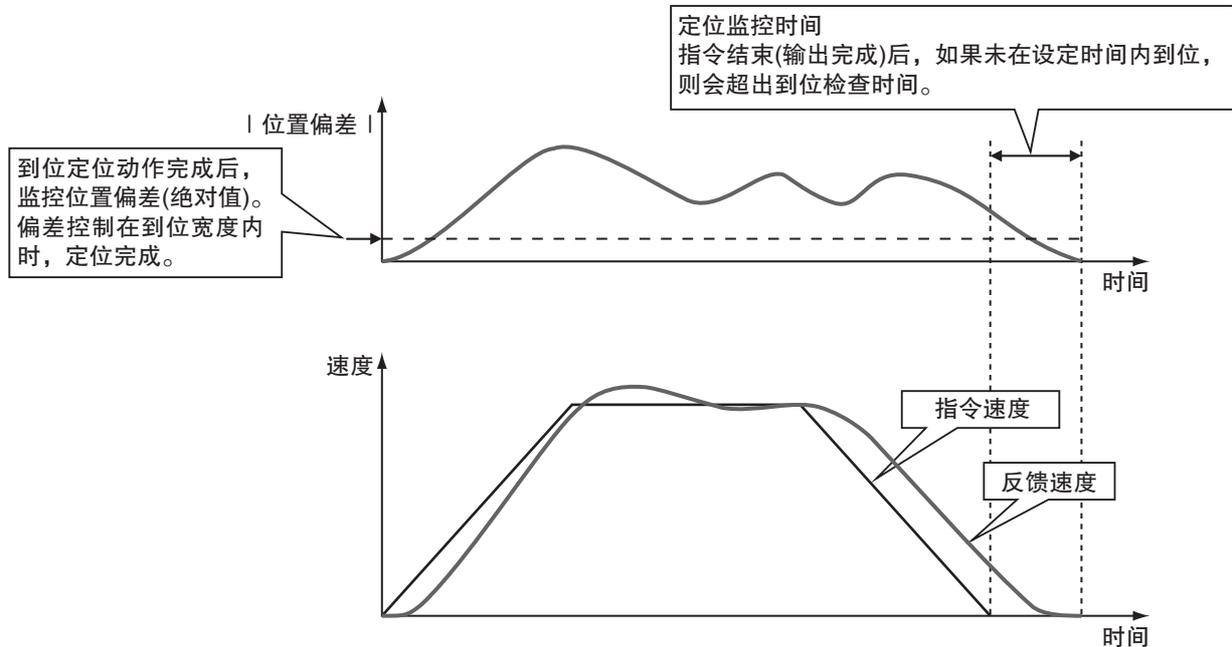
请通过输出变量 “Invalid(偏差超限)” 执行编程处理, 如通过用户程序停止轴的动作等。

轴间偏差监控功能的详情请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 中的 MC\_AxesObserve(轴间偏差监控) 指令。

关于使用 NX 系列 脉冲输出单元时的差异, 请参阅 □ “NX 系列 位置接口单元 用户手册 (SBCE-374)”。

### 9-8-9 到位检查

检查定位动作或原点复位动作时实际的反馈当前位置相对于目标位置是否到达指定范围内的功能。目标位置的指令输出完成后，如果目标位置与反馈当前位置之差小于 [ 到位宽度 ]，则定位完成。如果未在 [ 到位检查时间 ] 内完成定位，则指令会发生异常。



#### ● 到位检查相关的轴参数

可通过轴参数设定定位完成的检查条件。

切实完成定位后需开始下一个动作时，设定到位检查时间。

参数名称	功能	设定范围	初始值
到位宽度	设定定位完成宽度。 (单位：指令单位)	双精度实数型 正数、0	10
到位检查时间	以毫秒为单位，设定定位完成的检查时间。 设定为“0”时，仅原点复位时的原点确定时，持续定位完成的检查。非原点复位时，不作定位完成检查。 (单位：ms)	0 ~ 10000	0

可通过 Sysmac Studio 的轴设定、MC\_Write(写入 MC 设定) 指令或 MC\_WriteAxisParameter(写入轴参数) 指令设定上述轴参数。



#### 参考

- 到位检查在 MC 功能模块侧处理。未使用伺服驱动器侧的功能。
- 需尽早动作时，不设定到位检查时间，以便无需等待定位完成即可转入下一个动作。
- 重新接通 CPU 单元电源或通过 Sysmac Studio 的“同步”功能下载程序时，恢复为通过 Sysmac Studio 设定的值。  
MC\_Write(写入 MC 设定) 指令及 MC\_WriteAxisParameter(写入轴参数) 指令用于暂时变更到位检查时间的场合。

### ● 到位检查的相关监控

可通过用户程序读取轴变量，监控定位的完成情况。

另外，变量名称以 `_MC_AX[*]` 为例进行说明，`_MC1_AX[*]`、`_MC2_AX[*]` 同理。

变量名称	数据类型	名称	功能
<code>_MC_AX[0-255].Details.Idle</code>	BOOL	停止中	除到位等待以外，未进行指令值运算时也变为 TRUE。 <sup>*1</sup> Idle 与 InPosWaiting 相互排斥，不会同时变为 TRUE。
<code>_MC_AX[0-255].Details.InPosWaiting</code>	BOOL	到位等待	到位等待时变为 TRUE。 通过需执行到位检查的定位来执行到位检查的状态。

\*1. 以速度“0”动作、偏差计数器复位、同步控制中以及多轴协调动作中也属于正在运算的状态。

可通过读取轴组变量监控轴组定位的完成情况。

另外，变量名称以 `_MC_GRP[*]` 为例进行说明，`_MC1_GRP[*]`、`_MC2_GRP[*]` 同理。

变量名称	数据类型	名称	功能
<code>_MC_GRP[0-63].Details.Idle</code>	BOOL	停止中	除到位等待以外，未进行指令值运算时也变为 TRUE。 <sup>*1</sup> Idle 与 InPosWaiting 相互排斥，不会同时变为 TRUE。
<code>_MC_GRP[0-63].Details.InPosWaiting</code>	BOOL	到位等待	任意一个构成轴到位等待时变为 TRUE。 <sup>*2</sup> 通过需执行到位检查的定位来执行到位检查的状态。

\*1. 以速度“0”动作也属于正在运算的状态。

\*2. 轴组内的所有构成轴控制在各轴参数的到位宽度的范围内时，变为 FALSE。

写入 MC 设定的指令的详情请参阅 □□ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 中的 `MC_Write` (写入 MC 设定) 指令及 `MC_WriteAxisParameter` (写入轴参数) 指令。

关于使用 NX 系列 脉冲输出单元时的差异，请参阅 □□ “NX 系列 位置接口单元 用户手册 (SBCE-374)”。

### 9-8-10 轴使用变更功能

通过 MC\_ChangeAxisUse(轴使用变更)指令暂时切换轴参数中的 [轴使用] 的功能。可切换轴是指,轴参数的 [轴使用] 设定为 [未使用轴(可切换为使用轴)] 或 [使用轴] 的轴。  
[轴使用] 设定为 [未使用轴(可切换为使用轴)], [轴种类] 设定为 [伺服轴] 或 [虚拟伺服轴] 时,可分配至轴组。  
Ver1.04 以上的 CPU 单元和 Ver1.05 以上的 Sysmac Studio 组合时可使用。



#### 使用注意事项

- 请勿将设定为 [未使用轴(无法切换为使用轴)] 的轴切换为 [使用轴]。
- 轴参数中的 [轴使用] 设定为 [未使用轴(无法切换为使用轴)] 时,无法分配至轴组。

功能的详情请参阅 □□ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364E 以后)” 中的 MC\_ChangeAxisUse (轴使用变更) 指令。

MC\_ChangeAxisUse(轴使用变更)的应用示例请参阅 □□ “NJ/NX 系列 CPU 单元 用户手册 软件篇 (SBCA-359 以后)”。

### 9-8-11 数字凸轮开关有效功能

根据轴的位置,通过 MC\_DigitalCamSwitch(数字凸轮开关有效)指令将数字输出设为 ON 或 OFF。  
还可通过设定指令的输入变量 “ValueSource”, 加入加减速度的影响。  
该功能必须与支持 NX\_AryDOutTimeStamp 指令及时间戳方式的数字输出单元组合使用。NX\_AryDOutTimeStamp 指令会按指定时间戳的时间,将指定的数字输出设为 ON/OFF。  
Ver.1.06 以上的 CPU 单元和 Ver.1.07 以上的 Sysmac Studio 组合时可使用。



#### 使用注意事项

本指令仅对分配了 NX 系列位置接口单元的轴有效。  
可使用的 NX 单元为 NX-EC0 □□□、NX-EC □□□, 变化时刻获取功能需处于动作状态。

数字凸轮开关有效功能的详情请参阅 □□ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 的 MC\_DigitalCamSwitch(数字凸轮开关有效)指令。

NX\_AryDOutTimeStamp 指令的详情请参阅 □□ “NJ/NX 系列 指令基准 基本篇 (SBCA-360)”。

关于支持时间戳方式的数字输出单元,请参阅 □□ “NX 系列 数字 I/O 单元 用户手册 (SBCA-407B 以上)”。  
关于变化时刻获取功能及时间戳,请参阅 □□ “NX 系列 位置接口单元 用户手册 (SBCE-374B 以上)”。

## 9-8-12 任意坐标系的 3D 运动监视显示

通过用户程序实现 SCARA 机器人和垂直多关节机器人等笛卡尔坐标系以外的坐标系时，通过 Sysmac Studio 对机器人手部等的轨迹进行 3D 显示的功能。

创建 `_sMC_POSITION_REF` 型的用户定义变量，通过 3D 运动监视显示进行显示。

Ver.1.06 以上的 CPU 单元和 Ver.1.07 以上的 Sysmac Studio 组合时可使用该功能。

### ● `_sMC_POSITION_REF` 型

`_sMC_POSITION_REF` 型的结构要素如下所示。

结构要素	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

下面对各结构要素进行说明。

结构要素	说明
用户定义变量 .CommandPosition[0]	指令当前位置的 X 轴成分。 代入表示用户程序生成的指令当前位置的 X 轴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。 代入表示用户程序生成的指令当前位置的 Y 轴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。 代入表示用户程序生成的指令当前位置的 Z 轴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。 代入表示用户程序使用的反馈当前位置的 X 轴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。 代入表示用户程序使用的反馈当前位置的 Y 轴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。 代入表示用户程序使用的反馈当前位置的 Z 轴位置的用户定义变量。
用户定义变量 .ActualPosition[3] ~ [5]	未使用

各结构要素中代入用户定义变量。示例如下所示。

名称	数据类型	说明
3D_position	<code>_sMC_POSITION_REF</code>	用于 3D 显示的用户定义变量
MCS_Cmd_TransX	LREAL	表示用户程序生成的指令当前位置的 X 轴位置的用户定义变量
MCS_Cmd_TransY	LREAL	表示用户程序生成的指令当前位置的 Y 轴位置的用户定义变量
MCS_Cmd_TransZ	LREAL	表示用户程序生成的指令当前位置的 Z 轴位置的用户定义变量
MCS_Act_TransX	LREAL	表示用户程序使用的反馈当前位置的 X 轴位置的用户定义变量
MCS_Act_TransY	LREAL	表示用户程序使用的反馈当前位置的 Y 轴位置的用户定义变量
MCS_Act_TransZ	LREAL	表示用户程序使用的反馈当前位置的 Z 轴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` 型的用户定义变量。
- 2** 编写将表示3D显示的“指令当前位置”和“反馈当前位置”的用户定义变量代入创建的用户定义变量的各结构要素的程序。
- 3** 在3D机构模型列表的种类中选择 [ 指定坐标 ]。  
3D机构模型参数设定画面中会显示 `_sMC_POSITION_REF` 型。
- 4** 在3D机构模型参数设定的设定值栏，设定创建的用户定义变量。
- 5** 执行用户程序。
- 6** 通过数据跟踪开始跟踪，进行数据采样。
- 7** 在数据跟踪画面中确认跟踪结果。

3D运动监视显示的详情请参阅  “Sysmac Studio Version 1 操作手册 (SBCA-362)”。

# 示例程序

本章记载了原点复位、异常监控等基本使用方法和绝对值定位、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 设置和接线 .....	10-2
10-1-3 设定 .....	10-2
10-2 基本程序示例 .....	10-3
10-2-1 EtherCAT 通信的监控和伺服 ON .....	10-3
10-2-2 通过主站控制指令进行轴动作的联锁 .....	10-4
10-2-3 异常的监控和异常解除 (单轴动作、同步动作) .....	10-6
10-2-4 异常的监控和异常解除 (多轴协调动作) .....	10-8
10-2-5 指令异常的监控方法 .....	10-14
10-2-6 异常解除的监控方法 .....	10-16
10-2-7 单轴动作过程中的轴停止 .....	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 通过重启指令变更目标位置 .....	10-33
10-2-11 中断标准定位 .....	10-38
10-2-12 通过重启指令切换凸轮表 .....	10-42
10-2-13 使用凸轮曲线补偿同步位置 .....	10-51
10-2-14 凸轮动作时补偿主轴的相位 .....	10-60
10-2-15 速度控制中变更当前位置 .....	10-68
10-2-16 变更凸轮数据变量和保存凸轮表 .....	10-74
10-2-17 临时改写轴参数 .....	10-83
10-2-18 更新凸轮表的终点索引 .....	10-86

## 10-1 示例程序的概要

下面对本章记载的示例程序的通用事项进行说明。



### 使用注意事项

- 本章的示例程序仅记载了使用 MC 功能模块功能的部分。
- 创建实际装置的程序时，请创建装置动作相关联锁程序及来自其它设备的输入输出、控制步骤等相关的程序。
- 请创建用户程序，以确保实际的装置按照意图动作。
- 请先进行充分的动作确认，然后将已创建的用户程序过渡到正式运行。
- NX 系列 位置接口单元的使用示例，请参阅 “NX 系列 位置接口单元 用户手册 (SBCE-374)”。

### 10-1-1 使用设备

下面要说明的示例程序以使用下述设备为前提。

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-1SN □□□ -ECT
伺服电机	R88M-1 □□□□□□□
编码器输入终端	GX-EC0211 (Ver.1.1)

### 10-1-2 设置和接线

关于各使用设备的设置和接线，请参阅下述手册。

使用设备	参考手册
CPU 单元和电源单元	“NJ 系列 CPU 单元 用户手册 硬件篇 (SBCA-358)”
伺服驱动器 / 电机	“AC 伺服电机 / 驱动器 1S 系列 EtherCAT 通信内置型用户手册 (SBCE-377)”
编码器输入终端	“GX 系列 EtherCAT 从站 用户手册 (SBCD-350)”
EtherCAT 通信的接线	

### 10-1-3 设定

关于各种设定，请参阅下述手册。

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-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 1S 系列之间的连接 (P.A-2)”
伺服参数	“AC 伺服电机 / 驱动器 1S 系列 EtherCAT 通信内置型用户手册 (SBCE-377)”

## 10-2 基本程序示例

MC 功能模块的基本功能相关用户程序示例如下所示。



### 使用注意事项

- 参考本程序示例时，请追加合适的连锁程序，以确保符合装置的动作条件。
- 请在 Sysmac Studio 的 [ 编程 ] 画面中输入用于程序的变量。

### 10-2-1 EtherCAT 通信的监控和伺服 ON

如果对象伺服驱动器和 EtherCAT 通信的过程数据通信建立，则通过 MC\_Power(可运行) 指令使伺服驱动器变为伺服 ON 状态。

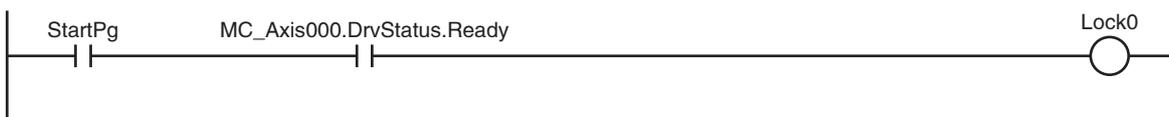
记载梯形图程序的示例和 ST 程序的示例。

#### 用于程序的主要变量

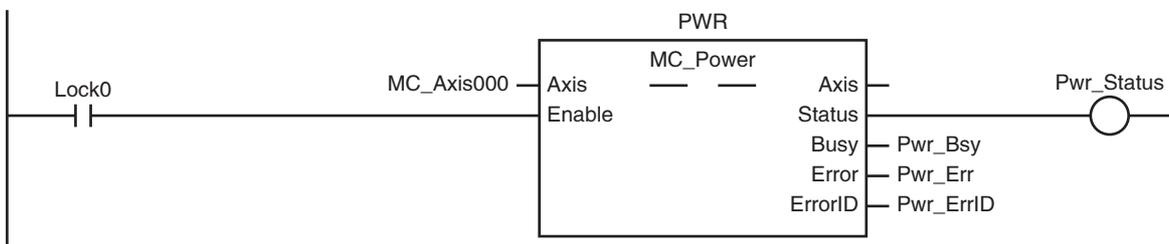
名称	数据类型	初始值	注释
MC_Axis000	_s_AXIS_REF	-	轴 0 的轴变量
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 0 发生轻度故障等级的异常时变为 TRUE
MC_Axis000.Cfg.NodeAddress	UINT	-	节点地址
_EC_PDslavTbl[N]	BOOL	FALSE	表示 TRUE 时，节点地址 N 的 EtherCAT 通信的过程数据通信为操作状态
StartPg	BOOL	FALSE	如果该变量为 TRUE，且 EtherCAT 通信的过程数据通信处于正常执行状态，则变为伺服 ON 状态

#### 梯形图程序

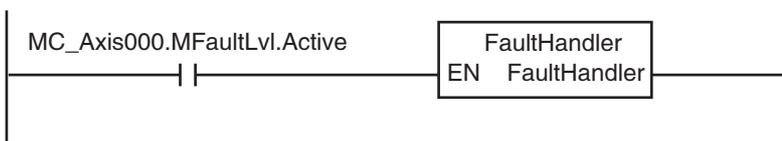
确认触点 StartPg 是否为 TRUE，伺服驱动器是否处于伺服准备就绪状态



如果伺服驱动器处于伺服准备就绪状态，则将轴 0 伺服 ON



轴 0 发生轻度故障等级的异常时，执行异常处理的 FaultHandler  
请根据装置，对异常处理的内容进行编程



## ST 程序

```

// 如果触点 StartPg 为 TRUE，且伺服驱动器处于伺服准备就绪状态，则将轴 0 伺服 ON。
IF (StartPg=TRUE)
  AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr_En:=TRUE;
  ELSE
    Pwr_En:=FALSE;
  END_IF;

// 轴 0 发生轻度故障等级的异常时，执行异常处理的 FaultHandler
// 请根据装置，对异常处理的内容进行编程
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 通过主站控制指令进行轴动作的联锁

通过梯形图程序在 MC(主站控制)指令和 MCR(主站控制复位)指令之间配置 MC\_Power(可运行)指令，以此进行轴动作的联锁控制。

如果将触点 Mc\_On 设定为 FALSE，则 MC 指令和 MCR 指令之间描述的 MC\_Power(可运行)指令会失效，变为伺服 OFF 状态。

此时，处于执行状态的动作指令的输出变量“CommandAborted(执行中断)”变为 TRUE，轴动作停止。

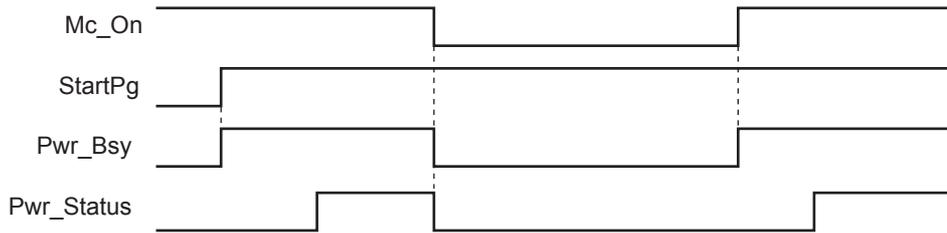
MC 指令无法用于 ST 程序。

因此，仅记载梯形图程序的示例。

### 用于程序的主要变量

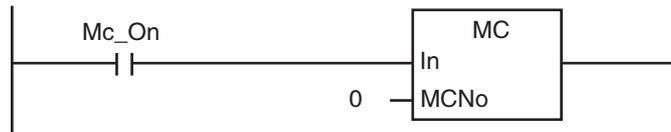
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	-	轴 0 的轴变量
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 0 发生轻度故障等级的异常时变为 TRUE
Mc_On	BOOL	FALSE	控制 MC 指令的有效及无效 该示例程序中未记载待控制的描述。请在实际的程序中根据装置的动作进行控制
StartPg	BOOL	FALSE	如果该变量为 TRUE，且 EtherCAT 通信的过程数据通信处于正常执行状态，则变为伺服 ON 状态

## 时序图

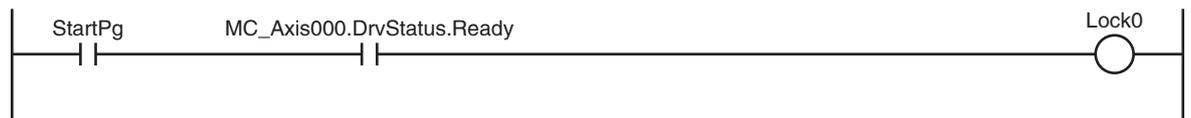


## 梯形图程序

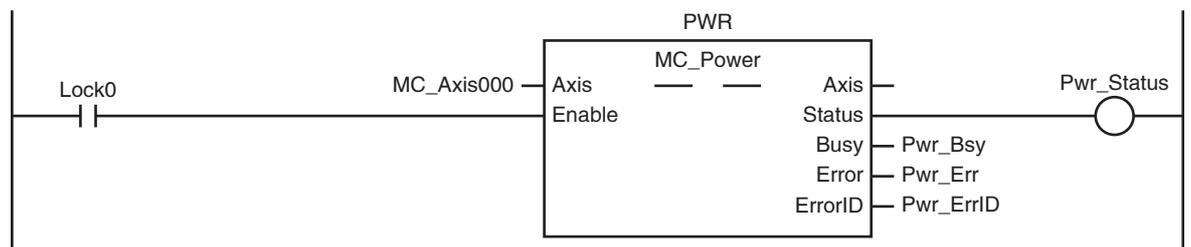
如果触点Mc\_On变为TRUE，则开始主站控制



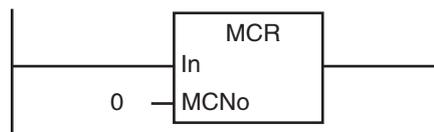
确认触点StartPg是否为TRUE，伺服驱动器是否处于伺服准备就绪状态



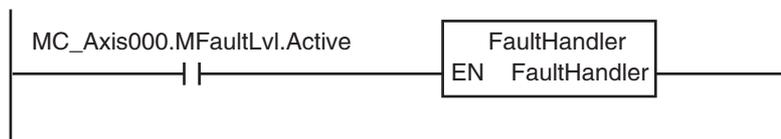
如果伺服驱动器处于伺服准备就绪状态，则将轴0伺服ON



结束主站控制



轴0发生轻度故障等级的异常时，执行异常处理的FaultHandler  
请根据装置，对异常处理的内容进行编程



### 10-2-3 异常的监控和异常解除 (单轴动作、同步动作)

通过轴变量中“轴轻度故障发生中”的状态监控有无异常。

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)”变为FALSE, 进入伺服OFF状态。

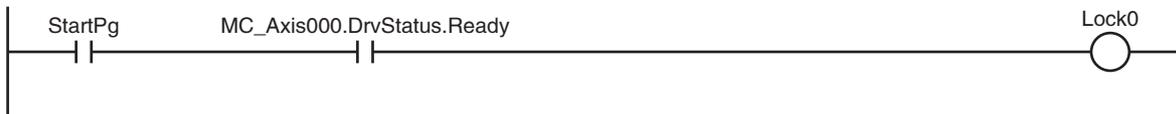
如果外部按钮变为ON状态, 且指令当前速度为零, 则通过 MC\_Reset(轴错误复位)指令解除异常。记载梯形图程序的示例和 ST 程序的示例。

#### 用于程序的主要变量

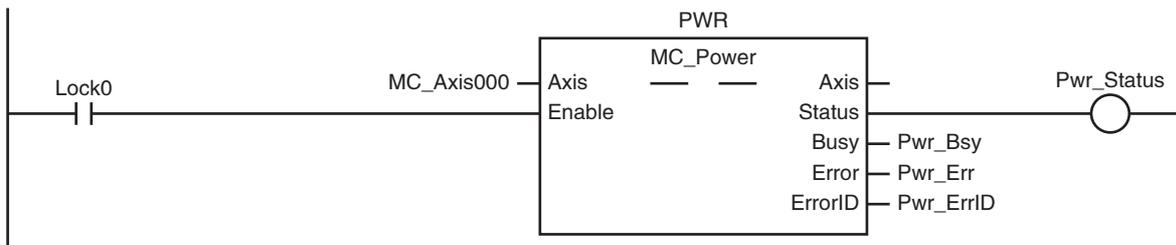
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	-	轴 0 的轴变量
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 0 发生轻度故障等级的异常时变为 TRUE
MC_Axis000.Status.ErrorStop	BOOL	FALSE	轴 0 发生轻度故障等级的异常, 轴处于减速停止中或停止中时变为 TRUE
MC_Axis000.Details.Idle	BOOL	FALSE	非到位等待时, 如果轴 0 的指令当前速度为零, 则变为 TRUE
StartPg	BOOL	FALSE	如果该变量为 TRUE, 且 EtherCAT 通信的过程数据通信处于正常执行状态, 则变为伺服 ON 状态
ResetON	BOOL	FALSE	表示用于解除异常的外部按钮的状态

#### 梯形图程序

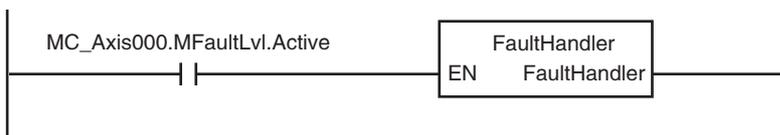
确认触点StartPg是否为TRUE, 伺服驱动器是否处于伺服准备就绪状态



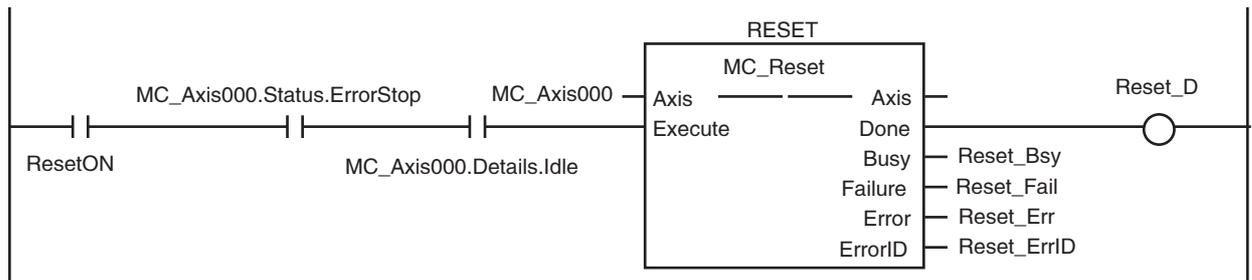
如果伺服驱动器处于伺服准备就绪状态, 则将轴0伺服ON



轴0发生轻度故障等级的异常时, 执行异常处理的FaultHandler  
请根据装置, 对异常处理的内容进行编程



外部按钮变为ON状态后，ResetON变为TRUE，如果指令当前速度为零，则解除异常



## ST 程序

```
// 确认 StartPg 为 TRUE，且伺服驱动器处于伺服准备就绪状态，则将轴 0 设为伺服 ON。
// 如果未处于伺服准备就绪状态，则将轴 0 设为伺服 OFF。
IF (StartPg=TRUE)
  AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr_En:=TRUE;
  ELSE
    Pwr_En:=FALSE;
  END_IF;

// 轴 0 发生轻度故障等级的异常时，执行异常处理的 FaultHandler
// 请根据装置，对异常处理的内容进行编程
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
  FaultHandler();
END_IF;

// 外部按钮变为 ON 状态后，ResetON 变为 TRUE，如果指令当前速度为零，则解除异常
IF (ResetOn=TRUE)
  AND (MC_Axis000.Status.ErrorStop=TRUE)
  AND (MC_Axis000.Details.Idle=TRUE) THEN
    Reset_Ex := TRUE;      // 解除轻度故障
  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 异常的监控和异常解除 (多轴协调动作)

通过轴变量中“轴轻度故障发生中”的状态及轴组变量中“轴组轻度故障发生中”的状态监控有无异常。

发生轻度故障等级的异常时，MC\_GroupDisable(不启用轴组)指令的输入变量“Execute(启动)”变为TRUE，使轴组失效。

如果外部按钮变为ON状态，且轴组的指令当前速度为零，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。

#### 用于程序的主要变量

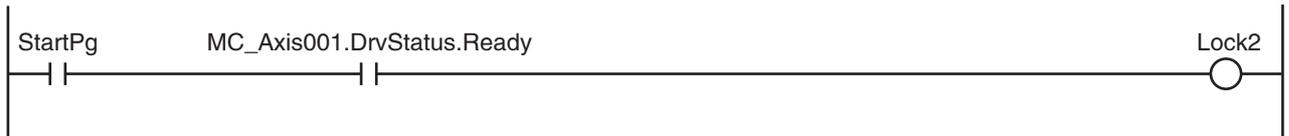
名称	数据类型	初始值	注释
MC_Group000	_sGROUP_REF	-	轴组 0 的轴组变量
MC_Group000.MFaultLvl.Active	BOOL	FALSE	轴组 0 发生轻度故障等级的异常时变为 TRUE
MC_Group000.Details.Idle	BOOL	FALSE	非到位等待时，如果轴组 0 的指令插补速度为零，则变为 TRUE
MC_Axis000	_sAXIS_REF	-	轴 0 的轴变量
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 0 发生轻度故障等级的异常时变为 TRUE
MC_Axis000.Details.Homed	BOOL	FALSE	轴 0 为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 为原点确定状态时变为 TRUE
Pwr1_Status	BOOL	FALSE	分配给 MC_Power 指令的实例 PWR1 的输出 Status 的变量 伺服进入 ON 状态时变为 TRUE
Pwr2_Status	BOOL	FALSE	分配给 MC_Power 指令的实例 PWR2 的输出 Status 的变量 伺服进入 ON 状态时变为 TRUE
StartPg	BOOL	FALSE	如果该变量为 TRUE，且 EtherCAT 通信的过程数据通信处于正常执行状态，则变为伺服 ON 状态
ResetON	BOOL	FALSE	表示用于解除异常的外部按钮的状态

## 梯形图程序

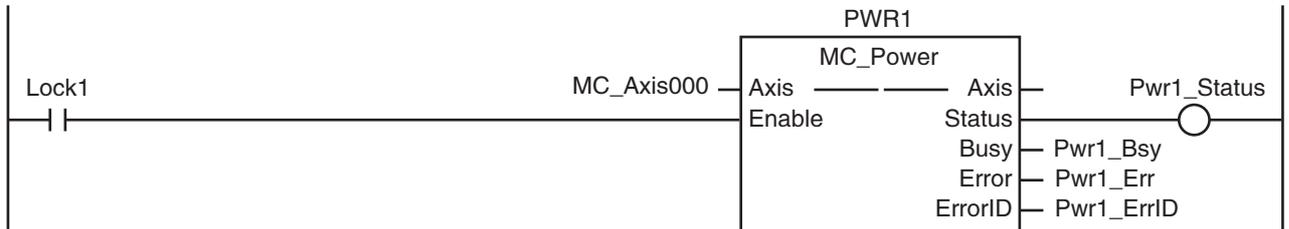
确认触点StartPg是否为TRUE，轴0的伺服驱动器是否处于伺服准备就绪状态



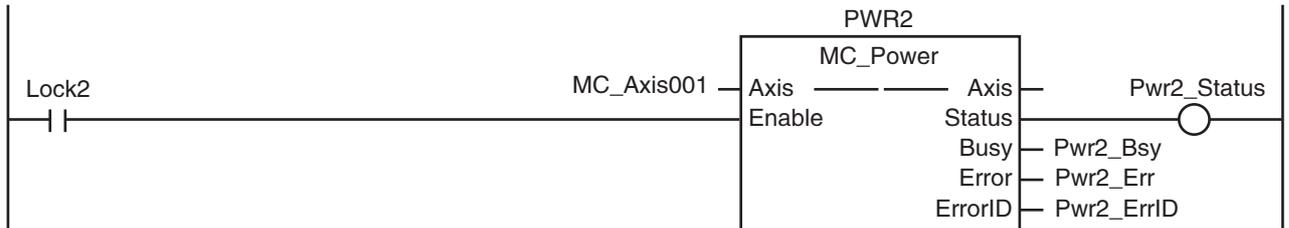
确认触点StartPg是否为TRUE，轴1的伺服驱动器是否处于伺服准备就绪状态



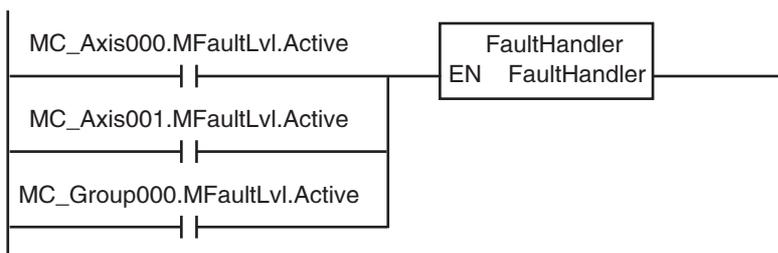
如果轴 0 的伺服驱动器处于伺服准备就绪状态，则将轴0伺服ON



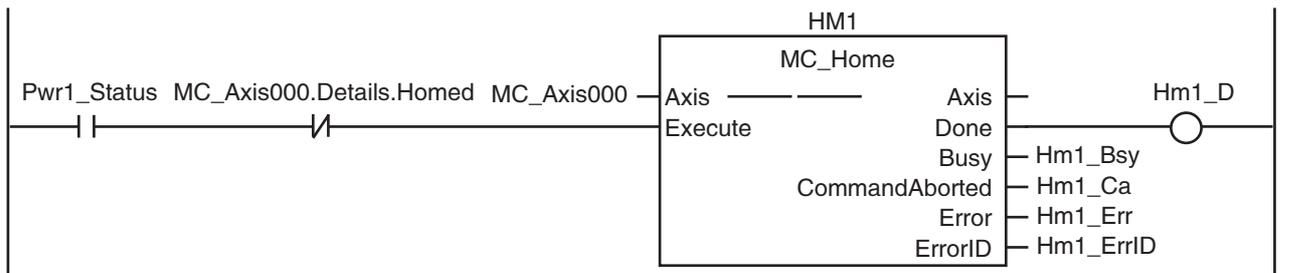
如果轴 1 的伺服驱动器处于伺服准备就绪状态，则将轴1伺服ON



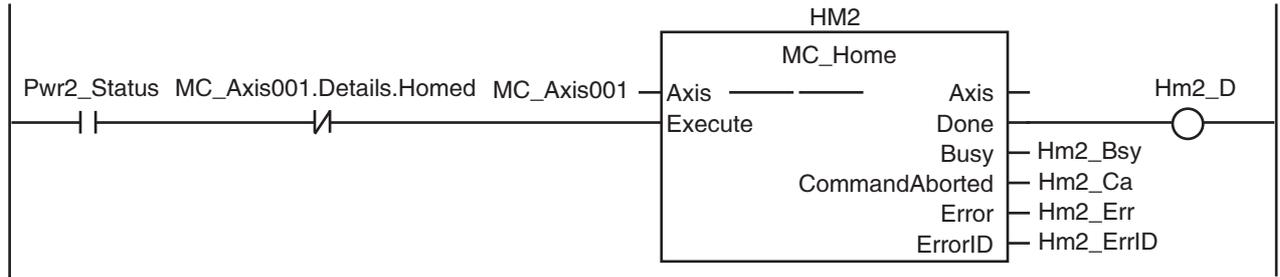
轴组或构成轴发生轻度故障等级的异常时，执行异常处理的FaultHandler  
请根据装置，对异常处理的内容进行编程



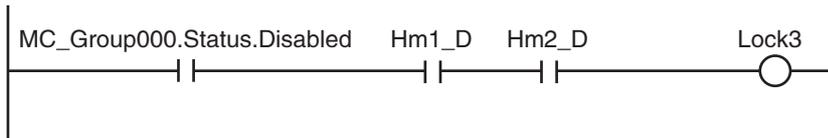
轴0处于伺服ON状态，且原点未确定时，执行原点复位



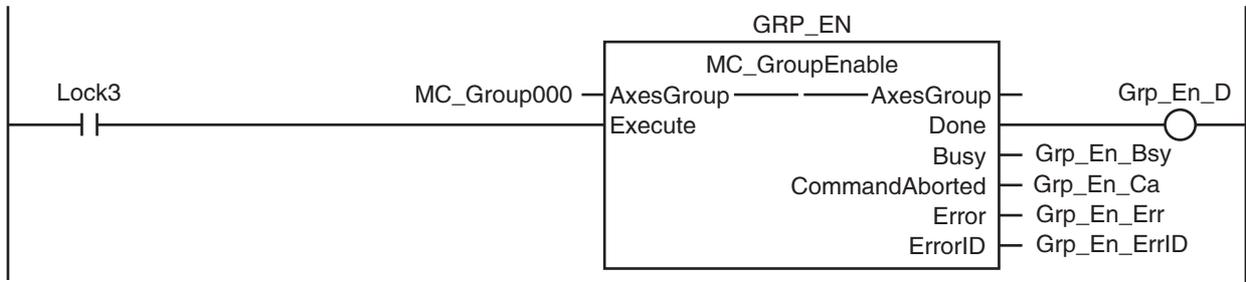
轴 1 处于伺服 ON 状态，且原点未确定时，执行原点复位



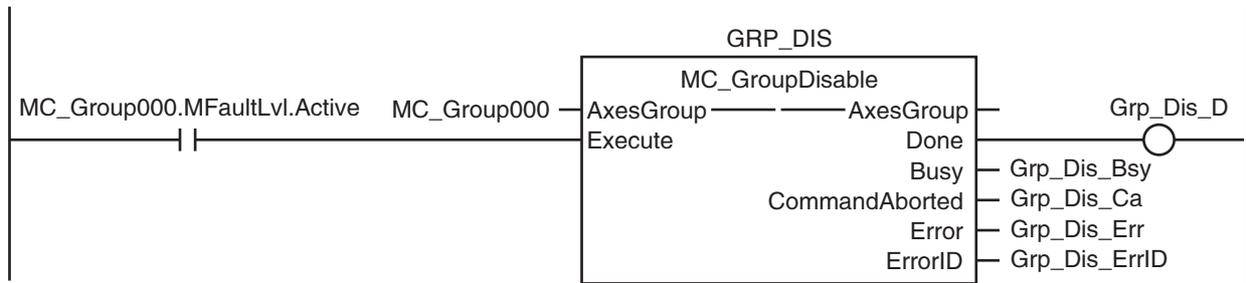
确认轴组的无效状态及轴 0、轴 1 的原点确定



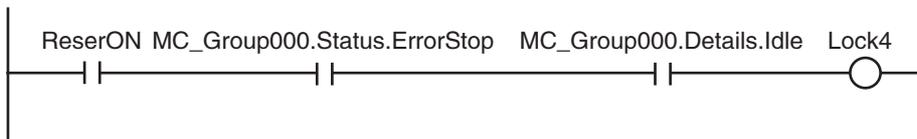
如果轴 0 和轴 1 确定了原点，则使轴组生效



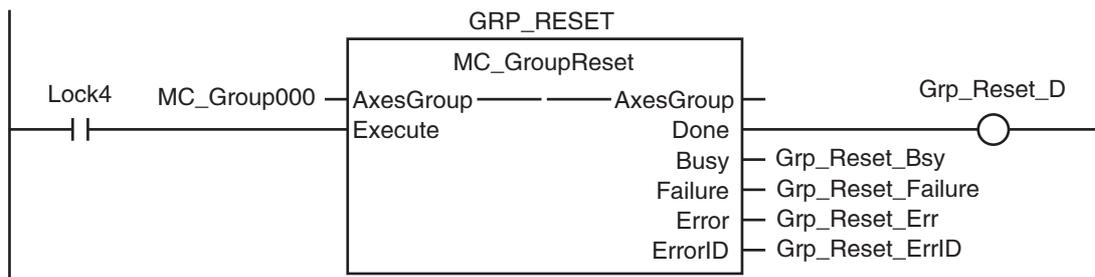
轴组发生轻度故障等级的异常时，使轴组失效



确认基于外部按钮 ON 的 ResetON 状态和轴组的停止



如果 ResetON 为 TRUE，且轴组停止，则解除异常



## ST 程序

```

// 确认 StartPg 为 TRUE, 且伺服驱动器处于伺服准备就绪状态, 则将轴 0 设为伺服 ON。
// 如果未处于伺服准备就绪状态, 则将轴 0 设为伺服 OFF。
IF (StartPg =TRUE)
  AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr1_En:=TRUE;           // 将轴 0 设为伺服 ON
  ELSE
    Pwr1_En:=FALSE;        // 将轴 0 设为伺服 OFF
  END_IF;

// 确认 StartPg 为 TRUE, 且伺服驱动器处于伺服准备就绪状态, 则将轴 1 设为伺服 ON。
// 如果未处于伺服准备就绪状态, 则将轴 1 设为伺服 OFF。
IF (StartPg =TRUE)
  AND (MC_Axis001.DrvStatus.Ready=TRUE) THEN
    Pwr2_En:=FALSE;        // 将轴 1 设为伺服 OFF
  ELSE
    Pwr2_En:=TRUE;         // 将轴 1 设为伺服 ON
  END_IF;

// 轴组或构成轴发生轻度故障等级的异常时, 执行异常处理的 FaultHandler
IF (MC_Axis000.MFaultLvl.Active=TRUE)
OR (MC_Axis001.MFaultLvl.Active=TRUE)
OR (MC_Group000.MFaultLvl.Active=TRUE) THEN
  FaultHandler();          // 请根据装置, 对异常处理的内容进行编程
END_IF;

// 轴 0 处于伺服 ON 状态, 且原点未确定时, 执行原点复位
IF (Pwr1_Status=TRUE)
AND (MC_Axis000.Details.Homed=FALSE) THEN
  Hm1_Ex:= TRUE;
END_IF;

// 轴 1 处于伺服 ON 状态, 且原点未确定时, 执行原点复位
IF (Pwr2_Status=TRUE)
AND (MC_Axis001.Details.Homed=FALSE) THEN
  Hm2_Ex:= TRUE;
END_IF;

// 如果轴组失效, 且轴 0 和轴 1 确定了原点, 则使轴组生效
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, 使轴组失效
IF MC_Group000.MFaultLvl.Active=TRUE THEN
  Grp_Dis_Ex:=TRUE;
END_IF;

// 如果 ResetON 因外部按钮进入 ON 状态而变为 TRUE, 且轴组停止, 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 指令异常的监控方法

启动可运行 (MC\_Power) 指令时如果发生异常，则不执行之后的处理。

本示例程序中，用变量 “UpgOn” 表示允许之后的处理。

记载梯形图程序的示例和 ST 程序的示例。

#### 用于程序的主要变量

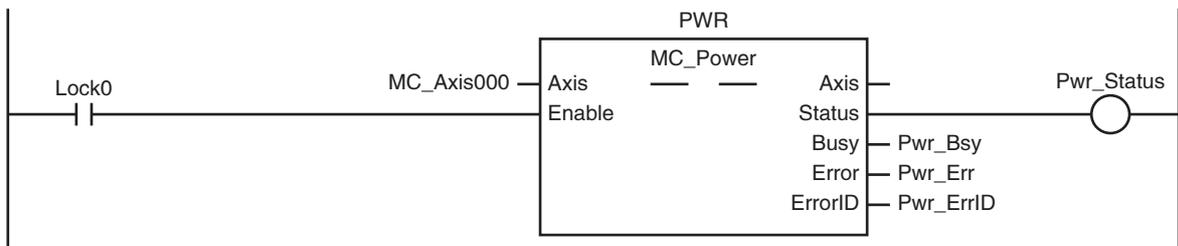
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	-	轴 0 的轴变量
Pwr_Status	BOOL	FALSE	分配给 MC_Power 指令的实例 PWR 的输出 Status 的变量 伺服进入 ON 状态时变为 TRUE
Pwr_Bsy	BOOL	FALSE	分配给 MC_Power 指令的实例 PWR 的输出 Busy 的变量 伺服进入 ON 状态时变为 TRUE
StartPg	BOOL	FALSE	如果该变量为 TRUE，且 EtherCAT 通信的过程数据通信处于正常执行状态，则变为伺服 ON 状态
UpgOn	BOOL	FALSE	TRUE 时执行用户程序之后的处理

#### 梯形图程序

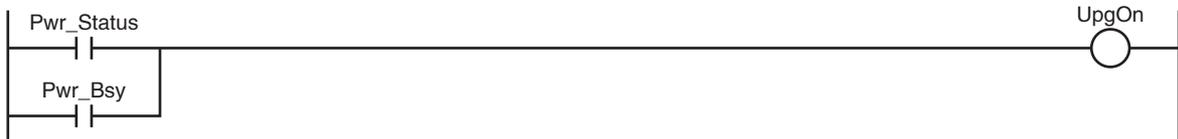
确认触点 StartPg 是否为 TRUE，伺服驱动器是否处于伺服准备就绪状态



如果伺服驱动器处于伺服准备就绪状态，则将轴0伺服ON



确认MC\_Power未发生异常，并执行之后的处理



## ST 程序

```

// 如果触点 StartPg 为 TRUE，且伺服驱动器处于伺服准备就绪状态，则将轴 0 伺服 ON。
// 如果未处于伺服准备就绪状态，则将轴 0 设为伺服 OFF。
IF (StartPg=TRUE)
  AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr_En:=TRUE;
  ELSE
    Pwr_En:=FALSE;
  END_IF;

IF (Pwr_Status=TRUE)
  OR (Pwr_Bsy=TRUE) THEN
    UpgOn := TRUE;           // 执行之后的处理
  ELSE
    UpgOn := FALSE;        // 不执行之后的处理
  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 异常解除的监控方法

如果在发生轻度故障等级的异常的状态下使外部按钮变为 ON 状态，则启动轴错误复位 (MC\_Reset) 指令。不执行下一正常处理，直至轴错误复位 (MC\_Reset) 指令的输出 “Done” 变为 TRUE。

如果输出 “Failure” 变为 TRUE，则轴处于减速停止中或发生 MC 通用异常。获取轴错误复位 (MC\_Reset) 指令的 “Failuer” 变为 TRUE 的原因。

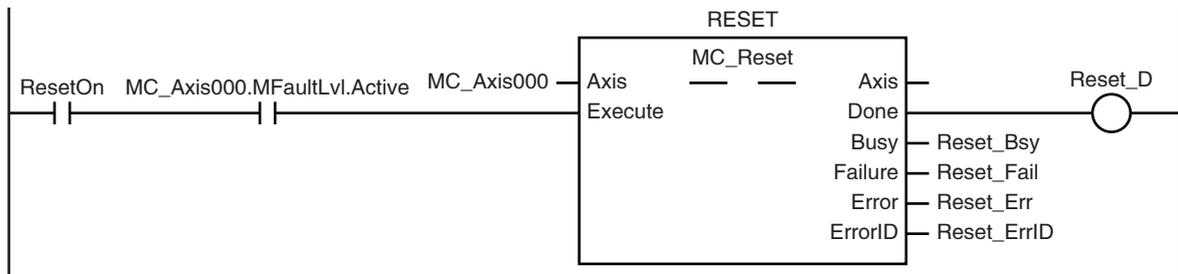
记载梯形图程序的示例和 ST 程序的示例。

### 用于程序的主要变量

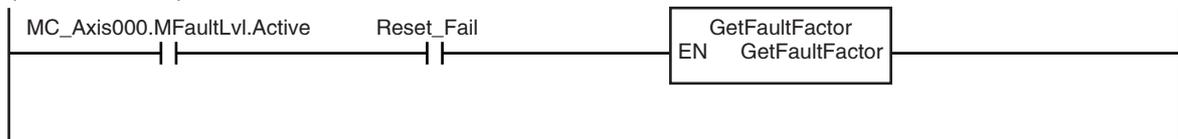
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	-	轴 0 的轴变量
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 0 发生轻度故障等级的异常时变为 TRUE
Pwr_Status	BOOL	FALSE	分配给 MC_Power 指令的实例 PWR 的输出 Status 的变量 伺服进入 ON 状态时变为 TRUE
ResetON	BOOL	FALSE	表示用于解除异常的外部按钮的状态
GetFaultFactor	-	-	异常原因的获取处理 请根据装置进行编程
RegularProcess	-	-	正常处理 请根据装置进行编程

### 梯形图程序

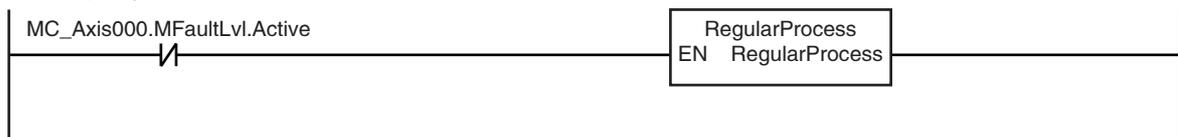
如果在发生轻度故障等级的异常的状态下使外部按钮(ResetOn)变为ON状态，则执行轴错误复位(MC\_Reset)指令



如果轴错误复位(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)。请根据装置，对处理内容进行编程



未发生轻度故障等级的异常或解除了轻度故障等级的异常时，执行装置的正常处理(RegularProcess)。请根据装置，对处理内容进行编程



## ST 程序

```

// 如果在发生轻度故障等级的异常的状态下使外部按钮 (ResetOn) 变为 ON 状态, 则执行轴错误复位 (MC_Reset) 指令
IF (MC_Axis000.MFaultLvl.Active=TRUE)
AND (ResetOn=TRUE) THEN
    Reset_Ex := TRUE;           // 解除轻度故障
ELSE
    Reset_Ex := FALSE;
END_IF;

// 如果轴错误复位 (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)
// 请根据装置, 对处理内容进行编程
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解除轻度故障等级的异常时, 执行装置的正常处理 (RegularProcess)
// 请根据装置, 对处理内容进行编程
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 单轴动作过程中的轴停止

执行绝对值定位 (MC\_MoveAbsolute) 指令时, 如果使外部按钮变为 ON 状态, 则执行强制停止 (MC\_Stop) 指令, 并使其停止减速。发生轻度故障等级的异常时, 强制停止 (MC\_Stop) 指令的输出 “CommandAborted” 变为 TRUE。

此时, 执行立即停止 (MC\_ImmediateStop) 指令, 使其立即停止。

此外, 因某种原因而导致强制停止 (MC\_Stop) 指令的输出 “Error” 变为 TRUE 时, 同样执行立即停止 (MC\_ImmediateStop) 指令, 使其立即停止。如果执行立即停止 (MC\_ImmediateStop) 指令, 则轴进入错误减速停止中。

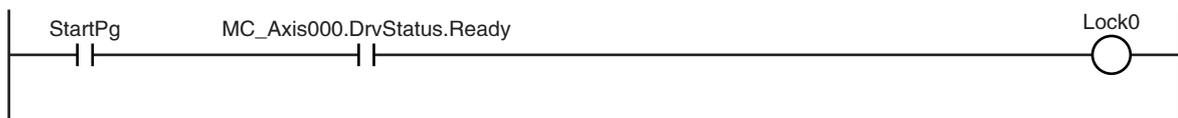
记载梯形图程序的示例和 ST 程序的示例。

#### 用于程序的主要变量

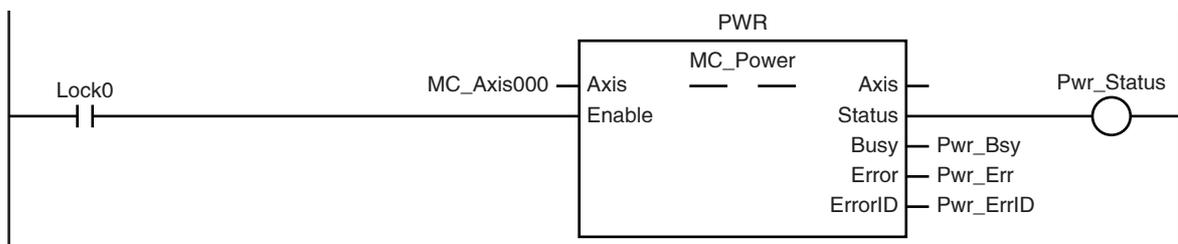
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	-	轴 0 的轴变量
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 0 发生轻度故障等级的异常时变为 TRUE
MC_Axis000.Details.Homed	BOOL	FALSE	轴 0 为原点确定状态时变为 TRUE
Pwr_Status	BOOL	FALSE	分配给 MC_Power 指令的实例 PWR 的输出 Status 的变量 伺服进入 ON 状态时变为 TRUE
Stp_Ca	BOOL	FALSE	分配给 MC_Stop 的实例 STP 的输出变量 CommandAborted 的变量
Stp_Err	BOOL	FALSE	分配给 MC_Stop 的实例 STP 的输出变量 Error 的变量
StartPg	BOOL	FALSE	如果该变量为 TRUE, 且 EtherCAT 通信的过程数据通信处于正常执行状态, 则变为伺服 ON 状态
StopOn	BOOL	FALSE	用于执行强制停止的外部按钮的状态 TRUE 时执行基于 MC_Stop 的强制停止

#### 梯形图程序

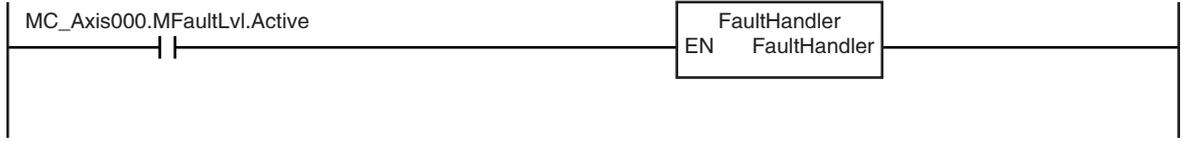
确认触点 StartPg 是否为 TRUE, 轴 0 的伺服驱动器是否处于伺服准备就绪状态



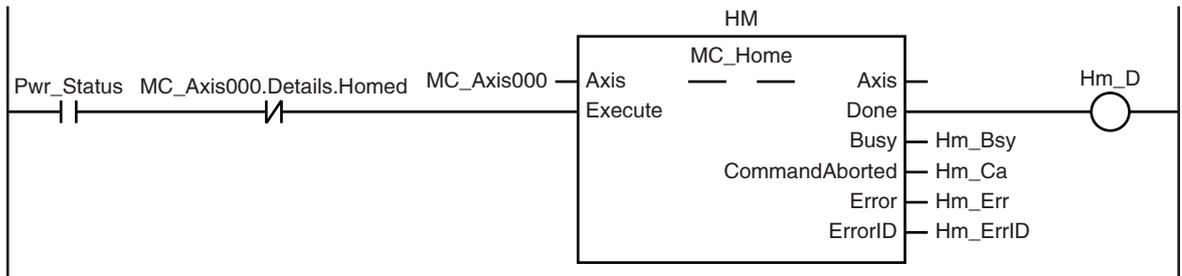
如果轴 0 的伺服驱动器处于伺服准备就绪状态, 则将轴 0 伺服 ON



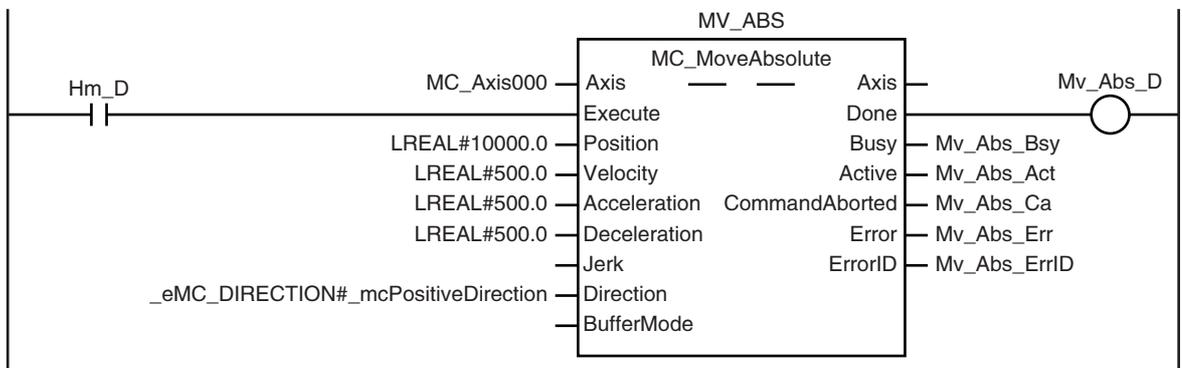
轴0发生轻度故障等级的异常时，执行异常时处理的FaultHandler  
 请根据装置，对异常处理的内容进行编程



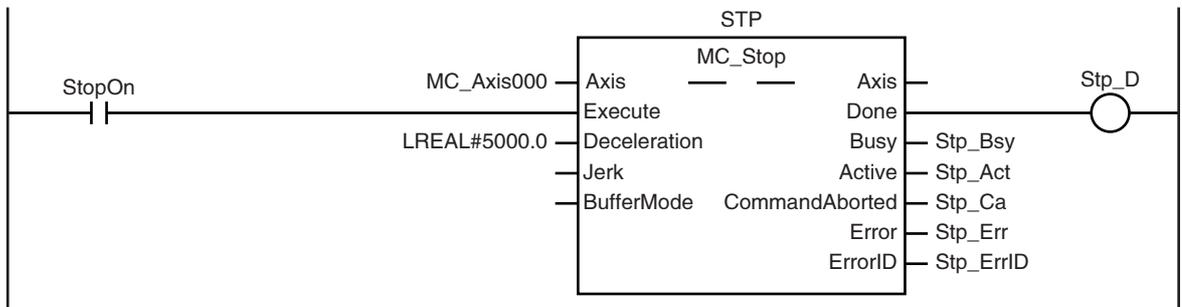
轴0处于伺服ON状态，且原点未确定时，执行轴0的原点复位



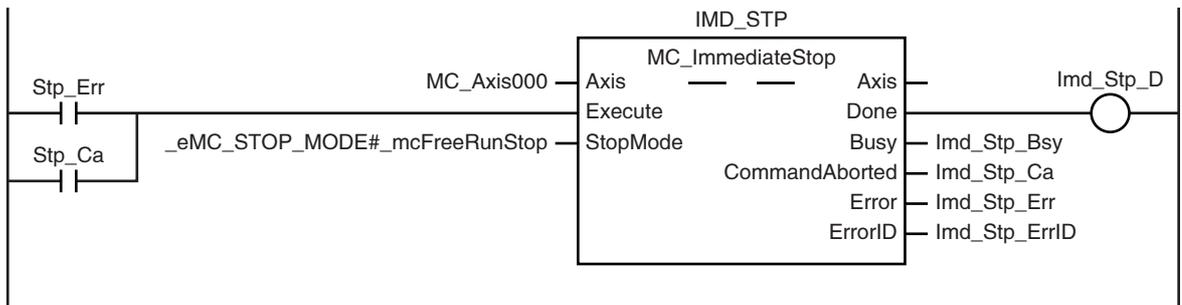
如果轴0的原点复位完成，则执行绝对值定位(MC\_MoveAbsolute)指令



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，使其立即停止



## ST 程序

```

// 绝对值定位和强制停止的输入参数未设定时，设定目标值等
IF InitFlag=FALSE THEN
    // 给绝对值定位 (MC_MoveAbsolute) 的输入参数设定数值
    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;

    // 给强制停止 (MC_Stop) 的输入参数设定数值
    Stp_Dec:=LREAL#5000.0;

    // 给立即停止 (MC_ImmediateStop) 的输入参数设定数值
    Imd_Stp_SM := _eMC_STOP_MODE#_mcFreeRunStop;

    // 将输入参数的设定完成标志设定为 TRUE
    InitFlag := TRUE;
END_IF;

// 如果触点 StartPg 为 TRUE，且伺服驱动器处于伺服准备就绪状态，则将轴 0 伺服 ON。
IF (StartPg=TRUE)
    AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr_En:=TRUE;
ELSE
    Pwr_En:=FALSE;
END_IF;

// 轴 0 发生轻度故障等级的异常时，执行异常处理的 FaultHandler
// 请根据装置，对异常处理的内容进行编程
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
    FaultHandler();
END_IF;

// 轴 0 处于伺服 ON 状态，且原点未确定时，执行原点复位
IF (Pwr_Status=TRUE)
    AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm_Ex:=TRUE;
END_IF;

// 如果原点复位完成，则执行绝对值定位
IF Hm_D=TRUE THEN
    Mv_Abs_Ex := TRUE;
END_IF;

// 如果 StopOn 变为 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) 指令，
使其立即停止
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

执行绝对值直线插补 (MC\_MoveLinearAbsolute) 指令时, 如果使外部按钮变为 ON 状态, 则执行强制停止 (MC\_GroupStop) 指令, 并使其停止减速。发生轻度故障等级的异常时, 轴组强制停止 (MC\_GroupStop) 的输出 “CommandAborted” 变为 TRUE。此时, 执行轴组立即停止 (MC\_GroupImmediateStop) 指令, 使其立即停止。

此外, 因某种原因而导致轴组强制停止 (MC\_GroupStop) 的输出 “Error” 变为 TRUE 时, 同样执行轴组立即停止 (MC\_GroupImmediateStop) 指令, 使其立即停止。如果执行轴组立即停止 (MC\_GroupImmediateStop) 指令, 则轴组处于错误减速停止中。

#### 用于程序的主要变量

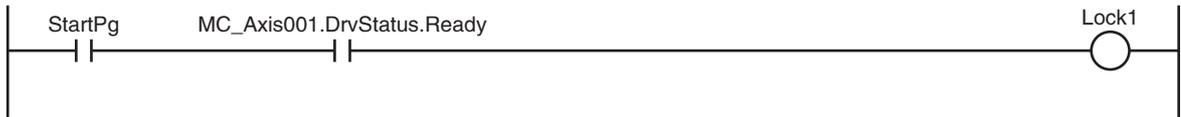
名称	数据类型	初始值	注释
MC_Group000	_sGROUP_REF	-	轴组 0 的轴组变量
MC_Group000.Status.Disabled	BOOL	FALSE	轴组 0 失效时变为 TRUE
MC_Group000.MFaultLvl.Active	BOOL	FALSE	轴组 0 发生轻度故障等级的异常时变为 TRUE
MC_Axis000	_sAXIS_REF	-	轴 0 的轴变量
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 0 发生轻度故障等级的异常时变为 TRUE
MC_Axis000.Details.Homed	BOOL	FALSE	轴 0 为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 为原点确定状态时变为 TRUE
Pwr1_Status	BOOL	FALSE	分配给 MC_Power 指令的实例 PWR1 的输出 Status 的变量 伺服进入 ON 状态时变为 TRUE
Pwr2_Status	BOOL	FALSE	分配给 MC_Power 指令的实例 PWR2 的输出 Status 的变量 伺服进入 ON 状态时变为 TRUE
Grp_Stp_Ca	BOOL	FALSE	分配给 MC_GroupStop 的实例 GRP_EN 的输出变量 CommandAborted 的变量
Grp_Stp_Err	BOOL	FALSE	分配给 MC_GroupStop 的实例 GRP_EN 的输出变量 Error 的变量
StartPg	BOOL	FALSE	如果该变量为 TRUE, 且 EtherCAT 通信的过程数据通信处于正常执行状态, 则变为伺服 ON 状态
StopOn	BOOL	FALSE	用于执行强制停止的外部按钮的状态 TRUE 时执行基于 MC_GroupStop 的强制停止
InitFlag	BOOL	FALSE	表示绝对值直线插补指令和轴组强制停止输入参数的设定完成的变量
Grp_En_Ex	BOOL	FALSE	用于启动 MC_GroupEnable 的实例 GRP_EN 的变量 用于 ST 程序
Mv_Lin_Abs_Ex	BOOL	FALSE	用于启动 MC_MoveLinear 的实例 MV_LIN_ABS 的变量 用于 ST 程序
Grp_Stp_Ex	BOOL	FALSE	用于启动 MC_GroupStop 的实例 GRP_STP 的变量 用于 ST 程序
Grp_Imd_Stp_Ex	BOOL	FALSE	用于启动 MC_GroupImmediateStop 的实例 GRP_IMD_STP 的变量 用于 ST 程序

## 梯形图程序

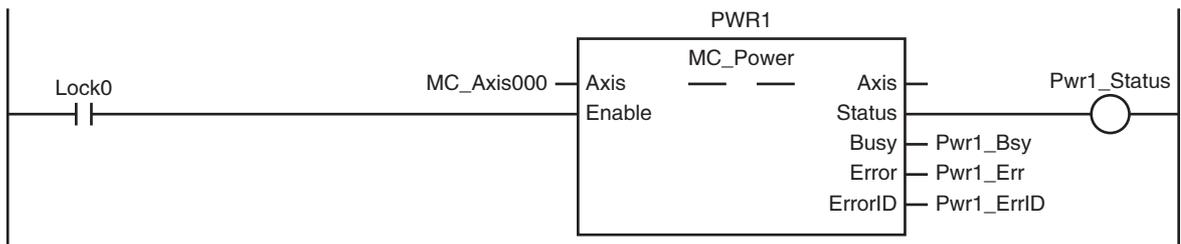
确认触点StartPg是否为TRUE，轴0的伺服驱动器是否处于伺服准备就绪状态



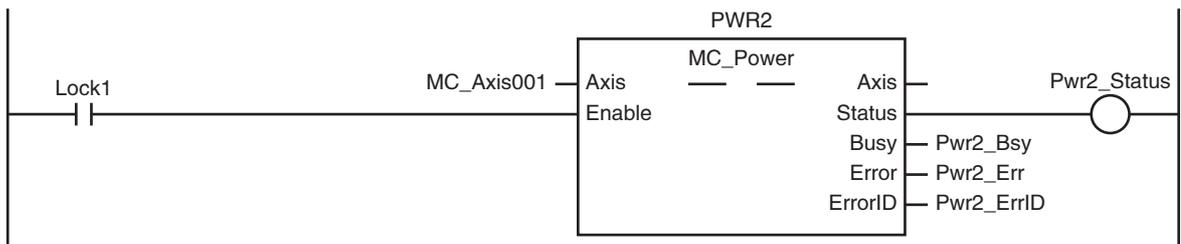
确认触点StartPg是否为TRUE，轴1的伺服驱动器是否处于伺服准备就绪状态



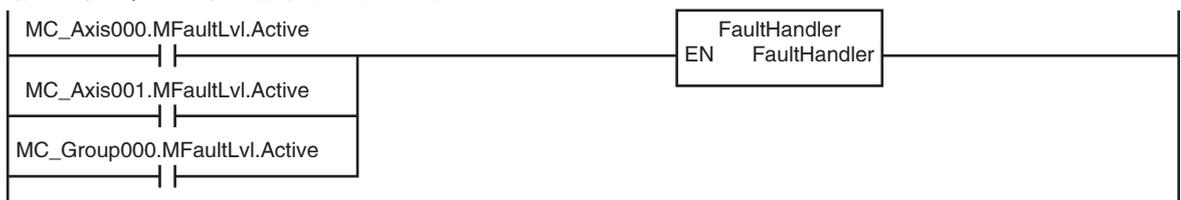
如果轴 0 的伺服驱动器处于伺服准备就绪状态，则将轴0伺服ON



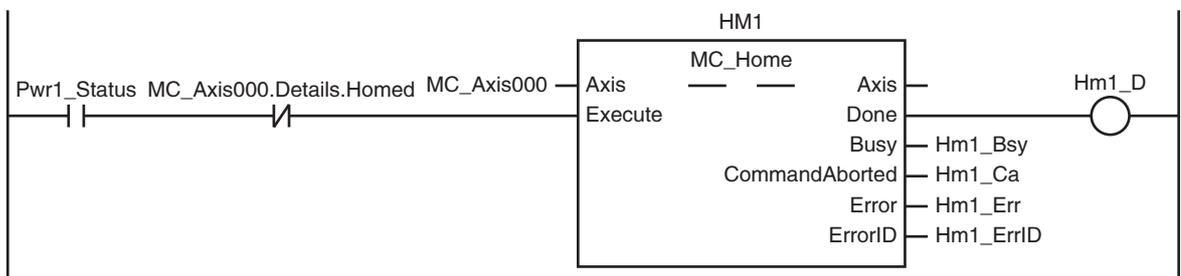
如果轴 1 的伺服驱动器处于伺服准备就绪状态，则将轴1伺服ON



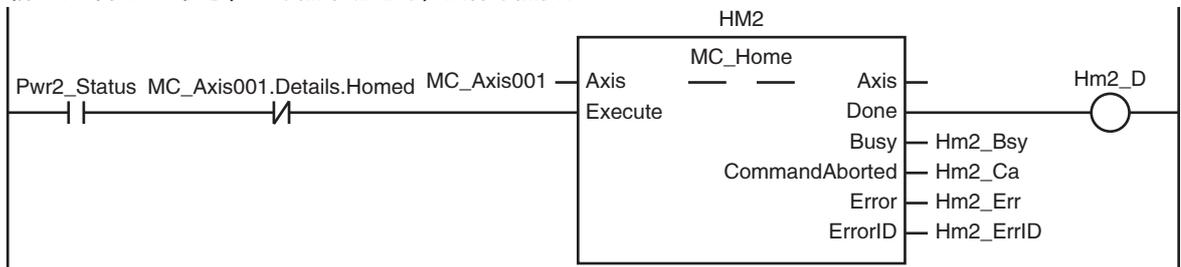
轴组或构成轴发生轻度故障等级的异常时，执行异常处理的FaultHandler。  
请根据装置，对异常处理的内容进行编程



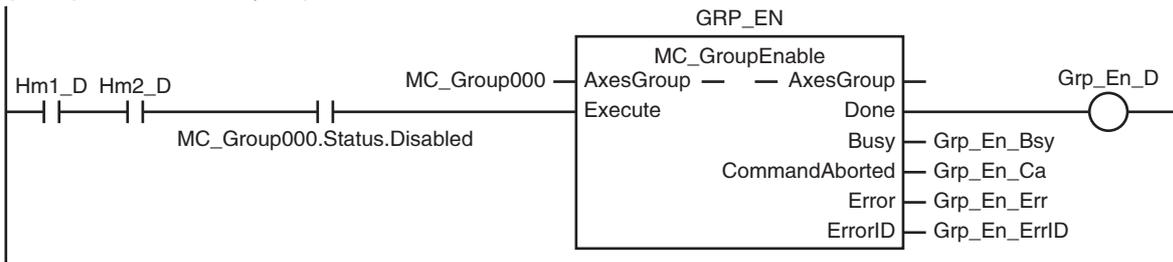
轴0处于伺服ON状态，且原点未确定时，执行原点复位



轴1处于伺服ON状态，且原点未确定时，执行原点复位



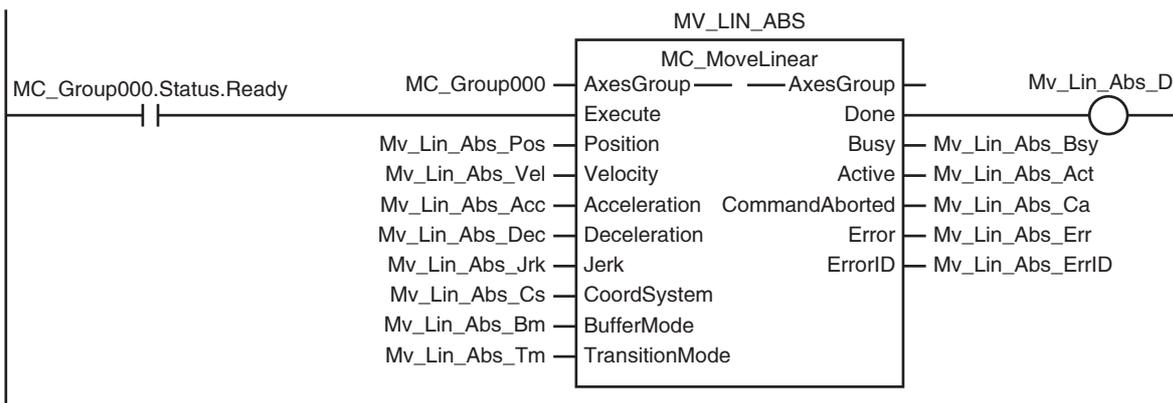
轴0和轴1的原点确定后，使轴组生效



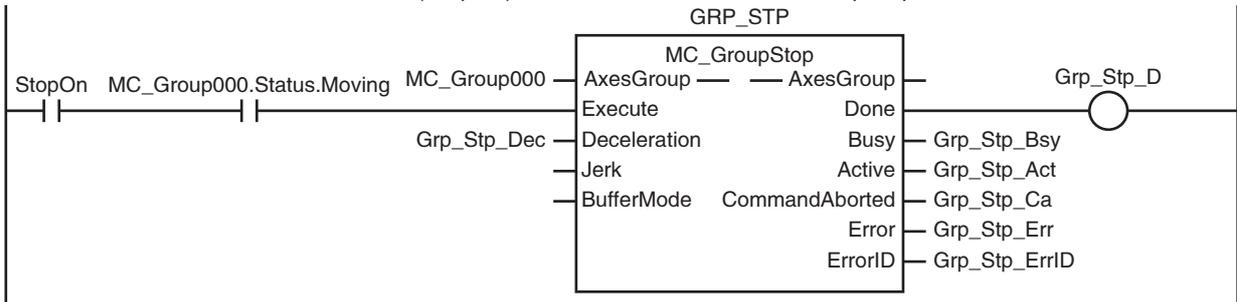
设定绝对值直线插补和轴组强制停止的输入参数



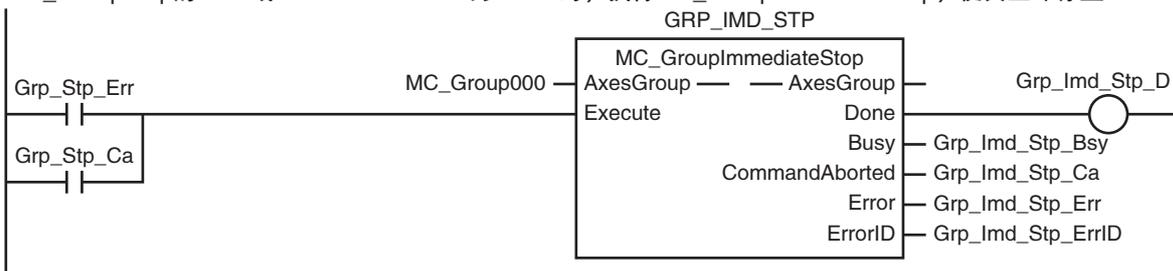
确认轴组生效，并执行直线插补



如果在直线插补动作过程中将外部按钮(StopOn)设定为TRUE，则执行MC\_GroupStop，使轴停止减速



MC\_GroupStop的Error或CommandAborted为TRUE时，执行MC\_GroupImmediateStop，使其立即停止



## ST 程序

```

// 绝对值直线插补和轴组强制停止的输入参数未设定时，设定目标值等
IF InitFlag=FALSE THEN

    // 给绝对值直线插补 (MC_MoveLinearAbsolute) 的输入参数设定数值
    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;

    // 给轴组强制停止 (MC_GroupStop) 的输入参数设定数值
    Grp_Stp_Dec := LREAL#1000.0;
    Grp_Stp_Jrk := LREAL#1000.0;

    // 将输入参数的设定完成标志设定为 TRUE
    InitFlag := TRUE;
END_IF;

// 如果 StartPg 为 TRUE，且过程数据通信处于正常执行状态，则使轴 0 变为伺服 ON 状态
// 未执行过程数据通信时，变为伺服 OFF 状态
IF (StartPg =TRUE)
AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr1_En:=TRUE ; // 将轴 0 设为伺服 ON
ELSE
    Pwr1_En:=FALSE; // 将轴 0 设为伺服 OFF
END_IF;

// 如果 StartPg 为 TRUE，且伺服驱动器处于伺服准备就绪状态，则将轴 1 伺服 ON。
// 如果未处于伺服准备就绪状态，则将轴 1 设为伺服 OFF。
IF (StartPg =TRUE)
AND (MC_Axis001.DrvStatus.Ready=TRUE) THEN
    Pwr2_En:=TRUE; // 将轴 1 设为伺服 ON
ELSE
    Pwr2_En:=FALSE; // 将轴 1 设为伺服 OFF
END_IF;

// 发生轻度故障等级的异常时，执行异常处理的 FaultHandler
// 请根据装置，对异常处理的内容进行编程
IF (MC_Axis000.MFaultLvl.Active=TRUE)
OR (MC_Axis001.MFaultLvl.Active=TRUE)
OR (MC_Group000.MFaultLvl.Active=TRUE) THEN
    FaultHandler();
END_IF;

// 轴 0 处于伺服 ON 状态，且原点未确定时，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 处于伺服 ON 状态，且原点未确定时，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 和轴 1 确定了原点时，如果轴组失效，则使轴组生效
IF (MC_Group000.Status.Disabled=TRUE)
AND (Hm1_D=TRUE) AND (Hm2_D=TRUE) THEN
    Grp_En_Ex:= TRUE;
END_IF;

// 轴组生效时，执行绝对值直线插补
IF MC_Group000.Status.Ready=TRUE THEN
    Mv_Lin_Abs_Ex:=TRUE;
END_IF;

// 执行绝对值直线插补时，如果通过外部按钮将 StopOn 设定为 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_lmd_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
);

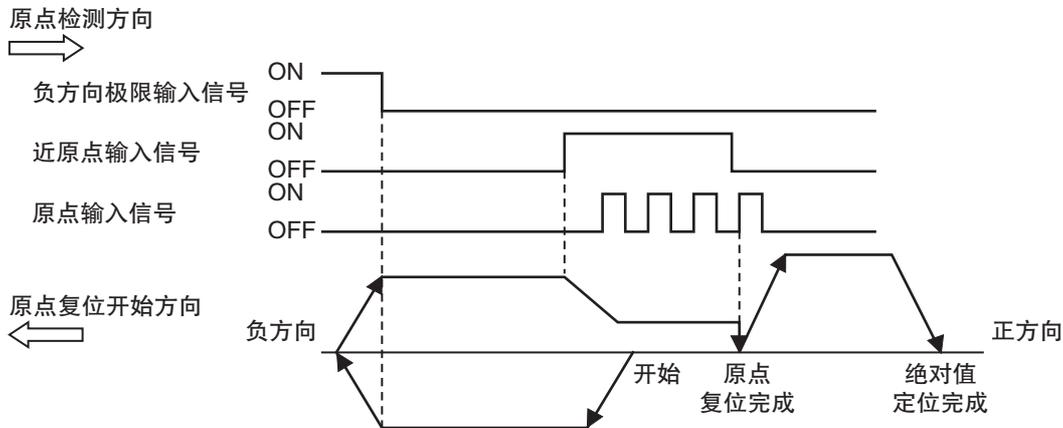
//MC_GroupStop
GRP_STP(
    AxesGroup           := MC_Group000,
    Execute             := Grp_Stp_Ex,
    Deceleration        := Grp_Stp_Dec,
    Done                => Grp_Stp_D,
    Busy                => Grp_Stp_Bsy,
    Active              => Grp_Stp_Act,
    CommandAborted     => Grp_Stp_Ca,
    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

近原点输入变为 ON 状态时的位置即为原点复位开始位置。通过“4: 指定为近原点输入 OFF”执行原点复位。如果原点复位正常完成且原点确定，则执行绝对值定位。



记载梯形图程序的示例和 ST 程序的示例。

### 原点复位相关轴参数的设定

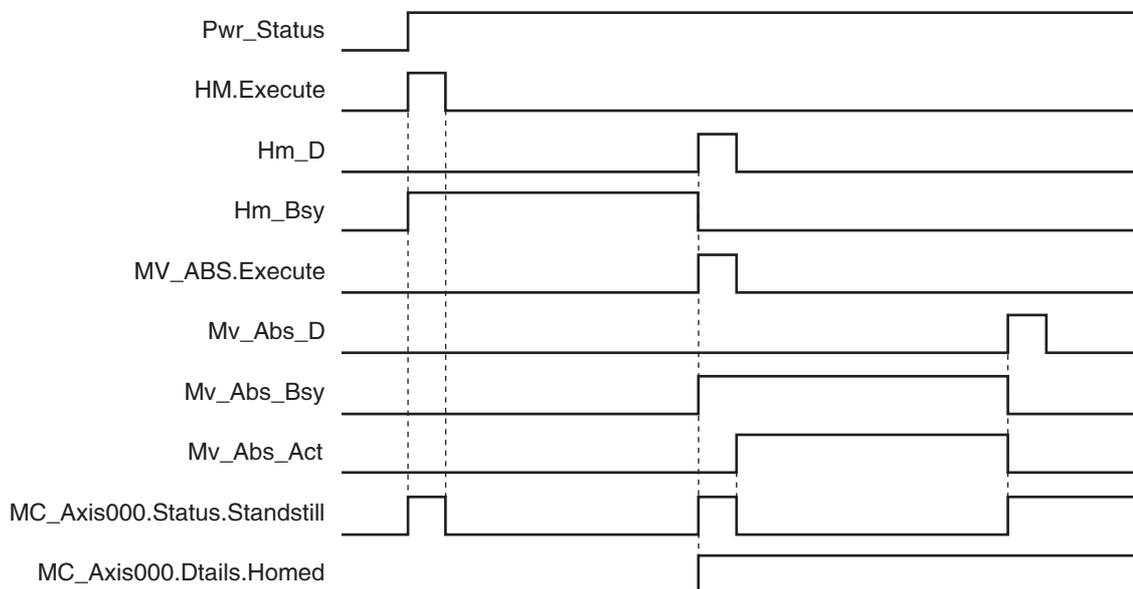
设定项目	设定内容	说明
原点复位动作	4: 指定为近原点输入 OFF	通过指定为近原点输入 OFF 确定原点
正方向极限输入时动作	1: 反转、立即停止	不使用正方向极限输入, 因此保持为初始值
负方向极限输入时动作	2: 反转、减速停止	如果负方向极限输入生效, 则停止减速并反转
原点复位开始方向	2: 负方向	如果启动原点复位, 则开始朝负方向动作
原点检测方向	1: 正方向	检测原点时的动作方向为正方向

### 用于程序的主要变量

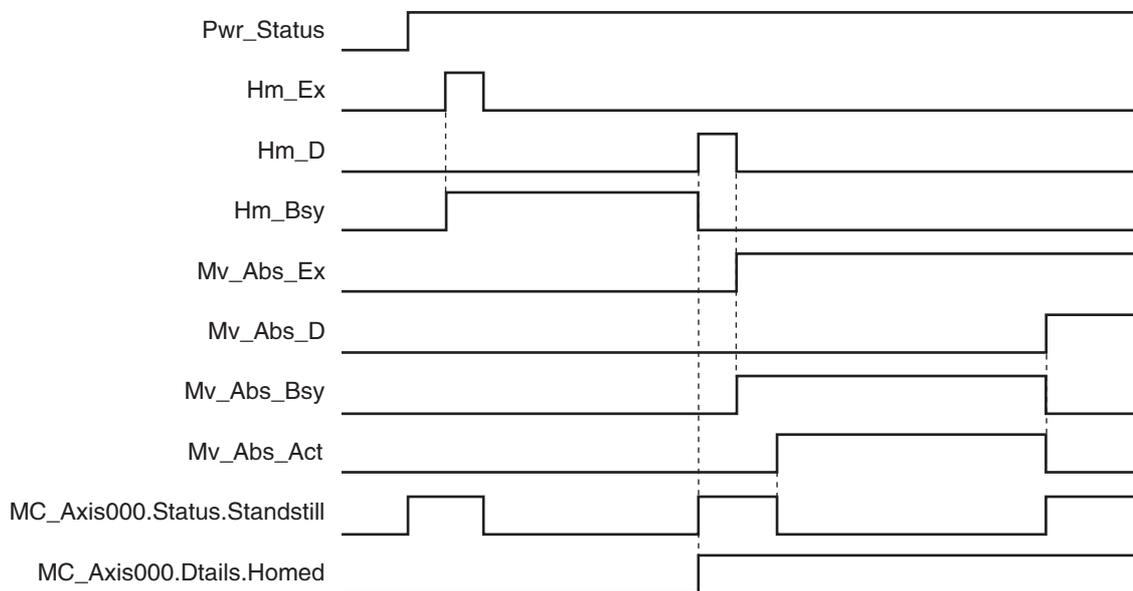
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	-	轴 0 的轴变量
MC_Axis000.Status.StandStill	BOOL	FALSE	在轴 0 由于伺服 ON 而停止的状态下变为 TRUE。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 0 发生轻度故障等级的异常时变为 TRUE
MC_Axis000.Details.Homed	BOOL	FALSE	轴 0 为原点确定状态时变为 TRUE
Pwr_Status	BOOL	FALSE	分配给 MC_Power 指令的实例 PWR 的输出 Status 的变量 伺服进入 ON 状态时变为 TRUE
StartPg	BOOL	FALSE	如果该变量为 TRUE, 且 EtherCAT 通信的过程数据通信处于正常执行状态, 则变为伺服 ON 状态
Hm_Ex	BOOL	FALSE	用于启动 MC_Home(原点复位) 指令的变量 用于 ST 程序
Mv_Abs_Ex	BOOL	FALSE	用于启动 MC_MoveAbsolute(绝对值定位) 指令的变量 用于 ST 程序

## 时序图

### ● 梯形图程序时

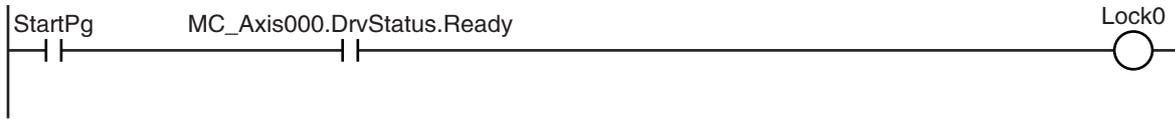


### ● ST 程序时

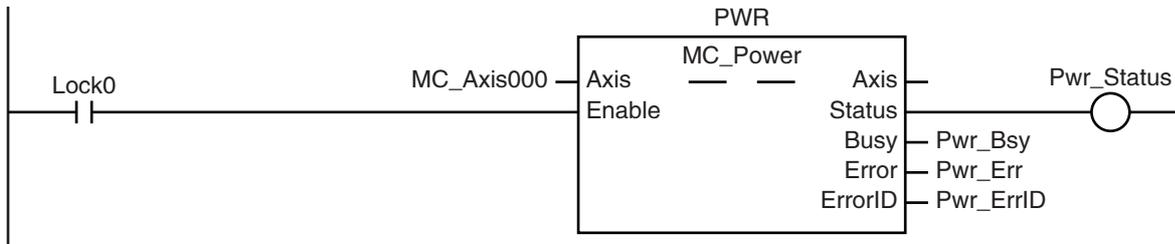


## 梯形图程序

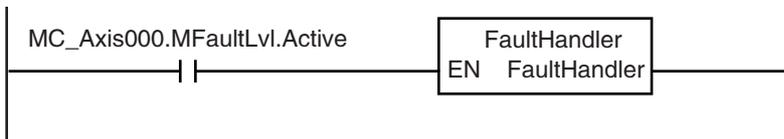
确认触点StartPg是否为TRUE，轴0的伺服驱动器是否处于伺服准备就绪状态



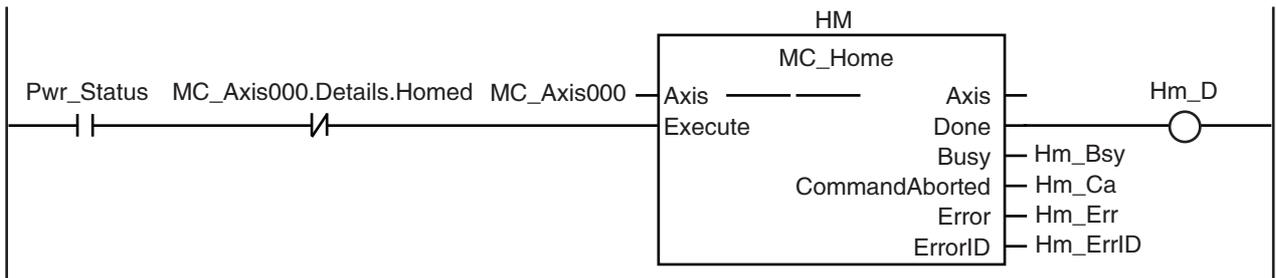
如果轴0的伺服驱动器处于伺服准备就绪状态，则将轴0伺服ON



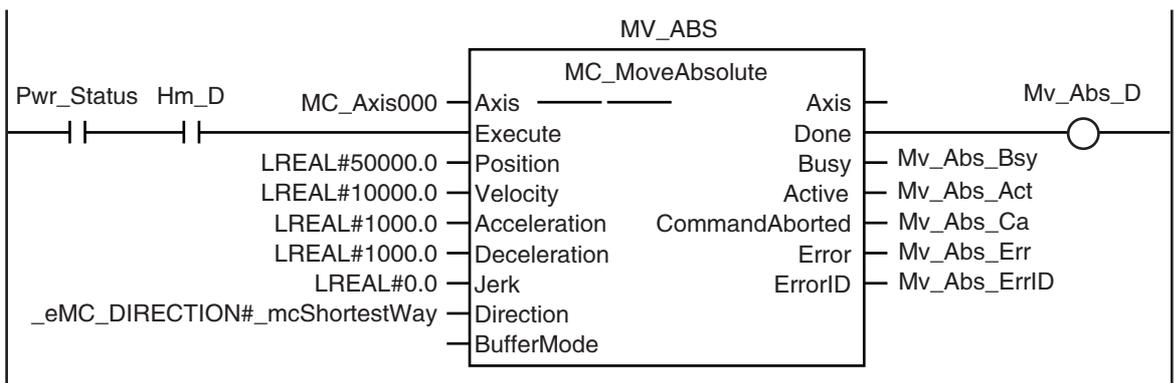
轴0发生轻度故障等级的异常时，执行异常处理的FaultHandler  
请根据装置，对异常处理的内容进行编程



轴0处于伺服ON状态，且原点未确定时，执行原点复位



如果在伺服ON状态下确定原点，则执行绝对值定位



## ST 程序

```

// 绝对值定位的输入参数未设定时, 设定目标值等
IF InitFlag=FALSE THEN
    // 给绝对值定位 (MC_MoveAbsolute) 的输入参数设定数值
    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;

    // 使输入参数的设定完成标志生效
    InitFlag := TRUE;
END_IF;

// 如果触点 StartPg 为 TRUE, 且过程数据通信处于正常执行状态, 则使轴 0 变为伺服 ON 状态
IF (StartPg=TRUE)
    AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr_En:=TRUE;
ELSE
    Pwr_En:=FALSE;
END_IF;

// 轴 0 发生轻度故障等级的异常时, 执行异常处理的 FaultHandler
// 请根据装置, 对异常处理的内容进行编程
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
    FaultHandler();
END_IF;

// 轴 0 处于伺服 ON 状态, 且原点未确定时, 执行原点复位
IF (Pwr_Status=TRUE)
    AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm_Ex:=TRUE;
ELSE
    Hm_Ex:=FALSE;
END_IF;

// 如果在伺服 ON 状态下确定原点, 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 通过重启指令变更目标位置

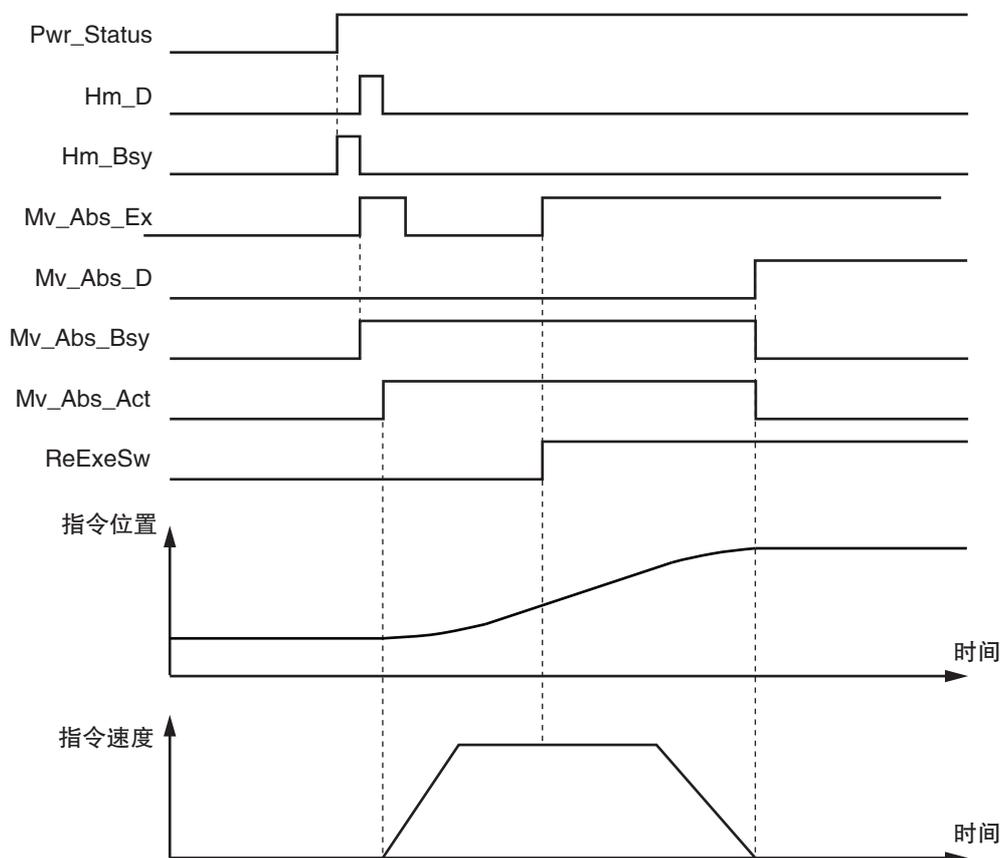
在针对目标位置 1000 执行绝对值定位动作的过程中，使用相同的指令实例将目标位置变更为 2000。

### 用于程序的主要变量

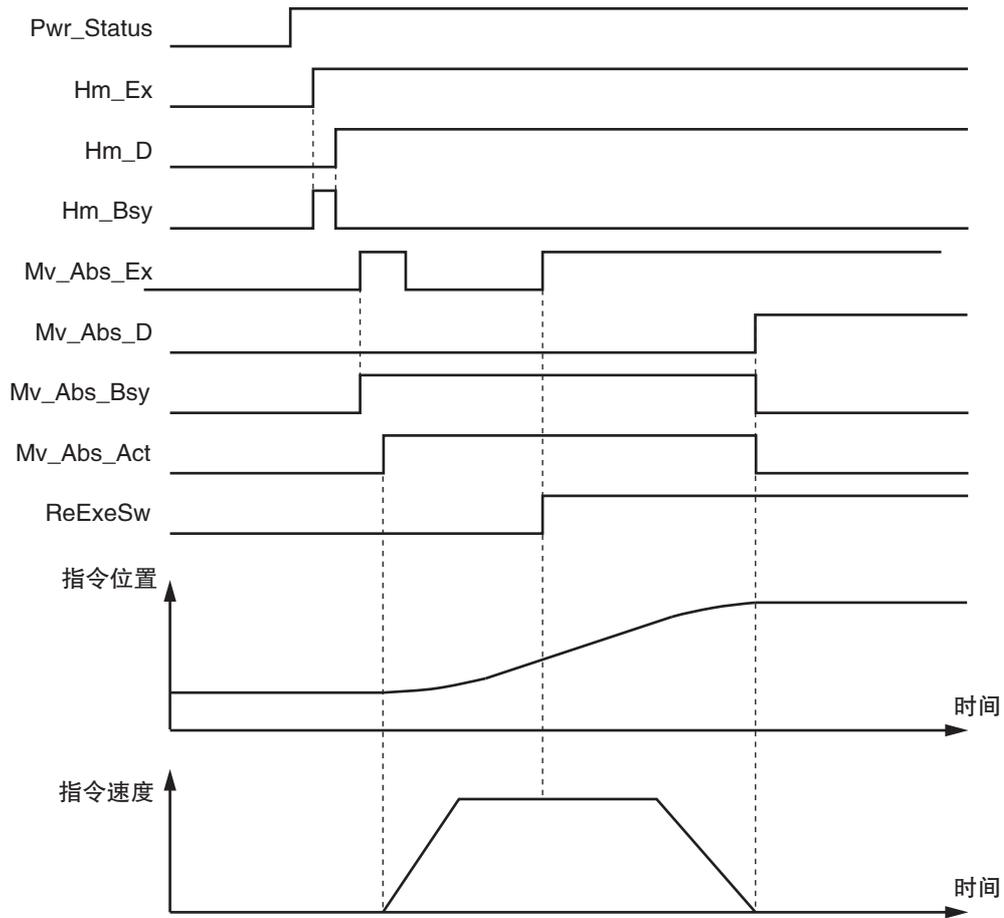
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	-	轴 0 的轴变量
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 0 发生轻度故障等级的异常时变为 TRUE
MC_Axis000.Details.Homed	BOOL	FALSE	轴 0 为原点确定状态时变为 TRUE
Pwr_Status	BOOL	FALSE	分配给 MC_Power 指令的实例 PWR 的输出 Status 的变量 伺服进入 ON 状态时变为 TRUE
StartPg	BOOL	FALSE	如果该变量为 TRUE，且 EtherCAT 通信的过程数据通信处于正常执行状态，则变为伺服 ON 状态
ReExeSw	BOOL	FALSE	用于重启指令的变量
Hm_Ex	BOOL	FALSE	用于启动 MC_Home(原点复位)指令的变量 用于 ST 程序
Mv_Abs_Ex	BOOL	FALSE	用于启动 MC_MoveAbsolute(绝对值定位)指令的变量 用于 ST 程序

### 时序图

#### ● 梯形图程序时

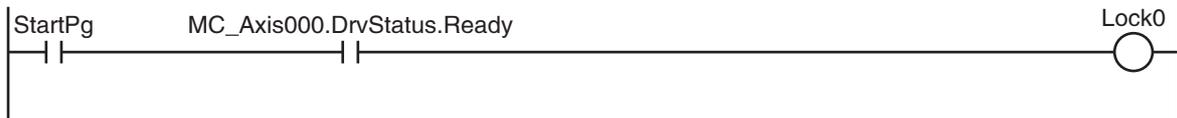


● ST 程序时

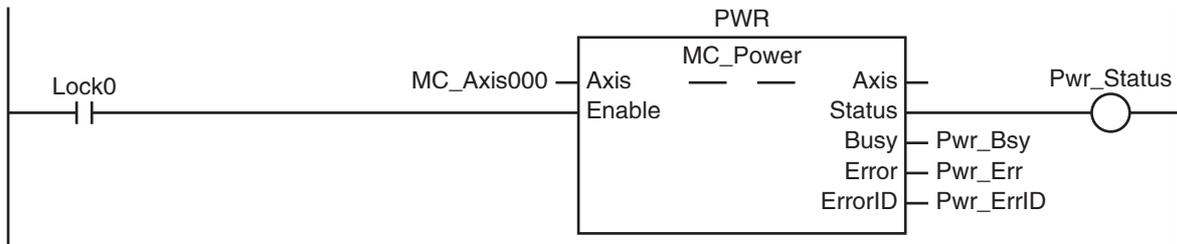


## 梯形图程序

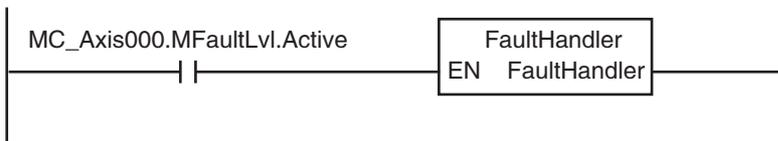
确认触点 StartPg 是否为 TRUE，轴 0 的伺服驱动器是否处于伺服准备就绪状态



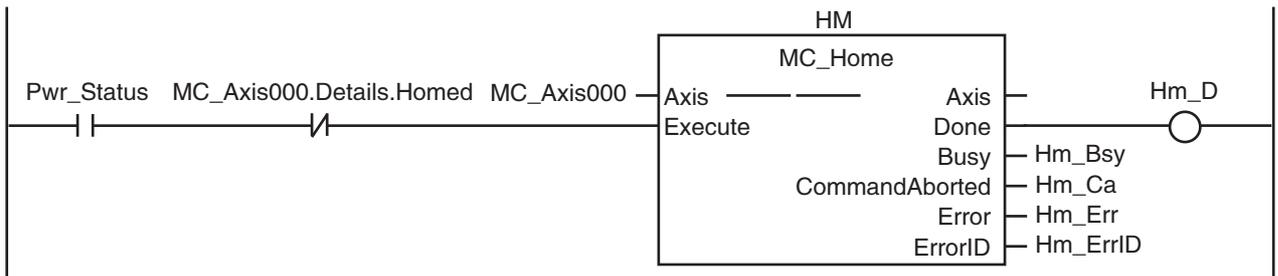
如果轴 0 的伺服驱动器处于伺服准备就绪状态，则将轴 0 伺服 ON



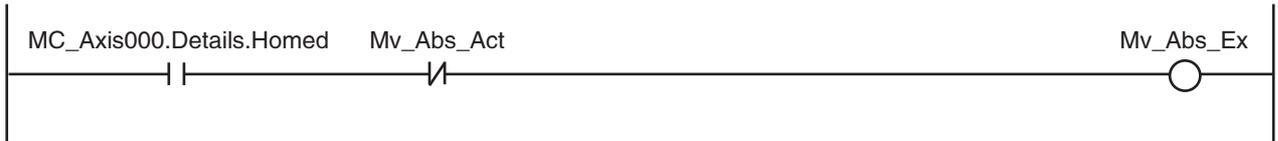
轴 0 发生轻度故障等级的异常时，执行异常处理的 FaultHandler  
请根据装置，对异常处理的内容进行编程



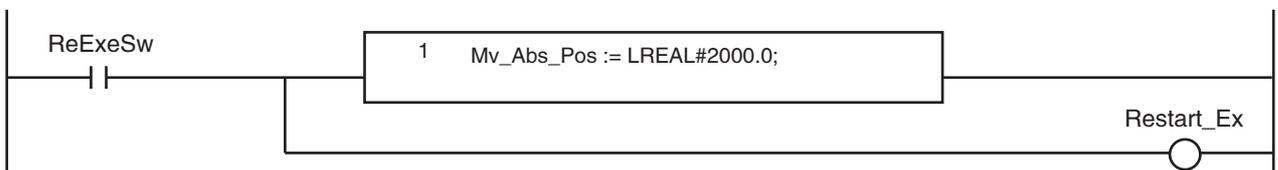
轴 0 处于伺服 ON 状态，且原点未确定时，执行原点复位



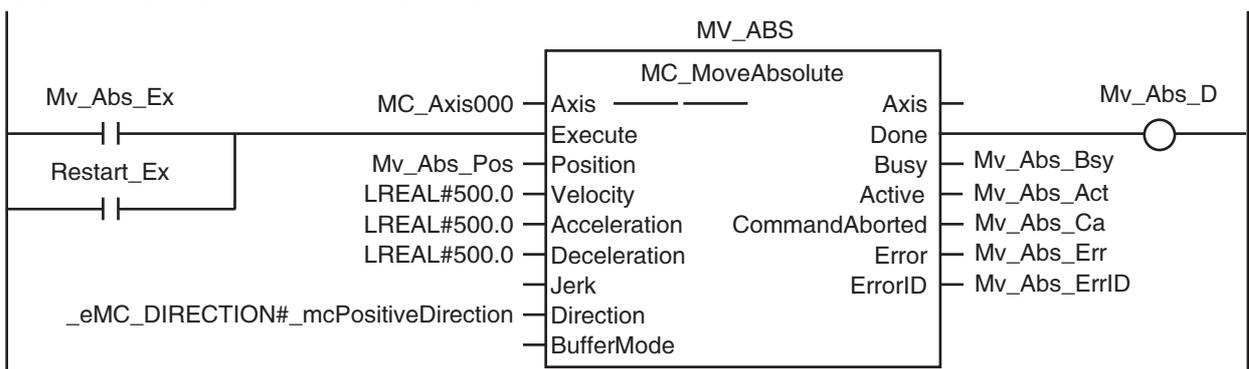
轴 0 确定了原点，如果未执行绝对值定位，则予以执行



如果将触点 ReExeSw 设定为 TRUE，则将目标位置变更为 2000 并重启绝对值定位



通过触点 Mv\_Abs\_Ex 的状态，执行绝对值定位



## ST 程序

```

// 绝对值定位的输入参数未设定时，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;

    // 使输入参数的设定完成标志生效
    InitFlag:=TRUE;
END_IF;

// 如果触点 StartPg 为 TRUE，且过程数据通信处于正常执行状态，则使轴 0 变为伺服 ON 状态
IF (StartPg=TRUE)
    AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr_En:=TRUE;
ELSE
    Pwr_En:=FALSE;
END_IF;

// 轴 0 发生轻度故障等级的异常时，执行异常处理的 FaultHandler
// 请根据装置，对异常处理的内容进行编程
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
    FaultHandler();
END_IF;

// 轴 0 处于伺服 ON 状态，且原点未确定时，执行原点复位
IF (Pwr_Status=TRUE)
    AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm_Ex:=TRUE;
END_IF;

// 轴 0 完成了原点复位后，如果未执行绝对值定位，则予以执行
IF (Hm_D=TRUE)
    AND (Mv_Abs_Act=FALSE) THEN
    Mv_Abs_Ex := TRUE;
ELSE
    Mv_Abs_Ex := FALSE;
END_IF;

// 如果将触点 ReExeSw 设定为 TRUE，则将目标位置变更为 2000 并重启绝对值定位
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 中断标准定位

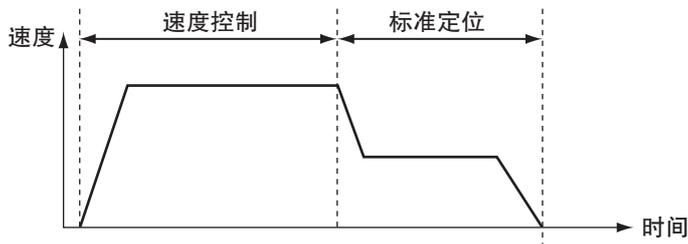
如果速度控制过程中发生中断输入，则切换至标准传送。

在旋转模式下进行速度控制时，方向选择“Direction”可指定下述任意一种。

- `_mcPositiveDirection`(指定为正方向)
- `_mcNegativeDirection`(指定为负方向)
- `_mcCurrentDirection`(指定为当前方向)

在该示例程序中指定“`_mcCurrentDirection`(指定为当前方向)”。

朝与发生中断输入之前相同的方向进行标准传送时，输入变量“`FeedDistance`(标准距离)”设定为正数。朝与发生中断输入之前相反的方向进行标准传送时，输入变量“`FeedDistance`(标准距离)”设定为负数。例如，发生中断输入之前的动作为负方向时如果输入变量“`FeedDistance`(标准距离)”设定为正数，则朝负方向执行标准传送。如果输入变量“`FeedDistance`(标准距离)”设定为负数，则朝正方向执行标准传送。



## 轴参数设定

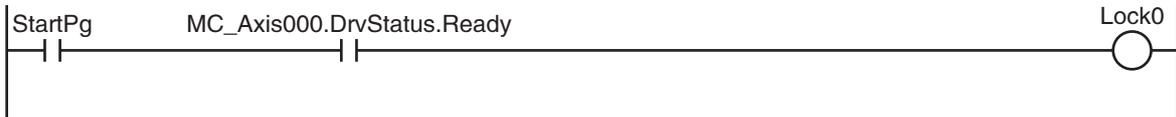
设定项目	设定内容	说明
计数模式	旋转模式	将位置的计数模式设定为旋转模式
环计数器上限设定值	360	将环计数器的上限值设定为“360”
环计数器下限设定值	0	将环计数器的下限值设定为“0”
原点复位动作	原点预设	通过原点预设确定原点

## 用于程序的主要变量

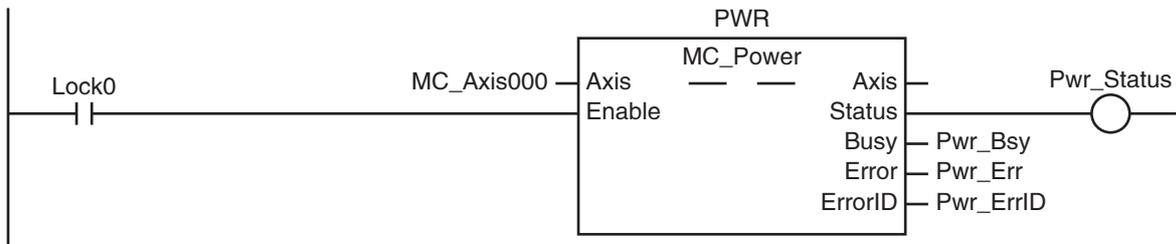
名称	数据类型	初始值	注释
<code>MC_Axis000</code>	<code>_sAXIS_REF</code>	-	轴 0 的轴变量
<code>MC_Axis000.MFaultLvl.Active</code>	BOOL	FALSE	轴 0 发生轻度故障等级的异常时变为 TRUE
<code>MC_Axis000.Details.Homed</code>	BOOL	FALSE	轴 0 为原点确定状态时变为 TRUE
<code>Pwr_Status</code>	BOOL	FALSE	分配给 <code>MC_Power</code> 指令的实例 PWR 的输出 Status 的变量 伺服进入 ON 状态时变为 TRUE
<code>StartPg</code>	BOOL	FALSE	如果该变量为 TRUE，且 EtherCAT 通信的过程数据通信处于正常执行状态，则变为伺服 ON 状态
<code>TrigRef</code>	<code>_sTRIGGER_REF</code>	-	指定中断输入的触发输入条件 该示例程序中使用了伺服驱动器的锁定功能 1
<code>Hm_Ex</code>	BOOL	FALSE	用于启动 <code>MC_Home</code> (原点复位)指令的变量 用于 ST 程序
<code>Mv_Feed_Ex</code>	BOOL	FALSE	用于启动 <code>MC_MoveFeed</code> (中断标准定位)指令的变量 用于 ST 程序
<code>InitFlag</code>	BOOL	FALSE	表示 <code>MC_MoveFeed</code> (中断标准定位)指令的输入参数设定完成的变量

## 梯形图程序

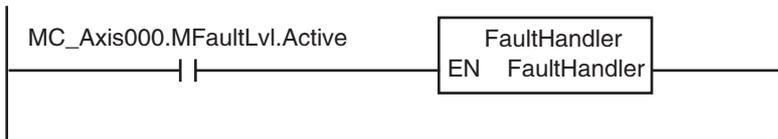
确认触点 StartPg 是否为 TRUE，轴 0 的伺服驱动器是否处于伺服准备就绪状态



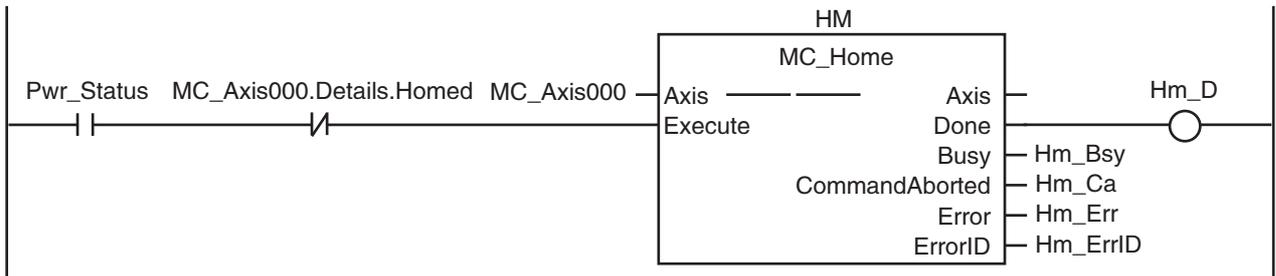
如果轴 0 的伺服驱动器处于伺服准备就绪状态，则将轴 0 伺服 ON



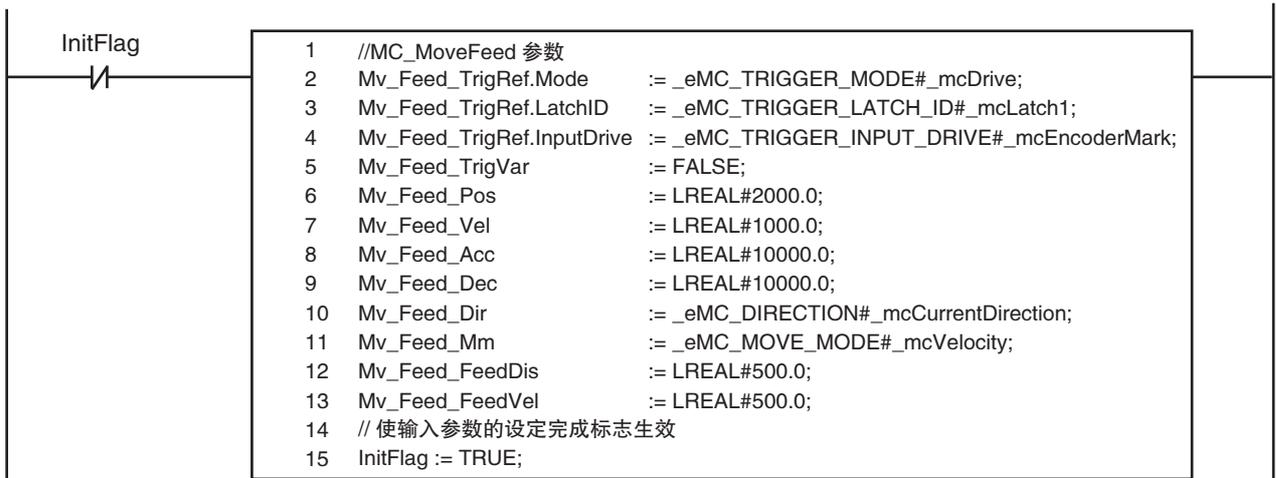
轴 0 发生轻度故障等级的异常时，执行异常处理的 FaultHandler  
请根据装置，对异常处理的内容进行编程



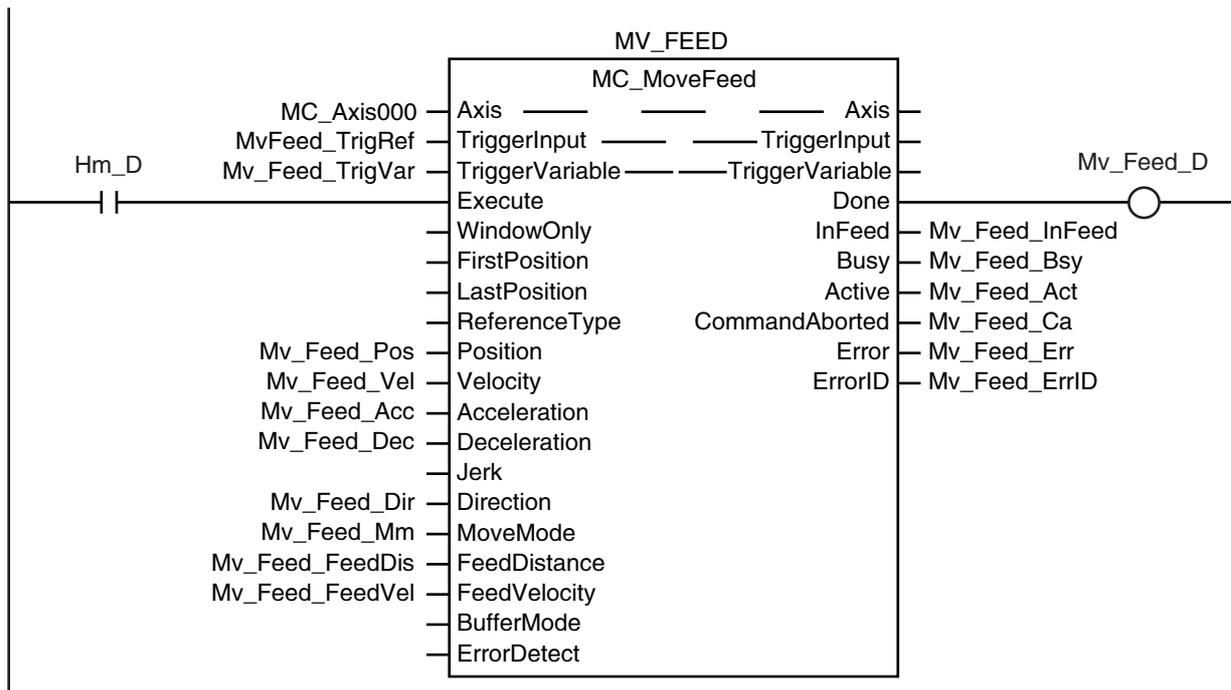
轴 0 处于伺服 ON 状态，且原点未确定时，执行原点复位



设定中断标准定位的输入参数



如果原点复位完成，则执行中断标准定位



## ST 程序

// 中断标准定位的输入参数未设定时，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;
```

// 使输入参数的设定完成标志生效

```
InitFlag := TRUE;
```

```
END_IF;
```

// 如果触点 StartPg 为 TRUE，且过程数据通信处于正常执行状态，则使轴 0 变为伺服 ON 状态

```
IF (StartPg=TRUE)
  AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
  Pwr_En:=TRUE;
ELSE
  Pwr_En:=FALSE;
END_IF;
```

// 轴 0 发生轻度故障等级的异常时，执行异常处理的 FaultHandler

```
// 请根据装置，对异常处理的内容进行编程
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
  FaultHandler();
END_IF;
```

```

//轴 0 处于伺服 ON 状态, 且原点未确定时, 执行原点复位
IF (Pwr_Status=TRUE)
AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm_Ex:=TRUE;
END_IF;

// 如果确定原点, 则执行中断标准定位
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

执行凸轮动作时切换凸轮表。

轴 0 的指令位置不超过 5000 时，使用 CamProfile0，如果超过 5000，则使用 CamTable1。

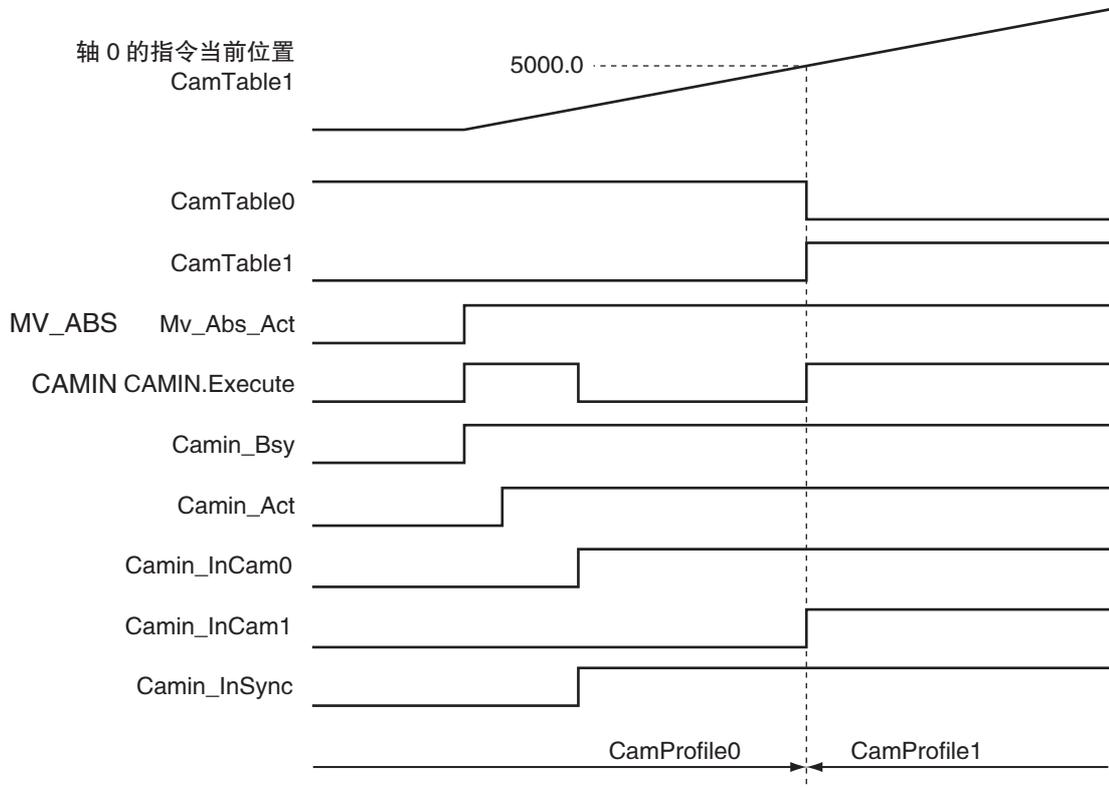
## 用于程序的主要变量

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	分配给 MC_Power 指令的实例 PWR1 的输出 Status 的变量 伺服进入 ON 状态时变为 TRUE
Pwr2_S	BOOL	FALSE	分配给 MC_Power 指令的实例 PWR2 的输出 Status 的变量 伺服进入 ON 状态时变为 TRUE
CamTable0	BOOL	FALSE	将 CamProfile0 作为凸轮表使用时设定为 TRUE
CamTable1	BOOL	FALSE	将 CamProfile1 作为凸轮表使用时设定为 TRUE
Camin_InCam0	BOOL	FALSE	分配给 MC_CamIn 指令的实例 CAMIN 的输出 InCam 的变量 通过 CamProfile0 执行凸轮动作时变为 TRUE 一旦变为 TRUE，则保持 TRUE，直至动作结束 或中断
Camin_InCam1	BOOL	FALSE	分配给 MC_CamIn 指令的实例 CAMIN 的输出 InCam 的变量 通过 CamProfile1 执行凸轮动作时变为 TRUE 一旦变为 TRUE，则保持 TRUE，直至动作结束 或中断
Mv_Abs_Act	BOOL	FALSE	分配给 MC_MoveAbsolute 指令的实例 MV_ABS 的输出 Active 的变量
StartPg	BOOL	FALSE	如果该变量为 TRUE，且 EtherCAT 通信的过程 数据通信处于正常执行状态，则变为伺服 ON 状态
Hm1_Ex	BOOL	FALSE	用于通过轴 0 启动 MC_Home(原点复位) 指令 的变量
Hm2_Ex	BOOL	FALSE	用于通过轴 1 启动 MC_Home(原点复位) 指令 的变量
Mv_Abs_Ex	BOOL	FALSE	用于启动 MC_MoveAbsolute(绝对值定位) 指令 的变量
Camin_Ex	BOOL	FALSE	用于启动 MC_CamIn(凸轮动作开始) 指令的变 量 用于 ST 程序

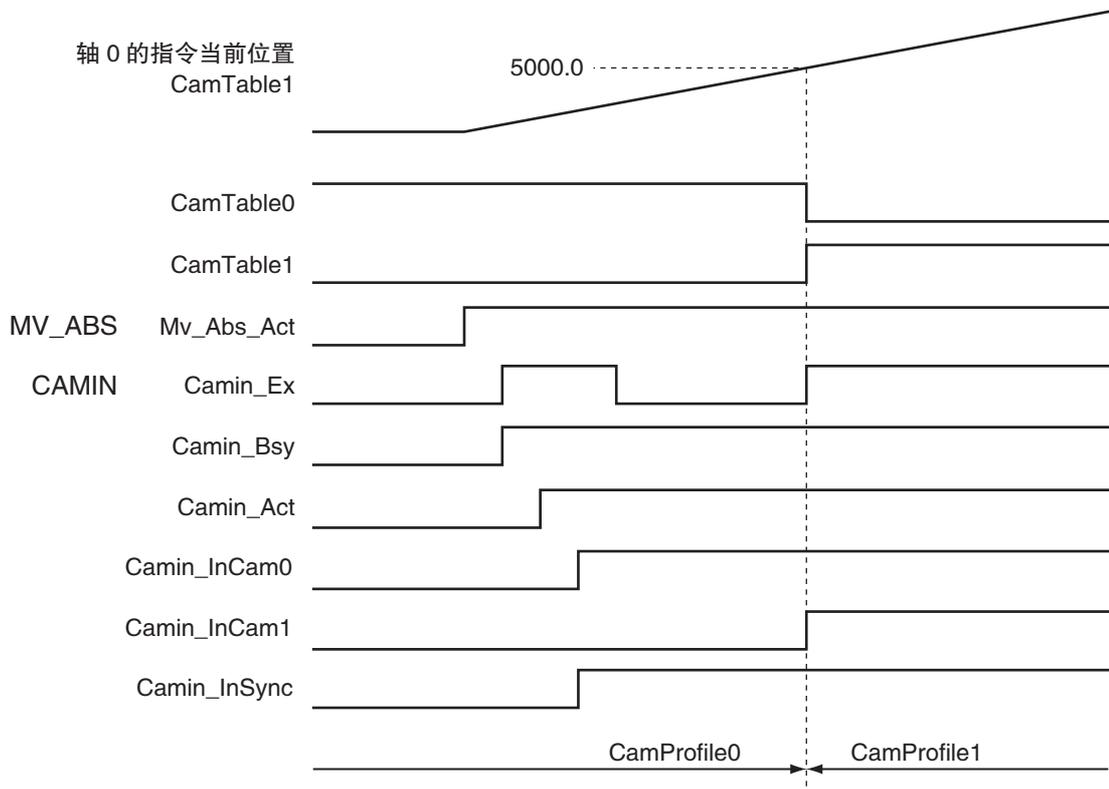
\*1. 数组的元素数 ARRAY[0..N] 由 Sysmac Studio 的凸轮编辑器功能设定。本示例程序中设定为 [0 ~ 109]。

## 时序图

### ● 梯形图程序时



### ● ST 程序时



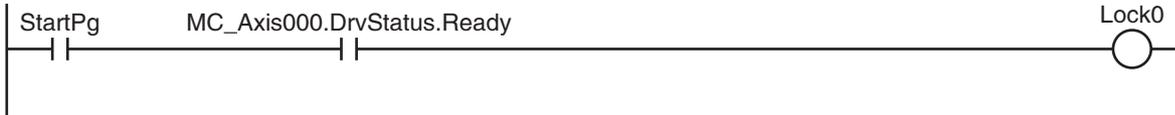
## 梯形图程序

切换 2 个凸轮表使用时，配置 2 个和 MC\_CamIn( 凸轮动作开始 ) 指令相同名称的实例。

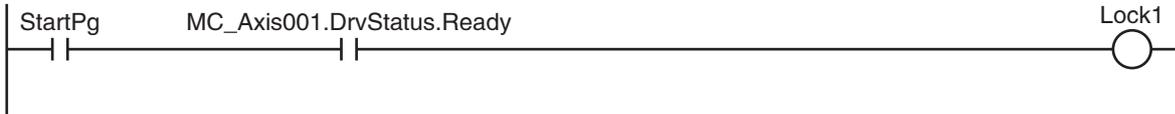
此时，将其它输出参数分配给各自的实例输出变量 “InCam( 凸轮动作中 )”。分配相同的输出参数时发生异常。

该示例程序中，不通过 JMP( 跳转 ) 指令同时执行 2 个实例。

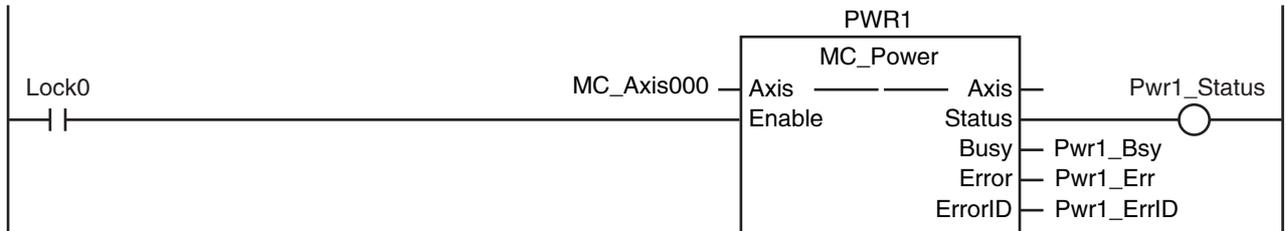
确认触点StartPg是否为TRUE，轴0的伺服驱动器是否处于伺服准备就绪状态



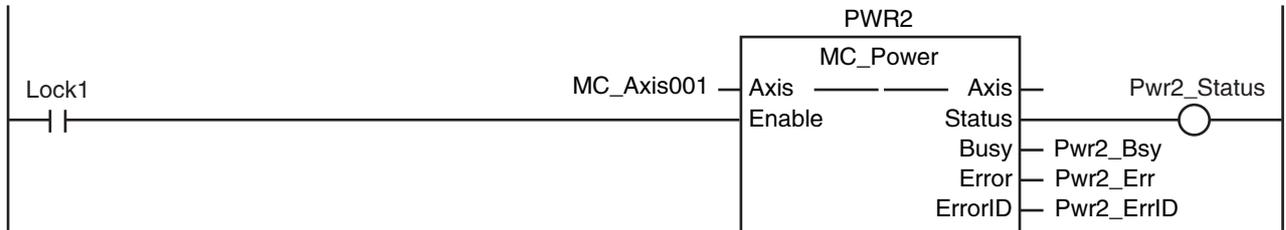
确认触点StartPg是否为TRUE，轴1的伺服驱动器是否处于伺服准备就绪状态



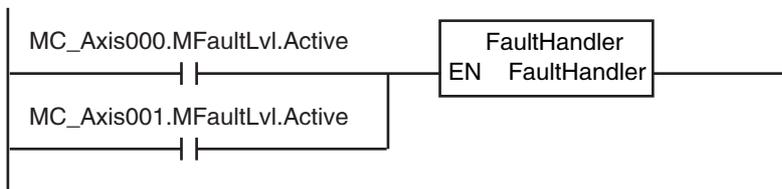
如果轴0的伺服驱动器处于伺服准备就绪状态，则将轴0伺服ON



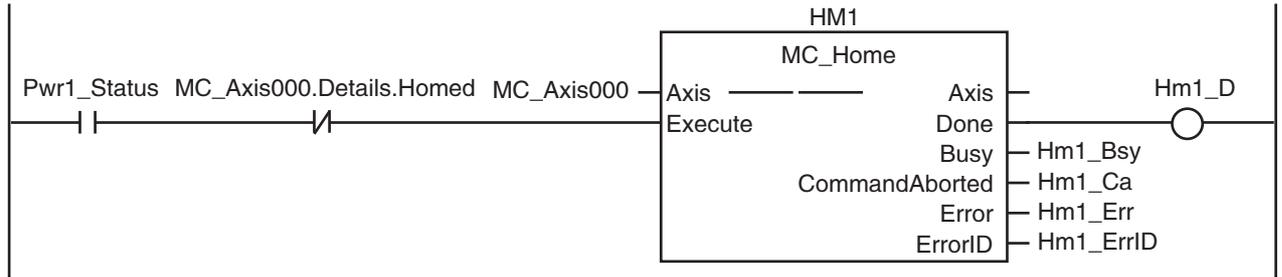
如果轴1的伺服驱动器处于伺服准备就绪状态，则将轴1伺服ON



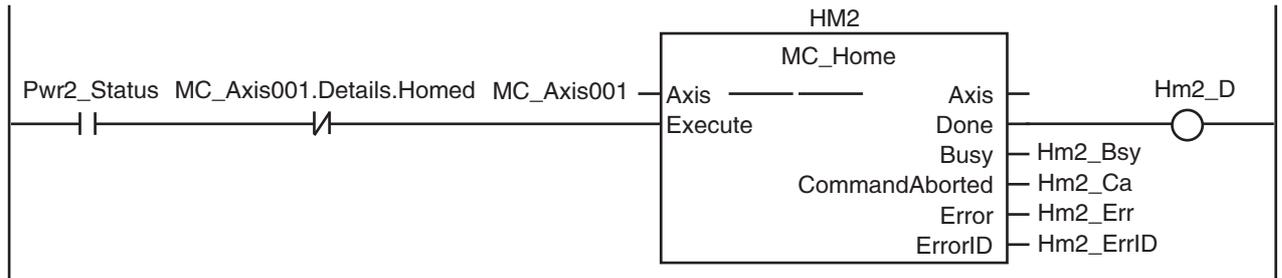
轴0或轴1发生轻度故障等级的异常时，执行异常处理的FaultHandler  
请根据装置，对异常处理的内容进行编程



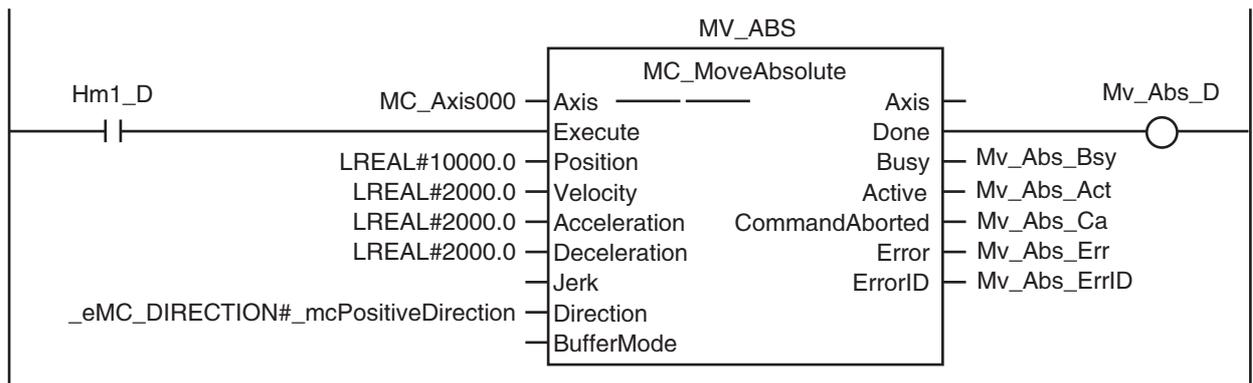
轴0处于伺服ON状态，且原点未确定时，执行原点复位



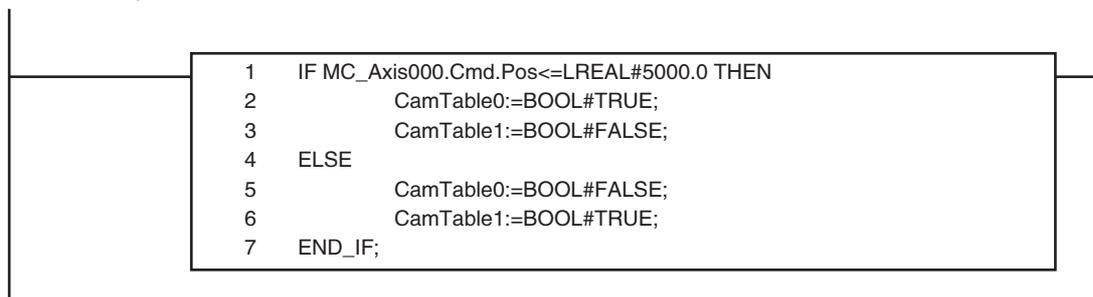
轴1处于伺服ON状态，且原点未确定时，执行原点复位



如果轴0的原点复位完成，则执行绝对值定位指令(MC\_MoveAbsolute)

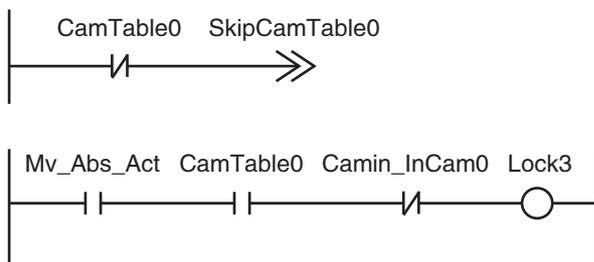


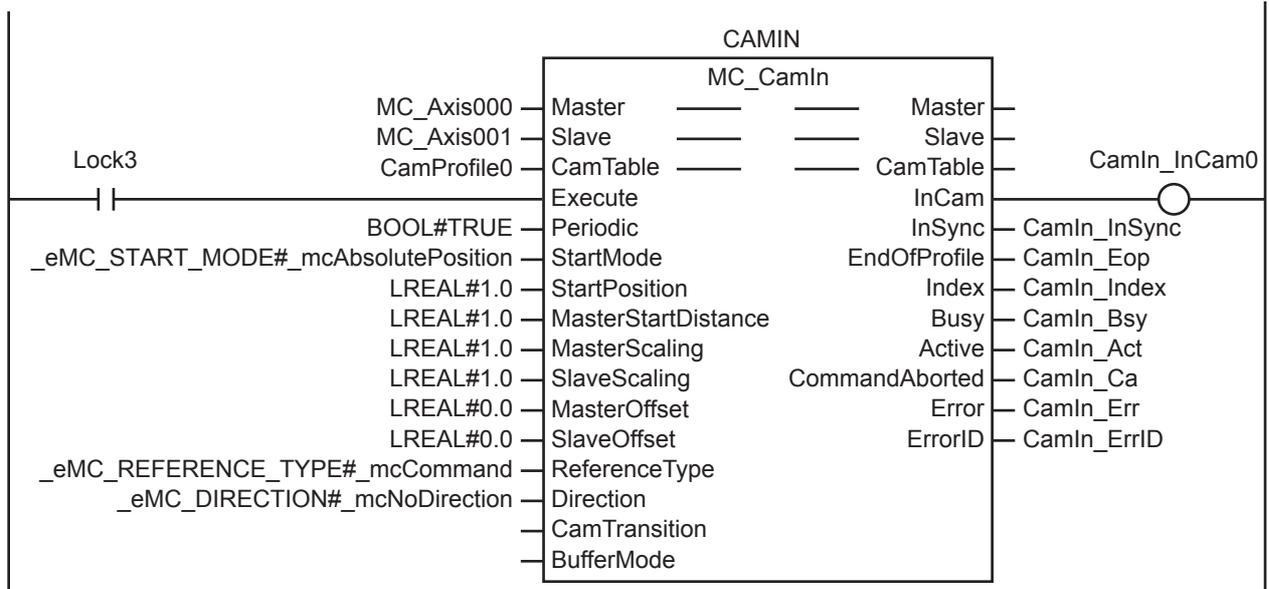
轴0的指令位置不超过5000时，设定CamTable0=TRUE、CamTable1=FALSE  
超过5000时，设定CamTable0=FALSE、CamTable1=TRUE



执行绝对值定位过程中CamTable0=TRUE时，将CamProfile0用于凸轮表  
执行实例

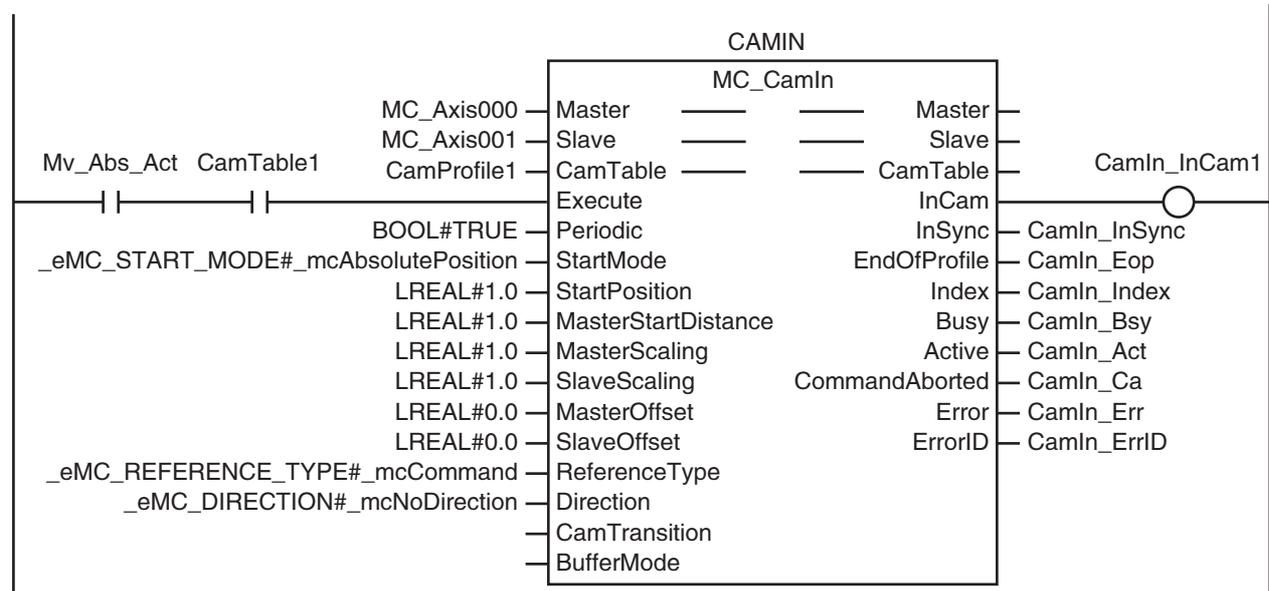
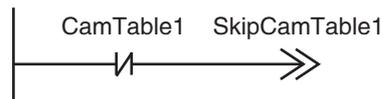
InCam=TRUE时，将Execute设定为FALSE



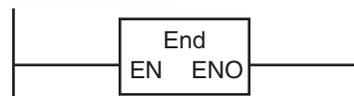


执行绝对值定位过程中CamTable1=TRUE时，将CamProfile1用于凸轮表  
执行实例

SkipCamTable0



SkipCamTable1



## ST 程序

```

// 绝对值定位和凸轮动作开始的输入参数未设定时，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;
    // 选择凸轮表
    CamTable0       := BOOL#TRUE;
    CamTable1       := BOOL#FALSE;

    // 使输入参数的设定完成标志生效
    InitFlag := TRUE;
END_IF;

// 如果触点 StartPg 为 TRUE，且过程数据通信处于正常执行状态，则使轴 0 变为伺服 ON 状态
IF (StartPg=TRUE)
AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr1_En:=TRUE;
ELSE
    Pwr1_En:=FALSE;
END_IF;

// 如果触点 StartPg 为 TRUE，且过程数据通信处于正常执行状态，则使轴 1 变为伺服 ON 状态
IF (StartPg=TRUE)
AND (MC_Axis001.DrvStatus.Ready=TRUE) 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，执行异常处理的 FaultHandler
// 请根据装置，对异常处理的内容进行编程
IF (MC_Axis000.MFaultLvl.Active=TRUE)
OR (MC_Axis001.MFaultLvl.Active=TRUE) THEN
    FaultHandler();
END_IF;

// 轴 0 处于伺服 ON 状态，且原点未确定时，执行原点复位
IF (Pwr1_S=TRUE)
AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm1_Ex:=TRUE;
END_IF;

// 轴 1 处于伺服 ON 状态，且原点未确定时，执行原点复位

```

```

IF (Pwr2_S=TRUE)
AND (MC_Axis001.Details.Homed=FALSE) THEN
    Hm2_Ex:=TRUE;
END_IF;

// 如果轴 0 的原点复位完成，则执行绝对值定位
IF Hm1_D=TRUE THEN
    Mv_Abs_Ex := TRUE;
END_IF;

// 轴 0 的指令位置不超过 5000 时，设定 CamTable0=TRUE、CamTable1=FALSE
// 超过 5000 时，设定 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;

// 执行绝对值定位过程中 CamTable0=TRUE 时，将 CamProfile0 用于凸轮表
// 执行实例
//InCam=TRUE 时，将 Execute 设定为 FALSE
IF (Mv_Abs_Act=TRUE)
AND (CamTable0=TRUE)
AND (Camin_InCam0=FALSE) THEN
    Camin_Ex      := TRUE;
ELSE
    Camin_Ex      := FALSE;
END_IF;

// 执行绝对值定位过程中 CamTable1=TRUE 时，将 CamProfile1 用于凸轮表
// 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

```

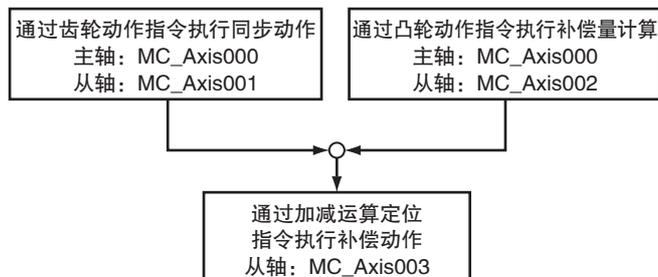
```
HM2(  
    Axis                := MC_Axis001,  
    Execute             := Hm2_Ex,  
    Done                => Hm2_D,  
    Busy                => Hm2_Bsy,  
    CommandAborted     => Hm2_Ca,  
    Error               => Hm2_Err,  
    ErrorID            => Hm2_ErrID  
);  
  
//MC_MoveAbsolute  
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 使用凸轮曲线补偿同步位置

使用凸轮曲线，对执行齿轮动作的从轴进行补偿。

执行齿轮动作的从轴为虚拟伺服轴的 MC\_Axis001、执行凸轮动作的从轴为虚拟伺服轴的 MC\_Axis002。通过 MC\_CombineAxes 对此类从轴进行加法运算，并将其结果输出至伺服轴的 MC\_Axis003。主轴为伺服轴的 MC\_Axis000。

处理流程如下所示。



### ● 设定轴种类

通过各轴的轴参数设定轴种类如下。

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

## 用于程序的主要变量

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 为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 为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	分配给 MC_Power 指令的实例 PWR1 的输出 Status 的变量 伺服进入 ON 状态时变为 TRUE
Pwr4_Status	BOOL	FALSE	分配给 MC_Power 指令的实例 PWR4 的输出 Status 的变量 伺服进入 ON 状态时变为 TRUE
Vel_InVel	BOOL	FALSE	轴 0 的 MC_MoveVelocity 到达目标速度时变为 TRUE
StartPg	BOOL	FALSE	如果该变量为 TRUE，且 EtherCAT 通信的过程数据通信处于正常执行状态，则变为伺服 ON 状态

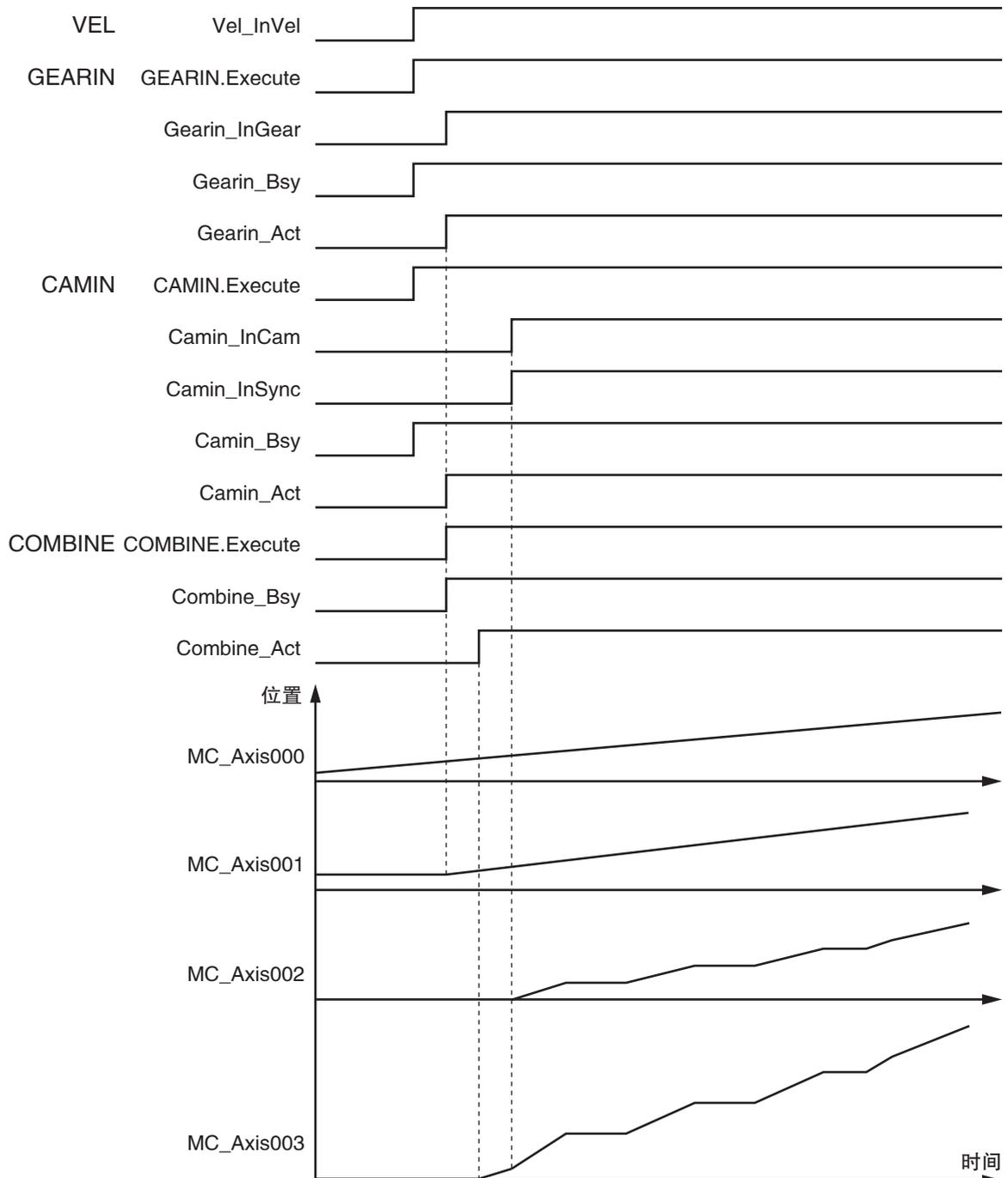
名称	数据类型	初始值	注释
Gearin_Ex	BOOL	FALSE	用于启动 MC_GearIn( 齿轮动作开始 ) 指令的变量 *2
Camin_Ex	BOOL	FALSE	用于启动 MC_CamIn( 凸轮动作开始 ) 指令的变量 *2
Combine_Ex	BOOL	FALSE	用于启动 MC_CombineAxes( 加减运算定位 ) 指令的变量 *2

\*1. 数组的元素数 ARRAY[0..N] 由 Sysmac Studio 的凸轮编辑器功能设定。本示例程序中设定为 [0 ~ 109]。

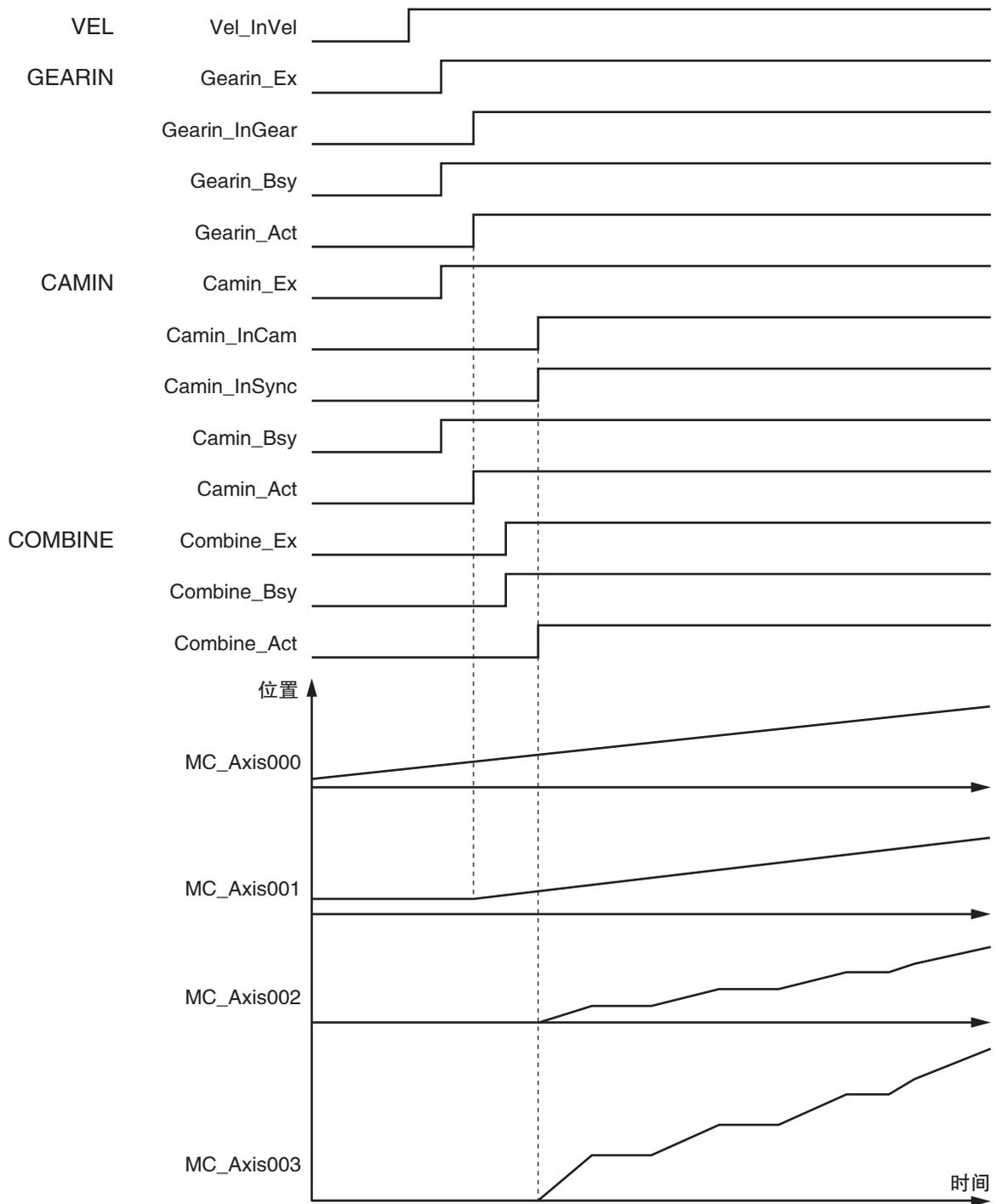
\*2. 用于 ST 程序。

## 时序图

### ● 梯形图程序时



## ● ST 程序时

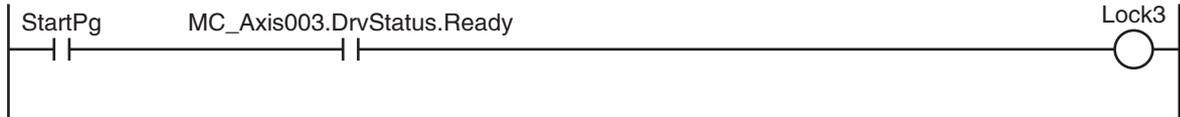


## 梯形图程序

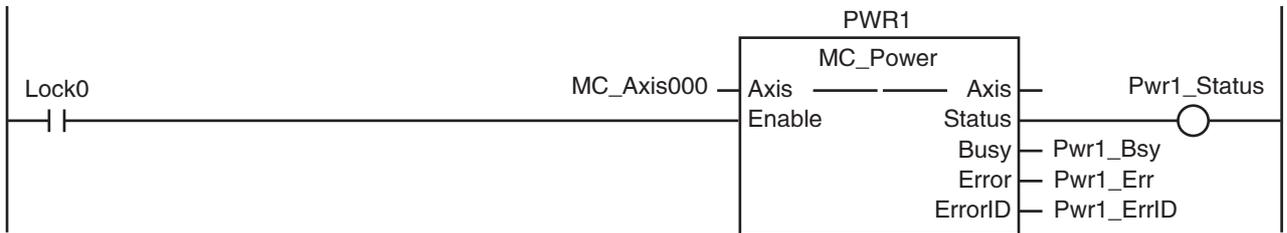
确认触点StartPg是否为TRUE，轴0的伺服驱动器是否处于伺服准备就绪状态



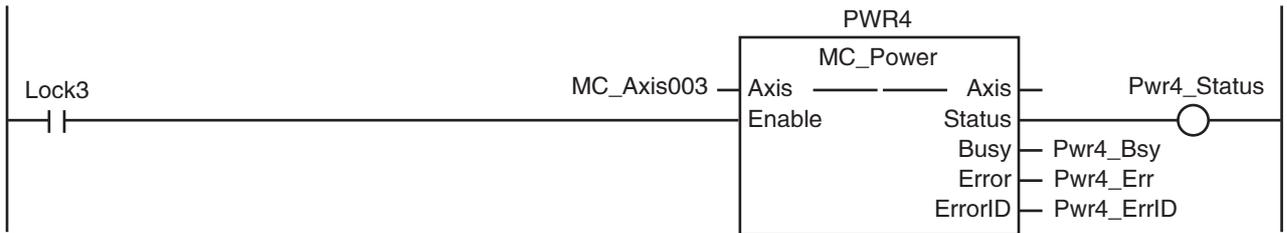
确认触点StartPg是否为TRUE，轴3的伺服驱动器是否处于伺服准备就绪状态



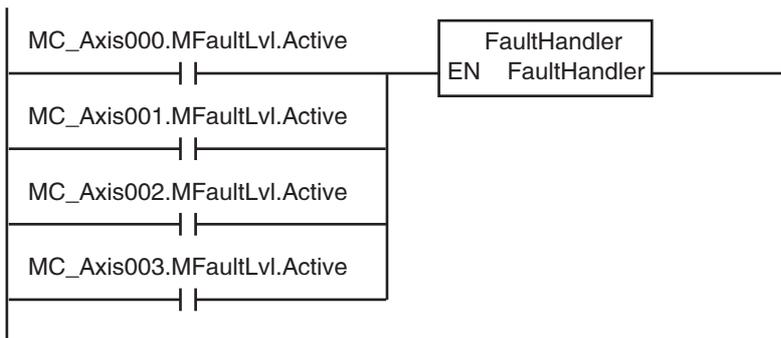
如果轴0的伺服驱动器处于伺服准备就绪状态，则将轴0伺服ON



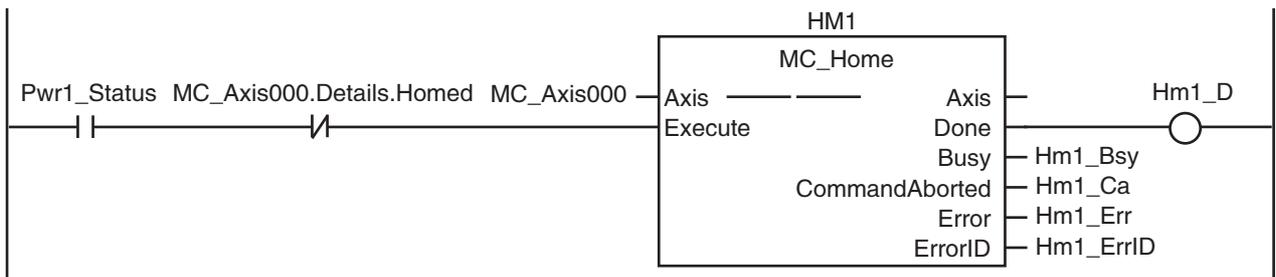
如果轴3的伺服驱动器处于伺服准备就绪状态，则将轴3伺服ON



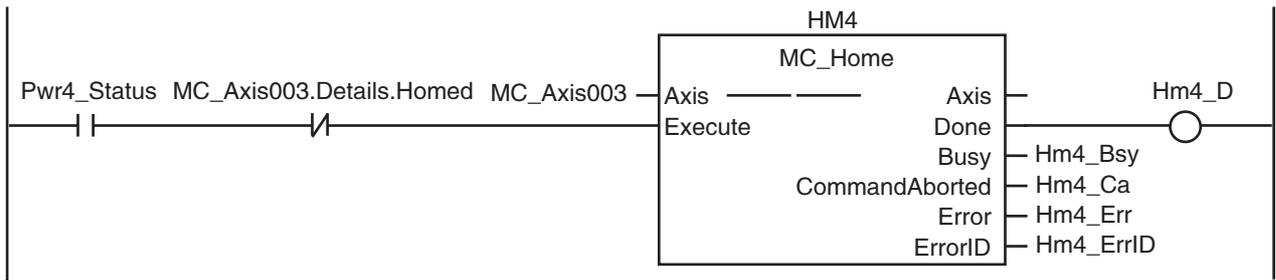
轴组或构成轴发生轻度故障等级的异常时，执行异常处理的FaultHandler  
请根据装置，对异常处理的内容进行编程



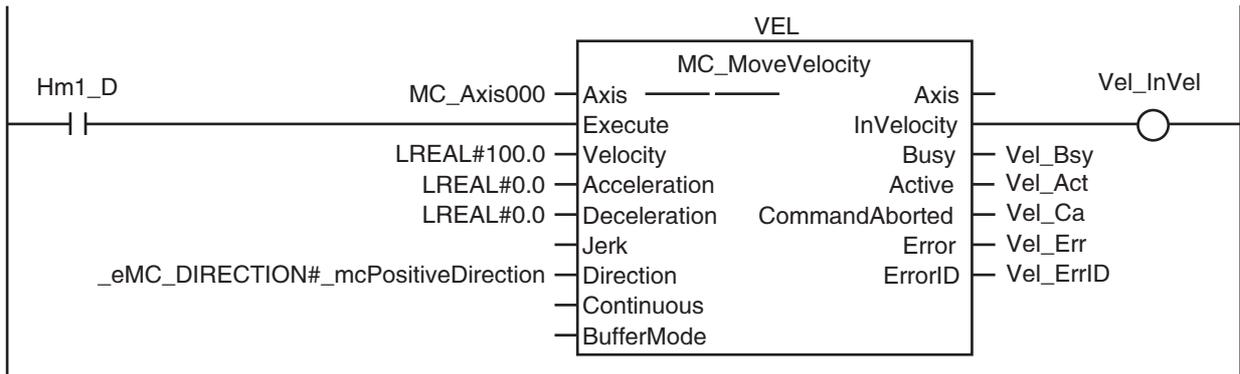
轴0处于伺服ON状态，且原点未确定时，执行原点复位



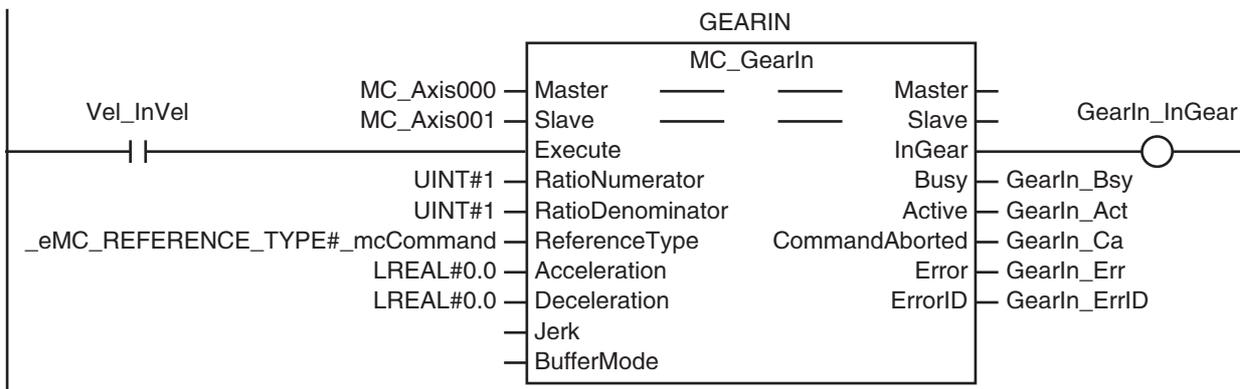
轴3处于伺服ON状态，且原点未确定时，执行原点复位



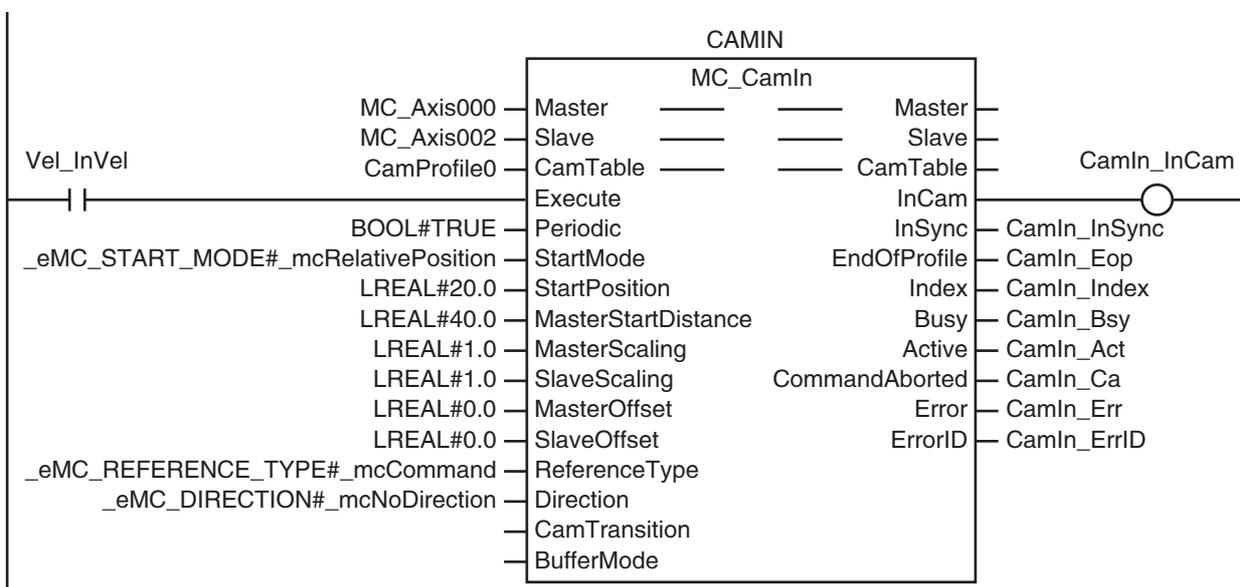
如果轴0的原点复位完成，则执行速度控制



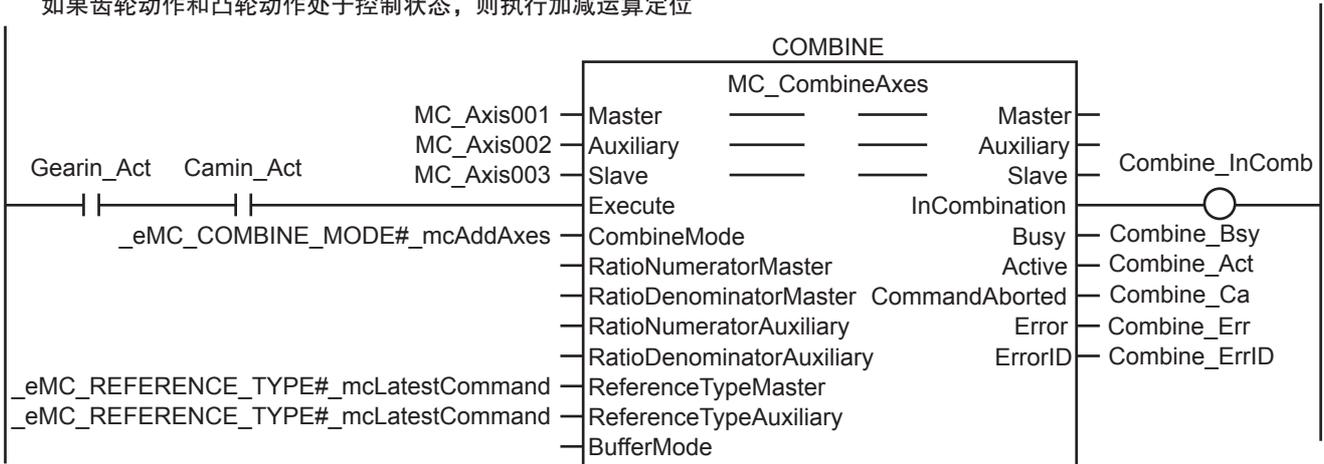
如果轴0到达目标速度，则执行齿轮动作



如果轴0到达目标速度，则执行凸轮动作



如果齿轮动作和凸轮动作处于控制状态，则执行加减运算定位



## ST 程序

```
// 动作指令的输入参数未设定时，设定目标值等
IF InitFlag=FALSE THEN
    // 给速度控制 (MC_MoveVelocity) 的输入参数设定数值
    Vel_Vel      := LREAL#100.0;
    Vel_Acc      := LREAL#0.0;
    Vel_Dec      := LREAL#0.0;
    Vel_Dir      := _eMC_DIRECTION#_mcPositiveDirection;

    // 给凸轮动作开始 (MC_CamIn) 的输入参数设定数值
    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;

    // 给齿轮动作开始 (MC_GearIn) 的输入参数设定数值
    Gearin_RatN  := UINT#1;
    Gearin_RatD  := UINT#1;
    Gearin_RefTyp := _eMC_REFERENCE_TYPE#_mcCommand;
    Gearin_Acc   := LREAL#0.0;
    Gearin_Dec   := LREAL#0.0;

    // 给加减运算定位 (MC_CombineAxes) 的输入参数设定数值
    Combine_Cm := _eMC_COMBINE_MODE#_mcAddAxes;
    Combine_RefMas := _eMC_REFERENCE_TYPE#_mcLatestCommand;
    Combine_RefAux := _eMC_REFERENCE_TYPE#_mcLatestCommand;
    // 使输入参数的设定完成标志生效
    InitFlag := TRUE;
END_IF;

// 如果触点 StartPg 为 TRUE，且伺服驱动器处于伺服准备就绪状态，则将轴 0 伺服 ON。
IF (StartPg=TRUE)
AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr1_En:=TRUE;
ELSE
    Pwr1_En:=FALSE;

```

```

END_IF;

// 如果触点 StartPg 为 TRUE，且伺服驱动器处于伺服准备就绪状态，则将轴 3 伺服 ON。
IF (StartPg=TRUE)
AND (MC_Axis003.DrvStatus.Ready=TRUE) 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，执行异常处理的 FaultHandler。
// 请根据装置，对异常处理的内容进行编程
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 处于伺服 ON 状态，且原点未确定时，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 处于伺服 ON 状态，且原点未确定时，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;

// 如果轴 0 的原点复位完成，则执行速度控制
IF Hm1_D=TRUE THEN
    Vel_Ex:=TRUE;
END_IF;

// 如果轴 0 到达目标速度，则执行齿轮动作
IF Vel_InVel=TRUE THEN
    Gearin_Ex := TRUE;
END_IF;

// 如果轴 0 到达目标速度，则执行凸轮动作
IF Vel_InVel=TRUE THEN
    Camin_Ex := TRUE;
END_IF;

// 如果齿轮动作和凸轮动作处于控制状态，则执行加减速定位
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 凸轮动作时补偿主轴的相位

执行凸轮动作的从轴与通过速度控制进行动作的主轴同步动作。如果将 StartOn 设定为 TRUE，则通过 MC\_Phasing( 主轴相对值相位补偿 ) 指令补偿主轴的相位。从轴对补偿的相位执行同步动作。

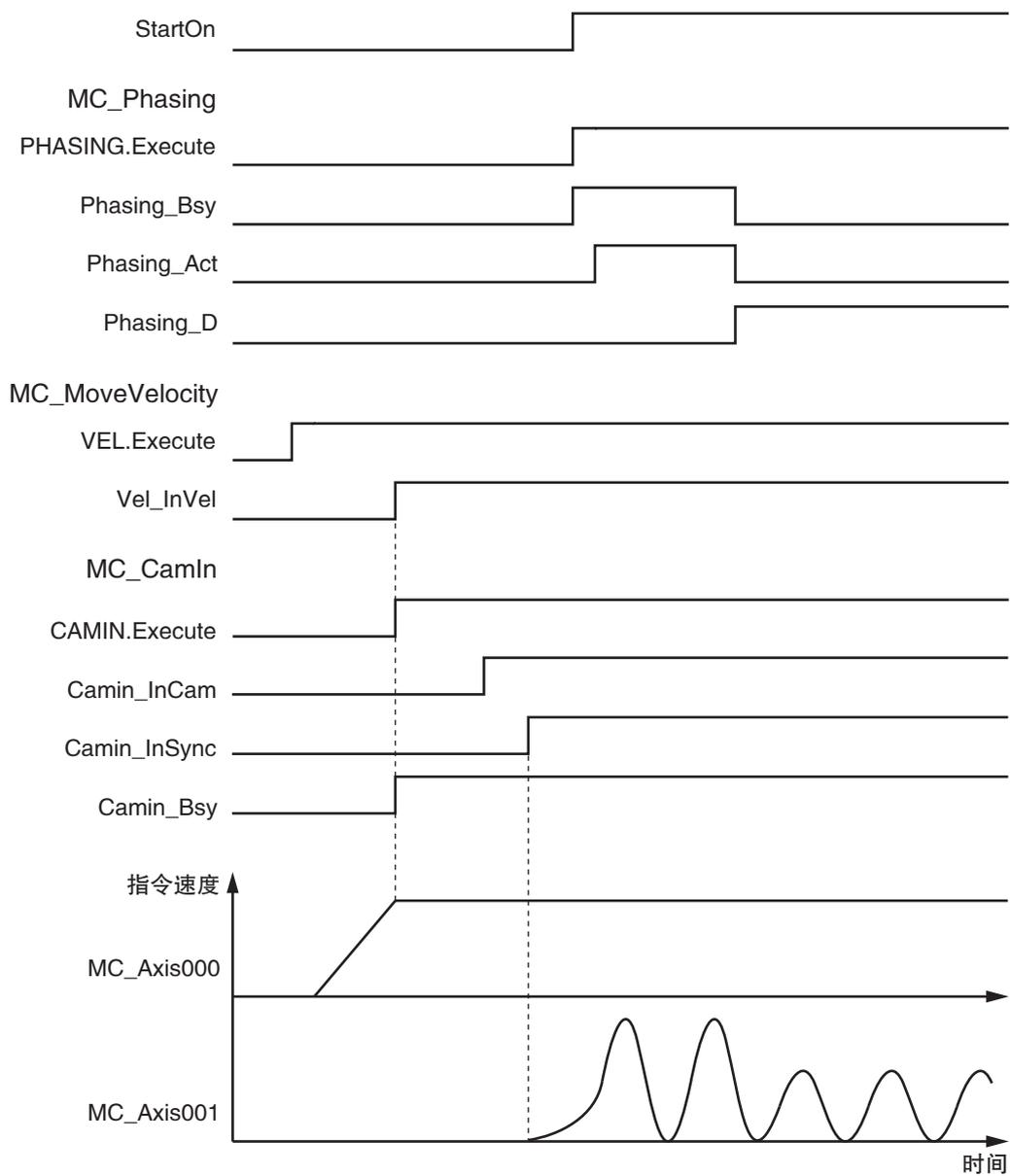
### 用于程序的主要变量

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 为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 为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	分配给 MC_Power 指令的实例 PWR1 的输出 Status 的变量 伺服进入 ON 状态时变为 TRUE
Pwr2_Status	BOOL	FALSE	分配给 MC_Power 指令的实例 PWR2 的输出 Status 的变量 伺服进入 ON 状态时变为 TRUE
StartOn	BOOL	FALSE	用于开始进行主轴相位补偿的变量
StartPg	BOOL	FALSE	如果该变量为 TRUE，且 EtherCAT 通信的过程 数据通信处于正常执行状态，则变为伺服 ON 状态
Camin_Ex	BOOL	FALSE	用于启动 MC_CamIn( 凸轮动作开始 ) 指令的变 量 用于 ST 程序
Vel_Ex	BOOL	FALSE	用于启动 MC_MoveVelocity( 速度控制 ) 指令的 变量 用于 ST 程序
Phasing_Ex	BOOL	FALSE	用于启动 MC_Phasing( 主轴相对值相位补偿 ) 指 令的变量 用于 ST 程序

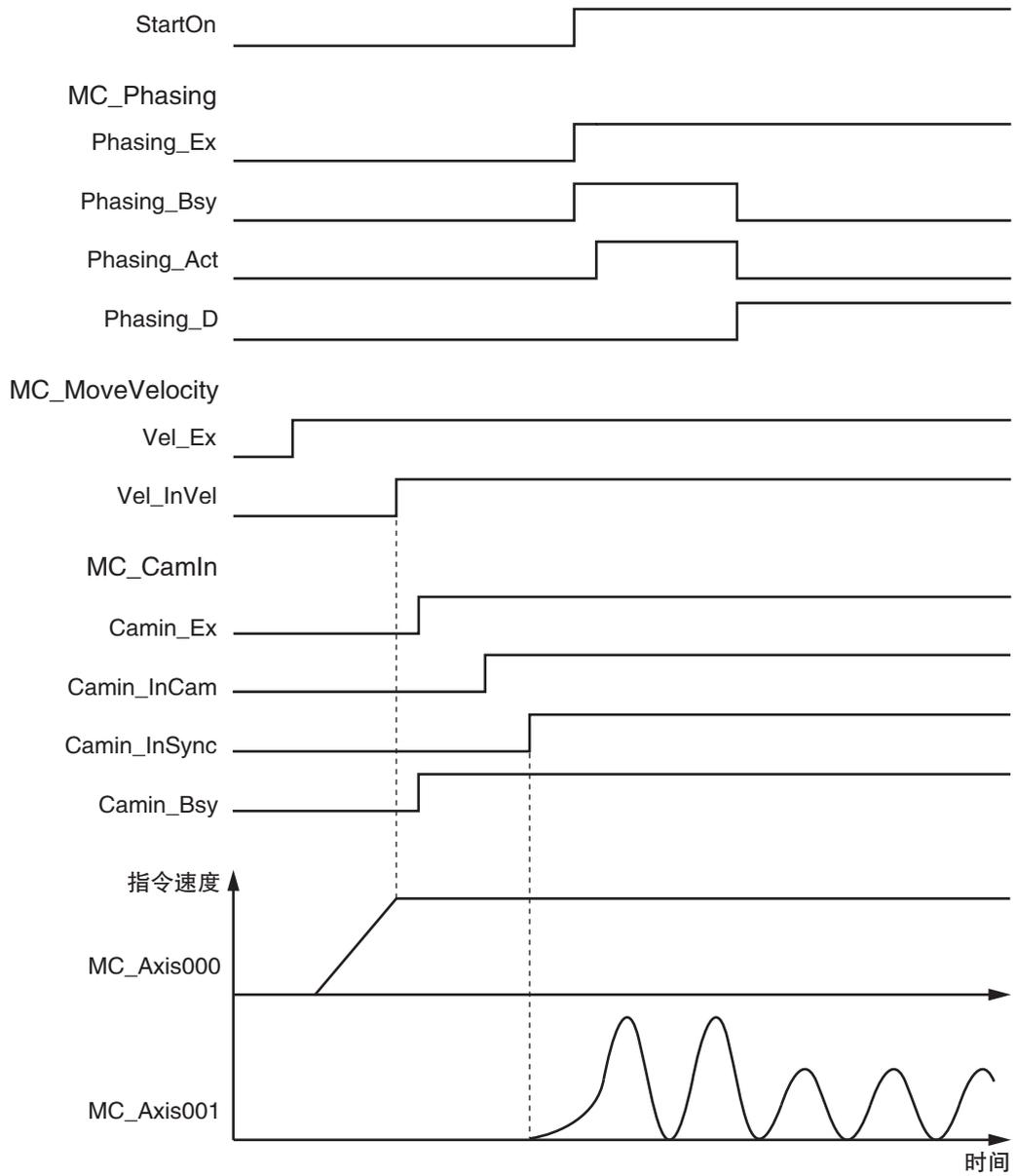
\*1. 数组的元素数 ARRAY[0..N] 由 Sysmac Studio 的凸轮编辑器功能设定。本示例程序中设定为 [0 ~ 360]。

## 时序图

### ● 梯形图程序时



● ST 程序时



## 梯形图程序

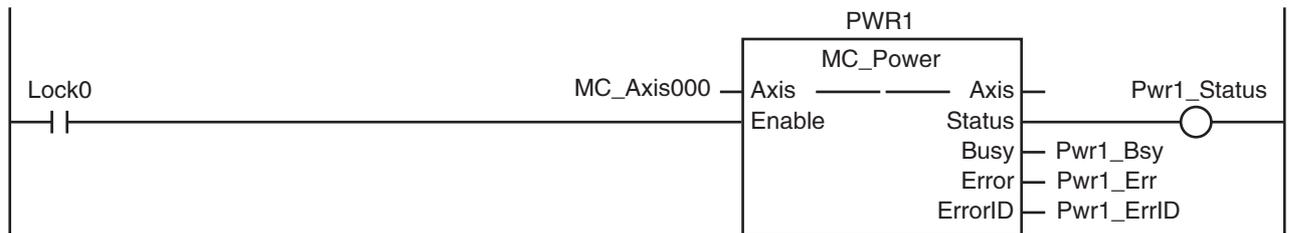
确认触点StartPg是否为TRUE，轴0的伺服驱动器是否处于伺服准备就绪状态



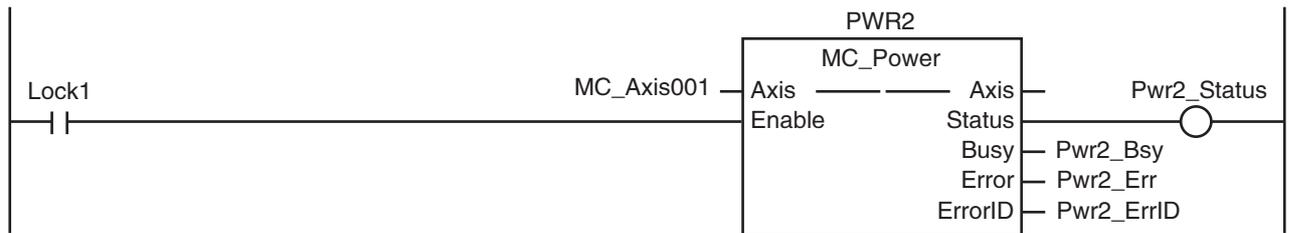
确认触点StartPg是否为TRUE，轴1的伺服驱动器是否处于伺服准备就绪状态



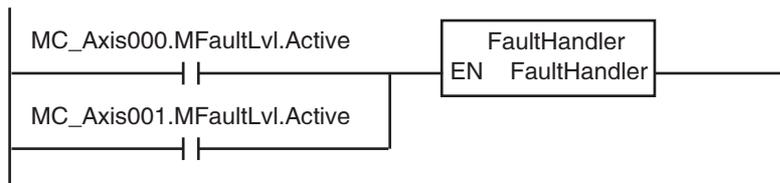
如果轴0的伺服驱动器处于伺服准备就绪状态，则将轴0伺服ON



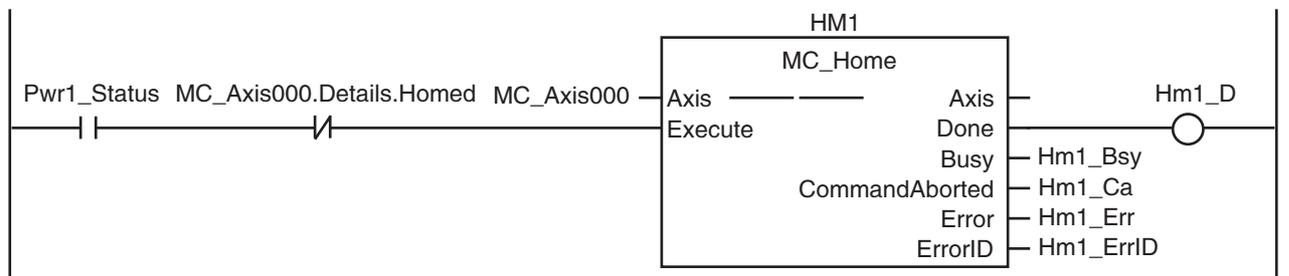
如果轴1的伺服驱动器处于伺服准备就绪状态，则将轴1伺服ON



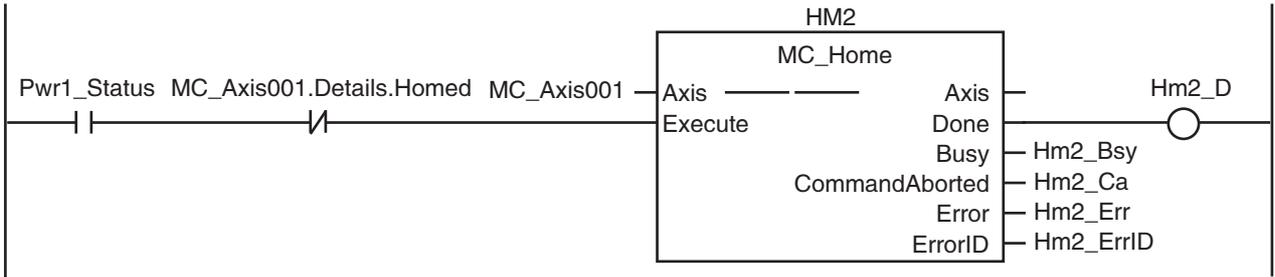
轴组或构成轴发生轻度故障等级的异常时，执行异常处理的FaultHandler  
请根据装置，对异常处理的内容进行编程



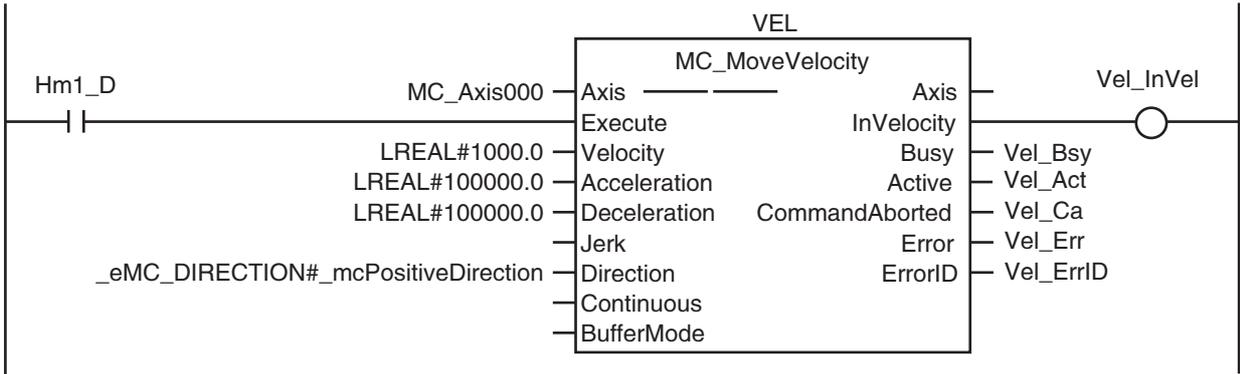
轴0处于伺服ON状态，且原点未确定时，执行原点复位



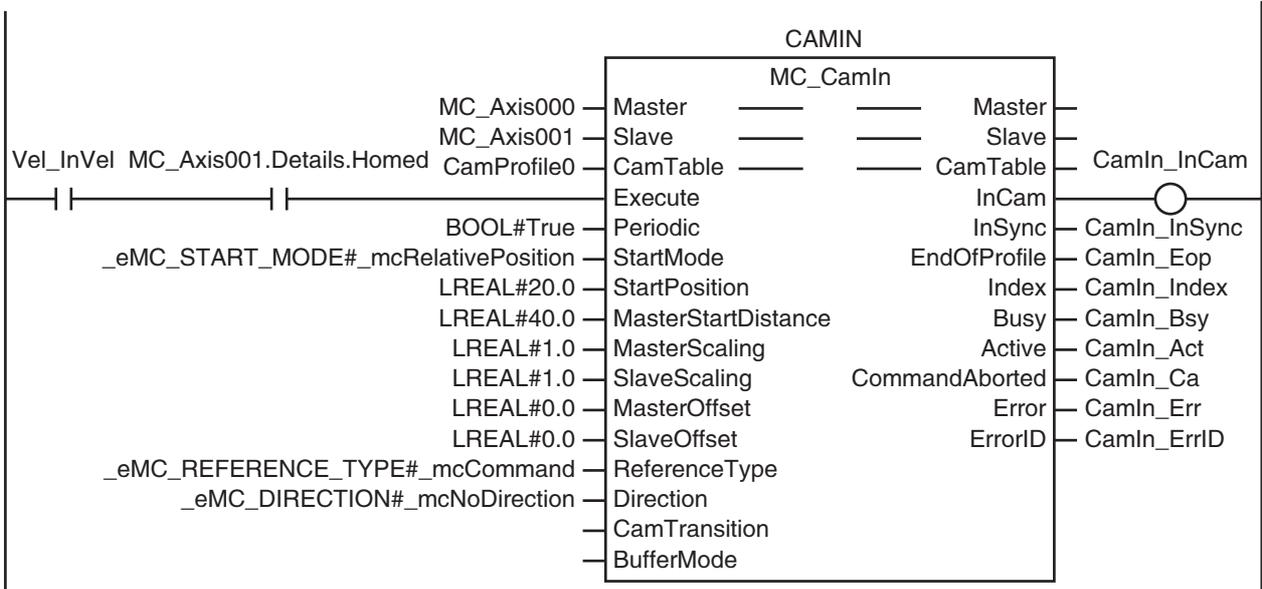
轴1处于伺服ON状态，且原点未确定时，执行原点复位



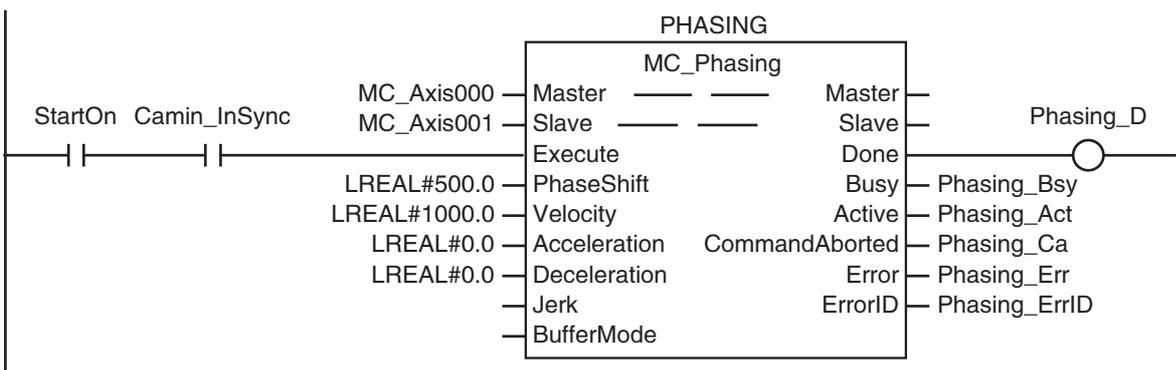
如果轴0的原点复位完成，则执行速度控制



如果轴0到达目标速度，则执行凸轮动作



如果StartOn为TRUE，且凸轮动作处于同步状态，则开始进行主轴的相位补偿



## ST 程序

```

// 动作指令的输入参数未设定时，设定目标值等
IF InitFlag=FALSE THEN
    // 给速度控制 (MC_MoveVelocity) 的输入参数设定数值
    Vel_Vel      := LREAL#1000.0;
    Vel_Acc      := LREAL#100000.0;
    Vel_Dec      := LREAL#100000.0;
    Vel_Dir      := _eMC_DIRECTION#_mcPositiveDirection;

    // 给主轴相对值相位补偿 (MC_Phasing) 的输入参数设定数值
    Phasing_Ps   := LREAL#500.0;
    Phasing_Vel  := LREAL#1000.0;
    Phasing_Acc  := LREAL#0.0;
    Phasing_Dec  := LREAL#0.0;

    // 给凸轮动作开始 (MC_CamIn) 的输入参数设定数值
    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;

    // 使输入参数的设定完成标志生效
    InitFlag := TRUE;
END_IF;

// 如果触点 StartPg 为 TRUE，且伺服驱动器处于伺服准备就绪状态，则将轴 0 伺服 ON。
IF (StartPg=TRUE)
AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr1_En:=TRUE;
ELSE
    Pwr1_En:=FALSE;
END_IF;

// 如果触点 StartPg 为 TRUE，且伺服驱动器处于伺服准备就绪状态，则将轴 1 伺服 ON。
IF (StartPg=TRUE)
AND (MC_Axis001.DrvStatus.Ready=TRUE) 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，执行异常处理的 FaultHandler
// 请根据装置，对异常处理的内容进行编程
IF (MC_Axis000.MFaultLvl.Active=TRUE)
OR (MC_Axis001.MFaultLvl.Active=TRUE) THEN
    FaultHandler();
END_IF;

// 轴 0 处于伺服 ON 状态，且原点未确定时，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 处于伺服 ON 状态，且原点未确定时，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;

// 如果轴 0 的原点复位完成，则执行速度控制
IF Hm1_D=TRUE THEN
    Vel_Ex:=TRUE;
END_IF;

// 如果轴 0 到达目标速度，且轴 1 确定了原点，则执行凸轮动作
IF (Vel_InVel=TRUE)
AND (MC_Axis001.Details.Homed=TRUE) THEN
    Camin_Ex := TRUE;
END_IF;

// 如果 StartOn 为 TRUE，且凸轮动作处于同步状态，则开始进行主轴的相位补偿
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速度控制中变更当前位置

变更通过速度控制执行动作时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

- 如果对处于动作状态的轴启动 MC\_SetPosition(当前位置变更)指令,则从启动到实际变更当前位置的移动距离会出现误差。
- 如果启动 MC\_SetPosition(当前位置变更)指令,则会出现原点未确定的情况。

## 轴参数设定

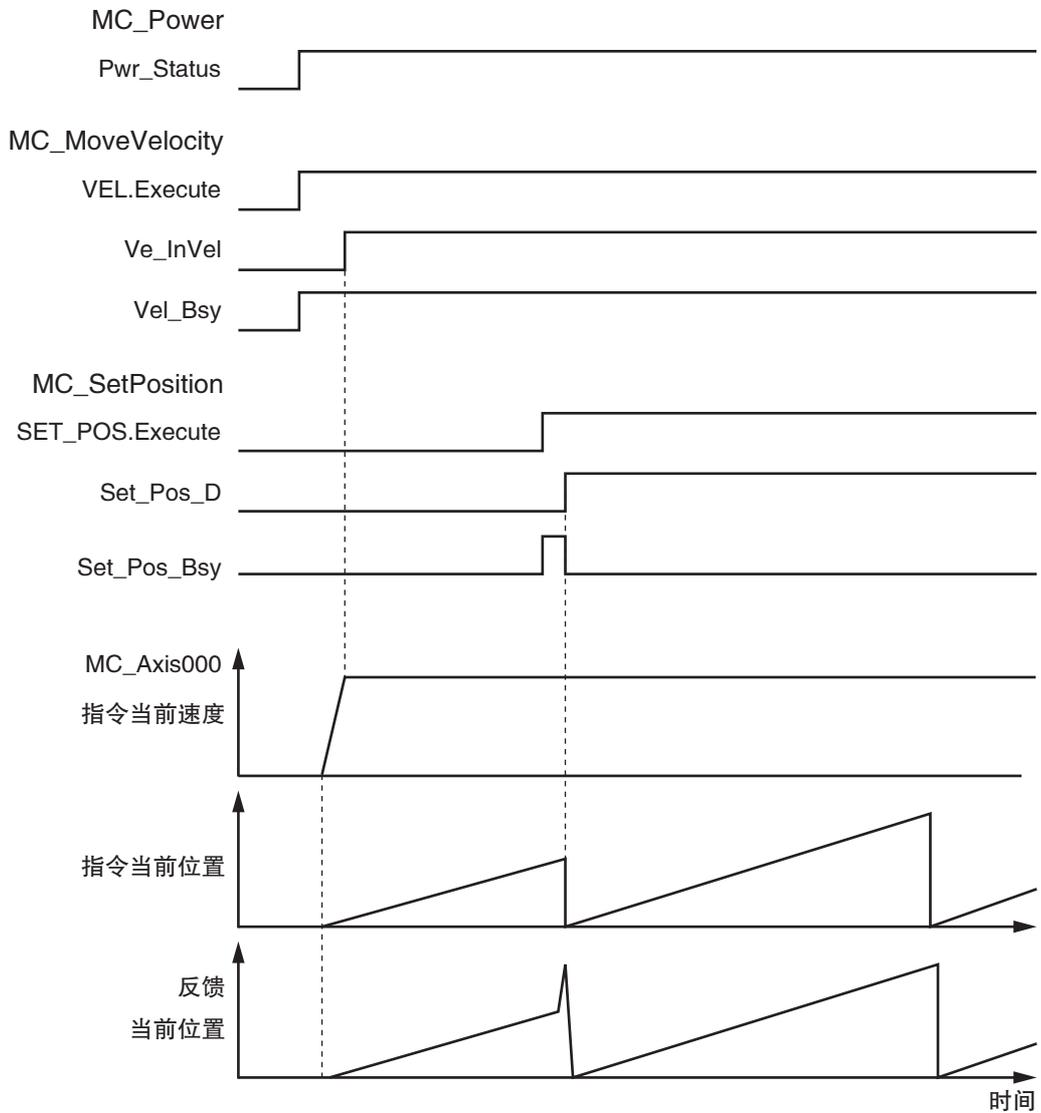
设定项目	设定内容	说明
计数模式	旋转模式	将位置的计数模式设定为旋转模式
环计数器上限设定值	360	将环计数器的上限值设定为“360”
环计数器下限设定值	0	将环计数器的下限值设定为“0”
原点复位动作	原点预设	通过原点预设确定原点

## 用于程序的主要变量

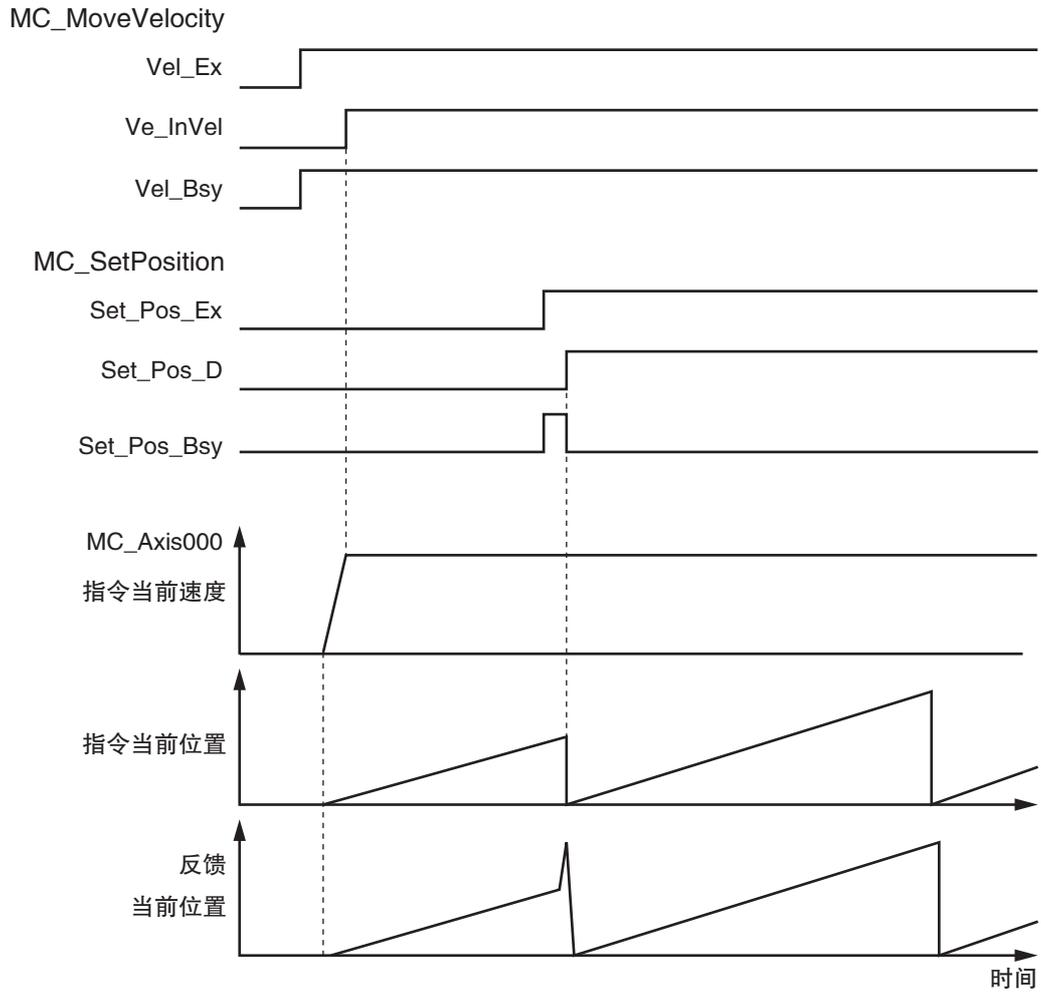
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	-	轴 0 的轴变量
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 0 发生轻度故障等级的异常时变为 TRUE
Pwr_Status	BOOL	FALSE	分配给 MC_Power 指令的实例 PWR 的输出 Status 的变量 伺服进入 ON 状态时变为 TRUE
StartSetPos	BOOL	FALSE	用于变更当前位置的外部按钮
StartPg	BOOL	FALSE	如果该变量为 TRUE,且 EtherCAT 通信的过程数据通信处于正常执行状态,则变为伺服 ON 状态
Vel_Ex	BOOL	FALSE	用于启动 MC_MoveVelocity(速度控制)指令的变量 用于 ST 程序
SetPos_Ex	BOOL	FALSE	用于启动 MC_SetPosition(当前位置变更)指令的变量 用于 ST 程序

## 时序图

### ● 梯形图程序时



● ST 程序时

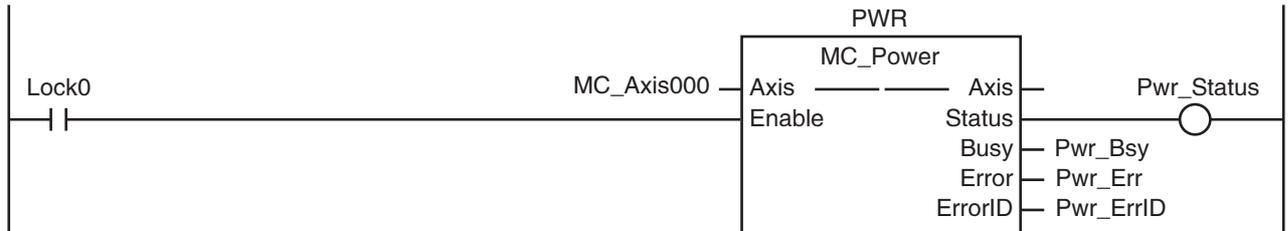


## 梯形图程序

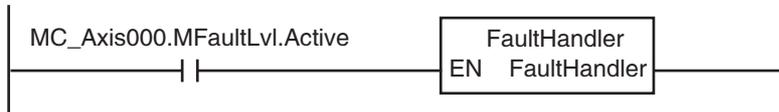
确认触点StartPg是否为TRUE，轴0的伺服驱动器是否处于伺服准备就绪状态



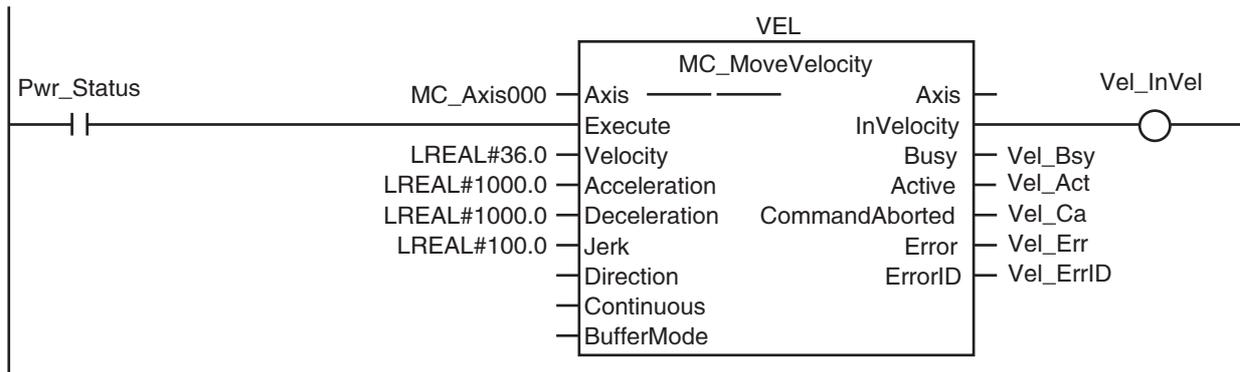
如果轴0的伺服驱动器处于伺服准备就绪状态，则将轴0伺服ON



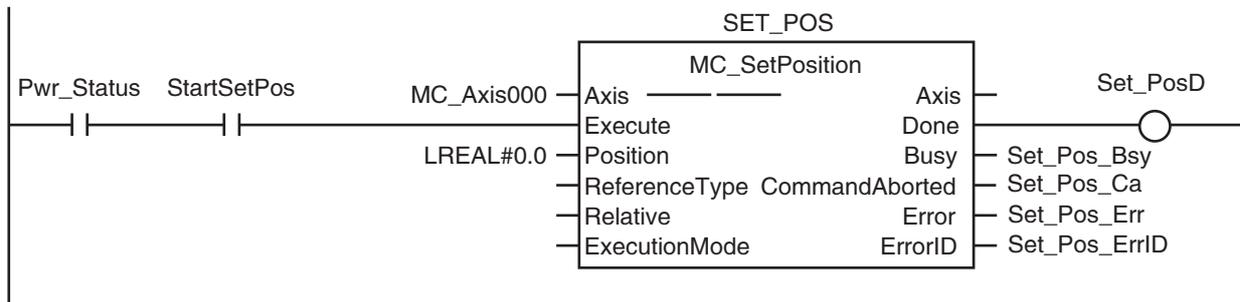
轴0发生轻度故障等级的异常时，执行异常处理的FaultHandler  
请根据装置，对异常处理的内容进行编程



如果轴0处于伺服ON状态，则执行速度控制



如果触点StartSetPos在伺服ON状态下变为TRUE，则执行当前位置变更



## ST 程序

```

// 指令的输入参数未设定时, 设定目标值等
IF InitFlag=FALSE THEN
    // 给速度控制 (MC_MoveVelocity) 的输入参数设定数值
    Vel_Vel      := LREAL#36.0;
    Vel_Acc      := LREAL#1000.0;
    Vel_Dec      := LREAL#1000.0;
    Vel_Jrk      := LREAL#100.0;

    // 给当前位置变更 (MC_SetPosition) 的输入参数设定数值
    Set_Pos_Pos  := LREAL#0.0;

    // 使输入参数的设定完成标志生效
    InitFlag:=TRUE;
END_IF;

// 如果触点 StartPg 为 TRUE, 且和轴 0 之间的过程数据通信处于正常执行状态, 则使轴 0 变为伺服 ON 状态
// 未执行过程数据通信时, 变为伺服 OFF 状态。
IF (StartPg=TRUE)
AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr_En:=TRUE;
ELSE
    Pwr_En:=FALSE;
END_IF;

// 轴 0 发生轻度故障等级的异常时, 执行异常处理的 FaultHandler
// 请根据装置, 对异常处理的内容进行编程
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
    FaultHandler();
END_IF;

// 如果轴 0 处于伺服 ON 状态, 则执行轴 0 的速度控制
IF Pwr_Status=TRUE THEN
    Vel_Ex:=TRUE;
END_IF;

// 如果触点 StartSetPos 在伺服 ON 状态下变为 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 变更凸轮数据变量和保存凸轮表

通过用户程序变更 Sysmac Studio 的凸轮编辑器创建的凸轮数据变量值。

相位为 0 度 ~ 180 度时将位移设定为 2 倍、181 度 ~ 360 度时将位移设定为 0.5 倍。

如果凸轮数据的变更完成，则通过 MC\_SaveCamTable(凸轮表保存)指令保存至 CPU 单元的非易失性存储器。如果完成了保存，则通过 MC\_CamIn(凸轮动作开始)指令开始凸轮动作。



### 使用注意事项

- 相位并非升序时，如果启动 MC\_CamIn(凸轮动作开始)指令，则会发生异常。  
该示例程序中未执行相位的升序检查。执行检查时，请执行 MC\_SetCamTableProperty(更新凸轮表属性)指令。
- CPU 单元非易失性存储器的写入次数受到限制。请仅在需要的情况下保存凸轮表。
- 如果在使用 MC\_SaveCamTable(凸轮表保存)指令进行保存前切断 CPU 单元的电，则通过用户程序变更的凸轮数据变量将返回变更前的内容。

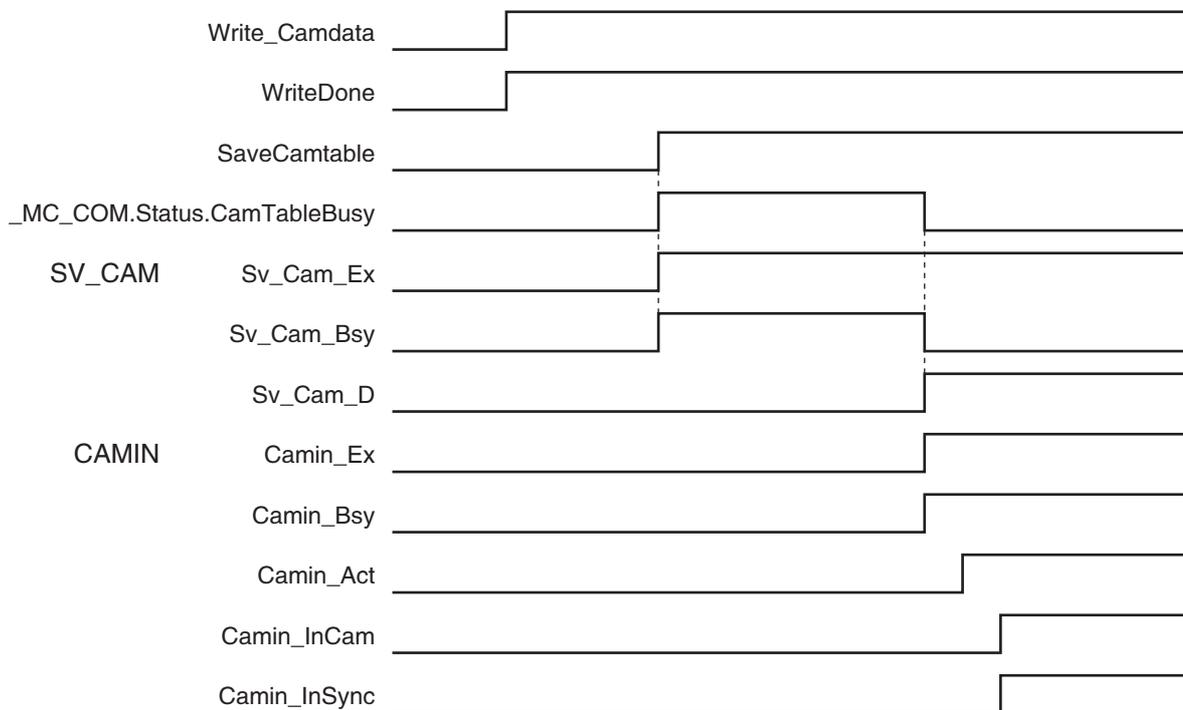
## 用于程序的主要变量

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	分配给 MC_Power 指令的实例 PWR1 的输出 Status 的变量 伺服进入 ON 状态时变为 TRUE
Pwr2_Status	BOOL	FALSE	分配给 MC_Power 指令的实例 PWR2 的输出 Status 的变量 伺服进入 ON 状态时变为 TRUE
StartPg	BOOL	FALSE	如果该变量为 TRUE，且 EtherCAT 通信的过程 数据通信处于正常执行状态，则变为伺服 ON 状态
WriteCamdata	BOOL	FALSE	用于启动凸轮数据变更的变量 开始编辑时请设定为 TRUE
WriteDone	BOOL	FALSE	用于表示凸轮数据变更完成的变量 变更完成时请设定为 TRUE
SaveCamtable	BOOL	FALSE	用于启动凸轮表保存的变量
_MC_COM.Status.CamTableBusy	BOOL	FALSE	表示凸轮表保存执行中的系统定义变量
Sv_Cam_Ex	BOOL	FALSE	用于启动 MC_SaveCamTable(凸轮表保存)指令 的变量
Camin_Ex	BOOL	FALSE	用于启动 MC_CamIn(凸轮动作开始)指令的变量 用于 ST 程序

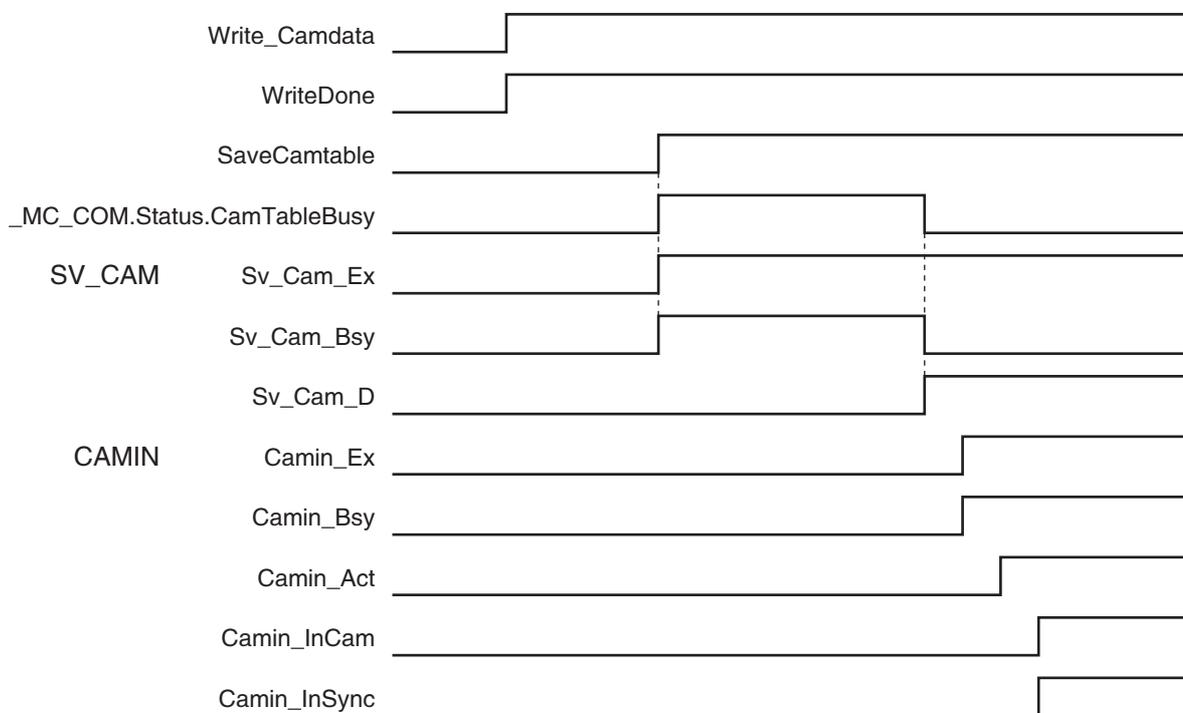
\*1. 数组的元素数 ARRAY[0..N] 由 Sysmac Studio 的凸轮编辑器功能设定。本示例程序中设定为 [0 ~ 360]。

## 时序图

### ● 梯形图程序时

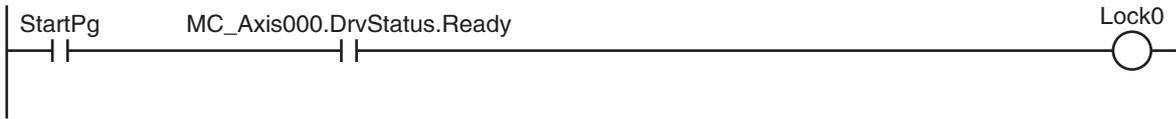


### ● ST 程序时

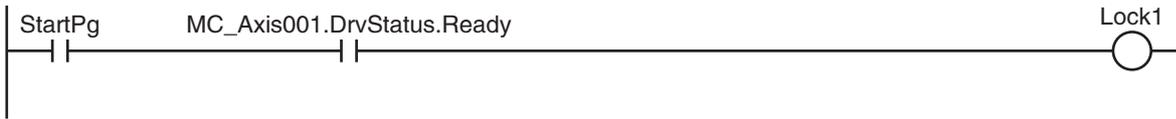


## 梯形图程序

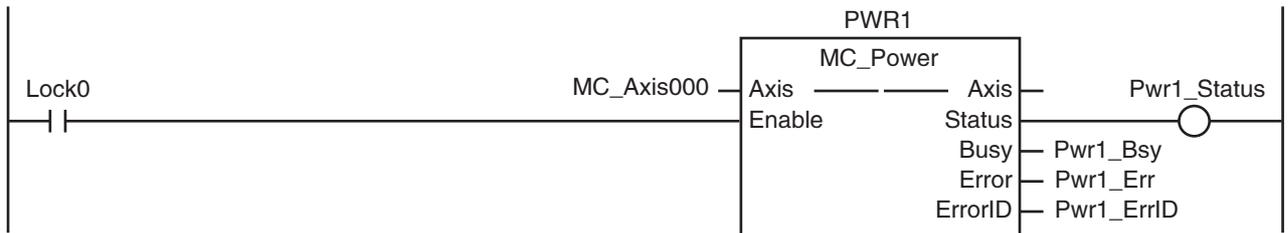
确认触点StartPg是否为TRUE，轴0的伺服驱动器是否处于伺服准备就绪状态



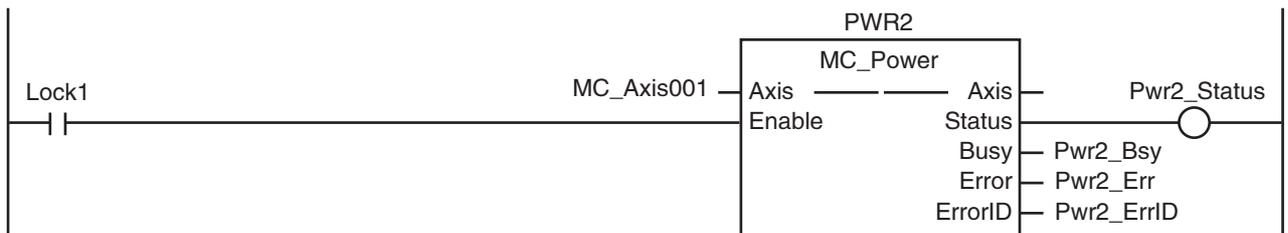
确认触点StartPg是否为TRUE，轴1的伺服驱动器是否处于伺服准备就绪状态



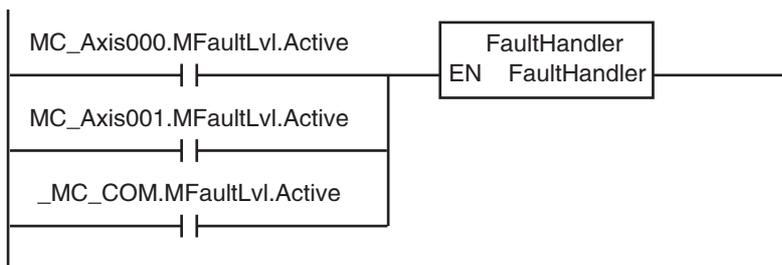
如果轴 0 的伺服驱动器处于伺服准备就绪状态，则将轴0伺服ON



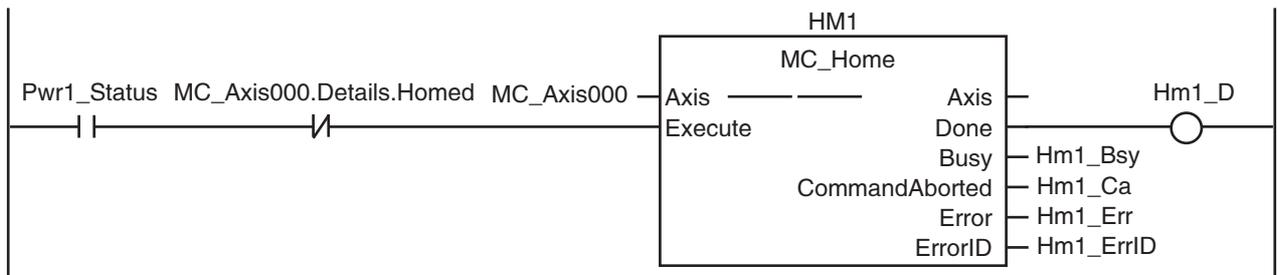
如果轴 1 的伺服驱动器处于伺服准备就绪状态，则将轴1伺服ON



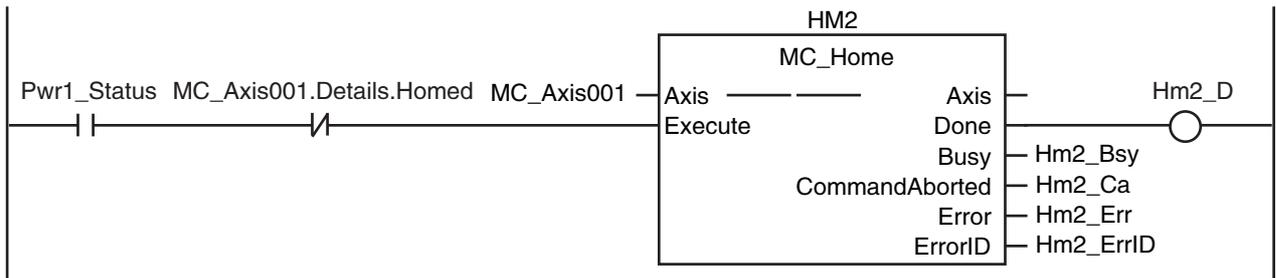
MC通用或轴发生轻度故障等级的异常时，执行异常处理的FaultHandler  
 请根据装置，对异常处理的内容进行编程



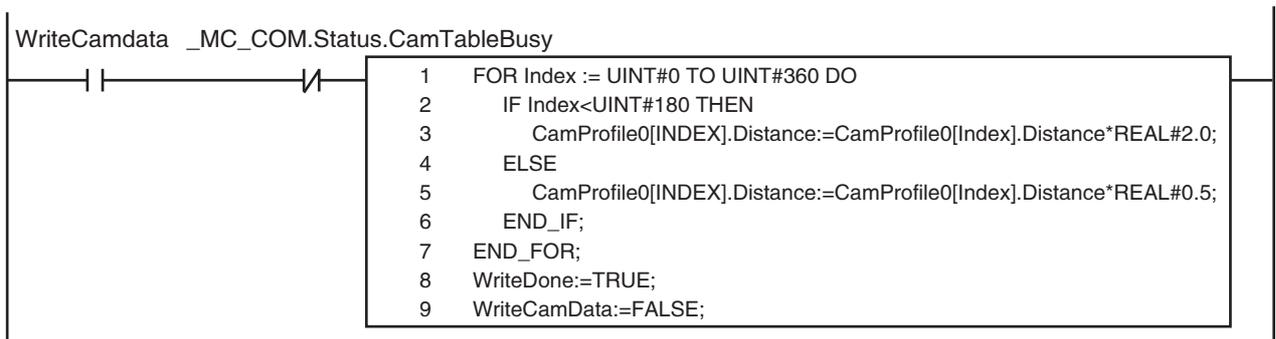
轴0处于伺服ON状态，且原点未确定时，执行原点复位



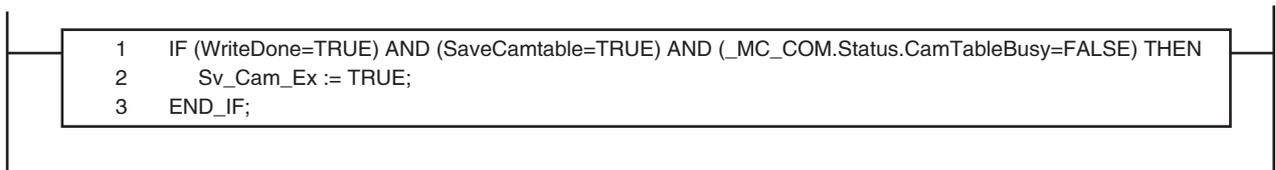
轴1处于伺服ON状态，且原点未确定时，执行原点复位



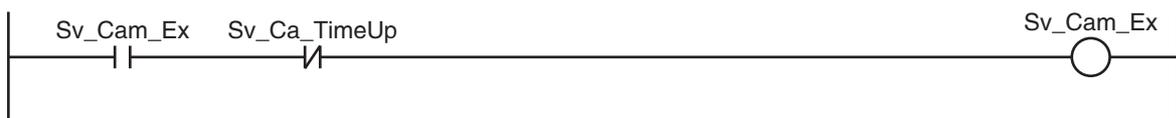
WriteCamData为TRUE，且凸轮表文件保存不处于执行状态时，变更凸轮数据变量值  
相位为0度~180度时将位移设定为2倍、181度~360度时将位移设定为0.5倍  
变更完成时，将WriteDone设定为TRUE



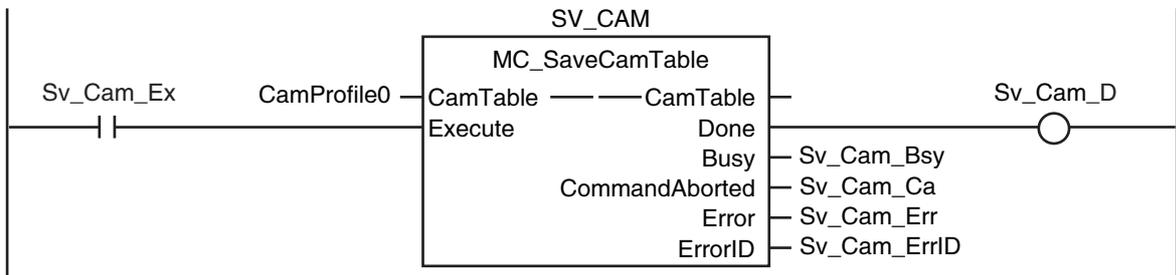
凸轮数据变量的变更完成，且SaveCamtable为TRUE、凸轮表文件保存不处于执行状态时，将Sv\_Cam\_Ex设定为TRUE



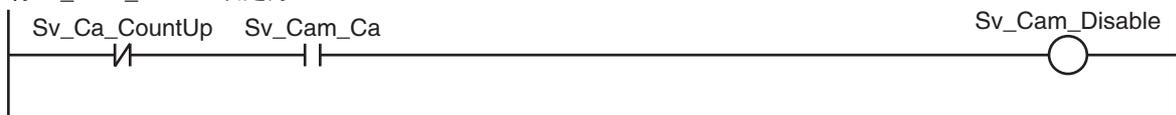
Sv\_Ca\_TimeUp为TRUE时，将Sv\_Cam\_Ex设定为FALSE  
如果Sv\_Cam\_Ex变为FALSE，则Sv\_Ca\_TimeUp变为FALSE，将Sv\_Cam\_Ex设定为TRUE  
重新执行MC\_SaveCamTable(凸轮表保存)指令



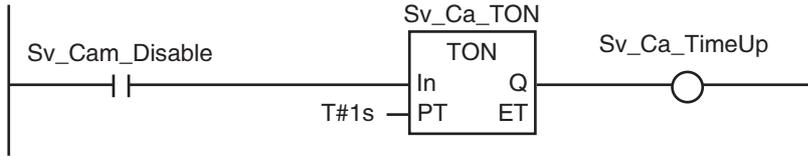
如果Sv\_Cam\_Ex为TRUE，则执行MC\_SaveCamTable(凸轮表保存)指令



Sv\_Ca\_CountUp为FALSE，且发生不可执行凸轮表保存的情况时，  
将Sv\_Cam\_Disable设定为TRUE



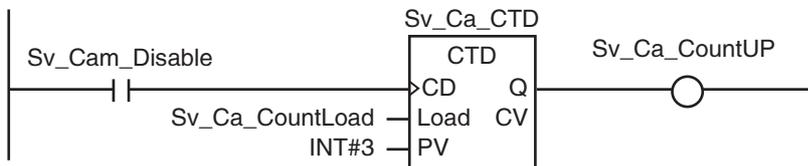
发生不可执行凸轮表保存的情况1秒后，将Sv\_Ca\_TimeUp设定为TRUE  
 如果Sv\_Ca\_TimeUp变为TRUE，则Sv\_Cam\_Ex变为FALSE



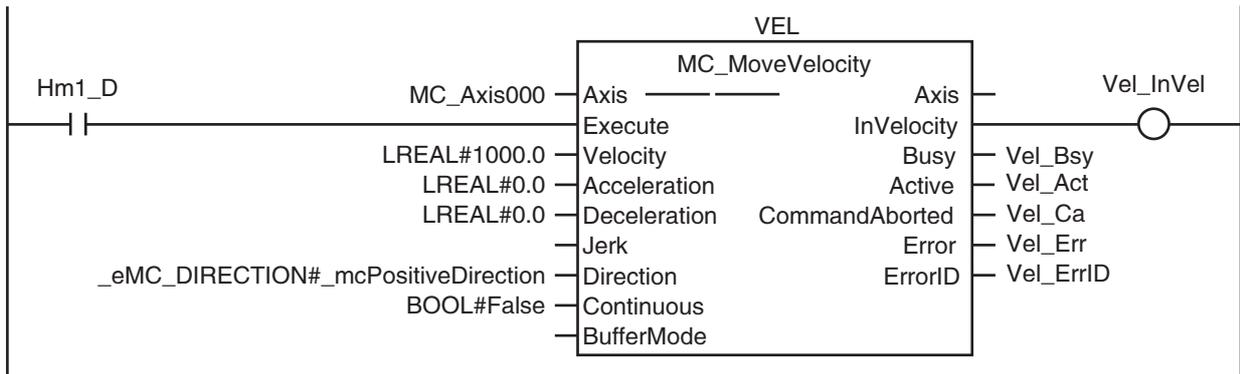
执行凸轮表保存时，在1个周期内将Sv\_Ca\_CountLoad设定为TRUE  
 如果Sv\_Ca\_CountLoad变为TRUE，则复位重试计数器



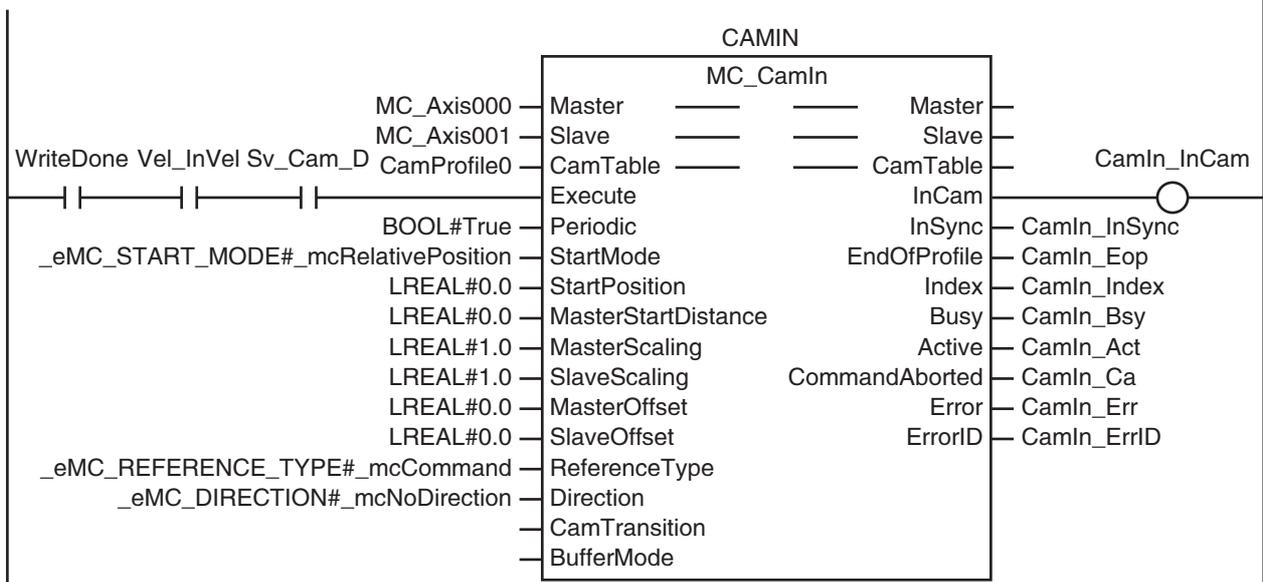
不可执行凸轮表保存的情况发生3次时，将Sv\_Ca\_CountUp设定为TRUE  
 如果Sv\_Ca\_CountUp变为TRUE，则将Sv\_Cam\_Disable设定为FALSE  
 结束MC\_SaveCamTable(凸轮表保存)指令的重试



如果轴0的原点复位完成，则执行速度控制



如果凸轮数据变量的变更和凸轮表的保存完成，且轴0到达目标速度，则执行凸轮动作



## ST 程序

```

// 指令的输入参数未设定时, 设定目标值等
IF InitFlag=FALSE THEN
    // 给速度控制 (MC_MoveVelocity) 的输入参数设定数值
    Vel_Vel      := LREAL#1000.0;
    Vel_Acc      := LREAL#0.0;
    Vel_Dec      := LREAL#0.0;
    Vel_Dir      := _eMC_DIRECTION#_mcPositiveDirection;

    // 给凸轮动作开始 (MC_CamIn) 的输入参数设定数值
    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;

    // 使输入参数的设定完成标志生效
    InitFlag:=TRUE;
END_IF;

// 如果触点 StartPg 为 TRUE, 且伺服驱动器处于伺服准备就绪状态, 则将轴 0 伺服 ON。
// 如果未处于伺服准备就绪状态, 则设为伺服 OFF。
IF (StartPg=TRUE)
AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr1_En:=TRUE;
ELSE
    Pwr1_En:=FALSE;
END_IF;

// 如果触点 StartPg 为 TRUE, 且伺服驱动器处于伺服准备就绪状态, 则将轴 1 伺服 ON。
// 如果未处于伺服准备就绪状态, 则设为伺服 OFF。
IF (StartPg=TRUE)
AND (MC_Axis001.DrvStatus.Ready=TRUE) THEN
    Pwr2_En:=TRUE;
ELSE
    Pwr2_En:=FALSE;
END_IF;

//MC 通用或轴发生轻度故障等级的异常时, 执行异常处理的 FaultHandler
// 请根据装置, 对异常处理的内容进行编程
IF (MC_Axis000.MFaultLvl.Active=TRUE)
OR (MC_Axis001.MFaultLvl.Active=TRUE)
OR (_MC_COM.MFaultLvl.Active=TRUE) THEN
    FaultHandler();
END_IF;

// 轴 0 处于伺服 ON 状态, 且原点未确定时, 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 处于伺服 ON 状态, 且原点未确定时, 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, 且凸轮表文件保存不处于执行状态时, 变更凸轮数据变量值
// 相位为 0 度 ~ 180 度时将位移设定为 2 倍、181 度 ~ 360 度时将位移设定为 0.5 倍
// 变更完成时, 将 WriteDone 设定为 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;

// 如果轴 0 的原点复位完成, 则执行速度控制
IF Hm1_D=TRUE THEN
    Vel_Ex:=TRUE;
END_IF;

// 凸轮数据变量的变更完成, 且SaveCamtable为TRUE、凸轮表文件保存不处于执行状态时, 将Sv_Cam_Ex设定为TRUE
// 如果 Sv_Cam_Ex 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 设定为 FALSE
// 如果 Sv_Cam_Ex 变为 FALSE, 则 Sv_Ca_TimeUp 变为 FALSE, 将 Sv_Cam_Ex 设定为 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, 且发生不可执行凸轮表保存的情况时, 将 Sv_Cam_Disable 设定为 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的情况 1 秒后, 将 Sv_Ca_TimeUp 设定为 TRUE
// 如果 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 设定为 TRUE
// 如果 Sv_Ca_CountLoad 变为 TRUE, 则复位重试计数器
R_TRIG1(SaveCamtable, Sv_Ca_CountLoad);

```

```

// 不可执行凸轮表保存的情况发生 3 次时, 将 Sv_Ca_CountUP 设定为 TRUE
// 如果 Sv_Ca_CountUP 变为 TRUE, 则将 Sv_Cam_Disable 设定为 FALSE
// 结束 MC_SaveCamTable( 凸轮表保存 ) 指令的重试
Sv_Ca_CTD(
    CD      := Sv_Cam_Disable ,
    LOAD    := Sv_Ca_CountLoad ,
    PV      := INT#3 ,
    Q       => Sv_Ca_CountUP
);

// 如果凸轮数据变量的变更和凸轮表的保存完成, 且轴 0 到达目标速度, 则执行凸轮动作
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临时改写轴参数

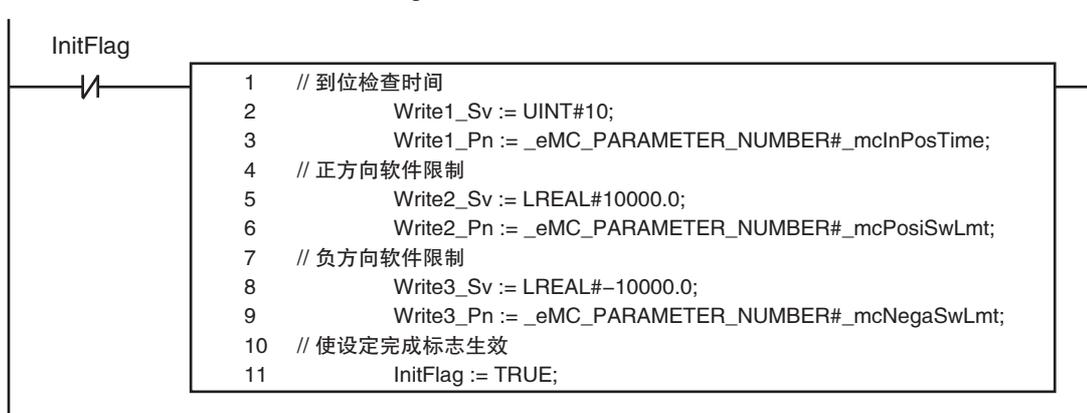
使用 MC 设定写入 (MC\_Write) 指令，变更到位检查时间、正方向软件限制、负方向软件限制的值

## 用于程序的主要变量

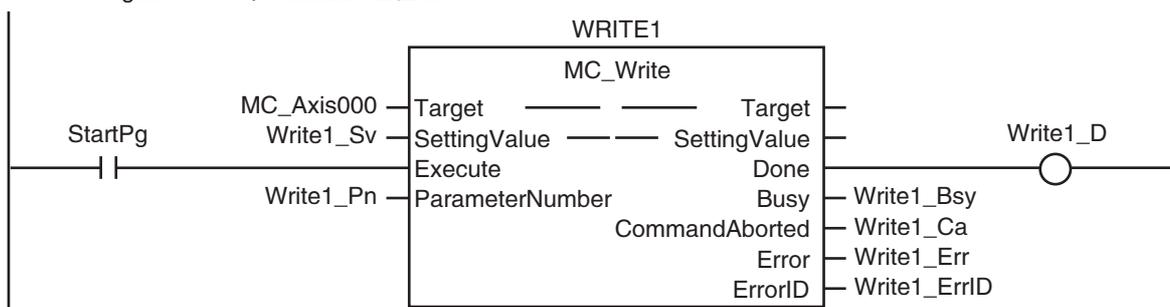
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	-	轴 0 的轴变量
InitFlag	BOOL	FALSE	表示参数设定状态的变量 FALSE 时执行参数变更，完成时设定为 TRUE
StartPg	BOOL	FALSE	用于执行 MC 设定写入的变量

## 梯形图程序

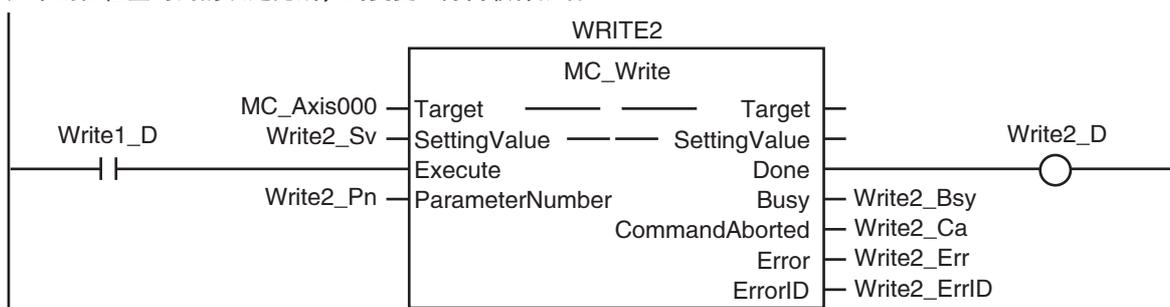
设定参数。如果设定完成，则将InitFlag设定为TRUE



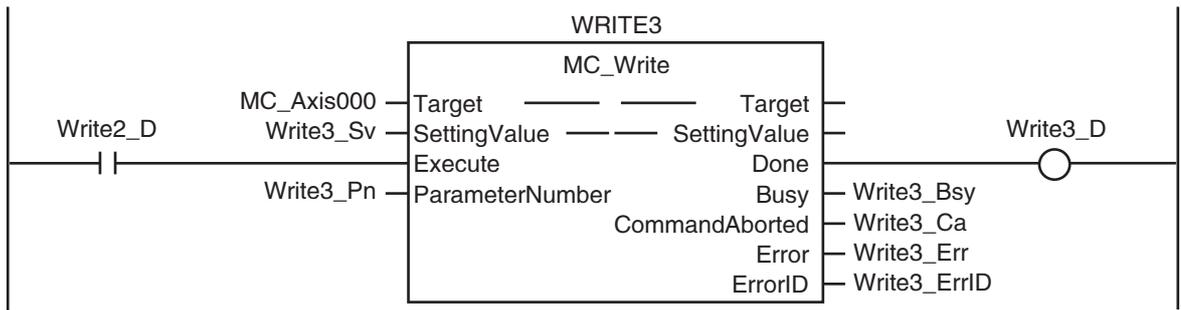
如果StartPg变为TRUE，则变更到位检查时间



如果到位检查时间的设定完成，则变更正方向软件限制



如果正方向软件限制的设定完成，则变更负方向软件限制



## ST 程序

```
// 设定参数。如果设定完成，则将 InitFlag 设定为 TRUE
IF InitFlag=FALSE THEN
    // 到位检查时间
    Write1_Sv := UINT#10;
    Write1_Pn := _eMC_PARAMETER_NUMBER#_mcInPosTime;
    // 正方向软件限制
    Write2_Sv := LREAL#10000.0;
    Write2_Pn := _eMC_PARAMETER_NUMBER#_mcPosiSwLmt;
    // 负方向软件限制
    Write3_Sv := LREAL#-10000.0;
    Write3_Pn := _eMC_PARAMETER_NUMBER#_mcNegaSwLmt;

    // 使输入参数的设定完成标志生效
    InitFlag := TRUE;
END_IF;

// 如果 StartPg 变为 TRUE，则变更到位检查时间
IF StartPg =TRUE THEN
    Write1_Ex := TRUE;
END_IF;

// 如果到位检查时间的变更完成，则变更正方向软件限制
IF Write1_D = TRUE THEN
    Write2_Ex := TRUE;
END_IF;

// 如果正方向软件限制的变更完成，则变更负方向软件限制
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更新凸轮表的终点索引

给最大凸轮数据数为 110 点、有效数据数为 100 点的凸轮表增加 10 点有效数据数，并更新终点索引。

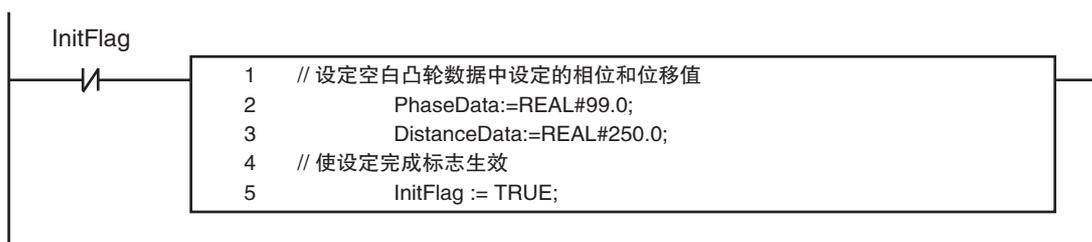
### 用于程序的主要变量

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 点的凸轮数据变量 *1 有效凸轮数据数为 100 点、空白凸轮数据数为 10 点
WriteCamdata	BOOL	FALSE	用于启动凸轮数据变更的变量 开始编辑时请设定为 TRUE
WriteDone	BOOL	FALSE	用于表示凸轮数据变更完成的变量 变更完成时请设定为 TRUE

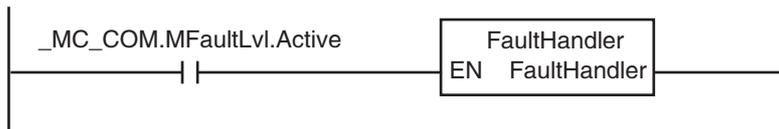
\*1. 数组的元素数 ARRAY[0..N] 由 Sysmac Studio 的凸轮编辑器功能设定。本示例程序中设定为 [0 ~ 109]。

### 梯形图程序

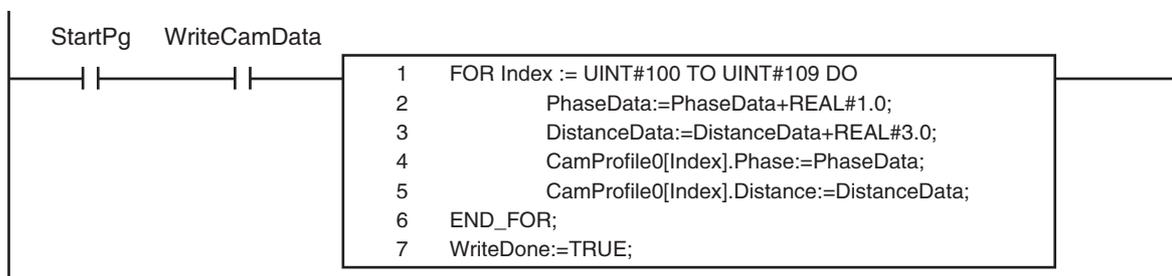
设定参数。如果设定完成，则将InitFlag设定为TRUE



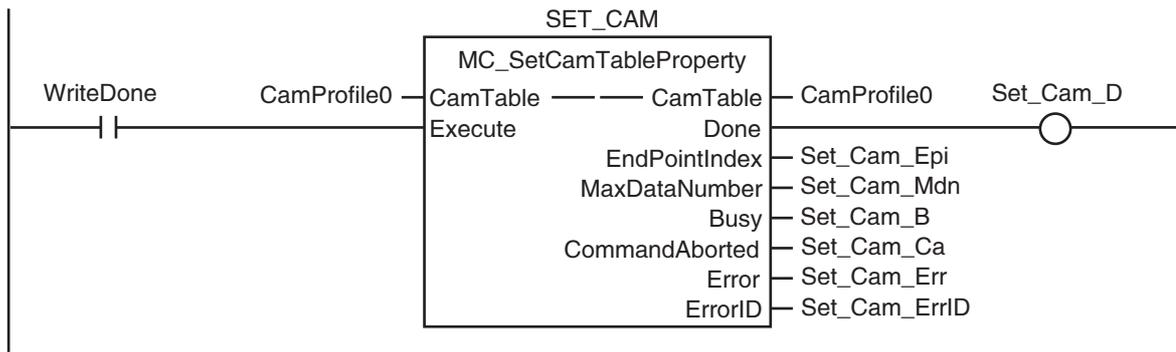
MC通用发生轻度故障等级的异常时，执行异常处理的FaultHandler  
请根据装置，对异常处理的内容进行编程



如果StartPg和WriteCamData变为TRUE，则变更凸轮数据变量值  
给CamProfile[100] ~ CamProfile[109]设定了相位和位移  
凸轮数据变量的变更完成时，将WriteDone设定为TRUE



如果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。如果设定完成，则将 InitFlag 设定为 TRUE
IF InitFlag=FALSE THEN
  // 设定空白凸轮数据中设定的相位和位移值
  PhaseData:=REAL#99.0;
  DistanceData:=REAL#250.0;

  // 将设定完成标志设定为 TRUE
  InitFlag := TRUE;
END_IF;

// MC 通用发生轻度故障等级的异常时，执行异常时处理
// 请根据装置，对异常处理的内容进行编程
IF _MC_COM.MFaultLvl.Active=TRUE THEN
  FaultHandler();
END_IF;

// 如果 StartPg 和 WriteCamData 变为 TRUE，则变更凸轮数据变量值
// 给 CamProfile[100] ~ CamProfile[109] 设定了相位和位移
// 凸轮数据变量的变更完成时，将 WriteDone 设定为 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;

// 如果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
);
```



# 异常的确认和处理

本章对 MC 功能模块中发生异常时的确认事项、根据显示的异常内容诊断异常及其措施、根据动作状态诊断异常及其措施进行说明。

---

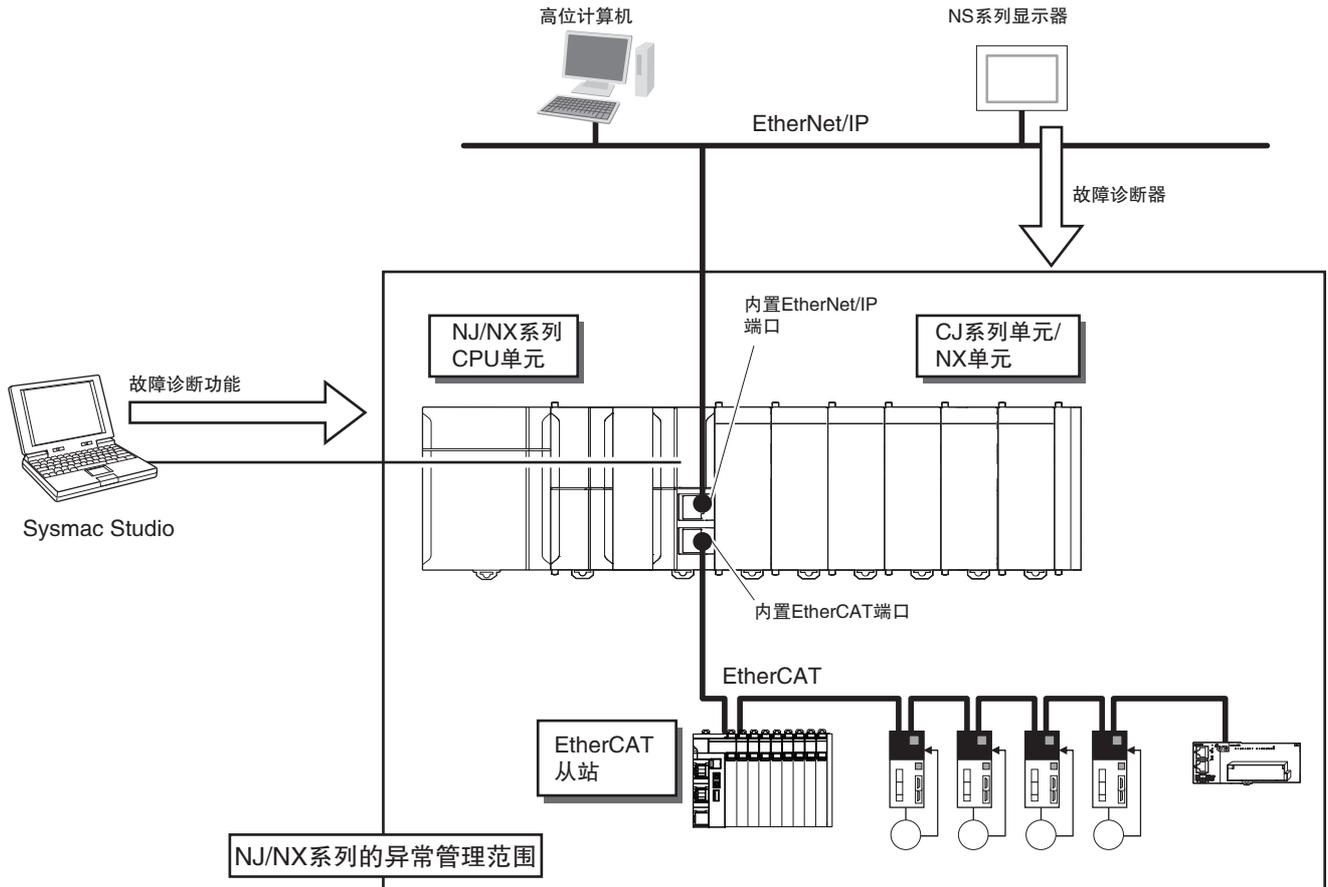
11-1 异常的概要 .....	11-2
11-1-1 异常的确认方法 .....	11-3
11-1-2 MC 功能模块相关的异常 .....	11-6
11-2 故障诊断 .....	11-10
11-2-1 异常一览表 .....	11-10
11-2-2 异常内容 .....	11-16
11-2-3 异常现象的推测原因和处理 .....	11-44

## 11-1 异常的概要

NJ/NX 系列采用相同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 (CPU 单元、NX 单元、NX 系列从站 EtherCAT 从站、CJ 系列单元) 的异常，可通过相同的方法确认异常内容和处理方法。

但是，EtherCAT 从站必须是 Sysmac 设备。

关于 Sysmac 设备，请参阅 □ “NJ/NX 系列 CPU 单元 用户手册 软件篇 (SBCA-359)”。



使用 Sysmac Studio 的故障诊断功能或显示器的故障诊断器，可确认异常内容及处理方法，因此可迅速做出应对。

使用 NS 系列显示器的故障诊断器时，请通过 CPU 单元内置的 EtherNet/IP 端口连接 NS 系列显示器。

本手册对 MC 功能模块中发生的异常进行说明。

发生异常时的具体处理方法和整个 NJ/NX 系列的故障诊断请参阅 □ “NJ/NX 系列 故障诊断手册 (SBCA-361)”。

启动运动控制指令时发生的异常请参阅 □ “NJ/NX 系列 指令基准手册 运动篇 (SBCE-364)” 的附录。



### 使用注意事项

- 仅 NJ 系列 CPU 单元可使用 CJ 单元。
- 仅 NX1P2 CPU 单元可安装 NX 单元。
- 关于显示器故障诊断器的适用范围，请参阅 □ “NJ/NX 系列 故障诊断手册 (SBCA-361)” 的附录。

## 11-1-1 异常的确认方法

可通过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 的故障诊断功能进行确认	正在发生的控制器异常、过去发生的控制器异常的履历、异常的发生源、异常发生原因及处理对策
通过显示器的故障诊断器进行确认 *1	正在发生的控制器异常、过去发生的控制器异常的履历、异常的发生源、异常发生原因及处理措施
通过各功能模块的异常状态获取指令进行确认	正在发生的控制器异常的最重要的状态和最重要的事件代码
通过系统定义变量进行确认	各功能模块正在发生的控制器异常的状态

\*1. 使用显示器的故障诊断器时，请通过 CPU 单元内置的 EtherNet/IP 端口连接显示器。关于显示器故障诊断器的适用范围，请参阅 □□ “NJ/NX 系列 故障诊断手册 (SBCA-361)” 的附录。

下面对各确认方法进行说明。

### 通过 LED 确认

可通过电源单元的 PWR LED 和 CPU 单元的 RUN、ERROR LED 的显示确认异常的重要程度。LED 显示与异常重要程度的关系如下所示。

#### ● NX 系列 CPU 单元

○：点亮 / ●：熄灭 / ◎：闪烁

LED			CPU 单元的动作状态	可否通过 Sysmac Studio 或显示器确认详细内容
PWR/ POWER (绿)	RUN (绿)	ERROR (红)		
●	●	●	电源部异常	不可
○	●	●	CPU 复位 *1 *2 *3 / 硬件初始化异常 *4 *2 *3	请参阅 □□ “NJ/NX 系列 故障诊断手册 (SBCA-361)”。
○	● 或 ◎	○	CPU 异常 *3 *5	
○	◎ 持续 30 秒以上	○	系统初始化异常	
○	●	○	全部停止故障等级 *3 *5	可 请连接 Sysmac Studio 或显示器，通过故障诊断功能或故障诊断器确认原因和处理措施。
○	○	◎	部分停止故障等级	
○	○	◎	轻度故障等级	
○	○	●	监控信息	
○	○	●	正常：运行模式下	-
○	●	●	正常：程序模式下 *2 *3	-
○	◎	●	正常：启动中	-

\*1. 使用 NX701 CPU 单元时的异常。记载的 LED 状态持续 30 秒以上，即为本异常。

\*2. 使用“USB- 直接连接”连接 Sysmac Studio，可在线连接时 CPU 单元为“程序模式下”，无法在线连接时 CPU 单元为“CPU 复位或硬件初始化异常”。

\*3. Sysmac Studio 无法在线连接时，也有可能是 USB 电缆故障或 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阅 □□ “NJ/NX 系列 故障诊断手册 (SBCA-361)”。

\*4. 使用 NX1P2 CPU 单元时的异常。记载的 LED 状态持续 30 秒以上，即为本异常。

\*5. 使用“USB- 直接连接”连接 Sysmac Studio，可在线连接时 CPU 单元为“全部停止故障等级”的异常，无法在线连接时为“CPU 异常”。

## ● NJ 系列 CPU 单元

○：点亮 / ●：熄灭 / ◎：闪烁

LED			CPU 单元的动作状态	可否通过 Sysmac Studio 或 NS 系列显示器 确认详细内容
PWR (绿)	RUN (绿)	ERROR (红)		
●	●	●	电源部异常	不可： 请参阅 □□ “NJ/NX 系列 故障诊断手册 (SBCA-361)”。
○	●	●	CPU 复位 *1*2	
○	◎	○	电源连接错误	
○	●	○	CPU 异常 (WDT 异常) *2*3	
○	●	○	全部停止故障等级 *2*3	可： 请连接 Sysmac Studio 或显示器，通过故障诊断功能或故障诊断器确认原因和处理。
○	○	◎	部分停止故障等级	
○	○	◎	轻度故障等级	
○	○	●	监控信息	
○	○	●	正常：运行模式下	
○	●	●	正常：程序模式下 *1*2	-
○	◎	●	正常：启动中	-

\*1 使用“USB-直接连接”连接 Sysmac Studio，可在线连接时 CPU 单元为“程序模式下”，无法在线连接时 CPU 单元为“CPU 复位”。

\*2 Sysmac Studio 无法在线连接时，也有可能是 USB 电缆故障或 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阅 □□ “NJ/NX 系列 故障诊断手册 (SBCA-361)”。

\*3 使用“USB-直接连接”连接 Sysmac Studio，可在线连接时 CPU 单元为“全部停止故障等级”的异常，无法在线连接时为“CPU 异常 (WDT 异常)”。

### 通过 Sysmac Studio 的故障诊断功能进行确认

异常发生时，将 Sysmac Studio 与控制器在线连接，可以确认正在发生的异常以及过去发生的异常的履历。还可确认异常的发生原因和处理措施。

使用 Sysmac Studio 确认异常的方法和处理方法请参阅 □□ “NJ/NX 系列 故障诊断手册 (SBCA-361)”。

### 通过 NS 系列显示器的故障诊断器进行确认

异常发生时，若 NS 系列显示器和控制器之间的通信已建立，可以确认正在发生的异常以及过去发生的异常的履历。

还可确认异常的发生原因和处理措施。

使用 NS 系列显示器的故障诊断器时，请通过 CPU 单元内置的 EtherNet/IP 端口连接 NS 系列显示器。

使用 NS 系列显示器确认异常的方法和处理方法请参阅 □□ “NJ/NX 系列 故障诊断手册 (SBCA-361)”。

### 通过异常状态获取指令进行确认

用户程序中可通过各功能模块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	运动控制异常状态获取	获取 MC 功能模块中发生的控制器异常的最重要状态 (部分停止故障、轻度故障) 和最重要的事件代码。

异常状态获取指令请参阅 □□ “NJ/NX 系列 指令基准手册 基本篇 (SBCA-360)”。

## 通过系统定义变量进行确认

MC功能模块中发生的异常可通过系统定义变量的“异常状态变量”和“运动控制系统变量”进行确认。

### ● 异常状态变量

NJ/NX 系列中发生的异常可通过各功能模块的异常状态变量进行确认。

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	集中表示 MC 功能模块中发生的所有异常状态
_MC_ComErrSta	WORD	MC 通用异常状态	集中表示 MC 功能模块的通用处理中发生的异常状态
_MC_AX_ErrSta	ARRAY[] OF WORD	轴异常状态	集中表示各轴发生的异常状态
_MC_GRP_ErrSta	ARRAY[] OF WORD	轴组异常状态	集中表示各轴组发生的异常状态

异常状态变量各位的含义如下所示。

位	名称	内容	值	含义
15	主站检测 *1	表示主站管理的从站有无异常	TRUE	有异常
			FALSE	无异常
14	从站集合 *2	集中表示分配至 MC 功能模块的轴的 EtherCAT 从站有无异常	TRUE	有异常
			FALSE	无异常
13 ~ 8	保留			
7	全部停止故障	表示有无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	表示有无部分停止故障等级的异常	TRUE	有异常
			FALSE	无异常
5	轻度故障	表示有无轻度故障等级的异常	TRUE	有异常
			FALSE	无异常
4	监控信息	表示有无监控信息等级的异常	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，仅 \_MC\_ErrSta(MC 异常状态) 中使用。

### ● 运动控制系统变量

通过查看运动控制系统变量的 MC 通用变量、轴变量 轴组变量，可确认 MC 功能模块中发生的异常。

运动控制系统变量请参阅 □ “6-6 运动控制系统变量 (P.6-16)”。

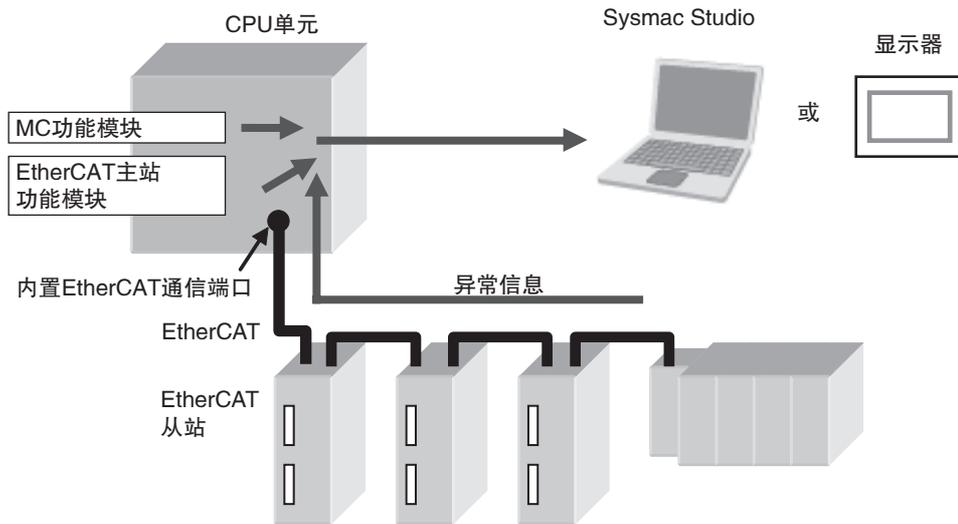
## 11-1-2 MC 功能模块相关的异常

下面对 MC 功能模块相关的异常进行说明。

### MC 功能模块相关的异常的发生部位

除 MC 功能模块内部发生的异常以外，还包括通过伺服驱动器等连接的 EtherCAT 通信导致的异常。

- MC 功能模块内部
- EtherCAT 主站功能模块
- 内置 EtherCAT 通信端口 (硬件)
- EtherCAT 从站



异常的发生源和发生原因可通过系统定义变量和 Sysmac Studio 或显示器进行确认。



#### 使用注意事项

关于显示器故障诊断器的适用范围，请参阅 □□ “NJ/NX 系列 故障诊断手册(SBCA-361)” 的附录。

### 分类

MC 功能模块异常的发生源分为以下 3 种。

分类	说明
MC 通用异常	检测到 MC 功能模块通用部分相关的异常时，将“MC 通用异常状态变量”的相应位设为“有异常”的状态
轴异常	检测到轴相关的异常时，将“轴异常状态变量”的相应位设为“有异常”的状态(*)
轴组异常	检测到轴组相关的异常时，将“轴组异常状态变量”的相应位设为“有异常”的状态

\*1. 发生轻度故障等级以上的轴异常时，构成轴中含有发生异常的轴的轴组也无法动作。

## 异常的重要程度

下面按各异常的重要程度对 MC 功能模块的动作进行说明。

异常的重要程度等级	动作
全部停止故障	NJ/NX 系列控制器整体控制停止的异常
部分停止故障	NJ/NX 系列控制器以功能模块为单位控制停止的异常 在 MC 功能模块中发生时，轴动作等 MC 功能模块具有的所有功能均停止
轻度故障	NJ/NX 系列控制器各功能模块的部分控制停止的异常 在 MC 功能模块中发生时，相应轴或相应轴组停止
监控信息	不影响 NJ/NX 系列控制器控制的异常 进行通知以防止发展至轻度故障等级以上的异常
一般信息	向用户通知的异常以外的信息

## MC 功能模块各发生源的异常

下面按发生源对各重要程度等级的异常进行说明。

### ● MC 通用为发生源的异常

重要程度	异常名称
全部停止故障	· 无
部分停止故障	<ul style="list-style-type: none"> <li>· 运动控制参数设定异常</li> <li>· 凸轮数据读取异常</li> <li>· 必需过程数据对象未设定</li> <li>· 禁用轴分配从站</li> <li>· 未登录轴分配从站网络构成信息</li> <li>· 运动控制初始化异常</li> <li>· 超过运动控制周期</li> <li>· 绝对值编码器原点位置偏置读取异常</li> </ul>
轻度故障	<ul style="list-style-type: none"> <li>· 凸轮表保存处理失败</li> <li>· 执行其它运动控制指令时的异常</li> </ul>
监控信息	<ul style="list-style-type: none"> <li>· 无法执行凸轮表保存指令</li> <li>· 运动控制异常解除指令数过多</li> </ul>
一般信息	<ul style="list-style-type: none"> <li>· 从 MC 试运转画面执行异常解除</li> </ul>

## ● 轴为发生源的异常

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	<ul style="list-style-type: none"> <li>· 凸轮动作中凸轮表数据异常</li> <li>· 启动立即停止指令</li> <li>· 超出正方向软件限制</li> <li>· 超出负方向软件限制</li> <li>· 超出到位检查时间</li> <li>· 超过位置偏差</li> <li>· 即停输入</li> <li>· 正方向极限输入检测</li> <li>· 负方向极限输入检测</li> <li>· 位置偏差错误</li> <li>· 伺服 OFF 异常</li> <li>· 无法计算绝对值编码器当前位置</li> <li>· 检测到伺服主电路电源 OFF</li> <li>· 无中断标准定位和中断信号</li> <li>· 检测到原点复位反方向极限输入</li> <li>· 检测到原点复位方向极限输入</li> <li>· 检测到原点复位双向极限输入</li> <li>· 检测到近原点、原点复位反方向极限输入</li> <li>· 检测到近原点、原点复位方向极限输入</li> <li>· 检测到原点输入、原点复位反方向极限输入</li> <li>· 检测到原点输入、原点复位方向极限输入</li> <li>· 原点输入屏蔽量错误</li> <li>· 无原点输入</li> <li>· 无近原点输入</li> <li>· 从站异常检测</li> <li>· MC 通用异常发生中</li> <li>· 锁定位置溢出</li> <li>· 锁定位置下溢</li> <li>· 主轴同步方向异常</li> <li>· 伺服 ON 中从站脱离</li> <li>· 标准距离溢出</li> <li>· 伺服驱动器的控制模式切换异常</li> <li>· 主轴位置读取异常</li> <li>· 辅助轴位置读取异常</li> <li>· 从站通信异常</li> <li>· 执行其它运动控制指令时的异常</li> </ul>
监控信息	<ul style="list-style-type: none"> <li>· 位置偏差警告</li> <li>· 速度警告</li> <li>· 加速度警告</li> <li>· 减速度警告</li> <li>· 正方向转矩警告</li> <li>· 负方向转矩警告</li> <li>· 指令位置溢出</li> <li>· 指令位置下溢</li> <li>· 反馈位置溢出</li> <li>· 反馈位置下溢</li> <li>· 从站监控信息检测</li> <li>· 到达合并中继速度之前的移动量不足通知</li> <li>· 执行其它运动控制指令时的异常</li> </ul>
一般信息	· 从站异常代码确定

## ● 轴组为发生源的异常

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	<ul style="list-style-type: none"> <li>· 启动轴组立即停止指令</li> <li>· 未确定多轴协调动作中的原点</li> <li>· 轴组构成轴发生异常</li> <li>· 执行其它运动控制指令时的异常</li> </ul>
监控信息	<ul style="list-style-type: none"> <li>· 速度警告</li> <li>· 加速度警告</li> <li>· 减速度警告</li> <li>· 到达合并中继速度之前的移动量不足通知</li> </ul>
一般信息	· 无

有些事件可以变更重要程度。关于重要程度的变更，请参阅 □ “NJ/NX 系列 CPU 单元 用户手册 软件篇 (SBCA-359)”。如需了解可变更重要程度的时间，请参阅本手册的 □ “11-2 故障诊断 (P.11-10)”。

## EtherCAT 通信、EtherCAT 从站、NX 单元相关的异常

因 EtherCAT 通信、EtherCAT 从站、NX 单元异常而发生的 MC 功能模块异常如下所示。

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	与分配至 MC 控制功能模块的轴的 EtherCAT 从站间，或 NX 单元的通信发生了异常*1	将发生异常的轴设为伺服 OFF，不接收解除异常以外的操作*2
从站异常检测	742F0000Hex	检测到分配至 MC 控制功能模块的轴的 EtherCAT 从站，或 NX 单元异常	将发生异常的轴设为伺服 OFF，不接收解除异常以外的操作

\*1 与 EtherCAT 从站之间的通信发生异常时，EtherCAT 主站功能模块也会发生异常。多设备分配给 1 个轴时，如某一设备发生通信异常，则该轴变为通信异常。

\*2 从站通信发生异常时，该轴变为未确定原点的状态。

## 伺服驱动器中发生的异常

下面对欧姆龙制 伺服驱动器 1S 系列及 G5 系列发生异常时的通知进行说明。

MC 功能模块检测到伺服驱动器发生异常的时间与可从伺服驱动器获取异常代码的时间之间存在时间差。因此，MC 功能模块将伺服驱动器发生的异常和异常代码作为不同事件进行通知。

### ● 发生异常的通知

MC 功能模块检测到伺服驱动器发生异常时，会发生轻度故障等级“检测到从站异常”(742F0000Hex)。此时，MC 功能模块会执行发生异常时的动作(伺服 OFF)。

### ● 异常代码的通知

在伺服驱动器中确定异常代码时，MC 功能模块会出现一般信息“确定从站异常代码”(94220000Hex)。将在“确定从站异常代码”附加信息的后 2 位显示伺服驱动器的异常代码(错误显示 No. 的主编号)。例如，附加信息显示“FF13”时，伺服驱动器出现错误显示 No.13(主电路电源电压不足)。



### 使用注意事项

通知“确定从站异常代码”时需进行设定。  
请在 PDO 映射编辑画面中映射对象 603FHex(错误代码)。

## NX 单元发生的异常

在欧姆龙制 NX 系列 位置接口单元发生异常时与欧姆龙制伺服驱动器 1S 系列及 G5 系列相同，将通知异常发生和异常代码。

但，NX 系列 位置接口单元没有与 603FHex(错误代码)相当的对象，因此“确定从站异常代码”(94220000Hex)的附加信息中显示“0000Hex”。

关于 NX 系列 位置接口单元发生异常的详情，请参阅 □□ “NX 系列 位置接口单元 用户手册 (SBCE-374)”及 □□ “NX 系列 EtherCAT 耦合器单元 用户手册 (SBCD-361)”。

## 11-2 故障诊断

下面对发生的异常及其处理进行说明。

### 11-2-1 异常一览表

以下是 MC 功能模块发生的异常 (事件) 的一览表。重要程度栏中使用了以下简称和记号。

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	监控信息
总	一般信息

记号	含义
○	系统定义的重要程度
◎	用户可变更的重要程度 *1

\*1. 只有存在可变更重要程度时才记载。

输出至运动控制系统变量的异常代码为下表事件代码的前 4 位。

事件代码栏 () 内表示增加事件的 CPU 单元的单元版本。

关于 NJ/NX 系列的所有事件代码, 请参阅 □ “NJ/NX 系列故障诊断手册 (SBCA-361)”。

事件代码	事件名称	内容	发生原因 (推测原因)	重要程度					参考页
				全	部	轻	监	总	
44210000Hex	运动控制功能处理异常	在运动控制功能模块检出严重异常。	软件发生了错误	○					P.11-17
14600000Hex	绝对值编码器原点位置偏置读取异常	电源 OFF 中保持的绝对值编码器当前位置消失	<ul style="list-style-type: none"> <li>保持变量为电池备份时, CPU 单元的电池寿命</li> <li>软件发生了错误</li> <li>备份存储器故障</li> </ul>		○				P.11-17
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	<ul style="list-style-type: none"> <li>下载运动控制参数设定过程中或存储器清除过程中, 出现了断电或切断了与 Sysmac Studio 之间的通信</li> <li>非易失性存储器故障</li> </ul>		○				P.11-18
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	<ul style="list-style-type: none"> <li>保存凸轮数据过程中断电</li> <li>非易失性存储器故障</li> </ul>		○				P.11-18
34600000Hex	必需过程数据对象未设定	轴类型对应的必需对象未分配至 PDO	<ul style="list-style-type: none"> <li>轴类型为伺服轴或编码器轴时, 未执行必要的 PDO 映射</li> <li>非易失性存储器故障</li> </ul>		○				P.11-19
34630000Hex	禁用轴分配从站	为轴分配的从站设定为禁用	<ul style="list-style-type: none"> <li>为轴分配的从站设定为禁用</li> </ul>		○				P.11-19
34640000Hex	未登录轴分配从站网络构成信息	轴上分配的从站未登录到网络构成信息中	<ul style="list-style-type: none"> <li>轴上分配的从站未登录到 EtherCAT 的网络构成信息中</li> </ul>		○				P.11-20

事件代码	事件名称	内容	发生原因(推测原因)	重要程度					参考页
				全	部	轻	监	总	
44200000Hex	运动控制初始化异常	系统中发生了致命性故障, 运动控制初始化失败	· 硬件故障		○				P.11-20
74200000Hex	超过运动控制周期	未在两个控制周期内完成原始恒定周期任务的处理	· 原始恒定周期任务程序的处理内容过多		○				P.11-20
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.11-21
54770000Hex	凸轮动作中凸轮表数据异常	凸轮表的相位非升序	· 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位移不是 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不是 1 个脉冲以上			○			P.11-21
54850000Hex	启动立即停止指令	执行了立即停止(MC_ImmediateStop)指令	· 执行了立即停止指令			○			P.11-21
54860000Hex	启动轴组立即停止指令	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.11-22
64450000Hex	超出正方向软件限制	在轴动作过程中, 位置超出了正方向软件限制	· 位置超出了正方向软件限制			○			P.11-22
64460000Hex	超出了负方向软件限制	在轴动作过程中, 位置超出了负方向软件限制	· 位置超出了负方向软件限制			○			P.11-22
64470000Hex	超出到位检查时间	在监控时间内未完成定位完成检查	· 定位完成需要花费很长时间			○			P.11-23
64480000Hex	超过位置偏差	指令当前位置和反馈当前位置的偏差超过了位置偏差超出值	· 定位动作的随动性较差, 实际动作比指令延迟			○			P.11-23
64490000Hex	即停输入	即停输入变为 ON	· 检测到即停输入信号 · 即停信号的连接或即停输入的逻辑设定错误			○			P.11-23
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.11-24
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.11-24
64560000Hex	位置偏差错误	将指定的位置和反馈当前位置的差值转换为脉冲单位时, 变为 30 位	· 在从轴中, 指定轴的移动量超过了轴的最高速度, 对指令当前位置执行限制, 防止超过最高速度 · 从轴定位动作的随动性较差, 实际动作比指令延迟			○			P.11-25
64570000Hex	伺服 OFF 异常	因轴组发生异常, 轴伺服 OFF	· 因轴组发生异常, 轴伺服 OFF			○			P.11-25

事件代码	事件名称	内容	发生原因 (推测原因)	重要程度					参考页
				全	部	轻	监	总	
64580000 Hex	无法计算绝对值编码器当前位置	不能通过断电保持的绝对值编码器信息正确恢复当前位置	<ul style="list-style-type: none"> <li>· 变更了环计数器设定或伺服驱动器设定的环计数器设定</li> <li>· 复位位置超出了脉冲单位的带符号 40 位的范围</li> </ul>			○			P.11-26
64590000 Hex	未确定多轴协调动作中的原点	轴组动作中或减速停止中未确定逻辑轴的原点	<ul style="list-style-type: none"> <li>· 轴组动作中或减速停止中, 发生了逻辑轴的指令位置或反馈位置的溢出或下溢, 未确定原点</li> <li>· 轴组动作中或减速停止中, 逻辑轴发生从站通信异常, 未确定原点</li> <li>· 轴组动作或减速停止时, 逻辑轴的从站脱离或禁用, 且未确定原点</li> </ul>			○			P.11-26
74210000 Hex	检测到伺服主电路电源 OFF	在伺服 ON 状态下, 将伺服驱动器的主电路电源设为 OFF	<ul style="list-style-type: none"> <li>· 在伺服 ON 状态下, 切断伺服驱动器的主电路电源</li> </ul>			○			P.11-27
74230000 Hex	无中断标准定位和中断信号	MC_MoveFeed( 中断标准定位 ) 指令执行中无中断输入	<ul style="list-style-type: none"> <li>· 锁定有效范围指定不恰当</li> <li>· 中断信号配线错误</li> <li>· 输出中断信号的传感器故障</li> </ul>			○			P.11-27
74240000 Hex	检测到原点复位反方向极限输入	原点复位动作中, 检测到原点复位方向和反方向极限信号	<ul style="list-style-type: none"> <li>· 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 “不反转”</li> <li>· 原点复位时的输入信号传感器配置、原点复位设定、原点复位开始位置到达极限信号</li> <li>· 输入信号传感器接线错误、故障</li> </ul>			○			P.11-28
74250000 Hex	检测到原点复位方向极限输入	原点复位动作中, 检测到原点复位方向的极限信号	<ul style="list-style-type: none"> <li>· 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 “不反转”</li> <li>· 原点复位时的输入信号传感器配置、原点复位设定、原点复位开始位置到达极限信号</li> <li>· 输入信号传感器接线错误、故障</li> </ul>			○			P.11-28
74260000 Hex	检测到原点复位双向极限输入	原点复位动作中, 同时检测到两侧的极限信号	<ul style="list-style-type: none"> <li>· 极限信号的接线错误</li> <li>· 极限传感器的安装位置错误</li> <li>· 极限信号的接点逻辑错误</li> <li>· 极限传感器故障</li> </ul>			○			P.11-29
74270000 Hex	检测到近原点、原点复位反方向极限输入	原点复位动作中, 同时检测到近原点输入、原点复位方向和反方向的极限信号	<ul style="list-style-type: none"> <li>· 近原点信号、极限信号的接线错误</li> <li>· 近原点传感器、极限传感器安装位置错误</li> <li>· 近原点信号、极限信号的接点逻辑错误</li> <li>· 近原点传感器、极限传感器故障</li> </ul>			○			P.11-29

事件代码	事件名称	内容	发生原因(推测原因)	重要程度					参考页
				全	部	轻	监	总	
74280000Hex	检测到近原点、原点复位方向极限输入	原点复位动作中,同时检测到近原点输入、原点复位方向的极限信号	<ul style="list-style-type: none"> <li>近原点信号、极限信号的接线错误</li> <li>近原点传感器、极限传感器安装位置错误</li> <li>近原点信号、极限信号的接点逻辑错误</li> <li>近原点传感器、极限传感器故障</li> </ul>			○			P.11-30
74290000Hex	检测到原点输入、原点复位反方向极限输入	原点复位动作中,同时检测到原点输入、原点复位方向和反方向的极限信号	<ul style="list-style-type: none"> <li>原点输入信号、极限信号的接线错误</li> <li>原点输入传感器、极限传感器安装位置错误</li> <li>原点输入信号、极限信号的接点逻辑错误</li> <li>原点输入信号输出设备、极限传感器故障</li> </ul>			○			P.11-30
742A0000Hex	检测到原点输入、原点复位方向极限输入	原点复位动作中,同时检测到原点输入、原点复位方向的极限信号	<ul style="list-style-type: none"> <li>原点输入信号、极限信号的接线错误</li> <li>原点输入传感器、极限传感器安装位置错误</li> <li>原点输入信号、极限信号的接点逻辑错误</li> <li>原点输入信号输出设备、极限传感器故障</li> </ul>			○			P.11-31
742B0000Hex	原点输入屏蔽量错误	MC_Home(原点复位)指令或 MC_Home WithParameter(参数指定原点复位)指令中,原点输入屏蔽量设定值不恰当	原点复位动作模式为“指定附近避让、原点输入屏蔽距离”时,原点输入屏蔽距离的设定值由原点复位速度减速为原点复位接近速度时的移动量不足			○			P.11-31
742C0000Hex	无原点输入	原点复位中无原点信号输入。或在无原点确定输入的情况下检测到极限信号	<ul style="list-style-type: none"> <li>原点复位中无原点信号输入。</li> <li>或在无原点确定输入的情况下检测到极限信号</li> </ul>			○			P.11-32
742D0000Hex	无近原点输入	原点复位中无近原点信号输入	设定为“有近原点信号”时,原点复位中无近原点信号输入			○			P.11-32
742F0000Hex	从站异常检测	检测到轴内分配的EtherCAT从站或NX单元异常	轴内分配的EtherCAT从站或NX单元检测到异常			○			P.11-32
74300000Hex	轴组构成轴发生异常	轴组构成轴发生了异常	轴组动作中,构成轴发生了异常			○			P.11-33
74330000Hex	MC通用异常发生中	发生了MC通用异常	发生了部分停止故障等级的MC通用异常			○			P.11-33
74340000Hex	锁定位置溢出	MC_TouchProbe(启用外部锁定)指令锁定的位置溢出	MC_TouchProbe(启用外部锁定)指令锁定的位置溢出			○			P.11-33
74350000Hex	锁定位置下溢	MC_TouchProbe(启用外部锁定)指令锁定的位置下溢	MC_TouchProbe(启用外部锁定)指令锁定的位置下溢			○			P.11-34
74360000Hex	主轴同步方向异常	主轴向同步方向相反的方向持续移动	与主轴和从轴同步方向相对,主轴持续向相反方向移动并溢出			○			P.11-34

事件代码	事件名称	内容	发生原因 (推测原因)	重要程度					参考页
				全	部	轻	监	总	
74370000Hex	伺服 ON 中从站脱离	伺服 ON 期间分配给轴的 EtherCAT 从站或 NX 单元出现了以下任一操作。 · 脱离 / 更换 · 禁用 · NX 系列 CPU 单元的 NX 总线重启	伺服 ON 期间分配给轴的 EtherCAT 从站或 NX 单元出现了以下任一操作。 · 脱离 / 更换 · 禁用 · NX 系列 CPU 单元的 NX 总线重启			○			P.11-34
74380000Hex	标准距离溢出	MC_MoveFeed(中断标准定位)指令中断输入后的目标位置溢出或下溢	· 将 MC_MoveFeed(中断标准定位)指令中断输入后的目标位置转换为脉冲数单位时,超出了带符号 40 位的范围			○			P.11-35
74390000Hex	伺服驱动器的控制模式切换异常	在切换控制模式时,在规定的时间内未完成处理	· 停止 MC_SyncMoveVelocity(周期同步速度控制)指令时,未完成规定的减速处理 · 使用欧姆龙制伺服驱动器 1S 系列或 G5 系列时,停止 MC_Torque Control(转矩控制)指令时,未完成规定的减速处理 · 切换控制模式 (GSP/CSV/CST) 后,1 秒以内未完成伺服驱动器的模式切换			○			P.11-35
743A0000Hex	主轴位置读取异常	因同步指令的主轴位置发生异常,无法执行同步指令	· 未确立同步控制指令主轴的 EtherCAT 过程数据通信,或 NX 单元的 I/O 数据无法用于控制。 · 同步指令的主轴发生从站脱离或禁用从站 · 同步指令的主轴中检测到无法计算绝对值编码器当前位置 (64580000Hex) · 同步指令主轴为未使用轴			○			P.11-36
743B0000Hex	辅助轴位置读取异常	因同步指令的辅轴位置发生异常,无法执行同步指令	· 未确立同步控制指令辅轴的 EtherCAT 过程数据通信,或 NX 单元的 I/O 数据无法用于控制。 · 同步指令的辅轴发生从站脱离或禁用 · 同步指令的辅轴中检测到无法计算绝对值编码器当前位置 (64580000Hex) · 同步指令辅轴为未使用轴			○			P.11-37
84400000Hex	从站通信异常	与轴内分配的 EtherCAT 从站或 NX 单元进行通信时发生了异常	· 与轴内分配的 EtherCAT 从站或 NX 单元进行通信时发生了异常			○			P.11-37
571D0000Hex (Ver.1.02 ~ Ver.1.09)	运动控制异常解除指令数过多	ResetMCErr(运动控制异常解除)指令的实例数超过了 100 个	· 包含功能块内的实例在内,在用户程序内宣布的 ResetMCErr(运动控制异常解除)指令中的实例数超过了 100 个			○			P.11-38
644C0000Hex	位置偏差警告	位置偏差超过了位置偏差警告值	· 定位动作的随动性较差,实际动作比指令延迟			○			P.11-38

事件代码	事件名称	内容	发生原因(推测原因)	重要程度					参考页
				全	部	轻	监	总	
644D0000Hex	速度警告	指令速度超过了速度警告值	· 指令速度超过了速度警告值			◎	○		P.11-39
644E0000Hex	加速度警告	指令加速度超过了加速度警告值	· 指令加速度超过了加速度警告值			◎	○		P.11-39
644F0000Hex	减速度警告	指令减速度超过了减速度警告值	· 指令减速度超过了减速度警告值			◎	○		P.11-39
64500000Hex	正方向转矩警告	转矩指令值超过了正方向转矩警告值	· 转矩指令值超过了正方向转矩警告值			◎	○		P.11-40
64510000Hex	负方向转矩警告	转矩指令值超过了负方向转矩警告值	· 转矩指令值超过了负方向转矩警告值			◎	○		P.11-40
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, 超过了带符号40位的上限值			◎	○		P.11-40
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, 超过了带符号40位的下限值			◎	○		P.11-41
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, 超出了带符号40位的上限值			◎	○		P.11-41
64550000Hex	反馈位置下溢	反馈位置对应的脉冲数下溢(超出范围)	· 将反馈位置转换为脉冲单位时, 超出了带符号40位的下限值			◎	○		P.11-41
74320000Hex	从站监控信息检测	检测到 EtherCAT 从站或 NX 单元的公告	· 轴内分配的 EtherCAT 从站或 NX 单元检测到警告			◎	○		P.11-42
743C0000Hex	无法执行凸轮表保存指令	因其它操作正在访问非易失性存储器, 未能执行凸轮表文件保存	· 因其它操作(使用 Sysmac Studio 的传送、数据跟踪等)正在访问非易失性存储器, 执行了 MC_SaveCamTable(凸轮表保存)指令				○		P.11-42
94200000Hex	到达合并中继速度之前的移动量不足通知	用于合并中继速度加减速的移动量不足	· “加减速超限”设定为“提高加减速(将合并切换为等待)”时, 指定 Blending 时, 生成曲线的结果, 加减速超限, 因此变更为 Buffered 处理 · 指定 Blending 时, 因已到达目标位置, 不能正确绘制曲线, 因此变更为 Buffered 处理			◎	○		P.11-42
94210000Hex	从 MC 试运转画面执行异常解除	从 Sysmac Studio 试运转画面执行了异常解除	· 从 Sysmac Studio 试运转画面执行了异常解除					○	P.11-43
94220000Hex	从站异常代码确定	因发生“从站异常检测”, 由从站发出异常代码通知	· 因发生“从站异常检测”(742F0000Hex), 由从站发出异常代码通知					○	P.11-43

## 11-2-2 异常内容

下面对各异常的内容进行说明。

### 控制器异常说明的解释

各异常的说明使用的表的各项目的含义在 [] 内表示。

事件名称	[ 异常 (事件) 的名称 ]		事件代码	[ 异常 (事件) 的代码 ]		
内容	[ 异常 (事件) 的内容 ]					
发生源	[ 异常 (事件) 发生的部位 ]		发生源详情	[ 发生源的详情 ]	检测时间	[ 异常检测的时间 ]
异常的属性	重要程度	[ 对控制产生影响的程度 ]*1	恢复方法	[ 恢复方法 ]*2	日志类别	[ 待保存的日志种类 ]*3
发生后的影响	用户程序	[ 用户程序的执行状态 ]*4	动作	[ 异常 (事件) 发生时的动作相关特别说明。 ]		
LED	[ 内置 EtherNet/IP 端口用 LED、内置 EtherCAT 端口用 LED 的显示状态。仅 EtherCAT 主站功能模块、EtherNet/IP 功能模块时记载发生源 ]					
系统定义变量	变量名称		数据类型		名称	
	[ 检测异常的系统定义变量、受异常影响的系统定义变量、导致异常的系统定义变量的变量名称和数据类型、名称 ]					
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	[ 异常 (事件) 的发生原因、处理措施及防止再次发生的方法 ]					
附加信息	[ Sysmac Studio/NS 系列显示器中显示的附加信息内容 ]*5					
注意事项 / 备注	[ 其他注意事项、限制事项、补充说明等。若有用户可以选择的重要程度, 则记载可以选择的重要程度、恢复方法、动作等相关事项 ]					

\*1. 以下中的任意一个

全部停止故障: 全部停止故障等级  
 部分停止故障: 部分停止故障等级  
 轻度故障: 轻度故障等级  
 监控信息  
 总体信息

\*2. 以下中的任意一个

自动恢复: 排除故障后自动恢复正常  
 异常解除: 排除故障后通过执行异常解除恢复正常  
 重新接通电源: 排除故障后通过重新接通电源恢复正常  
 制器复位: 排除故障后通过控制器复位恢复正常  
 基于发生原因: 取决于发生原因

\*3. 以下中的任意一个

系统日志: 系统事件日志  
 访问日志: 访问事件日志

\*4. 以下中的任意一个

继续: 继续执行用户程序  
 停止: 停止执行用户程序  
 开始: 开始执行用户程序

\*5. 关于显示器故障诊断器的适用范围, 请参阅 □□ “NJ/NX 系列 故障诊断手册 (SBCA-361)” 的附录。



#### 版本相关信息

对于 NX701 CPU 单元, 以 `_MC_AX[*]` 开头的变量名可能为 `_MC1_AX[*]`、`_MC2_AX[*]`。同样, 以 `_MC_GRP[*]` 开头的变量名可能为 `_MC1_GRP[*]`、`_MC2_GRP[*]`。

## 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	在运动控制功能模块检出严重异常。					
发生源	PLC 功能模块		发生源详情	MC 通用	检测时间	常时
异常的属性	重要程度	全部停止故障	恢复方法	重新接通电源	日志类别	系统
发生后的影响	用户程序	停止	动作	无法控制轴。控制器停止。		
系统定义变量	变量名称		数据类型		名称	
	无		-		-	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	软件发生了错误。		请向本公司营业部或代理商咨询。		无	
附加信息	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	绝对值编码器原点位置偏置读取异常			事件代码	1460 0000Hex	
内容	电源 OFF 中保持的绝对值编码器当前位置消失					
发生源	运动控制功能模块		发生源详情	MC 通用	检测时间	电源接通时 / 控制器复位时 / 下载时
异常的属性	重要程度	部分停止故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	无法控制轴		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.PFaultLvl.Active		BOOL		正在发生 MC 通用 部分停止故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	保持变量为电池备份时, CPU 单元的电池寿命		请在更换 CPU 单元的电池后解除异常, 执行原点复位并确定原点。		请定期更换 CPU 单元的电池。电池的寿命请参阅“NX 系列 CPU 单元用户手册 硬件篇 (SBCA-418)”或“NJ 系列 CPU 单元用户手册 硬件篇 (SBCA-358)”。	
	软件发生了错误		再次发生本异常时, 请在更换 CPU 单元后解除异常, 执行原点复位并确定原点。		无	
	备份存储器故障					
附加信息	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					
发生源	运动控制功能模块		发生源详情	MC 通用	检测时间	电源接通时 / 控制器复位时 / 下载时
异常的属性	重要程度	部分停止故障	恢复方法	电源接通 / 控制器复位	日志类别	系统
发生后的影响	用户程序	继续	动作	无法控制轴		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.PFaultLvl.Active		BOOL		正在发生 MC 通用 部分停止故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	下载运动控制参数设定过程中或存储器清除过程中, 出现了断电或切断了与 Sysmac Studio 之间的通信		请从 Sysmac Studio 下载运动控制参数设定		保存参数设定时请勿切断电源	
	非易失性存储器故障		如果进行上述处理后仍再次发生本异常, 则为非易失性存储器故障。请在更换 CPU 单元后从 Sysmac Studio 中下载含轴设定的所有设定		无	
附加信息	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					
发生源	运动控制功能模块		发生源详情	MC 通用	检测时间	电源接通时 / 控制器复位时 / 下载时
异常的属性	重要程度	部分停止故障	恢复方法	电源接通 / 控制器复位	日志类别	系统
发生后的影响	用户程序	继续	动作	无法控制轴		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.PFaultLvl.Active		BOOL		正在发生 MC 通用 部分停止故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	保存凸轮数据过程中断电		请从 Sysmac Studio 中下载凸轮数据		保存凸轮数据时请勿切断电源	
	非易失性存储器故障		如果进行上述处理后仍再次发生本异常, 则为非易失性存储器故障。请在更换 CPU 单元后从 Sysmac Studio 中下载含轴设定的所有设定		无	
附加信息	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	3460 0000 Hex	
内容	轴类型对应的必需对象未分配至 PDO					
发生源	运动控制功能模块		发生源详情	MC 通用	检测时间	电源接通时 / 控制器复位时 / 下载时
异常的属性	重要程度	部分停止故障	恢复方法	电源接通 / 控制器复位	日志类别	系统
发生后的影响	用户程序	继续	动作	无法控制轴		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.PFaultLvl.Active		BOOL		正在发生 MC 通用 部分停止故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	轴类型为伺服轴或编码器轴时, 未执行必要的 PDO 映射		请按相应轴种类进行必要的 PDO 映射。 必要的 PDO 映射请参阅 <input type="checkbox"/> “A-1-2 伺服驱动器的设定 (P.A-2)” 或 <input type="checkbox"/> “A-2-2 伺服驱动器的设定 (P.A-10)”。		请按需使用的轴种类进行必要的 PDO 映射。 必要的 PDO 映射请参阅 <input type="checkbox"/> “A-1-2 伺服驱动器的设定 (P.A-2)” 或 <input type="checkbox"/> “A-2-2 伺服驱动器的设定 (P.A-10)”。	
	非易失性存储器故障		如果进行上述处理后仍再次发生本异常, 则为非易失性存储器故障。请在更换 CPU 单元后从 Sysmac Studio 中下载含轴参数设定的所有设定		无	
附加信息	无					
注意事项 / 备注	无					

事件名称	禁用轴分配从站			事件代码	3463 0000 Hex	
内容	为轴分配的从站设定为禁用					
发生源	运动控制功能模块		发生源详情	MC 通用	检测时间	电源接通时 / 控制器复位时 / 下载时
异常的属性	重要程度	部分停止故障	恢复方法	电源重新接通 / 控制器复位	日志类别	系统
发生后的影响	用户程序	继续	动作	无法控制轴		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.PFaultLvl.Active		BOOL		正在发生 MC 通用 部分停止故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	为轴分配的从站设定为禁用		设定轴分配的相应从站的 EtherCAT 时, 请将启用 / 禁用从站设定为 “启用”。 无从站时, 请将相应轴的 “轴种类” 设定为 “虚拟轴”		设定轴分配的从站的 EtherCAT 时, 请将启用 / 禁用从站设定为 “启用”。 无从站但需在程序中使用轴时, 请将 “轴种类” 设成 “虚拟轴”	
附加信息	无					
注意事项 / 备注	无					

事件名称	未登录轴分配从站网络构成信息			事件代码	34640000Hex	
内容	轴上分配的从站未登录到网络构成信息中					
发生源	运动控制功能模块		发生源详情	MC 通用	检测时间	电源接通时 / 控制器复位时 / 下载时 / 伺服 ON 开始时 / 从未使用轴切换至使用轴时
异常的属性	重要程度	部分停止故障	恢复方法	电源重新接通 / 控制器复位	日志类别	系统
发生后的影响	用户程序	继续	动作	无法控制轴		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.PFaultLvl.Active		BOOL		正在发生 MC 通用 部分停止故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	轴上分配的从站未登录到 EtherCAT 的网络构成信息中		请将为轴分配的从站登录至 EtherCAT 的网络构成信息。或请将相应轴的“轴种类”设定为“虚拟轴”。		请将为轴分配的从站登录至网络构成信息	
附加信息	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	系统中发生了致命性故障，运动控制功能初始化失败					
发生源	运动控制功能模块		发生源详情	MC 通用	检测时间	电源接通时 / 控制器复位时 / 下载时
异常的属性	重要程度	部分停止故障	恢复方法	重新接通电源	日志类别	系统
发生后的影响	用户程序	继续	动作	无法控制轴、无法执行运动控制指令		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.PFaultLvl.Active		BOOL		正在发生 MC 通用 部分停止故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	硬件故障		请更换 CPU 单元		无	
附加信息	附加信息 1: 系统信息					
注意事项 / 备注	无					

事件名称	超过运动控制周期			事件代码	74200000Hex	
内容	未在两个控制周期内完成原始恒定周期任务的处理					
发生源	运动控制功能模块		发生源详情	MC 通用	检测时间	常时
异常的属性	重要程度	部分停止故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	所有轴无法动作、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		正在发生 MC 通用 部分停止故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	原始恒定周期任务程序的处理内容过多		请减少原始恒定周期任务的程序的处理内容或在不会使动作发生问题的范围内设定更大的控制周期值。任务周期可通过 Sysmac Studio 的“监控任务周期”进行确认		请务必仅将需在固定周期内执行的程序的处理记述在原始恒定周期任务的程序中。或请设定足够长的原始恒定周期任务的周期	
附加信息	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	1463 0000 Hex	
内容	凸轮表文件保存失败					
发生源	运动控制功能模块		发生源详情	MC 通用	检测时间	执行指令时
异常的属性	重要程度	轻度故障	恢复方法	解除异常 / 重新接通电源	日志类别	系统
发生后的影响	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会发生异常		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.MFaultLvl.Active		BOOL		正在发生 MC 通用轻度故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	凸轮表文件保存失败		请再次进行保存。如果再次发生，则为非易失性存储器故障。请更换 CPU 单元		无	
附加信息	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	5477 0000 Hex	
内容	凸轮表的相位非升序					
发生源	运动控制功能模块		发生源详情	轴	检测时间	执行指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴无法动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	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，以使相位为升序		请确保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位移不是 0		请修正凸轮表数据，以使起点相位和位移为 0		请使凸轮表数据起点的相位和位移为 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不是 1 个脉冲以上		请修正凸轮表数据，以便将终点的相位转换为脉冲单位时为 1 个脉冲以上		将终点的相位转换为脉冲单位时，请使凸轮表数据为 1 个脉冲以上	
附加信息	无					
注意事项 / 备注	无					

事件名称	启动立即停止指令			事件代码	5485 0000 Hex	
内容	执行了立即停止 (MC_ImmediateStop) 指令					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴根据立即停止指令的输入变量 “StopMode(选择停止方法)” 的设定立即停止 相应轴在轴组动作时，其它轴根据 “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		正在发生轴轻度故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	执行了立即停止指令		-		-	
附加信息	无					
注意事项 / 备注	无					

事件名称	启动轴组立即停止指令			事件代码	54860000Hex	
内容	执行了轴组立即停止 (MC_GroupImmediateStop) 指令					
发生源	运动控制功能模块		发生源详情	轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴组中的轴根据轴参数的“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		正在发生轴组轻度故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	执行了轴组立即停止指令		-		-	
附加信息	无					
注意事项 / 备注	无					

事件名称	超出正方向软件限制			事件代码	64450000Hex	
内容	在轴动作过程中，位置超出了正方向软件限制					
发生源	运动控制功能模块		发生源详情	轴	检测时间	执行指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	根据“软件限制功能”的设定		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	位置超出了正方向软件限制		请检查超过软件限制的原因并进行恰当的处理		(用于检测非预期原因导致的超过软件限制，因此无需防止)	
附加信息	无					
注意事项 / 备注	变更正方向软件限制的设定时，请注意安全性					

事件名称	超出了负方向软件限制			事件代码	64460000Hex	
内容	在轴动作过程中，位置超出了负方向软件限制					
发生源	运动控制功能模块		发生源详情	轴	检测时间	执行指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	根据“软件限制功能”的设定		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	位置超出了负方向软件限制		请检查超过软件限制的原因并进行恰当的处理		(用于检测非预期原因导致的超过软件限制，因此无需防止)	
附加信息	无					
注意事项 / 备注	变更负方向软件限制的设定时，请注意安全性					

事件名称	超出到位检查时间			事件代码	6447 0000 Hex	
内容	在监控时间内未完成定位完成检查					
发生源	运动控制功能模块		发生源详情	轴	检测时间	执行指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴无法动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	定位完成需要花费很长时间		检查完成定位所需较长时间的原因, 并排除该原因。或调整伺服驱动器、到位检查时间、到位宽度。调整伺服驱动器时, 请增大环路增益。请注意控制在合适的范围内, 以免控制时发生振荡		尽可能排除定位动作随动性较差的原因和振动	
附加信息	无					
注意事项 / 备注	无					

事件名称	超过位置偏差			事件代码	6448 0000 Hex	
内容	指令当前位置和反馈当前位置的偏差超过了位置偏差超出值					
发生源	运动控制功能模块		发生源详情	轴	检测时间	执行指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴无法动作、相应轴在轴动作时减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	定位动作的随动性较差, 实际动作比指令延迟		请排除定位动作随动性较差的原因。再在正常范围内将“位置偏差超限值”变更为较大值		请尽可能排除定位动作随动性较差的原因	
附加信息	无					
注意事项 / 备注	无					

事件名称	即停输入			事件代码	6449 0000 Hex	
内容	即停输入变为 ON					
发生源	运动控制功能模块		发生源详情	轴	检测时间	常时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	根据“立即停止输入停止方法”的设定		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	检测到即停输入信号		请将立即停止输入信号设为 OFF		(用于检测立即停止输入, 因此无需防止)	
	即停信号的连接或即停输入的逻辑设定错误		如果立即停止输入信号 OFF 时仍发生异常, 则请修正立即停止信号的连接或立即停止输入的逻辑设定。请同时确认轴参数和从站侧的逻辑设定		请确认立即停止信号的连接或立即停止输入的逻辑设定是否正确。请同时确认轴参数和从站侧的逻辑设定	
附加信息	无					
注意事项 / 备注	解除异常时, 请将立即停止输入设为 OFF					

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					
发生源	运动控制功能模块		发生源详情	轴	检测时间	常时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	根据“极限输入停止方法”的设定		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	检测到正方向极限输入信号		请解除异常, 在不超过正方向极限的位置范围内向负方向执行复位动作。执行轴组动作指令过程中发生异常时, 请将禁用发生异常的轴组, 再执行上述复位动作。请检查超过极限的原因并进行恰当的处理		用于检测正方向极限输入, 因此无需防止。请编写程序, 以免超过正方向极限输入	
	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或正方向极限输入的逻辑设定 请同时确认轴参数和从站侧的逻辑设定		请确认正方向极限信号的连接或正方向极限输入的逻辑设定是否正确 请同时确认轴参数和从站侧的逻辑设定	
附加信息	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					
发生源	运动控制功能模块		发生源详情	轴	检测时间	常时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	根据“极限输入停止方法”的设定		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	检测到负方向极限输入信号		请解除异常, 在不超过负方向极限的位置范围内向正方向执行复位动作。执行轴组动作指令过程中发生异常时, 请将禁用发生异常的轴组, 再执行上述复位动作。请检查超过极限的原因并进行恰当的处理		用于检测负方向极限输入, 因此无需防止。请编写程序, 以免超过负方向极限输入	
	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或负方向极限输入的逻辑设定 请同时确认轴参数和从站侧的逻辑设定		请确认负方向极限信号的连接或负方向极限输入的逻辑设定是否正确 请同时确认轴参数和从站侧的逻辑设定	
附加信息	无					
注意事项 / 备注	无					

事件名称	位置偏差错误			事件代码	6456 0000 Hex	
内容	将指定的位置和反馈当前位置的差值转换为脉冲单位时，变为 30 位					
发生源	运动控制功能模块		发生源详情	轴	检测时间	常时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴伺服 OFF		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	在从轴中，指定轴的移动量超过了轴的最高速度，对指令当前位置执行限制，防止超过最高速度		请修正程序或电子齿轮比，以免从轴超过最高速度		请编写程序或设定电子齿轮比，以免从轴超过最高速度	
	从轴定位动作的随动性较差，实际动作比指令延迟		请排除从轴定位动作随动性较差的原因		请尽可能排除从轴定位动作随动性较差的原因	
附加信息	无					
注意事项 / 备注	无					

事件名称	伺服 OFF 异常			事件代码	6457 0000 Hex	
内容	因轴组发生异常，轴伺服 OFF					
发生源	运动控制功能模块		发生源详情	轴	检测时间	常时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴伺服 OFF		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	因轴组发生异常，轴伺服 OFF		请确认异常原因并进行处理		无	
附加信息	无					
注意事项 / 备注	作为连锁使轴发生本异常，以免轴组发生异常后伺服 OFF 的轴因 MC_Power(可运行) 指令再次伺服 ON。仅在“轴组停止方法”中选择“立即停止，同时执行伺服 OFF(停止自由运行)”时发生					

事件名称	无法计算绝对值编码器当前位置			事件代码	64580000Hex	
内容	不能通过断电保持的绝对值编码器信息正确恢复当前位置					
发生源	运动控制功能模块		发生源详情	轴	检测时间	电源接通时 / 控制器复位时 / 下载时 / 伺服 ON 开始时 / 从未使用轴切换至使用轴时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	<ul style="list-style-type: none"> <li>· 变更了单位转换设定、环计数器设定或伺服驱动器设定的环计数器设定</li> <li>· 复位位置超出了脉冲单位的带符号 40 位的范围</li> </ul>		请在解除异常后执行原点复位。请在安装了绝对值编码器的位置附近执行原点复位，以免复位位置超过带符号 40 位的范围		请在变更环形计数器上限设定值等位置相关的参数后重新执行原点复位。请在安装了绝对值编码器的位置附近执行原点复位，以免复位位置超过带符号 40 位的范围。编码器位置超过带符号 40 位的范围时，请勿执行 MC_Power(可运行) 指令、从未使用轴切换至使用轴或断开、接通电源	
附加信息	无					
注意事项 / 备注	无					

事件名称	未确定多轴协调动作中的原点			事件代码	64590000Hex	
内容	轴组动作中或减速停止中未确定逻辑轴的原点					
发生源	运动控制功能模块		发生源详情	轴组	检测时间	执行指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴组减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	轴组动作中或减速停止中，发生了逻辑轴的指令位置或反馈位置的溢出或下溢，未确定原点		请修正程序，以确保指令位置或反馈位置在不溢出或下溢的范围内动作		请编写程序，以确保指令位置或反馈位置在不溢出或下溢的范围内动作	
	轴组动作中或减速停止中，逻辑轴发生从站通信异常，未确定原点		请采取处理从站通信异常的措施并确定原点		无	
	轴组动作中或减速停止时，逻辑轴的从站脱离或禁用，且未确定原点		请重新加入脱离或禁用的从站并确定原点。		请在轴组动作或减速停止时使逻辑轴的从站脱离或禁用从站	
附加信息	无					
注意事项 / 备注	无					

事件名称	检测到伺服主电路电源 OFF			事件代码	74210000Hex	
内容	在伺服 ON 状态下，将伺服驱动器的主电路电源设为 OFF					
发生源	运动控制功能模块		发生源详情	轴	检测时间	伺服 ON 时始终
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴伺服 OFF		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	在伺服 ON 状态下，切断伺服驱动器的主电路电源		请接通发生本异常的轴的伺服驱动器主电路电源并解除异常后，再执行伺服 ON		请将伺服 OFF 后再关闭伺服驱动器的主电路电源	
附加信息	无					
注意事项 / 备注	无					

事件名称	无中断标准定位和中断信号			事件代码	74230000Hex	
内容	MC_MoveFeed( 中断标准定位 ) 指令执行中无中断输入					
发生源	运动控制功能模块		发生源详情	轴	检测时间	执行指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	锁定有效范围指定不恰当		如果指定相应指令的锁定有效范围错误，则请修正对锁定有效范围的指定		请根据动作与传感器位置的关系，指定恰当的锁定有效范围	
	中断信号配线错误		与相应指令对应的中断信号的接线错误时，请更正接线		请确认中断信号的接线无误	
	输出中断信号的传感器故障		如果非上述 2 种原因，则为需输出中断信号的传感器故障。请更换输出发生本异常的指令相应的中断信号的传感器等		无	
附加信息	无					
注意事项 / 备注	无					

事件名称	检测到原点复位反方向极限输入			事件代码	74240000Hex	
内容	原点复位动作中，检测到原点复位方向和反方向极限信号					
发生源	运动控制功能模块		发生源详情	轴	检测时间	执行指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	根据原点复位执行状态相应的停止方法停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	将正方向极限输入时动作及负方向极限输入时动作设定为“不反转”		如果极限输入时未发生异常，则请将正方向极限输入时动作和负方向极限输入时动作设定为“取反”		请确认事先记述在发生原因中的内容无误	
	原点复位时的输入信号传感器配置、原点复位设定、原点复位开始位置到达极限信号		请重新确认输入信号传感器的配置、原点复位的设定及原点复位开始位置后进行修正，以免达到极限信号			
	输入信号传感器接线错误、故障		请修正、更换输入信号传感器的接线			
附加信息	无					
注意事项/备注	无					

事件名称	检测到原点复位方向极限输入			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					
发生源	运动控制功能模块		发生源详情	轴	检测时间	执行指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	根据原点复位执行状态相应的停止方法停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	将正方向极限输入时动作及负方向极限输入时动作设定为“不反转”		如果极限输入时未发生异常，则请将正方向极限输入时动作和负方向极限输入时动作设定为“取反”		请确认事先记述在发生原因中的内容无误	
	原点复位时的输入信号传感器配置、原点复位设定、原点复位开始位置到达极限信号		请重新确认输入信号传感器的配置、原点复位的设定及原点复位开始位置后进行修正，以免达到极限信号			
	输入信号传感器接线错误、故障		请修正、更换输入信号传感器的接线			
附加信息	无					
注意事项/备注	无					

事件名称	检测到原点复位双向极限输入			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					
发生源	运动控制功能模块		发生源详情	轴	检测时间	执行指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	根据原点复位执行状态相应的停止方法停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	极限信号的接线错误		请修正极限信号的接线		请确认事先记述在发生原因中的内容无误	
	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“ON”			
	极限信号的接点逻辑错误		请修正极限信号的 a 触点或 b 触点			
极限传感器故障		请更换极限传感器				
附加信息	无					
注意事项 / 备注	无					

事件名称	检测到近原点、原点复位反方向极限输入			事件代码	74270000Hex	
内容	原点复位动作中，同时检测到近原点输入、原点复位方向和反方向的极限信号					
发生源	运动控制功能模块		发生源详情	轴	检测时间	执行指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	根据原点复位执行状态相应的停止方法停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	近原点信号、极限信号的接线错误		请修正近原点信号和极限信号的接线		请确认事先记述在发生原因中的内容无误	
	近原点传感器、极限传感器安装位置错误		请修正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“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和极限传感器的 a 触点或 b 触点			
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				
附加信息	无					
注意事项 / 备注	无					

事件名称	检测到近原点、原点复位方向极限输入			事件代码	74280000Hex	
内容	原点复位动作中，同时检测到近原点输入、原点复位方向的极限信号					
发生源	运动控制功能模块		发生源详情	轴	检测时间	执行指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	根据原点复位执行状态相应的停止方法停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	近原点信号、极限信号的接线错误		请修正近原点信号和极限信号的接线		请确认事先记述在发生原因中的内容无误	
	近原点传感器、极限传感器安装位置错误		请修正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“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和极限传感器的 a 触点或 b 触点			
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				
附加信息	无					
注意事项 / 备注	无					

事件名称	检测到原点输入、原点复位反方向极限输入			事件代码	74290000Hex	
内容	原点复位动作中，同时检测到原点输入、原点复位方向和反方向的极限信号					
发生源	运动控制功能模块		发生源详情	轴	检测时间	执行指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	根据原点复位执行状态相应的停止方法停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	原点输入信号、极限信号的接线错误		请修正原点输入信号和极限信号的接线		请确认事先记述在发生原因中的内容无误	
	原点输入传感器、极限传感器安装位置错误		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“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和极限传感器的 a 触点或 b 触点			
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				
附加信息	无					
注意事项 / 备注	无					

事件名称	检测到原点输入、原点复位方向极限输入			事件代码	742A 0000Hex	
内容	原点复位动作中，同时检测到原点输入、原点复位方向的极限信号					
发生源	运动控制功能模块		发生源详情	轴	检测时间	执行指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	根据原点复位执行状态相应的停止方法停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	原点输入信号、极限信号的接线错误		请修正原点输入信号和极限信号的接线		请确认事先记述在发生原因中的内容无误	
	原点输入传感器、极限传感器安装位置错误		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“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和极限传感器的 a 触点或 b 触点			
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				
附加信息	无					
注意事项 / 备注	无					

事件名称	原点输入屏蔽量错误			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，原点输入屏蔽量设定值不恰当					
发生源	运动控制功能模块		发生源详情	轴	检测时间	执行指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	根据原点复位执行状态相应的停止方法停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	原点复位动作模式为“指定附近避让、原点输入屏蔽距离”时，原点输入屏蔽距离的设定值由原点复位速度减速为原点复位接近速度时的移动量不足		请确认原点输入屏蔽量、原点复位速度、原点复位接近速度，并根据 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，以免减速时移动量不足		请事先确认 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，并设定原点输入屏蔽量、原点复位速度、原点复位接近速度，以免减速时移动量不足	
附加信息	无					
注意事项 / 备注	无					

事件名称	无原点输入			事件代码	742C0000Hex	
内容	原点复位中无原点信号输入。或在无原点确定输入的情况下检测到极限信号					
发生源	运动控制功能模块		发生源详情	轴	检测时间	执行指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	根据原点复位执行状态相应的停止方法停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	<ul style="list-style-type: none"> <li>· 原点复位中无原点信号输入</li> <li>· 或在无原点确定输入的情况下检测到极限信号</li> </ul>		请确认原点输入的设置、接线，并根据 MC_Home(原点复位) 指令或 MC_HomeWithParameter(参数指定原点复位) 指令的动作规格进行修正，以便在原点复位动作中输入原点信号。 请在输入原点后再检测极限信号		请设定为在原点复位动作时输入原点信号。 请在输入原点后再检测极限信号。 并请确认输入原点的接线无误	
附加信息	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	原点复位中无近原点信号输入					
发生源	运动控制功能模块		发生源详情	轴	检测时间	执行指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	根据原点复位执行状态相应的停止方法停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	设定为“有近原点信号”时，原点复位中无近原点信号输入		请确认近原点输入的设置、接线，并根据 MC_Home(原点复位) 指令或 MC_HomeWithParameter(参数指定原点复位) 指令的动作规格进行修正，以便在原点复位时输入近原点信号。		请设定为在原点复位动作时输入近原点信号。并请确认输入近原点的接线无误	
附加信息	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	检测到轴内分配的 EtherCAT 从站或 NX 单元异常					
发生源	运动控制功能模块		发生源详情	轴	检测时间	常时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴的伺服 OFF		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	轴内分配的 EtherCAT 从站或 NX 单元检测到异常		请确认从站发生的异常或由“确定从站异常代码”(94220000Hex) 通知的从站异常代码，并采取必要的处理措施		无	
附加信息	无					
注意事项 / 备注	无					

事件名称	轴组构成轴发生异常			事件代码	7430 0000 Hex	
内容	轴组构成轴发生了异常					
发生源	运动控制功能模块		发生源详情	轴组	检测时间	常时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	构成轴立即停止后会根据“轴组停止方法”设定的动作 否则，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		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	轴组动作中，构成轴发生了异常		请确认需构成轴组的轴的异常代码，并排除异常原因		无	
附加信息	无					
注意事项/备注	发生轴异常时，含该轴的轴组会无法动作					

事件名称	MC 通用异常发生中			事件代码	7433 0000 Hex	
内容	发生了 MC 通用异常					
发生源	运动控制功能模块		发生源详情	轴	检测时间	常时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	发生了部分停止故障等级的 MC 通用异常		请确认发生的 MC 通用异常，并排除该异常原因		无	
附加信息	无					
注意事项/备注	发生部分停止故障等级的 MC 通用异常时，轴和轴组无法动作					

事件名称	锁定位置溢出			事件代码	7434 0000 Hex	
内容	MC_TouchProbe(启用外部锁定)指令锁定的位置溢出					
发生源	运动控制功能模块		发生源详情	轴	检测时间	执行指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴减速停止 启用外部锁定指令无法获取锁定位置		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	MC_TouchProbe(启用外部锁定)指令锁定的位置溢出		请修正程序，以免轴位置溢出		请编写程序，以免轴位置溢出	
附加信息	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	MC_TouchProbe(启用外部锁定)指令锁定的位置下溢					
发生源	运动控制功能模块		发生源详情	轴	检测时间	执行指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴减速停止 启用外部锁定指令无法获取锁定位置		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	MC_TouchProbe(启用外部锁定)指令锁定的位置下溢		请修正程序,以免轴位置下溢		请编写程序,以免轴位置下溢	
附加信息	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	主轴向同步方向相反的方向持续移动					
发生源	运动控制功能模块		发生源详情	轴	检测时间	执行指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	与主轴和从轴同步方向相对,主轴持续向相反方向移动并溢出		请修正程序,以便同步开始后主轴的移动方向和移动量向着同步方向		请编写程序,以便同步开始后主轴的移动方向和移动量向着同步方向	
附加信息	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 期间分配给轴的 EtherCAT 从站或 NX 单元出现了以下任一操作 <ul style="list-style-type: none"> <li>· 脱离 / 更换</li> <li>· 禁用</li> <li>· NX 系列 CPU 单元的 NX 总线重启</li> </ul>					
发生源	运动控制功能模块		发生源详情	轴	检测时间	伺服 ON 时始终
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴伺服 OFF		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	伺服 ON 期间分配给轴的 EtherCAT 从站或 NX 单元出现了以下任一操作 <ul style="list-style-type: none"> <li>· 脱离 / 更换</li> <li>· 禁用</li> <li>· NX 系列 CPU 单元的 NX 总线重启</li> </ul>		请将分配给相应轴的 EtherCAT 从站或 NX 单元重新加入		请在伺服 OFF 后再对 EtherCAT 从站或 NX 单元执行以下操作 <ul style="list-style-type: none"> <li>· 脱离 / 更换</li> <li>· 禁用</li> <li>· NX 系列 CPU 单元的 NX 总线重启</li> </ul>	
附加信息	无					
注意事项/备注	无					

事件名称	标准距离溢出		事件代码	7438 0000 Hex		
内容	MC_MoveFeed( 中断标准定位 ) 指令中断输入后的目标位置溢出或下溢					
发生源	运动控制功能模块		发生源详情	轴	检测时间	执行指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因 ( 推测原因 )		处理措施		防止再次发生	
	将 MC_MoveFeed( 中断标准定位 ) 指令中断输入后的目标位置转换为脉冲数单位时, 超出了带符号 40 位的范围		请修正程序, 以使指令位置的输入值满足中断输入后的目标位置的脉冲数不超过范围的条件		请编写程序, 以确保将中断输入后的目标位置转换为脉冲数单位时不超过带符号 40 位的范围	
附加信息	无					
注意事项 / 备注	无					

事件名称	伺服驱动器的控制模式切换异常		事件代码	7439 0000 Hex		
内容	在切换控制模式时, 在规定的时间内未完成处理					
发生源	运动控制功能模块		发生源详情	轴	检测时间	执行指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴伺服 OFF		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因 ( 推测原因 )		处理措施		防止再次发生	
	停止 MC_SyncMoveVelocity( 周期同步速度控制 ) 指令时, 从向指令速度输出 0 开始的 10 秒内, 未将反馈当前速度在连续 3 个周期内控制在最高速度的 10% 以内		请调整指令和负载, 以免处于会发生异常的状态		请调整指令和负载, 以免处于会发生异常的状态	
	如果使用欧姆龙制伺服驱动器 1S 系列或 G5 系列, 则在停止 MC_TorqueControl ( 转矩控制 ) 指令时, 10 秒内未将反馈当前速度在连续 3 个周期内控制在最高速度的 10% 以内					
	切换控制模式 (CSP/CSV/CST) 后, 1 秒以内未完成伺服驱动器的模式切换		请确认伺服驱动器是否发生了异常或设定是否错误, 并进行必要的处理。需切换控制模式进行控制时, 请在 PDO 映射中设定 “CSP 用基准位置。”		请确认伺服驱动器未发生异常和设定无误。需切换控制模式进行控制时, 请在 PDO 映射中设定 “CSP 用基准位置。”	
附加信息	无					
注意事项 / 备注	无					

事件名称	主轴位置读取异常			事件代码	743A0000Hex	
内容	因同步指令的主轴位置发生异常，无法执行同步指令					
发生源	运动控制功能模块	发生源详情	轴	检测时间	启动指令时 / 执行指令时	
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴无法动作。在相应从轴的动作中减速停止		
系统定义变量	变量名称	数据类型		名称		
	_MC_AX[*].MFAultLvl.Active	BOOL		正在发生轴轻度故障		
发生原因及其处理	发生原因 (推测原因)	处理措施		防止再次发生		
	未确立同步控制指令主轴的 EtherCAT 过程数据通信，或 NX 单元的 I/O 数据无法用于控制	主轴的 EtherCAT 主站系统变量的 _EC_PDslavTbl(过程数据通信中从站表)为 FALSE 时，请检查主轴的异常并排除该异常原因 主轴为 NX 单元时，对 NX 单元的过程数据通信中状态，也请实施相同的处理措施		需在电源 ON、下载或解除从站通信异常后立即执行同步指令时，请在确认 EtherCAT 主站系统变量的 _EC_PDslavTbl(过程数据通信中从站表)的主轴节点为 TRUE 后，执行同步指令 主轴为 NX 单元时，对 NX 单元的过程数据通信中状态，也请实施相同的处理措施		
	同步指令的主轴发生从站脱离或禁用从站	确认主轴中是否执行了从站脱离或禁用从站，如果执行了上述操作，则请重新加入或启用从站		执行同步指令时，请勿执行主轴的从站脱离或禁用从站		
	同步指令的主轴中检测到无法计算绝对值编码器当前位置 (64580000Hex)	请确认主轴中是否检测到无法计算绝对值编码器当前位置 (64580000Hex)，检测到时请采取该异常的处理措施进行恢复		请勿执行将正在发生无法计算绝对值编码器当前位置 (64580000Hex) 的轴作为主轴的同步指令		
	同步指令主轴为未使用轴	请将主轴设为使用轴		执行同步指令时，请勿将主轴设为未使用轴		
附加信息	无					
注意事项 / 备注	无					

事件名称	辅助轴位置读取异常		事件代码	743B0000Hex	
内容	因同步指令的辅轴位置发生异常，无法执行同步指令				
发生源	运动控制功能模块	发生源详情	轴	检测时间	启动指令时 / 执行指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别
发生后的影响	用户程序	继续	动作	相应从轴无法动作。在相应从轴的动作中减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生
	未确立同步控制指令辅轴的 EtherCAT 过程数据通信，或 NX 单元的 I/O 数据无法用于控制		辅轴的 EtherCAT 主站系统变量的 _EC_PDSlavTbl(过程数据通信中从站表)为 FALSE 时，请调查辅轴的异常并排除该异常原因 辅轴为 NX 单元时，对 NX 单元的过程数据通信中状态，也请实施相同的处理措施		需在电源 ON、下载或解除从站通信异常后立即执行同步指令时，请在确认 EtherCAT 主站系统变量的 _EC_PDSlavTbl(过程数据通信中从站表)的辅轴节点为 TRUE 后，执行同步指令 辅轴为 NX 单元时，对 NX 单元的过程数据通信中状态，也请实施相同的处理措施
	同步指令的辅轴发生从站脱离或禁用		确认辅轴中是否执行了从站脱离或禁用从站，如果执行了上述操作，则请重新加入或启用从站		执行同步指令时，请勿执行辅轴的从站脱离或禁用从站
	同步指令的辅轴中检测到无法计算绝对值编码器当前位置 (64580000Hex)		请确认辅轴中是否检测到无法计算绝对值编码器当前位置 (64580000Hex)，检测到时请采取该异常的处理措施进行恢复		请勿执行将正在发生无法计算绝对值编码器当前位置 (64580000Hex) 的轴作为辅轴的同步指令
同步指令辅轴为未使用轴		请将辅轴设为使用轴		执行同步指令时，请勿将辅轴设为未使用轴	
附加信息	无				
注意事项 / 备注	无				

事件名称	从站通信异常		事件代码	8440 0000 Hex	
内容	与分配给轴的 EtherCAT 从站或 NX 单元之间的通信发生了异常				
发生源	运动控制功能模块	发生源详情	轴	检测时间	常时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别
发生后的影响	用户程序	继续	动作	相应轴伺服 OFF	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生
	与分配给轴的 EtherCAT 从站或 NX 单元之间的通信发生了异常		请通过事件日志确认 EtherCAT 主功能模块或 NX 总线功能模块发生的异常，并在排除异常原因后解除本异常		无
附加信息	无				
注意事项 / 备注	解除本异常但不解除分配给轴的从站或 NX 单元连接的 EtherCAT 主功能模块或 NX 总线功能模块的异常。未解除 EtherCAT 主功能模块或 NX 总线功能模块的异常也可解除本异常，但此时将继续保持轴的无效状态				

事件名称	运动控制异常解除指令数过多			事件代码	571D0000Hex *1	
内容	ResetMCErr(运动控制异常解除)指令的实例数超过了100个					
发生源	运动控制功能模块		发生源详情	MC 通用	检测时间	电源接通时 / 控制器复位时 / 下载时 / 在线编辑时
异常的属性	重要程度	监控信息	恢复方法	-	日志类别	系统
发生后的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.Obsr.Active		BOOL		正在产生 MC 通用监控信息	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	包含功能块内的实例在内, 在用户程序内宣布的 ResetMCErr(运动控制异常解除)指令中的实例数超过了100个		请修正用户程序, 以确保 ResetMC Error(运动控制异常解除)指令的实例数小于100个。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		请记述程序, 以将 ResetMCErr(运动控制异常解除)指令的实例数控制在100个以内	
附加信息	无					
注意事项 / 备注	无					

\*1. 单元版本 Ver.1.02 ~ Ver.1.09 的 CPU 单元发生的事件。

事件名称	位置偏差警告			事件代码	644C0000Hex	
内容	位置偏差超过了位置偏差警告值					
发生源	运动控制功能模块		发生源详情	轴	检测时间	执行指令时
异常的属性	重要程度	监控信息	恢复方法	-	日志类别	系统
发生后的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].Obsr.Active		BOOL		正在产生轴监控信息	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	定位动作的随动性较差, 实际动作比指令延迟		请排除定位动作随动性较差的原因。再在正常范围内将“位置偏差警告值”变更为较大值		请尽可能排除定位动作随动性较差的原因	
附加信息	无					
注意事项 / 备注	无					

事件名称	速度警告			事件代码	644D0000Hex	
内容	指令速度超过了速度警告值					
发生源	运动控制功能模块		发生源详情	轴 / 轴组	检测时间	执行指令时
异常的属性	重要程度	监控信息	恢复方法	-	日志类别	系统
发生后的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].Obsr.Active		BOOL		正在产生轴监控信息	
	_MC_GRP[*].Obsr.Active		BOOL		正在产生轴组监控信息	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	指令速度超过了速度警告值		请检查超过速度警告值的原因并进行恰当的处理。 再在正常范围内将“速度警告值”变更为较大值		(用于检测是否超过速度警告值, 因此无需防止)	
附加信息	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“相应轴或轴组减速停止”。					

事件名称	加速度警告			事件代码	644E0000Hex	
内容	指令加速度超过了加速度警告值					
发生源	运动控制功能模块		发生源详情	轴 / 轴组	检测时间	执行指令时
异常的属性	重要程度	监控信息	恢复方法	-	日志类别	系统
发生后的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].Obsr.Active		BOOL		正在产生轴监控信息	
	_MC_GRP[*].Obsr.Active		BOOL		正在产生轴组监控信息	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	指令加速度超过了加速度警告值		请检查超过加速度警告值的原因并进行恰当的处理。 再在正常范围内将“加速度警告值”变更为较大值		(用于检测是否超过加速度警告值, 因此无需防止)	
附加信息	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“相应轴或轴组减速停止”。					

事件名称	减速度警告			事件代码	644F0000Hex	
内容	指令减速度超过了减速度警告值					
发生源	运动控制功能模块		发生源详情	轴 / 轴组	检测时间	执行指令时
异常的属性	重要程度	监控信息	恢复方法	-	日志类别	系统
发生后的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].Obsr.Active		BOOL		正在产生轴监控信息	
	_MC_GRP[*].Obsr.Active		BOOL		正在产生轴组监控信息	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	指令减速度超过了减速度警告值		请检查超过减速度警告值的原因并进行恰当的处理。 再在正常范围内将“减速度警告值”变更为较大值		(用于检测是否超过减速度警告值, 因此无需防止)	
附加信息	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“相应轴或轴组减速停止”。					

事件名称	正方向转矩警告			事件代码	6450000Hex	
内容	转矩指令值超过了正方向转矩警告值					
发生源	运动控制功能模块		发生源详情	轴	检测时间	执行指令时
异常的属性	重要程度	监控信息	恢复方法	-	日志类别	系统
发生后的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].Obsr.Active		BOOL		正在产生轴监控信息	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	转矩指令值超过了正方向转矩警告值		请检查超过转矩警告值的原因并进行恰当的处理。 再在正常范围内将“正方向转矩警告值”变更为较大值		(用于检测是否超过转矩警告值,因此无需防止)	
附加信息	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“相应轴减速停止”。					

事件名称	负方向转矩警告			事件代码	64510000Hex	
内容	转矩指令值超过了负方向转矩警告值					
发生源	运动控制功能模块		发生源详情	轴	检测时间	执行指令时
异常的属性	重要程度	监控信息	恢复方法	-	日志类别	系统
发生后的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].Obsr.Active		BOOL		正在产生轴监控信息	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	转矩指令值超过了负方向转矩警告值		请检查超过转矩警告值的原因并进行恰当的处理。 再在正常范围内将“负方向转矩警告值”变更为较大值		(用于检测是否超过转矩警告值,因此无需防止)	
附加信息	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					
发生源	运动控制功能模块		发生源详情	轴	检测时间	常时
异常的属性	重要程度	监控信息	恢复方法	-	日志类别	系统
发生后的影响	用户程序	继续	动作	不更新位置,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		正在产生轴监控信息	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	线性模式时,将指令位置转换为脉冲单位时,超过了带符号40位的上限值		请修正程序或变更电子齿轮比的设定,以使指令位置的输入值满足相应指令下脉冲数不超过范围的条件。 需从溢出状态恢复时,请执行“变更当前位置”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,以免转换后的脉冲数超过带符号40位的范围	
附加信息	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“相应轴减速停止”。					

事件名称	指令位置下溢		事件代码	6453 0000 Hex	
内容	指令位置对应的脉冲数下溢				
发生源	运动控制功能模块	发生源详情	轴	检测时间	常时
异常的属性	重要程度	监控信息	恢复方法	-	日志类别
发生后的影响	用户程序	继续	动作	不更新位置, 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		正在产生轴监控信息	
发生原因及其处理	发生原因 (推测原因)	处理措施		防止再次发生	
	线性模式时, 将指令位置转换为脉冲单位时, 超过了带符号 40 位的下限值	请修正程序或变更电子齿轮比的设定, 以使指令位置的输入值满足相应指令下脉冲数不超过范围的条件。 需从下溢状态恢复时, 请执行“变更当前位置”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, 以免转换后的脉冲数超过带符号 40 位的范围	
附加信息	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“相应轴减速停止”。				

事件名称	反馈位置溢出		事件代码	6454 0000 Hex	
内容	反馈位置对应的脉冲数溢出				
发生源	运动控制功能模块	发生源详情	轴	检测时间	常时
异常的属性	重要程度	监控信息	恢复方法	-	日志类别
发生后的影响	用户程序	继续	动作	不更新位置, 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		正在产生轴监控信息	
发生原因及其处理	发生原因 (推测原因)	处理措施		防止再次发生	
	将反馈位置转换为脉冲单位时, 超出了带符号 40 位的上限值	请修正程序或变更电子齿轮比的设定, 以确保相应指令下反馈位置的脉冲数不超过范围, 且目标位置对于脉冲数的范围留有余量。 需从溢出状态恢复时, 请执行“变更当前位置”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, 以将转换后的脉冲数控制在带符号 40 位的范围并留有余量	
附加信息	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“相应轴减速停止”。				

事件名称	反馈位置下溢		事件代码	6455 0000 Hex	
内容	反馈位置对应的脉冲数下溢				
发生源	运动控制功能模块	发生源详情	轴	检测时间	常时
异常的属性	重要程度	监控信息	恢复方法	-	日志类别
发生后的影响	用户程序	继续	动作	不更新位置, 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		正在产生轴监控信息	
发生原因及其处理	发生原因 (推测原因)	处理措施		防止再次发生	
	将反馈位置转换为脉冲单位时, 超出了带符号 40 位的下限值	请修正程序或变更电子齿轮比的设定, 以确保相应指令下反馈位置的脉冲数不超过范围, 且目标位置对于脉冲数的范围留有余量。 需从下溢状态恢复时, 请执行“变更当前位置”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, 以将转换后的脉冲数控制在带符号 40 位的范围并留有余量	
附加信息	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“相应轴减速停止”。				

## 11 异常的确认和处理

事件名称	从站监控信息检测			事件代码	74320000Hex	
内容	检测到 EtherCAT 从站或 NX 单元的警告					
发生源	运动控制功能模块		发生源详情	轴	检测时间	常时
异常的属性	重要程度	监控信息	恢复方法	-	日志类别	系统
发生后的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].Obsr.Active		BOOL		正在产生轴监控信息	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	分配给轴的 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	
附加信息	附加信息 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“相应轴减速停止”。					

事件名称	无法执行凸轮表保存指令			事件代码	743C0000Hex	
内容	因其它操作正在访问非易失性存储器, 未能执行凸轮表文件保存					
发生源	运动控制功能模块		发生源详情	MC 通用	检测时间	启动指令时
异常的属性	重要程度	监控信息	恢复方法	-	日志类别	系统
发生后的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.Obsr.Active		BOOL		正在产生 MC 通用监控信息	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	因其它操作 (使用 Sysmac Studio 的传送、数据跟踪等) 正在访问非易失性存储器, 执行了 MC_SaveCamTable (保存凸轮表) 指令		请再次执行 MC_SaveCamTable (保存凸轮表) 指令		无	
附加信息	无					
注意事项 / 备注	无					

事件名称	到达合并中继速度之前的移动量不足通知			事件代码	94200000Hex	
内容	用于合并中继速度加减速的移动量不足					
发生源	运动控制功能模块		发生源详情	轴 / 轴组	检测时间	开始多重启动指令时
异常的属性	重要程度	监控信息	恢复方法	-	日志类别	系统
发生后的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].Obsr.Active		BOOL		正在产生轴监控信息	
	_MC_GRP[*].Obsr.Active		BOOL		正在产生轴组监控信息	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	“加减速超限”设定为“提高加减速速度 (将合并切换为等待)”时, 指定 Blending 时, 生成曲线的结果, 加减速超限, 因此变更为 Buffered 处理		无需变更为 Buffered 处理时, “加减速超限”请勿设定为“提高加减速速度 (将合并切换为等待)”		无需变更为 Buffered 处理时, “加减速超限”请勿设定为“提高加减速速度 (将合并切换为等待)”	
	指定 Blending 时, 因已到达目标位置, 不能正确绘制曲线, 因此变更为 Buffered 处理		变更为 Buffered 处理后出现意外动作时, 请修正程序以免发生左侧所述原因		变更为 Buffered 处理后出现意外动作时, 请编写程序以免发生左侧所述原因	
附加信息	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“相应轴减速停止”。					

事件名称	从 MC 试运转画面执行异常解除			事件代码	94210000Hex	
内容	从 Sysmac Studio 试运转画面执行了异常解除					
发生源	运动控制功能模块		发生源详情	MC 通用	检测时间	解除 MC 试运行异常时
异常的属性	重要程度	一般信息	恢复方法	-	日志类别	系统
发生后的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	无		-		-	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	从 Sysmac Studio 试运转画面执行了异常解除		-		-	
附加信息	附加信息 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”，由从站发出异常代码通知					
发生源	运动控制功能模块		发生源详情	轴	检测时间	发生“检测到从站异常”(742F0000Hex)后
异常的属性	重要程度	一般信息	恢复方法	-	日志类别	系统
发生后的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	无		-		-	
发生原因及其处理	发生原因 (推测原因)		处理措施		防止再次发生	
	因发生“从站异常检测”(742F0000Hex)，由从站发出异常代码通知		本异常为“检测到从站异常”(742F0000Hex)附带发生的信息。请确认从站异常代码(附加信息)，并采取所需的处理措施		无	
附加信息	附加信息 1: 从站异常代码					
注意事项 / 备注	欧姆龙制伺服驱动器 1S 系列或 G5 系列时，附加信息的后 2 位显示伺服驱动器的异常代码(错误显示 No. 的主编号)。例如，附加信息显示“FF13”时，伺服驱动器出现错误显示 No.13(主电路电源电压不足)。					

### 11-2-3 异常现象的推测原因和处理

本章对使用 MC 功能模块时或开始运行后发生的故障的处理方法进行说明。

#### 事先确认的项目

发生异常时，请确认以下项目后再进行检查。

分类	确认部位
安装环境	周围是否存在粉尘?
	周围是否存在可能混入的导电性异物(金属、碳等)?
	环境温度是否较高?(与标准使用环境温度相比)
	周围是否存在湿气?(使用湿度、水等)
	周围是否存在腐蚀性气体?(酸、盐、硫磺等)
	周围是否存在干扰源(焊接机、变频器等)?
接线	电源线和信号线是否在相同配管(线槽)中接线?
	是否切实接地?
	电源是否附带噪声滤波器?
变化	最近是否进行了增设作业(焊接作业)?
	最近是否增加了电源设备?
	是否对系统(含程序)进行了改造或新增等作业?
事故	附近是否遭受过雷击?
	是否发生过接地短路事故或漏电断路器断开?
	是否发生过停电?

#### 现象与措施

下面对组合使用欧姆龙制 1S 系列或 G5 系列伺服驱动器时的现象和措施进行说明。

进行非预期动作时，各种参数设定和凸轮数据可能无法从 Sysmac Studio 正确传送至 CPU 单元。

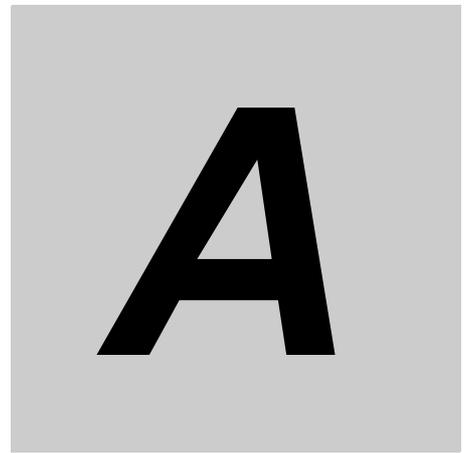
还可能是程序与 MC 功能模块之间无法顺利收发变量。

是否在正确的时间内收发变量请通过 Sysmac Studio 的数据跟踪功能进行确认。

现象	推测原因	确认部位	措施
电机不锁定	MC 功能模块未对伺服驱动器输出运行指令	请通过程序确认是否执行了 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
电机不转	启用伺服驱动器的驱动禁止输入	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。 请设定为不使用伺服驱动器的驱动禁止输入
	驱动器发生异常	请确认驱动器的异常	发生异常时请按照相应指示
	机械轴锁定	请确认是否达到了机械极限或是否存在机械卷入	请手动解除机械轴的锁定
	CPU 单元的故障	-	请更换 CPU 单元。

现象	推测原因	确认部位	措施
无法进行原点复位	发生异常	请确认异常内容	发生异常时请按照相应指示
	近原点输入接线错误	请根据轴变量的轴输入信息确认近原点输入传感器为 ON 或 OFF	请正确接线
	原点输入未正确接线	请确认原点输入的接线情况	请正确接线
	旋转方向和极限输入的方向不一致	如果极限处不取反而动作直至机械极限, 请根据轴变量的轴输入信息确认极限输入为 ON 或 OFF	请对极限输入正确接线
	极限输入接线错误	请确认极限输入的接线情况	请正确接线
	到位等待不变为 FALSE	请确认伺服驱动器的增益是否过低。 请确认到位宽度是否过窄	请增加伺服驱动器的增益。 请增加到位宽度
	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
	未正确设定轴参数	请通过 Sysmac Studio 确认轴参数	请正确设定轴参数后, 再下载至 MC 功能模块
	CPU 单元的故障	-	请更换 CPU 单元
复位后的原点有时会发生变化	联轴器等的机械松动	请通过标记油墨等对电机的轴、联轴器及其它机械连接部位进行标记, 确认有无偏移	请拧紧偏移的连接部位。
	Z 相无余量。 原点输入信号无余量	如果是伺服电机转 1 圈 (编码器转 1 圈的脉冲数) 的相近值或接近零的值, 则在传感器的导入时间发生细微变化后, 对电机转 1 圈的偏移进行原点复位时发生	请拆下电机的联轴器, 移动 1/4 圈左右后再次尝试进行原点复位, 以使 Z 相余量的脉冲为伺服电机转 1 圈 (编码器转 1 圈的脉冲数) 的 1/2 左右
电机转动不稳定	伺服电机的电机动力线或编码器线接线错误或缺相等	请确认电机动力线和编码器线的接线情况	请正确接线
	忘记拧紧连接电机轴和机械系统的联轴器的偏心螺丝, 负载转矩因滑轮和齿轮的啮合而发生变动	请检查机械。 请设为无负载 (从联轴器上拆下之前的机械) 后再尝试运行。	请重新检查并调整机械
	增益调整不足	-	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 (确认转矩和惯量比)	请变更为最合适的电机
	伺服电机的电机轴承损坏	请切断伺服驱动器的电源, 对于带制动器的电机, 请接通制动器电源, 再解除制动器, 并拆下电机的动力线 (有时可能是由于动态制动器的原因), 尝试用手转动电机的输出轴	请更换伺服电机
	伺服电机的电机绕组断线	请通过万用表确认电机动力线 U/V/W 间的电阻值。 如果不均衡, 则为异常	请更换伺服电机
旋转方向为反方向	伺服驱动器设定为反转	按微动移动动作, 如果与微动移动方向相反, 则伺服驱动器的旋转方向设定为反方向。请确认反馈信号 (A/B 相) 是否相反, 参数中是否设定了反转。	请重新设定伺服驱动器的旋转方向
	(原点复位时) 需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	请重新设定正确的轴参数
	(原点复位时) 近原点输入接线错误	请根据轴变量的轴输入信息确认近原点输入传感器为 ON 或 OFF	请对近原点输入正确接线

现象	推测原因	确认部位	措施
无法启动, 无法完成定位或延迟	到位宽度过窄, 无法完成定位 (未在到位宽度内时无法完成上一个动作, 因此无法启动下一个动作)	-	请增加到位宽度
	伺服驱动器的增益过低	-	请调整伺服驱动器的增益
	停止时受到外力作用, 不在到位宽度内	请通过轴变量的轴输入信息, 确认指令当前位置与反馈当前位置之差是否在到位宽度内	不在到位宽度内停止 (如接触控制等) 时, 可使用偏差计数器复位, 强制完成定位
发出异响	机械振动	请检查机械的活动部位是否有异物渗入、损坏、变形、松动	请修复故障部位
	伺服驱动器的增益调整不足 (增益过高)	-	请执行自动调谐。 请手动调整 (减少) 增益
	电机选择错误 (无法调整)	请重新选择电机 (确认转矩和惯量比)	请变更为最合适的电机
	连接电机轴和机械的联轴器偏心	-	请调整电机或机械的安装
电机轴晃动	增益调整不足 (增益过低)	-	请执行自动调谐。 请手动调整 (增加) 增益
	机械刚性较弱, 因此无法调整增益	尤其发生在对垂直轴、SCARA 机器人、堆垛机等轴施加弯曲、扭曲负载的系统中	请增加机械刚性。 请重新调整增益
	机械构造发生粘滑 (高粘性的静摩擦)	-	请执行自动调谐。 请手动调整增益
	电机选择错误 (无法调整)	请重新选择电机 (确认转矩和惯量比)	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	原点从定位前的位置偏移	请参阅“复位后的原点有时会发生变化”一栏	请参阅“复位后的原点有时会发生变化”一栏
	因受到焊接机和变频器等干扰而发生误动作	请确认附近有无焊接机和变频器等	请远离焊接机和变频器等
	机械系统的偏移	请确认偏移是否累计 (对机械的连接部位进行标记并确认偏移)	请拧紧机械需紧固的部位
无法使用 Sysmac Studio 的 MC 试运行功能	其它 Sysmac Studio 正在使用 MC 试运行功能	请确认相同 CPU 单元是否连接了其它支持设备	请关闭其它 Sysmac Studio 的 MC 试运行功能



# 附录

A

本附录中记载了欧姆龙制 1S 系列及 G5 系列伺服驱动器的对象设定值及连接方法。

---

A-1 与伺服驱动器 1S 系列之间的连接 .....	A-2
A-1-1 与伺服驱动器之间的接线 .....	A-2
A-1-2 伺服驱动器的设定 .....	A-2
A-2 与伺服驱动器 G5 系列之间的连接 .....	A-10
A-2-1 与伺服驱动器之间的接线 .....	A-10
A-2-2 伺服驱动器的设定 .....	A-10
A-3 与编码器输入终端之间的连接 .....	A-19
A-3-1 与编码器输入终端之间的接线 .....	A-19
A-3-2 编码器输入终端的设定 .....	A-19
A-4 与 NX 单元之间的连接 .....	A-25
A-5 PDS 状态切换 .....	A-26
A-5-1 PDS 状态控制方式 .....	A-27
A-5-2 主电路电源 OFF 检测 .....	A-28
A-6 用语说明 .....	A-29
A-6-1 NJ/NX 系列通用 .....	A-29
A-6-2 运动控制功能 .....	A-30
A-6-3 EtherCAT 通信 .....	A-31
A-7 版本相关信息 .....	A-32

## A-1 与伺服驱动器 1S 系列之间的连接

下面对与欧姆龙制 1S 系列伺服驱动器 EtherCAT 通信内置型之间的连接进行说明。

### A-1-1 与伺服驱动器之间的接线

通过 EtherCAT 通信与伺服驱动器连接。

关于连接方法，请参阅 □ “NJ/NX 系列 CPU 单元内置 EtherCAT 端口 用户手册 (SBCE-358)”。

### A-1-2 伺服驱动器的设定

下面，对连接作为 MC 功能模块控制对象的欧姆龙制伺服驱动器 1S 系列 (EtherCAT 通信内置型) 时的伺服驱动器侧的设定概要进行说明。

关于伺服驱动器的详情，请参阅 □ “AC 伺服电机 / 驱动器 1S 系列用户手册 (EtherCAT 通信内置型) (SBCE-377)”。

## 外部输入信号的分配

在 MC 功能模块中，使用伺服驱动器侧的下列输入信号。

- 即停输入
- 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 1S 系列伺服驱动器的出厂设定中，如下表所示分配了各输入信号。

信号名称	输入信号
即停输入	伺服驱动器的通用输入 1(IN1: CN1 连接器针脚 No.12 b 触点)* <sup>1</sup>
正方向极限输入	伺服驱动器的通用输入 2(IN2: CN1 连接器针脚 No.32 b 触点)* <sup>2</sup>
负方向极限输入	伺服驱动器的通用输入 3(IN3: CN1 连接器针脚 No.13 b 触点)* <sup>3</sup>
近原点输入	伺服驱动器的通用输入 4(IN4: CN1 连接器针脚 No.33 a 触点)

\*1. 伺服驱动器侧的信号名称为“异常停止输入”。

\*2. 伺服驱动器侧的信号名称为“正转侧驱动禁止输入”。

\*3. 伺服驱动器侧的信号名称为“反转侧驱动禁止输入”。

## ● 外部锁定功能的触发信号分配

在 1S 系列伺服驱动器的出厂设定中，在外部锁定功能的触发信号中分配了下表所示的输入信号。

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 Z 相
	1: mcEXT	1: mcLatch1	伺服驱动器的通用输入 7 (IN7: CN1 连接器的针脚 No.15 a 触点)* <sup>1</sup>
		2: mcLatch2	伺服驱动器的通用输入 8 (IN8: CN1 连接器的针脚 No.25 a 触点)* <sup>2</sup>
1: mcController	-	-	通过 TriggerVariable(触发变量)指定的变量

\*1. 伺服驱动器侧的信号名称为“外部锁定输入 1”。

\*2. 伺服驱动器侧的信号名称为“外部锁定输入 2”。

## 背隙补偿功能

MC 功能模块不具备背隙补偿功能。

需要背隙补偿时，使用伺服驱动器侧的补偿功能。

需要在伺服驱动器侧设定的对象如下所示。

索引	子索引	名称	说明
3001Hex	-	机械	-
	02Hex	背隙补偿选择	设定位置控制时的背隙补偿的有效/无效及补偿时的动作方向。* <sup>1</sup> 0: 背隙补偿无效 1: 伺服 ON 后, 首个正转侧动作时补偿。 2: 伺服 ON 后, 首个反转侧动作时补偿。
	03Hex	背隙补偿量	设定位置控制时的背隙补偿量。
	04Hex	背隙补偿时间常数	设定位置控制时的背隙补偿时间常数。

\*1. 出厂设定为“0: 背隙补偿无效”。

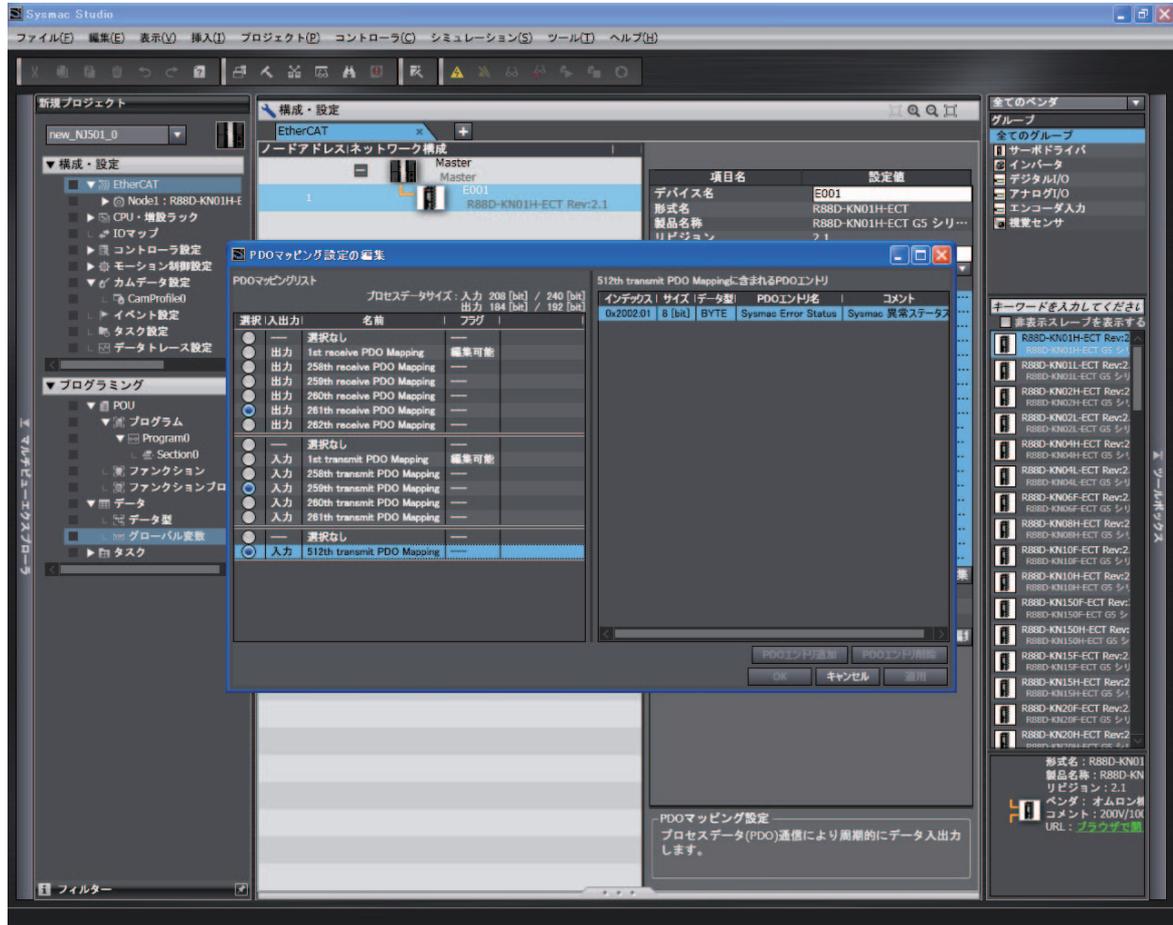
关于背隙功能的详情，请参阅 □□ “AC 伺服电机 / 驱动器 1S 系列 用户手册 (EtherCAT 通信内置型) (SBCE-377)”。

## PDO 映射

下面对作为 MC 功能模块的伺服轴进行控制的 PDO 映射进行说明。

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

通过 Sysmac Studio 的 [EtherCAT] 编辑画面中的 [PDO 映射设定的编辑] 选择将要使用的 PDO 映射。



Sysmac Studio 默认选择以下 PDO 映射。

RxPDO: [ 261th 接收 PDO 映射 ] (1704Hex)	控制字 (6040h)、目标位置 (607Ah)、目标速度 (60FFh)、 目标转矩 (6071h)、操作模式 (6060h)、锁定功能 (60B8h)、 最大曲线速度 (607Fh)、正转侧转矩限制 (60E0h)、 反转侧转矩限制 (60E1h)
TxPDO: [ 259th 发送 PDO 映射 ] (1B02Hex)	错误代码 (603Fh)、状态字 (6041h)、当前位置 (6064h)、 当前转矩 (6077h)、操作模式显示 (6061h)、 锁定状态 (60B9h)、锁定位置 1(60BAh)、锁定位置 2(60BCh)、 数字输入 (60FDh)



### 参考

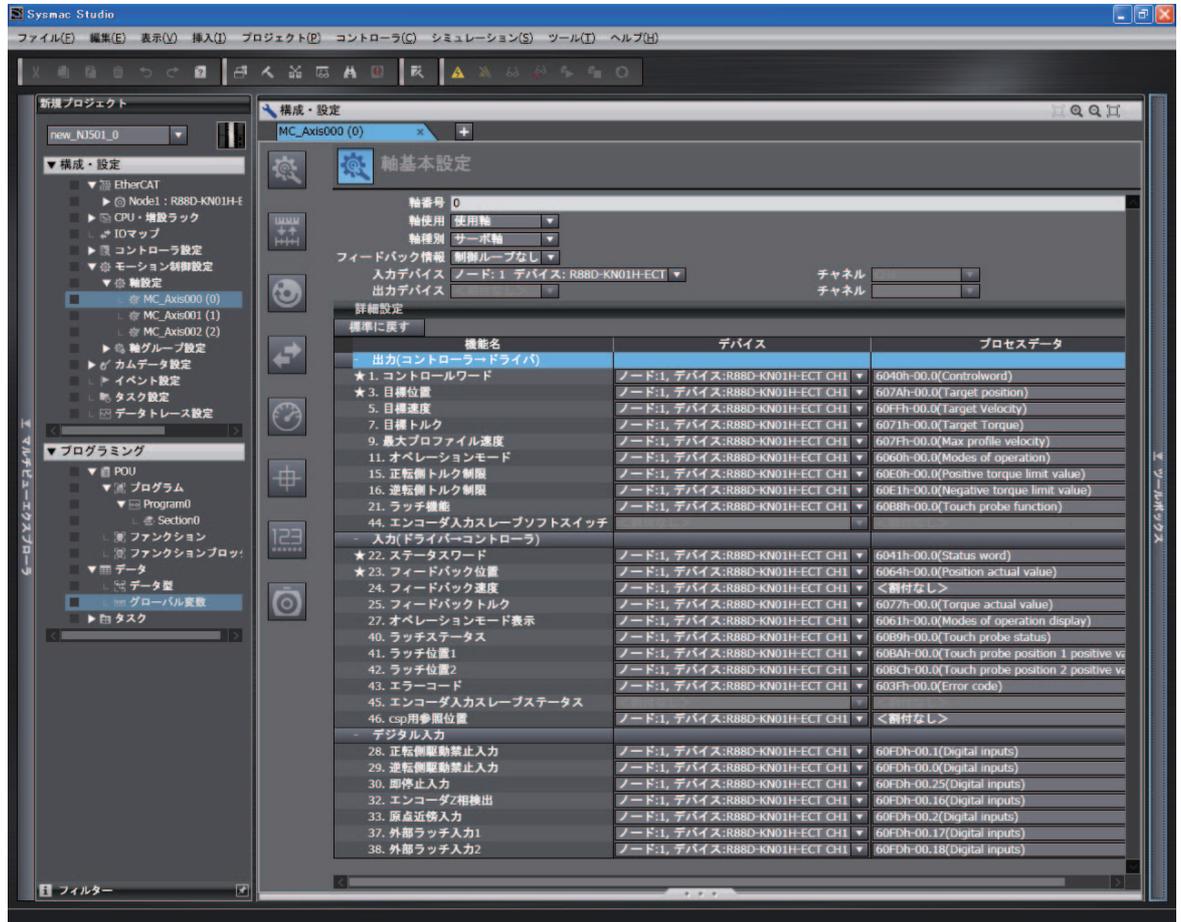
使用欧姆龙制推荐伺服驱动器 R88D-1SN □□□ -ECT 时，PDO 映射可直接使用 Sysmac Studio 默认的映射。

## MC 功能模块的功能和过程数据之间的关系

MC 功能模块具有的功能与过程数据的对象信息相关。

根据 MC 功能模块使用的 EtherCAT 从站构成及使用功能的不同，有时需要变更这些关系。

点击 Sysmac Studio 的 [ 轴基本设定 ] 画面中的 [ 高级设置 ]，打开设定画面。



### 参考

使用欧姆龙制推荐伺服驱动器 R88D-1SN □□□ -ECT 时，功能和过程数据之间的关系可直接使用 Sysmac Studio 默认设置。

## ● 输出 ( 控制器→驱动器 ) 的设定

对从 MC 功能模块输出到伺服驱动器的指令数据进行设定。

下列所示为 Sysmac studio 的默认设定。(带★标记的为所需对象)

功能名称	过程数据	说明
★ 控制字	6040h-00.0 (Controlword)	控制伺服驱动器状态的数据。 请设定为 [6040h: Controlword]。
★ 目标位置	607Ah-00.0 (Target position)	位置控制时的目标位置。 请设定为 [607Ah: Target position]。
目标速度	60FFh-00.0 (Target velocity)	速度控制时的目标速度。 在周期同步速度模式下, 向驱动器输出 MC_SyncMoveVelocity (周期同步速度控制) 指令等时需要该功能。 通常, 请设定为 [60FFh: Target velocity]。
目标转矩	6071h-00.0 (Target torque)	转矩控制时的目标转矩。 在周期同步转矩模式下, 向驱动器输出 MC_TorqueControl(转矩控制) 指令等时需要该功能。 通常请设定为 [6071h: Target torque]。
最大曲线速度	607Fh-00.0 (Max profile velocity)	转矩控制时的速度限制值。 在周期同步转矩模式下, 发出 MC_TorqueControl(转矩控制) 指令等, 进行速度限制时需要该功能。 通常请设定为 [607F: Max profile velocity]
操作模式	6060h-00.0 (Modes of operation)	切换控制模式的数据。 发出 MC_SyncMoveVelocity(周期同步速度控制) 指令及 MC_TorqueControl(转矩控制) 指令等, 切换为周期同步位置模式之外的控制模式时需要该功能。 通常请设定为 [6060h: Modes of operation]。*1
正转侧转矩限制	60E0h-00.0 (Positive torque limit value)	正方向侧的转矩限制值。 发出 MC_SetTorqueLimit(转矩限制) 指令等, 限制驱动器的输出转矩时需要该功能。 通常请设定为 [60E0h: Positive torque limit value]。
反转侧转矩限制值	60E1h-00.0 (Negative torque limit value)	负方向侧的转矩限制值。 发出 MC_SetTorqueLimit(转矩限制) 指令等, 限制驱动器的输出转矩时需要该功能。 通常请设定为 [60E1h: Negative torque limit value]。
锁定功能	60B8h-00.0 (Touch probe function)	控制锁定功能的数据。 发出 MC_Home(原点复位) 指令、MC_HomeWithParameter(参数指定原点复位) 指令、MC_MoveFeed(中断标准定位) 指令、MC_TouchProbe(启用外部锁定) 指令、MC_MoveLink(梯形模式凸轮) 指令等, 使用锁定功能时需要该功能。 通常请设定为 [60B8h: Touch probe function]。

\*1. 设定为 [6060h: Modes of operation] 时, 也请设定为 [6061h: Modes of operation display]。只设定其中一个将无法正常动作。



### 使用注意事项

- 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按照意图进行动作。
- 未使用欧姆龙制伺服驱动器 1S 系列 EtherCAT 通信内置型或 G5 系列 EtherCAT 通信内置型时, 请务必设定操作模式 (6060Hex)。

## ● 输入 ( 驱动器 → 控制器 ) 的设定

从伺服驱动器输出到 MC 功能模块的状态数据。

下列所示为 Sysmac studio 的默认设定。(带★标记的为所需对象)

功能名称	过程数据	说明
★ 状态字	6041h-00.0 (Statusword)	伺服驱动器的状态。请设定为 [6041h: Statusword]。
★ 当前位置	6064h-00.0 (Position actual value)	显示当前位置。请设定为 [6064h: Position actual value]。
当前速度	< 无设定 > *1	显示当前速度。使用时，通常设定为 [606Ch: Velocity actual value]。
当前转矩	6077h (Torque actual value)	显示当前转矩。 在周期同步转矩模式下，向驱动器输出 MC_TorqueControl(转矩控制) 指令等时需要该功能。 通常请设定为 [6077h: Torque actual value]。
操作模式显示	6061h-00.0 (Modes of operation display)	显示控制模式。 发出 MC_SyncMoveVelocity(周期同步速度控制) 指令及 MC_TorqueControl(转矩控制) 指令等，切换为周期同步位置模式之外的控制模式时需要该功能。 通常请设定为 [6061h: Modes of operation display]。*2
锁定状态	60B9h-00.0 (Touch probe status)	显示锁定功能的状态。 发出 MC_Home(原点复位) 指令、MC_HomeWithParameter(参数指定原点复位) 指令、MC_MoveFeed(中断标准定位) 指令、MC_TouchProbe(启用外部锁定) 指令、MC_MoveLink(梯形模式凸轮) 指令等，使用锁定功能时需要该功能。 通常请设定为 [60B9h: Touch probe status]。
锁定位置 1	60BAh-00.0 (Touch probe pos1 pos value)	通过锁定功能 1 锁定的位置。 发出 MC_Home(原点复位) 指令、MC_HomeWithParameter(参数指定原点复位) 指令、MC_MoveFeed(中断标准定位) 指令、MC_TouchProbe(启用外部锁定) 指令、MC_MoveLink(梯形模式凸轮) 指令等，使用锁定功能时需要该功能。通常请设定为 [60BAh: Touch probe pos1 pos value]。
锁定位置 2	60BCh-00.0 (Touch probe pos2 pos value)	通过锁定功能 2 锁定的位置。 发出 MC_Home(原点复位) 指令、MC_HomeWithParameter(参数指定原点复位) 指令、MC_MoveFeed(中断标准定位) 指令、MC_TouchProbe(启用外部锁定) 指令、MC_MoveLink(梯形模式凸轮) 指令等，使用锁定功能时需要该功能。通常请设定为 [60BCh: Touch probe pos2 pos value]。
错误代码	603Fh-00.0 (Error code)	驱动器内的异常代码。 通常请设定为 [603Fh: Error code]。
csp 用基准位置	无设定	csp 模式切换时的参照位置。使用速度控制模式 (CSV) 的指令及使用转矩控制模式 (CST) 的指令时参照的数据。 欧姆龙制伺服电机 / 驱动器 1S 系列中可使用。*3

\*1. 需要时，将所选的过程数据映射至 PDO 后再进行设定。606Ch-00.0(Velocity actual value) 为标准设定。

\*2. 设定为 [6061h: Modes of operation display] 时，也请设定为 [6060h: Modes of operation]。只设定其中一个将无法正常动作。

\*3. 使用时，请将 [3010Hex-87Hex: Reference Position for csp] 映射至 PDO。



### 使用注意事项

- 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按照意图进行动作。
- 未使用欧姆龙制伺服驱动器 1S 系列 EtherCAT 通信内置型或 G5 系列 EtherCAT 通信内置型时，请务必设定操作模式显示 (6061Hex)。
- 使用欧姆龙制伺服驱动器 1S 系列，并通过 MC\_SyncMoveVelocity(周期性同步速度控制) 指令切换控制模式时，请映射“csp 用基准位置”。

## ● 数字输入的设定

MC 功能模块使用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	60FDh-00.1 (Digital inputs)	用作正方向极限输入的信号。 通常请设定为 [60FDh-00: Digital inputs] 中的 [位 1: Positive limit switch]。
反转侧驱动禁止输入	60FDh-00.0 (Digital inputs)	用作负方向极限输入的信号。 通常请设定为 [60FDh-00: Digital inputs] 中的 [位 0: Negative limit switch]。
即停输入	60FDh-00.25 (Digital inputs)	用作即停输入的信号。 欧姆龙制伺服驱动器 1S 系列, 请设定为 [60FDHex-00: Digital inputs] 中的 [位 25: Error Stop Input]。
编码器 Z 相检测	60FDh-00.16 (Digital inputs)	Z 相输入的检测状态显示。 欧姆龙制伺服电机 / 驱动器 1S 系列, 请设定为 [60FDHex-00: Digital inputs] 中的 [位 16: Encoder Phase Z Detection]。
近原点输入	60FDh-00.2 (Digital inputs)	用作近原点输入的信号。 通常请设定为 [60FDh-00: Digital inputs] 中的 [位 2: Home switch]。
外部锁定输入 1	60FDh-00.17 (Digital inputs)	用作外部锁定输入 1 的信号状态显示。 欧姆龙制伺服驱动器 1S 系列中, 请设定为 [60FDh-00: Digital inputs] 中的 [位 17: External Latch Input 1]。
外部锁定输入 2	60FDh-00.18 (Digital inputs)	用作外部锁定输入 2 的信号状态显示。 欧姆龙制伺服驱动器 1S 系列中, 请设定为 [60FDh-00: Digital inputs] 中的 [位 18: External Latch Input 2]。



### 使用注意事项

- 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按照意图进行动作。



### 版本相关信息

- Ver.1.09 以下版本的 CPU 单元, 如未使用欧姆龙制伺服驱动器 1S 系列 EtherCAT 通信内置型, 需映射操作模式 (6060Hex) 及操作模式显示 (6061Hex)。
- Ver.1.10 以上版本的 CPU 单元, 根据有无操作模式 (6060Hex) 及操作模式显示 (6061Hex) 映射, 动作如下。

	操作模式显示	
	(6061Hex) 有	(6061Hex) 无
操作模式 (6060Hex) 有	<ul style="list-style-type: none"> <li>· 使用 CSP *1、CSV *2、CST *3 的指令可执行。</li> <li>· CSP、CSV、CST 以外, 将会伺服 OFF。</li> </ul>	<ul style="list-style-type: none"> <li>· 使用 CSP 的指令可执行。执行使用其他控制模式的指令时发生“过程数据对象设定不足 (错误代码: 3461Hex)”。</li> <li>· MC 功能模块将伺服驱动器的控制模式视为 CSP。向伺服驱动器发送 CSP 指令。</li> </ul>
操作模式 (6060Hex) 无	<ul style="list-style-type: none"> <li>· 使用 CSP 的指令可执行。执行使用其他控制模式的指令时发生“过程数据对象设定不足 (错误代码: 3461Hex)”。</li> <li>· CSP 以外, 将会伺服 OFF。</li> </ul>	使用 CSP 的指令可执行。执行使用其他控制模式的指令时发生“过程数据对象设定不足 (错误代码: 3461Hex)”。

\*1. 伺服驱动器控制模式中的周期性同步位置模式 (CSP)。

\*2. 伺服驱动器控制模式中的周期性同步速度模式 (CSV)。

\*3. 伺服驱动器控制模式中的周期性同步转矩模式 (CST)。

## 伺服驱动器侧的对象设定

使用 MC 功能模块控制功能的欧姆龙制伺服驱动器 1S 系列侧的设定如下。

使用伺服驱动器的各种功能时，请按照伺服驱动器手册设定各功能相关的对象。

索引	子索引	名称	推荐设定值	说明
3001Hex	-	机械	-	设伺服侧齿轮比为 1:1，在控制器侧设定用户单位。 <sup>*1</sup>
	05Hex	电子齿轮比分子	1	
	06Hex	电子齿轮比分母	1	
3330Hex	-	转矩限制	-	PCL 与 NCL 在 OFF 状态下受限于“60E0Hex”与“60E1Hex”的值。
	01Hex	切换选择	2	
3A00Hex	-	原点复位	-	以“0”使用伺服侧的偏置值。
	06Hex	编码器原点位置偏置	0	
3B10Hex	-	驱动禁止	-	伺服侧的驱动禁止输入设为无效，在控制器侧处理。
	01Hex	有效	0	
3B11Hex	-	软件限制	-	正方向、负方向的软件限位均设为无效。
	01Hex	有效选择	0	
3B30Hex	-	锁定功能 1	-	设为 Touch probe1= 外部锁定信号 1、Touch probe2= 外部锁定信号 2。
	01Hex	锁定 1 触发选择	1	
3B31Hex	-	锁定功能 2	-	设为 Touch probe1= 外部锁定信号 1、Touch probe2= 外部锁定信号 2。
	01Hex	锁定 2 触发选择	2	
4020Hex	-	警告定制	-	设为消除警告原因时自动复位。
	04Hex	警告保持选择	0	
4510Hex	-	编码器	-	用作绝对值编码器。 忽略多圈计数器超限。
	01Hex	绝对值编码器动作选择	2	
4630Hex	-	正方向驱动禁止	-	按负逻辑 (b 接点) 向通用输入 2(IN2) 分配正方向驱动禁止。
	01Hex	端口选择	2	
	02Hex	逻辑选择	1	
4631Hex	-	负方向驱动禁止	-	按负逻辑 (b 接点) 向通用输入 3(IN3) 分配负方向驱动禁止。
	01Hex	端口选择	3	
	02Hex	逻辑选择	1	
4632Hex	-	外部锁定输入 1	-	按正逻辑 (a 接点) 向通用输入 7(IN7) 分配外部锁定输入 1。
	01Hex	端口选择	7	
	02Hex	逻辑选择	0	
4633Hex	-	外部锁定输入 2	-	按正逻辑 (a 接点) 向通用输入 8(IN8) 分配外部锁定输入 2。
	01Hex	端口选择	8	
	02Hex	逻辑选择	0	
4634Hex	-	近原点输入	-	按正逻辑 (a 接点) 向通用输入 4(IN4) 分配近原点输入。
	01Hex	端口选择	4	
	02Hex	逻辑选择	0	

\*1. 单元版本 1.10 以下的 CPU 单元，无法按最高转速运行欧姆龙制伺服电机 1S 系列。要按最高转速运行 1S 系列，请将电子齿轮比设为 2:1 以上。

## A-2 与伺服驱动器 G5 系列之间的连接

下面对与欧姆龙制伺服驱动器 G5 系列 EtherCAT 通信内置型之间的连接进行说明。

### A-2-1 与伺服驱动器之间的接线

通过 EtherCAT 通信与伺服驱动器连接。

关于连接方法，请参阅 □ “NJ/NX 系列 CPU 单元内置 EtherCAT 端口 用户手册 (SBCD-358)”。

### A-2-2 伺服驱动器的设定

下面对连接作为 MC 功能模块控制对象的欧姆龙制伺服驱动器 G5 系列 EtherCAT 通信内置型时的伺服驱动器侧的设定概要进行说明。

伺服驱动器的详情请参阅 □ “AC 伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置型 用户手册 (SBCE-365)” 及 □ “AC 伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型 用户手册 (SBCE-366)”。

### 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

- 也可使用非推荐对象的欧姆龙制伺服驱动器 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侧规格的差异而有所不同。请设定 Sysmac Studio 中使用的功能和对象词典。
- 欧姆龙制伺服驱动器 G5 系列的 R88D-KN □□□ -ECT-R(单元版本 Ver.1.0)是仅适用于位置控制(cyclic synchronous position)的伺服驱动器。
- 可用于无需速度控制(cyclic synchronous velocity)或转矩控制(cyclic synchronous torque)的应用中。
- 功能详情请参阅 □ “AC 伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置型 用户手册 (SBCE-360)”。
- 欧姆龙制伺服电机 / 驱动器 G5 系列中，使用气缸型 Ver.2.0 以下的单元版本时，请将节点地址开关设为非“00”。如果设定为“00”，网络构成将出现异常。
- 因欧姆龙制伺服驱动器 G5 系列的单元版本导致的差异，请参阅 □ “AC 伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置型 用户手册 (SBCE-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下列输入信号使用。

- 即停输入
- 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

在欧姆龙制伺服驱动器 G5 系列的出厂设定中，如下表所示分配了各输入信号。

信号名称	输入信号
即停输入	伺服驱动器的通用输入 1(IN1: CN1 连接器针脚 No.5 b 触点)
正方向极限输入	伺服驱动器的通用输入 2(IN2: CN1 连接器针脚 No.7 b 触点) *1
负方向极限输入	伺服驱动器的通用输入 3(IN3: CN1 连接器针脚 No.8 b 触点) *2
近原点输入	伺服驱动器的通用输入 4(IN4: CN1 连接器针脚 No.9 a 触点)

\*1. 伺服驱动器侧的信号名称为“正转侧驱动禁止输入”。

\*2. 伺服驱动器侧的信号名称为“反转侧驱动禁止输入”。

### ● 外部锁定功能的触发信号分配

在欧姆龙制伺服驱动器 G5 系列的出厂设定中，在外部锁定功能的触发信号中分配了如下表所示的输入信号。

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 Z 相
	1: mcEXT	1: mcLatch1	伺服驱动器的通用输入 7 (IN7: CN1 连接器针脚 No.12 a 触点) *1
		2: mcLatch2	伺服驱动器的通用输入 6 (IN6: CN1 连接器针脚 No.11 a 触点) *2
1: mcController	-	-	通过 TriggerVariable( 触发变量 ) 指定的变量

\*1. 伺服驱动器侧的信号名称为“外部锁定输入 1”。

\*2. 伺服驱动器侧的信号名称为“外部锁定输入 2”。

## 背隙补偿功能

MC 功能模块不具备背隙补偿功能。

需要背隙补偿时，使用伺服驱动器侧的补偿功能。

需要在伺服驱动器侧设定的对象如下所示。

索引	名称	说明
3704Hex	背隙补偿选择	设定位置控制时背隙补偿的有效 / 无效及补偿时的动作方向。 出厂设定为“无效”。
3705Hex	背隙补偿量	设定位置控制时的背隙补偿量。
3706Hex	背隙补偿时间常数	设定位置控制时的背隙补偿时间常数。

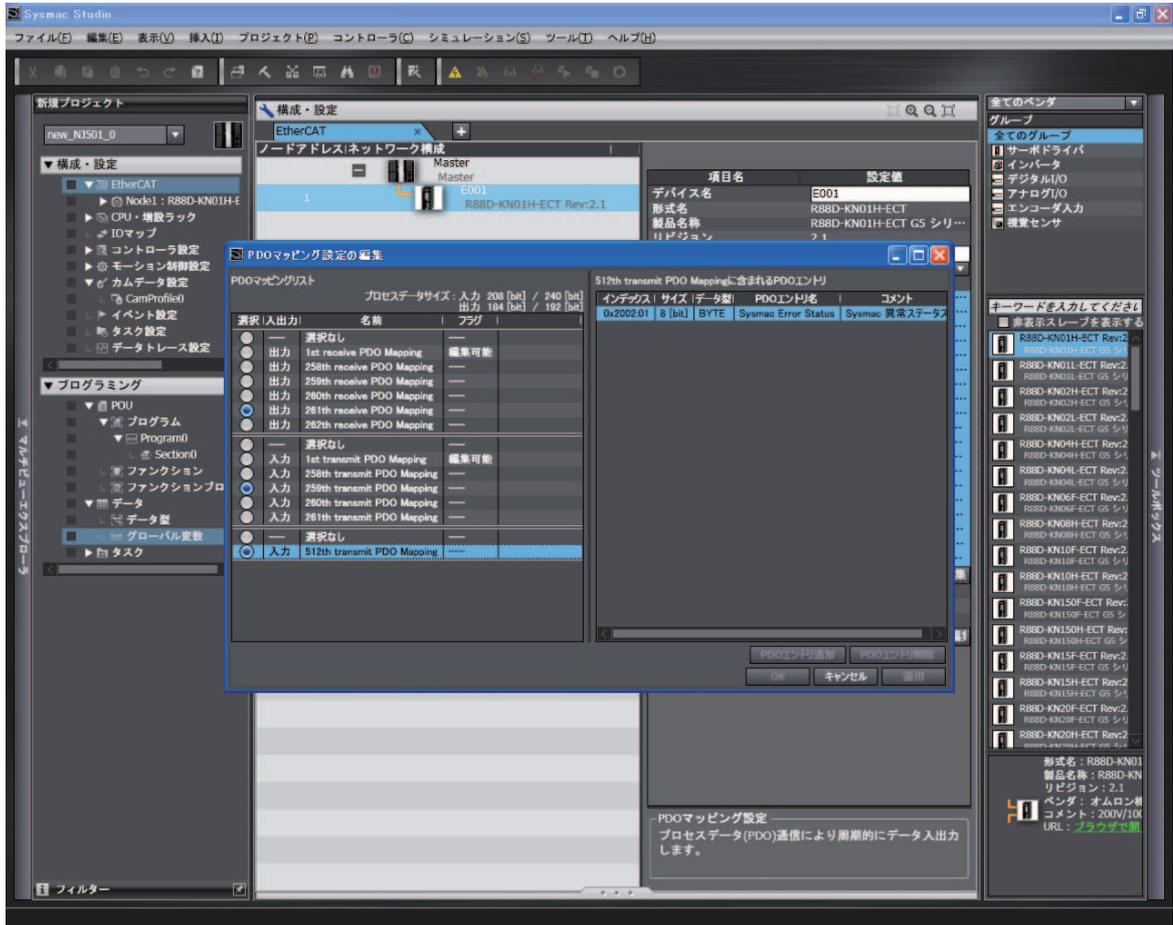
背隙功能的详情请参阅 □□ “AC 伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置型用户手册 (SBCE-365)”  
及 □□ “AC 伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型用户手册 (SBCE-366)”。

## PDO 映射

下面对作为 MC 功能模块的伺服轴进行控制的 PDO 映射进行说明。

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

通过 Sysmac Studio 的 [EtherCAT] 构成编辑画面中的 [PDO 映射设定的编辑] 选择将要使用的 PDO 映射。



Sysmac Studio 默认选择以下 PDO 映射。

RxPDO: [ 261th 接收 PDO 映射 ] (1704Hex)	控制字 (6040Hex)、目标位置 (607AHex)、目标速度 (60FFHex)、目标转矩 (6071Hex)、操作模式 (6060Hex)、锁定功能 (60B8Hex)、最大曲线速度 (607FHex)、正转侧转矩限制值 (60E0Hex)、反转侧转矩限制值 (60E1Hex)
TxPDO: [ 259th 发送 PDO 映射 ] (1B02Hex)	错误代码 (603FHex)、状态字 (6041Hex)、反馈位置 (6064Hex)、反馈转矩 (6077Hex)、操作模式显示 (6061Hex)、锁定状态 (60B9Hex)、锁定位置 1(60BAHex)、锁定位置 2(60BCHex)、数字输入 (60FDHex)



### 参考

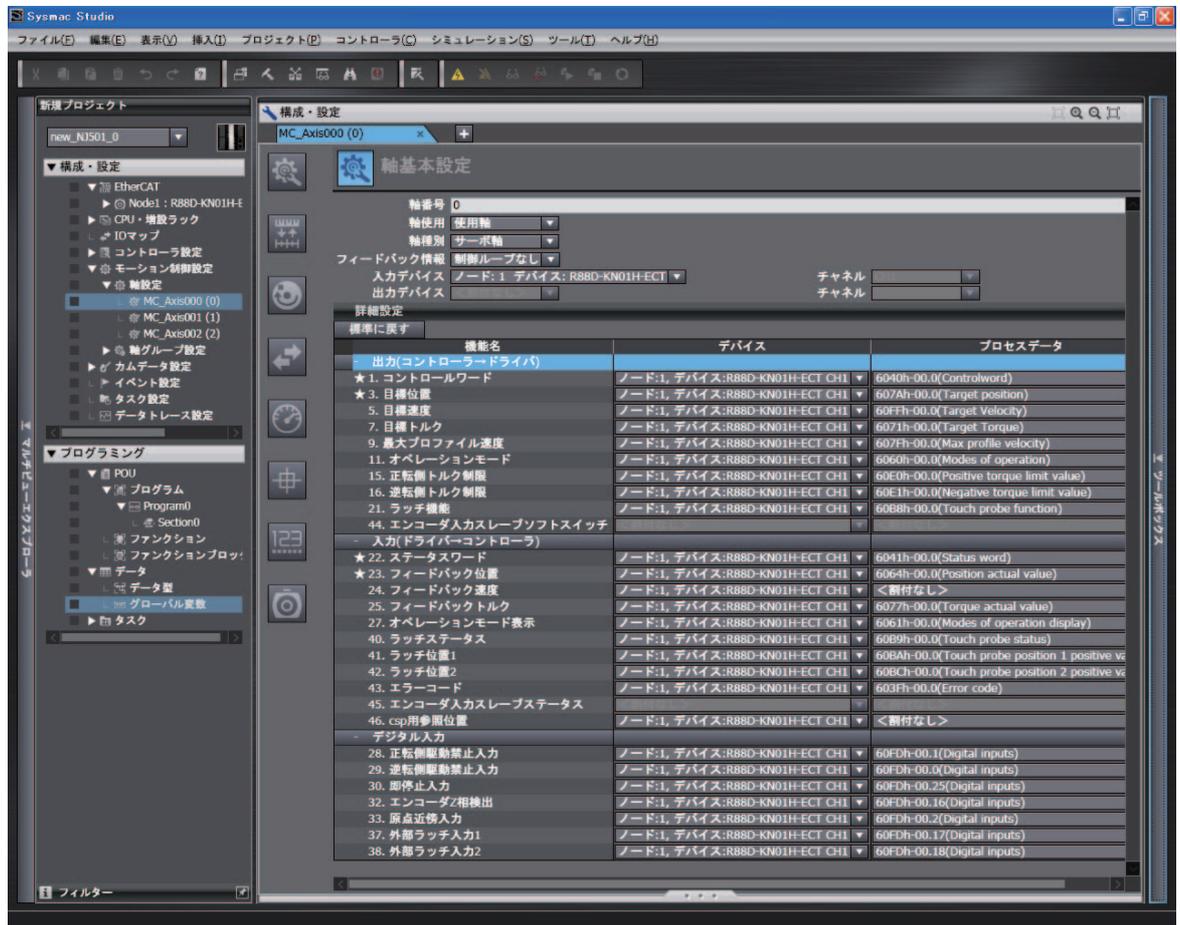
- 使用欧姆龙制推荐伺服驱动器R88D-KN□□□-ECT(Ver.2.1以上)或R88D-KN□□□-ECT-L(Ver.1.1以上)时，PDO映射可直接使用 Sysmac Studio 默认的设置。
- 在欧姆龙制 R88D-KN □□□ -ECT 中进行全闭环控制时，请给 RxPDO 选择 1701 Hex 或将对象的总大小设为 12 位以下的 1600Hex(Ver.2.1 以上时)。

## MC 功能模块的功能和过程数据之间的关系

MC 功能模块具有的功能与过程数据的对象信息相关。

根据 MC 功能模块使用的 EtherCAT 从站构成及使用功能的不同，有时需要变更这些关系。

点击 Sysmac Studio 的 [ 轴基本设定 ] 画面中的 [ 高级设置 ]，打开设定画面。



### 参考

使用欧姆龙推荐伺服驱动器 R88D-KN □□□ -ECT(Ver.2.1 以上) 或 R88D-KN □□□ -ECT-L (Ver.1.1 以上) 时，功能和过程数据之间的关系可直接使用 Sysmac Studio 默认的设置。

## ● 输出 ( 控制器→驱动器 ) 的设定

对从 MC 功能模块输出到伺服驱动器的指令数据进行设定。

下列所示为 Sysmac studio 的默认设定。(○标记的为所需对象)

功能名称	过程数据	说明
○ 控制字	6040Hex-00.0 (Controlword)	控制伺服驱动器状态的数据。 请设定为 [6040Hex: Controlword]。
○ 目标位置	607AHex-00.0 (Target position)	位置控制时的目标位置。 请设定为 [607AHex: Target position]。
目标速度	60FFHex-00.0 (Target velocity)	速度控制时的目标速度。 在周期同步速度模式下, 向驱动器输出 MC_SyncMoveVelocity( 周期同步速度控制 ) 指令等时需要该功能。 通常请设定为 [60FFHex: Target velocity]。
目标转矩	6071Hex-00.0 (Target torque)	转矩控制时的目标转矩。 在周期同步转矩模式下, 向驱动器输出 MC_TorqueControl( 转矩控制 ) 指令等时需要该功能。 通常请设定为 [6071Hex: Target torque]。
最大曲线速度	607FHex-00.0 (Max profile velocity)	转矩控制时的速度限制值。 在周期同步转矩模式下, 发出 MC_TorqueControl( 转矩控制 ) 指令等, 进行速度限制时需要该功能。 通常请设定为 [607FHex: Max profile velocity]。
操作模式	6060Hex-00.0 (Modes of operation)	切换控制模式的数据。 发出 MC_SyncMoveVelocity( 周期同步速度控制 ) 指令及 MC_TorqueControl( 转矩控制 ) 指令等, 切换为周期同步位置模式之外的控制模式时需要该功能。 通常请设定为 [6060Hex: Modes of operation]。*1
正转侧转矩限制值	60E0Hex-00.0 (Positive torque limit value)	正方向侧的转矩限制值。 发出 MC_SetTorqueLimit( 转矩限制 ) 指令等, 限制驱动器的输出转矩时需要该功能。 通常请设定为 [60E0Hex: Positive torque limit value]。
反转侧转矩限制值	60E1Hex-00.0 (Negative torque limit value)	负方向侧的转矩限制值。 发出 MC_SetTorqueLimit( 转矩限制 ) 指令等, 限制驱动器的输出转矩时需要该功能。 通常请设定为 [60E1Hex: Negative torque limit value]。
锁定功能	60B8Hex-00.0 (Touch probe function)	控制锁定功能的数据。 发出 MC_Home( 原点复位 ) 指令、MC_HomeWithParameter( 参数指定原点复位 ) 指令、MC_MoveFeed( 中断标准定位 ) 指令、MC_TouchProbe( 启用外部锁定 ) 指令、MC_MoveLink( 梯形模式凸轮 ) 指令等, 使用锁定功能时需要该功能。 通常请设定为 [60B8Hex: Touch probe function]。

\*1. 请设定为 [6060Hex: Modes of operation] 时也设定 [6061Hex: Modes of operation display]。只设定其中一个将无法正常工作。



### 使用注意事项

- 连接非推荐伺服驱动器的欧姆龙制伺服驱动器 G5 系列 EtherCAT 通信内置型单元版本时, 可能没有适用的功能。  
请确认所连接的伺服驱动器的手册。
- 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按照意图进行动作。
- 未使用欧姆龙制伺服驱动器 1S 系列 EtherCAT 通信内置型或 G5 系列 EtherCAT 通信内置型时, 请务必设定操作模式 (6060Hex)。
- 在欧姆龙制伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置型中进行全闭环控制时, 请将对象的总大小设定在 12 位以下。

## ● 输入 ( 驱动器 → 控制器 ) 的设定

从伺服驱动器输出到 MC 功能模块的状态数据。

下列所示为 Sysmac studio 的默认设定。(○标记的为所需对象)

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)	显示当前位置。请设定为 [6064Hex: Position actual value]。
反馈速度	无设定 *1	显示当前速度。使用时, 通常请设定为 [606CHex: Velocity actual value]。
反馈转矩	6077Hex (Torque actual value)	显示当前转矩。 在周期同步转矩模式下, 向驱动器输出 MC_TorqueControl( 转矩控制 ) 指令等时需要该功能。 通常请设定为 [6077Hex: Torque actual value]。
操作模式显示	6061Hex-00.0 (Modes of operation display)	显示控制模式。 发出 MC_SyncMoveVelocity( 周期同步速度控制 ) 指令及 MC_TorqueControl( 转矩控制 ) 指令等, 切换为周期同步位置模式之外的控制模式时需要该功能。 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。 发出 MC_Home( 原点复位 ) 指令、MC_HomeWithParameter( 参数指定原点复位 ) 指令、MC_MoveFeed( 中断标准定位 ) 指令、MC_TouchProbe( 启用外部锁定 ) 指令、MC_MoveLink( 梯形模式凸轮 ) 指令等, 使用锁定功能时需要该功能。 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。 发出 MC_Home( 原点复位 ) 指令、MC_HomeWithParameter( 参数指定原点复位 ) 指令、MC_MoveFeed( 中断标准定位 ) 指令、MC_TouchProbe( 启用外部锁定 ) 指令、MC_MoveLink( 梯形模式凸轮 ) 指令等, 使用锁定功能时需要该功能。通常请设定为 [60BAHex: Touch probe pos1 pos value]。
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。 发出 MC_Home( 原点复位 ) 指令、MC_HomeWithParameter( 参数指定原点复位 ) 指令、MC_MoveFeed( 中断标准定位 ) 指令、MC_TouchProbe( 启用外部锁定 ) 指令、MC_MoveLink( 梯形模式凸轮 ) 指令等, 使用锁定功能时需要该功能。通常请设定为 [60BCHex: Touch probe pos2 pos value]。
错误代码	603FHex-00.0 (Error code)	驱动器内的异常代码。 通常请设定为 [603FHex: Error code]。
esp 用基准位置	无设定	esp 模式切换时的参照位置。使用速度控制模式 (CSV) 的指令及使用转矩控制模式 (CST) 的指令时参照的数据。 欧姆龙制伺服电机 / 驱动器 G5 系列中, 气缸型单元版本 Ver.2.1 以上可使用。*3 欧姆龙制伺服驱动器 G5 系列中, 直线电机型不适用于该对象。

\*1. 需要时, 将所选的过程数据映射至 PDO 后再进行设定。606CHex-00.0(Velocity actual value) 为标准设定。

\*2. 请设定为 [6061Hex: Modes of operation display] 时也设定 [6060Hex: Modes of operation]。只设定其中一个将无法正常工作。

\*3. 使用时, 请将 [4020Hex: Reference Position for esp] 映射至 PDO。



### 使用注意事项

- 连接非推荐伺服驱动器的欧姆龙制伺服驱动器 G5 系列 EtherCAT 通信内置型单元版本时，可能没有适用的功能。请确认所连接的伺服驱动器的手册。
- 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按照意图进行动作。
- 使用欧姆龙制伺服驱动器 1S 系列 EtherCAT 通信内置型或 G5 系列 EtherCAT 通信内置型以外的类型时，请务必设定操作模式显示 (6061Hex)。
- 使用欧姆龙制伺服驱动器 G5 系列，并通过 MC\_SyncMoveVelocity(周期同步速度控制) 指令切换控制模式时，请映射“csp 用基准位置”。
- 在欧姆龙制伺服驱动器 G5 系列中使用“csp 用基准位置”时，请将原始周期设为 1ms。另外，电子齿轮比请设为 1:1。  
详情请参阅 □ “AC 伺服电机/驱动器 G5 系列 EtherCAT 通信内置型 用户手册(SBCE-365)” 及 □ “AC 伺服电机/驱动器 G5 系列 EtherCAT 通信内置直线电机型 用户手册(SBCE-366)”。

## ● 数字输入的设置

MC 功能模块使用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)	用作正方向极限输入的信号。 通常请设定为 [60FDHex-00: Digital inputs] 中的 [位 1: Positive limit switch]。
反转侧驱动禁止输入	60FDHex-00.0 (Digital inputs)	用作负方向极限输入的信号。 通常请设定为 [60FDHex-00: Digital inputs] 中的 [位 0: Negative limit switch]。
即停输入	60FDHex-00.25 (Digital inputs)	用作即停输入的信号。 在欧姆龙制伺服驱动器 G5 系列中请设定为 [60FDHex-00: Digital inputs] 中的 [位 25: Immediate Stop Input]。
编码器 Z 相检测	60FDHex-00.16 (Digital inputs)	Z 相输入的检测状态显示。 欧姆龙制伺服电机 / 驱动器 G5 系列中，气缸型单元版本 Ver.2.1 以上请设定为 [60FDHex-00: Digital inputs] 中的 [位 16: Encoder Phase Z Detection]。 伺服驱动器 G5 系列中，直线电机型不适用于该对象。
近原点输入	60FDHex-00.2 (Digital inputs)	用作近原点输入的信号。 通常请设定为 [60FDHex-00: Digital inputs] 中的 [位 2: Home switch]。
外部锁定输入 1	60FDHex-00.17 (Digital inputs)	用作外部锁定输入 1 的信号状态显示。 在欧姆龙制伺服驱动器 G5 系列中请设定为 [60FDHex-00: Digital inputs] 中的 [位 17: External Latch Input 1]。
外部锁定输入 2	60FDHex-00.18 (Digital inputs)	用作外部锁定输入 2 的信号状态显示。 在欧姆龙制伺服驱动器 G5 系列中请设定为 [60FDHex-00: Digital inputs] 中的 [位 18: External Latch Input 2]。



### 使用注意事项

- 连接非推荐伺服驱动器的欧姆龙制伺服驱动器 G5 系列 EtherCAT 通信内置型单元版本时，可能没有适用的功能。请确认所连接的伺服驱动器的手册。
- 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按照意图进行动作。

## 版本相关信息

- Ver.1.09 以下版本的 CPU 单元, 如未使用欧姆龙制伺服驱动器 G5 系列 EtherCAT 通信内置型, 需映射操作模式 (6060Hex) 及操作模式显示 (6061Hex)。
- Ver.1.10 以上版本的 CPU 单元, 根据有无操作模式 (6060Hex) 及操作模式显示 (6061Hex) 映射, 动作如下。

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

- \*1. 伺服驱动器的控制模式中的周期性同步位置模式 (CSP)。  
 \*2. 伺服驱动器的控制模式中的周期性同步速度模式 (CSV)。  
 \*3. 伺服驱动器的控制模式中的周期性同步转矩模式 (CST)。

## 伺服驱动器侧的对象设定

使用 MC 功能模块控制功能的欧姆龙制伺服驱动器 G5 系列侧的设定如下所示。  
 使用伺服驱动器的各种功能时, 请按照伺服驱动器手册设定各功能相关的对象。

索引	子索引	名称	推荐设定值	说明
3013Hex	00Hex	第 1 外部转矩限制 *1	1388Hex	初始值 =500.0%
3015Hex *2	00Hex	使用绝对值编码器时动作切换	0002Hex	用作绝对值, 忽略多圈计数器超限。
3317Hex	00Hex	速度限制选择	0001Hex	转矩控制时的速度限制方法为, 通过 607FHex: Max profile velocity 和 3321Hex: 速度限制值设定中较小的数值进行限制。
3323Hex	00Hex	外部反馈脉冲类型选择	0000Hex 或 0001Hex、0002Hex *3	所用外部标尺类型的设定。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	设定电机转 1 圈的编码器分辨率。设定为“0”时, 自动进行设定。
3401Hex	00Hex	输入信号选择 2	00818181Hex	正转侧驱动禁止输入 (b 触点)
3402Hex	00Hex	输入信号选择 3	00828282Hex	反转侧驱动禁止输入 (b 触点)
3403Hex	00Hex	输入信号选择 4	00222222Hex	近原点输入 (a 触点)
3404Hex	00Hex	输入信号选择 5	002B2B2BHex	外部锁定信号 3(a 触点)
3405Hex	00Hex	输入信号选择 6	00212121Hex	外部锁定信号 2(a 触点)
3406Hex	00Hex	输入信号选择 7	00202020Hex	外部锁定信号 1(a 触点)
3504Hex	00Hex	驱动禁止输入选择	0001Hex	将伺服侧的驱动禁止输入设为无效。在 MC 功能模块侧进行。
3508Hex	00Hex	欠电压错误选择	0001Hex	发生主电源欠电压并停止。
3521Hex	00Hex	转矩限制选择 *1	0006Hex	正反都有 2 个限制值。通过 PCL、NCL 进行切换。
3522Hex	00Hex	第 2 外部转矩限制 *1	1388Hex	初始值 =500.0%
3703Hex	00Hex	转矩限制中输出设定 *1	0001Hex	通过除转矩指令值之外的转矩限制设为 ON。
3801Hex	00Hex	软件限制功能	0003Hex	设为两侧软件限制无效。

索引	子索引	名称	推荐设定值	说明
3758Hex	00Hex	锁定触发选择	0100Hex	设为 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	警告原因消失后，自动清除警告。
607CHex	00Hex	编码器原点位置偏置	00000000Hex	以“0”使用伺服侧的偏置值。
6091Hex	01Hex	电子齿轮比分子	00000001Hex	将伺服驱动器侧的齿轮比设为 1:1。在 MC 功能模块侧设定同样的功能。
	02Hex	电子齿轮比分母	00000001Hex	

- \*1. 在欧姆龙制伺服驱动器 G5 系列 EtherCAT 通信内置直线电机型中，请将“转矩”改称为“推力”。
- \*2. 欧姆龙制伺服驱动器 G5 系列 EtherCAT 通信内置直线电机型不适用于该对象。
- \*3. 在欧姆龙制伺服电机 / 驱动器 G5 系列中进行全闭环控制时，或使用欧姆龙制伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型时，请根据所用外部标尺类型进行设定。

## A-3 与编码器输入终端之间的连接

下面对与欧姆龙制 GX 系列 EtherCAT 从站 编码器输入终端之间的连接进行说明。

### A-3-1 与编码器输入终端之间的接线

通过 EtherCAT 通信连接编码器输入终端。

关于连接方法，请参阅 □□ “NJ/NX 系列 CPU 单元内置 EtherCAT 端口 用户手册 (SBCD-358)”。

### A-3-2 编码器输入终端的设定

下面，对连接作为 MC 功能模块控制对象的欧姆龙制 EtherCAT 远程 I/O 终端 GX 系列 GX-EC0211/EC0241 时的编码器输入终端侧的设定概要进行说明。

关于编码器输入终端的详情，请参阅 □□ “GX 系列 EtherCAT 从站 用户手册 (SBCD-350)”。

### 推荐编码器输入终端

作为 MC 功能模块的编码器轴，可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

- 可用作编码器轴的 EtherCAT 从站仅为欧姆龙制 GX-EC0211/EC0241。
- 欧姆龙制 GX-EC0211/EC0241(单元版本 Ver.1.0) 也可用作编码器轴，但不适用于 Sysmac 设备功能。  
使用单元版本 Ver.1.0 时，请将节点地址开关设定为非“00”。如果设定为“00”，网络构成将出现异常。  
功能的详情，请参阅 □□ “GX 系列 EtherCAT 从站 用户手册 (SBCD-350)”。
- 关于 Sysmac 设备，请参阅 □□ “NJ/NX 系列 CPU 单元 用户手册 软件篇 (SBCA-359)”。

### 外部输入信号

使用作为编码器轴的所有功能时，使用编码器输入终端侧的下列输入信号。

- 计数器 A 相
- 计数器 B 相
- 计数器 Z 相
- 锁定输入 (A/B)

计数器包含 2 个通道，每个通道中有 2 个外部锁定。请连接实际使用的输入信号。

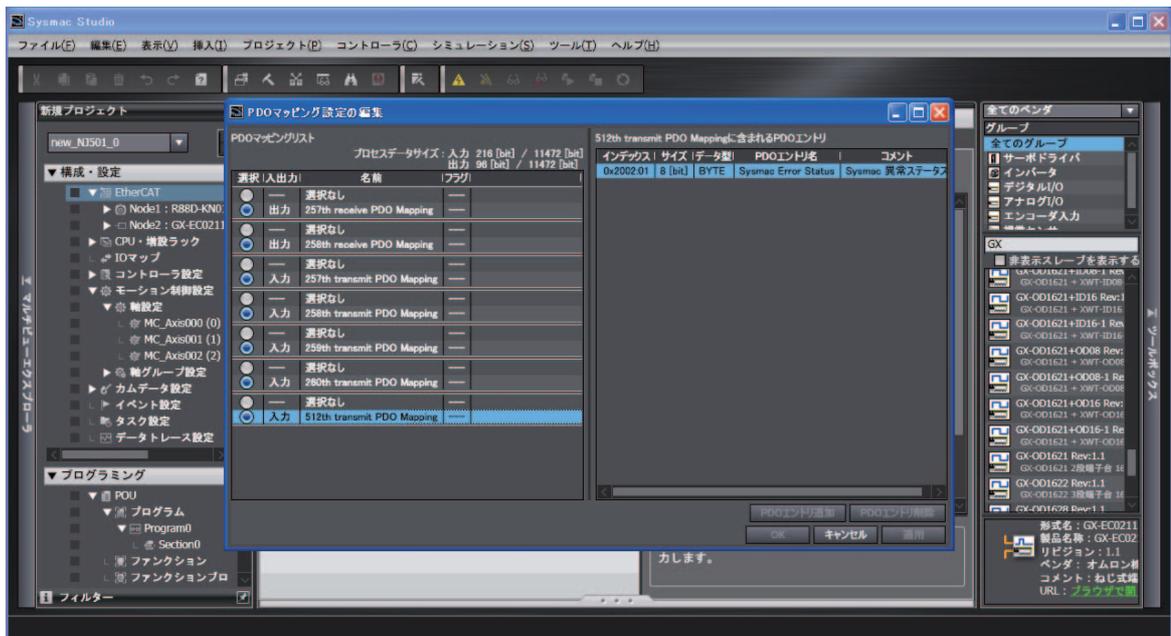
与输入信号的接线方法请参阅 □□ “GX 系列 EtherCAT 从站 用户手册 (SBCD-350)”。

## PDO 映射

下面对作为 MC 功能模块的编码器轴进行控制的 PDO 映射进行说明。

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

通过 Sysmac Studio 的 [EtherCAT] 编辑画面中的 [PDO 映射设定的编辑] 选择将要使用的 PDO 映射。



Sysmac Studio 默认选择以下 PDO 映射。

RxPDO(1700h)	通道 1 软开关 (4020h-01h)、通道 2 软开关 (4020h-02h)
RxPDO(1701h)	通道 1 预设当前值 (4011h-01h)、通道 2 预设当前值 (4011h-02h)
TxPDO(1B00h)	通道 1 计数器当前值 (4010-01h)、通道 2 计数器当前值 (4010-02h)
TxPDO(1B01h)	通道 1 锁定值 A(4012-01h)、通道 2 锁定值 A(4012-02h)
TxPDO(1B02h)	通道 1 锁定值 B(4013-01h)、通道 2 锁定值 B(4013-02h)
TxPDO(1B03h)	通道 1 状态位 (4030-01h)、通道 2 状态位 (4030-02h)
TxPDO(1BFFh)	Sysmac 异常状态 (2002-01h)



### 参考

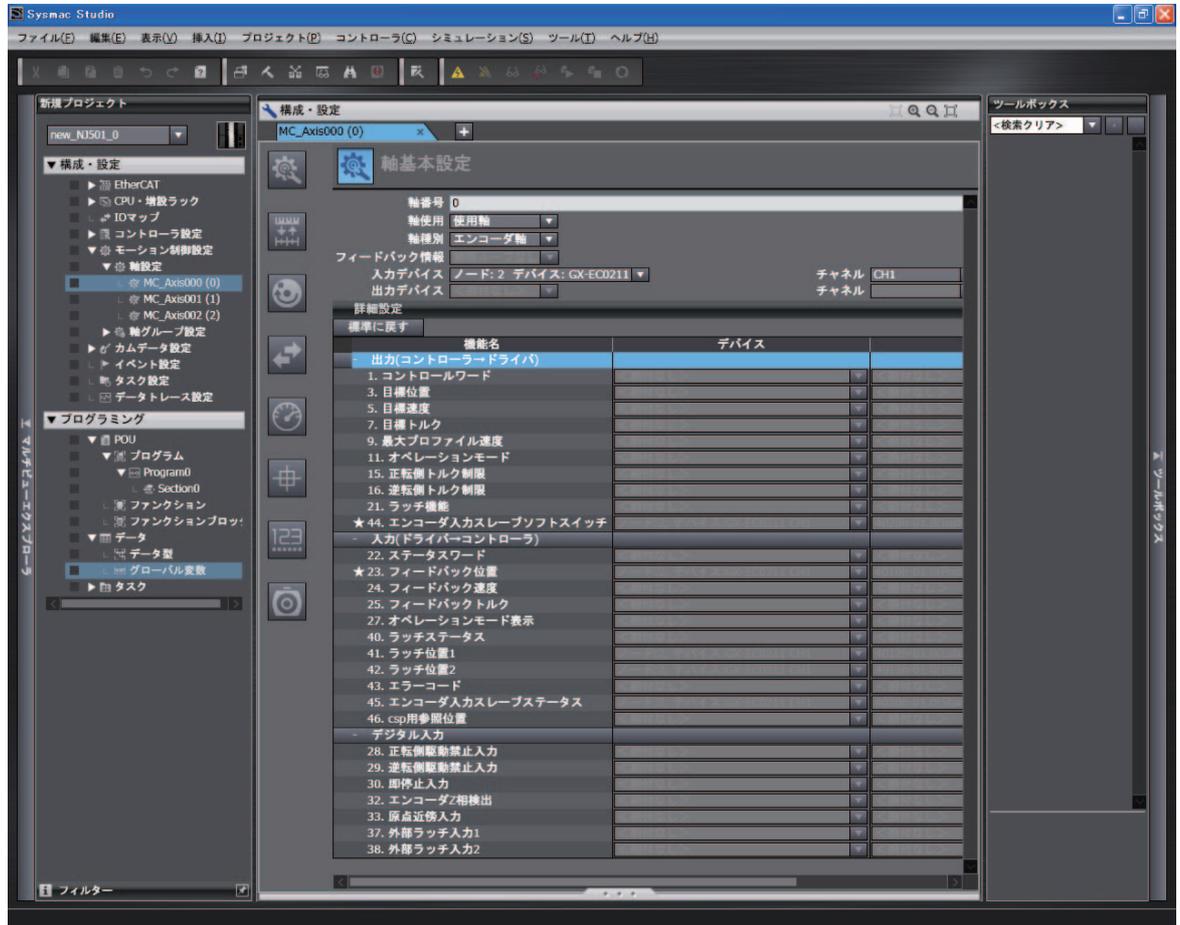
使用推荐编码器输入终端 GX-EC0211/EC0241(单元版本 Ver.1.1 以上)时，PDO 映射可直接使用 Sysmac Studio 默认的映射。

## MC 功能模块的功能和过程数据之间的关系

MC 功能模块具有的功能与过程数据的对象信息相关。

根据 MC 功能模块使用的 EtherCAT 从站构成及使用功能的不同，有时需要变更这些关系。

点击 Sysmac Studio 的 [ 轴基本设定 ] 画面中的 [ 高级设置 ]，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 Ver.1.1 以上)时，功能和过程数据之间的关系可直接使用 Sysmac Studio 默认的设置。

## ● 输出 ( 控制器→驱动器 ) 的设置

设定从 MC 功能模块向编码器输入终端输出的指令数据。

下列所示为 Sysmac studio 的默认设定。(带★标记的为所需对象)

功能名称	过程数据		说明
	通道 1	通道 2	
★ 编码器输入从站软开关	4020h-01.0 (Instruction Bits)	4020h-02.0 (Instruction Bits)	设定软开关的内容。 请通过所用通道, 设定左侧的对象。



### 使用注意事项

- 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按照意图进行动作。

## ● 输入 ( 驱动器→控制器 ) 的设置

从编码器输入终端向 MC 功能模块输出的状态数据。

下列所示为 Sysmac studio 的默认设定。(带★标记的为所需对象)

功能名称	过程数据		说明
	通道 1	通道 2	
★ 反馈位置	4010h-01.0 (Position Value)	4010h-02.0 (Position Value)	存储了计数器的当前值。 请通过所用通道, 设定左侧的对象。
锁定位置 1	4012h-01.0 (Latch Value A)	4012h-02.0 (Latch Value A)	通过锁定功能 1 锁定的位置。存储了锁定 A 的数值。 使用锁定功能时需要 MC_TouchProbe(启动外部锁定) 指令等。 请通过所用通道, 设定左侧的对象。
锁定位置 2	4013h-01.0 (Latch Value B)	4013h-02.0 (Latch Value B)	通过锁定功能 2 锁定的位置。存储了锁定 B 的数值。 使用锁定功能时需要 MC_TouchProbe(启动外部锁定) 指令等。 请通过所用通道, 设定左侧的对象。
编码器输入从站状态	4030h-01.0 (Status Bits)	4030h-02.0 (Status Bits)	保存了状态位。 使用锁定功能时需要 MC_TouchProbe(启动外部锁定) 指令等。 请通过所用通道, 设定左侧的对象。



### 使用注意事项

- 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按照意图进行动作。

## ● 数字输入的设置

用作编码器轴时, 无需设定。

## 编码器输入终端侧的对象设定

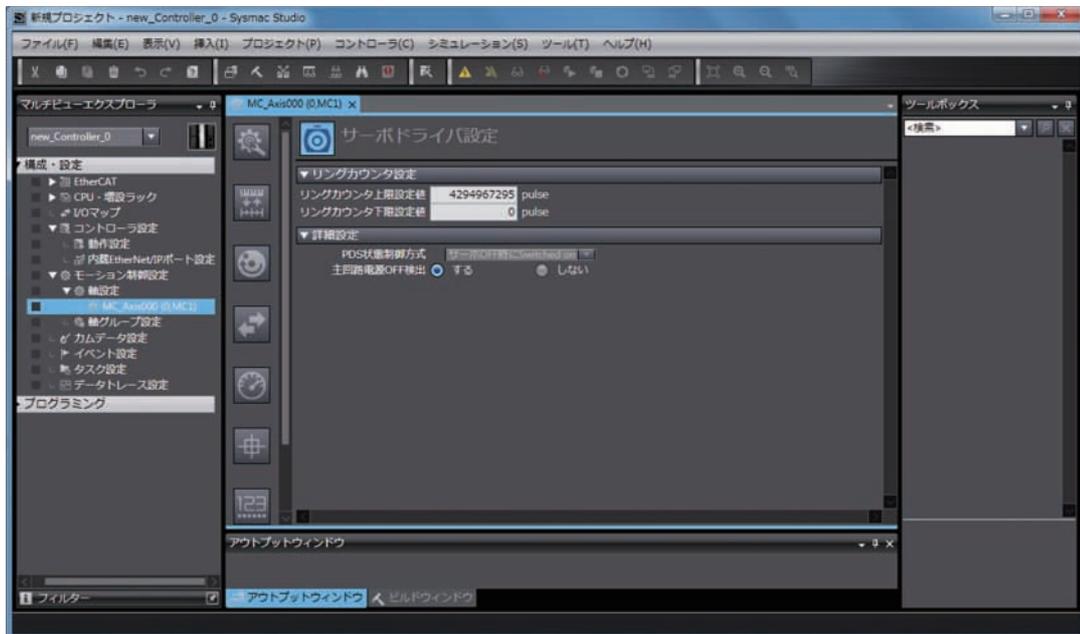
在编码器输入终端中, 没有需要设定的对象。

## MC 功能模块和编码器输入终端的环计数器的关系

需要使 MC 功能模块轴参数的伺服驱动器设定 ( 环计数器上限值、环计数器下限值 ) 和编码器输入终端的环计数器最大值设定保持一致。

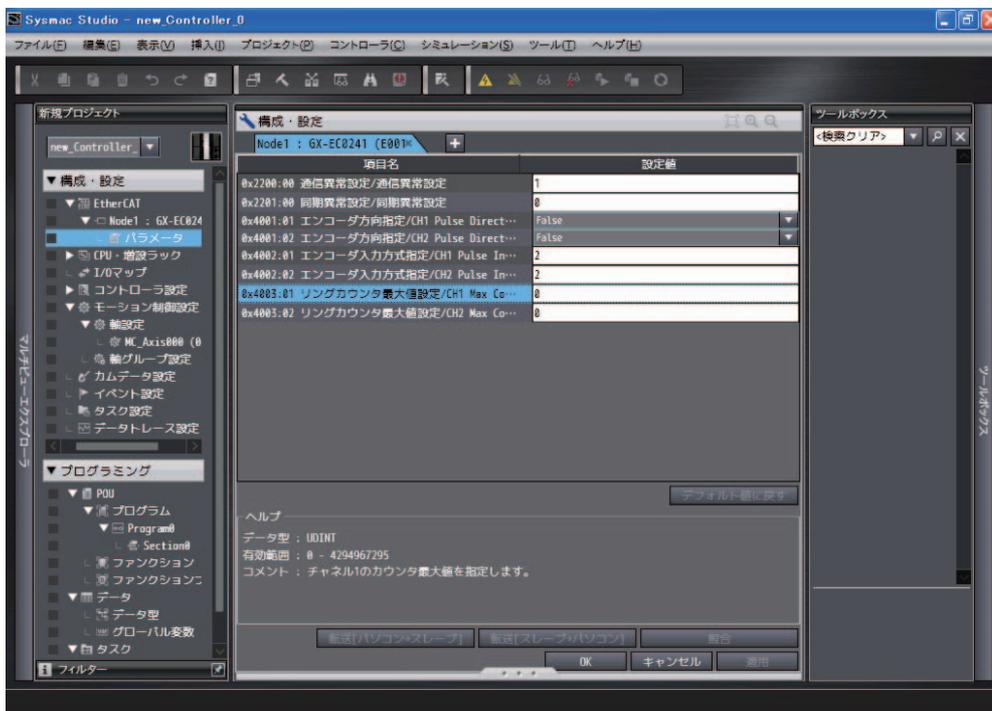
在 Sysmac Studio 的 [ 伺服驱动器设定 ] 画面中编辑伺服驱动器设定 ( 环计数器上限值、环计数器下限值 )。设定内容如下所示。

参数	含义	设定值
环计数器上限设定值	设定伺服驱动器或编码器输入终端设定的环计数器上限值。	请和编码器输入终端的环计数器最大值设定保持一致。
环计数器下限设定值	设定伺服驱动器或编码器输入终端设定的环计数器下限值。	请设定为 0。



在 Sysmac Studio 中 [EtherCAT] 编辑画面的 [参数] 中编辑编码器输入终端的环计数器最大值设定。设定内容如下所示。

索引	对象名称	设定值
0x4003	环计数器最大值设定 (Max Count Setting)	请和 MC 功能模块轴参数的伺服驱动器设定 (环计数器上限设定值) 保持一致。



## A-4 与 NX 单元之间的连接

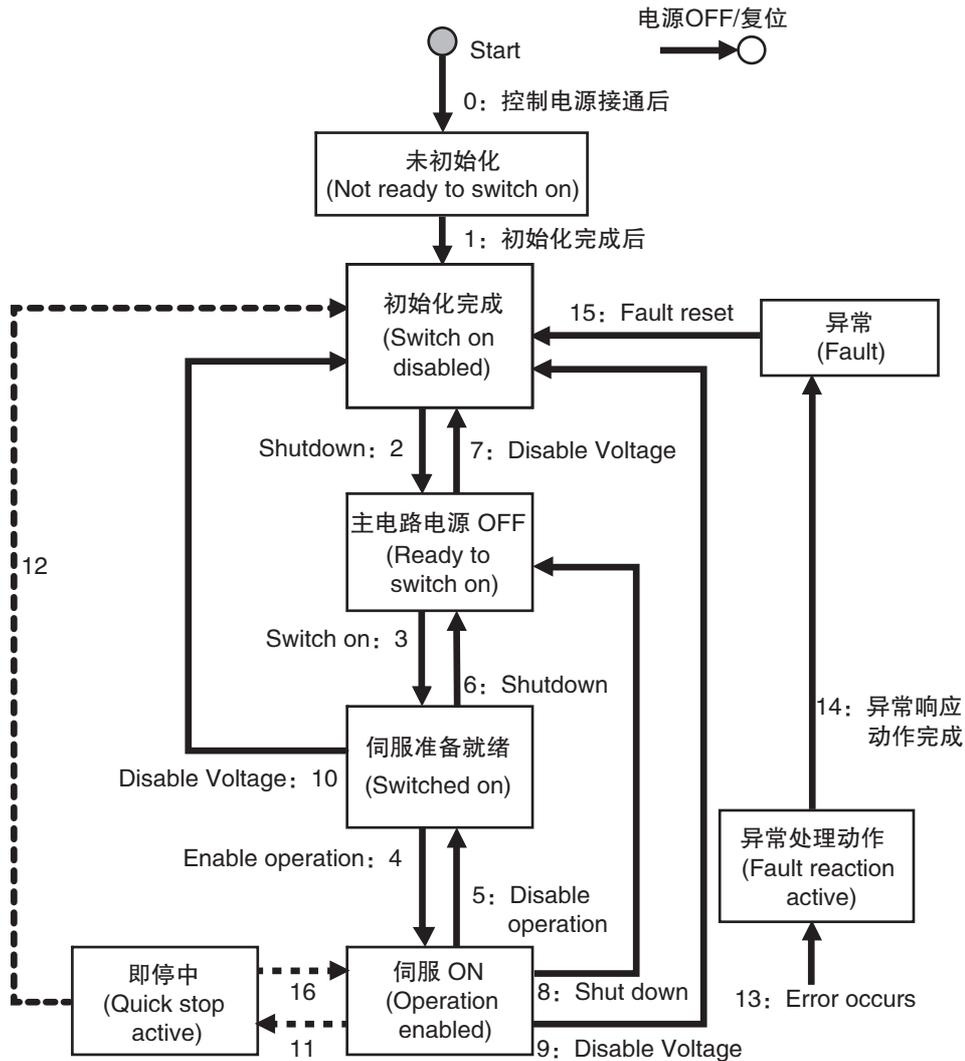
---

与 NX 系列 位置接口单元之间的连接，请参阅 □ “NX 系列 位置接口单元 用户手册 (SBCE-374)”。

## A-5 PDS 状态切换

PDS 状态切换在 CiA402 驱动器配置文件中定义。PDS 状态切换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。

CiA402 驱动器配置文件中定义的状态切换图如下所示。  
矩形内部表示状态，数字表示状态控制指令。



## A-5-1 PDS 状态控制方式

下面对轴参数 [PDS 状态控制方式] 的设定值与 PDS 状态的关系进行说明。

### ● [PDS 状态控制方式] 设为 “0” 时

[PDS 状态控制方式] 设为 “0: 伺服 OFF 时 Switched on” 后, 将执行如下动作。

- 与伺服驱动器确立通信后, MC 功能模块会自动将 PDS 状态切换至伺服准备就绪 (Switched on) 状态。
- 在伺服准备就绪 (Switched on) 状态下, 将 MC\_Power(可运行) 指令的 “Enable(启用)” 设为 TRUE 后, 切换至伺服 ON(Operation enabled)。实施状态切换图的 “4: Enable operation” 切换。
- 在伺服 ON(Operation enabled) 状态下, 将 MC\_Power(可运行) 指令的 “Enable(启用)” 设为 FALSE 后, 切换至伺服准备就绪 (Switched on) 状态。实施状态切换图的 “5: Disable operation” 切换。

### ● [PDS 状态控制方式] 设为 “1” 时

[PDS 状态控制方式] 设为 “1: 伺服 OFF 时 Ready to switch on” 后, 将执行如下动作。

- 与伺服驱动器确立通信后, MC 功能模块会自动将 PDS 状态切换至主电路电源 OFF(Ready to switch on) 状态。
- 在主电路电源 OFF(Ready to switch on) 状态下, 将 MC\_Power(可运行) 指令的 “Enable(启用)” 设为 TRUE 后, 切换至伺服 ON(Operation enabled)。同时实施状态切换图的 “3: Switch on” 和 “4: Enable operation” 切换。
- 在伺服 ON(Operation enabled) 状态下, 将 MC\_Power(可运行) 指令的 “Enable(启用)” 设为 FALSE 后, 切换至主电路电源 OFF(Ready to switch on) 状态。同时实施状态切换图的 “5: Disable operation” 和 “6: Shutdown” 切换。

### 版本相关信息

- Ver.1.09 以下版本的 CPU 单元的 [PDS 状态控制方式] 为 “0”。Ver.1.10 以上版本的 CPU 单元的 [PDS 状态控制方式] 可选择。
- Ver.1.10 以上版本的 CPU 单元的 [PDS 状态控制方式] 的初始值为 “0”。

## A-5-2 主电路电源 OFF 检测

可在伺服锁定状态下选择是否检测主电路电源 OFF。

执行 MC\_Power(可运行)指令出现“伺服主电路电源 OFF 检测”异常时,请设为“不检测”。



### 使用注意事项

---

使用欧姆龙制伺服驱动器 1S 系列或 G5 系列时,无法选择“不检测”。

选择“不检测”,并在伺服锁定状态下将伺服主电路电源 OFF 后,“伺服主电路电源 OFF 检测”异常。

---



### 版本相关信息

---

- Ver.1.09 以下版本的 CPU 单元的 [主电路电源 OFF 检测] 为“检测”。Ver.1.10 以上版本的 CPU 单元的 [主电路电源 OFF 检测] 可选择。
  - Ver.1.10 以上版本的 CPU 单元的 [主电路电源 OFF 检测] 的初始值为“检测”。
-

## A-6 用语说明

下面对运动控制相关的用语进行说明。

### A-6-1 NJ/NX 系列通用

用语	说明
主存储器 (main memory)	CPU 单元在 OS、用户程序执行时使用的 CPU 单元内部的存储器
固定周期任务 (periodic task)	每个周期执行 I/O 刷新和用户程序的任务
原始恒定周期任务 (primary periodic task)	优先度最高的任务
任务周期 (period of tasks)	原始恒定周期任务及固定周期任务的执行间隔
I/O 刷新 (I/O Refresh)	事先规定的存储区域与外部的周期性数据交换
程序 (program)	与函数、功能块并列的 3 种 POU 之一 分配至任务后动作
用户程序 (user program)	多个程序的集合体
内联 ST (Inline ST)	梯形图程序中，作为其一部分编写 ST 程序或编写的 ST 程序
系统定义变量 (system-defined variables)	所有的变量属性通过系统进行定义，用户无法进行任何变更的变量
全局变量 (global variable)	可通过所有 POU(程序、功能块、函数)进行读写的变量
局部变量 (local variable)	只能从所定义的 POU 内访问的变量 局部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)	通过 Sysmac Studio 的“同步”功能，将数据从 Sysmac Studio 传送至控制器
上传 (upload)	通过 Sysmac Studio 的“同步”功能，将数据从控制器传送至 Sysmac Studio
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 单元立即停止执行用户程序，断开包含远程 I/O 在内的所有从站 / 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 系列控制器某个功能模块整体无法控制的异常 部分停止故障发生后，NJ/NX 系列 CPU 单元仍将继续运行
轻度故障等级的控制器异常 (minor fault level Controller error)	NJ/NX 系列控制器某个功能模块的一部分无法控制的异常 轻度故障发生后，NJ/NX 系列 CPU 单元仍将继续运行
监控信息 (observation)	表示控制器信息及用户信息的事件重要度的等级之一 对控制无影响级别的轻微异常，记录在事件日志中并通知用户的信息

## A-6-2 运动控制功能

用 语	说 明
使用实轴	轴种类为 [ 伺服轴 ] 或 [ 编码器轴 ], 且轴使用设为 [ 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 根据用户程序内的运动控制命令所赋予的指令, 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功能块定义的指令 在运动控制功能模块中, 具有遵循 PLCopen® 的运动控制用功能块的指令, 以及运动控制功能模块独有的指令
运动控制轴 (motion control axes)	可使用运动控制功能的所有功能的轴
单轴位置控制轴 (single-axis position control axes)	运动控制功能中, 仅可使用 " 单轴位置控制 " 的轴
单轴位置控制 (single-axis position control)	以 1 轴进行定位控制
单轴速度控制 (single-axis velocity control)	以 1 轴进行速度控制 在 MC 功能模块中, 有时对伺服驱动器赋予速度指令值和位置指令值
单轴转矩控制 (single-axis torque control)	以 1 轴进行转矩控制
单轴同步控制 (single-axis synchronized control)	单轴的从轴与一个主轴同步的控制 包括按齿轮比同步的齿轮动作、与凸轮表的相位和位移关系同步的凸轮动作
单轴手动控制 (single-axis manual operation)	微动移动等手动操作的控制
单轴控制辅助 (auxiliary functions for single-axis control)	超调值设定及异常复位等时, 辅助单轴控制的功能
多轴协调控制 (multi-axes coordinated control)	直线插补、圆弧插补等时, 多轴协同动作的控制 通过轴组指定需要协调的轴
多轴协调控制辅助 (auxiliary functions for multi-axes coordinated control)	对轴组进行超调值设定及异常复位等时, 辅助多轴控制的功能
运动控制参数 (motion control parameters)	指定 MC 功能模块的各种动作的参数 MC 通用参数、轴参数、轴组参数的总称
轴参数 (axis parameters)	轴单位参数
轴组参数 (axes group parameters)	轴组单位的参数
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等的变量 MC 通用变量、轴变量、轴组变量的总称
MC 通用变量 (MC common variable)	通过结构体对 MC 功能模块通用的各种状态信息等进行定义的系统定义变量
轴变量 (axis variable)	通过结构体对每个轴的各种状态信息或部分轴参数设定信息等进行定义的系统定义变量
轴组变量 (axes group variable)	通过结构体对每个轴组的各种状态信息或部分轴组参数设定信息等进行定义的系统定义变量
原点复位 (homing)	确定机械原点 也称为确定原点、原点搜索、校准、Datum
机械原点 (home)	按装置确定的机械原点 原点复位动作时通过原点输入信号确定的原点
原点 (zero position)	根据机械原点通过用户程序处理为零位的原点 如果不变更原点位置, 则位置与机械原点相同

用語	説明
位置偏差 (following error)	指令当前位置和反馈当前位置的差 在非位置控制模式下，不存在指令当前位置，因此也不存在位置偏差 也称为偏差计数器值、累积脉冲等
偏差计数器复位 (following error reset)	将位置偏差归零
凸轮曲线 (cam profile curve)	用于实现凸轮动作的相位与位移之间关系的曲线 通过 Sysmac Studio 的凸轮编辑器绘制的曲线
凸轮数据 (cam data)	由实现凸轮动作的相位 ( 主轴 ) 和位移 ( 从轴 ) 构成的数据
凸轮数据变量 (cam data variable)	对凸轮数据进行结构体数组化的变量 由相位和变位数据构成的结构体类型的数组指定的变量
凸轮表 (cam table)	凸轮数据的数据表 将通过凸轮编辑器绘制的凸轮曲线传送至 CPU 单元时，在 CPU 单元内置的非易失性存储器中保存为凸轮表
超调 (override)	在作业过程中可暂时变更作业者编写的数值的功能
跃度 (jerk)	加速度 / 减速度变化的比率 通过指定跃度，加减速时的速度波形变为 S 型 也称为“跃动”或“加加速度”
多任务运动 (Multi-motion)	是指使用原始恒定周期任务和固定周期任务 ( 执行优先度 5 ) 两种任务并行执行运动控制

### A-6-3 EtherCAT 通信

用語	説明
CAN application protocol over EtherCAT(CoE)	EtherCAT 上的 CAN 应用协议服务
CAN in Automation(CiA)	实施高层协议的开发与支持的用户及厂家组成的国际团体
EtherCAT 协会 (ETG) (EtherCAT Technology Group)	致力于促进 EtherCAT 开放化普及的团体
对象 (Object)	设备内特定构成体的抽象表现，数据、参数、方法的集合体
对象词典 (Object Dictionary)	包含数据型对象、通信对象、应用对象描述的数据结构
过程数据 (Process Data)	以测量及控制为目的，按周期性或非周期性指定传送的应用对象的集合体
过程数据对象 (Process Data Object)	通过映射对具有 1 个以上的过程数据实体的参数进行描述的结构
服务数据对象 (Service Data Object)	通过 CoE 的非同步邮箱通信，可读写所有的对象词典
接收 PDO (Receive PDO)	通过 EtherCAT 从站接收的过程数据对象
发送 PDO (Transmit PDO)	从 EtherCAT 从站发送的过程数据对象
设备曲线 (Device Profile)	设备等级的应用参数及功能的描述集合体

## A-7 版本相关信息

下面对各单元版本的支持功能进行说明。

CPU 单元的单元版本的种类和 Sysmac Studio 版本之间的关系，请参阅 □ “NJ/NX 系列 CPU 单元 用户手册 软件篇 (SBCE-359)”。

运动控制指令的版本相关信息详情请参阅 □ “NJ/NX 系列 指令基准手册运动篇 (SBCE-364)”。

### 单元版本 Ver.1.01 中追加的运动控制功能

使用单元版本 Ver.1.01 中追加的功能时，请使用 Sysmac Studio Ver.1.02 以上。

功 能	概 要
写入轴组构成轴	通过 MC_ChangeAxesInGroup( 写入轴组构成轴 ) 指令暂时改写轴组参数 [ 构成轴 ] 的功能。
轴组位置获取	通过 MC_GroupReadPosition( 获取轴组位置 ) 指令获取轴组的指令当前位置和反馈当前位置的功能。
轴组周期同步绝对位置控制	通过 MC_GroupSyncMoveAbsolute( 轴组周期同步绝对位置控制 ) 指令，将各轴指定的目标位置周期性输出的功能。
控制对象伺服驱动器	追加欧姆龙制伺服电机 / 驱动器 G5 系列 (EtherCAT 通信内置直线电机型)。

### 单元版本 Ver.1.02 中追加的运动控制功能

使用单元版本 Ver.1.02 中追加的功能时，请使用 Sysmac Studio Ver.1.03 以上。

虽然没有单元版本 Ver.1.02 中追加的运动控制功能，但对部分指令规格进行了改善。运动控制指令的版本相关信息详情请参阅 □ “NJ/NX 系列 指令基准手册运动篇 (SBCE-364)”。

### 单元版本 Ver.1.03 中追加的运动控制功能

使用单元版本 Ver.1.03 中追加的功能时，请使用 Sysmac Studio Ver.1.04 以上。

功 能	概 要
周期同步绝对定位	通过 MC_SyncMoveAbsolute( 周期同步绝对定位 ) 指令，在位置控制模式中按控制周期输出指令位置的功能。
参数指定原点复位	通过 MC_HomeWithParameter( 参数指定原点复位 ) 指令，指定原点复位参数并驱动电机，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 Ver.1.04 中追加的运动控制功能

使用单元版本 Ver.1.04 中追加的功能时，请使用 Sysmac Studio Ver.1.05 以上。

功 能	概 要
轴使用变更	通过 MC_ChangeAxisUse( 轴使用变更 ) 指令暂时切换轴参数中的 [ 轴使用 ] 的功能。

## 单元版本 Ver.1.05 中追加的运动控制功能

使用单元版本 Ver.1.05 中追加的功能时，请使用 Sysmac Studio Ver.1.06 以上。

功能	概要
启动速度	设定开始轴动作时的初速度的功能。
输入信号的逻辑反转	将立即停止输入信号、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进行反转的功能。

(注) Ver.1.05 以上版本的 CPU 单元，可为轴分配 NX 系列 位置接口单元，进行运动控制。

## 单元版本 Ver.1.06 中追加、变更的运动控制规格

使用单元版本 Ver.1.06 中追加、变更的性能规格和功能规格时，请使用 Sysmac Studio Ver.1.07 以上版本。

功能	概要
控制轴最大数量	NJ301-□□□□ 的最大控制轴数扩展为 15 轴。 (最大使用实轴数无变更。)
单轴控制最大数量	NJ301-□□□□ 的最大单轴控制数扩展为 15 轴。 (最大使用实轴数无变更。)
数字凸轮开关有效	通过 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	使用 MC_TimeStampToPos(时间戳→轴位置计算) 指令，计算指定时间戳的轴位置的功能。
凸轮动作开始中追加合并	在 MC_CamIn(凸轮动作开始) 指令的缓冲模式选择中追加合并。
_sMC_POSITION_REF 型	3D 运动监控显示任意坐标系轨迹用的数据类型。

## 单元版本 Ver.1.08 中追加、变更的运动控制功能

使用单元版本 Ver.1.08 中追加、变更的功能时，请使用 Sysmac Studio Ver.1.09 以上。

功能	概要
凸轮表生成	使用 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	使用 MC_WriteAxisParameter(轴参数写入) 指令及 MC_ReadAxisParameter(轴参数读取) 指令，从用户程序查看、变更轴参数的功能。
设备变量分配	可为分配给轴变量的从站、单元的 I/O 端口分配设备变量。 <sup>*1</sup> 详情请参阅 □□ “2-4-2 EtherCAT 主站功能与运动控制功能模块之间的关系 (P.2-19)”。

\*1. 与 CPU 单元的版本无关。使用 Sysmac Studio Ver.1.09 以上版本即可分配。

## 单元版本 Ver.1.09 中追加、变更的运动控制功能

使用单元版本 Ver.1.09 中追加的功能时，请使用 Sysmac Studio Ver.1.10 以上。

虽然没有单元版本 Ver.1.09 中追加的运动控制功能，但对部分指令规格进行了改善。运动控制指令的版本相关信息详情请参阅 □□ “NJ/NX 系列 指令基准手册 (SBCE-364)”。

## 单元版本 Ver.1.10 中追加、变更的运动控制功能

使用单元版本 Ver.1.10 中追加的以下功能时，请使用 Sysmac Studio Ver.1.12 以上。

功 能	概 要
从轴位置补偿	对同步控制中的从轴进行位置补偿的功能。
合并动作的变更	“加减速超限”设定为[提高加减速速度(将合并切换为等待)]、[异常停止]时，仍将按最大加减速速度进行加减速，并继续合并。
绝对值编码器的原点确定时间	除传统的原点确定外，在 EtherCAT 从站的过程数据通信从未确立状态切换至确立状态时，也可确定原点。
过程数据通信未确立时的当前位置	轴变量的反馈当前位置及指令当前位置为尚未确立过程数据通信前的反馈当前位置。
有无操作模式 (6060Hex) 及操作模式显示 (6061Hex) 映射	根据有无操作模式 (6060Hex) 及操作模式显示 (6061Hex) 映射的组合，动作有所不同。
PDS 状态控制方式的追加	设定因 MC_Power 指令伺服 OFF 时 PDS 状态的切换目标。
主电路电源 OFF 检测的追加	发出伺服锁定指令时或在伺服锁定状态下，可选择是否检测主电路电源 OFF。

使用单元版本 Ver.1.10 中追加的以下功能时，请使用 Sysmac Studio Ver.1.13 以上。

功 能	概 要
多任务运动中的同步控制	多任务运动时，在分配给不同任务的轴间执行同步控制指令的功能。

## 单元版本 Ver.1.11 中追加、变更的运动控制功能

使用单元版本 Ver.1.11 中追加的以下功能时，请使用 Sysmac Studio Ver.1.15 以上。

功 能	概 要
追加单位转换设定的参数	追加参数“减速机使用”“工件侧转 1 圈的移动量”、“工件侧齿轮比”、“电机侧齿轮比”。

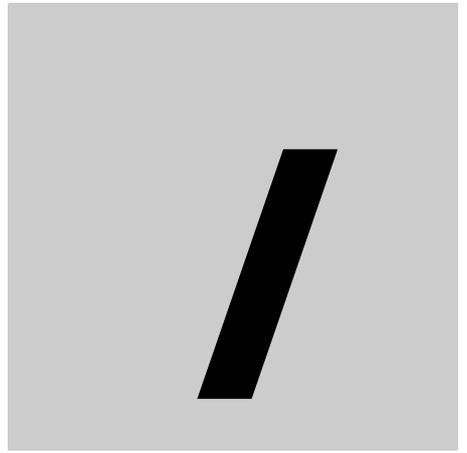
使用单元版本 Ver.1.11 中追加的以下功能时，请使用 Sysmac Studio Ver.1.16 以上。

功 能	概 要
控制对象伺服驱动器	追加欧姆龙制伺服驱动器 1S 系列 EtherCAT 通信内置型。

## 单元版本 Ver.1.13 中追加、变更的运动控制功能

使用单元版本 Ver.1.13 中追加的以下功能时，请使用 Sysmac Studio Ver.1.17 以上。

功 能	概 要
控制功能	选择要控制轴的功能。
下载期间向 I/O 设备发送指令	根据在开始下载时选择停止或持续向 I/O 设备发送指令，MC 功能模块的动作有所不同。



# 索引



# 索引

<b>B</b>	
编码器种类 .....	5-21, 5-23
编码器轴 .....	3-2, 5-8
表示单位 .....	5-11, 5-12
部分停止故障 .....	11-7
轴无效 .....	6-20
不启用轴组 .....	6-27
补偿补偿 .....	9-6
<b>C</b>	
CAN application protocol over EtherCAT (CoE) .....	2-18
插补加减速超限 .....	5-30
插补减速度警告值 .....	5-30
插补加速度警告值 .....	5-30
插补速度警告值 .....	5-30
超调 .....	9-11, 9-56
程序变更凸轮数据 .....	9-15
齿轮动作 .....	9-12
重启 .....	9-39, 9-60
伺服 ON .....	6-21, 7-3
伺服驱动器状态 .....	6-21
伺服准备就绪 .....	6-21
从轴 .....	9-15
错误减速停止中 .....	6-20, 6-27
<b>D</b>	
当前位置变更功能 .....	9-68
单位转换 .....	6-23
单位转换设定 .....	5-11
单轴控制的通用功能 .....	9-30
单轴速度控制 .....	9-27
单轴同步控制 .....	9-12
单轴位置控制 .....	9-3
单轴转矩控制 .....	9-29
到位等待 (轴控制状态) .....	6-21
到位等待 (轴组控制状态) .....	6-27
到位检查 .....	9-74, 9-76
到位检查时间 .....	5-18
到位宽度 .....	5-17
等待 .....	9-45, 9-62
电机转 1 圈的脉冲数 .....	5-11, 6-23
电机转 1 圈的移动量 .....	5-11, 6-23
电子齿轮比 (单位转换的表达式) .....	5-13
定位动作中 .....	6-20
动作设定 .....	5-17
动作中 .....	6-27
多重启动 .....	9-44, 9-61
多轴协调动作中 .....	6-20
多轴协调控制 .....	9-49
多轴协调控制中的停止 .....	9-54
当前位置上限值 .....	6-23
当前位置下限值 .....	6-23
对象词典 .....	2-18
<b>E</b>	
EtherCAT .....	1-2
EtherCAT 主站功能模块 .....	2-2
<b>F</b>	
反馈当前速度 .....	6-22
反馈当前位置 .....	6-22
反馈当前转矩 .....	6-22
反馈速度 .....	9-32
反馈速度滤波时间常数 .....	5-17
反馈位置 .....	6-14, 9-30
反转时动作 .....	5-17
发生源 .....	11-6
发生原因 (推测原因) .....	11-10
负方向极限输入 .....	6-21
负方向极限输入时动作 .....	5-24, 8-8
负方向软件限制 .....	5-21
负方向指令指定中 .....	6-21
负方向转矩警告值 .....	5-17
负方向转矩限制上限值 .....	5-20
附加角 .....	9-64, 9-67
服务数据对象 .....	2-18
<b>G</b>	
高速原点复位 .....	8-16
GetMCErr (运动控制异常状态获取) .....	11-4
功能规格 .....	1-10
构成轴 .....	5-28, 5-29
构成轴 (轴 A1) .....	6-28
构成轴 (轴 A2) .....	6-28
构成轴 (轴 A3) .....	6-28
构成轴 (轴 A0) .....	6-28
固定周期任务 .....	2-5
过程数据通信周期 .....	2-21
过渡模式 .....	9-64
故障诊断功能 .....	11-4
故障诊断器 .....	11-4
过程数据对象 .....	2-18
<b>H</b>	
合并 .....	9-46, 9-62
缓存模式 .....	9-44, 9-61
环计数器上限设定值 .....	5-21, 5-23
环计数器下限设定值 .....	5-21, 5-23

## J

加减速超限	5-17, 9-33
加减运算定位	9-23
监控信息	11-7
减速度的变更	9-41
减速度警告值	5-17, 9-33
减速停止指令值	9-9
减速停止中	6-20, 6-27
加速度的变更	9-41
加速度和减速度的单位	9-33
加速度警告值	5-17, 9-33
加速与减速	9-33
基本数据类型	6-17
节点地址	5-10, 6-23
结构体型	6-17
禁用过渡	9-65
近原点输入	6-21
计数模式	5-21, 5-22, 6-23
极限输入停止方法	5-20
机械原点	8-2
绝对值编码器	
对象伺服电机	8-14
绝对值编码器的安装	8-15
绝对值编码器原点位置偏置	8-14
旋转模式下的使用	8-15
原点设定	8-13
绝对值定位	9-4

## K

空白凸轮数据	9-15
扩展动作设定	5-20

## L

LED	11-3
连接加速度	9-15
连接速度	9-15
连续动作中	6-20
立即停止输入	6-21
立即停止输入停止方法	5-20
立即停止指令值	9-9
立即停止指令值, 同时执行偏差计数器复位	9-9
立即停止指令值, 同时执行伺服 OFF	9-10

## M

_MC_AX[*](轴变量)	6-20
_MC_AX[*].Act.Pos(反馈当前位置)	6-22
_MC_AX[*].Act.Timestamp(时间戳)	6-22
_MC_AX[*].Act.Trq(反馈当前转矩)	6-22
_MC_AX[*].Act.Vel(反馈当前速度)	6-22
_MC_AX[*].Cfg.AxEnable(轴使用)	6-23
_MC_AX[*].Cfg.AxNo(轴号)	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(指令当前转矩)	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(到位等待)	6-21
_MC_AX[*].Details.VelLimit(指令速度饱和)	6-21
_MC_AX[*].Dir.Nega(负方向指令指定中)	6-21
_MC_AX[*].Dir.Posi(正方向指令指定中)	6-21
_MC_AX[*].DrvStatus.CSP (周期同步位置(CSP)模式中)	6-21
_MC_AX[*].DrvStatus.CST (周期同步转矩(CST)模式中)	6-21
_MC_AX[*].DrvStatus.CSV (周期同步速度(CSV)模式中)	6-21
_MC_AX[*].DrvStatus.DrvAlarm(驱动器错误输入)	6-21
_MC_AX[*].DrvStatus.DrvWarning(驱动器警告输入)	6-21
_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)	6-21
_MC_AX[*].DrvStatus.ImdStop(立即停止输入)	6-21
_MC_AX[*].DrvStatus.Latch1(锁定输入 1)	6-21
_MC_AX[*].DrvStatus.Latch2(锁定输入 2)	6-21
_MC_AX[*].DrvStatus.MainPower(主电路电源)	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(伺服准备就绪)	6-21
_MC_AX[*].DrvStatus.ServoOn(伺服 ON)	6-21
_MC_AX[*].MFAultLvl.Active(正在发生轴轻度故障)	6-22
_MC_AX[*].MFAultLvl.Code(轴轻度故障代码)	6-22
_MC_AX[*].Obsr.Active(正在产生轴监控信息)	6-22
_MC_AX[*].Obsr.Code(轴监控信息代码)	6-22
_MC_AX[*].Scale.CountMode(计数模式)	6-23
_MC_AX[*].Scale.Den(电机转 1 圈的移动量)	6-23
_MC_AX[*].Scale.MaxPos(当前位置上限值)	6-23
_MC_AX[*].Scale.MinPos(当前位置下限值)	6-23
_MC_AX[*].Scale.Num(电机转 1 圈的脉冲数)	6-23
_MC_AX[*].Scale.Units(显示单位)	6-23
_MC_AX[*].Status.Continuous(连续动作中)	6-20
_MC_AX[*].Status.Coordinated(多轴协调动作中)	6-20
_MC_AX[*].Status.Disabled(不启用轴)	6-20
_MC_AX[*].Status.Discrete(定位动作中)	6-20
_MC_AX[*].Status.ErrorStop(错误减速停止中)	6-20
_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)	11-5
MC 参数设定	5-2
_MC_COM(MC 通用变量)	6-19
_MC_COM.MFAultLvl.Active (正在发生 MC 通用轻度故障)	6-19
_MC_COM.MFAultLvl.Code(MC 通用轻度故障代码)	6-19
_MC_COM.Obsr.Active (正在产生 MC 通用监控信息)	6-19
_MC_COM.Obsr.Code(MC 通用监控信息代码)	6-19

_MC_COM.PFaultLvl.Active (正在发生 MC 通用部分停止故障)	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 试运行中)	6-19
_MC_ComErrSta(MC 通用异常状态)	11-5
_MC_ErrSta(MC 异常状态)	11-5
_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)	6-27
_MC_GRP[*].Details.InPosWaiting(到位等待)	6-27
_MC_GRP[*].Kinematics.Axis[0](构成轴(轴 A0))	6-28
_MC_GRP[*].Kinematics.Axis[1](构成轴(轴 A1))	6-28
_MC_GRP[*].Kinematics.Axis[2](构成轴(轴 A2))	6-28
_MC_GRP[*].Kinematics.Axis[3](构成轴(轴 A3))	6-28
_MC_GRP[*].Kinematics.GrpType(轴构成)	6-28
_MC_GRP[*].MFaultLvl.Active (正在发生轴组轻度故障)	6-28
_MC_GRP[*].MFaultLvl.Code(轴组轻度故障代码)	6-28
_MC_GRP[*].Obsr.Active(正在产生轴组监控信息)	6-28
_MC_GRP[*].Obsr.Code(轴组监控信息代码)	6-28
_MC_GRP[*].Status.Disabled(不启用轴组)	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)	11-5
MC 试运行功能	4-2
MC 试运行功能一览	4-2
MC 试运行中	6-19
MC 通用变量	6-19
MC 通用 部分停止故障	6-19
MC 通用 部分停止故障代码	6-19
MC 通用 监控信息	6-19
MC 通用 监控信息代码	6-19
MC 通用 轻度故障	6-19
MC 通用 轻度故障代码	6-19
MC 通用异常	11-6
MC 通用状态	6-19
MC 运行中	6-19
枚举体型	6-17
目标速度的变更	9-41
目标位置的变更	9-39
减速超限的模式	9-40
三角控制的模式	9-40
因新的指令值而取反时	9-39

## P

偏差计数器复位	9-73
PLC 功能模块	2-2
PLCopen	1-2, 2-19
PLCopen 运动控制用功能块	6-4
Process Data Objects(PDO)	2-18

## Q

启动速度	5-17, 5-19, 9-32
启动准备完成	6-27
轻度故障	11-7
起始位置指定方式	9-15
启用 / 不启用轴组	9-50
全部停止故障	11-7
驱动器错误复位监视时间	5-20
驱动器错误输入	6-21
驱动器警告输入	6-21
驱动器内部功能限制中	6-21
确定原点	8-2
区域功能	9-70

## R

任务	2-5
任务周期	2-10
软件限制功能	5-21, 9-71

## S

Service Data Objects(SDO)	2-18
事件	11-2
事件代码	11-10
事件名称	11-10
试运行功能	4-2
手动运行	7-2
数据类型	6-17
输入 / 输出设备	5-6, 5-10
停止	
伺服驱动器的输入信号	9-6
伺服驱动器设定	5-23
伺服轴	3-2, 5-8
速度的单位	9-32
速度的种类	9-32
速度警告值	5-17
速度控制	9-27
锁定功能	9-69
S 形	9-35
时间戳	6-22

## T

停止	9-6
发生异常等后停止	9-8, 9-54
极限输入	9-7
立即停止输入	9-6
MC_GroupImmediateStop(轴组立即停止)指令	9-54
MC_GroupStop(轴组强制停止)指令	9-54

MC_ImmediateStop(立即停止)指令	9-7
MC_Stop(强制停止)指令	9-7
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, 执行偏差计数器复位	9-9
立即停止指令值, 同时执行伺服 OFF	9-10
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)	6-27
梯形模式凸轮动作	9-22
同步动作中	6-20
推测原因	11-44
凸轮表	6-29, 9-15, 9-16
创建方法	6-34
属性更新	9-19
凸轮表的保存	9-18
凸轮表的数据类型	9-17
凸轮表规格	9-17
凸轮表名称	6-31
用户程序的指定方法	6-31
凸轮表的切换	9-18
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	9-14, 9-15
凸轮块	9-15
凸轮块起点	9-15
凸轮块终点	9-15
凸轮起点	9-15
凸轮曲线	6-29, 9-15
名称	6-31
凸轮数据	9-15
凸轮数据变量	6-29, 9-15
凸轮数据的读写	9-18
凸轮数据开始位置	9-15
凸轮数据索引	9-15
凸轮终点	9-15
凸轮表生成	9-20

## W

外部锁定输入 1	6-21
外部锁定输入 2	6-21
微动移动	7-5
微动最高速度	5-17
未确定原点	8-2
位移	9-15
位置	9-30
位置的处理	6-14
位置的种类	9-30
位置计数设定	5-21
位置偏差超过值	5-21
位置偏差监视功能	9-72
位置偏差警告值	5-21
位置指定齿轮动作	9-13
无限长轴	5-22
无效凸轮数据	9-15

## X

相对值定位	9-4
相位	9-15
相位间隔宽度	9-15
显示单位	6-23
线性模式	5-22
限制设定	5-21
写入轴组构成轴	9-50
性能规格	1-7
系统定义变量	6-16, 11-5
系统构成	1-3
旋转模式	5-22
虚拟编码器轴	3-2, 5-8
虚拟伺服轴	3-2, 5-8

## Y

衍生数据类型	6-17
一般信息	11-7
异常的确证	11-3
异常状态变量	11-5
异常状态获取指令	11-4
移动距离的变更	9-41
用户程序中轴的指定方法	3-2
用户程序中轴组的指定方法	3-19, 3-22
有限长轴	5-22
有效凸轮数据	9-15
有效凸轮数据数	9-15
原点复位补偿速度	5-24, 8-10
原点复位补偿值	5-24, 8-10
原点复位参数	8-5
原点复位动作	5-24, 8-6
原点复位减速度	5-24, 8-9
原点复位加速度	5-24, 8-9
原点复位接触时间	5-24, 8-10
原点复位接近速度	5-24, 8-9
原点复位开始方向	5-24, 8-7
原点复位设定	5-24
原点复位速度	5-24, 8-9
原点复位跃度	5-24, 8-9
原点复位中	6-20
原点检测方向	5-24, 8-8
原点确定	6-21
原点输入	6-21
原点输入屏蔽距离	5-24, 8-9
原点输入信号	5-24, 8-7
原点停止	6-21
原点位置范围	5-18
原点位置偏置	5-24, 8-10
圆弧插补	9-53
原始恒定周期任务	2-5
原始凸轮数据	9-15
原始周期	2-7, 2-21
跃度	9-35
跃度的单位	9-35
运动控制参数设定	5-2
运动控制程序	6-2
编写方法	6-32

运动控制功能模块	2-2
运动控制系统变量	6-16, 11-5
属性	6-17
一览	6-19
运动控制指令	6-4
重启	9-39, 9-60
多重启动	9-44, 9-61
重启时的时序图	6-12
多重启动时的时序图	6-13
Enable 型	6-10
Enable 型的指令执行时序图	6-11
Execute 型	6-10
Execute 型的指令执行时序图	6-10
启动和状态	6-8
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	6-8
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	2-21
运动学转换设定	6-28
与 EtherCAT 通信之间的关系	2-18
原点附近输入逻辑反转	5-20
跃度	1-2

## Z

正方向极限输入	6-21
正方向极限输入时动作	5-24, 8-8
正方向软件限制	5-21
正方向指令指定中	6-21
正方向转矩警告值	5-17
正方向转矩限制上限值	5-20
正在产生 MC 通用监控信息	6-19
正在产生轴监控信息	6-22
正在产生轴组监控信息	6-28
正在发生 MC 通用部分停止故障	6-19
正在发生 MC 通用轻度故障	6-19
正在发生轴轻度故障	6-22
正在发生轴组轻度故障	6-28
指定动作方向	9-36
无方向指定	9-36
指定为当前方向	9-36
指定为负方向	9-36
指定为附近	9-36
指定为正方向	9-36
指令插补加减速速度	6-28
指令插补速度	6-28
指令当前加减速速度	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
指令当前转矩	6-22
指令方向状态	6-21
指令速度	9-32
指令速度饱和	6-21
指令位置	6-14, 9-30

直线插补	9-52
中断	9-45, 9-61
中断标准定位	9-5
中心点补偿容许率	5-30
重要程度	11-7
轴	3-2
轴变量	3-2, 6-20
轴变量概要	3-6
轴变量与轴种类的关系	6-24
轴参数	3-2
轴当前值	6-22
轴的状态	6-5
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
轴构成	5-28, 6-28
轴号	5-6, 6-23
轴监控信息	6-22
轴监控信息代码	6-22
轴间偏差监控功能	9-73
轴基本设定	5-6, 6-23
轴控制状态	6-21
轴启动准备完成	6-20
轴轻度故障	6-22
轴轻度故障代码	6-22
周期同步定位	9-6
周期同步速度控制	9-28
周期同步速度 (CSV) 模式中	6-21
周期同步位置 (CSP) 模式中	6-21
周期同步转矩 (CST) 模式中	6-21
轴使用	5-6, 6-23
轴异常	11-6
轴指令值	6-22
轴种类	5-6, 5-8, 6-23
轴状态	6-20
轴组	3-19
轴组编号	5-28, 6-28
轴组变量	3-19, 6-27
轴组变量概要	3-21
轴组参数	3-19, 5-27
轴组参数一览表	5-27
轴组错误复位	9-51
轴组状态	6-7
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
轴组动作设定	5-30
轴组概要	3-19
轴组监控信息	6-28
轴组监控信息代码	6-28
轴组基本设定	6-28
轴组控制状态	6-27

轴组 轻度故障 .....	6-28
轴组 轻度故障代码 .....	6-28
轴组使用 .....	5-28, 6-28
轴组停止方法 .....	5-30
轴组位置获取 .....	9-51
轴组异常 .....	11-6
轴组指令值 .....	6-28
轴组轴构成 .....	5-28
轴组周期同步位置控制 .....	9-53
轴组状态 .....	6-27
转矩限制功能 .....	9-69
转矩指令的变更 .....	9-42
主电路电源 .....	6-21
主轴 .....	9-15
主轴跟踪起始位置 .....	9-15
主轴相位补偿 .....	9-24
最大插补减速度 .....	5-30
最大插补加速度 .....	5-30
最大减速度 .....	5-17, 9-33
最大加速度 .....	5-17, 9-33
最大凸轮数据数 .....	9-15
最高插补速度 .....	5-30
最高速度 .....	5-17, 5-18, 9-32
执行 ID .....	6-28
执行 ID .....	6-23
轴错误复位 .....	9-3
周期同步定位 .....	9-6

## 承诺事项

承蒙对欧姆龙株式会社(以下简称“本公司”)产品的一贯厚爱和支持,藉此机会再次深表谢意。  
如果未特别约定,无论贵司从何处购买的产品,都将适用本承诺事项中记载的事项。  
请在充分了解这些注意事项基础上订购。

### 1. 定义

本承诺事项中的术语定义如下。

- (1) “本公司产品”:是指“本公司”的FA系统机器、通用控制器、传感器、电子/结构部件。
- (2) “产品目录等”:是指与“本公司产品”有关的欧姆龙综合产品目录、FA系统设备综合产品目录、安全组件综合产品目录、电子/机构部件综合产品目录以及其他产品目录、规格书、使用说明书、操作指南等,包括以电子数据方式提供的资料。
- (3) “使用条件等”:是指在“产品目录等”资料中记载的“本公司产品”的使用条件、额定值、性能、运行环境、操作使用方法、使用时的注意事项、禁止事项以及其他事项。
- (4) “客户用途”:是指客户使用“本公司产品”的方法,包括将“本公司产品”组装或运用到客户生产的部件、电子电路板、机器、设备或系统等产品中。
- (5) “适用性等”:是指在“客户用途”中“本公司产品”的(a)适用性、(b)动作、(c)不侵害第三方知识产权、(d)法规法令的遵守以及(e)满足各种规格标准。

### 2. 关于记载事项的的注意事项

对“产品目录等”中的记载内容,请理解如下要点。

- (1) 额定值及性能值是在单项试验中分别在各条件下获得的值,并不构成对各额定值及性能值的综合条件下获得值的承诺。
- (2) 提供的参考数据仅作参考,并非可在该范围内一直正常运行的保证。
- (3) 应用示例仅作参考,不构成对“适用性等”的保证。
- (4) 如果因技术改进等原因,“本公司”可能会停止“本公司产品”的生产或变更“本公司产品”的规格。

### 3. 使用时的注意事项

选用及使用本公司产品时请理解如下要点。

- (1) 除了额定值、性能指标外,使用时还必须遵守“使用条件等”。
- (2) 客户应事先确认“适用性等”,进而再判断是否选用“本公司产品”。“本公司”对“适用性等”不做任何保证。
- (3) 对于“本公司产品”在客户系统中的设计用途,客户应负责事先确认是否已进行了适当配电、安装等事项。
- (4) 使用“本公司产品”时,客户必须采取如下措施:(i)相对额定值及性能指标,必须在留有余量的前提下使用“本公司产品”,并采用冗余设计等安全设计(ii)所采用的安全设计必须确保即使“本公司产品”发生故障时也可将“客户用途”中的危险降到最小程度、(iii)构建随时提示使用者危险的完整安全体系、(iv)针对“本公司产品”及“客户用途”定期实施各项维护保养。
- (5) “本公司产品”是作为应用于一般工业产品的通用产品而设计生产的。如果客户将“本公司产品”用于以下所列用途,则本公司对产品不作任何保证。但“本公司”已表明可用于特殊用途,或已与客户有特殊约定时,另行处理。
  - (a) 必须具备很高安全性的用途(例:核能控制设备、燃烧设备、航空/宇宙设备、铁路设备、升降设备、娱乐设备、医疗设备、安全装置、其他可能危及生命及人身安全的用途)
  - (b) 必须具备很高可靠性的用途(例:燃气、自来水、电力等供应系统、24小时连续运行系统、结算系统、以及其他处理权利、财产的用途等)
  - (c) 具有苛刻条件或严酷环境的用途(例:安装在室外的设备、会受到化学污染的设备、会受到电磁波影响的设备、会受到振动或冲击的设备等)
  - (d) “产品目录等”资料中未记载的条件或环境下的用途
- (6) 除了不适用于上述3.(5)(a)至(d)中记载的用途外,“本产品目录等资料中记载的产品”也不适用于汽车(含二轮车,以下同)。请勿配置到汽车上使用。关于汽车配置用产品,请咨询本公司销售人员。

### 4. 保修条件

“本公司产品”的保修条件如下。

- (1) 保修期限 自购买之日起1年。(但是,“产品目录等”资料中有明确说明时除外。)
- (2) 保修内容 对于发生故障的“本公司产品”,由“本公司”判断并可选择以下其中之一方式进行保修。
  - (a) 在本公司的维修保养服务点对发生故障的“本公司产品”进行免费修理(但是对于电子、结构部件不提供修理服务。)
  - (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。

## 欧姆龙自动化(中国)有限公司

欧姆龙自动化(中国)有限公司北京分公司  
 欧姆龙自动化(中国)有限公司天津分公司  
 欧姆龙自动化(中国)有限公司广州分公司



服务



资讯

技术咨询

网 址: <http://www.fa.omron.com.cn>  
 400咨询热线: 400-820-4535

上海总公司	021-60230333	太原事务所	0351-5229870
南京事务所	025-83240556	天津分公司	022-83191580
徐州事务所	0516-83736516	沈阳事务所	024-22815131
武汉事务所	027-82282145	西安事务所	029-88851505
苏州事务所	0512-68669277	银川联络处	0951-5670076
昆山事务所	0512-50110866	成都事务所	028-86765345
杭州事务所	0571-87652855	重庆事务所	023-68796406
宁波事务所	0574-27888220	大连事务所	0411-39948181
温州事务所	0577-88919195	昆明事务所	0871-63527224
合肥事务所	0551-63454209	兰州事务所	0931-8720101
长沙事务所	0731-84585551	长春事务所	0431-81928301
无锡事务所	0510-85169303	乌鲁木齐事务所	0991-5198587
张家港事务所	0512-56313157	贵阳事务所	0851-84812320
南昌事务所	0791-86304711	广州分公司	020-87557798
郑州事务所	0371-65585192	深圳事务所	0755-26948238
北京分公司	010-57395399	厦门事务所	0592-2686709
唐山事务所	0315-4795118	东莞事务所	0769-22423200
石家庄事务所	0311-86906790	佛山事务所	0757-83305298
济南事务所	0531-82929795	中山事务所	0760-88224545
青岛事务所	0532-66775819	福州事务所	0591-88088551
烟台事务所	0535-6865018	南宁事务所	0771-5531371

汕头事务所 0754-88706001 香港事务所 00852-23753827

### 特约店

注:规格如有变更,恕不另行通知。请以最新产品说明书为准。