



可编程逻辑控制器

MP500系列

指令手册



安全注意事项

- 请在安装、运行、检查、维修设备前仔细阅读本手册。
- 请将此手册放在经常容易看见的地方。

在使用本产品之前...

为了能够更有效的进行操作，请在使用产品之前仔细阅读本安全使用须知。

- ▶ 为了安全、恰当的使用本产品，防止危险和意外发生，我们应该一直关注本安全注意事项。
- ▶ 此安全须知分为“警告”和“注意”两个事项，分别说明如下。



警告

此图标表示如果违反了应用程序的指令将会引起严重的损伤或者死亡



注意

此图标表示如果违反了应用程序的指令将会引起轻微的损伤或者毁坏产品。

另外，根据状况不同甚至以分类为注意的事项也可能导致严重故障。因为强烈建议用户检查比如警告等所有注意事项。

- ▶ 在产品或者用户手册上将会看到如下的标记，说明如下。



请小心，危险可能会发生。



请小心，可能会引起触电。

- ▶ 阅读后请将指令手册放在使用者容易看见并取到的地方。

设计时的安全使用



警告

- ▶ 请在 **PLC** 的外部安装保护电路以便保护整个控制系统远离外部电源或者 **PLC** 模块而引起的系统故障。任何不正常的输出或者运行都可能会对整个系统的安全运行造成一系列的问题。
 - 应当在**PLC**的外部安装保护电路以保护系统免受自然损伤，例如：紧急停止开关，保护回路，上下限位开关，正/反向操作互锁电路等等。
 - 如果在**PLC**运行的过程中检测到系统错误（看门狗定时器错误，模块安装错误等），为了系统的安全性，整个的输出将会关闭并且系统停止。但是，当**CPU**错误是由于输出设备自身引起的，例如继电器或者晶体管不能被检测到，那么输出将会保持**ON**，这将会引起严重的问题，因此，应当安装额外的回路来监控输出的状态。
 - ▶ 请不要连接超出输出模块额定负载，也不允许输出回路有短路的情况出现，否则，可能会引起火灾。
 - ▶ 请不要让输出回路的外部供应电源上电的时间早于 **PLC** 系统的供应电源，这可能会引起不正常的输出或者运行。
 - ▶ 在电脑或者其他外部设备和 **PLC** 通过通讯进行数据交换，或者进行 **PLC** 的其它操作（比如：运行模式改变）的时候，请在顺序程序中安装互锁以避免系统出现错误。否则，可能会引起不正常的输出或运行。

设计时的安全使用



注意

▶ I/O 信号或者通讯的接线应当远离高压电缆或者电源电缆至少 100mm。

否则，可能会引起不正常的输出或者运行。

设计时的安全使用



注意

- ▶ 应当在符合 PLC 手册或者在标准的环境下使用 PLC 系统。否则，可能会引起触电，火灾，不正常的运行或者引起电弧。
- ▶ 在安装模块之前，请确保 PLC 的供应电源已经关闭。否则，可能会引起触电或者导致产品损坏。
- ▶ 请确保 PLC 系统中的每一个模块都是正确的安装。如果模块的安装出现松动或者不正常的现象，可能会引起系统的不正常的运行，错误或者模块掉下。
- ▶ 请确保 I/O 或者扩展连接器都是正确的安装。否则，可能会引起触电，火灾或者系统的不正常运行。
- ▶ 如果在安装的环境中出现大量的震动，请不要直接让 PLC 承受震动，否则，可能会引起触电，火灾或者系统不正常的运行。
- ▶ 请不要让金属杂物进入产品内部。这可能会引起触电，火灾或者系统的不正常的运行。

配线时的安全使用



警告

- ▶ 在接线之前，请确保 **PLC** 的供应电源和外部的电源系统全部都处于关断状态。否则，可能会引起触电或者产品的损坏
- ▶ 在 **PLC** 的系统上电之前，请确保所有接线端子上的盖板都已经正确关闭。否则可能会引起触电。



注意

- ▶ 在检查完每一个产品的额定电压和接线端子的排列之后，应当确认所有的接线都是正确安装的。否则，可能会引起触电，火灾，或者系统的不正常运行。
- ▶ 在安装接线的时候请确保接线端子上的所有螺丝都已经拧紧。如果接线端子上的螺丝有松动，可能会引起短路，火灾或者系统的不正常运行。
- * ▶ 请确保 **FG** 接线端子上使用 **3 级** 接地方式，这是专门为 **PLC** 设计的。如果端子上的接地不正确，可能会导致系统的不正常运行。
- ▶ 在接线的过程中不要让杂物（如：线头，杂物）进入模块内。否则可能会引起火灾，产品的损坏或者系统的不正常运行。
- ▶ 请确保接线端子已经使用相应的扭矩拧紧。确保外部连接器使用相应的设备进行压紧或者固定。

测试运行和维修时的安全使用

警告

- ▶ 上电的以后请不要触摸接线端子。这可能会引起触电或者系统的不正常运行。
- ▶ 在清理或者紧固接线端子的螺丝之前，关闭外部的供应电源包括 **PLC** 供应的电源。否则，可能会引起触电或者系统的不正常运行。
- ▶ 不要对电池进行充电，分解，加热，短路或者焊接。这可能会引起爆炸或者明火，导致伤害或者火灾。

注意

- ▶ 不要将 **PCB** 板从模块中取出或者改变模块的外形，这可能会引起火灾，触电或者系统的不正常运行。
- ▶ 在安装或者拆离模块之前，应当让所有外部的供应电源包括 **PLC** 的供应电源关闭。否则，可能会引起触电或者系统的不正常运行。
- ▶ 请让所有的无线装置或电话远离 **PLC** 至少 **30cm**。否则，可能会引起系统的不正常运行。
- ▶ 请在充分阅读和理解本手册所有内容后，在 **PLC** 运行时进行修改程序或者使用运行中修改功能。不正确的使用将导致产品的毁坏或者事故。
- ▶ 请避免对电池进行很强的冲击以及避免跌落电池。电池的毁坏将导致液体的泄露。电池跌落或者受到很强的冲击时，请不要再使用电池。需要更换电池时请技术熟练的技术员来更换电池。

废物处理时的安全使用



注意

▶ 产品或者电池应当作为工业垃圾进行处理。

产品的废物自身可能会放出有毒的原料或者爆炸。

◎目录◎

第 1 章概述	1-1 ~ 1-10
1.1 特性	1 - 1
1.2 运动控制模块的运行图	1 - 3
1.3 运动功能模块的功能概述	1 - 4
1.3.1 位置控制	1 - 4
1.3.2 插值控制	1 - 5
1.3.3 速度控制	1 - 9
1.3.4 转矩控制	1 - 10
第 2 章规格	2-1 ~ 2-14
2.1 一般规格	2 - 1
2.2 电源规格	2 - 2
2.3 性能规格	2 - 3
2.3.1 功能规格	2 - 3
2.3.2 通讯规格	2 - 6
2.3.3 内部输入/输出规格	2 - 7
2.3.4 编码器输入规格	2 - 9
2.4 各部分名称	2 - 10
2.4.1 各部分名称	2 - 10
2.4.2 外部设备接口规格	2 - 11
第 3 章操作顺序与安装	3-1 ~ 3-35
3.1 操作顺序	3 - 1
3.2 安装	3 - 2
3.2.1 安全措施	3 - 2
3.2.2 安装环境	3 - 4
3.2.3 操作注意事项	3 - 4
3.2.4 安装/拆卸运动控制模块	3 - 5
3.3 配线注意事项	3 - 10
3.3.1 配线注意事项	3 - 10
3.3.2 电源配线	3 - 11
3.3.3 I/O设备配线	3 - 12
3.3.4 接地(LG)线	3 - 13
3.3.5 接线电缆规格	3 - 14
3.3.6 编码器输入(DC 5V电压输出)配线实例	3 - 15
3.3.7 编码器输入(DC 5V线性驱动输出)配线实例	3 - 16
3.3.8 外部输入信号配线实例	3 - 17

3.3.9 外部输出信号配线实例	3 - 17
3.4 EMC	3 - 18
3.4.1 EMC标准.....	3 - 18
3.5 故障保护.....	3 - 22
3.5.1 故障保护回路.....	3 - 22
3.6 维护	3 - 24
3.6.1 维护和检测.....	3 - 24
3.6.2 日常检测.....	3 - 24
3.6.3 定期检查.....	3 - 25
3.7 故障排除.....	3 - 26
3.7.1 故障排除基本步骤	3 - 26
3.7.2 故障排除.....	3 - 26
3.7.3 故障排除实例.....	3 - 32

第 4 章运动控制运行	4-1 ~ 4-13
-------------------	------------

4.1 运动控制模块的结构	4 - 1
4.2 运动控制配置.....	4 - 2
4.3 运动控制任务.....	4 - 3
4.3.1 任务类型.....	4 - 3
4.3.2 任务运行.....	4 - 4
4.3.3 运动指令的执行	4 - 7
4.4 EtherCAT通信	4 - 8
4.4.1 什么是EtherCAT	4 - 8
4.4.2 COE(CANopen覆盖EtherCAT)	4 - 8
4.4.3 EtherCAT状态机	4 - 9
4.4.4 EtherCAT过程数据目标(PDO)	4 - 10
4.4.5 运动控制模块EtherCAT通讯规格	4 - 11
4.5 运动控制程序.....	4 - 12
4.5.1 程序运行.....	4 - 12
4.5.2 运行模式.....	4 - 12

第5 章内存和参数	5-1 ~ 5-38
-----------------	------------

5.1 内存	5 - 1
5.1.1 程序和Data存储.....	5 - 1
5.1.2 设备	5 - 2
5.1.3 参数	5 - 8

第 6 章运动功能块	6-1 ~ 6-195
------------------	-------------

6.1 运动功能块通用要素	6 - 1
---------------------	-------

6.1.1	轴状态.....	6 - 1
6.1.2	组状态.....	6 - 3
6.1.3	基本I/O变量.....	6 - 4
6.1.4	缓存模式输入.....	6 - 7
6.1.5	运动功能块执行中改变参数.....	6 - 7
6.1.6	组运行路径改变设置.....	6 - 7
6.1.7	运动功能块错误.....	6 - 10
6.2	运动功能块.....	6 - 11
6.3	单轴运动功能块.....	6 - 14
6.3.1	Servo On/Off (MC_Power)	6 - 14
6.3.2	执行搜索原点(MC_Home)	6 - 15
6.3.3	立即停止 (MC_STOP)	6 - 18
6.3.4	Stop (MC_Halt)	6 - 19
6.3.5	绝对位置运行(MC_MoveAbsolute)	6 - 20
6.3.6	相对位置控制运行(MC_MoveRelative)	6 - 23
6.3.7	增加位控运行(MC_MoveAdditive)	6 - 26
6.3.8	定速运行(MC_MoveVelocity)	6 - 29
6.3.9	绝对位置运行以定速运行结束(MC_MoveContinuousAbsolute)	6 - 32
6.3.10	相对位置运行以定速结束(MC_MoveContinuousRelative)	6 - 36
6.3.11	转矩控制(MC_TorqueControl)	6 - 39
6.3.12	设置当前位置(MC_SetPosition)	6 - 41
6.3.13	速度/加速覆盖(MC_SetOverride)	6 - 43
6.3.14	读参数(MC_ReadParameter)	6 - 45
6.3.15	写参数(MC_WriteParameter)	6 - 48
6.3.16	轴错误复位(MC_Reset)	6 - 51
6.3.17	接触探头(MC_TouchProbe)	6 - 52
6.3.18	中止触发事件(MC_AbortTrigger)	6 - 56
6.3.19	叠加运行(MC_MoveSuperImposed)	6 - 57
6.3.20	叠加运行停止(MC_HaltSuperImposed)	6 - 58
6.4	多轴运动功能块.....	6 - 59
6.4.1	凸轮运行(MC_CamIn)	6 - 59
6.4.2	凸轮停止(MC_CamOut)	6 - 66
6.4.3	电子传动运行(MC_GearIn)	6 - 69
6.4.4	电子传动脱离(MC_GearOut)	6 - 72
6.4.5	指定位置电子传动(MC_GearInPos)	6 - 74
6.4.6	相位补偿(MC_Phasing)	6 - 79
6.5	组运动功能块.....	6 - 81
6.5.1	在组AxesGroup结构中增加一个轴 (MC_AddAxisToGroup)	6 - 81
6.5.2	在AxesGroup组结构中移除一个轴(MC_RemoveAxisFromGroup)	6 - 82
6.5.3	从组AxesGroup中移除所有轴(MC_UngroupAllAxes)	6 - 83

6.5.4	从GroupDisabled到GroupEnable改变组的状态(MC_GroupEnable)	6 - 84
6.5.5	改变组为GroupDisabled状态(MC_GroupDisable)	6 - 85
6.5.6	AxesGroup执行搜索原点顺序(MC_GroupHome)	6 - 86
6.5.7	不移动设置组中所有轴的位置(MC_GroupSetPosition)	6 - 87
6.5.8	立即停止组(MC_GroupStop)	6 - 89
6.5.9	停止组(MC_GroupHalt)	6 - 90
6.5.10	复位组错误(MC_GroupReset)	6 - 91
6.5.11	绝对位控线性插补运行(MC_MoveLinearAbsolute)	6 - 92
6.5.12	相对位置控制线性插补运行(MC_MoveLinearRelative)	6 - 96
6.5.13	绝对位控圆环插补运行(MC_MoveCircularAbsolute)	6 - 100
6.5.14	圆弧插补相对位控运行(MC_MoveCircularRelative)	6 - 105
6.6	专用功能块	6 - 110
6.6.1	连接伺服驱动(LS_Connect)	6 - 110
6.6.2	断开伺服驱动(LS_Disconnect)	6 - 111
6.6.3	读取 SDO (LS_ReadSDO)	6 - 112
6.6.4	写入 SDO (LS_SDO)	6 - 113
6.6.5	保存 SDO (LS_SaveSDO)	6 - 114
6.6.6	编码器预设(LS_EncoderPreset)	6 - 115
6.6.7	JOG运行(LS_Jog)	6 - 116
6.6.8	读取凸轮数据(LS_ReadCamData)	6 - 118
6.6.9	写入凸轮数据(LS_WriteCamData)	6 - 119
6.6.10	读取ESC (LS_ReadEsc)	6 - 120
6.6.11	写入ESC (LS_WriteEsc)	6 - 122
6.6.12	跳跃凸轮(LS_CamSkip)	6 - 125
6.6.13	Var凸轮运行(LS_VarCamIn)	6 - 126
6.6.14	变量齿轮运行(LS_VarGearIn)	6 - 128
6.6.15	变量定位齿轮运行(LS_VarGearInPos)	6 - 130
6.6.16	读取凸轮表从站位置(LS_ReadCamTableSlavePos)	6 - 132
6.6.17	写入变频器速度(LS_InverterWriteVel)	6 - 133
6.6.18	读取变频器速度(LS_InverterReadVel)	6 - 134
6.6.19	写入变频器控制字(LS_InverterControl)	6 - 135
6.6.20	读取变频器状态1 (LS_InverterStatus1)	6 - 138
6.6.21	读取变频器状态2 (LS_InverterStatus2)	6 - 139
6.6.22	速度控制运行(CSV模式) (LS_SyncMoveVelocity)	6 - 140
6.7	坐标系运行功能块	6 - 141
6.7.1	机械信息设置(MC_SetKinTransform)	6 - 141
6.7.2	PCS设置(MC_SetCartesianTransform)	6 - 144
6.7.3	工作区设置(LS_SetWorkspaceTransform)	6 - 146
6.7.4	坐标系统绝对位置的时间-线性插值运行(LS_MoveLinearTimeAbsolute)	6 - 149
6.7.5	坐标系统相对位置的时间-线性插值运行(LS_MoveLinearTimeRelative)	6 - 151

6.7.6	坐标系统绝对位置的圆弧插值运行(MC_MoveCircularAbsolute2D)	6 - 153
6.7.7	坐标系统相对位置的圆弧插值运行(MC_MoveCircularRelative2D)	6 - 157
6.7.8	传送带同步设置(MC_TrackConveyorBelt)	6 - 161
6.7.9	转盘同步设置(MC_TrackRotaryTable)	6 - 163
6.7.10	坐标系统JOG运行(MC_RobotJog)	6 - 164
6.7.11	设置路径运行数据(MC_SetMovePath)	6 - 166
6.7.12	删除路径运行数据(MC_RestMovePath)	6 - 168
6.7.13	读取路径运行数据(MC_GetMovePath)	6 - 169
6.7.14	执行路径运行(MC_RunMovePath)	6 - 171
6.8	NC控制功能块	6 - 172
6.8.1	指定NC程序(NC_LoadProgram)	6 - 172
6.8.2	指定块运行(NC_BlockControl)	6 - 173
6.8.3	复位(NC_Reset)	6 - 174
6.8.4	紧急停止(NC_Emergency)	6 - 176
6.8.5	启动自动运行(NC_CycleStart)	6 - 177
6.8.6	进给保持(NC_FeedHold)	6 - 178
6.8.7	原点返回(NC_Home)	6 - 179
6.8.8	快速移位覆盖(NC_RapidTraverseOverride)	6 - 180
6.8.9	进刀覆盖(NC_CuttingFeedOverride)	6 - 181
6.8.10	主轴覆盖(NC_SpindleOverride)	6 - 182
6.8.11	M代码运行完成(NC_McodeComplete)	6 - 183
6.8.12	S代码运行完成(NC_ScodeComplete)	6 - 184
6.8.13	T代码运行完成(NC_TcodeComplete)	6 - 185
6.8.14	读取NC参数(NC_ReadParameter)	6 - 186
6.8.15	写入NC参数(NC_WriteParameter)	6 - 195

第7 章程序.....	7-1 ~ 7-44
-------------	------------

7.1	程序结构	7 - 1
7.1.1	程序结构	7 - 1
7.1.2	如何设置程序	7 - 2
7.1.3	程序运行时间	7 - 4
7.2	状态信息读取	7 - 6
7.3	离散运动程序	7 - 7
7.3.1	运行准备	7 - 7
7.3.2	原点返回运行	7 - 8
7.3.3	绝对位置/相对位置运行	7 - 10
7.3.4	速度/转矩控制运行	7 - 13
7.3.5	轴停止	7 - 16
7.3.6	错误处理	7 - 18
7.3.7	运行中改变	7 - 20

7.3.8 参数写入读取.....	7 - 22
7.4 多轴运行程序.....	7 - 27
7.4.1 线性插值运行.....	7 - 27
7.4.2 圆弧插值运行.....	7 - 29
7.4.3 同步运行.....	7 - 31
7.4.4 CAM运行.....	7 - 34
7.4.5 轴组处理.....	7 - 36
7.4.6 轴组运行实例.....	7 - 38
7.5 I/O处理程序.....	7 - 44
7.5.1 输入信号处理.....	7 - 44
7.5.2 输出信号处理.....	7 - 44

第8 章运动控制功能	8-1 ~ 8-112
------------------	-------------

8.1 原点决定.....	8 - 1
8.1.1 原点决定.....	8 - 1
8.1.2 原点返回.....	8 - 2
8.2 控制运行类型.....	8 - 9
8.2.1 单轴位置控制.....	8 - 9
8.2.2 单轴速度控制.....	8 - 14
8.2.3 单轴转矩控制.....	8 - 16
8.2.4 位置运动后定速运行	8 - 18
8.2.5 开关控制.....	8 - 20
8.2.6 轴组控制.....	8 - 22
8.2.7 线性插值控制.....	8 - 24
8.2.8 圆弧插值控制.....	8 - 29
8.2.9 轴控制缓冲模式.....	8 - 37
8.2.10 轴组控制缓冲模式和过渡模式.....	8 - 40
8.2.11 同步控制.....	8 - 42
8.2.12 手动控制.....	8 - 53
8.2.13 叠加运行.....	8 - 55
8.2.14 相位校正控制.....	8 - 57
8.3 其他功能.....	8 - 59
8.3.1 转换控制功能.....	8 - 59
8.3.2 控制辅助功能.....	8 - 66
8.3.3 数据管理功能.....	8 - 71
8.3.4 EtherCAT通讯诊断功能.....	8 - 76
8.3.5 电缆复制功能.....	8 - 81
8.3.6 连接期间的替换功能	8 - 82
8.4 坐标系统运行功能.....	8 - 83
8.4.1 坐标系统运行概要	8 - 83

8.4.2 ACS/MCS/PCS/TCP	8 - 83
8.4.3 PCS设置	8 - 84
8.4.4 机械信息设置.....	8 - 85
8.4.5 工作区设置.....	8 - 88
8.4.6 坐标系统绝对位置的时间线性插值运行	8 - 90
8.4.7 坐标系统的圆弧插值运行.....	8 - 93
8.4.8 传送带的同步运行	8- 101
8.4.9 转盘的同步运行	8- 104
8.4.10 坐标系统的路径运行功能	8- 107
8.5 FoE(通过EtherCAT的文件访问)功能	8 - 109
8.5.1 FoE功能概述	8 - 109
8.5.2 FoE下载.....	8 - 109

第9章 NC控制功能	9-1 ~ 9-72
-------------------------	-------------------

9.1 NC 指令.....	9 - 1
9.1.1 NC指令定义.....	9 - 1
9.1.2 控制符定义.....	9 - 1
9.1.3 坐标系.....	9 - 3
9.2 程序配置.....	9 - 5
9.2.1 NC程序.....	9 - 5
9.2.2 NC程序配置.....	9 - 5
9.2.3 数据	9 - 8
9.3 NC指令.....	9 - 10
9.3.1 NC位置指令基本格式	9 - 10
9.3.2 NC指令列表.....	9 - 12
9.3.3 NC指令描述.....	9 - 14
9.4 NC 参数.....	9 - 62

第10章 CPU功能	10-1 ~ 10-45
-------------------------	---------------------

10.1 任务设计.....	10 - 1
10.1.1 任务概述.....	10 - 1
10.1.2 任务规格.....	10 - 2
10.1.3 任务基本操作.....	10 - 2
10.1.4 任务执行顺序实例.....	10 - 4
10.1.5 系统服务处理.....	10 - 5
10.1.6 程序占有率运行.....	10 - 6
10.1.7 任务设置项目.....	10 - 9
10.1.8 如何使用任务间变量的方式	10 - 10
10.1.9 任务标志位.....	10 - 11
10.1.10 任务-相关警告/错误.....	10 - 12

10.2	参数设置.....	10 - 14
10.2.1	基本参数设置.....	10 - 14
10.2.2	I/O参数设置.....	10 - 16
10.3	自诊断功能.....	10 - 18
10.3.1	主任务/周期性任务周期错误.....	10 - 18
10.3.2	任务程序占有率超出警告/错误.....	10 - 19
10.3.3	错误历史存储功能.....	10 - 19
10.3.4	失效管理.....	10 - 20
10.3.5	外部设备故障诊断功能.....	10 - 23
10.3.6	瞬时断电保护功能.....	10 - 24
10.4	RTC功能.....	10 - 25
10.4.1	如何使用RTC.....	10 - 25
10.5	远程功能.....	10 - 29
10.6	I/O强制On/Off功能.....	10 - 30
10.6.1	强制I/O设置方式.....	10 - 30
10.6.2	处理强制I/O On/Off的时间和处理方式.....	10 - 31
10.7	保存运行历史功能.....	10 - 32
10.7.1	错误历史.....	10 - 32
10.7.2	模式转换历史.....	10 - 32
10.7.3	断电历史.....	10 - 32
10.7.4	系统历史.....	10 - 32
10.7.5	错误动作历史.....	10 - 32
10.8	运行期间程序修改(RUN期间修改).....	10 - 33
10.8.1	RUN期间修改步骤.....	10 - 33
10.9	读取I/O信息.....	10 - 36
10.10	监控功能.....	10 - 37
10.11	运动控制器全部删除功能.....	10 - 40
10.12	内置输入/输出功能.....	10 - 41
10.12.1	输入滤波功能.....	10 - 41
10.12.2	紧急输出功能.....	10 - 43
10.13	序列号信息读取.....	10 - 44

第 11 章数据记录功能	11-1 ~ 11-108
---------------------------	----------------------

11.1	概要.....	11 - 1
11.1.1	特征.....	11 - 1
11.1.2	各部分的名称.....	11 - 2
11.1.3	动作顺序.....	11 - 3
11.1.4	控制信号流程图.....	11 - 4
11.2	性能规格.....	11 - 5
11.3	详细功能.....	11 - 6

11.3.1	数据类型和设备.....	11 - 6
11.3.2	数据保存方式.....	11 - 11
11.3.3	数据采样条件.....	11 - 13
11.3.4	保存文件夹构成.....	11 - 24
11.3.5	CSV文件格式.....	11 - 25
11.3.6	CSV文件保存方法.....	11 - 28
11.3.7	缓存内存.....	11 - 29
11.3.8	数据丢失.....	11 - 30
11.3.9	文件备份周期.....	11 - 31
11.4	一般保存.....	11 - 32
11.4.1	主任务周期保存方式.....	11 - 32
11.4.2	指定周期保存方式.....	11 - 35
11.4.3	指定时间保存方式.....	11 - 38
11.5	触发保存.....	11 - 42
11.5.1	触发条件.....	11 - 43
11.5.2	触发采样块改善方法.....	11 - 52
11.5.3	触发采样计算方法.....	11 - 53
11.5.4	触发采样保存周期.....	11 - 53
11.5.5	触发采样保存区间.....	11 - 54
11.5.6	设定方法.....	11 - 55
11.6	事件保存.....	11 - 66
11.6.1	事件条件.....	11 - 67
11.6.2	设定方法.....	11 - 82
11.7	附件功能.....	11 - 93
11.7.1	文件保存履历设定.....	11 - 93
11.7.2	格式化功能.....	11 - 94
11.7.3	诊断功能.....	11 - 97
11.8	CSV文件构成.....	11 - 98
11.8.1	文件保存格式.....	11 - 98
11.8.2	文件类型和保存顺序.....	11 - 98
11.9	Micro SD内存卡.....	11 - 99
11.9.1	MicroSD内存规格.....	11 - 99
11.9.2	使用时的注意事项.....	11 - 100
11.9.3	Micro SD内存使用容量.....	11 - 100
11.10	标志位一览.....	11 - 101
11.10.1	共通标志位.....	11 - 101
11.10.2	组别标志位.....	11 - 102
11.10.3	错误代码及对应方案.....	11 - 104
11.11	数据记录性能.....	11 - 105
11.11.1	数据记录保存性能.....	11 - 105

11.11.2 主任务周期的保存性能.....	11 - 105
11.11.3 保存处理时间确认.....	11- 106

第 12 章 SD附件功能	12-1 ~ 12-12
----------------------------	---------------------

12.1 概要.....	12 - 1
12.1.1 特征.....	12 - 1
12.1.2 SD卡导出功能.....	12 - 2
12.1.3 SD卡输入功能.....	12 - 5
12.1.4 PLC更新功能.....	12 - 5
12.1.5 PLC备份功能.....	12 - 7
12.1.6 PLC对比功能.....	12 - 8
12.1.7 PLC Boot运行功能.....	12 - 10
12.1.8 自动记录功能.....	12 - 12
12.1.9 错误代码及对应方案.....	12 - 12

第 13 章 内置模拟量功能	13-1 ~ 13-52
-----------------------------	---------------------

13.1 概要.....	13 - 1
13.2 模拟部分的名称和作用.....	13 - 4
13.3 I/O控制特性.....	13 - 5
13.3.1 输入特性.....	13 - 5
13.3.1 输出特性.....	13 - 7
13.4 精度.....	13 - 9
13.4.1 输入精度.....	13 - 9
13.4.2 输出精度.....	13 - 10
13.5 内置模拟量功能.....	13 - 11
13.5.1 采样处理.....	13 - 11
13.5.2 滤波功能.....	13 - 12
13.5.3 平均功能.....	13 - 13
13.5.4 警告功能(输入断线检出功能).....	13 - 15
13.5.5 有效变换值维持功能.....	13 - 16
13.5.6 警报功能.....	13 - 17
13.5.7 通道输出状态设定.....	13 - 18
13.5.8 插补方法设定.....	13 - 18
13.6 配线.....	13 - 21
13.6.1 模拟输入配线示例.....	13 - 21
13.6.2 模拟输出配线示例.....	13 - 23
13.7 运行参数设定.....	13 - 24
13.8 特殊模块监控功能.....	13 - 27
13.9 模块变量自动登录.....	13 - 31
13.10 内部内存的构成和功能.....	13 - 36

13.10.1 内置模拟数据输入输出领域	13 - 36
13.11 示例程序	13 - 48
13.12 故障排除	13 - 50
13.12.1 错误时LED状态区分	13 - 50
13.12.2 内置模拟状态确认	13 - 50
13.12.3 故障诊断及处理方法	13 - 51

第 14 章本地Ethernet功能	14-1 ~ 14-32
---------------------------------	---------------------

14.1 本地Ethernet功能	14 - 1
14.1.1 本地Ethernet参数设置	14 - 1
14.1.2 本地Ethernet和XG5000连接	14 - 4
14.1.3 本地Ethernet和XGT服务器连接	14 - 5
14.1.4 本地Ethernet和TCP/IP服务器连接	14 - 5
14.1.5 本地Ethernet诊断信息功能	14 - 7
14.2 FTP服务器信息	14 - 9
14.2.1 框架	14 - 9
14.2.2 支持功能	14 - 9
14.2.3 设置FTP服务器参数	14 - 10
14.2.4 方位FTP服务器	14 - 12
14.2.5 防火墙设置	14 - 21
14.2.6 FTP加速	14 - 24
14.3 SNTP客户端功能	14 - 26
14.3.1 时间同步协议概要	14 - 26
14.3.2 SNTP参数设置	14 - 26
14.3.3 设置本地NTP服务器	14 - 28

附录 1 标志位列表	A1-1 ~ A1-14
-------------------------	---------------------

附录 2 错误信息&解决方法	A2-1 ~ A2-44
---------------------------------	---------------------

附录3 设置示例	A3-1 ~ A3-17
-----------------------	---------------------

附录4 尺寸	A4-1
---------------------	-------------

附录5 ESC (EtherCAT从站控制器)注册	A5-1 ~ A5-4
--	--------------------

第 1 章概述

本手册描述了运动控制模块的标准，安装方法，每个功能、编程和外部设备接线的方法。

1.1 特性

运动控制模块的特性如下。

- (1) 多种运动控制功能
运动控制系统需要多种功能例如位置控制，速度控制等。
 - (2) 支持多种运动控制指令。
 - 1) 支持多个功能块。
 - 2) 支持多个符合 PLC 开放标准的运动功能模块。
 - 3) 支持在 MP500 中以 LD 或者 ST 形式的运动程序。
 - (a) 最多可控制 32 个实轴/虚轴，4 个虚轴和 64 个从站 EtherCAT I/O，并且支持数字输入 8 点，数字输出 16 点，模拟量输出 2 通道，和 2 通道编码器输入。
 - (b) 适用多种单轴运行。
 - 1) 位置控制
 - 2) 速度控制
 - 3) 同步控制
 - 4) 多轴同步启动
 - (c) 适用多种多轴运行。
 - 1) 圆弧插补
 - 2) 线性插补
 - 3) 螺旋插补
 - 4) 组原点返回/变更组位置
 - (d) 允许在运行中进行开关控制。
 - 1) 位置/速度控制开关
 - 2) 位置/转矩控制开关
 - 3) 速度/转矩控制开关
 - (e) 允许 Cam 控制。
可通过多种 MP500 的 cam 轮廓创建最多 32 种 cam 数据。
 - (f) 多种原点返回控制功能。
关于原点返回方式，可使用每个伺服驱动模式支持的原点返回方式。
(更多关于原点返回方式和伺服参数设置的内容请参考伺服驱动的说明书)
 - (g) 支持加速/减速方式，梯形加速/减速和 S 型加速/减速，并且 S 型加速/减速可通过在运动功能块上设置加速度来使用。
- (3) 运行程序执行加速
通过在开始运行的时候执行进程加速，在主任务中设置的运动程序可以以最多 0.5ms 的间隔执行。另外，在同步启动和插补启动中轴之间没有延迟时间。

第 1 章概述

- (4) 通过 EtherCAT*¹ 连接伺服驱动
 - (a) 通过 EtherCAT 实现直接连接最多 32 个伺服驱动和最多 64 个 EtherCAT I/O.
 - (b) 由于使用以太网电缆连接运动控制模块和伺服驱动，所以连接非常简单.
 - (c) 可以在运动控制模块中很便捷的检查 and 建立伺服驱动信息和参数
 - (d) 最大连接距离为 100m.
- (5) 可实现绝对位置系统
可以仅通过使用绝对位置编码器连接到伺服驱动，建立绝对位置系统，如果通过 ON/OFF，可以在没有原点返回的情况下了解电机的当前位置.
- (6) 易维护
为了在没有运动控制模块的情况下，保持寄存器，参数，cam 数据和位置数据保存到 MRAM (随机存储器)，数据可在无延迟情况下保存，并且没有写入的数量限制.
- (7) 在软件包 MP500 中可用自诊断，监控和测试.
 - (a) 监控功能(模块&伺服驱动)
 - (b) 跟踪功能
 - (c) 趋势功能
 - (d) 读取和保存模块程序/参数
 - (e) 读取和保存伺服参数
 - (f) 创建 CAM 数据
 - (g) 提供错误的详细信息和改善对策
 - (h) 多种形式的打印功能

备注

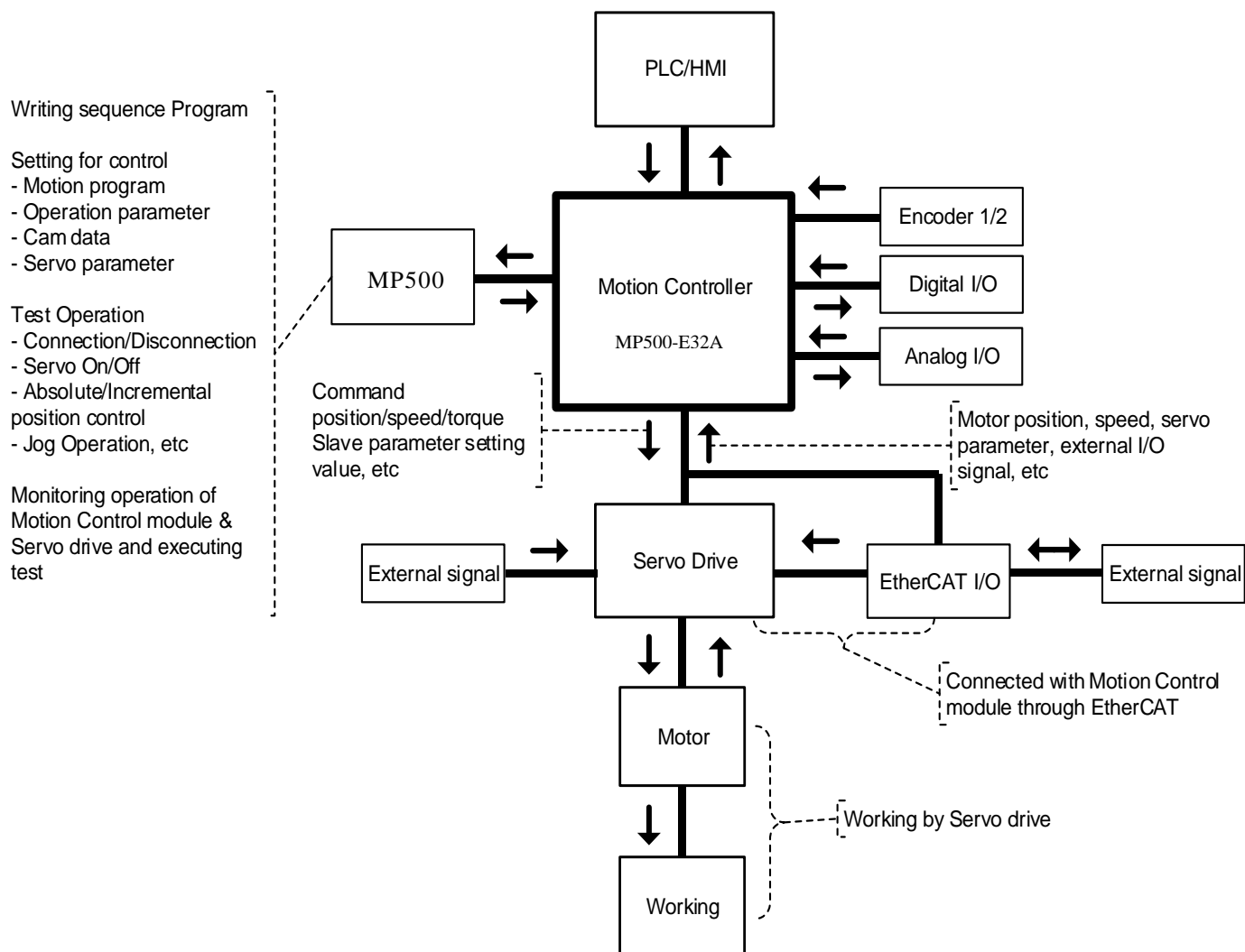
什么是EtherCAT?

EtherCAT，开放工业以太网解决方案，由德国倍福在2002年到2003年11月EtherCAT技术小组研发并公开其技术。在2005年2月，被授权为IEC标准规范。由于其快速控制速度和便于使用和维护，被广泛应用于工业领域，符合本位置模块的性能要求，服务驱动的数据通讯通过EtherCAT 以主从式方法连接，并使用以太网电缆。

对于位置模块，服务驱动器之间的数据通讯是通过 EtherCAT 的主战-从站方式，并通过使用以太网电缆实现。

1.2 运动控制模块的运行图

运动控制去系统流程如下.



1.3 运动控制模块的功能概述

简要描述运动控制模块的代表功能(坐标&线性插补，圆弧插补&停止)。

1.3.1 位置控制

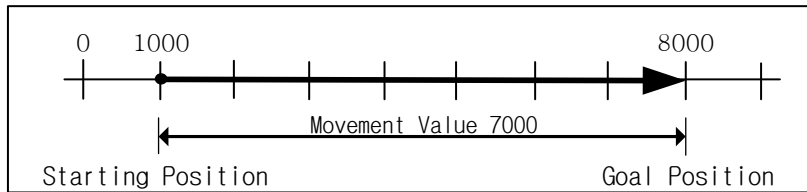
执行指定轴从起始位置(当前位置)到目标位置(需要移动的位置)的位置控制。

(1) 绝对坐标控制

- (a) 执行运动功能块中从起始位置到指定目标位置的位置控制.
- (b) 位置控制的执行基于原点返回中的指定原点位置.
- (c) 移动方向由起始位置和目标位置决定.
 - 1) 起始位置<目标位置：正向位置运行
 - 2) 起始位置>目标位置：反向位置运行

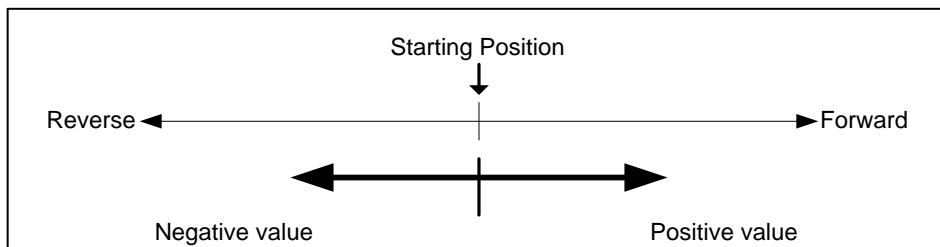
[示例]

- 1) 起始位置：1000
 - 2) 目标位置：8000
- 正向移动值 (7000=8000-1000)



(2) 相对坐标控制

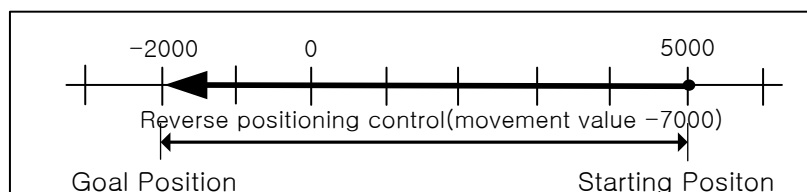
- (a) 从起始位置开始执行目标移动值得位置控制.
与绝对坐标控制的不同在于目标位置是移动值，而不是位置值.
- (b) 移动方向取决于移动值的符号.
 - 1) 正数(+ or 0)：正方向位置运行
 - 2) 负数(-)：反方向位置运行



[示例]

- 1) 起始位置 : 5000
- 2) 目标位置 : -7000

在此条件下, 相对移动并在 -2000 停止.



1.3.2 插补控制

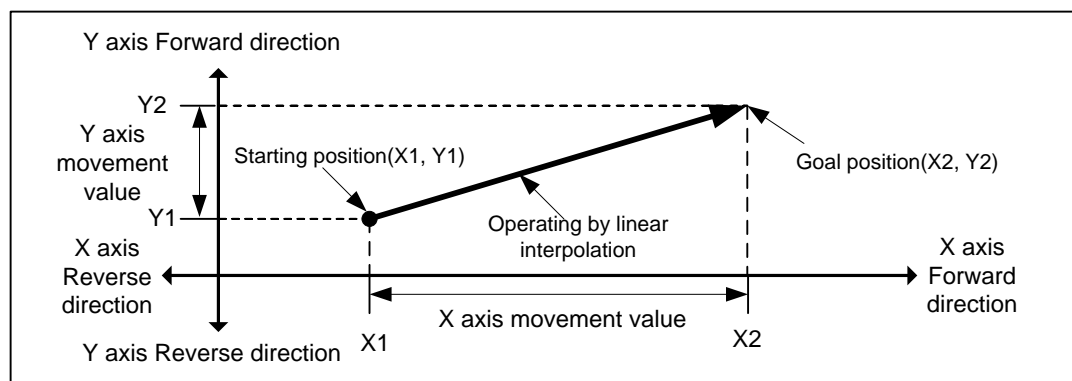
(1) 线性插补控制

在起始位置(当前位置)执行指定轴的线性插补控制.

插补轴组合是有限制的, 最多执行 4 轴的线性插补控制.

(a) 绝对坐标的线性插补

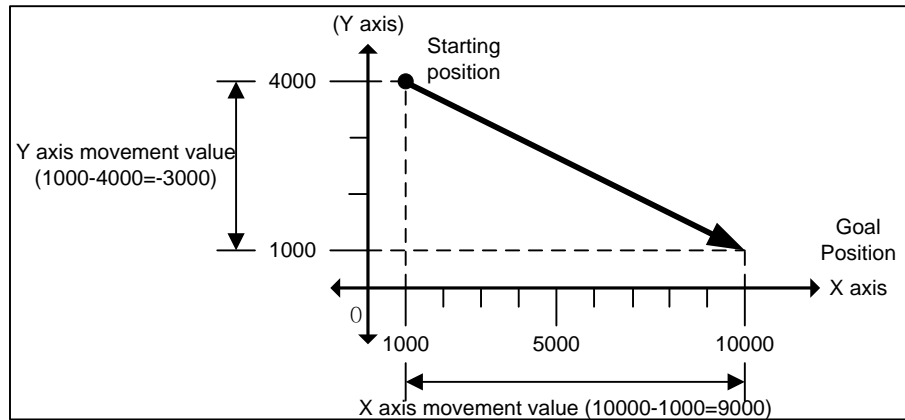
- 1) 通过位置数据执行从起始位置到目标位置的线性插补.
- 2) 位置控制的执行基于原点返回中的指定原点位置.
- 3) 移动方向由每轴的起始位置&目标位置决定.
 - a) 起始位置 < 目标位置 : 正方向位置运行
 - b) 起始位置 > 目标位置 : 反方向位置运行



[示例]

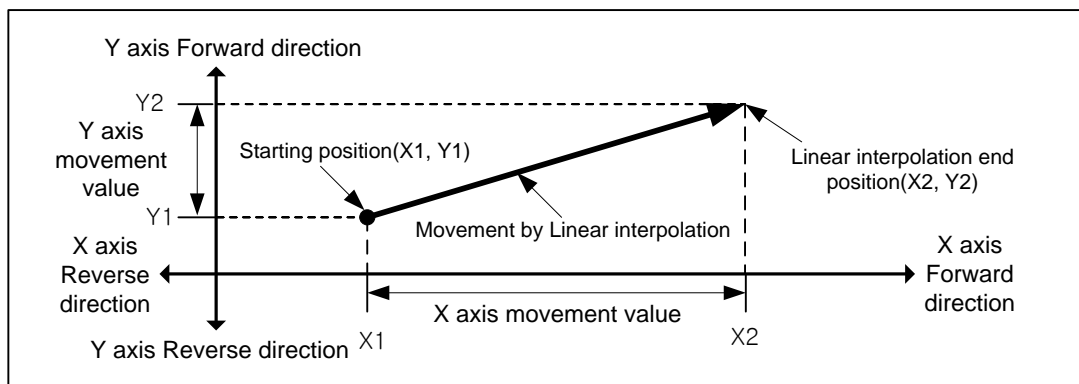
- a) 起始位置 (1000, 4000)
- b) 目标位置 (10000, 1000)

在此条件下, 运行如下.



(b) 相对坐标的线性插补

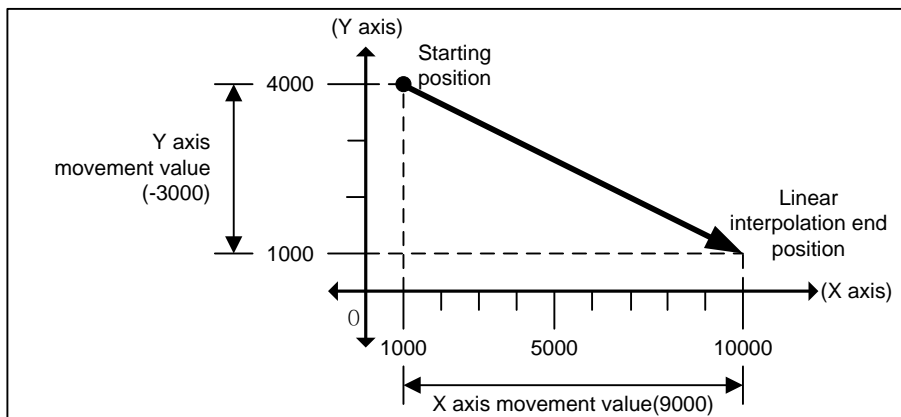
- 1) 目标值变为移动值.
- 2) 移动方向取决于移动值是正数还是负数.
 - a) 正数(+ 或 0): 正方向位置运行
 - b) 负数(-): 反方向位置运行



[示例]

- a) 起始位置 (1000, 4000)
- b) 目标位置(9000, -3000)

在此条件下, 运行如下.



(2) 圆弧插补控制

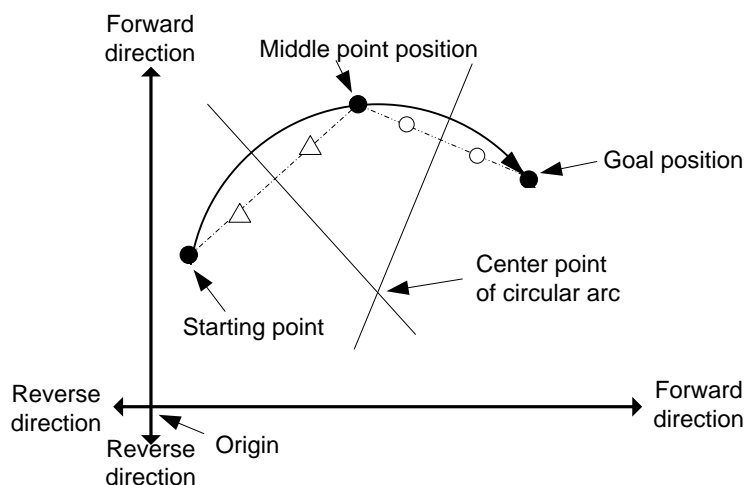
沿 2 轴中每轴指定的正方向圆弧执行插补运行。

圆弧插补根据辅助点有 3 种类型，通过辅助点的中间点方法，使用辅助点作为圆弧中点的中心点方法，使用辅助点作为圆弧半径的半径方法。

用于圆弧插补的 2 轴组合是有限制的。实际轴(1-轴到 32-轴)或者可使用的虚轴(37-轴到 40-轴)

(a) 指定中间点圆弧插补

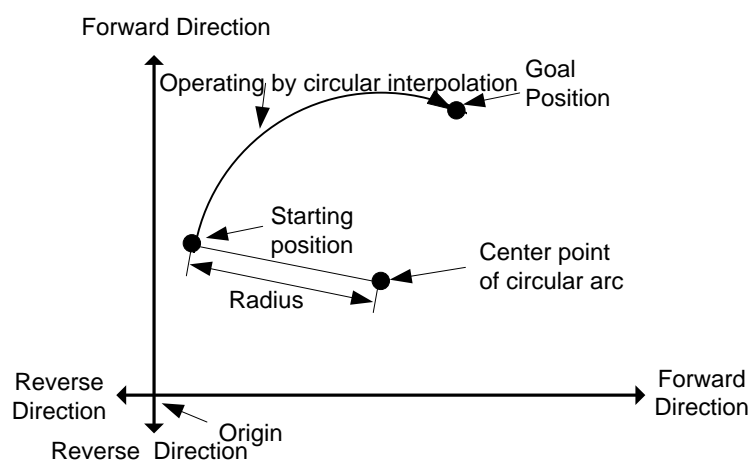
- 1) 在起始位置启动运行并通过指定中间点执行圆弧插补。
- 2) 圆弧的中心点是起始位置和中间点或者中间点和目标位置的垂直平分线的中心点。



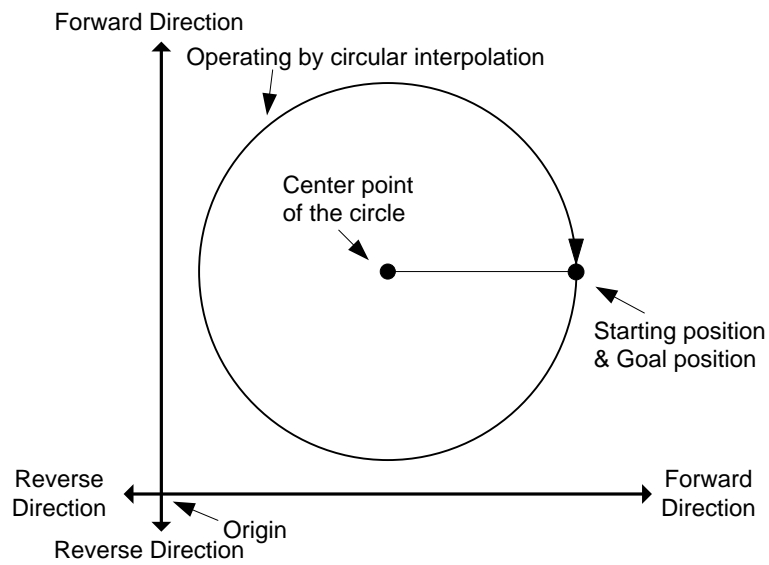
- 3) 移动位置由目标位置和圆弧插补辅助点自动指定。

(b) 圆弧插补的指定中心点

- 1) 从初始位置启动运行，沿着从起始点到作为半径的指定中心点的圆弧方向执行圆弧插补。



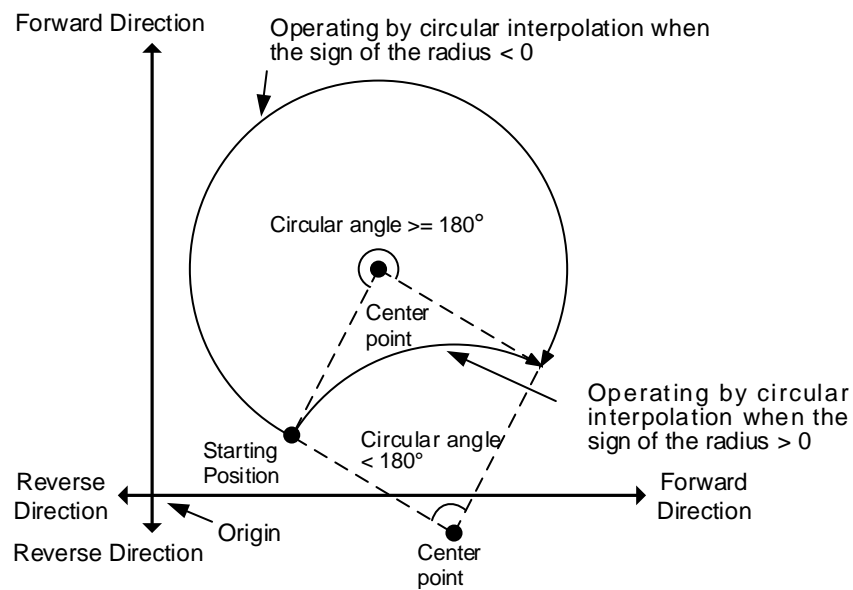
2) 如果目标位置和起始位置相同，可以通过从起始位置到作为半径的辅助点的一段圆弧进行运行。



3) 运行方向由运动功能块中设定的路径(CW, CCW)的选择而决定。

(3) 指定半径圆弧插补

(a) 从起始位置启动运行，沿着作为半径的主轴辅助点指定值的圆弧执行圆弧插补。圆弧的中心点变化取决于绘制圆弧的符号。



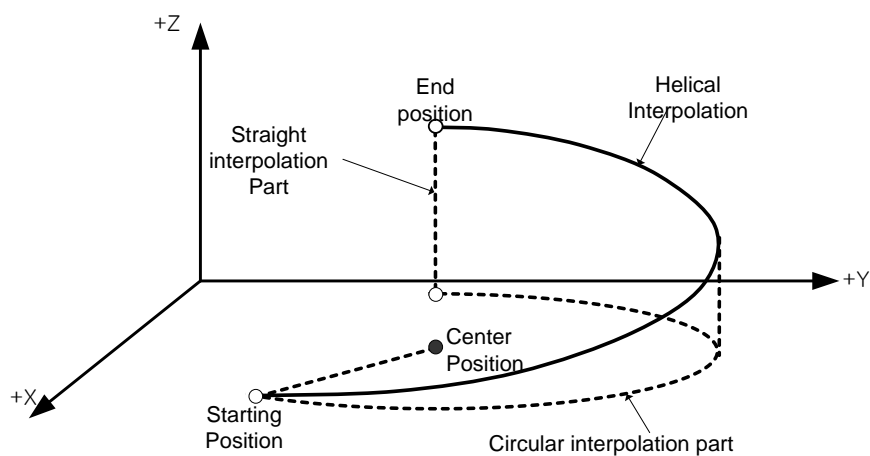
(b) 圆弧指定不可以将目标位置设置为与起始位置相同。

(c) 运行方向和圆弧尺寸由圆弧插补指令的路径(CW, CCW)选择和半径符号所决定。

(4) 螺旋插补

(a) 沿着设置的圆弧指定轨迹进行移动，并同步执行线性插补。

(b) 螺旋插补中使用的轴组合没有限制，实轴(1轴到32轴)或者虚轴(37轴到40轴)的任意三轴。

**1.3.3 速度控制**

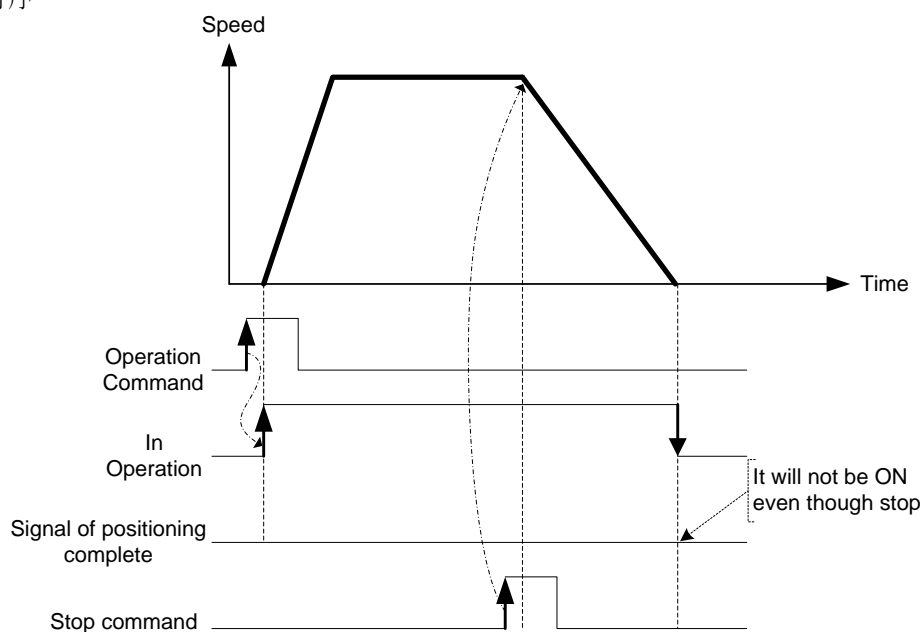
(1) 由速度控制指令执行，按照创建的速度值执行直到执行缓冲指令，或者停止指令。

(2) 速度控制有正向运行和反向运行。

(a) 正向运行: 速率 > 0 且正方向，或者速率 < 0 且反方向

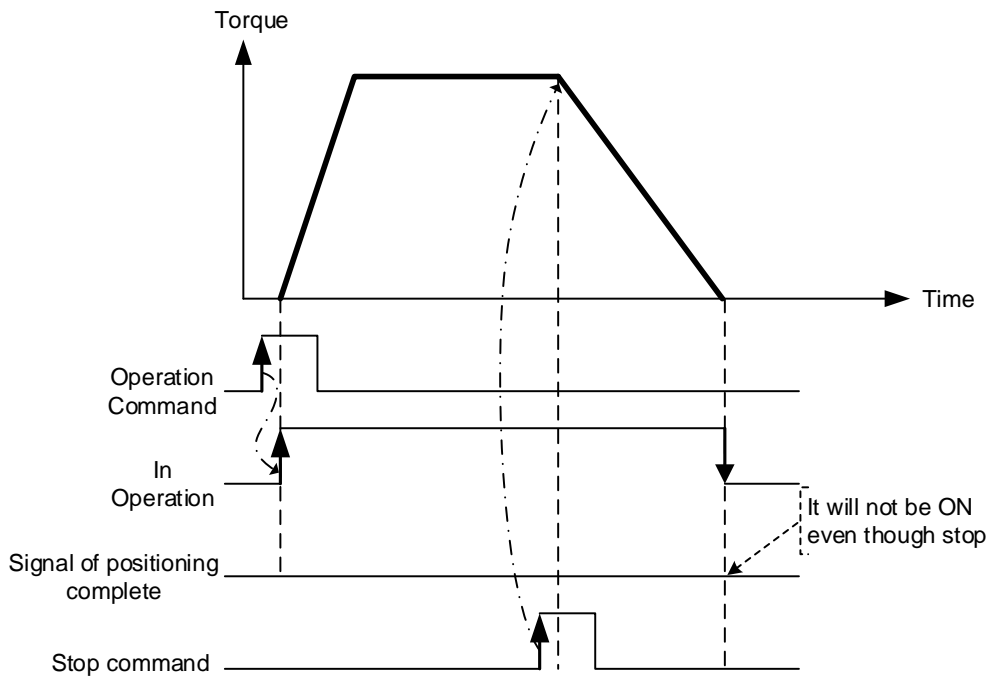
(b) 反向运行: 速率 > 0 且反方向，或者速率 < 0 且正方向。

(3) 运行时序



1.3.4 转矩控制

- (1) 通过转矩控制指令执行，按照设置转矩运行直到输入缓冲指令或者停止指令时停止。
- (2) 转矩控制包括正向运行和反向运行。
 - (a) 正向运行：当方向输入设置为‘1-正方向’
 - (b) 反向运行：当方向输入设置为‘2-反方向’
- (3) 运行时序



第 2 章规格

2.1 一般规格

下表显示MP500系列的一般规格。

编号	项目	规格	相关标准		
1	环境温度	0°C~+55°C	-		
2	存储温度	-25°C~+70°C	-		
3	环境湿度	5 ~ 95%RH (无冷凝)	-		
4	存储湿度	5 ~ 95%RH (无冷凝)	-		
5	抗震动性	断续震动		-	
		频率	加速度	频率	加速度
		5 ≤ f < 8.4 Hz	-	3.5 mm	每个方向 10 次 (X, Y 和 Z)
		8.4 ≤ f ≤ 150 Hz	9.8 m/s ² (1G)	-	
		连续震动		1.75 mm	
		频率	频率		
5 ≤ f < 8.4 Hz	-	1.75 mm			
8.4 ≤ f ≤ 150 Hz	4.9 m/s ² (0.5G)	-			
6	耐冲击性能	<ul style="list-style-type: none"> 最大冲击加速度: 147 m/s²(15G) 施加时间: 11ms 脉冲波形: 正弦半波波型(X,Y,Z 3 方向各 3 次) 	IEC61131-2		
7	抗噪声干扰	方波脉冲噪声	AC: ± 1,500V DC: ± 900V	LSIS 标准	
		静电释放	电压: 4kV (接触放电)	IEC 61131-2, IEC 61000-4-2	
		电磁场辐射噪声	80 ~ 1,000 MHz, 10V/m	IEC 61131-2, IEC 61000-4-3	
		快速瞬变/破裂噪声	类别 电压	电源模块 2kV	数字/模拟输入/输出通讯端口 1kV
8	运行环境	无腐蚀性气体和灰尘	-		
9	使用高度	2000m以下	-		
10	污染等级	2 以下	-		
11	冷却方式	自然空冷	-		

备注

- 1)IEC (国际电工委员会): 是一个促进电气/电子技术等的国际协作和标准化的国际协作团体, 发布国际标准和对以上的所列项目的作用进行运行评估。
- 2)污染程度: 一个指出了对使用环境的污染程度的指标, 以此来测定产品的绝缘性能, 例如, 环境污染 2 意味着无导电传导的污染发生时的状态, 但是是否发生无导电传导污染应当根据露点的形成来判断。

第 2 章规格

2.2 电源规格

下表显示运动控制模块的电源规格.

项目		规格			备注
输入	额定输入电压	AC100V~AC240V			
	输入频率	50/60Hz			
	输入电流	0.7A 及以下			AC100V
		0.4A 及以下			AC240V
	浪涌电流	120A 峰值及以下			AC240V, 单相 90 度
	漏电流	3mA 及以下			
	效率	65% 及以上			
允许瞬时掉电	10ms 及以下				
输出	输出电压	电压	输出电压波动范围	电流	
		+5V	4.90V~5.20V	4A	
		+24V	21.1V~26.9V	0.4A	
	纹波&噪声	电压	纹波	噪声	
		+5V	100mVpp 及以下	200mVpp 及以下	
		+24V	400mVpp 及以下		
	过电流保护	电压	电流		
		+5V	4.4A 及以上		
		+24V	0.44A 及以上		

* 对于电源功能的防护, 推荐使用最大4A保险丝的电源供应.

备注

- 允许瞬时中断时间
在输入电压(AC110/220V)小于最大/最小(AC85/170V)条件下维持正常输出电压(正常运行)的时间.
- 过电流保护
(a) 当电流超出 DC5V,DC24V 回路标准电流时, 过电流保护设备中断回路并停止系统.
(b) 如果过电流发生, 排除电流容量过低, 短路等原因之后, 复位系统.
- 过电压保护
当电流超出 DC5V 回路标准电流时, 过电压保护设备中断回路并停止系统.

2.3 性能规格

下表显示运动控制模块的性能规格。

2.3.1 功能规格

项目		规格	
运行方式		主任务/周期性任务: 固定循环运行, 重复运行 初始任务: 仅在输入 RUN 时运行一次	
控制循环		主任务循环时间: 0.5ms, 1ms, 2ms, 4ms 周期性任务循环时间: 主任务多重设置	
I/O 控制方式		与主任务循环同步(更新方式)	
程序语言		梯形图(功能块), 结构文本, G-代码	
指令数量	运行指令	18	
	基本功能指令	202	
	基本功能块指令	174	
	特殊功能块指令	97	
处理速度	基本	6.25ns 及以上(普通点/线圈)	
	移动	5ns 及以上(字类型)	
	加权	30ns 及以上(字类型)	
程序	数量	最大 256	
	容量	10MB(动作程序), 10MB(NC 程序)	
数据区域	符号变量(A)	4,096KB(保持设置最多可 2,048KB)	
	输入变量(I)	16KB	
	输出变量(Q)	16KB	
	直接变量(M)	2,048KB(保持设置最多可 1,024KB)	
	标记位变量	F	128KB
		K	18KB
U		1KB	
计时器		无限制点, 时间范围: 0.001~ 4,294,967.295sec(1,193 小时)	
计数器		无限制点, 计数器范围: 64 位范围	
程序		初始程序, 主任务程序, 周期性任务程序, NC 程序	
运行模式		RUN, STOP	
重启模式		冷启动, 热启动	
自诊断功能		循环错误监控, 时间共享超出主任务检测, 存储异常, 电源异常, 等.	
备份方法		基本参数或者变量维持区域设置	

第 2 章规格

项目		规格
控制轴数量		32 轴(实轴/虚轴), 4 轴(虚轴), 64 从站(包含实/虚轴)
通讯		EtherCAT (CoE: CANopen 覆盖 EtherCAT, FoE:文件访问覆盖 EtherCAT)
通讯/控制周期		0.5ms, 1ms, 2ms, 4ms (与主任务周期相同)
伺服驱动		支持 EtherCAT CoE 的伺服驱动
控制单元		脉冲, 毫米, 英寸, 度
控制方式		位置, 速度, 转矩(支持伺服驱动), 同步, 插值
位置范围		±长实数, 0
速度范围		±长实数, 0
转矩单位		指定额定转矩%
加速/减速进程		梯形, S-型(在功能块中设置指定加速度)
加速/减速范围		±长实数, 0
手动运行		JOG 运行
CAM 运行		最多 32 CAM 凸轮(32,768 点/32 CAM 凸轮)
绝对坐标系		可用 (当使用绝对编码器类型伺服驱动时)
编码器输入	通道	2 通道
	最大输入	500 kpps
	输入方式	线性驱动输入(RS-422A IEC 规格), 可用集电极开路输出类型编码器
	输入类型	CW/CCW, 脉冲/方向, 相 A/B
数字输入/输出	输入	8 点, 24V, 4mA
	输出	16 点(晶体管输出), 24V, 0.5A/1 点
模拟量输入/输出	输入	2 通道 电压输入范围: -10~10V, 0~10V, 1~5V, 0~5V 电流输入范围: 4~20mA, 0~20mA 分辨率: 14 bit (1/16,000) 精确度: ±0.2%(25°C), ±0.3%(0~55°C) 转换速度: 0.5ms/通道 绝对最大值输入: 电压±15VDC, 电流 ±30mADC
	输出	2 通道 电压输出范围: -10~10V, 0~10V, 1~5V, 0~5V 分辨率: 14 bit (1/16,000) 精度: ±0.2%(25°C), ±0.3%(0~55°C) 转换速度: 0.5ms/通道

项目		规格
远程 I/O		最多 64 个从站
坐标系		笛卡尔, Delta
SD 存储	类型	Micro SD/SDHC
	文件类型	FAT 32
	容量	最大安装 32GB(超出 8GB 存储时仅可使用全部 8GB 区域)
	服务	程序备份/复位, 启动运行, 数据记录
以太网	速度	自动/10Mbps/100Mbps, 1 个端口
	距离	100m
	服务	装载机服务 (MP500) FTP 服务:从其他设备读取/写入一个 SD 存储文件 SNTP 客户端
USB	性能	USB 2.0, 1 个端口
	服务	装载机服务 (MP500)
错误指示		LED 指示
重量		790g

备注

1. 长实数范围: $2.2250738585072^{e-308} \sim 1.79769313486232^{e+308}$
2. 长实数(正数)范围: $0 \sim 1.79769313486232^{e+308}$ (包括 0)

第 2 章规格

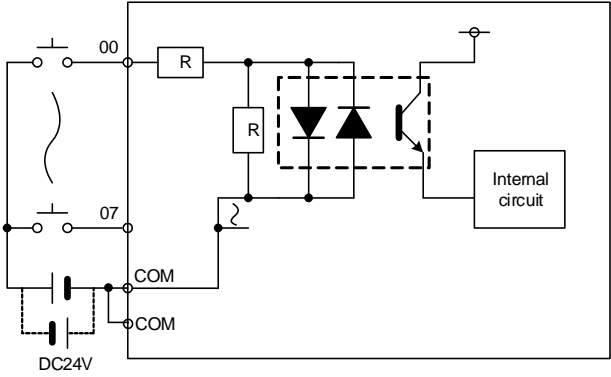

2.3.2 通讯规格

项目	规格
通讯协议	EtherCAT
支持	CoE(CANopen 覆盖 EtherCAT)
物理层	100BASE-TX
通讯速率	100Mbps
拓扑	环形拓扑
通讯电缆	大于 Cat. 5 STP(屏蔽双绞线)电缆
最大从站数量	64(可映射最多 32 个驱动到运动轴)
通讯周期	0.5ms/1ms/2ms/4ms
同步误差	0.5ms/1ms/2ms/4ms
同步通信	通过 CoEPDO(进程数据对象)
非同步通信	通过 CoESDO(服务数据对象)
通信设置	使用 MP500 设置通讯配置

2.3.3 内部输入/输出规格

1. 输入规格(共源/共漏型)

项目	规格
输入点	8 点
隔离方式	光耦隔离
额定输入电压	24V
额定输出电压	About 6mA
使用电压范围	DC20.4V~28.8V(波动范围 5%以内)
On 电压/On 电流	DC19V 及以上 / 3mA 及以上
Off 电压/Off 电流	DC6V 及以下 / 1mA 及以下
输入阻抗	约 4.7 kΩ
响应时间	Off → On On → Off
	初始: 1ms(0.5/1/3/5/10/20/70/100ms: I/O 参数设置)
绝缘电压	AC560Vrms/3 循环(高度 2,000m)
绝缘电阻	绝缘电阻 10 MΩ及以上
共通方式	8 点/COM

回路	No.	点	外形
	00	%IX0.0.0	
	01	%IX0.0.1	
	02	%IX0.0.2	
	03	%IX0.0.3	
	04	%IX0.0.4	
	05	%IX0.0.5	
	06	%IX0.0.6	
	07	%IX0.0.7	
	COM	-	
	COM	-	

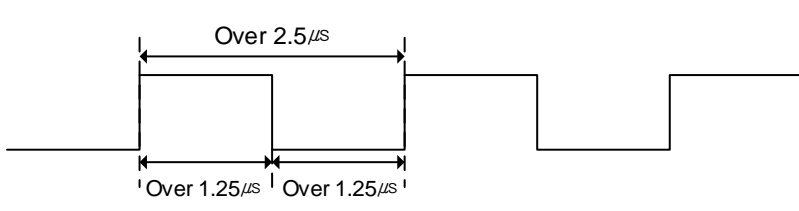
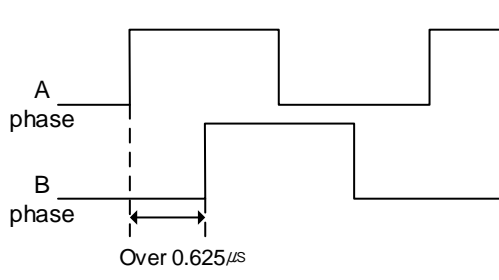
第 2 章规格

2. 输出规格 (共漏型)

项目		规格
输出点		16 点
隔离方式		光耦隔离
额定负载电压		DC 12V / 24V
使用负载电压范围		DC10.2V~26.4V
最大负载电流		0.5A/点, 2A/COM
断态漏电流		0.1mA 及以下
最大浪涌电流		4A / 10ms 及以下
最大压降(On)		DC 0.3V 及以下
浪涌吸收器		齐纳二极管
响应时间	Off→On	1ms 及以下
	On→Off	1ms 及以下(额定负载, 电阻负载)
共通方式		8 点/COM
外部电源	电压	DC12/24V±10% (纹波电压 4Vp-p 及以下)
	电流	10mA 及以下 (DC24V 连接)

回路	No.	点	外形
	00	%QX0.0.0	
	01	%QX0.0.1	
	02	%QX0.0.2	
	03	%QX0.0.3	
	04	%QX0.0.4	
	05	%QX0.0.5	
	06	%QX0.0.6	
	07	%QX0.0.7	
	08	%QX0.0.8	
	09	%QX0.0.9	
	10	%QX0.0.10	
	11	%QX0.0.11	
	12	%QX0.0.12	
	13	%QX0.0.13	
	14	%QX0.0.14	
	15	%QX0.0.15	
	V+	-	
	COM	-	

2.3.4 编码器输入规格

项目	规格	
输入电压	5V (3V ~ 6V)	RS-422A 线性驱动等级一致
输入电流	2 mA~7.5 mA	
最小 On 保证电压	2.5V	
最大 Off 保证电压	1.7V	
输入脉冲	<p>1) 脉冲宽度</p>  <p>2) 相位差</p>  <p>When A phase input pulse is ahead of B phase input pulse : Position value increases</p> <p>When B phase input pulse is ahead of A phase input pulse : Position value decreases</p>	

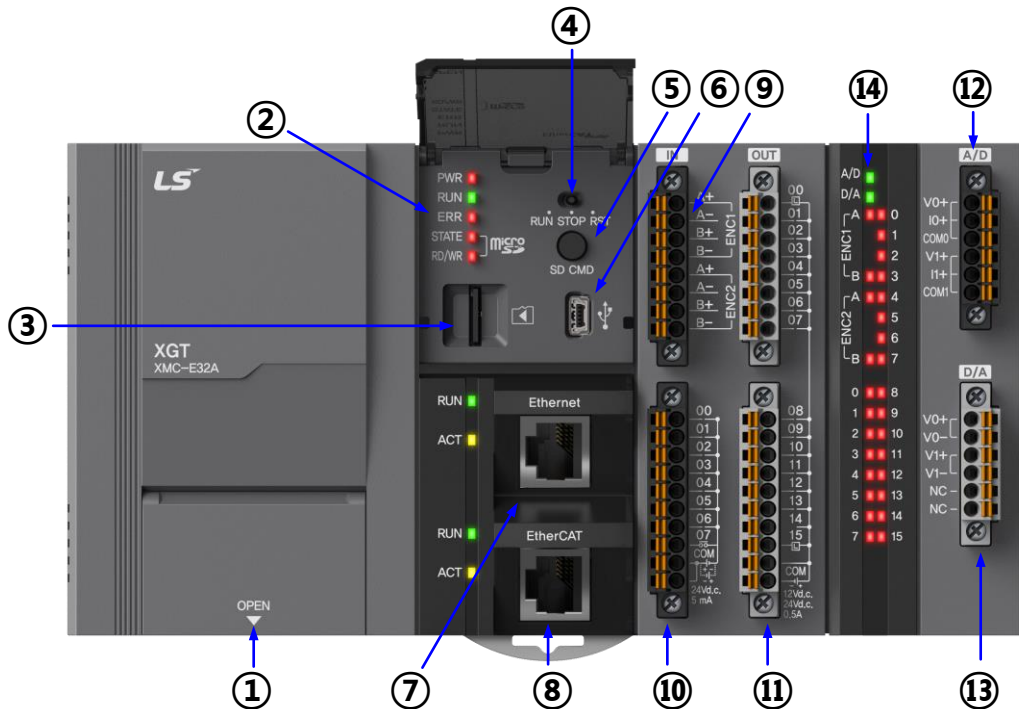
回路	No.	点	外形
<p>备注 1</p>  <p>备注 2</p> 	ENC1 A+	编码器 1 A+ 输入	
	ENC1 A-	编码器 1 A- 输入	
	ENC1 B+	编码器 1 B+输入	
	ENC1 B-	编码器 1 B-输入	
	ENC2A+	编码器 2 A+输入	
	ENC2 A-	编码器 2 A-输入	
	ENC2 B+	编码器 2 B+输入	
	ENC2 B-	编码器 2 B-输入	

备注

备注 1 : 5V 电压输出类型编码器(集电极开路)
 备注 2 : 5V 电压输出类型编码器(线性驱动器)

2.4 各部分名称

2.4.1 各部分名称

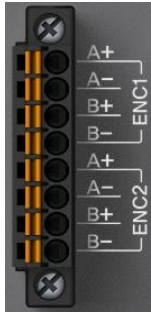
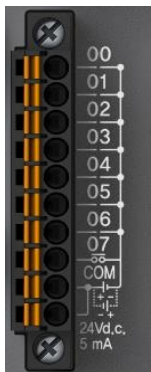



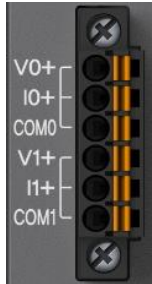
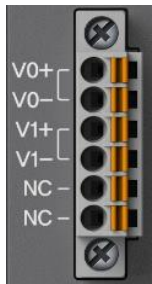

编号	名称	描述
①	电源端子	AC 110/220V 电源输入, LG 端子, DC24V 输出
②	状态显示 LED	显示运动控制模块的运行模式。 • PWR(红灯亮): 电源供应 • RUN(绿灯亮): RUN 模式 • ERR(红灯闪烁): 运行期间发生错误 • STATE(红灯亮/闪烁):当按照 SD 卡时, 红灯亮; 当 SD 卡发生错误时,红灯闪烁。 • RDWR(红灯闪烁): 当读取或者写入 SD 存储时
③	SD 卡连接器	连接 SD 存储卡
④	模式开关	设置运动控制模块的运行模式。 • RUN: 执行运行程序。 • STOP: 停止运行程序。 • RST: 运行程序复位。
⑤	SD 卡命令按钮	点击按钮小于 3 秒。 • 按照脚本设置运行附件功能(备份, 恢复,比较) 点击按钮超过 3 秒。 • SD 电源 On/Off 点击按钮并上电 • 引导运行

编号	名称	描述
⑥	USB 端口	访问 MP500 端口
⑦	以太网端口	以太网通讯端口
⑧	EtherCAT 端口	EtherCAT 通讯端口
⑨	编码器输入连接器	-
⑩	数字输入连接器	-
⑪	数字输出连接器	-
⑫	模拟量输入连接器	-
⑬	模拟量输出连接器	-
⑭	显示输入/输出运行 LED	数字输入/输出, 模拟量输入/输出, 编码器输入

2.4.2 外部设备接口的规格

1. 连接器的针排列

外形	信号名称		信号方向
	ENC1 A+	编码器 1 A+	输入
	ENC1 A-	编码器 1 A-	
	ENC1 B+	编码器 1 B+	
	ENC1 B-	编码器 1 B-	
	ENC2 A+	编码器 2 A+	
	ENC2 A-	编码器 2 A-	
	ENC2 B+	编码器 2 B+	
	ENC2 B-	编码器 2 B-	
	IN0	输入信号 0	输入
	IN1	输入信号 1	
	IN2	输入信号 2	
	IN3	输入信号 3	
	IN4	输入信号 4	
	IN5	输入信号 5	
	IN6	输入信号 6	
	IN7	输入信号 7	输入
COM	输入信号公共端	输入	

外形	信号名称		信号方向
	OUT0	输出信号 0	输出
	OUT1	输出信号 1	
	OUT2	输出信号 2	
	OUT3	输出信号 3	
	OUT4	输出信号 4	
	OUT5	输出信号 5	
	OUT6	输出信号 6	
	OUT7	输出信号 7	
	OUT8	输出信号 8	
	OUT9	输出信号 9	
	OUT10	输出信号 10	
	OUT11	输出信号 11	
	OUT12	输出信号 12	
	OUT13	输出信号 13	
	OUT14	输出信号 14	
	OUT15	输出信号 15	
		24V	
GND		DC24V GND	
V0+		模拟电压输入 0	
I0+		模拟电流输入 0	
COM0		模拟输入 0 公共端	
V1+		模拟电压输入 1	
	I1+	模拟电流输入 1	输入
	COM1	模拟输入 1 公共端	
	V0+	模拟电压输出 0+	
	V0-	模拟电压输出 0-	
	V1+	模拟电流输出 0+	
	V1-	模拟电流输出 0-	
	NC	未连接	输出
	NC	未连接	

2. 编码器内部电路

项目	针编号	信号	
*备注1 	ENC1A+	ENC1A+	编码器1A+ 输入
	ENC1A-	ENC1A-	编码器1 A-输入
	ENC1B+	ENC1B+	编码器1 B+ 输入
	ENC1B-	ENC1B-	编码器1 B-输入
*备注2 	ENC2A+	ENC2A+	编码器2 A+输入
	ENC2A-	ENC2A-	编码器2 A-输入
	ENC2B+	ENC2B+	编码器2 B+输入
	ENC2B-	ENC2B-	编码器2 B-输入

备注

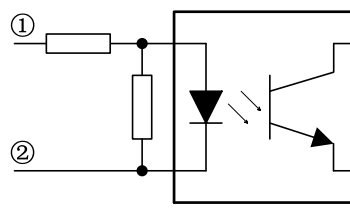
* 备注1

编码器1的配线是5V电压输出类型(集电极开路)的示例。当使用12V, 24V类型MPG时, 变更输入电压从5V到12V或者24V, 如果是12V, 在连接电源前(增加所需上拉电阻), 需连接910Ω电阻到ENC1 A+(针1), ENC1 B+(针3), 如果是24V, 则为2.4kΩ电阻。

* 备注2

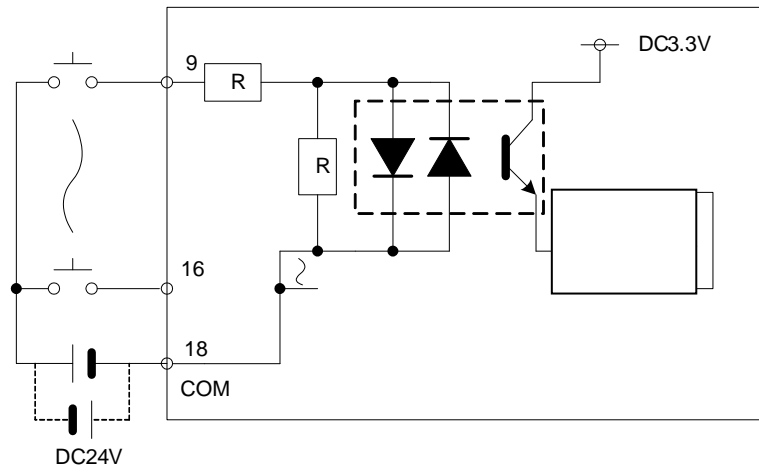
编码器2的配线是5V电压输出类型(线性驱动)的示例

以下描述为连接编码器时模块的内部电路。

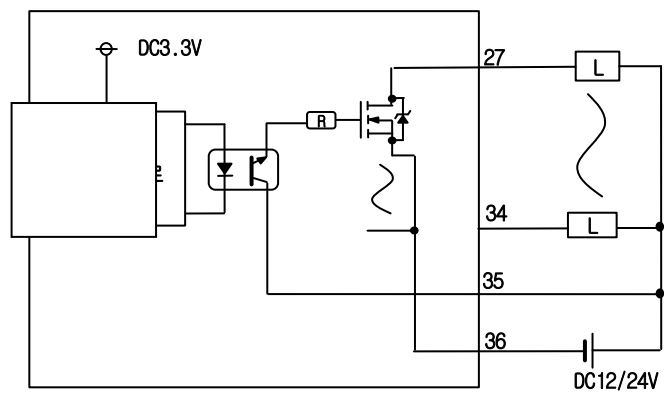
项目	内部电路	编号	端子	信号名称
输入		①	A+	A相脉冲输入 +
		②	A-	A 相脉冲输入 -
		①	B+	B 相脉冲输入+
		②	B-	B 相脉冲输入 -

第 2 章规格

3. 输入内部回路



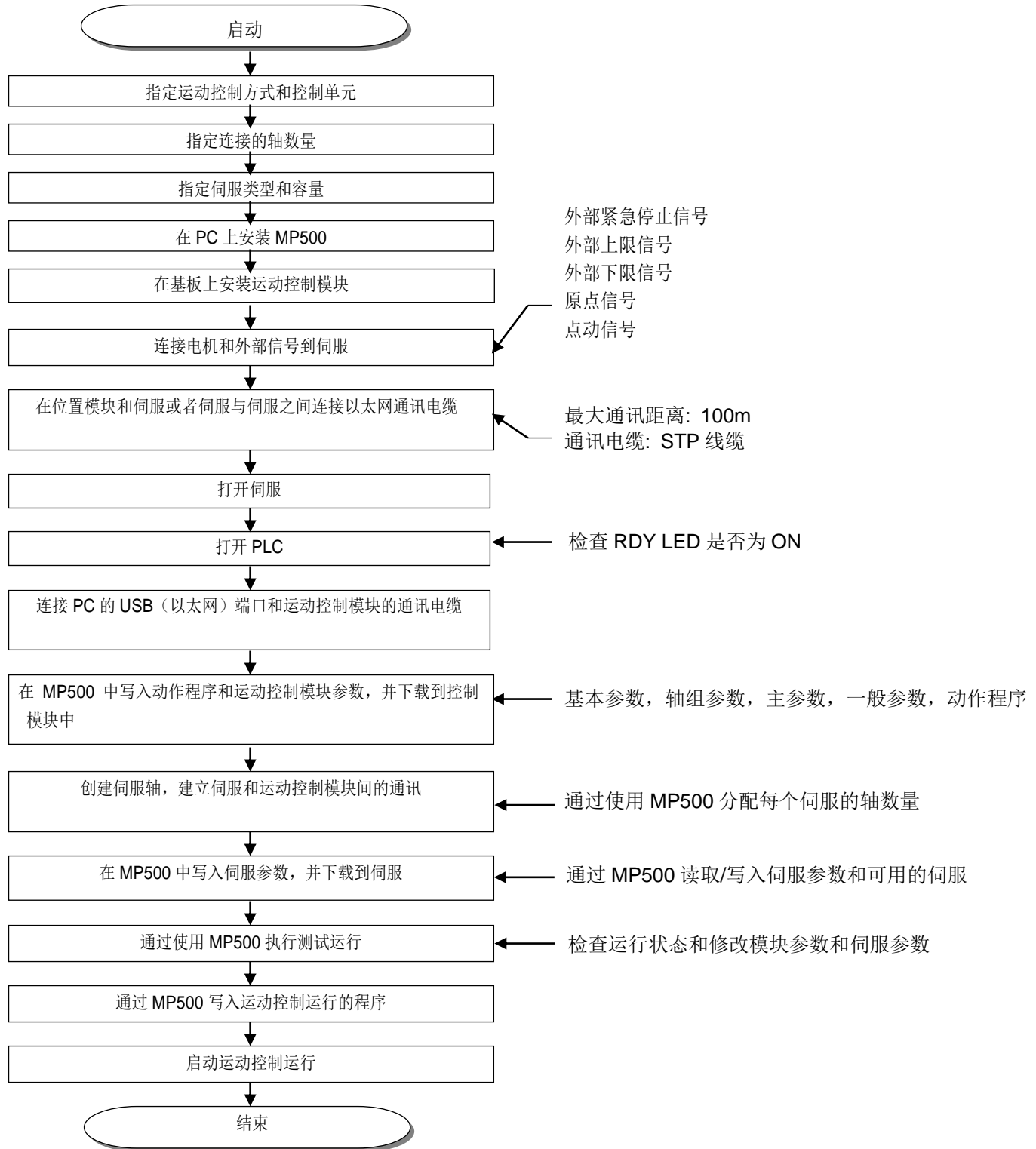
4. 输出内部回路



第 3 章 操作顺序与安装

3.1 操作顺序

以下描述运动控制模块的操作顺序。



3.2 安装

3.2.1 安全措施




- ▶ 为了系统安全运行，请在控制模块外部设计保护回路，因为当任何外部电源错误或者控制模块故障可能导致事故发生。
 - (1) 应该安装在控制模块的外侧以便运行紧急停止回路，保护回路和例如正向/反向运行的反向动作互锁回路，互锁回路防止例如位置上/下限的机器损坏。
 - (2) 如果控制模块检测到以下错误，所有运行停止并关断所有输出。
 - (根据参数设置允许维持输出)
 - (a) 当超出保护设备电流或者超出保护动作电压时
 - (b) 当例如在控制模块中发生 WDT 错误的自诊断功能错误时
- ▶ 当控制模块未检测到关于 IO 控制部分错误时,关断所有输出。

在控制模块外部设置故障保护回路以保证设备安全运行. 参考 4.1.1 故障保护回路.

 - (1) 由于输出设备，继电器，TR 等错误可能导致异常. 关于输出信号可能导致严重事故，在外部设计监控回路。
- ▶ 当负载电流超出额定或者由于连续加载短路时，可能发生过热或者火灾，所以在外部设计保护回路，例如保险丝。
- ▶ 在控制模块电源供应完成之后设计外部电源供应。如果外部电源供应首先完成，可能由于输出错误,运行错误导致事故。
- ▶ 关于通讯错误发生，对于每个站的运行状态，参考每个通讯手册。
- ▶ 当外部设备连接到控制模块时对于控制模块的控制，需配置用于系统安全运行的互锁回路.运行期间，对于执行程序变更，运行状态变更，参考手册并检查安全状态。特别是，对于控制远距离控制模块，由于通讯错误等造成的控制模块错误，用户无法立即响应。

限定如何在控制模块和外部设备间数据通讯错误的纠正措施，在控制模块程序中增加互锁回路。

 危险

- ▶ 请勿将控制线和通讯线靠近主回路或者电源线. 间距保持在 100mm 以上. 否则可能由于噪声出现故障.
- ▶ 对于控制灯负载, 加热器, 电磁阀等. 从 Off -> On, 可能出现大电流(10 倍与正常电流), 所以考虑变更模块间额定电流的余量.
- ▶ 根据控制模块主电源和处理控制模块电源 On-Off 的外部电源的延迟和启动时间的不同造成无法正常运行.
例如, 提供外部电源完成后启动控制模块主电源, 当控制模块运行时 DC 输出模块可能故障, 所以配置一个优先导通控制模块主电源的回路, 或者外部电源错误或者控制模块错误, 可能导致故障.
- ▶ 请勿在整个系统中出现以上错误, 机器或者事故的部分故障应配置在外部控制模块中.

3.2.2 安装环境

本产品具有非常高的可靠性，以应对各种安装环境，但是需要注意的是，在以下几种情况下，系统的可靠性和安全性可能受到影响。

1. 环境条件

- (1) 安装的控制面板适用于防水，抗震的环境。
- (2) 在远离连续冲击或者震动的地方。
- (3) 在远离直接暴露在强光下的地方。
- (4) 在随着温度的快速变化而不产生结露的地方。
- (5) 在环境温度保持在0-55°C的地方。

2. 安装构造

- (1) 如果加工螺丝孔或者接线，请注意不要使残余的线进入控制模块内部。
- (2) 请安装在一个便于操作的地方。
- (3) 不要在相同的面板上安装高电压机器。
- (4) 从导管到周围模块的距离不小于50mm。
- (5) 接地的地方，其周围的噪音环境必须非常好。

3.2.3 操作注意事项

下面指出了在对位置模块从打开到安装的注意事项。

- (1) 不要使产品发生跌落或者遭到剧烈的冲击。
- (2) 不要将PCB移开。这将可能导致故障的产生。
- (3) 在接线过程中，注意不要将剩余的线头或者异物放在模块的上部。如果有什么异物进入模块，应迅速清除。

3.2.4 安装/拆卸运动控制模块

! 备注

- ▶ 运动控制模块在固定安装前必须正确安装固定卡扣.强行拆装可能导致损坏. 如果模块未正确安装, 可能导致故障.
- ▶ 请勿跌落或者撞击模块外壳,端子连接器.
- ▶ 请勿将 PCB 从壳体中取出.

操作注意事项

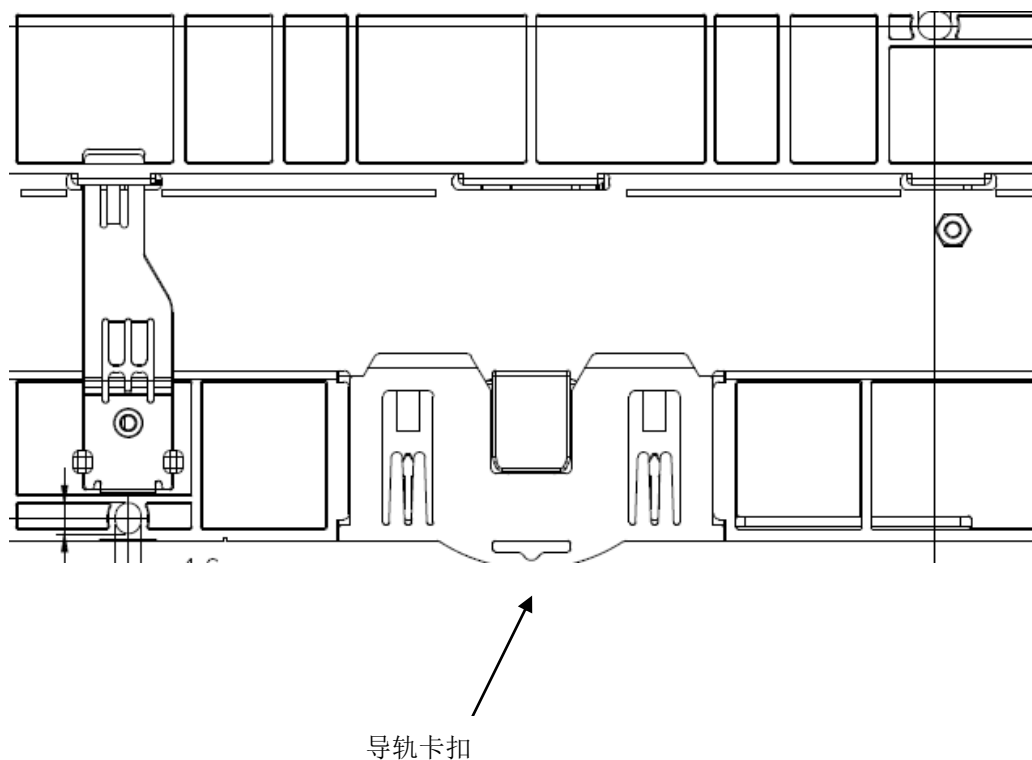
1. 安装手册中指定规格范围内使用运动控制模块.
2. 如果使用超出范围, 可能导致触电, 火灾, 故障, 产品损坏.

(1) 运动控制模块的安装

运动控制模块有一个用于固定在导轨上的导轨(轨道宽度: 35mm)固定卡扣.

(a) 导轨安装

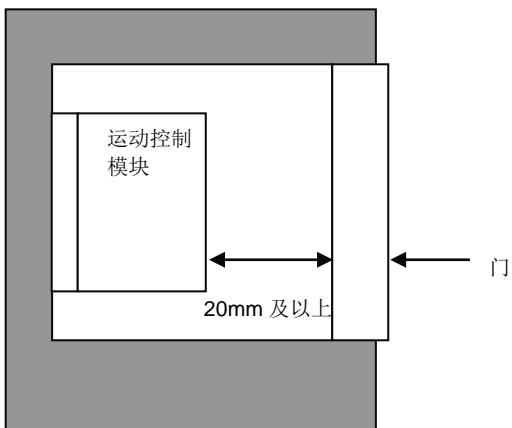
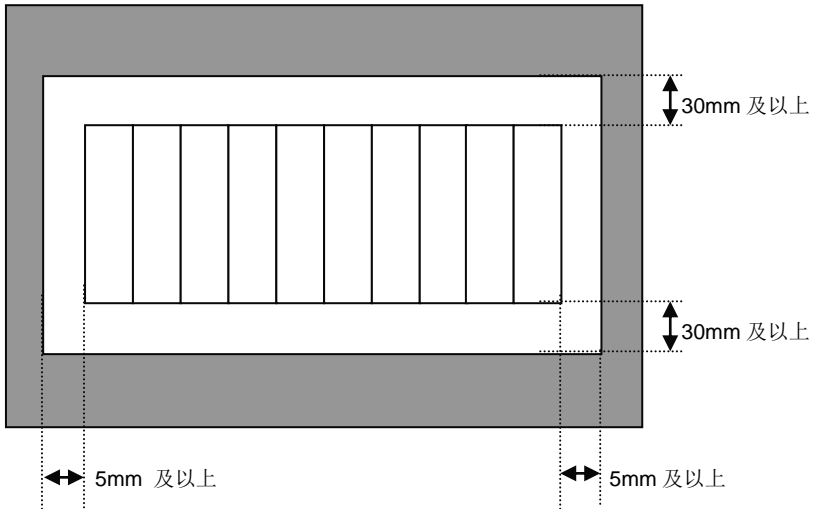
- 如下运动控制模块底部的导轨卡槽并安装在导轨上
- 推动模块的导轨固定卡扣以固定模块



第 3 章 操作顺序与安装

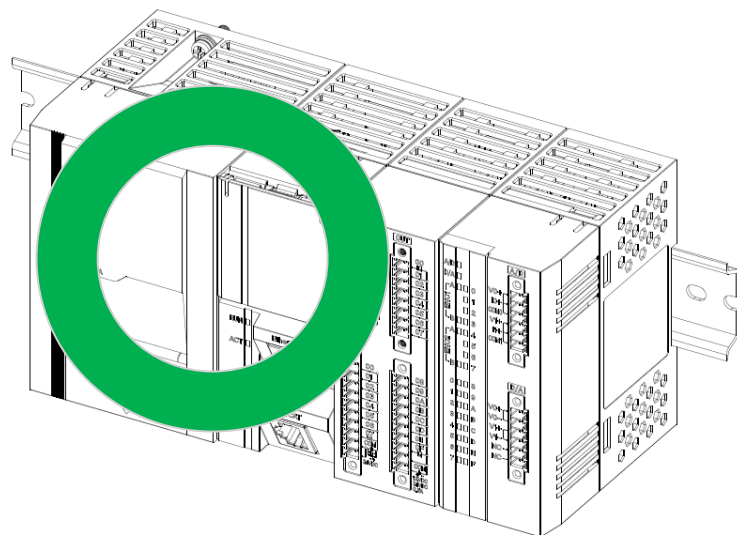
(2) 控制模块安装位置

如下保持模块与散热机构或者部分的距离, 便于拆装.

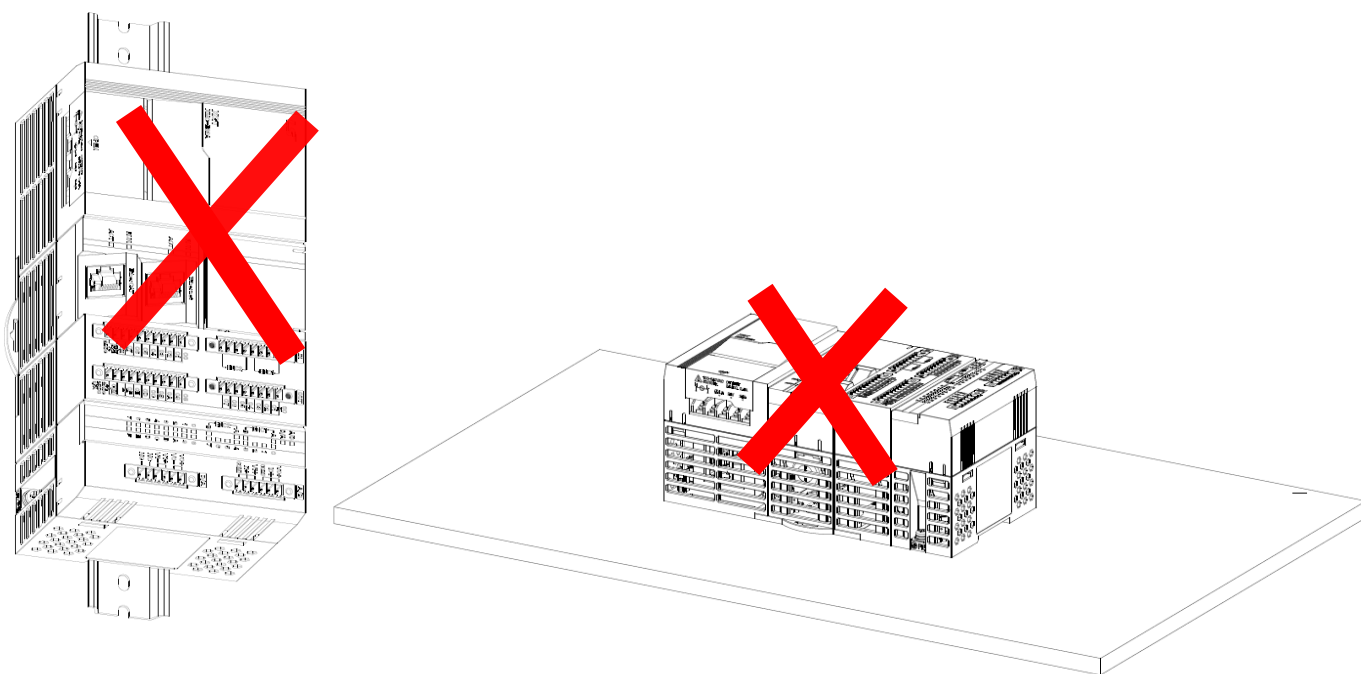


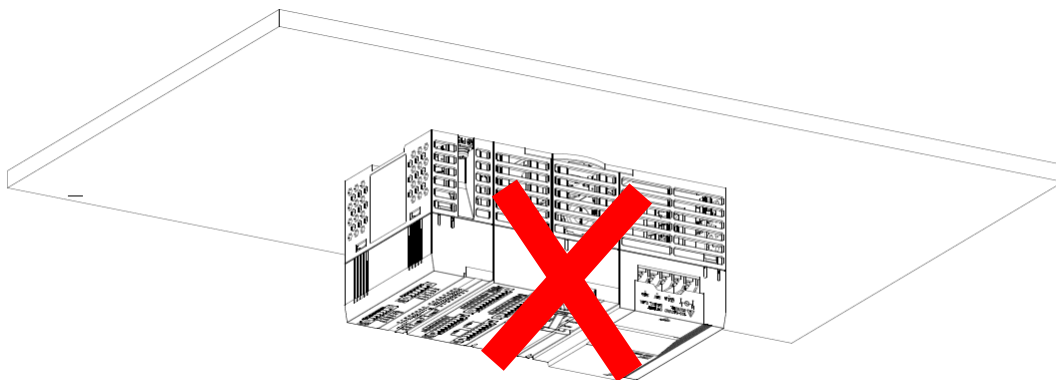
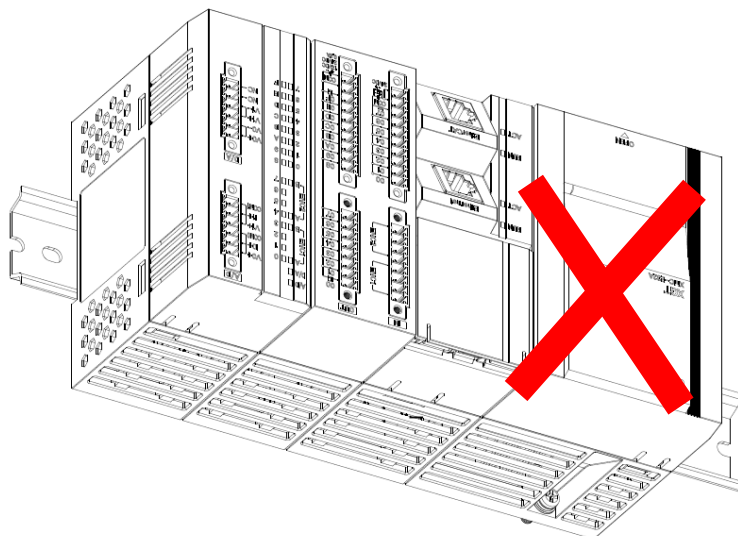
(3) 控制模块安装方向

(a) 为了便于散热，请如下图安装。



(b) 请勿如下安装。



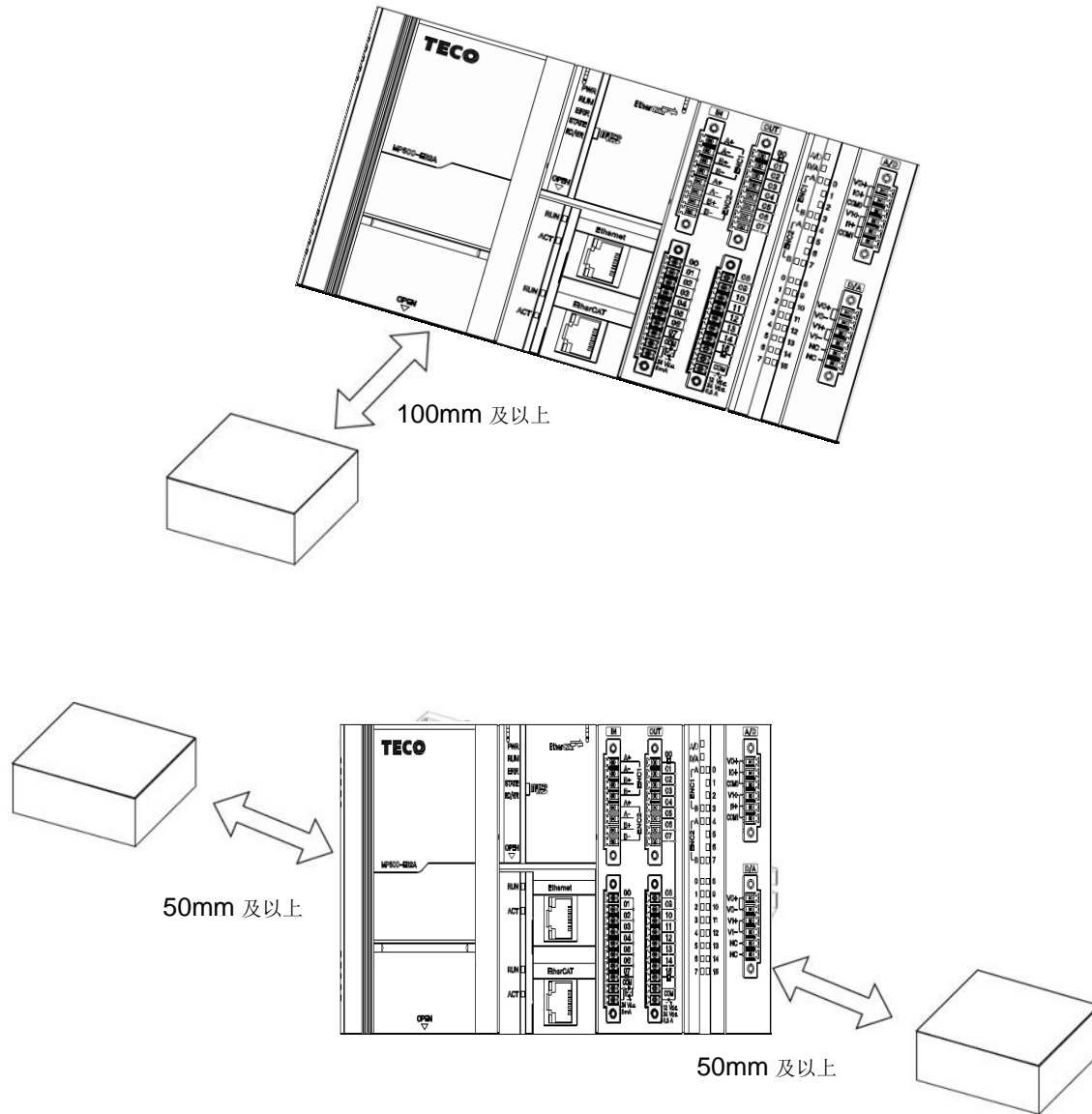


(4) 与其他设备间距离

为了避免辐射噪声或者高温, 如下图保持模块与设备(连接器和继电器)间距离。

设备安装在运动控制模块前部: 100mm 及以上

设备安装在运动控制模块侧面: 50mm 及以上



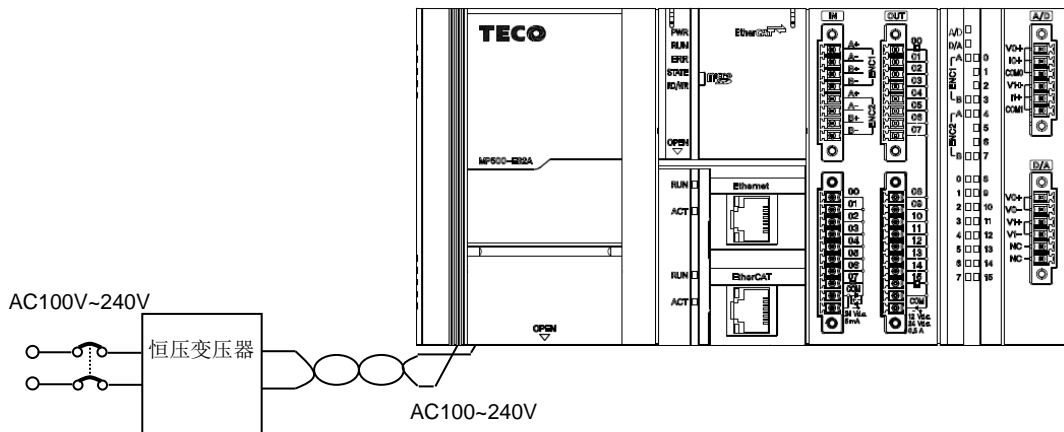
3.3 配线注意事项

3.3.1 配线注意事项

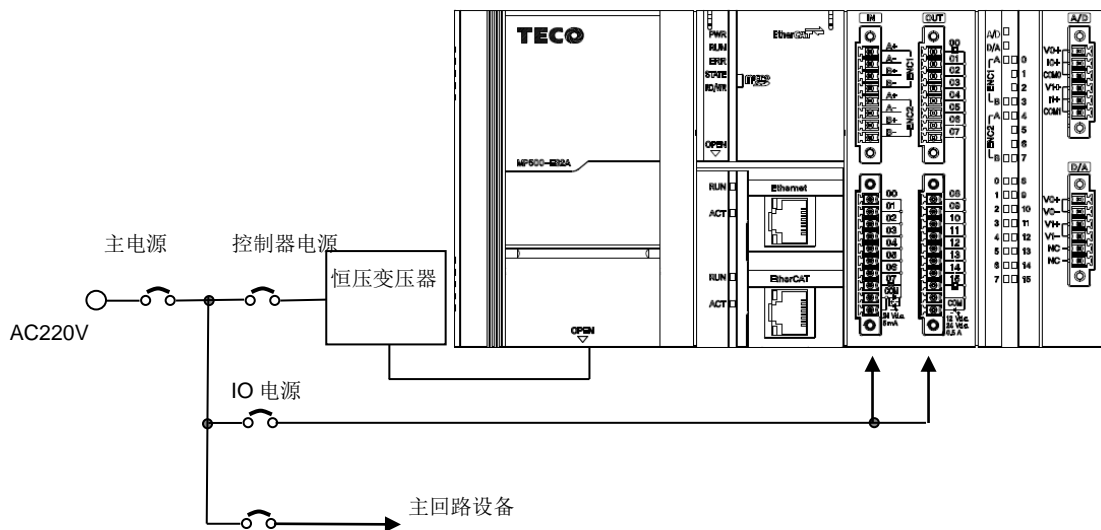
- (1) 在位置模块和驱动器之间的连接电缆的长度应该尽可能的短一点。(最长: 2m和10m).
- (2) 对于位置模块的交叉电流和外部 I/O 信号, 要求使用单独的电缆, 为了避免交叉电流所产生的浪涌或者感应噪声.
- (3) 在选择线的时候应该考虑环境的温度、允许电流和推荐使用大于最大尺寸AWG22(0.3mm²).
- (4) 接线时, 如果太接近高温机器、材料或者长时间直接接触到油, 可能发生短路, 导致错误和故障的产生.
- (5) 在运用外部连接信号到接线端之前, 务必检查极性.
- (6) 如果将高压电缆和电源电缆连接在一起时, 可能导致故障发生, 从而产生错误或者失败.
- (7) 如果通过管道进行接线, 要求有接地的管道.
- (8) 对于位置模块和驱动间的通讯电缆, 使用STP CAT-5或以上等级.
- (9) 当运动控制模块运行期间发生一个通讯错误(0x0F50, 0x0F51, 0x1F00, 0x1011, 0x2011等), 需在连接运动控制模块到EtherCAT从设备的通讯电缆附件铁氧体磁芯, 并运行模块, 因为有可能由运动控制模块和EtherCAT从设备的配线噪声干扰导致.
- (10) 当使用用于编码器信号和外I/O信号的配线连接器时, 将设备安装在没有灰尘或者腐蚀性气体的地方.

3.3.2 电源配线

- (1) 对于电压调整大于指定规格的情况，连接恒压变压器。

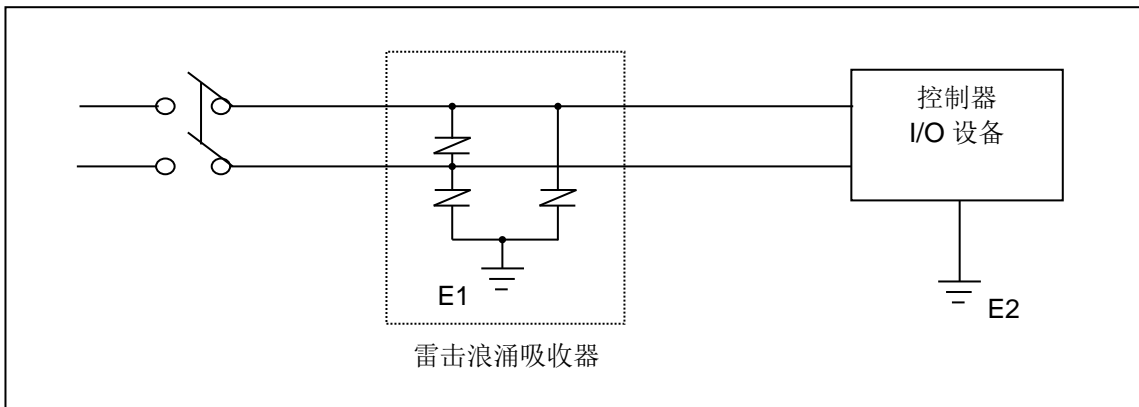


- (2) 在线和地之间连接包含微小噪声的噪声。
(当噪声过大时，连接绝缘变压器.)
- (3) 隔离控制器电源, I/O 设备和电源设备如下。



- (4) 如果使用控制模块 DC24V，不要并联几个电源模块的 DC24V。
- (5) AC 电源线应该紧密缠绕并且短距离连接。
- (6) AC 电源线应尽可能粗一点(2mm²)以减少压降。
- (7) AC 电源线不可以安装靠近主回路线 (高电压/高电流)和 I/O 信号线. 间距保持 100mm
- (8) 当发生噪声渗透时使用绝缘屏蔽变压器或者噪声过滤器。
- (9) 每个输入电源的配线应尽可能短的缠绕，并且屏蔽变压器或者噪声过滤器配线不可以使用导管配置。

(11) 为了防止雷击的浪涌，如下使用雷击浪涌吸收器。

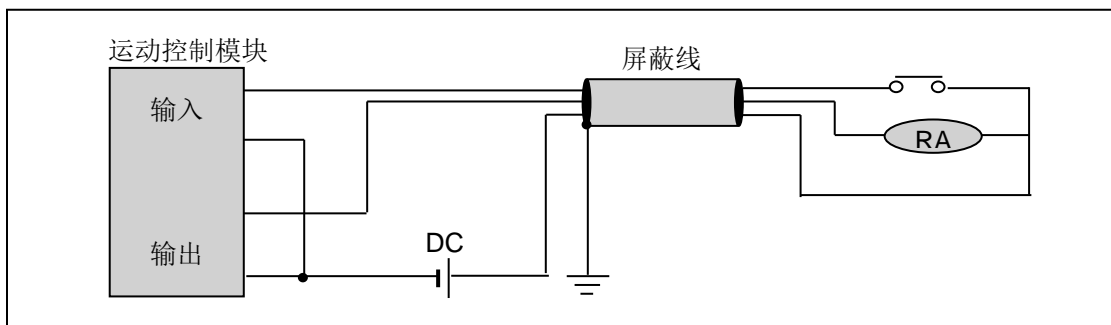


备注

隔离控制模块接地(E2)和雷击浪涌吸收器的接地(E1).
选择一个雷击浪涌吸收器类型，最大电压不是吸收器指定允许电压。

3.3.3 I/O 设备配线

- (1) I/O 设备电缆尺寸限制为 $0.3\sim 2\text{ mm}^2$ ，但是建议使用推荐尺寸(0.3 mm^2)。
- (2) 请将输出信号线和输入信号线隔离。
- (3) I/O 信号线应与高电压/高电流主回路电缆保持 100mm 及以上距离。
- (4) 应使用批次屏蔽线，并且运动控制模块侧应接地，除非主回路电缆和电源线没有隔离。

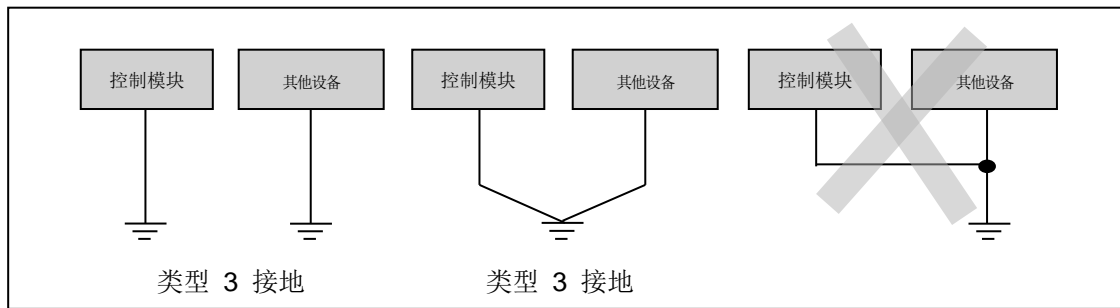


- (5) 当应用导管线时，确保导管完整接地。

3.3.4 接地(LG)线

- (1) 该控制模块有两种接地系统，例如 LG 和 PE。
- (2) LG (⏏)是电源滤波器接地方式，作为噪声对策使用。该控制模块执行噪声对策，但是如果没有任何特殊原因推荐使用 LG 接地。对于 LG 位置，请参考 2.4.1 各部分名称。
- (3) PE (⏏)是预防触电的接地方式。为防止事故该方式与地部分连接。当安装在导轨上时，其与导轨接触。当安装在面板上时，通过螺丝与安装的面板连接。对于 PE 位置，参考 3.2.3 运动控制模块的安装与移除。
- (4) 请参考以下用于 LG 接地的结构。

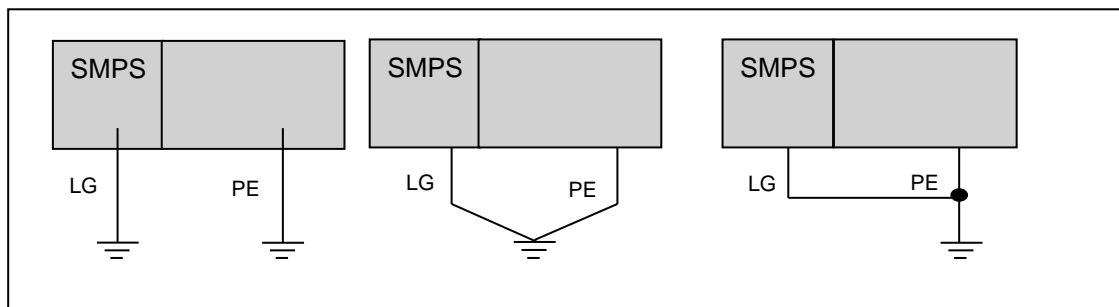
- (a) 对于接地，请确保使用单独接地。对于接地结构，应用类型 3 接地(接地电阻小于 100 Ω)
- (b) 如果不可以单独接地，使用如下 B)的公共接地方式。



A) 单独接地：最好 B) 公共接地：好 C) 公共接地: 不可用

- (c) 使用大于 2 mm²的接地电缆。为了缩短接地电缆的长度，请将接地点靠近控制模块。

- (d) 如果检测到任何接地故障，分离 PE 和 LG。



A) 单独接地：最好 B) 公共接地：好 C)公共接地:不可用

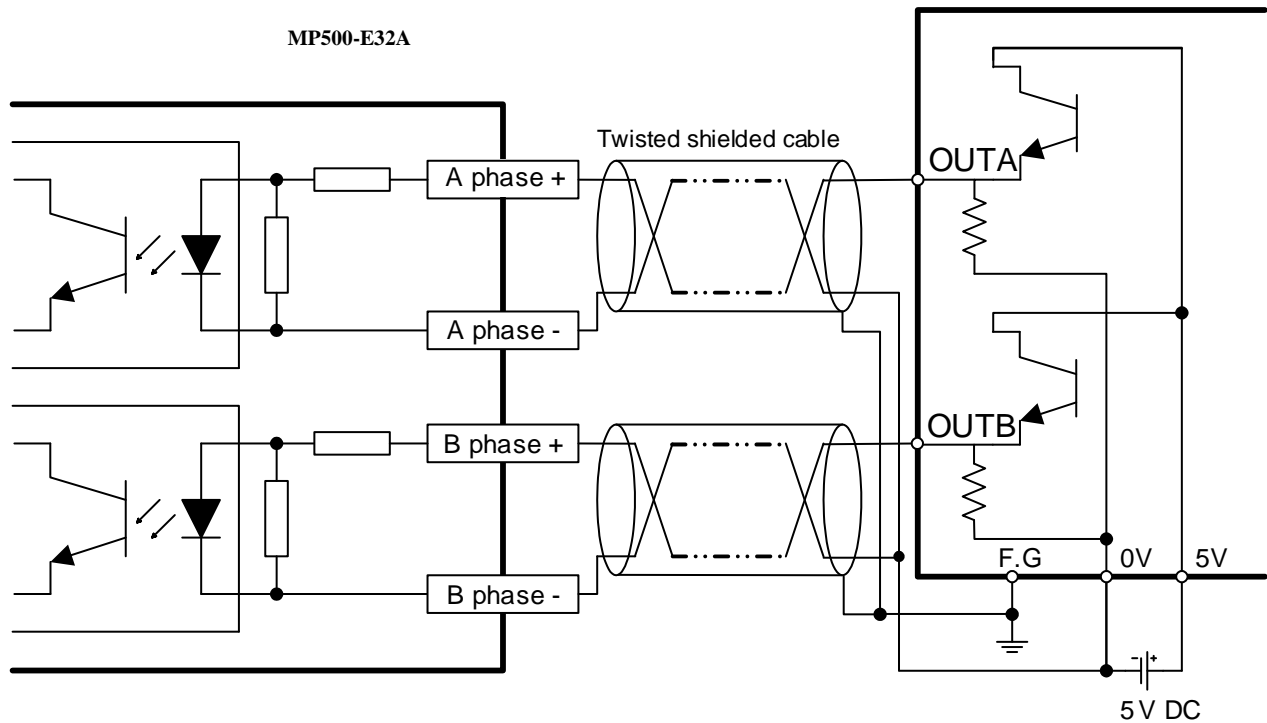
3.3.5 接线电缆规格

用于接线的电缆规格如下.

外接类型	线缆规格 (mm ²)	
	下限	上限
数字输入	0.18 (AWG24)	1.5 (AWG16)
数字输出	0.18 (AWG24)	1.5 (AWG16)
模拟 I/O	0.18 (AWG24)	1.5 (AWG16)
通讯	0.18 (AWG24)	1.5 (AWG16)
主电源	1.5 (AWG16)	2.5 (AWG12)
接地(LG)	1.5 (AWG16)	2.5 (AWG12)

3.3.6 编码器输入(DC 5V 电压输出)配线实例

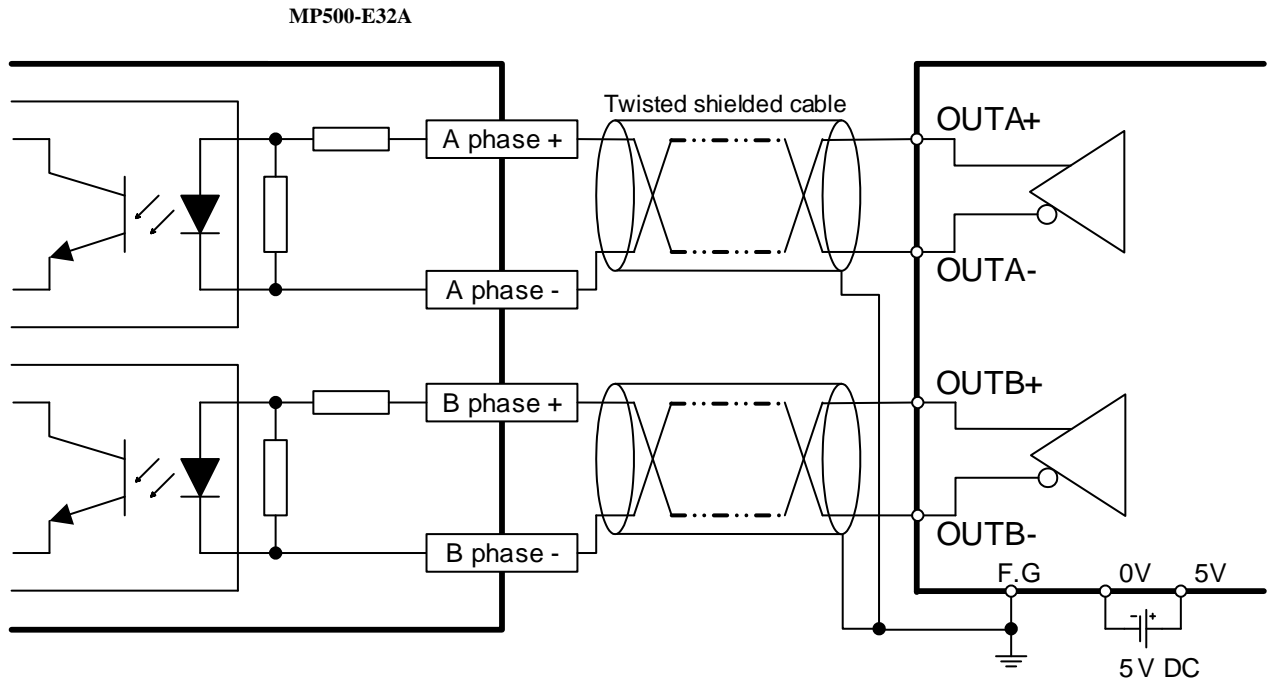
当脉冲发生器是电压输出类型时，位置模块和编码器输入部分的配线实例如下。
如果脉冲发生器是图腾柱输出，并用于电压输出类型，配线相同。



备注

配线前，请考虑脉冲发生器的最大输出距离。

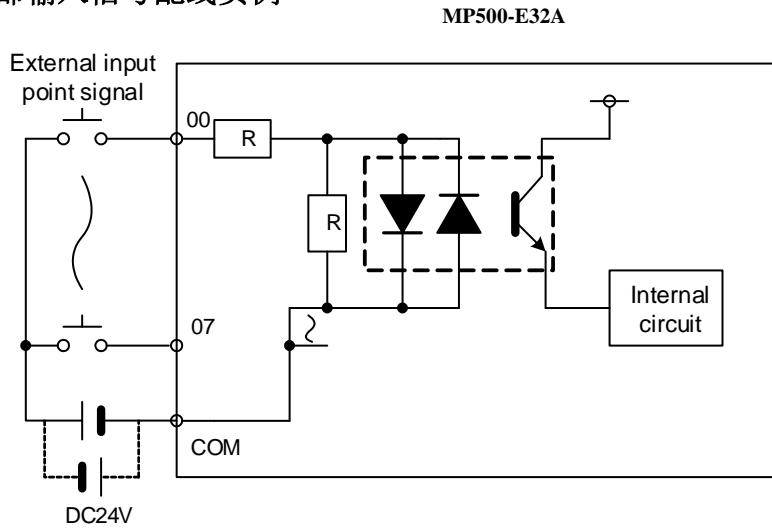
3.3.7 编码器输入(5V 线性驱动输出)配线实例



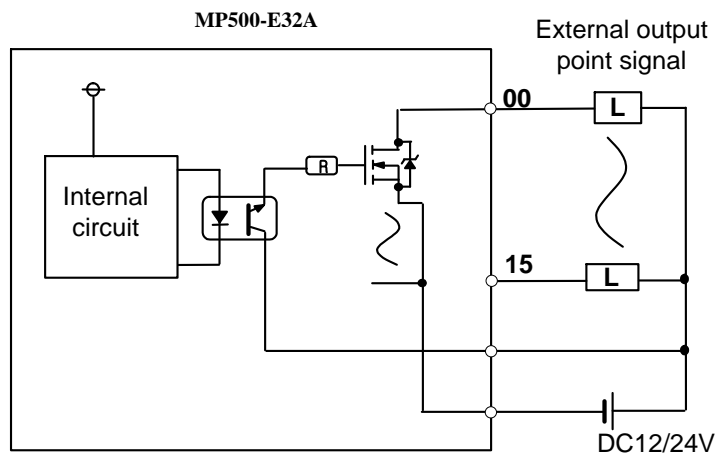
备注

配线前，请考虑脉冲发生器的最大输出距离。

3.3.8 外部输入信号配线实例



3.3.9 外部输出信号配线实例



3.4 EMC

3.4.1 EMC 标准

(1) EMC 指令一致性要求

EMC 指令指定产品必须“构建不会产生过度电磁干扰(辐射)并且不会受电磁干扰影响(免疫)的结构”。应用产品必须符合这些要求。

本节概述运动控制模块机械组装 EMC 指令一致性的注意事项。这些注意事项建立在适用标准控制和要求之上。然而, TECO 不保证所有机械制造细节符合符合下面的指令。EMC 指令一致性的方式和判断由机器制造商最终判定是否符合 EMC 指令。

(2) EMC 标准

EMC 指令适用标准如下表。

规格	测试项目	测试详情	标准值
EN50081-2	EN55011 辐射噪声 * 2	测量产品的电磁辐射	30~230 MHz QP : 50 dB μ V/m * 1 230~1000 MHz QP : 57 dB μ V/m
	EN55011 传导噪声	测量从产品到电源线的电磁辐射	150~500 kHz QP : 79 dB Mean: 66 dB 500~230 MHz QP : 73 dB Mean: 60 dB
EN61131-2	EN61000-4-2 静电免疫	应用于设备外壳的静电免疫测试	15 kV 空气放电 8 kV 接触放电
	EN61000-4-4 快速瞬态爆裂噪声	应用于电源线和信号线的脉冲噪声免疫测试	电源线: 2 kV 数字 I/O : 1 kV 模拟 I/O, 信号线: 1 kV
	EN61000-4-3 辐射场 AM 调制	产品辐射场的免疫测试	10V μ m, 26~1000 MHz 80%AM 调制 @ 1 kHz
	EN61000-4-12 阻尼振荡波免疫	叠加在电源线的阻尼振荡波免疫测试	电源线: 1 kV 数字 I/O (24V 及以上): 1 kV

* 1) QP: 准峰值, 含义: 平均值

* 2) 运动控制模块是一个开放式设备(安装在其他设备上的设备), 必须安装在一个导电控制面板。当运动控制模块安装在控制面板内侧时执行相应的测试项目。

(3) 控制面板

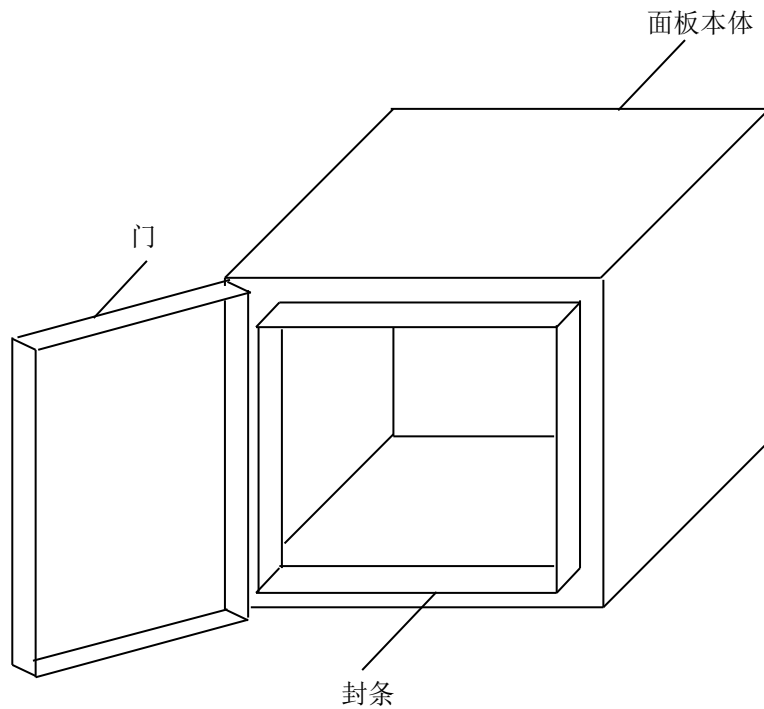
运动控制模块是一个开放式设备(安装在其他设备上的设备)，必须安装在一个控制面板上。需要防止接触运动控制模块造成的触电并降低运动控制模块产生的噪声。将运动控制模块安装在一个金属面板上以降低运动控制模块产生的 EMI (电磁干扰)，控制面板规格如下：

1) 控制面板

运动控制模块控制面板必须符合以下特性：

- (a) 控制面板使用 SPCC(冷轧低碳钢)。
- (b) 金属面板厚度大于 1.6mm。
- (c) 使用隔离变压器防止外部浪涌电压的电源供应。
- (d) 控制面板必须具备没有无线电泄露的结构。

例如，创建一个盒机构的门，确保门和面板本体彼此重叠。该结构降低运动控制模块产生的浪涌电压。



- (e) 为了确保与控制面板或者基座的良好电气接触，表面涂层和焊接以便面板和金属板之间的良好接触。

第 3 章 操作顺序与安装

2) 电源和接地线连接

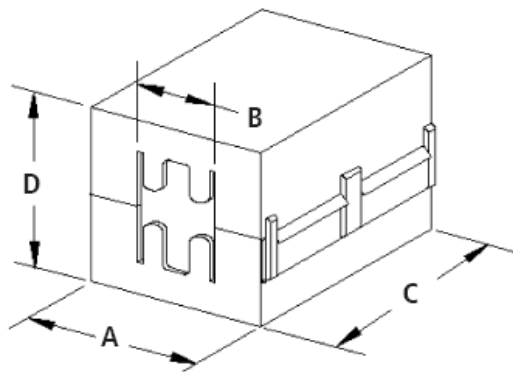
运动控制系统的接线盒电源线必须如下连接.



- (a) 控制面板的接地线需要使用粗线，以确保即使在高频情况下的接地低阻抗.
- (b) LG (线路接地)和 FG (壳体接地)端子的功能是将运动控制系统产生的噪声传导到大地，所以要确保阻抗尽可能小.
- (c) 接地线本身可能产生噪声，所以要选择短且粗的线，防止其作为一个天线.
- (d) 在电源线上附加铁氧体磁芯以满足 CE 规格.

[铁氧体磁芯]

制造商	名称	尺寸 (mm)				最大电缆直径(mm)	网址
		A	B	C	D		
Laird	28A3851-0A2	30.00	13.00	33.70	30.00	12.85	www.lairdtech.com
Laird	28A5776-0A2	29.20	20.00	42.00	42.00	19.40	www.lairdtech.com
Coilmaster	C2L RU130B	31.50	13.00	33.00	31.50	13.00	www.coilmaster.com.tw
TDK	ZCAT3035-1330	30.00	13.00	34.00	30.00	13.00	www.tdk.com



(4) 要求符合低压指令

低压指令要求每个设备电源供应范围在 50V~1000VAC 和 75V~1500VDC，以符合安全要求。本节描述满足低压指令要求的运动控制模块系列的配线和安装注意事项。

手册描述内容基于标准控制应用和要求。然而，LSIS 不保证所有机械制造细节符合符合下面的指令。EMC 指令一致性的方式和判断由机器制造商最终判定是否符合 EMC 指令。

1) 运动控制模块的标准应用

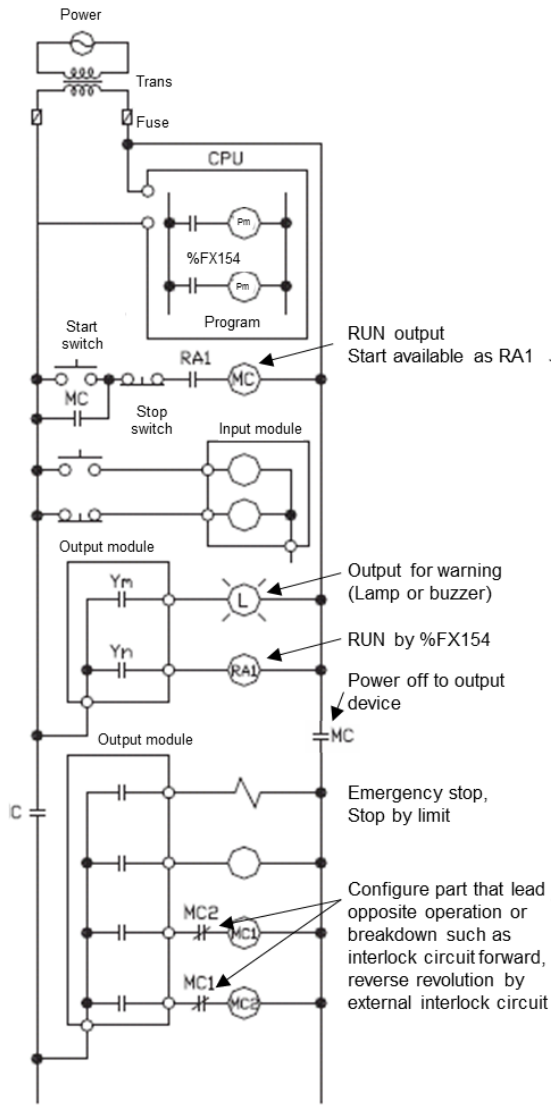
运动控制模块符合 EN6100-1 (用于测量室,控制室或者实验室的设备安全性)。在 AC50V/DC75V 或以上运行的运动控制模块同样符合以上标准。

3.5 故障保护

3.5.1 故障保护回路

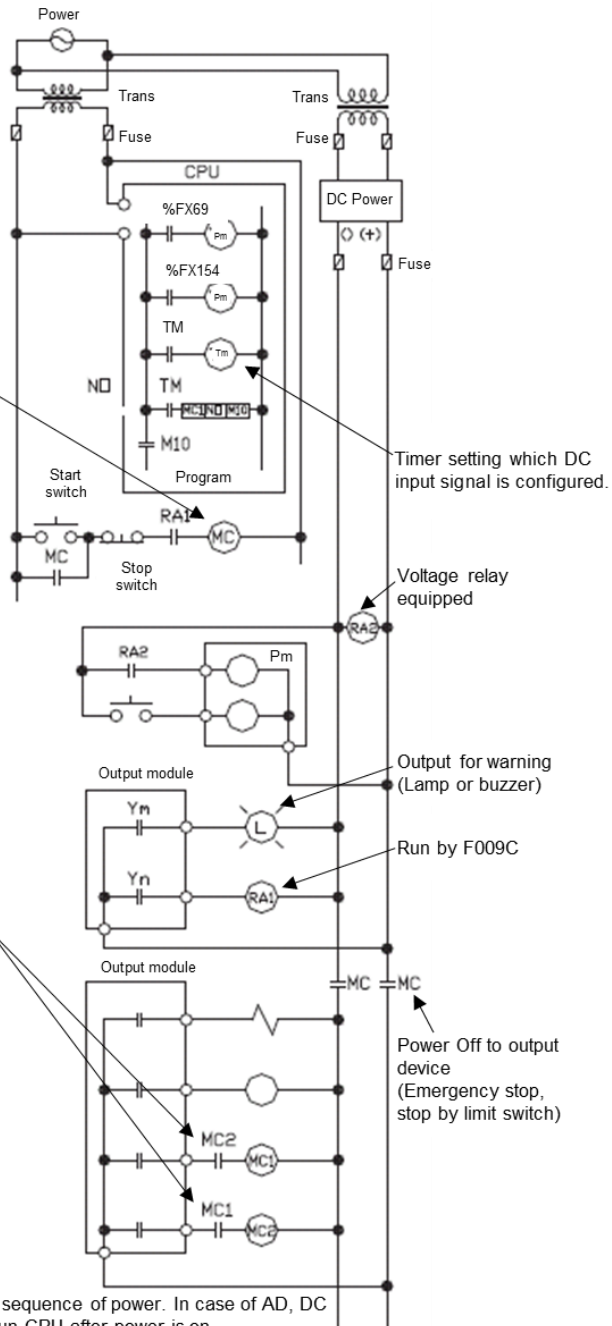
(1) 系统设计示例

AC 的情况



- Start sequence of power. In case of AC
 (1) Turn on power
 (2) Run CPU.
 (3) Turn on start switch
 (4) Output device runs by program through magnetic contactor (MC) [On]

AC, DC 的情况



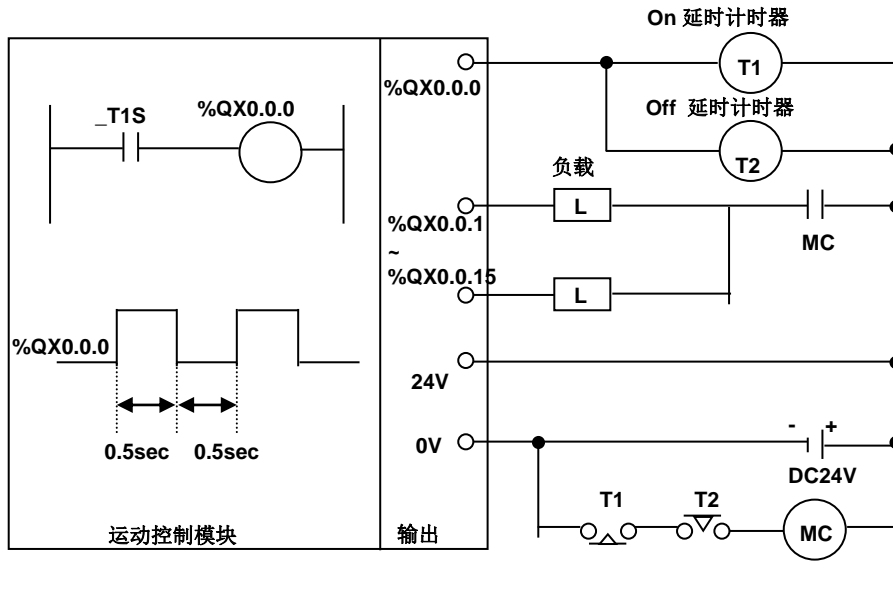
- Start sequence of power. In case of AD, DC
 (1) Run CPU after power is on
 (2) Turn on RA2 as DC power on
 (3) Turn on timer after DC power is stable.
 (4) Turn on start switch
 (5) Output device runs by program through magnetic contactor (MC) [On]

(2) 运动控制模块故障的故障保护措施

运动控制模块和存储的故障通过自诊断进行检测，但是比如 I/O 控制部分等故障问题无法检测。这种情况下，可能根据故障状态的不同而不同，所有触点可能为 On 或者 Off，所以无法保证控制对象正常运行。

我们尽最大努力保证品质，但是对于 PLC 可能发生的一些问题，请在外部配置故障保护回路，防止某些原因引起的设备损坏或者故障。以下示例为系统配置故障保护回路示例。

[故障保护回路示例]



第 3 章 操作顺序与安装

3.6 维护

请确保进行日常和周期性维护和检测，以确保运动控制模块维持在最好的状态。

3.6.1 维护和检测

I/O 模块主要由半导体设备组成，其寿命为半永久。然而，要求对于周围环境可能导致故障的设备进行定期检测。当每 6 个月进行 1 到 2 次检测时，检测项目参考如下。

检查项目		判定	纠正措施
输入电压变化率		在输入电压范围内	调整在允许范围内。
输入/输出电源供应		每个模块的输入/输出规格	维持在每个模块的允许范围内。
周围环境	温度	0 ~ + 55°C	根据定义范围调整运行稳定和湿度。
	湿度	5 ~ 95%RH	
	振动	无振动	使用耐振动橡胶或者防振动方法。
模块移动		不允许移动	紧固卡扣。
端子螺丝连接条件		不允许松动	紧固端子螺丝
备件		查看备件数量及存储条件	补充缺失部分并提高条件。

3.6.2 日常检测

下表显示日常检测的规格和项目。

检查项目		内容	判定	纠正措施
基板连接条件		查看螺丝。	螺丝无松动。	紧固螺丝。
端子台或者扩展电缆的连接条件		查看螺丝松动情况。	螺丝无松动。	紧固螺丝。
		查看无焊接端子间距离。	预留适当间隙。	纠正。
		扩展电缆连接。	连接器不可以松动。	纠正。
LED 显示	PWR LED	查看 LED 为 On。	On (Off 表示错误)	
	Run LED	运行期间查看 LED 为 On。	On (闪烁或者 Off 表示错误)	
	ERR LED	运行期间查看 LED 为 Off。	闪烁表示错误	
	输入 LED	查看 LED 的 On 和 Off。	当输入运行时为 On,当输入不运行时为 off。	
	输出 LED	查看 LED 的 On 和 Off	当输出运行时为 On,当输出不运行时为 off。	

3.6.3 定期检查

每 5 个月查看以下项目 1~2 次, 需要执行纠正措施.

查看项目		查看方式	判定	纠正措施
周围环境	环境温度	- 温度和湿度计测量 - 测量腐蚀性气体	0 ~ 55 °C	调整到一般标准(控制章节的内部环境标准)
	环境湿度		5 ~ 95%RH	
	环境污染等级		没有腐蚀性气体	
控制器条件	无松动, 进入	移动模块的单元	模块确保安装牢固	重新紧固螺丝或者挂钩
	灰尘或者外部杂质	目视检查	没有灰尘或者外部杂质	水平放置产品以防止灰尘进入通风孔, 使用干布擦去灰尘或者外部杂质. 确保外部杂质不要进入通风孔.
连接条件	松动端子螺丝	再次拧紧螺丝	螺丝不可以松动	重新紧固
	端子间距离	目视检查	适当空隙	纠正
	连接器松动	目视检查	连接不可以松动.	重新紧固连接装配螺丝
线性电压查看		测量输入端子间电压	3.3 电源规格	变更供电电源

3.7 故障排除

以下描述系统运行过程中可能发生的错误内容，诊断和纠正措施。

3.7.1 故障排除基本步骤

故障可靠性不仅在于可靠的设备，同时在于故障发生的短暂停机时间。快速运行系统需要快速发现问题及应对措施。以下显示故障排除的基本指令。

(1) 目视检查

查看以下内容。

- 设备运行条件(在停止和运行状态)
- 电源 On/Off
- I/O 设备状态
- 配线条件(I/O 配线, 扩展和通讯线)
- 多个显示状态(例如 POWER LED, RUN LED, ERR LED 和 I/O LED)

查看完成后，连接外部设备并查看运动控制器运行状态和程序内容。

(2) 故障查看

如下在错误条件下查看任何变化。

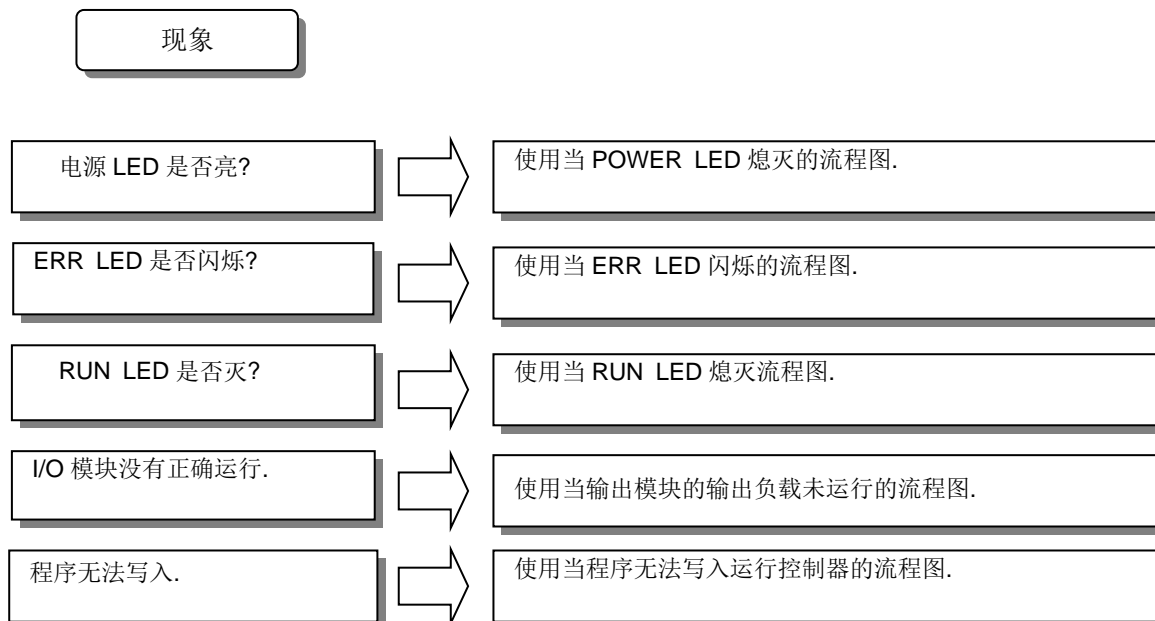
- 切换到 STOP 位置，然后切换电源。

(3) 在故障位置缩小故障可能发生的范围，例如：

- 运动控制器内部或者外部？
- I/O 模块或者其他模块？
- 运动程序？

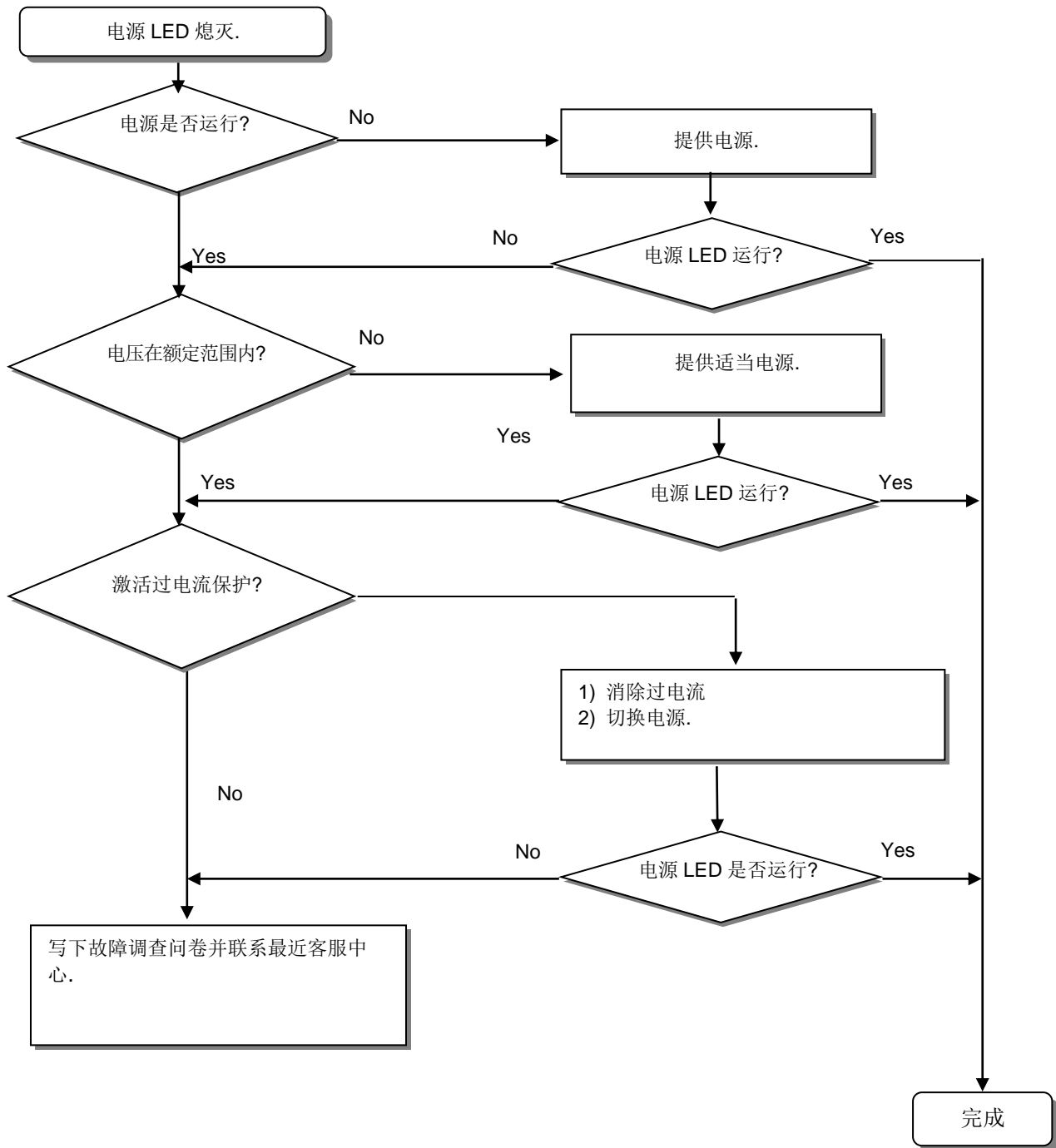
3.7.2 故障排除

本节描述根据错误确定故障原因和纠正措施的步骤。



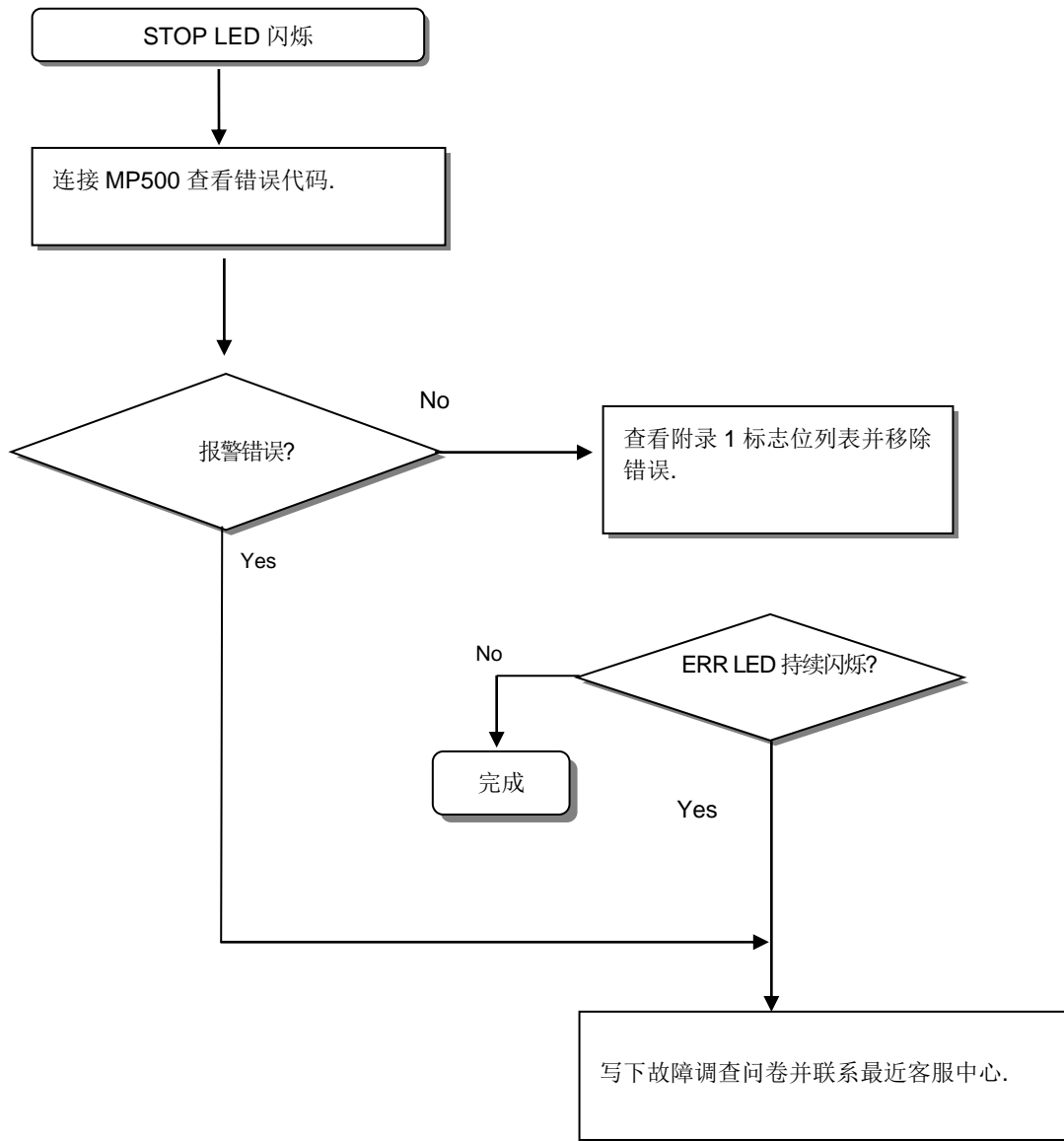
(1) 当PWR (电源) LED熄灭时的故障排除流程图

以下流程图为电源供应时和运行期间电源熄灭的故障排除流程.



(2) 当ERR(错误)LED闪烁时的故障排除流程

以下流程图用于运行期间 ERR LED 闪烁的故障排除流程.以下流程图解释电源提供或者电源 LED 熄灭的纠正措施

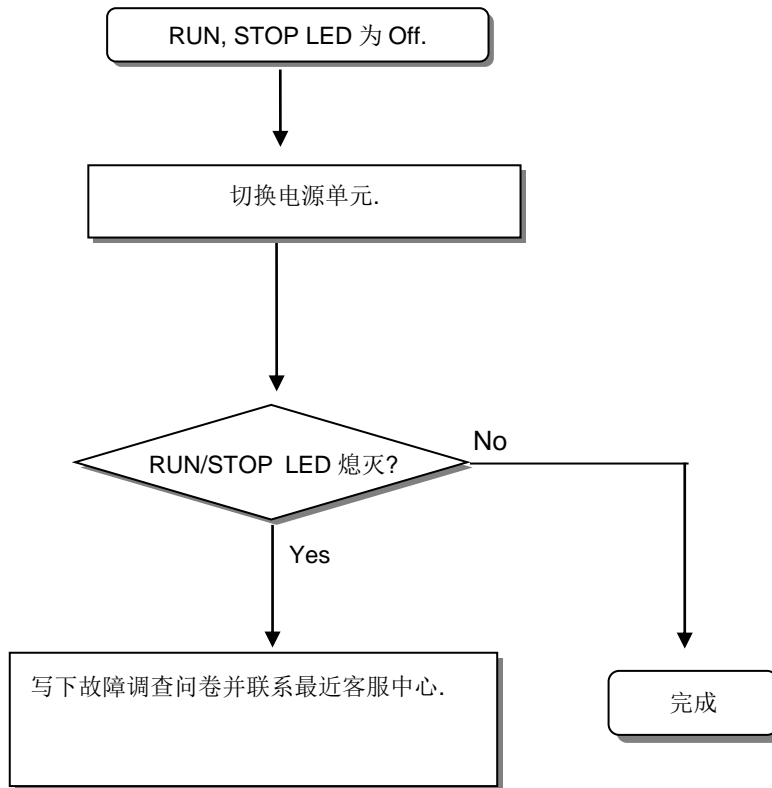


备注

通过出现的错误警告, 运动控制器系统不停止直到执行纠正措施. 如果没有, 可能导致系统错误.

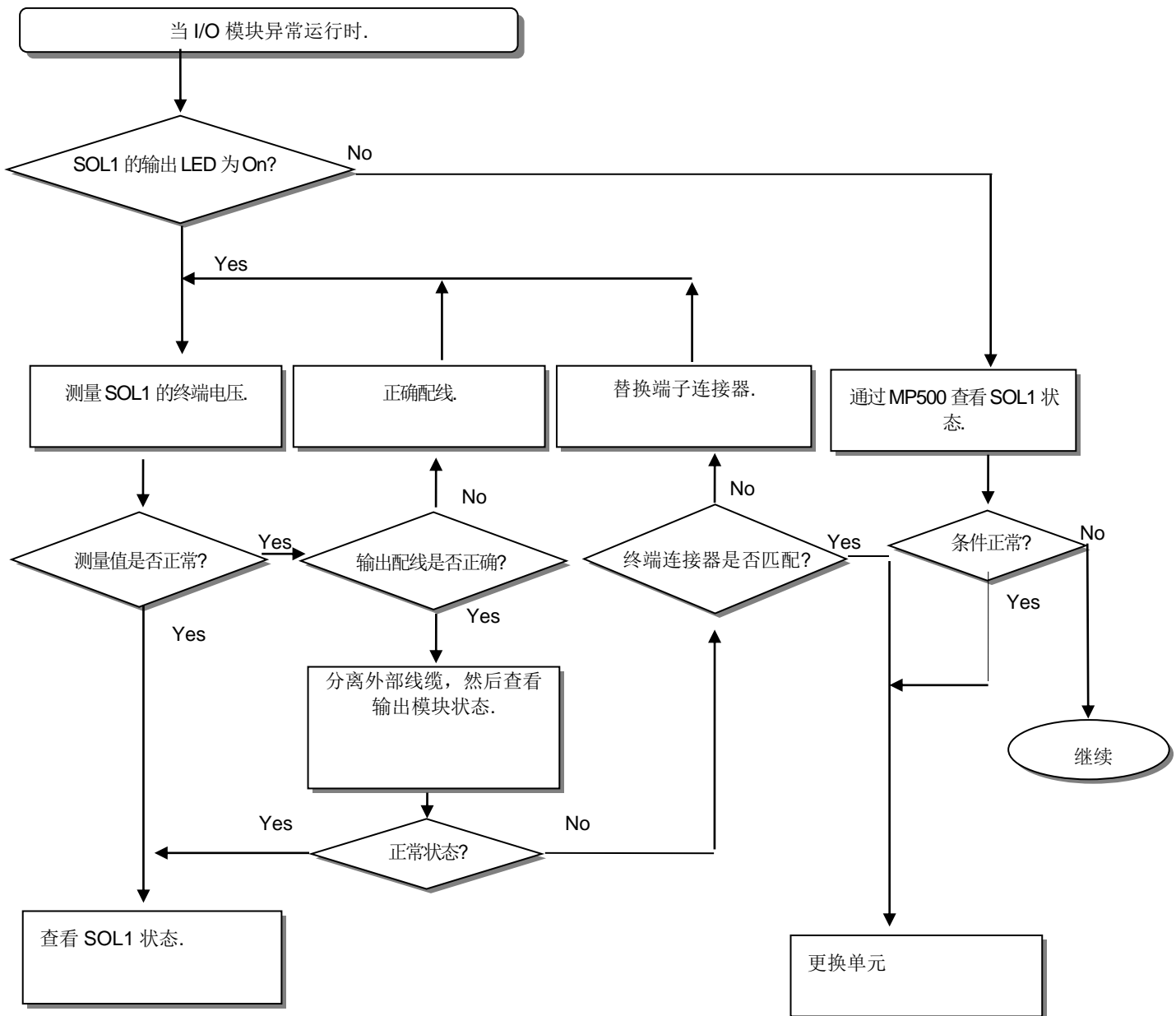
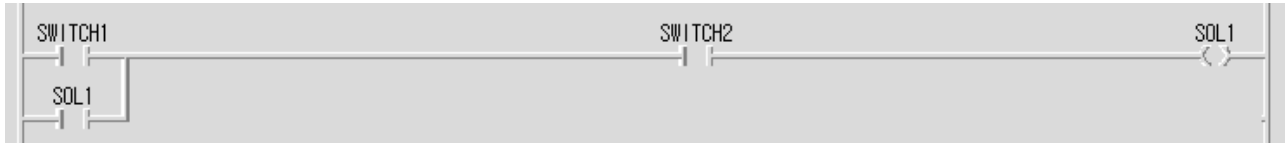
(3) 当RUN, STOP LED熄灭时的故障排除流程图.

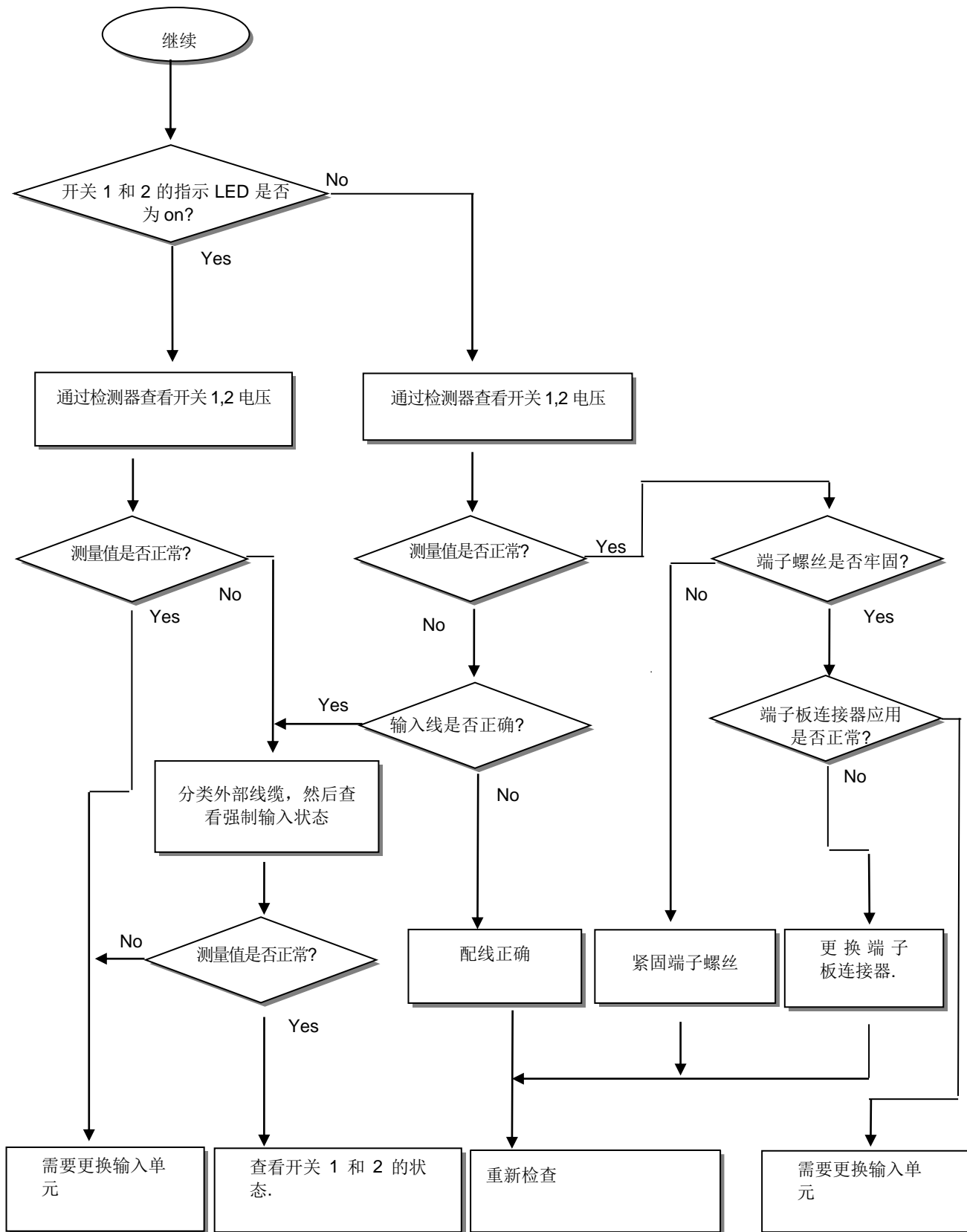
以下解释电源提供时 RUN LED 熄灭的故障排除流程.



(4) 当 I/O 部分未正常运行的故障排除流程图

以下流程图解释当 I/O 模块异常运行时的故障排除流程图。





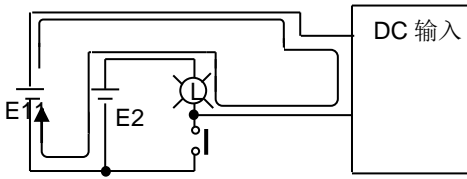
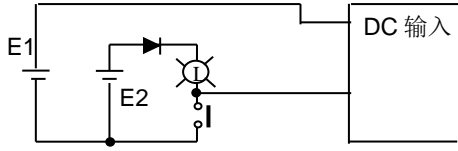
3.7.3 故障排除实例

各种回路中可能的故障以及采取的正确措施说明如下。

(1) 输入回路错误和纠正措施

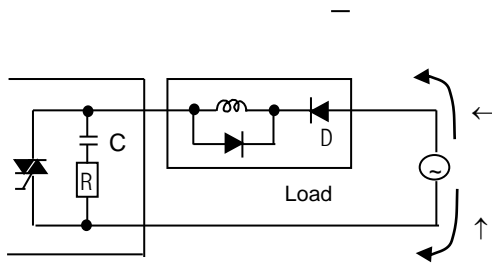
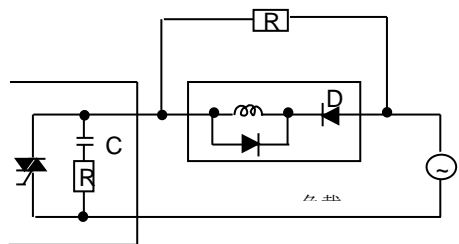
以下描述输入回路可能的错误及纠正措施。

状况	原因	纠正措施
输入信号没有关闭	外部设备漏电流 (比如通过无接近开关运行时)	<ul style="list-style-type: none"> 接入一个适当的电阻和电容，以降低经过输入模块端子的电压。
输入信号没有关闭 (氖灯仍然亮)	外部设备漏电流 (由氖灯限位开关驱动)	<ul style="list-style-type: none"> CR 值由漏电流值决定。 推荐值电容：0.1 ~ 0.47 μF 电阻：47 ~ 120 Ω (1/2W) 或者构成其他独立回路
输入信号没有关闭	由于配线电缆的容量产生漏电流	<ul style="list-style-type: none"> 如下所示在外部设备上定位电源供应。
输入信号没有关闭	外部设备漏电流 (由指示灯开关驱动)	<ul style="list-style-type: none"> 连接一个适当的电阻，以使输入模块端子和公共端子之间的电压高于关闭电压。

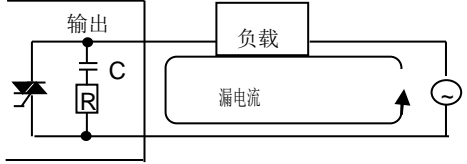
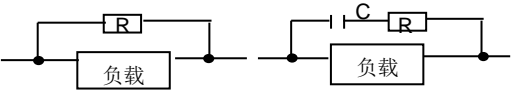
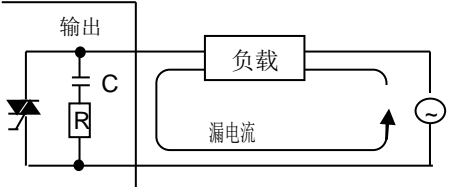
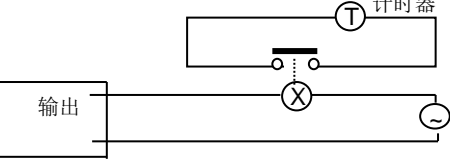
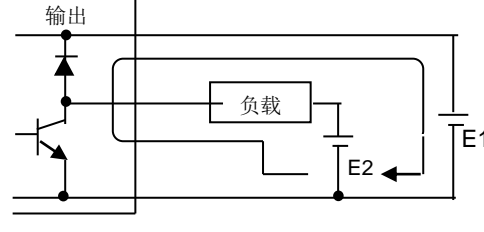
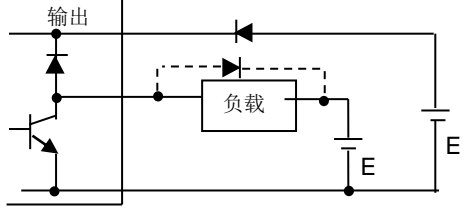
状况	原因	纠正措施
输入信号没有关闭	<ul style="list-style-type: none"> • 双电源的循环电流  <p>DC 输入</p> <ul style="list-style-type: none"> • E1 > E2, 的情况. 	<ul style="list-style-type: none"> • 仅适用单电源供应。 • 接入回路电流保护二极管。  <p>DC 输入</p>

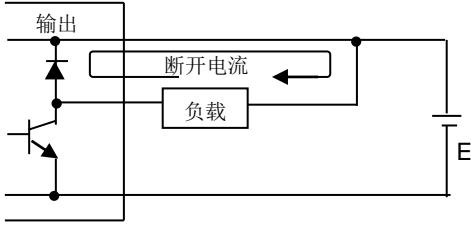
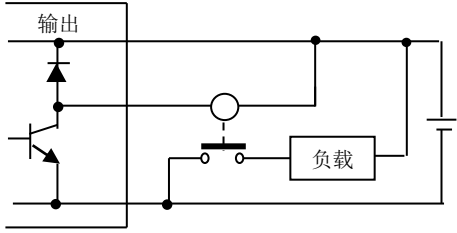
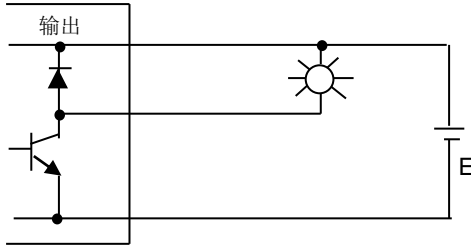
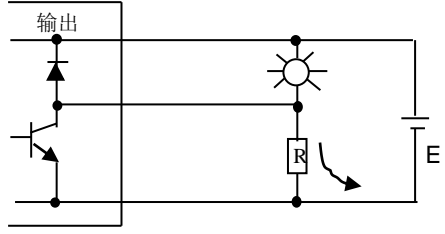
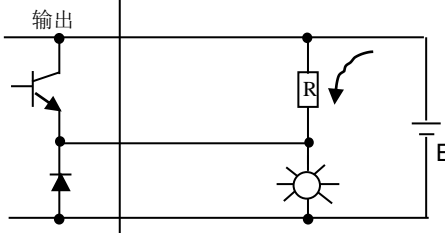
3.7.5 输出回路错误及纠正措施

以下描述输出回路可能的错误，及纠正措施。

状况	原因	纠正措施
当输出关闭时，负载上有过电压。	<ul style="list-style-type: none"> • 内部负载为半波调整(有些情况是螺线管) • 当电源极性如①所示，C 充电。当极性如②所示，电压充电到到 C+ 并提供给二极管 D。最大电压达到 $2\sqrt{2}$。  <p>Load</p> <p>*)如果电阻以这种方式使用,不会对输出单元造成故障.但是它将影响负载电路中的二极管 (D) 的性能,降低引起故障。</p>	<ul style="list-style-type: none"> • 并联一个几十到几百欧姆电阻到负载两边。 

第 3 章 操作顺序与安装

状况	原因	纠正措施
负载没有断开	<ul style="list-style-type: none"> 浪涌吸收器回路引起的漏电流，该浪涌吸收器并联连接到输出元件。 	<ul style="list-style-type: none"> 并联 C、R 元件到负载中，阻值在几十 KΩ。当输出模块与负载之间的配线距离比较长的时候，由于线性电容的存在可能产生漏电流 
当负载是 C-R 类型计时器时，时间常数是波动值。	<ul style="list-style-type: none"> 浪涌吸收器回路引起的漏电流，吸收器并联连接到输出元件。 	<ul style="list-style-type: none"> 用一个触点驱动继电器，用一个锁定触点驱动 C-R 类型的计时器。 使用定时器，而不使用 C-R，有时候有半波螺线管内部电路，因此要小心。 
负载没有断开	<ul style="list-style-type: none"> 2 个不同的电源产生的寄生电流。  <p>E1 < E2, 产生寄生电流. E1 关闭 (E2 打开), 产生寄生电流。</p>	<ul style="list-style-type: none"> 仅使用一个电源 并联寄生电流保护二极管  <p>如果负载是继电器等等，需要如上图中虚线所示的方式反向接入吸收二极管。</p>

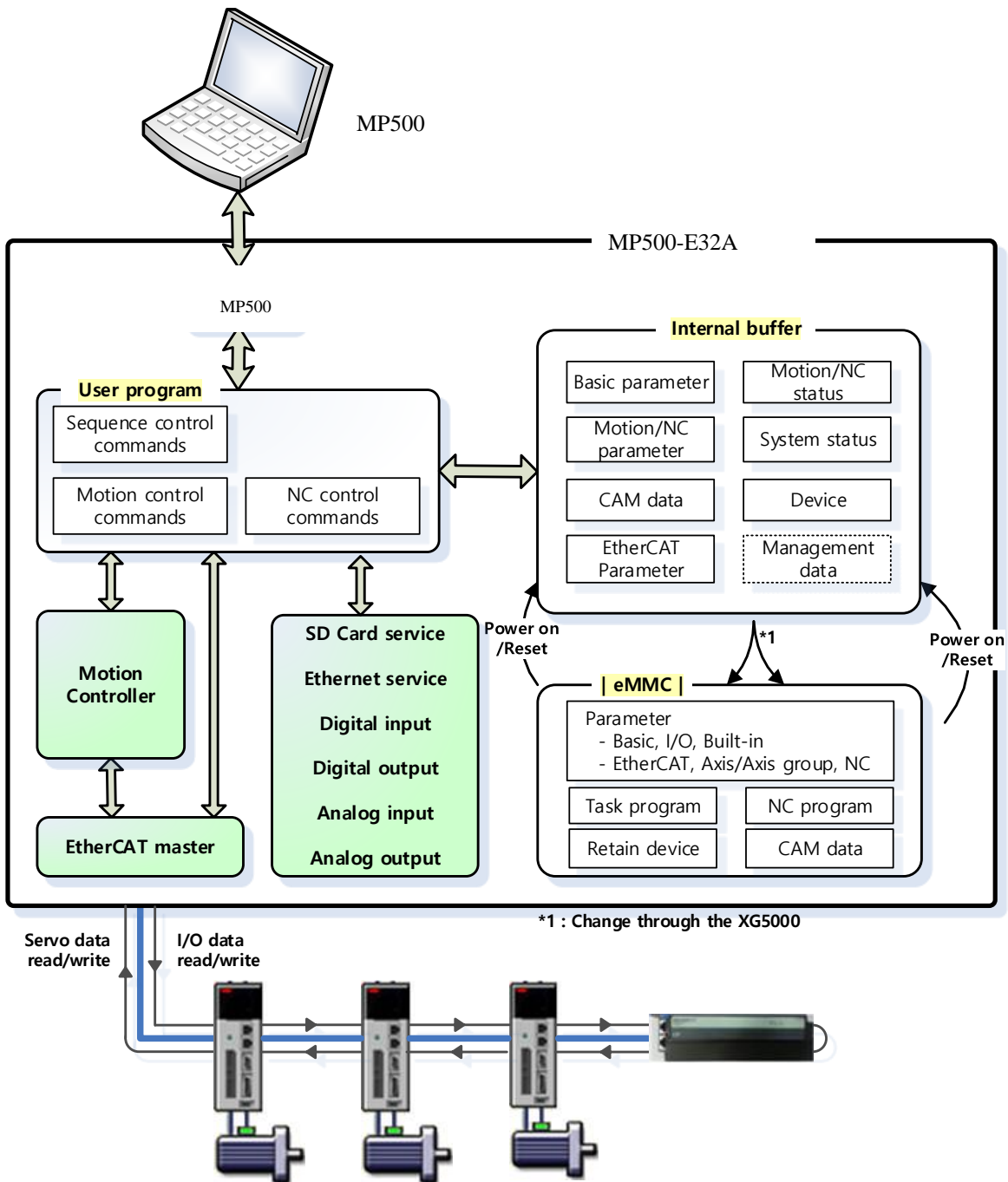
状态	原因	纠正措施
<p>负载断开响应时间长</p>	<p>• 过电流在断开状态[当运行感性负载如螺旋管时](L/R 较大)]直接由晶体管输出驱动。</p>  <p>• 当晶体管输出关闭时，电流会流过二极管，根据负载将会有超过 1 到 2 秒的时间延时。</p>	<p>• 插入一个小的 L/R 电磁接触器并利用同样的触点驱动负载。</p> 
<p>输出三极管损坏</p>	<p>白炽灯的浪涌电流</p>  <p>当灯开的时候将会有 10 倍或更多的浪涌电流。</p>	<p>• 要控制浪涌电流，需要减少额定电流的 1/3 到 1/5。</p>  <p>共地晶体管输出</p>  <p>共源晶体管输出</p>

第 4 章运动控制运行

本节描述运行控制模块的结构、参数和设备。

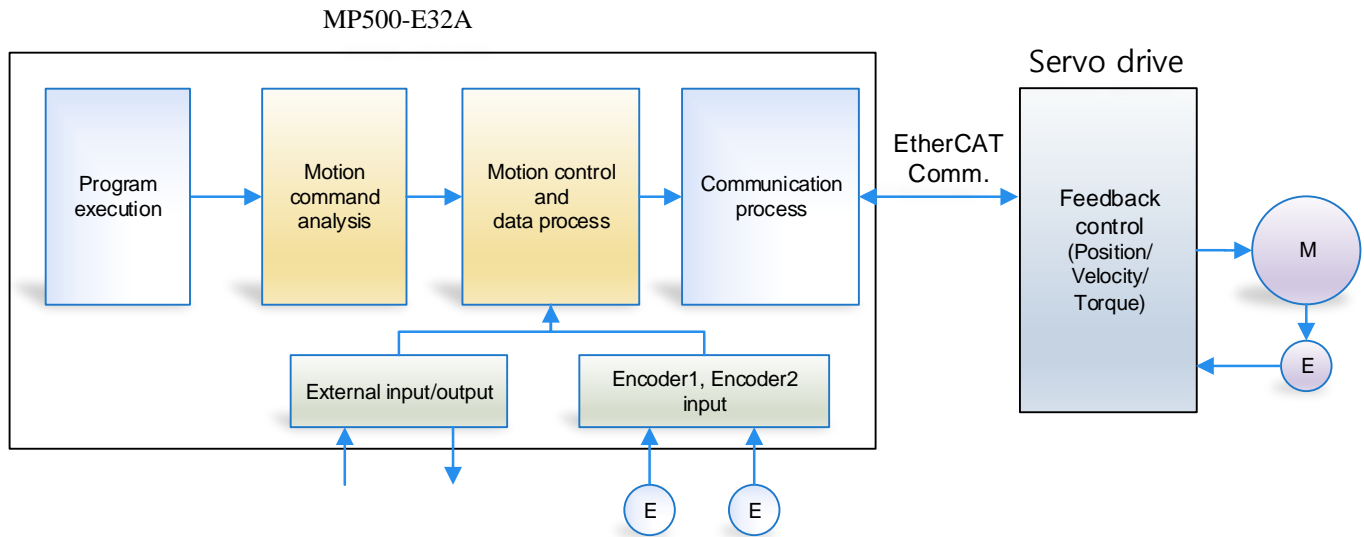
4.1 运动控制模块的结构

下图描述模块中的参数进程和运行数据保存。



4.2 运动控制配置

可通过 EtherCAT 控制最多 32 个电机实轴和 4 个虚轴. 在 32 轴中, 可以通过设置为虚轴控制为连接从站的轴, 仅提供 4 个虚轴. 除了内置 8 点输入和 16 点输出外, 最多可控制 64 个 EtherCAT I/O(包含轴). 运动控制模块的运动控制块状图如下所示.



4.3 运动控制任务

以下描述运动控制模块的任务。

4.3.1 任务类型

有 3 种类型的运动控制任务：主任务，周期性任务和初始化任务。

在用户设置周期内主任务完成动作，执行 I/O 更新，程序处理，运动控制和执行 EtherCAT 同步通讯。主任务设置周期为 1/2/4ms，可在运动控制模块中的基本参数中设置。

周期性任务的周期可由用户设置为主任务周期的倍数，在每个任务周期内，在主任务完成后的剩余时间执行周期性任务。因此，周期性任务可执行于多个主任务周期。

初始化任务仅在当运动控制模块进入 RUN 模式时的开始阶段执行，一般用于设置系统和参数的初始数据。

任务类型	程序数量	执行内容
主任务	最多 256	<ul style="list-style-type: none"> · 执行I/O更新，程序进程分配到主任务和运动控制。 · 在每个固定控制周期(主任务循环)执行一次以上的任务。 · 其优先级高于周期性任务。 · 其使用要求同步控制的程序，通过分配高速运行处理，因为可以快速处理程序。 · 周期可设置为：1ms, 2ms, 4ms
周期性任务		<ul style="list-style-type: none"> · 其执行处理程序进程分配到主任务。 · 在控制周期内，完成主任务运行后剩余时间内执行，可执行多个循环。 · 由于主任务程序中执行运动控制指令时，其优先级低于主任务，所以主任务程序中的运动控制指令优先执行。 · 使用处理其他监控数据的程序，并控制不要求通过分配的高速处理设备。 周期可设置为：1ms ~ 100ms (设置为主任务循环的倍数)
初始化任务		<ul style="list-style-type: none"> · 在完成I/O更新后，执行处理程序分配到初始化任务。 · 仅在进入RUN模式时执行。 · 当进入RUN 模式时优先执行。如果通过初始化任务程序设置初始化任务完成(_INIT_DONE)标志位，任务完成，启动执行主任务和周期性任务程序。

第 4 章运动控制运行

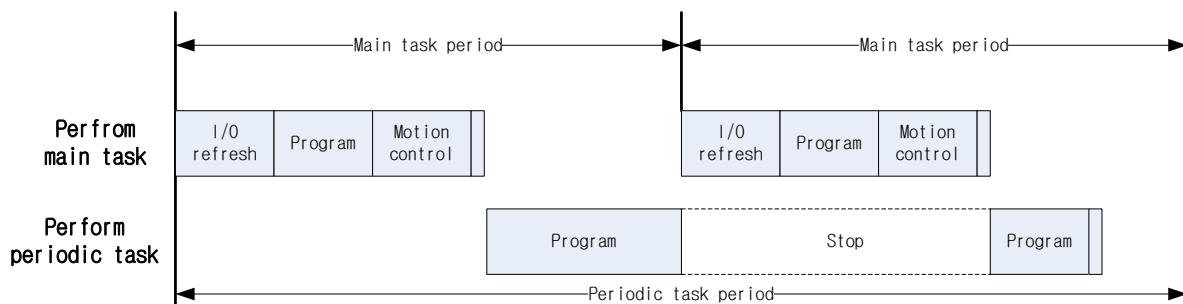
备注

如果主任务循环设置超出范围, 发生错误 0x0260.
 如果周期性任务循环没有设置为主任务的倍数, 发生错误 0x0261.
 如果发生错误, 查看任务循环.

4.3.2 任务运行

1. 总体任务运行

该任务由主任务和周期性任务组成。在控制周期内, 根据程序的进程, 主任务执行 I/O 更新、处理程序以及运动控制动作。周期性任务在主任务完成后剩余时间内的控制周期执行, 可以经过很多个控制周期后完成。

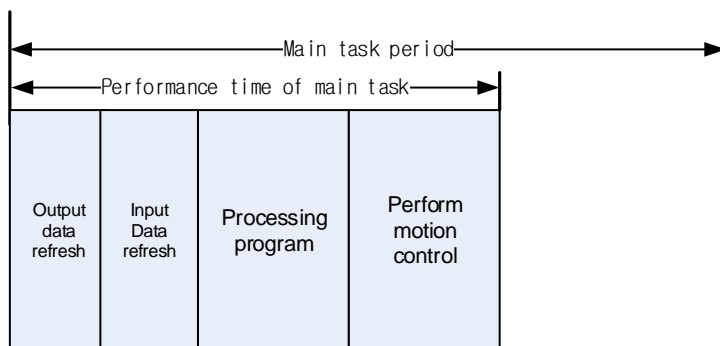


2. 主任务运行

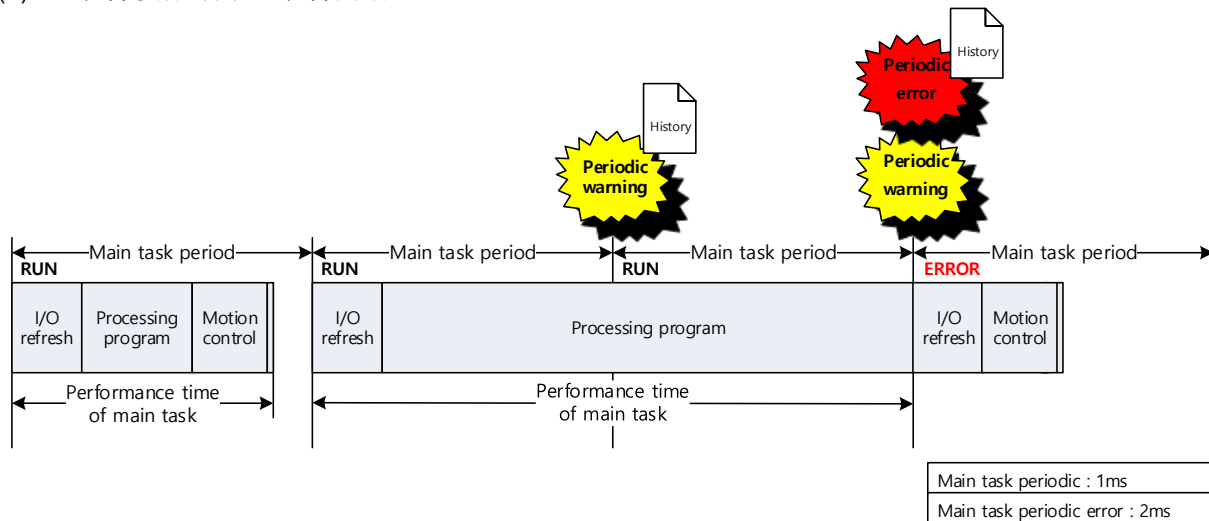
主任务必须在设置的主任务周期内执行, 如果主任务执行时间超出设置的主任务周期, 将发生错误, 如果运动控制模块在 RUN 状态, 变更为 STOP 状态。

如果在‘主任务循环错误’检测时间内主任务执行未完成, 运行立即停止, 如果运动控制模块在 RUN 状态则发生错误. 运动控制模块进入 ERR 状态.

(1) 主任务执行时间 ≤ 主任务周期



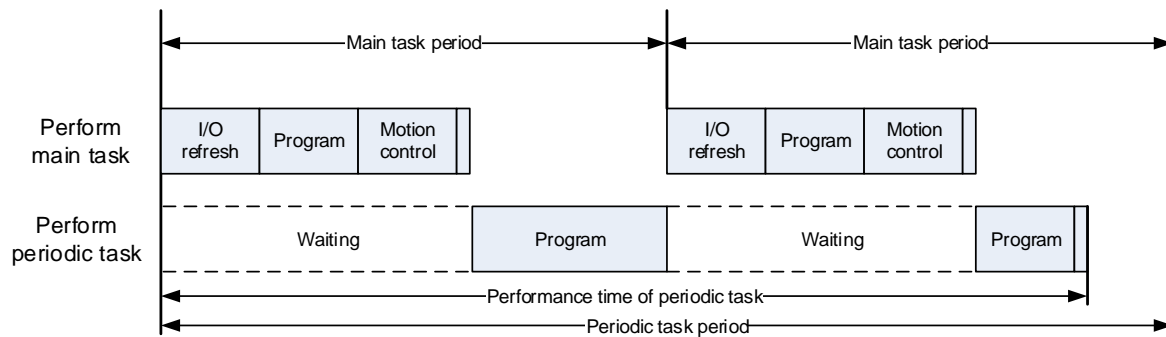
(2) 主任务执行时间 > 主任务周期



3. 周期性任务运行

周期性任务在主任务完成后剩余时间内的设置控制周期执行，可以执行于多个控制周期，取决于任务的执行时间。如果周期性任务的执行时间超出周期性任务设置时间时错误发生。如果在‘主任务循环错误’检测时间内主任务执行未完成，运行立即停止，如果运动控制模块在 RUN 状态则发生错误.运动控制模块进入 ERR 状态.

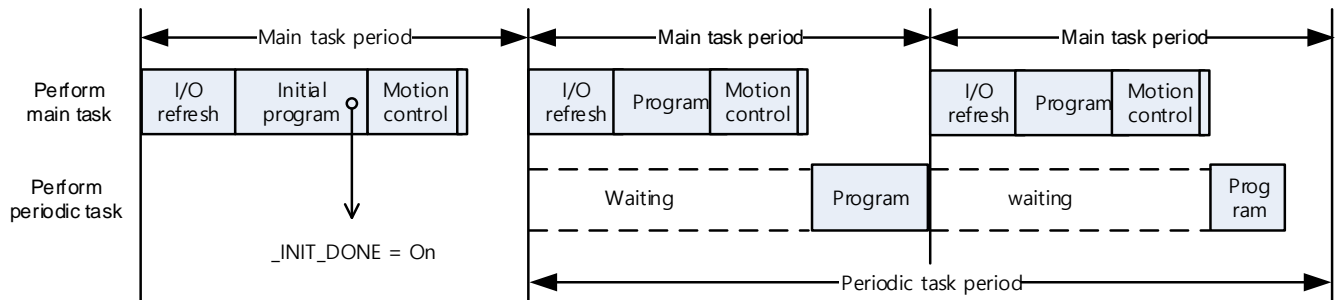
(1) 周期性任务执行时间 ≤ 周期性任务周期



4. 初始化任务运行

初始化任务仅在运动控制模块进入 RUN 模式的开始阶段执行一次。一般用于设置系统和参数的初始数据。初始化任务同样必须以例如主任务的设置任务周期执行，如果初始化任务超出主任务设置周期，将发生错误，并变为停止状态。

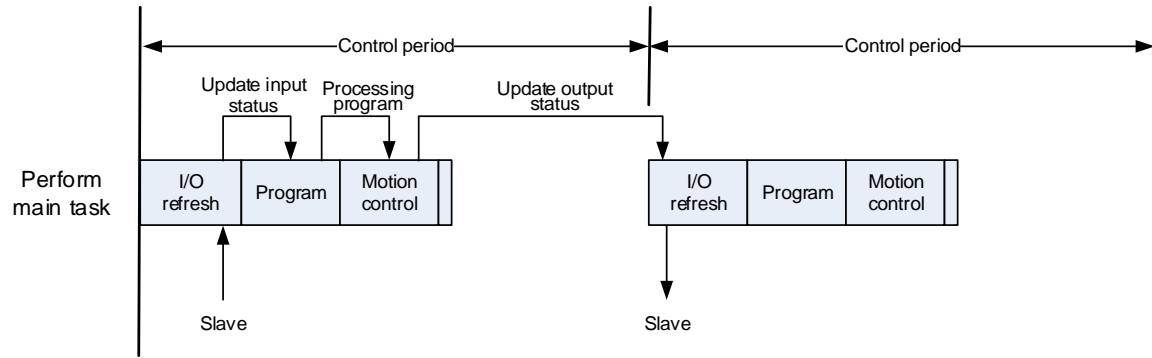
当使用初始任务程序中的基本功能块和运动功能块时，相关功能块的功能可能受到限制。这是由于初始化任务的特性，当进入 RUN 模式时仅执行一次，而对于功能块来说，输出参数不更新。因此，当使用初始化任务程序中的基本功能块和运动功能块时，相关功能块的输出可能不同于实际功能，所以使用时请注意



4.3.3 运动指令的执行

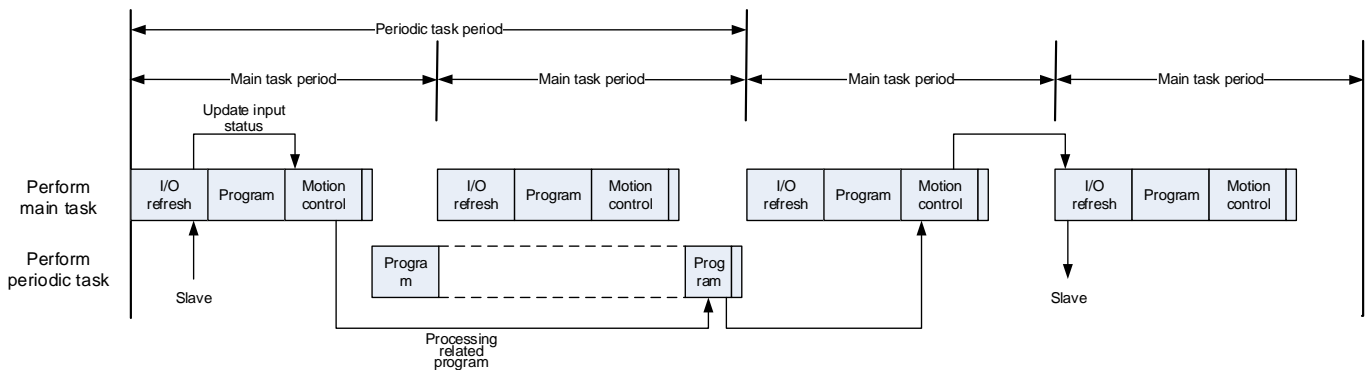
1. 主任务中运动指令的执行

主任务运动指令的执行如下图所示。从站输入值和系统参数由主任务 I/O 更新指令进行更新，基于这个信息，执行程序处理和运动控制。性能结果以下一个控制周期的 I/O 更新时间，在从站模块中输出。



2. 周期性任务运行指令的执行

周期性任务运动指令的执行如下图所示。根据主任务的 I/O 更新动作，从站的输入值和系统参数进行更新，基于这个信息，在主任务中执行运动控制。周期性任务的程序通过这个结果执行，当主任务在周期性任务执行后，并在控制周期内执行时，根据这个结果执行运动控制。同样该运动控制执行结果在从站中输出，以下一个控制周期的 I/O 更新时间。

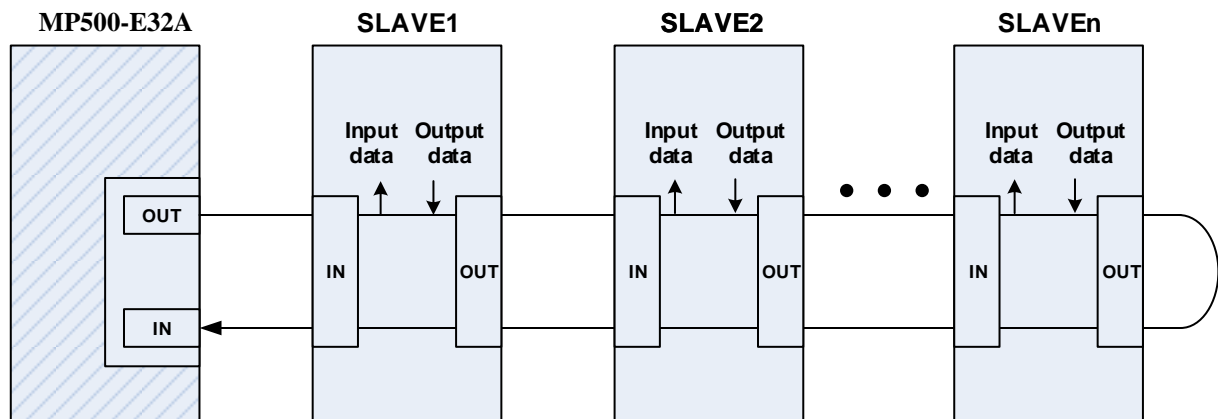


4.4 EtherCAT 通信

EtherCAT(以太网控制自动化技术)通信如下。

4.4.1 什么是EtherCAT

EtherCAT 是一个高性能工业网络系统，用于基于德国倍福公司开发的实时以太网。EtherCAT 是主站和从站直接的通讯，在节点之间通过高速传输以太网结构提供一个较短通讯循环时间。当从主站到从站数据结构通过从站时，EtherCAT 通讯发送接收数据到相关数据框架，同时从站接收传输数据。即，EtherCAT 不传输数据到网络中的每个从站节点，但是通过一个通讯框架依次传输到每个从站，当通讯框架通过每个从站时，每个从站在框架中的相关区域读取和写入数据。通讯框架以通过最后一个从站的结构执行高速数据传输，返回并通过每个从站，然后传输到主站。



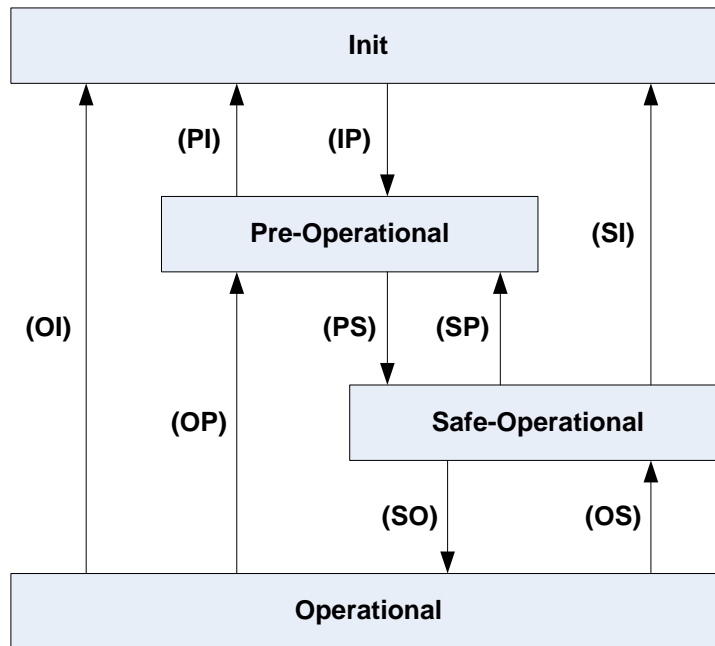
4.4.2 CoE(CANopen覆盖EtherCAT)

运动控制模块使用从站和 EtherCAT 通讯，使用 CoE(CANopen over EtherCAT)作为信息交换协议。在 CoE 中，从站的参数和数据信息由对象代码组成。对象代码包含的信息用于设备和通讯的配置，并且是一组可通过网络访问的对象(参数)。在主-从站通讯中使用 CoE，有一个使用过程数据对象(PDO)和同步传输信息的通讯，服务数据对象(SDO)通讯异步发生。运动控制模块周期性执行接收过程数据通讯和发生输入/输出信号，以及控制 EtherCAT 从站(伺服驱动)的位置。同样根据从站中的错误状态执行服务数据通讯，无论是否要求而进行参数读取/写入。

通讯类型	通讯时间	内容
过程数据通讯(PDO 通讯)	同步 (主任务周期)	伺服驱动位置控制数据, 数据输入/输出等
服务数据通讯(SDO 通讯)	异步 (必要)	伺服参数读取/写入, 伺服错误信息读取等

4.4.3 EtherCAT 状态器

状态及 EtherCAT 通讯状态间的动作如下图所示。

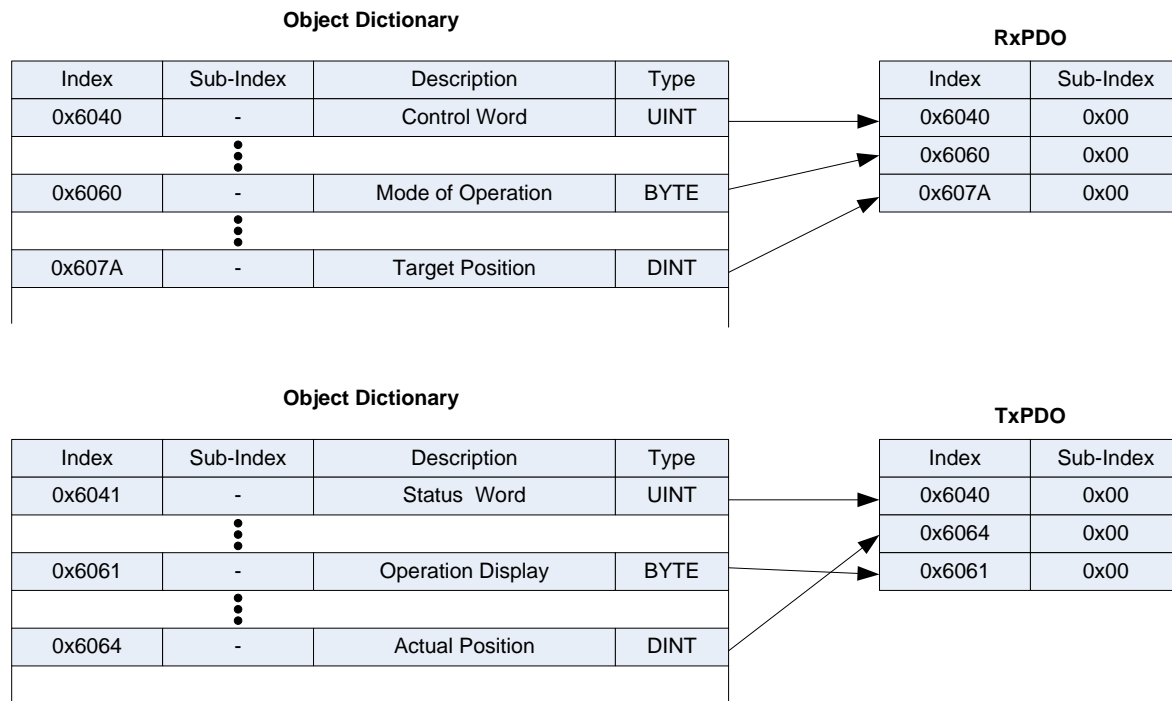


EtherCAT 主-从之间的通讯从初始状态开始到运行状态。在运动控制模块中，当处于运行状态时，从站伺服驱动可通过一个连接的一般处理数据进行控制。

如果当运行状态下运动控制模块执行从站和 EtherCAT 通讯时错误发生，通讯状态变为初始状态，并且从站间的通讯中断。在这种情况下，通讯错误的原因需排除，重新连接从站并重启通讯。

4.4.4 EtherCAT过程数据目标(PDO)

通过过程数据目标(PDO)的运动控制模块 EtherCAT 通讯的同步数据通讯。有两种类型的过程数据：TxPDO，从从站传输到运动控制模块；RxPDO，从运动控制模块传输到从站。在 RxPDO 和 TxPDO 中，将进行异步通讯的数据可如下图实例中的对象代码中进行组合设置。



从站制造商有时会预设很多 RxPDO 和 TxPDO，并提供 xml 格式的包含从站信息的文件。当初始设置和测试运行该从站信息文件时，应该使用 MP500 传输到运动控制模块。该从站信号文件应分析并传输到优化控制的 PDO 数据。

4.4.5 运动控制模块EtherCAT通讯规格

项目	规格
通讯协议	EtherCAT
支持规格	CoE(CANopen 覆盖 EtherCAT)
物理层	100BASE-TX
通讯速度	100Mbps
拓扑	环型
通讯电缆	大于 Cat. 5 STP(屏蔽双绞线)电缆
最大从站数量	64(最多可映射 32 个驱动到运动轴)
通讯间隔	0.5ms/1ms/2ms/4ms
同步抖动	0.5ms/1ms/2ms/4ms
同步通讯	通过 CoE 的 PDO(过程数据对象) 映射
异步通讯	通过 CoE 的 SDO(服务数据对象)通讯
通讯设置	使用 MP500 设置通讯配置

4.5 运动控制程序

4.5.1 程序运行

1. 程序配置

运动控制程序由执行某些特定控制的所需的功能要素组成，并且在运动控制模块的内部 RAM 中执行。程序在闪存中备份。

具有以下功能因素的程序分类如下。

程序	处理信息
主任务程序	处理在每个'主任务周期'执行的指令。
周期性任务程序	处理在每个'周期性任务周期'执行的指令。
初始化任务程序	一旦运动控制模块为 RUN 状态下执行的指令

备注

当电源 off 时由于运动控制程序保存在 eMMC 中，限制程序写入次数为 100,000 次。
当使用程序时请注意。

4.5.2 运行模式

1. RUN 模式

该模式正常执行运动程序计算。

(1) 当改变模式时处理

初始化在开始阶段的数据区域中执行，通过检查运动程序的有效性来决定执行的可能性。

(2) 计算处理的内容

动作程序，动作指令计算，输入/输出数据处理，执行 EtherCAT 通讯。

2. STOP 模式

该模式是不执行动作程序计算的停止状态。

(1) 当改变模式时处理

每个输出数据为 Off 状态。

(2) 运行处理的内容

执行 EtherCAT 通讯。

(3) 可以在不运行运动程序的情况下执行在指令窗口中的指令。执行指令窗口中的指令，执行 EtherCAT 通讯。

3. 变更运行模式

运动控制模块的运行模式可按照如下变更.

运行模式	备注
RUN	运动控制器执行程序.
STOP → RUN	运动控制器从 STOP 模式变更为 RUN 模式.
STOP → RUN	运动控制器从 RUN 模式变更为 STOP 模式.
STOP	仅当运动控制器在 MP500 的 STOP 模式下可以执行 TEST 指令.

第 5 章内存和参数

5.1 内存

5.1.1 程序和 DATA 存储

1. 程序存储

运动控制器中程序相关存储配置如下.

类型	尺寸	描述
参数	9,251KB	用户参数区域
运动程序	10,240KB	运行相关用户程序
NC 程序	10,240KB	NC 相关用户程序
程序运行表	4,478KB	程序相关表区域
系统运行	47,104KB	系统OS区域

2. DATA 存储

运动控制器中 DATA 存储的详情及尺寸如下.

类型	尺寸	描述
用户设备	自动变量 (A)	4,096KB 自动变量区域
	直接变量 (M)	2,048KB 内部设备区域
	输入变量 (I)	16KB 内置数字输入, EtherCAT 从站TxPDODATA
	输出变量 (Q)	16KB 内置数字输出, EtherCAT 从站RxPDODATA
	系统变量(F)	128KB 运动控制状态和模块运行状态变量
	特殊变量(U)	1KB 内置模拟量运行和状态变量
	特殊变量(K)	18KB SD存储, DATA记录和编码器标志位区域
	其他	456KB UDF/B 内部使用和NC局部变量
历史	88KB 用户历史(错误/模式/系统/电源/动作错误)	
系统运行	43,008KB 系统内部运行区域和其他功能	

第 5 章 内存和参数

5.1.2 设备

1. 设备类型

运动控制模块中支持的设备类型如下表。

类型	尺寸	描述
自动变量 (A)	4,096KB	自动变量区域 (可设置2,408KB保持)
直接变量 (G)	2,048KB	内部设备区域 (通过选择基本参数区域设置1,024KB保持)
输入变量 (I)	16KB	内置数字输入, EtherCAT 从站TxPDODATA
输出变量 (Q)	16KB	内置数字输出, EtherCAT 从站RxPDODATA
系统变量 (F)	128KB	运动控制状态和模块运行状态变量
特殊变量 (U)	1KB	内置模拟量运行和状态变量
特殊变量 (U)	18KB	SD存储, DATA记录和编码器标志位区域

(1) 自动变量

- (a) 这是一个通过编译器自动分配变量位置的变量; 用户不需要指定内部变量位置. 用户设置而不分配指定位置的变量被配置为自动变量.
- (b) 当电源重启或者从 STOP 变为 RUN 时没有保持设置的自动变量初始为 0.

(2) 直接变量

- (a) 该变量为用户使用名称和直接设备数量强制分配存储位置.
- (b) 直接变量可用的地址分配范围如下.

尺寸变量	变量地址指定范围
X(位)	%MX0 ~ %MX16777215
B(字节)	%MB0 ~ %MB2097151
W(字)	%MW0 ~ %MW1048575
D(双字)	%MD0 ~ %MD524287
L(长字)	%ML0 ~ %ML262143

(3) 输入变量

- (a) 该变量分配为内置数字输入和 EtherCAT 从站的 TxPDO.
- (b) 内置数字输入为 8 点.
- (c) 输入变量表示如下.

%I[尺寸前缀]n

数量	描述
尺寸前缀	X(1 位), B(1 字节), W(1 字), D(1 双字), L(1 长字)
n	NDATA 基于[尺寸前缀]

%I[尺寸前缀]n1.n2.n3

数量	描述
尺寸前缀	X(1 位), B(1 字节), W(1 字), D(1 双字), L(1 长字)
n1	0~127 块分配
n2	0~15 块分配
n3	64 位分配. n3DATA 基于[尺寸前缀]

例) %IW64 = %IB128 = %IW1.0.0 = %IB1.0.0, %IW1 = %IB2 = %IW0.0.1 = %IB0.0.2

(d) 基于输入变量表达的设备分配如下.

设备	描述
%IX0	内置数字输入0
%IX1	内置数字输入1
%IX2	内置数字输入2
%IX3	内置数字输入3
%IX4	内置数字输入4
%IX5	内置数字输入5
%IX6	内置数字输入6
%IX7	内置数字输入7
%IW64~	EtherCAT从站TxPDO映射DATA

(4) 输出变量

(a) 该变量分配为内置数字输出和 EtherCAT 从站 RxPDO.

(b) 内置数字输出为 16 点.

(c) 输入变量表达如下.

%Q[尺寸前缀]n

数量	描述
尺寸前缀	X(1 位), B(1 字节), W(1 字), D(1 双字), L(1 长字)
n	NDATA 基于[尺寸前缀]

%Q[尺寸前缀]n1.n2.n3

数量	描述
尺寸前缀	X(1 位), B(1 字节), W(1 字), D(1 双字), L(1 长字)
n1	0~127 块分配
n2	0~15 块分配
n3	64 位分配. n3DATA 基于[尺寸前缀]

例) %QW64 = %QB128 = %QW1.0.0 = %QB1.0.0, %QW1 = %QB2 = %QW0.0.1 = %QB0.0.2

第 5 章 内存和参数

(d) 基于输出变量表达的设备分配如下.

设备	描述
%QX0	内置数字输出 0
%QX1	内置数字输出 1
%QX2	内置数字输出 2
%QX3	内置数字输出 3
%QX4	内置数字输出 4
%QX5	内置数字输出 5
%QX6	内置数字输出 6
%QX7	内置数字输出 7
%QX8	内置数字输出 8
%QX9	内置数字输出 9
%QX10	内置数字输出 10
%QX11	内置数字输出 11
%QX12	内置数字输出 12
%QX13	内置数字输出 13
%QX14	内置数字输出 14
%QX15	内置数字输出 15
%QW64~	EtherCAT从站RxPDO映射DATA

(5) 特殊变量

(a) 该变量分配为内置模拟量输入和输出.

(b) 内置模拟量输入为 2 通道, 并且内置模拟量输出为 2 通道.

(c) 内置模拟量变量表达式如下.

%I[尺寸前缀]n1.n2.n3

数量	描述
尺寸前缀	X(1 位), B(1 字节), W(1 字), D(1 双字), L(1 长字)
n1	0: 运动控制器
n2	1: 内置模拟量
n3	n3DATA 基于 n2DATA 中的[尺寸前缀]

(d) 基于模拟量变量表达式的设备分配如下.

内置模拟量输入

变量	类型	设备	描述
_01_AD0_ACT	BOOL	%UX0.1.16	通道 0 激活
_01_AD0_AVGTYPE	BYTE	%UB0.1.34	通道 0 平均类型
_01_AD0_AVGVAL	WORD	%UW0.1.18	通道 0 平均值
_01_AD0_DATA	WORD	%UW0.1.5	通道 0 输出数据
_01_AD0_DATATYPE	BYTE	%UB0.1.26	通道 0 输出数据类型设置
_01_AD0_ERR	BOOL	%UX0.1.32	通道 0 错误

变量	类型	设备	描述
_01_AD0_FILTCONST	WORD	%UW0.1.15	通道0 滤波常数
_01_AD0_HOLDVAL	BOOL	%UX0.1.320	通道0 保持有效转换值设置
_01_AD0_HOOR	BOOL	%UX0.1.48	通道0 警报 (上限)
_01_AD0_IDD	BOOL	%UX0.1.72	通道0 输入断开标志位
_01_AD0_LOOR	BOOL	%UX0.1.56	通道0 警报 (下限)
_01_AD0_RANGE	BYTE	%UB0.1.22	通道0 范围设置
_01_AD0_RUN	BOOL	%UX0.1.160	通道0 运行设置
_01_AD1_ACT	BOOL	%UX0.1.17	通道1 激活
_01_AD1_AVGTYPE	BYTE	%UB0.1.35	通道1 平均处理
_01_AD1_AVGVAL	WORD	%UW0.1.19	通道1 平均值设置
_01_AD1_DATA	WORD	%UW0.1.6	通道1 输出数据
_01_AD1_DATATYPE	BYTE	%UB0.1.27	通道1 输出数据类型设置
_01_AD1_ERR	BOOL	%UX0.1.33	通道1 错误
_01_AD1_FILTCONST	WORD	%UW0.1.16	通道1 滤波常数
_01_AD1_HOLDVAL	BOOL	%UX0.1.321	通道1 维持有效转换值设置
_01_AD1_HOOR	BOOL	%UX0.1.49	通道1 警报 (上限)
_01_AD1_IDD	BOOL	%UX0.1.73	通道1 输入断开标志位
_01_AD1_LOOR	BOOL	%UX0.1.57	通道1 警报 (下限)
_01_AD1_RANGE	BYTE	%UB0.1.23	通道1 范围设置
_01_AD1_RUN	BOOL	%UX0.1.161	通道1 运行设置
_01_AD_ACT_ARY	阵列[0..1] OF BOOL	%UX0.1.16	每通道激活(阵列)
_01_AD_AVGTYPE_ARY	阵列[0..1] OF BYTE	%UB0.1.32	每通道平均类型(阵列)
_01_AD_AVGVAL_ARY	阵列[0..1] OF WORD	%UW0.1.18	每通道平均值(阵列)
_01_AD_DATATYPE_ARY	阵列[0..1] OF BYTE	%UB0.1.26	每通道数据类型设置(阵列)
_01_AD_DATA_ARY	阵列[0..1] OF WORD	%UW0.1.5	每通道输出数据(阵列)
_01_AD_ERR_ARY	阵列[0..1] OF BOOL	%UX0.1.32	每通道错误(阵列)
_01_AD_FILTCONST_ARY	阵列[0..1] OF WORD	%UW0.1.15	每通道滤波常数(阵列)
_01_AD_HOLDVAL_ARY	阵列[0..1] OF BOOL	%UX0.1.320	每通道维持有效转换值 (阵列) 设置
_01_AD_HOOR_ARY	阵列[0..1] OF BOOL	%UX0.1.48	每通道警报 (上限) (阵列)
_01_AD_IDD_ARY	阵列[0..1] OF BOOL	%UX0.1.72	每通道输入断开标志位(阵列)
_01_AD_LOOR_ARY	阵列[0..1] OF BOOL	%UX0.1.56	每通道警报 (下限)(阵列)
_01_AD_RANGE_ARY	阵列[0..1] OF BYTE	%UB0.1.22	每通道范围设置(阵列)
_01_AD_RUN_ARY	阵列[0..1] OF BOOL	%UX0.1.160	每通道运行设置(阵列)

第 5 章 内存和参数

内置模拟量输出

变量	类型	设备	描述
_01_DA0_ACT	BOOL	%UX0.1.24	通道 0(电压) 激活
_01_DA0_DATA	WORD	%UW0.1.8	通道 0(电压) 输入 DATA
_01_DA0_DATATYPE	BYTE	%UB0.1.28	通道 0(电压) 输入 DATA 类型
_01_DA0_ERR	BOOL	%UX0.1.40	通道 0(电压)错误
_01_DA0_INTP	BOOL	%UX0.1.64	通道 0(电压)插值使能
_01_DA0_INTPMTHD	BYTE	%UB0.1.46	通道 0(电压)插值方式
_01_DA0_INTPTIME	BYTE	%UB0.1.48	通道 0(电压)插值时间设置
_01_DA0_INTPVAL	WORD	%UW0.1.25	通道 0(电压) 插值数值
_01_DA0_OUTEN	BOOL	%UX0.1.112	通道 0(电压) 输出使能
_01_DA0_OUTSTAT	WORD	%UW0.1.21	通道 0(电压) 输出状态设置
_01_DA0_RANGE	BYTE	%UB0.1.24	通道 0(电压) 范围设置
_01_DA0_RUN	BOOL	%UX0.1.168	通道 0(电压) 运行设置
_01_DA1_ACT	BOOL	%UX0.1.25	通道 1(电压) 激活
_01_DA1_DATA	WORD	%UW0.1.9	通道 1(电压) 输入
_01_DA1_DATATYPE	BYTE	%UB0.1.29	通道 1(电压) 输入数据类型设置
_01_DA1_ERR	BOOL	%UX0.1.41	通道 1(电压) 错误
_01_DA1_INTP	BOOL	%UX0.1.65	通道 1(电压) 插值使能
_01_DA1_INTPMTHD	BYTE	%UB0.1.47	通道 1(电压) 插值方式
_01_DA1_INTPTIME	BYTE	%UB0.1.49	通道 1(电压) 插值时间设置
_01_DA1_INTPVAL	WORD	%UW0.1.26	通道 1(电压) 插值数值
_01_DA1_OUTEN	BOOL	%UX0.1.113	通道 1(电压) 输出使能
_01_DA1_OUTSTAT	WORD	%UW0.1.22	通道 1(电压) 输出状态设置
_01_DA1_RANGE	BYTE	%UB0.1.25	通道 1(电压) 范围设置
_01_DA1_RUN	BOOL	%UX0.1.169	通道 1(电压) 运行设置
_01_DA_ACT_ARY	阵列[0..1] OF BOOL	%UX0.1.24	每通道激活状态(阵列)
_01_DA_DATATYPE_ARY	阵列[0..1] OF BYTE	%UB0.1.28	每通道输入数据类型通道(阵列)
_01_DA_DATA_ARY	阵列[0..1] OF WORD	%UW0.1.8	每通道输入数据(阵列)
_01_DA_ERR_ARY	阵列[0..1] OF BOOL	%UX0.1.40	每通道错误(阵列)
_01_DA_INTPMTHD_ARY	阵列[0..1] OF BYTE	%UB0.1.46	每通道插值方式设置(阵列)
_01_DA_INTPTIME_ARY	阵列[0..1] OF BYTE	%UB0.1.48	每通道插值时间设置(阵列)
_01_DA_INTPVAL_ARY	阵列[0..1] OF WORD	%UW0.1.25	每通道插值数值(阵列)
_01_DA_INTP_ARY	阵列[0..1] OF BOOL	%UX0.1.64	每通道插值使能(阵列)
_01_DA_OUTEN_ARY	阵列[0..1] OF BOOL	%UX0.1.112	每通道输出使能设置(阵列)
_01_DA_OUTSTAT_ARY	阵列[0..1] OF WORD	%UW0.1.21	每通道输出状态设置(阵列)
_01_DA_RANGE_ARY	阵列[0..1] OF BYTE	%UB0.1.24	每通道范围设置(阵列)
_01_DA_RUN_ARY	阵列[0..1] OF BOOL	%UX0.1.168	每通道运行设置(阵列)

内置模拟量共用

变量	类型	设备	描述
_01_ERR	BOOL	%UJX0.1.0	运动控制器错误
_01_RDY	BOOL	%UJX0.1.15	运动控制器准备
_01_SETTINGERR	WORD	%UJW0.1.27	设置错误信息

(6) 特殊变量

- (a) 这些变量分配 SD 存储, 数据记录和嵌入式编码器标志位区域.
- (b) 内置编码器输入 2 通道.
- (c) 对于分配特殊变量的存储, 请参考附录 1 标志位列表 7) SD 存储标志位 ~ 9) 编码器标志位.

(7) 系统变量

- (a) 这些变量分配运动控制器状态和系统状态的状态变量.
- (b) 标志位类型更多内容, 请参考附录 1 标志位列表 1) 系统标志位.

2. 保持设置

当希望保持并使用运行或者运动控制模块停止并重启后的请求数据时, 用于默认(自动)变量保持, 通过基本参数设置 M 区域中设备的某一部分可用于保持区域.

用于设置保持的设备特性表如下.

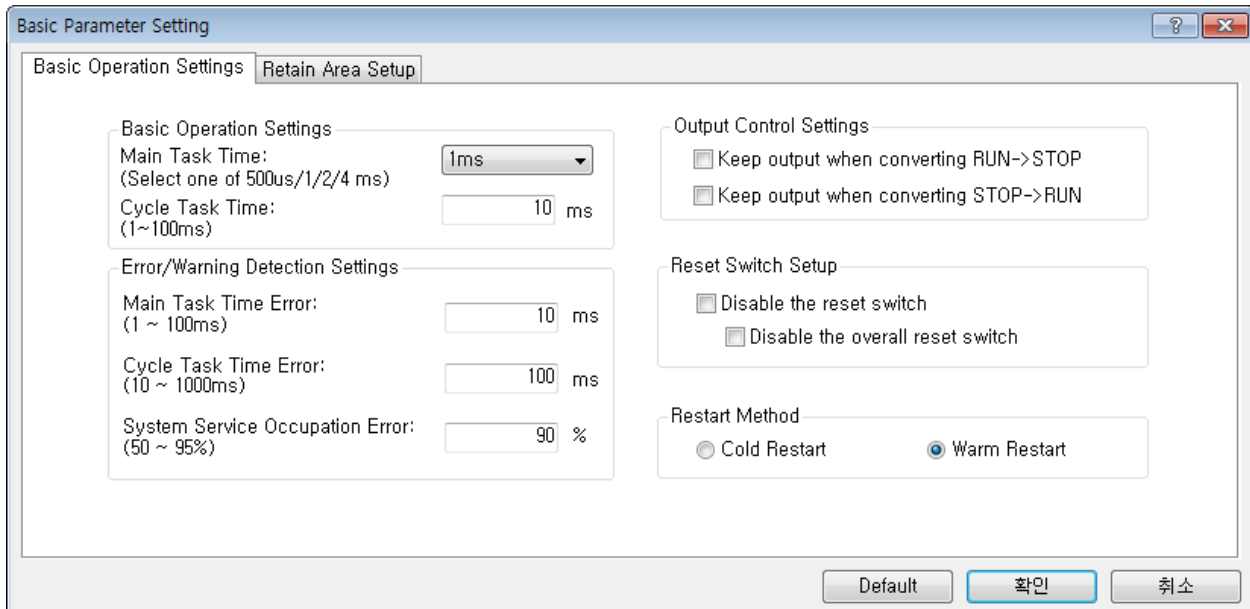
设备	保持设置	特性
默认	○	当增加变量到自动变量区域时使能保持设置
M	○	内置接触区域并使能参数中的保持设置
I	X	内置数字输入, EtherCAT 从站的TxPDO DATA
Q	X	内置数字输出, EtherCAT 从站的RxPDO DATA

5.1.3 参数

1. 基本参数

解释运动控制模块的基本参数。

(1) 基本动作设置



(a) 主任务循环

- 设置主任务的动作间隙。间隙可设置为 0.5ms/1ms/2ms/4ms。
- 设置运动控制模块中主任务执行的控制时间，要考虑程序执行时间。
- 当主任务执行时间超出主任务间隙时，错误发生，如果运动控制模块为 RUN 状态，则变为 STOP 状态，运动控制器立即停止运行。

(b) 周期性任务循环

- 设置周期性任务的动作间隙。周期可在 1~100ms 中设置主任务的倍数。
- 周期性任务在控制周期内主任务执行完成后的剩余时间内执行，因此，可以执行超过多个控制周期。

(c) 主任务检测循环错误

- 设置当任务运行超出设置时间时导致错误的主任务运行时间。设置范围为 1~100ms。

(d) 周期性任务检测循环错误

- 设置当任务运行超出设置时间时导致错误的周期性运行时间。设置范围为 10~100ms。

(e) 任务程序占有率警告

- 如果由于过多的主任务程序或者周期性任务程序而任务程序占有率超出设置值, 发生任务程序占有率警告. 可设置范围 50~95%. 如果任务程序占有率超过 100%, 发生任务程序占有率错误, 变更为 ERROR 状态.

(f) 输出控制设置

当模块发生错误或者变更运行方式时, 确定是否保持数据输出.

选择	运行
当 RUN 变更为 STOP 时保持输出	确定当运动控制模块的运行方式从 RUN 变为 STOP 时正常输出数据.
当 STOP 变更为 RUN 时保持输出	确定当运动控制模块的运行方式从 STOP 变为 RUN 时正常输出数据.

(g) 设置是否关断复位开关

- 设置是否'复位'产品前部面板的开关. 该项目可设置为'允许'或者'禁止'.

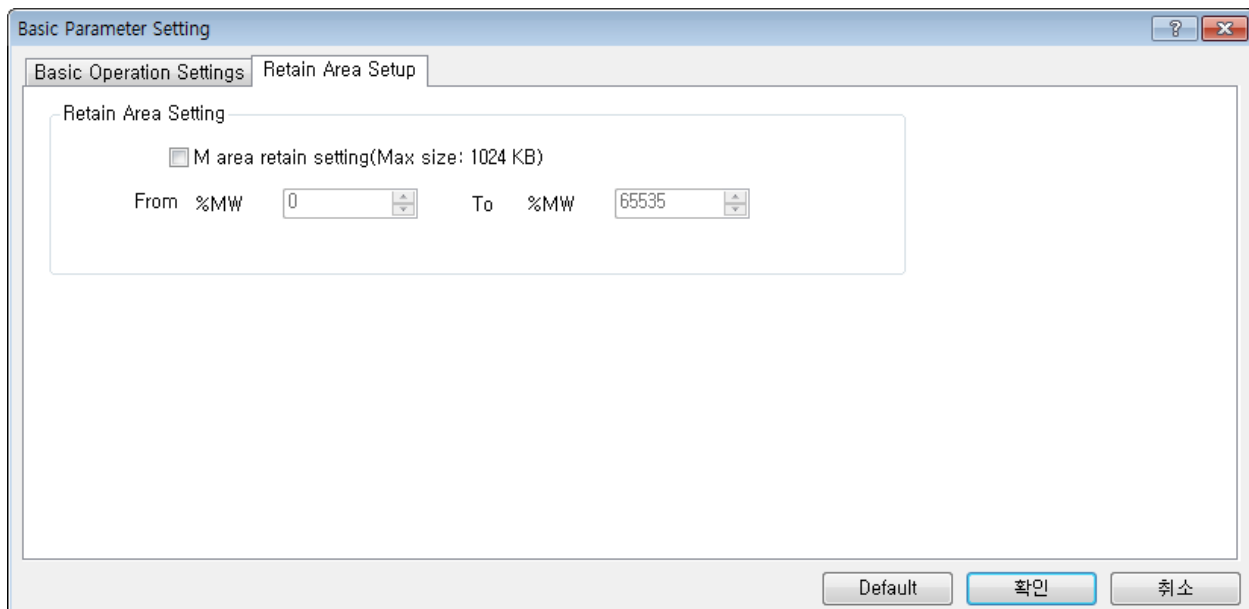
(h) 重启模式

- 通过运行控制器复位和电源切换的重启分为冷重启和热重启. 关于重启模式, 变量可设置为 3 个不同类型, 例如默认,初始,和保持; 根据重启模式的变量初始如下.

变量	冷重启	热重启
默认	初始为'0'	初始为'0'
保持	初始为'0'	保持前一个值
初始	初始为用户自定义值	初始为用户自定义值
保持&初始	初始为用户自定义值	保持前一个值

(2) 存储区设置

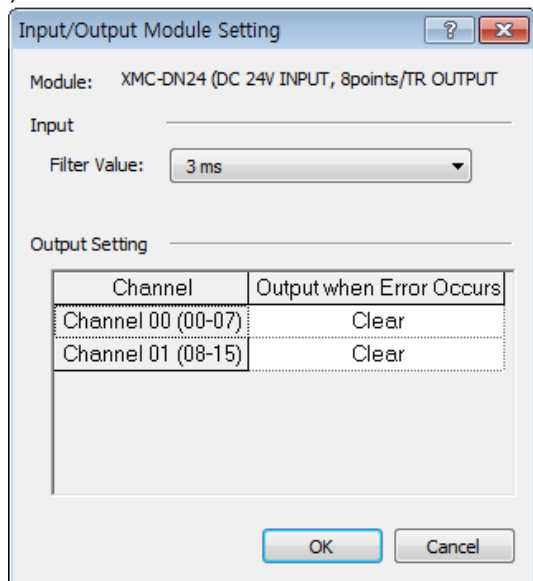
该参数项目在保持区域设置. 通过查看“M 区域保持设置”激活保持区域设置. 保持最多可设置 1,024Kbyte, 如果起始和结束地址设置为保持 M 区域, 即使电源断电, 相关区域值保持.



2. I/O 参数

第 5 章 内存和参数

(1) 内置输入/输出设置

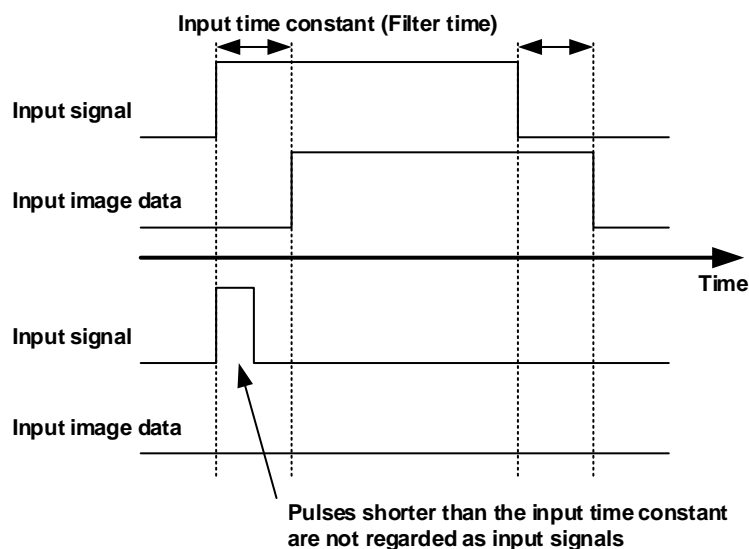


(a) 输入滤波功能

运动控制器内置输入部分具有防止外部干扰信号进入输入信号的输入滤波功能。在干扰较多或者输入信号脉宽作为重要因素的设备环境中，根据输入信号的状态系统可能导致异常输入。为了防止例如错误输入，输入滤波功能不接受时间小于用户作为输入信号的时间设置。

输入滤波时间设置为 1ms~100ms。

以下是输入滤波功能的时序图。



(b) 紧急输出功能

运动控制模块的内置输出部分提供紧急输出功能，确定当运行停止导致错误时是否保持输出状态或者清除。

当紧急输出设置为'No'(清除),当运行停止导致运动控制模块停止时输出停止; 通过选择'维持'保持输出状态。

(c) 内置模拟量设置

内置模拟量的更多内容，参考第 13 章内置模拟量功能。

3. 内部参数

(1) 数据记录

关于数据记录更多内容，参考第 11 章数据记录功能。

(2) 编码器

编码器参数如下解释。

项目	描述	设置	初始值
编码器1单元	设置编码器位置显示单元。	0: 脉冲 1: mm 2: inch 3: degree	0: 脉冲
每旋转编码器1脉冲	设置每旋转编码器1脉冲	1 ~ 4294967295	8192 pls
每旋转编码器1行程	设置负载侧编码器 1每旋转的移动距离。	0.000000001 ~ 4294967295	10 pls

第 5 章 内存和参数

项目	描述	设置	初始值
编码器1脉冲输入	设置对应编码器输出外形的输入模式.	0: CW/CCW (x1) 1: 脉冲/DIR (x1) 2: 脉冲/DIR (x2) 3: 相 A/B (x1) 4: 相 A/B (x2) 5: 相 A/B (x4)	3: 相 A/B (x1)
编码器1最大值	设置编码器位置显示范围.	长实数(LREAL)	2147483647 pls
编码器1最小值			-2147483648 pls
编码器1输入滤波值	脉冲输入到编码器的限定频率.	0: No use 1: 500kPPS 2: 200kPPS 3: 100kPPS 4: 10kPPS 5: 1kPPS 6: 0.2kPPS	0: No use
编码器2单元	设置编码器位置的显示单元.	0: 脉冲 1: mm 2: inch 3: degree	0: 脉冲
每旋转编码器2脉冲	设置每旋转编码器2脉冲	1 ~ 4294967295	8192 pls
每旋转编码器2行程	设置负载侧编码器2每旋转的移动距离.	0.000000001 ~ 4294967295	10 pls
编码器2脉冲输入	设置对应编码器输出外形的输入模式.	0: CW/CCW (x1) 1: 脉冲/DIR (x1) 2: 脉冲/DIR (x2) 3: 相 A/B (x1) 4: 相 A/B (x2) 5: 相 A/B (x4)	3: 相 A/B (x1)
编码器2最大值	设置编码器位置显示范围.	长实数(LREAL)	2147483647 pls
编码器2最小值			-2147483648 pls
编码器2输入滤波值	脉冲输入到编码器的限定频率.	0: No use 1: 500kPPS 2: 200kPPS 3: 100kPPS 4: 10kPPS 5: 1kPPS 6: 0.2kPPS	0: No use

(a) 编码器单元

用于设置编码器位置显示单元, 每个控制目标可设置脉冲, mm, inch, 和 degree. 对于同步运行编码器作为一个中心, 单元必须以同步运行中的相同单元进行设置.

备注

当编码器单元与同步运行轴不同, 通过与单元不相关的同步比率运行.

[设置实例]

- 编码器单位: 脉冲
- 编码器分辨率: 4096 脉冲
- 同步运行轴单位: mm
- 主轴:从轴 = 2 : 1

没旋转同步运行轴编码器 1 行程 = $4,096 \times 1 / 2 = 2,048$ [mm]

(b) 每旋转编码器脉冲

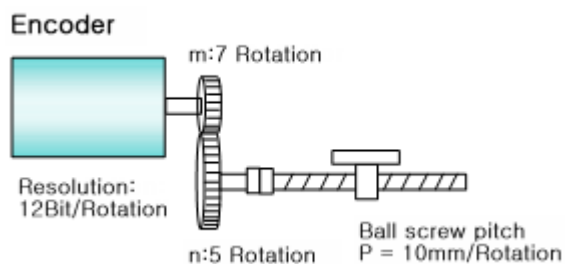
当使用 mm, inch, 和 degree 作为编码器单位时, 设置每编码器旋转的脉冲数量.

(c) 每旋转编码器行程

当使用 mm, inch, 和 degree 作为编码器单位时, 设置每编码器旋转负载侧的移动距离.

[设置实例]

当通过滚珠螺杆移动的设备以齿轮连接编码器时,编码器单元设置/每旋转编码器脉冲/每旋转编码器行程如下.



- 编码器单位: mm
- 每旋转编码器脉冲 = 编码器分辨率 x 编码器侧齿轮比
= 4096×7
= 28672 pls
- 每旋转编码器行程 = 滚珠螺杆螺距 x 设备侧齿轮比
= $10.0 \text{ mm} \times 5$
= 50.0 mm

(d) 编码器脉冲输入

当使用伺服驱动编码器信号或者手动脉冲发生器作为输入时, 可以选择使用从编码器或者手动脉冲发生器输出的正确信号.

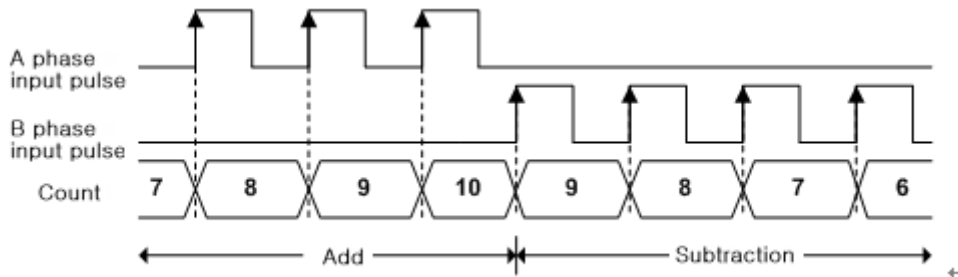
必须选择 CW/CCW (x1), 脉冲/DIR (x1), 脉冲/DIR (x2), 相 A/B (x1), 相 A/B (x2), 和相 A/B (x4) 中的一个并设置编码器输入信号.

第 5 章 内存和参数

1) CW/CCW (x1)

当 A 相输入脉冲增加或者 B 相输入脉冲增加时执行计数运行; 当 B 相输入脉冲低电平时 A 相输入脉冲增加, 则执行加运行; 当 A 相输入脉冲低电平时 B 相输入脉冲增加, 则执行减运行。

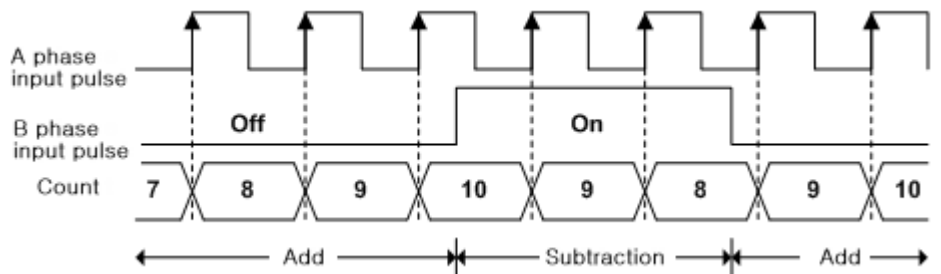
增加/减少	A相输入脉冲高	A相输入脉冲低
B 相输入脉冲高	-	减计数
B 相输入脉冲低	增计数	-



2) 脉冲/DIR (x1)

当 A 相输入脉冲增加时执行运行, 由 B 相确定是否是增加还是减少。

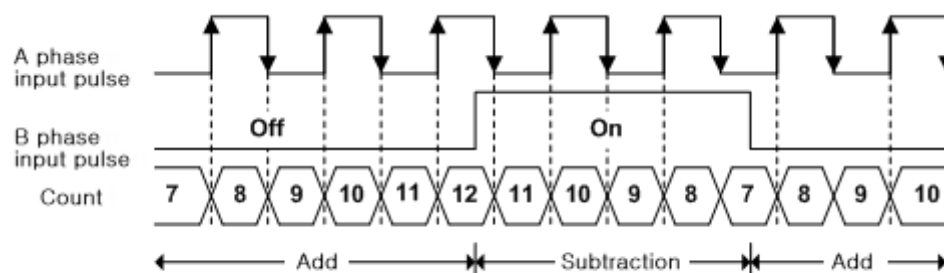
增加/减少	A相输入脉冲高	A相输入脉冲低
B 相输入脉冲 Off	增加计数	-
B 相输入脉冲 On	减少计数	-



3) 脉冲/DIR (x2)

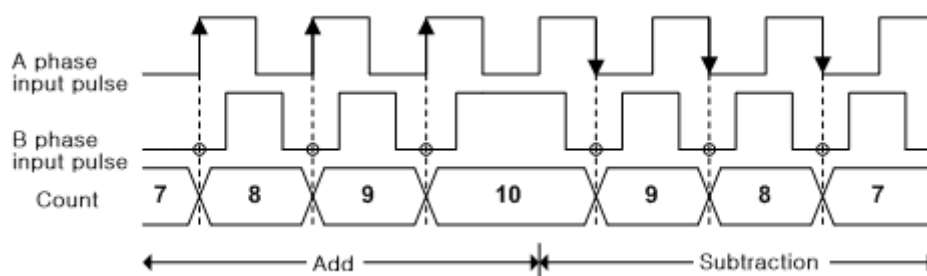
当 A 相输入脉冲增加和减少时执行计数运行, 由 B 相决定是否为增加或者减少。

增加/减少	A相输入脉冲高	A相输入脉冲低
B相输入脉冲Off	增加计数	增加计数
B相输入脉冲On	减少计数	减少计数



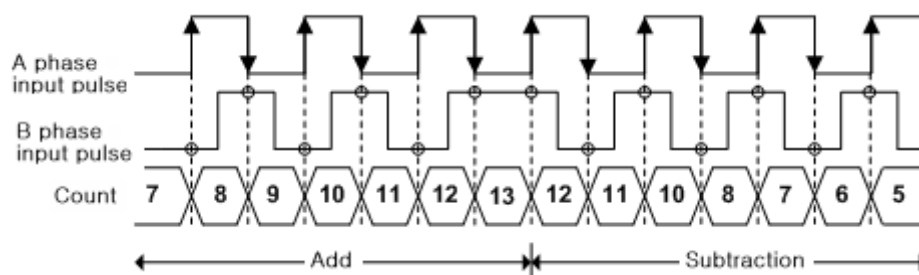
4) 相 A/B (x1)

当 A 相输入脉冲在 B 相输入脉冲之前，并且 A 相脉冲增加时执行增加运行，当 B 相输入脉冲在前面，并且 A 相脉冲减少时执行减少运行。



5) 相 A/B (x2)

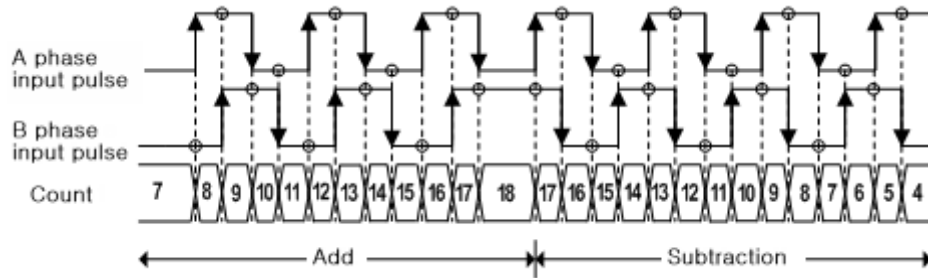
当 A 相输入脉冲增加和减少时执行计数运行。当 A 相输入在 B 相之前执行增加运行，当 B 相输入在 A 相之前执行减少运行。



第 5 章 内存和参数

6) 相 A/B (x4)

对于 A 相输入脉冲增加/减少和 B 相增加/减少的情况下执行计数运行; 当 A 相输入在 B 相之前时执行增加运行; 当 B 相输入在 A 相之前时执行减少运行.

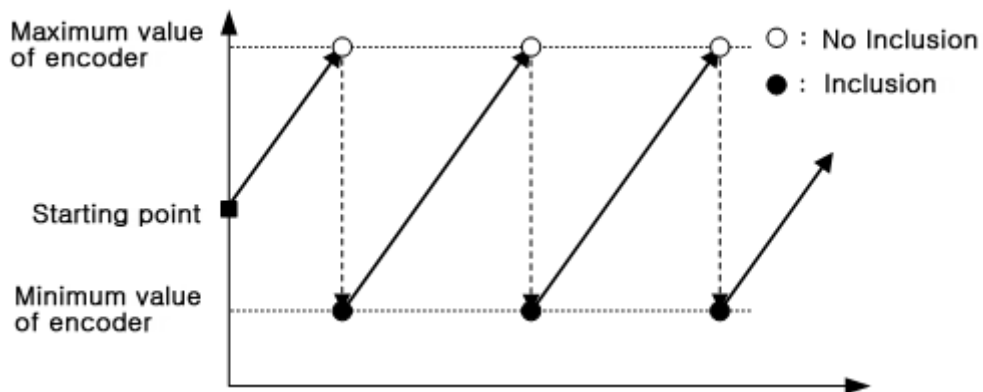


(e) 编码器最大和最小值

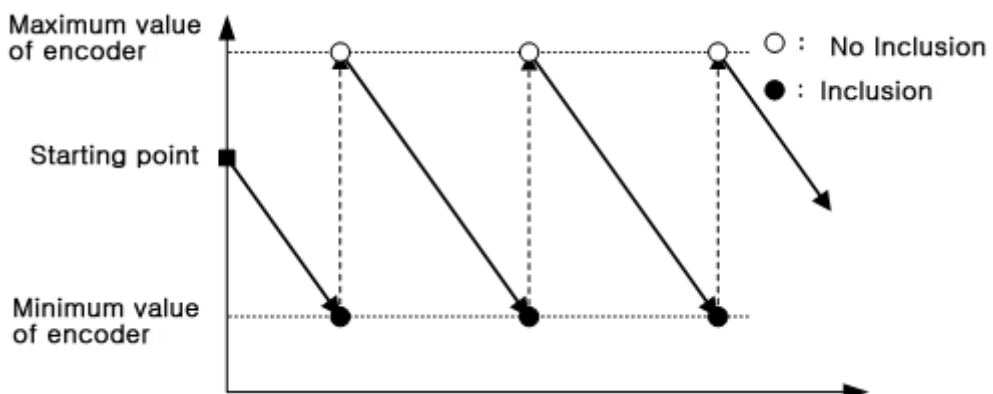
1) 当从伺服驱动或者手动脉冲发生器的编码器信号中计数输入脉冲时, 编码器值范围设置为编码器最大和最小值, 并显示到编码器值.

2) 运行入下图所示.

- 对于编码器值增加的情况



- 对于编码器值减少的情况



4. EtherCAT 参数

描述 EtherCAT 网络设置项目内容. 当修改 EtherCAT 参数时, 确保 EtherCAT 参数写入工程写入菜单.

(1) 主站

当连接到网络时设置 EtherCAT 从站连接相关的主站功能.

主站设置项目如下.

(a) 相关信息

项目	描述	设置范围	初始值
从站版本检查	当连接到网络时指定是否检查实际从站版本值与参数版本信息是否匹配.	0: 不检查 1: 检查	0: 不检查
从站序列号检查	当连接到网络时指定是否检查实际从站序列号与参数序列号信息是否匹配.	0: 不检查 1: 检查	0: 不检查
周期性通讯超时计数	指定发送周期性通讯超时错误次数的基本数量	1~8	2

1) 从站版本查看

当连接到网络时, 通过比较从站参数中设置的版本信息和实际连接从站确定是否连接进程.

根据设置值连接如下.

- '0: 不检查'

不比较从站参数中设置的版本信息与实际连接从站, 执行通讯连接进程.

- '1: Check'

比较从站参数中设置的版本信息与实际从站连接, 如果找到差异, 发生网络配置不匹配错误(错误代码: 0x0F1F), 并且通讯连接进程终止.

当'从站版本查看'设置为'0: 不检查', 如果连接的从站与从站参数版本不匹配, 可能无法正常运行.因此, 使用前检查版本匹配情况.

2) 从站序列号查看

当连接到网络时, 通过从站参数中设置的序列号信息与实际连接从站进行比较确定是否继续连接进程.

第 5 章 内存和参数

根据设置值的运行如下。

- '0: 不检查'

不比较从站参数中设置的序列号信息与实际连接从站，执行通讯连接进程。

- '1: Check'

比较从站参数中设置的序列号信息与实际连接从站，如果发现不匹配，发生网络配置不匹配错误(错误代码: 0x0F1F)，并且通讯连接进程终止。

如果‘从站序列号查看’设置为 1: 查看，当网络配置顺序变更或者从站更换时，可以查看网络变更，这在维护中非常有用，例如从站参数复位，等。需要复位 MP500 中连接到变更网络配置中的序列号。

3) 周期性通讯超时计数

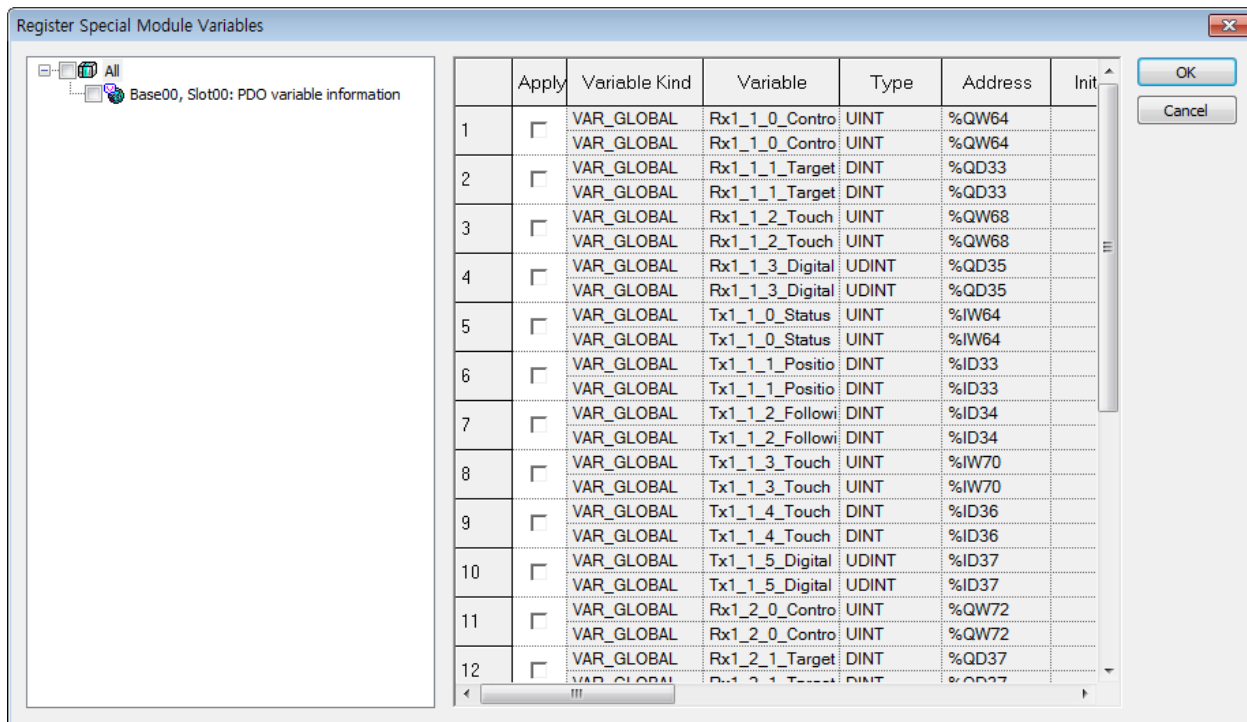
如果在运动控制模块和从站设备间周期性通讯期间未收到周期性数据，指定发生超时错误的基本次数。当在干扰环境(电涌，运动控制模块和从站设备线材间的感应噪声和噪声干扰，等)中通讯超时错误频繁发生，设置更高设置值。可用设置范围在 1 到 8 次之间。

(b) PDO 变量信息

显示注册从站 PDO 数据存储分配信息。

General Info.		PDO Variable					
Station number	Object index	Object Name	Variable	Type	Device	Monitor value	
1	0x1601	1. Rx PDO parameter					
2			Rx1_1_0_Controlword	UINT	%QW64		
3			Rx1_1_1_Target_Position	DINT	%QD33		
4			Rx1_1_2_Touch_Probe_	UINT	%QW68		
5			Rx1_1_3_Digital_Ouputs	UDINT	%QD35		
6	0x1A01	1. Tx PDO parameter					
7			Tx1_1_0_Statusword	UINT	%IW64		
8			Tx1_1_1_Position_Actual	DINT	%ID33		
9			Tx1_1_2_Following_Error	DINT	%ID34		
10			Tx1_1_3_Touch_Probe_S	UINT	%IW70		
11			Tx1_1_4_Touch_Probe_1	DINT	%ID36		
12			Tx1_1_5_Digital_Inputs	UDINT	%ID37		
13	0x1601	1. Rx PDO parameter					
14			Rx1_2_0_Controlword	UINT	%QW72		
15			Rx1_2_1_Target_Position	DINT	%QD37		
16			Rx1_2_2_Touch_Probe_	UINT	%QW76		
17			Rx1_2_3_Digital_Ouputs	UDINT	%QD39		
18	0x1A01	1. Tx PDO parameter					
19			Tx1_2_0_Statusword	UINT	%IW76		
20			Tx1_2_1_Position_Actual	DINT	%ID39		
21			Tx1_2_2_Following_Error	DINT	%ID40		
22			Tx1_2_3_Touch_Probe_S	UINT	%IW82		
23			Tx1_2_4_Touch_Probe_1	DINT	%ID42		
24			Tx1_2_5_Digital_Inputs	UDINT	%ID43		

如果想要注册变量名和在程序中使用，可以通过选择“注册变量”注册在程序中使用的变量。



(1) 从站

(a) 一般信息

查看用于网络连接的 EtherCAT 从站信息，可用于在执行连接的每个从站打开到 MP500 工程树中[EtherCAT 参数]-[从站]从树后显示从站信息。为了增加从站(伺服驱动, EtherCAT I/O,等), EtherCAT 参数需要写入工程中。

从站一般信息项目如下

项目	描述	设置范围	初始值
从站名	选择从站并显示所选从站的名称	XML	-
站号	显示应用于所选从站的站号。	-	1(当增加从站时自动增加)
供应商	所选从站供应商名称自动显示。	不可配置	-
版本	所选从站版本自动显示。	不可配置	-
序列号	显示所选从站序列号。	不可配置	0x1600 PDO 映射信息
是否使用 DC	设置是否使用从站的 DC。	0: 不使用 1: 使用	1: 使用
连接中的替换功能	设置是否在 EtherCAT 通讯期间进行从站替换。	0: 不使用 1: 使用	0: 不使用

第 5 章 内存和参数

1) 从站名

选择连接到运动控制模块的从站并显示所选从站的名称。当增加从站到从站数据时选择 **JSG2S** 伺服驱动为初始值。
当选择从站时，在可用列表下的文件夹中重新获得从站信息。

→ \MP500 初始文件夹中的 **EtherCATXML** 文件夹'

如果新增加一个从站，复制相应 XML 文件到以上文件夹，然后重启 **MP500** 或执行通过'ESI 库' 窗口中右击激活的'ESI 重新扫描' 菜单。

2) 站号

显示适用于所选从站的站号。显示范围从 **1** 到 **64**，用户无法任意修改。为了变更从站站号，选择工程树种的站并在菜单中右击，执行'属性'菜单，然后变更从站信息中的站号。

然而，当自动连接从站时根据连接顺序自动设置站号。

3) 供应商

自动显示所选从站供应商名称。用户无法任意变更。

4) 版本

自动显示所选从站版本信息。用户无法任意变更。

5) 序列号

显示所选从站序列号。当 **EtherCAT** 通讯期间执行"读取序列号"时，显示当前产品序列号。

6) 是否使用 DC

如果从站支持 **DC** 功能，从 XML 文件中自动设置。如果不使用 **DC** 功能，选择'0: 不使用'。

备注

DC(分布式时钟): 用于同步 **EtherCAT** 主站与 **EtherCAT** 从站，使能 **EtherCAT** 从站间的高精度同步控制。

DC 同步 **EtherCAT** 主站和 **EtherCAT** 从站间的时间信息以同步每个从站，连接运动控制模块的第一个从站提供参考时钟。参考时钟以每个通讯周期分布时间信息到每个从站。

7) 连接期间的替换功能

如果从站停止导致网络断开或者异常运行，使用线缆冗余功能从错误状态恢复并连接到网络，该功能会检查连接状态。然后，提供不连接到整个网络的各个从站的单独连接。

更多内容，参考 **8.3.6** '连接期间的替换'。

(b) PDO 设置

RxPDO 设置以每个通讯周期从运动控制器到从站传输的同步数据。当选择从站时 RxPDO 项目支持自动设置相应从站。可以使用‘编辑’功能增加或者删除对象。

备注

对于用于运动轴的从站，当编辑 RxPDO 对象时，必须包含以下用于运动控制模块的基本项目。

0x6040:0	控制字
0x607A:0	目标位置

在此分配的同步数据自动分配到 I/O 设备，可以以 I/O 变量进行分配，并在用户程序中参考。例如，连接到从站 JSDG2S 伺服驱动 RxPDO 同步数据‘控制字’对象注册为 I/O 标志位 Rx1_1_0_控制字 (%QW64)。

TxPDO 设置以每个通讯周期从运动控制器从站读取的同步数据。当选择从站时，相关从站支持的 TxPDO 项目自动设置。可以使用‘编辑’功能增加或者删除对象。

当编辑 PDO 对象时，以下对象作为运动控制模块中基本项目必须包含。

备注

对于用于运动轴的从站，当编辑 TxPDO 对象时，必须包含以下用于运动控制模块的基本项目。

0x6041:0	状态字
0x6064:0	实际位置

在此分配的同步数据自动分配到 I/O 设备，可以以 I/O 变量进行分配，并在用户程序中参考。例如，连接到从站 2 的 JSDG2S 伺服驱动 TxPDO 同步数据‘状态字’对象注册为 I/O 标志位 Tx_1_2_0_状态字(%IW68)。

(c) SDO 参数

- 设置从站中运行的 SDO (服务数据对象)参数。
- 参数不保存在运动控制器中，但是在从站中运行。
- 对于参数的设置和运行，参考附录 3 设置实例。

第 5 章 内存和参数

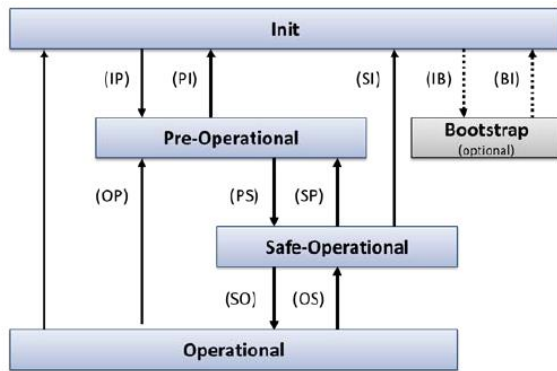
(d) '启动命令

- 该功能为 EtherCAT 连接运行的从站转换期间设置特定对象.
- 用于与从站 Rx 和 TxPDO 地址分配相同的从站参数初始化, 以及项目设置
- 每个从站最多提供 50 个.

项目	描述	设置范围	初始值
转换	在对象设置功能操作中设置转换进程.	IP, PS, SO, SP, OP, OS	无
索引	设置对象索引和分索引	XML	-
数据	设置对象配置的数据.	根据数据类型的变量	-
声明	增加想要设置对象的声明.	-	-
标志位	显示相关'启动'命令的标志位.	固定	-

备注

转换的配置按照以下 EtherCAT 状态转换图.



(e) 在线服务

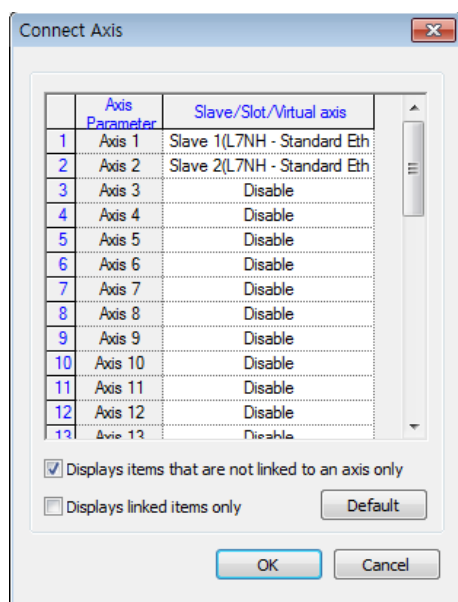
- 关于更多在线服务的内容, 参考第 08 章运动控制功能 -8.5.FoE 功能.

5. 轴参数

(1) 轴/从站连接

有两种类型可以通过运行控制器控制的轴; 实轴和虚轴. 实轴是指分配给实际 EtherCAT 从站的轴, 虚轴为运动控制器中任务生成和控制. 注册为 EtherCAT 从站的从站可以分配为可通过运动控制器控制的轴.

设置轴如下, [轴参数] - [增加项目] - [轴], 或者[轴参数] - [轴/从站连接].



轴可设置为“从站”, “虚轴”, “禁止”. 设置为禁止的轴不包含在轴参数中.

(2) 轴参数

(a) 基本设置

基本设置中的基本参数如下.

项目	描述	设置	初始值
单位	设置轴指令位置单位.	0: 脉冲 1: mm 2: inch 3: degree	0: 脉冲
每旋转脉冲	设置对应编码器分辨率的电机每旋转脉冲数量.	1 ~ 4294967295	524288 pls
每旋转行程	设置电机负载侧每旋转移动数据.	0.000000001 ~ 4294967295	10 pls
速度指令单位	设置轴指令速度单位.	0: Unit/Time 1: rpm	0: Unit/Time
速度限制	设置每轴速度指令的最大速度.	长实数(LREAL) 正数	2000000 pls/s
紧急减速停止	设置用于紧急停止条件的减速.	0或长实数(LREAL) 正数	0 pls/s ²
编码器选择	设置使用的编码器类型.	0: 增量编码器 1: 绝对编码器	0: 增量编码器
齿轮比(电机)	设置电机和负载间的齿轮比.	1~65535	1
齿轮比(机械)		1~65535	1
反转运行模式	在以最新执行指令作为输入条件的反向运行情况下, 指定运行方式.	0: 紧急停止 1: 停止	0: 紧急停止

第 5 章 内存和参数

1) 单位

用于设置在运动控制期间的指令单位, 根据控制目标, 脉冲单位, 每轴可设置为 **mm**, **inch**, 和 **degree**.

当变更单位设置时, 其他参数或者变量值不变更. 因此, 当变更单位时, 相关参数必须复位, 以便可以调整相关单位的设置范围.

2) 每旋转脉冲

当使用 **mm**, **inch**, 和 **degree** 作为运动控制指令单位时, 并且以 **rpm** 表示速度, 电机每旋转需要的脉冲数量设置为使用.

3) 每旋转行程

当使用 **mm**, **inch** 和 **degree** 作为运动控制指令单位时, 设置负载侧电机每旋转移动数量.

电机旋转的机械如何移动取决于机械结构.

4) 速度指令单位

设置用于运动控制指令的速度值基本单位.

如果设置为 '0: unit/time', 应用于「单位」参数中设置的相关单位位置每秒变化率. 例如, 如果以 **mm** 设置「单位」, 速度指令单位为 'mm/s'.

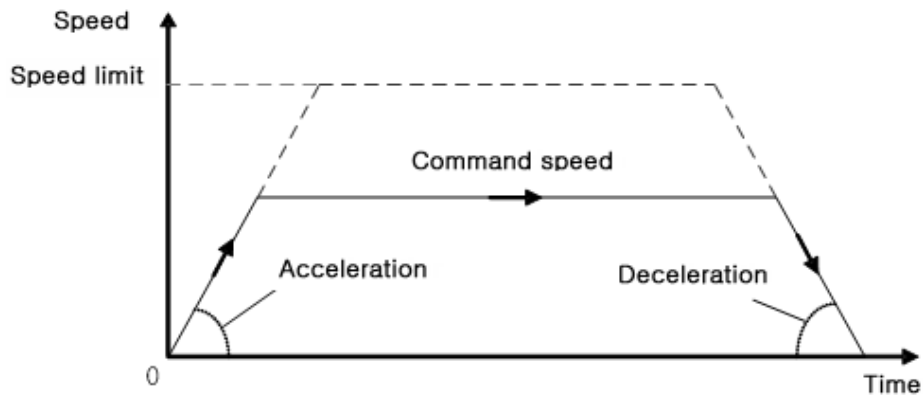
如果设置为 '1: rpm', **rpm** 应用于速度指令单位. 如果速度指令单位为 **rpm** 并且为单位速度的内部变化, 使用在「每旋转脉冲」和「每旋转行程」参数中设置的值.

当变更速度指令单位设置时, 其他参数或者变量值不变更. 因此, 根据相关单位设置范围, 相关参数必须复位.

5) 速度限制

参考运动控制运行可用设置最大速度的速度限制.

当运行相关轴时, 运行速度应按照如下速度限制进行设置.



6) 紧急减速停止

突然停止状况下的减速在由于内部或者外部因素导致的运行突然停止状况下设置.

紧急停止条件如下.

- 检测软件上限/下限
- 服务轴运行速度超出同步运行(齿轮, cam)的速度限制
- 设置「跟踪误差错误等级」为‘1: 发生跟踪误差‘警报’和错误
- 在 XG-PM 测试运行期间执行紧急停止指令
- 当执行条件(重启指令或者持续更新激活状况除外)查看期间当前轴运行时执行指令发生错误的情况.

7) 编码器选择

设置将要使用的编码器类型. 当使用绝对位置系统时, 选择 1: 绝对编码器.

以下显示“编码器选择”设置

项目	设置	描述
编码器选择	0: 增量编码器	电源on/off之后, 伺服电机不维持在前一个位置. 电源on/off之后, 起点固定状态为off.
	1: 绝对编码器	激活绝对位置系统. 电源on/off之后, 伺服电机维持在前一个位置. 电源on/off之前起点固定状态维持最后条件.

8) 齿轮比(电机, 机械)

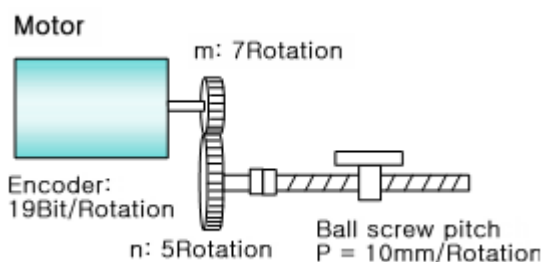
设置电机和负载间的齿轮比. 如果是当电机侧旋转 m 次时负载侧旋转 n 次的结构, 设置齿轮比如下.

- 电机侧齿轮比 = m
- 机械侧齿轮比 = n

如果「单元」设置为0: 脉冲, 该参数无效.

[设置实例]

当通过滚珠螺杆移动的机械通过齿轮连接到编码器, 编码器单位/每旋转脉冲/每旋转行程的设置如下.



- 单位: mm
- 每旋转脉冲 = 524288 (19Bit 编码器)
- 每旋转行程 = 滚珠螺杆螺距 = 10.0 mm
- 齿轮比(电机) = 7
- 齿轮比(机械) = 5

第 5 章 内存和参数

备注

如果在以上[设置实例]中[单位]设置为'0: 脉冲', 将移动到编码器脉冲数量对应的位置, 不考虑电机侧齿轮比或者机械侧齿轮比。
也就是说, $524,288 * 7/5 = 734,003$ 脉冲指令以 10mm 移动顺序发生。

9) 反转运行模式

当以最新执行指令作为输入条件反向运行时指定运行方式。

当启动或者重启缓冲区模式指令异常终止, 或者激活持续更新时, 指令条件和当前运行方向彼此相反, 通过在参数中设置的如下方式停止, 并以设置速度启动运行。

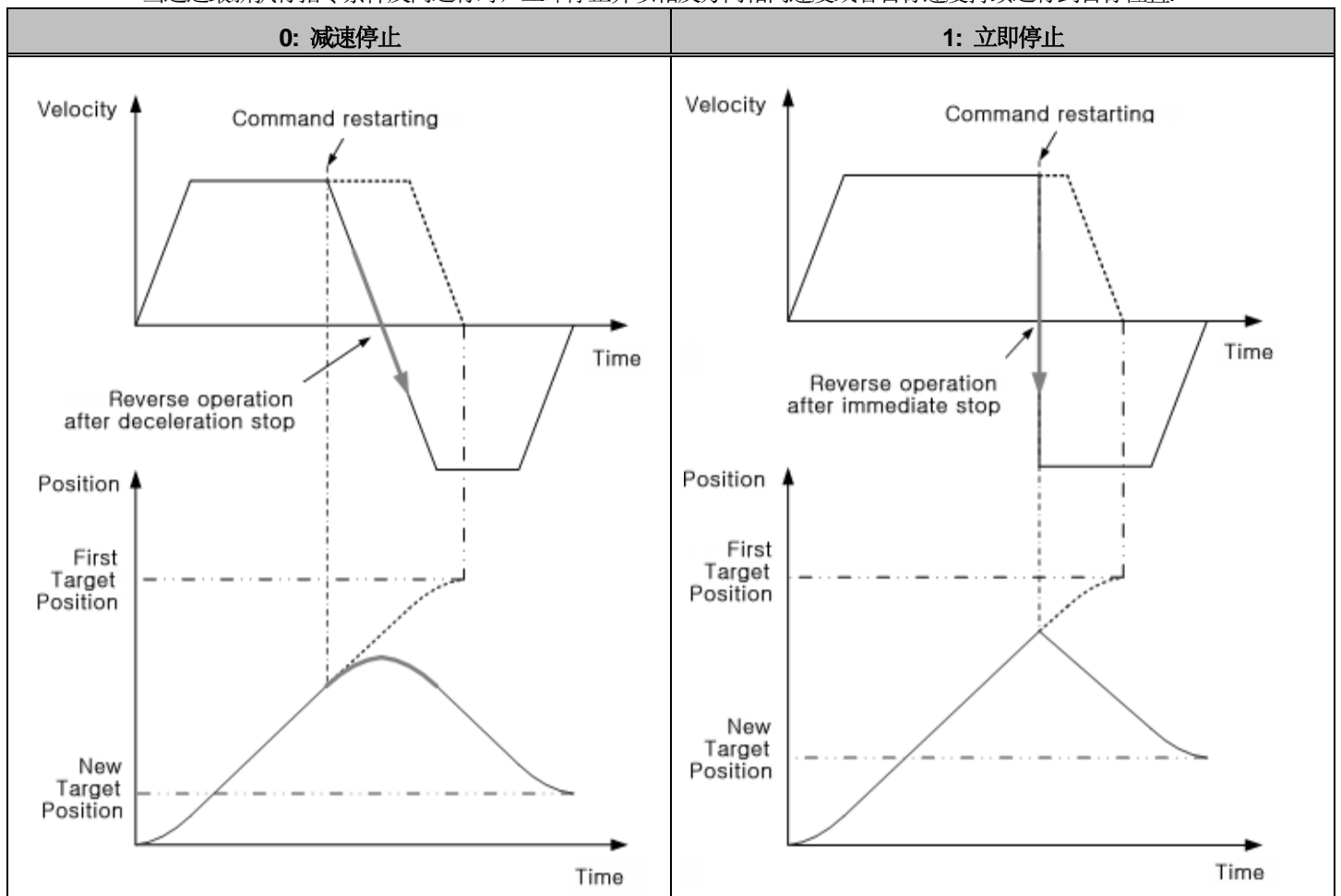
如果缓冲区模式未异常停止, 以缓冲区模式中指定持续运行方式运行, 而不是以参数中设置的方式运行。

- '0: 减速停止'

当通过最新执行指令条件反向运行时, 减速暂停到 0 速度, 持续加速到目标位置或者以目标速度运行。

- '1: 立即停止'

当通过最新执行指令条件反向运行时, 立即停止并以相反方向相同速度或者目标速度持续运行到目标位置。



(b) 扩展参数

以下解释运行参数的扩展参数

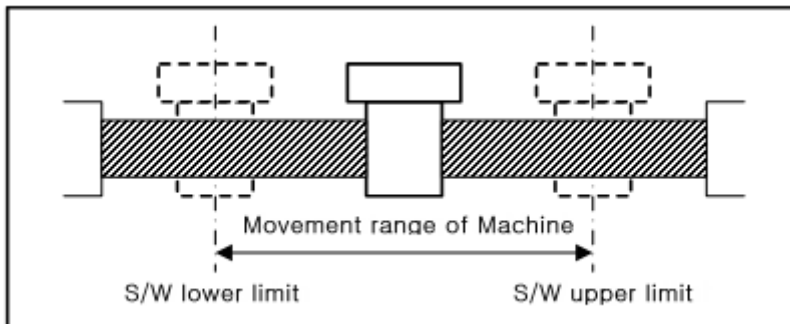
项目	描述	设置	初始值
S/W 上限	设置软件限制范围功能.	长实数(LREAL)	2147483647 pls
S/W 下限			-2147483648 pls
无限运行重复位置	使用无限运行重复模式时, 设置重复位置范围值.	长实数(LREAL) 正数	360 pls
无限运行重复	设置无限长度重复运行功能允许状态.	0: 禁止 1: 使能	0: 禁止
指令位置范围	设置完成定位之前到达位置信号为 On 的范围.	0 或者长实数(LREAL) 正数	0 pls
跟踪错误超出范围值	设置检测大于位置偏差的值.	0 或者长实数(LREAL) 正数	0
跟踪错误等级	设置大于偏差的错误等级.	0: 警告 1: 警报	0: 警告
当前位置补偿数	设置补偿阈值以表示当前位置值作为目标位置值.	0 or 长实数(LREAL) 正数	0
当前速度滤波时间常数	设置计算当前速度移动平均的时间.	0 ~ 100	0
错误复位监控时间	当伺服驱动中发生复位错误时设置监控时间.	1 ~ 1000	100
速度控制期间的软限制	设置速度控制期间是否检测软限制.	0: 不检测 1: 检测	0: 不检测
覆盖模式	当覆盖指令执行时, 设置应用输入值的方法.	0: 比率 1: 单位值	0: 比率
JOG 高速	设置速度/加速/减速/加加速度值, 参考 JOG 运行指令	长实数(LREAL) 正数	100000 pls/s
JOG 低速			10000 pls/s
JOG 加速		0 或者长实数(LREAL) 正数	100000 pls/s ²
JOG 减速			100000 pls/s ²
JOG 加加速度			0 pls/s ³

第 5 章 内存和参数

1) 软件上限/软件下限

该功能以上限&下限的形式设置机器移动可用范围，并且在超出设置范围后机器无法运行。换句话说，用于预防分离导致设置运行位置错误和由于用户程序错误发生的错误运行。

设置软件上限和软件下限范围以外的外部输入上限和下限。



在运行一开始和运行期间进行软件上限和下限的范围检查。

如果检测软件上限和下限，发生错误并且模块突然停止电机。因此，当重启运行时查看错误原因并在复位后使用。

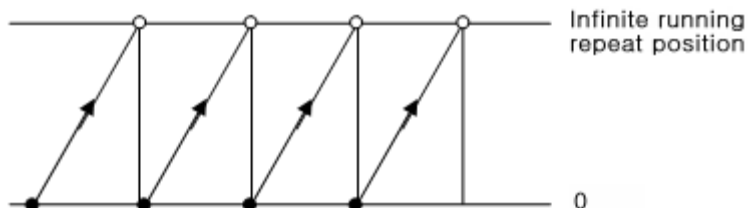
如果设置软件上限和下限为初始值(上限: 2,147,483,647, 下限: -2,147,483,648)或者相同值时，则不检测软件上限和下限。

2) 无限运行重复位置

当使用无限运行重复模式时，设置重复的位置值。

当设置扩展参数时应用，「无限运行重复」参数,为 '1:使能'。

当「无限运行重复」参数为'1:使能'，指令位置和当前位置表示为"0~ (-1 的无限运行重复位置)。 (「Unit」= 0: 脉冲基准)



3) 无限运行重复

设置无限运行重复操作的可用功能。

如果参数设置为'1: 使能'，指令位置和当前位置显示定期更新并在无限长度重复位置范围内自动设置。

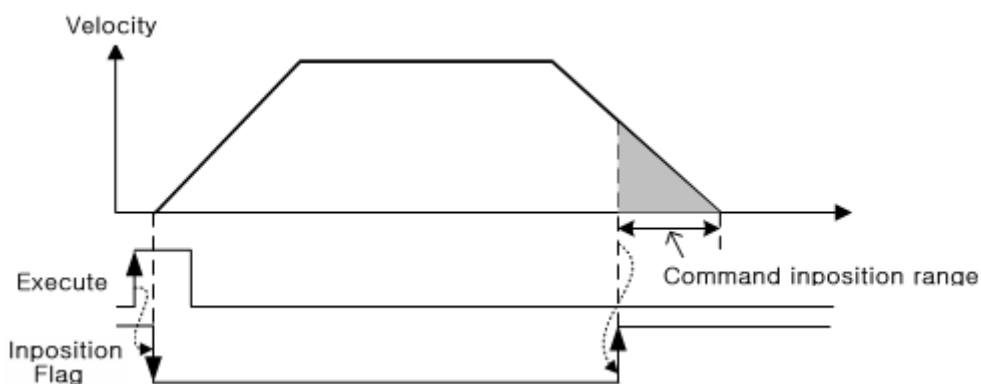
当不使用无限运行重复功能时，必须设置为'0: 禁止'。

4) 指令到达位置范围

该项目设置到到达位置标志位 ($_AXxx_INPOS$)为 On 的目标位置的距离。

当启动运动控制时，到达位置标志位($_AXxx_INPOS$)为 Off，当当前位置从目标位置进入「指令到达位置范围」时变为 On。

当在完成位置控制之前执行其他辅助工作时，到达位置标志位可以作为一个触发。



5) 跟踪误差超出值

设置用于检测超出位置偏差值的值.如果值超出范围,「超出偏差警告(_AXxx_DEV_WARN)」或者「超出偏差警报(_AXxx_DEV_ERR)」标志位为 On.

如果设置值为 0, 不检测超出偏差值. 可以在扩展参数的「跟踪误差错误等级」中设置是否希望警告或者警报.

6) 跟踪误差水平

当值检测超出偏差时设置是否警告或者警报.

根据设置值运行如下.

- '0: 警告'

当跟踪误差中发生错误,「超出偏差警告 (_AXxx_DEV_WARN)」标志位为 On, 发生跟踪误差警告错误(错误代码: 0x101D). 轴不停止并保持运行.

- '1: 警报'

当跟踪误差中发生错误,「超出偏差警报 (_AXxx_DEV_ERR)」标志位为 On, 发生跟踪误差警报错误(错误代码: 0x101C). 以基本参数的「紧急停止减速」轴突然停止.

在以下状况中, 不检测跟踪误差的错误.

- 「跟踪误差超出-范围值」为 0
- 原点返回运行或者转矩控制

7) 当前位置补偿数量

当前位置补偿数量是作为指令位置用于显示当前位置值的参数单位, 当伺服电机当前位置值不显示为固定值, 而根据用户应用和伺服驱动的个人设置略有变更.

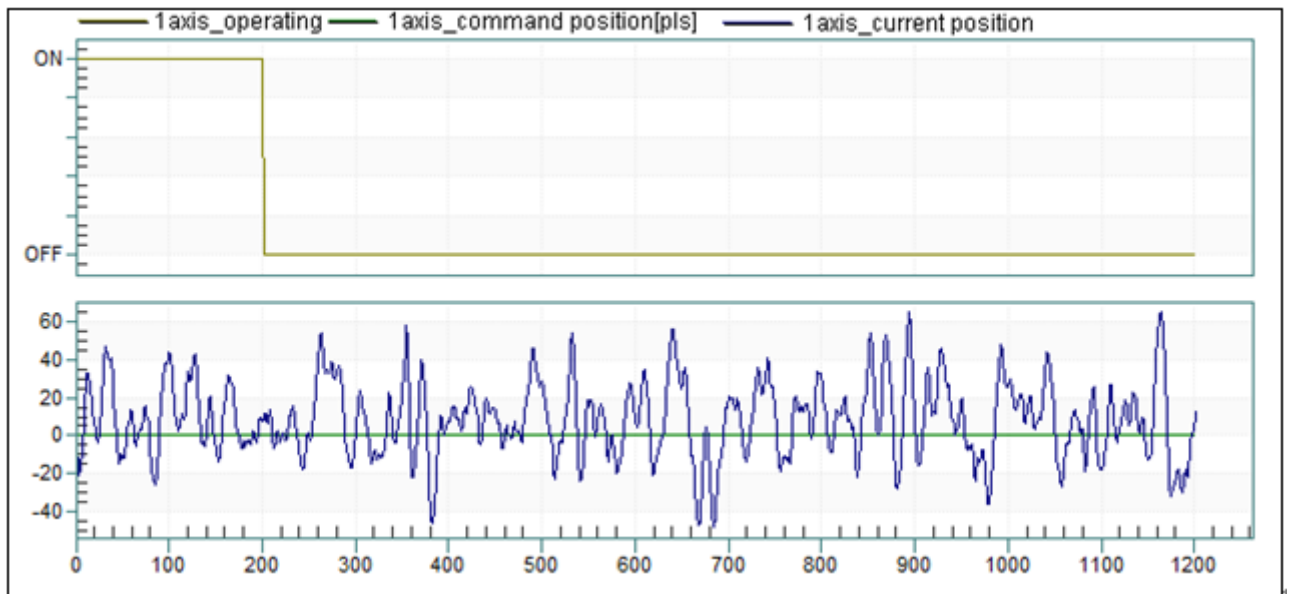
第 5 章 内存和参数

当不在运行状态中，指令位置和当前位置在显示当前位置补偿数量范围内存在不同时，当前位置值显示为指令位置值。当运行时，不体现当前位置补偿数量，显示实际位置值。

以下示例为当指令位置为0时根据当前位置补偿数量值应用当前位置补偿数量。

① 当前位置补偿数量 = 0 pls

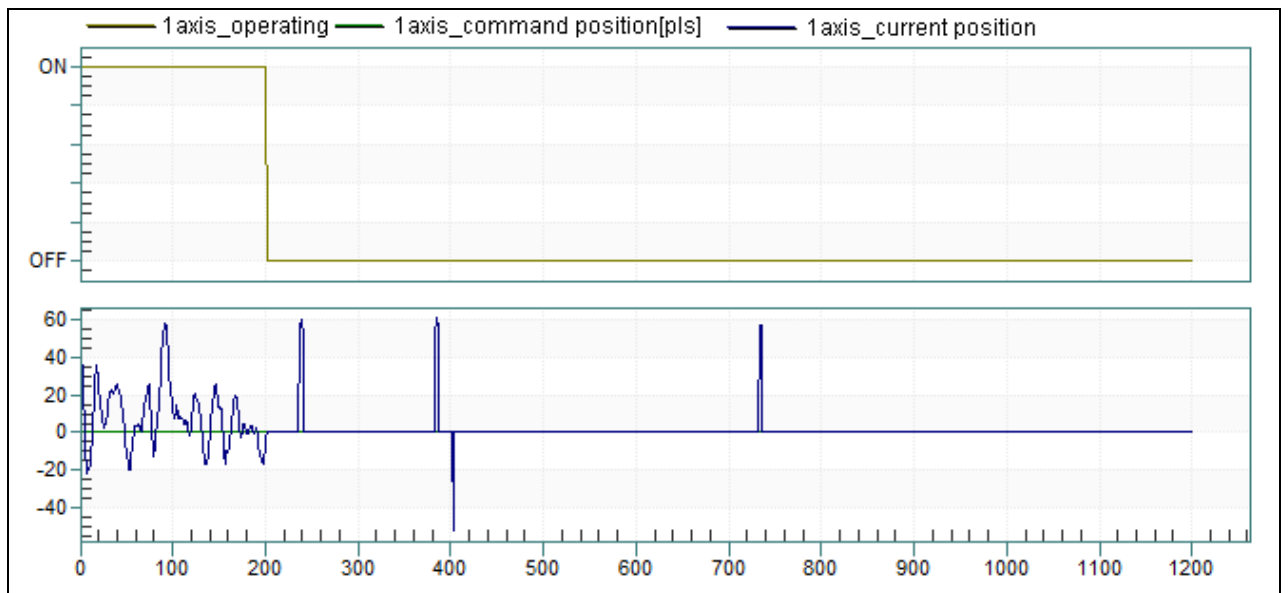
甚至在结束运行后，实际电机位置值显示为显示为当前位置值。



② 当

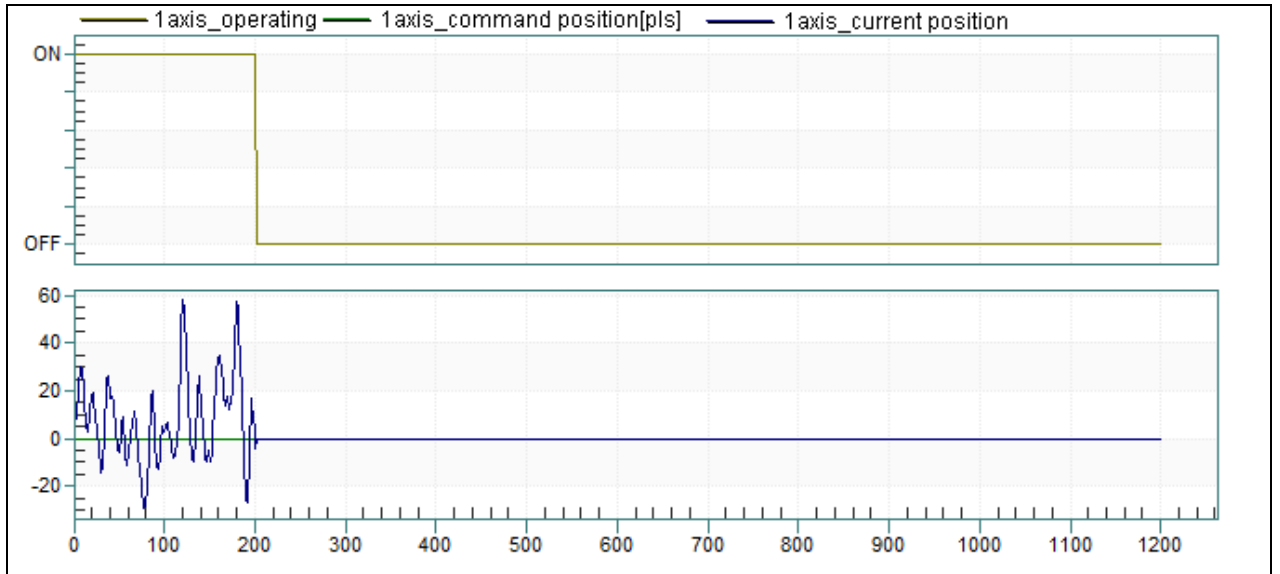
前位置补偿数量= 50 pls

结束运行后，如果当前位置值在指令位置±50 范围内，显示为指令位置值。



③当前位置补偿数量= 100 pls

结束运行后，如果当前位置值在指令位置±100 范围内，显示为指令位置值.



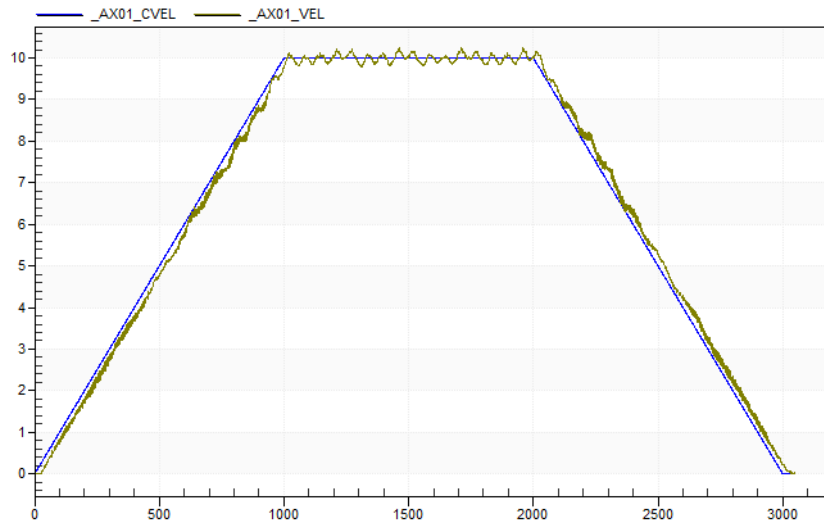
8) 当前速度滤波时间常数

设置计算当前速度移动平均的时间.(单位: ms) 如果设置为'0', 不应用当前速度滤波时间常数.

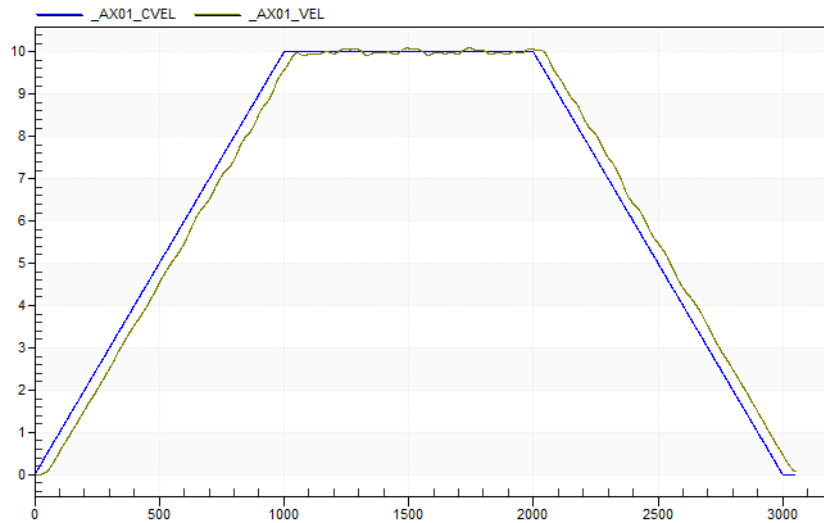
当轴速度放缓或者以当前速度有较大变化时, (例. 「unit」设置为'0: 脉冲', 通过应用移动到当前速度的平均实现稳定速度.

可以在以下列表中查看根据当前速度滤波时间常数值在当前速度差异, 下表为跟踪指令速度和以指令速度 10 mm/s 的当前速度.

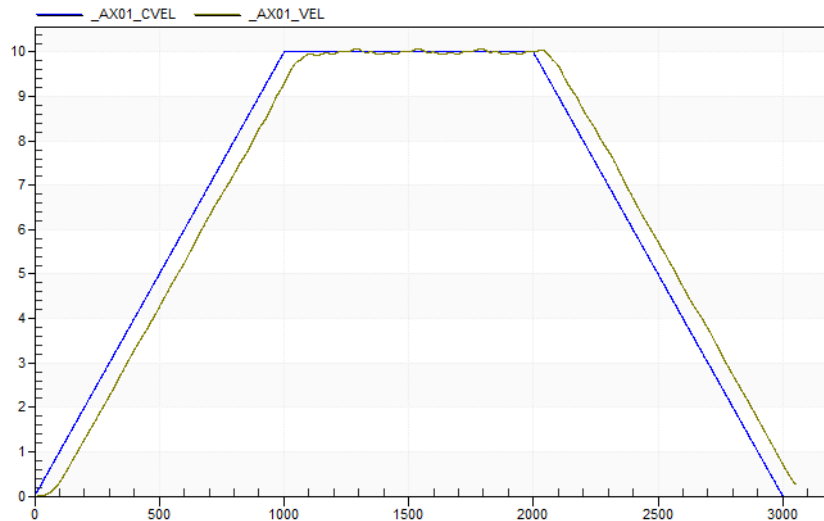
①当前速度滤波时间常数= 0 ms



② 当前速度滤波时间常数= 50 ms



③ 当前速度滤波时间常数= 100 ms



9) 错误复位监控时间

在伺服驱动错误复位时间发生中设置监控时间. (单位: ms) 如果伺服驱动中在错误复位监控时间内发生错误, 错误复位监控终止并且发生伺服驱动错误复位时间超出错误(错误代码: 0x1070).

10) 速度控制期间的 SW 限制

当通过速度控制的固定速度运行期间检测软件限制时, 用于停止电机. 根据如下设置值运行.

- '0': 不检测

如果在速度控制下, 即使当软件限制功能激活, 不检测软件限制.

- '1': 检测

如果在速度控制下, 即使当软件限制功能激活, 检测软件限制.

即使当参数值设置为1: 检测, 如果软件上限/下限设置为初始值(上限: 2,147,483,647, 下限: -2,147,483,648) 或者相同值, 软件限制不检测.

11) JOG 高速 /JOG 低速

当执行手动运行类型的 JOG 运行时，JOG 速度与速度相关。JOG 运行被分为 JOG 高速和 JOG 低速。

JOG 相对于与加速区域，固定速度和减速运行。因此，加速区域通过 JOG 加速时间控制，减速区域通过减速区域控制。

JOG 高速设置范围不可以超出速度限制。同样，JOG 高速必须相同或者大于 JOG 低速。

12) JOG 加速, JOG 减速, JOG 加速度

设置应用于 JOG 高速和 JOG 低速运行的加速值，减速值和加速度值。

如果 JOG 加速为 0, JOG 运行初期 JOG 设置速度超出加速区域时立即运行。

如果 JOG 减速为 0, JOG 运行停止 0 超出减速区域时立即停止。

如果 JOG 加速度为 0, 加速/减速形式为作为固定加速的线性形式

(c) NC 参数

描述轴参数的 NC 参数。

项目	描述	设置范围	初始值
达到主轴旋转命令速度的识别范围	确定主轴指令速度是否通过设置值达到。	0~100%	95%
达到主轴旋转速度为 0 的识别 RPM	确定主轴指令 0 速度是否通过设置值达到。	0~100rpm	5rpm

第 5 章 内存和参数

6. 轴组参数

(1) 基本设置

基本设置项目如下解释.

项目	描述	设置	初始值
配置轴1~10	设置轴组形式的轴.	无, 1轴 ~ 32轴(实/虚轴), 33轴 ~ 36轴(虚轴)	无
插值速度最大	设置轴组的最大运行速度.	长实数(LREAL) 正数	20000000 u/s

(a) 配置轴设置

设置属于相应轴组的每轴数量. 每个轴组可最多包含 10 个轴.

虚轴同样可以在轴组参数中设置.

轴设置必须按顺序在轴组中设置, 该轴组执行圆弧插值或者螺旋插值指令. 也就是说, 「轴设置 1」是圆弧 X-轴, 「轴设置 2」是圆弧 Y-轴, 「轴设置 3」为螺旋插值 Z-轴.

因此, 当设置轴组时执行圆弧插值指令, 发生如下错误.

- 轴组由 4 轴组成(错误代码: 0x20A9)
- 「axis 设置 1」或者「axis 设置 2」设置值为无 (错误代码: 0x20AA)
- 「axis 设置 3」设置值为无并设置余下轴(错误代码: 0x20AA)

(b) 插值速度最大

引用当通过属于相应轴组的轴控制插值时插值控制运行的配置最大速度.

对于相应轴组的插值运行, 插值速度必须设置为小于插值最大速度.

(2) 坐标系设置

坐标系设置项目如下.

项目	描述	设置	初始值
坐标系	设置应用于坐标系运行的机器人类型.	0: 无, 1: XYZ 2: Delta3 3: Delta3R 4: 线性Delta3 5: 线性Delta3R	0: 无
坐标系参数	根据坐标系类型设置机械参数.	-	-

(a) XYZ

XYZ 是在 X-轴的「轴设置 1」，Y-轴的「轴设置 2」，Z-轴的「轴设置 3」设置轴的机器人，确保一一对应，并以笛卡尔坐标移动。如果坐标系类型设置为 XYZ，不需要设置做表情参数。

(b) Delta3/3R

Delta 是由 3 个旋转轴组成的 delta 机器人。如果坐标系类型设置为 Delta，需要设置 4 个坐标系参数; Rf / Rm / Lf / Lm

	参数	描述
	固定帧半径(Rf)	从固定帧中心到固定帧连杆的长度(mm)
	固定帧连杆长度(Lf)	固定帧连杆长度(mm)
	移动帧连杆长度(Lm)	移动帧连杆长度(mm)
	移动帧半径(Rm)	从移动帧中心到移动帧连杆的长度(mm) (在左图中，固定帧的 X, Y 坐标和移动帧相同)

(c) 线性 Delta3/3R

线性 Delta 是由 3 个线性轴组成的 delta 机器人。如果坐标系类型设置为线性 Delta，需要设置 5 个坐标系参数; Lm / Hf / RfTop / RfBottom / Rm.

	参数	描述
	移动帧连杆长度(Lm)	移动帧连杆长度(mm)
	固定帧高度(Hf)	固定帧高度(mm)
	固定帧半径(RfTop)	固定帧半径(mm)
	固定帧半径(RfBottom)	固定帧半径(mm)
	移动帧半径(Rm)	从移动帧中心到移动帧连杆的长度(mm) (在左图中，固定帧的 X, Y 坐标和移动帧相同)

第 5 章 内存和参数

(3) 工具设置

工具设置项目如下.

项目	描述	设置	初始值
X轴补偿	在机器人结尾(工具)设置X轴补偿	长实数(LREAL)	0
Y轴补偿	在机器人结尾(工具)设置Y轴补偿	长实数(LREAL)	0
Z轴补偿	在机器人结尾(工具)设置Z轴补偿	长实数(LREAL)	0

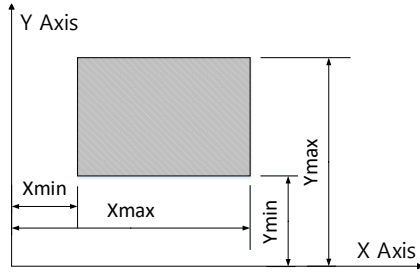
当在机器人结尾使用单独工具时, 使能工具位置的工具设置参数设置为补偿, 在工具结尾可以控制.

(4) 工作区设置

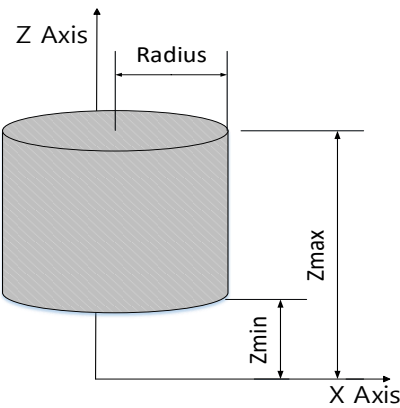
工作区设置项目如下.

项目	描述	设置	初始值
工作区类型	设置工作区类型.	0: 不使用 1: 矩形 2: 圆柱体 3: Delta 4: 扇形	0
工作区错误查看	设置如果从工作区分离, 是否错误发生	0: 禁止 1: 允许	0
工作区参数	根据工作区类型设置参数.	长实数(LREAL)	0

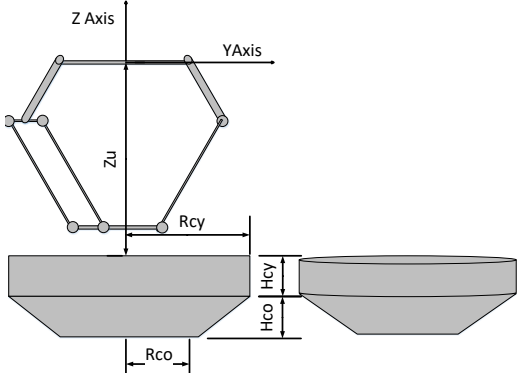
(a) 矩形

	参数	描述
	工作区参数 1	X 最小(mm)
	工作区参数 2	X 最大(mm)
	工作区参数 3	Y 最小(mm)
	工作区参数 4	Y 最大(mm)
	工作区参数 5	Z 最小(mm)
	工作区参数 6	Z 最大(mm)

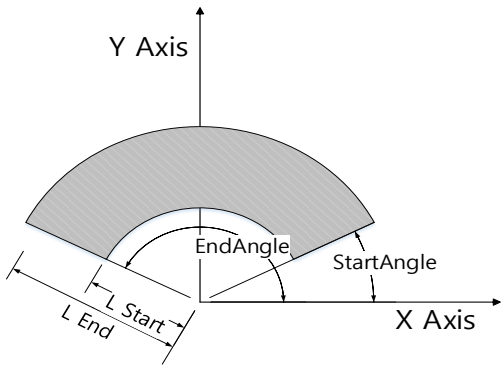
(b) 圆柱体

	参数	描述
	工作区参数 1	半径(mm)
	工作区参数 2	Z 最小(mm)
	工作区参数 3	Z 最大(mm)

(c) Delta

	参数	描述
	工作区参数 1	Zu(mm)
	工作区参数 2	Hcy(mm)
	工作区参数 3	Hco(mm)
	工作区参数 4	Rcy(mm)
	工作区参数 5	Rco(mm)

(d) 扇形

	参数	描述
	工作区参数 1	L 启动(mm)
	工作区参数 2	L 结束(mm)
	工作区参数 3	Z 最小(mm)
	工作区参数 4	Z 最大(mm)
	工作区参数 5	SartAngle(degree)
	工作区参数 6	EndAngle(degree)

第 5 章 内存和参数

(5) PCS 设置

PCS 设置项目如下.

PCS 参数设置工件起始到 PCS 促进在坐标系中移动工件的运行. 在 PCS 坐标系运行中, 根据作为起始的设置 PCS 执行坐标系运行.

项目	描述	设置	初始值
X-轴移动	设置从MCS 起始到PCS起始的X-轴移动距离	长实数(LREAL)	0 mm
Y-轴移动	设置从MCS起始到PCS起始的Y-轴移动距离	长实数(LREAL)	0 mm
Z-轴移动	设置从MCS 起始到PCS起始的Z-轴移动距离	长实数(LREAL)	0 mm
X-轴旋转	设置PCS X-轴旋转值.	-360~360	0 deg
Y-轴旋转	设置PCS Y-轴旋转值.	-360~360	0 deg
Z-轴旋转	设置PCS Z-轴旋转值.	-360~360	0 deg

(6) 坐标系 JOG 运行设置

JOG 运行设置项目如下.

坐标系 JOG 速度参数设置 JOG 运行期间的速度.

项目	描述	设置范围	初始值
XYZ 低速	设置在坐标系运行中线性轴低速 JOG 运行.	长实数(LREAL)小于或者等于 XYZ 高速.	1 mm/sec
ABC 低速	设置在坐标系运行中旋转轴低速 JOG 运行.	长实数(LREAL)小于或者等于 ABC 高速.	1 deg/sec
XYZ 高速	设置在坐标系运行中线性轴高速 JOG 运行.	长实数(LREAL)	5 mm/sec
ABC 高速	设置在坐标系运行中旋转轴高速 JOG 运行.	长实数(LREAL)	5 deg/sec

7. NC 参数

- 设置 NC 控制相关参数.
- 更多内容, 请参考第 9 章 NC 控制功能 -9.4 NC 参数.

8. CAM DATA

- 设置用于 CAM 运行的 CAM 轮廓数据.
- 更多内容, 请参考第 8 章, 运动控制功能 -8.2.11 (3) 同步控制部分的 CAM 运行.

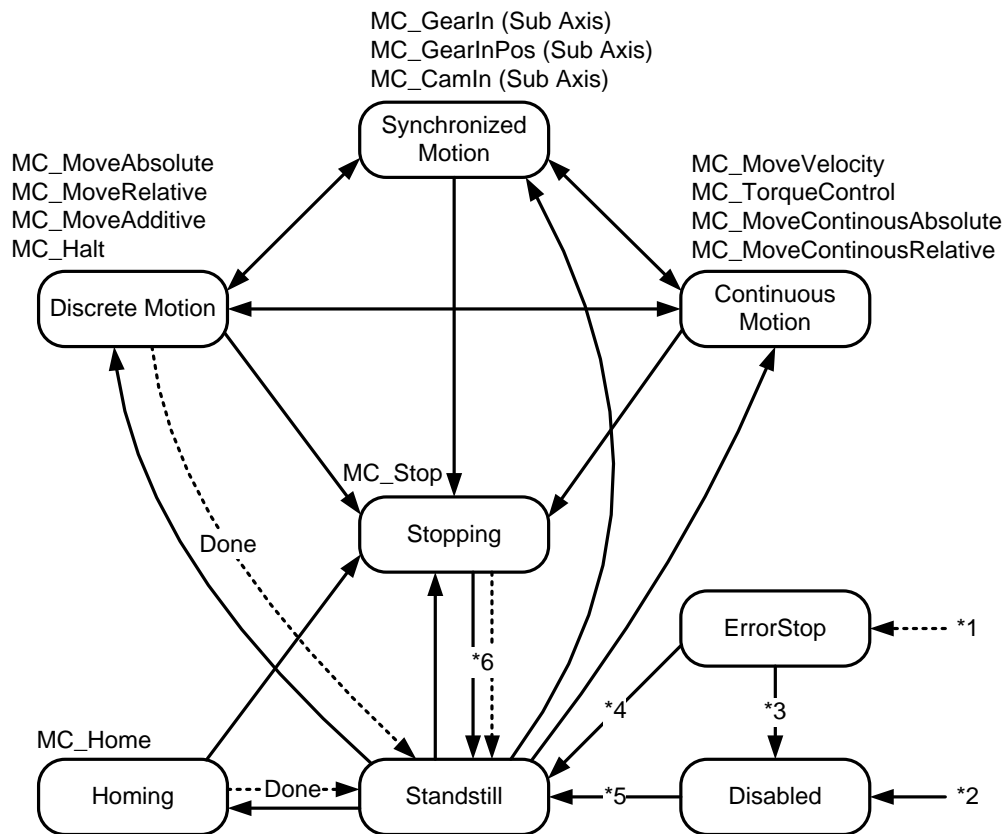
第 6 章运动功能块

本章主要说明前章节中和其它应用功能块库中所涉及到的基本功能块库。

6.1 运动功能块通用要素

6.1.1 轴状态

运动控制模块的每个轴根据运行情况和命令改变为相关联状态。每种情况的结构改变如下图所示。



*1 ErrorStop: 如果轴发生错误，无论轴的当前状态

*2 Disabled: 如果 `MC_Power.Enable` 输入为 Off，当轴没有发生错误

*3 ErrorStop → Disabled: 如果 `MC_Reset` 命令已执行，当 `MC_Power.Status` 输出为 Off

*4 ErrorStop → Standstill: 如果 `MC_Reset` 命令已执行，当 `MC_Power.Status` 输出为 on
`MC_Power.Enable` 输入为 On

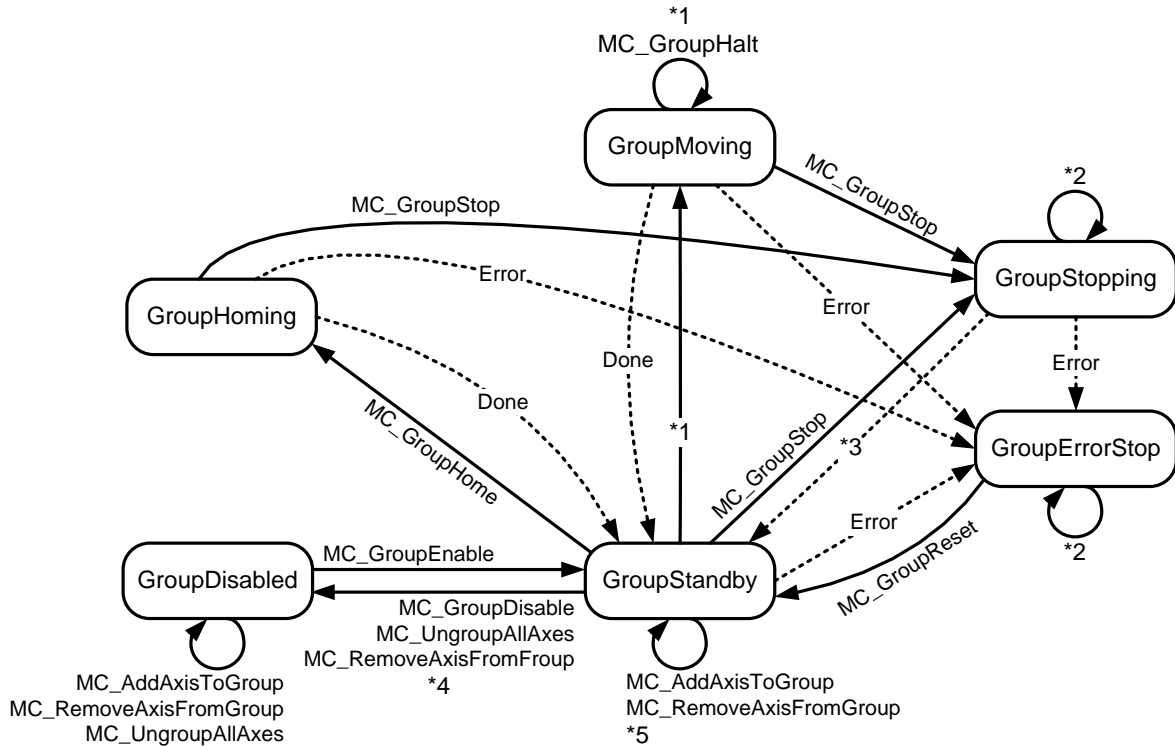
*5 Disabled → Standstill: 如果打开 `MC_Power.Enable` 输入当 `MC_Power.Status` 输出为 On

*6 Stopping → Standstill: 如果关断 `MC_Stop` 执行输入当 `MC_Stop.Done` 输出为 On

轴状态	描述
Disabled	Disabled 表示单轴无命令给定状态，没发生错误。如果第一次运行没有运动控制模块时，每轴开始处于 disabled 状态。如果当 Servo On/Off (MC_Power)电机使能输入功能块为 On。如果当 Servo On/Off (MC_Power)电机使能输入功能块为 Off，轴状态变为 disabled。如果不是错误停止状态，当前运动功能块动作，命令中断。(运动块功能指令中止为 On)
ErrorStop	无论当前轴是何种状态，当轴发生错误时，改变为 ErrorStop 状态，轴减速停止。当错误发生状态，即使执行 servo On/Off (MC_Power) 运动功能，ErrorStop 状态仍维持。处于 ErrorStop 状态的轴维持静止状态，除错误复位指令可执行外，其它命令不可用。
StandStill	当激活轴的电源时，轴没有错误且没有执行命令，轴状态表示 StandStill 状态。
Homing	Homing 状态表示轴正在原点返回运行。
Stopping	如果执行紧急停止 (MC_Stop) 功能块，轴状态改变为 stopping 状态。当轴处于 stopping 状态，直到停止完成后，其它监控命令才能应用到轴上，(直到 Done 输出激活)。如果 Done 输出为 On,执行输入为 On，状态转换为 Standstill 状态。
Continuous Motion	直到当前轴变为运行停止状态前，表示连续运行状态。
Discrete Motion	表明通过目标位置减少运行状态。
Synchronized Motion	Synchronized motion 表示轴同步运行。

6.1.2 组状态

运动控制模块每个组变化的相关联状态取决于当前状态和命令。状态的变化结构如下图所示。



*1 GroupMoving:如果常规组运行的运动功能块执行

*2 GroupStopping, GroupErrorStop

: 当不同的运动功能块 GroupStopping 或 GroupErrorStop 状态执行，相关联运动功能块不执行，当 MC_GroupReset 功能块在 GroupErrorStop 状态执行，相关联组的状态改变为 GroupStandby。

*3 GroupStopping → GroupStandby

: 当 MC_GroupStop.DONE 输出为 On 且 MC_SroupStop.执行输入为 Off

*4 GroupStandby → GroupDisabled

: 如果执行轴删除指令，轴不属于组

(MC_RemoveAxisFromGroup, MC_UnGroupAllAxes)

*5 GroupStandby

: 如果在组中执行轴增加或删除指令，超过一个轴属于组。

(MC_AddAxisToGroup, MC_RemoveAxisFromGroup)

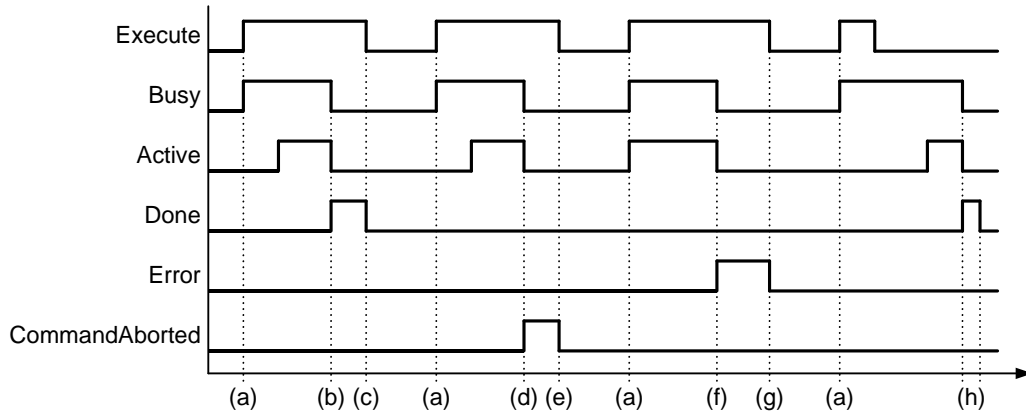
*6 GroupDisabled

: 当执行 MC_GroupDisable 或 MC_UnGroupAllDisable 功能块，不管当前状态，相关组变为 GroupDisabled 状态。

6.1.3 基本 I/O 变量

1. Edge 运行运动功能块

基本 I/O 参数在 Edge 运行运动功能块的关系如下。

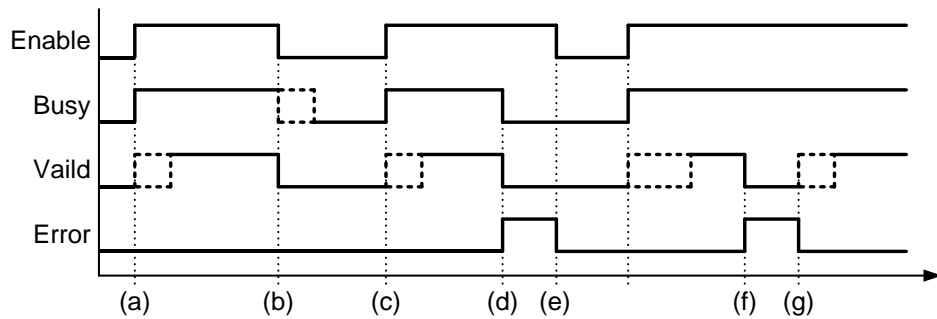


变量	描述
Execute	在 Edge 运行功能块时，输入运行相关功能块。功能块在上升沿执行。(图 a 状态)
Busy	该输出表示相关运动功能块正在运行(= 未完成)，表示运动功能块输出可以改变。在执行输入(图 a 状态)的上升沿，Busy 输出为 On，当 Done 输出为 On (图 b 状态)时为 Off，CommandAborted 输出为 On (图 d 状态)，或 Error 输出为 On (图 f 状态)。
Active	表示相关运动功能块为实际控制轴。 当一个轴运行多个运动功能块时(如果只有一个运动功能块正在控制，其它功能块缓存)，当只有一个运动功能块被控制时，Active 输出为 On，运动功能块缓存，Busy 输出为 On。
Done	该输出表示相关运动功能块的已经完成。 如果 Done 输出为 On, Busy 和 Active 输出为 Off。(图 d 状态) 当执行输入为 Off 时，Done 输出为 Off (图 e 状态)，如果当 Done 输出变为 On，执行输出为 Off，仅在一个扫描内保持 On (图 h 状态)。
Error	该输出表示运动功能块运行时，发生错误。 当执行输入为 Off 时，Error 输出为 Off (图 f 状态)。如果 Error 输出变为 On，执行输出为 Off，仅在一个扫描内保持 On (图 h 状态)。

变量	描述
ErrorID	运动功能块运行时发生错误，相关联错误的错误代码输出。ErrorID 输出和消除时间和 Error 输出相同。
CommandAborted	表示相关运动功能块被其它运动功能块中断。当执行输入为 Off 时，CommandAborted 输出为 Off (图 g 状态)。当 Done 输出变为 On 时，执行输出为 Off，仅维持在一个扫描内为 On。

※当 Edge 运行(执行输入) 运动功能块时，执行输入为 On，取决于轴状态，在 Busy, Done, Error 和 CommandAborted 中一个输出为 On。Busy, Done, Error 和 CommandAborted 输出在一个时间内为 On 可用，四个中一个输出为 On，其它三个变为 Off。

2. 运动功能块水平运动



变量	描述
Enable	用于水平动作输入运行功能块。 在上升沿运行运动功能块 (图 a 状态)，在下降沿停止(图 b 状态)。
Busy	该输出表示相关运动功能块正在运行(= 未完成)，表示运动功能块输出可变。Enable 输入的上升沿 Busy 输出为 On (图 b 状态)，当运动功能运行中维持为 on。
Valid	该输出表示相关运动功能块成功执行，输出&运动 Valid。 当 Enable 输入为 Off 时，Valid 输出为 Off(图 b 状态)。

第 6 章运动功能块

变量	描述
Error	<p>该输出表示当运行运动功能块时，发生错误。</p> <p>当运动功能块运行时，发生的错误不能自动存储，Error 输出为 On, Busy & Valid 输出为 Off (图 d 状态), 运动功能块停止运行。当 Enable 输入为 Off 时，Error 输出为 Off (图 e 状态)</p> <p>当功能块运行时，发生的错误自动存储，Error 输出为 On, Valid 输入为 Off (图 f 状态).</p> <p>当错误在相关运动功能块被存储，Error 输出为 Off, 运行恢复 (图 g 状态).</p>
ErrorID	<p>运动功能块运行时发生错误，相关联错误的错误代码输出。ErrorID 输出和消除时间和 Error 输出相同。</p>
<p>※ Valid 和 Error 输出不同时为 On。</p>	

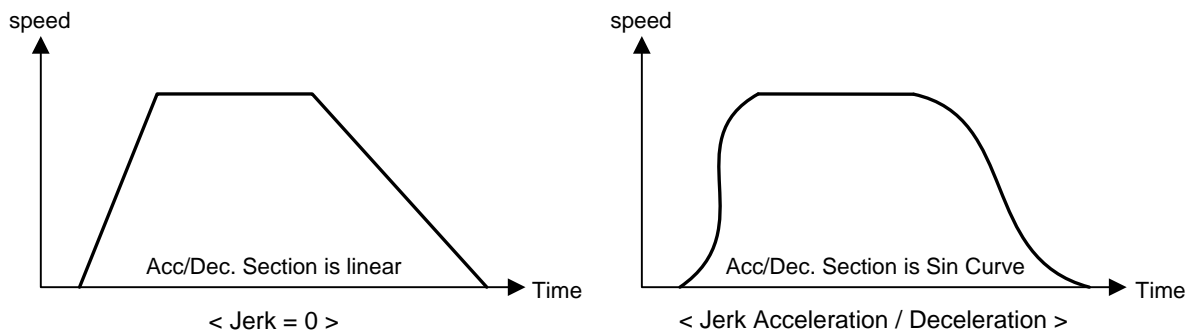
备注

1.轴输入

每个运动功能块的轴输入由相关命令控制。运动控制模块可控制 1-32 实轴和 33-36 虚轴，1001~1002 编码器用作于主轴.因此，根据运动功能块 1~32, 33~36 和 1001~1002 的值用于轴输入。当设置运动功能块超出可用范围，发生"error 0x0006"。

2.加速度

加速度设置加速/减速比率。低加速度值导致加速/减少接近于平整线。高加速度值导致加速/减少接近于正弦曲线。



6.1.4 缓存模式输入

该输入规定是否要等到现有指令完成或取消已有的运动功能块和执行命令的情况下，当其他运动功能块运行某一个轴，轴运行其他运动功能块。可指定 0-5 的编号，如果超出范围，轴命令出现 "error 0x101A"，轴组命令发生"error 0x201A"。缓存模式的设置可用值如下。

编号	缓存模式	说明
0	mcAborting	立即执行命令。现有命令在运行时中断
1	Mcbuffer	现有命令运行完成后执行命令
2	mcBlendingLow	结合现有指令的速度运行，命令分配给低速比较
3	mcBlendingPrevious	结合现有指令的速度运行
4	mcBlendingNext	结合现有指令分配的速度运行
5	mcBlendingHigh	结合现有指令的速度运行，命令给高速比较

6.1.5 运动功能块执行中改变参数

相关命令的参数在运动功能块运行时可以改变，详细操作如下。

- (1) 连续更新输入在 Off 状态下，运动功能块执行 Edge 操作 (启用执行输入)，相关运动功能块在执行输入为 On (上升沿)时，参数应用运行。这样的话，运动功能块在运行中改变参数输入值，不影响运行。当相关运动功能块在运行中，想要改变参数，再次打开执行输入。
- (2) 连续更新输入在 On 状态下，运动功能块执行 Edge 操作(启用执行输入)，当执行输入首次为 On (上升沿) 时，参数首先应用。
当连续更新输入为 On 时，改变参数，相关运动功能块运行反应参数的变化。
但是，如果相关运动功能块在完成或停止后(Busy 输出为 Off)，改变参数，不再反应改变。(参数改变运行使用连续更新，不再运行已完成或中断的运动功能块，也就是说，连续更新运行仅应用当前运行的运动功能块。)
- (3) 对于没有持续更新输入的功能块，在指令完成前通过再执行功能块(执行输入为 On)应用变更参数。
- (4) 对于 level 运行运动功能块，当 Enable 输入为 On(上升沿)时，根据应用的参数运行，当 Enable 输入为 On 时，连续改变参数可用。

- (5) 对于 MC_CAMIN 功能块, 仅以下输入可以更新:主站补偿,从站补偿,主站缩放,从站缩放,主站启动距离,和主站同步位置(如果 InSync=On, 仅站补偿,从站补偿,主站缩放和从站缩放更新.)
- (6) 对于 MC_GEARIN 功能块, 仅以下输入可以更新: 比例分子,比例分母,加速,和减速(如果 InGear=On, 仅比例分子和比例分母可以更新)

6.1.6 组运行路径改变设置

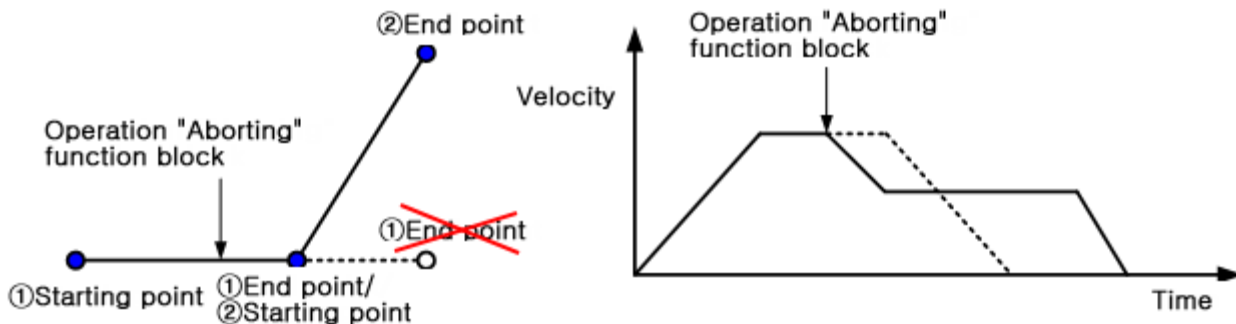
当前运动控制模块轴组执行命令时, 其他命令分配给相关轴组。此时, 关于路径, 下一个命令可获得, 可指定已有指令连接现存路径。连接轨迹参数在转换参数输入中指定。

编号	转换模式	说明
0	TMNone	不产生连接轨迹。
3	TMCornerDistance	生成指定连接轨迹的角距离, 在指定角距离处绘制圆弧角。

1. 转换模式“TMNone”

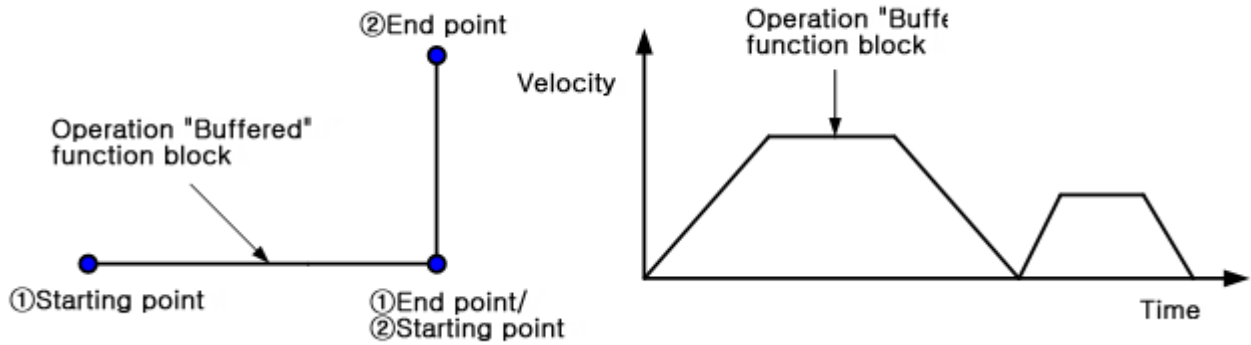
连接轨迹不产生。如果运动功能块的缓存模式输入“Aborting”或“缓存”, 转换模式输入仅为“TMNone”可用。

下图表示在运动功能的缓存模式块设置为‘Aborting’状态下运行时情形。左图表示运动功能块①运行时, 运动功能块②在‘Aborting’下执行。运动功能块①被强制结束/结束点① / 开始点②, 没有到达‘结束点①’。右图表示‘Aborting’功能块执行时, 执行减速停止, 下一个运动功能块执行。



<如果缓存模式为“Aborting”>

下图表示在运动功能的缓存模式块设置为“缓存”状态下运行时情形。左图表示运动功能块①运行时, 运动功能块②在‘缓存’下执行。当运动功能块①到达目标位置后, 执行运动功能块②。右图表示当‘缓存’功能块执行时, 当到达原始目标位置时, 执行运动功能块。



<如果缓存模式为“缓存”>

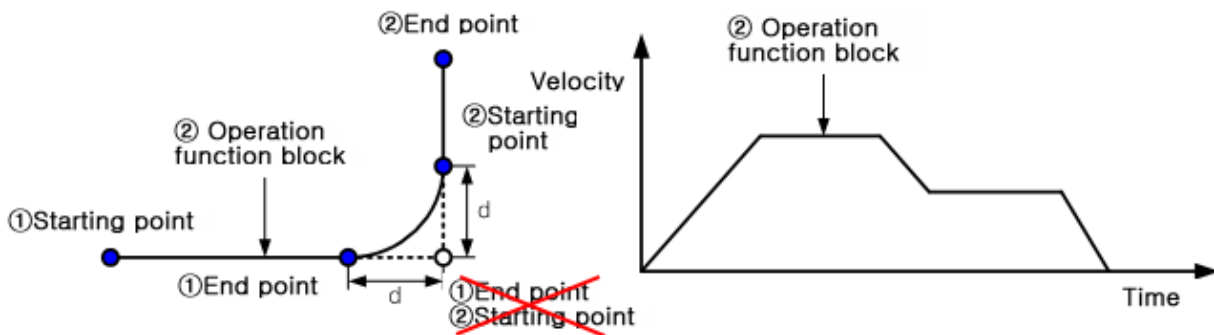
2. 转换模式“TMCornerDistance”

连接轨迹的半径被指定，连接轨迹绘制指定半径圆弧输出。仅当缓存模式为“BlendingXXXX”时可运行，当缓存模式为“Aborting”或“缓存”时，在“TMNone”中运行。

当绘制连接轨迹时，路径的最大速度在缓存模式下按照指定速度，半径长度按照转换参数指定的值。

下图表示连接轨迹在两个线性插补命令中绘制半径圆弧的生成。左图表示当运动功能块①运行时，在“TMcornerDistance”设置下执行运动功能块②。运动功能块①原始目标位置为结束点① / 开始点②，但是直线运动停止，圆弧运动尽可能远离半径'd' (结束点①) 开始。圆弧运动在结束点①开始，在开始点②结束，执行运动功能块②。

右图表示在两个功能块中间速度不停止，连续。



<如果缓存模式指定为“BlendingLow”，转换模式指定为“TMCornerDistance”>

第 6 章运动功能块

6.1.7 运动功能块错误

运动功能块变量错误 ErrorID 如下。

STAT	内容	详细描述
0x0000	正常	如果运动功能块正常执行，“O”显示在 ErrorID.
0x0005	当前运动模块不支持运动功能块.	运动功能块超出当前模块版本不执行。检查运动功能块版本。
0x0006	动作功能块 (轴输入) 轴数目超出允许范围。	通过运动功能块，检查轴能完成的分配，设置轴数目范围 1~36 和 1001~1002。
0x0007	运动功能块 (轴组输入)轴组数目超出允许范围。	设置轴组数目的值在 1 和 16 中间。
0x0008	功能块 NC 通道超出允许范围.	查看 NC 通道范围, 并且再次设置.
0x0009	功能块从站数量(从站输入)超出允许范围.	查看从站数量范围,并且再次设置.
0x000B	功能块输入超出允许范围.	查看功能块输入范围, 并且再次设置.
0x000C	功能块阵列输入超出允许范围.	查看功能块阵列输入尺寸, 并且再次设置.
0x0012	在执行运动功能块中, 运动功能块内部执行错误发生	检查 XG-PM 和 XMC-E32A 版本。
0x0013	运动功能块执行中, 运动响应错误发生	检查 XG-PM 和 XGF-M32E 版本。
0x0020 : 0x0FFF	表示运动控制模块发生常规错误。 更多细节, 参考‘附录 2 错误信息和措施’。	
0x1000 : 0x1FFF	表示运动控制模块的相关联轴控制中发生错误。 更多细节, 参考‘附录 2 错误信息和措施’。	
0x2000 : 0x2FFF	表示运动控制模块的相关联轴控制中发生错误。 更多细节, 参考‘附录 2 错误信息和措施’。	
0x3000 : 0x3FFF	表示运动控制模块的相关 NC 控制中发生错误。 更多细节, 参考‘附录 2 错误信息和措施’。	

6.2 运动功能块

编号	名称	描述	动作条件
单轴运动命令			
1	MC_Power	Servo On/OFF	水平
2	MC_Home	执行搜索原点	边沿
3	MC_Stop	立即停止	边沿
4	MC_Halt	停止	边沿
5	MC_MoveAbsolute	绝对位置控制运行	边沿
6	MC_MoveRelative	相对位置控制运行	边沿
7	MC_MoveAdditive	增加位控运行	边沿
8	MC_MoveVelocity	定速运行	边沿
9	MC_MoveContinuousAbsolute	使用定速运行结束位控运行	边沿
10	MC_MoveContinuousRelative	使用定速运行结束相对位控运行	边沿
11	MC_TorqueControl	转矩控制	边沿
12	MC_SetPosition	设置当前位置	边沿
13	MC_Set覆盖	速度/加速优先	水平
14	MC_ReadParameter	读参数	水平
15	MC_WriteParameter	写参数	边沿
16	MC_Reset	轴错误复位	边沿
17	MC_TouchProbe	接触探头	边沿
18	MC_AbortTrigger	中止触发事件	边沿
19	MC_MoveSuperImposed	超级叠加运行	边沿
20	MC_HaltSuperImposed	超级叠加运行停止	边沿
多轴命令			
21	MC_CamIn	凸轮运行	边沿
22	MC_CamOut	凸轮停止	边沿
23	MC_GearIn	电子传动运行	边沿
24	MC_GearOut	电子传动解除	边沿
25	MC_GearInPos	指定位置电子传动	边沿
26	MC_Phasing	相位补偿	边沿

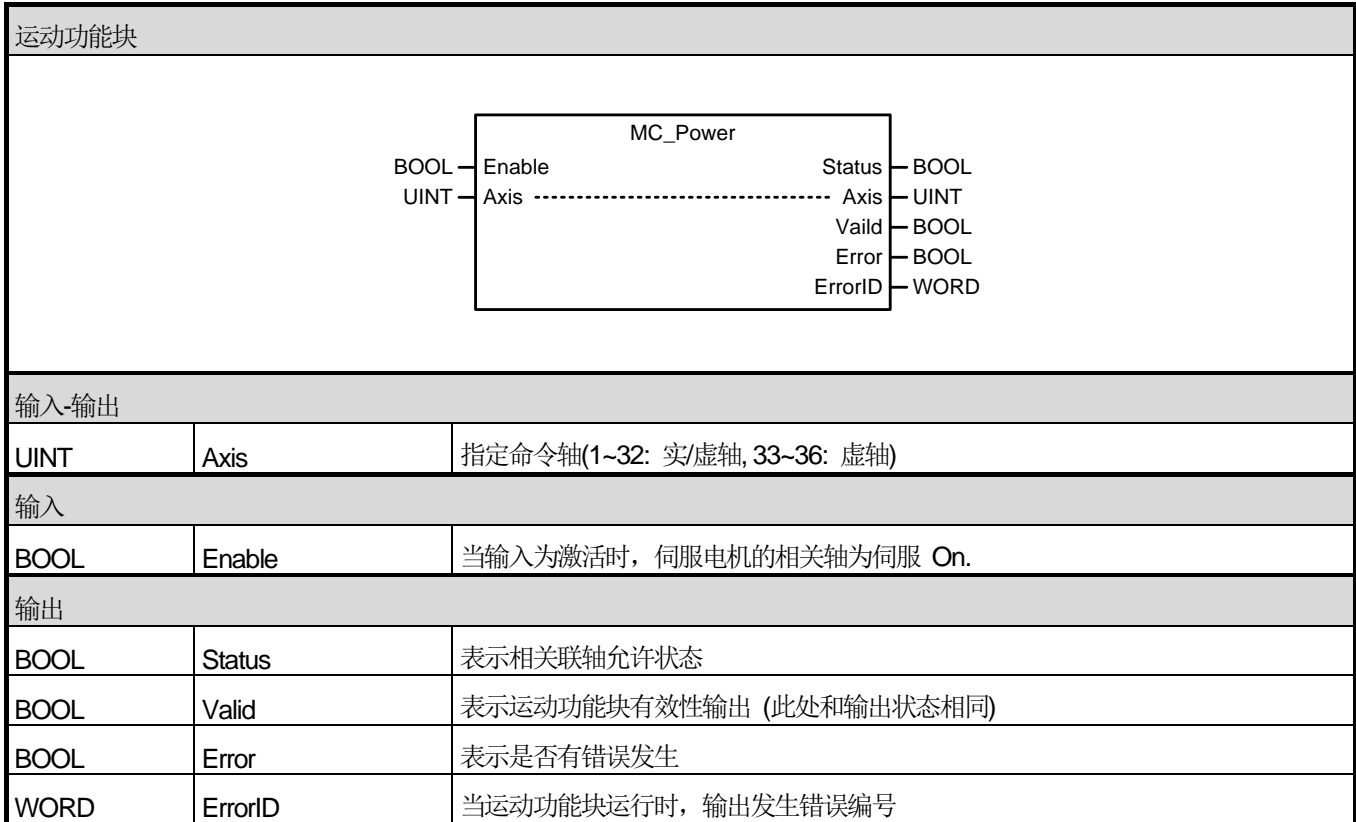
第 6 章运动功能块

编号	名称	描述	动作条件
组命令			
27	MC_AddAxisToGroup	在AxesGroup结构中添加一个轴到组中	边沿
28	MC_RemoveAxisFromGroup	在AxesGroup结构中在组中删除一个轴	边沿
29	MC_UngroupAllAxes	从组AxesGroup删除全部轴	边沿
30	MC_GroupEnable	从GroupDisabled 到 GroupEnable改变组的状态	边沿
31	MC_GroupDisable	改变组状态为 GroupDisabled	边沿
32	MC_GroupHome	执行AxesGroup 搜索原点顺序	边沿
33	MC_GroupSetPosition	无需移动在组中设置全部轴位置	边沿
34	MC_GroupStop	立即停止组	边沿
35	MC_GroupHalt	停止一个组	边沿
36	MC_GroupReset	复位一个组错误	边沿
37	MC_MoveLinearAbsolute	绝对位置线性插补运行	边沿
38	MC_MoveLinearRelative	相对位置线性插补运行	边沿
39	MC_MoveCircularAbsolute	绝对位置圆弧插补运行	边沿
40	MC_MoveCircularRelative	相对位置圆弧插补运行	边沿
LS指令			
41	LS_Connect	连接伺服驱动器	边沿
42	LS_Disconnect	断开伺服驱动器	边沿
43	LS_ReadSDO	读取SDO	边沿
44	LS_WriteSDO	写入SDO	边沿
45	LS_SaveSDO	保存SDO	边沿
46	LS_编码器Preset	编码器预设	边沿
47	LS_Jog	JOG 运行	水平
48	LS_ReadCamData	读取CAM数据	边沿
49	LS_WriteCamData	写入CAM数据	边沿
50	LS_ReadEsc	读取ESC	边沿
51	LS_WriteEsc	写入ESC	边沿
52	LS_CamSkip	跳跃CAM	边沿
53	LS_VarCamIn	变量CAM运行	边沿
54	LS_VarGearIn	变量齿轮运行	边沿
55	LS_VarGearInPos	变量定位齿轮运行	边沿
56	LS_ReadCAM tableSlavePos	读取CAM表从站位置	边沿
57	LS_InverterWriteVel	写入变频器速度	边沿
58	LS_InverterReadVel	读取变频器速度	水平
59	LS_InverterControl	写入变频器控制字	边沿
60	LS_InverterStatus1	读取变频器状态1	水平
61	LS_InverterStatus2	读取变频器状态1	水平
62	LS_SyncMoveVelocity	速度控制运行(csv模式)	边沿

编号	名称	描述	动作条件
坐标系指令			
63	MC_SetKinTransform	机械信息设置	边沿
64	MC_SetCartesianTransform	PCS设置	边沿
65	LS_SetWorkSpaceTransform	工作区设置	边沿
66	LS_MoveLinearTimeAbsolute	用于坐标系绝对位置的时间- 线性插值运行	边沿
67	LS_MoveLinearTimeRelative	用于坐标系相对位置的时间- 线性插值运行	边沿
68	MC_MoveCircularAbsolute2D	用于坐标系绝对位置的圆弧插值运行	边沿
69	MC_MoveCircularRelative2D	用于坐标系相对位置的圆弧插值运行	边沿
70	MC_TrackConveyorBelt	传送带同步设置	边沿
71	MC_TrackRotary table	转盘同步设置	边沿
72	LS_RobotJOG	坐标系点动运行	水平
73	LS_SetMovePath	设置路径运行数据	边沿
74	LS_ResetMovePath	删除路径运行数据	边沿
75	LS_GetMovePath	读取路径运行数据	边沿
76	LS_RunMovePath	执行路径运行	边沿
NC控制命令			
77	NC_LoadProgram	指定NC程序	边沿
78	NC_BlockControl	指定块运行	水平
79	NC_Reset	复位	边沿
80	NC_Emergency	紧急停止	水平
81	NC_CycleStart	启动自动运行	边沿
82	NC_FeedHold	馈送保持	水平
83	NC_Home	原点返回	边沿
84	NC_RapidTraverse覆盖	快速移位覆盖	水平
85	NC_CuttingFeed覆盖	剪切馈送覆盖	水平
86	NC_Spindle覆盖	主轴覆盖	水平
87	NC_M codeComplete	M代码运行完成	边沿
88	NC_ScodeComplete	S代码运行完成	边沿
89	NC_TcodeComplete	T代码运行完成	边沿
90	NC_ReadParameter	读取NC参数	水平
91	NC_WriteParameter	写入NC参数	边沿

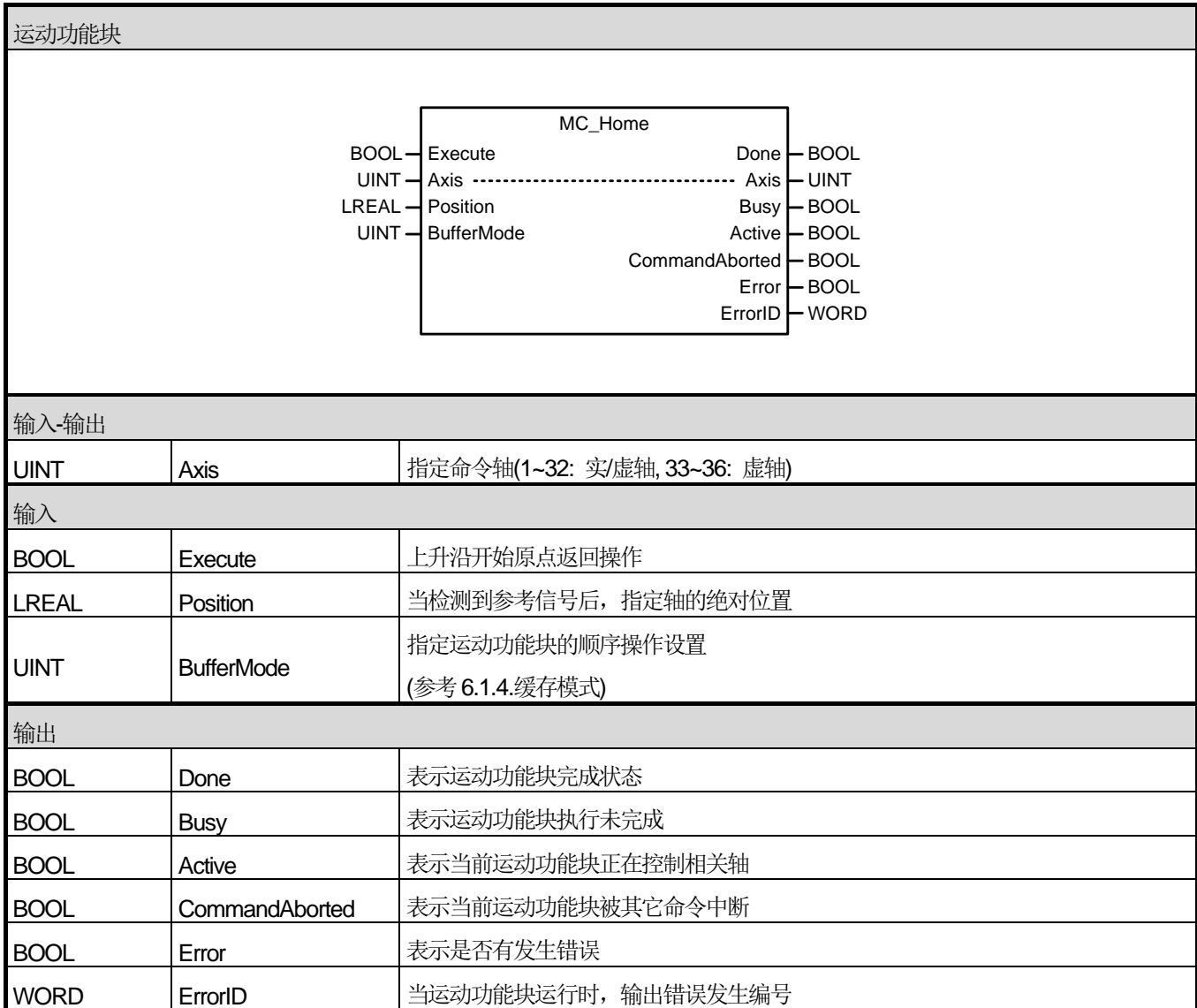
6.3 单轴运动功能块

6.3.1 伺服 On/Off (MC_Power)



- (1) 运动功能块给相关轴伺服 On/Off 指令。
- (2) 当 Enable 输入为 On 时, 伺服 On 指令给相关轴, 当为 Off 时, 给伺服 Off 指令。
- (3) 当轴为'Disable'状时, 执行伺服 On 指, 轴状态为'StandStill', 不能伺服 On, 发生 'ErrorStop'状态。

6.3.2 执行搜索原点 (MC_Home)

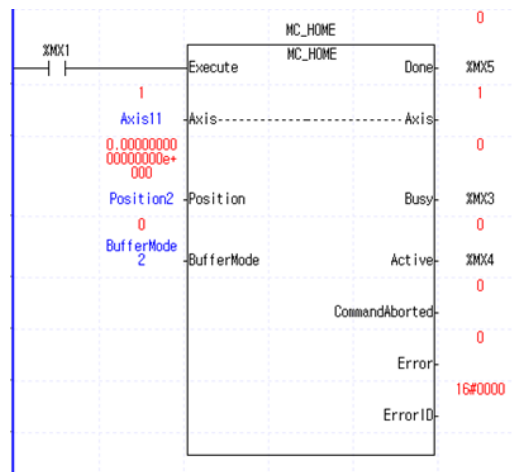


- (1) 运动功能块给相关轴原点返回命令。
- (2) 原点返回方式在相关轴运行运行参数前设置。
- (3) 对于位置输入, 当参考信号被检测到或原点返回完成指定轴的绝对位置。
- (4) 当运动功能块正在运行时, 轴的状态为 'Homing'状态, 当命令完成后, 转换到'Standstill'状态。
- (5) 指令完成前通过再次执行功能块(执行输入为 On)应用变更参数. 仅可以更新位置输入。
- (6) 实例程序

以下实例显示当当前指令位置为 100,000 时执行 MC_HOME 指令.

第 6 章运动功能块

(a) 功能块设置

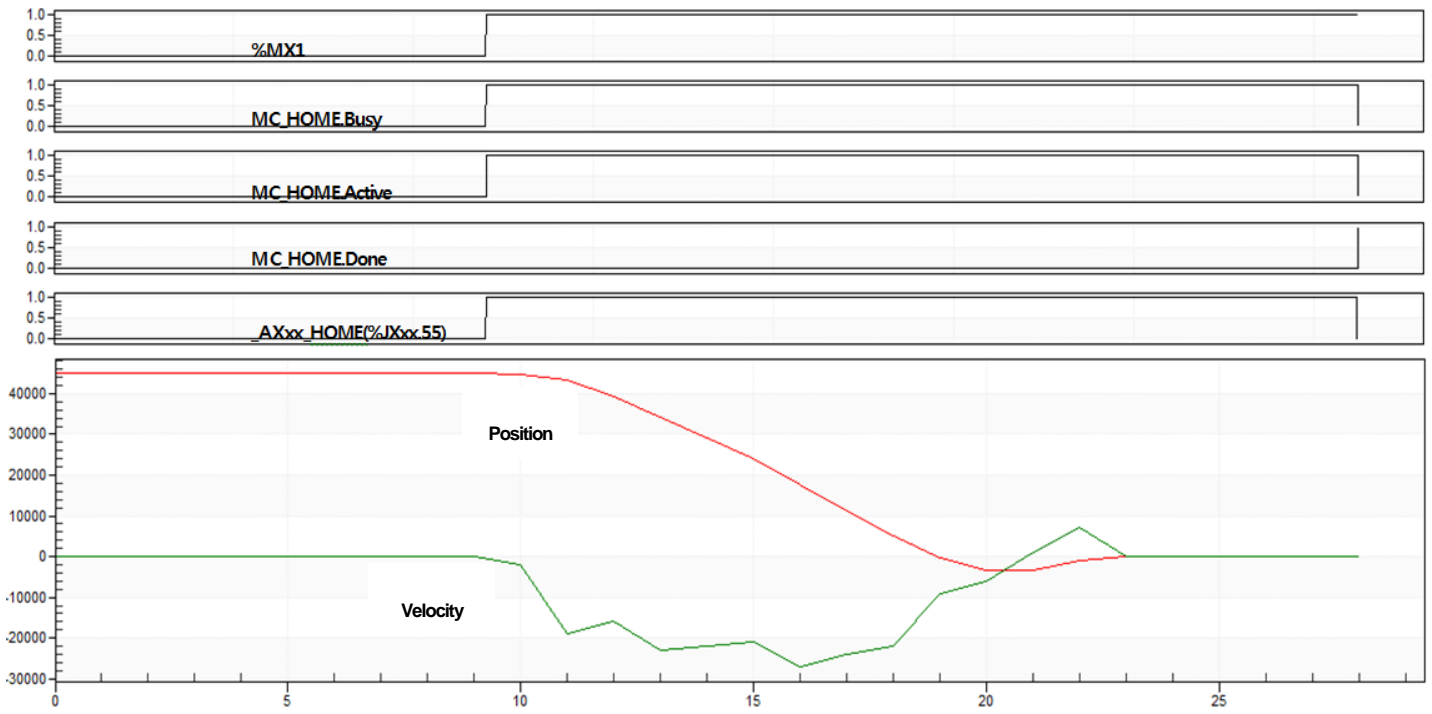


(b) 参数设置

- 在 SDO 参数中设置原点返回方式为 33.

Index	Name	Unit	Current Value	Initial Value	Access
607A	Target Position	UU	0	0	rw
607C	Home Offset	UU	0	0	rw
607D:00	Software Position Limit	-	2	2	rw
607F	Maximum Profile Velocity	UU/s	2147483647	2147483647	rw
6080	Maximum Motor Speed	rpm	0	0	ro
6081	Profile Velocity	UU/s	200000	200000	rw
6083	Profile Acceleration	UU/s ²	200000	200000	rw
6084	Profile Deceleration	UU/s ²	200000	200000	rw
6085	Quick Stop Deceleration	UU/s ²	2000	2000	rw
6087	Torque Slope	0.1%/s	1000	1000	rw
6091:00	Gear Ratio	-	2	2	rw
6098	Homing Method	-	33	34	rw
6099:00	Homing Speeds	-	2	2	rw
609A	Homing Acceleration	UU/s ²	200000	200000	rw
60B0	Position Offset	UU	0	0	rw
60B1	Velocity Offset	UU/s	0	0	rw
60B2	Torque Offset	0.1%	0	0	rw

(c) 时序图



6.3.3 立即停止 (MC_Stop)



- (1) 运动功能块给相关轴紧急停止。
- (2) 当执行立即停止 (MC_Stop) 运动功能块, 相关轴已有运动功能块停止, 轴状态变为 'Stopping'。当相关轴为'Stopping'状态, 其它运动功能块在 stopping 完成之前, 不能执行相关轴。(直到 Done 输出为激活)
- (3) CommandAborted 输出表示在运行时, 当前运动功能块被中断。当立即停止(MC_Stop) 运动功能块运行时, 其它运动功能块不能中断立即停止 (MC_Stop) 运动功能块。因此, 当伺服电源断开或执行伺服 Off 命令, CommandAborted 输出通常为 On。
- (4) 如果执行输入为 On 或轴的速度不为 0, 轴为'Stopping'状态, 当 Done 输出为 On , 执行输入为 Off 时, 转换至 'Standstill'状态。
- (5) 指令完成前通过再次执行功能块(执行输入为 On)应用变更参数. 仅可以更新减速和加速度输入.

(6) 6.3.4 停止 (MC_Halt)

运动功能块		
输入-输出		
UINT	Axis	指定命令轴 (1~32: 实/虚轴, 33~36: 虚轴)
输入		
BOOL	Execute	上升沿.给相关轴停止命令
LREAL	Deceleration	停止时指定减速[u/s ²]
LREAL	Jerk	指定加速/减速变化率 [u/s ³]
UINT	BufferMode	指定运动功能块的顺序运行设置 (参考 6.1.4.缓存模式)
输出		
BOOL	Done	表示相关轴速度达到 0.
BOOL	Busy	表示运动功能块执行未完成。
BOOL	Active	表示当前运动功能块控制相关轴。
BOOL	CommandAborted	表示当前运动功能块在运行时中断。
BOOL	Error	表示是否发生错误。
WORD	ErrorID	当运动功能块运行时，输出错误发生编号。

- (1) 运动功能块给相关轴一个停止命令。
- (2) 当运动功能块正在运行时，轴为'DiscreteMotion'状态。当相关轴速度为 0 时，'Done' 输出为 On 改变为Standstill'状态。
- (3) 与 MC_Stop 指令不同的是可以选择缓存模式. Halt 指令(MC_Halt)可以通过另一个运动功能块停止。
- (4) 指令完成前通过再次执行功能块(执行输入为 On)应用变更参数。

6.3.5 绝对位置运行 (MC_MoveAbsolute)

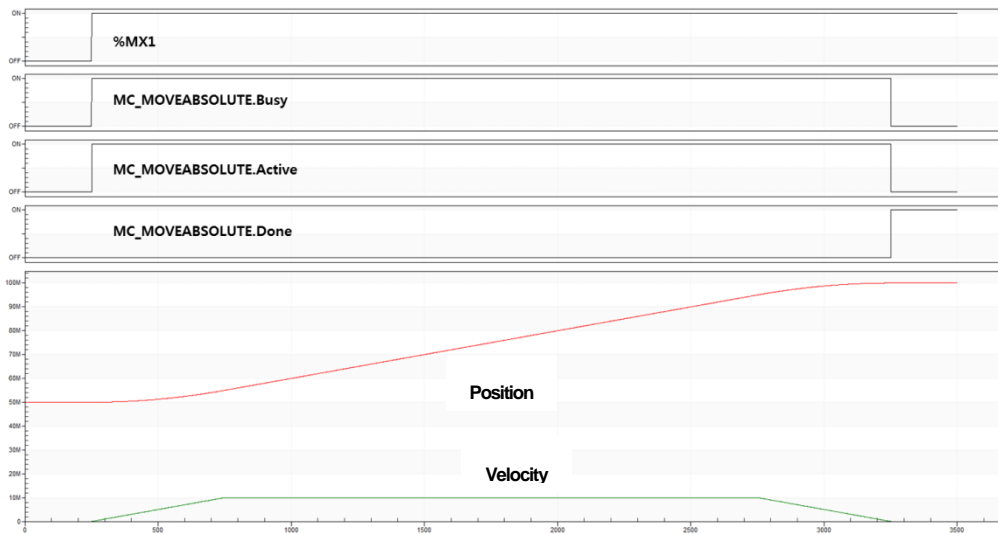
运动功能块		
<pre> graph LR subgraph MC_MoveAbsolute Execute[Execute] Axis[Axis] ContinuousUpdate[ContinuousUpdate] Position[Position] Velocity[Velocity] Acceleration[Acceleration] Deceleration[Deceleration] Jerk[Jerk] Direction[Direction] BufferMode[BufferMode] Done[Done] Busy[Busy] Active[Active] CommandAborted[CommandAborted] Error[Error] ErrorID[ErrorID] end Execute --- Done Axis --- Axis ContinuousUpdate --- Busy Position --- Active Velocity --- CommandAborted Acceleration --- Error Deceleration --- ErrorID Jerk --- ErrorID Direction --- ErrorID BufferMode --- ErrorID </pre>		
输入-输出		
UINT	Axis	指定命令轴 (1~32: 实/虚轴, 33~36: 虚轴)
输入		
BOOL	Execute	在上升沿给相关轴一个绝对位置运行命令。
BOOL	ContinuousUpdate	指定输入值设置更新。 (参考 6.1.5 在运动功能块执行改变参数)
LREAL	Position	指定目标位置。
LREAL	Velocity	指定最大速度。 [u/s]
LREAL	Acceleration	指定加速。 [u/s ²]
LREAL	Deceleration	指定减速。 [u/s ²]
LREAL	Jerk	指定加/减速变化率。 [u/s ³]
UINT	Direction	指定运行方向。 (0~4: 0-未指定, 1-正向, 2-最短距离, 3-反向, 4-当前方向)
UINT	BufferMode	指定运动功能块顺序运行设置。 (参考 6.1.4.缓存模式)
输出		
BOOL	Done	表示是否到达指定距离。
BOOL	Busy	表示运动功能块执行未完成。
BOOL	Active	表示当前运动功能块控制相关轴。
BOOL	CommandAborted	表示当前运动功能块在运行时被中断。
BOOL	Error	表示是否发生错误。
WORD	ErrorID	当运动功能块运行时, 输出错误发生编号。

- (1) 该运动功能块给相关绝对位置运行命令。
- (2) 轴的运行方向在方向输入设置无限长度重复运行, 如果无限长度重复运行设置为禁止, 方向输入为忽略。当方向输入为最短距离(=2), 相关轴做无限长度重复运行自动选择可允许的最短距离方向。可用范围 0-4 (0-未指定, 1-正向, 2-最短距离, 3-反向, 4-当前方向), 如果超出范围发生 "错误0x1017" 。
- (3) 如果当前运动功能块后, 没有运动功能块待命, 如果到达目标点后, 速度为 0, 运行完成且Done 输出为 On。

- (4) 当运动功能块正在运行时，轴为'DiscreteMotion'状态，当运行完成转换为'Standstill'状态。
- (5) 指令完成前通过再次执行功能块(执行输入为On)应用变更参数。仅可以更新位置,速率,加速,减速,加速度,方向输入。
- (6) 实例程序
该示例显示从当前指令位置 50,000,000 移动到 100,000,000.
- (a) 功能块设置



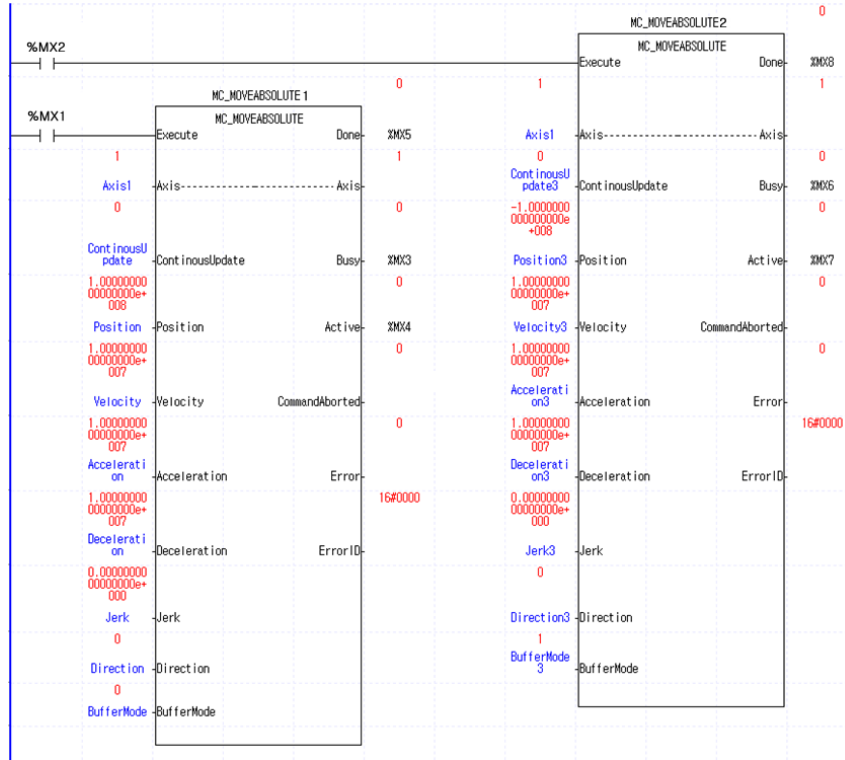
(b) 时序图



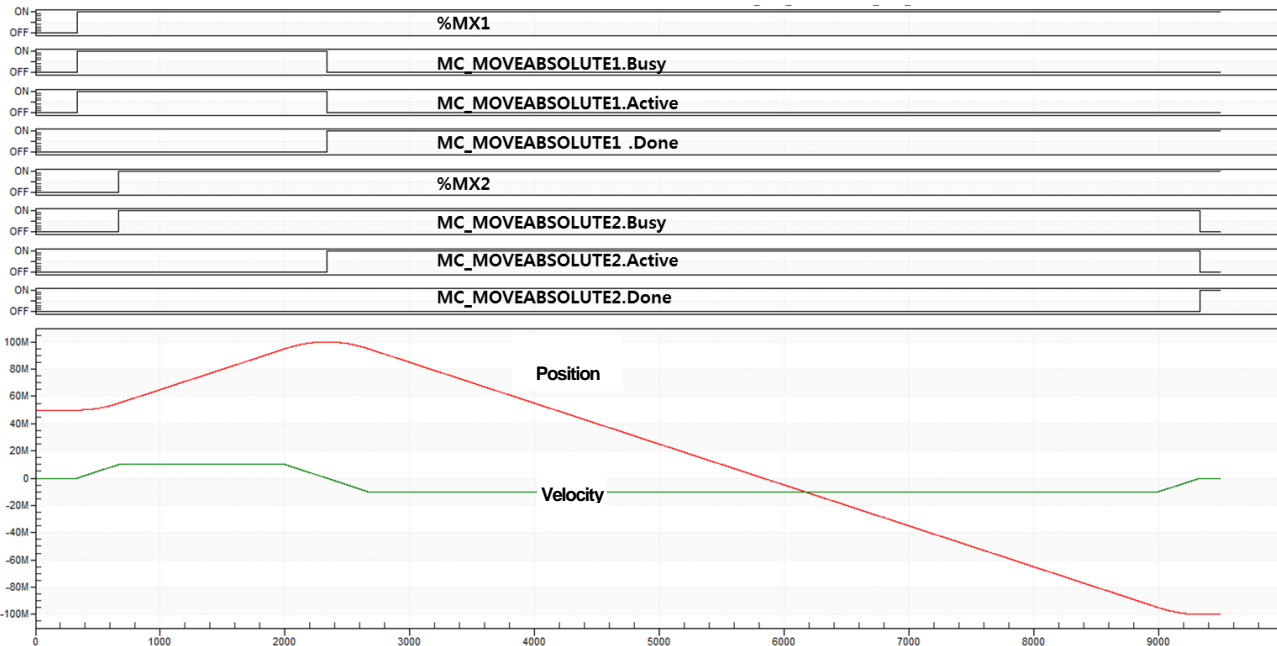
- (7) 应用实例程序
该实例显示当从50,000,000当前指令位置移动到100,000,000, 到-100,000,000通过设置缓存模式为1的另一个功能块执行

第 6 章运动功能块

(a) 功能块设置



(b) 时序图



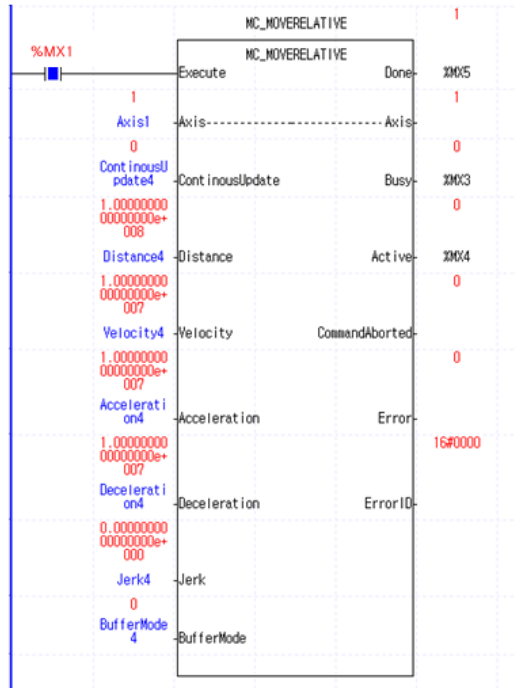
6.3.6 相对位置控制运行(MC_MoveRelative)

运动功能块		
<pre> graph LR subgraph MC_MoveRelative Execute[Execute] Axis[Axis] ContinuousUpdate[ContinuousUpdate] Distance[Distance] Velocity[Velocity] Acceleration[Acceleration] Deceleration[Deceleration] Jerk[Jerk] BufferMode[BufferMode] Done[Done] Busy[Busy] Active[Active] CommandAborted[CommandAborted] Error[Error] ErrorID[ErrorID] end Execute --- MC_MoveRelative Axis --- MC_MoveRelative ContinuousUpdate --- MC_MoveRelative Distance --- MC_MoveRelative Velocity --- MC_MoveRelative Acceleration --- MC_MoveRelative Deceleration --- MC_MoveRelative Jerk --- MC_MoveRelative BufferMode --- MC_MoveRelative MC_MoveRelative --- Done MC_MoveRelative --- Busy MC_MoveRelative --- Active MC_MoveRelative --- CommandAborted MC_MoveRelative --- Error MC_MoveRelative --- ErrorID </pre>		
输入-输出		
UINT	Axis	指定命令轴 (1~32: 实/虚轴, 33~36: 虚轴)
输入		
BOOL	Execute	在上升沿给相关轴一个绝对位置运行命令。
BOOL	ContinuousUpdate	指定输入值的更新设置。 (参考 6.1.5.运动功能块在执行中改变参数)
LREAL	Distance	指定目标距离。
LREAL	Velocity	指定最大速度。[u/s]
LREAL	Acceleration	指定加速。[u/s ²]
LREAL	Deceleration	指定减速。[u/s ²]
LREAL	Jerk	指定加/减速变化率。[u/s ³]
UINT	BufferMode	指定运动功能块的顺序运行设置。 (参考 6.1.4.缓存模式)
输出		
BOOL	Done	表示是否达到指定距离。
BOOL	Busy	表示运动功能块执行未完成。
BOOL	Active	表示当前运动功能块控制相关轴。
BOOL	CommandAborted	表示当前运动功能块在运行时被中断。
BOOL	Error	表示是否发生错误。
WORD	ErrorID	当运动功能块运行时，输出错误发生编号。

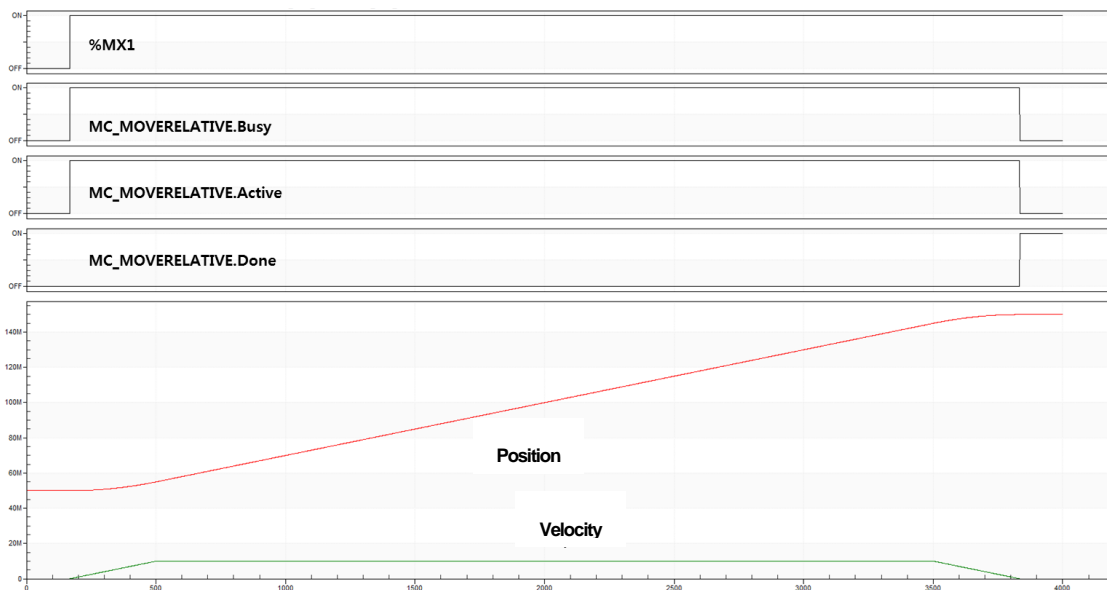
- (1) 该运动功能块给相关轴相对位置运行命令。
- (2) 相对位置运动 (MC_MoveRelative)运动功能块是指在指定距离输入中从当前位置移动至目标位置。
- (3) 移动方向取决于目标距离的指示，正 (+ 或 No 指示) 移动方向为正向，负 (-) 移动方向为反向。

第 6 章运动功能块

- (4) 如果当前运动功能块后，没有运动功能块待命，如果到达目标距离后，速度为 0，运行完成且 Done 输出为 On。
- (5) 当运动功能块正在运行时，轴为'DiscreteMotion'状态，当运行完成转换为'Standstill'状态。
- (6) 指令完成前通过再次执行功能块(执行输入为On)应用变更参数。仅可以更新位置,速率,加速,减速,加速度输入。
- (7) 实例程序
该实例显示通过对应设置值(100,000,000)移动距离从当前位置 50,000,000 移动到 150,000,000.
- (a) 功能块设置



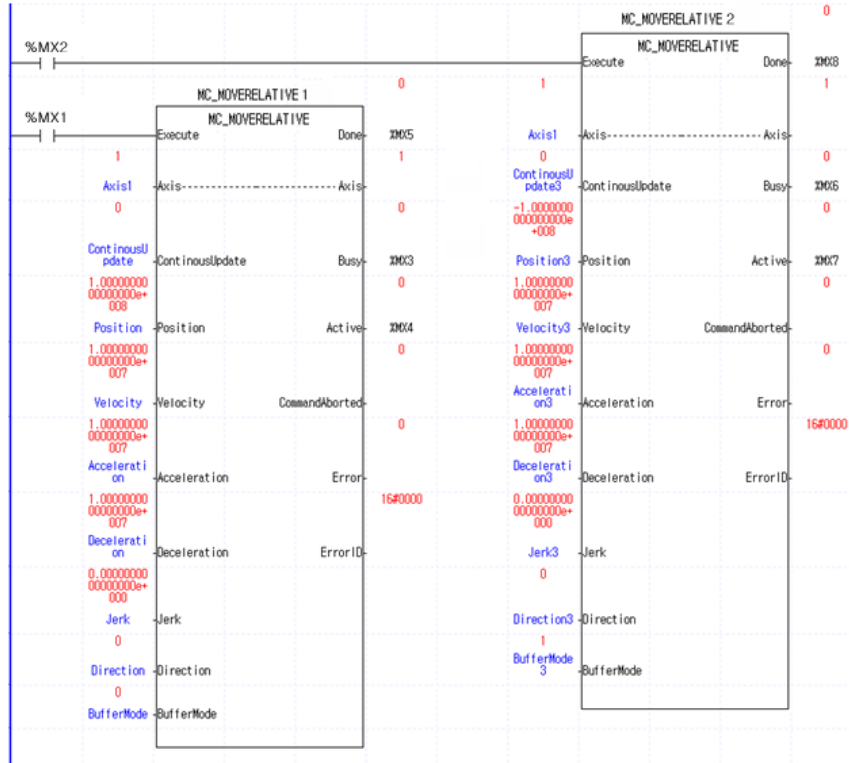
(b) 时序图



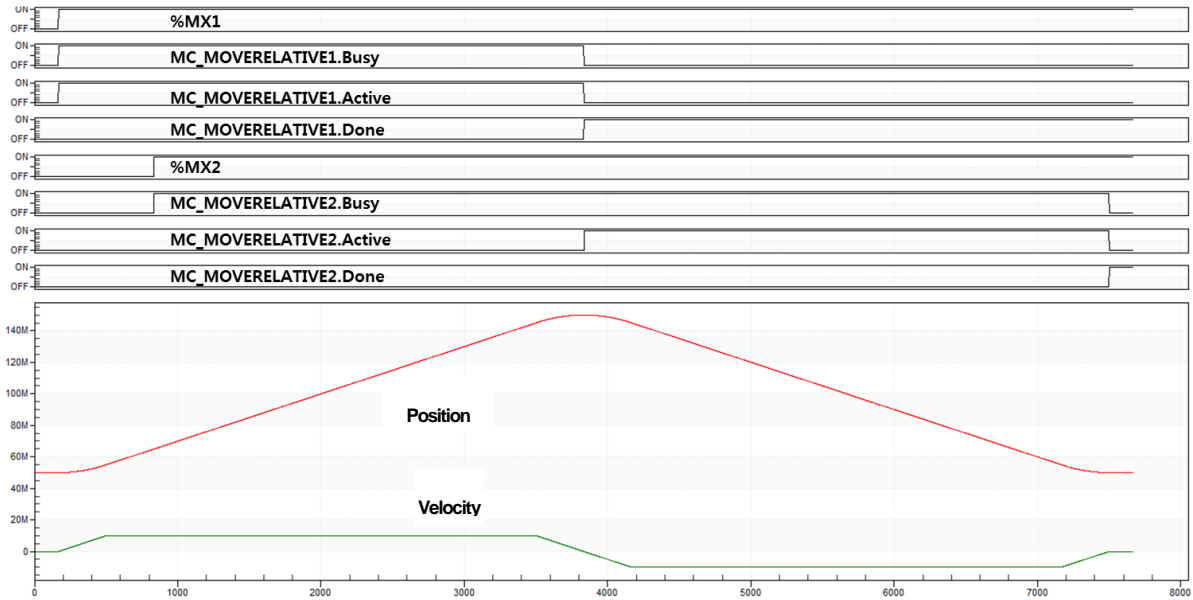
(8) 应用实例程序

该实例显示当从 50,000,000 当前指令位置移动到 150,000,000，到 50,000,000 通过设置缓存模式为 1 的另一个功能块执行。

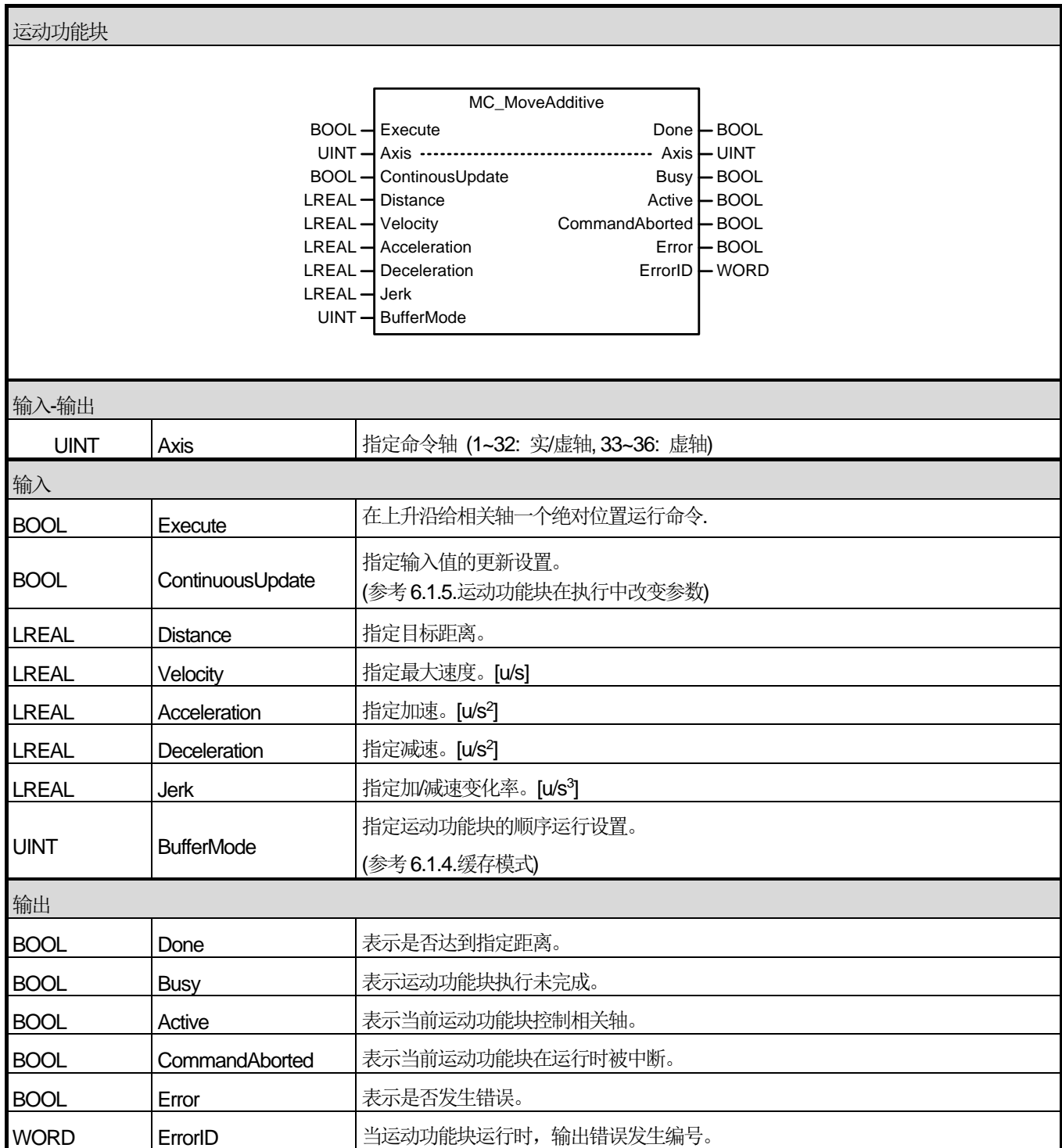
(a) 功能块设置



(b) 时序图



6.3.7 增加位控运行(MC_MoveAdditive)

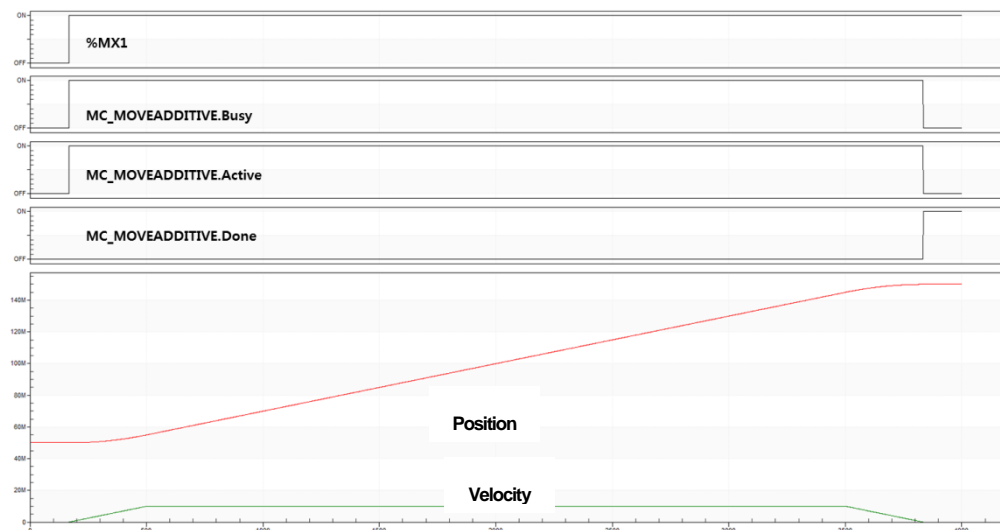


- (1) 该运动功能块给相关增加位控运行命令。
- (2) 增加位置运动 (MC_MoveAdditive) 表示增加的运动功能块移动到指定在距离输入中从运动功能块最终目标位置或最近执行'DiscreteMotion'状态位置。如果当前轴执行运动功能块为 'ContinuousMotion'状态，基于增加位置运动的位置执行 (MC_MoveAdditive) 。
- (3) 移动方向取决于目标距离的指示，正 (+ 或 No 指示) 移动方向为正向，负 (-) 移动方向为反向。
- (4) 当前运动功能块执行后，没有运动功能块到达目标位置，'Done' 输出为 On。
- (5) 当运动功能块正在运行时，轴为'DiscreteMotion'状态，当运行完成转换为'Standstill'状态。
- (6) 指令完成前通过再次执行功能块(执行输入为On)应用变更参数. 仅可以更新位置,速率,加速,减速,加速度输入。
- (7) 实例程序
该实例显示通过对应设置值(100,000,000)移动距离从当前位置 50,000,000 移动到 150,000,000.

(a) 功能块设置



(b) 时序图

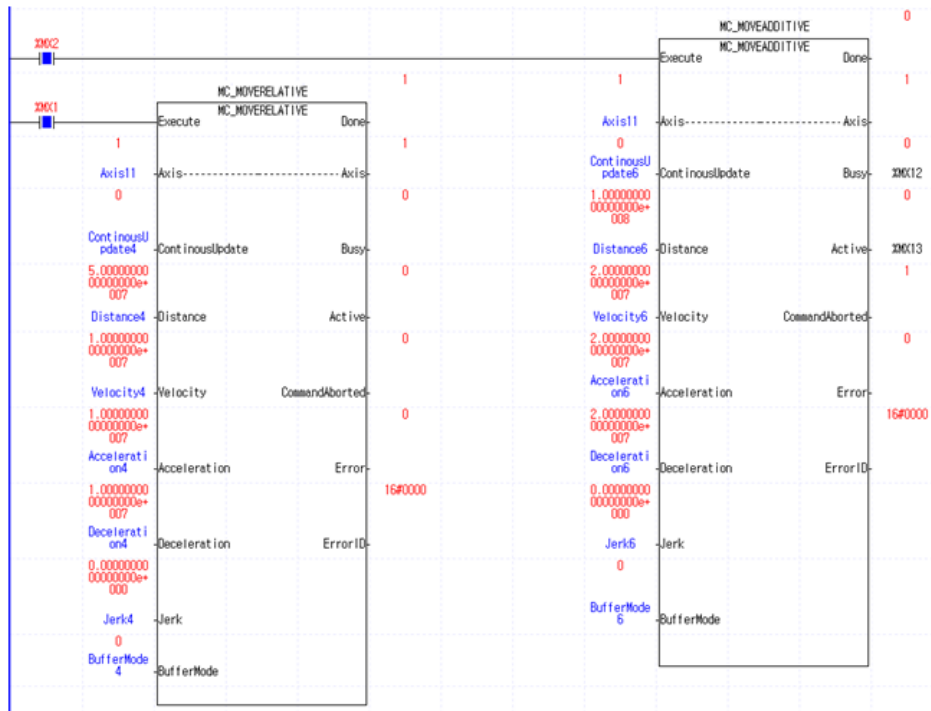


第 6 章运动功能块

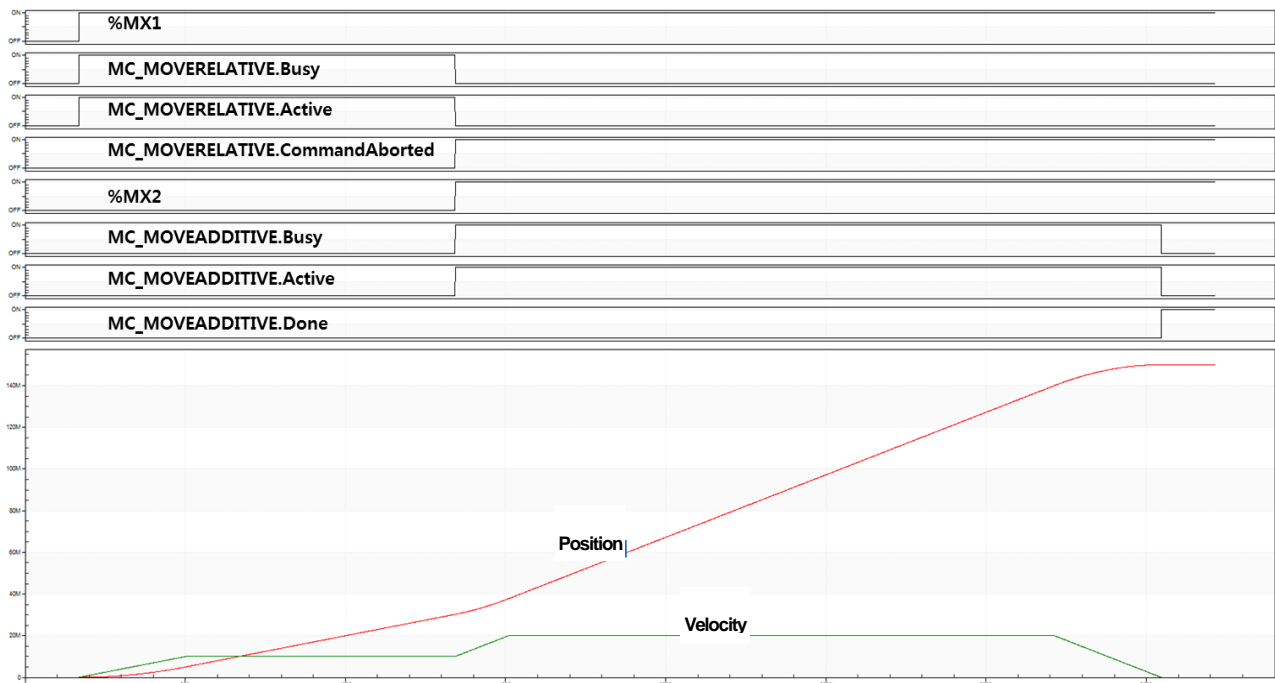
(8) 应用实例程序

该实例当前指令位置 0 移动到 50,000,000，移动一个额外 100,000,000 到 150,000,000 时执行 MC_MOVEADDITIVE 功能块。

(a) 功能块设置



(b) 时序图



6.3.8 定速运行 (MC_MoveVelocity)

运动功能块																													
<div style="border: 1px solid black; padding: 10px; margin: 0 auto; width: fit-content;"> <p style="text-align: center; margin: 0;">MC_MoveVelocity</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%; border-right: 1px solid black; padding: 2px;">BOOL Execute</td> <td style="width: 33%; padding: 2px;">InVelocity</td> <td style="width: 33%; border-right: 1px solid black; padding: 2px;">BOOL</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">UINT Axis</td> <td style="padding: 2px;">Axis</td> <td style="border-right: 1px solid black; padding: 2px;">UINT</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">BOOL ContinuousUpdate</td> <td style="padding: 2px;">Busy</td> <td style="border-right: 1px solid black; padding: 2px;">BOOL</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">LREAL Velocity</td> <td style="padding: 2px;">Active</td> <td style="border-right: 1px solid black; padding: 2px;">BOOL</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">LREAL Acceleration</td> <td style="padding: 2px;">CommandAborted</td> <td style="border-right: 1px solid black; padding: 2px;">BOOL</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">LREAL Deceleration</td> <td style="padding: 2px;">Error</td> <td style="border-right: 1px solid black; padding: 2px;">BOOL</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">LREAL Jerk</td> <td style="padding: 2px;">ErrorID</td> <td style="border-right: 1px solid black; padding: 2px;">WORD</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">UINT Direction</td> <td></td> <td style="border-right: 1px solid black;"></td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">UINT BufferMode</td> <td></td> <td style="border-right: 1px solid black;"></td> </tr> </table> </div>			BOOL Execute	InVelocity	BOOL	UINT Axis	Axis	UINT	BOOL ContinuousUpdate	Busy	BOOL	LREAL Velocity	Active	BOOL	LREAL Acceleration	CommandAborted	BOOL	LREAL Deceleration	Error	BOOL	LREAL Jerk	ErrorID	WORD	UINT Direction			UINT BufferMode		
BOOL Execute	InVelocity	BOOL																											
UINT Axis	Axis	UINT																											
BOOL ContinuousUpdate	Busy	BOOL																											
LREAL Velocity	Active	BOOL																											
LREAL Acceleration	CommandAborted	BOOL																											
LREAL Deceleration	Error	BOOL																											
LREAL Jerk	ErrorID	WORD																											
UINT Direction																													
UINT BufferMode																													
输入-输出																													
UINT	Axis	指定命令轴 (1~32: 实/虚轴, 33~36: 虚轴)																											
输入																													
BOOL	Execute	在上升沿给相关轴一个绝对位置运行命令。																											
BOOL	ContinuousUpdate	指定输入值的更新设置。 (参考 6.1.5.运动功能块在执行中改变参数)																											
LREAL	Velocity	指定最大速度。[u/s]																											
LREAL	Acceleration	指定加速。[u/s ²]																											
LREAL	Deceleration	指定减速。[u/s ²]																											
LREAL	Jerk	指定加/减速变化率。[u/s ³]																											
UINT	Direction	指定运行速度。(1 ~ 3: 1-正向, 2-反向, 3-当前方向)																											
UINT	BufferMode	指定运动功能块的顺序运行设置。 (参考 6.1.4.缓存模式)																											
输出																													
BOOL	InVelocity	表示是否到达指定速度。																											
BOOL	Busy	表示运动功能块执行未完成。																											
BOOL	Active	表示当前运动功能块控制相关轴。																											
BOOL	CommandAborted	表示当前运动功能块在运行时被中断。																											
BOOL	Error	表示是否发生错误。																											
WORD	ErrorID	当运动功能块运行时, 输出错误发生编号。																											

- (1) 该运动功能块给相关轴定速运行命令。
- (2) 给停止命令或执行其它允许中断的定速运动。
- (3) 在速度输入中指定运行速度。运行速度值的正符号 (+ 或 No 符号)为正向, 负(-)符号表示反向。
- (4) 在方向输入中指定运行方向。但是, 通过速度输入的定速值的符号影响运行方向。例如, 如果速度值指定为负数, 同时方向输入反向, 相关轴仍为正向运行。
- (5) 当相关轴的速度到达指定速度时, 输出 InVelocity 为 On , 当定速运行中断为 Off 。
- (6) 当运动功能块为运行时, 轴为'ContinuousMotion'状态。

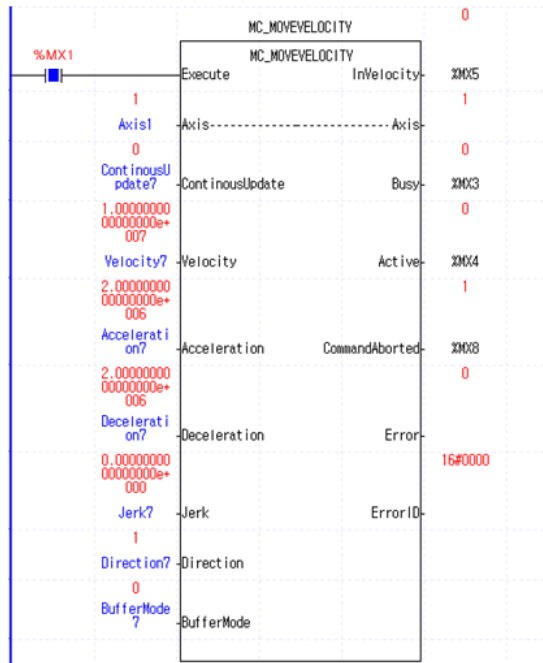
第 6 章运动功能块

(7) 指令完成前通过再次执行功能块(执行输入为On)应用变更参数. 仅可以更新位置,速率,加速,减速,加速度,方向输入.

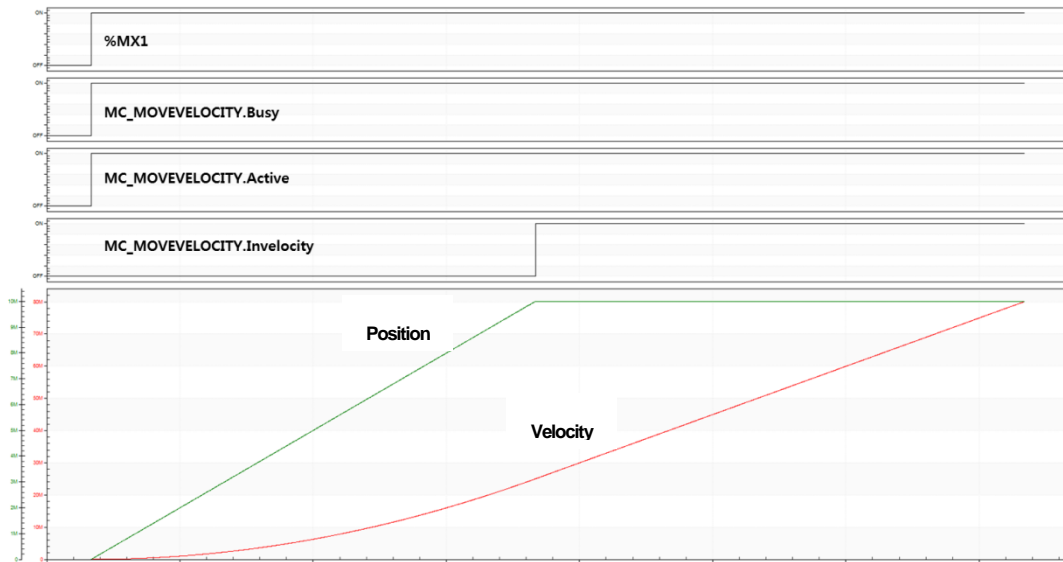
(8) 实例程序

该实例程序显示速度 10,000,000 的移动. 一旦达到设置速率, InVelocity 输出为 on.

(a) 功能块设置



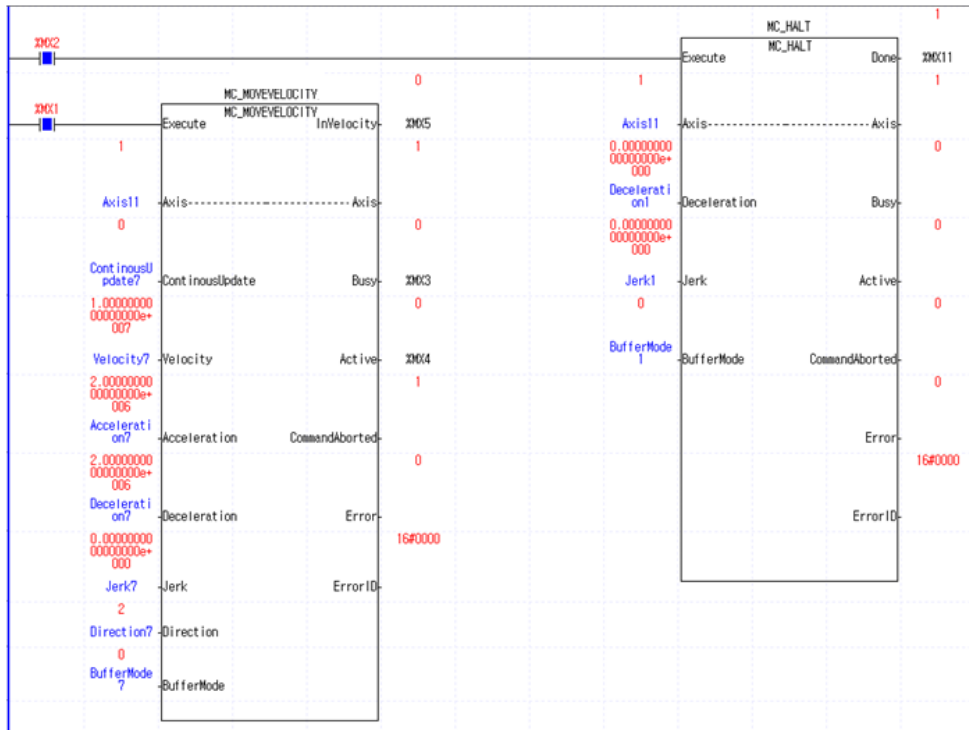
(b) 时序图



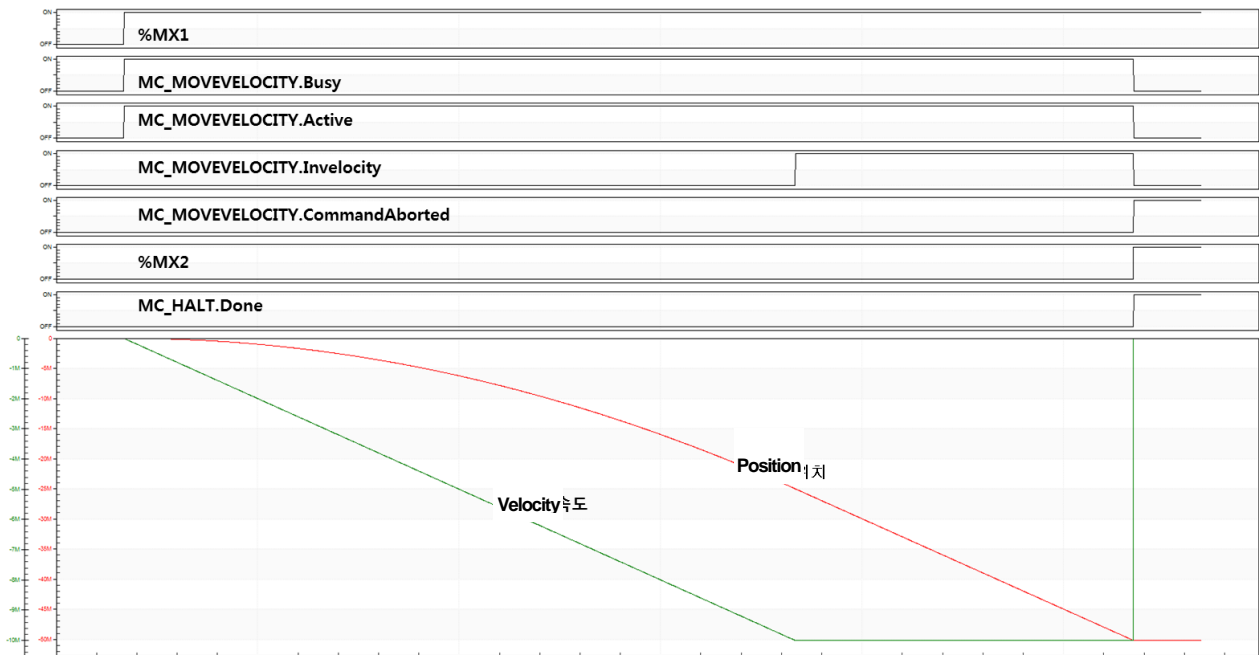
(9) 应用实例程序

该实例程序显示停止运行导致 MC-Halt 功能块执行, 当以速度 10,000,000 反向移动时.

(a) 功能块设置



(b) 时序图



6.3.9 绝对位置运行以定速运行结束

(MC_MoveContinuousAbsolute)

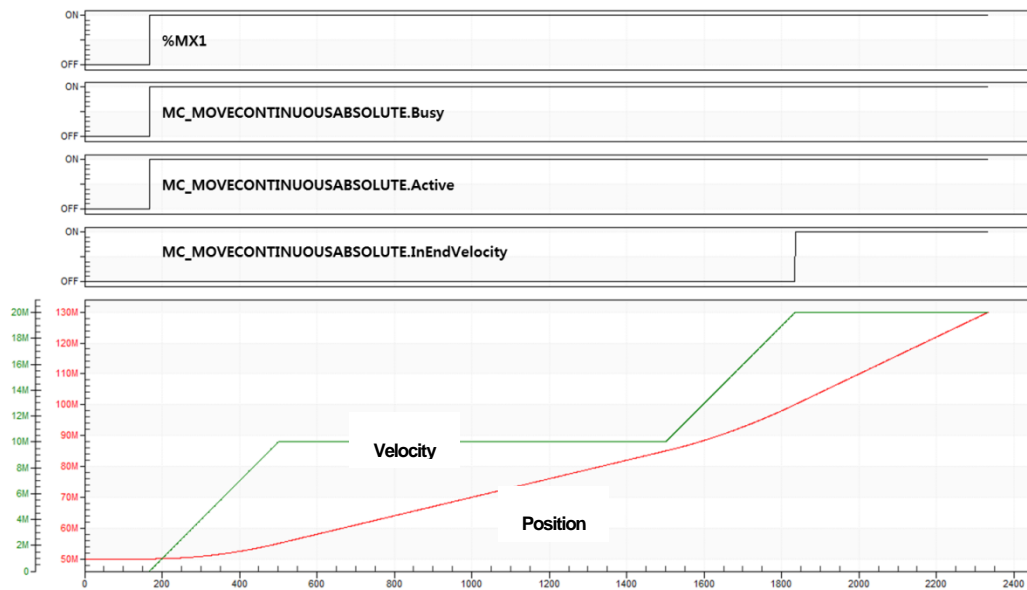
运动功能块																																			
<div style="border: 1px solid black; padding: 10px; width: fit-content; margin: 0 auto;"> <p style="text-align: center;">MC_MoveContinuousAbsolute</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">BOOL — Execute</td> <td style="width: 33%;"></td> <td style="width: 33%;">InEndVelocity — BOOL</td> </tr> <tr> <td>UINT — Axis</td> <td>----- Axis</td> <td>UINT</td> </tr> <tr> <td>BOOL — ContinuousUpdate</td> <td></td> <td>Busy — BOOL</td> </tr> <tr> <td>LREAL — Position</td> <td></td> <td>Active — BOOL</td> </tr> <tr> <td>LREAL — EndVelocity</td> <td>CommandAborted</td> <td>— BOOL</td> </tr> <tr> <td>LREAL — Velocity</td> <td></td> <td>Error — BOOL</td> </tr> <tr> <td>LREAL — Acceleration</td> <td>ErrorID</td> <td>— WORD</td> </tr> <tr> <td>LREAL — Deceleration</td> <td></td> <td></td> </tr> <tr> <td>LREAL — Jerk</td> <td></td> <td></td> </tr> <tr> <td>UINT — Direction</td> <td></td> <td></td> </tr> <tr> <td>UINT — BufferMode</td> <td></td> <td></td> </tr> </table> </div>			BOOL — Execute		InEndVelocity — BOOL	UINT — Axis	----- Axis	UINT	BOOL — ContinuousUpdate		Busy — BOOL	LREAL — Position		Active — BOOL	LREAL — EndVelocity	CommandAborted	— BOOL	LREAL — Velocity		Error — BOOL	LREAL — Acceleration	ErrorID	— WORD	LREAL — Deceleration			LREAL — Jerk			UINT — Direction			UINT — BufferMode		
BOOL — Execute		InEndVelocity — BOOL																																	
UINT — Axis	----- Axis	UINT																																	
BOOL — ContinuousUpdate		Busy — BOOL																																	
LREAL — Position		Active — BOOL																																	
LREAL — EndVelocity	CommandAborted	— BOOL																																	
LREAL — Velocity		Error — BOOL																																	
LREAL — Acceleration	ErrorID	— WORD																																	
LREAL — Deceleration																																			
LREAL — Jerk																																			
UINT — Direction																																			
UINT — BufferMode																																			
输入-输出																																			
UINT	Axis	指定命令轴 (1~32: 实/虚轴, 33~36: 虚轴)																																	
输入																																			
BOOL	Execute	在上升沿给相关轴一个绝对位置运行命令																																	
BOOL	ContinuousUpdate	指定输入值的更新设置. (参考 6.1.5.运动功能块在执行中改变参数)																																	
LREAL	EndVelocity	到达目标位之后指定运行速度. [u/s]																																	
LREAL	Velocity	指定到达目标位置的最大速度. [u/s]																																	
LREAL	Acceleration	指定加速. [u/s ²]																																	
LREAL	Deceleration	指定减速. [u/s ²]																																	
LREAL	Jerk	指定加/减速变化率. [u/s ³]																																	
UINT	Direction	指定运行方向. (0~4: 0-未指定, 1-正向, 2-最短距离, 3-反向, 4-当前方向)																																	
UINT	BufferMode	指定运动功能块的顺序运行设置. (参考 6.1.4.缓存模式)																																	
输出																																			
BOOL	InEndVelocity	表示到达目标位置以定速运行.																																	
BOOL	Busy	表示运动功能块执行未完成.																																	
BOOL	Active	表示当前运动功能块控制相关轴.																																	
BOOL	CommandAborted	表示当前运动功能块在运行时被中断.																																	
BOOL	Error	表示是否发生错误.																																	
WORD	ErrorID	当运动功能块运行时, 输出错误发生编号.																																	

- (1) 当相对位置运行后，该运动功能块给相关轴定速运行命令。
- (2) 当执行 MC_MoveContinuousAbsolute，如果没有运动功能块待命，相关轴移动到指定位置以定速运行。
- (3) 给停止命令或执行其他允许中断速度运行的运动功能块。
- (4) 轴的运行方向在方向输入设置无限长度重复运行，如果无限长度重复运行设置为禁止，方向输入为忽略。当方向输入为最短距离(=2)，相关轴做无限长度重复运行自动选择可允许的最短距离方向。可用范围 0-4 (0-未指定, 1-正向, 2-最短距离, 3-反向, 4-当前方向)，如果超出范围发生 "错误 0x1017"。
- (5) 当相关轴到达指定位置后，相关轴开始速度运行时，输出 InEndVelocity 为 on，当指定运行中断时为 Off。
- (6) 当执行命令时，轴为'ContinuousMotion'状态。
- (7) 指令完成前通过再次执行功能块(执行输入为On)应用变更参数。仅位置，结束速率，速率，加速，减速，加速度，方向输入可以更新。(然而，如果InEndVelocity=On，只反映到结束速率输入。
- (8) 实例程序
该实例程序显示从指令位置 50,000,000 到位置 100,000,000 移动后以速度 20,000,000 运行。一旦到达设置位置，InEndVelocity 输出为 on。
- (a) 功能块设置



第 6 章运动功能块

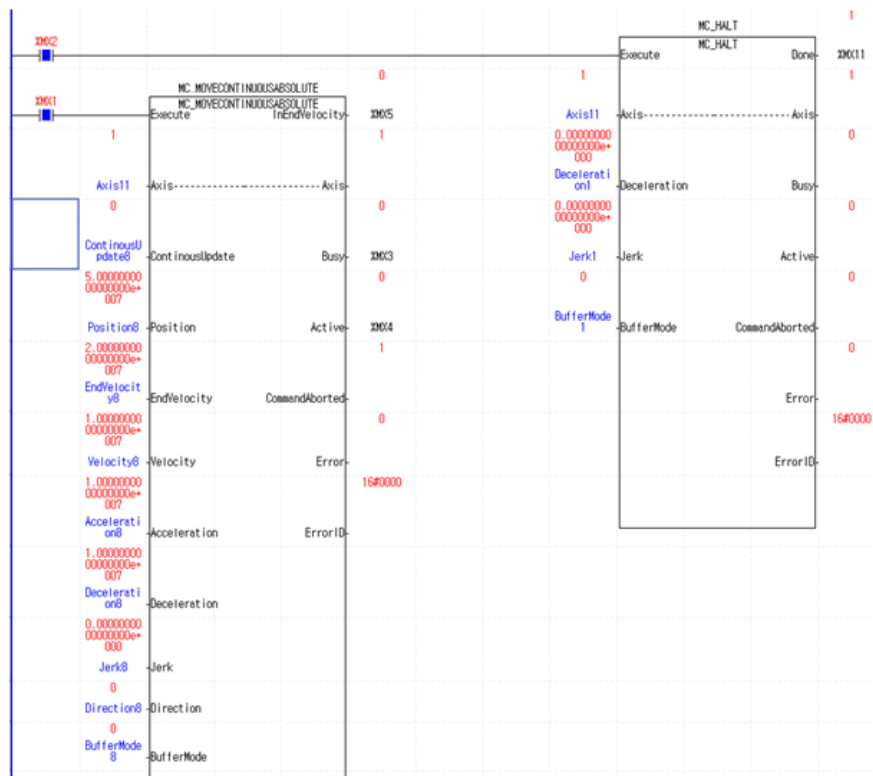
(b) 时序图



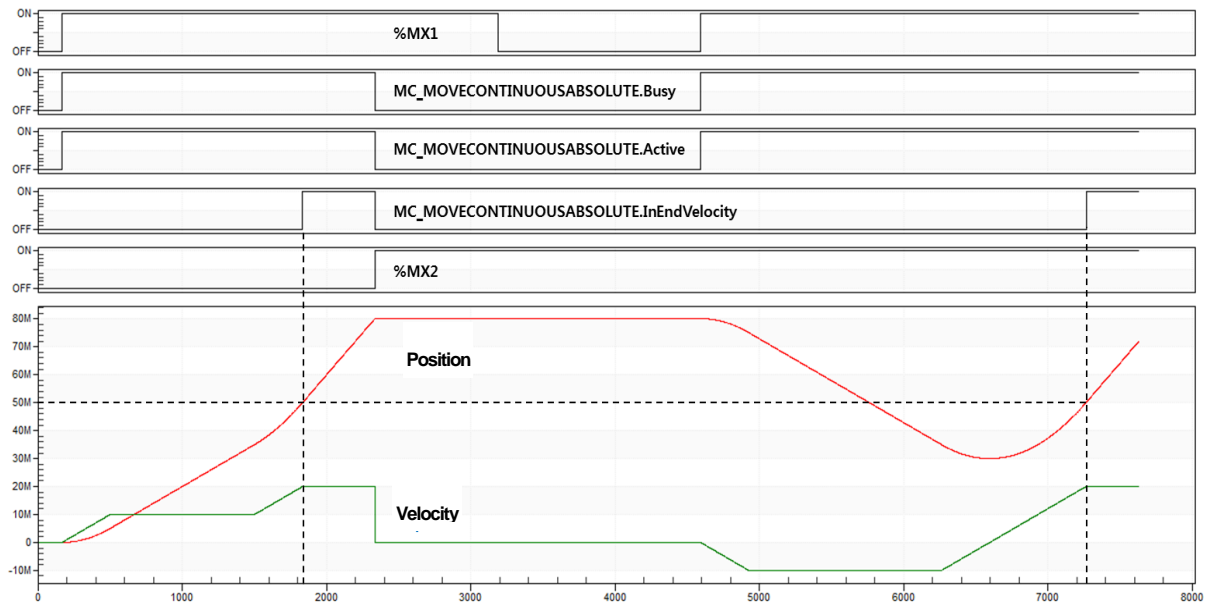
(9) 应用实例程序

该实例程序显示 MC-Halt 功能块停止执行后当再次执行功能块时，以相同速度移动，从当前指令位置 0 到 50,000,000，然后以速度 20,000,000 运行。

(a) 功能块设置



(b) 时序图



6.3.10 相对位置运行以定速结束

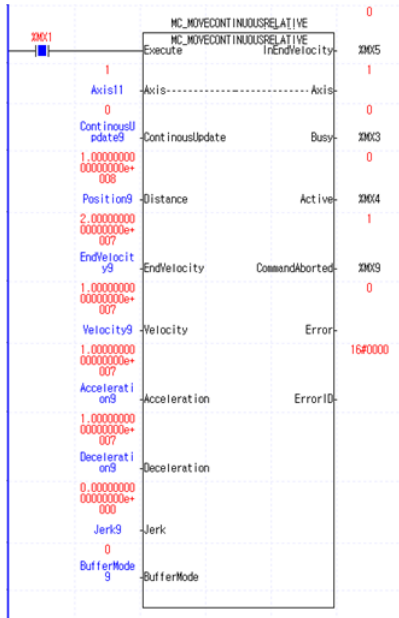
(MC_MoveContinuousRelative)

运动功能块		
<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="width: 30%;"> <p>BOOL — Execute</p> <p>UINT — Axis</p> <p>BOOL — ContinuousUpdate</p> <p>LREAL — Distance</p> <p>LREAL — EndVelocity</p> <p>LREAL — Velocity</p> <p>LREAL — Acceleration</p> <p>LREAL — Deceleration</p> <p>LREAL — Jerk</p> <p>UINT — BufferMode</p> </div> <div style="width: 40%; border: 1px solid black; padding: 5px; text-align: center;"> <p>MC_MoveContinuousRelative</p> </div> <div style="width: 30%;"> <p>InEndVelocity — BOOL</p> <p>Axis — UINT</p> <p>Busy — BOOL</p> <p>Active — BOOL</p> <p>CommandAborted — BOOL</p> <p>Error — BOOL</p> <p>ErrorID — WORD</p> </div> </div>		
输入-输出		
UINT	Axis	指定命令轴 (1~32: 实/虚轴, 33~36: 虚轴)
输入		
BOOL	Execute	在上升沿给相关轴绝对位置运动命令.
BOOL	ContinuousUpdate	指定输入值的更新设置. (参考 6.1.5.运动功能块在执行中改变参数)
LREAL	Distance	指定目标距离.
LREAL	EndVelocity	到达目标位之后指定运行速度. [u/s]
LREAL	Velocity	指定到达目标位置的最大速度. [u/s]
LREAL	Acceleration	指定加速. [u/s ²]
LREAL	Deceleration	指定减速. [u/s ²]
LREAL	Jerk	指定加/减速变化率. [u/s ³]
UINT	BufferMode	指定运动功能块的顺序运行设置. (参考 6.1.4.缓存模式)
输出		
BOOL	InEndVelocity	表示到达目标位置以定速运行.
BOOL	Busy	表示运动功能块执行未完成.
BOOL	Active	表示当前运动功能块控制相关轴.
BOOL	CommandAborted	表示当前运动功能块在运行时被中断.
BOOL	Error	表示是否发生错误.
WORD	ErrorID	当运动功能块运行时, 输出错误发生编号.

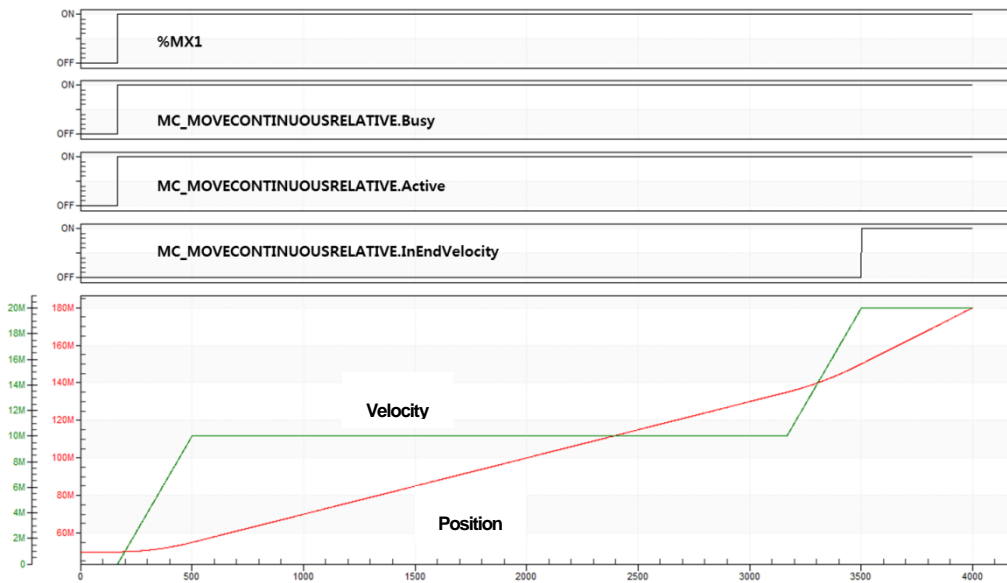
- (1) 该运动功能块给相关轴 MC_MoveContinuousRelative 命令。
- (2) 当执行 MC_MoveContinuousRelative, 当移动到指定距离如果没有运动功能块待命, 相关轴定速运行。
- (3) 给停止命令或执行其他允许中断定速运行的运动功能块。
- (4) 当相关轴开始速度运行达到指定速度后, 输入 InEndVelocity 为 On。当定速运行中断为 Off。
- (5) 当轴运行时, 轴为 'ContinuousMotion'状态。

- (6) 指令完成前通过再次执行功能块(执行输入为On)应用变更参数. 仅距离, 结束速率, 速率, 加速, 减速, 加速度输入可以更新. (然而, InEndVelocity=On, 仅反映EndVelocity 输入.)
- (7) 实例程序
该实例程序显示通过移动与设置值相对应的值(100,000,000), 从当前位置 50,000,000 到 150,000,000 之后, 以速度 20,000,000 运行.

(a) 功能块设置



(b) 时序图

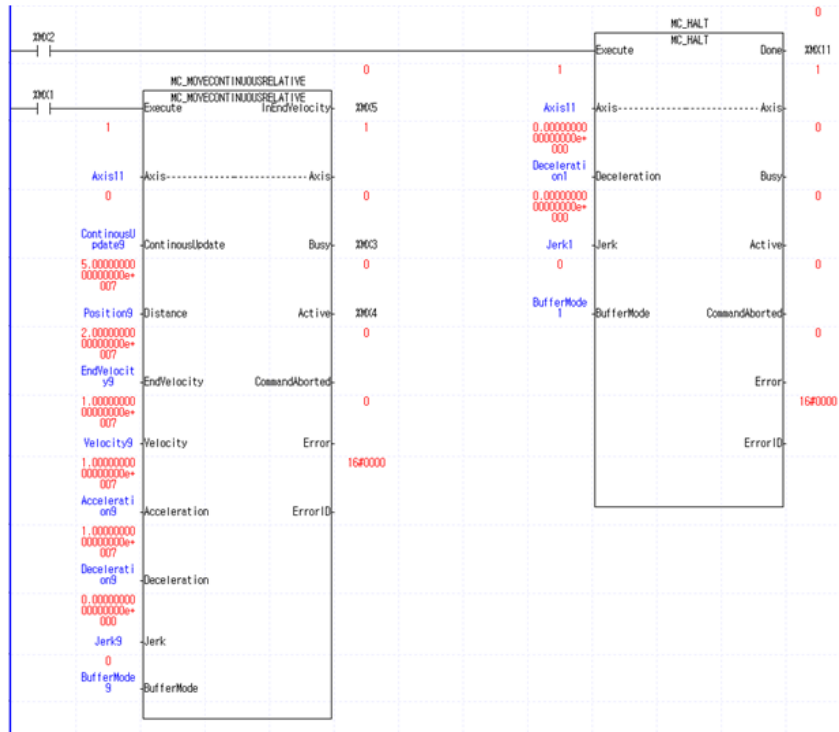


第 6 章运动功能块

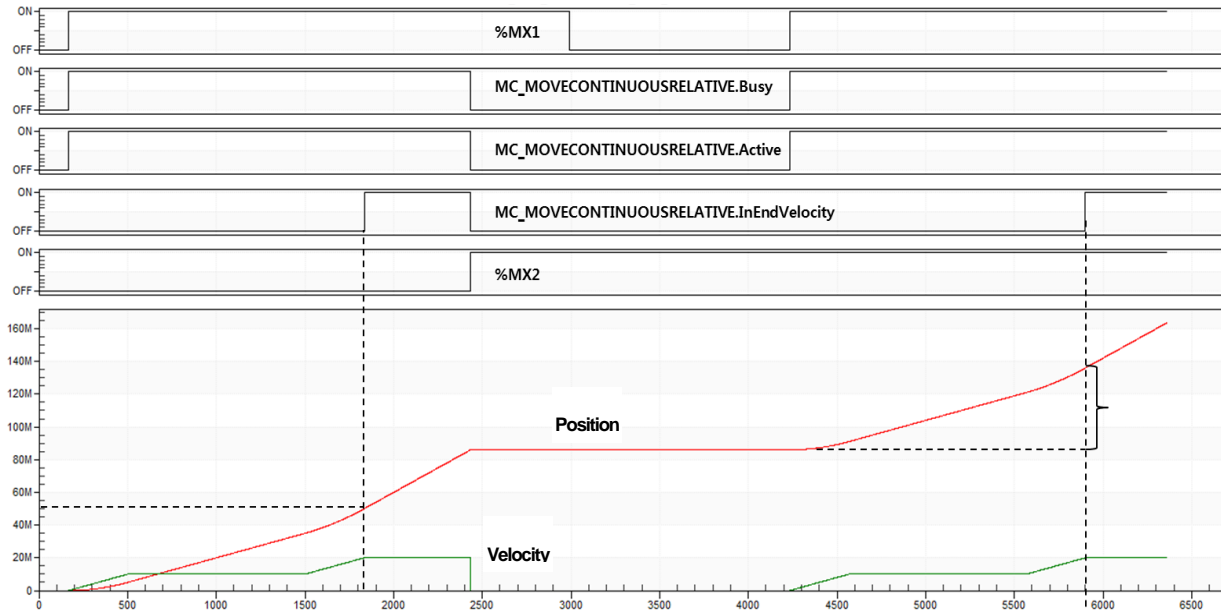
(8) 应用实例程序

该实例程序显示以速度 20,000,000 移动从位置 0 移动到 50,000,000, 然后以速度 20,000,000 运行, 通过执行 MC_Halt 功能块停止, 通过再次执行功能块移动到相同位置(20,000,000).

(a) 功能块设置



(b) 时序图



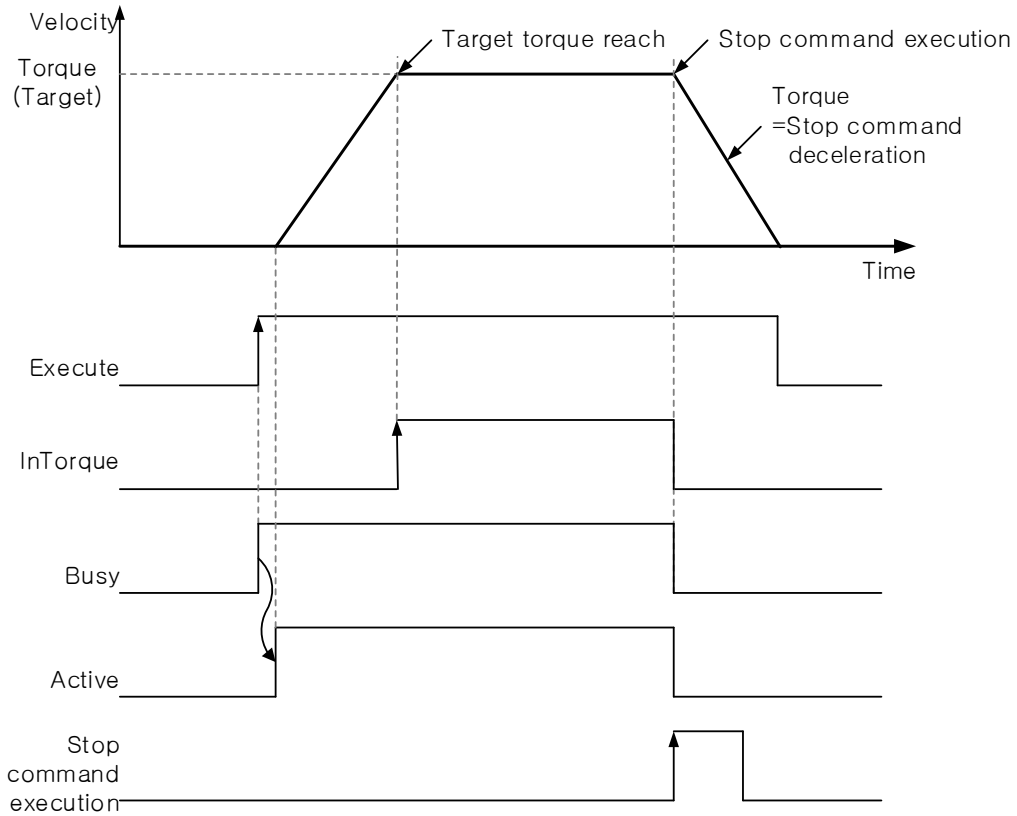
6.3.11 转矩控制(MC_TorqueControl)

运动功能块		
<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center; margin: 0;">MC_TorqueControl</p> <p>BOOL — Execute</p> <p>UINT — Axis</p> <p>BOOL — ContinuousUpdate</p> <p>LREAL — Torque</p> <p>LREAL — TorqueRamp</p> <p>LREAL — Velocity</p> <p>LREAL — Acceleration</p> <p>LREAL — Deceleration</p> <p>LREAL — Jerk</p> <p>UINT — Direction</p> <p>UINT — BufferMode</p> </div> <div style="border: 1px solid black; padding: 5px;"> <p>InTorque — BOOL</p> <p>Axis — UINT</p> <p>Busy — BOOL</p> <p>Active — BOOL</p> <p>CommandAborted — BOOL</p> <p>Error — BOOL</p> <p>ErrorID — WORD</p> </div> </div>		
输入-输出		
UINT	Axis	指定命令轴 (1~32: 实轴)
输入		
BOOL	Execute	在上升沿给相关轴一个绝对位置运行命令.
BOOL	ContinuousUpdate	指定输入值的更新设置. (参考 6.1.5.运动功能块在执行中改变参数)
LREAL	Torque	指定目标转矩 [u]
LREAL	TorqueRamp	指定转矩的上升斜率 [u/s]
LREAL	Velocity	未使用
LREAL	Acceleration	未使用
LREAL	Deceleration	未使用
LREAL	Jerk	未使用
UINT	Direction	指定运行方向. (1~2 : 1-正向, 2-反向)
UINT	BufferMode	指定运动功能块的顺序运行设置. (参考 6.1.4.缓存模式)
输出		
BOOL	InTorque	表示输入转矩值和当前运行转矩值相同.
BOOL	Busy	表示运动功能块执行未完成.
BOOL	Active	表示当前运动功能块控制相关轴.
BOOL	CommandAborted	表示当前运动功能块在运行时被中断.
BOOL	Error	表示是否发生错误.
WORD	ErrorID	当运动功能块运行时, 输出错误发生编号.

- (1) 该运动功能块给相关轴转矩控制命令。
- (2) 当执行转矩控制 (MC_Torque), 相关轴执行控制保持 Torque 输入中的转矩值。
- (3) 给停止命令或执行其他允许中断定速运行的运动功能块。
- (4) 指定斜度到达 TorqueRamp 输入中的目标转矩值。

第 6 章运动功能块

- (5) 在 **Direction** 输入中指定运行方向。当设置值超出范围执行运动功能块, **Error** 为 **On**, 出现错误代码 **ErrorID "0x1017"**。
- (6) 当相关轴到达指定转矩时, 输出 **InTorque** 为 **On**, 当转矩控制运行中断, 将为 **Off**。
- (7) 当该运动功能块运行时, 轴为 **ContinuousMotion** 状态。
- (8) 指令完成前通过再次执行功能块(执行输入为 **On**)应用变更参数. 仅转矩, 转矩斜坡, 输入方向可以更新。
- (9) 时序图



6.3.12 设置当前位置(MC_SetPosition)



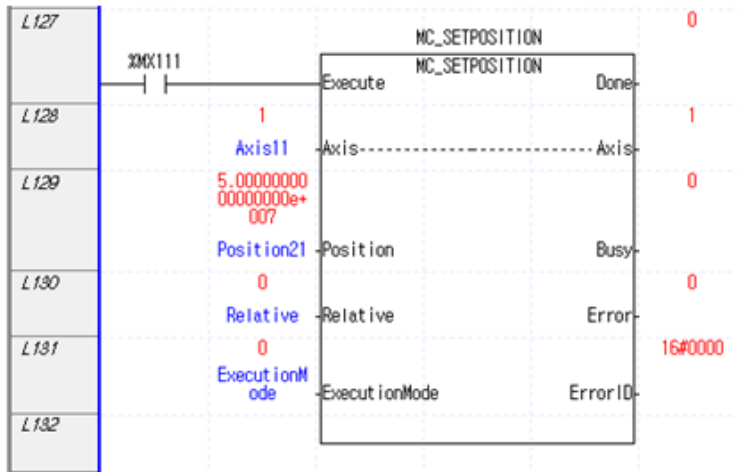
- (1) 该运动功能块设置相关轴的当前位置。
- (2) 在 Position 输入中指定位置。当执行运动功能块时, 如果 Relative 输入为 Off, 相关轴的位置被 Position 输入的值替换, 如果 Relative 输入为 On, Position 输入的值为相关轴增加到当前位置。
- (3) ExcutionMode 输入指定设置点。0 表示运动功能块执行后立即设置, 1 表示'缓存'在顺序运行设置的相同点设置。
"error0x101B"时不能设置值。
0 (mcImmediatly): 当执行功能块后立即改变参数值 (在执行输入上升沿)。如果相关轴在运行中, 运行受影响。
1 (mcQueued):在缓存模式'缓存'相同点改变。(参考 6.1.4 缓存模式输入)

第 6 章运动功能块

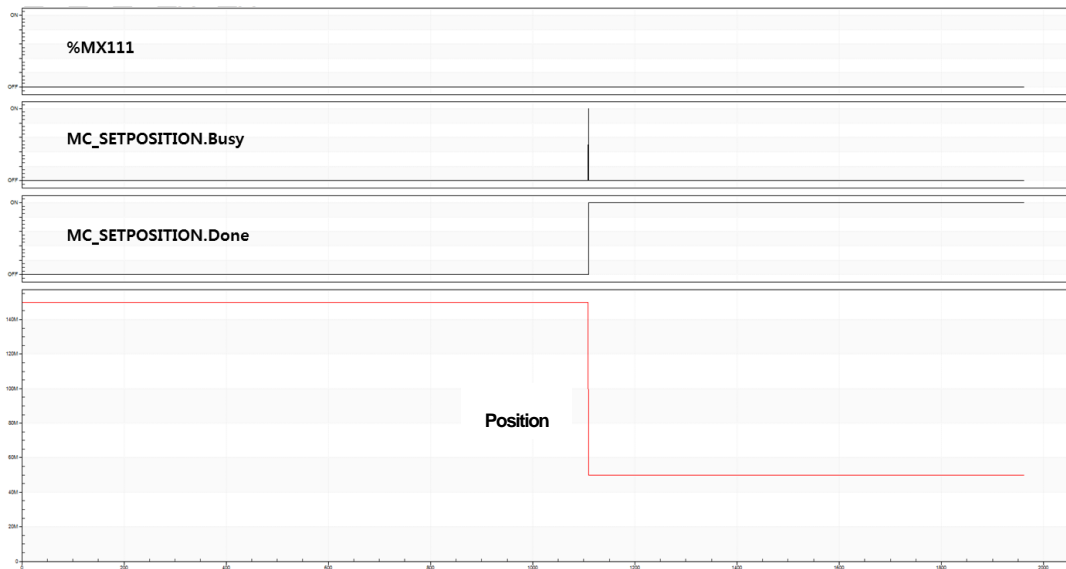
(4) 实例程序

该实例程序显示通过增加一个对应设置值(50,000,000)的相对位置(Relative=1)从位置 150,000,000 设置为 200,000,000.

(a) 功能块设置



(b) 时序图



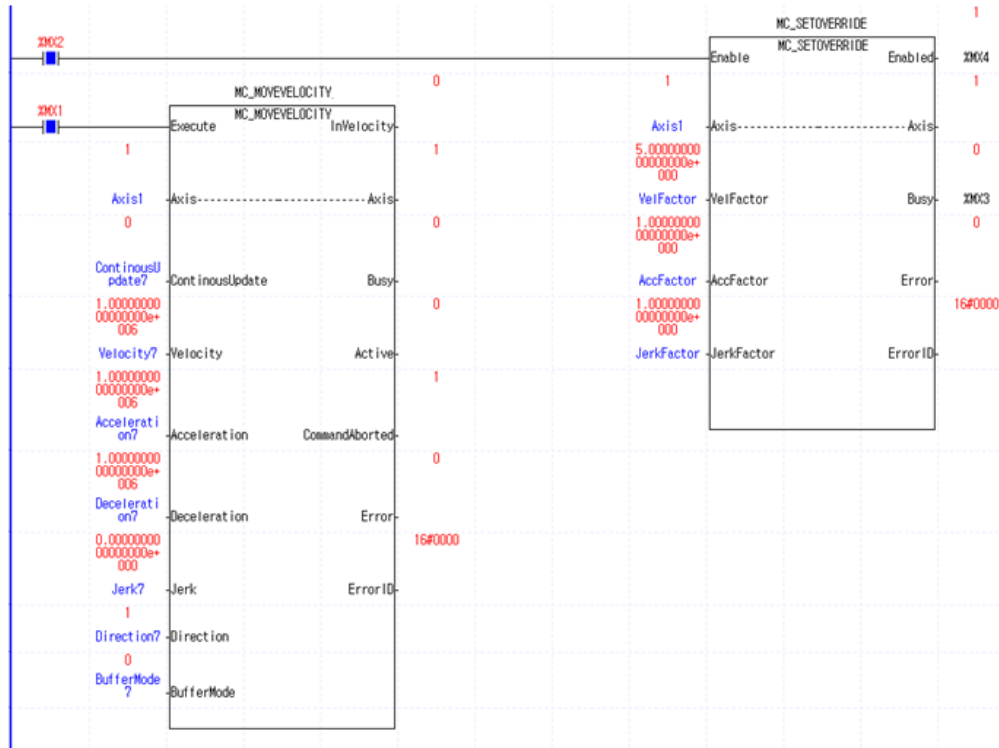
6.3.13 速度/加速覆盖 (MC_Set 覆盖)



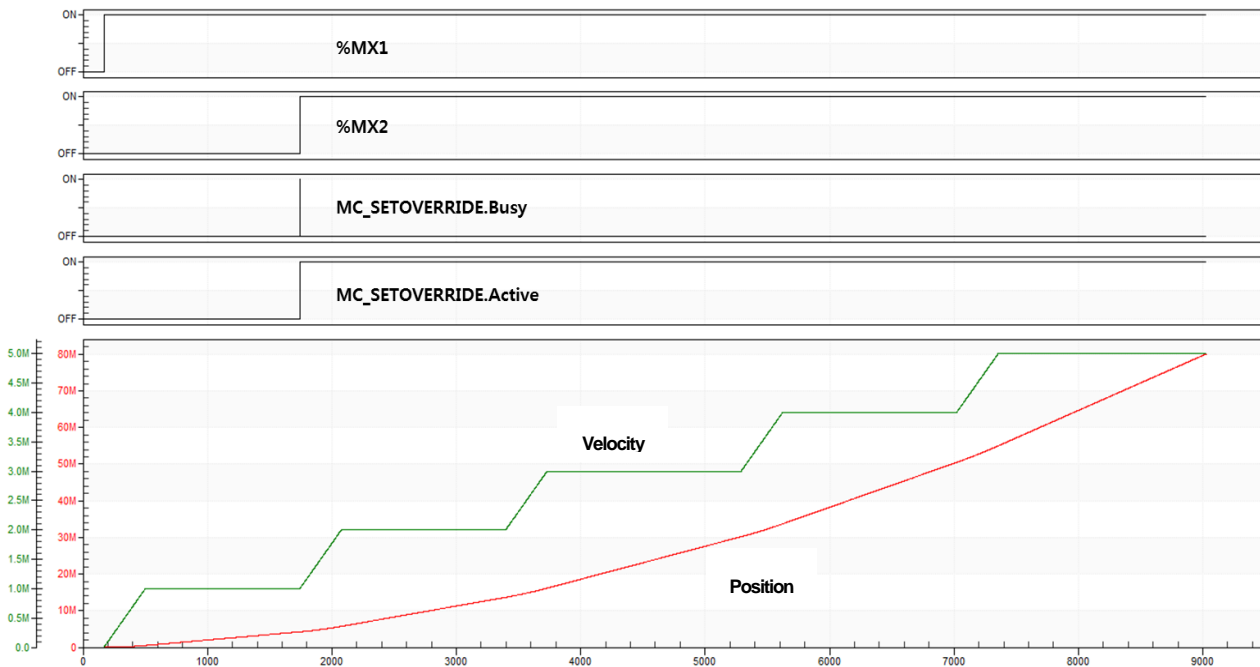
- (1) 该运动功能块是相关轴速度的覆盖，加速，加速变化率。
- (2) 覆盖率应用于指定相关轴，当 Enable 输入为 On 改变。如果 Enable 输入为 Off，在 Off 状态前，覆盖率不变。
- (3) 速度覆盖率在 VelFactor 输入中指定。如果指定值为 0.0，相关轴停止，但是'StandStill'状态不变化。
- (4) 在 AccFactor 和 JerkFactor 输入中分别指定加减速和 jerk (加速变化率) 的覆盖率。
- (5) Facotr 中不能输入负值，如果输入出现"error 0x10C1"。
- (6) 覆盖率的默认值为 1.0，表示当前运行功能块常规速度的 100%。
- (7) 覆盖运行不影响相关轴的几个轴。
- (8) 实例程序
 该实例显示如果在当前速率 1,000,000 状态下 VelFactor 变更为 2/3/4/5，当前运行速率变更为 2,000,000/ 3,000,000/ 4,000,000/ 5,000,000。

第 6 章运动功能块

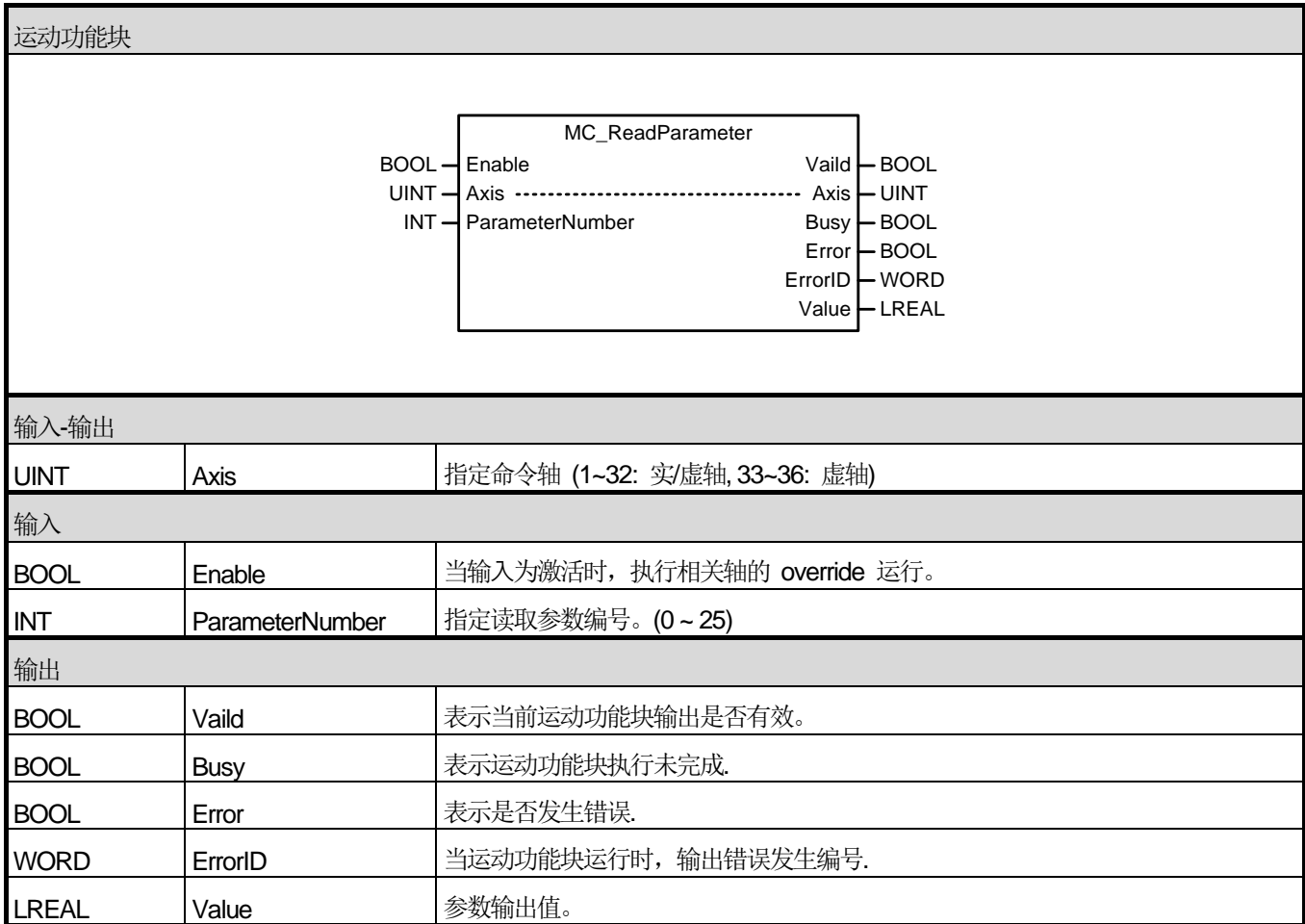
(a) 功能块设置



(b) 时序图



6.3.14 读参数 (MC_ReadParameter)



- (1) 该命令是运动功能块相关轴的输出参数。
- (2) 当 Enable 输入为 On 时, 相关参数的值在 Value 中持续输出。
- (3) 在 ParameterNumber 输入中指定需要读取的参数号。
- (4) 参数号如下所示。

No	参数	项目	描述
0	基本参数	单位	0:pulse, 1:mm, 2:inch, 3:degree
1		每转脉冲	1 ~ 4,294,967,295 [pulse]
2		每转距离	0.000000001 ~ 4,294,967,295 [Unit]
3		速度命令单位	0:Unit/Time, 1:rpm
4		速度限制	长实数型正数[Unit/s, rpm] (改变单位, 脉冲, 每转脉冲, 每转距离速度命令单位)
5		紧急停止	0 或长实数型正数[Unit/s ²]
6		编码器选择	0:增量编码器, 1:绝对编码器
7		齿轮齿数比(电机)	1 ~ 65,535
8		齿轮齿数比(机器)	1 ~ 65,535
9	反转运行模式	0:E.Stop, 1:Stop	
10	扩展参数	S/W 上限	长实数型 [Unit]
11		S/W 下限	长实数型[Unit]

第 6 章运动功能块

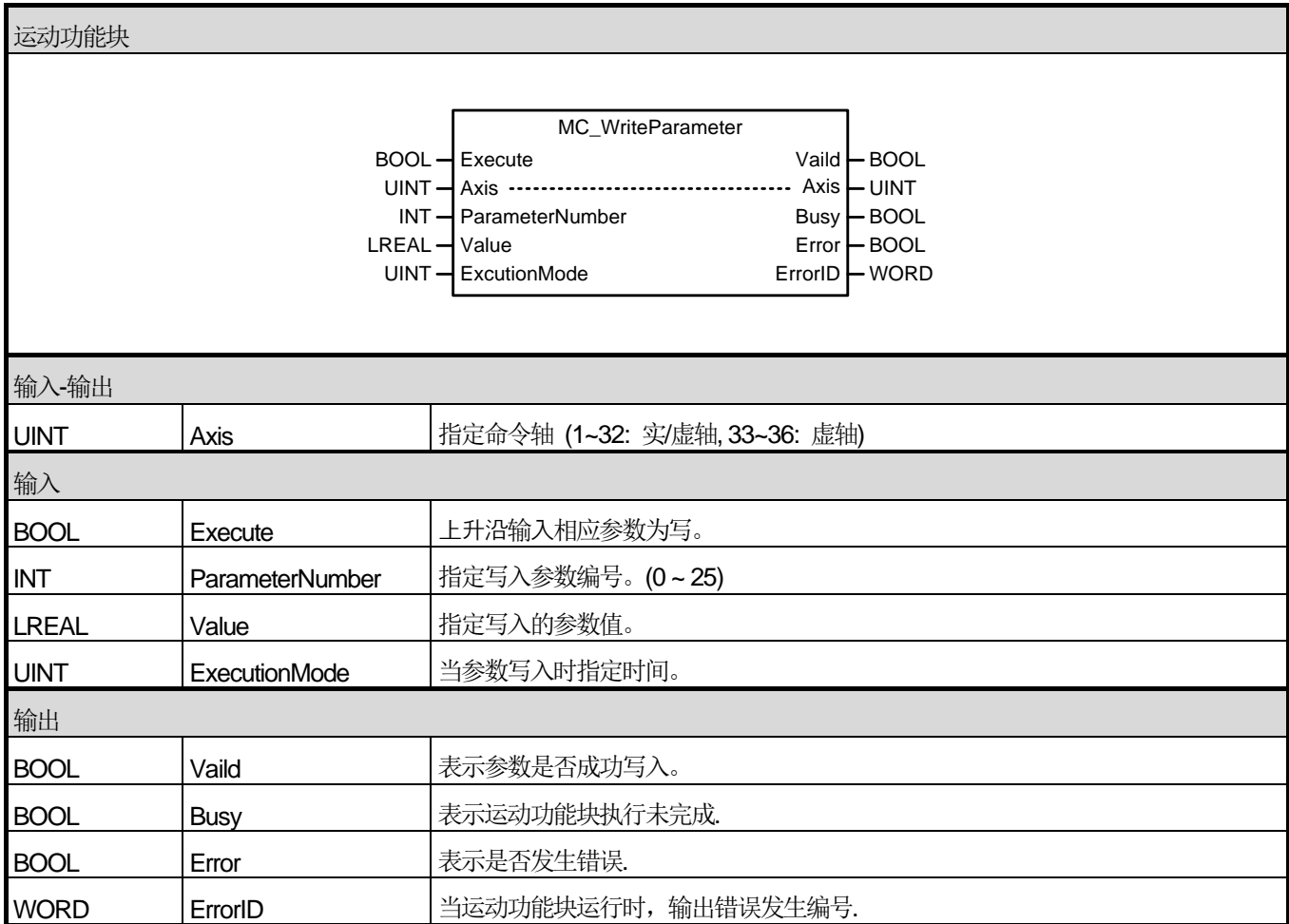
12	无限重复运行位置	长实数型正数[Unit]
13	无限重复运行	0:禁止, 1:使能
14	Inposition 范围命令	0 或长实数型正数[Unit]
15	跟踪误差超出范围值	0 或长实数型正数[Unit]
16	当前位置补偿量	0 或长实数型正数[Unit]
17	当前速度滤波时间常量	0 ~ 100
18	错误复位监控时间	1 ~ 1000 [ms]
19	速度控制中 SW 限制	0:不检测, 1:检测
20	错误跟踪电平	0:警告, 1:报警
21	JOG 高速	长实数型正数[Unit] (Jog 低速~速度限制) [Unit/s]
22	JOG 低速	长实数型正数[Unit] (< Jog 高速) [Unit/s]
23	JOG 加速	0 或长实数型正数[Unit/ s ²]
24	JOG 减速	0 或长实数型正数[Unit/ s ²]
25	JOG 加速度	0 或长实数型正数[Unit/ s ²]
26	覆盖模式	0: 比率指定, 1: 单位指定

No	参数	项目	描述
27	NC 参数	达到主轴旋转指令速度的识别范围	0~100%
28		达到主轴旋转 0 速度的识别 RPM	0~100rpm
100	编码器参数	编码器 1 单位	0: pulse, 1: mm, 2: inch, 3:degree
101		编码器 1 每旋转脉冲	1 ~ 4294967295
102		编码器 1 每旋转距离	0.000000001 ~ 4294967295
103		编码器 1 脉冲输入	0:CW/CCW 1 倍数, 1:PULSE/DIR 1 倍数 2:PULSE/DIR 2 倍数, 3:PHASE A/B 1 倍数 4:PHASE A/B 2 倍数, 5: PHASE A/B 4 倍数
104		编码器 1 最大值	(编码器 1 最小值+ 1) ~ 2147483647
105		编码器 1 最小值	-2147483648 ~ (编码器 1 最大值 - 1)
106		编码器 1 输入滤波值	0: 未使用, 1: 500kPPS 2: 200kPPS, 3: 100kPPS 4: 10kPPS, 5: 1kPPS 6: 0.1kPPS
200		编码器 2 单位	0: pulse, 1: mm, 2: inch, 3:degree
201		编码器 2 每旋转脉冲	1 ~ 4294967295
202		编码器 2 每旋转距离	0.000000001 ~ 4294967295
203		编码器 2 脉冲输入	0:CW/CCW 1 倍数, 1:PULSE/DIR 1 倍数 2:PULSE/DIR 2 倍数, 3:PHASE A/B 1 倍数 4:PHASE A/B 2 倍数, 5: PHASE A/B 4 倍数
204		编码器 2 最大值	(编码器 2 最小值+ 1) ~ 2147483647
205		编码器 2 最小值	-2147483648 ~ (编码器 2 最大值- 1)
206		编码器 2 输入滤波值	0: 未使用, 1: 500kPPS 2: 200kPPS, 3: 100kPPS 4: 10kPPS, 5: 1kPPS 6: 0.1kPPS

*备注 1) LREAL 范围: 2.2250738585072e-308 ~ 1.79769313486232e+308

LREAL 正向范围: 0 ~ 1.79769313486232e+308 (不包含 0)

6.3.15 写参数 (MC_WriteParameter)



- (1) 该运动功能块在相关轴的指定参数中写入值。
- (2) 执行输入上升沿参数写入。
- (3) 在 ParameterNumber 输入中写入指定的参数编号。当"error 0x10F0"不能设置值。
- (4) 在参数 Value 输入中写入指定的值。
- (5) ExecutionMode 输入指定设置点。0 表示运动功能块执行后立即设置，1 表示'Buffered'在顺序运行设置的相同点设置。"error0x101B"时不能设置值。

0 (mcImmediatly): 当执行功能块后立即改变参数值 (在执行输入上升沿)。如果相关轴在运行中，运行受影响。

1 (mcQueued):在缓存模式'Buffered'相同点改变。

(6) 参数号如下所示.

No	参数	项目	描述
0	基本参数	单位	0:pulse,1:mm,2:inch,3:degree
1		每转脉冲	1 ~ 4,294,967,295 [pulse]
2		每转距离	0.000000001 ~ 4,294,967,295 [Unit]
3		速度命令单位	0:Unit/Time, 1:rpm
4		速度限制	长实数型正数[Unit/s, rpm] (改变单位, 脉冲, 每转脉冲, 每转距离速度命令单位)
5		紧急停止	0 或长实数型正数[Unit/s ²]
6		编码器选择	0:增量编码器,1:绝对编码器
7		齿轮齿数比(电机)	1 ~ 65,535
8		齿轮齿数比(机器)	1 ~ 65,535
9		反转运行模式	0:E.Stop, 1:Stop
10	扩展参数	SAW 上限	长实数型 [Unit]
11		SAW 下限	长实数型[Unit]
12		无限重复运行位置	长实数型正数[Unit]
13		无限重复运行	0:Disable, 1:Enable
14		Inposition 范围命令	0 或长实数型正数[Unit]
15		跟踪误差超出范围值	0 或长实数型正数[Unit]
16		当前位置补偿量	0 或长实数型正数[Unit]
17		当前速度滤波时间常量	0 ~ 100
18		错误复位监控时间	1 ~ 1000 [ms]
19		速度控制中 SAW 限制	0:不检测, 1:检测
20		错误跟踪电平	0:警告, 1:报警
21		JOG 高速	长实数型正数[Unit] (Jog 低速~速度限制) [Unit/s]
22		JOG 低速	长实数型正数[Unit] (< Jog 高速) [Unit/s]
23		JOG 加速	0 或长实数型正数[Unit/ s ²]
24		JOG 减速	0 或长实数型正数[Unit/ s ²]
25		JOG 加速度	0 或长实数型正数[Unit/ s ²]
26		覆盖模式	0: 比率指定, 1: 单位指定

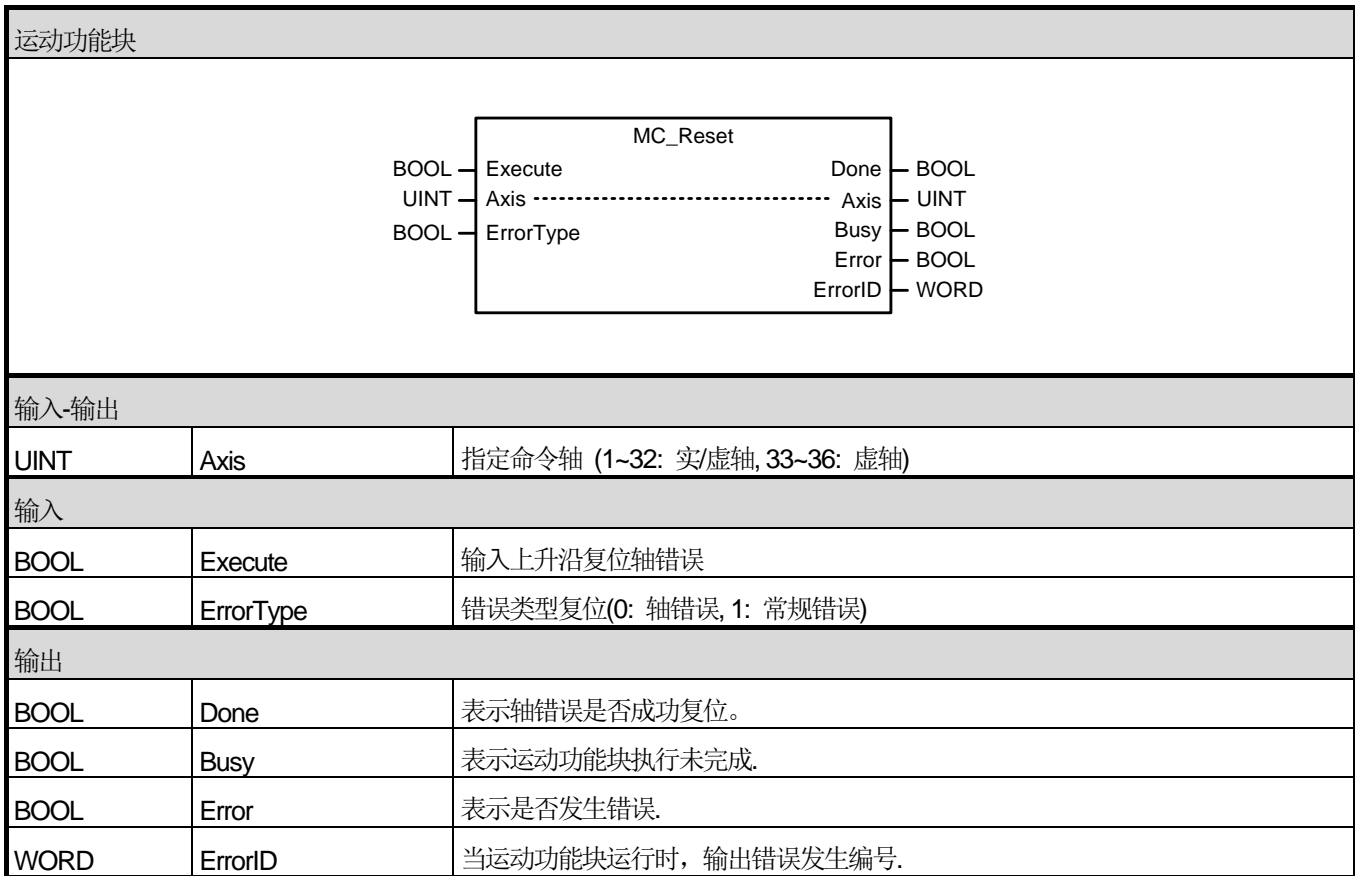
第 6 章运动功能块

No	参数	项目	描述
27	NC 参数	达到主轴旋转指令速度的识别范围	0~100%
28		达到主轴旋转 0 速度的识别 RPM	0~100rpm
100	编码器参数	编码器 1 单位	0: pulse, 1: mm, 2: inch, 3:degree
101		编码器 1 每旋转脉冲	1 ~ 4294967295
102		编码器 1 每旋转距离	0.000000001 ~ 4294967295
103		编码器 1 脉冲输入	0:CW/CCW 1 倍数, 1:PULSE/DIR 1 倍数 2:PULSE/DIR 2 倍数, 3:PHASE A/B 1 倍数 4:PHASE A/B 2 倍数, 5: PHASE A/B 4 倍数
104		编码器 1 最大值	(编码器 1 最小值+ 1) ~ 2147483647
105		编码器 1 最小值	-2147483648 ~ (编码器 1 最大值 - 1)
106		编码器 1 输入滤波值	0: 未使用, 1: 500kPPS 2: 200kPPS, 3: 100kPPS 4: 10kPPS, 5: 1kPPS 6: 0.1kPPS
200		编码器 2 单位	0: pulse, 1: mm, 2: inch, 3:degree
201		编码器 2 每旋转脉冲	1 ~ 4294967295
202		编码器 2 每旋转距离	0.000000001 ~ 4294967295
203		编码器 2 脉冲输入	0:CW/CCW 1 倍数, 1:PULSE/DIR 1 倍数 2:PULSE/DIR 2 倍数, 3:PHASE A/B 1 倍数 4:PHASE A/B 2 倍数, 5: PHASE A/B 4 倍数
204		编码器 2 最大值	(编码器 2 最小值+ 1) ~ 2147483647
205		编码器 2 最小值	-2147483648 ~ (编码器 2 最大值- 1)
206		编码器 2 输入滤波值	0: 未使用, 1: 500kPPS 2: 200kPPS, 3: 100kPPS 4: 10kPPS, 5: 1kPPS 6: 0.1kPPS

*备注 1) LREAL 范围: 2.2250738585072e-308 ~ 1.79769313486232e+308

LREAL 正向范围: 0 ~ 1.79769313486232e+308 (不包含 0)

6.3.16 轴错误复位(MC_Reset)



- (1) 该运动功能块用于复位相关轴错误。当 **ErrorType** 设置为 '0' 执行运动功能块, 如果相关轴为 'ErrorStop'状态, 轴错误复位, 轴状态转换为 'StandStill' 或'Disabled'状态。
- (2) 如果 **ErrorType** 被设置为 '1'运动功能块执行, 发生在相关模块的常规错误复位。
- (3) 运动功能块在 **Execute** 输入上升沿执行。

6.3.17 接触探头 (MC_TouchProbe)

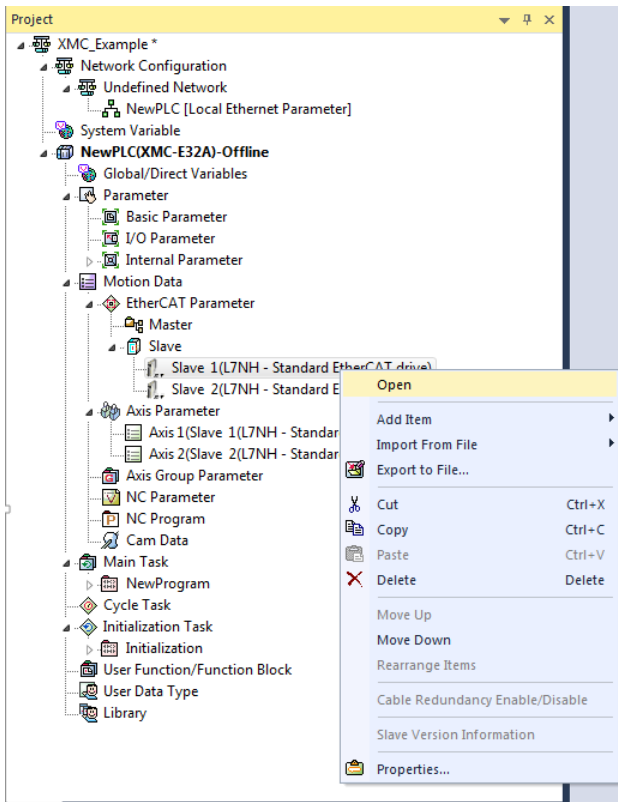


- (1) 该运动功能块当触发事件发生时，执行 '接触探头' 功能记录。
- (2) Execute 输入上升沿开始接触探头功能。
- (3) 在 TriggerInput 中指定信号作为触发器。发生 "error 0x10E1"时，不能设置值。
- (4) 当激活窗口模式时，在允许区域中可以设置接收到轴的触发信号。当激活窗口模式时，信号运行时序图如下所示。

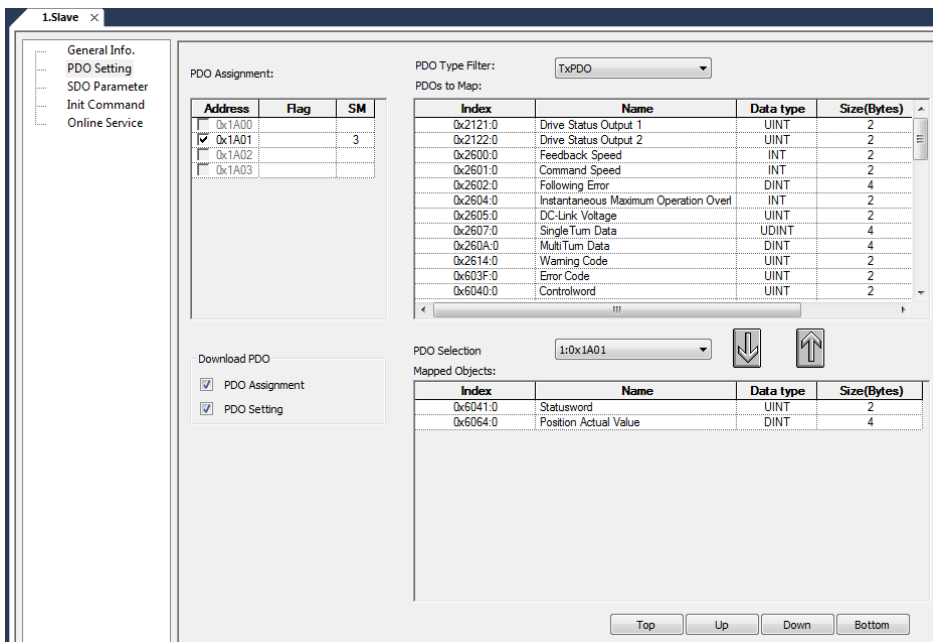
备注

如果使用接触探头，请在使用前设置从站参数。

1. MP500 中，点击伺服驱动注册信息。



2. 在从站信息窗口选择PDO设置



备注

3. 在编辑窗口选择接触探头项目，并点击箭头(向下)，并在PDO通讯数据中包含。接触探头相关PDO项目如下。

- 1) RxPDO
接触探头功能(0x60B8)
- 2) TxPDO
接触探头功能(0x60B8)
接触探头状态(0x60B9)
接触探头 1 正向位置值 (0x60BA)
接触探头 1 反向位置值 (0x60BB)
接触探头 2 正向位置值 (0x60BC)
接触探头 2 反向(0x60BD)

在 PDO 编辑窗口，选择接触探头 2 正向位置值，并选择向下箭头。对于某些伺服驱动，可能发生 PDO 设置错误 (0xF22)，防止连接到伺服驱动。在这种情况下，PDOs 选择数量需要如右侧显示进行调整(取消选择未使用 PDOs)。

The screenshot shows the PDO configuration interface with the following data:

PDO Assignment:

Address	Flag	SM
<input type="checkbox"/> 0x1A00		
<input checked="" type="checkbox"/> 0x1A01		3
<input type="checkbox"/> 0x1A02		
<input type="checkbox"/> 0x1A03		

PDO Type Filter: TxPDO

PDOs to Map:

Index	Name	Data type	Size(Bytes)
0x60B0:0	Position Offset	DINT	4
0x60B1:0	Velocity Offset	DINT	4
0x60B2:0	Torque Offset	INT	2
0x60B8:0	Touch Probe Function	UINT	2
0x60B9:0	Touch Probe Status	UINT	2
0x60BA:0	Touch Probe 1 Positive Edge Position V	DINT	4
0x60BB:0	Touch Probe 1 Negative Edge Position	DINT	4
0x60BC:0	Touch Probe 2 Positive Edge Position V	DINT	4
0x60BD:0	Touch Probe 2 Negative Edge Position	DINT	4
0x60E0:0	Positive Torque Limit Value	UINT	2
0x60E1:0	Negative Torque Limit Value	UINT	2
0x60F4:0	Following Error Actual Value	DINT	4

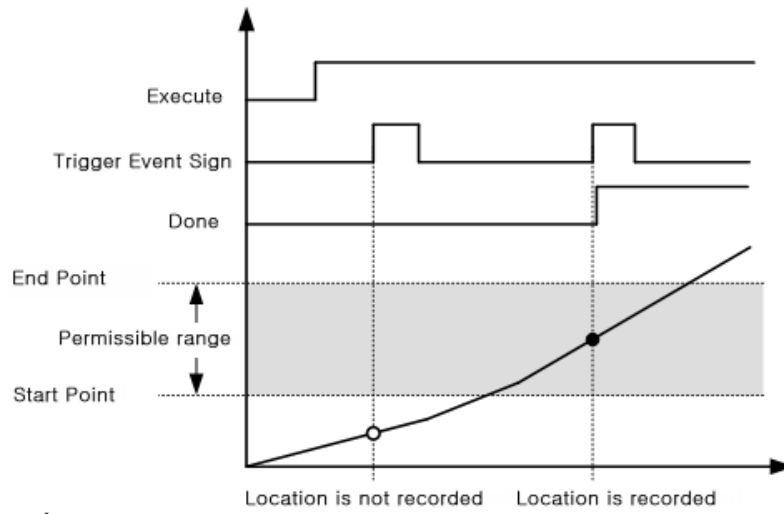
PDO Selection: 1:0x1A01

Mapped Objects:

Index	Name	Data type	Size(Bytes)
0x6041:0	Statusword	UINT	2
0x6064:0	Position Actual Value	DINT	4

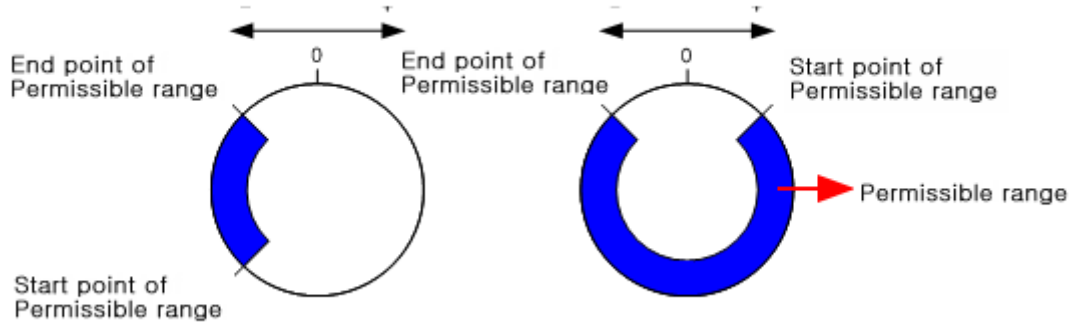
Buttons: Top, Up, Down, Bottom

4. 编辑 PDO 项目完成后,必须在运动控制器中写入'EtherCAT 参数'.

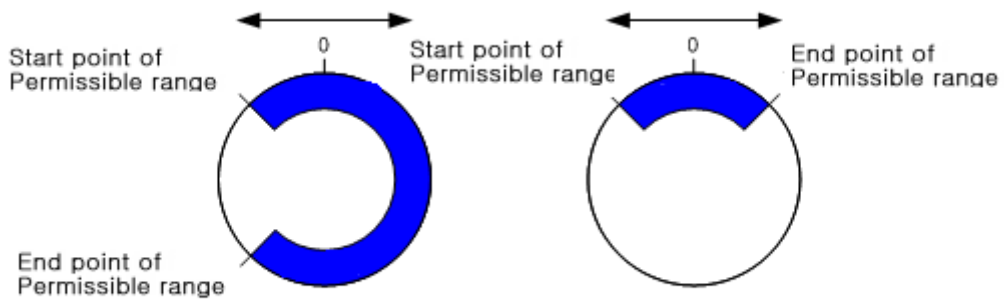


<接触探头功能为窗口模式, 运行时序>

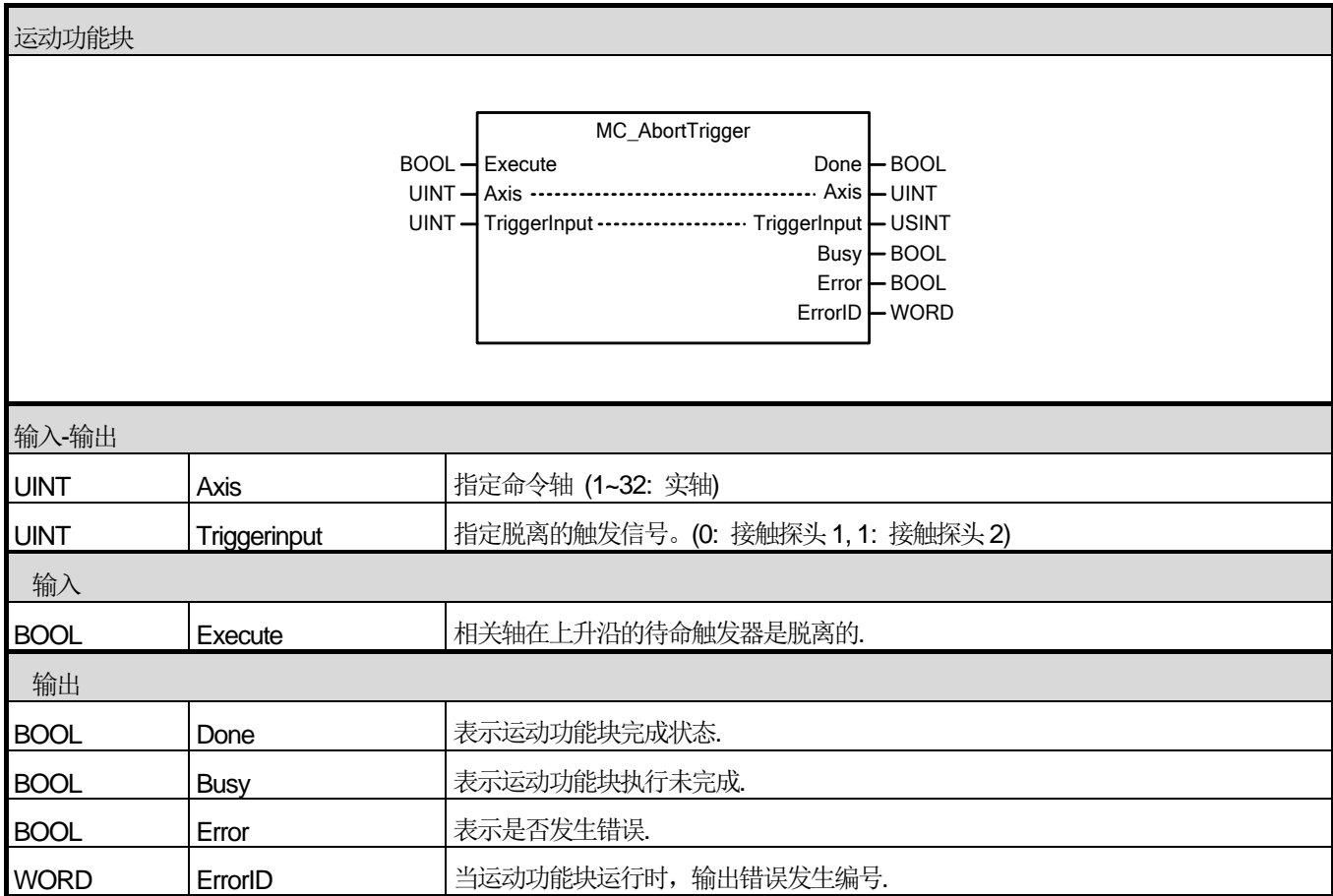
● In case of Permissible range start point < Permissible range end point



● In case of Permissible range start point > Permissible range end point

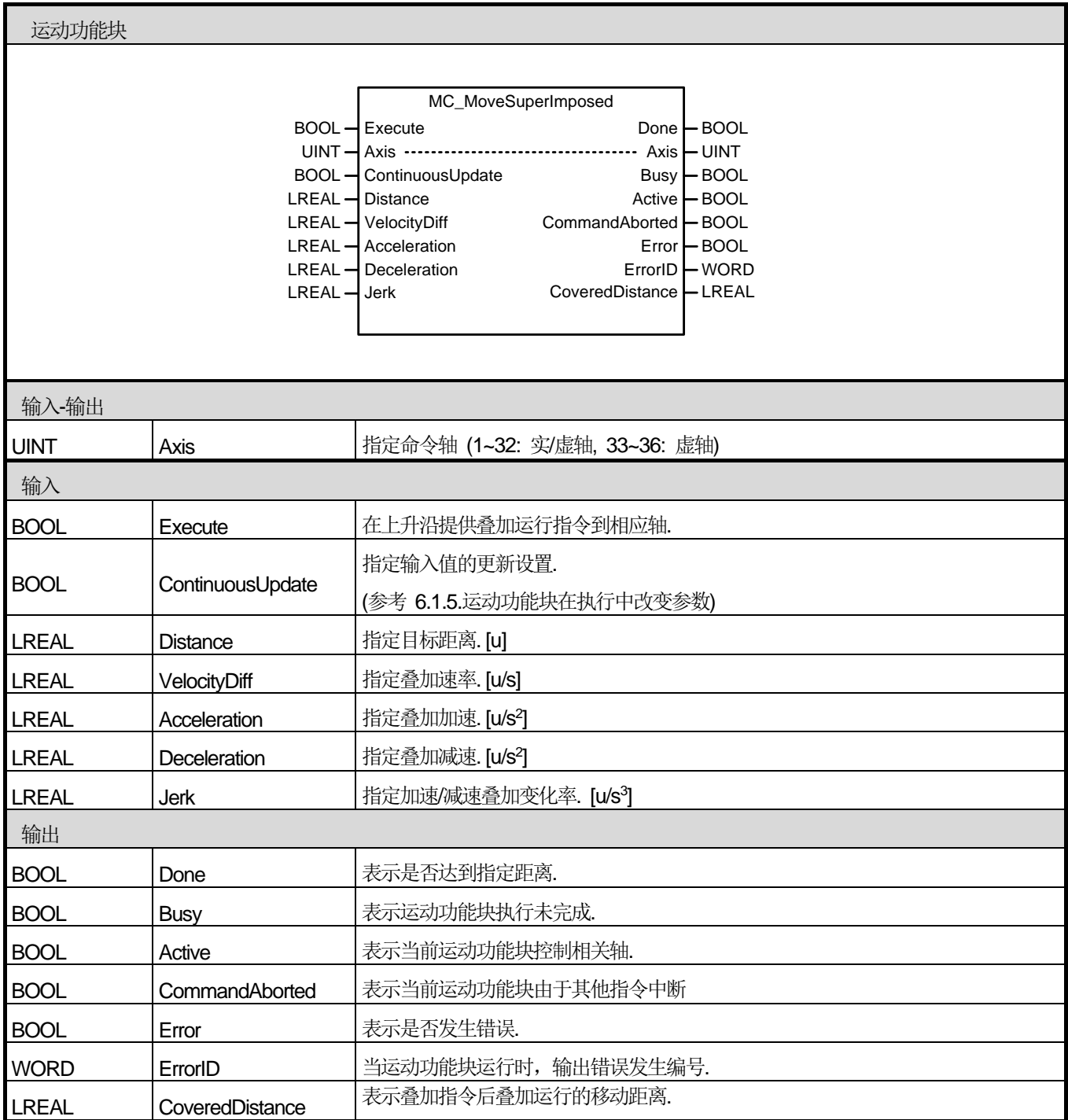


6.3.18 中止触发事件 (MC_AbortTrigger)



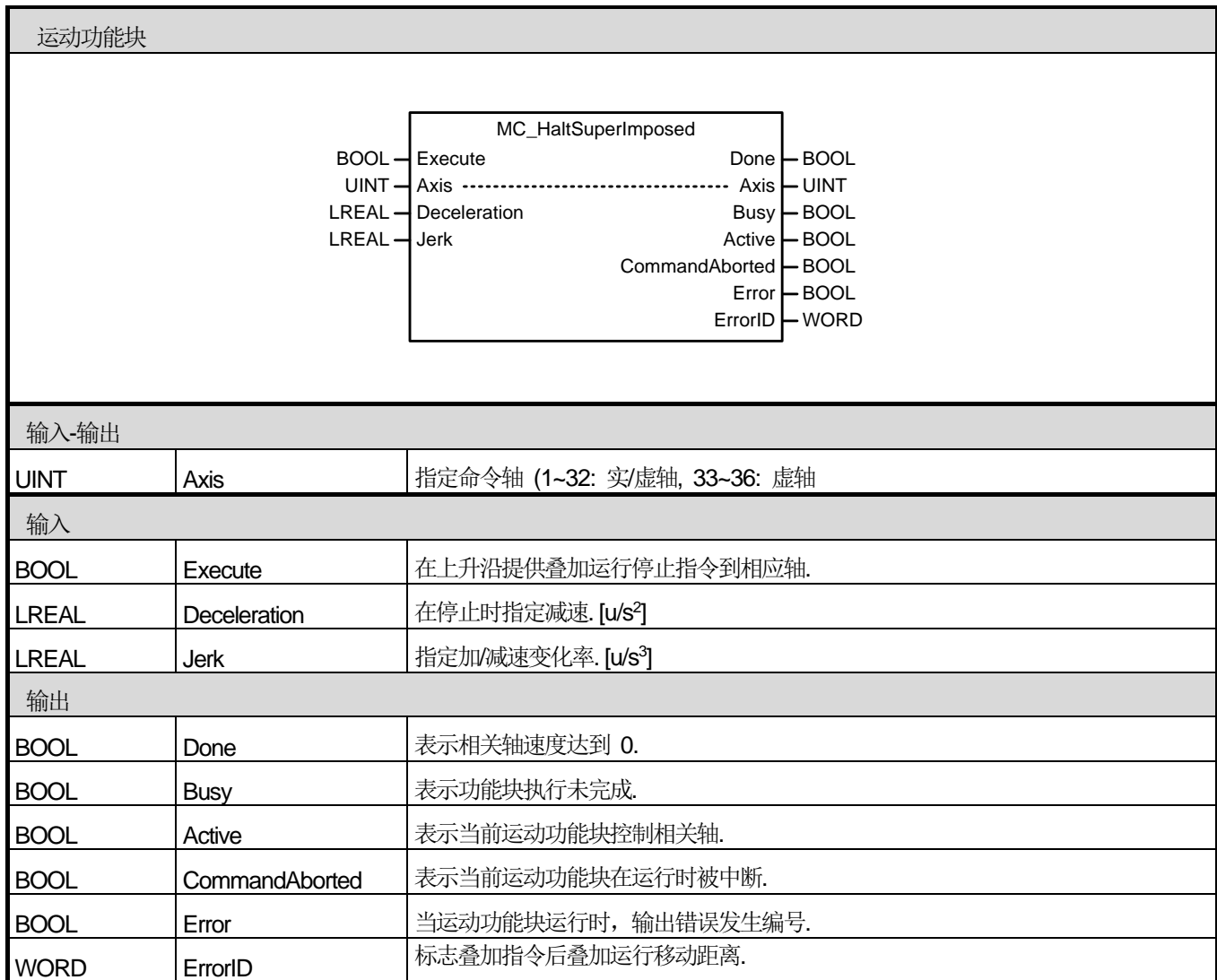
- (1) 该运动功能块用户脱离相关待命轴的触发器。
- (2) 在 TriggerInput 中指定脱离的触发信号。发生 “error 0x10E1”不能设置值。

6.3.19 叠加运行 (MC_MoveSuperImposed)



- (1) 该运动功能块是发布叠加运行到相关轴的指令.
- (2) 叠加指令是为了在指令运行时, 从当前位置移动到通过距离输入中设置的目标距离.
- (3) 运行方向由设置距离的正/负决定. 正数距离(+或者无符号)表示正向移动, 负数距离(-)表示反向移动.
- (4) 到达目标距离之后, 当速率为 0, 指令完成, Done 输出为 on.

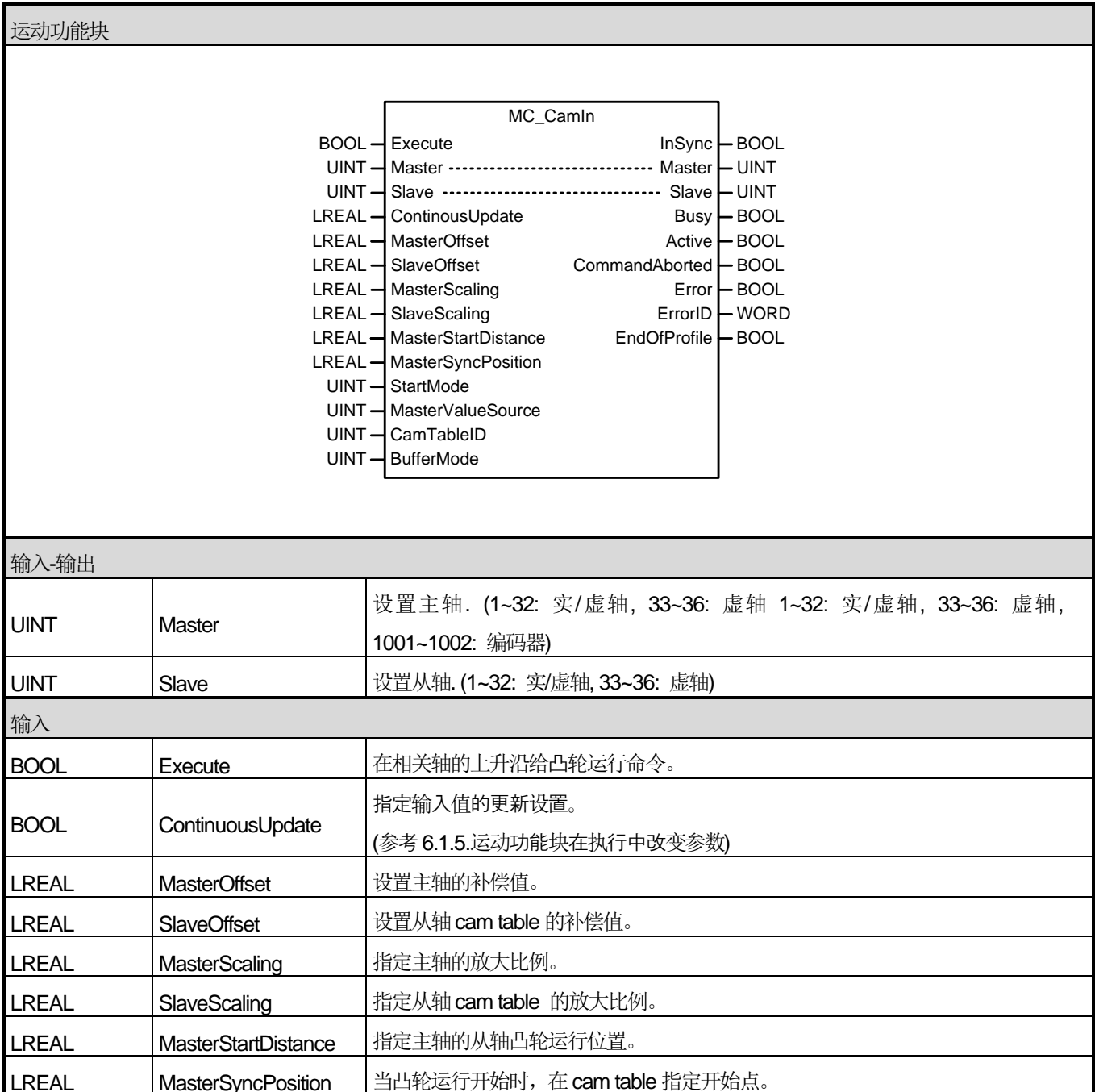
6.3.20 叠加运行停止 (MC_HaltSuperImposed)



- (1) 该运动功能块是发布停止叠加运行到相关轴的指令.
- (2) 叠加运行停止指令是用于减速, 并在加速和加速度时执行该指令停止.
- (3) 到达目标距离之后, 当速率为 0, 指令完成, Done 输出为 on.

6.4 多轴运动功能块

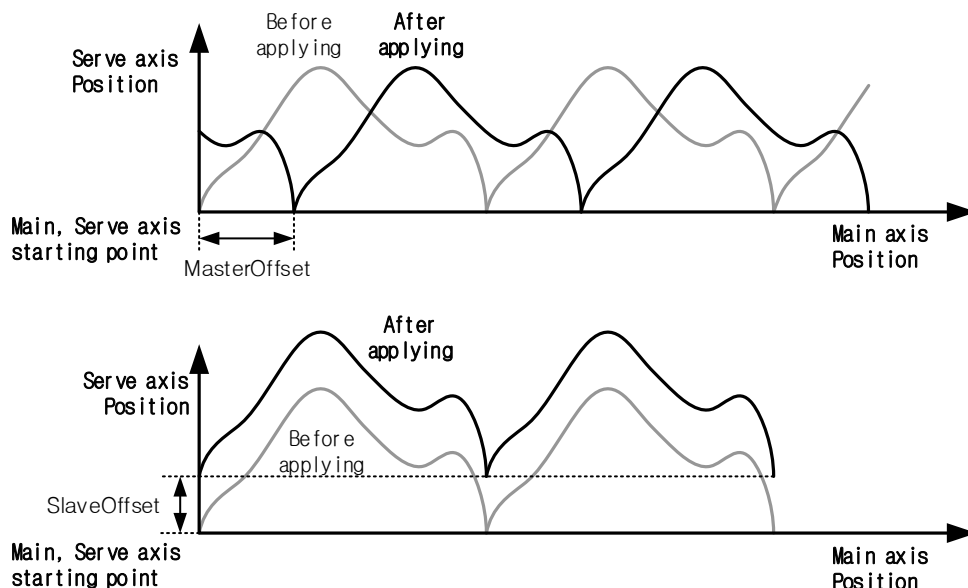
6.4.1 凸轮运行(MC_CamIn)



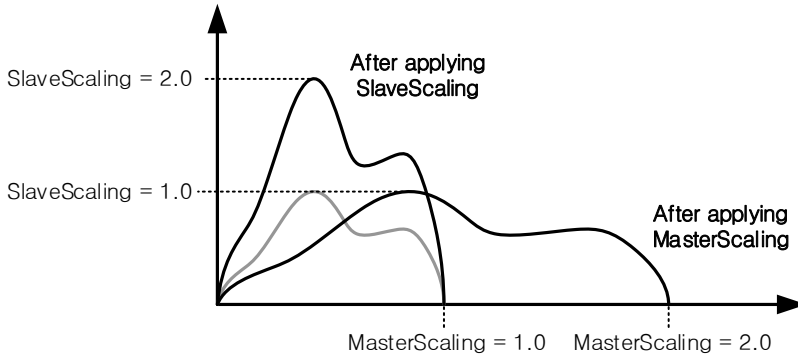
第 6 章运动功能块

UINT	StartMode	设置凸轮运行模式。 0 : cam table 作为绝对值应用。(mcAbsolute) 1 : cam table 作为基于命令开始点相对值应用。(mcRelative)
UINT	MasterValueSource	选择主轴凸轮运行源。 0 : 在主轴目标值同步 1 : 在从轴当前值同步
UINT	CamTableID	指定运行的 cam table
UINT	BufferMode	指定运动功能块的顺序运行设置。 (参考 6.1.4.缓存模式)
输出		
BOOL	InSync	表示凸轮运行正常完成。 (表示从轴遵循 cam table)
BOOL	Busy	表示运动功能块执行未完成。
BOOL	Active	表示当前运动功能块控制相关轴。
BOOL	CommandAborted	表示当前运动功能块在运行时被中断。
BOOL	Error	表示是否发生错误。
WORD	ErrorID	当运动功能块运行时, 输出错误发生编号。

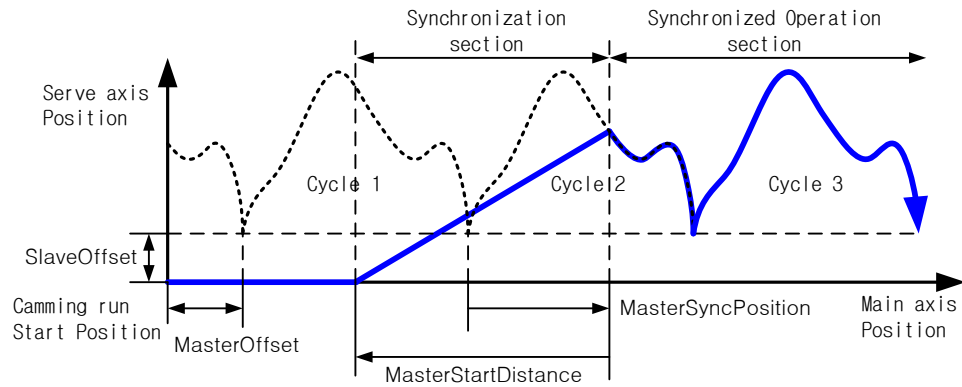
- (1) 该运动功能块运行从轴凸轮取决与主轴。
- (2) 即使主轴为停止状态, 凸轮可给从轴运行命令。
- (3) 必须给从轴或其它停止凸轮运行运动功能块中断命令。(MC_CamOut)
- (4) 如果该运动功能块由于其他指令而停止(最近执行指令缓存模式=0), 凸轮运行停止, 指令终止输出为 on.
- (5) 当该运动功能块执行时, 如果缓存执行其他指令(最近执行指令缓存模式=1~5), 凸轮轮廓循环运行停止, 然后运行最新执行指令. MC_CamIn 功能块的 InSync / Busy / Active / CommandAborted / Error 输出为 Off.
- (6) 当运动功能块运行时, 轴为 'Synchronized Motion' 。
- (7) 根据主轴的开始点设置应用于 MasterOffset 和 SlaveOffsecam table 的补偿量, 根据从轴的开始点设置 SlaveOffset 的补偿量。参考下图。



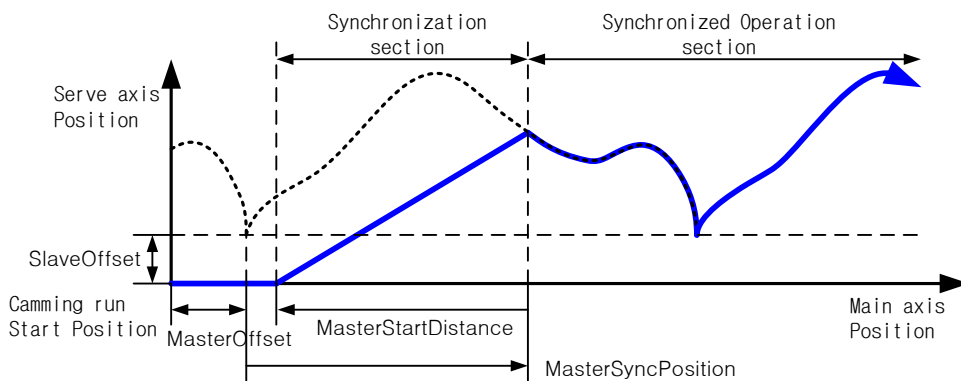
- (8) 设置适用于 MasterScaling 和 SlaveScaling 凸轮的放大率。在 MasterScaling 中设置主轴数据的放大率和从轴数据的放大率。参考下图。



- (9) MasterSyncPosition 输入指定主轴在表中实际凸轮运行同步位置完成, MasterStartDistance 输入指定主轴的相对位置同步启动。如果如下循环 1 无法同步启动运行(如果从起始位置到同步运行启动位置的距离小于 MasterStartDistance), 则以循环 2 同步启动运行。



<MasterScaling 为 1.0>



<MasterScaling 为 2.0>

MasterSyncPosition 位置基于 cam table, 实际同步位置由 MasterOffset 和 MasterScale 参数确定。

从轴移动至同步位置从基于 MasterSyncPosition 实际应用输入距离值。在启动移动前, 从轴在停止状态相关位置等待, 如果从轴在命令开始前已经移动至同步位置部分, 撤销同步开始启动点位置, 直到脱离 MasterStartDistance 范围。

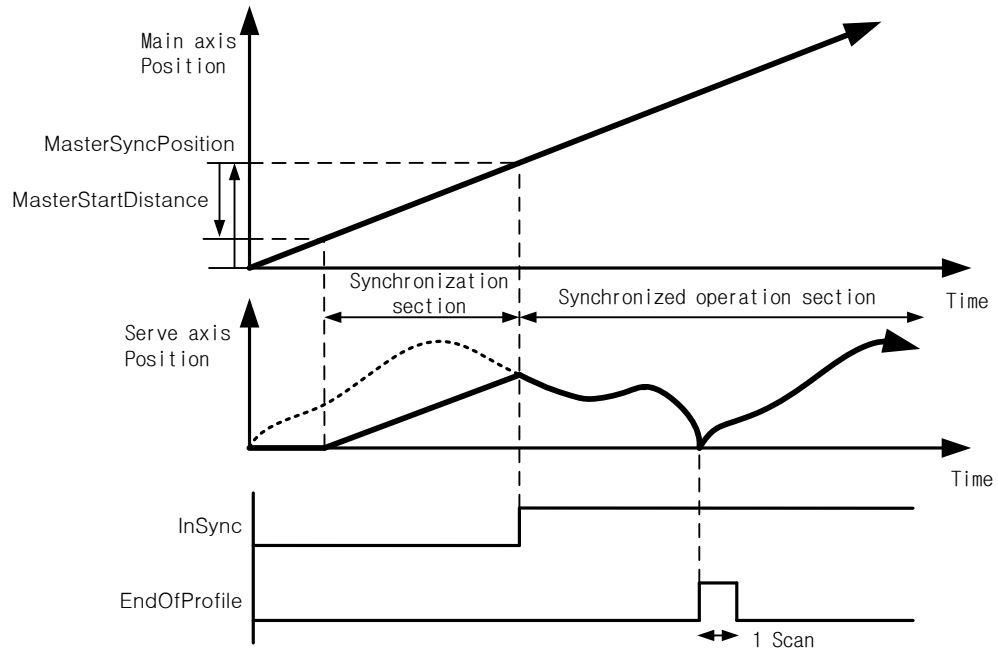
实际同步位置分类取决于 MasterScaling 和 SlaveScaling, 因为 MasterSyncPosition 是基于 cam table 内部的值, 但是 MasterOffset 和 MasterStartDistance 值保持不受影响。

第 6 章运动功能块

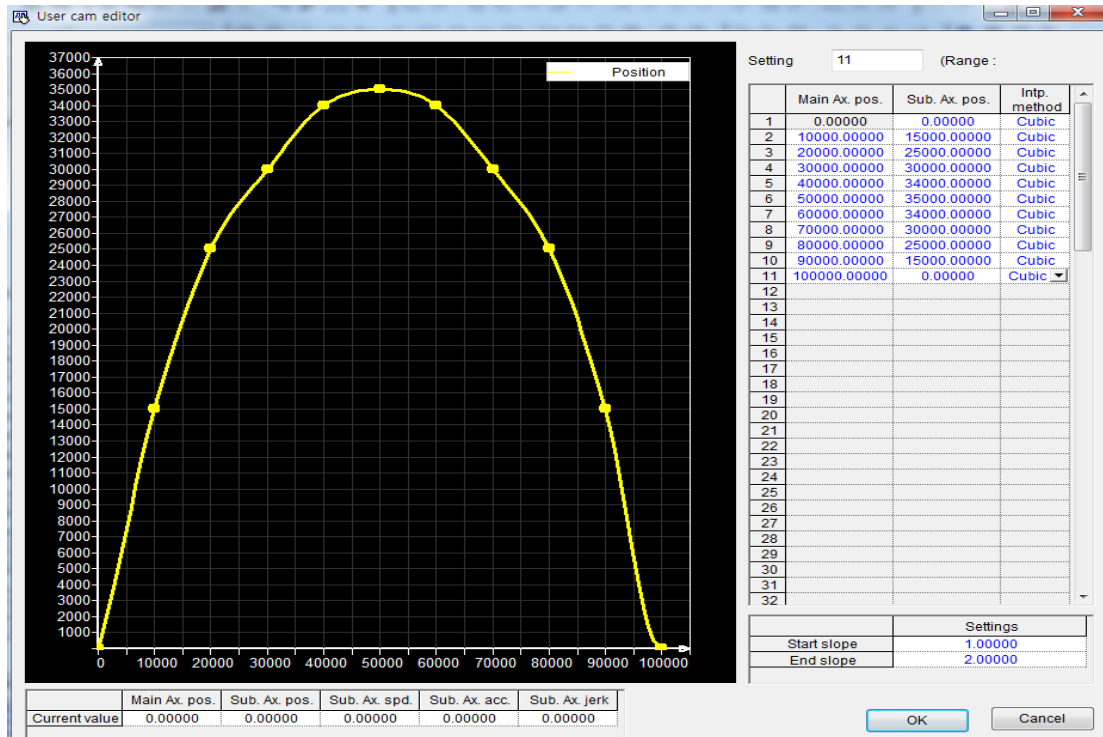
(10) 如果 ContinuousUpdate 输入为 On, 应用变更的参数.

仅 MasterOffset, SlaveOffset, MasterScaling, SlaveScaling, MasterStartDistance, MasterSyncPosition 可以更新(然而, InSync=On, 则 MasterOffset, SlaveOffset, MasterScaling, SlaveScaling 可以更新).

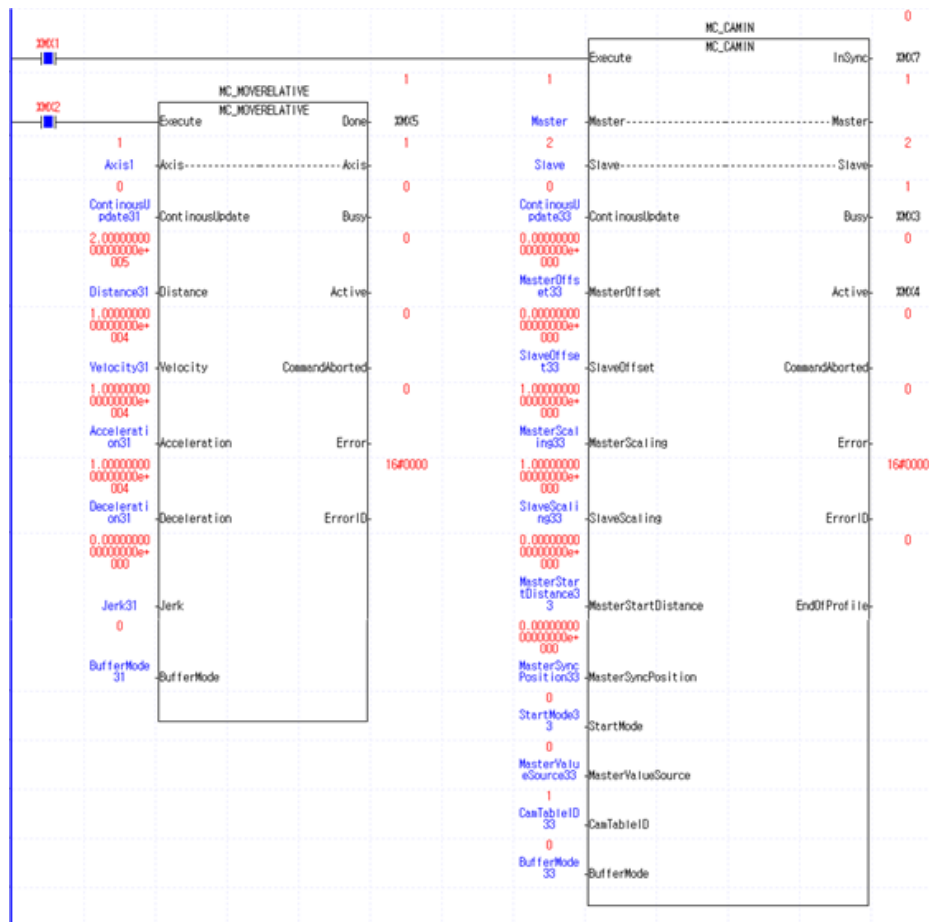
(11) 一旦凸轮运行正常启动, InSync 输出为 On, EndOfProfile 输出为 1, 扫描 On 每次完成一个 cam table 运行.



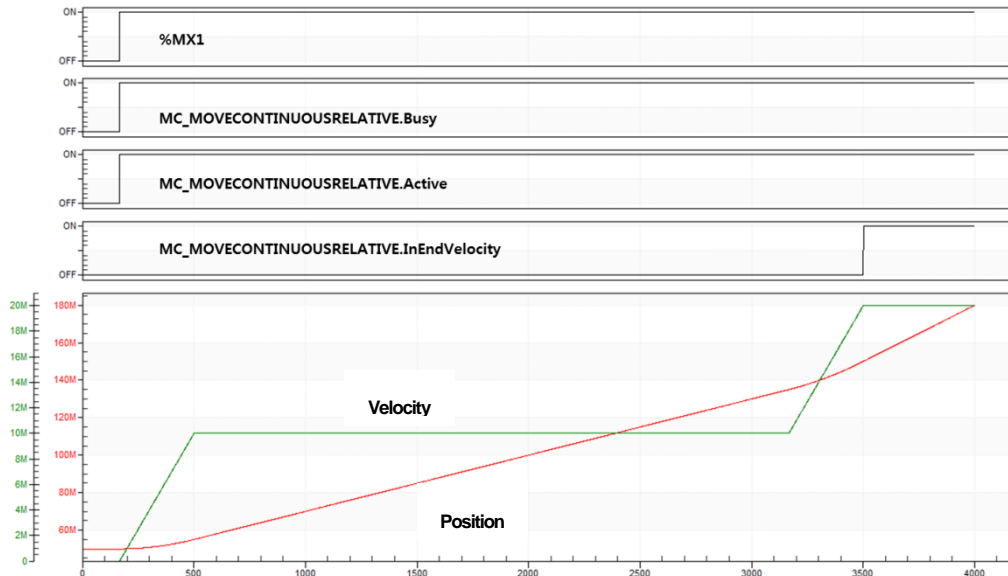
- (12) 凸轮运行模式在 StartMode 中设置，设置范围 0 或 1，输入值超出设置范围发生错误。
- (13) MasterValueSource 选择主轴同步的来源。如果设置为 0，从轴基于主轴在运动控制模块中计算命令位置执行凸轮运行，如果设置为 1，从轴基于主轴的伺服驱动器通信接收，执行凸轮运行。
- (14) CAM TableID 设置 cam table 应用于凸轮运行的数目。设置范围 1~32，输入值超出设置范围，在运动功能块中发生错误 "0x1115"。
- (15) 当该运动功能块运行时，相关轴在 "SynchronizedMotion"状态。
- (16) 实例程序
该实例为产生凸轮轮廓后显示从 0 到 200,000 位置的主轴移动，然后在从轴执行 MC_CAMIN 指令。
- (a) 功能块设置



第 6 章运动功能块



(b) 时序图

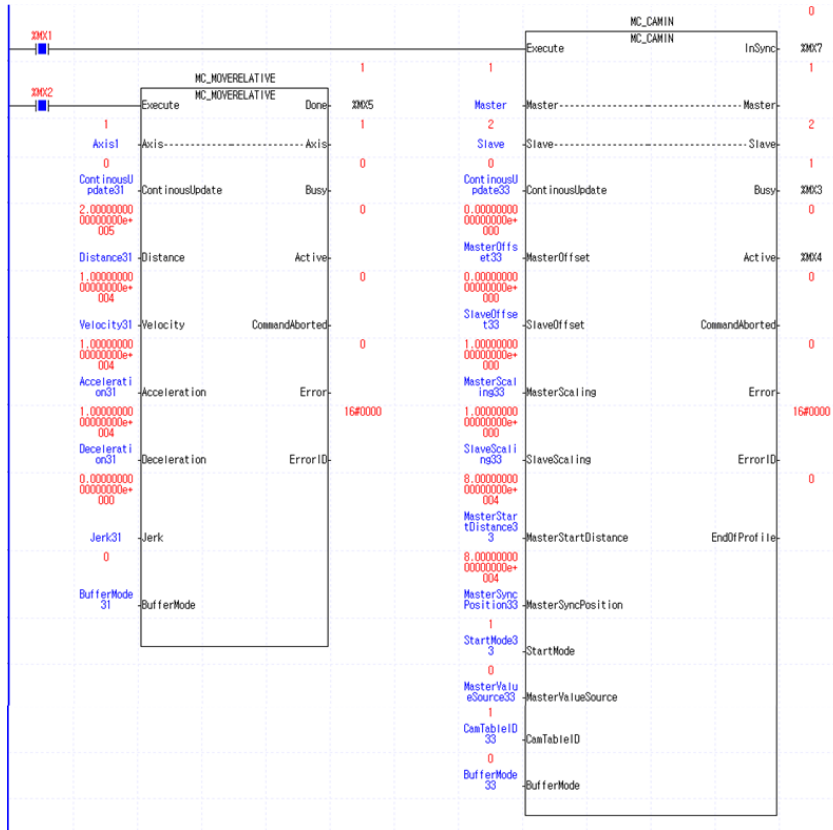


(17) 应用实例程序

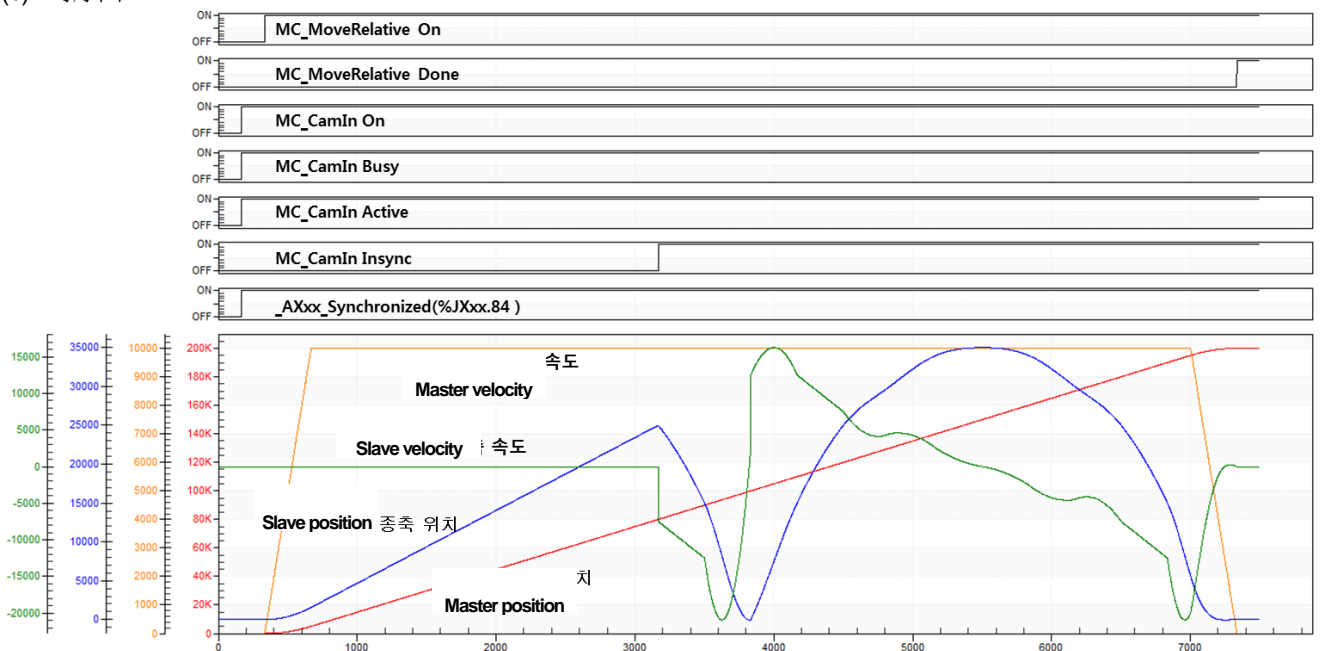
该实例为产生凸轮轮廓后显示从 0 到 200,000 位置的主轴移动，然后在从轴执行 MC_CAMIN 指令

该实例为产生凸轮轮廓后显示从 0 到 200,000 位置的主轴移动，然后在从轴执行 MC_CAMIN 指令，从轴中 MasterSyncPosition 和 MasterSyncDistance 设置为 80,000.

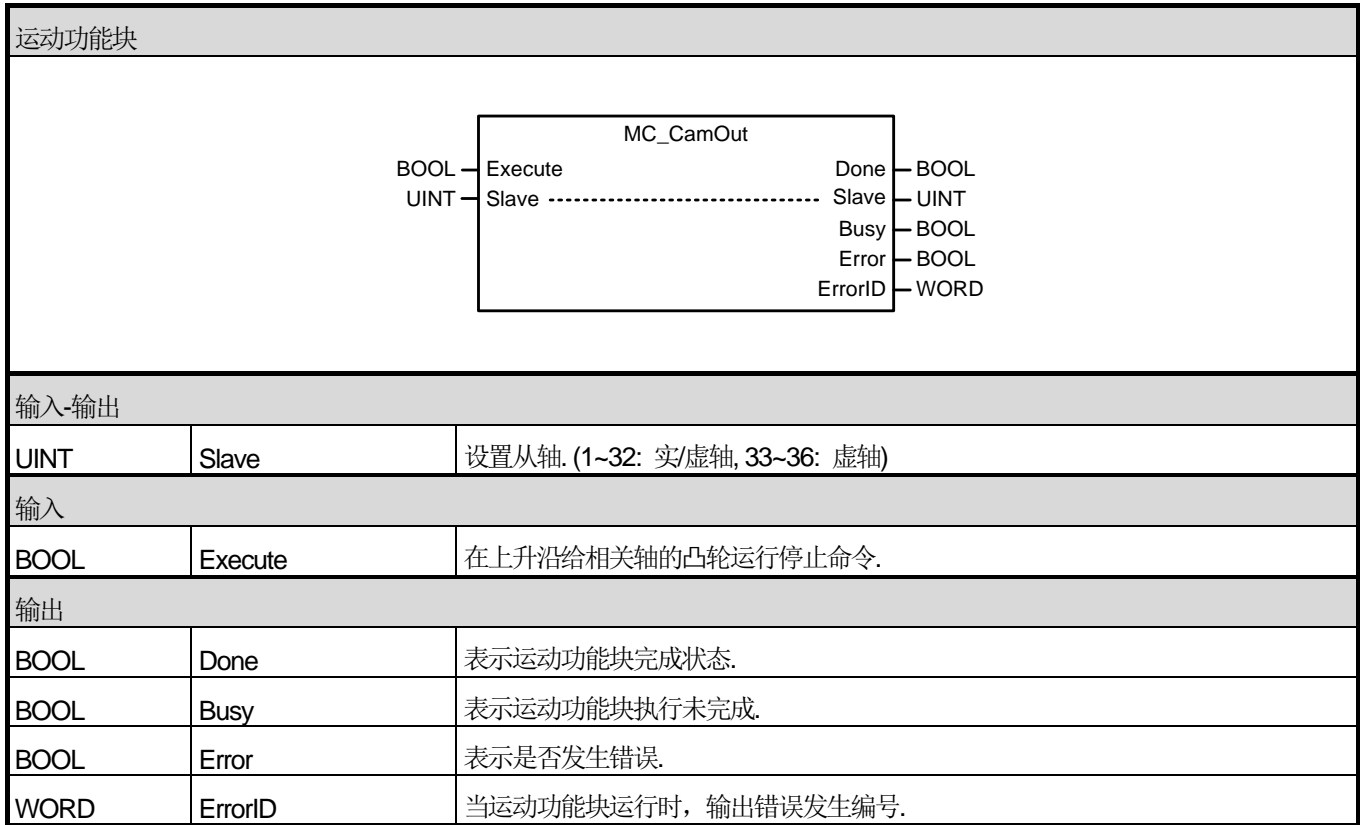
(a) 功能块设置



(b) 时序图

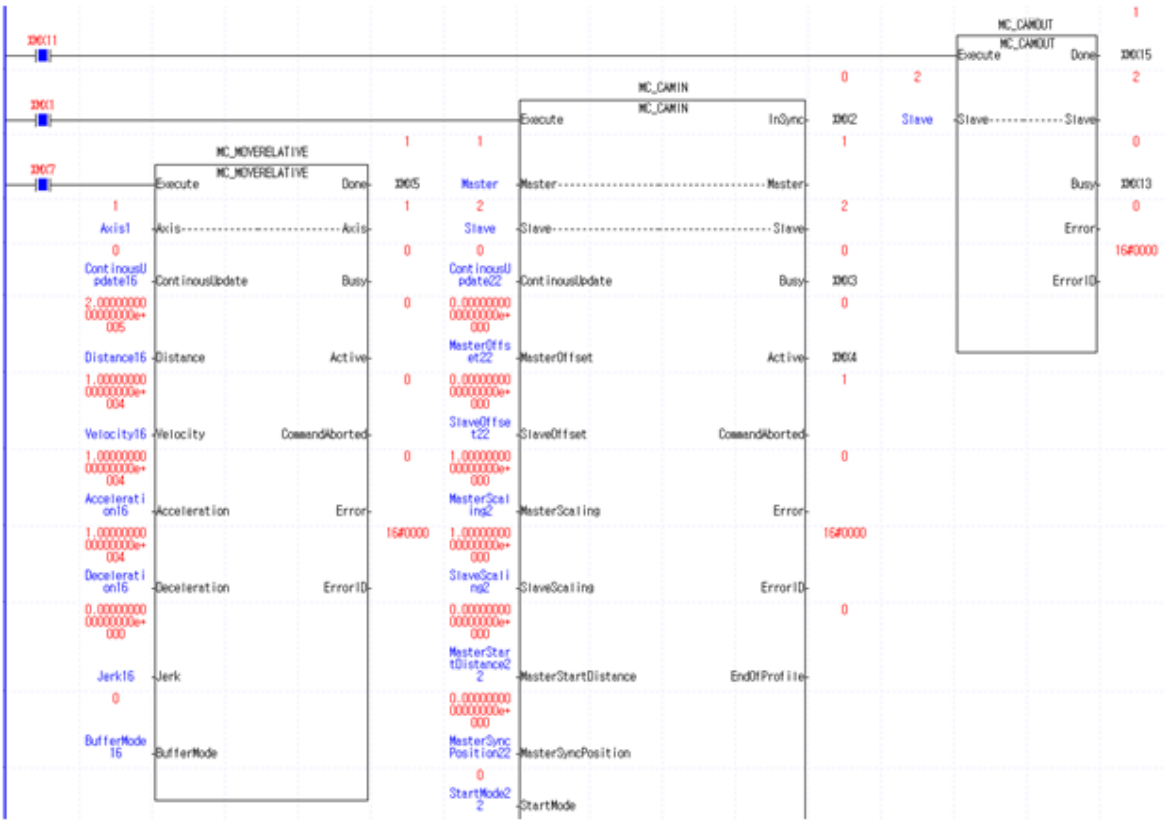
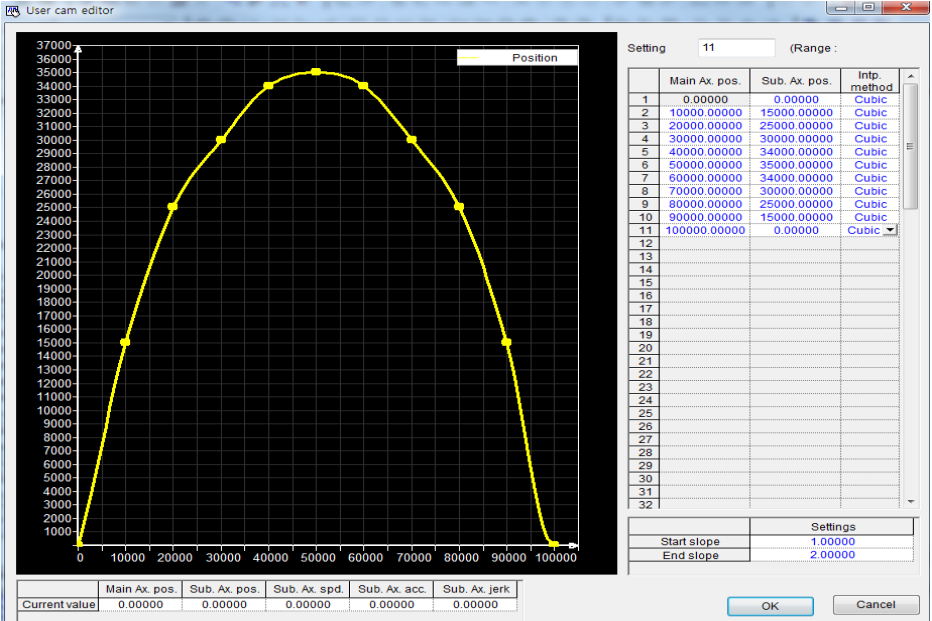


6.4.2 凸轮停止(MC_CamOut)

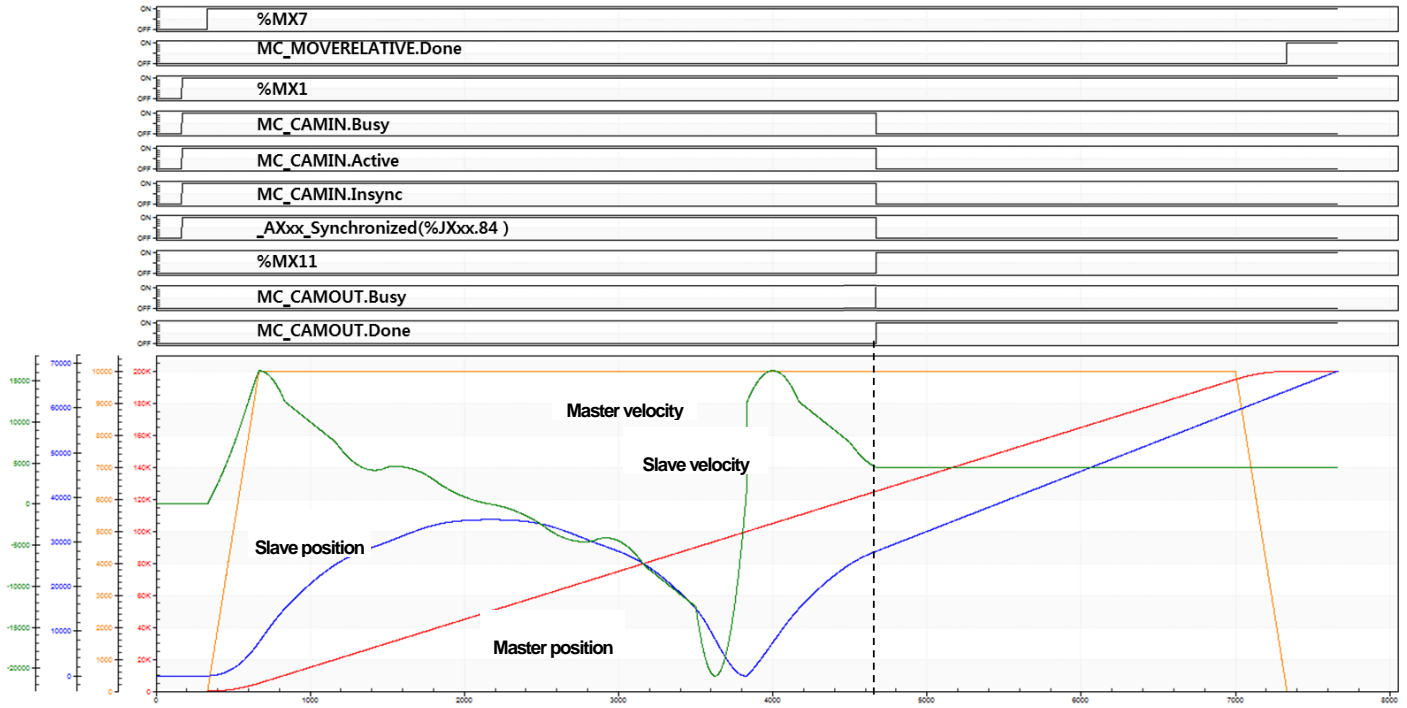


- (1) 该运动功能块在从轴中立即脱离凸轮运行。
- (2) 如果运动功能块的缓存模式在从轴凸轮运行时为 **Aborting**, 凸轮运行自动脱离, 相关运动功能块执行。执行凸轮运行 **abort (MC_CamOut)** 运动功能块, 相关轴在凸轮运行脱离时保持速度, 如果想要完成停止从轴, 使用停止 (**MC_Halt**) 或立即停止 (**MC_Stop**) 运动功能块。
- (3) 当执行 **MC_CamOut** 运动功能块时, **MC_CamIn** 功能块 **InSync** 输出和同步状态标志位(**_AXxx_Synchronized**)为 off.
- (4) 实例程序
 该实例显示发生凸轮轮廓, 在从轴执行 **MC_CAMIN** 指令, 移动主轴到 200,000 位置,然后执行 **MC_CAMOUT**. 当凸轮运行终止时从轴维持速率.

(a) 功能块设置



(b) 时序图



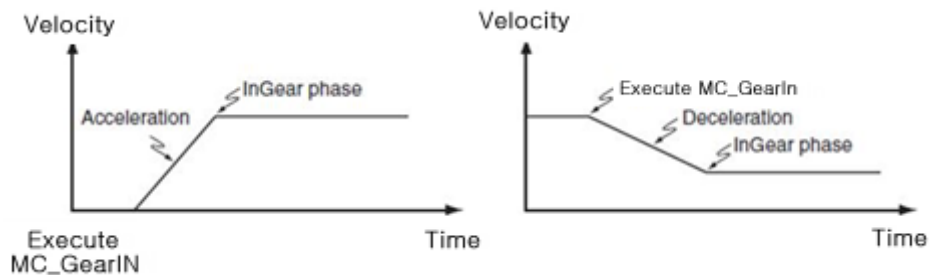
6.4.3 电子传动运行 (MC_GearIn)

运动功能块		
输入-输出		
UINT	Master	设置主轴。(1~32: 实/虚轴, 33~36: 虚轴 1~32: 实/虚轴, 33~36: 虚轴, 1001~1002: 编码器)
UINT	Slave	设置从轴。(1~32: 实/虚轴, 33~36: 虚轴)
输入		
BOOL	Execute	在上升沿给相关轴齿轮运行命令.
BOOL	ContinuousUpdate	指定输入值的更新设置. (参考 6.1.5.运动功能块在执行中改变参数)
INT	RatioNumerator	指定齿轮比数目。(-32768 ~ 32767)
UINT	RatioDenominator	指定齿轮比的分母。(0 ~ 65535)
UINT	MasterValueSource	选择主轴同步的数据。 0: 主轴命令位置同步 1: 主轴当前位置同步
LREAL	Acceleration	指定齿轮运行同步在加速开始处运行。[u/s ²]
LREAL	Deceleration	指定齿轮运行同步在减速开始处运行。[u/s ²]
LREAL	Jerk	指定加/减速变化率。[u/s ³]
UINT	BufferMode	指定运动功能块的顺序运行设置。 (参考 6.1.4.缓存模式)

第 6 章运动功能块

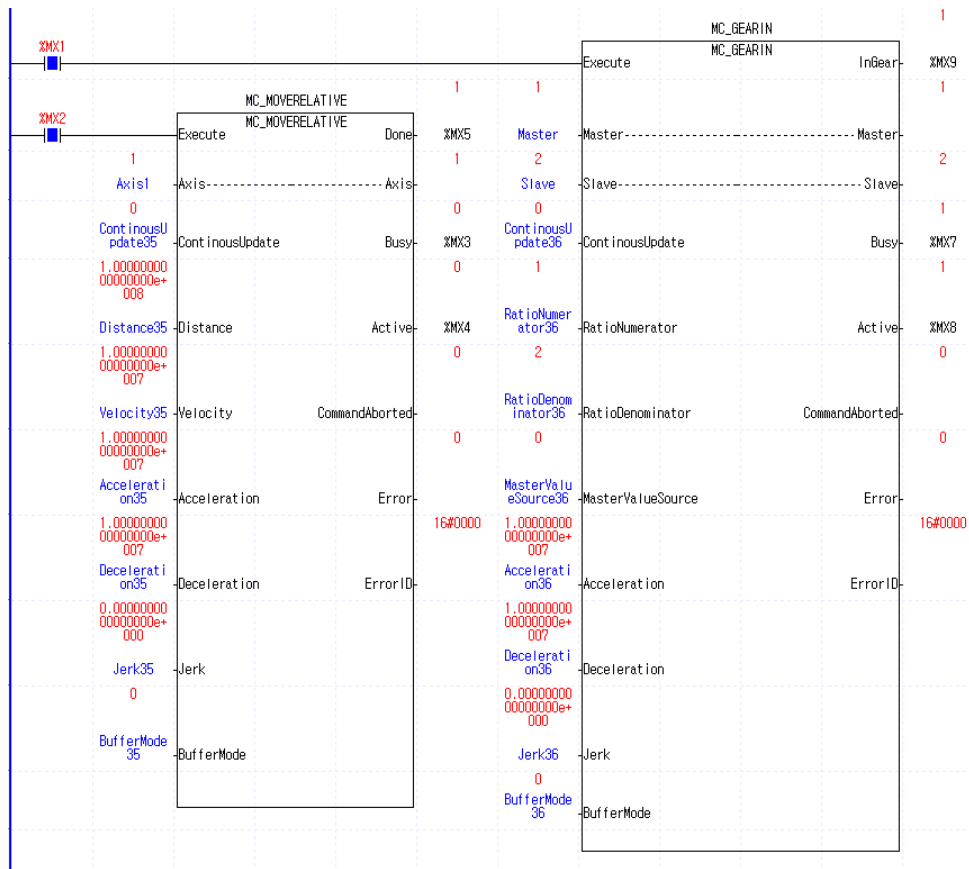
输出		
BOOL	InGear	表示应用齿轮比率执行齿轮运行.
BOOL	Busy	表示运动功能块执行未完成.
BOOL	Active	表示当前运动功能块控制相关轴.
BOOL	CommandAborted	表示当前运动功能块在运行时被中断.
BOOL	Error	表示是否发生错误.
WORD	ErrorID	当运动功能块运行时, 输出错误发生编号.

- (1) 该运动功能块是主轴同步速度运行, 从轴通过设置齿轮齿数比决定。
- (2) 给相关轴 gear 运行中止(MC_GearOut) 命令或执行其他允许脱离 gear 运行的运动功能块。
- (3) RatioNumerator 和 RatioDenominator 设置分子和分母各自应用于从轴。如果分子设置为负数, 从轴的旋转方向和主轴相反。
- (4) MasterValueSource 选择数据主轴同步的标准。如果设置为 0, 同步运行基于主轴运动控制模块命令位置, 如果设置为 1,同步运行基于当前位置。其他值设置运动功能块 On 时错误, 在 ErrorID 发生“0x1114”。
- (5) 如果该运动功能块由其他指令停止(最近执行指令缓存模式=0), 凸轮运行停止, CommandAborted 输出为 on.
- (6) 当运动功能块执行时如果通过缓存执行其他指令(最近执行指令缓存模式=1-5), 齿轮运行状态(InGear 相)停止, 随后运行最近执行指令。
功能块的 InGear / Busy / Active / CommandAborted / Error 输出都为 Off.
- (7) 当执行该运动功能块时, 从轴通过相关轴齿轮率加减速和主轴同步。
- (8) 当该运动功能块运行时, 从轴为'SynchronizedMotion'。

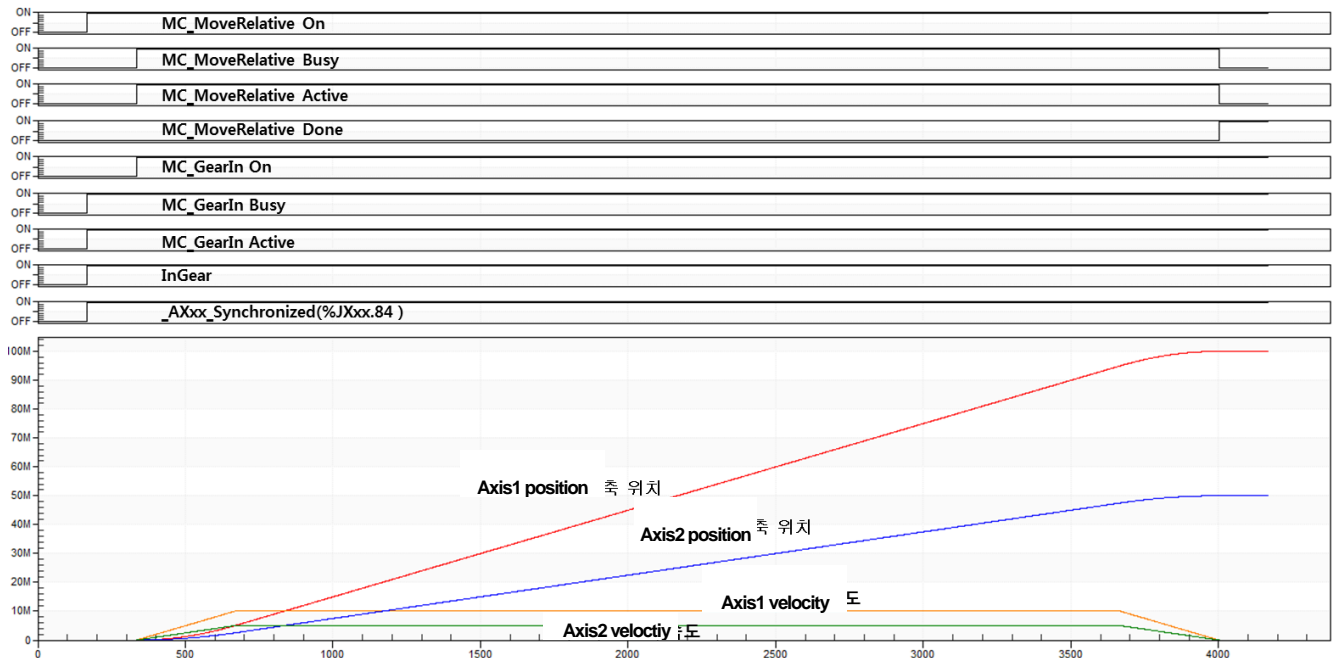


- (9) 当 ContinuousUpdate 输入为 On 时应用变更的参数. 仅 RatioNumerator, RatioDenominator, Acceleration, Deceleration 输入可以更新. (然而, InGear=On 时, RatioNumerator, RatioDenominator 输入可以更新)
- (10) 实例程序
实例显示当前位置为 0, 在轴 2 (从轴) 执行 MC_GearIn 指令后移动 1 轴 (主轴) 到 100,000,000 时, 2 轴到达 50,000,000 运行

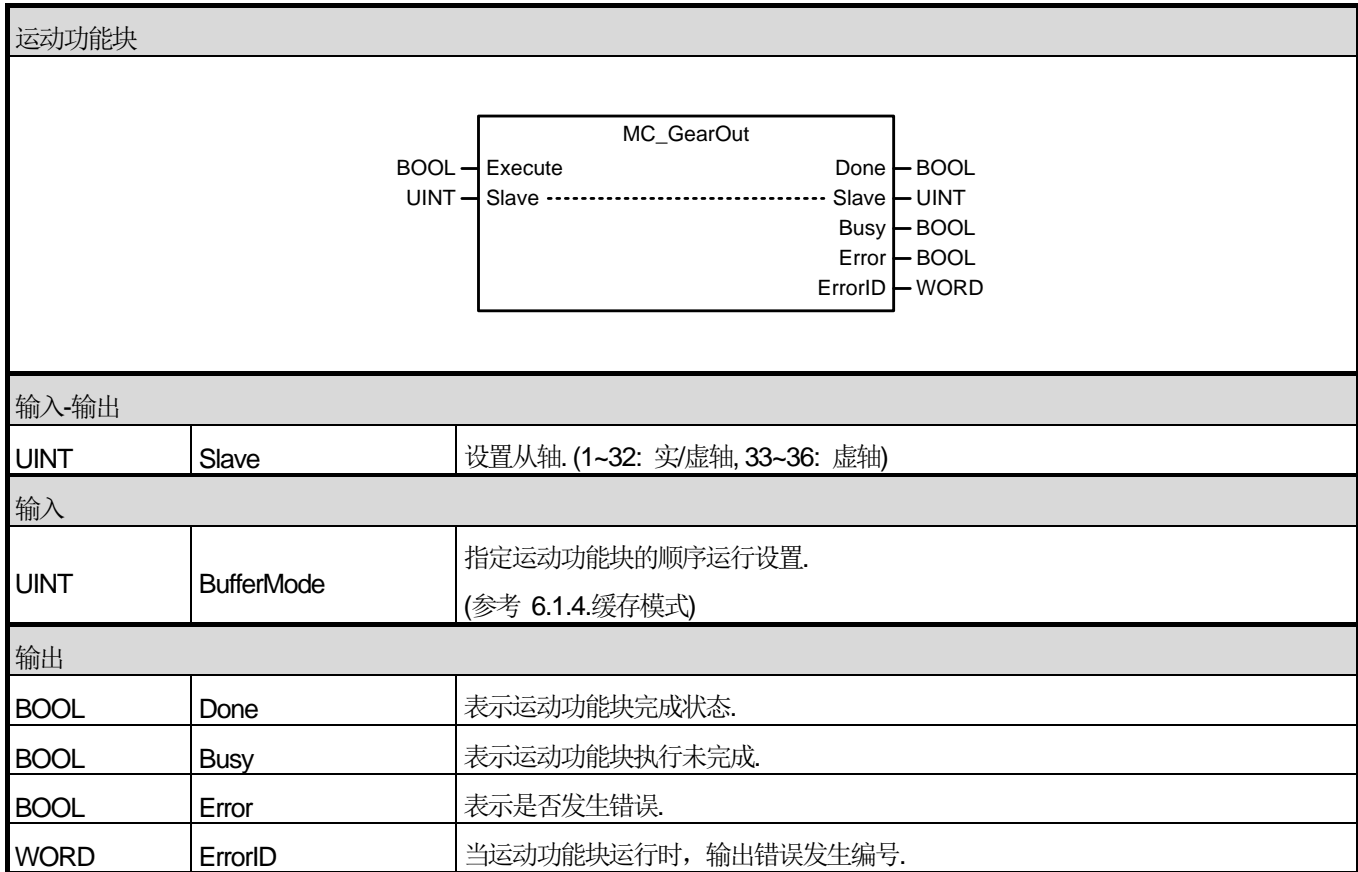
(a) 功能块设置



(b) 时序图

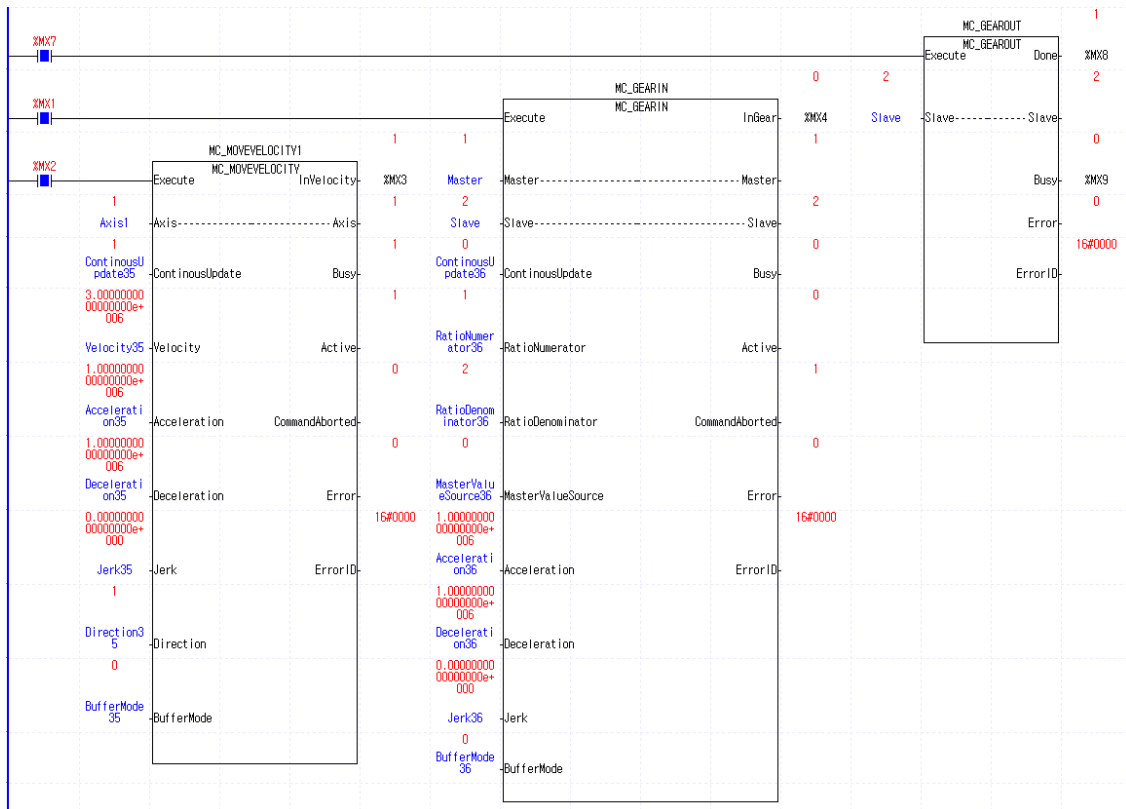


6.4.4 电子传动脱离(MC_GearOut)

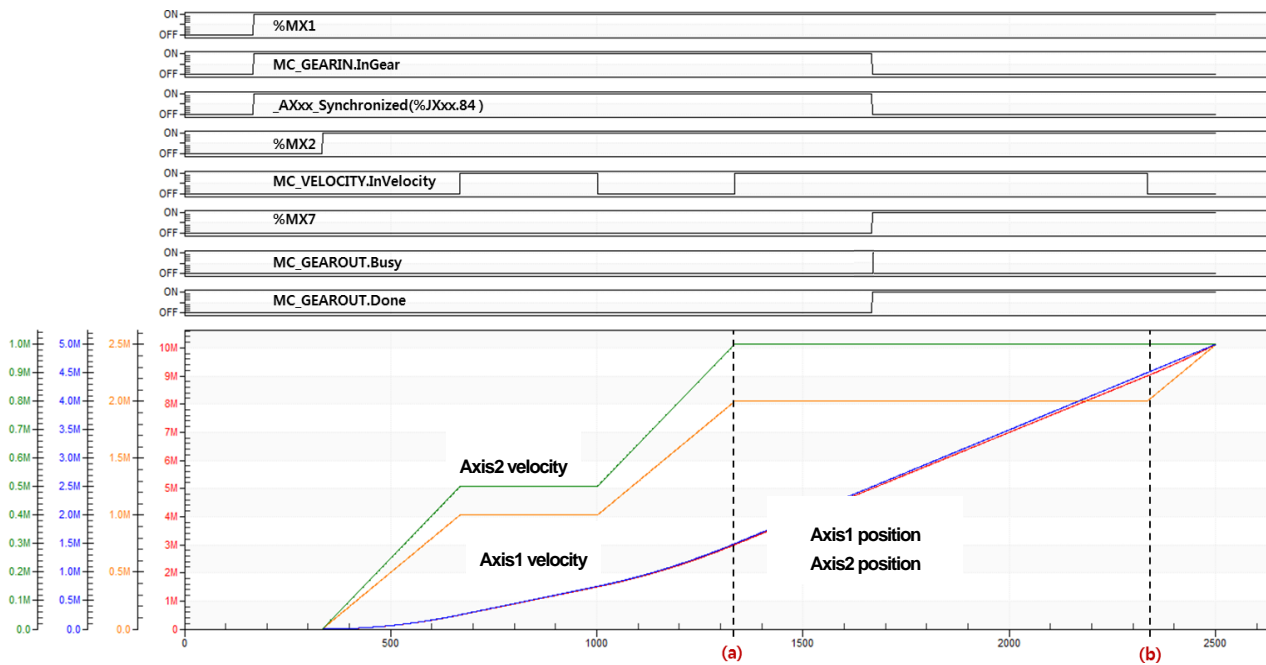


- (1) 该运动功能块在主轴中立即脱离齿轮运行。
- (2) 如果凸轮运行时, 运动功能块的主轴缓存模式为 **Aborting**, 齿轮运行自动脱离, 执行相关运动功能块。如果齿轮运行中止 (MC_GearOut) 只执行运动功能块, 相关轴在齿轮运行时, 维持速度执行运行。完成主轴停止, 使用停止 (MC_Halt) 或立即停止 (MC_Stop) 运动功能块。
- (3) 实例程序
 从轴当前位置为 0 时执行 MC GearIn 指令, 主轴 ContinuousUpdate 设置为 1, 然后速率逐步变更(1,000,000 → 2,000,000 → 3,000,000). 该实例显示当主轴速率为 2,000,000 时, 通过执行 MC_GearOut 指令取保从轴速率不再变更(b).

(a) 功能块设置



(b) 时序图

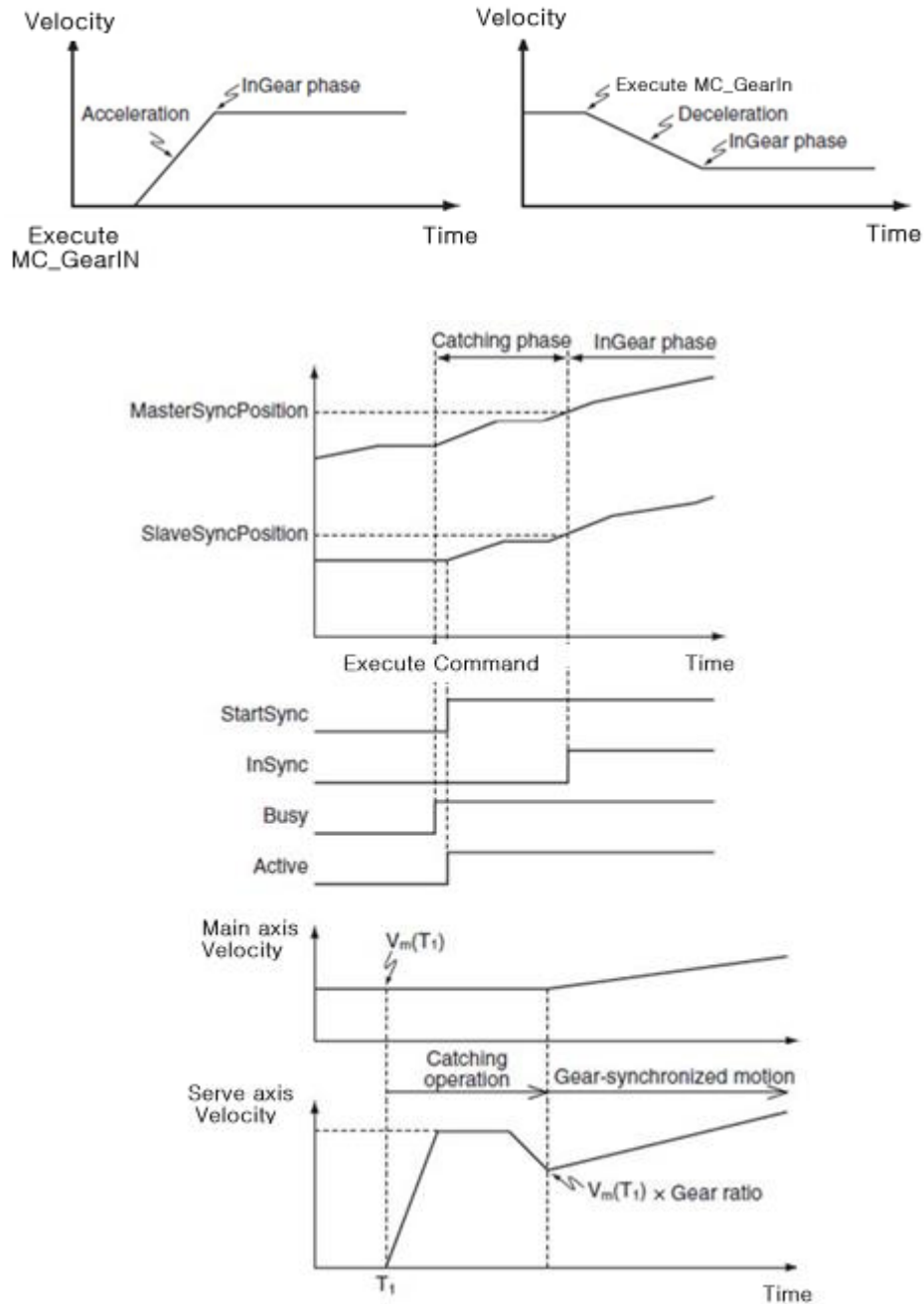


6.4.5 指定位置电子转动(MC_GearInPos)

运动功能块																																
<div style="border: 1px solid black; padding: 10px; margin: 0 auto; width: fit-content;"> <p style="text-align: center; margin: 0;">MC_GearInPos</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%; border-right: 1px solid black; padding: 2px;">BOOL — Execute</td> <td style="width: 33%; padding: 2px;">InSync — BOOL</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">UINT — Master</td> <td style="padding: 2px;">Master — UINT</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">UINT — Slave</td> <td style="padding: 2px;">Slave — UINT</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">INT — RatioNumerator</td> <td style="padding: 2px;">StartSync — BOOL</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">UINT — RatioDenominator</td> <td style="padding: 2px;">Busy — BOOL</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">UINT — MasterValueSource</td> <td style="padding: 2px;">Active — BOOL</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">LREAL — MasterSyncPosition</td> <td style="padding: 2px;">CommandAborted — BOOL</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">LREAL — SlaveSyncPosition</td> <td style="padding: 2px;">Error — BOOL</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">UINT — SyncMode</td> <td style="padding: 2px;">ErrorID — WORD</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">LREAL — MasterStartDistance</td> <td></td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">LREAL — Velocity</td> <td></td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">LREAL — Acceleration</td> <td></td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">LREAL — Deceleration</td> <td></td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">LREAL — Jerk</td> <td></td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">UINT — BufferMode</td> <td></td> </tr> </table> </div>			BOOL — Execute	InSync — BOOL	UINT — Master	Master — UINT	UINT — Slave	Slave — UINT	INT — RatioNumerator	StartSync — BOOL	UINT — RatioDenominator	Busy — BOOL	UINT — MasterValueSource	Active — BOOL	LREAL — MasterSyncPosition	CommandAborted — BOOL	LREAL — SlaveSyncPosition	Error — BOOL	UINT — SyncMode	ErrorID — WORD	LREAL — MasterStartDistance		LREAL — Velocity		LREAL — Acceleration		LREAL — Deceleration		LREAL — Jerk		UINT — BufferMode	
BOOL — Execute	InSync — BOOL																															
UINT — Master	Master — UINT																															
UINT — Slave	Slave — UINT																															
INT — RatioNumerator	StartSync — BOOL																															
UINT — RatioDenominator	Busy — BOOL																															
UINT — MasterValueSource	Active — BOOL																															
LREAL — MasterSyncPosition	CommandAborted — BOOL																															
LREAL — SlaveSyncPosition	Error — BOOL																															
UINT — SyncMode	ErrorID — WORD																															
LREAL — MasterStartDistance																																
LREAL — Velocity																																
LREAL — Acceleration																																
LREAL — Deceleration																																
LREAL — Jerk																																
UINT — BufferMode																																
输入-输出																																
UINT	Master	设置主轴. (1~32: 实/虚轴, 33~36: 虚轴 1~32: 实/虚轴, 33~36: 虚轴, 1001~1002: 编码器)																														
UINT	Slave	设置从轴. (1~32: 实/虚轴, 33~36: 虚轴)																														
输入																																
BOOL	Execute	在上升沿给相关轴齿轮运行命令。																														
INT	RatioNumerator	指定齿轮齿数比分子。(-32768~32767)																														
UINT	RatioDenominator	指定齿轮齿数比分母。(0~65535)																														
UINT	MasterValueSource	选择同步主轴值的标准。 0(mcSetValue): 在主轴的目标位置同步。 1(mcActualValue): 在主轴的当前位置同步。																														
LREAL	MasterSyncPosition	指定主轴齿轮运行启动的位置。																														
LREAL	SlaveSyncPosition	指定轴齿轮运行启动的位置。																														
UINT	SyncMode	未使用																														
LREAL	MasterStartDistance	指定主轴同步启动的距离。																														
LREAL	Velocity	指定轴开始同步的最大速度。[u/s]																														
LREAL	Acceleration	指定轴开始同步的最大加速。[u/s ²]																														
LREAL	Deceleration	指定轴开始同步的最大减速。[u/s ²]																														
LREAL	Jerk	指定加/减速变化率。[u/s ³]																														

UINT	BufferMode	指定运动功能块的顺序运行设置。 (参考 6.1.4.缓存模式)
输出		
BOOL	InSync	表示齿轮运行正常，以指定齿轮率运行。
BOOL	StartSync	表示同步启动。
BOOL	Busy	表示运动功能块执行未完成。
BOOL	Active	表示当前运动功能块控制相关轴。
BOOL	CommandAborted	表示当前运动功能块在运行时被中断。
BOOL	Error	表示是否发生错误。
WORD	ErrorID	当运动功能块运行时，输出错误发生编号。

- (1) 该运动功能块是主轴速度同步运行，轴的位置设置取决于指定位置的 gear 率。
- (2) 给 gear 运行的轴中止 (MC_GearOut)命令或其他允许停止 gear 运行的运动功能块。
- (3) RatioNumerator 和 RatioDenominator 分别设置应用于轴的 gear 率的分子和分母。如果分子设置为负数，轴的旋转方向和主轴相反。
- (4) MasterValueSource 选择主轴同步来源。如果设置为 0 (mcSetValue)，同步执行把主轴的目标位置放入当前监控控制周期作为来源。如果设置为 1(mcActualValue)，同步执行把主轴当前位置回馈到当前运动控制周期。其它设置值发生 "error 0x10D1"。
- (5) 主轴的输入位置和轴的 gear 运行同步完成，分别在 MasterSyncPosition 和 SlaveSyncPosition 输入。在 MasterStartDistance 输入轴同步启动的距离，轴同步启动的位置在 MasterStartDistance 输入远离在 MasterSyncPosition 输入的位置。
- (6) 一旦同步启动，StartSync 输出为 On。当同步完成时，gear 运行启动，StartSync 输出为 Off，InSync 输出为 On。
- (7) 如果该运动功能块由其他指令(最近执行指令缓存模式=0)，凸轮运行停止，CommandAborted 输出为 on。
- (8) 当运动功能块执行停止时如果通过缓存执行其他指令(最近执行指令缓存模式=1~5)，齿轮运行状态(InGear 相)为停止，然后运行最近执行指令。
功能块 InSync / Busy / Active / CommandAborted / Error 输出为 Off。
- (9) 当运行该运动功能块时，轴为 'SynchronizedMotion'。

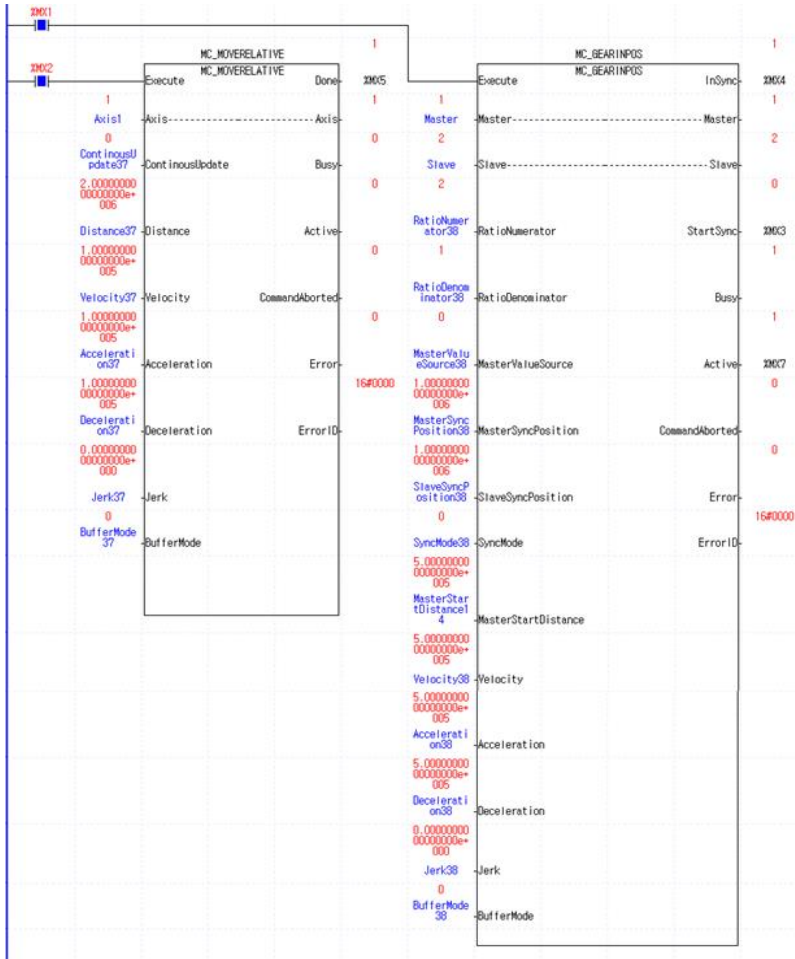


(10) 指令完成前通过再次执行功能块(执行输入为On)应用变更参数. 仅RatioNumerator, RatioDenominator, MasterSyncPosition, SlaveSyncPosition, MasterStartDistance, Velocity, Acceleration, Deceleration输入可以更新. (然而,InGear=On时, RatioNumerator, RatioDenominator 输入可以更新).

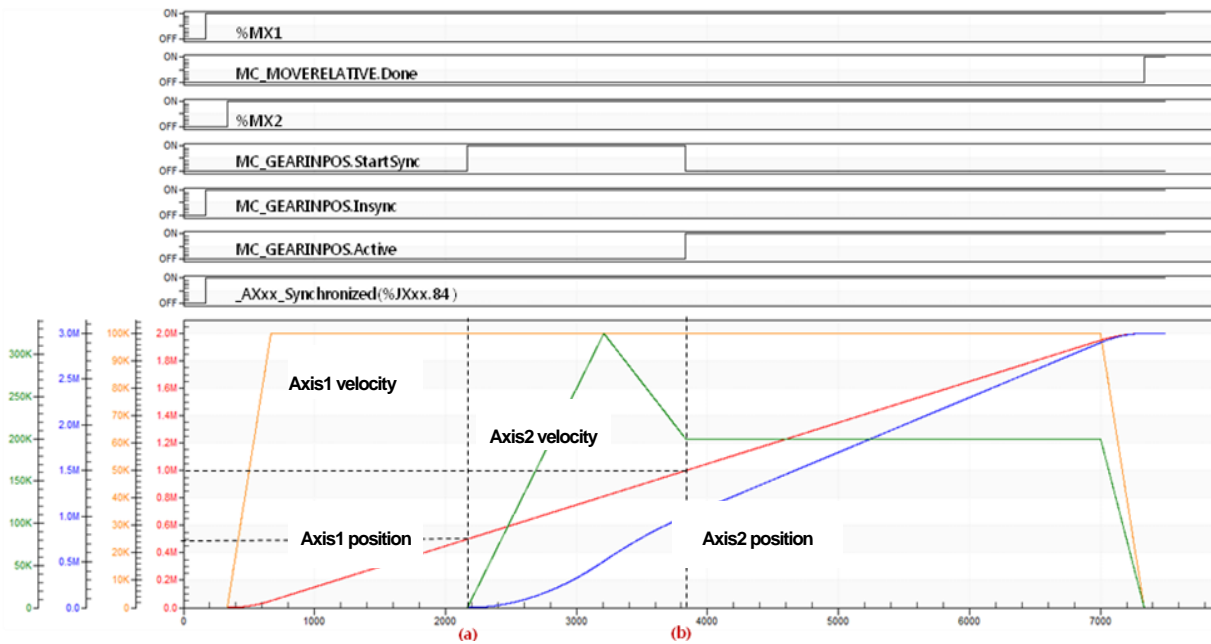
(11) 实例程序

实例程序执行 MC_GearInPos 功能块从 MasterSyncPosition(1,000,000)到 MasterStartDistance(500,000)距离相同的一个位置从轴同步启动,执行 MC_MoveRelative 相对移动到 2,000,000 位置. 一旦同步启动, StartSync 输出为 on(a), 当同步完成和齿轮运行启动时, StartSync 输出为 off, InSync 输出为 on. (b)

(a) 功能块设置



(b) 时序图

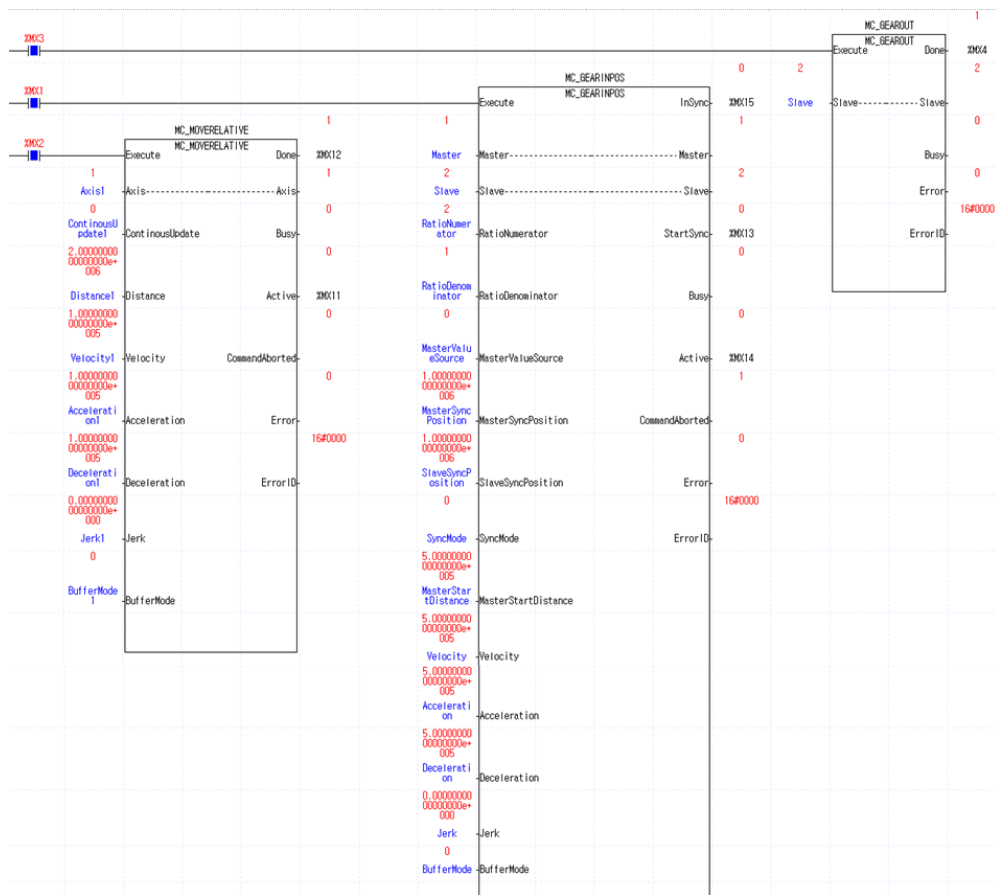


第 6 章运动功能块

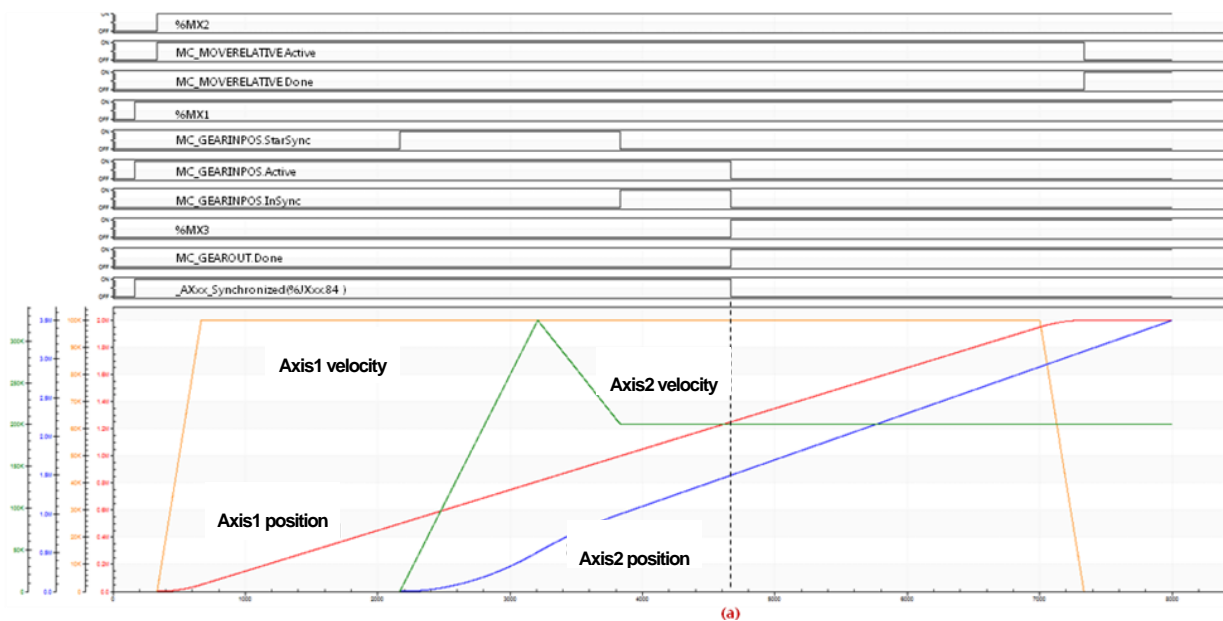
(12) 应用实例程序

该实例程序显示当在基本实例程序中显示移动期间 MC_GearOut 指令发布到 2 轴时, MC_GearInPos 激活, InSync 为 off, 并且齿轮运行停止. (齿轮运行终止可以通过 1 轴停止和 2 轴持续运行进行验证)

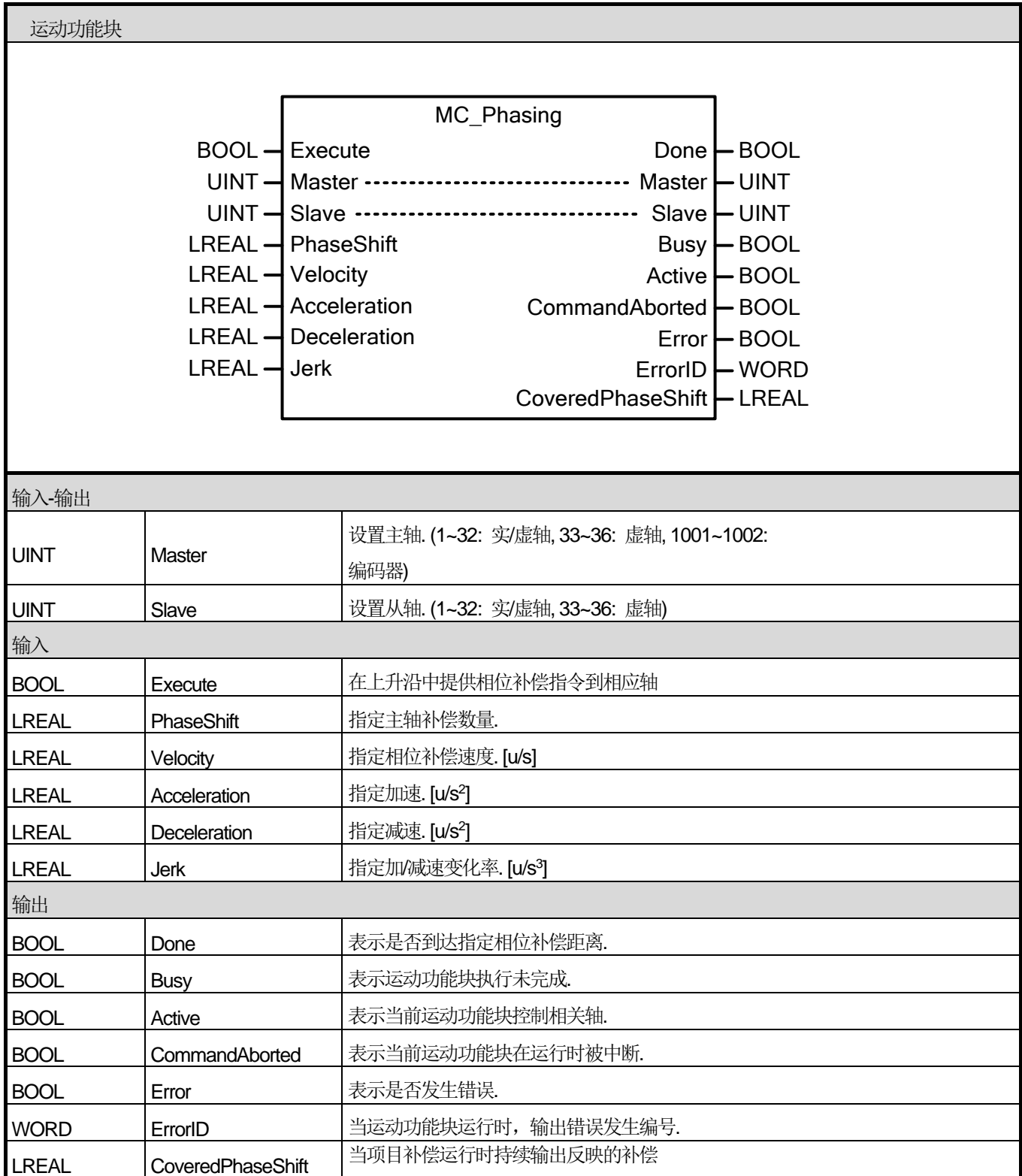
(a) 功能块设置



(b) 时序图



6.4.6 相位补偿 (MC_Phasing)



- (1) 该运动功能块在同步控制运行期间执行轴相位校正. 主轴位置相位校正参考同步控制运行的从轴位置, 执行从轴同步控制运行校正主轴位置.
- (2) 一旦执行相位校正指令, 主轴当前位置为使用在相移设置的相移-速率/加速/减速/加速度的相位校正.
- (3) 相位校正不可以改变实际指令位置或者主轴当前位置. 主轴位置的相位校正执行参考同步控制运行的从轴位置. 也就是说, 主轴

第 6 章运动功能块

无法知晓从轴的相位校正执行。

- (4) 在完成指令前, 可以通过再次执行功能块(执行输入为 on) 再次执行相同数量的相位校正. 也就是说, 相移是一个从执行点开始的相对值.
- (5) 执行相位校正后, 当达到相移时, Done 输出为 on.

6.5 组运动功能块

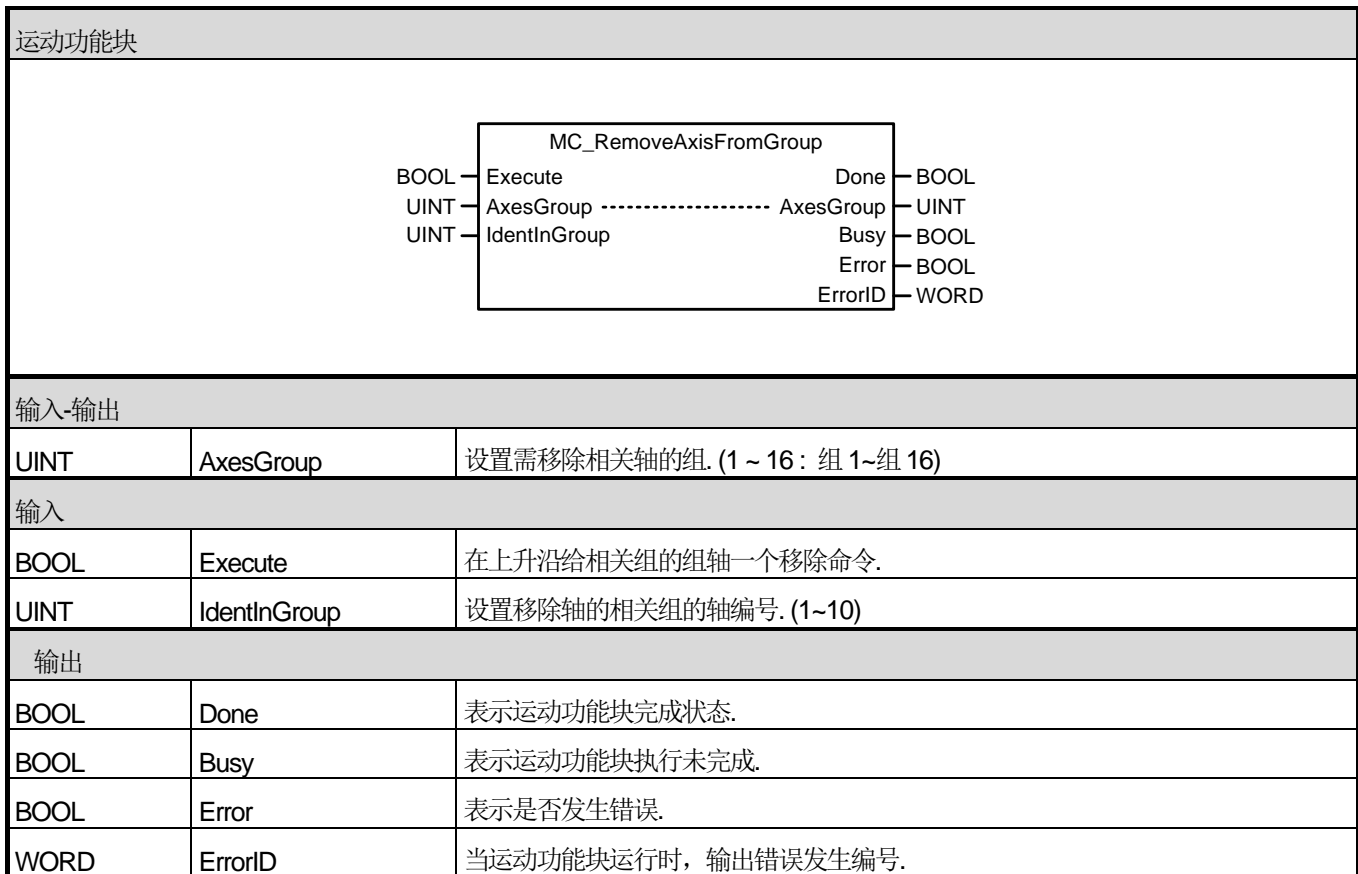
6.5.1 在组 AxesGroup 结构中增加一个轴(MC_AddAxisToGroup)



- (1) 该运动功能块在 AxesGroup 输入中增加指定轴到轴组。
- (2) 组轴中每个轴的 IdentInGroup 必须具有唯一的值。(每个轴的 ID 不能相同)每个轴组中最大包括 10 个轴，轴的 ID 指定范围为 1-10。如果指定轴的数目超出范围，发生“error 0x0006”，如果数目重复，出现“error 0x2051”。
- (3) 轴组设置可以与在 MP500 轴组参数设置中相同的方式执行。

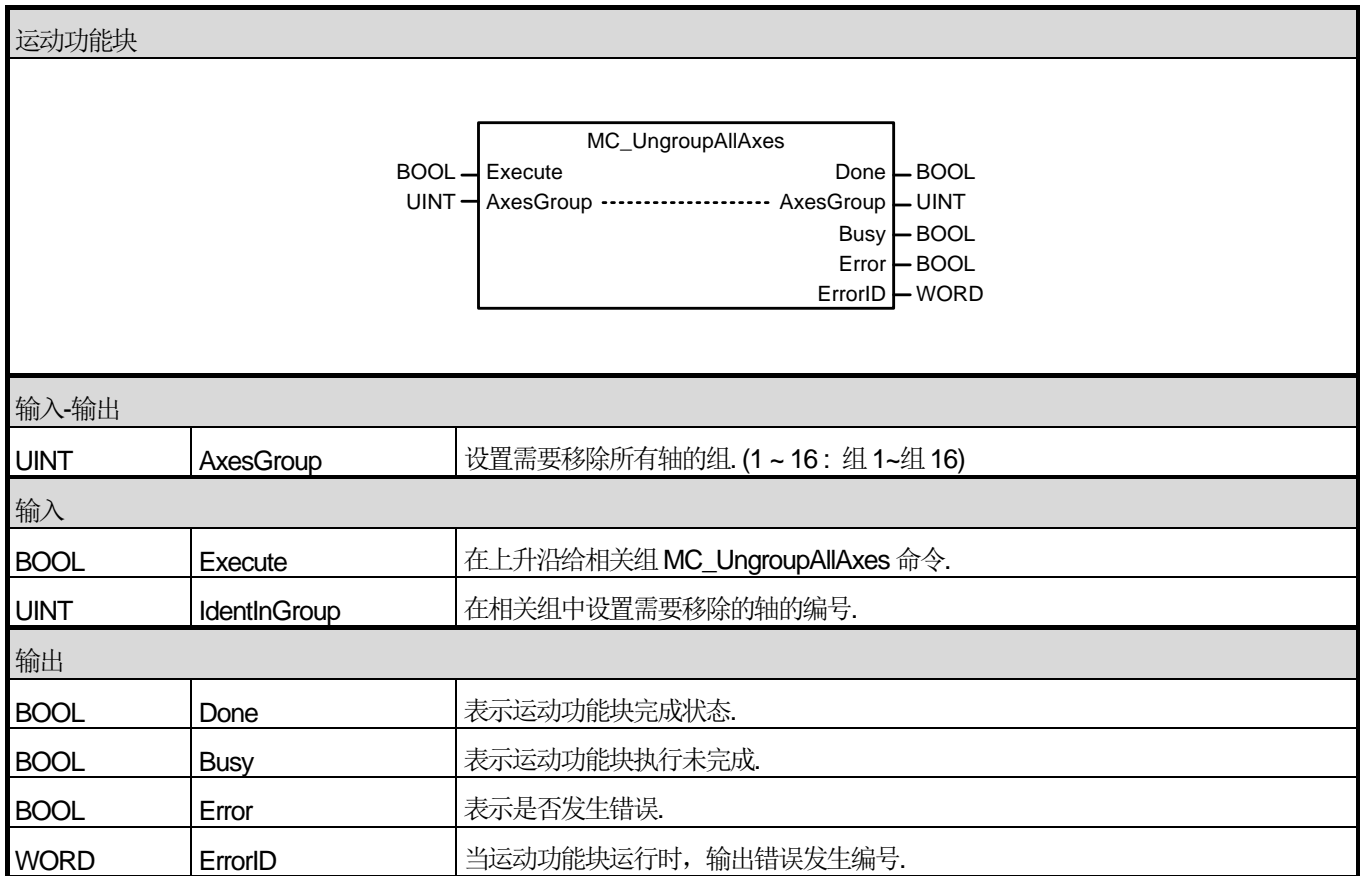
Group	Name	Axis group 1
Axis Group Parameter	Axis 01	0: None
	Axis 02	0: None
	Axis 03	0: None
	Axis 04	0: None
	Axis 05	0: None
	Axis 06	0: None
	Axis 07	0: None
	Axis 08	0: None
	Axis 09	0: None
	Axis 10	0: None
	Intp. speed Max	2000000 u/s

6.5.2 在 AxesGroup 组结构中移除一个轴(MC_RemoveAxisFromGroup)



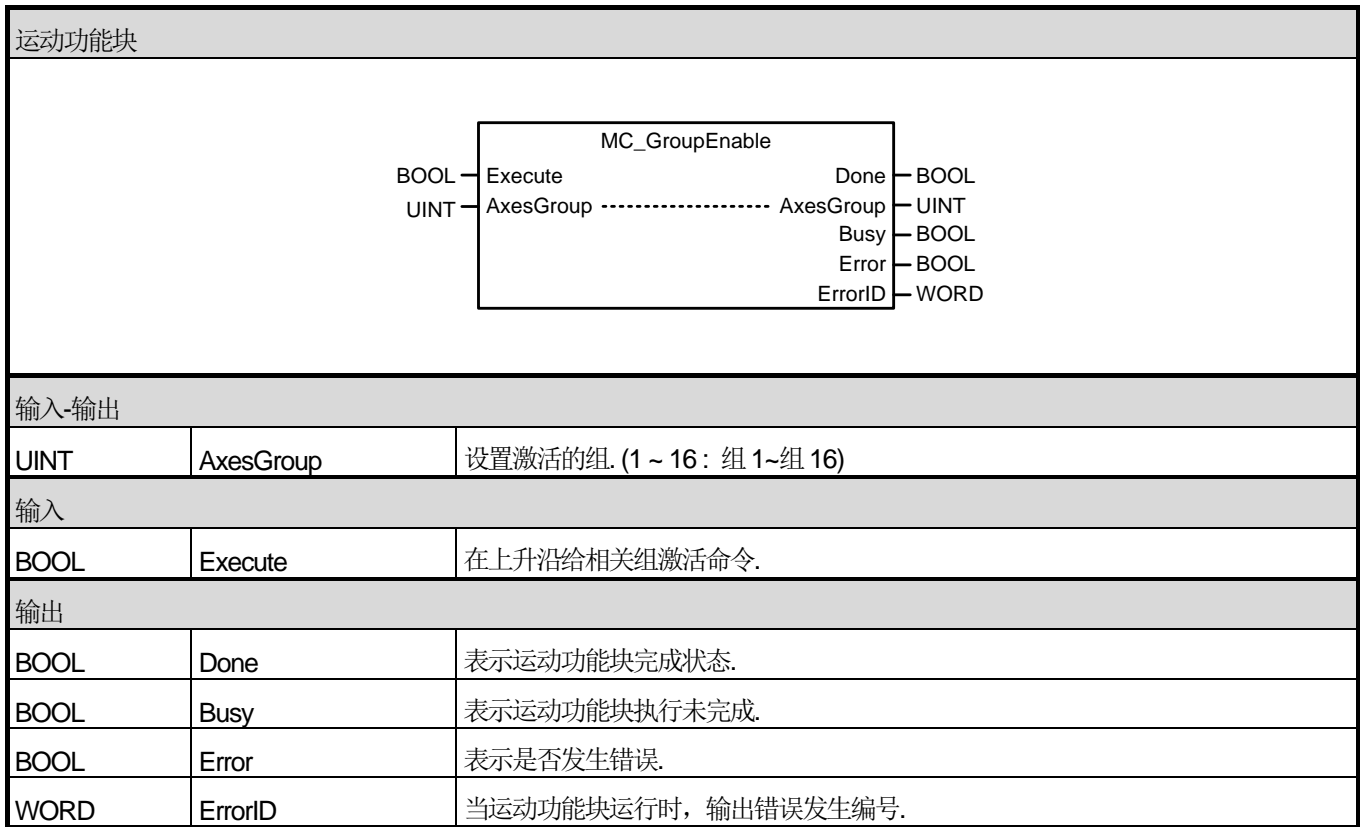
- (1) 该运动功能块在指定 AxesGroup 输入的轴组中 IdentInGroup 移除轴。
- (2) 当轴组非 GroupDisabled, GroupStandBy 和 GroupErrorStop 状态时, 如果组轴排斥执行, 出现"error 0x2003 or 0x2004 or 0x2005" 轴不移除。也就是说, 当轴组没完全停止时, 轴不能移除。

6.5.3 从组 AxesGroup 中移除所有轴(MC_UngroupAllAxes)



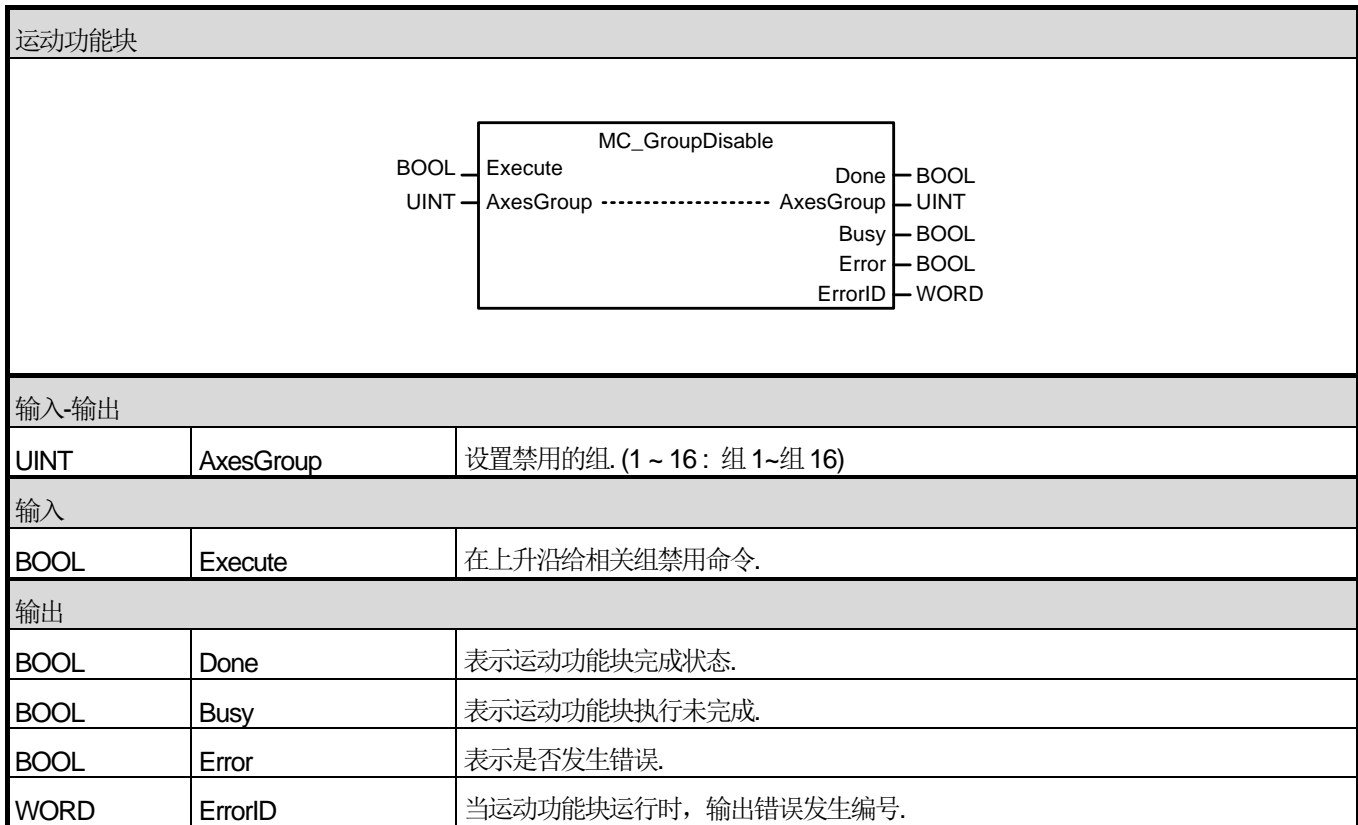
- (1) 该运动功能块移除的所有轴属于的轴组在 AxesGroup 输入中指定。.
- (2) 当轴组不是 GroupDisabled, GroupStandBy 和 GroupErrorStop 状态, 如果该运动功能块执行出现 "error 0x2003 or 0x2004 or 0x2005" 轴不移除。也就是说, 当轴组没完成停止时, 轴不能移除。
- (3) 当属于指定组的轴成功移除后, 相关组转换到 GroupDisabled 状态。

6.5.4 从 GroupDisabled 到 GroupEnable 改变组的状态(MC_GroupEnable)



- (1) 该运动功能块激活在 AxesGroup 输入的轴组。
- (2) 在 GroupDisable 状态中给轴组该命令时, 相关轴组转换到 GroupStandby 状态。
- (3) 该运动功能块不影响相关组中每个轴的电源状态。

6.5.5 改变组为 GroupDisabled 状态(MC_GroupDisable)



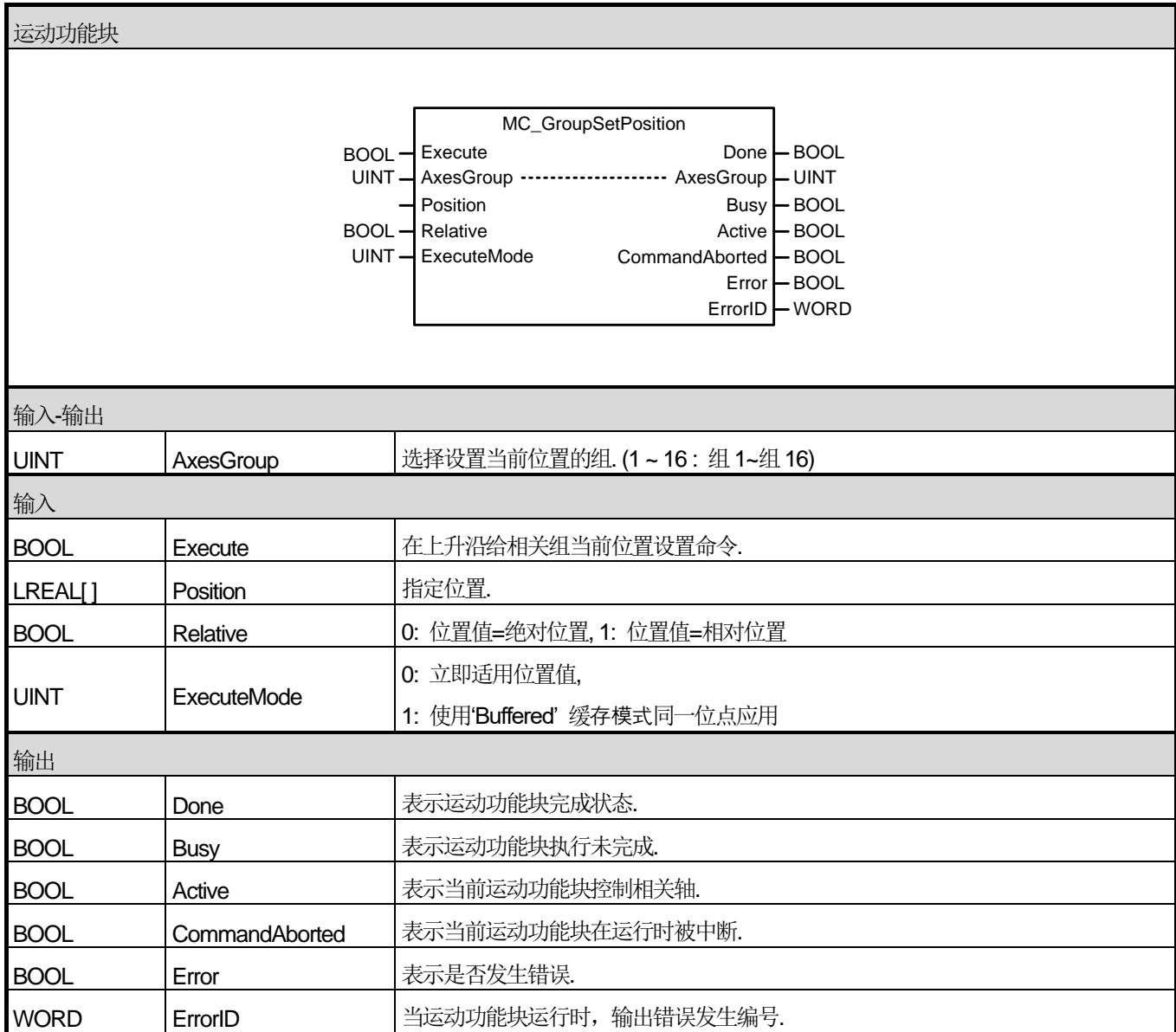
- (1) 该运动功能块在 AxesGroup 输入禁用指定的轴组。
- (2) 轴组执行该运动功能块转换到 GroupDisabled。
- (3) 该运动功能块不影响相关组中每个轴的电源状态。

6.5.6 AxesGroup 执行搜索原点顺序(MC_GroupHome)

运动功能块		
输入-输出		
UINT	AxesGroup	设置组返回原点. (1 ~ 16: 组 1~组 16)
输入		
BOOL	Execute	在上升沿给相关组原点返回命令。
LREAL[]	Position	当检测到参考信号, 指定每个轴的绝对位置。
UINT	BufferMode	指定运动功能块的顺序运行设置. (参考 6.1.4.缓存模式)
输出		
BOOL	Done	表示运动功能块完成状态.
BOOL	Busy	表示运动功能块执行未完成.
BOOL	Active	表示当前运动功能块控制相关轴.
BOOL	CommandAborted	表示当前运动功能块在运行时被中断.
BOOL	Error	表示是否发生错误.
WORD	ErrorID	当运动功能块运行时, 输出错误发生编号.

- (1) 该运动功能块给 AxesGroup 输入的轴组原点返回命令。
- (2) 返回方式按照预先设置在相关轴的伺服参数中运行。
- (3) 在 Position 输入, 当原点返回完成或检测到参考信号时, 指定绝对位置数列。数列中的值和组中的轴顺序[1, 2, 3, 4...9,10] (1~10 为轴组中的轴 ID)一致。
- (4) 当运动功能块正在运行时, 轴组为'GroupHoming'状态, 当运动功能块完成后, 转换为'GroupStandby'状态。

6.5.7 不移动设置组中所有轴的位置(MC_GroupSetPosition)



- (1) 该运动功能块设置相关轴组的当前位置.
- (2) 在组排列中指定每个轴的位置。当执行该运动功能块时, 如果 **Relative** 输入为 **Off**, 相关轴的位置被替换为 **Position** 输入值, 如果 **Relative** 输入为 **On**, **Position** 输入值增加到相关轴的当前位置。数列中的值和组中的轴顺序[1, 2, 3, 4...9,10] (1~10 为轴组中的轴 ID)一致

第 6 章运动功能块

(3) ExecutionMode 输入指定设定点。如果为 0，执行命令后立即设置，如果为 1，以 'Buffered' 顺序在同一位点运行。当发生 "error 0x201B" 不能设置值。

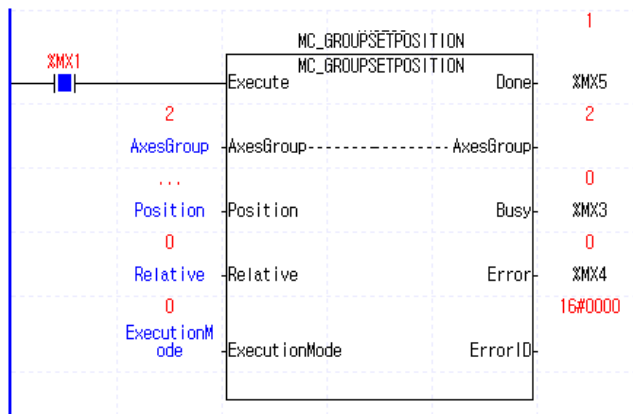
0 (mcImmediately): 执行运动功能块后，立即改变参数值。(在执行输入上升沿)。如果相关轴正在运行，运行可能受影响。

1 (mcQueued): 'Buffered' 缓存模式在同一位点改变。(参考 6.1.4.缓存模式)。

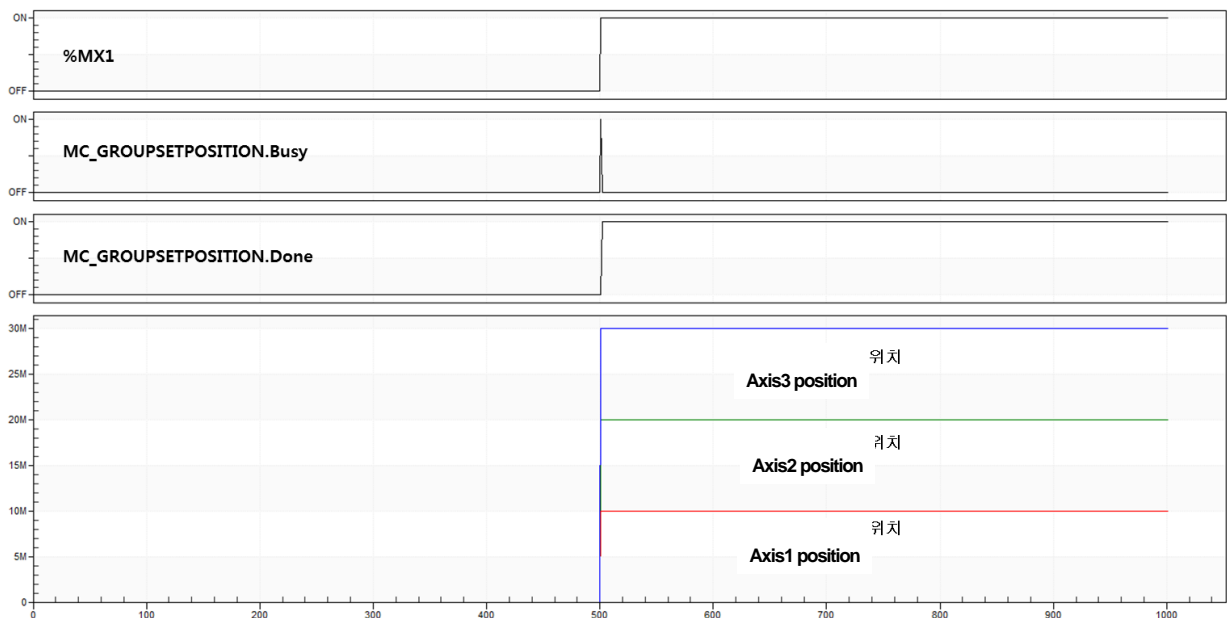
(4) 实例程序

该实例为当执行 MC_GroupSetPosition 功能块在 1 轴，2 轴和 3 轴设置为单独组的状态时，显示当前位置到位置变量中设置值(10,000,000/20,000,000/30,000,000)的变化。

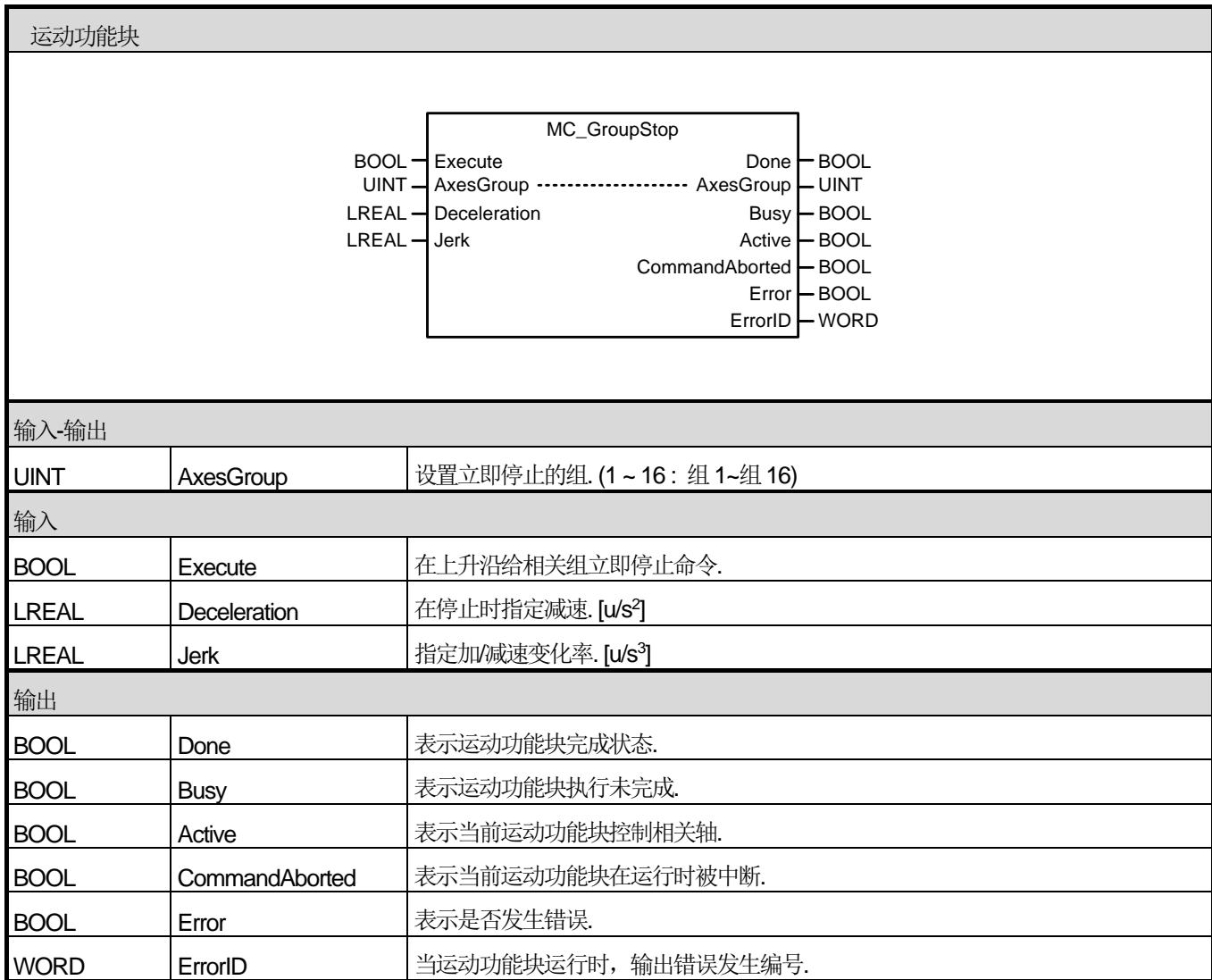
(a) 功能块设置



(b) 时序图



6.5.8 立即停止组(MC_GroupStop)



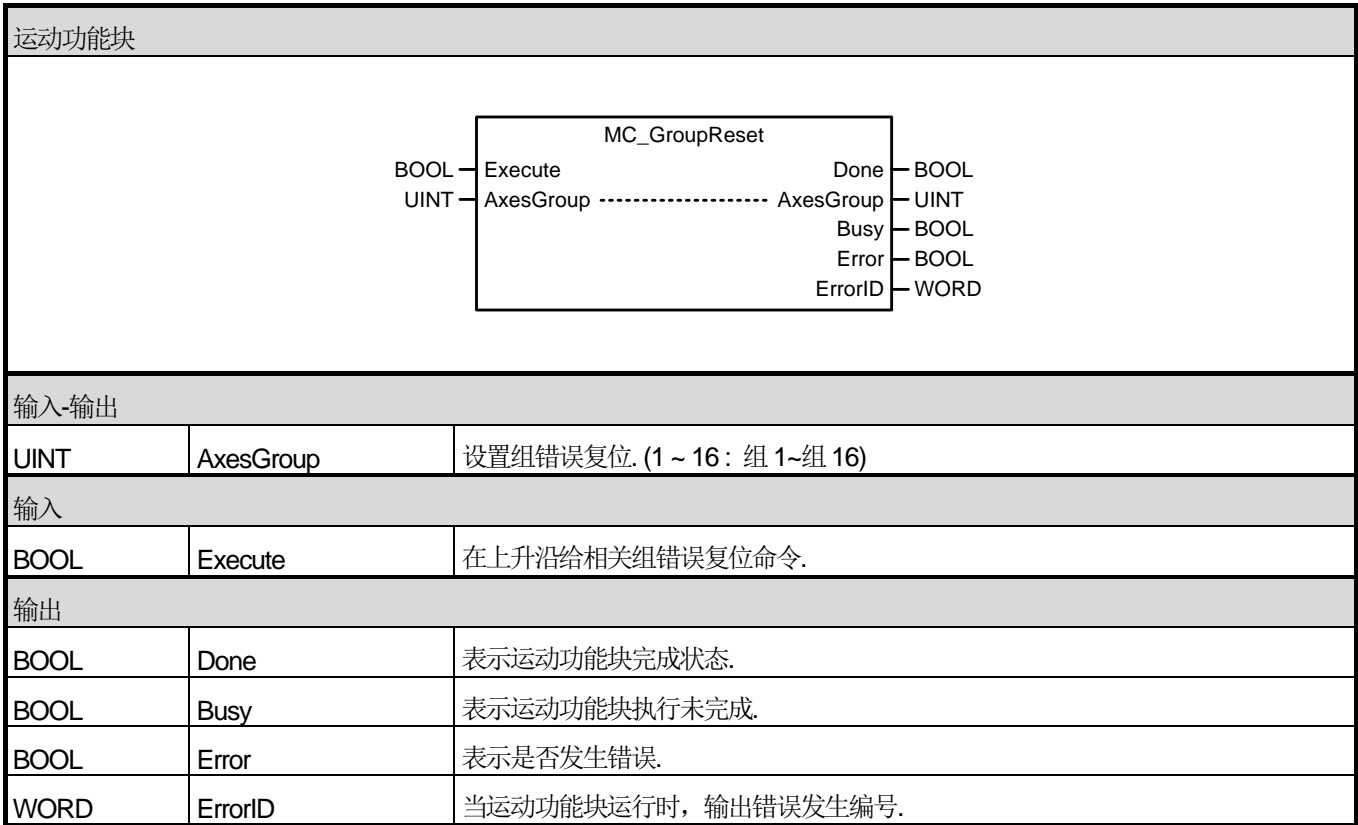
- (1) 该运动功能块给相关轴组一个紧急停止命令。
- (2) 相关轴组在完全停止前遵循移动路径。
- (3) 当执行立即停止(MC_GroupStop)运动功能块, 运动功能块的相关功能块执行中断, 轴变为 'GroupStopping'。当相关轴组为'GroupStopping'状态, 其它运动功能块在完成停止前, 不能给相关轴命令。
- (4) CommandAborted 输出表示当执行时, 当前运动功能块中断。因为当组立即停止(MC_GroupStop)命令执行时, 其它运动功能块不能中断组立即停止 (MC_GroupStop), 当伺服电源切断后, CommandAborted 输出为 On, 执行 servo Off 命令, 或伺服连接中断。
- (5) 如果 Execute 输入为 On 或轴的速度不为 0, 轴为'GroupStopping'状态, 如果 Done 输出为 On, Execute 输入为 Off, 轴切换为 'GroupStandBy'状态。

6.5.9 停止组 (MC_GroupHalt)



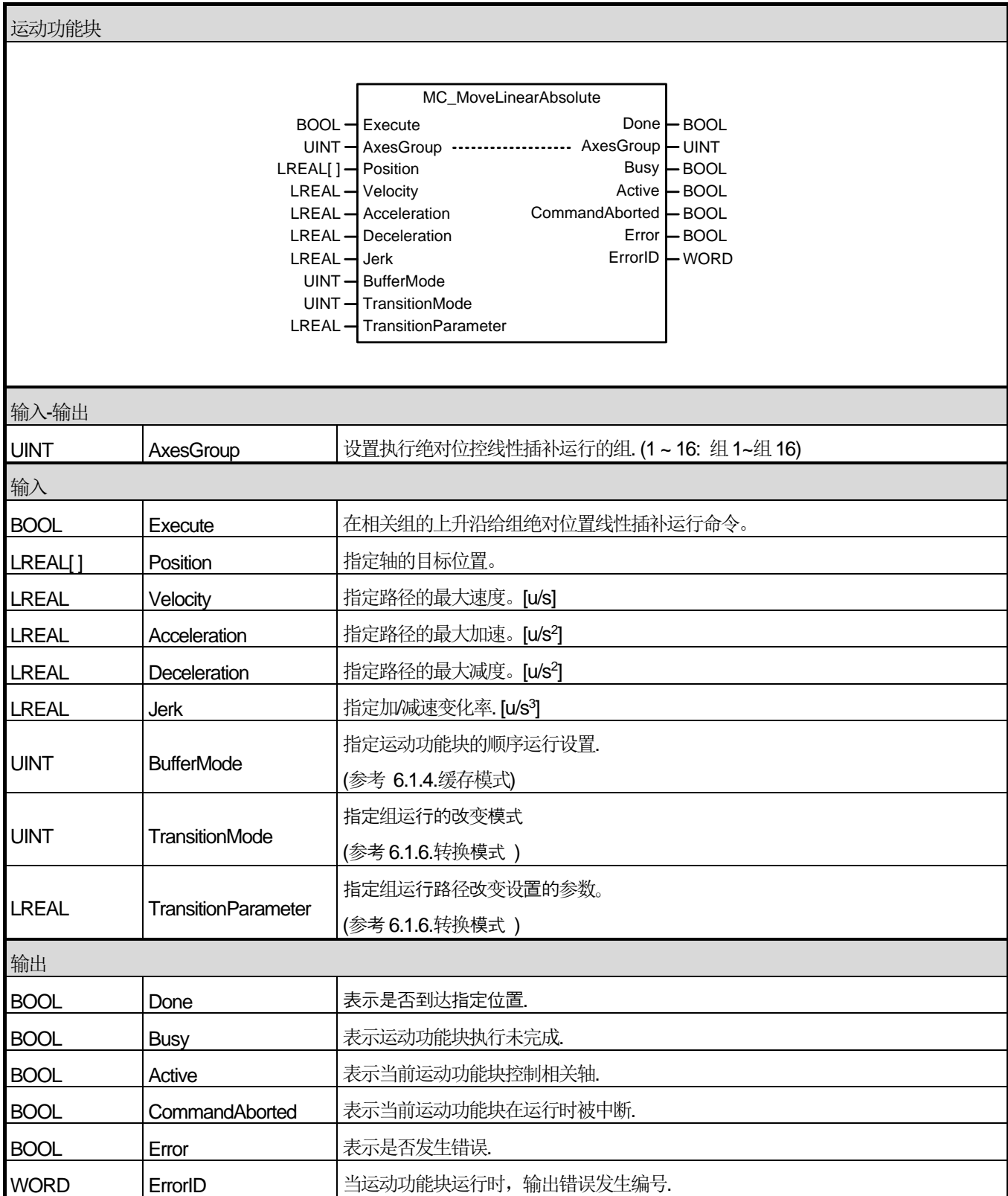
- (1) 该运动功能块给相关轴一个停止命令。
- (2) 相关轴组在完全停止前遵循移动路径。
- (3) 当运动功能块正在运行时, 轴为'GroupMoving'状态, 如果轴组完成停止, 'Done' 输出为 On 组状态改变为 'GroupStandBy'状态。

6.5.10 复位组错误 (MC_GroupReset)



- (1) 该运动功能块复位相关轴组错误。当相关轴为 'GroupErrorStop', 当执行运动功能块, 当前发生错误的轴复位, 轴组转换到 'GroupStandBy'状态。
- (2) 但执行该运动功能块, 该组中发生错误的轴复位。(当执行轴复位(MC_Reset), 效果相同。)

6.5.11 绝对位控线性插补运行(MC_MoveLinearAbsolute)



- (1) 该运动功能块给 AxesGroup 输入的轴组一个绝对位置线性插补命令。
- (2) 当执行该运动功能块时，中断控制在轴的当前位置到目标位置的支线路径执行，移动方向由轴的起始点和目标点决定。

开始位置<目标位置: 正向运行

开始位置>目标位置: 反向运行

- (3) 在 **Position** 输入中, 指定轴的目标位置作为矩阵。数列中的值和组中的轴顺序[1, 2, 3, 4...9,10] (1~10 为轴组中的轴 ID)一致。
- (4) 在 **Velocity**, **Acceleration**, **Deceleration** 和 **Jerk** 输入中分别指定插补路径的速度, 加速, 减速和加减速的变化率。
- (5) **Velocity** 用于设置轴组的插补速度, 表示轴的综合速度。
每个配置轴的运行速度计算如下。

Interpolation speed (F) = Target speed specified in the Velocity

$$\text{Interpolation movement amount (S)} = \sqrt{S_1^2 + S_2^2 + \dots + S_3^2 + S_4^2}$$

$$\text{Configuration axis 1 speed (V}_1\text{)} = \text{Interpolation speed (F)} \times \frac{\text{Configuration axis 1 movement amount (S}_1\text{)}}{\text{Interpolation movement amount (S)}}$$

$$\text{Configuration axis 2 speed (V}_2\text{)} = \text{Interpolation speed (F)} \times \frac{\text{Configuration axis 2 movement amount (S}_2\text{)}}{\text{Interpolation movement amount (S)}}$$

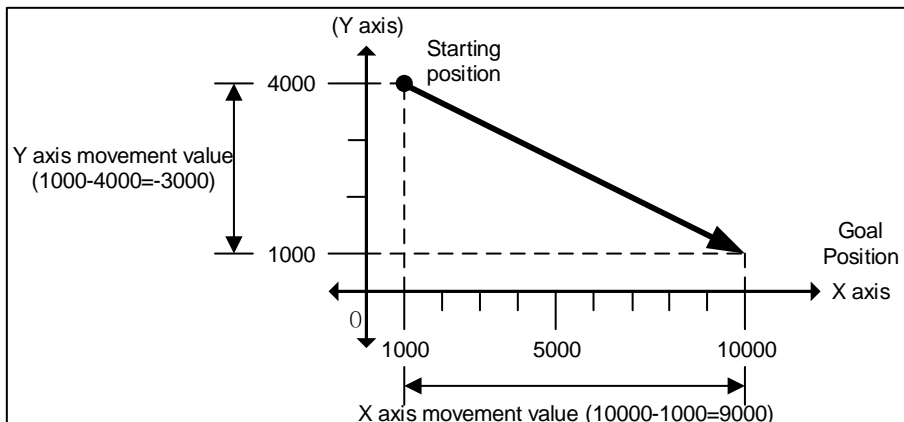
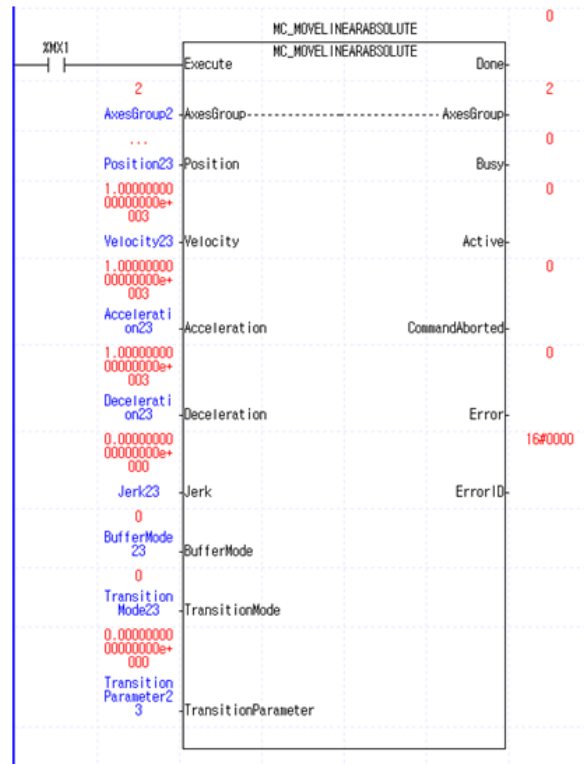
$$\text{Configuration axis 3 speed (V}_3\text{)} = \text{Interpolation speed (F)} \times \frac{\text{Configuration axis 3 movement amount (S}_3\text{)}}{\text{Interpolation movement amount (S)}}$$

$$\text{Configuration axis 4 speed (V}_4\text{)} = \text{Interpolation speed (F)} \times \frac{\text{Configuration axis 4 movement amount (S}_4\text{)}}{\text{Interpolation movement amount (S)}}$$

- (6) 更多内容请参考运动控制模块的 8.2.6 线性插补控制部分。
- (7) 指令完成前通过再次执行功能块(执行输入为 On)应用变更参数。仅 **Velocity**, **Acceleration**, **Deceleration**, **Jerk**, **Position** 输入可以更新。
- (8) 实例程序
该实例显示当当前指令位置为(1000, 4000)时, 线性插值到目标位置(10000, 1000)。

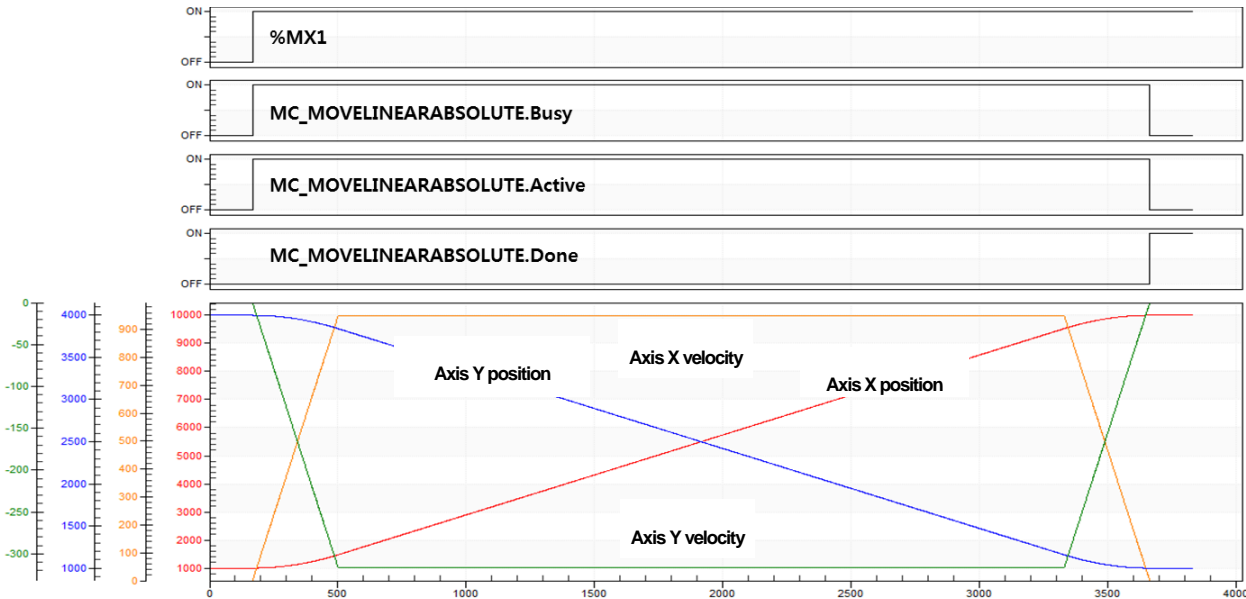
第 6 章运动功能块

(a) 功能块设置

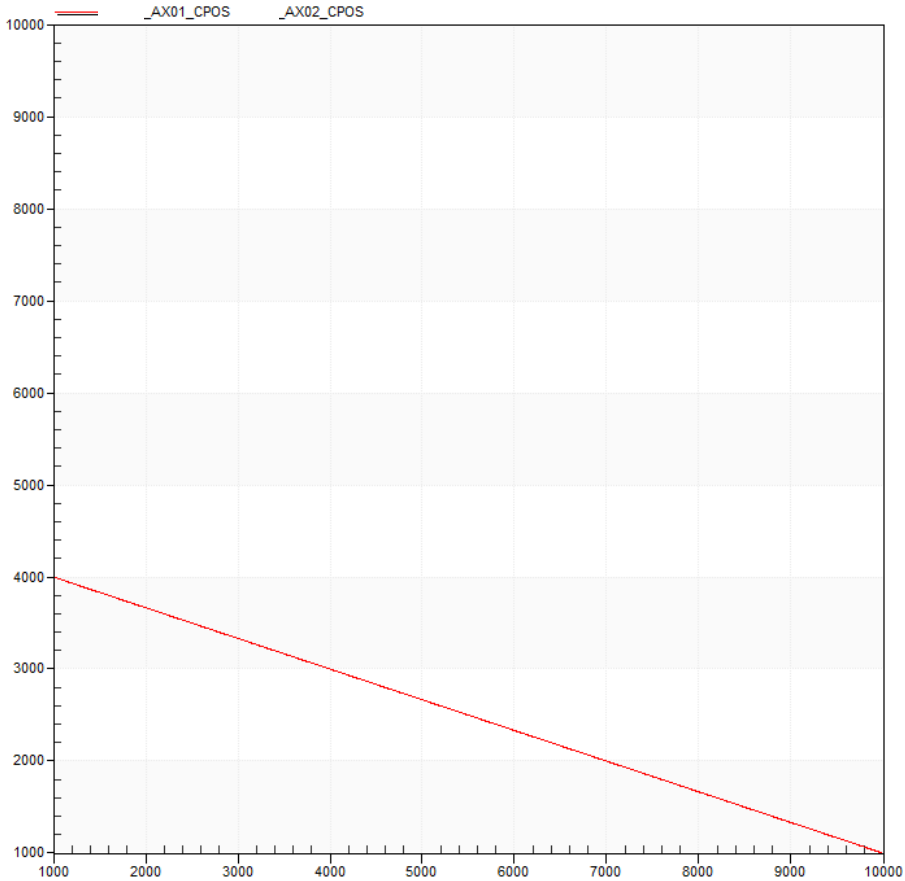


1	<GLOBAL>	%JL1.3	E10	1.0000000000000000e+003	LREAL	_AX01_CPOS
2	<GLOBAL>	%JL2.3	E10	4.0000000000000000e+003	LREAL	_AX02_CPOS
3	<GLOBAL>	%JL1.4	E10	0.0000000000000000e+000	LREAL	_AX01_CVEL
4	<GLOBAL>	%JL2.4	E10	0.0000000000000000e+000	LREAL	_AX02_CVEL
5	Group	GroupHome	E10	On	BOOL	
6	Group	GroupHalt	E10	Off	BOOL	
7	Group	Position23			ARRAY[0..3] OF LREAL	
8	Group	Position23[0]	E10	1.0000000000000000e+004	LREAL	
9	Group	Position23[1]	E10	1.0000000000000000e+003	LREAL	
10	Group	Position23[2]	E10	0.0000000000000000e+000	LREAL	
11	Group	Position23[3]	E10	0.0000000000000000e+000	LREAL	
12						

(b) 时序图



(c) XY graph



6.5.12 相对位置控制线性插补运行(MC_MoveLinearRelative)

运动功能块		
<pre> graph LR subgraph MC_MoveLinearRelative Execute[Execute] AxesGroup[AxesGroup] Distance[Distance] Velocity[Velocity] Acceleration[Acceleration] Deceleration[Deceleration] Jerk[Jerk] BufferMode[BufferMode] TransitionMode[TransitionMode] TransitionParameter[TransitionParameter] Done[Done] Busy[Busy] Active[Active] CommandAborted[CommandAborted] Error[Error] ErrorID[ErrorID] end Execute --- MC_MoveLinearRelative AxesGroup --- MC_MoveLinearRelative Distance --- MC_MoveLinearRelative Velocity --- MC_MoveLinearRelative Acceleration --- MC_MoveLinearRelative Deceleration --- MC_MoveLinearRelative Jerk --- MC_MoveLinearRelative BufferMode --- MC_MoveLinearRelative TransitionMode --- MC_MoveLinearRelative TransitionParameter --- MC_MoveLinearRelative MC_MoveLinearRelative --- Done MC_MoveLinearRelative --- Busy MC_MoveLinearRelative --- Active MC_MoveLinearRelative --- CommandAborted MC_MoveLinearRelative --- Error MC_MoveLinearRelative --- ErrorID </pre>		
输入-输出		
UINT	AxesGroup	设置相对位置线性插补运行组. (1 ~ 16: 组 1~组 16)
输入		
BOOL	Execute	在相关组的上升沿给组绝对位置线性插补运行命令。
LREAL[]	Distance	指定轴的目标位置。
LREAL	Velocity	指定路径的最大速度。[u/s]
LREAL	Acceleration	指定路径的最大加速。[u/s ²]
LREAL	Deceleration	指定路径的最大减度。[u/s ²]
LREAL	Jerk	指定加/减速变化率。[u/s ³]
UINT	BufferMode	指定运动功能块的顺序运行设置。 (参考 6.1.4.缓存模式)
UINT	TransitionMode	指定组运行的改变模式 (参考 6.1.6.转换模式)
LREAL	TransitionParameter	指定组运行路径改变设置的参数。 (参考 6.1.6.转换模式)
输出		
BOOL	Done	表示是否达到指定位置。
BOOL	Busy	表示运动功能块执行未完成。
BOOL	Active	表示当前运动功能块控制相关轴。
BOOL	CommandAborted	表示当前运动功能块在运行时被中断。
BOOL	Error	表示是否发生错误。
WORD	ErrorID	当运动功能块运行时，输出错误发生编号。

- (1) 该运动功能块给 AxesGroup 输入的轴组一个相对位置线性插补命令。
- (2) 当执行该运动功能块时，中断控制在轴的当前位置到目标位置的支线路径执行，移动方向由轴的目标距离标志决定。

目标距离> 0: 正向运行

目标距离< 0: 反向运行

- (3) 在 **Distance** 输入中, 指定轴的目标距离作为数组。指定数组和组轴的轴 ID 顺序一致。[ID1 目标举例, ID2 目标举例, ...]。
- (4) 在 **Velocity**, **Acceleration**, **Deceleration** 和 **Jerk** 输入中分别指定插补路径的速度, 加速, 减速和加减速的变化率。
- (5) **Velocity** 用于设置轴组的插补速度, 表示轴的综合速度。

每个配置轴的运行速度计算如下。

Interpolation speed (F) = Target speed specified in the Velocity

$$\text{Interpolation movement amount (S)} = \sqrt{S_1^2 + S_2^2 + \dots + S_3^2 + S_4^2}$$

$$\text{Configuration axis 1 speed (V}_1\text{)} = \text{Interpolation speed (F)} \times \frac{\text{Configuration axis 1 movement amount (S}_1\text{)}}{\text{Interpolation movement amount (S)}}$$

$$\text{Configuration axis 2 speed (V}_2\text{)} = \text{Interpolation speed (F)} \times \frac{\text{Configuration axis 2 movement amount (S}_2\text{)}}{\text{Interpolation movement amount (S)}}$$

$$\text{Configuration axis 3 speed (V}_3\text{)} = \text{Interpolation speed (F)} \times \frac{\text{Configuration axis 3 movement amount (S}_3\text{)}}{\text{Interpolation movement amount (S)}}$$

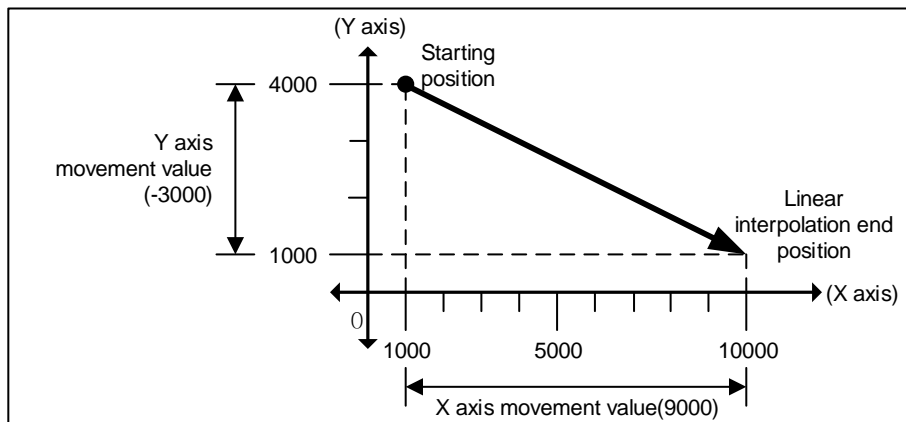
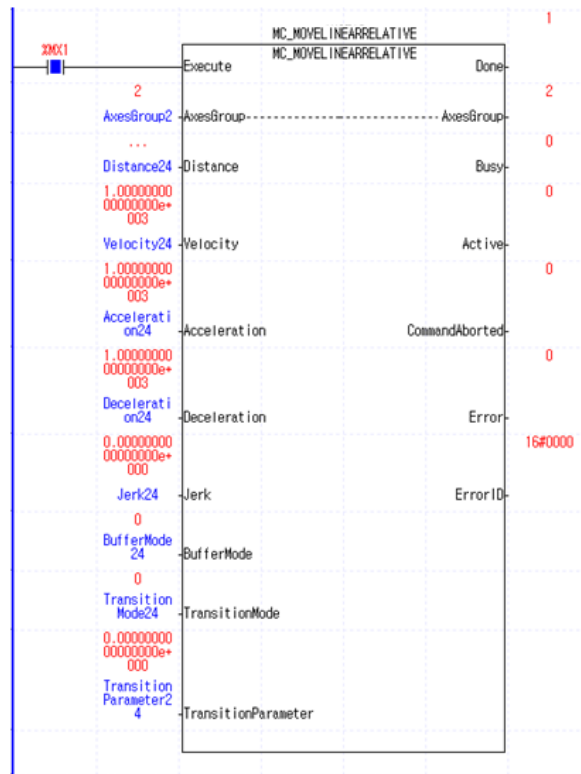
$$\text{Configuration axis 4 speed (V}_4\text{)} = \text{Interpolation speed (F)} \times \frac{\text{Configuration axis 4 movement amount (S}_4\text{)}}{\text{Interpolation movement amount (S)}}$$

- (6) 更多内容请参考运动控制模块的 8.2.6 线性插补控制部分。
- (7) 指令完成前通过再次执行功能块(执行输入为 On)应用变更参数。仅 **Velocity**, **Acceleration**, **Deceleration**, **Jerk**, **Position** 输入可以更新。
- (8) 实例程序

该实例为当前指令位置为(1000, 4000), 通过移动目标距离(X-轴: 9000, Y-轴: -3000)线性插值到目标位置 (10000, 1000)。

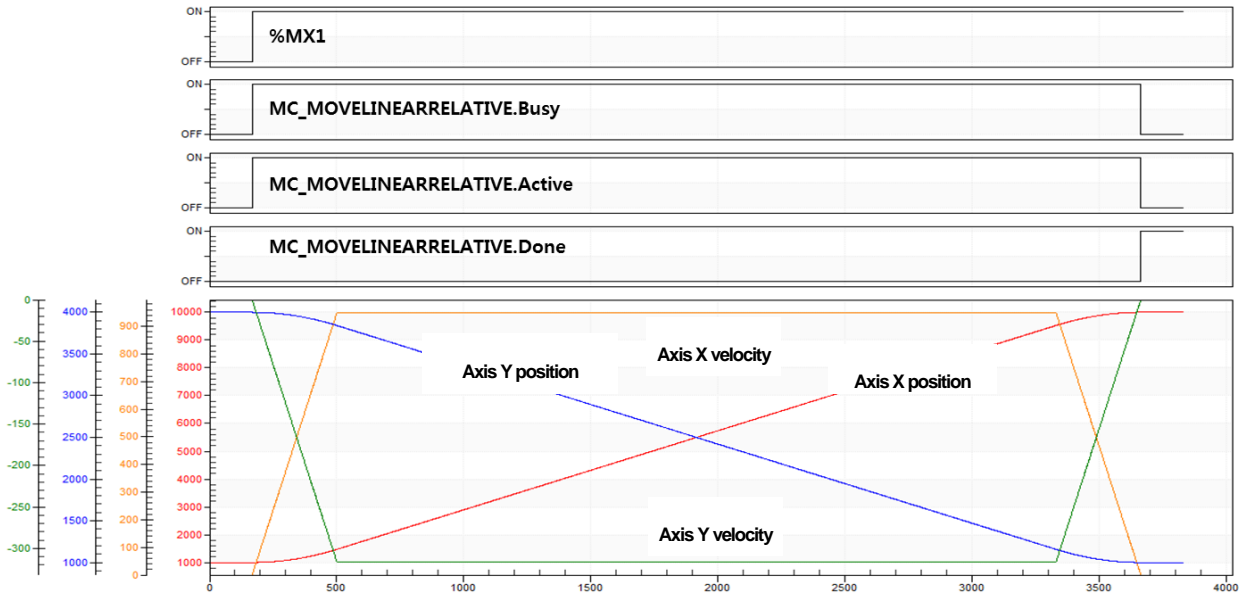
第 6 章运动功能块

(a) 功能块设置

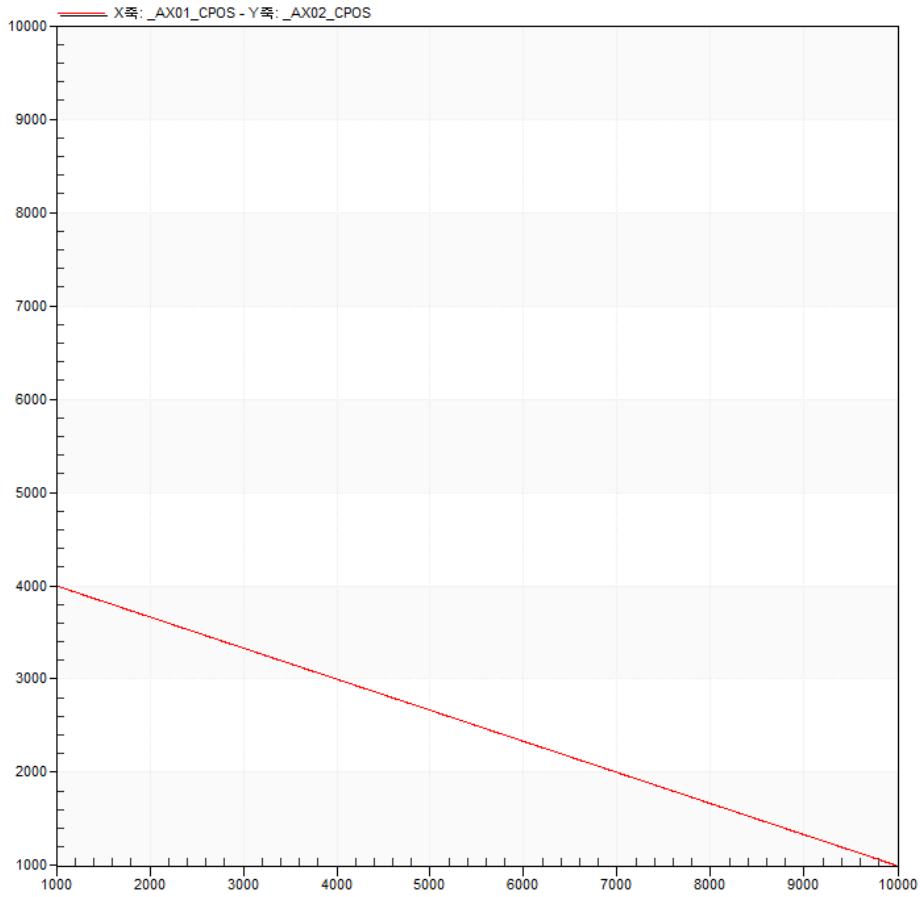


1	<GLOBAL>	%JL1.3	E10	1.000000000000000e+004	LREAL	_AX01_CPOS
2	<GLOBAL>	%JL2.3	E10	1.000000000000000e+003	LREAL	_AX02_CPOS
3	<GLOBAL>	%JL1.4	E10	0.000000000000000e+000	LREAL	_AX01_CVEL
4	<GLOBAL>	%JL2.4	E10	0.000000000000000e+000	LREAL	_AX02_CVEL
5	Group	GroupHome	E10	On	BOOL	
6	Group	GroupHalt	E10	Off	BOOL	
7	Group	Distance24			ARRAY[0..3] OF LREAL	
8	Group	Distance24[0]	E10	9.000000000000000e+003	LREAL	
9	Group	Distance24[1]	E10	-3.000000000000000e+003	LREAL	
10	Group	Distance24[2]	E10	0.000000000000000e+000	LREAL	
11	Group	Distance24[3]	E10	0.000000000000000e+000	LREAL	

(b) 时序图



(c) XY graph



第 6 章运动功能块

6.5.13 绝对位控圆环插补运行(MC_MoveCircularAbsolute)

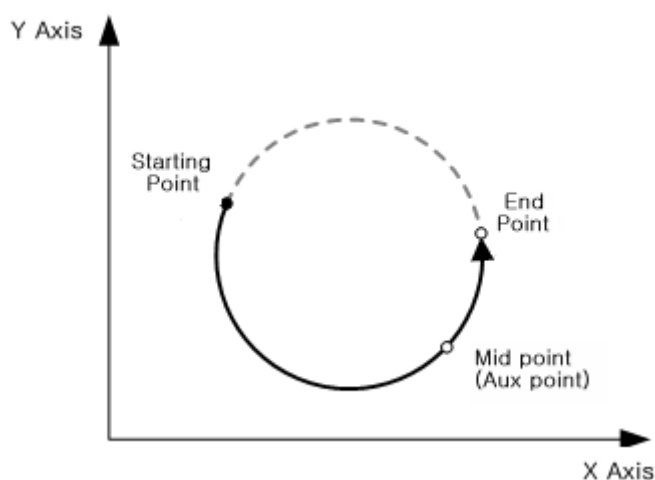
运动功能块		
<pre> graph LR subgraph MC_MoveCircularAbsolute Execute[Execute] AxesGroup[AxesGroup] CircMode[CircMode] AuxPoint[AuxPoint] EndPoint[EndPoint] PathChoice[PathChoice] Velocity[Velocity] Acceleration[Acceleration] Deceleration[Deceleration] Jerk[Jerk] BufferMode[BufferMode] TransitionMode[TransitionMode] TransitionParameter[TransitionParameter] Done[Done] Busy[Busy] Active[Active] CommandAborted[CommandAborted] Error[Error] ErrorID[ErrorID] end Execute --- Done AxesGroup -.- AxesGroup CircMode --- Busy AuxPoint --- Active EndPoint --- CommandAborted PathChoice --- Error Velocity --- ErrorID Acceleration --- ErrorID Deceleration --- ErrorID Jerk --- ErrorID BufferMode --- ErrorID TransitionMode --- ErrorID TransitionParameter --- ErrorID </pre>		
输入-输出		
UINT	AxesGroup	设置绝对位置圆环插补运行的组。(1 ~ 16: 组 1~组 16)
输入		
BOOL	Execute	在上升沿给相关组绝对位置圆环插补运行命令。
UINT	CircMode	圆弧插补方法设置。 [0: 中间点, 1: 中心点, 2: 半径]
LREAL[]	AuxPoint	指定位置的辅助点由绝对坐标的圆弧插补方式决定。
LREAL[]	EndPoint	在绝对坐标指定圆弧的结束点。
BOOL	PathChoice	圆弧路线选择 0: 正转, 1: 反转
LREAL	Velocity	指定路径的最大速度。 [u/s]
LREAL	Acceleration	指定路径的最大加速。 [u/s ²]
LREAL	Deceleration	指定路径的最大减度。 [u/s ²]
LREAL	Jerk	指定加/减速变化率。 [u/s ³]
UINT	BufferMode	指定运动功能块的顺序运行设置。 (参考 6.1.4.缓存模式)
UINT	TransitionMode	未使用
LREAL	TransitionParameter	未使用
输出		
BOOL	Done	表示是否到达指定位置。
BOOL	Busy	表示运动功能块执行未完成。
BOOL	Active	表示当前运动功能块控制相关轴。
BOOL	CommandAborted	表示当前运动功能块在运行时被中断。
BOOL	Error	表示是否发生错误。

WORD	ErrorID	当运动功能块运行时，输出错误发生编号。
------	---------	---------------------

- (1) 该运动功能块在 AxesGroup 输入中给相关轴组绝对位置插补命令。
- (2) 当启动该运动功能块，每个轴参照设置辅助点执行圆弧路线插补控制，移动方向由 PathChoice 输入决定。当 PathChoice 输入设置为 0，圆弧插补运行正转，当设置为 1，圆弧插补运行反转。
- (3) 当轴在 AuxPoint 和 EndPoint 输入作为数组中执行圆弧插补，指定绝对位置的辅助点参考。输入数组和指定轴 ID [ID1, ID2, ID3, ...] 顺序一致。(3 LEAL 类型大小数组 Position 输入，3 轴包括给组圆弧插补命令。)
- (4) 在 Velocity, Acceleration, Deceleration 和 Jerk 输入中分别指定插补路径的速度，加速，减速和加减速的变化率。
- (5) 在 CircMode 输入中设置圆弧插补方式。圆弧插补方式按照 CircMode 的值可分为如下几种。

(a) 中间点指定方式圆弧插补 (BORDER, CircMode = 0)

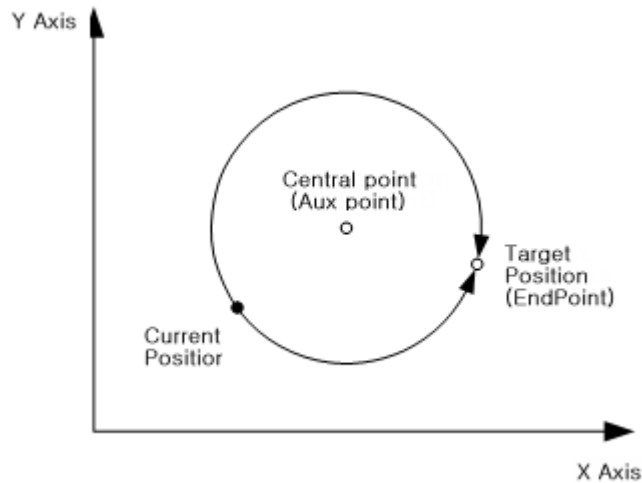
此方式中，在开始点运行，通过中心位置到目标位置的指定位置执行圆弧插补。下图表示轴组在命令开始点坐标符合开始点，在 AuxPoint 输入中输入坐标相当于中心点，在 EndPoint 中输入的坐标相当于目标位置绝对值。



第 6 章运动功能块

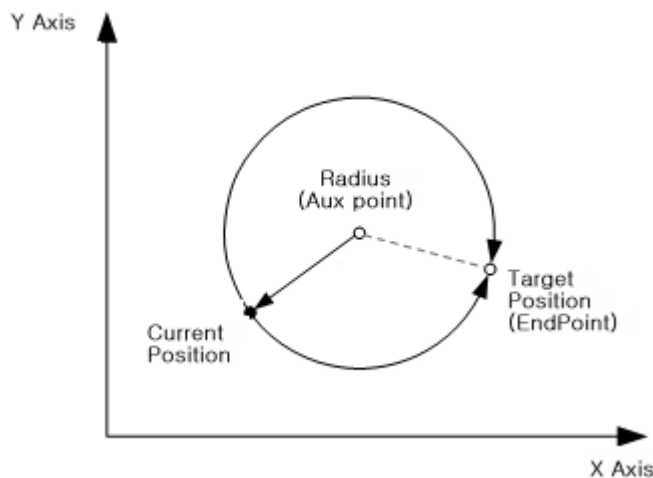
(b) 中心点圆弧插补指定方式

此方式中，在当前位置启动，在圆弧路线做圆弧插补到目标位置，指定中心位置的半径距离。下图表示轴组在命令相似的当前位置开始点的坐标，在 **AuxPoint** 输入符合中心点的坐标，在 **EndPoint** 中输入目标位置绝对值。



(c) 使用半径圆弧插补方式

此种方式，在当前位置运行，通过指定半径的圆弧路线的目标位置执行圆弧插补。下图表示轴组坐标在命令开始当前位置的相似位置，在 **AuxPoint** 相似处 X 轴输入半径，在 **EndPoint** 中输入目标位置绝对值。



(6) 更多内容请参考运动控制模块的 8.2.7 线性插补控制部分。

(7) 指令完成前通过再次执行功能块(执行输入为 On)应用变更参数。仅 Velocity, Acceleration, Deceleration, Jerk, AuxPoint, EndPoint 输入可以更新。

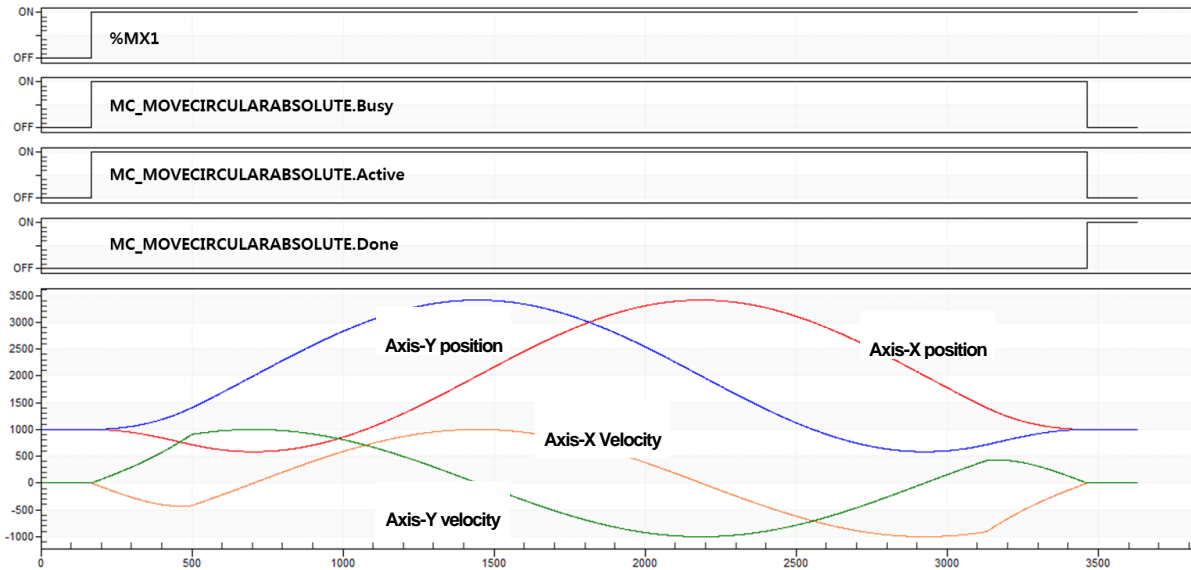
(8) 实例程序

实例显示当当前指令位置为(1000, 1000)时，设置中心点(2000,2000)规格方法后，以顺时针移动圆弧插值到目标位置 (1000, 1000)。

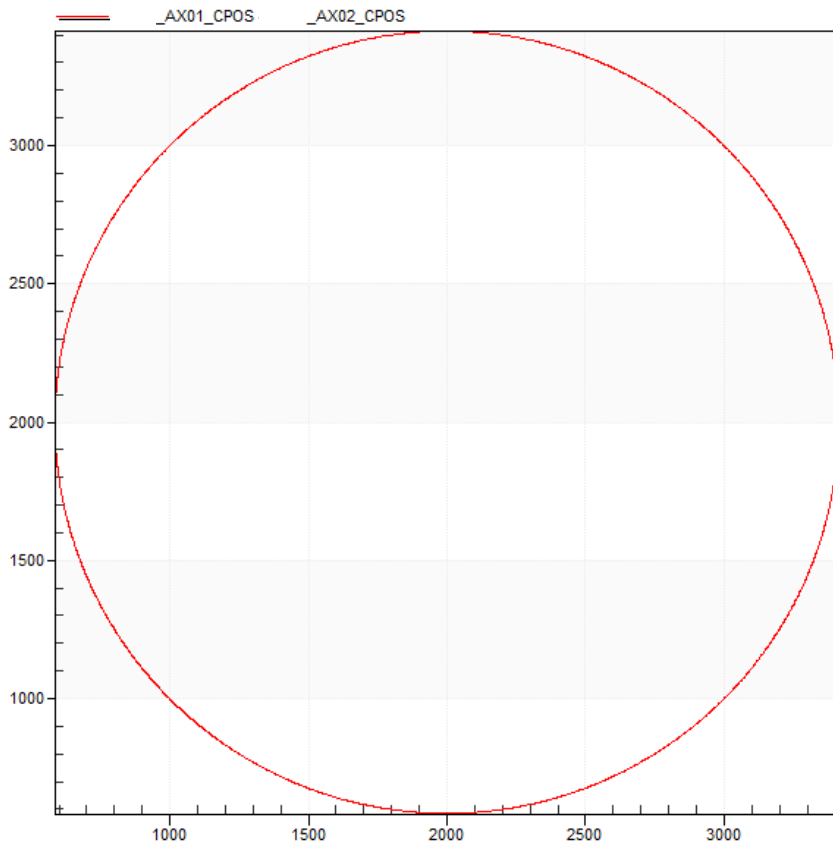
(a) 功能块设置

1	<GLOBAL>	%JL1.3	1.000000000000000e+003	LREAL	_AX01_CPOS
2	<GLOBAL>	%JL2.3	1.000000000000000e+003	LREAL	_AX02_CPOS
3	<GLOBAL>	%JL1.4	0.000000000000000e+000	LREAL	_AX01_CVEL
4	<GLOBAL>	%JL2.4	0.000000000000000e+000	LREAL	_AX02_CVEL
5	Group	AuxPoint		ARRAY[0..2] OF LREAL	
6	Group	AuxPoint[0]	2.000000000000000e+003	LREAL	
7	Group	AuxPoint[1]	2.000000000000000e+003	LREAL	
8	Group	AuxPoint[2]	0.000000000000000e+000	LREAL	
9	Group	Endpoint		ARRAY[0..2] OF LREAL	
10	Group	Endpoint[0]	1.000000000000000e+003	LREAL	
11	Group	Endpoint[1]	1.000000000000000e+003	LREAL	
12	Group	Endpoint[2]	0.000000000000000e+000	LREAL	

(b) 时序图



(c) XY 图



6.5.14 圆弧插补相对位控运行(MC_MoveCircularRelative)

运动功能块		
<pre> graph LR subgraph MC_MoveCircularRelative Execute[Execute] AxesGroup[AxesGroup] CircMode[CircMode] AuxPoint[AuxPoint] EndPoint[EndPoint] PathChoice[PathChoice] Velocity[Velocity] Acceleration[Acceleration] Deceleration[Deceleration] Jerk[Jerk] BufferMode[BufferMode] TransitionMode[TransitionMode] TransitionParameter[TransitionParameter] Done[Done] Busy[Busy] Active[Active] CommandAborted[CommandAborted] Error[Error] ErrorID[ErrorID] end Execute --- Done AxesGroup -.- AxesGroup CircMode --- Busy AuxPoint --- Active EndPoint --- CommandAborted PathChoice --- Error Velocity --- ErrorID Acceleration --- ErrorID Deceleration --- ErrorID Jerk --- ErrorID BufferMode --- ErrorID TransitionMode --- ErrorID TransitionParameter --- ErrorID </pre>		
输入-输出		
UINT	AxesGroup	设置相对位置圆环插补运行的组。(1 ~ 16: 组 1~组 16)
输入		
BOOL	Execute	在上升沿给相关组相对位置圆环插补运行命令。
UINT	CircMode	圆弧插补方法设置。[0: 中间点, 1: 中心点, 2: 半径]
LREAL[]	AuxPoint	指定位置的辅助点由相对坐标的圆弧插补方式决定。
LREAL[]	EndPoint	指定基于开始点的相对坐标圆弧结束点。
BOOL	PathChoice	圆弧路线选择 0:正转, 1: 反转
LREAL	Velocity	指定路径的最大速度。[u/s]
LREAL	Acceleration	指定路径的最大加速。[u/s ²]
LREAL	Deceleration	指定路径的最大减度。[u/s ²]
LREAL	Jerk	指定加/减速变化率。[u/s ³]
UINT	BufferMode	指定运动功能块的顺序运行设置。 (参考 6.1.4.缓存模式)
UINT	TransitionMode	未使用
LREAL	TransitionParameter	未使用
输出		
BOOL	Done	表示是否到达指定位置。
BOOL	Busy	表示运动功能块执行未完成。
BOOL	Active	表示当前运动功能块控制相关轴。
BOOL	CommandAborted	表示当前运动功能块在运行时被中断。

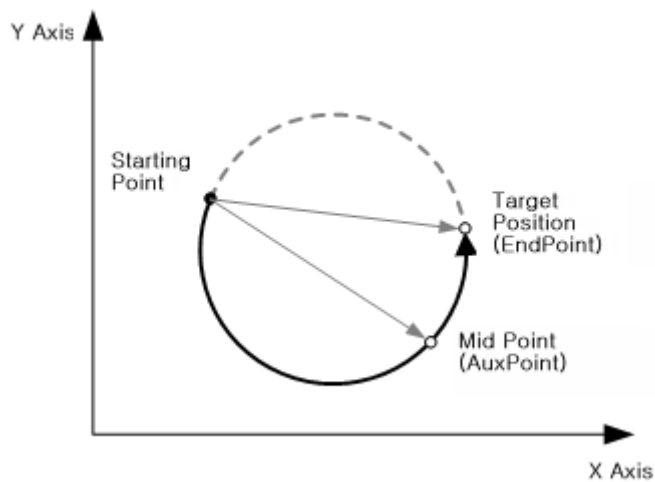
第 6 章运动功能块

BOOL	Error	表示是否发生错误.
WORD	ErrorID	当运动功能块运行时, 输出错误发生编号.

- (1) 该运动功能块在 **AxesGroup** 输入中给相关轴组相对位置插补命令。
- (2) 当启动该运动功能块, 每个轴参照设置辅助点执行圆弧路线插补控制, 移动方向由 **PathChoice** 输入决定。当 **PathChoice** 输入设置为 0, 圆弧插补运行正转, 当设置为 1, 圆弧插补运行反转。
- (3) 当轴在 **AuxPoint** 和 **EndPoint** 输入作为数组中执行圆弧插补, 指定相对位置的辅助点参考。输入数组和指定轴 ID [**ID1, ID2, ID3, ...**] 顺序一致。(3 LEAL 类型大小数组 **Position** 输入, 3 轴包括给组圆弧插补命令。)
- (4) 在 **Velocity, Acceleration, Deceleration** 和 **Jerk** 输入中分别指定插补路径的速度, 加速, 减速和加减速的变化率。
- (5) 在 **CircMode** 输入中设置圆弧插补方式。圆弧插补方式按照 **CircMode** 的值可分为如下几种。

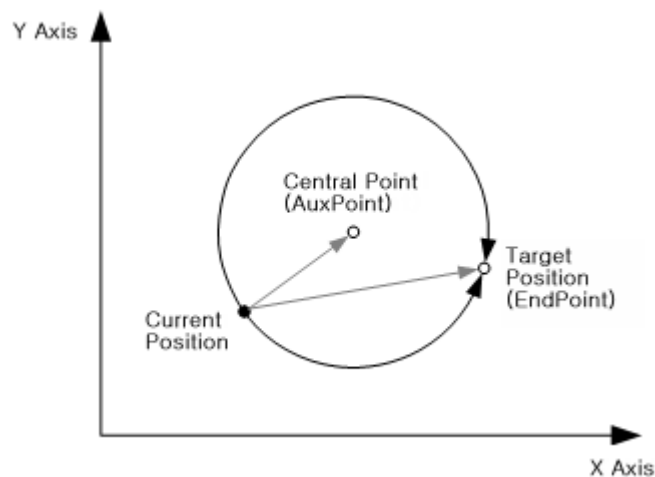
(a) 中间点指定方式圆弧插补 (**BORDER, CircMode = 0**)

此方式中, 在开始点运行, 通过中心位置到目标位置的指定位置执行圆弧插补。下图表示轴组在命令开始点坐标符合开始点, 在 **AuxPoint** 输入中输入坐标相当于中心点, 在 **EndPoint** 中输入的坐标相当于目标位置绝对值。



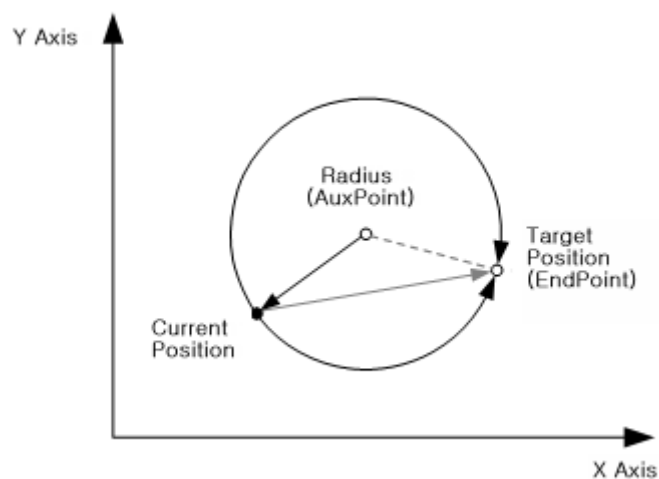
(b) 使用半径圆弧插补方式

此种方式，在当前位置运行，通过指定半径的圆弧路线的目标位置执行圆弧插补。下图表示轴组坐标在命令开始当前位置的相似位置，在 **AuxPoint** 相似处 X 轴输入半径，在 **EndPoint** 中输入目标位置绝对值。



(c) 使用半径指定方式圆弧插补

此种方式，在当前位置运行，通过指定半径的圆弧路线的目标位置执行圆弧插补。下图表示轴组坐标在命令开始当前位置的相似位置，在 **AuxPoint** 相似处 X 轴输入半径，在 **EndPoint** 中输入目标位置绝对值。



(6) 更多内容请参考运动控制模块的线性插补控制部分。

(7) 实例程序

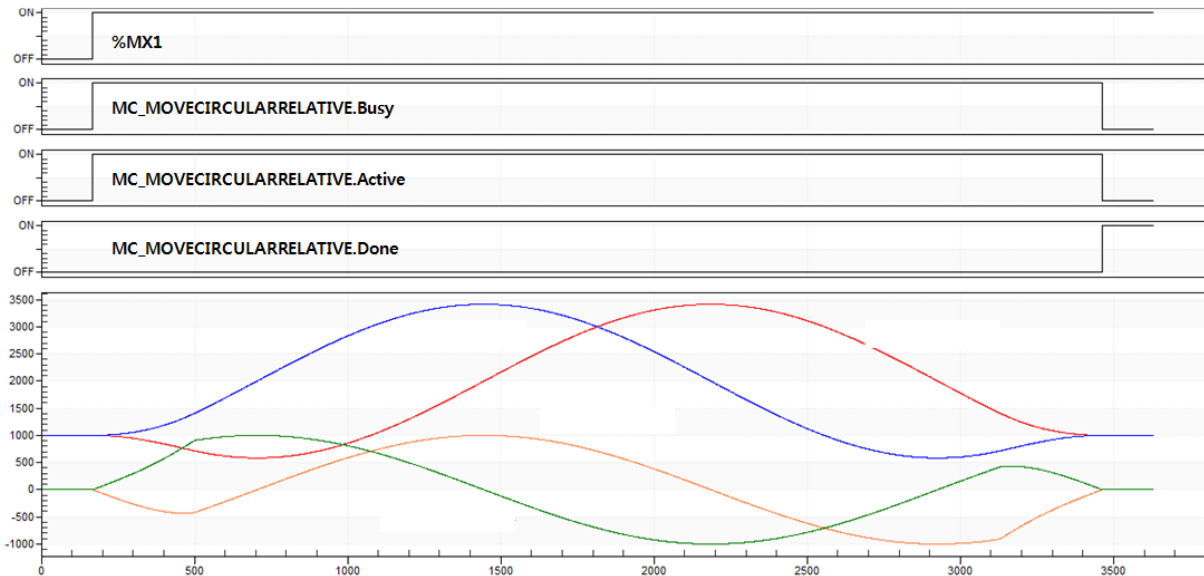
实例设置当前指令位置为(1000, 1000)设置中心点规格方法(设置从当前位置到中心点相对位置: 1000, 1000), 并顺时针执行圆弧插值到目标位置 (设置从当前位置到目标位置的相对位置 : 0, 0).

第 6 章运动功能块

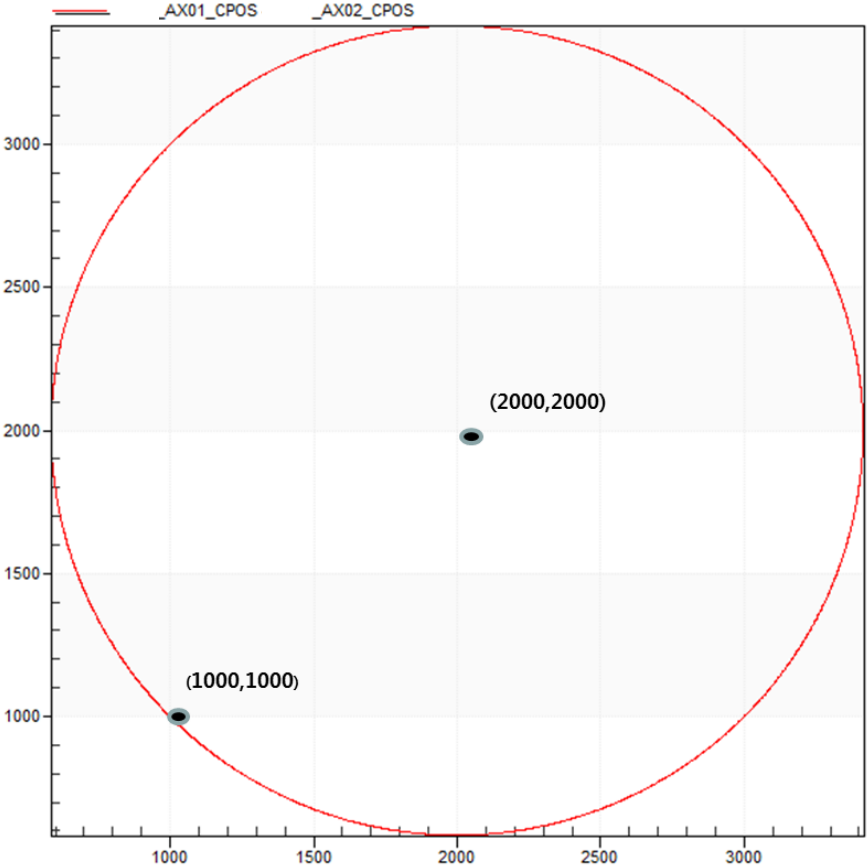
(a) 功能块设置

1	<GLOBAL>	%JL1.3	1.0000000000000000e+003	LREAL	_AX01_CPOS
2	<GLOBAL>	%JL2.3	1.0000000000000000e+003	LREAL	_AX02_CPOS
3	<GLOBAL>	%JL1.4	0.0000000000000000e+000	LREAL	_AX01_CVEL
4	<GLOBAL>	%JL2.4	0.0000000000000000e+000	LREAL	_AX02_CVEL
5	Group	AuxPoint52		ARRAY[0..2] OF LREAL	
6	Group	AuxPoint52 [0]	1.0000000000000000e+003	LREAL	
7	Group	AuxPoint52 [1]	1.0000000000000000e+003	LREAL	
8	Group	AuxPoint52 [2]	0.0000000000000000e+000	LREAL	
9	Group	Endpoint52		ARRAY[0..2] OF LREAL	
10	Group	Endpoint52 [0]	0.0000000000000000e+000	LREAL	
11	Group	Endpoint52 [1]	0.0000000000000000e+000	LREAL	
12	Group	Endpoint52 [2]	0.0000000000000000e+000	LREAL	

(b) 时序图

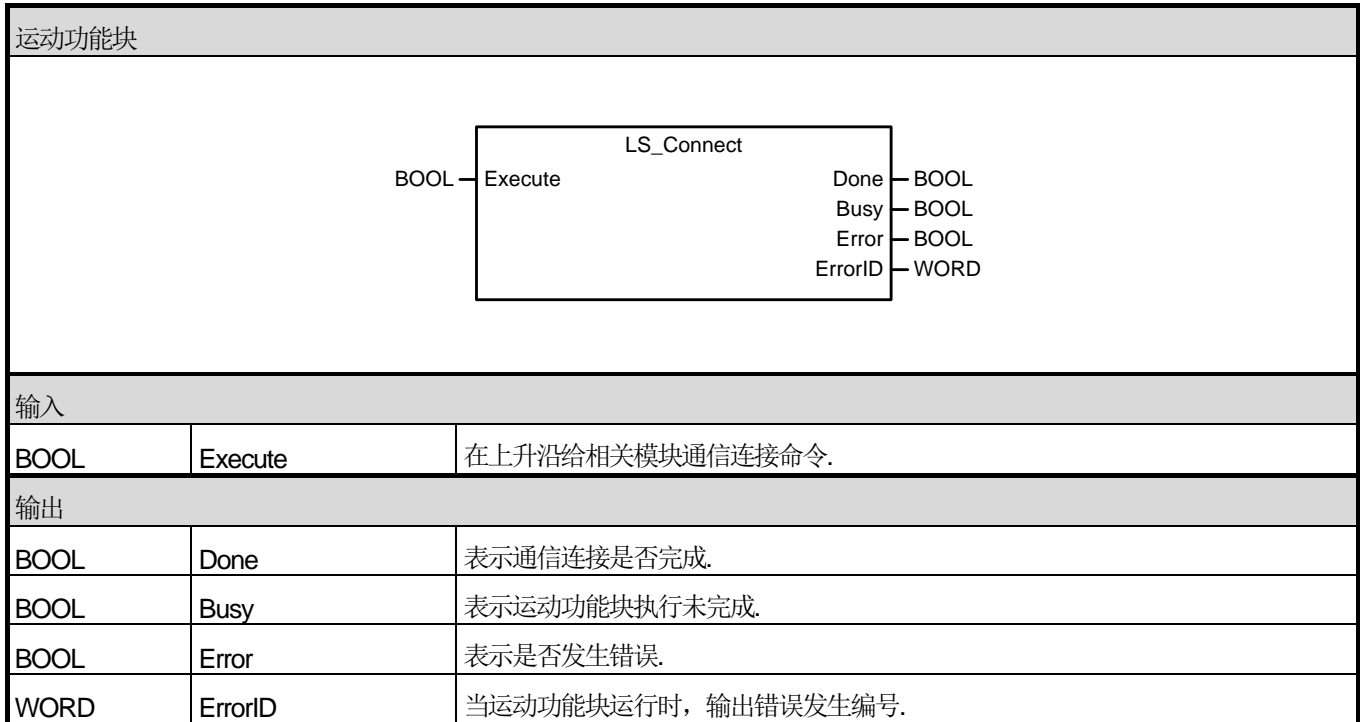


(c) XY 图



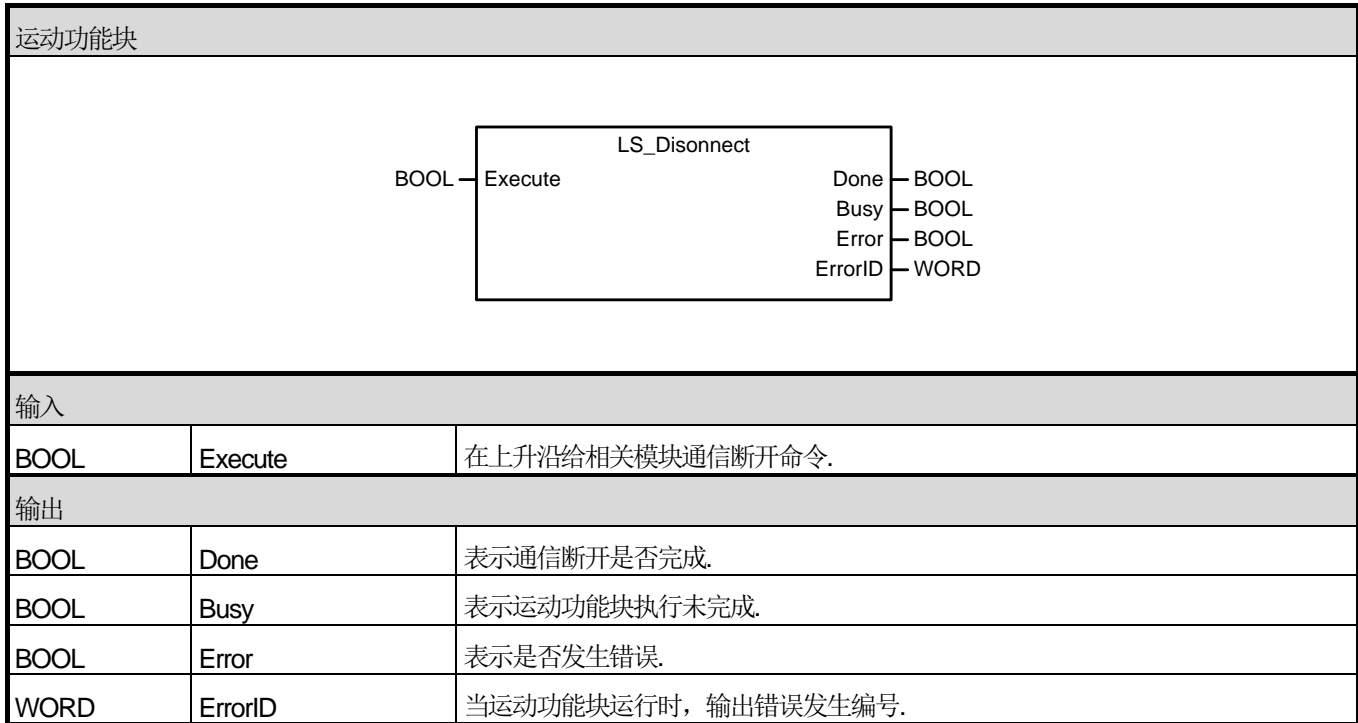
6.6 专用功能块

6.6.1 连接伺服驱动 (LS_Connect)



- (1) 该运动功能块给伺服驱动器或外部输入/输出装置通信连接命令。
- (2) 当从设备正常连接时, Done 为 On, Busy 为 Off。
- (3) 如果在通信连接时发生错误, Error 为 On, 错误编号根据错误原因在 ErrorID 输出。

6.6.2 断开伺服驱动 (LS_Disconnect)



- (1) 该运动功能块给伺服驱动器或外部输入/输出装置通信断开命令。
- (2) 如果通信从站断开， Done 为 On ， Busy 为 off。
- (3) 如果执行通信断开过程中出现错误， Error 为 On ， 根据错误情况， 错误编号在 ErrorID 输出。

6.6.3 读取 SDO(LS_ReadSDO)



- (1) 该运动功能块在相关轴读取伺服驱动器的 SDO(CoE Object)值, 在 Axis 输入读取指定 Index 和 SubIndex 的 SDO 值长度大小
- (2) 当运动功能块运行时, Value 输出为除去 0, 当运行完成时, 输出为读取值. (Done 输出为 On)
- (3) Index 输入设置如下. 当设置其他值时, 出现 “error 0x1F12”。

变量	描述
16#0000 ~ 16#0FFF	数据类型描述
16#1000 ~ 16#1FFF	通信对象
16#2000 ~ 16#5FFF	生产商指定 Profile 区域
16#6000 ~ 16#9FFF	标准设备 Profile 区域

- (4) 在 0~255 之间的值可以输入 SubIndex, 如果值超出设置范围, 出现 “error 0x1F12”。
- (5) 1~4 之间的值可以输入 Length, 表示 1~4 Byte. 如果超出设置范围, 出现 “error 0x1F12”。

6.6.4 写入 SDO (LS_SDO)

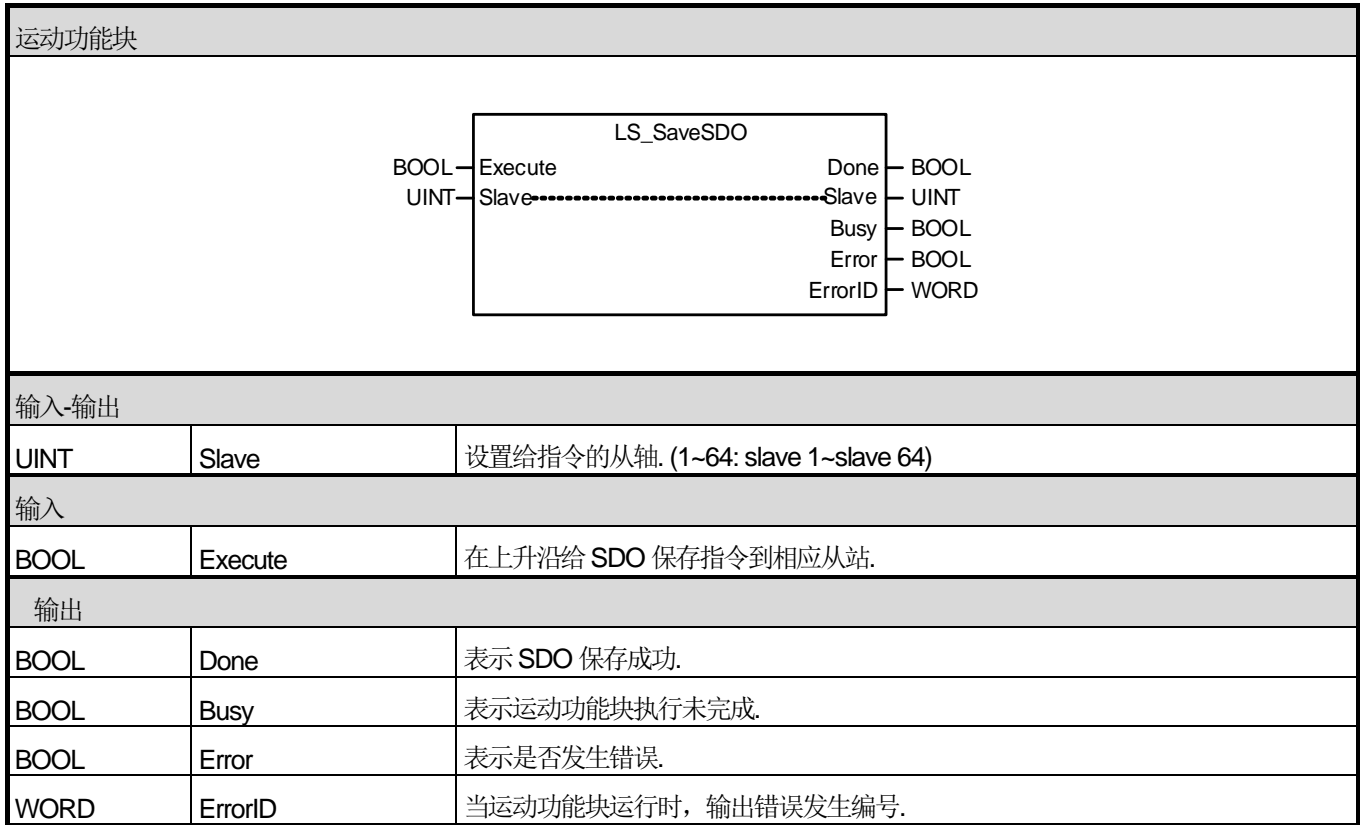


- (1) 该运动功能块将 SDO(CoE Object)值写入相关轴的伺服驱动器, 在指定位置如 Index 和 SubIndex 中写入 SDO 的大小和长度等参数.
- (2) Index 输入设置如下。当设置其他值时, 出现 “error 0x1F12”.

值	描述
16#0000 ~ 16#0FFF	数据类型描述
16#1000 ~ 16#1FFF	通信对象
16#2000 ~ 16#5FFF	生产商指定 Profile 区域
16#6000 ~ 16#9FFF	标准设备 Profile 区域

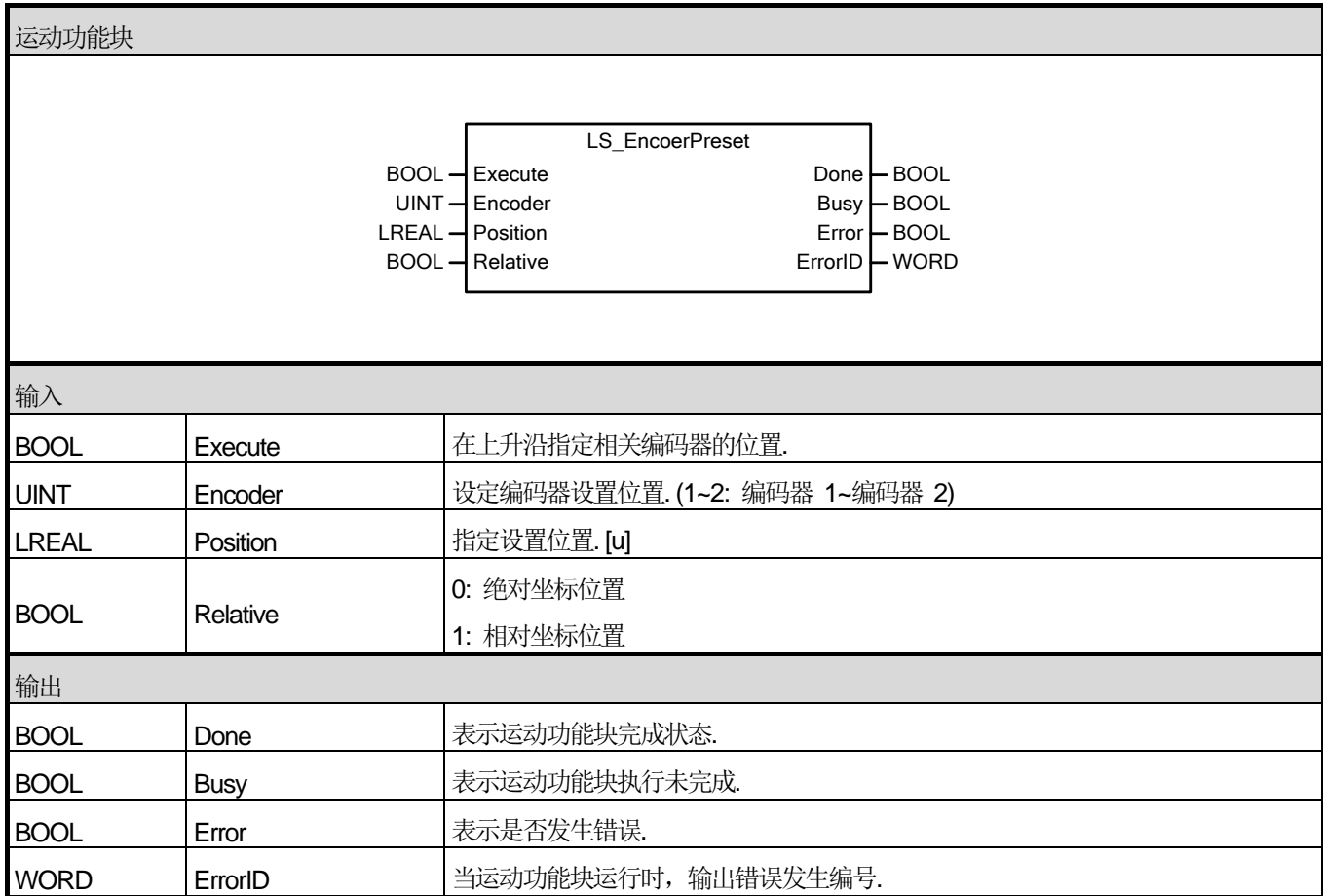
- (3) 在 0~255 之间的值可以输入 SubIndex, 如果值超出设置范围, 出现 “error 0x1F12”.
- (4) 1~4 之间的值可以输入 Length, 表示 1~4 Byte。如果超出设置范围, 出现 “error 0x1F12”.

6.6.5 保存 SDO (LS_SaveSDO)



(1) 运动功能块是一个保存指定从站 SDO 到从站内存的指令.

6.6.6 编码器预设 (LS_EncoerPreset)



- (1) 该运动功能块设置相关编码器的当前位置.
- (2) 在 Position 输入中指定位置。当执行运动功能命令时，如果 Relative 输入为 Off，当前轴的位置被替换为 Position 输入值，如果 Relative 输入为 On，Position 输入值被加到相关轴的当前位置.

6.6.7 JOG 运行(LS_Jog)

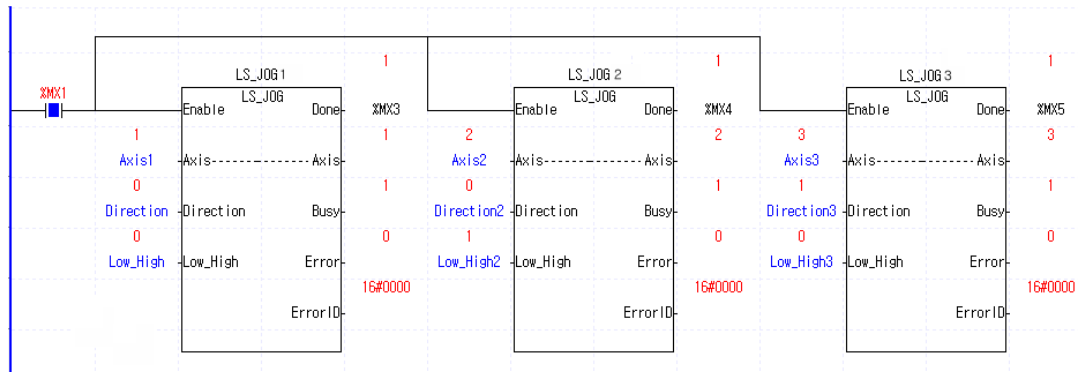
运动功能块		
输入-输出		
UINT	Axis	给轴命令设置. (1~32: 实/虚轴, 33~36: 虚轴)
输入		
BOOL	Enable	当输入为 On 时, 给相关轴 jog 命令。
BOOL	Direction	Jog 时设置旋转方向。(0: 正向, 1: 反向)
BOOL	Low/High	Jog 时设置 jog 速度。 (0: Jog 低速运行, 1: Jog 高速运行)
输出		
BOOL	Enabled	表示相关轴 jog 状态。
BOOL	Busy	表示运动功能块执行未完成。
BOOL	Error	表示是否发生错误。
WORD	ErrorID	当运动功能块运行时, 输出错误发生编号。

- (1) 该运动功能块使相关轴执行 Jog 运行。
- (2) Jog 是用于测试的手动运行功能, 用于确认系统运行的位置地址, 接线状况检查, teaching 等。Jog 可分为高速和低速。
- (3) 当 Enable 输入为 On (jog 中), 如果低速/高速设置值改变, jog 运行中, 无需停止, 速度即改变, 如果 JOG_DIR 设置值改变, Jog 减速停止后, 改变方向。

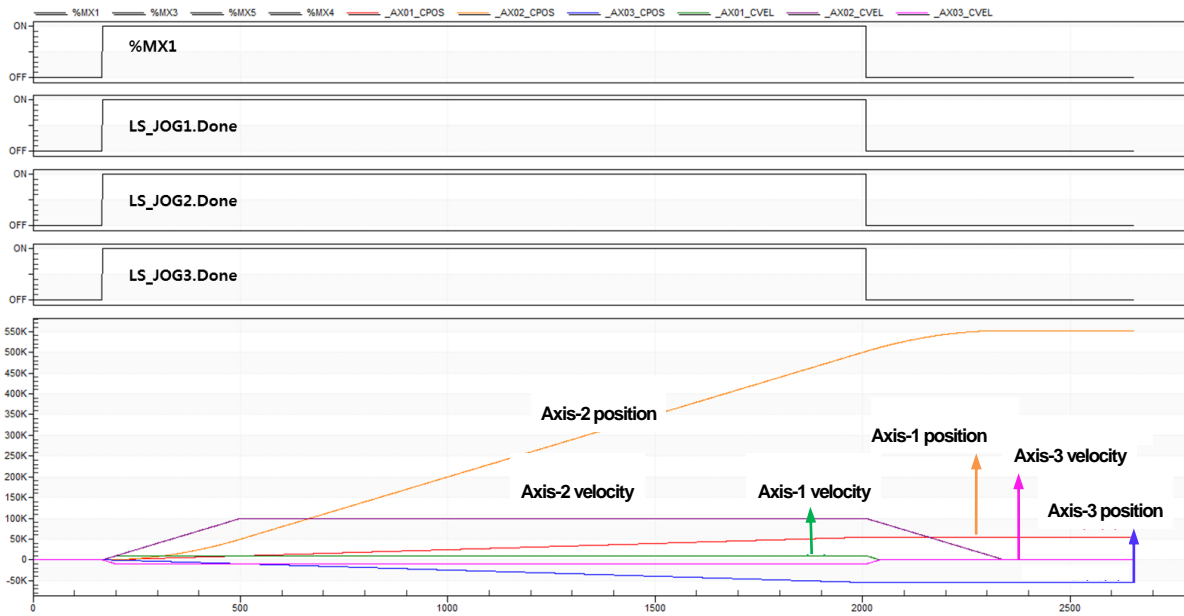
(4) 实例程序

实例显示当前指令位置为 0 时, 按照以下设置的 jog 运行。

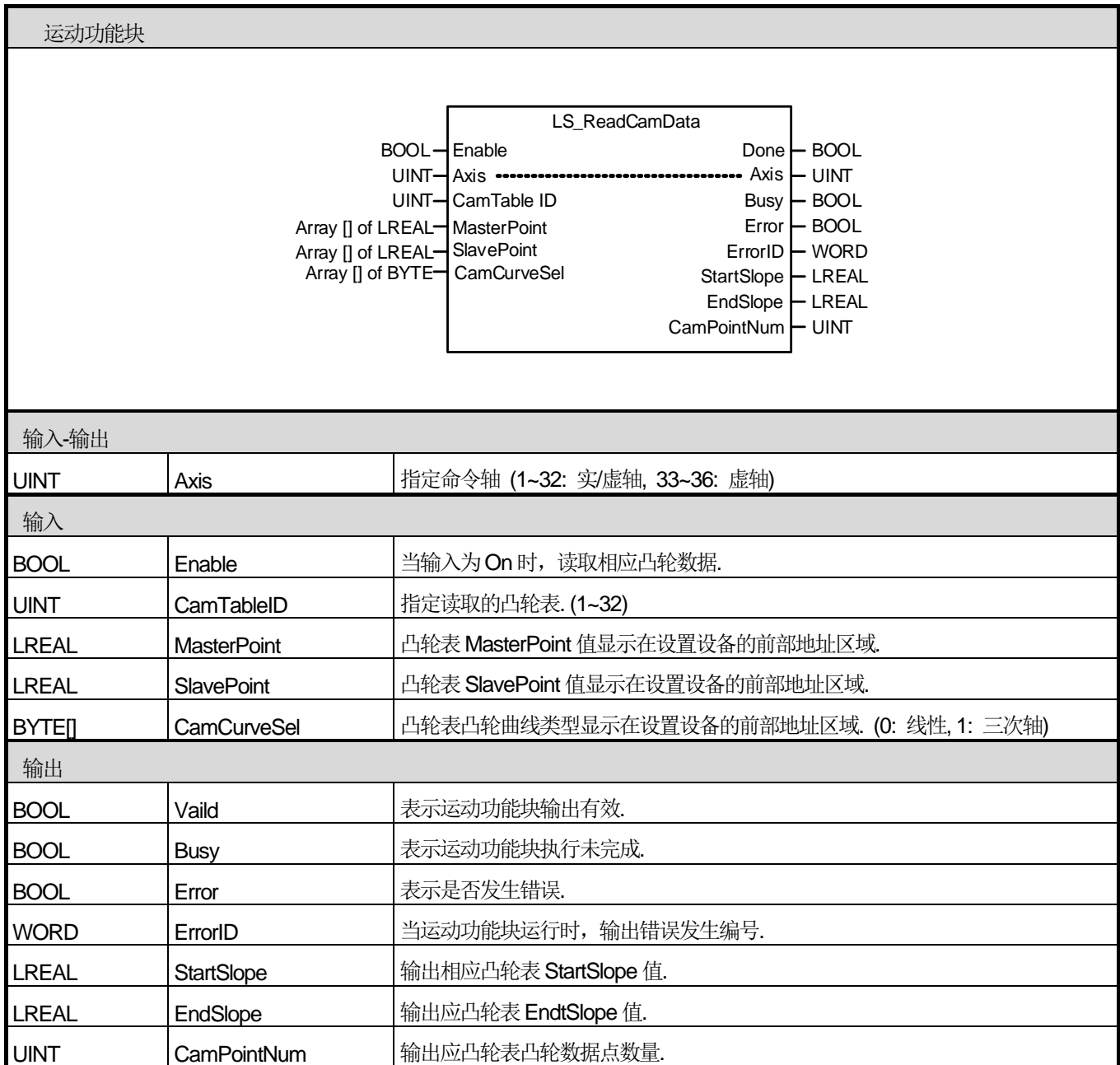
(a) 功能块设置



(b) 时序图



6.6.8 读取凸轮数据(LS_ReadCamData)



- (1) 该功能块显示凸轮表数据.
- (2) 当使能输入激活时, 凸轮表数据值显示连续.
- (3) 变量首地址保存从设置在主站点和从站点的凸轮轮廓中读取的"主轴位置"和"从轴位置". 如果阵列变量尺寸设置小于凸轮表中的数据数量, 由于凸轮数据读取仅通过阵列尺寸,所以可能无法读取整个凸轮表数据.

6.6.9 写入凸轮数据(LS_WriteCamData)

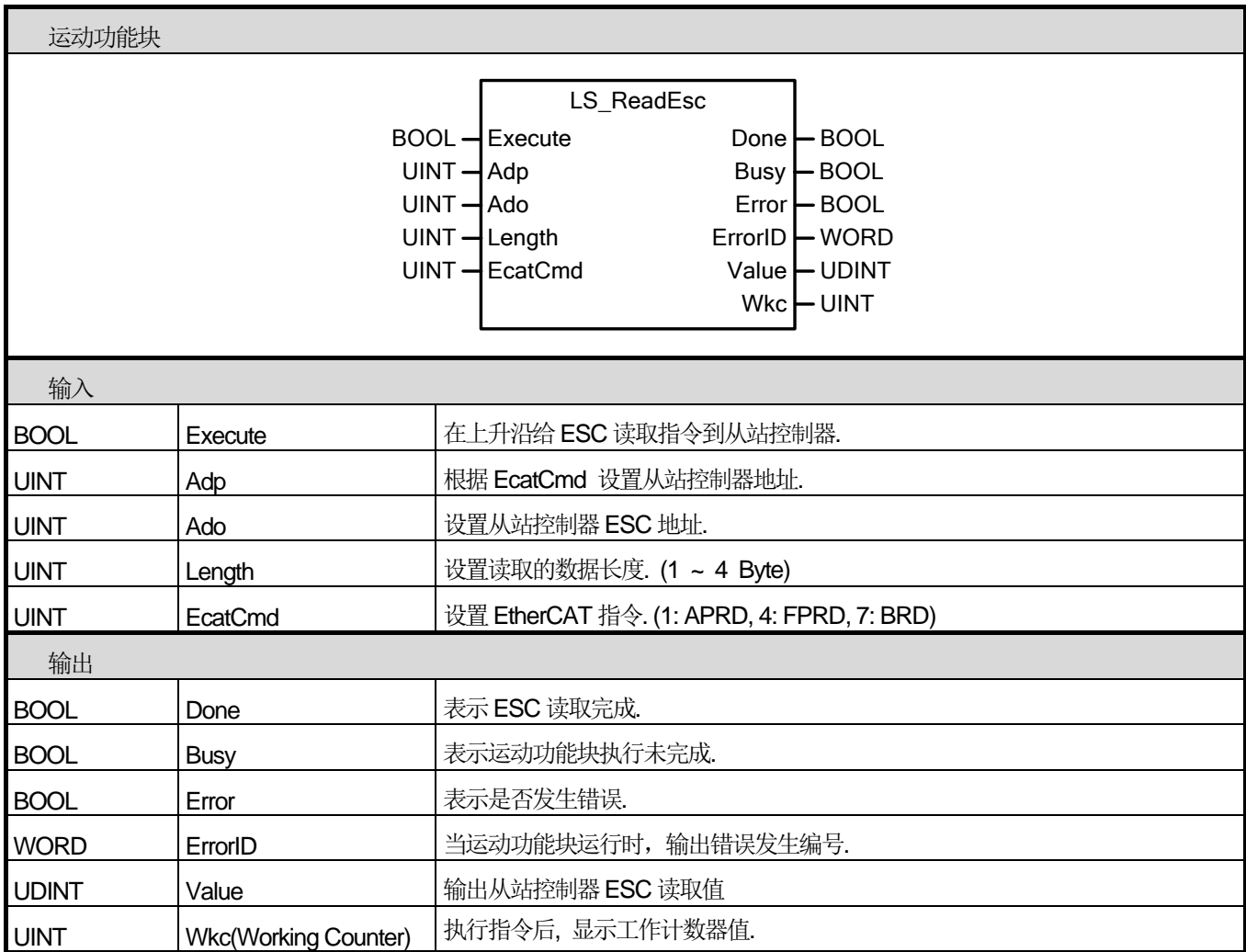
运动功能块		
<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: right;"> <p>BOOL—Execute</p> <p>UINT—Axis</p> <p>UINT—CamTable ID</p> <p>LREAL—StartSlope</p> <p>LREAL—EndSlope</p> <p>UINT—CamPointNum</p> <p>Array[] of LREAL—MasterPoint</p> <p>Array[] of LREAL—SlavePoint</p> <p>Array[] of BYTE—CamCurveSel</p> <p>UINT—ExecutionMode</p> </div> <div style="border: 1px solid black; padding: 5px; text-align: center;"> <p>LS_WriteCamData</p> </div> <div style="text-align: left;"> <p>Done—BOOL</p> <p>Axis—UINT</p> <p>Busy—BOOL</p> <p>Error—BOOL</p> <p>ErrorID—WORD</p> </div> </div>		
输入-输出		
UINT	Axis	指定命令轴 (1~32: 实/虚轴, 33~36: 虚轴)
输入		
BOOL	Execute	在输入上升沿给凸轮数据写入指令.
UINT	CamTableID	指定写入的凸轮表 ID. (1~32)
LREAL	StartSlope	指定写入的凸轮表 StartSlope 值.
LREAL	EndSlope	指定写入的凸轮表 EndSlope 值.
UINT	CamPointNum	指定写入的凸轮表凸轮数据点数量.
LREAL	MasterPoint	对于写入的凸轮数据, 设置保存 Master Point 值的设备主要地址.
LREAL	SlavePoint	对于写入的凸轮数据, 设置保存 Slave Point 值的设备主要地址.
BYTE[]	CamCurveSel	对于写入的凸轮数据, 设置保存凸轮曲线类型的主要地址. (0: 线性, 1: 三次轴)
UINT	ExecutionMode	设置写入凸轮数据的时序. 0 -立即应用, 1: 在相同点通过缓存模式的'缓存'进行应用
输出		
BOOL	Done	表示凸轮数据成功写入.
BOOL	Busy	表示运动功能块执行未完成.
BOOL	Error	表示是否发生错误.
WORD	ErrorID	当运动功能块运行时, 输出错误发生编号.

- (1) 该运动功能块是写入凸轮表数据值的指令. 对于通过 CamTableID 输入设置的凸轮表数据, 使用在 StartSlope 和 EndSlope 中设置的 MasterPoint 和 Slave Point 设备值和 CamPointNum 中作为 MasterPoint and SlavePoint 值的设置数量.
- (2) 凸轮表 ID 输入设置范围为 1 到 32. 设置超出该范围的值将导致 "错误 16#000B"
- (3) 可以设置一个小于当前凸轮轮廓的 CamPointNum 中设置数量的值, 如果 CamPointNum 值大于当前凸轮轮廓,发生错误 16#111C".
- (4) 如果 MasterPoint / SlavePoint / CamCurveSel 阵列尺寸设置值小于 CamPointNum, 发生"错误 16#000B".
- (5) ExecutionMode 输入设置设置时序. 当输入为 0 时, 根据执行指令执行设置. 当输入为 1 时, 在顺序执行"缓存"的同时执行设置. 设置错误值可能导致"错误 16#000B".

0(mcImmediatly) : 根据执行功能块变更(执行输入上升沿)参数值. 如果轴为运行状态, 可能影响动作.

1(mcQueued) : 在缓存模式的"缓存" 执行同时变更.

6.6.10 读取 ESC(LS_ReadEsc)



- (1) 该运动功能块是用于读取在动态数据对象中设置的指定从站设备 ESC (EtherCAT 从站控制器) 的地址数据.
- (2) 当执行运行功能块时值和 Wkc(工作计数器)显示为 0. 当执行完成时(Done 输出为 on),显示读取数据值, 在 Wkc 中显示工作计数器值.
- (3) Adp(地址位置)指定 EtherCAT 从站设备地址. 以下值可以根据 EcatCmd 设置值进行设置. 如果 EcatCmd 设置为 7(BRD), 忽略 Adp 输入值. 如果值超出 Adp 输入设置范围, 出现“Error 0x0F60”.

EcatCmd	Adp 范围
1 (APRD)	0x0000: 第一个从站连接 0xFFFF: 第二个从站连接 0xFFFE: 第三个从站连接 : 0xFFC1: 第 64 个从站连接
4 (FPRD)	1 ~ 64: 从站 1~从站 64
7 (BRD)	-

- (4) 长度可设置范围 1~4, 表示 1-4 字节. 设置超出该范围值将发生"Error 0x0F61. "
- (5) 在 EcatCmd 中, 当读取 ESC (EtherCAT 从站控制器)时设置使用的指令类型. 可以使用以下指令中的一种: 设置值超出以上 EcatCmd 中的范围将发生"Error 0x0F62."

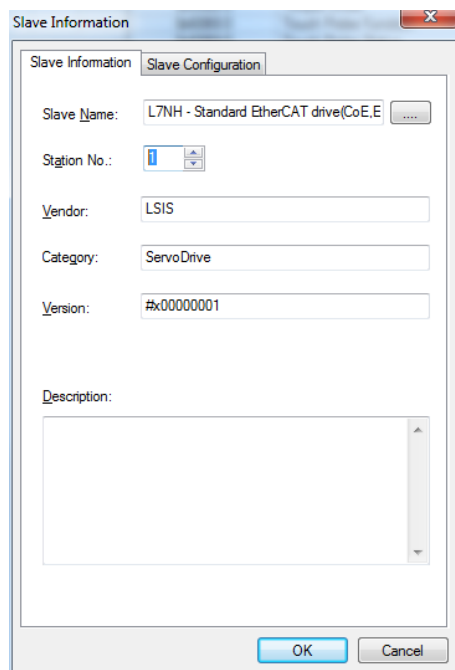
1) 1 - APRD (自动增加物理读取)

该指令用于主轴正常通讯连接之前当按照物理连接顺序读取从站设备数据时. 从站设备接受 0 值 Adp, 将通过读取指定长度尺寸的数据. 当接收 EtherCAT 帧时增加每个从站设备的 Adp. 例如,如果 EcatCmd 为 1, Adp 设置为 0xFFFF, 当执行 ESC 读取功能块时, 由于从第一个从站设备接收 EtherCAT frame 的 Adp 不是 1, 所以无法执行读取动作, 仅 Adp 增加 1. 当第二个从站设备接收 EtherCAT 帧时, 由于第一个从站的 Adp 值由 0 增加到 1,所以执行读取动作.根据从站设备连接顺序的 Adp 设置值如下.

从站控制器	设置值
第 1 个连接从站	0
第 2 个连接从站	0xFFFF
:	:
第 64 个连接从站	0xFFC1

2) 4 - FPRD (配置地址物理读取)

用于主站正常通讯连接后通过指定从站设备站号地址读取数据. 如果通过 EtherCAT 主站设置的从站设备站号地址与传输的 Adp 匹配, 从站设备读取由 Ado 区域中指定长度尺寸的数据. 当增加从站时, 主站设置的从站设备站号地址可通过从站信息对话框进行查看.

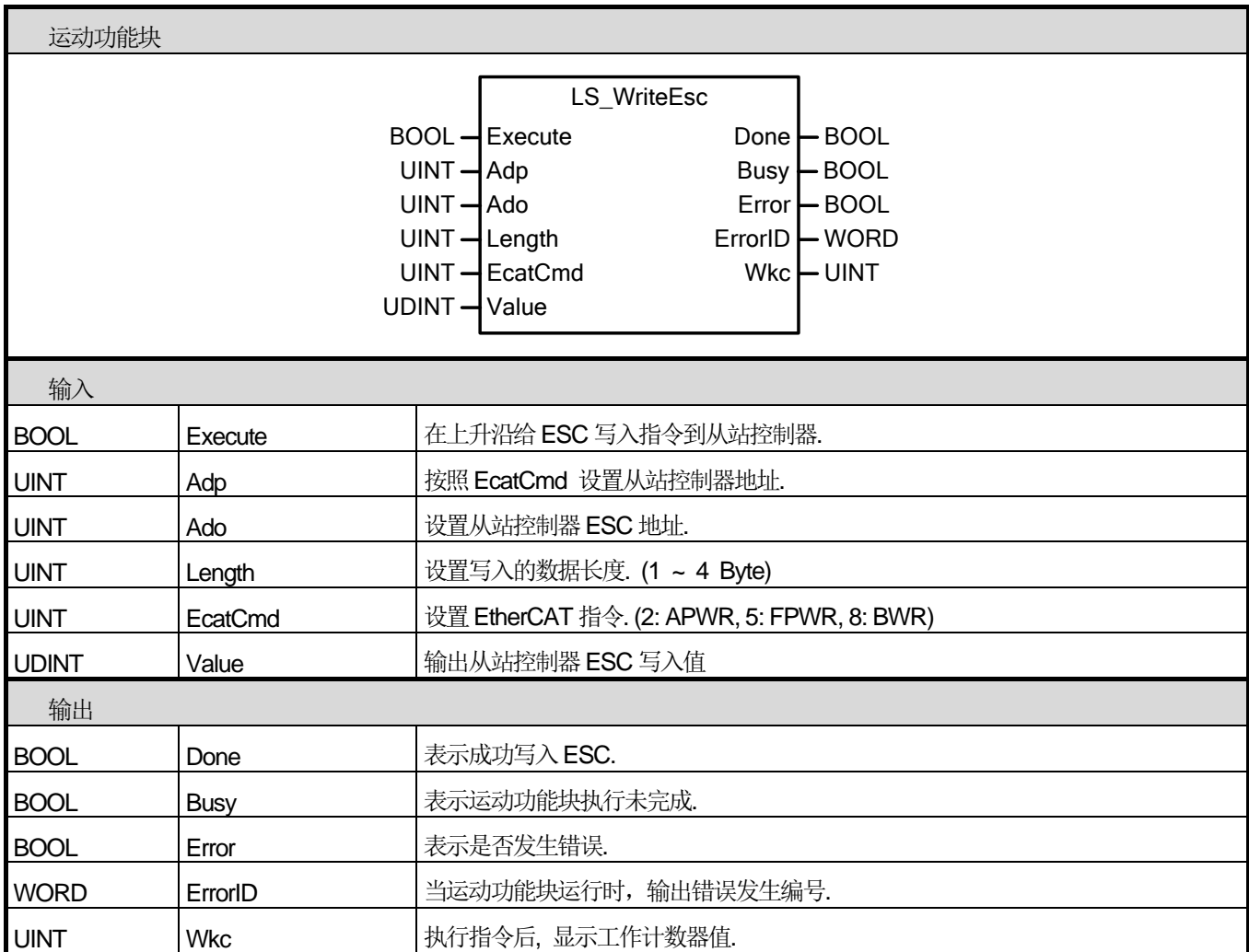


3) 7 - BRD (广播读取)

所有连接从站设备尺寸读取数据通过 Ado 区域的长度设置, Bitwise-OR(每个位的 OR 运行) 之后保存结果. 忽略 Adp 中指定地址值, Wkc 加 1 导致所有从站执行正常读取运行

- (6) Wkc 表示工作计数器. 如果在指定从站设备成功读取数据, 则增加 1. 如果 EcatCmd 为 7(BRD), 则增加 1 导致所有从站执行正常读取运行.
- (7) 执行 ESC 读取指令后, 如果从指定从站设备中执行正常数据读取运行, Done 输出为 on.

6.6.11 写入 ESC (LS_WriteEsc)



- (1) 该运动功能块使用 Ado 中设置的地址写入数据到由 Adp 设置的从站设备 ESC (EtherCAT 从站控制器).
- (2) 当执行运动功能块时 Wkc 值显示为 0, 当执行完成时, 显示工作计数器值(Done 输出为 on). Wkc 通过由 EcatCmd 和 Adp 指定的从站设备增加 1.
- (3) Adp 输入指定 EtherCAT 从站设备地址. 以下值可根据 EcatCmd 设置值进行设置. 如果 EcatCmd 设置为 8(BWR), 忽略 Adp 输入值.如果值超出设置范围, 发生“错误 0x0F70”.

EcatCmd	Adp 范围
2 (APWR)	0x0000: 第 1 个连接从站 0xFFFF: 第 2 个连接从站 0xFFFE: 第 3 个连接从站 : 0xFFC1: 第 64 个连接从站
5 (FPWR)	1~64: 从站 1~从站 64
8 (BWR)	-

- (4) 长度可设置为 1~ 4, 表示 1~4 字节. 设置值超出范围可能导致"错误 0x0F71".
- (5) 在 EcatCmd 中, 当读取 ESC(EtherCAT 从站控制器)时设置使用指令类型. 使用以下写入指令. 设置值超出范围可能导致"错误 0x0F72".

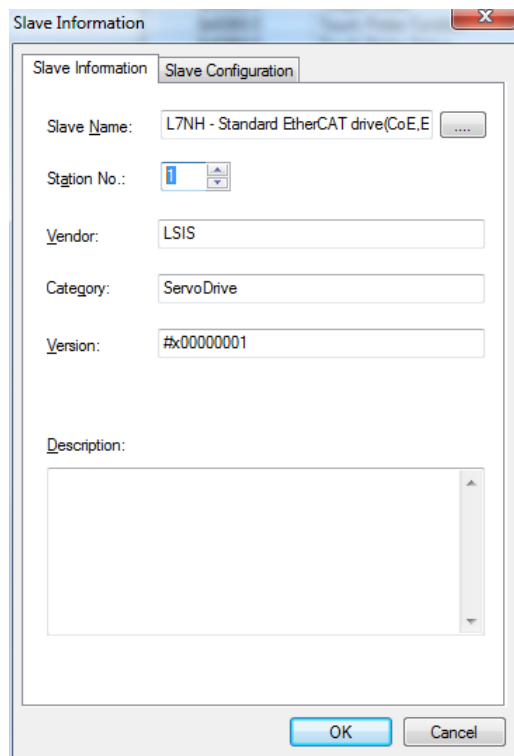
1) 2 - APW (自动增加物理写入)

该指令用于主轴正常通讯连接之前当按照物理连接顺序读取从站设备数据时. 从站设备接受 0 值 Adp, 将通过读取指定长度尺寸的数据. 当接收 EtherCAT 帧时增加每个从站设备的 Adp. 例如, 如果 EcatCmd 为 2, Adp 设置为 0xFFFF, 当执行 ESC 读取功能块时, 由于从第一个从站设备接收 EtherCAT frame 的 Adp 不是 1, 所以无法执行写入动作, 仅 Adp 增加 1. 当第二个从站设备接收 EtherCAT 帧时, 由于第一个从站的 Adp 值由 0 增加到 1, 所以执行写入动作. 根据从站设备连接顺序的 Adp 设置值如下.

从站控制器	设置值
第 1 个连接从站	0
第 2 个连接从站	0xFFFF
:	:
第 64 个连接从站	0xFFC1

2) 5 - FPWR (配置地址物理写入)

用于主站正常通讯连接后通过指定从站设备站号地址写入数据. 如果通过 EtherCAT 主站设置的从站设备站号地址与传输的 Adp 匹配, 从站设备读取由 Ado 区域中指定长度尺寸的数据. 当增加从站时, 主站设置的从站设备站号地址可通过从站信息对话框进行查看.

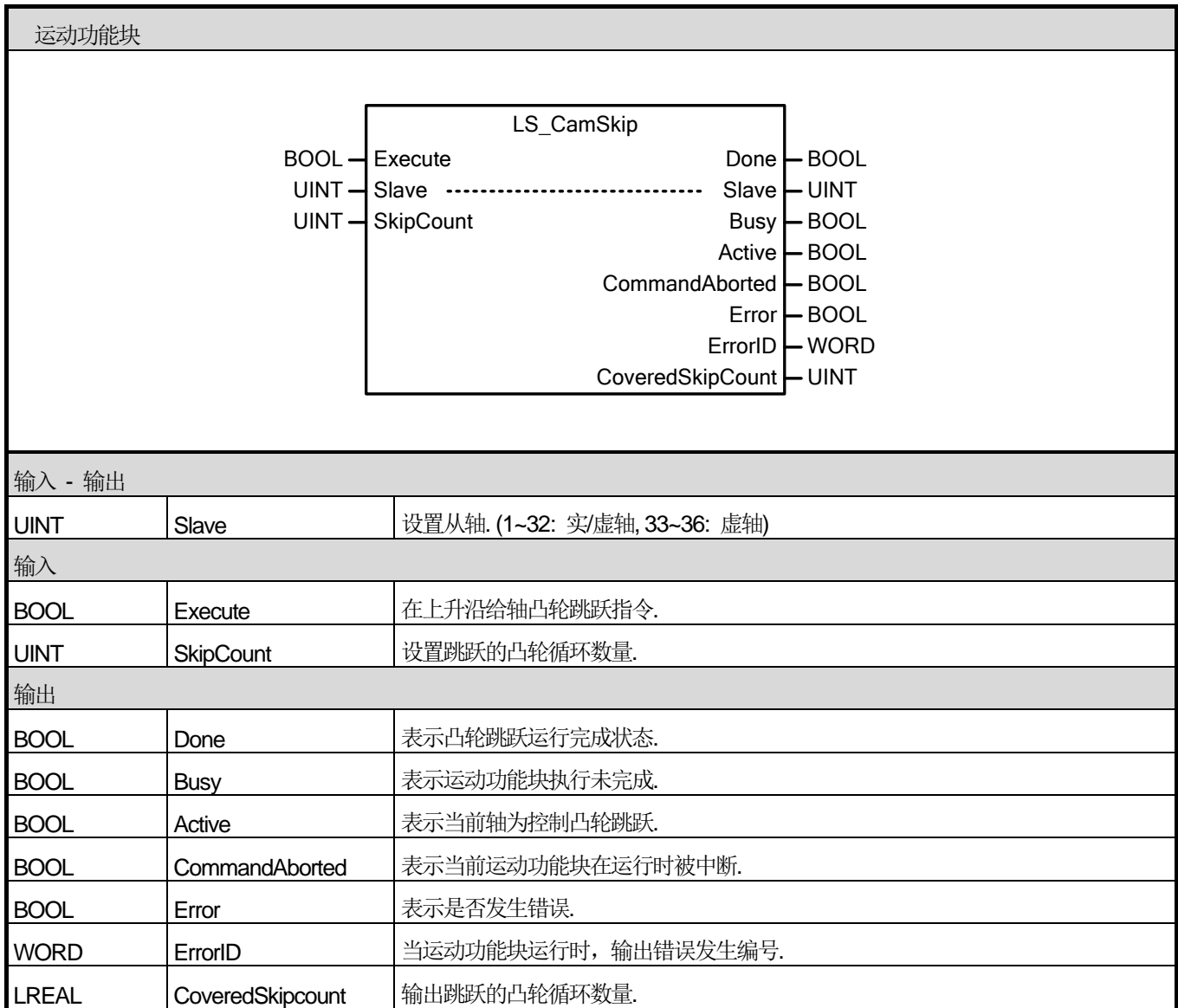


3) 8-BWR, 广播写入

所有连接从站设备尺寸写入数据通过 **Ado** 区域的长度设置, **Bitwise-OR**(每个位的 OR 运行) 之后保存结果. 忽略 **Adp** 中指定地址值, **Wkc** 加 1 导致所有从站执行正常写入运行

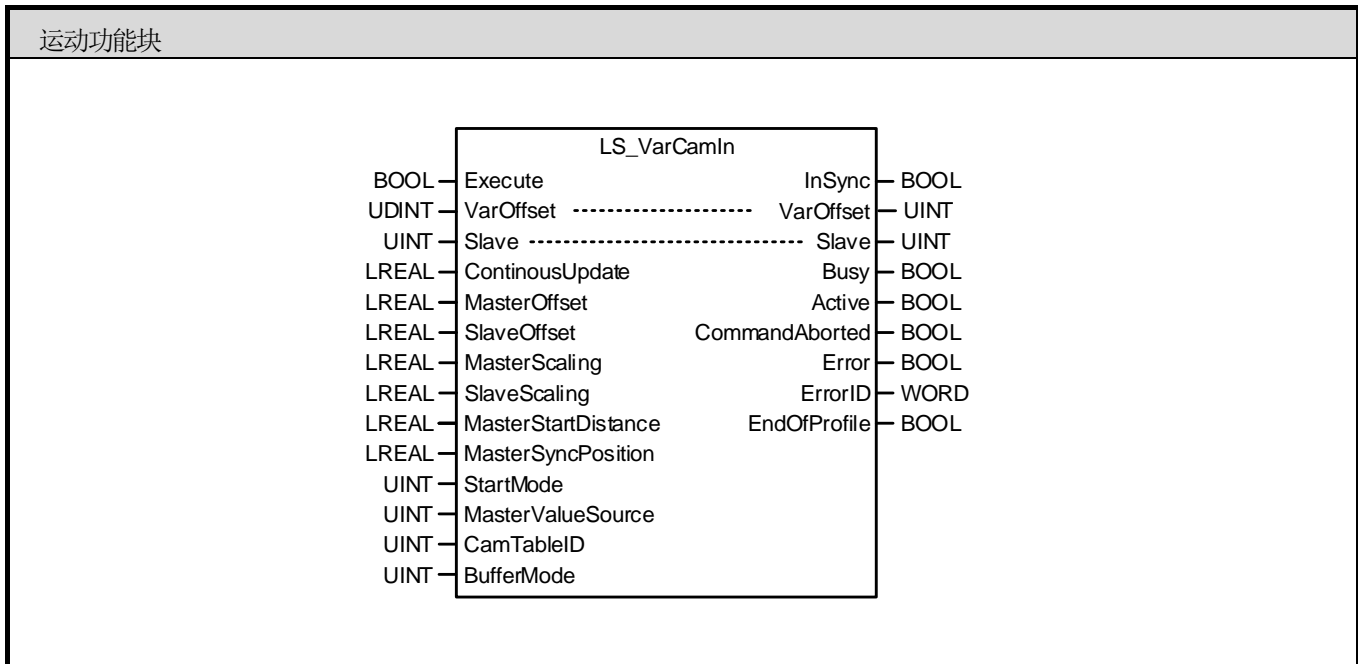
- (6) **Wkc** 表示工作计数器. 如果在指定从站设备成功写入数据, 则增加 1. 如果 **EcatCmd** 为 8(**BWR**), 则增加 1 导致所有从站执行正常写入运行.
- (7) 执行 **ESC** 读取指令后, 如果从指定从站设备中执行正常数据写入运行, **Done** 输出为 on.

6.6.12 跳跃凸轮 (LS_CamSkip)



- (1) 该运动功能块为凸轮跳跃指令, 该指令作为运行中当前凸轮指定跳跃凸轮运行循环.
- (2) SkipCount 确定跳跃的凸轮循环数量. 如果输入为 0, 显示 SkipCount 错误(错误 0x111E).
- (3) 当凸轮运行期间在从轴使用凸轮跳跃指令时, 当当前凸轮循环完成时跳跃动作启动. 凸轮跳跃期间, 从轴在凸轮表结束处于等待状态.
- (4) CoveredSkipCount显示已跳跃的凸轮循环数量. 以每个跳跃循环增加计数, 功能块动作完成后当Done 输出为off时该值变为0
- (5) 当执行凸轮跳跃指令后完成设置循环数的跳跃, Done 输出为 on.

6.6.13Var 凸轮运行 (LS_VarCamIn)



输入 - 输出

UDINT	VarOffset	设置变量用于主轴的 M 设备补偿值.
UINT	Slave	设置从轴. (1~32: 实/虚轴, 33~36: 虚轴)

输入

BOOL	Execute	在上升沿给轴凸轮运行指令.
BOOL	ContinuousUpdate	指定输入值的更新设置. (参考 6.1.5.运动功能块在执行中改变参数)
LREAL	MasterOffset	设置主轴补偿值.
LREAL	SlaveOffset	设置从轴凸轮表补偿值.
LREAL	MasterScaling	指定主轴放大倍率.
LREAL	SlaveScaling	指定从轴凸轮表放大倍率.
LREAL	MasterStartDistance	指定从站凸轮运行的主轴位置.
LREAL	MasterSyncPosition	当凸轮运行启动时在凸轮表中指定起始点.
UINT	StartMode	设置凸轮运行模式. 0: 凸轮表应用为绝对值(mcAbsolute) 1: 凸轮表应用为基于指令起始点的相对值(mcRelative)

UINT	MasterValueSource	设置用于凸轮运行主轴的源。 0：同步主轴目标值。 1：同步从轴当前值。
UINT	CamTableID	指定运行凸轮表。
UINT	BufferMode	指定运动功能块的顺序运行设置。 (参考 6.1.4.缓存模式)
输出		
BOOL	InSync	表示凸轮运行正常完成。 (表示从站符合凸轮表)
BOOL	Busy	表示运动功能块执行未完成。
BOOL	Active	表示当前运动功能块控制相关轴。
BOOL	CommandAborted	表示当前运动功能块在运行时被中断。
BOOL	Error	表示是否发生错误。
WORD	ErrorID	当运动功能块运行时，输出错误发生编号。

- (1) 该运动功能块是通过设置补偿指定变量值为主轴，运行跟随主轴的从轴凸轮的功能块。
- (2) 指定为主轴的变量值应为 LREL 类型.例). 当通过%ML100 指定变量分配到内存作为主轴值时，%ML100 应为 LREAL 类型，补偿值指定变量为 UDINT 类型，应该输入 100 到 VarOffset.
- (3) 剩余设置和功能与 MC_CamIn 功能块相同。

6.6.14 变量齿轮运行(LS_VarGearIn)

运动功能块																																														
<div style="border: 1px solid black; padding: 10px; margin: 0 auto; width: fit-content;"> <p style="text-align: center;">LS_VarGearIn</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">BOOL</td> <td style="width: 40%;">Execute</td> <td style="width: 30%;">InGear</td> <td>BOOL</td> </tr> <tr> <td>UDINT</td> <td>VarOffset</td> <td>VarOffset</td> <td>UINT</td> </tr> <tr> <td>UINT</td> <td>Slave</td> <td>Slave</td> <td>UINT</td> </tr> <tr> <td>BOOL</td> <td>ContinuousUpdate</td> <td>Busy</td> <td>BOOL</td> </tr> <tr> <td>INT</td> <td>RatioNumerator</td> <td>Active</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>RatioDenominator</td> <td>CommandAborted</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>MasterValueSource</td> <td>Error</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>Acceleration</td> <td>ErrorID</td> <td>WORD</td> </tr> <tr> <td>LREAL</td> <td>Deceleration</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Jerk</td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>BufferMode</td> <td></td> <td></td> </tr> </table> </div>			BOOL	Execute	InGear	BOOL	UDINT	VarOffset	VarOffset	UINT	UINT	Slave	Slave	UINT	BOOL	ContinuousUpdate	Busy	BOOL	INT	RatioNumerator	Active	BOOL	UINT	RatioDenominator	CommandAborted	BOOL	UINT	MasterValueSource	Error	BOOL	LREAL	Acceleration	ErrorID	WORD	LREAL	Deceleration			LREAL	Jerk			UINT	BufferMode		
BOOL	Execute	InGear	BOOL																																											
UDINT	VarOffset	VarOffset	UINT																																											
UINT	Slave	Slave	UINT																																											
BOOL	ContinuousUpdate	Busy	BOOL																																											
INT	RatioNumerator	Active	BOOL																																											
UINT	RatioDenominator	CommandAborted	BOOL																																											
UINT	MasterValueSource	Error	BOOL																																											
LREAL	Acceleration	ErrorID	WORD																																											
LREAL	Deceleration																																													
LREAL	Jerk																																													
UINT	BufferMode																																													
输入 - 输出																																														
UDINT	VarOffset	设置变量用于主轴的 M 设备补偿值.																																												
UINT	Slave	设置从轴. (1~32: 实/虚轴, 33~36: 虚轴)																																												
输入																																														
BOOL	Execute	在上升沿给齿轮运行指令到相应轴.																																												
BOOL	ContinuousUpdate	指定输入值的更新设置. (参考 6.1.5.运动功能块在执行中改变参数)																																												
LREAL	RatioNumerator	指定齿轮比的分子. (-32768 ~ 32767)																																												
LREAL	RatioDenominator	指定齿轮比的分母. (0 ~ 65535)																																												
LREAL	MasterValueSource	选择同步的主轴数据. 0: 同步主轴指令位置. 1: 同步主轴当前位置.																																												
LREAL	Acceleration	在齿轮运行同步的起始阶段指定加速. [u/s^2]																																												
LREAL	Deceleration	在齿轮运行同步的起始阶段指定减速. [u/s^2]																																												
LREAL	Jerk	指定加/减速变化率. [u/s^3]																																												
UINT	BufferMode	指定运动功能块的顺序运行设置. (参考 6.1.4.缓存模式)																																												

输出		
BOOL	InGear	表示通过应用齿轮比率执行齿轮运行.
BOOL	Busy	表示运动功能块执行未完成.
BOOL	Active	表示当前运动功能块控制相关轴.
BOOL	CommandAborted	表示当前运动功能块在运行时被中断.
BOOL	Error	表示是否发生错误.
WORD	ErrorID	当运动功能块运行时, 输出错误发生编号.

- (1) 该运动功能块为通过作为主轴的补偿设置变量值, 驱动齿轮运行(速度同步)的主轴和从轴.
- (2) 作为主轴的指定变量值应为 LREL 类型.例). 当通过%ML100 指定变量分配到内存作为主轴值时, %ML100 应为 LREAL 类型, 补偿值指定变量为 UDINT 类型, 应该输入 100 到 VarOffset.
- (3) 剩余设置和功能与 MC_GearIn 功能块相同.

6.6.15 变量定位齿轮运行 (LS_VarGearInPos)

运动功能块																																																										
<div style="border: 1px solid black; padding: 10px; margin: 0 auto; width: fit-content;"> <p style="text-align: center;">LS_VarGearInPos</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">BOOL</td> <td style="width: 40%;">Execute</td> <td style="width: 30%;">InGear</td> <td style="width: 10%;">BOOL</td> </tr> <tr> <td>UDINT</td> <td>VarOffset</td> <td>VarOffset</td> <td>UINT</td> </tr> <tr> <td>UINT</td> <td>Slave</td> <td>Slave</td> <td>UINT</td> </tr> <tr> <td>INT</td> <td>RatioNumerator</td> <td>Busy</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>RatioDenominator</td> <td>Active</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>MasterValueSource</td> <td>CommandAborted</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>MasterSyncPosition</td> <td>Error</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>SlaveSyncPosition</td> <td>ErrorID</td> <td>WORD</td> </tr> <tr> <td>UINT</td> <td>SyncMode</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>MasterStartDistance</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Acceleration</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Deceleration</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Jerk</td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>BufferMode</td> <td></td> <td></td> </tr> </table> </div>			BOOL	Execute	InGear	BOOL	UDINT	VarOffset	VarOffset	UINT	UINT	Slave	Slave	UINT	INT	RatioNumerator	Busy	BOOL	UINT	RatioDenominator	Active	BOOL	UINT	MasterValueSource	CommandAborted	BOOL	LREAL	MasterSyncPosition	Error	BOOL	LREAL	SlaveSyncPosition	ErrorID	WORD	UINT	SyncMode			LREAL	MasterStartDistance			LREAL	Acceleration			LREAL	Deceleration			LREAL	Jerk			UINT	BufferMode		
BOOL	Execute	InGear	BOOL																																																							
UDINT	VarOffset	VarOffset	UINT																																																							
UINT	Slave	Slave	UINT																																																							
INT	RatioNumerator	Busy	BOOL																																																							
UINT	RatioDenominator	Active	BOOL																																																							
UINT	MasterValueSource	CommandAborted	BOOL																																																							
LREAL	MasterSyncPosition	Error	BOOL																																																							
LREAL	SlaveSyncPosition	ErrorID	WORD																																																							
UINT	SyncMode																																																									
LREAL	MasterStartDistance																																																									
LREAL	Acceleration																																																									
LREAL	Deceleration																																																									
LREAL	Jerk																																																									
UINT	BufferMode																																																									
输入 - 输出																																																										
UDINT	Master	设置主轴. (1~32: 实/虚轴, 33~36: 虚轴)																																																								
UINT	Slave	设置从轴. (1~32: 实/虚轴, 33~36: 虚轴)																																																								
输入																																																										
BOOL	Execute	在上升沿给齿轮运行指令到相应轴.																																																								
INT	RatioNumerator	指定齿轮比的分子. (-32768 ~ 32767)																																																								
UINT	RatioDenominator	指定齿轮比的分子. (0 ~ 65535)																																																								
UINT	MasterValueSource	选择同步的主轴数据. 0: 同步主轴指令位置. 1: 同步主轴当前位置.																																																								
LREAL	MasterSyncPosition	指定齿轮运行启动的主轴位置.																																																								
LREAL	SlaveSyncPosition	指定齿轮运行启动的从轴位置.																																																								
UINT	SyncMode	未使用																																																								
LREAL	MasterStartDistance	指定同步启动主轴的距离.																																																								
LREAL	Velocity	在同步起始阶段指定主轴最大速度. [u/s]																																																								
LREAL	Acceleration	在同步起始阶段指定主轴最大加速. [u/s ²]																																																								
LREAL	Deceleration	在同步起始阶段指定主轴最大减速. [u/s ²]																																																								

LREAL	Jerk	指定加/减速变化率. [u/s ³]
UINT	BufferMode	指定运动功能块的顺序运行设置. (参考 6.1.4.缓存模式)
输出		
BOOL	InSync	表示作为指定齿轮比率应用的齿轮运行正常完成.
BOOL	StartSync	表示同步启动.
BOOL	Busy	表示运动功能块执行未完成.
BOOL	Active	表示当前运动功能块控制相关轴.
BOOL	CommandAborted	表示当前运动功能块在运行时被中断.
BOOL	Error	表示是否发生错误.
WORD	ErrorID	当运动功能块运行时, 输出错误发生编号.

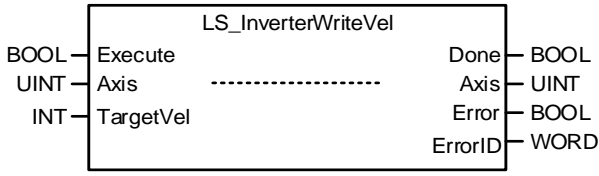
- (1) 该运动功能块为通过作为主轴的补偿设置变量值, 根据指定位置设置的齿轮比率同步主轴和伺服轴.
- (2) 作为主轴的指定变量值应为 LREL 类型.例).当通过%ML100 指定变量分配到内存作为主轴值时, %ML100 应为 LREAL 类型, 补偿值指定变量为 UDINT 类型, 应该输入 100 到 VarOffset.
- (3) 剩余设置和功能与 MC_GearInPos 功能块相同.

6.6.16 读取凸轮表从站位置(LS_ReadCamTableSlavePos)

运动功能块																																										
<div style="border: 1px solid black; padding: 10px; width: fit-content; margin: 0 auto;"> <p style="text-align: center;">LS_ReadCamTableSlavePos</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%; border-right: 1px solid black; padding: 2px;">BOOL</td> <td style="padding: 2px;">Execute</td> <td style="width: 30%;"></td> <td style="padding: 2px;">Done</td> <td style="width: 30%; border-left: 1px solid black; padding: 2px;">BOOL</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">UINT</td> <td style="padding: 2px;">Axis</td> <td style="text-align: center; padding: 2px;">-----</td> <td style="padding: 2px;">Axis</td> <td style="border-left: 1px solid black; padding: 2px;">UINT</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">UINT</td> <td style="padding: 2px;">CamTableID</td> <td></td> <td style="padding: 2px;">Busy</td> <td style="border-left: 1px solid black; padding: 2px;">BOOL</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">LREAL</td> <td style="padding: 2px;">MasterPos</td> <td></td> <td style="padding: 2px;">SlavePos</td> <td style="border-left: 1px solid black; padding: 2px;">LREAL</td> </tr> <tr> <td></td> <td></td> <td></td> <td style="padding: 2px;">SlaveVel</td> <td style="border-left: 1px solid black; padding: 2px;">LREAL</td> </tr> <tr> <td></td> <td></td> <td></td> <td style="padding: 2px;">SlaveAccel</td> <td style="border-left: 1px solid black; padding: 2px;">LREAL</td> </tr> <tr> <td></td> <td></td> <td></td> <td style="padding: 2px;">Error</td> <td style="border-left: 1px solid black; padding: 2px;">BOOL</td> </tr> <tr> <td></td> <td></td> <td></td> <td style="padding: 2px;">ErrorID</td> <td style="border-left: 1px solid black; padding: 2px;">WORD</td> </tr> </table> </div>			BOOL	Execute		Done	BOOL	UINT	Axis	-----	Axis	UINT	UINT	CamTableID		Busy	BOOL	LREAL	MasterPos		SlavePos	LREAL				SlaveVel	LREAL				SlaveAccel	LREAL				Error	BOOL				ErrorID	WORD
BOOL	Execute		Done	BOOL																																						
UINT	Axis	-----	Axis	UINT																																						
UINT	CamTableID		Busy	BOOL																																						
LREAL	MasterPos		SlavePos	LREAL																																						
			SlaveVel	LREAL																																						
			SlaveAccel	LREAL																																						
			Error	BOOL																																						
			ErrorID	WORD																																						
输入 - 输出																																										
UINT	Axis	指定命令轴 (1~32: 实/虚轴, 33~36: 虚轴)																																								
输入																																										
BOOL	Execute	在上升沿给相应轴 ReadCamTableSlavePos 运行指令.																																								
UINT	CAM tableID	指定读取的凸轮表数量(1~32)																																								
LREAL	MasterPos	在凸轮表指定主轴位置.																																								
输出																																										
BOOL	Done	表示 ReadCamTableSlavePos 运行完成.																																								
BOOL	Busy	表示运动功能块执行未完成.																																								
LREAL	SlavePos	输出从站位置.																																								
LREAL	SlaveVel	输出从站速度. [u/s]																																								
LREAL	SlaveAccel	输出从站加速. [u/s ²]																																								
BOOL	Error	表示是否发生错误.																																								
WORD	ErrorID	当运动功能块运行时, 输出错误发生编号.																																								

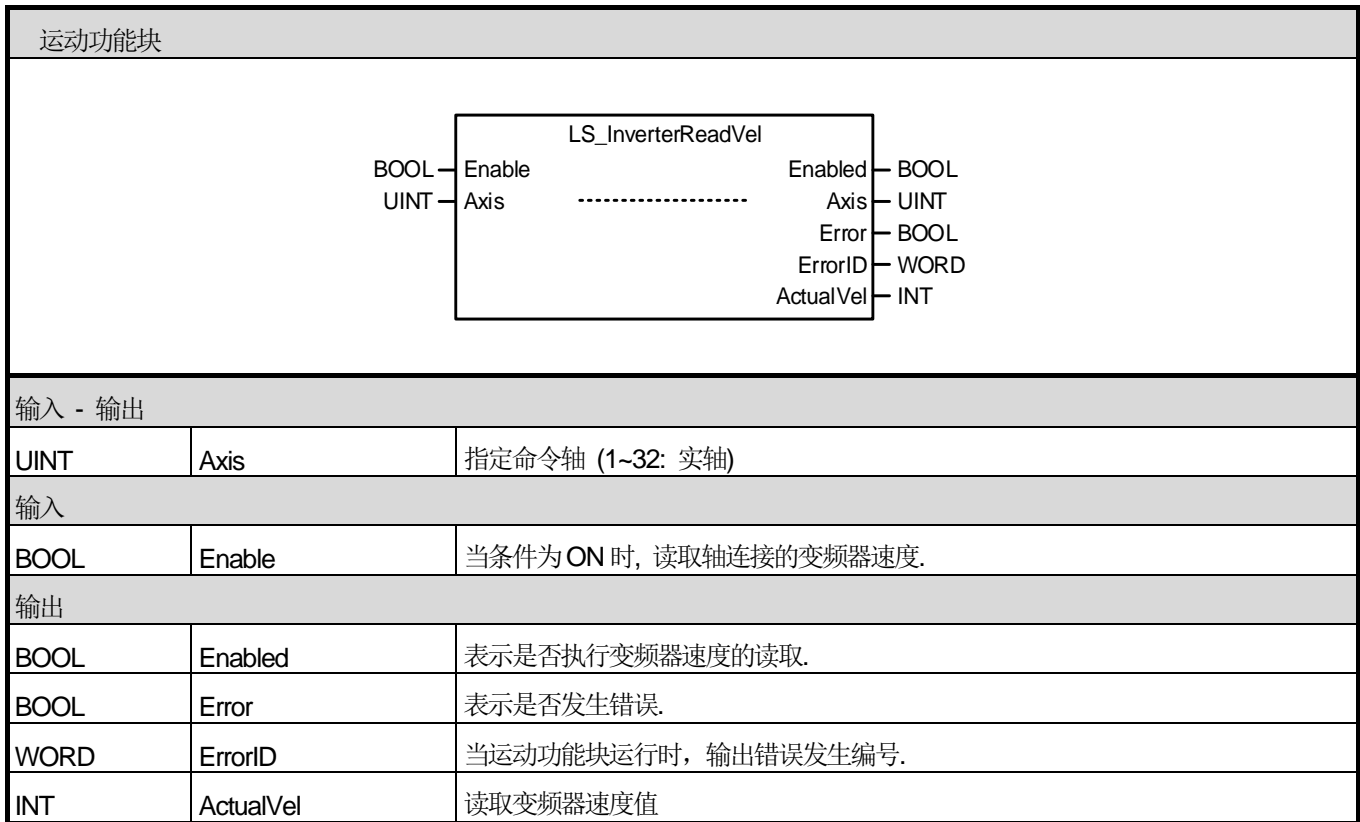
- (1) 该运动功能块根据指定凸轮表中主轴位置输出从轴位置.
- (2) 在凸轮表中设置读取的主轴位置值作为 MasterPos 值.应用到指令轴的补偿/齿轮比/相位校正运行等不会在 SlavePos 输出中反应.
- (3) 当在凸轮表中读取从站位置完成时, 'Done 输出' 变为on.

6.6.17 写入变频器速度 (LS_InverterWriteVel)

运动功能块		
		
输入 - 输出		
UINT	Axis	指定命令轴 (1~32: 实轴)
输入		
BOOL	Execute	在上升沿给 InverterWriteVel 运行指令到相应轴.
INT	TargetVel	设置变频器速度 (-30000 ~ 30000, unit: rpm)
输出		
BOOL	Done	表示 InverterWriteVel 运行完成状态.
BOOL	Error	表示是否发生错误.
WORD	ErrorID	当运动功能块运行时, 输出错误发生编号.

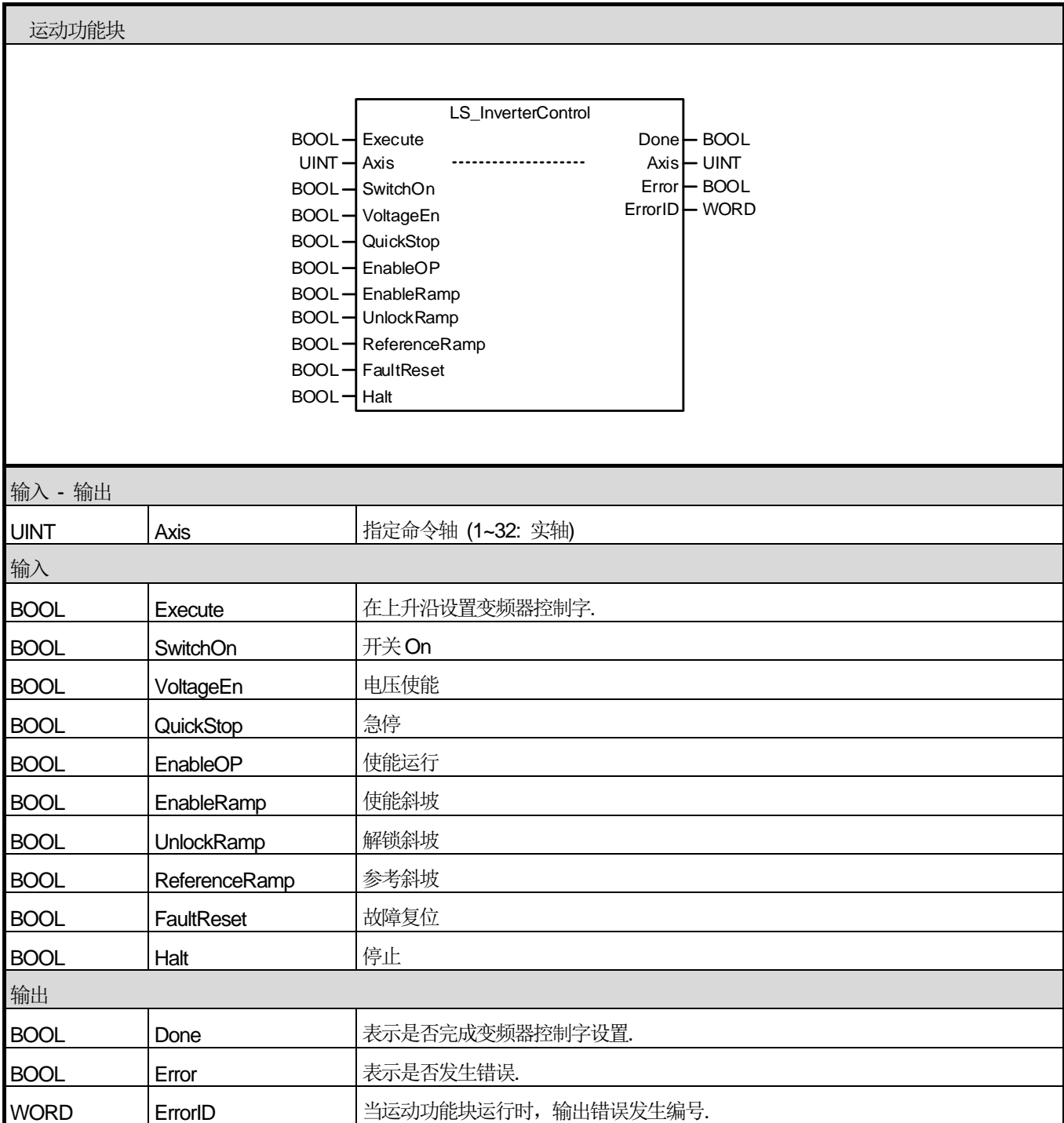
- (1) 当轴控制变频器时, 该运动功能块设置运行的变频器速度
- (2) 如果在 TargetVel 中设置速度, 并执行功能块, 连接到轴的变频器将以相应速度运行.
- (3) 在 TargetVel 中设置的速度值以 rpm 为单位, 可设置范围为-30000 到 30000.

6.6.18 读取变频器速度 (LS_InverterReadVel)



- (1) 当由轴控制变频器时, 该运动功能块读取连接的变频器速度.
- (2) 当执行功能块时, 读取连接轴的变频器当前速度并在 ActualVel 中显示.
- (3) 在 ActualVel 中设置的速度值以 rpm 为单位, 可显示范围为-30000 到 30000.

6.6.19 写入变频器控制字 (LS_InverterControl)



- (1) 当由轴控制变频器时, 该运动功能块设置连接的变频器控制字.
- (2) 为了运行变频器, 控制字必须设置为可以运行.

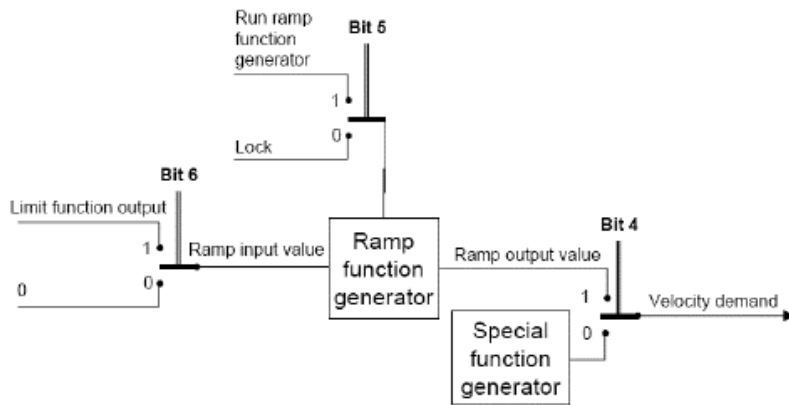
第 6 章运动功能块

(3) 请参考如下。

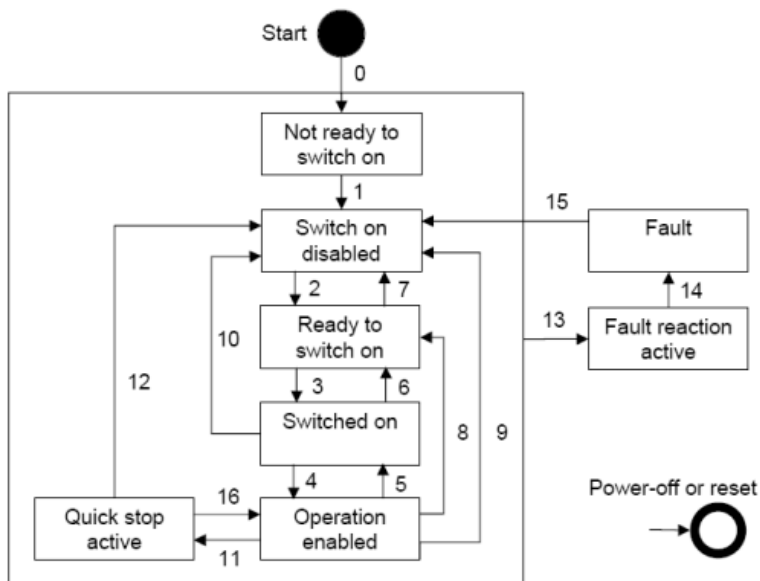
在使能运行中使用的指令位


位	值	描述
4 (使能斜坡)	0	保持前一个运行状态
	1	通过指令位的变频器运行
5 (解锁斜坡)	0	保持输出频率
	1	运行到目标频率
6 (参考斜坡)	0	输入目标频率为0
	1	输入目标频率为设置值
8 (停止)	X	未使用

根据控制字位设置的变频器状态



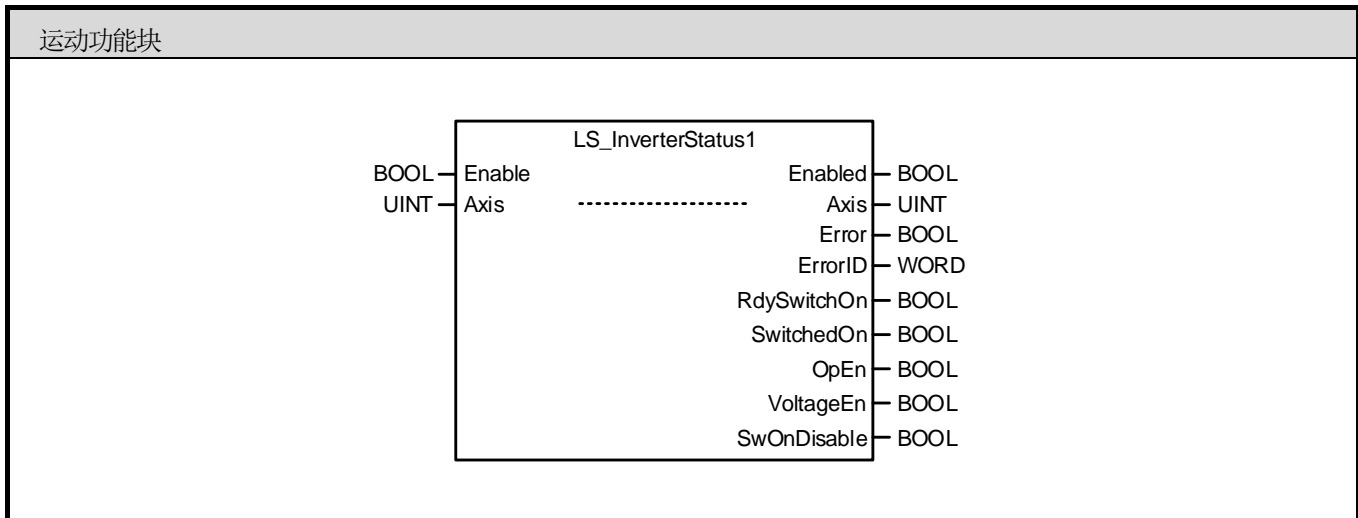
根据控制字位设置变更变频器状态



Command	Bits of the controlword					Transitions
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	X	1	1	0	2,6,8
Switch on	0	0	1	1	1	3
Switch on + enable operation	0	1	1	1	1	3 + 4 (NOTE)
Disable voltage	0	X	X	0	X	7,9,10,12
Quick stop	0	X	0	1	X	7,10,11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4,16
Fault reset		X	X	X	X	15

NOTE Automatic transition to Enable operation state after executing SWITCHED ON state functionality.

6.6.20 读取变频器状态 1 (LS_InverterStatus1)



输入 - 输出

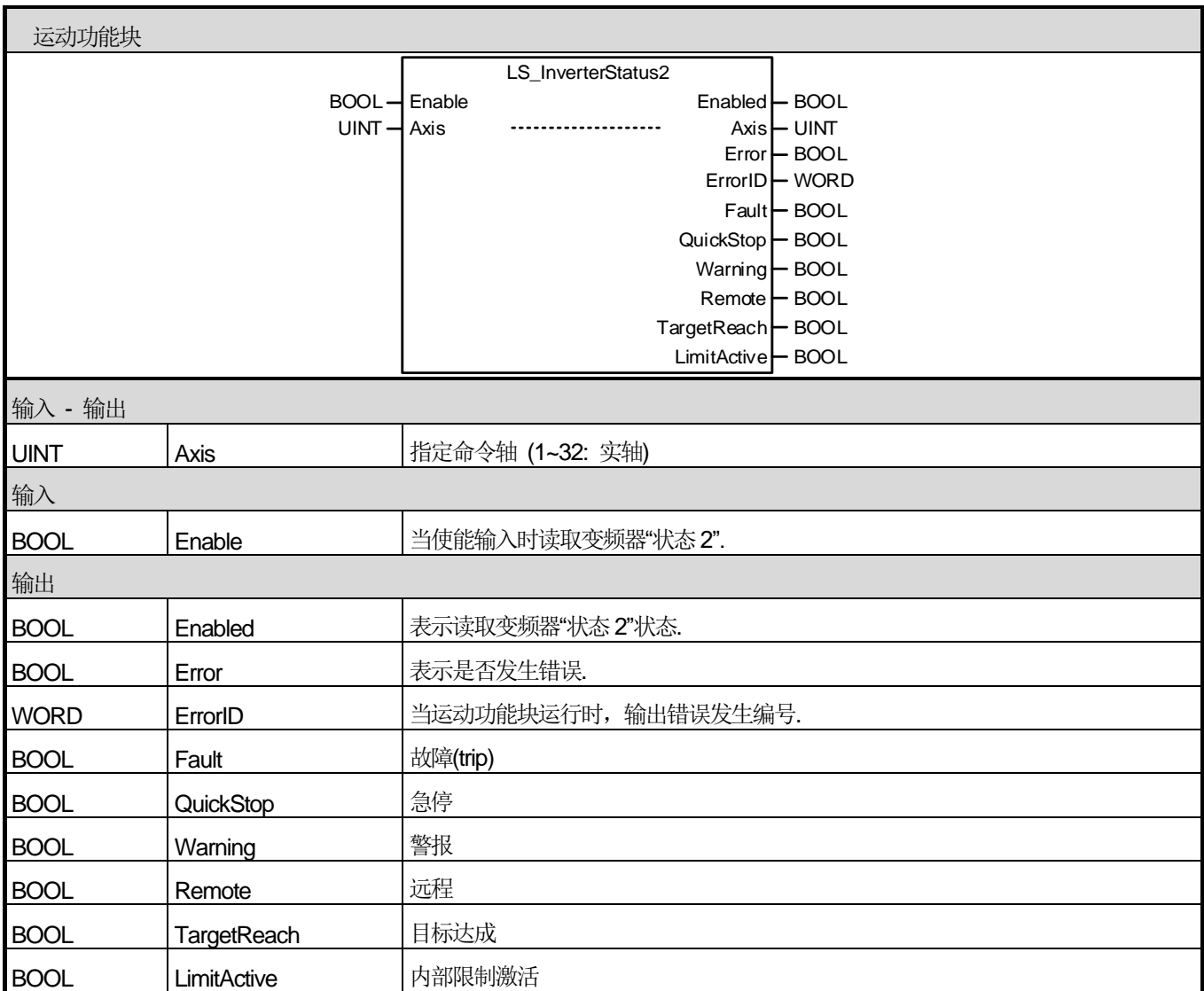
UINT	Axis	指定命令轴 (1~32: 实轴)
输入		
BOOL	Enable	当输入使能时读取变频器“状态 1”.
输出		
BOOL	Enabled	表示读取变频器“状态 1”的状态.
BOOL	Error	表示是否发生错误.
WORD	ErrorID	当运动功能块运行时，输出错误发生编号.
BOOL	RdySwitchOn	准备开关 On
BOOL	SwitchedOn	开关 On
BOOL	OpEn	运行使能
BOOL	VoltageEn	电压使能
BOOL	SwOnDisable	禁止开关 On

- 当通过轴控制变频器时，该运动功能块读取并显示连接变频器的“状态 1”.
- RdySwitchOn, SwitchedOn, OpEn, VoltageEn, SwOnDisable 分别是变频器 PDO 数据中状态字的低位值.

RdySwitchOn	位 0
SwitchedOn	位 1
OpEn	位 2
VoltageEn	位 4
SwOnDisable	位 6

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nu	nu	nu	Nu	lla	tr	rm	nu	w	sod	qs	Ve	f	oe	so	rtso

6.6.21 读取变频器状态 2 (LS_InverterStatus2)

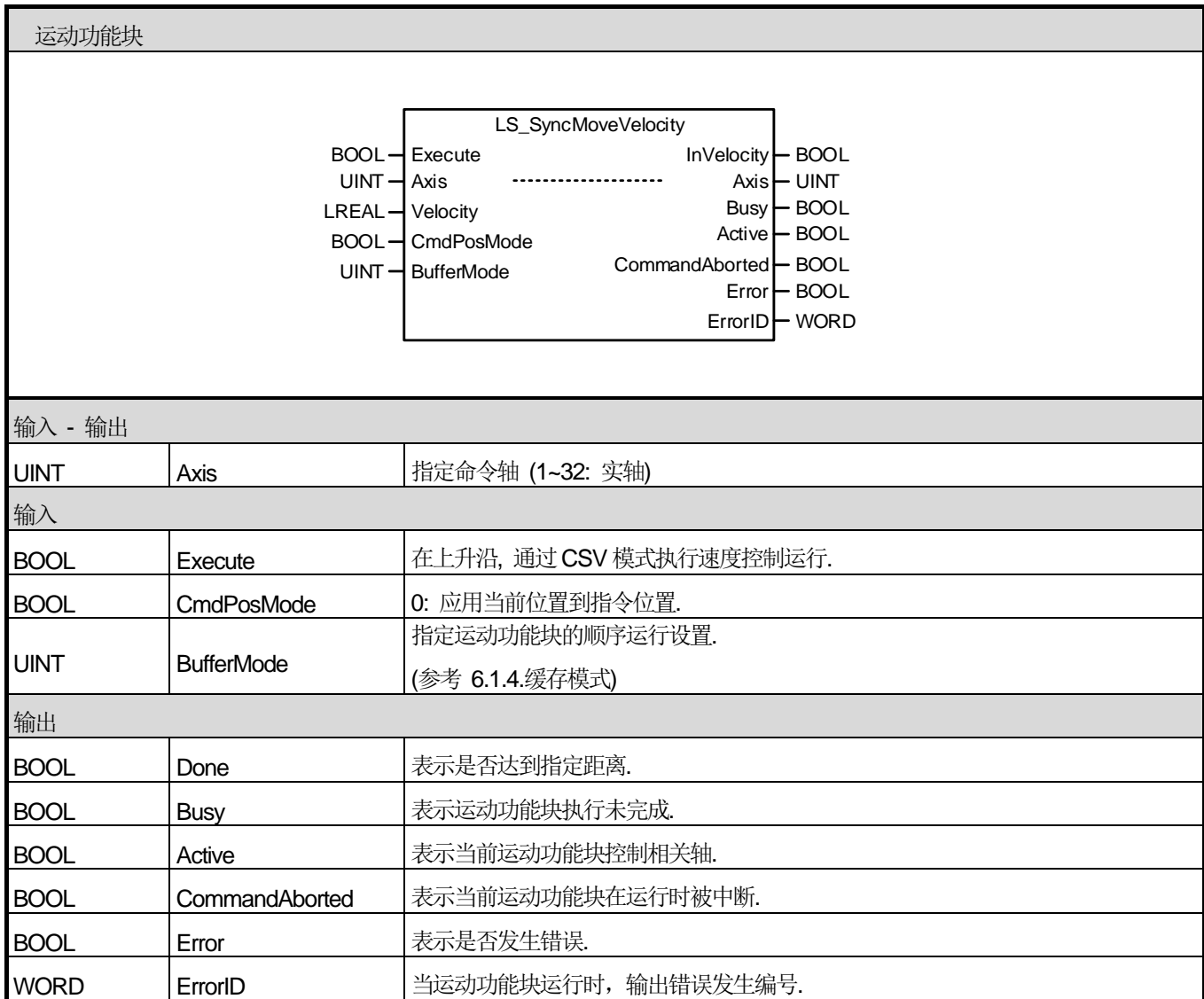


- 当通过轴控制变频器时，该运动功能块读取并显示连接变频器的“状态 2”。
- Fault, QuickStop, Warning, Remote, TargetReach, LimitActive 分别是变频器 PDO 数据中状态字的低位值。

Fault	位 3
QuickStop	位 5
Warning	位 7
Remote	位 6
TargetReach	位 10
LimitActive	位 11

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nu	nu	nu	Nu	lla	tr	rm	nu	w	sod	qs	Ve	f	oe	so	rtso

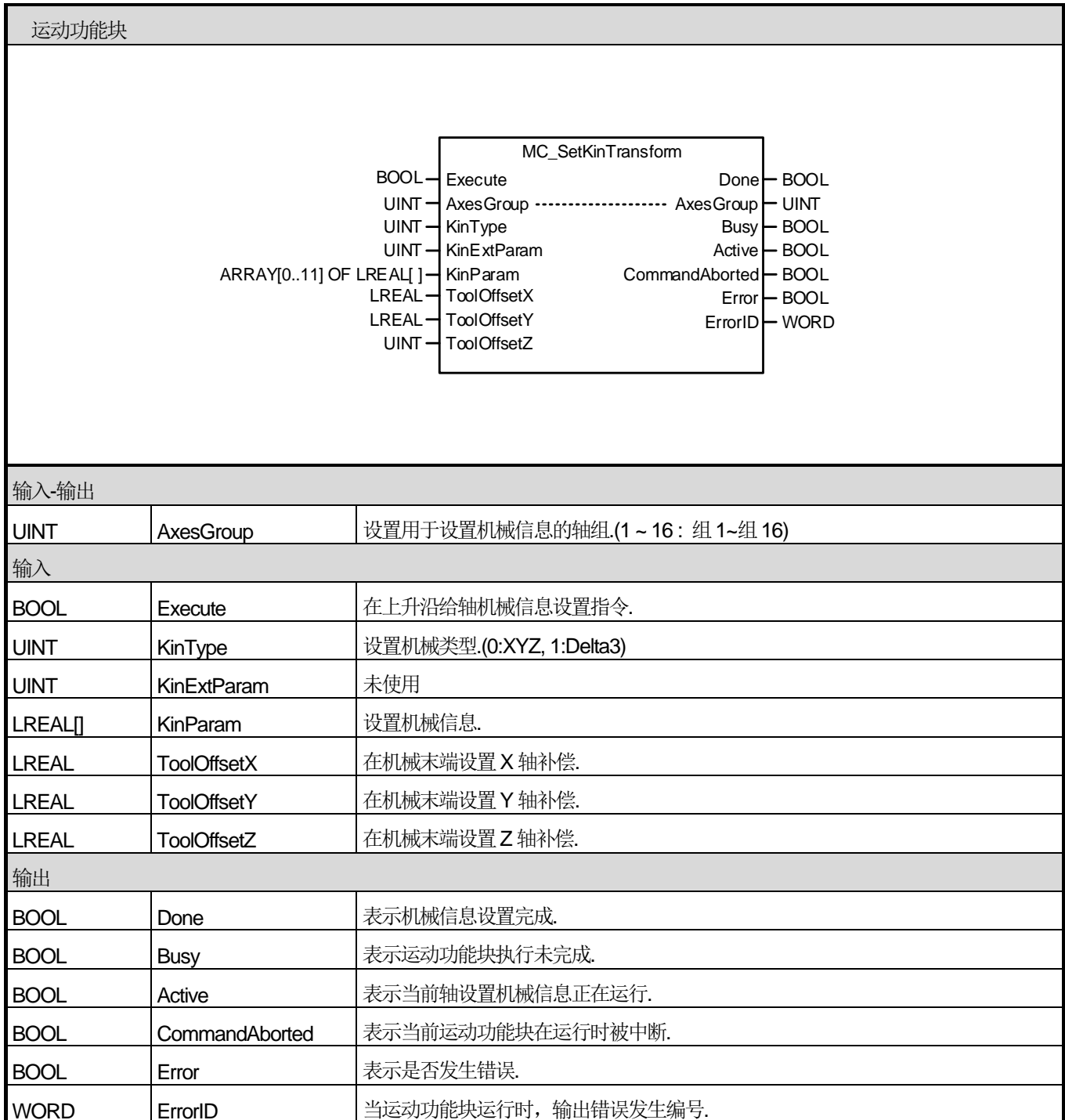
6.6.22 速度控制运行(CSV 模式) (LS_SyncMoveVelocity)



- (1) 该运动功能块允许使用在设置轴的 CiA402 轮廓 CSV(循环同步速率)模式进行速度控制.
- (2) 为了停止指定速度运行, 可以执行停止指令或者执行另一个运动功能块.
- (3) 指定速率输入的速度运行.当运行速度值符号为正数(+或者无符号)时, 正向移动, 当为负数(-)时, 反向移动.
- (4) CmdPosMode 用于在指令时间中设置当前位置更新方式.初始值仅可用为 0, 指令当前位置通过反馈当前位置进行更新.
- (5) 当轴到达指定速度时输出 InVelocity 变为 on, 当指定速度运行停止时变为 off.
- (6) 当运动功能块运行时, 轴状态为持续动作.

6.7 坐标系运行功能块

6.7.1 机械信息设置(MC_SetKinTransform)

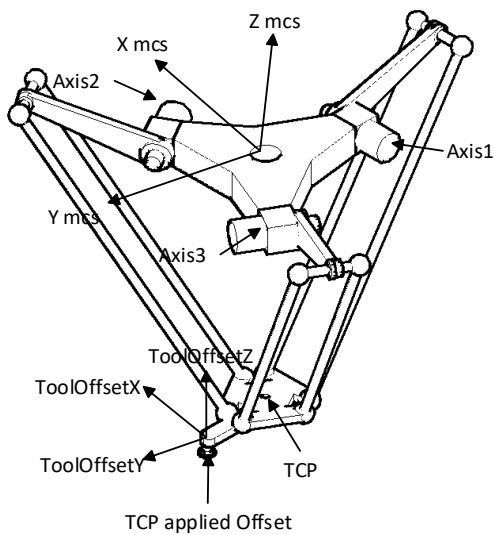


第 6 章运动功能块

- (1) 该运动功能块设置的 ACS 和 MCS 转换, 基于定义于轴组中的机械模式.
- (2) 相同设置可以应用于 MP500 组参数设置.

Group	Name	Axis group 1
Coordinate system configuration	Coordinate system Type	0: None
	Coordinate system parameter1	0
	Coordinate system parameter2	0
	Coordinate system parameter3	0
	Coordinate system parameter4	0
	Coordinate system parameter5	0
Tool configuration	X-axis offset	0 mm
	Y-axis offset	0 mm
	Z-axis offset	0 mm

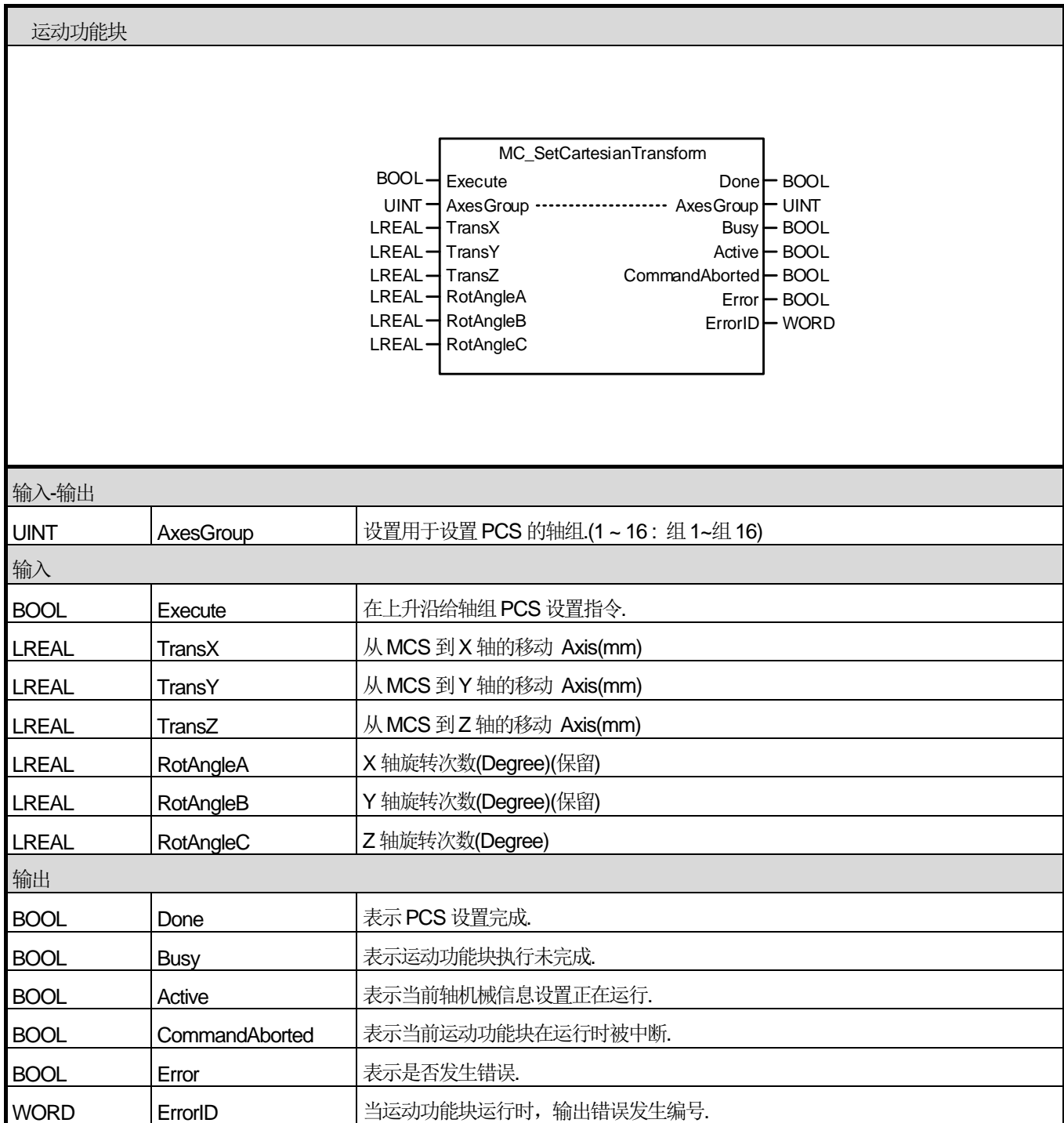
- (3) KinType输入用于设置设备类型. 可以设置如下设备.
 - 1) 0: None
 - 2) 1: XYZ
 - 3) 2: Delta3
 - 4) 3: Delta3R
 - 5) 4: LinearDelta3
 - 6) 5: LinearDelta3R
- (4) KinParam 输入用于设置设备信息. (不可以设置XYZ类型.)
- (5) ToolOffsetX / ToolOffsetY / ToolOffsetZ为在设备尾端设置补偿的功能. 为了处理连接到机器人TCP尾端的分类设备案例, 从设备信息中分别提供刀具偏置功能.



(6) 当使用 Delta3 时, 设备设置信息如下. 关于更多信息, 参考 8.4.4 机械信息设置.

	参数	描述
	KinParam[0]	Lf: 固定帧连接长度(mm)
	KinParam[1]	Lm: 移动帧连接长度(mm)
	KinParam[2]	Rf: 从固定帧中心到固定帧连接的长度(mm)
KinParam[3]	Rm: 从移动帧中心到移动帧连接的长度(mm)	

6.7.2 PCS 设置 (MC_SetCartesianTransform)

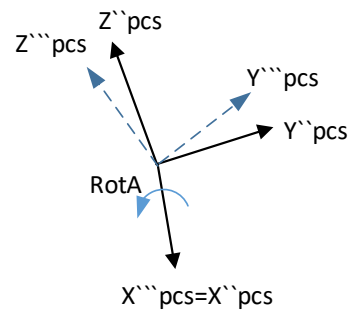
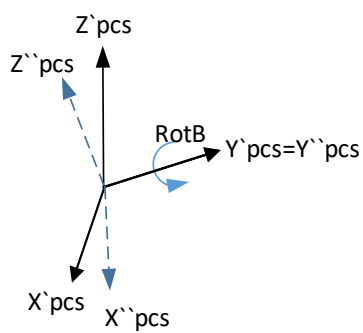
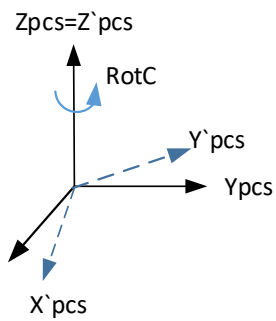
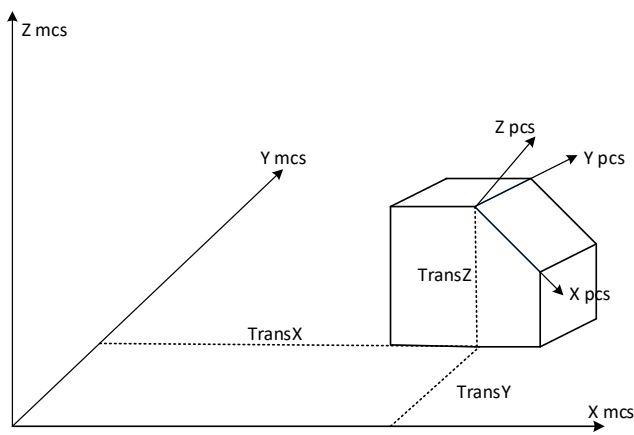


(1) 该运动功能块设置轴组中 MCS 和 PCS 的垂直坐标转换.

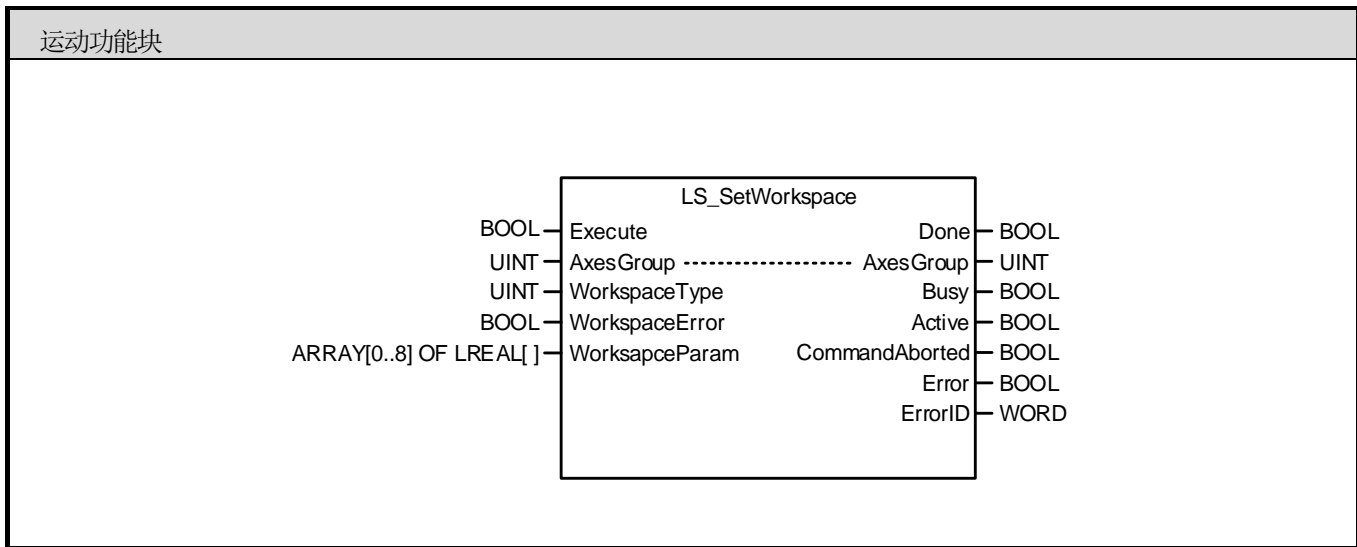
(2) 轴组设置可以以 MP500 轴组参数设置相同的方式进行设置.

Group	Name	Axis group 1
PCS Configuration	X-axis feed amount	0 mm
	Y-axis feed amount	0 mm
	Z-axis feed amount	0 mm
	X-axis rotation	0 deg
	Y-axis rotation	0 deg
	Z-axis rotation	0 deg

(3) TransX/TransY/TransZ 是从 MCS 原点到 PCS 原点的移动距离. RotA/RotB/RotC 是 PCS 旋转值, RotA 是 PCS 的 X 轴旋转 PCS 值, RotB 是 PCS 的 Y 轴旋转 PCS 值, RotC 是 PCS 的 Z 轴旋转 PCS 值.PCS 旋转必须按照 RotC, RotB, RotA 的顺序执行. 更多信息参考 8.4.3 运动控制器手册 PCS 设置.



6.7.3 工作区设置(LS_SetWorkspace)



输入-输出		
UINT	AxesGroup	设置用于设置工作区的轴组.(1 ~ 16: 组 1~组 16)
输入		
BOOL	Execute	在上升沿给轴组工作区设置指令.
UINT	WorkspaceType	设置工作区类型 (1:矩形 2:圆柱形 3:Delta3 4:扇形)
BOOL	WorkspaceError	设置当坐标系运行超出工作区时是否发生错误.
LREAL[]	WorkspaceParam	设置工作区参数.
输出		
BOOL	Done	表示 PCS 设置完成.
BOOL	Busy	表示运动功能块执行未完成.
BOOL	Active	表示当前轴机械信息设置正在运行.
BOOL	CommandAborted	表示当前运动功能块在运行时被中断.
BOOL	Error	表示是否发生错误.
WORD	ErrorID	当运动功能块运行时, 输出错误发生编号.

(1) 该运动功能块基于由轴组输入指定的轴组中的坐标系统设置工作区.

(2) 相同设置可适用于MP500组参数设置.

Workspace configuration	Workspace type	0: Rectangle
	Workspace error check	0: Disable
	Workspace Parameter1	170 mm
	Workspace Parameter2	-170 mm
	Workspace Parameter3	170 mm
	Workspace Parameter4	-170 mm
	Workspace Parameter5	-380 mm
	Workspace Parameter6	-580 mm
	Workspace Parameter7	0
Workspace Parameter8	0	

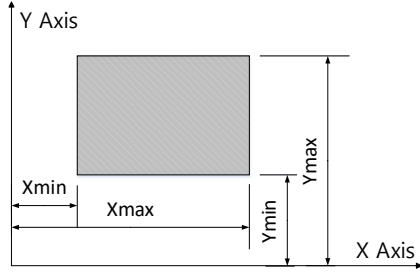
(3) 工作区类型可选择 4 种类型(1: 矩形 2:圆柱 3: Delta3 4:扇形).

(4) WorkspaceError 输入由当坐标系运行超出工作区范围时是否发生错误决定.

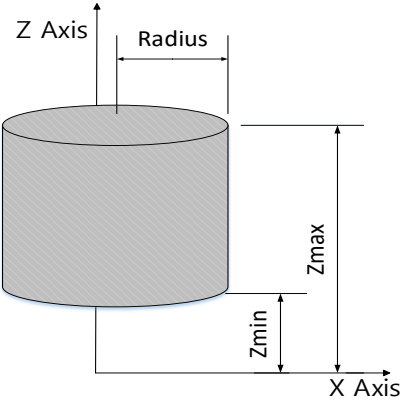
(5) WorkspaceParam 输入根据工作区类型设置参数.

(6) 工作区参数设置解释如下. 更多内容参考8.4.5运动控制器手册的工作区设置.

1) 矩形

	参数	值
	WorkspaceParam[0]	X 最大(mm)
	WorkspaceParam[1]	X 最小(mm)
	WorkspaceParam[2]	Y 最大(mm)
	WorkspaceParam[3]	Y 最小(mm)
	WorkspaceParam[4]	Z 最大(mm)
WorkspaceParam[5]	Z 最小(mm)	

2) 圆柱

	参数	值
	WorkspaceParam[0]	半径(mm)
	WorkspaceParam[1]	Z 最大(mm)
WorkspaceParam[2]	Z 最小(mm)	

第 6 章运动功能块

3) Delta

	参数	值
	WorkspaceParam[0]	Zu(mm)
	WorkspaceParam[1]	Hcy(mm)
	WorkspaceParam[2]	Hco(mm)
	WorkspaceParam[3]	Rcy(mm)
	WorkspaceParam[4]	Rco(mm)
WorkspaceParam[5]	-	

4) Sector

	参数	值
	WorkspaceParam[0]	L end (mm)
	WorkspaceParam[1]	L start(mm)
	WorkspaceParam[2]	Z max(mm)
	WorkspaceParam[3]	Z min(mm)
	WorkspaceParam[4]	结束角(度)
WorkspaceParam[5]	起始角(度)	

6.7.4 坐标系统绝对位置的时间-线性插值运行

(LS_MoveLinearTimeAbsolute)

运动功能块		
<pre> graph LR subgraph LS_MoveLinearTimeAbsolute Execute[Execute] AxesGroup[AxesGroup] CoordSystem[CoordSystem] Position[Position] TrajType[TrajType] TrajTime[TrajTime] BufferMode[BufferMode] TransitionMode[TransitionMode] TransitionParameter[TransitionParameter] Done[Done] Busy[Busy] Active[Active] Error[Error] ErrorID[ErrorID] end Execute --- Done AxesGroup --- AxesGroup CoordSystem --- Busy Position --- Active TrajType --- Error TrajTime --- ErrorID BufferMode --- ErrorID TransitionMode --- ErrorID TransitionParameter --- ErrorID </pre>		
输入-输出		
UINT	AxesGroup	设置用于设置绝对位置时间线性插值的轴组.(1 ~ 16: 组 1~组 16)
输入		
BOOL	Execute	在上升沿给轴组时间线性插值指令.
UINT	CoordSystem	设置坐标系统类型 (1:MCS 2:PCS)
LREAL[]	Position	输入机械结束点的目标位置.
UINT	TrajType	输入运行 acc/dec 类型.(0:梯形 1:正弦 2:正弦 2)
LREAL	TrajTime	设置到达目标位置的实际时间.(msec)
UINT	BufferMode	指定运动功能块的顺序运行设置. (参考 6.1.4.缓存模式)
UINT	TransitionMode	未使用
LREAL	TransitionParameter	未使用
输出		
BOOL	Done	表示 PCS 设置完成.
BOOL	Busy	表示运动功能块执行未完成.
BOOL	Active	表示当前轴机械信息设置正在运行.
BOOL	Error	表示是否发生错误.
WORD	ErrorID	当运动功能块运行时, 输出错误发生编号.

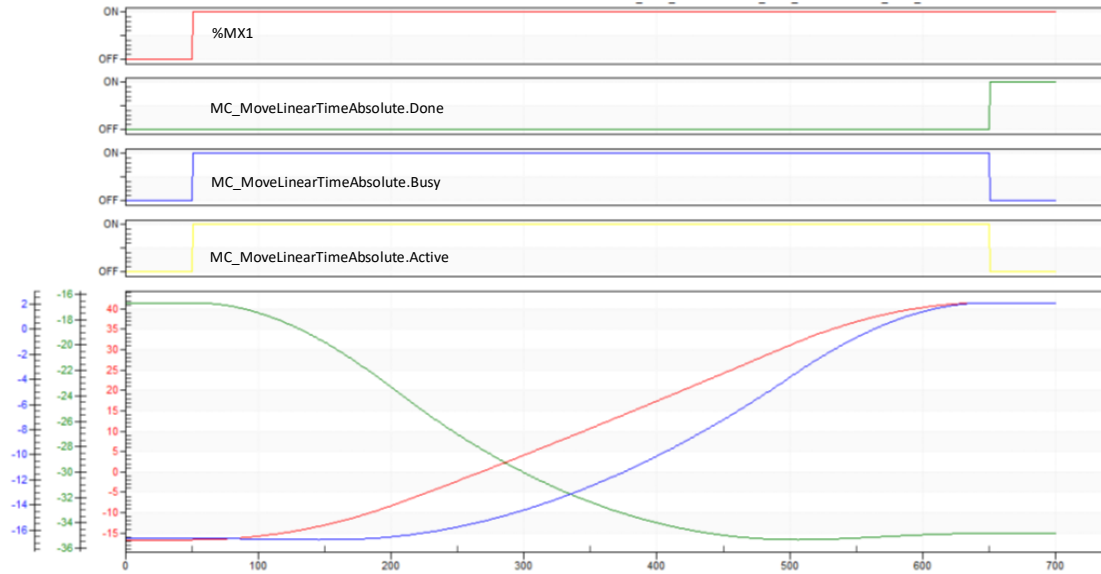
第 6 章运动功能块

- (1) 该运动功能块基于通过轴组输入指定轴组的坐标系发布绝对位置/时间线性插值指令
- (2) 当执行该运动功能块时, 在从每个轴组机械结束点到目标位置的直线轨迹中执行插值控制.
- (3) TrajType 输入设置插值轨迹的速率类型, 加速, 减速. 可以从以下 3 个类型中选择: 梯形/正弦 1/正弦 2.
- (4) TrajTime 设置到达目标位置的实际时间.
- (5) 更多内容参考 8. 4. 6 坐标系绝对位置/时间线性插值控制.
- (6) 实例程序
实例显示当前位置为 MCS 坐标系统的 0,0,-380 时, 线性插值到 MCS(100, 200,-380)的目标位置.

(a) 功能块设置

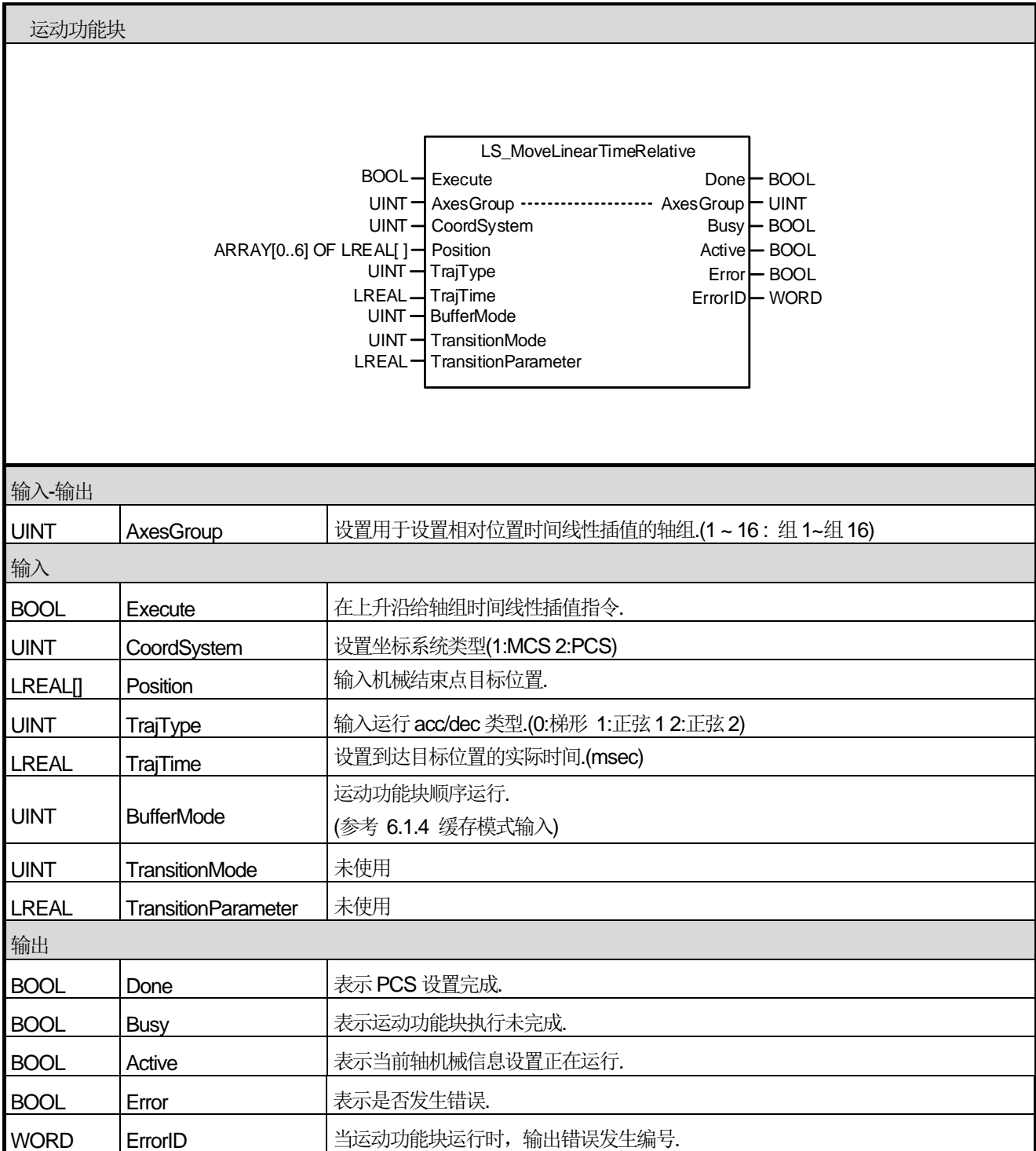


(b) 时序图



6.7.5 坐标系统相对位置的时间-线性插值运行

(LS_MoveLinearTimeRelative)



(1) 该运动功能块基于通过轴组输入指定轴组的坐标系统发布相对位置/时间线性插值指令

第 6 章运动功能块

- (2) 当执行该运动功能块时, 在从每个轴组机械结束点到目标位置的直线轨迹中执行插值控制.
- (3) TrajType 输入设置插值轨迹的速率类型,加速,减速. 可以从以下 3 个类型中选择: 梯形/正弦 1/正弦 2.
- (4) TrajTime 设置到达目标位置的实际时间.
- (5) 更多内容参考 8. 4. 6 坐标系统相对位置/时间线性插值控制.
- (6) 实例程序

实例显示当当前位置为 MCS 坐标系统的 0,0,-380 时, 线性插值到 MCS(100, 200,-380)的目标位置.

6.7.6 坐标系统绝对位置的圆弧插值运行

(MC_MoveCircularAbsolute2D)

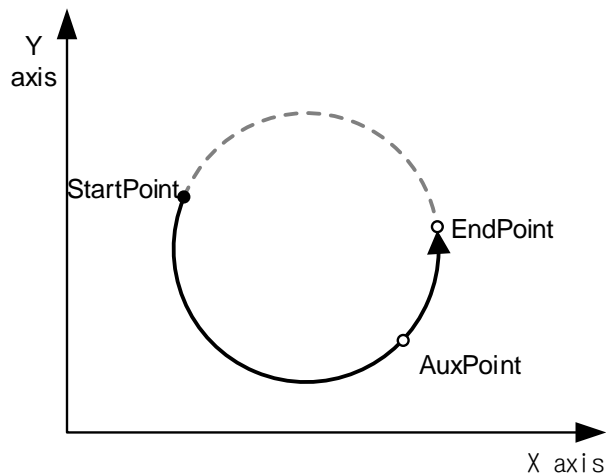
运动功能块																																												
<div style="border: 1px solid black; padding: 10px; margin: 0 auto; width: fit-content;"> <p style="text-align: center; margin: 0;">MC_MoveCircularAbsolute2D</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%; text-align: right;">BOOL — Execute</td> <td style="width: 30%;"></td> <td style="width: 30%; text-align: left;">Done — BOOL</td> </tr> <tr> <td style="text-align: right;">UINT — AxesGroup</td> <td style="text-align: center;">-----</td> <td style="text-align: left;">AxesGroup — UINT</td> </tr> <tr> <td style="text-align: right;">UINT — CircMode</td> <td></td> <td style="text-align: left;">Busy — BOOL</td> </tr> <tr> <td style="text-align: right;">LREAL[] — AuxPoint</td> <td></td> <td style="text-align: left;">Active — BOOL</td> </tr> <tr> <td style="text-align: right;">LREAL[] — EndPoint</td> <td></td> <td style="text-align: left;">CommandAborted — BOOL</td> </tr> <tr> <td style="text-align: right;">UINT — PathChoice</td> <td></td> <td style="text-align: left;">Error — BOOL</td> </tr> <tr> <td style="text-align: right;">LREAL — Velocity</td> <td></td> <td style="text-align: left;">ErrorID — WORD</td> </tr> <tr> <td style="text-align: right;">LREAL — Acceleration</td> <td></td> <td></td> </tr> <tr> <td style="text-align: right;">LREAL — Deceleration</td> <td></td> <td></td> </tr> <tr> <td style="text-align: right;">LREAL — Jerk</td> <td></td> <td></td> </tr> <tr> <td style="text-align: right;">UINT — CoordSystem</td> <td></td> <td></td> </tr> <tr> <td style="text-align: right;">UINT — BufferMode</td> <td></td> <td></td> </tr> <tr> <td style="text-align: right;">UINT — TransitionMode</td> <td></td> <td></td> </tr> <tr> <td style="text-align: right;">LREAL — TransitionParameter</td> <td></td> <td></td> </tr> </table> </div>			BOOL — Execute		Done — BOOL	UINT — AxesGroup	-----	AxesGroup — UINT	UINT — CircMode		Busy — BOOL	LREAL[] — AuxPoint		Active — BOOL	LREAL[] — EndPoint		CommandAborted — BOOL	UINT — PathChoice		Error — BOOL	LREAL — Velocity		ErrorID — WORD	LREAL — Acceleration			LREAL — Deceleration			LREAL — Jerk			UINT — CoordSystem			UINT — BufferMode			UINT — TransitionMode			LREAL — TransitionParameter		
BOOL — Execute		Done — BOOL																																										
UINT — AxesGroup	-----	AxesGroup — UINT																																										
UINT — CircMode		Busy — BOOL																																										
LREAL[] — AuxPoint		Active — BOOL																																										
LREAL[] — EndPoint		CommandAborted — BOOL																																										
UINT — PathChoice		Error — BOOL																																										
LREAL — Velocity		ErrorID — WORD																																										
LREAL — Acceleration																																												
LREAL — Deceleration																																												
LREAL — Jerk																																												
UINT — CoordSystem																																												
UINT — BufferMode																																												
UINT — TransitionMode																																												
LREAL — TransitionParameter																																												
输入-输出																																												
UINT	AxesGroup	设置用于设置绝对位置圆弧插值的轴组.(1 ~ 16 : 组 1~组 16)																																										
输入																																												
BOOL	Execute	在上升沿给轴组圆弧插值指令.																																										
UINT	CircMode	设置圆弧插值的方法 [0: 中点, 1: 中心点, 2: 半径]																																										
LREAL[]	AuxPoint	圆弧插值的辅助点位置指定为一个绝对坐标.																																										
LREAL[]	EndPoint	设置圆弧结束点绝对坐标.																																										
BOOL	PathChoice	设置圆弧路径. 0: 顺时针方向, 1: 逆时针方向																																										
LREAL	Velocity	设置路径最大速率. [u/s]																																										
LREAL	Acceleration	设置最大加速. [u/s ²]																																										
LREAL	Deceleration	设置最小减速. [u/s ²]																																										
LREAL	Jerk	设置最大加速度/减速度. [u/s ³]																																										
UINT	CoordSystem	设置坐标系类型. (1:MCS 2:PCS)																																										
UINT	BufferMode	指定运动功能块的顺序运行设置. (参考 6.1.4.缓存模式)																																										

第 6 章运动功能块

UINT	TransitionMode	未使用
LREAL	TransitionParameter	未使用
输出		
BOOL	Done	表示是否到达指定点.
BOOL	Busy	表示运动功能块执行未完成.
BOOL	Active	表示运动功能块是否控制组.
BOOL	CommandAborted	表示当前运动功能块在运行时被中断.
BOOL	Error	表示是否发生错误.
WORD	ErrorID	当运动功能块运行时, 输出错误发生编号.

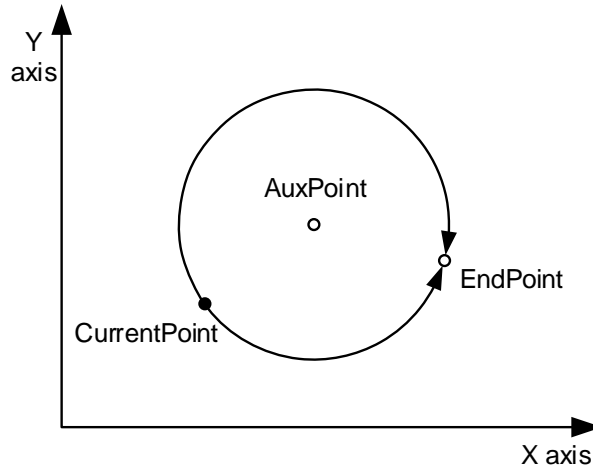
- (1) 该运动功能块基于通过轴组输入指定轴组的坐标系发布圆弧插值指令.
- (2) 当运动功能块启动时, 参考辅助点输入, 每个轴执行圆弧轨迹插值控制, 移动方向由路径选择输入决定. 如果路径选择输入设置为 0, 圆弧插值以顺时针运行, 如果设置为 1, 圆弧插值以逆时针运行.
- (3) 在 *AuxPoint* 和 *EndPoint* 输入中, 参考每轴圆弧插值确定指定辅助点绝对位置分配. 按照 X, Y, Z 顺序输入, 与 *MC_MoveCircularAbsolute* 不同.
- (4) *Velocity*, *Acceleration*, *Deceleration*, *Jerk* 输入分别设置速率, 加速, 减速, 和插值路径加速/减速变化比率.
- (5) *CircMode* 输入设置圆弧插值方式. 对应 *CircMode* 值的圆弧插值方式如下.
 - (a) 使用中点规格的圆弧插值(*CircMode* = 0)

该方法通过在起始位置启动运行执行圆弧插值, 通过指定中点, 到达目标位置. 在下图中, 起始位置对应于指令起始的轴组坐标, 中点对应于 *AuxPoint* 坐标输入, 目标位置对应于结束点绝对坐标输入.



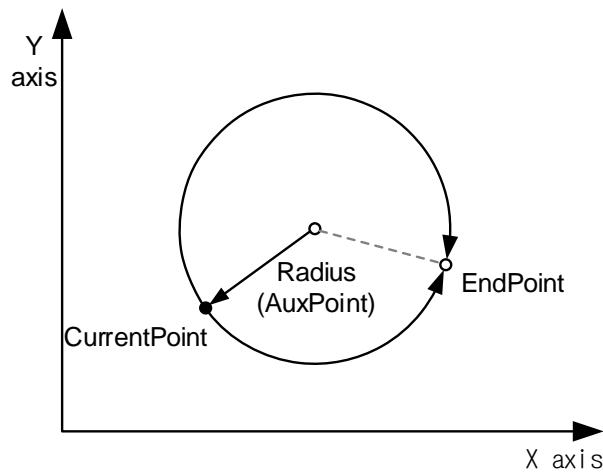
(b) 使用中心点规格的圆弧插值 (CircMode = 1)

该方法通过在当前位置启动运行执行圆弧插值到目标位置, 以下为一个半径对应于到指定中心点距离的圆弧轨迹. 在下图中, 当前位置对应于在指令起始的轴组坐标, 中心点对应于辅助点的坐标输入, 目标位置对应于结束点绝对坐标输入.



(c) 使用半径规格的圆弧插值 (CircMode = 2)

该方法通过在当前位置的启动运行执行圆弧插值到目标位置, 以下为从当前位置到目标位置指定半径的圆弧轨迹. 在下图中, 当前位置对应于指令起始的轴组坐标, 半径对应于辅助点 X 坐标输入, 目标位置对应于结束点绝对坐标输入.



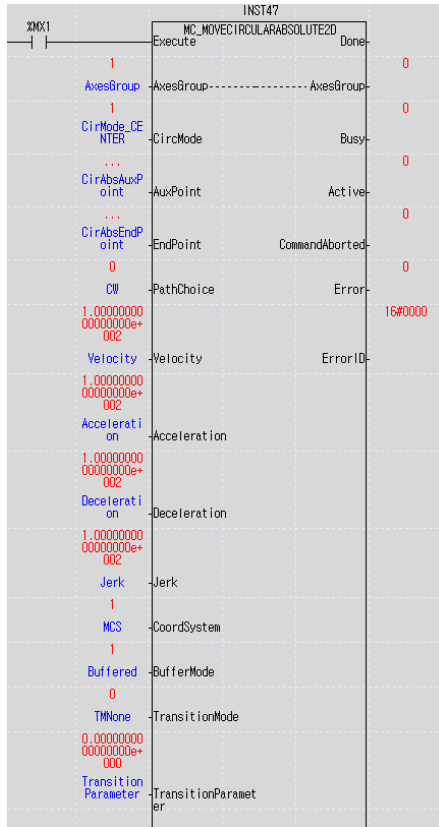
- (6) 更多内容参考 8.4.7 运动控制器手册的圆弧插值控制.
- (7) 指令完成前变更参数通过再次执行功能(执行输入为 On)块进行应用.
- (8) 仅速率, 加速, 减速, 加速度, 辅助点, 结束点输入可以更新.

第 6 章运动功能块

(9) 实例程序

实例为当前位置为 MCS (0,150,-580)时，设置中心点(0, 75, -580)，通过顺时针方向移动执行圆弧插值到目标位置 MCS(0,0,-580)。

(a) 功能块设置

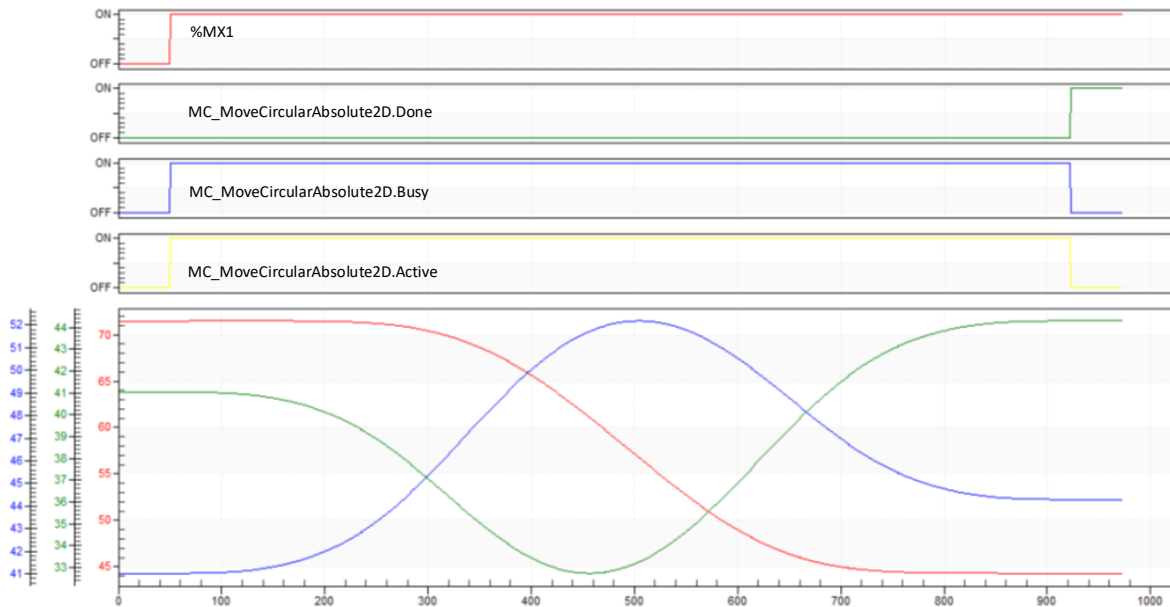


1	_AX01_CPOS	7.1485269050584748e+001	LREAL
2	_AX02_CPOS	4.1026455810822632e+001	LREAL
3	_AX03_CPOS	4.1026455810822632e+001	LREAL
4	CirAbsAuxPoint		ARRAY[0..2] OF LREAL
5	CirAbsAuxPoint[0]	0.0000000000000000e+000	LREAL
6	CirAbsAuxPoint[1]	7.5000000000000000e+001	LREAL
7	CirAbsAuxPoint[2]	-5.8000000000000000e+002	LREAL
8	CirAbsEndPoint		ARRAY[0..2] OF LREAL
9	CirAbsEndPoint[0]	0.0000000000000000e+000	LREAL
10	CirAbsEndPoint[1]	0.0000000000000000e+000	LREAL
11	CirAbsEndPoint[2]	-5.8000000000000000e+002	LREAL

CenterPoint

EndPoint

(b) 时序图



6.7.7 坐标系统相对位置的圆弧插值运行

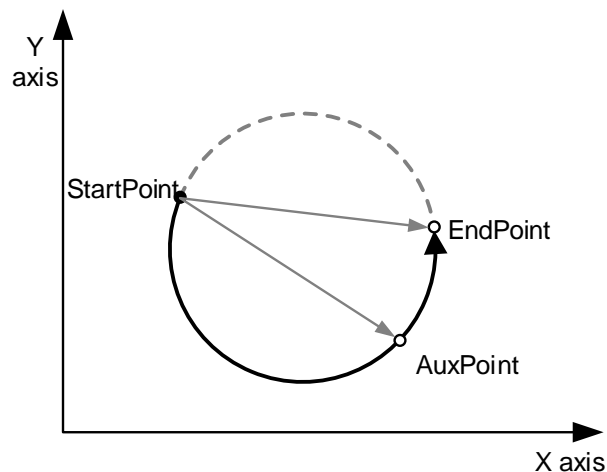
(MC_MoveCircularRelative2D)

运动功能块																																																										
<div style="border: 1px solid black; padding: 10px; margin: 0 auto; width: fit-content;"> <p style="text-align: center; margin: 0;">MC_MoveCircularRelative2D</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">BOOL</td> <td style="width: 40%;">Execute</td> <td style="width: 30%;">Done</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>AxesGroup</td> <td>AxesGroup</td> <td>UINT</td> </tr> <tr> <td>UINT</td> <td>CircMode</td> <td>Busy</td> <td>BOOL</td> </tr> <tr> <td>LREAL[]</td> <td>AuxPoint</td> <td>Active</td> <td>BOOL</td> </tr> <tr> <td>LREAL[]</td> <td>EndPoint</td> <td>CommandAborted</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>PathChoice</td> <td>Error</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>Velocity</td> <td>ErrorID</td> <td>WORD</td> </tr> <tr> <td>LREAL</td> <td>Acceleration</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Deceleration</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Jerk</td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>CoordSystem</td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>BufferMode</td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>TransitionMode</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>TransitionParameter</td> <td></td> <td></td> </tr> </table> </div>			BOOL	Execute	Done	BOOL	UINT	AxesGroup	AxesGroup	UINT	UINT	CircMode	Busy	BOOL	LREAL[]	AuxPoint	Active	BOOL	LREAL[]	EndPoint	CommandAborted	BOOL	UINT	PathChoice	Error	BOOL	LREAL	Velocity	ErrorID	WORD	LREAL	Acceleration			LREAL	Deceleration			LREAL	Jerk			UINT	CoordSystem			UINT	BufferMode			UINT	TransitionMode			LREAL	TransitionParameter		
BOOL	Execute	Done	BOOL																																																							
UINT	AxesGroup	AxesGroup	UINT																																																							
UINT	CircMode	Busy	BOOL																																																							
LREAL[]	AuxPoint	Active	BOOL																																																							
LREAL[]	EndPoint	CommandAborted	BOOL																																																							
UINT	PathChoice	Error	BOOL																																																							
LREAL	Velocity	ErrorID	WORD																																																							
LREAL	Acceleration																																																									
LREAL	Deceleration																																																									
LREAL	Jerk																																																									
UINT	CoordSystem																																																									
UINT	BufferMode																																																									
UINT	TransitionMode																																																									
LREAL	TransitionParameter																																																									
输入-输出																																																										
UINT	AxesGroup	设置用于设置相对位置圆弧插值的轴组. (1 ~ 16: 组 1~组 16)																																																								
输入																																																										
BOOL	Execute	在上升沿给轴组相对位置圆弧插值指令.																																																								
UINT	CircMode	设置圆弧插值的方法 [0: 中点, 1: 中心点, 2: 半径]																																																								
LREAL[]	AuxPoint	根据相对坐标中的圆弧插值方式指定辅助点位置.																																																								
LREAL[]	EndPoint	指定圆弧轨迹结束点作为从起始点的相对坐标.																																																								
BOOL	PathChoice	设置圆弧路径. 0: 顺时针方向, 1: 逆时针方向																																																								
LREAL	Velocity	设置路径最大速率. [u/s]																																																								
LREAL	Acceleration	设置最大加速. [u/s ²]																																																								
LREAL	Deceleration	设置最小减速. [u/s ²]																																																								
LREAL	Jerk	设置最大加速度/减速度. [u/s ³]																																																								
UINT	CoordSystem	设置坐标系类型. (1:MCS 2:PCS)																																																								
UINT	BufferMode	指定运动功能块的顺序运行设置. (参考 6.1.4.缓存模式)																																																								

第 6 章运动功能块

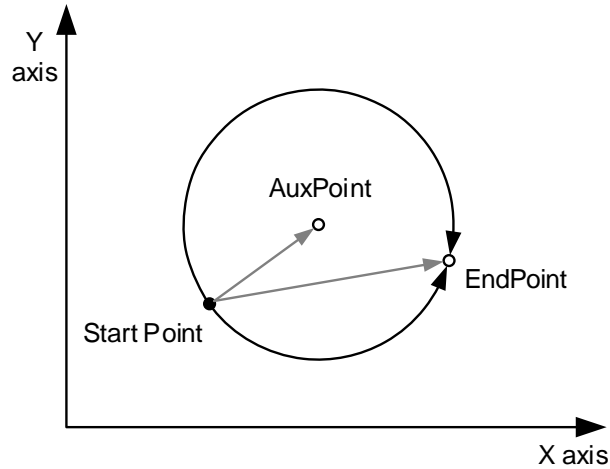
UINT	TransitionMode	未使用
LREAL	TransitionParameter	未使用
输出		
BOOL	Done	表示是否到达指定点.
BOOL	Busy	表示运动功能块执行未完成.
BOOL	Active	表示运动功能块是否控制组.
BOOL	CommandAborted	表示当前运动功能块在运行时被中断.
BOOL	Error	表示是否发生错误.
WORD	ErrorID	当运动功能块运行时, 输出错误发生编号.

- (1) 该运动功能块基于通过轴组输入指定轴组的坐标系统发布相对位置圆弧插值指令.
- (2) 当执行该运动功能块时, 参考辅助点输入, 每个轴执行圆弧轨迹插值控制, 移动方向由路径选择输入决定. 如果路径选择输入设置为 0, 圆弧插值以顺时针运行, 如果设置为 1, 圆弧插值以逆时针运行.
- (3) 在 AuxPoint 和 EndPoint 输入中, 参考每轴圆弧插值确定指定辅助点绝对位置分配. 输入分配和组轴对应于指定轴 IDs [ID1, ID2, ID3, ...], 按照这个顺序. (由于包含发布圆弧插值指令组的轴数量为 3, 对于位置输入应该分配 3 个尺寸输入)
- (4) Velocity, Acceleration, Deceleration, Jerk 输入分别设置速率, 加速, 减速, 和插值路径加速/减速变化比率.
- (5) CircMode 输入设置圆弧插值方式. 如下所示圆弧插值方式与 CircMode 值相同.
 - (a) 使用中点规格的圆弧插值 (BORDER, CircMode = 0)
该方法在当前位置启动运行后通过中点位置执行圆弧插值到目标位置. 在下图中, 当前位置对应于指令起始的轴组坐标, 中点对应于辅助点的坐标输入, 目标位置对应于结束点的相对坐标输入.



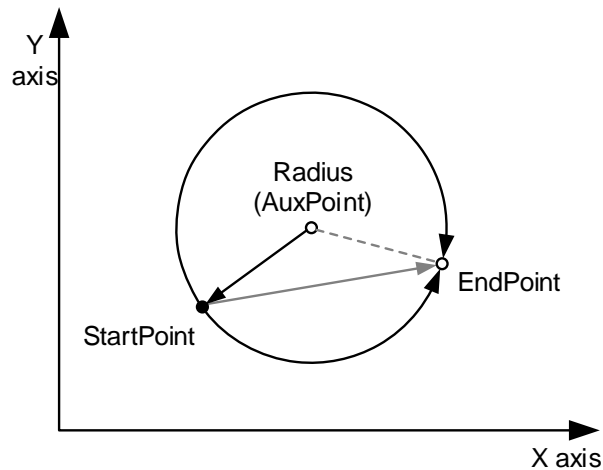
(b) 使用中心点规格的圆弧插值 (CircMode = 1)

该方法通过在启动位置启动运行执行圆弧插值到目标位置, 以下圆弧轨迹直径对应于到指定中心点的距离. 下图中, 当前位置对应于指令起始的轴组坐标, 中心点对应于辅助点坐标输入, 目标位置对应于结束点相对坐标输入.



(c) 使用半径规格的圆弧插值 (CircMode = 2)

该方法通过在当前位置启动运行执行圆弧插值到目标位置, 通过指定中心点, 到达目标位置. 下图中, 当前位置对应于在指令起始的轴组坐标, 直径对应于辅助点的 X 坐标输入, 目标位置对应于结束点的相对坐标输入.



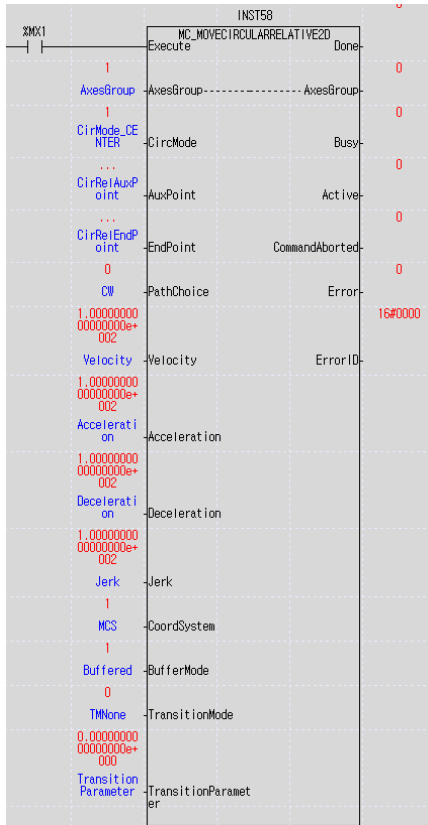
(6) 更多内容参考 8.4.7 运动控制器手册的圆弧插值控制.

(7) 实例程序

该实例当前指令位置为 1000 时设置中心点规格, 1000 (设置从当前位置到中心点的相对位置), 顺时针移动执行圆弧插值到目标位置(设置从当前位置: 0, 0 到目标位置的相对位置).

第 6 章运动功能块

(a) 功能块设置

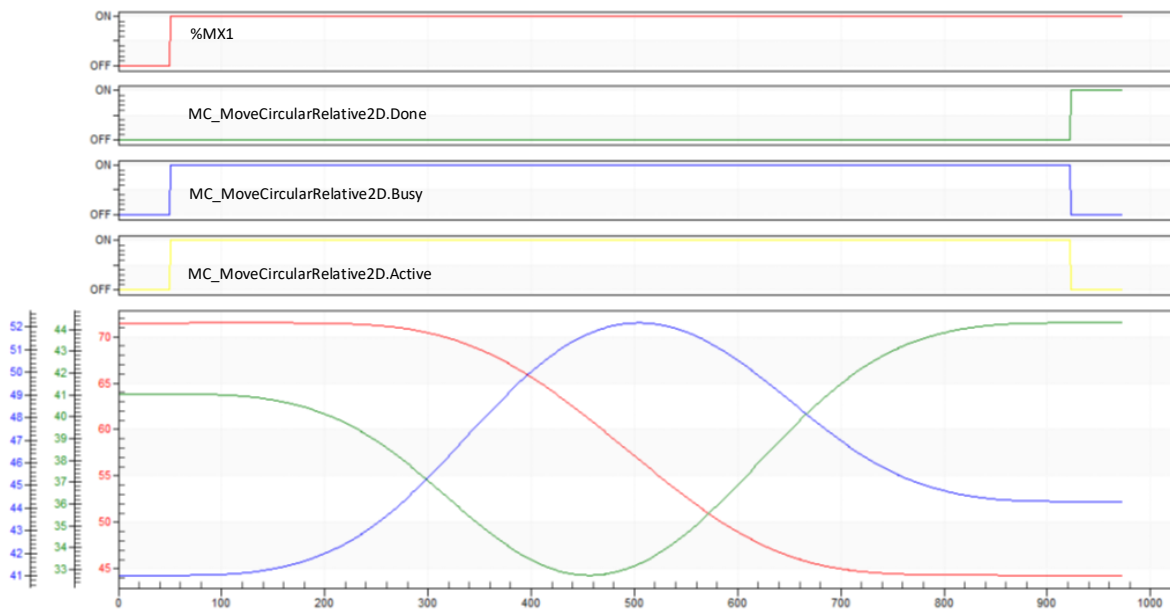


1	_AX01_CPOS	7.1485269050584748e+001	LREAL
2	_AX02_CPOS	4.1026455810822632e+001	LREAL
3	_AX03_CPOS	4.1026455810822632e+001	LREAL
4	CirRelAuxPoint	ARRAY[0..2] OF LREAL	
5	CirRelAuxPoint[0]	0.0000000000000000e+000	LREAL
6	CirRelAuxPoint[1]	-7.5000000000000000e+001	LREAL
7	CirRelAuxPoint[2]	0.0000000000000000e+000	LREAL
8	CirRelEndPoint	ARRAY[0..2] OF LREAL	
9	CirRelEndPoint[0]	0.0000000000000000e+000	LREAL
10	CirRelEndPoint[1]	-1.5000000000000000e+002	LREAL
11	CirRelEndPoint[2]	0.0000000000000000e+000	LREAL

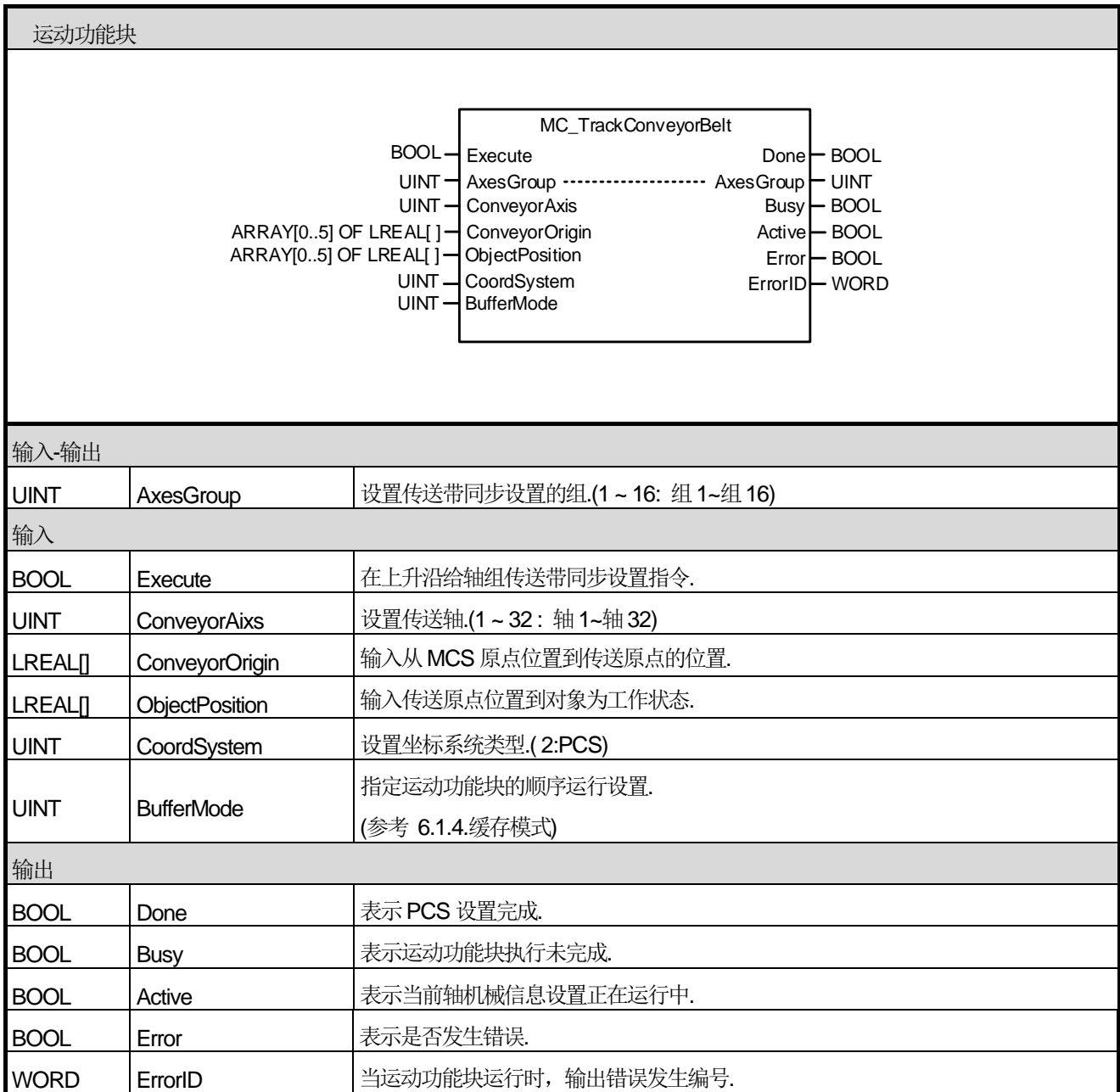
Center point

End point

(b) 时序图



6.7.8 传送带同步设置(MC_TrackConveyorBelt)

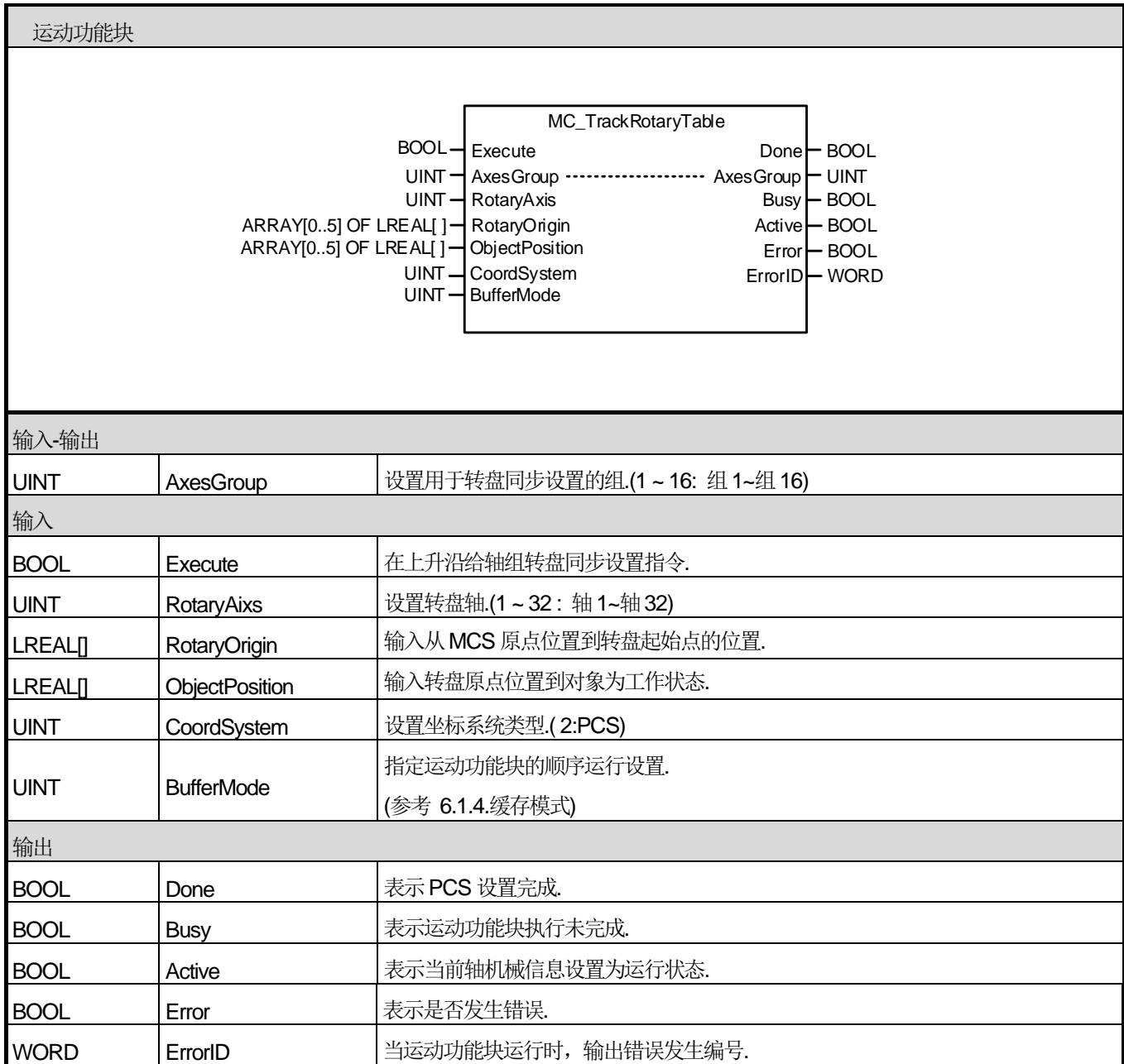


- (1) 该运动功能块设置通过轴组输入指定轴组的传送带同步运行.
- (2) 该运动功能块不直接参与运行. 当执行功能块时, 使用 PCS 坐标系统的坐标系统运行同步到指定传送带轴.

第 6 章运动功能块

- (3) 传送带轴可设置范围 1~32. 属于设置为 `AxesGroup` 轴组的轴无法指定.
- (4) 作为 `ConveyorAxis` 的指定轴运行参数必须是 `mm/inch`.
- (5) 无限重复运行必须作为 `ConveyorAxis` 设置为指定轴运行参数
- (6) 同步传送带运行通过执行使用 `PCS` 坐标系统的坐标系统运行, 或者执行 `MC_SetCartesian` 转换功能块 `PCS` 设置进行停止.
- (7) 更多内容参考8.4.9运动控制器手册的同步传送带运行

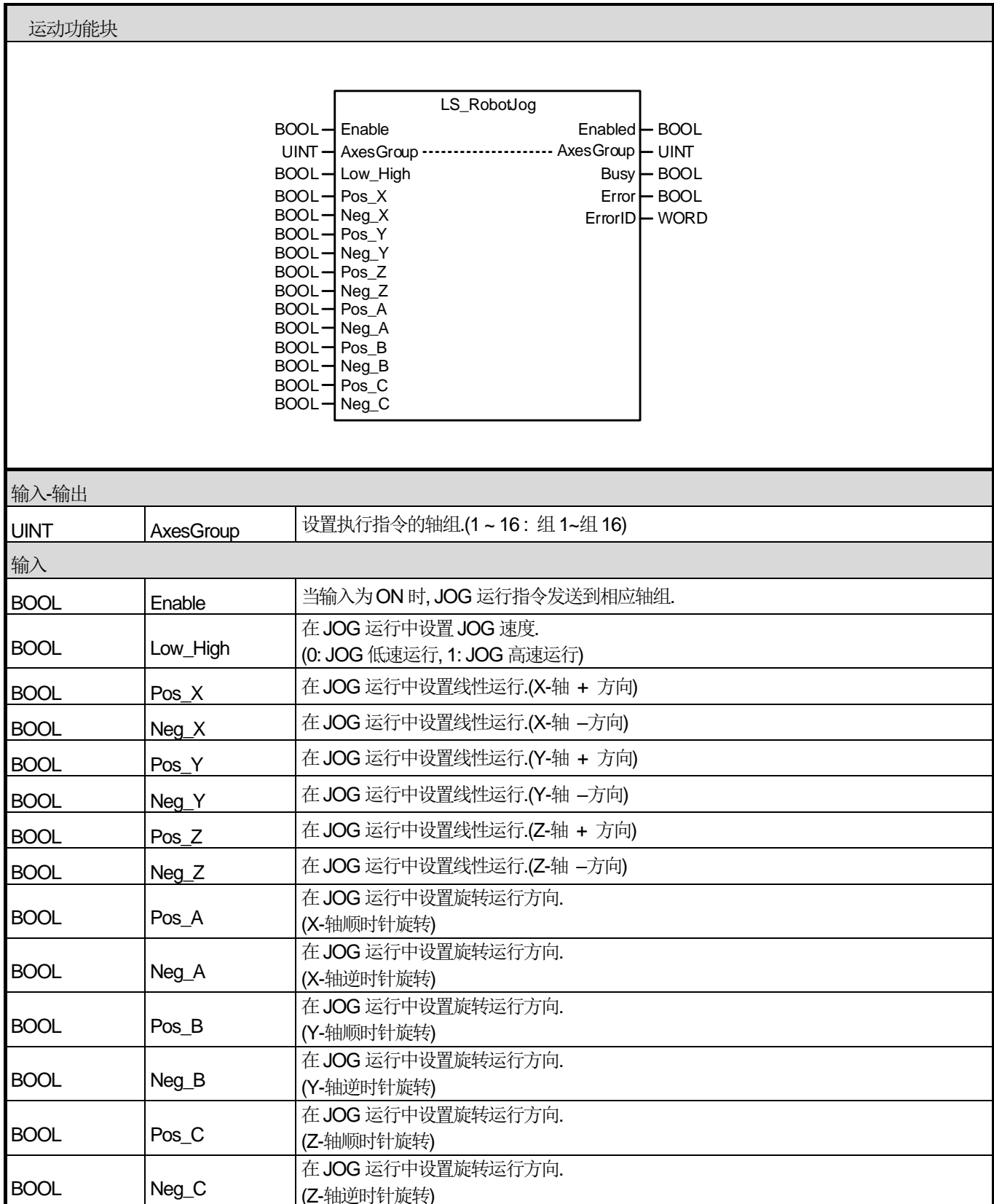
6.7.9 转盘同步设置(MC_TrackRotaryTable)



- (1) 该运动功能块通过轴组输入设置用于指定轴组的转盘同步运行.
- (2) 该运动功能块不直接参与运行. 当执行功能块时, 使用 PCS 坐标系统的坐标系统运行同步到指定转盘轴.
- (3) 转盘轴可设置范围 1~32. 属于设置为 AxesGroup 轴组的轴无法指定.
- (4) 作为 RotaryAxis 的指定轴运行参数必须是 mm/inch.
- (5) 无限重复运行必须作为 RotaryAxis 设置为指定轴运行参数
- (6) 同步转盘运行通过执行使用 PCS 坐标系统的坐标系统运行, 或者执行 MC_SetCartesian 转换功能块 PCS 设置进行停止.
- (7) 更多内容参考8.4.10运动控制器手册的同步转盘运行

第 6 章运动功能块

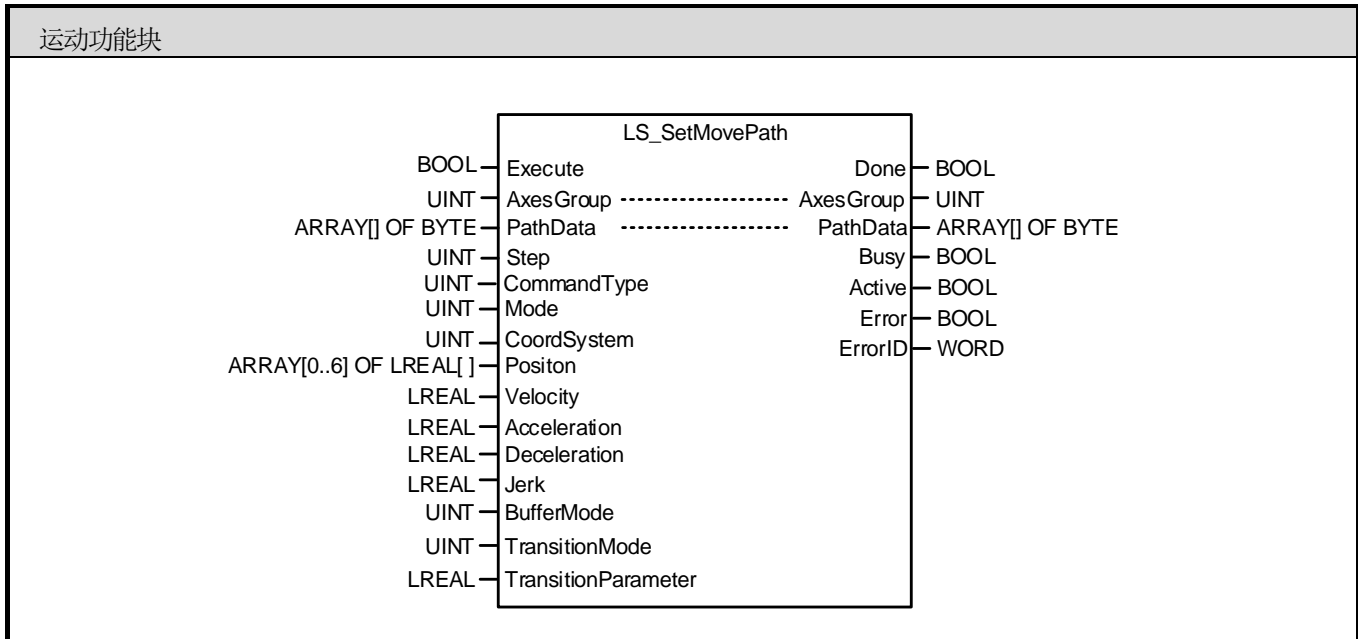
6.7.10 坐标系统 JOG 运行(MC_RobotJog)



输出		
BOOL	Enabled	表示轴组处于 JOG 运行进程.
BOOL	Busy	表示运动功能块执行未完成.
BOOL	Error	表示是否发生错误.
WORD	ErrorID	当运动功能块运行时, 输出错误发生编号.

- (1) 该运动功能块执行相应轴组的坐标系统 JOG 运行.
- (2) JOG 运行是用于测试的手动运行功能. 用于查看系统运行,连接状态,教学位置地址. 可以分别应用于高速和低速.
- (3) 当 Enable 输入为 ON(JOG 运行状态)时, 如果变更设置在低/高中的值, 将在不停止 JOG 运行的情况下改变速度.
- (4) 如果相同轴同时设置正向(Pos_) /反向(Neg_)输入, 轴将停止.

6.7.11 设置路径运行数据 (MC_SetMovePath)



输入-输出

UINT	AxesGroup	设置用于设置路径运行数据的组 (1 ~ 16: 组 1~组 16)
BYTE[]	PathData	设置保存路径数据的位置.

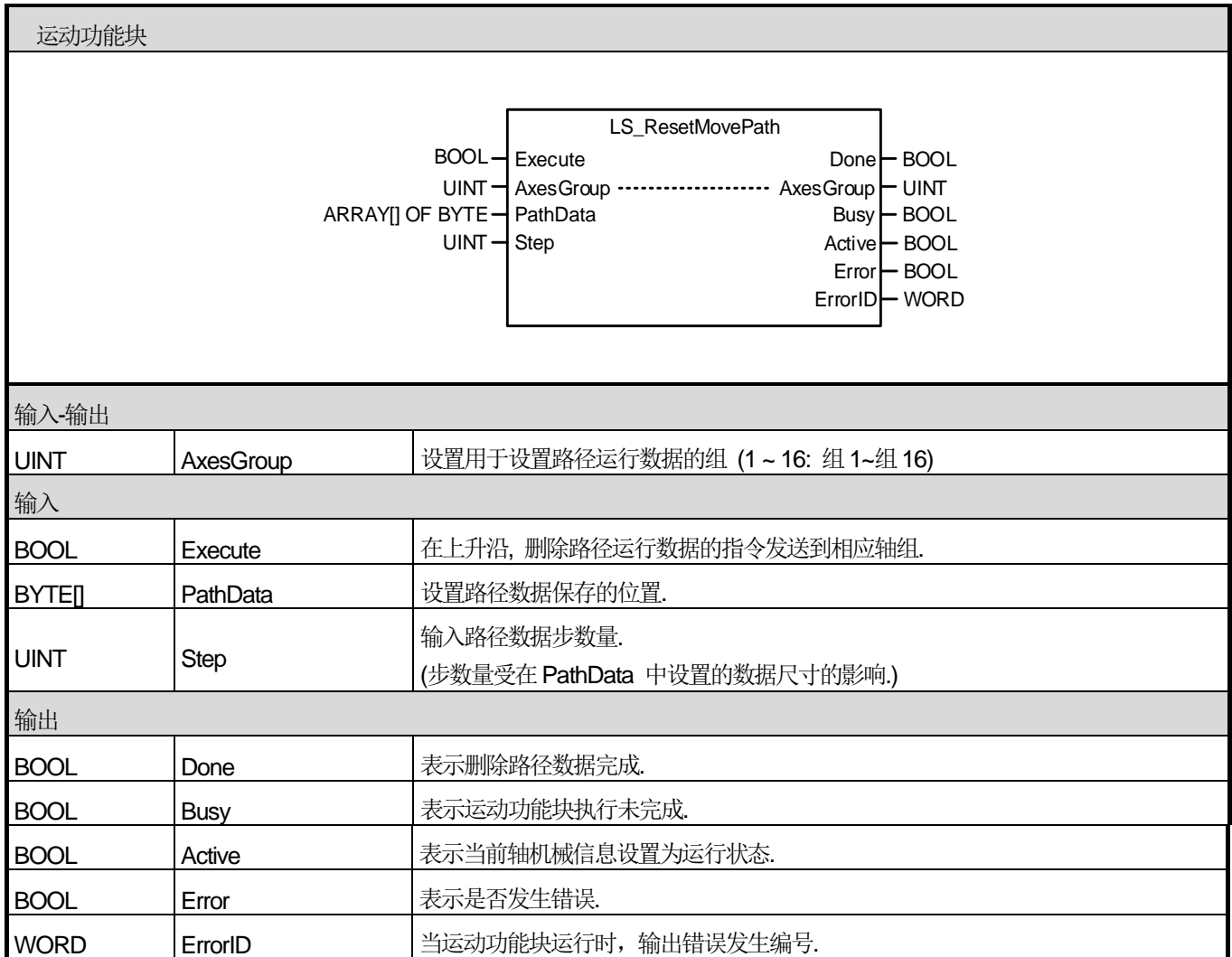
输入

BOOL	Execute	在上升沿, 发送用于设置路径运行数据到相应轴组的指令.
UINT	Step	输入路径数据步数量. (步数量受在路径数据中设置的数据尺寸所影响.)
UINT	CommandType	选择路径运行类型. 0: 无 1: 用于坐标系统绝对位置的线性插值运行 2: 用于坐标系统相对位置的线性插值运行 3: 用于坐标系统绝对位置的圆弧插值运行 4: 用于坐标系统相对位置的圆弧插值运行
UINT	Mode	选择用于坐标系统圆弧插值运行的方法和路径 0/1/2: 顺时针, 中点/中心点/半径 3/4/5: 逆时针, 中点/中心点/半径
UINT	CoordSystem	选择坐标系统类型.(1:MCS 2:PCS)

LREAL[]	Position	输入机械结束点目标位置。 在圆弧插值中，中心点/路径点应设置在位置[3]位置[4]位置[5]。 在圆弧插值中，半径应该在位置[3]。
LREAL	Velocity	指定路径最大速度. [u/s]
LREAL	Acceleration	指定加速. [u/s ²]
LREAL	Deceleration	指定减速. [u/s ²]
LREAL	Jerk	指定加/减速变化率. [u/s ³]
UINT	Direction	指定运行方向。 (0-4: 0-未指定, 1-正向, 2-最短距离, 3-反向, 4-当前方向)
UINT	BufferMode	指定运动功能块的顺序运行设置。 (参考 6.1.4.缓存模式)
UINT	TransitionMode	未使用
UREAL	TransitionParameter	未使用
输出		
BOOL	Done	表示路径数据设置完成。
BOOL	Busy	表示运动功能块执行未完成。
BOOL	Active	表示当前轴机械信息设置为运行状态。
BOOL	Error	表示是否发生错误。
WORD	ErrorID	当运动功能块运行时，输出错误发生编号。

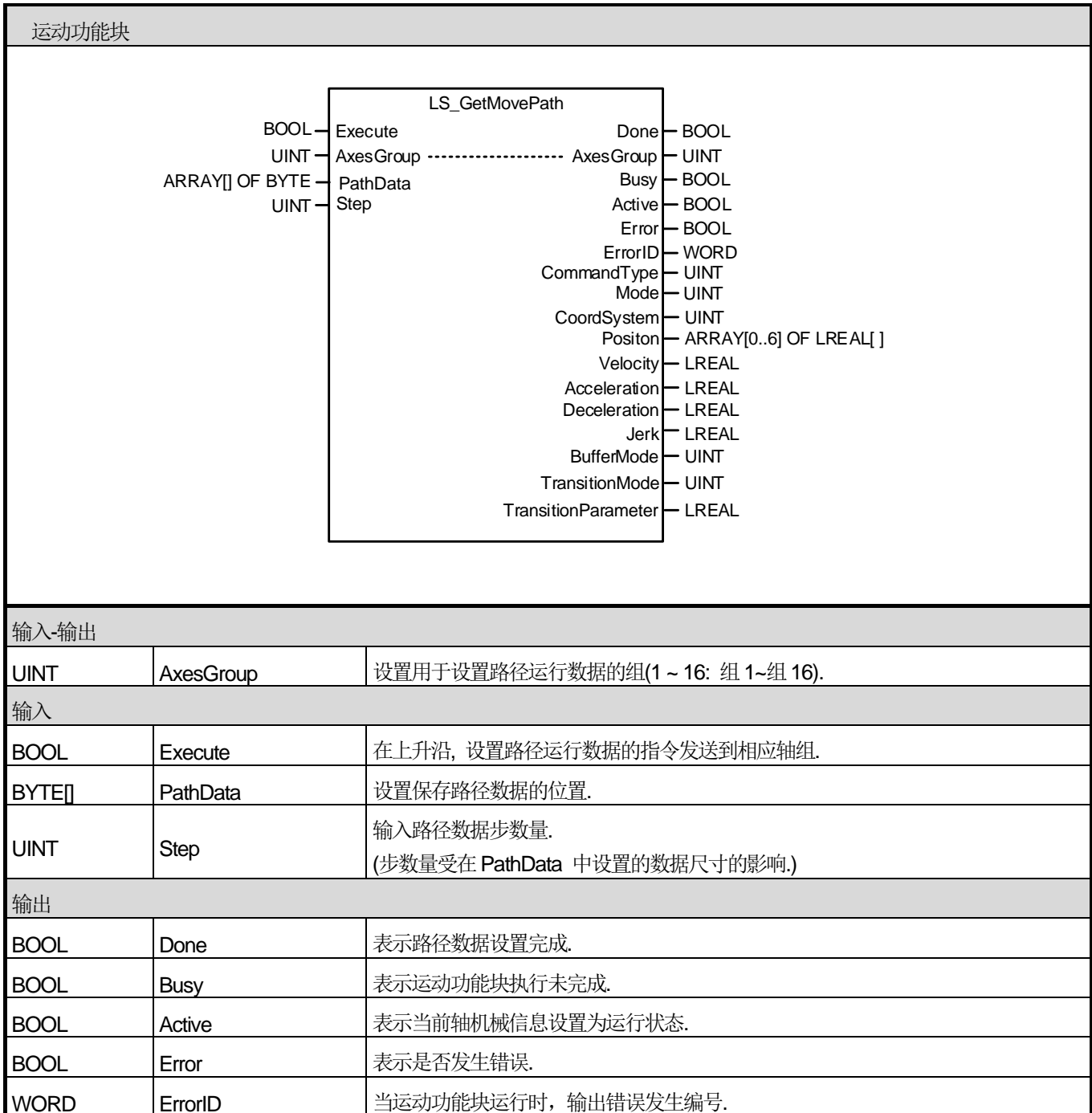
- (1) 该运动功能块为用于 AxesGroup 输入中指定轴组设置路径数据。
- (2) 步值可从0开始设置，每步的尺寸为96 Bytes。
- (3) 路径数据保存在设置在 PathData 的数据区域。设置在 PathData 的变量应设置为 96 次或者多于使用步数量的值。
- (4) CommandType值选择用于路径运行的运行方法。如果CommandType值设置为0，即认为对应于步的数据在路径运行期间不进行设置。
- (5) 当执行圆弧插值运行时，模式值设置圆弧插值方向。
- (6) BufferMode值应设置为1(缓存)。
- (7) 更多内容，参考8.4.11, "坐标系统的路径运行"。

6.7.12 删除路径运行数据 (MC_RestMovePath)



- (1) 该运动功能块删除在 AxesGroup 输入中指定轴组路径数据.
- (2) 步值可以从 0开始设置, 一个步的尺寸为96 Bytes.
- (3) 路径数据保存在设置与 PathData 的数据区域. 设置在 PathData 中的变量应为 96 次或者大于使用步数量的值.
- (4) 更多内容, 参考8.4.11, "坐标系统的路径运行".

6.7.13 读取路径运行数据 (MC_GetMovePath)



第 6 章运动功能块

UINT	CommandType	<p>路径运行输出类型.</p> <p>0: 无</p> <p>1: 用于坐标系统绝对位置的线性插值运行</p> <p>2: 用于坐标系统相对位置的线性插值运行</p> <p>3: 用于坐标系统绝对位置的圆弧插值运行</p> <p>4: 用于坐标系统相对位置的圆弧插值运行</p>
UINT	Mode	输出运行模式.
UINT	CoordSystem	输出坐标系统类型.(1:MCS 2:PCS)
LREAL[]	Position	输出目标位置.
LREAL	Velocity	输出路径最大速度. [u/s]
LREAL	Acceleration	输出最大加速 [u/s ²]
LREAL	Deceleration	输出最大减速 [u/s ²]
LREAL	Jerk	输出加速/减速变化率. [u/s ³]
UINT	BufferMode	<p>指定运动功能块的顺序运行设置.</p> <p>(参考 6.1.4.缓存模式)</p>
UINT	TransitionMode	未使用
LREAL	TransitionParameter	未使用

- (1) 该运动功能块读取 AxesGroup 输入中指定轴组路径数据.
- (2) 步值可以从 0 开始设置, 步尺寸为 96 Bytes.
- (3) 路径数据保存在设置在 PathData 中的数据区域. 设置在 PathData 中的变量应为 96 次或者大于使用步数量的值.
- (4) 更多内容请参考8.4.11, "坐标系统路径运行 ".

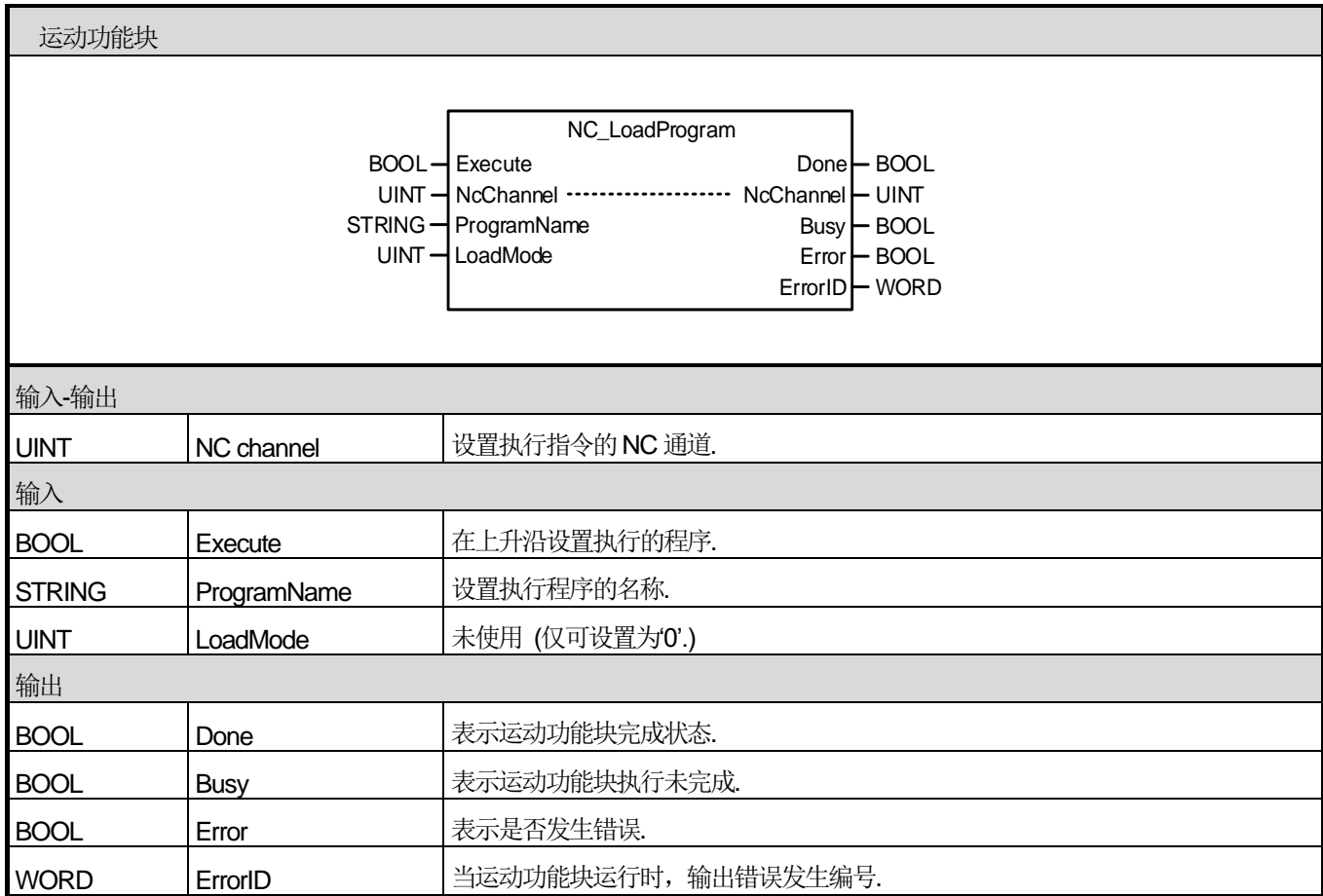
6.7.14 执行路径运行 (MC_RunMovePath)

运动功能块		
输入-输出		
UINT	AxesGroup	设置执行路径运行数据的组。 (1 ~ 16 : 组 1~组 16)
输入		
BOOL	Execute	在上升沿, 设置路径运行数据的指令发送到相应轴组.
BYTE[]	PathData	设置保存路径数据的位置.
UINT	StartStep	输入路径数据起始步数量. (步数量受在 PathData 中设置的数据尺寸的影响.)
UINT	EndStep	输入路径数据结束步数量. (步数量受在 PathData 中设置的数据尺寸的影响.)
输出		
BOOL	Done	表示路径数据设置完成.
BOOL	Busy	表示运动功能块执行未完成.
BOOL	Active	表示当前轴机械信息设置为运行状态.
BOOL	CommandAborted	表示当前运动功能块在运行时被中断.
BOOL	Error	表示是否发生错误.
WORD	ErrorID	当运动功能块运行时, 输出错误发生编号.
UINT	CurStep	输出当前运行步数量.

- (1) 该运动功能块执行在 AxesGroup 输入中指定轴组路径运行
- (2) 步值可以从 0 开始, 步尺寸为 96 Bytes.
- (3) 路径数据保存在设置在 PathData 中的数据区域. 设置在 PathData 中的变量应为 96 次或者大于使用步数量的值.
- (4) StartStep 和 EndStep 之间的不同不可以设置为 100 及以上. (一次最多可以执行 100 步)
- (5) 路径运行期间如果路径数据的 CommandType 为 0, 即使 EndStep 未到达, 运行终止.
- (6) 如果执行路径运行, 运行中当前步数量输出到 CurStep.
- (7) 更多内容请参考 8.4.11, "坐标系统路径运行".

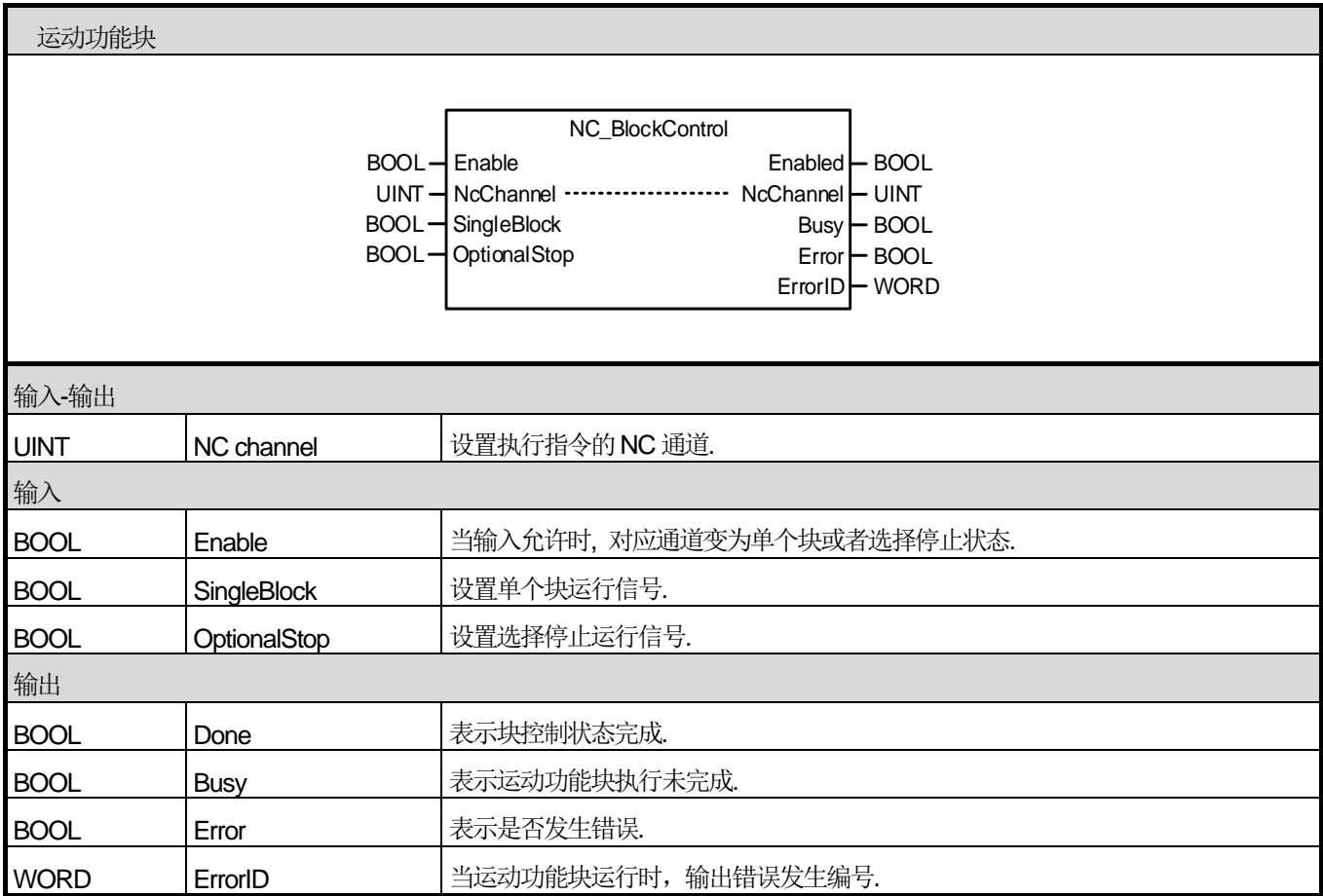
6.8 NC 控制功能块

6.8.1 指定 NC 程序 (NC_LoadProgram)



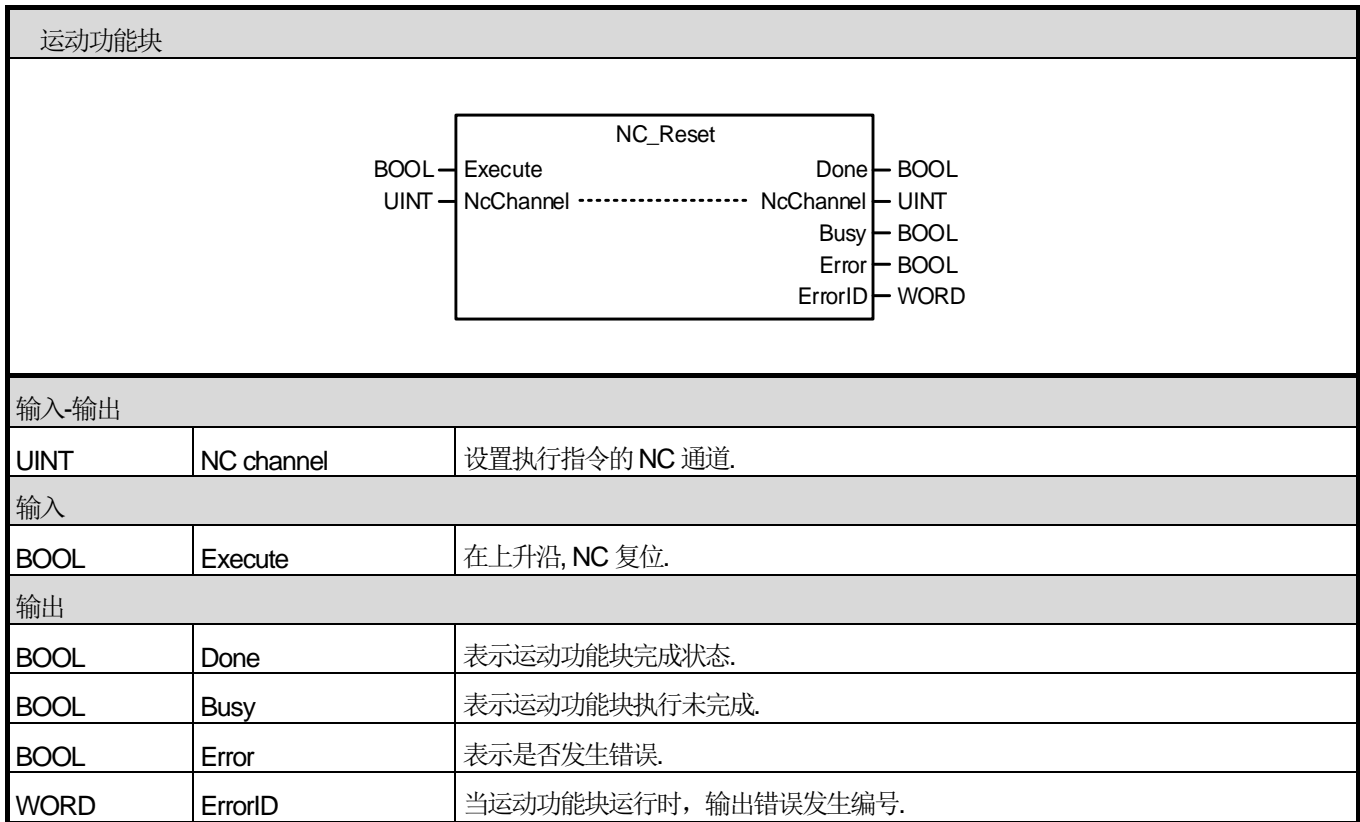
- (1) 该运动功能块当执行 NC 控制时, 指定执行 NC 程序.
- (2) 当通过在 NC 通道中设置通道的执行程序被设置到 ProgramName 时, 并执行功能块, 执行指定的程序.

6.8.2 指定块运行(NC_BlockControl)



- (1) 该运动功能块决定在 NC 控制下, 执行程序的方式.
- (2) 如果 SingleBlock 设置为'1', NC_CycleStart 一次执行一个块, 并且执行后停止. 如果自动运行期间 SingleBlock 变为'1', 执行 NC_BlockControl 功能块, 当前执行块终止后停止.
- (3) 如果 OptionalStop 设置为'1', 程序运行期间执行 M01, 将保持等待状态直到 NC_CycleStart 功能块再次执行.
- (4) 当 SingleBlock 和 OptionalStop 同时设置为'1'时, 应用 SingleBlock 设置.

6.8.3 复位(NC_Reset)

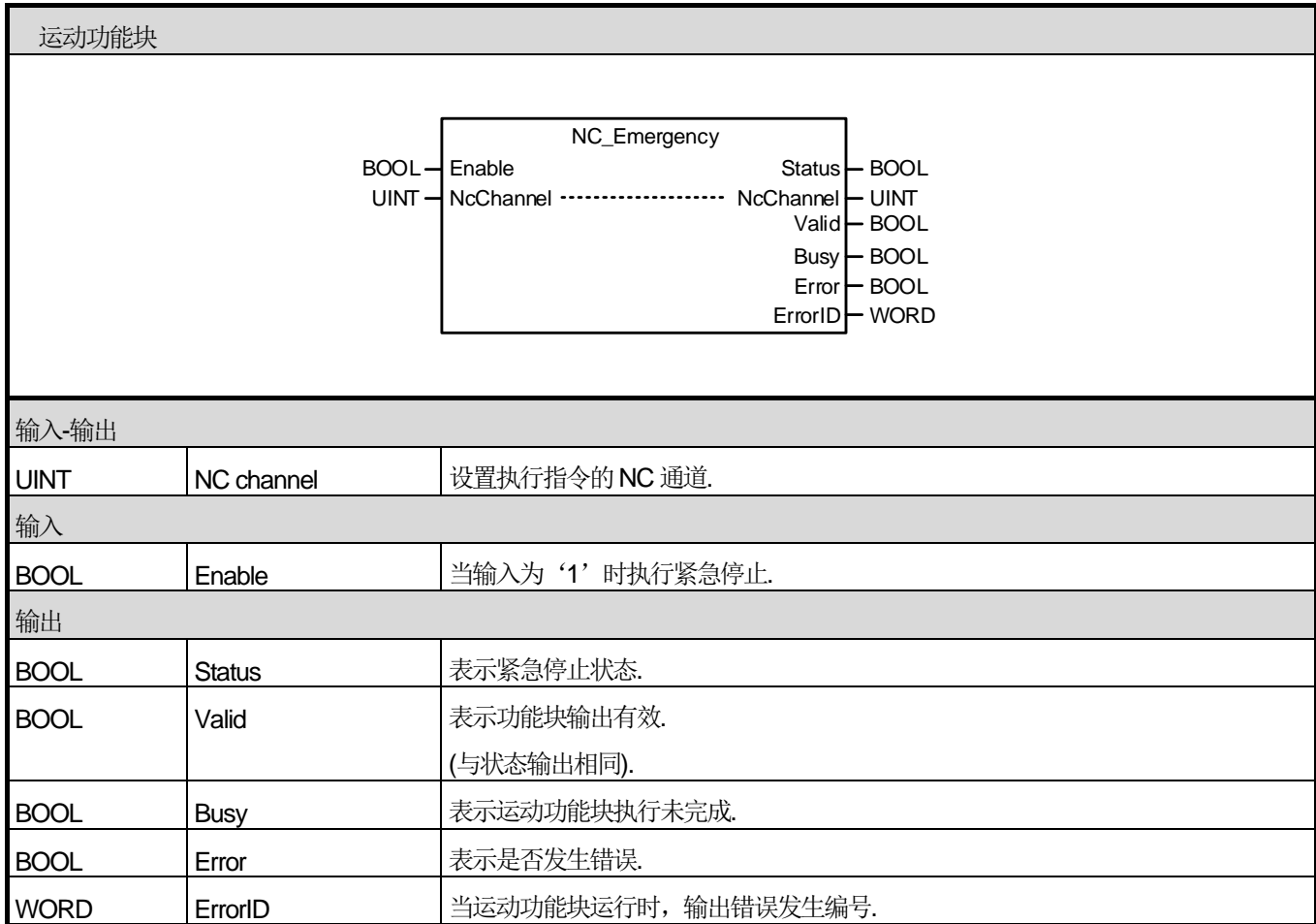


- (1) 该运动功能块在 NC 控制下执行 NC 复位状态.
- (2) 如果在自动运行期间执行 NC_Reset, 停止自动运行, 并变更为复位状态.
- (3) 复位状态如下.

	内容	状态
设置数据	补偿值	保持
	参数	保持
多种数据	内存中程序	保持
	缓存存储器中的内容	MDI: 保持 其他取消
	序号显示	保持
	单次执行 G 代码	取消
	模式 G 代码	保持
	F	保持
	S, T, M	保持
工作坐标值	K(重复数量)	取消
		保持

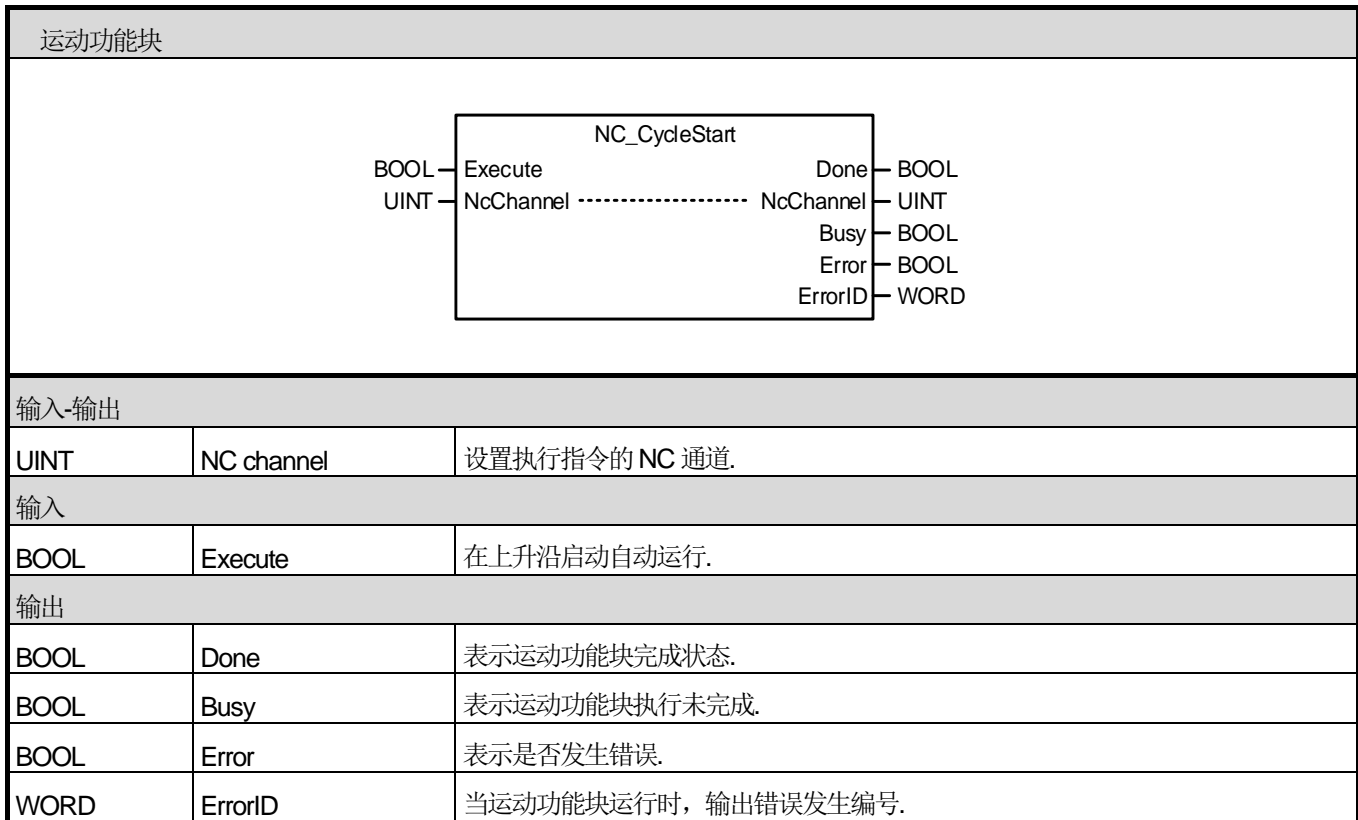
内容	状态	
运行动作	移动	取消
	停留	取消
	发布 M, S, T 代码	取消
	工具长度补偿	MDI: 保持 其他取消
	刀具补偿	MDI: 保持 其他取消
	存储调用子程序数量	MDI: 保持 其他取消
输出信号	CNC 警报信号 AL	如果没有报警的原因则熄灭
	参考位置返回完成 LED	保持 取消(紧急停止)
	S, T, B 代码	保持
	M 代码	取消
	M, S, T 熄灭信号	取消
	主轴旋转信号(S 模拟量信号)	保持
	CNC 准备信号 MA	保持
	伺服准备信号 SA	ON
	循环启动 LED	取消
	进给保持 LED	取消

6.8.4 紧急停止 (NC_Emergency)



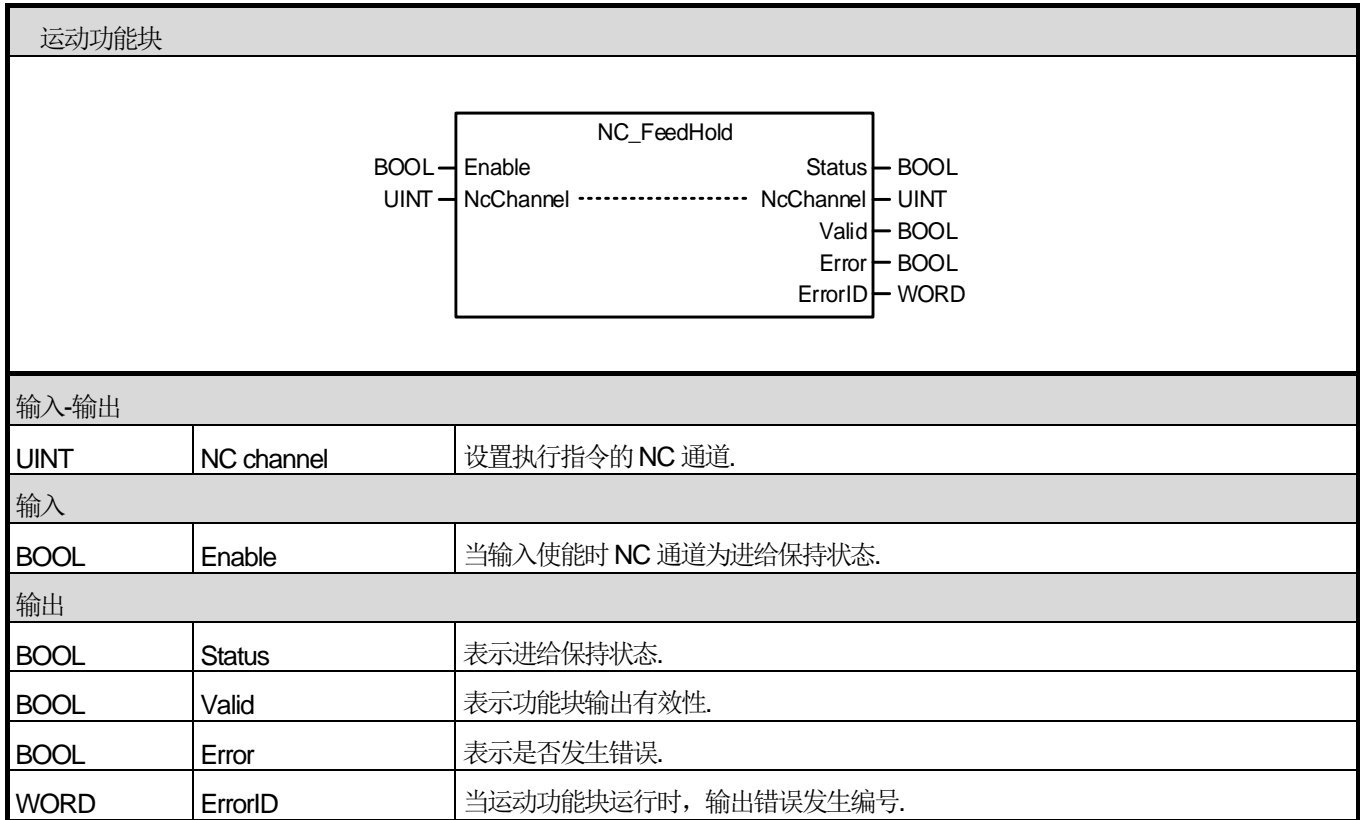
- (1) 该运动功能块在 NC 控制下，执行对应于 NC 通道的紧急停止.
- (2) 如果执行紧急停止，当前运行必须立即停止.

6.8.5 启动自动运行(NC_CycleStart)



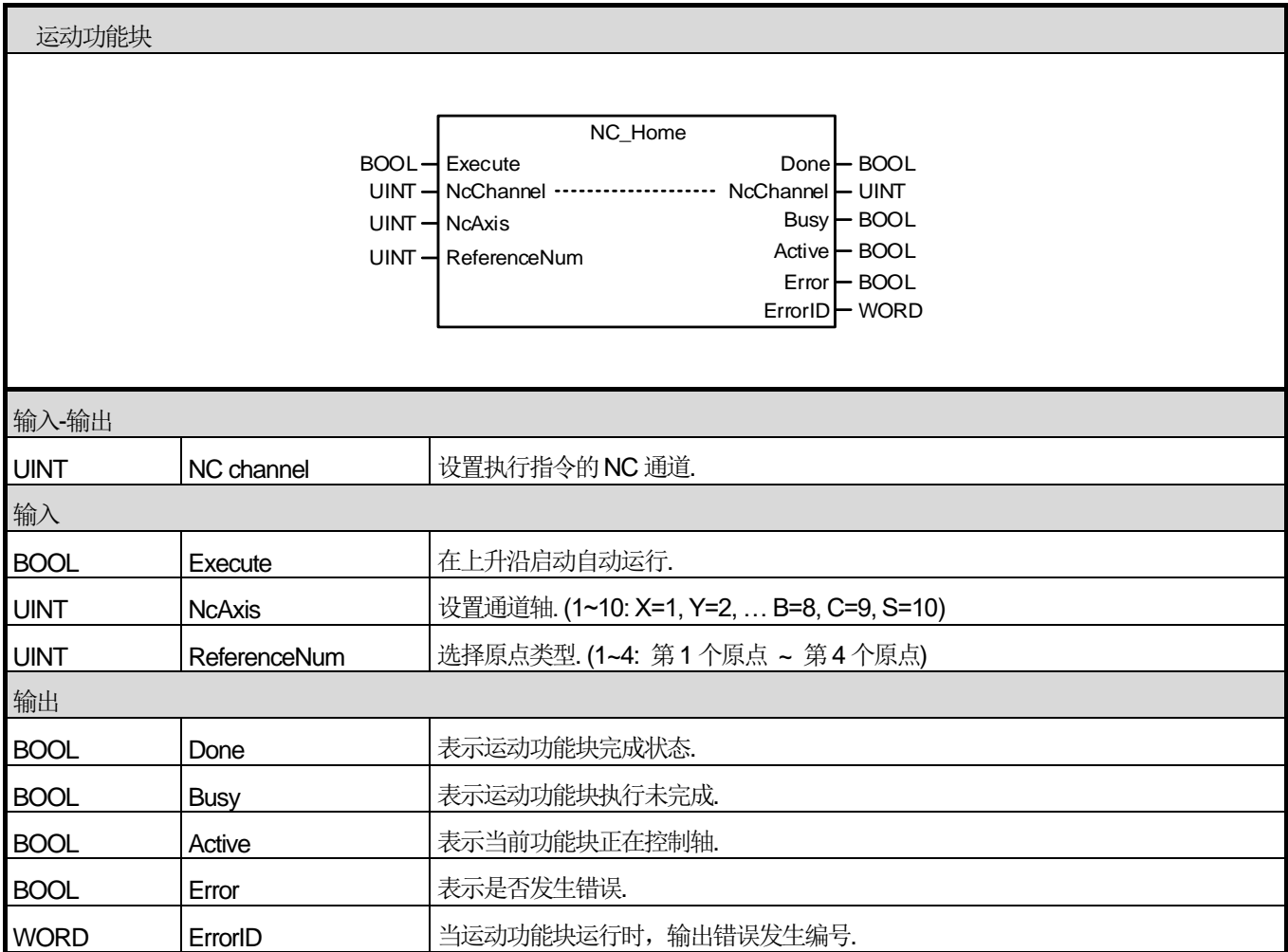
- (1) 该运动功能块在 NC 控制下执行对应于 NC 通道的自动运行.
- (2) 在 NC_LoadProgram 中设置的程序自动运行.
- (3) 当由于 M00, M01(选择停止)和单个块导致自动运行停止时, 自动运行重启.

6.8.6 进给保持 (NC_FeedHold)



- (1) 该运动功能块在 NC 控制下，发送进给保持指令到相应 NC 通道.
- (2) 如果自动运行期间执行 NC_FeedHold, 自动运行停止.
- (3) 如果在 NC_FeedHold 指令期间执行 NC_CycleStart, 忽略 NC_CycleStart.

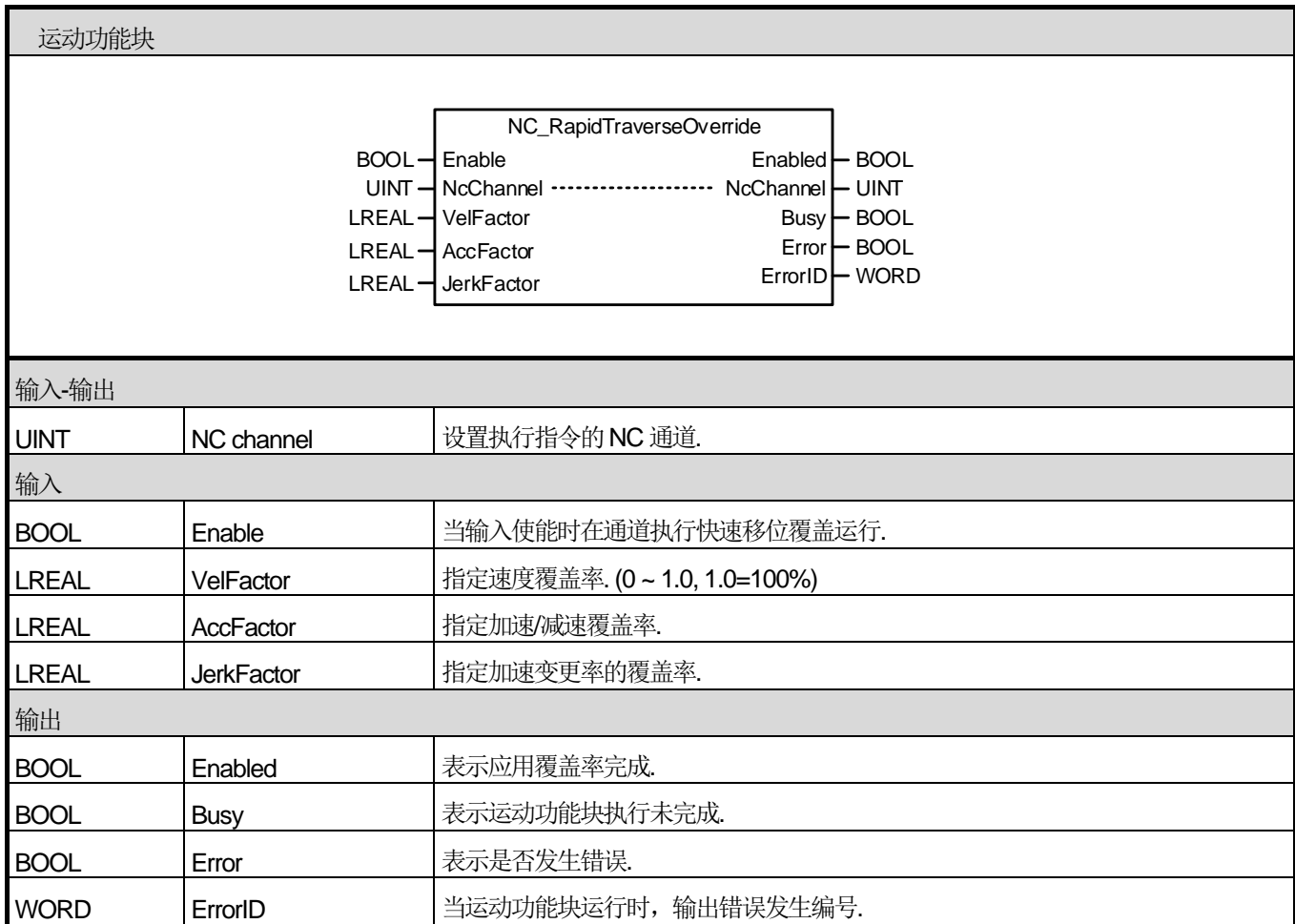
6.8.7 原点返回(NC_Home)



- (1) 该运动功能块在 NC 控制下执行原点返回到相应 NC 通道.
- (2) 根据在 ReferenceNum 中设置的值执行原点返回到第 1 个原点, 第 2 个原点, 第 3 个原点, 第 4 个原点. 可以在 MP500 中设置 NC 参数每个轴参数的原点坐标.

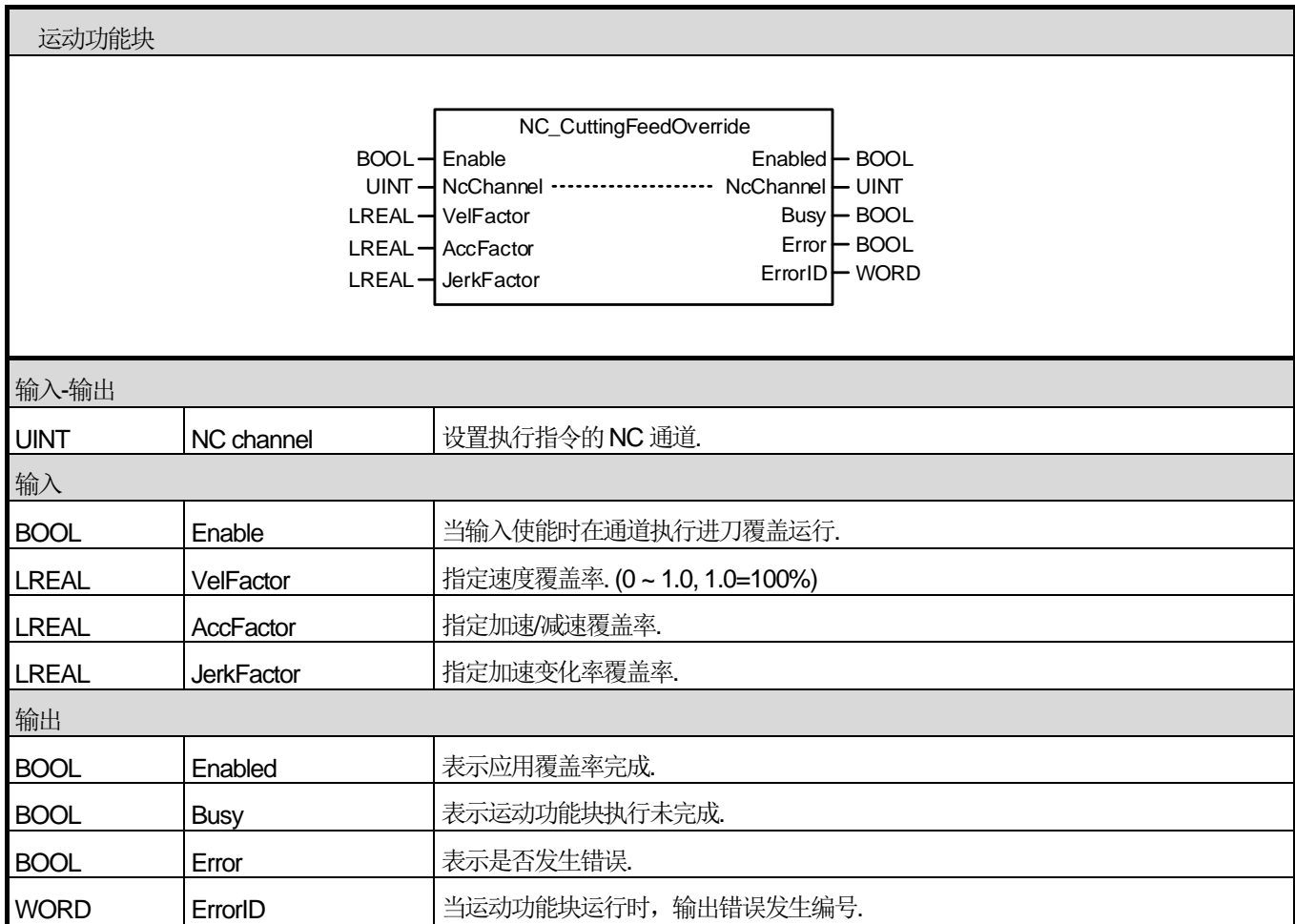
Group	Name	X Axis
Axis Settings	Command direction for the Modular Axis	0: Bidirectional
Home Settings	Position of 2nd home	0 mm
	Position of 3rd home	0 mm
	Position of 4rd home	0 mm

6.8.8 快速移位覆盖 (NC_RapidTraverseOverride)



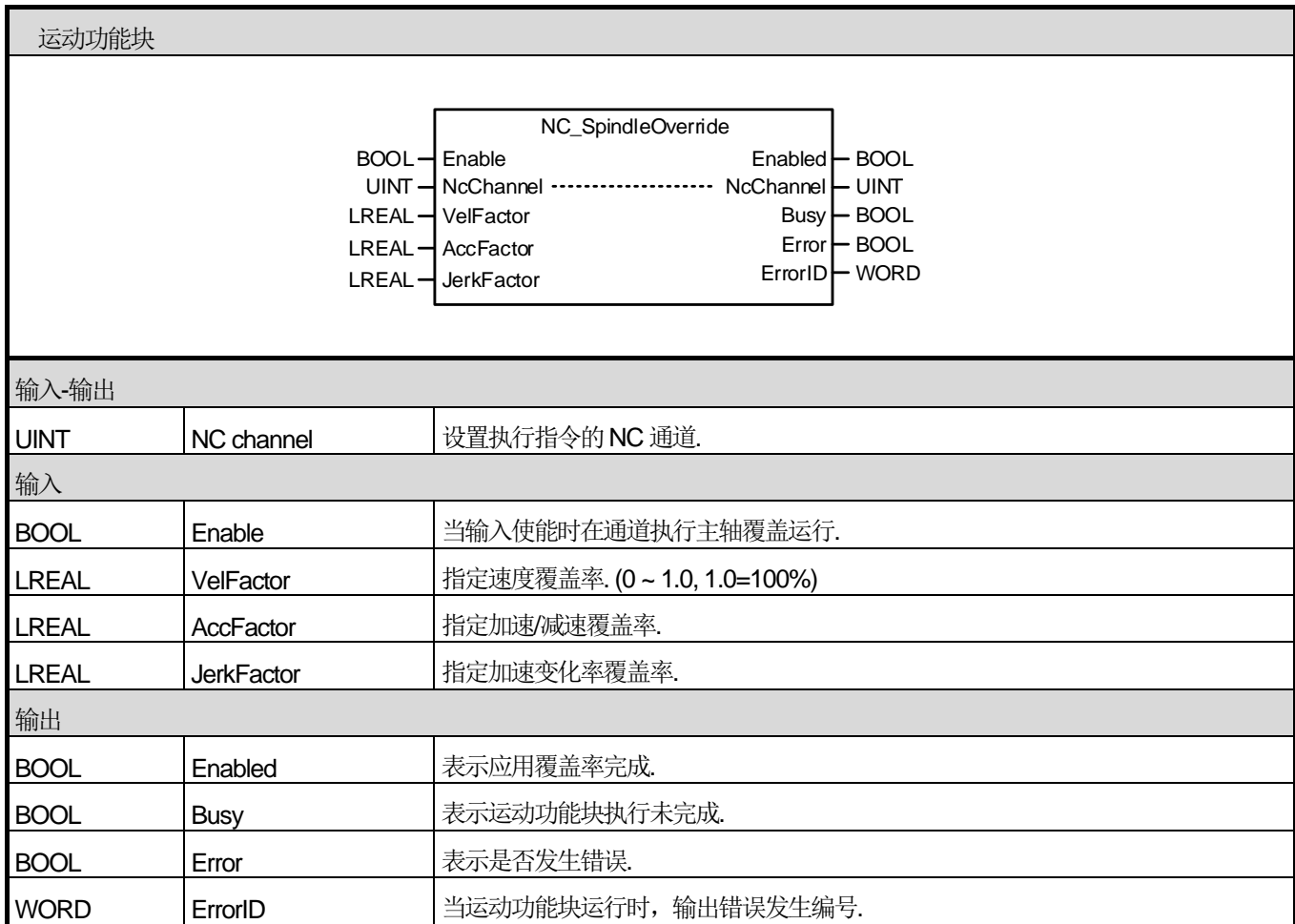
- (1) 该运动功能块在 NC 控制下执行对应 NC 通道的快速移位覆盖指令.
- (2) 指定 VelFactor 输入的速度覆盖率. 如果指定值为 0.0, 轴停止.
- (3) 每个要素的默认值为 1.0, 表示当前执行功能块指令速度 100%.
- (4) 分别指定 AccFactor 输入的加速/减速和 JerkFactor 输入加速度 (加速变化率)覆盖率.
- (5) 每个要素无法输入负数.

6.8.9 进刀覆盖 (NC_CuttingFeedOverride)



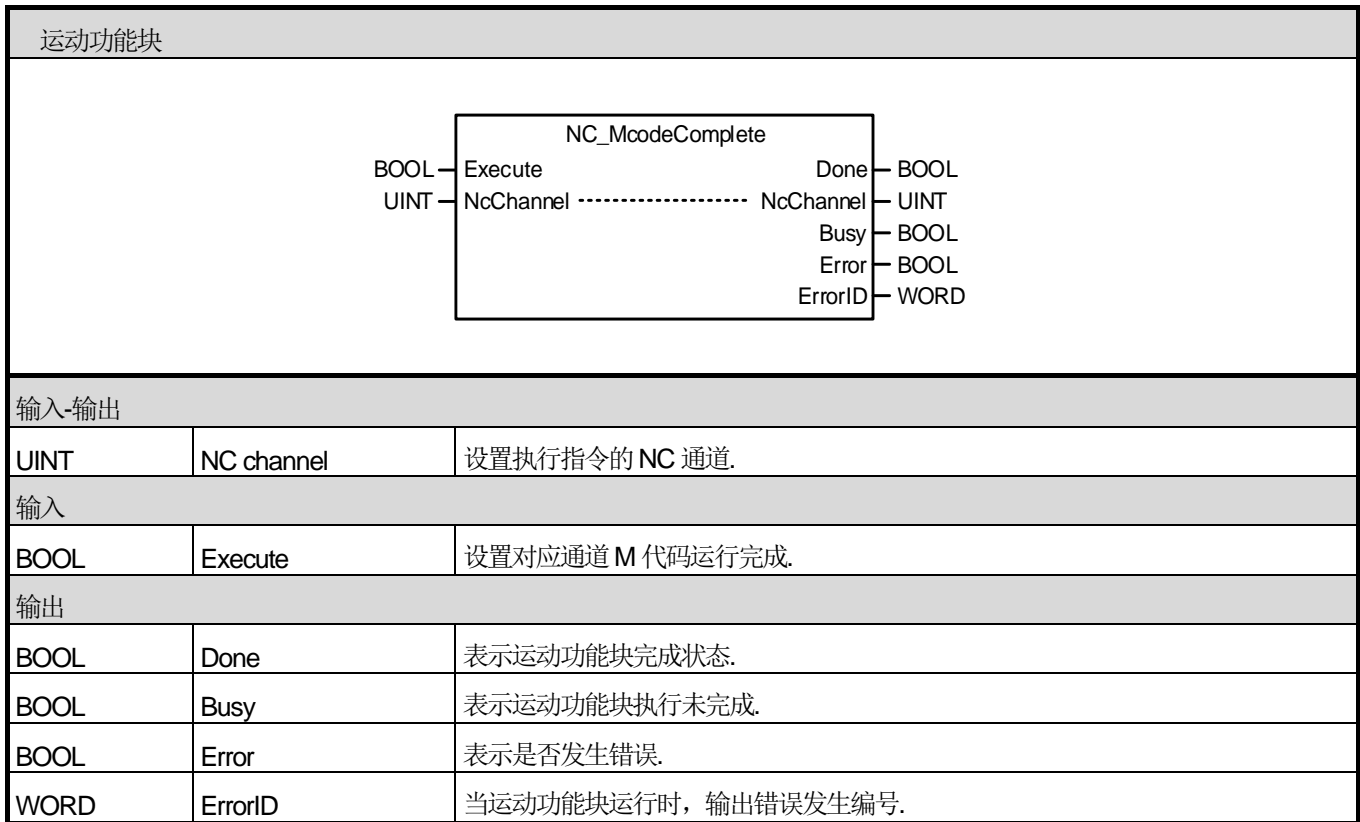
- (1) 该运动功能块在 NC 控制下执行对应 NC 通道的进刀覆盖指令.
- (2) 指定 VelFactor 输入的速度覆盖率. 如果指定值为 0.0, 轴停止.
- (3) 每个要素的默认值为 1.0, 表示当前执行功能块指令速度 100%.
- (4) 分别指定 AccFactor 输入的加速/减速和 JerkFactor 输入加速度 (加速变化率)覆盖率.
- (5) 每个要素无法输入负数.

6.8.10 主轴覆盖 (NC_SpindleOverride)



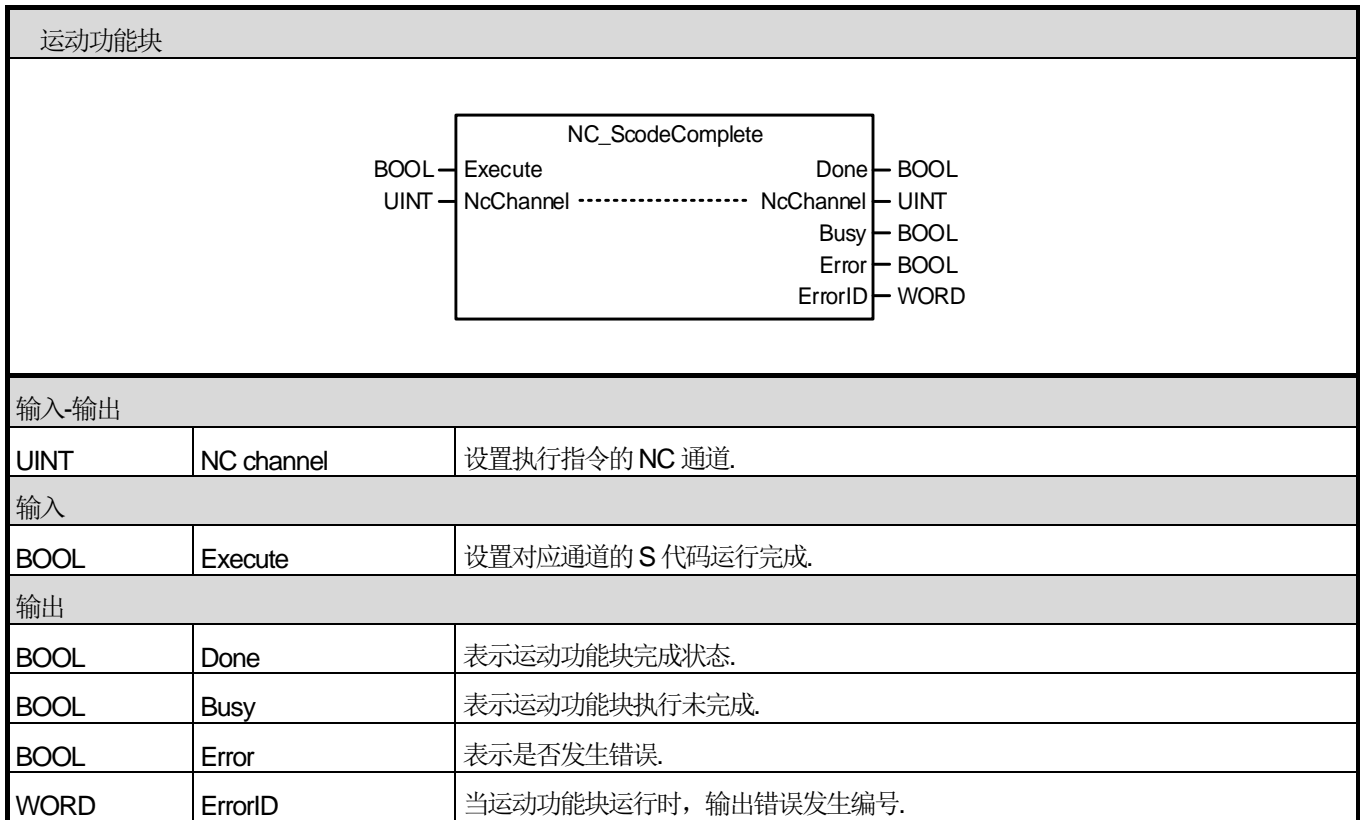
- (1) 该运动功能块在 NC 控制下执行对应 NC 通道的主轴覆盖指令.
- (2) 指定 VelFactor 输入的速度覆盖率. 如果指定值为 0.0, 轴停止.
- (3) 每个要素的默认值为 1.0, 表示当前执行功能块指令速度 100%.
- (4) 分别指定 AccFactor 输入的加速/减速和 JerkFactor 输入加速度 (加速变化率)覆盖率.
- (5) 每个要素无法输入负数.

6.8.11M 代码运行完成 (NC_McodeComplete)



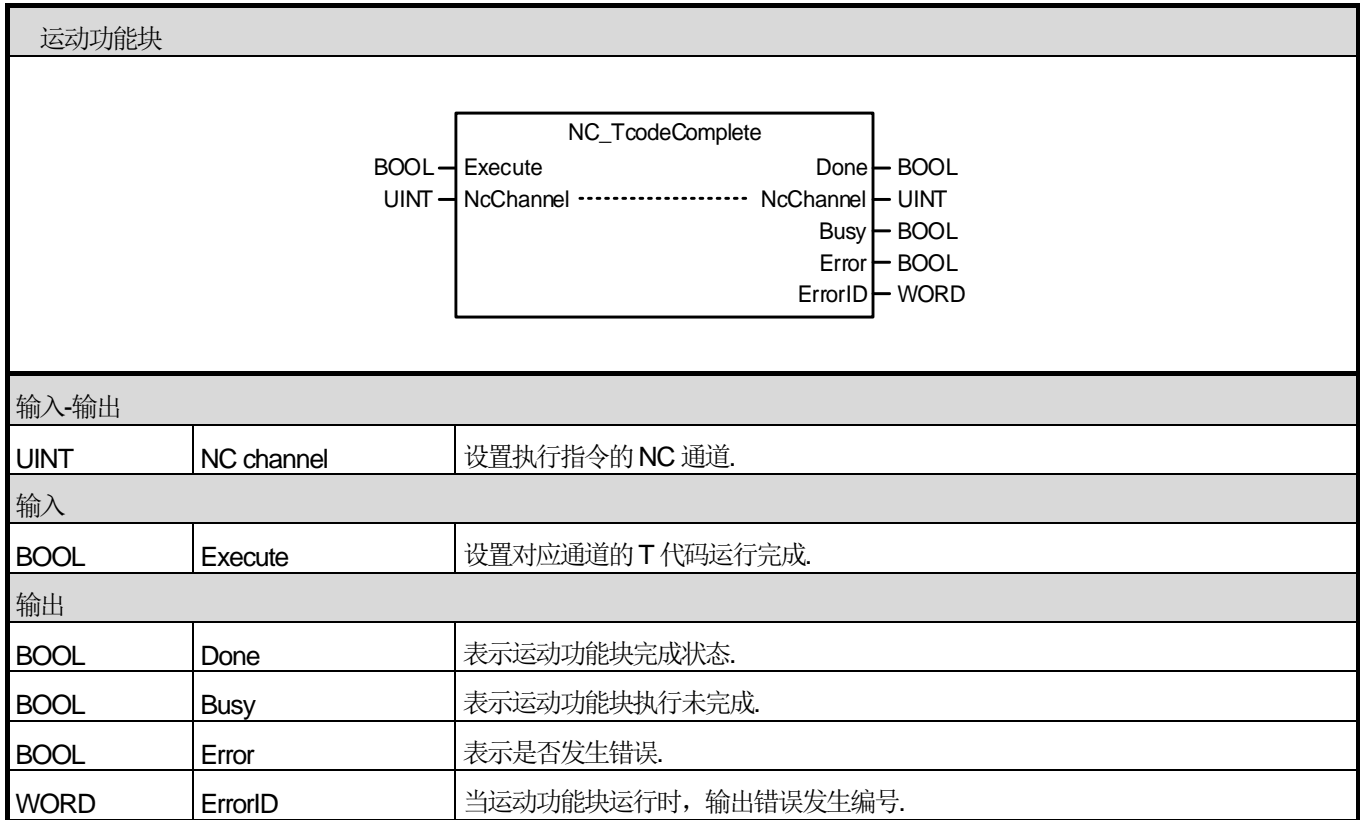
- (1) 该运动功能块在 NC 控制下执行对应 NC 通道的 M 代码运行完成指令.
- (2) 该指令为查看相应通道 M 代码的指令，并且设置 M 代码完成.

6.8.12S 代码运行完成 (NC_ScodeComplete)



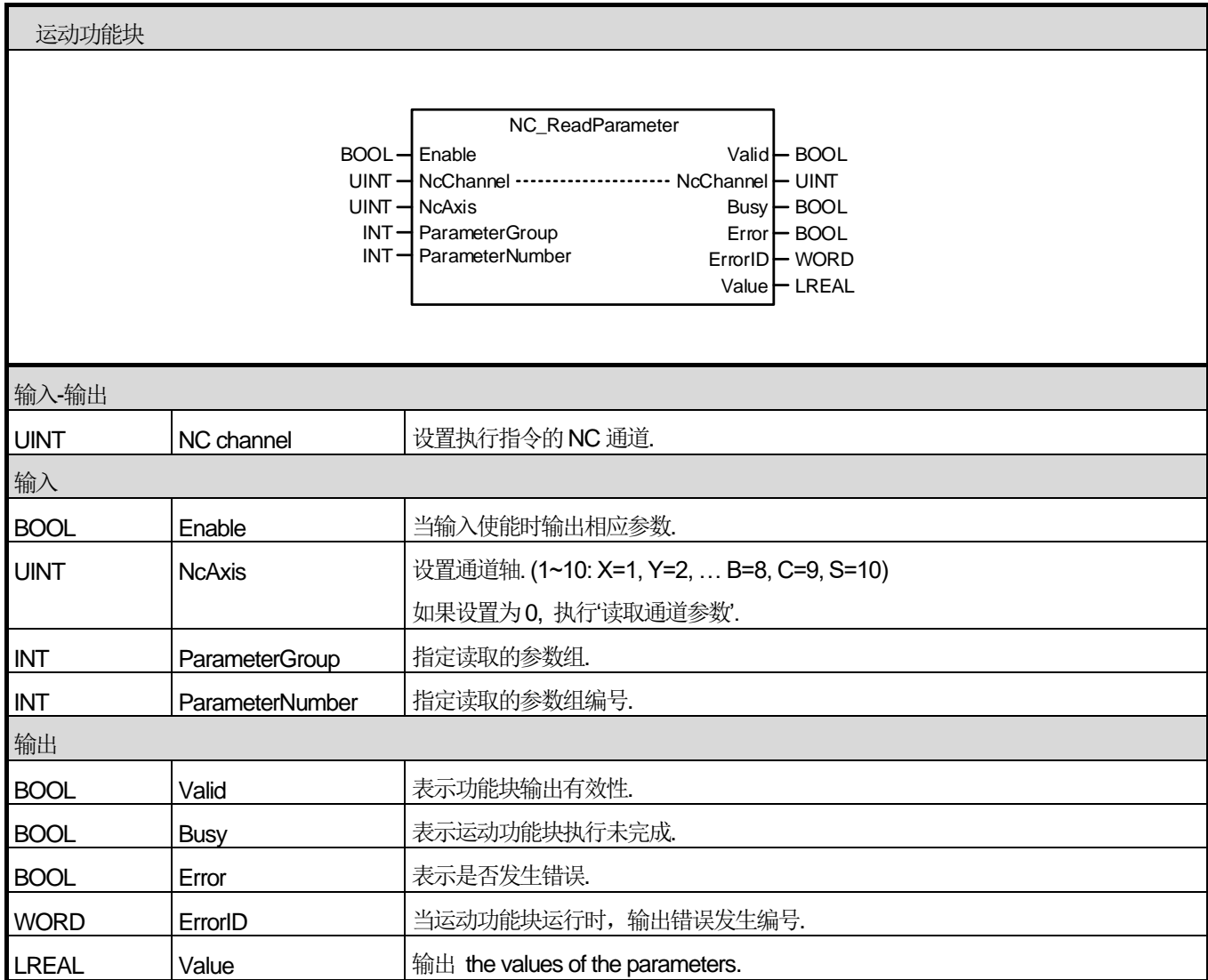
- (1) 该运动功能块在 NC 控制下执行对应 NC 通道的 S 代码运行完成指令.
- (2) 该指令为查看相应通道 S 代码的指令，并且设置 S 代码完成.

6.8.13T 代码运行完成 (NC_TcodeComplete)



- (1) 该运动功能块在 NC 控制下执行对应 NC 通道的 T 代码运行完成指令.
- (2) 该指令为查看相应通道 T 代码的指令，并且设置 T 代码完成.

6.8.14 读取 NC 参数(NC_ReadParameter)



- (1) 该运动功能块读取和输出对应通道的通道和通道/轴参数.
- (2) 当使能输入激活时, 相应参数值持续输出.
- (3) ParameterGroup 输入指定读取的参数组.
- (4) ParameterNumber 输入指定读取的参数组编号.

(5) 组编号和每个参数组数量如下.

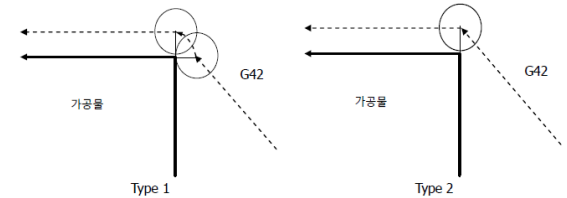
参数	组	No.	项目	描述
1. 通道参数	1. 基本设置	1	目标机械数量	设置目标机械数量. (0 ~ 2,147,483,647)
		2	在 M99 重复机械中目标机械数量	在 M99 重复机械中设置目标机械数量.如果设置值与当前机械数量匹配, 循环自动停止. (0 ~ 2,147,483,647)
		3	小数点查看	设置是否查看 NC 程序小数点. 0: 小数点查看 (如果有小数点则为 Mm, 如果没有小数点则为 um) 1: 不查看小数点(mm)
		4	保持工件坐标系统	当复位时设置是否保持工件坐标系统. 0: 保持 1: 不保持
		5	当执行 T 代码时是否调用宏	当执行 T 代码时设置是否调用宏程序 (9000.nc ~ 9009.nc). 0: 不调用 1: 调用
		6	停留方式	设置使用对应 X,P,作为时间或者主轴转数数据的停留功能(G04)X, P. 如果数据设置主轴转数, 应用于每转进刀状态 (G95). 0: 时间 1: 转数
		7	选择复位进程块	设置在复位状态下是否初始化程序启动块. ※如果设置为 0 (保持当前块), "保持工件坐标系统" 参数设置为 0 (保持). 0: 保持当前块 1: 初始化主程序启动块 2: 初始化主程序当前块

第 6 章运动功能块

参数	组	No.	项目	描述
1. 通道参数	1. 基本设置	8	是否搜索语句数量	<p>可以保存程序语句数量(N__)的缓存数量在系统中的限制范围是 1,000.</p> <p>如果使用 GOTO 语句改变程序顺序, 则需要该缓存.</p> <p>如果大于 1,000 个块具有 N__ 指令, 将发生警告. 该参数用于输入是否执行例如语句数量搜索.</p> <p>由于大容量 CAM 程序不具有使用 Statement Number 的 GOTO, 大多数情况下, 有大于 1,000 的语句数量, 应设置该参数为 1.</p> <p>0: 搜索 1: 不搜索</p>
		12	最小指令单位	当应用查看小数点时, 设置指令值最小单位. (0 ~ 0.999mm)
		18	是否使用 G22 无行程区域	0: '无行程区域'有效. 1: '无行程区域'无效.
		19	设置 G22 无行程区域内侧/外侧	0: 内侧 1: 外侧
		20	是否使用第 3 个'无行程区域'	0: '无行程区域'有效. 1: '无行程区域'无效.
		22	圆柱插值旋转轴	<p>在圆柱插值模式中, 在圆弧插值运行中轴映射旋转轴.轴为 X, Y, Z, 并通过映射旋转轴到选择轴执行圆弧插值.</p> <p>例如, 如果在 XY(G17)平面状态下旋转轴映射到 X 轴, 宽度变为旋转轴, 高度变为 Y 轴. 当选择 Z X (G18)作为平面时, 宽度变为 Z 轴, 高度变为旋转轴. 然而, 如果设置平面为 YZ (G19), 无法在旋转指令轴执行圆弧插值.</p> <p>0: X-轴, 1: Y-轴, 2: Z-轴,</p>
		23	用于两极坐标插值的线性轴	0: 未使用 1: X, 2: Y, 3: Z, 4: A, 5: B, 6: C, 7: U, 8: V, 9: W
		24	用于两极坐标插值的旋转轴	0: 未使用 1: X, 2: Y, 3: Z, 4: A, 5: B, 6: C, 7: U, 8: V, 9: W

参数	组	No.	项目	描述
1.通道参数	1. 基本设置	33	到达位置完成监控时间	0 ~ 65,535ms
	2. 圆周铣削设置	1	当圆弧报警发生时更新圆弧中心	设置当起始点和结束点之间的距离超出 I, J, K 指令下的两个不同半径间的公差时, 是否在不产生圆弧警报的情况下重新创建圆弧中心点. 0: 发生报警. 1: 重新生成圆弧中心点.
		2	关于循环铣 ON/OFF 的速度限制功能	0: 未使用 1: 使用
		3	圆弧半径公差	设置在圆弧指令下的启动点和结束点之间两个不同半径的公差.如果该值过大, 圆弧结束部分精度可能降低.当设置为 0 时, 则视为 0.001. (0~ 1 单元, 实数)
		5	用于圆弧加工的具有速度限制功能的圆弧半径	(0 ~ 10,000 单元, 实数)
		6	圆周铣削的向上切割速度限制	最大速度受限于如下圆弧的设置值"用于圆周铣削具有速度限制功能的圆弧半径" . (0 ~ 10,000 unit/min, 实数)
		7	圆周铣削的向下切割速度限制	如果"用于圆周铣削 ON/OFF 的速度限制功能" 设置为 ON, 切割速度限制于设置值及以上值. (0 ~ 10,000 unit/min, 实数)
		9	圆周铣削加速	在圆周铣削中设置加速.
		10	圆周铣削减速	在圆周铣削中设置减速.
		11	圆周铣削加速度	在圆周铣削中设置加速度.
	3. 进刀设置	1	设置进刀设置的上限速度	如果切割速度超出设置值, 切割速度保持设置值并发生警报. (0 ~ 100,000 unit/min, 实数)
		2	设置进刀设置的下限速度	仅应用于当在每分钟馈送模式下切割速度不执行时. (0 ~ 100,000 unit/min, 实数)
		4	插值运行的加速/减速方式	1: 插值前加速/减速

第 6 章运动功能块

参数	组	No.	项目	描述
1. 通道参数	3. 进刀设置	7	插值前用于加速/减速连续块的运行方式	当执行连续块时,创建一个以设置与下一个块的速度在连接轨迹拐角绘制的连接轨迹.1: 当设置到缓存时,圆弧不插入. 1: 缓存 2: 混合低 3: 混合前一个 4: 混合下一个 5: 混合高
		9	进刀加速 (插值前)	在进刀时加速
		10	进刀减速 (插值前)	在进刀时减速
		11	进刀加速度 (插值前)	在进刀时加速度
		8.工具直径校正	129	如果应用工具直径校正
	130		工具直径校正类型	在校正开始和结束工具直径设置传输方式类型.  0: 类型 1(支路穿过) 1: 类型 2(直接穿过)
	131		在工具直径校正时是否查看工具干扰	设置在工具直径校正期间是否查看工具干扰 0: 不查看 1: 查看
	1		工具直径 1 校正数量	用于校正工具直径的校正数量 1

	128		工具直径 128 校正数量	用于校正工具直径的校正数量 128

参数	组	No.	项目	描述
1. 通道参数	9. 工具长度校正	1	工具长度校正数量 1	用于校正工具长度的校正数量 1
	
		128	工具长度校正数量 128	用于校正工具长度的校正数量 128
	10. 工件坐标系统	1	是否使用工件坐标系统移位数量	设置是否使用工件坐标系统移位数量. 0: 未使用 1: 使用
		11	工件坐标系统移位数量 1	设置 X 轴工件坐标系统移位数量.
		设置 7 轴: Y, Z, A, B, C, U, V 工件坐标系统移位数量.
		19	工件坐标系统移位数量 9	设置 W 轴工件坐标系统移位数量.
		41	G54 工件坐标系统值 1	设置 X 轴工件坐标系统值.
		设置 7 轴: Y, Z, A, B, C, U, V 工件坐标系统值.
		49	G54 工件坐标系统值 9	设置 W 轴 G54 工件坐标系统值.
		51	G55 工件坐标系统值 1	设置 X 轴 G55 工件坐标系统值.
		设置 7 轴 Y, Z, A, B, C, U, VG55 工件坐标系统值.
		59	G55 工件坐标系统值 9	设置 W 轴 G55 工件坐标系统值.
		61	G56 工件坐标系统值 1	设置 X 轴 G56 工件坐标系统值.
		设置 7 轴 Y, Z, A, B, C, U, VG56 工件坐标系统值
		69	G56 工件坐标系统值 9	设置 W 轴 G56 工件坐标系统值.
		71	G57 工件坐标系统值 1	设置 X 轴 G57 工件坐标系统值.
		设置 7 轴 Y, Z, A, B, C, U, VG57 工件坐标系统值
		79	G57 工件坐标系统值 9	设置 W 轴 G57 工件坐标系统值.
		81	G58 工件坐标系统值 1	设置 X 轴 G58 工件坐标系统值.
		设置 7 轴 Y, Z, A, B, C, U, VG58 工件坐标系统值
		89	G58 工件坐标系统值 9	设置 W 轴 G58 工件坐标系统值.

第 6 章运动功能块

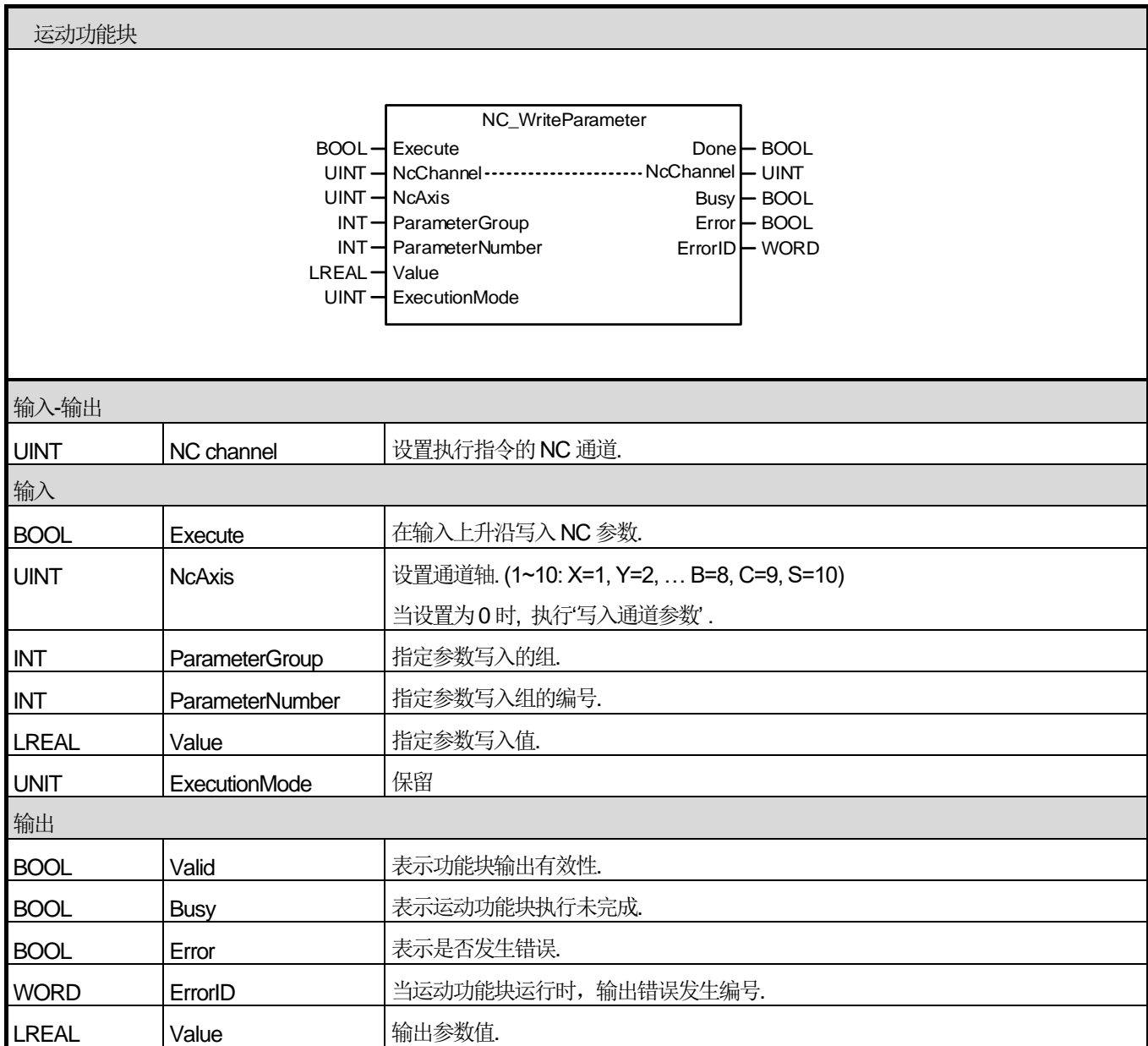
参数	组	No.	项目	描述	
1. 通道参数	10. 工件坐标系统	91	G59 工件坐标系统值 1	设置 X 轴 G59 工件坐标系统值.	
		设置 7 轴 Y, Z, A, B, C, U, VG59 工件坐标系统值	
		99	G59 工件坐标系统值 9	设置 W 轴 G59 工件坐标系统值.	
	11. 宏程序		1	是否应用单个块停止功能到宏程序中	设置是否应用单个块停止功能到宏程序中(9000.nc ~ 9999.nc) 0: 停止 1: 不停止
			2	显示宏程序块	当运行宏程序时设置是否在屏幕显示块进程状态 (9000.nc ~ 9999.nc). 0: 不显示
			9	T 代码调用宏程序数量	当执行 T 代码时, 输入调用的宏程序数量(9000.nc ~ 9009.nc). (9000 ~ 9009, 整数)
			10	宏程序调用 G 代码(9010.nc)	设置调用宏程序(9010.nc ~ 9019.nc)的 G 代码数量 . ※忽略设置值 0, 1, 2, 3. (0~255.9, 实数)
			
			19	宏程序调用 G 代码 (9019.nc)	设置调用宏程序(9010.nc ~ 9019.nc)的 G 代码数量 . ※忽略设置值 0, 1, 2, 3. (0~255.9, 实数)
			20	宏程序调用 M 代码 (9020.nc)	根据 M 代码分配可调用宏程序(9020.nc ~ 9029.nc) 的 M 代码数量. ※忽略输入值 0, 30. (0~255, 整数)
			
	29	宏程序调用 M 代码 (9029.nc)	根据 M 代码分配可调用宏程序(9020.nc ~ 9029.nc) 的 M 代码数量. ※忽略输入值 0, 30. (0~255, 整数)		

参数	组	No.	项目	描述
1. 通道参数	14. 默认设置	1	默认设置的模式移动	如果没有 G00 或者 G01, 选择 G 代码作为默认模式. 0: 快速移位(G00) 1: 进刀(G01)
		2	默认设置的模式平面	如果没有用于 G17, G18, G19 组的 G 代码指令, 选择 G 代码作为默认模式. 0: XY 平面(G17) 1: XZ 平面(G18) 2: YZ 平面(G19)
		3	默认设置的模式绝对/增量	如果没有用于 G90, G91 组的 G 代码指令,选择 G 代码作为默认模式. 0:绝对指令(G90) 1: 增量指令(G91)
		5	查看默认设置的模式禁止区域	如果没有用于 G22, G23 组的 G 代码指令,选择 G 代码作为默认模式. 0: 行程 On(G22) 1: 行程 Off(G23)
	16. 相对坐标设置	1	相对坐标补偿值 #1	设置 X 轴相对坐标补偿值.
		2	相对坐标补偿值 #2	设置 Y 轴相对坐标补偿值.
		3	相对坐标补偿值 #3	设置 Z 轴相对坐标补偿值.
		4	相对坐标补偿值 #4	设置 A 轴相对坐标补偿值.
		5	相对坐标补偿值 #5	设置 B 轴相对坐标补偿值.
		6	相对坐标补偿值 #6	设置 C 轴相对坐标补偿值.
		7	相对坐标补偿值 #7	设置 U 轴相对坐标补偿值.
		8	相对坐标补偿值 #8	设置 V 轴相对坐标补偿值.
		9	相对坐标补偿值 #9	设置 W 轴相对坐标补偿值.
	2. 通道/轴参数	1. 轴设置	2	设置用于模块轴的方向

第 6 章运动功能块

参数	组	No.	项目	描述
2. 通道/轴 参数	2. 原点	1	2 nd 原点坐标	设置 2 nd 原点坐标.
		2	3 rd 原点坐标	设置 3 rd 原点坐标.
		3	4 th 原点坐标	设置 4 th 原点坐标.
	3. 快速移位	2	快速移位加速	用于 G00 块加速的设置值.
		3	快速移位减速	用于 G00 块减速的设置值.
		4	快速移位加速度	用于 G00 块加速度的设置值.
		5	快速移位速度	用于 G00 块移位速度的设置值. (0~100000 unit/min, 实数)
	4.行程区域	1	用于 X, Y 和 Z 轴的 G22 行程-禁止区域范围的最小值	设置用于 X, Y 和 Z 轴的 G22 行程-禁止区域范围的最小值. (-100,000~100,000 unit, 实数)
		2	用于 X, Y 和 Z 轴的 G22 行程-禁止区域范围的最大值	设置用于 X, Y 和 Z 轴的 G22 行程-禁止区域范围的最大值. (-100,000~100,000 unit, 实数)
		3	用于 X, Y 和 Z 轴的 3 rd 行程-禁止区域范围的最小值	设置用于 X, Y 和 Z 轴的 3 rd 行程-禁止区域范围的最小值. (-100,000~100,000 unit, 实数)
		4	用于 X, Y 和 Z 轴的 3 rd 行程-禁止区域范围的最大值	设置用于 X, Y 和 Z 轴的 3 rd 行程-禁止区域范围的最大值. (-100,000~100,000 unit, 实数)
	5. 子设置	2	单方向定位的超限馈送率	<p>当使用单反向定位功能(G60)时, 设置 9 轴超限馈送率; X, Y, Z, A, B, C, U, V, W . 分别通过设置与 G60 指令块轴的值停止定位后,移动到指令位置以清除反冲影响.</p>  <p>(-100 ~ 100 unit, 实数)</p>

6.8.15 写入 NC 参数(NC_WriteParameter)



- (1) 该运动功能块写入 NC 通道和通道/轴中的指定值.
- (2) 参数在执行输入的上升沿写入.
- (3) ParameterGroup 输入参数写入的指定组.
- (4) ParameterNumber 输入参数写入组的指定编号. 如果无法应用设置值,发生"错误 16 # 000B".
- (5) 在 Value 输入中, 指定要写入的参数值.
- (6) 对于组编号和每个参数组的数量, 参考 6.7.14 读取 NC 参数.

第7章程序

7.1 程序结构

运动控制模块程序可分为主任务程序，周期任务程序和初始化任务程序。各程序功能如下所示。

7.1.1 程序结构

运动控制模块的初始化任务，主任务和周期任务基于循环执行。每个任务具有固定循环，并且用户在默认参数中设置。有两个配置循环：主任务循环和周期任务循环。初始化任务采用主任务的循环。

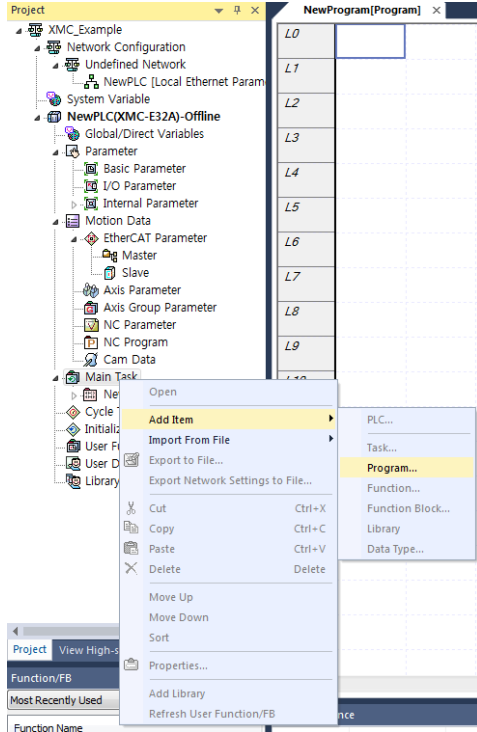
项目	描述
初始化任务程序	<ul style="list-style-type: none"> 初始化任务程序是在电源上电时，运动控制模块完成自身运行初始化要求后执行的第一个任务程序。保持执行直到执行INIT_DONE指令。 当执行初始化程序时，仅执行初始化程序，而主任务程序和周期任务程序不执行，直到INIT_DONE指令执行。 即使当初始化任务程序运行时，I/O更新和其他功能正常执行。 初始化任务程序时满足运动控制模块初始化设置的多种运行要求。
主任务程序	<ul style="list-style-type: none"> 该程序在运动控制模块中设置的每个主任务循环执行。 主任务的循环可以在基本参数"主任务循环"中设置，可选择范围500 μs, 1 ms, 2 ms或者4 ms。 当主任务程序运行时间超出设置的主任务循环，发生循环报警。如果在检测循环错误期间主任务程序未完成，发生循环错误。
周期任务程序	<ul style="list-style-type: none"> 在运动控制模块中设置的每个周期任务循环执行程序。 周期任务循环可以在基本参数的"周期任务循环"中设置，并且必须设置为主任务循环的倍数。 周期任务程序在每个主任务循环执行主任务程序完成后的空闲时间执行，然后在每个周期任务循环反复执行。

更多关于主任务程序和周期任务程序的内容，参考"4.3 运动控制任务"。

7.1.2 如何设置程序

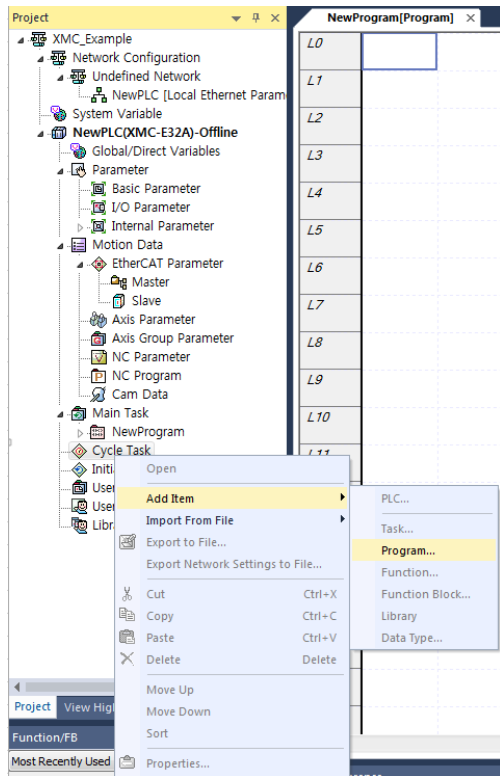
1) 如何设置主任务程序

在主任务位置, 右击并点击『增加项目』-『程序』.(然而, 当创建工程时, 主任务程序已经建立.)



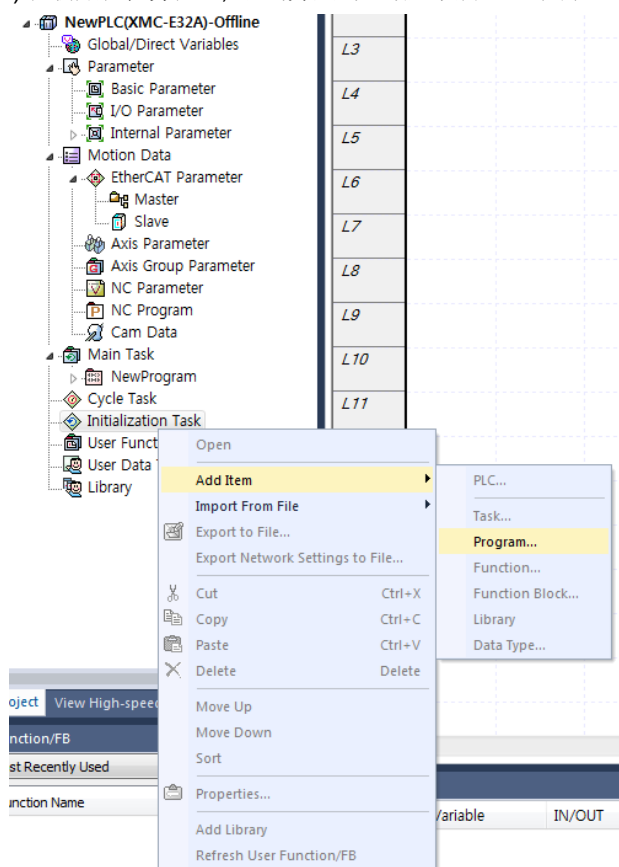
2) 如何设置循环任务程序

在循环任务位置, 右击并点击『增加项目』-『程序』.

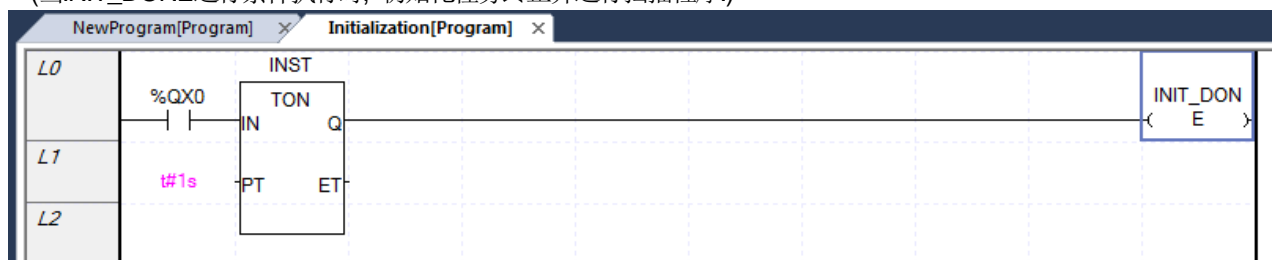


3) 如何设置初始化任务程序

(1) 在初始化任务位置, 右击并点击『增加项目』-『程序』.



创建必要的初始化程序. 确保在初始化任务程序中写入INIT_DONE指令.
(当INIT_DONE运行条件执行时, 初始化任务终止并运行扫描程序.)

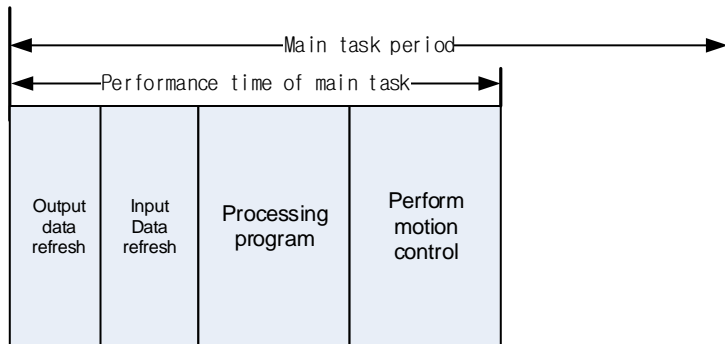


7.1.3 程序运行时间

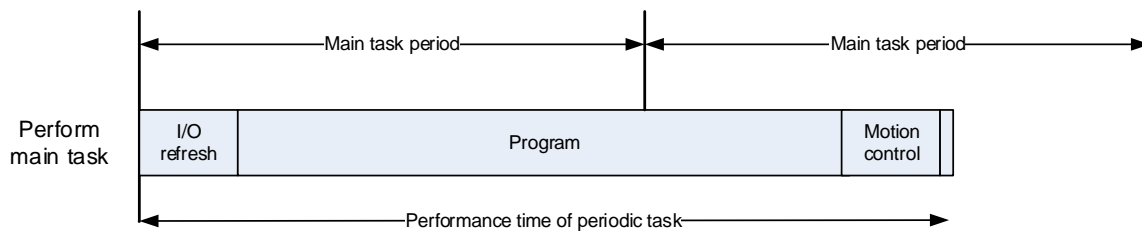
每个任务程序执行时间(扫描时间)如下.表示从每个控制循环起始到运动控制模块执行完成的时间要求, 即, 完成任务的时间.

1) 主任务运行时间

表示从主任务循环起始到运动控制模块执行完成的时间.

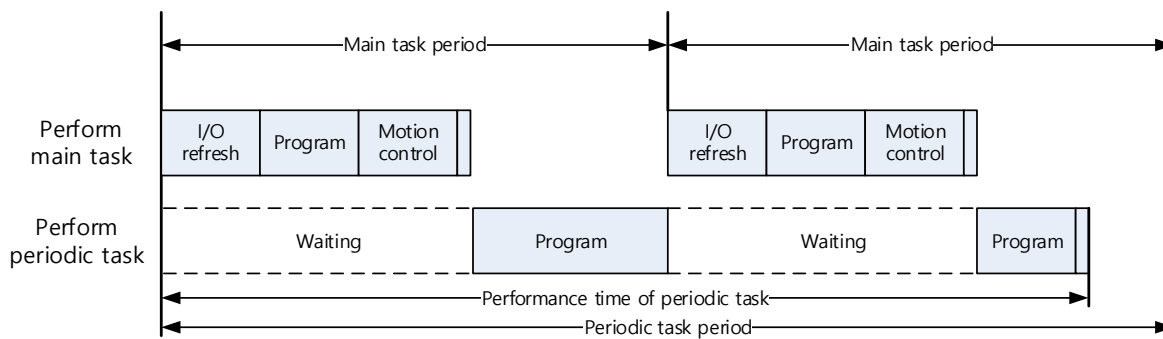


如果一个循环的主任务没有完成, 仅测量从主任务循环起始到主任务如下完成时的时间.



2) 周期任务运行时间

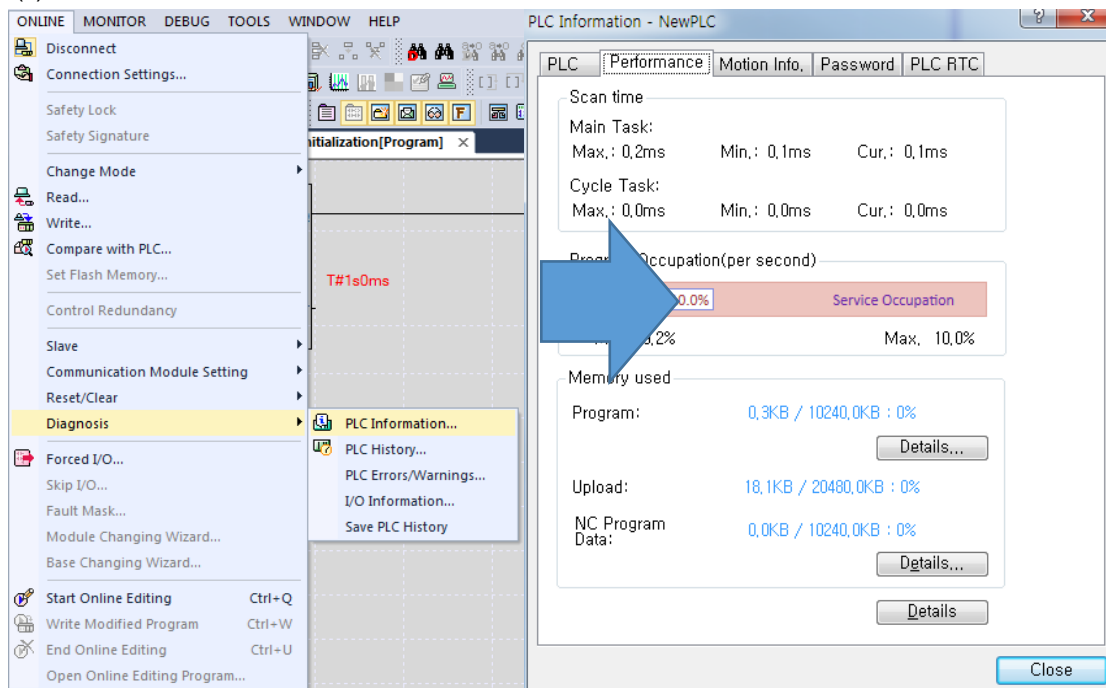
表示从周期任务循环起始到周期任务程序完成的时间. 由于周期任务在主任务完成后运行, 所以可以分别在多个主任务循环中执行.



3) 如何查看任务运行时间

运动控制模块任务运行时间可通过MP500或者如下标志位查看。

(1) 使用MP500: 点击『在线』-『诊断』-『PLC信息』-『性能』。



(2) 使用标志位: 扫描时间保存在如下系统标志位(F)区域中。

WORD	标志位	描述
%FW512	_PTASK_SCAN_MAX	主任务程序最大扫描时间(单位:100us)
%FW513	_PTASK_SCAN_MIN	主任务程序最小扫描时间(单位:100us)
%FW514	_PTASK_SCAN_CUR	主任务程序当前扫描时间(单位:100us)
%FW515	_CTASK_SCAN_MAX	周期任务程序最大扫描时间(单位:100us)
%FW516	_CTASK_SCAN_MIN	周期任务程序最小扫描时间(单位:100us)
%FW517	_CTASK_SCAN_CUR	周期任务程序当前扫描时间(单位:100us)

7.2 状态信息读取

在运动控制模块的程序中，在运动控制模块中每个轴，组状态和运行状态都可通过标志检查。

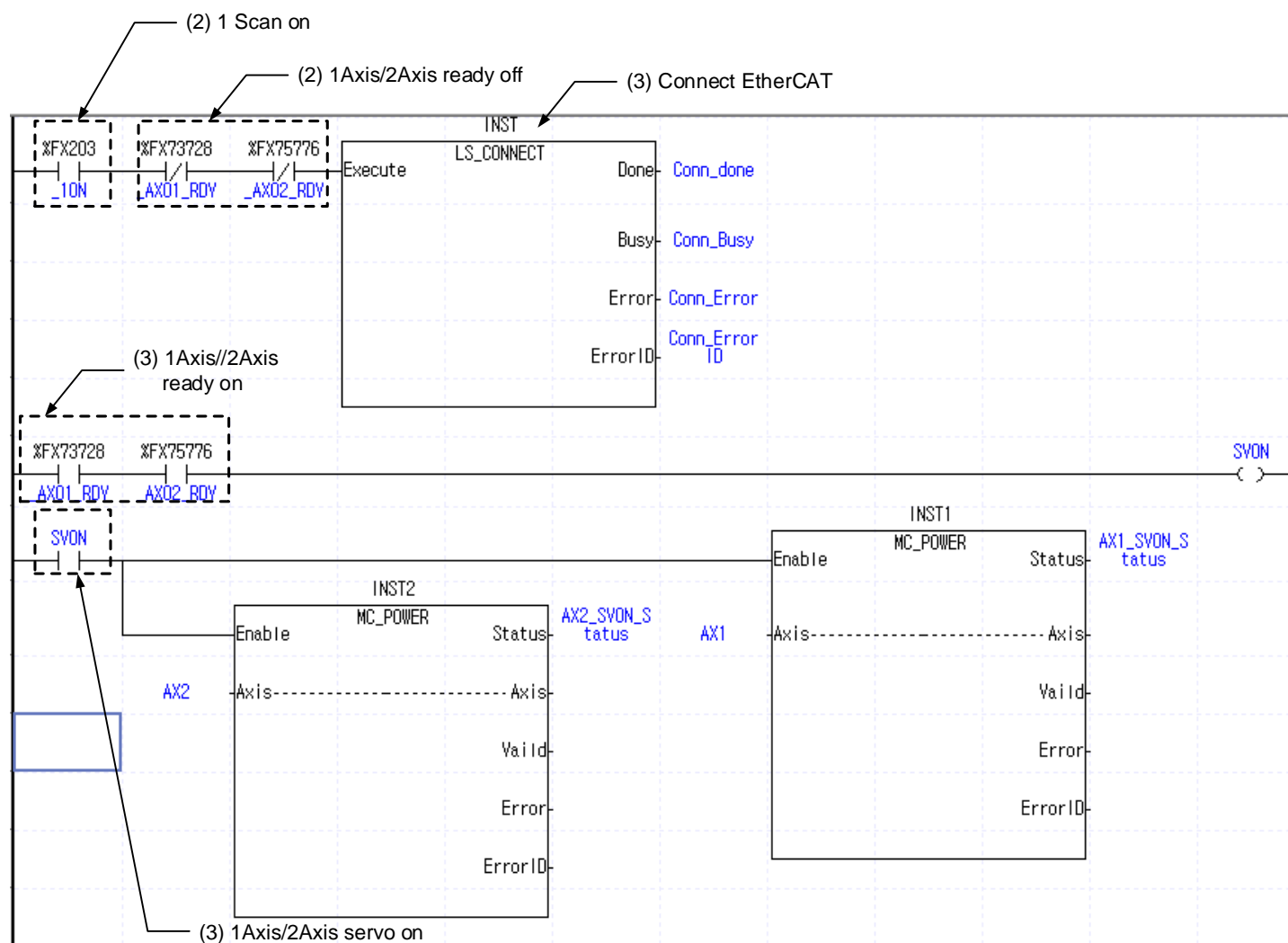
本章节中大部分程序范例创建使用标志表示轴和轴组的状态。

标志位表示在程序中使用的状态信息，可以传达给 PLC CPU，分配到运动控制模块共用设备。
更多关于标志位的类型和功能信息，参考“附录 1.标志位列表”。

7.3 离散运动程序

7.3.1 运行准备

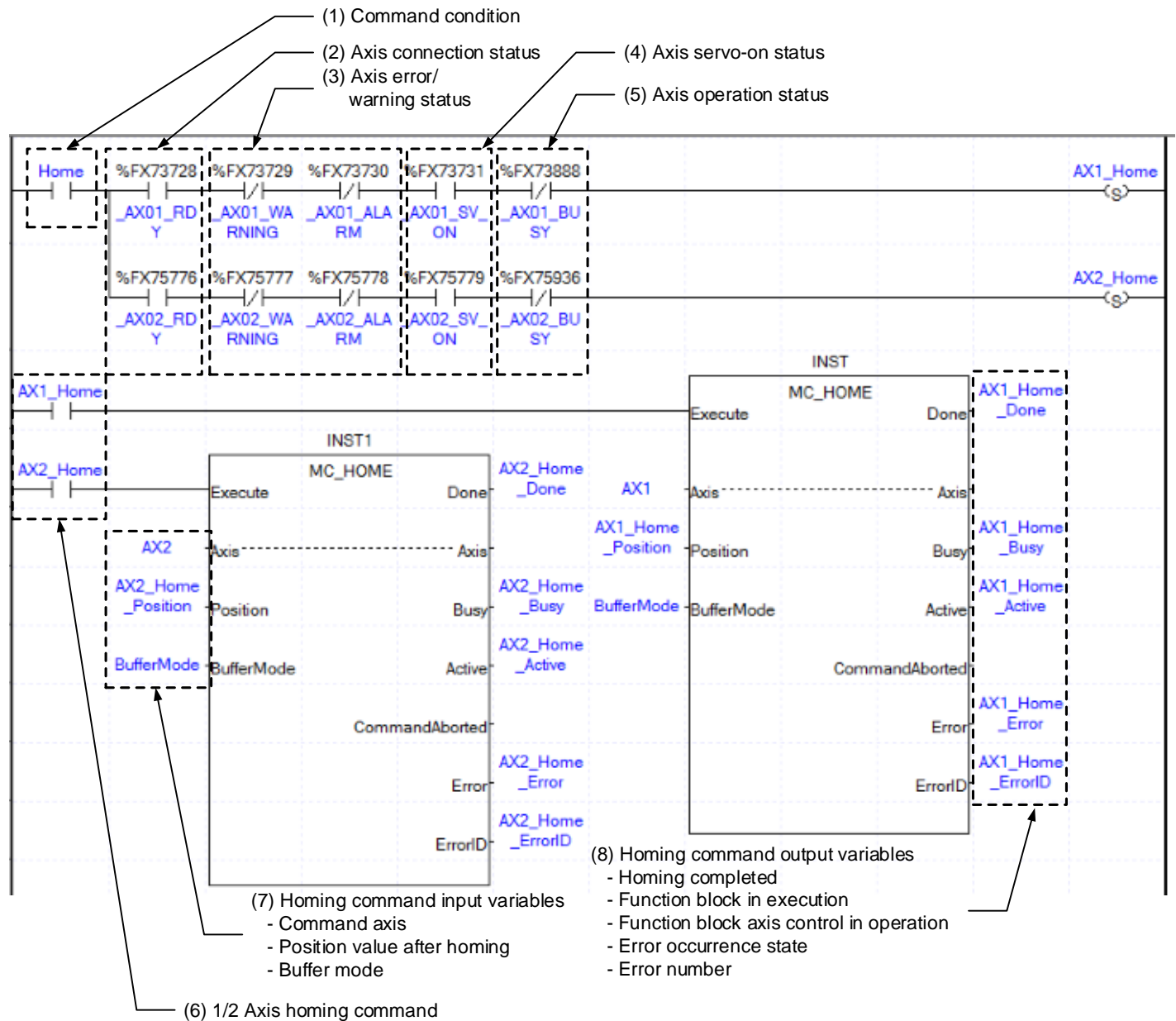
以下范例程序通过以太网线缆连接伺服驱动器，使得连接伺服驱动器为 On，运行 EtherCAT 伺服驱动器。



- (1) 以上范例假设 1-轴和 2-轴两个轴连接到运动控制模块的情况。
- (2) 当运动控制模块为 RUN 时，如果 1-轴和 2-轴没有连接，在运动控制模块和伺服驱动器之间使用功能块(LS_CONNECT)，启动 EtherCAT 通信连接。
- (3) 如果运动控制模块和伺服驱动器之间的 EtherCAT 通信正常连接，伺服 On/Off (MC_Power) 命令使 SVON 触点为 On，分给每个轴。
- (4) 如果连接 1-轴和 2-轴的伺服驱动器没有错误，伺服正常 On，准备运行 1-轴和 2-轴。

7.3.2 原点返回运行

返回表示当上电后，执行设置机器原点。伺服执行原点返回后，返回方式取决于伺服驱动器厂商。在运动控制模块中，返回命令完成 and 错误情况可监控。原点返回后原点位置控制。



- (1) 命令条件
：使轴执行原点返回运行的条件。
- (2) 轴连接状态标志
：如果运行的轴连接到运动控制模块，EtherCAT 通信连接的动控制模块正常执行，则为 On。
- (3) 轴错误/报警状态标志
：如果轴上没有错误和报警，则为 On。
- (4) 轴 servo-on 状态标志
：如果轴为 servo-on 状态，则为 On，若为 servo-off state，变为 Off。

(5) 轴运行状态标志

: 如果轴在运行, 则为 On。

(6) 1/2 轴运点返回命令

: 在示例中, 返回 (MC_Home) 运动功能块在以下条件时执行。

- 返回条件为 On
- 轴连接正常
- 无错误和报警
- Servo-on 状态
- 非运行

运动功能块执行条件分类取决于系统

(7) 返回命令输入变量

: 执行返回(MC_Home)运动功能块输入变量如下。

- 命令轴: 设置执行运动功能块的轴。
- 原点返回后位置值: 当返回完成后, 设置位置值。
- 缓存模式: 当运动功能块执行时, 设置时间点。用于设置是否立即执行或当前命令完成后执行。更多细节请参考功能块, “6.1.4 缓存模式输入”。

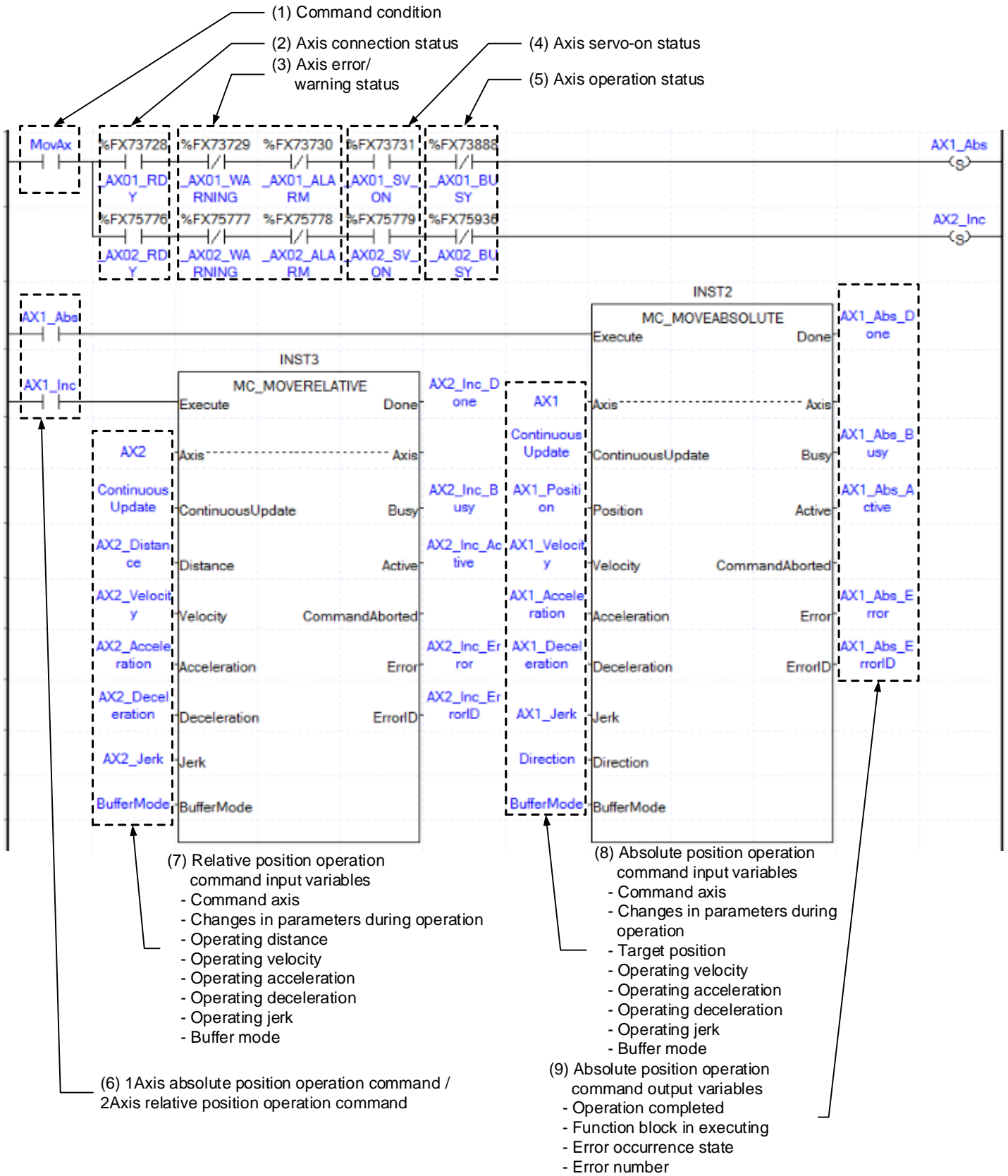
(8) 返回命令输出变量

: 当执行原点返回(MC_Home) 运动功能块时, 用于存放生成的输出值。

- 返回完成: 如果返回运行完成后, 则为 On。
- 执行功能块: 如果运动功能块正在执行, 则为 On, 返回完成为 On 时, 则为 Off。
- 功能块轴控制运行: 如果运动功能块控制轴, 则为 On。
- 错误发生状态: 当运动功能块执行时发生错误, 则为 On。
- 错误编号: 发生错误时, 根据错误生成的错误编号。
- 更多关于运动功能块输出细节, 请参考“6.1.3 基本 I/O 变量”中的“Edge 运行运动功能块”。

7.3.3 绝对位置/相对位置运行

以下是使用运动控制模块进行绝对位置和相对位置的程序。绝对位置基于原点，相对位置是当前位置。



(1) 命令条件

: 使轴执行位控运行的条件。

(2) 轴连接状态标志

: 如果运行的轴连接到运动控制模块, EtherCAT 通信连接的动控制模块正常执行, 则为 On。

(3) 轴错误/报警状态标志

: 如果轴上没有错误和报警, 则为 On。

(4) 轴 servo-on 状态标志

: 如果轴为 servo-on 状态, 则为 On, 若为 servo-off state, 变为 Off。

(5) 轴运行状态标志

: 如果轴在运行, 则为 On。

(6) 1-轴绝对位置运行/2-轴相对位置运行命令

: 在示例程序中, 绝对位置运行 (MC_MoveAbsolute)在 1-轴执行, 相对位置运行 (MC_MoveRelative) 2-轴在以下条件。

- 轴运行条件为 On
- 轴连接正常
- 无错误和报警
- Servo-on 状态
- 非运行

运动功能块执行条件分类取决于系统

(7) 相对位置运行命令输入变量

- : 执行相对位置运行 (MC_MoveRelative) 运动功能块输入变量。
- 命令轴: 设置执行运动功能块的轴。
- 运行中改变参数: 设置在运动功能块输入变量变化后是否应用运行。
更多细节, 参考“6.1.5 运动功能块执行中改变参数”。
- 运行距离: 设置执行相对坐标运行的距离。基于当前位置, + 值表示正向, - 值表示反向。
- 运行速率: 设置执行相对坐标运行的速率。
- 运行加速, 运行减速, 运行 jerk: 分别设置应用于相对坐标运行的值。
- 缓存模式: 当运动功能块执行时, 设置时间点。用于设置是否立即执行或当前命令完成后执行。更多细节请参考功能块, “6.1.4 缓存模式输入”。

(8) 绝对位置运行命令输入变量

- : 用于执行绝对位置运行 (MC_MoveAbsolute)运动功能块的输入变量。
- 命令轴: 设置执行运动功能块的轴。
- 运行中改变参数: 设置在运动功能块输入变量变化后是否应用运行。
更多细节, 参考“6.1.5 运动功能块执行中改变参数”。
- 目标位置: 设置移动到绝对坐标运行的位置。
- 运行速率: 当绝对位置运行执行到目标位置时, 设置速率。
- 运行加速, 运行减速, 运行 jerk: 分别设置应用于相对坐标运行的值。
- 运行方向: 当移动到目标位置时, 设置方向。设置 1, 移动到目标位置时正向运行。设置 2, 运行到目标区域的方向是基于当前位置的最短距离。设置为 3, 反向。设置为 4, 当前运行方向上移动到目标位置。
- 缓存模式: 当运动功能块执行时, 设置时间点。用于设置是否立即执行或当前命令完成后执行。更多细节请参考功能块, “6.1.4 缓存模式输入”。

第7 程序

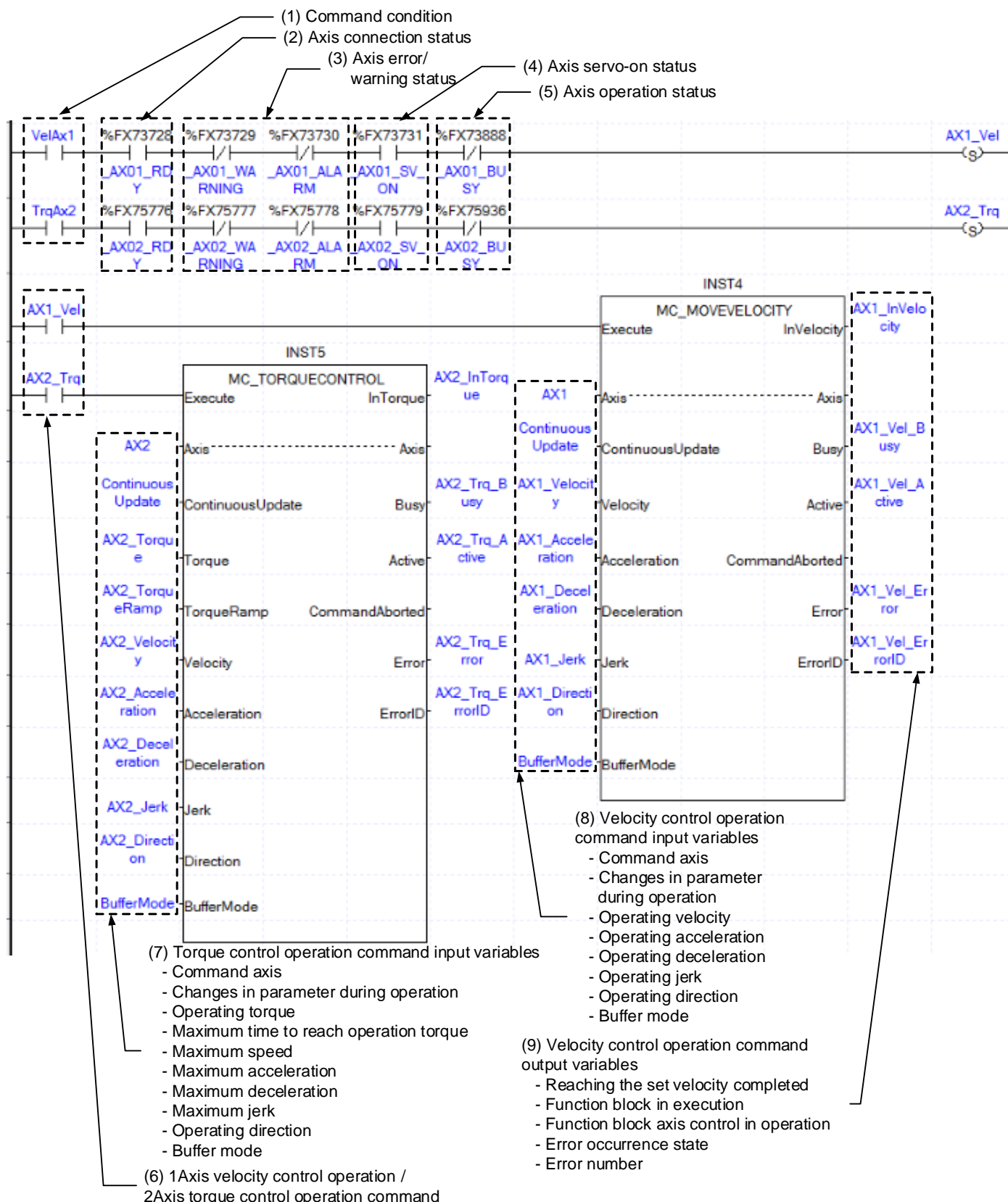
(9) 绝对位置运行命令输出变量

: 当绝对位置运行(MC_MoveAbsolute) 运动功能执行时, 变量用于存放生成的输出值。

- 运行完成: 当绝对坐标运行完成时, 则为 On。
- 功能块执行: 当运动功能块执行时, 则为 On, 如果运行完成为 On, 则为 Off。
- 功能块轴控制执行: 如果运动功能块控制轴, 则为 On。
- 错误发生状态: 当运动功能块执行时发生错误, 则为 On。
- 错误编号: 发生错误时, 根据错误生成的错误编号。
- 更多关于运动功能块输出细节, 请参考“6.1.3 基本 I/O 变量”中的“Edge 运行运动功能块”。

7.3.4 速度/转矩控制运行

以下示例使用运动功能块进行速度控制和转矩控制运行。对于转矩控制，使用伺服驱动器的转矩控制，在运动控制模块中，执行转矩控制的命令，执行完成和状态如下监控。



第7 程序

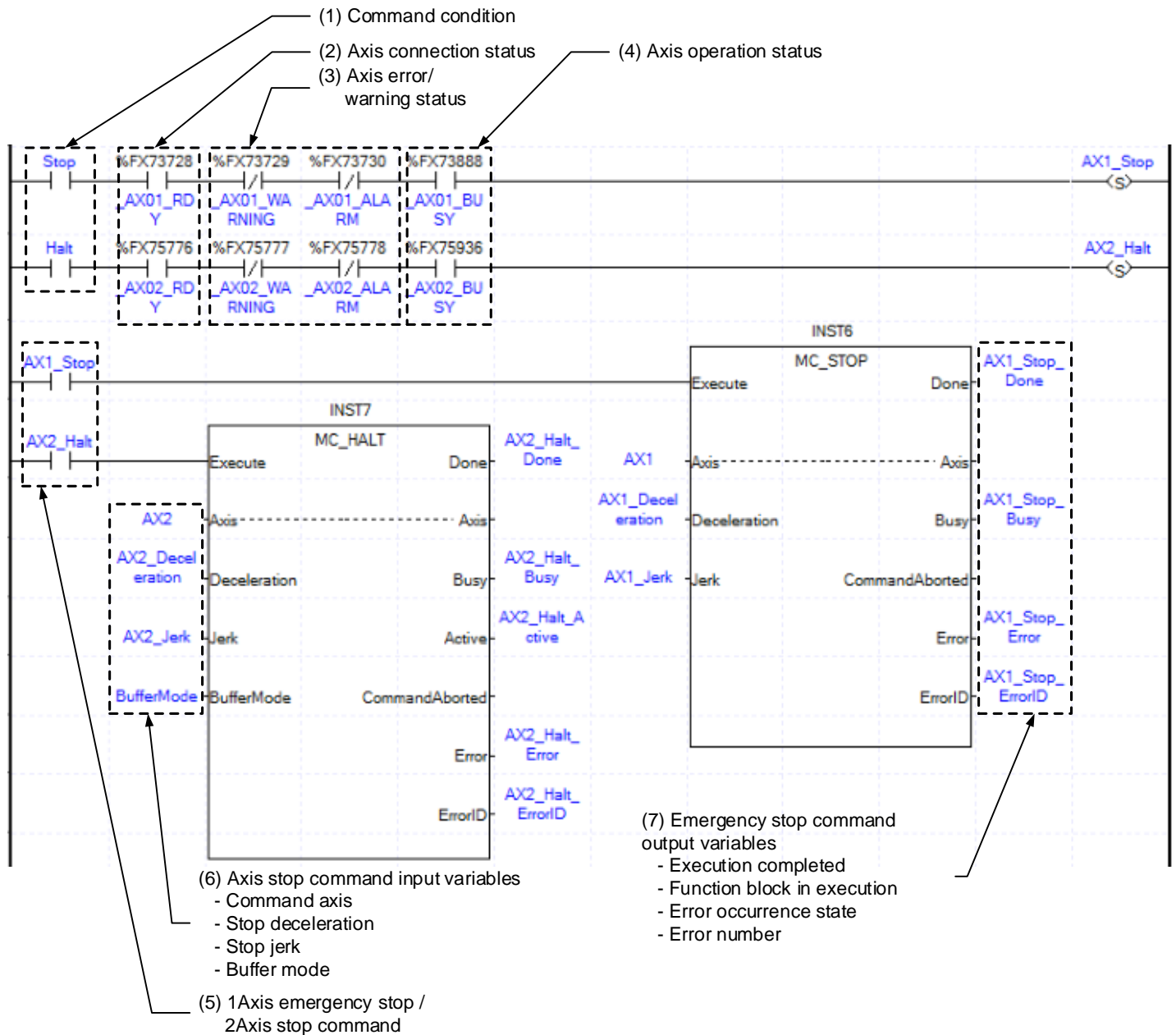
- (1) 命令条件
：使轴执行速度控制/转矩控制运行的条件。
- (2) 轴连接状态标志
：如果运行的轴连接到运动控制模块，EtherCAT 通信连接的动控制模块正常执行，则为 On。
- (3) 轴错误/报警状态标志
：如果轴上没有错误和报警，则为 On。
- (4) 轴 servo-on 状态标志
：如果轴为 servo-on 状态，则为 On，若为 servo-off state，变为 Off。
- (5) 轴运行状态标志
：如果轴在运行，则为 On。
- (6) 1-轴速度控制运行/2-轴转矩控制运行命令
：在示例程序中，1-轴执行定速运行(MC_MoveVelocity)运动功能块，2-轴在以下条件中执行转矩控制运行(MC_TorqueControl)运动功能块。
 - 周运行条件为 On.
 - 轴连接正常.
 - 无错误和报警.
 - Servo-on 状态
 - 非运行运动功能块执行条件分类取决于系统
- (7) 转矩控制运行命令输入变量
：输入变量执行转矩控制运行(MC_TorqueControl) 运动功能块。
 - 命令轴: 设置执行运动功能块的轴。
 - 运行中改变参数: 设置在运动功能块输入变量变化后是否应用运行更多细节，参考“6.1.5 运动功能块执行中改变参数”。
 - 运行转矩: 设置转矩控制运行的转矩值。
 - 到达运行转矩的最大时间: 从当前转矩到设置转矩的最大斜率。单位 [Unit/s]。
 - 最大速度，最大减速，最大加速，最大 jerk: 分别在转矩控制运行中设置值。
 - 运行方向: 转矩控制运行方向设置。设置为 1，正向。设置为 3，反向。设置为 4，当前运行方向。
 - 缓存模式: 当运动功能块执行时，设置时间点。用于设置是否立即执行或当前命令完成后执行。更多细节请参考功能块，“6.1.4 缓存模式输入”。
- (8) 速度控制运行命令输入变量
：输入变量执行定速运行(MC_MoveVelocity) 运动功能块。
 - 命令轴: 设置执行运动功能块的轴。
 - 运行中改变参数: 设置在运动功能块输入变量变化后是否应用运行更多细节，参考“6.1.5 运动功能块执行中改变参数”。
 - 运行速率: 在速度控运行中设置速率。
 - 运行加速，运行加速，运行加速度: 在速度控制运行中分别设置值。
 - 运行方向: 在速度控制运行中设置方向。设置为 1，正向运行。设置为 3，反向运行。设置为 4，当前运行方向运行。
 - 缓存模式: 当运动功能块执行时，设置时间点。用于设置是否立即执行或当前命令完成后执行。更多细节请参考功能块，“6.1.4 缓存模式输入”。

(9) 速度控制运行命令输出变量

- : 当定速运行(MC_MoveVelocity) 运动功能执行时, 变量用于存放生成的输出值。
- 到达设置速度完成: 当通过速度控制运行到达设置速度时, 则为 On。
- 功能块执行: 如果运动功能块执行, 则为 On, 如果运行完成, 则为 Off。
- 功能块轴控制执行:如果运动功能块控制轴, 则为 On。
- 错误发生状态: 如果运动功能块执行时发生错误, 则为 On。
- 错误编号: 发生错误时, 根据错误生成的错误编号。
- 更多关于运动功能块输出细节, 请参考“6.1.3 基本 I/O 变量”中的“Edge 运行运动功能块”。

7.3.5 轴停止

以下是在运行时使用轴停止程序。运动功能块停止运行中的轴包括“Immediate Stop (MC_Stop)”和“Halt (MC_Halt)”。如果使轴紧急停止，执行“Immediate Stop (MC_Stop)”，在停止时，不能执行其它运动功能块。如果使轴停止，执行“Halt (MC_Halt)”，停止时，停止状态被其它运动功能块中断，其它运动功能块执行。更多细节，参考“第六章指令”。

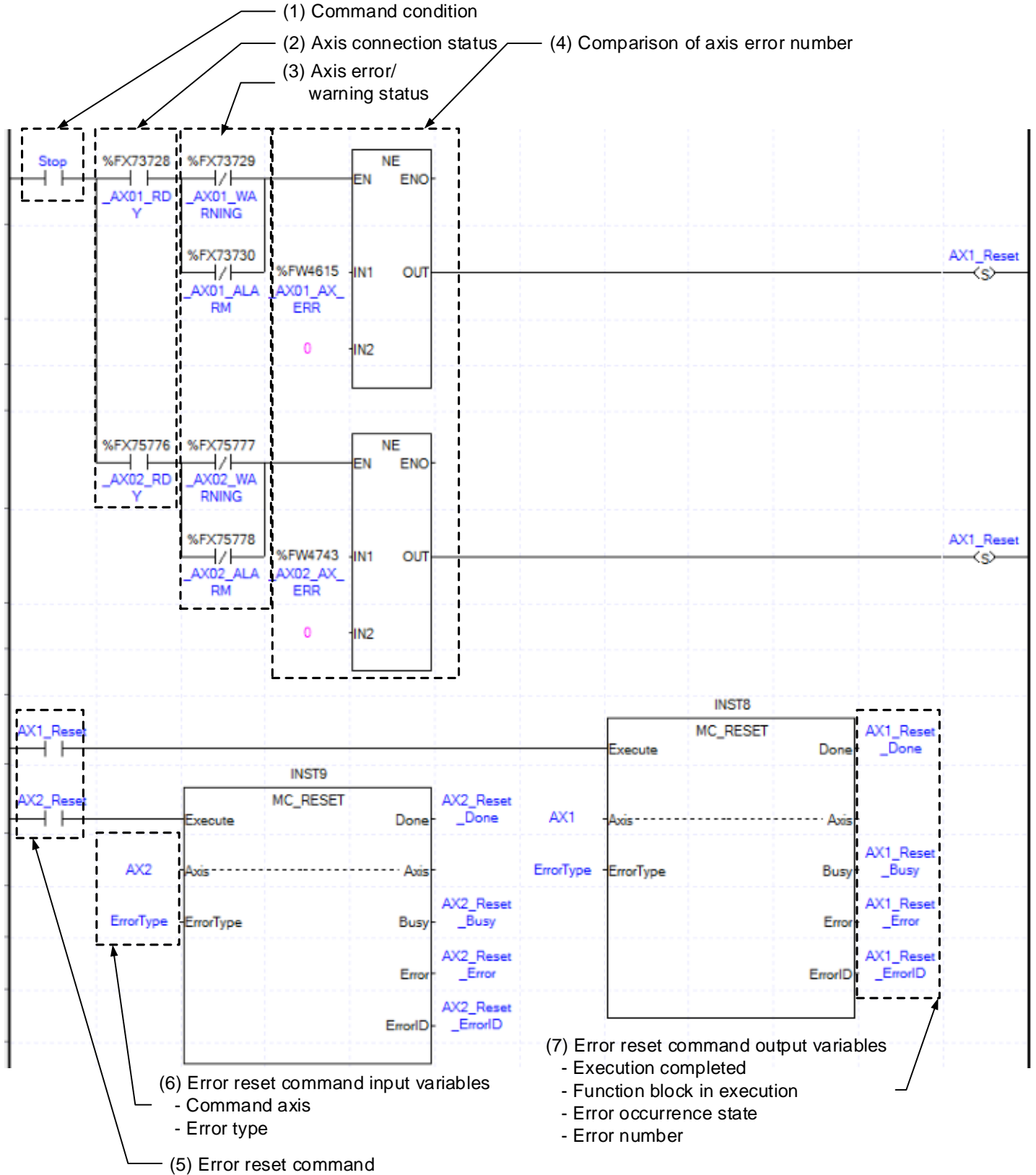


- (1) 命令条件
: 给轴紧急停止/停止命令的条件。
- (2) 轴连接状态标志
: 如果运行的轴连接到运动控制模块, EtherCAT 通信连接的动控制模块正常执行, 则为 On。
- (3) 轴错误/报警状态标志
: 如果轴有错误和报警发生, 则为 On。
- (4) 轴运行状态标志
: 如果轴在运行, 则为 On。
- (5) 1-轴紧急停止/2-轴轴停止命令
: 在示例程序中, 1-轴, 执行紧急停止(MC_Stop) 运动功能块, 2-轴在以下条件下执行 halt (MC_Halt) 运动功能块。
 - 轴停止状态为 On。
 - 轴连接正常。
 - 无错误和报警。
 - 运行中运动功能块执行条件分类取决于系统
- (6) 轴停止命令输入变量
: 执行 Halt (MC_Halt) 运动功能块的输入变量。
 - 命令轴: 设置执行运动功能块的轴。
 - 停止减速: 从给轴停止运行速度到停止设置减速。
 - 停止 jerk: 在停止时间设置 jerk。
 - 缓存模式: 当运动功能块执行时, 设置时间点。用于设置是否立即执行或当前命令完成后执行。更多细节请参考功能块“6.1.4 缓存模式输入”。
- (7) 紧急停止命令输出变量
: 当执行立即停止(MC_Stop) 运动功能块时, 用于存放生成输出值的变量。
 - 执行完成: 如果轴停止, 则为 On。
 - 功能块执行: 如果运动功能块执行, 则为 On。如果执行完成, 则为 Off。
 - 错误发生状态: 当运动功能块执行时, 发生错误, 则为 On。
 - 错误编号: 发生错误时, 根据错误生成的错误编号。
 - 更多关于运动功能块输出细节, 请参考“6.1.3 基本 I/O 变量”中的“Edge 运行运动功能块”。

第7 程序

7.3.6 错误处理

以下示例程序检查轴错误，进行复位。



(1) 命令条件

:给轴错误复位命令的条件。

(2) 轴连接状态标志

:如果运行的轴连接到运动控制模块, EtherCAT 通信连接的动控制模块正常执行, 则为 On。

(3) 轴错误/报警状态标志

: 如果轴上没有错误和报警, 则为 On。

(4) 比较轴错误编号

: 在示例程序中, 通过比较轴上的错误编号标志值不是 0, 决定错误复位条件。

(5) 错误复位命令

: 在示例程序中, 在以下条件中, 执行轴错误复位 (MC_Reset) 运动功能块。

- 轴运行条件为 On。

- 轴连接正常。

- 需为错误和报警

- 错误编号不为 0。

运动功能块执行条件分类取决于系统

(6) 错误复位命令输入变量

: 输入变量用于执行轴错误复位(MC_Reset) 运动功能块。

- 命令轴: 设置执行运动功能块的轴。

- 错误类型: 如果错误复位的错误类型设置 0 代表轴错误, 1 常规错误。

(7) 错误复位命令输出变量

:当执行轴错误复位(MC_Reset) 运动功能块时, 变量用于存放生成的输出值。

- 执行完成: 运动功能块执行完成, 则为 On。

- 功能块执行:如果运动功能块执行, 则为 On。如果执行完成, 则为 Off。

- 错误发生状态:当运动功能块执行时, 发生错误则为 On。

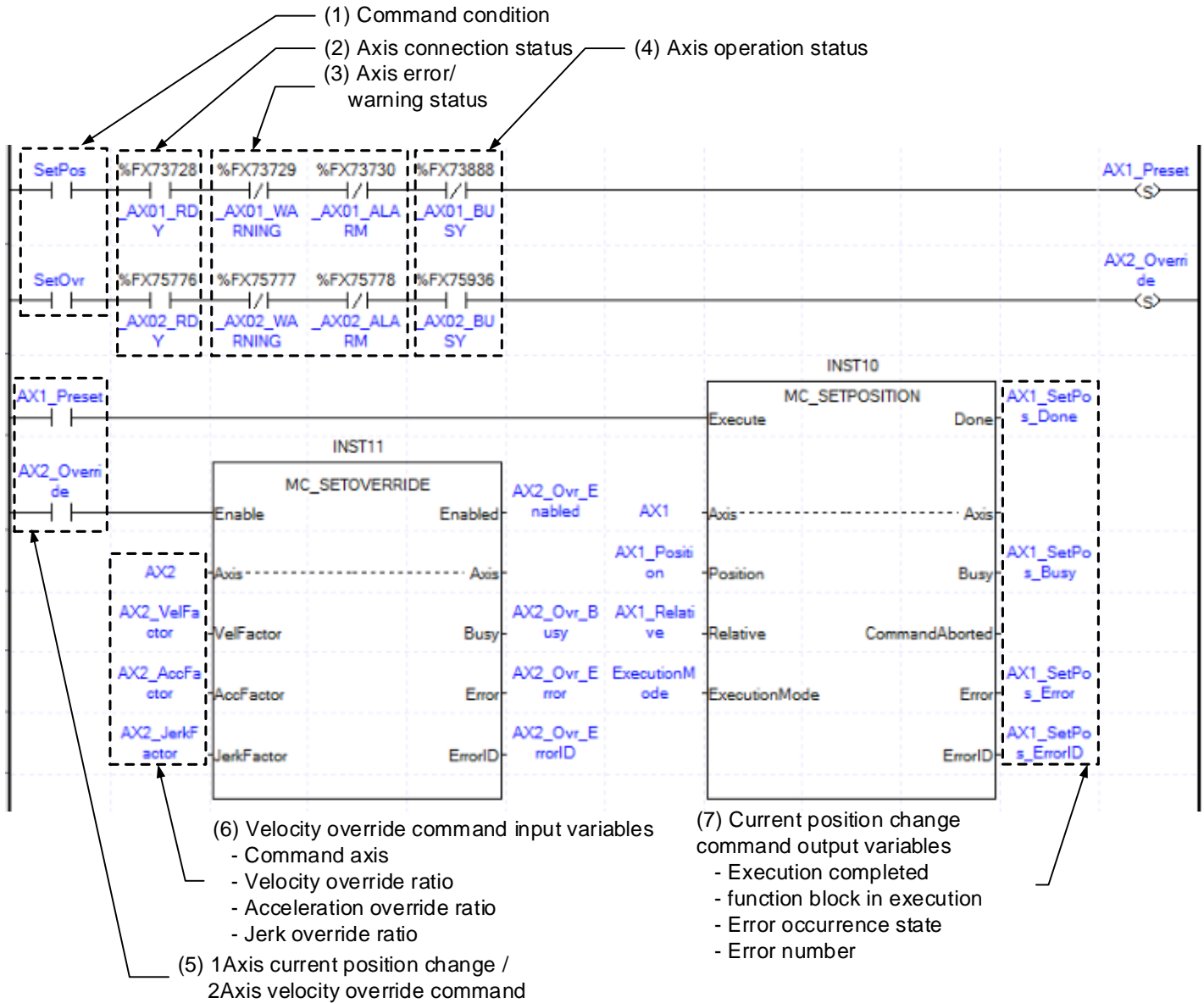
- 错误编号: 发生错误时, 根据错误生成的错误编号。

- 更多关于运动功能块输出细节, 请参考“6.1.3 基本 I/O 变量”中的“Edge 运行运动功能块”。

第7 程序

7.3.7 运行中改变

以下示例程序在运行中改变轴的当前位置和速度。



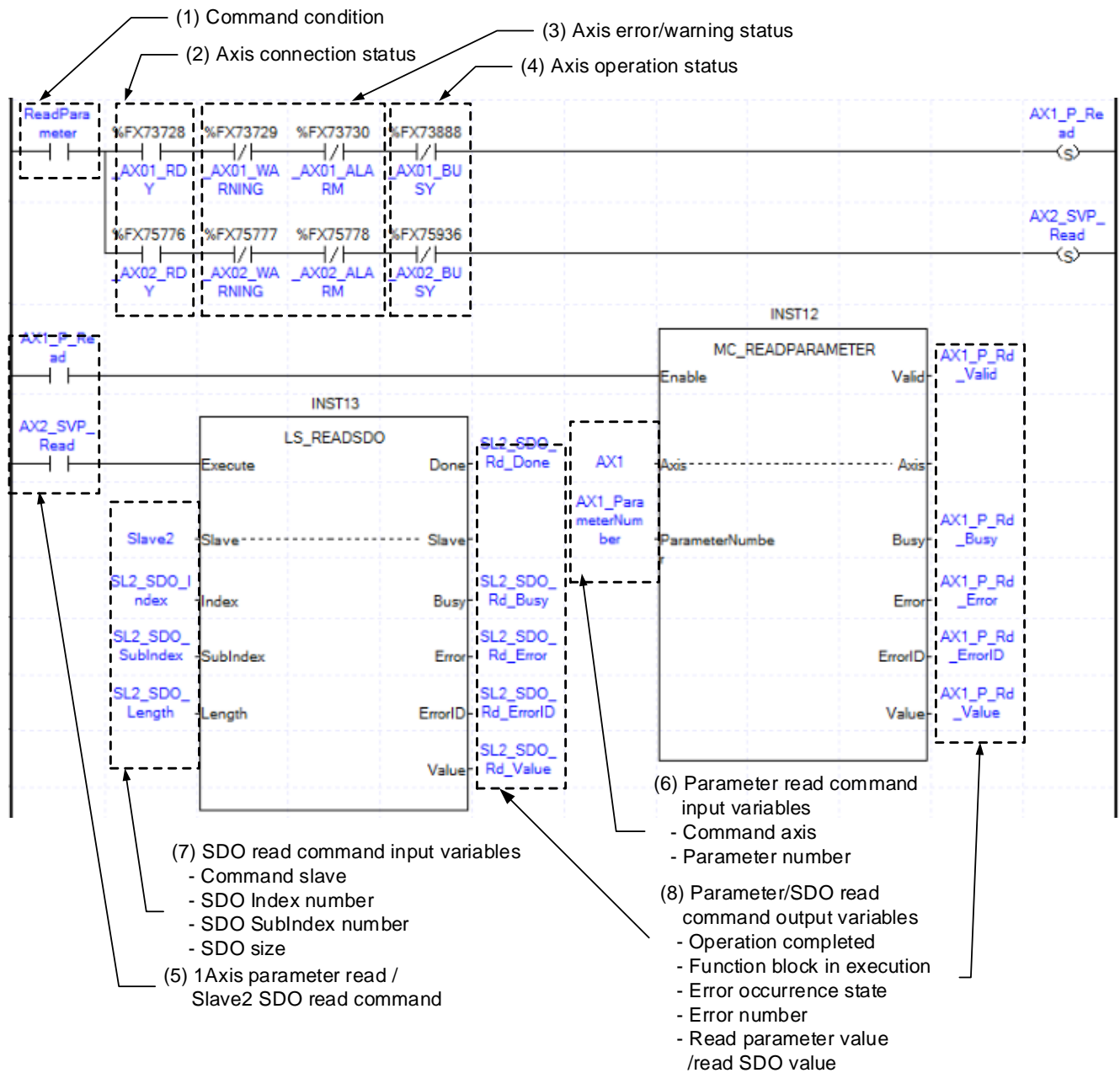
- (1) 命令条件
：给轴当前位置改变，运行速度改变的指令条件。
- (2) 轴连接状态标志
：如果运行的轴连接到运动控制模块，EtherCAT 通信连接的动控制模块正常执行，则为 On。
- (3) 轴错误/报警状态标志
：如果轴上没有错误和报警，则为 On。
- (4) 轴运行状态标志
：如果轴在运行，则为 On。
- (5) 1-轴当前位置改变/2-轴速度 override 指令
：在示例程序中，在以下条件中执行当前位置设置 (MC_SetPosition) 运动功能块。
- 当前位置改变条件为 On.
 - 轴连接正常.
 - 无错误和报警.
 - 轴非运行.
- 另外，在以下条件中执行速度/加速 override (MC_SetOverride) 运动功能块。
- 运行速度改变条件为 On.
 - 轴连接正常.
 - 无错误和报警.
 - 轴在运行.
- 执行运动功能块条件取决于系统。
- (6) 速度 override 命令输入变量
：输入变量执行速度/加速 override (MC_SetOverride) 运动功能块。
- 命令轴: 设置执行运动功能块的轴。
 - 速度 override 率: 比较运行速度和当前设置速度，设置速度变化率。
 - 加速 override 率: 比较当前加速至和当前设置值，设置加速变化率。
 - Jerk override 率: 比较 jerk 值和当前设置，设置 jerk 的变化率。如果设置为 2，设置双倍当前设置值。
- (7) 当前位置改变命令输出变量
- ：当执行当前位置设置(MC_SetPosition) 运动功能块时，变量用于存放生成的输出值。
 - 执行完成: 如果运动功能块执行完成，则为 On。
 - 功能块执行: 如果运动功能块执行，则为 On。如果执行完成，则为 Off。
 - 错误发生状态: 当运动功能块执行时，发生错误，则为 On。
 - 错误编号: 发生错误时，根据错误生成的错误编号。
 - 更多关于运动功能块输出细节，请参考“6.1.3 基本 I/O 变量”中的“Edge 运行运动功能块”。

7.3.8 参数写入/读取

参数读取/写入指令包含与“SDO Write (LS_WriteSDO)”和“SDO Read (LS_ReadSDO)”一样的“参数写入(MC_WriteParameter)”和参数读取(MC_ReadParameter)。“参数写入(MC_WriteParameter)”和“参数读取(MC_ReadParameter)”是轴参数写入或者读取的指令，而“SDO 写入(LS_WriteSDO)”和“SDO 读取(LS_ReadSDO)”是连接的 EtherCAT 从站读取或者写入 SDO 数据的指令。

如果连接 EtherCAT 的从站为伺服驱动，SDO 数据变为伺服参数。以下示例为程序希尔或者变更运行参数、使用读取/写入指令伺服驱动的伺服参数、当从站 2 为伺服驱动并且连接到 2 轴的参数。

■ 参数读取



(1) 命令条件

: 读取轴的伺服参数条件。

(2) 轴连接状态标志

: 如果运行的轴连接到运动控制模块, EtherCAT 通信连接的动控制模块正常执行, 则为 On。

(3) 轴错误/报警状态标志

: 如果轴上没有错误和报警, 则为 On。

(4) 轴运行状态标志

: 如果轴在运行, 则为 On。

(5) 1-轴参数写/2-轴伺服参数读命令

: 在示例程序中, 在以下条件下, 1-轴执行参数读(MC_ReadParameter) 运动功能块, 2-轴执行伺服参数读 (LS_ReadServoParameter) 运动功能块。

- 参数读条件为 On.

- 轴连接正常.

- 无错误和报警.

- 非运行

执行运动功能块条件取决于系统。

(6) 参数读命令输入变量

: 输入变量执行参数读 (MC_ReadParameter) 运动功能块。

- 命令轴: 设置执行轴的运动功能块。

- 参数编号: 设置读取轴的运动功能块参数编号。

参数编号如下所示。

编号	参数	项目	设置
0	基本参数	单位	0:pulse, 1:mm, 2:inch, 3:degree
1		每转脉冲	1~4,294,967,295 [pulse]
2		每转距离	0.000000001 ~ 4,294,967,295 [Unit]
3		速度指令单位	0:Unit/Time, 1:rpm
4		速度限制	LREAL (正数) [Unit/s, rpm]
5		减速急停	0, LREAL (正数) [Unit/s ²]
6		编码器选择	0:增量编码器,1:绝对编码器
7		齿轮齿数比(电机)	1 ~ 65,535
8		齿轮齿数比(机器)	1 ~ 65,535
9		反转运行模式	0:禁用, 1:使能
10	扩展参数	SW 上限	LREAL [Unit]
11		SW 下限	LREAL [Unit]
12		无限运行重复位置	LREAL (正数) [Unit]
13		无限运行重复	0:禁用, 1:使能
14		命令 inposition范围	0, LREAL (正数) [Unit]
15		过范围跟踪错误值	0, LREAL (正数) [Unit]
16		当前位置补偿量	0, LREAL (正数)
17		当前速度滤波时间常量	0 ~ 100
18		错误复位监控时间	0 ~ 1,000 [ms]
19		速度控制SW 限制	0:不检测, 1:检测
20		跟踪误差 level	0:警告, 1:警报
21		JOG 高速	LREAL (正数) [Unit/s] (JOG 低速 ~ 速度限制)

第7 程序

编号	参数	项目	设置
22	扩展参数	JOG 低速	LREAL (正数) [Unit/s] (< JOG 高速)
23		JOG加速	LREAL (正数) [Unit/s ²]
24		JOG减速	LREAL (正数) [Unit/s ²]
25		JOG加速度	LREAL (正数) [Unit/s ³]
26		覆盖模式	0: 百分比 1:设置值
27	NC 参数	到达主轴旋转指令速度的识别范围	0~100%
28		到达主轴0速度的识别RPM	0~100rpm

(7) 伺服参数读命令读输入变量

: 输入变量执行伺服参数读取 (LS_ReadSDO) 运动功能块。

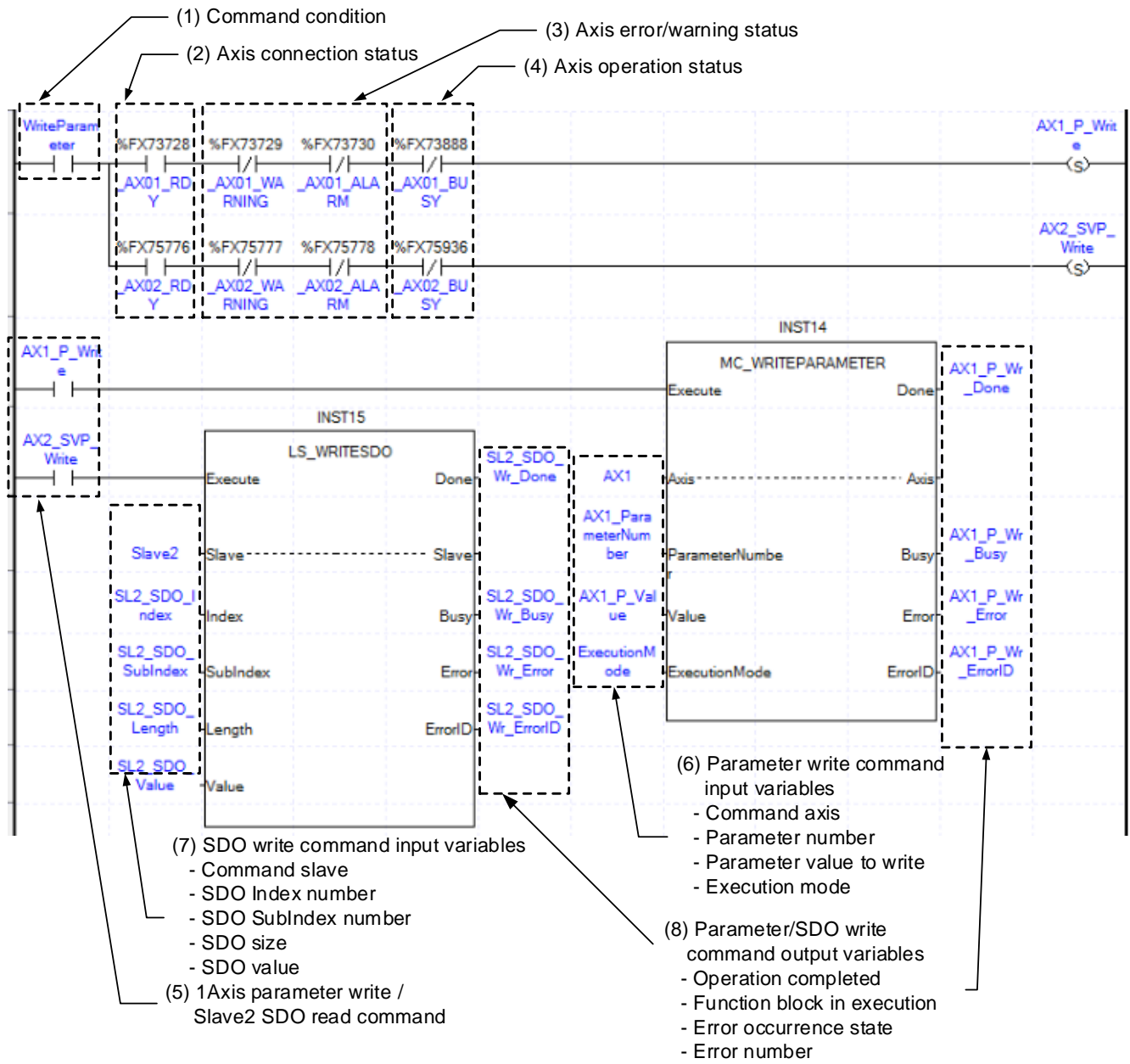
- 命令轴: 设置执行运动功能块的轴。
- 伺服参数索引编号, 子索引编号, 大小: 设置读取伺服参数的值。参考伺服驱动器索引编号, 子索引编号和伺服参数大小的指令手册。

(8) 参数读/伺服参数读取命令输出变量

: 当执行参数读(MC_ReadParameter) 和伺服参数读(LS_ReadSDO) 运动功能块时, 变量存放生成的输出值。

- 运行完成: 如果读参数值和伺服参数, 则为 On。
- 功能块执行: 当执行运动功能块时, 则为 On, 运行完成为 On 时, 变为 Off。
- 错误发生状态: 当运动功能块执行时发生错误, 则为 On。
- 错误编号: 如果发生错误, 生成相应的错误编号。
- 读参数值/读伺服参数值: 执行运动功能块读参数值和伺服参数。

■ 参数写入



第7 程序

(1) 命令条件

:写轴的参数和伺服参数的条件。

(2) 轴连接状态标志

:如果运行的轴连接到运动控制模块, EtherCAT 通信连接的动控制模块正常执行, 则为 On。

(3) 轴错误/报警状态标志

: 如果轴上没有错误和报警, 则为 On。

(4) 轴运行状态标志

: 如果轴在运行, 则为 On。

(5) 1-轴参数写/2-轴伺服参数写命令

: 在示例程序中, 在以下条件下, 1-轴执行参数写(MC_WriteParameter) 运动功能块, 2-轴执行伺服参数写 (LS_WriteServoParameter) 运动功能块。

- 写参数条件为 On

- 轴连接正常.

- 无错误和报警.

- 非运行

执行运动功能块的条件取决于系统。

(6) 参数写命令输入变量

: 输入变量执行参数写入(MC_WriteParameter) 运动功能块。

- 命令轴: 设置执行轴的运动功能块。

- 参数号: 设置运动功能块写参数号。

- 参数值写入: 设置写入的参数值。

- 执行模式: 当参数写入时, 指定时间点。如果设置为 0, 执行运动功能块改变参数值。如果设置为 1, 和缓存模式的“Buffered” 改变时间点相同。(参考 6.1.4 BufferMode)

(7) 伺服参数写命令输入变量

: 输入变量执行 Servo 参数写 (LS_WriteSDO) 运动功能块。

- 命令轴: 设置执行运动功能块的轴。

-伺服参数索引编号, 子索引编号, 大小: 设置读取伺服参数的值。参考伺服驱动器索引编号, 子索引编号和伺服参数大小的指令手册。

- 伺服参数写入值: 设置写入伺服参数值。

-执行模式: 当参数写入时, 指定时间点。如果设置为 0, 执行运动功能块改变参数值。如果设置为 1, 和缓存模式的“Buffered” 改变时间点相同。(参考 6.1.4 BufferMode)

(8) 参数写/Servo 参数写命令输出变量

:当执行参数写 (MC_WriteParameter) 和 Servo 参数写 (LS_WriteSDO) 运动功能块时, 变量存放生成的输出值。

- 运行完成: 如果参数值和伺服参数值写入, 则为 On。

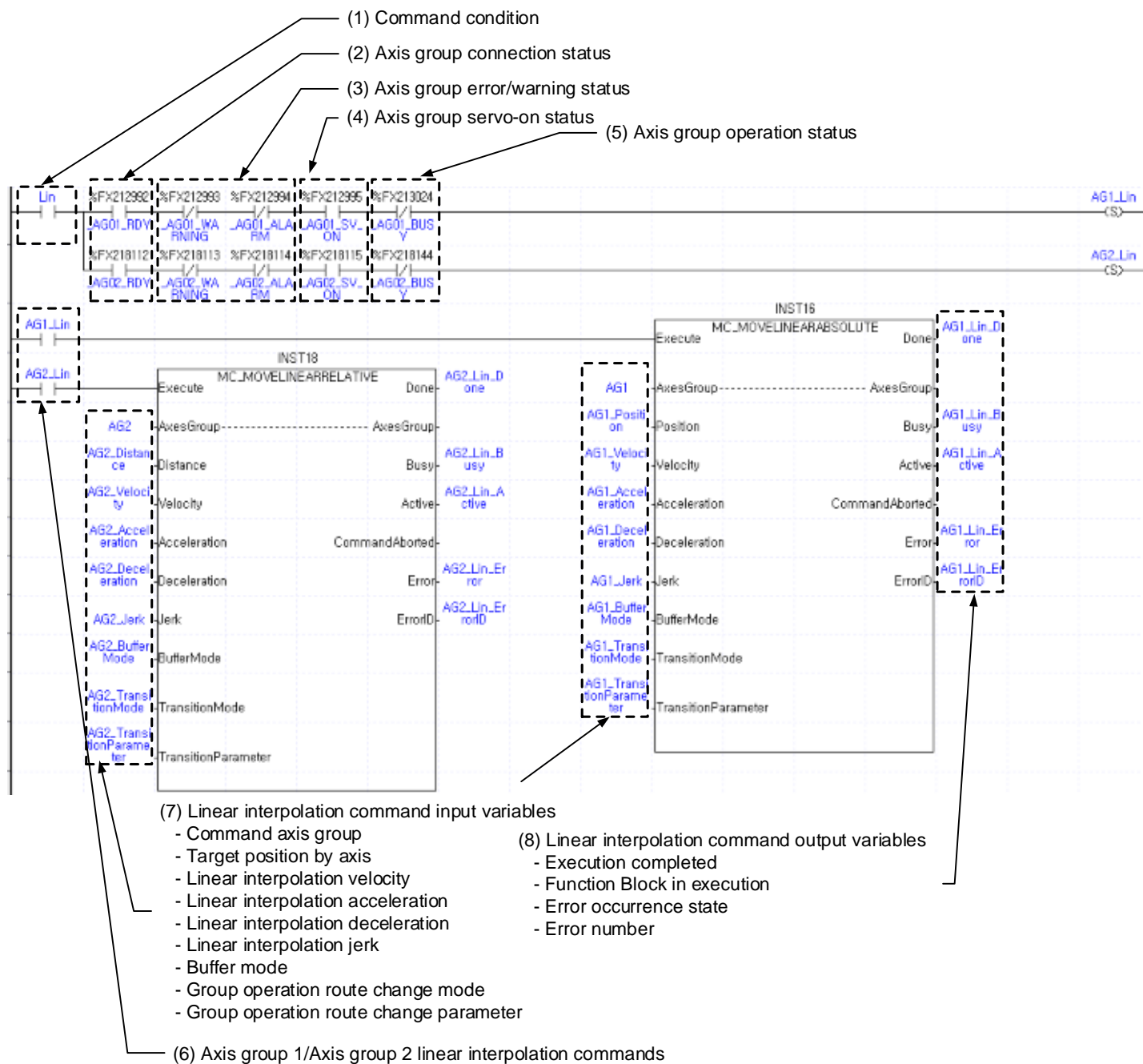
- 功能块执行:当执行运动功能块时, 则为 On, 运行完成为 On 时, 变为 Off。

- 错误发生状态: 当运动功能块执行时发生错误, 则为 On。关于错误编号, 如果发生错误, 根据错误生成错误编号。

7.4 多轴运行程序

7.4.1 线性插值运行

以下示例程序的轴设置相同组运行线性插补。在示例程序中，1-轴和 2-轴假设包含相同的轴组。参考“7.4.5 轴组处理”示例程序包括从轴组或删除轴。

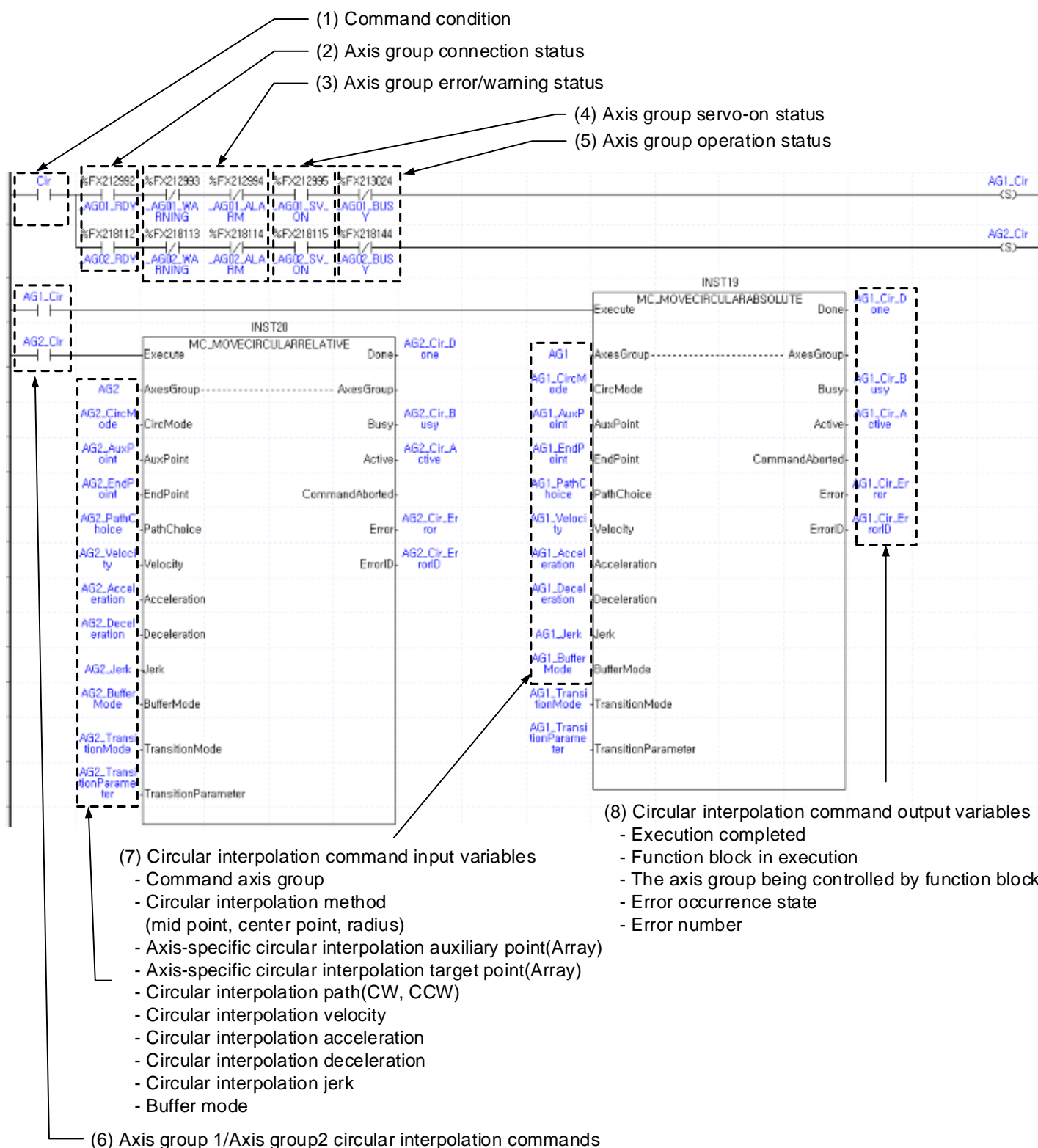


第7 程序

- (1) 命令条件
：给轴组线性插补命令的条件。
- (2) 轴组连接状态标志
：如果运行的轴连接到运动控制模块，EtherCAT 通信连接的动控制模块正常执行，则为 On。
- (3) 轴错误/报警状态标志
：如果轴上没有错误和报警，则为 On。
- (4) 轴组 servo-on 状态
：如果轴包括轴组为 Servo-on 状态，则为 On。
- (5) 轴组运行状态标志
：如果轴组的轴为运行，则为 On。
- (6) 轴组 1 绝对位置线性插补/轴组 2 相对位置线性插补指令
：在示例程序中，在下列条件中，轴组 1 执行绝对位置线性插补运行 (MC_MoveLinearAbsolute)，轴组 2 执行相对位置线性插补运行 (MC_MoveLinearRelative) 运动功能块。
 - 线性插补运行条件为 On.
 - 轴包括轴组正常连接。
 - 无错误和报警.
 - 轴组的轴非运行.执行运动功能块条件取决于系统。
- (7) 线性插补命令输入变量
：输入变量执行绝对位置线性插补运行 (MC_MoveLinearAbsolute)，相对位置线性插补运行 (MC_MoveLinearRelative) 运动功能块。
 - 命令轴组: 设置执行轴组的运动功能块。
 - 轴目标位置: 设置数组变量，按顺序设置包括轴组的轴线性插补运行目标位置。
 - 线性插补速度: 当速度参考插补速度时，设置执行线性插补的目标速度。
 - 当执行线性插补时，应用设置的线性插补加速，减速，jerk。
 - 缓存模式: 当运动功能块执行时，设置时间点。用于设置是否立即执行或当前命令完成后执行。更多细节请参考功能块，“6.1.4 缓存模式输入”。
 - 组运行路径改变模式和族运行路径改变参数: 当给线性插补命令时，指定轴组连接运行已存命令描述的路径。参考“6.1.6 组运行路径改变设置”。
- (8) 线性插补命令输出变量
：当绝对位置线性插补运行 (MC_MoveLinearAbsolute) 和相对位置线性插补运行运动功能块执行 (MC_MoveLinearRelative) 时，变量存放生成的输出值。
 - 执行完成: 当功能块执行完成，则为 On。
 - 功能块执行: 当执行运动功能块时，则为 On，运行完成为 On 时，变为 Off。
 - 错误发生状态: 如果运动功能块在执行时发生错误，则为 On。
 - 错误编号: 发生错误时，根据错误生成的错误编号。
 - 更多关于运动功能块输出细节，请参考“6.1.3 基本 I/O 变量”中的“Edge 运行运动功能块”。

7.4.2 圆弧插值运行

示例程序是设置相同组的轴运行圆弧插补运行。在示例程序中，1-轴和 2-轴假设包含相同的轴组。参考“7.4.5 轴组处理”示例程序包括从轴组或删除轴。



(1) 命令条件

: 给轴组圆弧插补命令的条件。

(2) 轴组连接状态标志

: 如果运行的轴连接到运动控制模块, EtherCAT 通信连接的动控制模块正常执行, 则为 On。

(3) 轴错误/报警状态标志

: 如果轴上没有错误和报警, 则为 On。

(4) 轴组 servo-on 状态

: 如果轴包括轴组为 Servo-on 状态, 则为 On。

(5) 轴组运行状态标志

: 如果轴组的轴为运行, 则为 On。

(6) 轴组 1 绝对位置圆弧插补/轴组 2 相对位置圆弧插补命令

: 在示例程序中, 在以下条件中, 轴组 1 执行绝对位置圆弧插补运行(MC_MoveCircularAbsolute), 轴组 2 执行相对位置圆弧插补 (MC_MoveCircularRelative) 运动功能块。

- 圆弧插补运行条件为 On。
- 轴包括轴组正常连接。
- 无错误和报警。
- 轴组的轴非运行。

执行运动功能块条件取决于系统。

(7) 圆弧插补命令输入变量

: 输入变量执行绝对位置圆弧插补运行 (MC_MoveCircularAbsolute) 和相对位置圆弧插补运行 (MC_MoveCircularRelative) 运动功能块。

- 命令轴组: 设置执行轴组的运动功能块。
- 轴目标位置: 设置数组变量, 按顺序设置轴包括轴组的线性插补运行目标位置。
- 圆弧插补方式: 通过选择中间点, 中心点和半径方式设置执行圆弧插补的方式。
- 轴指定圆弧插补辅助点: 按照轴包括轴组的顺序设置圆弧插补所需的辅助点。
- 轴指定圆弧插补目标点: 按照轴包括轴组的顺序设置目标位置。
- 圆弧插补速度: 当速度参考插补速度时, 设置执行圆弧插补的目标速度。
- 当执行圆弧插补时, 设置圆弧插补加速, 减速, jerk。
- 缓存模式: 当运动功能块执行时, 设置时间点。用于设置是否立即执行或当前命令完成后执行。更多细节请参考功能块, “6.1.4 缓存模式输入”。

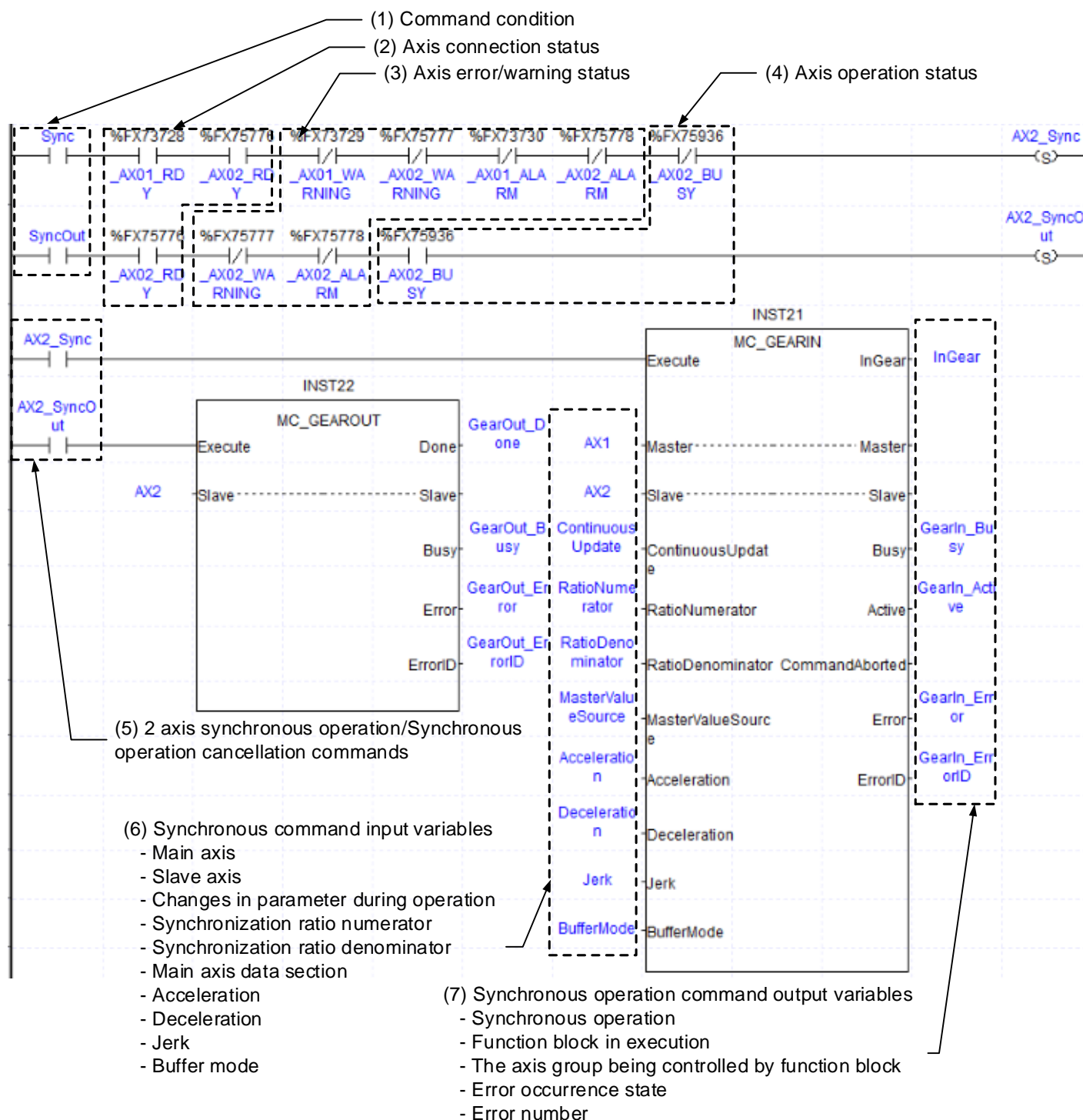
(8) 圆弧插补命令输出变量

: 当执行绝对位置圆弧插补运行 (MC_MoveCircularAbsolute) 和相对位置圆弧插补运行 (MC_MoveCircularRelative) 运动功能块时, 变量存放生成的输出值。

- 执行完成: 当运动功能块执行完成时, 则为 On。
- 功能块执行: 当执行运动功能块时, 则为 On, 运行完成为 On 时, 变为 Off。
- 功能块控制轴组: 当运动功能块控制轴组时, 则为 On。
- 错误发生状态: 如果运动功能块在执行时发生错误, 则为 On。
- 错误编号: 如果发生错误, 根据错误原因生成对应的错误编号。
- 更多关于运动功能块的输出细节, 参考“6.1.3 基本 I/O 变量”的“Edge 运动指令”。

7.4.3 同步运行

以下同步运行示例程序是从轴按主轴设置的同步率移动。



第7 程序

- (1) 命令条件
: 给同步运行/同步运行取消命令的条件。
- (2) 轴连接状态标志
: 如果运行的轴连接到运动控制模块, EtherCAT 通信连接的动控制模块正常执行, 则为 On。
- (3) 轴错误/报警状态标志
: 如果轴有错误和报警发生, 则为 On。
- (4) 轴运行状态标志
: 如果轴在运行, 则为 On。
- (5) 2 轴同步运行/同步运行取消命令
: 在示例程序中, 在以下条件下, 执行电子齿轮取消 (MC_GearOut) 运动功能块。
 - 同步运行条件为 On。
 - 轴和主轴正常连接。
 - 无错误和报警。
 - 轴为非运行。并且, 在以下条件下, 执行电子齿轮取消 (MC_GearOut) 运动功能块。
 - 同步运行取消条件为 On。
 - 轴连接正常。
 - 无错误和报警。
 - 轴运行。执行运动功能块条件取决于系统。
- (6) 同步命令输入变量
: 输入变量执行电子齿轮运行 (MC_GearIn) 运动功能块。
 - 主轴: 设置从轴同步运行。
 - 从轴: 设置同步运行执行的轴。
 - 运行中改变参数: 设置在运动功能块输入变量变化后是否应用运行。更多细节, 参考“6.1.5 运动功能块执行中改变参数”。
 - 同步率分子: 在主轴的同步运行中设置运行同步率的分子。
 - 同步率分母: 在主轴的同步运行中设置运行同步率的分母。
 - Gear 运行状态从轴速度设置如下。从轴在 gear 运行(InGear)状态设置速度如下。

从轴速度 = 主轴速度 x (同步率分子/同步分母)

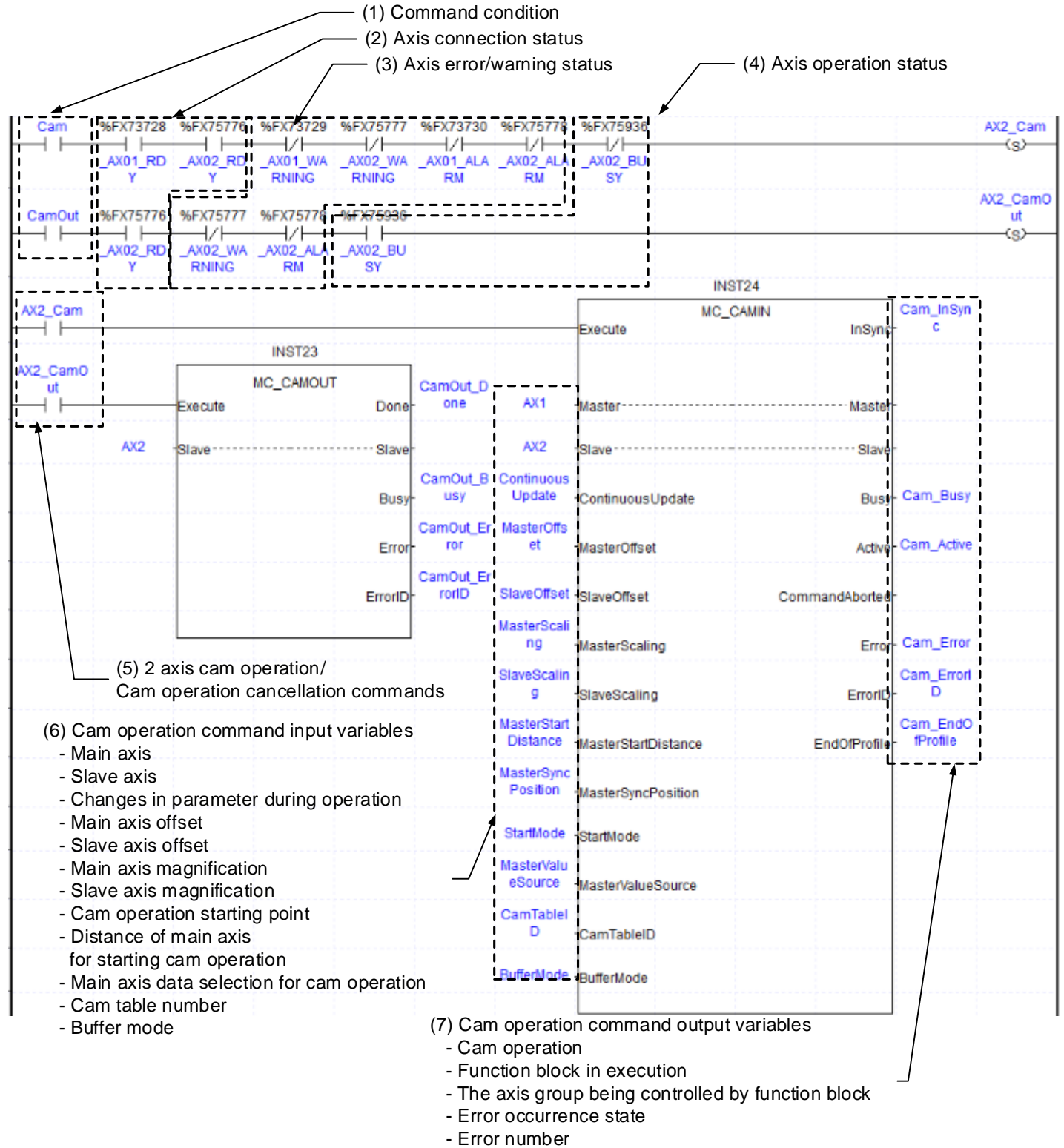
- 主轴数据选择: 选择主轴数据设置的命令速度或当前速度。
如果设置指令速度, 主轴基于运动控制模块速度的计算到达同步。
如果设置当前速度, 通过通信使用主轴伺服驱动器传送的速度数据达到同步。
- 加速, 减速, jerk: 在同步运行中设置值。
- 缓存模式: 当运动功能块执行时, 设置时间点。用于设置是否立即执行或当前命令完成后执行。更多细节请参考功能块, “6.1.4 缓存模式输入”。

(7) 同步运行命令输出变量

- : 当电子齿轮运行(MC_GearIn)执行时, 变量用于存放生成的输出值。
- 同步运行: 当从轴在执行运动功能块后正常和主轴同步, 则为 On。
- 功能块执行: 当执行运动功能块时, 则为 On, 运行完成为 On 时, 变为 Off。
- 轴组被功能块控制。运动功能块控制轴组时, 则为 On。
- 错误发生状态: 如果运动功能块在执行时发生错误, 则为 On。
- 错误编号: 如果发生错误, 根据错误生成相应的错误编号。
- 关于运动功能块的输出细节, 参考“6.1.3 基本 I/O 变量”的“Edge 运动指令”。

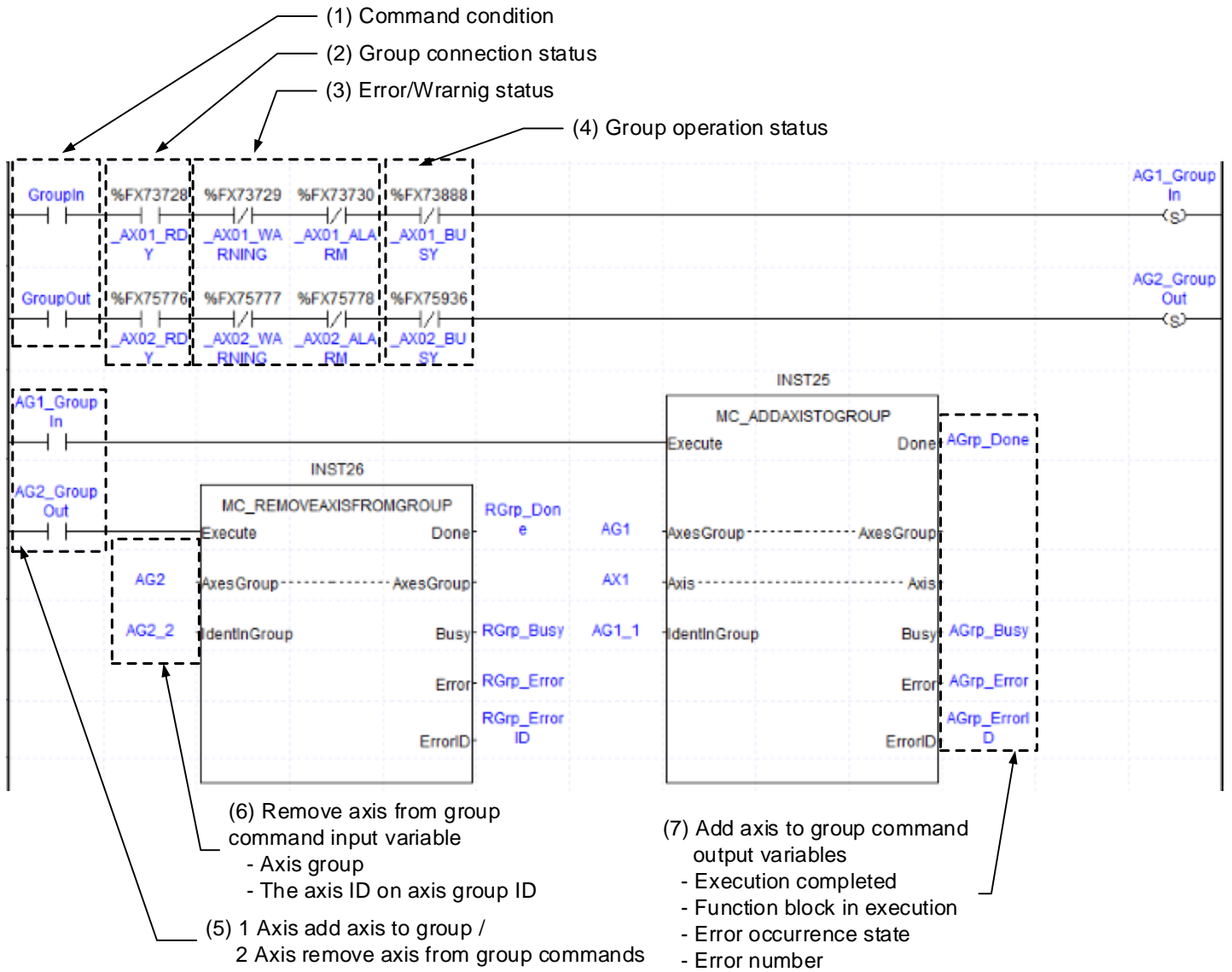
7.4.4CAM 运行

以下关于 Cam 运行示例程序，基于从轴设置的 cam (CAM)轮廓同步移动。



- (1) 命令条件
: 给轴 Cam 运行/Cam 运行取消命令的条件。
- (2) 轴连接状态标志
: 如果运行的轴连接到运动控制模块, EtherCAT 通信连接的动控制模块正常执行, 则为 On。
- (3) 轴错误/报警状态标志
: 如果轴上没有错误和报警, 则为 On。
- (4) 轴运行状态标志: 如果轴在运行, 则为 On。
- (5) 2-轴 Cam 运行/Cam 运行取消指令
: 在示例程序中, 在以下条件中执行 Cam 运行 (MC_CamIn) 运动功能块。
- Cam 运行条件为 On。
- 轴和主轴正常连接。
- 无错误和报警。
- 轴非运行。
另外, 在以下条件中执行 Cam 运行取消 (MC_CamOut) 运动功能块。
- Cam 运行取消条件为 On。
- 轴连接正常
- 无错误和报警
- 轴运行
执行运动功能块条件取决于系统。
- (6) Cam 运行命令输入变量
: 输入变量执行 Cam 运行 (MC_CamIn) 运动功能块。
- 主轴: 设置 Cam 运行主轴。
- 从轴: 设置执行 Cam 运行的轴。
- 运行中改变参数: 设置在运动功能块输入变量变化后是否应用运行。更多细节, 参考“6.1.5 运动功能块执行中改变参数”。
- 主轴补偿: 当应用 cam table 数据时, 设置主轴数据的补偿值。
- 从轴补偿: 当应用 cam table 数据时, 设置从轴数据的补偿值。
- 主轴放大: 当应用 cam table 数据时, 设置主轴数据的放大倍数。
- 从轴放大: 当应用 cam table 数据时, 设置从轴数据的放大倍数。
- Cam 运行开始点: 设置主轴位置的 cam table 开始点。
- 主轴开始 Cam 运行距离: 设置主轴实际 Cam 运行开始的距离。
- Cam 运行主轴数据选择: 在主轴命令位置和主轴当前位置选择主轴数据 Cam 运行数据。
- Cam table 号: 设置 Cam 运行的 cam data 编号。
- 更多 Cam 运行命令输入变量, 参考“6.4.1 Cam 运行 (MC_CamIn)”。
- 运行中改变参数: 设置在运动功能块输入变量变化后是否应用运行。更多细节, 参考“6.1.5 运动功能块执行中改变参数”。
- 缓存模式: 当运动功能块执行时, 设置时间点。用于设置是否立即执行或当前命令完成后执行。更多细节请参考功能块, “6.1.4 缓存模式输入”。
- (7) Cam 运行命令输出变量
: 当执行 Cam 运行 (MC_CamIn) 运动功能块时, 变量存放生成的输出值。
- Cam 运行: 当执行运动功能块后, 从轴根据 cam 数据同步于主轴时为 On。
- 功能块执行: 当执行运动功能块时, 则为 On, 运行完成为 On 时, 变为 Off。
- 功能块轴控制执行: 如果运动功能块控制轴, 控制轴, 则为 On。
- 错误发生状态: 如果运动功能块在执行时发生错误, 则为 On。
- 错误编号: 发生错误时, 根据错误生成的错误编号。
- 更多关于运动功能块输出, 参考“6.1.3 基本 I/O 变量”的“边沿运动指令”。

7.4.5 轴组处理



(1) 命令条件

: 给增加轴到组/从组删除轴的命令条件。

(2) 轴连接状态标志

:如果运行的轴连接到运动控制模块, EtherCAT 通信连接的动控制模块正常执行, 则为 On。

(3) 轴错误/报警状态标志

:如果轴发生错误和报警, 则为 On。

(4) 轴运行状态标志

: 如果轴在运行, 则为 On。

(5) 1-轴增加轴到组/2-轴从组删除轴命令

: 在示例程序中, 在以下条件中, 执行增加轴到组 (MC_AddAxisToGroup) 运动功能块。

- 增加轴到组条件为 On.
- 轴连接正常.
- 无错误和报警.
- 轴非运行.

另外, 在以下条件中执行组轴排除(MC_RemoveAxisFromGroup) 运动功能块。

- 从组删除轴条件为 On.
- 轴连接正常.
- 无错误和报警.
- 轴非运行.

执行运动功能块条件取决于系统。

(6)从组命令输入变量删除轴

: 变量执行组轴排除(MC_RemoveAxisFromGroup) 运动功能块。

- 轴组: 设置组中排除的轴。
- 轴 ID 位于轴组 ID: 当轴包括在轴组中, 设置其所属 ID 值。

(7)增加组命令输出变量轴

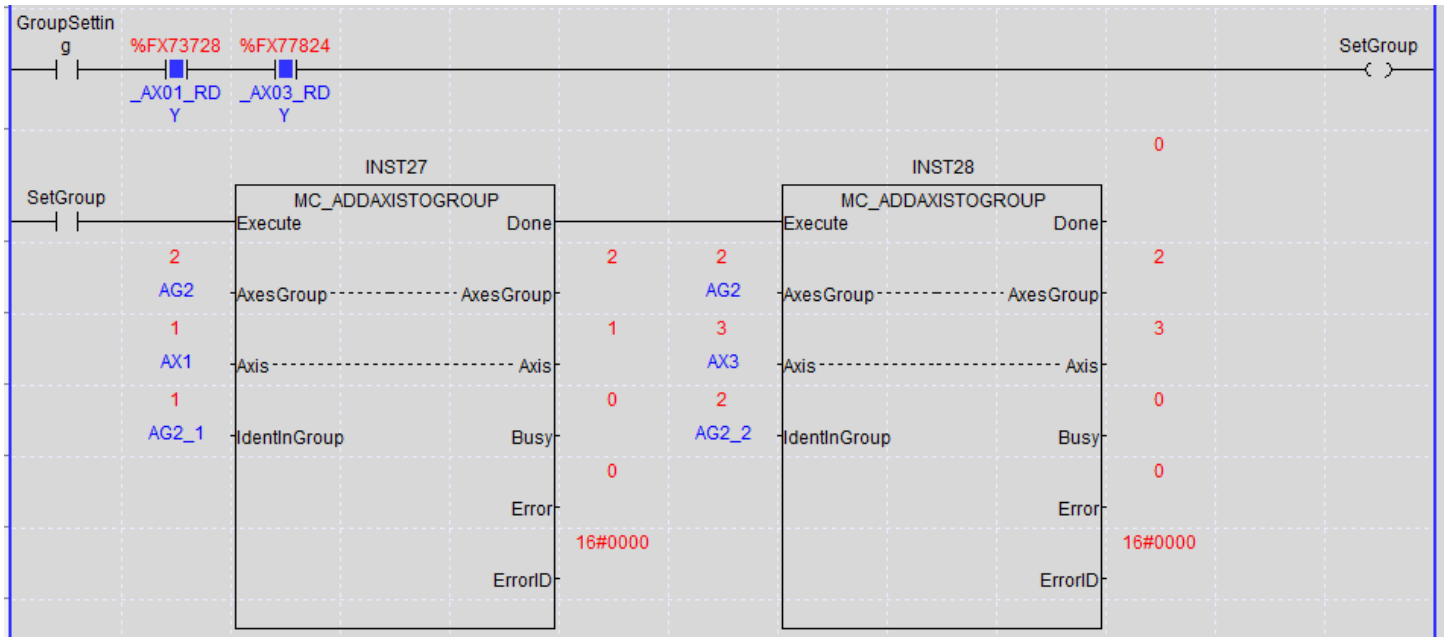
: 当执行增加轴到组运动功能块时, 变量用于存放生成的输出值。

- 执行完成: 当运动功能块正常执行, 则为 On。
- 功能块执行: 如果运动功能块执行, 则为 On。如果执行完成, 则为 Off。
- 错误发生状态:如果运动功能块在执行时发生错误, 则为 On。
- 错误编号: 发生错误时, 根据错误生成的错误编号。
- 更多关于运动功能块的输出,参考“6.1.3 基本 I/O 变量”的“Edge 运动指令”。

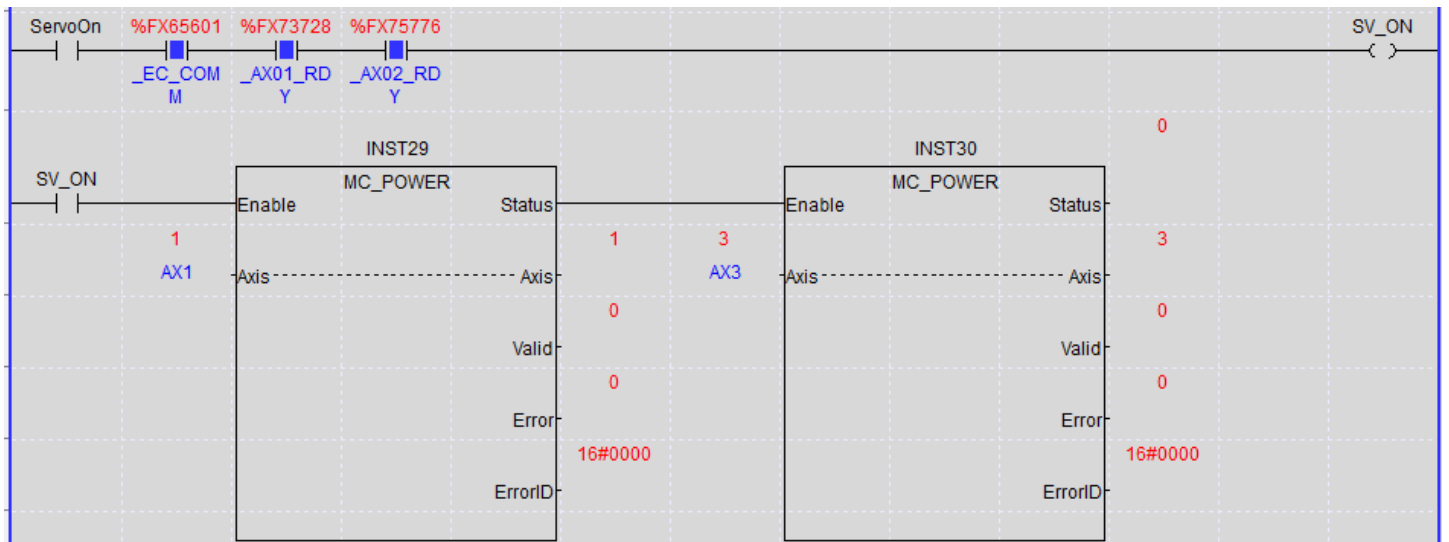
第7 程序

7.4.6 轴组运行实例

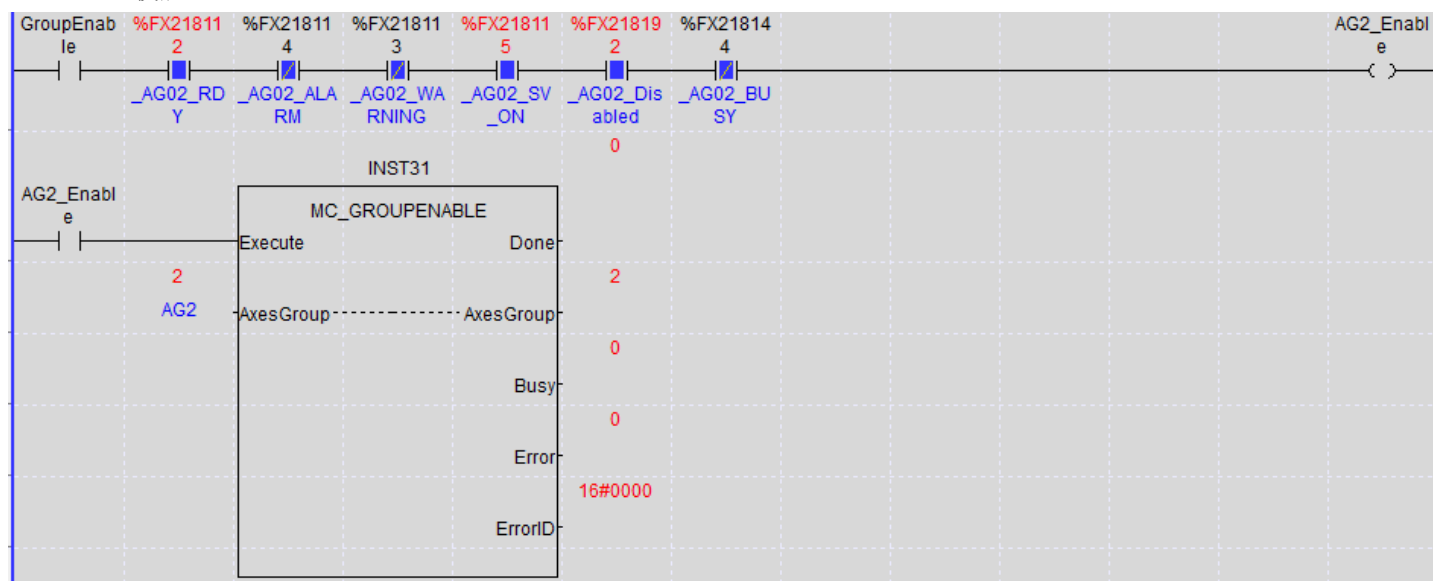
1. 组设置



2. 伺服 On



3. 组使能

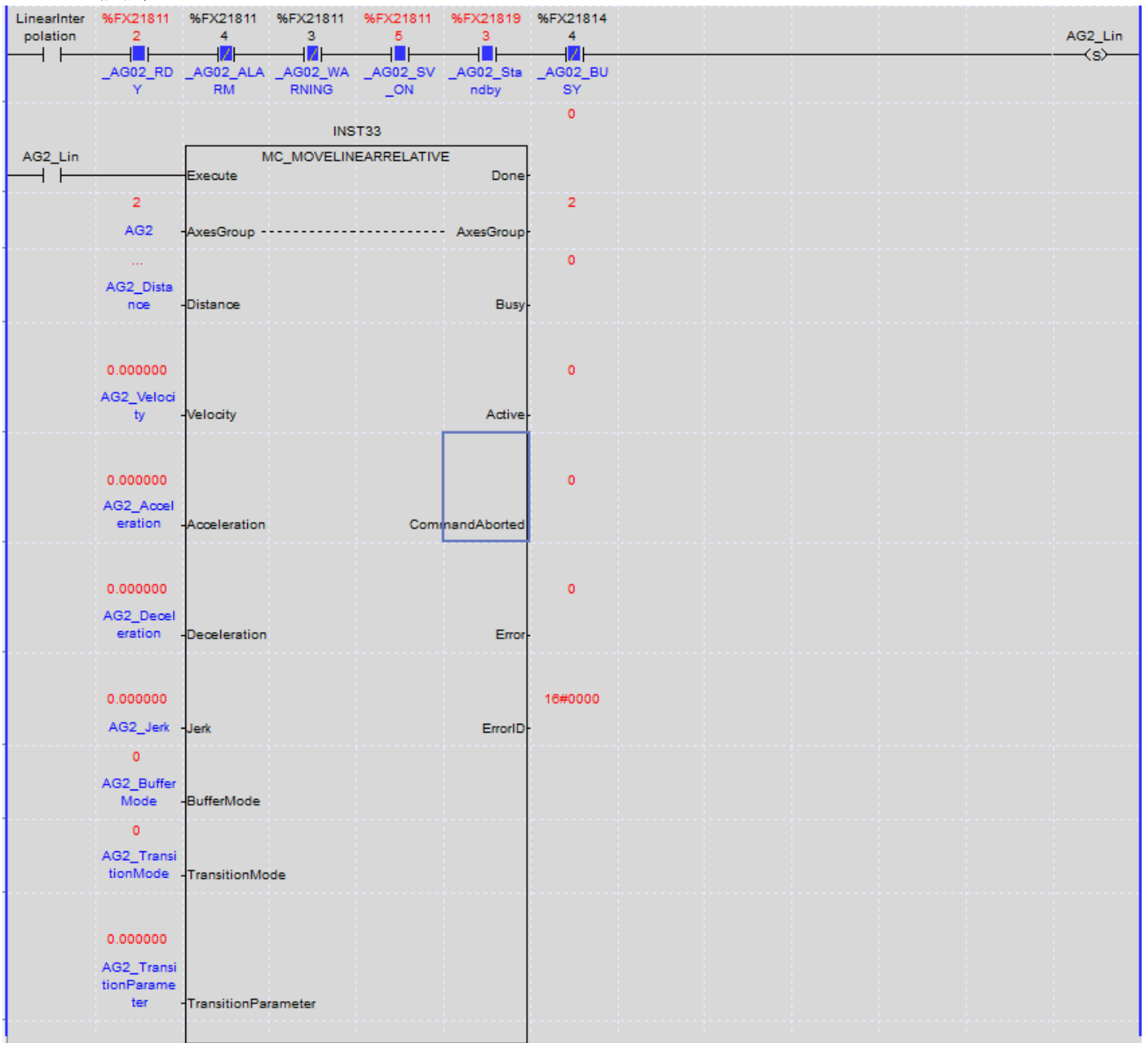


4. 组原点返回

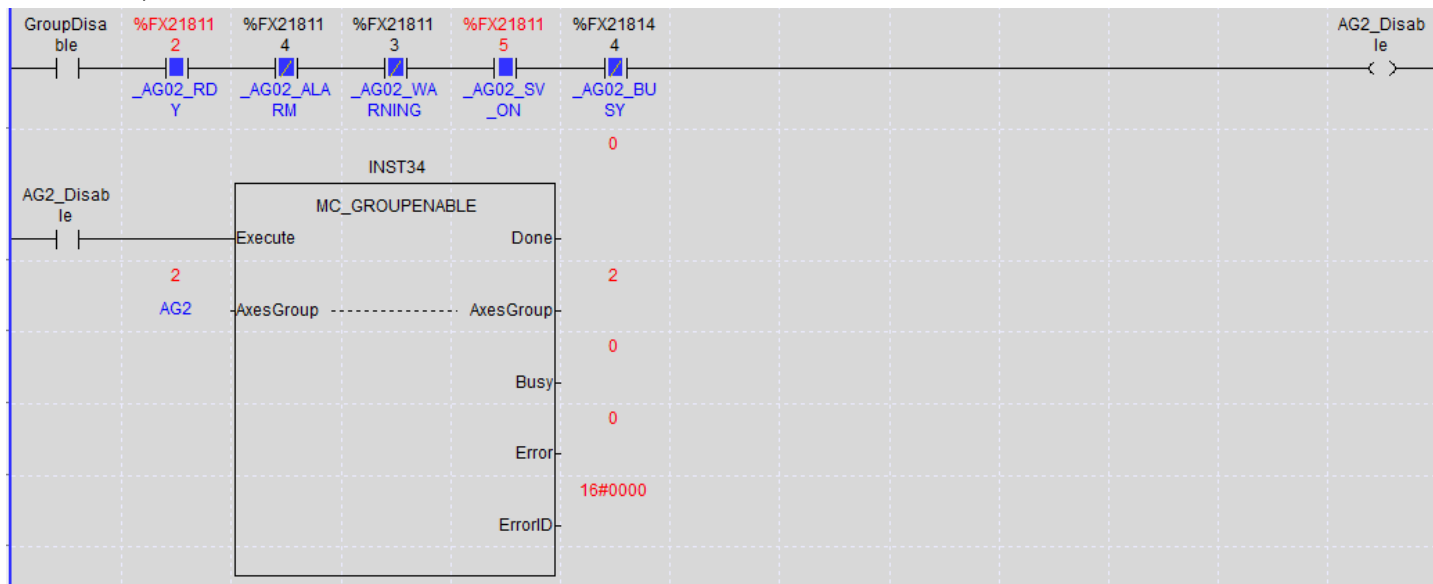


第7 程序

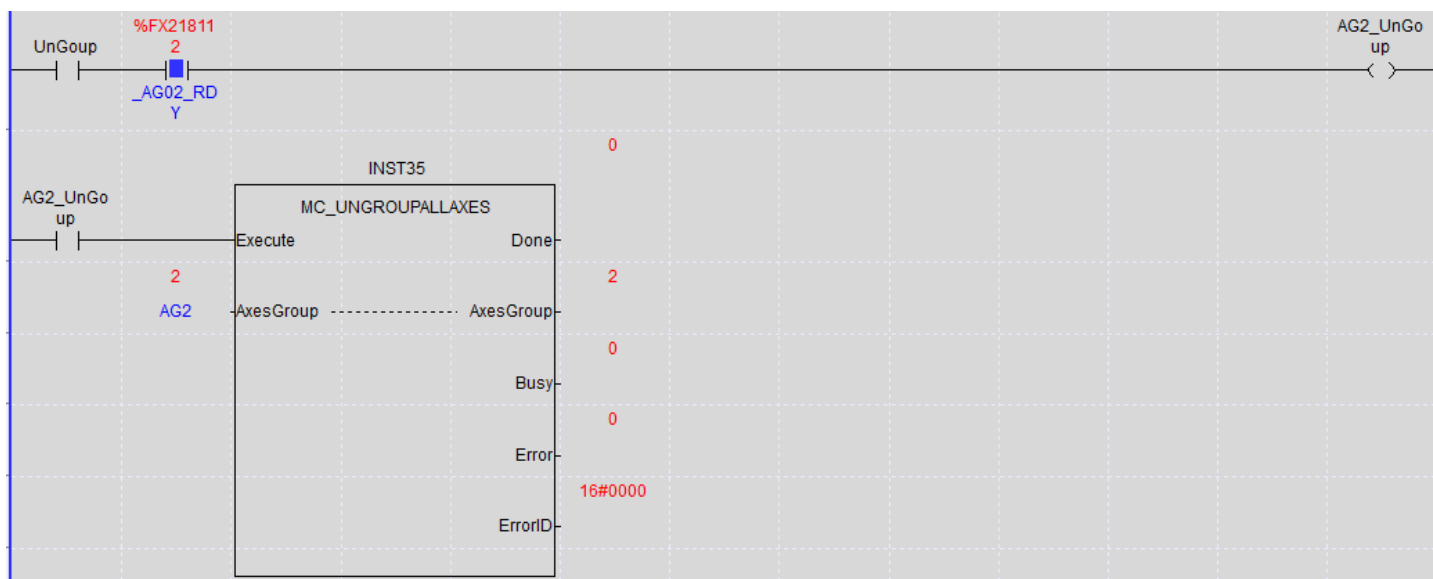
5. 线性插值



6. 组禁止

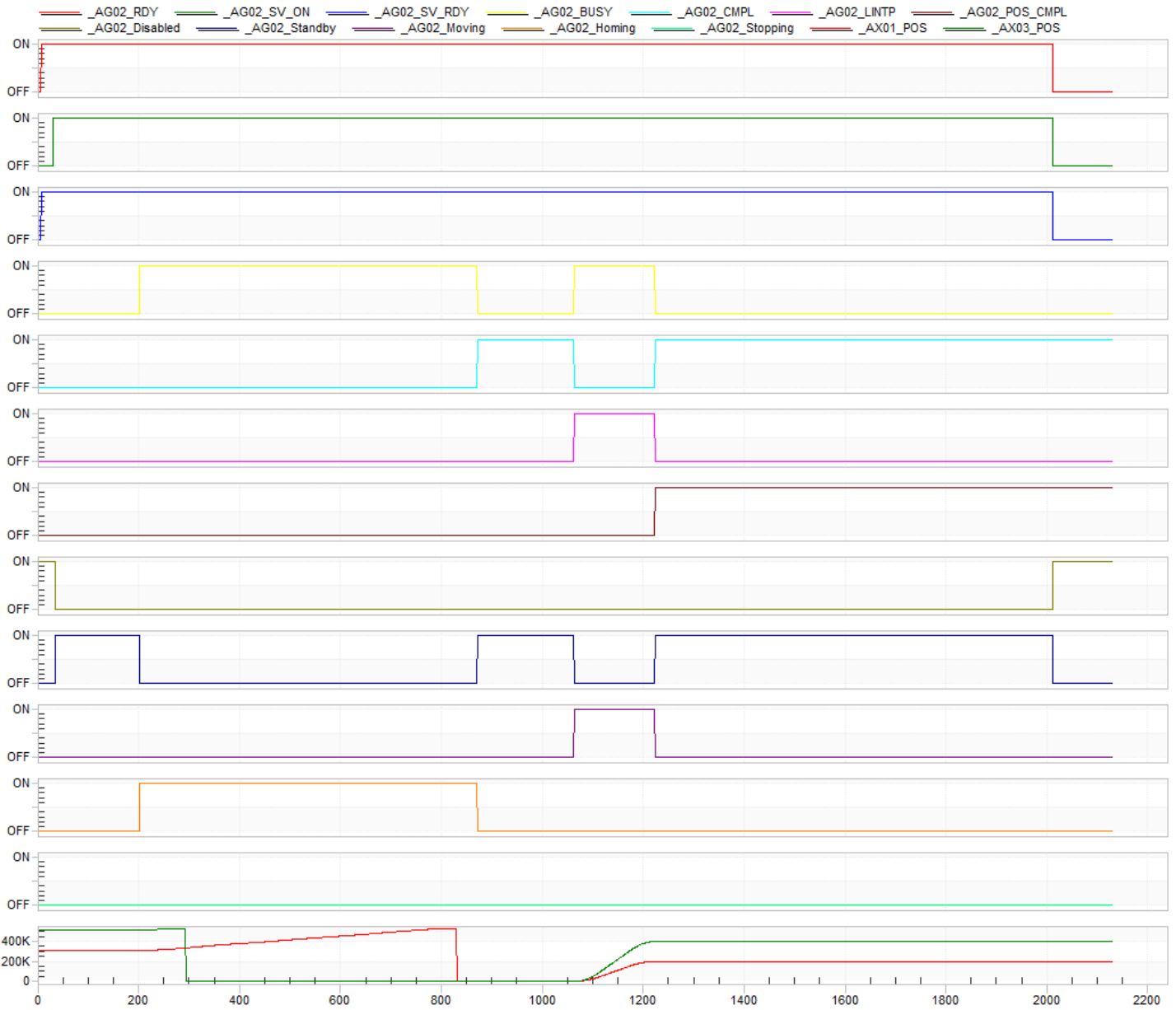


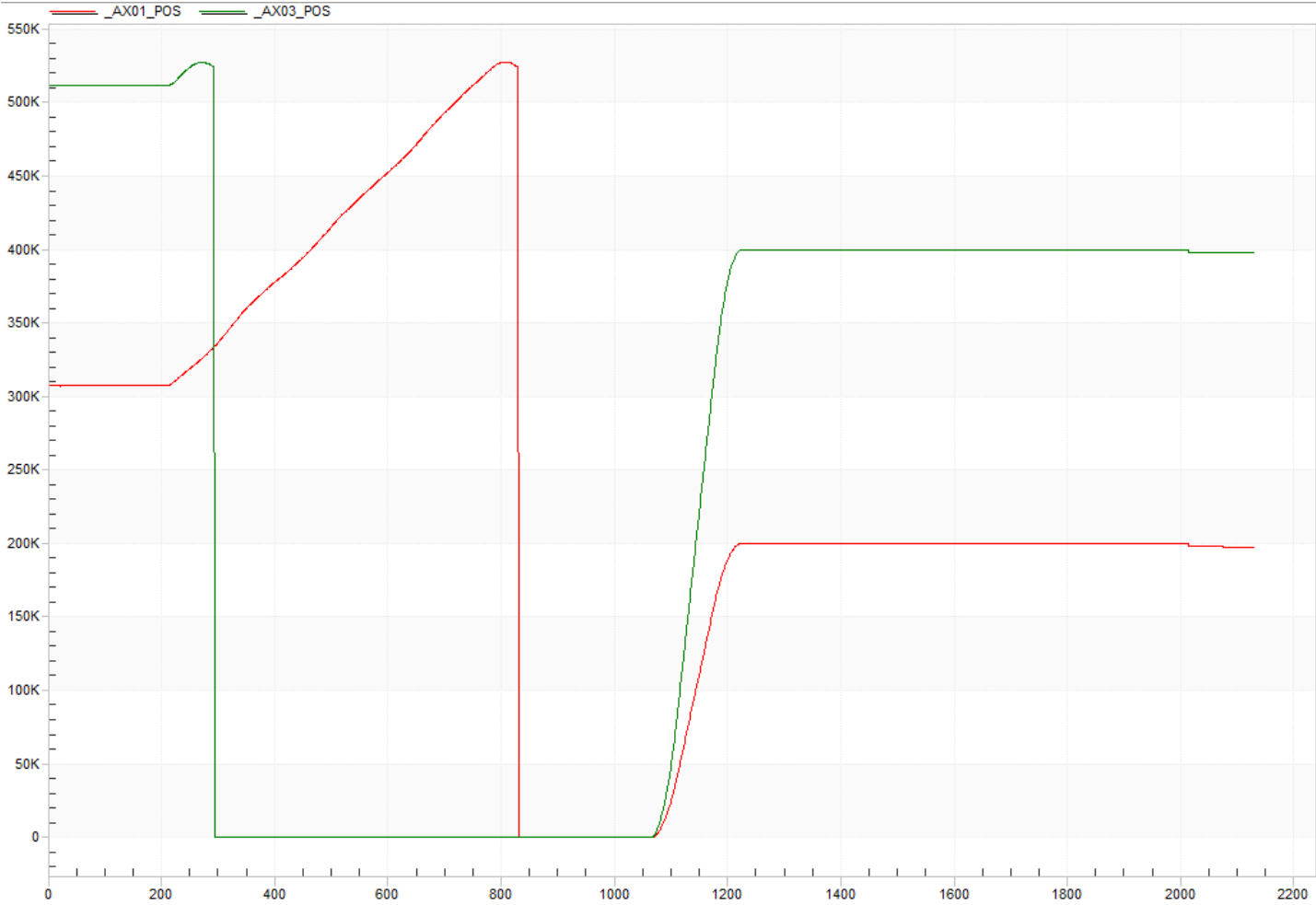
7. 取消组



第7 程序

8. 时序图





7.5 I/O处理程序

运动控制模块内部有 8 点输入和 16 点输出，可以扩展 EtherCAT 输入输出模块增加输入输出点。EtherCAT 输入和 EtherCAT 输出模块可安装扩展达 64 站，1024 点。

7.5.1 输入信号处理

内部输入信号和信号输入外部输入模块，可以使用运动控制块内部标志位程序。

更多细节，参考“附录 1.标志位”。

7.5.2 输出信号处理

内部输出信号和信号输出外部输出模块，可以使用运动控制块内部标志位程序。

更多细节，参考“附录 1. 标志位”。

第8章运动控制功能

8.1 原点决定

使用运动控制模块的位置控制功能，基于绝对坐标位置控制，先决定原点后再执行命令。绝对坐标的位置值是基于预先决定原点(0位置)的距离。原点决定表示使用绝对坐标控制时，设置机器位置的原点。

8.1.1 原点决定

1. 原点决定方式

以下有两种方法决定机器原点。

(1) 原点返回

使用连接到伺服驱动器的传感器，利用(MC_Home)运动功能块，通过移动机器确定机器原点的方式。当执行原点返回命令时，原点决定变成原点未定状态，返回成功完成后，变为原点决定状态。

(2) 当前位置设置

当使用JOG 运行(LS_Jog)或相对坐标位置控制(MC_MoveRelative) 运动功能块，移动机器到某一位置后，使用当前位置定位设置(MC_Setposition) 运动功能块设定为指定位置。该位置被认为绝对坐标，变为原点决定状态。

轴的原点决定状态等同于运动轴标志 AXxx_HOME_CMOL (%JXxx.67)。(xx: 轴编号)

2. 使用绝对编码器时原点决定

如果在伺服驱动器使用绝对编码器，绝对数据值即使断电后也可通过电池备份保持。当连接到伺服驱动器时，运动控制模块从绝对编码器读取当前位置值和计算绝对坐标位置可继续保持原点决定状态。

最后，如果使用绝对编码器，编码器选择运行基本参数设置为‘1: 绝对编码器’。建立目标决定状态后，即使运动控制模块和伺服驱动器断电，重新上电后，当伺服驱动器连接编码器选择参数‘1: 绝对编码器’，通过计算绝对坐标位置之前的原点决定状态保持。在绝对坐标系统中使用绝对编码器，即使电源off/on 后，绝对坐标位置不使用原点决定可控制。

3. 改变原点未决定状态

在以下情况中，运动控制模块变为原点未定状态前，不能执行绝对位置控制运行。

- (1) 当使用增量编码器时，伺服驱动器电源断电后重新连接。
- (2) 当使用增量编码器时，PLC电源off/on 后，重新连接。
- (3) 当执行原点返回命令后，原点返回未正常完成。

如果原点未定状态如上，原点决定需执行绝对坐标位置控制运行。

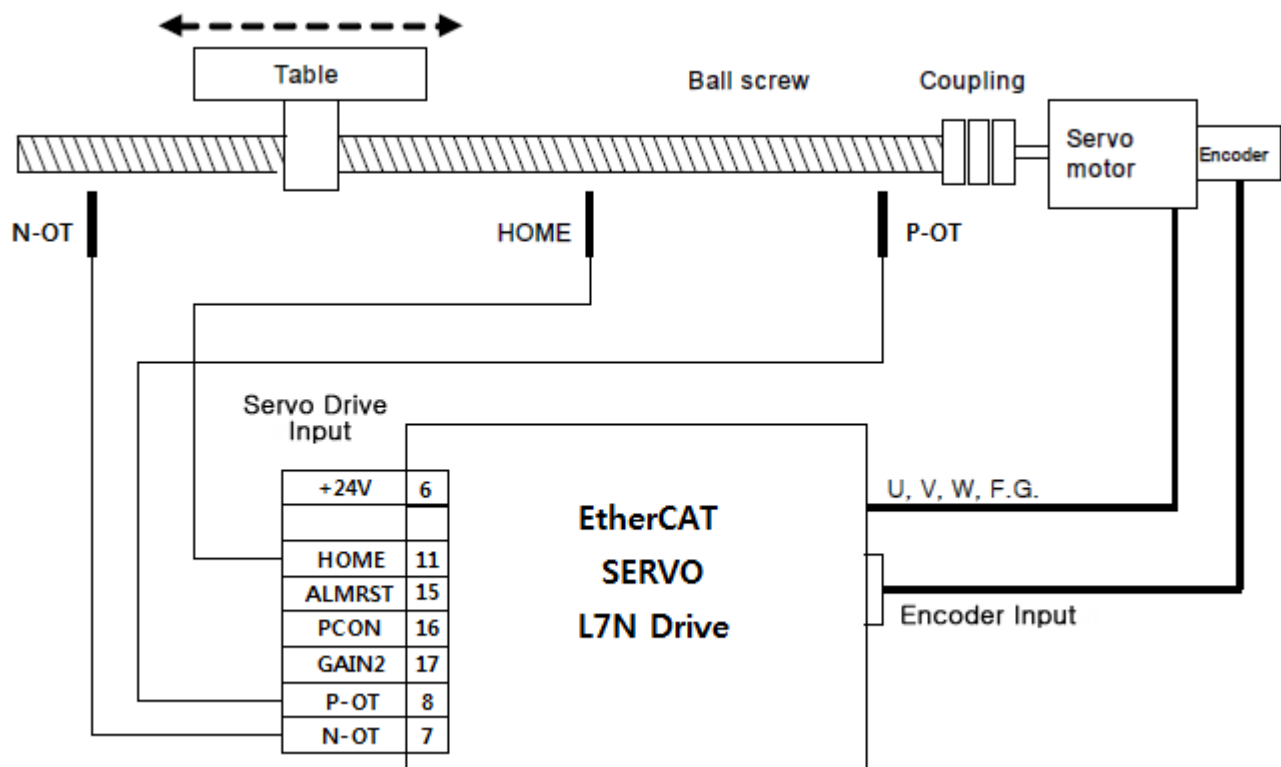
8.1.2 原点返回

1.运行

当机器上电后，执行原点返回确定机器的原点。执行原点返回前，设置伺服驱动器的轴有关原点返回参数。当通过原点返回确定原点位置时，在运动控制运行中，原点检测信号不识别。

通过连接伺服驱动器(EtherCAT CoE支持伺服驱动器)执行原点返回。

典型配线如下。



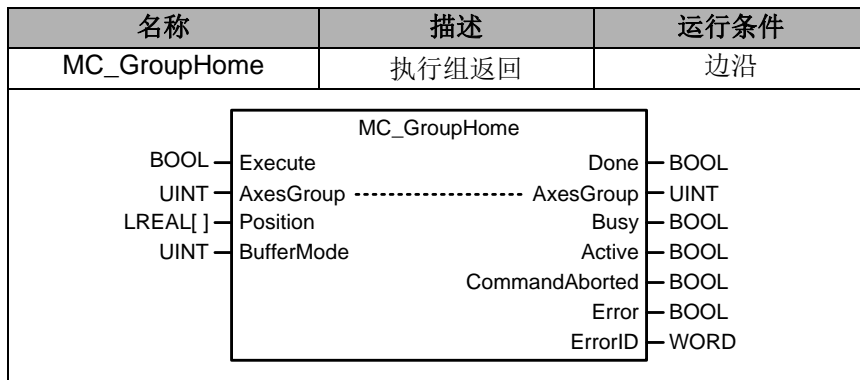
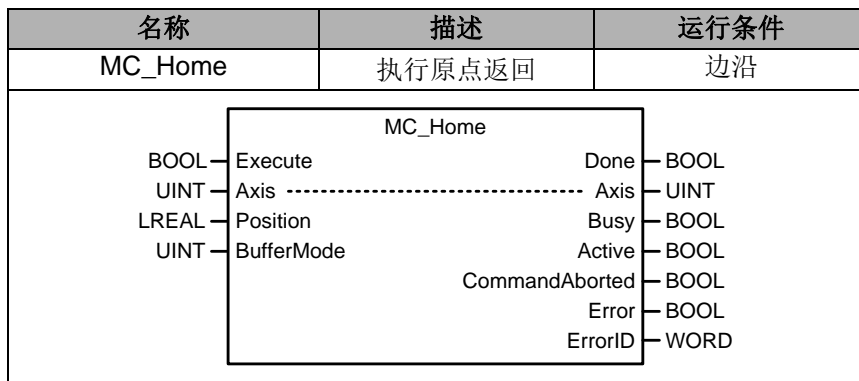
对于执行原点返回，选择一种适用于用户系统原点返回模式(EtherCAT CoE 支持驱动: 参考相关驱动指令手册)的方法。

在运动控制模块中，伺服驱动器开始执行原点返回后的实际操作，返回方式支持伺服驱动器。设置返回前，返回相关参数在轴的伺服参数设置。

■ 返回参数设置示例

Index	Name	Unit	Current Value	Initial Value	Access
<input checked="" type="checkbox"/> 6098	Homing Method	-	0x22	0x22	rw
<input checked="" type="checkbox"/> 6099:00	Homing Speeds	-	0x02	0x02	rw
<input checked="" type="checkbox"/> 6099:01	Speed during search for switch	Vel,Unit	0x000000A0	0x000000A0	rw
<input checked="" type="checkbox"/> 6099:02	Speed during search for zero	Vel,Unit	0x00000020	0x00000020	rw
<input checked="" type="checkbox"/> 609A	Homing Acceleration	Acc,Unit	0x0000C350	0x0000C350	rw

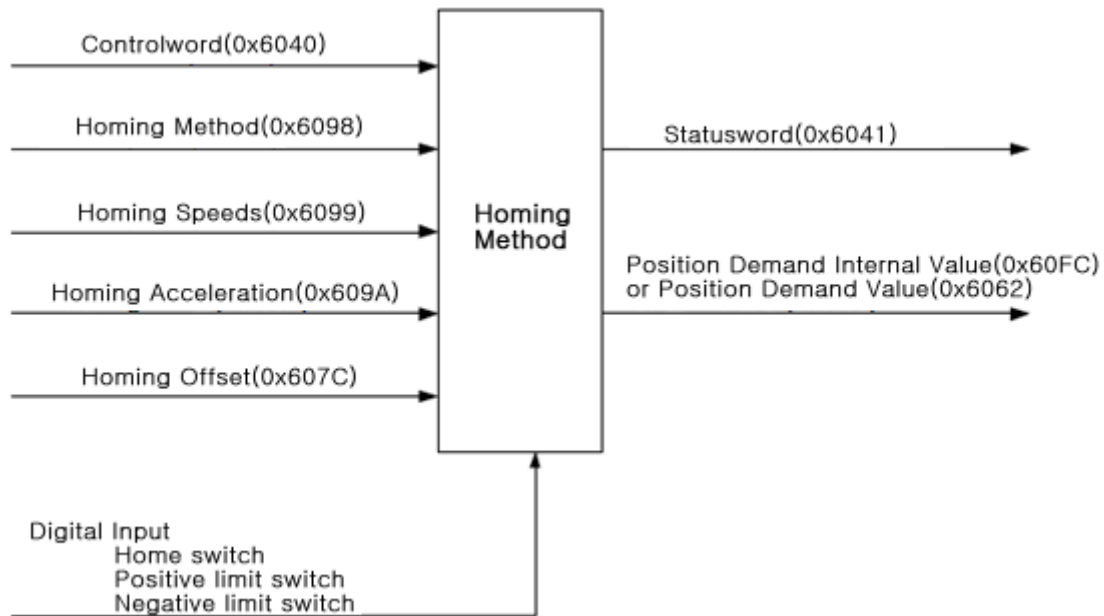
■ 相关运动功能块



第8章 运动控制功能

2.XDL- N系列伺服驱动器返回参数和运行

下图表示原点返回相关XDL-N 系列伺服驱动器参数输入和输出定义。指定速度，加速和返回方式。此时，原点 (Home) offset 得到用户坐标原点系统应用为原点。



(1) 关于原点返回参数

索引	Sub	名称	数据类型	单位
0x6040	-	控制字	UINT	-
0x6041	-	状态字	UINT	-
0x607C	-	返回补偿	DINT	[pls]
0x6098	-	返回方式	SINT	-
0x6099	-	返回速度	-	-
	0	项目编号	USINT	-
	1	搜索切换中速度	UDINT	[pls/s]
	2	搜索 0 中速度	UDINT	[pls/s]
0x607D	-	软件位置限制	-	-
	0	项目编号	USINT	-
	1	最小位置限制	DINT	[pls]
	2	最大位置限制	DINT	[pls]
0x609A	-	返回加速	UDINT	[pls/s ²]

(2) 返回方式(0x6098)

值	描述
0	不返回
1, 2	<p>(1) 如果NOT switch为 Off (Off), 初始化移动方向变为正向 (CW)。如果 NOT switch 为 On (On), 改变方向。NOT switch 为 On 变为原点位置后, 在反向(CCW)运行中定位满足第一索引脉冲。</p> <p>(2) 如果 POT switch 为 Off (Off), 初始化移动方向变为反向(CCW)。如果POT switch 为On (On), 改变方向。POTswitch为On变为原点位置后, 在正向 (CW)运行中定位满足第一索引脉冲。</p>
7~10	<p>通过 (7) 到 (10) 方式, 原点位置由原点 (Home) 切换和 POT 切换决定。</p> <p>(7)靠上图片: 如果 POT 切换为 Off (Off), 在切换 (切换)搜索速度时动作, 初始移动方向变为反向 (CCW)。如果原点 (Home) 切换为 On (On), 方向改变。然后, 在正向(CW)运行变为原点位置中, 位置满足第一索引脉冲, 在 Zero 搜索速度时运行动作。</p> <p>(7)中间图片: 如果 POT 切换为 Off (Off), 原点 (Home) 切换为 On (On), 在切换 (切换)搜寻速率时, 运行动作, 初始化移动方向变为正向(CW)。如果原点 (Home) 切换为 Off (Off), 速度改变为Zero 搜寻速率。然后, 在正向(CW)运行变为原点位置中, 位置满足第一索引脉冲。</p> <p>(7)靠下图片: 如果 POT 切换为 Off (Off), 原点 (Home) 切换为 On (On), 在切换 (切换) 搜寻速度时运行动作, 初始化移动方向变为反向 (CCW)。如果 POT 切换为 On (On), 方向改变。当原点(Home) 切换从On(On) 变为 Off (Off)时, 在Zero搜寻速率时动作运行, 在持续正向(CW)运行变为原点 (Home) 位置中, 位置满足第一索引脉冲。</p> <p>(8) 到(10) 中原点返回和上面(7)除初始运行方向和监控原点(Home) 切换极性不同外, 其余位控方面方式相同。参考下图。</p>

值	描述
7~10	<p>Index pulse</p> <p>Home switch</p> <p>Positive limit switch (POT)</p>
11~14	<p>使用原点切换和NOT切换的11到14决定原点位置方式描述。</p> <p>(11) 上图: 如果NOT切换为Off, 然后驱动以切换查找速度和旋转CW运行. 如果原点切换在此时导通, 变更旋转方向, 当在0查找速度的驱动CCW时, 第一个索引脉冲遇到的位置变为原点位置.</p> <p>(11) 中图: 如果NOT切换为Off, 而原点切换为On, 然后驱动以切换查找速度和旋转CCW运行. 如果此时原点切换关断, 转换到0查找速度. 在 原点切换关断后, 当驱动以CCW 方向运行时, 第一个索引脉冲遇到的位置变为原点位置.</p> <p>(11) 下图: 如果NOT切换为Off, 然后驱动以切换查找速度和旋转CW运行. 如果此时NOT切换导通, 变更方向, 并以切换查找速度持续驱动CCW. 如果原点切换从On到Off, 然后转换到0查找速度, 第一个索引脉冲遇到的位置变为原点位置.</p> <p>12到14的方式和11的如何决定原点位置相同. 唯一不同的是初始驱动方向和原点切换极性. 参考下图.</p> <p>Index pulse</p> <p>Home switch</p> <p>Negative limit switch (NOT)</p>

值	描述
24	<p>(8) 以(8)方式决定原点位置, 但是不使用索引脉冲. 此外, 原点切换为On/Off 的点变为原点.</p> <p>Home switch</p> <p>Positive limit switch (POT)</p>

第8章 运动控制功能

值	描述
28	<p>(12) 以(12)方式决定原点位置, 但是不使用索引脉冲. 此外, 原点切换为On/Off 的点变为原点.</p> <p>Home switch</p> <p>Negative limit switch (NOT)</p>
33, 34	<p>位置满足索引脉冲反向 (CCW)/正向(CW) 移动变为原点 (Home) 位置中.</p> <p>Index pulse</p>
35	<p>原点运行启动点作为原点位置.</p>

备注) ←: 搜索 switch 中速度(0x6099:01), →: 搜索 0 中速度 (0x6099:02)

8.2 控制运行类型

运动控制模块通过设置运动控制程序执行控制。运动控制运行的种类包括速度位置控制，速度速度控制，速度转矩控制，插补控制，位置/速度转换控制，位置/转矩转换控制，速率/转矩转换控制。

8.2.1 单轴位置控制

从开始位置(当前停止位置)到目标位置 (移动目标点)执行运动功能块 (「相对位置运行 (MC_MoveRelative)」和「绝对位置运行 (MC_MoveAbsolute)」) 后，执行轴的指定位置控制。

1. 绝对坐标方式控制 (「绝对位置运行 (MC_MoveAbsolute)」)

(1)从开始位置到目标位置执行位置控制 (在绝对位置运行命中 'Position' 指定位置)。

(2) 位置控制基于指定位置(原点) 在 原点返回中执行。

(3) 在方向 (direction)输入中，指定方向运行。仅运行参数「无限重复运行」设置为'1: Enable' 时有效。

- 设置值: 0-未指定，1-正向，2-最短距离方向，3-反向，4-当前方向
- 当指定为最短距离方向，运行方向根据轴的形式自动选择最短距离方向。
- 按照如下方向 (direction) 输入运动。

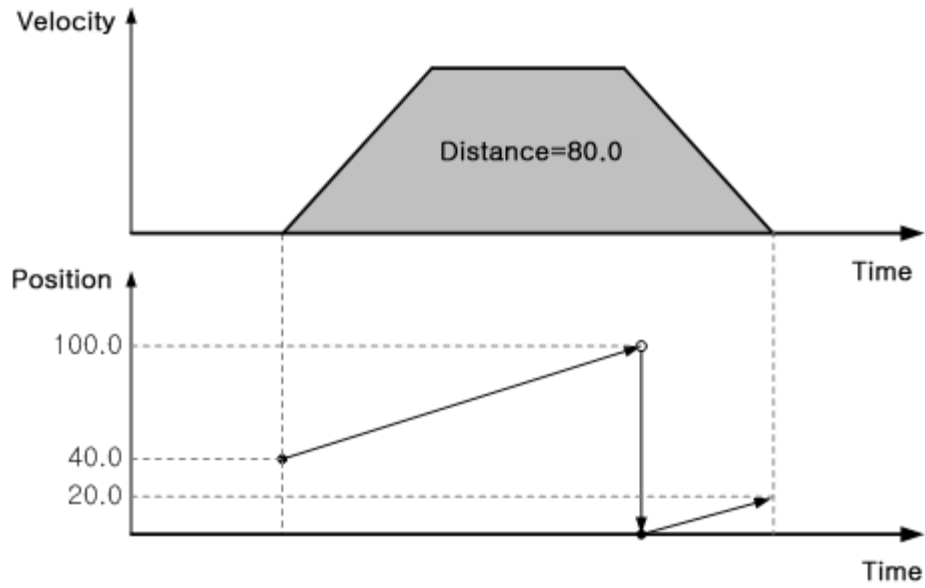
(a) 0- 未指定

超出重复长度重复位置的位置值可以指定。如果设置的位置值超出无限运行重复位置，从目标位置到当前位置的差别变为位控距离。绝对位置运行后命令位置通过以下等式计算。

命令位置 = 目标位置 - (无限运行重复位置 x n)
(n: 无限运行重复位置整数 x n 不超过目标位置)

[例] 按照以下设置执行绝对位置运行。

- 无限运行重复位置: 100.0
- 当前位置: 40.0
- 目标位置: 120.0
- 绝对位置运行命令位置= 120.0 - (100.0 x 1) = 20.0

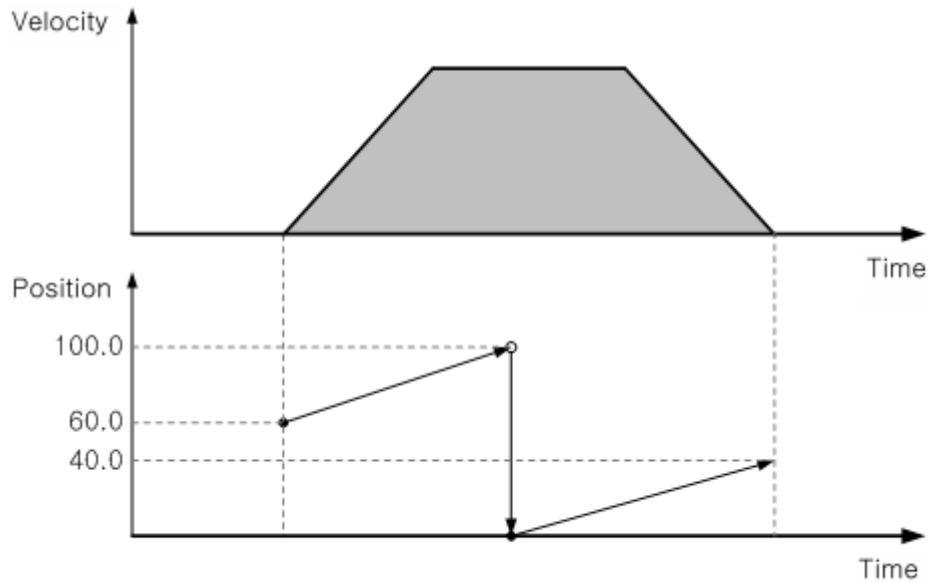


(b) 1-正向

位置控制向绝对位置的正向执行。如果目标位置设置超出无限运行重复位置范围，发生错误（错误代码：0x1081）

[例] 绝对位置运行在以下设置中执行。

- 无限运行重复位置: 100.0
- 当前位置: 60.0
- 目标位置: 40.0



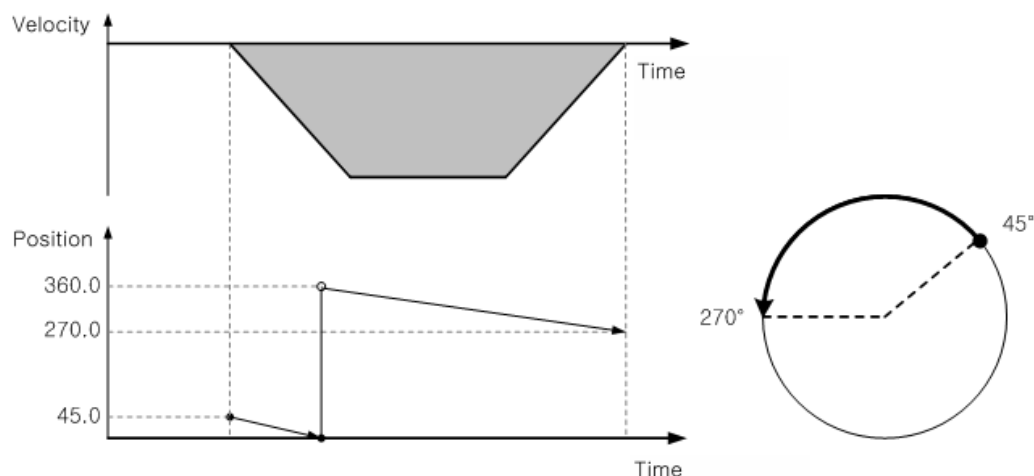
(c) 2-最短距离方向

位置控制的方向由当前位置到目标位置最短距离的旋转方向自动决定。也就是说，位置控制基于当前位置朝着最接近目标位置的方向执行。

如果目标位置设置超出无限运行重复位置范围，发生错误（错误代码：0x1081）

[例] 在以下设置中执行绝对位置运行。

- 无限运行重复位置: 360.0
- 当前位置: 45.0
- 目标位置: 270.0
- 直到移动距离为 225.0°，如果正向运行，如果 135.0°反向运行，运行在反向动作，最短距离方向。

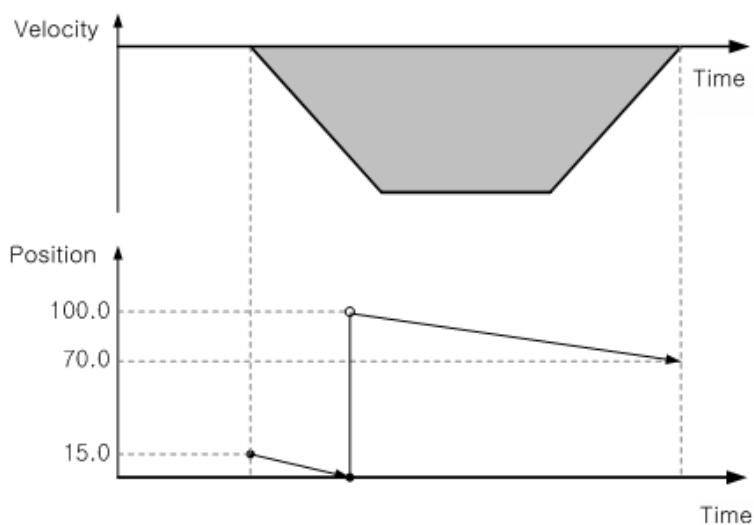


(d) 3-反向

位控执行朝着绝对位置的反向方向。如果目标位置设置超出无限运行重复位置范围，发生错误（错误代码：0x1081）

[例] 在以下设置中执行绝对位置运行。

- 无限运行重复位置: 100.0
- 当前位置: 15.0
- 目标位置: 70.0



第8章 运动控制功能

(e) 4- 当前方向

执行位控取决于当前运行方向。

如果当前运行方向为正向，以同样方式在方向='1-正向' 设置运行。

如果当前运行方向为反向，以同样方式在方向='3-反向方向' 设置。

(4) 如果运行参数「无限运行重复」设置为 '0: disable'，运行方向确定如下。

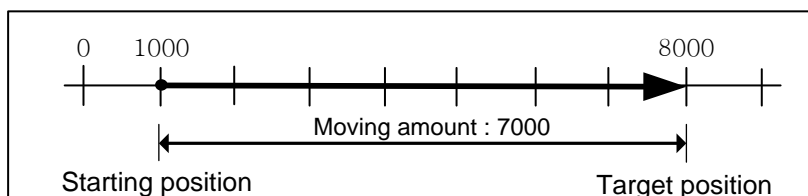
- 开始位置<目标位置: 正向位控运行
- 开始位置>目标位置: 反向位控运行

【例】 执行绝对坐标，在以下设置中，单轴位置控制

▷ 开始位置: 1000,

▷ 目标位置: 8000

正向移动量为 7000 (7000=8000-1000).

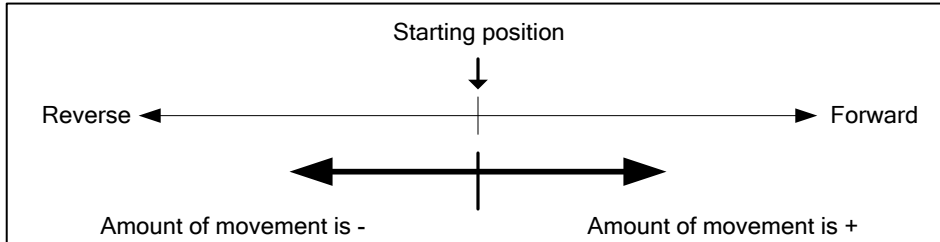


■ 相关运动功能块

名称	描述	运行条件
MC_MoveAbsolute	绝对位控运行	边沿
MC_MoveAbsolute		
BOOL	Execute	Done
UINT	Axis	Axis
BOOL	ContinuousUpdate	Busy
LREAL	Position	Active
LREAL	Velocity	CommandAborted
LREAL	Acceleration	Error
LREAL	Deceleration	ErrorID
LREAL	Jerk	
UINT	Direction	
UINT	BufferMode	

2.增量方式控制(「相对位控运行(MC_MoveRelative)」)

- (1) 对象从开始位置移动到目标目标的移动量。不同于绝对坐标的目标位置，目标位置的指定值不是位置值。而是从当前位置的移动量。
- (2) 转移方向由信号移动量决定
 - ▷ 转移方向(+): 正向位控运行(当前位置增加)
 - ▷ 转移方向(-): 反向位控运行(当前位置减少)

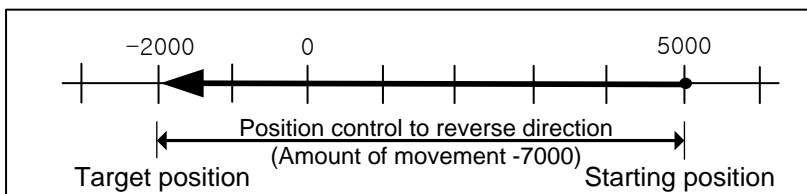


【例】执行绝对坐标，在以下设置中，单轴位置控制

▷ 开始位置: 5000

▷ 目标位置: -7000

反向运行停止在-2000处



■相关运动功能块

名称	描述	运行条件
MC_MoveRelative	相关位控运行	边沿
MC_MoveRelative		
BOOL	Execute	Done
UINT	Axis	Axis
BOOL	ContinuousUpdate	Busy
LREAL	Distance	Active
LREAL	Velocity	CommandAborted
LREAL	Acceleration	Error
LREAL	Deceleration	ErrorID
LREAL	Jerk	
UINT	BufferMode	

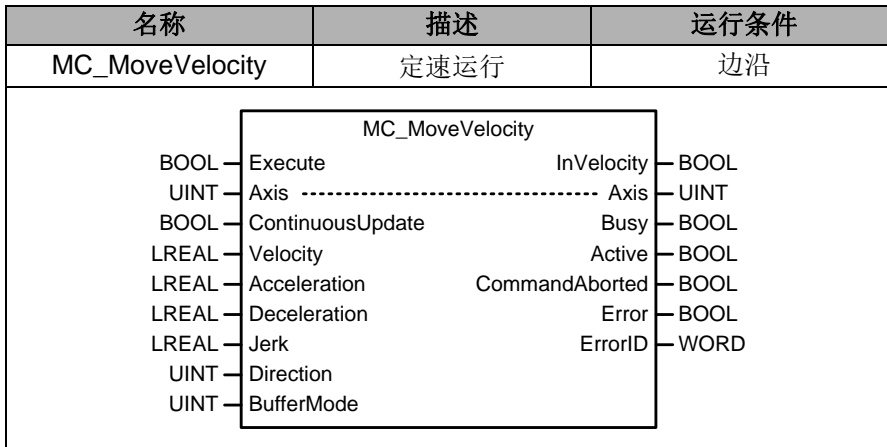
8.2.2 单轴速度控制

执行运动功能块(「定速运行 (MC_MoveVelocity)」), 直到停止条件输入前, 执行当前设置速度运行。

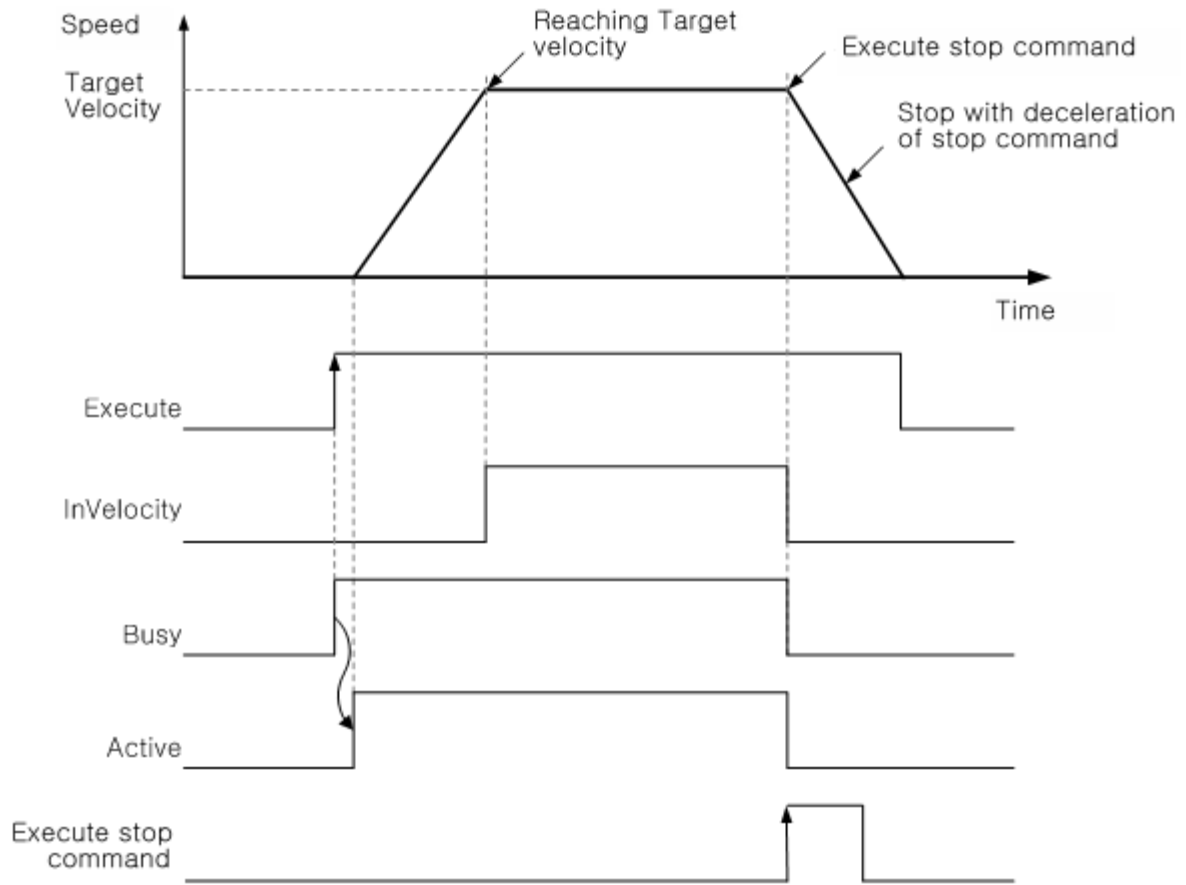
1. 控制特点

- (1) 使用指定速度和加减速执行指定轴的速度控制运行。执行速度控制通过发送目标位置值和位置控制的伺服驱动器目标速度相同。
- (2) 在方向输入中, 指定运行方向
(基于指定目标(Velocity)输入运行方向为正方向(Velocity)输入。例如, 如果指定目标速度(Velocity) 值为负数, 方向 (direction)输入为反方向, 轴最终运行为正方向。)
 - 设置值: 1-正向, 2-反向, 3-当前方向
- (3) 目标速度(Velocity) 输入值可设置为负数。如果目标速度设置为负数, 运行方向变的和之前指定方向相反。
 - 正向运行
 - 速度 > 0, 方向=1: 正向
 - 速度 < 0, 方向=2: 反向
 - 反向运行
- (4) 读取目标速度后, 功能块的速度输出为On (On)。如果有一个待执行指令, 待执行命令在 InSpeed 输出为 On 后执行。
- (5) 当前执行速度控制停止halt (MC_Halt) 或立即停止(MC_Stop) 运动功能块。

2. 相关功能块



3.运行时序图



8.2.3 单轴转矩控制

如果执行运动功能块(「转矩控制(MC_TorqueControl)」), 轴的转矩按照设置转矩值动作。

1. 控制特点

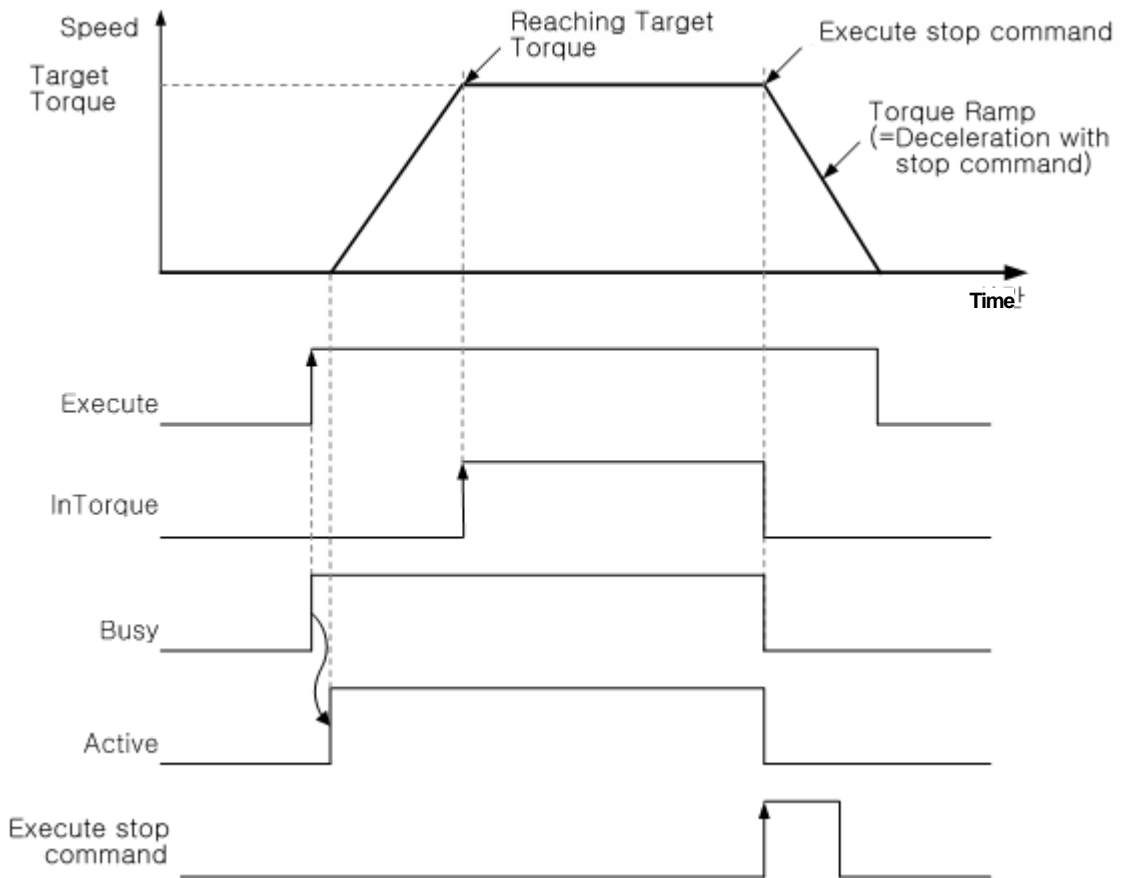
- (1) 指定轴的转矩控制利用目标转矩值和转矩上升斜率动作。
- (2) 转矩上升斜率(TorqueRamp)指转矩每秒到达目标转矩的变化率, 到达目标转矩的时间计算如下。
到达目标转矩的时间 (s) = 目标转矩(Torque) / 转矩上升斜率(TorqueRamp)
- (3) 使用伺服的转矩控制模式执行转矩控制。
- (4) 目标转矩值可达到两位小数, 反映在 [0.1%]单位。
- (5) 在方向输入中, 指定运行方向。
(然而, 正向基于指定转矩输入的运行方向。例如, 如果转矩值为负值, 在方向 (direction)输入中, 轴最终正向运行。)
 - 设置值: 1-正向, 2-反向, 3-当前方向
- (6) 转矩(目标转矩) 输入值设置为负数。如果目标转矩设置为负值, 运行方向变为之前指定方向的相反方向。
 - 正向运行
 - 转矩 > 0, 方向=1: 正向
 - 转矩 < 0, 方向=2: 反向
 - 反向运行
 - 转矩 > 0, 方向=2: 反向
 - 转矩 < 0, 方向=1: 正向
- (7) 转矩值设置范围如下。
-1000.0 % ~ 1000 %
- (8) 达到目标转矩后, 功能块的转矩输出为On。如果有一个待定命令, 转矩输出为 On后, 待定命令执行。
- (9) 执行停止halt(MC_Halt)或立即停止(MC_Stop)运动功能块, 当前转矩控制停止。

2. 相关运动功能块

名称	描述	运行条件
MC_TorqueControl	转矩控制	边沿

MC_TorqueControl			
BOOL	Execute	InTorque	BOOL
UINT	Axis	Axis	UINT
BOOL	ContinuousUpdate	Busy	BOOL
LREAL	Torque	Active	BOOL
LREAL	TorqueRamp	CommandAborted	BOOL
LREAL	Velocity	Error	BOOL
LREAL	Acceleration	ErrorID	WORD
LREAL	Deceleration		
LREAL	Jerk		
UINT	Direction		
UINT	BufferMode		

3. 运行时序图



8.2.4位置运行后定速运行

执行运动功能块（「相对位置运行后定速运行 (MC_MoveContinuousRelative)」后，执行指定轴的速度控制「绝对位控运行后定速运行(MC_MoveContinuousAbsolute)」），如果没有待执行命令，以指定结束率从开始位置（当前停止位置）到目标位置（移动位置点）定速结束速度(EndVelocity)位置控制。

1.控制特点

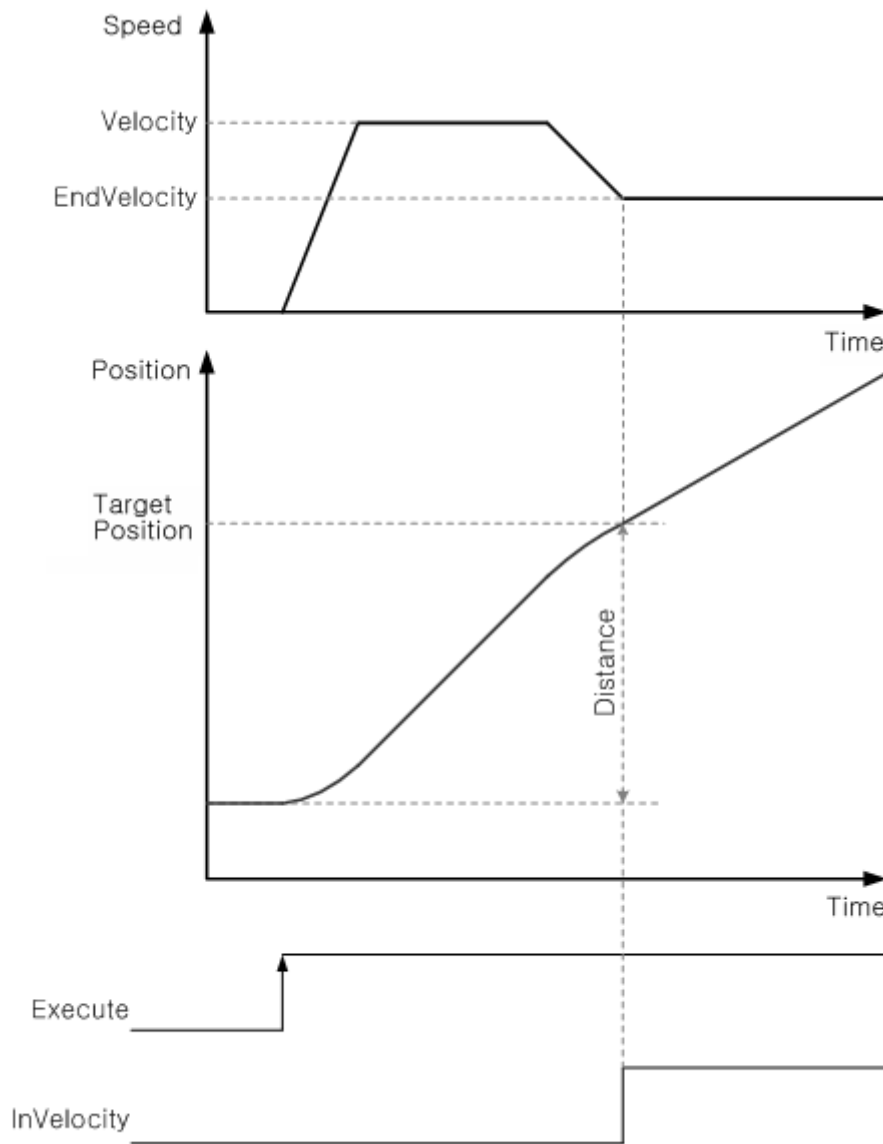
- (1) 位置控制从开始点到目标位置以指定的结束率执行结束。
- (2) 位置控制执行基于指定位置（原点位置）返回。
- (3) 绝对位置(MC_MoveContinuousAbsolute)运行后，定速运行。在方向输入中指定运行方向，仅当运行参数「无限运行重复」设置为 '1: Enable'时有效。
 - 设置值: 0-未指定， 1-正向， 2-最短距离方向， 3-反向， 4-当前方向
- (4) 位置控制运行到达目标位置完成后，到达结束率，功能块的InEnd速度输出为 On。如果有待执行命令，当 InEndspeed输出为 On 时执行。

2. 相关运动功能块

名称	描述	运行条件																																																
MC_MoveContinuousAbsolute	绝对位置运行后定速运行	边沿																																																
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="4">MC_MoveContinuousAbsolute</th> </tr> </thead> <tbody> <tr> <td>BOOL</td> <td>Execute</td> <td>InEndVelocity</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>Axis</td> <td>Axis</td> <td>UINT</td> </tr> <tr> <td>BOOL</td> <td>ContinuousUpdate</td> <td>Busy</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>Position</td> <td>Active</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>EndVelocity</td> <td>CommandAborted</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>Velocity</td> <td>Error</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>Acceleration</td> <td>ErrorID</td> <td>WORD</td> </tr> <tr> <td>LREAL</td> <td>Deceleration</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Jerk</td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>Direction</td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>BufferMode</td> <td></td> <td></td> </tr> </tbody> </table>			MC_MoveContinuousAbsolute				BOOL	Execute	InEndVelocity	BOOL	UINT	Axis	Axis	UINT	BOOL	ContinuousUpdate	Busy	BOOL	LREAL	Position	Active	BOOL	LREAL	EndVelocity	CommandAborted	BOOL	LREAL	Velocity	Error	BOOL	LREAL	Acceleration	ErrorID	WORD	LREAL	Deceleration			LREAL	Jerk			UINT	Direction			UINT	BufferMode		
MC_MoveContinuousAbsolute																																																		
BOOL	Execute	InEndVelocity	BOOL																																															
UINT	Axis	Axis	UINT																																															
BOOL	ContinuousUpdate	Busy	BOOL																																															
LREAL	Position	Active	BOOL																																															
LREAL	EndVelocity	CommandAborted	BOOL																																															
LREAL	Velocity	Error	BOOL																																															
LREAL	Acceleration	ErrorID	WORD																																															
LREAL	Deceleration																																																	
LREAL	Jerk																																																	
UINT	Direction																																																	
UINT	BufferMode																																																	

名称	描述	运行条件																																												
MC_MoveContinuousRelative	相对位置运行后定速运行	边沿																																												
<table border="1"> <thead> <tr> <th colspan="4">MC_MoveContinuousRelative</th> </tr> </thead> <tbody> <tr> <td>BOOL</td> <td>Execute</td> <td>InEndVelocity</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>Axis</td> <td>Axis</td> <td>UINT</td> </tr> <tr> <td>BOOL</td> <td>ContinuousUpdate</td> <td>Busy</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>Distance</td> <td>Active</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>EndVelocity</td> <td>CommandAborted</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>Velocity</td> <td>Error</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>Acceleration</td> <td>ErrorID</td> <td>WORD</td> </tr> <tr> <td>LREAL</td> <td>Deceleration</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Jerk</td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>BufferMode</td> <td></td> <td></td> </tr> </tbody> </table>			MC_MoveContinuousRelative				BOOL	Execute	InEndVelocity	BOOL	UINT	Axis	Axis	UINT	BOOL	ContinuousUpdate	Busy	BOOL	LREAL	Distance	Active	BOOL	LREAL	EndVelocity	CommandAborted	BOOL	LREAL	Velocity	Error	BOOL	LREAL	Acceleration	ErrorID	WORD	LREAL	Deceleration			LREAL	Jerk			UINT	BufferMode		
MC_MoveContinuousRelative																																														
BOOL	Execute	InEndVelocity	BOOL																																											
UINT	Axis	Axis	UINT																																											
BOOL	ContinuousUpdate	Busy	BOOL																																											
LREAL	Distance	Active	BOOL																																											
LREAL	EndVelocity	CommandAborted	BOOL																																											
LREAL	Velocity	Error	BOOL																																											
LREAL	Acceleration	ErrorID	WORD																																											
LREAL	Deceleration																																													
LREAL	Jerk																																													
UINT	BufferMode																																													

3. 运行时序图



8.2.5 开关控制

在动作控制模块中，开关控制表示位置控制/速度控制/转矩控制间的实时控制开关。

如果控制模式为当前执行（位置控制，速度控制，转矩控制）则意味着立即改变为不同的控制模式，指令缓冲模式设置为异常停止，并执行相关动作功能块。

1. 位置-速度开关控制

当在绝对/相对位置运行中的轴执行指定速度运行(MC_MoveVelocity)时，位置控制是切换到速度控制。在切换到速度控制的开始阶段速度持续运行，从前一个位置控制的速度运行到当前速度控制的目标速度。在速度控制运行期间下一个运行可通过传导停止(MC_Halt)持续运行，通过立即停止动作(MC_Stop)功能块执行运行停止或者执行其他动作功能块。

2. 速度-位置开关控制

当速度控制期间轴指定速度执行绝对/相对/增量位置控制(MC_MoveAbsolute, MC_MoveRelative, MC_MoveAdditive)动作功能块时，速度控制切换到位置控制。在切换到位置控制的开始阶段速度持续运行，从前一个位置控制的速度运行到当前速度控制的目标速度。在位置控制运行期间下一个运行可通过传导停止(MC_Halt)持续运行，通过立即停止动作(MC_Stop)功能块执行运行停止或者执行其他动作功能块。

3. 位置-转矩开关控制

当位置控制期间轴绝对/相对位置运行执行转矩控制(MC_TorqueControl)动作功能块时，位置控制切换到转矩控制。在切换到转矩控制的开始阶段转矩持续运行，从前一个位置控制的转矩运行到当前转矩控制的目标转矩。在转矩控制运行期间下一个运行可通过传导停止(MC_Halt)持续运行，通过立即停止动作(MC_Stop)功能块执行运行停止或者执行其他动作功能块。

4. 转矩-位置开关控制

当转矩控制期间轴转矩控制运行执行绝对/相对/增量位置控制转矩控制(MC_MoveAbsolute, MC_MoveRelative, MC_MoveAdditive)动作功能块时，转矩控制切换到位置控制，当转矩值减少到0时，一次停止后位置控制持续运行。在位置控制运行期间下一个运行可通过传导停止(MC_Halt)持续运行，通过立即停止动作(MC_Stop)功能块执行运行停止或者执行其他动作功能块。

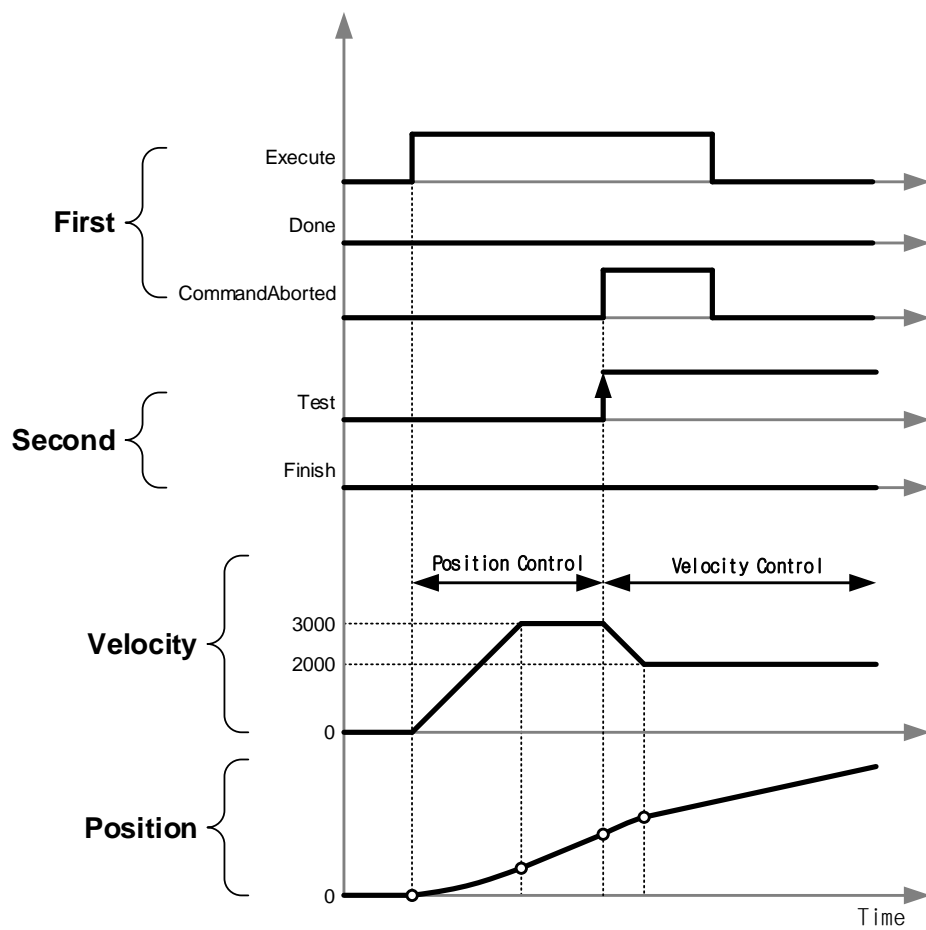
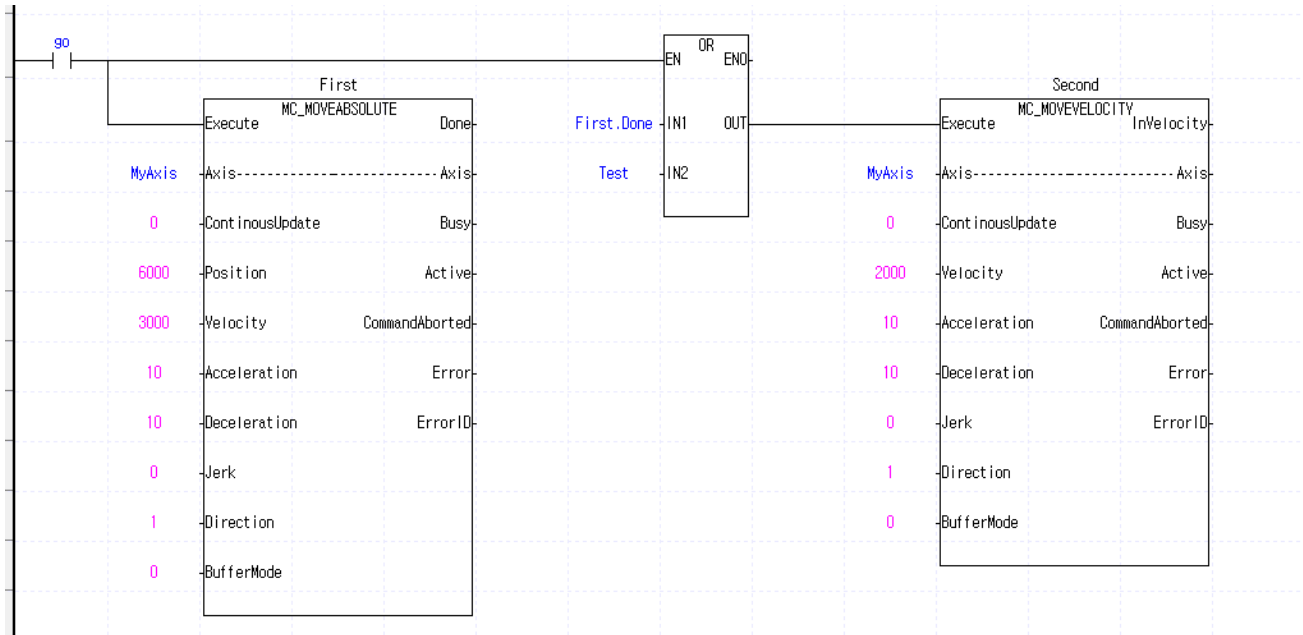
5. 速度-转矩开关控制

当速度控制期间轴以指定速度运行执行转矩控制(MC_TorqueControl)动作功能块，速度控制切换到转矩控制。在切换到转矩控制的开始阶段转矩持续运行，从前一个速度控制的转矩运行到当前转矩控制的目标转矩。在转矩控制运行期间下一个运行可通过传导停止(MC_Halt)持续运行，通过立即停止动作(MC_Stop)功能块执行运行停止或者执行其他动作功能块。

6. 转矩-速度开关控制

当速度控制期间轴以转矩控制运行执行指定速度运行(MC_MoveVelocity)动作功能块时，转矩控制切换到速度控制，当转矩值减少到0时，一次停止后速度控制持续运行。在速度控制运行期间下一个运行可通过传导停止(MC_Halt)持续运行，通过立即停止动作(MC_Stop)功能块执行运行停止或者执行其他动作功能块。

7. 使用开关控制实例



8.2.6轴组控制

轴组控制是一种通过设置相关多个轴到一个轴组中控制移动对象轨迹的功能。对于轴组控制，设置为轴组。轴组运行包括线性插值，圆弧插值和螺旋插值。

对于运行轴组控制的坐标系统，仅支持直角坐标系统。

1. 添加轴到组中

对于轴组控制，轴组应先设置并激活后执行运行。

配置轴可以指定，使用MP-PM设置轴组。另外，使用动作功能块添加轴到轴组中或者从轴组中移除。当轴组配置完成时，在轴组使能后可执行轴组运行。

(1) 添加轴到组中

表示添加一个轴到轴组中。指定配置轴到IdentInGroup是添加到AxesGroup输入中的指定轴组。仅在轴组处于组禁用(GroupDisabled)和组准备(GroupStandBy)状态下执行。

名称	描述	运行条件																					
MC_AddAxisToGroup	添加轴到组中	边沿																					
<table border="1" style="margin: auto;"> <thead> <tr> <th colspan="3">MC_AddAxisToGroup</th> </tr> </thead> <tbody> <tr> <td>BOOL</td> <td>Execute</td> <td>Done</td> </tr> <tr> <td>UINT</td> <td>AxesGroup</td> <td>AxesGroup</td> </tr> <tr> <td>UINT</td> <td>Axis</td> <td>Axis</td> </tr> <tr> <td>UINT</td> <td>IdentInGroup</td> <td>Busy</td> </tr> <tr> <td></td> <td></td> <td>Error</td> </tr> <tr> <td></td> <td></td> <td>ErrorID</td> </tr> </tbody> </table>			MC_AddAxisToGroup			BOOL	Execute	Done	UINT	AxesGroup	AxesGroup	UINT	Axis	Axis	UINT	IdentInGroup	Busy			Error			ErrorID
MC_AddAxisToGroup																							
BOOL	Execute	Done																					
UINT	AxesGroup	AxesGroup																					
UINT	Axis	Axis																					
UINT	IdentInGroup	Busy																					
		Error																					
		ErrorID																					

(2) 从组中移除

表示从轴组中移除一个轴。配置轴指定到IdentInGroup是从AxesGroup输入中的指定轴组移除。仅在轴组处于组禁用(GroupDisabled)和组准备(GroupStandBy)状态下执行。

如果在轴组中没有剩余的轴，轴组变为禁止状态。

名称	描述	运行条件																		
MC_RemoveAxisToGroup	从组中移除轴	边沿																		
<table border="1" style="margin: auto;"> <thead> <tr> <th colspan="3">MC_RemoveAxisFromGroup</th> </tr> </thead> <tbody> <tr> <td>BOOL</td> <td>Execute</td> <td>Done</td> </tr> <tr> <td>UINT</td> <td>AxesGroup</td> <td>AxesGroup</td> </tr> <tr> <td>UINT</td> <td>IdentInGroup</td> <td>Busy</td> </tr> <tr> <td></td> <td></td> <td>Error</td> </tr> <tr> <td></td> <td></td> <td>ErrorID</td> </tr> </tbody> </table>			MC_RemoveAxisFromGroup			BOOL	Execute	Done	UINT	AxesGroup	AxesGroup	UINT	IdentInGroup	Busy			Error			ErrorID
MC_RemoveAxisFromGroup																				
BOOL	Execute	Done																		
UINT	AxesGroup	AxesGroup																		
UINT	IdentInGroup	Busy																		
		Error																		
		ErrorID																		

(3) 从组中移除全部轴

表示从轴组中移除所有的轴。

名称	描述	运行条件																	
MC_UngroupAllAxes	从组中移除全部轴	边沿																	
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center;">MC_UngroupAllAxes</th> </tr> </thead> <tbody> <tr> <td style="width: 30%;">BOOL — Execute</td> <td style="width: 40%; text-align: center;">Done</td> <td style="width: 30%;">— BOOL</td> </tr> <tr> <td>UINT — AxesGroup</td> <td style="text-align: center;">----- AxesGroup</td> <td>— UINT</td> </tr> <tr> <td></td> <td style="text-align: center;">Busy</td> <td>— BOOL</td> </tr> <tr> <td></td> <td style="text-align: center;">Error</td> <td>— BOOL</td> </tr> <tr> <td></td> <td style="text-align: center;">ErrorID</td> <td>— WORD</td> </tr> </tbody> </table>			MC_UngroupAllAxes		BOOL — Execute	Done	— BOOL	UINT — AxesGroup	----- AxesGroup	— UINT		Busy	— BOOL		Error	— BOOL		ErrorID	— WORD
MC_UngroupAllAxes																			
BOOL — Execute	Done	— BOOL																	
UINT — AxesGroup	----- AxesGroup	— UINT																	
	Busy	— BOOL																	
	Error	— BOOL																	
	ErrorID	— WORD																	

(4) 组使能

执行轴组指令改变状态到使能允许。

以下情况下轴组无法使能。

- 如果没有轴组配置轴，或者轴组中的轴没有连接到网络。
- 如果使能的轴组配置轴属于其他使能轴组。
- 如果轴组配置轴中有一个轴为运行中。
- 如果轴组中配置轴‘单位’不相同。

名称	描述	运行条件																	
MC_GroupEnable	组使能	边沿																	
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center;">MC_GroupEnable</th> </tr> </thead> <tbody> <tr> <td style="width: 30%;">BOOL — Execute</td> <td style="width: 40%; text-align: center;">Done</td> <td style="width: 30%;">— BOOL</td> </tr> <tr> <td>UINT — AxesGroup</td> <td style="text-align: center;">----- AxesGroup</td> <td>— UINT</td> </tr> <tr> <td></td> <td style="text-align: center;">Busy</td> <td>— BOOL</td> </tr> <tr> <td></td> <td style="text-align: center;">Error</td> <td>— BOOL</td> </tr> <tr> <td></td> <td style="text-align: center;">ErrorID</td> <td>— WORD</td> </tr> </tbody> </table>			MC_GroupEnable		BOOL — Execute	Done	— BOOL	UINT — AxesGroup	----- AxesGroup	— UINT		Busy	— BOOL		Error	— BOOL		ErrorID	— WORD
MC_GroupEnable																			
BOOL — Execute	Done	— BOOL																	
UINT — AxesGroup	----- AxesGroup	— UINT																	
	Busy	— BOOL																	
	Error	— BOOL																	
	ErrorID	— WORD																	

(5) 组禁止

改变轴组到组禁止状态。

如果轴组运行中，在立即停止后轴组改变为禁止状态。

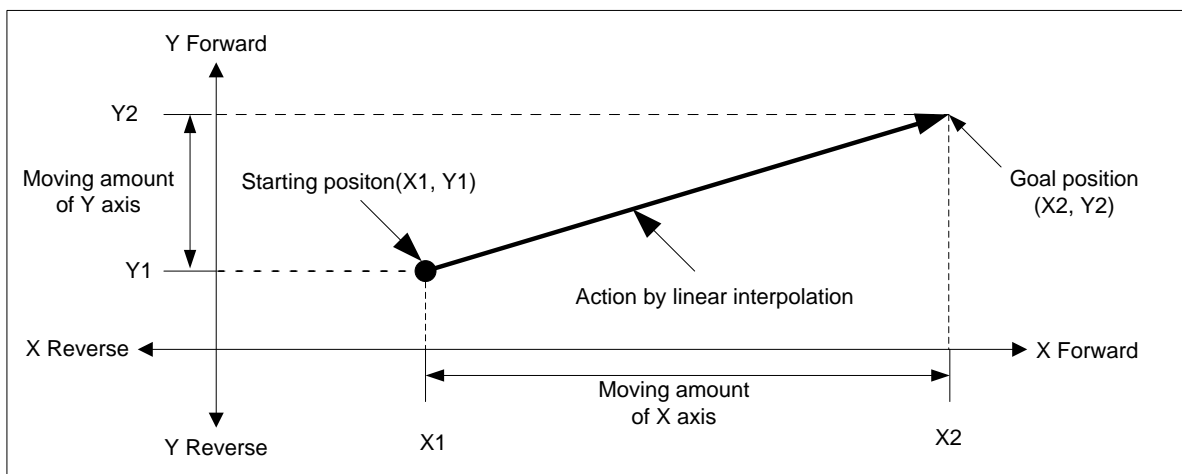
名称	描述	运行条件																	
MC_GroupDisable	组禁止	边沿																	
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center;">MC_GroupDisable</th> </tr> </thead> <tbody> <tr> <td style="width: 30%;">BOOL — Execute</td> <td style="width: 40%; text-align: center;">Done</td> <td style="width: 30%;">— BOOL</td> </tr> <tr> <td>UINT — AxesGroup</td> <td style="text-align: center;">----- AxesGroup</td> <td>— UINT</td> </tr> <tr> <td></td> <td style="text-align: center;">Busy</td> <td>— BOOL</td> </tr> <tr> <td></td> <td style="text-align: center;">Error</td> <td>— BOOL</td> </tr> <tr> <td></td> <td style="text-align: center;">ErrorID</td> <td>— WORD</td> </tr> </tbody> </table>			MC_GroupDisable		BOOL — Execute	Done	— BOOL	UINT — AxesGroup	----- AxesGroup	— UINT		Busy	— BOOL		Error	— BOOL		ErrorID	— WORD
MC_GroupDisable																			
BOOL — Execute	Done	— BOOL																	
UINT — AxesGroup	----- AxesGroup	— UINT																	
	Busy	— BOOL																	
	Error	— BOOL																	
	ErrorID	— WORD																	

8.2.7 线性插值控制

通过在轴组中相关轴设置，以线性轨迹执行从起始位置（当前停止位置）到目标位置的多轴插值。
线性插值最多执行10轴

1. 绝对坐标的线性插值控制(「绝对，线性插值运行(MC_MoveLinearAbsolute)」)

- (1) 指定位置数据执行从起始位置到目标位置的线性插值。位置控制基于从原点的指定位置来执行。
- (2) 基于每轴的起始位置和目標位置的移动方向。
 - 起始位置<目标位置:正方向运行
 - 起始位置>目标位置:负方向运行



- (3) 当前执行的插值通过组停止(MC_GroupHalt)或者组立即停止(MC_GroupStop)动作功能块而停止。
- (4) 设置在绝对位置线性插值运行(MC_MoveLinearAbsolute)动作功能块中的速度值表示组成轴组的轴的同步率。

每轴的速度和运行速度如下。

$$Interpolating\ speed (F) = Operations\ speed\ set\ in\ position\ data$$

$$Interpolating\ moving\ amount (S) = \sqrt{S_1^2 + S_2^2 + \dots + S_{10}^2}$$

$$Axis\ speed (V_1) = Interpolating\ speed (F) \times \frac{Main\ axis\ moving\ amount (S_1)}{Interpolating\ moving\ amount (S)}$$

$$Axis\ 2\ speed (V_2) = Interpolating\ speed (F) \times \frac{Sub - axis\ moving\ amount (S_2)}{Interpolating\ moving\ amount (S)}$$

$$Axis\ 10\ speed (V_{10}) = Interpolating\ speed (F) \times \frac{Sub - axis\ moving\ amount (S_{10})}{Interpolating\ moving\ amount (S)}$$

(5) 相关动作功能块

名称	描述	运行条件
MC_MoveLinearAbsolute	绝对位置线性插值运行	边沿
MC_MoveLinearAbsolute		
BOOL	Execute	Done
UINT	AxesGroup	AxesGroup
LREAL[]	Position	Busy
LREAL	Velocity	Active
LREAL	Acceleration	CommandAborted
LREAL	Deceleration	Error
LREAL	Jerk	ErrorID
UINT	BufferMode	
UINT	TransitionMode	
LREAL	TransitionParameter	

(6) 限制条件

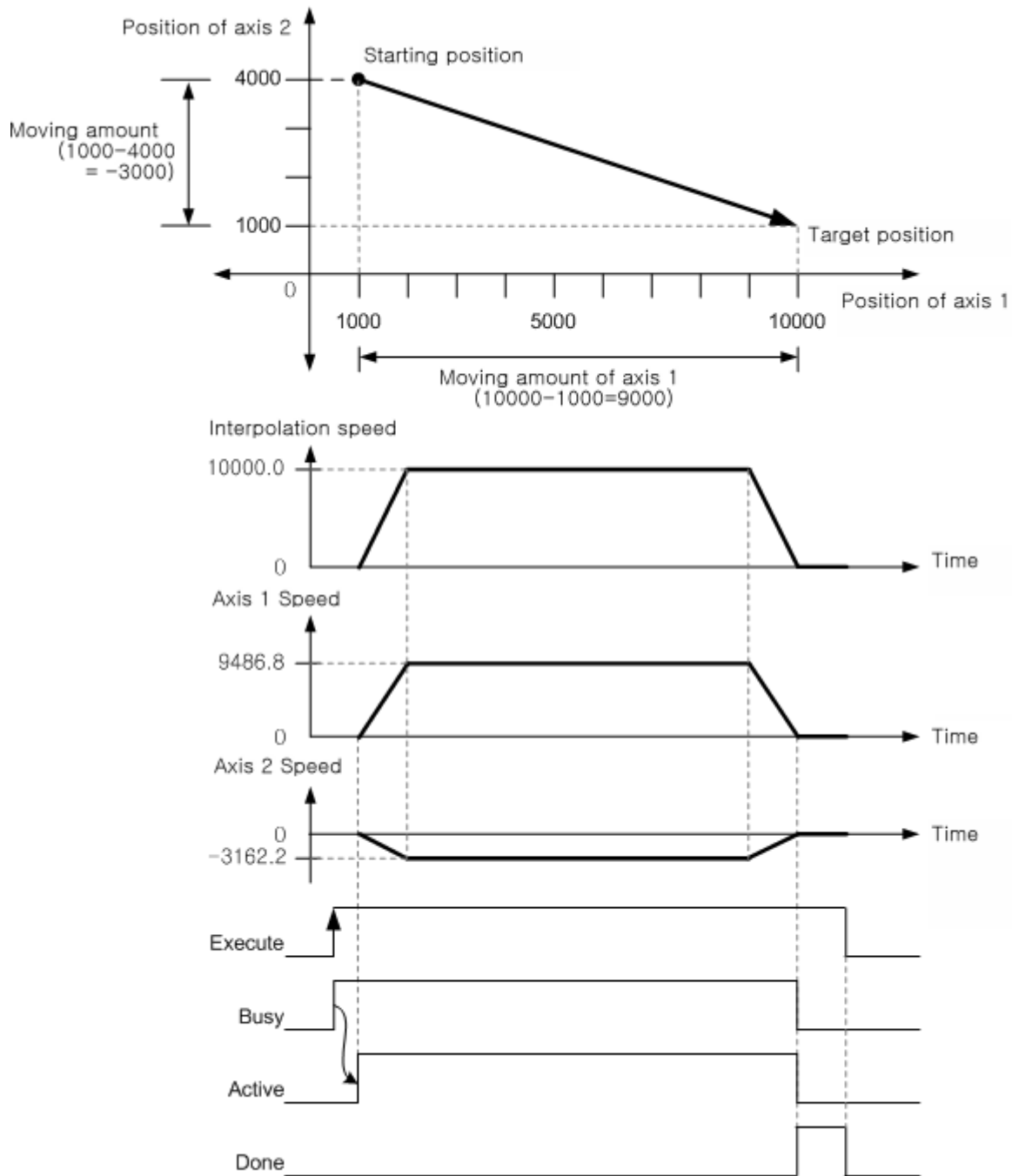
通过绝对坐标系统的线性插值不可以在以下条件下执行。

- 如果配置轴中一个轴为起始不确定状态(错误代码: 0x2090)
- 如果配置轴的运行速度超出每轴的速度限制(错误代码: 0x2091)
- 如果配置轴中的一个轴为无限重复运行(错误代码: 0x2094)

第8章 运动控制功能

(7) 运行时序

- 当前位置: (1000.0, 4000.0)
- 目标位置: (10000.0, 1000.0)
- 目标速率: 10000.0

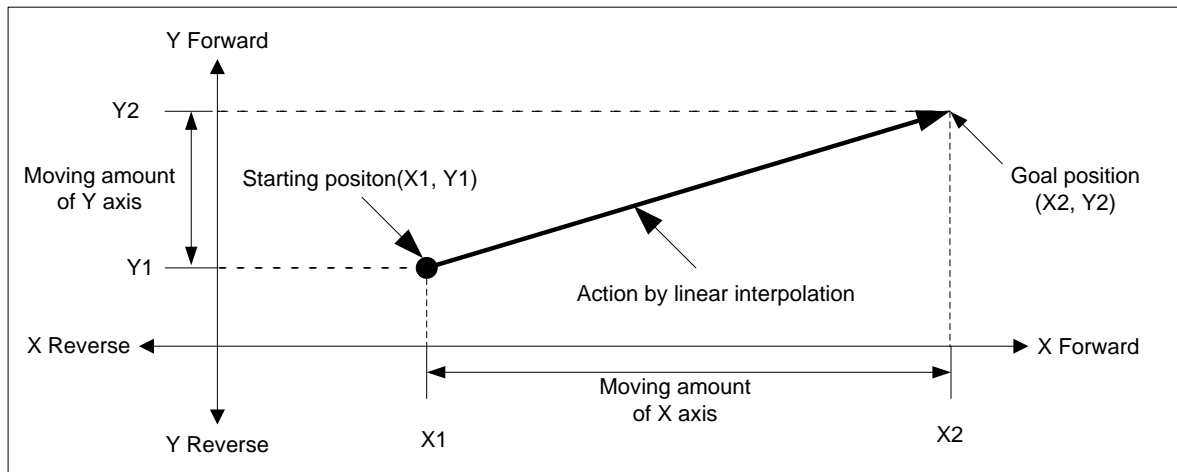


※ 每个配置轴速率为估算。

2. 相对坐标的线性插值控制

(「相对, 线性插值控制(MC_MoveLinearAbsolute)」)

- (1) 通过每轴和包括运行方向的位置执行从起始位置到目标运行方向的线性插值。基于当前停止位置的位置控制。
- (2) 运行方向由设置在每轴目标位置（运行距离）的符号决定。
 - 当移动距离符号为正数(+或者无符号): 正方向位置运行(当前位置增加方向)
 - 当移动距离符号为负数(-): 负方向位置运行 (当前位置减少方向)



- (3) 当前执行的插值通过组停止(MC_GroupHalt)或者组立即停止(MC_GroupStop)动作功能块而停止。
- (4) 设置在相对位置线性插值运行(MC_MoveLinearRelative)动作功能块中的速度值表示组成轴组的轴的同步率。
每个配置轴运行速度计算如下。

$$Interpolation\ speed\ (F) = Operations\ speed\ set\ in\ position\ data$$

$$Interpolation\ moving\ amount\ (S) = \sqrt{S_1^2 + S_2^2 + \dots + S_{10}^2}$$

$$Axis\ speed\ (V_1) = Interpolation\ speed\ (F) \times \frac{Main\ axis\ moving\ amount\ (S_1)}{Interpolation\ moving\ amount\ (S)}$$

$$Axis\ 2\ speed\ (V_2) = Interpolation\ speed\ (F) \times \frac{Sub - axis\ moving\ amount\ (S_2)}{Interpolation\ moving\ amount\ (S)}$$

$$Axis\ 10\ speed\ (V_{10}) = Interpolation\ speed\ (F) \times \frac{Sub - axis\ moving\ amount\ (S_{10})}{Interpolation\ moving\ amount\ (S)}$$

(5) 相关动作功能块

名称	描述	运行条件
MC_MoveLinearRelative	相对位置线性插值运行	边沿
MC_MoveLinearRelative		
BOOL	Execute	Done
UINT	AxesGroup	----- AxesGroup
LREAL[]	Distance	Busy
LREAL	Velocity	Active
LREAL	Acceleration	CommandAborted
LREAL	Deceleration	Error
LREAL	Jerk	ErrorID
UINT	BufferMode	
UINT	TransitionMode	
LREAL	TransitionParameter	

(6) 限制条件

通过相对坐标系统的线性插值不可以在以下条件下执行。

- 如果配置轴中的一个轴为无限重复运行(错误代码: 0x2094)
- 如果配置轴的运行速度超出每轴的速度限制(错误代码: 0x2091)

8.2.8 圆弧插值控制

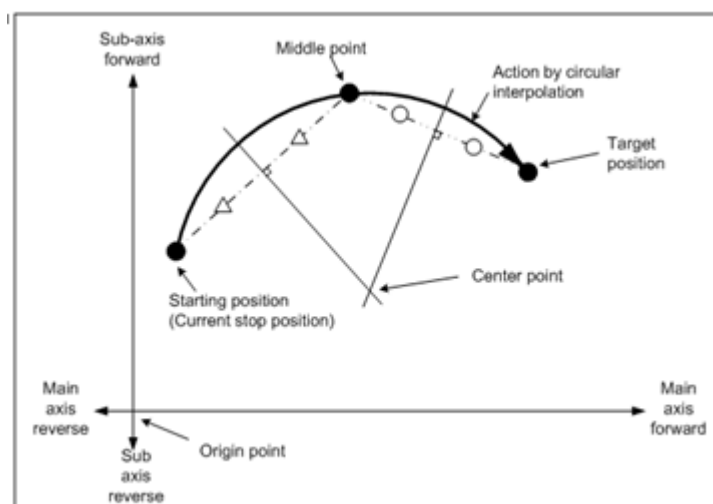
插值运行按照轴组中两轴设置的轴进程的圆弧轨道执行。

有三种方式的圆弧插值，例如穿过辅助点的指定位置的中点方式，认定辅助点中指定位置作为中心点的中心点方式，以及根据‘环模式’设置和辅助点确定以辅助点中的指定值作为圆弧的半径的半径方式。

当前执行的插值根据运动功能块中的组停止(MC_GroupHalt)或者组急停(MC_GroupStop)而停止。

1. 使用中点方式的圆弧插值

- (1) 通过设置在辅助点中的中点位置，圆弧插值执行从开始位置到目标位置。
- (2) 圆弧轨迹产生一个交点，由起始位置和中点位置，中点位置和目标位置的垂直平分线产生。
- (3) 移动方向由设置的目标位置和圆弧插值辅助点自动决定。



(4) 限制条件

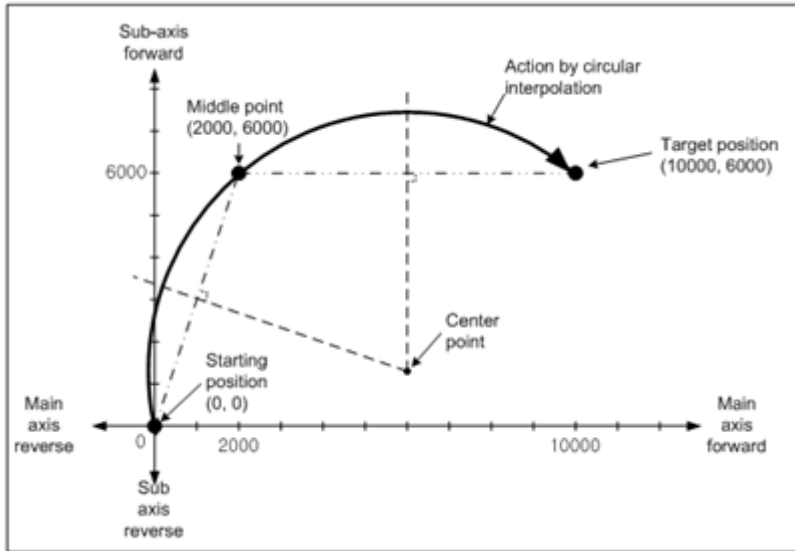
中点方式的圆弧插值在以下条件下无法执行。

- 如果在绝对坐标圆弧插值运行中，配置轴中有一个轴为起始未确定状态(错误代码: 0x20A0)
- 如果作为辅助点的指定中点与起始位置或者目标位置相同(错误代码: 0x20A4)
- 如果起始位置和目标位置相同(错误代码: 0x20A5)
- 如果圆弧计算半径超出2147483647pls (错误代码: 0x20A6)
- 如果起始位置，辅助点位置和目标位置在一条直线(错误代码: 0x20A7)
- 如果配置轴中有一个轴在无限重复运行(错误代码: 0x20A8)
- 如果轴组中配置轴数量为4(错误代码: 0x20A9)
- 如果轴组配置设置不是按照顺序进行设置(错误代码: 0x20AA)

第8章 运动控制功能

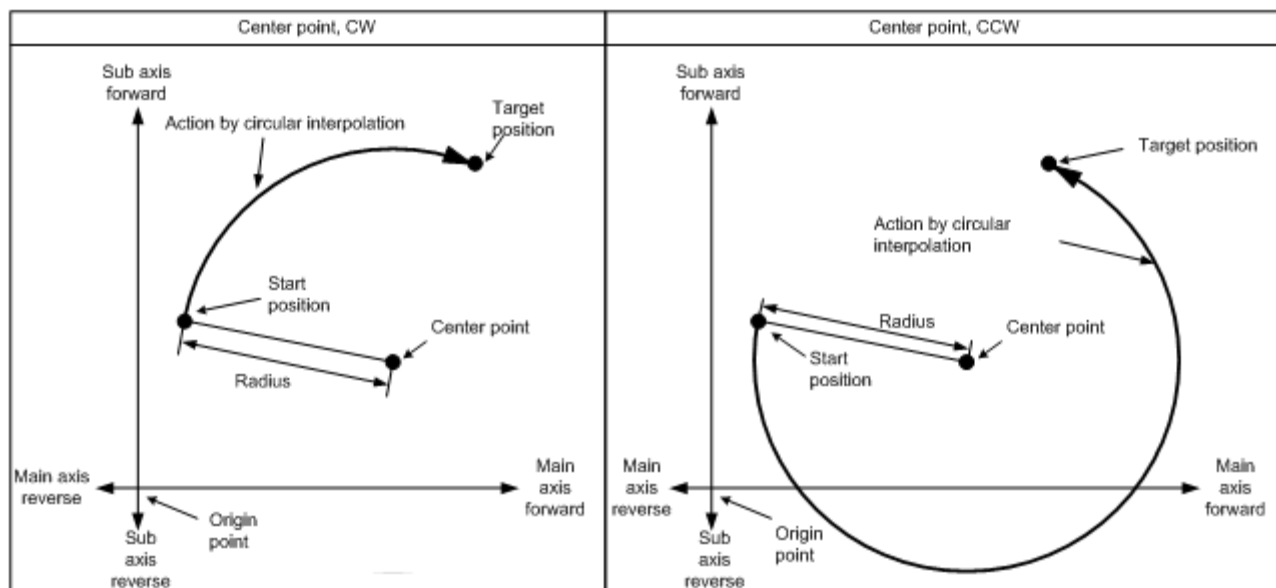
(5) 运行模式

- 当前位置: (0.0, 0.0)
- 目标位置: (10000.0, 6000.0)
- 中点: (2000.0, 6000.0)
- 方式(环模式):中点(0)
- 方向(路径选择): - (中点方式中忽略)

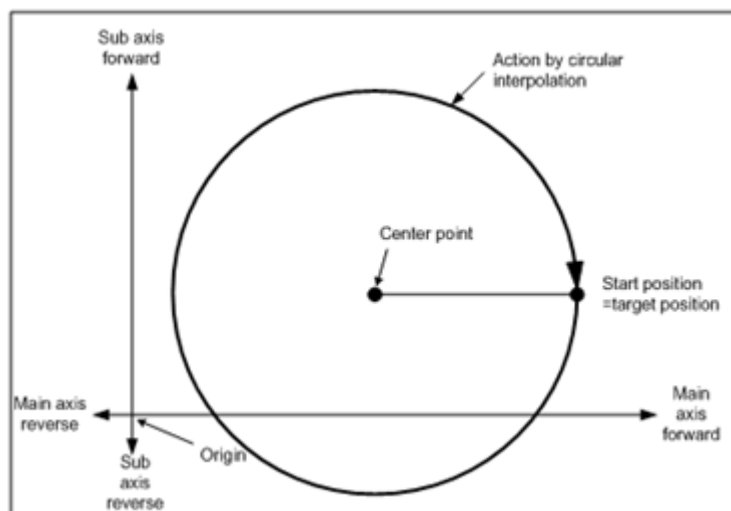


2. 使用中心点方式的圆弧插值

- (1) 圆弧插值执行从开始位置到目标位置，沿着起始位置到指定中心点位置的距离作为半径的圆弧轨迹运行。
- (2) 移动方向根据绝对位置圆弧插值运行(MC_绝对圆弧移动)中“路径选择”设置或者运动功能块相关位置圆弧插值运行(MC_相对圆弧移动)决定。
 - 0: 「CW」 - 圆弧插值从当前位置开始以顺时针方向执行。
 - 1: 「CCW」 - 圆弧插值从当前位置开始以逆时针方向执行。



- (3) 如果目标位置和开始位置相同，可以执行以中心点到开始点距离为圆弧半径的圆弧插值。



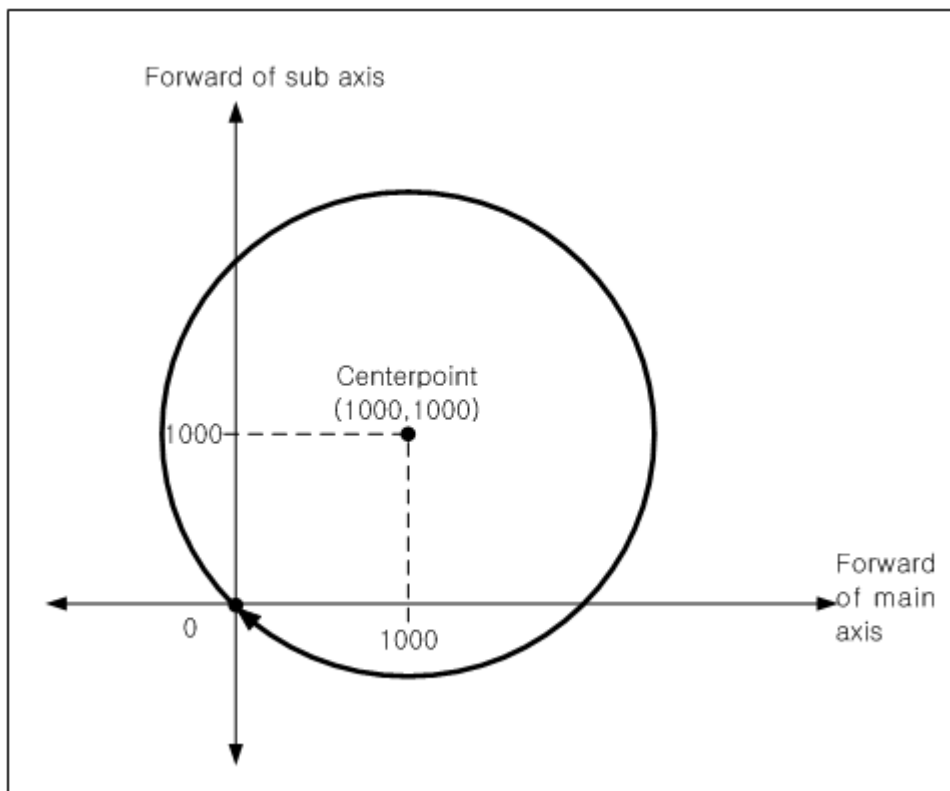
(4) 限制条件

中心点方式的圆弧插值在以下条件下无法执行。

- 如果在绝对坐标圆弧插值运行中，配置轴中有一个轴为起始未确定状态(错误代码: 0x20A0)
- 如果作为辅助点的指定中点与起始位置或者目标位置相同(错误代码: 0x20A4)
- 如果起始位置和目标位置相同(错误代码: 0x20A5)
- 如果圆弧计算半径超出2147483647pls (错误代码: 0x20A6)
- 如果起始位置，辅助点位置和目标位置在一条直线(错误代码: 0x20A7)
- 如果配置轴中有一个轴在无限重复运行(错误代码: 0x20A8)
- 如果轴组中配置轴数量为4(错误代码: 0x20A9)
- 如果轴组配置设置不是按照顺序进行设置(错误代码: 0x20AA)

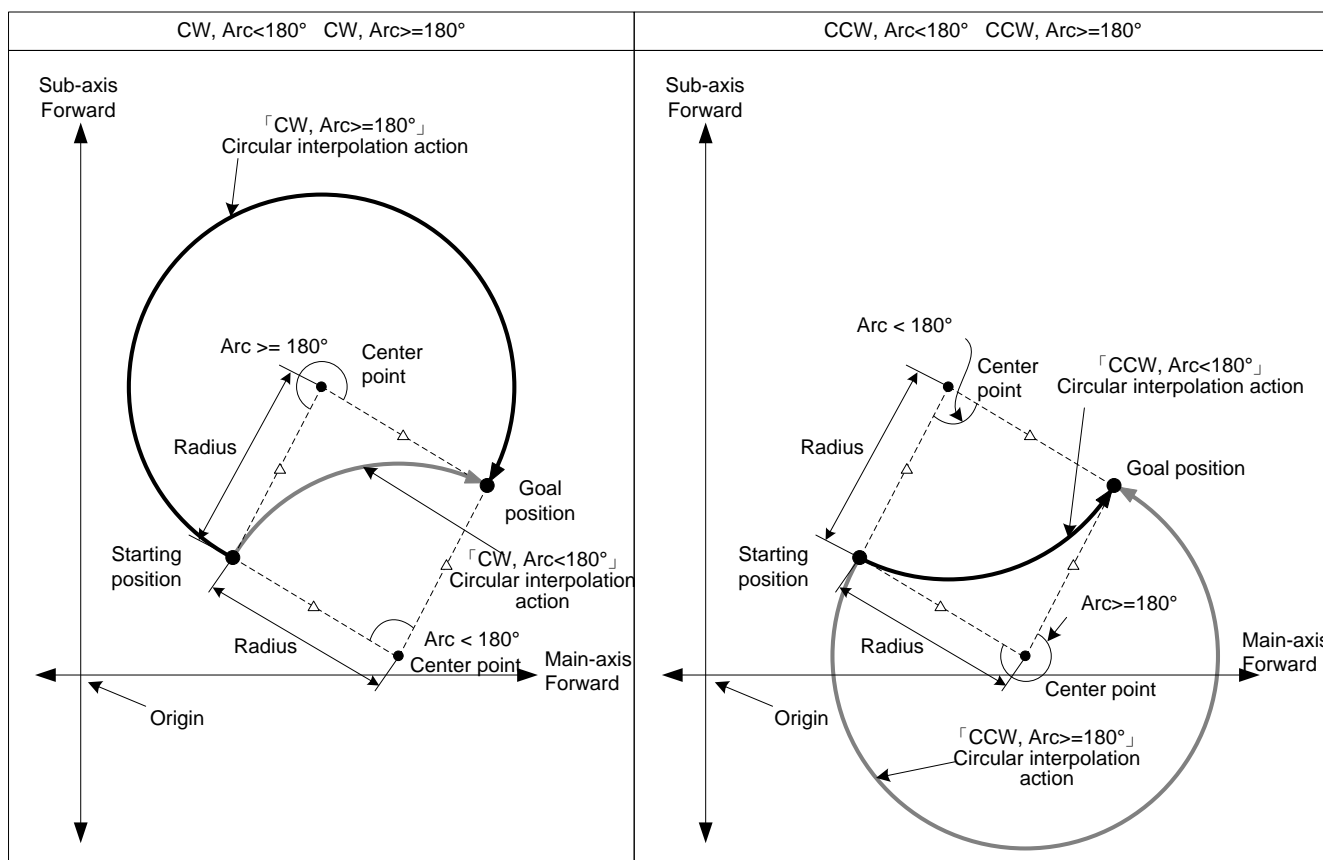
(5) 运行模式

- 当前位置: (0.0, 0.0)
- 目标位置: (0.0, 0.0)
- 中心点位置: (1000.0, 1000.0)
- 方式(环模式): 中心点(1)
- 方向(路径选择): CW(0)



3. 使用半径方式的圆弧插值

(1) 圆弧插值按照在圆弧插值辅助点中设置值的圆弧轨迹执行从起始位置到目标位置。根据绘出半径的符号圆弧具有的中心点((+): 圆弧角度 $<180^\circ$, (-):圆弧角度 $\geq 180^\circ$)。



(2) 在半径方式的圆弧插值中，目标位置不可以设置为与起始位置相同。

(3) 移动方向和圆弧的尺寸由辅助点的符号和绝对位置圆弧插值运行(MC_绝对圆弧移动)或者相对位置插值(MC_相对圆弧移动)运动功能块的“路径选择”(CW, CCW)方向设置决定。

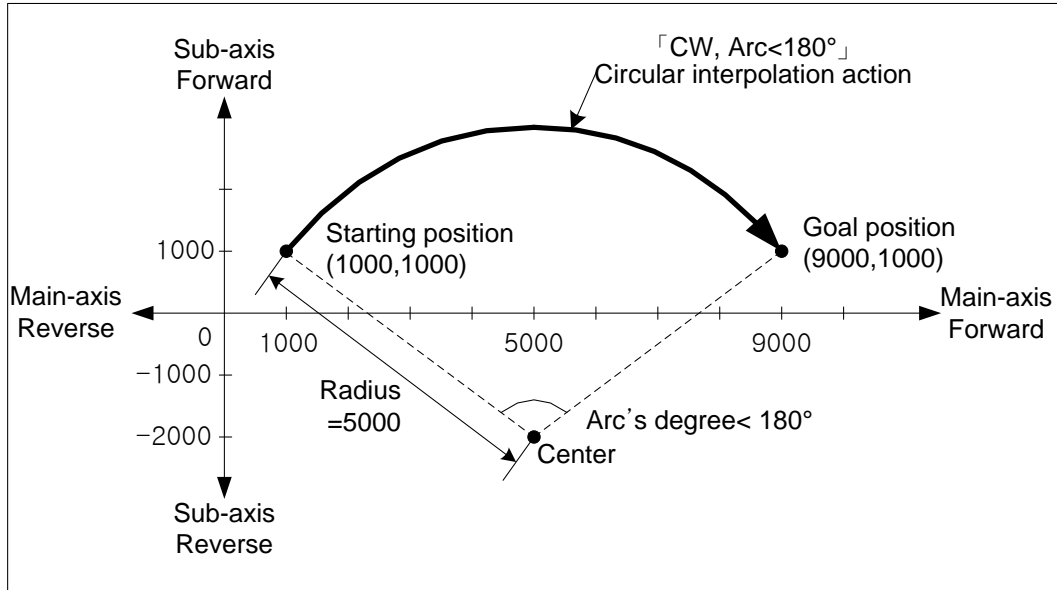
(4) 限制条件

半径方式的圆弧插值在以下条件下无法执行。

- 如果在绝对坐标圆弧插值运行中，配置轴中有一个轴为起始未确定状态(错误代码: 0x20A0)
- 如果起始位置和目标位置相同(错误代码: 0x20A5)
- 如果圆弧计算半径超出2147483647pls (错误代码: 0x20A6)
- 如果起始位置，辅助点位置和目标位置在一条直线(错误代码: 0x20A7)
- 如果配置轴中有一个轴在无限重复运行(错误代码: 0x20A8)
- 如果轴组中配置轴数量为4(错误代码: 0x20A9)
- 如果轴组配置设置不是按照顺序进行设置(错误代码: 0x20AA)

(5) 运行模式

- 当前位置: (1000.0, 1000.0)
- 目标位置: (9000.0, 1000.0)
- 中点: (5000.0, 0.0)
- 方式(环模式): 半径(2)
- 方向(路径选择): CW(0)



4. 相关动作功能块

(1) 绝对定位圆弧插值运行

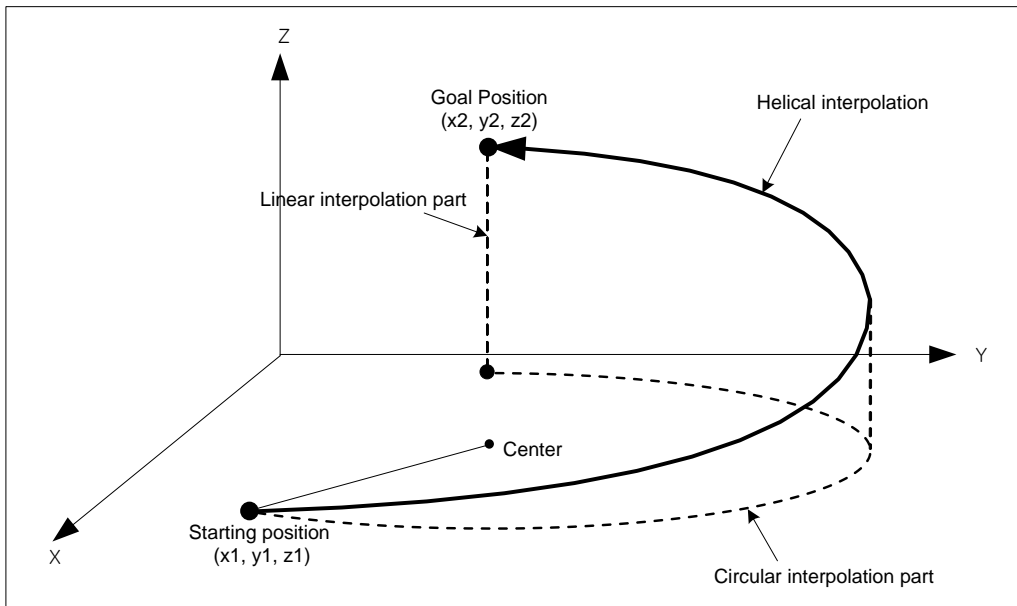
名称	描述	运行条件
MC_绝对圆弧移动	绝对定位圆弧插值运行	边沿
MC_MoveCircularAbsolute		
BOOL — Execute		Done — BOOL
UINT — AxesGroup	----- AxesGroup	UINT
UINT — CircMode		Busy — BOOL
LREAL[] — AuxPoint		Active — BOOL
LREAL[] — EndPoint	CommandAborted	— BOOL
UINT — PathChoice	Error	— BOOL
LREAL — Velocity	ErrorID	— WORD
LREAL — Acceleration		
LREAL — Deceleration		
LREAL — Jerk		
UINT — BufferMode		
UINT — TransitionMode		
LREAL — TransitionParameter		

(2) 相对定位圆弧插值运行

名称	描述	运行条件																																																																														
MC_相对圆弧移动	相对定位圆弧插值运行	边沿																																																																														
<div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <p style="text-align: center; margin: 0;">MC_MoveCircularRelative</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">BOOL</td> <td style="width: 40%;">Execute</td> <td style="width: 30%;"></td> <td style="width: 30%;"></td> <td style="width: 30%;">Done</td> <td style="width: 30%;">BOOL</td> </tr> <tr> <td>UINT</td> <td>AxesGroup</td> <td>-----</td> <td>AxesGroup</td> <td></td> <td>UINT</td> </tr> <tr> <td>UINT</td> <td>CircMode</td> <td></td> <td></td> <td>Busy</td> <td>BOOL</td> </tr> <tr> <td>LREAL[]</td> <td>AuxPoint</td> <td></td> <td></td> <td>Active</td> <td>BOOL</td> </tr> <tr> <td>LREAL[]</td> <td>EndPoint</td> <td></td> <td>CommandAborted</td> <td></td> <td>BOOL</td> </tr> <tr> <td>USINT</td> <td>PathChoice</td> <td></td> <td>Error</td> <td></td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>Velocity</td> <td></td> <td>ErrorID</td> <td></td> <td>WORD</td> </tr> <tr> <td>LREAL</td> <td>Acceleration</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Deceleration</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Jerk</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>BufferMode</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>TransitionMode</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>TransitionParameter</td> <td></td> <td></td> <td></td> <td></td> </tr> </table> </div>			BOOL	Execute			Done	BOOL	UINT	AxesGroup	-----	AxesGroup		UINT	UINT	CircMode			Busy	BOOL	LREAL[]	AuxPoint			Active	BOOL	LREAL[]	EndPoint		CommandAborted		BOOL	USINT	PathChoice		Error		BOOL	LREAL	Velocity		ErrorID		WORD	LREAL	Acceleration					LREAL	Deceleration					LREAL	Jerk					UINT	BufferMode					UINT	TransitionMode					LREAL	TransitionParameter				
BOOL	Execute			Done	BOOL																																																																											
UINT	AxesGroup	-----	AxesGroup		UINT																																																																											
UINT	CircMode			Busy	BOOL																																																																											
LREAL[]	AuxPoint			Active	BOOL																																																																											
LREAL[]	EndPoint		CommandAborted		BOOL																																																																											
USINT	PathChoice		Error		BOOL																																																																											
LREAL	Velocity		ErrorID		WORD																																																																											
LREAL	Acceleration																																																																															
LREAL	Deceleration																																																																															
LREAL	Jerk																																																																															
UINT	BufferMode																																																																															
UINT	TransitionMode																																																																															
LREAL	TransitionParameter																																																																															

5. 螺旋插值

- (1) 三轴用于执行圆弧插值指令(「绝对定位圆弧插值运行(MC_绝对圆弧移动)」,「相对定位圆弧插值运行(MC_相对圆弧移动)」)。即,两轴根据圆弧插值设置沿圆弧轨迹进行移动,一轴以圆弧插值动作同步执行线性插值。
- (2) 线性轴是圆弧插值轴组中的第三轴。
- (3) 关于执行螺旋插值,圆弧插值指令轴组需要设置3轴,线性插值目标位置设置在‘结束点’的第三轴。



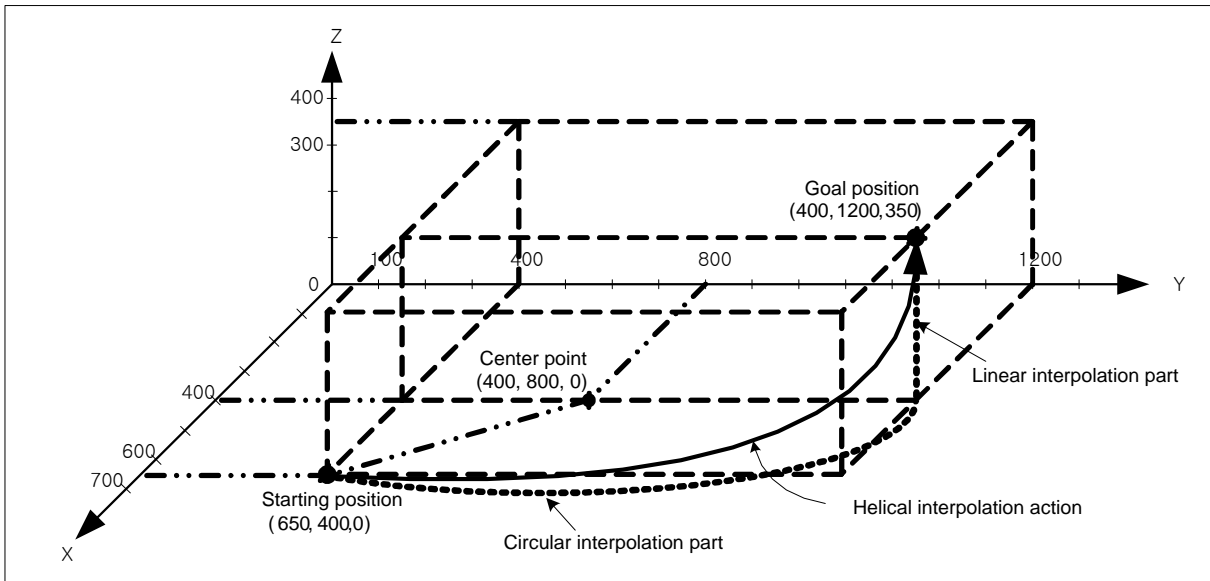
第8章 运动控制功能

(4) 限制条件

螺旋插值的限制条件与设置在圆弧插值模式中的相同。

(5) 运行模式

- 当前位置: (650.0, 400.0, 0)
- 目标位置: (400.0, 1200.350)
- 中心点: (800.0, 400.0)
- 方式(环模式): 中心点(1)
- 方向(路径选择): CCW(1)



8.2.9轴控制缓冲模式

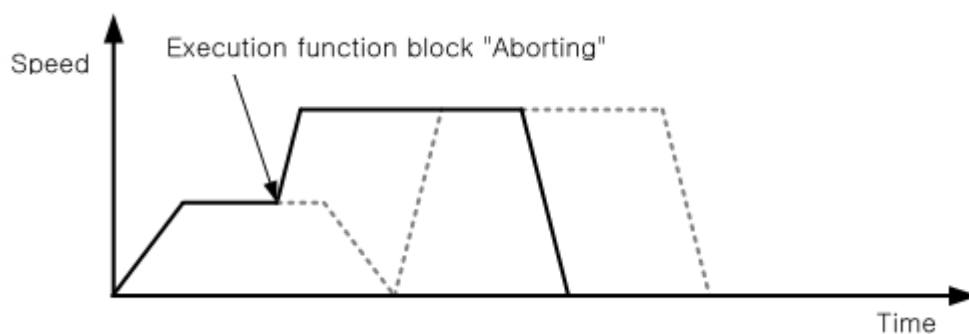
当轴运行状态下，取消当前轴运行并持续或者连续运行其他运动功能块来实现。指定动作是进入运动功能块的缓冲模式(BufferMode)。轴控制最大运行数量在缓冲中排列的数量为10。如果缓冲模式执行的指令大于这个数量，错误发生(错误代码: 0x1022)。

缓冲区域可设置值如下。

缓冲模式	描述
Aborting	立即执行指令，运行中的当前指令异常停止。
Buffered	运行中完成当前指令后执行该指令。
BlendingLow	通过比较当前指令和所需指令的速度，进行组合操作有助于融入低速度。
BlendingPrevious	执行组合运行来组合当前指令的速度。
BlendingNext	执行组合运行来组合下一个指令的速度。
BlendingHigh	通过比较当前指令和所需指令的速度，进行组合操作有助于融入高速度。

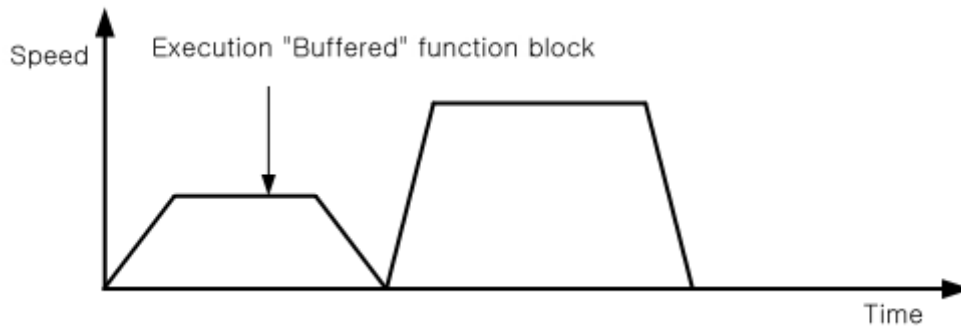
1. 缓冲模式“Aborting”

立即执行异常停止当前指令并执行下一个指令。当前运动功能块的CommandAborted指令输出为On。



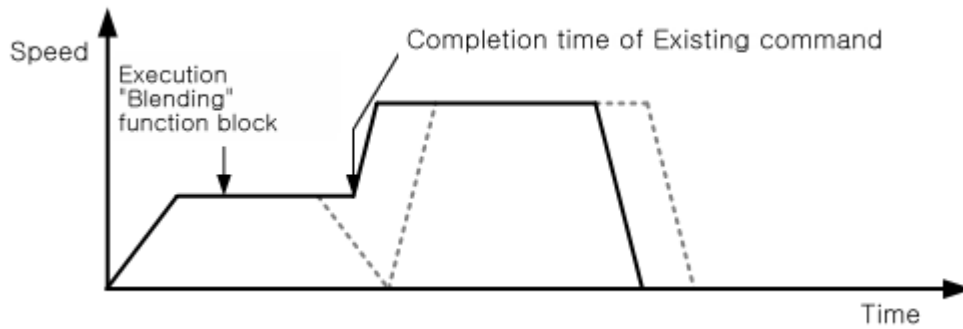
2. 缓冲模式 “Buffered”

在完成执行当前指令后执行下一个指令(Done输出为On)。



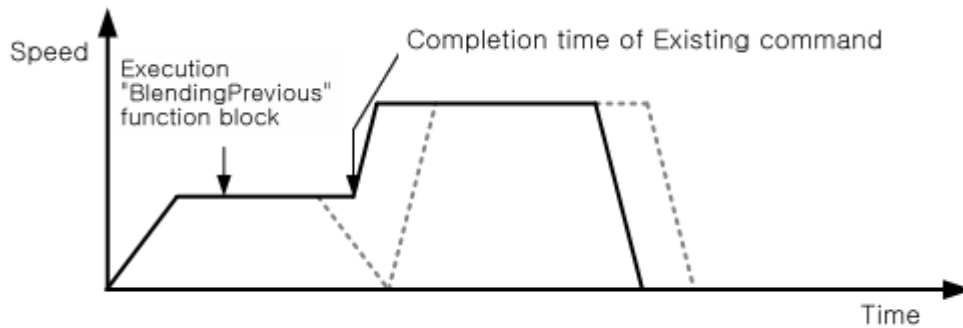
3. 缓冲模式 “BlendingLow”

组合运行以便在比较当前指令的目标速度，指令完成后并执行缓冲指令后以较低速运行。



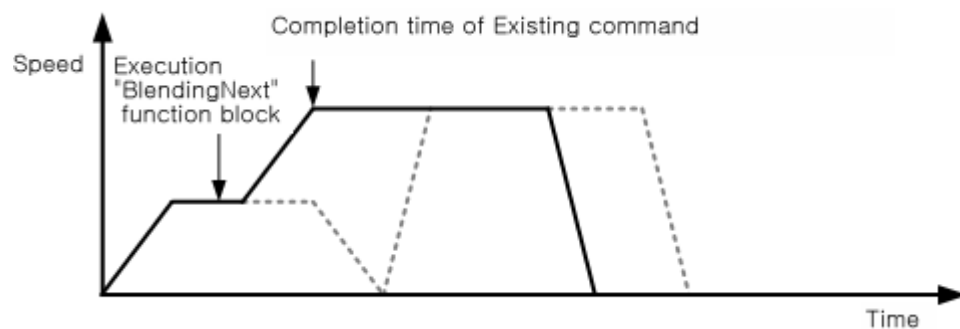
4. 缓冲模式 “BlendingPrevious”

当已有命令完成时，下一个命令缓冲目标速度的加减速执行，保持已执行命令速度后运行。



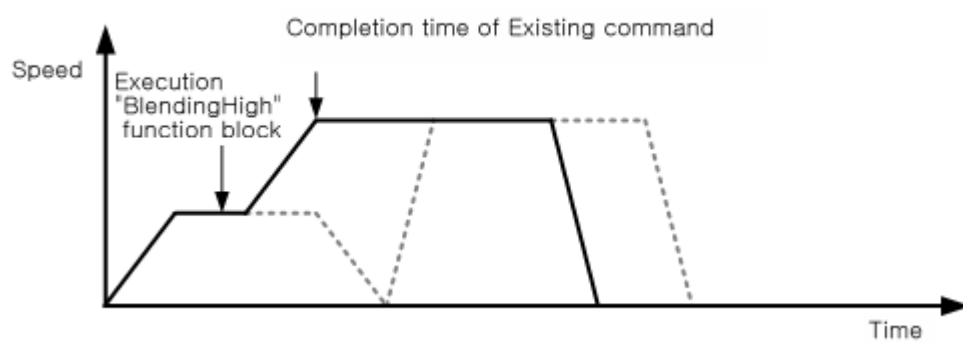
5. 缓冲模式 “BlendingNext”

当已有命令完成时，加减速后执行下一个命令，运行按照下一个命令的目标速度执行。



6. 缓冲模式 “BlendingHigh”

在命令完成时的已有命令的目标速度和缓冲命令之间比较，以一个较高速度执行联合运行



8.2.10 轴组控制缓冲模式和过渡模式

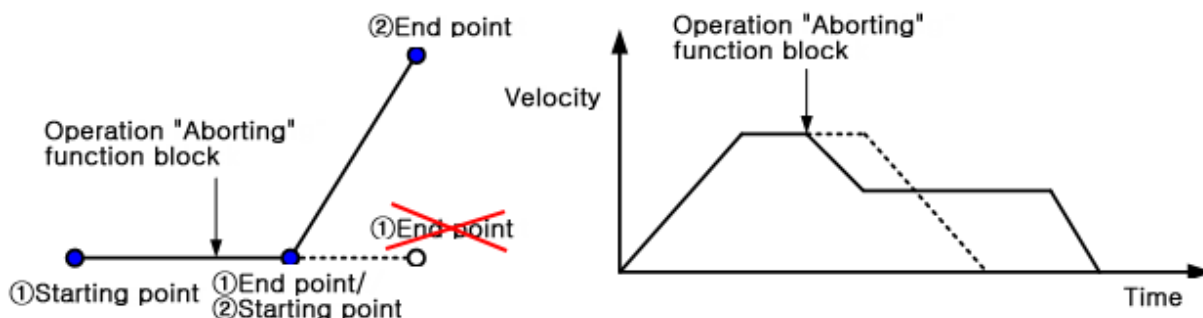
在轴组控制作为速度控制，动作指令可以通过使用缓冲模式持续执行，最大运行数量在缓冲区中可排列10个。如果缓冲模式的执行指令大于这个数量，错误发生(错误代码: 0x2022)。

另外，可通过使用过渡模式在两条线性轨迹中插入曲线来运行。

1. ‘缓冲模式’

(1) 异常停止

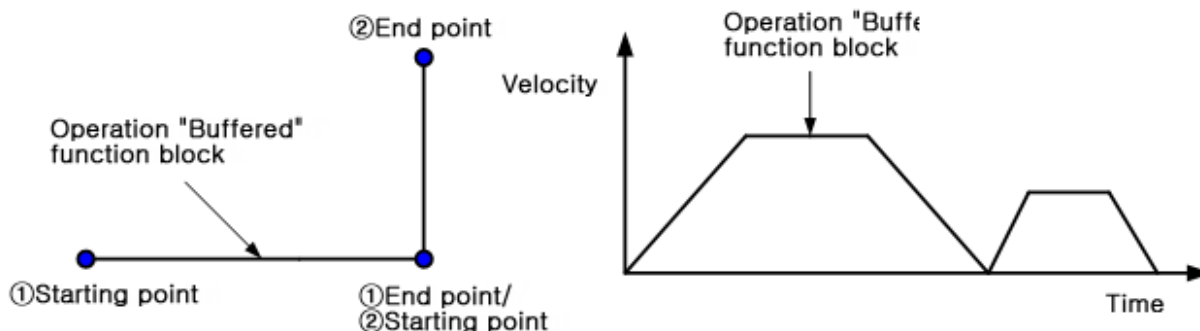
停止当前运行的动作，并立即执行新动作。



(2) 缓冲

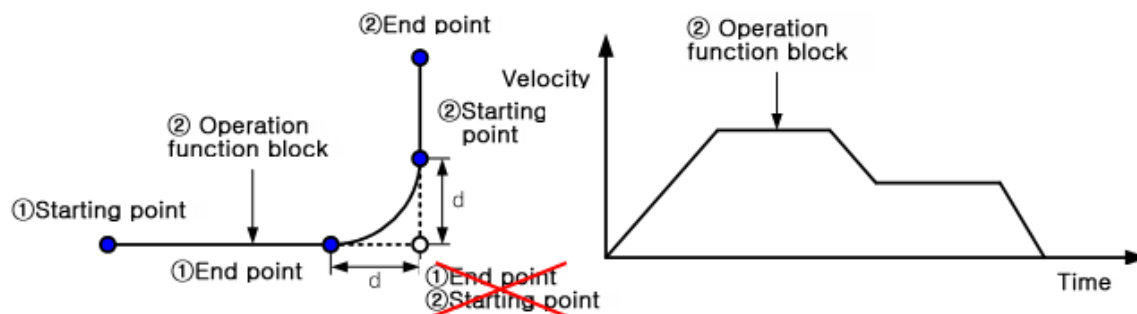
在当前执行的动作完成后执行下一个指令。

‘过渡模式’不体现。



(3) 混合

由于当前动作混合下一个动作，所以两个动作之间不停止。根据混合的模式(混合低，混合前一个，混合下一个，混合高)，速度可能改变。



※ 混合下一个情况下动作

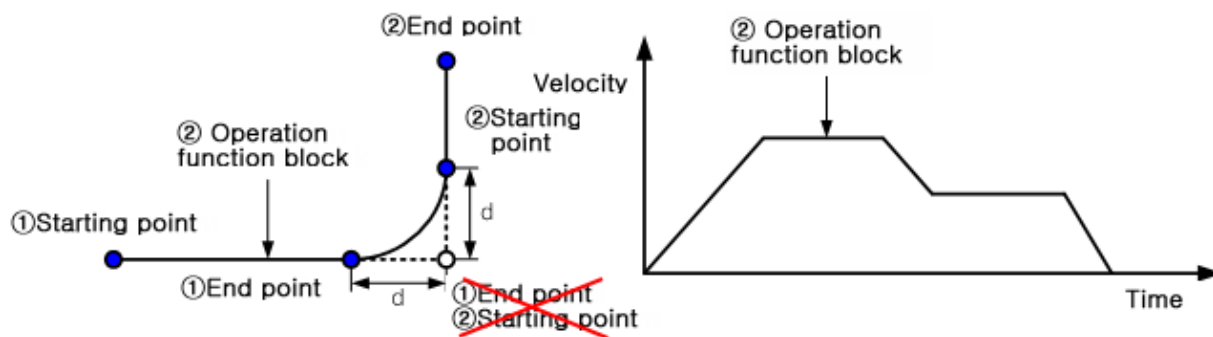
2. ‘过渡模式’

(1) TMNone

动作轨迹不改变，两个运行之间不插入曲线。
 如果缓冲区模式混合这个设置，运行缓冲模式。
 根据缓冲区模式的动作与以上异常停止和缓冲相同。

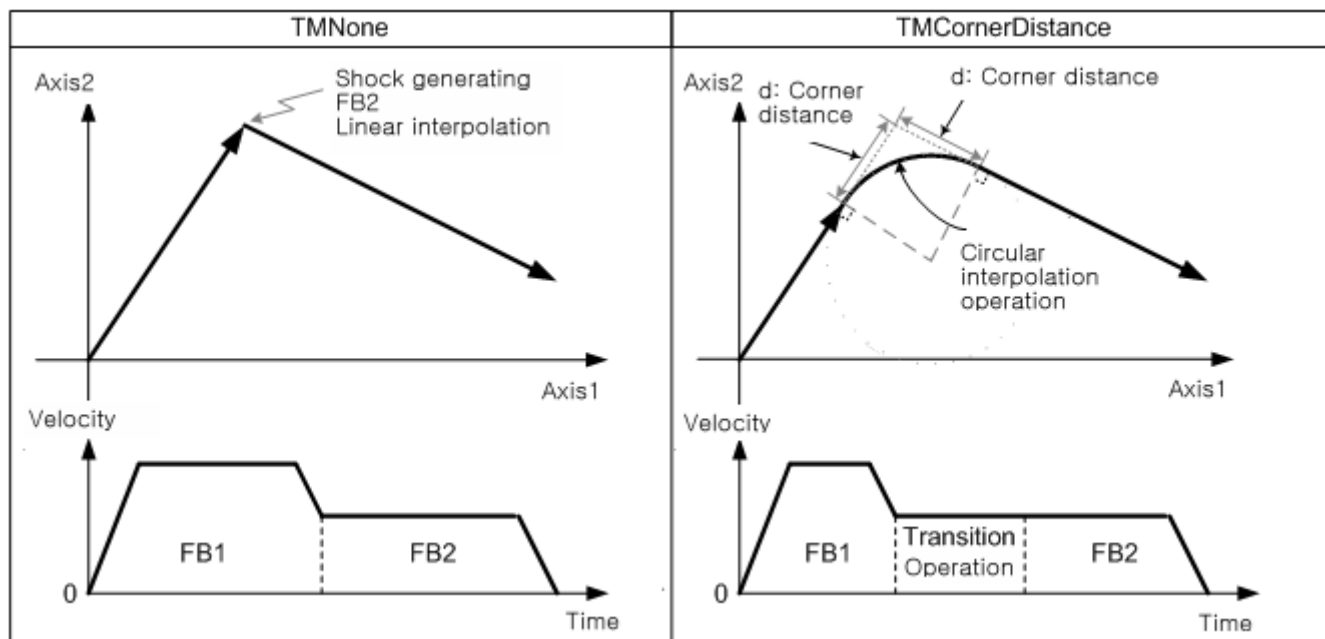
(2) TMCornerRadius

通过指定两个动作块角的距离插入曲线。转换速度通过缓冲模式指定。



- ※ 在混合下一个情况下动作
- ※ d: 在拐角处的曲线插入距离

(3) 过渡模式比较



8.2.11 同步控制

1. 传动运行

- (1) 传动运行根据设置比率使主轴（或者编码器）和从站速度同步。
- (2) 传动运行可通过齿轮运行取消指令而停止。
- (3) 传动比 (=速度同步率)计算如下。

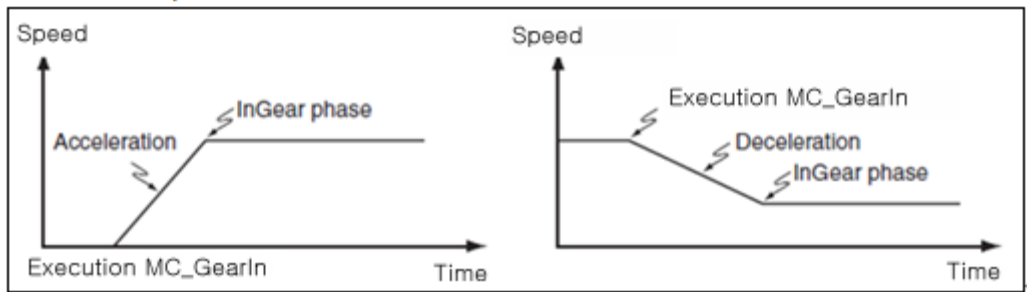
$$\text{齿轮比} = \text{主轴比}/\text{从轴比}$$

※ 主轴比<从轴比同样可以设置。

- (4) 从轴的旋转方向基于主轴的正方向。如果传动比是正数(>0)，正向旋转，如果是负数(< 0)，反向旋转。
- (5) 从轴的最终运行速度计算如下。

$$\begin{aligned} \text{从轴运行速度} &= \text{主轴运行速度} \times \text{传动比} \\ &= \text{主轴运行速度} \times \text{主轴比}/\text{从轴比} \end{aligned}$$

- (6) 可通过使用加速和减速输入设置，从传动运行加速/减速至目标速度。



- (7) 相关动作功能块

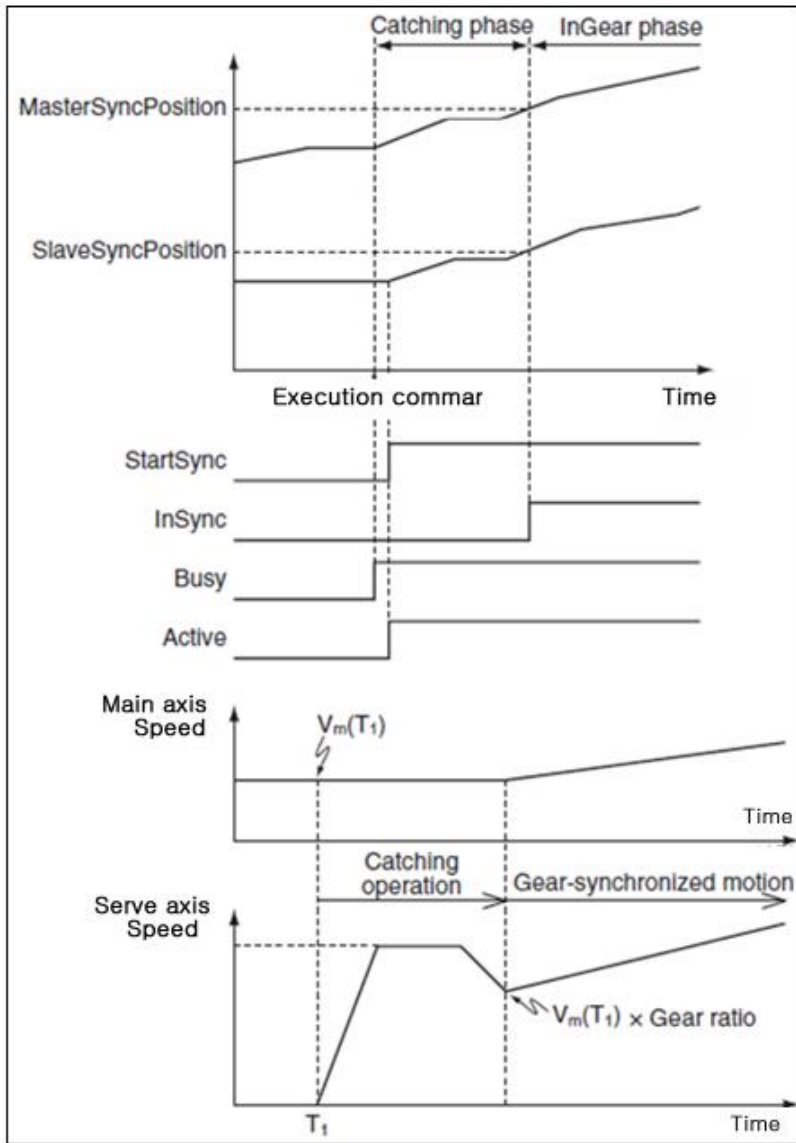
名称	描述	运行条件
MC_GearIn	传动运行	边沿
MC_GearIn		
BOOL	Execute	InGear
UINT	Master	Master
UINT	Slave	Slave
BOOL	ContinuousUpdate	Busy
INT	RatioNumerator	Active
UINT	RatioDenominator	CommandAborted
UINT	MasterValueSource	Error
LREAL	Acceleration	ErrorID
LREAL	Deceleration	
LREAL	Jerk	
UINT	BufferMode	

名称	描述	运行条件																																												
LS_VarGearIn	变量传动运行	边沿																																												
<div style="border: 1px solid black; padding: 5px; margin: 5px auto; width: fit-content;"> <p style="text-align: center;">LS_VarGearIn</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">BOOL</td> <td style="width: 40%;">Execute</td> <td style="width: 30%;">InGear</td> <td style="width: 10%;">BOOL</td> </tr> <tr> <td>UDINT</td> <td>VarOffset-----</td> <td>VarOffset</td> <td>UINT</td> </tr> <tr> <td>UINT</td> <td>Slave -----</td> <td>Slave</td> <td>UINT</td> </tr> <tr> <td>BOOL</td> <td>ContinousUpdate</td> <td>Busy</td> <td>BOOL</td> </tr> <tr> <td>INT</td> <td>RatioNumerator</td> <td>Active</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>RatioDenominator</td> <td>CommandAborted</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>MasterValueSource</td> <td>Error</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>Acceleration</td> <td>ErrorID</td> <td>WORD</td> </tr> <tr> <td>LREAL</td> <td>Deceleration</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Jerk</td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>BufferMode</td> <td></td> <td></td> </tr> </table> </div>			BOOL	Execute	InGear	BOOL	UDINT	VarOffset-----	VarOffset	UINT	UINT	Slave -----	Slave	UINT	BOOL	ContinousUpdate	Busy	BOOL	INT	RatioNumerator	Active	BOOL	UINT	RatioDenominator	CommandAborted	BOOL	UINT	MasterValueSource	Error	BOOL	LREAL	Acceleration	ErrorID	WORD	LREAL	Deceleration			LREAL	Jerk			UINT	BufferMode		
BOOL	Execute	InGear	BOOL																																											
UDINT	VarOffset-----	VarOffset	UINT																																											
UINT	Slave -----	Slave	UINT																																											
BOOL	ContinousUpdate	Busy	BOOL																																											
INT	RatioNumerator	Active	BOOL																																											
UINT	RatioDenominator	CommandAborted	BOOL																																											
UINT	MasterValueSource	Error	BOOL																																											
LREAL	Acceleration	ErrorID	WORD																																											
LREAL	Deceleration																																													
LREAL	Jerk																																													
UINT	BufferMode																																													

名称	描述	运行条件																				
MC_GearOut	传动脱离	边沿																				
<div style="border: 1px solid black; padding: 5px; margin: 5px auto; width: fit-content;"> <p style="text-align: center;">MC_GearOut</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">BOOL</td> <td style="width: 40%;">Execute</td> <td style="width: 30%;">Done</td> <td style="width: 10%;">BOOL</td> </tr> <tr> <td>UINT</td> <td>Slave -----</td> <td>Slave</td> <td>UINT</td> </tr> <tr> <td></td> <td></td> <td>Busy</td> <td>BOOL</td> </tr> <tr> <td></td> <td></td> <td>Error</td> <td>BOOL</td> </tr> <tr> <td></td> <td></td> <td>ErrorID</td> <td>WORD</td> </tr> </table> </div>			BOOL	Execute	Done	BOOL	UINT	Slave -----	Slave	UINT			Busy	BOOL			Error	BOOL			ErrorID	WORD
BOOL	Execute	Done	BOOL																			
UINT	Slave -----	Slave	UINT																			
		Busy	BOOL																			
		Error	BOOL																			
		ErrorID	WORD																			

2. 定位传动运行

- (1) 基于齿轮运行基本设置比率相同，定位传动运行使主轴（编码器）和从轴速度同步。
- (2) 主轴和从轴的同步起始位置可以指定。
- (3) 运行方式如下。



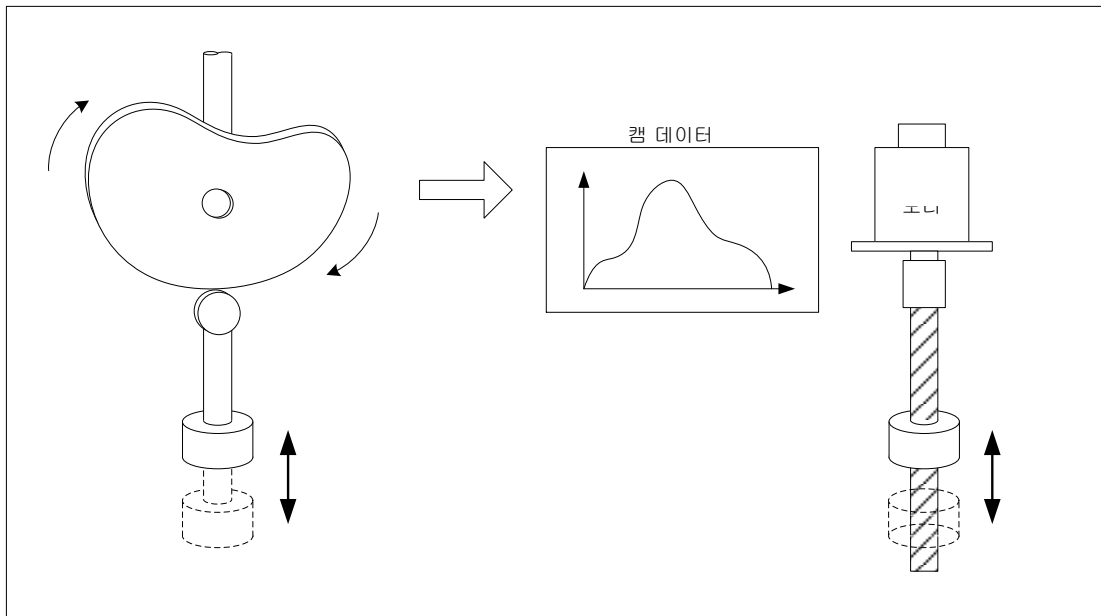
(4) 相关动作功能块

名称	描述	运行条件																																																																
MC_GearInPos	指定位置的传动运行	边沿																																																																
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="4" style="text-align: center;">MC_GearInPos</th> </tr> </thead> <tbody> <tr> <td>BOOL</td> <td>Execute</td> <td>InSync</td> <td>UINT</td> </tr> <tr> <td>UINT</td> <td>Master</td> <td>Master</td> <td>UINT</td> </tr> <tr> <td>UINT</td> <td>Slave</td> <td>Slave</td> <td>BOOL</td> </tr> <tr> <td>INT</td> <td>RatioNumerator</td> <td>StartSync</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>RatioDenominator</td> <td>Busy</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>MasterValueSource</td> <td>Active</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>MasterSyncPosition</td> <td>CommandAborted</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>SlaveSyncPosition</td> <td>Error</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>SyncMode</td> <td>ErrorID</td> <td>WORD</td> </tr> <tr> <td>LREAL</td> <td>MasterStartDistance</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Velocity</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Acceleration</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Deceleration</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Jerk</td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>BufferMode</td> <td></td> <td></td> </tr> </tbody> </table>			MC_GearInPos				BOOL	Execute	InSync	UINT	UINT	Master	Master	UINT	UINT	Slave	Slave	BOOL	INT	RatioNumerator	StartSync	BOOL	UINT	RatioDenominator	Busy	BOOL	UINT	MasterValueSource	Active	BOOL	LREAL	MasterSyncPosition	CommandAborted	BOOL	LREAL	SlaveSyncPosition	Error	BOOL	UINT	SyncMode	ErrorID	WORD	LREAL	MasterStartDistance			LREAL	Velocity			LREAL	Acceleration			LREAL	Deceleration			LREAL	Jerk			UINT	BufferMode		
MC_GearInPos																																																																		
BOOL	Execute	InSync	UINT																																																															
UINT	Master	Master	UINT																																																															
UINT	Slave	Slave	BOOL																																																															
INT	RatioNumerator	StartSync	BOOL																																																															
UINT	RatioDenominator	Busy	BOOL																																																															
UINT	MasterValueSource	Active	BOOL																																																															
LREAL	MasterSyncPosition	CommandAborted	BOOL																																																															
LREAL	SlaveSyncPosition	Error	BOOL																																																															
UINT	SyncMode	ErrorID	WORD																																																															
LREAL	MasterStartDistance																																																																	
LREAL	Velocity																																																																	
LREAL	Acceleration																																																																	
LREAL	Deceleration																																																																	
LREAL	Jerk																																																																	
UINT	BufferMode																																																																	

名称	描述	运行条件																																																												
LS_VarGearInPos	通过指定位置的变量传动	边沿																																																												
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="4" style="text-align: center;">LS_VarGearInPos</th> </tr> </thead> <tbody> <tr> <td>BOOL</td> <td>Execute</td> <td>InGear</td> <td>BOOL</td> </tr> <tr> <td>UDINT</td> <td>VarOffset</td> <td>VarOffset</td> <td>UINT</td> </tr> <tr> <td>UINT</td> <td>Slave</td> <td>Slave</td> <td>UINT</td> </tr> <tr> <td>INT</td> <td>RatioNumerator</td> <td>Busy</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>RatioDenominator</td> <td>Active</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>MasterValueSource</td> <td>CommandAborted</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>MasterSyncPosition</td> <td>Error</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>SlaveSyncPosition</td> <td>ErrorID</td> <td>WORD</td> </tr> <tr> <td>UINT</td> <td>SyncMode</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>MasterStartDistance</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Acceleration</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Deceleration</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Jerk</td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>BufferMode</td> <td></td> <td></td> </tr> </tbody> </table>			LS_VarGearInPos				BOOL	Execute	InGear	BOOL	UDINT	VarOffset	VarOffset	UINT	UINT	Slave	Slave	UINT	INT	RatioNumerator	Busy	BOOL	UINT	RatioDenominator	Active	BOOL	UINT	MasterValueSource	CommandAborted	BOOL	LREAL	MasterSyncPosition	Error	BOOL	LREAL	SlaveSyncPosition	ErrorID	WORD	UINT	SyncMode			LREAL	MasterStartDistance			LREAL	Acceleration			LREAL	Deceleration			LREAL	Jerk			UINT	BufferMode		
LS_VarGearInPos																																																														
BOOL	Execute	InGear	BOOL																																																											
UDINT	VarOffset	VarOffset	UINT																																																											
UINT	Slave	Slave	UINT																																																											
INT	RatioNumerator	Busy	BOOL																																																											
UINT	RatioDenominator	Active	BOOL																																																											
UINT	MasterValueSource	CommandAborted	BOOL																																																											
LREAL	MasterSyncPosition	Error	BOOL																																																											
LREAL	SlaveSyncPosition	ErrorID	WORD																																																											
UINT	SyncMode																																																													
LREAL	MasterStartDistance																																																													
LREAL	Acceleration																																																													
LREAL	Deceleration																																																													
LREAL	Jerk																																																													
UINT	BufferMode																																																													

3. Cam运行

- (1) CAM运行通过转换机械cam模块控制cams到cam轮廓中设置的cam，并且同步数据到作为主轴的指定电机位置。
- (2) 可以通过使用cam轮廓中cam数据的软件cam动作替换之前的机械cam运行。



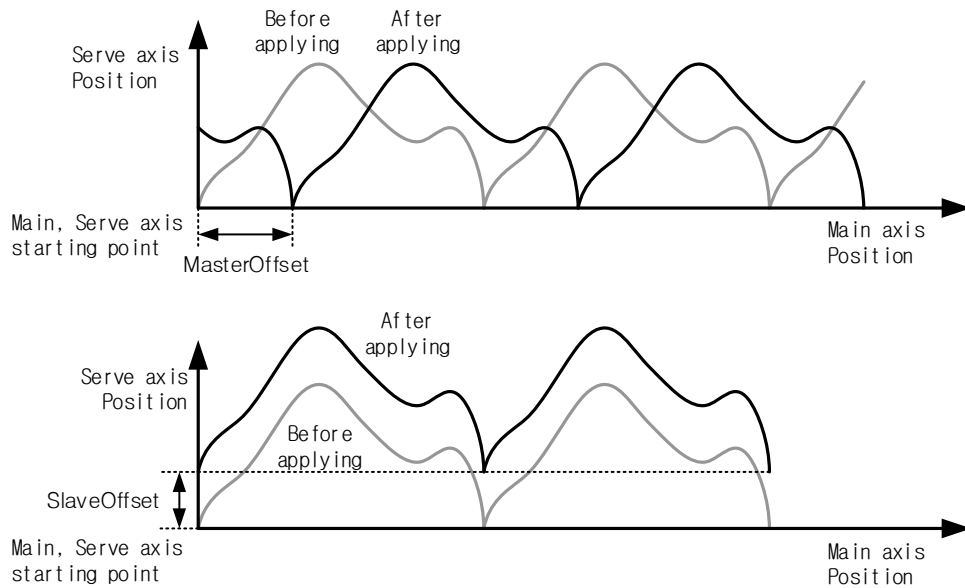
- (3) 总共可以指定32个cam轮廓，每一个都可以应用到一个轴，彼此互不干涉。
- (4) 每个cam轮廓由100个cam数据组成。
- (5) 为了停止cam运行，需要在子轴发布MC_CamOut指令，或者运行其他运动功能块(对于异常停止情况)。

(6) Cam运作指令的二级数据

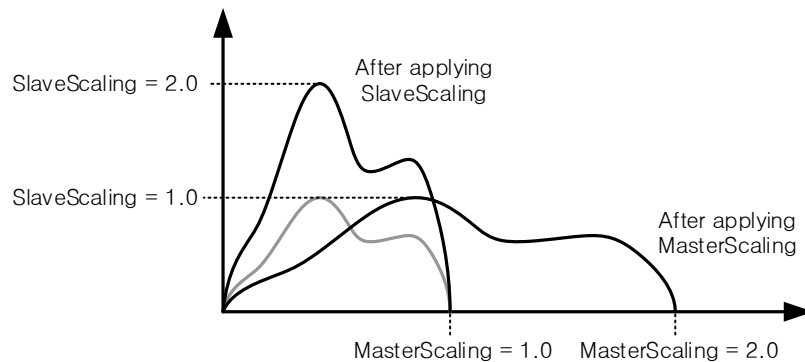
变量	描述
Master	设置主站轴 (1~32: 实轴, 37~40: 虚轴, 41~42: 编码器),
MasterOffset	设置主站轴补偿值.
SlaveOffset	设置从站 cam 表补偿值.
MasterScaling	指定主轴放大倍率.
SlaveScaling	指定从轴 cam 表放大倍率.
MasterStartDistance	指定主轴启动cam运行的主轴位置.
MasterSyncPosition	当cam运行启动时, 在cam表指定起始位置.
StartMode	设置cam运行模式. 0 : Cam表应用于一个绝对值. (mcAbsolute) 1 : Cam表基于指令启动位置, 应用于一个绝对值. (mcRelative)
MasterValueSource	选择用于cam运行的主轴源. 0 : 同步主轴目标值. 1 : 同步从轴当前值.
CamTableID	指定运行的 cam 表.

(a) 在MasterOffset和SlaveOffset中，设置应用的cam表补偿。MasterOffset确定从主轴起始点的补偿，而Slaveoffset确定从子轴起始点的补偿。请参考下图。使用补偿可能变更用于cam运行的起始位置，导致突变的

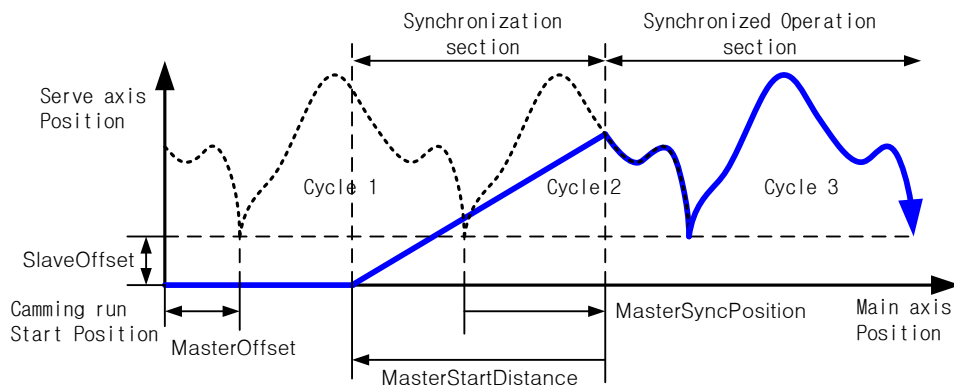
运行. 在此情况下, 应使用MasterSyncPosition, MasterStartDistance.



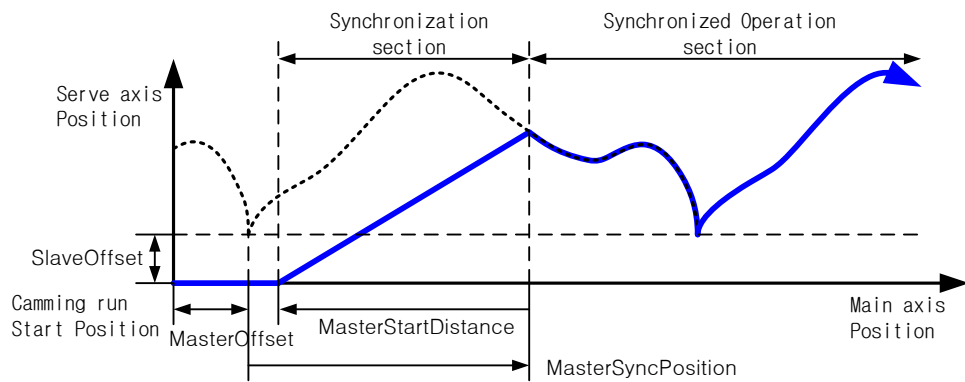
- (b) 在MasterScaling和SlaveScaling中, 设置应用的cam数据比例尺. MasterScaling确定主轴数据比例尺, SlaveScaling确定子轴数据比例尺. 请参考下图.



- (c) MasterSyncPosition 输入指定主轴位置, 该位置在实际cam运行同步完成的cam表的范围内, MasterStartDistance输入指定同步启动的主轴相对位置. 如果无法如下以循环1启动同步运行(如果启动位置到同步运行起始点的距离小于MasterStartDistance), 同步运行以循环2启动.



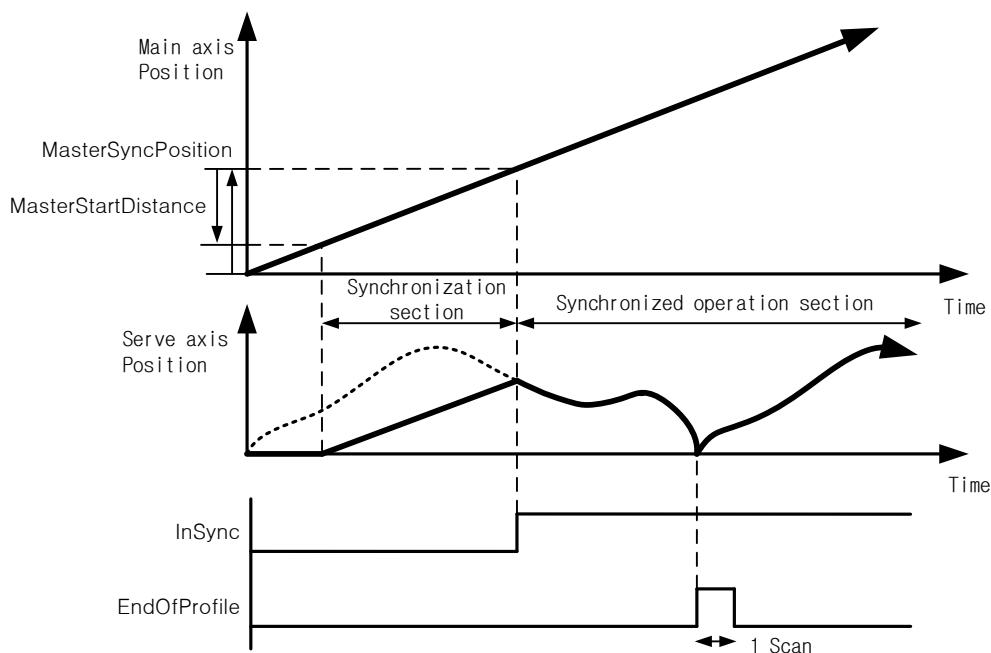
对于MasterScaling为1.0的情况



对于MasterScaling为2.0的情况

MasterSyncPosition位置基于cam表范围内的位置，实际同步位置由MasterOffset和MasterScale参数决定。从轴以输入值距离起始移动到同步位置，该输入距离基于实际应用的MasterSyncPosition位置。如果在起始移动之前，从轴以停止状态在相对位置等待，如果在指定起始的时候，从站已经移动到同步位置，通过表的长度返回同步启动点的位置，直到超出MasterStartDistance范围。实际同步位置根据MasterScaling和SlaveScaling发生变化，由于MasterSyncPosition是基于cam表内部的值，但是MasterOffset 和MasterStartDistance值保持不变。

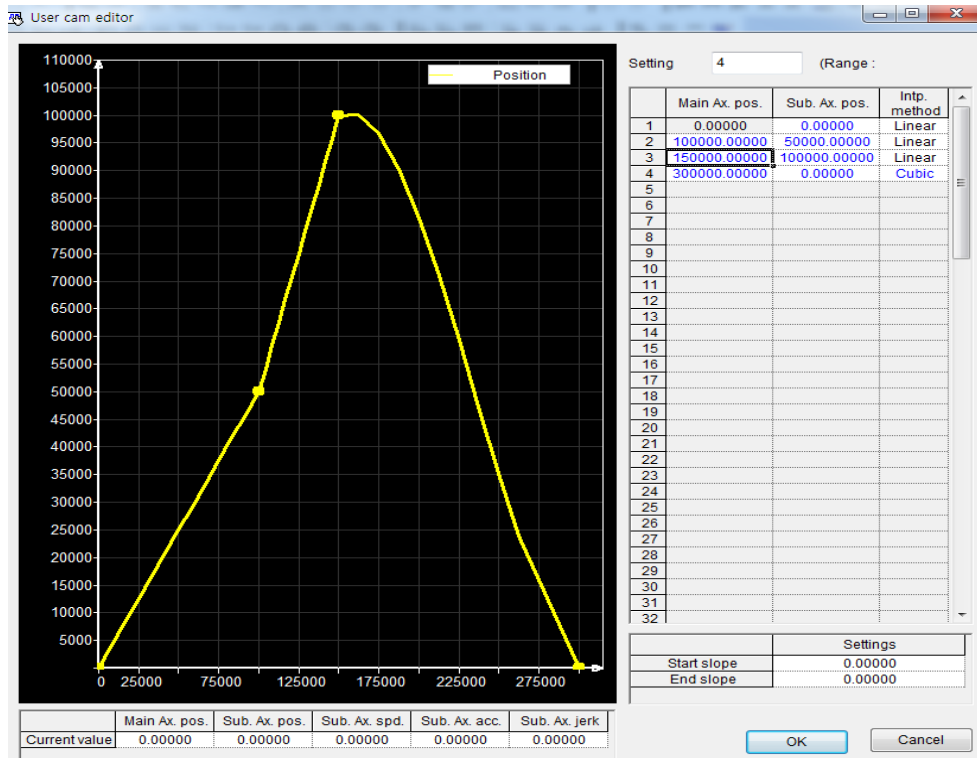
(d) 当cam启动同步时InSyncoutput为on. EndOfProfileoutput 1扫描为on，无论单个cam表运行何时完成。



- (e) 在StartMode中, 设置cam运行模式. 设置范围0或1. 如果输入值超出设置范围, 错误发生. 如果设置为0, cam表起始位置设置为0的主轴位置. 如果设置为1, cam表起始位置设置到主轴当前位置.
- (f) MasterValueSource选择同步主轴的源. 如果设置为0, 子轴执行基于通过运动控制模块计算的主轴指令位置的cam运行, 设置为1, 则执行基于从主轴伺服驱动通讯接收的当前位置的cam运行.
- (g) CamTableID设置应用于cam运行的cam表编号. 设置范围1到32. 如果输入值超出设置范围, 在运动功能块中发生错误"0x1115".

(7) Cam轮廓

部分		描述
CAM数据	主轴位置	设置对应于主轴的子轴cam位置
	从轴位置	
	插值类型	设置cam数据间的特性曲线. (线性, 立体)
启动斜坡	如果第一个或者最后一个部分的插值类型设置为'立体', 设置立体运行的启动斜坡和结束斜坡.	
结束斜坡		



(8) 运动功能块

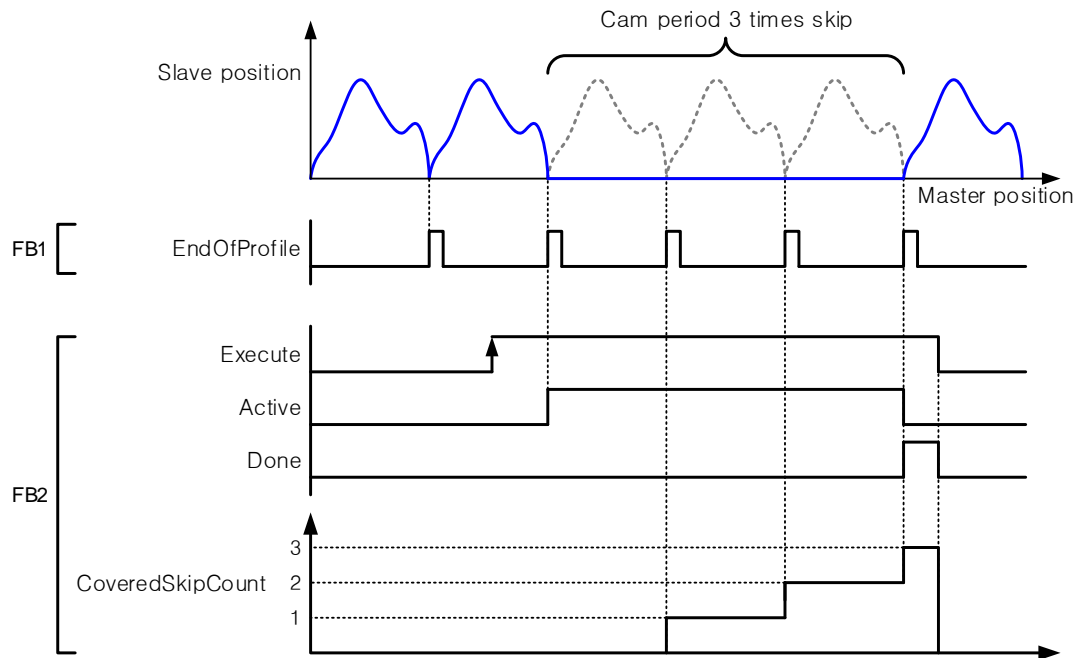
名称	描述	运行条件
MC_CamIn	Cam运行	边沿
MC_CamIn		
BOOL	Execute	InSync
UINT	Master	Master
UINT	Slave	Slave
LREAL	ContinousUpdate	Busy
LREAL	MasterOffset	Active
LREAL	SlaveOffset	CommandAborted
LREAL	MasterScaling	Error
LREAL	SlaveScaling	ErrorID
LREAL	MasterStartDistance	EndOfProfile
LREAL	MasterSyncPosition	
UINT	StartMode	
UINT	MasterValueSource	
UINT	CamTableID	
UINT	BufferMode	

名称	描述	运行条件																																																											
LS_VarCamIn	变量Cam运行	边沿																																																											
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="3">LS_VarCamIn</th> </tr> </thead> <tbody> <tr> <td>BOOL</td> <td>Execute</td> <td>InSync</td> <td>BOOL</td> </tr> <tr> <td>UDINT</td> <td>VarOffset</td> <td>VarOffset</td> <td>UINT</td> </tr> <tr> <td>UINT</td> <td>Slave</td> <td>Slave</td> <td>UINT</td> </tr> <tr> <td>LREAL</td> <td>ContinuousUpdate</td> <td>Busy</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>MasterOffset</td> <td>Active</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>SlaveOffset</td> <td>CommandAborted</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>MasterScaling</td> <td>Error</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>SlaveScaling</td> <td>ErrorID</td> <td>WORD</td> </tr> <tr> <td>LREAL</td> <td>MasterStartDistance</td> <td>EndOfProfile</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>MasterSyncPosition</td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>StartMode</td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>MasterValueSource</td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>CamTableID</td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>BufferMode</td> <td></td> <td></td> </tr> </tbody> </table>			LS_VarCamIn			BOOL	Execute	InSync	BOOL	UDINT	VarOffset	VarOffset	UINT	UINT	Slave	Slave	UINT	LREAL	ContinuousUpdate	Busy	BOOL	LREAL	MasterOffset	Active	BOOL	LREAL	SlaveOffset	CommandAborted	BOOL	LREAL	MasterScaling	Error	BOOL	LREAL	SlaveScaling	ErrorID	WORD	LREAL	MasterStartDistance	EndOfProfile	BOOL	LREAL	MasterSyncPosition			UINT	StartMode			UINT	MasterValueSource			UINT	CamTableID			UINT	BufferMode		
LS_VarCamIn																																																													
BOOL	Execute	InSync	BOOL																																																										
UDINT	VarOffset	VarOffset	UINT																																																										
UINT	Slave	Slave	UINT																																																										
LREAL	ContinuousUpdate	Busy	BOOL																																																										
LREAL	MasterOffset	Active	BOOL																																																										
LREAL	SlaveOffset	CommandAborted	BOOL																																																										
LREAL	MasterScaling	Error	BOOL																																																										
LREAL	SlaveScaling	ErrorID	WORD																																																										
LREAL	MasterStartDistance	EndOfProfile	BOOL																																																										
LREAL	MasterSyncPosition																																																												
UINT	StartMode																																																												
UINT	MasterValueSource																																																												
UINT	CamTableID																																																												
UINT	BufferMode																																																												

名称	描述	运行条件																							
MC_CamOut	Cam运行输出	边沿																							
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="3">MC_CamOut</th> </tr> </thead> <tbody> <tr> <td>BOOL</td> <td>Execute</td> <td>Done</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>Slave</td> <td>Slave</td> <td>UINT</td> </tr> <tr> <td></td> <td></td> <td>Busy</td> <td>BOOL</td> </tr> <tr> <td></td> <td></td> <td>Error</td> <td>BOOL</td> </tr> <tr> <td></td> <td></td> <td>ErrorID</td> <td>WORD</td> </tr> </tbody> </table>			MC_CamOut			BOOL	Execute	Done	BOOL	UINT	Slave	Slave	UINT			Busy	BOOL			Error	BOOL			ErrorID	WORD
MC_CamOut																									
BOOL	Execute	Done	BOOL																						
UINT	Slave	Slave	UINT																						
		Busy	BOOL																						
		Error	BOOL																						
		ErrorID	WORD																						

4. Cam跳跃

- (1) 该功能以用户在cam运行轴中设置的数量跳跃cam运行.
- (2) 当Cam跳跃指令在cam运行的子轴中发布时,当前cam循环结束, 并且跳跃运行启动. 子轴为在cam跳跃动作的cam表结束位置进行准备.



- (3) 通过MC_CamIn指令(FB1)执行cam动作后, 如果使用LS_CamSkip指令(FB2)跳过3个循环, 每个功能块FB1, FB2输出和cam子轴动作在下图中显示
- (4) 如果在cam跳跃动作中再次执行Cam跳跃指令, 或者cam跳跃动作由于其他cam跳跃指令而终止, 应用下一个Cam跳跃指令的SkimCount, 并从开始启动一个新的cam跳跃动作. 在这种情况下, 再次执行时的跳跃循环数量包含在再次执行后的跳跃循环中. 因此, CoveredSkipCount值是1点大于用户设置的SkipCount.
- (5) 即使在执行cam跳跃动作结束之前执行触点为off, 直到运行完成前激活触点为on. Done和CoveredKipCount仅应用于运行完成后的扫描执行.
- (6) 动作功能块

名称	描述	运行条件																																								
LS_CamSkip	Cam跳跃	边沿																																								
<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> <p style="text-align: center;">LS_CamSkip</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">BOOL</td> <td>Execute</td> <td style="width: 30%;"></td> <td>Done</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>Slave</td> <td>-----</td> <td>Slave</td> <td>UINT</td> </tr> <tr> <td>UINT</td> <td>SkipCount</td> <td></td> <td>Busy</td> <td>BOOL</td> </tr> <tr> <td></td> <td></td> <td></td> <td>Active</td> <td>BOOL</td> </tr> <tr> <td></td> <td></td> <td></td> <td>CommandAborted</td> <td>BOOL</td> </tr> <tr> <td></td> <td></td> <td></td> <td>Error</td> <td>BOOL</td> </tr> <tr> <td></td> <td></td> <td></td> <td>ErrorID</td> <td>WORD</td> </tr> <tr> <td></td> <td></td> <td></td> <td>CoveredSkipCount</td> <td>UINT</td> </tr> </table> </div>			BOOL	Execute		Done	BOOL	UINT	Slave	-----	Slave	UINT	UINT	SkipCount		Busy	BOOL				Active	BOOL				CommandAborted	BOOL				Error	BOOL				ErrorID	WORD				CoveredSkipCount	UINT
BOOL	Execute		Done	BOOL																																						
UINT	Slave	-----	Slave	UINT																																						
UINT	SkipCount		Busy	BOOL																																						
			Active	BOOL																																						
			CommandAborted	BOOL																																						
			Error	BOOL																																						
			ErrorID	WORD																																						
			CoveredSkipCount	UINT																																						

8.2.12 手动控制

1. 点动运行

- (1) 点动运行通过用户手动点动指令进行位置控制。
- (2) 点动运行甚至可以在轴初始状态不确定的状态下进行。
- (3) 点动指令可以在初始确定或者未确定状态下执行，确保可以监控轴位置值的改变。
- (4) 加速/减速进程和点动速度

关于进程加速和减速，加速和减速控制基于[运行参数- 扩展参数]项目中的点动加速/减速/加速度设置。
点动速度在[运行参数- 扩展参数]项目中的点动高速和低速设置。

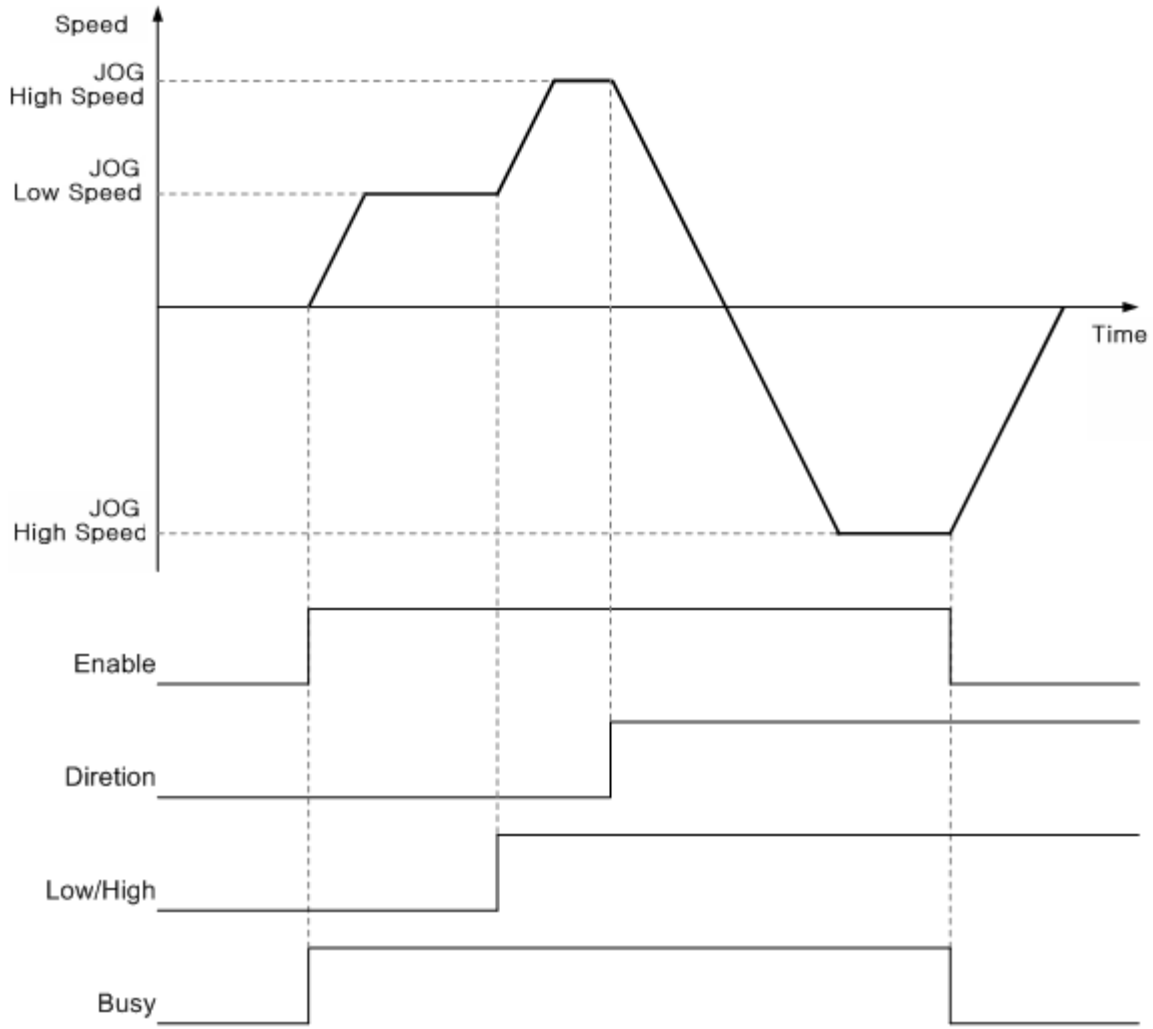
- (5) 点动高速应该设置在[运行参数- 扩展参数]项目中的速度限制或者不低于点动低速。
- (6) 相关参数设置

项目	设置	初始值
点动高速	长实数(LREAL)正数	100000 pls/s
点动低速		10000 pls/s
点动加速	0或者长实数(LREAL)正数	100000 pls/s ²
点动减速		100000 pls/s ²
点动加速度		0 pls/s ³

- (7) 运动功能块

名称	描述	运行条件
MC_Jog	点动运行	水平
LS_Jog		
BOOL — Enable	Enabled	BOOL
UINT — Axis	Axis	UINT
BOOL — Direction	Busy	BOOL
BOOL — Low/High	Error	BOOL
	ErrorID	WORD

(8) 运行时序



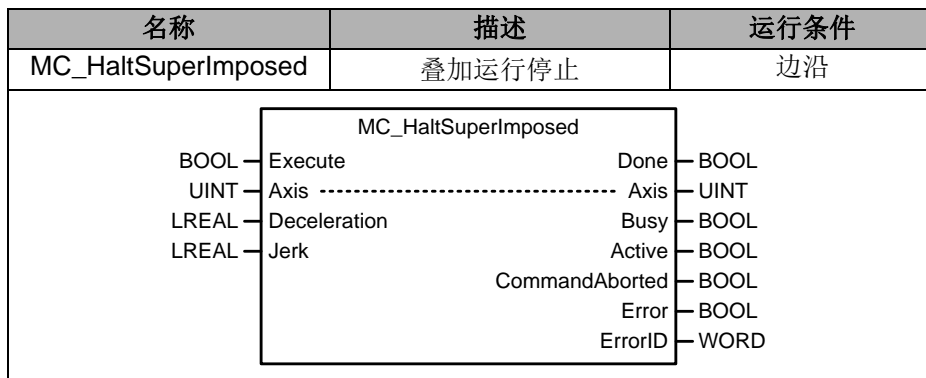
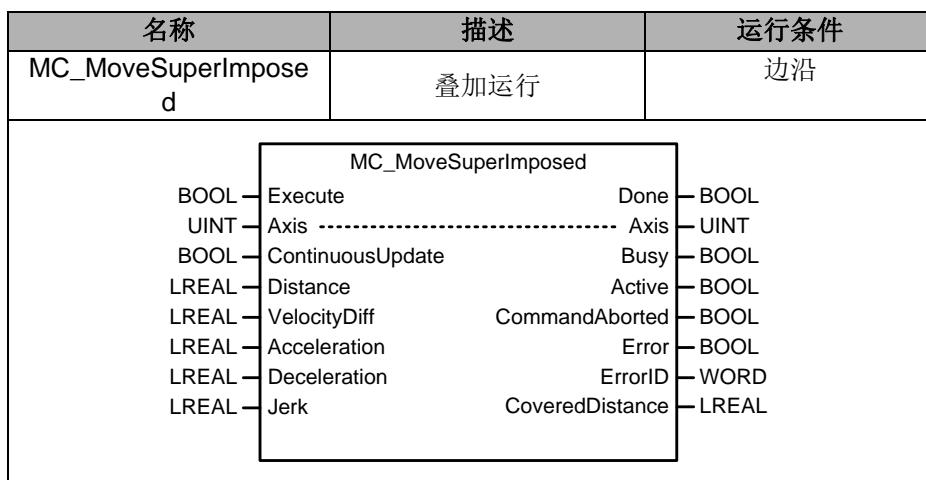
8.2.13 叠加运行

叠加运行执行按照在当前动作运行中指定移动距离的额外定位控制。

1. 控制特性

- (1) 当执行叠加运行指令时，轴移动从指令执行点到距离输入中指定目标距离的位置。
- (2) 移动方向由距离输入中指定目标距离的符号所决定，如果移动方向为正数(+或者无符号)，表示正向移动，如果是负数(-)，表示反向移动。
- (3) 当前动作没有取消，但是根据叠加运行进行叠加处理。
- (4) 即使当当前动作完成时，叠加运行持续运行，除非移动数量没有到达叠加运行中指定的数量。
- (5) 如果轴不是运行状，但是在"StandStill"状态，叠加运行以 MC_MoveRelative 相同运行方式执行。
- (6) 当前叠加运行可用通过 MC_HaltSuperImposed 指令停止。执行指令后，叠加运行在给加速和加速度时减速和停止。当前执行的当前动作不影响。

2. 运动功能块

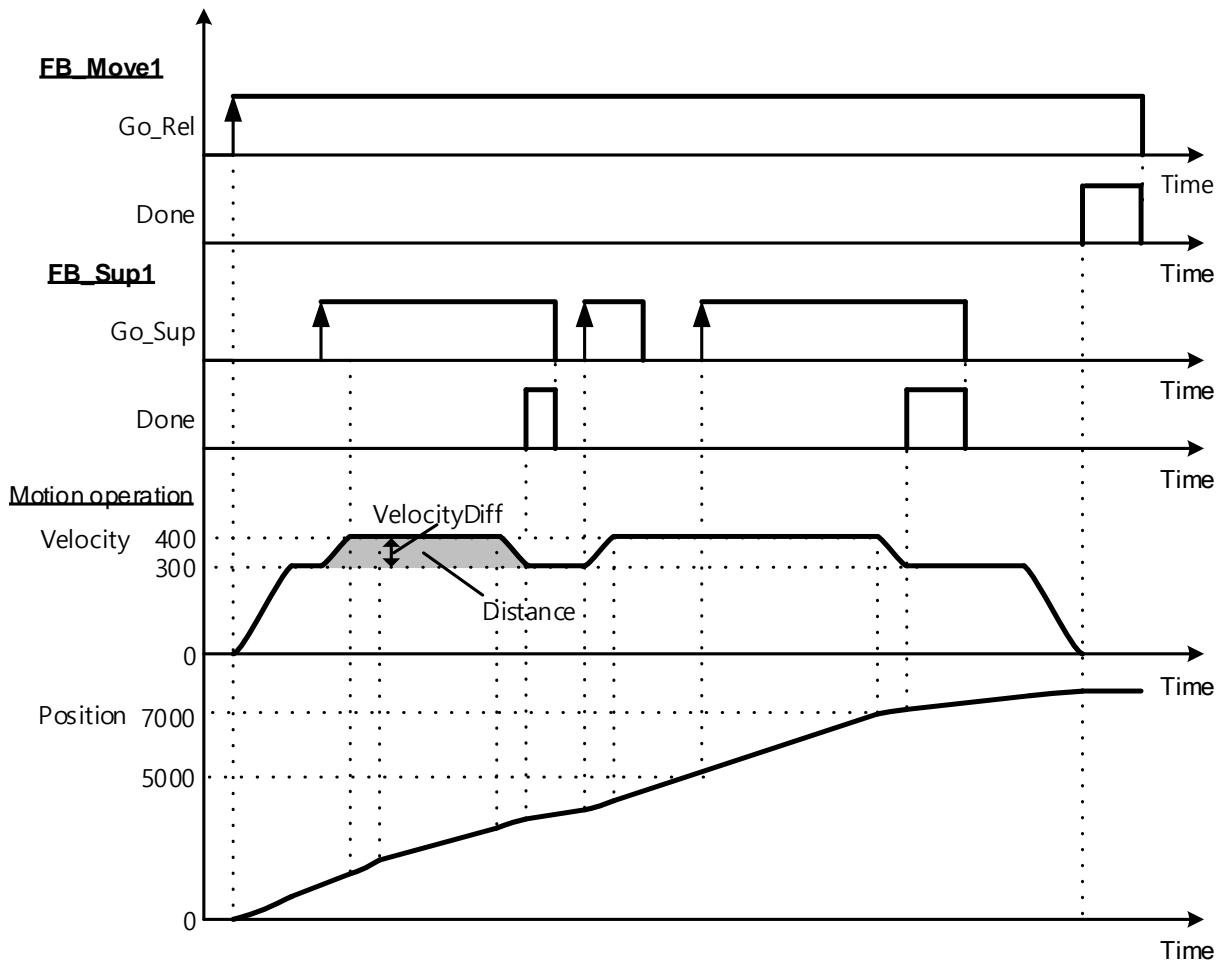
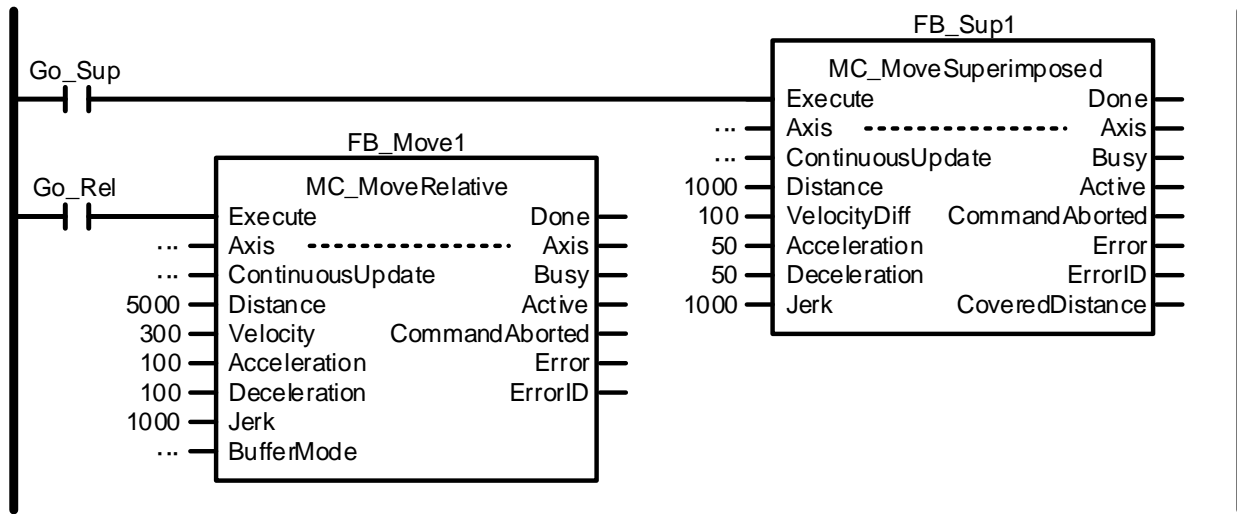


3. 限制

在以下情况下，叠加运行可能导致错误而无法执行。

- (1) 在速率控制或者转矩控制运行中执行叠加运行(错误代码: 0x1082)
- (2) 当叠加运行不执行时执行MC_HaltSuperImposed指令(错误代码: 0x1083)

4. 运行时序



8.2.14相位校正控制

相位校正控制在同步控制运行中执行主轴相位校正。执行主轴位置的虚拟移动，该主轴的子轴参考同步控制运行，并且子轴执行同步运行到移动的主轴位置。

1. 控制特性

- (1) 相位校正顺序可以按照齿轮运行或者 cam 运行的同步运行轴执行。
- (2) 相位校正不改变实际指令位置或者主轴当前位置，参考同步控制运行的子轴执行主轴位置相位校正。换句话说，主轴不了解子轴执行的相位校正。相位校正速率与当前主轴运行速率相关。
- (3) 如果主轴为编码器，当执行相位校正控制时，运行使用子轴速率限制。
- (4) 子轴参考同步运行的主轴位置是"实际主轴位置+ 相位校正控制位置。"
- (5) 如果在相位校正运行期间再次执行指令，从当前位置再次执行相位校正。也就是说，以相关值运行 PhaseShift。
- (6) 如果相位校正运行期间通过设置 PhaseShiftto 0 再次执行相位校正，当前相位校正运行立即停止。

2. 运动功能块

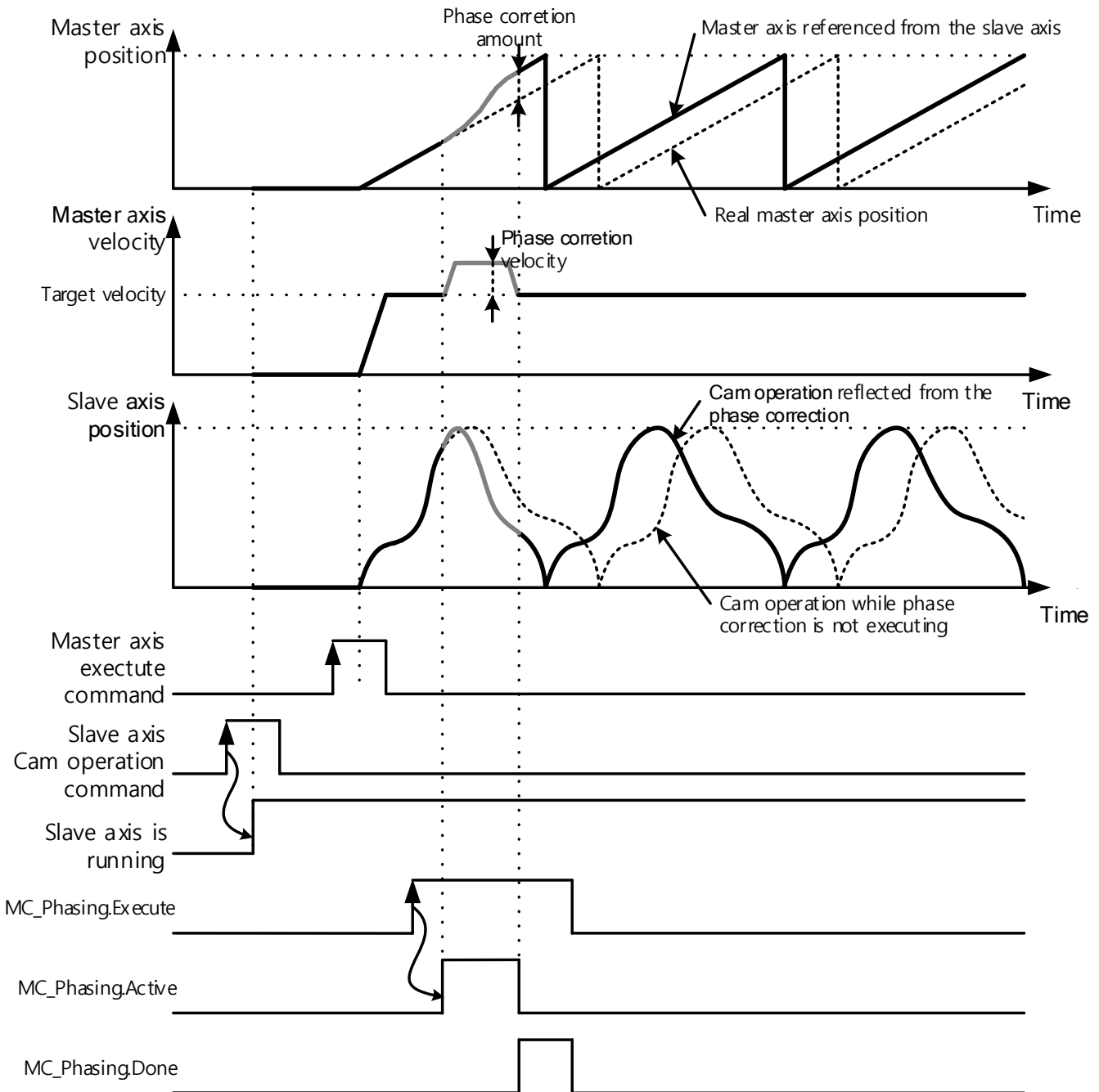
名称	描述	运行条件
MC_Phasing	相位校正	边沿
MC_Phasing		
BOOL	Execute	Done
UINT	Master	Master
UINT	Slave	Slave
LREAL	PhaseShift	Busy
LREAL	Velocity	Active
LREAL	Acceleration	CommandAborted
LREAL	Deceleration	Error
LREAL	Jerk	ErrorID
		CoveredPhaseShift

3. 限制

在以下情况下，相位校正可能导致错误而无法执行。

- (1) 子轴不执行同步控制运行(错误代码: 0x1130)
- (2) 指定主轴实际同步运行的主轴(错误代码: 0x1131)
- (3) PhaseShift超出脉冲单位位置表达范围(INT) (错误代码: 0x1132)
- (4) 速率设置小于0, 或者超出主轴速率限制(错误代码: 0x1133)
- (5) 加速设置小于0 (错误代码: 0x1014)
- (6) 减速设置小于0 (错误代码: 0x1015)
- (7) 加速度设置小于0 (错误代码: 0x1016)

4. 运行时序



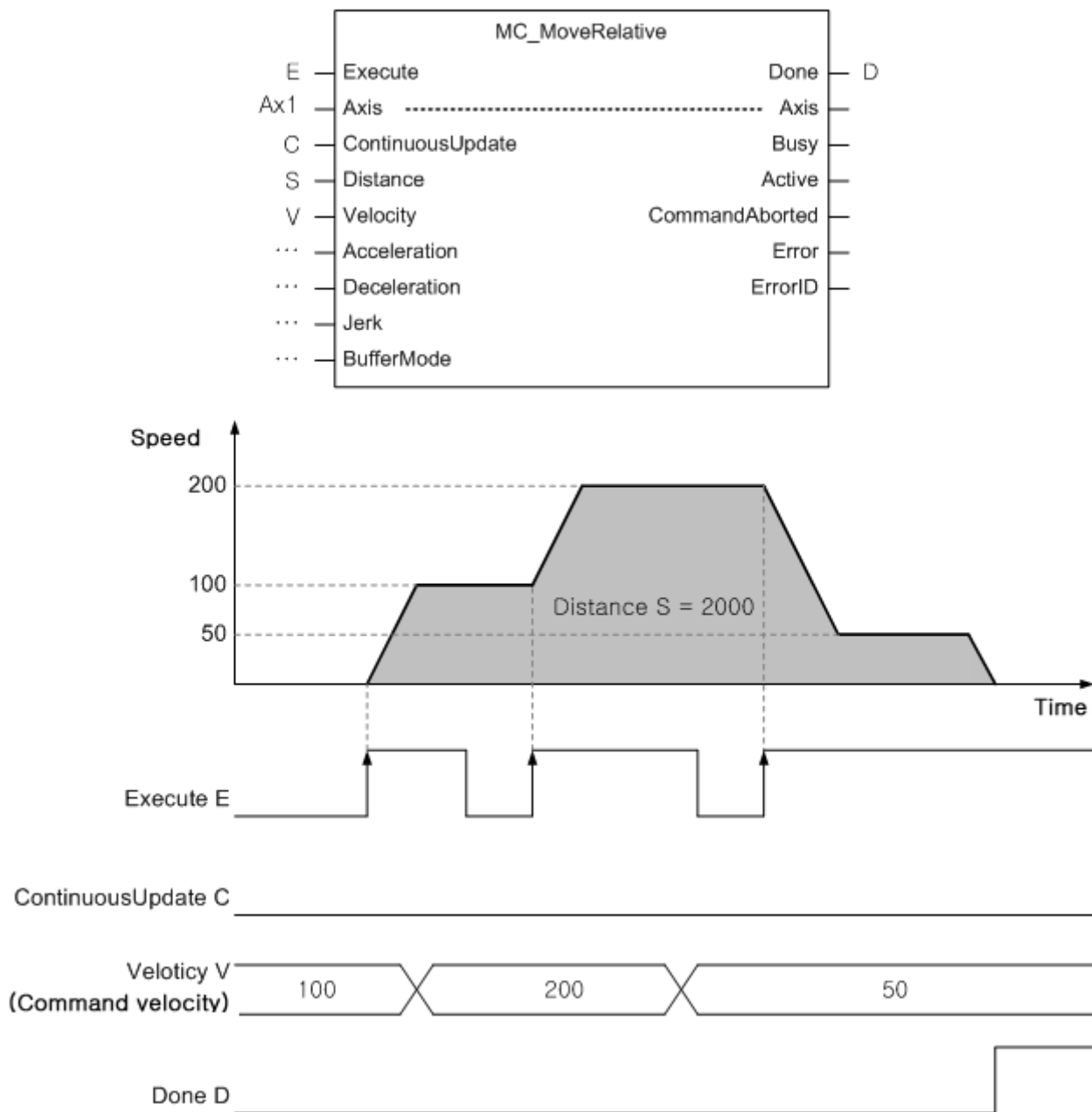
8.3 其他功能

8.3.1 转换控制功能

1. 执行动作功能块的输入变量改变

- (1) 如果动作功能块中没有持续更新输入，或者当持续更新输入为off时执行（执行输入使能），当应用执行输入为On（上升沿）时，通过输入动作功能块运行。运行期间通过改变动作功能块的输入，改变输入值后使执行输入为On，改变值立即应用于运行。

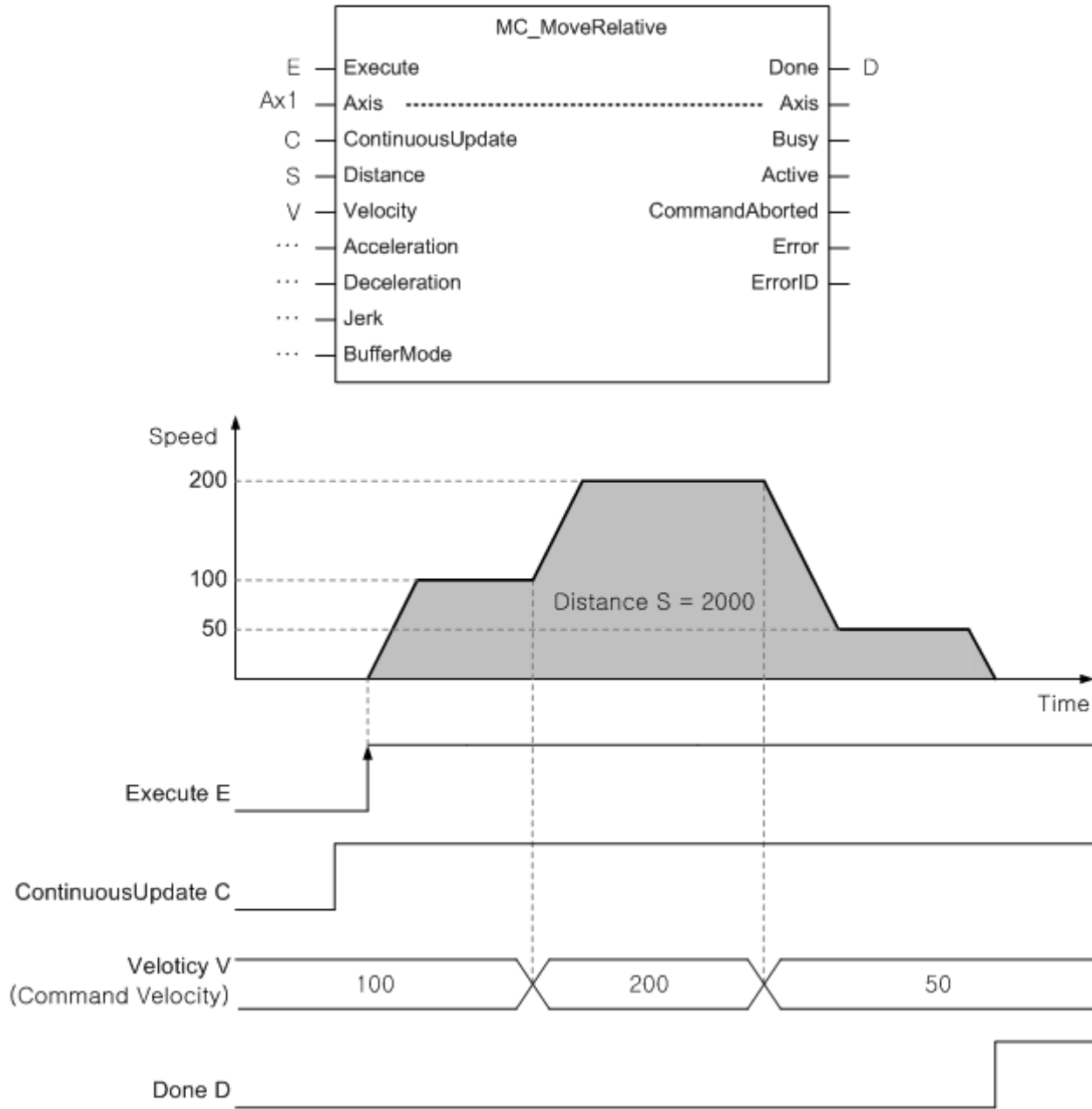
[例] 通过再次执行进行运动控制指令的输入变量更新



第8章 运动控制功能

- (2) 如果在动作控制块的边沿运行中持续更新输入为On，当执行输入为On（上升沿）时输入应用于动作功能块，如果输入改变并当持续更新输入为On，动作功能块产生一个体现改变的动作。然而，在动作功能块运行完成或者停止（繁忙输出禁止）后，不在体现输入改变。

[例]当持续更新为On时，动作控制指令输入变量更新。

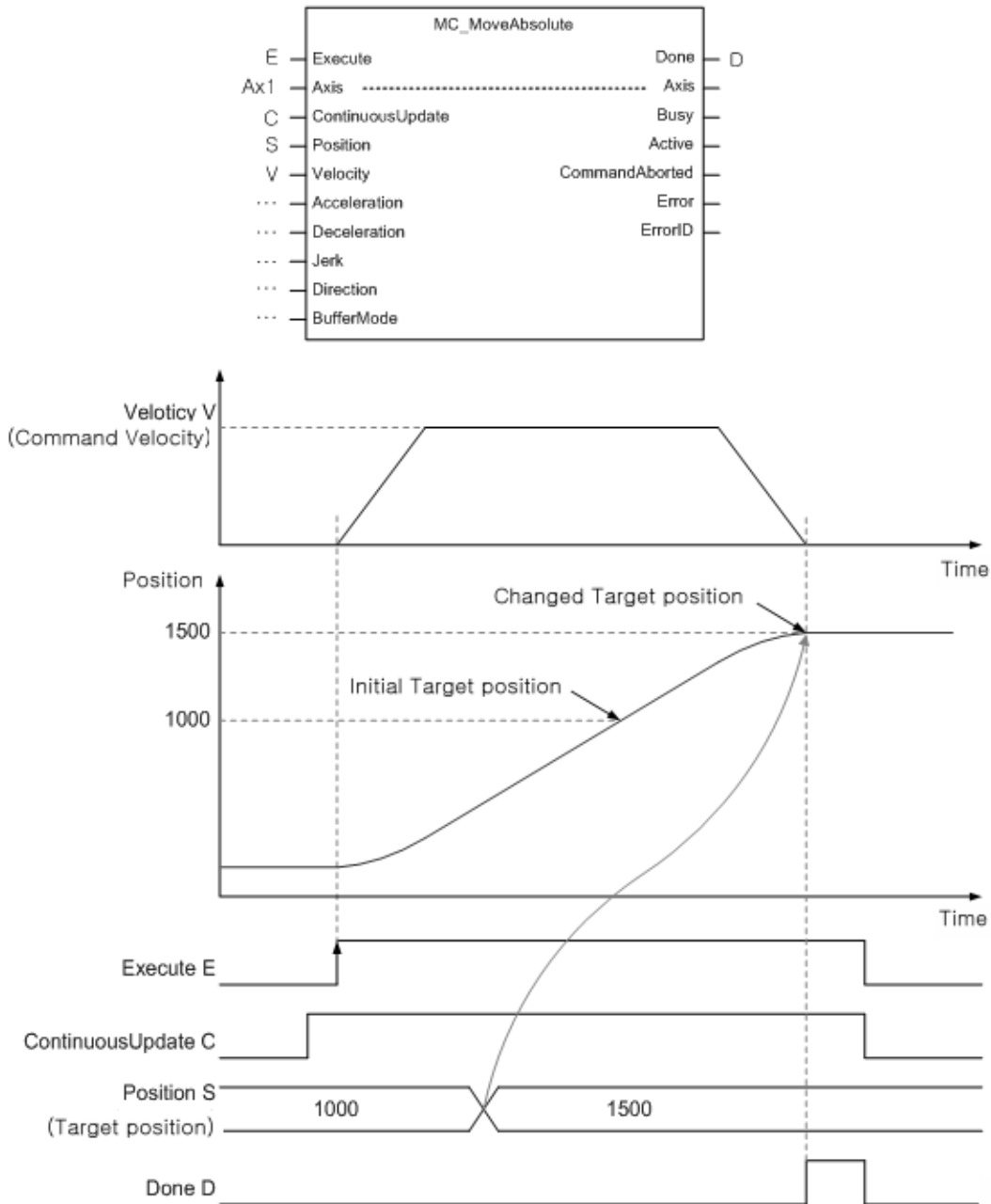


- (3) 如果是水平运行动作功能块，当使能输入为On时输入变量适用于运行，当使能输入为On时，输入变量可以持续变更。

2. 位置重置

- (1) 该功能用于位置运行中重置轴目标位置。通过使用位置运行动作功能块的持续更新输入而使能重置功能。当执行位置运行动作功能块时，在使动作功能块的持续更新输入为On后改变目标位置，并通过再次激活执行输入执行位置运行体现改变目标。
- (2) 当目标位置的改变体现大于从当前速度降低到停止的位置时目标位置改变，定位沿着当前的移动方向。反之，如果改变位置小于从当前速度降低到停止的位置，减速停止后定位沿着当前的相反移动方向。

[例]使用持续更新进行位置重置



3. 速度，加速/减速，加速度重置

- (1) 该功能用于指定轴速度，加速/减速和加速度的重置。
- (2) 通过使用运行中动作功能块的持续更新输入重置速度，加速/减速，加速度到绝对值。当执行运行动作功能块时，导通动作功能块的持续更新输入后改变速度和加速，运行体现速度的改变并通过再次激活执行输入执行加速。
- (3) 对于以当前指令速度运行的速度重置执行，不可以用一个绝对值重置(MC_SetOverride)动作功能块用于重置。
 - 如果值为1.0, 当前运行速度, 加速/减速, 加速度与之前一样.
 - 对于指定VelFactor值为0.0, 轴变为停止, 但是不可以变更为'StandStil'状态.
 - 如果AccFactor值为0.0, 不应用加速/减速而立即变更速率值.
 - 如果JerkFactor值为0.0, 加速/减速比率立即应用, 因此指令速率线性增加/减少.
 - 重置(MC_SetOverride)运动功能块指定要素值的含义根据一般参数重置项目值的不同而不同.

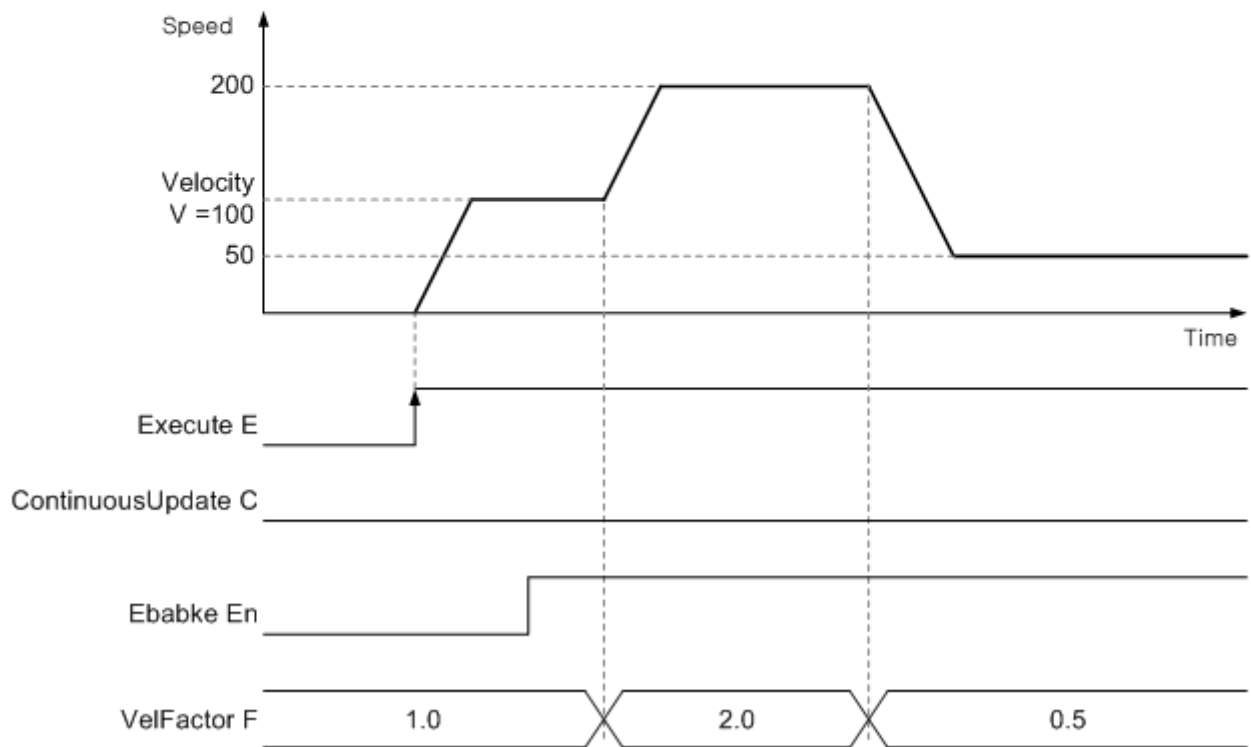
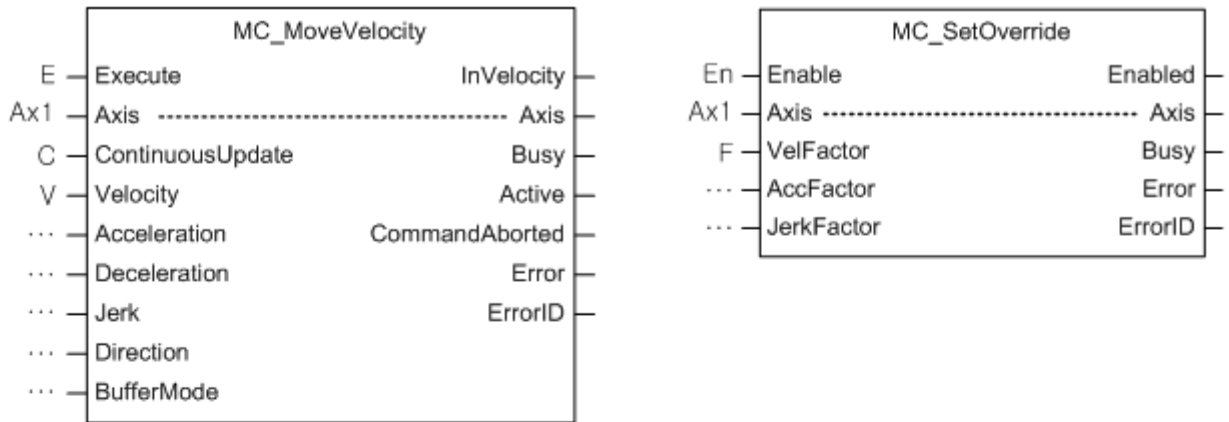
0: 指定百分比 -以当前指令速度运行因素值

1: 指定单位值 -因素值是一个设置项中的绝对单位指定值

(4) 运动功能块

名称	描述	运行条件
MC_SetOverride	速度重置	水平
MC_SetOverride		
BOOL	Enable	Enabled
UINT	Axis	Axis
LREAL	VelFactor	Busy
LREAL	AccFactor	Error
LREAL	JerkFactor	ErrorID
		BOOl
		UINt
		BOOl
		BOOl
		WORD

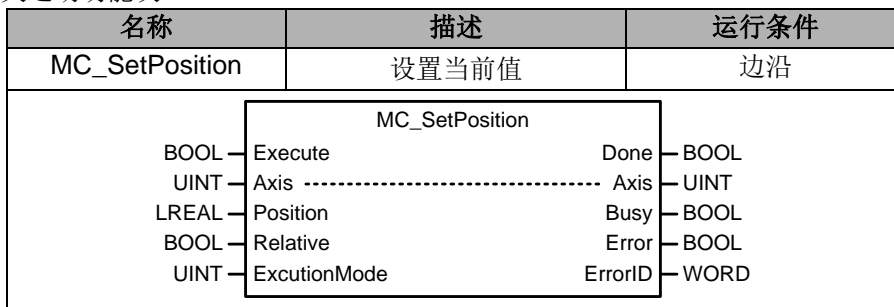
[例] 改变重置(MC_SetOverride)动作功能块的速度



4. 改变当前位置

- (1) 该功能是改变轴的当前位置到用户的指定值。
- (2) 在位置输入中，位置是指定的。当指令执行时相关输入为Off状态，轴位置替代为位置输入值，如果相关输入为On状态，位置输入值添加到轴当前位置。
 - 0: 绝对坐标位置
 - 1: 相对坐标位置
- (3) 设置点可以指定为执行模式输入。当输入值为0时，在执行指令后立即应用设置值，如果为1时，按照顺序操作设置与‘缓冲’相同的值。
 - 0: 立即应用位置值
 - 1: 应用与缓冲模式中的‘缓冲’相同值

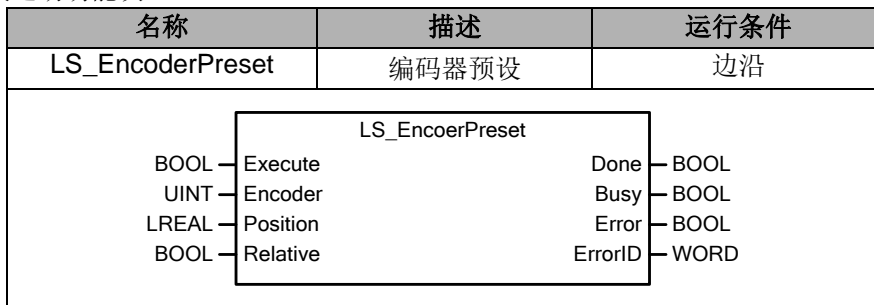
(4) 相关运动功能块



5. 编码器预设

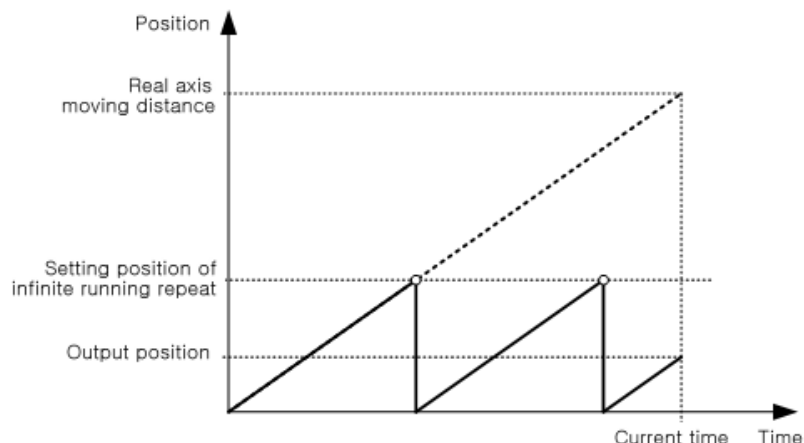
- (1) 该功能为改变当前编码器位置值到用户指定的任何位置值。
- (2) 在编码器输入中，编码器可以指定。
 - 1: 编码器 1
 - 2: 编码器 2
- (3) 在位置输入中，指定编码器位置。当执行指令时相关输入为Off状态，轴编码器位置变更为位置输入值，如果相关输入为On，位置输入值添加到编码器当前位置。
 - 0:绝对坐标位置
 - 1:相对坐标位置编码器

(4) 相关运动功能块



6. 无限重复运行

- (1) 无限重复运行功能是执行指令位置显示值的周期性更新和当前位置自动添加运行参数中扩展参数‘无限重复运行定位’中的设置值。使用无限重复运行定位功能可以决定在相同方向的重复位置值。



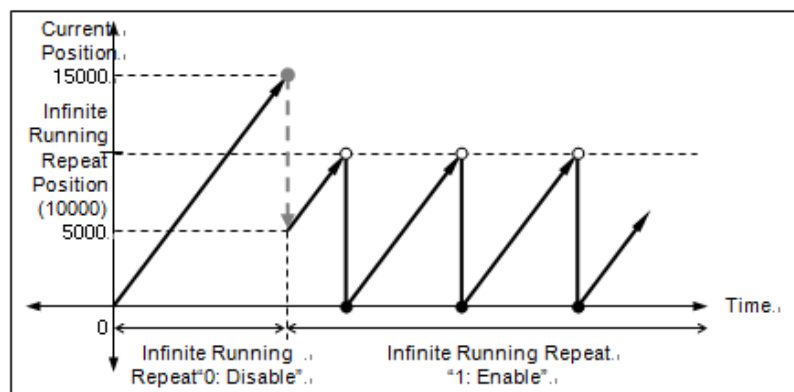
- (2) 运行参数中扩展参数‘无限重复运行’一旦设置为允许，如果值不在无限重复运行定位范围内，当前位置自动变为无限重复运行定位中的值。

[例 1] 如果当前位置为-32100，无限重复运行定位为10000。

当无限重复运行设置为“1:允许”，当前位置变为7900。

[例 2] 如果当前位置为15000，无限重复运行定位为10000。

当无限重复运行设置为“1:允许”，当前位置变为5000。



- (3) 驱动轴无限重复运行可以通过软件或者轴参数变更功能实现。

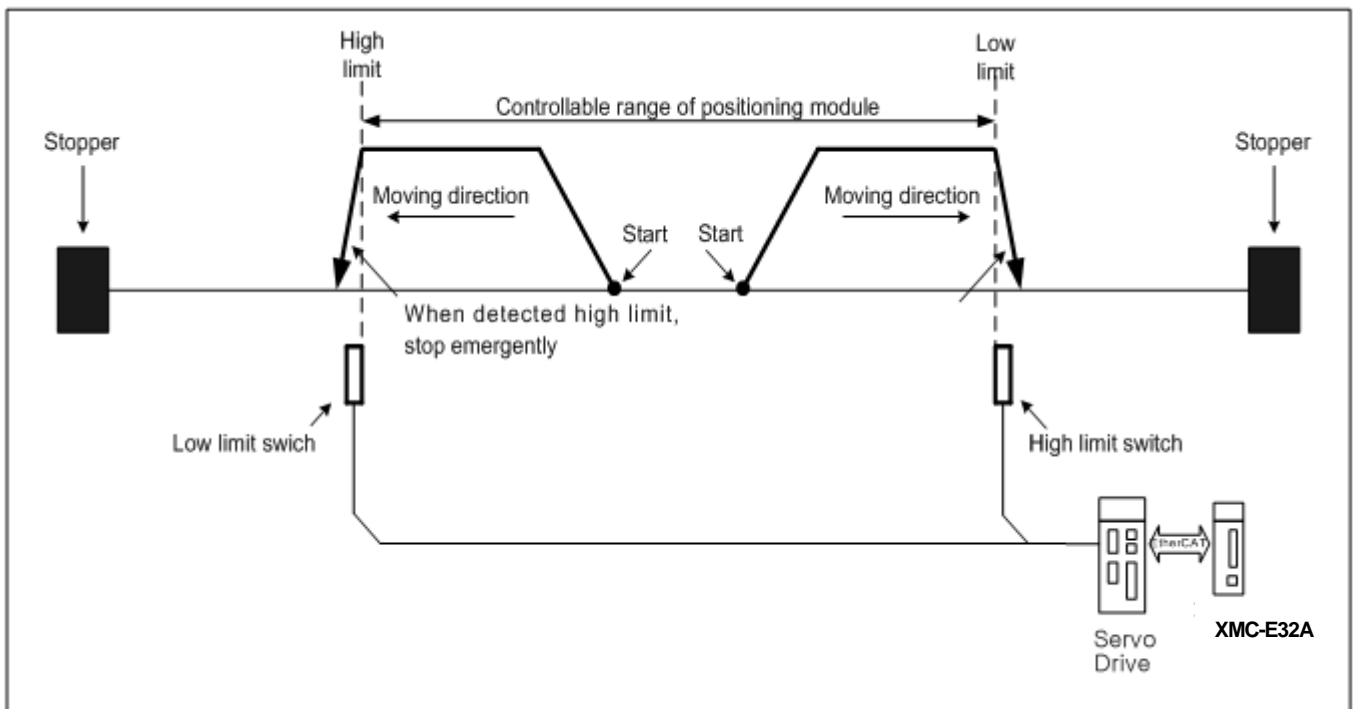
(4) 相关参数设置

项目	描述	设置	初始值
无限重复运行定位	如果用于无限重复运行模式时设置重复运行定位范围	长实数(LREAL)正数	360 pls
无限重复运行	设置是否允许重复运行功能	0: 禁止 1: 使能	0:禁止

8.3.2 控制辅助功能

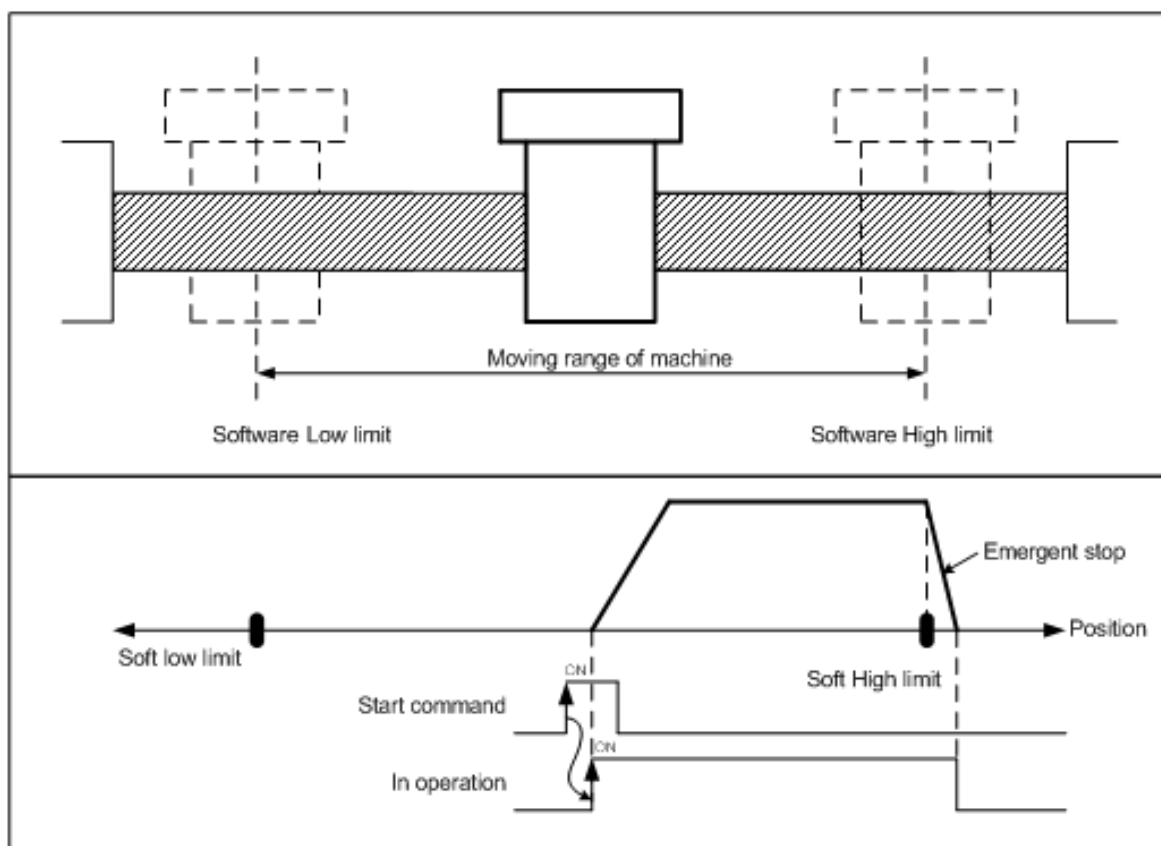
1. 硬件上/下限

- (1) 通过安装上/下限内的上/下限位开关，设置在达到设备的上/下限之前进行急停，该上/下限为设备的物理运行范围。如果范围超出上限，发生错误‘0x1200’，如果低于下限，发生错误‘0x1201’。
- (2) 硬件上/下限开关的输入连接到每个伺服驱动，通过上/下限检测进行伺服驱动运行停止，模块立即停止当前运行动作。
- (3) 如果由于检测到硬件上/下限信号而停止，要求点动运行移动到与检测信号相反方向的运动控制模块可控范围内。
- (4) 硬件上/下限动作如下。



2. 软件上/下限

- (1) 软件上/下限功能是超出用户设置的软件上/下限范围而运行停止。
- (2) 每个驱动轴的软件上/下限可以通过使用软件和轴参数改变功能设置。
- (3) 如果轴超出范围，轴错误发生。
- (4) 当轴定位超出范围，轴运行除了点动以外其他都禁止。通过点动运行移动到范围内或者重置当前位置而进入范围内，运行可重新开始。



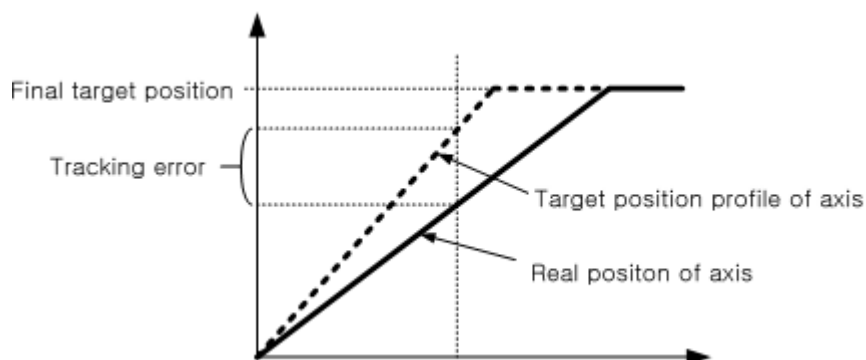
- (5) 在以下情况下不检测软件上/下限
 - 如果软件上限值和下限值设置为初始值(上限: 2147483647, 下限: -2147483648)
 - 软件上限值和下限值相同(软件上限 = 软件下限)
 - 当扩展参数“速度控制期间S/W限制”设置为“0: 不检测”时进行速度控制运行。

(6) 相关参数设置

项目	描述	设置	初始值
S/W上限	设置软件限制功能范围	长实数(LREAL)	2147483647 pls
S/W下限			-2147483648 pls

3. 定位跟踪错误

- (1) 当驱动轴到达位置运行或者轴的实际读取位置进一步超出位置运行指令框架中目标位置的跟踪公差时，该功能输出一个错误
- (2) 每个驱动轴的定位跟踪公差可通过软件或者轴参数变更功能设置



- (3) 是否在跟踪错误发生时设置异常警告或者警报，可以在扩展参数跟踪错误等级中设置。

根据设置值监控如下

- '0: 警告'

当跟踪错误发生时，「以上误差警告(_AXxx_DEV_WARN)」标志位变为On，跟踪错误警报错误(错误代码: 0x101D)发生。轴在不停止的情况下持续运行。

- '1: 警报'

当跟踪错误发生时，「以上误差警报(_AXxx_DEV_ERR)」标志位变为On，跟踪错误警报错误(错误代码: 0x101C)发生。轴由于「急停减速」而发生急停。

- (4) 以下情况不执行检查跟踪错误。

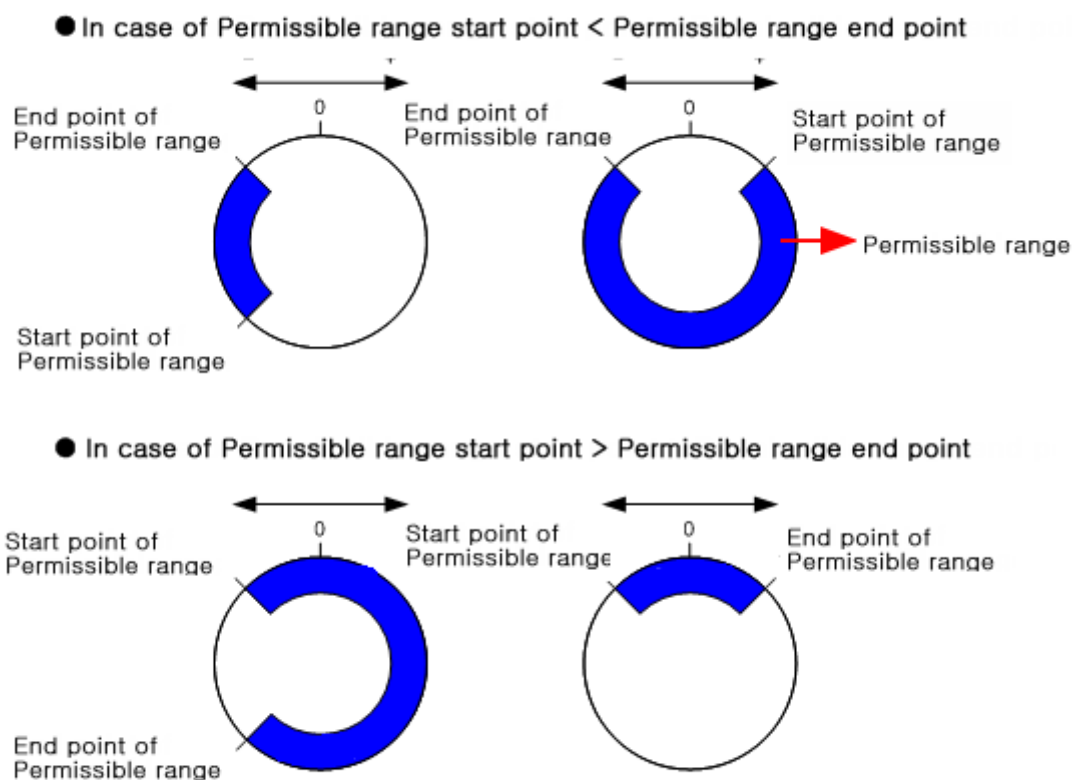
- 如果「跟踪错误超出值」为 0
- 如果是原点返回或者转矩控制运行

- (5) 相关参数设置

项目	描述	设置	初始值
跟踪误差超出值	设置检测位置偏差值	0或者长实数(LREAL)正数	0
跟踪错误等级	设置上述误差错误等级	0: 警告 1: 警报	0: 警告

4. 锁存(接触探头)

- (1) 当轴的特定情况（触发事件）发生时，该功能记录轴的位置。
- (2) 根据触发输入（触发输入）设置选择接触探头1和 2。
 - 触发输入(触发输入)=0：当接触探头1信号为Off->On时执行锁存功能。
 - 触发输入(触发输入)=1：当接触探头2信号为Off->On时执行锁存功能。
- (3) 锁存（接触探头）功能运行区域可以指定。
 - 当指定允许区域时，仅可以在指定区域内运行。
 - 如果是无限重复运行(旋转轴)，根据允许区域的开始和结束，锁存（接触探头）运行区域的关系如下。



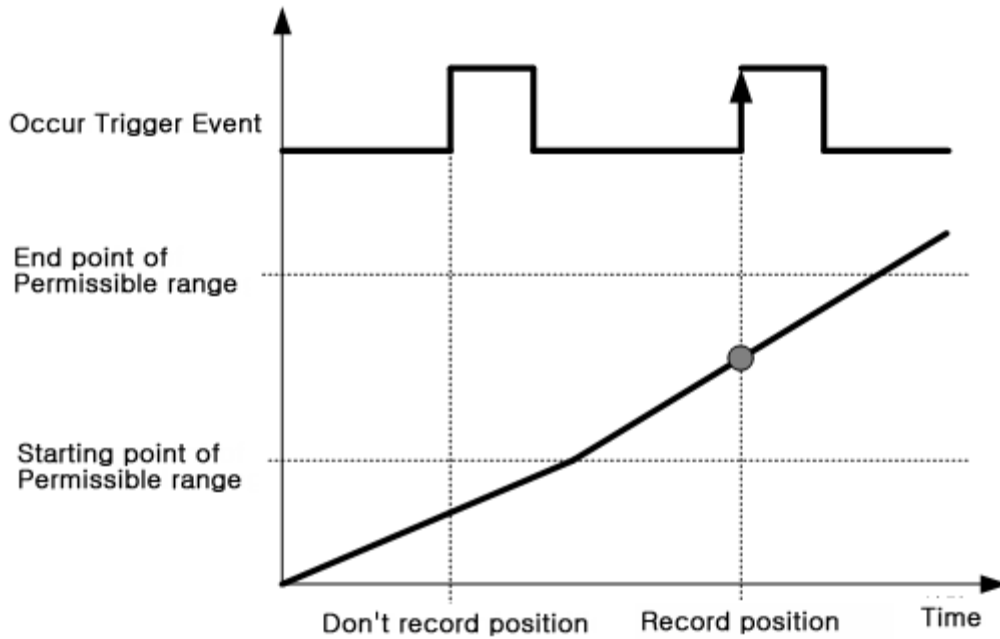
- (4) 关于锁存（接触探头）功能的使用，设置从站参数时应包含以下对象。

触发输入	RxPDO	TxPDO
接触探头1	0x60B8:0接触探头功能	0x60B9:0接触探头状态 0x60BA:接触探头1 0正方向位置值
接触探头2	0x60B8:0接触探头功能	0x60B9:0接触探头状态 0x60BC:接触探头1 0正方向位置值

如果没有以上对象，当使用锁存（接触探头）指令时发生错误(错误代码: 0x10E0).

第8章 运动控制功能

(5) 运行时序



(6) 相关动作功能块

名称	描述	运行条件																		
MC_TouchProbe	接触探头	边沿																		
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2">MC_TouchProbe</th> </tr> </thead> <tbody> <tr> <td>BOOL Execute</td> <td>Done</td> </tr> <tr> <td>UINT Axis</td> <td>Axis</td> </tr> <tr> <td>UINT TriggerInput</td> <td>TriggerInput</td> </tr> <tr> <td>BOOL WindowOnly</td> <td>Busy</td> </tr> <tr> <td>LREAL FirstPosition</td> <td>CommandAborted</td> </tr> <tr> <td>LREAL LastPosition</td> <td>Error</td> </tr> <tr> <td></td> <td>ErrorID</td> </tr> <tr> <td></td> <td>RecordedPosition</td> </tr> </tbody> </table>			MC_TouchProbe		BOOL Execute	Done	UINT Axis	Axis	UINT TriggerInput	TriggerInput	BOOL WindowOnly	Busy	LREAL FirstPosition	CommandAborted	LREAL LastPosition	Error		ErrorID		RecordedPosition
MC_TouchProbe																				
BOOL Execute	Done																			
UINT Axis	Axis																			
UINT TriggerInput	TriggerInput																			
BOOL WindowOnly	Busy																			
LREAL FirstPosition	CommandAborted																			
LREAL LastPosition	Error																			
	ErrorID																			
	RecordedPosition																			

名称	描述	运行条件														
MC_AbortTrigger	中止触发	边沿														
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2">MC_AbortTrigger</th> </tr> </thead> <tbody> <tr> <td>BOOL Execute</td> <td>Done</td> </tr> <tr> <td>UINT Axis</td> <td>Axis</td> </tr> <tr> <td>UINT TriggerInput</td> <td>TriggerInput</td> </tr> <tr> <td></td> <td>Busy</td> </tr> <tr> <td></td> <td>Error</td> </tr> <tr> <td></td> <td>ErrorID</td> </tr> </tbody> </table>			MC_AbortTrigger		BOOL Execute	Done	UINT Axis	Axis	UINT TriggerInput	TriggerInput		Busy		Error		ErrorID
MC_AbortTrigger																
BOOL Execute	Done															
UINT Axis	Axis															
UINT TriggerInput	TriggerInput															
	Busy															
	Error															
	ErrorID															

2. 错误复位监控

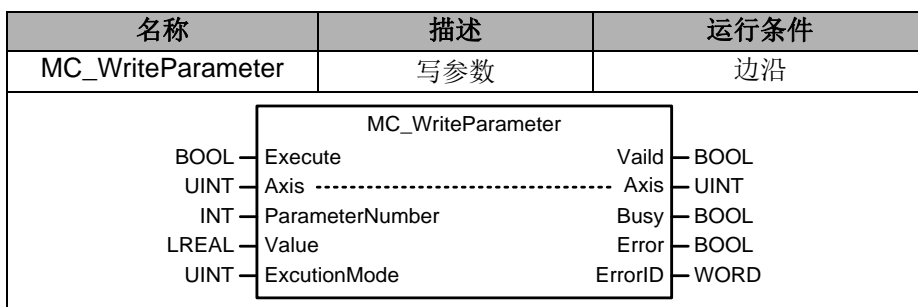
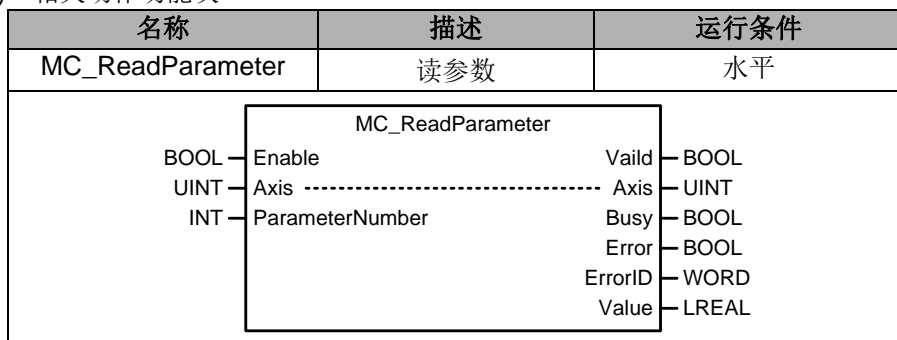
- (1) 如果在通过错误复位指令而在轴中发生错误复位的时候伺服驱动发生错误，通过设置错误复位监控时间而验证是否适当复位伺服驱动错误。
- (2) 如果监控时间超出，即使驱动错误不复位，也不再执行错误复位。
- (3) 每个驱动轴的错误复位监控时间可通过软件或轴参数变更功能进行设置。
- (4) 相关参数设置。

项目	描述	设置	初始值
错误复位监控时间	在伺服驱动中发生错误复位情况下设置监控时间	1 ~ 1000 ms	100 ms

8.3.3数据管理功能

1. 参数管理

- (1) 该功能用于读取和改变存储在模块中的轴参数。
- (2) 通过指定轴数量和匹配参数数量而改变所需参数值。
- (3) 如果没有错误发生，通过参数写入功能修改的参数值自动存储备份。
- (4) 关于在“参数数量”中设置的参数，参考动作功能块项目。
- (5) 相关动作功能块



2. Cam数据管理

可以通过cam数据读取/写入指令读取和改变程序中的cam数据

(1) Cam数据读取

- 当输入使能时， CmDataRead指令通过CamTable ID读取指定cam轮廓数据，并作为MasterPoint和SlavePoint保存数据到指定数据区域.
- 保存从营文件中读取的"Main-axis Position"和"Sub-axis Position" 变量首地址在MasterPoint和SlavePoint中设置. 例如，保存"Main-axis Position"到调用MainAxPos[100]的阵列变量，和"Sub-axis Position"值到SubAsPos[100]阵列变量， MainAxPos[0]应该在功能块MasterPoint中设置，而SubAsPos[0]应该在SlavePoint中设置.
- 在CamCurveSel[4]中设置，用于每个位保存的应用点插值类型.

CamCurveSel[0]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Point 8	Point 7	Point 6	Point 5	Point 4	Point 3	Point 2	Point 1
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	Point 16	Point 15	Point 14	Point 13	Point 12	Point 11	Point 10	Point 9
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
	Point 24	Point 23	Point 22	Point 21	Point 20	Point 19	Point 18	Point 17
	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
CamCurveSel[1]	Point 32	Point 31	Point 30	Point 29	Point 28	Point 27	Point 26	Point 25
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Point 40	Point 39	Point 38	Point 37	Point 36	Point 35	Point 34	Point 33
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	Point 48	Point 47	Point 46	Point 45	Point 44	Point 43	Point 42	Point 41
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
	Point 56	Point 55	Point 54	Point 53	Point 52	Point 51	Point 50	Point 49
CamCurveSel[2]	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
	Point 64	Point 63	Point 62	Point 61	Point 60	Point 59	Point 58	Point 57
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Point 72	Point 71	Point 70	Point 69	Point 68	Point 67	Point 66	Point 65
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	Point 80	Point 79	Point 78	Point 77	Point 76	Point 75	Point 74	Point 73
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
CamCurveSel[3]	Point 88	Point 87	Point 86	Point 85	Point 84	Point 83	Point 82	Point 81
	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
	Point 96	Point 95	Point 94	Point 93	Point 92	Point 91	Point 90	Point 89
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	不使用				Point 100	Point 99	Point 98	Point 97
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	不使用							
Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	
不使用								
Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	
不使用								

(2) Cam数据写入

- 当执行输入为on时，CamDataWrite指令写入通过CamTable ID在指定cam轮廓的StartSlope和EndSlope中设置的值，以及通过CamPointNum 作为"Main-axis Position"和"Sub-axis Position" 在指定数量MasterPoint和SlavePoint 中设置的设备值。
- 在CamCurveSel[4]中，用于每个位保存的应用点插值类型。

CamCurveSel[0]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Point 8	Point 7	Point 6	Point 5	Point 4	Point 3	Point 2	Point 1
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	Point 16	Point 15	Point 14	Point 13	Point 12	Point 11	Point 10	Point 9
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
	Point 24	Point 23	Point 22	Point 21	Point 20	Point 19	Point 18	Point 17
	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
CamCurveSel[1]	Point 32	Point 31	Point 30	Point 29	Point 28	Point 27	Point 26	Point 25
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Point 40	Point 39	Point 38	Point 37	Point 36	Point 35	Point 34	Point 33
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	Point 48	Point 47	Point 46	Point 45	Point 44	Point 43	Point 42	Point 41
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 1...09	Bit 18	Bit 17	Bit 16
	Point 56	Point 55	Point 54	Point 53	Point 52	Point 51	Point 50	Point 49
CamCurveSel[2]	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
	Point 64	Point 63	Point 62	Point 61	Point 60	Point 59	Point 58	Point 57
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Point 72	Point 71	Point 70	Point 69	Point 68	Point 67	Point 66	Point 65
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	Point 80	Point 79	Point 78	Point 77	Point 76	Point 75	Point 74	Point 73
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
CamCurveSel[3]	Point 88	Point 87	Point 86	Point 85	Point 84	Point 83	Point 82	Point 81
	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
	Point 96	Point 95	Point 94	Point 93	Point 92	Point 91	Point 90	Point 89
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	不使用				Point 100	Point 99	Point 98	Point 97
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	不使用							
Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	
不使用								
Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	
不使用								

- CamTableID 输入可设置范围 1~32. 设置值超出范围可能导致错误"16#000B"
- CamPointNum 可设置范围 1~100. 设置值超出范围可能导致错误"16#000B"

(3) 运动功能块

■ Cam数据读取

名称	描述	运行条件																											
LS_ReadCamData	Cam数据读取	水平																											
<div style="border: 1px solid black; padding: 5px; margin: 5px auto; width: fit-content;"> <p style="text-align: center;">LS_ReadCamData</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">BOOL—Enable</td> <td style="width: 40%;"></td> <td style="width: 30%;">Done—BOOL</td> </tr> <tr> <td>UINT—Axis</td> <td>----- Axis</td> <td>UINT</td> </tr> <tr> <td>UINT—CamTable ID</td> <td></td> <td>Busy—BOOL</td> </tr> <tr> <td>LREAL—MasterPoint</td> <td></td> <td>Error—BOOL</td> </tr> <tr> <td>LREAL—SlavePoint</td> <td></td> <td>ErrorID—WORD</td> </tr> <tr> <td></td> <td></td> <td>StartSlope—LREAL</td> </tr> <tr> <td></td> <td></td> <td>EndSlope—LREAL</td> </tr> <tr> <td></td> <td></td> <td>CamPointNum—UINT</td> </tr> <tr> <td></td> <td></td> <td>CamCurveSel—Array [4] of DWORD</td> </tr> </table> </div>			BOOL—Enable		Done—BOOL	UINT—Axis	----- Axis	UINT	UINT—CamTable ID		Busy—BOOL	LREAL—MasterPoint		Error—BOOL	LREAL—SlavePoint		ErrorID—WORD			StartSlope—LREAL			EndSlope—LREAL			CamPointNum—UINT			CamCurveSel—Array [4] of DWORD
BOOL—Enable		Done—BOOL																											
UINT—Axis	----- Axis	UINT																											
UINT—CamTable ID		Busy—BOOL																											
LREAL—MasterPoint		Error—BOOL																											
LREAL—SlavePoint		ErrorID—WORD																											
		StartSlope—LREAL																											
		EndSlope—LREAL																											
		CamPointNum—UINT																											
		CamCurveSel—Array [4] of DWORD																											

■ Cam数据写入

名称	描述	运行条件																														
LS_WriteCamData	Cam数据写入	边沿																														
<div style="border: 1px solid black; padding: 5px; margin: 5px auto; width: fit-content;"> <p style="text-align: center;">LS_WriteCamData</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">BOOL—Execute</td> <td style="width: 40%;"></td> <td style="width: 30%;">Done—BOOL</td> </tr> <tr> <td>UINT—Axis</td> <td>----- Axis</td> <td>UINT</td> </tr> <tr> <td>UINT—CamTable ID</td> <td></td> <td>Busy—BOOL</td> </tr> <tr> <td>LREAL—StartSlope</td> <td></td> <td>Error—BOOL</td> </tr> <tr> <td>LREAL—EndSlope</td> <td></td> <td>ErrorID—WORD</td> </tr> <tr> <td>UINT—CamPointNum</td> <td></td> <td></td> </tr> <tr> <td>Array[4] of DWORD—CamCurveSel</td> <td></td> <td></td> </tr> <tr> <td>LREAL—MasterPoint</td> <td></td> <td></td> </tr> <tr> <td>LREAL—SlavePoint</td> <td></td> <td></td> </tr> <tr> <td>UINT—ExecutionMode</td> <td></td> <td></td> </tr> </table> </div>			BOOL—Execute		Done—BOOL	UINT—Axis	----- Axis	UINT	UINT—CamTable ID		Busy—BOOL	LREAL—StartSlope		Error—BOOL	LREAL—EndSlope		ErrorID—WORD	UINT—CamPointNum			Array[4] of DWORD—CamCurveSel			LREAL—MasterPoint			LREAL—SlavePoint			UINT—ExecutionMode		
BOOL—Execute		Done—BOOL																														
UINT—Axis	----- Axis	UINT																														
UINT—CamTable ID		Busy—BOOL																														
LREAL—StartSlope		Error—BOOL																														
LREAL—EndSlope		ErrorID—WORD																														
UINT—CamPointNum																																
Array[4] of DWORD—CamCurveSel																																
LREAL—MasterPoint																																
LREAL—SlavePoint																																
UINT—ExecutionMode																																

3. SDO参数管理

- (1) 该功能读取或者变更通过网络连接的从站设备SDO参数。
- (2) 特定轴数量的参数值和对应对象数量可以读取或者变更。参数数量由索引和分索引指定。参数尺寸由长度指定。
- (3) 索引输入可以如下设置。如果不设置为设置值, "错误0x1F12"发生。

项目	描述
16#0000 ~ 16#0FFF	数据类型描述
16#1000 ~ 16#1FFF	通讯对象
16#2000 ~ 16#5FFF	供应商指定资料区域
16#6000 ~ 16#9FFF	标准设备资料区域

(4) 在分索引中, 可输入值范围从0到255, 如果设置超出范围, 发生"错误0x1F12. "。

(5) 在长度中, 可输入值

范围从1到4, 表示1到4字节。设置值超出以上范围可能发生"错误0x1F12."

- (6) 通过SDO写入功能变更的参数值无法自动保存到从站设备的ROM中。为了保存变更参数到伺服驱动EEPROM中, 请使用SDO保存指令。
- (7) 运动功能块

■ SDO读取

名称	描述	运行条件														
MC_ReadSDO	SDO读取	水平														
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center;">LS_ReadSDO</th> </tr> </thead> <tbody> <tr> <td>BOOL Execute</td> <td>Done</td> </tr> <tr> <td>UINT Slave</td> <td>Slave</td> </tr> <tr> <td>UINT Index</td> <td>Busy</td> </tr> <tr> <td>UINT SubIndex</td> <td>Error</td> </tr> <tr> <td>UINT Length</td> <td>ErrorID</td> </tr> <tr> <td></td> <td>Value</td> </tr> </tbody> </table>			LS_ReadSDO		BOOL Execute	Done	UINT Slave	Slave	UINT Index	Busy	UINT SubIndex	Error	UINT Length	ErrorID		Value
LS_ReadSDO																
BOOL Execute	Done															
UINT Slave	Slave															
UINT Index	Busy															
UINT SubIndex	Error															
UINT Length	ErrorID															
	Value															

■ SDO写入

名称	描述	运行条件																
MC_WriteSDO	SDO写入	边沿																
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center;">LS_WriteSDO</th> </tr> </thead> <tbody> <tr> <td>BOOL Execute</td> <td>Done</td> </tr> <tr> <td>UINT Slave</td> <td>Slave</td> </tr> <tr> <td>UINT Index</td> <td>Busy</td> </tr> <tr> <td>UINT SubIndex</td> <td>Error</td> </tr> <tr> <td>UINT Length</td> <td>ErrorID</td> </tr> <tr> <td>DINT Value</td> <td></td> </tr> <tr> <td>UINT ExecutionMode</td> <td></td> </tr> </tbody> </table>			LS_WriteSDO		BOOL Execute	Done	UINT Slave	Slave	UINT Index	Busy	UINT SubIndex	Error	UINT Length	ErrorID	DINT Value		UINT ExecutionMode	
LS_WriteSDO																		
BOOL Execute	Done																	
UINT Slave	Slave																	
UINT Index	Busy																	
UINT SubIndex	Error																	
UINT Length	ErrorID																	
DINT Value																		
UINT ExecutionMode																		

■ SDO保存

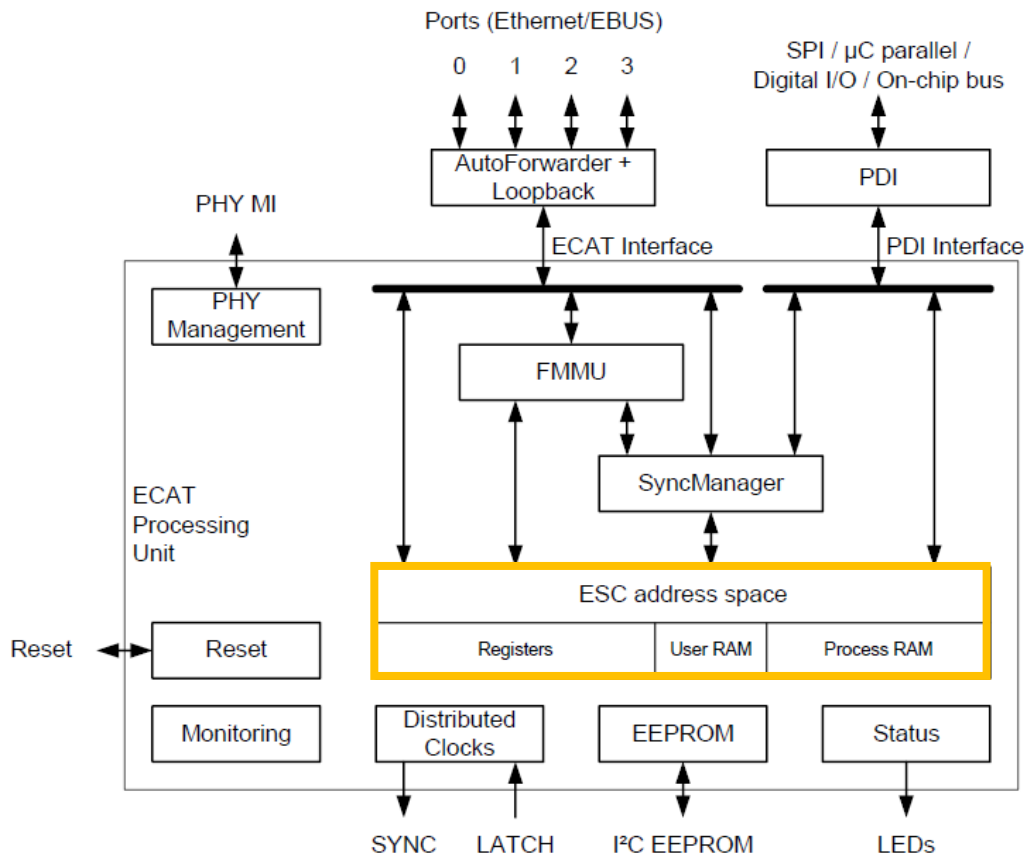
名称	描述	运行条件												
MC_SaveSDO	SDO保存	边沿												
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center;">LS_SaveSDO</th> </tr> </thead> <tbody> <tr> <td>BOOL Execute</td> <td>Done</td> </tr> <tr> <td>UINT Slave</td> <td>Slave</td> </tr> <tr> <td>UINT ExecutionMode</td> <td>Busy</td> </tr> <tr> <td></td> <td>Error</td> </tr> <tr> <td></td> <td>ErrorID</td> </tr> </tbody> </table>			LS_SaveSDO		BOOL Execute	Done	UINT Slave	Slave	UINT ExecutionMode	Busy		Error		ErrorID
LS_SaveSDO														
BOOL Execute	Done													
UINT Slave	Slave													
UINT ExecutionMode	Busy													
	Error													
	ErrorID													

8.3.4 EtherCAT通讯诊断功能

EtherCAT从站设备使用ASIC, FPGA, 或者包含于标准微控制器的EtherCAT从站控制器(ESC), 执行EtherCAT通讯, EtherCAT通讯诊断功能读取和写入ECS (EtherCAT从站控制器)寄存器和从站设备存储, 允许用户查看EtherCAT通讯状态和错误. EtherCAT 通讯诊断功能可以用于是否通讯正常.

1. EtherCAT从站控制器(ESC)

(1) ECS配置如下. 诊断功能指令可用于读取和写入ESC地址空间
显示块图如下.



(2) ESC地址空间配置如下.

地址	定义	备注
0x0000	ESC注册	ESC信息, FMMU, SyncManager, 分布式时钟(DC).
:		
0x0FFF		
0x1000	过程数据RAM	数字I/O输入数据, 过程数据RAM (1KB ~ 60KB)
:		
0xFFFF		

※请参考参考EtherCAT从站控制器(ESC)数据表寡欲注册和进程RAM的详细信息.

2. ESC读取

- (1) 该功能读取通过网络连接的从站设备ESC中的数据。
- (2) Adp(地址位置)是指定EtherCAT从站设备地址。以下值可根据EcatCmd设置值进行设置。如果EcatCmd设置为7(BRD), Adp输入值忽略

EcatCmd	Adp范围
1 (APRD)	0x0000: 连接的第 1 个从站 0xFFFF: 连接的第 2 个从站 0xFFFE: 连接的第 3 个从站 : 0xFFDD: 连接的第 36 个从站
4 (FPRD)	1001 ~ 1032: 1 轴 ~ 32 轴 1033 ~ 1036: 33 轴 IO ~ 36 轴 IO
7 (BRD)	-

- (3) 在长度中, 设置值范围从1到4, 表示1-4字节。
- (4) 在EcatCmd中, 当指定读取ESC (EtherCAT 从站控制器)时使用指令类型。如下可使用如下3个指令:

1) 1- APRD(自动递增物理读取)

当主站正常通讯之前按照物理连接顺序读取从站设备数据时, 使用该指令。接收0值Adp的从站设备将通过长度读取指定大小数据。当接收EtherCAT帧时, 每个从站设备的Adp增加。例如, 如果EcatCmd为1, 而Adp设置为0xFFFF, 当执行ESC读取功能块时,不执行读取, 因为从第一个从站设备接收EtherCAT帧同时的Adp不是1, 仅Adp增加1。当第2个从站设备接收EtherCAT帧时, 由于第1个从站值的Adp值由1变为0, 因此执行读取。根据从站设备连接顺序的Adp值如下

从站控制器	设置值
第1个连接的从站	0
第2个连接的从站	0xFFFF
:	:
第36个连接的从站	0xFFDD

2) 4 - FPRD (配置地址物理读取)

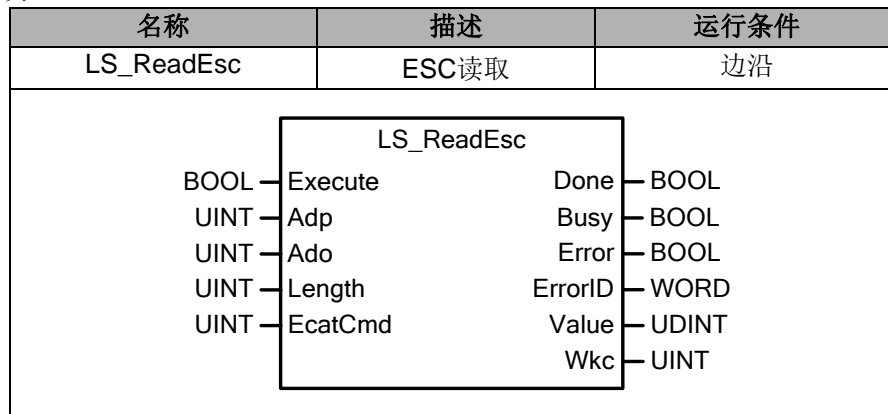
用于在主轴正常通讯连接后通过指定从站设备站号地址的数据读取。如果通过EtherCAT 主站设置的从站设备站号地址与传送的Adp相匹配, 从站设备通过Ado区域中的长度读取指定大小的数据。由主站设备设置的从站设备站号地址如下。

从站控制器	设置值
1轴	1001
2轴	1002
:	:
32轴	1032
33轴IO	1033
:	:
36轴IO	1036

3) 7 – BRD (广播读取)

所有连接的从站设备读取的尺寸数据，这些数据由Ado区域中的长度进行设置，并在Bitwise-OR 后保存结果。在Adp的指定地址值被忽略，Wkc增加1导致正常执行的所有从站读取运行

- (5) 当执行运行功能块时，值和Wkc显示为0。当执行完成时(Done输出为on)，读取数据值在Value中显示，工作计数器在Wkc中显示。
- (6) Wkc代表工作计数器。如果指定从站设备中成功写入，则增加1。如果EcatCmd为7(BRD)，则增加1，导致正常执行的所有从站读取运行。
- (7) ESC读取指令执行完成后，如果从指定从站设备执行正常数据读取运行，Done输出为on。
- (8) ESC读取指令和ESC写入指令不可以同时执行。如果同时执行，执行在程序中最后执行的那个指令，并发生错误(0x1021)
- (9) 功能块



(10) 在以下情况中，由于可能导致错误而ESC读取无法执行。

- 1) 没有从站设备连接到模块 (错误代码: 0x0F09)
- 2) Adp设置值超出范围 (错误代码: 0x0F60)
- 3) 长度设置值超出范围 (错误代码: 0x0F61)
- 4) EcatCmd设置值超出范围 (错误代码: 0x0F62)
- 5) ESC读取指令没有响应 (错误代码: 0x0F63)

3. ESC写入

(1) 该功能实现通过网络连接从站设备的ESC数据写入。

Adp输入指定EtherCAT从站设备地址，以下数据可以在EcatCmd设置中进行设置。如果EcatCmd设置为8(BWR)，Adp输入值忽略

EcatCmd	Adp范围
2 (APWR)	0x0000: 第 1 个连接的从站 0xFFFF: 第 2 个连接的从站 0xFFFE: 第 3 个连接的从站 : 0xFFDD: 第 36 个连接的从站
5 (FPWR)	1001 ~ 1032: 1 轴 ~ 32 轴 1033 ~ 1036: 33 轴 IO ~ 36 轴 IO
8 (BWR)	-

(2) 在EcatCmd中，当指定读取ESC (EtherCAT Slave Controller)时，使用指令类型。

可使用如下3个指令：

1) 2- APWR(自动递增物理写入)

当主站正常通讯之前按照物理连接顺序读取从站设备数据时，使用该指令。接收0值Adp的从站设备将通过长度读取指定大小数据。当接收EtherCAT帧时，每个从站设备的Adp增加。例如，如果EcatCmd为2，而Adp设置为0xFFFF，当执行ESC读取功能块时，不执行读取，因为从第一个从站设备接收EtherCAT帧同时的Adp不是0，仅Adp增加11。当第2个从站设备接收EtherCAT帧时，由于第1个从站值的Adp值由1变为0，因此执行写入。根据从站设备连接顺序的Adp值如下。

从站控制器	设置值
第1个连接的从站	0
第2个连接的从站	0xFFFF
:	:
第36个连接的从站	0xFFDD

2) 5 – FPWR(配置地址物理写入)

用于在主轴正常通讯连接后通过指定从站设备站号地址的数据写入。如果通过EtherCAT 主站设置的从站设备站号地址与传送的Adp相匹配，从站设备通过Ado区域中的长度写入指定大小的数据。由主站设备设置的从站设备站号地址如下。

从站控制器	设置值
1轴	1001
2轴	1002
:	:
32轴	1032
33轴IO	1033
:	:
36轴IO	1036

3) 8 –BWR(广播写入)

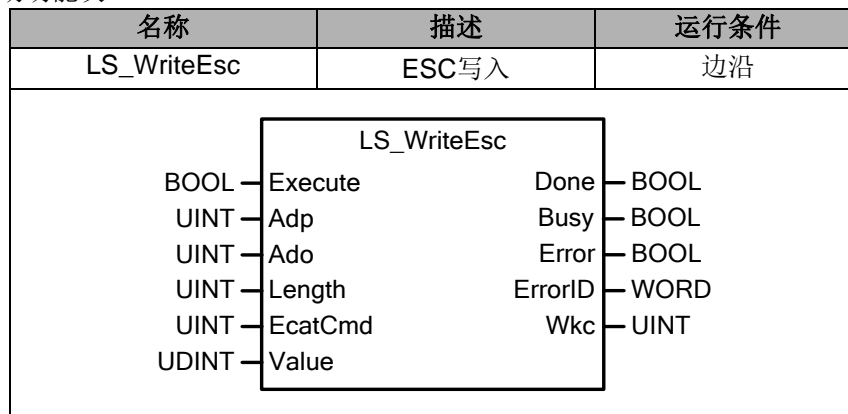
所有连接的从站设备写入的尺寸数据，这些数据由Ado区域中的长度进行设置。在Adp的指定地址值被忽略，Wkc增加1导致正常执行的所有从站写入运行。

- (3) 当执行运行功能块时，值和Wkc显示为0，当执行完成时(Done输出为on)，显示工作计数器值。通过在EcatCmd和Adp 中每个指定从站设备，Wkc加1。
- (4) Wkc代表工作计数器。如果指定从站设备中成功写入，则增加1。如果EcatCmd为8(BWR)，通过每个执行正常写入运行的从站设备，Wkc加1。
- (5) ESC读取指令执行完成后，如果从指定从站设备执行正常数据写入运行，Done输出为on。
- (6) 从站设备使用ESC执行EtherCAT通讯。因此，当执行连接/断开指令或者正常EtherCAT通讯时变更ESC注册值可以防止位置当前动作的从站设备，或者导致通讯错误。因此，在不执行写入动作情况下使用以下ESC注册导致错误发生。(错误代码: 0x0F74)

Ado范围	定义
0x0010 ~ 0x0011	配置站号地址
0x0020 ~ 0x0021	写保护
0x0030 ~ 0x0031	
0x0040	ESC复位ECAT
0x0100 ~ 0x0103	DL控制
0x0120 ~ 0x0121	AL控制
0x0600 ~ 0x06FF	FMMU
0x0800 ~ 0x087F	SyncManager
0x0900 ~ 0x09FF	分布式时钟

※0x0120 (AL控制)注册可以在正常通讯连接后写入，不执行连接/断开指令。

- (7) ESC读取指令和ESC写入指令不可以同时执行.如果同时执行，执行在程序中最后执行的那个指令，并发生错误(0x1021)。
- (8) 运动功能块



- (9) 在以下情况中，由于可能导致错误而ESC读取无法执行
 - 1) 没有从站设备连接到模块(错误代码: 0x0F09)
 - 2) Adp 设置值超出范围 (错误代码: 0x0F70)
 - 3) 长度设置值超出范围(错误代码: 0x0F71)
 - 4) EcatCmd设置值超出范围 (错误代码: 0x0F72)
 - 5) ESC读取指令没有响应(错误代码: 0x0F73)
 - 6) Ado设置值错误 (错误代码: 0x0F74)

8.3.5 电缆复制功能

提供电缆复制功能用于端口倍增。构建用于端口倍增的环形拓扑将防止从站间网络连接断开，即使在电缆在一次断开的情况下。当断开的电缆再次连接时，将恢复原有的通讯方式。

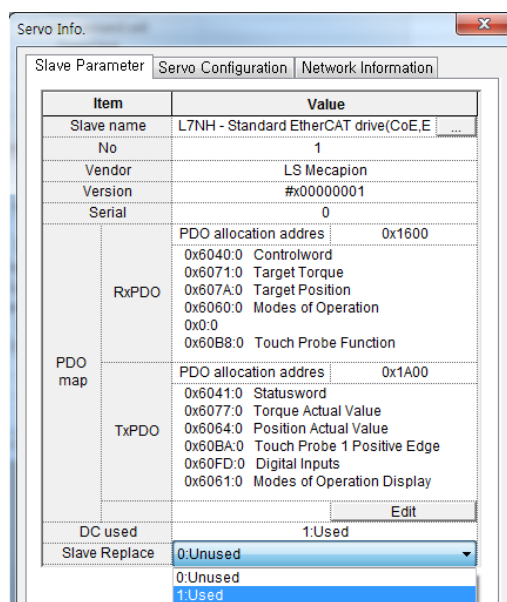
※最多可使用1个端口倍增。对于使用端口倍增的情况，其占有一个IO从站编号。因此，需要注意当使用端口倍增而减少可用IO从站的情况发生。

8.3.6 连接期间的替换功能

当使用电缆复制功能时，如果之前的一个从站设备不是在运行状态，而导致网络断开或者故障修复并连接到网络，该功能检测连接，并在不再次连接全部网络的情况下连接到单个从站设备的网络。

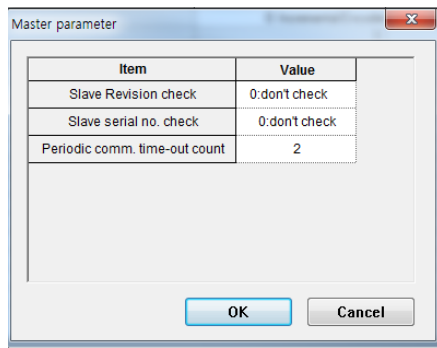
1. 连接设置期间的替换功能

为了在连接期间设置从站设备使用替换从站的功能，在从站信息中的"在连接中替换功能"设置为使能。对于不设置为替换功能使用的从站，如果该从站从网络中移除，将发生网络错误，可能导致整个网络停止。



2. 主站参数设置

当使用在连接中替换功能时，替换的从站必须是与被替换从站相同的从站。为了确定替换从站是否相同，查看供应商/产品代码是否匹配。此外，根据主轴参数设置中的内容查看版本/序列号是否匹配。



3. 连接期间的从站交换方法

- (1) 网络连接期间移除将要替换的从站输入/输出电缆。
- (2) 切断从站电源。
- (3) 打开将要替换的从站电源。
- (4) 连接端口一侧电缆。
- (5) 连接端口一侧电缆。
(输入/输出电缆的同步连接可以保证替换正常.)
- (6) 当从站通讯恢复时，连接另一侧电缆。

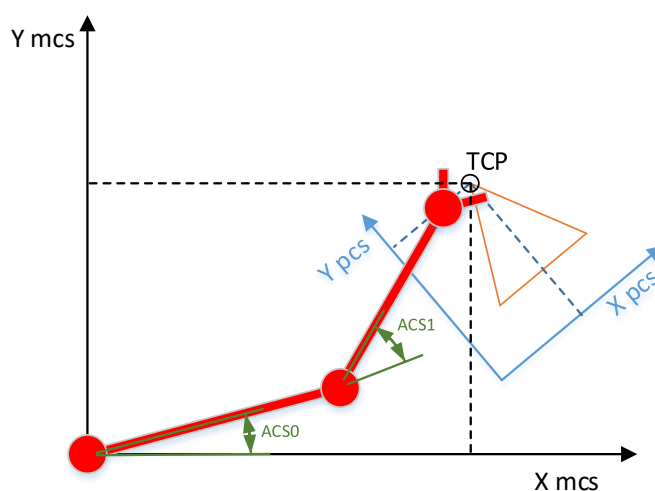
8.4 坐标系统运行功能

8.4.1 坐标系统运行概要

不同坐标系统定义多种指定中心位置或者空间中方向的方式。下图显示通过每个坐标系统如何表示一个特定TCP。在ACS坐标系统中，TCP表示为由两个连接组成的机器人关节的旋转角。在MCS坐标系统中，TCP位置表示基于MCS的原点位置。

在PCS坐标系统中，TCP位置表示基于TCP的原点位置

作为PCS/MCS表示的TCP不可以转换到用于运行的机器人连接电机。为了运行连接到机器人的电机，应使用转换为ACS的值，作为表示电机的实际运动。因此，对于在坐标系统中的运行，通过笛卡尔坐标系统转换将PCS转换为MCS坐标，通过逆运动转换将MCS坐标转换为ACS坐标，并发送CA值到每个电机以启动运行。



8.4.2 ACS/MCS/PCS/TCP

ACS: 轴坐标系统(ACS)表示物理电机的实际运动。

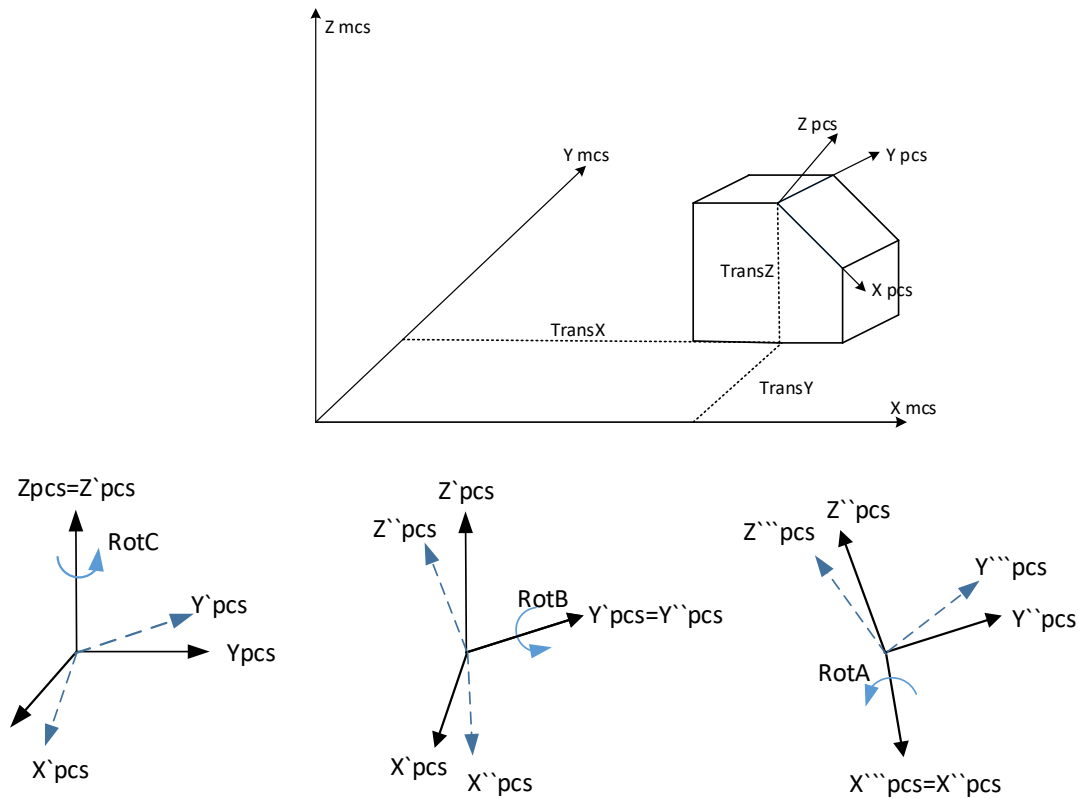
MCS: 机械坐标系统(MCS)与机械(机器人)有关。这是作为笛卡尔坐标系统的机械系统的固定原点位置。

PCS: 产品坐标系统基于MCS表示移动或者旋转产品的位置。PCS通过程序连接产品，用户可以变更。

TCP(工具中心点)是作为指令移动机械(机器人)的位置的工具中心或者结束点。对于使用MCS或者PCS运行的情况，目标位置通过TCP表示。TCP由6个RotC数据组成: Px,Py,Pz,表示沿着XYZ轴的移动; RotA表示沿着X轴旋转; RotB示沿着Y轴旋转; RotC示沿着Z轴旋转。

8.4.3 PCS设置

PCS表示工作站的TCP。TCP通过从原始点的旋转和移动进行定义。转换PCS为MCS的参数在MC_SetCartesianTransform功能块或者设置轴组参数中设置。在MC_SetCartesianTransform中，TransX/TransY/TransZ表示从MCS起始点到PCS起始点的移动距离。对于PCS的RotA/RotB/RotC旋转值。RotA表示沿着X轴的PCS旋转。RotB表示沿着Y轴的PCS旋转。RotC表示沿着Z轴的PCS旋转PCS，旋转按照以下顺序执行: RotC->RotB->RotA



1. 功能块

名称	描述	运行条件
MC_SetCartesianTransform	PCS设置	边沿

MC_SetCartesianTransform		
BOOL	Execute	Done
UINT	AxesGroup	AxesGroup
LREAL	TransX	Busy
LREAL	TransY	Active
LREAL	TransZ	CommandAborted
LREAL	RotAngleA	Error
LREAL	RotAngleB	ErrorID
LREAL	RotAngleC	

8.4.4 机械信息设置

为了运行用于坐标系统运行的机器人，需要提前在轴组参数中设置机器人(机械)类型和机械参数。机械参数可以使用MC_SetKinTransform功能块进行设置。MP500轴组参数可以用相同方式设置。

1. 机械信息，机械类型设置

在机械类型设置中，选择用于执行坐标系统运行的机械(机器人)类型。机器人可以选择XYZ或者Delta3。

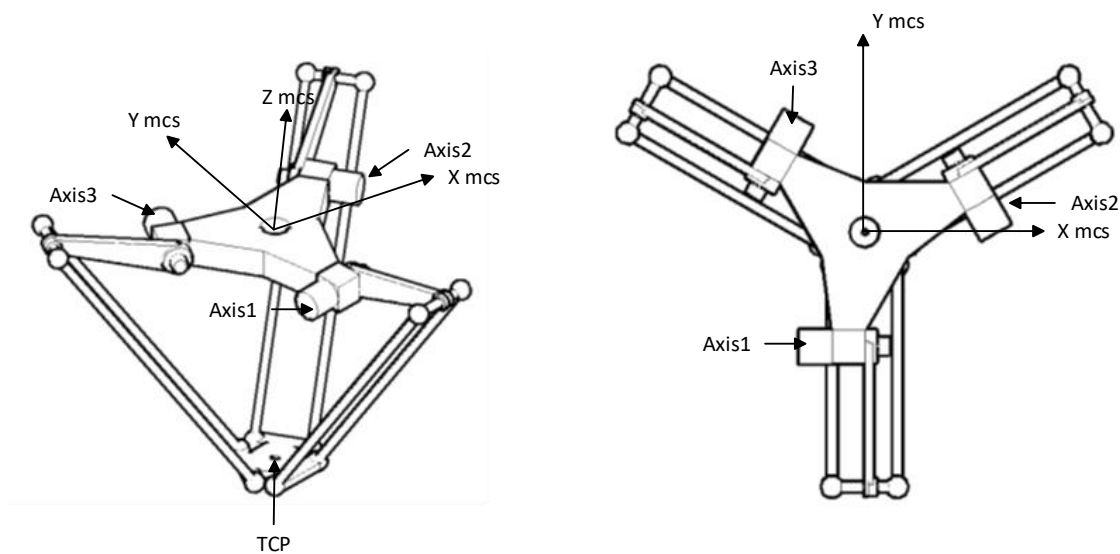
(1) XYZ(笛卡尔坐标)机器人

XYZ是通过连接到X/Y/Z轴的伺服电机的机器人类型，可以以笛卡尔坐标运行，不需要ACS和MCS间的额外运行分析。

(2) 三自由度

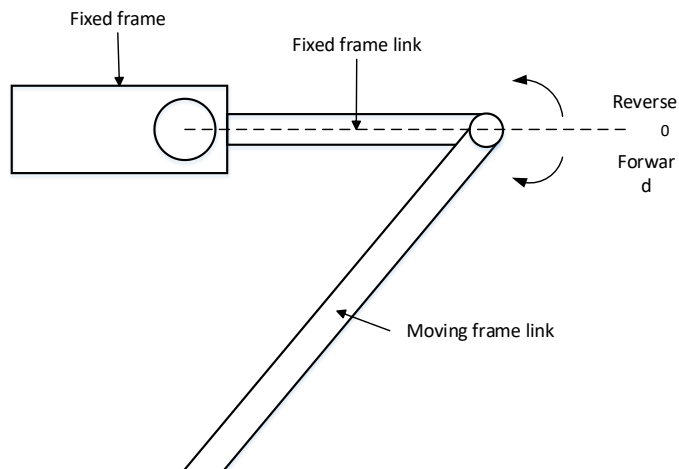
三自由度机器人的MCS

对于三自由度机器人,固定帧的中心定义为MCS，每个连接到三自由度的轴和MCS之间的关系如下。



三自由度机器人的ACS

连接到三自由度的固定帧临近表面的方向被定义为轴运行连接的正方向，其他方向被定义为相对方向。

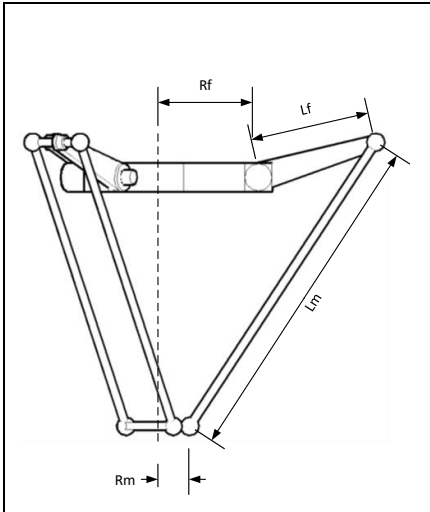


2. 机械信息, 机械参数设置

(1) XYZ

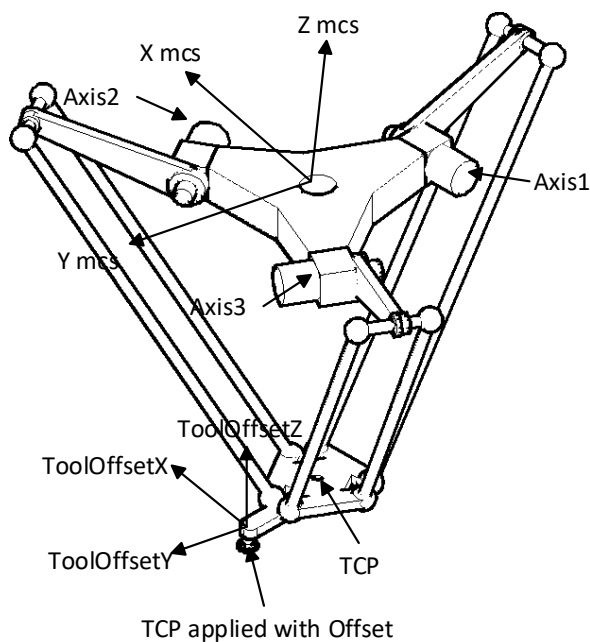
XYZ机器人不要求单独机械参数, 作为匹配TCP的XYZ坐标的每个轴位置.

(2) 三自由度

	参数	描述
	KinParam[0]	Lf:固定帧的连接长度(mm)
	KinParam[1]	Lm:移动帧的连接长度(mm)
	KinParam[2]	Rf: 从固定帧中心到固定帧连接的长度(mm)
	KinParam[3]	Rm:从移动帧中心到移动帧连接的长度(mm)

3. 机械信息, 工具补偿设置

工具补偿功能提供额外机械信息, 作为辅助设备连接到机器人TCP的结尾. 激活应用与补偿的工具补偿到应用于坐标系统运行的TCP目标位置.

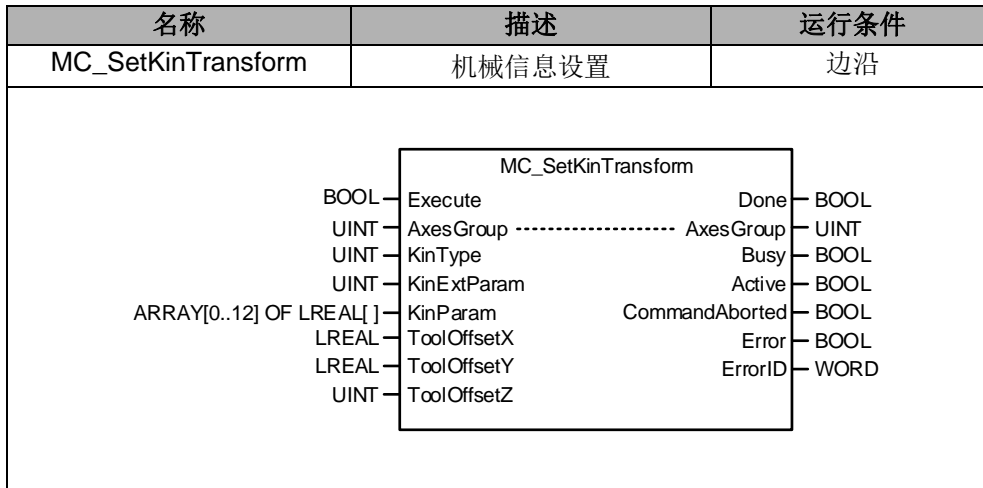


4. 轴组, 轴配置设置

为了执行坐标系统运行, 轴设置应符合机械类型. 如果轴数量或者轴类型不匹配, 则不执行坐标系统运行.

机械类型	轴数量	EA			
		1轴	2轴	3轴	4轴
XYZ	3轴	mm	mm	mm	-
三自由度	3轴	度	度	度	-

5. 功能块



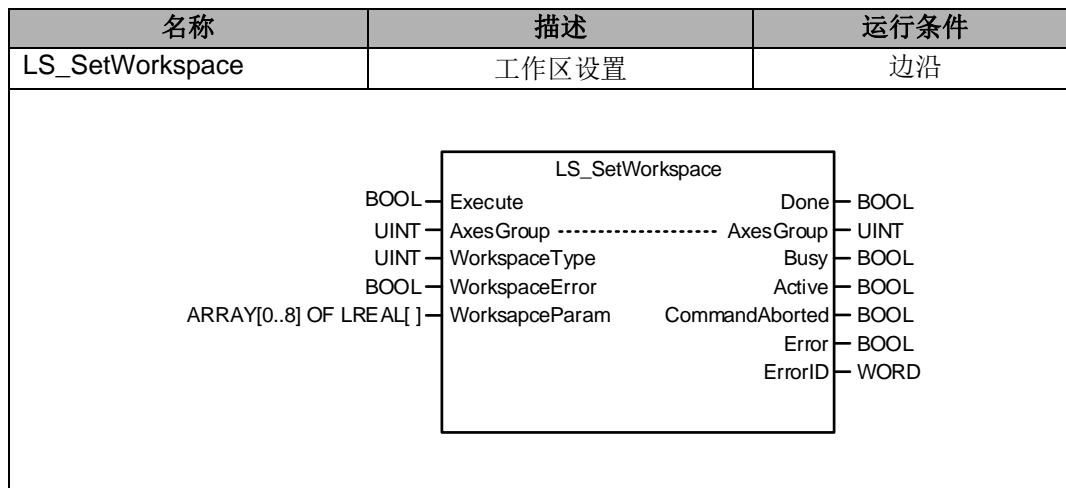
8.4.5 工作区设置

关于坐标系统运行，为防止机械损坏或者机器人未执行动作导致的安全事故，工作区功能的提供防止机器人超出当前工作区。如果机器人当前位置或者目标位置超出工作区，则不执行坐标系统运行。可以使用LS_SetWorkspace 功能块设置工作区。MP500轴组参数可以使用相同方式设置

1. 工作区设置

执行工作区设置，和工作区错误发生可以进行设置。设置WorkspaceType中的工作区类型，作为AxesGroup 输入中指定轴组的WorkspaceParam中的工作区参数。如果WorksaceError值设置为0，即使超出工作区也持续运行，并且不出现错误。

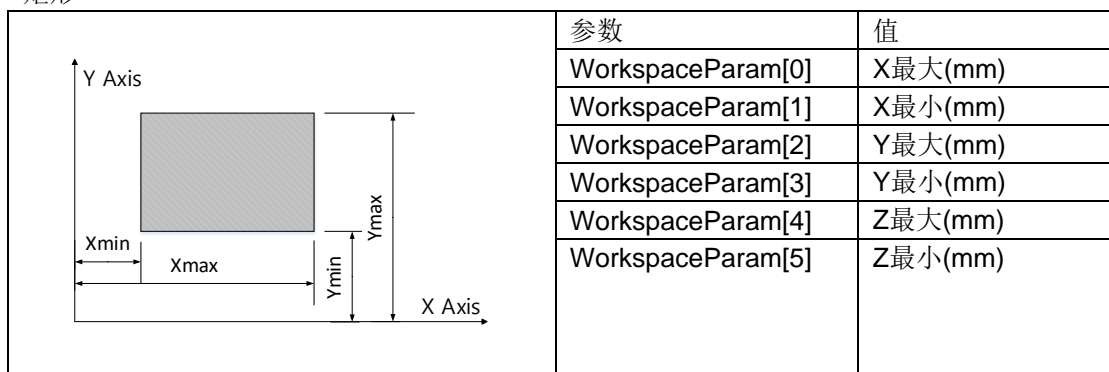
当轴组为运行状态则无法执行工作区设置。



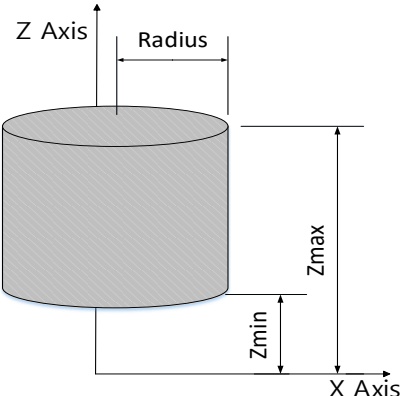
2. 工作区类型和参数

工作区支持4种类型：矩形/圆柱/三自由度/扇形。

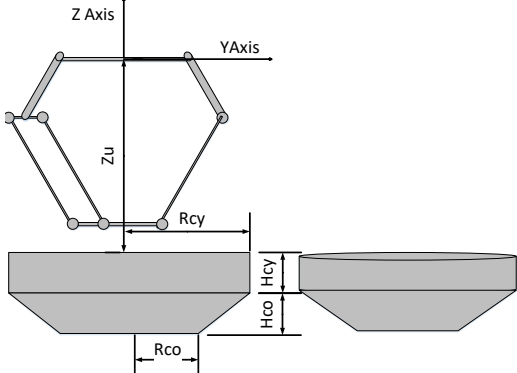
矩形



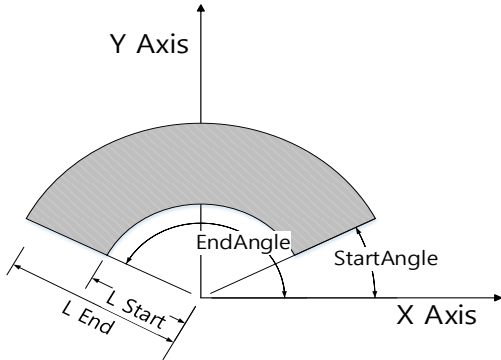
圆柱

	参数	值
	WorkspaceParam[0]	半径(mm)
	WorkspaceParam[1]	Z最大(mm)
	WorkspaceParam[2]	Z最小(mm)

三自由度

	参数	值
	WorkspaceParam[0]	Zu(mm)
	WorkspaceParam[1]	Hcy(mm)
	WorkspaceParam[2]	Hco(mm)
	WorkspaceParam[3]	Rcy(mm)
	WorkspaceParam[4]	Rco(mm)
WorkspaceParam[5]	-	

扇形

	参数	值
	WorkspaceParam[0]	L结束(mm)
	WorkspaceParam[1]	L启动(mm)
	WorkspaceParam[2]	Z最大(mm)
	WorkspaceParam[3]	Z最小(mm)
	WorkspaceParam[4]	结束角(度)
WorkspaceParam[5]	启动角(度)	

8.4.6坐标系统绝对位置的时间线性插值运行

通过在线性轨迹内的设置时间内移动TCP从当前位置到目标位置，使用设置与轴组的相关轴执行插值控制。

1. 从起始位置到目标位置执行线性插值(通过定位数据的指定位置)。定位控制基于返回到原点的指定位置。
2. 设置位置[]以定义TCP目标位置。

变量	定义	单位
Position[0]	X轴位置	mm
Position[1]	Y轴位置	mm
Position[2]	Z轴位置	mm
Position[3]	X轴旋转数量	度
Position[4]	Y轴旋转数量	度
Position[5]	Z轴旋转数量	度

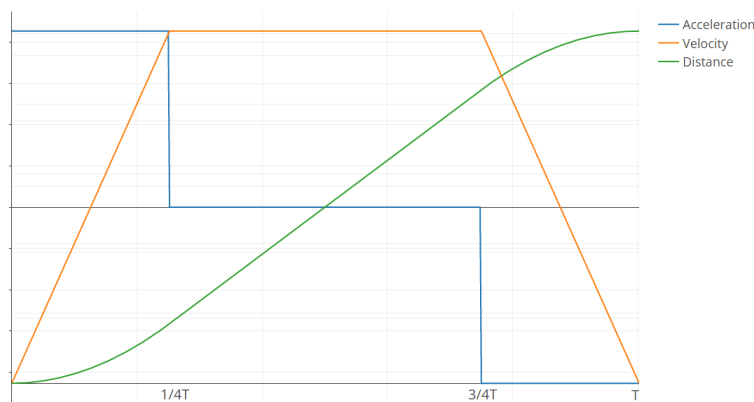
3. 基于机器人类型，某些位置变量区域可能无法应用。无法应用区域的数据输入在坐标系统运行中不反映。

变量	定义	
	XYZ	三自由度
Position[0]	应用	应用
Position[1]	应用	应用
Position[2]	应用	应用
Position[3]	不应用	不应用
Position[4]	不应用	不应用
Position[5]	不应用	不应用

4. 在TrajTime 中设置的时间内执行线性插值到目标TCP。
5. TrajType输入决定用于到达插值轨迹的加速/减速类型。0的三种类型：梯形/正弦1/正弦2。

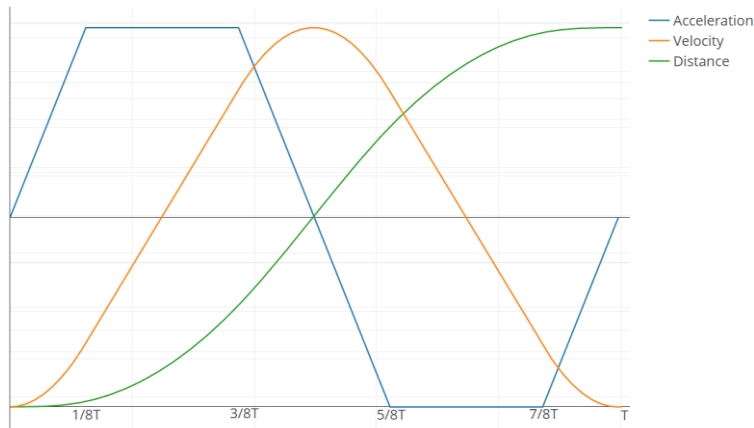
梯形

基本梯形线性加速/减速的运行变化图



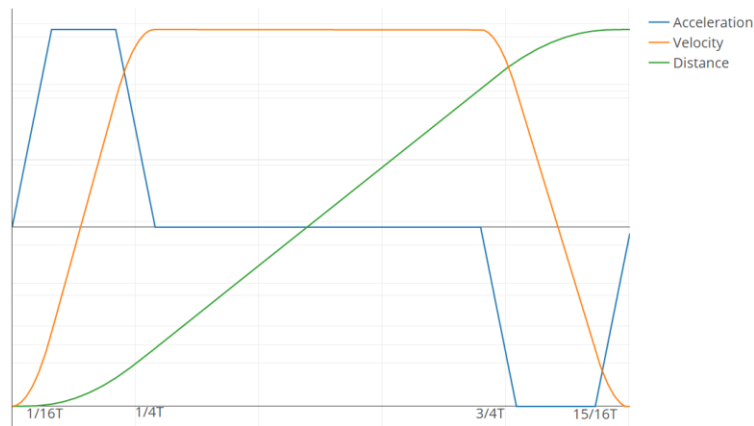
正弦1

该运行类型的速率变化由正弦曲线组成。适用于低负载高速运行，并降低负载变化对电机的影响。



正弦2(常数正弦)

该运行类型的速率变化由正弦曲线和恒速段组成. 适用于高负载, 中速运行.



6. 当CoordSystem输入设置为1时, 使用作为MCS坐标的位置值执行机器人运行. 当设置为2, 使用作为PCS坐标的位置值执行机器人运行.
7. 为了停止当前插值控制, 使用MC_GroupHalt或者MC_GroupStop运动功能块.
8. 功能块

名称	描述	运行条件																											
MC_MoveLinearTimeAbsolute	坐标系统绝对位置时间线性插值运行	边沿																											
<div style="border: 1px solid black; padding: 10px; width: fit-content; margin: 0 auto;"> <p style="text-align: center; margin: 0;">LS_MoveLinearTimeAbsolute</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">BOOL</td> <td style="width: 40%;">Execute</td> <td style="width: 30%;"></td> </tr> <tr> <td>UINT</td> <td>AxesGroup</td> <td>Done</td> </tr> <tr> <td>UINT</td> <td>CoordSystem</td> <td>AxisGroup</td> </tr> <tr> <td>ARRAY[0..6] OF LREAL[]</td> <td>Position</td> <td>Busy</td> </tr> <tr> <td>UINT</td> <td>TrajType</td> <td>Active</td> </tr> <tr> <td>LREAL</td> <td>TrajTime</td> <td>Error</td> </tr> <tr> <td>UINT</td> <td>BufferMode</td> <td>ErrorID</td> </tr> <tr> <td>UINT</td> <td>TransitionMode</td> <td></td> </tr> <tr> <td>LREAL</td> <td>TransitionParameter</td> <td></td> </tr> </table> </div>			BOOL	Execute		UINT	AxesGroup	Done	UINT	CoordSystem	AxisGroup	ARRAY[0..6] OF LREAL[]	Position	Busy	UINT	TrajType	Active	LREAL	TrajTime	Error	UINT	BufferMode	ErrorID	UINT	TransitionMode		LREAL	TransitionParameter	
BOOL	Execute																												
UINT	AxesGroup	Done																											
UINT	CoordSystem	AxisGroup																											
ARRAY[0..6] OF LREAL[]	Position	Busy																											
UINT	TrajType	Active																											
LREAL	TrajTime	Error																											
UINT	BufferMode	ErrorID																											
UINT	TransitionMode																												
LREAL	TransitionParameter																												

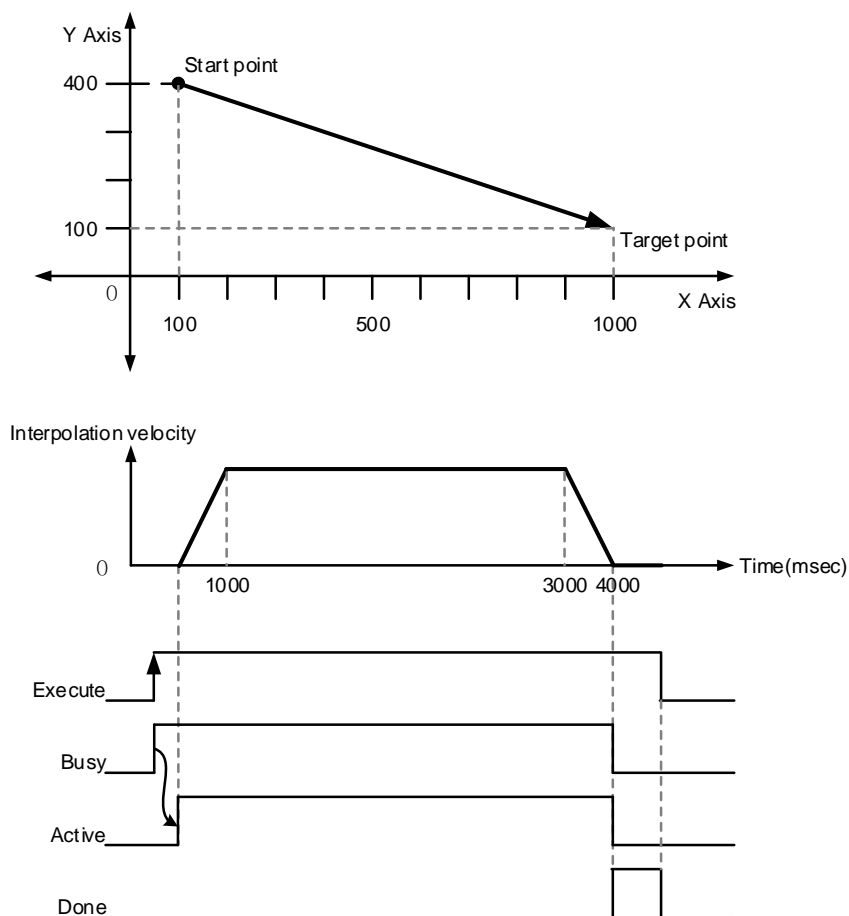
9. 限制

在以下错误情况下坐标系统绝对位置时间线性插值控制无法执行

- CoordSystem输入可以设置为除1或2以外的值 (错误代码: 0x20BC)
- 组成轴的运行参数单位与坐标系统类型不匹配 (错误代码: 0x2063)
- 关于组成轴, 其中一个轴原点位置当前不确定 (错误代码: 0x20B0)
- 组成轴的运行速率超出每个轴的速率限制 (错误代码: 0x20B9)
- 关于组成轴, 有一个轴执行无限重复运行 (错误代码: 0x20BA)

10. 运行时序

- 起始点: (100.0, 400.0, 0.0)
- 目标点: (1000.0, 100.0, 0.0)
- 目标时间: 4000msec
- 运行类型 : 0



8.4.7 坐标系统的圆弧插值运行

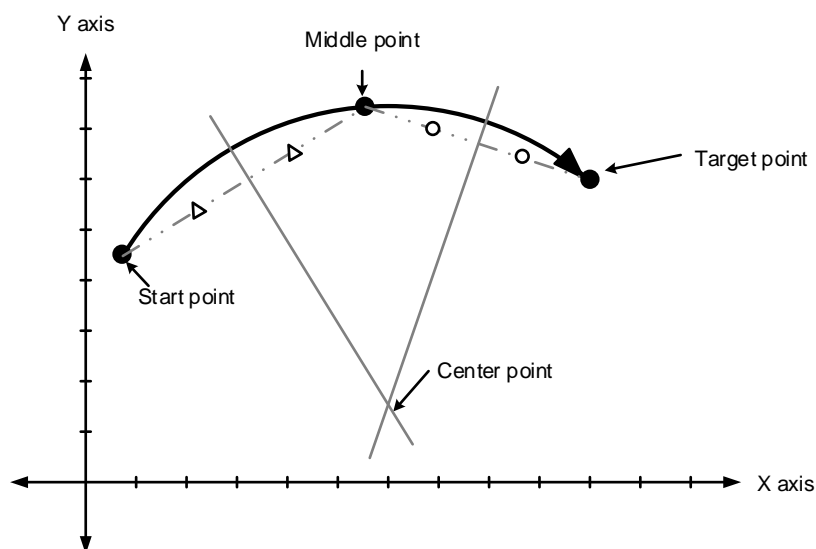
执行基于圆弧插值运行的坐标系统，TCP使用轴组中指定轴在XY平面以圆弧轨迹运行。坐标系统绝对位置圆弧插值控制包含相同设置和除了基于同一坐标系统的动作。有3种类型的圆弧插值：中心点方式，TCP通过辅助点并按照CircMode设置和辅助点传递指定位置；通过辅助点指定位置的中心点方式为中心点，设置为辅助点值的直径方式为圆弧直径。使用MC_GroupHalt或者MC_GroupStop功能块停止当前插值控制。

1. 结束点/辅助点

对于坐标系统圆弧插值控制，输入TCP的Px,Py,Pz到结束点/辅助点。RotA, RotB, RotC值，确定TCP内容，则不输入，在起始位置取代维持的值。

2. 使用中点规格的圆弧插值

- (1) 通过在起始位置启动执行圆弧插值，通过中心点设置为辅助点，并移动到目标位置。
- (2) 创建的圆弧轨迹中心点是起始点和中点之间的垂直平分线和中点和目标位置交叉点。
- (3) 移动方向由设置的目标位置和圆弧插值辅助点自动确定。



(4) 限制

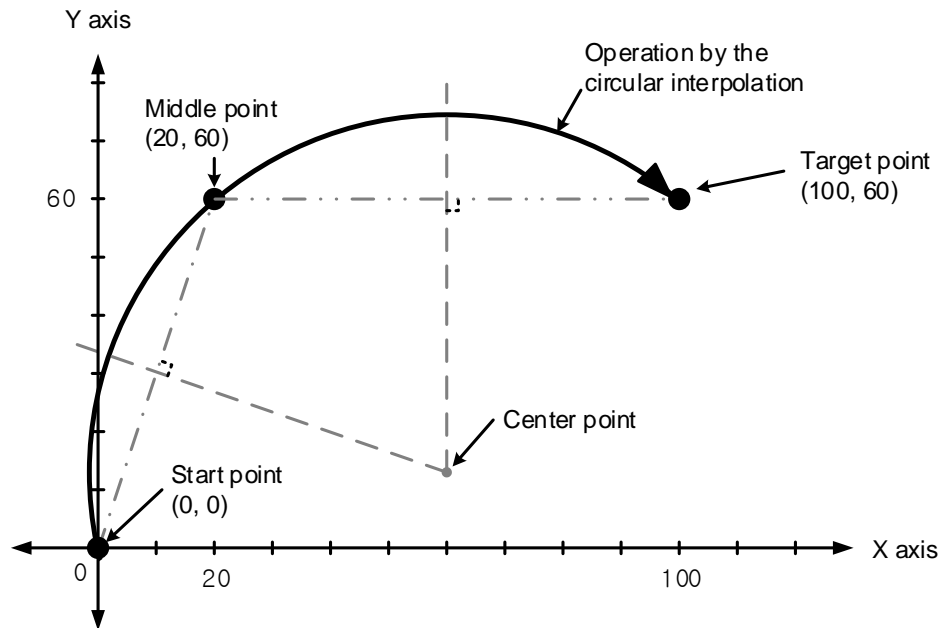
使用中点规格方式的圆弧插值控制无法在以下错误情况下执行。

- 在绝对坐标圆弧插值期间，原地位置不是由1个或者多个组成轴决定 (错误代码: 0x20A0)
- 作为辅助点的中点规格与起始或者目标位置相同 (错误代码: 0x20A4)
- 起始点和目标点相同 (错误代码: 0x20A5)
- 圆弧计算半径超出2147483647pls (错误代码: 0x20A6)
- 起始点, 辅助点, 目标点在同一直线上 (错误代码: 0x20A7)
- 1个或者多个组成轴执行无限重复运行 (错误代码: 0x20A8)

运行模式

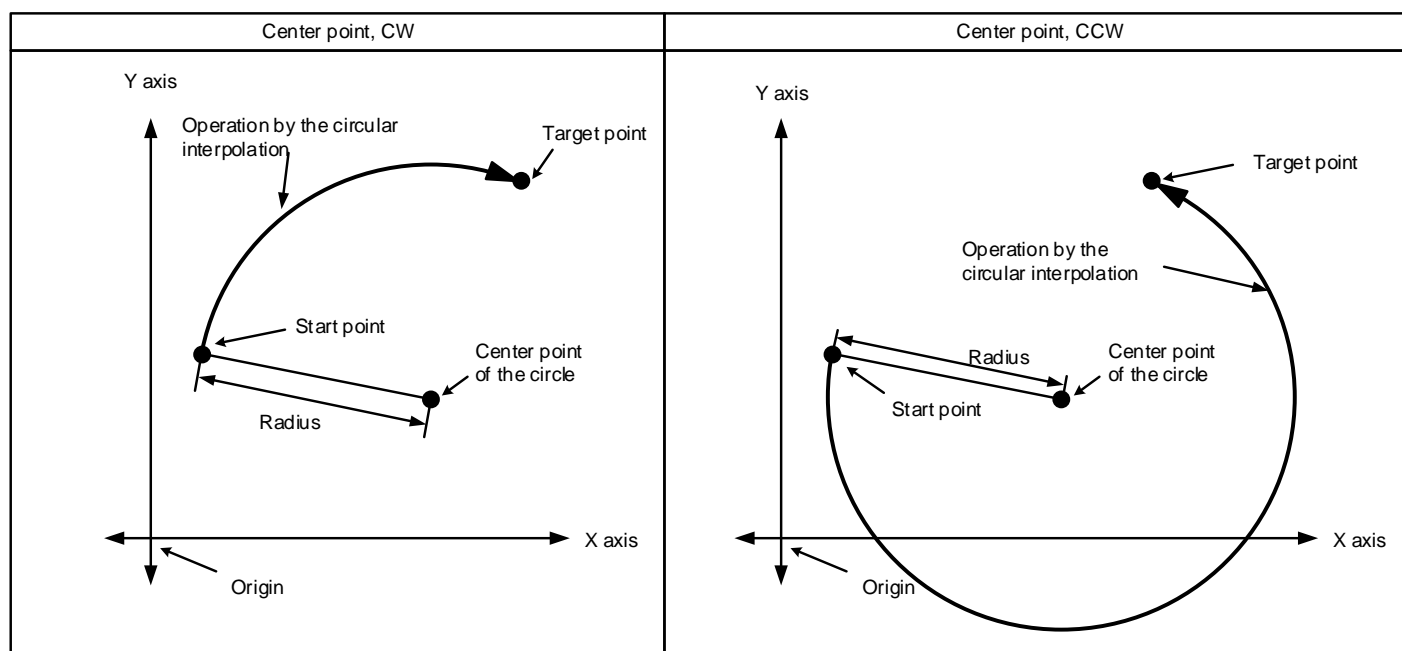
第8章 运动控制功能

- 启动点: (0.0, 0.0,0.0)
- 目标点: (100.0, 60.0,0.0)
- 中点: (20.0, 60.0)
- CircMode: 中点(0)
- PathChoice: - (忽略圆弧插值中使用中点)

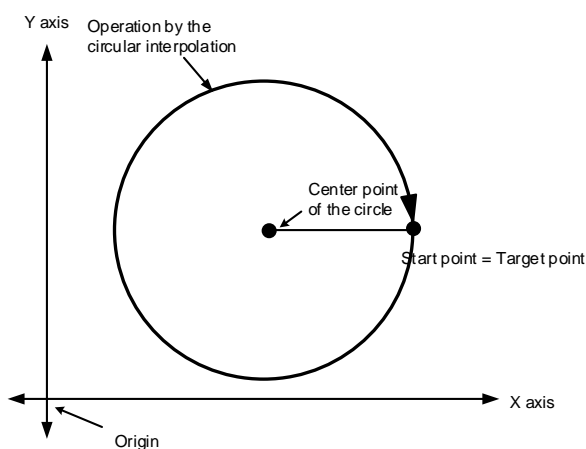


3. 使用中心点规格的圆弧插值

- (1) 通过起始位置启动执行圆弧插值，到达圆弧轨迹中的目标位置，该圆弧轨迹由到指定中心点距离为直径。
- (2) 移动方向由在绝对位置圆弧插值运行 (MC_MoveCircularAbsolute2D)，相对位置圆弧插值 (MC_MoveCircularRelative2D)，或者运动功能块的“PathChoice”中的设置所决定。
- 0: 「CW」 - 执行从起始位置顺时针的圆弧插值.
 - 1: 「CCW」 - 执行从起始位置逆时针的圆弧插值.



- (3) 设置目标位置与起始位置创建用于圆弧插值的适当圆弧轨迹相同，该轨迹直径由起始点和圆中心点间的距离所决定。



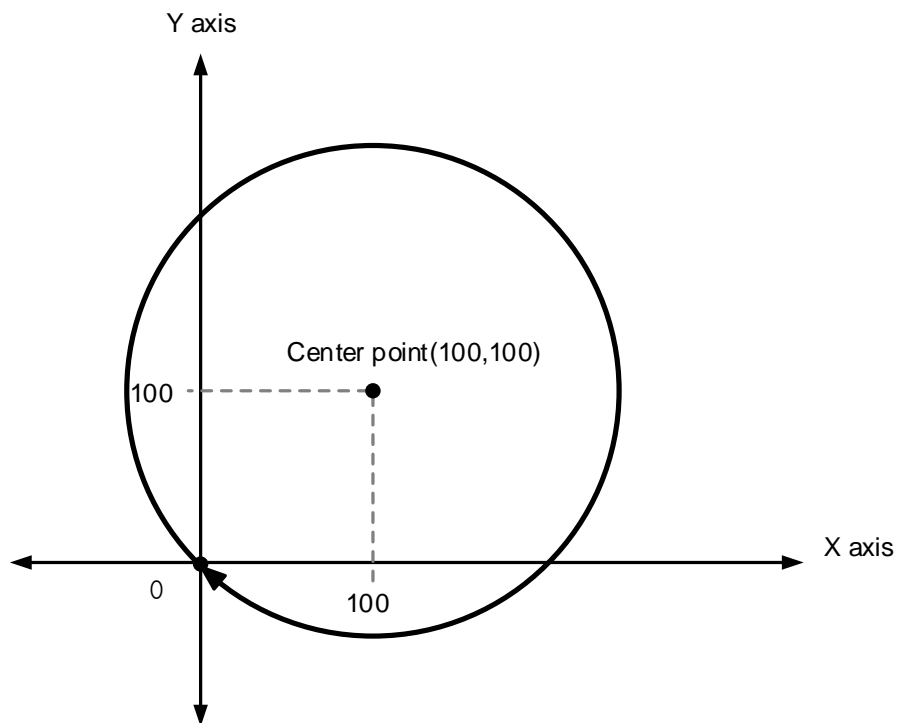
(4) 限制

使用中心点规格方式的圆弧插值无法在以下错误情况下执行。

- 在绝对坐标圆弧插值期间，原地位置不是由1个或者多个组成轴决定 (错误代码: 0x20A0)
- 作为辅助点的中点规格与起始或者目标位置相同 (错误代码: 0x20A4)
- 圆弧计算半径超出2147483647pls (错误代码: 0x20A6)
- 起始点, 辅助点, 目标点在同一直线上 (错误代码: 0x20A7)
- 1个或者多个组成轴执行无限重复运行 (错误代码: 0x20A8)

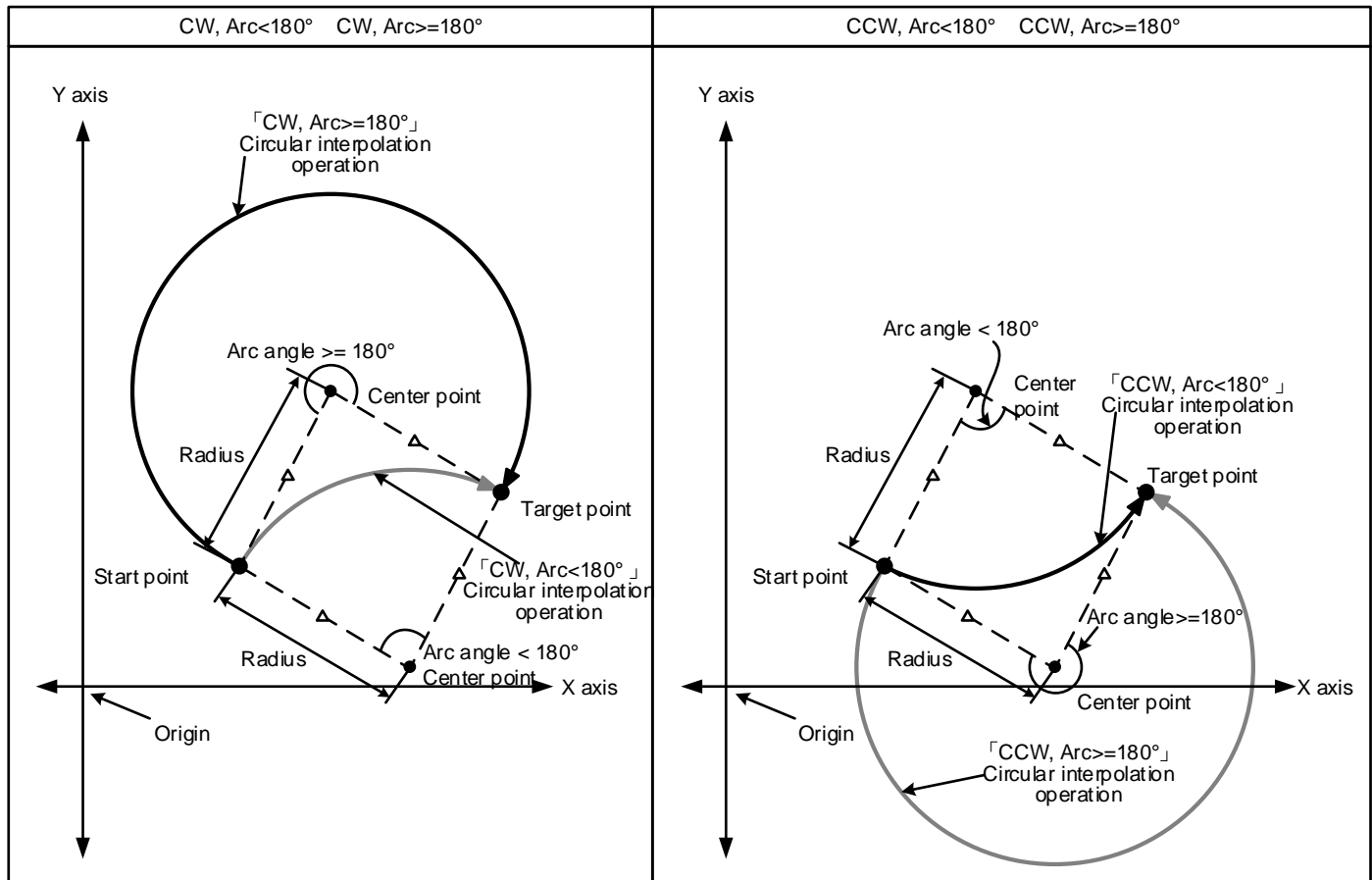
(5) 运行模式

- 起始点: (0.0, 0.0,0.0)
- 目标点: (0.0, 0.0,0.0)
- 辅助点: (100.0, 100.0,0.0)
- CircMode: 中心点(1)
- PathChoice: - CW(0)



4. 使用半径规格的圆弧插值

(1) 通过在起始点启动执行圆弧插值，到达圆弧轨迹中的目标位置，该圆弧轨迹由圆弧插值的辅助点设置的距离为直径. 中心点圆弧根据半径的正数/负数而不同((+): 弧度 $<180^\circ$, (-): 弧度 $\geq 180^\circ$).



(2) 对于指定直径的情况，目标位置不可以和起始位置相同。

移动方向和圆弧尺寸由辅助点符号和绝对位置坐标系圆弧插补运行(MC_MoveCircularAbsolute2D)，相对位置坐标系圆弧插值运行(MC_MoveCircularRelative2D)，或者运动功能块“PathChoice”中设置的方向所决定。

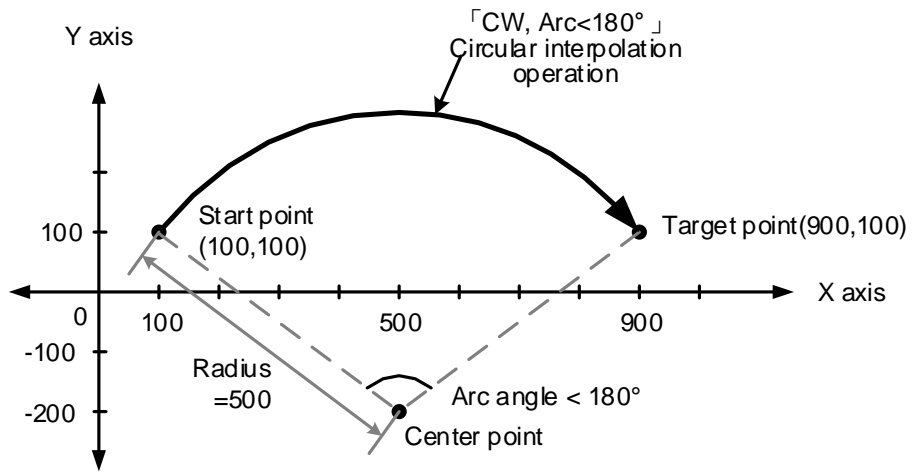
(3) 限制

使用半径指定方式的圆弧插值无法在以下错误情况下执行。

- 在绝对坐标圆弧插值期间，原地位置不是由1个或者多个组成轴决定 (错误代码: 0x20A0)
- 起始位置和目标位置相同 (错误代码: 0x20A5)
- 圆弧计算半径超出2147483647pls (错误代码: 0x20A6)
- 起始点，辅助点，目标点在同一直线上 (错误代码: 0x20A7)
- 1个或者多个组成轴执行无限重复运行 (错误代码: 0x20A8)

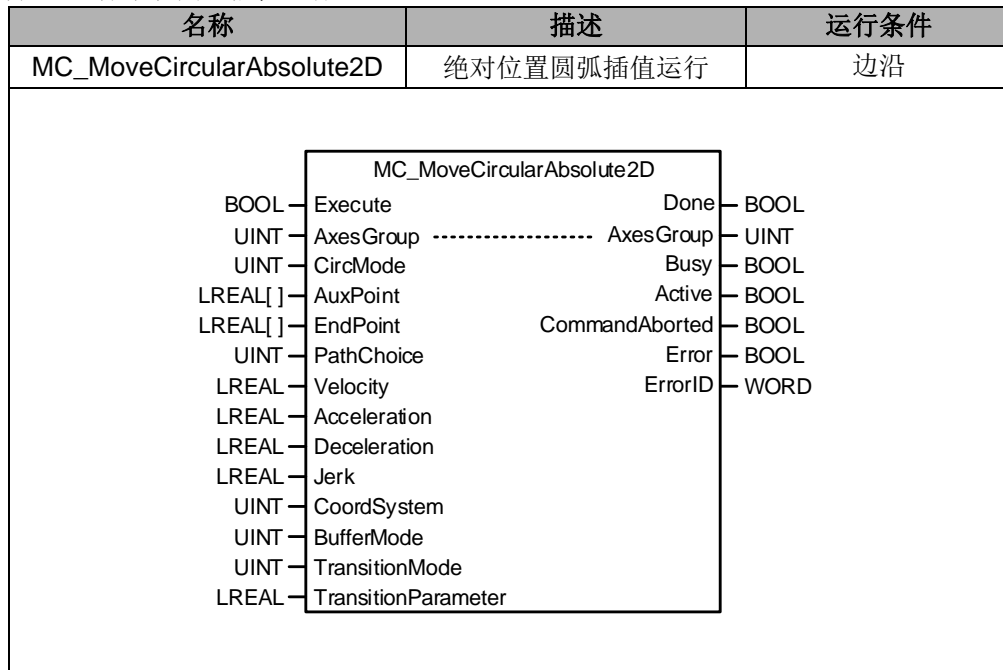
(4) 运行模式

- 起始点: (100.0, 100.0, 0.0)
- 目标点: (900.0, 100.0)
- 辅助点: (500.0, 0.0)
- CircMode: 半径(2)
- PathChoice: - CW(0)

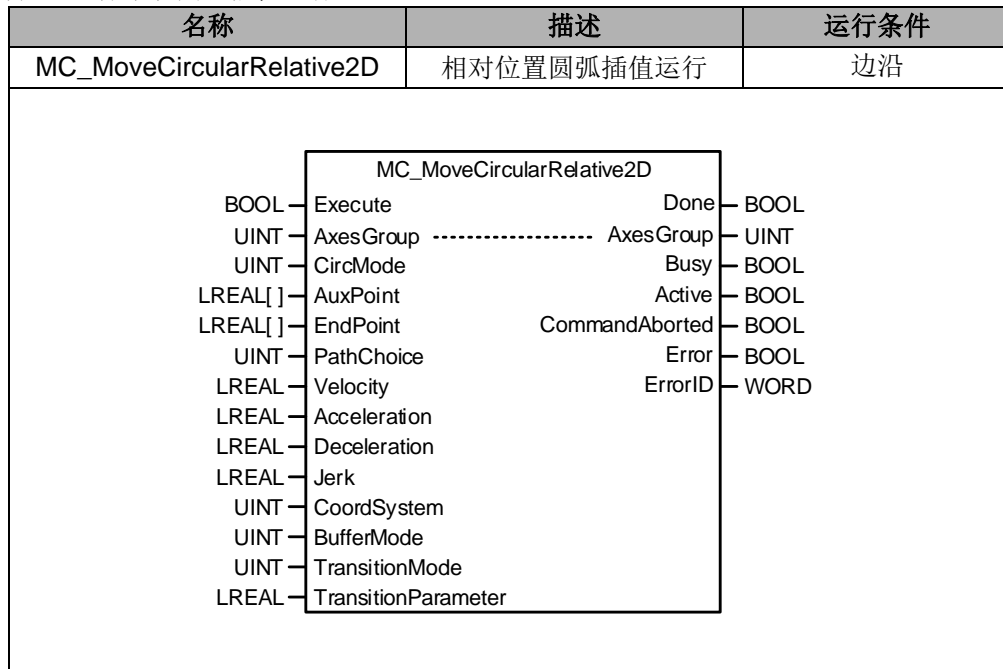


5. 功能块

(1)绝对位置坐标系统圆弧插值运行

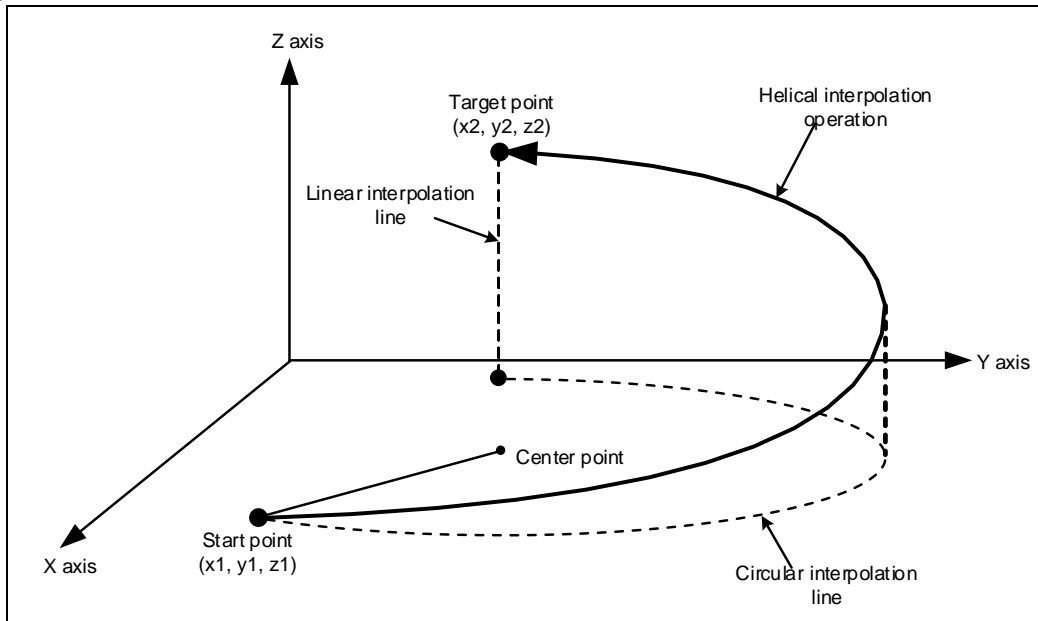


(2)相对位置坐标系统圆弧插值运行



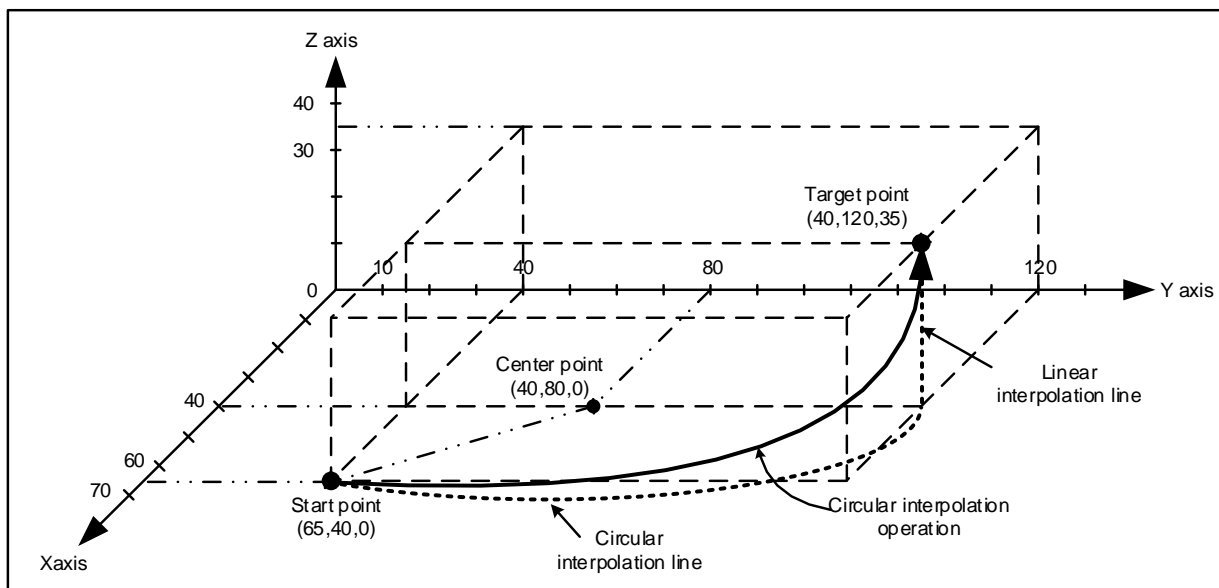
6. 螺旋插值

- (1) 当执行圆弧插值指令(绝对位置坐标系圆弧插值运行 (MC_MoveCircularAbsolute2D), 相对位置坐标系圆弧插值运行 (MC_MoveCircularRelative2D))时, 通过在XY平面内的圆弧轨迹移动以执行圆弧插值, 当Z轴相关的线性插值同步到圆弧插值动作时
- (2) 为了执行螺旋插值, 在'结束点'的Pz设置线性插值的目标位置.



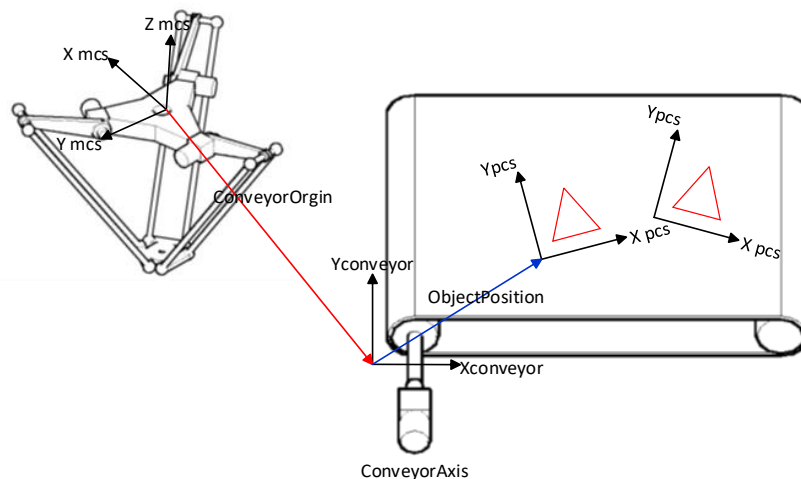
- (3) 限制
用于螺旋插值的指定圆弧插值模式限制条件同样适用于螺旋插值.
- (4) 运行模式

- 起始点: (65.0, 40.0, 0.0)
- 目标点: (40.0, 120.0, 35.0)
- 中心点: (40.0, 80.0, 0.0)
- CircMode: 中心点(1)
- PathChoice: - CCW(1)



8.4.8 传送带的同步运行

在坐标基础运行中，轴组中的一个轴被指定为传送带轴，传送带的对象移动以直线形式移动。



1. 设置和禁止传送带同步运行

MC_TrackConveyorBelt功能块执行用于传送带同步运行的设置。不直接参与运行。在通过MC_TrackConveyorBelt 功能块执行用于传送带同步的设置后，CoordSystem 执行的坐标系统基础功能块在设置为PCS同步后执行传送带运行。完成同步的传送带运行后，再执行那些不进行传送带同步运行的PCS运行，同步传动带运行应禁止。为了通过执行MCS运行或者使用MC_TrackConveyorBelt功能块禁止同步传送带运行，PCS坐标系统应使用MC_SetCartesian 功能块复位。

2. 功能块

(1) 传送带同步设置

名称	描述	运行条件																								
MC_TrackConveyorBelt	传送带同步设置	边沿																								
<table border="1" style="margin: auto;"> <thead> <tr> <th colspan="3">MC_TrackConveyorBelt</th> </tr> </thead> <tbody> <tr> <td>BOOL</td> <td>Execute</td> <td>Done</td> </tr> <tr> <td>UINT</td> <td>AxesGroup</td> <td>AxisGroup</td> </tr> <tr> <td>UINT</td> <td>ConveyorAxis</td> <td>Busy</td> </tr> <tr> <td>ARRAY[0..6] OF LREAL[]</td> <td>ConveyorOrigin</td> <td>Active</td> </tr> <tr> <td>ARRAY[0..6] OF LREAL[]</td> <td>ObjectPosition</td> <td>Error</td> </tr> <tr> <td>UINT</td> <td>CoordSystem</td> <td>ErrorID</td> </tr> <tr> <td>UINT</td> <td>BufferMode</td> <td></td> </tr> </tbody> </table>			MC_TrackConveyorBelt			BOOL	Execute	Done	UINT	AxesGroup	AxisGroup	UINT	ConveyorAxis	Busy	ARRAY[0..6] OF LREAL[]	ConveyorOrigin	Active	ARRAY[0..6] OF LREAL[]	ObjectPosition	Error	UINT	CoordSystem	ErrorID	UINT	BufferMode	
MC_TrackConveyorBelt																										
BOOL	Execute	Done																								
UINT	AxesGroup	AxisGroup																								
UINT	ConveyorAxis	Busy																								
ARRAY[0..6] OF LREAL[]	ConveyorOrigin	Active																								
ARRAY[0..6] OF LREAL[]	ObjectPosition	Error																								
UINT	CoordSystem	ErrorID																								
UINT	BufferMode																									

(2) 传送带同步设置禁止(PCS设置)

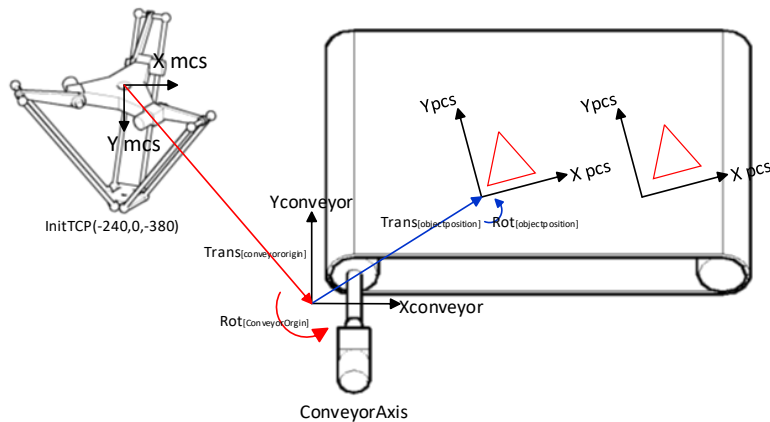
名称	描述	运行条件
MC_SetCartesianTransform	PCS设置	边沿

MC_SetCartesianTransform

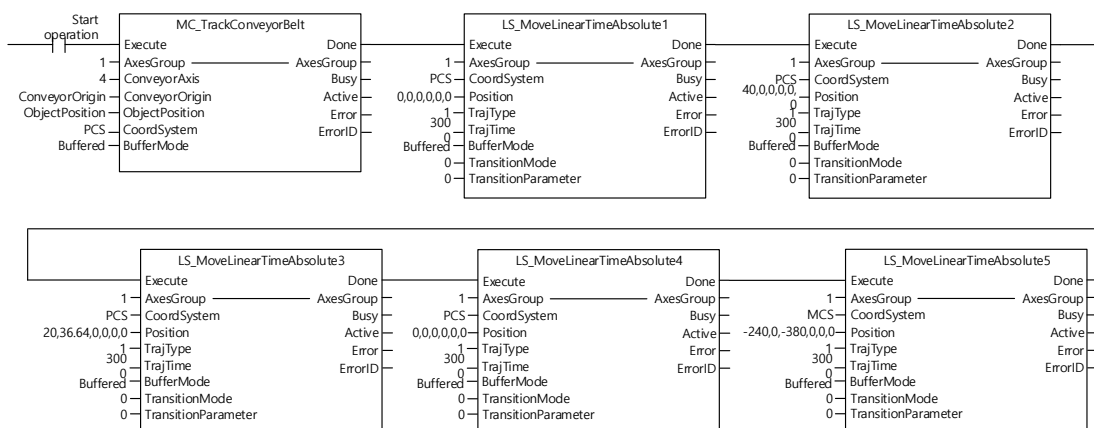
BOOL	Execute	Done	BOOL
UINT	AxesGroup	AxesGroup	UINT
LREAL	TransX	Busy	BOOL
LREAL	TransY	Active	BOOL
LREAL	TransZ	CommandAborted	BOOL
LREAL	RotAngleA	Error	BOOL
LREAL	RotAngleB	ErrorID	WORD
LREAL	RotAngleC		

(3) 传送带同步功能运行实例

传送带同步功能以使用MC_TrackConveyorBelt功能块设置的传动带同步开始。关于MC_TrackConveyorBelt功能块，在ConveyorAxis 输入中输入传送轴值，在ConveyorAxis输入中从机器人的原点输入传送带位置，在ConveyorOrigin 输入中输入从传送原点的产品原点位置。一旦MC_TrackConveyorBelt功能块设置完成，随后应用CoordSystem 输入的LS_MoveLinearTimeAbsolute功能块设置传送同步运行的PCS。同步传送运行在产品中执行绘制三角的运行。完成同步传送带运行后，执行 CoordSystem 设置为 MCS 的 LS_MoveLinearTimeAbsolute功能块返回到传送工作未执行的前一个状态。



功能块	CoordSystem	位置[]	描述
MoveLinearTimeAbsolute1	PCS	0,0,0	移动到ConveyorOrigin
MoveLinearTimeAbsolute2	PCS	40,0,0	绘制三角1
MoveLinearTimeAbsolute3	PCS	20,36.64,0	绘制三角2
MoveLinearTimeAbsolute4	PCS	0,0,0	绘制三角3
MoveLinearTimeAbsolute5	MCS	-240,0,-380	移动机器人到初始位置



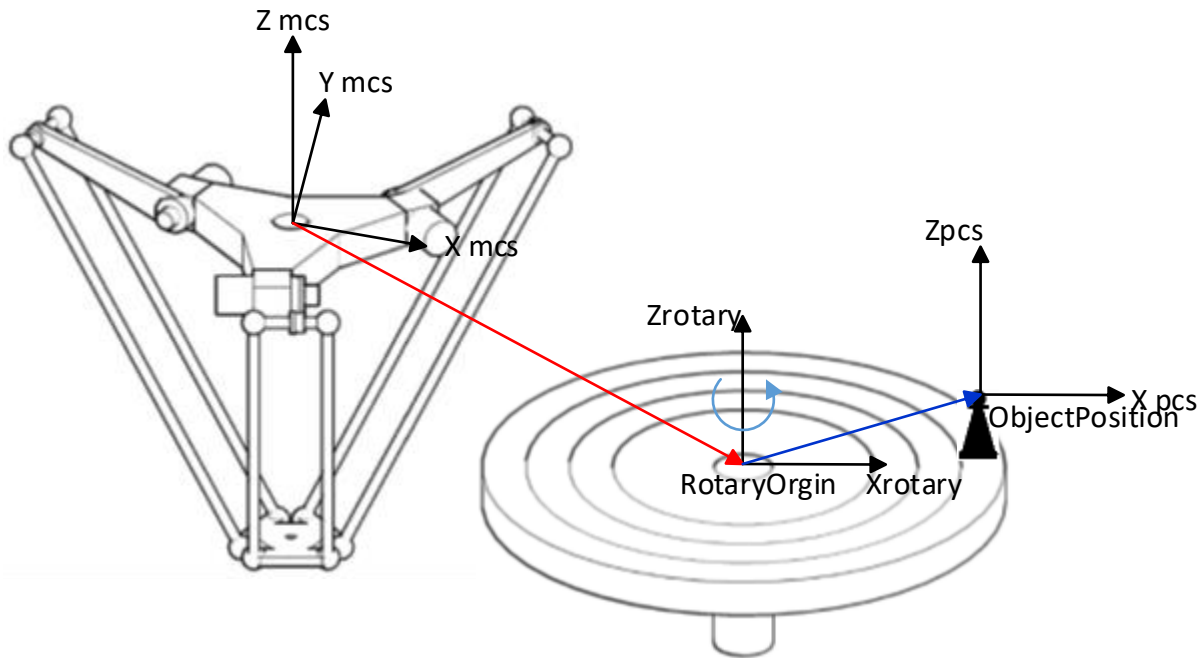
(4) 限制

传送带同步不可以在以下错误状态下设置.

- 在CoordSystem 中设置并执行的值是2以外的值 (错误代码: 0x20BC)
- 设置于ConveyorAxis中的轴无法连接 (错误代码: 0x20C3)
- 设置于CoordSystem中的参数轴单位不是mm/inch(错误代码: 0x20C2)
- 设置于ConveyorAxis中的轴没有设置与无限重复运行 (错误代码: 0x20C6)
- 设置于ConveyorAxis中的轴是应用轴组的组成轴 (错误代码: 0x20C1)
- 关于组成轴, 有一个轴原点位置当前未确定 (错误代码 0x20B0)
- 关于组成轴, 有一个轴执行无限重复运行 (错误代码: 0x20BA)

8.4.9 转盘同步运行

在坐标基准运行中，轴组中的一个轴指定为旋转轴，转盘上的对象移动进行跟踪。



1. 设置和禁止转盘同步运行

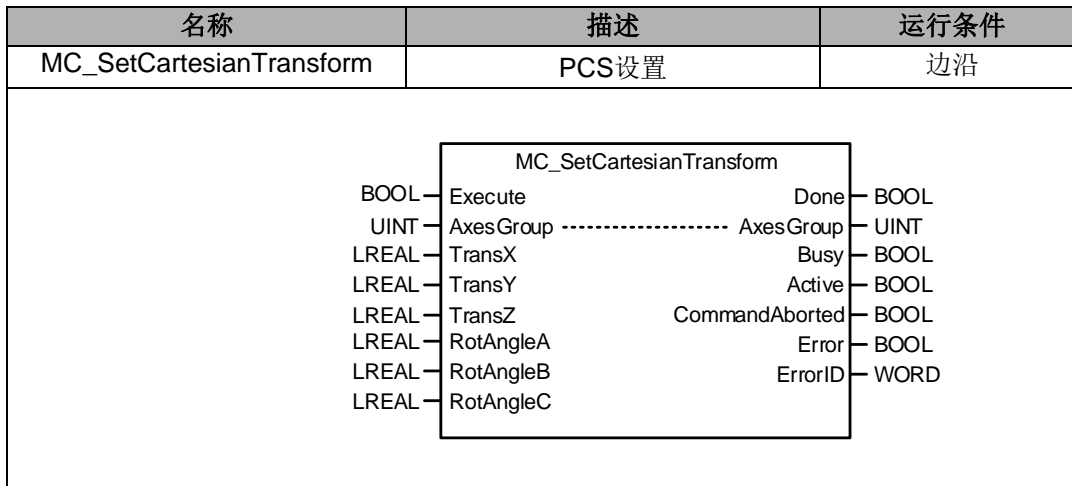
MC_TrackRotaryTable功能块执行用于转盘同步运行的设置。不直接参与运行。在通过MC_TrackRotaryTable功能块执行用于转盘同步的设置后，CoordSystem执行的坐标系统基础功能块在设置为PCS同步后执行转盘运行。完成同步的转盘运行后，再执行那些不进行转盘同步运行的PCS运行，同步转盘运行应禁止。为了通过执行MCS运行或者使用MC_TrackRotary功能块禁止同步转盘运行，PCS坐标系统应使用MC_SetCartesian 功能块复位。

2. 功能块

(1) 转盘同步设置

名称	描述	运行条件																																
MC_TrackRotaryTable	转盘同步设置	边沿																																
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="4">MC_TrackRotaryTable</th> </tr> </thead> <tbody> <tr> <td>BOOL</td> <td>Execute</td> <td>Done</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>AxesGroup</td> <td>AxesGroup</td> <td>UINT</td> </tr> <tr> <td>UINT</td> <td>RotaryAxis</td> <td>Busy</td> <td>BOOL</td> </tr> <tr> <td>ARRAY[0..6] OF LREAL[]</td> <td>RotaryOrigin</td> <td>Active</td> <td>BOOL</td> </tr> <tr> <td>ARRAY[0..6] OF LREAL[]</td> <td>ObjectPosition</td> <td>Error</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>CoordSystem</td> <td>ErrorID</td> <td>WORD</td> </tr> <tr> <td>UINT</td> <td>BufferMode</td> <td></td> <td></td> </tr> </tbody> </table>			MC_TrackRotaryTable				BOOL	Execute	Done	BOOL	UINT	AxesGroup	AxesGroup	UINT	UINT	RotaryAxis	Busy	BOOL	ARRAY[0..6] OF LREAL[]	RotaryOrigin	Active	BOOL	ARRAY[0..6] OF LREAL[]	ObjectPosition	Error	BOOL	UINT	CoordSystem	ErrorID	WORD	UINT	BufferMode		
MC_TrackRotaryTable																																		
BOOL	Execute	Done	BOOL																															
UINT	AxesGroup	AxesGroup	UINT																															
UINT	RotaryAxis	Busy	BOOL																															
ARRAY[0..6] OF LREAL[]	RotaryOrigin	Active	BOOL																															
ARRAY[0..6] OF LREAL[]	ObjectPosition	Error	BOOL																															
UINT	CoordSystem	ErrorID	WORD																															
UINT	BufferMode																																	

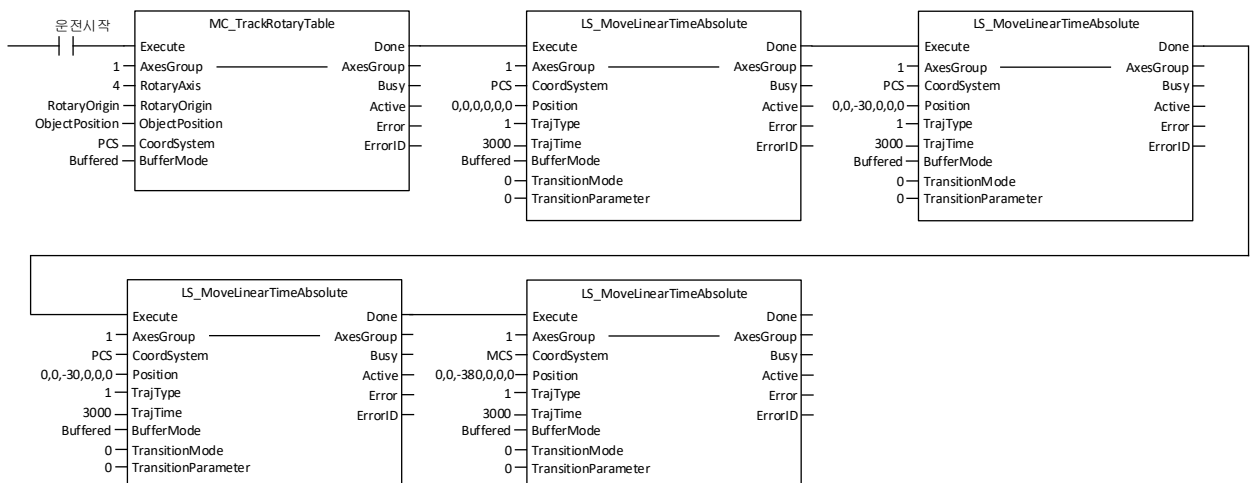
(2) 转盘同步设置禁止(PCS设置)



(2) 转盘同步功能运行实例

转盘同步功能以使用MC_TrackRotaryTable功能块设置的转盘同步开始. 关于MC_TrackRotaryTable功能块, 在RotaryAxis输入中输入旋转轴值, 在RotaryOrigin中输入从机器人原点的转盘中心位置, 在ObjectPosition 输入中输入从转盘中心点的产品原点位置.一旦MC_TrackRotaryTable功能块设置完成, 随后应用CoordSystem 输入的LS_MoveLinearTimeAbsolute 功能块设置旋转同步运行的PCS.同步旋转运行通过Z正向和反向移动执行对象跟踪.同步转盘运行完成后, 执行CoordSystem设置为MCS的LS_MoveLinearTimeAbsolute功能块返回到传送工作未执行的前一个状态.

功能块	CoordSystem	位置[]	描述
MoveLinearTimeAbsolute1	PCS	0,0,0	移动到旋转中心
MoveLinearTimeAbsolute2	PCS	0,0,-30	跟踪对象1
MoveLinearTimeAbsolute4	PCS	0,0,30	跟踪对象2
MoveLinearTimeAbsolute5	MCS	0,0,-380	移动机器人到初始位置



(3) 限制

转盘同步不可以在以下错误状态下设置.

- 在CoordSystem 中设置并执行的值是2以外的值 (错误代码: 0x20BC)
- 设置于RotaryAxis中的轴无法连接 (错误代码: 0x20C3)
- 设置于RotaryAxis中的参数轴单位不是度(错误代码: 0x20C2)
- 设置于RotaryAxis中的轴没有设置与无限重复运行 (错误代码: 0x20C6)
- 设置于RotaryAxis中的轴是应用轴组的组成轴 (错误代码: 0x20C1)
- 关于组成轴, 有一个轴原点位置当前未确定 (错误代码 0x20B0)
- 关于组成轴, 有一个轴执行无限重复运行 (错误代码: 0x20BA)

8.4.10 坐标系统的路径运行功能

坐标系统路径运行功能在特定存储区域保存运行指令信息，并执行保存的运行指令从而间接直接坐标系统运行，例如坐标系统线性插值运行/圆弧插值运行。

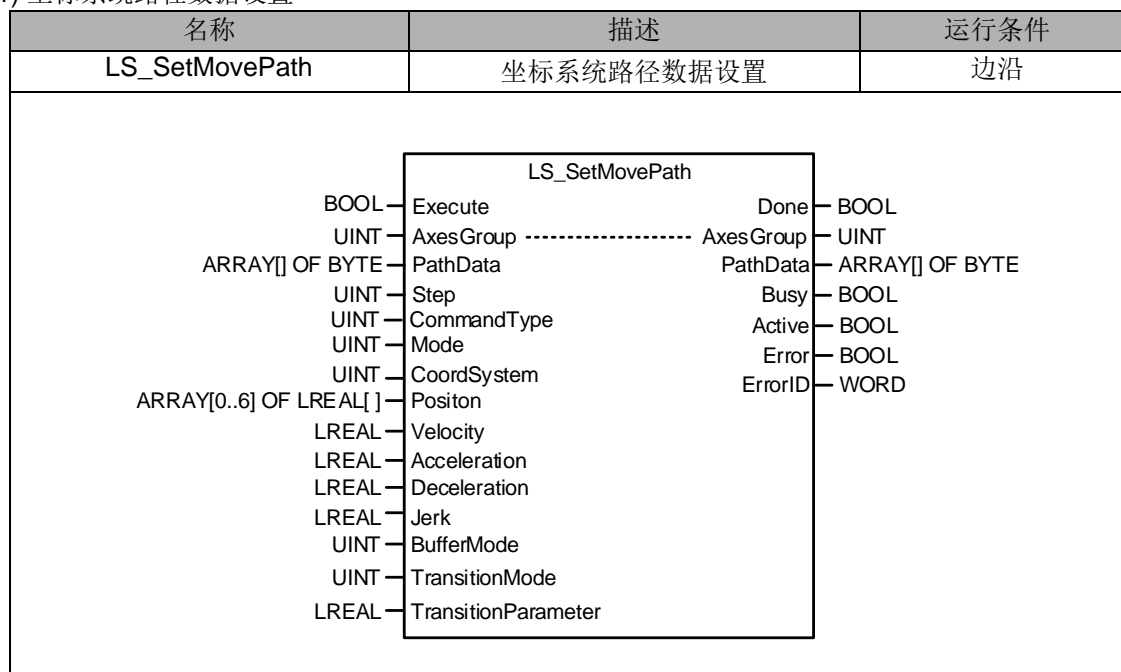
(1) 坐标系统路径运行设置

坐标系统路径数据使用LS_SetMovePath功能块进行设置。设置与LS_SetMovePath的路径数据保存在作为PathData输入的阵列变量中。作为路径数据输入的指定阵列变量，应使用空间足够保存作为输入的坐标系统路径数据的阵列。由于坐标系统路径数据一步的大小为96字节，PathaData应至少使用96阵列。设置于步中的路径运行顺序是指定的。在指令类型中，指定执行运行的指令类型(0:无 1: 1: 坐标系统绝对位置线性插值运行 2: 坐标系统相对位置线性插值运行3: 坐标系统绝对位置圆弧插值运行 4: 坐标系统相对位置圆弧插值运行)。模式输入是一种是否在指令类型中选择圆弧插值的圆弧路径输入，并且可以选择圆弧方向(0: 顺时针 1: 逆时针)。位置是设置目标位置的输入，X, Y, Z, A, B 和C方向顺序输入。

使用LS_RunMovePath功能块执行坐标系统路径运行。当执行坐标系统路径运行时，作为StartStep和EndStep的指定步路径数据顺序执行。即使在路径运行时间内EndStep没有到达，如果步的CommandType值设置为0，则路径运行停止。坐标系统路径运行期间当前执行步编号通过CurStep显示。

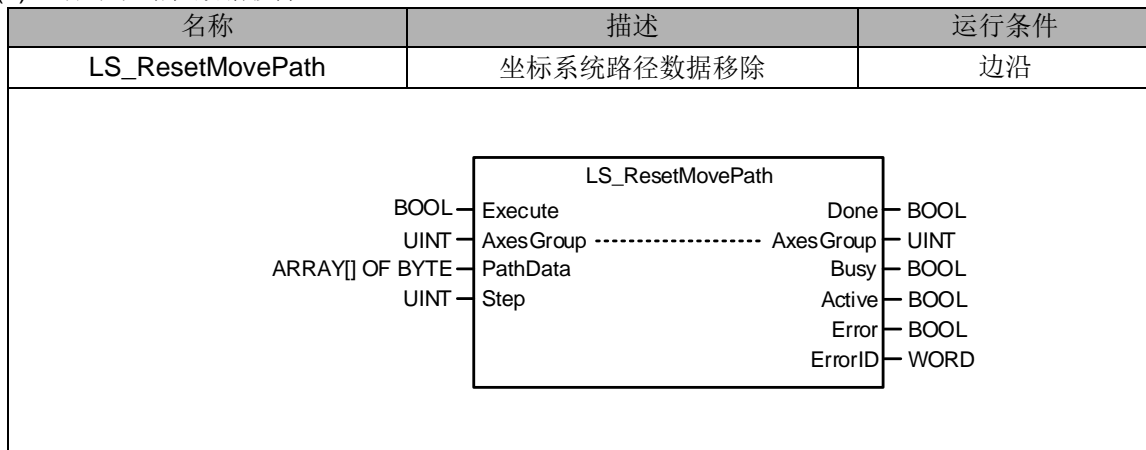
(2) 运动功能块

(1) 坐标系统路径数据设置

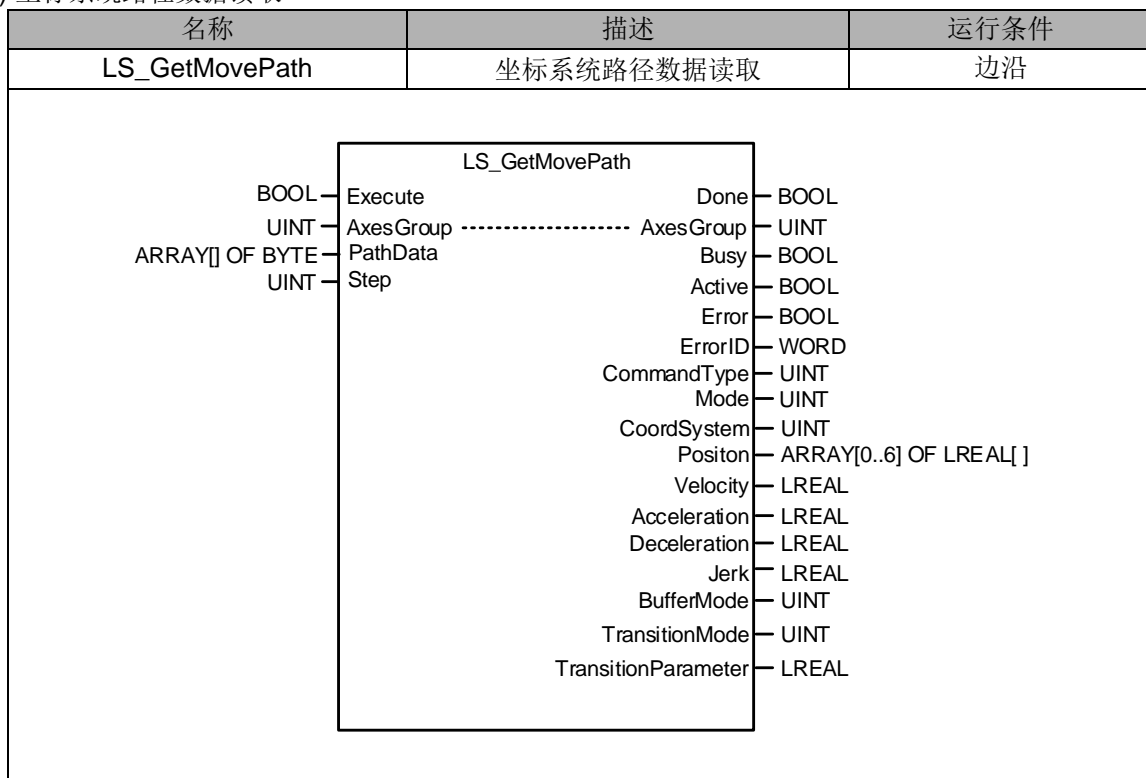


第8章 运动控制功能

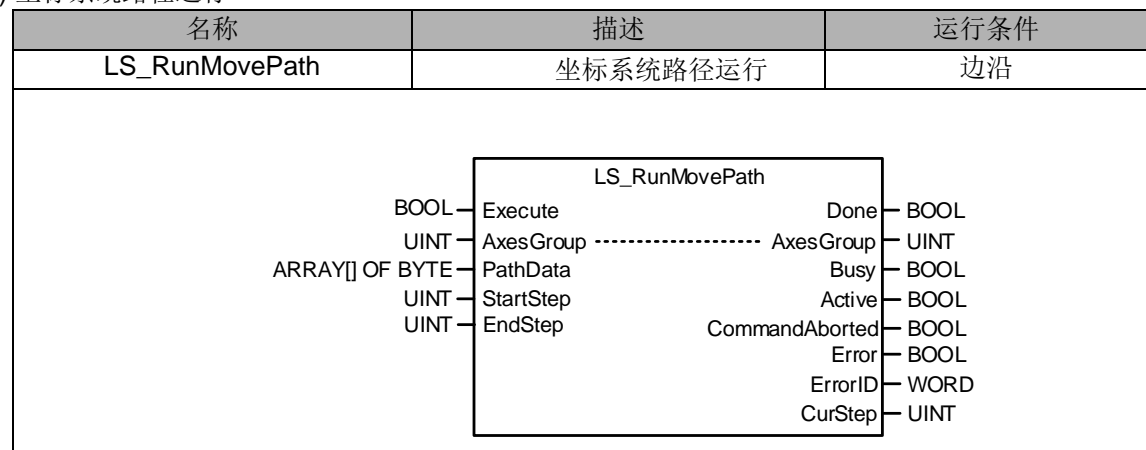
(2) 坐标系统路径数据移除



(3) 坐标系统路径数据读取



(4) 坐标系统路径运行



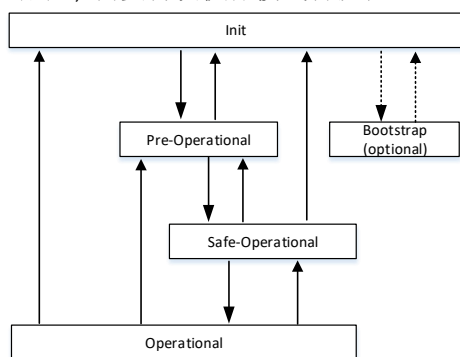
8.5 FoE(通过EtherCAT的文件访问)功能

8.5.1 FoE功能概述

FoE支持具有通过EtherCAT通讯提供的简单文件访问协议，由EtherCAT网络从运动控制器到引导模式状态的从站的固件下载功能。为了使用FoE功能，主站和从站都应支持FoE协议。因此，使用该功能之前需要查看是否支持FoE。

8.5.2 FoE下载

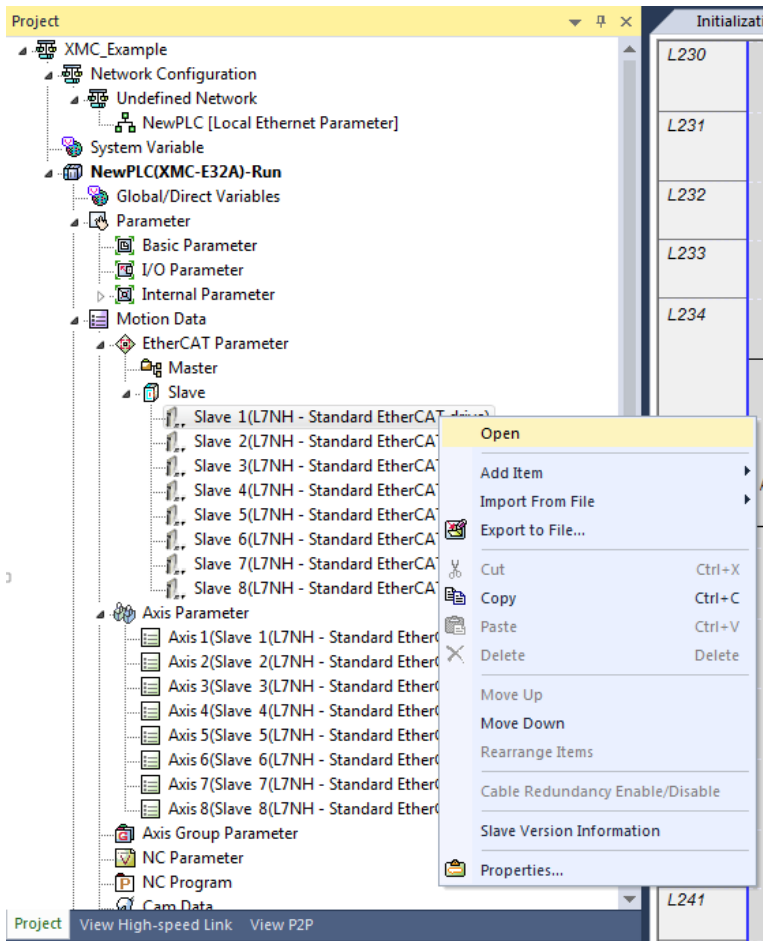
状态机的从站根据其实际支持功能运行。由于FoE功能仅支持Boot(引导程序)模式，在Boot程序状态下，为了使用FoE功能需要将站状态转换为Boot模式。如果运动控制器通过全伺服连接指令执行网络连接，所有连接从站的状态由Init变为Op(操作)模式。因此，模式应由开关切换Op->Init->Boot，从Op模式到Boot模式。当引导模式运行时执行FoE下载。FoE下载完成后，需要再次执行模式切换从 Boot->Init



(1) StateMachine设置

当从站连接时，在工程树中选择从站执行StateMachine设置，从信息对话框中快捷键菜单>>注册信息和在线服务标签。当前从站的StateMachine状态和进入请求在屏幕上显示。由于从站在大多数情况下不支持引导程序，必须查看是否支持引导程序模式。如果引导程序模式设置在不支持该模式的从站中，可能导致故障。

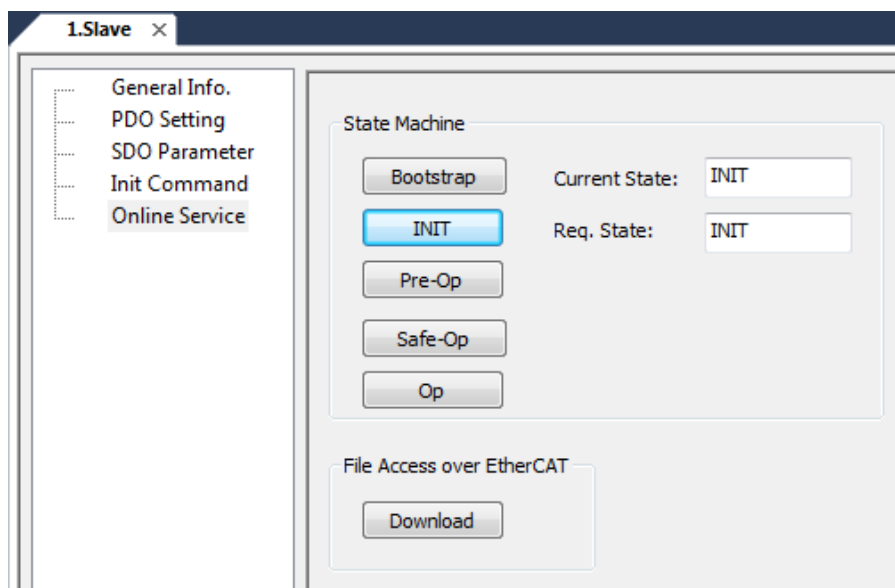
第8章 运动控制功能



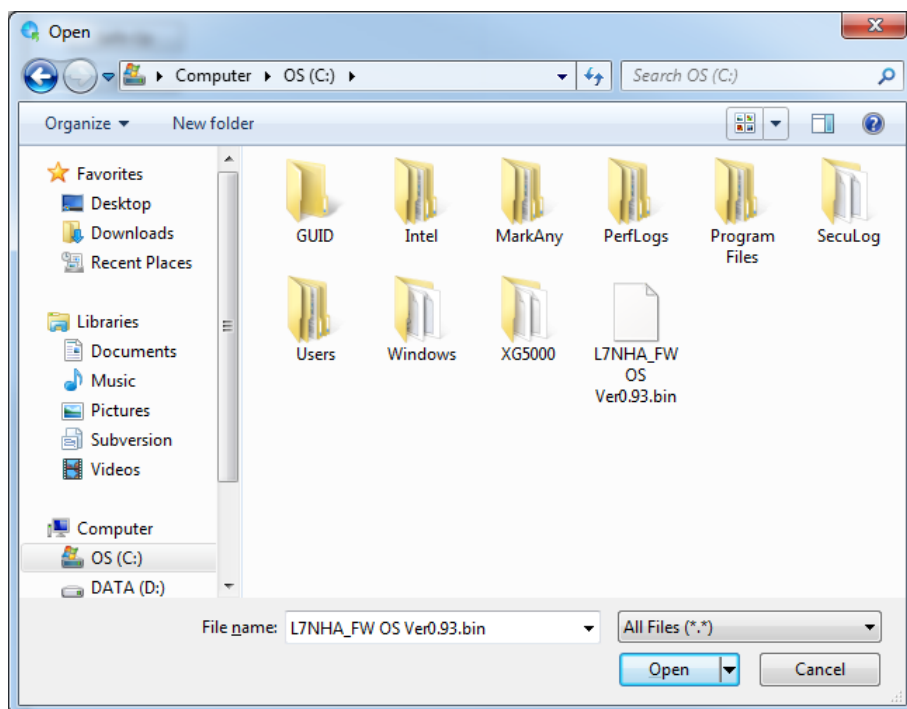
(2) 下载文件

使用FoE协议下载文件. 当StateMachine状态为Boot模式时可以执行FoE下载. 下载FoE文件的步骤根据从站不同而变化. 请参考从轴程序手册.

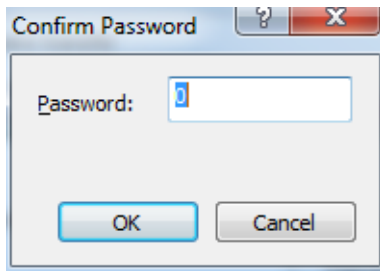
1) 变更Op模式到Boot, 并在FoE中点击下载按钮开始下载.



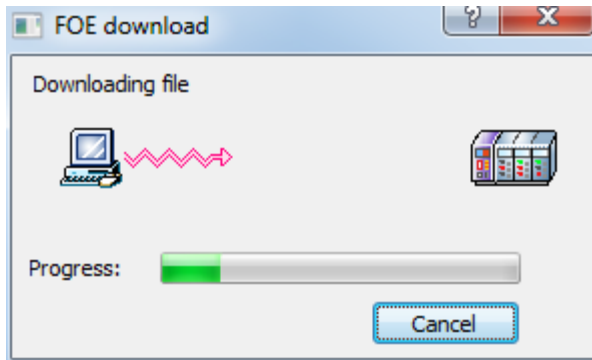
2) 在打开的对话框中选择需要的文件.



3) 在密码确认对话框中输入密码(数字).



4) 选择OK按钮下载文件.



5) 当下载完成后, 变更StateMachine状态为初始化.

第 9 章 NC 控制功能

第 9 章模式运动控制器用户如何创建 G 代码格式的动作程序。

运动控制器可以通过一种被称为 G 代码的脚本语言进行程序动作。第 9 章描述 G 代码编程的基本术语和概念解释，并解释如何配置程序。同样描述例如运动控制器支持的 G、M 和 F 的指令脚本。

9.1 NC 指令

9.1.1 NC 指令定义

用户使用运动控制以指定速度移动设备从任何位置到目标位置，或者控制输入/输出。此时，移动轴根据每个控制环境而以各种形式移动设备。这是用于控制多种轴移动的运动控制程序，而在运动控制器中，用于控制信息构成的运动控制程序中的 G 代码被称为 NC 指令。

9.1.2 控制符定义

运动控制器的 NC 指令具有对于特定字母字符的具体描述。用于 NC 指令的保留字符类型和描述如下。

保留字符	描述
字符设置	
(开始注释
)	结束注释
[做括号
]	右括号
+, -, *, /	四种基本算术运算
A(AND), O(OR)	比较运行
=	分配运行
0~9	数字数据
;	分程序结束
#	变量
地址符	
X	XYZ 直角坐标系的 X 轴(主坐标轴)
Y	XYZ 直角坐标系的 Y 轴(副坐标轴)
Z	XYZ 直角坐标系的 Z 轴(第三轴)
A	在 XYZ 直角坐标系中，旋转轴平行于 X 轴。(当在参数设置中设置旋转轴时)
B	在 XYZ 直角坐标系中，旋转轴平行于 Y 轴。(当在参数设置中设置旋转轴时)
C	在 XYZ 直角坐标系中，旋转轴平行于 Z 轴。(当在参数设置中设置旋转轴时)
U	1 st 附加线性轴(旋转轴，当在参数设置中设置旋转轴时)
V	2 nd 附加线性轴(旋转轴，当在参数设置中设置旋转轴时)
W	3 rd 附加线性轴(旋转轴，当在参数设置中设置旋转轴时)

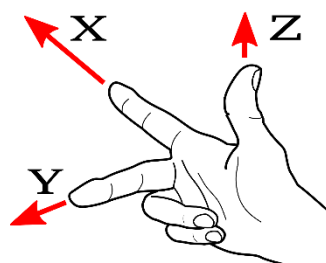
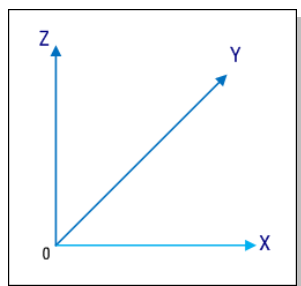
第 9 章 NC 控制功能

保留字符	描述
S	用于指定旋转速度或者主轴的定位指令
G	预备功能
F	馈送比率
M	其他功能
S	指定旋转速度
I	X 轴圆弧插值的旋转中心坐标值
J	Y 轴圆弧插值的旋转中心坐标值
K	Z 轴圆弧插值的旋转中心坐标值
N	语句编号 (序号)
O	用于作为 N 的相同目的
P	用于表示预备功能和其他功能的可选数据.
Q	第二可选数据表达. 钻孔的切割深度等等.
R	圆弧半径
T	工具功能
L	重复率
H	工具长度补偿数量
其他特性	
IF	条件分支运行
GOTO	分支跳转
WHILE, DO	循环迭代声明
END	结束环
%, ;, ()	注释
LE,GE,EQ,LT,GT,NE	比较指令

保留字符	描述
SIN, COS, TAN, ATAN, SQRT, ABS, ROUND, AND, OR, RIX, FUP	其他功能

9.1.3 坐标系

坐标系表示用于运行设备的基础空间。运动控制器使用右手法则坐标系，并且支持 4 种模式：机械坐标系，工件坐标系，局部坐标系，相对坐标系。



[基本坐标系]

(1) 机械坐标系

每个用于运动控制的机械具有其特定的位置设置，坐标系基于特定位置进行设置。该特定位置是机械的“机械原点”，基于该机械原点的坐标系为“机械坐标系”。该“机械原点”和其附属的“机械坐标系”根据应用的运动控制器不同而存在差异。因此，请参考应用机械的指令手册。

一般而言，当电源上电并且机械启动时，首先执行原点返回。原点返回后，复位机械原点为“0”，而此时机械坐标系变更到原点位置。然而，对于配备绝对编码器位置反馈的机械，绝对位置不受原点返回的影响。

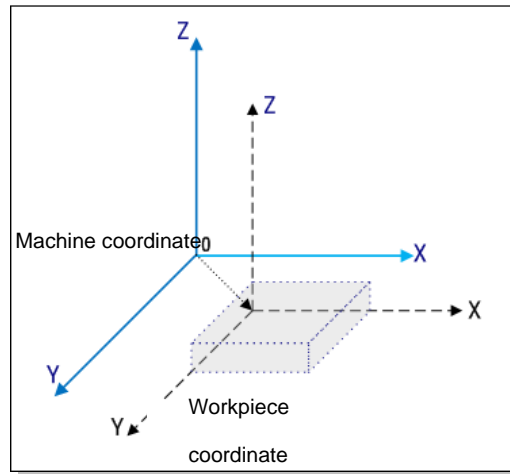
更多关于机械坐标系设置的内容，参考 G53 指令描述。

(2) 工件坐标系

工件坐标系表示原点为产品机械参考点的坐标系。一般而言，工件坐标系的原点由工件坐标系设置指令进行设置。当设置工件坐标系时，从此以后，新坐标中指令运行的原点为产品机械起点。关于工件坐标系的设置(G92)和选择 (G54 ~ G59)，请参考指令描述部分。

(3) 局部坐标系

在工件坐标系任何位置设置的参考点被称为局部坐标系，当在工件坐标系中编程时执行指令，参考包含工件坐标程序在内的新创建坐标系。



[每个坐标系和补偿]

9.2 程序配置

9.2.1 NC 程序

NC 程序由轴控制信息指令文件组成。程序以"程序名称扩展"格式保存,用于运动控制器的 NC 程序扩展为".nc"。NC 程序分为两种类型,"主程序"和"子程序",根据文件性质区分。

"主程序"的名称可以由用户任意指定。例如,比如"主.nc"或者"主控制.nc",可以应用任何英文或者韩文名称。"子程序"以四位数字命名。(例: 1234.nc) "子程序"必要时可以禁用,嵌套调用最多 9 个等级。

(1) 主程序

主程序为控制整个运动程序流程的程序。主程序名称可以以任何*.nc 命名。而对于嵌套调用的情况,基于主程序可以有 10 个等级。

(2) 子程序

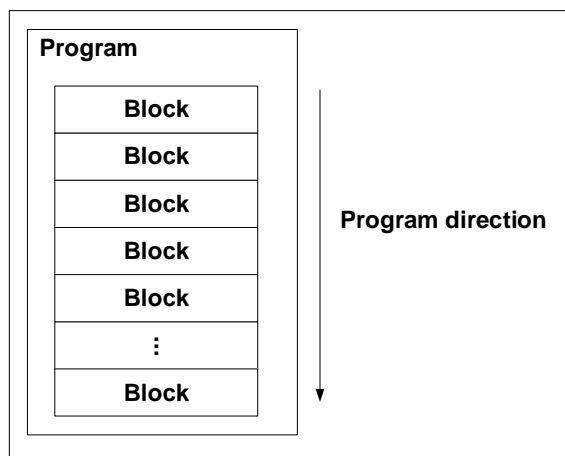
子程序通过主程序的调用执行,仅 4-位数字应用于程序文件名.(0000.nc 到 9999.nc) 扩展为与主程序相同的".nc"。子程序可以从主程序中直接调用,自定义宏所以必须以数字名称写入。

9.2.2 NC 程序配置

(1) 程序基本配置

由多种指令(G / M 代码和指令)创建的 NC 程序可以由运动控制器识别,由每个运行指令的信息块设置组成。

NC 程序以 ST 语言写入。如果使用不是由 NC 指令(G / M 代码和指令)指定的特性,则发生错误。程序以第一个块依次启动。



第 9 章 NC 控制功能

(2) 块配置

由用于驱动机械的基本 NC 指令和指令信息组成。

一个块相当于程序的一行。一个块中的最大字符数量为 300，包含空格字符。每个块中最多使用 10 个 NC 指令。当一个块中的可用字符数或者 NC 指令超出最大限制，则发生错误。

N~~	G~~	X~~.	Y~~.	Z~~.	F~~.~ (M/S/T ...)
语句编号	预指令 (G/M 代码和指令)	指令信息 (坐标指令信息)			辅助指令 (G/M 代码和指令)

NC 程序块如上表显示为正常输入。除非在相同块中有特殊限制符号，每个动作指令的顺序则不重要。然而，按照指令顺序执行，对于模式指令，始终保持运行直到另一个 NC 指令执行。

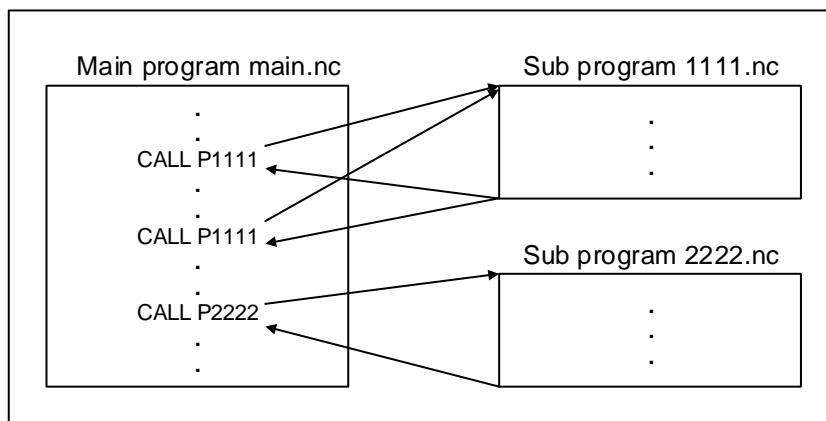
在每个块的头部，有一个包含运动控制的顺序信息的语句编号。语句编号经常引用 GOTO 和 CALL 指令，必要时使用。语句编号同样可以单独写入一个块。

指令信息根据更早使用的预指令(G / M 代码和指令)。虽然不使用指令信息，但是没有错误发生。然而，即使需要指令信息，不使用指令信息的预指令将被忽略。

与预指令一起使用的辅助指令(G / M 代码和指令)进一步提炼预指令的运行信息。

(3) 主程序和子程序

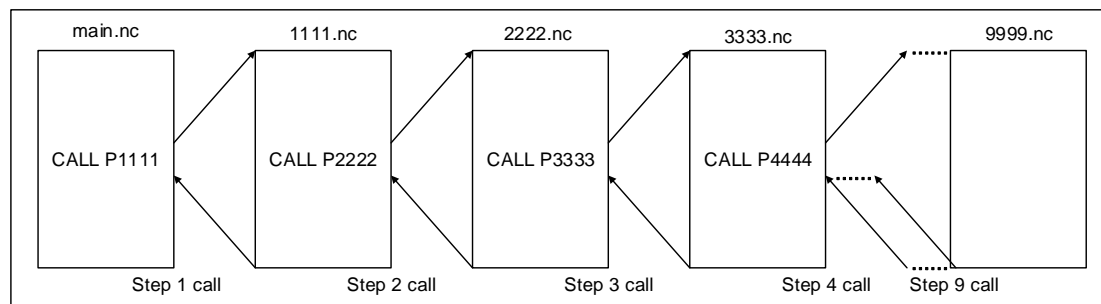
基本上，仅可以配置和控制主程序的 NC 程序。然而，如果多个相同指令或者控制间隔重复，可以使这部分进入子程序或者必要时调用。当调用子程序时，子程序详情随后执行。当子程序的所有指令已执行，主程序的详情将再次运行。



(4) 子程序多重调用

主程序可以调用一个子程序，子程序可以调用其他子程序。如果首次调用子程序为调用步 1，运动控制器的 NC 程序可以如下最多调用 9 个步。如果子程序多重调用超出步 9，则发生错误。

运动控制器的 NC 程序支持最多 9 步的多重调用，但是不允许可以调用子程序本身的"子程序递归调用"或者可能导致子程序间调用停滞的"递归调用"。如果按照此方式调用子程序，可能导致错误。



(5) 用户宏

宏程序是可以模块化的编程方式，并且同样可以在子程序中使用重复功能。不同于子程序调用，宏程序可以传递参数。如果完全重复相同运行，可以调用和使用子程序。然而，如果想要通过发送变量或者指令执行运动程序，或者计算其他运行到变量值，则使用宏。

所有指令进程例如参数传递，除变量范围，运行指令，重复运行等以外的运动指令。以相同方式在子程序和宏程序中执行。子程序调用指令间的不同详情，M98 和宏的调用指令，G65 如下。

- 1) G65 可以指定论证。
- 2) M98 执行其他 M 代码，相同块中的 P 或者 L 和子程序的分支。G65 仅作为分支功能使用。
- 3) 当块中有 O, N, P, L 时，M98 通过单独停止进行停止操作。G65 不停止。
- 4) G65 可以变更局部变量等级，除了 M98。即，即使 G65 指令之前 #Li 进行定义，通过 G65 调用的程序 #Li 是与前一个 #Li 完全不同的变量。
- 5) 如果在 M98 之前定义 #Li，即使通过 M98 调用子程序，#Li 为相同变量。
- 6) 包含 G66 的 G65 可以被调用 4 次。M98 可以最多调用包含 G65 和 G66 的 8 个等级。

(6) 子程序的调用和返回

子程序或者宏程序的调用和返回相同。宏程序由 G65 调用，子程序由 M98 调用。子程序调用和返回的步骤如下。

M98 P_ Q_ R_ L_

M99 P_

M98: 子程序调用

M99: 子程序结束

P _: 对于 M98, 子程序相同(_ 为 4-位数字.)

对于 M99, 返回的块语句编号

Q _: 子程序的启动块语句编号(如果遗漏, 则由第一个块启动).

R _: 子程序的结束块语句编号(如果遗漏, 持续进程直到 M99.)

L _: 子程序调用重复数量

如果没有调用的程序, 或者语法或者论据错误, 则发生错误. 对于警报发生, 请参考附录 2'错误信息&解决方案'.

(7) 主程序重复

当在主程序中执行 M99 时, 主程序详情可以通过重复模式执行. 如果没有执行 P_, 从主程序的第一个块重复执行. 如果没有执行 L_, 则无限重复运行.

M99 P_ L_

M99: 主程序重复

P _: 重复启动块的语句编号

L _: 调用重复数量

9.2.3 数据

(1) 用于 NC 程序的数据类型

NC 程序使用用于每个轴指令, 馈送率指令, DWELL 指令, 宏变量等的数值数据. 当应用每个操作时, 常数直接用于程序.

此时,数据类型范围如下用于运动控制器的 NC 程序. 如果应用错误的数据范围, 发生警告. 对于警告信息等的位置, 请参考附录 2, 错误信息&解决方案'.

整数可用范围

-2,147,483,648 ~ 2,147,483,647

- 可显示实数范围

可以显示最多 12 位, 包含小数点和符号.

(2) 单位系统

对于用于 NC 程序的数字，根据是否输入小数点(".")而应用不同单位系统。取决于设置与运动控制器 NC 通道参数中的小数点查看项目。

Group	Name	Channel 1
Basic Settings	Target Machining Quantity	0
	Target machining quantity at M99 repetition	0
	Check of decimal point	1: Unused
	Keep workpiece coordinate system	0: Keep
	Macro call on T-code command	0: Do not call
	DWELL Method	0: Time
	Block selection at NC reset	0: Keep the Current Block
	Statement number search	0: Search
	Minimum command unit	0 mm
	Whether to use G22 [No traveling area]	0: Used
	Inner/Outer side of G22 [No traveling area]	0: Inner side
	Whether to use the 3rd [No traveling area]	0: Used
	Rotary Axis of Cylindrical Interpolation	0: X Axis
	Linear axis for interpolating the polar coordi	0: None
	Rotary axis for interpolating the polar coordi	0: None
Monitoring time for in-position completion	5000 ms	

如果参数设置为"0: 查看", 值内部变更, 如下实例显示.

X100	→ 100 um	= 100 / 1000.0	= 0.1mm
X100.	→ 100.0 mm	= 100 / 1.0	= 100.0 mm
例)			
X10.4	→ 10.4 mm	= 10.4 / 1.0	= 10.4 mm
X104	→ 104 um	= 104 / 1000.0	= 0.104 mm

仅当直接输入轴坐标信息时, 指定单位转换有效.

如果小数点查看参数设置为"0: 查看", 用于 G04 (TIME)指令的单位系统如下.

G04 X1	→ 1 msec	= 1 / 1000.0	= 0.001 sec
G04 X1.0	→ 1.0 sec	= 1 / 1.0	= 1 sec

9.3 NC 指令

NC 指令为基于三种数据类型的基本描述: 移动的动作类型, 目标位置和目标速度. 位置指令和速度指令的基本格式如下.

9.3.1 NC 位置指令基本格式

运动控制器支持两种指令类型; 使用“X, Y, Z, A, B, C, U, V, W, S”指令方式, 使用“I, J, K”指令方式

(1) 绝对/相对位置 (X, Y, Z, A, B, C, U, V, W, S)

```
X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_
```

X_ Y_ Z_ ~ S_: 指定轴位置

轴位置指令, “X_ Y_ Z_ ... ”主要用于点坐标中修改位置, 该点为插值或者穿过指令时每个轴最终移动的点.

希望移动的坐标位置应在“X, Y, Z, ... S”后指定. G 代码的指定轴数量限制为 10. 运动控制器可以通过运动控制最多控制 32 轴, 但是通过 NC 程序的轴最大数量为 10 轴, 更多轴可以通过运动控制程序的轴指令进行控制.

对于绝对指令的情况, 指定馈送目标点的坐标值(馈送结束点坐标值). 如果是相对指令, 指令从当前位置到馈送目标点的增量值.

当指定希望移动的坐标位置时, 根据是否使用“.”而指令的运行模式不同. 请参考“9.2.3 数据单位系统(2)”中的位置公式

```
G90  
G00 X100. Y100. Z100. U100.    %快速移位, X-轴 100, Y-轴 100, Z-轴 100, U-轴 100  
G01 X150. Y200  
U300.  
Z325.  
M02
```

(2) 指定圆弧(I, J, K)中心点

I_ J_ K_

I_ J_ K_: 圆弧插值的圆弧中心点位置指令

轴位置指令, 当执行圆弧插值的圆弧中心点时, "I_ J_ K_"用于坐标中指定每个原点位置.

可以指定"I_ J_ K_"之后的圆弧中心点位置.

"I_ J_ K_" 必须指定从当前位置到原点的增量值.

当指定坐标位置时, 根据是否使用"." 决定指令运行模式. 请参考"9.2.3 数据单位系统(2)"中的位置公式.

```
G90
G02 X100. Y100. I50. J50. %顺时针圆弧插值, X-轴 100, Y-轴 100, 中心点(X50, Y50)
M02
```

以上程序显示穿过目标点, X-轴 100, Y-轴 100 和 X-轴 50, Y-轴 50 的顺时针圆弧插值原点直接坐标.

(3) 速度指令(F)

速度指令具有指导插值指令的速度的功能.

如果没有单独速度指令, 以 NC 通道参数中切削低速中的基本速度运行.

Group	Name	Channel 1
Cutting Feed Settings	Upper speed limit of the cutting feed	10000 mm/m
	Lower speed limit of the cutting feed	1000 mm/m
	Acc./Dec. method of the interpolation	1: Acc./Dec. before Intp.
	Blocks opr. for Acc./Dec. before Intp.	1: Buffered
	Cutting feed acceleration (Before-Intp.)	200 mm/s ²
	Cutting feed deceleration (Before-Intp.)	200 mm/s ²
	Cutting feed jerk (Before-Intp.)	0 mm/s ³

速度指令仅在插值指令中有效, 在快速移位指令中无效.

F_

F_: 速度指令

第 9 章 NC 控制功能

速度指令指定插值指令运行速度。

速度指令可以指示每个插值指令或者单独指示。

由于速度指令为模式指令，一旦指示，则对于插值指令运行有效，直到另一个速度指令执行。

对于线性插值速度指令，运行速度与"9.2.3 数据单位系统(2)"速度格式相同。

对于圆弧插值指令，速度指令计算为切线方向的线性速率。

```
G90
F5000
G02 X100. Y100. I50. J50.      % 顺时针圆弧插值, 速度: 5000
G01 X400. Y250. Z300. F3500   % 线性插值, 速度: 3500
M02
```

9.3.2 NC 指令列表

用于运动控制器的 NC 指令(G / M 代码和其他指令)如下。

分类	程序指令	功能
G 代码指令	G00	快速定位控制
	G01	线性插值馈送控制
	G02	顺时针圆弧/螺旋插值
	G03	逆时针圆弧/螺旋插值
	G04	停留功能
	G09	精准停止
	G17	选择圆弧插值平面(XY 平面)
	G18	选择圆弧插值平面(ZX 平面)
	G19	选择圆弧插值平面(YZ 平面)
	G21	标准输入
	G22	行程查看功能 ON
	G23	行程查看功能 OFF
	G27	原点返回查看
	G28	自动原点返回
	G29	在自动原点返回
	G30	自动 2 nd 和 3 rd 原点返回
	G40	取消工具直径补偿

分类	程序指令	功能
G 代码指令	G41	补偿工具直径到左侧
	G42	补偿工具直径到右侧
	G43	以+方向补偿工具长度
	G49	取消工具长度补偿
	G52	设置局部坐标系
	G53	选择机械坐标系
	G54	选择工件坐标系 1
	G55	选择工件坐标系 2
	G56	选择工件坐标系 3
	G57	选择工件坐标系 4
	G58	选择工件坐标系 5
	G59	选择工件坐标系 6
	G60	单方向定位
	G90	绝对指令
	G91	增量指令
	G92	设置工件坐标系, 主轴最大速度
	G94	每分钟馈送模式指令
	G95	每旋转馈送模式指令
	G107	圆柱插值模式设置
	G112	极坐标插值模式 ON
G113	极坐标插值模式 OFF	
M 代码	M00	程序停止
	M01	选择停止
	M02	PROGRAM END
	M03	主轴正向旋转
	M04	主轴反向旋转
	M05	主轴停止
	M06	工具变更
	M08	冷却剂 ON
	M09	冷却剂 OFF
	M30	程序结束
	M98	辅助程序调用
	M99	辅助程序结束

第 9 章 NC 控制功能

分类	程序指令	功能
定位指令	X, Y, Z, A, B, C, U, V, W, S	指令轴位置
	I, J, K	圆弧插值的每轴旋转中心点坐标
速度指令	F	馈送率指令
停留时间	X	指定停留时间
其他指令	N	指定语句编号
	P	指定子程序调用数量
	IF	条件分支指令和条件运行
	GOTO	分支指令
	WHILE, DO	重复执行特定程序
	END	环结束
	%, ;	注释处理指令
	LE, GE, EQ, LT, GT, NE	比较指令
	AND, OR, XOR, +, -, *, /	运行指令
	=	赋值运算符
	SIN, COS, TAN, ATAN, SQRT, ABS, ROUND, AND, OR, RIX, FUP	数学运算功能

9.3.3 NC 指令描述

(1) G 代码

G 代码定义指令类型，例如机械运行期间每轴的馈送和机械方式，并且这是执行机械驱动和 NC 程序运行的指令,等等. 如下有两种 G 代码类型.

分类	描述
单次G代码指令	G代码指令仅在G代码执行块中有效
模式G代码指令	G代码指令从功能块开始直到其他命令解除为止，有效的G代码命令语

模式指令

```
G01 X10. F100
Y100.
Z300.
G00 X100
```

G01 为在如上程序中显示的模式指令，所以 G01 指令将直到 G00 完成时执行，即使 G01 指令没有单独指定。

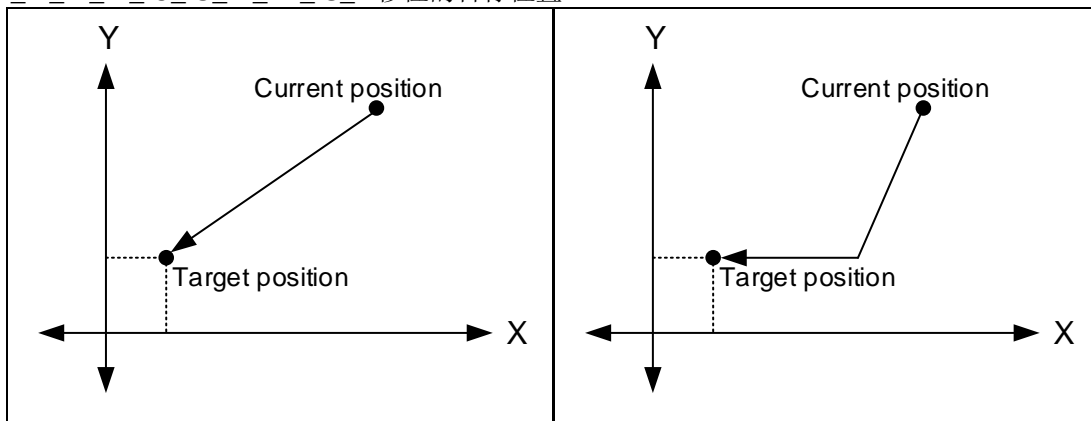
1) 快速移位 (G00)

```
(G90, G91) G00 X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_
```

G90, G91: 绝对/相对指令

G00: 快速定位控制指令

X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_: 移位的目标位置



[移位的当前位置和目标点]

[行程部分的形状]

如上左图显示，快速移位(G00)迅速转换指定轴从指令信息或者当前位置提供的坐标 X,Y 点到指令信息的增量位置。在 G00 指令下,根据设置与每个轴的 G00 馈送率移动。

G00 是对于每个轴单独移位。由于短行程的首先到达目标点，行程部分的形状不是如上的直线。

快速移位指令为模式指令，所以一旦指示，对于轴移位指令始终有效直到另一个移位指令执行。

第 9 章 NC 控制功能

```
G90
G00 X100 Y100
Z100
```

以上程序为使用 G90 设置绝对指令的实例。在此以后，快速移位 X 和 Y 轴到(100,100)点，然后再次移位 Z 轴到 100 点。

快速移位相关参数如下。

NC 参数	组	参数名
NC 通道/轴参数	快速移位	快速移位加速
		快速移位减速
		快速移位加速度
		快速移位速度

NC 参数	组和参数名		
	Group	Name	X Axis
NC 通道/轴参数	Rapid Traverse Settings	Rapid traverse acceleration	500 mm/s ²
		Rapid traverse deceleration	500 mm/s ²
		Rapid traverse jerk	0 mm/s ³
		Rapid traverse speed	10000 mm/m

2) 线性插值 (G01)

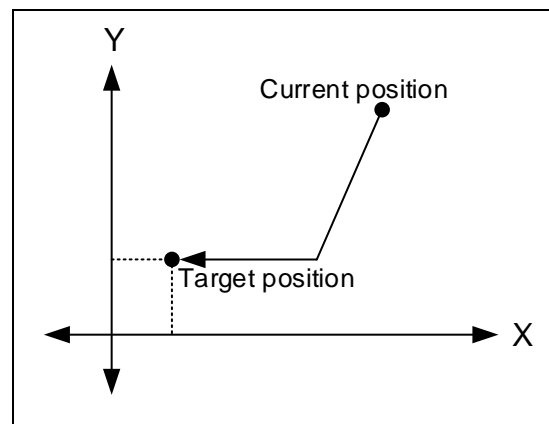
```
(G90, G91) G01 X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_ (F_)
```

G90, G91: 绝对/增量指令

G01: 插值馈送控制指令

X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_: 移位目标位置

F_: 馈送率



线性插值功能为以 F 指令设置的速度以直线同步移位每轴到指令位置，为了执行如上图显示的目标机械(例. 切削). 对于增量指令(G91)，通过指令信息以直线形式从当前轴位置移动到增量位置.

由于 G01 指令为模式指令，一旦指示，根据轴馈送指令持续有效，直到另一个馈送指令执行. 馈送速度指令根据"F"代码只是. 有两种馈送方式; 每旋转馈送和每分钟馈送. 一般而言，应用每分钟馈送.

X 轴馈送率: D_x

Y 轴馈送率: D_y

Z 轴馈送率: D_z

$$D = \sqrt{D_x^2 + D_y^2 + D_z^2}$$

$$\text{X 轴馈送率 } F_x = D_x / D \times F$$

$$\text{Y 轴馈送率 } F_y = D_y / D \times F$$

$$\text{Z 轴馈送率 } F_z = D_z / D \times F$$

如以上公式显示根据每轴距离每个轴的馈送率不同.


```
G90
G01 X50 Y35 F3000    % 插值馈送控制, 移位目标位置(X=50, Y=35), 速度 3000
G91
X100 Y55            %插值馈送控制
```

以上程序显示在绝对指令下以速度 3000 执行插值馈送控制到点 X 轴 50 和 Y 轴 35 的实例, 然后在增量指令下执行 X 轴到 100 和 Y 轴到 55 的增量馈送控制.

对于"X100. Y55.", 如上所述, G01 和 F 代码为模式指令所以在没有单独指令的插值馈送控制下运行.

3) 圆弧插值 (G02/G03)

```
(G90, G91) G17 (G02, G03) X_ Y_ (I_ J_, R_) (F_)
(G90, G91) G18 (G02, G03) X_ Z_ (I_ K_, R_) (F_)
(G90, G91) G19 (G02, G03) Y_ Z_ (J_ K_, R_) (F_)
```

G90, G91: 绝对/增量指令

G17, G18, G19: 指定执行圆弧插值的平面

G02, G03: 顺时针, 计数器顺时针圆弧插值

X_ Y_ Z_: 移位目标位置

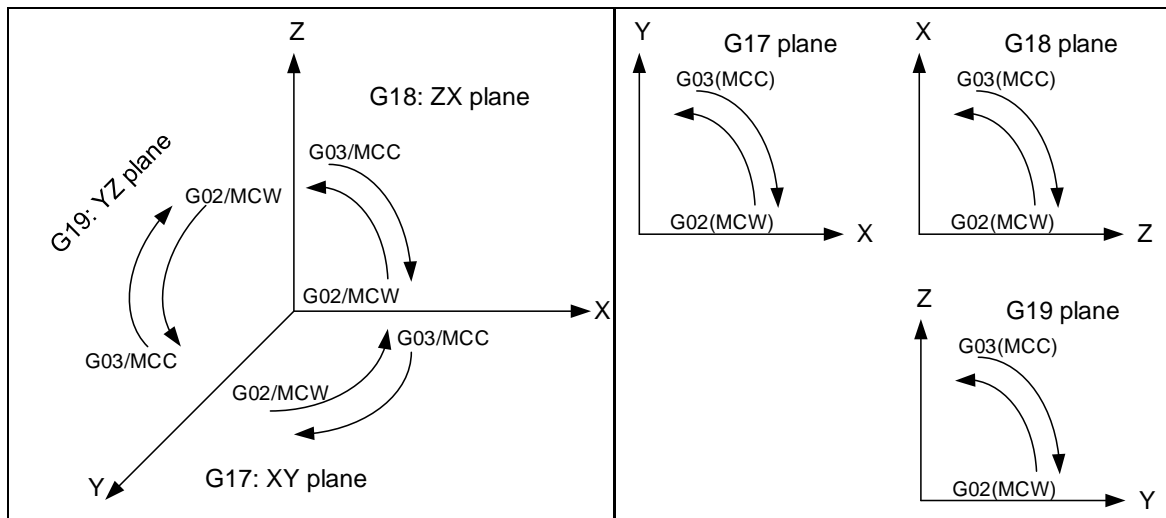
I_ J_ K_/R_: 圆弧参考点或半径

F_: 馈送率

圆弧插值 (G02 / G03)是通过"F"以指定速度执行圆周馈送的指令, 该"F"基于指令或者从中心点到移位目标位置的计算. 在圆弧插值指令中, 速度表示切线方向的线性速率.

对于圆弧插值指令, 在执行前需要选择执行圆弧插值的平面.

NC 程序具有指定每个平面的指令; G17 定义 XY 平面, G18 定义 ZX 平面, 和 G19 定义 YZ 平面. 如果在选择平面而不输入指令信息, 则发生错误.



圆弧插值通过作为指令信息设置的圆弧中心点或者圆弧半径执行.为了执行圆弧插值,执行中必须应用以上其中两者之一.

当在圆弧插值指令中同时应用使用 I, J,和 K 的参考点方式和使用"R"代码的半径方式时, I, J,和 K 指令将被忽略,然后以使用"R"的半径指令执行圆弧插值.

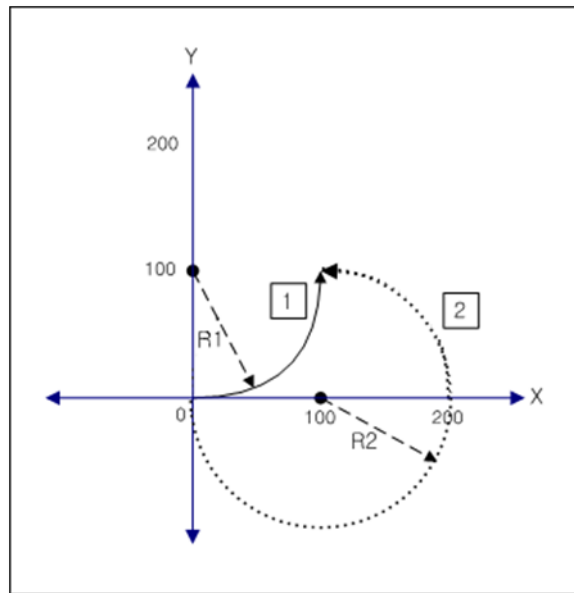
当通过指定中心点执行圆弧插值时,圆弧插值指令的中心点指令信息(I, J, K)始终为从起点位置到中心点的增量位置,无论是绝对/增量指令(G90 / G91). 中心点指令信息 I, J,和 K 分别对应于 X, Y, 和 Z. 如果 I, J, K 指令信息的值为"0",可能被忽略.

当当前位置和移位目标位置相同,指定中心点的圆弧插值可以执行一个 360 度正圆.

指定 R(半径)的圆弧插值不指定确定圆弧部分的中心点,但是仅圆弧形 R(半径)从当前位置到移位目标位置.

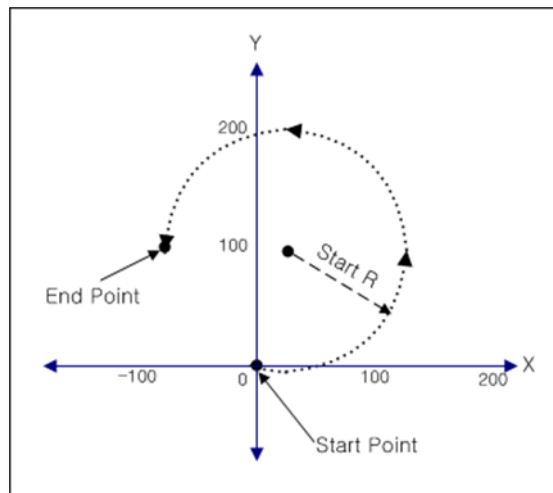
当使用 R (半径)指定方式执行圆弧插值时,圆弧中心点有两个形状.此时,如下所示运动控制器通过选择最短圆弧的中心点执行圆弧插值到移位目标位置.

不同于指定中心点的圆弧插值,指定 R(半径)圆弧插值不可以执行一个 360 度正圆.



[指定半径(R)圆弧插值]

圆弧插值期间，如果圆弧启动和结束不同，发生警报。如果在错误半径范围内，移位到原轨迹，然后以直线到达最终位置。



[不同旋转半径的部分圆弧插值]

G90
G00 X0 Y0 Z0
% XY 平面

G90
G17
G02 X50 Y50 I50 F100 % 顺时针圆弧插值, 中心点(X=X+50, Y=0), speed 100
G03 X0 Y0 R50 % 计数器顺时针圆弧插值, R(半径)=50
G91 % 相对坐标
G03 X100 Y100 J100 % 计数器顺时针圆弧插值, 中心点(X=0, Y=Y+100)
G02 X-100 Y-100 R100 % 顺时针圆弧插值, R(半径)=100

% ZX 平面

G90
G18
G02 Z50 X50 K50 F200 % 顺时针圆弧插值, 中心点(X=X+0, Z=Z+50), 速度 00
G03 Z0 X0 R50 % 计数器顺时针圆弧插值, R(半径)=50
G91
G03 Z100 X100 I100 % 计数器顺时针圆弧插值, 中心点(X=X+100, Z=Z+0)
G02 Z-100 X-100 R100 % 顺时针圆弧插值, R(半径)=100,
 % 移位目标位置(X=X-100, Z=Z-100)

% YZ 平面

G90
G19
G02 Y50 Z50 J50 F300 % 顺时针圆弧插值, 中心点(Y=Y+50, Z=Z+0), 速度 300
G03 Y0 Z0 R50 % 计数器顺时针圆弧插值, R(半径)=50
G91
G03 Y100 Z100 K100 % 计数器顺时针圆弧插值, 中心点(Y=Y+0, Z=Z+100)
G02 Y-100 Z-100 R100 % 顺时针圆弧插值, R(半径)=100

% 正圆

G17
G02 I50 % 顺时针圆弧插值(360-度正圆), 中心点(X=X+50, Y=Y+0)
G03 J50 % 计数器顺时针圆弧插值(360-度正圆), 中心点(X=X+50, Y=Y+0)
G02 I50 J50 % 顺时针圆弧插值(360-度正圆), 中心点(X=X+50, Y=Y+50)

% I 忽略 R 应用
G02 X-100 I30 R50

第 9 章 NC 控制功能

圆弧插值相关参数如下.

NC 参数	组名和参数名		
	Group	Name	Channel 1
NC 通道参数	Circular Machining Settings	Operation when the circular alarm occurs	0: Generate an alarm
		Speed limitation for circular interpolation	0: Unused
		Tolerance of arc radius	0 mm
		Circular radius with speed limitation	0 mm
		Upper speed limit of circular interpolation	1000 mm/m
		Lower speed limit of circular interpolation	100 mm/m
		Circular interpolation acceleration	100 mm/s ²
		Circular interpolation deceleration	100 mm/s ²
		Circular interpolation jerk	0 mm/s ³

4) 螺旋插值(G02/G03)

```
(G90, G91) G17 (G02, G03) X_ Y_ (I_ J_, R_) Z_ F_
(G90, G91) G18 (G02, G03) X_ Z_ (I_ K_, R_) Y_ F_
(G90, G91) G19 (G02, G03) Y_ Z_ (J_ K_, R_) X_ F
```

G90, G91: 绝对/相对指令

G17, G18, G19: 指定执行圆弧插值的平面

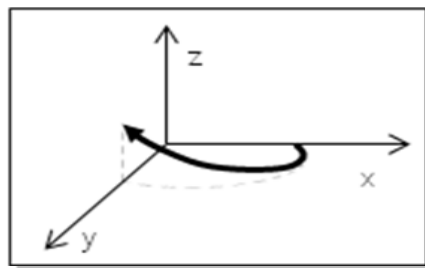
G02, G03: 顺时针, 计数器顺时针圆弧插值

X_ Y_ Z_: 移位目标位置

I_ J_ K_/R_: 圆弧半径参考点

F_: 馈送率

螺旋插值指示圆弧插值指令中不指定平面的另一个轴, 轴以直线同步, 并按照圆弧插值进程. 即, 当执行 XY 平面 G17 时, 可以转换 Z 轴.



```

G90
G00 X0 Y0 Z0

% XY 平面
G90
G17
G02 X50 Y50 I50 Z10 F100 % 顺时针圆弧插值, 中心点(X=X+50, Y=0), Z 位置 10, 速度 100
G03 X0 Y0 Z 20 R50 % 计数器顺时针圆弧插值, R(半径)=50,
Z 位置 20

```

5) 停留功能(G04)

```
G04 (X_, P_)
```

G04: 停留指令

X_, P_: 停留时间指令信息(sec, msec)

停留指令 (G04)是按照"X"或者"P"指定时间停止的指令, 然后执行下一个.

X 的单位为 sec, P 的单位为 msec.

```

G90
G00 X0. Y0. Z0.
G01 X100. Y100. F1500
G04 P100          % 停留时间: 100 msec
G00 X500. Y500.
G04 X1           % 停留时间: 1 sec
G91
G01 X100. Y100. F1500

```

第 9 章 NC 控制功能

停留指令相关参数如下.

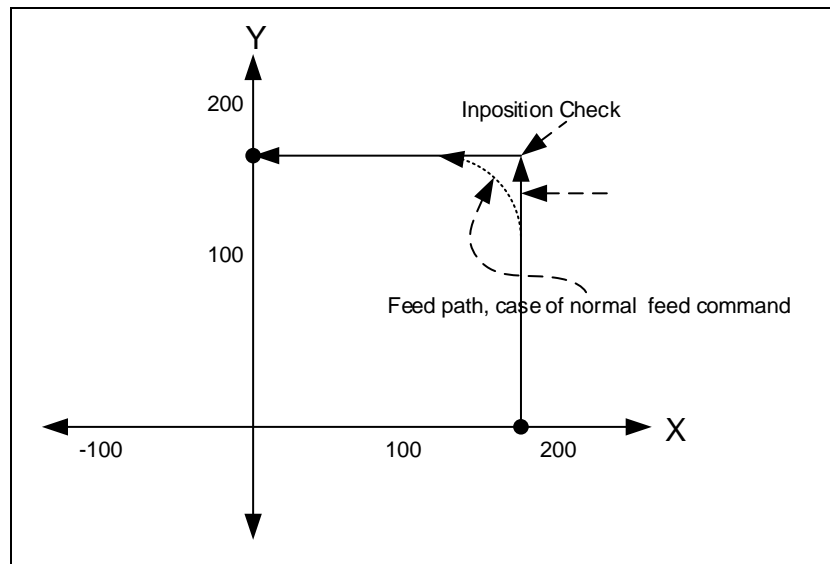
NC 参数	组名和参数名		
	Group	Name	Channel 1
NC 通道参数	Basic Settings	Target Machining Quantity	0
		Target machining quantity at M99 repetition	0
		Check of decimal point	1: Unused
		Keep workpiece coordinate system	0: Keep
		Macro call on T-code command	0: Do not call
		DWELL Method	0: Time
		Block selection at NC reset	0: Keep the Current Block
		Statement number search	0: Search
		Minimum command unit	0 mm
		Whether to use G22 [No traveling area]	0: Used
		Inner/Outer side of G22 [No traveling area]	0: Inner side
		Whether to use the 3rd [No traveling area]	0: Used
		Rotary Axis of Cylindrical Interpolation	0: X Axis
		Linear axis for interpolating the polar coordi	0: None
		Rotary axis for interpolating the polar coordi	0: None
Monitoring time for in-position completion	5000 ms		

6) 精准停止 (G09)

G09

G09: 精准停止 (精准停止指令)

在正常进给/切削运行中, 当轴移位加速或者减速时, 由于受物理惯性影响, 转角段使当前块减速并使下个块加速. 这就是‘形成圆’的发生原因. 该功能执行‘到达位置检查’, 并如下以正确进给/切削显示位置, 执行指令块并进入下一个块.



该功能为单次指令，所以仅在对指令中有效。

如果 G09 指令用于例如“G01”的单个进给指令，在移位目标位置执行‘到达位置检查’。

如果继续使用该功能指令例如切削，在弯曲表面的连接交叉点执行准确的停止，导致一些问题；机械表面不良状态，工具较大磨损，和较长的加工时间。

```
G90
G00 X0. Y0. Z0
G09 G01 X100. Y100. F5000          % 通过精准停止的线性进给
X200. Y250.                        % 线性进给
G10
```

以上程序为使用精准停止(G09)指令用于线性进给的实例。以上程序中的 G09 指令为单次指令，所以"X200. Y250."指令不受 G09 指令影响。

7) 选择圆弧插值的平面(G17, G18, G19)

```
(G90, G91) G17 (G02, G03) X_ Y_ (I_ J_ / R_) F_
(G90, G91) G18 (G02, G03) X_ Z_ (I_ K_ / R_) F_
(G90, G91) G19 (G02, G03) Y_ Z_ (J_ K_ / R_) F
```

G90, G91: 绝对/增量指令

G17: X-Y 平面

G18: Z-X 平面

G19: Y-Z 平面

G02, G03: 顺时针，计数器顺时针圆弧插值

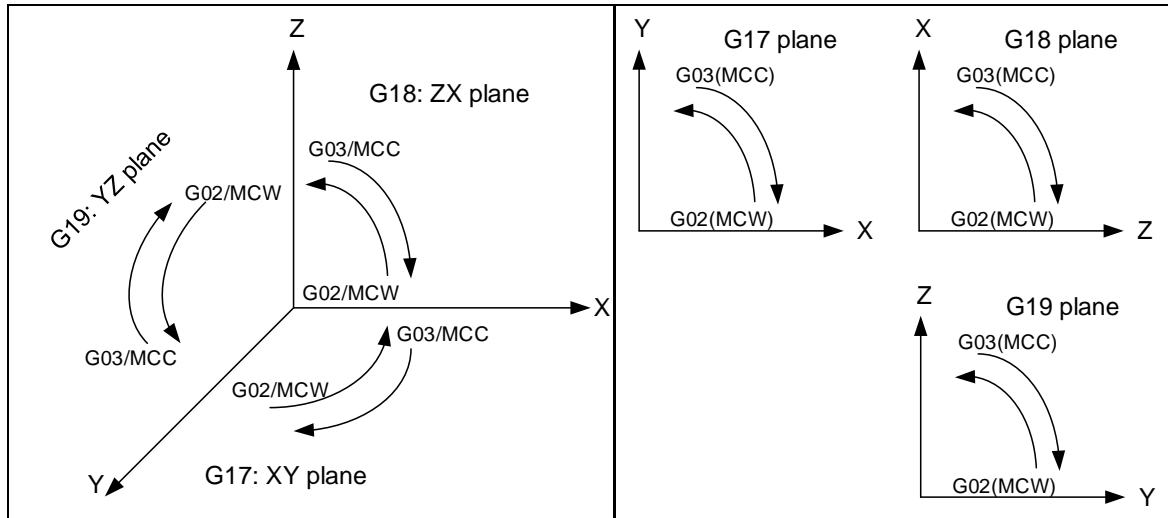
X_ Y_ Z_: 移位目标位置

I_ J_ K_/R_: 圆弧半径参考点

F_: 进给率

第 9 章 NC 控制功能

该指令指定执行圆弧插值的两个平面。



选择圆弧插值平面指令的相关参数如下。

NC 参数	组名和参数名		
	Group	Name	Channel 1
NC 通道参数		Default modal G-code for TRAVERSE	0: G00
	Default Settings	Default modal G-code for PLANE	0: G17
		Default modal G-code for ABS/INC	0: G90
		Default modal G-code for Limit check	0: G22

8) 米制单位输入(G21)

G21

G21: 米制单位输入

该指令设置以后输入的位置单位为米制。

由于当前运动控制器仅支持公制单位，如果不使用 G21，则默认位置单位为米制。

9) 使能/禁止行程功能(G22, G23)

G22 (X_ Y_ Z_) (I_ J_ K_)
G23

G22: 行程查看功能 On

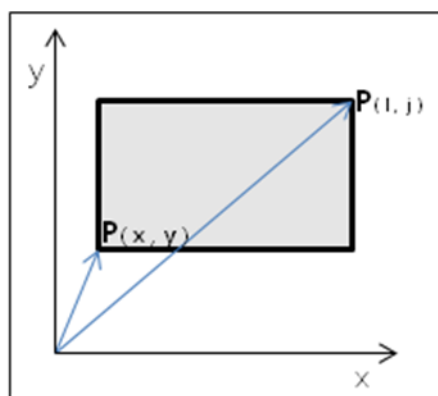
G23: 行程查看功能 Off

X_ Y_ Z_: 基于每个坐标的机械原点输入位置下限.

I_ J_ K_: 基于每个坐标的机械原点输入位置上限.

该指令设置坐标系软件限制. 可以输入每个行程的下限坐标(X, Y, Z)和上限坐标(I, J, K). 如果超出该范围, 则出现错误. 然后, 在手动模式中, 可以输入工作区域或者在关闭行程查看功能后进行驱动. 这是一个模式指令, 所以一旦执行则持续有效.

对于不同于 X, Y 和 Z 轴的 A, B, C, U, V, W, S 轴, 应通过参数进行设置. 此时, 参数‘是否使用 G22 无行程区域’应设置为 1, 表示使用 G22 指令.



第 9 章 NC 控制功能

相对于使能/禁止行程的参数如下。

NC 参数	组名和参数名		
	Group	Name	Channel 1
NC 通道参数	Basic Settings	Target Machining Quantity	0
		Target machining quantity at M99 repetition	0
		Check of decimal point	1: Unused
		Keep workpiece coordinate system	0: Keep
		Macro call on T-code command	0: Do not call
		DWELL Method	0: Time
		Block selection at NC reset	0: Keep the Current Block
		Statement number search	0: Search
		Minimum command unit	0 mm
		Whether to use G22 [No traveling area]	0: Used
		Inner/Outer side of G22 [No traveling area]	0: Inner side
		Whether to use the 3rd [No traveling area]	0: Used
		Rotary Axis of Cylindrical Interpolation	0: X Axis
		Linear axis for interpolating the polar coordi	0: None
Rotary axis for interpolating the polar coordi	0: None		
Monitoring time for in-position completion	5000 ms		

10) 原点返回检查 (G27)

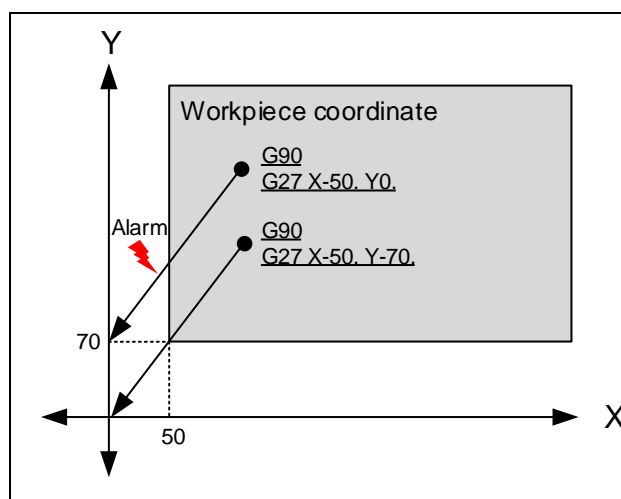
(G90, G91) G27 X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_

G90, G91: 绝对/增量指令

G27: 原点返回检查

X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_: 目标坐标

通过该指令，移位到指定 X, Y, Z 坐标。当移位完成后当前位置为原点时，原点返回完成。如果不是原点，则发生警报。当执行指令时，工具直径和工具长度补偿取消。



11) 自动原点返回 (G28)

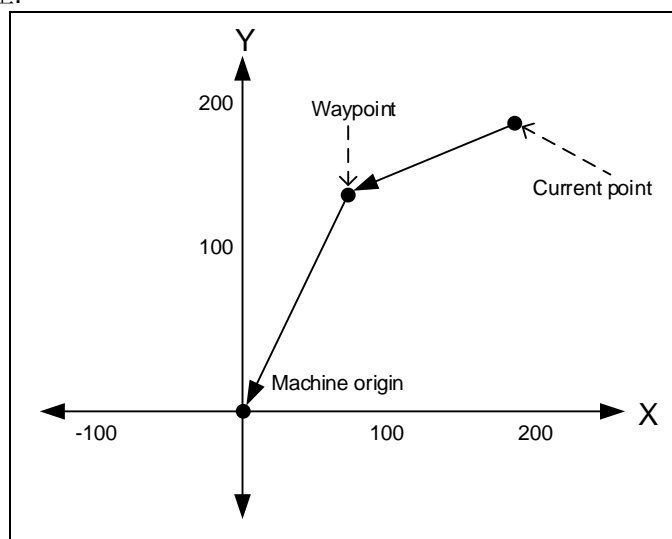
```
(G90, G91) G28 X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_
```

G90, G91: 绝对/增量指令

G28: 自动原点返回指令

X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_: 每轴路径坐标为原点

该指令为自动返回轴到机械参考点。当程序执行期间遇到 G28 指令时，每轴以快速移位比率移动到机械原点。此时，以每轴指令位置方式停止。



不接收自动原点返回指令的轴不移动。

增量指令可用于轴定位。

如果轴位置指令为"0"，在不进行路径点返回的情况下直接返回到机械原点。

第 9 章 NC 控制功能

```
G90
G01 X100. Y100. Z100 F552.% 线性插值, 移位目标位置(X=100, Y=100),速度 552
G28 X40. Y55. Z32.      % 自动原点返回, 路径点(X=40, Y=55, Z=32)
G91 G01 X50. Y50. F550.
```

以上程序为通过 G28 自动原点返回指令线性移动转换为 X, Y, Z 轴的位置到机械原点.

12) 从自动原点返回 (G29)

```
(G90, G91) G29 X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_
```

G90, G91: 绝对/增量指令

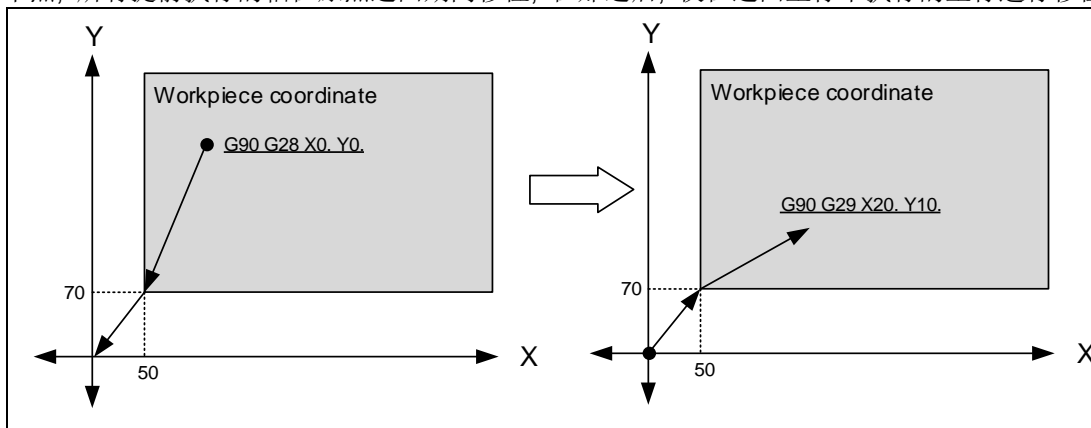
G29: 从原点返回指令

X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_: 返回坐标

当自动原点返回(G28)之后移位完成时使用该指令, 执行 2nd, 3rd, 4th 原点返回(G30). 通过用于原点返回的路径点快速移位(G00)到返回坐标. 如果原点返回指令没有预先执行, 机械原点变为中点并移位到返回坐标. 在该指令中, 不应用工具直径补偿和工具长度补偿.

所有执行的轴在中点移位之前, 原点返回时已经进行移位, 仅在返回坐标中已执行的坐标进行移位.

对于中点, 所有提前执行的轴在自动原点返回期间移位, 在那之后, 仅在返回坐标中执行的坐标进行移位.



13) 2nd, 3rd, 4th 原点返回(G30)

```
(G90, G91) G30 (P2, P3, P4) X_ Y_ Z_ U_
```

G90, G91: 绝对/增量指令

G30: 自动原点返回指令

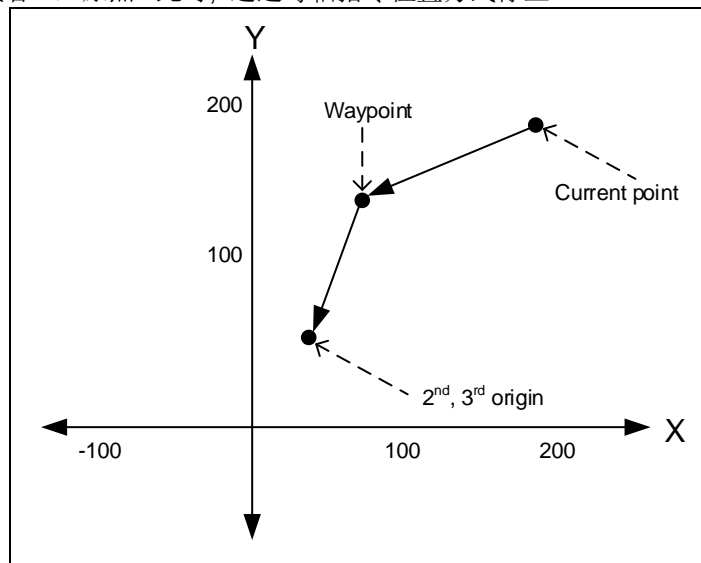
P2: 2nd 原点

P3: 3rd 原点

P4: 4th 原点

X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_: 每个轴的路径点坐标返回原点

该指令自动返回每个指令轴以配置 2nd, 3rd, 4th 原点. 如果在程序执行期间遇到 G30 指令, 每个轴以快速移位率移动到指定 2nd 或者 3rd 或者 4th 原点. 此时, 通过每轴指令位置方式停止.



每轴 2nd, 3rd, 4th 原点坐标应在 NC 通道/轴参数中分别指定.

通过 P2, P3 和 P4 指令, 对于当前执行的 2nd 和 3rd 原点返回, 可以指定 2nd, 3rd 原点之间的原点."P2"参考 2nd 原点, "P3"参考 3rd 原点.

增量指令可用于轴定位.

NC 参数	组名和参数名		
	Group	Name	X Axis
NC 通道/轴参数	Home Settings	Position of 2nd home	0 mm
		Position of 3rd home	0 mm
		Position of 4rd home	0 mm

第 9 章 NC 控制功能

14) 禁止工具直径补偿(G40)

{G40} [G00/G01] X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_

G40: 禁止工具直径补偿

X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_: 执行下一个功能块矢量

G40 指令用于取消工具直径补偿。当 G40 在 G00 和 G01 模式中执行时，模式将从使能工具直径补偿变为禁止工具直径补偿。G40 模式中的补偿始终为 0，工具中心路径与程序路径匹配。程序应在 G40 模式中终止。如果在 G41 / G42 模式中终止，程序将通过补偿数量在一定距离补偿中终止。另外，不可以在圆弧插值(G02, G03)中取消工具直径补偿。

G40 X_ Y_

15) 工具直径补偿(G41, G42)

{G41/G42} [G00 / G01] X_ Y_ Z_ A_ B_ C_ U_ V_ W_ S_ D_

G41: 工具直径左补偿

G42: 工具直径右补偿

D_: 保存工具直径补偿值的补偿数量

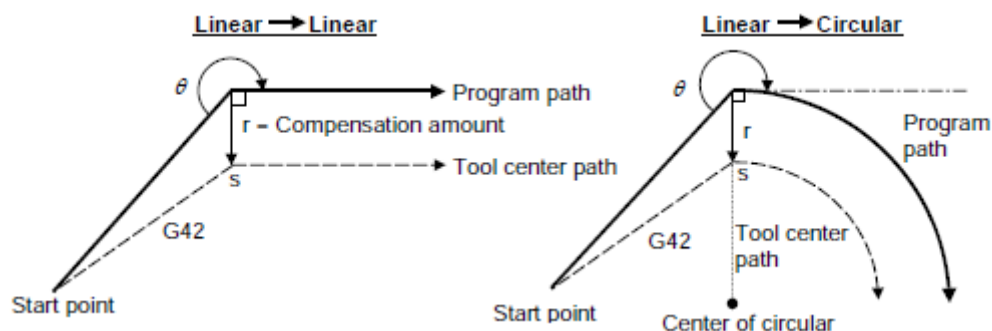
X_ Y_ Z_ U_: 执行下一个指令块的矢量

(1) 启动模式

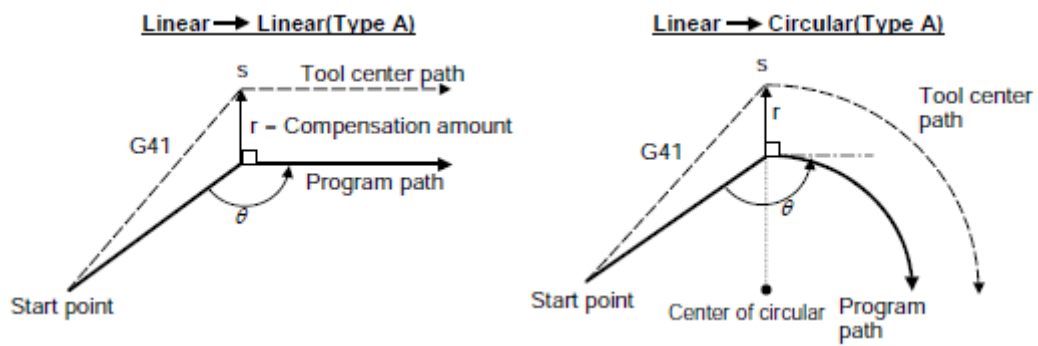
当工具直径补偿在禁止工具直径补偿中通过 G41 / G42 指令启动时，则调用启动模式。

G41 / G42 执行或者轴运动块首次执行的情况下，调用启动模式。

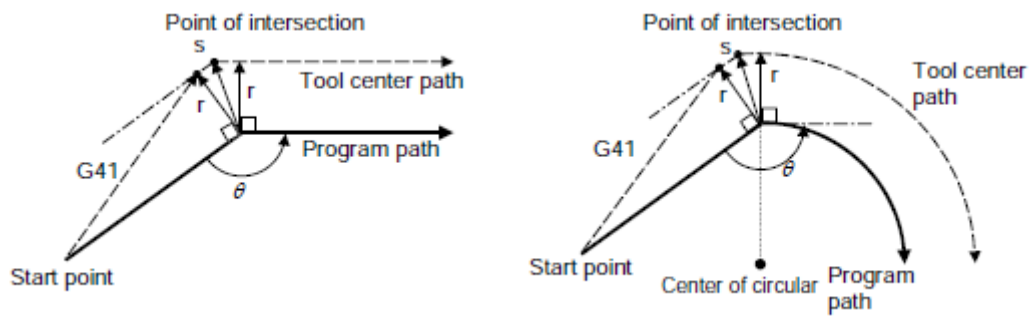
在启动块中，轴运动指令必须大于工具半径。在启动或者取消模式中，圆弧指令[G02 / G03]不执行。当执行该指令时，发生警报。



[内角工具路径]

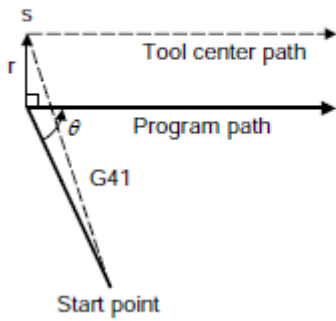


[外角工具路径(钝角)] (类型 A)

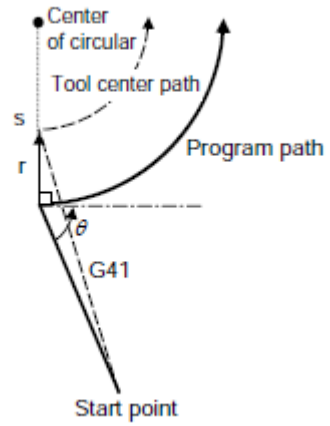


[外角工具路径(钝角)] (类型 B)

Linear → Linear(Type A)

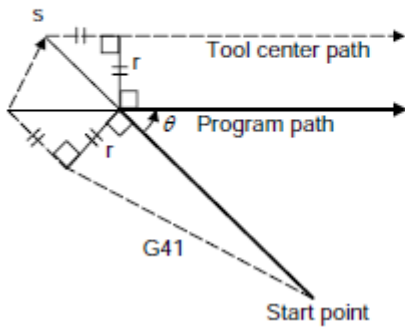


Linear → Circular(Type A)

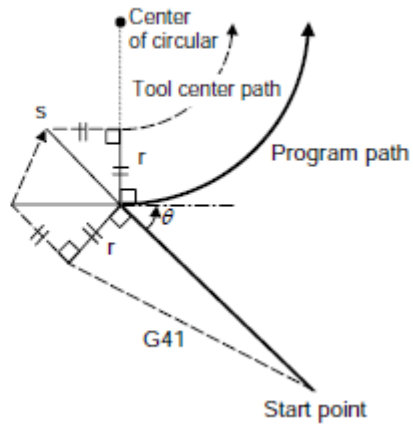


[外角工具路径(锐角)] (类型 A)

Linear → Linear(Type B)

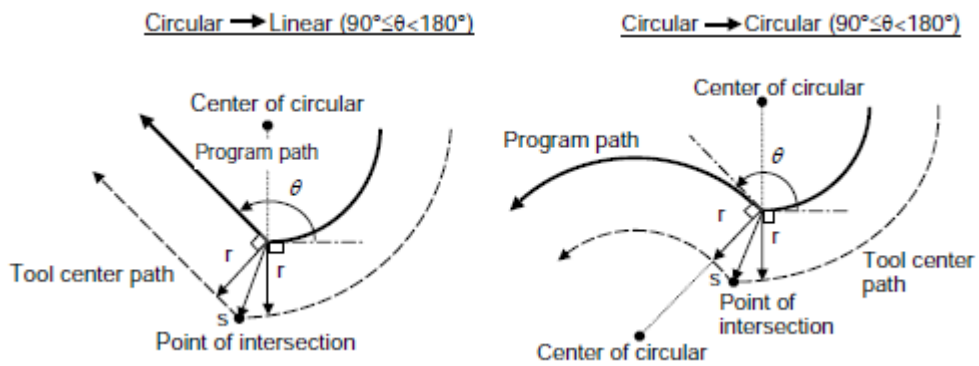
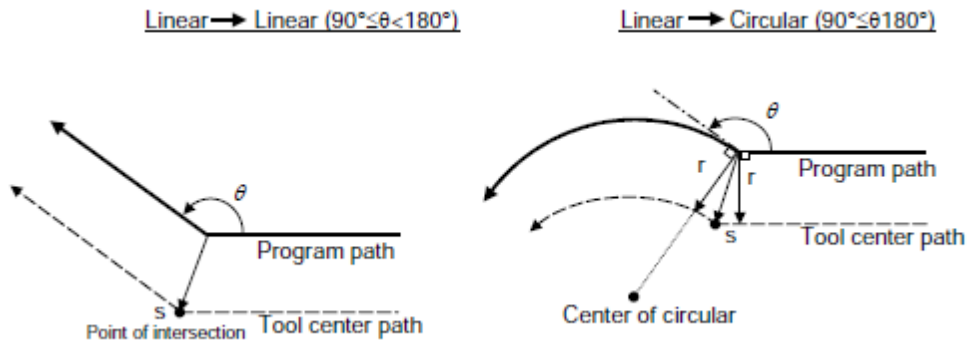


Linear → Circular(Type B)

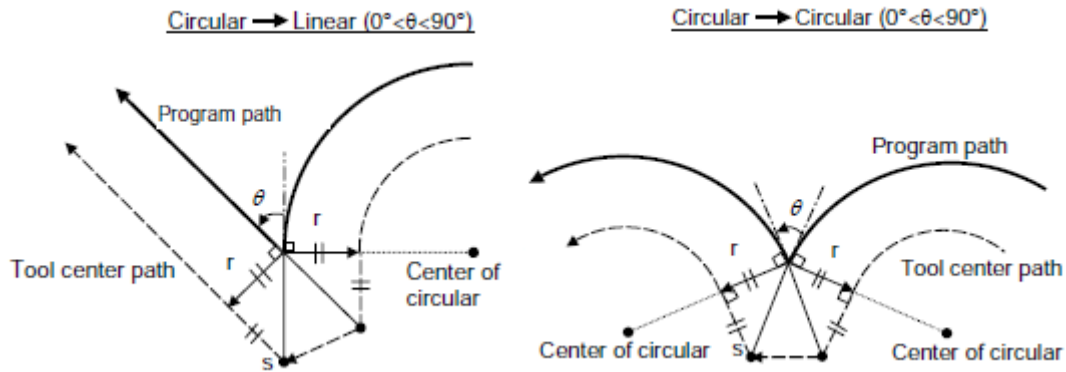


[外角工具路径(锐角)] (类型 B)

(2) 补偿模式

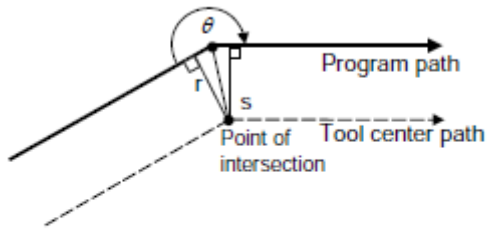


[外部加工(钝角)]

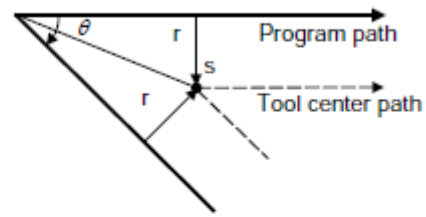


[外部加工(锐角)]

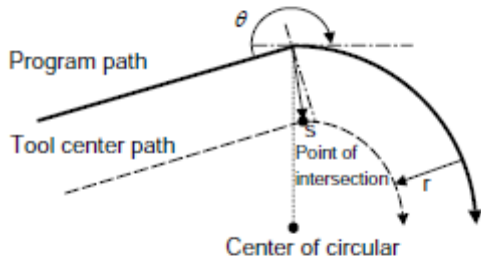
Linear → Linear (Obtuse angle)



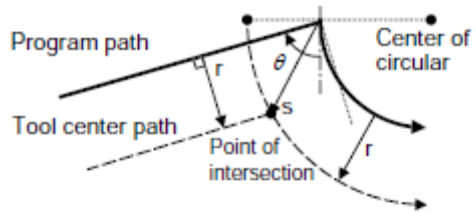
Linear → Linear (Acute angle)



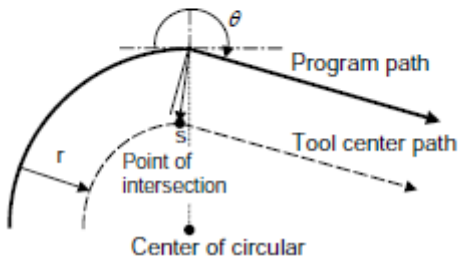
Linear → Circular (Obtuse angle)



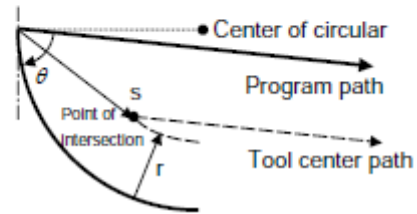
Linear → Circular (Acute angle)



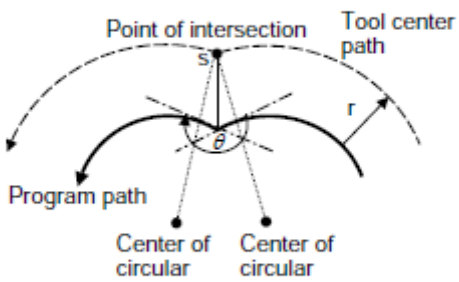
Circular → Linear (Obtuse angle)



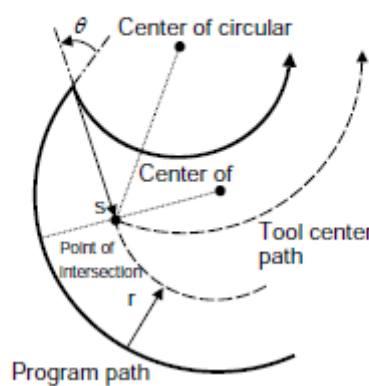
Circular → Linear (Acute angle)



Circular → Linear (Obtuse angle)

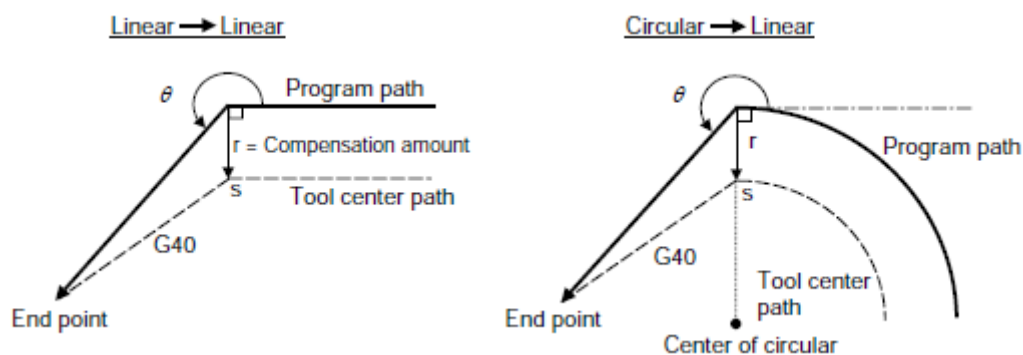


Circular → Linear (Acute angle)

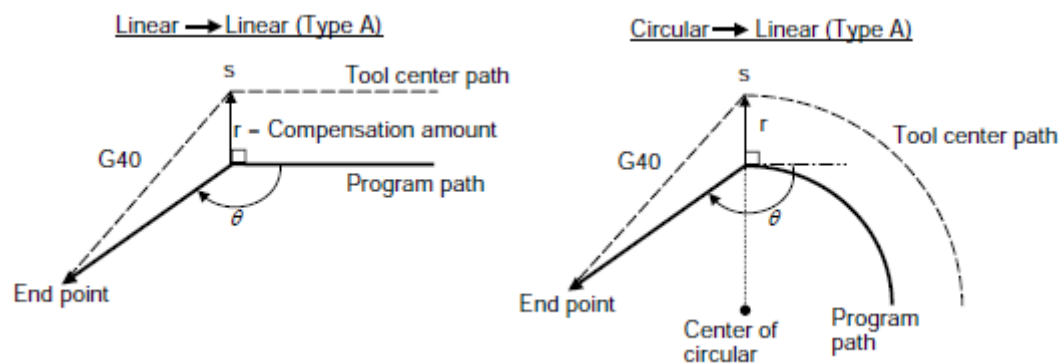


[内部加工]

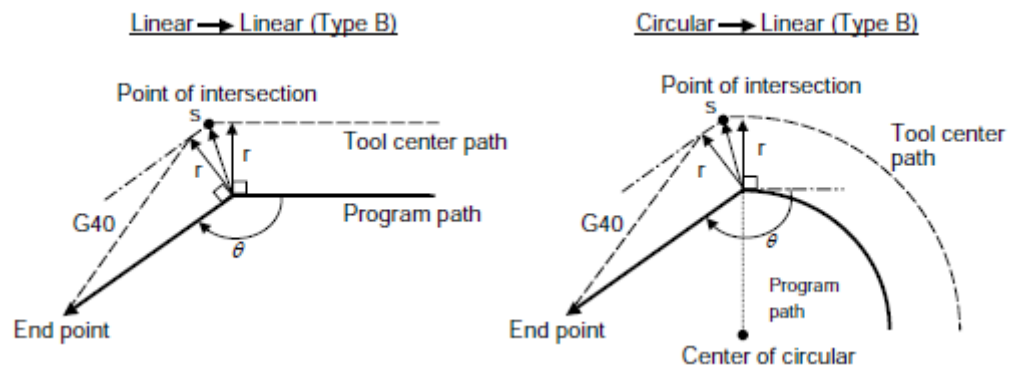
(3) 取消模式



[内角工具路径]

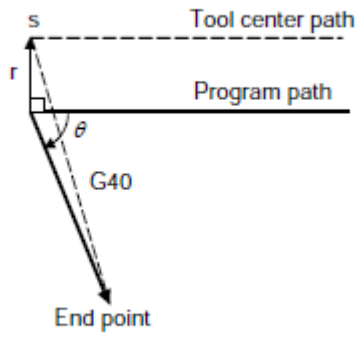


[外角工具路径(钝角)] (类型 A)

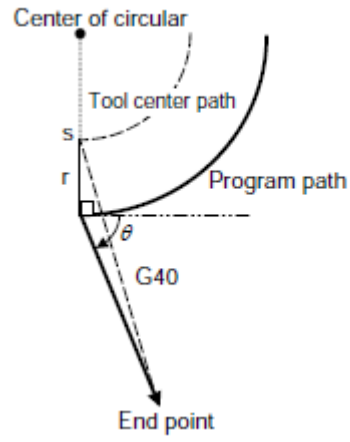


[外角工具路径(钝角)] (类型 B)

Linear → Linear (Type A)

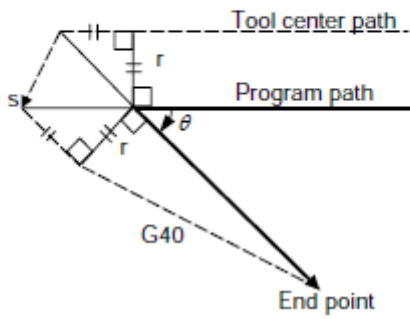


Circular → Linear (Type A)

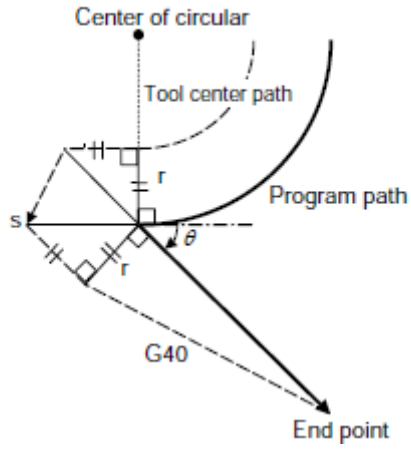


[外角工具路径(锐角)] (类型 A)

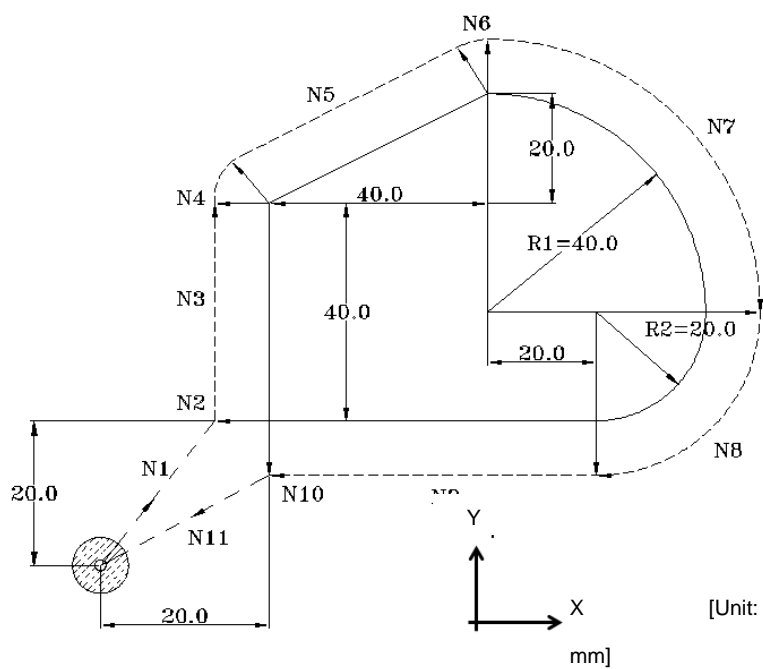
Linear → Linear (Type B)



Circular → Linear (Type B)



[外角工具路径(锐角)] (类型 B)



N1 G91 G17 G00 G41 X20. Y20. D08 (D08 工具补偿数量)
 N2 G01 Z-25. F100 (以对应数量输入工具半径值)
 N3 Y40. F250
 N4 G39 X40. Y20. (圆弧类型补偿路径)
 N5 X40 Y20.
 N6 G39 X40. (圆弧类型补偿路径)
 N7 G02 X40. Y-40. R40.0
 N8 X-20. Y-20. R20
 N9 G01 X-60.
 N10 G00 Z25.
 N11 G40 X-20. Y-20.
 N12 M30

第 9 章 NC 控制功能

关于工具直径校正的参数如下.

NC 参数	组名和参数名		
	Group	Name	Channel 1
NC 通道参数		How to Apply the Compensation Value of th	0: Apply the diameter value
		Compensation Type of the Tool Diameter	0: Bypass Traverse
		Whether to check the tool interference duri	0: Do not check
		Compensation amount of the tool diameter	0 mm
		Compensation amount of the tool diameter	0 mm
		Compensation amount of the tool diameter	0 mm
		Compensation amount of the tool diameter	0 mm
		Compensation amount of the tool diameter	0 mm
		Compensation amount of the tool diameter	0 mm
		Compensation amount of the tool diameter	0 mm
		Compensation amount of the tool diameter	0 mm
		Compensation amount of the tool diameter	0 mm
		Compensation amount of the tool diameter	0 mm
		Compensation amount of the tool diameter	0 mm

* 对应工具直径补偿数量的参数为"工具直径补偿数量 1 ~工具直径补偿数量 128".

16) 工具长度补偿(G43, G49)

G43 Z_ H_
G49 Z_

G43: 工具长度 + 长度补偿

G49: 取消工具长度补偿

Z_: Z-轴移动指令(G17 平面的情况)

可用绝对指令和增量指令.

H_: 补偿数量保存工具长度补偿值

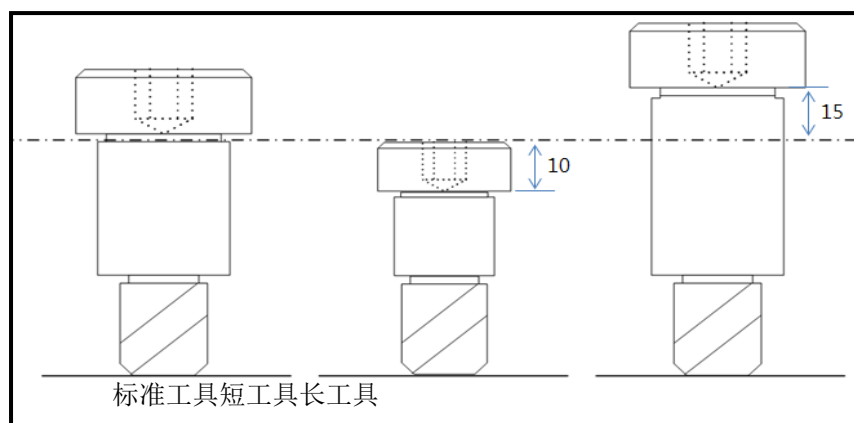
当通过补偿数量执行时, 在 Z 轴移动指令结束点的坐标值保存工具长度补偿的 H 代码通过增量和绝对指令执行, 对于 G43, 增加结果坐标值变为结束点. 长度值可以增加+, -符号.

如果忽略 Z-轴移动指令, G43 在块中通过下一个 Z 轴指令的+方向应用长度补偿.

以下方式应用于首次测量工具长度。

- (1) 将工件放在一个较宽表面的桌面上。
- (2) 将参考工具的尾部与工件平面接触。
- (3) 补偿 Z-轴值。
- (4) 替换测量工具，将工具顶部与平面接触。
- (5) 该状态的相关坐标系统 Z-轴值保存在存储中作为工具补偿数量。

通过以上设置，较短工具的修正数量设置为-值，较长工具的修正数量设置为+值。因此，在编程期间，工具长度补偿始终以 G43 进行指定。



G43 和 G49 功能为模式指令，所以一旦执行则持续有效。因此，如果一旦工具变更后 G43 执行，然后 G49 在工具任务完成后并且在工具变更前执行，工具长度补偿取消。

警告 1. 为了取消偏移补偿，执行 G49 或者 H00。

警告 2. 补偿数量可以指定到 H00 - H128 中，补偿数量 00，即，对应于 H00 的补偿数量始终为 0，并且不可以设置对应于 H00 的补偿数量。

警告 3. 建议创建一个用于使能/禁止工具长度补偿指令的例如 Z 轴移动指令的程序。原因是如果以 G43 H01 相同方式执行的话，通过长度补偿地址 01 中的工具长度(或者长度补偿数量)输入进行移动，并且如果仅执行 G49，在执行 G49 之前通过工具长度补偿以相反方向移动，如果工具长度补偿为"+", 通过工具长度从当前位置向下移动，可能导致工具冲突。

因此，建议执行例如 Z 轴移动指令的使能/禁止工具长度补偿指令，使其大于工具长度值。

第 9 章 NC 控制功能

对应于工具长度补偿的参数如下。

NC 参数	组名和参数名		
	Group	Name	Channel 1
NC 通道参数	Tool Length Compensation	Compensation amount of the tool length 62	0 mm
		Compensation amount of the tool length 63	0 mm
		Compensation amount of the tool length 64	0 mm
		Compensation amount of the tool length 65	0 mm
		Compensation amount of the tool length 66	0 mm

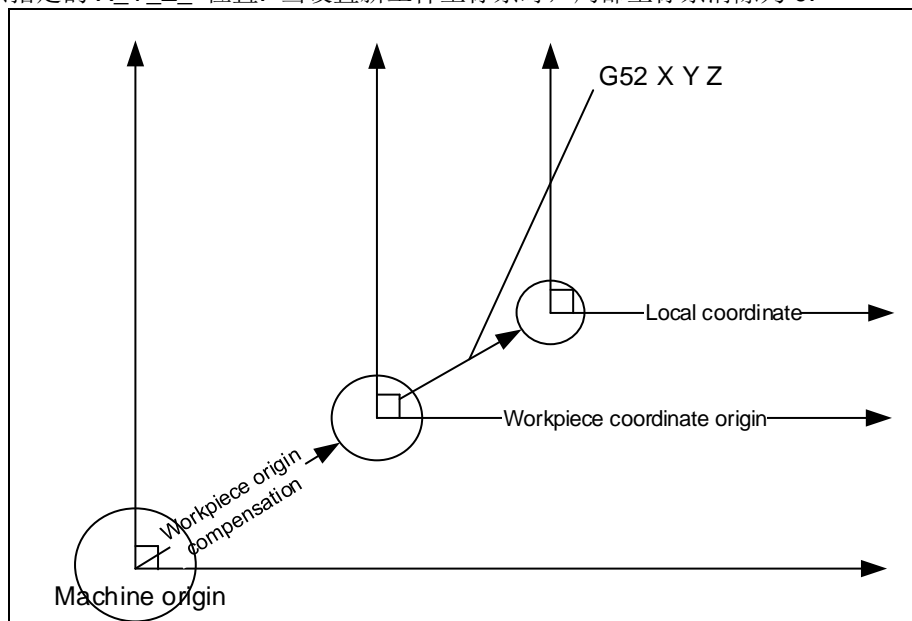
* 相对于工具长度补偿数量的参数为"工具长度补偿数量 1 ~ 长度补偿数量 128".

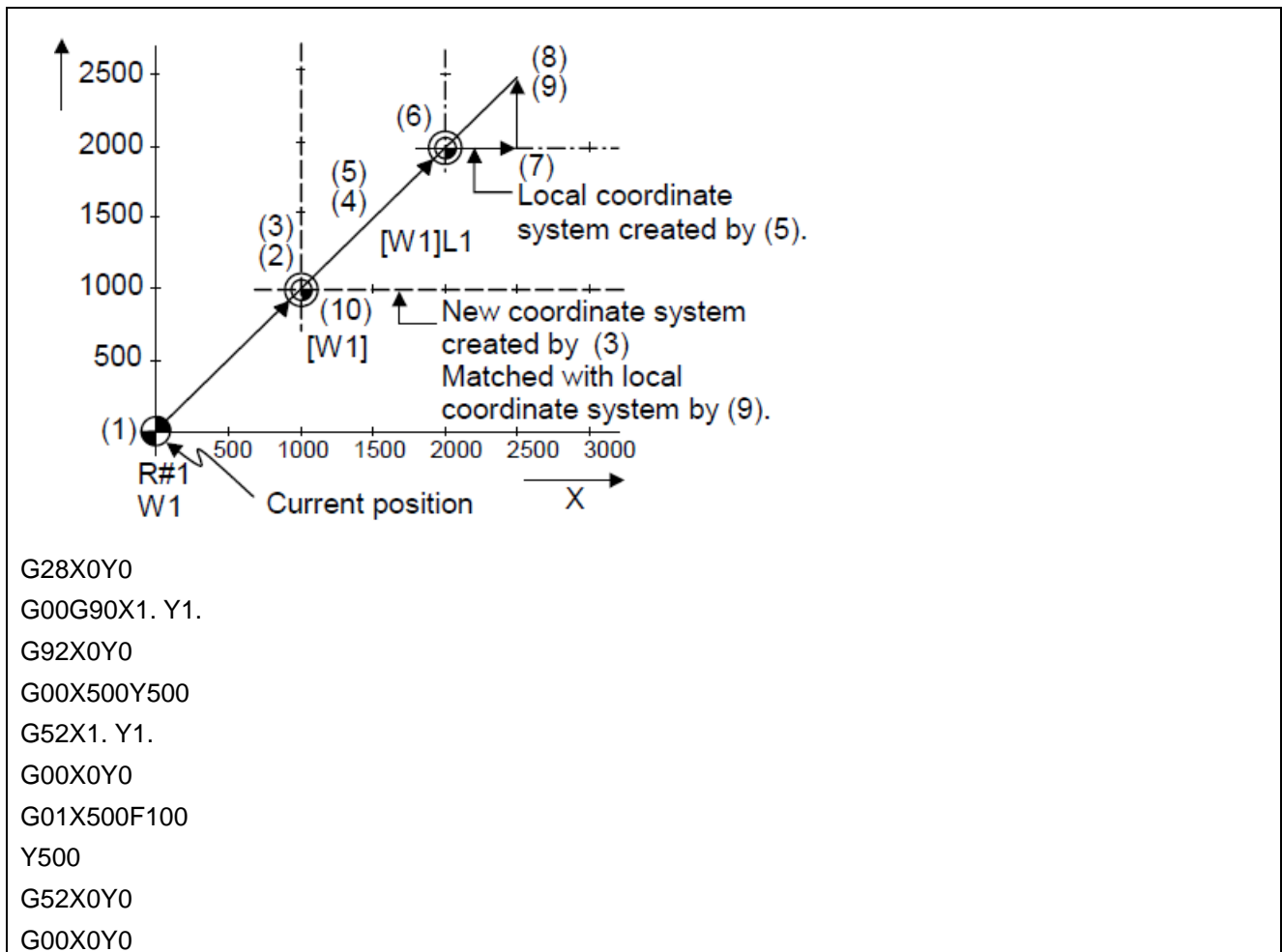
17) 局部坐标系设置(G52)

G52 X_Y_Z_

G52: 局部坐标系设置

当通过工件坐标系创建程序时，局部坐标系是设置和使用基于设置工件坐标系的任意点中的参考点的坐标系。通过局部坐标系指令，新坐标系，即，局部坐标系可以在所有工件坐标系(G54 到 G59)中设置。每个坐标系的原点为每个工具坐标系指定的 X_Y_Z_ 位置。当设置新工件坐标系时，局部坐标系清除为 0。





18) 选择机械坐标系 (G53)

```
G90 G53 X_ Y_ Z_
```

G90: 绝对指令

G53: 选择机械坐标系

X_ Y_ Z_: 馈送位置

G53 为使用机械坐标系的指令，工具快速移动到以上机械坐标系的 X_Y_Z_位置。**G53** 为单独 G 代码，所以仅在执行块中有效。在绝对指令(G90)中仍然有效，但是在增量指令(G91)中无效。如果想要移动工具到机械指定位置，例如工具变更位置，应该在机械坐标系中通过 **G53** 编程。工具直径补偿，工具长度补偿，和工具位置补偿必须在 **G53** 指令前取消，否则，将移动到补偿状态。另外，由于机械坐标系必须在 **G53** 执行前设置，手动原点返回或者通过 **G28** 原点返回必须在电源导通后执行。

第 9 章 NC 控制功能

```
G40 G80
G53 G90 X-140 Y-120 Z0    (移动到机械坐标系的 X-140 Y-120 Z0 位置)
G92 X0 Y0 Z150           (通过变更工件坐标系停止)
G30 G91 Z0
G54 G00 G90 X0 Y0
M30
```

19) 选择工件坐标系 1~6 (G54, G55, G56, G57, G58, G59)

```
G54 X_ Y_ Z_
G55 X_ Y_ Z_
G56 X_ Y_ Z_
G57 X_ Y_ Z_
G58 X_ Y_ Z_
G59 X_ Y_ Z_
```

G54: 选择工件坐标系 1

G55: 选择工件坐标系 2

G56: 选择工件坐标系 3

G57: 选择工件坐标系 4

G58: 选择工件坐标系 5

G59: 选择工件坐标系 6

X_ Y_ Z_: 工具坐标系位置

用于工件设备的坐标系被称为工件坐标系。这个坐标系允许操作者在图纸基础上创建程序，通过应用 NC 程序设置工件任何点作为机械原点。在导通电源后，必须执行用于坐标系合理应用的原点返回。当使用 G54 ~ G59 时，不需要通过 G92 设置坐标系。

```
G40 G80
G28 G91 X0 Y0 Z0    (返回到路径点为当前位置值的机械原点[G91 模式])
G54 G00 G90 X0 Y0 Z0 % 使用 54 工件坐标系和快速移位到原点。即，G54
                    % 快速移位到坐标系原点
M30
```

20) 单向定位(G60)

G90 G60 X_ Y_ Z_ U_

G60: 单向定位指令

G00: 定位指令

X_ Y_ Z_ U_: 移位目标位置

单向定位(G60)是用于工具移位的功能，替换快速移位或者运行持续。分别由用于执行定位方向的过度运行行程设置进行停止后，移动到结束位置并获得间隙补偿。

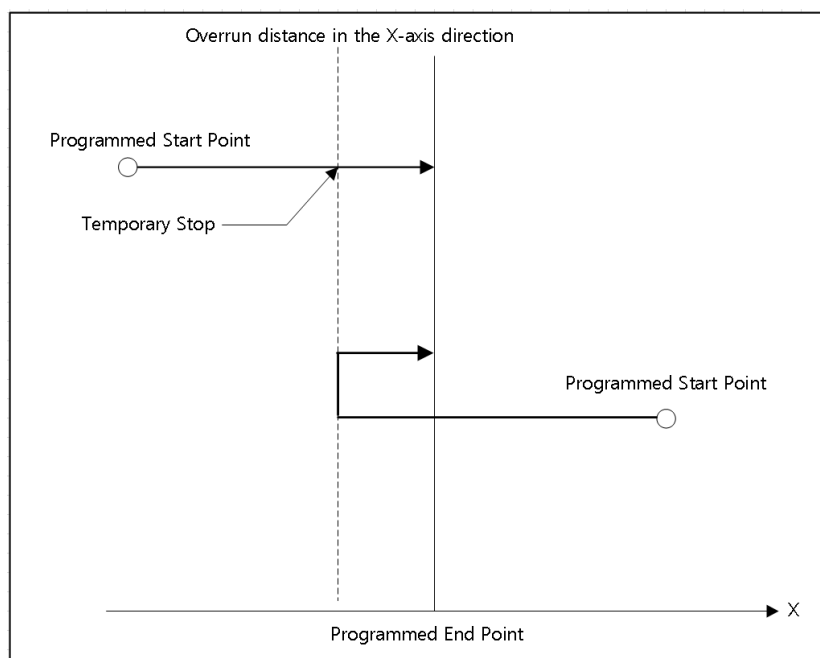
因此，应用 G60 指令，始终以相同方向移动到目标位置。

过度运行数量保存在参数中。

请参考 NC 通道/轴参数单方向定位过度运行馈送数量。

NC 参数	组名和参数名		
NC 通道/轴参数	Group	Name	X Axis
	Auxiliary Function	Overrun distance in single dir. positioning	0 mm

注意单方向定位指令无法消除物理间隙。

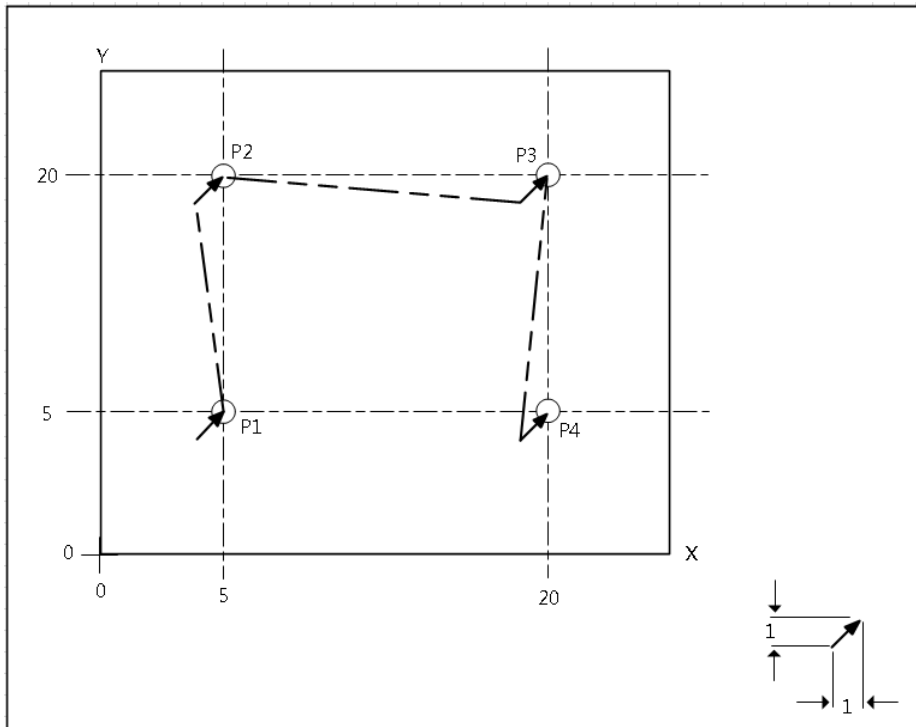


第 9 章 NC 控制功能

如果定位方向指定为如图中显示的 X + 方向, 将始终以相同方向移动到目标位置.

如果不设置过度运行数量或者馈送数量为 0, 不应用单方向定位指令.

另外, 钻孔循环中无法应用到 Z 轴, 并且不受用于设置方向的镜像的影响.



```
G21
G17 G40 G80 T01
M06
G90 G54 G60 X5.0 Y5.0
S1200 M03 T02
G43 Z2.5 H01 M08
G99 G82 R2.5 Z-2.0 P200
F150.0
G60 Y20.0
G60 X20.0
G60 Y5.0
G80 Z2.5 M09
G28 Z2.5 M05
M01
```

21) 绝对指令 (G90)

```
G90 G01 X_ Y_ Z_ A_ B_ C_ U_ V_ W_
```

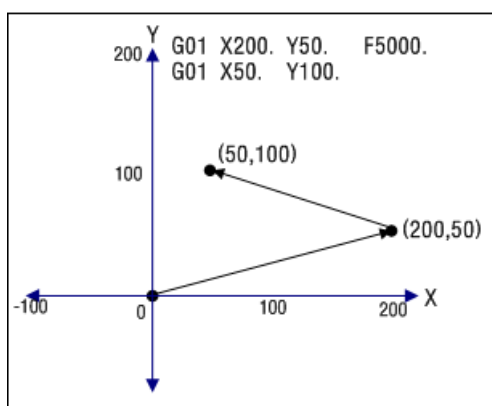
G90: 绝对指令

G01/G00: 线性插值/定位

X_ Y_ Z_ A_ B_ C_ U_ V_ W_: 移位目标位置

绝对指令(G90)是基于当前设置坐标系的执行馈送位置的方式.

馈送结束点使用从当前指定坐标系原点中的计算值, 忽略坐标中的当前位置. 绝对指令(G90)为模式指令, 一旦执行, 保持有效执行直到设置另一个馈送目标的位置指令.



G90	% 绝对指令
G01 X200 Y50 F5000	% 线性插值, 移位目标位置(X=200, Y=50), 速度 5000
X50 Y100	% 线性插值, 移位目标位置(X=50, Y=100), 速度 5000.

以上程序代表上图的根据 G 代码移动.

第 9 章 NC 控制功能

22) 增量指令 (G91)

```
G91 G01 X_ Y_ Z_ A_ B_ C_ U_ V_ W_
```

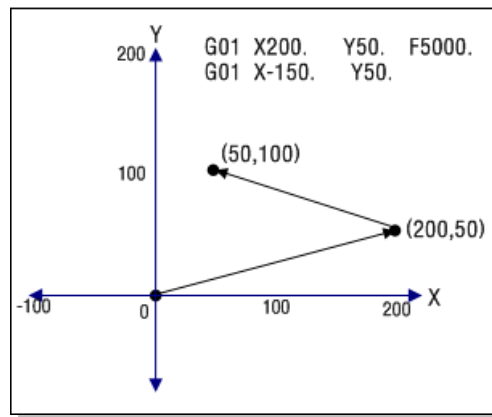
G91: 增量指令

G01/G00: 线性插值/定位

X_ Y_ Z_ A_ B_ C_ U_ V_ W_: 移位目标位置

增量指令(G91)是基于当前设置坐标系执行到目标位置的移动数量的移位方式.

增量指令(G91)是模式指令,一旦执行,保持有效执行直到设置另一个馈送目标的位置指令.



```
G91                % 增量指令  
G01 X200  Y50  F5000.  % 线性插值, 移位目标位置(X=200, Y=50), 速度 5000  
X-150 Y50          %线性插值, 移位目标位置(X=50, Y=100), 速度 5000
```

以上程序代表上图的根据 G 代码移动.

23) 设置工件坐标系, 主轴最大速度(G92)

当设置工件坐标系时

```
G92 X_ Y_ Z_ A_ B_ C_ U_ V_ W_
```

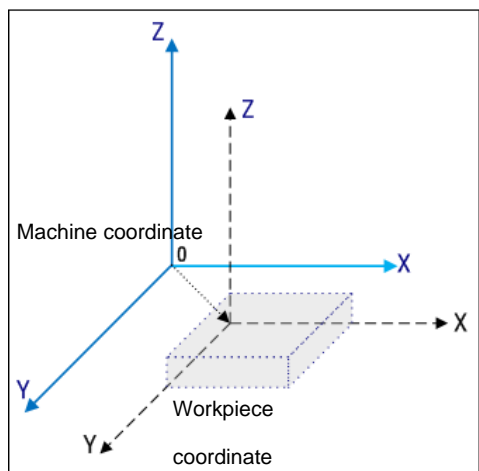
G92: 工件坐标系指令

X_ Y_ Z_ A_ B_ C_ U_ V_ W_: 输入每轴补偿值

工件坐标系设置是通过输入补偿移动当前位置的指令。

用于基于工件参考点的复位坐标系。在应用指令后知道选择工件坐标系，基于移动坐标系(G54 ~ G59)运行。

* 注意：原点返回后应用。



% 原点返回后应用

G90 X10 Y10

% 移动到工件坐标系位置 X=10 Y=10

G92 X100 Y100

% 移动工件坐标系 X=10,Y10 到 X=100,Y=100

G90 X10 Y10

% 移动到工件坐标系位置 -80,-80

24) 每分钟馈送模式指令(G94)

G94 G01 X_ F_

G94: 每分钟馈送模式指令

G01: 线性插值馈送指令

X_: 通过线性插值馈送移动的目标位置坐标值

F_: 速度指令

这是输入单位到每分钟用户输入单位(mm, degree)的指令。

在指令执行情况下，对于 F 输入单位，应用单位/分(mm / min, deg / min)。

G94 G01 X10 F10 % 如果单位是 mm, 推荐馈送率为 10mm / min.

第 9 章 NC 控制功能

25) 每旋转进给模式指令(G95)

```
G95 G01 X_ F_
```

G95: 主轴每旋转进给率

G01: 线性插值进给指令

X_: 通过线性插值进给移动的目标位置坐标值

F_: 速度指令

这是设置主轴每旋转进给率的输入单位的指令。

在此指令下, 对于 F 输入单位, 应用单位/旋转(mm/rev).

```
G95 G01 X10 F10    % 如果单位为 mm, 设置速度到 10mm / rev.
```

26) 圆柱插值模式设置(G107)

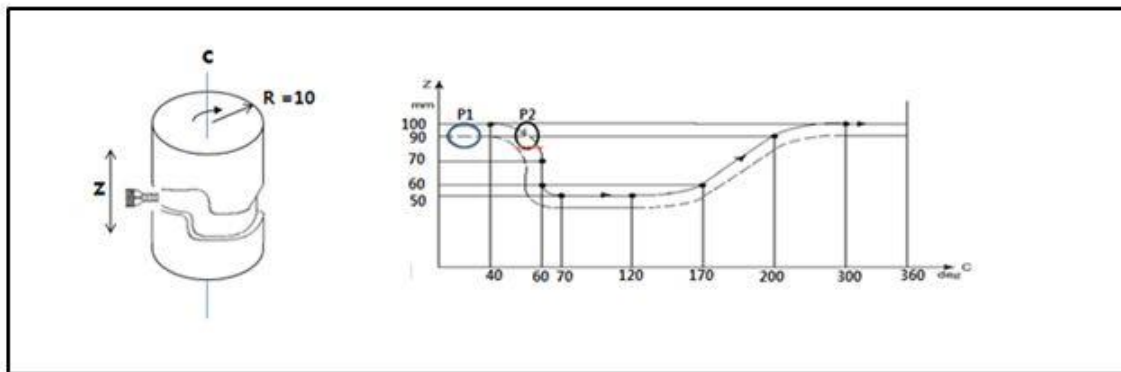
```
G107 C_
```

G107: 圆柱插值指令

C_: 设置圆柱的旋转轴和半径(如果值为 0, 则取消圆柱插值.)

圆柱插值是轮廓控制的一种类型, 是加工一个圆柱表面的控制模式. 当切割圆弧 CAM 时可以很便捷地创建.

通过转换指定旋转轴移动数量执行其他轴的圆弧(G02, G03)和线性插值(G01), 该旋转轴通过圆周线性轴距离的角度进行指定.



```

G107 C10
G90 G01 G18 Z0 C0    % 选择圆弧插值平面 (ZX) Z0 C10 线性插值指令

C40                    % P1:线性插值
G03 Z70 C60 R3        % P2: 圆弧插值 CCW 方向(G03)馈送位置指令(Z70 C60)
                        圆弧半径 (3)

G107 C0                % 取消圆柱插值

```

注意

在圆柱插值模式中，圆弧半径指令仅可以通过 R 执行。

在圆柱插值模式中，不可用定位指令(G00)。

在圆柱插值模式中，不可用坐标系指令。

27) 极坐标插值模式 ON/OFF (G112, G113)

```

G112
G01/G02/G03
G113

```

G112: 设置极坐标插值模式，保持直到执行 G113。

G01/G02/G03: 执行线性和旋转轴直角坐标系。

G113: 禁止极坐标插值模式

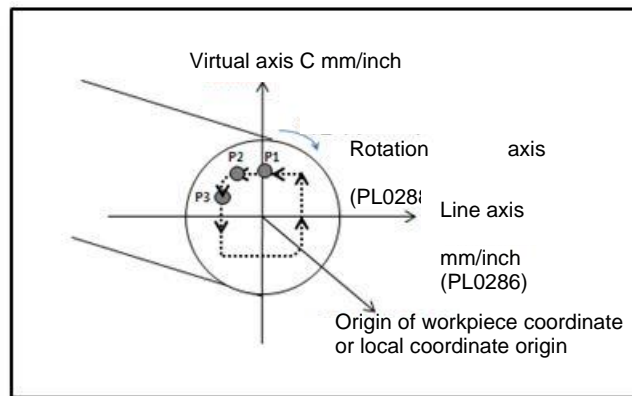
极坐标插值转换输入的指令到直角坐标系到线性轴(X, Y, Z: 工具运动)和旋转轴(A, B, C:工件)的运动。在由线性轴和与线性轴正交的虚轴创建的极坐标插值平面执行。

在极坐标插值前线性和旋转轴应在参数中设置。

第 9 章 NC 控制功能

NC 参数	组名和参数名		
	Group	Name	Channel 1
NC 通道参数	Basic Settings	Target Machining Quantity	0
		Target machining quantity at M99 repetition	0
		Check of decimal point	1: Unused
		Keep workpiece coordinate system	0: Keep
		Macro call on T-code command	0: Do not call
		DWELL Method	0: Time
		Block selection at NC reset	0: Keep the Current Block
		Statement number search	0: Search
		Minimum command unit	0 mm
		Whether to use G22 [No traveling area]	0: Used
		Inner/Outer side of G22 [No traveling area]	0: Inner side
		Whether to use the 3rd [No traveling area]	0: Used
		Rotary Axis of Cylindrical Interpolation	0: X Axis
		Linear axis for interpolating the polar coordi	0: None
		Rotary axis for interpolating the polar coordi	0: None
Monitoring time for in-position completion	5000 ms		

在该模式中，工具直径可以补偿，极坐标插值执行用于工具直径的补偿路径。主要用于 CAM 轴的研磨等。



G112	% 极坐标插值模式 On
G01 C10 F100	% P1: C10 馈送
G01 X-8	% P2: X -8 位置馈送
G03 X-10. C8. R2.	% P3: X -10 Y 8
G113	% 取消极坐标插值模式

注意

- 在极坐标插值中, 仅可以使用直线(G01)和圆弧插值(G02 / G03).
- 虚轴指令单位和线性轴相同. 在 G112 指令下虚轴坐标变为 0.
- 极坐标插值中的 F 指令为线性速率(工件和工具的相对速率).
- 圆弧插值虚构 X 和 Y 平面, 所以从起始点到中心点的距离仅由 I 和 J 执行.

(2) M 代码

通过用于多种机械运行的运动控制要求功能运行机械, 除了例如馈送和使用 G 代码的插值功能. 为了控制机械使用与那些支持 G 代码不同的功能, 运动控制器支持 M 代码.

运动控制器 M 代码支持暂停, 功能指示 NC 程序的结束和与每个运动控制标志位交界的功能.

M 代码仅可以在代码所有运行已经执行完成之后才可以再次执行.

Mxx

Mxx: 辅助指令

对应"Mxx"代码的机械顺序功能已激活.

```
G90
G01 X100. Y100.
G01 X150. Y200
M00                % 暂停指令
G01 X10. Y10.
G01 Y30. Z30.
M02
```

实际运行可能由于机械制造商对机械功能提供的 M 代码不同而不同. 然而, 一般代码表使用如下.

第 9 章 NC 控制功能

M 代码	功能	描述
M00	程序停止	在自动运行期间，当执行 M00 时，自动运行停止。模式信息保持有效直到例如单个模块停止，点击循环启动按钮保持自动运行。
M01	选择停止	当选择停止开关为 On 时，该功能和 M00 功能有效相同。如果开关没有打开则忽略该功能。
M02	程序结束	该功能表示程序结束。在块运行完成后，主轴和冷却剂停止。 然后，光标返回到程序开始。所有指令通过例如 M30 的相同运行而 RESET。
M03	主轴正转[CW]	主轴正转 程序执行前，必须提前调整主轴齿轮移动和旋转次数。
M04	主轴反转[CCW]	主轴反转。 程序执行前，必须提前调整主轴齿轮移动和旋转次数。
M05	主轴停止	主轴停止。 用于变更旋转方向或者移动齿轮。
M06	工具变更	工具变更 根据自动工具变更(ATC)的类型，可以作为指定宏程序调用功能使用。
M08	冷却剂 On	冷却电机运行。 指令执行前，机械控制面板上的冷却剂自动开关必须设置为 On。如果开关为 off，程序不执行。
M09	冷却剂 Off	冷却电机停止。
M30	Tap 结束	在程序结束，程序再次返回到起始，所有指令为 RESET。
M98	辅助程序调用	当自动程序运行时，调用辅助程序的功能
M99	辅助程序结束	该功能为终止辅助程序。 [即使主程序反复执行时使用]

(3) NC 程序的其他运行指令

运动控制器的其他指令为控制程序进度的指令，这些程序不支持 G 代码，M 代码，或者逻辑/数值运行功能。使用变量和指令，可以灵活编程和同步 G/M 代码的复杂格式运行。

这些用于 NC 程序的运行功能与用于运动程序的功能相同，但是由于直接运行于 NC 程序，可以提高程序运行更灵活的机械。运动程序中运行进程项目的不同，当仅通过一个流执行 NC 程序运行时，除了指定为叠加语句，运动程序以固定循环模式运行。

运动控制器支持 NC 程序中作为宏变量的可用变量。宏变量替换变量使用其自身值的部分。通过使用宏变量，可以通过控制器提供机械控制的灵活性。

1) 变量(#)**1-1) 局部变量**

#Ni (N = X, D, W, L; i = 1, 2, 3, ...)

#N[表达式]

关于变量，#跟随一个变量类型和数量。通过#之后的数量分离为多个变量。变量使用的限制如下。

- 可以使用变量替代如下地址值。

例) F[#L103] → F100 (当, #L103 = 100)

Z-[#L110] → Z-250 (当, #L110 = 250).

- 当变量数量适用于变量时

例) #L100 = 105

#L105 = -500

##L100 ; 错误表达式

#[#L100] ; 使用括号表达.

第 9 章 NC 控制功能

-超出设置于每个地址的最大指令值范围的值无法指定.

例) #L140 = 1000

G[#L140] ; 最大指令值 OVER.

- 未定义的变量值为 0

1-2) 全局变量

#MNi (N = X, D, W, L; i= 1, 2, 3, ...)

#MN[表达式]

对于全局变量, #跟随设备类型, 包括变量类型和数量的 M. 每个变量类型的输入值可用范围不同. 使用变量的限制条件与局部变量相同.

1-3) 系统变量

#FNi (N = X, D, W, L; i= 1, 2, 3, ...)

对于系统变量, #跟随设备类型, 包括变量类型和数量的 M. 每个变量类型的输入值可用范围不同. 使用变量限制条件如下.

- 对于系统变量, 只读.

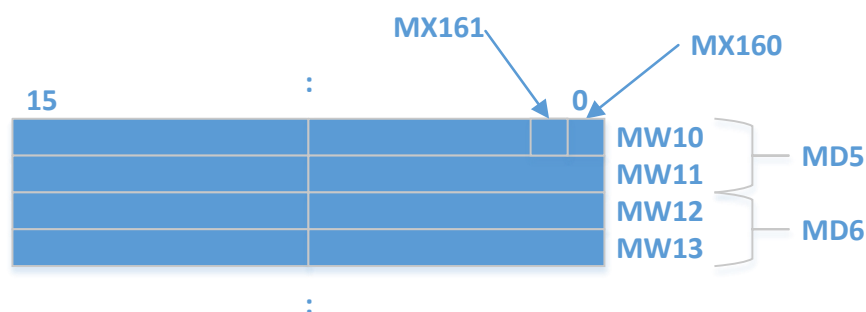
2) 根据变量范围的使用

变量数量限制的变量范围. 根据范围可分为局部变量, 全局变量, 和系统变量, 详情如下

2-1) 每个变量尺寸

类型	尺寸	备注
局部变量	4KB	每个程序尺寸(5 通道 x10 程序)
全局变量(M)	2MB	
系统标志位(F)	128KB	

2-2) 根据每个数据类型的寻址方式



数据类型	寻址方式
Bit (X)	在'X'指示后寻址从位 0 开始的精简位值
16bit (W)	在'W'指示后寻址字单位(16 位)值
32bit (D)	在'D'指示后寻址双字单位(32 位)的值
64bit (L)	在'L'指示后寻址长字单位(64 位)的值

2-3) 执行每个数据类型的地址范围查看

类型	局部变量	全局变量	系统标志位
Bit	#X0 ~ #X32767	#MX0 ~ #MX16777215	#FX0 ~ #FX1048575
16bit	#W0 ~ #W2047	#MW0 ~ #MW1048575	#FW0 ~ #FW65535
32bit	#D0 ~ #D1024	#MD0 ~ #MD524287	#FD0 ~ #FD32767
64bit	#L0 ~ #L511	#ML0 ~ #ML262143	#FL0 ~ #FL16383

3) 指定语句编号(N)

N_

N_:指定语句编号

用于指定语句编号的“N”为用于例如 IF, GOTO, 等指令的命令, 显示对应块以便其他指令可以识别. 由于运动控制器的 NC 程序无法单独记住每个程序块的行数量, “N”指令用于显示块.

第 9 章 NC 控制功能

"N"指令可以和块中的其他指令一起使用，或者在块中单独使用。

然而，用于"N"指令的数字必须是每个程序动作文件(.nc 文件)的唯一数字。如果有一个重复语句号，将发生错误。

通过"N"指令指定的语句号范围从 0 到 2147483647。如果使用超出范围的数字，则将发生错误。另外，一个运动程序中可以指定的语句号数量最多为 1000。如果指定语句号总数超出 1000，则发生错误。

通过"N" 执行的语句号不必要使用程序流水号。即，语句号随机使用。

4) 条件语句 (IF)

```
IF [.....] GOTO N_
```

IF: IF 语句

[.....]: IF 语句条件

GOTO N_: 指定分支语句号

当符合"IF"条件时，条件语句立即分支到指定语句号"N_" 的块。如果不符合条件，则从以下块顺序启动。

对于跟随条件语句"IF"的条件，通过比较局部变量常数，比较局部变量，和比较常数进行创建。

在条件语句"IF"后的条件中，用于条件的变量和常数最多应用 2 个，包含变量和常数。

无法分支到另一个运动文件(.nc file)或者其他运动程序。

5) 分支指令(GOTO)

```
GOTO N_
```

GOTO: 无条件转移到指定为"N_"的块

N_: 指定 Jump 的语句号

分支指令是无条件分支到由"GOTO"之后指定的语句号"N_"标记的块。

不可以分支到另一个运动文件(.nc 文件)或者到另一个运动程序。

6) 重复语句(DO, WHILE)

```
WHILE [<条件表达式>] DO n  
(n = 1, 2, 3, ...)  
~  
END n
```

WHILE: 条件重复语句

DO n: 重复直到 n 重复语句

[.....]: 条件语句

END n: 重复块结束

当符合<条件表达式>时, 从 DO n 到 END n 以下块重复执行. 如果不满足<条件表达式>, 跳到 END n 以下的块.

当[<条件表达式>]可以遗漏, 并且如果已经遗漏, 无限重复从 DO n 到 END n.

当[<条件表达式>] Do n 和 END n 始终成对使用, 通过识别号 n, 成对识别. 如果选择迭代环中的另一个环, 由两个识别因子进行区分.

7) 运行指令

具有变量和整数的替代, 四则基本运算, 数学运算, 等. 对于可用运行. 指令类型如下表显示. 当结合使用多种运行时, 优先级按照变量, 乘法/除法, 加法/减法. 括号"[]"用于设置优先级.

分类	语法	备注
替代	#Li = #Lj	
加法	#Li = #Lj + #Lk	
减法	#Li = #Lj - #Lk	
OR	#Li = #Lj OR #Lk	
XOR	#Li = #Lj XOR #Lk	
乘法	#Li = #Lj * #Lk	
除法	#Li = #Lj / #Lk	
AND	#Li = #Lj AND #Lk	
Sin	#Li = SIN[#Lj]	
Cos	#Li = COS[#Lj]	
Tan	#Li = TAN[#Lj]	
Atan	#Li = ATAN[#Lj]	

分类	语法	备注
Sqrt	#Li = SQRT[#Lj]	
Abs	#Li = ABS[#Lj]	
Round	#Li = ROUND[#Lj]	四舍五入
And	#Li = AND[#Lj]	
Or	#Li = OR[#Lj]	
Fix	#Li = FIX[#Lj]	四舍操作
Fup	#Li = FUP[#Lj]	五入操作

8) 注释 (;, %, ())

%, ;, ()

- % : 注释语句
- ; : 注释语句
- () : 注释语句

指令描述

所有字符, 表达式和跟随"%"或者";"后的数字都添加注释, 在程序执行期间忽略.

"%"仅在一行中有效.

括号("("和")"中的语句也可以视为注释.

G90
G00 X100. Y100. Z100. U100. %从此, 整行为注释.
M02

9) 使用程序运行指令的实例

```

G90

% % 符号注释描述
; 冒号后的注释描述

#L100=1           % 在局部变量# L100 中替代常数 1
#L102=3          %在局部变量# L102 中替代常数 3

IF [#L100 EQ 1] GOTO N3      % 在 IF STATEMENT 中使用局部变量条件比较
#L101 = #L100 + #L102       % 使用局部变量的数值运行
N3                        % 指定语句号
G02 X100. Y100. I50. J50.
G01 X[#L102]              % 与 G01 X3 (由于 3 以# L102 进行替代)的相同运行

N150 WHILE [#L100 LE [360.-#L102]] DO 210      % 重复 N210 直到条件满足

N200 WHILE [#L101 GE 10.] DO 220              % 重复 N220 直到条件满足

G0 Z10.
#L101 = #L101 + 10. (INCREASE) % 等式

END 220          % DO 220 语句的迭代结束

#L100 = #L100 + #L102
#L101 = 50.

END 210          % DO 210 语句的迭代结束

M02

```

9.4NC 参数

NC 参数为通道参数和轴参数.
每个参数如下.

参数	组	No.	项目	描述
1.通道参数	1. 基本设置	1	目标加工数量	设置目标加工数量. (0 ~ 2,147,483,647)
		2	M99 重复加工的目标加工数量	以 M99 设置用于重复加工的目标加工数量.如果设置值与当前加工数量匹配, 循环自动停止. (0 ~ 2,147,483,647)
		3	小数点查看	设置是否查看 NC 程序的小数点. 0: 小数点查看 (如果有小数点则为 Mm, 如果没有小数点则为 um) 1: 小数点不查看(mm)
		4	保持工件坐标系	设置当复位时是否保持工件坐标系. 0: 保持 1: 不保持
		5	当执行 T 代码时是否调用宏	设置当执行 T 代码时是否调用宏程序(9000.nc ~ 9009.nc). 0: 不调用 1: 调用
		6	停留方式	设置对应 X, P 作为时间或者主轴旋转次数的数据停留功能(G04). 如果数据设置为主轴旋转次数, 应用于每旋转馈送状态(G95). 0: 时间 1: 旋转数量

参数	组	No.	项目	描述
1.通道参数	1.基本设置	7	旋转复位的进程块	<p>设置是否复位初始化程序启动块.</p> <p>※如果设置为 0 (保持当前块), "保持工件坐标系"参数应设置为 0 (保持).</p> <p>0: 保持当前块</p> <p>1: 初始化到主程序启动块</p> <p>2: 初始化到主程序当前块</p>
		8	是否搜索语句号	<p>系统中可以保存程序语句号(N__)的缓存数量最多为 1,000.</p> <p>如果程序变更使用 GOTO 语句的顺序, 则需要缓存.</p> <p>如果大于 1,000 块具有 N__指令, 发生警报.</p> <p>该参数用于输入是否执行例如语句号搜索.</p> <p>由于大容量 CAM 程序没有使用语句号的 GOTO, 并且大多数情况下, 有大于 1,000 个语句号, 应设置该参数为 1.</p> <p>0: 搜索</p> <p>1: 不搜索</p>
		12	最小指令单位	<p>当应用小数点查看时, 设置指令值最小单位.</p> <p>(0 ~ 0.999mm)</p>
		18	是否使用 G22 无行程区域	<p>0: '无行程区域'有效.</p> <p>1: '无行程区域'无效.</p>
		19	设置 G22 无行程区域内侧/外侧	<p>0: 内侧</p> <p>1: 外侧</p>

第 9 章 NC 控制功能

参数	组	No.	项目	描述
1.通道参数	1. 基本设置	20	是否使用 3rd ‘无行程区域’	0: ‘无行程区域’有效. 1: ‘无行程区域’无效.
		22	圆柱插值旋转轴	在圆柱插值模式中, 圆弧插值期间轴映射旋转轴.轴为 X, Y, Z, 并通过所选轴的映射旋转轴执行圆弧插值. 例如, 如果映射到 X 周的旋转轴在 XY 平面 (G17)状态下, 宽度变为旋转轴, 高度变为 Y 轴. 当所选 ZX (G18)为平面时, 宽度变为 Z 轴, 高度变为旋转轴. 然而, 如果设置平面为 YZ (G19), 无法在旋转执行轴中执行圆弧插值. 0: X-轴, 1: Y-轴, 2: Z-轴
		23	极坐标插值线性轴	0: 不使用 1: X, 2: Y, 3: Z, 4: A, 5: B, 6: C, 7: U, 8: V, 9: W
		24	极坐标插值旋转轴	0: 不使用 1: X, 2: Y, 3: Z, 4: A, 5: B, 6: C, 7: U, 8: V, 9: W
		33	到达位置完成监控时间	0 ~ 65,535ms

参数	组	No.	项目	描述
1.通道参数	2. 圆弧切削设置	1	当圆形警报发生时, 重置圆形中心	设置当执行 I, J, K 圆弧指令, 起始点和结束点间距离超出两个不同半径间公差时, 在没有发生圆弧报警的情况下是否重置圆弧中心点. 0: 警报发生. 1: 重置圆弧中心点.
		2	圆弧切削 ON/OFF 的速度限制功能	0: 不使用 1: 使用
		3	圆弧半径公差	在圆弧指令下, 设置起始点和结束点的两个不同半径间公差.如果值过大, 圆弧结束点精度降低.当设置为 0 时, 视为 0.001. (0~ 1 单位, 实数)
		5	用于圆弧加工具有速度限制功能的圆弧半径	(0 ~ 10,000 单位, 实数)
		6	圆弧切削的向上切割速度限制	最大限制速度设置与用于圆弧半径下"用于圆弧切削具有速度限制功能的圆弧半径"的设置值. (0 ~ 10,000 单位/min, 实数)
		7	圆弧切削的向下切割速度限制	如果“圆弧切削 ON/OFF 速度限制功能”设置为 ON, 切割速度限制于设置值或者更多. (0 ~ 10,000 单位/min, 实数)
		9	圆弧切削加速	设置圆弧切削加速.
		10	圆弧切削减速	设置圆弧切削减速.
		11	圆弧切削加速度	设置圆弧切削加速度.

第 9 章 NC 控制功能

参数	组	No.	项目	描述
1.通道参数	3. 切割馈送设置	1	设置切割馈送上限限制	如果执行的切割速度超出设置值, 切割速度限制于设置值, 并发生警报. (0 ~ 100,000 单位/min, 实数)
		2	设置切割馈送下限限制	仅当切割速度不以每分钟馈送模式执行时应用. (0 ~ 100,000 单位/min, 实数)
		4	插值运行的加速/减速方式	1: 插值前加速/减速
		7	插值前用于加速/减速的连续块运行方式	当执行连续块时,创建以下个块中的设置速度, 在连接轨迹转角绘制圆弧的连接轨迹.1: 当设置到缓存时, 圆弧不插入. 1: 缓存 2: 混合低 3: 混合前 4: 混合后 5: 混合高
		9	以切割馈送时间加速(插值前)	以切割馈送时间加速
		10	以切割馈送时间减速(插值前)	以切割馈送时间减速
		11	以切割馈送时间加速度(插值前)	以切割馈送时间加速度

参数	组	No.	项目	描述
1.通道参数	8.工具直径补偿	129	如何应用工具直径补偿值	当工具直径补偿时，设置应用工具直径补偿数量的方式。 0: 应用直径值 1: 应用半径值
		130	工具直径补偿类型	在刻度的开始和结尾，工具直径设置行程方式类型。  0: 类型 1(旁路行程) 1: 类型 2(直接行程)
		131	在工具直径补偿期间是否查看工具干扰	设置在工具直径补偿期间是否查看工具干扰 0: 不查看 1: 查看
		1	工具直径 1 补偿数量	用于补偿工具直径的补偿数量 1
	
		128	工具直径 128 补偿数量	用于补偿工具直径的补偿数量 128
		9. 工具长度补偿	1	工具长度补偿数量 1
	
	128	工具长度补偿数量 128	用于补偿工具长度的补偿数量 128	

第 9 章 NC 控制功能

参数	组	No.	项目	描述
1.通道参数	10. 工件坐标系	1	是否使用工件坐标系移动数量.	设置是否使用工件坐标系移动数量. 0: 不使用 1: 使用
		11	工件坐标系移动数量 1	设置 X 轴工件坐标系移动数量.
		设置 7 轴工件坐标系移动数量; Y, Z, A, B, C, U, V.
		19	工件坐标系移动数量 9	设置 W 轴工件坐标系移动数量.
		41	G54 工件坐标系值 1	设置 X 轴工件坐标系.
		设置 7 轴的 G54 工件坐标系值; Y, Z, A, B, C, U, V.
		49	G54 工件坐标系值 9	设置 W 轴 G54 工件坐标系值.
		51	G55 工件坐标系值 1	设置 X 轴 G55 工件坐标系值.
		设置 7 轴的 G55 工件坐标系值; Y, Z, A, B, C, U, V.
		59	G55 工件坐标系值 9	设置 W 轴 G55 工件坐标系值.
		61	G56 工件坐标系值 1	设置 X 轴 G56 工件坐标系值.
		设置 7 轴的 G56 工件坐标系值; Y, Z, A, B, C, U, V
		69	G56 工件坐标系值 9	设置 W 轴 G56 工件坐标系值.
		71	G57 工件坐标系值 1	设置 X 轴 G57 工件坐标系值.
		设置 7 轴的 G57 工件坐标系值; Y, Z, A, B, C, U, V

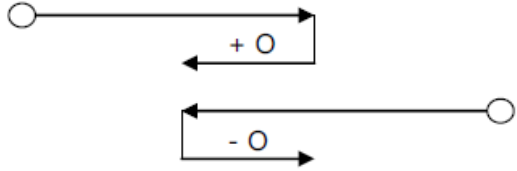
参数	组	No.	项目	描述	
1.通道参数	10. 工件坐标系	79	G57 工件坐标系值 9	设置 W 轴 G57 工件坐标系值.	
		81	G58 工件坐标系值 1	设置 X 轴 G58 工件坐标系值.	
		设置 7 轴的 G58 工件坐标系值; Y, Z, A, B, C, U, V	
		89	G58 工件坐标系值 9	设置 W 轴 G58 工件坐标系值.	
		91	G59 工件坐标系值 1	设置 X 轴 G59 工件坐标系值.	
		设置 7 轴的 G59 工件坐标系值; Y, Z, A, B, C, U, V	
		99	G59 工件坐标系值 9	设置 W 轴 G59 工件坐标系值.	
	11. 宏程序		1	是否应用单个块停止功能到宏程序	设置是否应用单个块停止功能到宏程序(9000.nc ~ 9999.nc) 0: 停止 1: 不停止
			2	显示宏程序块	设置在执行宏程序时, 是否在屏幕上显示块进程状态(9000.nc ~ 9999.nc). 0: 不显示 1:显示
			9	T 代码调用宏程序编号	当执行 T 代码时, 输入调用的宏程序编号 (9000.nc ~ 9009.nc). (9000 ~ 9009, 整数)
			10	宏程序调用 G 代码(9010.nc ~ 9019.nc)	设置可以被 G 代码调用的宏程序调用 G 代码编号(9010.nc ~ 9019.nc). ※忽略设置值 0, 1, 2, 3. (0~255.9, 实数)
			

第 9 章 NC 控制功能

参数	组	No.	项目	描述	
1.通道参数	11. 宏程序	19	宏程序调用 G 代码(9019.nc)	设置可以被 G 代码调用的宏程序调用 G 代码编号(9010.nc ~ 9019.nc). ※忽略设置值 0, 1, 2, 3. (0~255.9, 实数)	
		20	宏程序调用 M 代码(9020.nc)	通过 M 代码分配调用宏程序的 M 代码编号(9020.nc ~ 9029.nc). ※忽略输入值的 0, 30. (0~255, 整数)	
			
		29	宏程序调用 M 代码(9029.nc)	通过 M 代码分配调用宏程序的 M 代码编号(9020.nc ~ 9029.nc). ※ 忽略输入值的 0, 30. (0~255, 整数)	
	14. 默认设置	14. 默认设置	1	默认设置模式移位	如果没有 G00 或者 G01, 选择应用为默认模式的 G 代码. 0: 快速移位(G00) 1: 切割馈送(G01)
			2	默认设置模式平面	如果没有 G17, G18, G19 组的 G 代码指令, 选择应用为默认模式的 G 代码. 0: XY 平面(G17) 1: XZ 平面(G18) 2: YZ 平面(G19)
			3	默认设置模式绝对/增量	如果没有 G90, G91 组的 G 代码指令, 选择应用为默认模式的 G 代码. 0: 绝对指令(G90) 1: 增量指令(G91)
			5	查看默认设置的模式禁止区域	如果没有 G22, G23 组的 G 代码指令, 选择应用为默认模式的 G 代码. 0: 行程 On(G22) 1: 行程 Off(G23)

参数	组	No.	项目	描述
1.通道参数	16. 相对坐标设置	1	相对坐标补偿值 #1	设置 X 轴相对坐标补偿值.
		2	相对坐标补偿值 #2	设置 Y 轴相对坐标补偿值.
		3	相对坐标补偿值 #3	设置 Z 轴相对坐标补偿值.
		4	相对坐标补偿值 #4	设置 A 轴相对坐标补偿值.
		5	相对坐标补偿值 #5	设置 B 轴相对坐标补偿值.
		6	相对坐标补偿值 #6	设置 C 轴相对坐标补偿值.
		7	相对坐标补偿值 #7	设置 U 轴相对坐标补偿值.
		8	相对坐标补偿值 #8	设置 V 轴相对坐标补偿值.
		9	相对坐标补偿值 #9	设置 W 轴相对坐标补偿值.
2.通道/轴参数	1. 轴设置	2	设置模块轴方向	设置作为模块轴的轴设置移位指令. 0: 单向 1: 双向
	2. 原点	1	2nd 原点坐标	设置 2nd 原点坐标
		2	3rd 原点坐标	设置 3rd 原点坐标
		3	4th 原点坐标	设置 4th 原点坐标
	3. 快速移位	2	快速移位加速	用于 G00 块加速的设置值.
		3	快速移位减速	用于 G00 块减速的设置值.
		4	快速移位加速度	用于 G00 块加速度的设置值.

第 9 章 NC 控制功能

参数	组	No.	项目	描述
2.通道/轴参数	3. 快速移位	5	快速移位速度	用于 G00 块移位速度的设置值. (0~100000 单位/min, 实数)
	4. 移位区域	1	X, Y, 和 Z 轴 G22 移位-禁止区域范围最小值.	设置 X, Y, 和 Z 轴 G22 移位-禁止区域范围最小值. (-100,000~100,000 单位, 实数)
		2	X, Y, 和 Z 轴 G22 移位-禁止区域范围最大值.	设置 X, Y,和 Z 轴 G22 移位-禁止区域范围最大值. (-100,000~100,000 单位, 实数)
		3	X, Y, 和 Z 轴 3 rd 移位-禁止区域范围最小值.	设置 X, Y, 和 Z 轴 3 rd 移位-禁止区域范围最小值. (-100,000~100,000 单位, 实数)
		4	X, Y, 和 Z 轴 3 rd 移位-禁止区域范围最大值.	设置 X, Y, 和 Z 轴 3 rd 移位-禁止区域范围最大值. (-100,000~100,000 单位, 实数)
	5. 子设置	2	单方向定位过度运行馈送率	<p>当使用单方向定位功能(G60)时, 设置 9 轴过度运行馈送率; X, Y, Z, A, B, C, U, V, W.</p> <p>通过 G60 指令块轴设置值分别停止定位后, 移动到指令位置以消除反冲影响.</p>  <p>(-100 ~ 100 单位, 实数)</p>

第 10 CPU功能

10.1 任务设计

10.1.1 任务概述

有 3 中动作控制任务类型: 主任务, 周期性任务和初始化任务.

任务类型	程序数量	动作
主任务	最多 256 个	<ul style="list-style-type: none"> • 执行 I/O 更新, 分配到主任务和运动控制的程序进程. • 用于每个确定控制间隙一次执行以上任务(主任务循环). • 比周期性任务更高的优先级. • 由于可以快速处理程序, 使用要求同步控制和通过分配高速运行进程的的程序. • 可设置间隔: 1ms, 2ms, 4ms
周期性任务		<ul style="list-style-type: none"> • 执行分配到主任务的程序进程. • 在实现控制间隙范围内的主任务运行后, 用于维持时间的执行, 可以在多周期内执行. • 由于在主任务程序的运动控制指令执行中, 其优先级低于主任务, 执行于主任务程序中的运动控制指令首先执行. • 其使用处理其他监控数据和设备控制的程序, 其中设备不要求通过分配的高速处理. • 可设置间隔: 1ms ~ 100ms (设置主任务循环的倍数)
初始化任务		<ul style="list-style-type: none"> • 完成 I/O 更新后执行分配到初始化任务的程序进程. • 仅在输入 RUN 模式的时候执行一次. • 当输入 RUN 模式时首先执行. 如果初始化任务完成(_INIT_DONE)标志位通过初始化任务程序设置, 任务完成, 主任务和周期性任务程序启动.

(1) 主任务和周期性任务

主任务和周期性任务都是以固定间隔执行.主任务和周期性任务执行的间隔被称为'任务周期'.主任务可设置周期为500 μ s, 1ms, 2ms and 4ms, 周期性任务可设置周期为主任务周期的倍数. 然而, 周期性任务周期可设置为ms单位.

在任务中, 可使用1到256程序. 按照分配的顺序执行程序.

参考数字I/O模块和模拟量模块间数据交换的I/O更新. I/O更新在每个周期的主任务执行起始阶段执行.

第 10 章 CPU 功能

(2) 任务分割

所有程序分配到一个任务中. 用户要求根据如下表中创建程序的特性分配任务.

任务	适用程序
主任务	<ul style="list-style-type: none">• I/O 更新执行周期应严格遵守.• 最高执行优先级• 包含高等级运动控制.
周期性任务	<ul style="list-style-type: none">• 程序执行可以忽略 I/O 更新• 程序优先级低于主任务, 并要求周期性执行
初始化任务	<ul style="list-style-type: none">• 在 RUN 运行期间要求在主任务程序执行前执行• 设备初始化和初始值设置程序

10.1.2 任务规格

任务规格如下.

项目	规格
任务类型	<ul style="list-style-type: none">• 主任务• 周期性任务• 初始化任务
任务程序数量	<ul style="list-style-type: none">• 最多 256 个
主任务周期	<ul style="list-style-type: none">• 500 μs, 1 ms, 2 ms, 4 ms
周期性任务周期	<ul style="list-style-type: none">• 可以设置主任务从 1 到 100 ms 的倍数
初始化任务周期	<ul style="list-style-type: none">• 与主任务周期相同

周期性任务周期根据主任务周期进行设置

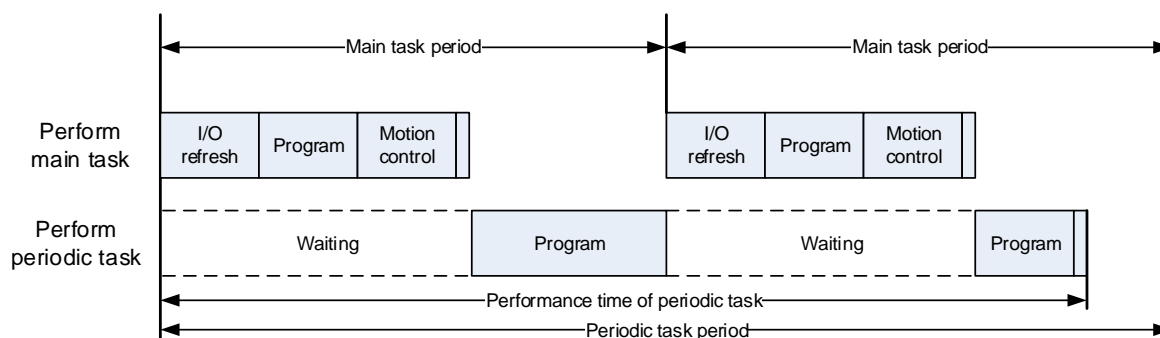
主任务周期	可设置周期性任务周期
500 μ s	1 ms, 2ms, 3ms, 4ms, 5ms, ... 95ms, 96ms, 97ms, 98ms, 99ms, 100ms
1 ms	1 ms, 2ms, 3ms, 4ms, 5ms, ... 95ms, 96ms, 97ms, 98ms, 99ms, 100ms
2 ms	2ms, 4ms, 6ms, 8ms, 10ms, ... 92ms, 94ms, 96ms, 98ms, 100ms
4 ms	4ms, 8ms, 12ms, 16ms, 20ms, ... 84ms, 88ms, 92ms, 96ms, 100ms

10.1.3 任务基本操作

MP500-E32A 无法同时执行多个任务. 根据其优先级执行每个任务, 主任务程序具有比周期性任务程序更高的优先级.

如果在周期性任务执行期间主任务程序到达执行周期, 执行主任务程序. 因此, 当根据周期执行主任务程序时, 周期性任务程序可以在多个主任务周期中执行. 如果使用周期性任务程序, 可以参考这点写入程序.

另外, 应设置周期, 以便在设置周期内任务程序执行完成. 如果任务超出设置周期, 发生报警. 如果直到周期错误检测时间任务执行未完成, 系统切换到 ERROR 状态.

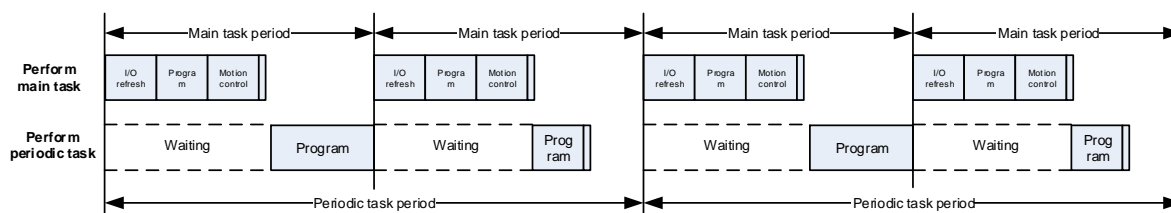


在主任务执行中，在程序执行运动控制或者任务执行完成的周期性任务执行显示后显示隔行。

(1) 主任务和周期性任务周期

主任务和周期性任务以周期反复执行。两个任务都有任务执行周期，周期性任务周期可以设置为主任务周期的倍数。

例如，如果主任务周期是1ms，而周期性任务周期是2ms，主任务每执行两次，执行一次周期性任务。



(2) 初始化任务

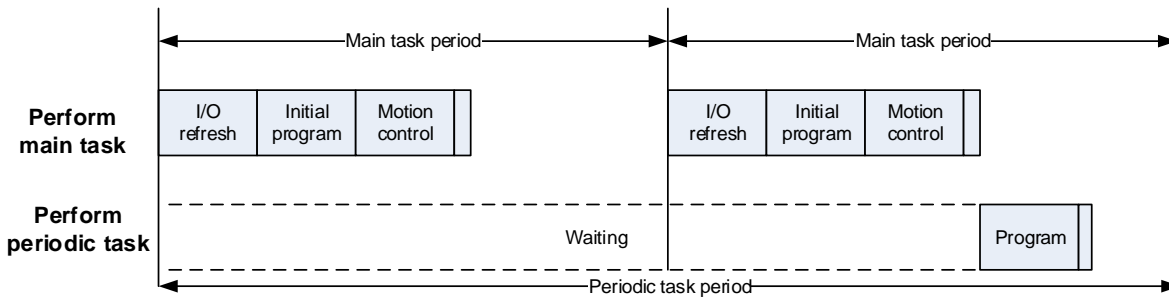
在主任务执行前，直到初始化任务执行完成(`_INIT_DONE`)标志位设置后执行初始化任务，当程序中用户设置 `_INIT_DONE`标志位时停止任务。仅当初始化任务停止时，执行主任务和周期性任务程序。初始化任务周期延续主任务周期。

备注

如果通过用户写入初始化任务程序设置初始化任务执行完成(`_INIT_DONE`)标志位,初始化任务程序执行终止,启动主任务和周期性任务程序执行。
初始化任务在主任务周期内运行，包含在主任务执行时间内。

第 10 章 CPU 功能

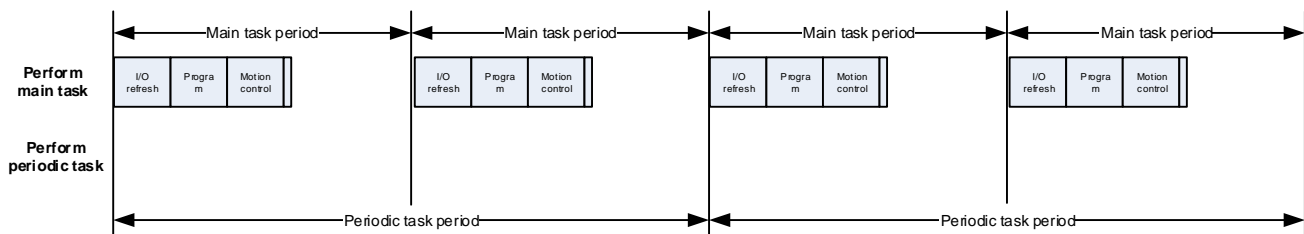
当初始化程序执行完成时，如下初始化任务执行终止，主任务程序和周期性任务程序执行。



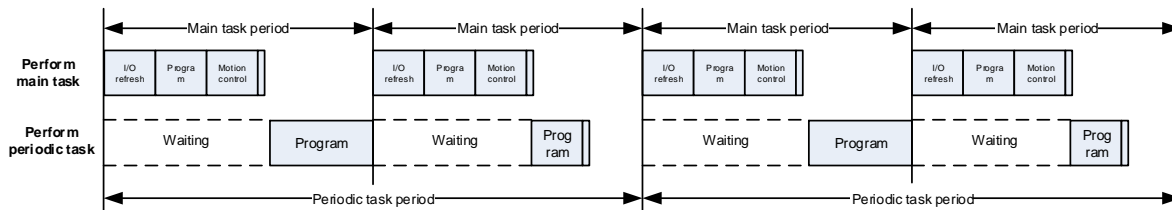
10.1.4 任务执行顺序实例

以下描述主任务和周期性任务的顺序执行。

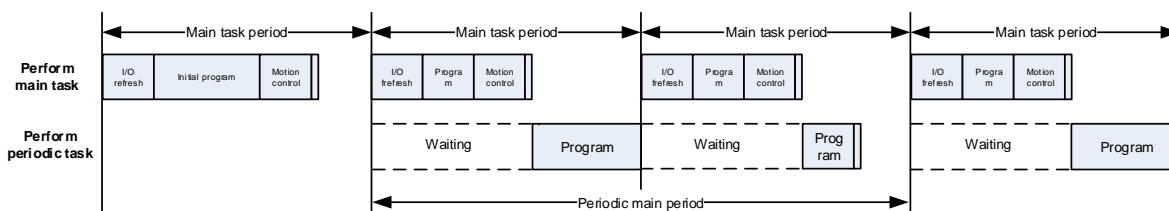
(1) 如果仅有主任务程序



(2) 如果有主任务/周期性任务程序



(3) 如果有初始化任务程序

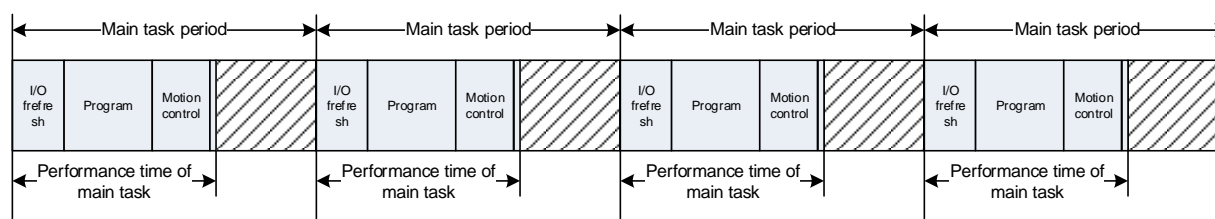


10.1.5 系统服务处理

系统服务包括以下服务。

系统服务名称	内容
USB 服务	<ul style="list-style-type: none"> • MP500 中服务请求处理
内置 Ethernet 端口服务	<ul style="list-style-type: none"> • MP500 中服务请求处理 • 通讯(P2P)服务处理 • FTP 服务处理
SD 存储卡服务	<ul style="list-style-type: none"> • SD 存储卡指令执行 • 数据记录

主任务或者周期性任务完成后执行系统服务，当任务不运行时，在空闲时间运行。如下显示在阴影部分执行系统服务。

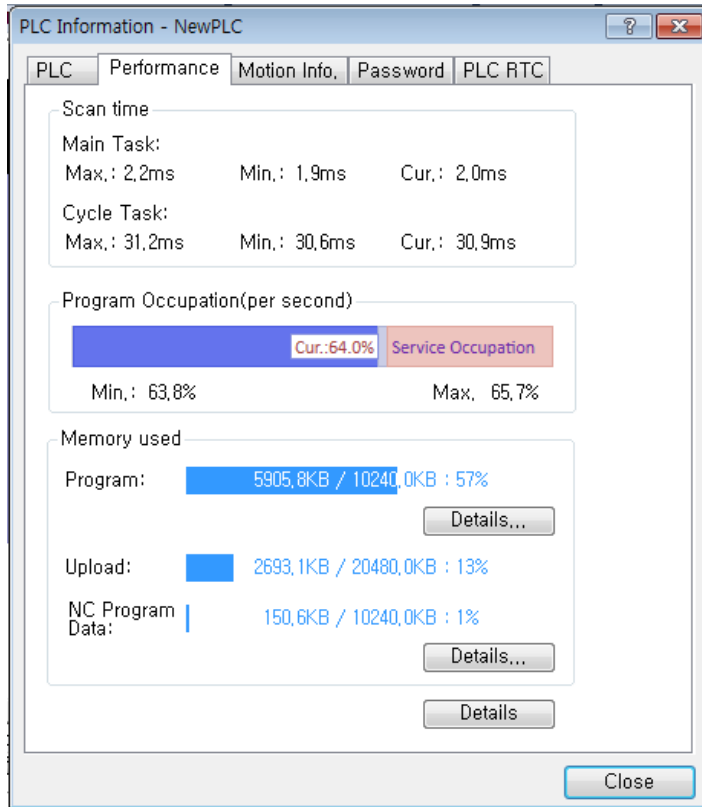


系统服务和每个任务的优先顺序为主任务>周期性任务>系统服务，主任务具备最高优先级。在系统服务运行期间达到主任务执行周期时，系统服务暂停，执行主任务。另外，在系统服务运行期间达到主任务执行周期时，系统服务暂停，执行周期性任务。当主任务和周期性任务执行完成时，暂停的系统服务按顺序执行。

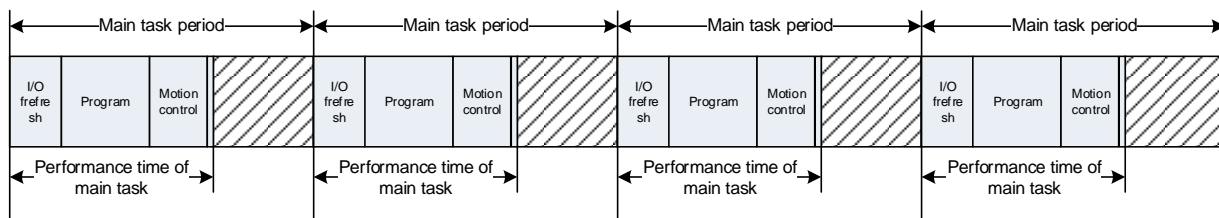
10.1.6 程序占有率运行

程序占有率参考系统RUN运行期间每秒任务执行时间的比率。如果仅有主任务,主任务执行时间的总和以百分比形式显示。如果有周期性任务,计算主任务和周期性任务执行时间,并以百分比形式显示。

在下图中,程序占用率当前为64%,表示每秒时间主任务和周期性任务执行约64%,系统服务在剩余时间内运行。

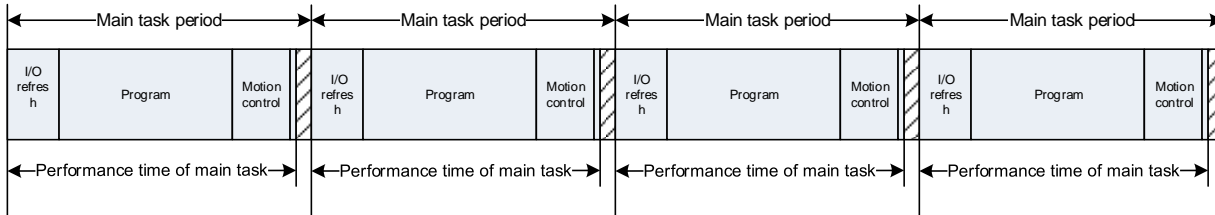


如果如下没有周期性任务,如果主任务执行完成,系统服务在阴影部分执行。

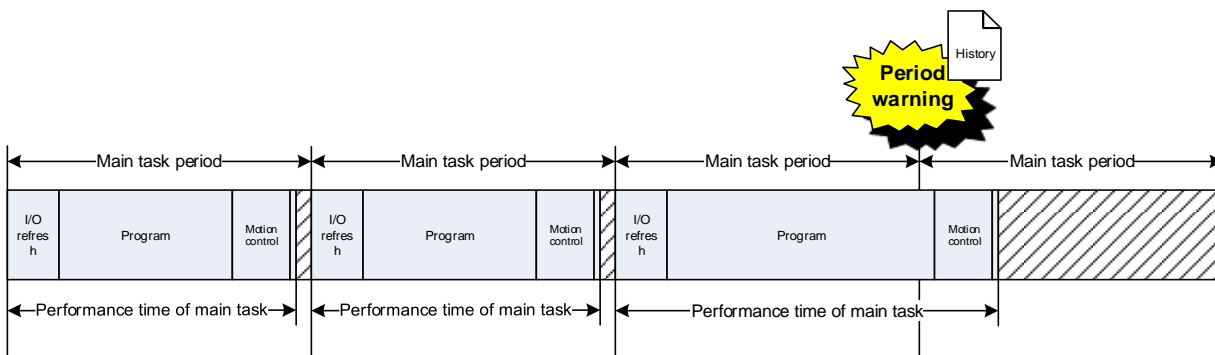


如果如下程序占有率很高(系统服务占有率低), 系统服务无法正常执行. 在基本参数中, 用户可设置范围值从50到95%, 如果设置值超出范围, 发生任务程序占有率警报. 如程序占有率超出100%, 系统进入ERROR状态.

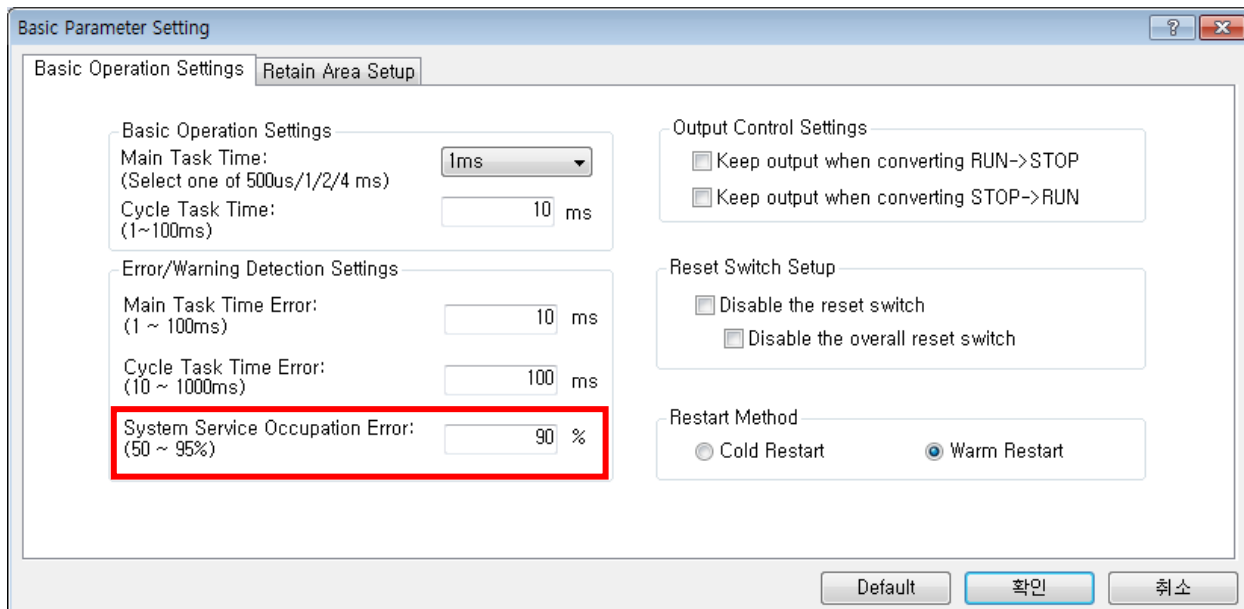
※调整主任务周期, 以便程序占有率不超过90%.



如果如下发生周期警报, 程序占有率增加.



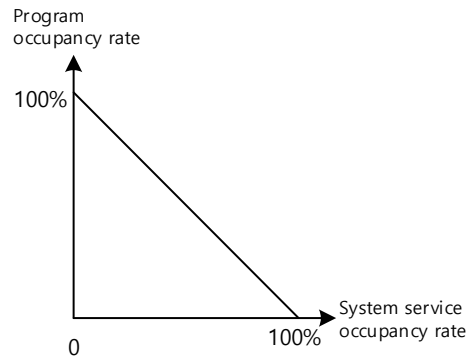
基本参数设置中设置任务程序占有超出报警检测值从50%到95%.



第 10 章 CPU 功能

如果任务程序占有率超出设置值,发生任务占有率超出警报(`_TASK_PRM_USAGE_OVER_WAR`). 如果任务程序占有率超出100%, 系统状态从RUN切换到ERROR, 发生任务程序占有率超出错误(`_TASK_PRM_USAGE_OVER_ER`).

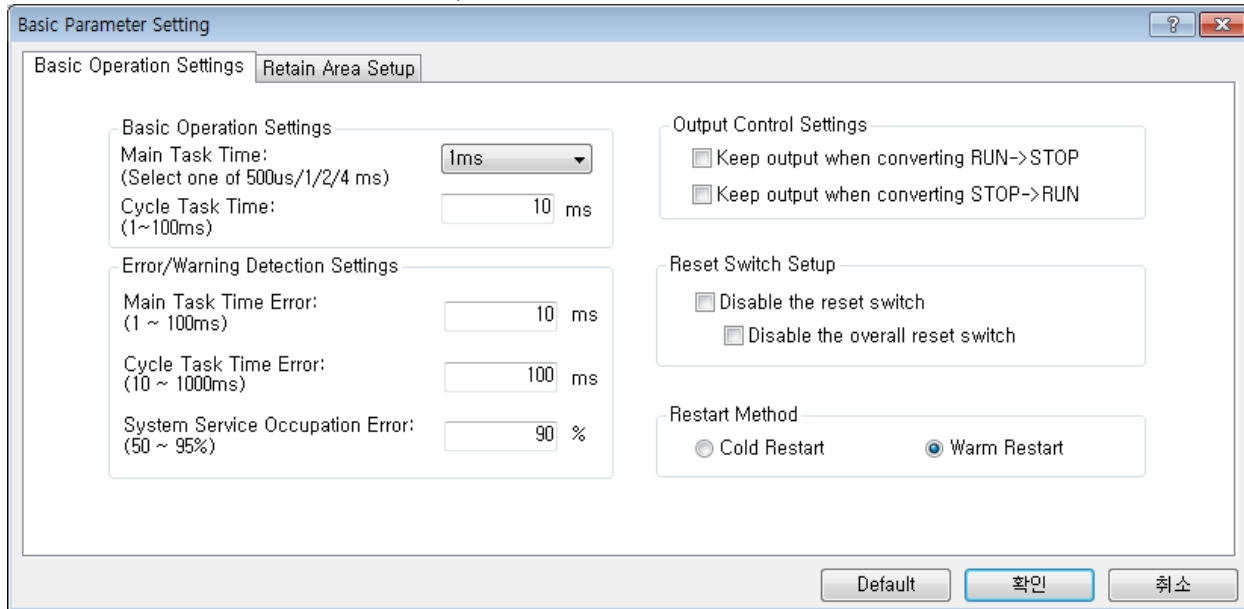
程序占有率与系统占有率成反比. 如果程序占有率是20%, 系统占有率是80%. 但是如果程序占有率为80%, 系统占有率为20%.



任务程序占有率增加表示主任务和周期性任务占一个周期的大部分, 因此系统服务执行要求时间减少. 请确保程序占有率不超过95%. 如果超出95%, 变更主任务周期.

10.1.7 任务设置项目

为了执行任务程序，设置以下任务相关项目。当基本参数项目发送时，立即反映每个项目。即使不使用周期性周期，也应设置周期。更多基本参数设置内容，参考10.2.1基本参数设置。



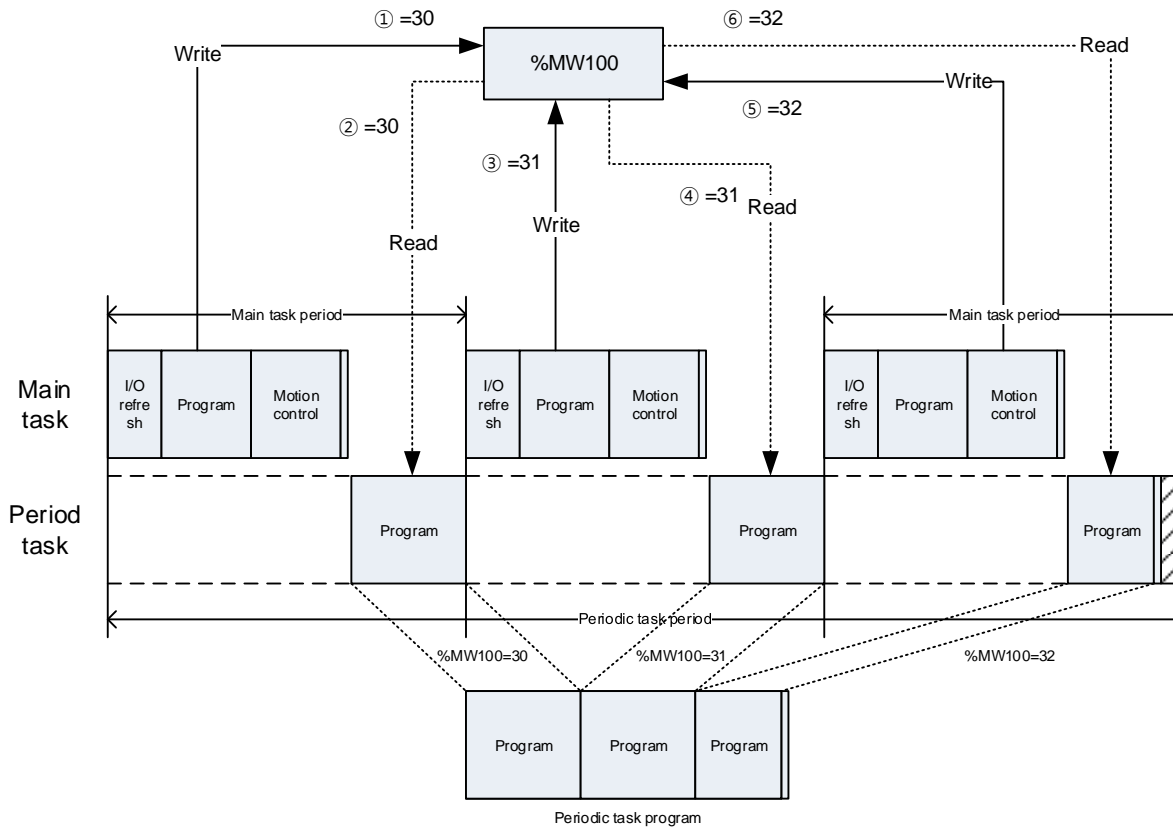
项目	描述	设置值	默认值
主任务周期	设置主任务时间	500 μ s, 1 ms, 2 ms, 4 ms	1 ms
周期性任务周期	设置周期性任务时间为主任务周期的倍数	1~100 ms	10 ms
主任务周期错误	当主任务执行超出设置时间时，设置主任务执行时间	1~100 ms	10 ms
周期性任务周期错误	当周期性任务执行超出设置时间时，设置导致错误的周期性任务执行时间	10~1000 ms	100 ms
任务程序占有率警告	如果由于有太多主任务程序和或者周期性任务程序，而导致任务程序占有率超出设置值，发生任务程序占有率警告。如果任务程序占有率超出 100%，发生任务程序占有率错误，并切换到 ERROR 状态。	50~95%	90%

10.1.8 如何使用任务间变量的方式

特别注意当在主任务和周期性任务中读取和写入相同全局变量时.

如果如下在主任务和周期性任务程序中读取和写入 %MW100 值时, 根据周期性任务中使用位置持续变更 %MW100 值.

- ① 在主任务程序中写入 %MW100 值为 30
- ② 当从周期性任务读取时 %MW100 值为 30
- ③ 在主任务程序中写入 %MW100 值为 31
- ④ 当从周期性任务读取时 %MW100 值为 31
- ⑤ 在主任务程序中写入 %MW100 值为 32
- ⑥ 当从周期性任务读取时 %MW100 值为 32



如果在周期性任务的一个周期内 %MW100 值持续变更为 30, 31 和 32, 值可能因为设备使用位置的不同而不同. 如果在周期性任务程序中, 以全局变量(例: %MW100)写入的值用于主任务程序, 主任务程序运行结果可能受到影响.

※ 请注意当编程时尽可能避免使用主任务程序和周期性任务之间的相同设备.

10.1.9 任务标志位

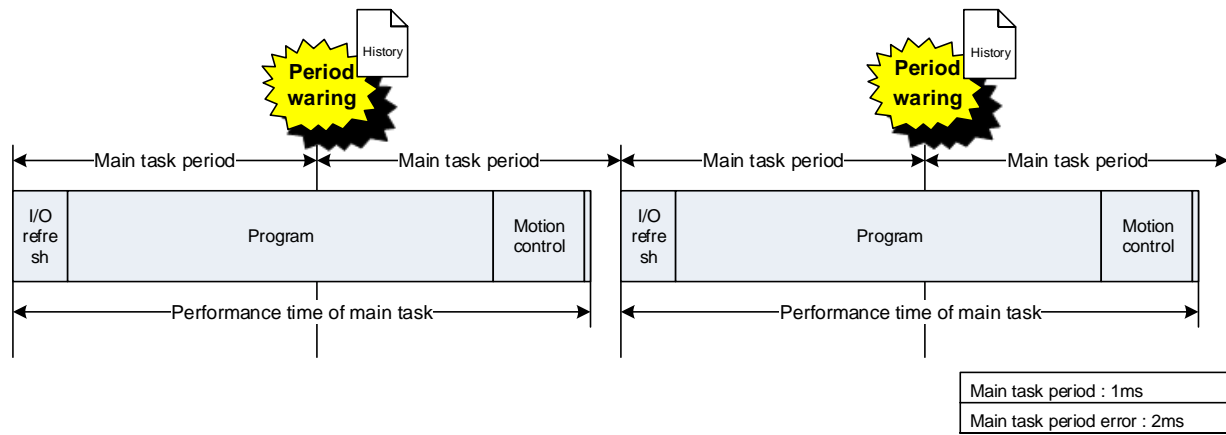
以下为任务标志位描述.

标志位名称	类型	设备	描述
_PROGRAM_RATIO_MAX	UINT	%FW518	用户程序最大执行占有(1sec)
_PROGRAM_RATIO_MIN	UINT	%FW519	用户程序最小执行占有(1sec)
_PROGRAM_RATIO_CUR	UINT	%FW520	用户程序当前执行占有(1sec)
_PTASK_SCAN_WR	BOOL	%FX20486	主任务扫描值初始化
_PTASK_CYCLE_WAR_NUM	UINT	%FW748	主任务间隔超出警告计数
_PTASK_CYCLE_WAR	BOOL	%FX129	主任务间隔超出警告
_PTASK_SCAN_MAX	UINT	%FW512	主任务最大扫描时间(单位:100 us)
_PTASK_SCAN_MIN	UINT	%FW513	主任务最小扫描时间(单位:100 us)
_PTASK_SCAN_CUR	UINT	%FW514	主任务当前扫描时间(单位:100 us)
_CTASK_SCAN_WR	BOOL	%FX20487	周期性任务扫描时间初始化
_CTASK_CYCLE_WAR_NUM	UINT	%FW749	周期性任务间隔超出警告计数
_CTASK_CYCLE_WAR	BOOL	%FX130	周期性任务间隔超出警告
_CTASK_SCAN_MAX	UINT	%FW515	周期性任务最大扫描时间(单位:100us)
_CTASK_SCAN_MIN	UINT	%FW516	周期性任务最小扫描时间(单位:100us)
_CTASK_SCAN_CUR	UINT	%FW517	周期性任务当前扫描时间(单位:100us)
_CTASK_CYCLE_ER	BOOL	%FX92	周期性任务周期错误
_PTASK_CYCLE_ER	BOOL	%FX91	主任务周期错误
_INIT_DONE	BOOL	%FX20496	初始化任务完成
_INIT_RUN	BOOL	%FX24	执行初始化任务
_TASK_PRM_USAGE_OVER_WAR	BOOL	%FX135	任务程序占有超出警告
_TASK_PRM_USAGE_OVER_ER	BOOL	%FX94	任务程序占有超出错误

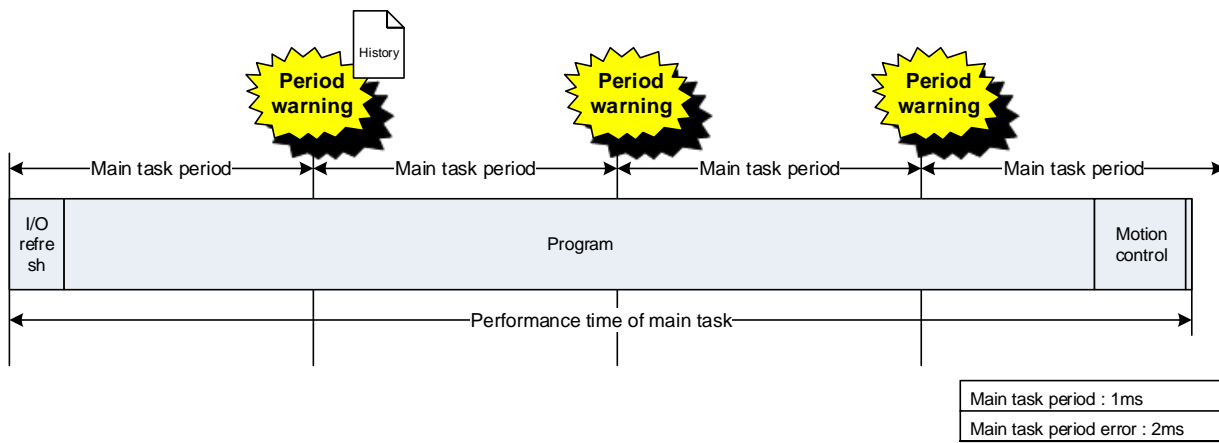
10.1.10 任务-相关警告/错误

(1) 任务周期超出警告

如果主任务或者周期性任务超出用户设置周期，发生周期超出警告。警告保存在错误历史中。

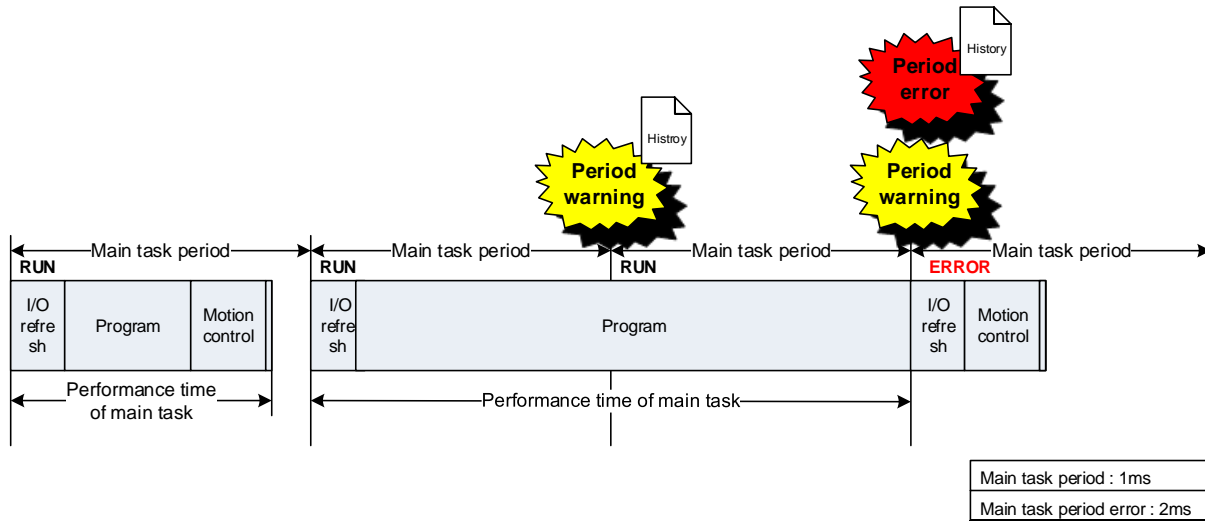


如下图显示，在前一个周期任务执行完成，保存历史。如果周期超出警告持续发生，周期超出警告历史仅保存第一个发生的警告。可以在错误历史中查看保存的历史。



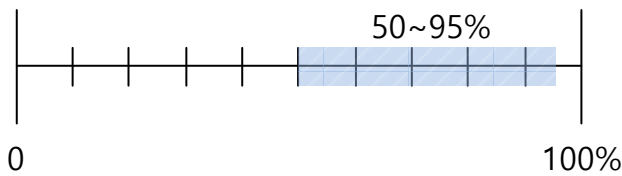
(2) 任务周期错误

如果以超出基本参数中设置的周期错误时间执行任务，发生周期超出错误。参考部分0任务程序占有率超出警告/错误纠正措施。

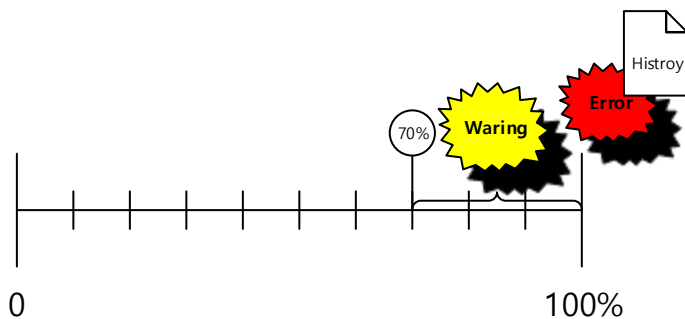


(3) 程序占有率超出警告/错误

在基本参数中，程序占有率超出警告检测设置值可设置为50~95%。如果超出用户设置值，发生程序占有率超出警告，如果为100%，发生程序占有率超出错误。



如下图中显示，如果基本参数程序占有率超出警告检测值设置为70%，当程序占有率范围从70到99%时发生警告，当到达100%时发生错误。

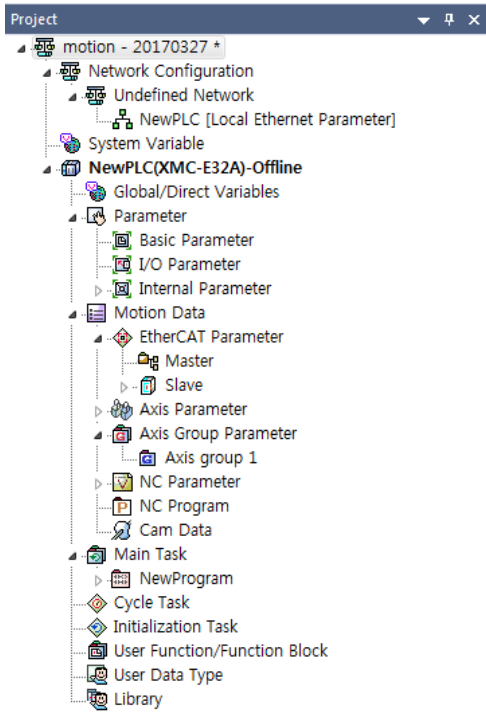


10.2 参数设置

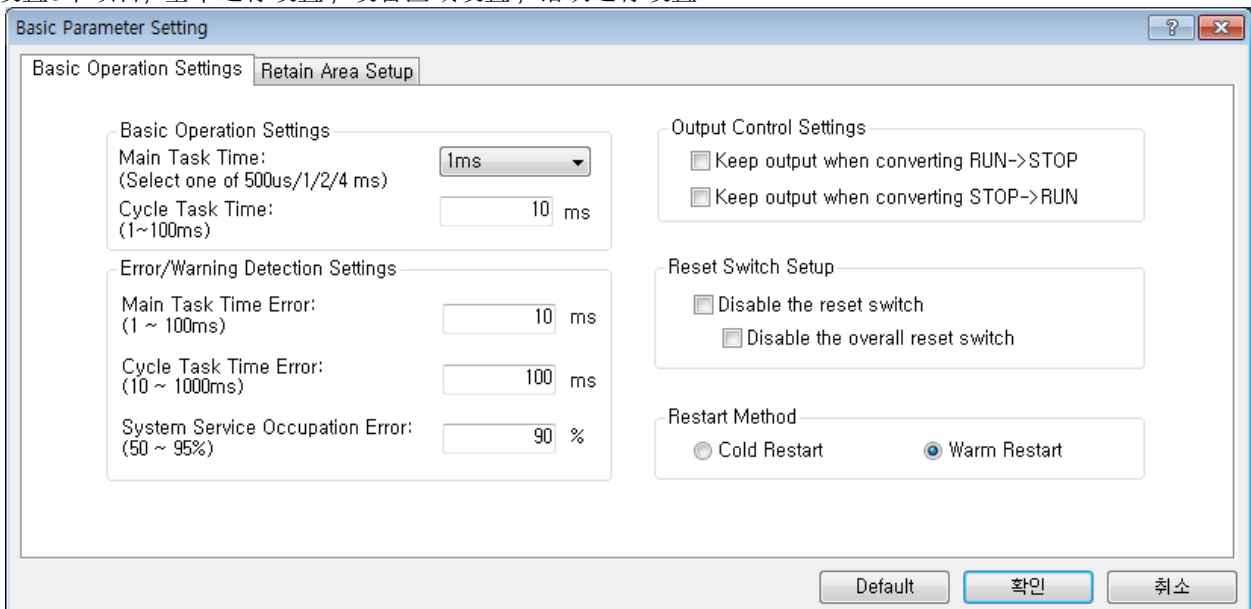
本节描述运动控制的参数设置.

10.2.1 基本参数设置

如果在工程窗口点击基本参数, 显示如下画面.



可以设置3个项目; '基本运行设置', '设备区域设置', '错误运行设置'.



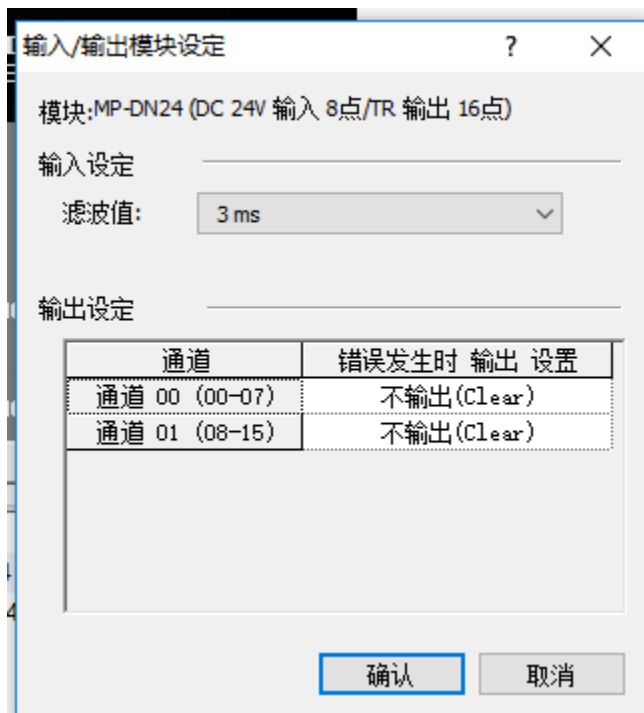
分类	项目	描述	设置值
基本运行	主任务周期	设置主任务时间	500us, 1ms, 2ms, 4ms
	周期性任务周期	设置周期性任务时间	1~100ms(主任务的倍数)
	主任务周期错误	当执行主任务超出设置时间时, 设置导致错误的主任务执行时间	1~100ms
	周期性任务周期错误	当执行周期性任务超出设置时间时, 设置导致错误的周期性任务执行时间	10~1000ms
	任务程序占有率警告	如果由于主任务程序或者周期性任务程序过多导致任务程序占有率超出设置值, 发生任务程序占有率警告. 如果任务程序占有率超过 100%, 发生任务程序占有率错误, 并切换到 ERROR 状态.	50~95%
	对于 Run->Stop 转换的输出维护	当 Run->Stop 转换允许时保持输出	允许/禁止
	对于 Stop → Run 转换的输出维护	当 Stop->Run 转换允许时保持输出	允许/禁止
	复位开关运行停止	设置是否通过产品前端面板的开关执行复位运行	允许/禁止
	全部复位开关运行停止	设置是否通过产品前端面板的开关执行全部复位运行	允许/禁止
	重启模式	选择重启模式	冷启动, 热启动
存储区域设置	保持区域设置	设置 M 区域保持范围	在%MW0~%MW1048575, 524,288 字间设置

10.2.2 I/O参数设置

该功能用于建立和保存每个I/O的信息. 如果点击工程窗口的『I/O参数』, 显示如下设置窗口.



如果在『槽』位置点击『模块』, 显示每个模块的列表. 然后, 选择与配置的实际系统匹配的模块. 如下显示所选择槽.



特殊模块参数设定

? X

XMF-AH04E (输入 2通道, 电压输出 2通道)

输入 参数	通道0	通道1
<input type="checkbox"/> 运行通道	停止	停止
<input type="checkbox"/> 输入范围	4~20mA	4~20mA
插入输出数据	0~16000	0~16000
过滤器相数	0	0
<input type="checkbox"/> 平均方法	样品循环	样品循环
平均值	0	0
<input type="checkbox"/> 有效变换值维持	禁止	禁止
输出 参数	电压 Ch0	电压 Ch1
<input type="checkbox"/> 运行通道	停止	停止
<input type="checkbox"/> 输出范围	1~5V	1~5V
输入数据插入	0~16000	0~16000
<input type="checkbox"/> 通道输出状态设定	之前的值输出	之前的值输出
<input type="checkbox"/> 步间方法设定	禁止	禁止
步间时间	10[ms]	10[ms]

确认

取消

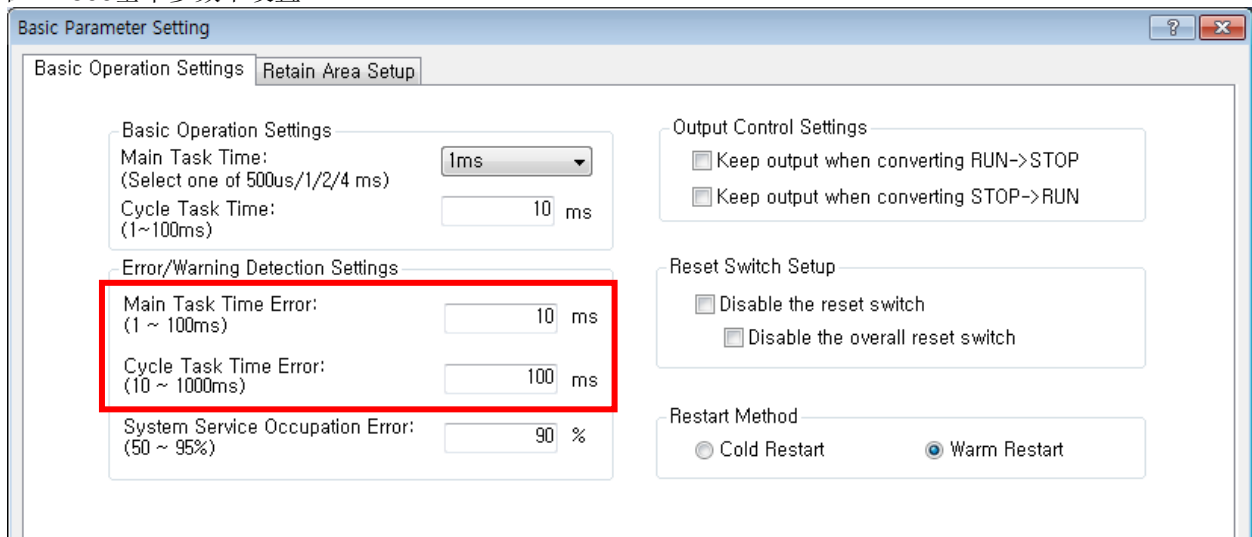
10.3 自诊断功能

自诊断功能时 CPU 部分诊断运动控制系统错误的功能。在运动控制系统供电或者运行期间发生错误的情况，其检测错误以防止系统故障和预防性维护。

10.3.1 主任务/周期性任务周期错误

主任务/周期性任务周期错误是运动控制器软件错误或者通过用户程序导致的周期性错误的检测功能。

- (1) 用于检测程序超出用户设定时间导致主任务/周期性任务程序运行延迟错误。主任务/周期性任务周期错误检测时间如下在MP500基本参数中设置。



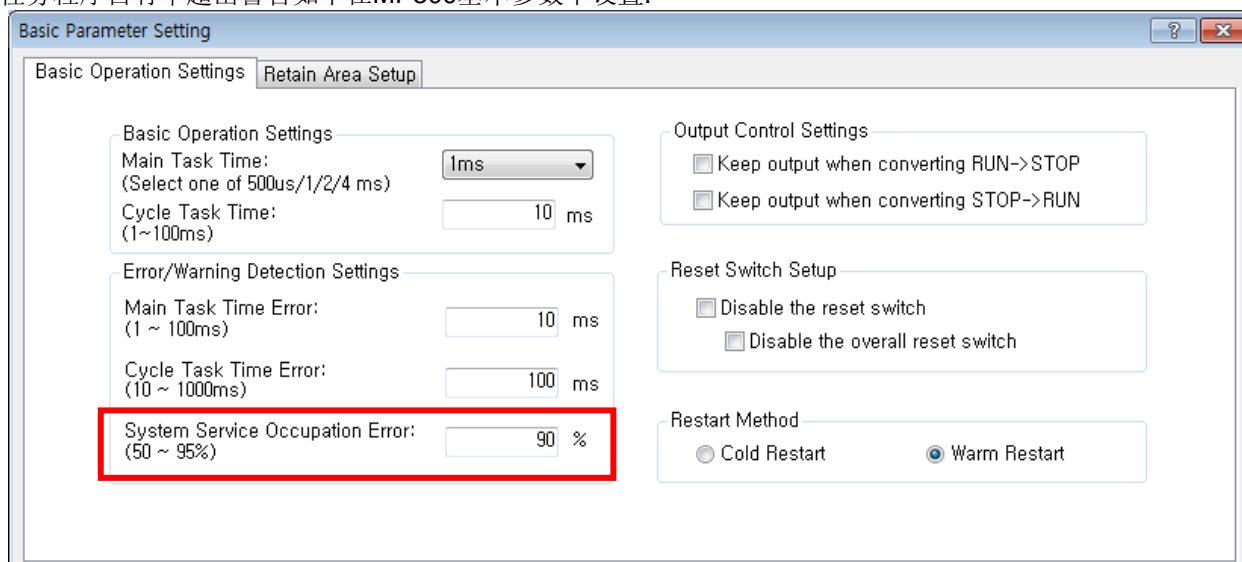
- (2) 当程序运行时，监控扫描时间，如果设置检测时间超出，运动控制运行立即停止，并发生错误。

- (3) 当主任务/周期性任务错误发生，如果电源再次上电，则错误清除，或者将模式切换到Stop模式

10.3.2 任务程序占有率超出警告/错误

如果由于主任务/周期性任务执行导致程序占有率增加，系统服务无法执行。为了防止这个情况发生，该功能允许用户检测任务程序占有率超出警告/错误。(系统服务:除了主任务/周期性任务/初始化任务的服务)

(1) 任务程序占有率超出警告如下在MP500基本参数中设置。



(2) 如果在执行程序期间监控任务程序占有率时，设置占有率超出，发生任务程序占有率超出警告。如果当警告发生时任务程序占有率为100%，发生任务程序占有率错误。

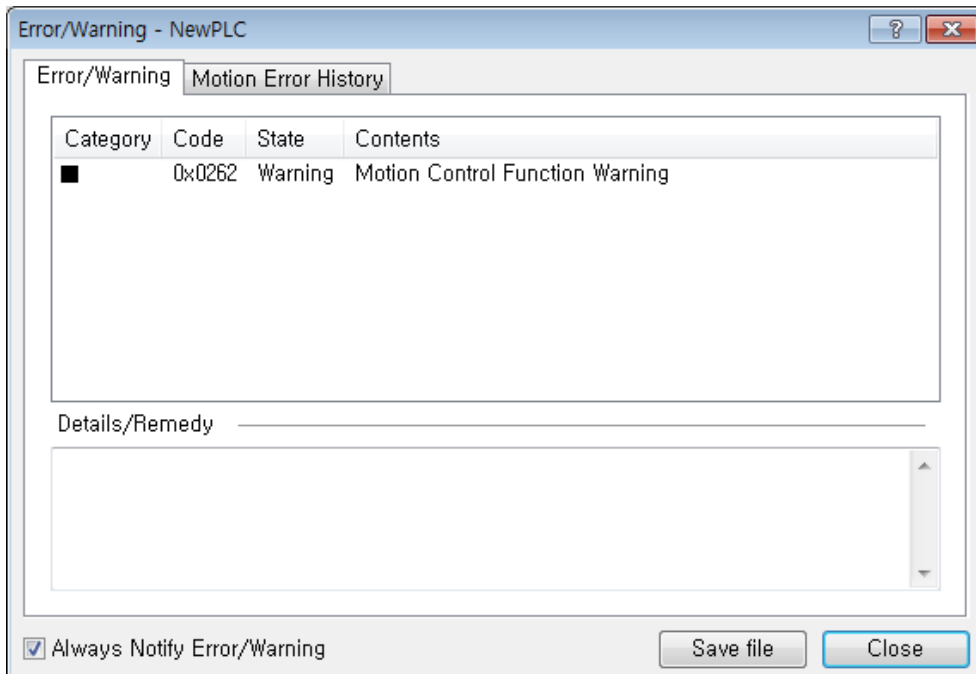
(3) 当任务程序占有率警告/错误发生时要求执行如下措施。

- (a) 通过减少用户执行主任务/周期性任务数量确保系统服务运行时间。
- (b) 通过增加基本参数中主任务/周期性任务执行周期确保系统服务运行时间。
- (c) 增加基本参数任务程序占有率超出警告设置。

10.3.3 错误历史存储功能

当错误发生时运动控制器用于记录错误历史，找出错误原因并纠正。

点击菜单中『在线』-『诊断』-『PLC错误/警告』项目查看当前错误和错误历史。请参考每个错误项目和排除错误原因的详情和措施内容。



项目	描述	备注
错误/警告	显示当前错误/警告	-
错误历史	显示按时间顺序发生的错误/警告	最多 100 个

备注

通过点击错误/警告窗口的‘清除’，删除保存的错误历史。
 如果错误历史超出 1,024，将移除最早的历史记录，保存最近的 1,024 个历史记录。

10.3.4 失效管理

(1) 失效类型

由于运动控制器自身，系统配置错误，运行结果错误检测等造成的问题。可以分为为了系统安全的失效模式停止运行；通知用户失效警告并恢复运行的一般失效模式。

运动控制系统失效主要由以下原因导致。

- 运动控制器硬件问题
- 用户程序执行期间的运行错误
- 由于外部设备失效的错误检测

(2) 失效情况下的运行模式

对于失效发生，运动控制系统在特殊标志位(F区域)中记录失效详情，并基于失效模式确定是否重新运行。

- 对于运动控制器硬件失效的情况
 如果运动控制器，电源等有问题，运动控制器无法正常工作，则系统停止；对于一般故障例如电池低电压,显示警告并恢复运行。

- 执行用户程序期间的计算误差
对于执行用户程序期间发生数字运行错误(例: 除运算分母为0), 详情在错误标志位中显示, 系统恢复运行. 如果在运行期间运行时间超出运行延迟监控设置时间, 或者配置的I/O模块无法正常控制, 系统将停止.
- 外部设备失效导致错误的检测
外部控制设备的失效可以通过运动控制器用户程序进行检测; 如果检测到失效, 系统停止; 如果检测到一般失效, 仅显示检测状态而运行持续.(对于检测外部设备失效功能使用详情, 参考10.3.5 外部设备失效诊断功能.)

发生失效的信息保存在特殊继电器(F区域)中. 在F区域标志位中, 失效相关信息如下.

双字	位	标志位名称	功能	描述
%FD0	%FX2	_ERROR	错误	错误状态
%FD2	-	_CNF_ER	系统错误	报告系统失效状态.
	%FX70	_ANNUM_ER	外部设备失效	从外部设备检测失效.
	%FX72	_BPRM_ER	基本参数	基本参数中有某些问题.
	%FX73	_IOPRM_ER	IO 参数	I/O 参数中有某些问题.
	%FX74	_SPPRM_ER	特殊模块参数	异常特殊模块参数
	%FX75	_CPPRM_ER	通讯模块参数	异常通讯模块参数
	%FX76	_PGM_ER	程序错误	程序有错误.
	%FX78	_SWDT_ER	系统看门狗	系统看门狗工作.
	%FX80	_SWDT_ER	系统看门狗	系统看门狗工作.
	%FX85	_ENCPRM_ER	编码器参数错误	异常编码器错误
	%FX86	_AXISPRM_ER	轴参数	异常轴参数
	%FX87	_GROUPPRM_ER	轴组参数	异常轴组参数
	%FX88	_ECPRM_ER	EtherCAT 参数	异常 EtherCAT 参数
	%FX89	_NCPRM_ER	NC 参数	异常 NC 参数
	%FX90	_NCPGM_ER	NC 程序	NC 参数错误
	%FX91	_PTASK_CYCLE_ER	主任务	主任务周期错误
	%FX92	_CTASK_CYCLE_ER	周期性任务	周期性任务周期错误
	%FX93	_SYSTEM_ER	系统错误	系统错误
%FX94	_TASK_PRM_USAGE_OVER_ER	任务程序占有率超出错误	任务程序占有率超出 100%	

第 10 章 CPU 功能

双字	位	标志位名称	功能	描述
%FD4	-	_CNF_WAR	系统警告	报告系统一般失效状态.
	%FX128	_RTC_ER	RTC 数据错误	异常 RTC 数据
	%FX129	_PTASK_CYCLE_WAR	主任务	主任务周期错误
	%FX130	_CTASK_CYCLE_WAR	周期性任务	周期性任务周期错误
	%FX131	_ABSD_ER	异常运行导致关机	异常运行导致停止.
	%FX132	_MOTION_CONTRO_WAR	运动控制警告	运动控制功能警告
	%FX134	_ANNUM_WAR	外部设备失效	从外部设备检测一般失效.
%FD7	%FX135	_TASK_PRM_USAGE_OVER_WAR	任务程序占有率超出警告	任务程序占有率超出.
	%FX224	_ERR	计算错误	对于计算错误, 1 扫描期间变为 ON
	%FX227	_ALL_OFF	全部输出 OFF	当全部输出为 OFF, 则变为 ON
	%FX229	_LER	运算错误锁存	运行错误期间保持 0.
	%FX247	_ARY_IDX_ERR	数组索引范围超出	对于数组索引范围超出错误, 1 扫描期间变为 ON
	%FX248	_ARY_IDX_LER	数组索引范围超出锁存	对于数组索引范围超出错误, 1 扫描期间变为 ON
	%FX249	_UDF_STACK_ERR	UDF 堆栈超出	对于 UDF 堆栈超出错误, 1 扫描期间变为 ON
%FX250	_UDF_STACK_LER	UDF 堆栈超出锁存	对于 UDF 堆栈超出错误, 1 扫描期间变为 ON	
%FW202	-	_ANC_ERR	外部设备失效的信息	显示外部设备失效信息
%FW203	-	_ANC_WAR	外部设备一般失效的信息	显示外部设备一般失效信息

双字	位	标志位名称	功能	描述
%FW1282	-	_ANC_ERR	外部设备失效信息	显示外部设备失效信息
%FW1283	-	_ANC_WAR	外部设备一般失效信息	显示外部设备一般失效信息

备注

关于更多全部标志位详情, 参考附录 1 手册概要标志位表.

10.3.5 外部设备故障诊断功能

该功能检测与运动控制器连接的外部设备的失效，以便捷的识别系统警告并停止。通过该功能，可以在不需要复杂程序的情况下检测外部设备失效，在不需要特殊设备(MP500 等)或者程序的情况下识别一般失效位置。

可以如下使用外部设备失效诊断功能。

(1) 外部设备失效类型

- 外部设备失效分为两类；通过用户程序和要求运动控制器运行停止的特殊继电器(F区域)组合检测失效(错误);持续运动控制器运行并仅显示检测状态的一般失效(警告)。

(2) 外部设备检测失效标志位

以下标志位类型用于诊断外部设备失效。

字	位	标志位名称	功能	描述
%FW1282	-	_ANC_ERR	外部设备失效信息	输入外部设备用户自定义严重失效错误代码。
%FW1283	-	_ANC_WAR	外部设备一般失效信息	输入外部设备用户自定义一般失效错误代码。
-	%FX70	_ANNUM_ER	外部严重错误检测	当发生外部设备严重错误时为 On.
-	%FX134	_ANNUM_WAR	外部轻微错误检测	当发生外部设备一般错误时为 On.
-	%FX20482	_CHK_ANC_ERR	请求外部严重错误检测	这是邀请检测外部设备严重失效的指令标志位。
-	%FX20483	_CHK_ANC_WAR	请求外部轻微错误检测	这是邀请检测外部设备一般失效的指令标志位。

(3) 如何检测外部设备严重失效

以下编程用于检测外部设备严重失效。

- 通过如下MOVE指令保存可以通过%FW1282(_ANC_ERR)中外部设备严重失效识别的错误代码。(输入包括0的值)
- 对于外部设备发生严重失效，%FX20482(_CHK_ANC_ERR)标志位为On.
- 当主任务程序完成，运动控制器查看%FX20482 (_CHK_ANC_ERR)是否为ON，并检测严重失效。
- 如果发生外部设备严重失效，运动控制器为错误状态并将停止运行.然后，%FX70 (_ANNUM_ER)变为ON，而%FX20482(_CHK_ANC_ERR)标志位自动为Off. 所有输出基于IO参数紧急输出设置执行。
- 当失效发生时，通过MP500，监控%FW1282 (_ANC_ERR)标志位用户可以找出失效原因。

第 10 章 CPU 功能

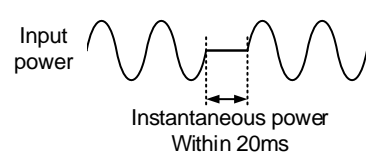
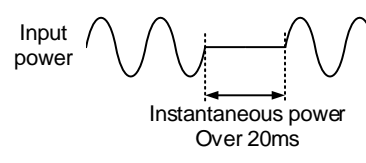
(4) 如何检测外部设备一般失效

以下编程用于检测外部设备一般失效.

- (a) 通过如下MOVE指令保存可以通过%FW1283(_ANC_WAR)中外部设备一般失效识别的错误代码.(输入包括0的值)
- (b) 对于外部设备发生一般失效, %FX20483(_CHK_ANC_ERR)标志位为On.
- (c) 当主任务程序完成, 运动控制器查看%FX20483(_CHK_ANC_ERR)是否为ON, 并检测一般失效.
- (d) 如果发生外部设备一般失效, %FX134(_ANNUM_WAR)变为ON并持续运行. 然后, %FX20483(_CHK_ANC_ERR)自动为Off.
- (e) 当一般失效发生时, 通过MP500, 监控%FW20483(_ANC_WAR)标志位用户可以找出失效原因.
- (d) 如果在移除失效原因后再次输入0到%FW1283(_ANC_WAR), 并再次打开%FX20483(_CHK_ANC_WAR), 取消检测一般失效.

10.3.6 瞬时断电保护功能

当提供运动控制器的输入电压低于标准值时检测瞬时断电.
如果检测到瞬时掉电, 执行以下进程.

断电时间	处理操作
 <p>Input power</p> <p>Instantaneous power Within 20ms</p>	<ol style="list-style-type: none"> 1. 如果首次发生瞬时断电, 内部计时器启动, 执行如之前的运行(不停止). 2. 如果取消瞬时断电(参考时间在 20ms 以内), 内部计时器启动停止, 执行如之前的运行.
 <p>Input power</p> <p>Instantaneous power Over 20ms</p>	<ol style="list-style-type: none"> 1. 如果在首次发生瞬时断电后超过 20ms 无法提供电源, 以电源输入相同方式执行重启运行.

备注

瞬时断电参考通电源条件中运动控制器指定供电电压状态低于允许波动范围, 临时的断电(几到几十 ms)被称为瞬时断电.

10.4 RTC 功能

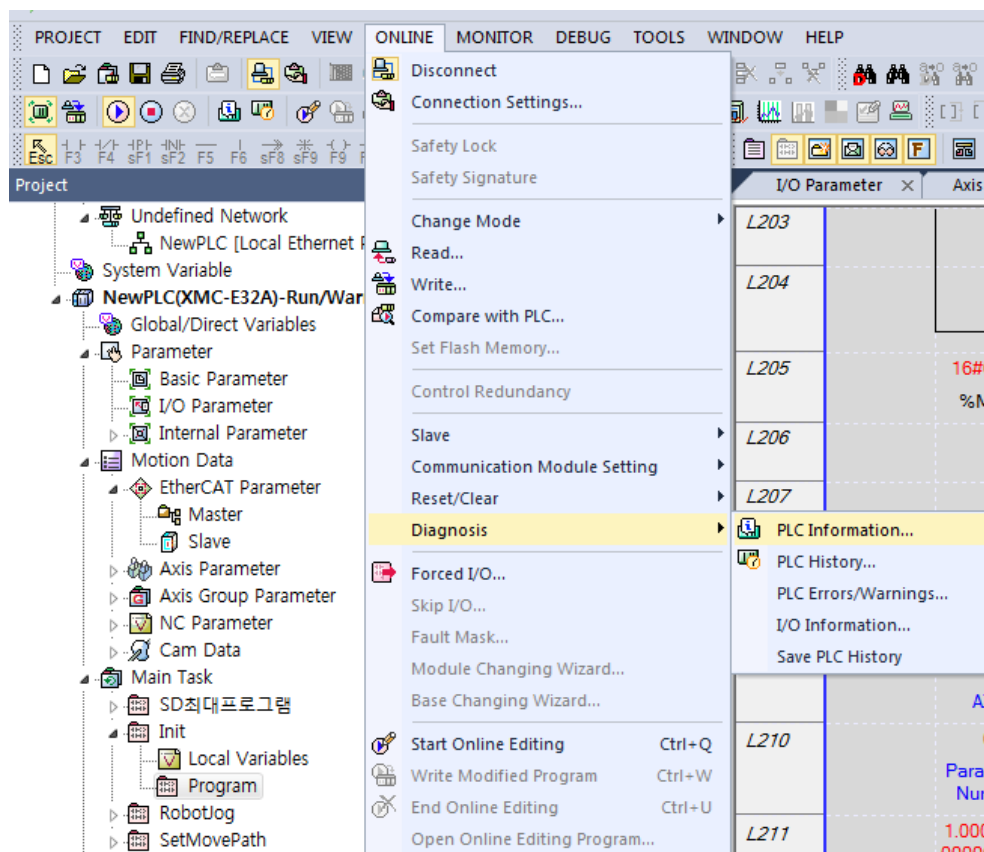
运动控制器具有内置时钟(RTC)功能，可以在电源断电情况下通过电池备份保持运行。内置RTC时间数据用于时间管理，例如系统运行历史或者失效历史等。RTC当前时间通过用于系统运行状态信息每秒进行更新。

10.4.1 如何使用 RTC

(1) 读取/设置时钟数据

(a) 从MP500中读取数据并设置

- 1) 点击『在线』-『诊断』-『运动控制器信息』。
- 2) 点击『运动控制器信息』的运动控制器时钟选项。



- 3) 如果希望发送PC时间到运动控制器，点击‘与PC时钟同步’按钮。

- 4) 如果希望建立用户自定义时间，变更数据设置值和时间对话框之后，点击‘发送到运动控制器’。

(b) 通过特殊继电器读取

如下在以下示例中通过特殊继电器监控数据。

存储	标志位名称	功能	数据	描述
%FB52	_RTC_TIME	RTC 数据[]		
%FB52~%FB59	_RTC_TIME[0]	RTC 数据(年)	h16	2016 年
	_RTC_TIME[1]	RTC 数据(月)	h11	11 月
	_RTC_TIME[2]	RTC 数据(日)	h08	8 th 天

第 10 章 CPU 功能

	_RTC_TIME[3]	RTC 数据(时)	h19	下午 7 点
	_RTC_TIME[4]	RTC 数据(分)	h12	12 分
	_RTC_TIME[5]	RTC 数据(秒)	h54	54 秒
	_RTC_TIME[6]	RTC 数据(星期)	h02	星期二
	_RTC_TIME[7]	RTC 数据(百年)	h20	2000s(十年)
%FW30	_RTC_DATE	RTC 当前日期	2016-11-08	11, 8, 2016
%FW31	_RTC_WEEK	RTC 当前星期	2	星期二
%FD16	_RTC_TOD	RTC 当前时间	19:17:14.345	19:17:14.345

(c) 以下示例为通过程序修改时钟数据

如下用户可以通过程序使用 RTC-SET 功能块建立时钟数据。

功能块	I/O 变量	描述
	REQ	在上升沿执行功能块。
	DATA	输入的时间数据(参考下表.)
	DONE	如果执行进程正常, 则输出 1.
	STAT	如果发生错误, 输出错误代码.

变量	详情	示例	变量	详情	示例
DATA[0]	年	16#16	DATA[4]	分	16#30
DATA[1]	月	16#11	DATA[5]	秒	16#11
DATA[2]	日	16#30	DATA[6]	星期	-
DATA[3]	时	16#12	DATA[7]	世纪	16#20

如果是 12:30:11, 30th, 十一月, 2016, 不需要输入单独的星期数据, 因为日期对应星期的数据自动建立.

(d) 以下示例为通过系统标志位修改时钟数据

通过在以下区域中填写块数据建立时钟数据, 在不使用功能块的情况下打开%FX20480 (_RTC_WR).

存储	标志位名称	描述
%FB2568	_RTC_TIME_USER	时间设置
%FB2568~ %FB2575	_RTC_TIME_USER[0]	时间设置 (年)
	_RTC_TIME_USER[1]	时间设置 (月)
	_RTC_TIME_USER[2]	时间设置 (日)
	_RTC_TIME_USER[3]	时间设置 (时)
	_RTC_TIME_USER[4]	时间设置 (分)
	_RTC_TIME_USER[5]	时间设置 (秒)
	_RTC_TIME_USER[6]	时间设置(星期)
	_RTC_TIME_USER[7]	时间设置(百年)

(e) 如何表达星期

编号	0	1	2	3	4	5	6
星期	星期天	星期一	星期二	星期三	星期四	星期五	星期六

第 10 章 CPU 功能

(2) 时间错误

RTC 错误可能根据常温而不同.

运行温度	最大差异(秒/1 天)	一般情况(秒/1 天)
0°C	-12.26 ~ -1.03	-6.64
25°C	-10.37 ~ 0.86	-4.75
55°C	-13.09 ~ -1.86	-7.47

备注

- 由于产品出厂时无法确定时钟数据，所以在使用前需要建立正确的时钟数据.
 - 如果应用不可用时钟数据到 RTC，则无法正常工作.
- 例) 25:00, 32th, 14 月
- 如果由于电池问题或者错误发生导致 RTC 停止，当输入新时钟数据到 RTC，则错误清除.

10.5 远程功能

在运动控制器中，可以通过模块附属的钥匙开关或者通过通讯变更运行模式。对于远程运行，将基本单元的模式变更开关切换到STOP位置。

(1) 远程运行分类如下。

- 访问 MP500，并通过安装在基本单元的 USB 端口运行
- 当 MP500 连接到基本单元时，可以通过运动控制器的通讯功能运行连接到网络的其他运动控制器。
- 可以通过 HMI 软件等等由专用通讯控制运动控制器的运行状态

(2) 远程RUN/STOP

- 该功能通过外部通讯模块执行RUN/STOP。
- 该便捷功能用于在恶劣环境下使用运动控制器，或者从外部控制面板CPU模块RUN/STOP。

(3) 远程复位

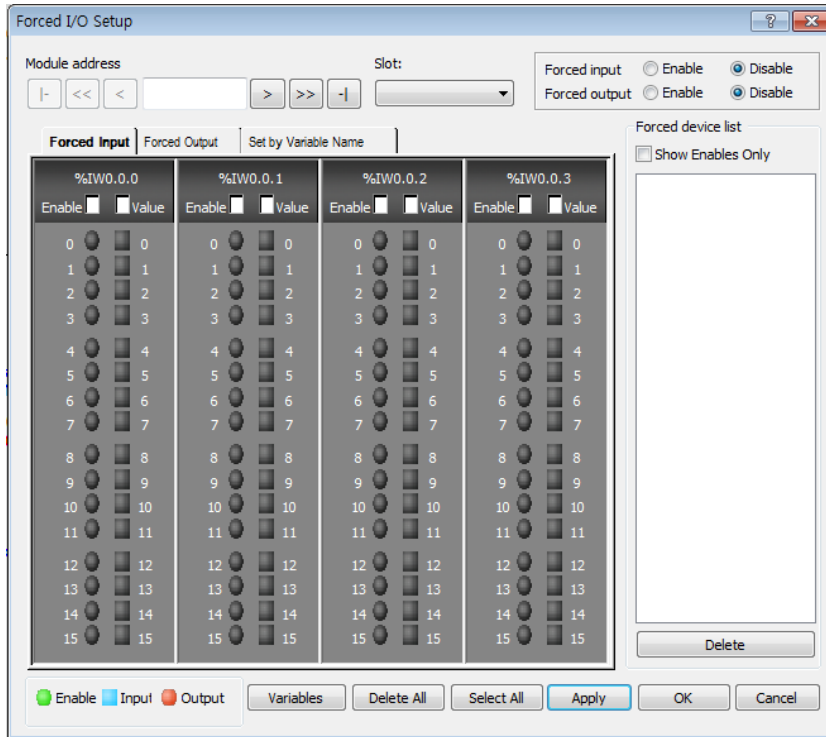
- 当错误发生时，该功能通过远程控制复位CPU模块。
- 可用'复位'和'全部复位'。

10.6 I/O 强制 On/Off 功能

强制 I/O 功能用于打开/关断 I/O 区域，而不管程序执行结果如何。

10.6.1 强制 I/O 设置方式

点击「在线」-「强制 I/O 设置」。



下表代表强制 I/O 设置对应项目。

项目		描述	备注
地址移动		可以选择基板和槽。	
应用		可以设置强制输入和输出使能/禁止。	
个别	标志位	可以设置位强制 I/O 使能/禁止。	
	数据	可以设置位强制 I/O 数据(On/Off)。	
查看变量/内容		可以查看设置输入，输出变量。	
全部选择		在全部 I/O 区域为 On 的条件下可以设置强制 I/O 使能。	
全部删除		在全部 I/O 区域为 Off 的条件下可以删除强制 I/O 使能。	
设置设备		显示 I/O 区域，哪怕只有一个位设置。	

10.6.2 处理强制 I/O On/Off 的时间和处理方式

(1) 强制输入

当强制输入设置时，在更新时从输入模式中读取的数据中，设置为强制 On/Off 的触点数据通过更新到输入图像区域的强制设置数据进行替换。因此，在程序运行中，实际输入数据中，强制设置区域以强制设置数据替换的结果进行运行。

(2) 强制输出

完成用户程序运行后，在输出更新的同时，输出图像区域中的数据包含运行结果，设置为强制On/Off 的触点数据由强制设置数据进行替代，然后，进行输出。因此，与强制输入不同的时，对于强制输出，输出图像区域数据显示与程序运行结果相同的数据，而实际输出变更为强制输出On/Off设置内容。

(3) 使用强制I/O功能的指令

- 在设置强制数据后设置每个I/O‘使能’的时候使用。
- 虽然实际I/O模块不配置，可以设置强制输入。
- 尽管电源从Off → On，运行模式变更，并通过复位键运行。
之前的设置On/Off数据保存在运动控制器中。
- 即使在STOP模式中，不清除强制输入和输出数据。
- 当尝试从开始设置新数据时，在使用前取消所有通过‘全部删除’取消的所有设置。

(4) 错误运行

- 当设置强制输出后发生错误时，基于基本参数中输出控制设置的「错误发生时输出保持」和I/O参数的「紧急输出」工作。对于错误发生，如果在设置「错误发生时输出保持」后选择紧急输出为「清除」，当错误发生时输出off；如果选择「保持」，维持输出状态。

10.7 保存运行历史功能

有 5 种类型的运行历史; 错误历史, 模式转换历史, 断电历史, 系统历史和错误动作历史. 每个事件的发生时间, 频率, 运行详情保存在内存中, 可以通过 MP500 进行监控. 运行历史保存在运动控制器中, 除非通过 MP500 删除.

10.7.1 错误历史

保存在运行期间发生的错误历史.

- 保存错误代码, 日期, 时间, 错误详情.
- 最多可保存历史为 1,024 EA.
- 当由于电池低电压等原因造成内存备份清除时自动取消.

10.7.2 模式转换历史

当变更运行方式时保存变更模式和时间信息.

- 保存数据, 时间, 模式转换信息.
- 历史最多可保存 1,024 EA.

10.7.3 断电历史

电源 On 或者 Off 时间以 ON/OFF 信息进行保存.

- 保存 ON/OFF 信息, 日期和时间.
- 历史最多可保存 2,048 EA.

10.7.4 系统历史

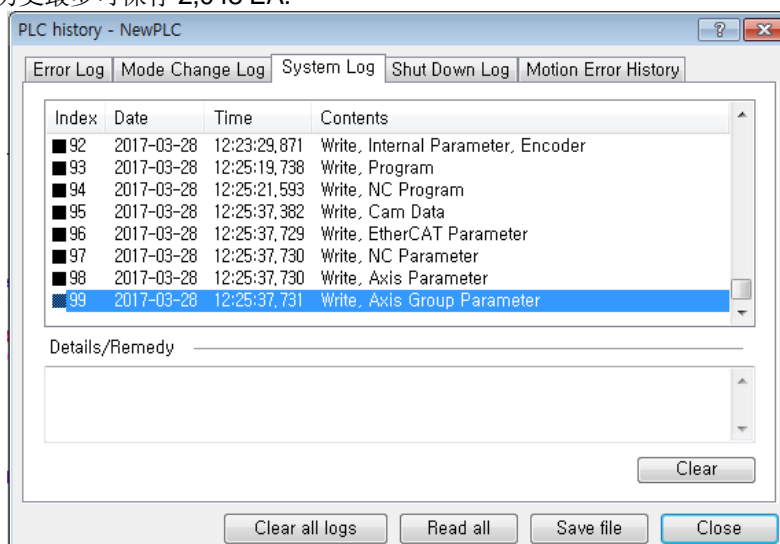
保存运行期间系统发生的运行历史.

- 保存运行变更日期, 时间和详情.
- 保存系统运行相关历史; MP500 运行信息, 钥匙开关位置变更等.
- 历史最多可保存 1,024 EA.

10.7.5 错误动作历史

保存运动控制期间发生的错误历史.

- 保存错误代码, 日期, 时间, 错误详情.
- 保存系统运行相关历史; MP500 运行信息, 钥匙开关位置变更等.
- 历史最多可保存 2,048 EA.



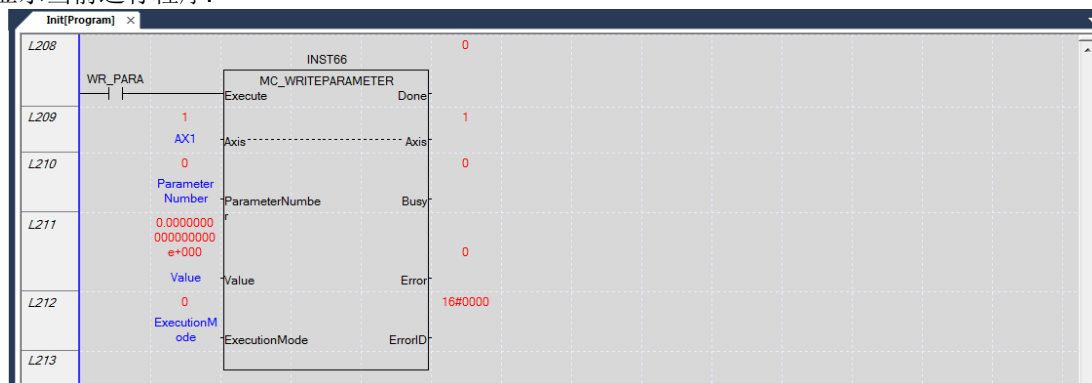
10.8 运行期间程序修改(RUN 期间修改)

可以在运动控制器运行中而不停止控制操作的情况下，修改程序和通讯参数。以下描述基本修改方式。更多内容,参考 MP500手册。

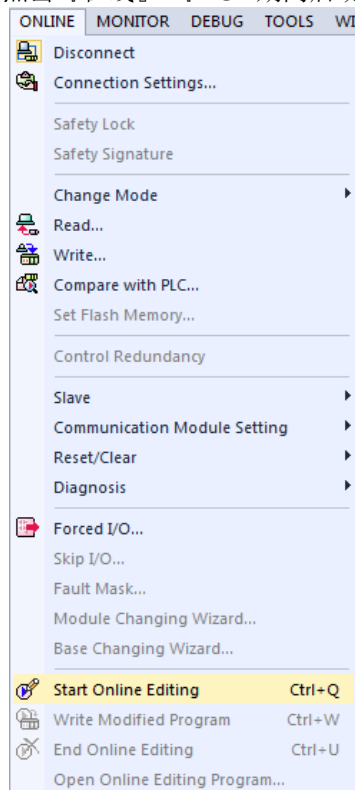
RUN期间可修改项目仅限于程序，网络参数。
无法修改增加任务，删除，参数等。

10.8.1 RUN 期间修改步骤

(1) 显示当前运行程序。

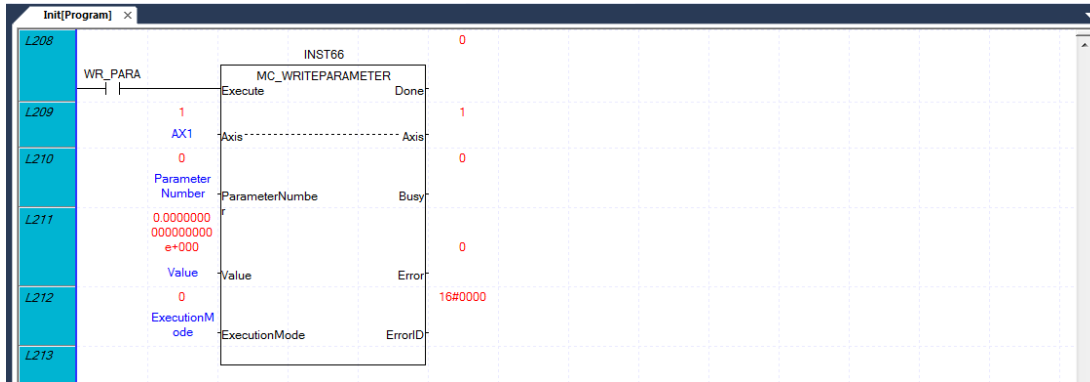


(2) 点击『在线』-『RUN期间启动修改』。

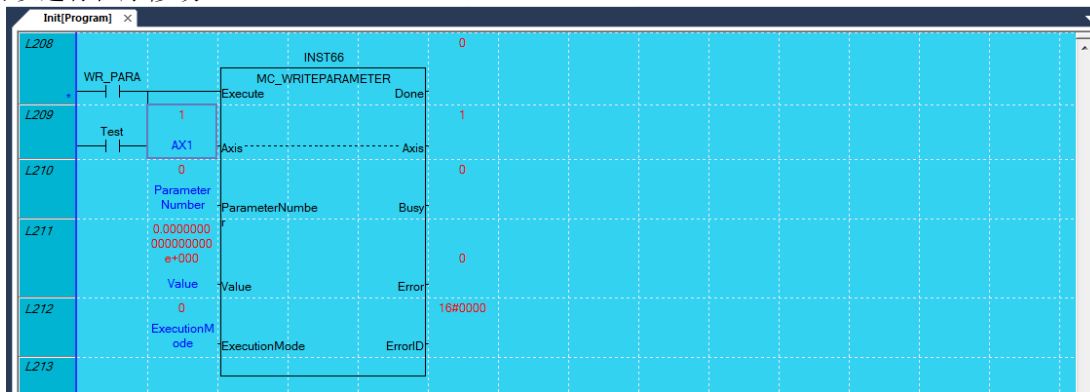


第 10 章 CPU 功能

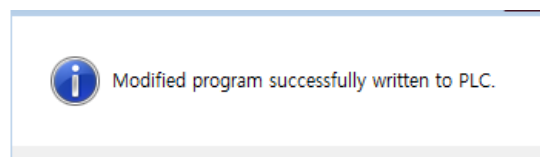
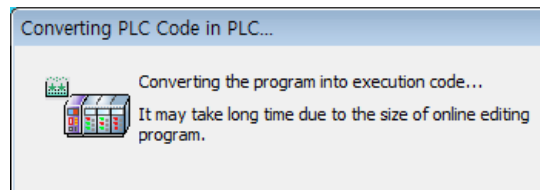
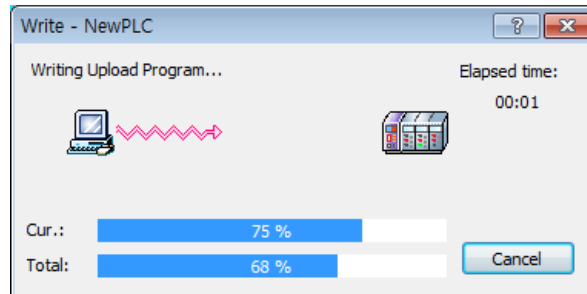
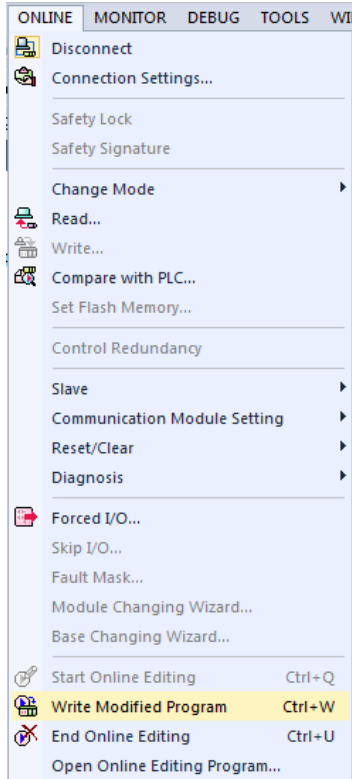
(3) 然后，程序窗口背景色变更，并转换到RUN期间修改模式。



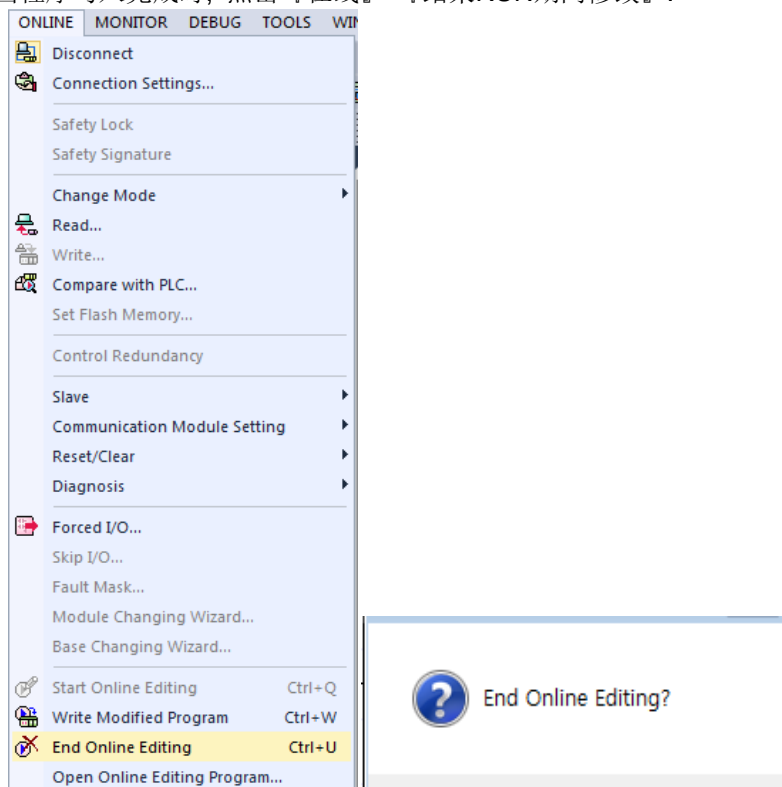
(4) 可以进行程序修改。



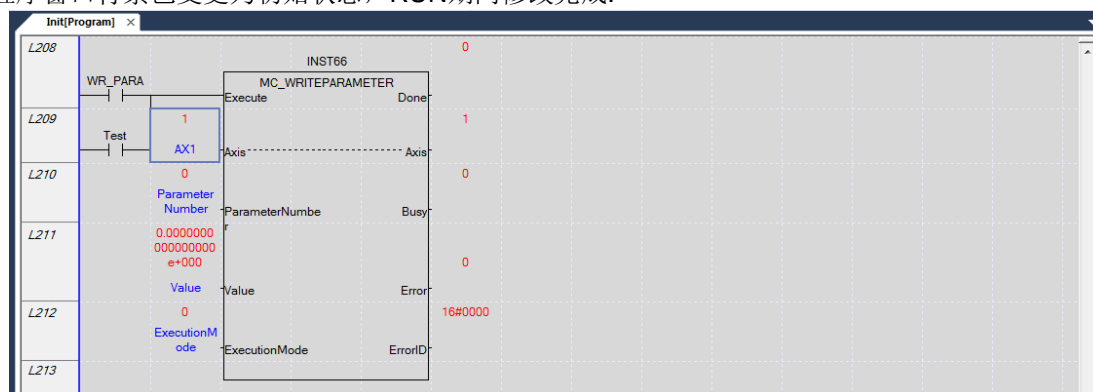
(5) 当完成程序修改时，点击『在线』-『写入RUN期间修改』



(6) 当程序写入完成时, 点击『在线』-『结束RUN期间修改』.



(7) 程序窗口背景色变更为初始状态, RUN期间修改完成.



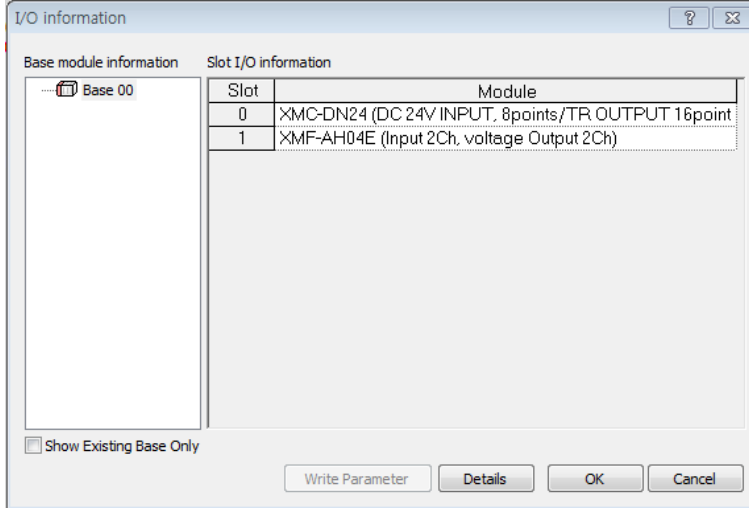
备注

- 对于 RUN 期间通讯参数的修改, 在不进入 RUN 期间修改菜单而变更 MP500 网络配置项目后, 点击『在线』-『写入』然后选择执行写入的‘网络参数’.

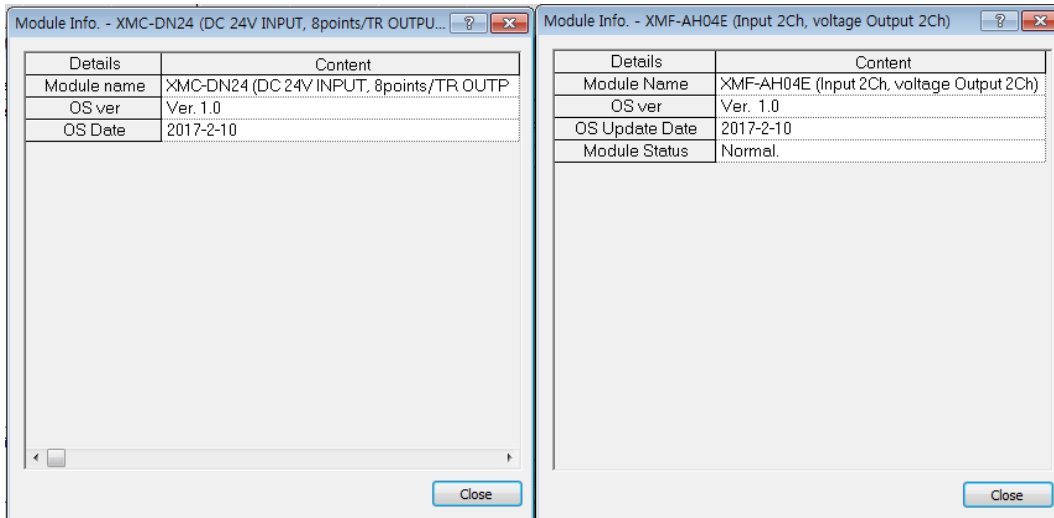
10.9 读取 I/O 信息

该功能用于监控组成运动控制器系统的每个模块的信息。

(1) 如果点击『在线』-『诊断』-『I/O信息』，监控连接到系统的每个模块信息。



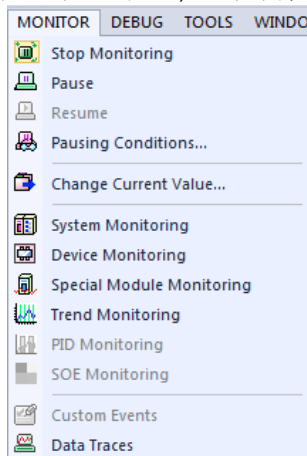
(2) 如果在选择模块后点击‘详情信息’,显示模块详情.



10.10 监控功能

该功能监控运动控制器系统的一般信息.

(1) 如下点击『监控』, 显示子菜单.



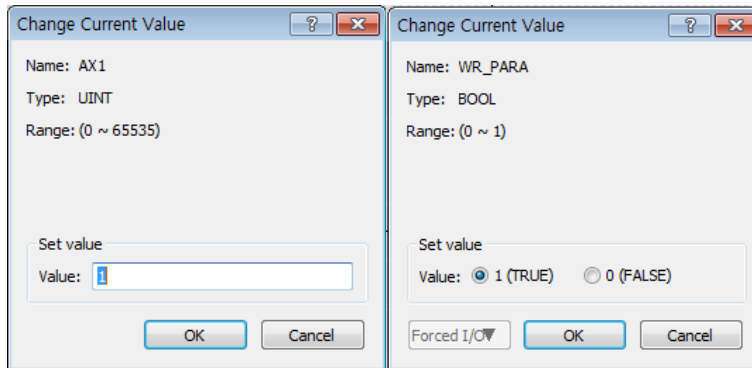
(2) 下表提供每个项目的描述.

项目	描述	备注
启动/结束监控	指定监控的启动和结束.	每次点击时变更
暂停监控	暂停监控.	
重启监控	再次执行暂停的监控.	
监控暂停设置	该功能为当设置设备值与条件匹配时暂停监控.	当点击‘重启监控’时重启
变更当前值	变更当前所选设备的当前值.	
系统监控	监控当前系统的一般信息.	
设备监控	该功能监控每个设备.	
趋势监控	监控设置设备的趋势	更多内容,参考 MP500 手册.
用户事件	当用户指定事件发生时监控设置设备值.	
数据追踪	追踪设置设备值.	

第 10 章 CPU 功能

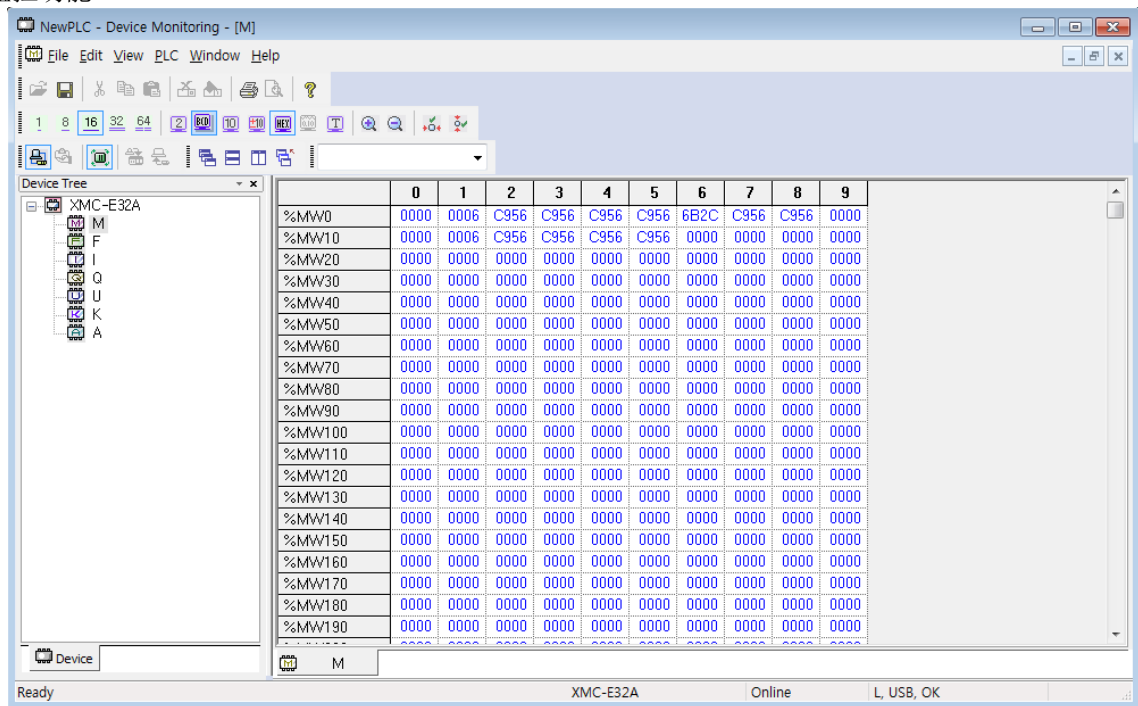
(a) 变更当前值

该功能在程序窗口中变更每个所选设备的当前值。



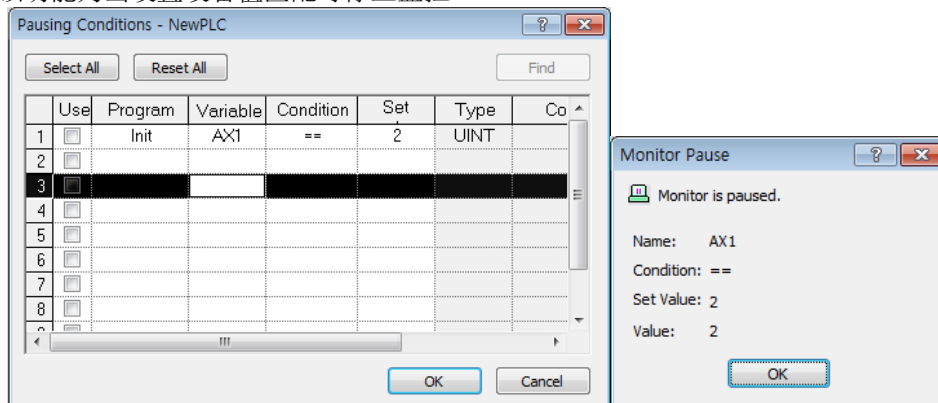
(b) 设备监控

设备监控功能。



(c) 监控暂停设置

该功能为当设置设备值匹配时停止监控。



(d) 趋势监控

该功能以图形形式表示设置设备值. 图表中代表的值不是运动控制器在正确时间收集的数据, 而是通过通讯功能从MP500读取的值. 因此, 可以发生通讯延迟, 在正确周期中无法与实际收集数据匹配. 推荐使用趋势监控功能查看粗略数据趋势.



(e) 数据追踪

该功能收集在用户设置事件的时间内或期望时间内的设备值, 并通过图表或者数据进行监控. 与趋势监控不同的是, 其在用用户设置采样周期内收集实际数据, 用于查看特定点特定时间内的时间数据.

- 1) 数据跟踪条件设置
- 2) 设备设置
- 3) 数据显示

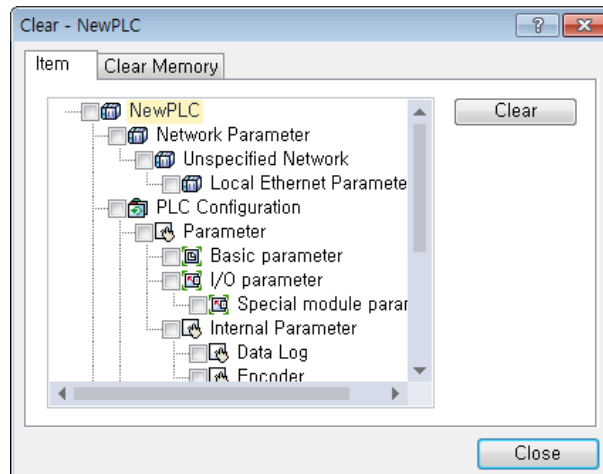
备注

监控功能的更多详情, 参考 MP500 手册.

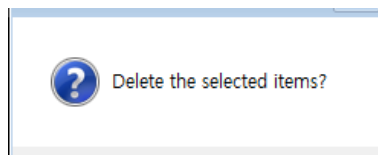
10.11 运动控制器全部删除功能

运动控制器全部删除功能是删除保存于运动控制器中的所有程序,参数,密码,数据的初始化功能.

- (1) 如果删除全部运动控制器
 - (a) 点击『在线』-『复位/清除』-『清除PLC』.



- (b) 如果在对话框中选择『Yes』, 将出现所选运动控制器连接方式删除请求窗口.



- (c) 如果在确认窗口选择“**Yes**”, 数据值清除为“0”. 由于参数可能需要基于数据的默认值, 清除运行后可能发生错误. 如果发生错误, 必须写入作为初始值的数据.

10.12 内置输入/输出功能

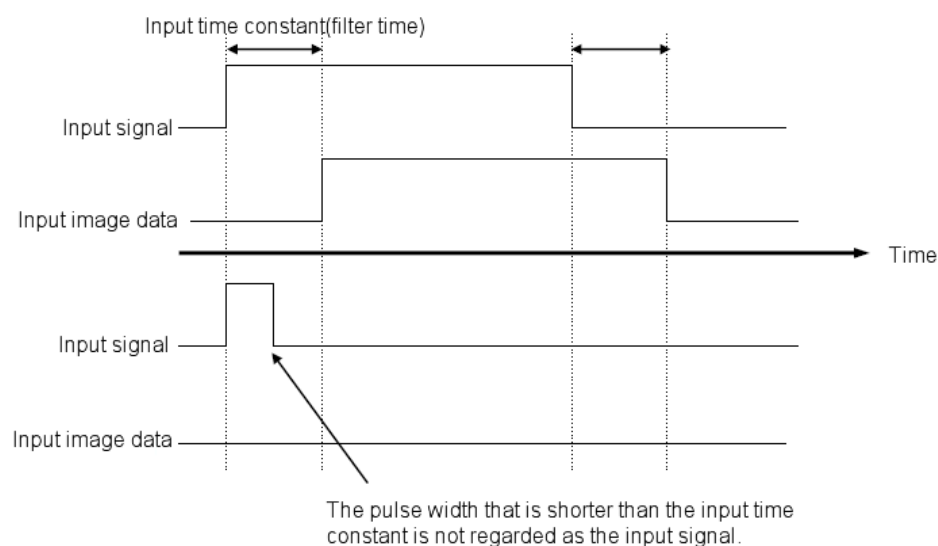
10.12.1 输入滤波功能

运动控制器的输入模块具有输入滤波功能以防止伴随输入信号的外部噪声干扰. 更多输入滤波功能内容,参考如下.

(1) 输入滤波功能的目的是操作

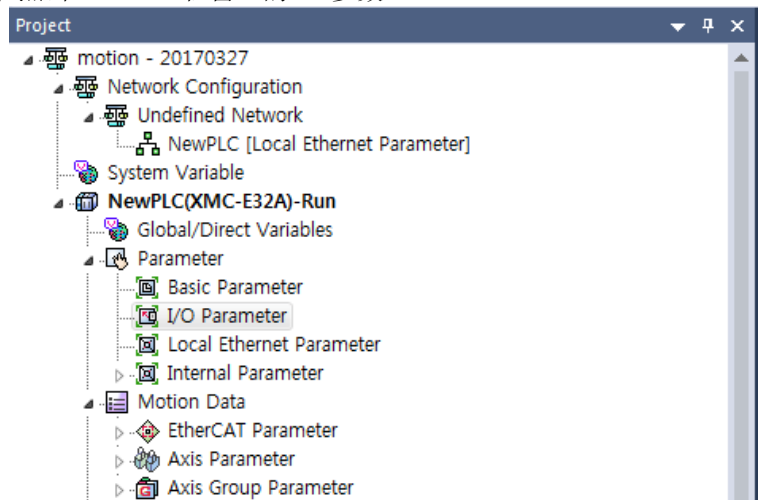
在严重噪声环境或者装备对输入信号脉宽影响较大时,系统可能根据输入信号状态接收到错误输入. 为了防止例如此类的错误输入,输入滤波功能不考虑作为输入并小于用户设置时间的信号. 对于运动控制器,可设置输入滤波时间范围为1ms~100ms.

以下时序图表示输入滤波功能的运行.



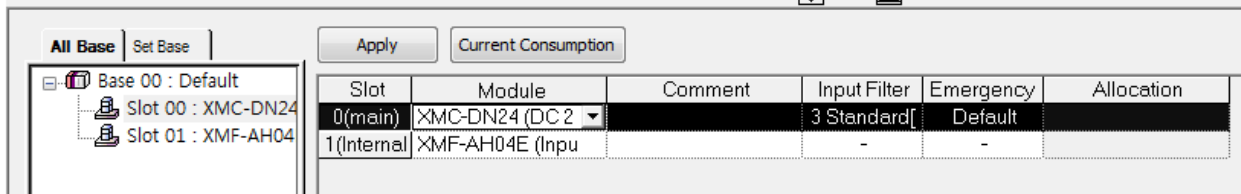
(2) 输入滤波设置方式

(a) 点击MP500工程窗口的‘I/O参数’.

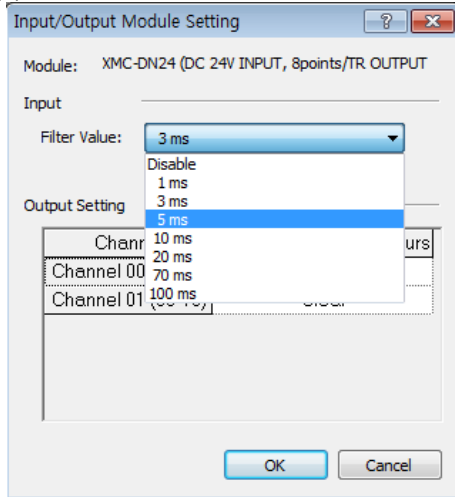


第 10 章 CPU 功能

(b) 选择 I/O 参数设置窗口的‘数字输入/输出(XMC-DN24)’并双击。



(c) 设置滤波值。



备注

当设置滤波值时，主任务周期应设置小于设置滤波值。

例如，如果滤波值设置为 3ms，主任务周期应设置为 1ms 或者 2ms。

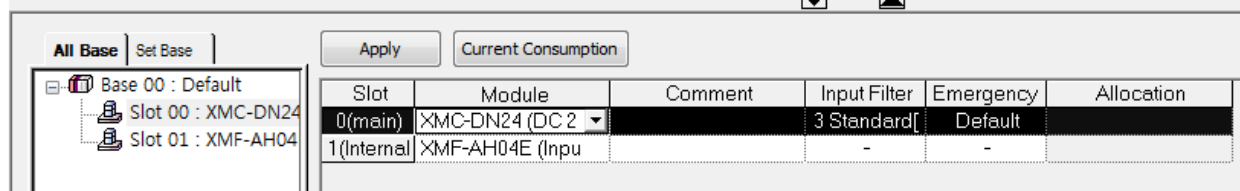
10.12.2 紧急输出功能

MP输出模块支持紧急输出功能，确定当由于错误导致运动控制器停止时，是否维持或者清除输出模块的输出状态。

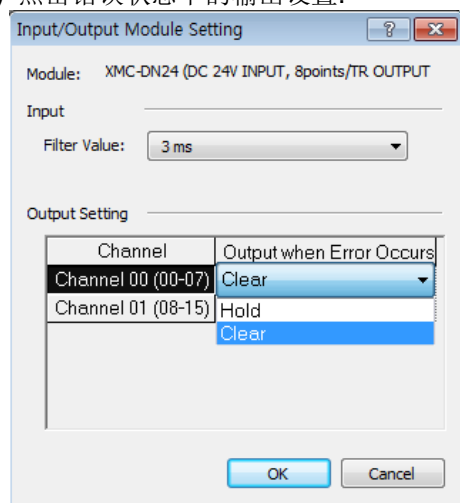
可以设置8位紧急输出。关于设置紧急输出的更多内容，参考如下。

(1) 错误情况的输出状态设置

(a) 在 I/O 参数设置窗口检测‘数字输入/输出(XMC-DN24)’并双击。



(b) 点击错误状态下的输出设置。



如果当错误发生时输出设置选择[清除]，当运动控制器中发生错误导致停止运行时输出关断。如果选择[保持]，则保持输出状态。

10.13 序列号信息读取

该功能为监控运行控制器序列号信息.

(1) 可以通过以下变量监控.

存储	标志位名称	数据	描述
%FB80	_SERIAL_NUM		序列号数据[]
	_SERIAL_NUM[0]	h08	序列号 1~2th 数字
	_SERIAL_NUM[1]	h08	序列号 3~4th 数字
	_SERIAL_NUM[2]	h08	序列号 5~6th 数字
	_SERIAL_NUM[3]	h08	序列号 7~8th 数字
	_SERIAL_NUM[4]	h08	序列号 9~10th 数字
	_SERIAL_NUM[5]	h08	序列号 11~12th 数字
	_SERIAL_NUM[6]	h08	序列号 13~14th 数字
	_SERIAL_NUM[7]	h08	序列号 15~16th 数字
	_SERIAL_NUM[8]	h08	序列号 17~18th 数字
	_SERIAL_NUM[9]	h08	序列号 19~20th 数字
	_SERIAL_NUM[10]	h08	序列号 21~22th 数字
	_SERIAL_NUM[11]	h08	序列号 23~24th 数字
	_SERIAL_NUM[12]	h08	序列号 25~26th 数字
	_SERIAL_NUM[13]	h08	序列号 27~28th 数字
	_SERIAL_NUM[14]	h08	序列号 29~30th 数字
	_SERIAL_NUM[15]	h08	序列号 31~32th 数字
	_SERIAL_NUM[16]	h08	序列号 33~34th 数字
	_SERIAL_NUM[17]	h08	序列号 35~36th 数字
	_SERIAL_NUM[18]	h08	序列号 37~38th 数字
_SERIAL_NUM[19]	h08	序列号 39~40th 数字	

例) 如果序列号为 123456789, 标志位显示如下
(序列号未使用区域显示为 0)

Monitor 1							
	PLC	Program	Variable/Device	Value	Type	Device/Variable	Comment
1	NewPLC	<GLOBAL>	<input type="checkbox"/> _SERIAL_NUM		ARRAY[%FB80	Serial Number
2	NewPLC	<GLOBAL>	_SERIAL_NUM[0]	16#50	BYTE		
3	NewPLC	<GLOBAL>	_SERIAL_NUM[1]	16#07	BYTE		
4	NewPLC	<GLOBAL>	_SERIAL_NUM[2]	16#21	BYTE		
5	NewPLC	<GLOBAL>	_SERIAL_NUM[3]	16#31	BYTE		
6	NewPLC	<GLOBAL>	_SERIAL_NUM[4]	16#BA	BYTE		
7	NewPLC	<GLOBAL>	_SERIAL_NUM[5]	16#20	BYTE		
8	NewPLC	<GLOBAL>	_SERIAL_NUM[6]	16#00	BYTE		
9	NewPLC	<GLOBAL>	_SERIAL_NUM[7]	16#00	BYTE		
10	NewPLC	<GLOBAL>	_SERIAL_NUM[8]	16#00	BYTE		
11	NewPLC	<GLOBAL>	_SERIAL_NUM[9]	16#00	BYTE		
12	NewPLC	<GLOBAL>	_SERIAL_NUM[10]	16#00	BYTE		
13	NewPLC	<GLOBAL>	_SERIAL_NUM[11]	16#00	BYTE		
14	NewPLC	<GLOBAL>	_SERIAL_NUM[12]	16#00	BYTE		
15	NewPLC	<GLOBAL>	_SERIAL_NUM[13]	16#00	BYTE		
16	NewPLC	<GLOBAL>	_SERIAL_NUM[14]	16#00	BYTE		
17	NewPLC	<GLOBAL>	_SERIAL_NUM[15]	16#00	BYTE		
18	NewPLC	<GLOBAL>	_SERIAL_NUM[16]	16#00	BYTE		
19	NewPLC	<GLOBAL>	_SERIAL_NUM[17]	16#00	BYTE		
20	NewPLC	<GLOBAL>	_SERIAL_NUM[18]	16#00	BYTE		
21	NewPLC	<GLOBAL>	_SERIAL_NUM[19]	16#00	BYTE		
22							

第11章数据记录功能

11.1 概要

运动控制是将数据记录功能进行内置。在本章节中对数据记录功能相关的规格及使用方法等进行说明。

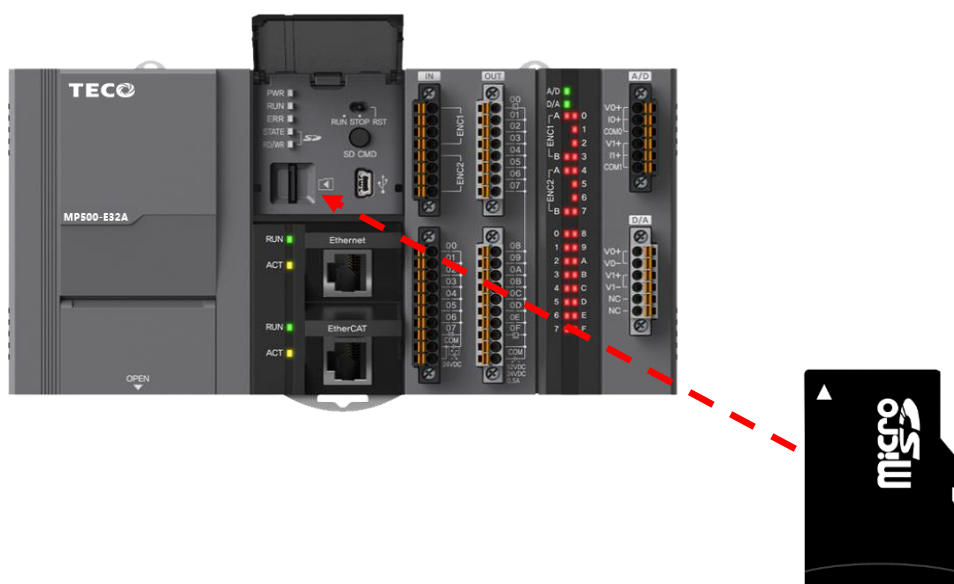
11.1.1 特征

利用运动控制内置数据记录功能仅进行简单的参数设定来收集运动控制的运行数据，可在micro SD 内存卡中按 CSV(逗号分隔值)格式的文件进行保存有以下特征。

(1) 简易内存数据保存

仅通过简单的参数设定，可以对运动控制的各种内数据进行保存。不构成网络也可以收集大容量的运行数据，所以可以进行费用的节俭。

或者利用网络的数据收集方式，将可能发生的通信终端或者缆线松动导致的数据保存失败在原则上清除。



(2) 精密的数据收集

根据主任务, 1ms 单位或者多样的启动条件, 可进行精密的数据收集

或者, 使用触发功能, 触发发生以后的数据值保存后, 使用事件功能时, 发生时间点开始的数据变化可以保存, 所以可以分析简单的系统的运行状态, 可节约系统维护保养费用。

(3) 大容量的运行数据

最大可使用8GB的容量, 所以可以保持长时间的运行数据

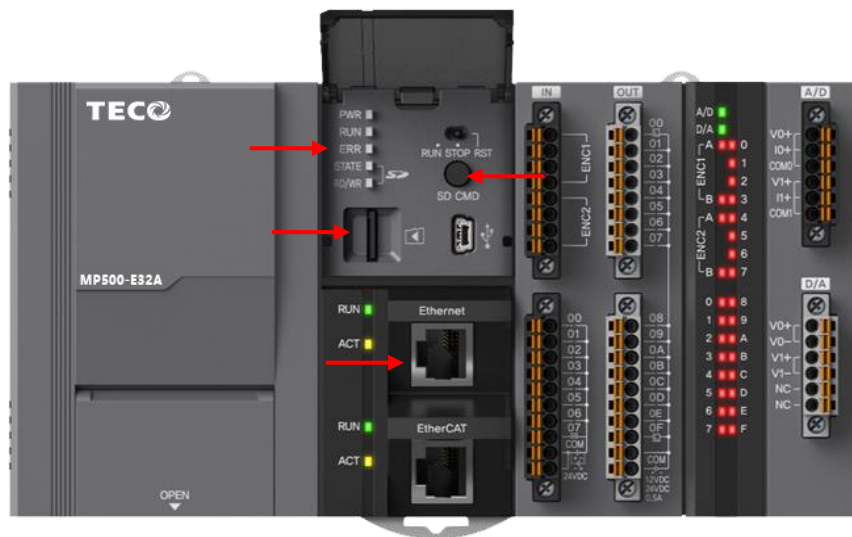
第11章数据记录功能

(4) FTP 联动

数据记录中保存的文件通过 FTP，在远距离也可读取该文件，能更方便的了解数据变化。

11.1.2 各部分的名称

(1) 各部分的名称



	名称	内容
①	状态显示LED	SD内存和数据记录动作状态显示
②	SD内存安装槽	SD 内存安装槽
③	内置 Ethernet端口	利用内置 Ethernet的 FTP功能，传送文件时使用
④	SD CMD BUTTON	SD PWR ON, OFF 或者SD附加功能使用中使用 <ul style="list-style-type: none"> ● 不足1秒~2秒按压: SD 附加功能动作 ● 2秒以上按压: SD PWR ON, OFF

(2) LED 显示

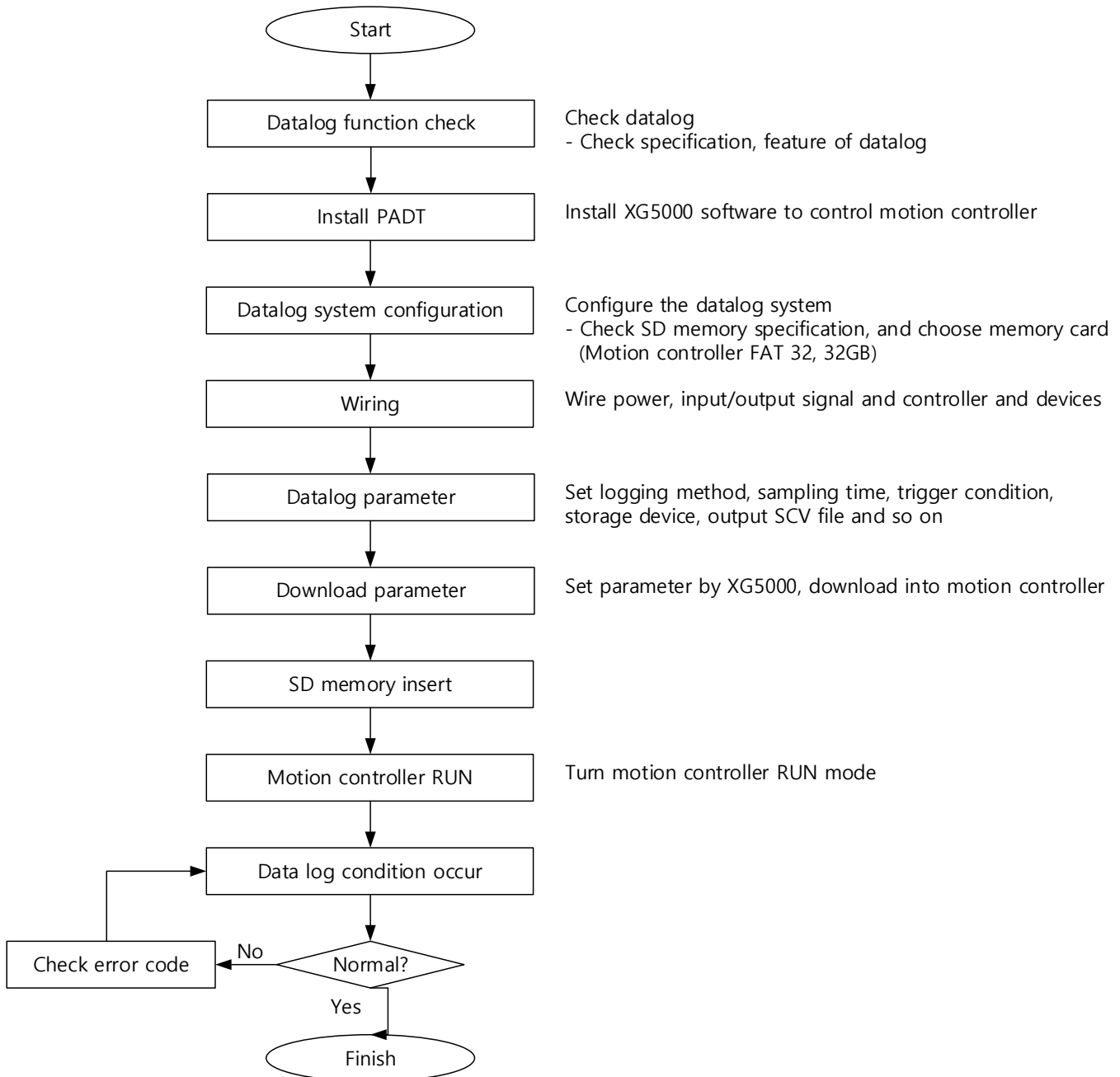


名称	内容	动作规格
PWR	运动控制的供电状态显示	电源提供时灯亮
RUN	运动控制的运行状态显示	RUN灯亮, STOP, ERR时灯灭

名称	内容	动作规格
ERR	运动控制的错误发生状态显示	错误发生时闪烁
STATE	安装的SD内存卡的状态显示	灯亮: SD卡安装, 正常状态 闪烁: SD卡安装, 错误状态 (2s 周期) 灯灭: SD卡分离
RD/WR	SD卡拔出状态显示	闪烁: SD卡读取或者写入中 (100ms 周期) 灯灭: SD卡紧急中止中

11.1.3 动作顺序

数据记录动作顺序如下

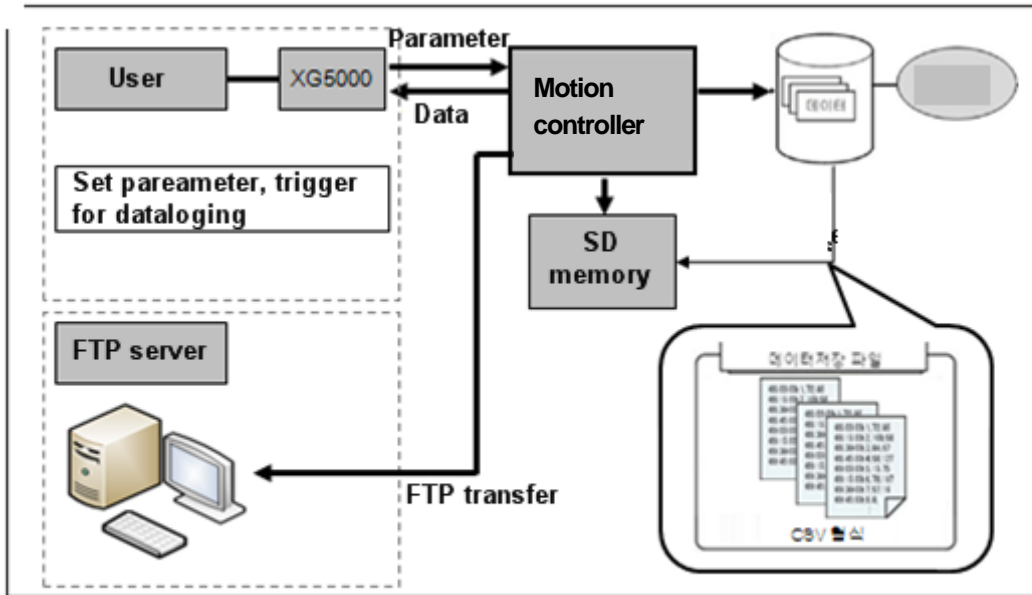


提示

- (1) 运动控制数据记录功能中使用的SD 内存必须是以 FAT32状态格式的状态
- (2) 支持的 SD 内存容量最大32GB

11.1.4 控制信号流程图

数据记录功能是将运动控制的设备值按照如下数据流程，在 SD内存中保存或者外部设备或者和软件交换。



11.2 性能规格

项目		性能规格	备注	
功能设定	组设定	最大 16 组		
	设定数据	组每个最大 64 个		
	数据收集种类	一般保存, 触发保存, 事件保存		
	文件形态	CSV 文件		
	文件大小	最大 16MByte		
	数据类型	BOOL, BYTE, WORD, DWORD, LWORD, SINT, INT, DINT, LINT, USINT, UINT, UDINT, ULINT, REAL, LREAL, STRING		
	保存数据形态	10 进制, 16 进制, 指数, 文字列		
一般保存	采样周期	主任务周期, 指定周期, 指定时间		
	采样对象	每个文件 64 个		
	文件转换	变换时点	文件大小指定: 10 ~ 16,384KB 保存行个数指定: 1,000~50,000 个	
		最大文件数	每个文件夹 256 个	
触发保存	单一条件	位: 上升, 下降 字: 小, 大, 相同, 不同, 大于等于, 小于等于		
	演算条件	AND, OR 条件		
	触发保存范围	最大 69,905 数据		
	文件变换	变换时点	文件大小指定: 10 ~ 16,384KB 保存行个数指定: 1,000~50,000 个	
		最大文件数	每个文件夹 256 个	
事件保存	单一条件	位: ON, OFF, 上升, 下降, 转移 字: 小, 大, 相同, 不同, 大于等于, 小于等于		
	演算条件	AND, OR 条件		
	文件变换	变换时点	文件大小指定: 10 ~ 16,384KB 保存行个数指定: 1,000~32,768 个	
		最大文件数	每个文件夹 256 个	
格式化功能	格式化种类	快速格式化		
	大小	32KByte		
	卷标	LSIS (固定)		

项目		性能规格	备注
SD 内存	提供电压	2.7 ~ 3.6VDC	
	卡尺寸	15mm * 11mm * 1.0mm	
	最大容量	可安装最大容量 : 32GB (8GB 以上的内存仅可使用 8GB)	
	内存类型	micro SD, SDHC (制造商:SanDisk, Transcend)	
	文件系统	FAT 32	

提示

内置数据记录内使用的SD内存推荐SanDisk, Transcend生产的。如果使用其他制造商SD内存, 可能会使预计值不能启动。请留意使用。

11.3 详细功能

所谓数据记录功能，是指运动控制的设备值在选定时间间隔或者触发条件发生时，保存的功能。此时，收集的数据在安装的SD内存卡上以 CSV形态保存。

11.3.1 数据类型和设备

使用运动控制的数据记录功能，可对设备内存进行保存，时钟功能正常动作时，时间也会一起保存。如果，时钟功能未正常动作，时间信息按初始值2000/ 01/01 00:00:00.000进行保存。

(1) 数据类型

可利用运动控制的内置数据记录功能进行保存的数据类型和文字列如下。

数据类型	输出内容	大小(包括BYTE)
BOOL	0 或者 1	2
BYTE	00 ~ FF	3
WORD	0000 ~ FFFF	5
DWORD	00000000 ~ FFFFFFFF	9
LWORD	00000000 00000000 ~ FFFFFFFF FFFFFFFF	17
SINT	-128 ~ 127	5
INT	-32,768 ~ 32,767	7
DINT	-2,147,483,648 ~ 2,147,483,647	12
LINT	-576,460,752,303,423,488 ~ 576,460,752,303,423,487	21
USINT	0 ~ 255	4
UINT	0 ~ 65,535	6
UDINT	0 ~ 4,294,967,295	11
ULINT	0 ~ 1,152,921,504,606,846,975	20
REAL	-3.402823466e+038 ~ -1.175494351e-038 or 0 or 1.175494351e-038 ~ 3.402823466e+038	17
LREAL	-1.7976931348623157e+308 ~ -2.2250738585072014e-308 or 0 or 2.2250738585072014e-308 ~ 1.7976931348623157e+308	24
STRING	固定文字 (最大 32个字)	33

ASCII代码	显示	ASCII代码	显示	ASCII代码	显示	ASCII代码	显示
0x20	SP	0x2A	*	0x3E	>	0x7B	{
0x21	!	0x2B	+	0x3F	?	0x7C	
0x22	“	0x2D	-	0x41 ~ 0x5A	英文(大写)	0x7D	}
0x23	#	0x2E	.	0x5B	[0x7E	~

0x24	\$	0x2F	/	0x5C	\	
0x25	%	0x30 ~ 0x39	숫자	0x5D]	
0x26	&	0x3A	:	0x5E	^	
0x27	'	0x3B	;	0x5F	_	
0x28	(0x3C	<	0x60	`	
0x29)	0x3D	=	0x61~0x7A	英文(小写)	

(2) 可保存设备

利用运动控制的内置数据记录功能可保存的设备如下

数据类型	内容	比较
BOOL	I, Q, M, K, A, F, U	
WORD	I, Q, M, K, A, F, U	

(3) 缓存保存时数据单位计算

内置数据记录中支持的数据基本保存单位是BYTE。因此数据收集时，缓存中堆积的数据计算如下。

(单位: BYTE)

类型	计算单位
BOOL	1
BYTE	1
WORD	2
DWORD	4
LWORD	8
INT	2
SINT	1
DINT	4
LINT	8
UINT	2
USINT	1
UDINT	4
ULINT	8
REAL	4
LREAL	8
STRING	32

(4) 数据变换方法

数据类型变换如下依次收集数据，转换为设定的类型

1) 2word数据(DWORD, DINT, UDINT, REAL)

例) %MW0: 0x1234, %MW1: 0x0000 → 0000 1234 变换

顺序	#2	#1
设备	%MW1	%MW0

2) 4word数据(LWORD, LINT, ULINT, LREAL)

例) %MW0: 0x1234, %MW1: 0x5678, %MW2: 0x000, %MW3: 0x000 → 0000 0000 5678 1234变换

顺序	#4	#3	#2	#1
设备	%MW3	%MW2	%MW1	%MW0

3) 文字列变换方法

- 文字列变换和其他类型不同，最大 32字符保存，一个word变为2字符。

如果，变换时，有0x00 值，该文字列为止变换，之后不转换

例) 16word内无 0x00 → 32 文字

16word内有 0x00 → 0x00 之前为止的文字列变换

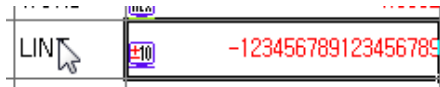
- 文字列变换时,ASCII 不能对应的文字列 (5.3.1 参考)变为 NULL 值。

顺序	#16	#15	#14	...	#1
设备	%MW15	%MW14	%MW13	...	%MW0

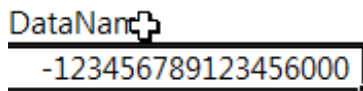
提示

数据类型保存为 LINT，通过 Excel 确认数据后，可能会发生如下无法显示的情况。

实际保存数据



Excel 中确认的数据



这种情况下，通过[记事本]或者[写字板]来确认数据时，可确认正常数据。

提示

如同 REAL 类型，浮点变换支持 IEEE754 规格如下。

位31		位0
符号(S)	指数(E)	浮动小数点部分 (F)

符号(S): 1位

指数(E): 8位

浮动小数点部分(F): 23位

变换值: $(-1)^S \times (1 + F \times 2^{-23}) \times 2^{(E-127)}$

0 < 指数(E) < 255 → 定数

指数(E) = 0, 浮动小数点部分(F) = 0 → 0 (ZERO)

指数(E) = 0, 浮动小数点部分(F) > 0 → 0的近似值变换值

指数(E) = 255, 浮动小数点部分(F) = 0 → 无限大(INFINITY)

指数(E) = 255, 浮动小数点部分(F) > 0 → NAN

提示

REAL, LREAL 类型，对于未定数据超出 -NaN, +NaN 值范围，无限大范围的数据通过 -INF, +INF 进行文字列保存，请确认数据保存范围后使用

11.3.2 数据保存方式

数据记录功能中数据采样的方式如下，共分三点。

(1) 一般保存

一般保存是每个主任务或者每个指定的周期相应时点的数据进行采样的方式。即，不考虑保存条件之前的状态和以后的状态，仅将保存条件发生的时点数据进行采样的方式。此方法是在想收集周期性特定数据时的保存方式

(2) 触发保存

触发保存是满足指定的数据保存条件，该时点之前的数据和之后的数据按照在时点的数据参数中设定的个数进行采样的方式。即，特定事件发生时，想要确认之前日程时间的数据和之后时间的数据时使用的方式。

(3) 事件保存

事件保存是监测收集的设备值，若满足事件条件则那个时点的数据进行采样的功能。即，事件发生时点开始到事件解除为止的数据保存，想要分析事件值的变动和时点时使用的方式

11.3.3 数据采样条件

数据记录功能中数据采样条件及间隔根据参数设定如下区分

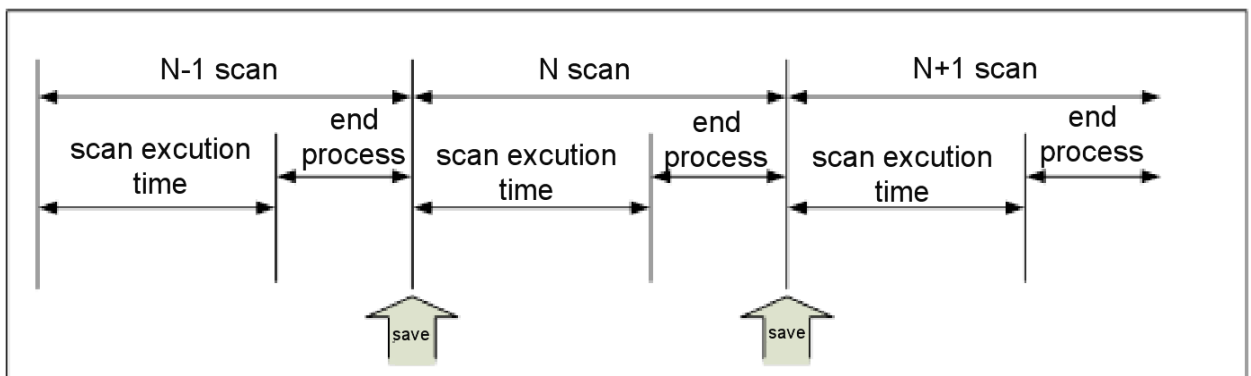
(1) 一般保存

一般保存条件设定项目如下

设定内容	运行	备注
每个主任务周期保存	每个主任务的End处理后，数据采样实现。	
每个指定周期保存	每个指定时间，数据采样实现。	
每个指定时间保存	每个指定时间，数据采样实现。	

1) 每个主任务周期保存

若使用主任务周期保存方式，每个主任务END处理后，进行数据采样。此时数据保存量较多时，主任务周期错误可能会发生，请注意基本参数的主任务周期的设定。



2) 指定周期保存

到指定的周期时，数据采样

3) 指定时间保存

到指定的时间时，数据采样

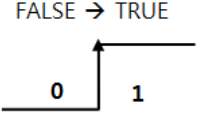
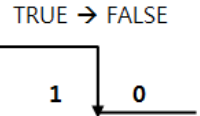


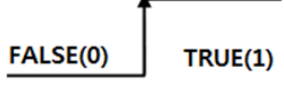
提示

- (1) 数据采样是在参数中每个设定周期中进行的
- (2) 每个组都有缓存领域，日程数据收集完成后，在 SD内存内保存。
- (3) 数据丢失时，DLxx_Ovf标志位为ON

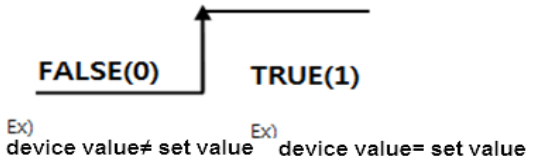
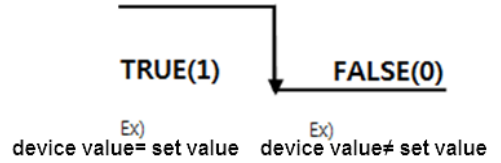
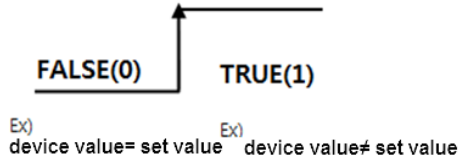
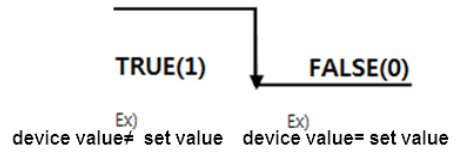
第11章数据记录功能

(2) 触发保存

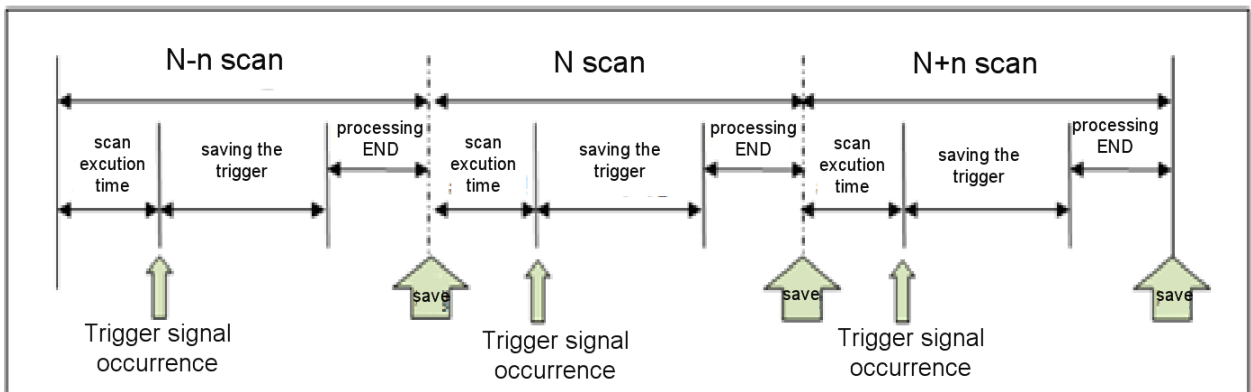
触发条件是触发发生时，按之前设定的采样数据个数保存数据的方式，保存条件设定项目如下。

	触发发生条件	设备设定条件	运行	备注
位条件	上升	X	设定设备位在 FALSE(0) 变为 TRUE(1) 时，数据采样 	
	下降		设定设备位在 TRUE(1) 变为 FALSE(0) 时，数据采样 	
字条件	上升	<	设定字设备值和输入设定值对比时，不满足条件时(FALSE)，变为满足时(TRUE)，数据采样  Ex) device value >= set value Ex) device value < set value device value = set value device value > set value	
	下降		设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE)时数据采样  Ex) device value < set value Ex) device value >= set value device value = set value device value <= set value	
	上升	≤	设定字设备值和输入设定值对比时，不满足条件时(FALSE)，变为满足时 (TRUE)，数据采样。  Ex) device value > set value Ex) device value <= set value	

	触发发生条件	设备设定条件	运行	备注
字条件	下降	\leq	<p>设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE)时，数据采样。</p> <p>Ex) device value \leq set value Ex) device value $>$ set value</p>	
	上升	$>$	<p>设定字设备值和输入设定值对比时，不满足条件时(FALSE)变为满足(TRUE)，数据采样。</p> <p>Ex) device value \leq set value Ex) device value $>$ set value device value $=$ set value device value $>=$ set value</p>	
	下降		<p>设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE)时，数据采样。</p> <p>Ex) device value $>$ set value Ex) device value $>=$ set value</p>	
	上升	\geq	<p>设定字设备值和输入设定值对比时，不满足条件时(FALSE)变为满足(TRUE)，数据采样。</p> <p>Ex) device value $<$ set value Ex) device value $>=$ set value</p>	
	下降		<p>设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE)时，数据采样。</p> <p>Ex) device value $>=$ set value Ex) device value $<$ set value</p>	

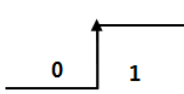
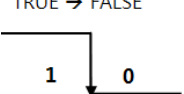
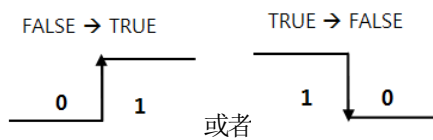
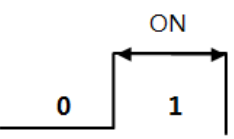
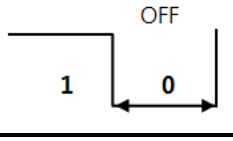
	触发发生条件	设备设定条件	运行	备注
字条件	上升	相同	设定字设备值和输入设定值对比时，不满足条件时(FALSE)变为满足(TRUE)，数据采样。 	
	下降		设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE)时，数据采样。 	
	上升	不同	设定字设备值和输入设定值对比时，不满足条件时(FALSE)变为满足(TRUE)，数据采样。 	
	下降		设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE)时，数据采样。 	

触发发生与否是主任务 END判断的，数据采样中设定的触发再次发生时，该触发无视，继续数据采样。



(3) 事件保存

事件保存条件是在事件发生时, 到条件不成立为止, 数据保存的方式。发生条件如下

	事件发生条件	设备设定条件	运行	解除值设定与否
位条件	上升		设定设备位从 FALSE(0) 变为 TRUE(1) 时, 数据采样 FALSE → TRUE 	
	下降		设定设备位从 TRUE(1) 变为 FALSE(0) 时, 数据采样 TRUE → FALSE 	
	转移		设定设备位从 TRUE(1) 变为 FALSE(0) 或者 TRUE(1) 变为 FALSE(0) 时 数据采样 FALSE → TRUE TRUE → FALSE 或者 	
	ON		设定设备位在 TRUE(1) 期间, 数据采样. ON 	
	OFF		设定设备位在 FALSE(0) 期间, 数据采样. OFF 	

	事件发生条件	设备设定条件	运行	解除值设定与否
字条件	上升		<p>设定字设备值和输入设定值对比时，不满足条件时(FALSE)变为满足(TRUE)，数据采样。</p>	设定变量
	下降		<p>设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE)时，数据采样。</p>	
	转移	<	<p>设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE) 或者从不满足(FALSE)变为满足(TRUE)时，数据采样。</p>	
	ON		<p>设定字设备值和输入设定值对比时，条件满足(TRUE)期间数据采样。</p>	
	OFF		<p>设定字设备值和输入设定值对比时，不满足条件(FALSE) 期间数据采样。</p>	

	事件发生条件	设备设定条件	运行	解除值设定与否
字条件	上升	≤	<p>设定字设备值和输入设定值对比时，不满足条件时(FALSE)变为满足(TRUE)，数据采样。</p>	
	下降		<p>设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE)时，数据采样。</p>	
	转移		<p>设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE)或者不满足(FALSE)变为满足(TRUE)时，数据采样。</p>	
	ON		<p>设定字设备值和输入设定值对比时，满足条件(TRUE)期间数据采样。</p>	
	OFF		<p>设定字设备值和输入设定值对比时，不满足条件(FALSE)期间数据采样。</p>	

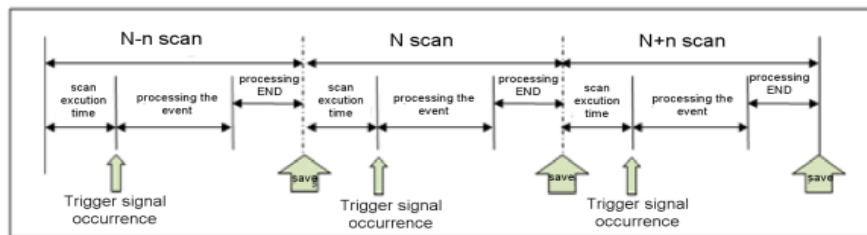
	事件发生条件	设备设定条件	运行	解除值设定与否
字条件	上升		<p>设定字设备值和输入设定值对比时，不满足条件时(FALSE)变为满足(TRUE)，数据采样。</p> <p>Ex) device value \geq set value device value = set value device value $>$ set value</p> <p>Ex) device value $<$ set value</p>	
	下降		<p>设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE)时，数据采样。</p> <p>Ex) device value $<$ set value</p> <p>Ex) device value \geq set value device value = set value device value \leq set value</p>	
	转移	>	<p>设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE)或者不满足(FALSE)变为满足(TRUE)时，数据采样。</p> <p>Ex) device value \geq set value device value = set value device value $>$ set value</p> <p>Ex) device value $<$ set value</p> <p>或者</p> <p>Ex) device value $<$ set value</p> <p>Ex) device value \geq set value device value = set value device value \leq set value</p>	
	ON		<p>设定字设备值和输入设定值对比时，满足条件(TRUE)期间数据采样。</p> <p>Ex) Device value \geq Set value Device value = Set value Device value $>$ Set value</p> <p>Ex) Device value $<$ Set value</p>	
	OFF		<p>设定字设备值和输入设定值对比时，不满足条件(FALSE)期间数据采样。</p> <p>Ex) Device value $<$ Set value</p> <p>Ex) Device value \geq Set value Device value = Set value Device value $>$ Set value</p>	

	事件发生条件	设备设定条件	运行	解除值设定与否
字条件	上升		<p>设定字设备值和输入设定值对比时，不满足条件时(FALSE)变为满足(TRUE)，数据采样。</p> <p>Ex) device value >= set value device value = set value device value > set value</p> <p>Ex) device value < set value</p>	
	下降		<p>设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE)时，数据采样。</p> <p>Ex) device value < set value</p> <p>Ex) device value >= set value device value = set value device value <= set value</p>	
	转移	≥	<p>设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE) 或者不满足(FALSE)变为满足(TRUE)时，数据采样。</p> <p>Ex) device value >= set value device value = set value device value > set value</p> <p>Ex) device value < set value</p> <p>或者</p> <p>Ex) device value < set value</p> <p>Ex) device value >= set value device value = set value device value <= set value</p>	
	ON		<p>设定字设备值和输入设定值对比时，满足条件(TRUE)期间数据采样。</p> <p>Ex) Device value ≥ Set value Device value = Set value Device value > Set value</p> <p>Ex) Device value < Set value</p>	
	OFF		<p>设定字设备值和输入设定值对比时，不满足条件(FALSE) 期间数据采样。</p> <p>Ex) Device value < Set value</p> <p>Ex) Device value ≥ Set value Device value = Set value Device value > Set value</p>	

	事件发生条件	设备设定条件	运行	解除值设定与否
字条件	上升	相同	设定字设备值和输入设定值对比时，不满足条件时(FALSE)变为满足(TRUE)，数据采样。 <p>Ex) device value >= set value device value = set value device value > set value</p> <p>Ex) device value < set value</p>	
	下降		设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE)时，数据采样。 <p>Ex) device value < set value</p> <p>Ex) device value >= set value device value = set value device value <= set value</p>	
	转移		设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE)或者不满足(FALSE)变为满足(TRUE)时，数据采样。 <p>Ex) device value >= set value device value = set value device value > set value</p> <p>Ex) device value < set value</p> <p>或者</p> <p>Ex) device value < set value</p> <p>Ex) device value >= set value device value = set value device value <= set value</p>	
	ON		设定字设备值和输入设定值对比时，满足条件(TRUE)期间数据采样。 <p>Ex) Device value ≥ Set value Device value = Set value Device value > Set value</p> <p>Ex) Device value < Set value</p>	
	OFF		设定字设备值和输入设定值对比时，不满足条件(FALSE)期间数据采样。 <p>Ex) Device value < Set value</p> <p>Ex) Device value ≥ Set value Device value = Set value Device value > Set value</p>	

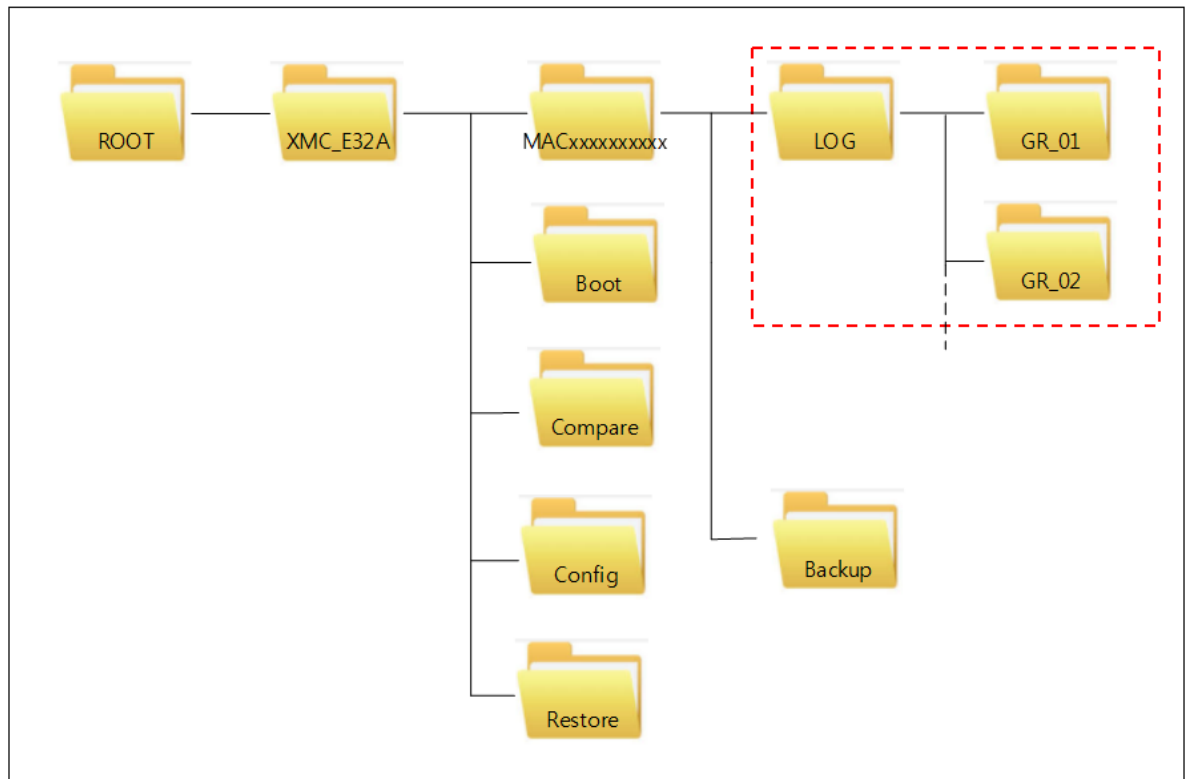
	事件发生条件	设备设定条件	运行	解除值设定与否
字条件	上升	不一致	设定字设备值和输入设定值对比时，不满足条件时(FALSE)变为满足(TRUE)，数据采样。 	
	下降		设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE)时，数据采样。 	
	转移		设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE)或者不满足(FALSE)变为满足(TRUE)时，数据采样。 	
	ON		设定字设备值和输入设定值对比时，满足条件(TRUE)期间数据采样。 	
	OFF		设定字设备值和输入设定值对比时，不满足条件(FALSE)期间数据采样。 	

若使用事件保存方式，指定事件条件由主任务END判断。事件保存在条件发生后，每个主任务数据采样。



11.3.4 保存文件夹构成

数据记录中保存的文件如下文件构成保存



- (1) 文件夹名: 文件夹名是固定的。如果, SD内存内不是如图构成, 而是生成另外的文件夹, 则数据记录功能不能正常运行。这点请注意
- (2) 数据保存文件夹: 根据数据记录生成的记录数据所保存的文件夹。各参数设定组使用另外的文件夹。生成的文件名如下规格。数据文件夹名英为基准, 最多32字(禁止空格)。(文件夹构成图中明确指出的文件夹名是任意名称。可根据用户的设定来变更)

11.3.5 CSV 文件格式

数据记录功能中生成的 CSV文件如下，规格不同

项目名称	内容
区分文字	(,)
换行代码	CR, LF(0x0D, 0x0A)
文字代码	ASCII代码
字段数据	10进制, 16进制, 指数, 文字列
文件大小	最大 16Mbyte

	A	B	C	D	E	F
1	Remark	Project = NewPLC				
2	Remark	Filename = FILE0001.CSV				
3	Remark	Start Date = 2016/11/09/16:27:37.267				
4	Remark	Controller = XMC-E32A				
5	Remark	LogType = Trigger				
6	Remark	DataType	INT	WORD	INT	WORD
7	Remark	Device	mw1	mw2	mw5	mw6
8						
9	TIME	INDEX	DataName	DataName	DataName	DataName
10	2016/11/09/16:27:37.267	441	20521	h'1029	4137	h'1029
11	2016/11/09/16:27:37.271	442	20522	h'102A	4138	h'102A

(1) 头文件构成

SD内存中保存的数据记录文件的头构成如下.

备注	工程名称
备注	保存文件名称
备注	文件生成时间
备注	运动控制种类
备注	数据记录保存类型
备注	数据变换类型
备注	设备

第11章数据记录功能

(2) 数据文件构成

SD内存中保存的数据记录文件的内部构成如下。

First data row	Data save time	Index	Data0	Data1	Data2	...	Data63
Second data row	2016/12/16/10:06:39.977	0	1	1	1	...	xx
Third data row	2016/12/16/10:06:39.978	1	1	1	1	...	xx
...
Last data row	2016/12/xx/xx:xx:xx.xxx	65535	x	x	x	...	xx

Parameter	Group 0	
<input type="checkbox"/> Group Settings	Not used	
Data collection mode	General	
Save Settings	Setting	
<input type="checkbox"/> Buffer size	128KB	
Data 0	<input type="checkbox"/> Type	NONE
	<input type="checkbox"/> Name	DataName
	Device	
Data 1	<input type="checkbox"/> Type	NONE
	<input type="checkbox"/> Name	DataName
	Device	
Data 2	<input type="checkbox"/> Type	NONE
	<input type="checkbox"/> Name	DataName
	Device	
Data 3	<input type="checkbox"/> Type	NONE
	<input type="checkbox"/> Name	DataName
	Device	

	A	B	C	D	E
1	Remark	Project = NewPLC			
2	Remark	Filename = FILE00000.CSV			
3	Remark	Start Date = 2016/12/16/10:06:39.977			
4	Remark	Controller = XMC-E32A			
5	Remark	LogType = Scan			
6	Remark	DataType BOOL	BOOL	BOOL	
7	Remark	Device	MX0	MX1	MX2
8					
9	TIME	INDEX	DATA0	DATA1	DATA2
10	2016/12/16/10:06:39.977	0	1	1	1
11	2016/12/16/10:06:39.978	1	1	1	1
12	2016/12/16/10:06:39.979	2	0	0	0
13	2016/12/16/10:06:39.980	3	0	0	0
14	2016/12/16/10:06:39.981	4	0	0	0
15	2016/12/16/10:06:39.982	5	0	0	0

提示

- (1) 索引在保存数据行显示数字
- (2) 数据0, 数据1, ..., 数据63显示数据名称

提示

- (1) Microsoft Office Excel中CSV 文件读取时, 多个数据额在1个cell中显示, 在本Excel中打开 CSV 文件时, “符号可作为文本限定字来使用。此时, 按以下顺序打开 CSV 文件, 可正常显示
 - (1) excel执行后, 数据菜单中选择text后, 选择要打开的CSV 文件
 - (2) 文本向导程序 2阶段中选择区分记号的休止符, text(无)后, 点击终止

(3) 数据文件项目说明

1) 先行数据行

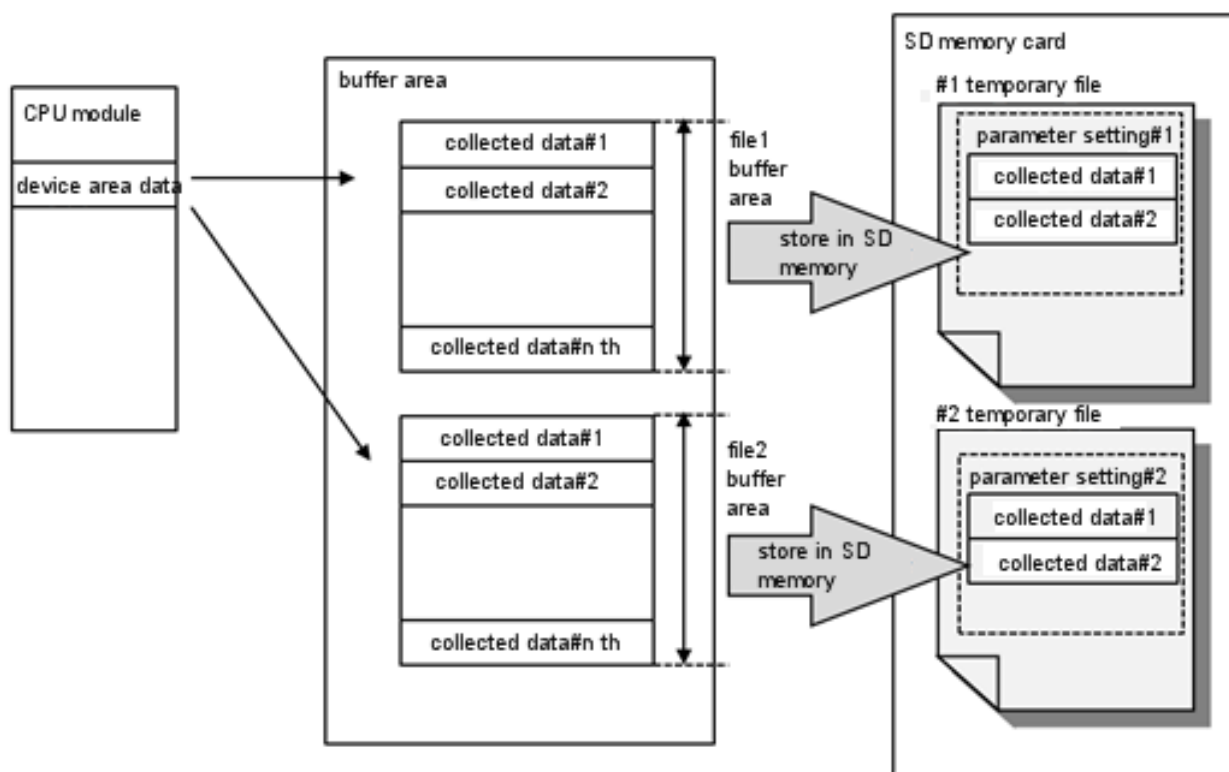
列名	输出内容	大小(Word)
日期列	用固定的文字进行日期及时间的显示	5
索引列	索引名出现	2
数据列	数据设定中指定的「数据名」输出	1~64(根据参设定)

2) 数据行重复

列名	输出内容	大小(Byte)	
日期列	根据CSV 输出设定中设定的数据输出格式来输出。 例) 2014/09/17 10:15:20:243	24	
索引列	从0 开始, 按递增排序进行计数, 输出数值	10	
数据列	BOOL	0 或者 1	2
	BYTE	00 ~ FF	3
	WORD	0000 ~ FFFF	5
	DWORD	00000000 ~ FFFFFFFF	9
	LWORD	00000000 00000000 ~ FFFFFFFF FFFFFFFF	17
	SINT	-128 ~ 127	5
	INT	-32,768 ~ 32,767	7
	DINT	-2,147,483,648 ~ 2,147,483,647	12
	LINT	-576,460,752,303,423,488 ~ 576,460,752,303,423,487	21
	USINT	0 ~ 255	4
	UINT	0 ~ 65,535	6
	UDINT	0 ~ 4,294,967,295	11
	ULINT	0 ~ 1,152,921,504,606,846,975	20
	REAL	-3.402823466e+038 ~ -1.175494351e-038 or 0 or 1.175494351e-038 ~ 3.402823466e+038	17
LREAL	-1.7976931348623157e+308 ~ -2.2250738585072014e-308 or 0 or 2.2250738585072014e-308 ~ 1.7976931348623157e+308	24	
STRING	固定的文字 (最大 32字符)	33	

11.3.6 CSV 文件保存方法

运动控制在每次采样条件发生时，都会收集数据，SD 内存内以CSV文件进行保存。
满足文件转换时点，新文件生成，进行数据保存。



(1) 文件转换时点
CSV文件转换的时点如下。

保存时点	设定范围
文件中指定的个数的保存完成	1000 ~ 50,000
文件指定的大小达成	10KB ~ 16,384KB

(2) 超出文件保存个数时运行
超出参数中设定的该组的最大文件保存个数时，如下文件超出时，根据运行设定来运行。

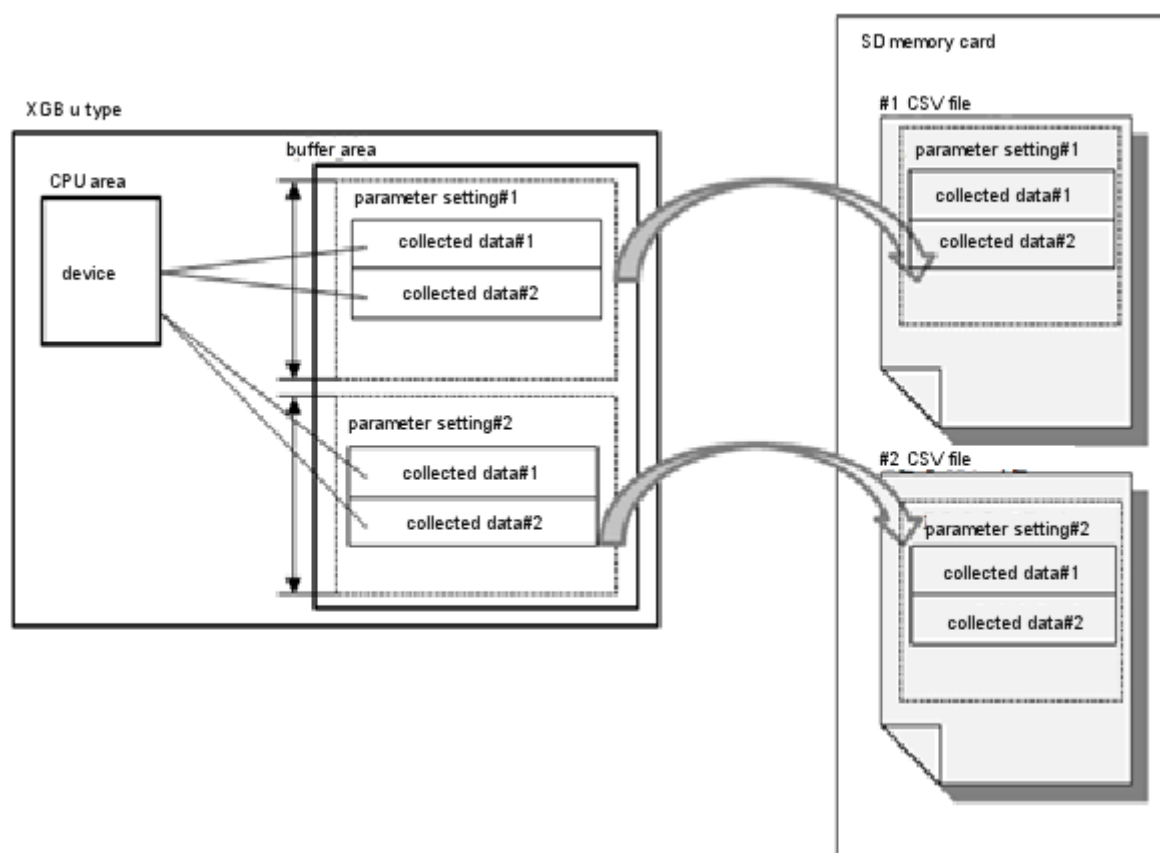
超出时运行设定	运行	备注
近期履历覆盖	在之前的文件上进行新数据覆盖后保存	
最初履历维持	不再进行文保存	

提示

对于SD存储无法保存256个文件并且存储已满的情况，维持初始化历史保存文件直到存储已满，忽略参数中的[历史设置]和相关错误代码6(%KW522)，而停止保存

11.3.7 缓存内存

运动控制为了数据记录功能，在运动控制内部维持缓存内存。所谓缓存内存说收集的数据在SD内存的临时文件中保存前，临时保管的挥发性内存。



根据指定的采样条件，收集的数据先在缓存内存中保存，数据记录条件发生时，在SD内存卡的临时内存中保存。设定较快的数据采样条件或者较多的数据进行采样时，因为超出缓存内存，可能会发生数据丢失，所以请注意。发生数据丢失时，丢失数据中‘C’文字列一起保存。此时，调整数据记录参数中保存周期或者减少收集数据量，尽量避免数据的发生。

11.3.8 数据丢失

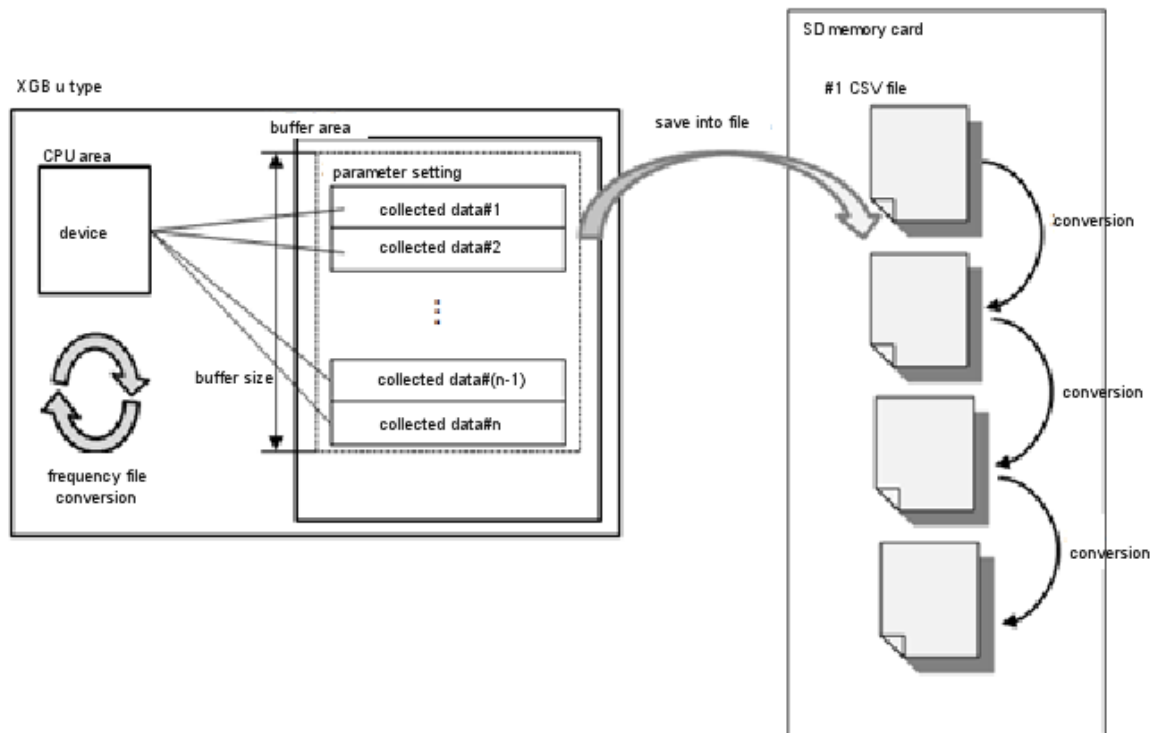
所谓数据丢失是指不能正常地进行数据收集。数据收集间隔设定为很短的周期时，每个指定的周期会发生无法数据采样的情况，因此也会发生数据丢失。代表情况如下。

(1) 缓存超出

设定较短的周期，或者大量的数据收集，缓存内存的值在SD内存卡的临时文件中保存的速度比起内部缓存内存中数据收集的速度慢，超出缓存的保存容量，会发生数据的丢失

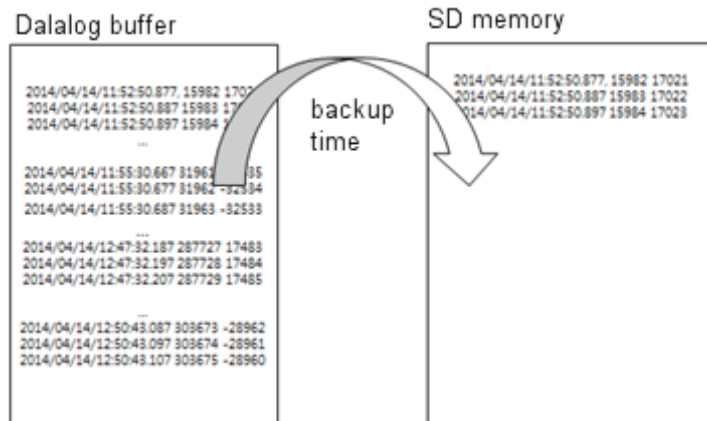
(2) 频繁文件转换

文件转换条件发生时，临时文件转换为 CSV文件后输出，因为必须要生成新临时文件，此期间缓存内存的值不能变为临时文件进行保存。因此文件转换条件经常发生时，超出缓存内存的保存容量，可能发生数据的丢失。

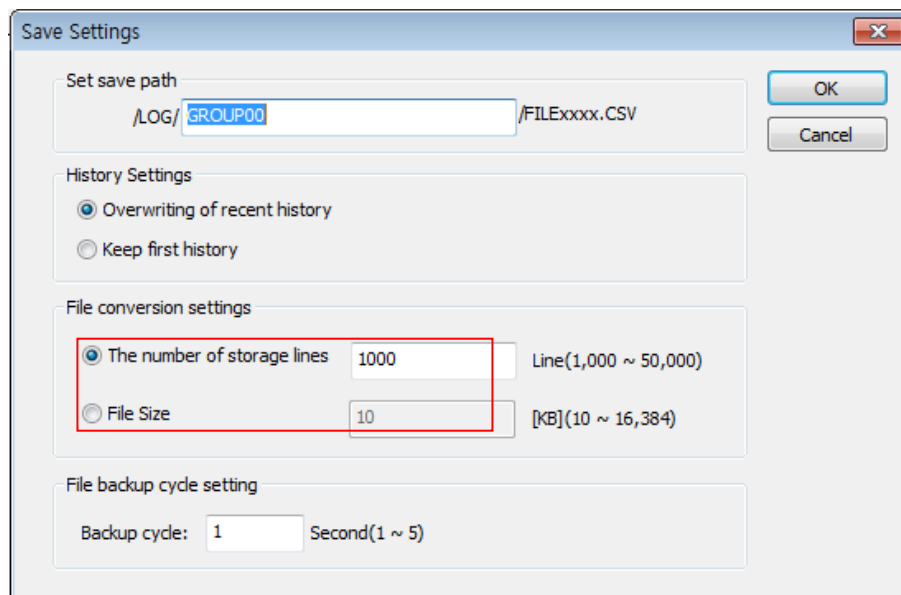


11.3.9 文件备份周期

数据记录中收集的数据没有立刻在SD内存内保存，在指定的缓存中保存后，日程容量收集完成时，在SD内存保存。数据保存周期长，要收集的数据量不多时，到SD内存中数据保存为止需要较多时间。如果收集的数据仅在缓存中保存，突然电源OFF 或者 Reset发生时，该数据会全部丢失。



为了防止此类情况，每个日程周期，和容量无关，通过 SD 内存收集的数据必须保存。通过SD 内存保存的数据及时突然电源变化也不会数据丢失。备份时间最多可设定 1~5秒。但是为了频繁备份，时间设定较短时，会对数据记录性能有影响



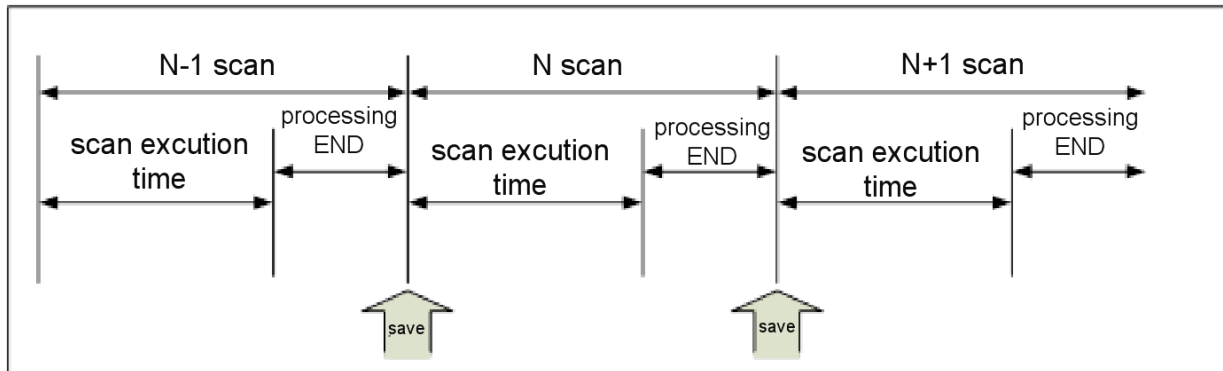
11.4 一般保存

运动控制的内置数据记录功能中一般保存是通过[主任务周期保存], [指定周期保存] 方式来进行区分运行的。
 [主任务周期保存]是主任务相应数据值采样的方法, [指定周期保存]方式是每个用户设定的周期, 数据值采样的方法。

11.4.1 主任务周期保存方式

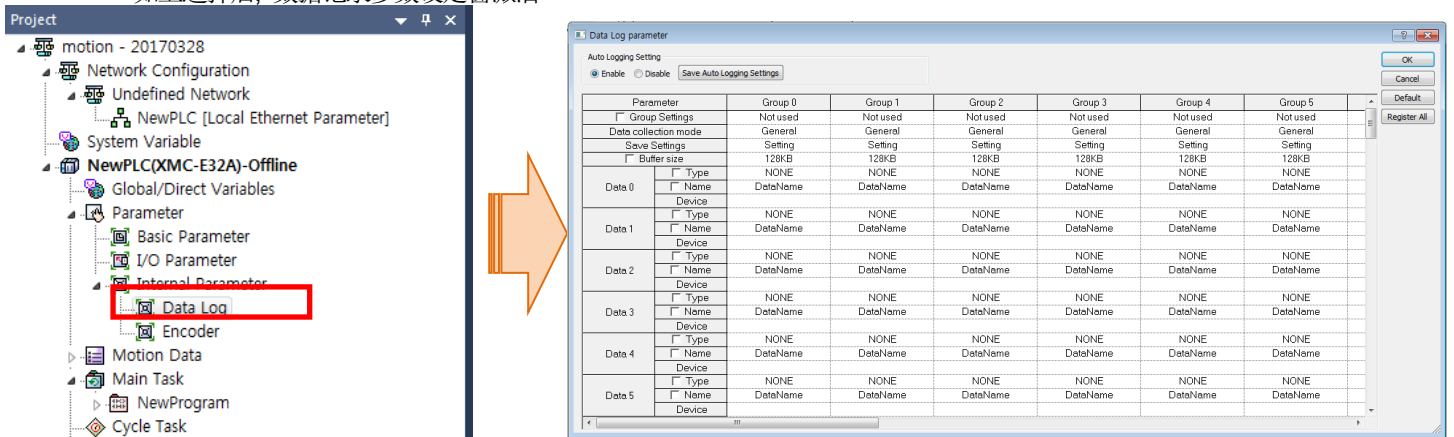
(1) 运行说明

运动控制的数据记录功能中[主任务周期保存]方式是每个主任务周期, 用户设定的数据采样的方式。若使用主任务周期保存方式, 每个主任务数据采样。较短周期或者大量的数据采样进行设定时, 数据保存量较多, 可能发生主任务周期错误, 基本参数的主任务周期设定时请注意。



(2) 设定方法

- 1) MP500 -[工程窗]- [内置参数]- [数据记录]选择。
 如上选择后, 数据记录参数设定窗激活



- 2) 数据记录参数窗中设定要使用的组

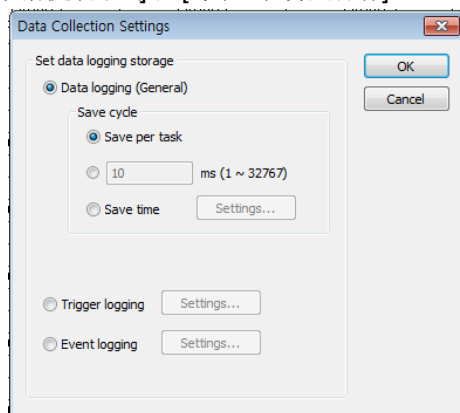
Parameter	Group 0
<input type="checkbox"/> Group Settings	Not used
Data collection mode	Not used
Save Settings	Used
<input type="checkbox"/> Buffer size	128KB

Parameter	Group 0
<input type="checkbox"/> Group Settings	Used
Data collection mode	General
Save Settings	Setting
<input type="checkbox"/> Buffer size	128KB

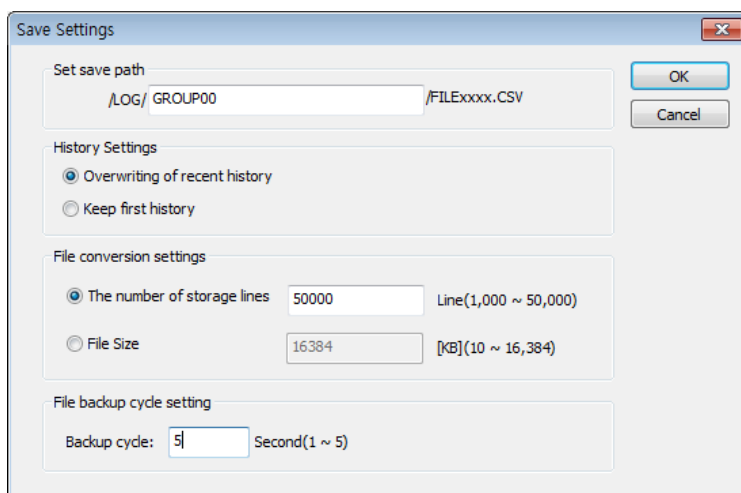
提示

数据记录参数设定和数据记录 EN 标志位全部设定后运行。即使遗漏一个条件, 数据记录都不会运行。请确定参数设定与
 否和该标志位值
 (11.10 节标志位一览参考)

- 3) 选择[数据收集方式]中 [每个主任务周保存]



- 4) [保存设定]中路径, 履历设定, 文件变换时点等设定



- 5) 数据变换类型和保存设备, 名称等设定

Parameter	Type	Group 0
Data 0	Type	NONE
	Name	NONE
	Device	BOOL
Data 1	Type	BYTE
	Name	WORD
	Device	DWORD
Data 2	Type	LWORD
	Name	SINT
	Device	INT
Data 3	Type	DINT
	Name	LINT
	Device	USINT
Data 4	Type	UINT
	Name	UDINT
	Device	ULINT
	Type	REAL
	Name	LREAL
	Device	STRING

Parameter	Type	Group 0
Data 0	Type	INT
	Name	DataName
	Device	%MW0
Data 1	Type	NONE
	Name	DataName
	Device	
Data 2	Type	NONE
	Name	DataName
	Device	
Data 3	Type	NONE
	Name	DataName
	Device	
Data 4	Type	NONE
	Name	DataName
	Device	

- 6) SD 内存卡连接后, _DL_Rdy(%KX8800) 标志位为 ON, 数据记录使能标志位(%KX8224)ON, 功能运行。
_DL_Rdy(%KX8800)标志位为 OFF 的状态下, 使能标志位为ON, 数据记录不运行。

- 数据记录组别使能标志位如下

变量	内存	类型	说明
-	%KW514	WORD	数据记录使能标志位
_DL00_Enable	%KX8224	BOOL	组 00 数据记录允许 1: 运行, 0: 停止
_DL01_Enable	%KX8225	BOOL	组 01 数据记录允许 1: 运行, 0: 停止

_DL02_Enable	%KX8226	BOOL	组 02 数据记录允许 1: 运行, 0: 停止
_DL03_Enable	%KX8227	BOOL	组 03 数据记录允许 1: 运行, 0: 停止
_DL04_Enable	%KX8228	BOOL	组 04 数据记录允许 1: 运行, 0: 停止
_DL05_Enable	%KX8229	BOOL	组 05 数据记录允许 1: 运行, 0: 停止
_DL06_Enable	%KX8230	BOOL	组 06 数据记录允许 1: 运行, 0: 停止
_DL07_Enable	%KX8231	BOOL	组 07 数据记录允许 1: 运行, 0: 停止
_DL08_Enable	%KX8232	BOOL	组 08 数据记录允许 1: 运行, 0: 停止
_DL09_Enable	%KX8233	BOOL	组 09 数据记录允许 1: 运行, 0: 停止
_DL10_Enable	%KX8234	BOOL	组 10 数据记录允许 1: 运行, 0: 停止
_DL11_Enable	%KX8235	BOOL	组 11 数据记录允许 1: 运行, 0: 停止
_DL12_Enable	%KX8236	BOOL	组 12 数据记录允许 1: 运行, 0: 停止
_DL13_Enable	%KX8237	BOOL	组 13 数据记录允许 1: 运行, 0: 停止
_DL14_Enable	%KX8238	BOOL	组 14 数据记录允许 1: 运行, 0: 停止
_DL15_Enable	%KX8239	BOOL	组 15 数据记录允许 1: 运行, 0: 停止

数据保存停止时，相应数据记录使能标志位为 OFF。

此时，若还有要在SD内存卡中保存的数据，_DLxx_Stopping(%KX8963)标志位为ON，剩下的所有数据保存完后，变为OFF。

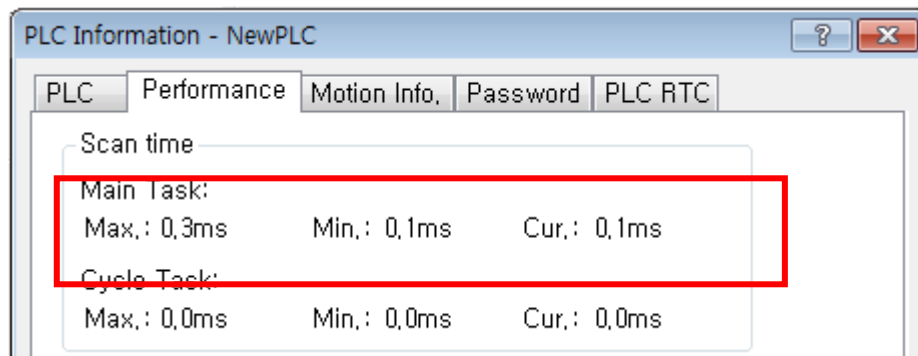
提示

通过主任务周期方式来保存数据时，参考主任务执行周期，设定数据记录参数。

较快周期期间进行较多数据设定时，可能会发生数据丢失。

- 主任务执行时间的确认可在如下菜单中进行

[在线] - [诊断] - [PLC 信息] - [性能tap]



11.4.2 指定周期保存方式

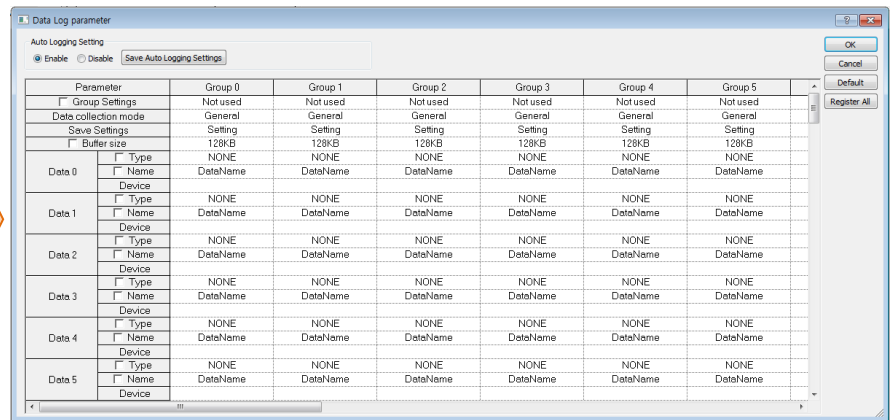
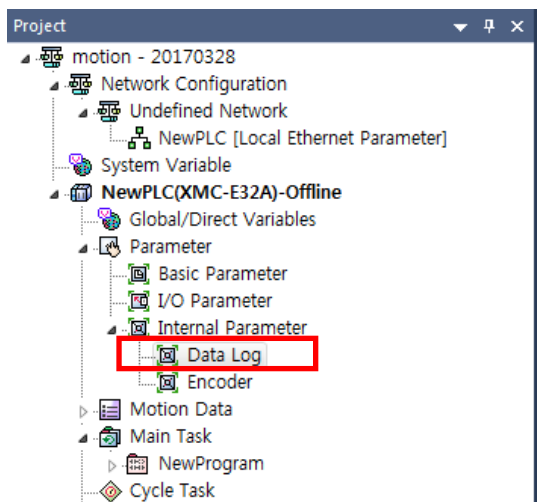
(1) 运行说明

指定周期方式是在每个用户设定的周期进行采样的方式。[每个主任务周期保存] 方式不同的点是每个日程周期都可进行数据采样，比起每个日程周期变化的值，可保存正确的时点的数据。

(2) 设定方法

1) MP500-[工程窗]-[内置参数]-[数据记录]选择

如上选择后，数据记录参数设定窗激活



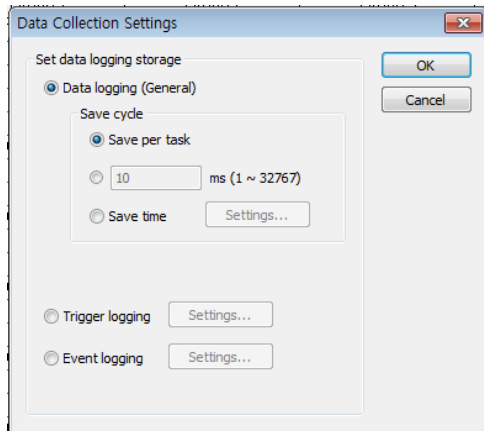
2) 数据记录参数窗中设定要使用的组

Parameter	Group 0
<input type="checkbox"/> Group Settings	Not used
Data collection mode	Not used
Save Settings	Used
<input type="checkbox"/> Buffer size	128KB

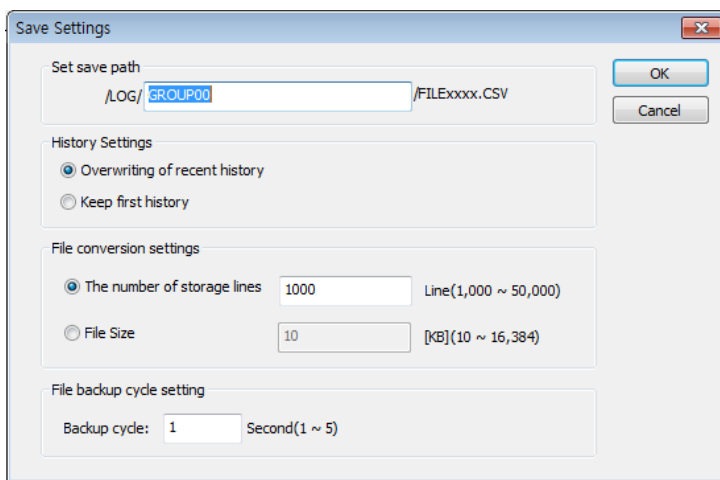
Parameter	Group 0
<input checked="" type="checkbox"/> Group Settings	Used
Data collection mode	General
Save Settings	Setting
<input type="checkbox"/> Buffer size	128KB



3) [数据收集方式设定]中直接输入保存周期。(范围: 1~32,767ms)



4) [保存设定]中路径, 履历设定, 文件变换时点等设定



5) 数据变换类型和保存设备, 名称等设定

Parameter	Type	Name	Device	Group 0
Data 0	<input type="checkbox"/>			NONE
	<input type="checkbox"/>			NONE
	<input type="checkbox"/>			BOOL
Data 1	<input type="checkbox"/>			BYTE
	<input type="checkbox"/>			WORD
	<input type="checkbox"/>			DWORD
Data 2	<input type="checkbox"/>			LWORD
	<input type="checkbox"/>			SINT
	<input type="checkbox"/>			INT
Data 3	<input type="checkbox"/>			DINT
	<input type="checkbox"/>			LINT
	<input type="checkbox"/>			USINT
Data 4	<input type="checkbox"/>			UINT
	<input type="checkbox"/>			UDINT
	<input type="checkbox"/>			ULINT
	<input type="checkbox"/>			REAL
	<input type="checkbox"/>			LREAL
	<input type="checkbox"/>			STRING

Parameter	Type	Name	Device	Group 0
Data 0	<input type="checkbox"/>			INT
	<input type="checkbox"/>			DataName
	<input type="checkbox"/>			%MW0
Data 1	<input type="checkbox"/>			NONE
	<input type="checkbox"/>			DataName
	<input type="checkbox"/>			Device
Data 2	<input type="checkbox"/>			NONE
	<input type="checkbox"/>			DataName
	<input type="checkbox"/>			Device
Data 3	<input type="checkbox"/>			NONE
	<input type="checkbox"/>			DataName
	<input type="checkbox"/>			Device
Data 4	<input type="checkbox"/>			NONE
	<input type="checkbox"/>			DataName
	<input type="checkbox"/>			Device

- 6) SD 内存卡连接后, _DL_Rdy(%KX8800)标志位为 ON, 数据记录使能标志位(%KX8224)为 ON后功能运行。_DL_Rdy(%KX8800)标志位为OFF的状态下, 使能标志位为 ON, 数据记录不运行。
- 数据记录组别使能标志位如下

变量	内存	类型	说明
-	%KW514	WORD	数据记录使能标志位
_DL00_Enable	%KX8224	BOOL	组 00 数据记录允许 1: 运行, 0: 停止
_DL01_Enable	%KX8225	BOOL	组 01 数据记录允许 1: 运行, 0: 停止
_DL02_Enable	%KX8226	BOOL	组 02 数据记录允许 1: 运行, 0: 停止
_DL03_Enable	%KX8227	BOOL	组 03 数据记录允许 1: 运行, 0: 停止
_DL04_Enable	%KX8228	BOOL	组 04 数据记录允许 1: 运行, 0: 停止
_DL05_Enable	%KX8229	BOOL	组 05 数据记录允许 1: 运行, 0: 停止
_DL06_Enable	%KX8230	BOOL	组 06 数据记录允许 1: 运行, 0: 停止
_DL07_Enable	%KX8231	BOOL	组 07 数据记录允许 1: 运行, 0: 停止
_DL08_Enable	%KX8232	BOOL	组 08 数据记录允许 1: 运行, 0: 停止
_DL09_Enable	%KX8233	BOOL	组 09 数据记录允许 1: 运行, 0: 停止
_DL10_Enable	%KX8234	BOOL	组 10 数据记录允许 1: 运行, 0: 停止
_DL11_Enable	%KX8235	BOOL	组 11 数据记录允许 1: 运行, 0: 停止
_DL12_Enable	%KX8236	BOOL	组 12 数据记录允许 1: 运行, 0: 停止
_DL13_Enable	%KX8237	BOOL	组 13 数据记录允许 1: 运行, 0: 停止
_DL14_Enable	%KX8238	BOOL	组 14 数据记录允许 1: 运行, 0: 停止
_DL15_Enable	%KX8239	BOOL	组 15 数据记录允许 1: 运行, 0: 停止

数据保存停止时, 相应数据记录使能标志位为 OFF。

此时, 若还有未在 SD 内存卡中保存的数据, _DLxx_Stoping(%KX8963) 标志位为 ON, 剩下的数据保存后, 变为OFF。

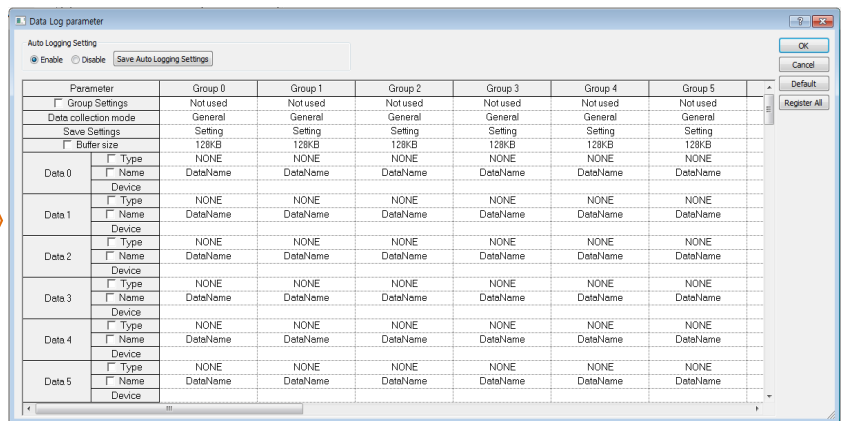
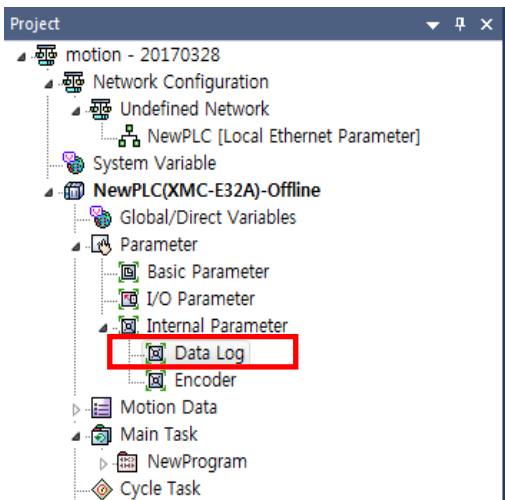
11.4.3 指定时间保存方式

(1) 运行说明

指定时间方式是在每个用户设定的时间内进行采样的方式。[指定周期保存] 方式不同的点设定的时间上可进行数据采样，在正确的时间进行数据保存

(2) 设定方法

- 1) MP500 -[工程窗] - [内置参数] - [数据记录]选择
如上选择后，数据记录参数设定窗激活

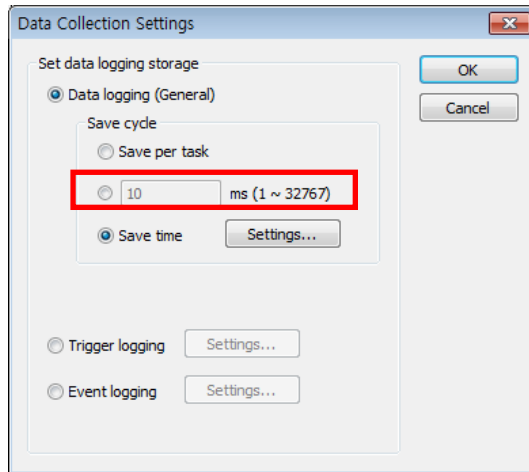


- 2) 数据记录参数窗中设定要使用的组

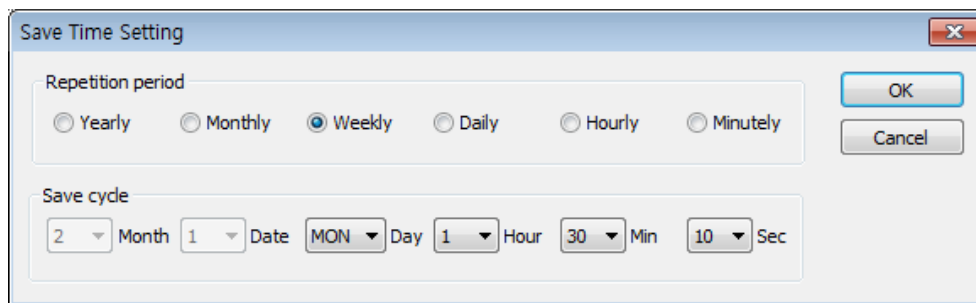
Parameter	Group 0
<input type="checkbox"/> Group Settings	Not used
Data collection mode	Not used
Save Settings	Used
<input type="checkbox"/> Buffer size	128KB

Parameter	Group 0
<input checked="" type="checkbox"/> Group Settings	Used
Data collection mode	General
Save Settings	Setting
<input type="checkbox"/> Buffer size	128KB

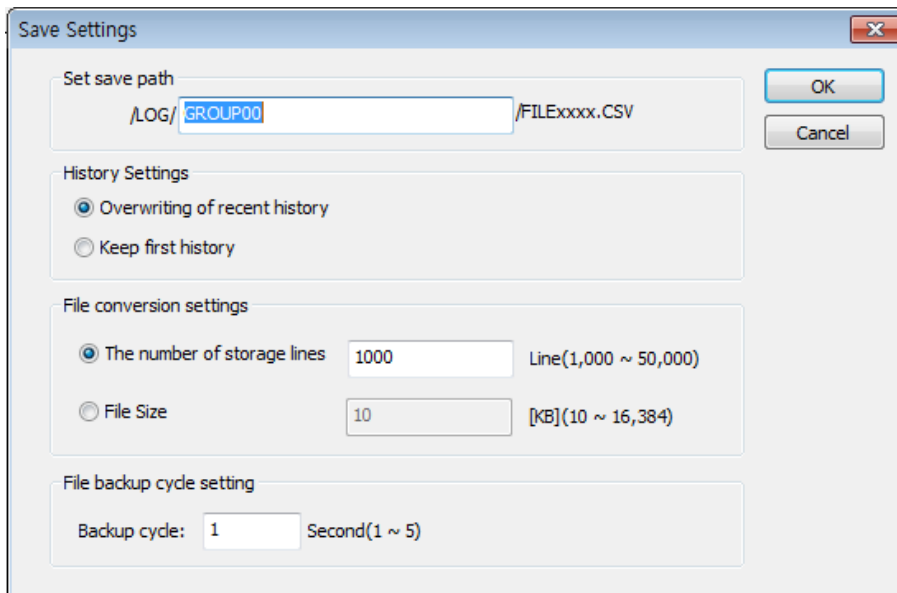
- 3) [数据收集方式设定]中直接输入保存周期



- 4) 每年/每月/每周/每天/每小时/每分钟中选择一个



- 5) [保存设定]中路径, 履历设定, 文件变换时点等设定



6) 数据变换类型和保存设备, 名称等设定

Parameter		Group 0	Parameter		Group 0
Data 0	Type	NONE	Data 0	Type	INT
	Name	NONE		Name	DataName
	Device	BOOL		Device	%MW0
Data 1	Type	BYTE	Data 1	Type	NONE
	Name	WORD		Name	DataName
	Device	DWORD		Device	
Data 2	Type	LWORD	Data 2	Type	NONE
	Name	SINT		Name	DataName
	Device	INT		Device	
Data 3	Type	DINT	Data 3	Type	NONE
	Name	LINT		Name	DataName
	Device	USINT		Device	
Data 4	Type	UIINT	Data 4	Type	NONE
	Name	UDINT		Name	DataName
	Device	ULINT		Device	
Data 4	Type	REAL	Data 4	Type	NONE
	Name	LREAL		Name	DataName
	Device	STRING		Device	

7) SD 内存卡连接后, $_DL_Rdy(\%KX8800)$ 标志位为 ON, 数据记录使能标志位($\%KX8224$)为 ON时, 功能运行。 $_DL_Rdy(\%KX8800)$ 标志位为 OFF 的状态下, 使能标志位为ON, 数据记录不运行。

- 数据记录组别使能标志位如下

变量	内存	类型	说明
-	$\%KW514$	WORD	数据记录使能标志位
$_DL00_Enable$	$\%KX8224$	BOOL	组 00 数据记录允许 1: 运行, 0: 停止
$_DL01_Enable$	$\%KX8225$	BOOL	组 01 数据记录允许 1: 运行, 0: 停止
$_DL02_Enable$	$\%KX8226$	BOOL	组 02 数据记录允许 1: 运行, 0: 停止
$_DL03_Enable$	$\%KX8227$	BOOL	组 03 数据记录允许 1: 运行, 0: 停止
$_DL04_Enable$	$\%KX8228$	BOOL	组 04 数据记录允许 1: 运行, 0: 停止
$_DL05_Enable$	$\%KX8229$	BOOL	组 05 数据记录允许 1: 运行, 0: 停止
$_DL06_Enable$	$\%KX8230$	BOOL	组 06 数据记录允许 1: 运行, 0: 停止
$_DL07_Enable$	$\%KX8231$	BOOL	组 07 数据记录允许 1: 运行, 0: 停止
$_DL08_Enable$	$\%KX8232$	BOOL	组 08 数据记录允许 1: 运行, 0: 停止
$_DL09_Enable$	$\%KX8233$	BOOL	组 09 数据记录允许 1: 运行, 0: 停止
$_DL10_Enable$	$\%KX8234$	BOOL	组 10 数据记录允许 1: 运行, 0: 停止
$_DL11_Enable$	$\%KX8235$	BOOL	组 11 数据记录允许 1: 运行, 0: 停止
$_DL12_Enable$	$\%KX8236$	BOOL	组 12 数据记录允许 1: 运行, 0: 停止
$_DL13_Enable$	$\%KX8237$	BOOL	组 13 数据记录允许 1: 运行, 0: 停止
$_DL14_Enable$	$\%KX8238$	BOOL	组 14 数据记录允许 1: 运行, 0: 停止
$_DL15_Enable$	$\%KX8239$	BOOL	组 15 数据记录允许 1: 运行, 0: 停止

数据保存停止时，相应数据记录使能标志位为 OFF。

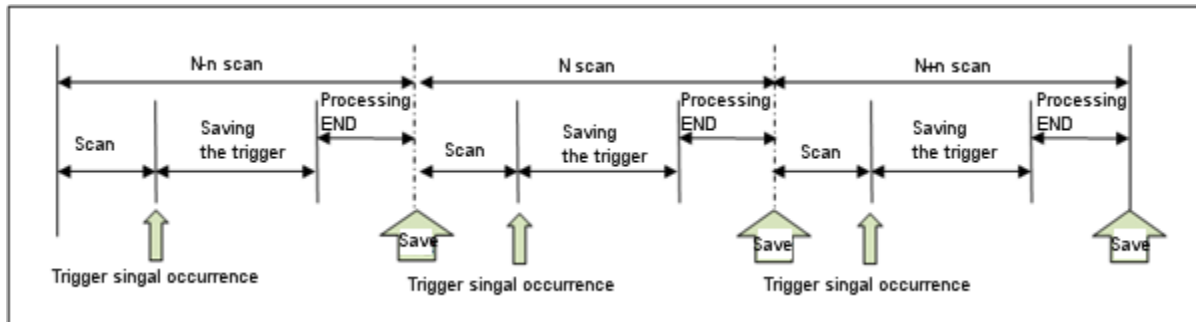
此时，若还有为在 SD 内存卡中保存数据_DLxx_Stoping(%KX8963) 标志位为 ON，
剩余数据保存后，变为OFF。

提示

SD内存连接后_DL_Rdy标志位若不是 ON，数据记录功能不运行。_DL_Rdy标志位为OFF，请确认错误代码及SD内存连接状态。

11.5 触发保存

触发保存功能是满足指定的数据保存条件，在相应时点之前数据和之后数据时点的数据保存在参数中设定的个数的方式。即，特定值发生时，要确认之前日程时间的数据和之后日程时间数据时，有用的方式。触发发生条件在主任务END中查看。



提示

[触发保存] 设定后，最初触发条件发生，数据收集期间又发生其他触发条件的话，忽略相应触发，增加触发冲突次数标志位值

提示

收集比触发之前采样块数少，若发生触发，那就保存到此时为止收集的块数，因此触发后，采样开始。在这种情况下，收集的全体块数中比触发块较少则可以收集。

11.5.1 触发条件

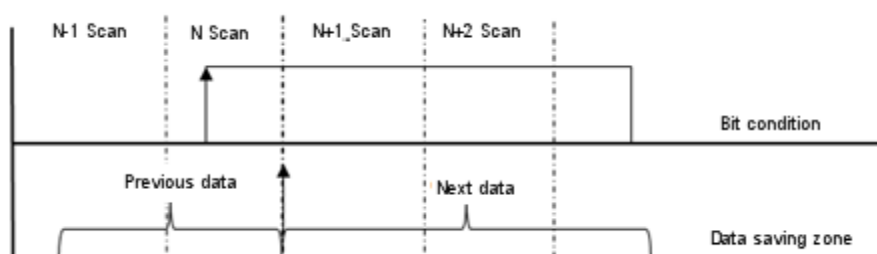
触发保存功能是根据单一条件，多种条件来运行的。单一条件，多种条件设定项目如下。多种条件是通过取单一条件AND, OR 来运行的方法，最多4个的单一条件设定，条件可设定。触发条件发生，数据保存开始时，最初数据列中“T”文字列插入，告知触发开始位置。

(1) 单一条件

单一条件是通过较大的 [位条件], [字条件]进行区分运行的

1) 位条件

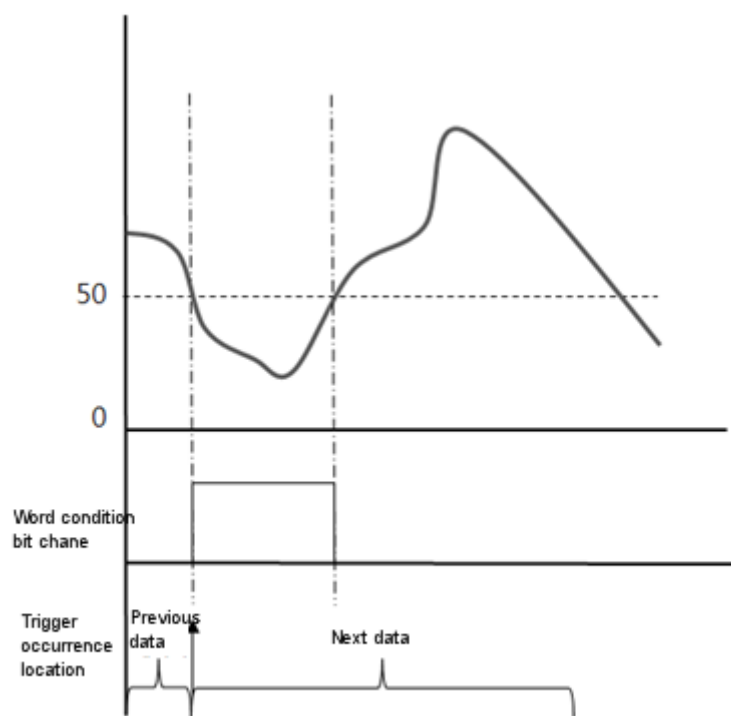
位条件是设定的设备bit值查看，值在 [上升] 或者 [下降] 时，检测触发，数据收集的



2) 字条件

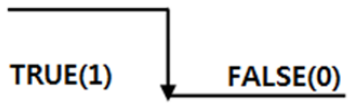
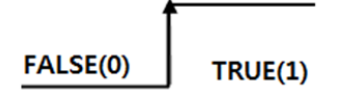
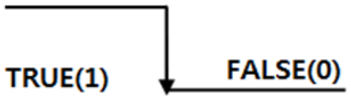
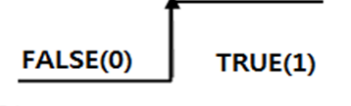

字条件是和设定设备的输入值对比，变为TRUE，FALSE。设定设备值满足输入的条件时，相应变换bit值 FALSE→TRUE 或者 TRUE→FALSE 时，数据收集。

例) 设定值<50, 上升条件



3) 条件说明

	触发发生条件	设备设定条件	运行	备注
位条件	上升	X	设定设备位从 FALSE(0) 变为 TRUE(1) 时，数据保存。 	
	下降		设定设备位从 TRUE(1) 变为 FALSE(0) 时，数据保存。 	
字条件	上升	<	设定字设备值和输入设定值对比时，不满足条件时(FALSE)变为满足(TRUE)，数据采样.. <p>Ex) $device\ value \geq set\ value$ Ex) $device\ value < set\ value$ $device\ value = set\ value$ $device\ value > set\ value$</p>	
	下降		设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE)时，数据采样.. <p>Ex) $device\ value < set\ value$ Ex) $device\ value \geq set\ value$ $device\ value = set\ value$ $device\ value \leq set\ value$</p>	
	上升	\leq	设定字设备值和输入设定值对比时，不满足条件时(FALSE)变为满足(TRUE)，数据采样。 <p>Ex) $device\ value > set\ value$ Ex) $device\ value \leq set\ value$</p>	

	触发发生条件	设备设定条件	运行	备注
字条件	下降	\leq	<p>设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE)时，数据采样.</p>  <p>Ex) device value <= set value Ex) device value > set value</p>	
	上升	$>$	<p>设定字设备值和输入设定值对比时，不满足条件时(FALSE)变为满足(TRUE)，数据采样.</p>  <p>Ex) device value <= set value Ex) device value > set value device value = set value device value >= set value</p>	
	下降		<p>设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE)时，数据采样.</p>  <p>Ex) device value > set value Ex) device value >= set value</p>	
	上升	\geq	<p>设定字设备值和输入设定值对比时，不满足条件时(FALSE)变为满足(TRUE)，数据采样.</p>  <p>Ex) device value < set value Ex) device value >= set value</p>	
	下降		<p>设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE)时，数据采样.</p>  <p>Ex) device value >= set value Ex) device value < set value</p>	

	触发发生条件	设备设定条件	运行	备注
字条件	上升	相同	设定字设备值和输入设定值对比时，不满足条件时(FALSE)变为满足(TRUE)，数据采样。 	
	下降		设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE)时，数据采样。 	
	上升	不同	设定字设备值和输入设定值对比时，不满足条件时(FALSE)变为满足(TRUE)，数据采样。 	
	下降		设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE)时，数据采样。 	

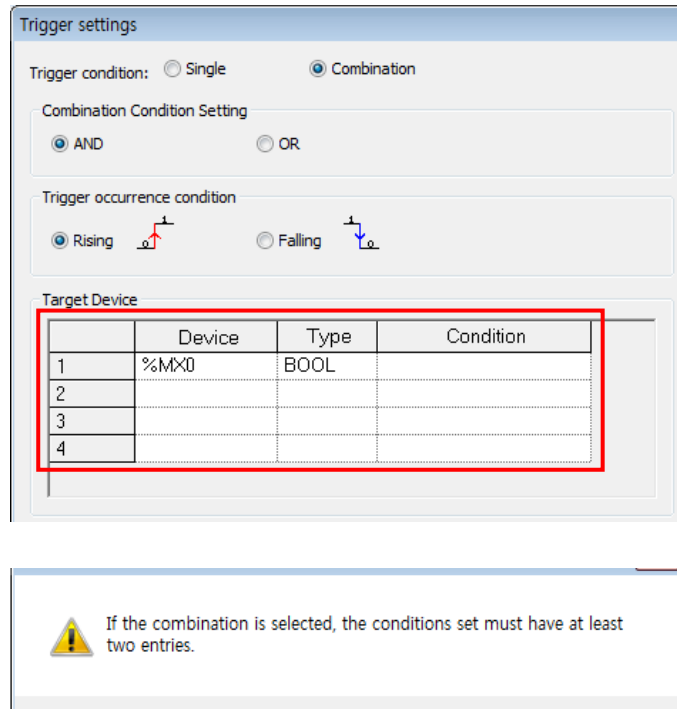
(2)多种条件

多种条件是指单一条件最多可设定 4 个, 在符合条件的演算后运行的方法。

多种条件是指单一条件最少 2 个以上设定后, 通过设定的单一条件进行演算, 其结果满足条件, 触发开始保存。多种条件按照 [AND演算], [OR 演算]区分后运行。

提示

触发多种条件设定时, 没有按最少 2 个以上条件来设定时, 会出现如下窗口。



Trigger settings

Trigger condition: Single Combination

Combination Condition Setting


AND OR

Trigger occurrence condition

Rising Falling

Target Device

	Device	Type	Condition
1	%MX0	BOOL	
2			
3			
4			

 If the combination is selected, the conditions set must have at least two entries.

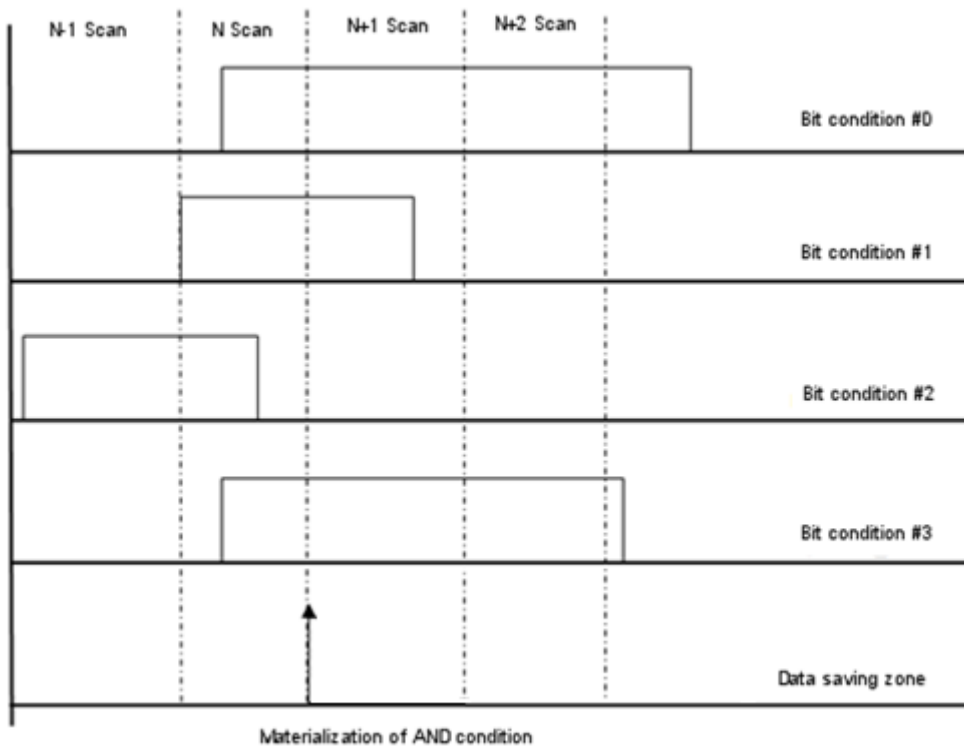
1) AND演算

主任务 END 中满足所有相应条件时，触发发生。

以下是触发上升，下降在相同的主任务周期内发生，触发保存开始的例子如下

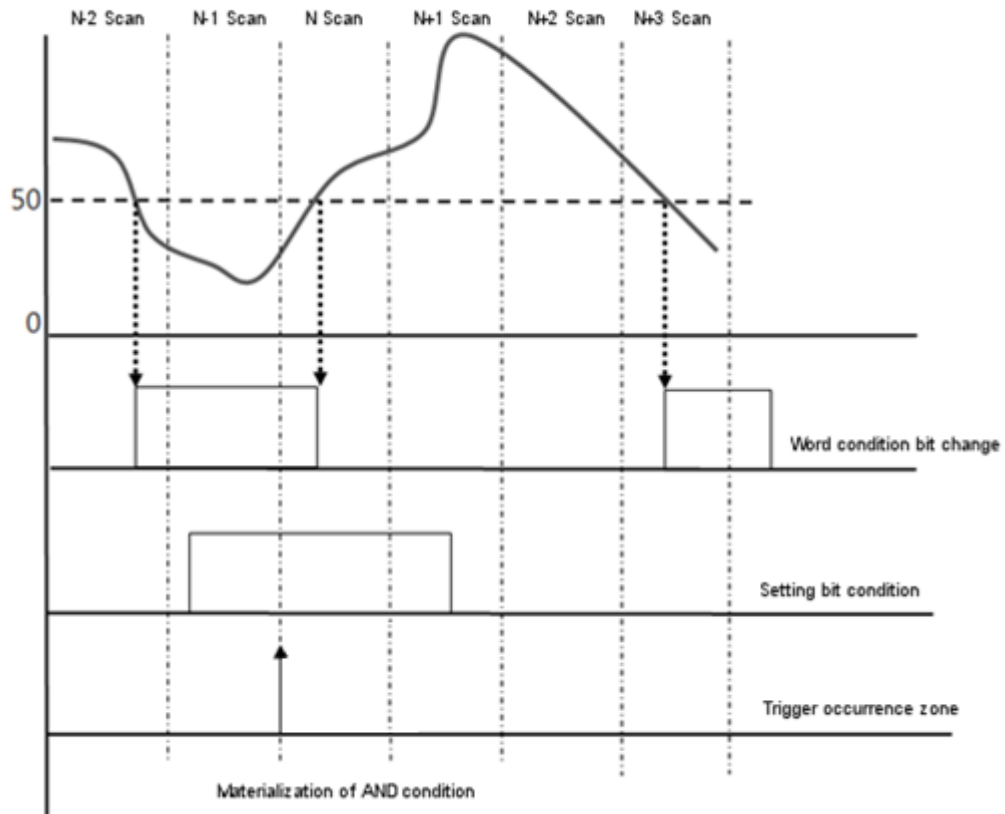
☞ 仅通过位条件设定

	条件	设定设备	触发发生条件
0 号条件	bit	%MX1010	上升
1 号条件	bit	%IX1	
2 号条件	bit	%MX2010	
3 号条件	bit	%QX130	



1. 位, 字条件混合设定

	条件	比较条件	设定值	设定设备	触发发生条件
0号条件	字	<	50	%MW10	上升
1号条件	位			%MX15	

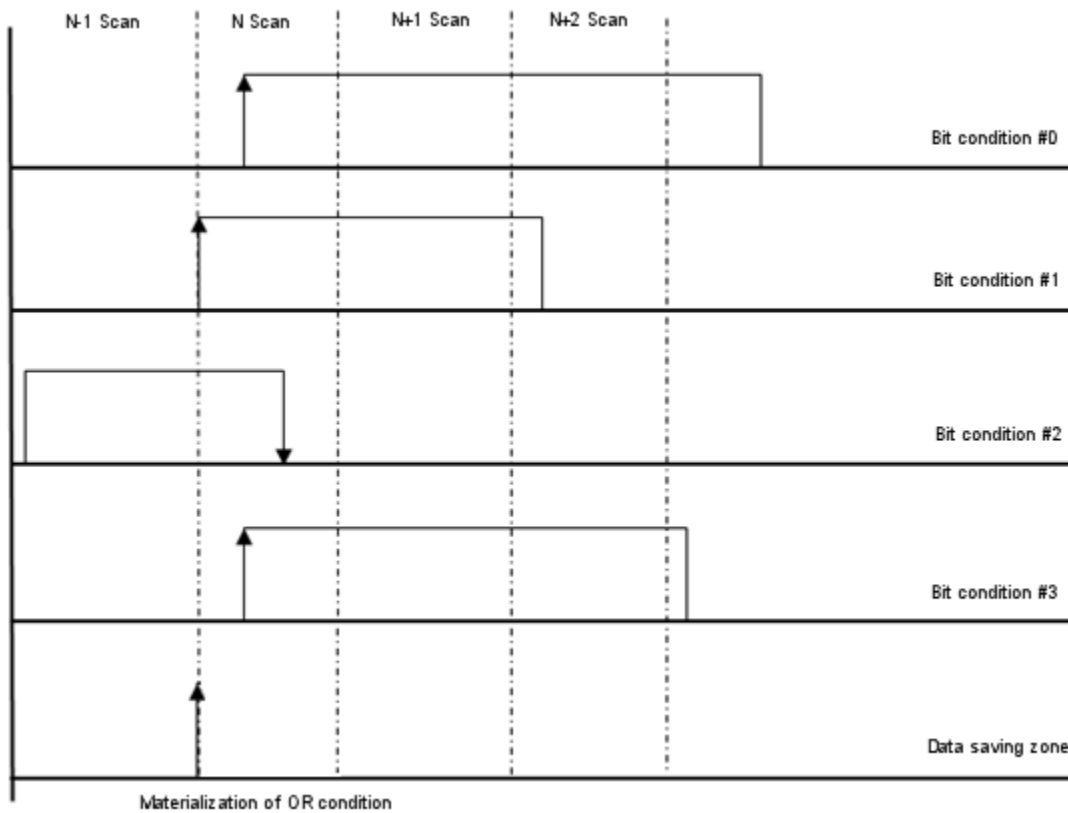


2) OR 演算

相应主任务周期内只要满足 1 个相应条件，触发都会发生。触发发生之后数据保存完成前，触发条件再次满足时，增加触发冲突次数标志位，忽略相应标志位。

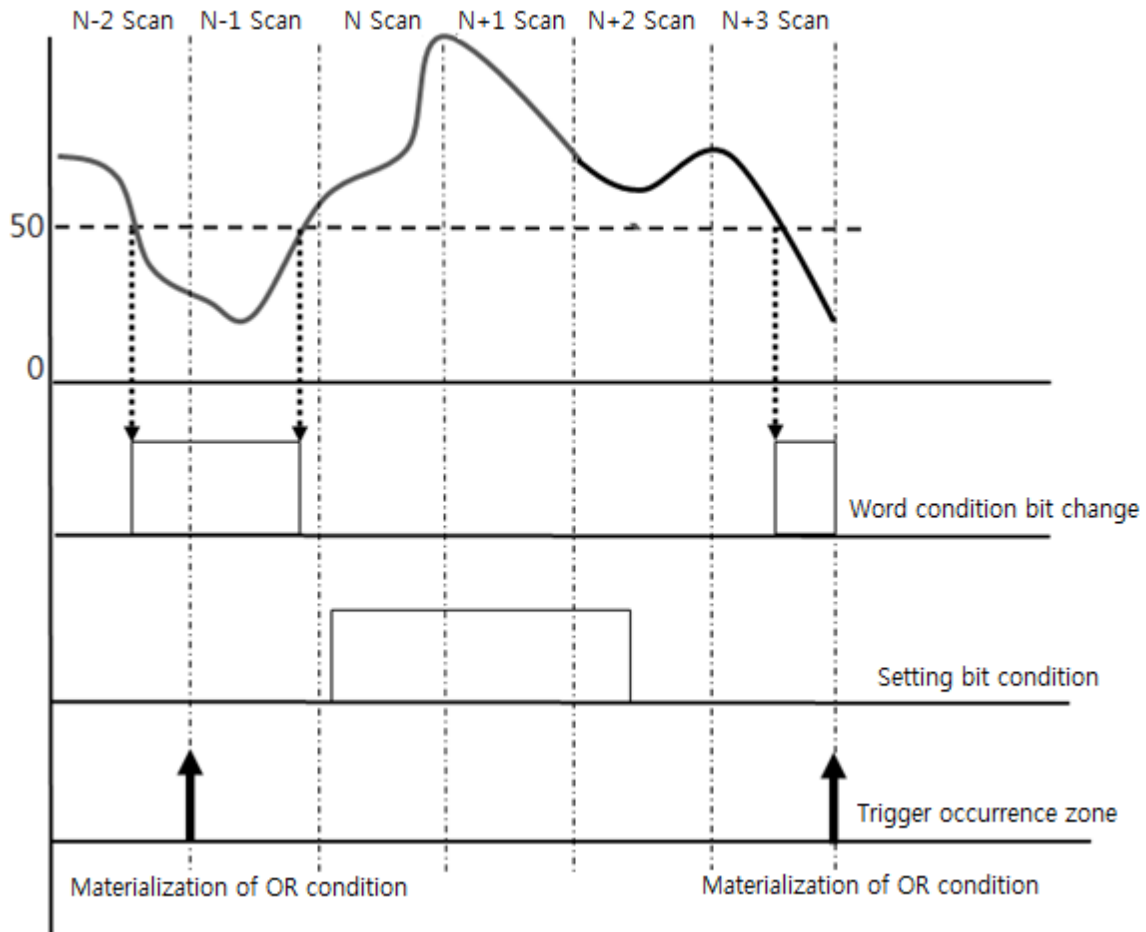
☞ 仅通过位条件设定

	条件	设定设备	触发发生条件
0 号条件	位	%MX1010	上升
1 号条件	位	%IX1	
2 号条件	位	%MX2010	
3 号条件	位	%QX130	



位, 字条件混合设定

	条件	比较条件	设定值	设定设备	触发发生条件
0号条件	字	<	50	%MW10	上升
1号条件	位			%MX15	



11.5.2 触发采样块改善方法

触发保存时，数据收集按采样块单位进行。采样块是指数据记录参数中设定的收集数据单位，采样表示各数据值。触发采样块和总采样块数计算如下。

$$\text{采样块数} = \frac{\text{设定缓存内存size}^1}{\{(\text{设定数据个数}^2 * \text{设定数据大小}^3) + (\text{RTC数据大小}^4)\}}$$

$$\text{触发缓存空间}^1 / \{(\text{设定数据个数}^2 * \text{设定数据大小}^3) + (\text{RTC数据大小}^4)\}$$

$$\text{保存块数} = \text{采样块} * \text{设定数据个数}$$

- (1) 设定缓存内存大小: 64KB ~ 2,048KB 选择(组别)
- (2) 设定数据个数: 64 个 (最大设定个数)
- (3) 设定数据大小

数据类型	数据大小
BOOL	1
BYTE	1
WORD	2
DWORD	4
LWORD	8
INT	2
SINT	1
DINT	4
LINT	8
UINT	2
USINT	1
UDINT	4
ULINT	8
REAL	4
LREAL	8
STRING	32

例)

- 设定缓存内存: 128KB
- 设定数据个数及类型: 64个 (BYTE)
 - ◆ 最大可设定的采样块数:
 $131,072 / \{(64 * 1) + 29\} = 1409$ 采样块

- (4) RTC 数据大小: 29(固定值)

11.5.3 触发采样计算方法

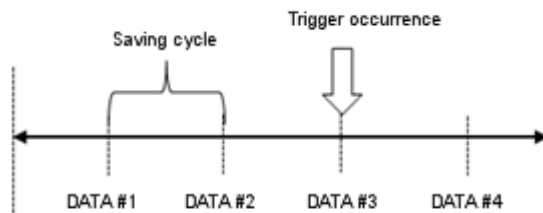
参数中可设定的项目是总采样块数和触发后采样块数，触发前块数的2个输入的值来自动决定。

$$\begin{array}{l} \text{总采样块数} = \text{触发全采样块数} + \text{触发后采样块数} \\ \text{(可设定)} \quad \quad \quad \text{(自动计算)} \quad \quad \quad \text{(可设定)} \end{array}$$

11.5.4 触发采样保存周期

触发条件发生时，数据收集按照在参数中设定的采样周期间隔来保存，保存周期如下。

→主任务周期, 20ms, 50ms, 100ms, 200ms, 500ms, 1000ms, 2000ms

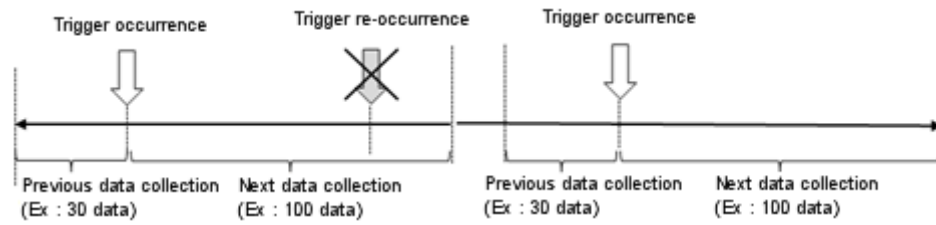


注意事项

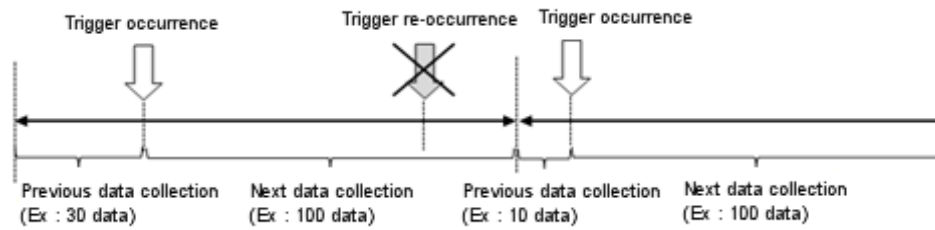
触发条件发生后，数据采样中再次触发发生时，忽略相应触发，增加触发冲突次数标志位。保存设定的触发采样块数后，查看触发条件，数据采样。

11.5.5 触发采样保存区间

- (1) 参数中设定的之前数据收集个数保存后，触发发生时，
→按参数中设定的个数数据采样



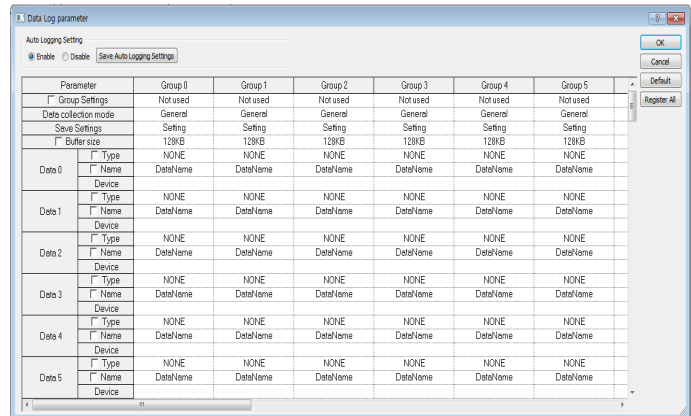
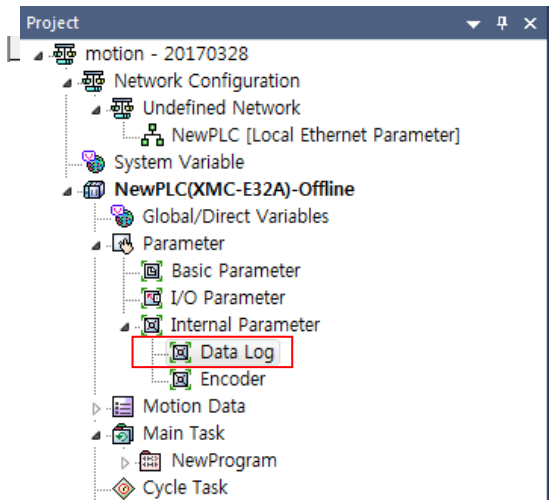
- (2) 按照参数中设定的之前块，在保存前，触发发生时，
→到那时为止，按照收集的块数，采样后，之后数据收集 (比参数中设定的数据个数少进行采样)



11.5.6 设定方法

(1) 单一条件

- 1) MP500-[工程窗]-[内置参数]-[数据记录]选择
如上选择后，数据记录参数设定窗激活

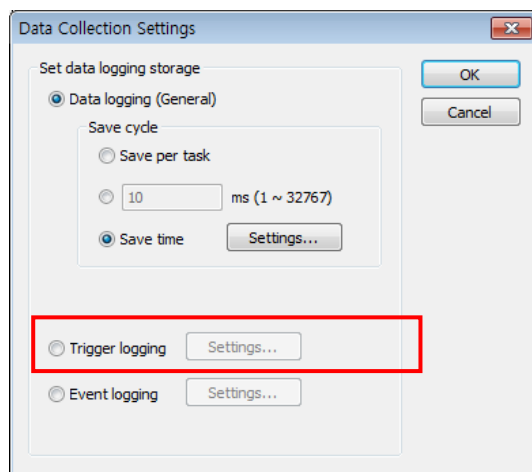


- 2) 数据记录参数窗中设定要使用的组

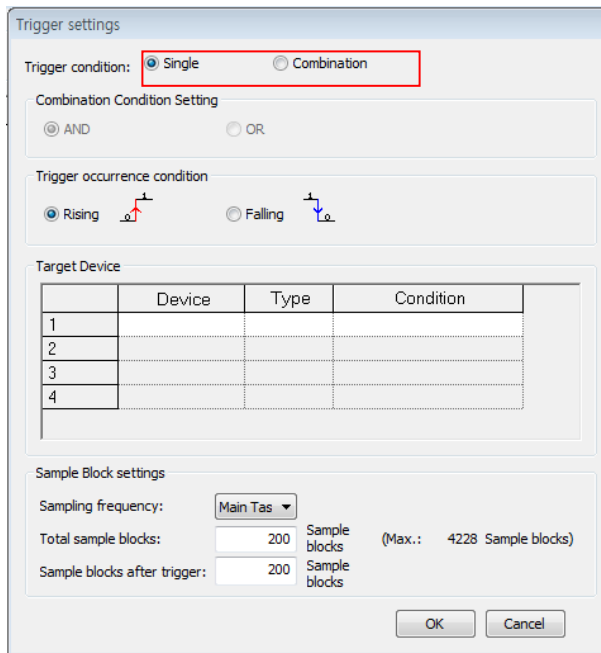
Parameter	Group 0
<input type="checkbox"/> Group Settings	Not used
Data collection mode	Not used
Save Settings	Used
<input type="checkbox"/> Buffer size	128KB

Parameter	Group 0
<input type="checkbox"/> Group Settings	Used
Data collection mode	General
Save Settings	Setting
<input type="checkbox"/> Buffer size	128KB

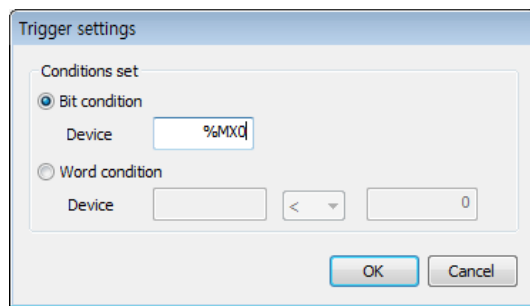
- 3) 选择[数据收集方式设定]中 [触发记录], 左侧 [设定]菜单激活。
此时, 选择左侧 [设定] 菜单



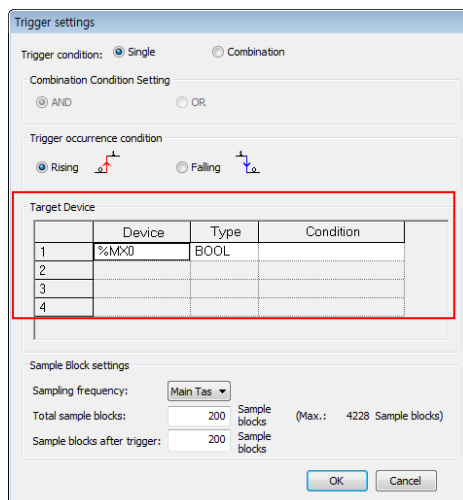
- 4) 选择[设定] 菜单, 触发设定如下图, 窗口激活。
 触发条件通过 [单一条件]选择。触发发生条件选择 [上升]或者[下降]



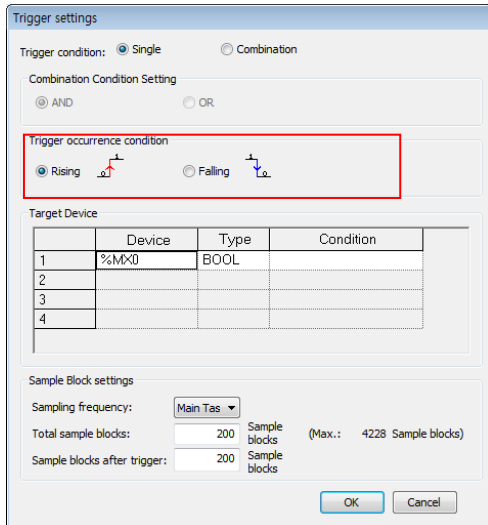
- 5) 选择条件设定菜单, 如下图, 窗口激活。
 选择[位条件]后, 设备窗中通过位类型输入设备值。



设定完成后, 窗关闭, 最初触发设定条件菜单中设定的条件如下显示。



6) 触发发生条件项目值选择



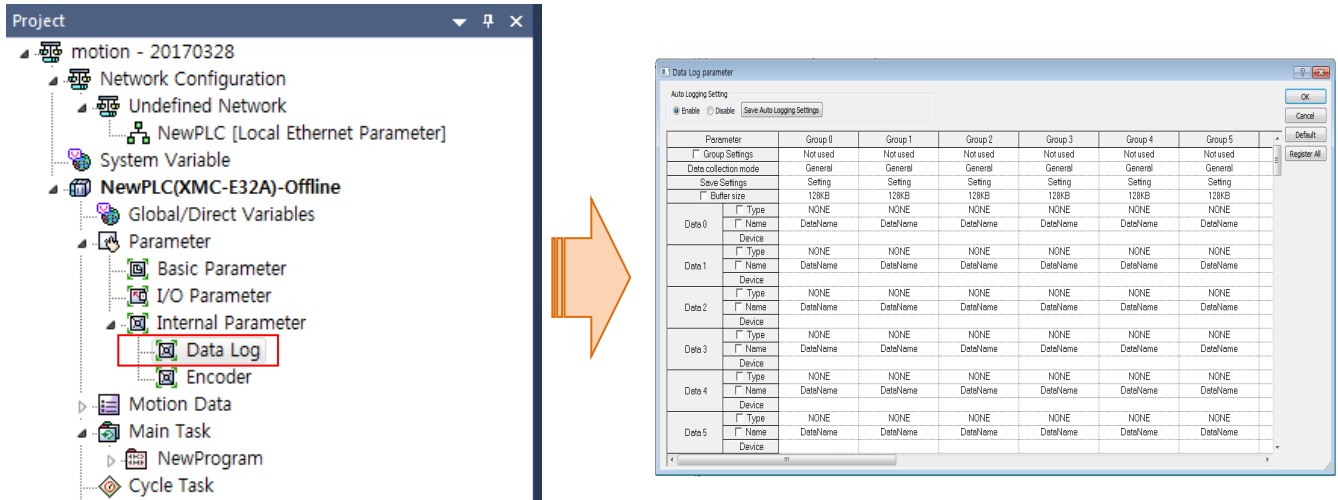
- 7) 采样周期, 总采样块数, 触发后采样块数输入后, 触发设定完成. 采样块数计算可参考 [11.5.2 触发采样块算法]
- 8) 数据在数据记录基本设定窗中收集设定的设备值, 类型变换后 SD 内存中保存。

Parameter	Group 0
Data 0	Type: NONE
	Name: NONE
	Device: BOOL
Data 1	Type: BYTE
	Name: WORD
	Device: DWORD
Data 2	Type: SINT
	Name: INT
	Device: DINT
Data 3	Type: LINT
	Name: USINT
	Device: UWORD
Data 4	Type: ULINT
	Name: REAL
	Device: LREAL
	STRING

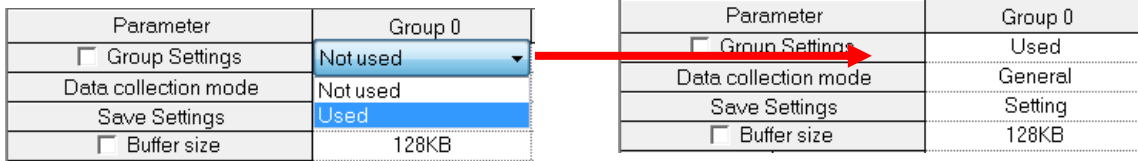
第11章数据记录功能

(2) 单一字条件

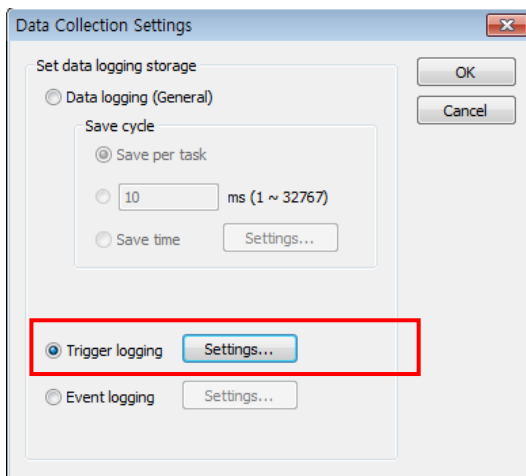
- 1) MP500 -[工程窗]- [内置参数]- [数据记录]选择
如上选择后，数据记录参数设定窗激活



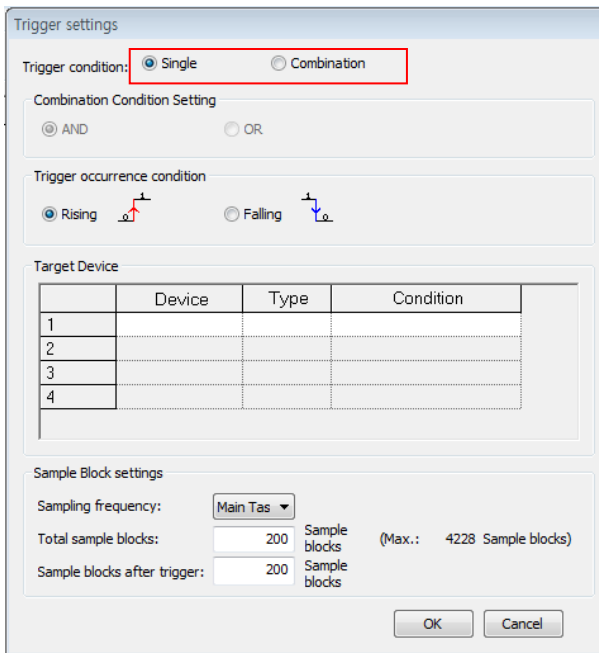
- 2) 数据记录参数窗中设定要使用的组



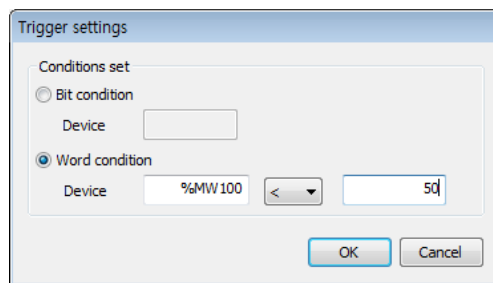
- 3) [数据收集方式设定]中选择 [触发记录], 左侧 [设定]菜单激活.
此时, 左侧 [设定] 菜单选择



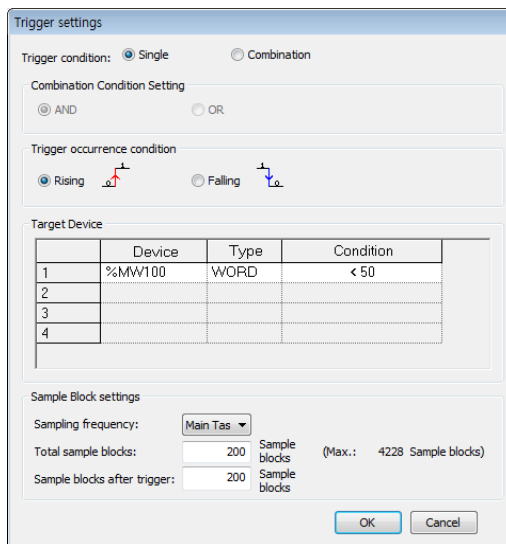
- 4) 选择[设定] 菜单, 触发可设定的如下窗口激活
 触发条件通过 [单一条件]选择, 触发发生条件选择 [上升] 或者 [下降]



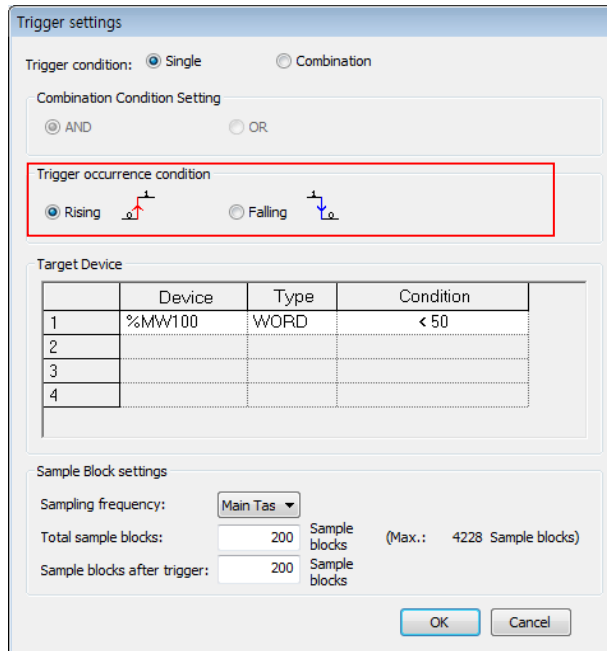
- 5) 选择条件设定菜单, 如下图, 窗口激活。
 选择[字条件] 后, 设备窗中通过字类型输入设备值, 或者输入比较条件和比较值。
 比较条件: >, ≥, 相同, <, ≤, 不一致。



设定完成, 窗口关闭时, 最初触发设定条件菜单中设定的条件如下显示



6) 选择触发发生条件项目值



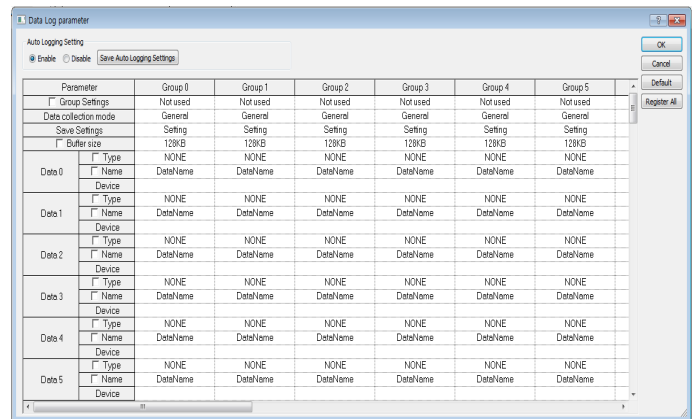
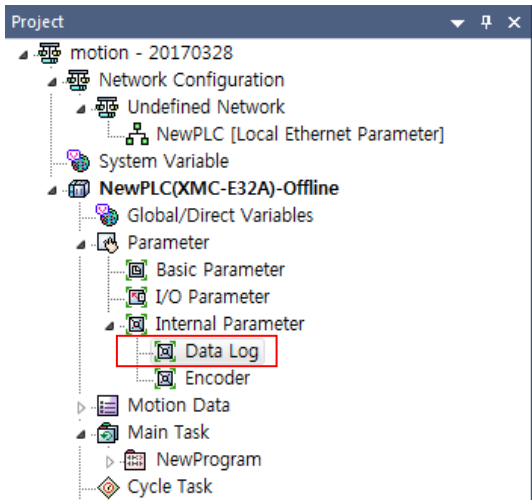
- 7) 采样周期, 总采样块数, 触发后采样块数输入后触发设定完成. 采样块数计算参考 [11.5.2 触发采样块算法].
- 8) 数据在数据记录基本设定窗中收集设定的设备值, 类型变换后SD 内存中保存.

注意事项

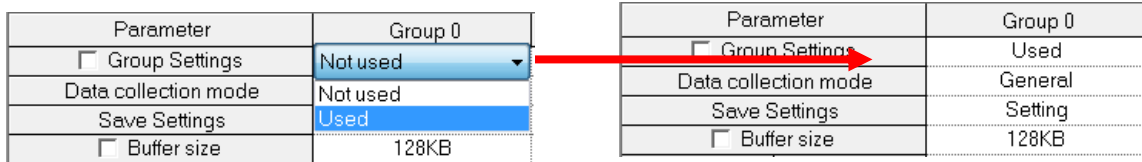
单一, 字条件设定值输入时, 设备类型按照各个 [位], [字] 格式输入

(3) 多种AND条件

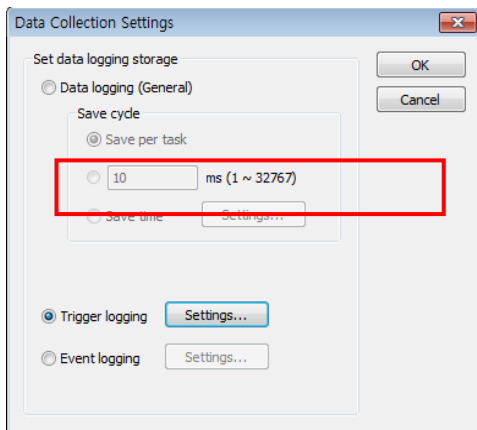
- 1) MP500 -[工程窗]- [内置参数]- [数据记录]选择
如上选择后，数据记录参数设定窗激活



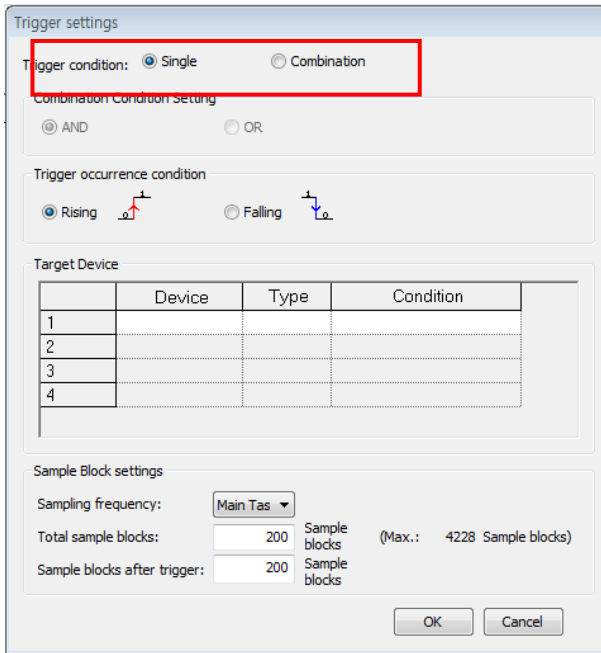
- 2) 数据记录参数窗中设定要使用的组



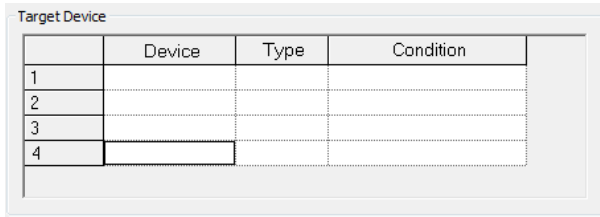
- 3) [数据收集方式设定]中选择 [触发记录]，左侧[设定]菜单激活。
此时，左侧[设定] 菜单选择



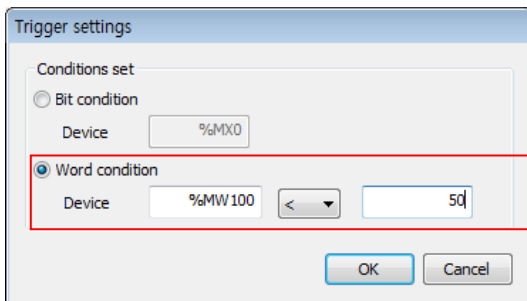
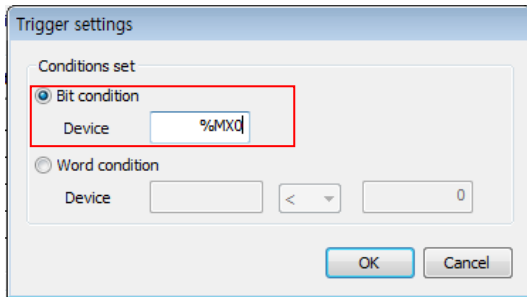
- 4) 选择[设定] 菜单, 触发可设定的如下窗口激活.
 触发条件通过 [多种条件]选择, 触发发生条件选择 [上升] 或者 [下降].



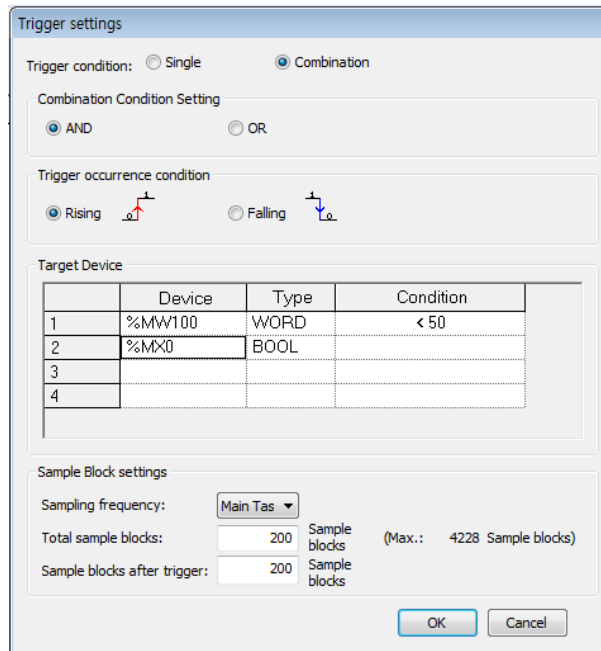
- 5) 选择[触发条件]-[多种条件], 条件设定窗最多输入 4个数



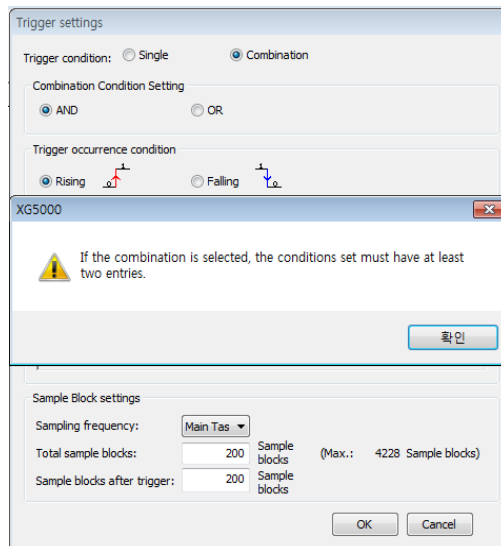
- 6) 选择一个条件设定菜单, 输入详细设定值.
 [多种条件]通过[单一条件]演算方式, 触发条件发生, 按照数据保存的方式, 基本设定方法如下, 和单一条件相同.



设定完成, 窗口关闭时, 最初触发设定条件菜单中设定的条件如下显示



如果, 选择演算条件, [条件设定]输入, 设定后会出现如下语句, 则设定未完成.



- 7) 采样周期, 总采样块数, 触发后采样块数输入后触发设定完成
- 8) 数据记录基本设定窗中设定的设备值在触发条件发生时, 会变为数据收集的后设定的类型, 保存在SD内存内

Parameter		Group 0
Data 0	<input type="checkbox"/> Type	NONE
	<input type="checkbox"/> Name	NONE
	Device	BOOL
Data 1	<input type="checkbox"/> Type	BYTE
	<input type="checkbox"/> Name	WORD
	Device	DWORD
Data 2	<input type="checkbox"/> Type	LWORD
	<input type="checkbox"/> Name	SINT
	Device	INT
Data 3	<input type="checkbox"/> Type	DINT
	<input type="checkbox"/> Name	LINT
	Device	USINT
Data 4	<input type="checkbox"/> Type	UINT
	<input type="checkbox"/> Name	UDINT
	Device	ULINT
	<input type="checkbox"/> Name	REAL
	<input type="checkbox"/> Device	LREAL
	<input type="checkbox"/> Type	STRING

第11章数据记录功能

(4) 多种 OR 条件

[触发设定]为止，方法和现有的[多种 AND演算]方法相同。

1) 选择一个条件设定菜单，输入详细设定值。

[多种条件]通过[单一条件]演算方式，触发条件发生时，按照数据保存方式，基本设定方法如下，和单一条件相同。

Trigger settings

Conditions set

Bit condition

Device: %MXD

Word condition

Device: [] < [] 0

OK Cancel

Trigger settings

Conditions set

Bit condition

Device: %MX0

Word condition

Device: %MW100 < 5d

OK Cancel

设定结束，窗口关闭时，最初触发设定条件菜单中设定的条件如下显示

Trigger settings

Trigger condition: Single Combination

Combination Condition Setting

AND OR

Trigger occurrence condition

Rising Falling

Target Device

	Device	Type	Condition
1			
2			
3			
4			

Sample Block settings

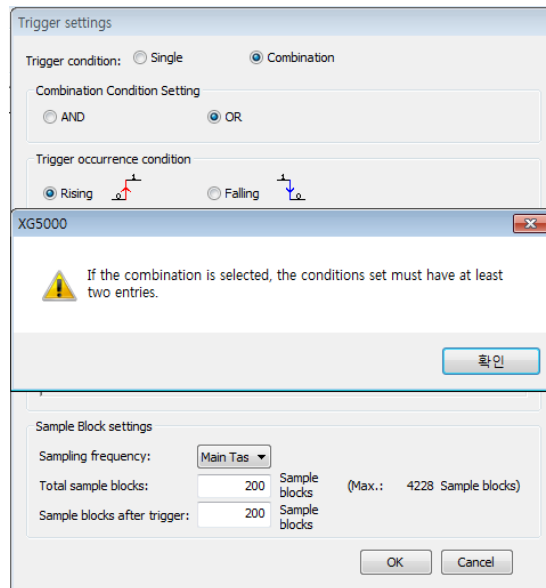
Sampling frequency: Main Tas

Total sample blocks: 200 Sample blocks (Max.: 4228 Sample blocks)

Sample blocks after trigger: 200 Sample blocks

OK Cancel

如果, 选择演算条件, 输入[条件设定], 设定后出现如下语句
则设定未完成.

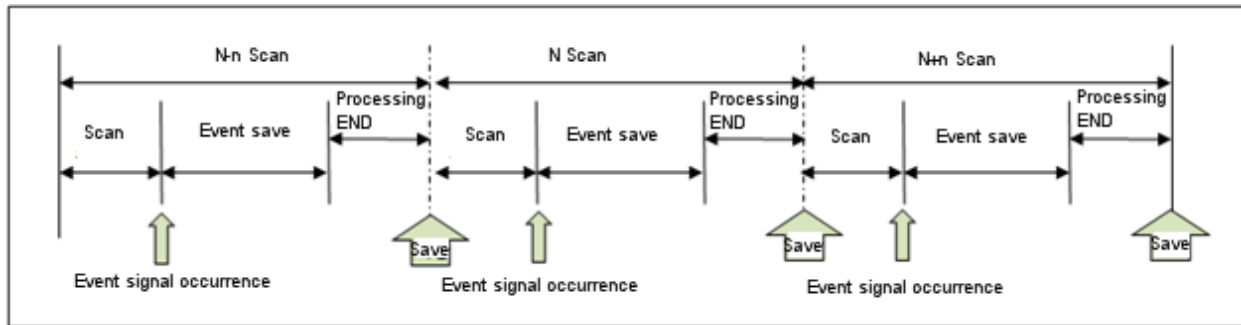


- 2) 采样周期, 总采样块数, 触发后采样块数输入后触发设定完成.
- 3) 数据记录基本设定窗口中设定的设备值在触发条件发生时, 变为数据收集后设定的类型, 在 SD内存中保存

Parameter	Group 0
Data 0	Type: NONE
	Name: NONE
	Device: BOOL
Data 1	Type: BYTE
	Name: WORD
	Device: DWORD
Data 2	Type: SINT
	Name: INT
	Device: DINT
Data 3	Type: LINT
	Name: USINT
	Device: UINT
Data 4	Type: UDINT
	Name: ULINT
	Device: REAL
	Device: LREAL
	Device: STRING

11.6 事件保存

事件保存是检测收集的设备值并且在满足事件条件时，在当前时点的数据进行采样的功能。即，事件发生时点开始到事件解除为止，想要分析数据采样，事件值变动和时点时使用的方式。



提示

[事件保存] 设定后，最初事件条件发生，数据收集期间又发生其他事件条件的话，忽略相应事件

11.6.1 事件条件

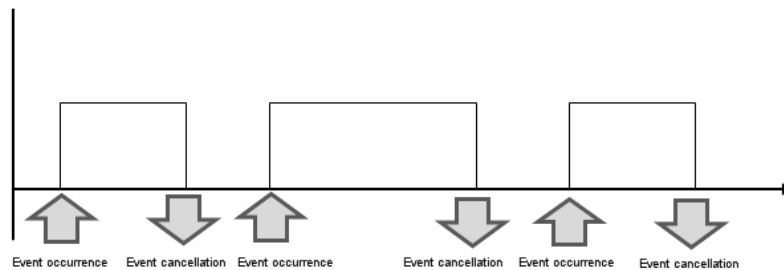
事件保存功能是根据单一条件，演算条件来运行的。单一条件，演算条件设定项目如下。多种条件是取单一条件来演算来运行的方法，最多可设定 4 个的单一条件。事件条件发生，数据保存开始时，最初数据列中插入‘E’文字列告知事件发生

(1) 单一条件

单一条件大致分为 [位条件], [字条件] 运行。

1) 位条件

位条件查看设定的设备bit值，值[上升], [下降], [转移], [ON], [OFF]时，数据收集



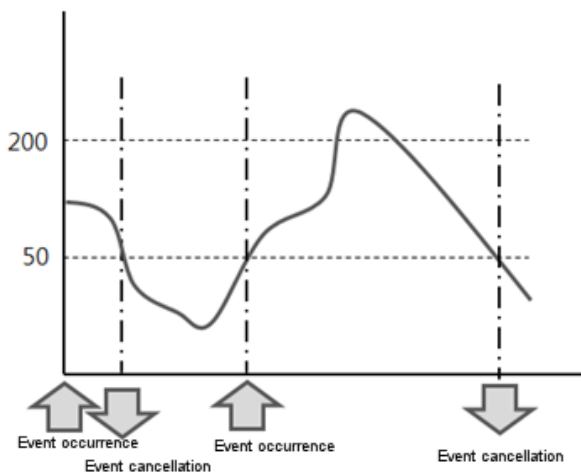
提示

[上升], [下降], [转移] 条件边沿运行, [ON],[OFF] 条件是水平运行。
请确认这点后设定

2) 字条件

字条件和设定的设备输入值比较，转换为TRUE, FALSE。满足设定的设备值输入条件，相应变换位值的[上升], [下降], [转移], [ON], [OFF] 时数据保存。

例) 设定值>50, 上升条件



提示

字条件满足与否的判断是否按UINT值基准。以及用户程序中设定设备的变量型是INT时-即使输入100也会在数据记录功能中取0xFF9C，请注意这点。

(例: %MW0整数型设定，字条件设定为正数时，即使%MWO中输入负数，也会判断为满足事件条件，所以请注意。)

3) 解除值设定

事件保存功能中解除值设定仅在字条件中可设定，根据使用会和数据保存周期频率不同。若设定解除值，事件发生之后条件会在解除值为止，都进行数据保存。

	解除值使用	解除值不使用
%MW0 > 100	解除值: 50 事件发生后到设定值 50 为止， 都会数据保存	满足事件发生后条件为止，都会数据保存
%MW0 >= 100	解除值不能设定	
%MW0 == 100	解除值不能设定	
%MW0 < 100	解除值: 120 事件发生后到设定值 120 为止， 都会数据保存	
%MW0 <= 100	解除值不能设定	
%MW0 <> 100	解除值不能设定	

提示

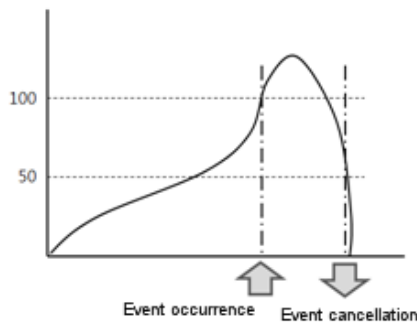
解除值可以如下设定，若违反则会发生错误，无法输入。参数设定时请确认。

解除值和设定值范围不可重复

条件	解除值范围
>	设定值 >= 解除值
≥	设定值 > 解除值
<	设定值 <= 解除值
≤	设定值 < 解除值
相同	不能设定
不一致	

例1) 字条件中 %MW0>100设定, 解除值设定为 50时,

%MW0超出100, 事件发生, 数据虽然保存, 但是由于解除值设定为 50, 在到50为止, 都会数据保存.

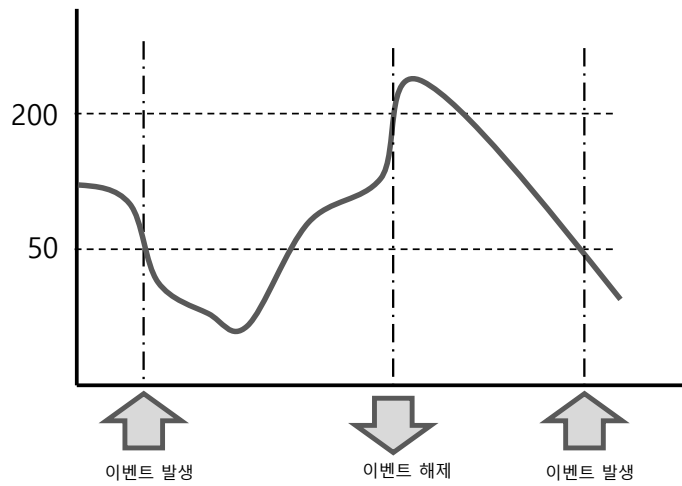


满足相应条件时, 事件发生条件在设定的时点上数据保存。

[上升],[下降],[转移] 条件通过运行保存数据 1块, [ON],[OFF] 条件按水平运行, 不满足相应条件为止, 都会进行数据保存。

例2) 字条件中设定为 %MW0<50, 解除值设定为 200时,

%MW0不足 50时, 事件发生, 数据虽然保存, 但由于解除值为 200, 直到满200为止, 都会数据保存



4) 条件说明




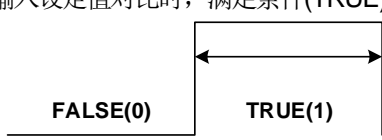
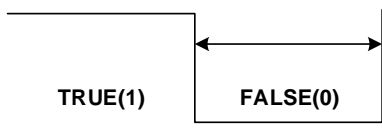
	事件发生条件	设备设定条件	运行	解除值设定与否
位条件	上升		设定设备位从 FALSE(0) 变为 TRUE(1) 时，数据保存。 FALSE → TRUE 	
	下降		设定设备位从 TRUE(1) 变为 FALSE(0) 时，数据保存。 TRUE → FALSE 	
	转移		设定设备位才 FALSE(0) 变为 TRUE(1) 或者 TRUE(1) 变为 FALSE(0) 时，数据保存。 FALSE → TRUE TRUE → FALSE 或者 	
	ON		设定设备位为 TRUE(1) 时，期间数据保存。 ON 	
	OFF		设定设备位为 FALSE(0) 时，期间数据保存。 OFF 	




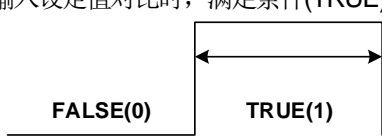
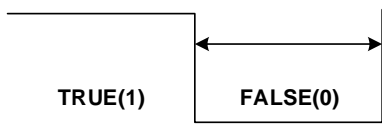
	事件发生条件	设备设定条件	运行	解除值设定与否
字条件	上升		设定字设备值和输入设定值对比时，不满足条件时(FALSE)变为满足(TRUE)，数据采样。 <p>FALSE(0) TRUE(1)</p> <p>Ex) Ex)</p> <p>device value >= set value device value < set value</p> <p>device value = set value</p> <p>device value > set value</p>	设定变量
	下降		设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE)时，数据采样。 <p>TRUE(1) FALSE(0)</p> <p>Ex) Ex)</p> <p>device value < set value device value >= set value</p> <p>device value = set value</p> <p>device value <= set value</p>	
	转移	<	设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE) 或者不满足(FALSE)变为满足(TRUE)时，数据采样。 <p>FALSE(0) TRUE(1) TRUE(1) FALSE(0)</p> <p>Ex) Ex) Ex) Ex)</p> <p>device value >= set value device value < set value device value < set value device value >= set value</p> <p>device value = set value</p> <p>device value > set value</p> <p>或者</p> <p>device value >= set value</p> <p>device value = set value</p> <p>device value <= set value</p>	
	ON		设定字设备值和输入设定值对比时，满足条件(TRUE) 期间数据采样。 <p>FALSE(0) TRUE(1)</p> <p>Ex) Ex)</p> <p>Device value ≥ Set value Device value < Set value</p> <p>Device value = Set value</p> <p>Device value > Set value</p>	
	OFF		设定字设备值和输入设定值对比时，不满足条件(FALSE) 期间数据采样。 <p>TRUE(1) FALSE(0)</p> <p>Ex) Ex)</p> <p>Device value < Set value Device value ≥ Set value</p> <p>Device value = Set value</p> <p>Device value > Set value</p>	

	事件发生条件	设备设定条件	运行	解除值设定与否
字条件	上升	≤	<p>设定字设备值和输入设定值对比时，不满足条件时(FALSE)变为满足(TRUE)，数据采样。</p> <p>Ex) device value ≥ set value device value = set value device value > set value</p> <p>Ex) device value < set value</p>	
	下降		<p>设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE)时，数据采样。</p> <p>Ex) device value < set value</p> <p>Ex) device value ≥ set value device value = set value device value ≤ set value</p>	
	转移		<p>设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE) 或者不满足(FALSE)变为满足(TRUE)时，数据采样。</p> <p>Ex) device value ≥ set value device value = set value device value > set value</p> <p>Ex) device value < set value</p> <p>或者</p> <p>Ex) device value ≥ set value device value = set value device value ≤ set value</p>	
	ON		<p>设定字设备值和输入设定值对比时，满足条件(TRUE) 期间数据采样。</p> <p>Ex) Device value ≥ Set value Device value = Set value Device value > Set value</p> <p>Ex) Device value < Set value</p>	
	OFF		<p>设定字设备值和输入设定值对比时，不满足条件(FALSE) 期间数据采样。</p> <p>Ex) Device value < Set value</p> <p>Ex) Device value ≥ Set value Device value = Set value Device value > Set value</p>	

	事件发生条件	设备设定条件	运行	解除值设定与否
字条件	上升		设定字设备值和输入设定值对比时，不满足条件时(FALSE)变为满足(TRUE)，数据采样。 	
	下降		设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE)时，数据采样。 	
	转移	>	设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE) 或者不满足(FALSE)变为满足(TRUE)时，数据采样。 	
	ON		设定字设备值和输入设定值对比时，满足条件(TRUE) 期间数据采样。 	
	OFF		设定字设备值和输入设定值对比时，不满足条件(FALSE) 期间数据采样。 	

	事件发生条件	设备设定条件	运行	解除值设定与否
字条件	上升		<p>设定字设备值和输入设定值对比时，不满足条件时(FALSE)变为满足(TRUE)，数据采样。</p> <p>FALSE(0) → TRUE(1)</p> <p>Ex) device value >= set value device value = set value device value > set value</p> <p>Ex) device value < set value</p>	
	下降		<p>设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE)时，数据采样。</p> <p>TRUE(1) → FALSE(0)</p> <p>Ex) device value < set value</p> <p>Ex) device value >= set value device value = set value device value <= set value</p>	
	转移	≥	<p>设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE) 或者不满足(FALSE)变为满足(TRUE)时，数据采样。</p> <p>FALSE(0) → TRUE(1) TRUE(1) → FALSE(0)</p> <p>Ex) device value >= set value Ex) device value < set value</p> <p>Ex) device value = set value Ex) device value >= set value</p> <p>Ex) device value > set value Ex) device value = set value</p> <p>或者 Ex) device value <= set value</p>	
	ON		<p>设定字设备值和输入设定值对比时，满足条件(TRUE) 期间数据采样。</p> <p>FALSE(0) TRUE(1)</p> <p>Ex) Device value ≥ Set value Device value = Set value Device value > Set value</p> <p>Ex) Device value < Set value</p>	
	OFF		<p>设定字设备值和输入设定值对比时，不满足条件(FALSE) 期间数据采样。</p> <p>TRUE(1) FALSE(0)</p> <p>Ex) Device value < Set value</p> <p>Ex) Device value ≥ Set value Device value = Set value Device value > Set value</p>	

	事件发生条件	设备设定条件	运行	解除值设定与否
字条件	上升	相同	设定字设备值和输入设定值对比时，不满足条件时(FALSE)变为满足(TRUE)，数据采样。  Ex) device value >= set value device value = set value device value > set value	
	下降		设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE)时，数据采样。  Ex) device value < set value Ex) device value >= set value device value = set value device value <= set value	
	转移		设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE) 或者不满足(FALSE)变为满足(TRUE)时，数据采样。  Ex) device value >= set value device value = set value device value > set value 或者 Ex) device value < set value Ex) device value >= set value device value = set value device value <= set value	
	ON		设定字设备值和输入设定值对比时，满足条件(TRUE) 期间数据采样。  Ex) Device value ≥ Set value Device value = Set value Device value > Set value Ex) Device value < Set value	
	OFF		设定字设备值和输入设定值对比时，不满足条件(FALSE) 期间数据采样。  Ex) Device value < Set value Ex) Device value ≥ Set value Device value = Set value Device value > Set value	

	事件发生条件	设备设定条件	运行	解除值设定与否
字条件	上升	不一致	设定字设备值和输入设定值对比时，不满足条件时(FALSE)变为满足(TRUE)，数据采样。  Ex) device value >= set value device value = set value device value > set value	
	下降		设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE)时，数据采样。  Ex) device value < set value device value >= set value device value = set value device value <= set value	
	转移		设定字设备值和输入设定值对比时，满足条件时(TRUE)，变为不满足条件(FALSE) 或者不满足(FALSE)变为满足(TRUE)时，数据采样。  Ex) device value >= set value device value = set value device value > set value 或者 Ex) device value < set value device value >= set value device value = set value device value <= set value	
	ON		设定字设备值和输入设定值对比时，满足条件(TRUE) 期间数据采样。  Ex) Device value ≥ Set value Device value = Set value Device value > Set value Ex) Device value < Set value	
	OFF		设定字设备值和输入设定值对比时，不满足条件(FALSE) 期间数据采样。  Ex) Device value < Set value Ex) Device value ≥ Set value Device value = Set value Device value > Set value	

第11章数据记录功能

(2) 多种条件

多种条件是单一条件最多可设置4个，取符合条件的演算来运行的方法。
按照设定的条件来演算，满足结果时，事件条件发生

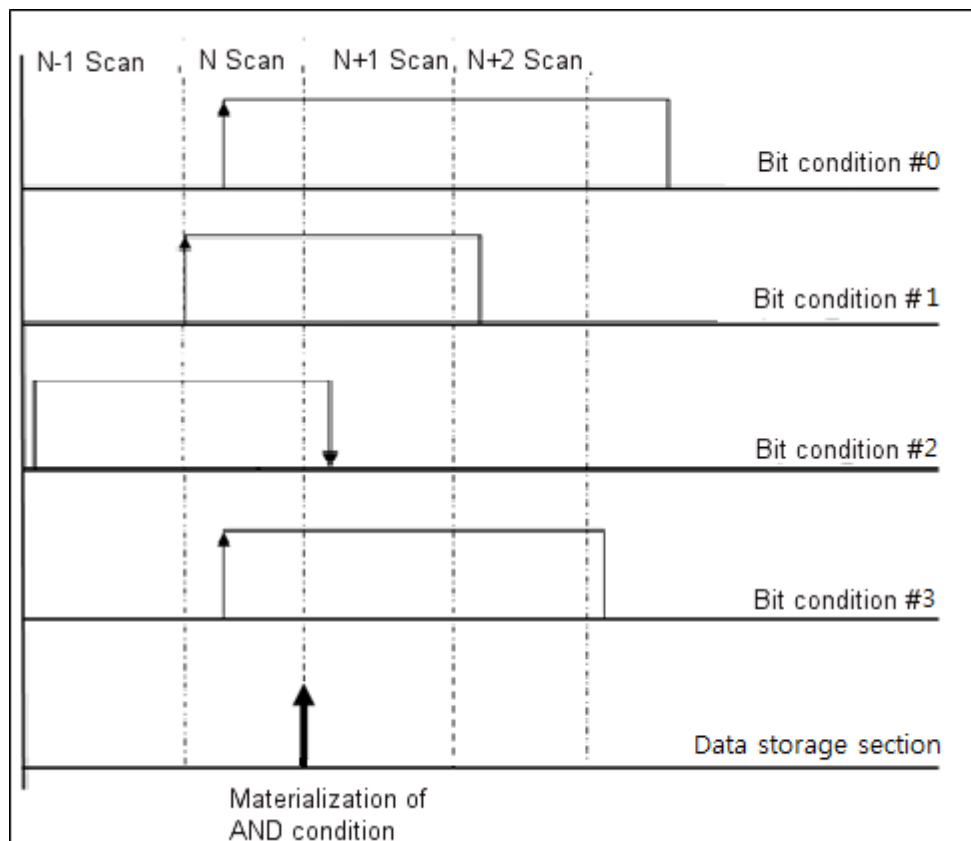
设定内容	动作	备注
AND 条件	设定的条件进行 AND 演算，结果值满足 1时，数据保存	
OR 条件	设定的条件进行 OR 演算，结果值满足 1时，数据保存	

1) AND 演算

在相同主任务周期内，满足所有相应条件时，事件发生。以下是事件保存开始的例子

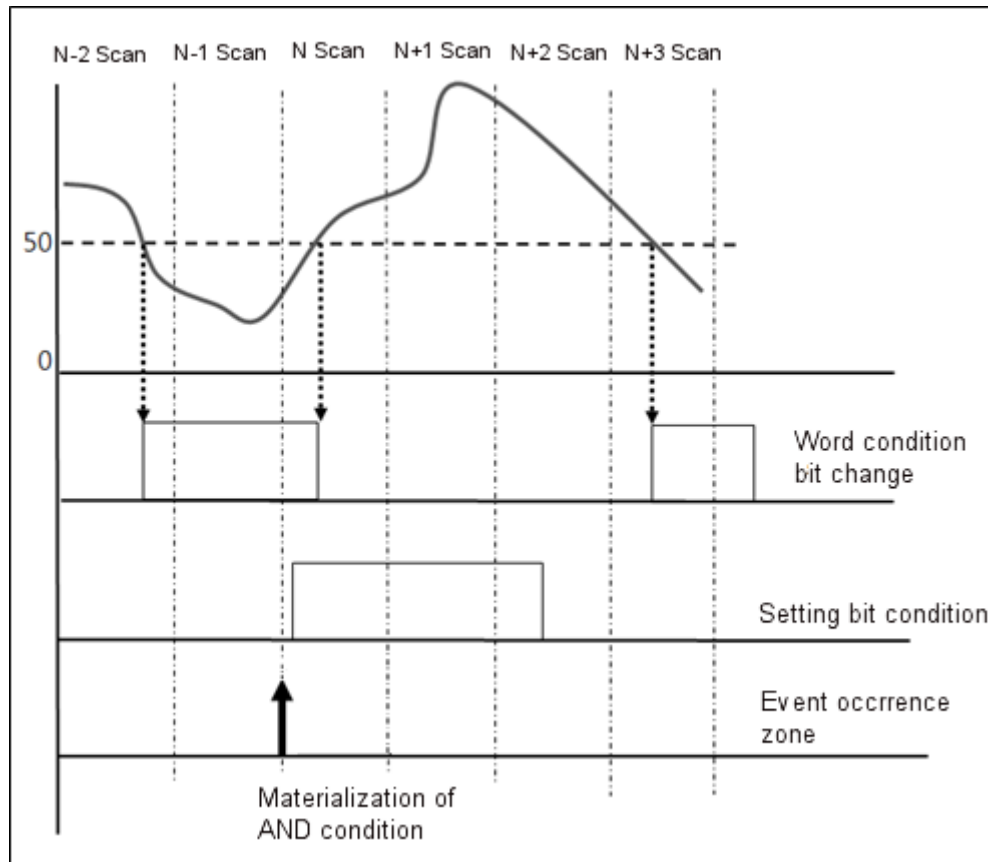
☞ 仅位条件设定

	条件	设定设备	事件发生条件
0号条件	位	%MX1010	上升
1号条件	位	%IX1	
2号条件	位	%MX2010	
3号条件	位	%QX130	



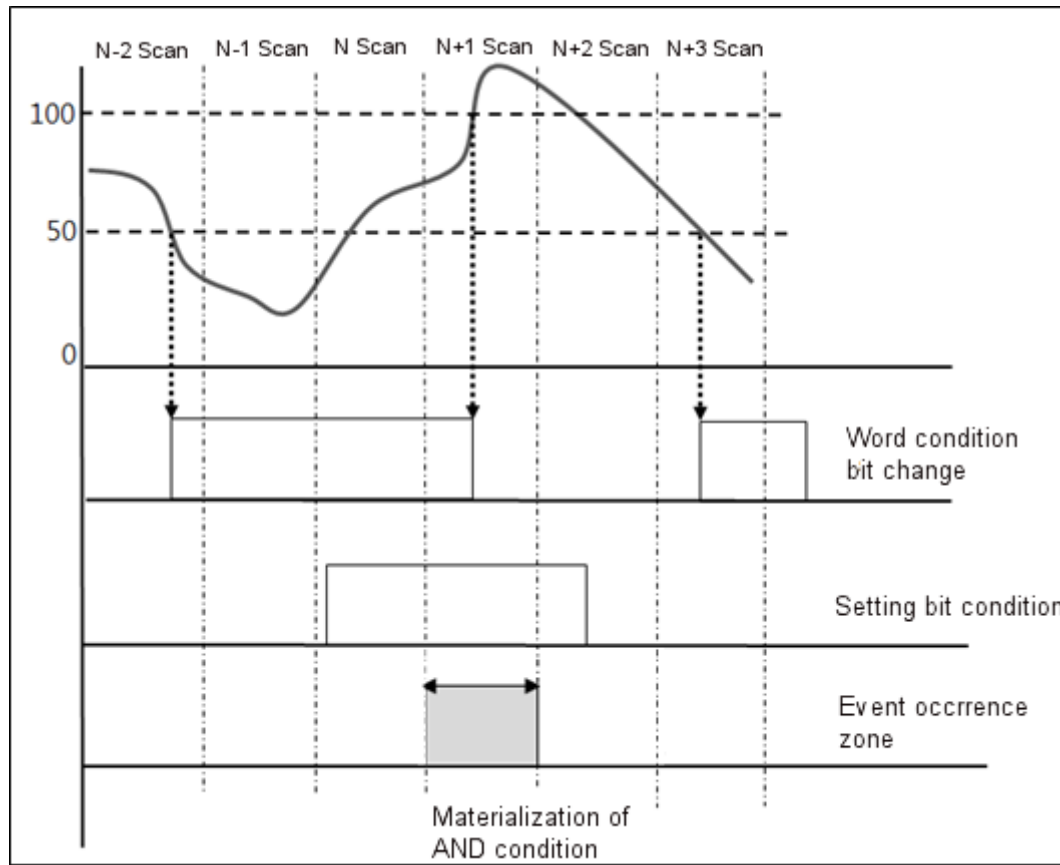
位, 字条件混合设定时 (解除值不设定)

	条件	比较条件	设定值	解除值	设定设备	事件发生条件
0号条件	字	<	50	-	%MW100	上升
1号条件	位				%MX15	



位, 字条件混合设定时 (解除值设定)

	条件	比较条件	设定值	解除值	设定设备	事件发生条件
0号条件	字	<	50	100	%MW100	ON
1号条件	位				%MX15	

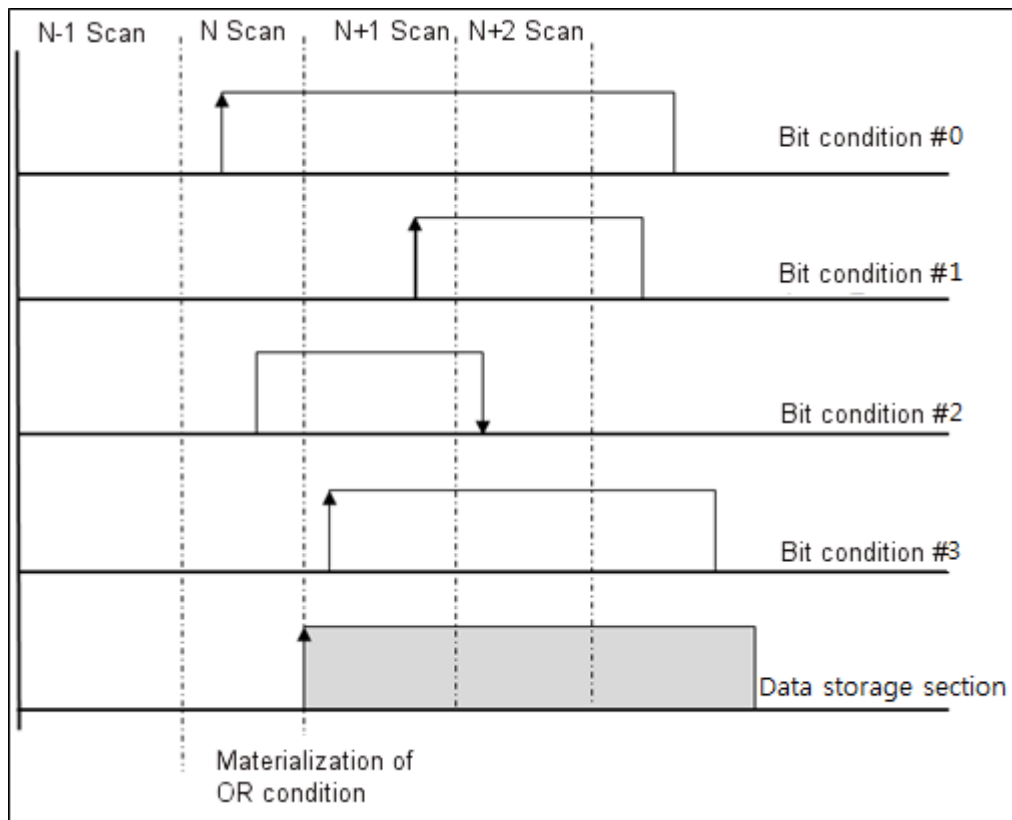


2) OR 演算

在主任务周期，即使只满足1个相应条件时，事件发生。事件发生之后数据保存完成前，再次满足事件条件时相应标志位忽略。

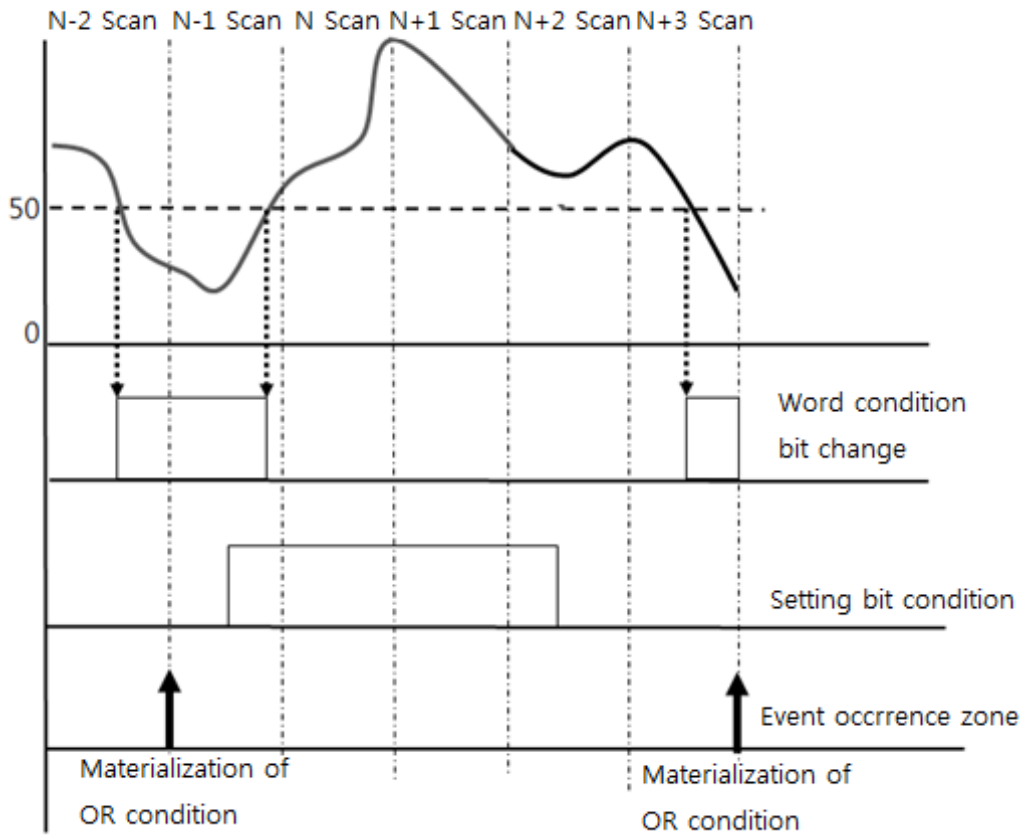
☞ 仅位条件设定时

	条件	设定设备	事件发生条件
0号条件	位	%MX1010	ON
1号条件	位	%IX1	
2号条件	位	%MX2010	
3号条件	位	%QX130	



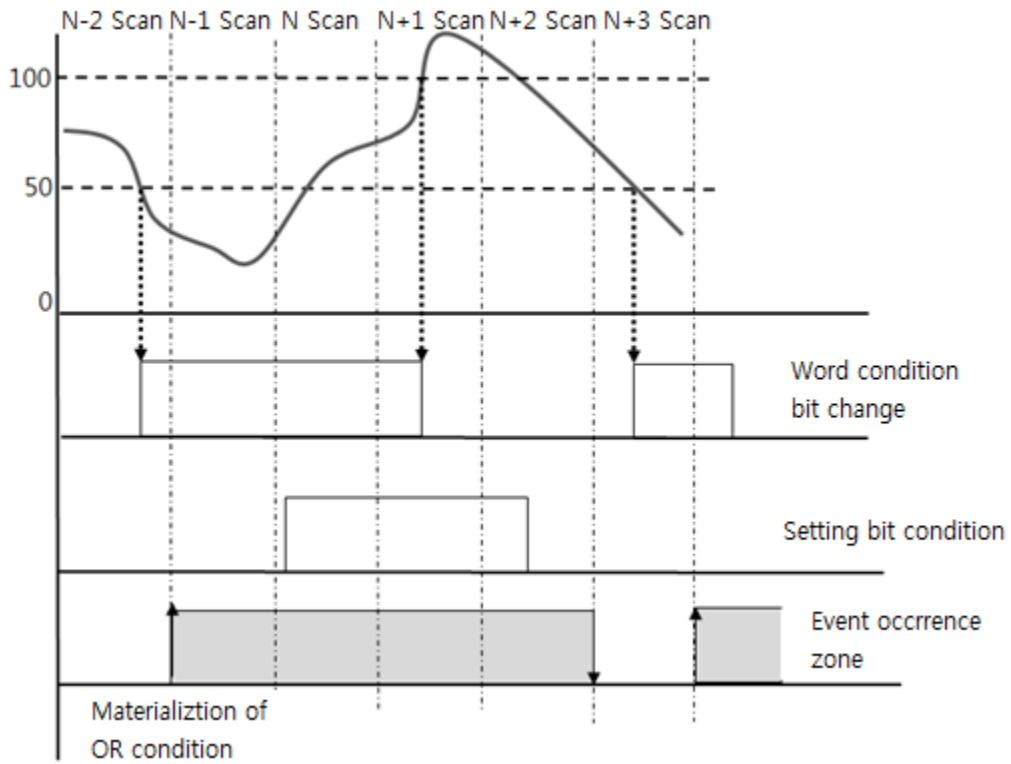
位, 字条件混合设定时 (解除值不设定)

	条件	比较条件	设定值	解除值	设定设备	事件发生条件
0号条件	字	<	50	-	%MW10	上升
1号条件	位				%MX15	



位, 字条件混合设定时 (解除值设定)

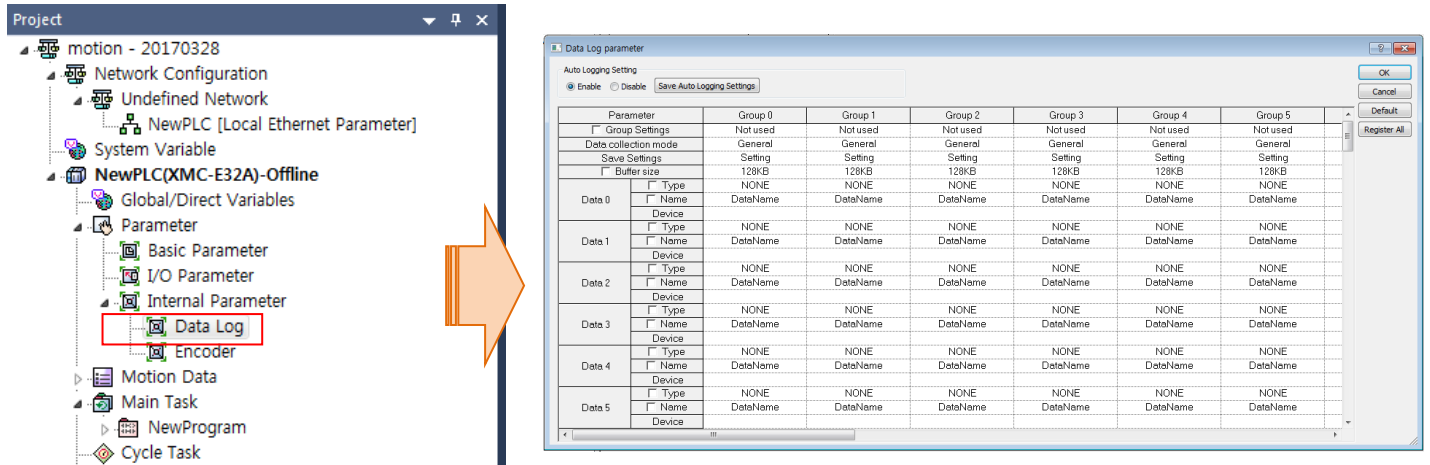
	条件	比较条件	设定值	解除值	设定设备	事件发生条件
0号条件	字	<	50	100	%MW10	ON
1号条件	位				%MX15	



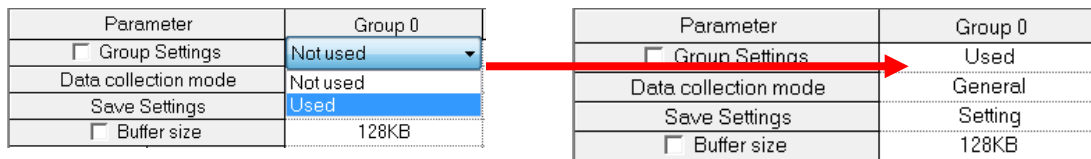
11.6.2 设定方法

(1) 单一条件

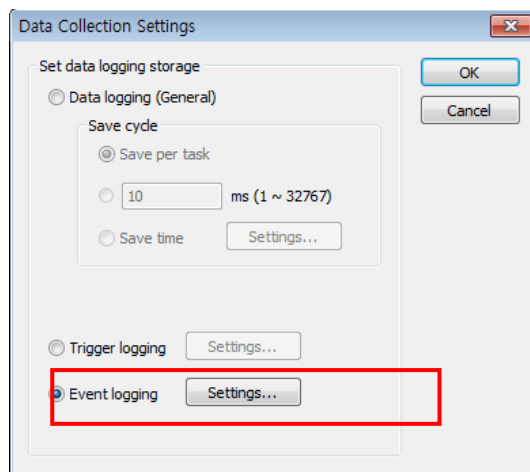
- 1) MP500-[工程窗]-[内置参数]-[数据记录]选择
如上选择后，数据记录参数设定窗激活



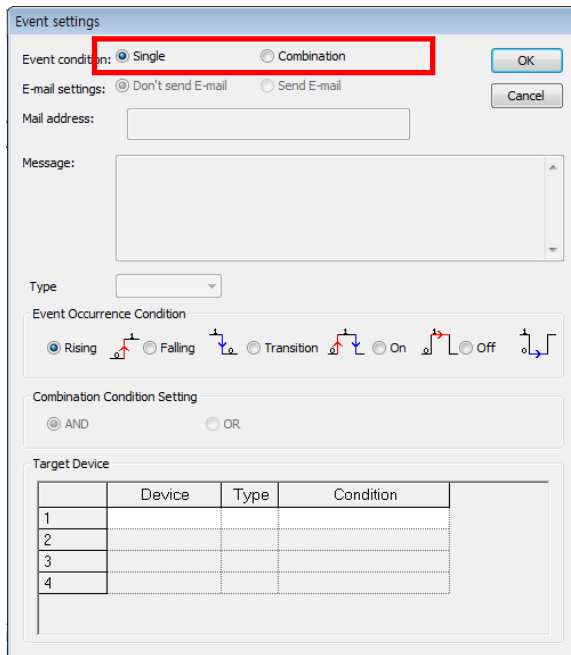
- 2) 数据记录参数窗中设定要使用的组



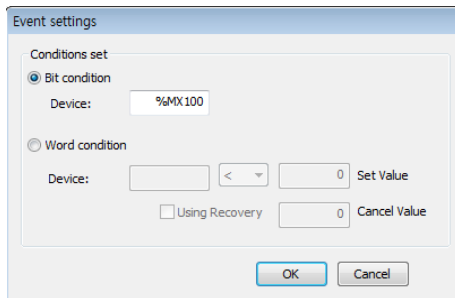
- 3) 在[数据收集方式设定]中选择 [事件记录], 左侧 [设定]菜单激活.
此时, 左侧[设定] 菜单选择



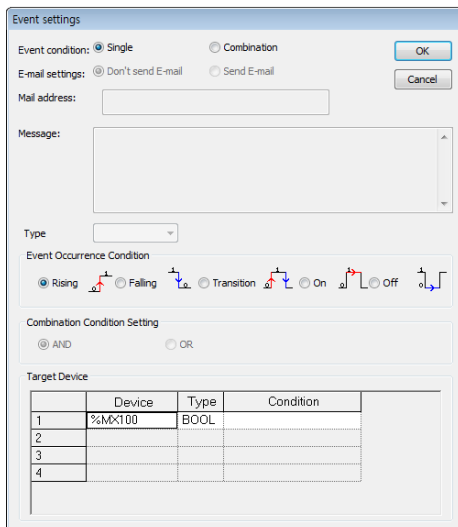
- 4) 选择[设定] 菜单, 可进行事件设定的窗口如下激活。
事件条件选择 [单一条件]



- 5) 选择条件设定菜单, 如下图, 窗口激活。
选择[位条件]后, 设备窗中的位类型输入设备值。

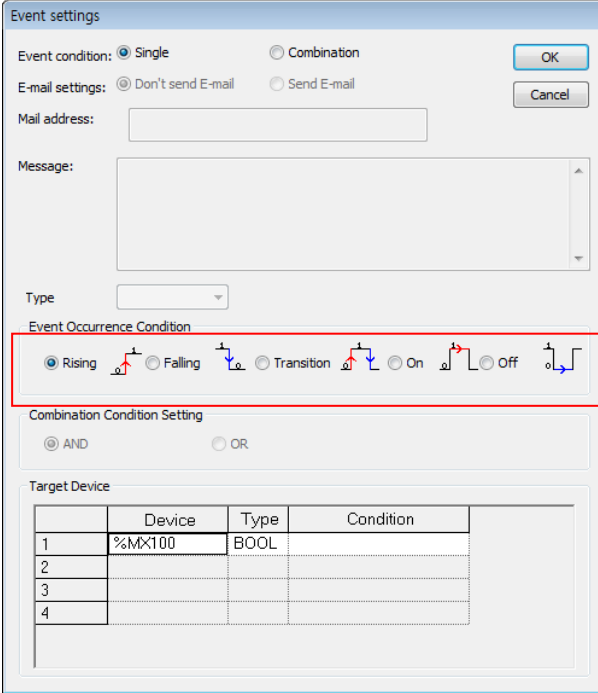


设定完成, 窗关闭时, 最初事件设定条件菜单中的设定条件如下显示。



第11章数据记录功能

事件发生条件项目中选择数据保存时点。
根据相应设定值，数据个数和时点会不同。

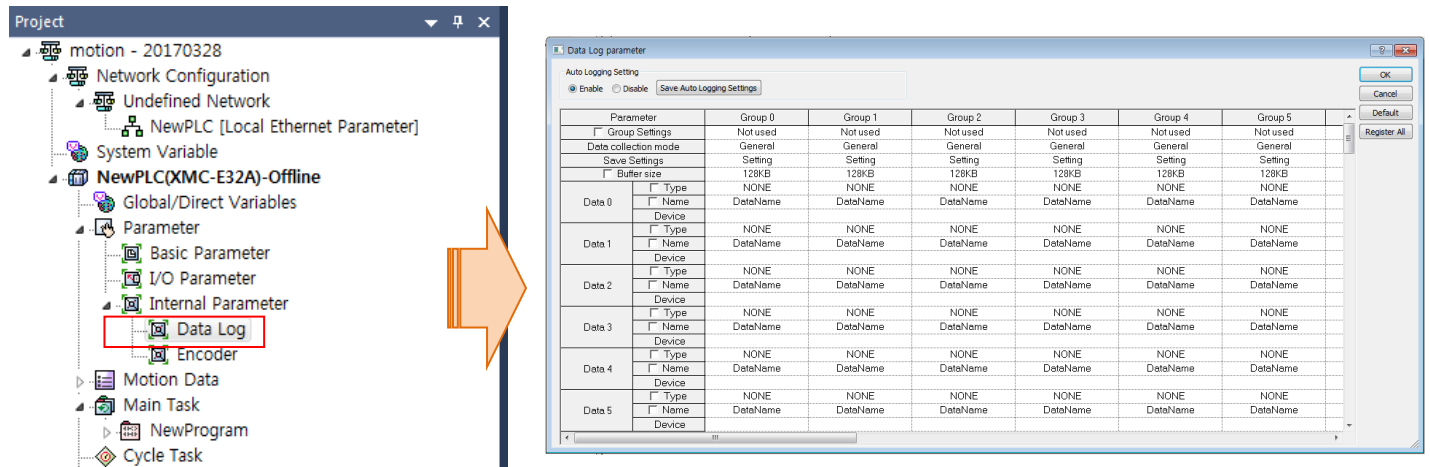


The image shows a software dialog box titled "Event settings". It contains several sections for configuring event recording parameters. The "Event Occurrence Condition" section is highlighted with a red box and includes radio buttons for "Rising", "Falling", "Transition", "On", and "Off", each accompanied by a small waveform icon. Below this is the "Combination Condition Setting" section with "AND" and "OR" options. The "Target Device" section contains a table with columns for "Device", "Type", and "Condition".

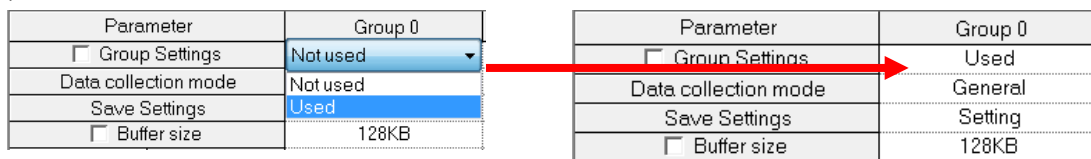
	Device	Type	Condition
1	%MX100	BOOL	
2			
3			
4			

(2) 单一字条件

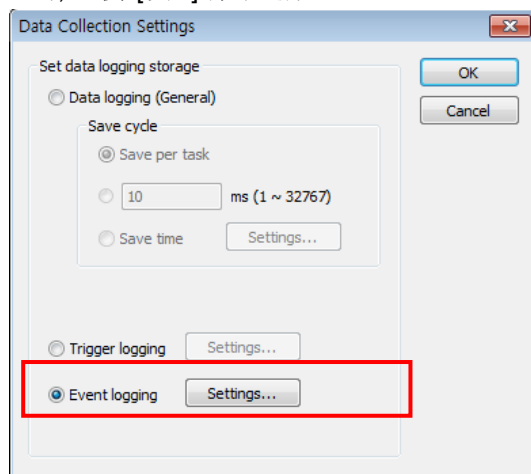
- 1) MP500 -[工程窗]- [内置参数]- [数据记录]选择
如上选择后，数据记录参数设定窗激活



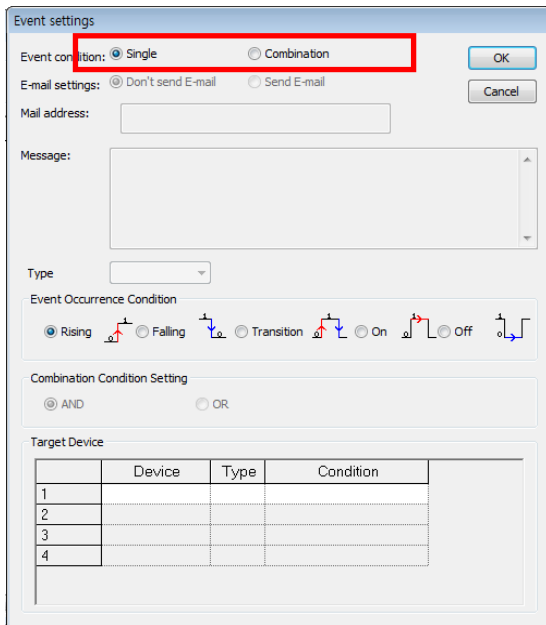
2) 数据记录参数窗中设定要使用的组



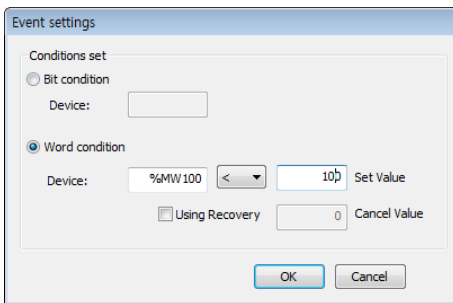
- 3) [数据收集方式设定]中选择[事件 logging]，左侧 [设定]菜单激活。
此时，左侧 [设定] 菜单选择



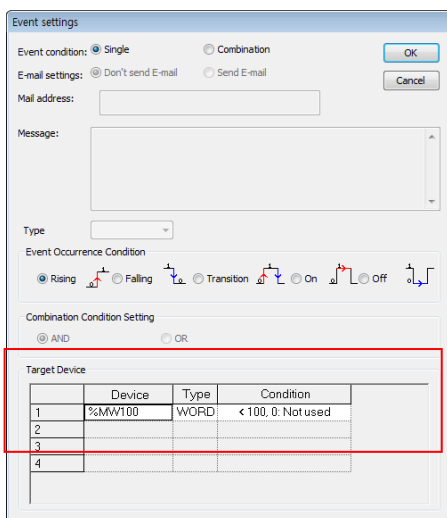
- 4) 选择[设定] 菜单, 可进行事件设定的窗口如下激活。
事件条件选择[单一条件].



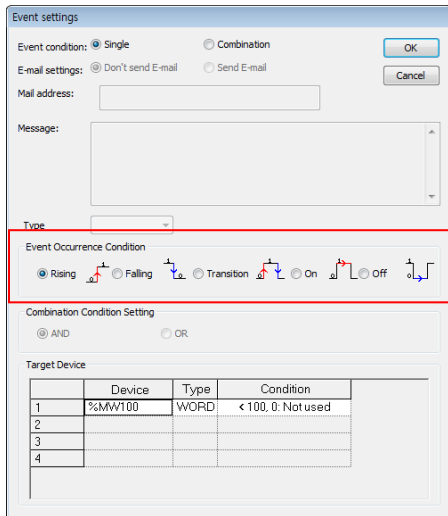
- 5) 选择条件设定菜单, 如下图, 窗口激活。
选择[字条件]后, 设备窗中输入设备值.



设定完成, 窗关闭时, 最初事件设定条件菜单中的设定条件如下显示.

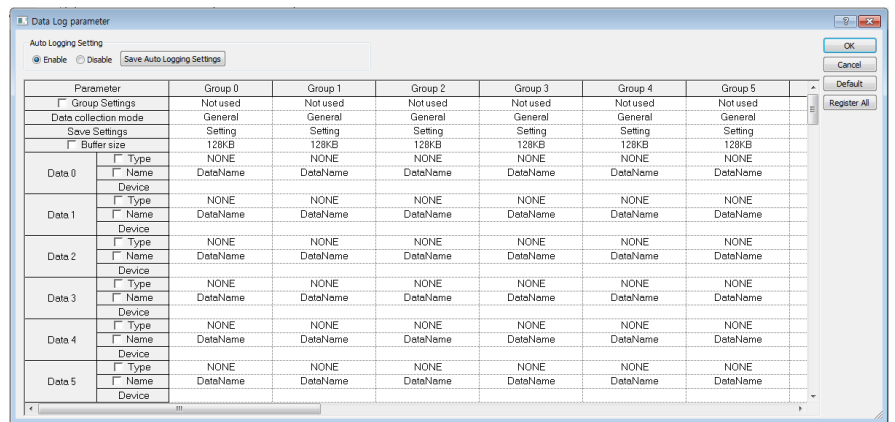
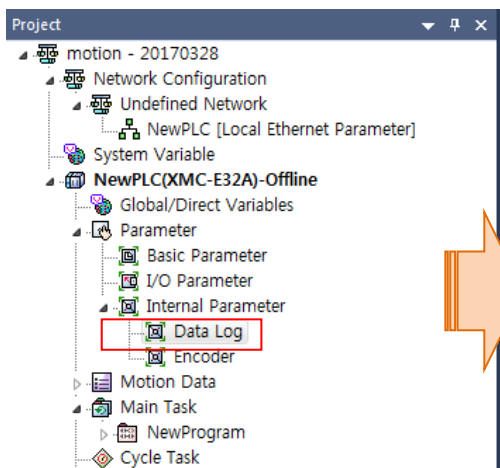


6) 事件发生条件项目中选择数据保存时点。根据相应设定值，数据个数和时点不同



(3) 多种 AND 条件

- 1) MP500 -[工程窗] - [内置参数] - [数据记录]选择
如上选择后，数据记录参数设定窗激活

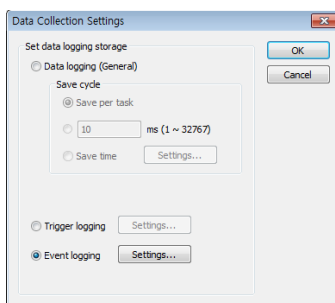


2) 数据记录参数窗中设定要使用的组

Parameter	Group 0
<input type="checkbox"/> Group Settings	Not used
Data collection mode	Not used
Save Settings	Used
<input type="checkbox"/> Buffer size	128KB

Parameter	Group 0
<input type="checkbox"/> Group Settings	Used
Data collection mode	General
Save Settings	Setting
<input type="checkbox"/> Buffer size	128KB

3) [数据收集方式设定]中选择 [事件记录]，左侧 [设定]菜单激活。
此时，左侧[设定]菜单选择



4) 选择[事件条件]-[多种条件]，条件设定窗最多可输入4个

Target Device			
	Device	Type	Condition
1			
2			
3			
4			

- 5) 事件发生条件项目中选择数据保存时点，演算条件设定按AND演算来设定。根据相应设定值，数据个数和时点不同。

Event settings

Event condition: Single Combination

E-mail settings: Don't send E-mail Send E-mail

Mail address:

Message:

Type:

Event Occurrence Condition

Rising Falling Transition On Off

Combination Condition Setting

AND OR

Target Device

	Device	Type	Condition
1			
2			
3			
4			

- 6) 选择一个条件设定菜单，输入具体设定值，[多种条件]按[单一条件]设定的演算方式计算，事件条件发生。基本设定方法和单一条件相同。

Event settings

Conditions set

Bit condition

Device:

Word condition

Device: < Set Value

Using Recovery Cancel Value

OK Cancel

Event settings

Conditions set

Bit condition

Device:

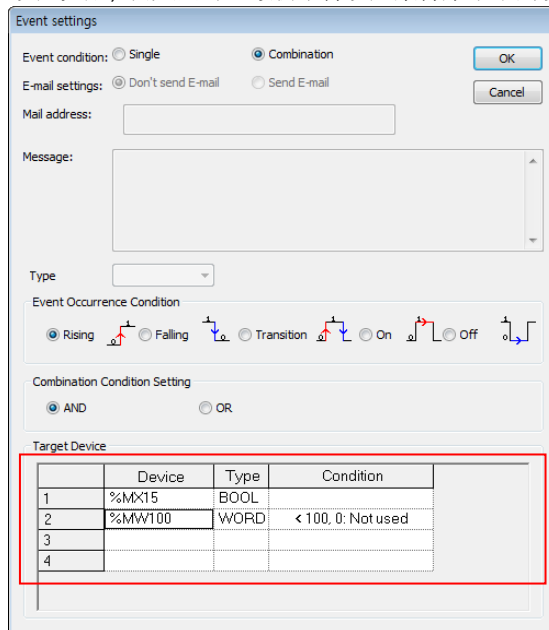
Word condition

Device: < Set Value

Using Recovery Cancel Value

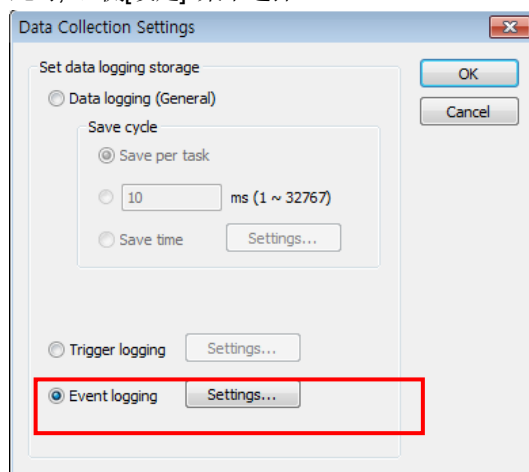
OK Cancel

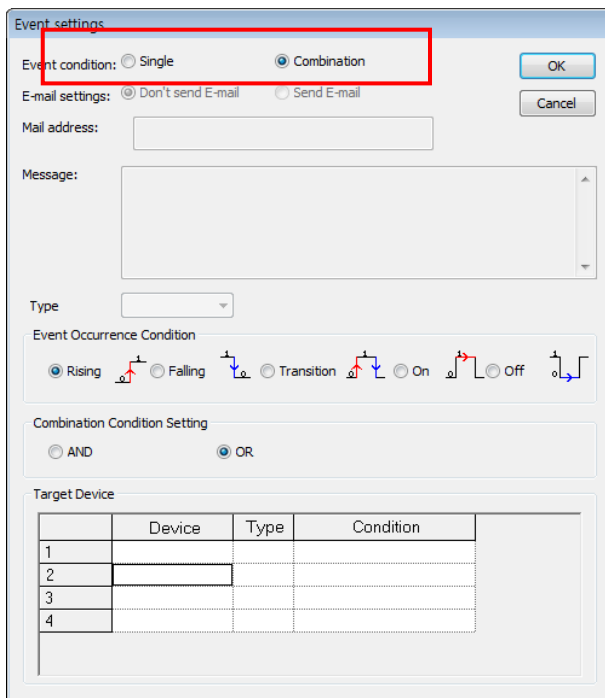
7) 设定完成, 窗关闭时, 最初事件设定条件菜单中的设定条件如下显示.



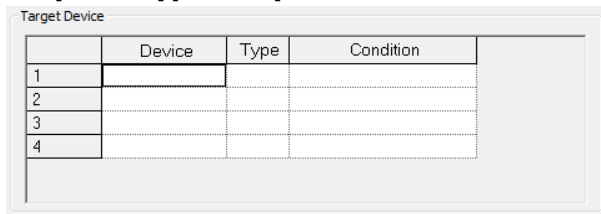
(4) 多种OR条件

- 1) 到[事件设定] 菜单为止的方法和 [AND演算条件]相同.
- 2) [数据收集方式]中选择[事件记录], 左侧[设定]菜单激活. 此时, 左侧[设定] 菜单选择

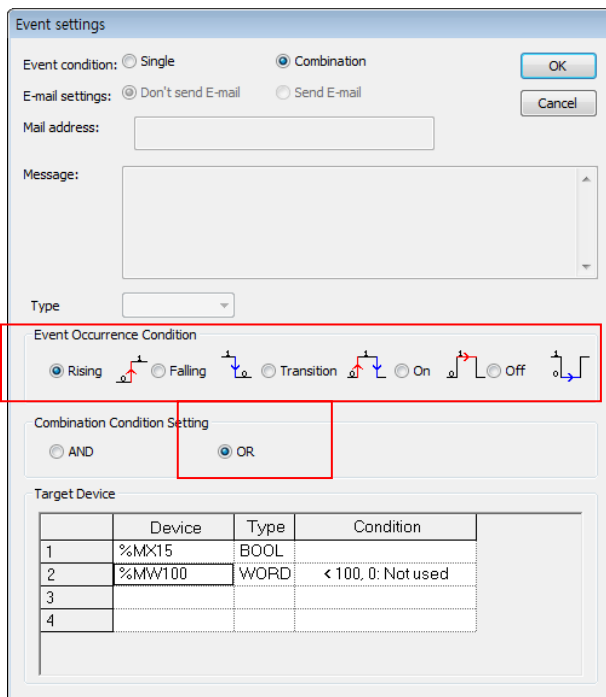




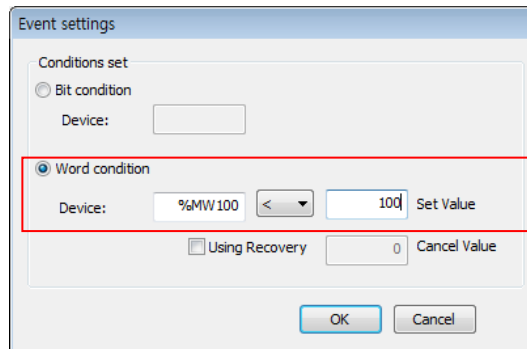
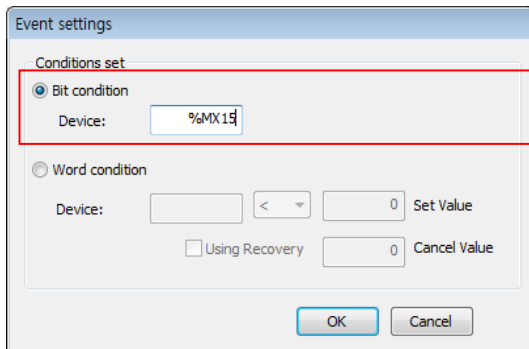
- 3) 选择[事件条件]-[多种条件]，条件设定窗中最多可输入 4个



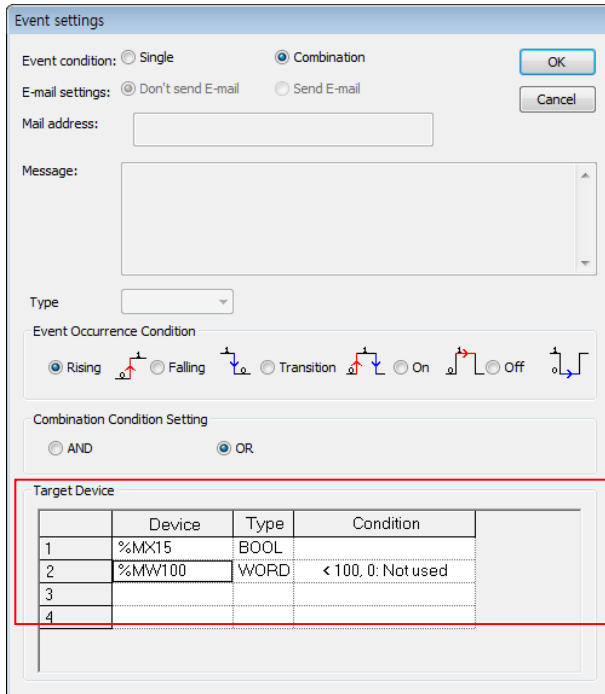
- 4) 事件发生条件项目中选择数据保存时点，演算条件设定按OR演算来设定。根据相应设定值，数据个数和时点不同。



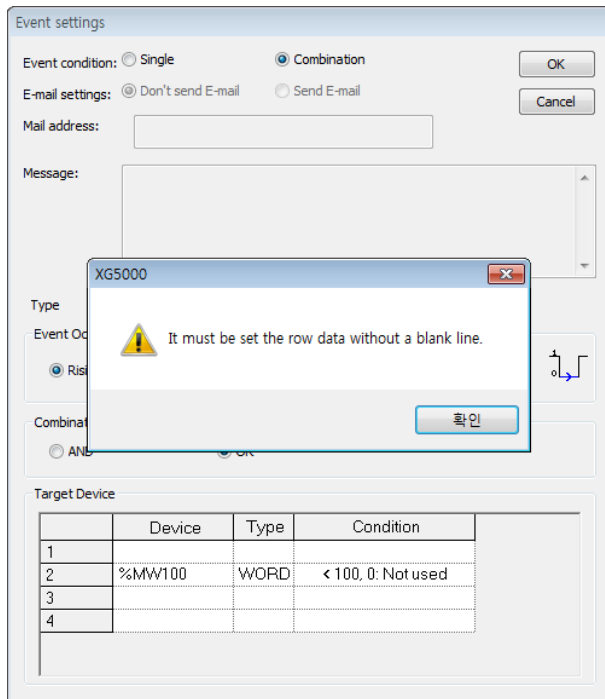
- 5) 选择一个条件设定菜单，输入具体设定值。[多种条件]按[单一条件]设定的演算方式计算，事件条件发生，基本设定方法和单一条件相同



6) 设定完成, 窗关闭时, 最初事件设定条件菜单中的设定条件如下显示.



7) 如果, 选择多种条件, [条件设定]输入, 设定完成后如下, 显示则设定未完成



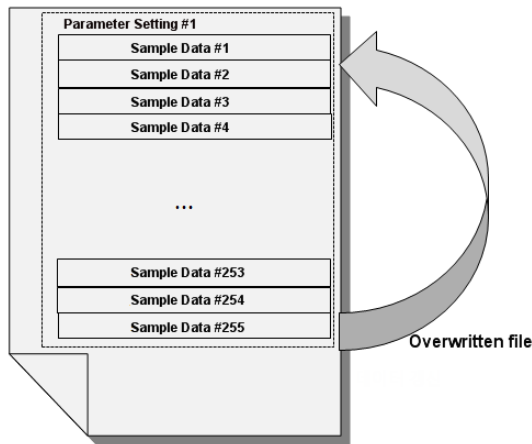
11.7 附加功能

对内置数据记录附加功能进行详细说明.

11.7.1 文件保存履历设定

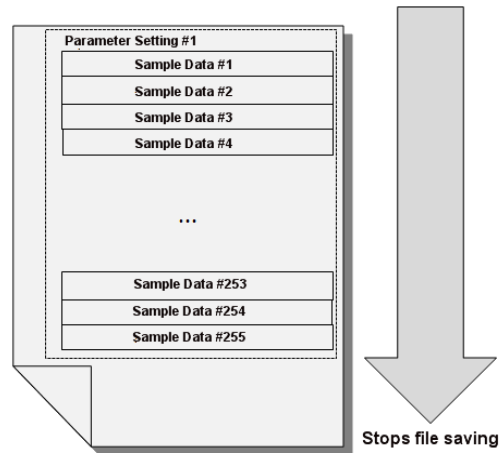
按数据记录中最大保存文件进行保存时, 根据[履历设定]中[覆盖最近履历], [最初履历维持]设定, 文件保存会有不同.

覆盖最近履历



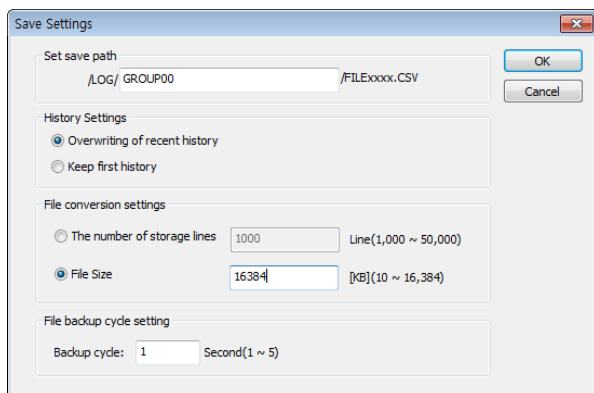
- 按最大保存文件数(256文件/文件夹) 数据保存的后, 维持最初的数据, 保存近期履历数据。
- [覆盖最近履历]设定后, 到保存最后文件数据为止, 文件保存超出次数, 标志位值增加

最初履历维持

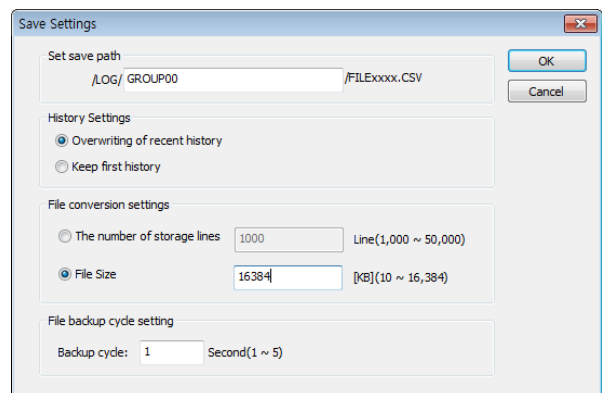


- 按最大保存文件数(256文件/文件夹), 数据保存的后的数据保存停止.
- 如果, SD内存容量剩余在10%以内, 数据保存停止.

设定方法



设定方法



11.7.2 格式化功能

内置数据记录支持 SD 内存格式化功能。SD 内存格式化是通过 MP500进行的。
仅在SD 内存格式化是运动控制STOP模式时支持。

(1) 格式化规格

数据记录中支持的SD内存格式化规格如下

项目	设定规格
文件系统 ¹⁾	FAT32
支持SD内存容量 ²⁾	2GByte ~ 32GByte
集群大小 ³⁾	32KByte (512sector ⁴⁾ * 8个)
卷标 ⁵⁾	LSIS (固定)
运行模式 ⁶⁾	STOP (可远程)
格式化模式 ⁷⁾	快速格式化

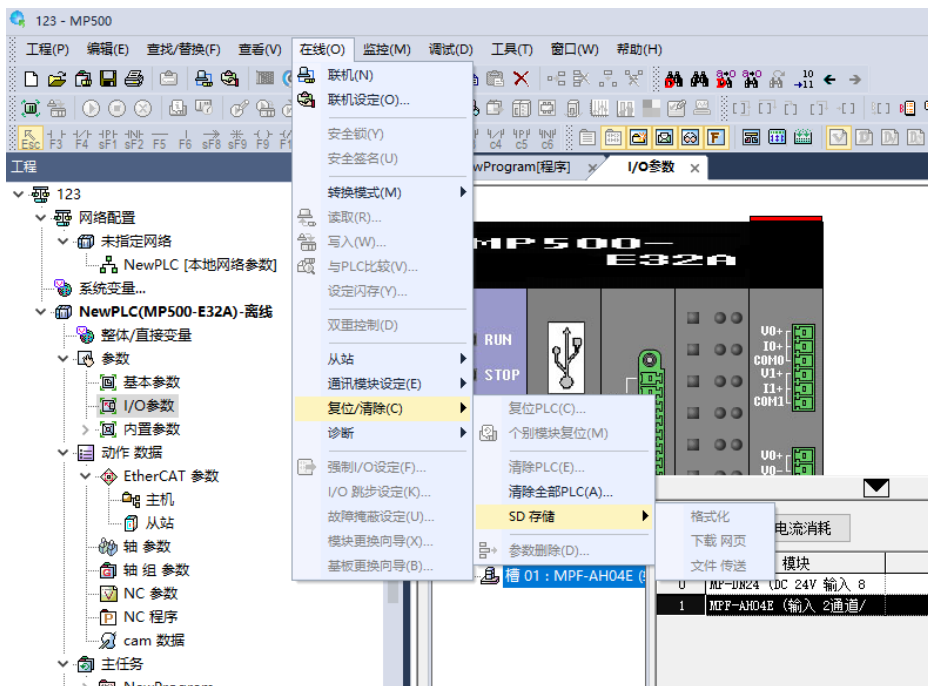
- 1) 文件系统: disk内文件保存规格
- 2) SD 内存容量: 2GByte~ 32GByte 容量的 micro SD 内存支持 (SD, SDHC支持)
- 3) 集群大小: 32KByte
- 4) 扇形: 数据保存最少单位 (基本: 512bite)
- 5) 卷标: SD 内存卡名称
- 6) 运动控制运行模式: 仅在STOP 模式下可运行
- 7) 格式化模式: 通过较快格式化进行SD 内存格式化。仅文件系统内的FAT, 目录区域的删除方式

提示

- 在运动控制中执行[格式化功能], SD内存内的所有内容都会删除, 之后参数以设定的名称生成文件夹。
- SD卡文件系统不是FAT32时, MP500不支持格式化功能。
不是FAT32, 要尝试SD卡格式化时, “FAT32 以外的文件系统不支持。” 警告窗口生成。
PC中请用FAT32来格式化后使用。

(2) 执行方法

1)MP500 -[在线] -[复位/清除] -[SD内存] -[格式化]选择



- 2) SD内存格式化功能执行前, 格式化过程中注意事项语句激活..
确认语句后, 如[例]进入下一阶段。

予的

- 格式化途中, SD 内存卡强制拔出或者电源 OFF 或者 Reset发生时, 连接的卡内部会发生损伤, 不能正常运行。
- SD 内存连接时识别中, 相应运行完成后, 可进行format。
SD 内存连接时, 请确认 RD/WR LED 及相关标志位。
- 格式化进行中, 运动控制模式变更, format不能正常执行。(仅在STOP 状态下支持)
- 格式化进行中, 不能变为 RUN模式

- 3) 之后, 激活格式化设定窗口, 设定窗如下。
容量, 文件系统, 单位大小在 SD 内存连接时, 读取的数据为 Default 值。
或者, 格式化格式仅支持快速format。volume label也是固定的英文'LSIS'。
如下设定页面中点击 [开始], 进行format。进行与否通过状态工具可以确认。

(3) 格式化完成及err代码

1) 状态信息

F 领域地址		标志位名	说明文
%KW523		_SD_FmtInfo	SD 内存 format 信息
bit	%KX8368	_SD_FmtRun	SD 内存格式化进行中
	%KX8369	_SD_FmtDone	SD 内存格式化正常完成
	%KX8370	_SD_FmtNg	SD 内存格式化失败
%KW524		_SD_FmtEcode	SD 内存格式化 errcode

2) err代码

errcode	Err 名称	err 说明文
0x0001	SD 脱离	SD 内存格式化中 SD 卡强制拔出
0x0002	文件系统损伤	SD 内存 format 中文件系统损伤

3) 完成语句

格式化正常完成时_SD_FmtDone(%KX8369) bit为 On。

Format失败时, err发生, 相应代码出现

11.7.3 诊断功能

提供数据记录中连接的 SD 内存进行诊断的功能。

下列项目中，不能使用违反的SD 内存。因此连接时，数据记录功能不能执行。

(1) FAT32 文件系统诊断

☞ 数据记录中保存文件，按FAT32 格式来format。

如果，有这种情况，文件不保存。

注意

由于可能会发生突然电源 OFF导致的文件系统及文件丢失，或者异常数据保存，想要停止数据记录功能时，必须数据记录允许标志位为OFF或者按SD CMD 按键2秒以上，数据保存正常结束。

11.8 CSV文件构成

11.8.1 文件保存格式

CSV 文件类型如下格式

名称	F	I	L	E	0	0	0	0	0	.CSV
说明	文件名称				组编号		文件编号			扩展
范围	固定值				0~15		000 ~ 255			固定值

文件名称首字母 4 字符固定为‘FILE’，之后5~6号位是设定的组编号,之后7~9号为文件编号。例如，11号组中的8号文件生成，则在‘组11’文件夹中生成‘FILE11008. CSV’。

11.8.2 文件类型和保存顺序

设定一个参数的组，数据记录功能执行时，文件顺序从 0号文件开始，但是多数组时，文件生成顺序在0号组文件生成后，15号组为止依序生成。如果参数设定时[组设定] 项目设定为[不使用]，文件不保存，进入下个组文件生成。

组	0号组	1号组	2号组	3号组	4号组	5号组	...	15号组
文件名称 及 生成顺序	FILE00000	FILE01000	FILE02000	FILE03000	FILE04000	FILE05000		FILE15000
	FILE00255	FILE01255	FILE02255	FILE03255	FILE04255	FILE05255		FILE15255

提示

运动控制开始收集的数据值按参数中设定的周期来保存，但是SD内存保存是在每次周期的主任务处理完成时，将进行分配来实现的，最初顺序是先从 0号组开始，但是根据文件保存可能会流动性变更。

11.9 Micro SD内存卡

11.9.1 Micro SD内存规格

要使用数据记录功能请满足以下规格的 micro SD内存

项目	内容
内存容量	可安装的最大容量 : 32GB(SD, SDHC 支持) (8GB 以上内存中可以使用到 8GB 为止)
文件系统	FAT32
电压范围	2.7 ~ 3.6V
运行温度范围	-25°C ~ 85°C
耐静电量	IEC61000-4-2 满足
插拔次数	最大 10,000 次
耗电	最大 100mA 以下(读取, 写入时)
读取写入次数	最大 100,000 次(SLC 基准)
尺寸	15mm * 11mm * 1mm
推荐产品	SanDisk, Transcend

提示

运动控制数据记录功能可以使用满足以上规格的所有 SD 内存。推荐产品SanDisk, Transcend 内存使用时, 可以发挥出最优化的性能。没有特别原因, 请使用推荐的产品

11.9.2 使用时的注意事项

使用SD 内存卡，数据记录功能在使用时请注意以下内容

(1) SD内存写入中电源断开

- 1) 运动控制收集的数据在SD内存写入时电源断开或者运动控制复位时，内存卡的文件系统可能会损伤。运动控制在通电时会确认SD内存的文件系统，将损坏的文件转换为可使用的状态，但是根据损坏程度也有可能无法修复，所以在断开运动控制的电源时，一定要确认SD内存中写入是否运行后再断电。
- 2) 运动控制收集的数据在 SD内存写入时电源断开或者运动控制复位时，运动控制内部的缓存内存中保存的数据全部消失。因此电源断开之前收集的数据可能在SD内存中未正常保存，运动控制的电源断开时，必须确认SD内存不是写入运行状态后再电源断开。

(2) SD内存写入中止时所需时间

数据保存中状态下，使用K领域标志位，数据记录允许标志位为Off 时，相应标志位指令接受之前，所有收集的数据中SD内存保存后，数据记录运行中断。因此，在实际数据记录功能停止以前，需要一些时间。数据记录中断为止，所需的时间会根据数据收集量和用户程序的执行状态而有所不同。

数据记录中止途中，DLxx_Stoping(xx是组编号)标志位为ON，中止后_DLxx_Finish标志位为On。中止进行期间SD内存保存剩余数据的大小在DLxx_Waiting数据标志位中显示。

(3) SD内存读取/写入时内存卡拔

- 1) 运动控制收集的数据在SD内存中写入或读取时，内存卡强制从运动控制拔出的话，内存卡的文件系统可能会损坏。因此必须使用指令标志位，使数据记录功能不激活，按SD CMD 按钮2秒以上，SD电源为OFF，STATE LED 灯灭后，再拔出 SD内存。
- 2) 数据记录运行中发生电源OFF或者复位时，数据可能未正常保存。或者，文件系统损坏，SD 内存和文件可能会无法识别。

(4) 使用盖子防止SD 内存误插拔

运动控制中SD 内存连接时，方向必须正确。

SD 内存插拔时，SD 内存按一下再拔出内存。

或者，为了防止振动而导致 SD 内存脱出，请使用盖子。

11.9.3 Micro SD 内存使用容量

运动控制最大可安装32GB内存，但是8GB的大概只能用80%。安装8GB以上的SD卡时，不是全体容量的80%而是可使用8GB的80%相应容量，约6.4GB以上使用的情况下，不能再用更多。这是防止数据保存时，SD内存接近时间过度的增加。。

注意

根据SD内存状态，会影响主任务执行时间和保存性能。使用SD内存前，格式化执行后再使用，长时间使用SD内存时，周期性执行格式化后才能维持性能

11.10 标志位一览

11.10.1 共通标志位

地址	数据类型	变量	功能	说明
%KX8800	BOOL	_DL_Rdy	数据记录准备状态	出现数据记录可能与否的标志位
%KX8192	BOOL	_DL_AutoLogStop	自动记录停止	自动记录停止指令输入标志位
%KX8801	BOOL	_DL_Err	数据记录错误	数据记录错误状态显示
%KX8256	BOOL	_SD_Attach	SD 内存安装状态	SD 内存的安装状态显示
%KX8257	BOOL	_SD_Rdy	SD 内存准备	出现 SD 内存使用可能与否的标志位
%KX8258	BOOL	_SD_Err	SD 内存错误	SD 内存错误状态显示
%KX8259	BOOL	_SD_Init	SD 识别中	SD 内存识别中状态显示
%KX8260	BOOL	_SD_Closing	SD 内存结束中	SD 内存结束中状态显示
%KX8261	BOOL	_SD_FATErr	文件系统错误	SD 内存文件系统错误状态显示
%KX8262	BOOL	_SD_AutoLogAct	自动记录中	SD 内存自动记录中状态显示
%KX8263	BOOL	_SD_Busy	SD 内存使用中	显示 SD 内存使用中
%KX8264	BOOL	_SD_SpaceWarn	SD 内存容量不足	SD 内存剩余容量不足显示.
%KX8265	BOOL	_SD_Detach	SD 内存脱离	SD 内存脱离显示.
%KD259	UDINT	_SD_VolTot	SD 内存容量	连接的 SD 内存容量 (GB) (8GB 以上显示为 8GB)
%KD260	UDINT	_SD_VolAvail	SD 内存可使用容量	SD 内存的可使用容量 (KB)
%KW522	WORD	_SD_Ecode	SD 错误代码	SD 内存使用中发生的错误编号出现
%KW523	WORD	_SD_FmtInfo	SD 内存格式化信息	SD 内存格式化信息显示
%KX8368	BOOL	_SD_FmtRun	SD 内存格式化中	SD 内存格式化中显示.
%KX8369	BOOL	_SD_FmtDone	SD 内存格式化正常结束	SD 内存格式化正常结束显示.
%KX8370	BOOL	_SD_FmtErr	SD 内存格式化失败	SD 内存格式化中失败显示.
%KW524	WORD	_SD_FmtEcode	SD 内存格式化错误代码	SD 内存格式化中发生的错误的错误编号显示
%KW525	WORD	_SD_FmtProgress	SD 内存格式化结束程度	SD 内存格式化进行率显示. (0~100(%))
%KW526	WORD	_SD_AttachCnt	SD 内存安装次数	SD 内存安装次数显示

地址	数据类型	变量	功能	说明
%KW527	WORD	_SD_DetachCnt	SD 内存脱离次数	SD 内存脱离次数显示
%KX8640	BOOL	_SD_AddfuncAct	SD 附加功能运行中	SD 内存附加功能运行中显示.
%KX8641	BOOL	_SD_AddfuncErr	SD 附加功能错误	SD 内存附加功能错误状态显示
%KX8642	BOOL	_SD_AddfuncDone	SD 附加功能结束	SD 内存附加功能运行结束状态显示
%KX8643	BOOL	_SD_CmpResult	SD 比较结果	SD 内存比较运行结果显示
%KW541	WORD	_SD_AddfuncKind	SD 附加功能种类	执行中的 SD 内存附加功能种类显示
%KW542	WORD	_SD_AddfuncEcode	SD 附加功能错误代码	SD 内存附加功能执行中发生的错误编号显示

11.10.2 组别标志位

(1) 参数组0号标志位

地址	数据类型	变量	功能	说明
%KX8224	BOOL	_DL00_Enable	组 00 数据记录允许	数据记录设定标志位 0: 停止, 1: 保存设定
%KX8960	BOOL	_DL00_Rdy	组 00 准备状态	数据记录准备状态状态 0: 未准备, 1: 准备好
%KX8961	BOOL	_DL00_Act	组 00 记录状态	数据记录保存进行状态 0: 停止, 1: 保存中
%KX8962	BOOL	_DL00_Err	组 00 记录错误	数据记录错误状态 0: 无错误, 1: 错误发生
%KX8963	BOOL	_DL00_Stopping	组 00 停止中	数据记录停止状态 0: 停止, 1: 停止中
%KX8964	BOOL	_DL00_Finish	组 00 停止结束	数据记录停止结束状态 0: 未结束, 1: 停止结束
%KX8965	BOOL	_DL00_Trig	组 00 触发发生	触发条件发生状态 0: 停止, 1: 运行中
%KX8966	BOOL	_DL00_TrigDone	组 00 触发结束	触发条件结束状态 0: 未结束, 1: 结束
%KX8967	BOOL	_DL00_Evt	组 00 事件发生	事件条件发生状态 0: 停止, 1: 运行中
%KX8968	BOOL	_DL00_Ovf	组 00 缓存超出	缓存超出状态标志位 0: 正常, 1: 缓存超出
%KW561	WORD	_DL00_Ecode	组 00 错误代码	错误代码
%KW562	WORD	_DL00_FileIdx	组 00 文件编号	保存文件编号 范围 :0~255
%KW563	WORD	_DL00_FileRollcnt	组 00 覆盖写入次数	文件覆盖写入次数
%KD282	UDINT	_DL00_FileSize	组 00 文件大小	文件大小

第11章数据记录功能

地址	数据类型	变量	功能	说明
%KD283	UDINT	_DL00_DataRow	组 00 数据行编号	数据行编号
%KD284	UDINT	_DL00_RemainBuf	组 00 剩余缓存容量	剩余缓存容量
%KD285	UDINT	_DL00_Waiting 数据	组 00 剩余数据	剩余数据
%KW572	WORD	_DL00_OvfCnt	组 00 缓存超出次数	缓存超出次数
%KW573	WORD	_DL00_TrigCnt	组 00 触发次数	触发条件发生次数
%KW574	WORD	_DL00_TrigOvrp	组 00 触发冲突次数	触发冲突次数
%KW575	WORD	_DL00_EvtCnt	组 00 事件次数	事件条件发生次数

(2) 参数组1 ~ 15号标志位

地址	size	变量	功能	说明
%KW580	20Word	-	-	组 1 的参数(和组 0 相同构成)
%KW600	20Word	-	-	组 2 的参数(和组 0 相同构成)
%KW620	20Word	-	-	组 3 的参数(和组 0 相同构成)
%KW640	20Word	-	-	组 4 的参数(和组 0 相同构成)
%KW660	20Word	-	-	组 5 的参数(和组 0 相同构成)
%KW680	20Word	-	-	组 6 的参数(和组 0 相同构成)
%KW700	20Word	-	-	组 7 的参数(和组 0 相同构成)
%KW720	20Word	-	-	组 8 的参数(和组 0 相同构成)
%KW740	20Word	-	-	组 9 的参数(和组 0 相同构成)
%KW760	20Word	-	-	组 10 的参数(和组 0 相同构成)
%KW780	20Word	-	-	组 11 的参数(和组 0 相同构成)
%KW800	20Word	-	-	组 12 的参数(和组 0 相同构成)
%KW820	20Word	-	-	组 13 的参数(和组 0 相同构成)
%KW840	20Word	-	-	组 14 的参数(和组 0 相同构成)
%KW860	20Word	-	-	组 15 的参数(和组 0 相同构成)

* _DLxx_Enable(组别数据记录使能标志位)按%KW514 的各位设定

11.10.3 错误代码及对应方案

数据记录功能相关错误代码如下.

分类	错误代码	错误	原因及对策	备注
全体错误代码	0x0000	无错误	-	
	0x0001	SD 内存卡识别错误	SD 内存卡损坏或者安装未通过 FAT32 格式化的 SD 内存卡下发生 通过 FAT32 格式化后再次安装, 若还无法识别请替换 SD 内存卡	
	0x0002	位置信息错误	位置信息无法读取 通过 FAT32 格式化后再次安装, 还无法识别请替换 SD 内存卡	
	0x0003	文件系统错误	没有通过 FAT32 格式化的 SD 内存卡。 通过 FAT32 格式化后请再次安装。	
	0x0004	支持的容量	请连接 2GB ~ 32GB 相应的 SD 内存卡	
	0x0005	SD 内存卡容量检查失败	SD 内存卡容量检查失败不能使用的状态。更换 SD 内存卡或者格式化后再次连接使用。	
	0x0006	SD 内存卡空间不足	SD 内存卡可使用容量比最大容量的 20%小时。(8GB 以上的 SD 内存卡, 约 8GB 的 80%, 6.4GB 以上可使用) 更换为其他 SD 内存卡或者格式化后确认容量, 再次连接使用。	
	0x0007	文件夹生成失败	SD 内存卡中无法生成数据记录组文件夹。确认 SD 内存卡是否异常, 更换或者格式化后再次连接使用	
组别错误代码	0x0100	组 xx 号文件夹生成错误	通过 FAT32 格式化后 SD 内存槽连接	
	0x0200	组 xx 号文件 off 错误		
	0x0400	组 xx 号文件写入错误		

11.11 数据记录性能

11.11.1 数据记录保存性能

对数据记录功能的保存性能进行说明。相应保存性能不是绝对值，各系统构成，设定实际测定的性能显示。实际应用到系统时，保存性能随着和数据收集方式，收集数据个数，收集数据的形态，SD 内存的种类及容量，还有SD 内存内的文件个数相同的设定环境来变动

11.11.2 主任务周期的保存性能

是随着主任务周期和设定数据保存个数所保存性能的测定结果。这个测定值作为对象值，可能和作成程序设定参数，应用SD内存，可能会发生差异。使用数据记录功能时，请作为参考资料使用。

(1) 设定条件

数据保存性能测定条件如下。

项目		内容	备注
主任务周期		1ms, 2ms, 4ms	
缓存大小		128KByte	
数据记录 设定	采样方法	每个主任务周期保存	
	数据	M领域, 类型:字	
	CSV输出	时间, 索引信息包括	
	文件保存	16MByte	
SD 存储卡		超出16G	

(2) 系统构成

为了测定性能系统利用运动控制的内置功能构成。

(3) 测试结果

根据主任务周期，保存性能参考下表

1) WORD 类型

	设备个数				
	4 WORD (4个 * 1组)	8 WORD (8个 * 1组)	16 WORD (12个 * 1组)	32 WORD (32个 * 1组)	64 WORD (64个 * 1组)
1ms	正常	正常	缓存超出发生	缓存超出发生	缓存超出发生
2ms	正常	正常	正常	正常	缓存超出发生

4ms	正常	正常	正常	正常	正常
-----	----	----	----	----	----

2) LWORD类型

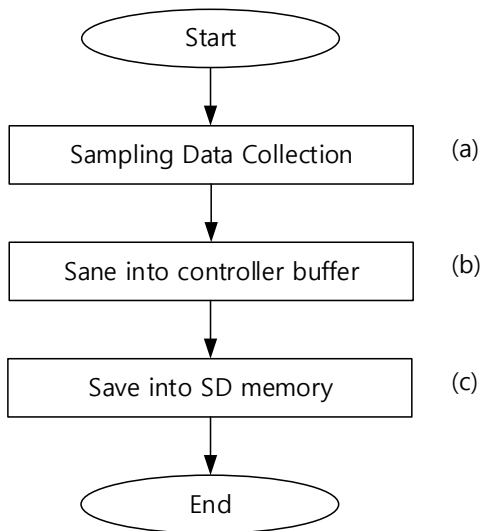
	设备个数				
	4 LWORD (4个 * 1组)	8 LWORD (8个 * 1组)	16 LWORD (16个 * 1组)	32 LWORD (32个 * 1组)	64 WORD (64个 * 1组)
1ms	正常	缓存超出发生	缓存超出发生	缓存超出发生	缓存超出发生
2ms	正常	正常	缓存超出发生	缓存超出发生	缓存超出发生
4ms	正常	正常	正常	正常	缓存超出发生

11.11.3 保存处理时间确认

数据记录功能对于所有设定，不能保证数据全部保存，会按照数据记录条件发生的时点可执行的最大来运行的功能。即，数据记录的处理时间根据参数设定，采样数据的个数，主任务周期，运动控制的其他功能运行状态可能会有变动，所有有可能不按照设定的收集条件运行。因此，建议使用数据记录功能前，增加系统的处理时间后再数据记录功能。

(1) 数据记录处理时间确认步骤

下图是运动控制在执行数据记录功能，SD内存中保存运行流程，各个内容如下。



阶段	运行	备注
(1) 数据收集	数据记录开始时，收集数据，执行内部的缓存中保存的运行。数据收集符合参数设定的条件(每个主任务保存,指定周期等)，但是可能会依据数据的个数，功能使用，无法按照设定的条件收集。	
(2) 数据设计	收集的数据以可以在CSV文件中保存的形态来设计处理。设计后SD卡中保存的缓存下保存。数据保存缓存较满时，设计的数据要在缓存中留有空间。	

第11章数据记录功能

(3)	数据保存	设定的数据在SD卡中保存为文件后运行。数据保存速度不如数据收集速度时，可能是超出内部缓存，数据可能丢失。	
-----	------	--	--

(2) 数据保存处理时间的确认方法

确认收集的数据是否正常在SD卡中保存，要确认的内容如下。

确认事项	内容及解决方法		备注
缓存超出 次数标志位	内容	确认K领域的缓存超出发生次数是否为0。 如果不是 0，比起数据收集时间，数据收集太快，代表可能会有数据的保存丢失。 保存文件内插入 ‘C’ 文字列	
	解决方法	主任务周期，主任务周期延长，指定周期采样，采样周期增加， 每次采样，收集的数据的量减少。仅保存必需的数据文件(触发保存功能使用)	

第12章SD 附加功能

12.1 概要

使用运动控制 SD 卡内置的附加功能。本章节中对 SD 附加功能的规格及使用方法等进行了说明

12.1.1 特征

使用运动控制SD附加功能，不是MP500的直接接触而是使用SD卡，可进行PLC 更新,备份,对比,boot运行。该功能会使PLC的 SD CMD按钮操作动作。

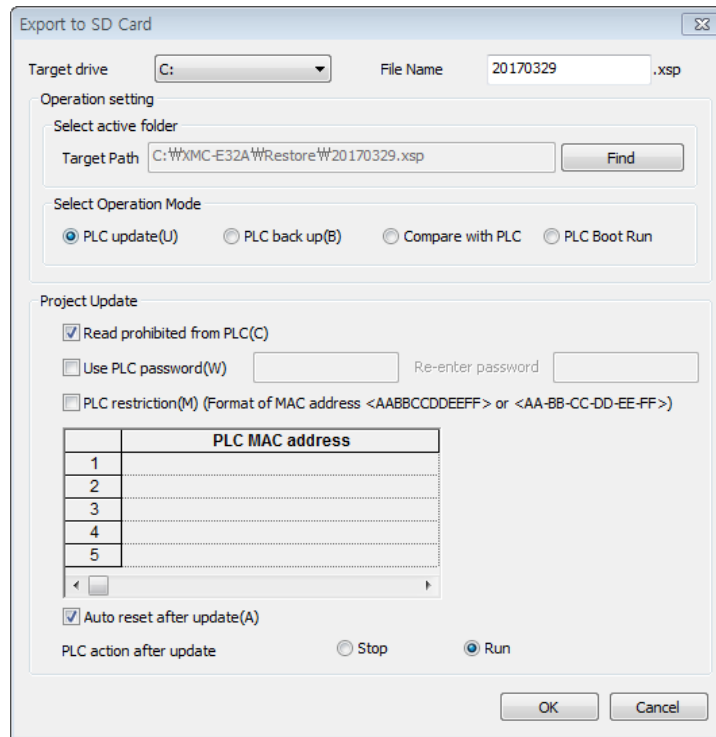
- (1) 使用SD 附加功能的SD卡设定功能提供
 - 使用MP500的SD卡设定可以
- (2)使用SD卡的PLC 更新
 - 可以使用密码设定的 PLC 程序泄露防止
 - 使用PLC的MAC 地址的更新执行限制
 - 更新执行后PLC自动reset及动作全部设定可以
- (3) 使用SD卡的 PLC 备份
 - PLC的程序没有MP500， SD卡中可以保存
 - PLC履历也可以一起备份
- (4) 和使用SD卡的PLC对比
 - 可以PLC 参数,动作数据, PLC 程序, NC卡,cam数据对比
 - 对比结果在SD卡中保存，可以确认
- (5) 使用SD卡的PLC boot运行
 - 使用密码设定，可以保护 PLC 程序
 - 使用PLC的MAC地址，限制boot运行

12.1.2 SD卡导出功能

[MP500] - [工程] - [SD卡设定] - [SD卡导出]执行的话，可执行SD卡的设定窗口。(SD卡导出功能在MP500不在线的情况下可以使用)

(1) PLC 更新

PLC 更新功能是SD卡中保存的程序可以在PLC中更新的功能。

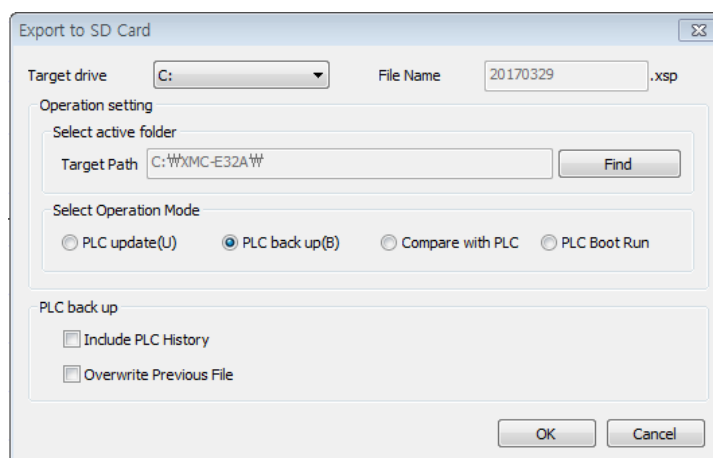


全部PLC更新时，各项目的说明如下

区分	内容
目标驱动器	工程数据保存媒体选择
动作文件夹选择	工程数据保存位置(文件夹)
动作全部选择	SD卡插入时PLC 动作设定 PLC更新: 利用SD卡中保存的数据, PLC更新
PLC禁止读取	利用SD, PLC工程更新时, PLC禁止读取
PLC密码使用	PLC 密码设定, 利用SD确认 PLC 工程是否可以更新
PLC使用限制	PLC 工程更新可以指定PLC
更新完成后自动复位	PLC 更新执行后 PLC 复位执行与否
更新完成后PLC 状态	PLC 更新执行后 PLC 动作全部设定

(2) PLC备份

PLC中保存的程序在SD卡中备份的功能

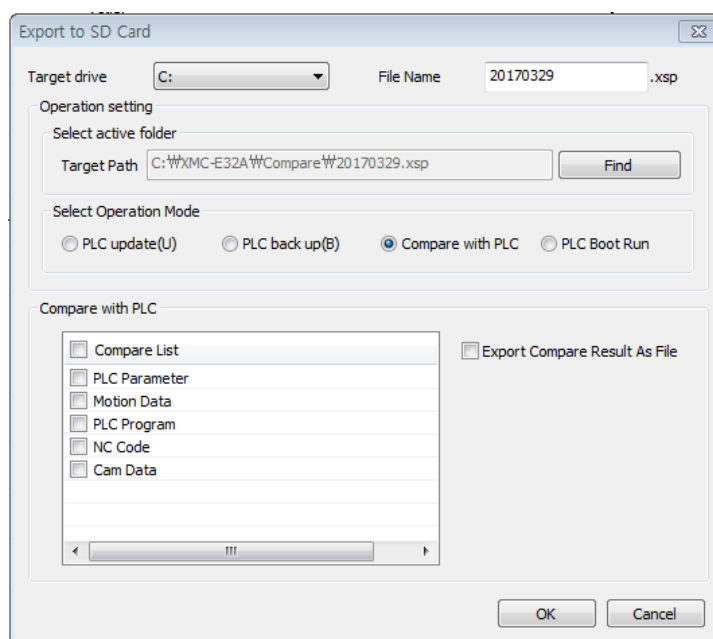


PLC全部备份时，各目别说明如下

区分	内容
目标驱动器	工程数据保存媒体选择
动作文件夹选择	工程数据保存位置(文件夹)
动作全部选择	SD卡插入时PLC 动作设定 PLC 备份: PLC 工程在SD卡中保存
PLC 履历包含	PLC 备份时，确认PLC中保存的履历是否也一起备份
覆盖现有文件	SD卡中有曾经备份的文件时，确认是否覆盖

(3) 和PLC对比

PLC中保存的工程和SD卡中保存的工程对比的功能。

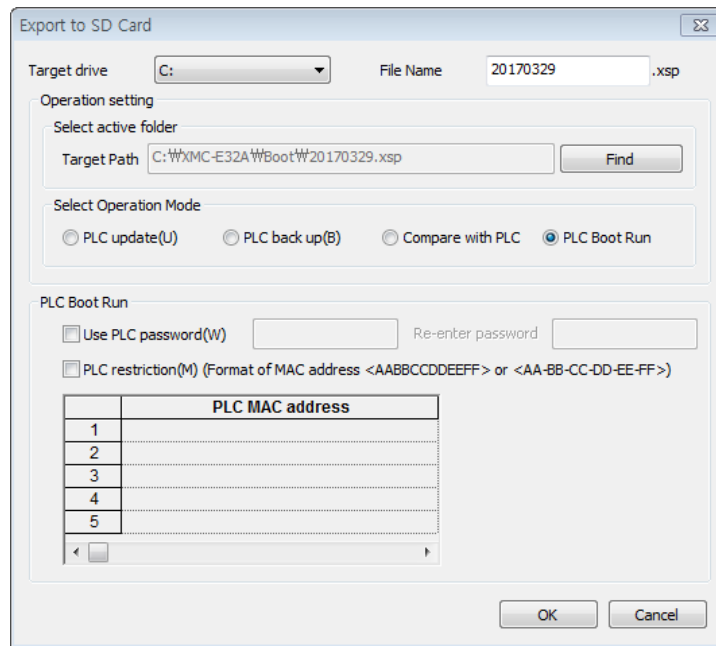


和PLC全部对比时，各目别说明如下

区分	内容
目标驱动器	工程数据保存媒体选择
动作文件夹选择	工程数据保存位置(文件夹)
动作全部选择	SD卡插入时PLC 动作设定 和PLC 对比: PLC工程和SD卡中保存的工程对比
对比项目	仅可以对比需要的项目
对比结果文件保存	确认对比结果是否以csv形态文件保存

(4) PLCboot运行

利用SD中保存的工程，通过可使PLC驱动的功能，SD 拔出后PLC电源Off, On时，通过boot运行前驱动的程序来驱动

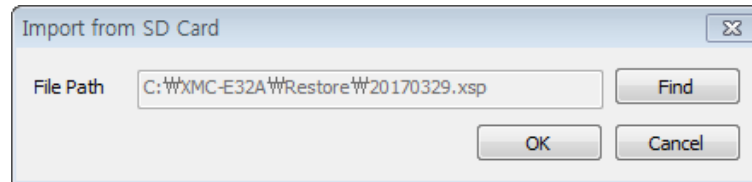


PLCboot全部运行时，各项目别说明如下

区分	内容
目标驱动器	工程数据保存媒体选择
动作文件夹选择	工程数据保存位置(文件夹)
动作全部选择	SD卡插入时 PLC 动作设定 PLC boot运行: 利用SD卡中保存的数据, PLC驱动
PLC 密码使用	利用SD, 确认PLC boot运行是否可以更新
PLC 使用限制	可PLC boot运行的PLC 指定

12.1.3 SD卡输入功能

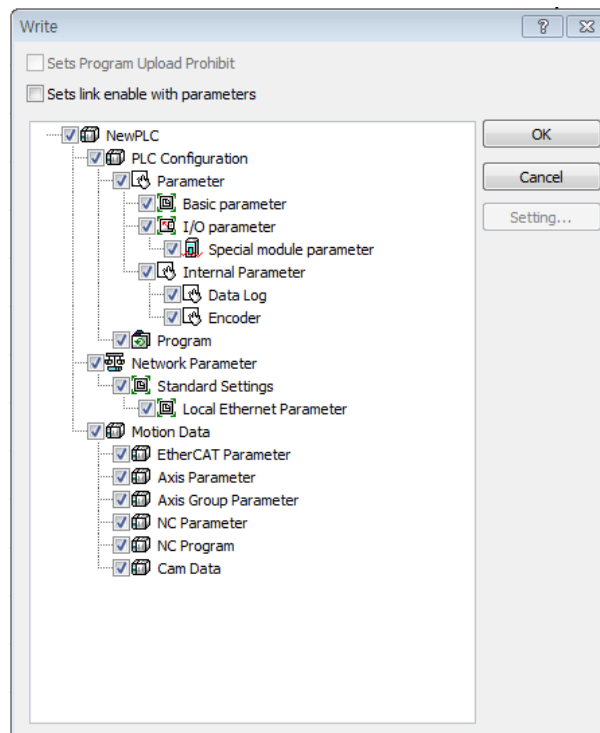
[MP500] - [工程] - [SD卡设定] - [SD 输入]执行的话，文件读取窗口出现。
在该路径下确认时，MP500中SD中保存的工程打开。



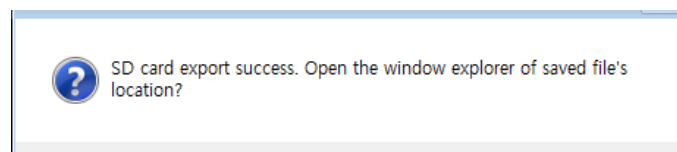
12.1.4 PLC更新功能

PLC 更新功能仅在PLC全部STOP时，可以动作。

SD卡导出中选择PLC 更新后，点击确认，出现如下写入窗口。



正常SD卡导出动作完成的话，会出现成功完成窗口，显示保存的驱动器。



确认保存的驱动器时，机种文件夹下生成附加功能文件夹，配置，恢复文件夹内生成文件。

该SD卡安装在PLC的SD卡插口上，按照配置中设定的值，在SD 附加功能结束标志位(%KW541)中显示。

%KW541

SD 附加功能全部

0	附加功能X
1	PLC 备份功能
2	PLC 更新功能
3	PLC 对比功能
4	boot运行功能

SD CMD按钮按1秒以上, 2秒以下时, SD RD/WR LED和 SD附加功能动作中标志位(%KX8640)为 On, PLC更新动作执行。正常更新完成, SD RD/WR LED和 SD附加功能动作中标志位(%KX8640)为Off, SD附加功能完成标志位(%KX8642)为On。如果动作中错误发生时, SD附加功能错误标志位(%KX8641)为On, SD附加功能错误代码(%KW542)中显示错误值

(1)禁止从PLC处读取时, 更新即使正常, 也禁止通过 MP500从 PLC读取, PLC 密码选项设定后, PLC中密码没有设定时, 和工程更新执行一起, 密码在PLC中保存。或者利用SD, 工程更新执行前, PLC中密码未设定的话, 通过选项设定的密码和 PLC的密码一致的情况下才能执行更新。

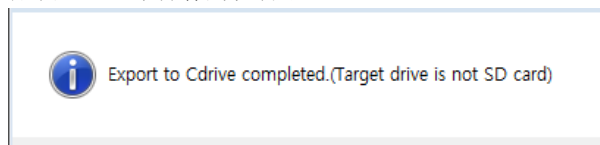
(2)PLC 使用限制选项设定后, 通过选项保存的 MAC 地址和 PLC的 MAC 地址一致才能执行PLC 更新。

(3)更新后完成后自动复位设定时, 更新在完成后PLC自动复位, 通过PLC状态选项指定值变更全部PLC 动作。

12.1.5 PLC备份功能

此功能是通过在PLC中保存的工程在SD卡中备份的功能，以在相应产品MAC地址文件夹内的备份文件夹中PLC的工程备份文件保存。PLC备份功能和PLC无关，可以动作。

正常SD卡来导出动作完成后，生成完成窗口，显示保存的驱动器。



确认保存的驱动器时，机种文件夹下生成附加功能文件夹，配置文件夹内生成文件。

该SD卡安装在PLCSD卡上的话，根据配置中设定的值，SD附加功能结束标志位(%KW541)中显示。

%KW541	SD 附加功能全部
0	附加功能X
1	PLC备份功能
2	PLC更新功能
3	PLC对比功能
4	boot运行功能

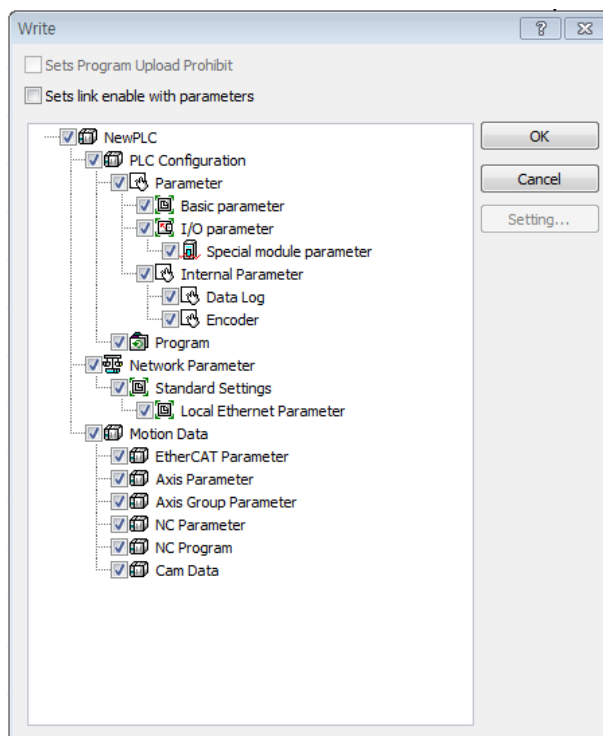
SD CMD按钮按1秒以上，2秒以下时，SD RD/WR LED和SD 附加功能动作中标志位(%KX8640)为 On，PLC更新动作执行。正常备份完成的话，SD RD/WR LED和SD 附加功能动作中标志位(%KX8640)为Off，SD附加功能完成标志位(%KX8642)为On。如果动作中发生错误，SD附加功能错误标志位(%KX8641)为On，SD附加功能错误代码(%KW542)中显示错误值。

确认SD卡拔出后，该产品路径备份文件夹下工程保存后，MP500中SD输入时，打开保存的工程。

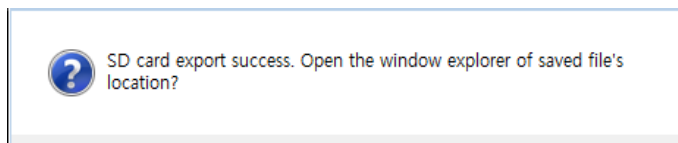
12.1.6 PLC 对比功能

本功能是通过 PLC中保存的工程和SD卡中保存的程序对比功能，可以将对比结果以标志位或者.csv文件来确认。

SD卡来导出中PLC对比选择后，点击确认，生成如下写入窗口。



正常SD卡来导出动作完成后，生成成功完成窗口，显示保存的驱动器。



确认保存的驱动器时，机种文件夹下生成附加功能文件夹，配置，比较文件夹下生成文件。

该SD卡安装在PLC的SD卡插口中，根据配置中设定的值，SD 附加功能结束标志位(%KW541)中显示。

%KW541	SD 附加功能全部
0	附加功能 X
1	PLC备份功能
2	PLC更新功能
3	和PLC对比功能
4	boot运行功能

SD CMD 按钮按1秒以上，2秒以下，SD R/W LED和SD附加功能动作中标志位(%KX8640)为On，PLC 更新动作执行。正常对比处理完成后，SD R/W LED和 SD附加功能动作中标志位(%KX8640)为Off，SD附加功能完成标志位(%KX8642)为On。

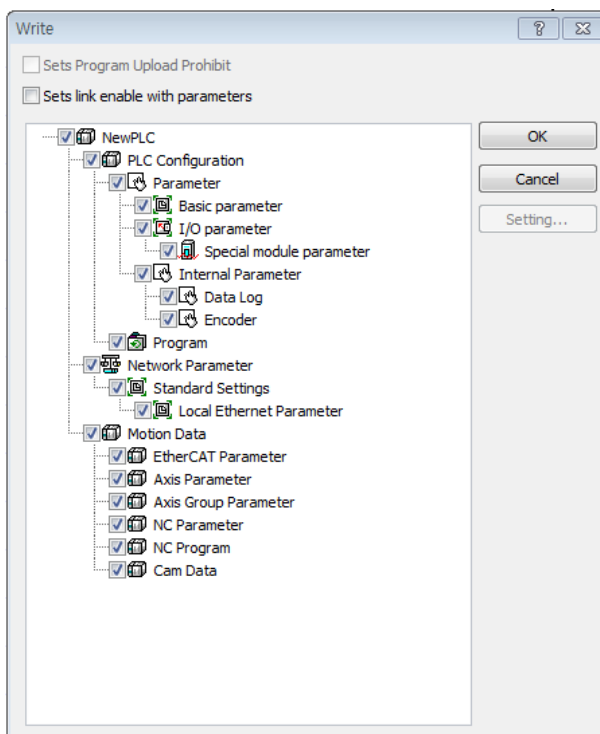
如果动作中错误发生时，SD附加功能错误标志位(%KX8641)为On，SD附加功能错误代码(%KW542)中显示错误值。

- (1) 对比中有其他部分时，SD对比结果标志位(%KX8643)为Off，相同情况下SD对比标志位为 On 。
- (2) SD卡来导出动作时，‘对比结果以文件形式保存’项目中查看时，比较文件夹中生成结果值文件(CmpResult.csv)，和对比结果保存。

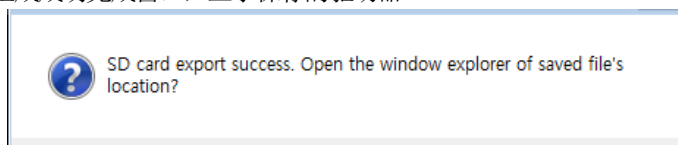
12.1.7 PLCboot运行功能

此功能并非是通过PLC中保存的工程而是SD中保存的程序来驱动PLC的。目前驱动过的程序在PLC中保存，SD卡拔出后，PLC电源Off, On时，通过现有程序再次驱动。

SD卡来导出中选择PLCboot运行后，点击确认，生成如下写入窗口。



正常SD卡来导出动作完成后，生成成功完成窗口，显示保存的驱动器。



确认保存的驱动器时，机种文件夹下生成附加功能文件夹，配置，Boot文件夹下生成文件。

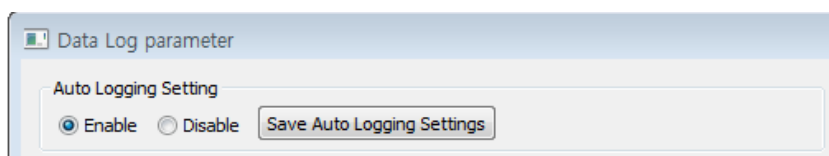
boot运行要在PLC电源Off状态下操作。PLC电源Off状态下SD卡安装后按SD CMD 按钮，PLC电源为On。正常boot运行全部完成的话，SD RW LED和SD附加功能动作中标志位(%KX8640)为Off，SD附加功能完成标志位(%KX8642)为On。

如果动作中错误发生时，SD附加功能错误标志位(%KX8641)为On，SD附加功能错误代码(%KW542)中显示错误值。

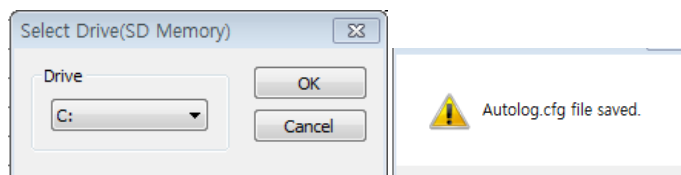
12.1.8 自动记录功能

此功能不是 PLC中保存的数据记录参数而是SD卡中保存的设定文件数据记录执行的功能。

在数据记录参数窗口的左上角有自动记录设置的项目。



设定数据记录相关参数后，自动记录设定保存的话，设定文件可在需要的路径下保存。



自动记录是自动记录参数在保存SD卡插入后动作。

- (1) PLC中保存的参数在禁止自动记录的情况下，Autolog.cfg的设定未反映，通过现有程序设定来动作。
- (2) SD中保存的该参数在PLC中未保存，自动记录中的SD卡拔出后，插入其他SD卡，会再次根据原来的参数动作。
- (3) 自动记录中自动记录中止(_DL_AutoLogStop : %KX8192)指令标志位为On，目前进行中的自动记录中断，此后组别数据记录允许标志位为 On或者自动记录中止指令标志位为 Off，记录不再进行。

12.1.9 错误代码及对应方案

SD附加功能相关的错误如下。错误显示时，附加功能全部一起显示。

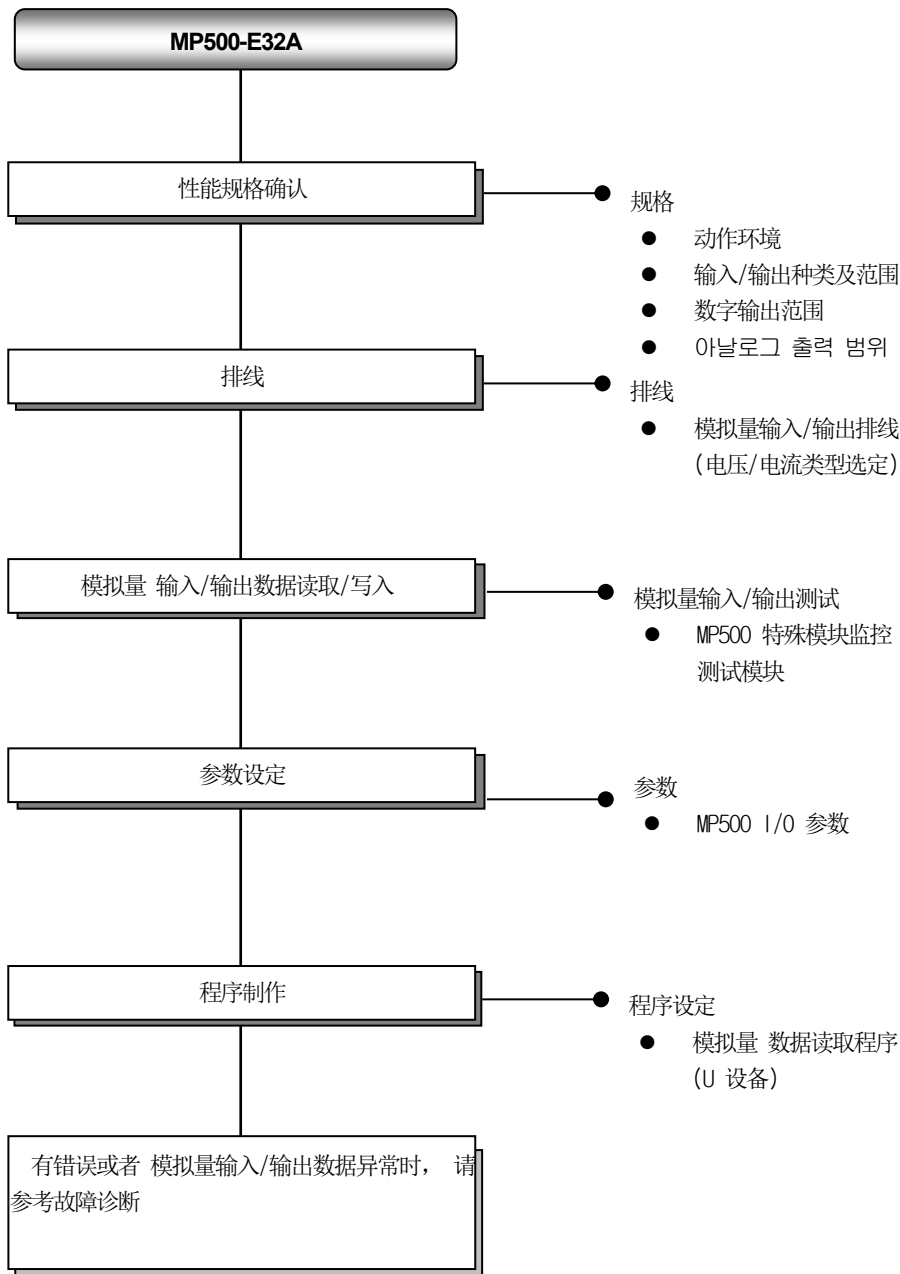
(例:PLC更新功能中没有文件密码时 (0x2005) ->2: 附加功能全部, 5: 错误动作)

分类	错误代码	错误名称	备注
全部错误代码	0xX001	文件错误(文件off失败, CRC错误)	
	0xX002	受损文件(头, 尾等损伤)	
	0xX003	不支持的文件版本	
	0xX004	机种不一致	
	0xX005	无文件中密码	
	0xX006	密码不一致	
	0xX007	MAC地址不一致	
	0xX008	文件重复错误	
	0xX009	IO构成不一致	
	0xX00A	无保存文件	
	0xX00B	PLC全部RUN状态	
	0xX00C	无SD卡	
	0xX00D	SD卡错误状态	
	0xX00E	SD卡电源断开中	
	0xX00F	SD卡电源 Off 状态	

第13章 内置模拟量功能

13.1 概要

内置模拟输入/输出前，要按如下顺序进行。



第 13 章 内置模拟量功能

内置模拟模块的性能规格说明

1) 输入性能规格

项目		性能规格		
通道数		2通道		
模拟 输入范围	种类	电压	电流	
	范围	DC 1~5V DC 0~5V DC 0~10V DC -10~10V (输入电阻 : 1MΩ 异常)	DC 4~20mA DC 0~20mA (输入电阻 : 250Ω)	
		输入范围按通道别在用户程序或者 I/O 参数中设定后通过外部配线对电压/电流 设定(电压输入时相应通道的V+和COM 端子进行使用, 电流输入时使用V+和I+端子连接 后 I+和COM端子。)		
	范围	无符号值	0~16,000	
		有符号值	-8,000~8,000	
		准确值	1,000~5,000(1~5V) 0~5,000(0~5V) 0~10,000(0~10V) -10,000~10,000(±10V)	4,000~20,000(4~20mA) 0~20,000(0~20mA)
		百分位值	0~10,000	
	最大分辨率		1/16,000	
			0.250mV(1~5V) 0.3125mV(0~5V) 0.625mV(0~10V) 1.250mV(±10V)	1.0μA(4~20mA) 1.25μA(0~20mA)
	精度		±0.2% 以下(周围 温度 25±5℃) ±0.3% 以下(周围 温度 0~55℃)	
最大变换速度		0.5ms/通道		
绝对最大输入		DC ±15V	DC ±30mA	
附加功能	滤波功能	数字滤波(4~64,000ms)		
	平均功能	时间平均(4~16,000ms)		
		计数平均(2~64,000次)		
		移动平均(2~100个)		
		加权平均(1~99%)		
	警告功能	断开检出(DC 1~5V, DC 4~20mA)		
	维持最后值	超出有效输入范围时, 变换值维持与否决定		
警报功能	超出有效输入范围时, 警报 上/下限标志位发生			
输入端子		6点端子台		

2) 输出性能规格

项目		性能规格
输出通道数		电压2通道
	范围	DC 1~5V DC 0~5V DC 0~10V DC -10~10V (负载电阻 : 1kΩ 以上)
		输出范围按照通道别在用户程序 或者 I/O 参数中设定
	范围	无符号值
		有符号值
		准确值
百分位值		
最大分辨率		1/16,000 0.250mV(1~5V) 0.3125mV(0~5V) 0.625mV(0~10V) 1.250mV(±10V)
精度		±0.2% 以下(周围 温度 常温 25±5℃) ±0.3% 以下(周围 温度 0 ~ 55℃)
最大变换速度		0.5ms/通道
附加功能		<ul style="list-style-type: none"> • 通道输出状态设置 - 在前一个值, 最小值, 平均值, 最大值之间选择一个 • 插值方式设置 - 线性插值, S-类型插值
输出端子		6点 端子台

3) 输入输出共通性能规格

项目	性能 规格
绝缘方式	输入输出端子和运动控制模块电源间绝缘(通道间不绝缘)
输入输出点数	固定式 : 64点

13.2 模拟部分的名称和作用



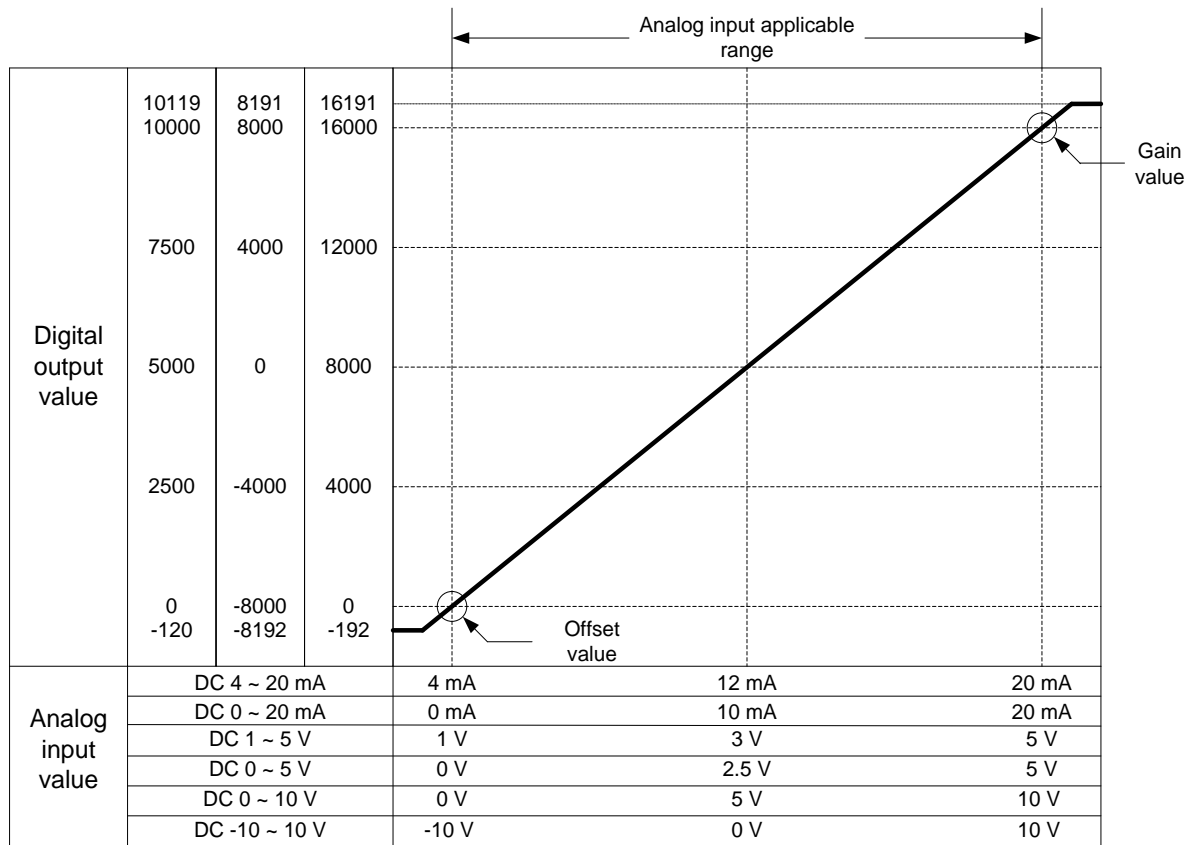
编号	名称	内容
①	A/D LED	<ul style="list-style-type: none"> ▶ 输入部的动作状态显示 灯亮：正常动作中 闪烁：错误发生(1S 周期闪烁) 灯灭：电源OFF 或者模块异常
②	D/A LED	<ul style="list-style-type: none"> ▶ 输出部的动作状态时 灯亮：正常动作中 闪烁：错误发生(1S周期 闪烁) 灯灭：电源 OFF 或者模块异常
③	输入连接器	▶ 外部设备和连接的模拟输入配线用连接
④	输出连接器	▶ 外部设备和连接的模拟输出配线用连接

13.3 I/O控制特性

电压/电流输入输出范围利用用户程序或者I/O参数按通道别设定。
 数字数据的输入输出形态如下

- 1) 无符号值
- 2) 有符号值
- 3) 精准值
- 4) 百分位值

13.3.1 输入特性



1) DC 4~20mA 输入范围

数字 输出范围	模拟输入电流(mA)						
	3.808	4	8	12	16	20	20.191
无符号值 (-192~16,191)	-192	0	4,000	8,000	12,000	16,000	16,191
有符号值 (-8,192~8,191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191
精准值 (3,808~20,191)	3,808	4,000	8,000	12,000	16,000	20,000	20,191
百分位值 (-120~10,119)	-120	0	2,500	5,000	7,500	10,000	10,119

第 13 章 内置模拟量功能

2) DC 0~20mA 输入范围

数字 输出范围	模拟输入电流(mA)						
	-0.24	0	5	10	15	20	20.239
无符号值 (-192~16,191)	-192	0	4,000	8,000	12,000	16,000	16,191
有符号值 (-8,192~8,191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191
精准值 (-240~20,239)	-240	0	5,000	10,000	15,000	20,000	20,239
百分位值 (-120~10,119)	-120	0	2,500	5,000	7,500	10,000	10,119

3) DC 1 ~ 5V 输入范围

数字 输出范围	模拟输入电压(V)						
	0.952	1	2	3	4	5	5.047
无符号值 (-192~16,191)	-192	0	4,000	8,000	12,000	16,000	16,191
有符号值 (-8,192~8,191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191
精准值 (952~5,047)	952	1,000	2,000	3,000	4,000	5,000	5,047
百分位值 (-120~10,119)	-120	0	2,500	5,000	7,500	10,000	10,119

4) DC 0 ~ 5V 输入范围

数字 输出范围	模拟输入电压(V)						
	-0.06	0	1.25	2.5	3.75	5	5.059
无符号值 (-192~16,191)	-192	0	4,000	8,000	12,000	16,000	16,191
有符号值 (-8,192~8,191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191
精准值 (-60~5,059)	-60	0	1,250	2,500	3,750	5,000	5,059
百分位值 (-120~10,119)	-120	0	2,500	5,000	7,500	10,000	10,119

5) DC 0 ~ 10V 输入范围

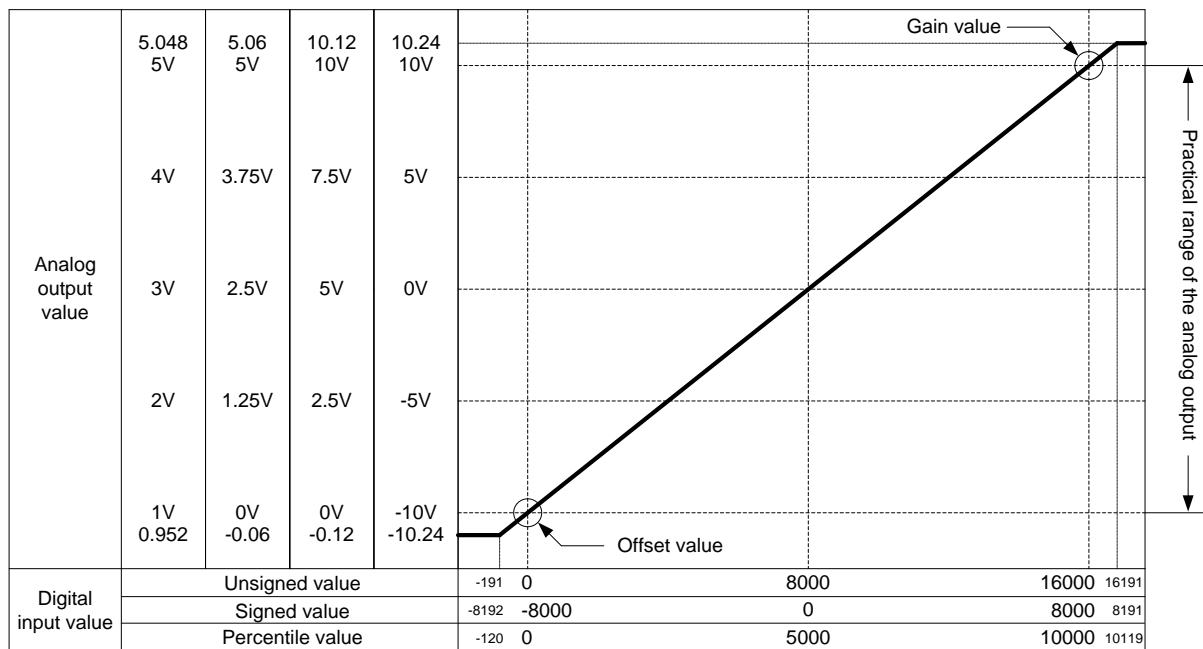
数字 输出范围	模拟输入电压(V)						
	-0.12	0	2.5	5	7.5	10	10.119
无符号值 (-192~16,191)	-192	0	4,000	8,000	12,000	16,000	16,191
有符号值 (-8,192~8,191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191
精准值 (-120~10,119)	-120	0	2,500	5,000	7,500	10,000	10,119
百分位值 (-120~10,119)	-120	0	2,500	5,000	7,500	10,000	10,119

6) DC -10 ~ 10V 输入范围

数字 输出范围	模拟输入电压(V)						
	-10.24	-10	-5	0	5	10	10.239
无符号值 (-192~16,191)	-192	0	4,000	8,000	12,000	16,000	16,191
有符号值 (-8,192~8,191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191
精准值 (-10,240~10,239)	-10,240	-10,000	-5,000	0	5,000	10,000	10,239
百分位值 (-120~10,119)	-120	0	2,500	5,000	7,500	10,000	10,119

13.3.2 输出特性

根据数字输入的范围，输出变换特性如下。



1) DC 1~5V 输出范围

数字 输入	模拟输出电压(V)						
	0.952	1	2	3	4	5	5.047
无符号值 (-192~16,191)	-192	0	4,000	8,000	12,000	16,000	16,191
有符号值 (-8,192~8,191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191
精准值 (952~5,047)	952	1,000	2,000	3,000	4,000	5,000	5,047
百分位值 (-120~10,119)	-120	0	2,500	5,000	7,500	10,000	10,119

第 13 章 内置模拟量功能

2) DC 0 ~ 5V 范围输出

数字 输入	模拟输出电压(V)						
	-0.06	0	1.25	2.5	3.75	5	5.059
无符号值 (-192~16,191)	-192	0	4,000	8,000	12,000	16,000	16,191
有符号值 (-8,192~8,191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191
精准值 (-60~5,059)	-60	0	1,250	2,500	3,750	5,000	5,059
百分位值 (-120~10,119)	-120	0	2,500	5,000	7,500	10,000	10,119

3) DC 0~10V 输出范围

数字 输入	模拟输出电压(V)						
	-0.12	0	2.5	5	7.5	10	10.119
无符号值 (-192~16,191)	-192	0	4,000	8,000	12,000	16,000	16,191
有符号值 (-8,192~8,191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191
精准值 (-120~10,119)	-120	0	2,500	5,000	7,500	10,000	10,119
百分位值 (-120~10,119)	-120	0	2,500	5,000	7,500	10,000	10,119

4) DC -10~10V 输出范围

数字 输入	模拟输出电压(V)						
	-10.24	-10	-5	0	5	10	10.239
无符号值 (-192~16,191)	-192	0	4,000	8,000	12,000	16,000	16,191
有符号值 (-8,192~8,191)	-8,192	-8,000	-4,000	0	4,000	8,000	8,191
精准值 (-10,240~10,239)	-10,240	-10,000	-5,000	0	5,000	10,000	10,239
百分位值 (-120~10,119)	-120	0	2,500	5,000	7,500	10,000	10,119

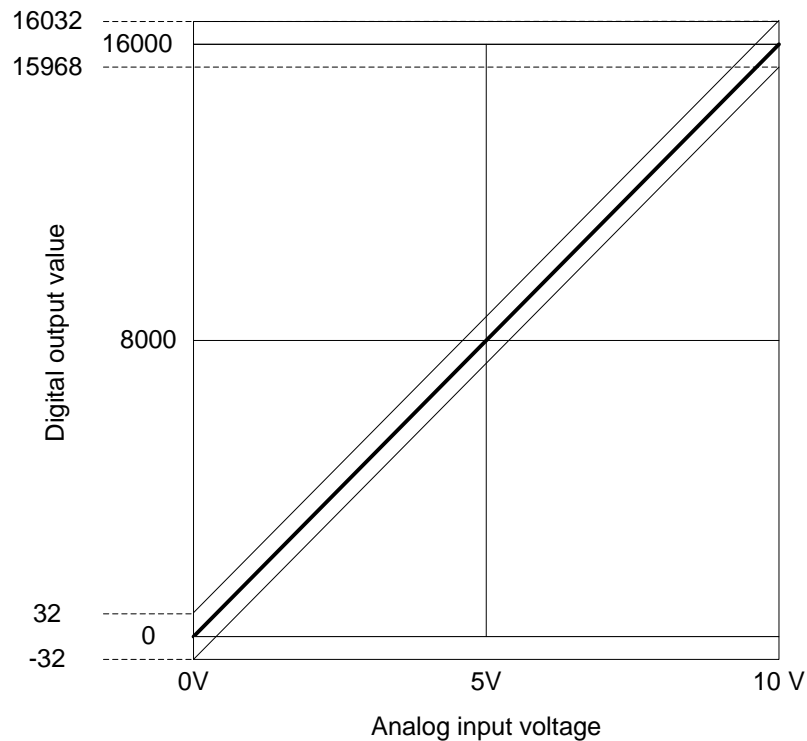
13.4 精度

13.4.1 输入精度

对于数字输出值的精度即使输入范围变更也不能改。

如下图模拟输入范围选择为0~10V，通过数字输出形态选择无符号值时，精度变动范围显示。

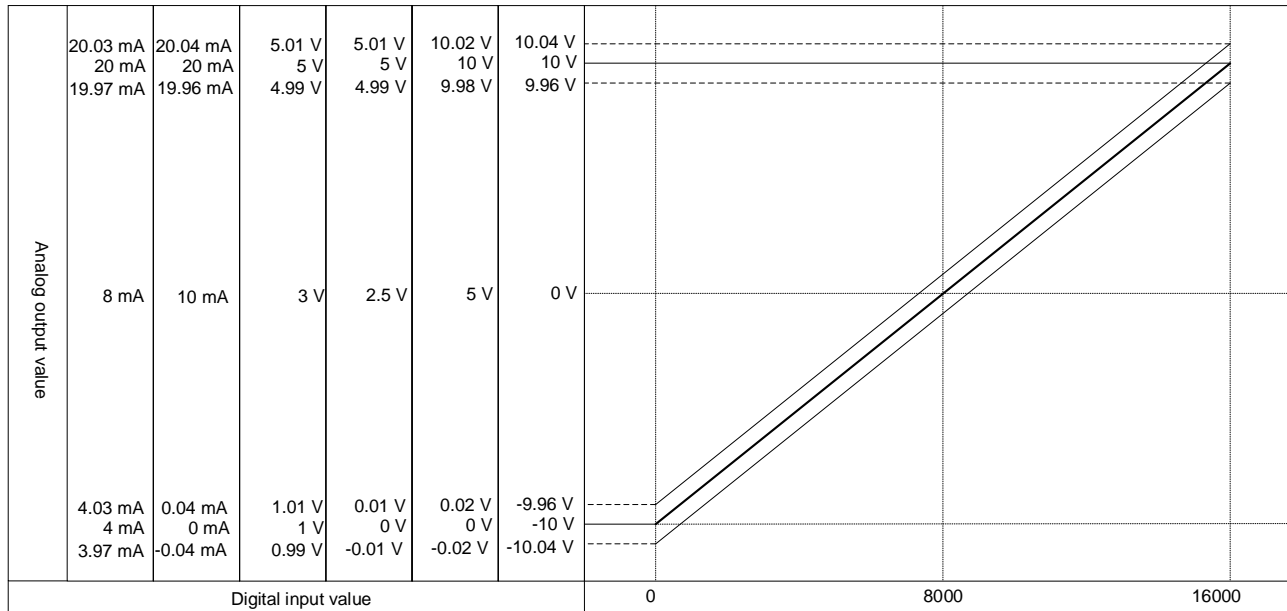
精度为±0.2%(常温, 25±5℃)。



- 1) 5V 输入使用时的精度= $16,000 \times 0.2\% = 32$
因此, 5V 输入时精度范围为 $(8,000-32) \sim (8,000+32) = 7,968 \sim 8,032$ 。
- 2) 10V 输入使用时的精度= $16,000 \times 0.2\% = 32$
因此, 10V 输入时精度范围为 $(16,000-32) \sim (16,000+32) = 15,968 \sim 16,032$ 。

13.4.2 输出精度

对于模拟输出的精度即使输入范围变更也无法修改。
 如下图选择数字输入范围无符号值时，精度变动范围显示精度为±0.2%(常温, 25±5℃)。



- 1) -10~10V 输出使用时的精度= $16,000 \times 0.2\% = 32$
 -10V 输出时精度范围为 $(-10V - 32 \times 1.25 \text{ mV}) \sim (-10V + 32 \times 1.25 \text{ mV}) = -10.04 \sim -9.96V$, 10V 输出时精度范围为 $(10V - 32 \times 1.25 \text{ mV}) \sim (10V + 32 \times 1.25 \text{ mV}) = 9.96 \sim 10.04V$ 。
- 2) 1~5V 输出使用时的精度= $16,000 \times 0.2\% = 32$
 1V 输出时精度范围为 $(1V - 32 \times 0.25 \text{ mV}) \sim (1V + 32 \times 0.25 \text{ mV}) = 0.992V \sim 1.008V$, 5V 输出时精度范围为 $(5V - 32 \times 0.25 \text{ mV}) \sim (5V + 32 \times 0.25 \text{ mV}) = 4.992V \sim 5.008V$ 。

13.5 内置模拟量功能

内置模拟模块的功能说明。

功能项目	内容
通道运行/停止设定	<ul style="list-style-type: none"> ● 指定要执行 A/D, D/A 变换的通道运行/停止 ● 不使用的通道设定为停止, 整体运行时间缩短。
输入输出电压/电流范围设定	<ul style="list-style-type: none"> ● 指定要使用的模拟输入输出范围。 ● 根据外部端子台配线选择电压·电流后, 选择参数范围。 ● 内置模拟模块提供电流输入 2 个范围(4~20mA, 0~20mA)和电压输入·输出 4 个范围(1~5V, 0~5V, 0~10V, -10~10V)。
输入输出数据类型设定	<ul style="list-style-type: none"> ● 数字输入输出形态指定 ● 该模块中提供 4 个输出数据类型。(无符号值, 有符号值, 精准值, 百分位值)
A/D 输入变换方式	<ul style="list-style-type: none"> ● 采样处理 <ul style="list-style-type: none"> - A/D 变换方式未指定时, 采样处理。 ● 滤波处理 <ul style="list-style-type: none"> - 过滤外部噪声导致输入值的剧烈变动。 ● 平均处理 <ul style="list-style-type: none"> - 根据时间, 计数基准, 平均的A/D变换值输出 ● 警告功能(输入断线检出功能) <ul style="list-style-type: none"> - 检测输入回路是否断线, 断线标志位显示。(输入信号范围: 4~20mA, 1~5V) ● 有效变换值维持功能 <ul style="list-style-type: none"> - 超出有效输入范围时, 设定变换值是否维持。 ● 警报功能 <ul style="list-style-type: none"> - 超出有效输入范围时, 警报上/下限标志位显示。
D/A 输出状态设定	<ul style="list-style-type: none"> ● 从 run 转换为 stop 时, 设定通道的输出状态 ● 提供4个输出状态。(在前一个值, 最小值, 中间值, 最大值之间选择)
插补方法设定	<ul style="list-style-type: none"> ● 设定直线插补, S 型插补方法

13.5.1 采样处理

一般通过A/D 变换处理方式, 模拟输入信号在一定时间间隔中收集, A/D 变换。模拟输入信号变为数字, 要保存在内存时, 所需的时间和用通道数有关。

$$(\text{处理时间}) = (\text{使用通道数}) \times (\text{变换速度})$$

(例) 4个通道中使用3个通道时, 处理时间= $3 \times 0.5\text{ms} = 1.5\text{ms}$

所谓采样是连续模拟信号在一定间隔作为抽样值取出的意思。

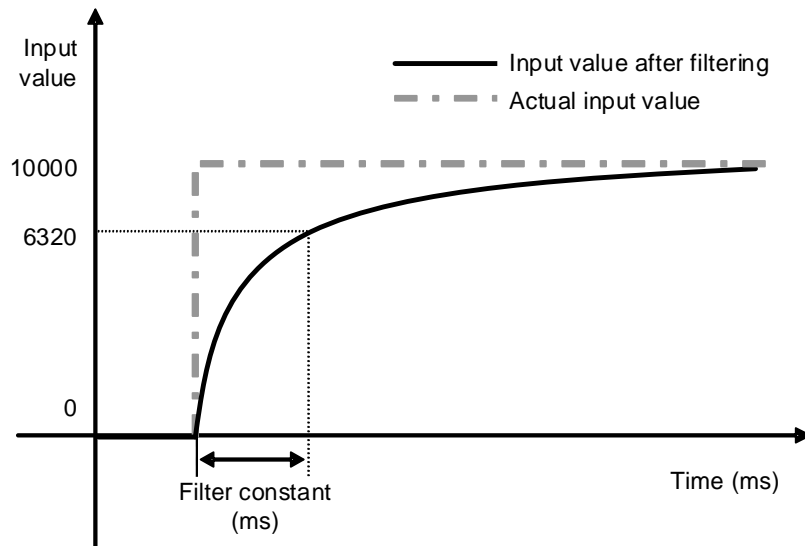
13.5.2 滤波功能

指定的通道输入值设定的滤波值(时定数 63.2%)，之前滤波的输入值如下计算后输出。

$$\text{当前过滤的输入值} = \frac{(\text{之前过滤的输入值} \times \text{过滤常数}) + (\text{当前输入值} \times 0.5\text{ms} \times \text{使用通道数})}{\text{过滤常数} + (0.5\text{ms} \times \text{使用通道数})}$$

模拟输入信号变为数字，要在内存中保存所需的时间和使用通道数有关。

滤波常数设定范围= 4~64,000 [ms]

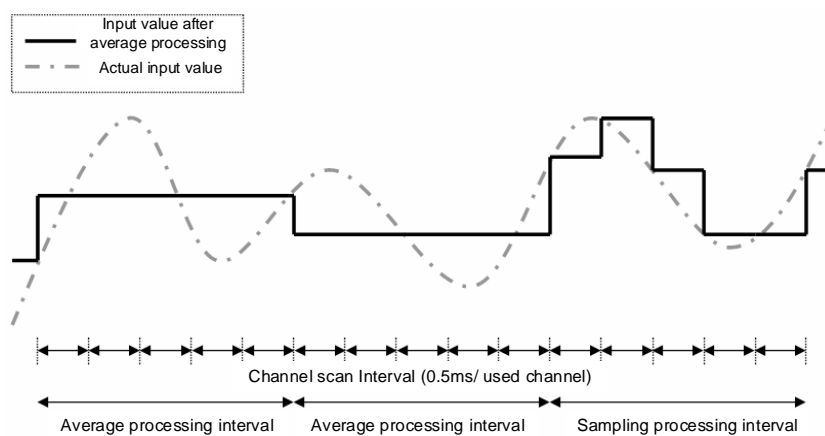


上述图表是从0到 10,000的输入值急剧变化的情况下，输入值滤波。
指定为滤波常数的时间是输入值变为实际 输入值时定数 63.2%左右。

13.5.3 平均功能

1) 时间平均

指定的通道输入值在设定时间期间累计，对此平均值以数字数据输出



设定范围= 4~16,000 [ms]

时间平均时，根据使用通道数进行如下平均处理次数计算

$$\text{平均处理次数[次]} = \frac{\text{平均时间值}}{\text{使用通道数} \times 0.5\text{ms}}$$

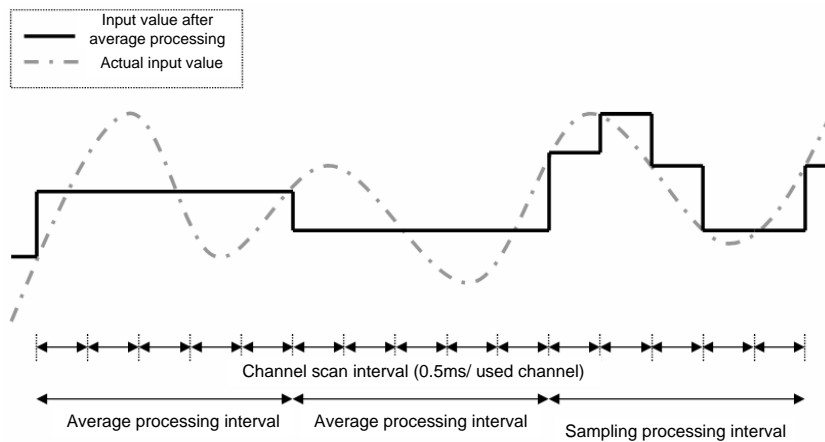
时间平均是 A/D 变换模块内变为计数平均来处理。此时，区分设定时间(使用通道数×变换速度)的过程中，剩下的可能发生。此时发生的剩余，废弃处理，平均处理测试是(设定时间)÷(使用通道数×变换速度)的商

(例)使用通道数 4, 设定时间为 151 ms,

$$151\text{ms} \div (4 \times 0.5\text{ms}) = 75\text{次} \cdots \text{剩余}1 \rightarrow 75\text{次}$$

2) 计数平均

指定的通道输入值在设定次数期间累计，对此平均值以数字数据输出



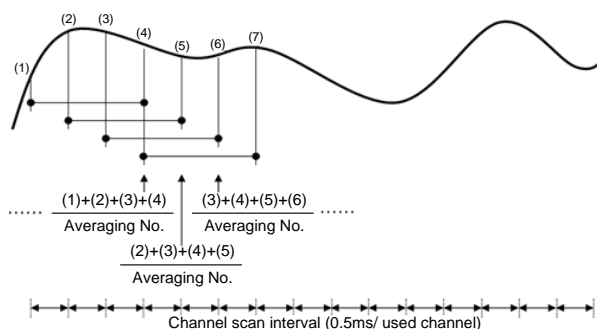
设定范围= 2 ~ 64,000 [次]

计数平均时，根据使用通道数来计算平均处理间隔

$$\text{平均处理间隔[ms]} = \text{平均次数} \times \text{使用通道数} \times 0.5\text{ms}$$

3) 移动平均

指定的通道输入值按设定的个数累计，对此平均值以数字数据输出但，移动平均时，每次扫描平均值输出



4) 加权平均

加权平均功能是因为输入采样数据作为滤波(延迟) 处理处理，输入数据的变化可圆满处理。

设定范围: 1~99(%)

$$F[n] = (1 - \alpha) \times A[n] + \alpha \times F[n - 1]$$

$F[n]$: 当前加重平均输出值

$A[n]$: 当前 A/D变动值

$F[n - 1]$: 之前的加重平均输出值

α : 加重平均常数(0.01 ~ 0.99 : 之前值的加重值)

设定值	滤波输出值				说明
	-	1scan	2scan	3scan	
未设定时	0	8,000	8,000	8,000	加权平均不处理
1	0	7,920	7,999	7,999	之前值 1% 反映的值
50	0	4,000	6,000	7,000	之前值 50% 反映的值
99	0	80	159	237	之前值 99% 反映的值

提示

(1) 时间/计数平均，输入值每次变换时间，不输出，到达平均时间/平均测试为止，维持之前状态的特性。

(2) 平均功能 2 种类可以和是以上介绍的滤波功能一起同时处理。同时选择时，处理顺序是滤波功能优先执行后，选择的平均功能应用，数字数据输出。此时数字数据输出值显示为最终处理后的值。

(3) 使用通道数包括输入，输出通道

13.5.4 警告功能(输入断线检出功能)

内置模拟模块的模拟输入范围为电压输入DC 1~5V或者电流输入DC 4~20mA时，输入的断线感知并显示的诊断功能。

模块检出断线并显示时，配线连接路径中，一部分连接不良时，请初始化。

1) 检出条件

使用4~20mA和1~5 V的输入信号范围时，输入回路的断线可以检出。
各自的输入信号范围，检出条件如下。

输入信号范围	识别为断线的电压/电流值
4~20mA	0.8mA 以下
1~5V	0.2V 以下

2) 使用的输入配线和模块之前断线时，LED 以 1S 为周期闪烁。

3) 各通道通道别断线可检出。但仅对于运行指定的通道，显示断线。LED 从通道 0 到通道 1 为止，共通使用，1 个通道异常断线时闪烁。

输入连接状态	通道运行	AD LED状态	断线标志位
正常	运行	灯亮	OFF
	停止	灯亮	OFF
输入断线或者不连接	运行	闪烁(1s)	ON
	停止	灯亮	OFF

4) 断线发生时相应通道的断线标志位 ON，断线解除时 OFF。

断线标志位	内容	状态说明
%UX0.1.72	通道0断线	OFF：正常 ON：断线
%UX0.1.73	通道1断线	

5) 断线发生时输入值是各输入范围中最低值显示

13.5.5 有效变换值维持功能

输入信号超出有效范围时，最后变换的有效输入值维持。
有效变换值维持功能是根据 用户程序或者 I/O参数设定，每个通道可指定。

1) 使用输入范围

允许有效变换值维持功能的通道中各数字变换范围中通过的使用范围显示。例如，输出数据类型设定为无符号值，运行时，现有数字输出值在 -192~16,191范围或者在允许此功能的状态下仅显示为0~16,000。设定有效变换值维持功能时，建议在有效输入范围以内的值读取状态下设定。

(1) 输入范围的数字输出值(无符号值，有符号值，百分位值)

区分	无符号值	有符号值	精准值	百分位值
功能未适用时	-192~16,191	-8,192~8,191	(2) 参考	-120~10,119
功能适用时	0~16,000	-8,000~8,000		0~10,000

(2) 输入范围的数字输出值(精确值)

模拟输入范围	区分	精准值
4~20mA	功能未适用时	3,808~20,191
	功能适用时	4,000~20,000
0~20mA	功能未适用时	-240~20,239
	功能适用时	0~20,000
1~5V	功能未适用时	952~5,047
	功能适用时	1,000~5,000
0~5V	功能未适用时	-60~5,059
	功能适用时	0~5,000
0~10V	功能未适用时	-120~10,119
	功能适用时	0~10,000
-10~10V	功能未适用时	-10,240~10,239
	功能适用时	-10,000~10,000

2) 动作

允许此功能的状态下按 4~20mA范围运行，采样的瞬间输入值变化的话输出值如下。(输出数据类型：0~16,000时)

输入电流(mA)	12mA	3mA	4mA	12mA	21mA	20mA
数字输出值	8,000	8,000	0	8,000	8,000	16,000
备注	-	之前值维持	-	-	之前值维持	-

13.5.6 警报功能

输入信号超出输入范围时，通过相应通道的警告标志位提示。

1) 输入检出条件

各个输入信号范围的检出条件如下

模拟输入范围	信号差(差异)	允许范围	警报下限	警报上限
4~20mA	16mA	1.2%	3.808mA	20.192mA
0~20mA	20mA		-0.240mA	20.240mA
1~5V	4V		0.952V	5.048V
0~5V	5V		-0.060V	5.060V
0~10V	10V		-0.120V	10.120V
-10~10V	20V		-10.240V	10.240V

2) 通道别警报显示

各输入通道的警报检出信号在%UX0.1.48~%UX0.1.49和%UX0.1.56~%UX0.1.57中显示。
输入信号恢复为有效范围内，警报检出显示会自动恢复。

(1) 警报上限

变量	设备分配	说明文	状态说明
_01_AD0_HOOR	%UX0.1.48	输入通道0警报上限	OFF：正常 ON：警报上限发生
_01_AD1_HOOR	%UX0.1.49	输入通道1警报上限	

(2) 警报下限

变量	设备分配	说明文	状态说明
_01_AD0_LOOR	%UX0.1.56	输入通道0警报下限	OFF：正常 ON：警报下限发生
_01_AD1_LOOR	%UX0.1.57	输入通道1警报下限	

提示

运行状态的输入通道中指定有效值 范围为ON的状态，在输入超出如下表有效值指定范围的信号的状态下初期电源投入时，通道变换数据为0， 警报下限标志位为ON。

模拟输入范围	有效值范围指定	输入信号	警报下限	通道变换值
4~20mA	ON	3.808mA~4mA	ON	0
		20mA~20.192mA		
0~20mA	ON	-0.24mA~0mA	ON	0
		20mA~20.24mA		
1~5V	ON	0.952V~1V	ON	0
		5V~5.048V		
0~5V	ON	-0.06V~0V	ON	0
		5V~5.06V		
0~10V	ON	-0.12V~0V	ON	0
		10V~10.12V		
10~10V	ON	-10.24V~-10V	ON	0
		10V~10.24V		

13.5.7 通道输出状态设定

对应运动控制停止及异常状态的输出设定。

1) 功能

模块的初始化或者运动控制的错误发生时，为了阻止模块的误输出而使用。

2) 种类

通道输出状态可设定为之前值输出，最小值输出，中间值输出，最大值输出的4个中的一个。

- (1) 之前值输出: 正常运行的最后输出维持
- (2) 最小值输出: 各输出范围别最小值输出
- (3) 中间值输出: 各输出范围别中间值输出
- (4) 最大值输出: 各输出范围别最大值输出

3) 示例

输出通道的范围设定为0 ~ 10V，输出为7V时，系统run变为stop 状态，通道输出状态设定内容如下输出

- (1) 之前值输出: 之前输出是7V 输出维持
- (2) 最小值输出: 相应范围的最小值是0V输出
- (3) 中间值输出: 相应范围的中间值是5V输出
- (4) 最大值输出: 相应范围的最大值是10V输出

13.5.8 插补方法设定

1) 功能

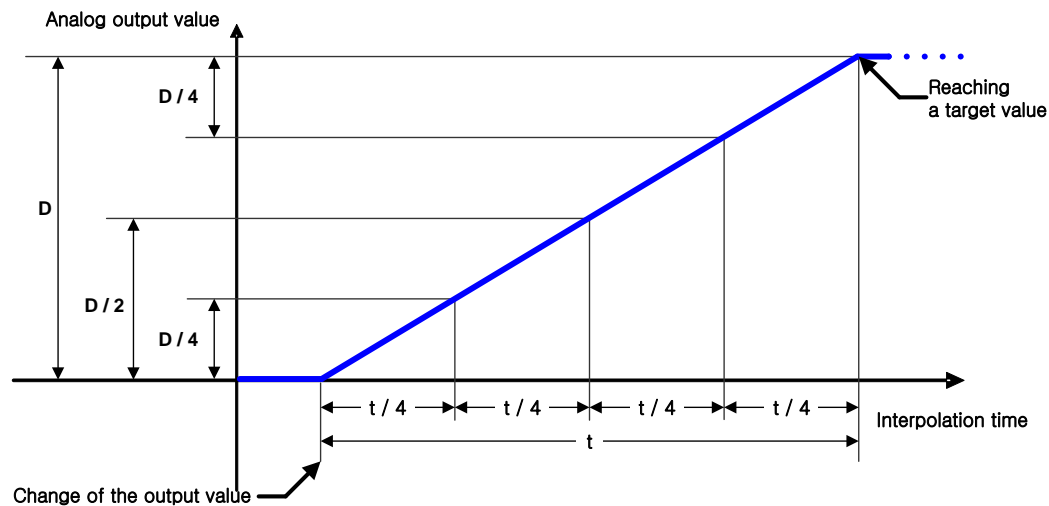
根据设定的插补时间，模块的输出信号为了插补输出而使用。电压输出时通过举例的输出，防止负载系统过度应答而使用。

2) 插补方法设定

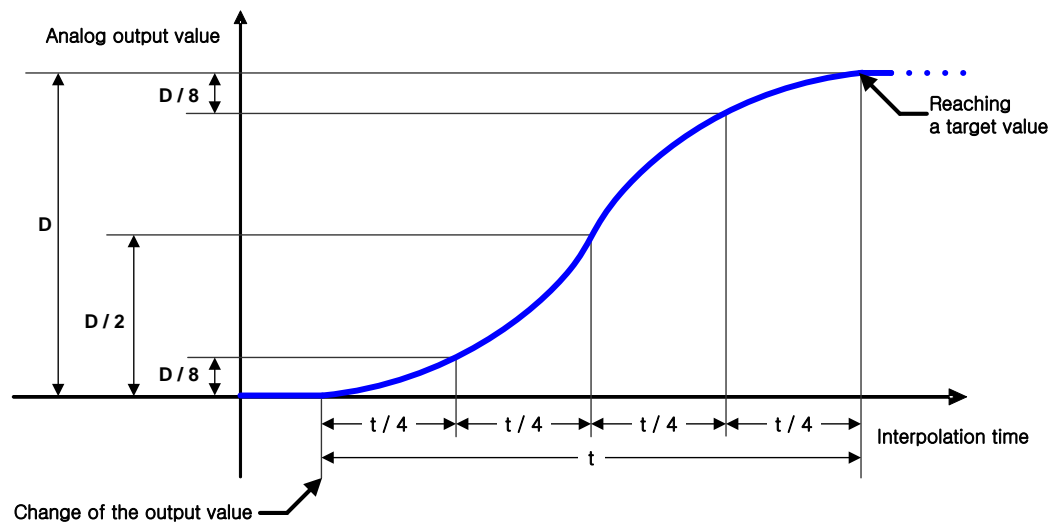
插补方法设定为插补禁止，直线插补，S型插补中的一个

(1) 插补禁止：插补动作不执行，数字输入值按原样输出。

(2) 直线插补：插补时间期间输出到目标值为止按直线变化。



(3) S型插补：插补时间期间输出到目标值为止，S型变化



3) 插补时间设定

插补时间设定为10[ms], 100[ms], 1[s], 60[s] 中的一个
根据设定的插补时间期间插补方法设定输出变化

4) 插补输出值

插补功能使用时当前输出，插补演算值在内部内存领域可确认。

变量	插补输出值地址	内容
_01_DA0_INTPVAL	%UW0.1.25	通道0 插补演算值
_01_DA1_INTPVAL	%UW0.1.26	通道1 插补演算值

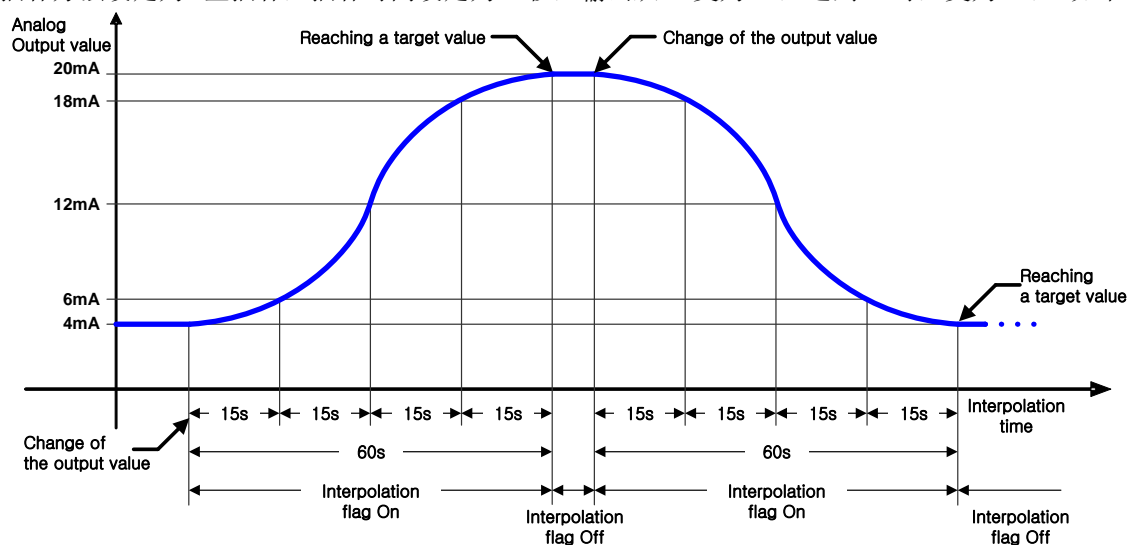
5) 插补输出中的插补标志位为 ON，插补输出值达到目标值的话，变为 OFF。

变量	插补标志位	内容
_01_DA0_INTP	%UX0.1.64	通道0 插补输出中
_01_DA1_INTP	%UX0.1.65	通道1 插补输出中

※ 插补时间设定为1[s], 60[s]时，插补标志位可监控。

6) 示例

插补方法设定为S型插补，插补时间设定为60秒，输出从1V变为5V，达到5V时，变为1V，如下输出。



提示

- 1) 插补输出中相应模块的内部参数变更的话，插补演算临时中断，输出作为目标值及时变更
- 2) 模块内部参数需要变更时，插补输出中标志位为OFF后，在模拟输出值不变更的状态下，进行参数变更。

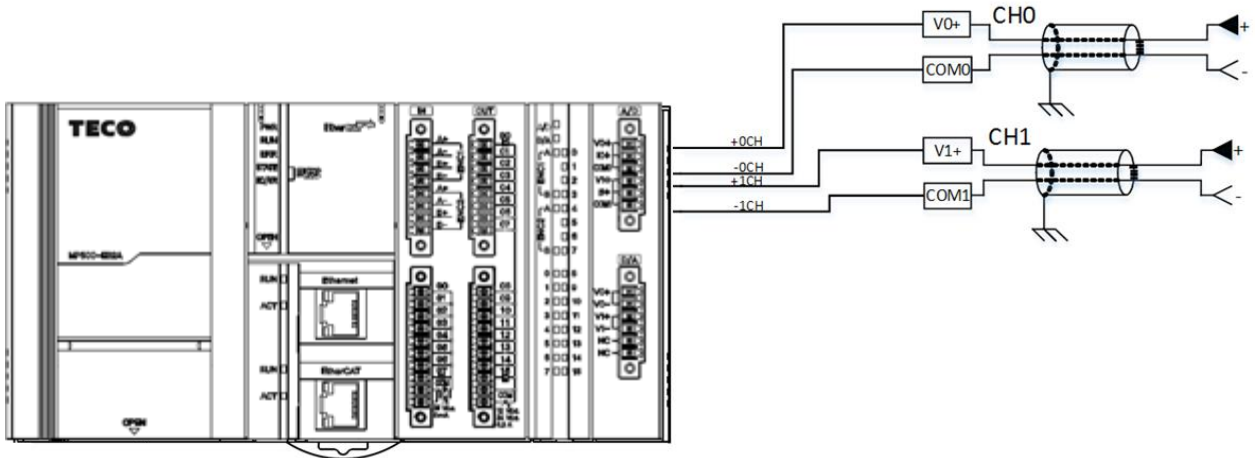
13.6 配线

13.6.1 模拟输入配线示例

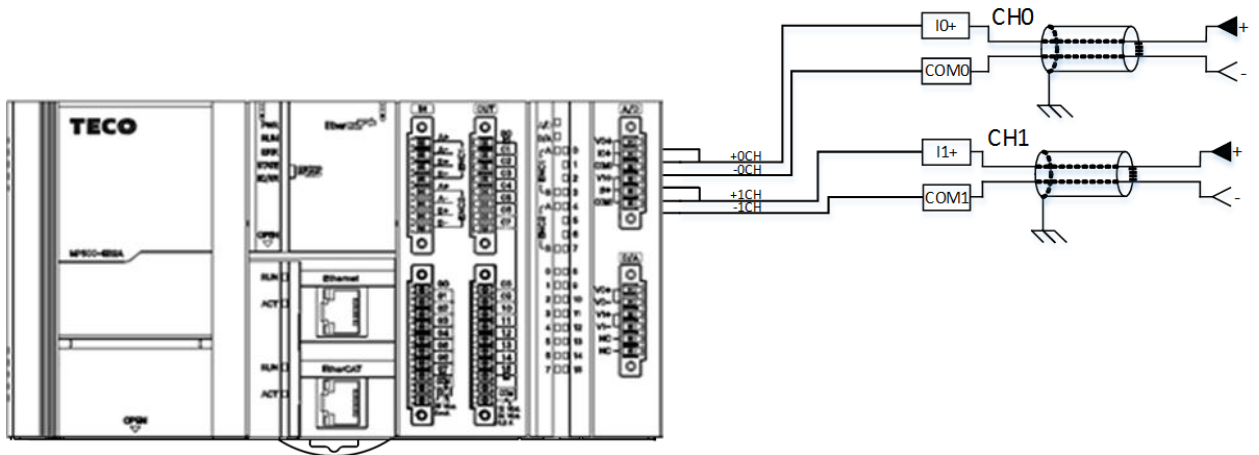
- 1) 电流输入回路的输入电阻为 250Ω (typ.)
- 2) 电压输入回路的输入电阻 $1M\Omega$ 以上
- 3) 仅使用的通道设定为通道运行来使用。
- 4) 模拟输入配线示例

使用电压输入时相应通道的V+和COM 端子，电流输入时使用V+和I+端子连接后 I+和COM端子。

(1) 电压配线

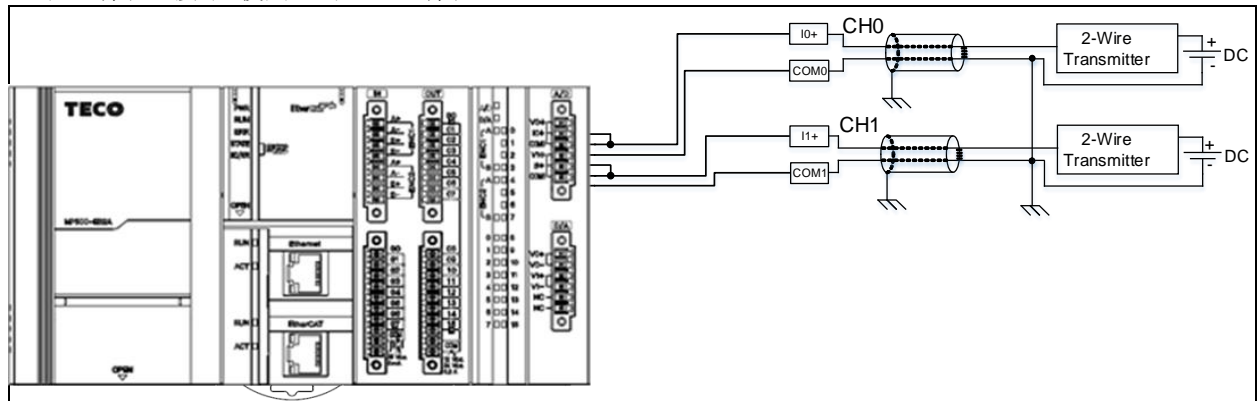


(2) 电流配线

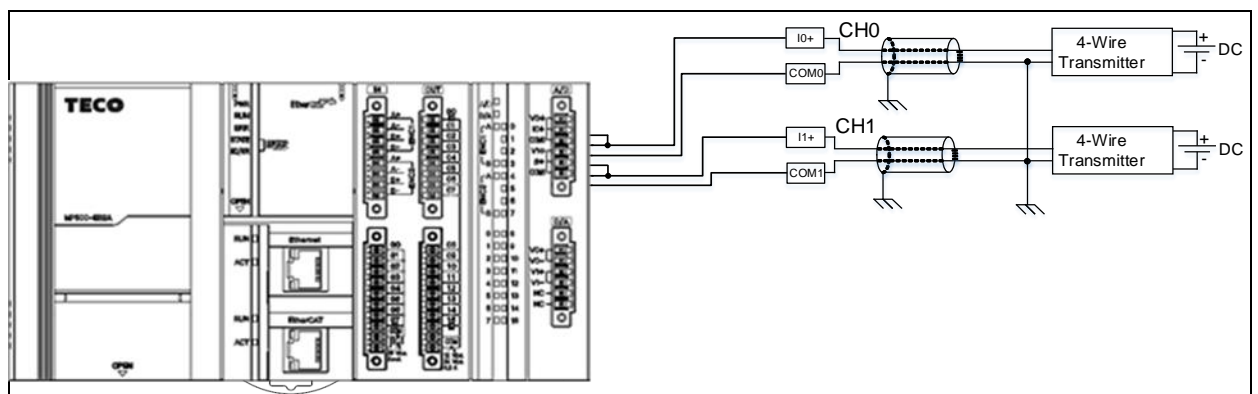


第 13 章 内置模拟量功能

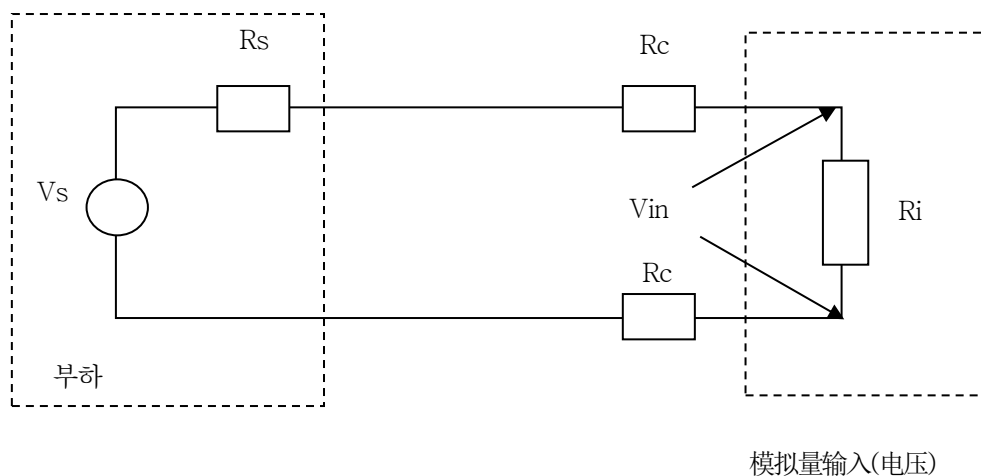
- 5) 模拟输入 2-线传感器/发送器配线示例(电流输入)
V+和I+端子连接 后使用 I+和COM端子



- 6) 模拟输入 4-线传感器/发送器配线示例(电流输入)
V+和I+端子连接 后使用I+和COM端子



- 7) 电压输入精度和配线长度的关系
电压输入中发送器或者传感器和模块间的配线长度对模块的数字变换值有影响。
其值如下。



此处

- Rc : 电线线路电阻的电阻值
- Rs : 发送器(发送器)或者传感器(传感器)的内部电阻值
- Ri : 电压输入模块的内部电阻值(1MΩ)
- Vin : 模拟输入投入的电压

% Vi : 根据电压输入中的电源和电线长度, 变换值误差(%)

$$V_{in} = \frac{R_i \times V_s}{R_s + (2 \times R_c) + R_i}$$

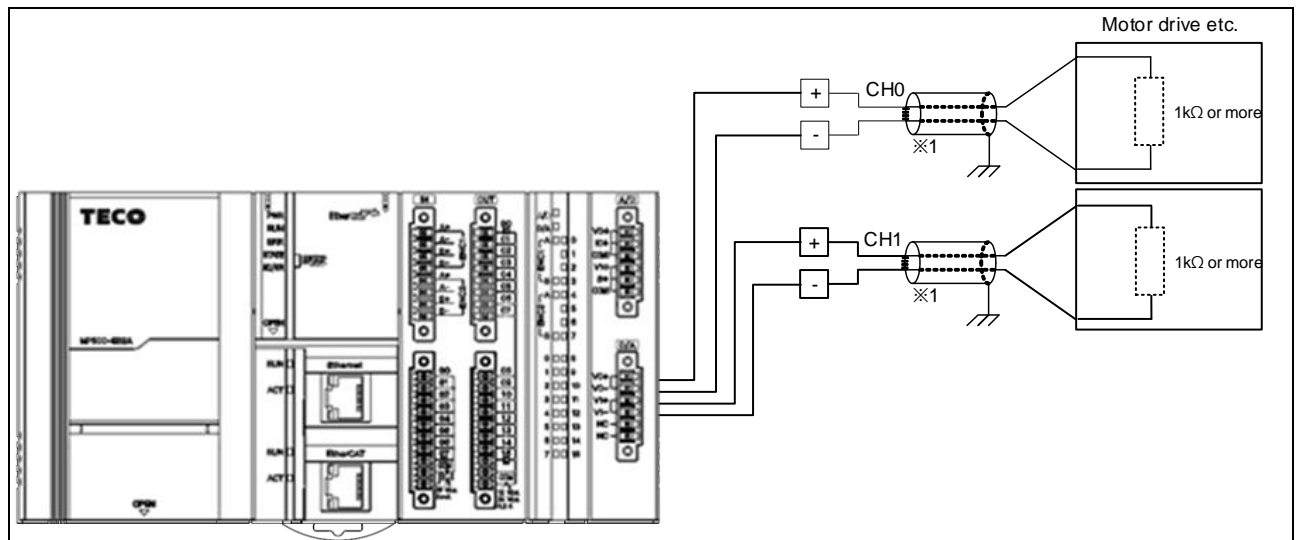
$$\%V_i = \left(1 - \frac{V_{in}}{V_s}\right) \times 100\%$$

提示

1) 输入范围在电压(1~5V, 0~5V, 0~10V, -10~10V)的状态下使用时, 外部配线断线时, 输出数据值从 0V 变为相应值为止, 需要一定的时间。如果想要减少这个时间, 输入通道V+和COM直接要接入0.1 MΩ~1MΩ电阻

13.6.2 模拟输出配线示例

1) 模拟电压输出配线示例



※1: 电线请使用2芯的屏蔽线

13.7 运行参数设定

内置模拟的运行参数通过 MP500的[I/O参数]来设定。

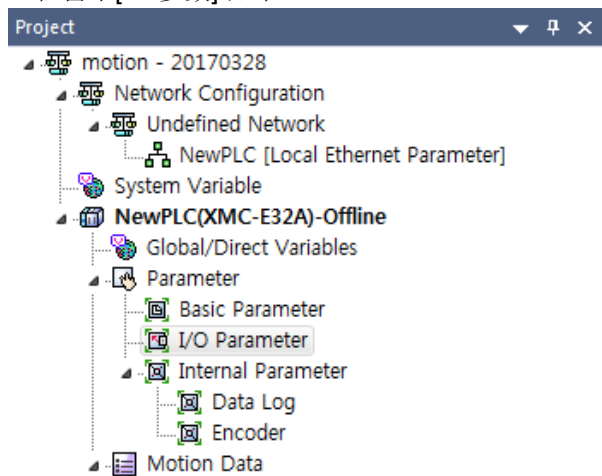
1) 设定项目

为了便于使用，MP500中内置模拟模块的参数设定通过GUI(图形用户界面)方式提供。MP500的工程窗口中通过[I/O参数]来设定的项目如下。

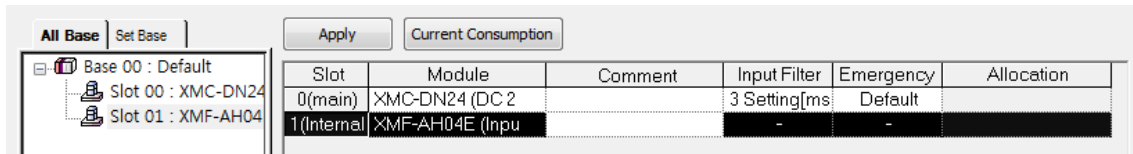
项目	内容
[I/O参数]	<p>(a) 输入参数设定 模块动作所需的如下项目设定。</p> <ol style="list-style-type: none"> 1) 运行通道 2) 输入范围 3) 输出数据类型 4) 滤波常数 5) 平均方法 6) 平均值 7) 有效变换值维持 <p>(b) 输出参数设定 模块动作所需的如下项目设定。</p> <ol style="list-style-type: none"> 1) 运行通道 2) 输出范围 3) 输入数据类型 4) 通道输出状态设定 5) 插补方法设定 6) 插补时间 <p>(c) MP500 中用户设定的参数下载结束后， 在运动控制中保存</p>

2) [I/O 参数]使用方法

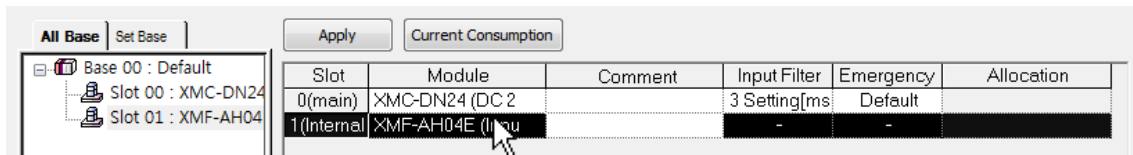
- (1) MP500运行，工程选择。(工程生成方法参考MP500使用说明书)
- (2) 工程窗中[I/O参数]双击



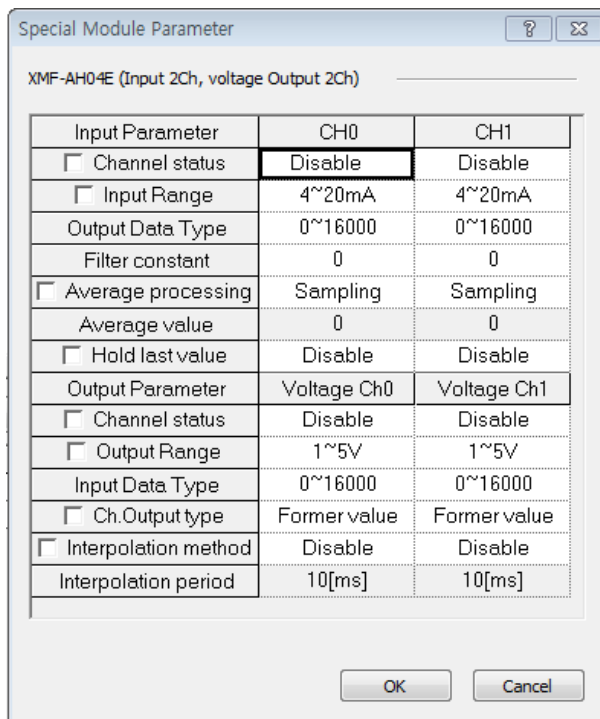
(3) [I/O参数]页面中槽1(内置功能)点击



(4) 参数设定要在选择相应槽的状态下双击



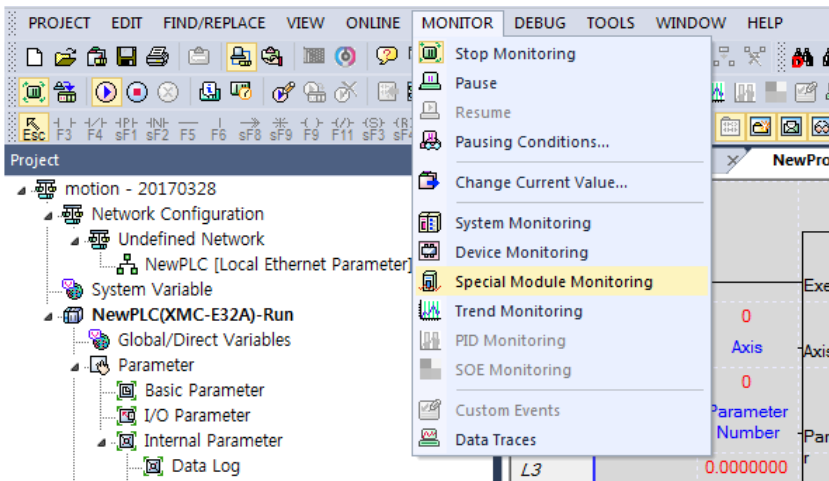
(5) 出现如下通道别进行参数设定的窗口
点击要设定的项目，显示项目别可设定的参数



13.8 特殊模块监控功能

监控的功能如下。

- 1) [特殊模块监控]启动[在线] -> [连接]状态下, [监控] -> [特殊模块监控]启动。不是[在线]状态时, [特殊模块监控] 菜单不激活

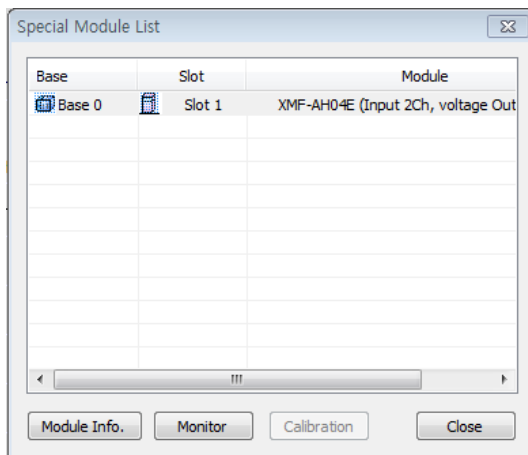


提示

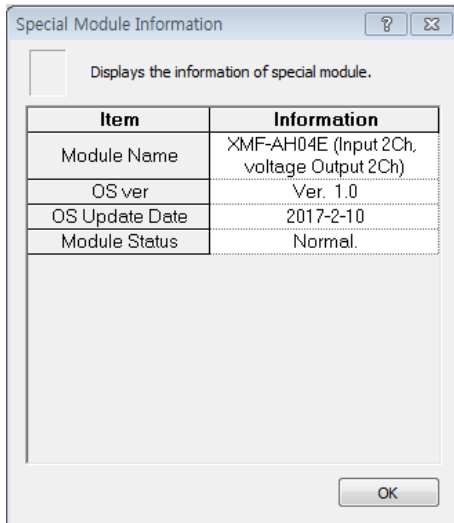
- 1) 系统资源不足导致无法正常显示。把这样的画面关闭, 其他应用结束后再次执行MP500。
- 2) [特殊模块监控]状态下执行设定的I/O 参数要临时设定。因此, [特殊模块监控]状态结束的话, 设定的I/O参数消失。
- 3) [特殊模块监控]的测试执行在不生成顺序程序的状态下, 模拟输入模块可以检查是否正常动作的功能。

2) [特殊模块监控]使用方法

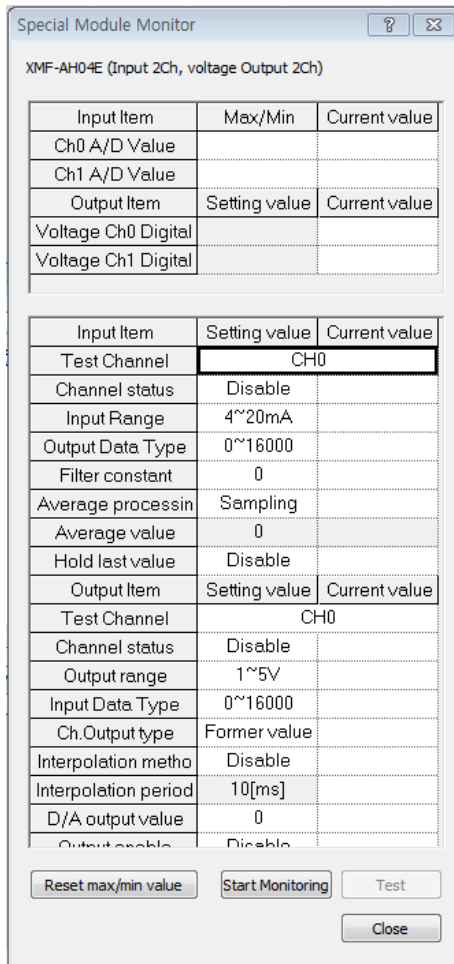
- (1) MP500和运动控制连接的状态(在线状态)下点击[监控] -> [特殊模块监控]。
如下图 '特殊模块列表' 页面出现, 特殊模块种类基板/槽信息显示。



(2) 选择模块，点击[模块信息]，出现如下画面



(3) “特殊模块列表”画面中点击[监控]出现如下‘特殊模块监控’。



第 13 章 内置模拟量功能

(4) [监控开始]: 点击 [监控开始], 显示当前运行的通道的数字输入输出数据

Special Module Monitor

XMF-AH04E (Input 2Ch, voltage Output 2Ch)

Input Item	Max/Min	Current value
Ch0 A/D Value	0 / 0	0
Ch1 A/D Value	0 / 0	0
Output Item	Setting value	Current value
Voltage Ch0 Digital		0
Voltage Ch1 Digital		0

检测

Input Item	Setting value	Current value
Test Channel	CH0	
Channel status	Disable	Disable
Input Range	4~20mA	4~20mA
Output Data Type	0~16000	0~16000
Filter constant	0	0
Average processin	Sampling	Sampling
Average value	0	0
Hold last value	Disable	Disable
Output Item	Setting value	Current value
Test Channel	CH0	
Channel status	Disable	Disable
Output range	1~5V	1~5V
Input Data Type	0~16000	0~16000
Ch.Output type	Former value	Former value
Interpolation metho	Disable	Disable
Interpolation period	10[ms]	10[ms]
D/A output value	0	0
Output enable	Disable	Disable

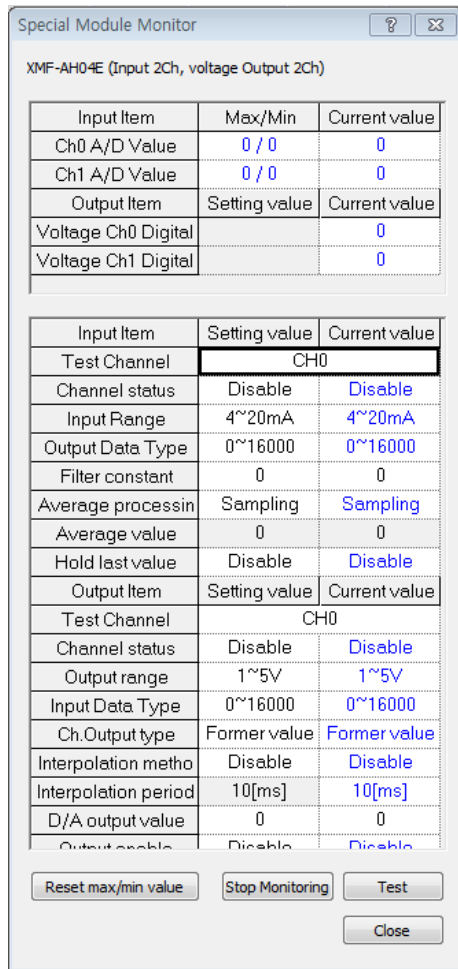
输入通道0 详细信息

电压输出通道0 详细信息

Reset max/min value Stop Monitoring Test

Close

- (5) [测试执行]：[测试执行]是当前设定的模拟模块参数变更时使用的功能。点击画面下端的设定值，参数变更。
 [测试执行]仅在运动控制运行状态为STOP时可以设定。



[测试执行]执行页面

第 13 章 内置模拟量功能

(6) 最大值/最小值监控

运行中的输入通道的最大值和最小值可监控。

但, 这里显示的最大值/最小值是按 页面显示的当前值基准的值。

[监控/测试页面]关闭时, 最大值/最小值保存。

Special Module Monitor

XMF-AH04E (Input 2Ch, voltage Output 2Ch)

Input Item	Max/Min	Current value
Ch0 A/D Value	-192 / -192	-192
Ch1 A/D Value	0 / 0	0

Output Item	Setting value	Current value
Voltage Ch0 Digital		0
Voltage Ch1 Digital		0

Input Item	Setting value	Current value
Channel status	Enable	Enable
Input Range	4~20mA	4~20mA
Output Data Type	0~16000	0~16000
Filter constant	0	0
Average processin	Sampling	Sampling
Average value	0	0
Hold last value	Disable	Disable

Output Item	Setting value	Current value
Test Channel	CH0	
Channel status	Enable	Enable
Output range	1~5V	1~5V
Input Data Type	0~16000	0~16000
Ch.Output type	Former value	Former value
Interpolation metho	Disable	Disable
Interpolation period	10[ms]	10[ms]
D/A output value	0	0
Output enable	Enable	Enable

Reset max/min value Stop Monitoring Test Close

最大/最小值监控

最大/最小值reset

(7) 关闭

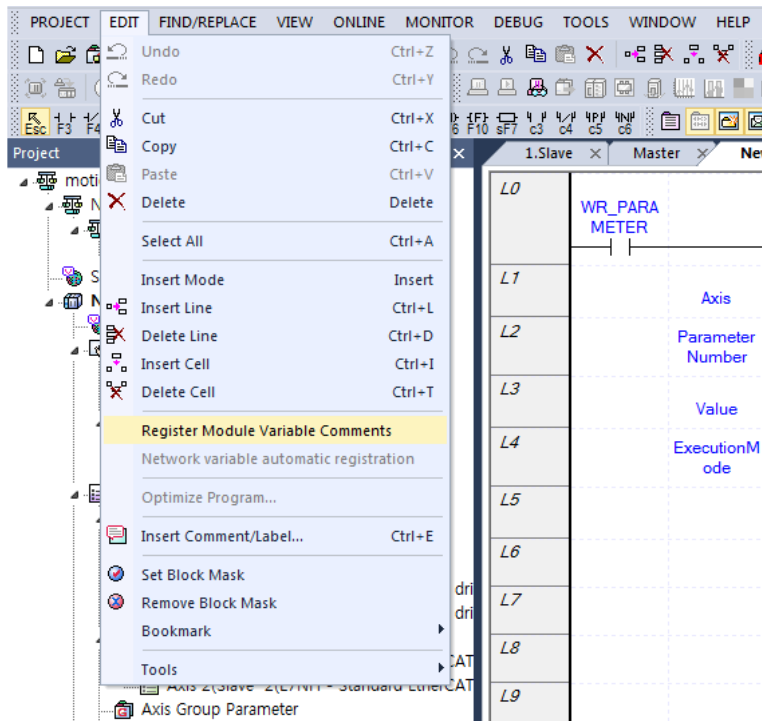
[关闭]按钮是监控/测试页面跳转时使用的监控/测试页面关闭时, 最大值、最小值、当前值不能再储存更多。

13.9 模块变量自动登录

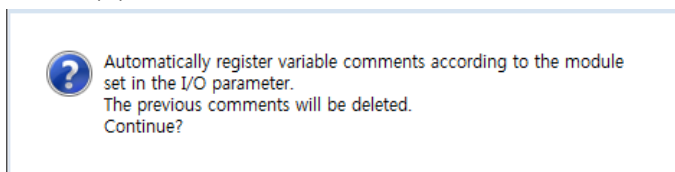
运动控制内置模拟模块的变量自动登录。
用户可以修改变量及说明文

1) 登录顺序

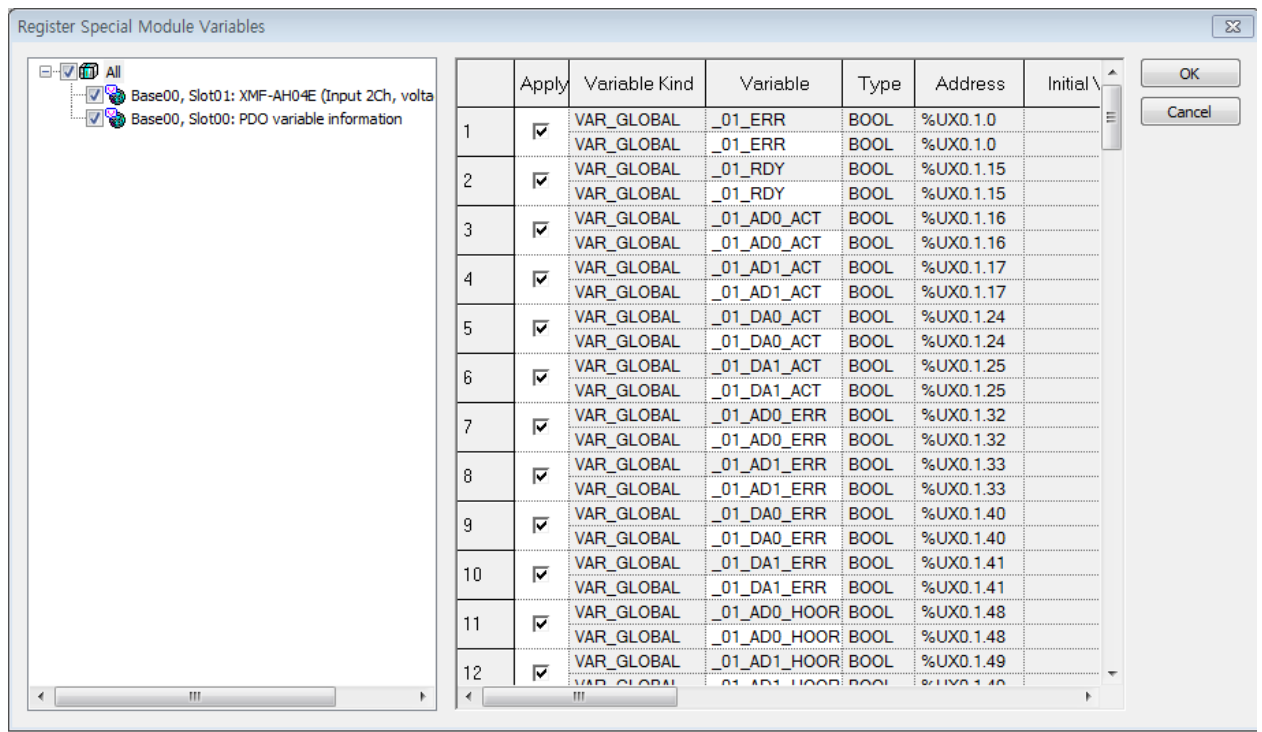
(1) 菜单‘编辑’中选择‘模块变量自动登录(G)’



(2) 点击‘是(Y)’:



(3) 如下页面，变量登录。

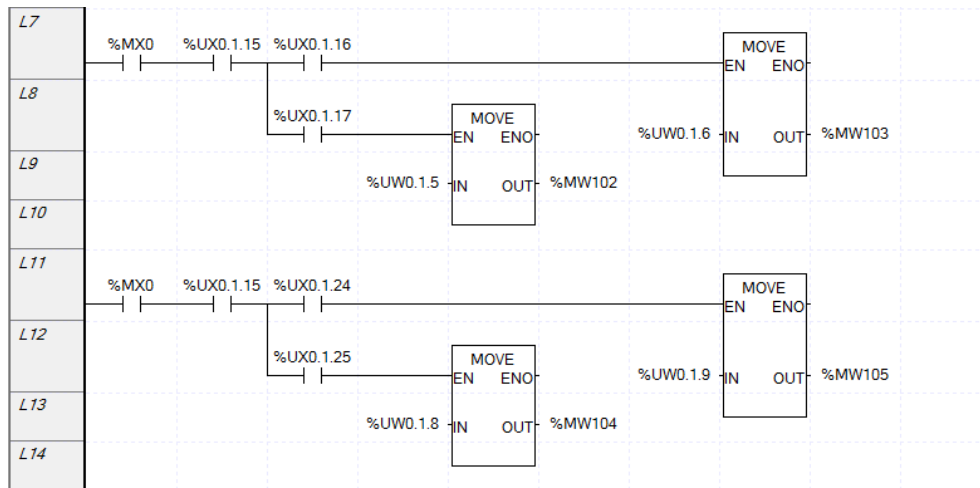


2) 变量保存

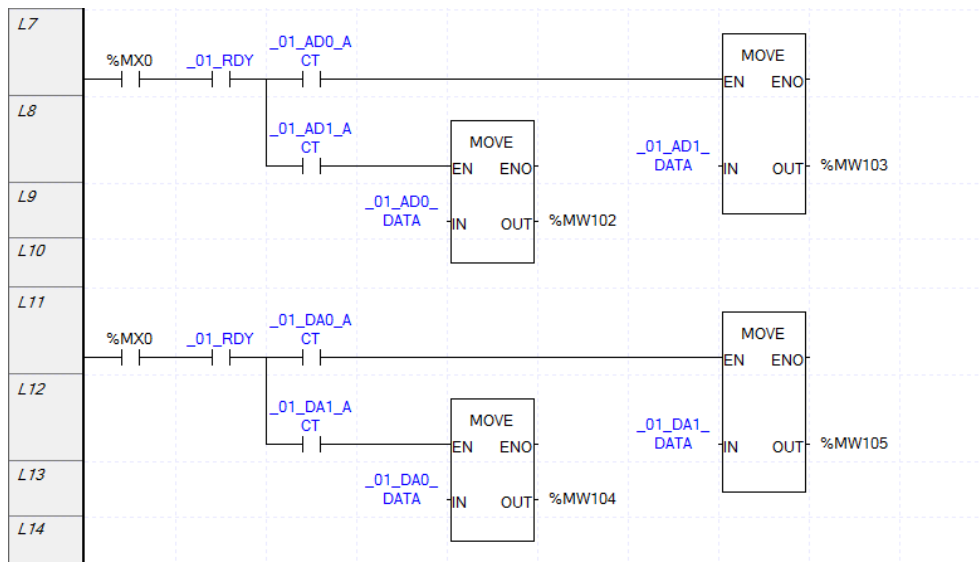
- (1) ‘查看变量’内容可以保存为文本文件
- (2) 选择[编辑] → [文本导出]
- (3) ‘查看变量’内容保存为文本文件

3) 程序的变量查看

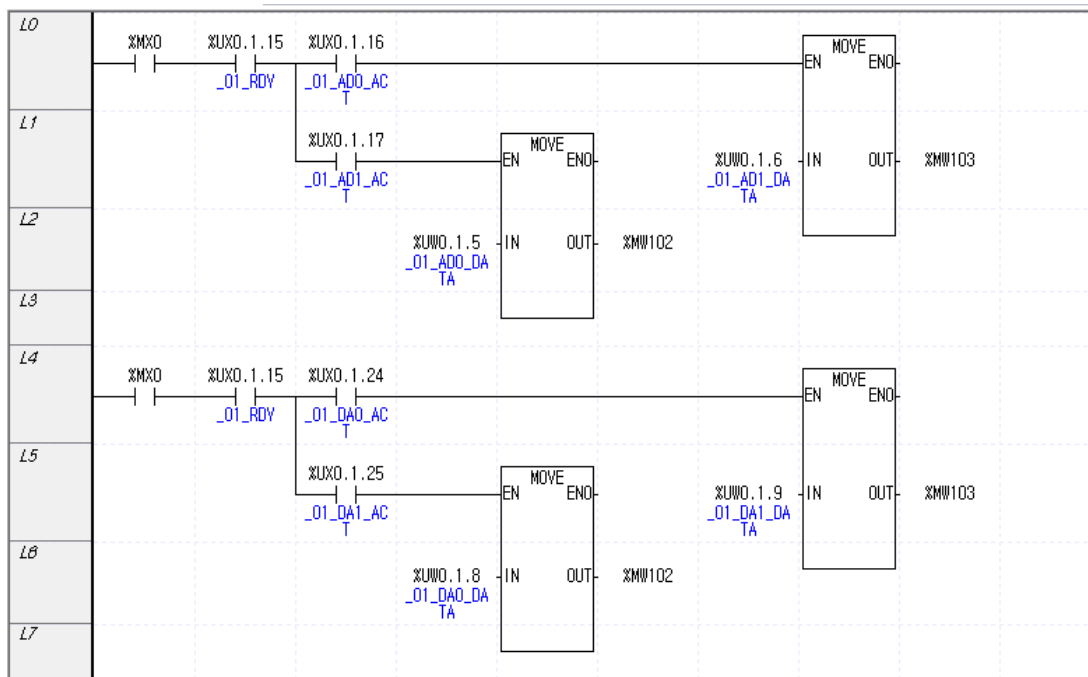
- (1) MP500的示例程序如下。



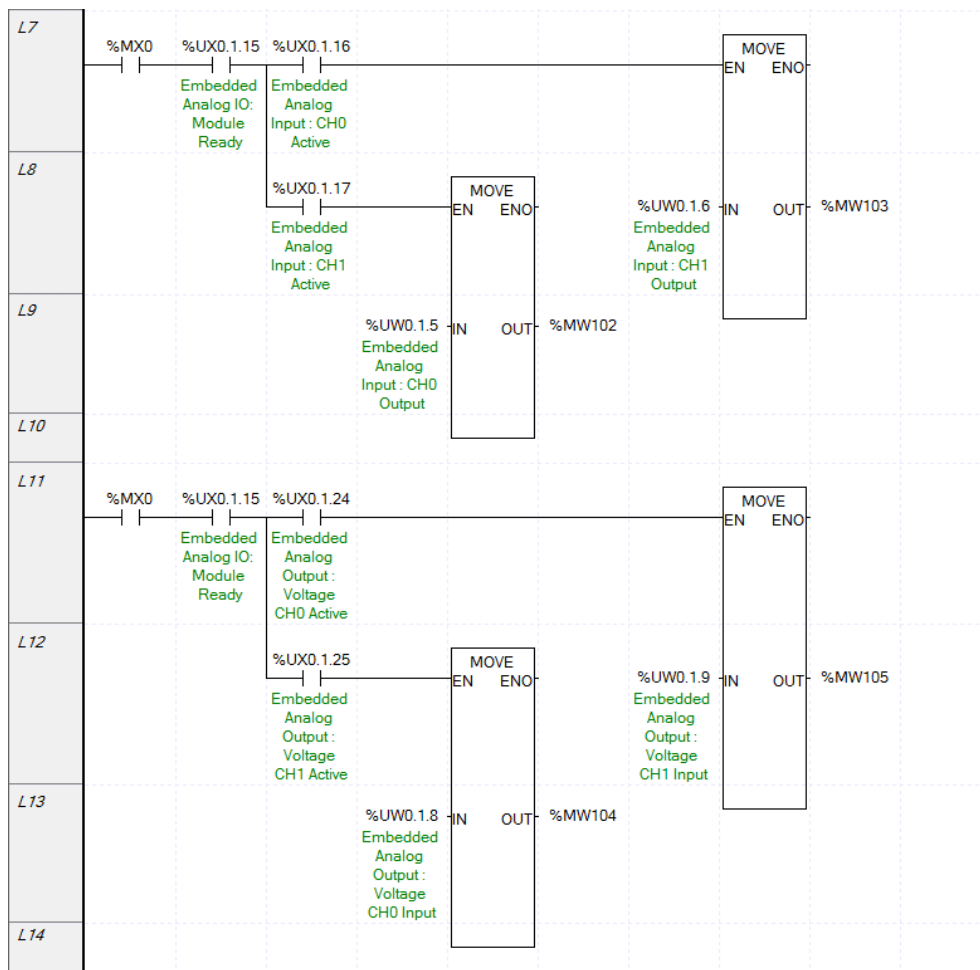
- (2) 菜单的‘查看’中点击‘变量查看’。设备变更为变量。



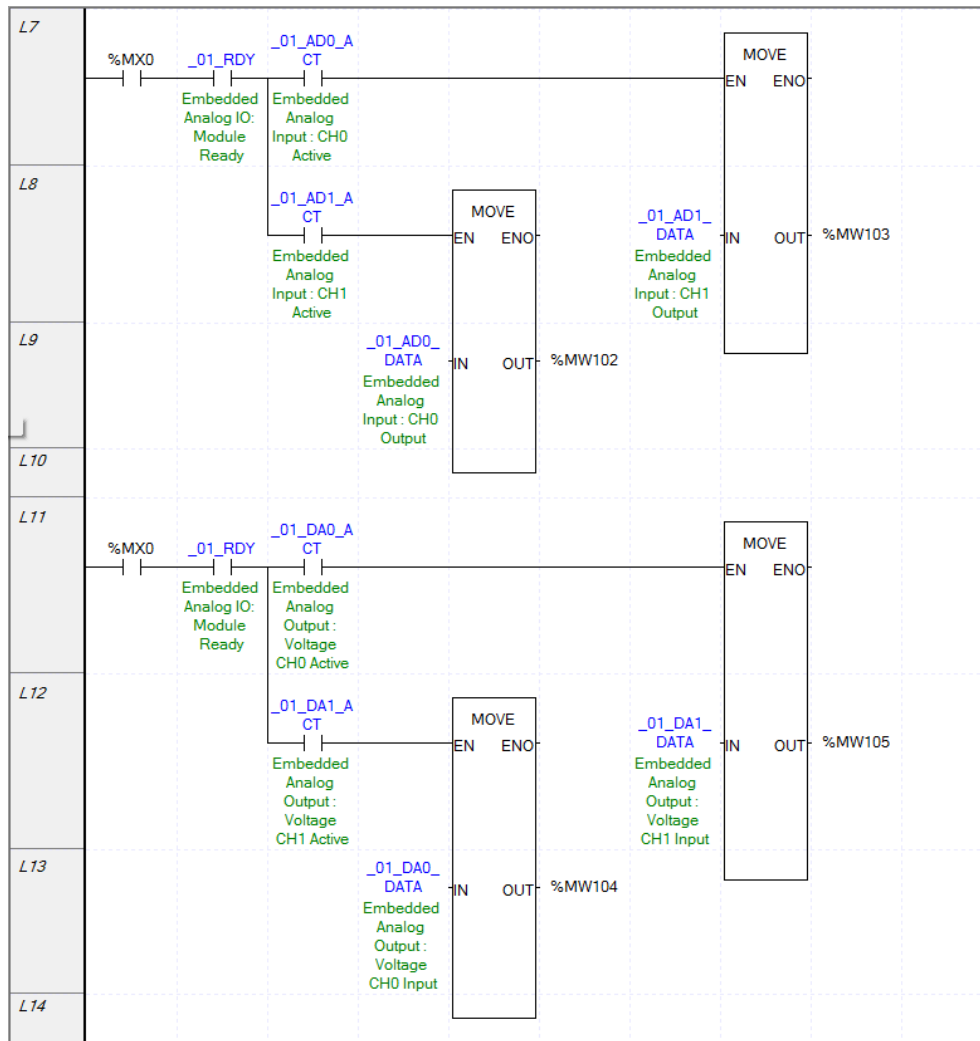
(3) 菜单的‘查看’中点击‘设备/变量查看’。设备和变量同时可变。



(4) 菜单的‘查看’点击‘设备/注释查看’。设备和注释同时可变。



(5) 菜单的‘查看’中点击‘变量/注释查看’。变量和注释同时可变。



13.10 内部内存的构成和功能

13.10.1 内置模拟数据输入输出领域

内置模拟数据输入输出领域如下

内置模拟输入

变量	类型	内存分配	注释
_01_AD0_ACT	BOOL	%UX0.1.16	通道0运行中
_01_AD0_AVGTYPE	BYTE	%UB0.1.34	通道0平均处理
_01_AD0_AVGVAL	WORD	%UW0.1.18	通道0平均值
_01_AD0_DATA	WORD	%UW0.1.5	通道0变换值
_01_AD0_DATATYPE	BYTE	%UB0.1.26	通道0数据类型指定
_01_AD0_ERR	BOOL	%UX0.1.32	通道0错误
_01_AD0_FILTCONST	WORD	%UW0.1.15	通道0滤波常数
_01_AD0_HOLDVAL	BOOL	%UX0.1.320	通道0有效变换值维持指定
_01_AD0_HOOR	BOOL	%UX0.1.48	通道0警报上限
_01_AD0_IDD	BOOL	%UX0.1.72	通道0输入断线检出
_01_AD0_LOOR	BOOL	%UX0.1.56	通道0警报下限
_01_AD0_RANGE	BYTE	%UB0.1.22	通道0范围指定
_01_AD0_RUN	BOOL	%UX0.1.160	通道0运行设定
_01_AD1_ACT	BOOL	%UX0.1.17	通道1运行中
_01_AD1_AVGTYPE	BYTE	%UB0.1.35	通道1平均处理
_01_AD1_AVGVAL	WORD	%UW0.1.19	通道1平均值
_01_AD1_数据	WORD	%UW0.1.6	通道1变换值
_01_AD1_数据类型	BYTE	%UB0.1.27	通道1数据类型指定
_01_AD1_ERR	BOOL	%UX0.1.33	通道1错误
_01_AD1_FILTCONST	WORD	%UW0.1.16	通道1滤波常数
_01_AD1_HOLDVAL	BOOL	%UX0.1.321	通道1有效变换值维持指定
_01_AD1_HOOR	BOOL	%UX0.1.49	通道1警报上限
_01_AD1_IDD	BOOL	%UX0.1.73	通道1输入断线检出
_01_AD1_LOOR	BOOL	%UX0.1.57	通道1警报下限
_01_AD1_RANGE	BYTE	%UB0.1.23	通道1范围指定
_01_AD1_RUN	BOOL	%UX0.1.161	通道1运行设定
_01_AD_ACT_ARY	ARRAY[0..1] OF BOOL	%UX0.1.16	通道别运行中
_01_AD_AVGTYPE_ARY	ARRAY[0..1] OF BYTE	%UB0.1.32	通道别平均处理
_01_AD_AVGVAL_ARY	ARRAY[0..1] OF WORD	%UW0.1.18	通道别平均值
_01_AD_DATATYPE_ARY	ARRAY[0..1] OF BYTE	%UB0.1.26	通道别数据类型指定
_01_AD_DATA_ARY	ARRAY[0..1] OF WORD	%UW0.1.5	通道别变换值
_01_AD_ERR_ARY	ARRAY[0..1] OF BOOL	%UX0.1.32	通道别错误
_01_AD_FILTCONST_ARY	ARRAY[0..1] OF WORD	%UW0.1.15	通道别滤波常数

变量	类型	内存分配	注释
_01_AD_HOLDVAL_ARY	ARRAY[0..1] OF BOOL	%UX0.1.320	通道别有效变换值维持指定
_01_AD_HOOR_ARY	ARRAY[0..1] OF BOOL	%UX0.1.48	通道别警报上限
_01_AD_IDD_ARY	ARRAY[0..1] OF BOOL	%UX0.1.72	通道别输入断线检出
_01_AD_LOOR_ARY	ARRAY[0..1] OF BOOL	%UX0.1.56	通道别警报下限
_01_AD_RANGE_ARY	ARRAY[0..1] OF BYTE	%UB0.1.22	通道别范围指定
_01_AD_RUN_ARY	ARRAY[0..1] OF BOOL	%UX0.1.160	通道别运行设定

内置模拟输出

变量	类型	内存分配	注释
_01_DA0_ACT	BOOL	%UX0.1.24	电压通道0运行中
_01_DA0_DATA	WORD	%UW0.1.8	电压通道0输入值
_01_DA0_DATATYPE	BYTE	%UB0.1.28	通道0数据类型指定
_01_DA0_ERR	BOOL	%UX0.1.40	电压通道0错误
_01_DA0_INTP	BOOL	%UX0.1.64	电压通道0插补输出中
_01_DA0_INTPMTHD	BYTE	%UB0.1.46	通道0插补方法设定
_01_DA0_INTPTIME	BYTE	%UB0.1.48	通道0插补时间设定
_01_DA0_INTPVAL	WORD	%UW0.1.25	通道0插补演算值
_01_DA0_OUTEN	BOOL	%UX0.1.112	电压通道0输出允许设定
_01_DA0_OUTSTAT	WORD	%UW0.1.21	通道0输出状态设定
_01_DA0_RANGE	BYTE	%UB0.1.24	通道0范围指定
_01_DA0_RUN	BOOL	%UX0.1.168	通道0运行设定
_01_DA1_ACT	BOOL	%UX0.1.25	电压通道1运行中
_01_DA1_DATA	WORD	%UW0.1.9	电压通道1输入值
_01_DA1_DATATYPE	BYTE	%UB0.1.29	通道1数据类型指定
_01_DA1_ERR	BOOL	%UX0.1.41	电压通道1错误
_01_DA1_INTP	BOOL	%UX0.1.65	电压通道1插补输出中
_01_DA1_INTPMTHD	BYTE	%UB0.1.47	通道1插补方法设定
_01_DA1_INTPTIME	BYTE	%UB0.1.49	通道1插补时间设定
_01_DA1_INTPVAL	WORD	%UW0.1.26	通道1插补演算值
_01_DA1_OUTEN	BOOL	%UX0.1.113	电压通道1输出允许设定
_01_DA1_OUTSTAT	WORD	%UW0.1.22	通道1输出状态设定
_01_DA1_RANGE	BYTE	%UB0.1.25	通道1范围指定
_01_DA1_RUN	BOOL	%UX0.1.169	通道1运行设定
_01_DA_ACT_ARY	ARRAY[0..1] OF BOOL	%UX0.1.24	通道别运行中
_01_DA_数据类型_ARY	ARRAY[0..1] OF BYTE	%UB0.1.28	通道别数据类型指定
_01_DA_数据_ARY	ARRAY[0..1] OF WORD	%UW0.1.8	电压通道别输入值
_01_DA_ERR_ARY	ARRAY[0..1] OF BOOL	%UX0.1.40	通道别错误
_01_DA_INTPMTHD_ARY	ARRAY[0..1] OF BYTE	%UB0.1.46	通道别插补方法设定
_01_DA_INTPTIME_ARY	ARRAY[0..1] OF BYTE	%UB0.1.48	通道别插补时间设定

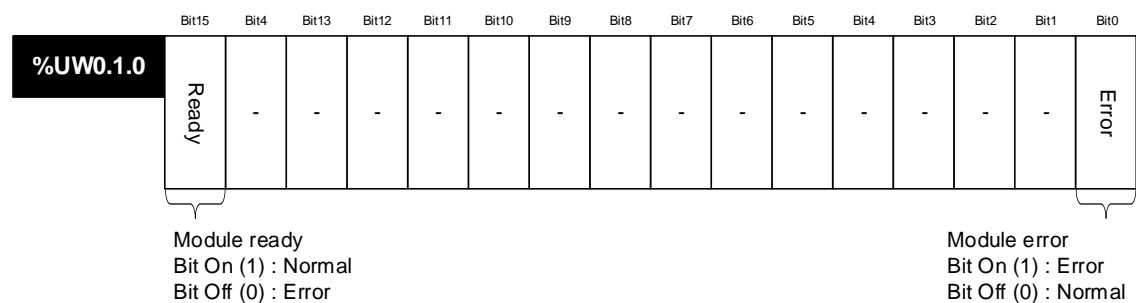
变量	类型	内存分配	注释
_01_DA_INTPVAL_ARY	ARRAY[0..1] OF WORD	%UW0.1.25	通道别插补演算值
_01_DA_INTP_ARY	ARRAY[0..1] OF BOOL	%UX0.1.64	通道别插补输出中
_01_DA_OUTEN_ARY	ARRAY[0..1] OF BOOL	%UX0.1.112	通道别输出允许设定
_01_DA_OUTSTAT_ARY	ARRAY[0..1] OF WORD	%UW0.1.21	通道别输出状态设定
_01_DA_RANGE_ARY	ARRAY[0..1] OF BYTE	%UB0.1.24	通道别范围指定
_01_DA_RUN_ARY	ARRAY[0..1] OF BOOL	%UX0.1.168	通道别运行设定

内置模拟共通

变量	类型	内存分配	注释
_01_ERR	BOOL	%UX0.1.0	模块错误
_01_RDY	BOOL	%UX0.1.15	模块准备
_01_SETTINGERR	WORD	%UW0.1.27	设定错误信息

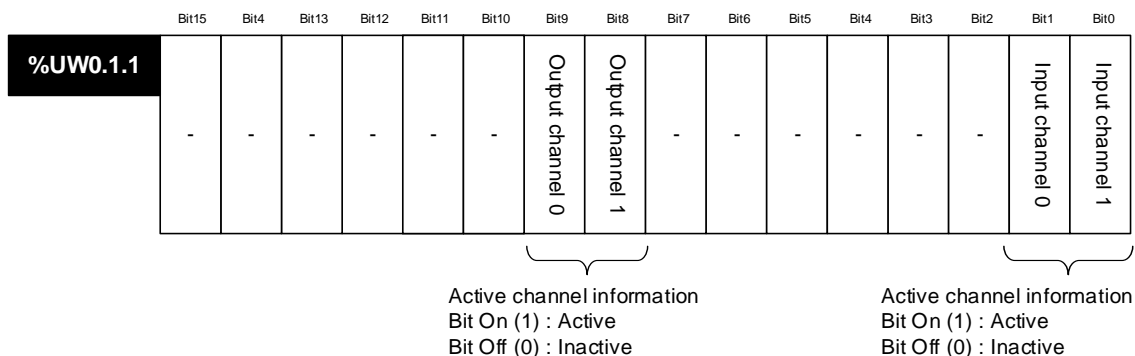
1) 内置模拟模块准备/错误标志位 (_01_RDY/_01_ERR)

- (1) %UX0.1.15：运动控制电源投入或者复位时模拟输入变换准备结束的时点上变为ON，模拟变换处理。
- (2) %UX0.1.0：内置模拟模块的错误状态出现的标志位。



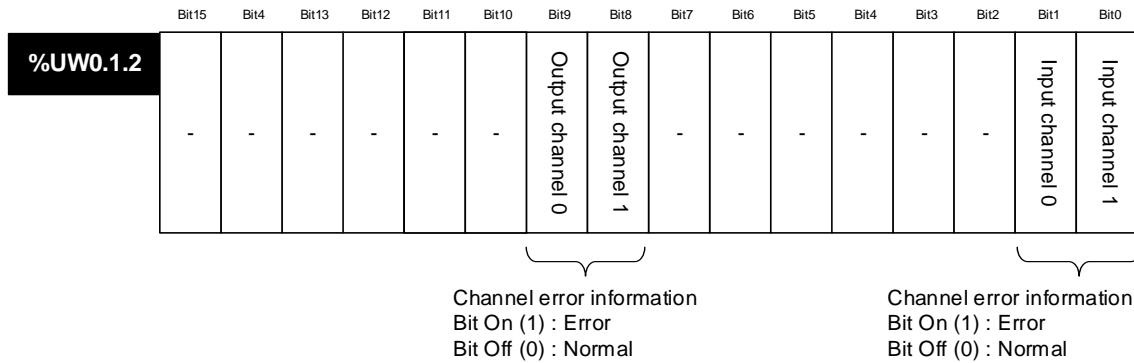
2) 运行通道信息

- (1) 通道别运行信息保存的领域
- (2) _01_AD0_ACT(%UX0.1.16)：输入通道0运行中
- _01_AD1_ACT(%UX0.1.17)：输入通道1运行中
- _01_DA0_ACT(%UX0.1.24)：输出通道0运行中
- _01_DA1_ACT(%UX0.1.25)：输出通道1运行中



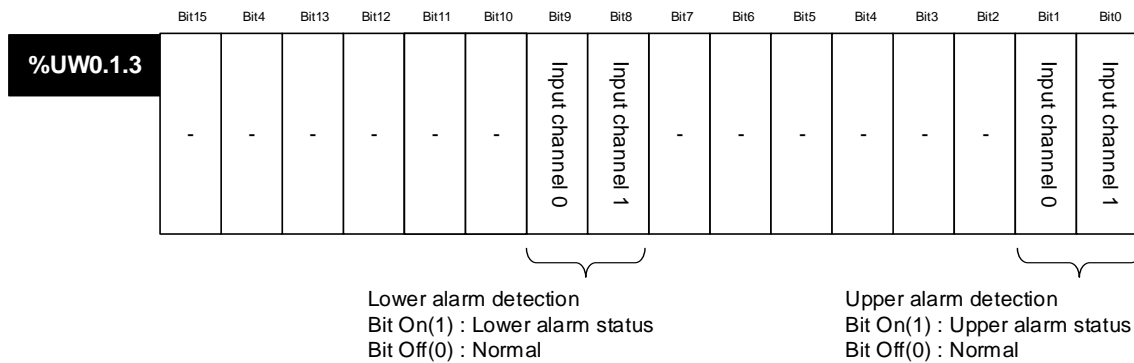
3) 错误通道信息

- (1) 通道别错误信息保存的领域。
- (2) `_01_AD0_ERR(%UX0.1.32)` : 输入通道0错误
- `_01_AD1_ERR(%UX0.1.33)` : 输入通道1错误
- `_01_DA0_ERR(%UX0.1.40)` : 输出通道0错误
- `_01_DA1_ERR(%UX0.1.41)` : 输出通道1错误



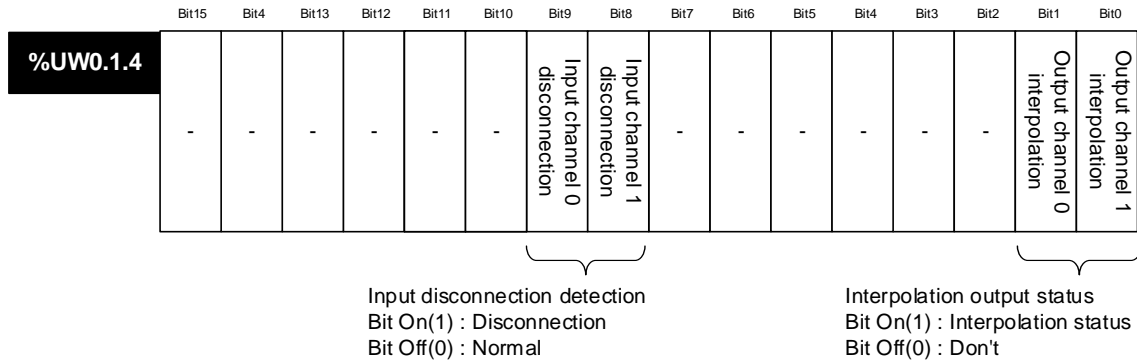
4) 输入警报上限/下限标志位

- (1) 通道别上限/下限警报检出信号保存的领域。
- (2) `_01_AD0_HOOR(%UX0.1.48)` : 输入通道0上限警报
- `_01_AD1_HOOR(%UX0.1.49)` : 输入通道1上限警报
- `_01_AD0_LOOR(%UX0.1.56)` : 输入通道0下限警报
- `_01_AD1_LOOR(%UX0.1.57)` : 输入通道1下限警报



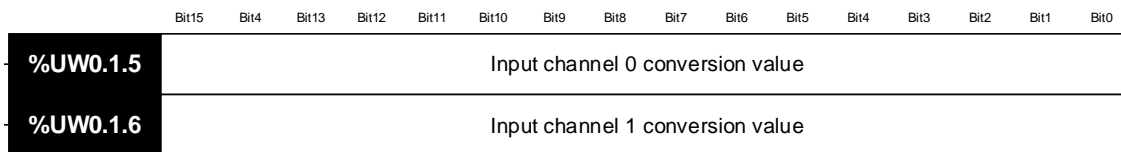
5) 断线输入/插补输出状态

- (1) 显示断线的输入通道和插补输出中的通道
- (2) `_01_DA0_INTP(%UX0.1.64)` : 输出通道0插补输出中
- `_01_DA1_INTP(%UX0.1.65)` : 输出通道1插补输出中
- `_01_AD0_IDD(%UX0.1.72)` : 输入通道0断线检出
- `_01_AD1_IDD(%UX0.1.73)` : 输入通道1断线检出



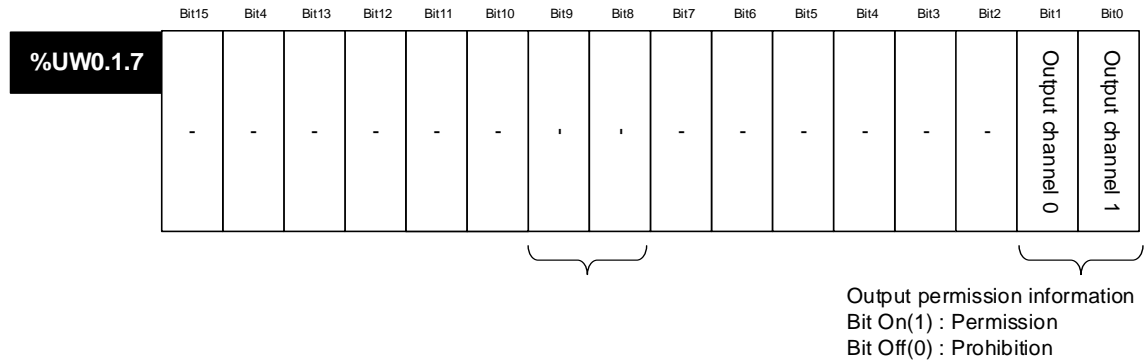
6) 数字输出值

- (1) A/D 变换的数字输出值在缓存内存地址%UW0.1.5 ~%UW0.1.6中通道别输出
- (2) 数字输出值是16bit的进制保存的
- (3) _01_AD0_数据(%UW0.1.5) : 输入通道0变换值
_01_AD1_数据(%UW0.1.6) : 输入通道1变换值



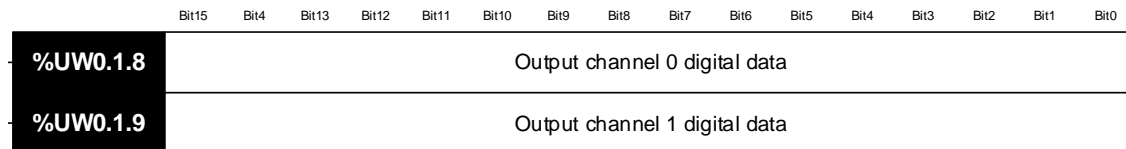
7) 输出允许设定

- (1) 输出允许/禁止每个通道都可设定。。
- (2) 输出允许不设定时，全通道输出禁止
- (3) `_01_DA0_OUTEN(%UX0.1.112)`：输出通道0输出允许
`_01_DA1_OUTEN(%UX0.1.113)`：输出通道1输出允许



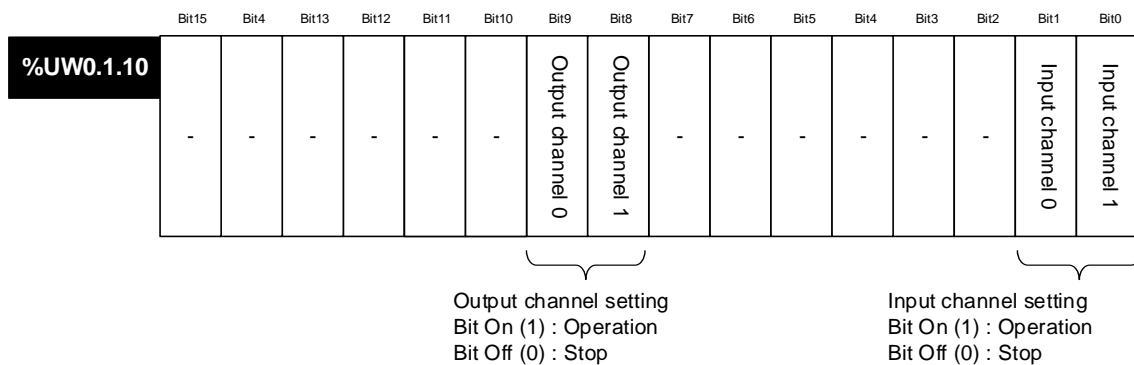
8) 输出数字输入值

- (1) 根据输入数据类型的设定，无符号值(-192~16,191 / 0~16,191)，有符号值(-8,192~8,191 / -8,000~8,191)，精确值(-952~5,047 / -60~5,059 / -120~10,119 / -10,240~10,239)，百分位值(-120~10,119 / 0~10,119)的范围内使用。
- (2) 数字输入值不设定时，按0处理
- (3) `_01_DA0_数据(%UW0.1.8)`：输出通道0数字输入数据
`_01_DA1_数据(%UW0.1.9)`：输出通道1数字输入数据



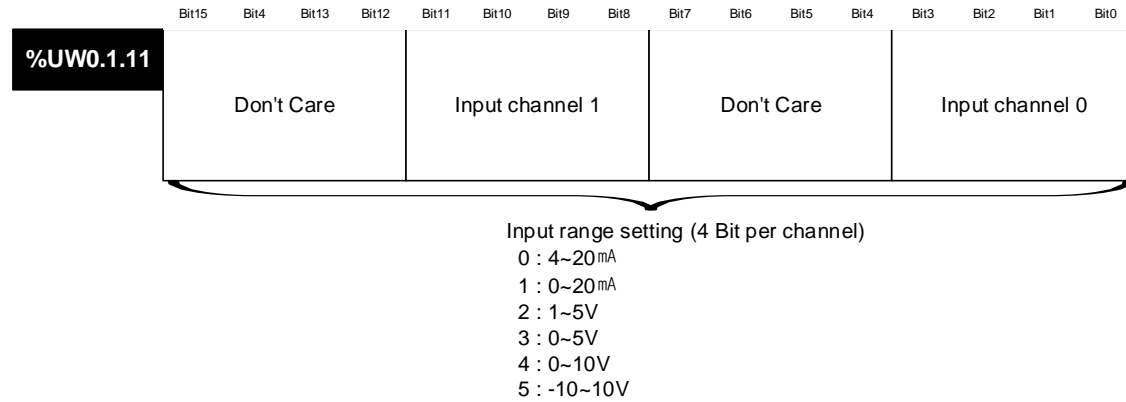
9) 运行通道设定

- (1) 运行通道不设定时，全通道停止。
- (2) `_01_AD0_RUN(%UX0.1.160)`：输入通道0 运行设定
`_01_AD1_RUN(%UX0.1.161)`：输入通道1 运行设定
`_01_DA0_RUN(%UX0.1.168)`：输出通道0 运行设定
`_01_DA1_RUN(%UX0.1.169)`：输出通道1 运行设定



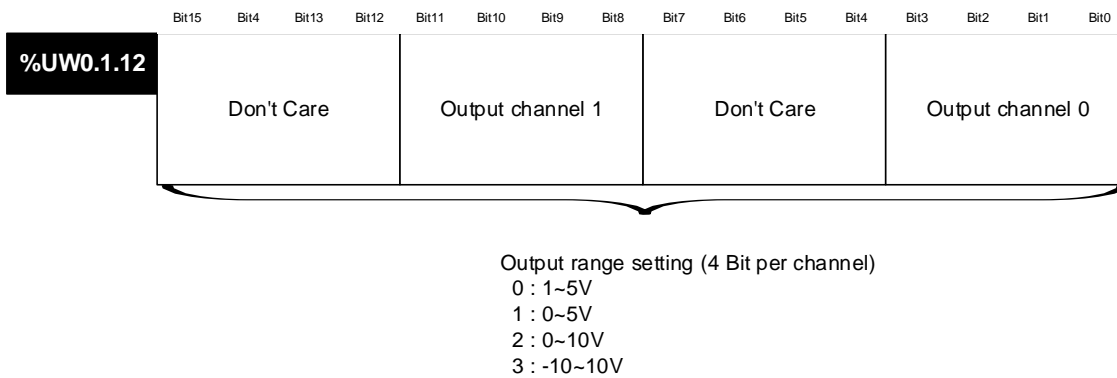
10) 输入范围设定

- (1) 模拟电压输入范围为DC 1~5V, DC 0~5V, DC 0~10V, DC -10~10V , 模拟电流 输入范围为DC 4~20mA, DC 0~20mA。
- (2) 输入范围不设定时, 按 DC 4~20mA 范围处理。.
- (3) `_01_AD0_RANGE(%UB0.1.22)` : 输入通道0 范围指定
`_01_AD1_RANGE(%UB0.1.23)` : 输入通道1 范围指定



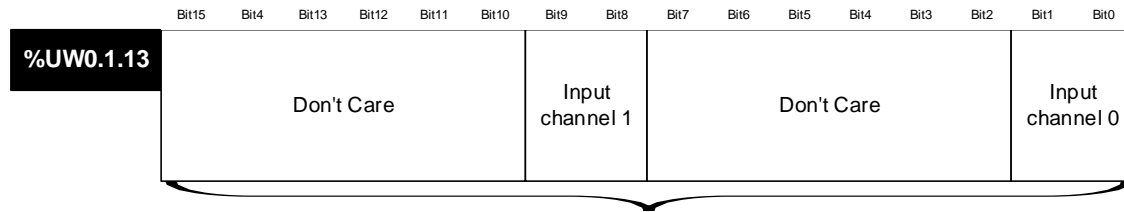
11) 输出范围设定

- (1) 模拟输出电压范围为DC 1~5V, DC 0~5V, DC 0~10V, DC -10~10V 。
- (2) 设定为输出范围设定值以外的范围时, 电压通道设定为1~5V 输出范围。
- (3) `_01_DA0_RANGE(%UB0.1.24)` : 输出通道0范围指定
`_01_DA1_RANGE(%UB0.1.25)` : 输出通道1范围指定



12) 内置模拟输入通道的输出数据类型设定

- (1) A/D 变换输出数据类型每个通道都可各自设定。
- (2) A/D 变换输出数据类型不设定时，全通道按 0~16,000范围处理。
- (3) `_01_AD0_`数据类型(%UB0.1.26)：输入通道0 A/D变换输出数据类型设定
`_01_AD1_`数据类型(%UB0.1.27)：输入通道1 A/D变换输出数据类型设定

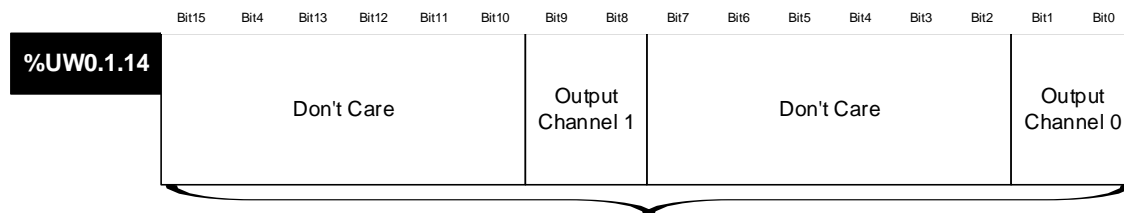


A/D conversion output data type setting (2 Bit per channel)

0 : 1~16,000	}	4~20 mA : 4,000~20,000
1 : -8,000~8,000		0~20 mA : 0~20,000
2 : Precise value		1~5V : 1,000~5,000
3 : 0~10,000		0~5V : 0~5,000
		0~10V : 0~10,000
		-10~10V : -10,000~10,000

13) 内置模拟输出通道的输入数据类型设定

- (1) D/A 变换输入数据类型每个通道都可各自设定。
- (2) D/A 变换输入数据类型不设定时，全通道按 0~16,000范围处理。
- (3) `_01_DA0_`数据类型(%UB0.1.28)：输出通道0 D/A 变换输入数据类型设定
`_01_DA1_`数据类型(%UB0.1.29)：输出通道1 D/A 变换输入数据类型设定



D/A conversion input data type setting (2 Bit per channel)

0 : 1~16,000	}	1~5V : 1,000~5,000
1 : -8,000~8,000		0~5V : 0~5,000
2 : Precise value		0~10V : 0~10,000
3 : 0~10,000		-10~10V : -10,000~10,000

第 13 章 内置模拟量功能

14) 输入滤波常数指定

- (1) 输入滤波常数指定为0，不滤波处理。
- (2) 输入滤波常数指定不设定时，按0处理不滤波处理。
- (3) `_01_AD0_FILCONST(%UW0.1.15)`：输入通道0滤波常数
`_01_AD1_FILCONST(%UW0.1.16)`：输入通道1滤波常数

	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
%UW0.1.15	Input channel 0 filter constant(0 or 64,000ms)															
%UW0.1.16	Input channel 1 filter constant(0 or 64,000ms)															

15) 平均处理方法设定

- (1) 平均处理设定时平均处理方法通过时间，计数，移动，加权平均可设定。
- (2) 平均处理方法设定不指定时全通道不平均处理。
- (3) `_01_AD0_AVGTYPE(%UB0.1.34)`：输入通道0 平均处理方法指定
`_01_AD1_AVGTYPE(%UB0.1.35)`：输入通道1 平均处理方法指定

	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
%UW0.1.17	Don't Care				Input channel 1				Don't Care				Input channel 0			

Average process method setting (4 Bit per channel)

- 0 : Sampling
- 1 : Time average
- 2 : Count average
- 3 : Moving average
- 4 : Weighted average

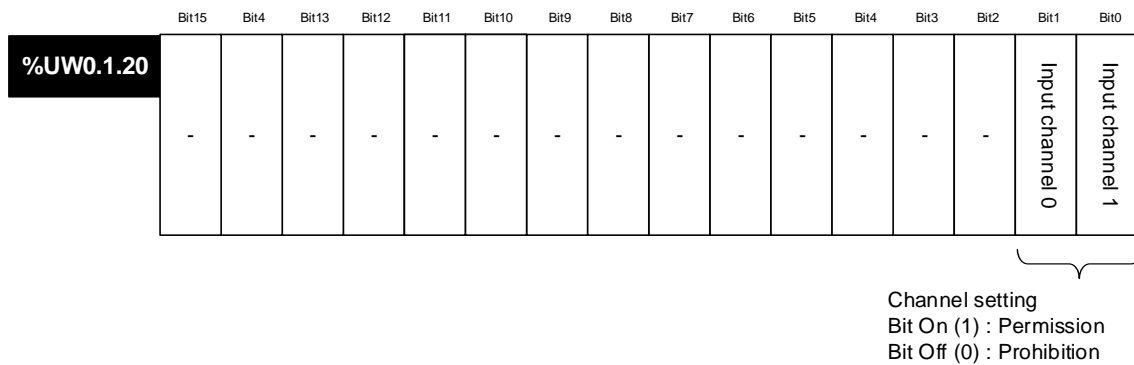
16) 平均值

- (1) 时间平均设定为4 ~ 16,000 范围值
- (2) 计数平均设定为2 ~ 64,000 范围值
- (3) 移动平均设定为2 ~ 100 范围值
- (4) 加权平均设定为1 ~ 99 范围值
- (5) 平均处理方法设定为0(采样)，时间/测试/移动/加权平均值设定为0，相应通道不平均处理，采样处理的模拟输入值输出
- (6) `_01_AD0_AVGVAL(%UW0.1.18)`：输入通道0平均值
`_01_AD1_AVGVAL(%UW0.1.19)`：输入通道1平均值

	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
%UW0.1.18	Input channel 0 average value															
%UW0.1.19	Input channel 1 average value															

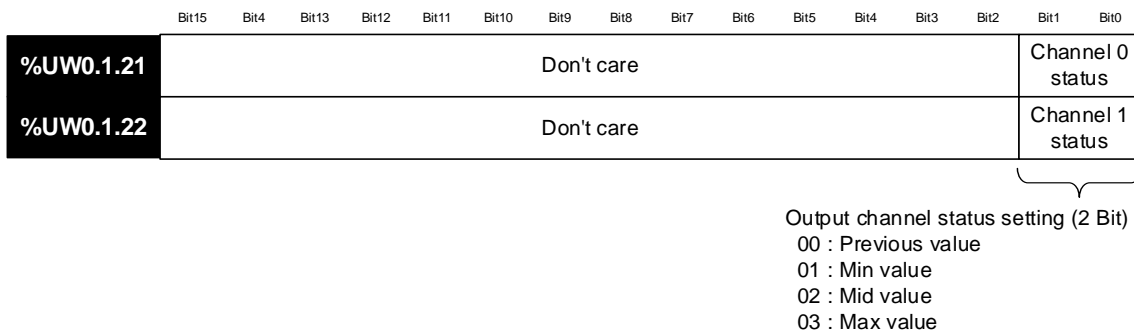
17) 有效变换值维持指定

- (1) 设定有效变换值维持功能，无效的输入值写入时，会维持最后的有效变换值。
 示例，4 ~ 20mA来运行的状态下，10mA的输入值写入信号逐渐减弱，立即变为3mA时，相应通道在10mA的相应输出值维持。
- (2) 设定此功能时，仅显示模拟输入使用范围内相应的数字输出值。
 模拟使用范围为"13.3 输入输出范围别变换特性"参考
- (3) 详细的使用方法"13.5.5 有效变换值维持功能"参考。
- (4) 有效变换值维持设定如下。
 _01_AD0_HOLDVAL(%UX0.1.320)：输入通道0有效变换值维持指定
 _01_AD1_HOLDVAL(%UX0.1.321)：输入通道1有效变换值维持指定



18) 输出状态设定

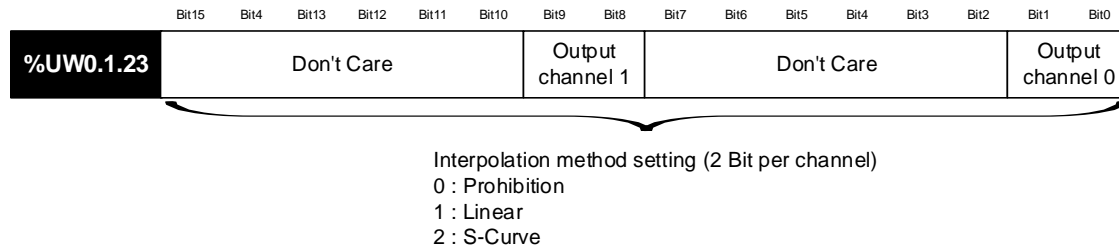
- (1) 运动控制为stop状态时，设定模拟输出状态。
- (2) 输出状态设定不指定时，输出之前值。



变量	内存	内容	设定
_01_DA0_OUTSTAT	%UW0.1.21	通道0输出状态设定	输入数据类型设定(bit) → 00 : 之前值 → 01 : 最小值 → 10 : 中间值 → 11 : 最大值
_01_DA1_OUTSTAT	%UW0.1.22	通道1输出状态设定	

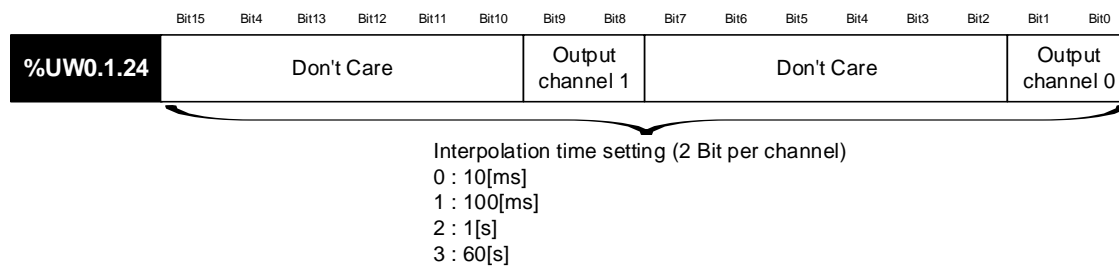
19) 插补方法设定

- (1) 显示各通道的插补方法的设定。
- (2) `_01_DA0_INTPMTHD(%UB0.1.46)` : 输出通道0 插补方法设定
`_01_DA1_INTPMTHD(%UB0.1.47)` : 输出通道1 插补方法设定



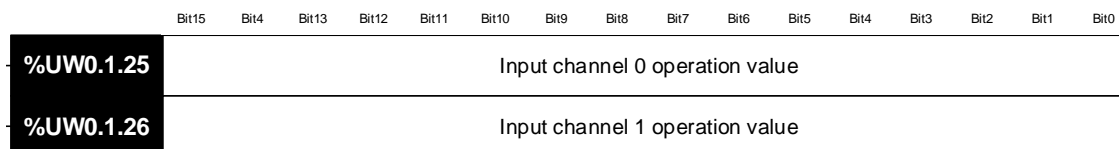
20) 插补时间设定

- (1) 显示各通道的插补时间的设定
- (2) `_01_DA0_INTPTIME(%UB0.1.48)` : 输出通道0 插补时间设定
`_01_DA1_INTPTIME(%UB0.1.49)` : 输出通道1 插补时间设定



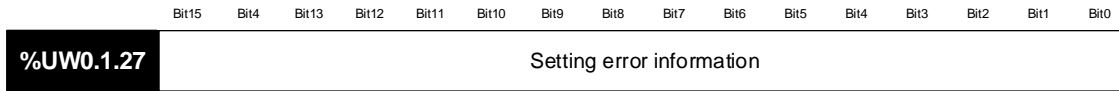
21) 插补演算值

- (1) 显示各通道的插补演算值。
- (2) `_01_DA0_INTPVAL(%UW0.1.25)` : 输出通道0 插补演算值
`_01_DA1_INTPVAL(%UW0.1.26)` : 输出通道1 插补演算值



22) 设定错误代码

- (1) 显示各通道的设定错误的错误代码
- (2) 正常情况下错误代码为0。
- (3) `_01_SETTINGERR(%UW0.1.27)`：设定错误信息

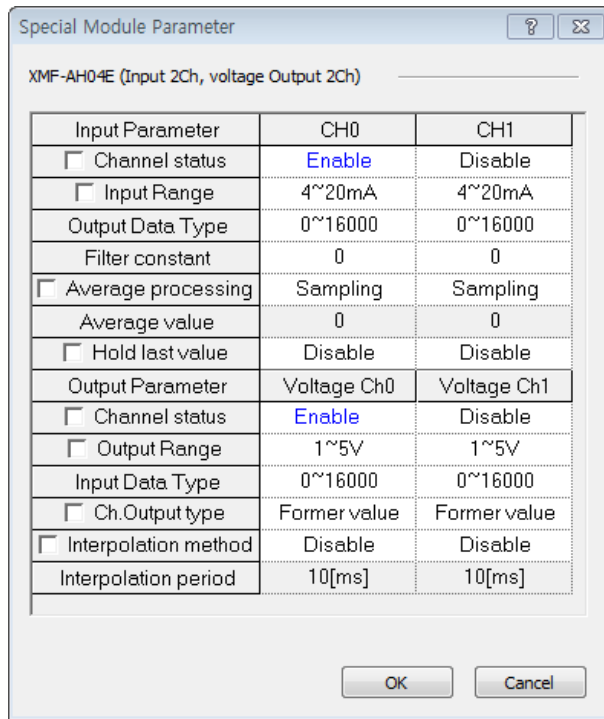
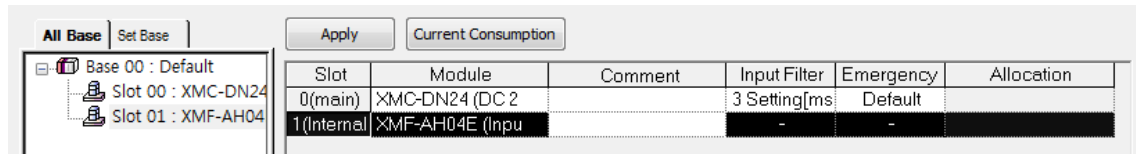


种类	错误代码 (10 进数)	LED 显示	说明	优先顺位	备注
错误	10#	AD LED 1s 闪烁	输入通道范围设定错误	1	#: 通道编号 (通道 0~1)
	20#		输入通道滤波值设定错误	2	
	30#		输入通道平均值设定错误	3	
	40#	DA LED 1s 闪烁	输出通道范围设定错误	4	
	50#		输出通道输入值设定错误	5	
	60#		插补方法设定范围错误	6	

- (4) 2个异常的错误发生时，错误代码优先顺位高的错误代码保存，系统错误代码 2 通道以上发生时，通道编号低的通道的错误代码优先保存。

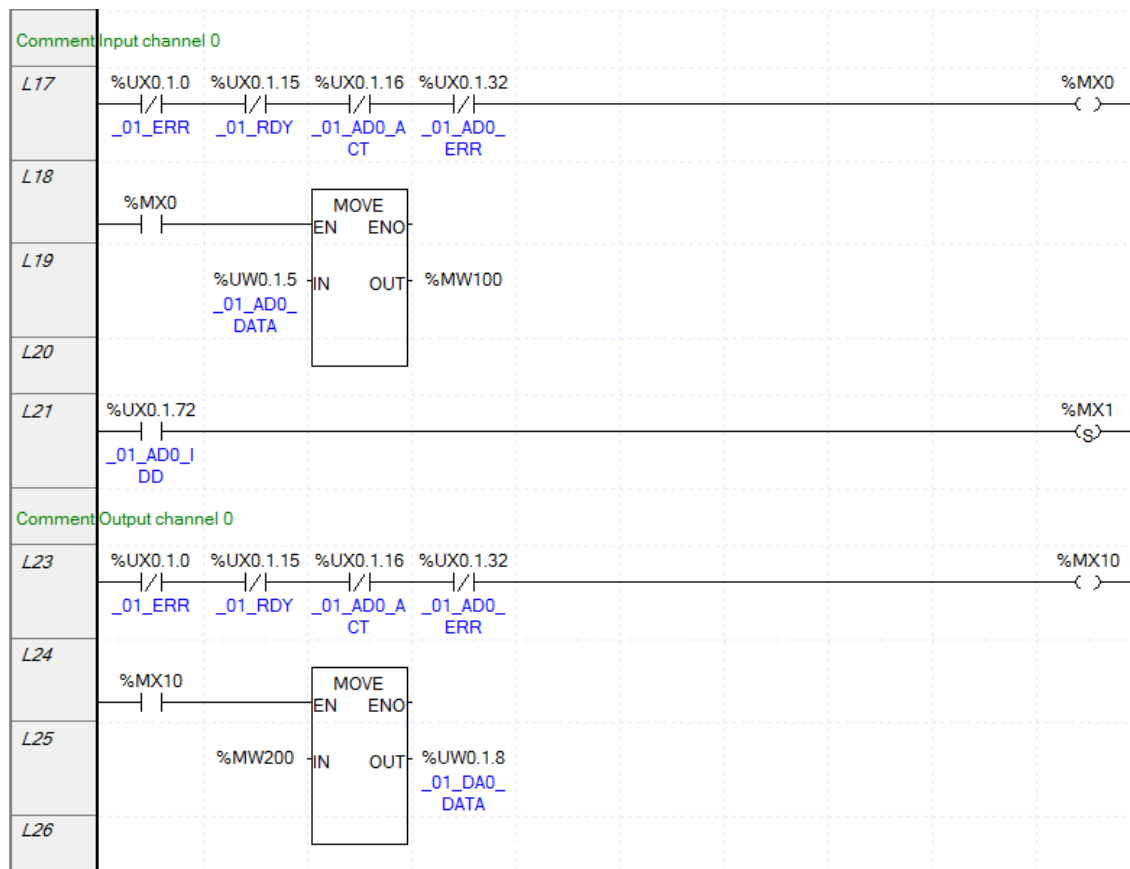
13.11 示例程序

1) I/O 参数设定窗



- (1) '输入参数'的'通道 0'设定为'运行'， '输入范围'设定为'4~20 mA'。
 '输出参数'的'电压通道 0' 设定为'运行'，'输出范围' 设定为'1~5V'。
- (2)

2) 程序示例



(1) 输入程序示例

- 1) 模块正常动作中，%MX0为ON。
 %UX0.1.0(模块错误) = OFF
 %UX0.1.15(模块准备) = ON
 %UX0.1.16(通道0运行中) = ON
 %UX0.1.32(通道0错误) = OFF
- 2) %MX0为ON，通道0 变换值(%UW0.1.5)移动为%MW100。
- 3) 通道0号断线错误发生时，%UX0.1.72(通道0 断线)为ON，%MX1bit 为ON。

(2) 输出程序示例

- 1) 模块正常动作中，%MX10为ON。
 %UX0.1.0(模块错误) = OFF
 %UX0.1.15(模块准备) = ON
 %UX0.1.24(电压输出通道0运行中) = ON
 %UX0.1.40(电压输出通道0错误) = OFF
- 2) %MX10为ON，电压输出通道0 输出允许设定(%UX0.1.112)为ON时，输出允许
- 3) %MX10为ON，%MW200的数据移动为电压输出通道0 输入值(%UW0.1.8)后输出。

13.12 故障排除

使用内置模拟模块时发生故障的诊断及处理方法的说明。

13.12.1 错误时LED状态区分

内置模拟模块有 2个LED，通过LED显示状态来确认模块的err有无

项目	正常状态	通道断线时	参数设定错误时
AD LED	灯亮	1s 周期闪烁	1s 周期闪烁 (输入参数设定错误时)
DA LED	灯亮	-	1s 周期闪烁 (输出参数设定错误时)
模块动作	正常动作 所有功能动作	所有功能动作输入 最低值显示	所有功能动作 (按参数基本值动作)
处理	-	配线确认	参数设定确认

13.12.2 内置模拟状态确认

使用MP500的系统监控，可确认内置模拟输入输出的状态。

1) 执行顺序

按照以下 项目中的方法执行。

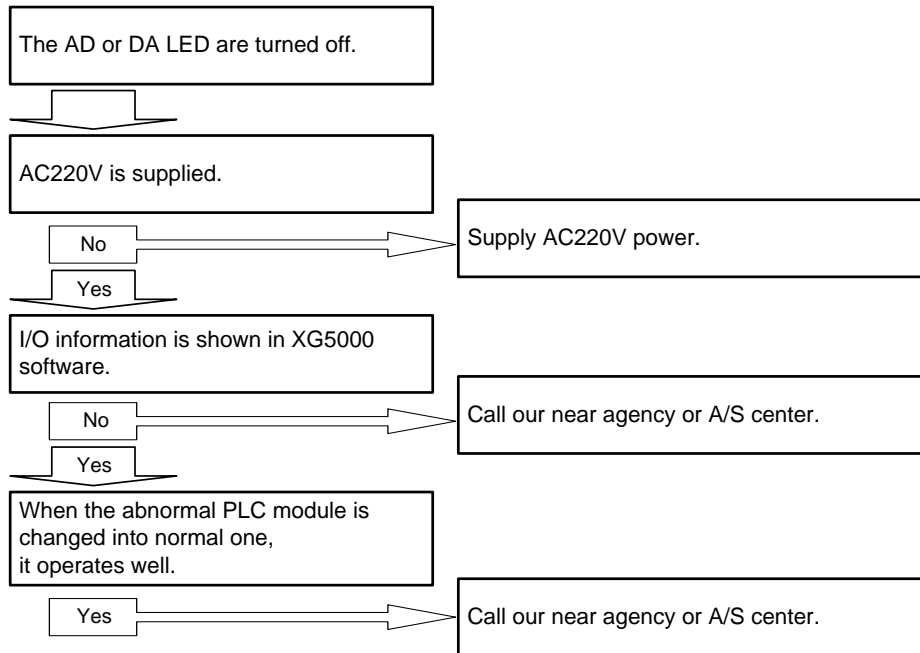
- (1) [监控] → [系统监控] → 模块图标右击 → [特殊模块信息]
- (2) [监控] → [系统监控] → 模块 图标双击
- (3) [监控] → [特殊模块监控] → 内置模拟模块选择 → 模块信息点击
- (4) [在线] → [诊断] → [I/O信息] → 内置模拟模块选择 -> 详细信息点击
- (5) [在线] → [诊断] → [I/O信息] -> 内置模拟模块双击

2) 模块信息

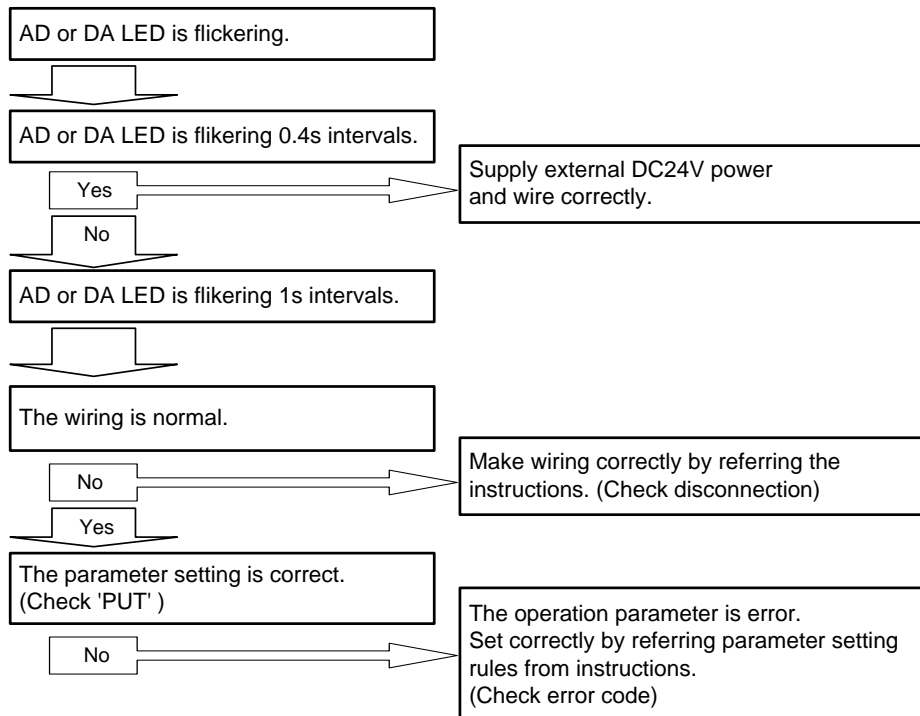
- (1) OS 版本: 模块的OS 版本信息显示.
- (2) OS 日期: 模块 OS的制作日期显示。
- (3) 模块状态: 模块的错误信息显示

13.12.3 故障诊断及处理方法

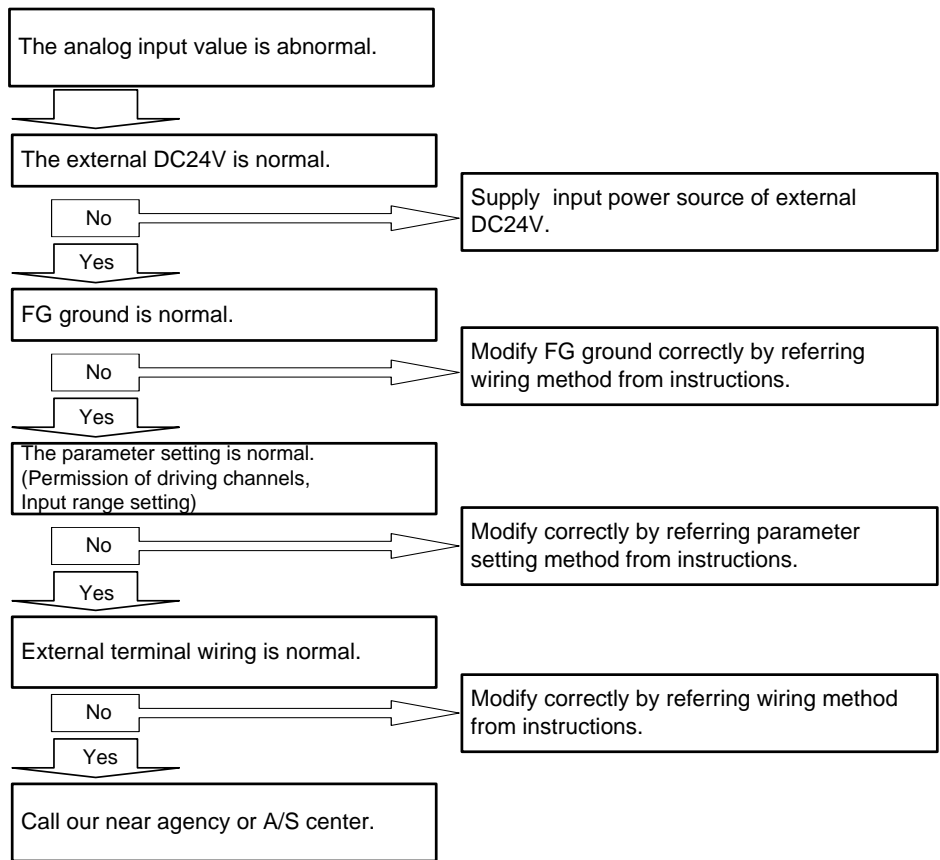
1) AD, DA LED 灯灭.



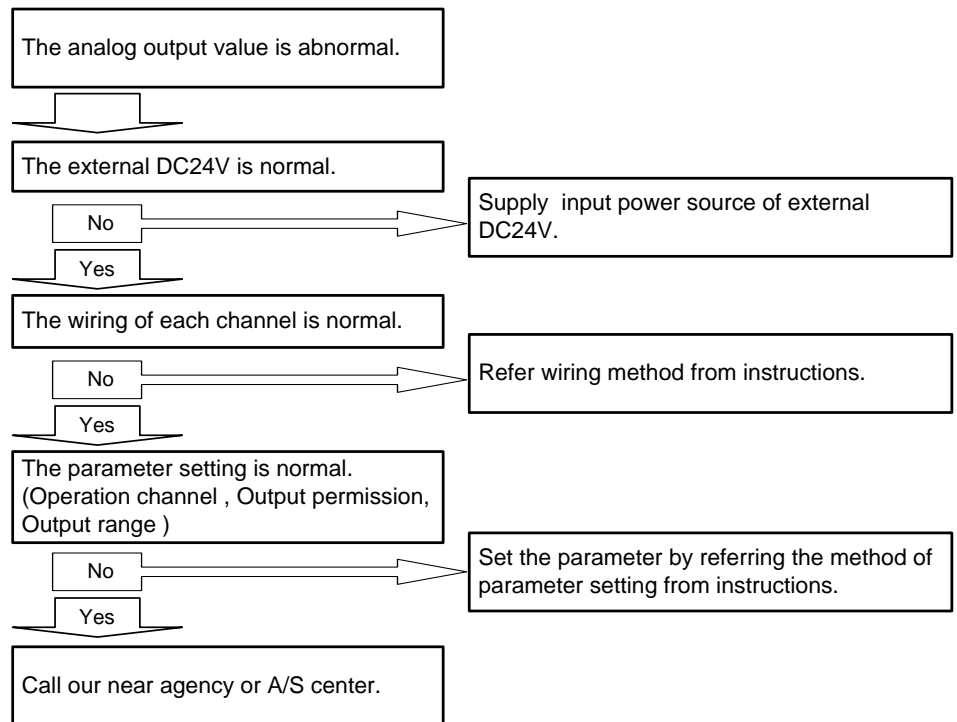
2) LED 闪烁.



3) 模拟输入值异常.



4) 模拟输出值异常.



附录 1 标志位列表

(1) 标志位类型

(a) 系统标志位

该标志位表示运行状态和运动控制器信息。

变量	类型	地址	描述
_SYS_STATE	DWORD	%FD0	PLC模式和状态
_RUN	BOOL	%FX0	RUN
_STOP	BOOL	%FX1	STOP
_ERROR	BOOL	%FX2	ERROR
_LOCAL_CON	BOOL	%FX4	局部控制
_REMOTE_CON	BOOL	%FX6	远程模式ON
_RUN_EDIT_ST	BOOL	%FX8	以在线编辑模式下载程序
_RUN_EDIT_CHK	BOOL	%FX9	内部处理在线编辑
_RUN_EDIT_DONE	BOOL	%FX10	在线编辑完成
_RUN_EDIT_NG	BOOL	%FX11	在线编辑异常终止
_CMOD_KEY	BOOL	%FX12	通过开关变更运行模式
_CMOD_LPADT	BOOL	%FX13	通过本地PADT变更运行模式
_FORCE_IN	BOOL	%FX16	强制输入
_FORCE_OUT	BOOL	%FX17	强制输出
_MON_ON	BOOL	%FX20	监控模式
_USTOP_ON	BOOL	%FX21	通过STOP功能进行STOP
_ESTOP_ON	BOOL	%FX22	通过ESTOP功能进行STOP
_INIT_RUN	BOOL	%FX24	执行初始任务
_PB1	BOOL	%FX28	程序代码1
_PB2	BOOL	%FX29	程序代码2
_CNF_ER	DWORD	%FD2	系统错误(有效位错误)
_ANNUM_ER	BOOL	%FX70	外部设备的有效位错误检测
_BPRM_ER	BOOL	%FX72	基本参数错误
_IOPRM_ER	BOOL	%FX73	IO配置参数错误
_SPPRM_ER	BOOL	%FX74	特殊模块参数错误
_CPPRM_ER	BOOL	%FX75	局部以太网参数错误
_PGM_ER	BOOL	%FX76	程序错误
_SWDT_ER	BOOL	%FX78	CPU异常结束
_ENCPRM_ER	BOOL	%FX85	编码器参数错误
_AXISPRM_ER	BOOL	%FX86	轴参数错误
_GROUPPRM_ER	BOOL	%FX87	轴组参数错误
_ECPRM_ER	BOOL	%FX88	EtherCAT参数错误

附录1 标志位列表

变量	类型	地址	描述
_NCPRM_ER	BOOL	%FX89	NC参数错误
_NCPGM_ER	BOOL	%FX90	NC程序检查错误
_PTASK_CYCLE_ER	BOOL	%FX91	主任务周期错误
_CTASK_CYCLE_ER	BOOL	%FX92	循环任务周期错误
_SYSTEM_ER	BOOL	%FX93	系统错误
_TASK_PRM_USAGE_OVER_ER	BOOL	%FX94	任务程序占有超出错误
_CNF_WAR	DWORD	%FD4	系统警告(一般错误)
_RTC_ER	BOOL	%FX128	异常RTC数据
_PTASK_CYCLE_WAR	BOOL	%FX129	主任务周期超出警告
_CTASK_CYCLE_WAR	BOOL	%FX130	循环任务周期超出警告
_AB_SD_ER	BOOL	%FX131	异常运行停止
_MOTION_CONTROL_WAR	BOOL	%FX132	运动控制异常警告
_ANNUM_WAR	BOOL	%FX134	外部设备一般错误检测
_TASK_PRM_USAGE_OVER_WAR	BOOL	%FX135	任务程序占有超出警告
_T20MS	BOOL	%FX192	20ms CLOCK
_T100MS	BOOL	%FX193	100ms CLOCK
_T200MS	BOOL	%FX194	200ms CLOCK
_T1S	BOOL	%FX195	1s CLOCK
_T2S	BOOL	%FX196	2s CLOCK
_T10S	BOOL	%FX197	10s CLOCK
_T20S	BOOL	%FX198	20s CLOCK
_T60S	BOOL	%FX199	60s CLOCK
_ON	BOOL	%FX201	始终ON
_OFF	BOOL	%FX202	始终OFF
_1ON	BOOL	%FX203	1扫描ON
_1OFF	BOOL	%FX204	1扫描OFF
_STOG	BOOL	%FX205	每个扫描切换
_ERR	BOOL	%FX224	计算错误标志位
_ALL_OFF	BOOL	%FX227	所有输出OFF
_LER	BOOL	%FX229	计算错误锁存标志位
_ARY_IDX_ERR	BOOL	%FX247	当使用阵列时索引范围超出错误
_ARY_IDX_LER	BOOL	%FX248	当使用阵列时索引范围超出错误锁存
_UDF_STACK_ERR	BOOL	%FX249	UDF堆栈超出错误标志位
_UDF_STACK_LER	BOOL	%FX250	UDF堆栈超出错误锁存标志位
_CPU_TYPE	WORD	%FW18	CPU类型
_CPU_VER	WORD	%FW19	CPU版本
_OS_VER	DWORD	%FD10	OS版本
_OS_DATE	DWORD	%FD11	OS日期
_OS_VER_PATCH	DWORD	%FD12	OS补丁版本

变量	类型	地址	描述
_RTC_TIME	ARRAY[0..7] OF BYTE	%FB52	RTC时间
_RTC_DATE	DATE	%FW30	当前RTC日期
_RTC_WEEK	UINT	%FW31	当前RTC天
_RTC_TOD	TIME_OF_DAY	%FD16	RTC当前时间(ms单位)
_KEY	DWORD	%FD17	本地钥匙开关当前状态
_AC_F_CNT	UINT	%FW36	短期电力中断计数
_FALS_NUM	UINT	%FW37	FALS指令使用范围
_SYS_ERR_TYPE	WORD	%FW38	系统错误详情标志位
_ENCODER_HW_ERR	BOOL	%FX608	编码器输入处理HW设置错误
_BACKPLANE_IF_ERR	BOOL	%FX609	背板接口错误
_SERIAL_NUM	ARRAY[0..19] OF BYTE	%FB80	序列号
_PTASK_SCAN_MAX	UINT	%FW512	主任务最大扫描时间(单位:100us)
_PTASK_SCAN_MIN	UINT	%FW513	主任务最小扫描时间(单位:100us)
_PTASK_SCAN_CUR	UINT	%FW514	主任务当前扫描时间(单位:100us)
_CTASK_SCAN_MAX	UINT	%FW515	循环任务最大扫描时间(单位:100us)
_CTASK_SCAN_MIN	UINT	%FW516	循环任务最小扫描时间(单位:100us)
_CTASK_SCAN_CUR	UINT	%FW517	循环任务当前扫描时间(单位:100us)
_PROGRAM_RATIO_MAX	UINT	%FW518	用户程序最大执行占有(1sec)
_PROGRAM_RATIO_MIN	UINT	%FW519	用户程序最小执行占有(1sec)
_PROGRAM_RATIO_CUR	UINT	%FW520	用户程序当前执行占有(1sec)
_PTASK_CYCLE_WAR_NUM	UINT	%FW748	主任务周期超出警告计数
_CTASK_CYCLE_WAR_NUM	UINT	%FW749	循环任务周期超出警告计数
_RTC_WR	BOOL	%FX20480	用户RTC设置请求
_CHK_ANC_ERR	BOOL	%FX20482	外部设备有效位错误检测请求
_CHK_ANC_WAR	BOOL	%FX20483	外部设备一般错误检测请求
_PTASK_SCAN_WR	BOOL	%FX20486	主任务扫描值初始化
_CTASK_SCAN_WR	BOOL	%FX20487	循环任务扫描值初始化
_INIT_DONE	BOOL	%FX20496	初始化任务完成
_ANC_ERR	WORD	%FW1282	外部设备有效位错误信息
_ANC_WAR	WORD	%FW1283	外部设备一般错误信息
_RTC_TIME_USER	ARRAY[0..7] OF BYTE	%FB2568	用户RTC时间

附录1 标志位列表

(b) 动作标志位

标志位显示如下. 显示运动控制器的状态和数据.

对应轴标志位显示为“_AXxx...”(xx 表示相应轴编号:十进制), 轴组标志位显示为“_AGyy...”(yy 表示轴组编号:十进制).

1) 动作常用标志位

变量	类型	地址	描述
_MC_RUN	BOOL	%FX65536	MC RUN
_MC_STOP	BOOL	%FX65537	MC STOP
_MC_TEST	BOOL	%FX65538	MC TEST
_MC_WARNING	BOOL	%FX65539	MC一般警告发生
_MC_ALARM	BOOL	%FX65540	MC一般警告发生
_MC_COM_ERR	BOOL	%FX65541	MC一般错误发生
_MC_COM_ERR_CODE	WORD	%FW4097	MC一般错误代码
_EC_LINKUP_INFO	BOOL	%FX65600	EtherCAT连接/断开信息
_EC_COMM	BOOL	%FX65601	EtherCAT通讯连接状态
_EC_COMM_ERR	BOOL	%FX65602	EtherCAT通讯超时错误
_EC_PDO_ERR_CNT	UINT	%FW4102	EtherCAT PDO错误计数
_EC_SLAVE_RDY	ARRAY[0..63] OF BOOL	%FX65664	EtherCAT从站准备
_EC_SDO_BUSY	ARRAY[0..63] OF BOOL	%FX65792	EtherCAT从站SDO处理繁忙
_EC_SDO_ERR	ARRAY[0..63] OF BOOL	%FX65920	EtherCAT从站SDO处理错误
_EC_LINE_FAIL	ARRAY[0..63] OF BOOL	%FX66048	EtherCAT电缆断开状态
_EC_MASTER_STATE	BYTE	%FB8264	EtherCAT主站STATE
_EC_SLAVE_NUM	WORD	%FW4133	连接EtherCAT从站数量
_EC_ERR_INFO1	STRING	%FB8272	EtherCAT错误信息1
_EC_ERR_INFO2	STRING	%FB8304	EtherCAT错误信息2
_EC_TRANSMITTED_OK	UDINT	%FD2084	EtherCAT帧发送数量
_EC_RECEIVED_OK	UDINT	%FD2085	EtherCAT帧接收数量
_EC_CRCERR_CNT	UDINT	%FD2086	EtherCAT接收CRC错误帧
_EC_COLLISION_CNT	UDINT	%FD2087	EtherCAT冲突帧数量
_EC_CARRIER_SENSE_ERR	UDINT	%FD2088	EtherCAT载波检测错误
_EC_LINKOFF_CNT	UDINT	%FD2089	EtherCAT连接断开数量
_EC_OVERSIZE_FRAME	UDINT	%FD2090	EtherCAT 接收超出尺寸帧
_EC_UNDERSIZE_FRAME	UDINT	%FD2091	EtherCAT 接收尺寸不足帧
_EC_JABBER_FRAME	UDINT	%FD2092	EtherCAT 接收超时传输帧
_EC_PDO_CUR_TRANS CYCLE	UDINT	%FD2093	EtherCAT PDO转换循环ns
_EC_PDO_MAX_TRANS CYCLE	UDINT	%FD2094	EtherCAT最大PDO转换循环ns
_EC_PDO_MIN_TRANS CYCLE	UDINT	%FD2095	EtherCAT最小PDO转换循环ns
_EC_PDO_TRANS_JITTER	UDINT	%FD2096	EtherCAT PDO帧转换误差ns

参考) _AXxx_HOME(用于原点返回指令的标志位)和_AXxx_Homing(PLC公开标准运行状态) 标志位表示相同状态.

2) 动作轴标志位

地址信息为轴01标志位内存. 每轴地址有2,048位(32LREAL)补偿.

变量	类型	地址	描述
_AXxx_RDY	BOOL	%FX73728	轴xx准备
_AXxx_WARNING	BOOL	%FX73729	轴xx警告发生
_AXxx_ALARM	BOOL	%FX73730	轴xx报警发生
_AXxx_SV_ON	BOOL	%FX73731	轴xx伺服On/Off
_AXxx_SV_RDY	BOOL	%FX73732	轴xx伺服准备
_AXxx_MSTSLV_STS	BOOL	%FX73733	轴xx主站/从站状态
_AXxx_NC	BOOL	%FX73734	轴xx NC运行
_AXxx_MST_INFO	UINT	%FW4609	轴xx主站轴信息
_AXxx_AXIS_TYPE	UINT	%FW4610	轴xx轴类型
_AXxx_LINKED_NODE	UINT	%FW4611	轴xx连接节点信息
_AXxx_LINKED_SLOT	UINT	%FW4612	轴xx连接槽信息
_AXxx_UNIT	UINT	%FW4613	轴xx轴单位
_AXxx_VEL_UNIT	UINT	%FW4614	轴xx速度单位
_AXxx_AX_ERR	WORD	%FW4615	轴xx错误代码
_AXxx_SVON_INCMPL	BOOL	%FX73856	轴xx伺服不完全
_AXxx_COMM_WARN	BOOL	%FX73857	轴xx通讯警告
_AXxx_DEV_WARN	BOOL	%FX73858	轴xx误差警告
_AXxx_SV_ERR	BOOL	%FX73872	轴xx伺服驱动错误
_AXxx_HW_POT	BOOL	%FX73873	轴xx正数限制检测
_AXxx_HW_NOT	BOOL	%FX73874	轴xx负数限制检测
_AXxx_SW_POT	BOOL	%FX73875	轴xx S/W正数限制检测
_AXxx_SW_NOT	BOOL	%FX73876	轴xx S/W负数限制检测
_AXxx_SV_OFF	BOOL	%FX73877	轴xx在伺服-off状态运行指令执行错误
_AXxx_POS_OVR	BOOL	%FX73878	轴xx超出定位传输数量设置范围
_AXxx_VEL_OVR	BOOL	%FX73879	轴xx超出最大速度
_AXxx_DEV_ERR	BOOL	%FX73880	轴xx偏差报警
_AXxx_HOME_INCMPL	BOOL	%FX73881	轴xx在未定HOME执行绝对定位指令
_AXxx_COMM_ERR	BOOL	%FX73882	轴xx通讯报警
_AXxx_BUSY	BOOL	%FX73888	轴xx运动指令繁忙状态
_AXxx_PAUSE	BOOL	%FX73889	轴xx动作指令暂停状态(速度为0)
_AXxx_STOP	BOOL	%FX73890	轴xx通过停止指令的停止状态
_AXxx_CMD_FAIL	BOOL	%FX73891	轴xx运动指令异常完成
_AXxx_CMD_CMPL	BOOL	%FX73892	轴xx运动指令正常完成

附录1 标志位列表

变量	类型	地址	描述
_AXxx_DIR	BOOL	%FX73893	轴xx运行方向
_AXxx_JOG	BOOL	%FX73894	轴xx JOG运行
_AXxx_HOME	BOOL	%FX73895	轴xx 原点返回运行
_AXxx_POS_CTRL	BOOL	%FX73896	轴xx 定位控制运行
_AXxx_VEL_CTRL	BOOL	%FX73897	轴xx 速度控制运行
_AXxx_TRQ_CTRL	BOOL	%FX73898	轴xx 转矩控制运行
_AXxx_LINTP	BOOL	%FX73899	轴xx 线性插值运行
_AXxx_CINTP	BOOL	%FX73900	轴xx 圆弧插值运行
_AXxx_SYNC	BOOL	%FX73901	轴xx 同步控制运行
_AXxx_COORD	BOOL	%FX73902	轴xx 坐标运行
_AXxx_POS_CMPL	BOOL	%FX73920	轴xx 定位完成
_AXxx_INPOS	BOOL	%FX73921	轴xx定位检测
_AXxx_LATCH_CMPL	BOOL	%FX73922	轴xx 锁存完成
_AXxx_HOME_CMPL	BOOL	%FX73923	轴xx 原点返回完成
_AXxx_Disabled	BOOL	%FX73936	轴xx 禁止状态
_AXxx_Standstill	BOOL	%FX73937	轴xx 停止状态
_AXxx_Discrete	BOOL	%FX73938	轴xx 分离状态
_AXxx_Continuous	BOOL	%FX73939	轴xx 持续状态
_AXxx_Synchronized	BOOL	%FX73940	轴xx 同步状态
_AXxx_Homing	BOOL	%FX73941	轴xx 原点返回状态
_AXxx_Stopping	BOOL	%FX73942	轴xx 停止状态
_AXxx_ErrorStop	BOOL	%FX73943	轴xx 错误停止状态
_AXxx_CMD_TPOS	LREAL	%FL1156	轴xx 目标位置
_AXxx_CMD_CPOS	LREAL	%FL1157	轴xx 当前扫描指令位置
_AXxx_CMD_VEL	LREAL	%FL1158	轴xx 指令速度
_AXxx_CMD_ACCDEC	LREAL	%FL1159	轴xx 指令加速/减速
_AXxx_CMD_JERK	LREAL	%FL1160	轴xx 指令加速度
_AXxx_CMD_TRQ	LREAL	%FL1161	轴xx 指令转矩
_AXxx_ACT_POS	LREAL	%FL1162	轴xx 实际当前位置
_AXxx_ACT_VEL	LREAL	%FL1163	轴xx 实际当前速度
_AXxx_ACT_TRQ	LREAL	%FL1164	轴xx 实际当前转矩
_AXxx_POS_DEV	LREAL	%FL1165	轴xx 位置偏差
_AXxx_DRV_ALARM	BOOL	%FX74624	轴xx 驱动报警状态
_AXxx_DRV_WARNING	BOOL	%FX74625	轴xx 驱动警告状态
_AXxx_DRV_SV_ON	BOOL	%FX74626	轴xx 伺服导通状态
_AXxx_DRV_POT	BOOL	%FX74627	轴xx 正数限制输入
_AXxx_DRV_NOT	BOOL	%FX74628	轴xx 负数限制输入
_AXxx_DRV_HOME	BOOL	%FX74629	轴xx 原点输入
_AXxx_DRV_LATCH1	BOOL	%FX74630	轴xx 锁存1输入
_AXxx_DRV_LATCH2	BOOL	%FX74631	轴xx 锁存2输入

变量	类型	地址	描述
_AXxx_DRV_PARAMBUSY	BOOL	%FX74632	SDO参数的轴xx读取/写入运行
_AXxx_DRV_IN	DWORD	%FD2333	轴xx驱动输入
_AXxx_DRV_ERR	WORD	%FW4668	轴xx驱动错误代码

参考) _AXxx_HOME(用于原点返回指令的标志位)和 _AXxx_Homing(PLC公开标准运行状态) 标志位表示相同状态。

3) 动作轴组标志位

地址信息为轴01标志位内存. 每轴地址有5,120位(80LREAL)补偿.

变量	类型	地址	描述
_AGxx_RDY	BOOL	%FX212992	轴组xx准备
_AGxx_WARNING	BOOL	%FX212993	轴组xx警告发生
_AGxx_ALARM	BOOL	%FX212994	轴组xx报警发生
_AGxx_SV_ON	BOOL	%FX212995	轴组xx伺服On/Off
_AGxx_SV_RDY	BOOL	%FX212996	轴组xx伺服准备
_AGxx_ERR	WORD	%FW13313	轴组xx错误代码
_AGxx_BUSY	BOOL	%FX213024	轴组xx运动指令繁忙状态
_AGxx_PAUSE	BOOL	%FX213025	轴组xx运动指令暂停状态(速度为0)
_AGxx_STOP	BOOL	%FX213026	轴组xx通过停止指令的停止状态
_AGxx_CMD_FAIL	BOOL	%FX213027	轴组xx指令错误退出状态
_AGxx_CMD_CMPL	BOOL	%FX213028	轴组xx指令执行完成
_AGxx_LINTP	BOOL	%FX213029	轴组xx线性插值运行
_AGxx_CINTP	BOOL	%FX213030	轴组xx圆弧插值运行
_AGxx_HOME	BOOL	%FX213031	轴组xx原点返回运行
_AGxx_SYNC	BOOL	%FX213032	轴组xx同步运行
_AGxx_TLINTP	BOOL	%FX213033	轴组xx坐标时间运行
_AGxx_CDMOVE	BOOL	%FX213034	轴组xx坐标直接运行
_AGxx_CCINTP	BOOL	%FX213035	轴组xx坐标圆弧插值运行
_AGxx_POS_CMPL	BOOL	%FX213056	轴组xx定位完成
_AGxx_Disabled	BOOL	%FX213072	轴组xx禁止状态
_AGxx_Standby	BOOL	%FX213073	轴组xx准备状态
_AGxx_Moving	BOOL	%FX213074	轴组xx移动状态
_AGxx_Homing	BOOL	%FX213075	轴组xx原点返回状态
_AGxx_Stopping	BOOL	%FX213076	轴组xx停止状态
_AGxx_ErrorStop	BOOL	%FX213077	轴组xx错误停止状态
_AGxx_CMD_TPOS	ARRAY[0..9] OF LREAL	%FL3330	轴组xx目标位置
_AGxx_CMD_CPOS	ARRAY[0..9] OF LREAL	%FL3340	轴组xx当前扫描指令位置
_AGxx_CMD_VEL	LREAL	%FL3350	轴组xx目标速度

附录1 标志位列表

变量	类型	地址	描述
_AGxx_CMD_ACCDEC	LREAL	%FL3351	轴组xx指令加速/减速
_AGxx_CMD_JERK	LREAL	%FL3352	轴组xx指令加速度
_AGxx_ACT_POS	ARRAY[0..9] OF LREAL	%FL3353	轴组xx实际当前位置
_AGxx_ACT_VEL	LREAL	%FL3363	轴组xx实际当前速度
_AGxx_CFG_AX_NUM	UINT	%FW13456	轴组xx轴数量
_AGxx_CFG_A1	UINT	%FW13458	组成轴1轴组xx轴编号
_AGxx_CFG_A2	UINT	%FW13459	组成轴2轴组xx轴编号
_AGxx_CFG_A3	UINT	%FW13460	组成轴3轴组xx轴编号
_AGxx_CFG_A4	UINT	%FW13461	组成轴4轴组xx轴编号
_AGxx_CFG_A5	UINT	%FW13462	组成轴5轴组xx轴编号
_AGxx_CFG_A6	UINT	%FW13463	组成轴6轴组xx轴编号
_AGxx_CFG_A7	UINT	%FW13464	组成轴7轴组xx轴编号
_AGxx_CFG_A8	UINT	%FW13465	组成轴8轴组xx轴编号
_AGxx_CFG_A9	UINT	%FW13466	组成轴9轴组xx轴编号
_AGxx_CFG_A10	UINT	%FW13467	组成轴10轴组xx轴编号
_AGxx_MTCP_Px	LREAL	%FL3367	轴组xx X轴位置(MCS)
_AGxx_MTCP_Py	LREAL	%FL3368	轴组xx Y轴位置(MCS)
_AGxx_MTCP_Pz	LREAL	%FL3369	轴组xx Z轴位置(MCS)
_AGxx_MTCP_A	LREAL	%FL3370	轴组xx X轴旋转(MCS)
_AGxx_MTCP_B	LREAL	%FL3371	轴组xx Y轴旋转(MCS)
_AGxx_MTCP_C	LREAL	%FL3372	轴组xx Z轴旋转(MCS)
_AGxx_PTCP_Px	LREAL	%FL3373	轴组xx X轴位置(PCS)
_AGxx_PTCP_Py	LREAL	%FL3374	轴组xx Y轴位置(PCS)
_AGxx_PTCP_Pz	LREAL	%FL3375	轴组xx Z轴位置(PCS)
_AGxx_PTCP_A	LREAL	%FL3376	轴组xx X轴旋转(PCS)
_AGxx_PTCP_B	LREAL	%FL3377	轴组xx Y轴旋转(PCS)
_AGxx_PTCP_C	LREAL	%FL3378	轴组xx Z轴旋转(PCS)

4) Slave flag

变量	类型	地址	描述
_SLVxx_EC_STATE	SINT	%FB47104	EtherCAT从站xx状态
_SLVxx_LINK_STATUS	BYTE	%FB47105	EtherCAT从站xx连接信息
_SLVxx_ERROR	WORD	%FW23553	EtherCAT从站xx错误
_SLVxx_VENDOR_ID	DWORD	%FD11777	EtherCAT从站xx供应商ID
_SLVxx_PRODUCT_CODE	DWORD	%FD11778	EtherCAT从站xx产品代码
_SLVxx_REVISION_NUMBER	DWORD	%FD11779	EtherCAT从站xx版本号

5) NC通道标志位

显示NC通道状态. NC通道标志位显示为“_NCyy...”

(yy表示NC通道编号(十进制))

变量	类型	地址	描述
_NCyy_Ready	BOOL	%FX524288	NC Ch. yy NC准备
_NCyy_Warning	BOOL	%FX524289	NC Ch. yy警告发生
_NCyy_Alarm	BOOL	%FX524290	NC Ch. yy报警发生
_NCyy_ResetStatus	BOOL	%FX524291	NC Ch. yy复位状态
_NCyy_CycStartBegin	BOOL	%FX524292	NC Ch. yy循环启动开始信息
_NCyy_CycStartFinish	BOOL	%FX524293	NC Ch. yy循环启动完成信息
_NCyy_TargetQtyCmpl	BOOL	%FX524294	NC Ch. yy目标数量到达信号
_NCyy_PrgmNormalCmpl	BOOL	%FX524295	NC Ch. yy程序执行正常完成
_NCyy_PwrFailInAuto	BOOL	%FX524296	NC Ch. yy自动运行中电源故障
_NCyy_ErrorCode	WORD	%FW32770	NC Ch. yy 错误代码
_NCyy_IPR_HeartBeat	UDINT	%FD16386	NC Ch. yy IPR心跳
_NCyy_IPR_Run	BOOL	%FX524384	NC Ch. yy IPR 运行状态(0:停止, 1:运行)
_NCyy_IPR_WaitEoM	BOOL	%FX524400	NC Ch. yy运动状态等待结束(0:不等待, 1:等待)
_NCyy_IPR_EndOfMot	UINT	%FW32776	NC Ch. yy动作结束
_NCyy_IPR_AfBufSts	UINT	%FW32777	NC Ch. yy自动FIFO缓存状态(0:空, 其他: 缓存使用)
_NCyy_IPR_ErrorCode	UINT	%FW32778	NC Ch. yy IPR错误代码
_NCyy_PA_ErrorCode	UINT	%FW32779	NC Ch.yy程序进程错误代码
_NCyy_IPR_AlarmSts	ARRAY[0..4] OF DWORD	%FD16390	NC Ch. yy IPR报警信息
_NCyy_CycleStart	BOOL	%FX524672	NC Ch.yy循环启动状态
_NCyy_FeedHold	BOOL	%FX524673	NC Ch.yy馈送保持状态
_NCyy_AutoOperation	BOOL	%FX524674	NC Ch.yy自动运行状态
_NCyy_RapidTrvsOpr	BOOL	%FX524736	NC Ch. yy快速移位状态
_NCyy_CuttingFeedOpr	BOOL	%FX524737	NC Ch. yy进刀运行
_NCyy_TargetVelocity	LREAL	%FL8200	NC Ch. yy目标速度(F指令值)
_NCyy_CmdVelocity	LREAL	%FL8201	NC Ch. yy指令速度y
_NCyy_TVelOfSpindle	LREAL	%FL8203	NC Ch. yy主轴目标速度(S指令值)
_NCyy_CVelOfSpindle	LREAL	%FL8204	NC Ch. yy主轴目标速度
_NCyy_FeedOverride	LREAL	%FL8206	NC Ch. yy馈送覆盖
_NCyy_RapidOverride	LREAL	%FL8207	NC Ch. yy移位覆盖

附录1 标志位列表

变量	类型	地址	描述
_NCyy_SpindleOverride	LREAL	%FL8208	NC Ch. yy 主轴覆盖
_NCyy_SpindleStop	BOOL	%FX525376	NC Ch. yy 主轴停止状态
_NCyy_SpindleCW	BOOL	%FX525377	NC Ch. yy 主轴CW运行
_NCyy_SpindleCCW	BOOL	%FX525378	NC Ch. yy 主轴CCW运行
_NCyy_SpindleCVelAgr	BOOL	%FX525380	NC Ch. yy 主轴指令速度到达信号
_NCyy_SpindleZeroVel	BOOL	%FX525381	NC Ch. yy 主轴0速度到达信号
_NCyy_DwellCount	UDINT	%FD16422	NC Ch. yy 停留计数
_NCyy_ErrorBlockNum	UDINT	%FD16423	NC Ch. yy 错误块编号
_NCyy_BlockCmdType	UINT	%FW32848	NC Ch. yy 当前块指令类型
_NCyy_CurrentToolNum	UINT	%FW32856	NC Ch. yy 当前工具编号
_NCyy_ToolRadiusComp	UINT	%FW32857	NC Ch. yy 当前工具半径补偿的补偿数量
_NCyy_ToolLengthComp	UINT	%FW32858	NC Ch. yy 当前工具长度补偿的补偿数量
_NCyy_McodeStrobe	BOOL	%FX526080	NC Ch. yy M代码输出选通信号
_NCyy_McodeDistCmpl	BOOL	%FX526081	NC Ch. yy M代码分配完成信号
_NCyy_McodeM00	BOOL	%FX526082	NC Ch. yy 特殊M代码输出信号(M00)
_NCyy_McodeM01	BOOL	%FX526083	NC Ch. yy 特殊M代码输出信号(M01)
_NCyy_McodeM02	BOOL	%FX526084	NC Ch. yy 特殊M代码输出信号(M02)
_NCyy_McodeM30	BOOL	%FX526085	NC Ch. yy 特殊M代码输出信号(M30)
_NCyy_McodeData	UDINT	%FD16441	NC Ch. yy M代码数据输出
_NCyy_ScodeStrobe	BOOL	%FX526144	NC Ch. yy S代码输出选通信号
_NCyy_ScodeDistCmpl	BOOL	%FX526145	NC Ch. yy S代码分配完成信号
_NCyy_ScodeData	UDINT	%FD16443	NC Ch. yy S代码数据输出
_NCyy_TcodeStrobe	BOOL	%FX526208	NC Ch. yy T代码输出选通信号
_NCyy_TcodeDistCmpl	BOOL	%FX526209	NC Ch. yy T代码分配完成信号
_NCyy_TcodeData	UDINT	%FD16445	NC Ch. yy T代码数据输出
_NCyy_CycleTime	REAL	%FD16446	NC Ch. yy 机械循环时间
_NCyy_TotalRunTime	REAL	%FD16447	NC Ch. yy 总机械循环时间
_NCyy_PartCount	UDINT	%FD16448	NC Ch. yy 机械数量
_NCyy_PartCountByM99	UDINT	%FD16449	NC Ch. yy 在重复机械状态的M99机械数量
_NCyy_MainProgram	STRING	%FB65800	NC Ch. yy 主程序名称
_NCyy_CurrentProgram	STRING	%FB65832	NC Ch. yy 当前运行程序名称
_NCyy_MainBlkNum	UDINT	%FD16466	NC Ch. yy 主程序块编号
_NCyy_CurrentBlkNum	UDINT	%FD16468	NC Ch. yy 当前运行程序块编号

变量	类型	地址	描述
_NCyy_ModalG_OneShot	REAL	%FD16476	NC Ch. yy G代码模块值组0 –单触发
_NCyy_ModalG_Motion	REAL	%FD16477	NC Ch. yy G代码模块值组1 –动作
_NCyy_ModalG_CmdMode	REAL	%FD16479	NC Ch. yy G代码模块值组3 –指令模式(ABS或者INC)
_NCyy_ModalG_Feed	REAL	%FD16481	NC Ch. yy G代码模块值组5 –馈送模式
_NCyy_ModalG_Unit	REAL	%FD16482	NC Ch. yy G代码模块值组6 –单位
_NCyy_ModalG_TRComp	REAL	%FD16483	NC Ch. yy G代码模块值组7 –工具半径补偿
_NCyy_ModalG_Stroke	REAL	%FD16485	NC Ch. yy G代码模块值组9 –行程查看
_NCyy_ModalG_TLComp	REAL	%FD16489	NC Ch. yy G代码模块值组13 –工具长度补偿
_NCyy_ModalG_WpCoord	REAL	%FD16490	NC Ch. yy G代码模块值组14 –工件坐标系
_NCyy_ModalG_Plane	REAL	%FD16492	NC Ch. yy G代码模块值组16 –圆形平面
_NCyy_ModalG_RPolar	REAL	%FD16496	NC Ch. yy G代码模块值组20 –反向极坐标插值
_NCyy_ModalG_CylIntp	REAL	%FD16498	NC Ch. yy G代码模块值组22 –圆柱插值
_NCyy_ModalFeed	LREAL	%FL8254	NC Ch. yy 模式馈送
_NCyy_ModalScode	UDINT	%FD16510	NC Ch. yy 模式S代码
_NCyy_ModalSpindleM	UDINT	%FD16511	NC Ch. yy 模式主轴M代码
_NCyy_ModelMcode	UDINT	%FD16512	NC Ch. yy 模式M代码
_NCyy_ModelHcode	UDINT	%FD16513	NC Ch. yy 模式H代码
_NCyy_ModalWorkCoord	UDINT	%FD16514	NC Ch. yy 模式工件坐标

6) NC通道/轴标志位

显示NC通道上配置轴状态. NC通道/轴标志位显示为“_NCyy_X...”, “NCyy_Y...”

(yy表示NC通道编号(十进制)和X,Y,Z,A,B,C,U,V,W配置轴)

变量	类型	地址	描述
_NC01X_Ready	BOOL	%FX532480	NC Ch. 01轴X准备
_NC01X_Warning	BOOL	%FX532481	NC Ch. 01轴X警告发生
_NC01X_Alarm	BOOL	%FX532482	NC Ch. 01轴X报警发生
_NC01X_ServoOn	BOOL	%FX532483	NC Ch. 01轴X伺服On/Off
_NC01X_ServoReady	BOOL	%FX532484	NC Ch. 01轴X伺服准备
_NC01X_ServoAlarm	BOOL	%FX532485	NC Ch. 01轴X伺服报警发生
_NC01X_OprRdy	BOOL	%FX532544	NC Ch. 01轴X运行准备
_NC01X_FeedMode	BOOL	%FX532552	NC Ch. 01轴X轴馈送模式(0: 线性轴, 1:旋转轴)
_NC01X_LinkedAxNum	UINT	%FW33285	NC Ch. 01轴X IPR轴的实际轴编号
_NC01X_Busy	BOOL	%FX532608	NC Ch. 01轴X繁忙状态

附录1 标志位列表

变量	类型	地址	描述
_NC01X_Direction	BOOL	%FX532609	NC Ch. 01轴X运行反向
_NC01X_ForwardRun	BOOL	%FX532610	NC Ch. 01轴X运行到正方向
_NC01X_ReverseRun	BOOL	%FX532611	NC Ch. 01轴X运行到负方向
_NC01X_RapidTraverse	BOOL	%FX532612	NC Ch. 01轴X快速移位运行
_NC01X_CuttingFeed	BOOL	%FX532613	NC Ch. 01轴X进刀运行
_NC01X_Homing	BOOL	%FX532614	NC Ch. 01轴X原点返回运行
_NC01X_PosCmpl	BOOL	%FX532672	NC Ch. 01轴X定位完成
_NC01X_Inposition	BOOL	%FX532673	NC Ch. 01轴X达到位置检测
_NC01X_HomeCmpl	BOOL	%FX532675	NC Ch. 01轴X原点返回完成
_NC01X_CmdPosInWC	LREAL	%FL8325	NC Ch. 01轴X在工件坐标系中指令位置
_NC01X_CmdPosInRC	LREAL	%FL8326	NC Ch. 01轴X在相对坐标系中指令位置
_NC01X_ActualVel	LREAL	%FL8327	NC Ch. 01轴X实际当前速度
_NC01X_RemDistance	LREAL	%FL8329	NC Ch. 01轴X剩余距离
_NC01X_PosDeviation	LREAL	%FL8330	NC Ch. 01轴X伺服位置偏差(跟踪误差)
_NC01X_WcOffset	LREAL	%FL8334	NC Ch. 01轴X工件坐标系补偿值
_NC01X_WcBasicOffset	LREAL	%FL8335	NC Ch. 01轴X工件坐标系基本补偿值
_NC01X_WcShiftOffset	LREAL	%FL8336	NC Ch. 01轴X工件坐标系移动补偿值
_NC01X_LocalWcOffset	LREAL	%FL8337	NC Ch. 01轴X本地工件坐标系补偿值
_NC01X_CmdPosInMC	LREAL	%FL8339	NC Ch. 01轴X机械坐标系指令位置
_NC01X_ActualPosInMC	LREAL	%FL8341	NC Ch. 01轴X机械坐标系实际当前位置
_NC01X_AxErr	WORD	%FW33372	NC Ch. 01轴X错误代码
_NC01X_DrvErr	WORD	%FW33373	NC Ch. 01轴X驱动错误代码

7) SD memory flag

变量	类型	地址	描述
_SD_Attach	BOOL	%KX8256	SD附件状态
_SD_Rdy	BOOL	%KX8257	SD存储准备
_SD_Err	BOOL	%KX8258	SD存储错误
_SD_Init	BOOL	%KX8259	SD存储初始化状态
_SD_Closing	BOOL	%KX8260	SD存储关闭状态
_SD_FATErr	BOOL	%KX8261	文件系统错误
_SD_AutoLogAct	BOOL	%KX8262	Act自动记录
_SD_Busy	BOOL	%KX8263	SD存储繁忙状态
_SD_SpaceWarn	BOOL	%KX8264	SD存储不足状态
_SD_Detach	BOOL	%KX8265	SD存储分离状态
_SD_VolTot	UDINT	%KD259	SD存储存储容量(GB)
_SD_VolAvail	UDINT	%KD260	可用存储容量(KB)
_SD_Ecode	WORD	%KW522	SD存储错误代码
_SD_FmtInfo	WORD	%KW523	SD存储格式信息
_SD_FmtRun	BOOL	%KX8368	SD存储格式运行状态
_SD_FmtDone	BOOL	%KX8369	SD存储格式完成状态
_SD_FmtErr	BOOL	%KX8370	SD存储格式失效状态
_SD_FmtEcode	WORD	%KW524	SD存储格式错误代码
_SD_FmtProgress	WORD	%KW525	SD存储格式进程率(%)
_SD_AttachCnt	WORD	%KW526	SD存储附件计数
_SD_DetachCnt	WORD	%KW527	SD存储分离计数
_SD_AddfuncAct	BOOL	%KX8640	SD附件功能运行状态
_SD_AddfuncErr	BOOL	%KX8641	SD附件功能错误状态
_SD_AddfuncDone	BOOL	%KX8642	SD附件功能完成状态
_SD_CmpResult	BOOL	%KX8643	SD比较结果
_SD_AddfuncKind	WORD	%KW541	SD附件功能类型
_SD_AddfuncEcode	WORD	%KW542	SD附件功能错误代码

附录1 标志位列表

8) 数据记录标志位

变量	类型	地址	描述
_DL00_Enable	BOOL	%KX8224	组00数据记录使能状态
_DL00_Rdy	BOOL	%KX8960	组00数据记录准备
_DL00_Act	BOOL	%KX8961	组00数据记录运行状态
_DL00_Err	BOOL	%KX8962	组00数据记录错误状态
_DL00_Stopping	BOOL	%KX8963	组00数据记录停止状态
_DL00_Finish	BOOL	%KX8964	组00数据记录完成状态
_DL00_Trig	BOOL	%KX8965	组00触发发生状态
_DL00_TrigDone	BOOL	%KX8966	组00触发完成状态
_DL00_Evt	BOOL	%KX8967	组00事件发生状态
_DL00_Ovf	BOOL	%KX8968	组00缓存溢出状态
_DL00_Ecode	WORD	%KW561	组00数据记录错误代码
_DL00_FileIdx	WORD	%KW562	组00数据记录文件索引数量
_DL00_FileRollcnt	WORD	%KW563	组00覆盖计数
_DL00_FileSize	UDINT	%KD282	组00文件尺寸(字节)
_DL00_DataRow	UDINT	%KD283	组00数据行数量
_DL00_RemainBuf	UDINT	%KD284	组00剩余缓存尺寸(字节)
_DL00_WaitingData	UDINT	%KD285	组00等待数据尺寸(Byte)
_DL00_OvfCnt	WORD	%KW572	组00缓存溢出计数
_DL00_TrigCnt	WORD	%KW573	组00触发发生计数
_DL00_TrigOvlap	WORD	%KW574	组00触发重叠计数
_DL00_EvtgCnt	WORD	%KW575	组00事件发生计数

9) 编码器标志位

变量	类型	地址	描述
_ENC1_POS	LREAL	%KL0	编码器1输入位置
_ENC2_POS	LREAL	%KL1	编码器2输入位置
_ENC1_UNIT	UINT	%KW8	编码器1单位(0:脉冲, 1:mm, 2:英寸, 3:度)
_ENC2_UNIT	UINT	%KW9	编码器2单位(0:脉冲, 1:mm, 2:英寸, 3:度)

附录2错误信息&解决方法

以下描述信息错误类型和解决方法.

(1) 功能块错误信息

错误代码		错误描述	解决方法
十六进制	十进制		
0005	5	当前运动控制器不支持功能块.	在当前版本控制器中无法执行该指令. 在通过执行指令查看版本后请联系我司客户支援部门.
0006	6	功能块轴数量(轴输入)或者编码器数量(编码器输入)超出可用范围.	设置轴数量为1~36. (编码器数量: 1001~1002)
0007	7	功能块轴组数量(轴组输入)超出可用范围.	设置轴组数量在1到16之间.
0008	8	功能块NC通道数量超出可用范围.	查看NC通道范围并再次设置.
0009	9	功能块从站数量(从站输入)超出可用范围.	查看从站数量范围并再次设置.
000B	11	功能块输入超出可用范围.	查看功能块输入范围并再次设置.
000C	12	功能块阵列输入超出可用范围.	查看功能块阵列输入尺寸并再次设置.
0012	18	当执行功能块时错误块内部执行错误发生.	该问题在当前控制器版本中出现. 请查看MP500和控制器支持版本.
0013	19	功能块执行期间发生动作响应错误.	该问题在当前控制器版本中出现. 请查看MP500和控制器支持版本.
0014	20	超出功能块cam ID(Cam表ID输入)可用范围.	查看cam ID范围并再次设置.

附录 2 错误信息&解决方法

(2)系统错误信息

错误代码		错误描述	解决方法
十六进制	十进制		
000E	14	系统错误	当电源重启后如果问题仍然出现, 请联系售后部门.
0017	23	程序错误	修改并重新加载程序后启动程序
0018	24	IO配置参数异常	更新IO参数后查看保留状态. 如果损坏, 修改并重新下载查看运行. 如果仍然存在异常, 请更换.
0019	25	基本参数异常	上传基本参数后查看保留状态. 如果损坏, 修改并重新下载查看运行. 如果仍然存在异常, 请更换.
001D	29	特殊模块参数异常	上传特殊模块参数后查看保留状态. 如果损坏, 修改并重新下载查看运行. 如果仍然存在异常, 请更换.
0027	39	CPU异常终止或者失效	由于噪声和硬件异常导致系统异常终止. 1) 当电源重启后如果问题仍然出现, 请联系售后部门. 2)阻止噪声措施
002B	43	内置参数-编码器异常	上传内置参数后查看保留状态. 如果损坏, 修改并重新下载查看运行. 如果仍然存在异常, 请更换.
002C	44	轴参数异常	修改参数并重新下载
002D	45	轴组参数异常	修改参数并重新下载
002E	46	EtherCAT参数异常	修改参数并重新下载
002F	47	NC参数异常	修改参数并重新下载
0030	48	NC程序检查错误	查看程序并重新下载
0032	50	外部设备严重失效检测错误	通过参考外部设备(根据参数)严重失效检测错误标志位修理错误设备并重启
0038	56	主任务周期错误	通过增加基本参数主任务周期修改主任务程序或者下载后, 查看主任务周期标志位, 重新下载
0039	57	周期性任务周期错误	通过增加基本参数周期性任务周期修改周期性任务程序或者下载后, 查看周期性任务周期标志位, 重新下载
003A	58	任务程序占有率超出错误	1) 通过减少主任务/周期性任务中用户程序执行数量, 保证系统内部服务运行的时间. 2) 通过设置更高的基本参数中主任务/周期性任务执行周期, 保证系统内部服务运行的时间
003B	59	局部以太网参数检测错误	修改参数并重新下载
01F5	501	RTC数据异常	使用RTC时钟功能复位. 如果仍然发生错误, 请联系售后部分

(3) 数据记录, SD增加功能错误信息

错误代码		错误描述	解决方法
十六进制	十进制		
整体错误代码			
0001	1	SD卡识别错误	FAT32格式并连接到SD存储
0002	2	分割信息错误	FAT32格式并连接到SD存储
0003	3	文件系统错误	FAT32格式并连接到SD存储
0004	4	不支持SD卡	连接SD卡的容量为2GB到32GB
0005	5	SD卡容量检查错误	SD存储容量测试失败, DS无法使用. 更换SD存储或者格式化后重新连接
0006	6	SD卡容量超出	SD存储容量已满, 无法保存数据. 更换SD存储或者格式化后重新连接. 如果可用容量小于20%
0007	7	文件夹创建失败	在SD中创建数据记录文件夹失败. 换SD存储或者格式化后重新连接
数据记录组的错误代码			
1000	4096	组 x 文件夹创建错误	FAT32格式并连接到SD存储
2000	8192	组 x 文件打开错误	FAT32格式并连接到SD存储
4000	16384	组 x 文件写入错误	FAT32格式并连接到SD存储
SD增加功能错误代码			
0001	1	文件错误(文件打开失败, CRC错误)	再次创建文件后运行
0002	2	文件损坏(头和尾损坏, 等)	再次创建文件后运行
0005	5	文件中无密码	密码在PLC中设置, 但是在SD卡的保存文件中没有密码. 设置密码并创建文件.
0006	6	密码错误	设置与PLC的密码与保存在SD卡中的文件密码不匹配. 查看密码后再次确认.
0007	7	MAC地址不匹配	设置MAC地址与PLC的MAC地址不匹配. 查看MAC地址并复位.
000A	10	无保存文件	SD卡中每个保存文件. 创建文件后运行.
000B	11	PLC模式为RUN状态	切换PLC模式到STOP后查看.

(4) 模拟量错误信息

错误代码		错误描述	解决方法
十六进制	十进制		
0064	0100	输入通道0范围设置错误	设置允许范围
0065	0101	输入通道0范围设置错误	设置允许范围
00C8	0200	输入通道0滤波值设置错误	设置允许滤波值
00C9	0201	输入通道0滤波值设置错误	设置允许滤波值
012C	0300	输入通道0平均值设置错误	设置允许平均值
012D	0301	输入通道0平均值设置错误	设置允许平均值
0190	0400	输出通道0范围设置错误	设置允许输出范围
0191	0401	输出通道0范围设置错误	设置允许输出范围
01F4	0500	输出通道0输入值设置错误	设置允许输入值
01F5	0501	输出通道0输入值设置错误	设置允许输入值
0258	0600	输出通道0插值方式范围设置错误	设置允许插值方式范围
0259	0601	输出通道0插值方式范围设置错误	设置允许插值方式范围

附录 2 错误信息&解决方法

(5) 动作错误信息

错误代码		错误描述	解决方法
十六进制	十进制		
0E00	3584	从MP500传输的指令数据范围超出允许值.	在当前控制器版本中可能出现该问题. 请查看MP500和控制器的支持版本.
0E01	3585	如果控制器为RUN状态, MP500测试运行功能无法执行.	变更控制器为STOP状态后执行MP500测试运行.
0E02	3586	如果有一个轴在运行, Cam数据无法写入.	当所有轴不运行时写入cam数据.
0E03	3587	如果有一个轴在运行, 编码器参数无法写入.	当所有轴不运行时写入编码器参数.
0E04	3588	当EtherCAT通讯连接时EtherCAT参数无法写入.	断开EtherCAT 通讯后再次写入EtherCAT参数.
0E10	3600	编码器参数数据异常.	如果重新执行后再次发生错误, 从MP500中再次下载数据并联系售后.
0E11	3601	编码器参数的编码器1脉冲输入类型超出设置范围.	设置编码器参数的编码器1脉冲输入在0到5之间.
0E12	3602	编码器参数的编码器1最大值超出脉冲单位表示值范围.	当转换为脉冲单位时设置编码器参数的编码器1最大值范围在-2,147,483,648到2,147,483,647.
0E13	3603	编码器参数的编码器1最小值超出脉冲单位表示值范围.	当转换为脉冲单位时设置编码器参数的编码器1最小值范围在-2,147,483,648到2,147,483,647.
0E14	3604	编码器参数的编码器1最大值和最小值超出范围.	设置编码器参数的编码器1最小值小于最大值.
0E15	3605	编码器参数的编码器2脉冲输入类型超出范围.	设置编码器参数的编码器2脉冲输入在0和5之间.
0E16	3606	编码器参数的编码器2最大值超出脉冲单位表示值范围.	当转换为脉冲单位时设置编码器参数的编码器2最大值范围在-2,147,483,648到2,147,483,647之间.
0E17	3607	编码器参数的编码器2最小值超出脉冲单位表示值范围.	当转换为脉冲单位时设置编码器参数的编码器2最小值范围在-2,147,483,648到2,147,483,647之间.
0E18	3608	编码器参数的编码器2最大值和最小值超出范围.	设置编码器参数的编码器2最小值小于最大值.
0E19	3609	编码器输入设置不可以在编码器参数的编码器设置之上.	确定编码器参数的编码器相关项目, 并根据范围设置相应值.
0E1A	3610	编码器参数的编码器1每旋转脉冲数量超出设置范围.	设置编码器参数的编码器1每旋转脉冲数量大于0, 小于或等于4294967295.
0E1B	3611	编码器参数的编码器1每旋转传输距离超出设置范围.	设置编码器参数的编码器1每旋转传输距离大于0.00000001, 小于或等于4294967295.
0E1C	3612	编码器参数的编码器2每旋转脉冲数量超出设置范围.	设置编码器参数的编码器2每旋转脉冲数量大于0, 小于或等于4294967295.
0E1D	3613	编码器参数的编码器2每旋转传输距离超出设置范围.	设置编码器参数的编码器2每旋转传输距离大于0.00000001, 小于或等于4294967295.

错误代码		错误描述	解决方法
十六进制	十进制		
0E1E	3614	编码器参数的编码器1输入滤波值超出设置范围.	设置编码器参数的编码器1输入滤波值为0到6之间.
0E1F	3615	编码器参数的编码器2输入滤波值超出设置范围.	设置编码器参数的编码器2输入滤波值为0到6之间.
0E20	3616	编码器参数的编码器1最大值和最小值设置不包含编码器1的当前位置.	设置编码器参数的编码器1最小和最大值包含编码器1当前位置. 或者, 为了根据设置参数运行, 变更编码器当前位置到使用编码器预设指令的参数范围内.
0E21	3617	编码器参数的编码器2最大值和最小值设置不包含编码器2的当前位置.	设置编码器参数的编码器2最小和最大值包含编码器2当前位置. 或者, 为了根据设置参数运行, 变更编码器当前位置到使用编码器预设指令的参数范围内.
0E30	3632	EtherCAT 参数数据异常.	如果重新执行后再次发生错误, 从MP500中再次下载数据并联系售后.
0E31	3633	EtherCAT 参数的周期性通讯超出计数超出范围.	设置EtherCAT 参数的周期性通讯超时计数在1到8之间.
0E32	3634	EtherCAT 参数解析期间发生错误	查看EtherCAT参数并再次设置.
0E40	3648	连接指令无法在EtherCAT 参数以外执行.	查看EtherCAT参数并再次设置.
0E41	3649	EtherCAT从站连接指令运行.	当EtherCAT从站连接指令运行时, 查看EtherCAT从站连接指令是否再次输入.
0E42	3650	EtherCAT从站未连接指令运行.	当EtherCAT从站断开连接指令运行时, 查看EtherCAT从站未连接指令是否再次输入.
0E43	3651	由于模式切换, 连接/未连接指令无法执行.	在EtherCAT从站连接/未连接指令运行时, 查看模式切换开关是否执行.
0E44	3652	连接/未连接指令无法通过ESTOP指令执行.	在EtherCAT从站连接/未连接指令运行时, 查看ESTOP指令是否执行.
0E50	3664	由于异常编码器参数, 编码器预设指令无法执行.	确认编码器参数的编码器相关项目, 查看设置值是否在范围内, 并通过MP500设置编码器参数为正常值.
0E51	3665	由于有一个轴作为主轴进行运行, 预设指令无法执行.	当有一个轴作为编码器主轴运行时, 查看编码器预设指令是否输入.
0E52	3666	编码器预设位置超出编码器最大值或者最小值范围.	设置编码器预设位置到大于或者等于最小值的范围, 和小于或者等于编码器最大值.
0E53	3667	编码器预设指令的所选编码器超出范围.	设置所选编码器为0和1 (0: 编码器1, 1: 编码器2).
0E60	3680	由于基本参数数据异常导致指令无法执行.	如果重新执行后'基本参数错误'再次发生, 从MP500下载基本参数并联系售后部门.

附录 2 错误信息&解决方法

错误代码		错误描述	解决方法
十六进制	十进制		
0E61	3681	Cam数据异常.	如果重新执行后再次发生, 从MP500中下载数据并联系客服部门.
0F00	3840	变更EtherCAT INIT状态失败.	查看通讯电缆状态和从站运行状态(电源导通和错误发生). 并查看通讯电缆是否接触噪声.
0F06	3846	EtherCAT INIT状态初始化(DC_INIT)错误.	查看通讯电缆状态和从站运行状态(电源导通和错误发生). 并查看通讯电缆是否接触噪声.
0F09	3849	没有EtherCAT从站连接到控制器.	查看控制器和EtherCAT 从站间的通讯电缆是否正确安装, 电源正常应用于EtherCAT从站, 或者如果从站连接到控制器, 通讯电缆接触噪声.
0F0A	3850	超出连接从站的最大数量.	确保没有超过64 EtherCAT从站连接到控制器.
0F0E	3854	节点ID和EtherCAT从站设置EtherCAT 参数不同.	查看控制器和EtherCAT从站之间网络电缆连接顺序和EtherCAT参数设置匹配.
0F0F	3855	设置EtherCAT 从站节点ID中有错误.	查看是否有双重节点ID或者设置中错误.
0F10	3856	变更EtherCAT PREOP阶段失败.	查看通讯电缆连接状态和从站运行状态(电源导通和错误发生). 并查看通讯电缆是否接触噪声.
0F1E	3870	没有EtherCAT 参数从站设置数据.	使用MP500设置EtherCAT参数从站, 然后发生EtherCAT 参数到控制器.
0F1F	3871	EtherCAT 参数从站设置数据和连接从站不一致.	设置与实际连接从站信息匹配的EtherCAT 参数从站设置. EtherCAT参数从站使用MP500中的 'EtherCAT从站自动连接功能自动设置.
0F20	3872	变更EtherCAT SAFEOP状态失败.	查看通讯电缆连接状态和从站运行状态(电源导通和错误发生). 并查看通讯电缆是否接触噪声.
0F30	3888	变更EtherCAT OP状态失败.	查看通讯电缆连接状态和从站运行状态(电源导通和错误发生). 并查看通讯电缆是否接触噪声.
0F40	3904	从EtherCAT OP状态到INIT状态变更失败.	查看通讯电缆连接状态和从站运行状态(电源导通和错误发生). 并查看通讯电缆是否接触噪声.
0F50	3920	在通讯连接状态中没有响应.	查看通讯电缆连接状态和从站运行状态(电源导通和错误发生). 并查看通讯电缆是否接触噪声.
0F51	3921	发生周期性通讯错误. (超出主站参数的周期性通讯超时计数错误发生)	查看通讯期间伺服电源是否Off, 通讯电缆安装正确, 或者通讯电缆接触噪声.

错误代码		错误描述	解决方法
十六进制	十进制		
0F60	3936	ESC读取指令的从站设备地址(Adp)设置值超出范围.	查看根据EtherCAT指令代码(EcatCmd)设置值设置的从站设备地址(Adp)范围.
0F61	3937	ESC读取指令的数据尺寸设置值超出范围.	设置ESC读取指令的数据尺寸设置值为1 ~ 4 (BYTE).
0F62	3938	ESC读取指令的EtherCAT 指令代码(EcatCmd)设置值错误.	在1(APRD), 4(FPRD)和7(BRD)之间设置EtherCAT 指令节点.
0F63	3939	用于ESC读取指令的从站设备没有响应.	查看指定为Adp 的从站设备安装正确, 或者Ado地址值在读取-允许区域内.
0F70	3952	ESC写入指令的从站设备地址(Adp)设置值超出范围.	查看根据EtherCAT指令代码(EcatCmd)设置值设置的从站设备地址(Adp)范围.
0F71	3953	ESC写入指令的数据尺寸设置值超出范围.	设置ESC读取指令的数据尺寸设置值为1 ~ 4 (BYTE).
0F72	3954	ESC写入指令的EtherCAT 指令代码(EcatCmd)设置值错误.	在(APWR), 5(FPWR)和8(BWR)之间设置一个EtherCAT指令代码.
0F73	3955	用于ESC写入指令的从站设备没有响应.	查看作为Adp指定的从站设备是否正确安装, 或者Ado地址值在读取-允许区域内.
0F74	3956	在通讯连接/未连接指令执行期间或者在通讯连接状态下, 无法写入指定区域到ESC地址(Ado).	在执行通讯连接/未连接指令或者在通讯连接状态中设置可写入的ESC地址(Ado).
0FF2	4082	由于控制器HW异常, 编码器输入的正常运行无法执行.	再次上电后问题仍然发生时, 请联系售后部门.
1000	4096	轴驱动未准备. (无法连接到EtherCAT网络.)	当轴运行准备时执行指令.
1001	4097	在"禁止"状态下无法执行.	当指令运行时查看指令运行轴状态和执行的指令.
1002	4098	在"停止"状态下无法执行.	当指令运行时查看指令运行轴状态和执行的指令.
1003	4099	在"分离"状态下无法执行.	当指令运行时查看指令运行轴状态和执行的指令.
1004	4100	在"持续"状态下无法执行.	当指令运行时查看指令运行轴状态和执行的指令.
1005	4101	在"同步"状态下无法执行.	当指令运行时查看指令运行轴状态和执行的指令.
1006	4102	在"原点返回"状态下无法执行.	当指令运行时查看指令运行轴状态和执行的指令.
1007	4103	在"停止"状态下无法执行.	当指令运行时查看指令运行轴状态和执行的指令.
1008	4104	在"错误停止"状态下无法执行.	当指令运行时查看指令运行轴状态和执行的指令.
100A	4106	当所属轴组激活时动作指令无法执行.	通过轴组禁止指令变更轴组到组禁止状态后, 执行指令.
100B	4107	该指令无法提供给虚轴.	该指令无法在虚轴执行. 查看指令是否在虚轴运行.

附录 2 错误信息&解决方法

错误代码		错误描述	解决方法
十六进制	十进制		
100C	4108	如果注册为NC通道/轴并在NC控制运行时, 指令无法执行.	当指令运行时查看指令运行轴状态和执行的指令.
100D	4109	由于轴不是使能, 则指令无法执行.	查看指令设置轴是否在轴参数中注册. 在轴参数中注册的轴可以在MP500的动作数据项目中选择.
100E	4110	在动作测试运行指令期间变更到'运行'状态, 并且该运行无法持续.	当轴运行时, 查看控制器是否变更为'运行'状态.
100F	4111	由于控制器因ESTOP 指令而停止, 轴运行无法持续.	查看在轴运行期间, 控制器是否通过ESTOP指令停止.
1010	4112	控制器变更为'停止'或者'错误'状态, 运行无法持续.	当轴运行时, 查看控制器是否变更为'停止'或者'错误'状态.
1011	4113	EtherCAT网络连接丢失, 该运行无法持续.	查看是否由于电源错误, 网络电缆错误和轴运行时网络电缆的噪声流入造成EtherCAT网络连接断开.
1012	4114	指令位置设置值超出脉冲单位表示值范围.	当指令位置值转换为脉冲单位时超出32-位区域. 当转换指令位置值到脉冲时, 设置范围值在-2,147,483,648到2,147,483,647之间.
1013	4115	运行速度值小于0, 或者超出最大速度值.	设置轴中设置的运行速度值大于0并小于最大速度值.
1014	4116	加速设置为负数.	设置加速值为大于或者等于0的值.
1015	4117	减速设置为负数.	设置减速值为大于或者等于0的值.
1016	4118	加速度设置为负数.	设置加速度值为大于或者等于0的值.
1017	4119	指定方向超出范围.	查看指令方向设置值范围, 并设置在范围内的值. (参考第6章指令和功能)
1018	4120	转矩设置值超出范围.	设置转矩设置值在1000%以内.
1019	4121	转矩斜坡设置值超出范围.	设置转矩斜坡设置值为大于或者等于0的值.
101A	4122	缓存模式设置值超出输入范围.	设置值 (0 ~ 5)可以在缓存模式中设置.
101B	4123	执行模式设置值超出输入范围.	设置值 (0 ~ 1)可以在执行模式中设置.
101C	4124	发生跟踪误差超出范围, 该运行无法持续.	指令位置和当前位置间的误差超出'跟踪误差超出范围值'. 为了防止警报发生, 调整伺服驱动或者设置'跟踪误差超出范围值'到一个更大值.
101D	4125	发生跟踪误差超出范围.	指令位置和当前位置间的误差超出'跟踪误差超出范围值'. 为了防止警报发生, 调整伺服驱动或者设置'跟踪误差超出范围值'到一个更大值.

错误代码		错误描述	解决方法
十六进制	十进制		
101F	4127	指令位置值传输到伺服驱动超出脉冲单位表示值范围.	当指令位置值转换为脉冲单位时超出32-位区域. 当转换指令位置值到脉冲时设置值在-2,147,483,648到2,147,483,647范围内.
1020	4128	未定义轴指令.	该指令在当前版本控制器中无法执行. 查看可执行指令的版本后, 联系我司售后部门.
1021	4129	执行相同指令, 前一个执行指令取消.	仅在每次扫描时执行动作指令. 变更程序运行条件, 以便一个动作指令可以在一次扫描中执行.
1022	4130	超出可执行缓存指令的指令数量.	由于轴指令缓存已满, 指令无法执行.通过缓存指令可执行的指令数量为100. 调整指令执行时间.
1030	4144	当轴运行中, 轴参数无法写入.	当轴不运行时执行参数写入.
1040	4160	轴参数数据异常.	如果重新运行后错误仍然存在, 从MP500下载数据后联系售后部门.
1041	4161	由于轴参数错误, 不可以执行运行.	查看轴参数和再次设置.
1042	4162	轴参数速度限制无法设置为小于0的值.	设置基本参数的速度限制为大于0的值.
1043	4163	轴参数软件上/下限值超出范围.	设置轴参数软件上限值大于或者等于软件下限值.
1044	4164	轴参数当前速度滤波时间常数数值超出范围.	设置参数设置值为0 ~ 100.
1045	4165	轴参数错误复位监控时间超出范围.	设置参数设置值为1 ~ 1000.
1046	4166	每旋转传输距离设置值超出范围.	设置参数设置值大于0.000000001和小于4294967295.
1047	4167	无限长度重复位置设置值超出范围.	设置参数以脉冲单位设置值大于0和小于2,147,483,647.
1048	4168	到达位置宽度设置值超出范围.	设置参数以脉冲单位设置值大于0和小于2,147,483,647.
1049	4169	跟踪误差超出范围设置值超出范围.	设置参数以脉冲单位设置值大于0和小于2,147,483,647.
104A	4170	当前位置显示补偿计数设置值超出范围.	设置参数以脉冲单位设置值大于0和小于2,147,483,647.
104B	4171	点动高速设置值超出范围.	设置参数设置值大于0, 大于点动低速值和小于速度限制.
104C	4172	点动低速设置值超出范围.	设置参数设置值大于0, 大于点动高速值和小于速度限制.
104D	4173	点动加速设置值超出范围.	设置参数设置值大于0.

附录 2 错误信息&解决方法

错误代码		错误描述	解决方法
十六进制	十进制		
104E	4174	点动减速设置值超出范围.	设置参数设置值大于0.
104F	4175	点动加速度设置值超出范围.	设置参数设置值大于0.
1050	4176	马达侧齿轮比设置值超出范围.	设置参数设置值为1 ~ 65535.
1051	4177	机械侧齿轮比设置值超出范围.	设置参数设置值为1 ~ 65535.
1052	4178	每旋转脉冲数量设置值超出范围.	设置参数设置值大于 0和小于等于4294967295.
1053	4179	连接设备设置值超出范围.	设置可支持从站的设备数量. 节点设置范围为0(未连接设备), 和1 ~ 64.
1054	4180	轴类型设置值超出范围.	设置参数设置值为0: 实轴' 或者'1: 虚轴'.
1055	4181	速度指令单位设置值超出范围.	设置参数设置值为0: 单位/sec', '1: 单位/min', '2: rpm'.
1060	4192	由于发生伺服驱动错误, 伺服On无法执行.	查看伺服驱动错误原因并移除错误后执行伺服On.
1061	4193	在伺服On进程中间伺服On指令再次执行.	查看在程序或者MP500伺服On进程中间执行伺服On指令.
1062	4194	由于伺服驱动无法变更为"准备切换ON" 状态, 无法完成伺服On.	查看伺服驱动状态.伺服On指令在某些情况下可能无法执行.
1063	4195	由于伺服驱动无法变更为"切换on"阶段, 无法完成伺服On.	查看伺服驱动状态.伺服On指令在某些情况下可能无法执行.
1064	4196	由于伺服驱动无法变更为"运行使能"阶段, 无法完成伺服On.	查看伺服驱动状态.伺服On指令在某些情况下可能无法执行.
1065	4197	由于伺服驱动"快速停止"使能, 无法完成伺服On.	查看伺服驱动状态.伺服On指令在某些情况下可能无法执行.
1066	4198	在伺服Off进程中间伺服Off指令再次执行.	查看在程序或者MP500伺服Off进程中间执行伺服Off指令.
1067	4199	伺服Off 指令执行无法完成.	查看伺服驱动状态.
1070	4208	超出伺服错误复位监控时间.	即使在之前轴参数中设置的错误复位监控时间完成后, 伺服驱动错误也不会清除. 移除伺服驱动错误原因后执行错误复位指令.
1080	4224	使用绝对坐标的指令在未定原点状态的绝对坐标中无法执行.	在制作通过原点返回指令确定初始状态或者当前位置设置指令后, 执行绝对坐标运行指令.
1081	4225	在无限长度重复使能状态, 目标位置超出规范方向的限长度重复位置范围.	在规范方向的限长度重复位置范围内设置目标位置.

错误代码		错误描述	解决方法
十六进制	十进制		
1082	4226	在速度控制或者转矩控制运行中无法执行叠加指令.	当不以速度控制或转矩控制运行时执行叠加指令.
1083	4227	当叠加运行不执行时, 叠加运行停止指令无法执行.	当叠加运行进程时执行叠加运行停止指令.
1090	4240	当前位置变更指令位置值超出范围.	在设置位置设置值大于扩展参数软件下限和小于软件上限后, 执行当前位置预设指令.
1091	4241	对于原点返回运行, 速度同步, cam和转矩控制, 当前位置变更指令无法执行.	当轴不在原点返回, 速度同步, cam 和转矩控制中运行时, 执行当前位置变更.
10A0	4256	伺服驱动不支持转矩控制模式.	通过使用支持EtherCAT CoECST 模式的服务驱动执行转矩控制.
10A1	4257	没有可以以EtherCAT参数从站数据中执行RxPDO 进入设置执行转矩控制的目标转矩对象(0x6071) 设置.	设置支持转矩控制到MP500中EtherCAT参数从站数据RxPDO 项目的目标转矩对象(0x6071), 然后发送到控制器.
10B0	4272	伺服驱动不支持原点返回模式.	通过使用支持EtherCAT CoE 原点返回模式的伺服驱动执行原点返回.
10B1	4273	在伺服驱动执行原点返回期间发生错误.	在排除伺服驱动错误后, 查看伺服驱动错误原因并执行原点返回.
10B2	4274	当轴运行时无法执行原点返回指令.	轴停止后, 在停止状态下再次执行原点返回指令.
10C0	4288	当位置/速度控制不执行时, 覆盖指令无法执行.	以位置控制或者速度控制期间执行覆盖指令.
10C1	4289	覆盖指令覆盖因素超出范围.	设置覆盖指令的VelFactor, AccFactor和JerkFactor值为0或以上值, 然后执行覆盖指令.
10C2	4290	反应覆盖指令因素后, 运行速度值超出最大速度值.	在不超出轴最大速度值的范围内执行覆盖.
10D0	4304	齿轮比分母值不可以为0.	设置齿轮比分母为0以外的值之后, 执行指令.
10D1	4305	齿轮运行主站值源设置值超出范围.	设置主站值源输入值为0或者1之后, 执行指令.
10D2	4306	齿轮运行主轴设置超出范围.	在1~36 (轴)和1001~1002(编码器)区域内设置主轴后执行指令. 如果主轴可用, 查看是否VarOff设置值超出存储区域.
10D3	4307	齿轮运行主轴设置与从轴相同.	设置主轴为从轴(指令轴)以外的其他轴之后, 执行指令.
10D4	4308	齿轮运行主轴未准备.	当主轴准备时执行指令.

附录 2 错误信息&解决方法

错误代码		错误描述	解决方法
十六进制	十进制		
10D5	4309	如果齿轮运行主轴设置为编码器, 如果发生内置参数编码器参数设置错误, 则齿轮运行指令无法执行.	确认编码器参数编码器相关项目, 查看是否设置在范围内, 并通过MP500设置编码器参数为正确值.
10D6	4310	当主轴以转矩控制运行时, 无法执行MC_GearInPos指令.	当主轴不以转矩控制运行时, 执行MC_GearInPos 指令.
10D7	4311	齿轮运行从轴速度超出速度限制.	降低主轴速度或者齿轮运行的从轴齿轮速度应超出从轴设置的速度限制.
10D8	4312	如果不是齿轮运行, 无法执行齿轮发布指令.	齿轮发布指令仅适用于齿轮运行中.
10D9	4313	由于MC_GearInPos 指令目标速度设置值小于当前运行速度或者齿轮运行速度, 无法执行指令.	设置MC_GearInPos指令目标速度设置值到当前运行速度或者齿轮运行速度及以上时, 执行指令.
10DA	4314	当MC_GearInPos 运行期间主轴运行到主轴同步位置时, 无法在规定时间内到达从轴同步位置.	增加目标速度设置值MC_GearInPos 指令, 或者当主轴运动到主轴同步位置时, 调整主站启动距离以便从轴在规定时间内移动到从轴同步位置之后执行指令.
10DB	4315	如果主轴为原点返回运行, 无法执行同步运行指令(齿轮, cam等).	当主轴不在原点返回运行时, 执行同步运行(齿轮, cam等).
10E0	4320	在EtherCAT 参数从站数据中没有对象设置可以使能接触探头到PDO项目设置.	设置支持接触探头到MP500中EtherCAT 参数从站数据的PDO项目的对象, 然后发送到控制器.
10E1	4321	触发输入输入设置值超出范围.	色织触点输入设置值为0(接触探头1)或1(接触探头2).
10F0	4336	参数读取/写入指令的参数数字设置值超出范围.	设置参数读取/写入指令的参数数字设置值为0~28, 100~106和200~206后执行指令.
10F1	4337	参数写入指令设置参数的数据设置值超出范围.	查看设置参数的数据设置范围.
10F2	4338	当编码器参数变更时, 由于编码器1最大值超出脉冲单位表达值, 参数无法变更.	当转换为脉冲单位时预先防止错误, 变更编码器1最大值, 然后变更参数.
10F3	4339	当编码器参数变更时, 由于编码器1最小值超出脉冲单位表达值, 参数无法变更.	当转换为脉冲单位时预先防止错误, 变更编码器1最小值, 然后变更参数.
10F4	4340	当编码器参数变更时, 由于编码器2最大值超出脉冲单位表达值, 参数无法变更.	当转换为脉冲单位时预先防止错误, 变更编码器2最大值, 然后变更参数.

错误代码		错误描述	解决方法
十六进制	十进制		
10F5	4341	当编码器参数变更时, 由于编码器2最小值超出脉冲单位表达值, 参数无法变更.	当转换为脉冲单位时预先防止错误, 变更编码器2最小值, 然后变更参数.
1100	4352	当轴运行时, 无法执行点动运行指令.	当轴为停止状态时执行点动指令.
1110	4368	在cam运行MasterScaling输入值中有错误.	在MasterScaling输入值中不可以输入0.
1111	4369	在cam运行MasterStartDistance输入值中有错误.	设置MasterStartDistance输入值为大于0的数, 并执行指令.
1112	4370	在cam运行MasterSyncPosition输入值中有错误.	设置MasterSyncPosition输入值为大于0的数, 并执行指令.
1113	4371	cam运行启动模式输入值超出范围.	设置启动模式输入值为0和1, 并执行指令.
1114	4372	Cam运行MasterValueSource输入值超出范围.	设置MasterValueSource输入值为0和1, 并执行指令.
1115	4373	指定cam表不存在.	调整cam表数字为有效cam表数字, 并执行指令.
1116	4374	Cam运行主轴设置超出范围.	设置主轴为1~36 (轴)和1001~1002(编码器)区域后执行指令. 如果主轴为变量, 查看是否VarOff设置值偏离存储区域.
1117	4375	Cam运行主轴设置与从轴相同.	设置主轴与从轴(指令轴)不同时执行指令.
1118	4376	Cam运行主轴未准备.	当主轴准备时执行指令.
1119	4377	对于cam运行主轴设置为编码器, 如果内部参数发生错误, 指令无法执行.	确认编码器参数编码器相关项目查看是否设置值在范围内, 并通过MP500设置编码器参数为正常值.
111A	4378	cam运行主轴速度超出速度限制.	降低主轴速度, 或者调整cam表防止cam运行中的从轴速度超出从轴速度限制.
111B	4379	如果没有cam运行, cam发布指令无法执行.	仅当cam 运行时使用cam相关指令.
111C	4380	cam数据写入指令cam数据的设置数量值超出范围.	设置用于设置cam数据写入指令cam数据数量的值小于注册cam点数量. 如果cam表不注册, 增加cam表.
111D	4381	指定cam数据读取指令的cam表数据异常.	如果重启后再次发生, 复位cam数据并联系客服.
111E	4382	如果cam未运行, 无法执行cam跳跃指令.	当cam运行时, 执行cam跳跃指令.
111F	4383	通过cam跳跃指令的cam循环数量设置为0.	通过cam跳跃指令设置跳跃的cam循环数量大于0.
1121	4385	cam跳跃指令的跳跃模式设置值超出范围.	设置cam跳跃指令的跳跃模式设置值在0和2之间后执行指令.

附录 2 错误信息&解决方法

错误代码		错误描述	解决方法
十六进制	十进制		
1122	4386	Cam表未注册.	注册cam表, 或者再次设置数据执行指令.
1123	4387	cam数据写入指令的cam数据异常.	正确设置cam数据写入指令的数据.
1130	4400	如果指令轴不是在同步控制(cam, 齿轮运行)运行的InSync 或者InGear 状态时, 相位校正指令无法执行.	当相位校正指令的指令轴同步控制运行并在InSync 或者InGear 状态时, 执行相位校正指令.
1131	4401	在相位校正指令的主轴设置中有错误.	设置相位校正指令主轴与当前同步运行实轴相同之后, 执行指令.
1132	4402	相位校正指令的相位校正数量超出位置表达式范围.	设置相位校正数量, 保证相位校正数量在脉冲单位范围-2,147,483,648到2,147,483,647之间后执行指令.
1133	4403	相位校正指令速度设置值超出范围.	设置相位校正指令速度值大于0和小于主轴速度限制后执行指令.
1140	4416	连接从站设备不支持速度控制模式.	通过使用支持EtherCAT CoE 速度模式的从站设备执行P速度控制.
1150	4432	连接从站设备不支持位置控制模式.	通过使用支持EtherCAT CoE CSP模式的从站设备执行位置控制.
1200	4608	硬件上限错误发生.	通过反向点动指令脱离外部上限信号范围后, 通过执行错误复位指令移除错误.
1201	4609	硬件下限错误发生.	通过正向点动指令脱离外部下限信号范围后, 通过执行错误复位指令移除错误.
1203	4611	由于运行期间的伺服驱动错误, 指令无法执行.	排除伺服错误因素后通过错误复位指令移除伺服错误.
1204	4612	由于运行期间伺服-off, 指令无法执行.	通过伺服-on 指令变更指令轴为伺服-on状态后, 重启指令.
1205	4613	软件上限错误发生.	通过反向点动指令脱离软件上限范围后, 通过执行错误复位指令移除错误.
1206	4614	软件下限错误发生.	通过正向点动指令脱离软件下限范围后, 通过执行错误复位指令移除错误.
1F00	7936	周期性通讯错误发生. (通讯错误超出主站参数, 发生周期性通讯超时)	查看通讯期间伺服电源是否关断, 通讯电缆是否正常安装和通讯电缆是否接触噪声.
1F10	7952	由于之前执行的从站设备SDO进程失败, 导致SDO指令无法执行	查看是否从站设备状态为正常后再次连接.

错误代码		错误描述	解决方法
十六进制	十进制		
1F11	7953	运行期间SDO参数写入指令无法执行.	当轴不运行时执行SDO 参数写入指令.
1F12	7954	例如SDO 参数索引, 子索引等数据范围超出允许值.	设置SDO参数索引为0x0000~0x9FFF, 分索引为0x00~0xFF, 且数据尺寸为4个字之后执行SDO参数写入指令.
1F13	7955	SDO参数写入指令期间发生中止.	在从站设备写入SDO参数中间停止写入运行. 查看写入数据和从站设备状态.
1F14	7956	相关写入SDO参数指令的从站设备无响应.	在写入SDO 参数中间, 从站设备无响应. 查看从站设备状态.
1F16	7958	当保存SDO参数EEPROM时发生中止.	从站设备中保存SDO参数EEPROM的中间取消操作. 查看从站设备状态.
1F17	7959	相关保存SDO参数EEPROM的从站设备无响应.	在保存SDO 参数EEPROM中间, 从站设备无响应. 查看从站设备状态.
1F19	7961	当写入SDO 参数或者保存SDO 参数EEPROM时其他指令无法执行.	保存SDO 参数EEPROM之后执行其他指令.
1F20	7968	在写入SDO 参数指令中间发生中止.	从站设备中写入SDO参数中间停止写入运行. 查看写入数据和从站设备状态.
1F21	7969	相应读取SDO参数指令的从站设备无响应.	在读取SDO参数EEPROM中间从站设备无响应. 查看从站设备状态.
1F22	7970	当SDO 参数读取/写入指令执行时, SDO参数读取/写入指令无法执行.	当前执行SDO 参数读取/写入完成后执行指令.
1F33	7987	变更伺服驱动的运行模式为位置控制(CSP)模式失效	确认是否伺服驱动支持EtherCAT CoE CSP模式, 并查看伺服驱动状态.
1F34	7988	变更伺服驱动的运行模式为原点返回模式失效	确认是否伺服驱动支持EtherCAT CoE 原点返回模式, 并查看伺服驱动状态.
1F35	7989	变更伺服驱动的运行模式为转矩控制 (CST) 模式失效.	确认是否伺服驱动支持EtherCAT CoE CST 模式, 并查看伺服驱动状态.
1F36	7990	变更伺服驱动的运行模式为速度控制 (CSV) 模式失效.	确认是否伺服驱动支持EtherCAT CoE CSV 模式, 并查看伺服驱动状态.
1F50	8016	由于控制器为RUN状态, MP500手动调谐功能无法执行.	变更控制器为STOP模式后, 执行MP500手动调谐.
1F60	8032	在当前从站运行模式下指令不可用.	设置从站为Boot状态后执行指令.
1F61	8033	文件传输过程中发生发送超时.	查看传输行或者从站状态, 并执行指令.
1F62	8034	文件传输过程中发生接收超时.	查看传输行或者从站状态, 并执行指令.
1F63	8035	文件传输过程中发生数据包错误.	查看传输行或者从站状态, 并执行指令.
1F64	8036	从站存储不足.	查看传输文件, 并执行指令.

附录 2 错误信息&解决方法

错误代码		错误描述	解决方法
十六进制	十进制		
1F65	8037	设备不存在.	查看是否FOE功能为可用从站, 并执行指令.
1F66	8038	访问从站受限.	查看是否FOE功能为可用从站, 并执行指令.
1F67	8039	密码不匹配.	查看密码, 并执行指令.
1F68	8040	通过FoE功能下载的数据没有传输到控制器.	查看通讯电缆连接状态和控制器运行状态.
1F6F	8047	文件传输过程中有一个从站错误.	移除从站错误和执行指令.
2000	8192	轴组未准备运行.	当轴组准备运行时, 执行指令.
2001	8193	轴组无法在"禁止"状态下执行.	当指令运行时查看指令运行轴状态和执行的指令.
2002	8194	轴组无法在"禁止"准备"下执行.	当指令运行时查看指令运行轴状态和执行的指令.
2003	8195	轴组无法在"禁止"移动"下执行.	当指令运行时查看指令运行轴状态和执行的指令.
2004	8196	轴组无法在"禁止"原点返回"下执行.	当指令运行时查看指令运行轴状态和执行的指令.
2005	8197	轴组无法在"禁止"停止"下执行.	当指令运行时查看指令运行轴状态和执行的指令.
2006	8198	轴组无法在"禁止"错误停止"下执行.	当指令运行时查看指令运行轴状态和执行的指令.
2007	8199	轴组的轴配置不在伺服-on状态.	当指令运行时查看指令运行轴状态和执行的指令.
200F	8207	由于通过ESTOP指令停止控制器, 轴组运行无法持续.	查看在轴组运行期间, 是否通过ESTOP指令停止控制器.
2010	8208	控制器变更为'停止' 或者'错误' 状态, 该运行无法持续.	查看轴组运行期间, 是否控制器变更为'停止' 或者'错误'状态.
2011	8209	EtherCAT网络连接丢失, 该运行无法持续.	轴组运行期间, 查看是否由于电源供应错误, 网络电缆错误和网络电缆噪声流入导致EtherCAT网络连接断开.
2012	8210	指令位置设置值超出脉冲单位表示值.	当指令位置值转换为脉冲单位时, 超出32-位区域. 当转换指令位置值为脉冲时, 设置值在-2,147,483,648到2,147,483,647范围内.
2013	8211	运行速度值小于0, 或超出最大速度值.	设置运行速度值大于0, 小于等于轴组中设置的最大速度值.
2014	8212	加速设置为负数.	设置加速值大于等于0.
2015	8213	减速设置为负数.	设置减速值大于等于0
2016	8214	加速度设置为负数.	设置加速度值大于等于0
201A	8218	缓存模式设置值超出范围.	设置可以在缓存模式中设置的值.

错误代码		错误描述	解决方法
十六进制	十进制		
201B	8219	执行模式设置值超出输入范围.	设置在执行模式下可设置(0 ~ 1)的值.
201C	8220	转换模式设置值超出范围.	设置在传输模式指令下可设置的值.
201D	8221	转换参数设置值超出范围.	设置在传输模式指令下可设置的值.
201E	8222	由于轴组配置轴发生错误导致轴组运行停止.	通过轴或者轴组复位指令排除错误因素和移除错误后, 执行指令.
201F	8223	传输到伺服驱动的指令位置值超出脉冲单位表示值.	当指令位置值转换为脉冲单位时超出31-位区域. 当转换为脉冲单位时设置-2,147,483,648 ~ 2,147,483,647范围内的值.
2020	8224	未定义轴组指令.	在当前版本控制器中无法执行指令. 查看可执行指令版本后联系我司客服部门.
2021	8225	由于执行相同指令, 之前执行的指令取消.	查看执行相同指令期间指令是否再次执行.
2022	8226	超出可执行缓存指令的指令数量.	由于轴指令缓存已满, 无法执行指令. 可通过缓存指令执行的指令数量为100. 调整指令执行时序.
2030	8240	如果轴组运行中, 轴组参数无法写入.	当轴组不运行时执行轴组参数写入.
2040	8256	轴组参数数据异常.	如果重新运行后错误仍然存在, 从MP500下载数据后联系售后部门.
2041	8257	由于轴组参数错误导致无法执行运行.	查看轴组参数并再次设置.
2042	8258	轴组参数速度限制无法设置为小于等于0的值.	设置速度限制值大于0.
2043	8259	轴组参数配置轴数量设置值超出范围.	设置轴组配置轴在1~36范围内.
2051	8273	计划添加的轴在轴组中已注册.	查看在轴组中是否存在相同轴数量, 然后设置另一个轴.
2052	8274	当前轴组激活, 计划添加的轴已包含在激活轴组中.	变更包含Group禁止d状态轴的使能轴组后执行指令.
2053	8275	轴组增加/移除指令的IdentInGroup设置值超出范围.	在1~10范围内设置IdentInGroup设置值.
2060	8288	在轴组使能/禁止指令的指定轴组中没有轴设置.	在轴组中设置一个以上的轴, 并执行指令.
2061	8289	由于在当前轴组配置轴中有一个运行轴, 轴组无法使能.	当所有属于轴组的轴不运行时, 执行指令.

附录 2 错误信息&解决方法

错误代码		错误描述	解决方法
十六进制	十进制		
2062	8290	由于当前轴组的配置轴为其他使能轴组的配置轴, 指定轴组无法使能.	查看属于轴组的轴是否包含于另一个使能轴组.
2063	8291	由于轴组的配置轴具有不同的单位, 无法执行轴组运行.	为了顺序执行运行, 设置属于轴组的配置轴单位相同.
2064	8292	由于轴组配置轴参数错误, 轴组无法使能.	在正常范围内设置属于轴组的配置轴参数.
2065	8293	由于轴组配置轴速度指令单位彼此不同, 轴组无法使能.	属于轴组的配置轴设置相同速度指令单位.
2066	8294	由于轴组配置轴中有一个轴的速度指令单位为rpm, 轴组无法使能.	属于轴组的配置轴速度指令单位不可以以设置为rpm, 请设置除rpm以外的单位.
2067	8295	由于轴组配置轴单位与坐标类型不同, 坐标系运行无法执行.	为了顺序执行运行, 设置属于轴组的配置轴单位符合坐标系统类型.
206F	8303	如果轴组配置轴以NC控制运行, 轴组无法激活.	当属于轴组的配置轴不以NC 控制运行时, 执行指令.
2070	8304	配置轴伺服驱动不支持原点返回模式.	确认是否伺服驱动支持EtherCAT CoE原点返回模式, 并查看伺服驱动状态.
2071	8305	配置轴中有一个轴原点返回未完成.	通过查看配置轴错误代码查看错误原因之后, 再次执行指令.
2072	8306	当轴组运行时, 轴组原点返回指令无法执行.	轴组运行停止后, 以GroupStandby状态再次执行轴组原点返回指令.
2080	8320	在配置轴当前位置设置期间, 有一个轴发生错误.	通过查看配置轴错误代码查看错误原因之后, 再次执行指令.
2090	8336	如果配置轴在原点未确定状态下, 绝对坐标线性插值指令无法执行.	通过原点返回指令或者当前位置设置指令确定原点状态后, 执行指令.
2091	8337	超出线性插值配置轴速度限制.	以低速指令速度执行指令, 以免超出配置轴速度限制.
2092	8338	对于指定转角距离转换, 由于转角距离指定值大于到目标位置的移动距离, 转换运行无法执行.	设置指定于传输参数的转角距离值小于到目标位置的移动距离.
2093	8339	对于指定转角距离转换, 由于插入的圆弧半径超出2,147,483,647脉冲, 转换运行无法执行.	通过复位目标位置或者变更传输模式执行线性插值, 以便两条直线无法处于一条直线.

错误代码		错误描述	解决方法
十六进制	十进制		
2094	8340	当主轴或者从轴为无限长度重复"允许"状态, 线性插值运行无法执行	变更主轴或者从轴无限长度重复设置为"0: 禁止"后执行指令.
20A0	8352	当配置轴为原点未确定状态时, 绝对坐标圆弧插值指令无法执行.	通过原点返回指令或者当前位置设置指令确定原点状态后, 执行指令.
20A1	8353	圆弧插值模式设置值超出范围.	设置圆弧插值模式为0和2 (0:辅助点, 1:中心点, 2:半径)之间.
20A2	8354	圆弧插值方向选择设置值超出范围.	设置圆弧插值方向选择设置值为 0和1 (0: CW, 1: CCW).
20A3	8355	以圆弧插值半径方式的半径设置超出范围.	设置圆弧插值主轴运行数据半径设置值大于从起始点到结束点一半长度的 80%.
20A4	8356	在圆弧插值中,如果起始点= 中心点(中点) 或者中心点(中点) = 结束点, 无法执行运行.	执行 the 圆弧插值after 设置中心点(或者中点)为与起始点(或者结束点)不同位置后执行圆弧插值.
20A5	8357	在圆弧插值中, 如果是中点(或者半径)方式, 起始点和结束点不可以相同.	如果设置圆弧插值为中点(或者半径), 设置起始点位置与结束点不同, 然后执行圆弧插值.
20A6	8358	在圆弧插值中的半径设置错误.	可执行圆弧插值的圆弧半径大于0和小于2,147,483,647脉冲. 在可计算半径设置范围内设置输入值之后, 执行指令.
20A7	8359	由于线性轮廓出现在圆弧插值中, 无法执行运行.	在圆弧插值中点方式的情况下, 执行圆弧插值变更中点, 以便起始点和结束点无法位于一条直线.
20A8	8360	当主轴或者从轴为无限长度重复 "允许"状态, 无法执行线性插值运行.	变更主轴或者从轴无限重复设置为"0: 禁止"后执行指令.
20A9	8361	如果轴组构成超过4个轴, 无法执行圆弧插值.	圆弧插值设置轴组为2-轴, 螺旋插值设置为3-轴.
20AA	8362	如果轴组轴配置不是以规定顺序配置, 无法执行圆弧插值.	关于圆弧插值, 以规定顺序设置轴组的配置轴.
20AB	8363	超出圆弧插值配置轴的速度限制.	以低速执行指令, 以免超出配置轴速度限制.
20AC	8364	在圆弧插值中,中点(中心点)必须在相同XY 平面, 作为中点(或者半径) 方式中的起始点.	设置中心点 (or 中点)到相同XY平面位置, 作为圆弧插值起始点(结束点)之后执行圆弧插值.
20C0	8384	当构成轴为原点未确定状态时, 坐标系统运行指令无法执行.	通过原点返回指令或者当前位置设置指令确定原点状态后执行指令.
20C1	8385	坐标系统参数的PCS设置参数数据异常.	查看PCS设置参数, 并再次设置.

附录 2 错误信息&解决方法

错误代码		错误描述	解决方法
十六进制	十进制		
20C2	8386	坐标系统参数, 坐标系统-类型参数数据异常.	查看坐标系统-类型参数, 并再次设置.
20C3	8387	坐标系统参数, 坐标系统-类型参数数据异常.	查看工具参数, 并再次设置.
20C4	8388	坐标系统参数, 工作区-类型数据异常.	查看工作区-类型参数, 并再次设置.
20C5	8389	坐标系统参数, 工作区参数数据异常.	查看工作区参数, 并再次设置.
20C6	8390	无法启动坐标系统运行的位置.	移动到可以启动坐标系统运行的位置, 并执行指令.
20C7	8391	无法通过坐标系统运行到达的目标位置.	查看在目标位置或者坐标系统参数中是否有异常, 并再次设置.
20C8	8392	运行超出工作区.	查看在工作区参数或者目标位置中是否有异常, 并再次设置.
20C9	8393	由于轴组配置轴单位与坐标系统类型不同, 轴组无法激活.	确定属于轴组的配置轴单位与坐标系统时间匹配.
20CA	8394	坐标系统运行超出最大插值速度.	以低速指令速度执行指令, 以免超出最大插值速度.
20CB	8395	当配置轴为无限长度重复 "允许"状态时, 坐标系统运行无法.	变更配置轴无限长度重复设置为"0: 禁止"后执行指令.
20CC	8396	不支持的CoordSystem.	设置支持的CoordSystem后执行指令.
20CD	8397	不支持的TrajType.	设置支持的TrajType后执行指令.
20D0	8400	传送带轴设置值超出范围.	在范围1~36中设置传送带轴.
20D1	8401	设置为传送带轴的轴被设置为轴组配置轴.	当传送带轴设置为另一个轴时执行指令.
20D2	8402	在设置传送带轴单位中有错误.	设置传送带轴单位为mm/inch.
20D3	8403	传送带轴未准备.	当传送带轴准备运行时, 执行指令.
20D4	8404	如果主轴以原点返回运行, 传送带同步指令无法执行.	当传送带轴不在原点返回运行时, 执行指令.
20D5	8405	如果主轴以转矩控制运行, 传送带同步指令无法执行.	当传送带轴不以转矩控制运行时, 执行指令.
20D6	8406	如果传送带轴为无限长度重复 "禁止"状态, 传送带同步功能无法执行.	变更传送带轴无限长度重复设置为"1: 允许"之后执行指令.
20E0	8416	坐标系统路径运行SETP值超出范围.	设置坐标系统路径运行STEP值为0到99之间的值之后, 执行指令.
20E1	8417	坐标系统路径运行指令类型值超出范围.	设置坐标系统路径运行指令类型值为0到4之间的值之后, 执行指令.
20E2	8418	坐标系统路径运行模式值超出范围.	设置坐标系统路径运行模式值为0到2之间的值之后, 执行指令.

附录 2 错误信息&解决方法

错误代码		错误描述	解决方法
十六进制	十进制		
20E3	8419	超出可通过坐标系统路径运行执行的路径数量.	设置坐标系统路径运行STEP值为0到99之间的值之后, 执行指令.
3000	12288	NC通道未准备运行.	查看是否NC通道准备运行. 为了使用NC通道, 应在MP500的NC参数中注册NC通道.
3001	12289	NC程序数据异常.	如果重启后问题仍然存在, 从MP500再次下载数据后并联系售后部门.
3002	12290	当NC通道自动运行时, 程序无法写入.	如果NC通道为自动运行, 当自动运行停止时执行程序写入.
3003	12291	NC 程序写入未正常完成. (在写入NC程序中文件处理(删除)失败)	如果重启后问题仍然存在, 从MP500再次下载数据后并联系售后部门.
3004	12292	NC 程序写入未正常完成. (在写入NC程序中文件处理(打开)失败)	如果重启后问题仍然存在, 从MP500再次下载数据后并联系售后部门.
3005	12293	NC 程序写入未正常完成. (在写入NC程序中文件处理(写入)失败)	如果重启后问题仍然存在, 从MP500再次下载数据后并联系售后部门.
3006	12294	NC 程序写入未正常完成. (在写入NC程序中文件处理(关闭)失败)	如果重启后问题仍然存在, 从MP500再次下载数据后并联系售后部门.
3007	12295	当NC通道自动运行时参数无法写入.	如果NC通道为自动运行, 当自动运行停止时执行程序写入.
3008	12296	由于控制器模式变更为STOP或者NC通道自动运行期间的错误状态, 自动运行无法持续.	查看在NC通道自动运行期间, 是否控制器模式变更为STOP或者错误状态.
3009	12297	由于在NC通道自动运行期间EtherCAT网络连接断开, 自动运行无法持续.	查看在NC通道自动运行期间, 是否由于从站电源供应错误, 网路电缆错误和网络电缆噪声干扰, EtherCAT网络连接断开.
300A	12298	由于在NC通道自动运行期间, 控制器通过ESTOP指令停止, 自动运行无法持续.	查看在NC通道自动运行期间, 控制器是否通过ESTOP指令停止.
3011	12305	由于 NC X 轴未准备运行, 自动运行无法持续.	查看是否NC X轴在伺服-off状态或者驱动警报状态. 当配置轴为伺服-on状态或者驱动警报不发生时, 执行NC通道自动运行.
3012	12306	由于 NC Y轴未准备运行, 自动运行无法持续.	查看是否NC Y轴在伺服-off状态或者驱动警报状态. 当配置轴为伺服-on状态或者驱动警报不发生时, 执行NC通道自动运行.

附录 2 错误信息&解决方法

错误代码		错误描述	解决方法
十六进制	十进制		
3013	12307	由于 NC Z轴未准备运行, 自动运行无法持续.	查看是否NC Z轴在伺服-off状态或者驱动警报状态. 当配置轴在伺服-on状态或者驱动警报不发生时, NC通道自动运行可用执行.
3014	12308	由于 NC A轴未准备运行, 自动运行无法持续.	查看是否NC A轴在伺服-off状态或者驱动警报状态. 当配置轴在伺服-on状态或者驱动警报不发生时, NC通道自动运行可用执行.
3015	12309	由于 NC B轴未准备运行, 自动运行无法持续.	查看是否NC B轴在伺服-off状态或者驱动警报状态. 当配置轴在伺服-on状态或者驱动警报不发生时, NC通道自动运行可用执行.
3016	12310	由于 NC C轴未准备运行, 自动运行无法持续.	查看是否NC C轴在伺服-off状态或者驱动警报状态. 当配置轴在伺服-on状态或者驱动警报不发生时, NC通道自动运行可用执行.
3017	12311	由于 NC U轴未准备运行, 自动运行无法持续.	查看是否NC U轴在伺服-off状态或者驱动警报状态. 当配置轴在伺服-on状态或者驱动警报不发生时, NC通道自动运行可用执行.
3018	12312	由于 NC V轴未准备运行, 自动运行无法持续.	查看是否NC V轴在伺服-off状态或者驱动警报状态. 当配置轴在伺服-on状态或者驱动警报不发生时, NC通道自动运行可用执行.
3019	12313	由于 NC W轴未准备运行, 自动运行无法持续.	查看是否NC W轴在伺服-off状态或者驱动警报状态. 当配置轴在伺服-on状态或者驱动警报不发生时, NC通道自动运行可用执行.
301A	12314	由于 NC S轴未准备运行, 自动运行无法持续.	查看是否NC S轴在伺服-off状态或者驱动警报状态. 当配置轴在伺服-on状态或者驱动警报不发生时, NC通道自动运行可用执行.
3020	12320	未定义NC通道指令.	NC指令无法支持当前控制器版本. 查看支持指令执行版本后联系我司客户支援部门.
3021	12321	由于执行相同NC通道指令, 前一个执行指令取消.	NC通道指令仅在一次扫描执行. 变更程序运行条件以便NC通道指令可以在一次扫描内执行.
3030	12336	由于在NC通道自动运行期间发生注释(IPR)警报, 自动运行无法持续.	查看NC通道标志位的注释(IPR)错误代码. 通过NC复位指令(NC_Reset)复位NC通道后再次执行自动运行启动指令(NC_CycleStart).
3031	12337	由于在NC通道自动运行期间发生程序处理器(PA)警报, 自动运行无法持续.	查看NC通道标志位的程序处理器(PA)错误代码. 通过NC复位指令(NC_Reset)复位NC通道后再次执行自动运行启动指令(NC_CycleStart).

错误代码		错误描述	解决方法
十六进制	十进制		
3040	12352	NC通道自动运行期间, 指令位置设置值超出脉冲单位表达式值.	当指令位置值转换为脉冲单位时超出32-位区域. 当转换指令位置值为脉冲时, 设置范围值-2,147,483,648到2,147,483,647.
3041	12353	NC X轴指令位置设置值超出脉冲单位表达式值.	当NC X轴指令位置值转换为脉冲单位时超出32-位区域. 当转换指令位置值为脉冲时, 设置范围值-2,147,483,648到2,147,483,647.
3042	12354	NC Y指令位置设置值超出脉冲单位表达式值.	当NC Y轴指令位置值转换为脉冲单位时超出32-位区域. 当转换指令位置值为脉冲时, 设置范围值-2,147,483,648到2,147,483,647.
3043	12355	NC Z指令位置设置值超出脉冲单位表达式值.	当NC Z轴指令位置值转换为脉冲单位时超出32-位区域. 当转换指令位置值为脉冲时, 设置范围值-2,147,483,648到2,147,483,647.
3044	12356	NC A指令位置设置值超出脉冲单位表达式值.	当NC A轴指令位置值转换为脉冲单位时超出32-位区域. 当转换指令位置值为脉冲时, 设置范围值-2,147,483,648到2,147,483,647.
3045	12357	NC B指令位置设置值超出脉冲单位表达式值.	当NC B轴指令位置值转换为脉冲单位时超出32-位区域. 当转换指令位置值为脉冲时, 设置范围值-2,147,483,648到2,147,483,647.
3046	12358	NC C指令位置设置值超出脉冲单位表达式值.	当NC C轴指令位置值转换为脉冲单位时超出32-位区域. 当转换指令位置值为脉冲时, 设置范围值-2,147,483,648到2,147,483,647.
3047	12359	NC U指令位置设置值超出脉冲单位表达式值.	当NC U轴指令位置值转换为脉冲单位时超出32-位区域. 当转换指令位置值为脉冲时, 设置范围值-2,147,483,648到2,147,483,647.
3048	12360	NC V指令位置设置值超出脉冲单位表达式值.	当NC V轴指令位置值转换为脉冲单位时超出32-位区域. 当转换指令位置值为脉冲时, 设置范围值-2,147,483,648到2,147,483,647.
3049	12361	NC W指令位置设置值超出脉冲单位表达式值.	当NC W轴指令位置值转换为脉冲单位时超出32-位区域. 当转换指令位置值为脉冲时, 设置范围值-2,147,483,648到2,147,483,647.
304A	12362	NC S指令位置设置值超出脉冲单位表达式值.	当NC S轴指令位置值转换为脉冲单位时超出32-位区域. 当转换指令位置值为脉冲时, 设置范围值-2,147,483,648到2,147,483,647.

附录 2 错误信息&解决方法

错误代码		错误描述	解决方法
十六进制	十进制		
3050	12368	NC通道配置轴指令位置超出软件上限位置.	在错误发生的NC配置轴中使用反向点动指令, 从软件上限范围中分离, 然后通过执行错误复位指令移除错误.
3051	12369	NC X轴指令位置超出软件上限位置.	在错误发生的NC X轴中使用反向点动指令, 从软件上限范围中分离, 然后通过执行错误复位指令移除错误.
3052	12370	NC Y轴指令位置超出软件上限位置.	在错误发生的NC Y轴中使用反向点动指令, 从软件上限范围中分离, 然后通过执行错误复位指令移除错误.
3053	12371	NC Z轴指令位置超出软件上限位置.	在错误发生的NC Z轴中使用反向点动指令, 从软件上限范围中分离, 然后通过执行错误复位指令移除错误.
3054	12372	NC A轴指令位置超出软件上限位置.	在错误发生的NC A轴中使用反向点动指令, 从软件上限范围中分离, 然后通过执行错误复位指令移除错误.
3055	12373	NC B轴指令位置超出软件上限位置.	在错误发生的NC B轴中使用反向点动指令, 从软件上限范围中分离, 然后通过执行错误复位指令移除错误.
3056	12374	NC C轴指令位置超出软件上限位置.	在错误发生的NC C轴中使用反向点动指令, 从软件上限范围中分离, 然后通过执行错误复位指令移除错误.
3057	12375	NC U轴指令位置超出软件上限位置.	在错误发生的NC U轴中使用反向点动指令, 从软件上限范围中分离, 然后通过执行错误复位指令移除错误.
3058	12376	NC V轴指令位置超出软件上限位置.	在错误发生的NC V轴中使用反向点动指令, 从软件上限范围中分离, 然后通过执行错误复位指令移除错误.
3059	12377	NC W轴指令位置超出软件上限位置.	在错误发生的NC W轴中使用反向点动指令, 从软件上限范围中分离, 然后通过执行错误复位指令移除错误.
305A	12378	NC S轴指令位置超出软件上限位置.	在错误发生的NC S轴中使用反向点动指令, 从软件上限范围中分离, 然后通过执行错误复位指令移除错误.
305B	12379	NC通道/轴指令位置超出G22移位禁止区域内部范围.	在错误发生的NC配置轴中使用点动指令, 从G22移位禁止区域范围中分离, 然后通过执行错误复位指令移除错误.
305C	12380	NC通道/轴指令位置超出G22移位禁止区域外部范围.	在错误发生的NC配置轴中使用点动指令, 从G22移位禁止区域范围中分离, 然后通过执行错误复位指令移除错误.
305D	12381	NC通道/轴指令位置超出第三移位禁止区域外部范围.	在错误发生的NC配置轴中使用点动指令, 从第三移位禁止区域范围中分离, 然后通过执行错误复位指令移除错误.

错误代码		错误描述	解决方法
十六进制	十进制		
3060	12384	NC通道配置轴指令位置超出软件下限位置.	在错误发生的NC配置轴中使用正向点动指令, 从软件下限范围中分离, 然后通过执行错误复位指令移除错误.
3061	12385	NC X轴指令位置超出软件下限位置.	在错误发生的NC X轴中使用正向点动指令, 从软件下限范围中分离, 然后通过执行错误复位指令移除错误.
3062	12386	NC Y轴指令位置超出软件下限位置.	在错误发生的NC Y轴中使用正向点动指令, 从软件下限范围中分离, 然后通过执行错误复位指令移除错误.
3063	12387	NC Z轴指令位置超出软件下限位置.	在错误发生的NC Z轴中使用正向点动指令, 从软件下限范围中分离, 然后通过执行错误复位指令移除错误.
3064	12388	NC A轴指令位置超出软件下限位置.	在错误发生的NC A轴中使用正向点动指令, 从软件下限范围中分离, 然后通过执行错误复位指令移除错误.
3065	12389	NC B轴指令位置超出软件下限位置.	在错误发生的NC B轴中使用正向点动指令, 从软件下限范围中分离, 然后通过执行错误复位指令移除错误.
3066	12390	NC C轴指令位置超出软件下限位置.	在错误发生的NC C轴中使用正向点动指令, 从软件下限范围中分离, 然后通过执行错误复位指令移除错误.
3067	12391	NC U轴指令位置超出软件下限位置.	在错误发生的NC U轴中使用正向点动指令, 从软件下限范围中分离, 然后通过执行错误复位指令移除错误.
3068	12392	NC V轴指令位置超出软件下限位置.	在错误发生的NC V轴中使用正向点动指令, 从软件下限范围中分离, 然后通过执行错误复位指令移除错误.
3069	12393	NC W轴指令位置超出软件下限位置.	在错误发生的NC W轴中使用正向点动指令, 从软件下限范围中分离, 然后通过执行错误复位指令移除错误.
306A	12394	NC S轴指令位置超出软件下限位置.	在错误发生的NC S轴中使用正向点动指令, 从软件下限范围中分离, 然后通过执行错误复位指令移除错误.
3071	12401	由于 NC X轴不是原点确定完成状态, 自动运行无法持续.	查看NC X轴是否不在原点未确定状态. 可以通过使用原点返回指令(MC_Home, NC_Home)变更轴到确定原点状态.
3072	12402	由于 NC Y轴不是原点确定完成状态, 自动运行无法持续.	查看NC Y轴是否不在原点未确定状态. 可以通过使用原点返回指令(MC_Home, NC_Home)变更轴到确定原点状态.
3073	12403	由于 NC Z轴不是原点确定完成状态, 自动运行无法持续.	查看NC Z轴是否不在原点未确定状态. 可以通过使用原点返回指令(MC_Home, NC_Home)变更轴到确定原点状态.
3074	12404	由于 NC A轴不是原点确定完成状态, 自动运行无法持续.	查看NC A轴是否不在原点未确定状态. 可以通过使用原点返回指令(MC_Home, NC_Home)变更轴到确定原点状态.

附录 2 错误信息&解决方法

错误代码		错误描述	解决方法
十六进制	十进制		
3075	12405	由于 NC B轴不是原点确定完成状态, 自动运行无法持续.	查看NC B轴是否在原点未确定状态. 可以通过使用原点返回指令(MC_Home, NC_Home)变更轴到确定原点状态.
3076	12406	由于 NC C轴不是原点确定完成状态, 自动运行无法持续.	查看NC C轴是否在原点未确定状态. 可以通过使用原点返回指令(MC_Home, NC_Home)变更轴到确定原点状态.
3077	12407	由于 NC U轴不是原点确定完成状态, 自动运行无法持续.	查看NC U轴是否在原点未确定状态. 可以通过使用原点返回指令(MC_Home, NC_Home)变更轴到确定原点状态.
3078	12408	由于 NC V轴不是原点确定完成状态, 自动运行无法持续.	查看NC V轴是否在原点未确定状态. 可以通过使用原点返回指令(MC_Home, NC_Home)变更轴到确定原点状态.
3079	12409	由于 NC W轴不是原点确定完成状态, 自动运行无法持续.	查看NC W轴是否在原点未确定状态. 可以通过使用原点返回指令(MC_Home, NC_Home)变更轴到确定原点状态.
307A	12410	由于 NC S轴不是原点确定完成状态, 自动运行无法持续.	查看NC S轴是否在原点未确定状态. 可以通过使用原点返回指令(MC_Home, NC_Home)变更轴到确定原点状态.
3080	12416	由于在NC通道自动运行期间发生NC配置轴驱动异常条件(上/下限, 警报, 伺服off), 自动运行无法持续.	查看NC通道自动运行期间, 是否NC配置轴驱动状态变更为上限/下限, 警报发生或者伺服-off状态后移除异常条件原因.
3081	12417	由于在NC通道自动运行期间发生NC X轴驱动异常条件(上/下限, 警报, 伺服off), 自动运行无法持续.	查看NC通道自动运行期间, 是否NC X轴驱动状态变更为上限/下限, 警报发生或者伺服-off状态后移除异常条件原因.
3082	12418	由于在NC通道自动运行期间发生NC Y轴驱动异常条件(上/下限, 警报, 伺服off), 自动运行无法持续.	查看NC通道自动运行期间, 是否NC Y轴驱动状态变更为上限/下限, 警报发生或者伺服-off状态后移除异常条件原因.
3083	12419	由于在NC通道自动运行期间发生NC Z轴驱动异常条件(上/下限, 警报, 伺服off), 自动运行无法持续.	查看NC通道自动运行期间, 是否NC Z轴驱动状态变更为上限/下限, 警报发生或者伺服-off状态后移除异常条件原因.
3084	12420	由于在NC通道自动运行期间发生NC A轴驱动异常条件(上/下限, 警报, 伺服off), 自动运行无法持续.	查看NC通道自动运行期间, 是否NC A轴驱动状态变更为上限/下限, 警报发生或者伺服-off状态后移除异常条件原因.
3085	12421	由于在NC通道自动运行期间发生NC B轴驱动异常条件(上/下限, 警报, 伺服off), 自动运行无法持续.	查看NC通道自动运行期间, 是否NC B轴驱动状态变更为上限/下限, 警报发生或者伺服-off状态后移除异常条件原因.
3086	12422	由于在NC通道自动运行期间发生NC C轴驱动异常条件(上/下限, 警报, 伺服off), 自动运行无法持续.	查看NC通道自动运行期间, 是否NC C轴驱动状态变更为上限/下限, 警报发生或者伺服-off状态后移除异常条件原因.

错误代码		错误描述	解决方法
十六进制	十进制		
3087	12423	由于在NC通道自动运行期间发生NC U轴驱动异常条件(上/下限, 警报, 伺服off), 自动运行无法持续.	查看NC通道自动运行期间, 是否NC U轴驱动状态变更为上限/下限, 警报发生或者伺服-off状态后移除异常条件原因.
3088	12424	由于在NC通道自动运行期间发生NC V轴驱动异常条件(上/下限, 警报, 伺服off), 自动运行无法持续.	查看NC通道自动运行期间, 是否NC V轴驱动状态变更为上限/下限, 警报发生或者伺服-off状态后移除异常条件原因.
3089	12425	由于在NC通道自动运行期间发生NC W轴驱动异常条件(上/下限, 警报, 伺服off), 自动运行无法持续.	查看NC通道自动运行期间, 是否NC W轴驱动状态变更为上限/下限, 警报发生或者伺服-off状态后移除异常条件原因.
308A	12426	由于在NC通道自动运行期间发生NC S轴驱动异常条件(上/下限, 警报, 伺服off), 自动运行无法持续.	查看NC通道自动运行期间, 是否NC S轴驱动状态变更为上限/下限, 警报发生或者伺服-off状态后移除异常条件原因.
3100	12544	NC通道参数数据异常.	如果重启后错误仍然存在, 从MP500下载数据后联系售后部门.
3101	12545	由于NC参数异常, 运行无法执行.	如果例如数据范围的设置错误, 查看NC参数并复位.
3102	12546	NC通道参数进刀上/下限比率设置值超出范围.	设置NC通道参数进刀上限/下限比率值大于0. 设置进刀上限比率值大于进刀下限比率值.
3103	12547	NC通道圆弧加工速度限制上/下限进刀比率设置值超出范围.	设置NC通道参数圆弧加工速度限制上限/下限进刀值大于0. 设置圆弧加工速度限制上限速度值大于圆弧加工速度限制下限速度值.
3200	12800	NC通道/轴参数数据异常.	如果重启后错误仍然存在, 从MP500下载数据后联系售后部门.
3310	13072	NC进给保持指令在自动运行以外的状态执行, 或者当前执行程序块无法执行NC进给保持.	当NC通道自动运行时, 执行 NC 馈送保持指令(NC_FeedHold). 查看在自动运行时, 是否当前执行程序块准备馈送保持.
3320	13088	NC快速移位覆盖指令的覆盖因素超出范围.	设置覆盖指令的VelFactor, AccFactor和JerkFactor值大于0之后执行覆盖指令.
3321	13089	出现NC快速移位覆盖指令的覆盖因素后, NC X轴运行速度值超出速度限制值.	查看连接到NC X轴的轴速度限制值后, 在以不超出速度限制范围的值执行覆盖.
3322	13090	出现NC快速移位覆盖指令的覆盖因素后, NC Y轴运行速度值超出速度限制值.	查看连接到NC Y轴的轴速度限制值后, 在以不超出速度限制范围的值执行覆盖.

附录 2 错误信息&解决方法

错误代码		错误描述	解决方法
十六进制	十进制		
3323	13091	出现NC快速移位覆盖指令的覆盖因素后, NC Z轴运行速度值超出速度限制值.	查看连接到NC Z轴的轴速度限制值后, 在以不超出速度限制范围的值执行覆盖.
3324	13092	出现NC快速移位覆盖指令的覆盖因素后, NC A轴运行速度值超出速度限制值.	查看连接到NC A轴的轴速度限制值后, 在以不超出速度限制范围的值执行覆盖.
3325	13093	出现NC快速移位覆盖指令的覆盖因素后, NC B轴运行速度值超出速度限制值.	查看连接到NC B轴的轴速度限制值后, 在以不超出速度限制范围的值执行覆盖.
3326	13094	出现NC快速移位覆盖指令的覆盖因素后, NC C轴运行速度值超出速度限制值.	查看连接到NC C轴的轴速度限制值后, 在以不超出速度限制范围的值执行覆盖.
3327	13095	出现NC快速移位覆盖指令的覆盖因素后, NC U轴运行速度值超出速度限制值.	查看连接到NC U轴的轴速度限制值后, 在以不超出速度限制范围的值执行覆盖.
3328	13096	出现NC快速移位覆盖指令的覆盖因素后, NC V轴运行速度值超出速度限制值.	查看连接到NC V轴的轴速度限制值后, 在以不超出速度限制范围的值执行覆盖.
3329	13097	出现NC快速移位覆盖指令的覆盖因素后, NC W轴运行速度值超出速度限制值.	查看连接到NC W轴的轴速度限制值后, 在以不超出速度限制范围的值执行覆盖.
332A	13098	出现NC快速移位覆盖指令的覆盖因素后, NC S轴运行速度值超出速度限制值.	查看连接到NC S轴的轴速度限制值后, 在以不超出速度限制范围的值执行覆盖.
332B	13099	通过NC_Reset 或者NC_Emergency 指令执行复位, 如此NC快速移位覆盖指令取消.	如果需要NC快速移位覆盖功能, NC_Reset或者NC_Emergency 指令结束后再次执行指令.
332C	13100	当通过NC_Reset 或者NC_Emergency 指令复位时, NC_RapidTraverseOverride指令无法执行.	NC_Reset或者NC_Emergency 指令结束后执行NC指令.
3330	13104	NC进刀覆盖指令覆盖因素超出范围.	设置覆盖指令的VelFactor, AccFactor和JerkFactor值大于0之后, 执行覆盖指令.
3331	13105	出现NC进刀覆盖指令的覆盖因素后, 运行速度值超出进刀上限比率值.	查看NC通道参数进刀上限比率值之后, 执行不超过进刀上限比率值的覆盖.

错误代码		错误描述	解决方法
十六进制	十进制		
3332	13106	通过NC_Reset或者NC_Emergency 指令执行复位, 如此NC进刀覆盖指令取消.	如果需要NC进刀覆盖功能, 在NC_Reset或者NC_Emergency 指令结束后再次执行指令.
3333	13107	当通过NC_Reset 或者NC_Emergency 指令复位时, NC_CuttingFeedOverride 指令无法执行.	在NC_Reset或者NC_Emergency 指令结束后执行NC指令.
350A	13578	当通过NC_Reset 或者NC_Emergency 指令复位时, NC_CycleStart 指令无法执行.	在NC_Reset或者NC_Emergency 指令结束后执行NC指令.
3340	13120	NC主轴覆盖指令覆盖因素超出范围.	设置覆盖指令VelFactor, AccFactor和JerkFactor值大于0后执行覆盖指令
3341	13121	出现NC主轴覆盖指令的覆盖因素后, 主轴运行速度值超出速度限制值.	查看连接到主轴的轴速度限制值后, 在不超出速度限制范围内执行覆盖.
3342	13122	当通过NC_Reset 或者NC_Emergency 指令复位时, NC_SpindleOverride 指令无法执行.	如果需要NC主轴覆盖功能, 在NC_Reset或者NC_Emergency 指令结束后再次执行指令.
3343	13123	当通过NC_Reset 或者NC_Emergency 指令复位时, NC_SpindleOverride 指令无法执行.	在NC_Reset或者NC_Emergency 指令结束后执行NC指令.
3350	13136	NC 参数读取指令的设置轴无法作为NC轴.	查看是否NC参数读取指令的设置轴注册为NC通道/轴参数. NC通道/轴可以注册为MP500动作数据项目中的NC通道参数.
3351	13137	NC参数读取指令轴设置值超出允许范围.	设置轴数量值在0和10之间. 如果轴值为0, 执行通道参数读取如果为1~10, 读取NC轴X ~ NC轴S.
3352	13138	NC 参数读取指令参数组设置值超出允许范围.	通道参数的参数组设置范围为1 ~ 17, 通道/轴参数的参数组设置范围为1 ~ 5. 查看想要读取的组数量, 然后执行参数读取指令(NC_Read参数).
3353	13139	不支持设置与NC参数读取指令的参数组参数数量.	查看是否支持设置于通道参数或者通道/轴参数的参数数量. 查看想要读取的参数的组数量和参数数量, 然后执行参数读取指令 (NC_Read参数).
3500	13568	如果在自动运行中取消NC通道, 自动运行启动运行无法执行.	查看是否NC通道当前为自动运行. 自动运行完成后自动运行可以重启.
3501	13569	由于NC进给保持指令为使能状态, 自动运行启动运行无法执行.	发布NC馈送保持指令 (NC_FeedHold)的使能输入后, 再次执行自动运行启动指令 (NC_CycleStart).

附录 2 错误信息&解决方法

错误代码		错误描述	解决方法
十六进制	十进制		
3502	13570	由于NC紧急停止指令为使能状态, 自动运行启动运行无法执行.	发布NC紧急停止指令 (NC_Emergency)的使能输入后, 再次执行自动运行启动指令 (NC_CycleStart).
3503	13571	当NC注释(IPR)没有正常终止, 自动运行启动运行无法执行.	通过NC复位指令(NC_Reset)复位NC通道后, 再次执行自动运行启动指令 (NC_CycleStart).
3504	13572	由于 NC注释(IPR)或者程序处理器(PA)为错误状态, 自动运行启动运行无法执行.	查看NC通道标志位中的注释(IPR)和程序处理器(PA)错误代码. 通过NC复位指令(NC_Reset)复位NC通道后, 再次执行自动运行启动指令 (NC_CycleStart).
3505	13573	由于程序执行的NC通道没有设置, 自动运行启动运行无法执行.	通过NC程序指定指令指令(NC_LoadProgram)指定程序执行后, 再次执行自动运行启动指令 (NC_CycleStart).
3506	13574	由于NC通道到达目标处理数量或者在M99重复处理中的处理数量, 自动运行启动运行无法执行.	确定NC通道标志位进程数量或者M99重复进程的进程数量, 然后查看是否到达目标进程数量. 复位进程数量或者M99重复进程中的进程数量标志位后, 再次执行自动运行启动指令 (NC_CycleStart).
3507	13575	如果NC M/S/T-代码输出选通信号为on, 自动运行启动运行无法执行.	完成NC M/S/T-代码输出选通信号后, 执行自动运行启动指令 (NC_CycleStart).
3508	13576	NC通道注释(IPR)非正常执行.	通过NC复位指令(NC_Reset)复位NC通道后, 再次执行自动运行启动指令 (NC_CycleStart).
3509	13577	当进入NC移位禁止区域时, 自动运行启动运行无法执行.	通过在错误发生的NC配置轴使用点动指令, 从移位禁止区域范围进行分离, 然后通过执行错误复位指令移除错误.
350A	13578	当通过NC_Reset或者NC_Emergency指令复位时, NC_CycleStart 指令无法执行.	NC_Reset或者NC_Emergency指令结束后执行NC指令.
3510	13584	由于 NC通道配置轴未准备, 自动运行启动运行无法执行.	当NC通道配置轴全部准备时, 执行自动运行启动指令 (NC_CycleStart). 为了启动自动运行, NC通道配置轴应连接到 EtherCAT网络或者设置为虚轴.
3511	13585	由于NC X轴未准备, 自动运行启动运行无法执行.	当NC通道配置轴全部准备时, 执行自动运行启动指令 (NC_CycleStart). 为了启动自动运行, NC通道配置轴应连接到 EtherCAT网络或者设置为虚轴.

错误代码		错误描述	解决方法
十六进制	十进制		
3512	13586	由于NC Y轴未准备, 自动运行启动运行无法执行.	当NC通道配置轴全部准备时, 执行自动运行启动指令 (NC_CycleStart). 为了启动自动运行, NC通道配置轴应连接到 EtherCAT 网络或者设置为虚轴.
3513	13587	由于NC Z轴未准备, 自动运行启动运行无法执行.	当NC通道配置轴全部准备时, 执行自动运行启动指令 (NC_CycleStart). 为了启动自动运行, NC通道配置轴应连接到 EtherCAT 网络或者设置为虚轴.
3514	13588	由于NC A轴未准备, 自动运行启动运行无法执行.	当NC通道配置轴全部准备时, 执行自动运行启动指令 (NC_CycleStart). 为了启动自动运行, NC通道配置轴应连接到 EtherCAT 网络或者设置为虚轴.
3515	13589	由于NC B轴未准备, 自动运行启动运行无法执行.	当NC通道配置轴全部准备时, 执行自动运行启动指令 (NC_CycleStart). 为了启动自动运行, NC通道配置轴应连接到 EtherCAT 网络或者设置为虚轴.
3516	13590	由于NC C轴未准备, 自动运行启动运行无法执行.	当NC通道配置轴全部准备时, 执行自动运行启动指令 (NC_CycleStart). 为了启动自动运行, NC通道配置轴应连接到 EtherCAT 网络或者设置为虚轴.
3517	13591	由于NC U轴未准备, 自动运行启动运行无法执行.	当NC通道配置轴全部准备时, 执行自动运行启动指令 (NC_CycleStart). 为了启动自动运行, NC通道配置轴应连接到 EtherCAT 网络或者设置为虚轴.
3518	13592	由于NC V轴未准备, 自动运行启动运行无法执行.	当NC通道配置轴全部准备时, 执行自动运行启动指令 (NC_CycleStart). 为了启动自动运行, NC通道配置轴应连接到 EtherCAT 网络或者设置为虚轴.
3519	13593	由于NC W轴未准备, 自动运行启动运行无法执行.	当NC通道配置轴全部准备时, 执行自动运行启动指令 (NC_CycleStart). 为了启动自动运行, NC通道配置轴应连接到 EtherCAT 网络或者设置为虚轴.
351A	13594	由于NC S轴未准备, 自动运行启动运行无法执行.	当NC通道配置轴全部准备时, 执行自动运行启动指令 (NC_CycleStart). 为了启动自动运行, NC通道配置轴应连接到 EtherCAT 网络或者设置为虚轴.

附录 2 错误信息&解决方法

错误代码		错误描述	解决方法
十六进制	十进制		
3520	13600	由于NC通道配置轴在运行中, 自动运行启动指令无法执行.	当NC通道配置轴停止时, 执行自动运行启动(NC_CycleStart).
3521	13601	由于NC X轴在运行中, 自动运行启动指令无法执行.	当NC通道配置轴停止时, 执行自动运行启动(NC_CycleStart).
3522	13602	由于NC Y轴在运行中, 自动运行启动指令无法执行.	当NC通道配置轴停止时, 执行自动运行启动(NC_CycleStart).
3523	13603	由于NC Z轴在运行中, 自动运行启动指令无法执行.	当NC通道配置轴停止时, 执行自动运行启动(NC_CycleStart).
3524	13604	由于NC A轴在运行中, 自动运行启动指令无法执行.	当NC通道配置轴停止时, 执行自动运行启动(NC_CycleStart).
3525	13605	由于NC B轴在运行中, 自动运行启动指令无法执行.	当NC通道配置轴停止时, 执行自动运行启动(NC_CycleStart).
3526	13606	由于NC C轴在运行中, 自动运行启动指令无法执行.	当NC通道配置轴停止时, 执行自动运行启动(NC_CycleStart).
3527	13607	由于NC U轴在运行中, 自动运行启动指令无法执行.	当NC通道配置轴停止时, 执行自动运行启动(NC_CycleStart).
3528	13608	由于NC V轴在运行中, 自动运行启动指令无法执行.	当NC通道配置轴停止时, 执行自动运行启动(NC_CycleStart).
3529	13609	由于NC W轴在运行中, 自动运行启动指令无法执行.	当NC通道配置轴停止时, 执行自动运行启动(NC_CycleStart).
3530	13616	由于NC通道配置轴作为动作轴组配置轴使能, 自动运行启动指令无法执行.	作为动作轴组配置轴, 通过禁止的NC通道配置轴执行自动运行启动 (NC_CycleStart).
3531	13617	由于NC X轴作为动作轴组配置轴使能, 自动运行启动指令无法执行.	作为动作轴组配置轴, 通过禁止的NC通道配置轴执行自动运行启动 (NC_CycleStart).
3532	13618	由于NC Y轴作为动作轴组配置轴使能, 自动运行启动指令无法执行.	作为动作轴组配置轴, 通过禁止的NC通道配置轴执行自动运行启动 (NC_CycleStart).
3533	13619	由于NC Z轴作为动作轴组配置轴使能, 自动运行启动指令无法执行.	作为动作轴组配置轴, 通过禁止的NC通道配置轴执行自动运行启动 (NC_CycleStart).
3534	13620	由于NC A轴作为动作轴组配置轴使能, 自动运行启动指令无法执行.	作为动作轴组配置轴, 通过禁止的NC通道配置轴执行自动运行启动 (NC_CycleStart).
3535	13621	由于NC B轴作为动作轴组配置轴使能, 自动运行启动指令无法执行.	作为动作轴组配置轴, 通过禁止的NC通道配置轴执行自动运行启动 (NC_CycleStart).
3536	13622	由于NC C轴作为动作轴组配置轴使能, 自动运行启动指令无法执行.	作为动作轴组配置轴, 通过禁止的NC通道配置轴执行自动运行启动 (NC_CycleStart).
3537	13623	由于NC U轴作为动作轴组配置轴使能, 自动运行启动指令无法执行.	作为动作轴组配置轴, 通过禁止的NC通道配置轴执行自动运行启动 (NC_CycleStart).

错误代码		错误描述	解决方法
十六进制	十进制		
3538	13624	由于NC V轴作为动作轴组配置轴使能, 自动运行启动指令无法执行.	作为动作轴组配置轴, 通过禁止的NC通道配置轴执行自动运行启动 (NC_CycleStart).
3539	13625	由于NC W轴作为动作轴组配置轴使能, 自动运行启动指令无法执行.	作为动作轴组配置轴, 通过禁止的NC通道配置轴执行自动运行启动 (NC_CycleStart).
353A	13626	由于NC S轴作为动作轴组配置轴使能, 自动运行启动指令无法执行.	作为动作轴组配置轴, 通过禁止的NC通道配置轴执行自动运行启动 (NC_CycleStart).
3540	13632	NC通道配置轴位置单位或者速度单位设置无效.	对于NC运行, 设置NC通道配置轴 (除主轴以外)单位为mm或者deg. 设置主轴(S轴)速度单位为RPM, 其他轴unit/min.
3541	13633	NC X轴位置单位或者速度单位设置无效.	对于NC运行, 设置NC通道配置轴单位为mm或者deg. 设置速度单位为unit/min.
3542	13634	NC Y轴位置单位或者速度单位设置无效.	对于NC运行, 设置NC通道配置轴单位为mm或者deg. 设置速度单位为unit/min.
3543	13635	NC Z轴位置单位或者速度单位设置无效.	对于NC运行, 设置NC通道配置轴单位为mm或者deg. 设置速度单位为unit/min.
3544	13636	NC A轴位置单位或者速度单位设置无效.	对于NC运行, 设置NC通道配置轴单位为mm或者deg. 设置速度单位为unit/min.
3545	13637	NC B轴位置单位或者速度单位设置无效.	对于NC运行, 设置NC通道配置轴单位为mm或者deg. 设置速度单位为unit/min.
3546	13638	NC C轴位置单位或者速度单位设置无效.	对于NC运行, 设置NC通道配置轴单位为mm或者deg. 设置速度单位为unit/min.
3547	13639	NC U轴位置单位或者速度单位设置无效.	对于NC运行, 设置NC通道配置轴单位为mm或者deg. 设置速度单位为unit/min.
3548	13640	NC V轴位置单位或者速度单位设置无效.	对于NC运行, 设置NC通道配置轴单位为mm或者deg. 设置速度单位为unit/min.
3549	13641	NC W轴位置单位或者速度单位设置无效.	对于NC运行, 设置NC通道配置轴单位为mm或者deg. 设置速度单位为unit/min.
354A	13642	NC S轴位置单位或者速度单位设置无效.	对于NC运行, 设置主轴速度单位为RPM.
3600	13824	由于在控制器中的NC程序指定指令没有程序设置, 程序无法加载.	写入创建于MP500控制器中的NC程序后, 再次执行程序指定指令(NC_LoadProgram).
3601	13825	如果NC通道为自动运行, 无法执行程序指定指令.	查看是否NC通道当前为自动运行. 可以在自动运行结束后指定新程序.
3602	13826	由于NC程序数据异常, 无法执行程序指定指令.	查看NC程序数据异常错误(0x3001)发生后再次从MP500下载数据, 如果重启后错误仍然存在请联系售后部门.
3603	13827	NC程序指定指令Load模式无效.	在NC程序指定指令(NC_LoadProgram)Load模式中输入0值之后, 再次执行程序指定指令(NC_LoadProgram).

附录 2 错误信息&解决方法

错误代码		错误描述	解决方法
十六进制	十进制		
3604	13828	当通过NC_Reset或者NC_Emergency 指令复位时, NC_LoadProgram指令无法执行.	NC_Reset 或者NC_Emergency 指令结束后执行NC指令.
3610	13840	NC通道注释(IPR)未正常复位.	通过NC复位指令(NC_Reset)复位NC通道. 如果重启后错误仍然存在, 请联系售后部门.
3620	13856	NC_Emergency 指令在自动运行以外的状态下执行.	当NC通道为自动运行时, 执行NC紧急停止指令(NC_Emergency).
3630	13872	不是在NC原点返回运行中指定的原点范围内.	原点范围(ReferenceNum)从第一个原点到第四个原点. 指定1 和4之间的值.
3631	13873	当通道在自动运行时, NC原点返回指令无法执行.	自动运行完成后执行原点返回指令.
3632	13874	由于NC紧急停止指令在使能状态, 原点返回指令无法执行.	发布NC紧急停止指令(NC_Emergency)使能输入后再次执行原点返回指令(NC_Home).
3633	13875	由于NC通道配置轴未准备, 原点返回指令无法执行.	执行所有NC通道配置轴准备的原点返回指令(NC_Home). 为了执行原点返回指令, NC通道配置轴应连接到EtherCAT网络, 或者设置为虚轴.
3634	13876	由于NC通道配置轴作为动作轴组配置轴使能时, 原点返回指令无法执行.	作为动作轴组配置轴, 通过禁止的NC通道配置轴执行原点返回指令(NC_Home).
3635	13877	伺服驱动中原点返回期间发生错误.	移除伺服驱动错误后, 查看伺服驱动错误原因并执行原点返回.
3636	13878	当通过NC_Reset或者NC_Emergency 指令复位时, 无法执行NC_Home 指令.	NC_Reset或者NC_Emergency指令结束后执行NC指令.
3640	13888	当M-代码输出选通信号为off时, NC M-代码运行完成指令无法执行.	查看NC通道标志位中M-代码输出选通信号状态后, 通过M-代码输出选通信号on执行M-代码运行完成指令 (NC_McodeComplete).
3641	13889	当通过NC_Reset或者NC_Emergency 指令复位时, NC_McodeComplete 指令无法执行.	NC_Reset或者NC_Emergency指令结束后执行NC指令.
3650	13904	当S-代码输出选通信号为off时, NC S-代码运行完成指令无法执行.	查看NC通道标志位中S-代码输出选通信号状态后, 通过M-代码输出选通信号on执行S-代码运行完成指令 (NC_McodeComplete).
3651	13905	当通过NC_Reset或者NC_Emergency 指令复位时, NC_ScodeComplete 指令无法执行.	NC_Reset或者NC_Emergency指令结束后执行NC指令.
3660	13920	当T-代码输出选通信号为off时, NC T-代码运行完成指令无法执行.	查看NC通道标志位中T-代码输出选通信号状态后, 通过M-代码输出选通信号on执行T-代码运行完成指令 (NC_McodeComplete).

错误代码		错误描述	解决方法
十六进制	十进制		
3661	13921	当通过NC_Reset或者NC_Emergency 指令复位时, NC_TcodeComplete 指令无法执行.	NC_Reset或者NC_Emergency指令结束后执行NC指令.
3670	13936	当通道为自动运行时, NC 参数写入指令无法执行	查看NC通道当前是否为自动运行C. 结束自动运行后执行停止状态的NC 参数写入指令(NC_写入参数).
3671	13937	NC 参数写入指令设置轴无法作为NC轴使能.	查看是否NC参数写入指令的设置轴注册为NC通道/轴参数. NC通道/轴可以在MP500的动作数据项目的NC通道参数中注册.
3672	13938	NC 参数写入指令轴设置值超出允许范围.	设置轴数量值范围在0和10之间. 如果轴值为0, 执行通道参数书写入, 如果轴值为1~10, 执行NC轴X ~ NC 轴S写入.
3673	13939	NC 参数写入指令参数组设置值超出允许范围.	通道参数的参数组设置范围为1 ~ 17, 而通道/轴参数的设置范围为1 ~ 5. 查看希望写入的参数组编号, 然后执行参数写入指令(NC_写入参数).
3674	13940	不支持以NC参数写入指令设置的参数组参数数量.	查看设置与通道参数或者通道/轴参数组中的参数编码是否支持. 查看希望写入的组编号和参数编号, 然后执行参数写入指令(NC_写入参数).
3675	13941	以NC参数写入指令的参数数据设置值超出范围.	查看参数数据设置范围, 然后通过范围内值执行参数写入指令(NC_写入参数).
3800	14336	在NC快速移位指令中, 配置轴运行速度值超出速度限制值.	查看连接到发生错误NC配置轴的轴速度限制值之后, 以不超出速度限制值范围内的值设置快速移位比率.
3801	14337	在NC快速移位指令中, NC X 轴运行速度值超出速度限制值.	查看连接到发生错误NC X轴的轴速度限制值之后, 以不超出速度限制值范围内的值设置快速移位比率.
3802	14338	在NC快速移位指令中, NC Y 轴运行速度值超出速度限制值.	查看连接到发生错误NC Y轴的轴速度限制值之后, 以不超出速度限制值范围内的值设置快速移位比率.
3803	14339	在NC快速移位指令中, NC Z轴运行速度值超出速度限制值.	查看连接到发生错误NC Z轴的轴速度限制值之后, 以不超出速度限制值范围内的值设置快速移位比率.
3804	14340	在NC快速移位指令中, NC A轴运行速度值超出速度限制值.	查看连接到发生错误NC A轴的轴速度限制值之后, 以不超出速度限制值范围内的值设置快速移位比率.
3805	14341	在NC快速移位指令中, NC B轴运行速度值超出速度限制值.	查看连接到发生错误NC B轴的轴速度限制值之后, 以不超出速度限制值范围内的值设置快速移位比率.

附录 2 错误信息&解决方法

错误代码		错误描述	解决方法
十六进制	十进制		
3806	14342	在NC快速移位指令中, NC C轴运行速度值超出速度限制值.	查看连接到发生错误NC C轴的轴速度限制值之后, 以不超出速度限制值范围内的值设置快速移位比率.
3807	14343	在NC快速移位指令中, NC U轴运行速度值超出速度限制值.	查看连接到发生错误NC U轴的轴速度限制值之后, 以不超出速度限制值范围内的值设置快速移位比率.
3808	14344	在NC快速移位指令中, NC V轴运行速度值超出速度限制值.	查看连接到发生错误NC V轴的轴速度限制值之后, 以不超出速度限制值范围内的值设置快速移位比率.
3809	14345	在NC快速移位指令中, NC W轴运行速度值超出速度限制值.	查看连接到发生错误NC W轴的轴速度限制值之后, 以不超出速度限制值范围内的值设置快速移位比率.
380A	14346	在NC快速移位指令中, NC S轴运行速度值超出速度限制值.	查看连接到发生错误NC S轴的轴速度限制值之后, 以不超出速度限制值范围内的值设置快速移位比率.
380B	14347	NC快速移位运行中, 快速移位组合轴到达位置未在到达位置完成监控时间内容完成.	查看连接到NC轴参数中NC通道参数的指令到达位置宽度, 以及到达位置完成监控时间.
380C	14348	NC快速移位运行中, 由于NC配置轴中有一个轴有错误, 自动运行无法持续.	查看NC配置轴中错误发生轴. 可以查看以NC通道/轴标志位中轴错误代码发生在NC轴的错误.
3810	14352	在每旋转NC馈送模式中, 进刀运行比率设置为0.	在NC每旋转馈送模式中设置进刀运行比率值大于0.
3811	14353	NC进刀指令运行速度超出NC 通道参数的进刀上限比率值.	查看NC通道参数进刀上限比率值之后, 设置不超过参数值范围的进刀比率值(F).
3812	14354	NC进刀运行时, 进刀配置轴到达位置未在到达位置完成监控时间内完成.	查看连接到NC轴参数中NC通道参数的指令到达位置宽度, 以及到达位置完成监控时间.
3820	14368	在NC进刀指令, 配置轴运行速度值超出速度限制值.	查看连接到发生错误NC 配置轴的轴速度限制值之后, 以不超出速度限制值范围内的值设置进刀比率.
3821	14369	在NC进刀指令, NC X轴运行速度值超出速度限制值.	查看连接到发生错误NC X轴的轴速度限制值之后, 以不超出速度限制值范围内的值设置进刀比率.
3822	14370	在NC进刀指令, NC Y轴运行速度值超出速度限制值.	查看连接到发生错误NC Y轴的轴速度限制值之后, 以不超出速度限制值范围内的值设置进刀比率.
3823	14371	在NC进刀指令, NC Z轴运行速度值超出速度限制值.	查看连接到发生错误NC Z轴的轴速度限制值之后, 以不超出速度限制值范围内的值设置进刀比率.

错误代码		错误描述	解决方法
十六进制	十进制		
3824	14372	在NC进刀指令, NC A轴运行速度值超出速度限制值.	查看连接到发生错误NC A轴的轴速度限制值之后, 以不超出速度限制值范围内的值设置进刀比率.
3825	14373	在NC进刀指令, NC B轴运行速度值超出速度限制值.	查看连接到发生错误NC B轴的轴速度限制值之后, 以不超出速度限制值范围内的值设置进刀比率.
3826	14374	在NC进刀指令, NC C轴运行速度值超出速度限制值.	查看连接到发生错误NC C轴的轴速度限制值之后, 以不超出速度限制值范围内的值设置进刀比率.
3827	14375	在NC进刀指令, NC U轴运行速度值超出速度限制值.	查看连接到发生错误NC U轴的轴速度限制值之后, 以不超出速度限制值范围内的值设置进刀比率.
3828	14376	在NC进刀指令, NC V轴运行速度值超出速度限制值.	查看连接到发生错误NC V轴的轴速度限制值之后, 以不超出速度限制值范围内的值设置进刀比率.
3829	14377	在NC进刀指令, NC W轴运行速度值超出速度限制值.	查看连接到发生错误NC W轴的轴速度限制值之后, 以不超出速度限制值范围内的值设置进刀比率.
382A	14378	在NC进刀指令, NC S轴运行速度值超出速度限制值.	查看连接到发生错误NC S轴的轴速度限制值之后, 以不超出速度限制值范围内的值设置进刀比率.
3840	14400	在NC圆弧插值中,对于起始点= 中心点或者中心点 = 结束点的情况, 运行无法执行.	在NC圆弧插值中, 设置与启动点(或者结束点)位置不同的中心点位置.
3841	14401	在NC圆弧插值中半径设置不正确.	可执行的NC圆弧插值圆弧半径值大于0, 且小于等于基于脉冲单位的2,147,483,647脉冲. 设置中心点或者半径输入值, 以便可以在设置范围内计算半径.
3850	14416	在NC圆柱插值中, 轴指定错误.	在NC圆柱插值执行圆弧插值运行中, Y-轴应指定于XY 平面, Z-轴指定于YZ 平面, Z-轴指定于ZX 平面.
3860	14432	静止方式指定为旋转数量, 但是旋转数量为0.	以NC程序中的MC_MoveVelocity 运行S-轴.
3F00	16128	注释(IPR)分析错误 - LEX MAIN TABLE 配置无效.	确认NC通道标志位中'错误块编号'之后, 查看在块中是否有程序错误. 通过NC复位指令(NC_Reset)进行NC通道复位后执行自动运行.
3F01	16129	注释(IPR)分析错误—未定义字符存在.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误. 通过NC复位指令NC_Reset复位后再次执行自动运行
3F02	16130	注释(IPR)分析错误—数量超出最大缓存.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误. 通过NC复位指令NC_Reset复位后再次执行自动运行

附录 2 错误信息&解决方法

错误代码		错误描述	解决方法
十六进制	十进制		
3F03	16131	注释(IPR)分析错误 - LEX符号数量超出最大缓存.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F04	16132	注释(IPR)分析错误—有1个及以上小数点.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F05	16133	注释(IPR)分析错误—公式中括号数量错误.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F06	16134	注释(IPR)分析错误—公式中当前字符无法使用.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F07	16135	注释(IPR)分析错误—公式语法错误.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F08	16136	注释(IPR)分析错误—不被允许的宏变量.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F09	16137	注释(IPR)分析错误—TANGENT运行错误.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F0A	16138	注释(IPR)分析错误 - SQUARE ROOT运行错误.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F0B	16139	注释(IPR)分析错误—除法分母不可以为0.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F0C	16140	注释(IPR)分析错误—语法错误.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F0D	16141	注释(IPR)分析错误 - YACC MAIN TABLE 配置无效.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行

错误代码		错误描述	解决方法
十六进制	十进制		
3F0E	16142	注释(IPR)分析错误 - YACC符号数量超出最大缓存.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F0F	16143	注释(IPR)分析错误-无法打开IPR信号.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F10	16144	注释(IPR)分析错误-不通过M02或者M30停止.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F11	16145	注释(IPR)分析错误-仅在块头部执行.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F12	16146	注释(IPR)分析错误-存在相同进程块.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F13	16147	注释(IPR)分析错误 -状态数量超出最大缓存.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F14	16148	注释(IPR)分析错误-无法找到下一个进程的块.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F15	16149	注释(IPR)分析错误-子程序调用语法错误.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F16	16150	注释(IPR)分析错误 -超出最大子程序调用.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F17	16151	注释(IPR)分析错误-这是一个已被调用的程序.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F18	16152	注释(IPR)分析错误-在子程序中沒有M99.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行

附录 2 错误信息&解决方法

错误代码		错误描述	解决方法
十六进制	十进制		
3F19	16153	注释(IPR)分析错误 - M99语法错误.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F1A	16154	注释(IPR)分析错误—有较大数量的环.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F1B	16155	注释(IPR)分析错误—没有起始环.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F1C	16156	注释(IPR)分析错误—环连接无效.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F1D	16157	注释(IPR)分析错误 - 超出一个块中 M指令限制.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F1E	16158	注释(IPR)分析错误—未使用G代码.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F1F	16159	注释(IPR)分析错误—无法在一个块中同时执行.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F20	16160	注释(IPR)分析错误 - 圆弧中心点无法找到.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F21	16161	注释(IPR)分析错误—无法创建循环代码路径.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F22	16162	注释(IPR)分析错误—循环代码锥化量太大.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F23	16163	注释(IPR)分析错误—无法在循环形状块中执行.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行

错误代码		错误描述	解决方法
十六进制	十进制		
3F24	16164	注释(IPR)分析错误—在循环形状块指令中有问题.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F26	16166	注释(IPR)分析错误—工具补偿数量无效.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F27	16167	注释(IPR)分析错误—圆弧中心点错误.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F28	16168	注释(IPR)分析错误—无法在MDI模式下调用子程序.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F29	16169	注释(IPR)分析错误—倒角和圆形加工仅在进刀指令中应用.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F2A	16170	注释(IPR)分析错误 - 倒角和圆形加工在指令中重复应用.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F2B	16171	注释(IPR)分析错误—仅单个轴指令应用于倒角和圆形加工.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F2C	16172	注释(IPR)分析错误 - 倒角和圆形加工参考值大于馈送数量.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F2D	16173	注释(IPR)分析错误—在倒角和圆形加工中无法获得下一个块信息.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F2E	16174	注释(IPR)分析错误—下一个块中的圆弧不允许在倒角和圆形加工中.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F2F	16175	注释(IPR)分析错误—在相同直线馈送中无法执行圆形加工.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行

附录 2 错误信息&解决方法

错误代码		错误描述	解决方法
十六进制	十进制		
3F30	16176	注释(IPR)分析错误—仅在线性馈送中修正启动和结束.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F31	16177	注释(IPR)分析错误—在循环外形结束块中没有馈送指令.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F32	16178	注释(IPR)分析错误—在倒角和圆形加工中有与平面不相干的轴指令.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F33	16179	注释(IPR)分析错误—在调用宏中一个块超出IJK指令限制.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F34	16180	注释(IPR)分析错误—模式宏无法从子程序中调用.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F35	16181	注释(IPR)分析错误 -超出模式宏多重调用限制.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F36	16182	注释(IPR)分析错误—未使用M代码.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F37	16183	注释(IPR)分析错误—在刚性攻丝中无法计算螺距.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F38	16184	注释(IPR)分析错误—字符串超出最大缓存.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F39	16185	注释(IPR)分析错误—字符串语法错误.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F3A	16186	注释(IPR)分析错误—到达目标进程数量.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行

错误代码		错误描述	解决方法
十六进制	十进制		
3F3B	16187	注释(IPR)分析错误—宏程序用户停止.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F3C	16188	注释(IPR)分析错误—无法创建用于符合螺距循环的路径.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F3D	16189	注释(IPR)分析错误—在极坐标插值中无法执行.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F3E	16190	注释(IPR)分析错误—在极坐标超值中无法移位到0.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F3F	16191	注释(IPR)分析错误—在圆柱插值指令中语法错误.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F40	16192	注释(IPR)分析错误—在圆柱插值期间无法执行.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F41	16193	注释(IPR)分析错误—在极坐标和圆柱插值的常数表面速度控制模式.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F42	16194	注释(IPR)分析错误—不是原点.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3F43	16195	注释(IPR)分析错误—发生工具干扰.	确认NC通道标志位中'错误块数量'后, 查看在块中是否有程序错误。通过NC复位指令NC_Reset复位后再次执行自动运行
3FE0	16352	程序处理器(PA) 错误—没有程序文件的对应指针位置.	通过NC复位指令(NC_Reset)复位NC通道后再次执行自动运行.
3FE1	16353	程序处理器(PA) 错误—无法从程序文件中读取.	通过NC复位指令(NC_Reset)复位NC通道后再次执行自动运行.
3FE2	16354	程序处理器(PA) 错误—选择程序文件不存在.	查看是否指定程序保存在控制器中. 通过NC复位指令(NC_Reset)复位NC通道后再次执行自动运行.
3FE3	16355	程序处理器(PA) 错误—无法打开NcAccess信号.	通过NC复位指令(NC_Reset)复位NC通道后再次执行自动运行.
3FE4	16356	程序处理器(PA) 错误 – 每个块的字符数量限制为300.	查看是否指定程序每个块字符数量是否超出300C. 通过NC复位指令(NC_Reset)复位NC通道后再次执行自动运行.

附录 3设置示例

开始时使用运动控制器的设置方法.

(1) 安装伺服驱动器.

伺服驱动器连接电源线和马达线, 根据需要连接外部信号.

(2) 安装运动控制器.

安装运动控制器. 一开始试运行, 为了安全, 确保运动控制器在停止模式.

(3) 连接运动控制器和伺服驱动器.

通过Ethernet电缆连接运动控制器和第一台伺服驱动器, 再连接到另外一台伺服驱动器上.

这时要检查伺服驱动器通讯端口的I/O方向. 下面是包含伺服驱动器和EtherCAT IO连接到运动控制器的网络设置信息的伺服运动器列表.



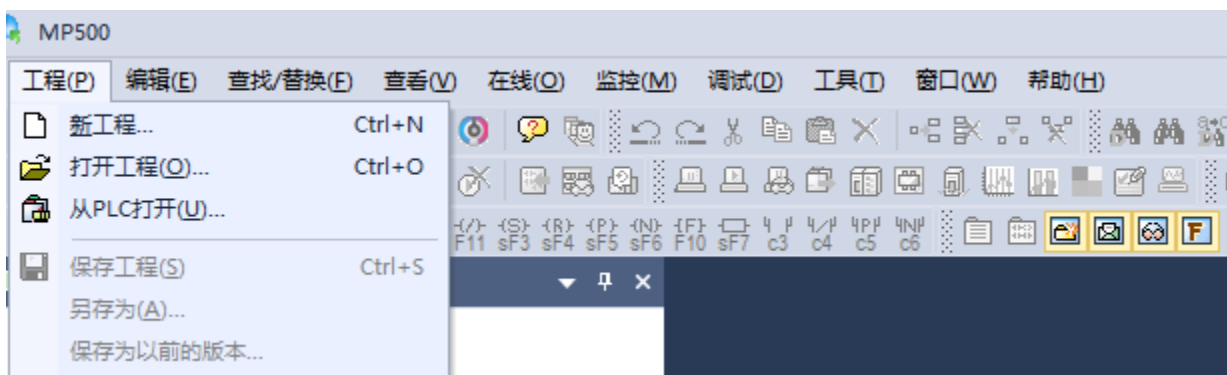
备注

伺服驱动器安装好后,通过伺服驱动器厂商提供的专用设置工具确保以下几点; 如果标准设置失败, 需要复位重新设置满足用户实际情况.

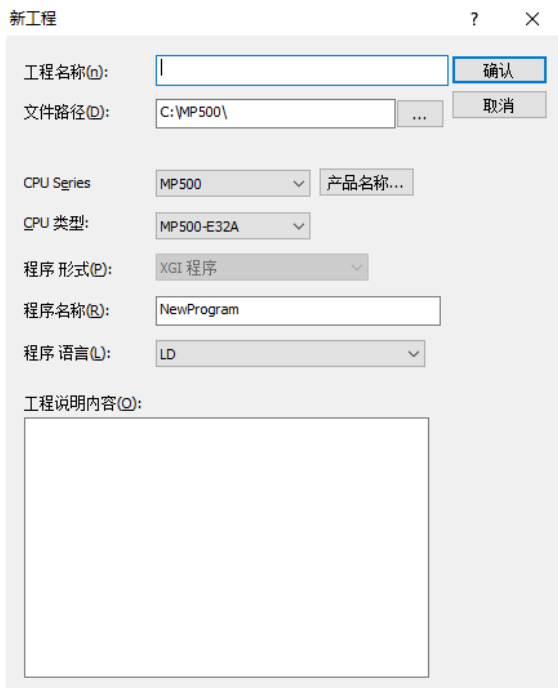
1. 电源供应
检查伺服驱动器电源连接和电源情况.
(根据伺服驱动器类型有些情况不需要电源设置参数.)
2. 马达和编码器类型 (反馈)
(根据连接到伺服的编码器和马达的类型设置参数.)
3. 指令位置单位设定
通过伺服驱动器参数设定指令位置单位, 确保设置成脉冲单位(Inc. 或者 Counts), 根据使用的编码器位数设置每个电机旋转的编码器分辨率.
(根据伺服驱动器类型有些情况没有单独的设置项目.)

(4) 电脑安装MP500软件.

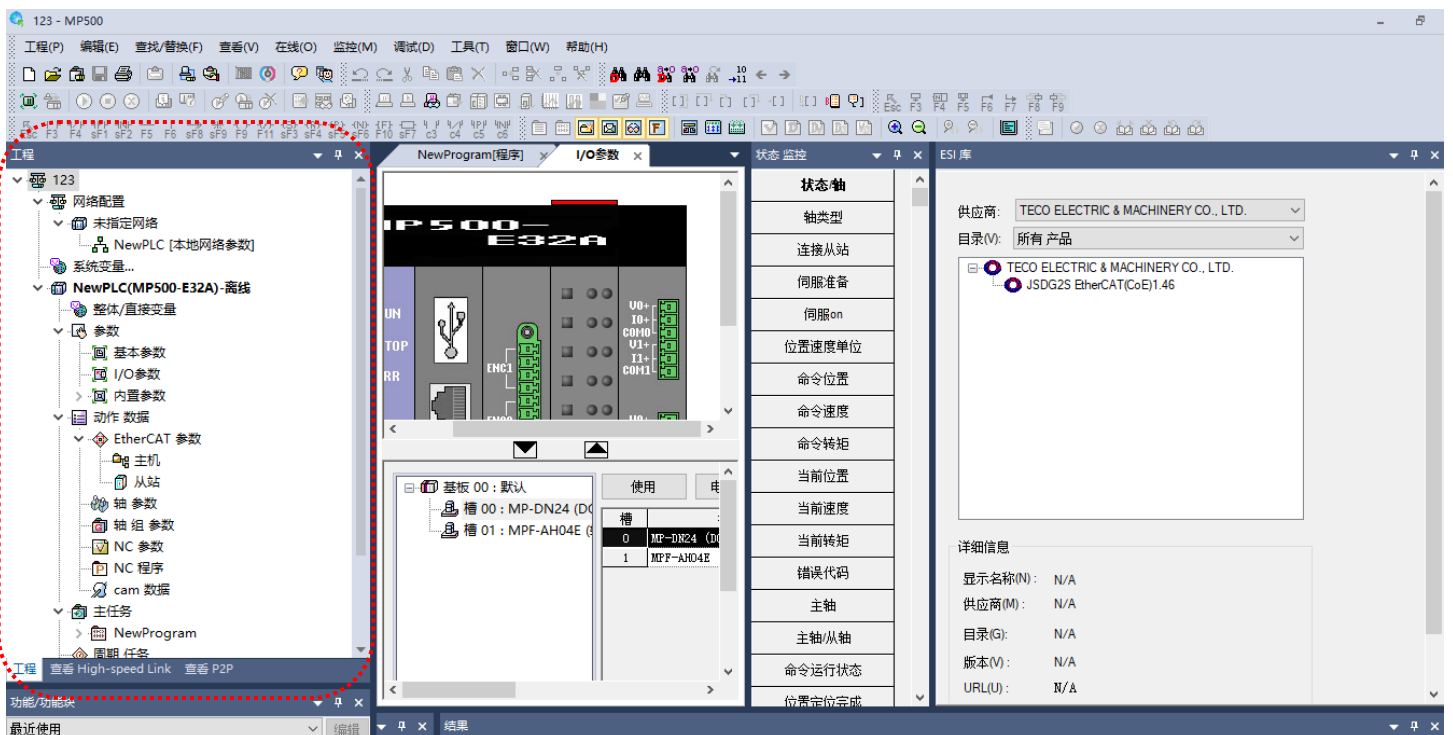
(5) 运行MP500, 通过选择 “Project(P) – New Project(N)”新建马达控制工程.



(6) 下图中，输入工程名称，CPU系列，CPU 类型，程序名称来新建一个工程。



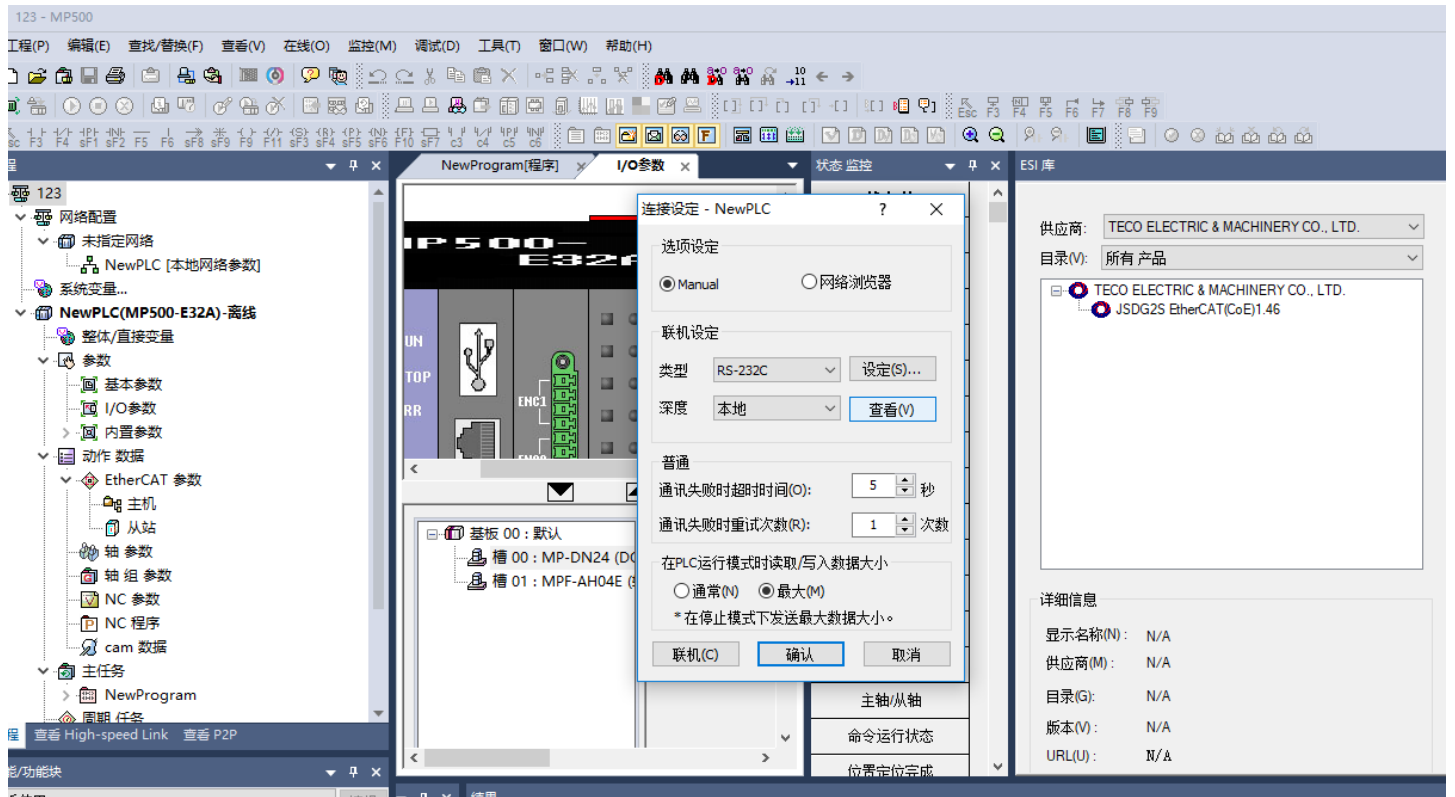
(7) 如上图设置的话，工程生成后如下所示。



(8) 打开运动控制器，伺服驱动器，通过USB和Ethernet电缆连接个人电脑和运动控制器。

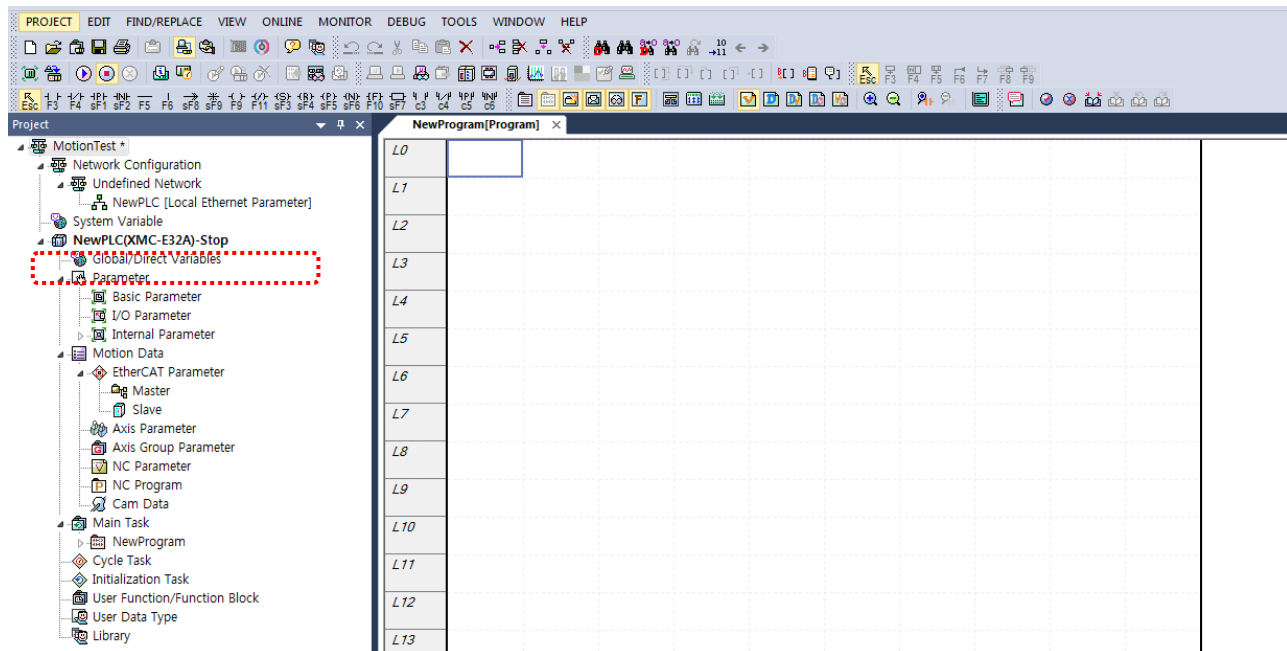
附录 3 设置示例

(9) 选择“在线(O)- 连接设置(O)”， 建立设置连接。



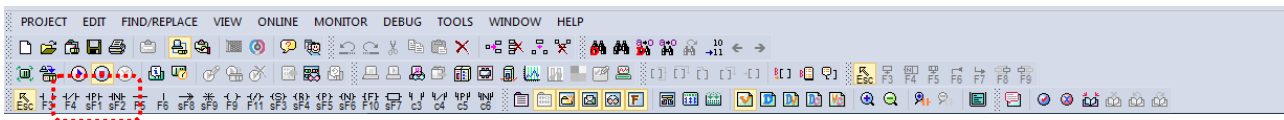
(10) 选择“在线(O)-连接(N)” 连接电脑和运动控制器。

(11)连接完成，控制器会显示‘Run’或者‘Stop’如下。



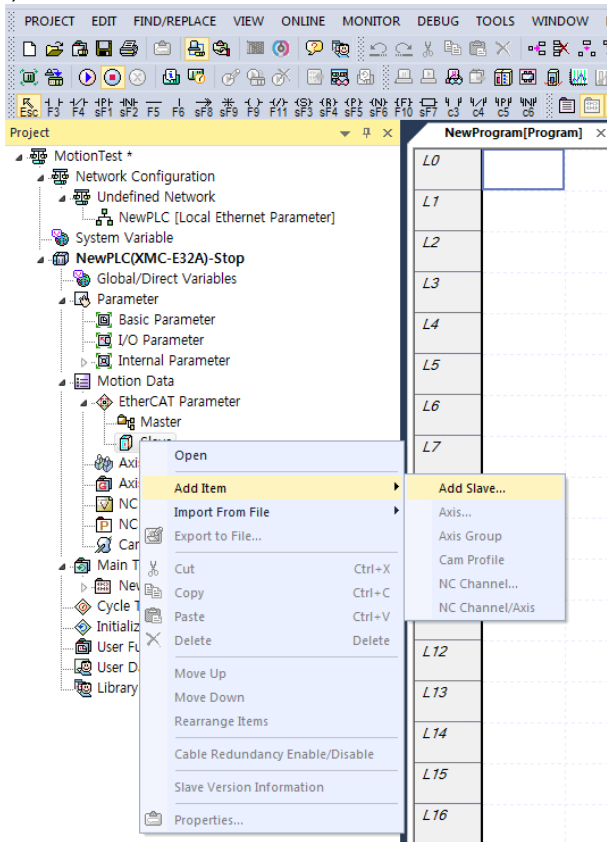
(12) 如果控制器没有变成“Online” 保持“Offline”， 检查控制器的连接线， 控制器是否打开。

(13)检查运动控制器是否在停止状态。如果运动控制器在运行状态， 改为状态执行下一步。



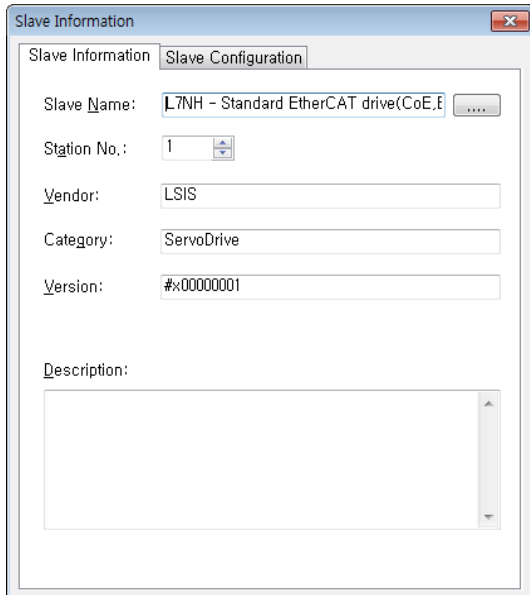
(14) 为了和伺服驱动器连接，驱动器的网络连接参数设置后，运动控制必须执行写入。首先，检查相关控制器是否处于离线状态设置网络参数。如果处于在线状态，执行"Online -Disconnect"，切换到离线状态。

(15) 鼠标右键单击工程树中从站参数，选择 "Add item –Slave-servo drive" 增加伺服驱动器网络参数。

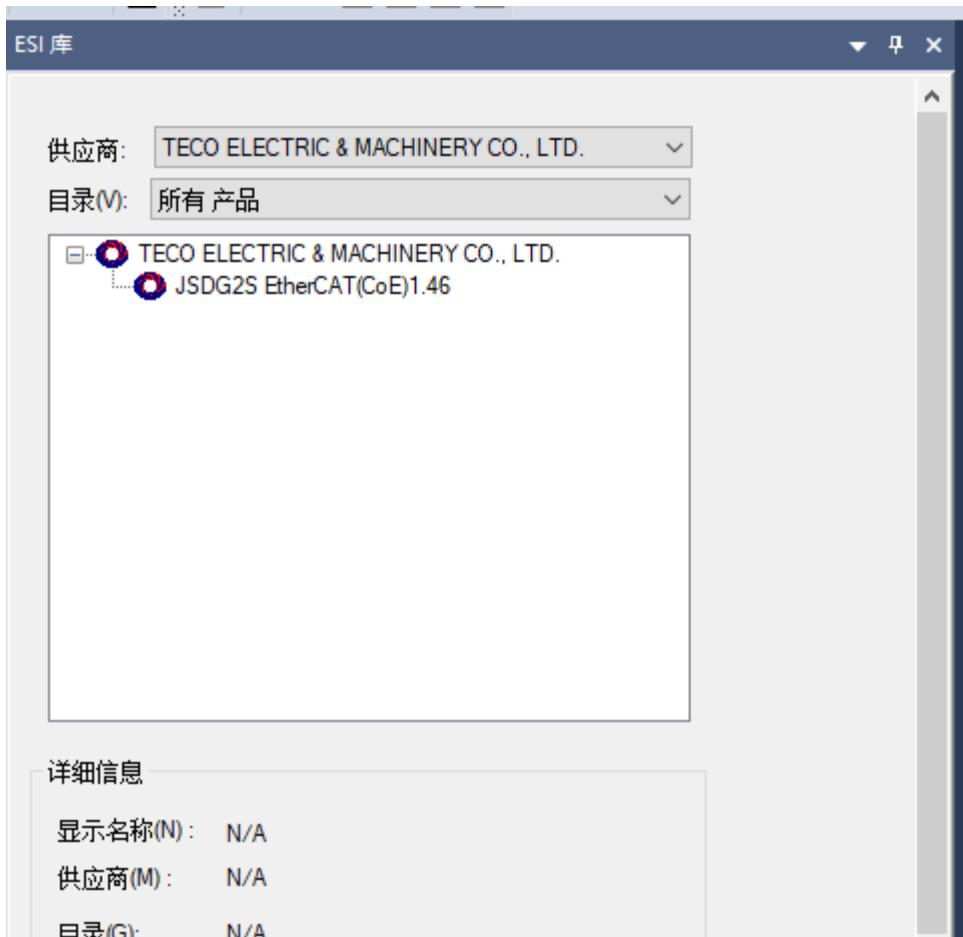


附录 3 设置示例

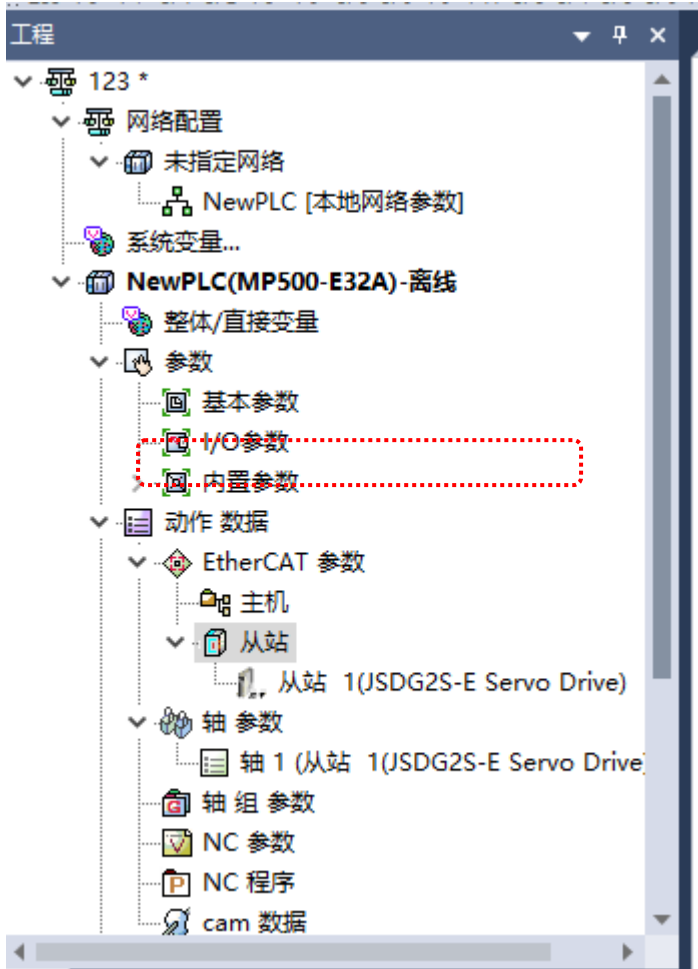
(16) 弹出从站信息窗口，点击“...”按钮从站名称。



(17) 在伺服驱动器选择窗口中选择第一个连接到运动控制器的伺服驱动器，点击OK。

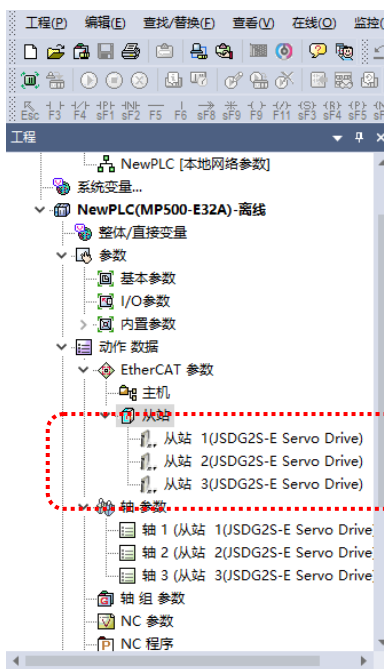


(18)轴号设置完成后，先前增加的伺服驱动器在EtherCAT参数的从站中显示。



(19)其他伺服驱动器用同样的方法增加。

增加所有伺服驱动器的连接从站参数，界面显示如下。



附录 3 设置示例

(20) 设置从站后，连接运动控制器控制的从站和轴。

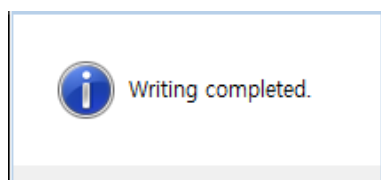
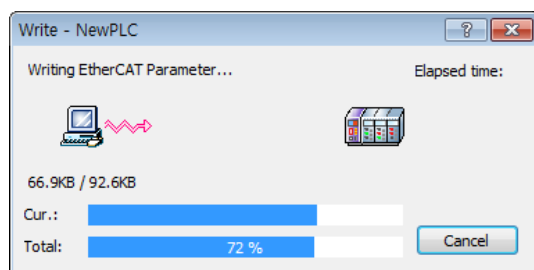
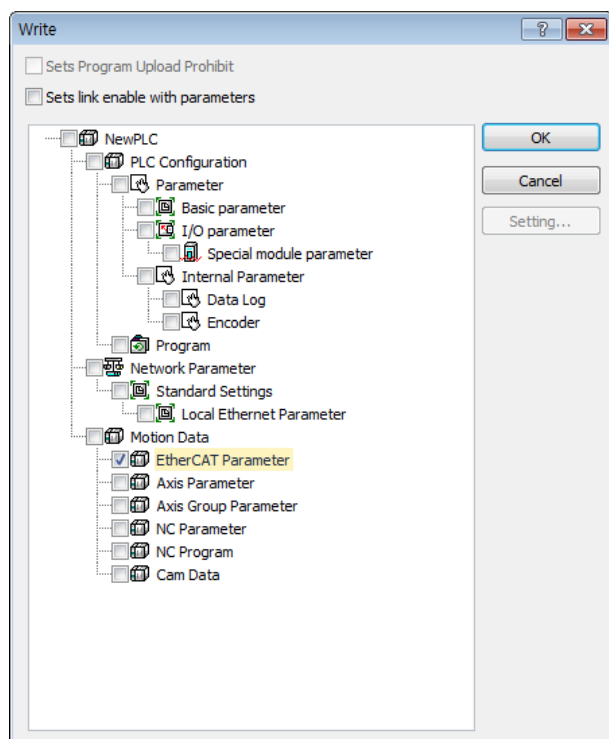
轴的设置顺序是从站的设置顺序，用户可以任意指定从站和轴。

工程树中选择轴参数，右击选择"轴Axis / 从站连接slave connection" 出现以下窗口，指定从站和轴。轴可以设置为设置的从轴和虚轴。



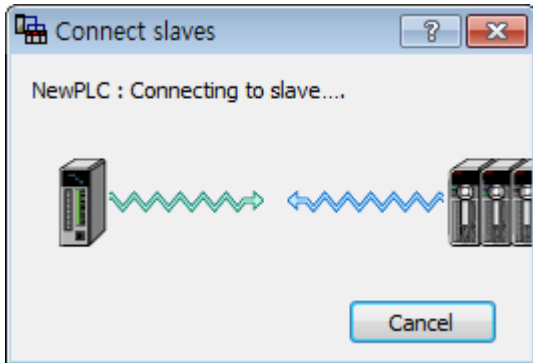
(21) 增加EtherCAT从站连接到EtherCAT的，先执行"在线Online-连接Connection" 再执行"Online-Write"，将EtherCAT参数写入到运动控制器中。

(22)出现工程写入窗口，检查EtherCAT 参数，检查OK执行写入确认。下面的界面是工程写入的执行过程。



附录 3 设置示例

(23) 选择“在线Online –从站Slave –连接Connect” 运动控制器和伺服驱动器开始通信连接.



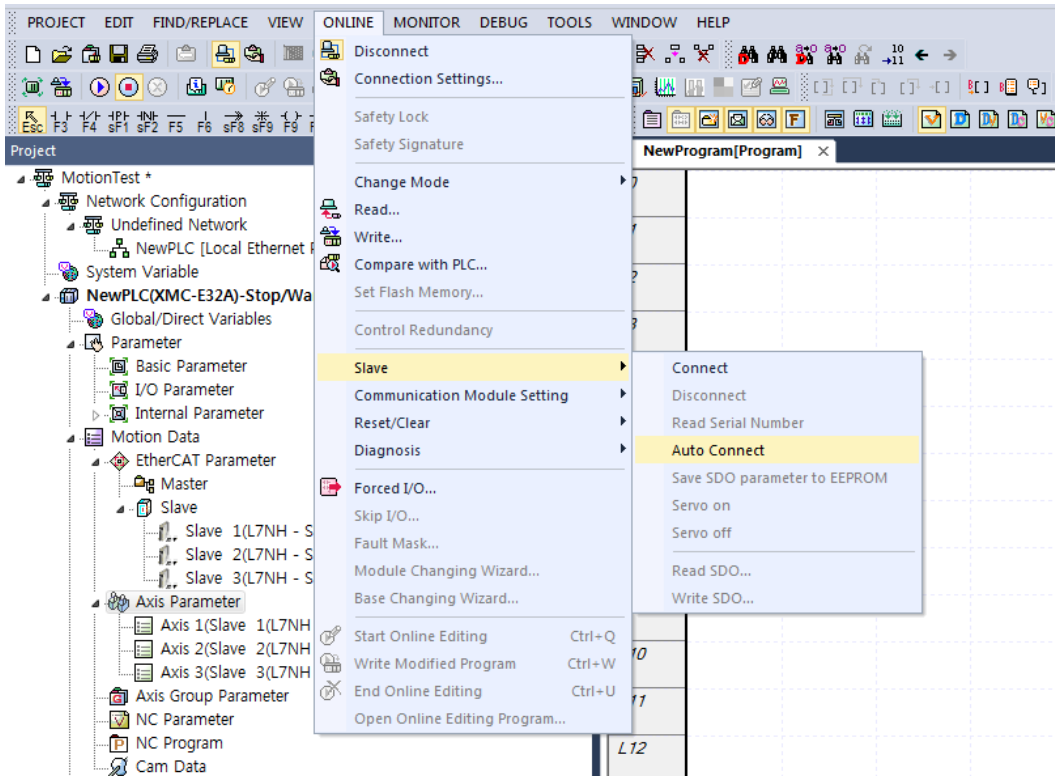
(24) 连接完成后, 从站参数的伺服驱动器名称会激活, 黑色变成灰色.

菜单中执行“查看View – 显示EtherCAT网络 (Display EtherCAT Network)” 检查伺服驱动器连接情况.

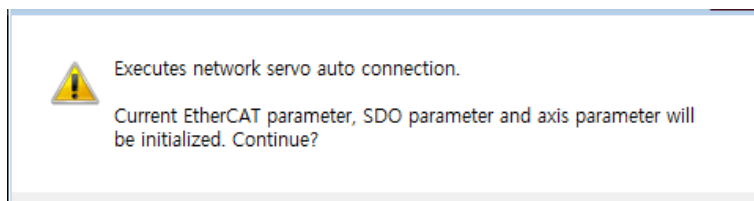
备注

使用运动控制器创建系统后，第一次连接网络，使用"从站自动连接（slave auto connect）"连接伺服驱动器比较方便，不用设置EtherCAT从站。

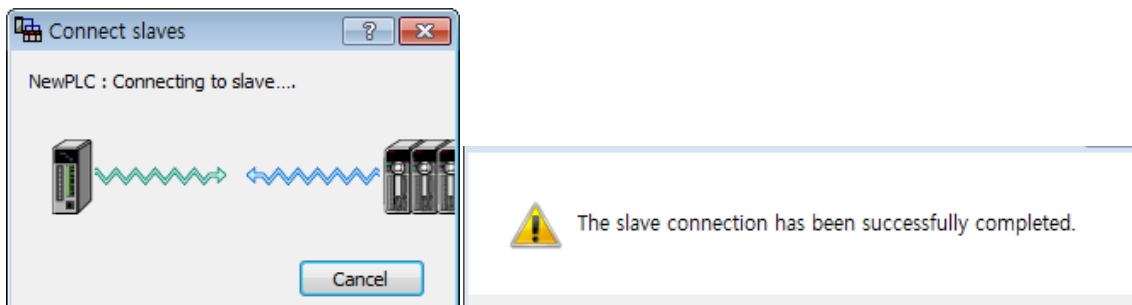
1. 执行“在线Online—从站slave—自动连接auto connect”。



2. 弹出通知信息，如下所示。这个是执行从站自动连接时的警告信息，当前MP500的网络参数，运动控制器，伺服参数（SDO参数）初始化，检查信息，点击OK。



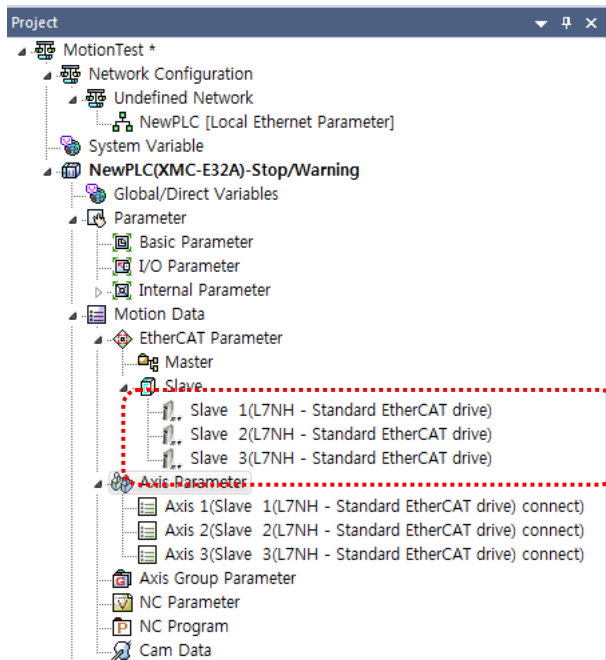
3. 从站连接信息出现，如果连接正常完成，会提示完成信息。



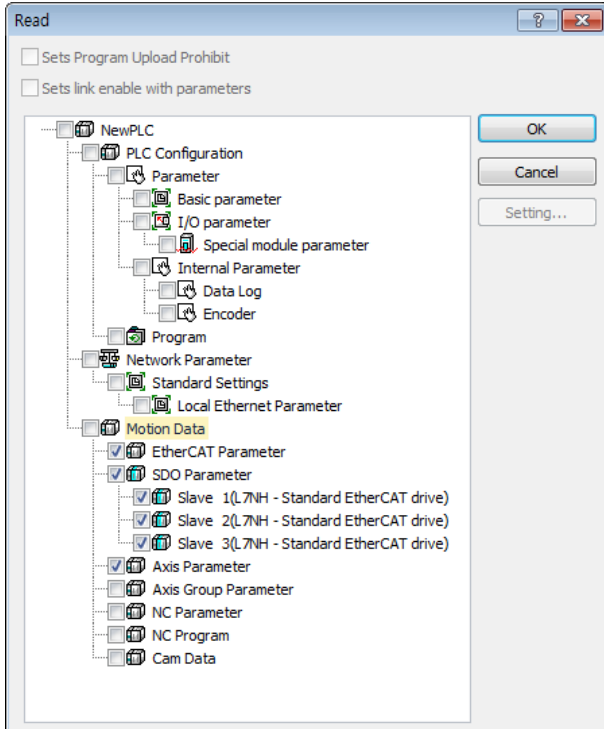
备注

附录 3 设置示例

4. 执行“从站自动连接 (slave auto connect)”命令, 如果连接正常完成, 连接到EtherCAT参数的当前EtherCAT从站信息会自动注册。



- (25) 为了设置运行参数和EtherCAT从站的SDO参数, 读取SDO参数。
目录中选择“在线Online –读取Read”, 选择读取的项目。



(26) 以下是读取的伺服驱动器的伺服参数。根据伺服驱动器的类型不同伺服参数的内容也有不同。详细的参考伺服驱动器的说明手册。

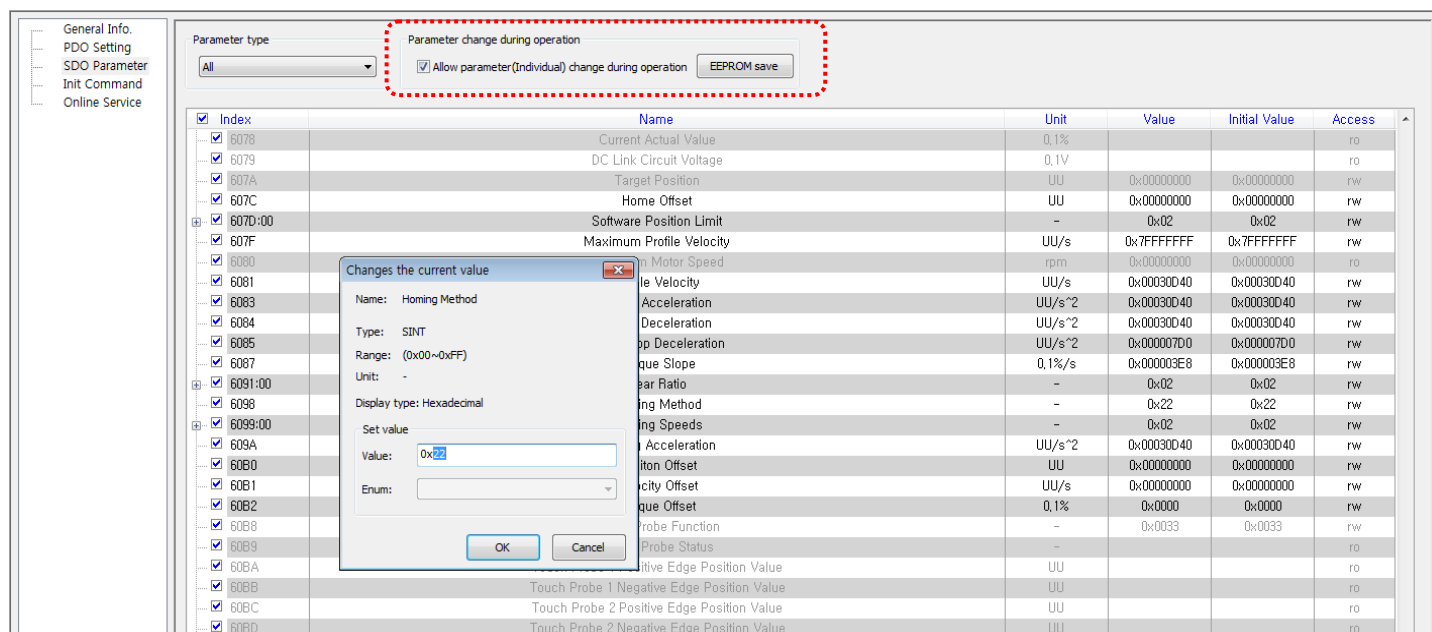
The screenshot shows a software interface with a project tree on the left and a parameter list on the right. The parameter list is titled '1.Slave' and contains the following data:

Index	Name	Unit	Value	Initial Value	Access
2000	Motor ID	-	13	13	rw
2001	Encoder Type	-	2	2	rw
2002	Encoder Pulse per Revolution	pulse	524288	524288	rw
2003	Node ID	-	-	-	ro
2004	Rotation Direction Select	-	0	0	rw
2005	Absolute Encoder Configuration	-	1	1	rw
2006	Main Power Fail Check Mode	-	0	0	rw
2007	Main Power Fail Check Time	ms	20	20	rw
2008	7SEG Display Selection	-	0	0	rw
2009	Regen, Brake Resistor Configuration	-	0	0	rw
200A	Regen, Brake Resistor Derating Factor	%	100	100	rw
200B	Regen, Brake Resistor Value	ohm	0	0	rw
200C	Regen, Brake Resistor Power	watt	0	0	rw
200D	Peak Power of Regen, Brake Resistor	watt	100	100	rw
200E	Duration Time @ Peak Power of Regen, Brake R...	ms	5000	5000	rw
200F	Overload Check Base	%	100	100	rw
2010	Overload Warning Level	%	50	50	rw
2011	PWM Off Delay Time	ms	10	10	rw
2012	Dynamic Brake Control Mode	-	0	0	rw
2013	Emergency Stop Configuration	-	1	1	rw
2014	Warning Mask Configuration	-	0	0	rw
2015	U Phase Current Offset	0.1%	0	0	rw
2016	V Phase Current Offset	0.1%	0	0	rw
2017	W Phase Current Offset	0.1%	0	0	rw
2018	Magnetic Pole Pitch	0,01mm	2400	2400	rw
2019	Linear Scale Resolution	nm	1000	1000	rw
201A	Commutation Method	-	0	0	rw
201B	Commutation Current	0.1%	500	500	rw
201C	Commutation Time	ms	1000	1000	rw
201D	Grating Period of Sinusoidal Encoder	um	40	40	rw
201E	Homing Done Behaviour	-	0	0	rw

附录 3 设置示例

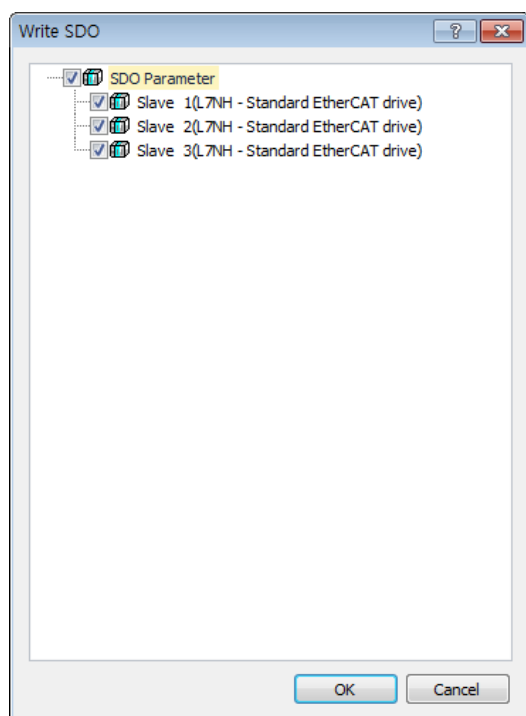
(27) SDO参数两种设置方法.

第1种只要改变SDO参数中一个项目的值, 选择'运行汇总允许SOD参数(单独)改变(Allow SDO Parameter(Individual) Change during Operation)'复选框, 设置想改变的SDO parameter, 设置值会立即适用到从站(伺服驱动器). SOD参数'当前参数(current value)'列会反映出修正值说明属性值正常传送了.



为了连接/关闭从站(伺服驱动器)电源后保存数据, 执行"Online-Save slave parameter to EEPROM"命令, 因为修正SOD参数(单独)只有电源连接时才有效.

第2种设置所有想修正的SDO参数执行'在线Online -写入Write', 将所有SDO参数一次写入从站(伺服驱动器).



写入所有SDO参数后，执行 "保存SDO参数到EEPROM(Save SDO parameter to EEPROM)" 命令. 不用单独执行"SDO parameter to EEPROM". 参考相关从站（伺服驱动器）的说明手册，根据SOD参数的类型，修正值在电源连接/关断后适用。

(28) SDO参数设置完成后，设置每个轴的操作参数，"在线Online-写入Write"中选择相关轴的操作参数写入到控制器中。

Group	Name	Axis 1	Axis 2	Axis 3
Basic Settings	Unit	0: pulse	0: pulse	0: pulse
	Pulse count per rotation	524288 pls	524288 pls	524288 pls
	Travel distance per rotation	10 pls	10 pls	10 pls
	Speed command unit	0: Unit/sec	0: Unit/sec	0: Unit/sec
	Speed limit	20000000 pls/s	20000000 pls/s	20000000 pls/s
	Emg. stop deceleration	0 pls/s ²	0 pls/s ²	0 pls/s ²
	Encoder selection	0: Incremental encoder	0: Incremental encoder	0: Incremental encoder
	Gear ratio of Motor side	1	1	1
	Gear ratio of Machine side	1	1	1
	Operating mode of the reverse rotation	0: Deceleration stop	0: Deceleration stop	0: Deceleration stop
Extended Settings	S/W upper limit	2147483647 pls	2147483647 pls	2147483647 pls
	S/W lower limit	-2147483648 pls	-2147483648 pls	-2147483648 pls
	Infinite running repeat. pos.	360 pls	360 pls	360 pls
	Infinite running repeat. pos.	0: Disable	0: Disable	0: Disable
	Command in-position range	0 pls	0 pls	0 pls
	Tracking error over-range value	0 pls	0 pls	0 pls
	Tracking error level	0: Warning	0: Warning	0: Warning
	Current pos. compensation amount	0 pls	0 pls	0 pls
	Current speed filter time constant	0 ms	0 ms	0 ms
	Error reset monitoring time	100 ms	100 ms	100 ms
	S/W limit during speed control	0: Do not detect	0: Do not detect	0: Do not detect
	Override mode	0: Specified by ratio	0: Specified by ratio	0: Specified by ratio
	JOG high speed	100000 pls/s	100000 pls/s	100000 pls/s
	JOG low speed	10000 pls/s	10000 pls/s	10000 pls/s
JOG acceleration	100000 pls/s ²	100000 pls/s ²	100000 pls/s ²	
JOG deceleration	100000 pls/s ²	100000 pls/s ²	100000 pls/s ²	
JOG jerk	0 pls/s ³	0 pls/s ³	0 pls/s ³	
NC Settings	Spindle command speed ack. range	95 %	95 %	95 %
	Spindle zero speed ack. rpm	5 rpm	5 rpm	5 rpm

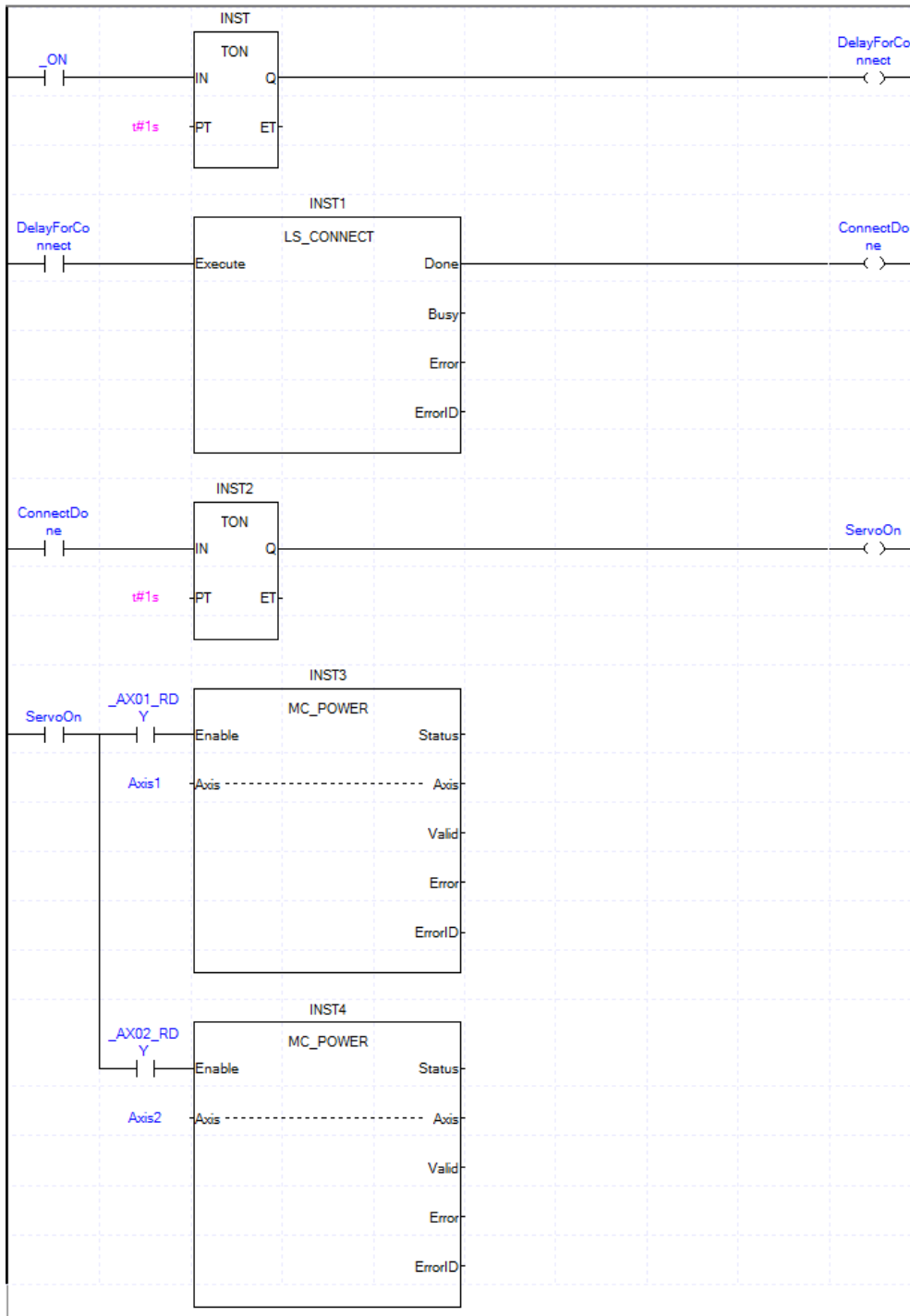
(29) 如果步骤（28）中关断从站（伺服驱动器）电源再打开，再执行"在线Online – 从站Slave – 连接Connect"连接模块和从站（伺服驱动器）。

附录 3 设置示例

- (30) 选择命令轴打开相关轴的伺服驱动器后，检查伺服中相关轴是否打开，通过点动或者其他方法检查马达运行。
- (31) 马达运行时产生振动或者噪音，调整马达参数的增益值、惯性比并传送到伺服驱动器。细节设置比如自整定使用伺服驱动器的专用设置工具。
- (32) 创建动作程序。

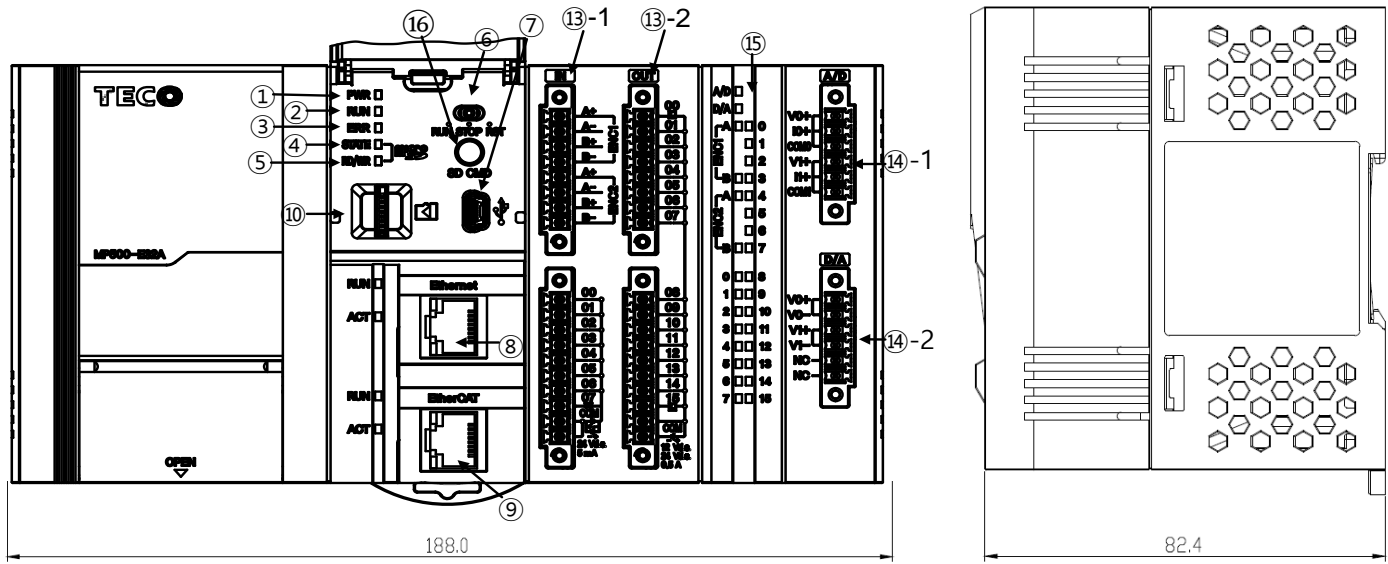
如下的例子，使用XMP500设置2个伺服器为轴1、轴2，连接使用LS_CONNECT，通过MC控制连接轴的伺服电源开通。其余的程序根据用户的需要增加。

运动任务可以分解为主要任务，阶段任务，初始任务。根据程序的特点，增加程序到项目树的相关任务中。



附录 4 尺寸

附录4尺寸



附录5 ESC(EtherCAT从站控制器)注册

下表为ESC(EtherCAT从站控制器)注册信息. 关于所有区域的信息, 参考如下BECKHOFF 网站的EtherCAT注册(第II部分)数据表.

http://www.beckhoff.com/english.asp?download/ethercat_development_products.htm

1. ESC DL状态 (0x0110:0x0111)

Bit	描述	ECAT	PDI	复位值
0	PDI运行/EEPROM正确加载: 0: EEPROM未加载, PDI不运行(不能访问处理数据RAM) 1: EEPROM正确加载, PDI运行(访问处理数据RAM)	r*/-	r/-	0
1	PDI看门狗状态: 0: 看门狗失效 1: 看门狗重新加载	r*/-	r/-	0
2	增强连接检测: 0: 所有端口激活 1: 激活至少一个端口 备注:仅在电源导通或者复位后第一个EEPROM加载时接收EEPROM值	r*/-	r/-	ET1100/ET1200: 1 直到第一个EEPROM 加载, 然后根据特性 的 EEPROM ADR 0x0000.9 IP Core: 1 直到第一 个EEPROM加载, 然 后 EEPROM ADR 0x0000.9 或者 0x0000[15:12] 其他: 0
3	保留	r*/-	r/-	0
4	物理连接端口0: 0: 未连接 1: 检测连接	r*/-	r/-	0
5	物理连接端口1: 0: 未连接 1: 检测连接	r*/-	r/-	0
6	物理连接端口2: 0: 未连接 1: 检测连接	r*/-	r/-	0
7	物理连接端口3: 0: 未连接 1: 检测连接	r*/-	r/-	0
8	环端口0: 0: 打开 1: 关闭	r*/-	r/-	0
9	通讯端口 0: 0: 无稳定通讯 1: 通讯稳定	r*/-	r/-	0
10	环端口1: 0: 打开 1: 关闭	r*/-	r/-	0

附录5ESC(EtherCAT从站控制器)注册

Bit	描述	ECAT	PDI	复位值
11	通讯端口1: 0: 无稳定通讯 1: 通讯稳定	r*/-	r/-	0
12	环端口2: 0: 打开 1: 关闭	r*/-	r/-	0
13	通讯端口2: 0: 无稳定通讯 1: 通讯稳定	r*/-	r/-	0
14	环端口3: 0: 打开 1: 关闭	r*/-	r/-	0
15	通讯端口3: 0: 无稳定通讯 1: 通讯稳定	r*/-	r/-	0

表 1-1: 注册ESC DL状态(0x0110:0x0111)

注册0x0111	端口3	端口2	端口1	端口0
0x55	未连接, 关闭	未连接, 关闭	未连接, 关闭	未连接, 关闭
0x56	未连接, 关闭	未连接, 关闭	未连接, 关闭	连接, 打开
0x59	未连接, 关闭	未连接, 关闭	连接, 打开	未连接, 关闭
0x5A	未连接, 关闭	未连接, 关闭	连接, 打开	连接, 打开
0x65	未连接, 关闭	连接, 打开	未连接, 关闭	未连接, 关闭
0x66	未连接, 关闭	连接, 打开	未连接, 关闭	连接, 打开
0x69	未连接, 关闭	连接, 打开	连接, 打开	未连接, 关闭
0x6A	未连接, 关闭	连接, 打开	连接, 打开	连接, 打开
0x95	连接, 打开	未连接, 关闭	未连接, 关闭	未连接, 关闭
0x96	连接, 打开	未连接, 关闭	未连接, 关闭	连接, 打开
0x99	连接, 打开	未连接, 关闭	连接, 打开	未连接, 关闭
0x9A	连接, 打开	未连接, 关闭	连接, 打开	连接, 打开
0xA5	连接, 打开	连接, 打开	未连接, 关闭	未连接, 关闭
0xA6	连接, 打开	连接, 打开	未连接, 关闭	连接, 打开
0xA9	连接, 打开	连接, 打开	连接, 打开	未连接, 关闭
0xAA	连接, 打开	连接, 打开	连接, 打开	连接, 打开
0xD5	Link, 关闭	未连接, 关闭	未连接, 关闭	未连接, 关闭
0xD6	Link, 关闭	未连接, 关闭	未连接, 关闭	连接, 打开
0xD9	Link, 关闭	未连接, 关闭	连接, 打开	未连接, 关闭
0xDA	Link, 关闭	未连接, 关闭	连接, 打开	连接, 打开

表1-2: ESC DL 状态寄存器0x0111 (仅典型模式)中的解码端口状态

2. RX 错误计数器(0x0300:0x0307)

如果对应端口使能时仅计数错误。

Bit	描述	ECAT	PDI	复位值
7:0	端口y (当0xFF到达时计数停止)无效帧计数器.	r/- w(clr)	r/-	0
15:8	端口y RX错误计数器 (当0xFF到达时计数停止). 这是直接耦合到MII端口/EBUS端口RX ERR.	r/- w(clr)	r/-	0

表2: 注册 RX错误计数器端口 y (0x0300+y*2:0x0301+y*2)

3. 正向RX错误计数器(0x0308:0x030B)

Bit	描述	ECAT	PDI	复位值
7:0	端口y正常错误计数器(当0xFF到达时计数停止).	r/- w(clr)	r/-	0

表3: 注册正向RX 错误计数器端口 y (0x0308+y)

备注: 如果写入RX错误计数器 0x0300-0x030B中的一个, 错误计数器0x0300-0x030B清除. 写入值忽略(写入 0).

4. ECAT 处理单位错误计数器(0x030C)

Bit	描述	ECAT	PDI	复位值
7:0	ECAT 处理单位错误计数器(当0xFF到达时计数停止). 计算通过处理单位的帧错误(例, FCS错误或者数据包错误).	r/- w(clr)	r/-	0

表4: 注册 ECAT 处理单位错误计数器(0x030C)

备注: 如果写入错误计数器0x030C, 错误计数器0x030C清除. 写入值忽略(写入0).

5.连接丢失计数器(0x0310:0x0313)

Bit	描述	ECAT	PDI	复位值
7:0	端口y连接丢失计数器 (当0xFF到达时计数停止). 仅到端口环自动时计数.	r/ w(clr)	r/-	0

表 5: 注册连接丢失计数器端口 y (0x0310+y)

备注: 仅在打开端口计数是连接丢失. 如果写入连接丢失计数器0x0310-0x0313中的一个时清除连接丢失计数器0x0310-0x0313. 写入值忽略 (写入 0).

6.AL 状态 (0x0130:0x0131)

Bit	描述	ECAT	PDI	复位值
3:0	设备状态机实际状态: 1: 初始化状态 3: 请求引导程序状态 2: 预运行状态 4: 安全运行状态 8: 运行状态	r*/-	r/(w)	1
4	错误 Ind: 0: 设备为请求或者通过指令的标志位清除状态 1: 设备没有进入请求状态, 或者变更为局部动作结果状态	r*/-	r/(w)	0
5	设备标识: 0: 设备标识无效 1: 设备标识加载	r*/-	r/(w)	0
15:6	保留, 写入0	r*/-	r/(w)	0

表6: 注册 AL状态(0x0130:0x0131)

备注: 如果设备仿真为off(0x0140.8=0), AL 状态注册仅可从PDI写入, 否则AL状态注册将反映AL控制注册值.

*从ECAT读取AL状态清除ECAT事件请求0x0210[3].

附录5ESC(EtherCAT从站控制器)注册

7. AL状态代码 (0x0134:0x0135)

Bit	描述	ECAT	PDI	复位值
15:0	AL状态代码	r/-	r/w	0

表7: 注册 AL状态代码 (0x0134:0x0135)

8. ECAT 事件请求 (0x0210:0x0211)

Bit	描述	ECAT	PDI	复位值
0	DC锁存事件: 0:在DC锁存输入中无变更 1: 在DC锁存输入至少一个变更 (用于ECAT控制锁存单位的位被从ECAT读取的DC锁存事件时间清除, 所以锁存0/1状态0x09AE:0x09AF表示无事件)	r/-	r/-	0
1	保留	r/-	r/-	0
2	DL状态事件: 0: DL状态中无变更 1: DL状态变更 (位通过从ECAT中读取的DL状态 0x0110:0x0111清除)	r/-	r/-	0
3	AL状态事件: 0: AL状态中无变更 1: AL状态变更 (位通过从ECAT中读取的AL状态 0x0130:0x0131清除)	r/-	r/-	0
4	每个SyncManager状态的镜像值:	r/-	r/-	0
5	0: 无同步通道0事件 1: 同步通道0事件挂起 0: 无同步通道1事件 1: 同步通道1事件挂起			
11	0: 无同步通道7事件 1: 同步通道7事件挂起	r/-	r/-	0
15:12	保留	r/-	r/-	0

表8: 注册 ECAT 事件请求 (0x0210:0x0211)

保修内容

1. 保修时间

购入产品的保修时间是自制造日起 18 个月。

2. 保修范围

对于保修期间内发生的故障，一部分可以换货或者修理。但是，下面这些情况下不在保修范围内，请谅解。

- (1) 未遵照使用说明书，由于不当的条件，环境，搬取等原因造成的故障。
- (2) 故障是由于本公司产品以外的原因造成的。
- (3) 在本公司及本公司指定的代理店以外的场所改造及修理的情况下。
- (4) 不是产品本来的使用方法的情况下。
- (5) 在本公司出库时的科学，技术水准下不可预料的原因情况下。
- (6) 其它天灾，火灾等无须承担责任的情况下。

3. 上面的保修只是指 PLC 单位体，所以系统构成或者产品应用时，请考虑安全性后再使用。

环境方针

TECO 产电遵守如下环境方针。

环境经营

TECO产电以环境保护作为经营的优先课题，全体员工为了愉快的地球环境保护而尽最大努力。

产品废弃向导

TECO产电 PLC是被设计为能够保护环境的产品。
产品废弃时，可以分解为铝，铁，合成树脂（外壳）后再利用。

保修内容

1. 保修时间

购入产品的保修时间是自制造日起 18 个月。

2. 保修范围

对于保修期间内发生的故障，一部分可以换货或者修理。但是，下面这些情况下不在保修范围内，请谅解。

- (1) 未遵照使用说明书，由于不当使用的条件，环境，搬取等原因造成的故障。
- (2) 故障是由于本公司产品以外的原因造成的。
- (3) 在本公司及本公司指定的代理店以外的场所改造及修理的情况下。
- (4) 不是产品本来的使用方法的情况下。
- (5) 在本公司出库时的科学，技术水准下不可预料的原因情况下。
- (6) 其它天灾，火灾等无须承担责任的情况下。

3. 上面的保修只是指 PLC 本身，所以系统构成或者产品应用时，请考虑安全性后再使用。



台安科技(無錫)有限公司

地址:江苏省无锡国家高新技术产业开发区 65-C 号

电话: 0510-85227555

传真: 0510-85227556

<http://www.taian-technology.com>

经销联络处:

VER:01 2019.3

为持续改善产品，本公司保留变更设计规格的权利。