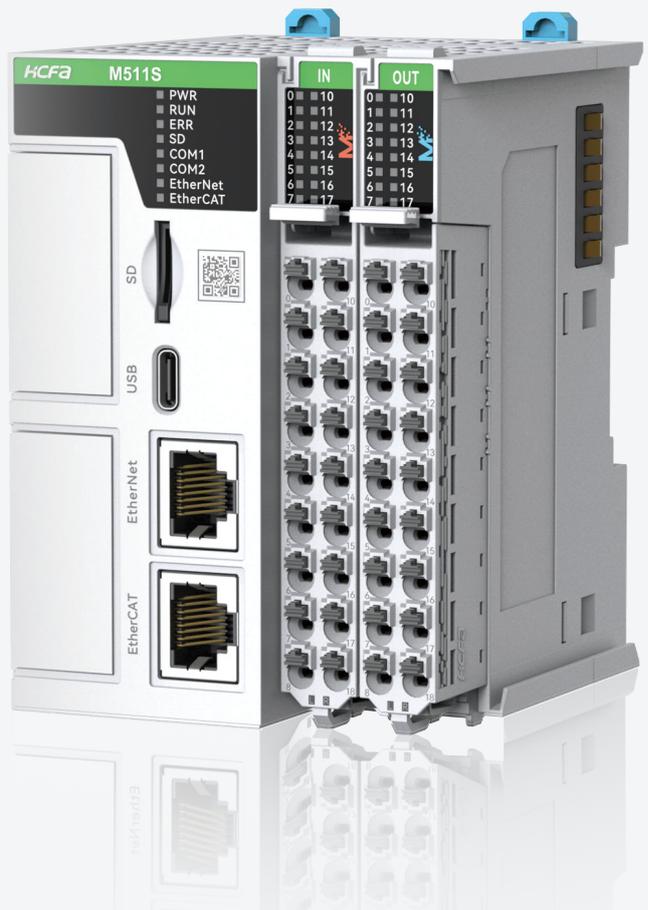


# M系列 可编程逻辑控制器

运动控制

## 指令手册



# 目录

目录 .....	I
前言 .....	IV
第一章 运动控制指令基本知识 .....	1
1.1 运动控制指令构成 .....	2
1.2 运动控制指令支持的语言 .....	2
1.3 运动控制指令的执行 .....	3
1.4 运动控制指令参数单位 .....	3
1.5 运动控制指令相关参数说明 .....	4
第二章 变量 .....	6
2.1 轴变量 .....	7
第三章 运动控制相关 .....	9
3.1 轴状态机 .....	10
3.2 运动控制指令的多重启动时缓存模式说明 (BufferMode) .....	11
3.3 不同机种支持的轴规格及运动控制功能 .....	16
第四章 单轴指令 .....	17
4.1 MC_Power (使能运行/解除运行) .....	18
4.2 MC_Reset (轴异常复位) .....	21
4.3 MC_Jog (点动) .....	27
4.4 MC_HomeByPLCIO (通过控制输入信号的原点设置) .....	32
4.5 MC_Home (原点设置) .....	38
4.6 MC_HomeWithParm (参数指定原点设置) .....	44
4.7 MC_MoveVelocity (速度) .....	49
4.8 MC_MoveContinuousVelocity (速度实时更改) .....	56
4.9 MC_Halt (停止) .....	59
4.10 MC_Stop (停止并锁定) .....	65
4.11 MC_StopAtPhase (指定相位停止) .....	71
4.12 MC_MoveRelative (相对位移) .....	75
4.13 MC_MoveAbsolute (绝对位移) .....	82
4.14 MC_MoveAdditive (附加位移) .....	90
4.15 MC_MoveSuperimposed (叠加相对位移) .....	97
4.16 MC_HaltSuperimposed (停止叠加位移) .....	104
4.17 MC_SetOverride (速度比例设置) .....	110
4.18 MC_TorqueControl (扭矩设定) .....	112
4.19 MC_TorqueControlWithVelocity (带速度限制的扭矩控制) .....	115
4.20 MC_SyncMoveAbsolute (周期增量位置控制) .....	122
4.21 MC_SetPosition (位置设置) .....	124
4.22 MC_ReadActualPosition (反馈位置读取) .....	131
4.23 MC_TouchProbe (锁定轴位置) .....	133
4.24 MC_TouchprobeCyc (信号触发锁定轴位置周期执行) .....	140
4.25 MC_MoveFeed (外部中断后移动指定的距离) .....	144
4.26 MC_ReadStatus (读取轴状态) .....	155

4.27 MC_ReadAxisError (轴异常读取)	161
4.28 MC_ReadMotionState (轴运动状态读取)	163
4.29 MC_GetFollowingStatus (读取位置误差状态)	165
4.30 MC_SetFollowingParm (位置误差参数设定)	167
4.31 MC_EnableSoftLimit (软件限位激活设置)	169
4.32 MC_SetAxisParm (更改机构参数)	171
第五章 多轴指令	173
5.1 MC_GearIn (电子齿轮)	174
5.2 MC_CombineAxes (双主轴电子齿轮)	183
5.3 MC_GearOut (电子齿轮关系解除)	192
5.4 MC_CamIn (电子凸轮耦合)	198
5.5 MC_CamOut (电子凸轮脱离)	209
5.6 MC_SetCamPoint (更改指定凸轮点数据)	210
5.7 MC_ChangeCamCurve (凸轮数据变更)	220
5.8 MC_GetCamPoint (读取指定凸轮点数据)	222
5.9 MC_GetCamTappetStatus (读取凸轮挺杆点状态)	226
5.10 MC_SetCamTappet (设置凸轮挺杆点数据)	228
5.11 MC_GetCamTappet (读取凸轮挺杆点设置数据)	231
5.12 MC_AddCamTappet (新增凸轮挺杆点)	234
5.13 MC_DeleteCamTappet (删除凸轮挺杆点)	235
第六章 旋切工艺及相关指令	238
6.1 旋切工艺简介及适用场合	239
6.2 MC_SetRotaryKnifeParameter (旋切机构参数设置)	240
6.3 MC_RotaryKnife_In (旋切耦合)	242
6.4 MC_RotaryKnife_Out (旋切关系解除)	244
第七章 轴组指令	251
7.1 MC_AxesGroupAddAxis (指定轴在轴组中的编号)	252
7.2 MC_AxesGroupDeleteAxis (删除逻辑轴在轴组中编号)	254
7.3 MC_AxesGroupDeleteAllAxis (删除所有逻辑轴在轴组中编号)	256
7.4 MC_AxesGroupEnable (轴组运行和解除运行)	258
7.5 MC_AxesGroupPause (轴组运动暂停)	260
7.6 MC_AxesGroupResume (轴组运动暂停解除)	262
7.7 MC_AxesGroupExit (轴组运动急停)	264
7.8 MC_AxesGroupSetOverride (轴组速度比例设置)	266
7.9 MC_MoveLinearRelative (相对值直线插补)	269
7.10 MC_MoveLinearAbsolute (绝对值直线插补)	273
7.11 MC_MoveDirectRelative (相对值单独定位)	277
7.12 MC_MoveDirectAbsolute (绝对值单独定位)	280
7.13 MC_MoveCircularRelative (相对值圆弧插补)	284
7.14 MC_MoveCircularAbsolute (绝对值圆弧插补)	288
7.15 MC_AxesGroupReadActualPosition (轴组中各个轴反馈位置读取)	292
7.16 轴组运动指令示例程序	294
第八章 CNC 介绍	299

8.1 CNC 简介.....	300
8.2 G 代码功能详细介绍.....	302
8.2.1 G90 (绝对值模式) .....	302
8.2.2 G91 (相对模式) .....	302
8.2.3 G0 (快速定位) .....	302
8.2.4 G1 (直线插补) .....	304
8.2.5 G2 (顺时针圆弧/螺旋插补) .....	306
8.2.6 G3 (逆时针圆弧/螺旋插补) .....	311
8.2.7 G17/G18/G19 (指定圆弧插补平面) .....	316
8.2.8 G4 (延时指令) .....	317
8.2.9 G50 (无过渡曲线模式) .....	318
8.2.10 G51 (指定附加角过渡模式) .....	319
8.2.11 G52 (以恒定的速度过渡模式) .....	320
8.2.12 M 代码 .....	321
8.3 MC_SetMoveLinearParm (CNC 插补运动默认参数设定) .....	323
8.4 MC_SetMoveDirectParm (CNC 单独定位运动参数设定) .....	325
8.5 MC_SetStartPosition (轴组初始位置设置) .....	327
8.6 MC_CoorMotion (CNC 执行) .....	329
8.7 MC_GetMCodeStatus (M 代码状态读取) .....	332
8.8 MC_ResetMCode (M 代码复位) .....	334
8.9 CNC 代码调用执行示例程序.....	336
附录 .....	341
附录 1 原点返回模式.....	342
附录 1.1 原点返回模式中的控制字设定 (60400010h) .....	343
附录 1.2 原点返回模式的状态字定义 (60410010h) .....	343
附录 1.3 原点返回模式相关的参数.....	344
附录 1.4 原点返回模式简单使用教程 (以 X3E 伺服举例) .....	344
附录 1.5 原点回归模式介绍 .....	345
附录 2 指令错误代码描述.....	365

## 前言

非常感谢您购买M系列运动控制器。该手册主要介绍运动控制器运动指令，如运行/解除运行能指令、速度指令、相对位移指令、电子齿轮、电子凸轮、轴组等。

### 阅读对象

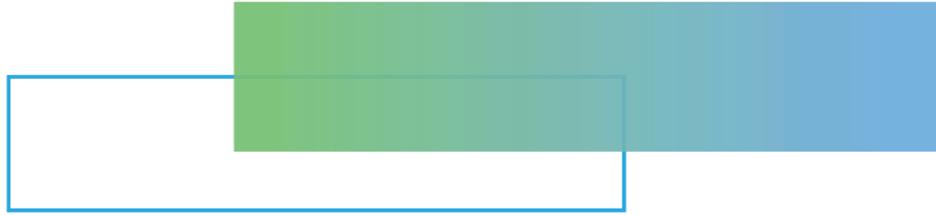
本手册面向M200系列、M311、M312、M500S系列、M500系列运动控制器编程和调试的技术人员。读者需要具备一定的可编程控制器相关的基础知识和编程思维。

### 手册修订说明

版本	变更时间	修订内容
V1.00	2023/09/07	初版

### 其他说明

- 本手册内容基于产品信息和客户需求编辑，用户对手册内容有疑问或错误之处，欢迎致电禾川或发送邮件至 [400@hcfa.cn](mailto:400@hcfa.cn)，并按照封面标注版本号协助说明。
- 本手册内容，包括文字、图片、标识、表格等，未经授权时，不得以任何形式复制和传递本手册中的内容，否则，我司将依法追究违规者的法律责任



# 第一章 运动控制指令基本知识



## 1.1 运动控制指令构成

梯形图中的运动指令如下图所示，运动控制指令由实例化名称、指令名称、输入变量、输出变量组成。实例化名称用于给对应的指令分配内存，使用时，要为每个指令分配不同的实例化名称。

通过输入参数给输入变量设定运动控制指令需要的值，如目标位置、目标速度等。输入参数省略时，输入变量的值为初始值。保留的输入变量或者使用中对指令执行没有影响的输入变量可以省略输入参数。

输出变量用于输出指令的状态，有输出变量时，将输出变量的值传给输出参数

输入变量和输出变量可以通过实例化名称使用，如下图所示，可以通过输入参数Pos给输入变量Position赋值，也可以将输出参数Done的值赋值给输出参数Abs\_Done。

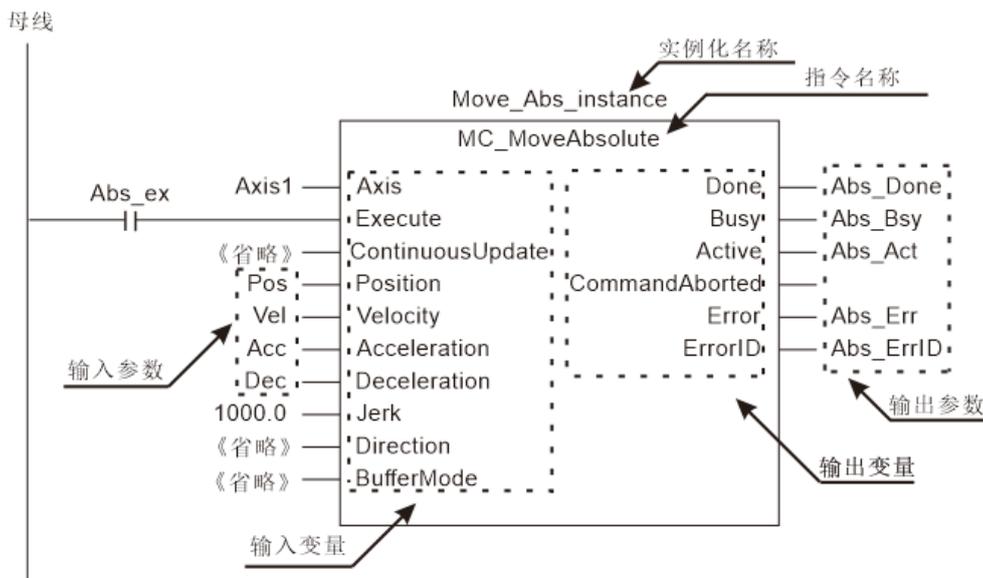


图1-1 运动控制指令构成图

## 1.2 运动控制指令支持的语言

运动控制指令在软件中编辑时支持梯形图（LD）和结构化文本（ST）两种编程语言。

- 梯形图（LD）
- 结构化文本（ST）

下图所示为MC\_MoveAbsolute(绝对值位移)指令在梯形图中编辑的范例：

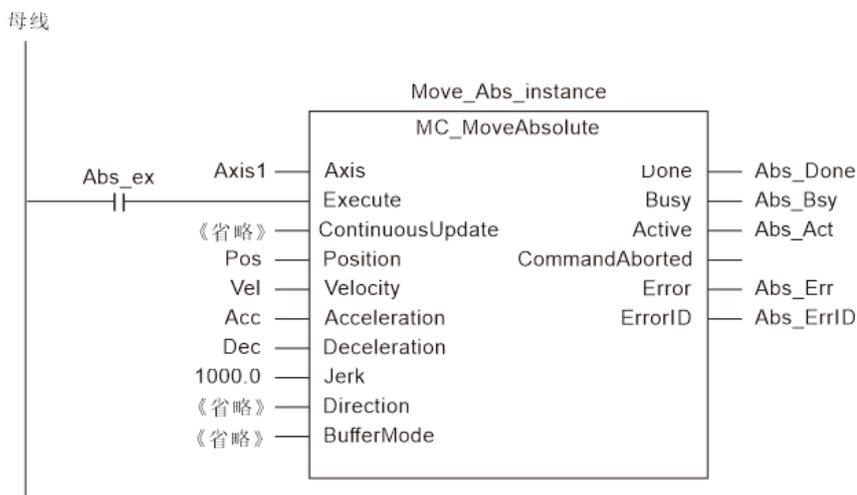


图1-2 MC\_MoveAbsolute指令在梯形图中的示例图

以下为MC\_MoveAbsolute(绝对值位移)指令在结构化文本(ST)语言中编辑的范例:

输入变量和输入参数之间的符号为:=, 输出变量和输出参数之间的符号为=>。

```
Move_Abs_instance(
  Axis:=axis1 ,
  Execute:= abs_ex,
  Position:=pos ,
  Velocity:=vel ,
  Acceleration:= acc,
  Deceleration:= dec,
  Jerk:=1000.0 ,
  Done=>abs_done ,
  Busy=> abs_bsy,
  Active=> abs_act,
  Error=> abs_err,
  ErrorID=>abs_errID);
```

### 1.3 运动控制指令的执行

运动控制指令只能添加在事件触发(运动事件)任务中才可以执行, 添加在其它任务中不能正常执行:

任务类型		是否可以执行
事件触发	运动事件	是
	非运动事件	否
固定周期		否
自由滑行		否

### 1.4 运动控制指令参数单位

运动控制指令中的Position(绝对位置)和Distance(移动距离)参数的单位和软件中设置的机构“工作每转的工作行程”的单位相同。

软件中设置的地方在“运动控制”→“轴设置”→“基本设置”。

如下图所示的机构, 电机转1圈, 对应丝杠的工作行程为10毫米, 软件中按照下图所示的设置, 下载后, 通过MC\_MoveRelative(相对位移)指令控制丝杠运转10毫米时, 则该指令输入变量Position(绝对位置)值的单位为毫米, Velocity(目标速度)的单位为毫米/秒, Acceleration(加速度)和Deceleration(减速度)的单位为毫米/秒<sup>2</sup>, Jerk(跃度)的单位为毫米/秒<sup>3</sup>。

运动控制指令的参数单位, 如Distance(移动距离), Velocity(目标速度)、Acceleration(加速度)、Deceleration(减速度)、Jerk(跃度)等和实际的机构及软件设置有关, 我们将位置相关的单位称之为行程单位。



图1-3 轴设置参数图

下表所示为MC\_MoveRelative指令在轴经过“基本设置”后的相关参数单位。

输入变量	名称	单位
Distance	移动距离	毫米
Velocity	目标速度	毫米/秒
Acceleration	加速度	毫米/秒 <sup>2</sup>
Deceleration	减速度	毫米/秒 <sup>2</sup>
Jerk	跃度	毫米/秒 <sup>3</sup>

## 1.5 运动控制指令相关参数说明

控制器速度控制采用S型速度曲线，可有效减小运动控制中的冲击。S型速度曲线运行时需要设定Velocity（速度）、Acceleration（加速度）、Deceleration（减速度）、Jerk（跃度）参数值，这些参数可以在运动控制指令中设置。

轴运动过程中对应的位置、速度、加速度、跃度的简称如下表所示：

位置	速度	加速度/减速度	跃度
p	v	a	j

轴运动过程中对应的位置、速度、加速度、跃度的关系如下：

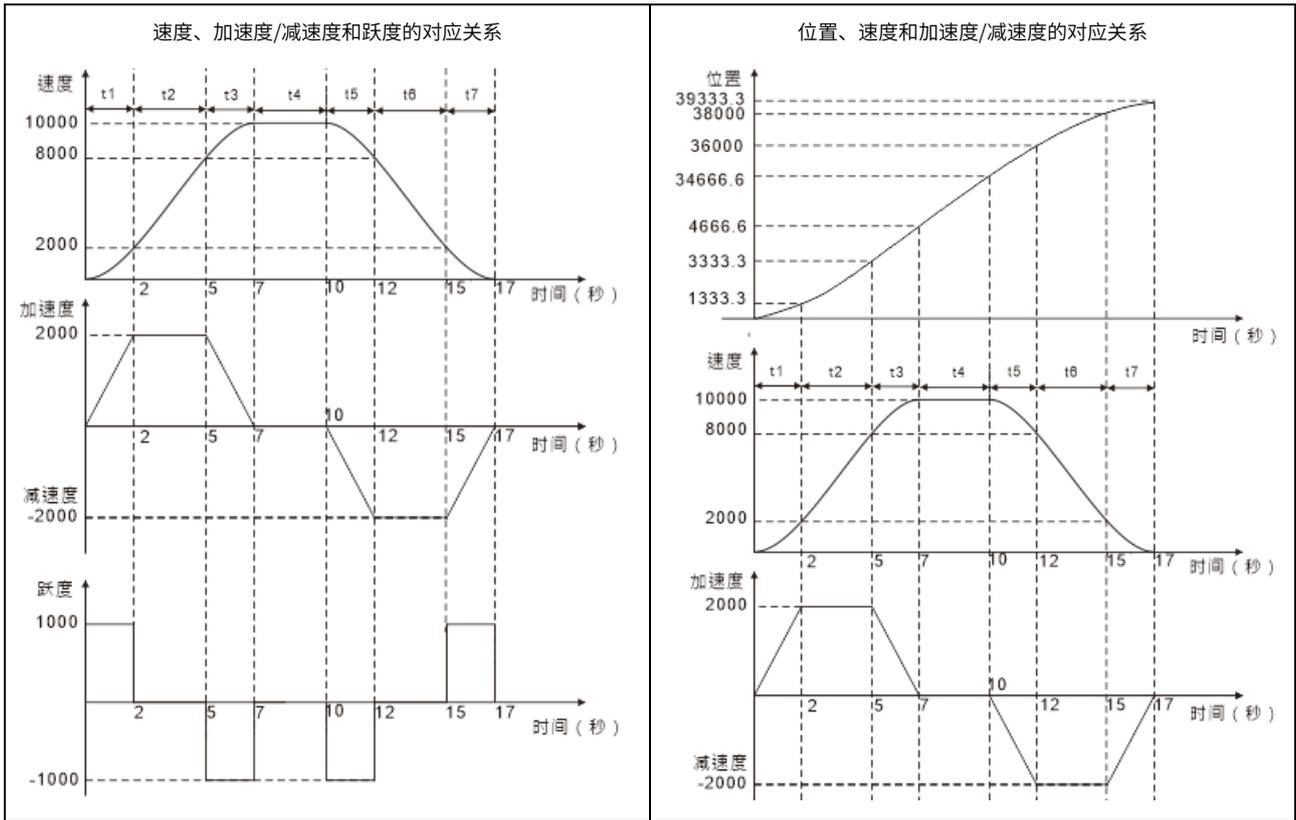
加速度未到达设定加速度时， $a=jt$ ；加速度到达设定加速度时， $a$ 为设定加速度的值；

加速度大于0，加速度减小时， $a=设定加速度+jt \times -1$ 。

速度对时间的导数为加速度： $a=dv/dt$ 。

位置对时间的导数为速度： $v=dp/dt$ 。

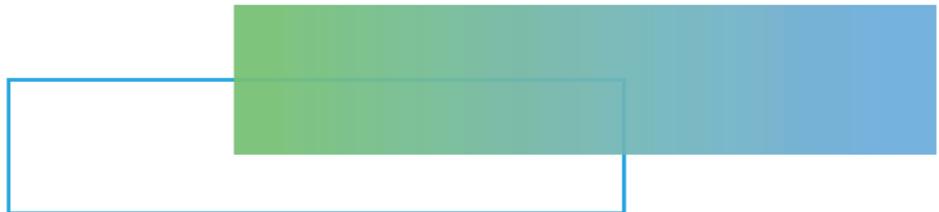
下图所示为轴当前位置为0，MC\_MoveRelative指令Velocity（速度）=10000、Acceleration（加速度）=2000、Deceleration（减速度）=2000、Jerk（跃度）=1000时，位置、速度、加速度、跃度和时间的对应关系图。



阶段	名称	时间 (秒)	跃度 (j)	加速度/减速度 (a)	速度值 (v)	位置值 (p)
t1	加加速段	2	1000	从0变化到2000	从0变化到2000	从0变化到1333.3
t2	匀加速段	3	0	固定为2000	从2000变化到8000	从1333.3变化到3333.3
t3	减加速度段	2	-1000	从2000变化到0	从8000变化到10000	从3333.3变化到4666.6
t4	匀速段	3	0	0	固定为10000	从4666.6变化到34666.6
t5	加减速段	2	-1000	从0变化到-2000	从10000变化到8000	从34666.6变化到36000
t6	匀减速段	3	0	固定为-2000	从8000变化到2000	从36000变化到38000
t7	减减速段	2	1000	从-2000变化到0	从2000变化到0	从38000变化到39333.3

上图和上表所示，t1~t4阶段跃度、加速度、速度、位置的计算方法如下表所示。

阶段	时间 (秒)	跃度 (j)	加速度/减速度 (a)	速度值 (v)	位置值 (p)
t1	2	1000	$j \times t1$	$1/2 \times j \times t1^2 = 2000$	$1/6 \times j \times t1^3$
t2	3	0	固定为2000	$2000 + a \times t2 = 8000$	$1/6 \times j \times t1^3 + 1/2 \times a \times t2^2$
t3	2	-1000	$2000 + j \times t3 \times -1$	$8000 + 1/2 \times j \times t3^2$	$1/6 \times j \times t1^3 + 1/2 \times a \times t2^2 + 1/6 \times j \times t3^3$
t4	3	0	0	固定为10000	$1/6 \times j \times t1^3 + 1/2 \times a \times t2^2 + 1/6 \times j \times t3^3 + 10000 \times t4$



## 第二章 变量



## 2.1 轴变量

轴变量为系统定义的数组型参考型全局变量，名称为Axis，对应不同的轴，可以通过其成员访问，对应成员为Axis[对应轴号]。在工程中使用时，直接使用轴变量成员的变量，无需声明，如获取轴1的反馈位置时，可以通过Axis[1].ActPos获取。轴变量成员变量的详细说明如下表所示。

轴变量成员	数据类型	含义	单位	
Axis[对应轴号].	CmdPos	LREAL	轴命令位置	行程单位
	CmdVel	LREAL	轴命令速度	行程单位/秒
	CmdAcc	LREAL	轴命令加速度	行程单位/秒 <sup>2</sup>
	CmdTrq	INT	轴命令扭矩	[0.1%]
	ActPos	LREAL	轴反馈位置	行程单位
	ActVel	LREAL	轴反馈速度	行程单位/秒
	ActTrq	INT	轴实际扭矩	[0.1%]
	AxState	USINT	轴状态 (参考下方“轴变量中轴状态对应值的含义”)	
	ErrPos	LREAL	同周期内轴命令位置和反馈位置的差值	行程单位
	AxErrCode	UINT	轴错误码	
	AxType	USINT	轴位置模式 1: 线性模式 0: 循环模式	
	AxModulo	LREAL	模 (轴位置模式为循环模式时有效)	
	ScaleDen	LREAL	工作齿轮减速比分母	
	ScaleNum	LREAL	工作齿轮减速比分子	
	UnitPerTurn	LREAL	工作每转的工作行程	
	VelFactor	LREAL	目标速度的倍数	
	EnableSoftLimit	BOOL	软件限位是否激活 1: 激活 0: 未激活	
	PositivePosLimit	LREAL	正向软件限位	行程单位
NegativePosLimit	LREAL	反向软件限位	行程单位	

轴变量中轴状态对应值的含义

轴变量成员	对应值	对应值的含义
AxState	0	Disabled (解除运行)
	1	Standstill (停止)
	2	ErrorStop (错误停止)
	3	Stopping (减速停止)
	4	Homing (原点寻找)
	5	DiscreteMotion (定位动作中)
	10	旋切状态
	11	电子齿轮啮合中
	12	电子凸轮啮合中
	14	扭矩状态
	15	ContinuousMotion (连续动作中)
	16	电子凸轮同步
	17	电子齿轮同步

---

	18	双主轴电子齿轮同步
--	----	-----------

---



## 第三章 运动控制相关





## 3.2 运动控制指令的多重启动时缓存模式说明 (BufferMode)

当前运动控制指令（当前指令）控制轴时，对同一个轴启动另外一个运动控制指令（缓存指令），称之为多重启动。缓存指令和当前指令之间的速度过渡方式由缓存指令的BufferMode的值决定，称之为缓存模式。通过缓存模式，可以实现两个指令执行时，轴的速度不降低为0，提高工作效率，如多段速时可以通过缓存模式实现。多重启动时相关用语含义入下表所示。

用语	含义
当前指令	当前正在控制轴的运动控制指令。
缓存指令	轴被其它指令控制时，对同一个轴启动并且处于等待状态的运动控制指令。
缓存模式	当前指令目标位置到达时，缓存指令控制轴的模式。
中继速度	当前指令目标位置到达时使用的速度。

多重启动时，缓存模式选择中断时，缓存指令立即执行，无缓存；缓存模式选择其它模式时，缓存指令会缓存，但缓存只能有1级缓存。如当前指令执行时，缓存指令启动后，如果再启动另外一个指令进行缓存，最后执行的指令会报错。

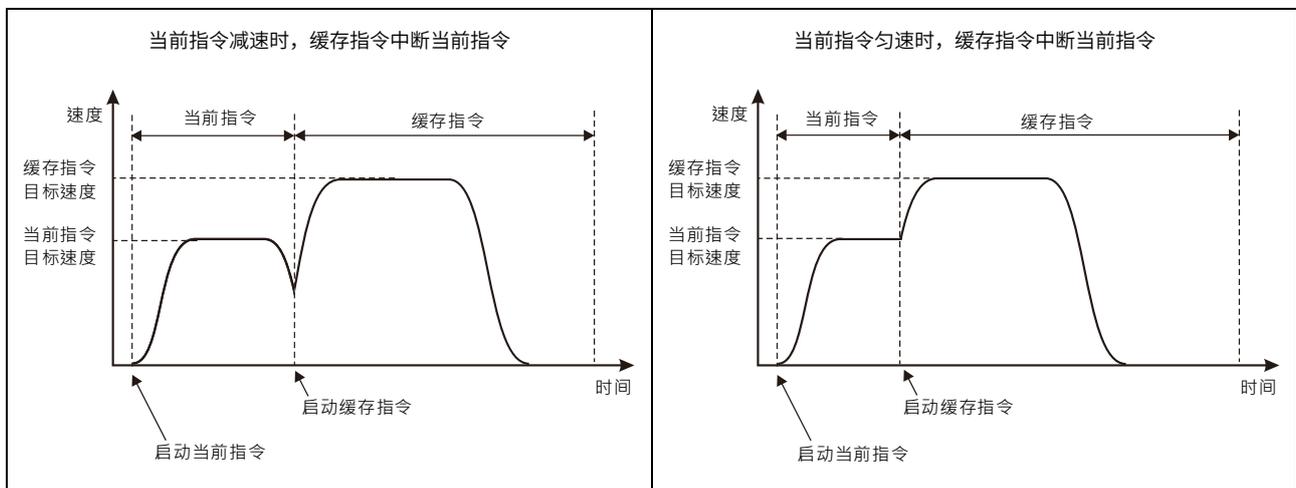
多重启动时，mcAborting（中断）模式除外，缓存指令的启动时间要尽量靠前，如当前指令将要完成时再启动缓存指令，则中继速度无法到达。建议使用当前指令的输出变量Active（控制中）触发缓存指令启动。

缓存模式不同模式的含义入下表所示。

缓存模式	含义
mcAborting (中断)	中断当前正在控制轴的指令，切换为缓存指令控制。
mcBuffered (等待)	等待当前指令执行完成时（如目标位置或者目标速度到达），切换为缓存指令控制轴。
中继	当前指令目标位置到达时，切换为缓存指令控制轴，并且切换为缓存指令时的目标速度为通过缓存模式选择的速度。这些模式下，当前指令和缓存切换时，轴速度不会停止。
mcBlendingLow (以低速中继)	比较当前指令的目标速度和已缓存该指令的目标速度高低，当前指令目标位置到达时，以速度较低者做为中继速度。
mcBlendingPrevious (以当前指令的目标速度中继)	当前指令目标位置到达时，以当前指令的目标速度做为中继速度。
mcBlendingNext (以缓存指令的目标速度中继)	当前指令目标位置到达时，以缓存指令的目标速度做为中继速度。
mcBlendingHigh (以高速中继)	比较当前指令的目标速度和已缓存该指令的目标速度高低，当前指令目标位置到达时，以速度较高者做为中继速度。

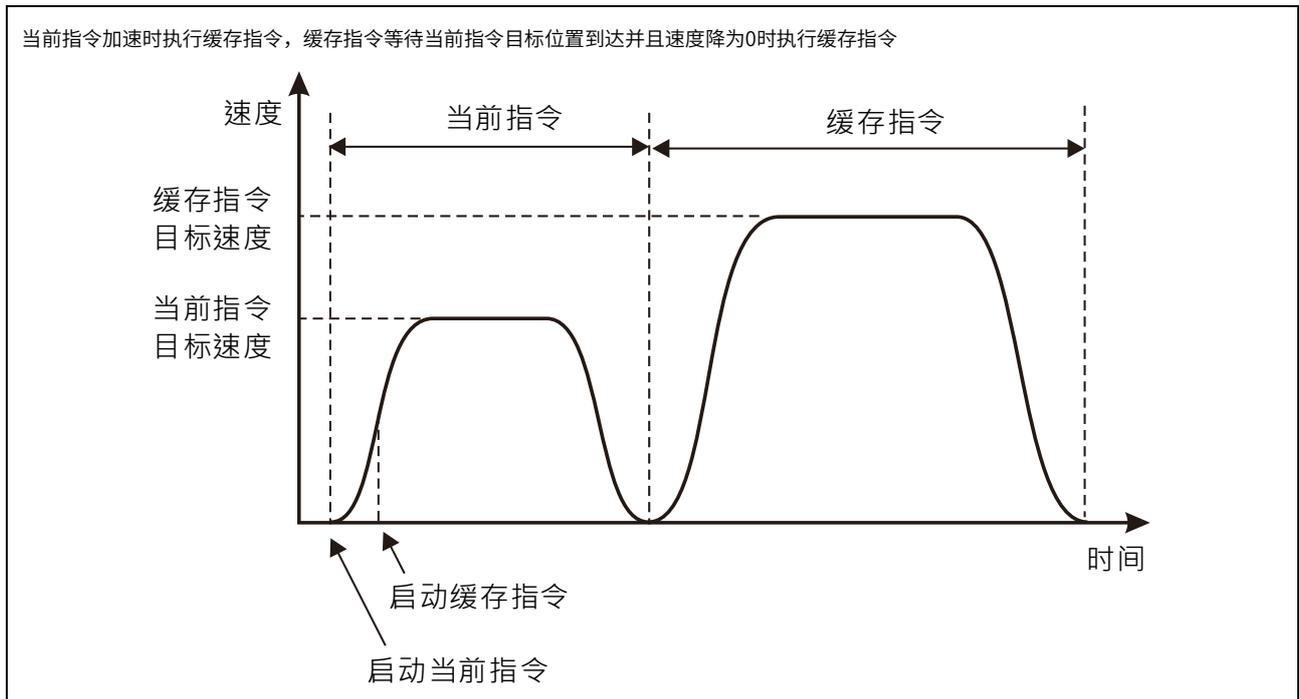
### ■ mcAborting（中断）

缓存模式的默认模式，多重启动时，缓存指令立即控制轴，无缓存。



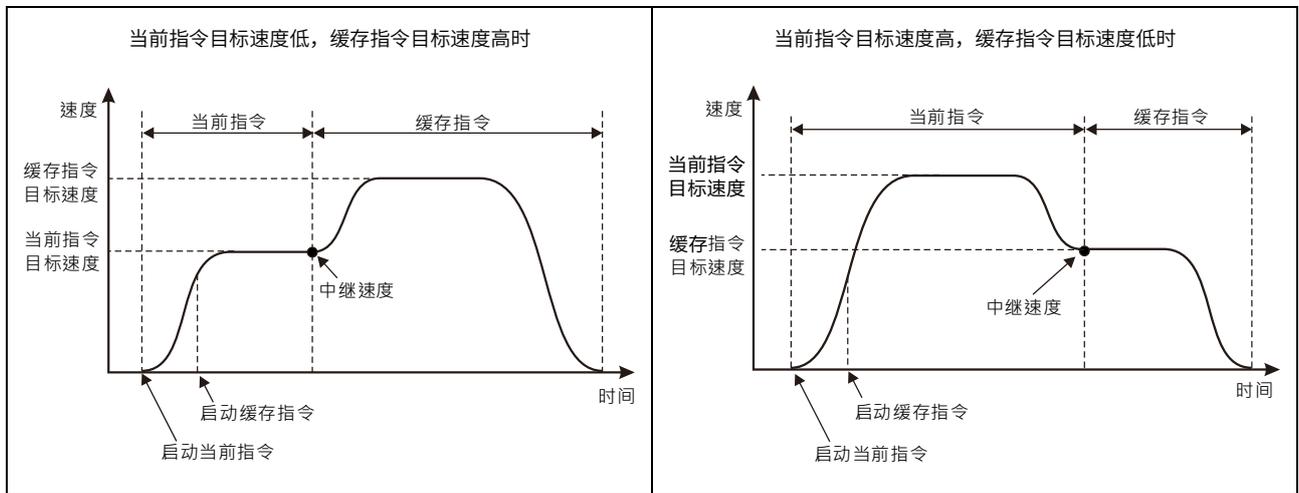
■ mcBuffered (等待)

等待当前执行的指令完成时（如目标位置或者目标速度到达），切换为缓存指令控制轴。



■ mcBlendingLow (以低速中继)

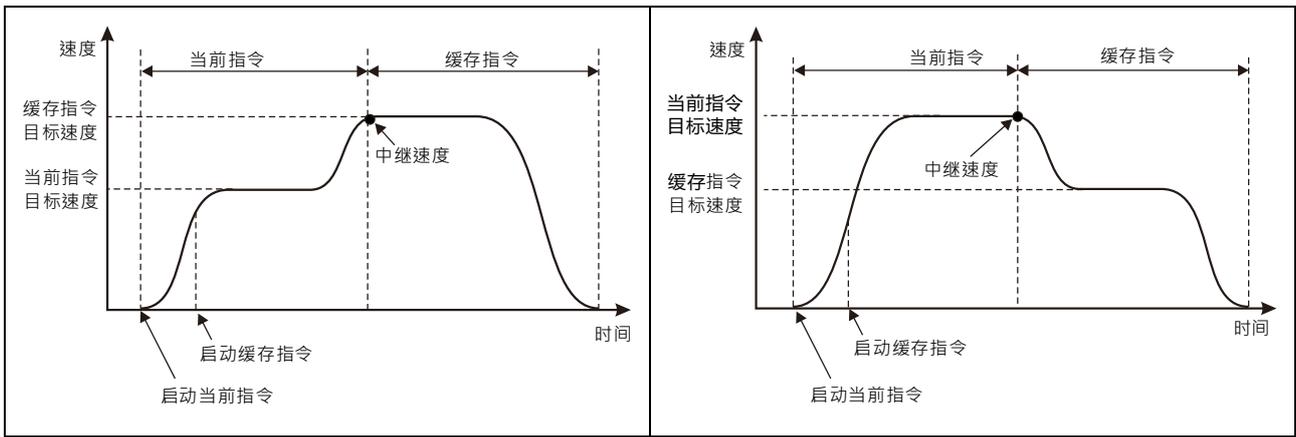
比较当前指令的目标速度和缓存指令的目标速度大小，当前指令目标位置到达时，以速度较低者为中继速度。



■ mcBlendingHigh (以高速中继)

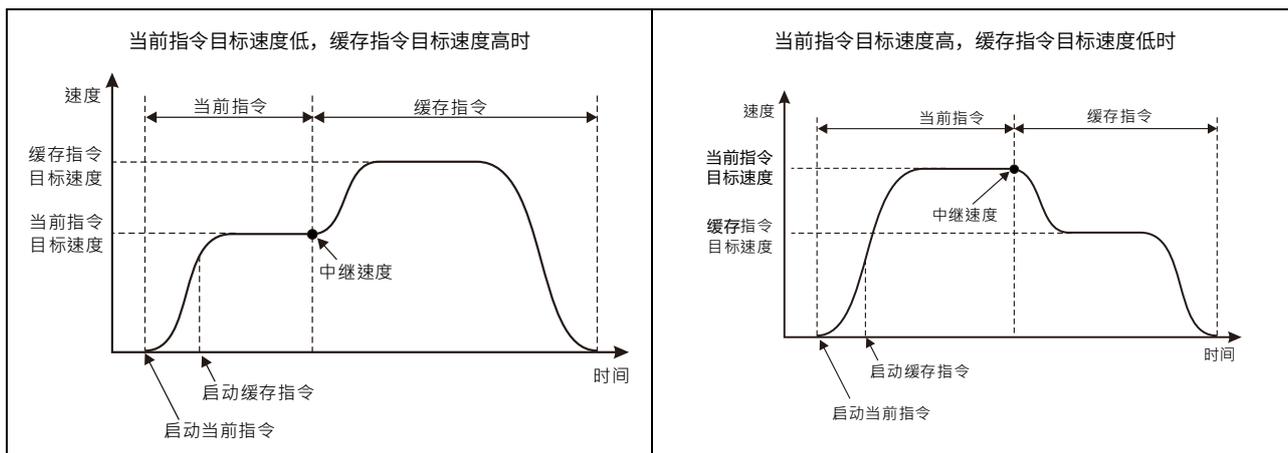
比较当前指令的目标速度和缓存指令的目标速度大小，当前指令目标位置到达时，以速度较高者为中继速度。

当前指令目标速度低，缓存指令目标速度高时	当前指令目标速度高，缓存指令目标速度低时
----------------------	----------------------



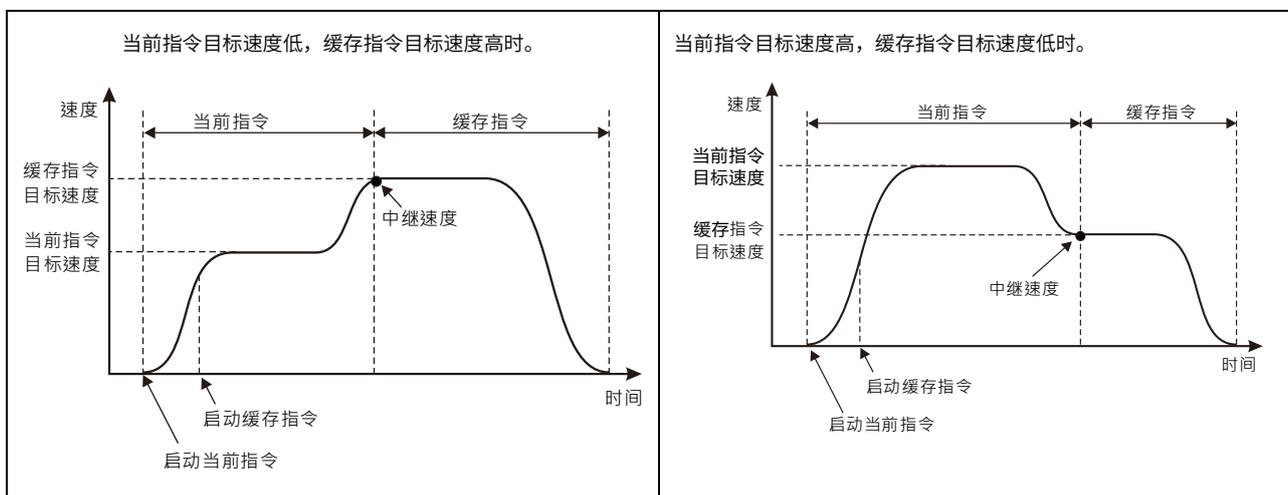
■ mcBlendingPrevious (以当前指令的目标速度中继)

当前指令目标位置到达时，以当前指令的目标速度为中继速度。



■ mcBlendingNext (以缓存指令的目标速度中继)

当前指令目标位置到达时，以缓存指令的目标速度为中继速度。



插补指令BufferMode介绍

BufferMode值	TransitionMode值	说明
1: mcBuffered (等待)	0: mcTMNone(无过渡曲线)	等待当前插补指令执行完成后 (位置到达, 速度降为0), 执行缓存插补指令。
3: mcBlending-Previous (以当前指令目标速度中继)	2: mcTMConstantVelocity(以设定的速度过渡)	等待当前插补指令执行完成, 执行缓存插补指令, 交接速度为当前插补指令的合成速度。
	3: mcTMCornerDistance(以附加角过渡)	当前插补指令完成, 速度不降为0的情况下执行缓存插补指令, 当前插补指令的终点位置和缓存插补指令的起点位置之间为圆弧过渡, 过渡时插补速度不降为0。

■ 各指令支持的缓存模式

缓存指令和当前指令进行缓存时，缓存指令和当前指令如何缓存由缓存指令的BufferMode (缓存模式) 引脚设置，缓存指令BufferMode (缓存模式) 可以设置的值和当前指令及缓存指令都有关系。

- 情况 1: 若 MC\_MoveAbsolute 指令为当前指令, MC\_MoveVelocity 指令为缓存指令, 则 MC\_MoveVelocity 指令的 BufferMode (缓存模式) 引脚可以设置的缓存模式为 mcAborting、mcBuffered、mcBlendingLow、mcBlendingPrevious、mcBlendingNext、mcBlendingHigh。

- 情况 2: 若 MC\_MoveVelocity 指令为当前指令, MC\_Absolute 指令为缓存指令, 则 MC\_MoveRelative 指令的 BufferMode (缓存模式) 引脚可以设置的缓存模式为 mcAborting 和 mcBuffered。

根据当前指令, 缓存指令可以选择的 BufferMode (缓存模式) 列表如下:

当前指令	缓存指令可以选择的 BufferMode (缓存模式)
MC_MoveAbsolute	【mcAborting、mcBuffered、mcBlendingLow、mcBlendingPrevious、mcBlendingNext、mcBlendingHigh】*1
MC_MoveRelative	【mcAborting、mcBuffered、mcBlendingLow、mcBlendingPrevious、mcBlendingNext、mcBlendingHigh】*1
MC_MoveAdditive	【mcAborting、mcBuffered、mcBlendingLow、mcBlendingPrevious、mcBlendingNext、mcBlendingHigh】*1
MC_MoveSuperimposed	mcAborting
MC_HaltSuperimposed	mcAborting
MC_MoveVelocity	mcAborting、mcBuffered
MC_Home	只有 MC_Stop 指令可以打断 MC_Home 指令
MC_Halt	mcAborting、mcBuffered
MC_GearIn	mcAborting、mcBuffered
MC_GearOut	mcAborting、mcBuffered
MC_CombineAxes	mcAborting、mcBuffered
MC_CamIn	mcAborting、mcBuffered
MC_CamOut	mcAborting、mcBuffered

\*1: 缓存指令为 MC\_GearIn、MC\_CamIn、MC\_CombineAxes 时, BufferMode 只能选择 mcAborting 和 mcBuffered。

当前指令是否执行完成要看各个指令的完成位进行判断, 完成位为 TRUE 表示该指令执行完成, 并且开始执行 Buffer 指令。下表即说明各个指令的完成位是哪一个, 便于在选择 BufferMode 模式时进行判断。

指令名称	是否是缓存指令	是否可以跟随一个缓存指令	完成位
MC_MoveAbsolute	是	是	Done
MC_MoveRelative	是	是	Done
MC_MoveAdditive	是	是	Done
MC_Home	否	否	Done
MC_Stop	否	否	Done
MC_Halt	是	是	Done
MC_MoveSuperimposed	否	否	---
MC_HaltSuperimposed	否	否	---
MC_MoveVelocity	是	是	InVelocity
MC_CamIn	是	是	EndOfProfile
MC_CamOut	否	是	Done
MC_GearIn	是	是	InGear
MC_GearOut	否	是	Done
MC_CombineAxes	是	是	InSync

## 3.3 不同机种支持的轴规格及运动控制功能

机种名称	支持轴号范围及单轴数量							电子凸轮 编号范围	凸轮点编 号范围	旋切编 号范围	轴组数量范 围	CNC功能
	轴号范围	所有轴 数量	虚拟伺服 轴数量	伺服轴数 量	脉冲轴 数量	编码器 轴数量	虚拟编码 器轴数量					
HCM511S-32MT4-D	1~16	16	16	8	4	2	16	1~16	1~32	1~2	1	不支持
HCM501S-16MT4-D	1~16	16	16	8	4	2	16	1~16	1~32	1~2	1	不支持
HCM301-16MT4-D	1~16	16	16	不支持	4	2	16	不支持	不支持	不支持	不支持	不支持
HCM302-16MT4-D	1~16	16	16	不支持	4	2	16	1~16	1~32	1~2	1	不支持
HCM310-16MT4-D	1~16	16	16	不支持	4	2	16	不支持	不支持	不支持	不支持	不支持
HCM311-16MT4-D	1~16	16	16	不支持	4	2	16	1~16	1~32	1~2	1	不支持
HCM312-32MT6-D	1~16	16	16	不支持	6	2	16	1~16	1~32	1~2	1	不支持



## 第四章 单轴指令



## 4.1 MC\_Power (使能运行/解除运行)

控制指定的轴处于运行或解除运行状态。所属库：MotionControl

指令	名称	FB/FUN	梯形图样式	ST样式
MC_Power	使能运行/ 解除运行	FB	<pre> graph LR     subgraph MC_Power_Instance         direction TB         Axis --- MC_Power_Instance         Enable --- MC_Power_Instance         EnablePositive --- MC_Power_Instance         EnableNegative --- MC_Power_Instance         BufferMode --- MC_Power_Instance         MC_Power_Instance --- Status         MC_Power_Instance --- Busy         MC_Power_Instance --- Active         MC_Power_Instance --- Error         MC_Power_Instance --- ErrorID     end </pre>	<pre> MC_Power_Instance ( Axis := 参数 , Enable := 参数 , EnablePositive:= 参数 , EnableNegative:= 参数 , BufferMode:= 参数 , Status =&gt; 参数 , Busy =&gt; 参数 , Active=&gt; 参数 , Error =&gt; 参数 , ErrorID =&gt; 参数 ); </pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Enable	有效	BOOL	TRUE或FALSE	FALSE	设置为TRUE, 控制轴进入运行状态 设置为FALSE, 解除轴的运行状态
EnablePositive	正转有效	BOOL	TRUE或FALSE	FALSE	设置为TRUE, 允许轴正转 设置为FALSE, 禁止轴正转
EnableNegative	反转有效	BOOL	TRUE或FALSE	FALSE	设置为TRUE, 允许轴反转 设置为FALSE, 禁止轴反转
BufferMode	缓存模式	MC_Buffer_Mode	0: mcAborting 1: mcBuffered	0	Enable由TRUE变为FALSE时, 指定轴解除运行状态

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Status	可运行	BOOL	TRUE或FALSE	该输出变量为TRUE时, 表示轴状态进入运行状态 该输出变量为FALSE时, 表示轴状态处于解除运行状态
Busy	执行中	BOOL	TRUE或FALSE	该输出变量为TRUE时表示指令正在执行中
Active	控制中	BOOL	TRUE或FALSE	该输出变量为TRUE时表示指令正在控制轴
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时, 输出错误代码 值的含义请参考“ <a href="#">指令错误代码描述</a> ”

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Status	指定轴进入运行状态时	指定轴解除运行状态时Error变为TRUE时
Busy	Enable为TRUE时	当Error为TRUE时
Active	指令开始控制轴时	当Error为TRUE时
Error	指令输入变量值不在允许的范围	Enable由TRUE变为FALSE时

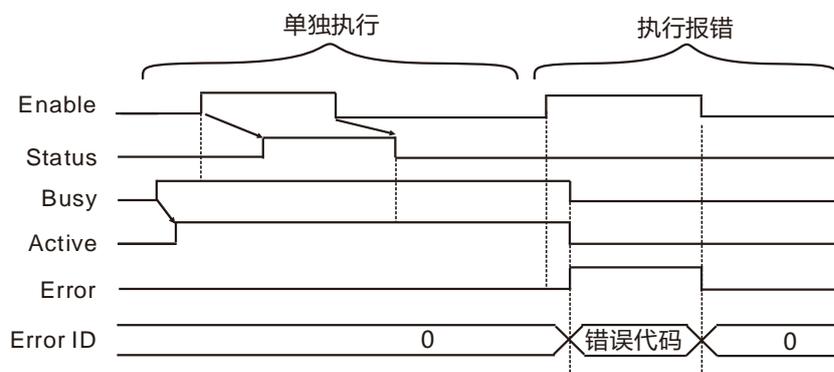
■ 功能说明

- 该指令用于控制指定的轴使能运行或解除运行状态。该指令可以对伺服轴、虚拟伺服轴、脉冲轴操作，不可以对编码器轴操作。虚拟伺服轴无解除运行状态，对虚拟伺服轴执行该指令，该指令的 Enable 为 TRUE 或者 FALSE 时，该指令的 Status 位都为 TRUE。
- 当 Enable 设为 TRUE 时，控制轴进入运行状态；Status 为 TRUE 时，表示该轴处于运行状态，轴接收动作指令，可以控制轴。
- 当 Enable 设为 FALSE 时，解除轴的运行状态；Status 为 FALSE 时，表示轴处于解除运行状态，轴不接收动作指令，不可以控制轴，MC\_Reset（轴异常复位）仍可以执行。
- EnablePositive 为 TRUE 时，允许轴正转，运动指令可以控制轴正转；为 FALSE 时，不可以控制轴正转，运动指令控制轴正转时会报错；为 FALSE 时，运动指令控制轴从反转变为正转时，正在执行的运动指令会被中断，轴按照被中断指令设定的减速度、跃度减速停止，轴停止后进入 Standstill 状态。
- 运动指令控制轴正转时，EnablePositive 由 TRUE 变为 FALSE，正在执行的运动指令会被中断，轴按照被中断指令设定的减速度、跃度减速停止，轴停止后进入 Standstill 状态。
- EnableNegative 为 TRUE 时，允许轴反转，运动指令可以控制轴反转；为 FALSE 时，不可以控制轴反转，运动指令控制轴反转时会报错；为 FALSE 时，运动指令控制轴从正转变为反转时，正在执行的运动指令会被中断，轴按照被中断指令设定的减速度、跃度减速停止，轴停止后进入 Standstill 状态。
- 运动指令控制轴反转时，EnableNegative 由 TRUE 变为 FALSE，正在执行的运动指令会被中断，轴按照被中断指令设定的减速度、跃度减速停止，轴停止后进入 Standstill 状态。
- BufferMode

输入变量 Enable 从 TRUE 变为 FALSE 时，可以通过 BufferMode 设定不同的值，选择解除运行的方式。

BufferMode 设定值	含义
mcAborting (中断)	轴在静止状态 (standstill) 时，Enable由TRUE变为FALSE时，立即解除轴运行状态，轴状态进入Disable状态。 轴在运转时，Enable从TRUE变为FALSE时，立即解除轴运行状态，轴状态进入errostop状态，需要执行MC_Reset指令复位才可以再控制轴。 <b>注意：</b> 轴运转时，Enable从TRUE变为FALSE控制轴解除运行，可能会造成设备碰撞或者人员伤亡，使用时请务必了解现场设备是否可以执行此操作。
mcBuffered (等待)	轴在静止状态 (standstill) 时，Enable由TRUE变为FALSE时，轴会立即解除运行状态，轴状态进入Disable状态。 轴在运转时，Enable从TRUE变为FALSE时，不会解除轴运行状态，直到轴状态变为静止状态 (standstill) 时，才会解除轴运行状态，轴状态进入Disable状态。

■ 输出变量变化时序说明



- **单独执行**

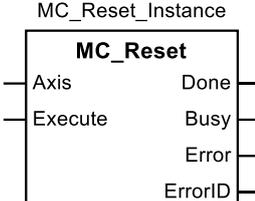
该指令在 Enable（有效）为 FALSE 或者 TRUE 时都会执行，因为该指令有运行有效和解除运行两个功能。任务执行后，Busy（执行中）就会变为 TRUE，下一个周期 Active（控制中）变为 TRUE。Enable 为 TRUE 后，控制指定轴运行，进入运行状态后，Status 变为 TRUE，Busy（执行中）和 Active（控制中）仍为 TRUE。Enable 变为 FALSE 后，控制指定轴解除运行，解除运行后，Status 变为 FALSE，Busy（执行中）和 Active（控制中）仍为 TRUE。

- **执行报错**

该指令输入变量值不在允许范围内时（如轴号超过范围），该指令 Error（错误）变为 TRUE，ErrorID（错误码）输出对应的错误码，可通过 ErrorID（错误码）的值，查找发生问题原因，同时 Busy（执行中）和 Active（控制中）变为 FALSE。该指令 Enable（有效）由 TRUE 变为 FALSE 时，同时 Error（错误）变为 FALSE，ErrorID（错误码）的值变为 0。

## 4.2 MC\_Reset (轴异常复位)

指定轴有报警、状态机异常或者位置误差异常时，通过该指令复位。所属库：MotionControl

指令	名称	FB/FUN	梯形图样式	ST样式
MC_Reset	轴异常复位	FB		<pre>MC_Reset_Instance ( Axis := 参数 , Execute := 参数 , Done =&gt; 参数 , Busy =&gt; 参数 , Error =&gt; 参数 , ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

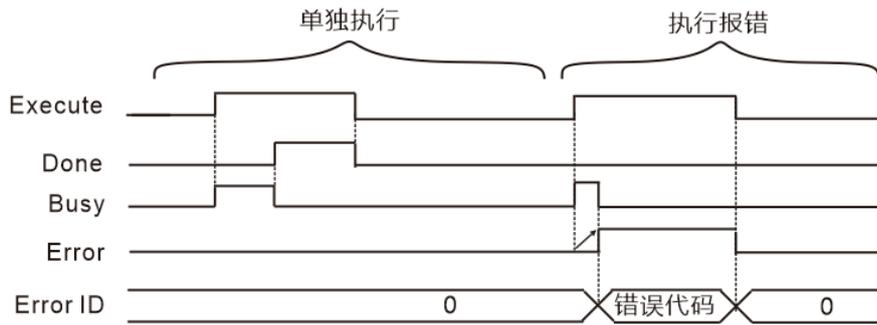
### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	轴异常解除时	<ul style="list-style-type: none"> <li>◆ Done为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE</li> </ul>
Busy	Execute的上升沿	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> </ul>
Error	指令输入变量值不在允许的范围或者轴异常不能正常复位时	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Execute从TRUE变为FALSE时</li> </ul>

### ■ 功能说明

- 该指令用于复位轴异常，指定轴有报警、状态机异常或者位置误差超过设定值等异常时，均可通过该指令进行复位。
- 指定轴有异常时，才可以通过该指令复位。Execute（有效）由 FALSE 变为 TRUE 时，开始对指定轴的异常进行处理。该指令对轴异常进行复位时，有时需要多个周期才可以复位完成。
- 驱动器有报警或者错误时，建议先将驱动器的报警或者错误消除后，再执行该指令进行复位。
- 伺服轴、虚拟伺服轴、脉冲轴、编码器轴都可以通过该指令进行复位。
- 该指令执行时，可以搭配 MC\_ReadStatus（读取轴状态）指令使用。MC\_ReadStatus 指令读取轴的状态进入 ErrorStop(错误停止)状态时，执行该指令。比较常见的异常为指定轴命令位置和反馈位置的差值超过软件中的设定值，导致轴进入 ErrorStop(错误停止)状态。

### ■ 输出变量变化时序说明



● 单独执行

当轴有异常时，Execute（启动）由 FALSE 变为 TRUE 时，Busy（执行中）同时变为 TRUE，轴异常消除时 Done（完成）变为 TRUE，同时 Busy（执行中）变为 FALSE。当 Execute（启动）变为 FALSE 时，Done（完成）同时变为 FALSE。

● 执行报错

该指令输入变量值不在允许范围内时，该指令 Execute（启动）由 FALSE 变为 TRUE 时，同时 Busy（执行中）变为 TRUE，下一个周期 Error（错误）变为 TRUE，同时 Busy（执行中）变为 FALSE，ErrorID（错误码）输出对应的错误码，可通过 ErrorID（错误码）的值，查找发生问题原因。该指令 Execute（启动）由 TRUE 变为 FALSE 时，同时 Error（错误）变为 FALSE，ErrorID（错误码）的值变为 0。

■ 示例程序

● 实现功能

当轴遇到报警或其它错误时，可通过执行 MC\_Reset（轴异常复位）对轴报警或者控制器内部报警进行复位。

● 轴参数设置

轴 1 轴参数设置如下图所示。



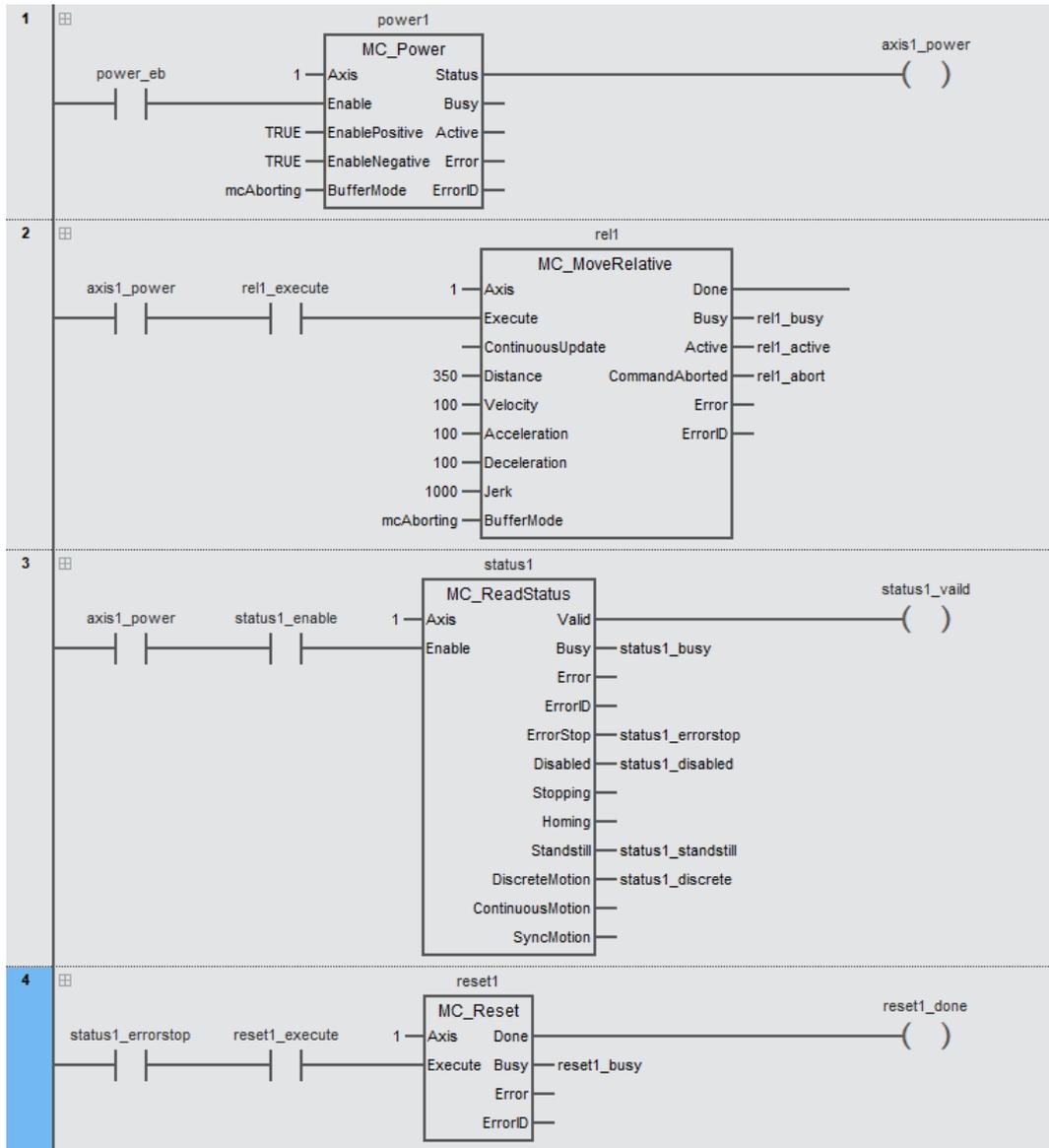
轴命令位置和反馈位置允许的位置误差如下图红色方框处所示，设置值为 150，下图红色方框处“位置跟随检测”处设置为 TRUE。



● 变量表

类别	名称	分配到	数据类型	初始值	注释
VAR	power_eb		BOOL		
VAR	power1		MC_Power		
VAR	axis1_power		BOOL		
VAR	rel1_execute		BOOL		
VAR	rel1		MC_MoveRelative		
VAR	rel1_done		BOOL		
VAR	rel1_busy		BOOL		
VAR	rel1_active		BOOL		
VAR	rel1_abort		BOOL		
VAR	reset1_execute		BOOL		
VAR	reset1		MC_Reset		
VAR	reset1_done		BOOL		
VAR	reset1_busy		BOOL		
VAR	status1		MC_ReadStatus		
VAR	status1_enable		BOOL		
VAR	status1_vaiId		BOOL		
VAR	status1_busy		BOOL		
VAR	status1_disabled		BOOL		
VAR	status1_standstill		BOOL		
VAR	status1_discrete		BOOL		
VAR	status1_errorstop		BOOL		

● 梯形图(LD)



● 结构化文本(ST)

```

power1(
    Axis:=1,
    Enable:=power_eb,
    EnablePositive:=TRUE,
    EnableNegative:=TRUE,
    BufferMode:=mcAborting,
    Status=>axis1_power
);

rel1(
    Axis:=1,
    Execute:=axis1_power AND rel1_execute,
    Distance:=350,
    Velocity:=100,
    Acceleration:=100,
    Deceleration:=100,
    Jerk:=1000,
    BufferMode:=mcAborting,
    Done=>rel1_done,
    Busy=>rel1_busy,

```

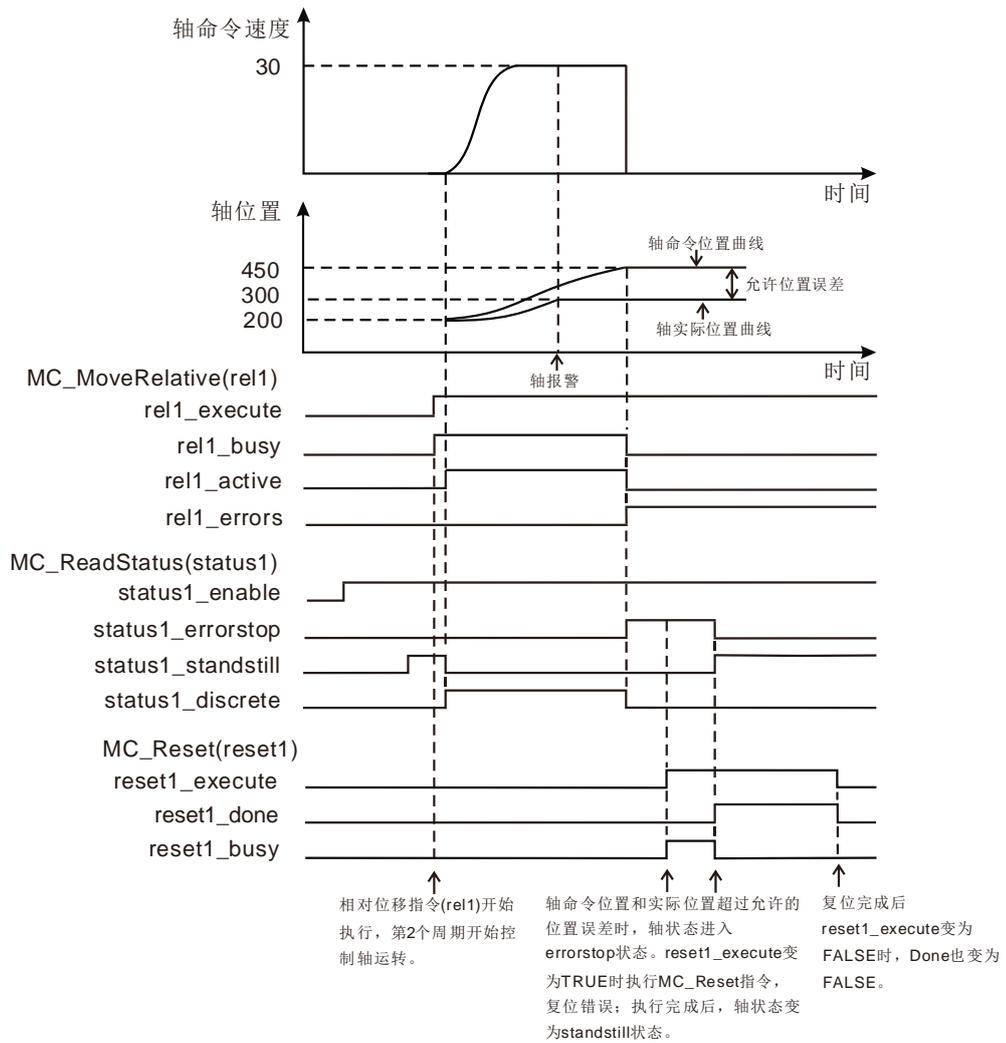
```
Active=>rel1_active ,
CommandAborted=>rel1_abort
);

status1(
Axis:=1 ,
Enable:= axis1_power AND status1_enable ,
Valid=> status1_vaild,
Busy=> status1_busy ,
ErrorStop=> status1_errorstop,
Disabled=> status1_disabled ,
Standstill=> status1_standstill ,
DiscreteMotion=> status1_discrete
);

reset1(
Axis:=1 ,
Execute:=status1_errorstop AND reset1_execute ,
Done=>reset1_done ,
Busy=>reset1_busy
);
```

#### ● 程序说明

- 轴使能后，re1\_execute变为TRUE时，相对位移指令（rel1）开始执行，re1\_execute变为TRUE的第2个周期开始控制轴。
- 轴运转过程中如遇到极限报警等时，轴命令位置继续增加，轴实际位置不再增加，当轴命令位置和实际位置的差值超过软件中“轴设置”→“操作设置”内设置的允许位置误差时，轴状态进入errorstop状态。reset1\_execute变为TRUE时，执行复位指令，复位指令完成后，轴状态从errorstop变为standstill状态，轴命令位置变为轴实际位置的值。
- reset1\_execute变为FALSE时，复位指令的Done也变为FALSE，为下次再次执行复位指令做准备。



## 4.3 MC\_Jog (点动)

控制指定的轴进行点动。所属库：MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_Jog	点动	FB		<pre> MC_Jog_Instance ( Axis := 参数 , JogForward:= 参数 , JogBackward:= 参数 , Velocity:= 参数 , Acceleration:= 参数 , Deceleration:= 参数 , Jerk:= 参数 , Done =&gt; 参数 , Busy =&gt; 参数 , CommandAborted =&gt; 参数 , Error =&gt; 参数 , ErrorID=&gt; 参数 ); </pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
JogForward	正向点动	BOOL	TRUE或FALSE	FALSE	设为TRUE, 控制轴正向移动 设为FALSE, 则停止正向移动
JogBackward	反向点动	BOOL	TRUE或FALSE	FALSE	设为TRUE, 控制轴反向移动 设为FALSE, 则停止反向移动
Velocity	目标速度	LREAL	正数	不可缺省	指定目标速度*1 (单位: 行程单位/秒) *2
Acceleration	加速度	LREAL	正数	不可缺省	指定加速度*1 (单位: 行程单位/秒 <sup>2</sup> ) *2
Deceleration	减速度	LREAL	正数	不可缺省	指定减速度*1 (单位: 行程单位/秒 <sup>2</sup> ) *2
Jerk	跃度	LREAL	正数	不可缺省	指定跃度*1 (单位: 行程单位/秒 <sup>3</sup> ) *2

\*1: Velocity、Acceleration、Deceleration、Jerk的关系请参考“[运动控制指令相关参数说明](#)”章节。

\*2: 指令单位的详细介绍请参考“[运动控制指令参数单位](#)”章节。

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	点动停止时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时, 输出错误代码 值的含义请参考“ <a href="#">指令错误代码描述</a> ”

## ■ 输出变量刷新时机

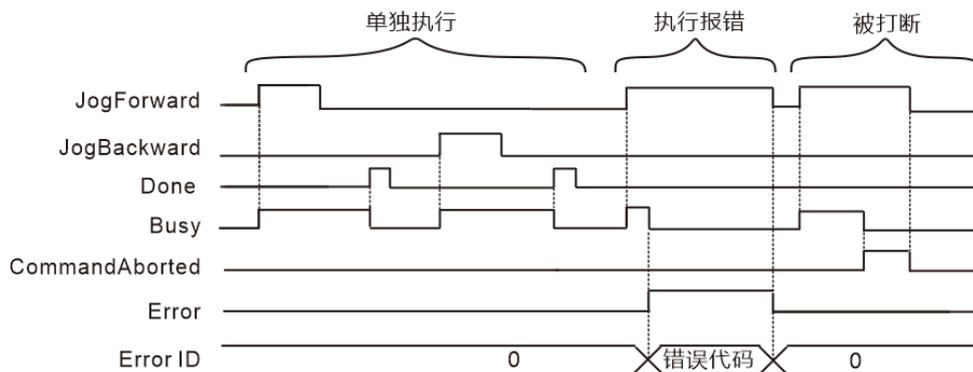
名称	变为TRUE的时机	变为FALSE的时机
Done	点动结束后, Done变为TRUE一个周期	◆ 指定轴停止后, Done变为TRUE一个周期后变为FALSE
Busy	JogForward 或 JogBackward的上升沿。	◆ Done变为TRUE时 ◆ Error变为TRUE时 ◆ CommandAborted变为TRUE时
CommandAborted	指令的执行被中断时。	◆ CommandAborted为TRUE, JogForward 或 JogBackward由TRUE变为FALSE时。 ◆ JogForward 或 JogBackward为FALSE时, 该指令被其它指令中断时, CommandAborted变为TRUE一个周期后, 变为FALSE。
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常。	◆ Execute由TRUE变为FALSE时

## ■ 功能说明

### ● 基本功能说明

该指令用于控制指定的轴根据输入变量进行正向点动或者反向点动。JogForward 为 TRUE 时, 轴正向运行, JogForward 为 FALSE 时, 轴停止运行。JogBackward 为 TRUE 时, 轴反向运行, JogBackward 为 FALSE 时, 轴停止运行。JogForward 和 JogBackward 不能同时为 TRUE。

## ■ 输出变量变化时序说明



### ● 单独执行

JogForward (正向点动) 或 JogBackward (反向点动) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE; JogForward (正向点动) 或 JogBackward (反向点动) 由 TRUE 变为 FALSE 时, 轴按照输入变量设定值, 减速停止, 轴停止且轴状态变为 Standstill 后, Done (完成) 由 FALSE 变为 TRUE, 一个周期后, 再变为 FALSE。

### ● 被中断

该指令执行后被其它指令中断时, 该指令 CommandAborted (中断) 变为 TRUE, Busy (执行中) 和 Active (控制中) 同时变为 FALSE; 当 Execute (执行中) 变为 FALSE 时, CommandAborted (中断) 同时变为 FALSE。

### ● 执行报错

该指令输入变量值不在允许范围内时, 该指令 JogForward (正向点动) 或 JogBackward (反向点动) 由 FALSE 变为 TRUE 时, 同时 Busy (执行中) 变为 TRUE, 下一个周期 Error (错误) 变为 TRUE, 同时 Busy (执行中) 变为 FALSE, ErrorID (错误码) 输出对应的错误码, 可通过 ErrorID (错误码) 的值, 查找发生问题原因。该指令 JogForward (正向点动) 或 JogBackward (反向点动) 由 TRUE 变为 FALSE 时, 同时 Error (错误) 变为 FALSE, ErrorID (错误码) 的值变为 0。

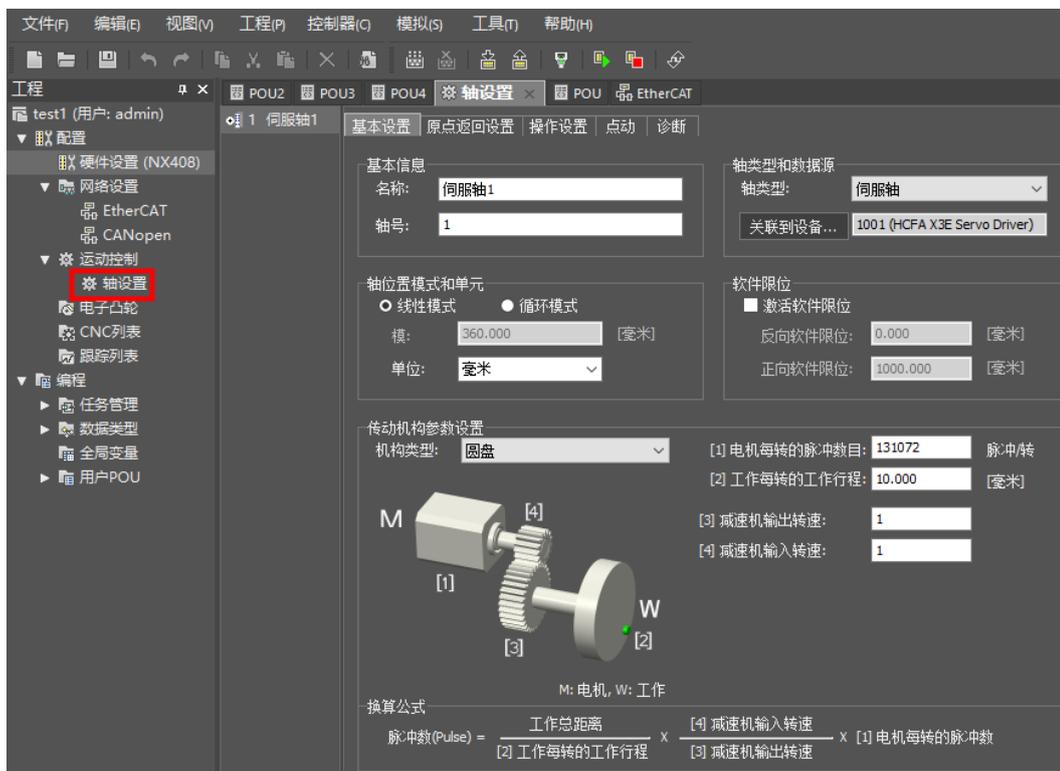
■ 示例程序

● 实现功能

轴使能后，通过 MC\_JOG（点动）指令控制轴正转点动和反转点动。正转按钮按下时，轴正转；正转按钮松开时，轴减速停止。反转按钮按下时，轴反转；反转按钮松开时，轴减速停止。

● 轴参数设置

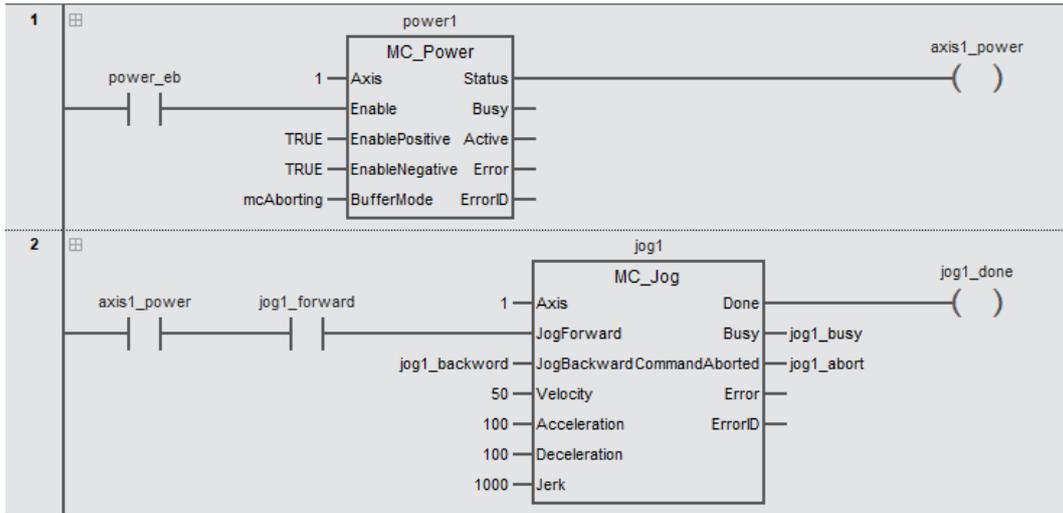
轴 1 轴参数设置如下图所示。



● 变量表

类别	名称	分配到	数据类型	初始值	注释
VAR	power_eb		BOOL		
VAR	power1		MC_Power		
VAR	axis1_power		BOOL		
VAR	jog1_forward	%IX0.0	BOOL		
VAR	jog1_backward	%IX0.1	BOOL		
VAR	jog1		MC_Jog		
VAR	jog1_done		BOOL		
VAR	jog1_busy		BOOL		
VAR	jog1_abort		BOOL		

## ● 梯形图(LD)



## ● 结构化文本(ST)

```

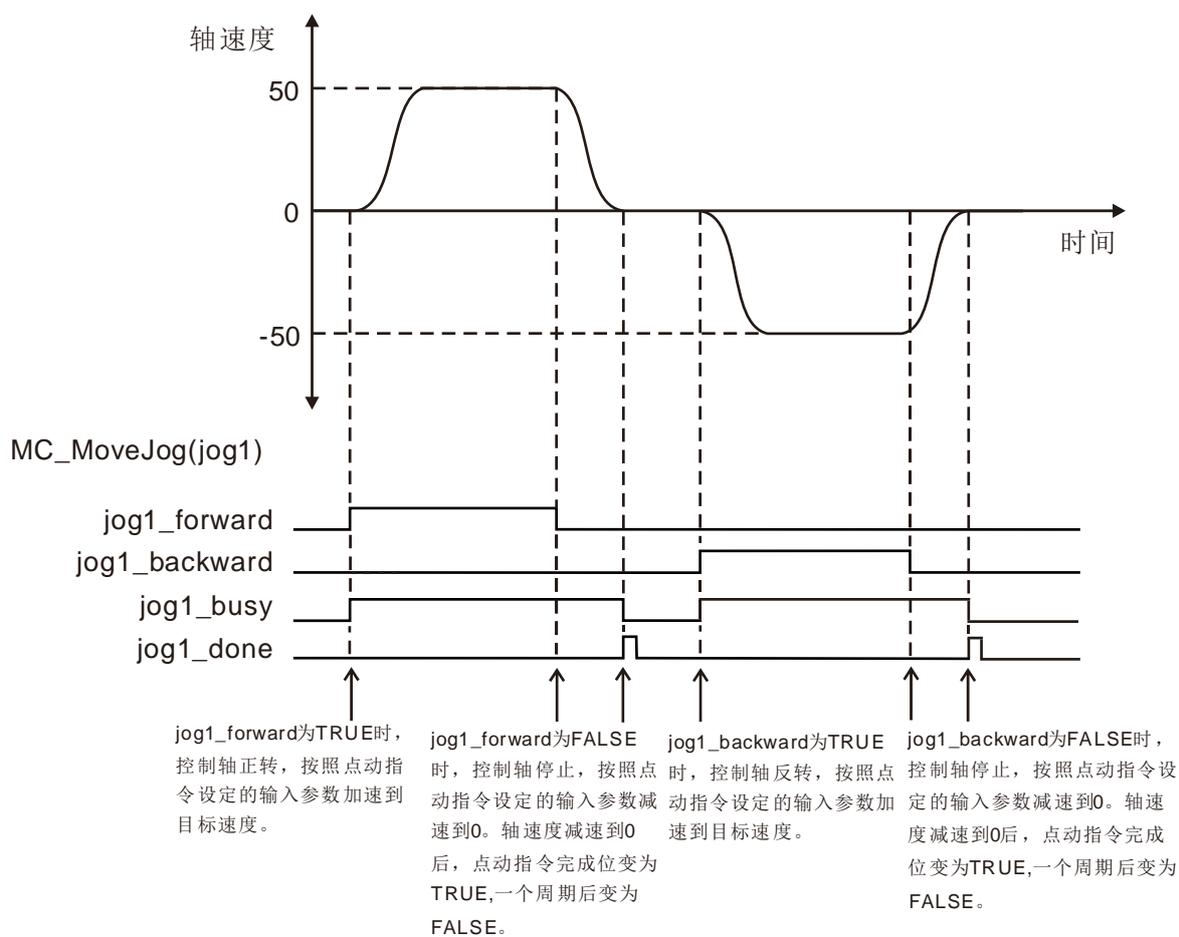
power1(
  Axis:=1,
  Enable:=power_eb,
  EnablePositive:=TRUE,
  EnableNegative:=TRUE,
  BufferMode:=mcAborting,
  Status=>axis1_power
);

jog1(
  Axis:=1,
  JogForward:=axis1_power AND jog1_forward,
  JogBackward:=jog1_backward,
  Velocity:=50,
  Acceleration:=100,
  Deceleration:=100,
  Jerk:=1000,
  Done=>jog1_done,
  Busy=>jog1_busy,
  CommandAborted=>jog1_abort
);

```

## ● 程序说明

- 轴使能后，通过外部按钮控制轴正转点动和反转点动。外部正转按钮接控制器的输入点%IX0.0,对应的变量为 jog1\_forward；外部反转按钮接控制器的输入点%IX0.1,对应的变量为 jog1\_backward。
- 正转按钮按下时，轴正转，按照MC\_Jog指令设定的输入参数加速到目标速度；正转按钮松开时，轴减速停止。
- 反转按钮按下时，轴反转，按照MC\_Jog指令设定的输入参数加速到目标速度；反转按钮松开时，轴减速停止。



## 4.4 MC\_HomeByPLCIO (通过控制输入信号的原点设置)

控制指定轴使用控制器的原点返回模式、原点信号、极限信号等确定机械原点。所属库：MotionControl\_Part2。

指令	名称	FB/FUN	梯形图样式	ST样式
MC_HomeByPLCIO	通过控制输入信号的原点设置	FB		<pre>MC_HomeByPLCIO_Instance( Axis:= 参数, Execute:= 参数, Position:= 参数, HomeMode:= 参数, HomeSignal:= 参数, ZPhaseSignal:= 参数, NegaLimitSignal:= 参数, PosiLimitSignal:= 参数, VelocityFast:= 参数, VelocitySlow:= 参数, Acceleration:= 参数, Deceleration:= 参数, Jerk:= 参数, BufferMode:= 参数, Options:= 参数, Done=&gt; 参数, Busy=&gt; 参数, Active=&gt; 参数, CommandAborted=&gt; 参数, Error=&gt; 参数, ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
Position	原点位置	LREAL	负数、正数、0	0	轴找到原点后的设定位置（单位：行程单位）
HomeMode	原点返回模式	INT	17-30、35	0	轴回到原点位置的不同方式 详细请参考 <a href="#">附录1 原点返回模式</a>
HomeSignal	原点信号	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时，根据不同的原点返回模式，轴执行相应动作（该信号需要接在控制器上）
ZPhaseSignal	Z脉冲信号	BOOL	TRUE或FALSE	FALSE	暂时无法实现
NegaLimitSignal	负极限信号	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时，根据不同的原点返回模式，轴执行相应动作（该信号需要接在控制器上）
PosiLimitSignal	正极限信号	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时，根据不同的原点返回模式，轴执行相应动作（该信号需要接在控制器上）
VelocityFast	原点返回开始速度	LREAL	负数、正数、0	0	设定寻找原点时的速度
VelocityLow	原点返回接近速度	LREAL	负数、正数、0	0	遇到原点信号后减速停止时的速度
Acceleration	加速度	LREAL	负数、正数、0	0	指定加速度*1 (单位：行程单位/秒 <sup>2</sup> ) *2
Deceleration	减速度	LREAL	负数、正数、0	0	指定减速度*1 (单位：行程单位/秒 <sup>2</sup> ) *2
Jerk	跃度	LREAL	负数、正数、0	0	指定跃度*1 (单位：行程单位/秒 <sup>3</sup> ) *2
BuffMode	缓存模式	MC_Buffer_Mode	0: mcAborting	0	保留
Options	保留	保留	保留	保留	保留

\*1: Velocity、Acceleration、Deceleration、Jerk的关系请参考“[运动控制指令相关参数说明](#)”章节。

\*2: 指令单位的详细介绍请参考“[运动控制指令参数单位](#)”章节。

## ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	点动停止时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Active	控制中	BOOL	TRUE或FALSE	该输出变量为TRUE时表示指令正在控制轴
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

## ■ 输出变量刷新时期

名称	变为TRUE的时机	变为FALSE的时机
Done	指令执行完成后	<ul style="list-style-type: none"> <li>◆ Done为TRUE, Execute由TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE</li> </ul>
Busy	Execute 由FALSE变为TRUE时	<ul style="list-style-type: none"> <li>◆ Done变为TRUE时</li> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>
Active	指令开始控制轴时	<ul style="list-style-type: none"> <li>◆ Done变为TRUE时</li> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>
CommandAborted	指令的执行被中断时	<ul style="list-style-type: none"> <li>◆ Execute由TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时, 该指令被其它指令中断时, CommandAborted变为TRUE一个周期后, 变为FALSE</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Execute由TRUE变为FALSE时</li> </ul>

## ■ 功能说明

### ● 基本功能说明

- 该指令用于控制轴使用该指令设定的原点返回模式、原点信号、极限信号等确定机械原点。将原点开关、正向极限或反向极限开关接到控制器的本体输入点上以实现原点回归功能。（该指令指定的轴可以为伺服轴、虚拟伺服轴、脉冲轴、编码器轴）。
- 编码器轴只能设置为原点返回模式 35。
- 该指令只有轴状态处于StandStill状态时才可以执行，其他状态下执行时，该指令报错。

### ● 指令完成时机

- 当原点设置完毕后，该指令完成，Done由FALSE变为TRUE。

### ● 重启该指令

- 当指令执行完成后，Execute再次由FALSE变为TRUE时，该指令可以重新执行；当指令正在执行中，Execute再次由FALSE变为TRUE时，对指令的执行不会产生影响，指令仍按照未执行完成的输入变量执行指令。

### ● 其它指令执行时启动该指令

- 其它运动指令执行时，启动该指令，该指令会报错，只有轴状态处于StandStill状态时才可以执行。

### ● 该指令执行时启动其它指令

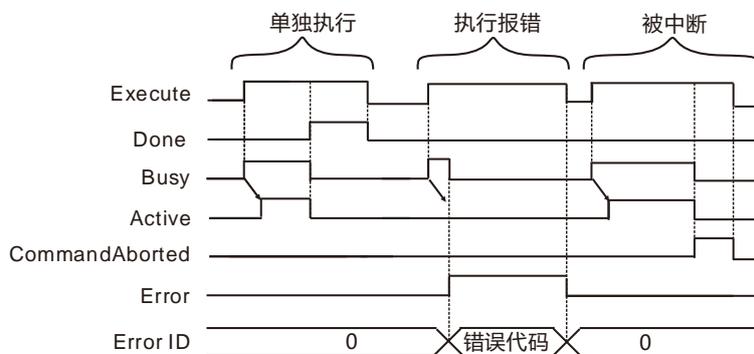
- 该指令执行时启动其它运动指令，其它运动指令和该指令如何中继由其它运动指令的BufferMode（缓存模式）引脚参数值决定，BufferMode（缓存模式）可以设定的参数值如下表所示。其他运动指令和该指令缓存时，其他指令执行的时机为该指令Done（完成）变为 TRUE时。如果其它运动指令无BufferMode引脚，一般为中断该指令。

缓存模式	含义
mcAborting (中断)	中断当前正在控制轴的指令，切换为缓存指令控制
mcBuffered (等待)	等待当前指令执行完成时（如目标位置或者目标速度到达），切换为缓存指令控制轴

### ● 异常排除

- 该指令执行时，如果指令输入变量值不在允许的范围或者不满足指令的执行条件，该指令Error变为TRUE，ErrorID中有对应的错误码，可以通过该错误码判断错误原因。该指令执行时，如遇到异常（如轴报警等），该指令也会报错，轴会立即停止。

### ■ 输出变量变化时序说明



### ● 单独执行

Execute（启动）由 FALSE 变为 TRUE 时，Busy（执行中）同时变为 TRUE，下一个周期 Active（控制中）变为 TRUE。指令完成时，Done（完成）变为 TRUE，Busy（执行中）和 Active（控制中）同时变为 FALSE。当 Execute（启动）变为 FALSE 时，Done（完成）同时变为 FALSE。

### ● 执行报错

该指令输入变量值不在允许范围内时，该指令 Execute（启动）由 FALSE 变为 TRUE 时，同时 Busy（执行中）变为 TRUE，下一个周期 Error（错误）变为 TRUE，同时 Busy（执行中）变为 FALSE，ErrorID（错误码）输出对应的错误码，可通过 ErrorID（错误码）的值，查找发生问题原因。该指令 Execute（启动）由 TRUE 变为 FALSE 时，同时 Error（错误）变为 FALSE，ErrorID（错误码）的值变为 0。

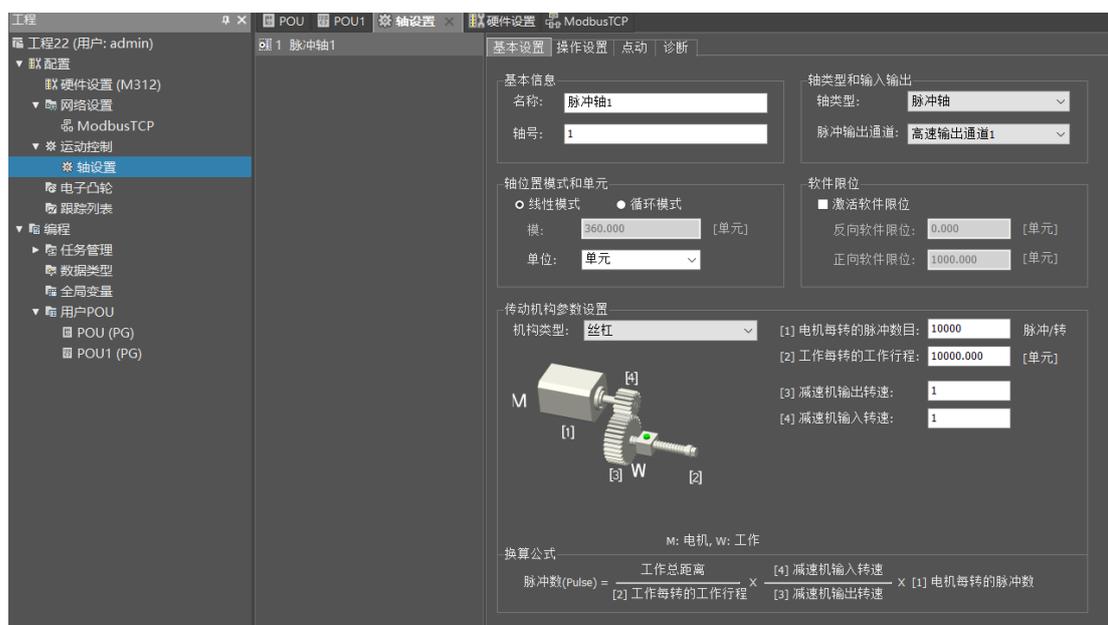
### ● 被中断

该指令输入变量值不在允许范围内时，该指令 Execute（启动）由 FALSE 变为 TRUE 时，Busy（执行中）同时变为 TRUE，下一个周期 Error（错误）变为 TRUE，ErrorID（错误码）输出对应的错误码，可通过 ErrorID（错误码）的值，查找发生问题原因。该指令 Execute（启动）由 TRUE 变为 FALSE 时，同时 Error（错误）变为 FALSE，ErrorID（错误码）的值变为 0。

### ■ 示例程序

#### ● 参数配置

轴 1 轴参数设置如下图所示



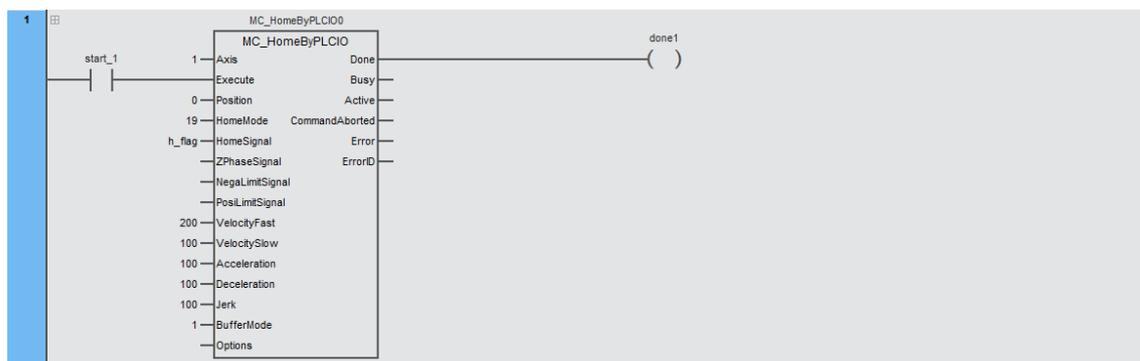
● 以原点返回模式 19 为例进行原点返回模式的功能说明

将外部传感器信号触发位置设定为用户指定的原点位置。原点返回模式和原点返回速度等通过 MC\_HomeByPLCIO 指令设置，外部传感器信号接控制器的输入点，该指令执行后，按照该指令 homemode（原点返回模式）引脚设定的原点返回模式执行回原点动作。

● 变量表

类别	名称	分配到	数据类型	初始值	注释
VAR	start_1		BOOL		
VAR	MC_HomeByPLCIO		MC_HomeByPLCIO		
VAR	h_flag		BOOL		
VAR	done1		BOOL		

● 梯形图(LD)



- 结构化文本(ST)

```
MC_HomeByPLCIO(  
Axis:= 1,  
Execute:= start_1,  
Position:= 0,  
HomeMode:= 19,  
HomeSignal:= h_flag,  
VelocityFast:= 200,  
VelocitySlow:= 100,  
Acceleration:= 100,  
Deceleration:= 100,  
Jerk:= 100,  
BufferMode:= 1,  
Done=> done1  
);
```

- 程序说明

- Start\_1变由FALSE变为TRUE时，根据MC\_HomeByPLCIO(通过控制输入信号的原点设置)设置的参数，执行原点返回动作。控制器输入点检测到原点信号从ON→OFF状态，原点设置完成。原点设置完成后，将外部传感器信号触发位置设定为用户指定的原点位置，原点设置完成标志会变TRUE，即MC\_HomeByPLCIO指令的输出参数Done引脚会变TRUE。
- 起步时原点信号无效则以高速朝正向运行，在正向运行时遇到原点信号由OFF→ON 状态之后减速停止，然后换低速朝负向运行。在低速负向运行遇到原点信号状态由ON→OFF的状态时减速停止，以停止位置作为原点。
- 起步时原点信号有效则以高速朝负向运行。在负向运行时遇到原点信号由ON→OF状态之后减速停止，然后高速回退到原点信号有效的位置之后再减速停止，此后换低速朝负向运行。在低速负向运行遇到原点信号由ON→OF状态时减速停止，以停止位置作为原点。
- 无论遇到负极限还是正极限的ON状态，都是停止回原点流程并报警。



## 4.5 MC\_Home (原点设置)

控制指定轴使用驱动器的原点返回模式、原点信号、极限信号等确定机械原点。所属库：MotionControl

指令	名称	FB/FUN	梯形图样式	ST样式
MC_Home	原点设置	FB		<pre>MC_Home_Instance ( Axis := 参数 , Execute := 参数 , Position:= 参数 , BufferMode:= 参数 , Done =&gt; 参数 , Busy =&gt; 参数 , Active=&gt; 参数 , CommandAborted =&gt; 参数 , Error =&gt; 参数 , ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
Position	原点位置	LREAL	负数、正数、0	0	轴找到原点后的设定位置（单位：行程单位）*1
BufferMode	缓存模式	MC_Buffer_Mode	0：保留	0	保留

\*1: 指令单位的详细介绍请参考“[运动控制指令参数单位](#)”章节。

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成后变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Active	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	指令执行完成后	<ul style="list-style-type: none"> <li>◆ Done为TRUE，Execute由TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时，Done变为TRUE一个周期后，变为FALSE</li> </ul>
Busy	Execute 由FALSE变为TRUE时	<ul style="list-style-type: none"> <li>◆ Done变为TRUE时</li> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>
Active	指令开始控制轴时	<ul style="list-style-type: none"> <li>◆ Done变为TRUE时</li> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>

CommandAborted	指令的执行被中断时	<ul style="list-style-type: none"> <li>Execute由TRUE变为FALSE时</li> <li>指令执行且Execute为FALSE时, 该指令被其它指令中断时, CommandAborted变为TRUE一个周期后, 变为FALSE</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>Execute由TRUE变为FALSE时</li> </ul>

## ■ 功能说明

### ● 基本功能说明

- 该指令控制轴为伺服轴时, 用于控制轴使用驱动器的原点返回模式、原点信号、极限信号等确定机械原点。将原点开关、正向极限或反向极限开关接到伺服驱动器的外部输入点上以实现原点回归功能。
- 虚拟伺服轴和编码器轴原点返回模式只能设置为35。
- 在软件中运动控制→轴设置→原点返回设置部分设置原点返回模式、原点返回开始速度、原点返回接近速度。设置方法请参考示例程序。原点返回模式说明请参考指定轴对应驱动器的说明。禾川伺服驱动器原点返回模式说明请参考该手册最后章节“[原点返回模式](#)”。
- 该指令只有轴状态处于StandStill状态时才可以执行, 其它状态下执行时, 该指令报错。

### ● 数据对象的映射 (PDO 映射)

M500S 系列控制器不需要用户映射数据对象。M500 系列控制器需要用户映射数据对象, 映射对象如下表所示。

接收PDO (主站→从站) (16进制)	映射数据含义	发送PDO (从站→主站)	映射数据含义
6040_0 (索引_子索引)	控制字	6041_0 (索引_子索引)	状态字
6060_0 (索引_子索引)	控制模式	6061_0 (索引_子索引)	反馈模式

### ● 指令完成时机

当原点设置完毕后, 该指令完成, Done 由 FALSE 变为 TRUE。

### ● 重启该指令

当指令执行完成后, Execute 再次由 FALSE 变为 TRUE 时, 该指令可以重新执行; 当指令正在执行中, Execute 再次由 FALSE 变为 TRUE 时, 对指令的执行不会产生影响, 指令仍按照未执行完成的输入变量执行指令。

### ● 其它指令执行时启动该指令

其它运动指令执行时, 启动该指令, 该指令会报错, 只有轴状态处于 StandStill 状态时才可以执行。

### ● 该指令执行时启动其它指令

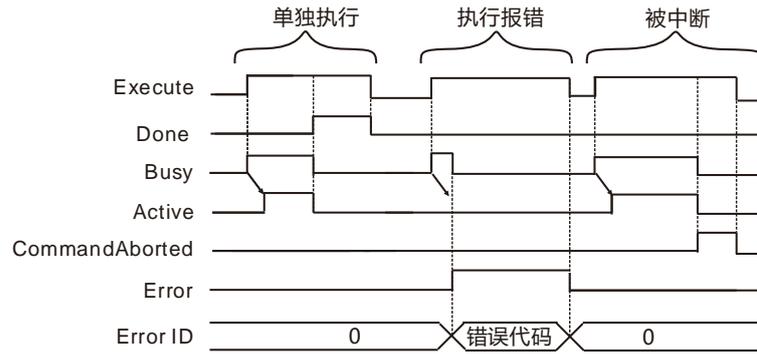
该指令执行时启动其它运动指令, 其它运动指令和该指令如何中继由其它运动指令的 BufferMode (缓存模式) 引脚参数值决定, BufferMode (缓存模式) 可以设定的参数值如下表所示。其它运动指令和该指令缓存时, 其它指令执行的时机为该指令 Done(完成) 变为 TRUE 时。如果其它运动指令无 BufferMode 参数一般为中断该指令。

缓存模式	含义
mcAborting (中断)	中断当前正在控制轴的指令, 切换为缓存指令控制
mcBuffered (等待)	等待当前指令执行完成时 (如目标位置或者目标速度到达), 切换为缓存指令控制轴

### ● 异常排除

该指令执行时, 如果指令输入变量值不在允许的范围或者不满足指令的执行条件, 该指令 Error 变为 TRUE, ErrorID 中有对应的错误码, 可以通过该错误码判断错误原因。该指令执行时, 如遇到异常 (如轴报警等), 该指令也会报错, 轴会立即停止。

## ■ 输出变量变化时序说明



### ● 单独执行

Execute (启动) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE, 下一个周期 Active (控制中) 变为 TRUE。指令完成时, Done (完成) 变为 TRUE, Busy (执行中) 和 Active (控制中) 同时变为 FALSE。当 Execute (启动) 变为 FALSE 时, Done (完成) 同时变为 FALSE。

### ● 执行报错

该指令输入变量值不在允许范围内时, 该指令 Execute (启动) 由 FALSE 变为 TRUE 时, 同时 Busy (执行中) 变为 TRUE, 下一个周期 Error (错误) 变为 TRUE, 同时 Busy (执行中) 变为 FALSE, ErrorID (错误码) 输出对应的错误码, 可通过 ErrorID (错误码) 的值, 查找发生问题原因。该指令 Execute (启动) 由 TRUE 变为 FALSE 时, 同时 Error (错误) 变为 FALSE, ErrorID (错误码) 的值变为 0。

### ● 被中断

该指令执行后被其它指令中断时, 该指令 CommandAborted (中断) 变为 TRUE, Busy (执行中) 和 Active (控制中) 同时变为 FALSE; 当 Execute (执行中) 变为 FALSE 时, CommandAborted (中断) 同时变为 FALSE。

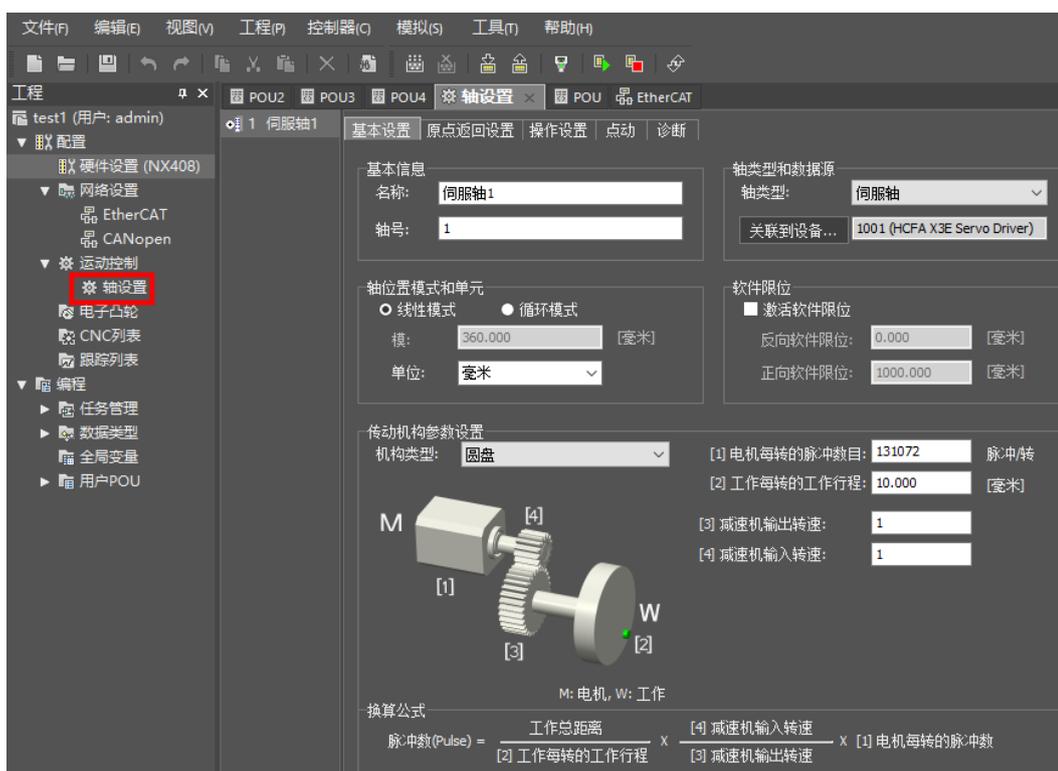
## ■ 示例程序

### ● 实现功能

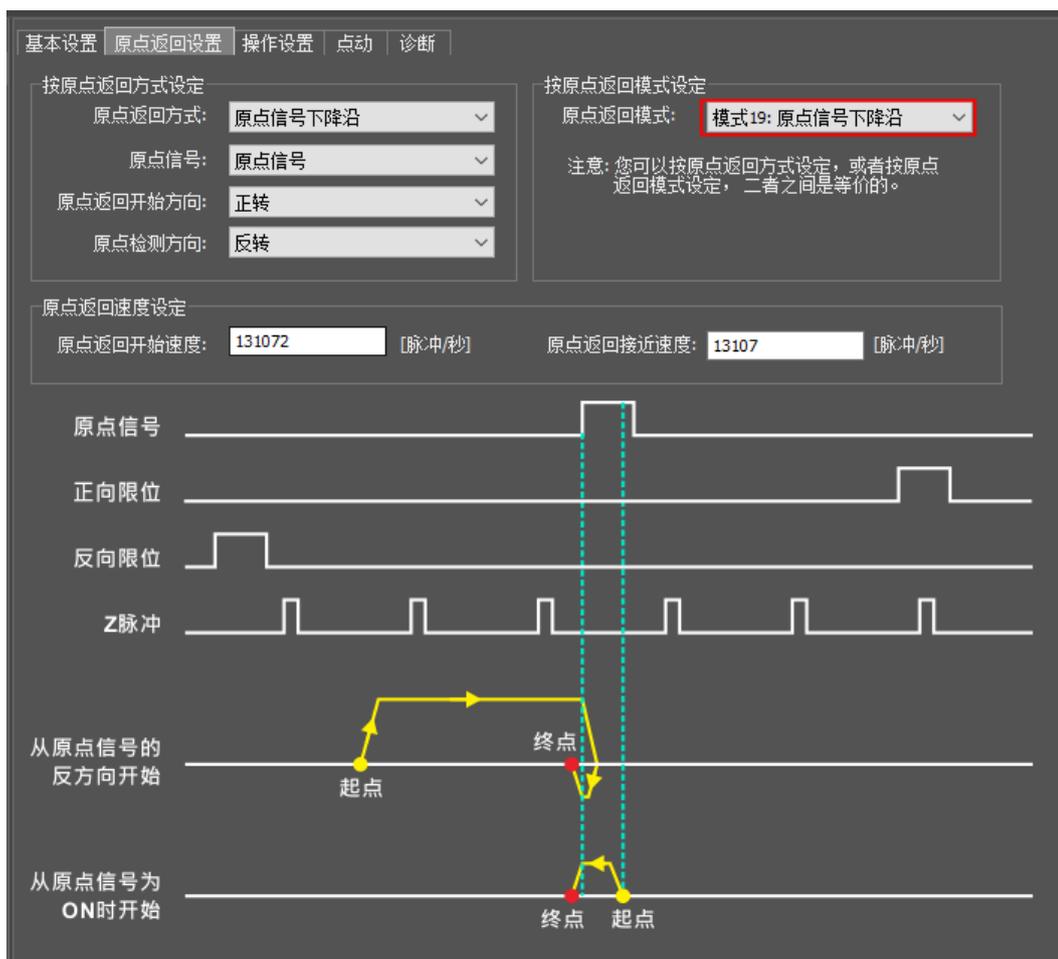
将外部传感器信号触发位置设定为用户指定的原点位置。原点返回模式和原点回归速度等通过软件设置, 外部传感器信号接伺服驱动器的输入点, 原点设置指令执行后, 按照软件轴参数“原点返回设置”界面中设定原点返回模式执行回原点动作。

### ● 轴参数设置

轴 1 轴参数设置及原点返回模式设置如下图所示。



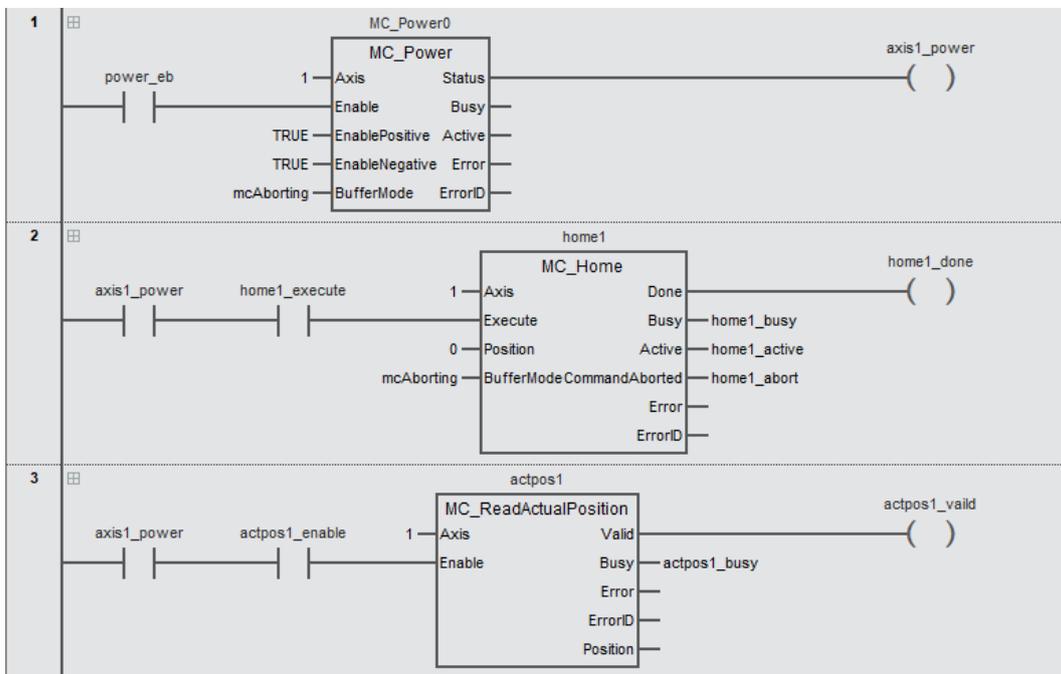
原点返回模式设置如下图红色方框处。



● 变量表

类别	名称	分配到	数据类型	初始值	注释
VAR	power_eb		BOOL		
VAR	power1		MC_Power		
VAR	axis1_power		BOOL		
VAR	home1_execute		BOOL		
VAR	home1		MC_Home		
VAR	home1_done		BOOL		
VAR	home1_busy		BOOL		
VAR	home1_active		BOOL		
VAR	actpos1_enable		BOOL		
VAR	actpos1		MC_ReadActualPosition		
VAR	actpos1_vaid		BOOL		
VAR	actpos1_busy		BOOL		

● 梯形图(LD)



● 结构化文本(ST)

```

power1 (
  Axis:=1 ,
  Enable:=power_eb ,
  EnablePositive:=TRUE ,
  EnableNegative:=TRUE ,
  BufferMode:=mcAborting ,
  Status=>axis1_power
);

home1(
  Axis:=1 ,
  Execute:= axis1_power AND home1_execute ,
  Position:=0 ,
  BufferMode:=mcAborting ,
  Done=>home1_done ,
  Busy=>home1_busy ,
  Active=>home1_active ,

```

```

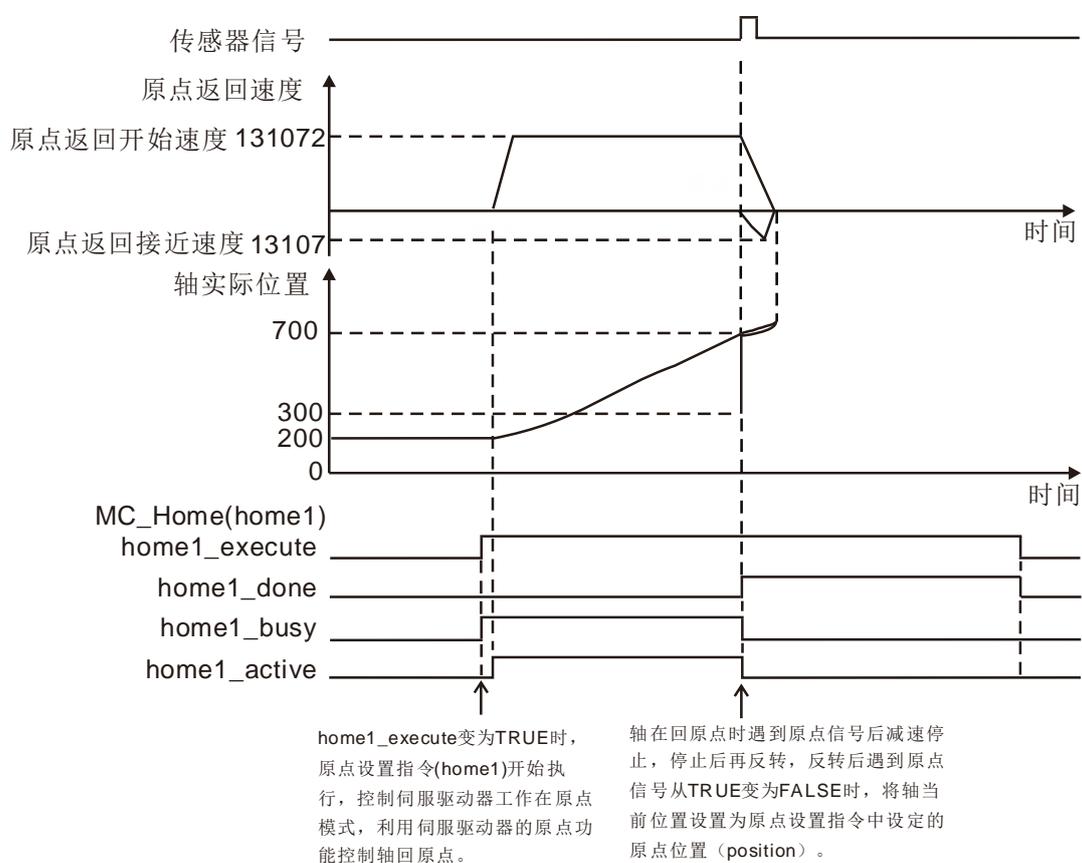
CommandAborted=>home1_abort
);

actpos1(
  Axis:=1,
  Enable:=axis1_power AND actpos1_enable,
  Valid=>actpos1_vaild,
  Busy=>actpos1_busy
);

```

### ● 程序说明

- 轴使能后，home1\_execute变为TRUE时，开始执行原点设置指令（home1）。该指令执行后，伺服驱动器工作于原点模式，利用伺服驱动器的原点功能控制轴回原点。
- 轴回原点过程中遇到原点信号后（外部传感器信号接伺服驱动器的输入点），轴减速停止，停止后再反转，反转时遇到原点信号从TRUE变为FALSE时，将轴当前位置设置为原点设置指令（home1）设定的原点位置（position）。



## 4.6 MC\_HomeWithParm (参数指定原点设置)

控制指定轴使用驱动器的原点返回模式、原点信号、极限信号等确定机械原点，回原点模式、速度加速度等可以通过该指令设置。所属库：MotionControl\_Part2

适用机种： M500S系列、M500系列

指令	名称	FB/FUN	梯形图样式	ST样式
MC_HomeWithParm	参数指定原点设置	FB		<pre>MC_HomeWithParm_Instance ( Axis := 参数 , Execute := 参数 , HomeMode := 参数 , Position := 参数 , SpeedToSwitch := 参数 , SpeedToZero := 参数 , HomeAcc := 参数 , BufferMode := 参数 , Done =&gt; 参数 , Busy =&gt; 参数 , Active =&gt; 参数 , CommandAborted =&gt; 参数 , Error =&gt; 参数 , ErrorID =&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
HomeMode	原点返回模式	USINT	1~35	不可缺省	指定轴寻找原点的方式
Position	原点位置	LREAL	负数、正数、0	0	轴找到原点后的设定位置（单位：行程单位）*1
SpeedToSwitch	原点返回开始速度	UDINT	正数、0		设定寻找原点时的速度
SpeedToZero	原点返回接近速度	UDINT	正数、0		遇到原点信号后减速停止时的速度
HomeAcc	寻找原点时的加速度	UDINT	正数、0		从静止到到达原点返回开始速度时的加速度。
BufferMode	缓存模式	MC_Buffer_Mode	0: 保留	0	保留

\*1: 指令单位的详细介绍请参考“[运动控制指令参数单位](#)”章节。

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成后变为TRUE。
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE。
Active	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE。
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE。
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE。
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码。 值的含义请参考“ <a href="#">指令错误代码描述</a> ”。

■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	指令执行完成后。	<ul style="list-style-type: none"> <li>◆ Done为TRUE, Execute由TRUE变为FALSE时。</li> <li>◆ 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE。</li> </ul>
Busy	Execute 由FALSE变为TRUE时。	<ul style="list-style-type: none"> <li>◆ Done变为TRUE时。</li> <li>◆ Error变为TRUE时。</li> <li>◆ CommandAborted变为TRUE时。</li> </ul>
Active	指令开始控制轴时。	<ul style="list-style-type: none"> <li>◆ Done变为TRUE时。</li> <li>◆ Error变为TRUE时。</li> <li>◆ CommandAborted变为TRUE时。</li> </ul>
CommandAborted	指令的执行被中断时。	<ul style="list-style-type: none"> <li>◆ Execute由TRUE变为FALSE时。</li> <li>◆ 指令执行且Execute为FALSE时, 该指令被其它指令中断时, CommandAborted变为TRUE一个周期后, 变为FALSE。</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常。	<ul style="list-style-type: none"> <li>◆ Execute由TRUE变为FALSE</li> </ul>

■ 功能说明

● 基本功能说明

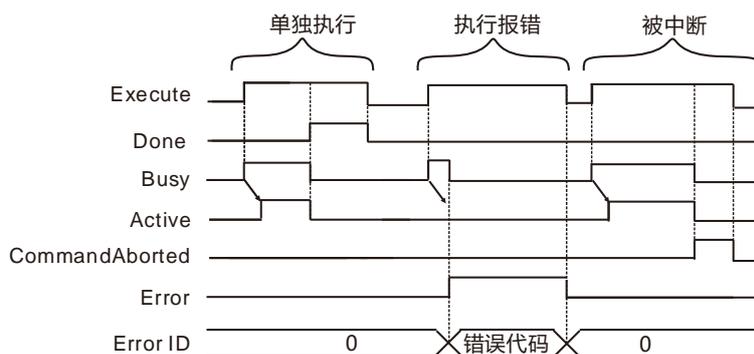
- 该指令用于控制轴使用驱动器的原点返回模式、原点信号、极限信号等确定机械原点,回原点模式、速度等可以通过该指令设置。将原点开关、正向极限或反向极限开关接到伺服驱动器的外部输入点上实现原点回归功能。原点返回模式、原点位置、原点返回开始速度、原点返回接近速度等可以在该指令执行前更改。
- 原点返回模式通过输入变量HomeMode设定, 原点返回开始速度通过输入变量SpeedToSwitch设定, 原点返回接近速度通过输入变量SpeedToZero设定。原点返回模式说明请参考指定轴对应驱动器的说明。禾川伺服驱动器原点返回模式说明请参考该手册最后章节“[原点返回模式](#)”
- 虚拟伺服轴和编码器轴只能设置为原点返回模式35。
- 该指令只有轴状态处于StandStill状态时才可以执行, 其它状态下执行时, 该指令报错。

● 数据对象的映射 (PDO 映射)

M500S 系列控制器不需要用户映射数据对象。M500 系列控制器需要用户映射数据对象, 映射对象如下表所示。

接收PDO (主站=>从站) (16进制)	映射数据含义	发送PDO (从站=>主站)	映射数据含义
6040_0 (索引_子索引)	控制字	6041_0 (索引_子索引)	状态字
6060_0 (索引_子索引)	控制模式	6061_0 (索引_子索引)	反馈模式

■ 输出变量变化时序说明



- 单独执行

Execute (启动) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE, 下一个周期 Active (控制中) 变为 TRUE。指令完成时, Done (完成) 变为 TRUE, Busy (执行中) 和 Active (控制中) 同时变为 FALSE。当 Execute (启动) 变为 FALSE 时, Done (完成) 同时变为 FALSE。

- 执行报错

该指令输入变量值不在允许范围内时, 该指令 Execute (启动) 由 FALSE 变为 TRUE 时, 同时 Busy (执行中) 变为 TRUE, 下一个周期 Error (错误) 变为 TRUE, 同时 Busy (执行中) 变为 FALSE, ErrorID (错误码) 输出对应的错误码, 可通过 ErrorID (错误码) 的值, 查找发生问题原因。该指令 Execute (启动) 由 TRUE 变为 FALSE 时, 同时 Error (错误) 变为 FALSE, ErrorID (错误码) 的值变为 0。

- 被中断

该指令执行后被其它指令中断时, 该指令 CommandAborted (中断) 变为 TRUE, Busy (执行中) 和 Active (控制中) 同时变为 FALSE; 当 Execute (执行中) 变为 FALSE 时, CommandAborted (中断) 同时变为 FALSE。

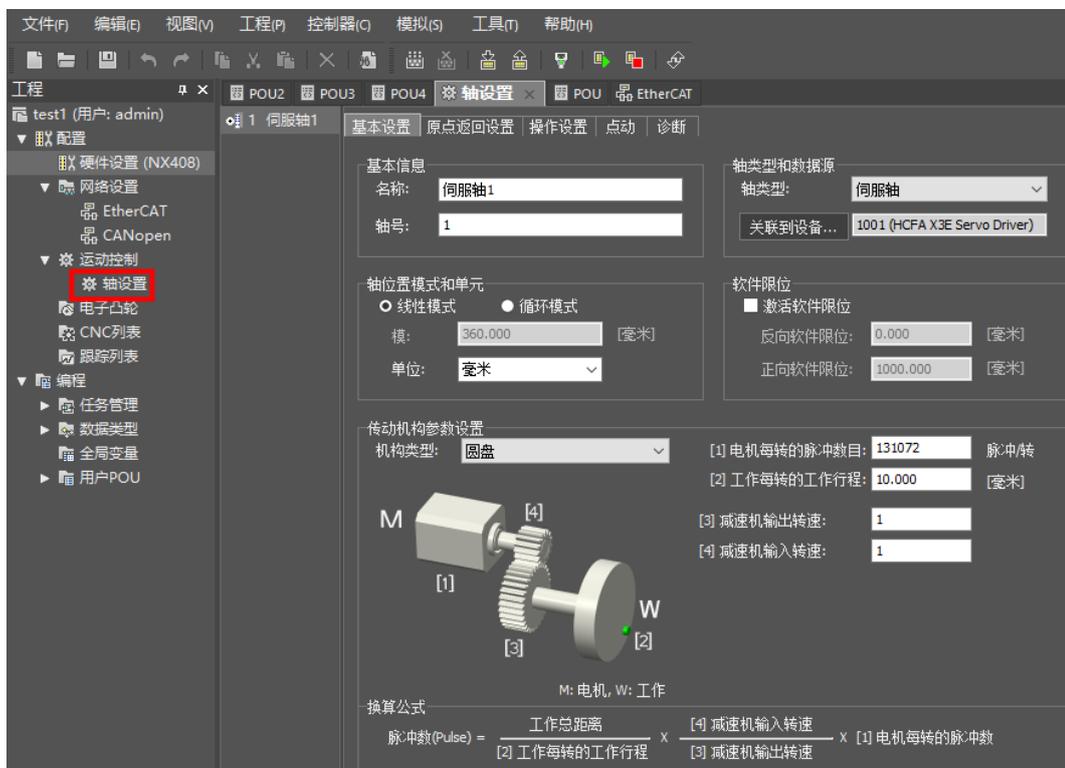
## ■ 示例程序

- 实现功能

将外部传感器信号触发位置设定为用户指定的原点位置。原点返回模式和原点回归速度等通过指令 MC\_HomeWithParm (参数指定原点设置) 设置, 外部传感器信号接伺服驱动器的输入点, 原点设置指令执行后, 按照指令设定原点返回模式执行回原点动作。

- 轴参数设置

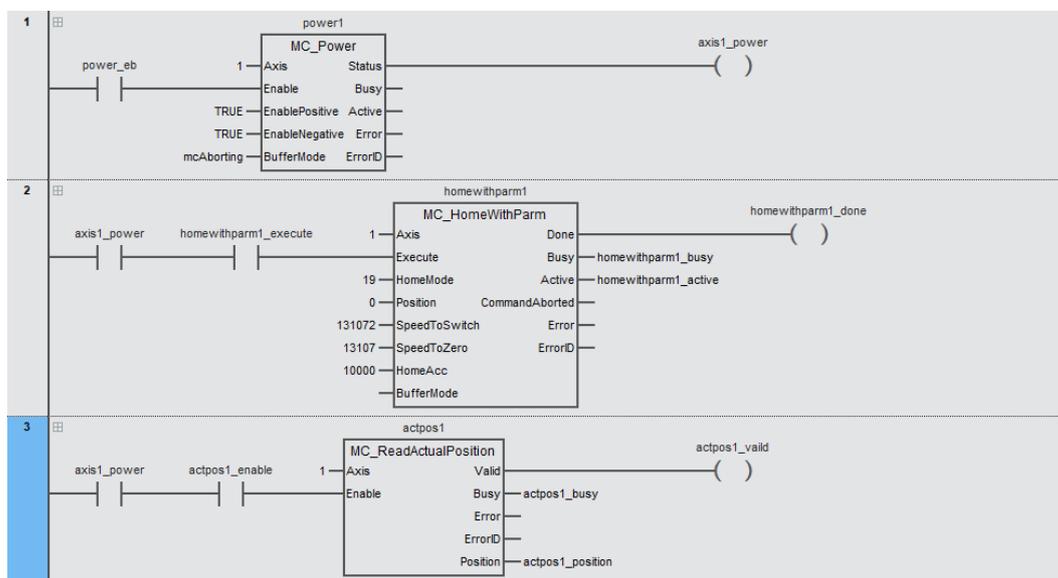
轴 1 轴参数设置及原点返回模式设置如下图所示。



● 变量表

类别	名称	分配到	数据类型	初始值	注释
VAR	power_eb		BOOL		
VAR	power1		MC_Power		
VAR	axis1_power		BOOL		
VAR	homewithparm1_execute		BOOL		
VAR	homewithparm1		MC_HomeWithParm		
VAR	homewithparm1_done		BOOL		
VAR	homewithparm1_busy		BOOL		
VAR	homewithparm1_active		BOOL		
VAR	actpos1_enable		BOOL		
VAR	actpos1		MC_ReadActualPosition		
VAR	actpos1_vaild		BOOL		
VAR	actpos1_busy		BOOL		
VAR	actpos1_position		LREAL		

● 梯形图(LD)



● 结构化文本(ST)

```
MC_Power0(
  Axis:=1 ,
  Enable:=power_eb ,
  EnablePositive:=TRUE ,
  EnableNegative:=TRUE ,
  BufferMode:=mcAborting ,
  Status=>axis1_power
);
```

```
homewithparm1 (
  Axis:=1 ,
  Execute:= axis1_power AND homewithparm1_execute ,
  HomeMode:=19,
  Position:=0 ,
  SpeedToSwitch:=131072,
  SpeedToZero:=13107,
  HomeAcc:=1000,
```

```

Done=> homewithparm1_done ,
Busy=> homewithparm1_busy ,
Active=> homewithparm1_active
);

```

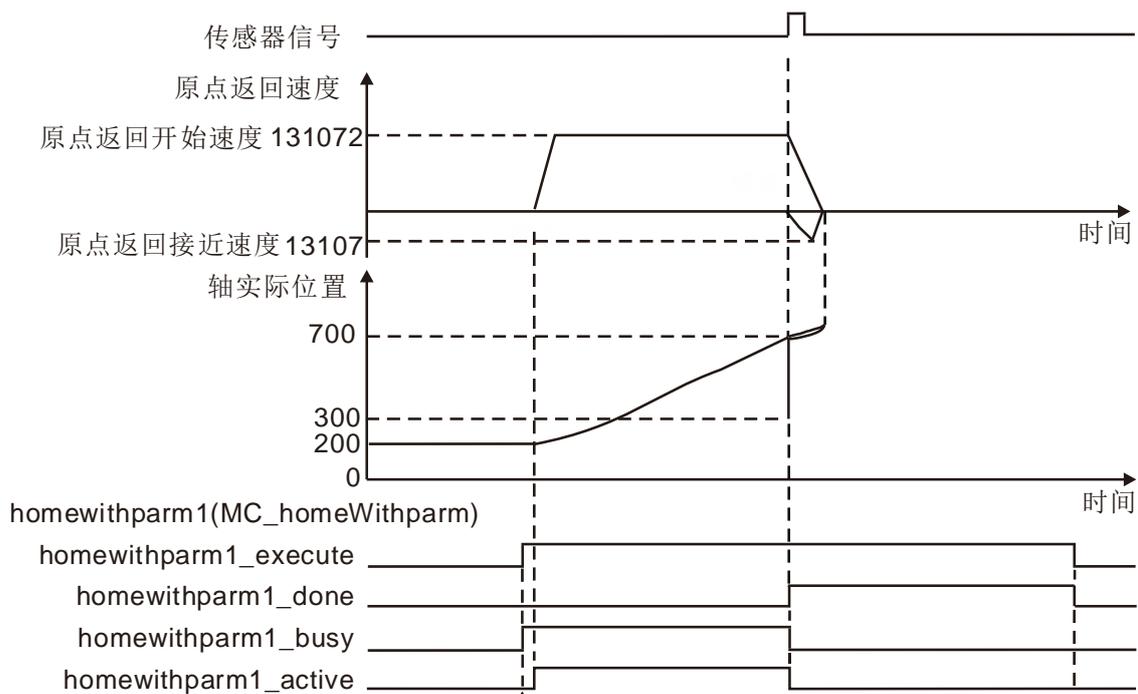
```

actpos1(
  Axis:=1 ,
  Enable:=axis1_power AND actpos1_enable ,
  Valid=>actpos1_vaild ,
  Busy=>actpos1_busy ,
  Position=>actpos1_position
);

```

#### ● 程序说明

- 轴使能后，HomeWithParm1\_execute变为TRUE时，开始执行参数指定原点设置指令（HomeWithParm1）。该指令执行后，伺服驱动器工作于原点模式，利用伺服驱动器的原点功能控制轴回原点。
- 轴回原点过程中遇到原点信号后（外部传感器信号接伺服驱动器的输入点），轴减速停止，停止后再反转，反转时遇到原点信号从TRUE变为FALSE时，将轴当前位置设置为参数指定原点设置指令（HomeWithParm1）设定的原点位置（position）。



homewithparm1\_execute变为TRUE时，原点设置指令(homewithparm1)开始执行，控制伺服驱动器工作在原点模式，利用伺服驱动器的原点功能控制

轴在回原点时遇到原点信号后减速停止，停止后再反转，反转后遇到原点信号从TRUE变为FALSE时，将轴当前位置设置为原点设置指令中设定的原点位置（position）。

## 4.7 MC\_MoveVelocity (速度)

使用驱动器的位置控制模式，控制器根据输入变量，以设定周期传送位置给驱动器，模拟速度控制。所属库：MotionControl

指令	名称	FB/FUN	梯形图样式	ST样式
MC_MoveVelocity	速度	FB		<pre>MC_MoveVelocity_Instance ( Axis := 参数, Execute := 参数, ContinuousUpdate:= 参数, Velocity:= 参数, Acceleration:= 参数, Deceleration:= 参数, Jerk:= 参数, Direction:= 参数, BufferMode:= 参数, InVelocity=&gt; 参数, Busy =&gt; 参数, Active=&gt; 参数, CommandAborted =&gt; 参数, Error =&gt; 参数, ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
ContinuousUpdate	持续更新	BOOL	TRUE或FALSE	FALSE	保留
Velocity	目标速度	LREAL	0、正数	不可缺省	指定目标速度*1 (单位: 行程单位/秒) *2
Acceleration	加速度	LREAL	正数	不可缺省	指定加速度*1 (单位: 行程单位/秒 <sup>2</sup> ) *2
Deceleration	减速度	LREAL	正数	不可缺省	指定减速度*1 (单位: 行程单位/秒 <sup>2</sup> ) *2
Jerk	跃度	LREAL	正数	不可缺省	指定跃度*1 (单位: 行程单位/秒 <sup>3</sup> ) *2
Direction	方向	MC_Direction	1: mcPositiveDirection 3: mcNegativeDirection 4: mcCurrentDirection	1	设定该指令控制轴运转时轴的运转方向 1: 正方向 3: 反方向 4: 按当前运转方向运转 (轴静止时为 正方向移动)
BufferMode	缓存模式	MC_Buffer_Mode	0: mcAborting 1: mcBuffered 2: mcBlendingLow 3: mcBlendingPrevious 4: mcBlendingNext 5: mcBlendingHigh	0	设定两个指令之间缓存模式。*3 0: 中断 1: 等待 2: 以低速中继 3: 以前一个速度中继 4: 以后一个速度中继 5: 以高速中继

\*1: Velocity、Acceleration、Deceleration、Jerk的关系请参考“[运动控制指令相关参数说明](#)”章节。

\*2: 指令单位的详细介绍请参考“[运动控制指令参数单位](#)”章节。

\*3: BufferMode的详细介绍请参考“[运动控制指令的多重启动时缓存模式说明 \(BufferMode\)](#)”章节。

## ■ 输出变量

名称	名称	数据类型	输出范围	功能
Invelocity	目标速度到达	BOOL	TRUE或FALSE	轴命令速度到达目标速度时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Active	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

## ■ 输出变量刷新时期

名称	变为TRUE的时机	变为FALSE的时机
Invelocity	轴命令速度到达目标速度时	<ul style="list-style-type: none"> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>
Busy	Execute的上升沿	<ul style="list-style-type: none"> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>
Active	指令开始控制轴时	<ul style="list-style-type: none"> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>
CommandAborted	指令的执行被中断时	<ul style="list-style-type: none"> <li>◆ CommandAborted为TRUE，Execute由TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时，该指令被其它指令中断时，CommandAborted变为TRUE一个周期后，变为FALSE</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Execute由TRUE变为FALSE时</li> </ul>

## ■ 功能说明

### ● 基本功能说明

使用驱动器的位置控制模式，控制器根据输入变量，以设定周期传送位置给驱动器，模拟速度控制。该指令在 Execute 由 FALSE 变为 TRUE 时开始执行；指令根据输入变量的设定值，轴的速度会发生改变，将会从当前速度加速或减速到设定的速度。

### ● 方向选择

该指令控制轴运转时轴的运转方向由 Direction（方向选择）指定。Direction（方向选择）的值为 1 时，轴向正方向移动；Direction（方向选择）的值为 3 时，轴向反方向移动；Direction（方向选择）的值为 4 时，轴按当前运行方向继续移动。Direction（方向选择）的值为 4 时，如其它指令控制轴正方向移动时，速度指令中断其它指令后仍正方向移动；如其它指令控制轴反方向移动时，速度指令中断其它指令后仍反方向移动；如轴处于停止状态时，轴向正方向移动。

### ● 重启该指令

当指令正在执行中，Execute 由 FALSE 变为 TRUE 时，该指令可以重新执行；此时能够重新生效的参数包括 Velocity、Acceleration、Deceleration、Jerk、Direction，其它参数不会生效。当速度指令与其它运动指令产生 BufferMode 关系时，改变速度指令的参数，重新触发后速度指令的参数会生效，原来的交接关系仍然保持，交接速度会重新计算。

### ● 其它指令执行时启动该指令

其它运动指令执行时，启动该指令，可切换或缓存到该指令。该指令和其它运动指令如何中继由该指令输入变量 BufferMode（缓存模式）的值决定，而且该指令输入变量 BufferMode（缓存模式）可以设置的值和正在执行的运动指令有关。

### ● 该指令执行时启动其它指令

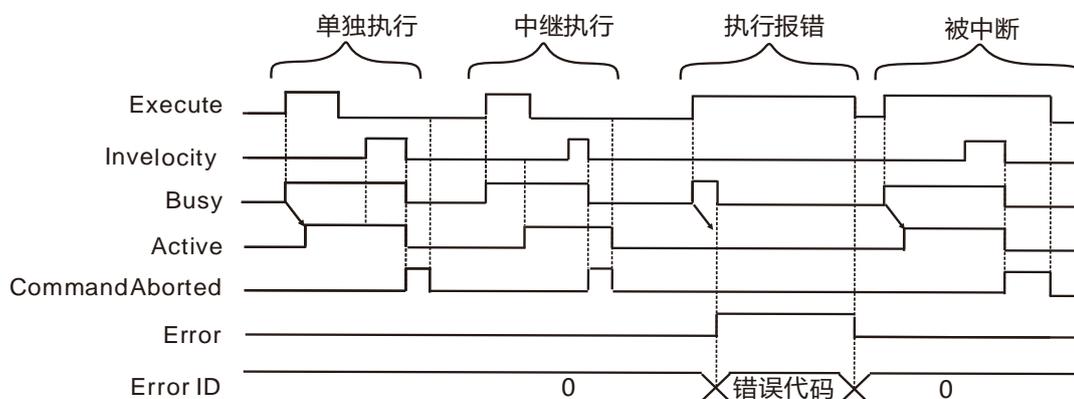
该指令执行时启动其它运动指令，其它运动指令和该指令如何中继由其它指令的 BufferMode（缓存模式）参数值决定，其它运动指令 BufferMode 的值只能选择中断或者等待，如下表所示；如果其它运动指令无 BufferMode 参数，一般为中断该指令。其它运动指令和该指令缓存时，其它指令执行的时机为该指令“lnVelocity（目标速度到达）”变为 TRUE 时。

缓存模式	含义
mcAborting (中断)	中断当前正在控制轴的指令，切换为缓存指令控制
mcBuffered (等待)	等待当前指令执行完成时（如目标位置或者目标速度到达），切换为缓存指令控制轴

### ● 异常排除

该指令执行时，如果输入变量不合法，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。该指令执行时，如遇到异常（如轴报警等），该指令也会报错，轴会立即停止。

### ■ 输出变量变化时序说明



### ● 单独执行

Execute（启动）由 FALSE 变为 TRUE 时，Busy（执行中）同时变为 TRUE，下一个周期 Active（控制中）变为 TRUE。当轴命令速度达到目标速度时，lnvelocity（目标速度到达）变为 TRUE，Busy（执行中）和 Active（控制中）仍为 TRUE。

### ● 中继执行

其它指令控制轴时，执行该指令（BufferMode 的值不为 mcAborting），当 Execute（启动）由 FALSE 变为 TRUE 时，Busy（执行中）同时变为 TRUE，Active（控制中）变为 TRUE 的时机为前一个指令完成时。

### ● 执行报错

该指令输入变量值不在允许范围内时，该指令 Execute（启动）由 FALSE 变为 TRUE 时，同时 Busy（执行中）变为 TRUE，下一个周期 Error（错误）变为 TRUE，同时 Busy（执行中）变为 FALSE，ErrorID（错误码）输出对应的错误码，可通过 ErrorID（错误码）的值，查找发生问题原因。该指令 Execute（启动）由 TRUE 变为 FALSE 时，同时 Error（错误）变为 FALSE，ErrorID（错误码）的值变为 0。

### ● 被中断

该指令执行后被其它指令中断时，该指令 CommandAborted（中断）变为 TRUE，Busy（执行中）和 Active（控制中）同时变为 FALSE；当 Execute（执行中）变为 FALSE 时，CommandAborted（中断）同时变为 FALSE。

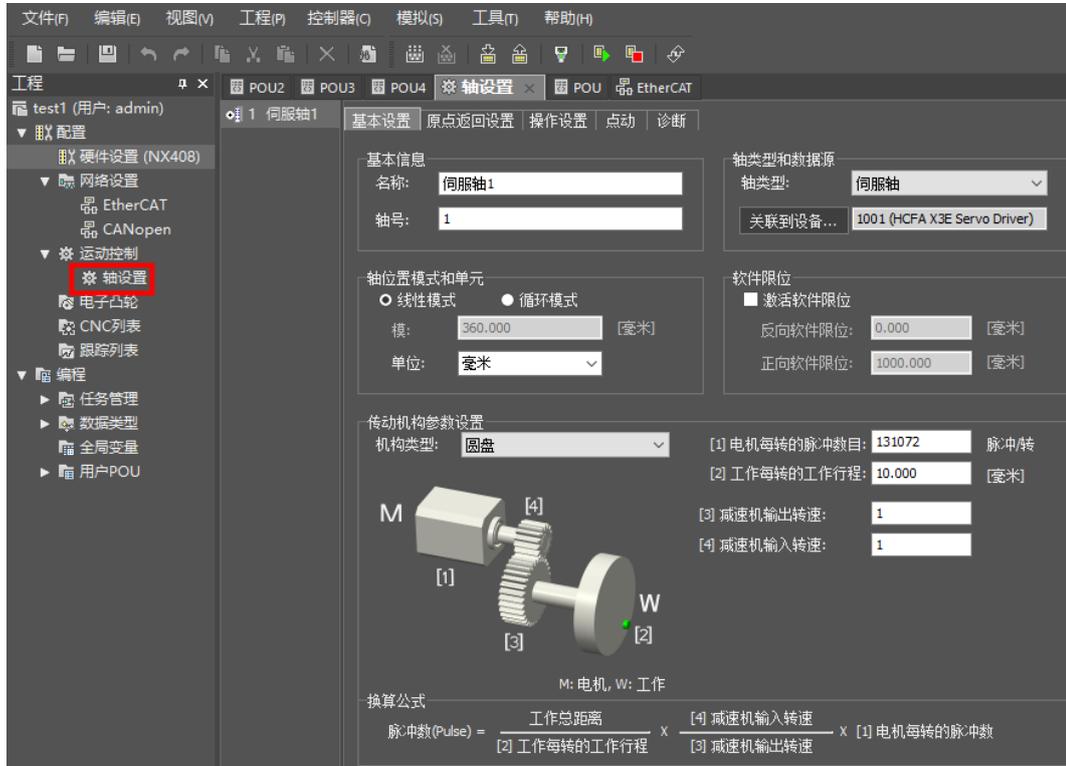
### ■ 示例程序

#### ● 实现功能

实时更改轴的目标速度。速度指令和速度比例设置指令都执行时，通过更改速度比例设置指令的目标速度比例值，实现修改轴的目标速度。该指令实现的功能，也可以通过执行 MC\_MoveContinuousVelocity（速度实时更改）指令实时修改轴的目标速度。

### ● 轴参数设置

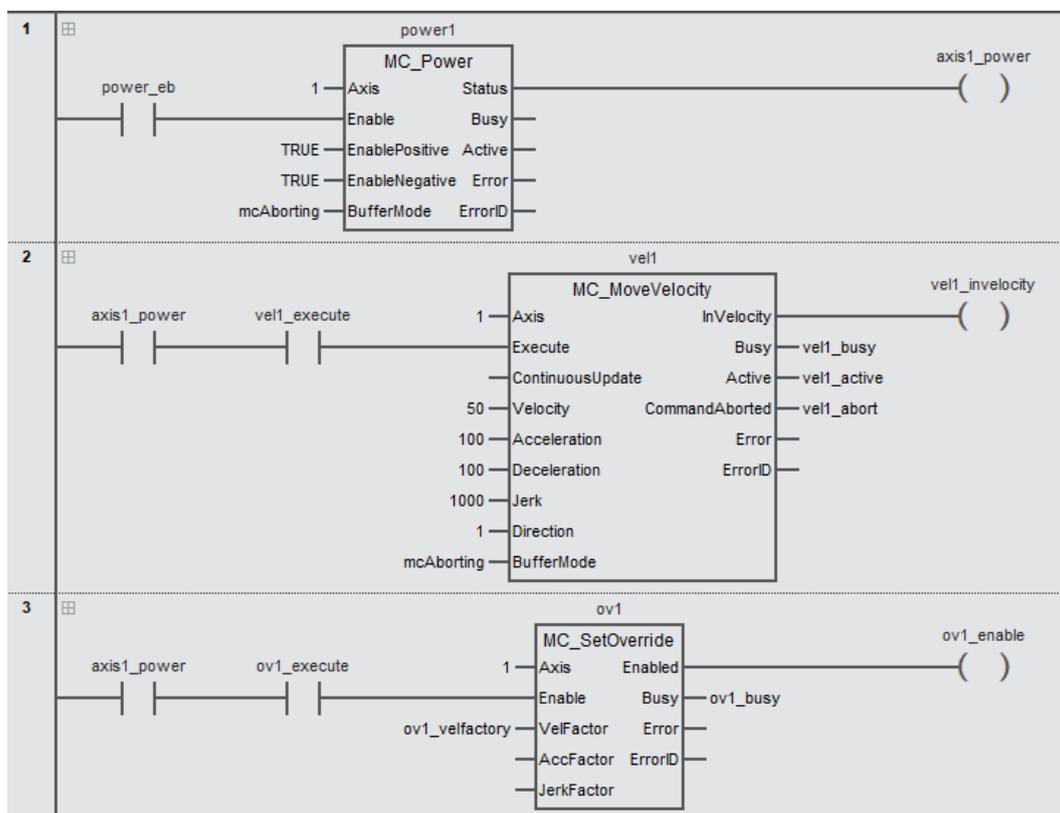
轴 1 轴参数设置如下图所示。



● 变量表

类别	名称	分配到	数据类型	初始值	注释
VAR	power_eb		BOOL		
VAR	power1		MC_Power		
VAR	axis1_power		BOOL		
VAR	vel1_execute		BOOL		
VAR	vel1		MC_MoveVelocity		
VAR	vel1_invelocity		BOOL		
VAR	vel1_busy		BOOL		
VAR	vel1_active		BOOL		
VAR	vel1_abort		BOOL		
VAR	ov1		MC_SetOverride		
VAR	ov1_execute		BOOL		
VAR	ov1_enable		BOOL		
VAR	ov1_velfactory		LREAL	100	
VAR	ov1_busy		BOOL		

● 梯形图(LD)

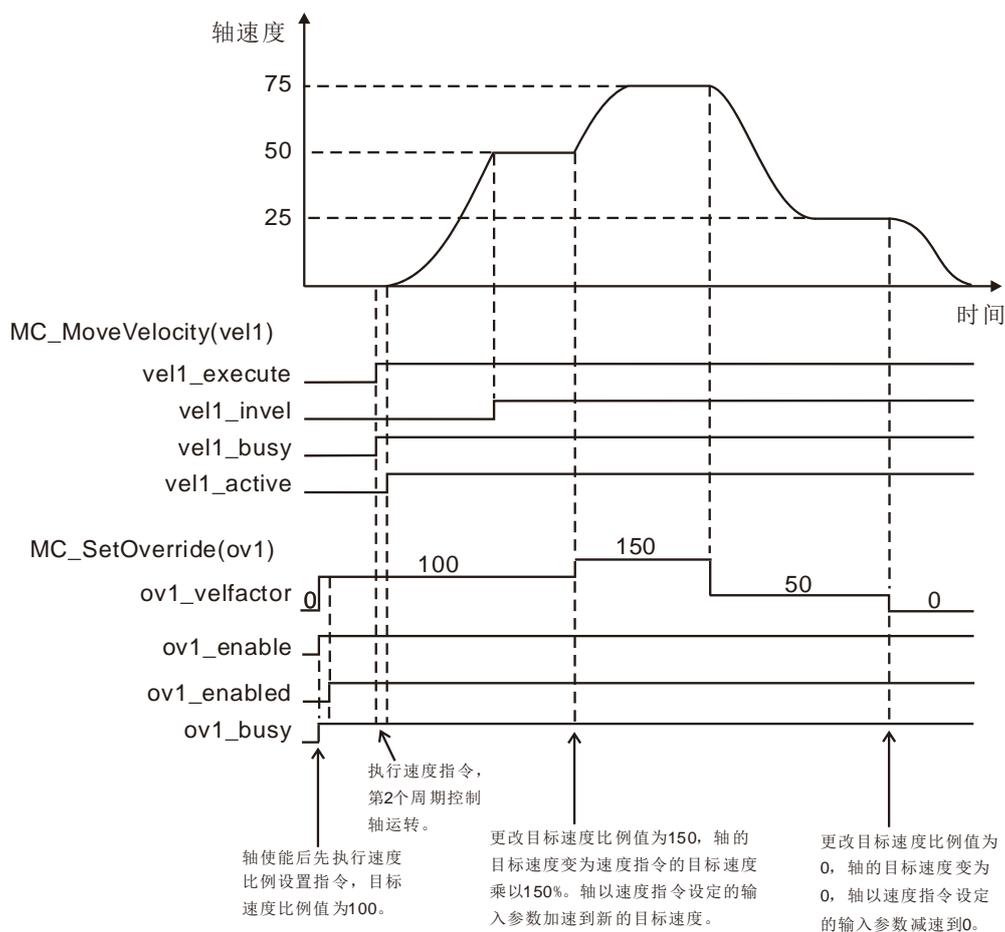


- 结构化文本(ST)

```
power1(  
  Axis:=1 ,  
  Enable:=power_eb ,  
  EnablePositive:=TRUE ,  
  EnableNegative:=TRUE ,  
  BufferMode:=mcAborting ,  
  Status=>axis1_power  
);  
  
vel1(  
  Axis:=1 ,  
  Execute:= axis1_power AND vel1_execute ,  
  Velocity:=50 ,  
  Acceleration:=100 ,  
  Deceleration:=100 ,  
  Jerk:=1000 ,  
  Direction:=1 ,  
  BufferMode:=mcAborting ,  
  InVelocity=> vel1_invelocity ,  
  Busy=> vel1_busy ,  
  Active=> vel1_active ,  
  CommandAborted=> vel1_abort  
);  
  
ov1(  
  Axis:=1 ,  
  Enable:=axis1_power AND vel1_execute ,  
  VelFactor:=ov1_velfactory ,  
  Enabled=>ov1_enable ,  
  Busy=>ov1_busy  
);
```

- 程序说明

- 轴使能后，ov1\_enable变为TRUE，先执行速度比例设置指令（ov1），目标速度比例值为100（100%）。vel1\_execute变为TRUE时，开始执行速度指令（vel1），速度指令执行的第2个周期开始控制轴。
- 如果轴运转过程中想更改轴的目标速度，通过更改速度比例设置指令的目标速度比例值即可。如下图所示，目标速度比例值由100变为150时，轴的目标速度变为75，计算方法：速度指令的目标速度(50)\*150%=75。目标速度比例值变化后，轴以速度指令设定的输入参数进行加速或者减速。如目标速度比例值变为0时，轴目标速度变为0，轴以速度指令设定的输入参数减速到0。



## 4.8 MC\_MoveContinuousVelocity (速度实时更改)

该指令用于控制指定轴按照设定设定加减速以及加速度变化率，加速或者减速到设定的速度进行匀速运动。所属库：MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_MoveContinuousVelocity	速度实时更改	FB		<pre> MC_MoveContinuousVelocity _Instance ( Axis := 参数 , Enable := 参数 , ContinuousUpdate:= 参数 , Velocity:= 参数 , Acceleration:= 参数 , Deceleration:= 参数 , Jerk:= 参数 , Direction:= 参数 , BufferMode:= 参数 , Invelocity =&gt; 参数 , Busy=&gt; 参数 , Active=&gt; 参数 , CommandAborted=&gt; 参数 , Error=&gt; 参数 , ErrorID=&gt; 参数 );                     </pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Enable	有效	BOOL	TRUE或FALSE	FALSE	Enable 为TRUE时执行该指令
ContinuousUpdate	持续更新	BOOL	TRUE或FALSE	FALSE	保留
Velocity	目标速度	LREAL	正数	不可缺省	指定目标速度*1 (单位: 行程单位/秒) *2
Acceleration	加速度	LREAL	正数	不可缺省	指定加速度*1 (单位: 行程单位/秒²) *2
Deceleration	减速度	LREAL	正数	不可缺省	指定减速度*1 (单位: 行程单位/秒²) *2
Jerk	跃度	LREAL	正数	不可缺省	指定跃度*1 (单位: 行程单位/秒³) *2
Direction	方向	MC_Direction	1: mcPositiveDirection 3: mcNegativeDirection 4: mcCurrentDirection	1	设定该指令控制轴运转时轴的运转方向 1: 正方向 3: 反方向 4: 按当前运转方向运转 (轴静止时为 正方向移动)
BufferMode	缓存模式	MC_Buffer_Mode	0: mcAborting 1: mcBuffered 2: mcBlendingLow 3: mcBlendingPrevious 4: mcBlendingNext 5: mcBlendingHigh	0	设定两个指令之间缓存模式。*3 0: 中断 1: 等待 2: 以低速中继 3: 以前一个速度中继 4: 以后一个速度中继 5: 以高速中继

\*1: Velocity、Acceleration、Deceleration、Jerk的关系请参考“[运动控制指令相关参数说明](#)”章节。

\*2: 指令单位的详细介绍请参考“[运动控制指令参数单位](#)”章节。

\*3: BufferMode的详细介绍请参考“[运动控制指令的多重启动时缓存模式说明 \(BufferMode\)](#)”章节。

## ■ 输出变量

名称	名称	数据类型	输出范围	功能
Invelocity	目标速度到达	BOOL	TRUE或FALSE	轴命令速度到达目标速度时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Active	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

## ■ 输出变量刷新时期

名称	变为TRUE的时机	变为FALSE的时机
Invelocity	轴命令速度到达目标速度时	<ul style="list-style-type: none"> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>
Busy	Execute的上升沿	<ul style="list-style-type: none"> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>
Active	指令开始控制轴时	<ul style="list-style-type: none"> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>
CommandAborted	指令的执行被中断时	<ul style="list-style-type: none"> <li>◆ CommandAborted为TRUE，Execute由TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时，该指令被其它指令中断时，CommandAborted变为TRUE一个周期后，变为FALSE</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Execute由TRUE变为FALSE时</li> </ul>

## ■ 功能说明

### ● 基本功能说明

使用驱动器的位置控制模式，控制器根据输入变量，以设定周期传送位置给驱动器，模拟速度控制。该指令在 Enable 为 TRUE 时执行该指令，指令根据输入变量的设定值，控制轴从当前速度加速或减速到设定的目标速度进行匀速运动。当 Enable 为 TRUE 时，更改该指令输入变量 Velocity（目标速度）、Acceleration（加速度）、Deceleration（减速度）、Jerk（跃度）跃度或者 Direction（方向）的值，这些参数值会立即生效，无需重新触发。

### ● 方向选择

该指令控制轴运转时轴的运转方向由 Direction（方向选择）指定。Direction（方向选择）的值为 1 时，轴向正方向移动；Direction（方向选择）的值为 3 时，轴向反方向移动；Direction（方向选择）的值为 4 时，轴按当前运行方向继续移动。Direction（方向选择）的值为 4 时，如其它指令控制轴正方向运动时，速度指令中断其它指令后仍正方向移动；如其它指令控制轴反方向移动时，速度指令中断其它指令后仍反方向移动；如轴处于停止状态时，轴向正方向移动。

### ● 重启该指令

当 Enable 为 TRUE 时，执行该指令；当 Enable 为 FALSE 时，按照 Enable 从 TRUE 变为 FALSE 前的输入变量执行该指令。

### ● 其它指令执行时启动该指令

其它运动指令执行时，启动该指令，可切换或缓存到该指令。该指令和其它运动指令如何中继由该指令输入变量 BufferMode（缓存模式）的值决定，而且该指令输入变量 BufferMode（缓存模式）可以设置的值和正在执行的运动指令有关。

- **该指令执行时启动其它指令**

该指令执行时启动其它运动指令，其它运动指令和该指令如何中继由其它指令的 BufferMode（缓存模式）参数值决定，其它运动指令 BufferMode 的值只能选择中断或者等待，如下表所示；如果其它运动指令无 BufferMode 参数，一般为中断该指令。其它运动指令和该指令缓存时，其它指令执行的时机为该指令“lnVelocity（目标速度达到）”变为 TRUE 时。

缓存模式	含义
mcAborting (中断)	中断当前正在控制轴的指令，切换为缓存指令控制
mcBuffered (等待)	等待当前指令执行完成时（如目标位置或者目标速度到达），切换为缓存指令控制轴

- **异常排除**

该指令执行时，如果输入变量不合法，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。该指令执行时，如遇到异常（如轴报警等），该指令也会报错，轴会立即停止。

## 4.9 MC\_Halt (停止)

控制指定轴从当前速度进行减速停止。所属库：MotionControl

指令	名称	FB/FUN	梯形图样式	ST样式
MC_Halt	停止	FB		<pre>MC_Halt_Instance ( Axis := 参数 , Execute := 参数 , Deceleration:= 参数 , Jerk:= 参数 , BufferMode:= 参数 , Done=&gt; 参数 , Busy=&gt; 参数 , Active=&gt; 参数 , CommandAborted =&gt; 参数 , Error =&gt; 参数 , ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
Deceleration	减速度	LREAL	正数	不可缺省	指定减速度* <sup>1</sup> (单位: 行程单位/秒 <sup>2</sup> ) * <sup>2</sup>
Jerk	跃度	LREAL	正数	不可缺省	指定跃度* <sup>1</sup> (单位: 行程单位/秒 <sup>3</sup> ) * <sup>2</sup>
BufferMode	缓存模式	MC_Buffer_Mode	0: mcAborting 1: mcBuffered	0	设定两个指令之间缓存模式。* <sup>3</sup> 0: 中断 1: 等待

\*1: Deceleration、Jerk的关系请参考“[运动控制指令相关参数说明](#)”章节。

\*2: 指令单位的详细介绍请参考“[运动控制指令参数单位](#)”章节。

\*3: BufferMode的详细介绍请参考“[运动控制指令的多重启动时缓存模式说明 \(BufferMode\)](#)”章节。

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Active	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时, 输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	减速完成, 指定轴速度变为0时	<ul style="list-style-type: none"> <li>◆ Done为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE</li> </ul>

Busy	Execute的上升沿	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
Active	指令开始控制轴时	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
CommandAborted	指令的执行被其它指令中断时	<ul style="list-style-type: none"> <li>◆ CommandAborted 为TRUE , Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令被其它指令中断时, CommandAborted变为TRUE, 一个周期后, 变为FALSE</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令执行过程中遇到异常时, Error变为TRUE, 一个周期后, 变为FALSE</li> </ul>

## ■ 功能说明

### ● 基本功能说明

- 控制指定轴从当前速度进行减速停止，减速度及跃度由指令输入变量设定。
- MC\_Halt指令开始执行，状态机进入DiscreteMotion，当轴速度降为0时，Done变为TRUE，同时状态机变为Standstill。
- 和MC\_Stop指令相比较，MC\_Halt指令不会锁定轴，控制器可以执行其它的运动指令。在MC\_Halt指令执行期间，轴在进行减速时，可以执行其它的运动指令来中断MC\_Halt指令；在MC\_Halt指令执行完毕，轴已经停下来后，控制器可以执行其它的运动指令来重新启动轴。

### ● 重启该指令

当指令执行完成后，Execute 再次由 FALSE 变为 TRUE 时，该指令可以重新执行；当指令正在执行中，Execute 再次由 FALSE 变为 TRUE 时，对指令的执行不会产生影响，指令仍继续按照之前的方式执行。

### ● 其它指令执行时启动该指令

其它运动指令执行时，启动该指令，该指令和其它运动指令如何中继由该指令输入变量 BufferMode（缓存模式）的值决定。该指令输入变量 BufferMode（缓存模式）可以设置的值只可以设置为 mcAborting（中断）或者 mcBuffered（等待）。

缓存模式	含义
mcAborting (中断)	中断当前正在控制轴的指令，切换为缓存指令控制
mcBuffered (等待)	等待当前指令执行完成时（如目标位置或者目标速度到达），切换为缓存指令控制轴

### ● 该指令执行时启动其它指令

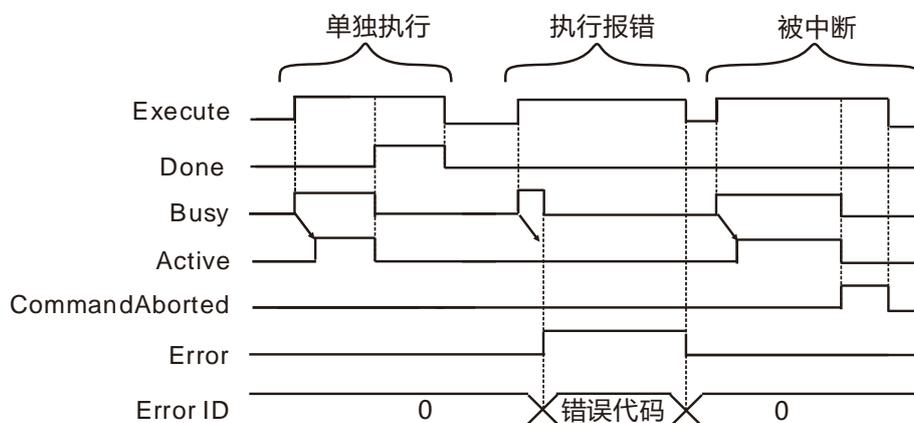
该指令执行时启动其它运动指令，其它运动指令和该指令如何中继由其它运动指令的 BufferMode（缓存模式）参数值决定，BufferMode（缓存模式）可以设定的参数值如下表所示。其它运动指令和该指令缓存时，其它指令执行的时机为该指令“Done（完成）”变为 TRUE 时。如果其它运动指令无 BufferMode 参数，一般为中断该指令。

缓存模式	含义
mcAborting (中断)	中断当前正在控制轴的指令，切换为缓存指令控制
mcBuffered (等待)	等待当前指令执行完成时（如目标位置或者目标速度到达），切换为缓存指令控制轴

- 异常排除

该指令执行时，如果指令输入变量值不在允许的范围或者不满足指令的执行条件，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。该指令执行时，如遇到异常（如轴报警等），该指令也会报错，轴会立即停止。

- 输出变量变化时序说明



- 单独执行

Execute（启动）由 FALSE 变为 TRUE 时，Busy（执行中）同时变为 TRUE，下一个周期 Active（控制中）变为 TRUE。该指令控制轴停止后，Done（完成）变为 TRUE，Busy（执行中）和 Active（控制中）同时变为 FALSE。当 Execute（启动）变为 FALSE 时，Done（完成）同时变为 FALSE。

- 执行报错

该指令输入变量值不在允许范围内时，该指令 Execute（启动）由 FALSE 变为 TRUE 时，同时 Busy（执行中）变为 TRUE，下一个周期 Error（错误）变为 TRUE，同时 Busy（执行中）变为 FALSE，ErrorID（错误码）输出对应的错误码，可通过 ErrorID（错误码）的值，查找发生问题原因。该指令 Execute（启动）由 TRUE 变为 FALSE 时，同时 Error（错误）变为 FALSE，ErrorID（错误码）的值变为 0。

- 被中断

该指令执行后被其它指令中断时，该指令 CommandAborted（中断）变为 TRUE，Busy（执行中）和 Active（控制中）同时变为 FALSE；当 Execute（执行中）变为 FALSE 时，CommandAborted（中断）同时变为 FALSE。

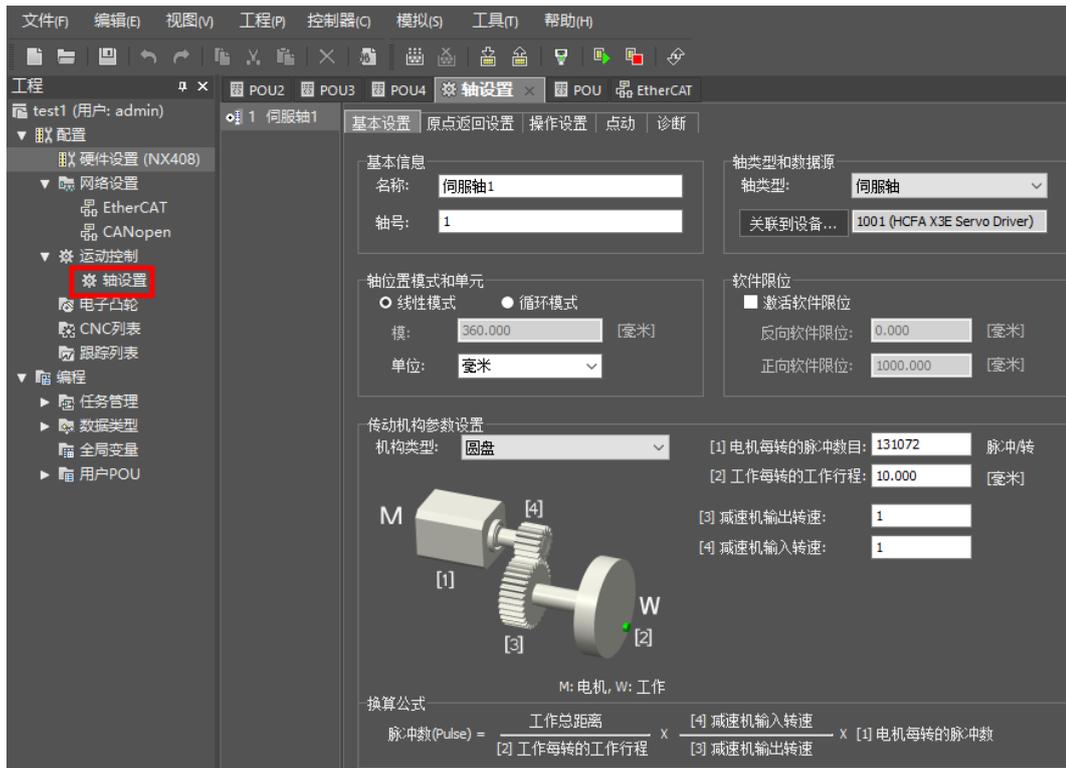
- 示例程序

- 实现功能

速度指令控制轴运转时，执行 MC\_Halt 指令控制轴按照该指令设定的输入参数减速停止。

- 轴参数设置

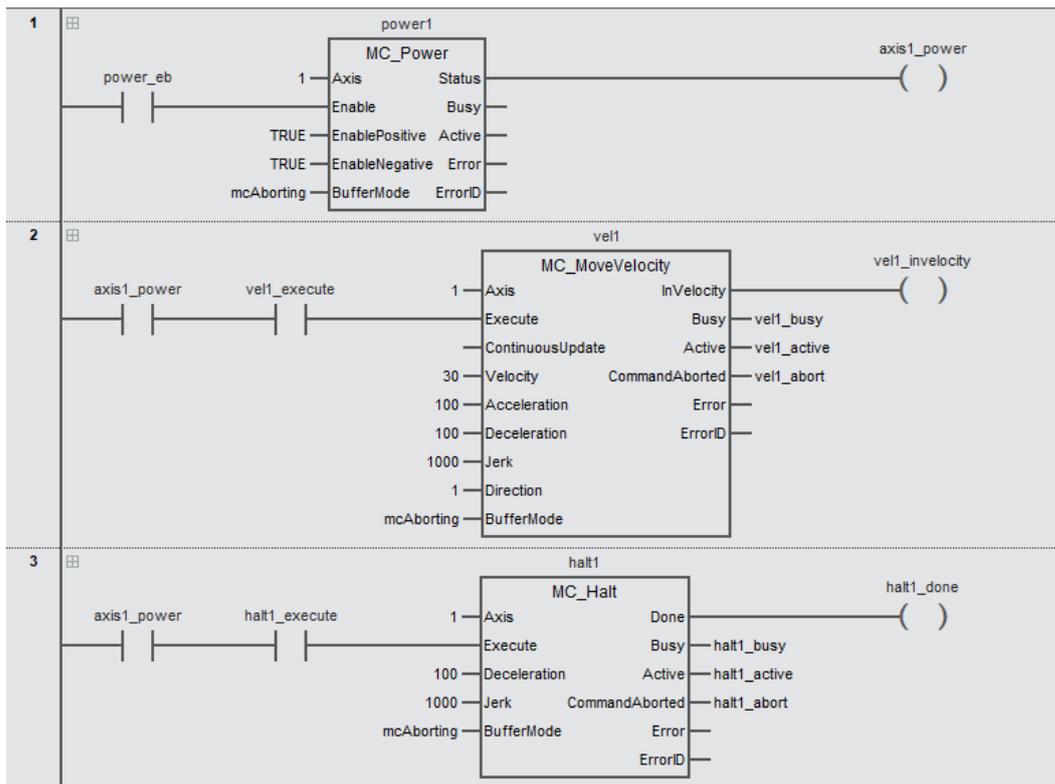
轴 1 轴参数设置如下图所示。



● 变量表

类别	名称	分配到	数据类型	初始值	注释
VAR	power_eb		BOOL		
VAR	power1		MC_Power		
VAR	axis1_power		BOOL		
VAR	vel1_execute		BOOL		
VAR	vel1		MC_MoveVelocity		
VAR	vel1_invelocity		BOOL		
VAR	vel1_busy		BOOL		
VAR	vel1_active		BOOL		
VAR	vel1_abort		BOOL		
VAR	halt1_execute		BOOL		
VAR	halt1		MC_Halt		
VAR	halt1_done		BOOL		
VAR	halt1_busy		BOOL		
VAR	halt1_active		BOOL		
VAR	halt1_abort		BOOL		

● 梯形图(LD)



● 结构化文本(ST)

```

power1(
    Axis:=1 ,
    Enable:=power_eb ,
    EnablePositive:=TRUE ,
    EnableNegative:=TRUE ,
    BufferMode:=mcAborting ,
    Status=>axis 1_power
);

vel1(
    Axis:=1 ,
    Execute:= axis 1_power AND vel1_execute ,
    Velocity:=30 ,
    Acceleration:=100 ,
    Deceleration:=100 ,
    Jerk:=1000 ,
    Direction:=1 ,
    BufferMode:=mcAborting ,
    InVelocity=> vel1_invelocity ,
    Busy=> vel1_busy ,
    Active=> vel1_active ,
    CommandAborted=> vel1_abort
);

halt1(
    Axis:=1 ,
    Execute:= axis 1_power AND halt_execute ,
    Deceleration:=100 ,

```

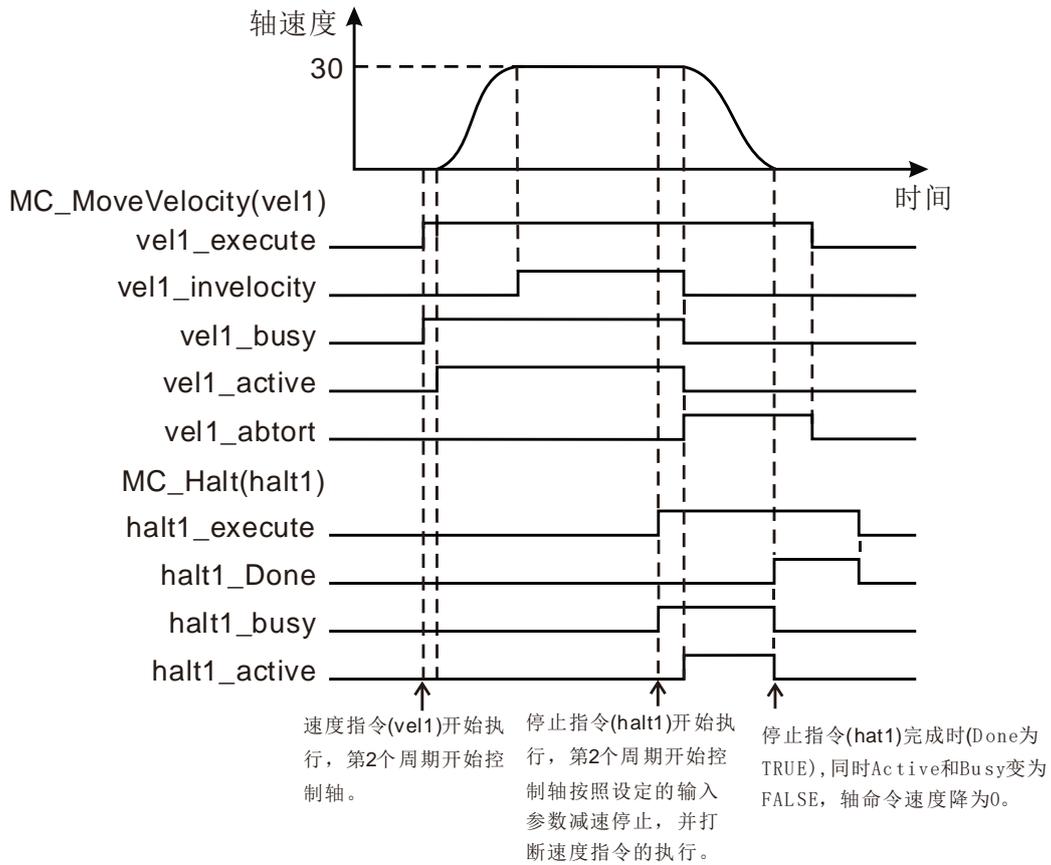
```

Jerk:=1000,
BufferMode:= mcAborting,
Done=> halt1_done,
Busy=> halt1_busy,
Active=> halt1_active,
CommandAborted=> halt1_abort
);

```

● 程序说明

- 轴使能后，vel1\_execute变为TRUE时执行速度指令控制轴运转。在需要轴停止运转时，将halt1\_execute设置为TRUE执行MC\_Halt指令，控制轴按照halt1指令设定的输入参数减速停止，halt1指令的halt1\_done变为TRUE时，轴的命令速度降为0。



## 4.10 MC\_Stop (停止并锁定)

控制指定轴从当前速度进行减速停止，停止后会锁定在停止状态。所属库：MotionControl

指令	名称	FB/FUN	梯形图样式	ST样式
MC_Stop	停止锁定	FB		<pre>MC_Stop_Instance ( Axis := 参数 , Execute := 参数 , Deceleration:= 参数 , Jerk:= 参数 , Done=&gt; 参数 , Busy =&gt; 参数 , Active=&gt; 参数 , CommandAborted =&gt; 参数 , Error =&gt; 参数 , ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Execute	启动	BOOL	TRUE或FALSE	FALSE	Execute 为TRUE时执行该指令
Deceleration	减速度	LREAL	正数	不可缺省	指定减速度*1 (单位: 行程单位/秒 <sup>2</sup> ) *2
Jerk	跃度	LREAL	正数	不可缺省	指定跃度*1 (单位: 行程单位/秒 <sup>3</sup> ) *2

\*1: Deceleration、Jerk的关系请参考“[运动控制指令相关参数说明](#)”章节。

\*2: 指令单位的详细介绍请参考“[运动控制指令参数单位](#)”章节。

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Active	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码 值的含义请参考“ <a href="#">指令错误代码描述</a> ”

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	减速完成，指定轴速度变为0时	<ul style="list-style-type: none"> <li>◆ Done为TRUE，Execute从TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时，Done变为TRUE一个周期后，变为FALSE</li> </ul>
Busy	Execute的上升沿	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>

Active	指令开始控制轴时	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
CommandAborted	指令的执行被其它指令中断时	<ul style="list-style-type: none"> <li>◆ CommandAborted 为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令被其它指令中断时, CommandAborted变为TRUE, 一个周期后, 变为FALSE</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令执行过程中遇到异常时, Error变为TRUE, 一个周期后, 变为FALSE</li> </ul>

■ 功能说明

● 基本功能说明

- 控制指定轴从当前速度进行减速停止，减速度及跃度由指令输入变量设定，停止后轴状态会锁定在停止状态。
- 该指令执行后，只要该指令输入变量Execute为TRUE，轴状态会一直维持在Stopping（停止）状态。该指令执行时，只有另外一个实例化的MC\_Stop可以中断该指令，执行其它运动指令会报错。
- 该指令执行后，轴停止后并且该指令输入变量Execute为FALSE时，轴状态变为standstill状态，此时可以执行其它运动指令。

● 重启该指令

当该指令正在执行中，Execute 再次由 FALSE 变为 TRUE 时，对指令的执行不会产生影响，指令仍继续按照之前的方式执行。

● 其它指令执行时启动该指令

其它运动指令执行时，启动该指令，该指令会中断正在执行的其它运动指令。

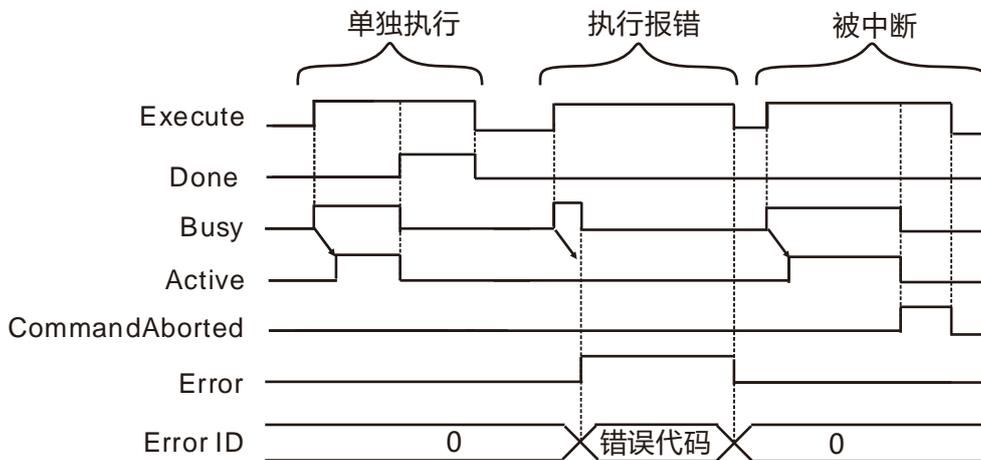
● 该指令执行时启动其它指令

该指令执行后，只要该指令输入变量 Execute 为 TRUE，轴状态会一直维持在 Stopping（停止）状态。该指令执行时，只有另外一个实例化的 MC\_Stop 可以中断该指令，执行其它运动指令会报错。

● 异常排除

该指令执行时，如果指令输入变量值不在允许的范围或者不满足指令的执行条件，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。该指令执行时，如遇到异常（如轴报警等），该指令也会报错，轴会立即停止。

■ 输出变量变化时序说明



● **单独执行**

Execute (启动) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE, 下一个周期 Active (控制中) 变为 TRUE。该指令控制轴停止后, Done (完成) 变为 TRUE, Busy (执行中) 和 Active (控制中) 仍为 TRUE。当 Execute (启动) 变为 FALSE 时, Done (完成)、Busy (执行中)、Active (控制中) 同时变为 FALSE。

● **执行报错**

该指令输入变量值不在允许范围内时, 该指令 Execute (启动) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE, 下一个周期 Error (错误) 变为 TRUE, ErrorID (错误码) 输出对应的错误码, 可通过 ErrorID (错误码) 的值, 查找发生问题原因。

● **被中断**

该指令执行后被其它指令中断时, 该指令 CommandAborted (中断) 变为 TRUE, Busy (执行中) 和 Active (控制中) 同时变为 FALSE; 当 Execute (执行中) 变为 FALSE 时, CommandAborted (中断) 同时变为 FALSE。

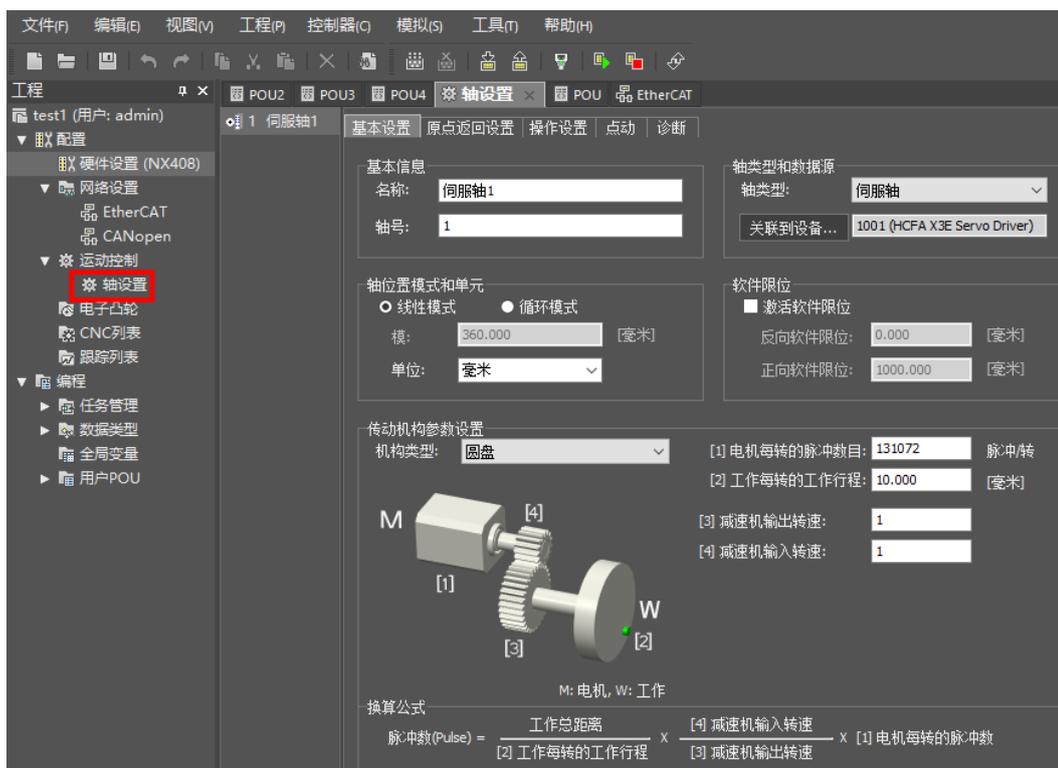
■ **示例程序**

● **实现功能**

速度指令控制轴运转时, 执行 MC\_Stop 指令控制轴按照该指令设定的输入参数减速停止。MC\_Stop 指令执行时, 执行其它运动指令会报错。如需执行其它运动指令 (非 MC\_Stop 指令), 需要将 MC\_Stop 指令的 execute 位变为 FALSE 后再执行其它运动指令。

● **轴参数设置**

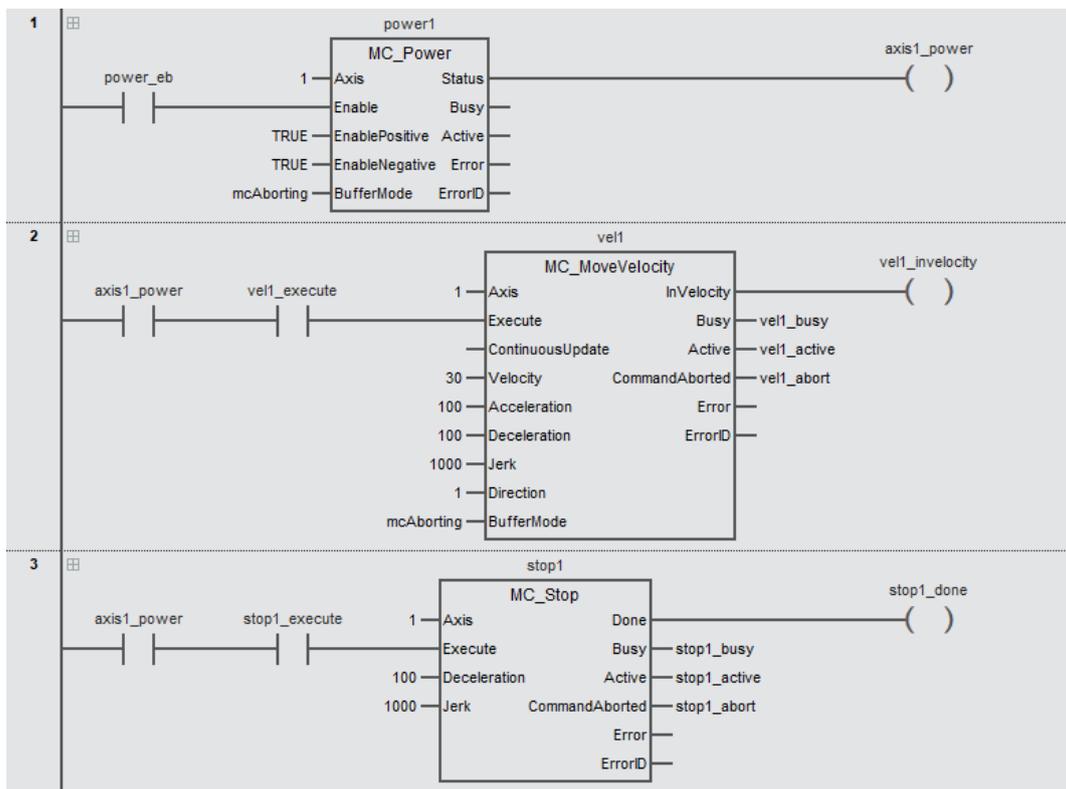
轴 1 轴参数设置如下图所示。



● 变量表

类别	名称	分配到	数据类型	初始值	注释
VAR	power_eb		BOOL		
VAR	power1		MC_Power		
VAR	axis1_power		BOOL		
VAR	vel1_execute		BOOL		
VAR	vel1		MC_MoveVelocity		
VAR	vel1_invelocity		BOOL		
VAR	vel1_busy		BOOL		
VAR	vel1_active		BOOL		
VAR	vel1_abort		BOOL		
VAR	stop1_execute		BOOL		
VAR	stop1		MC_Stop		
VAR	stop1_done		BOOL		
VAR	stop1_busy		BOOL		
VAR	stop1_active		BOOL		
VAR	stop1_abort		BOOL		

● 梯形图(LD)



● 结构化文本(ST)

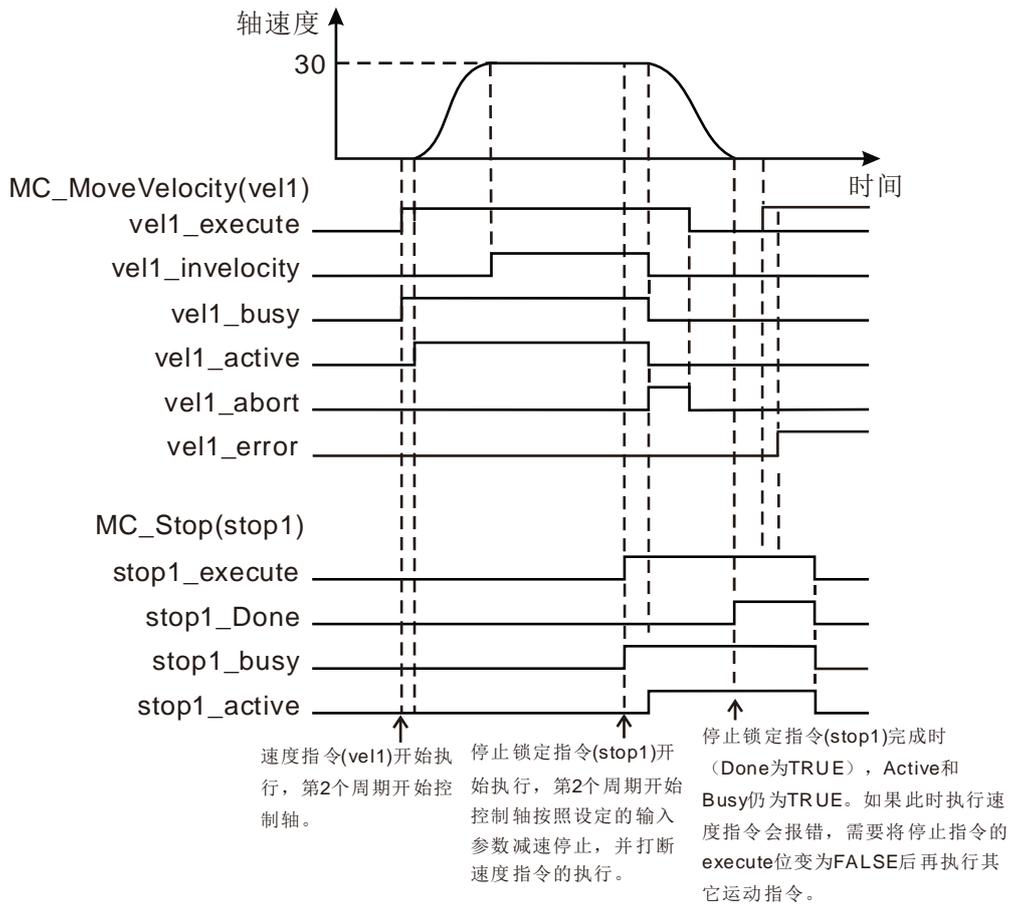
```

power1 (
    Axis:=1,
    Enable:=power_eb,
    EnablePositive:=TRUE,
    EnableNegative:=TRUE,
    BufferMode:=mcAborting,
    Status=>axis_1_power
);
    
```

```
vel1 (  
  Axis:=1 ,  
  Execute:= axis 1_power AND vel1_execute ,  
  Velocity:=30 ,  
  Acceleration:=100 ,  
  Deceleration:=100 ,  
  Jerk:=1000 ,  
  Direction:=1 ,  
  BufferMode:=mcAborting ,  
  InVelocity=> vel1_invelocity ,  
  Busy=> vel1_busy ,  
  Active=> vel1_active ,  
  CommandAborted=> vel1_abort  
);  
  
stop1(  
  Axis:=1 ,  
  Execute:= axis 1_power AND stop1_execute ,  
  Deceleration:=100 ,  
  Jerk:=1000 ,  
  Done=> halt1_done ,  
  Busy=> halt1_busy ,  
  Active=> halt1_active ,  
  CommandAborted=> halt1_abort  
);
```

- **程序说明**

- 轴使能后，vel1\_execute变为TRUE时执行速度指令控制轴运转。需要将轴停止运转时，将stop1\_execute设置为TRUE执行MC\_Stop指令，控制轴按照stop1指令设定的输入参数减速停止，stop1指令的stop\_done变为TRUE时，轴的命令速度降为0，该指令的Busy和active仍为TRUE。
- MC\_Stop指令执行时，只能再执行另外的MC\_Stop指令打断当前的MC\_Stop指令；如果执行其它运动指令（非MC\_Stop指令），其它运动指令会报错。如需执行其它运动指令，需要将MC\_Stop指令的execute位变为FALSE后再执行其它运动指令。



## 4.11 MC\_StopAtPhase (指定相位停止)

控制指定轴运转时停止在指定的相位处。所属库: MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_StopAtPhase	指定相位停止	FB		<pre> MC_StopAtPhase_Instance ( Axis := 参数 , Execute:= 参数 , Stop:= 参数 , Velocity:= 参数 , Acceleration:= 参数 , Deceleration:= 参数 , Jerk:= 参数 , Direction:= 参数 , RoundPhase:= 参数 , StopPhase:= 参数 , BufferMode:= 参数 , InVelocity =&gt; 参数 , Stop_Done =&gt; 参数 , Busy =&gt; 参数 , Active=&gt; 参数 , CommandAborted =&gt; 参数 , Error =&gt; 参数 , ErrorID=&gt; 参数 );                     </pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时, 控制指定轴按照设定的输入变量加速到Velocity设定目标速度并持续运转
Stop	停止	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时, 指定轴开始进行减速停止, 最终停止位置为StopPhase指定的相位
Velocity	目标速度	LREAL	正数	不可缺省	指定目标速度*1 (单位: 行程单位/秒) *2
Acceleration	加速度	LREAL	正数	不可缺省	指定加速度*1 (单位: 行程单位/秒 <sup>2</sup> ) *2
Deceleration	减速度	LREAL	正数	不可缺省	指定减速度*1 (单位: 行程单位/秒 <sup>2</sup> ) *2
Jerk	跃度	LREAL	正数	不可缺省	指定跃度*1 (单位: 行程单位/秒 <sup>3</sup> ) *2
Direction	方向	MC_Direction	1: mcPositiveDirection 3: mcNegativeDirection	1	检测到Execute参数的上升沿时, 轴的运转方向 1: 正方向 3: 反方向
RoundPhase	周期	LREAL	正数	不可缺省	用于计算StopPhase (停止相位) 的周期 StopPhase (停止相位) 的值为轴命令位置除以RoundPhase (周期) 计算而来
StopPhase	停止相位	LREAL	0~RoundPhase设定值	0	指定轴停止时的相位StopPhase (停止相位) 的值为轴命令位置除以RoundPhase (周期) 计算而来

BufferMode	缓存模式	MC_Buffer_Mode	0: mcAborting 1: mcBuffered 2: mcBlendingLow 3: mcBlendingPrevious 4: mcBlendingNext 5: mcBlendingHigh	0	设定两个指令之间中继模式*3 0: 中断 1: 等待 2: 以低速中继 3: 以前一个速度中继 4: 以后一个速度中继 5: 以高速中继
------------	------	----------------	---	---	--

\*1: Velocity、Acceleration、Deceleration、Jerk的关系请参考“[运动控制指令相关参数说明](#)”章节。

\*2: 指令单位的详细介绍请参考“[运动控制指令参数单位](#)”章节。

\*3: BufferMode的详细介绍请参考“[运动控制指令的多重启动时缓存模式说明 \(BufferMode\)](#)”章节。

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Invelocity	目标速度到达	BOOL	TRUE或FALSE	轴命令速度到达目标速度时变为TRUE
Stop_Done	到达指定相位	BOOL	TRUE或FALSE	指定轴位置到达指定相位时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Active	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Invelocity	目标速度到达时	<ul style="list-style-type: none"> <li>◆ 当CommandAborted为TRUE时</li> <li>◆ 当Error为TRUE时</li> <li>◆ Execute从FALSE变为TRUE时</li> </ul>
Stop_Done	当指定停止相位到达时	<ul style="list-style-type: none"> <li>◆ 当CommandAborted为TRUE时</li> <li>◆ 当Error为TRUE时</li> <li>◆ Stop为FALSE时</li> </ul>
Busy	Execute的上升沿	<ul style="list-style-type: none"> <li>◆ Stop_Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
Active	指令开始控制轴时	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
CommandAborted	指令的执行被其它指令中断时	<ul style="list-style-type: none"> <li>◆ CommandAborted 为TRUE , Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE，该指令被其它指令中断时，CommandAborted变为TRUE，一个周期后，变为FALSE</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE，该指令执行过程中遇到异常时，Error变为TRUE，一个周期后，变为FALSE</li> </ul>

### ■ 功能说明

#### ● 基本功能说明

- 该指令用于控制指定轴运转时停止在指定的相位处。该指令在Execute由FALSE变为TRUE时，以Velocity的设定值为目标速度，按照输入变量设定的值进行加速或者减速至目标速度，到达目标速度后，输出变量Invelocity为TRUE，并保持匀速运转。Stop由FALSE变为TRUE时，按照输入变量设定的值进行减速，最终停止在StopPhase设定的相位处。但指定轴可能转1圈停止在设定的相位处，也可能转2圈停止在设定的相位处。

处，该指令保证相位准确，但转几圈到达设定的相位和Stop为TRUE时指定轴的速度及输入变量有关。

- 该指令输入变量StopPhase（停止相位）和RoundPhase（周期）的单位为行程单位，和软件中“工作每转的工作行程”的单位相同。指定轴停止后StopPhase（停止相位）的值等于轴命令位置对RoundPhase（周期）取余的值。该指令使用时一般将RoundPhase和软件中“工作每转的工作行程”设置为相同的值。
- 如下图所示，StopPhase（停止相位）的值为180，RoundPhase（周期）的值为360，该指令执行后，如果轴位置在0-180之间时Stop\_Done由FALSE变为TRUE，如果轴停止的位置180，轴位置在180-540之间时Stop\_Done由FALSE变为TRUE，轴停止的位置是540。以此类推。

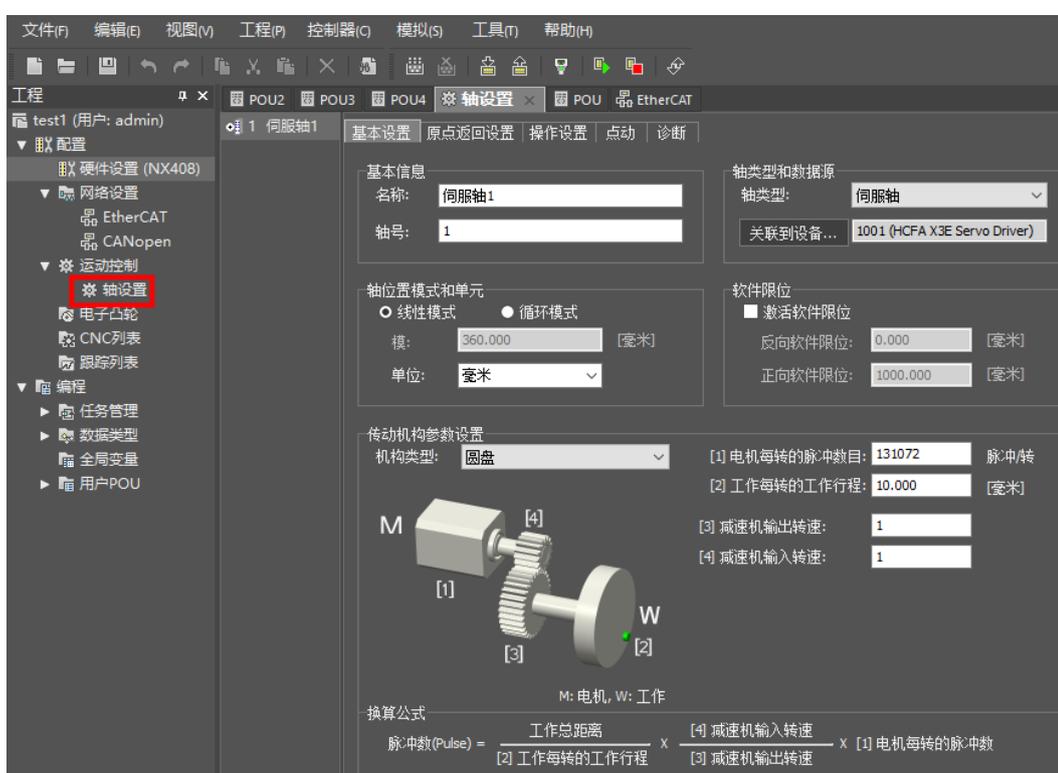
## ■ 示例程序

### ● 实现功能

执行 MC\_StopAtPhase 指令控制轴以一定速度运行，并且在设定相位处停止。

### ● 轴参数设置

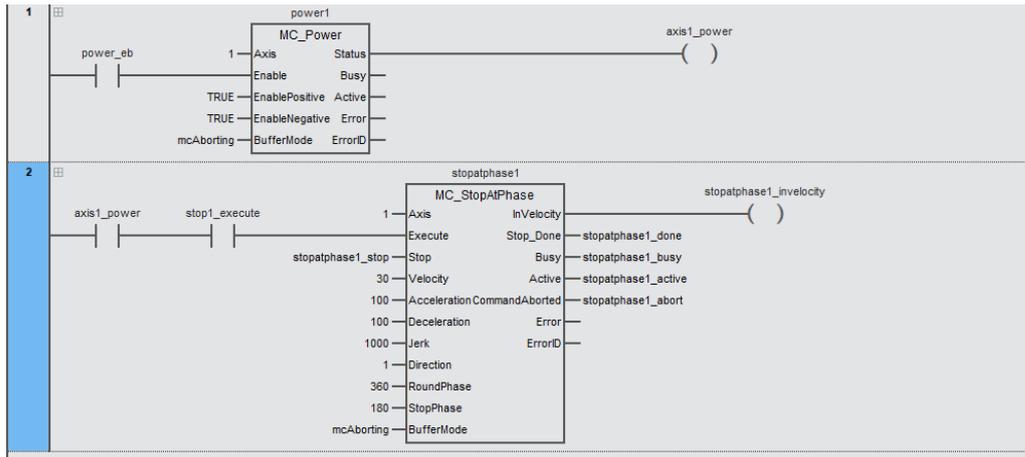
轴 1 轴参数设置如下图所示。



### ● 变量表

类别	名称	分配到	数据类型	初始值	注释
VAR	power_eb		BOOL		
VAR	power1		MC_Power		
VAR	axis1_power		BOOL		
VAR	stopatphase1_execute		BOOL		
VAR	stopatphase1		MC_StopAtPhase		
VAR	stopatphase1_invelocity		BOOL		
VAR	stopatphase1_stop		BOOL		
VAR	stopatphase1_done		BOOL		
VAR	stopatphase1_busy		BOOL		
VAR	stopatphase1_active		BOOL		
VAR	stopatphase1_abort		BOOL		

- 梯形图(LD)



- 结构化文本(ST)

```
power1
(
  Axis:=1 ,
  Enable:=power_eb ,
  EnablePositive:=TRUE ,
  EnableNegative:=TRUE ,
  BufferMode:=mcAborting ,
  Status=>axis1_power
);
```

```
stopatphase1
(
  Axis:=1 ,
  Execute:= axis1_power AND stopatphase1_execute ,
  Stop:=stopatphase1_stop
  Velocity:=30 ,
  Acceleration:=100 ,
  Deceleration:=100 ,
  Jerk:=1000 ,
  Direction:=1 ,
  RoundPhase:=360 ,
  StopPhase:=180 ,
  BufferMode:=mcAborting ,
  InVelocity=> stopatphase1_invelocity ,
  Done=> stopatphase1_done
  Busy=> stopatphase1_busy ,
  Active=> stopatphase1_active ,
  CommandAborted=> stopatphase1_abort
);
```

- 程序说明:

轴使能后，stopatphase1\_execute 变为 TRUE 时控制轴运转。在需要轴停止在设定的相位，将 stopatphase1\_stop 设置为 TRUE，控制轴按照 stopatphase1 指令设定的输入参数减速停止，halt1 指令的 stopatphase1\_done 变为 TRUE 时，轴的命令速度降为 0。

## 4.12 MC\_MoveRelative (相对位移)

控制指定轴以当前位置为参考点，移动指定距离。移动距离由Distance指定。所属库：MotionControl

指令	名称	FB/FUN	梯形图样式	ST样式
MC_MoveRelative	相对位移	FB		<pre>MC_MoveRelative_Instance ( Axis := 参数 , Execute := 参数 , ContinuousUpdate:= 参数 , Distance:= 参数 , Velocity:= 参数 , Acceleration:= 参数 , Deceleration:= 参数 , Jerk:= 参数 , BufferMode:= 参数 , Done =&gt; 参数 , Busy =&gt; 参数 , Active=&gt; 参数 , CommandAborted =&gt; 参数 , Error =&gt; 参数 , ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
ContinuousUpdate	持续更新	BOOL	TRUE或FALSE	FALSE	保留
Distance	移动距离	LREAL	正数、负数、0	0	指定以当前位置为参考点的移动距离 (单位: 行程单位) *2
Velocity	目标速度	LREAL	正数	不可缺省	指定目标速度*1 (单位: 行程单位/秒) *2
Acceleration	加速度	LREAL	正数	不可缺省	指定加速度*1 (单位: 行程单位/秒 <sup>2</sup> ) *2
Deceleration	减速度	LREAL	正数	不可缺省	指定减速度*1 (单位: 行程单位/秒 <sup>2</sup> ) *2
Jerk	跃度	LREAL	正数	不可缺省	指定跃度*1 (单位: 行程单位/秒 <sup>3</sup> ) *2
BufferMode	缓存模式	MC_Buffer_Mode	0: mcAborting 1: mcBuffered 2: mcBlendingLow 3: mcBlendingPrevious 4: mcBlendingNext 5: mcBlendingHigh	0	设定两个指令之间缓存模式。*3 0: 中断 1: 等待 2: 以低速中继 3: 以前一个速度中继 4: 以后一个速度中继 5: 以高速中继

\*1: Velocity、Acceleration、Deceleration、Jerk的关系请参考“[运动控制指令相关参数说明](#)”章节。

\*2: 指令单位的详细介绍请参考“[运动控制指令参数单位](#)”章节。

\*3: BufferMode的详细介绍请参考“[运动控制指令的多重启动时缓存模式说明 \(BufferMode\)](#)”章节。

## ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Active	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考 <a href="#">“指令错误代码描述”</a>

## ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	定位完成时	<ul style="list-style-type: none"> <li>◆ Done为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE</li> </ul>
Busy	Execute的上升沿	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
Active	指令开始控制轴时	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
CommandAborted	指令的执行被其它指令中断时	<ul style="list-style-type: none"> <li>◆ CommandAborted 为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令被其它指令中断时, CommandAborted变为TRUE, 一个周期后, 变为FALSE</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令执行过程中遇到异常时, Error变为TRUE, 一个周期后, 变为FALSE</li> </ul>

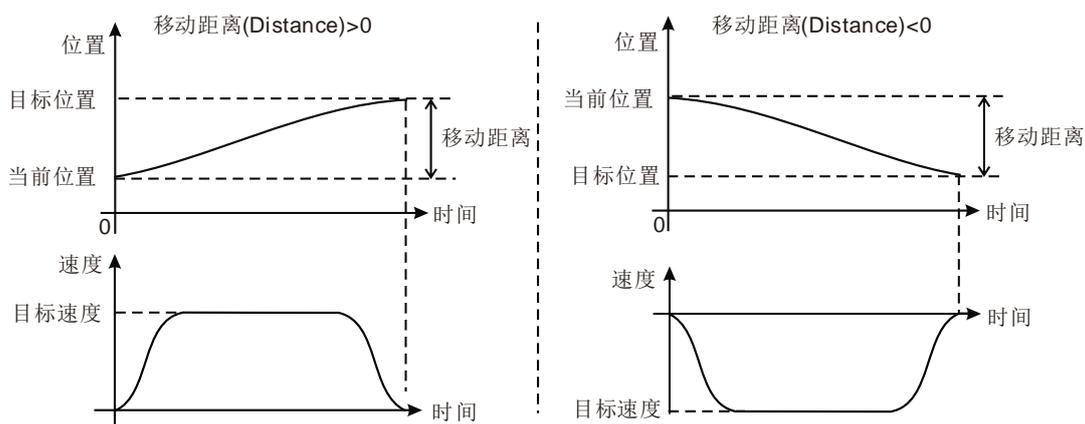
## ■ 功能说明

### ● 基本功能说明

该指令以轴当前位置为参考点，移动 Distance 指定距离。可在输入变量中指定 Distance(距离)、Velocity(目标速度)、Acceleration(加速度)、Deceleration(减速度)、Jerk(跃度)，该指令在 Execute 由 FALSE 变为 TRUE 时，按照输入变量设定的值进行运动。

### ● 距离

- 轴的目标位置由轴的当前位置和Distance的值共同决定，即目标位置= 当前位置+Distance。Distance的值可以为正数、0、负数。Velocity、Acceleration、Deceleration、Jerk的值只能为正数。
- 轴静止时执行该指令，Distance的值大于0和小于0时的位置曲线以及速度曲线如下图所示。Distance的值大于0时，轴正向运转；Distance的值小0时，则轴反向运转。
- Distance的值为0时，该指令不控制轴运转，该指令的Done在执行该指令一个扫描周期后变为TRUE。



- **指令完成时机**

当轴的命令位置到达计算的位置后，该指令完成，Done 由 FALSE 变为 TRUE。

- **重启该指令**

当指令执行完成后，Execute 再次由 FALSE 变为 TRUE 时，该指令可以重新执行；当指令正在执行中，Execute 再次由 FALSE 变为 TRUE 时，对指令的执行不会产生影响，指令仍按照未执行完成的输入变量执行指令。

- **其它指令执行时启动该指令**

其它运动指令执行时，启动该指令，可切换或缓存到该指令。该指令和其它运动指令如何中继由该指令输入变量 BufferMode（缓存模式）的值决定，而且该指令输入变量 BufferMode（缓存模式）可以设置的值和正在执行的运动指令有关。其它指令执行时启动该指令，该指令输入变量 Distance（移动距离）的参考点为该指令 Active(控制中)变为 TRUE 时轴的命令位置。

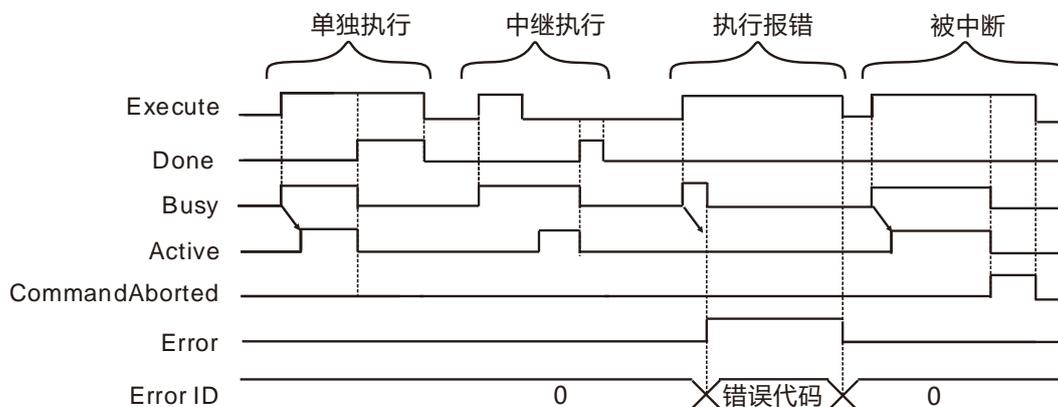
- **该指令执行时启动其它指令**

该指令执行时启动其它运动指令，其它运动指令和该指令如何中继由其它运动指令的 BufferMode（缓存模式）引脚参数值决定。其它运动指令和该指令缓存时，其它指令执行的时机为该指令“Done（完成）”变为 TRUE 时。如果其它运动指令无 BufferMode 引脚，一般为中断该指令。

- **异常排除**

该指令执行时，如果指令输入变量值不在允许的范围或者不满足指令的执行条件，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。该指令执行时，如遇到异常（如轴报警等），该指令也会报错，轴会立即停止。

- **输出变量变化时序说明**



- **单独执行**

Execute (启动) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE, 下一个周期 Active (控制中) 变为 TRUE。当达到 Distance (移动距离)、定位完成时, Done (完成) 变为 TRUE, Busy (执行中) 和 Active (控制中) 同时变为 FALSE。当 Execute (启动) 变为 FALSE 时, Done (完成) 同时变为 FALSE。

- **中继执行**

其它指令控制轴时, 执行该指令 (BufferMode 的值不为 mcAborting), 当 Execute (启动) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE, Active (控制中) 变为 TRUE 的时机为前一个指令完成时。

- **执行报错**

该指令输入变量值不在允许范围内时, 该指令 Execute (启动) 由 FALSE 变为 TRUE 时, 同时 Busy (执行中) 变为 TRUE, 下一个周期 Error (错误) 变为 TRUE, 同时 Busy (执行中) 变为 FALSE, ErrorID (错误码) 输出对应的错误码, 可通过 ErrorID (错误码) 的值, 查找发生问题原因。该指令 Execute (启动) 由 TRUE 变为 FALSE 时, 同时 Error (错误) 变为 FALSE, ErrorID (错误码) 的值变为 0。

- **被中断**

该指令执行后被其它指令中断时, 该指令 CommandAborted (中断) 变为 TRUE, Busy (执行中) 和 Active (控制中) 同时变为 FALSE; 当 Execute (执行中) 变为 FALSE 时, CommandAborted (中断) 同时变为 FALSE。

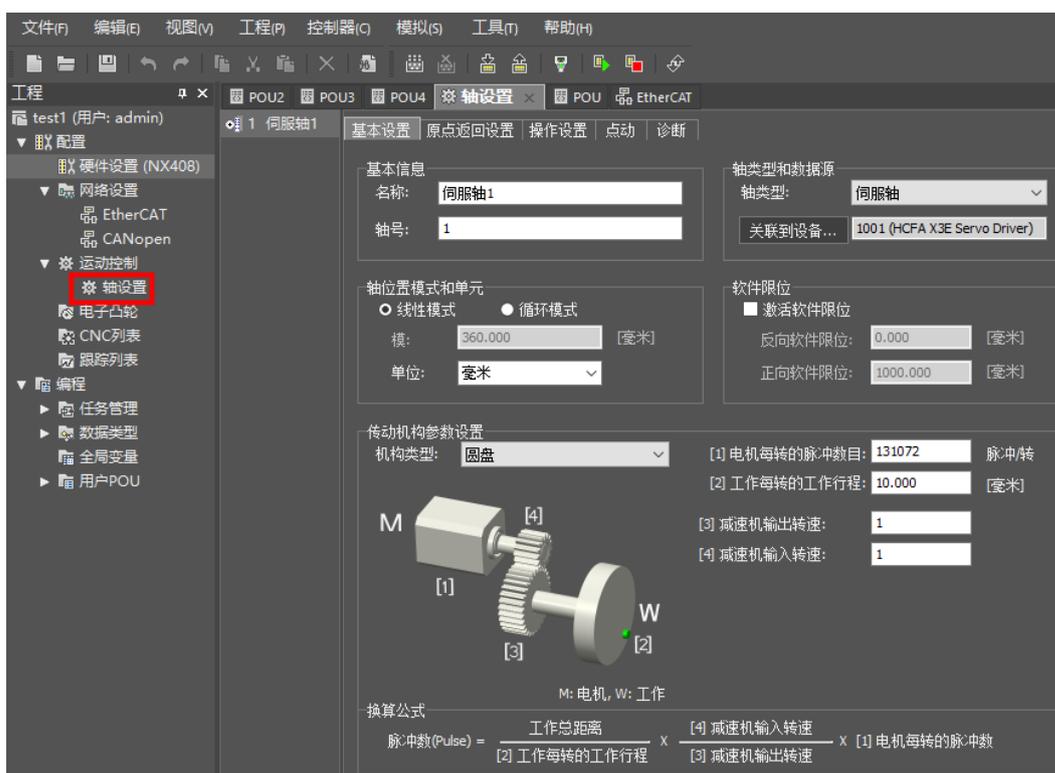
- **示例程序**

- **实现功能**

通过相对位移指令 (MC\_MoveRelative) 的 BufferMode 功能实现控制轴运动两段位移, 第一段位移运动完成时, 速度不需要降为 0 直接走第二段位移。

- **轴参数设置**

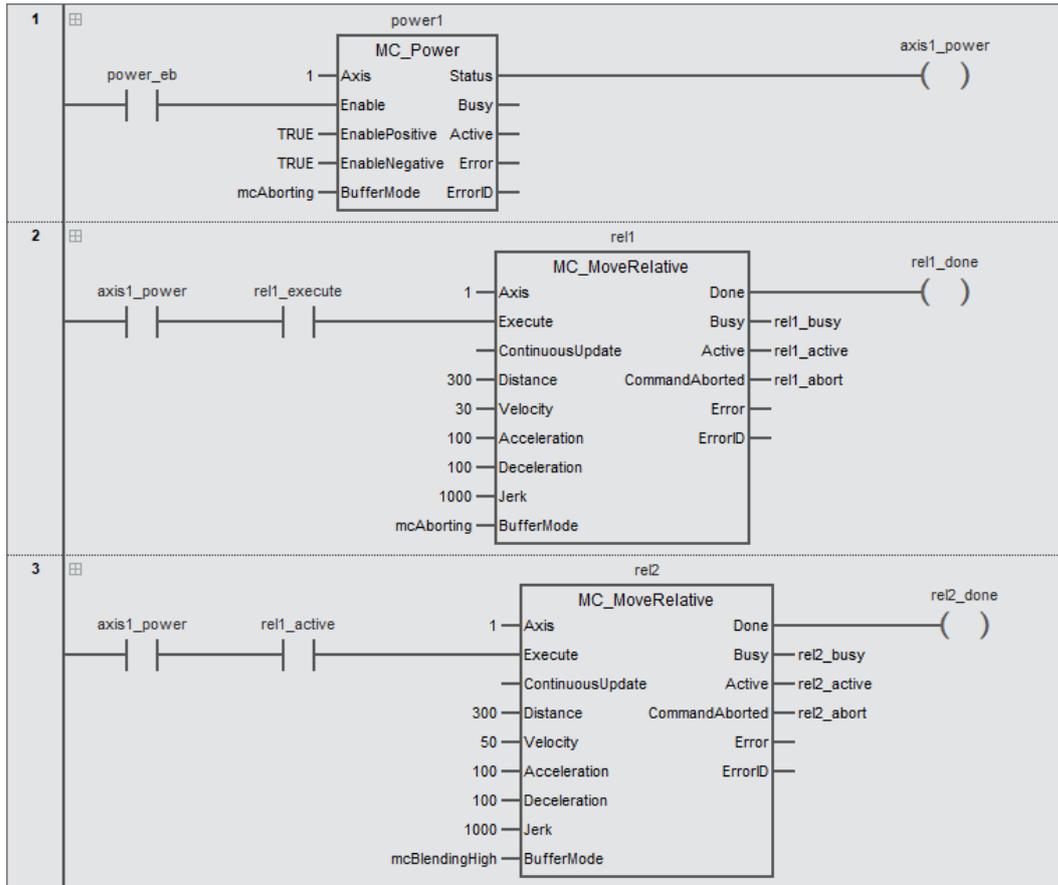
轴 1 轴参数设置如下图所示。



● 变量表

类别	名称	分配到	数据类型	初始值	注释
VAR	power_eb		BOOL		
VAR	power1		MC_Power		
VAR	axis1_power		BOOL		
VAR	rel1_execute		BOOL		
VAR	rel1		MC_MoveRelative		
VAR	rel1_done		BOOL		
VAR	rel1_busy		BOOL		
VAR	rel1_active		BOOL		
VAR	rel1_abort		BOOL		
VAR	rel2		MC_MoveRelative		
VAR	rel2_done		BOOL		
VAR	rel2_busy		BOOL		
VAR	rel2_active		BOOL		
VAR	rel2_abort		BOOL		

● 梯形图(LD)



● 结构化文本(ST)

```

power1(
  Axis:=1,
  Enable:=power_eb,
  EnablePositive:=TRUE,
  EnableNegative:=TRUE,
  BufferMode:=mcAborting,
  Status=>axis1_power
);

rel1(
  Axis:=1,
  Execute:=axis1_power AND rel1_execute,
  Distance:=300,
  Velocity:=30,
  Acceleration:=100,
  Deceleration:=100,
  Jerk:=1000,
  BufferMode:=mcAborting,
  Done=>rel1_done,
  Busy=>rel1_busy,
  Active=>rel1_active,
  CommandAborted=>rel1_abort
);
    
```

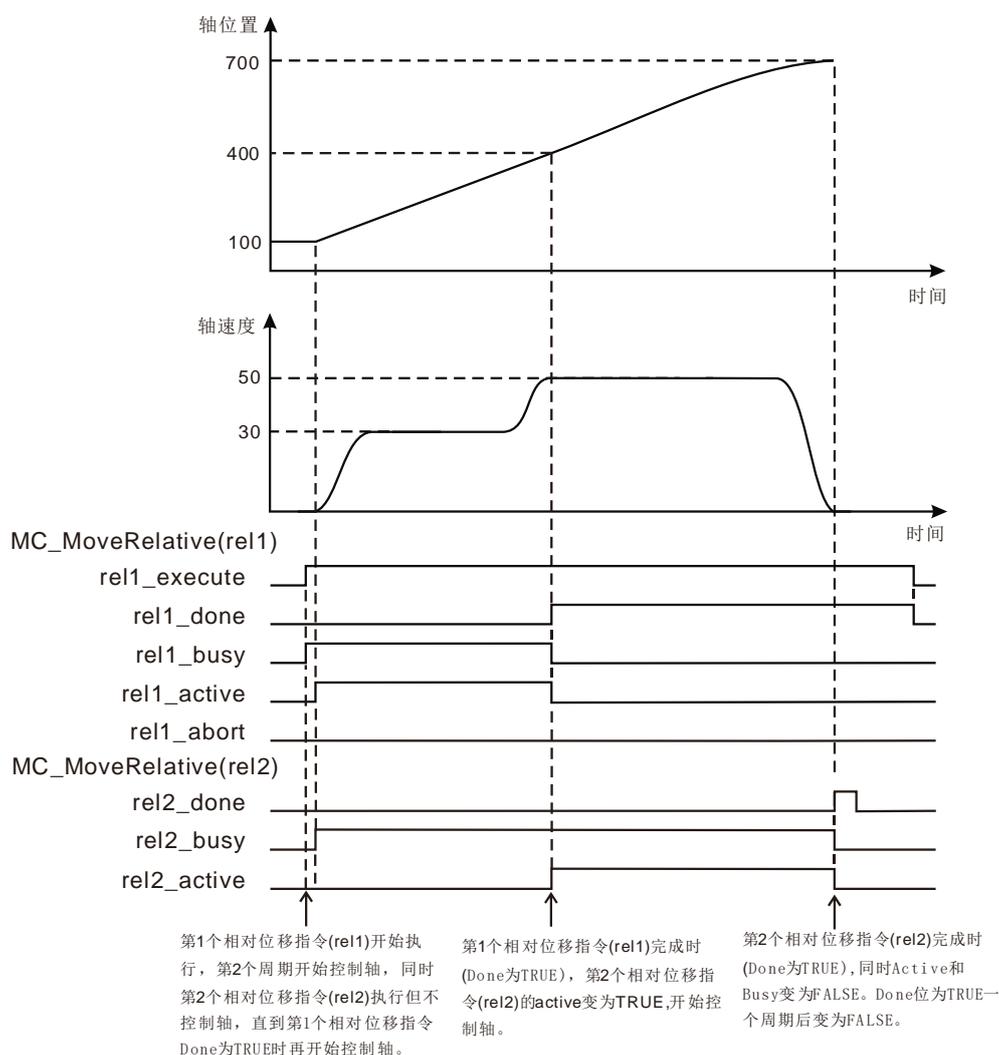
```

rel2(
  Axis:=1,
  Execute:=axis1_power AND rel1_active,
  Distance:=300,
  Velocity:=50,
  Acceleration:=100,
  Deceleration:=100,
  Jerk:=1000,
  BufferMode:=mcBlendingHigh,
  Done=>rel2_done,
  Busy=>rel2_busy,
  Active=>rel2_active,
  CommandAborted=>rel2_abort
);

```

### ● 程序说明

- 轴使能后，re1\_execute变为TRUE时，第1个相对位移指令（rel1）开始执行，re1\_execute变为TRUE的第2个周期开始控制轴，同时该指令的Active触发第2个相对位移指令（rel2）执行但不控制轴，直到第1个相对位移指令（rel1）完成时（Done为TRUE）才开始控制轴。
- 第2个相对位移指令（rel2）控制轴时，轴的命令速度为50（由rel2指令输入变量BufferMode的值决定），轴命令速度不用降为0。第2个相对位移指令（rel1）完成时（Done为TRUE），因为该指令的Execute为FALSE，Done变为TRUE一个周期后变为FALSE。



### 4.13 MC\_MoveAbsolute (绝对位移)

控制指定轴移动至以0点位置为参考点的绝对目标位置。所属库：MotionControl

指令	名称	FB/FUN	梯形图样式	ST样式
MC_MoveAbsolute	绝对位移	FB		<pre>MC_MoveAbsolute_Instance ( Axis := 参数 , Execute := 参数 , ContinuousUpdate:= 参数 , Position:= 参数, Velocity:= 参数 , Acceleration:= 参数 , Deceleration:= 参数 , Jerk:= 参数 , Direction:= 参数, BufferMode:= 参数 , Jerk:= 参数 , Done =&gt; 参数 , Busy =&gt; 参数 , Active=&gt; 参数 , CommandAborted =&gt; 参数 , Error =&gt; 参数 , ErrorID=&gt; 参数 );</pre>

#### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
ContinuousUpdate	持续更新	BOOL	TRUE或FALSE	FALSE	保留
Position	绝对位置	LREAL	正数、负数、0	0	指定以0点位置为参考点的绝对位置 (单位: 行程单位) *2
Velocity	目标速度	LREAL	正数	不可缺省	指定目标速度*1 (单位: 行程单位/秒) *2
Acceleration	加速度	LREAL	正数	不可缺省	指定加速度*1 (单位: 行程单位/秒²) *2
Deceleration	减速度	LREAL	正数	不可缺省	指定减速度*1 (单位: 行程单位/秒²) *2
Jerk	跃度	LREAL	正数	不可缺省	指定跃度*1 (单位: 行程单位/秒³) *2
Direction	方向	MC_Direction	1: mcPositiveDirection 2: mcShortestWay 3: mcNegativeDirection 4: mcCurrentDirection	1	轴运转方向。 1: 正方向 2: 移动距离最短 3: 反方向 4: 按当前运转方向运转, 轴静止时为 正方向移动
BufferMode	缓存模式	MC_Buffer_Mode	0: mcAborting 1: mcBuffered 2: mcBlendingLow 3: mcBlendingPrevious 4: mcBlendingNext 5: mcBlendingHigh	0	设定两个指令之间缓存模式。*3 0: 中断 1: 等待 2: 以低速中继 3: 以前一个速度中继 4: 以后一个速度中继 5: 以高速中继

\*1: Velocity、Acceleration、Deceleration、Jerk的关系请参考“[运动控制指令相关参数说明](#)”章节。

\*2: 指令单位的详细介绍请参考“[运动控制指令参数单位](#)”章节。

\*3: BufferMode的详细介绍请参考“[运动控制指令的多重启动时缓存模式说明 \(BufferMode\)](#)”章节。

## ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Active	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

## ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	定位完成时	<ul style="list-style-type: none"> <li>◆ Done为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE</li> </ul>
Busy	Execute的上升沿	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
Active	指令开始控制轴时	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
CommandAborted	指令的执行被其它指令中断时	<ul style="list-style-type: none"> <li>◆ CommandAborted 为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令被其它指令中断时, CommandAborted变为TRUE, 一个周期后, 变为FALSE</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令执行过程中遇到异常, Error变为TRUE, 一个周期后, 变为FALSE</li> </ul>

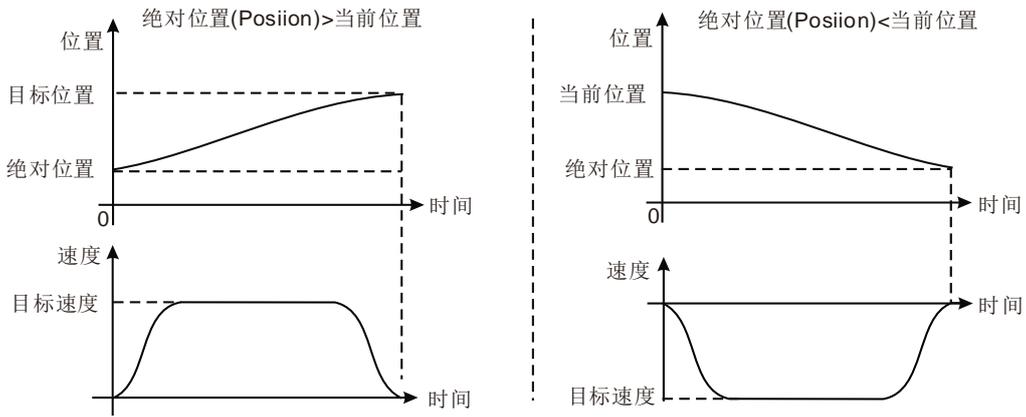
## ■ 功能说明

### ● 基本功能说明

该指令用于控制轴移动至以 0 点位置为参考点的绝对位置。可在输入变量中指定 Position（绝对位置）、Velocity（目标速度）、Acceleration（加速度）、Deceleration（减速度）、Jerk（跃度），该指令在 Execute 由 FALSE 变为 TRUE 时，按照输入变量设定的值进行运动。

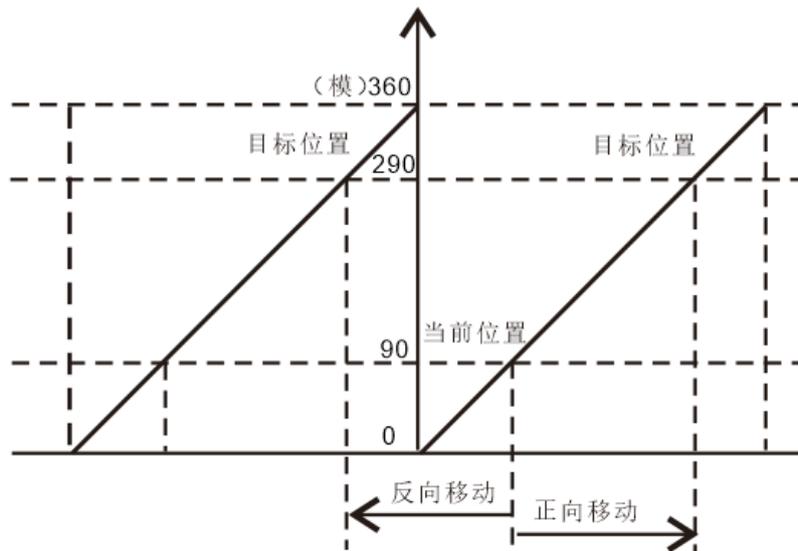
### ● 绝对位置

- Position的值可以为正数、0、负数。Velocity、Acceleration、Deceleration、Jerk的值只能为正数。轴在软件【轴配置】→【基本设置】中配置为旋转轴时，Position的值须小于软件中模的值。
- 轴在软件【轴配置】→【基本设置】中配置为直线轴时，轴静止时执行该指令，Position的值大于当前位置和小于当前位置时的位置曲线以及速度曲线如下图所示。Position的值大于当前位置时，轴正向运转；Position的值小于当前位置时，轴反向运转。
- Position的值等于当前位置时，该指令不控制轴运转，该指令的Done在执行该指令一个扫描周期后变为TRUE。



● Direction

- 轴在软件【轴配置】→【基本设置】中配置为循环模式时，输入变量Direction才有效，该参数用于设定轴的运转方向。
- 轴的模设置为360（【轴配置】→【基本设置】中配置）。如下图所示，轴当前位置为90，目标位置为290，Direction选择不同的值，轴静止时执行该指令，对应轴的运转方向如下表所示。Direction的值为2（移动距离最短）时，反向移动时的距离为 $160=360-290+90$ ，正向移动时的距离为 $200=290-90$ ，反向移动时距离短，以为轴实际运转方向为反向移动。



参数	Direction			
	参数值	mcPositiveDirection	mcShortestWay	mcNegativeDirection
含义	正方向	移动距离最短	反方向	按当前运转方向运转，轴静止时为正向移动
轴运转方向	正向移动	反向移动	反向移动	正向移动

- Direction的值为mcCurrentDirection（按当前运转方向移动，轴静止时为正向移动）时，执行该指令，轴移动方向按照该指令触发执行时的移动方向进行移动。如执行该指令前轴为正向移动，使用该指令中断其它指令时，轴仍按正向移动到达目标位置；如执行该指令前轴为静止，执行该指令后，控制轴正向运转。

- **指令完成时机**

当轴的命令位置到达计算的位置后，该指令完成，Done 由 FALSE 变为 TRUE。

- **重启该指令**

当指令执行完成后，Execute 再次由 FALSE 变为 TRUE 时，该指令可以重新执行；当指令正在执行中，Execute 再次由 FALSE 变为 TRUE 时，对指令的执行不会产生影响，指令仍按照未执行完成的输入变量执行指令。

- **其它指令执行时启动该指令**

其它运动指令执行时，启动该指令，可切换或缓存到该指令。该指令和其它运动指令如何中继由该指令输入变量 BufferMode（缓存模式）的值决定，而且该指令输入变量 BufferMode（缓存模式）可以设置的值和正在执行的运动指令有关。其它指令执行时启动该指令，该指令输入变量 Position（绝对位置）的参考点为该指令 Active(控制中)变为 TRUE 时轴的命令位置。

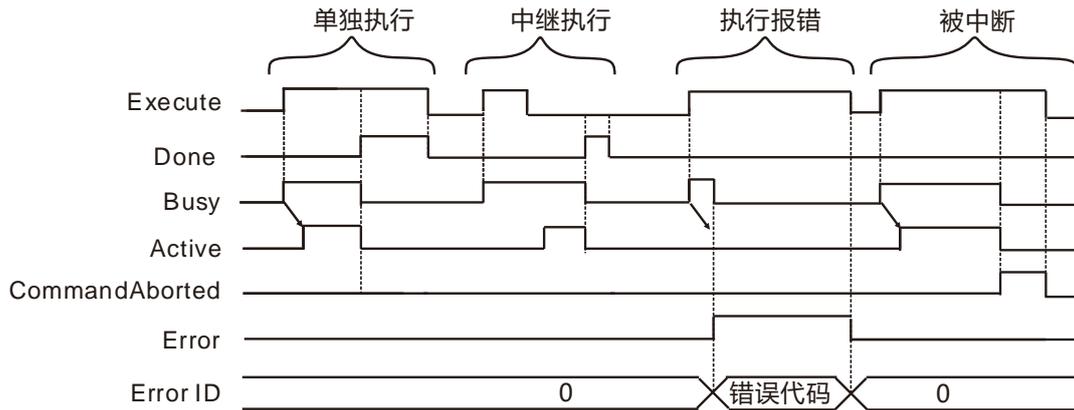
- **该指令执行时启动其它指令**

该指令执行时启动其它运动指令，其它运动指令和该指令如何中继由其它运动指令的 BufferMode（缓存模式）引脚参数值决定。其它运动指令和该指令缓存时，其它指令执行的时机为该指令“Done（完成）”变为 TRUE 时。如果其它运动指令无 BufferMode 引脚，一般为中断该指令。

- **异常排除**

该指令执行时，如果指令输入变量值不在允许的范围或者不满足指令的执行条件，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。该指令执行时，如遇到异常（如轴报警等），该指令也会报错，轴会立即停止。

## ■ 输出变量变化时序说明



### ● 单独执行

Execute (启动) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE, 下一个周期 Active (控制中) 变为 TRUE。当达到 Distance (移动距离)、定位完成时, Done (完成) 变为 TRUE, Busy (执行中) 和 Active (控制中) 同时变为 FALSE。当 Execute (启动) 变为 FALSE 时, Done (完成) 同时变为 FALSE。

### ● 中继执行

其它指令控制轴时, 执行该指令 (BufferMode 的值不为 mcAborting), 当 Execute (启动) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE, Active (控制中) 变为 TRUE 的时机为前一个指令完成时。

### ● 执行报错

该指令输入变量值不在允许范围内时, 该指令 Execute (启动) 由 FALSE 变为 TRUE 时, 同时 Busy (执行中) 变为 TRUE, 下一个周期 Error (错误) 变为 TRUE, 同时 Busy (执行中) 变为 FALSE, ErrorID (错误码) 输出对应的错误码, 可通过 ErrorID (错误码) 的值, 查找发生问题原因。该指令 Execute (启动) 由 TRUE 变为 FALSE 时, 同时 Error (错误) 变为 FALSE, ErrorID (错误码) 的值变为 0。

### ● 被中断

该指令执行后被其它指令中断时, 该指令 CommandAborted (中断) 变为 TRUE, Busy (执行中) 和 Active (控制中) 同时变为 FALSE; 当 Execute (执行中) 变为 FALSE 时, CommandAborted (中断) 同时变为 FALSE。

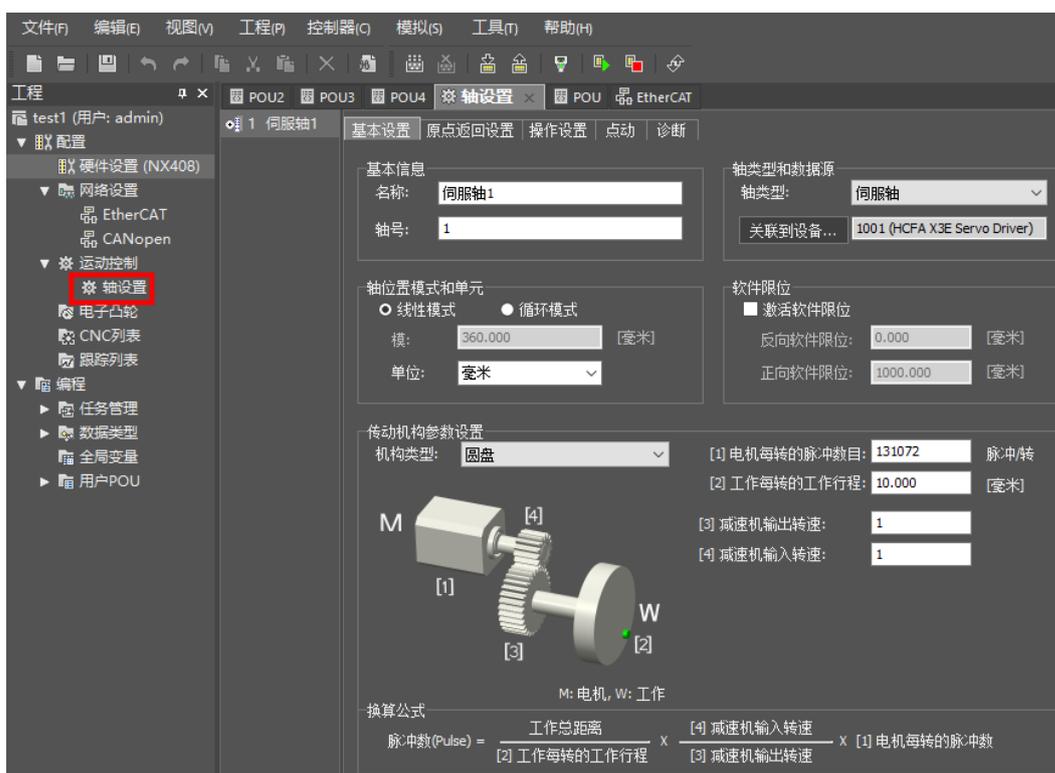
## ■ 示例程序

### ● 实现功能

通过绝对位移指令 (MC\_MoveAbsolute) 的 BufferMode 功能实现功能实现控制轴运动两段位移, 第一段位移运动完成时, 速度不需要降为 0 直接走第二段位移。

### ● 轴参数设置

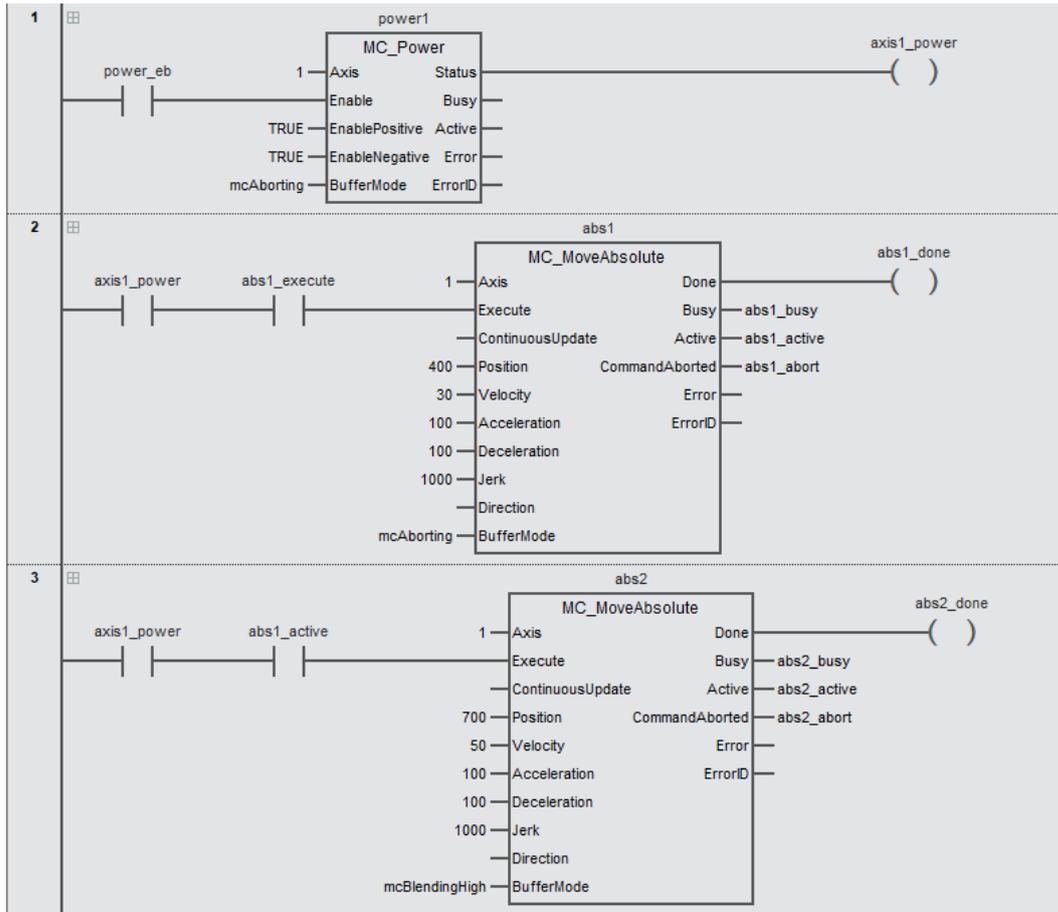
轴 1 轴参数设置如下图所示。



● 变量表

类别	名称	分配到	数据类型	初始值	注释
VAR	power_eb		BOOL		
VAR	power1		MC_Power		
VAR	axis1_power		BOOL		
VAR	abs1_execute		BOOL		
VAR	abs1		MC_MoveAbsolute		
VAR	abs1_done		BOOL		
VAR	abs1_busy		BOOL		
VAR	abs1_active		BOOL		
VAR	abs1_abort		BOOL		
VAR	abs2		MC_MoveAbsolute		
VAR	abs2_done		BOOL		
VAR	abs2_busy		BOOL		
VAR	abs2_active		BOOL		
VAR	abs2_abort		BOOL		

● 梯形图(LD)



● 结构化文本(ST)

```

power1 (
    Axis:=1 ,
    Enable:=power_eb ,
    EnablePositive:=TRUE ,
    EnableNegative:=TRUE ,
    BufferMode:=mcAborting ,
    Status=>axis1_power
);

abs1(
    Axis:=1 ,
    Execute:=axis1_power AND abs1_execute ,
    Position:=400 ,
    Velocity:=30 ,
    Acceleration:=100 ,
    Deceleration:=100 ,
    Jerk:=1000 ,
    BufferMode:=mcAborting ,
    Done=>abs1_done ,
    Busy=>abs1_busy ,
    Active=>abs1_active ,
    CommandAborted=>abs1_abort
);
    
```

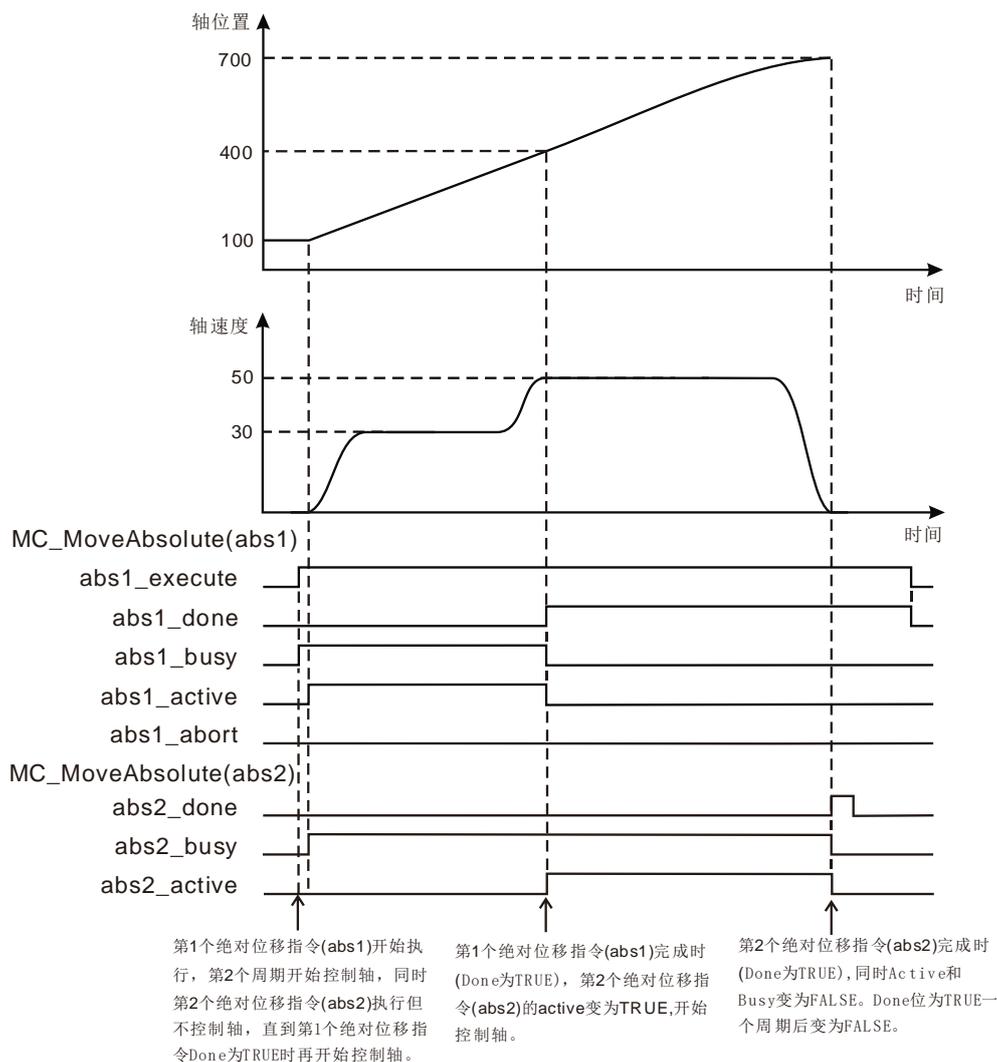
```

abs2(
  Axis:=1,
  Execute:=axis1_power AND abs1_active,
  Position:=700,
  Velocity:=50,
  Acceleration:=100,
  Deceleration:=100,
  Jerk:=1000,
  BufferMode:=mcBlendingHigh,
  Done=>abs2_done,
  Busy=>abs2_busy,
  Active=>abs2_active,
  CommandAborted=>abs2_abort
);

```

### ● 程序说明

- 轴使能后，轴的当前位置为100，abs1\_execute变为TRUE时，第1个绝对位移指令（abs1）开始执行，abs1\_execute变为TRUE的第2个周期开始控制轴，同时该指令的Active触发第2个绝对位移指令（abs2）执行但不控制轴，直到第1个绝对位移指令（abs1）完成时（Done为TRUE）才开始控制轴。
- 第2个绝对位移指令（abs2）控制轴时，轴的命令速度为50（由abs2指令输入变量BufferMode的值决定），轴命令速度不用降为0。第2个绝对位移指令（abs2）完成时（Done为TRUE），因为该指令的Execute为FALSE，Done变为TRUE一个周期后变为FALSE。



## 4.14 MC\_MoveAdditive (附加位移)

控制指定轴移动一段附加距离，最终的目标位置为其它指令目标位置和该指令指定附加距离的和。所属库：MotionControl

指令	名称	FB/FUN	梯形图样式	ST样式
MC_MoveAdditive	附加位移	FB		<pre>MC_MoveAdditive_Instance ( Axis := 参数 , Execute := 参数 , ContinuousUpdate := 参数 , Distance := 参数 , Velocity := 参数 , Acceleration := 参数 , Deceleration := 参数 , Jerk := 参数 , BufferMode := 参数 , Done =&gt; 参数 , Busy =&gt; 参数 , Active =&gt; 参数 , CommandAborted =&gt; 参数 , Error =&gt; 参数 , ErrorID =&gt; 参数 );</pre>

## ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
ContinuousUpdate	持续更新	BOOL	TRUE或FALSE	FALSE	保留
Distance	移动距离	LREAL	正数、负数、0	0	指定以当前位置为参考点的移动距离 (单位: 行程单位) *2
Velocity	目标速度	LREAL	正数	不可缺省	指定目标速度*1 (单位: 行程单位/秒) *2
Acceleration	加速度	LREAL	正数	不可缺省	指定加速度*1 (单位: 行程单位/秒 <sup>2</sup> ) *2
Deceleration	减速度	LREAL	正数	不可缺省	指定减速度*1 (单位: 行程单位/秒 <sup>2</sup> ) *2
Jerk	跃度	LREAL	正数	不可缺省	指定跃度*1 (单位: 行程单位/秒 <sup>3</sup> ) *2
BufferMode	缓存模式	MC_Buffer_Mode	0: mcAborting 1: mcBuffered 2: mcBlendingLow 3: mcBlendingPrevious 4: mcBlendingNext 5: mcBlendingHigh	0	设定两个指令之间缓存模式。*3 0: 中断 1: 等待 2: 以低速中继 3: 以前一个速度中继 4: 以后一个速度中继 5: 以高速中继

\*1: Velocity、Acceleration、Deceleration、Jerk的关系请参考“[运动控制指令相关参数说明](#)”章节。

\*2: 指令单位的详细介绍请参考“[运动控制指令参数单位](#)”章节。

\*3: BufferMode的详细介绍请参考“[运动控制指令的多重启动时缓存模式说明 \(BufferMode\)](#)”章节。

## ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Active	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

## ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	定位完成时	<ul style="list-style-type: none"> <li>◆ Done为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE</li> </ul>
Busy	Execute的上升沿	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
Active	指令开始控制轴时	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
CommandAborted	指令的执行被其它指令中断时	<ul style="list-style-type: none"> <li>◆ CommandAborted 为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令被其它指令中断时, CommandAborted变为TRUE, 一个周期后, 变为FALSE</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令执行过程中遇到异常时, Error变为TRUE, 一个周期后, 变为FALSE</li> </ul>

## ■ 功能说明

### ● 基本功能说明

- 控制指定轴移动一段附加距离，最终的目标位置为其它指令目标位置 and 该指令指定附加距离的和。可在输入变量中指定Distance（距离）、Velocity（目标速度）、Acceleration（加速度）、Deceleration（减速度）、Jerk（跃度），该指令在Execute由FALSE变为TRUE时，按照输入变量设定的值进行运动。Distance的值可以为正数、0、负数。Velocity、Acceleration、Deceleration、Jerk的值只能为正数。
- 当没有其它指令控制轴运转时，单独执行该指令，和相对位移指令MC\_MoveRelative的功能相同，用于控制轴按照设定速度，加减速以及跃度移动设定的距离，该距离的参考起点为指令开始执行时的轴位置。
- 该指令中断位移相关指令时，该指令执行完成后，终端执行机构移动的距离为其它指令剩余的距离和该指令设定的附加距离总和；该指令中断非位移相关指令（如速度指令等）时，该指令会中断其它指令执行，并按设定的速度，加减速移动设定的距离后停止，附加距离的参考起点为该指令开始执行时轴的当前位置。
- MC\_MoveSuperimposed单独执行时，执行该指令，如果该指令的输入变量BufferMode的值为mcAborting（中断），则该指令会中断正在执行的MC\_MoveSuperimposed指令，最终的目标位置为MC\_MoveSuperimposed指令执行前的轴当前位置、该指令通过Distance指定的附加距离、MC\_MoveSuperimposed通过Distance指定的距离总和；若该指令输入变量BufferMode的值不为mcAborting（中断），则执行该指令会报错。
- MC\_MoveSuperimposed与其它位移指令同时执行时，执行该指令，如果该指令的输入变量BufferMode的

值为mcAborting（中断），则该指令会中断正在执行的所有位移指令，最终的目标位置为该指令通过Distance指定的附加距离、MC\_MoveSuperimposed通过Distance指定的距离、其它位移指令目标位置的总和；若该指令输入变量BufferMode的值不为mcAborting（中断），则该指令根据BufferMode的值和MC\_MoveSuperimposed同时执行的其它位移指令进行中继，其它指令目标位置到达后再执行该指令。

- **指令完成时机**

当轴的命令位置到达计算的位置后，该指令完成，Done 由 FALSE 变为 TRUE。

- **重启该指令**

当指令执行完成后，Execute 再次由 FALSE 变为 TRUE 时，该指令可以重新执行；当指令正在执行中，Execute 再次由 FALSE 变为 TRUE 时，对指令的执行不会产生影响，指令仍按照未执行完成的输入变量执行指令。

- **其它指令执行时启动该指令**

其它运动指令执行时，启动该指令，可切换或缓存到该指令。该指令和其它运动指令如何中继由该指令输入变量 BufferMode（缓存模式）的值决定，而且该指令输入变量 BufferMode（缓存模式）可以设置的值和正在执行的运动指令有关。其它指令执行时启动该指令，该指令输入变量 Distance（移动距离）的参考点为该指令 Active(控制中)变为 TRUE 时轴的命令位置。

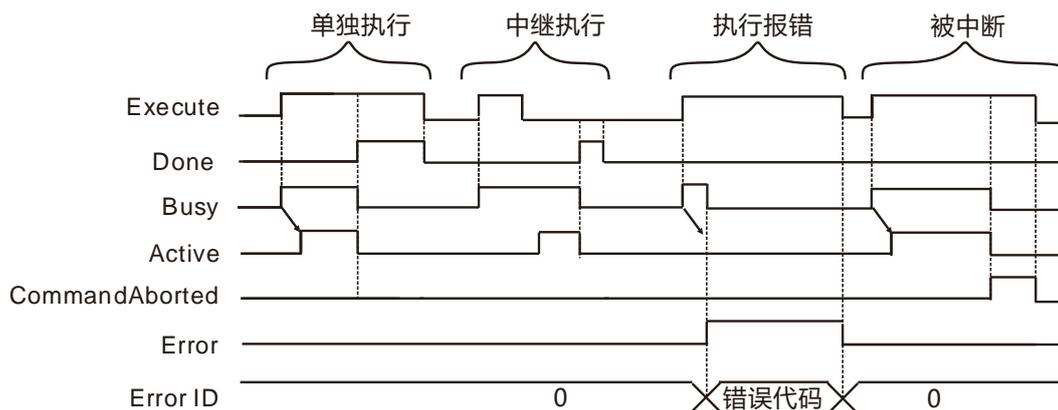
- **该指令执行时启动其它指令**

该指令执行时启动其它运动指令，其它运动指令和该指令如何中继由其它运动指令的 BufferMode（缓存模式）引脚参数值决定。其它运动指令和该指令缓存时，其它指令执行的时机为该指令 Done（完成）变为 TRUE 时。如果其它运动指令无 BufferMode 引脚，一般为中断该指令。

- **异常排除**

该指令执行时，如果指令输入变量值不在允许的范围或者不满足指令的执行条件，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。该指令执行时，如遇到异常（如轴报警等），该指令也会报错，轴会立即停止。

## ■ 输出变量变化时序说明



### ● 单独执行

Execute (启动) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE, 下一个周期 Active (控制中) 变为 TRUE。当达到 Distance (移动距离)、定位完成时, Done (完成) 变为 TRUE, Busy (执行中) 和 Active (控制中) 同时变为 FALSE。当 Execute (启动) 变为 FALSE 时, Done (完成) 同时变为 FALSE。

### ● 中继执行

其它指令控制轴时, 执行该指令 (BufferMode 的值不为 mcAborting), 当 Execute (启动) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE, Active (控制中) 变为 TRUE 的时机为前一个指令完成时。

### ● 执行报错

该指令输入变量值不在允许范围内时, 该指令 Execute (启动) 由 FALSE 变为 TRUE 时, 同时 Busy (执行中) 变为 TRUE, 下一个周期 Error (错误) 变为 TRUE, 同时 Busy (执行中) 变为 FALSE, ErrorID (错误码) 输出对应的错误码, 可通过 ErrorID (错误码) 的值, 查找发生问题原因。该指令 Execute (启动) 由 TRUE 变为 FALSE 时, 同时 Error (错误) 变为 FALSE, ErrorID (错误码) 的值变为 0。

### ● 被中断

该指令执行后被其它指令中断时, 该指令 CommandAborted (中断) 变为 TRUE, Busy (执行中) 和 Active (控制中) 同时变为 FALSE; 当 Execute (执行中) 变为 FALSE 时, CommandAborted (中断) 同时变为 FALSE。

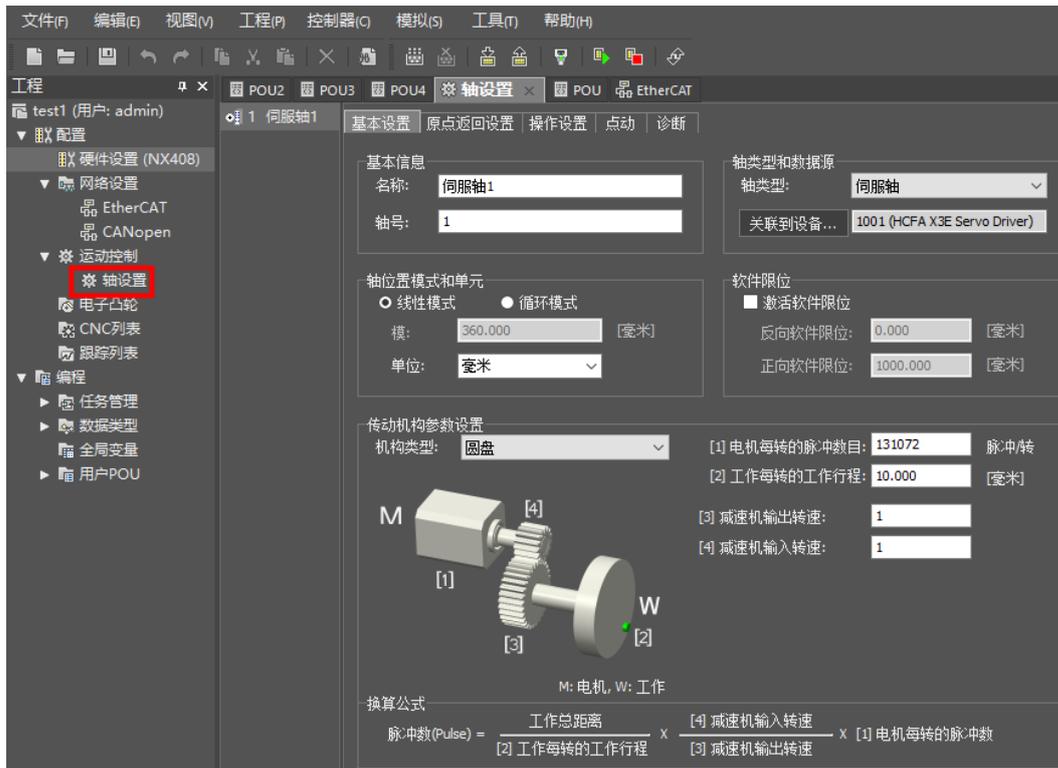
## ■ 示例程序

### ● 实现功能

相对位移指令 (MC\_MoveRelative) 控制轴运转时被附加位移指令 (MC\_MoveAdditive) 打断, 打断后, 轴以附加位移指令 (MC\_MoveAdditive) 设定的速度为目标速度, 该指令完成后, 轴的移动距离为相对位移指令和附加位移指令设定的距离之和。

### ● 轴参数设置

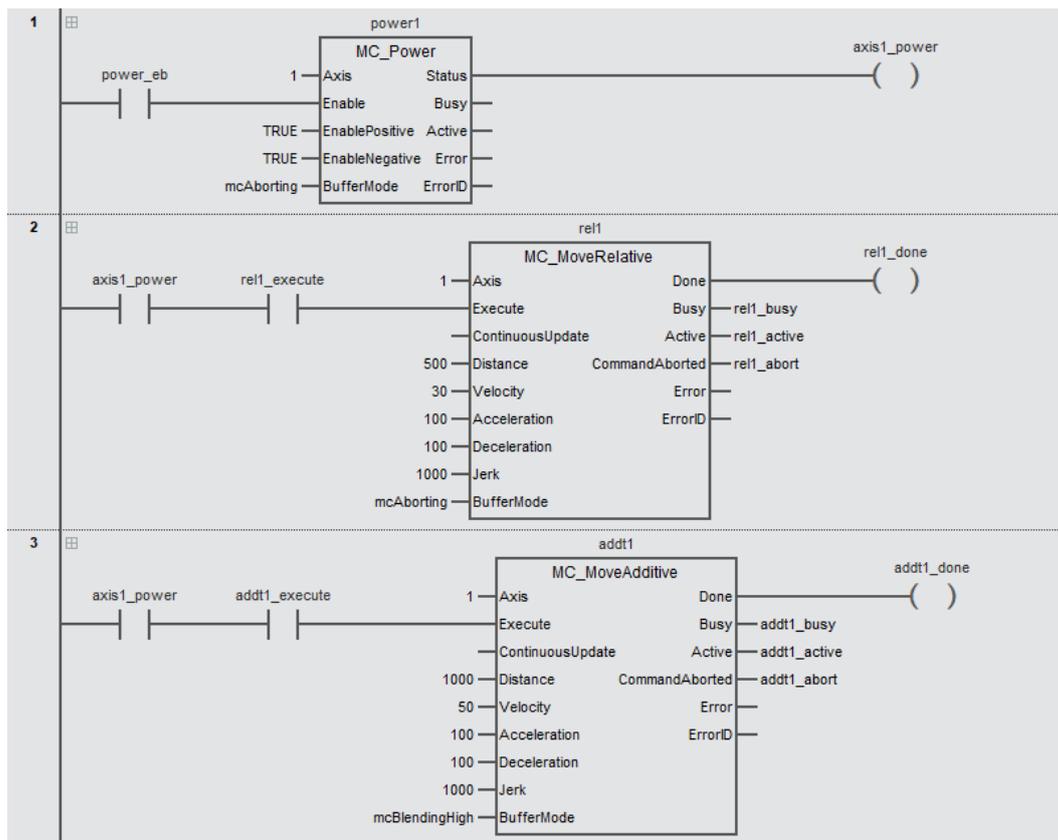
轴 1 轴参数设置如下图所示。



● 变量表

类别	名称	分配到	数据类型	初始值	注释
VAR	power_eb		BOOL		
VAR	power1		MC_Power		
VAR	axis1_power		BOOL		
VAR	rel1_execute		BOOL		
VAR	rel1		MC_MoveRelative		
VAR	rel1_done		BOOL		
VAR	rel1_busy		BOOL		
VAR	rel1_active		BOOL		
VAR	rel1_abort		BOOL		
VAR	addt1_execute		BOOL		
VAR	addt1		MC_MoveAdditive		
VAR	addt1_done		BOOL		
VAR	addt1_busy		BOOL		
VAR	addt1_active		BOOL		
VAR	addt1_abort		BOOL		

- 梯形图(LD)



- 结构化文本(ST)

```

power1 (
  Axis:=1 ,
  Enable:=power_eb ,
  EnablePositive:=TRUE ,
  EnableNegative:=TRUE ,
  BufferMode:=mcAborting ,
  Status=>axis1_power
);

rel1(
  Axis:=1 ,
  Execute:=axis1_power AND rel1_execute ,
  Distance:=500 ,
  Velocity:=30 ,
  Acceleration:=100 ,
  Deceleration:=100 ,
  Jerk:=100 ,
  BufferMode:=mcAborting ,
  Done=>rel1_done ,
  Busy=>rel1_busy ,
  Active=>rel1_active ,
  CommandAborted=>rel1_abort
);

addt1 (
  Axis:=1 ,
  Execute:= axis1_power AND addt1_execute ,

```

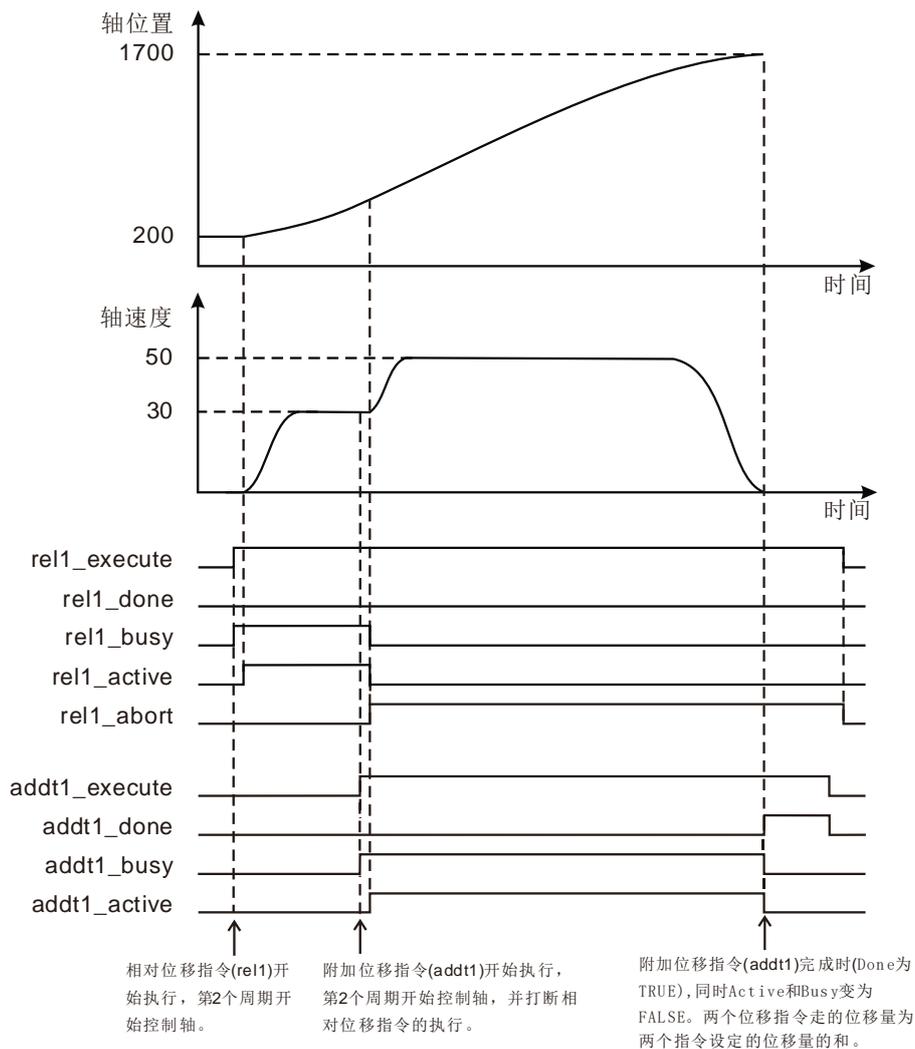
```

Distance:=1000,
Velocity:=50,
Acceleration:=100,
Deceleration:=100,
Jerk:=1000,
BufferMode:= mcBlendingHigh,
Done=> addt1_done,
Busy=> addt1_busy,
Active=>addt1_active,
CommandAborted=> addt1_abort
);

```

### ● 程序说明

- 轴使能后，rel1\_execute变为TRUE时，相对位移指令（rel1）开始执行。相对位移指令（rel1）控制轴运转时，addt1\_execute变为TRUE时，附加位移指令执行（addt1），附加位移指令控制轴后打断相对位移指令的执行。
- 附加位移指令打断相对位移指令的执行后，轴以附加位移指令（addt1）设定的速度为目标速度。如下图所示，附加位移指令（addt1）没有执行时，轴的速度为30，执行后，轴的目标速度变为50。附加位移指令执行完成后，轴的移动距离为相对位移指令和附加位移指令设定的距离之和。如下图所示，轴的初始位置为200，附加位移指令完成后，轴的位置为1700，移动距离为1700-200=1500，为相对位移指令和附加位移指令设定的距离之和，相对位移指令设定的移动距离为500，附加位移指令设定的移动距离为1000。



## 4.15 MC\_MoveSuperimposed (叠加相对位移)

该指令对指定轴产生的位置、速度、加速度和其它指令产生的位置、速度、加速度进行叠加。所属库：MotionControl

指令	名称	FB/FUN	梯形图样式	ST样式
MC_MoveSuperImposed	叠加相对位移	FB		<pre> MC_MoveSuperimposed_Instance ( Axis := 参数 , Execute := 参数 , ContinuousUpdate:= 参数 , Distance:= 参数 , Velocity:= 参数 , Acceleration:= 参数 , Deceleration:= 参数 , Jerk:= 参数 , Done =&gt; 参数 , Busy =&gt; 参数 , Active=&gt; 参数 , CommandAborted =&gt; 参数 , Error =&gt; 参数 , ErrorID=&gt; 参数 CoveredDistance =&gt; 参数 ); </pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
ContinuousUpdate	持续更新	BOOL	TRUE或FALSE	FALSE	保留
Distance	移动距离	LREAL	正数、负数、0	0	指定以当前位置为参考点的移动距离 (单位: 行程单位) *2
Velocity	目标速度	LREAL	正数	不可缺省	指定目标速度*1 (单位: 行程单位/秒) *2
Acceleration	加速度	LREAL	正数	不可缺省	指定加速度*1 (单位: 行程单位/秒 <sup>2</sup> ) *2
Deceleration	减速度	LREAL	正数	不可缺省	指定减速度*1 (单位: 行程单位/秒 <sup>2</sup> ) *2
Jerk	跃度	LREAL	正数	不可缺省	指定跃度*1 (单位: 行程单位/秒 <sup>3</sup> ) *2

\*1: Velocity、Acceleration、Deceleration、Jerk的关系请参考“[运动控制指令相关参数说明](#)”章节。

\*2: 指令单位的详细介绍请参考“[运动控制指令参数单位](#)”章节。

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Active	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时, 输出错误代码 值的含义请参考“ <a href="#">指令错误代码描述</a> ”
CoveredDistance	已移动的距离	LREA	正数、负数、0	该指令执行后, 该指令控制轴已经移动的距离

## ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	定位完成时	<ul style="list-style-type: none"> <li>◆ Done为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE</li> </ul>
Busy	Execute的上升沿	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
Active	指令开始控制轴时	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
CommandAborted	指令的执行被其它指令中断时	<ul style="list-style-type: none"> <li>◆ CommandAborted 为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令被其它指令中断时, CommandAborted变为TRUE, 一个周期后, 变为FALSE</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令执行过程中遇到异常时, Error变为TRUE, 一个周期后, 变为FALSE</li> </ul>

## ■ 功能说明

### ● 基本功能说明

- 该指令用于将该指令产生的位置、速度、加速度和其它运动指令产生的位置、速度、加速度进行叠加。常用于实际值和理论值不符时的误差修正。该指令可以对多轴关系中的从轴或者主轴操作，一般对从轴执行该指令。
- 该指令以轴当前位置为参考点，移动Distance指定距离。可在输入变量中指定Distance（距离）、Velocity（目标速度）、Acceleration（加速度）、Deceleration（减速度）、Jerk（跃度），该指令在Execute由FALSE变为TRUE时，按照输入变量设定的值进行运动。
- 轴静止时，单独执行该指令实现的功能和执行MC\_MoveRelative指令实现的功能相同。
- 其它指令执行时，再执行该指令控制同一个轴，该指令不会中断正在执行的指令，和其它指令并行执行，该指令产生的位置、速度、加速度和其它指令产生的位置、速度、加速度进行叠加。

### ● 重启该指令

当指令执行完成后，Execute 再次由 FALSE 变为 TRUE 时，该指令可以重新执行；当指令正在执行中，Execute 再次由 FALSE 变为 TRUE 时，对指令的执行不会产生影响，指令仍继续按照之前的方式执行。

### ● 其它指令执行时启动该指令

- 其它指令执行时，再执行该指令控制同一个轴，该指令不会中断正在执行的指令，和其它指令并行执行，该指令产生的位置、速度和其它指令产生的位置、速度进行叠加。
- 其它指令和该指令都执行时（控制同一个轴），如果再执行另外一个该指令（控制同一个轴），后执行的该指令会中断前面执行的该指令，对原来执行的其它指令无影响。

### ● 该指令执行时启动其它指令

- 该指令单独执行时，如果再执行MC\_HaltSuperimposed指令（控制同一个轴），该指令会被中断。
- 该指令单独执行时，如果再执行另外一个该指令（控制同一个轴），后执行的该指令会中断前面执行的该指令。
- 其它指令和该指令都执行时（控制同一个轴），如果再执行MC\_HaltSuperimposed指令（控制同一个轴），该指令会被中断，对原来执行的其它指令无影响。
- 其它指令和该指令都执行时（控制同一个轴），如果再执行MC\_MoveAdditive指令（控制同一个轴），详细

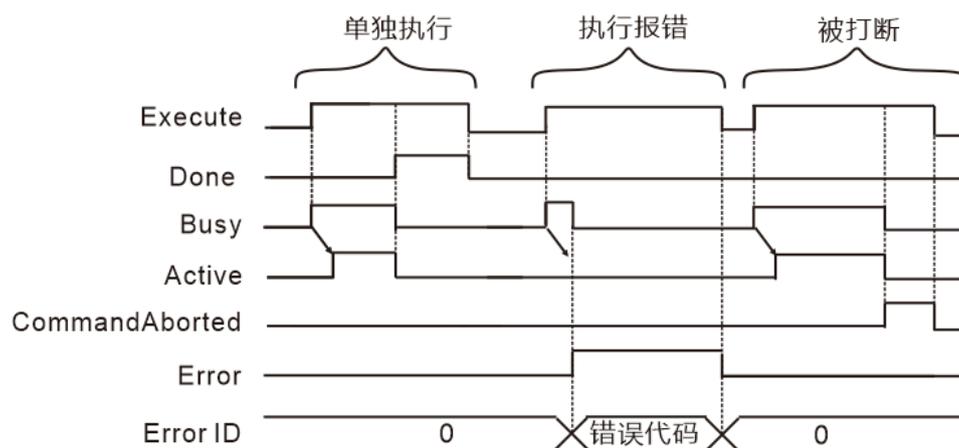
说明请参考MC\_MoveAdditive指令的功能说明。

- 其它指令和该指令都执行时（控制同一个轴），如果再执行其它运动指令（控制同一个轴，不包括MC\_MoveSuperimposed、MC\_HaltSuperimposed、MC\_MoveAdditive指令），如果后执行的运动指令无BufferMode参数，其它指令和该指令都被中断；如果后执行的运动指令BufferMode参数选择mcAborting（中断），其它指令和该指令都被中断；如果后执行的运动指令BufferMode参数选择不是mcAborting（中断），该指令被中断，其它指令仍正常执行。

#### ● 异常排除

该指令执行时，如果输入变量不合法，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。该指令执行时，如遇到异常（如轴报警等），该指令也会报错，轴会立即停止。

#### ■ 输出变量变化时序说明



#### ● 单独执行

Execute（启动）由 FALSE 变为 TRUE 时，Busy（执行中）同时变为 TRUE，下一个周期 Active（控制中）变为 TRUE。当达到 Distance（移动距离）、定位完成时，Done（完成）变为 TRUE，Busy（执行中）和 Active（控制中）同时变为 FALSE。当 Execute（启动）变为 FALSE 时，Done（完成）同时变为 FALSE。

#### ● 执行报错

该指令输入变量值不在允许范围内时，该指令 Execute（启动）由 FALSE 变为 TRUE 时，同时 Busy（执行中）变为 TRUE，下一个周期 Error（错误）变为 TRUE，同时 Busy（执行中）变为 FALSE，ErrorID（错误码）输出对应的错误码，可通过 ErrorID（错误码）的值，查找发生问题原因。该指令 Execute（启动）由 TRUE 变为 FALSE 时，同时 Error（错误）变为 FALSE，ErrorID（错误码）的值变为 0。

#### ● 被中断

该指令执行后被其它指令中断时，该指令 CommandAborted（中断）变为 TRUE，Busy（执行中）和 Active（控制中）同时变为 FALSE；当 Execute（执行中）变为 FALSE 时，CommandAborted（中断）同时变为 FALSE。

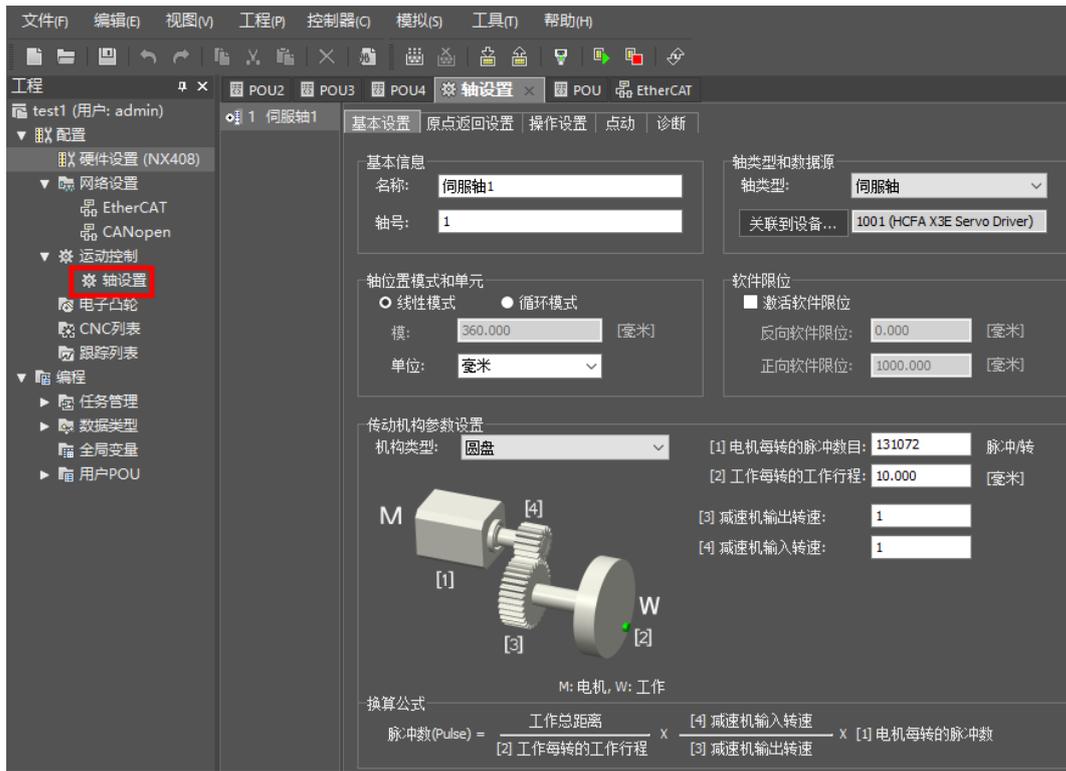
#### ■ 示例程序

##### ● 实现功能

MC\_MoveRelative（相对位移指令）和 MC\_MoveSuperimposed（叠加相对位移指令）对同一个轴一起控制时，可以将两个指令的速度、加速度、位置进行叠加后作用在同一个轴上。一般用于实际值和理论值不符时的位置修正。

##### ● 轴参数设置

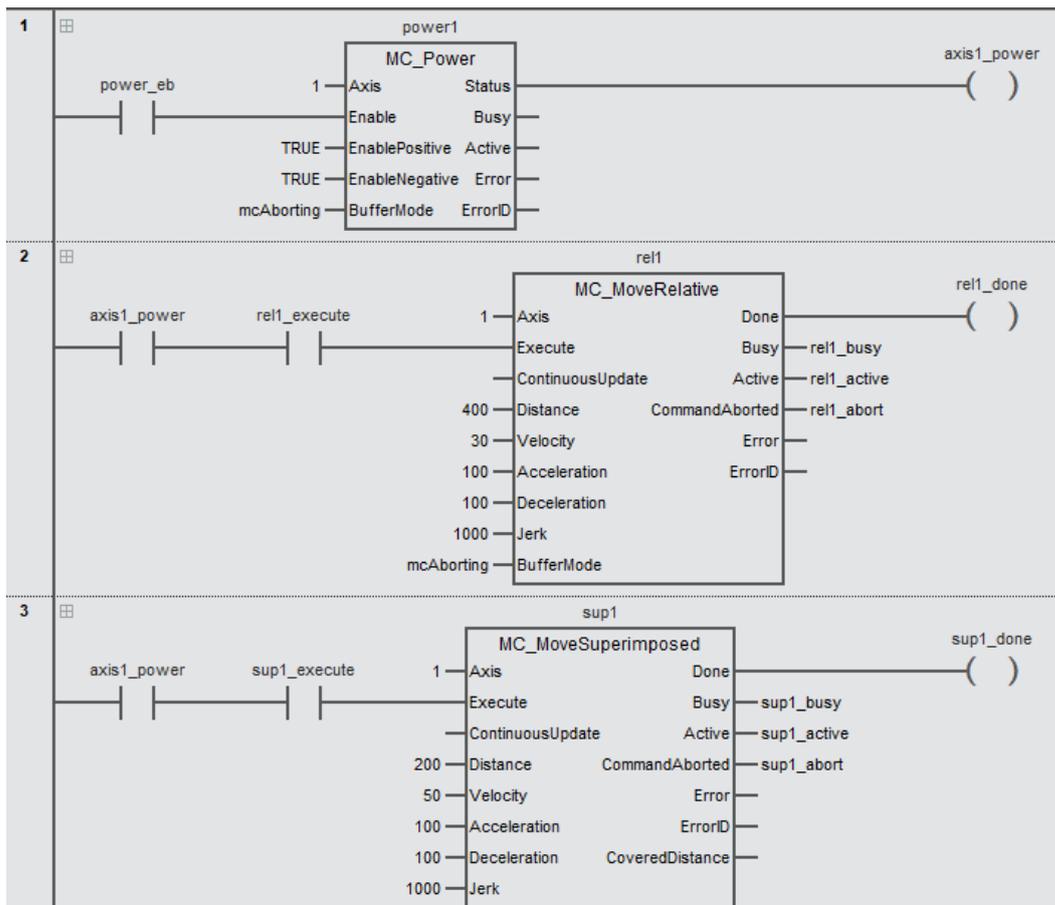
轴 1 轴参数设置如下图所示。



● 变量表

类别	名称	分配到	数据类型	初始值	注释
VAR	power_eb		BOOL		
VAR	power1		MC_Power		
VAR	axis1_power		BOOL		
VAR	rel1_execute		BOOL		
VAR	rel1		MC_MoveRelative		
VAR	rel1_done		BOOL		
VAR	rel1_busy		BOOL		
VAR	rel1_active		BOOL		
VAR	rel1_abort		BOOL		
VAR	sup1_execute		BOOL		
VAR	sup1		MC_MoveSuperimposed		
VAR	sup1_done		BOOL		
VAR	sup1_busy		BOOL		
VAR	sup1_active		BOOL		
VAR	sup1_abort		BOOL		

● 梯形图(LD)



● 结构化文本(ST)

```

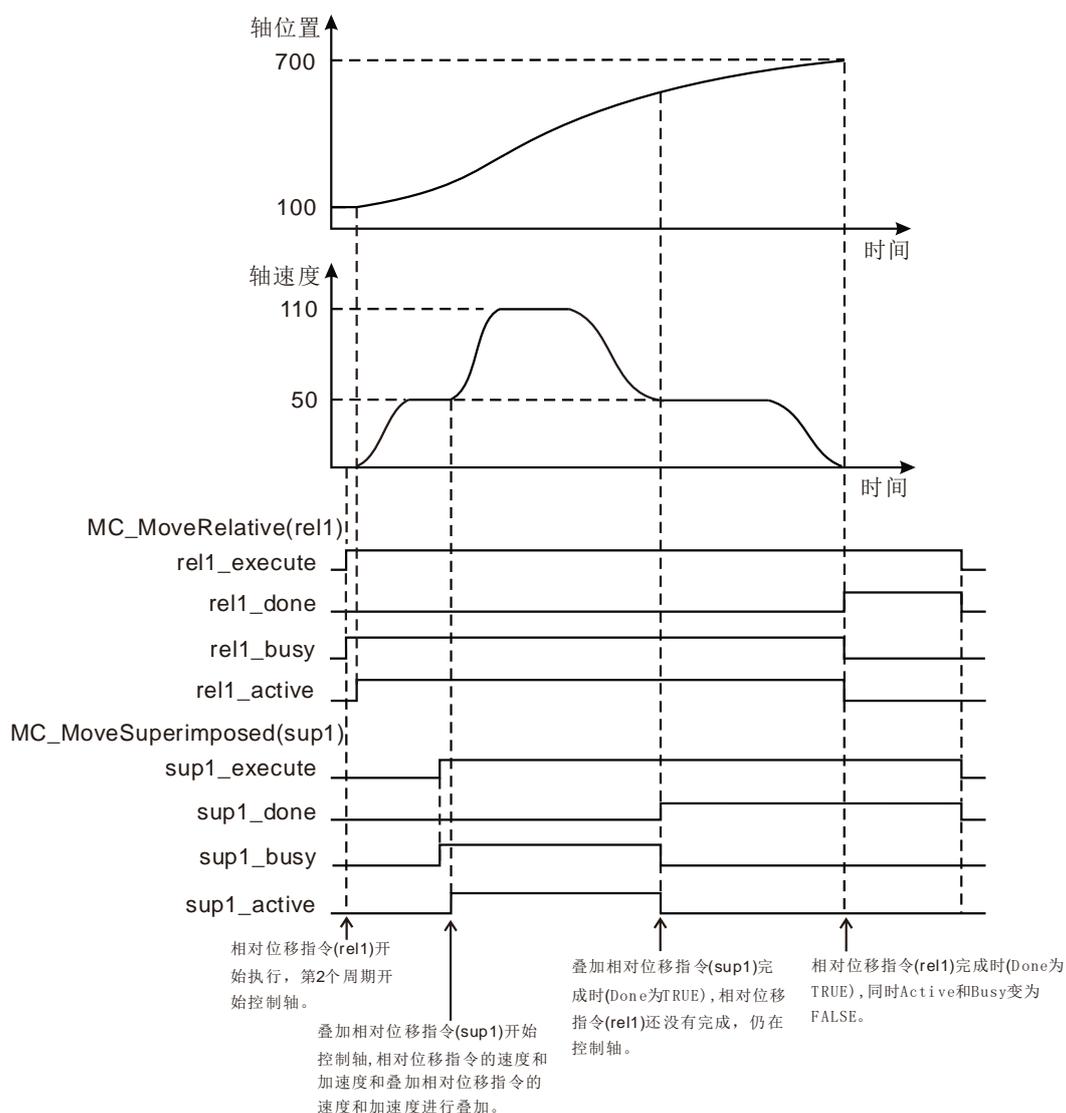
power1 (
  Axis:=1 ,
  Enable:=powerEb ,
  EnablePositive:=TRUE ,
  EnableNegative:=TRUE ,
  BufferMode:=mcAborting ,
  Status=>axis1_power
);

rel1(
  Axis:=1 ,
  Execute:=axis1_power AND rel1_execute ,
  Distance:=400 ,
  Velocity:=50 ,
  Acceleration:=100 ,
  Deceleration:=100 ,
  Jerk:=100 ,
  BufferMode:=mcAborting ,
  Done=>rel1_done ,
  Busy=>rel1_busy ,
  Active=>rel1_active ,
  CommandAborted=>rel1_abort
);
    
```

```
sup1 (  
  Axis:=1 ,  
  Execute:= axis1_power AND sup1_execute ,  
  Distance:=200 ,  
  Velocity:=60 ,  
  Acceleration:=100 ,  
  Deceleration:=100 ,  
  Jerk:=1000 ,  
  Done=> sup1_done ,  
  Busy=> sup1_busy ,  
  Active=>sup1_active ,  
  CommandAborted=> sup1_abort  
);
```

#### ● 程序说明

- 轴使能后，rel\_execute变为TRUE时，相对位移指令（rel1）开始执行。相对位移指令（rel1）控制轴运转时，sup1\_execute变为TRUE时，叠加相对位移指令（sup1）执行，叠加相对位移指令控制轴后，两个指令的产生的速度、加速度、位置进行叠加后作用在一个轴上。
- 如下图所示，叠加相对位移指令（sup1）没有执行时，轴的速度为50（相对位移指令的目标速度），叠加相对位移指令（sup1）执行并达到设定速度后，轴的速度为110，为相对位移指令目标速度（50）和叠加相对位移指令目标速度（60）的叠加。当叠加相对位移指令完成时（设定的目标位置走完），相对位移指令还没有完成，相对位移指令继续控制轴运转，轴速度变为相对位移指令设定的目标速度。
- 相对位移指令执行完成后，轴的移动距离为相对位移指令和叠加位移指令设定的距离之和。如下图所示，轴的初始位置为100，相对位移指令完成后，轴的位置为700，移动距离为700-100=600，为相对位移指令和叠加位移指令设定的距离之和，相对位移指令设定的移动距离为400，叠加位移指令设定的移动距离为200。



## 4.16 MC\_HaltSuperimposed (停止叠加位移)

控制指定轴按照设定的减速度，跃度停止叠加位移。所属库：MotionControl

指令	名称	FB/FUN	梯形图样式	ST样式
MC_HaltSuperImposed	停止叠加位移	FB		<pre> MC_HaltSuperImposed_Instance ( Axis := 参数, Execute := 参数, Deceleration:= 参数, Jerk:= 参数, Done =&gt; 参数, Busy =&gt; 参数, Active=&gt; 参数, CommandAborted =&gt; 参数, Error =&gt; 参数, ErrorID=&gt; 参数 );                     </pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
Deceleration	减速度	LREAL	正数	不可缺省	指定减速度* <sup>1</sup> (单位: 行程单位/秒 <sup>2</sup> ) * <sup>2</sup>
Jerk	跃度	LREAL	正数	不可缺省	指定跃度* <sup>1</sup> (单位: 行程单位/秒 <sup>3</sup> ) * <sup>2</sup>

\*1: Deceleration、Jerk的关系请参考“[运动控制指令相关参数说明](#)”章节。

\*2: 指令单位的详细介绍请参考“[运动控制指令参数单位](#)”章节。

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Active	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	定位完成时	<ul style="list-style-type: none"> <li>◆ Done为TRUE，Execute从TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时，Done变为TRUE一个周期后，变为FALSE</li> </ul>
Busy	Execute的上升沿	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>

Active	指令开始控制轴时	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
CommandAborted	指令的执行被其它指令中断时	<ul style="list-style-type: none"> <li>◆ CommandAborted 为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令被其它指令中断时, CommandAborted变为TRUE, 一个周期后, 变为FALSE</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令执行过程中遇到异常时, Error变为TRUE, 一个周期后, 变为FALSE</li> </ul>

## ■ 功能说明

### ● 基本功能说明

- 控制指定轴按照设定的减速度，跃度停止叠加位移。
- MC\_MoveSuperimposed指令执行时，执行该指令，该指令会中断MC\_MoveSuperimposed指令；MC\_MoveSuperimposed指令没有执行时执行该指令，该指令会报错。
- MC\_MoveSuperimposed和其它运动指令同时执行并且控制轴时，执行该指令，该指令会中断MC\_MoveSuperimposed指令，其它运动指令仍正常执行。

### ● 重启该指令

MC\_HaltSuperimposed 指令不能单独执行，MC\_MoveSuperimposed 指令执行时才可以执行该指令。当指令正在执行中，Execute 再次由 FALSE 变为 TRUE 时，对指令的执行不会产生影响，指令仍继续按照之前的方式执行。

### ● 其它指令执行时启动该指令

只有 MC\_MoveSuperimposed 指令执行时才可以执行该指令。其它指令执行时，启动该指令，该指令会报错。

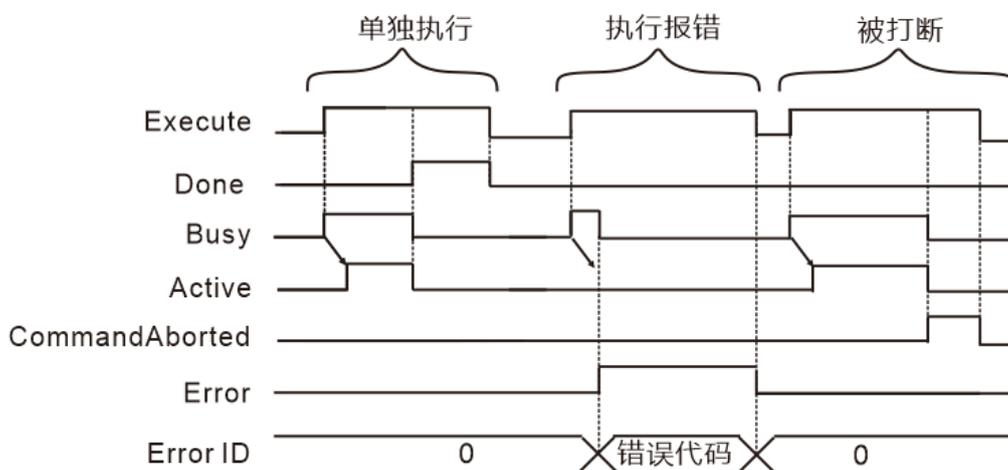
### ● 该指令执行时启动其它指令

该指令执行时启动其它指令，其它指令和该指令如何交接由其它指令的 BufferMode（缓存模式）引脚参数值决定，如果其它指令无 BufferMode 引脚，一般为中断该指令。

### ● 异常排除

该指令执行时，如果输入变量不合法，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。该指令执行时，如遇到异常（如轴报警等），该指令也会报错，轴会立即停止。

## ■ 输出变量变化时序说明



### ● 单独执行

Execute (启动) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE, 下一个周期 Active (控制中) 变为 TRUE。该指令完成时, Done (完成) 变为 TRUE, Busy (执行中) 和 Active (控制中) 同时变为 FALSE。当 Execute (启动) 变为 FALSE 时, Done (完成) 同时变为 FALSE。

### ● 执行报错

该指令输入变量值不在允许范围内时, 该指令 Execute (启动) 由 FALSE 变为 TRUE 时, 同时 Busy (执行中) 变为 TRUE, 下一个周期 Error (错误) 变为 TRUE, 同时 Busy (执行中) 变为 FALSE, ErrorID (错误码) 输出对应的错误码, 可通过 ErrorID (错误码) 的值, 查找发生问题原因。该指令 Execute (启动) 由 TRUE 变为 FALSE 时, 同时 Error (错误) 变为 FALSE, ErrorID (错误码) 的值变为 0。

### ● 被中断

该指令执行后被其它指令中断时, 该指令 CommandAborted (中断) 变为 TRUE, Busy (执行中) 和 Active (控制中) 同时变为 FALSE; 当 Execute (执行中) 变为 FALSE 时, CommandAborted (中断) 同时变为 FALSE。

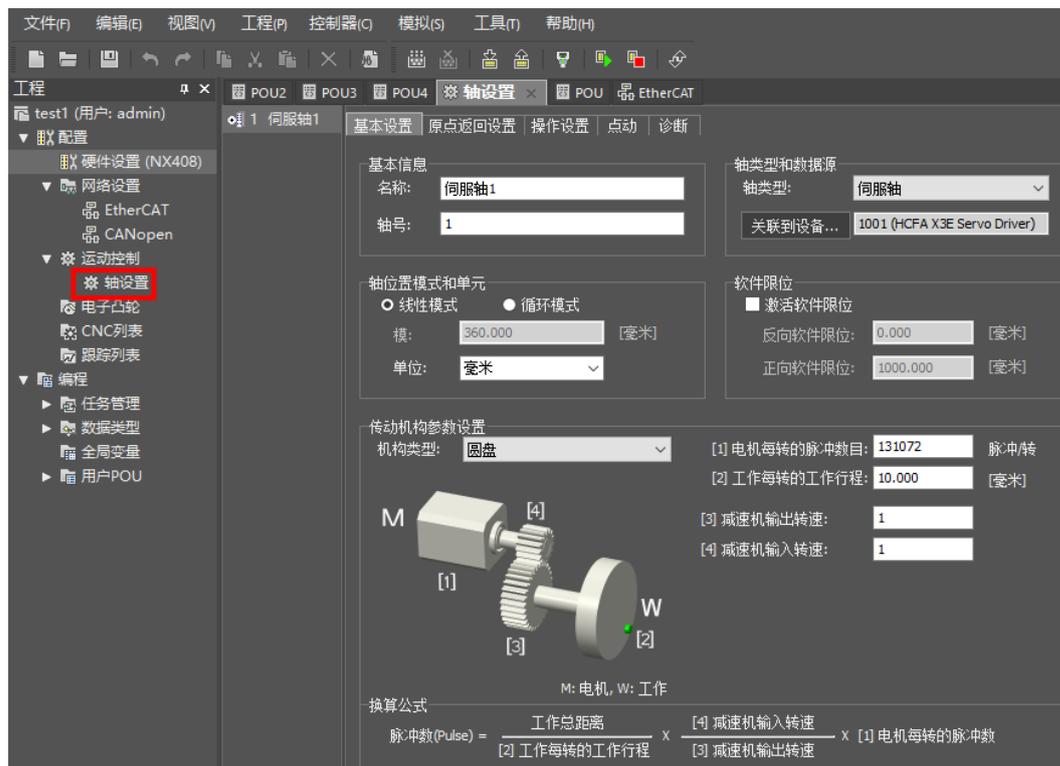
## ■ 示例程序

### ● 实现功能

当实际值和理论值不符时可以通过叠加相对位移(MC\_MoveSuperImposed)指令进行位置修正。如果启动位置修正后, 发现修正位置不合理或者不正确时, 可以通过停止叠加位移指令 (MC\_HaltSuperImposed) 停止位置的修正。

### ● 轴参数设置

轴 1 轴参数设置如下图所示。

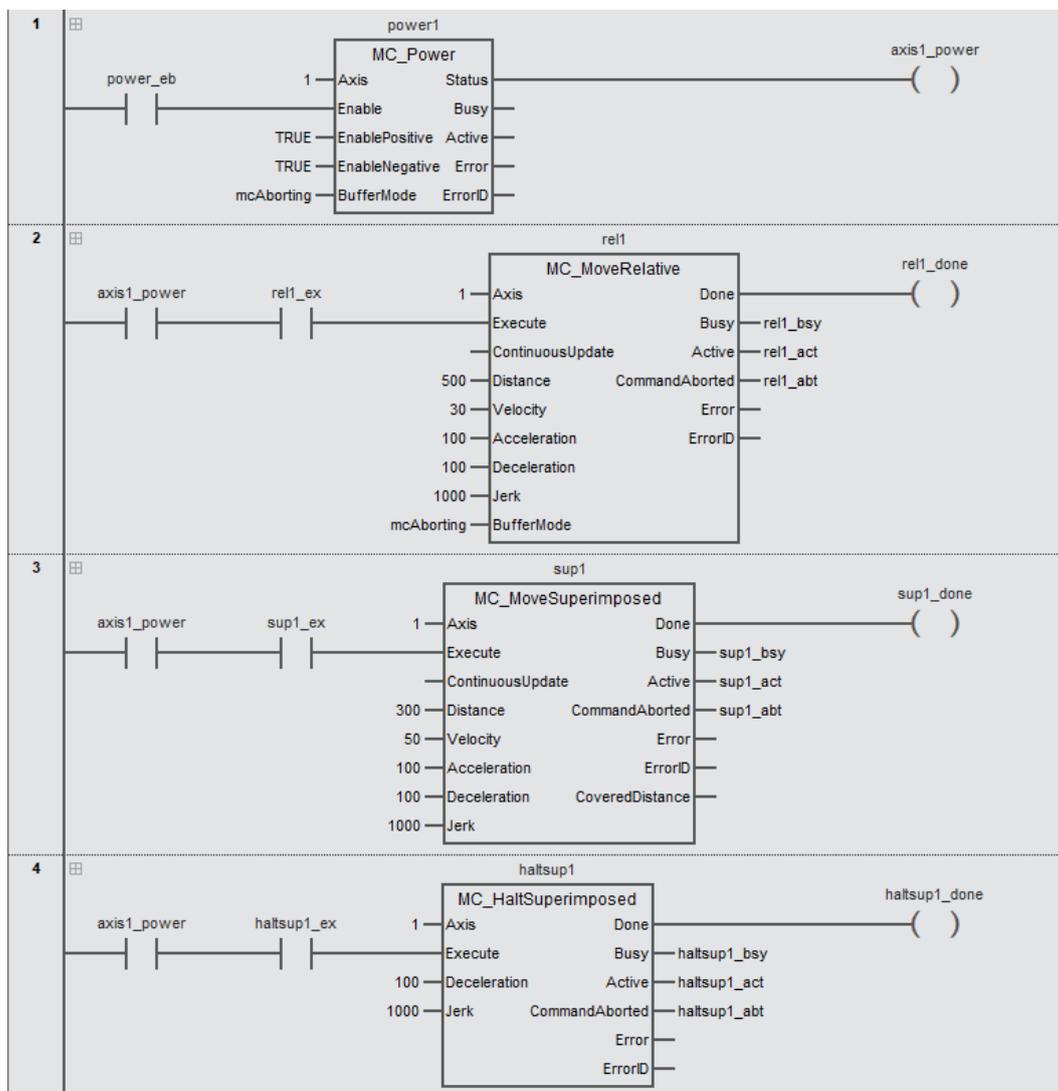


### ● 变量表

类别	名称	分配到	数据类型	初始值	注释
VAR	power_eb		BOOL		
VAR	power1		MC_Power		
VAR	axis1_power		BOOL		
VAR	rel1_ex		BOOL		

VAR	rel1		MC_MoveRelative		
VAR	rel1_done		BOOL		
VAR	rel1_bsy		BOOL		
VAR	rel1_act		BOOL		
VAR	rel1_abt		BOOL		
VAR	sup1_ex		BOOL		
VAR	sup1		MC_MoveSuperimposed		
VAR	sup1_done		BOOL		
VAR	sup1_bsy		BOOL		
VAR	sup1_act		BOOL		
VAR	sup1_abt		BOOL		
VAR	hatsup1_ex		BOOL		
VAR	hatsup1		MC_HaltSuperimposed		
VAR	hatsup1_done		BOOL		
VAR	hatsup1_bsy		BOOL		
VAR	hatsup1_act		BOOL		
VAR	hatsup1_abt		BOOL		

● 梯形图(LD)

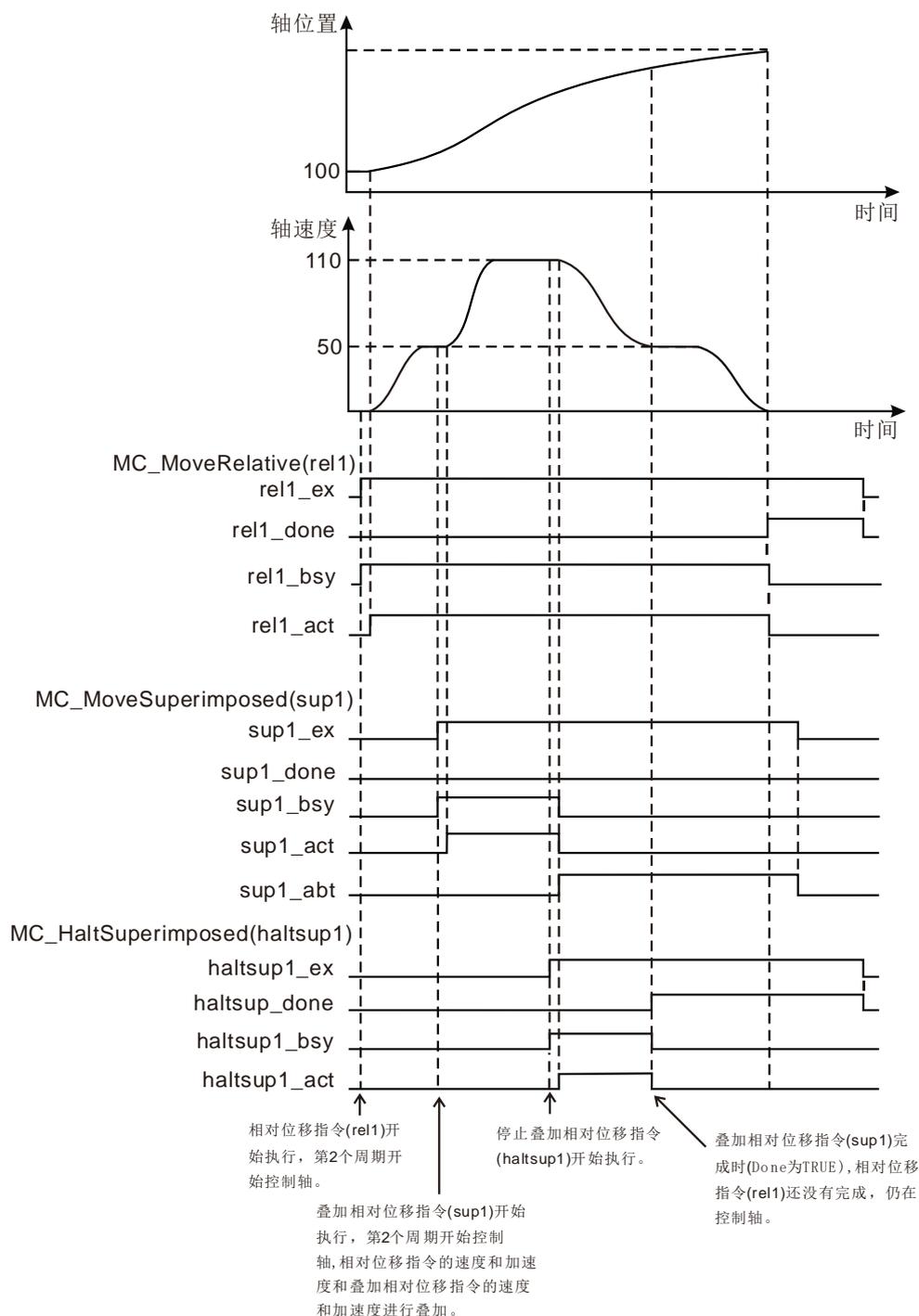


## ● 结构化文本(ST)

```
power1(  
  Axis:=1 ,  
  Enable:=power_eb ,  
  EnablePositive:=TRUE ,  
  EnableNegative:=TRUE ,  
  BufferMode:=mcAborting ,  
  Status=>axis1_power  
);  
  
rel1(  
  Axis:=1 ,  
  Execute:=axis1_power AND rel1_ex ,  
  Distance:=500 ,  
  Velocity:=50 ,  
  Acceleration:=100 ,  
  Deceleration:=100 ,  
  Jerk:=100 ,  
  BufferMode:=mcAborting ,  
  Done=>rel1_done ,  
  Busy=>rel1_bsy ,  
  Active=>rel1_act ,  
  CommandAborted=>rel1_abort  
);  
  
sup1 (  
  Axis:=1 ,  
  Execute:= axis1_power AND sup1_ex ,  
  Distance:=300 ,  
  Velocity:=60 ,  
  Acceleration:=100 ,  
  Deceleration:=100 ,  
  Jerk:=1000 ,  
  Done=> sup1_done ,  
  Busy=> sup1_bsy ,  
  Active=>sup1_act ,  
  CommandAborted=> sup1_abt  
);  
  
hatsup1 (  
  Axis:=1 ,  
  Execute:= axis1_power hatsup1_ex  
  Deceleration:=100 ,  
  Jerk:=1000 ,  
  Done=> hatsup1_done ,  
  Busy=> hatsup1_bsy ,  
  Active=> hatsup1_act ,  
  CommandAborted=> hatsup1_abt  
);
```

## ● 程序说明

- 轴使能后, rel1\_execute变为TRUE时, 相对位移指令 (rel1) 开始执行。相对位移指令 (rel1) 控制轴运转时, sup1\_execute变为TRUE时, 叠加相对位移指令 (sup1) 执行, 叠加相对位移指令控制轴后, 两个指令产生的速度、加速度、位置进行叠加后作用在一个轴上。
- 如果启动位置修正后, 发现修正位置不合理或者不正确时, 可以通过执行停止叠加位移指令 (MC\_HaltSuperImposed) 停止位置的修正。
- 如下图所示, haltsup1\_execute变为TRUE时, 停止叠加位移指令 (haltsup1) 执行, 该指令执行后, 只打断叠加相对位移指令 (sup1) 执行, 对相对位移指令的执行无影响。当停止叠加位移指令 (haltsup1) 完成时, 相对位移指令仍在执行, 轴的速度从110 (相对位移指令和叠加相对位移指令的叠加速度) 变为50 (相对位移指令的目标速度)。



## 4.17 MC\_SetOverride (速度比例设置)

该指令用于设置指定轴的全局速度比例。所属库：MotionControl

指令	名称	FB/FUN	梯形图样式	ST样式
MC_SetOverride	速度比例设置	FB		<pre>MC_SetOverride _Instance ( Axis := 参数, Enable:= 参数, VelFactor:= 参数, AccFactor:= 参数, JerkFactor:= 参数, Enabled =&gt; 参数, Busy =&gt; 参数, Error =&gt; 参数, ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Enable	有效	BOOL	TRUE或FALSE	FALSE	设为TRUE时，目标速度比例值生效；设为FALSE时，目标速度比例值变为“100%”
VelFactor	目标速度比例值	LREAL	0~500	100	目标速度比例值（单位：%）
AccFactor	加速度比例值	LREAL	正数	不可缺省	保留
JerkFactor	跃度比例值	LREAL	正数	不可缺省	保留

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Enabled	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Enabled	当 Enable 为TRUE时	<ul style="list-style-type: none"> <li>◆ 当Enable变为FALSE时</li> <li>◆ 当Error变为TRUE时</li> </ul>
Busy	当 Enable 为TRUE时	<ul style="list-style-type: none"> <li>◆ 当Enable变为FALSE时</li> <li>◆ 当Error为TRUE时</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Enable从TRUE变为FALSE时</li> </ul>

### ■ 功能说明

#### ● 基本功能说明

- 该指令和其它运动指令一起控制指定轴的目标速度比例。该指令的Enable为TRUE时，可以通过变更该指令VelFactor（目标速度比例）的值实时变更指定轴的目标速度。VelFactor的单位为百分比。如“50”表示“50%”。VelFactor参数值范围为0~500，超出范围时执行该指令，该指令会报错。

- 运动指令执行时，可以通过该指令更改目标速度比例的指令如下表所示。

MC_MoveAbsolute (绝对位移指令)	MC_MoveRelative (相对位移指令)
MC_MoveAdditive (附加位移指令)	MC_MoveVelocity (速度指令)
MC_MoveSuperimposed (叠加位移指令)	MC_JOG (点动指令)

- 该指令执行后，新的目标速度如下：

**变更后的目标速度 = 当前控制轴指令的目标速度 × 该指令速度比例值。**

- 该指令的Enable为TRUE时，更改该指令输入变量VelFactor的值，根据当前控制轴指令的输入变量进行加速或减速到目标速度。如MC\_MoveVelocity指令执行时，更改该指令输入变量VelFactor的值，则根据MC\_MoveVelocity指令输入变量Acceleration、Deceleration、Jerk的设定值进行加速或者减速到目标速度。
  - 将VelFactor设置为0时，目标速度变为0。希望保持运动指令执行，但又想暂停运动指令控制轴时，可以通过将该指令VelFactor设为0实现。此种情况下，轴状态不会发生变化，轴的状态由控制轴的运动指令决定。
  - 该指令的Enable从TRUE变为FALSE时，指定轴从当前速度变化到当前控制轴指令的目标速度，相当于该指令执行时VelFactor的值为100的情形。轴的速度变化根据当前控制轴指令的输入变量进行加速或减速到目标速度。如MC\_MoveVelocity指令执行时，该指令VelFactor的值为200并且Enable为TRUE，该指令的Enable从TRUE变为FALSE时，则根据MC\_MoveVelocity指令输入变量Acceleration、Deceleration、Jerk的设定值的进行加速或者减速到MC\_MoveVelocity指令设定的目标速度。
  - MC\_MoveVelocity指令执行并且InVelocity变为 TRUE 后，使用该指令更改VelFactor（目标速度比例）的值，轴的目标速度会变化，但MC\_MoveVelocity指令输出变量InVelocity保持为TRUE不变。
- **重启该指令**  
该指令的 Enable 为 TRUE 时持续执行。
  - **异常排除**  
该指令执行时，如果指令输入变量值不在允许的范围或者不满足指令的执行条件，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。
  - **示例程序**  
示例程序请参考 [MC\\_MoveVelocity 指令中的示例程序](#)。

## 4.18 MC\_TorqueControl (扭矩设定)

该指令用于控制轴工作在扭矩模式下，并实时更改轴的扭矩值。所属库：MotionControl

适用机种：M500S系列、M500系列

指令	名称	FB/FUN	梯形图样式	ST样式
MC_TorqueControl	扭矩设定	FB		<pre>MC_TorqueControl_Instance ( Axis := 参数, Enable := 参数, TargetTorque:= 参数, InTorque =&gt; 参数, Busy =&gt; 参数, Active=&gt; 参数, CommandAborted =&gt; 参数, Error =&gt; 参数, ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Enable	有效	BOOL	TRUE或FALSE	FALSE	设为TRUE时，控制器驱动器进入扭矩模式，将目标扭矩值传给驱动器； 设为FALSE时，退出扭矩模式，进入周期同步位置模式(CSP模式)
TargetTorque	目标扭矩	INT	负数、正数、0	0	控制器传给驱动器的目标扭矩值。此参数单位为 [0.1%]，以额定转矩“100.0%”时的比率进行指定。如目标扭矩设定为30时，表示设定目标扭矩为额定扭矩的百分之三。Enable设为TRUE时，该参数值实时传给驱动器

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
InTorque	命令扭矩到达	BOOL	TRUE或FALSE	控制器给驱动器的命令扭矩到达目标扭矩时为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Active	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
InTorque	控制器传给轴的命令扭矩到达设定的目标扭矩时	<ul style="list-style-type: none"> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>
Busy	Enable为TRUE时	<ul style="list-style-type: none"> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>
Active	指令开始控制轴时	<ul style="list-style-type: none"> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>

CommandAborted	指令的执行被中断时	◆ CommandAborted为TRUE, Enable由TRUE变为FALSE时
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	◆ Enable由TRUE变为FALSE时

## ■ 功能说明

### ● 基本功能说明

- 该指令Enable为TURE时，控制轴工作在扭矩模式下，控制器将设定的目标扭矩（TargetTorque）通过EtherCAT总线实时传给指定轴，控制器不参与运算。如果需要指定轴的扭矩值缓慢变化，建议使用MC\_TorqueControlWithVelocity指令或者编写程序控制目标扭矩（TargetTorque）值的变化过程，或者设置指定轴的扭矩变化率参数实现。
- 该指令Enable为FALSE时，轴立即退出扭矩模式，进入周期同步位置模式(CSP模式)。

### ● 数据对象的映射（PDO 映射）

M500S 系列控制器不需要用户映射数据对象。M500 系列控制器需要用户映射数据对象，映射对象如下表所示。

接收PDO（主站=>从站） （16进制）	映射数据含义	发送PDO（从站=>主站）	映射数据含义
6040_0（索引_子索引）	控制字	6041_0（索引_子索引）	状态字
6060_0（索引_子索引）	控制模式	6061_0（索引_子索引）	反馈模式
6071_0（索引_子索引）	设定扭矩		

### ● 其它指令执行时启动该指令

其它运动指令执行时启动该指令，该指令会报错。

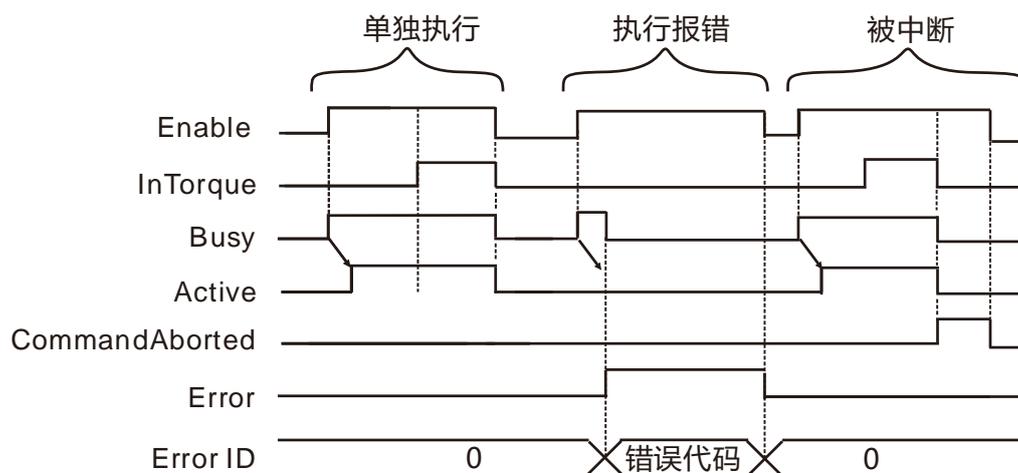
### ● 该指令执行时启动其它指令

该指令执行时，可以执行MC\_Stop指令中断该指令，执行其它指令，其它指令会报错。

### ● 异常排除

该指令执行时，如果输入变量不合法，该指令Error变为TRUE，ErrorID中有对应的错误码，可以通过该错误码判断错误原因。

## ■ 输出变量变化时序说明



- **单独执行**

Enable（有效）由 FALSE 变为 TRUE 时，Busy（执行中）同时变为 TRUE，下一个周期 Active（控制中）变为 TRUE。控制器给伺服驱动器的命令扭矩到达后，InTorque（命令扭矩到达）变为 TRUE，Busy（执行中）和 Active（控制中）仍为 TRUE。

- **执行报错**

该指令输入变量值不在允许范围内时，该指令 Enable（有效）由 FALSE 变为 TRUE 时，同时 Busy（执行中）变为 TRUE，下一个周期 Error（错误）变为 TRUE，同时 Busy（执行中）变为 FALSE，ErrorID（错误码）输出对应的错误码，可通过 ErrorID（错误码）的值，查找发生问题原因。该指令 Enable（有效）由 TRUE 变为 FALSE 时，同时 Error（错误）变为 FALSE，ErrorID（错误码）的值变为 0。

- **被中断**

该指令执行后被其它指令中断时，该指令 CommandAborted（中断）变为 TRUE，Busy（执行中）和 Active（控制中）同时变为 FALSE；当 Enable（有效）变为 FALSE 时，CommandAborted（中断）同时变为 FALSE。

## 4.19 MC\_TorqueControlWithVelocity (带速度限制的扭矩控制)

该指令用于控制轴工作在扭矩模式下，实时更改轴的扭矩值并限制轴的速度。所属库：MotionControl

适用机种： M500S系列、M500系列

指令	名称	FB/FUN	梯形图样式	ST样式
MC_TorqueControlWithVelocity	带速度限制的扭矩控制	FB		<pre> MC_TorqueControlWithVelocity_Instance ance ( Axis := 参数, Enable := 参数 , ContinuousUpdate := 参数 , Torque:= 参数, TorqueRamp:= 参数, Mode:= 参数, Velocity:= 参数, Acceleration:= 参数, Deceleration:= 参数, Jerk:= 参数, Direction:= 参数, BufferMode:= 参数 , InTorque=&gt; 参数 , Busy =&gt; 参数 , Active=&gt; 参数 , CommandAborted =&gt; 参数 , Error =&gt; 参数 , ErrorID=&gt; 参数 );                     </pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Enable	有效	BOOL	TRUE或FALSE	FALSE	设为TRUE时，控制器驱动器进入扭矩模式，将目标扭矩值传给驱动器。 设为FALSE时，退出扭矩模式，进入CSP模式
ContinuousUpdate	持续更新	BOOL	TRUE或FALSE	FALSE	保留
Torque	目标扭矩	INT	负数、正数、0	0	控制器传给驱动器的目标扭矩值。此参数单位为 [0.1%]，以额定转矩“100.0%”时的比率进行指定。如目标扭矩设定为30时，表示设定目标扭矩为额定扭矩的百分之三。 Enable设为TRUE时，该参数值实时传给驱动器
TorqueRamp	扭矩变化率	LREAL	负数、正数	不可缺省	从当前设定扭矩到更改后设定扭矩的变化率（单位：千分比/秒）
Mode	限制速度参数选择	INT	0、1	0	0：Velocity（限制速度）和伺服内部参数16#6080关联； 1：Velocity（限制速度）和伺服内部参数16#607F关联；
Velocity	限制速度	LREAL	正数	不可缺省	扭矩指令控制轴时，限制轴的速度不能超过该设定值。 Mode参数的值不同时，该参数关联的伺服的参数不同，无论选择哪个伺服参数，该参数的（单位：行程单位/秒）
Acceleration	加速度	LREAL	正数	保留	保留
Deceleration	减速度	LREAL	正数	保留	保留

Jerk	跃度	LREAL	正数	保留	保留
Direction	方向	MC_Direction	1: mcPositiveDirection 3: mcNegativeDirection	1	设定轴的运转方向 1:正方向 3:反方向
BufferMode	缓存模式	MC_Buffer_Mode	0: mcAborting	保留	保留

## ■ 输出变量

名称	名称	数据类型	输出范围	功能
InTorque	命令扭矩到达	BOOL	TRUE或FALSE	控制器给驱动器的命令扭矩到达目标扭矩时为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Active	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码 值的含义请参考“ <a href="#">指令错误代码描述</a> ”

## ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
InTorque	控制器传给轴的命令扭矩到达设定的目标扭矩时	<ul style="list-style-type: none"> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>
Busy	Enable为TRUE时	<ul style="list-style-type: none"> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>
Active	指令开始控制轴时	<ul style="list-style-type: none"> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>
CommandAborted	指令的执行被中断时	<ul style="list-style-type: none"> <li>◆ CommandAborted为TRUE，Enable由TRUE变为FALSE时</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Enable由TRUE变为FALSE时</li> </ul>

## ■ 功能说明

### ● 基本功能说明

- 该指令Enable为TURE时，控制指定轴工作在扭矩模式下，并将扭矩模式运行的限制速度（Velocity）传给指定轴。控制器以设定的目标扭矩（Torque）为目标，控制器根据设定的扭矩变化率（TorqueRamp）变化扭矩值，并将运算的扭矩值实时传给指定轴。控制器给指定轴的扭矩到达目标扭矩后，保持目标扭矩输出。目标扭矩（Torque）值更改后，控制器再以新更改的扭矩值为目标，按照扭矩变化率（TorqueRamp）变化扭矩值并实时传给指定轴。
- 扭矩变化率（TorqueRamp）的单位为千分比/秒，目标扭矩（Torque）为额定扭矩的千分比。从当前扭矩变化到目标扭矩需要时间的计算方法如下：

$$\text{当前扭矩到达目标扭矩的时间} = (\text{目标扭矩} - \text{当前给定扭矩}) / \text{扭矩变化率 (TorqueRamp)}$$

- 该指令Enable为FALSE时，轴立即退出扭矩模式，进入CSP模式，限制速度（Velocity）变为Enable从FALSE变为TRUE前的值。
- ### ● 数据对象的映射（PDO 映射）

M500S 系列控制器不需要用户映射数据对象。M500 系列控制器需要用户映射数据对象，映射对象如下表所示。

接收PDO（主站=>从站） （16进制）	映射数据含义	发送PDO（从站=>主站）	映射数据含义
-------------------------	--------	---------------	--------

6040_0 (索引_子索引)	控制字	6041_0 (索引_子索引)	状态字
6060_0 (索引_子索引)	控制模式	6061_0 (索引_子索引)	反馈模式
6071_0 (索引_子索引)	设定扭矩		
607F_0 (索引_子索引) 或者 6080_0, 根据该指令Mode值配置对应的参数。	限制速度		

- **其它指令执行时启动该指令**

其它运动指令执行时启动该指令，该指令会报错。

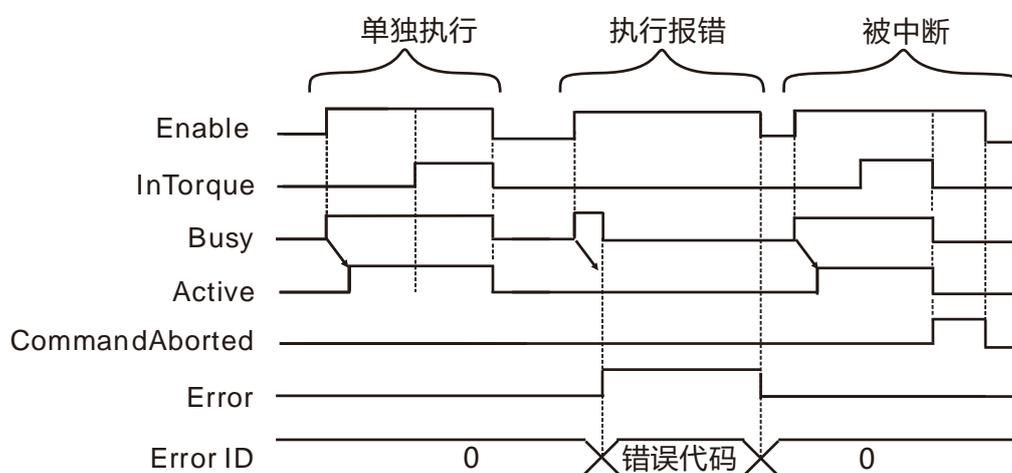
- **该指令执行时启动其它指令**

该指令执行时，可以执行 MC\_Stop 指令中断该指令，执行其它指令，其它指令会报错。

- **异常排除**

该指令执行时，如果输入变量不合法，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。

- **输出变量变化时序说明**



- **单独执行**

Enable (有效) 由 FALSE 变为 TRUE 时，Busy (执行中) 同时变为 TRUE，下一个周期 Active (控制中) 变为 TRUE。控制器给伺服驱动器的命令扭矩到达后，InTorque (命令扭矩到达) 变为 TRUE，Busy (执行中) 和 Active (控制中) 仍为 TRUE。

- **执行报错**

该指令输入变量值不在允许范围内时，该指令 Enable (有效) 由 FALSE 变为 TRUE 时，同时 Busy (执行中) 变为 TRUE，下一个周期 Error (错误) 变为 TRUE，同时 Busy (执行中) 变为 FALSE，ErrorID (错误码) 输出对应的错误码，可通过 ErrorID (错误码) 的值，查找发生问题原因。该指令 Enable (有效) 由 TRUE 变为 FALSE 时，同时 Error (错误) 变为 FALSE，ErrorID (错误码) 的值变为 0。

- **被中断**

该指令执行后被其它指令中断时，该指令 CommandAborted (中断) 变为 TRUE，Busy (执行中) 和 Active (控制中) 同时变为 FALSE；当 Enable (有效) 变为 FALSE 时，CommandAborted (中断) 同时变为 FALSE。

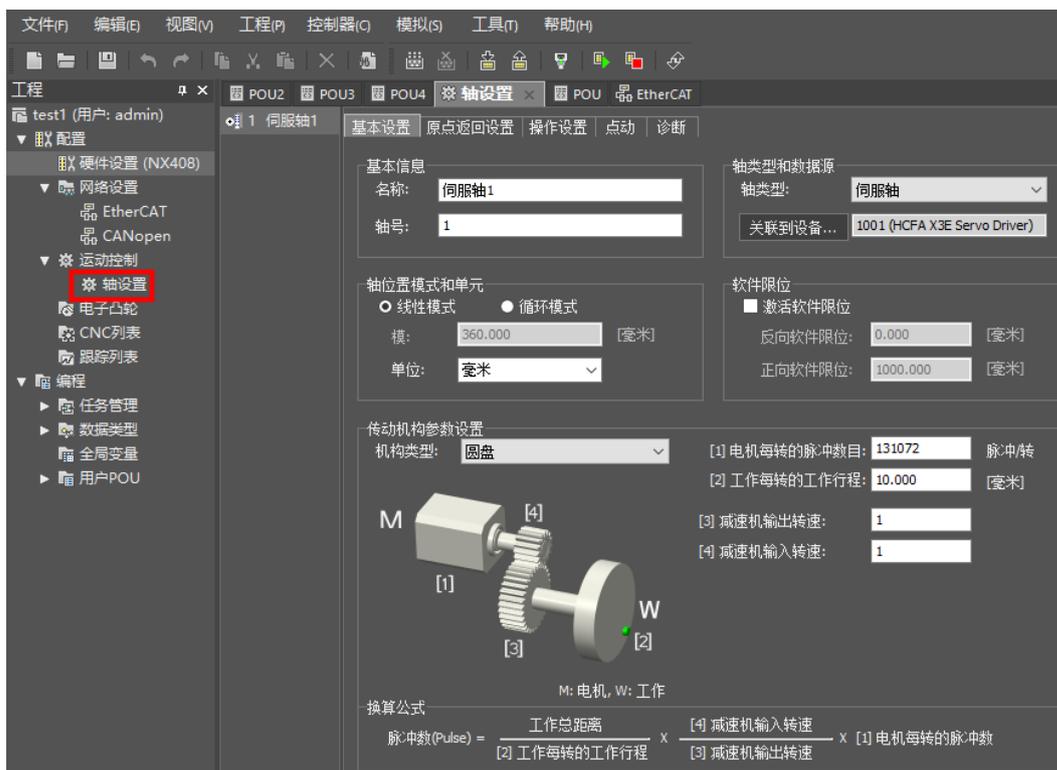
- **示例程序**

- **实现功能**

控制指定轴工作在扭矩模式下，按照设定的扭矩变化率到达目标扭矩，并限制轴的最大速度不超过设定的限制速度。

- **轴参数设置**

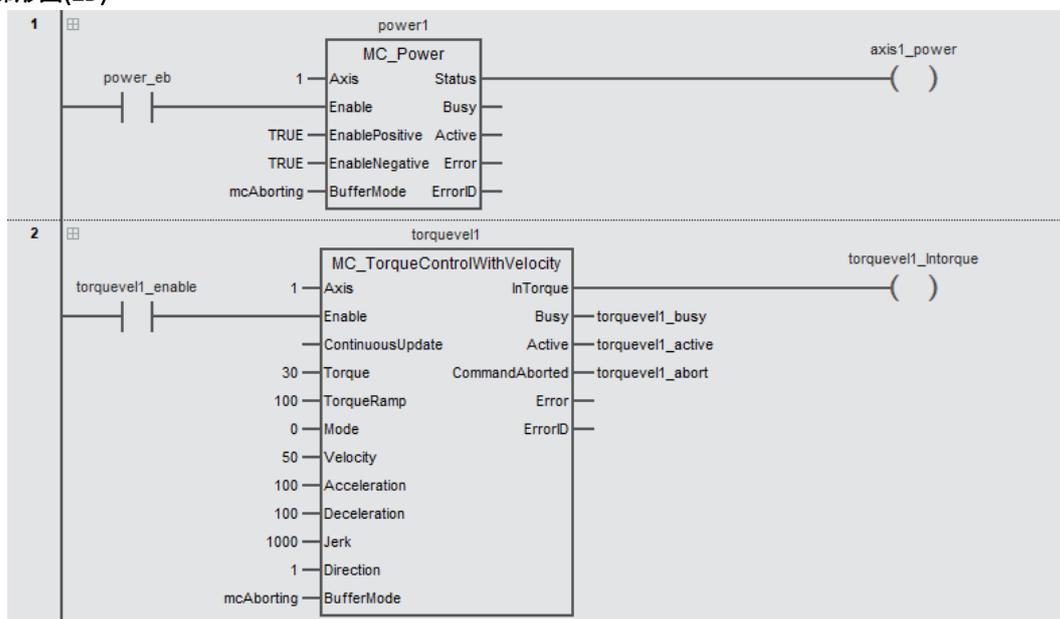
轴 1 轴参数设置如下图所示



● 变量表

类别	名称	分配到	数据类型	初始值	注释
VAR	power_eb		BOOL		
VAR	power1		MC_Power		
VAR	axis1_power		BOOL		
VAR	torquevel1		MC_TorqueControlWithVelocity		
VAR	torquevel1_enable		BOOL		
VAR	torquevel1_intorque		BOOL		
VAR	torquevel1_busy		BOOL		
VAR	torquevel1_active		BOOL		
VAR	torquevel1_abort		BOOL		

● 梯形图(LD)

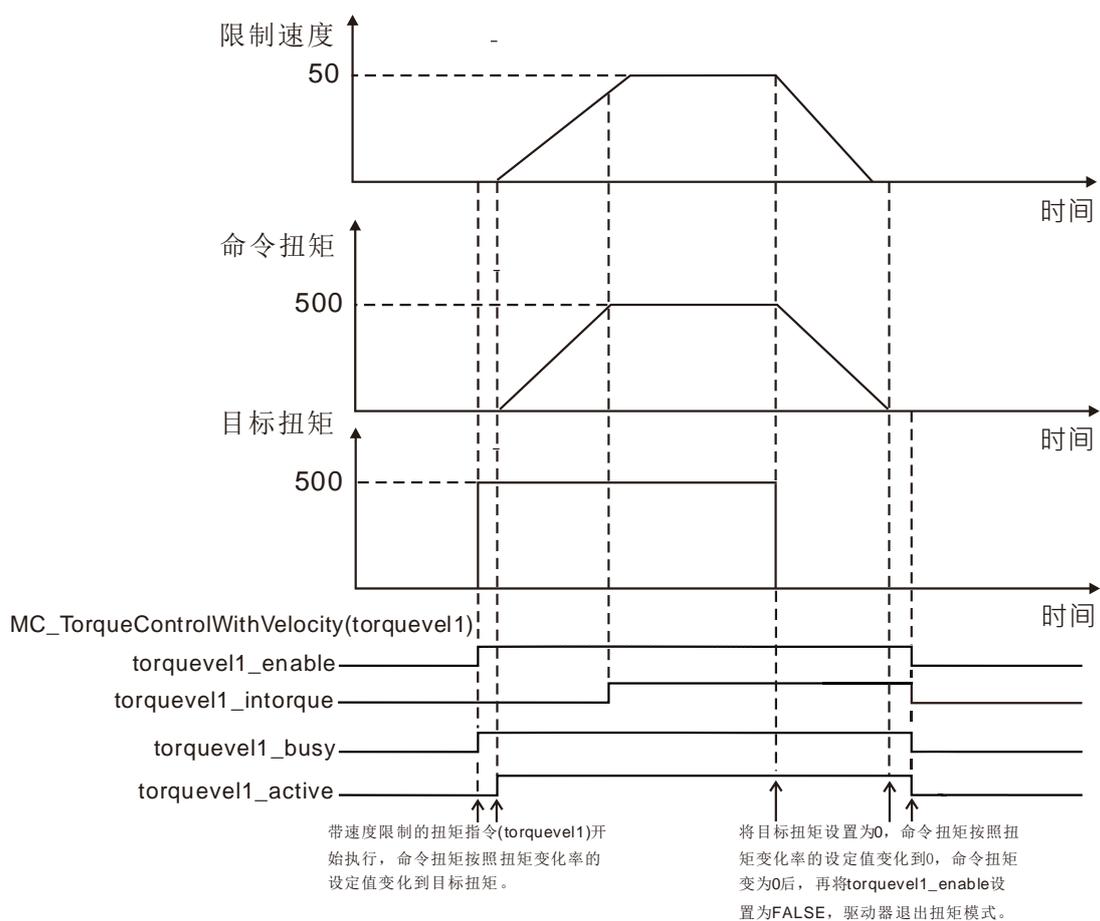


- 结构化文本(ST)

```
power1(  
    Axis:=1,  
    Enable:=power_eb,  
    EnablePositive:=TRUE,  
    EnableNegative:=TRUE,  
    BufferMode:=mcAborting,  
    Status=>axis1_power  
);  
  
torquevel1(  
    Axis:=1,  
    Enable:=torquevel1_enable,  
    Torque:=30,  
    TorqueRamp:=100,  
    Mode:=0,  
    Velocity:=50,  
    Acceleration:=100,  
    Deceleration:=100,  
    Jerk:=1000,  
    Direction:=1,  
    BufferMode:=mcAborting,  
    InTorque=>torquevel1_intorque,  
    Busy=>torquevel1_busy,  
    Active=>torquevel1_active,  
    CommandAborted=>torquevel1_abort,  
);
```

- 程序说明

- 轴使能后，torquevel1\_enable变为TRUE时，带速度限制的扭矩控制指令（torquevel1）开始执行。该指令控制轴1工作在扭矩模式下，按照设定的扭矩变化率到达目标扭矩，并限制轴的最大速度不超过设定的限制速度。控制器传给轴1的命令扭矩变化过程如下图所示，命令扭矩到达设定扭矩后，torquevel1\_intorque变为TRUE并保持目标扭矩输出。
- 将目标扭矩设置为0后，命令扭矩按照设定的扭矩变化率变化到0，变化过程如下图所示。
- 命令扭矩变为0后，torquevel1\_enable设置为FALSE时，torquevel1\_intorque、torquevel1\_busy、torquevel1\_active同时变为FALSE，轴1退出扭矩模式，进入CSP模式，限制速度（Velocity）变为Enable从FALSE变为TRUE前的值。



## 4.20 MC\_SyncMoveAbsolute (周期增量位置控制)

按每个周期的增量位置控制轴。所属库: MotionControl

指令	名称	FB/FUN	梯形图样式	ST样式
MC_SyncMoveAbsolute	周期增量位置控制	FB		<pre>MC_SyncMoveAbsolute_Instance ( Axis:= 参数 , Execute:= 参数 , ContinuousUpdate:= 参数 , Position:= 参数 , Velocity:= 参数 , Acceleration:= 参数 , Deceleration:= 参数 , Jerk:= 参数 , InControl=&gt; 参数 , Busy=&gt; 参数 , Active=&gt; 参数 , CommandAborted=&gt; 参数 , Error=&gt; 参数 , ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
ContinuousUpdate	持续更新	BOOL	TRUE或FALSE	FALSE	保留
Position	绝对位置	LREAL	正数、负数、0	0	指定绝对位置，一个周期内从当前位置移动至设定位置 (单位: 行程单位) *1
Velocity	目标速度	LREAL	正数	不可缺省	保留
Acceleration	加速度	LREAL	正数	不可缺省	保留
Deceleration	减速度	LREAL	正数	不可缺省	保留
Jerk	跃度	LREAL	正数	不可缺省	保留

\*1: Velocity、Acceleration、Deceleration、Jerk的关系请参考“[运动控制指令相关参数说明](#)”章节。

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
InControl	完成	BOOL	TRUE或FALSE	指令执行完成时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Active	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
InControl	指令控制轴时	<ul style="list-style-type: none"> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>

Busy	Execute的上升沿	<ul style="list-style-type: none"> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
Active	指令开始控制轴时	<ul style="list-style-type: none"> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
CommandAborted	指令的执行被其它指令中断时	<ul style="list-style-type: none"> <li>◆ CommandAborted 为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令被其它指令中断时, CommandAborted变为TRUE, 一个周期后, 变为FALSE</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令执行过程中遇到异常时, Error变为TRUE, 一个周期后, 变为FALSE</li> </ul>

## ■ 功能说明

### ● 基本功能说明

- 该指令用于将每个周期输入变量Position的增量值加上上个周期的轴命令位置传给指定轴，周期的增量值在输出变量active变为TRUE时开始计算，即第一次的增量值以Execute变为TRUE时输入变量Position的值为基准。建议Execute变为TRUE时，将轴反馈位置搬移到输入变量Position内。
- 该指令一般用于用户自己规划运动轨迹，轴的控制由用户通过控制输入变量Position的值进行。
- 下表所示为通过该指令控制轴时的两种情形，可以通过这两种情形更好的理解该指令的控制方法。
- Execute由FALSE变为TRUE时，该指令执行，即使Execute再变为FALSE，该指令仍可以执行。如果要终止该指令的执行，可以执行MC\_Halt或者MC\_Stop指令打断该指令，终止该指令的执行。

情形1:

执行周期	Busy变为TRUE周期	Active变为TRUE第1周期	Active变为TRUE第2周期
Position的值	500	530	580
轴命令位置	500	530	580
轴反馈位置	500	530	580

情形2:

执行周期	Busy变为TRUE周期	Active变为TRUE第1周期	Active变为TRUE第2周期
Position的值	500	530	580
轴命令位置	300	330	380
轴反馈位置	300	330	380

### ● 指令完成时机

Execute 由 FALSE 变为 TRUE 后，该指令持续执行，将 Position 值每个任务周期的变化量传给轴。

### ● 重启该指令

Execute 由 FALSE 变为 TRUE 后，该指令持续执行。该指令执行后，Execute 再由 FALSE 变为 TRUE，对该指令的执行无影响。如果要重新执行该指令，可以执行 MC\_Halt 或者 MC\_Stop 指令打断该指令，然后再次执行该指令进行重启该指令。

### ● 其它指令执行时启动该指令

其它运动指令执行时，启动该指令，该指令会报错。

### ● 该指令执行时启动其它指令

该指令执行时启动其它运动指令，其它运动指令会打断该指令。

### ● 异常排除

该指令执行时，如果指令输入变量值不在允许的范围或者不满足指令的执行条件，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。该指令执行时，如遇到异常（如轴报警等），该指令也会报错，轴会立即停止。

## 4.21 MC\_SetPosition (位置设置)

将指定轴的命令位置或者实际位置变更为设定值。所属库: MotionControl

指令	名称	FB/FUN	梯形图样式	ST样式
MC_SetPosition	位置设置	FB		<pre>MC_SetPosition_Instance ( Axis := 参数, Execute := 参数, Distance:= 参数, Relative:= 参数, Reference Type:= 参数, ExecutionMode:= 参数, Done =&gt; 参数, Busy =&gt; 参数, Error =&gt; 参数, ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
Position	设定位置	LREAL	正数、负数、0	0	将指定轴的位置设定为Position指定的位置 (单位: 行程单位)
Relative	模式设置	BOOL	TRUE或FALSE	FALSE	设定位置为相对模式或绝对模式选择 TRUE:相对模式 FALSE:绝对模式
ReferenceType	位置类型选择	MC_Refe renceType	0: mcCommandPosition 1: mcActualPosition	0	设定该指令对指定轴的命令位置或者反馈位置 进行操作 0:命令位置 1:反馈位置
ExecutionMode	执行模式	保留			保留 (该输入引脚可以不填写变量或者常数)

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时, 输出错误代码 值的含义请参考 <a href="#">“指令错误代码描述”</a>

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	执行完成时	<ul style="list-style-type: none"> <li>◆ Done为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE</li> </ul>
Busy	Execute的上升沿	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> </ul>
Error	指令输入变量值不在允许的范围	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Execute从TRUE变为FALSE时</li> </ul>

## ■ 功能说明

### ● 基本功能说明

- 该指令用于将指定轴的命令位置或者反馈位置更改为该指令的设定值。该指令的执行不会让轴产生移动，也不会让指定轴的物理位置发生变化，一般用于重新定义位置坐标或者坐标偏移。该指令为控制器内部运算实现，对伺服轴操作时不会改变驱动器的位置。
- 伺服轴、虚拟伺服轴、脉冲轴、编码器轴都可以执行该指令。因为虚拟伺服轴和编码器轴的命令位置和反馈位置始终相同，选择命令位置或者反馈位置操作时无差别。
- 该指令执行后，保持轴命令位置和反馈位置的差值不变进行变更，以该指令无论指定对轴的命令位置还是反馈位置进行操作，轴的命令位置和反馈位置都会跟随变化。绝对模式时，该指令指定对命令位置或反馈位置操作时，轴位置计算公式如下：

**该指令指定对命令位置操作时，轴的反馈位置计算公式如下：**

*轴的反馈位置 = 该指令设定的位置 (Position) - 该指令触发执行时刻轴命令位置和反馈位置的差值。*

**该指令指定对反馈位置操作时，轴的命令位置计算公式如下：**

*轴的命令位置 = 该指令设定的位置 (Position) + 该指令触发执行时刻轴命令位置和反馈位置的差值。*

- 轴静止和移动时均可以执行该指令。

### ● 设定位置、模式选择、位置类型选择

- 该指令执行后，轴的命令位置及反馈位置会根据该指令的输入变量进行变化。输入变量Position表示设定的位置；ReferenceType用来选择对轴命令位置或反馈位置进行操作，设定为0表示对轴命令位置操作，设定为1表示对轴反馈位置操作；Relative用于设置Position和轴的位置为相对还是绝对关系，Relative为FALSE表示Position和轴位置为绝对关系，将轴位置设定为Position的值，Relative为TRUE表示Position和轴位置为相对关系，将轴位置加上Position的值变为轴位置。
- 轴静止时，一般轴命令位置和反馈位置相同，执行该指令后，轴命令位置和反馈位置仍相同。轴运动时，一般轴命令位置和反馈位置有差值，假设在轴命令位置和反馈位置差值为200时执行该指令，该指令根据输入变量不同的设定值，得到的命令位置和反馈位置如下表所示。

相关参数	相关参数值			
	FALSE: 绝对模式	FALSE: 绝对模式	TRUE: 相对模式	TRUE: 相对模式3
Relative (模式设置)	FALSE: 绝对模式	FALSE: 绝对模式	TRUE: 相对模式	TRUE: 相对模式3
ReferenceTyp (位置类型选择)	1: 反馈位置	0: 命令位置	1: 反馈位置	0: 命令位置
Position (设定位置)	1000	1000	1000	1000
执行前命令位置	2000	2000	2000	2000
执行前反馈位置	1800	1800	1800	1800
执行前位置差	200	200	200	200
执行后命令位置	1200	1000	3000	3000
执行后反馈位置	1000	800	2800	2800
执行后位置差	200	200	200	200

### ● 指令适用情形说明

- 对已建立多轴关系的主轴执行MC\_SetPosition时，由MC\_SetPosition引起的主轴位置变化不会影响从轴，也即是主轴的位置在MC\_SetPosition的作用下发生了变化，但从轴不会因为该变化而动作。当作用于从轴时，从轴的位置发生变化，但其与主轴原有的同步关系不受影响。
- 若在MC\_Stop指令执行过程中（未执行完成）执行MC\_SetPosition指令，MC\_SetPosition指令将报错；若

在MC\_Stop指令执行完成后再执行MC\_SetPosition指令，则可正常执行。

- **重启该指令**

当指令执行完成后，Execute 再次由 FALSE 变为 TRUE 时，该指令可以重新执行；当指令正在执行中，Execute 再次由 FALSE 变为 TRUE 时，对指令的执行不会产生影响，指令仍继续按照之前的方式执行。

- **其它指令执行时启动该指令**

其它令执行时启动该指令，该指令不会影响其它指令的运动。但是该指令及其它运动指令执行完后，因为轴的命令位置和反馈位置会发生变化，要确保对该指令完全了解，再执行其它的运动指令。

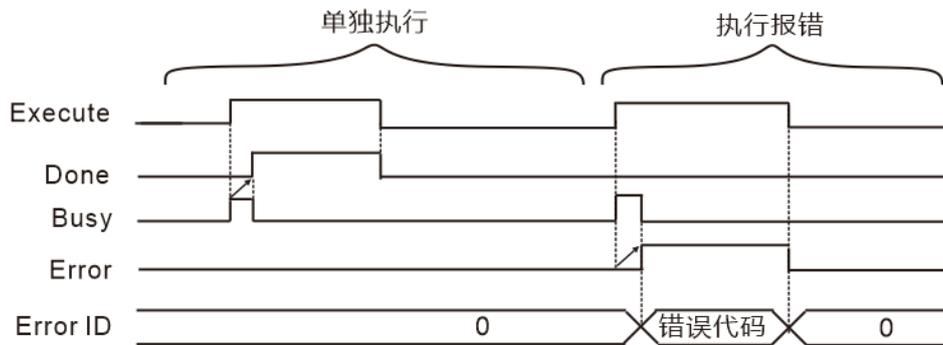
- **该指令执行时启动其它指令**

该指令执行时可以执行其它运动指令。因为执行该指令轴的命令位置和反馈位置会发生变化，要确保对该指令完全了解，再执行其它的运动指令。

- **异常排除**

该指令执行时，如果输入变量不合法，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。

### ■ 输出变量变化时序说明



- **单独执行**

Execute (启动) 由 FALSE 变为 TRUE 时，Busy (执行中) 同时变为 TRUE，下一个周期 Done (完成) 变为 TRUE，同时 Busy (执行中) 变为 FALSE。当 Execute (启动) 变为 FALSE 时，Done (完成) 同时变为 FALSE。

- **执行报错**

该指令输入变量值不在允许范围内时，该指令 Execute (启动) 由 FALSE 变为 TRUE 时，同时 Busy (执行中) 变为 TRUE，下一个周期 Error (错误) 变为 TRUE，同时 Busy (执行中) 变为 FALSE，ErrorID (错误码) 输出对应的错误码，可通过 ErrorID (错误码) 的值，查找发生问题原因。该指令 Execute (启动) 由 TRUE 变为 FALSE 时，同时 Error (错误) 变为 FALSE，ErrorID (错误码) 的值变为 0。

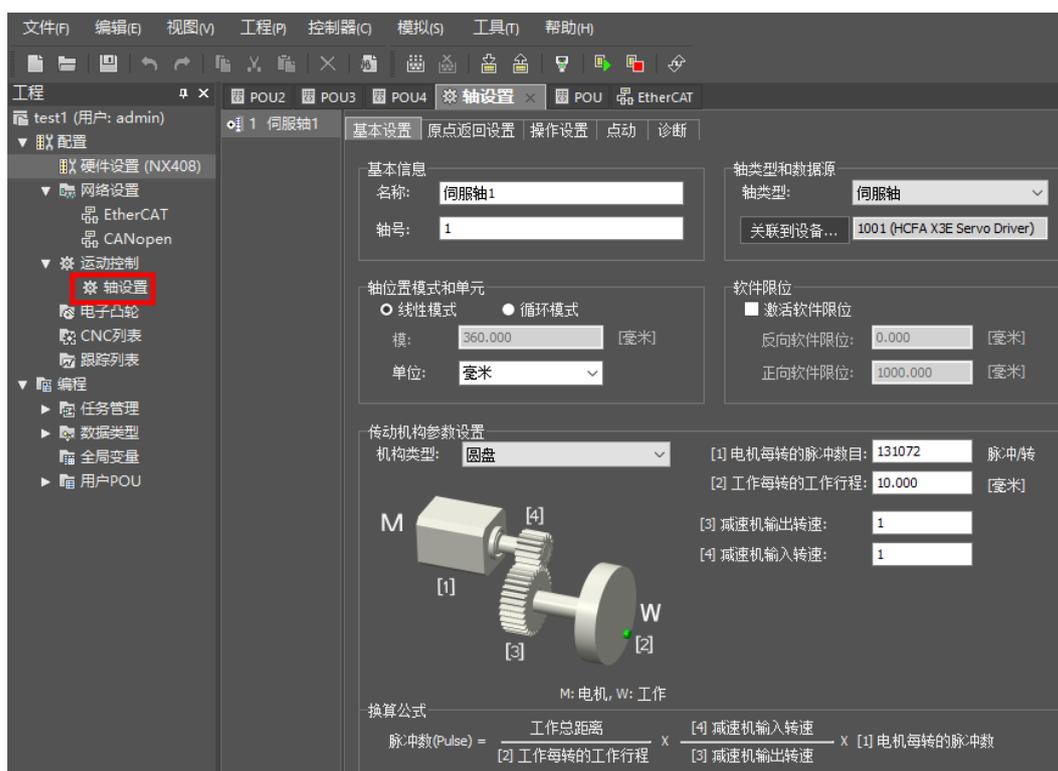
### ■ 示例程序

- **实现功能**

通过位置设置指令将轴当前位置设置为用户设定的位置，设置位置后，以设置的位置为参考点，再走指定的相对位置。

- **轴参数设置**

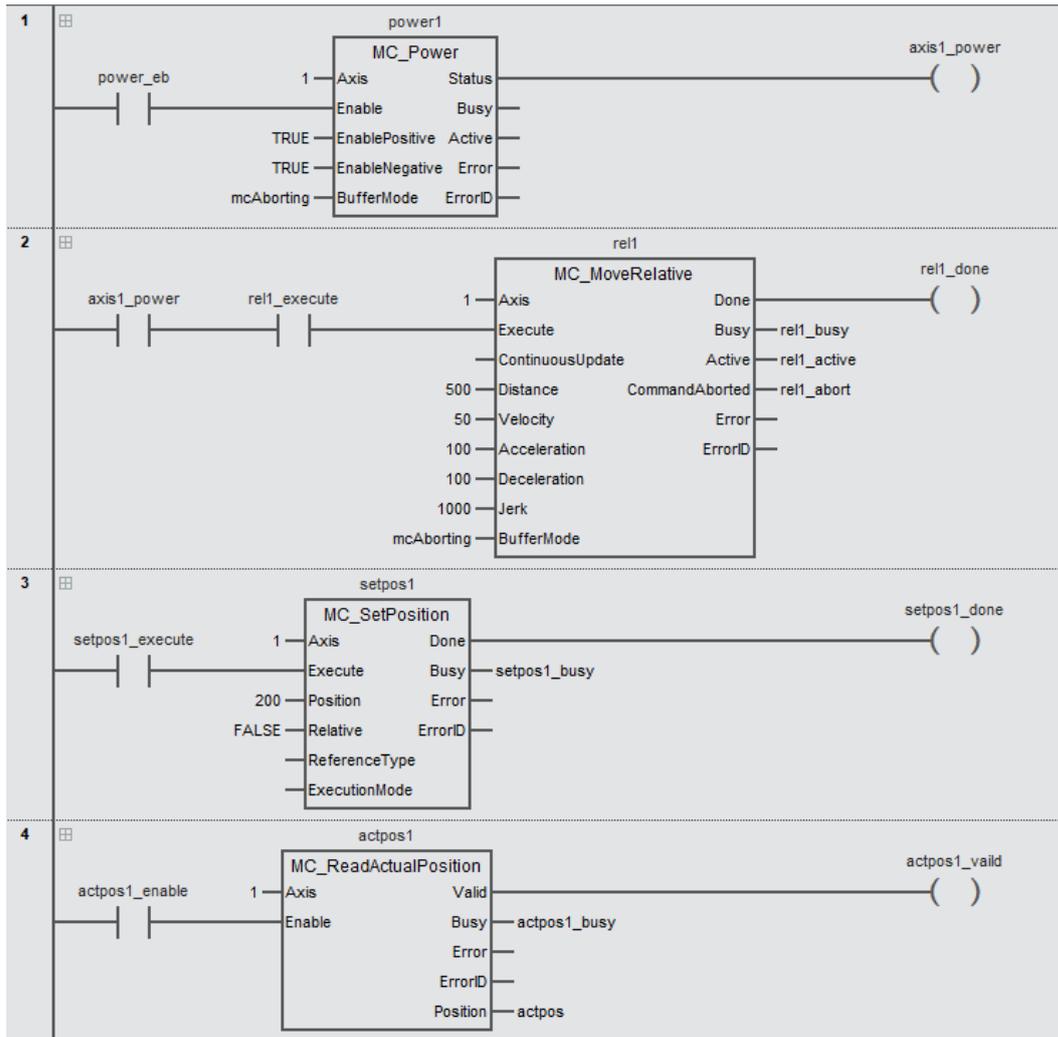
轴 1 轴参数设置如下图所示。



● 变量表

类别	名称	分配到	数据类型	初始值	注释
VAR	power_eb		BOOL		
VAR	power1		MC_Power		
VAR	axis1_power		BOOL		
VAR	rel1_execute		BOOL		
VAR	rel1		MC_MoveRelative		
VAR	rel1_done		BOOL		
VAR	rel1_busy		BOOL		
VAR	rel1_active		BOOL		
VAR	rel1_abort		BOOL		
VAR	setpos1_execute		BOOL		
VAR	setpos1		MC_SetPosition		
VAR	setpos1_done		BOOL		
VAR	setpos1_busy		BOOL		
VAR	actpos1		MC_ReadActualPosition		
VAR	actpos1_enable		BOOL		
VAR	actpos1_vaild		BOOL		
VAR	actpos1_busy		BOOL		
VAR	actpos		LREAL		

● 梯形图(LD)



● 结构化文本(ST)

```

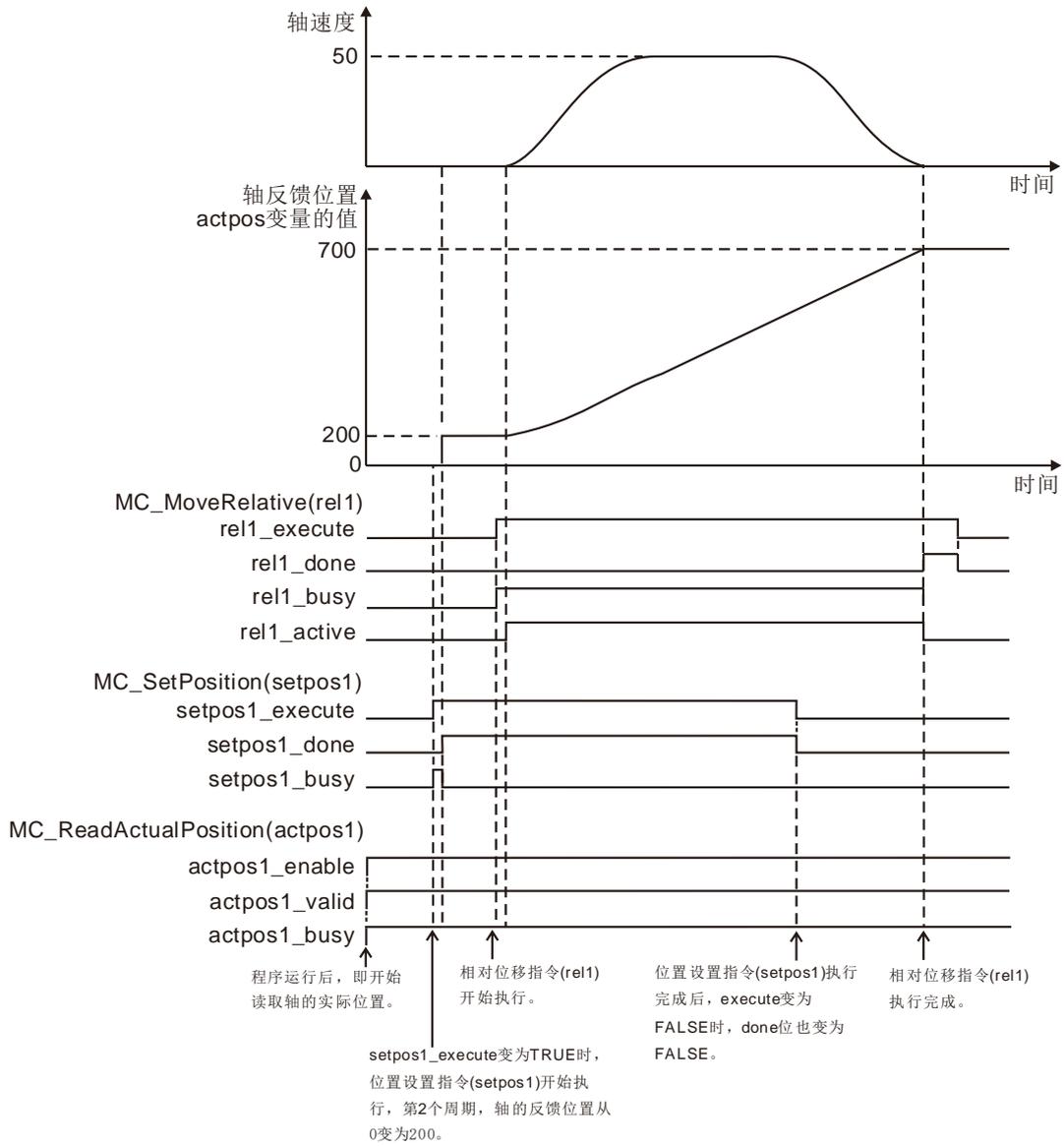
power1(
  Axis:=1,
  Enable:=power_eb,
  EnablePositive:=TRUE,
  EnableNegative:=TRUE,
  BufferMode:=mcAborting,
  Status=>axis1_power
);

rel1(
  Axis:=1,
  Execute:=axis1_power AND rel1_execute,
  Distance:=500,
  Velocity:=30,
  Acceleration:=100,
  Deceleration:=100,
  Jerk:=1000,
  BufferMode:=mcAborting,
  Done=>rel1_done,
  Busy=>rel1_busy,
  Active=>rel1_active,
  CommandAborted=>rel1_abort
);
    
```

```
);  
  
setposition1(  
    Axis:=1,  
    Execute:=axis1_power AND setpos1_execute,  
    Position:=200,  
    Relative:=FALSE,  
    Done=>setpos1_done,  
    Busy=>setpos1_busy,  
);  
  
actposition1(  
    Axis:=1,  
    Enable:=actpos1_enable,  
    Valid=>actpos1_vaild,  
    Busy=>actpos1_busy,  
    Position=>actpos,  
);
```

#### ● 程序说明

- 程序执行后，actpos1\_enable为TRUE，通过反馈位置读取指令（actpos1）读取轴的反馈位置，读取的反馈位置存放在actpos变量中。
- setpos1\_execute变为TRUE时，位置设置指令（setposition1）开始执行，第2个周期，轴的反馈位置从当前位置变为位置设置指令设置的位置，轴的反馈位置从0变为200。位置设置指令（setposition1）执行完成后，setpos1\_execute变为FALSE时，setpos1\_done也同时变为FALSE。
- 位置设置指令（setposition1）执行完后，执行相对位移指令。相对位移指令以设置的位置200为参考点，再走500个单元位置。相对位移指令完成后，轴的位置为700。



## 4.22 MC\_ReadActualPosition (反馈位置读取)

该指令用于周期性读取指定轴的反馈位置。所属库：MotionControl

指令	名称	FB/FUN	梯形图样式	ST样式
MC_ReadActualPosition	反馈位置读取	FB		<pre>MC_ReadActualPosition_Instance ( Axis := 参数 , Enable := 参数 , Valid =&gt; 参数 , Busy =&gt; 参数 , Error =&gt; 参数 , ErrorID=&gt; 参数 , Position=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Enable	启动	BOOL	TRUE或FALSE	FALSE	设为TRUE, 读取指定轴反馈位置。 设为FALSE, 停止读取指定轴反馈位置

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Valid	完成	BOOL	TRUE或FALSE	该指令读取指定轴的反馈位置时, 该参数值状态为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时, 输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”
Position	反馈位置	LREAL	负数、正数、0	根据指定轴的反馈位置通过轴“设置参数”换算后的位置。 单位：行程单位

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Valid	Position (反馈位置) 的值可以读取时	<ul style="list-style-type: none"> <li>◆ Enable变为FALSE时</li> <li>◆ Error变为TRUE时</li> </ul>
Busy	当Enable 为TRUE时	<ul style="list-style-type: none"> <li>◆ Enable变为FALSE时</li> <li>◆ Error为TRUE时</li> </ul>
Error	指令输入变量值不在允许的范围时。	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Enable由TRUE变为FALSE时</li> </ul>

### ■ 功能说明

#### ● 基本功能说明

该指令用于周期读取指定轴的反馈位置, 指定轴可以为伺服轴、虚拟伺服轴、脉冲轴、编码器轴。该指令输入参数 Enable 为 TRUE 时, 轴反馈位置 Position 的值每个任务周期更新一次; Enable 为 FALSE 时, 轴反馈位置 Position 的值停止更新, 并保持最后 Enable 为 TRUE 时的值不变。

- **反馈位置 (Position) 计算方法**

反馈位置 (Position) 的值为通过电机置换算至机构中的位置，单位为行程单位。指定轴为伺服轴时，该指令读取位置来源为电机反馈给驱动器的脉冲数；编码器轴时，该指令读取位置来源为控制器编码器接口接收的脉冲数；虚拟伺服轴时，该指令读取位置来源为虚拟伺服轴的命令位置，不需要换算。指定轴为伺服轴或者编码器轴时，反馈位置 (Position) 的值通过上述来源经过机构参数换算后得到，机构参数可以在软件中 【运动控制】 → 【轴设置】 → 【基本设置】 中查看，转换关系计算方法如下图所示，此转换都是控制器进行运算，不需要用户运算。

$$\text{反馈位置 (Position)} = \frac{\text{电机反馈给驱动器的脉冲数} \times \text{工作每转的工作行程} \times \text{减速机输出转速}}{\text{或编码器接收脉冲数} \times \text{电机每转的脉冲数} \times \text{减速机输入转速}}$$

- **反馈位置更新周期**

- 该指令读取轴的反馈位置，一个任务周期更新一次，同一任务内不同时刻，该指令读取的反馈位置相同。
- 该指令读取轴实际位置的及时性没有MC\_TouchuProbe和MC\_TouchuProbeCyc指令记录轴位置精准，如需轴的精准位置，请使用MC\_TouchuProbe和MC\_TouchuProbeCyc指令记录锁定轴位置。

- **MC\_Home、MC\_SetPosition 和 MC\_HomeWithParm 对实际位置的影响**

MC\_Home、MC\_SetPosition 和 MC\_HomeWithParm 指令执行后，轴命令位置和反馈位置会跟随这些指令的设置位置变化，这些指令执行后，该指令读取到的位置和轴结构体中的反馈位置一致（轴为线性模式）。

- **异常排除**

该指令执行时，如果输入变量不合法，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。

- **示例程序**

示例程序请参考 [MC\\_SetPosition 指令的示例程序](#)。

## 4.23 MC\_TouchProbe (锁定轴位置)

该指令用于记录触发信号发生时刻轴的位置。所属库: MotionControl

指令	名称	FB/FUN	梯形图样式	ST样式
MC_TouchProbe	信号触发锁定轴位置	FB		<pre>MC_Touchprobe_Instance ( Axis := 参数 , Execute := 参数 , TriggerInput := 参数 , Windowonly := 参数 , Firstpos := 参数 , Lastpos := 参数 , Mode := 参数 , Mask := 参数 , Done =&gt; 参数 , Busy =&gt; 参数 , Active=&gt; 参数 , CommandAborted =&gt; 参数 , Error =&gt; 参数 , ErrorID=&gt; 参数 , RecordedPosition=&gt; 参数 );</pre>

#### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
TriggerInput	触发信号指定	MC_Triggerinput	0:mcTriggerinput_l0 1:mcTriggerinput_l1 ... 7: mcTriggerinput_l7 8:mcTriggerinputl_l10 9:mcTriggerinputl_l11 ... 15:mcTriggerinputl_l17	0	该输入变量仅在锁定编码器轴位置时有效,即Mode设置为0和1时有效。如该输入参数置为0时,使用控制器本体输入点%IX0.0进行锁定编码器轴位置;如该输入参数置为1时,使用控制器本体输入点%IX0.1进行锁定编码器轴位置
Windowonly	窗口功能有效或无效设定	BOOL	保留	保留	保留
Firststops	起始位置	LREAL	保留	保留	保留
Laststops	结束位置	LREAL	保留	保留	保留
Mode	模式	INT	0、1、5、6	0	该输入变量用于设定触发锁定轴位置的模式。可以通过设定该变量的值,选择通过控制器的输入点触发锁定编码器轴位置或者通过驱动器的输入点触发锁定伺服轴位置 0: 模式0*1 通过控制器本体输入点的上升沿锁定编码器轴位置,具体使用哪个输入点由输入变量 TriggerInput 指定。锁定位置 RecordedPosition为控制器编码器接口接收脉冲数通过轴参数换算后的位置 1: 模式1*1 通过控制器本体输入点的下降沿锁定编码器轴位置,具体使用哪个输入点由输入变量 TriggerInput 指定。锁定位置 RecordedPosition为控制器编码器接口接收脉冲数通过轴参数换算后的位置

					5: 模式5 *2 通过驱动器本体输入点的上升沿锁定伺服轴位置, 具体使用哪个输入点由驱动器探针功能决定。锁定位置RecordedPosition为电机反馈位置通过轴参数换算后的位置 6: 模式6 *2 通过驱动器本体输入点的下降沿锁定伺服轴位置, 具体使用哪个输入点由驱动器探针功能决定。锁定位置RecordedPosition为电机位置通过轴参数换算后的位置
Mask	过滤	INT	保留	保留	保留

\*1: 多个信号触发锁定轴位置指令同时使用模式0和模式1时, 模式0和模式1不能指定为同一个输入点。

\*2: 模式5和模式6仅M500S系列、M500系列控制器支持。

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	触发信号触发, 记录锁定位置时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Active	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时, 输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”
RecordedPosition	锁定位置	LREAL	正数、负数、0	触发信号发生时刻锁定轴的反馈位置该位置为电机反馈位置或者控制器编码器接口接收脉冲数通过轴参数换算后的位置

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	触发信号触发, 记录锁定位置时	<ul style="list-style-type: none"> <li>◆ Done为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE</li> </ul>
Busy	Execute的上升沿	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
Active	指令开始控制轴时	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
CommandAborted	指令的执行被其它指令中断时	<ul style="list-style-type: none"> <li>◆ CommandAborted 为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令被其它指令中断时, CommandAborted变为TRUE, 一个周期后, 变为FALSE</li> </ul>
Error	指令输入变量值不在允许的范围	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Execute从TRUE变为FALSE时</li> </ul>

### ■ 功能说明

#### ● 基本功能说明

该指令用于记录触发信号发生时刻指定轴的反馈位置, 指定轴可以为伺服轴和编码器轴。Execute 由 FALSE 变为 TRUE 时, 该指令根据设定的输入变量, 执行该指令。该指令指定的触发条件触发时, 指定轴触发时刻的位置输出到 RecordedPosition (锁定位置)内, 该指令执行完成。

#### ● 数据对象的映射 (PDO 映射)

M500S 系列控制器不需要用户映射数据对象。M500 系列控制器需要用户映射数据对象，映射对象如下表所示。

接收PDO (主站=>从站) (16进制)	映射数据含义	发送PDO (从站=>主站)	映射数据含义
6040_0 (索引_子索引)	控制字	6041_0 (索引_子索引)	状态字
6060_0 (索引_子索引)	控制模式	6061_0 (索引_子索引)	反馈模式
60B8_0 (索引_子索引)	锁定轴位置功能	60B9_0 (索引_子索引)	锁定轴位置状态
		60BA_0 (索引_子索引)	上升沿锁定轴位置1
		60BB_0 (索引_子索引)	下降沿锁定轴位置1

### ● 锁定位置

- RecordedPosition (锁定位置)在外部触发条件成立时被记录并保持不变，直到下次外部触发条件成立时再次记录新的锁定位置。RecordedPosition (锁定位置)的值为数据源通过轴参数换算后得到，单位为行程单位。数据源请参考下表的说明。轴参数可以在软件中【运动控制】→【轴设置】→【基本设置】中查看。输入变量Axis指定轴为伺服轴时，RecordedPosition (锁定位置)由电机反馈给驱动器的脉冲数通过轴参数换算后得到；输入变量Axis指定轴为编码器轴时，RecordedPosition (锁定位置)由编码器接收脉冲数通过轴参数换算后得到。

数据源	对应Mode (模式) 设定	触发信号
电机反馈给驱动器的脉冲数	Mode的值设置为5或6	通过驱动器本体输入点的上升沿，输入点由驱动器定义。
编码器1接收脉冲数。 <sup>*1</sup> 编码器轴和编码器的对应关系在软件中指定。	Mode的值设置为0或1	由输入变量TriggerInput指定，M300系列、M500S系列可以指定为%IX0.2~%IX0.7。
编码器2接收脉冲数。 <sup>*1</sup> 编码器轴和编码器的对应关系在软件中指定。	Mode的值设置为0或1	由输入变量TriggerInput指定，M300系列、M500S系列可以指定为可以指定为%IX0.0~%IX0.1,%IX0.4~%IX0.7。

\*1: 编码器1接口为输入点%IX0.0、%IX0.1，编码器2接口为输入点%IX0.2、%IX0.3。

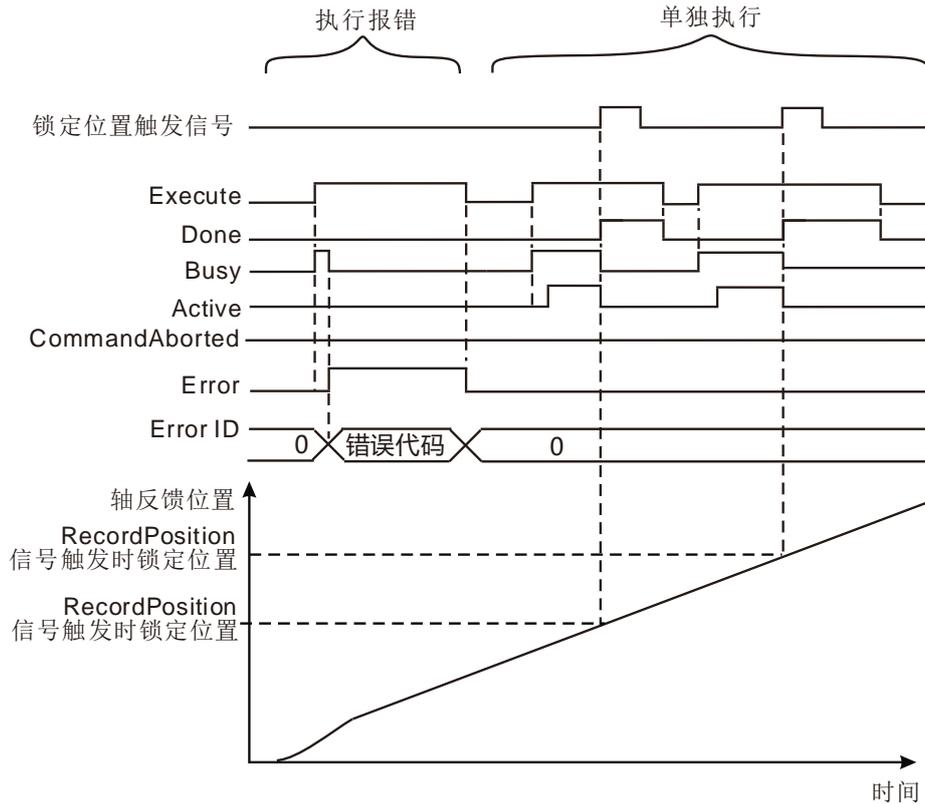
- RecordedPosition (锁定位置)为双精度浮点数，该位置的数据来源越界时，控制器内部有做处理，该位置不会跟随越界，会累计。如驱动器位置类型为32有符号数，当驱动器正转时，位置从0运转至2147483647后，再正转1个脉冲时，驱动器位置变为-2147483648，而该指令锁定到的位置为2147483648。
- 电机位置或者编码轴接收脉冲数没有越界时，RecordedPosition (锁定位置)的值计算方法如下：

$$RecordedPosition(锁定位置) = \frac{\text{电机反馈给驱动器的脉冲数} \times \text{工作每转的工作行程} \times \text{减速机输出转速} \text{ 或 } \text{编码器接收脉冲数}}{\text{电机每转的脉冲数} \times \text{减速机输入转速}}$$

### ● 异常排除

该指令执行时，如果输入变量不合法，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。

## ■ 输出变量变化时序说明



### ● 执行报错

该指令输入变量值不在允许范围内时，该指令 Execute（启动）由 FALSE 变为 TRUE 时，同时 Busy（执行中）变为 TRUE，下一个周期 Error（错误）变为 TRUE，同时 Busy（执行中）变为 FALSE，ErrorID（错误码）输出对应的错误码，可通过 ErrorID（错误码）的值，查找发生问题原因。该指令 Execute（启动）由 TRUE 变为 FALSE 时，同时 Error（错误）变为 FALSE，ErrorID（错误码）的值变为 0。

### ● 单独执行

Execute（启动）由 FALSE 变为 TRUE 时，Busy（执行中）同时变为 TRUE，下一个周期 Active（控制中）变为 TRUE。当检测到外部触发信号时，Done（完成）变为 TRUE，Busy（执行中）和 Active（控制中）同时变为 FALSE，记录的锁定位置存放在 RecordedPosition 对应的变量中。当 Execute（启动）变为 FALSE 时，Done（完成）同时变为 FALSE。

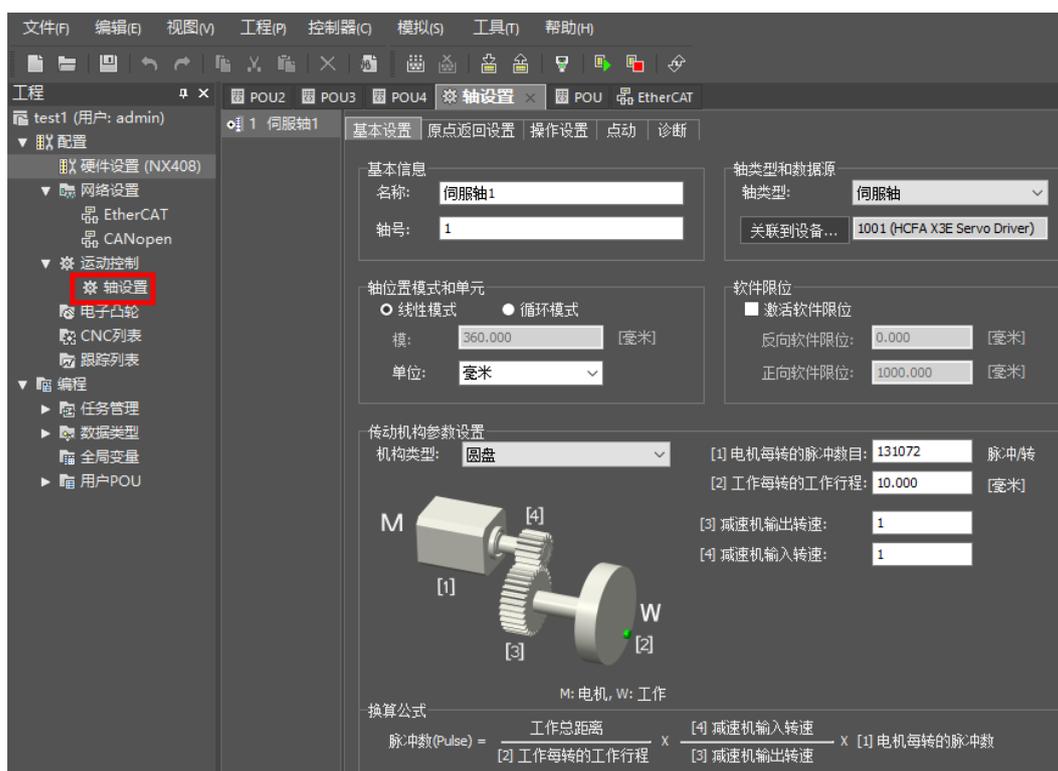
## ■ 示例程序

### ● 实现功能

速度指令控制轴运转时，外部触发信号条件成立时，记录轴的反馈位置。

### ● 轴参数设置

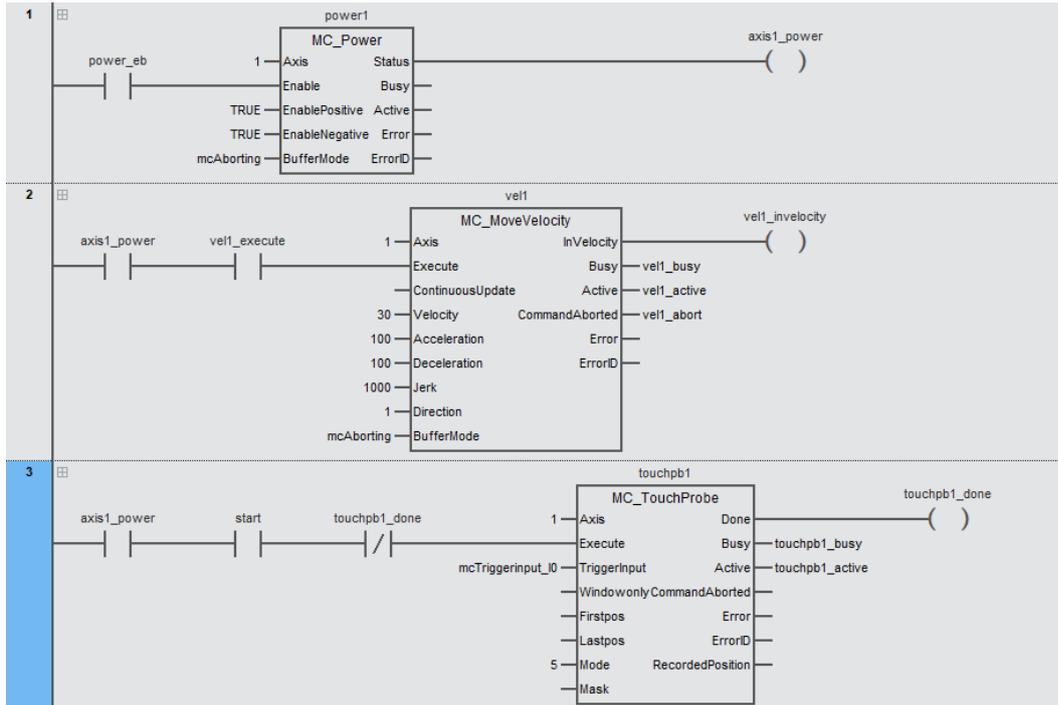
轴 1 轴参数设置如下图所示。



● 变量表

类别	名称	分配到	数据类型	初始值	注释
VAR	power_eb		BOOL		
VAR	power1		MC_Power		
VAR	axis1_power		BOOL		
VAR	vel1_execute		BOOL		
VAR	vel1		MC_MoveVelocity		
VAR	vel1_invelocity		BOOL		
VAR	vel1_busy		BOOL		
VAR	vel1_active		BOOL		
VAR	vel1_abort		BOOL		
VAR	touchpb1		MC_TouchProbe		
VAR	touchpb1_done		BOOL		
VAR	touchpb1_busy		BOOL		
VAR	touchpb1_active		BOOL		
VAR	start		BOOL		
VAR	touchpb1_actpos		LREAL		

● 梯形图(LD)



● 结构化文本(ST)

```

power1(
    Axis:=1 ,
    Enable:=powerEb ,
    EnablePositive:=TRUE ,
    EnableNegative:=TRUE ,
    BufferMode:=mcAborting ,
    Status=>axis1_power
);

vel1(
    Axis:=1 ,
    Execute:= axis1_power AND vel1_execute ,
    Velocity:=30 ,
    Acceleration:=100 ,
    Deceleration:=100 ,
    Jerk:=1000 ,
    Direction:=1 ,
    BufferMode:=mcAborting ,
    InVelocity=> vel1_invelocity ,
    Busy=> vel1_busy ,
    Active=> vel1_active ,
    CommandAborted=> vel1_abort ,
);

touchpb1(
    Axis:=1 ,
    Execute:=axis1_power AND start AND NOT touchpb1_done ,
    TriggerInput:=mcTriggerinput_I0 ,
    Mode:=5
    Done=>touchpb1_done ,
);
    
```

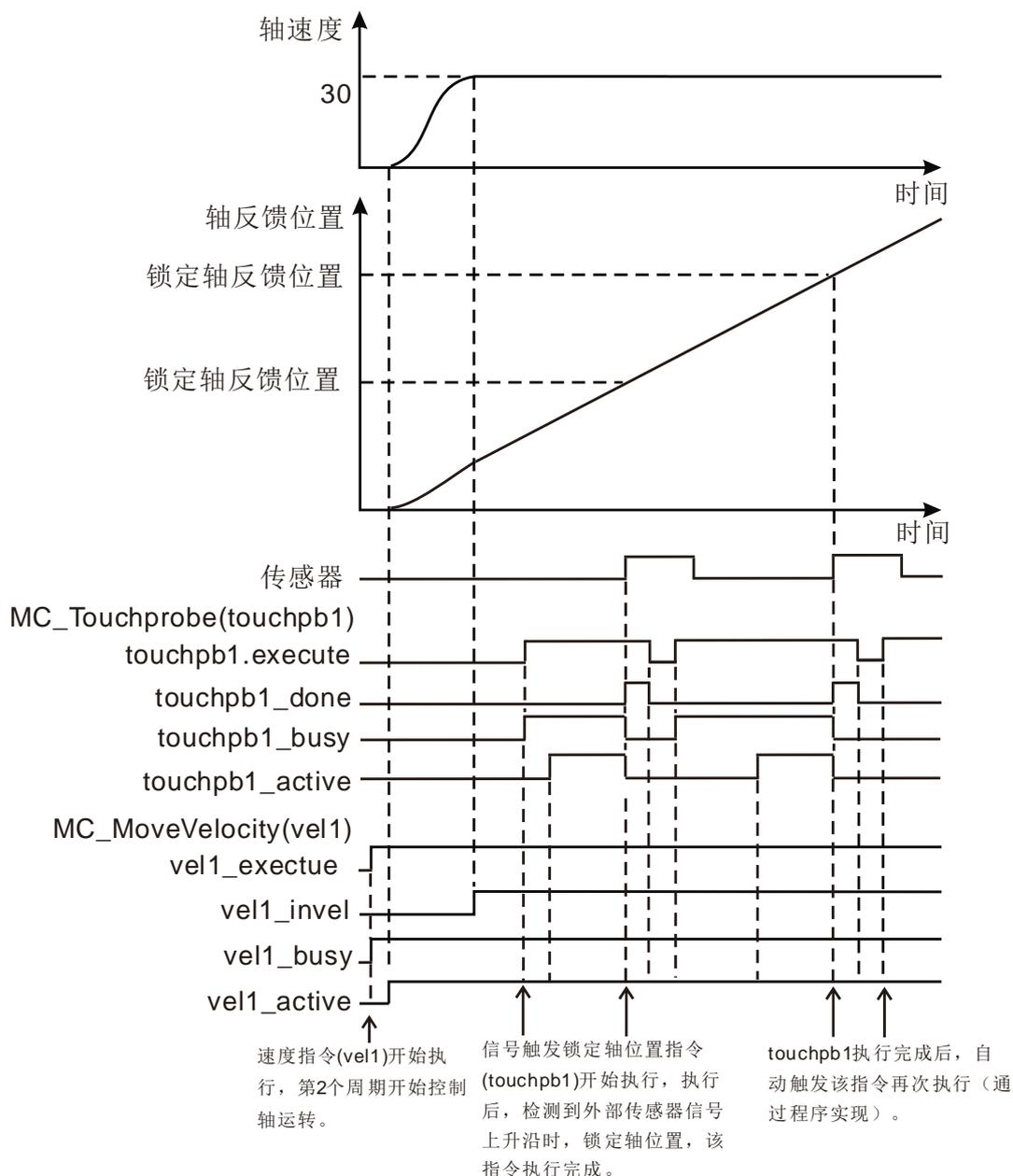
```

Busy=>touchpb1_busy,
Active=>touchpb1_active,
Active=>touchpb1_actpos
);

```

### ● 程序说明

- vel1\_execute变为TRUE时，速度指令（vel1）开始执行，第2个周期，开始控制轴运转。
- start为TRUE时，信号触发锁定轴位置（touchpb1）指令执行，检测到外部传感器信号上升沿时，记录轴反馈位置到touchpb1\_actpos变量内，该指令执行完成。该指令执行完成后，会自动触发该指令再次执行（通过程序实现），等待外部传感器信号上升沿，检测到外部传感器信号上升沿时，记录轴反馈位置，该指令执行完成，如此循环。



## 4.24 MC\_TouchprobeCyc (信号触发锁定轴位置周期执行)

该指令用于周期性记录触发信号发生时刻轴的位置。所属库：MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_TouchProbeCyc	信号触发锁定轴位置 周期执行	FB		<pre> MC_TouchprobeCyc_Instance ( Axis := 参数 , Enable:= 参数 , TriggerInput := 参数 , Windowonly:= 参数 , Firstpos := 参数 , Lastpos := 参数 , Mode := 参数 , Mask := 参数, Vaild =&gt; 参数 , Busy =&gt; 参数 , Active=&gt; 参数 , CommandAborted =&gt; 参数 , Error =&gt; 参数 , ErrorID=&gt; 参数 , Touched=&gt; 参数 , RecordedPosition=&gt; 参数 );                     </pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Enable	启动	BOOL	TRUE或FALSE	FALSE	该参数为TRUE时执行该指令
TriggerInput	触发信号指定	MC_Triggerinput	0:mcTriggerinput_I0 1:mcTriggerinput_I1 ... 7: mcTriggerinput_I7 8:mcTriggerinput_I10 9:mcTriggerinput_I11 ... 15:mcTriggerinput_I17	0	该输入变量仅在锁定编码器轴位置时有效，即Mode设置为0和1时有效。如该输入参数置为0时，使用控制器本体输入点%IX0.0进行锁定编码器轴位置；如该输入参数置为1时，使用控制器本体输入点%IX0.1进行锁定编码器轴位置
Windowonly	窗口功能有效或无效设定	BOOL	保留	保留	保留
Firststops	起始位置	LREAL	保留	保留	保留
Laststops	结束位置	LREAL	保留	保留	保留
Mode	模式	INT	0、1、5、6	0	该输入变量用于设定触发锁定轴位置的模 式。可以通过设定该变量的值，选择通过 控制器的输入点触发锁定编码器轴位置或 者通过驱动器的输入点触发锁定伺服轴位 置 0: 模式0*1 通过控制器本体输入点的上升沿锁定编码 器轴位置，具体使用哪个输入点由输入变 量 TriggerInput 指定。锁定位置 RecordedPosition为控制器编码器接口 接收脉冲数通过轴参数换算后的位置 1: 模式1*1 通过控制器本体输入点的下降沿锁定编码 器轴位置，具体使用哪个输入点由输入变 量 TriggerInput 指定。锁定位置 RecordedPosition为控制器编码器接口

					接收脉冲数通过轴参数换算后的位置。 5: 模式5 *2 通过驱动器本体输入点的上升沿锁定伺服轴位置, 具体使用哪个输入点由驱动器探针功能决定。锁定位置RecordedPosition为电机反馈位置通过轴参数换算后的位置 6: 模式6*2 通过驱动器本体输入点的下降沿锁定伺服轴位置, 具体使用哪个输入点由驱动器探针功能决定。锁定位置RecordedPosition为电机位置通过轴参数换算后的位置
Mask	过滤	INT	保留	保留	保留

\*1: 多个信号触发锁定轴位置指令同时使用模式0和模式1时, 模式0和模式1不能指定为同一个输入点。

\*2: 模式5和模式6仅M500S系列、M500系列控制器支持。

## ■ 输出变量

名称	名称	数据类型	输出范围	功能
Vaild	完成	BOOL	TRUE或FALSE	该指令正常执行时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Active	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时, 输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”
Touched	锁定位置完成	BOOL	TRUE或FALSE	触发信号触发, 记录锁定位置时变为TRUE
RecordedPosition	锁定位置	LREAL	正数、负数、0	触发信号发生时刻锁定轴的反馈位置该位置为电机反馈位置或者控制器编码器接口接收脉冲数通过轴参数换算后的位置

## ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Vaild	Enable变为TRUE时	<ul style="list-style-type: none"> <li>◆ Enable变为FALSE时</li> <li>◆ Error变为TRUE时</li> </ul>
Busy	Enable变为TRUE时	<ul style="list-style-type: none"> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
Active	指令正常执行时	<ul style="list-style-type: none"> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
CommandAborted	指令的执行被其它指令中断时	<ul style="list-style-type: none"> <li>◆ CommandAborted 为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令被其它指令中断时, CommandAborted变为TRUE, 一个周期后, 变为FALSE</li> </ul>
Error	指令输入变量值不在允许的范围	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Execute从TRUE变为FALSE时</li> </ul>

## ■ 功能说明

### ● 基本功能说明

该指令用于记录触发信号发生时刻指定轴的反馈位置, 指定轴可以为伺服轴和编码器轴。Enable 为 TRUE 时, 外部触发条件成立时, 指定轴触发时刻的反馈位置输出到 RecordedPosition (锁定位置)内, 外部触发条件触发一次, 记录一次。该指令和 MC\_Touchprobe 指令的区别为该指令不需要循环触发执行, Enable 为 TRUE 时, 记录锁定位置后会为下次记录锁定位置做准备, 不需要程序处理; MC\_Touchprobe 指令执行时, 记录锁定

位置后需要通过程序处理触发该指令再次执行，该指令再次执行后，指定的触发条件成立时，才可以记录下次轴的锁定位置。

### ● 锁定位置

RecordedPosition (锁定位置)在外部触发条件成立时被记录并保持不变，直到下次外部触发条件成立时再次记录新的锁定位置。RecordedPosition (锁定位置)的值为数据源通过轴参数换算后得到，单位为行程单位。数据源请参考下表的说明。轴参数可以在软件中【运动控制】→【轴设置】→【基本设置】中查看。输入变量 Axis 指定轴为伺服轴时，RecordedPosition (锁定位置)由电机反馈给驱动器的脉冲数通过轴参数换算后得到；输入变量 Axis 指定轴为编码器轴时，RecordedPosition (锁定位置)由编码器接收脉冲数通过轴参数换算后得到。

数据源	对应Mode (模式) 设定	触发信号
电机反馈给驱动器的脉冲数	Mode的值设置为5或6	通过驱动器本体输入点的上升沿，输入点由驱动器定义。
编码器1接收脉冲数。*1 编码器轴和编码器的对应关系在软件中指定。	Mode的值设置为0或1	由输入变量TriggerInput指定，M300系列、M500S系列可以指定为%IX0.2~%IX0.7。
编码器2接收脉冲数。*1 编码器轴和编码器的对应关系在软件中指定。	Mode的值设置为0或1	由输入变量TriggerInput指定，M300系列、M500S系列可以指定为可以指定为%IX0.0~%IX0.1, %IX0.4~ %IX0.7。

\*1: M500S系列编码器1接口为输入点%IX0.0、%IX0.1，编码器2接口为输入点%IX0.2、%IX0.3。

### ● 数据对象的映射 (PDO 映射)

M500S 系列控制器不需要用户映射数据对象。M500 系列控制器需要用户映射数据对象，映射对象如下表所示。

接收PDO (主站→从站) (16进制)	映射数据含义	发送PDO (从站→主站)	映射数据含义
6040_0 (索引_子索引)	控制字	6041_0 (索引_子索引)	状态字
6060_0 (索引_子索引)	控制模式	6061_0 (索引_子索引)	反馈模式
60B8_0 (索引_子索引)	锁定轴位置功能	60B9_0 (索引_子索引)	锁定轴位置状态
		60BA_0 (索引_子索引)	上升沿锁定轴位置1
		60BB_0 (索引_子索引)	下降沿锁定轴位置1

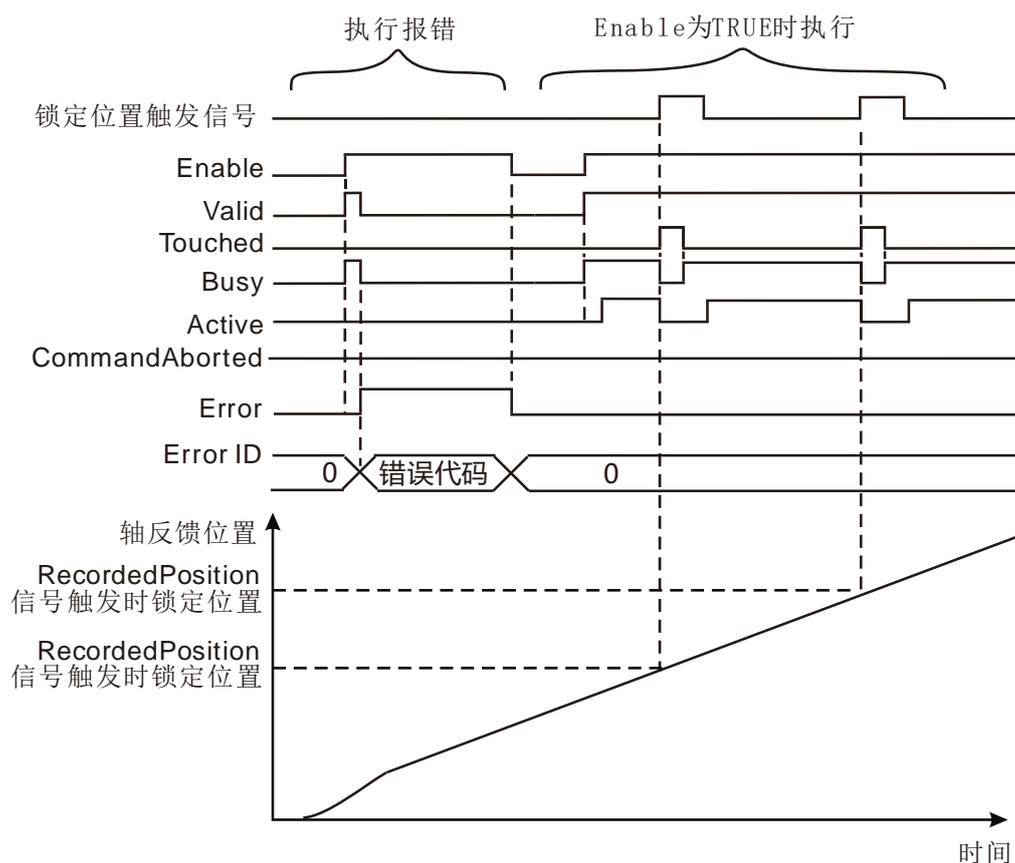
- RecordedPosition (锁定位置)为双精度浮点数，该位置的数据来源越界时，控制器内部有做处理，该位置不会跟随越界，会累计。如驱动器位置类型为32位有符号数，当驱动器正转时，位置从0运转至2147483647后，再正转1个脉冲时，驱动器位置变为-2147483648，而该指令捕获到的位置为2147483648。
- 电机位置或者编码轴接收脉冲数没有越界时，RecordedPosition (锁定位置)的值计算方法如下所示。

$$RecordedPosition(\text{锁定位置}) = \frac{\text{电机反馈给驱动器的脉冲数} \times \text{工作每转的工作行程} \times \text{减速机输出转速}}{\text{或编码器接收脉冲数} \times \text{电机每转的脉冲数} \times \text{减速机输入转速}}$$

### ● 异常排除

该指令执行时，如果输入变量不合法，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。

## ■ 输出变量变化时序说明



### ● 执行报错

该指令输入变量值不在允许范围内时，该指令 Enable（启动）由 FALSE 变为 TRUE 时，同时 Valid（有效）和 Busy（执行中）变为 TRUE，下一个周期 Error（错误）变为 TRUE，同时 Valid（有效）和 Busy（执行中）变为 FALSE，ErrorID（错误码）输出对应的错误码，可通过 ErrorID（错误码）的值，查找发生问题原因。该指令 Enable（启动）由 TRUE 变为 FALSE 时，同时 Error（错误）变为 FALSE，ErrorID（错误码）的值变为 0。

### ● 执行

Enable（有效）由 FALSE 变为 TRUE 时，Valid（有效）和 Busy（执行中）同时变为 TRUE，下一个周期 Active（控制中）变为 TRUE。当检测到外部触发信号时，Touched（锁定位置完成）变为 TRUE 一个任务周期，Busy（执行中）和 Active（控制中）同时变为 FALSE，触发时刻锁定轴的位置存放在 RecordedPosition 对应的变量中。Touched 变为 TRUE 一个任务周期后，又自动变为 FALSE，Busy（执行中）同时变为 TRUE，一个任务周期后，Active（控制中）变为 TRUE。

## 4.25 MC\_MoveFeed (外部中断后移动指定的距离)

此指令启动后，控制指定轴进行位移或者速度运动，该轴控制轴移动位置超过指定的相对位置后，轴运动速度由高速切换为低速，以低速寻找外部中断信号，检测到外部中断输入后，以中断输入信号抓取轴的实际位置为参考点，再移动指定的相对距离。所属库：MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_MoveFeed	外部中断后移动指定的距离	FB		<pre>MC_MoveFeed_Instance(     Axis:= 参数 ,     EncoderAxis:= 参数 ,     Execute:= 参数 ,     Mode:= 参数 ,     TriggerMode:= 参数 ,     TriggerInput:= 参数 ,     TriggerVar:= 参数 ,     Windowonly:= 参数 ,     WindowMode:= 参数 ,     FirstPosition:= 参数 ,     LastPosition:= 参数 ,     Position:= 参数 ,     Velocity:= 参数 ,     Acceleration:= 参数 ,     Deceleration:= 参数 ,     Jerk:= 参数 ,     Direction:= 参数 ,     MoveMode:= 参数 ,     TouchPosition:= 参数 ,     TouchVelocity:= 参数 ,     FeedDistance:= 参数 ,     FeedVelocity:= 参数 ,     BufferMode:= 参数 ,     Done=&gt; 参数 ,     Busy=&gt; 参数 ,     Active=&gt; 参数 ,     InFeed=&gt; 参数 ,     CommandAborted=&gt; 参数 ,     Error=&gt; 参数 ,     ErrorID=&gt; 参数 ,     TriggerPosition=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
EncoderAxis	编码器轴的轴号	USINT	由机种决定	不可缺省	指定编码器轴的轴号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时，根据 MoveMode (移动模式) 的值执行位移或者速度运动
Mode	触发模式	INT	0、1	0	指定触发的模式 0: 外部中断输入锁定轴位置后，再移动指定的相对距离 1: 检测到输入变量TriggerVar的上升沿后，再移动指定的相对距离
TriggerMode	触发方式	INT	0、1、5、6	0	该输入变量用于设定触发锁定轴位置的模式。可以通过设定该变量的值，选择通过控制器的输入点触发锁定编码器轴位置或

					<p>者通过驱动器的输入点触发锁定伺服轴位置</p> <p>0: 模式0*1 通过控制器本体输入点的上升沿锁定编码器轴位置, 具体使用哪个输入点由输入变量TriggerInput指定。锁定位置RecordedPosition为控制器编码器接口接收脉冲数通过轴参数换算后的位置</p> <p>1: 模式1*1 通过控制器本体输入点的下降沿锁定编码器轴位置, 具体使用哪个输入点由输入变量TriggerInput指定。锁定位置RecordedPosition为控制器编码器接口接收脉冲数通过轴参数换算后的位置</p> <p>5: 模式5*5 通过驱动器本体输入点的上升沿锁定伺服轴位置, 具体使用哪个输入点由驱动器探针功能决定。锁定位置RecordedPosition为电机反馈位置通过轴参数换算后的位置</p> <p>6: 模式6*5 通过驱动器本体输入点的下降沿锁定伺服轴位置, 具体使用哪个输入点由驱动器探针功能决定。锁定位置RecordedPosition为电机位置通过轴参数换算后的位置</p>
TriggerInput	触发信号指定	MC_Triggerinput	<p>0:mcTriggerinput_I0 1:mcTriggerinput_I1 ... 7: mcTriggerinput_I7 8:mcTriggerinputl_I10 9:mcTriggerinputl_I11 ... 15:mcTriggerinputl_I17</p>	0	<p>该输入变量仅在锁定编码器轴位置时有效, 即Mode设置为0和1时有效。 如该输入参数设置为0时, 使用控制器本体输入点%IX0.0进行锁定编码器轴位置; 如该输入参数设置为1时, 使用控制器本体输入点%IX0.1进行锁定编码器轴位置。</p>
TriggerVar	触发变量	BOOL	TRUE或FALSE	FALSE	Mode设置为1时, 检测到该输入变量的上升沿后, 再移动指定的相对距离。
Windwonly	窗口功能	BOOL	TRUE或FALSE	FALSE	<p>指定窗口功能是否有效 FALSE: 窗口功能无效 TRUE: 窗口功能有效</p>
WindowMode	窗口模式	BOOL	TRUE或FALSE	FALSE	<p>指定窗口功能的模式 FALSE: 绝对模式 TRUE: 相对模式</p>
FirstPosition	窗口起始位置	LREAL	负数、正数、0	0	<p>设定窗口区域的起始位置 线性模式: 窗口区域的起始位置 循环模式: <math>0 \leq \text{FirstPosition} &lt; \text{模}</math></p>
LastPosition	窗口终止位置	LREAL	负数、正数、0	0	<p>设定窗口区域的终止位置 线性模式: 窗口区域的终止位置 循环模式: <math>0 \leq \text{LastPosition} &lt; \text{模}</math></p>
Position	目标位置	LREAL	负数、正数、0	0	<p>目标位置的含义由输入变量MoveMode(移动模式)的值决定。 MoveMode值为1(相对值定位)时, Position的值为相对位置, 以 Exexute由FALSE变为TRUE时的命令位置为参考点。 MoveMode值为0(绝对值定位)时, Position的值为绝对位置。 线性模式: 无限制</p>

					循环模式：0≤Position<模轴位置模式可以在软件中“轴设置”→“基本设置”中进行设置。 MoveMode值为2（速度控制）时，Position的值无意义。
Velocity	目标速度	LREAL	正数	不可缺省	指定目标速度*2 (单位: 行程单位/秒) *3
Acceleration	加速度	LREAL	正数	不可缺省	指定加速度*2 (单位: 行程单位/秒²) *3
Deceleration	减速度	LREAL	正数	不可缺省	指定减速度*2 (单位: 行程单位/秒²) *3
Jerk	跃度	LREAL	正数	不可缺省	指定跃度*2 (单位: 行程单位/秒³) *3
Direction	方向	MC_Direction	1: mcPositiveDirection 2: mcShortestWay 3: mcNegativeDirection 4: mcCurrentDirection	1	轴运转方向 1: 正方向 2: 移动距离最短 3: 反方向 4: 按当前运转方向运转, 轴静止时为正方向移动
MoveMode	移动模式	MC_MoveMode	0: mcAbsolute 1: mcRelative 2: mcVelocity		设置该指令启动时的运动方式 0: 绝对值定位 1: 相对值定位 2: 速度控制
TouchPosition	切换为低速时的相对位置	LREAL	负数、正数、0	0	该参数从该指令开始执行计算相对位置, 超过该参数设定的位置后, 轴的目标速度由 Velocity设定的速度变为 TouchVelocity设定的速度。 该位置为相对位置, 初始参考位置为该指令执行时的轴的命令位置。
TouchVelocity	超过 TouchPosition后的目标速度	LREAL	负数、正数、0	0	超过 TouchPosition参数设定的位置后, 轴的目标速度由 Velocity设定的速度变为 TouchVelocity设定的速度。
FeedDistance	检测到外部中断后需要移动的相对距离	LREAL	负数、正数、0	0	检测到外部中断输入后需要移动的相对距离*2 (单位: 行程单位) *3
FeedVelocity	检测到外部中断后进行相对定位时的目标速度	LREAL	正数	不可缺省	检测到外部中断输入后, 指定相对距离移动时的目标速度*2 (单位: 行程单位/秒) *3
BufferMode	缓存模式	MC_Buffer_Mode	0: mcAborting 1: mcBuffered 2: mcBlendingLow 3: mcBlendingPrevious 4: mcBlendingNext 5: mcBlendingHigh	0	设定两个指令之间中继模式*4 0: 中断 1: 等待 2: 以低速中继 3: 以当前指令目标速度中继 4: 以缓存指令目标速度中继 5: 以高速中继

\*1: 多个信号触发锁定轴位置指令同时使用模式0和模式1时, 模式0和模式1不能指定为同一个输入点。

\*2: Velocity、Acceleration、Deceleration、Jerk的关系请参考“[运动控制指令相关参数说明](#)”章节。

\*3: 指令单位的详细介绍请参考“[运动控制指令参数单位](#)”章节。

\*4: BufferMode的详细介绍请参考“[运动控制指令的多重启动时缓存模式说明 \(BufferMode\)](#)”章节。

\*5: 模式5和模式6仅M500S系列、M500系列控制器支持。

## ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	外部中断触发相对定位完成时 没有外部中断时MoveMode指定的定位完成时
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Active	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE
InFeed	中断触发	BOOL	TRUE或FALSE	指定的中断信号触发时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码 值的含义请参考“ <a href="#">指令错误代码描述</a> ”
TriggerPosition	触发锁定轴位置	LREAL	负数、正数、0	指定信号触发时锁定轴的位置

## ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	外部中断触发相对定位完成时 没有外部中断时，MoveMode指定的定位完成时	<ul style="list-style-type: none"> <li>◆ Done为TRUE，Execute从TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时，Done变为TRUE一个周期后，变为FALSE</li> </ul>
Busy	Execute的上升沿	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
Active	指令开始控制轴时	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
InFeed	指定的中断信号触发时	<ul style="list-style-type: none"> <li>◆ InFeed变为TRUE后，当Execute由TRUE变为FALSE时</li> </ul>
CommandAborted	指令的执行被其它指令中断时	<ul style="list-style-type: none"> <li>◆ CommandAborted 为TRUE，Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE，该指令被其它指令中断时，CommandAborted变为TRUE，一个周期后，变为FALSE</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Error为TRUE，Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE，该指令执行过程中遇到异常时，Error变为TRUE，一个周期后，变为FALSE</li> </ul>

## ■ 功能说明

### ● 基本功能说明

- 该指令启动后，控制指定轴进行位移或者速度运动，该指令控制轴位置超过TouchPosition设定的相对位置时，轴运动速度由高速切换为低速，以低速寻找外部中断信号，检测到外部中断输入后，以中断输入信号抓取轴的实际位置为参考点，再移动指定的相对距离。
- 该指令Execute由FALSE变为TRUE时，轴根据MoveMode参数的设定，按照绝对值移动，相对值移动或者速度控制中的一种方式进行运动。采用绝对值移动时，Position参数为设定绝对位置；采用相对值移动时，Position参数为设定的相对位置；采用速度控制时，参数Position无效，指令控制轴以Velocity为目标速度开始运动。无论MoveMode选择何种模式，都以Velocity的设定值为目标速度控制轴运动。该指令控制轴位置超过TouchPosition设定的相对位置时（TouchPosition是以该指令Execute由FALSE变为TRUE时轴位置为参考点的相对位置），轴的目标速度由Velocity切换为TouchVelocity，速度切换的目的是在低速时寻找中断信号，如此遇到中断信号后停止，停止的位置更准确。该指令切换到TouchVelocity设定的目标速度后，寻找外部中断信号，检测到外部中断信号后，以中断输入信号抓取轴的实际位置为参考点，以FeedVelocity的设定值为目标速度，以FeedDistance指定的距离进行相对位移运动，FeedDistance中填正数，轴向正方向移

动相对距离，填负数，轴沿负方向移动相对距离。如果没有外部中断信号，MoveMode选择绝对值定位或者相对值定位时，轴到达Position设定的目标位置后停止；MoveMode选择速度控制时，先以Velocity的设定值为目标速度控制轴运动，该指令控制轴位置超过TouchPosition设定的相对位置时，轴以TouchVelocity设定的速度为目标速度运动。

#### ● 数据对象的映射（PDO 映射）

M500S 系列控制器不需要用户映射数据对象。M500 系列控制器需要用户映射数据对象，映射对象如下表所示。

接收PDO（主站=>从站） (16进制)	映射数据含义	发送PDO（从站=>主站）	映射数据含义
6040_0（索引_子索引）	控制字	6041_0（索引_子索引）	状态字
6060_0（索引_子索引）	控制模式	6061_0（索引_子索引）	反馈模式
60B8_0（索引_子索引）	锁定轴位置功能	60B9_0（索引_子索引）	锁定轴位置状态
		60BA_0（索引_子索引）	上升沿锁定轴位置1
		60BB_0（索引_子索引）	下降沿锁定轴位置1

#### ● 触发模式

- Mode为0时，为外部中断触发定位模式，外部中断输入锁定轴位置后，再移动FeedDistance指定的相对距离。参数TriggerMode指定通过驱动器的输入点锁定伺服轴的位置或者通过控制器的输入点锁定编码器轴的位置。

- (a) TriggerMode为模式0和模式1时，通过控制器输入点锁定编码器轴的位置再对伺服轴进行定位。编码器轴和伺服轴通过硬件接线建立关系（驱动器的脉冲输出接控制器的编码器输入），可使用该模式通过捕获编码器轴的位置间接得到伺服轴位置。参数EncoderAxis设定编码器轴的轴号。

参数TriggerInput用于设定锁定编码轴位置的输入信号，该参数仅TriggerMode设置为模式0或者模式1有效。如该输入参数置为0时，使用控制器本体输入点%IX0.0进行锁定编码器轴位置；如该输入参数设置为1时，使用控制器本体输入点%IX0.1进行锁定编码器轴位置。

- (b) TriggerMode为模式5，6时，通过触发驱动器输入点锁定伺服轴位置。

模式5：通过驱动器输入点的上升沿锁定伺服轴位置。锁定伺服轴位置为伺服电机反馈给伺服驱动器的脉冲数通过轴参数换算后的位置。

模式6：通过驱动器输入点的下降沿锁定伺服轴位置。锁定伺服轴位置为伺服电机反馈给伺服驱动器的脉冲数通过轴参数换算后的位置

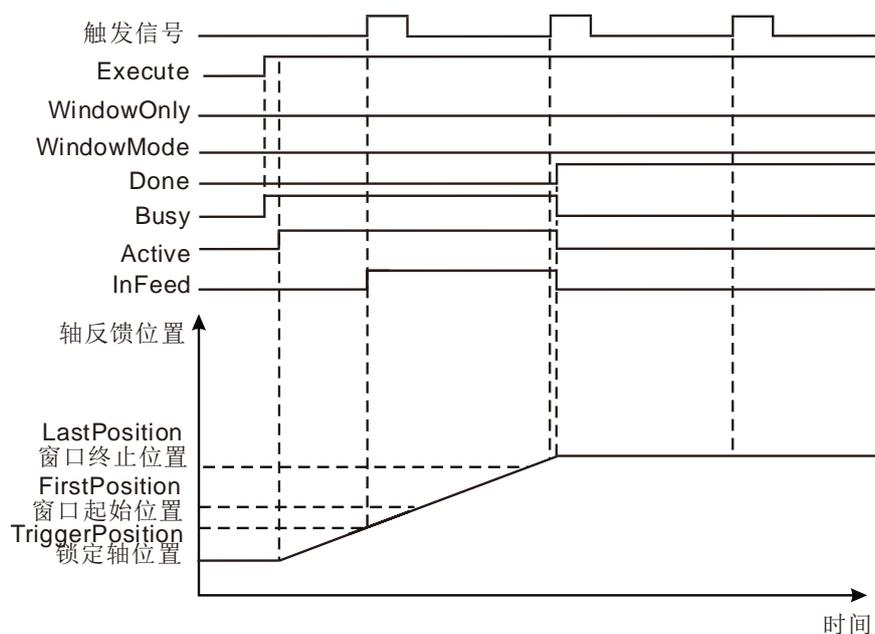
- Mode为1时，为变量触发定位模式，控制器检测到输入变量TriggerVar的上升沿后，再移动FeedDistance指定的相对距离。参数TriggerVar设定触发的变量，可以为控制器内部变量，也可以使用该变量绑定控制器本体输入点做为外部触发信号。使用该变量的上升沿进行触发，是通过扫描程序的方式检测TriggerVar变量的上升沿，不是通过中断的方式检测，由于IO刷新的延迟，Mode设置值为1的定位精度低于Mode设置值为0时的定位精度。

注：在使用变量触发定位模式时，触发变量必须在运动事件任务内为 TRUE 一个任务周期，以防止检测不到TriggerVar 的上升沿。

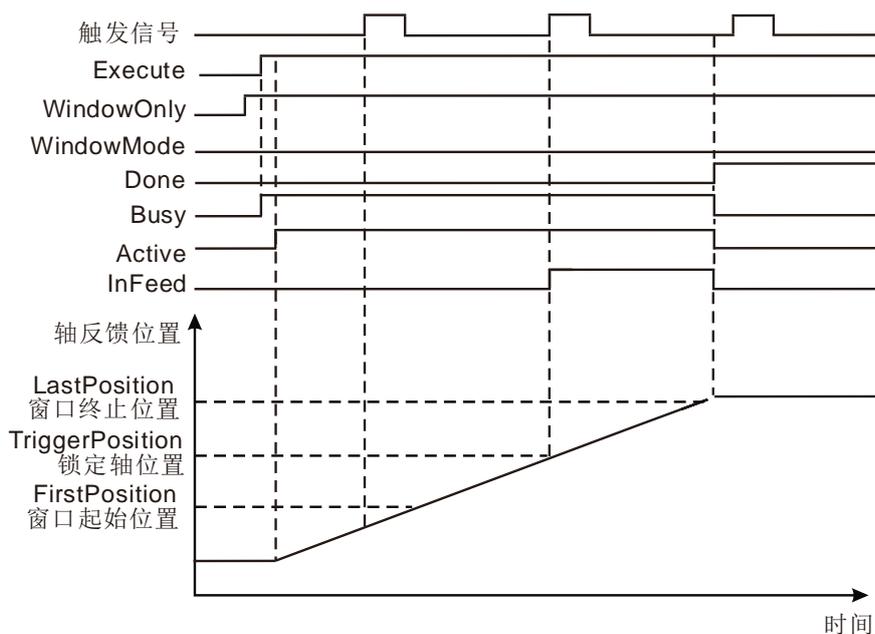
#### ● 窗口功能

- Windowly设定窗口功能是否有效

1) Windowly为FALSE时，窗口功能无效，轴在任意位置检测到外部中断信号，都可以触发相对定位动作。



- 2) WindowOnly为TRUE时，窗口功能有效，轴的位置仅在窗口有效范围内检测到外部中断信号时，才可以触发相对定位动作。



#### ➤ WindowMode设定窗口模式

- 1) WindowMode为FALSE时，窗口模式为绝对模式。

窗口模式为绝对模式时，如果指令控制的轴为线性模式，窗口有效范围为 $\text{FirstPosition} \leq \text{轴的位置} \leq \text{LastPosition}$ 。即轴的位置仅在窗口有效范围内检测到外部中断信号时，才可以触发相对定位动作。

如果指令控制的轴为循环模式时，窗口有效范围为FirstPosition正方向旋转到LastPosition的区间。当 $\text{FirstPosition} \geq \text{该轴的模}$ ，或者 $\text{LastPosition} \geq \text{该轴的模}$ ，指令报错。

- 2) WindowMode为TRUE时，窗口模式为相对模式。

窗口模式为相对模式时，FirstPosition和LastPosition为相对位置，这两个位置的参考点为该指令Execute由FALSE变为TRUE时的轴位置。如该指令执行时的轴位置为100，FirstPosition和LastPosition分别设置为500和800，则轴位置在600和900之间时，检测到外部中断信号才有效，才可以触发相对定位动作。

窗口模式为相对模式时，如果指令控制的轴为线性模式，外部信号可以触发锁定轴位置的有效范围：（该指令执行时的轴位置+FirstPosition） $\leq$ 轴的位置 $\leq$ （该指令执行时的轴位置+LastPosition）。

窗口模式为相对模式时，如果指令控制的轴为循环模式时，窗口位置的计算方法如下：

**FirstPosition=MODABS(该指令执行时的轴位置+FirstPosition,模);**

**LastPosition =MODABS(该指令执行时的轴位置+FirstPosition,模);** 即新计算的位置对模取余。

**注：**

指令控制的轴为直线轴，当FirstPosition $\geq$ LastPosition时，指令报错。

指令控制的轴为旋转轴时，当FirstPositon $\geq$ 该轴的模，或者LastPosition $\geq$ 该轴的模，指令报错。

- **重启该指令**

当指令执行完成后（Done 变为 TRUE 时），Execute 再次由 FALSE 变为 TRUE 时，该指令可以重新执行；当指令正在执行中，Execute 再次由 FALSE 变为 TRUE 时，对指令的执行不会产生影响，指令仍按照未执行完成的输入变量执行指令。

- **其它指令执行时启动该指令**

其它运动指令执行时，启动该指令，可切换或缓存到该指令。该指令和其它运动指令如何中继由该指令输入变量 BufferMode（缓存模式）的值决定，而且该指令输入变量 BufferMode（缓存模式）可以设置的值和正在执行的运动指令有关。

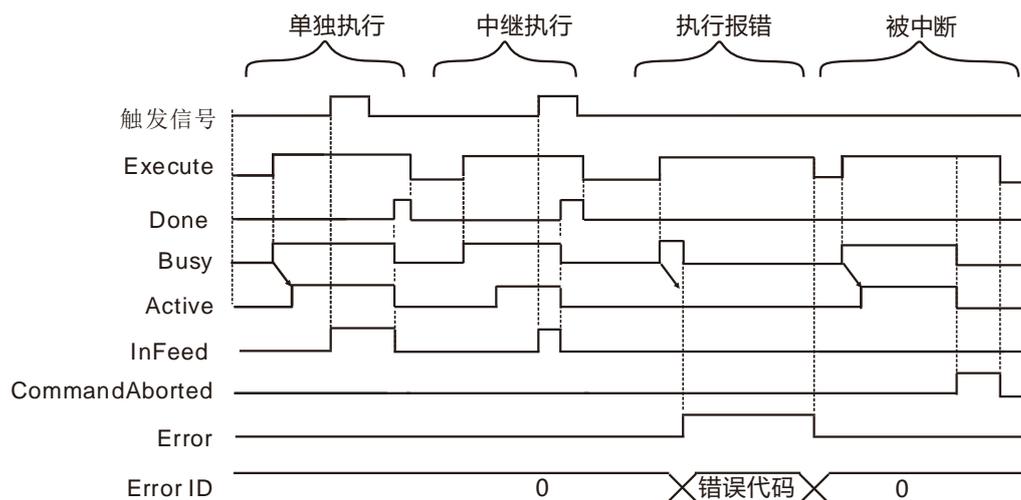
- **该指令执行时启动其它指令**

该指令执行时不建议执行其它指令，因为执行其它指令，如果其它指令选择打断该指令，该指令执行没有意义，如果其它指令和该指令交接，该指令外发触发信号触发时会打断后执行的指令。

- **异常排除**

该指令执行时，如果输入变量不合法，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。该指令执行时，如遇到异常（如轴报警等），该指令也会报错，轴会立即停止。

## ■ 输出变量变化时序说明



### ● 单独执行

Execute (启动) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE, 下一个周期 Active (控制中) 变为 TRUE。外部中断触发锁定位置时, InFeed 变为 TRUE, 外部中断触发相对定位完成或者没有外部中断时 MoveMode 指定的定位完成时, Done (完成) 变为 TRUE, Busy (执行中)、Active (控制中)、InFeed (中断触发) 同时变为 FALSE。当 Execute (启动) 变为 FALSE 时, Done (完成) 同时变为 FALSE。

### ● 中继执行

其它指令控制轴时, 执行该指令 (BufferMode 的值不为 mcAborting), 当 Execute (启动) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE, Active (控制中) 变为 TRUE 的时机为前一个指令完成时。

### ● 执行报错

该指令输入变量值不在允许范围内时, 该指令 Execute (启动) 由 FALSE 变 TRUE 时, 同时 Busy (执行中) 变为 TRUE, 下一个周期 Error (错误) 变为 TRUE, 同时 Busy (执行中) 变为 FALSE, ErrorID (错误码) 输出对应的错误码, 可通过 ErrorID (错误码) 的值, 查找发生问题原因。该指令 Execute (启动) 由 TRUE 变为 FALSE 时, 同时 Error (错误) 变为 FALSE, ErrorID (错误码) 的值变为 0。

### ● 被中断

该指令执行后被其它指令中断时, 该指令 CommandAborted (中断) 变为 TRUE, Busy (执行中) 和 Active (控制中) 同时变为 FALSE; 当 Execute (执行中) 变为 FALSE 时, CommandAborted (中断) 同时变为 FALSE。

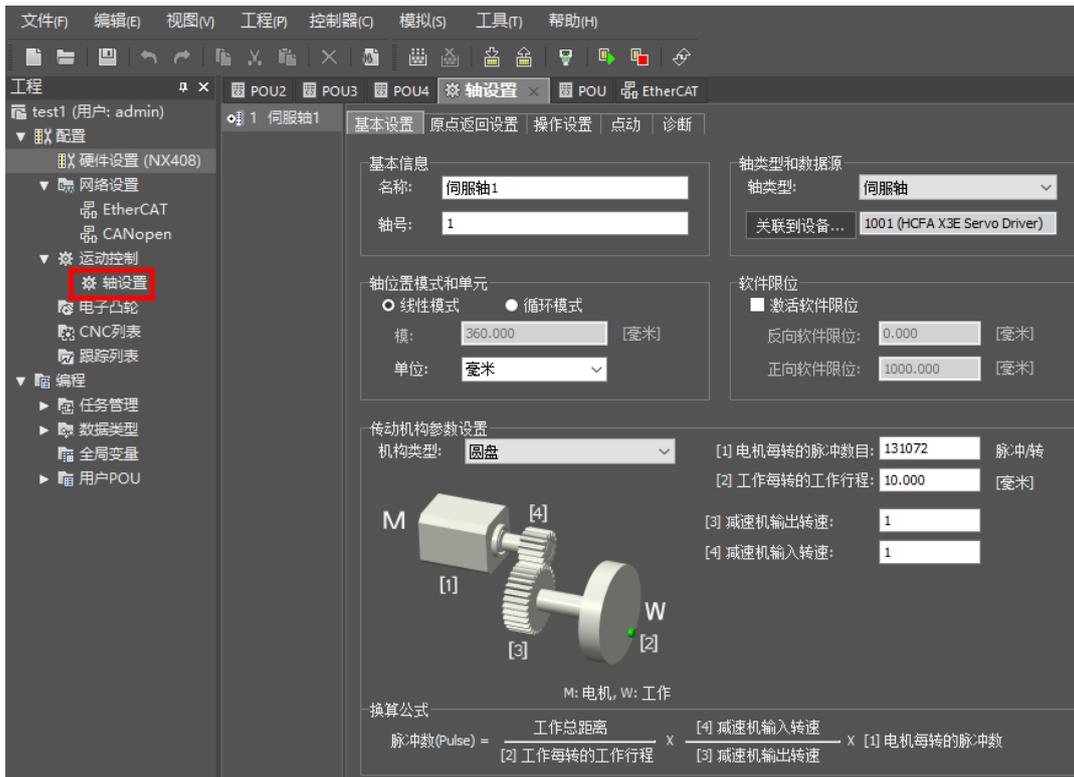
## ■ 示例程序

### ● 实现功能

指定轴先进行绝对值定位, 该指令控制轴位置超过 TouchPosition 设定的相对位置 500 时, 轴运动速度由高速切换为低速, 并以低速寻找外部中断信号(外部中断信号接驱动器的输入点), 检测到外部中断输入后, 以中断输入信号抓取轴的实际位置为参考点, 再移动 FeedDistance 指定的相对距离, 移动完指定的相对距离后停止。

### ● 轴参数设置

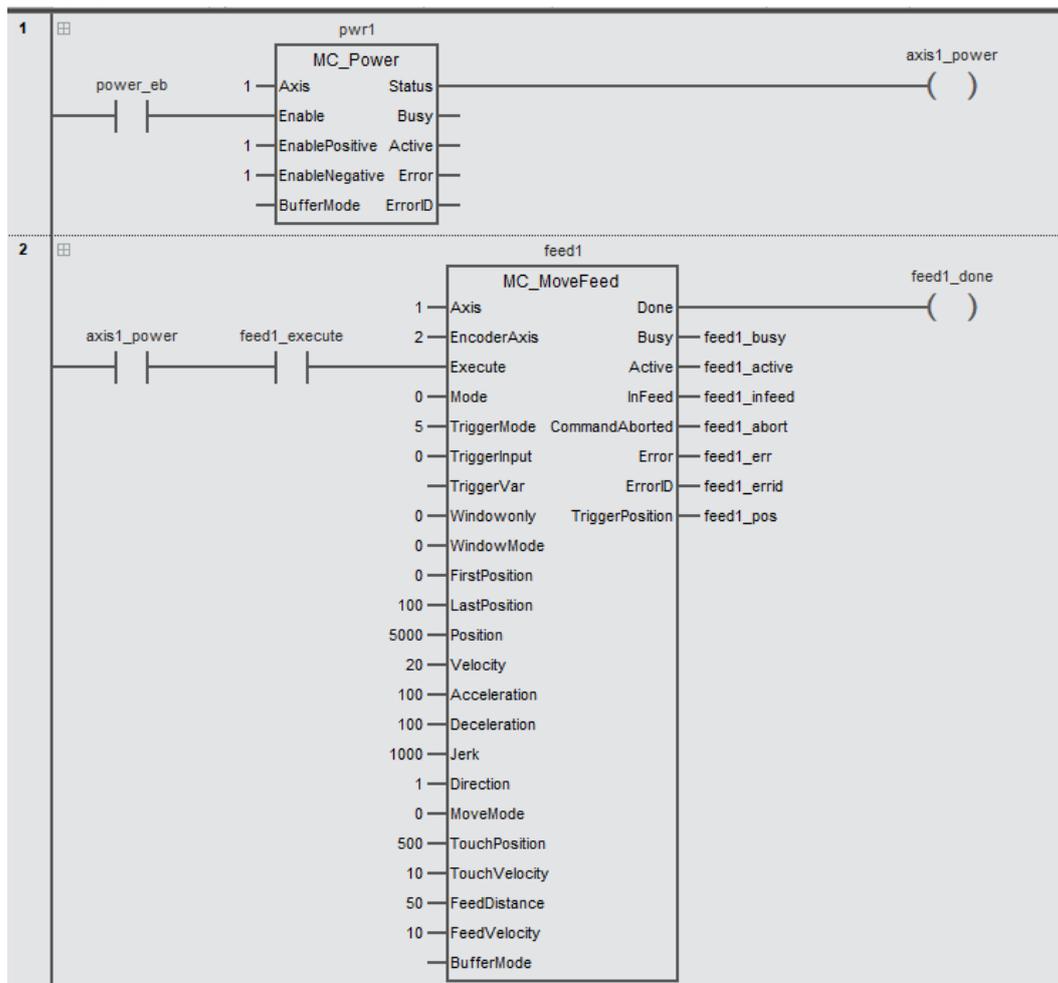
轴 1 轴参数设置如下图所示。



● 变量表

类别	名称	分配到	数据类型	初始值	注释
VAR	power_eb		BOOL		
VAR	power1		MC_Power		
VAR	axis1_power		BOOL		
VAR	feed1_execute		BOOL		
VAR	feed1		MC_MoveFeed		
VAR	捕获触发		BOOL		
VAR	feed1_done		BOOL		
VAR	feed1_busy		BOOL		
VAR	feed1_active		BOOL		
VAR	feed1_infeed		BOOL		
VAR	feed1_abort		BOOL		
VAR	feed1_err		BOOL		
VAR	feed1_errid		WORD		
VAR	feed1_pos		LREAL		

● 梯形图(LD)



● 结构化文本(ST)

```
power1 (
    Axis:=1 ,
    Enable:=power_eb ,
    EnablePositive:=TRUE ,
    EnableNegative:=TRUE ,
    BufferMode:=mcAborting ,
    Status=>axis1_power
);
```

```
feed1(Axis:= 1,
    EncoderAxis:=2 ,
    Execute:= axis1_power AND feed1_execute,
    Mode:=0 ,
    TriggerMode:=5 ,
    TriggerInput:=0 ,
    Windowonly:=0 ,
    WindowMode:=0 ,
    FirstPosition:=0 ,
    LastPosition:=0 ,
    Position:=5000 ,
    Velocity:=20 ,
    Acceleration:=100 ,
    Deceleration:=100,
    Jerk:= 100,
```

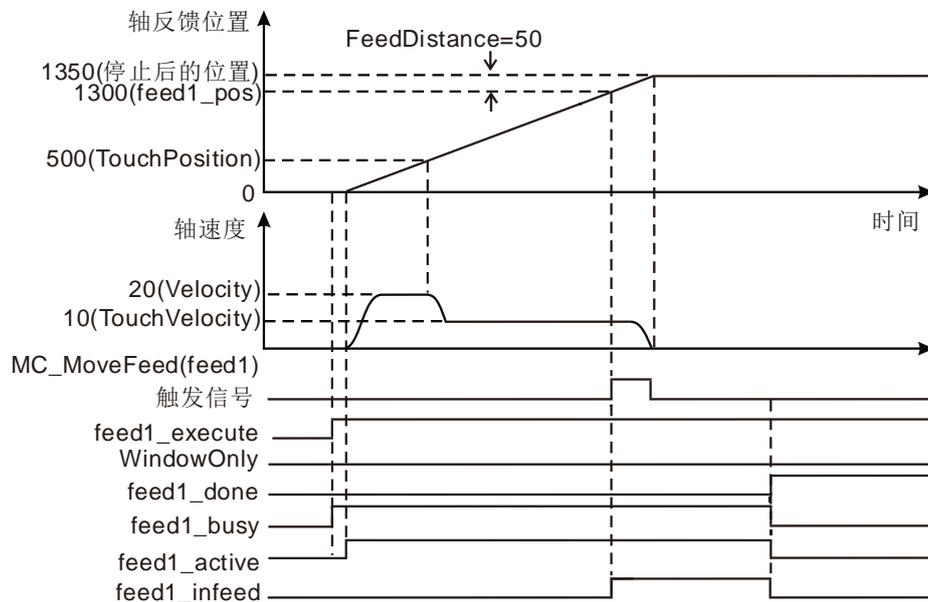
```

Direction:= 1,
MoveMode:= 0,
TouchPosition:= 500,
TouchVelocity:=10,
FeedDistance:= 50,
FeedVelocity:=10,
BufferMode:= mcAborting,
Done=> feed1_done,
Busy=> feed1_busy,
Active=> feed1_active,
InFeed=> feed1_infeed,
CommandAborted=> feed1_abort,
Error=> feed1_err,
ErrorID=> feed1_errid,
TriggerPosition=> feed1_pos
);

```

### ● 程序说明

- 轴使能后，轴的当前位置为0。Mode设置为0（外部中断输入锁定轴位置），MoveMode设置为0（绝对值定位方式），feed1\_execute由FALSE变为TRUE时，轴开始根据Position, Velocity, Acceleration, Deceleration, Jerk设定的参数进行运动，并计算该指令控制轴走过的相对位置。
- 该指令以开始执行时轴命令位置为参考位置，当该指令控制轴位置超过TouchPosition设定的相对位置500时，轴目标速度由20降到10（由Velocity设定的速度变为TouchVelocity设定的速度），并在低速状态下寻找外部中断信号。当检测到外部中断信号的上升沿时，锁定指定轴的位置，feed1\_infeed变为TRUE，feed1\_pos输出锁定的轴位置，轴再根据FeedDistance和FeedVelocity进行相对定位。定位完成后，轴停止运动，feed1\_Done变为TRUE，feed1\_busy、feed1\_active、feed1\_infeed同时变为FALSE。feed1\_execute变为FALSE后，同周期feed1\_done变为FALSE，feed1\_pos保持外部中断信号锁定轴的位置，不会清零。



## 4.26 MC\_ReadStatus (读取轴状态)

该指令用于周期性读取控制器内配置轴的状态信息。所属库：MotionControl

指令	名称	FB/FUN	梯形图样式	ST样式
MC_ReadStatus	读取轴状态	FB		<pre> MC_ReadStatus_Instance ( Axis := 参数, Enable := 参数, Valid =&gt; 参数, Busy =&gt; 参数, Error =&gt; 参数, ErrorID =&gt; 参数, ErrorStop =&gt; 参数, Disabled =&gt; 参数, Stopping =&gt; 参数, Homing =&gt; 参数, Standstill =&gt; 参数, DiscreteMotion =&gt; 参数, ContinuousMotion =&gt; 参数, SyncMotion =&gt; 参数 ); </pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Enable	启动	BOOL	TRUE或FALSE	FALSE	设为TRUE，读取指定轴状态。 设为FALSE，停止读取指定轴状态

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Valid	完成	BOOL	TRUE或FALSE	该指令正常执行时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”
ErrorStop	错误停止	BOOL	TRUE或FALSE	执行运动指令时，轴状态会根据执行的指令进行变化，这些输出位用于显示指定轴当前的轴状态
Disabled	解除运行	BOOL	TRUE或FALSE	
Stopping	减速停止	BOOL	TRUE或FALSE	
Homing	原点寻找	BOOL	TRUE或FALSE	
Standstill	停止	BOOL	TRUE或FALSE	
DiscreteMotion	定位动作中	BOOL	TRUE或FALSE	
ContinuousMotion	连续动作中	BOOL	TRUE或FALSE	
SyncMotion	同步动作中	BOOL	TRUE或FALSE	

## ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Valid	Enable 为TRUE时	<ul style="list-style-type: none"> <li>◆ Enable变为FALSE时</li> <li>◆ Error变为TRUE时</li> </ul>
Busy	Enable 为TRUE时	<ul style="list-style-type: none"> <li>◆ Enable变为FALSE时</li> <li>◆ Error为TRUE时</li> </ul>
Error	指令输入变量值不在允许的范围内时	◆ Error为TRUE, Enable由TRUE变为FALSE时
ErrorStop	轴状态变为ErrorStop状态时	◆ 轴状态不为ErrorStop状态时
Disabled	轴状态变为Disabled状态时	◆ 轴状态不为Disabled状态时
Stopping	轴状态变为Stopping状态时	◆ 轴状态不为Stopping状态时
Homing	当轴状态变为Homing状态时	◆ 轴状态不为Homing状态时
Standstill	当轴状态处于Standstill状态时	◆ 轴状态不为Standstill状态时
DiscreteMotion	轴状态变为DiscreteMotion状态时	◆ 轴状态不为DiscreteMotion状态
ContinuousMotion	当轴状态变为ContinuousMotion状态时	◆ 轴状态不为ContinuousMotion状态时
SyncMotion	当轴状态变为SyncMotion状态时	◆ 轴状态不为SyncMotion状态时

## ■ 功能说明

### ● 基本功能说明

- 该指令用于周期性读取控制器内配置轴的状态信息，输入变量Enable为TRUE时有效。输出变量Busy为TRUE表示该指令正在执行，Valid为TRUE表示输出变量对应的轴状态有效。
- 轴状态和执行的运动指令及驱动器的状态有关，运动指令和轴状态的对应关系的详细介绍请参考“运动控制相关→状态机”章节。

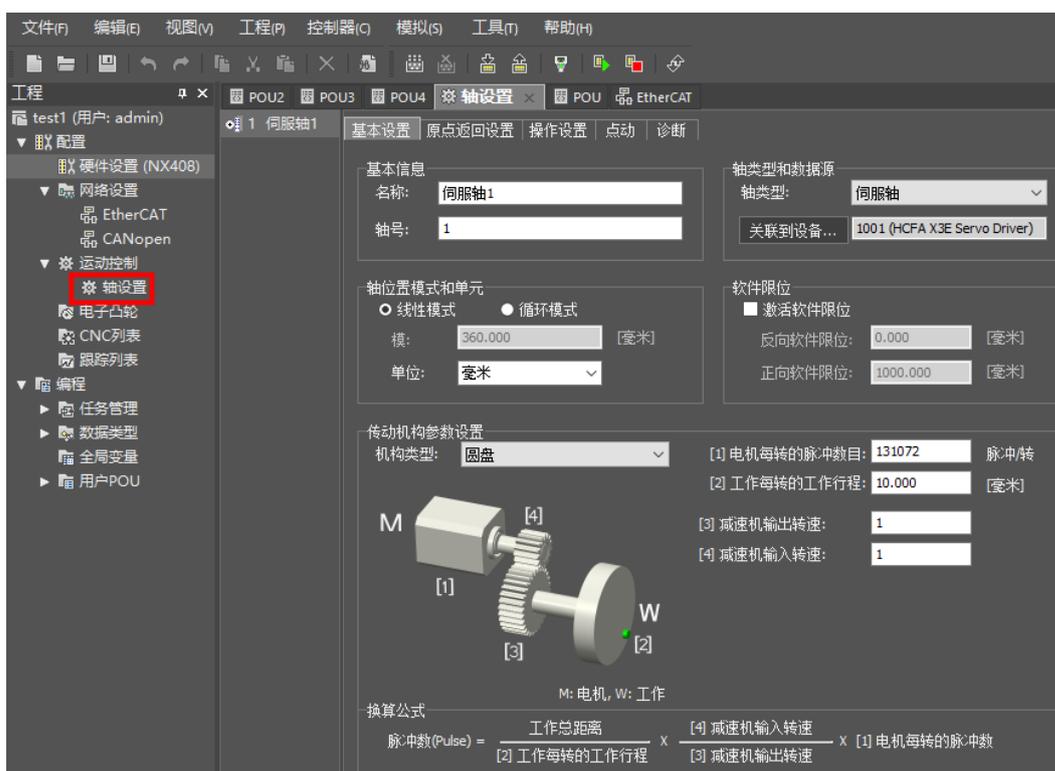
## ■ 示例程序

### ● 实现功能

相对位移指令执行时，通过 MC\_ReadStatus（读取轴状态）指令读取轴状态。

### ● 轴参数设置

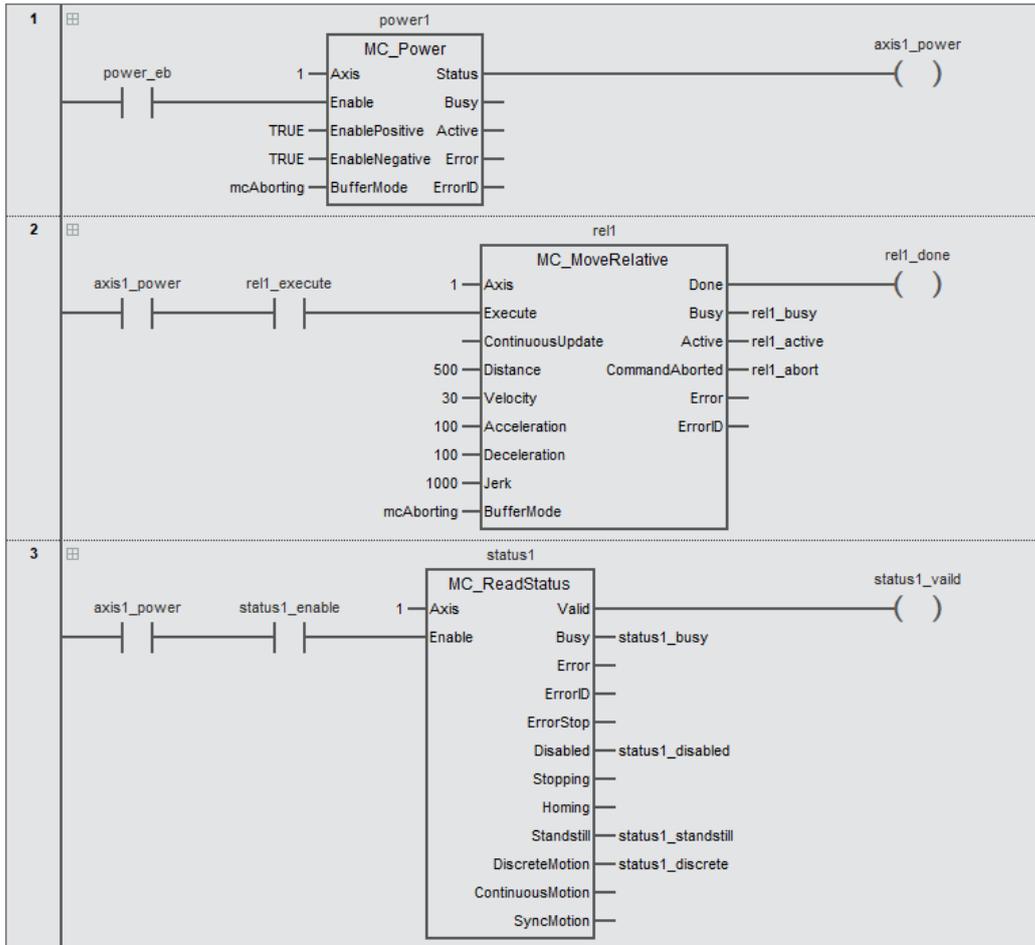
轴 1 轴参数设置如下图所示。



● 变量表

类别	名称	分配到	数据类型	初始值	注释
VAR	power_eb		BOOL		
VAR	power1		MC_Power		
VAR	axis1_power		BOOL		
VAR	rel1_execute		BOOL		
VAR	rel1		MC_MoveRelative		
VAR	rel1_done		BOOL		
VAR	rel1_busy		BOOL		
VAR	rel1_active		BOOL		
VAR	rel1_abort		BOOL		
VAR	status1		MC_ReadStatus		
VAR	status1_enable		BOOL		
VAR	status1_vaiId		BOOL		
VAR	status1_busy		BOOL		
VAR	status1_disabled		BOOL		
VAR	status1_standstill		BOOL		
VAR	status1_discrete		BOOL		

● 梯形图(LD)



● 结构化文本(ST)

```

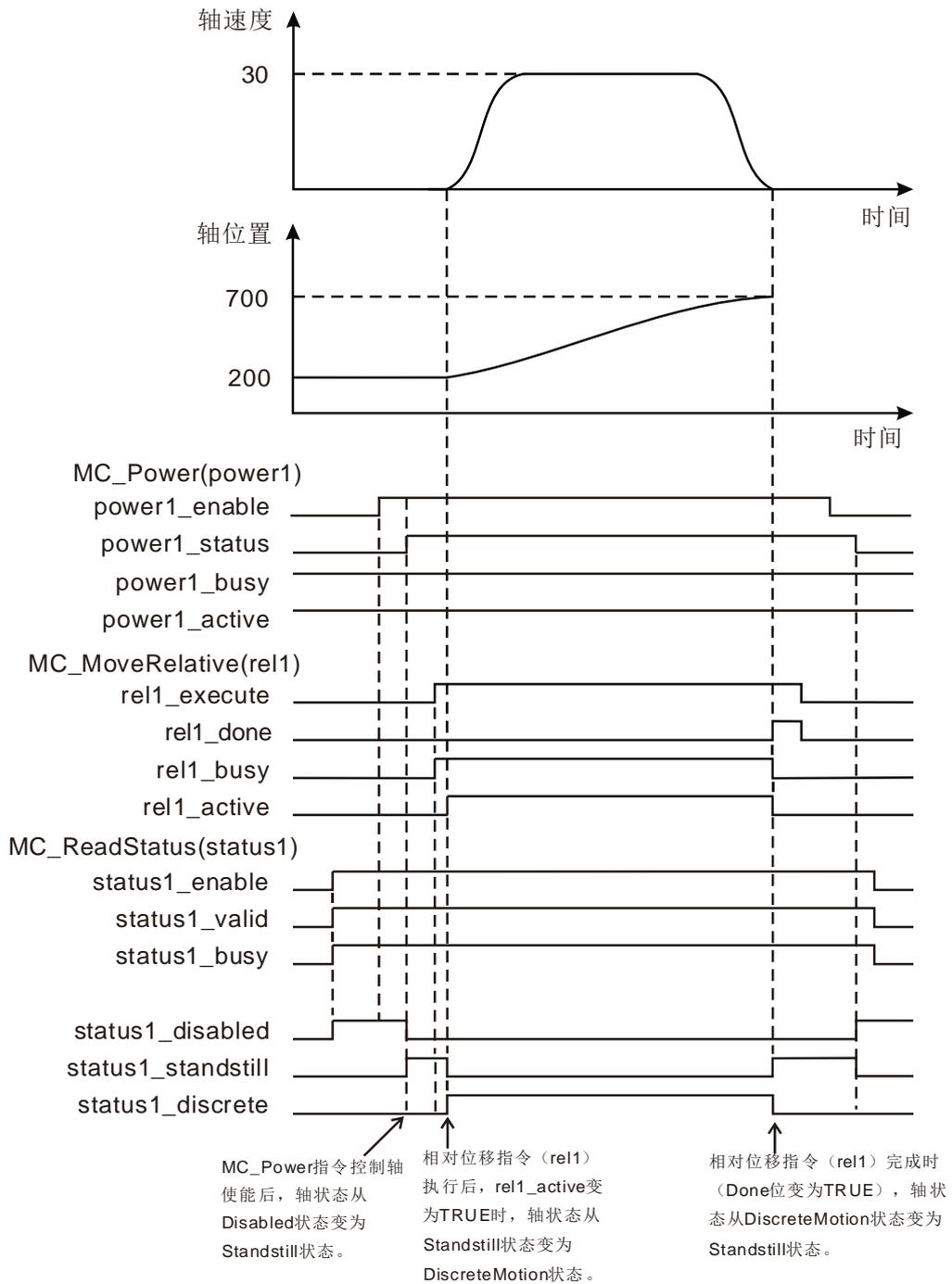
power1(
  Axis:=1 ,
  Enable:=power_eb ,
  EnablePositive:=TRUE ,
  EnableNegative:=TRUE ,
  BufferMode:=mcAborting ,
  Status=>axis 1_power
);

rel1(
  Axis:=1 ,
  Execute:=axis1_power AND rel1_execute ,
  Distance:=500 ,
  Velocity:=30 ,
  Acceleration:=100 ,
  Deceleration:=100 ,
  Jerk:=1000 ,
  BufferMode:=mcAborting ,
  Done=>rel1_done ,
  Busy=>rel1_busy ,
  Active=>rel1_active ,
  CommandAborted=>rel1_abort
);
    
```

```
status(  
  Axis:=1 ,  
  Enable:= axis1_power AND status_enable ,  
  Valid=> status_vaild,  
  Busy=> status_busy ,  
  Disabled=> status_disabled ,  
  Standstill=> status_standstill ,  
  DiscreteMotion=> status_discrete ,  
);
```

#### ● 程序说明

- status1\_enable为TRUE时，读取轴状态指令（status1）开始读取轴状态。该指令执行时，轴没有使能，轴状态为disabled状态，disabled状态位status1\_disabled从FALSE变为TRUE，其它状态位为FALSE。
- power1\_enable变为TRUE后，power1指令控制轴使能运行，power1\_status位变为TRUE时，轴状态从disabled状态变为standstill状态。disabled状态位status1\_disabled从TRUE变为FALSE，standstill状态位status1\_standstill从FALSE变为TRUE。
- rel1\_excute变为TRUE时，相对位移指令（rel1）执行，rel1\_active变为TRUE时，相对位移指令控制轴运转，轴状态从standstill状态变为discrete状态。standstill状态位status1\_standstill从TRUE变为FALSE，discrete状态位status1\_discrete从FALSE变为TRUE。
- 相对位移指令（rel1）执行完成时（Done变为TRUE），轴状态从DiscreteMotion状态变为standstill状态。standstill状态位status1\_standstill从FALSE变为TRUE，dDiscreteMotion状态位status1\_discrete从TRUE变为FALSE。
- power1\_enable变为FALSE后，power1指令控制轴解除运行，power1\_status位变为FALSE时，轴状态从standstill状态变为disabled状态。standstill状态位status1\_standstill从TRUE变为FALSE，disabled状态位status1\_disabled从FALSE变为TRUE。



## 4.27 MC\_ReadAxisError (轴异常读取)

该指令用于周期性读取指定轴的异常。所属库：MotionControl

指令	名称	FB/FUN	梯形图样式	ST样式
MC_ReadAxisError	轴异常读取	FB		<pre>MC_ReadAxisError_Instance ( Axis := 参数, Enable := 参数, Valid =&gt; 参数, Busy =&gt; 参数, Error =&gt; 参数, ErrorID =&gt; 参数, AxisErrorID =&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Enable	启动	BOOL	TRUE或FALSE	FALSE	设为TRUE, 读取指定轴错误码。 设为FALSE, 停止读取指定轴错误码

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Valid	完成	BOOL	TRUE或FALSE	该指令正常执行时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时, 输出错误代码 值的含义请参考“ <a href="#">指令错误代码描述</a> ”
AxisErrorID	轴异常代码	WORD	正数、0	轴异常时, 对应的异常代码详见 <a href="#">轴异常代码表</a>

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Valid	指定轴错误可以读取时	<ul style="list-style-type: none"> <li>◆ Enable变为FALSE时</li> <li>◆ Error变为TRUE时</li> </ul>
Busy	当Enable 为TRUE时	<ul style="list-style-type: none"> <li>◆ Enable变为FALSE时</li> <li>◆ Error为TRUE时</li> </ul>
Error	指令输入变量值不在允许的范围时。	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Enable由TRUE变为FALSE时</li> </ul>

### ■ 功能说明

#### ● 基本功能说明

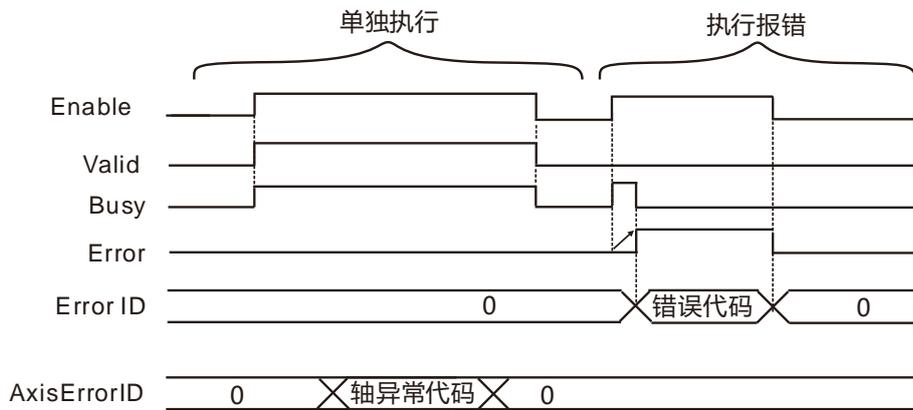
该指令用于周期读取指定轴的异常, 指定轴可以为伺服轴、虚拟伺服轴、脉冲轴、编码器轴。该指令输入参数 Enable 为 TRUE 时执行, 如轴命令位置和反馈位置差值超过位置误差的设定值时, 可以通过该指令读取, AxisErrorID 会输出对应的错误码, 通过该错误码查询轴异常原因。

AxisErrorID(轴异常代码)不为 0 时, 对应的数据含义如下表所示。

AxisErrorID(轴异常代码)		ErrorID值含义	处理方法
十六进制	十进制		
FF01	65281	轴命令位置和反馈位置的差值超过允许的设定值	检查轴配置界面中轴命令位置和反馈位置的差值的警告值是否合理

			检查轴是否报警
FF02	65282	EtherCAT网络状态异常	检查对应轴所在的EtherCAT网络是否和控制器建立连接
FF03	65283	轴位置超过设定的软极限位置	先执行MC_Reset指令，然后再反向运转，将轴位置运行至软极限范围内
FF04	65284	使用驱动器功能回原点时，控制器由运行状态变为停止状态	联系控制器技术人员分析原因
FF05	65285	驱动器报错（驱动器0x6041的位3（错误位）为ON）	根据驱动器的错误代码判断错误原因
FF06	65286	驱动器运转过程中去使能	联系控制器和驱动器技术人员分析原因
FF07	65287	驱动器内部命令位置和反馈位置的差值超过允许的设定值	联系驱动器技术人员分析原因

**输出变量变化时序说明**



● **单独执行**

Enable（有效）由 FALSE 变为 TRUE 时，Valid（有效）和 Busy（执行中）同时变为 TRUE。当 Enable（有效）为 TRUE 时，指定轴有异常时，AxisErrorID 会显示轴异常代码，轴异常解决时，AxisErrorID 的值会变为 0。

● **执行报错**

该指令输入变量值不在允许范围内时，该指令 Enable（有效）由 FALSE 变为 TRUE 时，同时 Busy（执行中）变为 TRUE，下一个周期 Error（错误）变为 TRUE，同时 Busy（执行中）变为 FALSE，ErrorID（错误码）输出对应的错误码，可通过 ErrorID（错误码）的值，查找发生问题原因。该指令 Enable（有效）由 TRUE 变为 FALSE 时，同时 Error（错误）变为 FALSE，ErrorID（错误码）的值变为 0。

## 4.28 MC\_ReadMotionState (轴运动状态读取)

该指令用于读取指定轴运动状态及运转方向。所属库：MotionControl

指令	名称	FB/FUN	梯形图样式	ST样式
MC_ReadMotionState	轴运动状态读取	FB		<pre>MC_ReadMotionState_Instance ( Axis := 参数 , Enable:= 参数 , Source:= 参数 , Valid =&gt; 参数 , Busy =&gt; 参数 , Error =&gt; 参数 , ErrorID=&gt; 参数 , ConstantVelocity=&gt; 参数 , Acceleration =&gt; 参数 , Deceleration =&gt; 参数 , DirectionPositive=&gt; 参数 , DirectionNegative =&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Enable	启动	BOOL	TRUE或FALSE	FALSE	设为TRUE, 读取指定轴的运动状态。 设为FALSE, 停止读取指定轴状态

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Valid	有效	BOOL	TRUE或FALSE	该指令正常执行时, 该参数值状态为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时, 输出错误代码 值的含义请参考“ <a href="#">指令错误代码描述</a> ”
ConstantVelocity	匀速状态	BOOL	TRUE或FALSE	轴运动状态处于匀速时为TRUE, 反之为FALSE
Acceleration	加速状态	BOOL	TRUE或FALSE	轴运动状态处于加速时为TRUE, 反之为FALSE
Deceleration	减速状态	BOOL	TRUE或FALSE	轴运动状态处于减速时为TRUE, 反之为FALSE
DirectionPositive	正向运转	BOOL	TRUE或FALSE	轴运转方向为正向时为TRUE, 反之为FALSE
DirectionNegative	反向运转	BOOL	TRUE或FALSE	轴运转方向为反向时为TRUE, 反之为FALSE

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Valid	指定轴输出变量对应的轴状态有效	Enable变为FALSE时 Error变为TRUE时
Busy	当Enable 为TRUE时	Enable变为FALSE时 Error为TRUE时
Error	指令输入变量值不在允许的范围时。	Error为TRUE, Enable由TRUE变为FALSE时
ConstantVelocity	轴运转状态处于匀速时	轴运转状态不处于匀速时
Acceleration	轴运转状态处于加速时	轴运转状态不处于加速时
Deceleration	轴运转状态处于减速时	轴运转状态不处于减速时

DirectionPositive	轴运转方向为正向时	轴运转方向不为正向时
DirectionNegative	轴运转方向为反向时	轴运转方向不为反向时

## ■ 功能说明

### ● 基本功能说明

- 该指令用于读取指定轴运动状态及运转方向，输入变量Enable为TRUE时有效。输出变量Busy为TRUE表示该指令正在执行，Valid为TRUE表示输出变量对应的轴状态有效。
- ConstantVelocity（匀速状态）、Acceleration（加速状态）、Deceleration（减速状态）为控制器通过轴命令速度和命令加速度判断，轴在运行状态下才可以判断，轴没有进入运行状态时，无法判断。
- DirectionPositive（正向运转）和DirectionNegative（反向运转）为控制器通过轴命令速度判断，轴在运行状态下才可以判断，轴没有进入运行状态时，无法判断。

## 4.29 MC\_GetFollowingStatus (读取位置误差状态)

该指令用于实时检测指定轴命令位置 and 实际位置的位置差是否超过设定值，如超过，输出标志位变为TRUE。所属库：MotionControl\_Part2

适用机种： M500S系列、M500系列

指令	名称	FB/FUN	梯形图样式	ST样式
MC_GetFollowingStatus	读取位置误差状态	FB		<pre>MC_GetFollowingStatus_Instance ( Axis := 参数 , Enable := 参数 , OutOfRange =&gt; 参数 , Busy =&gt; 参数 , Error =&gt; 参数 , ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Enable	启动	BOOL	TRUE或FALSE	FALSE	设为TRUE，读取位置误差范围； 设为FALSE，停止读取位置误差范围

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
OutOfRange	超出范围	BOOL	TRUE或FALSE	指定轴命令位置 and 实际位置的位置差超过的设定值时，该参数值变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

### ■ 输出变量刷新时机

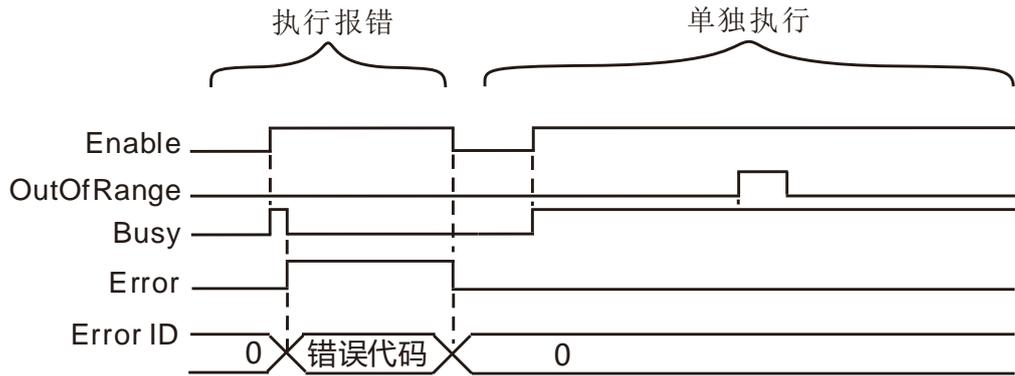
名称	变为TRUE的时机	变为FALSE的时机
OutOfRange	轴命令位置 and 实际位置的位置差超过设定值时	◆ 轴命令位置 and 实际位置的位置差未超过设定值时
Busy	Enable为TRUE时	◆ Error变为TRUE时
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	◆ Enable由TRUE变为FALSE时

### ■ 功能说明

#### ● 基本功能说明

该指令用于读取轴命令位置 and 实际位置差值及滤波时间是否超设定值。轴命令位置 and 实际位置差值超过该设定值后，轴会进入 errorstop 状态，执行 MC\_Reset 后，轴命令位置 and 实际位置会重新对位。轴命令位置 and 实际位置差值及滤波时间可以通过软件中”轴设置”中的”参数设定”进行设定，也可以通过指令 MC\_SetFollowingParm (位置误差参数设定) 设置的方式进行设定。

## ■ 输出变量变化时序说明



### ● 执行报错

该指令输入变量值不在允许范围内时，该指令 Enable（有效）由 FALSE 变为 TRUE 时，同时 Busy（执行中）变为 TRUE，下一个周期 Error（错误）变为 TRUE，同时 Busy（执行中）变为 FALSE，ErrorID（错误码）输出对应的错误码，可通过 ErrorID（错误码）的值，查找发生问题原因。该指令 Enable（有效）由 TRUE 变为 FALSE 时，同时 Error（错误）变为 FALSE，ErrorID（错误码）的值变为 0。

### ● 单独执行

Enable（有效）由 FALSE 变为 TRUE 时，Busy（执行中）同时变为 TRUE。当 Enable（有效）为 TRUE 时，轴命令位置和实际位置的位置差超过允许的范围时，OutOfRange 变为 TRUE；轴命令位置和实际位置的位置差未超过允许的范围时，OutOfRange 变为 FALSE。

## 4.30 MC\_SetFollowingParm (位置误差参数设定)

该指令用于设定指定轴命令位置 and 实际位置的允许的位置差及位置差持续的时间。所属库：MotionControl\_Part2  
适用机种：M500S系列、M500系列

指令	名称	FB/FUN	梯形图样式	ST样式
MC_SetFollowingParm	位置误差参数设定	FB		<pre>MC_SetFollowingParm_Instance ( Axis := 参数 , Execute := 参数 , Lag := 参数 , HoldTime := 参数 , Done =&gt; 参数 , Busy =&gt; 参数 , Error =&gt; 参数 , ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号。
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令。
Lag	位置误差	LREAL	正数	不可缺省	位置误差设定。
HoldTime	滤波时间	LREAL	正数	不可缺省	超过设定位置误差设定值的持续时间。(单位：秒)

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE。
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE。
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码。 值的含义请参考“ <a href="#">指令错误代码描述</a> ”。

### ■ 输出变量刷新时机

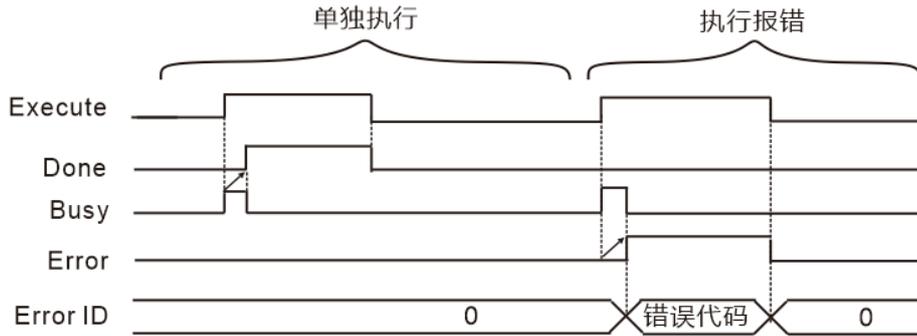
名称	变为TRUE的时机	变为FALSE的时机
Done	执行完成时。	<ul style="list-style-type: none"> <li>◆ Done为TRUE, Execute从TRUE变为FALSE时。</li> <li>◆ 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE。</li> </ul>
Busy	Execute的上升沿。	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时。</li> <li>◆ Error从FALSE变为TRUE时。</li> </ul>
Error	指令输入变量值不在允许的范围。	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Execute从TRUE变为FALSE时。</li> </ul>

### ■ 功能说明

#### ● 基本功能说明

该指令用于设定轴命令位置 and 实际位置差值的持续时间及允许值。滤波时间及允许值也可以通过软件中【轴设置】→【参数设定】进行设定。轴命令位置 and 实际位置差值超过该设定值后，轴会进入 errorstop 状态，执行 MC\_Reset 后，轴命令位置 and 实际位置会重新对位。可以通过 MC\_GetFollowingStatus 指令读取轴命令位置 and 实际位置差值是否超过该设定值。

## ■ 输出变量变化时序说明



### ● 单独执行

Execute (启动) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE, 下一个周期 Done (完成) 变为 TRUE, 同时 Busy (执行中) 变为 FALSE。当 Execute (启动) 变为 FALSE 时, Done (完成) 同时变为 FALSE。

### ● 执行报错

该指令输入变量值不在允许范围内时, 该指令 Execute (启动) 由 FALSE 变为 TRUE 时, 同时 Busy (执行中) 变为 TRUE, 下一个周期 Error (错误) 变为 TRUE, 同时 Busy (执行中) 变为 FALSE, ErrorID (错误码) 输出对应的错误码, 可通过 ErrorID (错误码) 的值, 查找发生问题原因。该指令 Execute (启动) 由 TRUE 变为 FALSE 时, 同时 Error (错误) 变为 FALSE, ErrorID (错误码) 的值变为 0。

## 4.31 MC\_EnableSoftLimit (软件限位激活设置)

该指令用于设定软件限位功能是否激活。所属库：MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_EnableSoftLimit	软件限位激活设置	FB		<pre>MC_EnableSoftLimit_Instance ( Axis := 参数 , Enable := 参数 , EnableLimit := 参数 , Done =&gt; 参数 , Error =&gt; 参数 , ErrorID =&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Execute	启动	BOOL	TRUE或FALSE	FALSE	设为TRUE, 该指令执行。可以激活或者取消激活软极限。设为FALSE, 该指令不执行。
EnableLimit	软极限是否激活	LREAL	TRUE或FALSE	FALSE	TRUE:激活 FALSE:取消激活

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时, 输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

### ■ 输出变量刷新时机

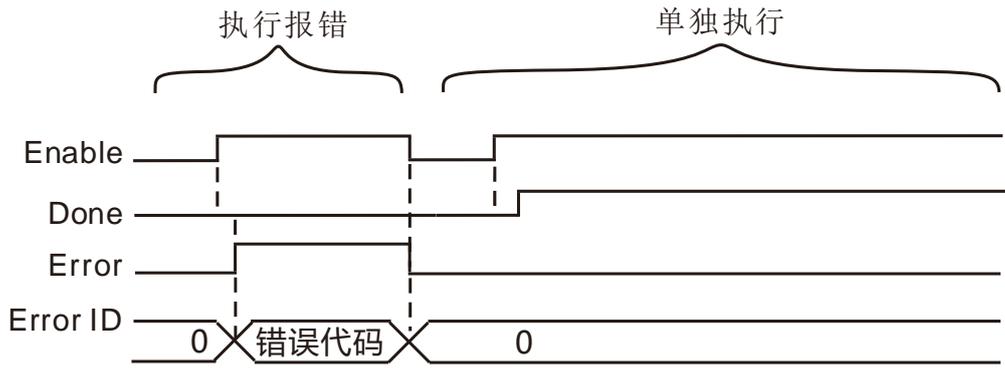
名称	变为TRUE的时机	变为FALSE的时机
Done	执行完成时	<ul style="list-style-type: none"> <li>◆ Done为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE</li> </ul>
Error	指令输入变量值不在允许的范围	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Execute从TRUE变为FALSE时</li> </ul>

### ■ 功能说明

#### ● 基本功能说明

该指令用于设置是否激活软件限位。软件限位可以通过软件设置, 也可以通过该指令设置及更改。无该指令时, 以软件设置为准。激活软件限位后, 轴位置超过正向软件限位或者反向软件限位后, 轴状态会进入ErrorStop (错误停止) 状态, 需要执行MC\_Reset复位清除。

### ■ 输出变量变化时序说明



#### ● 执行报错

该指令输入变量值不在允许范围内时，该指令 Enable（有效）由 FALSE 变为 TRUE 时，下一个周期 Error（错误）变为 TRUE，ErrorID（错误码）输出对应的错误码，可通过 ErrorID（错误码）的值，查找发生问题原因。该指令 Enable（有效）由 TRUE 变为 FALSE 时，同时 Error（错误）变为 FALSE，ErrorID（错误码）的值变为 0。

#### ● 单独执行

Enable（有效）由 FALSE 变为 TRUE 时，下一个周期，Done 变为 TRUE。

## 4.32 MC\_SetAxisParm (更改机构参数)

该指令用于变更执行机构参数。所属库: MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_SetAxisParm	更改机构参数	FB		<pre>MC_ChangeAxisParm_Instance ( Axis := 参数 , Execute := 参数 , ScaleDen := 参数 , ScaleNum := 参数 , UnitsPerTurn := 参数, AxisType:= 参数, Modulo:= 参数, Done =&gt;参数 , Busy =&gt; 参数 , Error =&gt; 参数 , ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
ScaleDen	工作齿轮减速比分子	LREAL	正整数	不可缺省	工作齿轮减速比分子
ScaleNum	工作齿轮减速比分母	LREAL	正整数	不可缺省	工作齿轮减速比分母
UnitsPerTurn	工作每转的工作行程	LREAL	正数	不可缺省	工作机构转一圈移动的行程单位数目 (单位: 行程单位)
AxisType	轴类型	USINT	0、1	0	轴类型 0: 旋转模式 1: 线性模式
Modulo	模	LREAL	正数	不可缺省	工作机构位置的周期

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时, 输出错误代码 值的含义请参考“ <a href="#">指令错误代码描述</a> ”

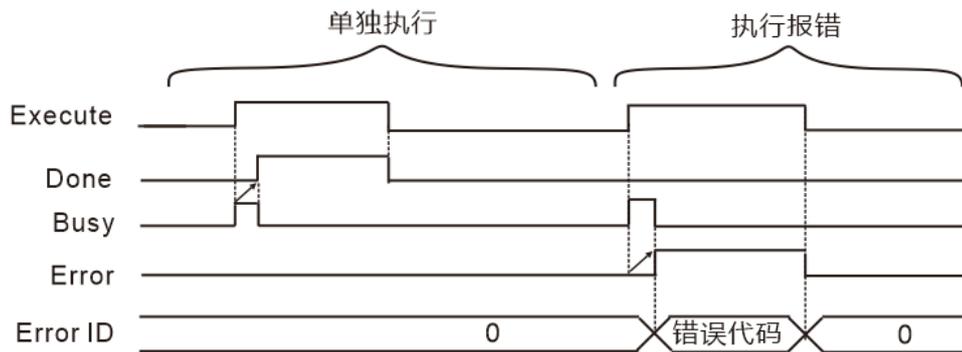
### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	执行完成时	<ul style="list-style-type: none"> <li>◆ Done为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE</li> </ul>
Busy	Execute的上升沿	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> </ul>
Error	指令输入变量值不在允许的范围	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Execute从TRUE变为FALSE时</li> </ul>

## ■ 功能说明

- 该指令用于变更终端机构参数。当轴的联动机构发生变更时，如减速机变更变化时，可以使用该指令更改轴参数和实际机构参数一致，方便用户使用。
- 该指令必须在状态机是 Disable 或者 Standstill 状态下执行，在其它状态下执行，指令报错。
- 控制器上电后，该指令须重新执行，才可以使用该指令设定的参数进行运算；控制器上电后，不执行该指令，按照软件【轴设置】→【基本设置】中的参数进行运算。
- 使用该指令时，需要熟悉指令每个参数的含义。否则可能会导致轴速度和预期不符，造成事故或者危险。
- 使用该指令后，请仔细检查程序中的运动指令的速度和位置是否需要更改，确认无误后，再运行程序。

## ■ 输出变量变化时序说明

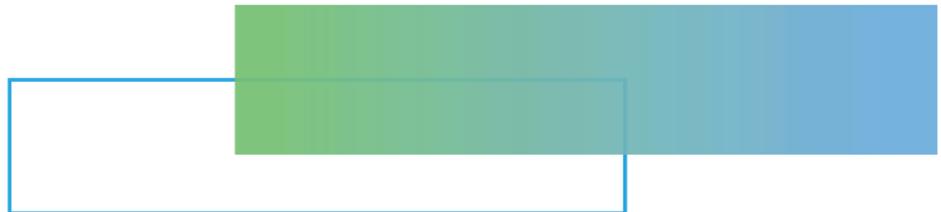


### ● 单独执行

Execute (启动) 由 FALSE 变为 TRUE 时，Busy (执行中) 同时变为 TRUE，下一个周期 Done (完成) 变为 TRUE，同时 Busy (执行中) 变为 FALSE。当 Execute (启动) 变为 FALSE 时，Done (完成) 同时变为 FALSE。

### ● 执行报错

该指令输入变量值不在允许范围内时，该指令 Execute (启动) 由 FALSE 变为 TRUE 时，同时 Busy (执行中) 变为 TRUE，下一个周期 Error (错误) 变为 TRUE，同时 Busy (执行中) 变为 FALSE，ErrorID (错误码) 输出对应的错误码，可通过 ErrorID (错误码) 的值，查找发生问题原因。该指令 Execute (启动) 由 TRUE 变为 FALSE 时，同时 Error (错误) 变为 FALSE，ErrorID (错误码) 的值变为 0。



## 第五章 多轴指令



## 5.1 MC\_GearIn (电子齿轮)

根据从轴与主轴之间的齿轮比，进行齿轮动作。所属库：MotionControl

指令	名称	FB/FUN	梯形图样式	ST样式
MC_GearIn	电子齿轮	FB		<pre> MC_GearIn_Instance (   Master :=参数 ,   Slave:=参数 ,   Execute :=参数 ,   ContinuousUpdate:= 参数 ,   RatioNumerator:= 参数 ,   RatioDenominator:= 参数 ,   MasterValueSource:= 参数 ,   Acceleration:= 参数 ,   Deceleration:= 参数 ,   Jerk:= 参数 ,   BufferMode:= 参数 ,   InGear=&gt; 参数 ,   Busy =&gt; 参数 ,   Active=&gt; 参数 ,   CommandAborted =&gt; 参数 ,   Error =&gt; 参数 ,   ErrorID=&gt; 参数 );                     </pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Master	主轴轴号	USINT	由机种决定	不可缺省	指定电子齿轮中主轴的轴号
Slave	从轴轴号	USINT	由机种决定	不可缺省	指定电子齿轮中从轴的轴号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
ContinuousUpdate	持续更新	BOOL	TRUE或FALSE	FALSE	保留
RatioNumerator	电子齿轮比分子	LREAL	正数、负数	不可缺省	电子齿轮比分子
RatioDenominator	电子齿轮比分母	LREAL	正数	不可缺省	电子齿轮比分母
MasterValueSource	主轴位置类型设定	MC_Source	0:mcSetValue 1:mcActualValue	0	设定从轴跟随主轴命令位置或者反馈位置。 0: 从轴跟随主轴的命令位置 1: 从轴跟随主轴的实际位置
Acceleration	加速度	LREAL	正数	不可缺省	从轴和主轴啮合时从轴的加速度*1 (单位: 行程单位/秒 <sup>2</sup> ) *2
Deceleration	减速度	LREAL	正数	不可缺省	从轴和主轴啮合时从轴的减速度*1 (单位: 行程单位/秒 <sup>2</sup> ) *2
Jerk	跃度	LREAL	正数	不可缺省	从轴和主轴啮合时从轴的跃度*1 (单位: 行程单位/秒 <sup>3</sup> ) *2
BufferMode	缓存模式	MC_Buffer_Mode	0: mcAborting 1: mcBuffered	0	设定两个指令之间缓存模式*3 0: 中断 1: 等待

\*1: Velocity、Acceleration、Deceleration、Jerk的关系请参考“[运动控制指令相关参数说明](#)”章节。

\*2: 指令单位的详细介绍请参考“[运动控制指令参数单位](#)”章节。

\*3: BufferMode的详细介绍请参考“[运动控制指令的多重启动时缓存模式说明 \(BufferMode\)](#)”章节。

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
InGear	齿轮比到达	BOOL	TRUE或FALSE	指定从轴到达目标速度时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Active	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

## ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
InGear	从轴命令速度到达目标速度时	<ul style="list-style-type: none"> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>
Busy	Execute 由FALSE变为TRUE时	<ul style="list-style-type: none"> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>
Active	指令开始控制轴时	<ul style="list-style-type: none"> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>
CommandAborted	指令的执行被中断时	<ul style="list-style-type: none"> <li>◆ CommandAborted为TRUE，Execute由TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时，该指令被其它指令中断时，CommandAborted变为TRUE一个周期后，变为FALSE</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Execute由TRUE变为FALSE时</li> </ul>

## ■ 功能说明

### ● 基本功能说明

- 该指令执行后，从轴按照该指令输入变量设定值与主轴进行齿轮动作。从轴可以选择跟随主轴命令位置或者反馈位置进行齿轮动作，通过输入变量MasterValueSource的值进行选择。主轴轴号可以设定为伺服轴、虚拟伺服轴、脉冲轴、编码器轴，从轴轴号可以设定为伺服轴、虚拟伺服轴、脉冲轴。
- 该指令执行时，从轴状态须在使能状态下才可以执行，主轴状态在使能或去使能状态下都可以执行。
- 主轴和从轴解除凸轮关系时通过执行MC\_GearOut指令解除，解除后从轴保持解除关系时的从轴速度。
- 主轴和从轴啮合时可以通过MC\_Halt或者MC\_Stop指令停止从轴。
- 该指令执行后，两个轴在建立电子齿轮关系过程中（如主轴运转时执行该指令），从轴没有达到目标速度时（该指令的InGear为FALSE时），从轴按照该指令设定的Acceleration（加速度）、Deceleration（减速度）、Jerk（跃度）加速或减速到达目标速度。从轴的目标速度如下式所示。该过程中主轴和从轴没有同步，加速过程中造成的主从轴同步误差不会自动补偿。
- 该指令执行后，两个轴建立电子齿轮关系后（该指令的InGear为TRUE时），从轴速度、加速度、减速度和主轴速度、加速度、减速度的关系如下式所示：

$$\text{从轴目标速度} = \text{主轴速度} \times \frac{\text{电子齿轮比分子}}{\text{电子齿轮比分母}}$$

$$\text{从轴加(减)速度} = \text{主轴加(减)速度} \times \frac{\text{电子齿轮比分子}}{\text{电子齿轮比分母}}$$

$$\text{电子齿轮比} = \frac{\text{电子齿轮比分子}}{\text{电子齿轮比分母}}$$

➤ 电子齿轮比为正数时，从轴和主轴的运动方向相同。电子齿轮比为负数时，从轴和主轴的运动方向相反。

#### ● 重启该指令

当指令正在执行中，Execute 由 FALSE 变为 TRUE 时，该指令可以重新执行；此时能够重新生效的引脚参数包括 RatioNumerator、RatioDenominator、Velocity、Acceleration、Deceleration、Jerk，其它引脚参数不会生效。当该指令与其它运动指令产生 BufferMode 关系时，改变该指令的参数，重新触发后该令的参数会生效，原来的交接关系仍然保持，交接速度会重新计算。

#### ● 其它指令执行时启动该指令

其它运动指令执行时，启动该指令，该指令和其它运动指令如何中继由该指令输入变量 BufferMode（缓存模式）的值决定。该指令输入变量 BufferMode（缓存模式）可以设置的值只可以设置为 mcAborting（中断）或者 mcBuffered（等待）。

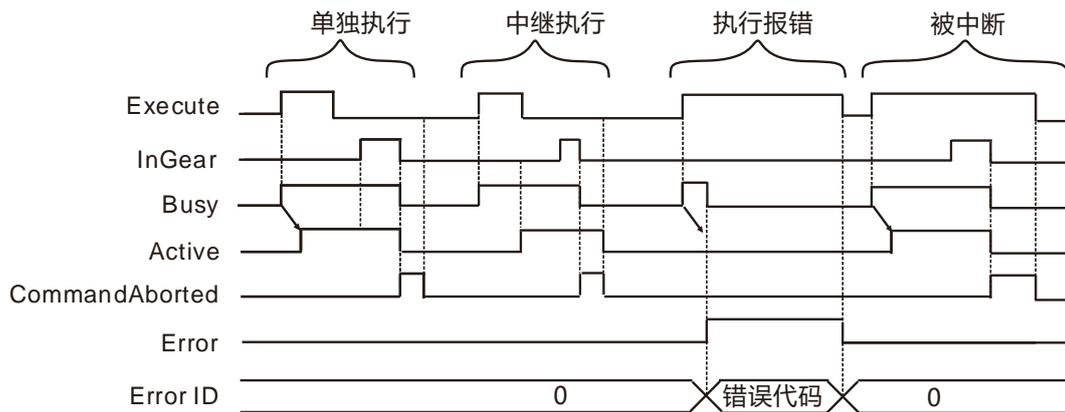
#### ● 该指令执行时启动其它指令

该指令执行时启动其它运动指令，其它运动指令和该指令如何中继由其它指令的 BufferMode（缓存模式）引脚参数值决定，其它运动指令 BufferMode 的值只能选择中断或者等待；如果其它运动指令无 BufferMode 参数，一般为中断该指令。其它运动指令和该指令缓存时，其它指令执行的时机为该指令“InGear(齿轮比到达)”变为 TRUE 时。

#### ● 异常排除

该指令执行时，如果输入变量不合法，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。该指令执行时，如遇到异常（如轴报警等），该指令也会报错，轴会立即停止。

#### ■ 输出变量变化时序说明



- **单独执行**

Execute (启动) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE, 下一个周期 Active (控制中) 变为 TRUE。当从轴命令速度达到目标速度时, InGear 变为 TRUE, Busy (执行中) 和 Active (控制中) 仍为 TRUE。

- **中继执行**

其它指令控制轴时, 执行该指令 (BufferMode 的值不为 mcAborting), 当 Execute (启动) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE, Active (控制中) 变为 TRUE 的时机为前一个指令完成时。

- **执行报错**

该指令输入变量值不在允许范围内时, 该指令 Execute (启动) 由 FALSE 变为 TRUE 时, 同时 Busy (执行中) 变为 TRUE, 下一个周期 Error (错误) 变为 TRUE, 同时 Busy (执行中) 变为 FALSE, ErrorID (错误码) 输出对应的错误码, 可通过 ErrorID (错误码) 的值, 查找发生问题原因。该指令 Execute (启动) 由 TRUE 变为 FALSE 时, 同时 Error (错误) 变为 FALSE, ErrorID (错误码) 的值变为 0。

- **被中断**

该指令执行后被其它指令中断时, 该指令 CommandAborted (中断) 变为 TRUE, Busy (执行中)、Active (控制中)、InGear (从轴命令速度达到目标速度) 同时变为 FALSE; 当 Execute (执行中) 变为 FALSE 时, CommandAborted (中断) 同时变为 FALSE。

## ■ 示例程序

- **实现功能**

电子齿轮比分子变化时, 修改电子齿轮比分子的值, 实现对从轴速度进行实时调速。两个电子齿轮指令, 检测到电子齿轮比变化时, 两个指令交替执行。

- **轴参数设置**

主轴轴 1 和从轴轴 2 的轴参数相同, 轴 1 轴参数设置如下图所示。

文件(F) 编辑(E) 视图(V) 工程(P) 控制器(C) 模拟(S) 工具(T) 帮助(H)

工程 test1 (用户: admin) POU1 POU2 POU3 POU4 轴设置 POU EtherCAT

配置

硬件设置 (NX408)

网络设置

EtherCAT

CANopen

运动控制

轴设置

电子凸轮

CNC列表

跟踪列表

编程

任务管理

数据类型

全局变量

用户POU

伺服轴1 基本设置 原点返回设置 操作设置 点动 诊断

基本信息

名称: 伺服轴1

轴号: 1

轴类型和数据源

轴类型: 伺服轴

关联到设备... 1001 (HCFA X3E Servo Driver)

轴位置模式和单元

线性模式  循环模式

模: 360.000 [毫米]

单位: 毫米

软件限位

激活软件限位

反向软件限位: 0.000 [毫米]

正向软件限位: 1000.000 [毫米]

传动机构参数设置

机构类型: 圆盘

[1] 电机每转的脉冲数: 131072 脉冲/转

[2] 工作每转的工作行程: 10.000 [毫米]

[3] 减速机输出转速: 1

[4] 减速机输入转速: 1

M: 电机, W: 工作

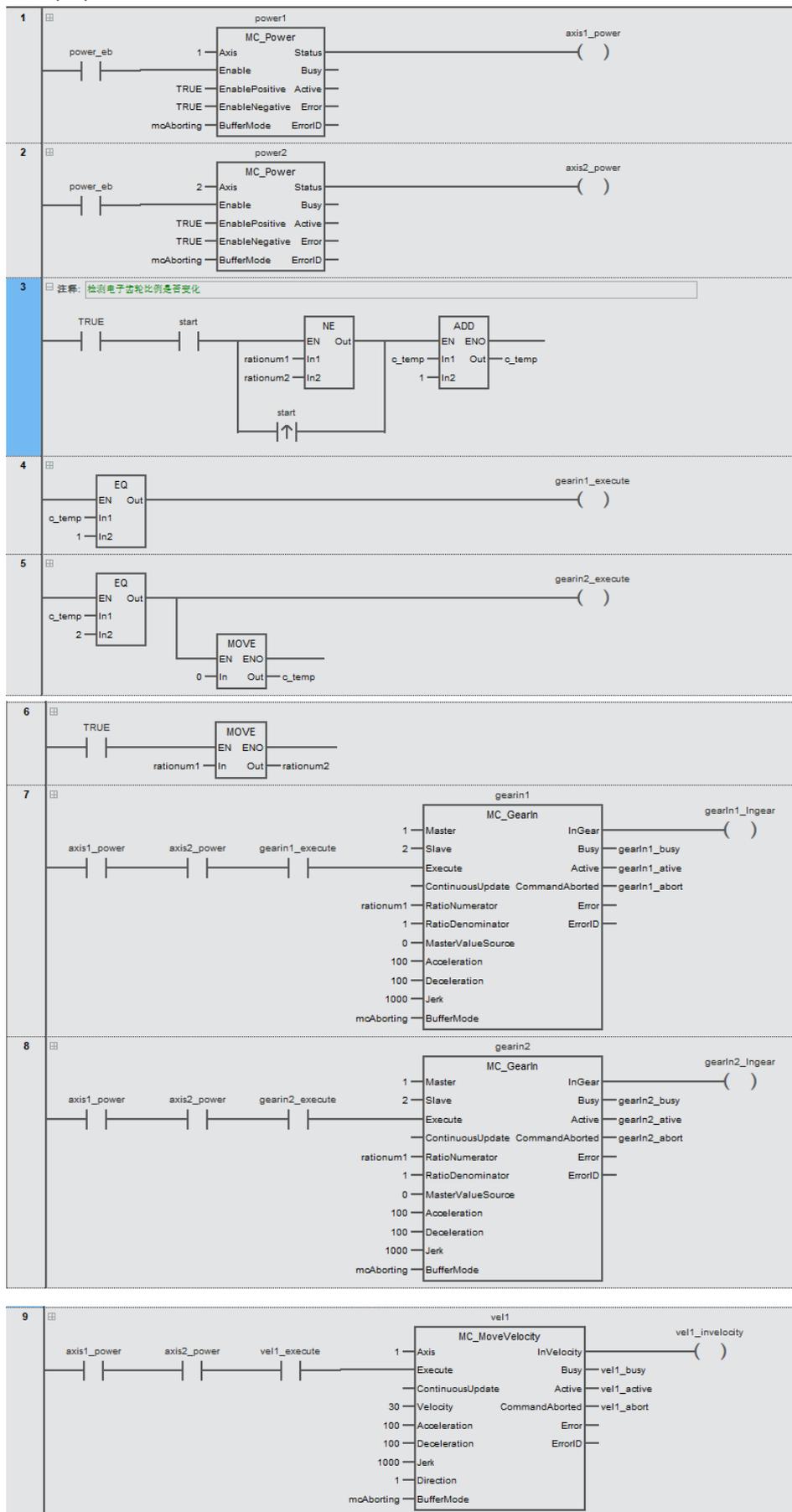
换算公式

$$\text{脉冲数(Pulse)} = \frac{\text{工作总距离}}{[2] \text{ 工作每转的工作行程}} \times \frac{[4] \text{ 减速机输入转速}}{[3] \text{ 减速机输出转速}} \times [1] \text{ 电机每转的脉冲数}$$

## ● 变量表

类别	名称	分配到	数据类型	初始值	注释
VAR	power_eb		BOOL		
VAR	power1		MC_Power		
VAR	axis1_power		BOOL		
VAR	power2		MC_Power		
VAR	axis2_power		BOOL		
VAR	gearin1		MC_GearIn		
VAR	gearIn1_Ingear		BOOL		
VAR	gearIn1_busy		BOOL		
VAR	gearIn1_ative		BOOL		
VAR	gearIn1_abort		BOOL		
VAR	vel1		MC_MoveVelocity		
VAR	vel1_execute		vel1_invelocity		
VAR	vel1_busy		BOOL		
VAR	vel1_active		BOOL		
VAR	vel1_abort		BOOL		
VAR	rationum1		LREAL		
VAR	rationum2		LREAL		
VAR	start		BOOL		
VAR	gearin2		MC_GearIn		
VAR	gearIn2_Ingear		BOOL		
VAR	gearIn2_busy		BOOL		
VAR	gearIn2_ative		BOOL		
VAR	gearIn2_abort		BOOL		
VAR	gearin1_execute		BOOL		
VAR	gearin2_execute		BOOL		
VAR	c_temp		UINT		

● 梯形图(LD)



- 结构化文本(ST)

```

power1(
  Axis:=p1 ,
  Enable:=power_eb ,
  EnablePositive:=TRUE ,
  EnableNegative:=TRUE ,
  BufferMode:=mcAborting ,
  Status=>axis1_power
);

power2(
  Axis:=2 ,
  Enable:=power_eb ,
  EnablePositive:=TRUE ,
  EnableNegative:=TRUE ,
  BufferMode:=mcAborting ,
  Status=>axis2_power
);

IF start AND rationum1<>rationum2 THEN
  c_temp:=c_temp+1;
END_IF;

IF c_temp=0 THEN
  gearin1_execute:=TRUE;
ELSE
  gearin1_execute:=FALSE;
END_IF;

IF c_temp=1 THEN
  gearin2_execute:=TRUE;
  c_temp:=0;
ELSE
  gearin1_execute:=FALSE;
END_IF;

rationum2:=rationum1;

gearin1(
  Master:=1 ,
  Slave:=2 ,
  Execute:= axis1_power AND axis2_power AND gearin1_execute ,
  RatioNumerator:=rationum1 ,
  RatioDenominator:=1 ,
  MasterValueSource:=0 ,
  Acceleration:=100 ,
  Deceleration:=100 ,
  Jerk:=1000 ,
  BufferMode:=mcAborting ,
  InGear=>gearIn1_Ingear ,
  Busy=>gearIn1_busy ,
  Active=>gearIn1_busy ,

```

```

CommandAborted=>gearIn1_abort
);

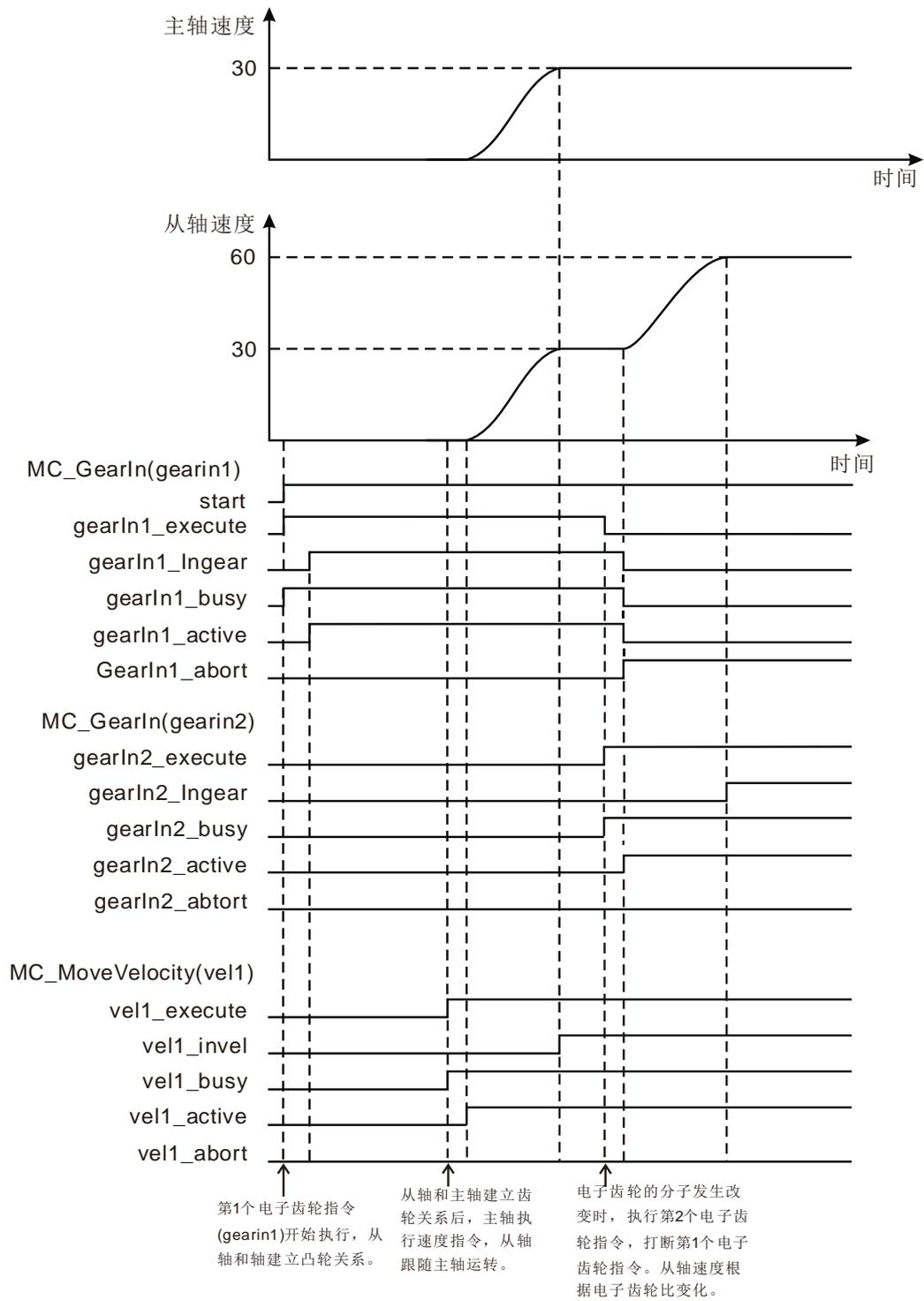
gearin2(
  Master:=1,
  Slave:=2,
  Execute:=axis1_power AND axis2_power AND gearin2_execute,
  RatioNumerator:=rationum1,
  RatioDenominator:=1,
  MasterValueSource:=0,
  Acceleration:=100,
  Deceleration:=100,
  Jerk:=1000,
  BufferMode:=mcAborting,
  InGear=>gearIn2_Ingear,
  Busy=>gearIn2_busy,
  Active=>gearIn2_busy,
  CommandAborted=>gearIn2_abort
);

vel1(
  Axis:=1,
  Execute:= axis1_power AND axis2_power AND vel1_execute,
  Velocity:=30,
  Acceleration:=100,
  Deceleration:=100,
  Jerk:=1000,
  Direction:=1,
  BufferMode:=mcAborting,
  InVelocity=> vel1_invelocity,
  Busy=> vel1_busy,
  Active=> vel1_active,
  CommandAborted=> vel1_abort
);

```

#### ● 程序说明

- 轴使能后，rationum1（电子齿轮比分子）初始值设置为1，start变量变为TRUE时，gearin1\_execute变为TRUE，第1个电子齿轮指令（gearin1）开始执行，第2个周期后，从轴2和主轴1建立齿轮比为1:1的电子齿轮关系。
- 从轴和主轴建立齿轮关系后，主轴执行速度指令，从轴跟随主轴运转。
- 如果想调整从轴速度时，可以通过更改MC\_GearIn指令电子齿轮比的分子值实现调整从轴速度的目的。如rationum1（电子齿轮比分子）的值由1变为2时，程序会触发gearin2\_execute变为TRUE，第2个电子齿轮指令（gearin2）执行，gearin2执行后会打断gearin1的执行，通过更改齿轮比实现调整从轴速度的目的。
- 两个电子齿轮指令相互切换的做法为start变量为TRUE后，c\_temp变量的值加1，并且开始检测rationum1和rationum2的值是否相同，即检测电子齿轮比的分子的值是否变化，如果变化，则c\_temp变量的值加1，c\_temp变量的值为1时，gearin1\_execute变为TRUE，触发第1个电子齿轮指令（gearin1）执行，c\_temp变量的值为2时，gearin2\_execute变为TRUE，触发第2个电子齿轮指令（gearin2）执行，并将c\_temp的值设置为0，如此循环。



## 5.2 MC\_CombineAxes (双主轴电子齿轮)

该指令用于将两个主轴和一个从轴根据设定的齿轮比建立电子齿轮关系。所属库：MotionControl

指令	名称	FB/FUN	梯形图样式	ST样式
MC_CombineAxes	双主轴电子齿轮	FB		<pre> MC_CombineAxes_Instance (   Master1 := 参数,   Master2 := 参数,   Slave := 参数,   Execute := 参数,   ContinuousUpdate := 参数,   CombineMode := 参数,   GearRatioNumeratorM1 := 参数,   GearRatioDenominatorM1 := 参数,   GearRatioNumeratorM2 := 参数,   GearRatioDenominatorM2 := 参数,   MasterValueSourceM1 := 参数,   MasterValueSourceM2 := 参数,   Acceleration := 参数,   Deceleration := 参数,   Jerk := 参数,   BufferMode := 参数,   InSync =&gt; 参数,   Busy =&gt; 参数,   Active =&gt; 参数,   CommandAborted =&gt; 参数,   Error =&gt; 参数,   ErrorID =&gt; 参数 ); </pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Master1	主轴1的轴号	USINT	由机种决定	不可缺省	指定电子齿轮中主轴1的轴号
Master2	主轴2的轴号	USINT	由机种决定	不可缺省	指定电子齿轮中主轴2的轴号
Slave	从轴轴号	USINT	由机种决定	不可缺省	指定电子齿轮中从轴的轴号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
ContinuousUpdate	持续更新	BOOL	TRUE或FALSE	FALSE	保留
CombineMode	加减法模式选择	MC_Combine_Mode	0: mcAddAxes 1: mcSubAxes	0	设定主轴1和主轴2加减法模式 0: 主轴1和主轴2变化位置相加 1: 主轴1和主轴2变化位置相减
GearRatioNumeratorM1	主轴1齿轮比分子	LREAL	正数、负数	不可缺省	设定主轴1齿轮比分子
GearRatioDenominatorM1	主轴1齿轮比分母	LREAL	正数、负数	不可缺省	设定主轴1齿轮比分母
GearRatioNumeratorM2	主轴2齿轮比分子	LREAL	正数、负数	不可缺省	设定主轴2齿轮比分子
GearRatioDenominatorM2	主轴2齿轮比分母	LREAL	正数、负数	不可缺省	设定主轴2齿轮比分母
MasterValueSourceM1	主轴1位置类型设定	MC_Source	0: mcSetValue 1: mcActualValue	0	设定主轴1的位置类型 0: 命令位置 1: 实际位置
MasterValueSourceM2	主轴2位置类型设定	MC_Source	0: mcSetValue 1: mcActualValue	0	设定主轴2的位置类型 0: 命令位置 1: 实际位置
Acceleration	加速度	LREAL	正数	不可缺省	从轴和主轴啮合时从轴的加速度*1 (单位: 行程单位/秒 <sup>2</sup> )*2

Deceleration	减速度	LREAL	正数	不可缺省	从轴和主轴啮合时从轴的减速度*1 (单位: 行程单位/秒 <sup>2</sup> ) *2
Jerk	跃度	LREAL	正数	不可缺省	从轴和主轴啮合时从轴的跃度*1 (单位: 行程单位/秒 <sup>3</sup> ) *2
BufferMode	缓存模式	MC_Buffer_Mode	0: mcAborting 1: mcBuffered	0	设定两个指令之间缓存模式*3 0: 中断 1: 等待

\*1: Velocity、Acceleration、Deceleration、Jerk的关系请参考“[运动控制指令相关参数说明](#)”章节。

\*2: 指令单位的详细介绍请参考“[运动控制指令参数单位](#)”章节。

\*3: BufferMode的详细介绍请参考“[运动控制指令的多重启动时缓存模式说明 \(BufferMode\)](#)”章节。

## ■ 输出变量

名称	名称	数据类型	输出范围	功能
InSync	齿轮比到达	BOOL	TRUE或FALSE	指定从轴到达目标速度时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Active	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时, 输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

## ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
InSync	从轴命令速度到达目标速度时	<ul style="list-style-type: none"> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>
Busy	Execute 由FALSE变为TRUE时	<ul style="list-style-type: none"> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>
Active	指令开始控制轴时	<ul style="list-style-type: none"> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>
CommandAborted	指令的执行被中断时	<ul style="list-style-type: none"> <li>◆ CommandAborted为TRUE, Execute由TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时, 该指令被其它指令中断时, CommandAborted变为TRUE一个周期后, 变为FALSE</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Execute由TRUE变为FALSE时</li> </ul>

## ■ 功能说明

### ● 基本功能说明

- 该指令执行后, 控制器计算每个周期主轴1和主轴2产生的位置相对值, 每个主轴产生的相对位置值再乘以各自对应的齿轮比, 两个主轴经过齿轮比后的位置相加或者相减, 最后得到的值加上从轴的当前位置传给从轴。
- 两个主轴每个周期产生的位置相对值乘以电子齿轮比后相加或者相减, 通过输入变量CombineMode的值进行选择。从轴可以选择跟随主轴命令位置或者反馈位置进行齿轮动作, 通过输入变量MasterValueSourceM1和MasterValueSourceM2的值进行选择。主轴轴号可以设定为伺服轴、虚拟伺服轴、脉冲轴、编码器轴, 从轴轴号可以设定为伺服轴、虚拟伺服轴、脉冲轴。
- 该指令执行时, 从轴状态须在使能状态下才可以执行, 主轴状态在使能或去使能状态下都可以执行。
- 该指令执行后, 两个轴在建立电子齿轮关系过程中 (如主轴运转时执行该指令), 从轴没有达到目标速度时 (该指令的InGear为FALSE时), 从轴按照该指令设定的Acceleration (加速度)、Deceleration (减速

度)、Jerk (跃度) 加速或减速到达目标速度。从轴的目标速度如下式所示。该过程中主轴和从轴没有同步, 加速过程中造成的主从轴同步误差不会自动补偿。

- 该指令执行后, 两个轴建立电子齿轮关系后 (该指令的InGear为TRUE时), 从轴速度、加速度、减速度和主轴速度、加速度、减速度的关系如下式所示:

$$\text{从轴目标速度} = \text{主轴1速度} \times \frac{\text{主轴1齿轮比分子}}{\text{主轴1齿轮比分母}} \pm \text{主轴2速度} \times \frac{\text{主轴2齿轮比分子}}{\text{主轴2齿轮比分母}}$$

$$\text{从轴目标加(减)速度} = \text{主轴1加(减)速度} \times \frac{\text{主轴1齿轮比分子}}{\text{主轴1齿轮比分母}} \pm \text{主轴2加(减)速度} \times \frac{\text{主轴2齿轮比分子}}{\text{主轴2齿轮比分母}}$$

- 该指令执行后, 如果需要解除从轴和主轴的耦合关系, 只能通过对从轴执行MC\_Stop指令解除。

- **重启该指令**

该指令执行后, 不可以再重新执行。如果想更改该指令的输入变量, 可以执行另外一个该指令中断正在执行的该指令。

- **其它指令执行时启动该指令**

其它运动指令执行时, 启动该指令, 该指令和其它运动指令如何中继由该指令输入变量 BufferMode (缓存模式) 的值决定。该指令输入变量 BufferMode (缓存模式) 可以设置的值只可以设置为 mcAborting (中断) 或者 mcBuffered (等待)。

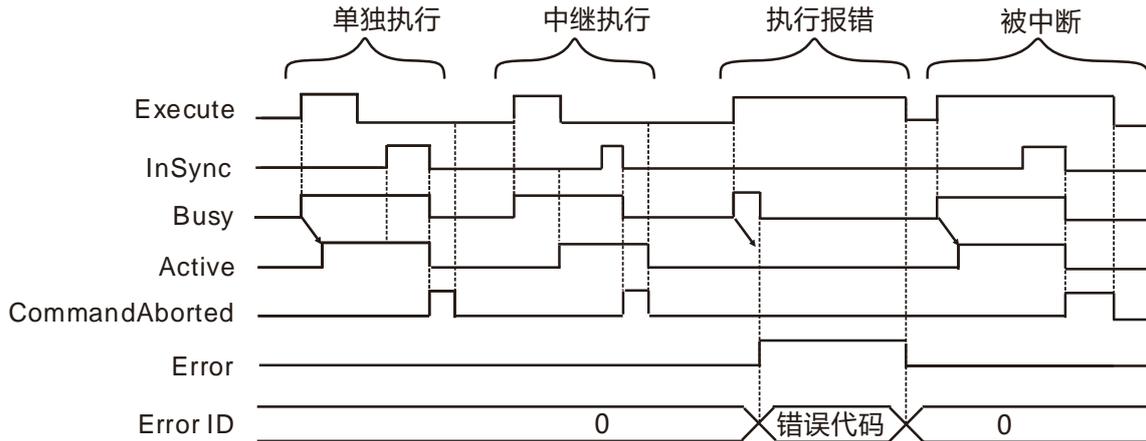
- **该指令执行时启动其它指令**

该指令执行时启动其它运动指令, 其它运动指令和该指令如何中继由其它指令的 BufferMode (缓存模式) 参数数值决定, 其它运动指令 BufferMode 的值只能选择中断或者等待; 如果其它运动指令无 BufferMode 参数一般为中断该指令。其它运动指令和该指令缓存时, 其它指令执行的时机为该指令 InSync(齿轮比到) 变为 TRUE 时。

- **异常排除**

该指令执行时, 如果输入变量不合法, 该指令 Error 变为 TRUE, ErrorID 中有对应的错误码, 可以通过该错误码判断错误原因。该指令执行时, 如遇到异常 (如轴报警等), 该指令也会报错, 轴会立即停止。

## ■ 输出变量变化时序说明



### ● 单独执行

Execute (启动) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE, 下一个周期 Active (控制中) 变为 TRUE。当从轴命令速度达到目标速度时, InSync 变为 TRUE, Busy (执行中) 和 Active (控制中) 仍为 TRUE。

### ● 中继执行

其它指令控制轴时, 执行该指令 (BufferMode 的值不为 mcAborting), 当 Execute (启动) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE, Active (控制中) 变为 TRUE 的时机为前一个指令完成时。

### ● 执行报错

该指令输入变量值不在允许范围内时, 该指令 Execute (启动) 由 FALSE 变为 TRUE 时, 同时 Busy (执行中) 变为 TRUE, 下一个周期 Error (错误) 变为 TRUE, 同时 Busy (执行中) 变为 FALSE, ErrorID (错误码) 输出对应的错误码, 可通过 ErrorID (错误码) 的值, 查找发生问题原因。该指令 Execute (启动) 由 TRUE 变为 FALSE 时, 同时 Error (错误) 变为 FALSE, ErrorID (错误码) 的值变为 0。

### ● 被中断

该指令执行后被其它指令中断时, 该指令 CommandAborted (中断) 变为 TRUE, Busy (执行中)、Active (控制中)、InSync (从轴命令速度达到目标速度) 同时变为 FALSE; 当 Execute (执行中) 变为 FALSE 时, CommandAborted (中断) 同时变为 FALSE

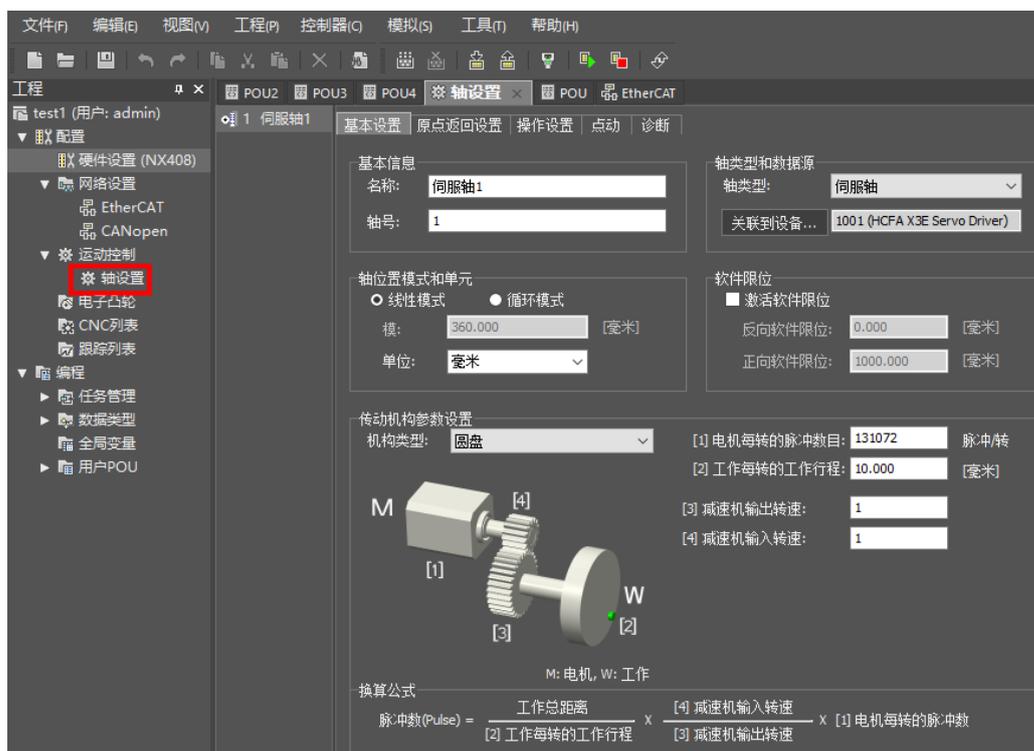
## ■ 示例程序

### ● 实现功能

将两个主轴的速度叠加后作用在从轴上, 从轴速度为两个主轴速度的叠加和。

### ● 轴参数设置

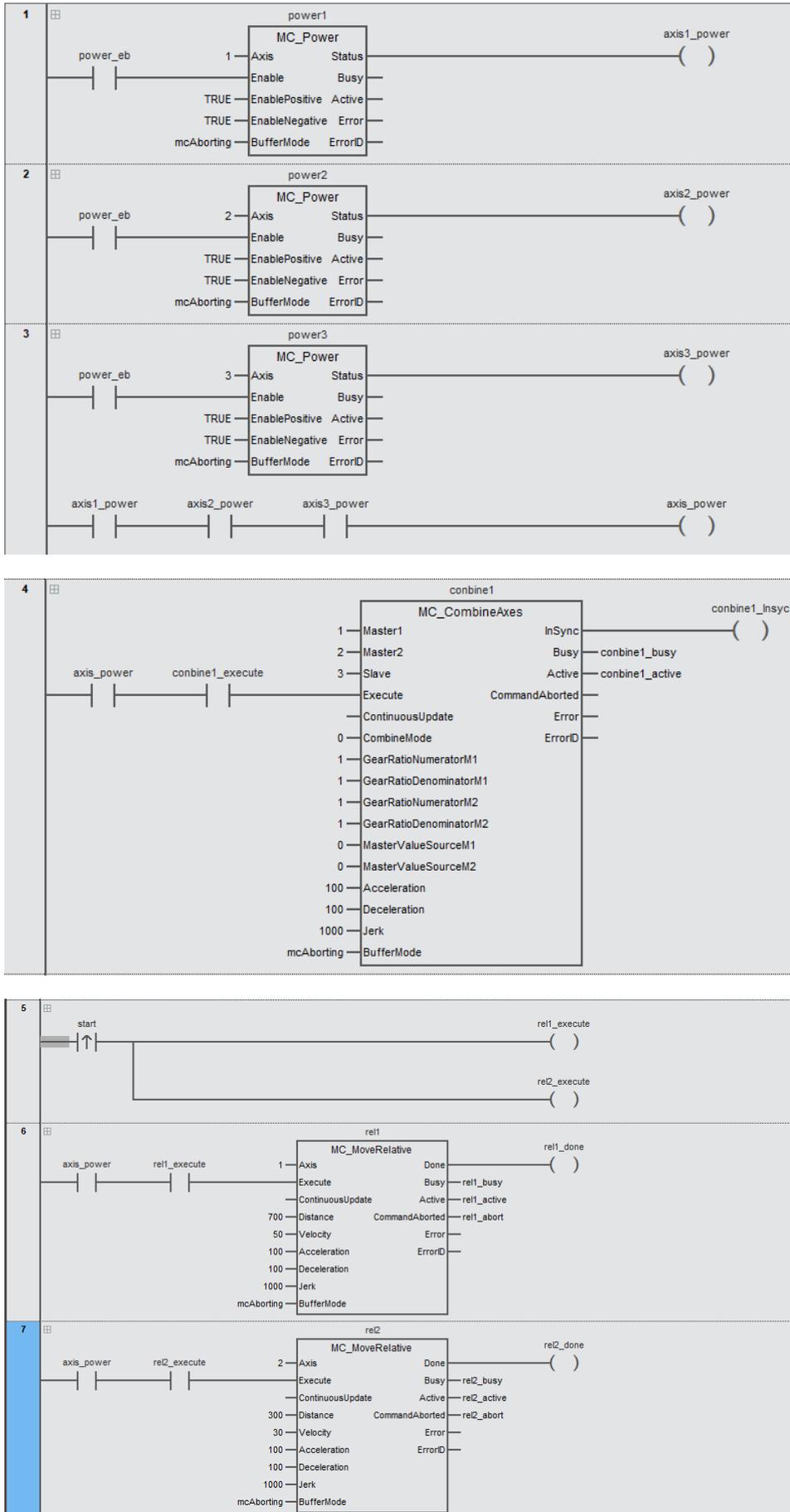
主轴轴 1、主轴轴 2 和从轴轴 3 的轴参数相同, 轴 1 轴参数设置如下图所示。



## ● 变量表

类别	名称	分配到	数据类型	初始值	注释
VAR	power_eb		BOOL		
VAR	power1		MC_Power		
VAR	axis1_power		BOOL		
VAR	power2		MC_Power		
VAR	axis2_power		BOOL		
VAR	power3		MC_Power		
VAR	axis3_power		BOOL		
VAR	axis_power		BOOL		
VAR	conbin1_execute		BOOL		
VAR	conbin1		MC_CombineAxes		
VAR	conbin1_Insys		BOOL		
VAR	conbin1_busy		BOOL		
VAR	conbin1_ative		BOOL		
VAR	conbin1_abort		BOOL		
VAR	rel1		MC_MoveRelative		
VAR	rel1_execute		BOOL		
VAR	rel1_done		BOOL		
VAR	rel1_busy		BOOL		
VAR	rel1_active		BOOL		
VAR	rel1_abort		BOOL		
VAR	rel2		MC_MoveRelative		
VAR	rel2_execute		BOOL		
VAR	rel2_done		BOOL		
VAR	rel2_busy		BOOL		
VAR	rel2_active		BOOL		
VAR	rel2_abort		BOOL		
VAR	start		BOOL		

● 梯形图(LD)



- 结构化文本(ST)

```
power1(
  Axis:=1 ,
  Enable:=power_eb ,
  EnablePositive:=TRUE ,
  EnableNegative:=TRUE ,
  BufferMode:=mcAborting ,
  Status=>axis1_power
);
```

```
power2(
  Axis:=1 ,
  Enable:=power_eb ,
  EnablePositive:=TRUE ,
  EnableNegative:=TRUE ,
  BufferMode:=mcAborting ,
  Status=>axis1_power
);
```

```
power3(
  Axis:=1 ,
  Enable:=power_eb ,
  EnablePositive:=TRUE ,
  EnableNegative:=TRUE ,
  BufferMode:=mcAborting ,
  Status=>axis1_power
);
```

```
axis_power:=axis1_power AND axis2_power AND axis3_power;
```

```
combine1(
  Master1:=1 ,
  Master2:=2 ,
  Slave:=3 ,
  Execute:=axis_power AND combine1_execute ,
  CombineMode:=0 ,
  GearRatioNumeratorM1:=1 ,
  GearRatioDenominatorM1:=1 ,
  GearRatioNumeratorM2:=1 ,
  GearRatioDenominatorM2:=1 ,
  MasterValueSourceM1:=0 ,
  MasterValueSourceM2:=0 ,
  Acceleration:=100 ,
  Deceleration:=100 ,
  Jerk:=1000 ,
  InSync=>combine1_Insync ,
  Busy=>combine1_busy ,
  Active=>combine1_busy ,
  CommandAborted=>combine1_abort
);
```

```
rel1_execute:=EDGEPOS(start) ;
```

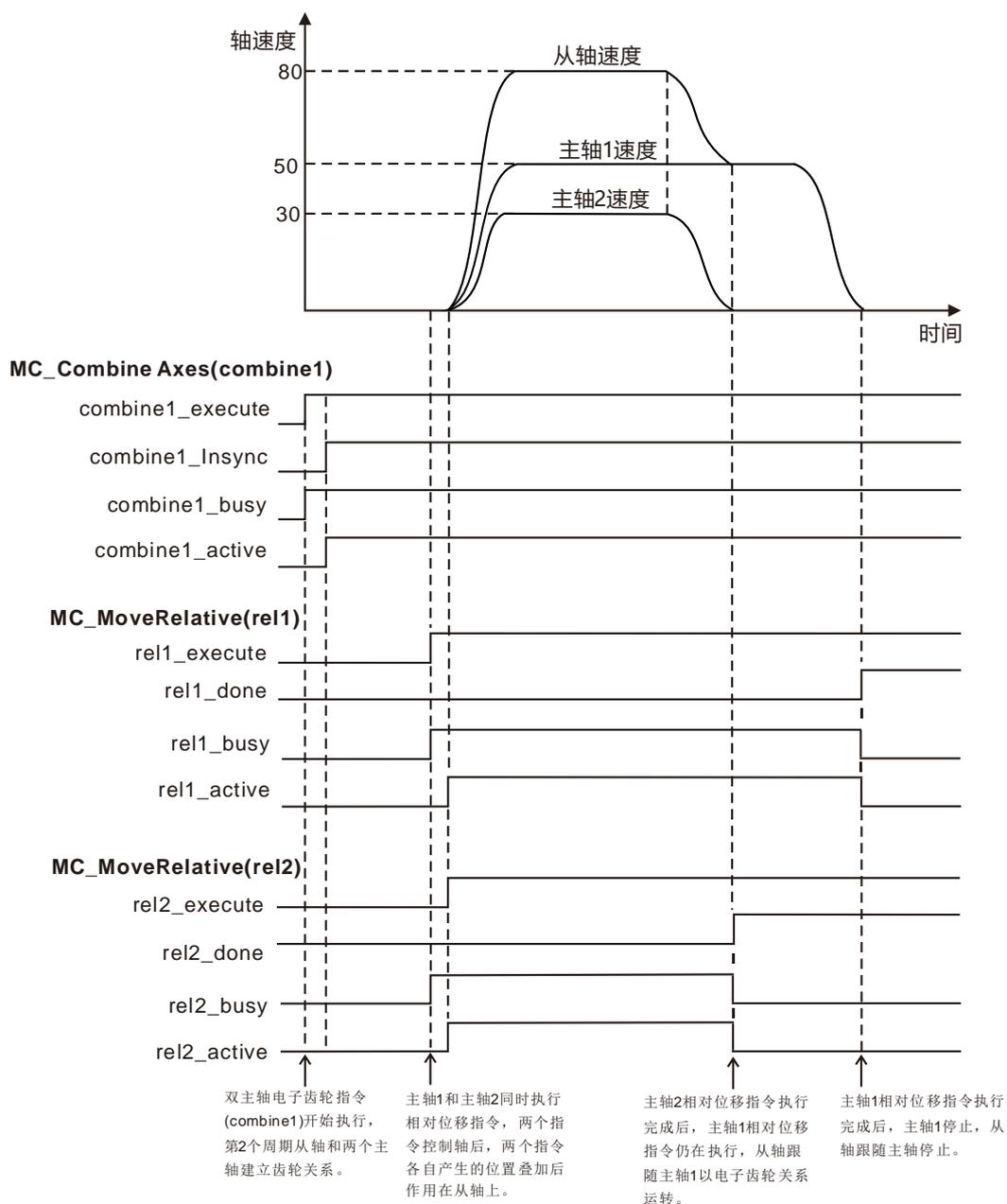
```
rel2_execute:=EDGEPOS(start) ;
```

```
rel1(  
  Axis:=1 ,  
  Execute:=axis_power AND rel1_execute ,  
  Distance:=700 ,  
  Velocity:=50 ,  
  Acceleration:=100 ,  
  Deceleration:=100 ,  
  Jerk:=1000 ,  
  BufferMode:=mcAborting ,  
  Done=>rel1_done ,  
  Busy=>rel1_busy ,  
  Active=>rel1_active ,  
  CommandAborted=>rel1_abort  
);
```

```
rel2(  
  Axis:=2 ,  
  Execute:=axis_power AND rel1_execute ,  
  Distance:=300 ,  
  Velocity:=30 ,  
  Acceleration:=100 ,  
  Deceleration:=100 ,  
  Jerk:=1000 ,  
  BufferMode:= mcAborting ,  
  Done=>rel2_done ,  
  Busy=>rel2_busy ,  
  Active=>rel2_active ,  
  CommandAborted=>rel2_abort  
);
```

#### ● 程序说明

- 轴使能后, combine1\_execute变为TRUE时, 双主轴电子齿轮指令 (combine1) 开始执行, 第2个周期, 从轴轴3和主轴轴1及主轴轴2建立齿轮比为1:1的电子齿轮关系。
- 从轴和主轴建立齿轮关系后, 两个主轴同时执行相对位移指令, 从轴速度为两个主轴速度的和。
- 主轴2相对位移指令执行完成后, 主轴1仍在执行相对位移指令, 从轴轴3跟随主轴轴1运转, 从轴轴3和主轴轴1的速度相同。
- 主轴轴1相对位移指令执行完成后停止, 从轴轴3跟随主轴轴 1停止。



### 5.3 MC\_GearOut (电子齿轮关系解除)

该指令用于中止MC\_GearIn指令的执行，从轴和主轴的电子齿轮关系也随之解除。所属库：MotionControl

指令	名称	FB/FUN	梯形图样式	ST样式
MC_GearOut	电子齿轮关系解除	FB		<pre>MC_GearOut_Instance ( Slave:= 参数 , Execute := 参数 , Done =&gt; 参数 , Busy =&gt; 参数 , CommandAborted =&gt; 参数 , Error =&gt; 参数 , ErrorID=&gt; 参数 );</pre>

#### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Slave	从轴轴号	USINT	由机种决定	不可缺省	电子齿轮中从轴的轴号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令

#### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	电子齿轮关系解除	BOOL	TRUE或FALSE	从轴和主轴的电子齿轮关系解除时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

#### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	从轴和主轴的电子齿轮关系解除时	<ul style="list-style-type: none"> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>
Busy	Execute 由FALSE变为TRUE时	<ul style="list-style-type: none"> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>
Active	指令开始控制轴时	<ul style="list-style-type: none"> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>
CommandAborted	指令的执行被中断时	<ul style="list-style-type: none"> <li>◆ CommandAborted为TRUE，Execute由TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时，该指令被其它指令中断时，CommandAborted变为TRUE一个周期后，变为FALSE</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Execute由TRUE变为FALSE时</li> </ul>

#### ■ 功能说明

- **基本功能说明**

- 该指令用于中止MC\_GearIn指令的执行。该指令执行后，从轴和主轴通过MC\_GearIn建立的电子齿轮关系也随之解除，解除后从轴保持解除关系时的从轴速度。如从轴运转时执行该指令，该指令执行后，从轴仍会继续运转。如果想解除电子齿轮关系并希望从轴减速停止，可以对从轴执行MC\_Halt或者MC\_Stop指令。
- 该指令的执行对主轴的动作无影响。

- **重启该指令**

该指令无法重启。两个轴没有建立电子齿轮关系时，执行该指令会报错。

- **其它指令执行时启动该指令**

只有 MC\_GearIn 指令执行时，才可以执行该指令；其它运动指令执行时，执行该指令会报错。

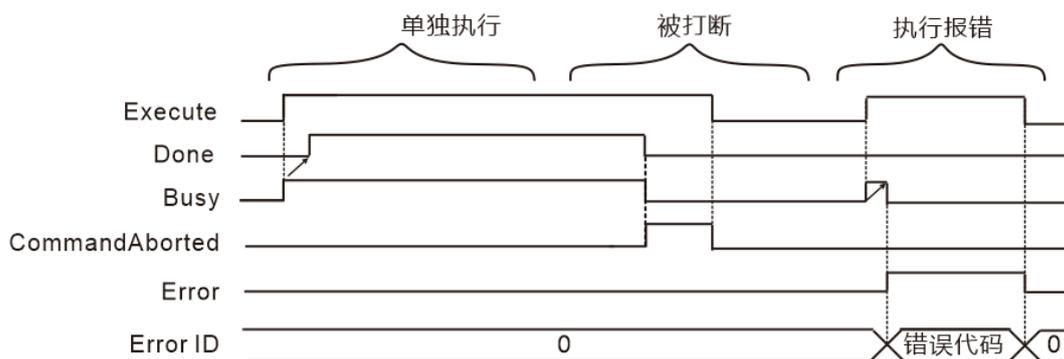
- **该指令执行时启动其它指令**

该指令执行时启动其它运动指令，其它运动指令和该指令如何中继由其它指令的 BufferMode（缓存模式）参数值决定，其它运动指令 BufferMode 的值只能选择中断或者等待，如下表所示；如果其它运动指令无 BufferMode 参数，一般为中断该指令。其它运动指令和该指令缓存时，其它指令执行的时机为该指令 Done(电子齿轮关系解除)变为 TRUE 时。

- **异常排除**

该指令执行时，如果输入变量不合法，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。该指令执行时，如遇到异常（如轴报警等），该指令也会报错，轴会立即停止。

- **输出变量变化时序说明**



- **单独执行**

Execute（启动）由 FALSE 变为 TRUE 时，Busy（执行中）同时变为 TRUE，下一个周期 Done（电子齿轮关系解除）变为 TRUE，Busy（执行中）仍为 TRUE，轴的命令速度保持脱离前的速度不变。当 Execute（启动）变为 FALSE 时，Busy（执行中）和 Done（电子齿轮关系解除）仍为 TRUE。

- **被中断**

该指令执行后被其它指令中断时，该指令 CommandAborted（中断）变为 TRUE，Busy（执行中）和 Done（电子齿轮关系解除）同时变为 FALSE；当 Execute（执行中）变为 FALSE 时，CommandAborted（中断）同时变为 FALSE。

- **执行报错**

该指令输入变量值不在允许范围内时，该指令 Execute（启动）由 FALSE 变为 TRUE 时，同时 Busy（执行中）变为 TRUE，下一个周期 Error（错误）变为 TRUE，同时 Busy（执行中）变为 FALSE，ErrorID（错误码）输出对应的错误码，可通过 ErrorID（错误码）的值，查找发生问题原因。该指令 Execute（启动）由 TRUE 变为 FALSE 时，同时 Error（错误）变为 FALSE，ErrorID（错误码）的值变为 0。

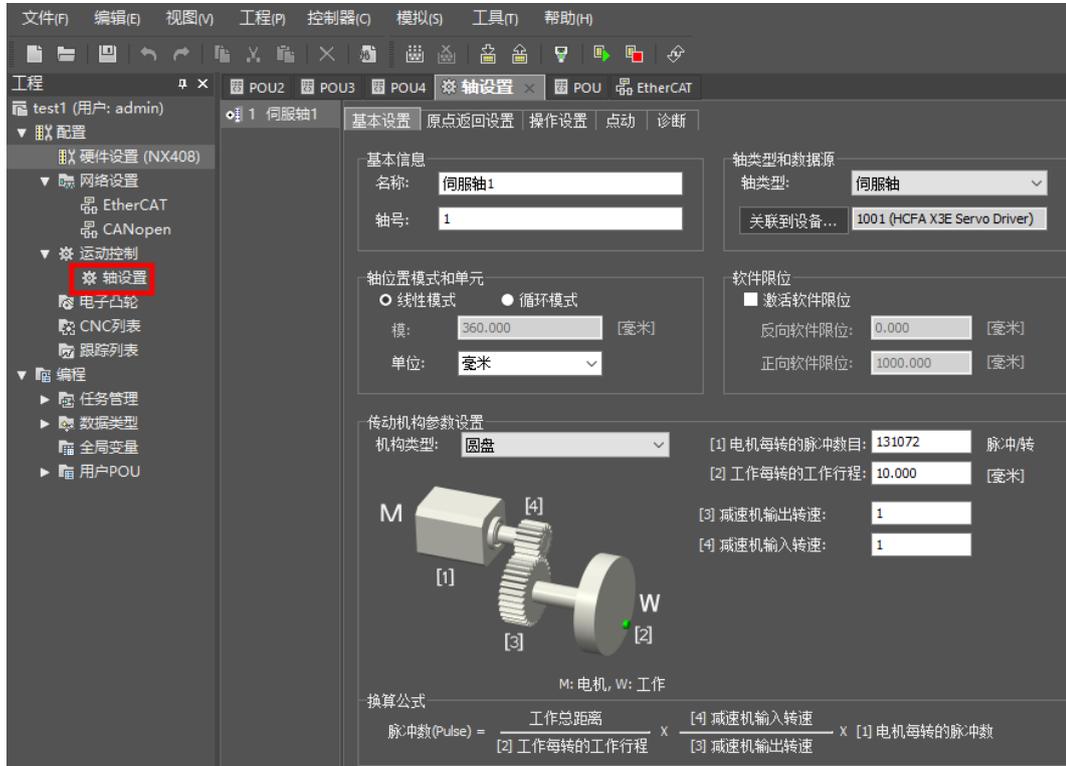
- **示例程序**

● 实现功能

从轴和主轴建立电子齿轮关系后，如果从轴和主轴的电子齿轮关系需要解除并保持从轴当前速度继续运转，可以执行 MC\_GearOut（电子齿轮关系解除）实现。

● 轴参数设置

主轴轴 1 和从轴轴 2 的轴参数相同，轴 1 轴参数设置如下图所示。

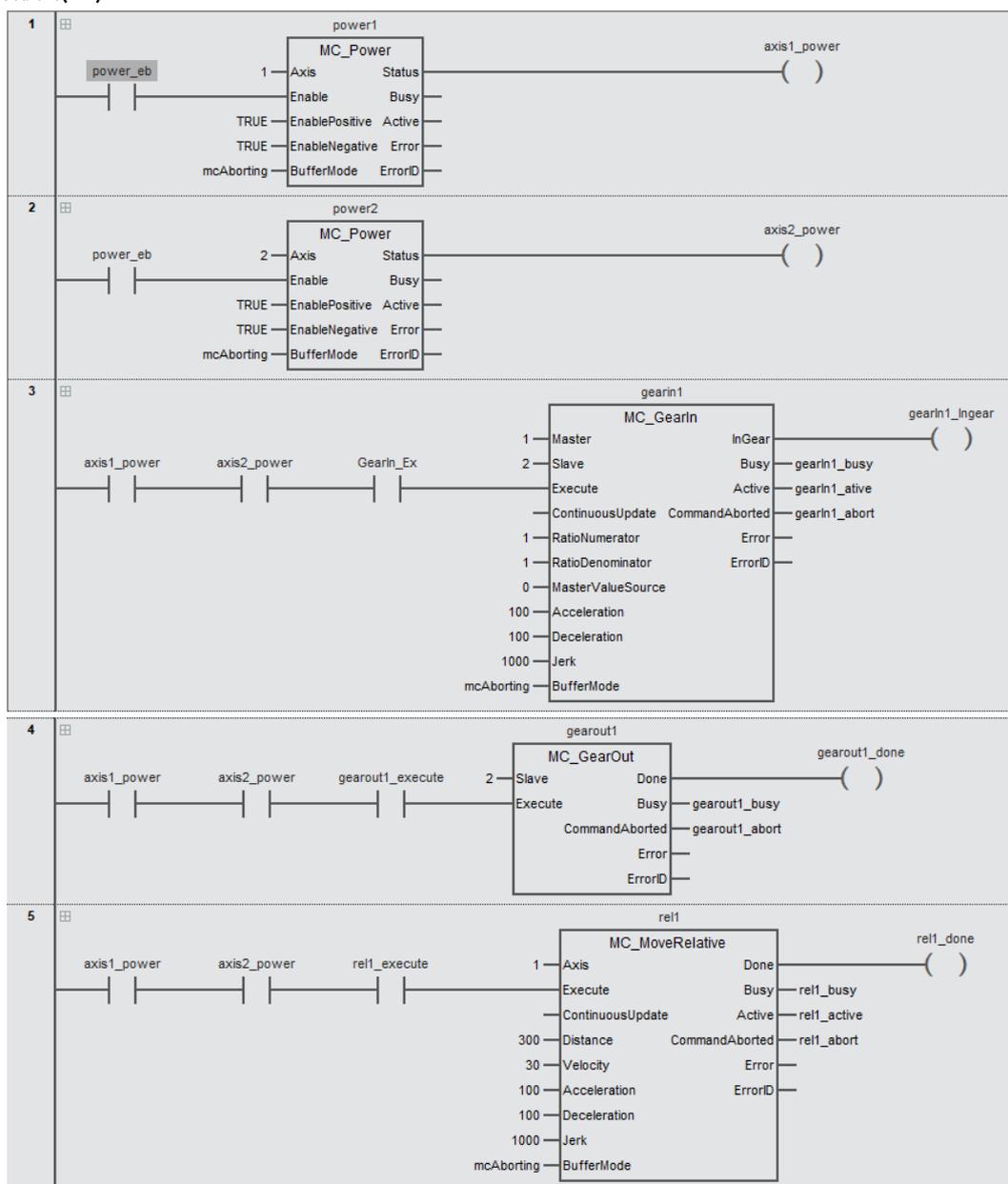


● 变量表

类别	名称	分配到	数据类型	初始值	注释
VAR	power_eb		BOOL		
VAR	power1		MC_Power		
VAR	axis1_power		BOOL		
VAR	power2		MC_Power		
VAR	axis2_power		BOOL		
VAR	GearIn_Ex		BOOL		
VAR	gearin1		BOOL		
VAR	gearIn1_Ingear		BOOL		
VAR	gearIn1_busy		BOOL		
VAR	gearIn1_ative		BOOL		
VAR	gearIn1_abort		BOOL		
VAR	gearout1		MC_GearOut		
VAR	gearout1_execute		BOOL		
VAR	gearout1_done		BOOL		
VAR	gearout1_busy		BOOL		
VAR	gearout1_abort		BOOL		
VAR	rel1		MC_MoveRelative		
VAR	rel1_execute		BOOL		
VAR	rel1_done		BOOL		
VAR	rel1_busy		BOOL		

VAR	rel1_active		BOOL		
VAR	rel1_abort		BOOL		

● 梯形图(LD)



● 结构化文本(ST)

```
power1 (
  Axis:=1,
  Enable:=power_eb,
  EnablePositive:=TRUE,
  EnableNegative:=TRUE,
  BufferMode:=mcAborting,
  Status=>axis1_power
);
```

```
power2(
  Axis:=2,
  Enable:=power_eb,
  EnablePositive:=TRUE,
```

```

    EnableNegative:=TRUE ,
    BufferMode:=mcAborting ,
    Status=>axis2_power
);

gearin1(
    Master:=1 ,
    Slave:=2 ,
    Execute:=axis1_power AND axis2_power AND GearIn_Exec ,
    RatioNumerator:=1 ,
    RatioDenominator:=1 ,
    MasterValueSource:=0 ,
    Acceleration:=100 ,
    Deceleration:=100 ,
    Jerk:=1000 ,
    BufferMode:=mcAborting ,
    InGear=>gearIn1_Ingear ,
    Busy=>gearIn1_busy ,
    Active=>gearIn1_busy ,
    CommandAborted=>gearIn1_abort
);

gearout1(
    Slave:=2 ,
    Execute:=axis1_power AND axis2_power AND gearout1_execute ,
    Done=>gearout1_done ,
    Busy=>gearout1_busy ,
    CommandAborted=>gearout1_abort
);

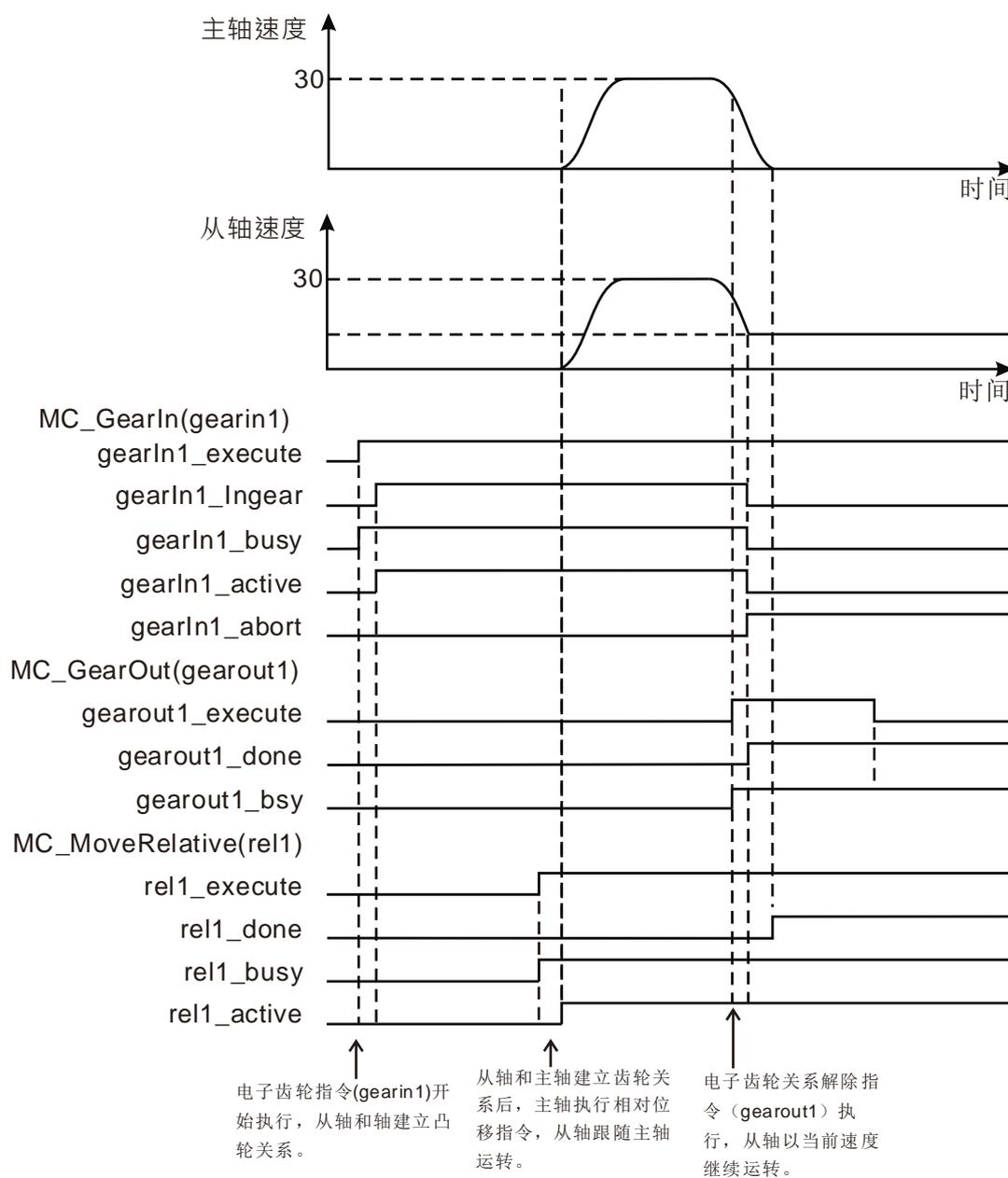
rel1(
    Axis:=1 ,
    Execute:=axis1_power AND axis2_power AND rel1_execute ,
    Distance:=300 ,
    Velocity:=30 ,
    Acceleration:=100 ,
    Deceleration:=100 ,
    Jerk:=1000 ,
    BufferMode:=mcAborting ,
    Done=>rel1_done ,
    Busy=>rel1_busy ,
    Active=>rel1_active ,
    CommandAborted=>rel1_abort
);

```

#### ● 程序说明

- 轴使能后，gearin1\_execute变为TRUE时，电子齿轮指令（gearin1）开始执行，第2个周期后，从轴轴2和主轴轴1建立齿轮比为1:1的电子齿轮关系。
- 从轴和主轴建立齿轮关系后，主轴执行相对位移指令，从轴跟随主轴运转。
- 从轴和主轴建立齿轮关系后，gearout1\_execute变为TRUE时，电子齿轮脱离指令（gearout1）指令执行，

第2个周期，gearout1的Done变为TRUE，从轴和主轴脱离电子齿轮关系，从轴以当前速度继续运转。如果要从轴和主轴脱离电子齿轮关系并停止，从轴可以执行MC\_Halt指令实现。



## 5.4 MC\_CamIn (电子凸轮耦合)

控制从轴与主轴按规划的凸轮关系同步运动。所属库: MotionControl

指令	名称	FB/FUN	梯形图样式	ST样式
MC_CamIn	电子凸轮耦合	FB		<pre>MC_CamIn_Instance( Master:= 参数 , Slave:= 参数 , Execute:= 参数 , ContinuousUpdate:= 参数, CamTable:= 参数 , Periodic:= 参数 , MasterAbsolute:= 参数 , SlaveAbsolute:= 参数 , MasterOffset:= 参数 , SlaveOffset:= 参数 , MasterScaling:= 参数 , SlaveScaling:= 参数 , MasterStartDistance:= 参数 , MasterSyncPosition:= 参数 , ActivationPosition:= 参数 , ActivationMode:= 参数 , StartMode:= 参数 , Velocity:= 参数 , Acceleration:= 参数 , Deceleration:= 参数 , Jerk:= 参数 , MasterValueSource:= 参数 , BufferMode:= 参数 , InSync=&gt; 参数 , EndOfProfile=&gt; 参数 , Busy=&gt; 参数 , Active=&gt; 参数 , CommandAborted=&gt; 参数 , Error=&gt; 参数 , ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Master	主轴轴号	USINT	由机种决定	不可缺省	指定电子凸轮中主轴的轴号
Slave	从轴轴号	USINT	由机种决定	不可缺省	指定电子凸轮中从轴的轴号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
ContinuousUpdate	持续更新	BOOL	TRUE或FALSE	FALSE	保留
CamTable	电子凸轮关系表编号	USINT	由机种决定	不可缺省	设定电子凸轮关系表编号
Periodic	周期执行	BOOL	TRUE或FALSE	FALSE	设定凸轮执行规律 TRUE:周期循环执行 FALSE:仅执行一个周期
MasterAbsolute	主轴绝对位置模式	BOOL	TRUE或FALSE	FALSE	设定主轴轴位置与其凸轮相位之间的对应关系。 TRUE:绝对模式 FALSE:相对模式
SlaveAbsolute	从轴绝对位置模式	BOOL	TRUE或FALSE	FALSE	设定从轴轴位置与其凸轮相位之间的对应关系。 TRUE:绝对模式

					FALSE:相对模式。
MasterOffset	主轴偏移	LREAL	负数、0、正数	0	设定主轴偏移位置
SlaveOffset	从轴偏移	LREAL	负数、0、正数	0	设定从轴偏移位置
MasterScaling	主轴比例	LREAL	正数	0	设定主轴位置变换比例
SlaveScaling	从轴比例	LREAL	负数、正数	不可缺省	设定从轴位置变换比例
MasterStartDistance	-	-		-	保留
MasterSyncPosition	-	-		-	保留
ActivationPosition	凸轮耦合启动位置	LREAL	负数、0、正数	0	设定凸轮耦合动作启动位置,当主轴经过该位置时,凸轮耦合正式开始
ActivationMode	启动位置的类型	MC_ACTIVATION_MODE	0: mcRelative 1: mcAbsolute 2: mcPhaseAxis 3: mcPhaseCAM	0	设定ActivationPosition的类型。 0:相对位置 1:绝对位置 2:轴相位 3:凸轮相位
StartMode	耦合动作模式	MC_START_MODE	-1、0、1	0	设定耦合动作的模式。 -1(mcRampInNegative):反向 0(mcRampInShortest):距离最短 1(mcRampInPositive):正向
Velocity	耦合补偿速度	LREAL	正数	0	设定耦合过程中的补偿速度的上限值。*1 (单位:行程单位/秒)*2
Acceleration	耦合补偿加速度	LREAL	正数	0	设定耦合过程中的补偿加速度的上限值。*1 (单位:行程单位/秒 <sup>2</sup> )*2
Deceleration	耦合补偿减速度	LREAL	正数	0	设定耦合过程中的补偿减速度的上限值。*1 (单位:行程单位/秒 <sup>2</sup> )*2
Jerk	跃度	LREAL	正数	0	保留
MasterValueSource	主轴位置来源	MC_SOURCE	0: mcSetValue 1: mcActualValue	0	设定主轴位置的来源。 0: 命令位置 1: 实际位置
BufferMode	缓存模式	MC_Buffer_Mode	0: mcAborting 1: mcBuffered	0	设定两个指令之间缓存模式。*3 0: 中断 1: 等待

\*1: Velocity、Acceleration、Deceleration、Jerk的关系请参考“[运动控制指令相关参数说明](#)”章节。

\*2: 指令单位的详细介绍请参考“[运动控制指令参数单位](#)”章节。

\*3: BufferMode的详细介绍请参考“[运动控制指令的多重启动时缓存模式说明\(BufferMode\)](#)”章节。

## ■ 输出变量

名称	名称	数据类型	输出范围	功能
InSync	凸轮同步中	BOOL	TRUE或FALSE	从轴与主轴处于凸轮同步中。
EndOfProfile	凸轮周期结束	BOOL	TRUE或FALSE	一个凸轮周期结束。
Busy	正在执行	BOOL	TRUE或FALSE	指令正在执行中。
Active	正在控制轴	BOOL	TRUE或FALSE	指令正在控制轴。
CommandAborted	被中断	BOOL	TRUE或FALSE	指令执行被中断
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE。
ErrorID	错误代码	WORD	0~65535	指令执行异常时,输出错误代码。 值的含义请参考“ <a href="#">指令错误代码描述</a> ”。

## ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
InSync	耦合完成时(从轴追上主轴)。	<ul style="list-style-type: none"> <li>◆ CommandAborted变为TRUE时;</li> <li>◆ Error变为TRUE时;</li> <li>◆ Periodic为FALSE, EndOfProfile变为TRUE时。</li> </ul>
EndOfProfile	凸轮同步后, 每次经过凸轮结束点时。	<ul style="list-style-type: none"> <li>◆ 变为TRUE后下一程序周期变为FALSE。</li> </ul>
Busy	指令触发执行时。	<ul style="list-style-type: none"> <li>◆ CommandAborted变为TRUE时;</li> <li>◆ Error变为TRUE时;</li> <li>◆ Periodic为FALSE, EndOfProfile变为TRUE时。</li> </ul>
Active	指令正式控制轴时。	<ul style="list-style-type: none"> <li>◆ CommandAborted变为TRUE时;</li> <li>◆ Error变为TRUE时;</li> <li>◆ Periodic为FALSE, EndOfProfile变为TRUE时。</li> </ul>
CommandAborted	指令被中断时。	<ul style="list-style-type: none"> <li>◆ Execute从TRUE变为FALSE时;</li> <li>◆ Execute为FALSE, CommandAborted变为TRUE后下一程序周期变为FALSE。</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常。	<ul style="list-style-type: none"> <li>◆ Execute从TRUE变为FALSE时;</li> <li>◆ Execute为FALSE, Error变为TRUE后下一程序周期变为FALSE。</li> </ul>

## ■ 功能说明

### ● 基本功能说明

本指令为电子凸轮耦合指令, 用于控制从轴跟随主轴按照规划的凸轮关系同步运动。本指令仅参考主轴位置, 不控制主轴, 故主轴可以是任意类型的轴, 但从轴只允许是伺服轴和虚拟伺服轴。

### ● 指令执行过程

本指令执行过程包含三个阶段: 等待耦合开始、耦合中、同步中。

等待耦合开始: 指令触发执行, 但主轴尚未经过指定的耦合启动位置。

耦合中: 从轴追赶主轴, 但与主轴尚未实现凸轮同步。

同步中: 从轴与主轴实现凸轮同步并持续同步运行。

### ● 耦合启动位置设定

在实际应用过程中经常遇到需要在主轴的特定位置(相位、角度)启动凸轮耦合, 在主轴未经过该特定位置时, 从轴不动作, 处于等待状态。面对这种需求, 可通过设置耦合启动位置实现。耦合启动位置是指当主轴经过该位置时, 耦合动作正式启动, 此后从轴开始追赶主轴。

耦合启动位置由参数 ActivationPosition 和 ActivationMode 综合设定, 参数 ActivationPosition 设定位置, ActivationMode 设定该位置的类型。耦合启动位置分为四种: 相对位置、绝对位置、轴相位、凸轮相位。

**ActivationMode=0 (相对位置)**: 耦合启动位置定义为主轴的轴位置, 此时耦合启动位置为本指令触发执行时刻(Execute 上升沿时刻)的主轴位置加上 ActivationPosition。例如, 指令触发执行时刻的主轴位置为 1000, ActivationPosition 的值为 5000, 则耦合启动位置为 6000 ( $6000=1000+5000$ )。

**ActivationMode=1 (绝对位置)**: 耦合启动位置定义为主轴的轴位置, 此时耦合启动位置为 ActivationPosition。例如, 指令触发执行时刻的主轴位置为 1000, ActivationPosition 的值为 5000, 则耦合启动位置为 5000。

**ActivationMode=2 (轴相位)**: 耦合启动位置定义为主轴的轴相位。例如, 主轴的模为 360 (在软件轴参数设置界面中可设置), 指令触发执行时刻的主轴位置为 500, 此时主轴的轴相位为 140 ( $140=500\%360$ ), 若 ActivationPosition 设定 100, 则耦合启动位置为 820 ( $820\%360=100$ ) 或 460 ( $460\%360=100$ )。当主轴正向运动经过 820 或反向运动经过 460 时均可启动耦合动作。

**ActivationMode=3 (凸轮相位)**: 耦合启动位置定义为主轴的绝对凸轮相位。例如, 凸轮曲线规划主轴凸轮周期为 1000, 指令触发执行时刻的主轴位置为 1500, 此时主轴的绝对凸轮相位为 500 ( $500=1500\%1000$ ), 若

ActivationPosition 设定 100，则耦合启动位置为 2100 ( $2100\%1000=100$ ) 或 1100 ( $1100\%1000=100$ )。当主轴正向运动经过 2100 或反向运动经过 1100 时均可启动耦合动作。

**注意：**主轴可能多次经过耦合启动位置，但仅在指令触发执行后第一次遇到耦合启动位置时有效，此后耦合启动位置将不再作用。

### ● 轴位置与凸轮相位之间的对应关系

凸轮曲线中规划的凸轮关系仅为一个凸轮周期中主从轴之间的对应关系，但在凸轮执行过程中，可能需要持续多个凸轮周期，那么凸轮曲线规划的同步关系实际是主从轴的凸轮相位对应关系。主从轴的轴位置是线性的，在建立凸轮同步关系时，首先需要确定轴位置与凸轮相位之间的对应关系。

参数 MasterAbsolute 用于设定主轴轴位置与主轴凸轮相位之间的对应关系，参数 SlaveAbsolute 用于设定从轴轴位置与从轴凸轮相位之间的对应关系。

**MasterAbsolute=FALSE (主轴相对模式)：**耦合启动时刻的主轴轴位置对应其凸轮相位为 0 (凸轮曲线的起点)。例如，耦合启动时刻的主轴位置为 1500，则主轴轴位置与其凸轮相位之间的对应关系基准为：轴位置 1500 对应于凸轮相位 0，那么凸轮将从相位 0 (起点) 开始执行。

**MasterAbsolute=TRUE (主轴绝对模式)：**主轴轴位置与其凸轮相位之间的对应关系基准为：轴位置 0 对应于凸轮相位 0。例如，耦合启动时刻的主轴位置为 1500，凸轮关系中主轴周期为 1000，则耦合启动时刻的主轴凸轮相位为 500，那么凸轮将从相位 500 的位置开始执行。

**SlaveAbsolute=FALSE (从轴相对模式)：**耦合启动时刻从轴的凸轮相位与主轴的凸轮相位符合凸轮同步关系。耦合启动时刻，无论从轴处于何位置，其凸轮相位均与该时刻的主轴凸轮相位满足同步关系。例如，耦合启动时刻，主轴的凸轮相位为 500，根据凸轮关系，与之对应的从轴凸轮相位为 100，若该时刻从轴的轴位置为 0，则从轴的轴位置与其凸轮相位之间的对应关系基准为：轴位置 100 对应于凸轮相位 0。

**SlaveAbsolute=TRUE (从轴绝对模式)：**从轴轴位置与其凸轮相位之间的对应关系基准为：轴位置 0 对应于凸轮相位 0。例如，耦合启动时刻的从轴位置为 100，凸轮关系中从轴周期为 1000，则耦合启动时刻的从轴凸轮相位为 100。

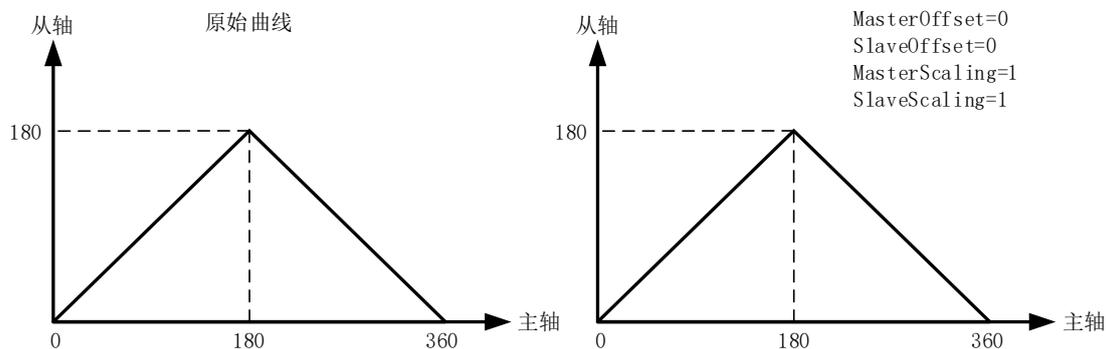
### ● 凸轮关系曲线变换

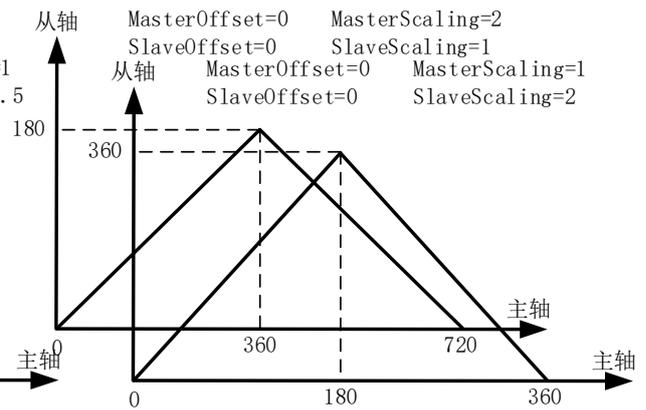
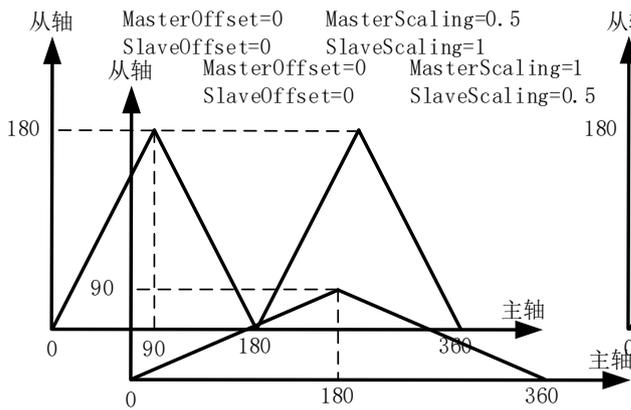
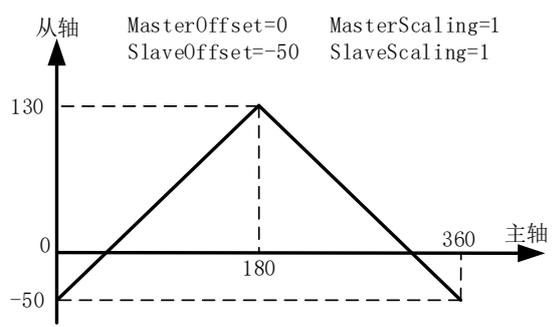
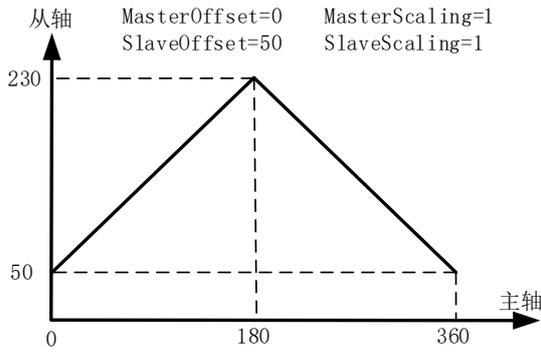
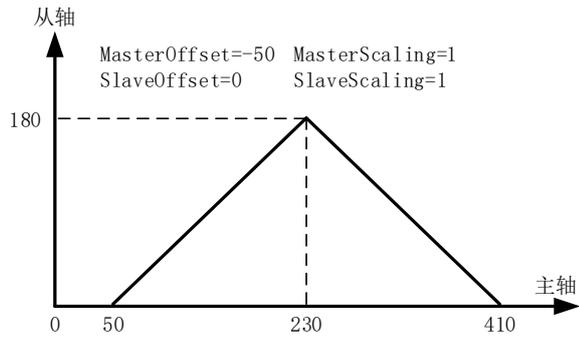
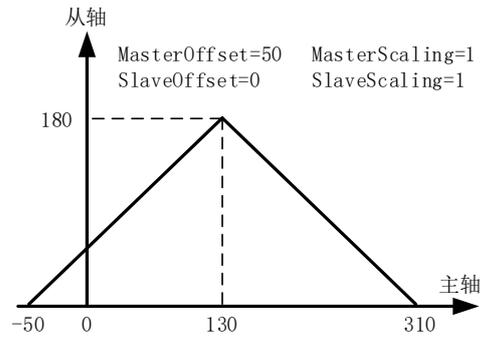
**MasterOffset：**原始凸轮关系曲线在主轴方向进行偏移。

**SlaveOffset：**原始凸轮关系曲线在从轴方向进行偏移 (从轴为相对模式时无效)。

**MasterScaling：**原始凸轮关系曲线在主轴方向进行比例变换。

**SlaveScaling：**原始凸轮关系曲线在从轴方向进行比例变换。





● 耦合动作模式

由于凸轮同步是凸轮相位同步，所以在耦合过程中（从不同步到同步），从轴可正向运动到满足同步要求的相位，亦可反向运动到满足同步要求的相位，通过参数 StartMode 即可设定该方向。

**StartMode=0 (最短距离)**：根据移动距离最小原则确定从轴在耦合过程中的运动方向。

**StartMode=1 (正向)**：从轴正向运动到满足同步要求的相位。

**StartMode=-1 (反向)**：从轴反向运动到满足同步要求的相位。

- **耦合动作补偿**

耦合过程中，从轴追赶主轴的快慢可通过补偿参数进行调节。补偿参数 Velocity、Acceleration、Deceleration 值越大，从轴追赶动作越快。

- **周期执行凸轮**

**Periodic=TRUE (周期执行)**：凸轮将无限循环执行，直到凸轮同步被终止。

**Periodic=FALSE (非周期执行)**：凸轮执行完一个周期即结束（经过凸轮结束点），此后从轴与主轴脱离，不再跟随主轴运动。

- **重启该指令**

非周期凸轮，当一个凸轮周期结束时（EndOfProfile 变为 TRUE 后），该指令可以重新执行。周期性凸轮，该指令不能重新执行，可以执行另外一个该指令打断当前正在执行的指令。

- **其它指令执行时启动该指令**

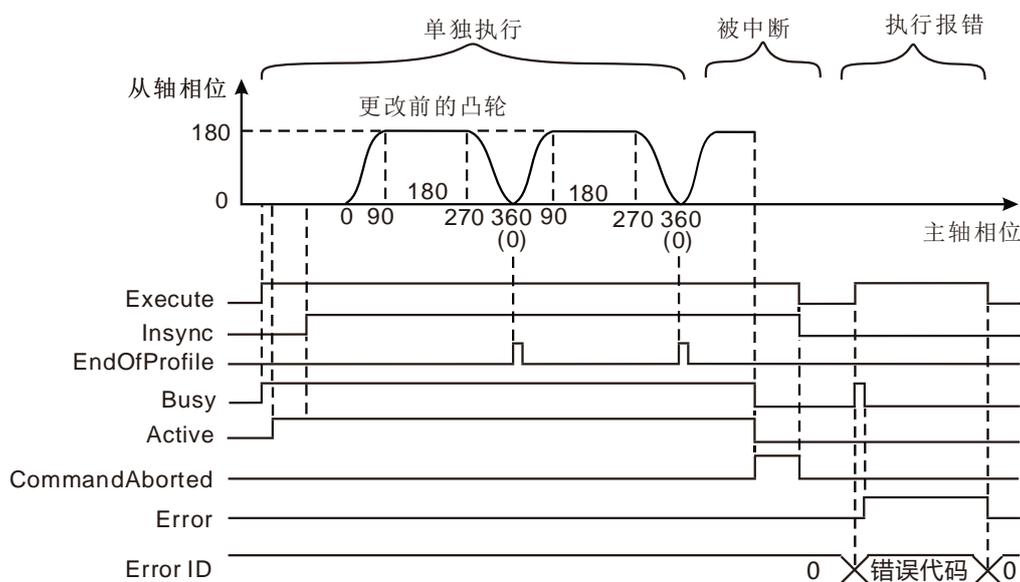
其它运动指令执行时，启动该指令，该指令和其它运动指令如何中继由该指令输入变量 BufferMode（缓存模式）的值决定。该指令输入变量 BufferMode（缓存模式）可以设置的值只可以设置为 mcAborting（中断）或者 mcBuffered（等待）。

缓存模式	含义
mcAborting (中断)	中断当前正在控制轴的指令，切换为缓存指令控制。
mcBuffered (等待)	等待当前指令执行完成时（如目标位置或者目标速度到达），切换为缓存指令控制轴。

- **该指令执行时启动其它指令**

该指令执行时启动其它运动指令，其它运动指令和该指令如何中继由其它指令的 BufferMode（缓存模式）引脚参数值决定，其它运动指令 BufferMode 的值只能选择中断或者等待；如果其它运动指令无 BufferMode 参数，一般为中断该指令。其它运动指令和该指令缓存时，其它指令执行的时机为该指令 EndOfProfile (凸轮周期结束) 变为 TRUE 时。

- **输出变量变化时序说明**



- **单独执行**

Execute（启动）由 FALSE 变为 TRUE 时，Busy（执行中）同时变为 TRUE，下一个周期 Active（控制中）变为 TRUE，从轴和主轴相位符合凸轮关系时，InSync 位变为 TRUE。该指令执行后，InSync 位可能和 Active 同时变为 TRUE，也可能在 Active 变为 TRUE 后很多个周期再变为 TRUE，和该指令的输入参数有关系。当

Execute 由 TRUE 变为 FALSE 时, Busy (执行中)、Active (控制中)、InSync (凸轮同步中) 状态位无变化。凸轮周期结束时, EndofProfile 会变为 TRUE 一个任务周期。

- **被中断**

该指令执行后被其它指令中断时, 该指令 CommandAborted (中断) 变为 TRUE, Busy (执行中)、Active (控制中)、InSync (凸轮同步中) 同时变为 FALSE; 当 Execute (执行中) 变为 FALSE 时, CommandAborted (中断) 同时变为 FALSE。

- **执行报错**

该指令输入变量值不在允许范围内时, 该指令 Execute (启动) 由 FALSE 变为 TRUE 时, 同时 Busy (执行中) 变为 TRUE, 下一个周期 Error (错误) 变为 TRUE, 同时 Busy (执行中) 变为 FALSE, ErrorID (错误码) 输出对应的错误码, 可通过 ErrorID (错误码) 的值, 查找发生问题原因。该指令 Execute (启动) 由 TRUE 变为 FALSE 时, 同时 Error (错误) 变为 FALSE, ErrorID (错误码) 的值变为 0。

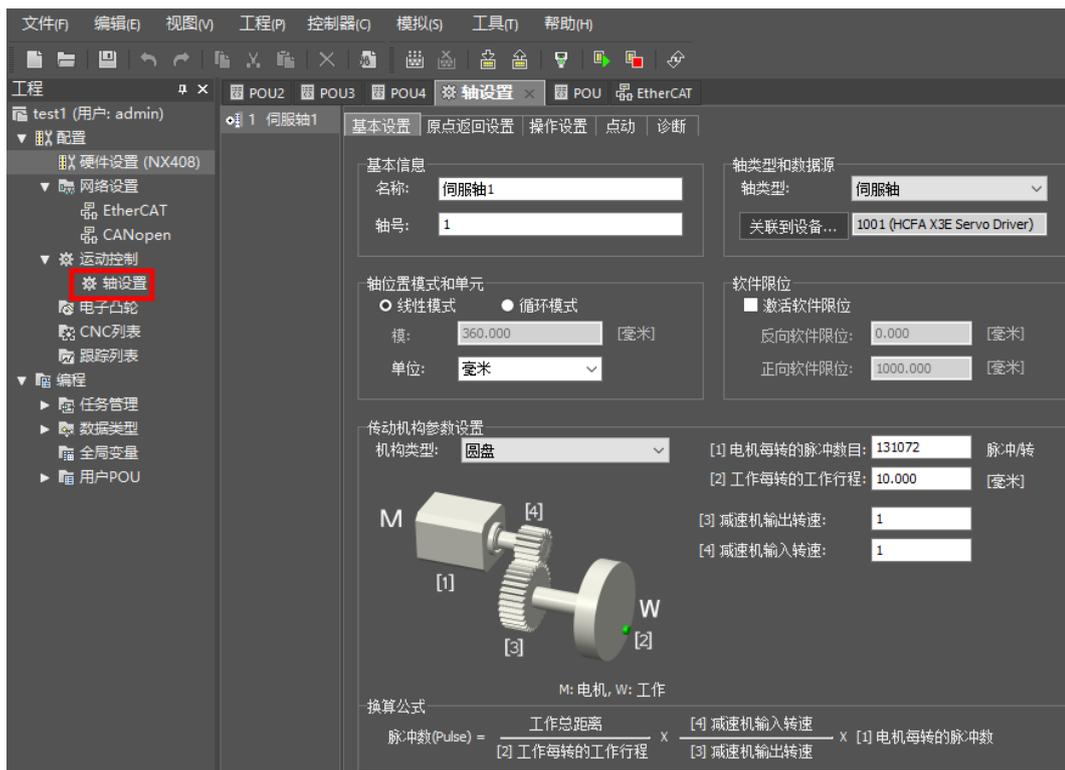
## ■ 示例程序

- **实现功能**

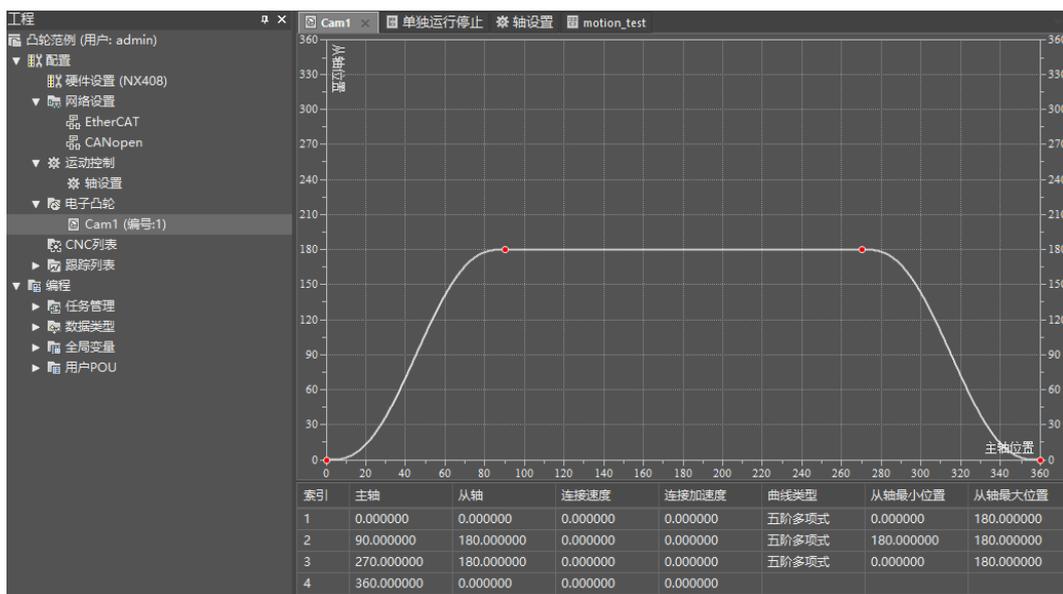
两个轴通过 MC\_CamIn 指令建立凸轮关系, 从轴跟随主轴根据软件中建立的凸轮运转。如果从轴需要和主轴解除凸轮关系时, 从轴执行 MC\_CamOut 指令解除凸轮关系。

- **轴参数及凸轮曲线设置**

主轴轴 1 和从轴轴 2 的轴参数相同, 轴 1 轴参数设置如下图所示。



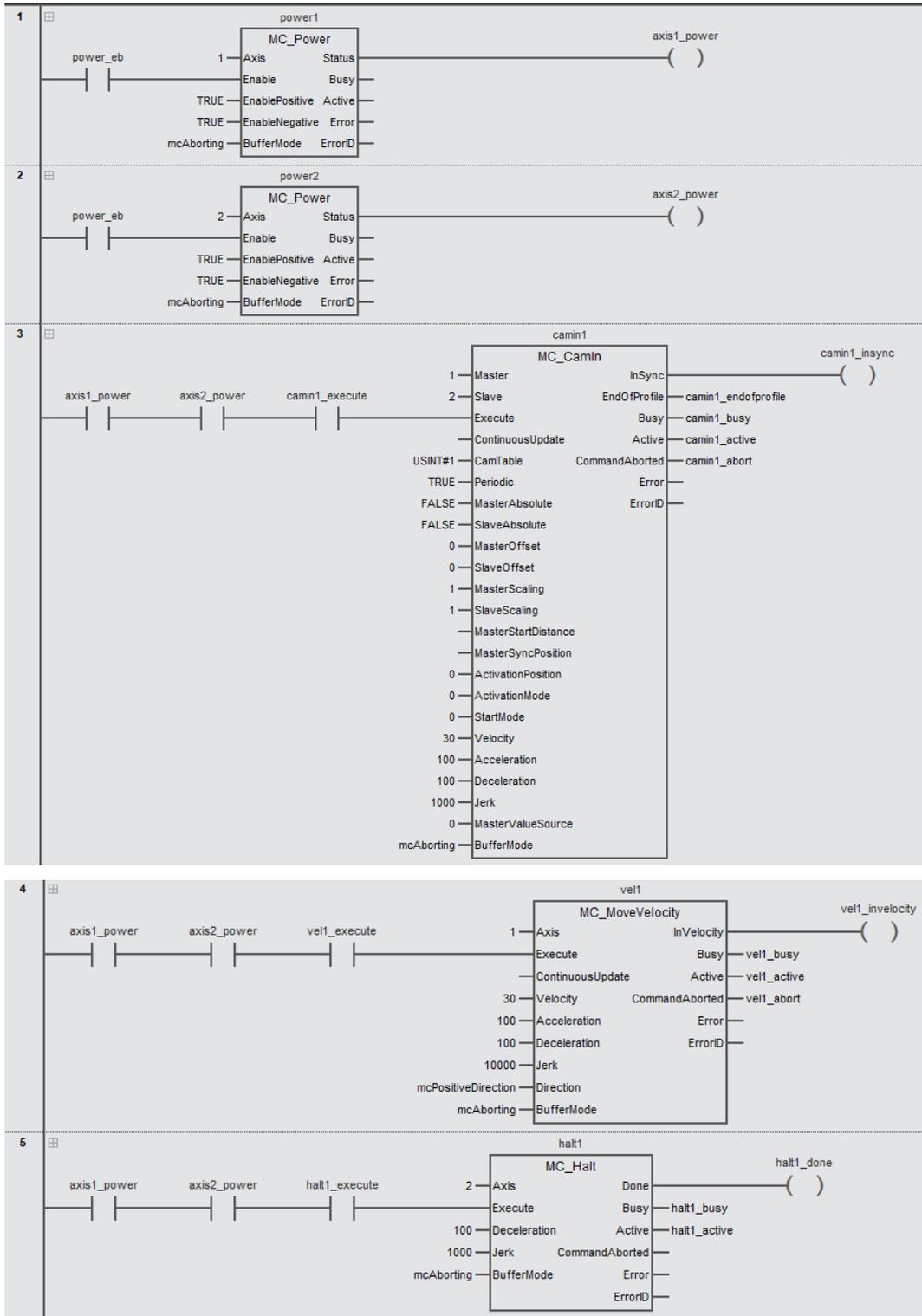
软件中建立的凸轮曲线如下图所示:



### ● 变量表

类别	名称	分配到	数据类型	初始值	注释
VAR	power_eb		BOOL		
VAR	power1		MC_Power		
VAR	axis1_power		BOOL		
VAR	power2		MC_Power		
VAR	axis2_power		BOOL		
VAR	camin1		MC_CamIn		
VAR	camin1_execute		BOOL		
VAR	camin1_insync		BOOL		
VAR	camin1_busy		BOOL		
VAR	camin1_active		BOOL		
VAR	camin1_abort		BOOL		
VAR	vel1		MC_MoveVelocity		
VAR	vel1_execute		BOOL		
VAR	vel1_done		BOOL		
VAR	vel1_busy		BOOL		
VAR	vel1_active		BOOL		
VAR	vel1_abort		BOOL		
VAR	halt1		MC_Halt		
VAR	halt 1_execute		BOOL		
VAR	halt 1_done		BOOL		
VAR	halt 1_busy		BOOL		
VAR	halt 1_active		BOOL		

● 梯形图(LD)



● 结构化文本(ST)

```

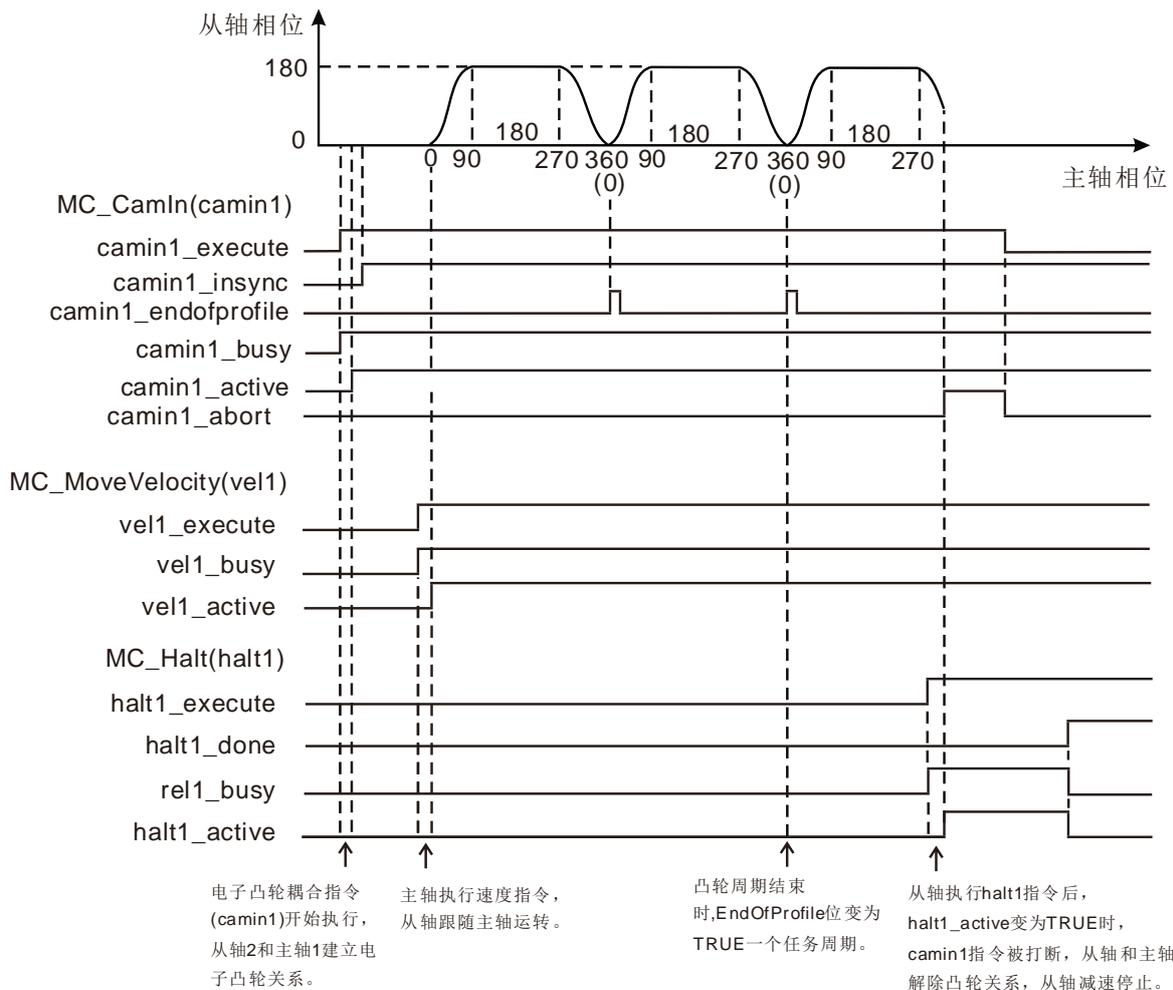
power1(
  Axis:=1,
  Enable:=power_eb,
  EnablePositive:=TRUE,
  EnableNegative:=TRUE,
  BufferMode:=mcAborting,
  Status=>axis1_power
);
    
```

```
power2(  
  Axis:=2 ,  
  Enable:=power_eb ,  
  EnablePositive:=TRUE ,  
  EnableNegative:=TRUE ,  
  BufferMode:=mcAborting ,  
  Status=>axis2_power  
);  
  
camin1(  
  Master:=1,  
  Slave:=2,  
  Execute:= axis1_power AND axis2_power AND camin1_execute ,  
  CamTable:= 1,  
  Periodic:=TRUE,  
  MasterAbsolute:=FALSE,  
  SlaveAbsolute:=FALSE,  
  MasterOffset:=0,  
  SlaveOffset:=0,  
  MasterScaling:=1,  
  SlaveScaling:=1,  
  ActivationPosition:=0,  
  ActivationMode:=0,  
  StartMode:=0,  
  Velocity:=30,  
  Acceleration:=100,  
  Deceleration:=100,  
  Jerk:=1000,  
  MasterValueSource:=0,  
  BufferMode:=mcAborting,  
  InSync=>camin1_insync,  
  Busy=>camin1_busy,  
  Active=>camin1_busy,  
  CommandAborted=>camin1_abort  
);  
  
vel1(  
  Axis:=1,  
  Execute:= axis1_power AND axis2_power AND vel1_execute,  
  Velocity:=30,  
  Acceleration:=100,  
  Deceleration:=100,  
  Jerk:=1000,  
  Direction:= mcPositiveDirection,  
  BufferMode:=mcAborting,  
  InVelocity=>vel1_invelocity,  
  Busy=>vel1_busy,  
  Active=>vel1_active,  
  CommandAborted=>vel1_abort  
);  
halt1(  
  Axis:=2,
```

```
Execute:=axis1_power AND halt1_execute,
Deceleration:=100,
Jerk:=1000,
BufferMode:=mcAborting,
Done=>halt1_done,
Busy=> halt1_busy,
Active=> halt1_active,
CommandAborted=> halt1_abort
);
```

● 程序说明

- 轴使能后, camIn1\_execute由FALSE变为TRUE时, 电子凸轮耦合指令 (camIn1) 开始执行, 第2个周期, camIn1\_active变为TRUE, camIn1\_insync变为TRUE时, 从轴2和主轴1建立凸轮关系。
- 建立的凸轮关系为主轴为相对模式, 从轴为相对模式, 主轴比例为1, 从轴比例为1。
- 从轴和主轴建立凸轮关系后, vel1\_execute由FALSE变为TRUE后, 主轴执行速度指令, 从轴将跟随主轴按规划的凸轮关系运动。
- 一次凸轮周期结束时, camIn1\_endofprofile由FALSE变为TRUE一个任务周期, 然后再变为FALSE。
- 从轴需要和主轴解除凸轮关系并停止时, 将halt1\_execute由FALSE变为TRUE, halt1指令开始执行, halt1\_active变为TRUE时, camIn1指令执行被打断, 从轴和主轴解除凸轮关系, 从轴减速停止。



## 5.5 MC\_CamOut (电子凸轮脱离)

该指令用于解除从轴与主轴之间通过MC\_CamIn指令建立电子凸轮关系，解除后从轴仍以当前速度继续运转。所属库：MotionControl

指令	名称	FB/FUN	梯形图样式	ST样式
MC_CamOut	电子凸轮脱离	FB		<pre>MC_CamOut_Instance ( Slave:= 参数 , Execute := 参数 , Done =&gt; 参数 , Busy =&gt; 参数 , CommandAborted =&gt; 参数 , Error =&gt; 参数 , ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Slave	从轴轴号	USINT	由机种决定	不可缺省	电子凸轮中从轴的轴号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	电子凸轮关系解除	BOOL	TRUE或FALSE	从轴和主轴的电子凸轮关系解除时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	从轴和主轴的电子凸轮关系解除时	<ul style="list-style-type: none"> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>
Busy	Execute 由FALSE变为TRUE时	<ul style="list-style-type: none"> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>
CommandAborted	指令的执行被中断时	<ul style="list-style-type: none"> <li>◆ CommandAborted为TRUE，Execute由TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时，该指令被其它指令中断时，CommandAborted变为TRUE一个周期后，变为FALSE</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Execute由TRUE变为FALSE时</li> </ul>

### ■ 功能说明

#### ● 基本功能说明

- 该指令用于中止MC\_CamIn指令的执行。该指令执行后，从轴和主轴通过MC\_CamIn建立的电子凸轮关系也随之解除，解除后从轴保持解除关系时的从轴速度。如从轴运转时执行该指令，该指令执行后，从轴仍会继续运转。如果想解除电子凸轮关系并希望从轴减速停止，可以直接对从轴执行MC\_Halt或者MC\_Stop指令

(不需要先执行MC\_CamOut指令, 再执行MC\_Halt或者MC\_Stop指令)。

➤ 该指令的执行对主轴的动作无影响。

- **重启该指令**

该指令无法重启。两个轴没有建立电子凸轮关系时, 执行该指令会报错。

- **其它指令执行时启动该指令**

只有 MC\_CamIn 指令执行时, 才可以执行该指令; 其它运动指令执行时, 执行该指令会报错。

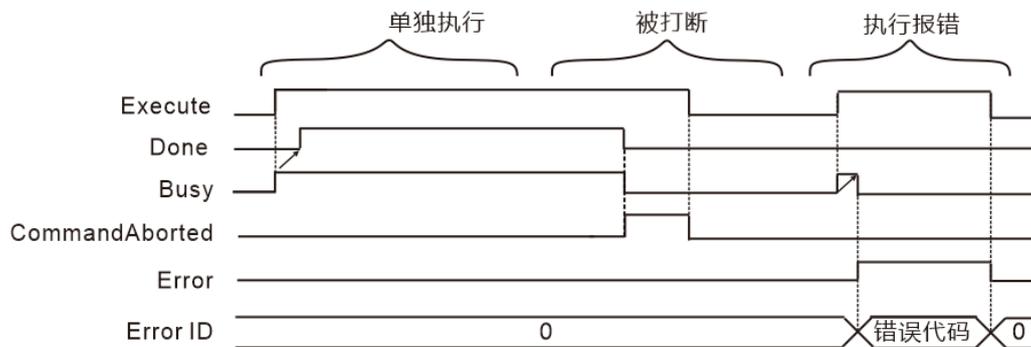
- **该指令执行时启动其它指令**

该指令执行时启动其它运动指令, 其它运动指令和该指令如何中继由其它指令的 BufferMode (缓存模式) 参数值决定, 其它运动指令 BufferMode 的值只能选择中断和等待; 如果其它运动指令无 BufferMode 参数, 一般为中断该指令。其它运动指令和该指令缓存时, 其它指令执行的时机为该指令 Done(电子凸轮关系解除)变为 TRUE 时。

- **异常排除**

该指令执行时, 如果输入变量不合法, 该指令 Error 变为 TRUE, ErrorID 中有对应的错误码, 可以通过该错误码判断错误原因。该指令执行时, 如遇到异常 (如轴报警等), 该指令也会报错, 轴会立即停止。

### ■ 输出变量变化时序说明



- **单独执行**

Execute (启动) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE, 下一个周期 Done (电子凸轮关系解除) 变为 TRUE, Busy (执行中) 仍为 TRUE, 轴的命令速度保持脱离前的速度不变。当 Execute (启动) 变为 FALSE 时, Busy (执行中) 和 Done (电子凸轮关系解除) 仍为 TRUE。

- **被中断**

该指令执行后被其它指令中断时, 该指令 CommandAborted (中断) 变为 TRUE, Busy (执行中) 和 Done (电子凸轮关系解除) 同时变为 FALSE; 当 Execute (执行中) 变为 FALSE 时, CommandAborted (中断) 同时变为 FALSE。

- **执行报错**

该指令输入变量值不在允许范围内时, 该指令 Execute (启动) 由 FALSE 变为 TRUE 时, 同时 Busy (执行中) 变为 TRUE, 下一个周期 Error (错误) 变为 TRUE, 同时 Busy (执行中) 变为 FALSE, ErrorID (错误码) 输出对应的错误码, 可通过 ErrorID (错误码) 的值, 查找发生问题原因。该指令 Execute (启动) 由 TRUE 变为 FALSE 时, 同时 Error (错误) 变为 FALSE, ErrorID (错误码) 的值变为 0。

该指令用于设置指定凸轮表中指定凸轮点的数据。所属库：MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_SetCamPoint	更改指定凸轮点数据	FB		<pre>MC_SetCamPoint_Instance ( Execute := 参数 , CamTable:= 参数 , Index:= 参数 , MasterPos:= 参数 , SlavePos := 参数 , Velocity:= 参数 , Acceleration:= 参数 , Done=&gt; 参数 , Busy=&gt; 参数 , Error =&gt; 参数 , ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
CamTable	电子凸轮表编号	USINT	由机种决定	不可缺省	定义主轴和从轴关系的凸轮表
Index	凸轮点编号	UINT	由机种决定	不可缺省	凸轮点在凸轮表中的编号
MasterPos	主轴相位	LREAL	正数、0	0	凸轮点在凸轮表中的主轴相位
SlavePos	从轴相位	LREAL	正数、0、负数	0	凸轮点在凸轮表中的从轴相位
Velocity	连接速度	LREAL	正数、0、负数	0	凸轮点处从轴速度和主轴速度比例值
Acceleration	连接加速度	LREAL	正数、0、负数	0	凸轮点处从轴加速度和主轴加速度比例值

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	执行完成时	<ul style="list-style-type: none"> <li>◆ Done为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE</li> </ul>
Busy	Execute的上升沿	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> </ul>
Error	指令输入变量值不在允许的范围	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Execute从TRUE变为FALSE时</li> </ul>

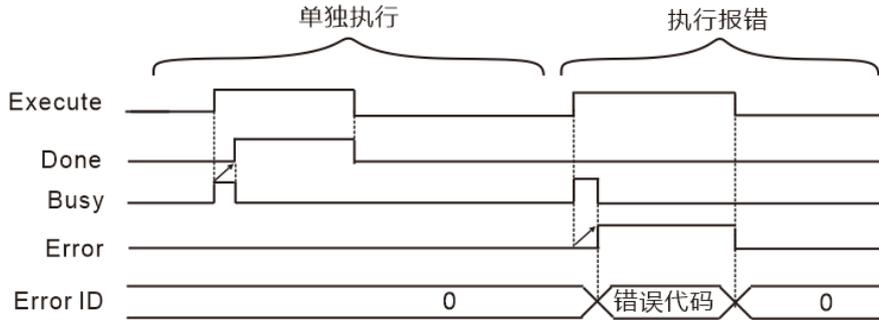
### ■ 功能说明

#### ● 基本功能说明

该指令用于更改指定凸轮表中指定凸轮点的数据，凸轮点的数据包括凸轮点的主轴相位、从轴相位、连接速度、连接加速度。该指令执行后，更改的凸轮点的数据不会立即生效，须搭配 MC\_ChangeCamCurve 指令和

MC\_CamIn 指令使用。该指令执行完成后，先执行 MC\_ChangeCamCurve 指令，再执行 MC\_CamIn 指令，该指令更改的凸轮点信息在 MC\_CamIn 指令执行后立即生效；MC\_CamIn 指令执行后，先执行该指令，再执行 MC\_ChangeCamCurve 指令，该指令更改的凸轮点信息在当前凸轮周期执行完后生效（轴相位经过凸轮主轴终点相位后生效）。

■ 输出变量变化时序说明



● 单独执行

Execute（启动）由 FALSE 变为 TRUE 时，Busy（执行中）同时变为 TRUE，下一个周期 Done（完成）变为 TRUE，同时 Busy（执行中）变为 FALSE。当 Execute（启动）变为 FALSE 时，Done（完成）同时变为 FALSE。

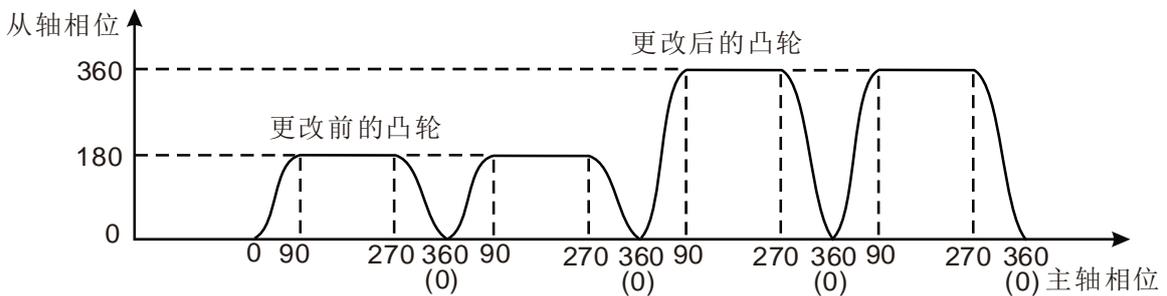
● 执行报错

该指令输入变量值不在允许范围内时，该指令 Execute（启动）由 FALSE 变为 TRUE 时，同时 Busy（执行中）变为 TRUE，下一个周期 Error（错误）变为 TRUE，同时 Busy（执行中）变为 FALSE，ErrorID（错误码）输出对应的错误码，可通过 ErrorID（错误码）的值，查找发生问题原因。该指令 Execute（启动）由 TRUE 变为 FALSE 时，同时 Error（错误）变为 FALSE，ErrorID（错误码）的值变为 0。

■ 示例程序

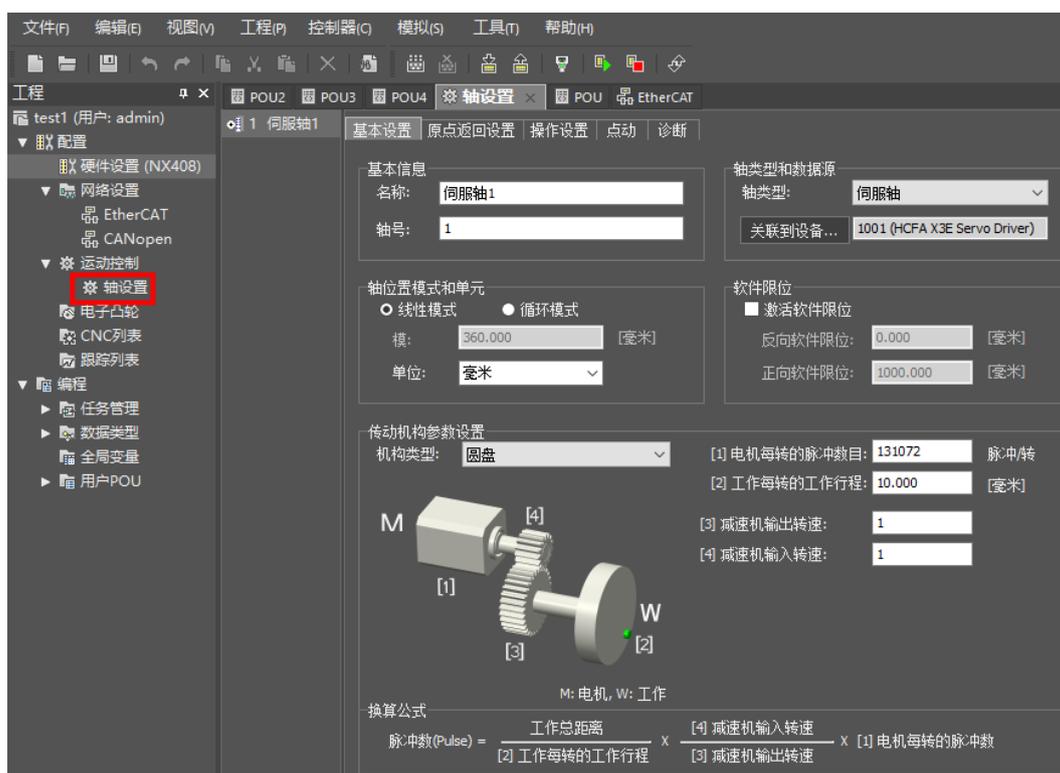
● 实现功能

凸轮运行过程中更改凸轮关键点数据，新更改的凸轮数据在当前凸轮周期结束后生效。从轴和主轴建立电子凸轮关系后，通过更改凸轮关键点数据来更改凸轮曲线。可以通过该指令和 MC\_ChangeCamCurve 指令配合使用更改凸轮关键点数据。

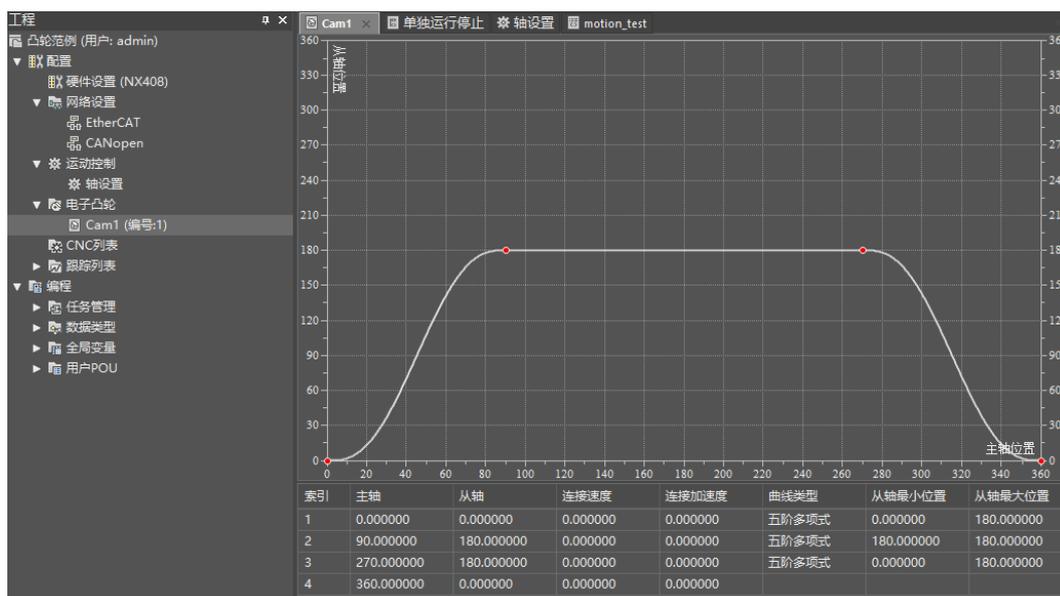


● 轴参数及凸轮曲线设置

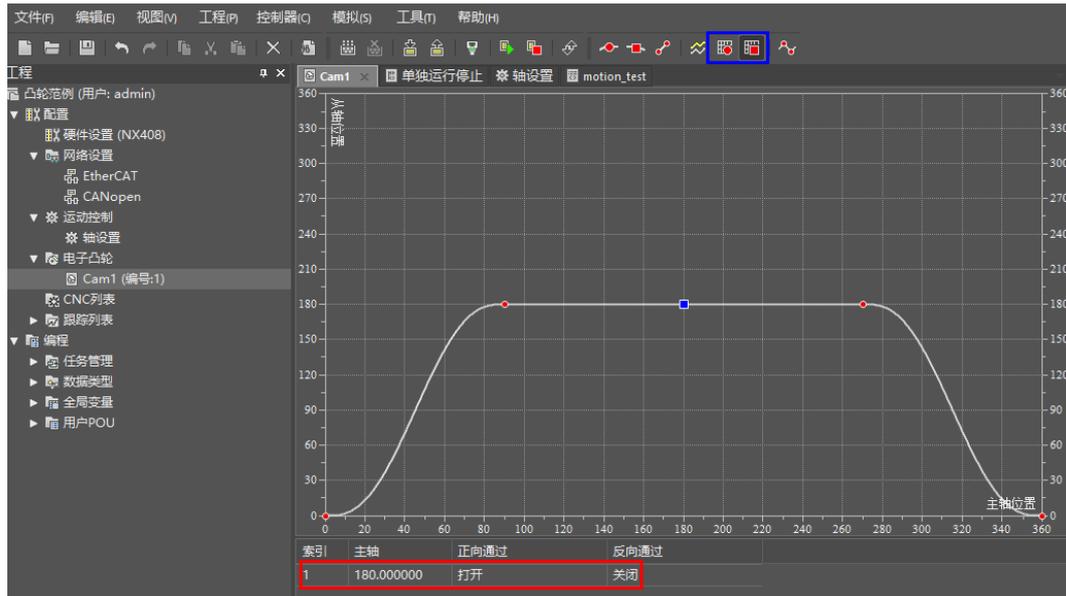
主轴轴 1 和从轴轴 2 的轴参数相同，轴 1 轴参数设置如下图所示。



软件中建立的凸轮曲线如下图所示:



凸轮曲线中添加的挺杆点入下图红色方框处所示，单击下图蓝色方框处可以添加和显示挺杆点。

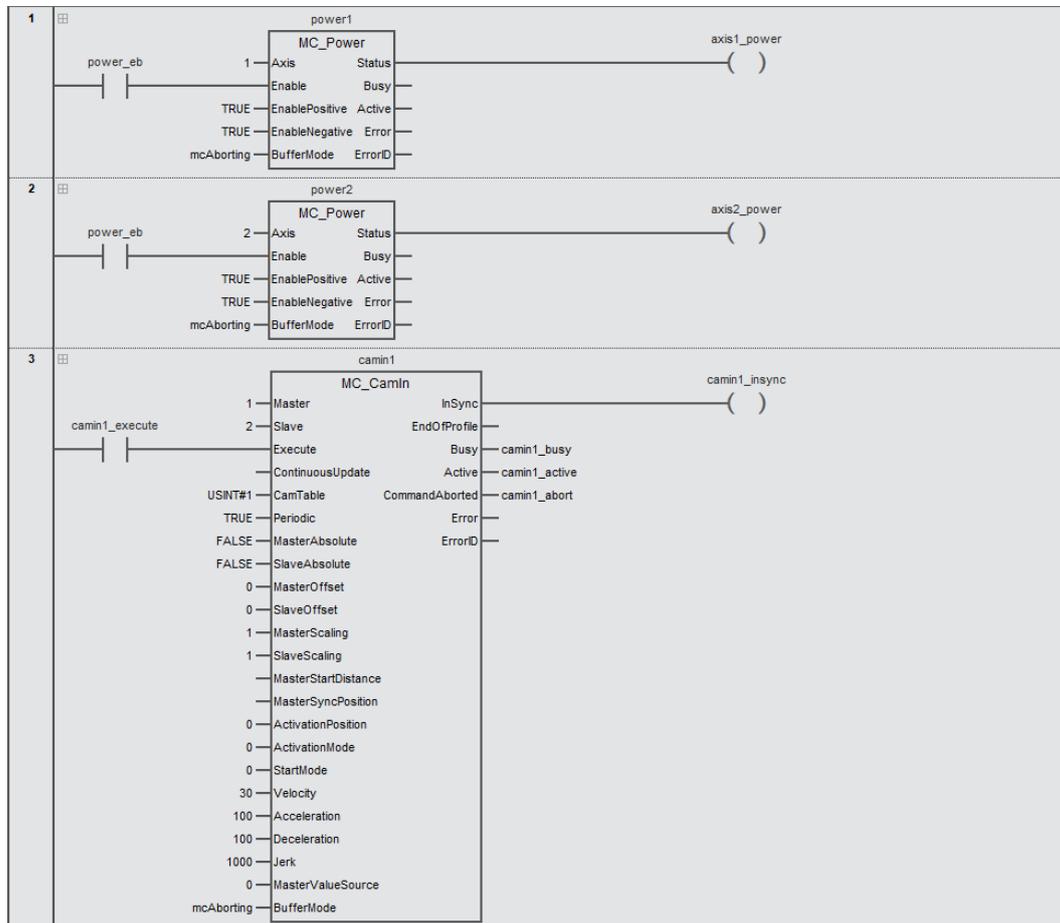


● 变量表

类别	名称	分配到	数据类型	初始值	注释
VAR	power_eb		BOOL		
VAR	powe1		MC_Power		
VAR	axis1_power		BOOL		
VAR	power2		MC_Power		
VAR	axis2_power		BOOL		
VAR	camin1		MC_CamIn		
VAR	camin1_execute		BOOL		
VAR	gearIn1_Ingear		BOOL		
VAR	camin1_insync		BOOL		
VAR	camin1_busy		BOOL		
VAR	camin1_active		BOOL		
VAR	camin1_abort		BOOL		
VAR	changecam1		MC_ChangeCamCurve		
VAR	changecam1_execute		BOOL		
VAR	changecam1_done		BOOL		
VAR	changecam1_busy		BOOL		
VAR	vel1		MC_MoveVelocity		
VAR	vel1_execute		BOOL		
VAR	vel1_done		BOOL		
VAR	vel1_busy		BOOL		
VAR	vel1_active		BOOL		
VAR	vel1_abort		BOOL		
VAR	setcam1		ARRAY[1..4] OF MC_SetCamPoint		
VAR	setcam1_execute		BOOL		
VAR	masterpos		ARRAY[1..4] OF LREAL	[0,90,270,360]	
VAR	slavepos		ARRAY[1..4] OF LREAL	[0,360,360, 0]	
VAR	done_true		BOOL		
VAR	i		INT		
VAR	tappetstatus1		MC_GetCamTappetStatus		

VAR	tappetstatus1_vaid		BOOL		
VAR	tappetstatus1_busy		BOOL		
VAR	tappetstatus1_statue		BOOL		
VAR	start		BOOL		

● 梯形图(LD)





● 结构化文本(ST)

```

power1(
  Axis:=1,
  Enable:=power_eb,
  EnablePositive:=TRUE,
  EnableNegative:=TRUE,
  BufferMode:=mcAborting,
  Status=>axis1_power
);
    
```

```

power2(
  Axis:=2,
  Enable:=power_eb,
  EnablePositive:=TRUE,
  EnableNegative:=TRUE,
  BufferMode:=mcAborting,
  Status=>axis2_power
);
    
```

```

camin1(
  Master:=1,
  Slave:=2,
  Execute:=camin1_execute,
  CamTable:=USINT#1,
  Periodic:=TRUE,
  MasterAbsolute:=FALSE,
  SlaveAbsolute:=FALSE,
  MasterOffset:=0,
  SlaveOffset:=0,
    
```

```

MasterScaling:=1,
SlaveScaling:=1,
ActivationPosition:=0,
ActivationMode:=0,
StartMode:=0,
Velocity:=30,
Acceleration:=100,
Deceleration:=100,
Jerk:=1000,
MasterValueSource:=0,
BufferMode:=mcAborting,
InSync=>camin1_insync,
Busy=>camin1_busy,
Active=>camin1_busy,
CommandAborted=>camin1_abort
);
//读取挺杆点状态
tappetstatus1(
  Enable:=TRUE ,
  CamTable:=1 ,
  Index:=1 ,
  Vaild=>tappetstatus1_vaild ,
  Busy=>tappetstatus1_busy ,
  Status=>tappetstatus1_statue
);
IF tappetstatus1_statue THEN
  setcam1_execute:=TURE;
END_IF;
//更改凸轮点
done_true:=TRUE;
FOR i:=2 TO 3 DO
  setcam1[i](
    Execute:=setcam1_execute AND start,
    CamTable:=1,
    Index:=i,
    MasterPos:=masterpos[i],
    SlavePos:=slavepos[i],
    Velocity:=0,
    Acceleration:=0
  );
  done_true:=done_trueandsetcam1[i].Done;
END_FOR;
IFEDGEPOS(done_true)THEN
  setcam1_execute:=FALSE;
  changecam1_execute:=TRUE;
END_IF;
//为更改凸轮点做准备
changecam1(
  Execute:=changecam1_execute,
  CamTable:=1,
  Done=>changecam1_done,
  Busy=>changecam1_busy
);

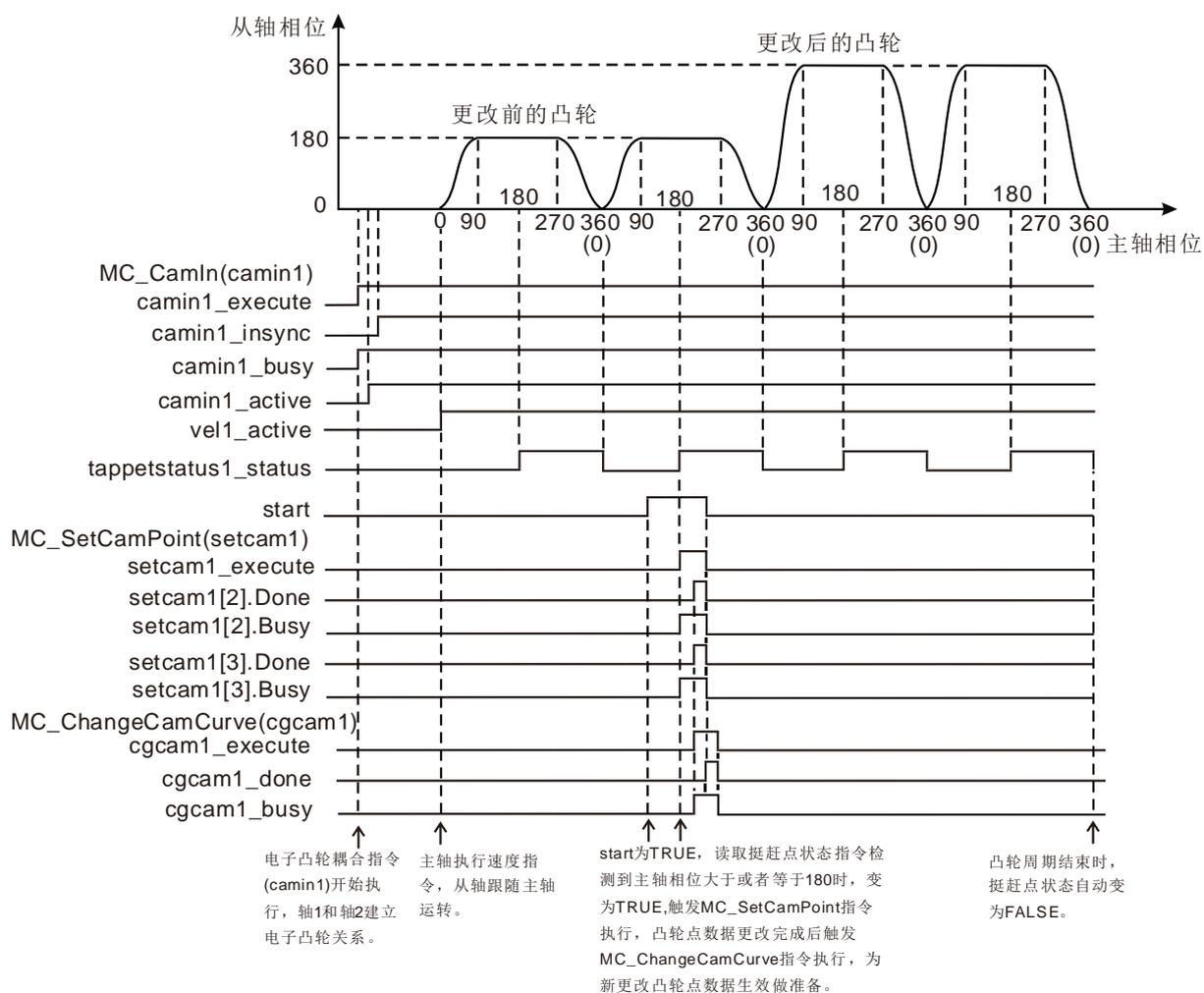
```

```
IF changecam1_done THEN
    changecam1_execute:=FALSE;
    start:=FALSE;
END_IF;

vel1(
    Axis:=1,
    Execute:=axis1_power AND vel1_execute,
    Velocity:=30,
    Acceleration:=100,
    Deceleration:=100,
    Jerk:=1000,
    Direction:=1,
    BufferMode:=mcAborting,
    InVelocity=>vel1_invelocity,
    Busy=>vel1_busy,
    Active=>vel1_active,
    CommandAborted=>vel1_abort
);
```

#### ● 程序说明

- 从轴和主轴使能后，camin1\_execute变为TRUE时，电子凸轮耦合指令（camin1）开始执行，从轴和主轴通过电子凸轮耦合指令（camin1）建立电子凸轮关系。
- 从轴和主轴建立电子凸轮关系后，主轴执行速度指令，从轴跟随主轴运转。
- start为TRUE时，读取凸轮挺杆点状态指令（tappetstatus1）检测到主轴相位大于或者等于180时，挺杆点状态变量tappetstatus1\_statue 从FALSE变为TRUE时，触发MC\_SetCamPoint指令执行更改凸轮点数据。需要修改凸轮点数据全部更改完成后，触发MC\_ChangeCamCurve指令执行，当前凸轮周期结束后，新更改的凸轮点数据生效。
- 该示例程序中将第2个和第3个凸轮点从轴相位由180更改为360，需要更改的凸轮点的数据在masterpos和slavepos数组变量的初始值中赋值，详细请参考变量表。
- 当需要更改多个凸轮点数据时，可以将MC\_SetCamPoint指令实例化为数组，通过FOR循环更改凸轮点数据，如该示例程序中SetCam1的数据类型为ARRAY[1..4] OF MC\_SetCamPoint。



## 5.7 MC\_ChangeCamCurve (凸轮数据变更)

该指令为MC\_SetCamPoint设置的凸轮点的数据生效做准备。所属库：MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_ChangeCamCurve	凸轮点数据变更	FB		<pre>MC_ChangeCamCurve_Instance ( Execute := 参数, CamTable := 参数, Done =&gt; 参数, Busy =&gt; 参数, Error =&gt; 参数, ErrorID =&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
CamTable	电子凸轮表编号	USINT	由机种决定	不可缺省	定义主轴和从轴关系的凸轮表

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成时变为TRUE。
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE。
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE。
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码。值的含义请参考“ <a href="#">指令错误代码描述</a> ”。

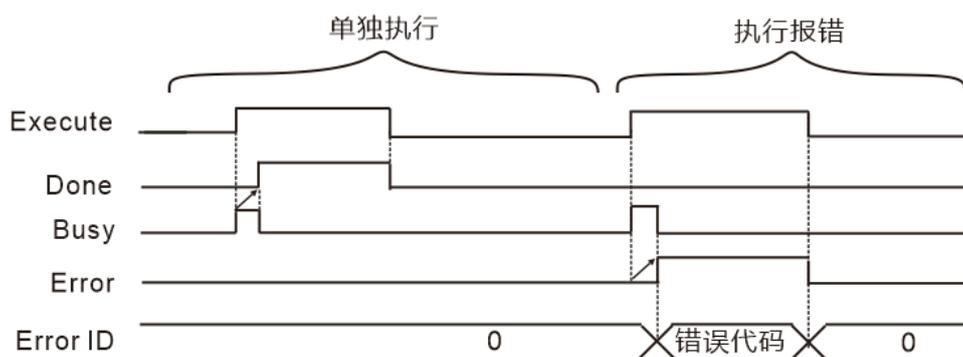
### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	执行完成时	<ul style="list-style-type: none"> <li>◆ Done为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE</li> </ul>
Busy	Execute的上升沿	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> </ul>
Error	指令输入变量值不在允许的范围	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Execute从TRUE变为FALSE时</li> </ul>

### ■ 功能说明

- 该指令为 MC\_SetCamPoint 设置的凸轮点的数据生效做准备。该指令需搭配 MC\_SetCamPoint 和 MC\_CamIn 指令使用。该指令执行后，会为 MC\_SetCamPoint 设置的所有凸轮点的数据生效做准备。
- MC\_CamIn 指令执行后，先执行 MC\_SetCamPoint 指令，再执行该指令，MC\_SetCamPoint 指令更改的凸轮点信息在当前凸轮周期执行完后生效（轴相位经过凸轮主轴终点相位后生效）。先执行 MC\_SetCamPoint 指令，再执行该指令，再执行 MC\_CamIn 指令，MC\_SetCamPoint 指令更改的凸轮点信息在 MC\_CamIn 指令触发执行时立即生效。

### ■ 输出变量变化时序说明



- **单独执行**

Execute (启动) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE, 下一个周期 Done (完成) 变为 TRUE, 同时 Busy (执行中) 变为 FALSE。当 Execute (启动) 变为 FALSE 时, Done (完成) 同时变为 FALSE。

- **执行报错**

该指令输入变量值不在允许范围内时, 该指令 Execute (启动) 由 FALSE 变为 TRUE 时, 同时 Busy (执行中) 变为 TRUE, 下一个周期 Error (错误) 变为 TRUE, 同时 Busy (执行中) 变为 FALSE, ErrorID (错误码) 输出对应的错误码, 可通过 ErrorID (错误码) 的值, 查找发生问题原因。该指令 Execute (启动) 由 TRUE 变为 FALSE 时, 同时 Error (错误) 变为 FALSE, ErrorID (错误码) 的值变为 0。

- **示例程序**

示例程序请参考 [MC\\_SetCamPoint 指令中的示例程序](#)。

## 5.8 MC\_GetCamPoint (读取指定凸轮点数据)

该指令用于读取指定凸轮表中指定凸轮点的数据。所属库: MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_GetCamPoint	读取指定凸轮点数据	FB		<pre>MC_GetCamPoint_Instance ( Execute := 参数 , CamTable:= 参数 , Mode= 参数 , Index:= 参数 , Done=&gt; 参数 , Busy=&gt; 参数 , Error=&gt; 参数 , ErrorID=&gt; 参数 , MasterPos=&gt; 参数 , SlavePos =&gt; 参数 , Velocity=&gt; 参数 , Acceleration=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
CamTable	电子凸轮表编号	USINT	由机种决定	不可缺省	定义主轴和从轴关系的凸轮表
Mode	模式选择	BOOL	TRUE或FALSE	FALSE	当为FALSE时, 读取当前使用凸轮的凸轮点数据; 当为TRUE时, 读取备用凸轮的凸轮点数据。
Index	凸轮点编号	UINT	由机种决定	不可缺省	凸轮点在凸轮表中的编号

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时, 输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”
MasterPos	主轴相位	LREAL	正数、0	凸轮点在凸轮表中的主轴相位
SlavePos	从轴相位	LREAL	正数、负数、0	凸轮点在凸轮表中的从轴相位
Velocity	连接速度	LREAL	正数、负数、0	凸轮点处从轴速度和主轴速度比例值
Acceleration	连接加速度	LREAL	正数、负数、0	凸轮点处从轴加速度和主轴加速度比例值

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	执行完成时	<ul style="list-style-type: none"> <li>Done为TRUE, Execute从TRUE变为FALSE时</li> <li>指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE</li> </ul>
Busy	Execute的上升沿	<ul style="list-style-type: none"> <li>Done从FALSE变为TRUE时</li> <li>Error从FALSE变为TRUE时</li> </ul>
Error	指令输入变量值不在允许的范围	<ul style="list-style-type: none"> <li>Error为TRUE, Execute从TRUE变为FALSE时</li> </ul>

### ■ 功能说明

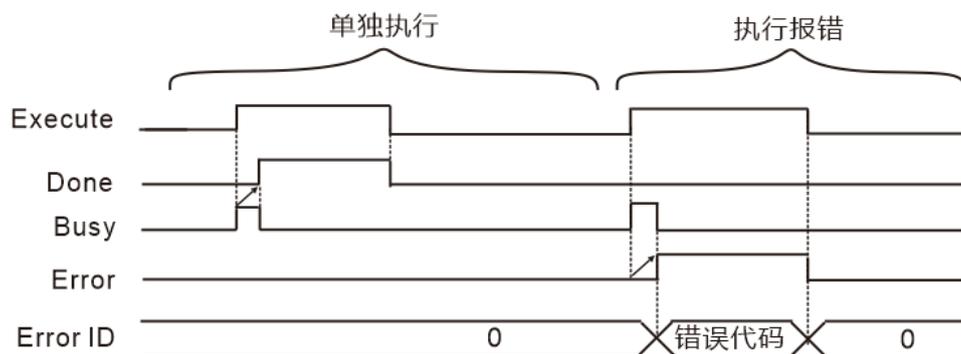
该指令用于读取指定凸轮表中指定凸轮点的数据。凸轮点的数据包括凸轮点的主轴相位、从轴相位、连接速度、连接加速度。

当 Mode 为 FALSE 时，读取到的是当前使用凸轮的凸轮点数据；

当 Mode 为 TRUE 时，读取到的是备用凸轮的凸轮点数据。

如凸轮周期为 360，当前执行凸轮点的主轴相位为 100、从轴相位为 100、连接速度为 0、连接加速度为 0，通过 MC\_SetCamPoint 和 MC\_ChangeCamCurve 指令更改凸轮点的主轴相位为 200、从轴相位为 200、连接速度为 0、连接加速度为 0。更改的凸轮点数据还没有生效时，该指令 Mode 为 FALSE 时，读取凸轮点的主轴相位为 100、从轴相位为 100、连接速度为 0、连接加速度为 0。该指令 Mode 为 TRUE 时，读取凸轮点的主轴相位为 200、从轴相位为 200、连接速度为 0、连接加速度为 0。当凸轮主轴位置运转超过一个凸轮周期后，新更改的凸轮点数据生效，该指令 Mode 为 FALSE 时，读取凸轮点的主轴相位为 200、从轴相位为 200、连接速度为 0、连接加速度为 0。

## ■ 输出变量变化时序说明



### ● 单独执行

Execute（启动）由 FALSE 变为 TRUE 时，Busy（执行中）同时变为 TRUE，下一个周期 Done（完成）变为 TRUE，同时 Busy（执行中）变为 FALSE。当 Execute（启动）变为 FALSE 时，Done（完成）同时变为 FALSE。

### ● 执行报错

该指令输入变量值不在允许范围内时，该指令 Execute（启动）由 FALSE 变为 TRUE 时，同时 Busy（执行中）变为 TRUE，下一个周期 Error（错误）变为 TRUE，同时 Busy（执行中）变为 FALSE，ErrorID（错误码）输出对应的错误码，可通过 ErrorID（错误码）的值，查找发生问题原因。该指令 Execute（启动）由 TRUE 变为 FALSE 时，同时 Error（错误）变为 FALSE，ErrorID（错误码）的值变为 0。

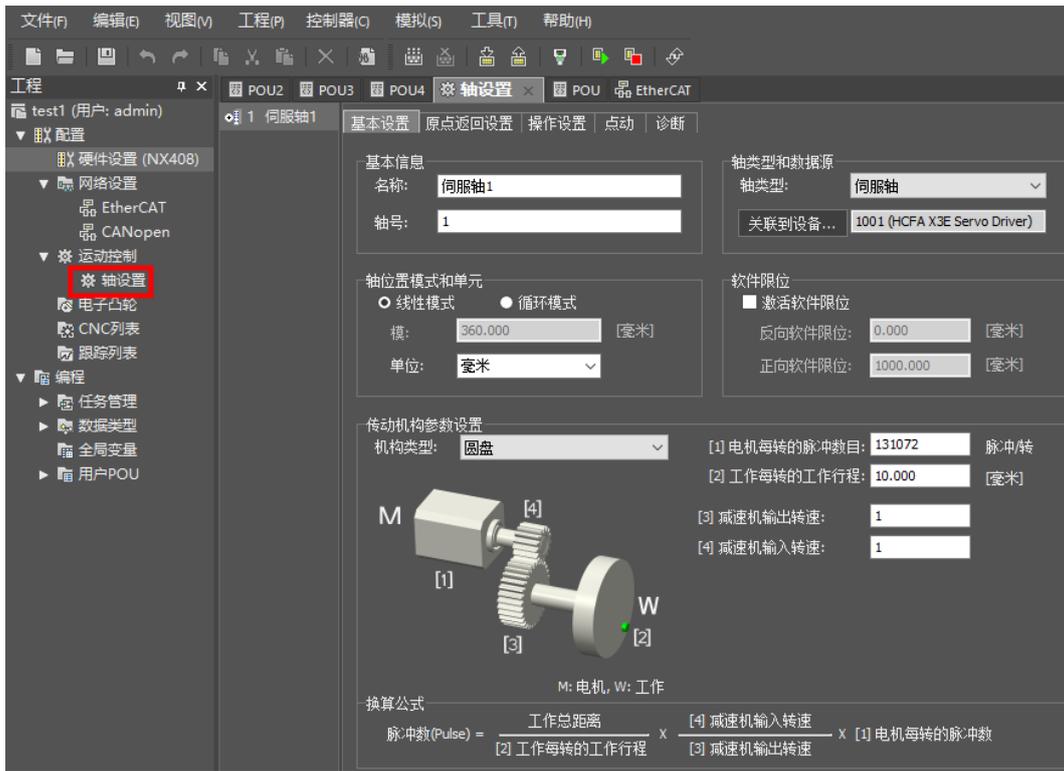
## ■ 示例程序

### ● 实现功能

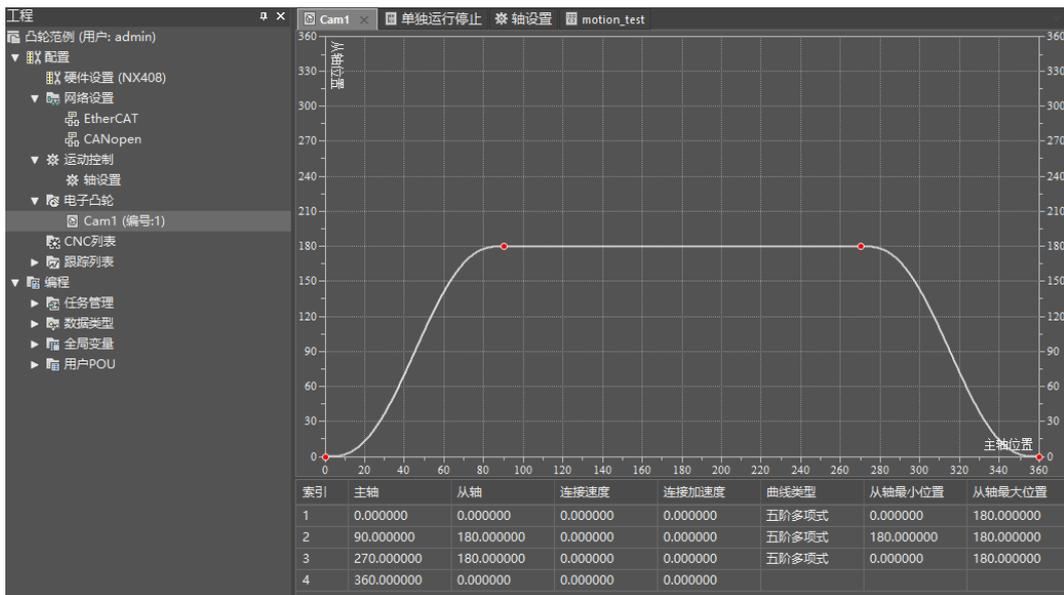
读取凸轮表中指定凸轮点数据。

### ● 轴参数及凸轮曲线设置

主轴轴 1 和从轴轴 2 的轴参数相同，轴 1 轴参数设置如下图所示。



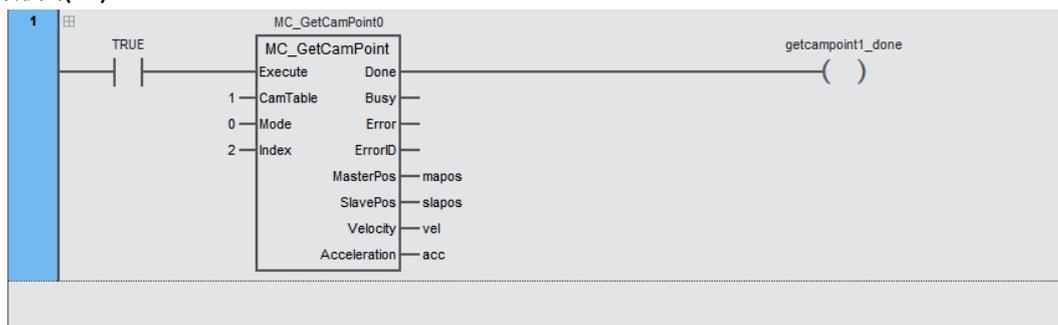
软件中建立的凸轮曲线如下图所示:



● 变量表

类别	名称	分配到	数据类型	初始值	注释
VAR	getcampoint1		MC_GetCamPoint		
VAR	getcampoint1_execute		BOOL		
VAR	getcampoint1_done		BOOL		
VAR	masterpos		LREAL		
VAR	slavepos		LREAL		
VAR	vel		LREAL		
VAR	acc		LREAL		

- 梯形图(LD)



- 结构化文本(ST)

```

getcampoint1
(Execute:= getcampoint1_execute ,
CamTable:=1 ,
Mode:=0 ,
Index:=2 ,
Done=> getcampoint1_done ,
MasterPos=> masterpos,
SlavePos=>slavepos,
Velocity=>vel ,
Acceleration=>acc
);

```

## 5.9 MC\_GetCamTappetStatus (读取凸轮挺杆点状态)

该指令用于读取轴的当前位置是否经过凸轮表中设定的挺杆点的位置。所属库: MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_GetCamTappetStatus	读取凸轮挺杆点状态	FB	<pre> MC_GetCamTappetStatus_Instance   Enable --- Valid   CamTable --- Busy   Index --- Error   ErrorID   Status           </pre>	<pre> MC_GetCamTappetStatus_Instance (   Enable := 参数 ,   CamTable:= 参数 ,   Index:= 参数 ,   Valid=&gt; 参数 ,   Busy=&gt; 参数 ,   Error =&gt; 参数 ,   ErrorID=&gt; 参数 ,   Status=&gt; 参数 );           </pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Enable	有效	BOOL	TRUE或FALSE	FALSE	设为TRUE, 读取指定凸轮挺杆点的状态 设为FALSE, 停止读取凸轮挺杆点的状态
CamTable	电子凸轮表编号	USINT	由机种决定	不可缺省	定义主轴和从轴关系的凸轮表
Index	挺杆点编号	UINT	1~8	不可缺省	挺杆点编号

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Valid	读取挺杆点状态有效	BOOL	TRUE或FALSE	该指令读取指定挺杆点状态时, 该参数值状态为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时, 输出错误代码 值的含义请参考“ <a href="#">指令错误代码描述</a> ”
Status	挺杆点状态	BOOL	TRUE或FALSE	指定挺杆点编号的状态

### ■ 输出变量刷新时机

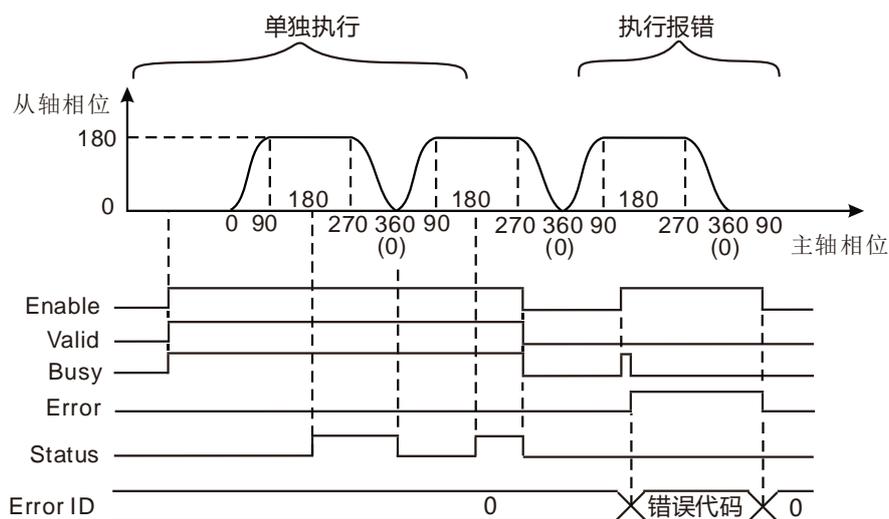
名称	变为TRUE的时机	变为FALSE的时机
Valid	Enable为TRUE时	Valid为TRUE, Enable从TRUE变为FALSE时
Busy	Enable为TRUE时	Enable从FALSE变为TRUE时 Error从FALSE变为TRUE时
Error	指令输入变量值不在允许的范围	Error为TRUE, Enable从TRUE变为FALSE时

### ■ 功能说明

#### ● 基本功能说明

该指令用于读取指定挺杆点的状态, 用于判断轴当前位置是否经过凸轮表中设定的挺杆点的位置。指定挺杆点状态为主轴相位正向或者反向通过挺杆点时的值, 挺杆点状态由软件中“电子凸轮”内挺杆点进行设置或者通过 MC\_SetCamTappet 指令进行设置。每个挺杆点的状态在主轴正向通过凸轮结束相位或者反向通过凸轮起始相位时变为 FALSE。

### ■ 输出变量变化时序说明



- **单独执行**

Enable (有效) 由 FALSE 变为 TRUE 时, Busy (执行中) 和 Valid(读取挺杆点状态有效)同时变为 TRUE, Status(挺杆点状态)可以根据挺杆点的设置输出对应的状态。如该示意图中, 设置挺杆点 1 的值为 180, 正向经过时为 TRUE, 则凸轮主轴相位大于或者等于 180 时, Status 位为 TRUE, 凸轮周期结束时, 自动变为 FALSE。当 Enable (有效) 变为 FALSE 时, Busy (执行中)、Valid(读取挺杆点状态有效)、Status(挺杆点状态)也同时变为 FALSE。

- **执行报错**

该指令输入变量值不在允许范围内时, 该指令 Enable (有效) 由 FALSE 变为 TRUE 时, 同时 Busy (执行中) 变为 TRUE, 下一个周期 Error (错误) 变为 TRUE, 同时 Busy (执行中) 变为 FALSE, ErrorID (错误码) 输出对应的错误码, 可通过 ErrorID (错误码) 的值, 查找发生问题原因。该指令 Enable (有效) 由 TRUE 变为 FALSE 时, 同时 Error (错误) 变为 FALSE, ErrorID (错误码) 的值变为 0。

- **示例程序**

示例程序请参考 [MC\\_SetCamPoint 指令中的示例程序](#)。

## 5.10 MC\_SetCamTappet (设置凸轮挺杆点数据)

该指令用于设置挺杆点数据。所属库：MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_SetCamTappet	设置凸轮挺杆点数据	FB		<pre>MC_SetCamTappet_Instance ( Execute := 参数 , CamTable:= 参数 , Index:= 参数 , MasterPos:= 参数 , PositiveCross= 参数 , NegativeCross= 参数 , Done=&gt; 参数 , Busy=&gt; 参数 , Error =&gt; 参数 , ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
CamTable	电子凸轮表编号	USINT	由机种决定	不可缺省	定义主轴和从轴关系的凸轮表
Index	挺杆点编号	UINT	由机种决定	不可缺省	挺杆点编号
MasterPos	主轴相位	LREAL	正数、0	不可缺省	设定挺杆点在凸轮表中的位置
PositiveCross	正向通过模式	TappetCrossMode	0: TappetDisable 1: TappetOn 2: TappetOff 3: TappetInvert	0: TappetDisable	轴正向通过挺杆点时的模式: 0: 无 1: 打开 2: 关闭 3: 转换
NegativeCross	反向通过模式	TappetCrossMode	0: TappetDisable 1: TappetOn 2: TappetOff 3: TappetInvert	0: TappetDisable	轴反向通过挺杆点时的模式: 0: 无 1: 打开 2: 关闭 3: 转换

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	执行完成时	<ul style="list-style-type: none"> <li>◆ Done为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE</li> </ul>

Busy	Execute的上升沿	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> </ul>
Error	指令输入变量值不在允许的范围	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Execute从TRUE变为FALSE时</li> </ul>

## ■ 功能说明

### ● 基本功能说明

➢ 该指令用于设定指定凸轮挺杆点设置数据。挺杆点设置数据包括该挺杆点的主轴相位、正向通过模式和反向通过模式。主轴相位正向通过和反向通过指定凸轮点的模式可以单独设置。

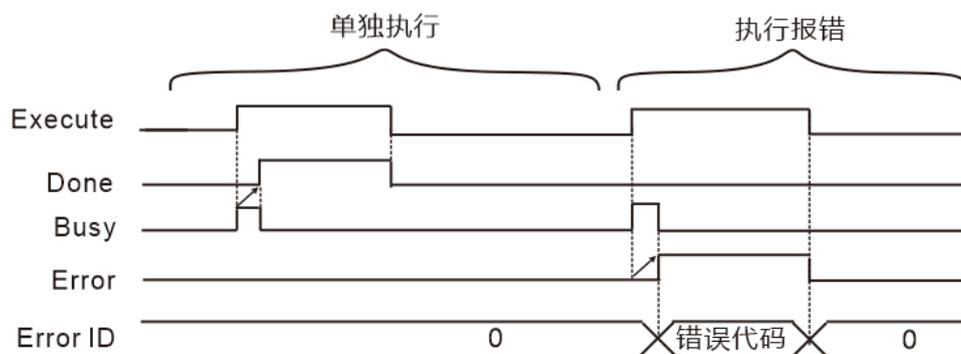
➢ 主轴相位正向通过指定凸轮点的模式及功能说明，如下表所示：

模式	功能	含义
TappetDisable	无	无功能，主轴正向通过指定的凸轮挺杆点时，挺杆点的状态无变化
TappetOn	打开	主轴正向通过指定的凸轮挺杆点时，挺杆点的状态为TRUE
TappetOff	关闭	主轴正向通过指定的凸轮挺杆点时，挺杆点的状态为FALSE
TappetInvert	转换	将原来的挺杆点状态反转（主轴正向通过前指定的凸轮挺杆点时FALSE，正向通过后为TRUE；主轴没有正向通过指定的凸轮挺杆点时TRUE，正向通过后为FALSE）

➢ 主轴相位反向通过指定凸轮点的模式及功能说明，如下表所示

模式	功能	含义
TappetDisable	无	无功能，主轴反向通过指定的凸轮挺杆点时，挺杆点的状态无变化
TappetOn	打开	主轴反向通过指定的凸轮挺杆点时，挺杆点的状态为TRUE
TappetOff	关闭	主轴反向通过指定的凸轮挺杆点时，挺杆点的状态为FALSE
TappetInvert	转换	将原来的挺杆点状态反转（主轴正向通过前指定的凸轮挺杆点时FALSE，正向通过后为TRUE；主轴没有正向通过指定的凸轮挺杆点时TRUE，正向通过后为FALSE）

## ■ 输出变量变化时序说明



### ● 单独执行

Execute（启动）由 FALSE 变为 TRUE 时，Busy（执行中）同时变为 TRUE，下一个周期 Done（完成）变为 TRUE，同时 Busy（执行中）变为 FALSE。当 Execute（启动）变为 FALSE 时，Done（完成）同时变为 FALSE。

### ● 执行报错

该指令输入变量值不在允许范围内时，该指令 Execute（启动）由 FALSE 变为 TRUE 时，同时 Busy（执行中）变为 TRUE，下一个周期 Error（错误）变为 TRUE，同时 Busy（执行中）变为 FALSE，ErrorID（错误码）

输出对应的错误码，可通过 ErrorID（错误码）的值，查找发生问题原因。该指令 Execute（启动）由 TRUE 变为 FALSE 时，同时 Error（错误）变为 FALSE，ErrorID（错误码）的值变为 0。

## 5.11 MC\_GetCamTappet (读取凸轮挺杆点设置数据)

该指令用于读取指定凸轮挺杆点设置数据。所属库：MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_GetCamTappet	读取挺杆数据	FB		<pre>MC_GetCamTappet_Instance ( Execute := 参数 , CamTable:=参数 , Index:=参数, Vaild=&gt;参数 , Busy=&gt;参数 , Error =&gt;参数 , ErrorID=&gt;参数 , MasterPos=&gt;参数 , PositiveMode=&gt;参数 , NegativeMode=&gt;参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
CamTable	电子凸轮表编号	USINT	由机种决定	不可缺省	用于设定主从轴建立凸轮关系所依据的凸轮表
Index	挺杆点编号	UINT	由机种决定	不可缺省	指定凸轮表中挺杆点编号

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”
MasterPos	主轴相位	LREAL	正数、负数、0	指定凸轮挺杆点对应的主轴相位
PositiveCross	正向通过模式	TappetCrossMode	0: TappetDisable 1: TappetOn 2: TappetOff 3: TappetInvert	轴正向通过挺杆点时的模式 0: 无 1: 打开 2: 关闭 3: 转换
NegativeCross	反向通过模式	TappetCrossMode	0: TappetDisable 1: TappetOn 2: TappetOff 3: TappetInvert	轴反向通过挺杆点时的模式 0: 无 1: 打开 2: 关闭 3: 转换

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Valid	Enable为TRUE时	◆ Valid为TRUE，Enable从TRUE变为FALSE时

Busy	Enable为TRUE时	<ul style="list-style-type: none"> <li>◆ Enable从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> </ul>
Error	指令输入变量值不在允许的范围	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Execute从TRUE变为FALSE时</li> </ul>

■ 功能说明

● 基本功能说明

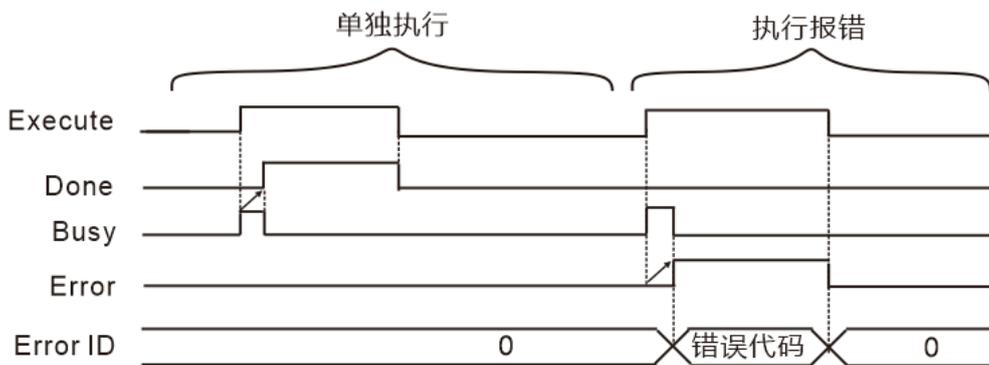
- 该指令用于读取指定凸轮挺杆点设置数据。挺杆点设置数据包括该挺杆点的主轴相位、正向通过模式和反向通过模式。主轴相位正向通过和反向通过指定凸轮点的模式可以单独设置。
- 主轴相位正向通过指定凸轮点的模式及功能说明，如下表所示：

模式	功能	含义
TappetDisable	无	无功能，主轴正向通过指定的凸轮挺杆点时，挺杆点的状态无变化
TappetOn	打开	主轴正向通过指定的凸轮挺杆点时，挺杆点的状态为TRUE
TappetOff	关闭	主轴正向通过指定的凸轮挺杆点时，挺杆点的状态为FALSE
TappetInvert	转换	将原来的挺杆点状态反转（主轴正向通过前指定的凸轮挺杆点时FALSE，正向通过后为TRUE；主轴没有正向通过指定的凸轮挺杆点时TRUE，正向通过后为FALSE）

- 主轴相位反向通过指定凸轮点的模式及功能说明，如下表所示

模式	功能	含义
TappetDisable	无	无功能，主轴反向通过指定的凸轮挺杆点时，挺杆点的状态无变化
TappetOn	打开	主轴反向通过指定的凸轮挺杆点时，挺杆点的状态为TRUE
TappetOff	关闭	主轴反向通过指定的凸轮挺杆点时，挺杆点的状态为FALSE
TappetInvert	转换	将原来的挺杆点状态反转（主轴正向通过前指定的凸轮挺杆点时FALSE，正向通过后为TRUE；主轴没有正向通过指定的凸轮挺杆点时TRUE，正向通过后为FALSE）

■ 输出变量变化时序说明



● 单独执行

Execute（启动）由 FALSE 变为 TRUE 时，Busy（执行中）同时变为 TRUE，下一个周期 Done（完成）变为 TRUE，同时 Busy（执行中）变为 FALSE。当 Execute（启动）变为 FALSE 时，Done（完成）同时变为 FALSE。

● 执行报错

该指令输入变量值不在允许范围内时，该指令 Execute（启动）由 FALSE 变为 TRUE 时，同时 Busy（执行中）变为 TRUE，下一个周期 Error（错误）变为 TRUE，同时 Busy（执行中）变为 FALSE，ErrorID（错误码）

输出对应的错误码，可通过 ErrorID（错误码）的值，查找发生问题原因。该指令 Execute（启动）由 TRUE 变为 FALSE 时，同时 Error（错误）变为 FALSE，ErrorID（错误码）的值变为 0。

## 5.12 MC\_AddCamTappet (新增凸轮挺杆点)

该指令用于在有的凸轮挺杆点表末尾加一个凸轮挺杆点。所属库：MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_AddCamTappet	新增凸轮挺杆点	FB		<pre>MC_AddCamTappet_Instance ( Execute := 参数 , CamTable:= 参数 , MasterPos:= 参数 , PositiveMode:= 参数 , NegativeMode:= 参数 , Done=&gt; 参数 , Busy=&gt; 参数 , Error =&gt; 参数 , ErrorID=&gt; 参数 , Index=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
CamTable	电子凸轮表编号	USINT	由机种决定	不可缺省	定义主轴和从轴关系的凸轮表
MasterPos	主轴相位	LREAL	正数、0	不可缺省	指定凸轮挺杆点对应的主轴相位
PositiveCross	正向通过模式	TappetCrossMode	0: TappetDisable 1: TappetOn 2: TappetOff 3: TappetInvert	0	轴正向通过挺杆点时的模式： 0: 无 1: 打开 2: 关闭 3: 转换
NegativeCross	反向通过模式	TappetCrossMode	0: TappetDisable 1: TappetOn 2: TappetOff 3: TappetInvert	0	轴反向通过挺杆点时的模式： 0: 无 1: 打开 2: 关闭 3: 转换

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	该输出变量为TRUE时表示指令执行完成。
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE。
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE。
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码。 值的含义请参考“ <a href="#">指令错误代码描述</a> ”。
Index	挺杆点编号	UINT	1~8	增加的挺杆点编号。

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	执行完成时。	<ul style="list-style-type: none"> <li>◆ Done为TRUE，Execute从TRUE变为FALSE时。</li> <li>◆ 指令执行且Execute为FALSE时，Done变为TRUE一个周期后，变为FALSE。</li> </ul>
Busy	Execute的上升沿。	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时。</li> <li>◆ Error从FALSE变为TRUE时。</li> </ul>
Error	指令输入变量值不在允许的范围。	<ul style="list-style-type: none"> <li>◆ Error为TRUE，Execute从TRUE变为FALSE时。</li> </ul>

## ■ 功能说明

### ● 基本功能说明

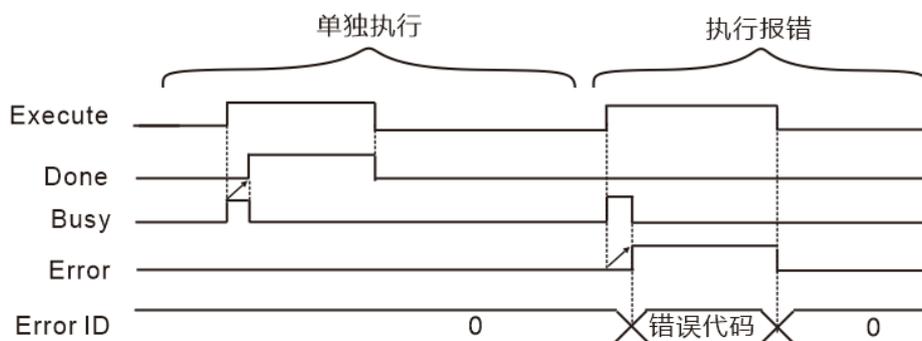
- 该指令用于在已有的凸轮挺杆点末尾增加一个凸轮挺杆点。新增的凸轮挺杆点包括主轴相位、正向通过模式和反向通过模式。该指令执行一次，新增加的凸轮挺杆点的编号在已有编号的最大值基础上依次增加。如指定凸轮表中有2个挺杆点，编号分别为1、2，该指令执行三次后，编号为3、4、5的挺杆点依次被增加，凸轮表中有编号为1、2、3、4、5的5个挺杆点。
- 主轴相位正向通过指定凸轮点的模式及功能说明，如下表所示：

模式	功能	含义
TappetDisable	无	无功能，主轴正向通过指定的凸轮挺杆点时，挺杆点的状态无变化。
TappetOn	打开	主轴正向通过指定的凸轮挺杆点时，挺杆点的状态为TRUE。
TappetOff	关闭	主轴正向通过指定的凸轮挺杆点时，挺杆点的状态为FALSE。
TappetInvert	转换	将挺杆点状态反转。（主轴正向通过前指定的凸轮挺杆点时FALSE，正向通过后为TRUE；主轴没有正向通过指定的凸轮挺杆点时TRUE，正向通过后为FALSE）

- 主轴相位反向通过指定凸轮点的模式及功能说明，如下表所示

模式	功能	含义
TappetDisable	无	无功能，主轴反向通过指定的凸轮挺杆点时，挺杆点的状态无变化。
TappetOn	打开	主轴反向通过指定的凸轮挺杆点时，挺杆点的状态为TRUE。
TappetOff	关闭	主轴反向通过指定的凸轮挺杆点时，挺杆点的状态为FALSE。
TappetInvert	转换	将挺杆点状态反转。（主轴正向通过前指定的凸轮挺杆点时FALSE，正向通过后为TRUE；主轴没有正向通过指定的凸轮挺杆点时TRUE，正向通过后为FALSE）

## ■ 输出变量变化时序说明



### ● 单独执行

Execute（启动）由 FALSE 变为 TRUE 时，Busy（执行中）同时变为 TRUE，下一个周期 Done（完成）变为 TRUE，同时 Busy（执行中）变为 FALSE。当 Execute（启动）变为 FALSE 时，Done（完成）同时变为 FALSE。

### ● 执行报错

该指令输入变量值不在允许范围内时，该指令 Execute（启动）由 FALSE 变为 TRUE 时，同时 Busy（执行中）变为 TRUE，下一个周期 Error（错误）变为 TRUE，同时 Busy（执行中）变为 FALSE，ErrorID（错误码）输出对应的错误码，可通过 ErrorID（错误码）的值，查找发生问题原因。该指令 Execute（启动）由 TRUE 变为 FALSE 时，同时 Error（错误）变为 FALSE，ErrorID（错误码）的值变为 0。

## 5.13 MC\_DeleteCamTappet（删除凸轮挺杆点）

该指令用于删除指定凸轮表中最后一个凸轮挺杆点。所属库：MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_DeleteCamTappet	删除凸轮挺杆点	FB		<pre>MC_DeleteCamTappet_Instance ( Execute := 参数 , CamTable:= 参数 , Done=&gt; 参数 , Busy=&gt; 参数 , Error =&gt; 参数 , ErrorID= &gt;参数 );</pre>

■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
CamTable	电子凸轮表编号	USINT	由机种决定	不可缺省	定义主轴和从轴关系的凸轮表

■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

■ 输出变量刷新时机

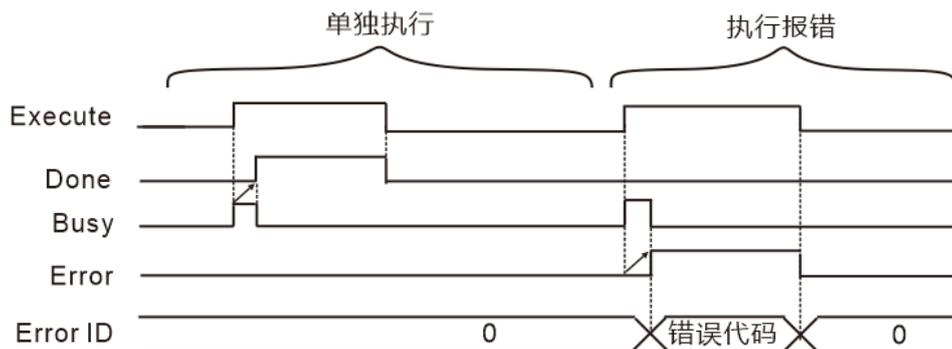
名称	变为TRUE的时机	变为FALSE的时机
Done	执行完成时	<ul style="list-style-type: none"> <li>◆ Done为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE</li> </ul>
Busy	Execute的上升沿	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> </ul>
Error	指令输入变量值不在允许的范围	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Execute从TRUE变为FALSE时</li> </ul>

■ 功能说明

● 基本功能说明

该指令用于删除指定凸轮表中最后一个凸轮挺杆点，即删除编号最大的挺杆点。该指令执行一次，删除一个凸轮表中编号最大的挺杆点。如指定凸轮表中有 5 个挺杆点，编号分别为 1、2、3、4、5，该指令执行三次后，编号为 5、4、3 的挺杆点依次被删除，凸轮表中只有编号为 1、2 的挺杆点。

## ■ 输出变量变化时序说明

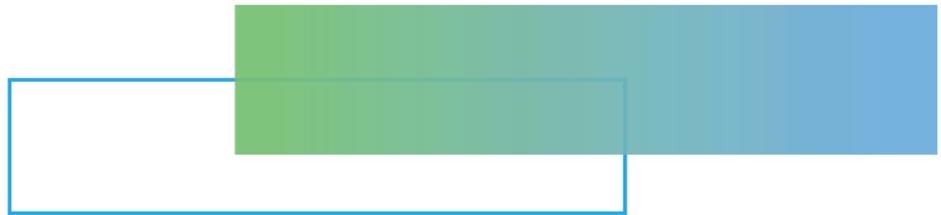


### ● 单独执行

Execute (启动) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE, 下一个周期 Done (完成) 变为 TRUE, 同时 Busy (执行中) 变为 FALSE。当 Execute (启动) 变为 FALSE 时, Done (完成) 同时变为 FALSE。

### ● 执行报错

该指令输入变量值不在允许范围内时, 该指令 Execute (启动) 由 FALSE 变为 TRUE 时, 同时 Busy (执行中) 变为 TRUE, 下一个周期 Error (错误) 变为 TRUE, 同时 Busy (执行中) 变为 FALSE, ErrorID (错误码) 输出对应的错误码, 可通过 ErrorID (错误码) 的值, 查找发生问题原因。该指令 Execute (启动) 由 TRUE 变为 FALSE 时, 同时 Error (错误) 变为 FALSE, ErrorID (错误码) 的值变为 0。



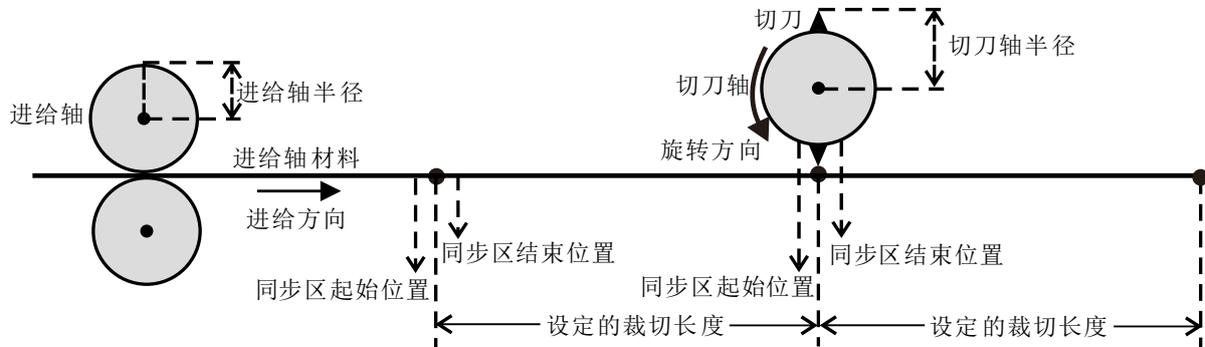
## 第六章 旋切工艺及相关指令



## 6.1 旋切工艺简介及适用场合

### ■ 旋切工艺

旋切工艺是指切刀往一个方向运转，对连续进给的材料按照用户设定的长度进行裁切。以用户指定的裁切长度为周期，在裁切时，即在同步区内，切刀的速度与进给材料速度同步，没有裁切时，切刀轴可以进行加速或减速，为下次在同步区和与进给材料速度同步做准备。旋切工艺示意图及常用参数如下图所示。



### ■ 适用场合

适用薄料切割的场合，如可广泛应用于包装、印刷、薄膜加工、纸巾生产等设备。

## 6.2 MC\_SetRotaryKnifeParameter (旋切机构参数设置)

该指令根据旋切机构设置相关参数，为切刀轴和进给轴建立旋切关系做准备。所属库：MotionControl

指令	名称	FB/FUN	梯形图样式	ST样式
MC_SetRotaryKnifeParameter	旋切机构参数设置	FB		<pre>MC_SetRotaryKnifeParameter _Instance ( Execute:= 参数 , KnifeRadius:= 参数 , KnifeNum:= 参数 , FeedRadius:= 参数 , CutLength:= 参数 , SyncStartPos:= 参数 , SyncStopPos:= 参数 , KnifeStartPos:= 参数 , FeedStartPos:= 参数 , ID:= 参数 , Done=&gt; 参数 , Busy=&gt; 参数 , Error=&gt; 参数 , ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
KnifeRadius	切刀轴半径	LREAL	正数	不可缺省	切刀轴圆心到切刀刀尖的距离 详细参考功能说明中示意图说明
KnifeNum	切刀数量	USINT	1~8	不可缺省	切刀轴上切刀的数量
FeedRadius	进给轴半径	LREAL	正数	不可缺省	进给轴半径 详细参考功能说明中示意图说明
CutLength	切长	LREAL	正数	不可缺省	设定的切割长度 详细参考功能说明中示意图说明
SyncStartPos	同步区起始位置	LREAL	正数	不可缺省	切刀轴和进给轴速度开始同步时的位置 详细参考功能说明中示意图说明
SyncStopPos	同步区结束位置	LREAL	正数	不可缺省	切刀轴和进给轴速度结束同步时的位置 详细参考功能说明中示意图说明
KnifeStartPos	保留	LREAL	保留	保留	保留
FeedStartPos	保留	LREAL	保留	保留	保留
ID	旋切编号	USINT	由机种决定	不可缺省	旋切编号

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	旋切参数设置完成时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码 值的含义请参考“ <a href="#">指令错误代码描述</a> ”

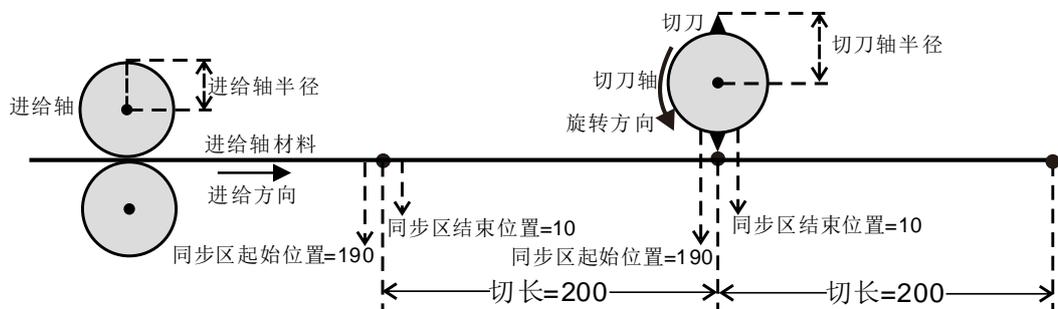
## ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	执行完成时	Done为TRUE, Execute从TRUE变为FALSE时 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE
Busy	Execute的上升沿	Done从FALSE变为TRUE时 Error从FALSE变为TRUE时
Error	指令输入变量值不在允许的范围	Error为TRUE, Execute从TRUE变为FALSE时

## ■ 功能说明

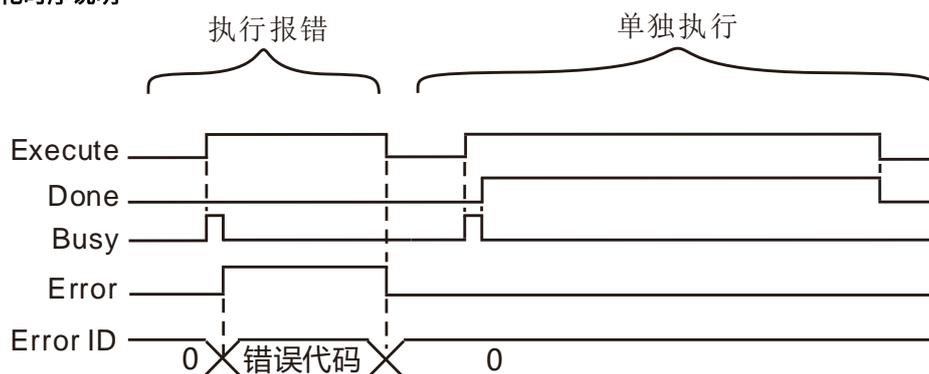
### ● 基本功能说明

- 该指令用于设置旋切系统相关参数，该指令输入变量的含义可以参考下图旋切关系示意图。切刀轴半径、进给轴半径、切刀数量等参数由实际机构决定，切长由客户实际应用需求决定。
- 进给轴和切刀轴没有建立旋切关系时，执行该指令后，再执行MC\_RotaryKnife\_In指令，设置的参数立即生效。进给轴和切刀轴建立旋切关系后，再执行该指令更改输入变量的值，如切长、同步区起始位置、同步区结束位置等参数，更改的值在当前旋切周期结束时生效，即在下个旋切周期开始时生效。



- 同步区起始位置 (SyncStartPos)：从同步区起始位置开始，切刀的速度与进给材料速度保持同步，一直到同步区结束位置 (SyncStopPos)。
- 同步区结束位置 (SyncStopPos)：从同步区结束位置开始，切刀的速度可以与进给材料速度不同步，可以根据需要进行加速或者减速，为下次在同步区和与进给材料速度同步做准备。

## ■ 输出变量变化时序说明



### ● 执行报错

该指令输入变量值不在允许范围内时，该指令 Execute (启动) 由 FALSE 变为 TRUE 时，同时 Busy (执行中) 变为 TRUE，下一个周期 Error (错误) 变为 TRUE，同时 Busy (执行中) 变为 FALSE，ErrorID (错误码) 输出对应的错误码，可通过 ErrorID (错误码) 的值，查找发生问题原因。该指令 Execute (启动) 由 TRUE 变为 FALSE 时，同时 Error (错误) 变为 FALSE，ErrorID (错误码) 的值变为 0。

### ● 单独执行

Execute (启动) 由 FALSE 变为 TRUE 时，Busy (执行中) 同时变为 TRUE，下一个周期 Done (完成) 变为 TRUE，旋切参数设置完成。当 Execute (启动) 变为 FALSE 时，Done (完成) 同时变为 FALSE。

### 6.3 MC\_RotaryKnife\_In (旋切耦合)

根据设置的旋切机构参数，切刀轴和进给轴建立控制器计算的旋切关系。所属库：MotionControl

指令	名称	FB/FUN	梯形图样式	ST样式
MC_RotaryKnife_In	旋切耦合	FB		<pre>MC_RotaryKnife_In_Instance ( Execute:=参数, KnifeAxis:=参数, FeedAxis:=参数, ID:= 参数, Done=&gt;参数, Busy=&gt;参数, Error=&gt;参数, ErrorID=&gt;参数 );</pre>

#### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
KnifeAxis	切刀轴轴号	USINT	由机种决定	不可缺省	指定切刀轴的轴号
FeedAxis	进给轴轴号	USINT	由机种决定	不可缺省	指定进给轴的轴号
ID	旋切编号	USINT	由机种决定	不可缺省	旋切关系编号

#### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	旋切关系建立后变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

#### ■ 输出变量刷新时机

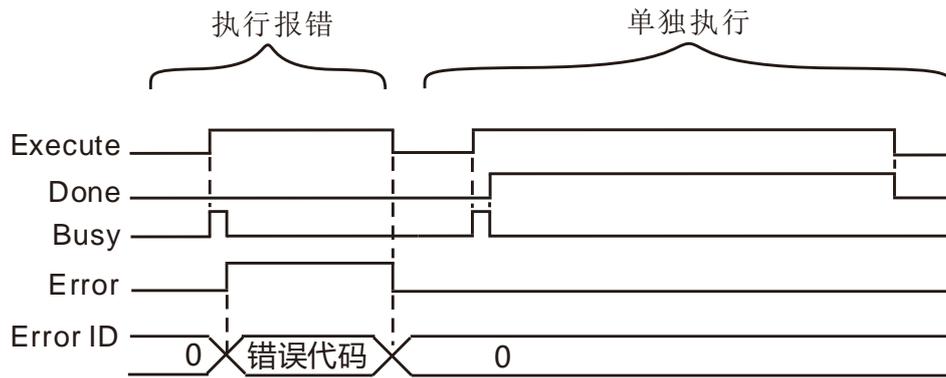
名称	变为TRUE的时机	变为FALSE的时机
Done	执行完成时	Done为TRUE，Execute从TRUE变为FALSE时 指令执行且Execute为FALSE时，Done变为TRUE一个周期后，变为FALSE
Busy	Execute的上升沿	Done从FALSE变为TRUE时 Error从FALSE变为TRUE时
Error	指令输入变量值不在允许的范围	Error为TRUE，Execute从TRUE变为FALSE时

#### ■ 功能说明

##### ● 基本功能说明

根据设置的旋切机构参数，切刀轴和进给轴建立控制器计算的旋切关系。建立旋切关系后，切都轴跟随进给轴周期性运转。

## ■ 输出变量变化时序说明



### ● 执行报错

该指令输入变量值不在允许范围内时，该指令 Execute（启动）由 FALSE 变为 TRUE 时，同时 Busy（执行中）变为 TRUE，下一个周期 Error（错误）变为 TRUE，同时 Busy（执行中）变为 FALSE，ErrorID（错误码）输出对应的错误码，可通过 ErrorID（错误码）的值，查找发生问题原因。该指令 Execute（启动）由 TRUE 变为 FALSE 时，同时 Error（错误）变为 FALSE，ErrorID（错误码）的值变为 0。

### ● 单独执行

Execute（启动）由 FALSE 变为 TRUE 时，Busy（执行中）同时变为 TRUE，下一个周期 Done（完成）变为 TRUE，旋切关系建立。当 Execute（启动）变为 FALSE 时，Done（完成）同时变为 FALSE。

## 6.4 MC\_RotaryKnife\_Out (旋切关系解除)

根据设置的旋切机构参数，切刀轴和进给轴脱离控制器计算的旋切关系。所属库：MotionControl

指令	名称	FB/FUN	梯形图样式	ST样式
MC_RotaryKnife_Out	旋切关系解除	FB		<pre>MC_RotaryKnife_Out_Instance ( Execute:= 参数 , KnifeAxis:= 参数 , ID:= 参数 , Done=&gt; 参数 , Busy=&gt; 参数 , Error=&gt; 参数 , ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
KnifeAxis	切刀轴轴号	USINT	由机种决定	不可缺省	指定切刀轴的轴号
ID	旋切编号	USINT	由机种决定	不可缺省	旋切关系编号

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	切刀轴和进给轴解除旋切关系并停止后变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

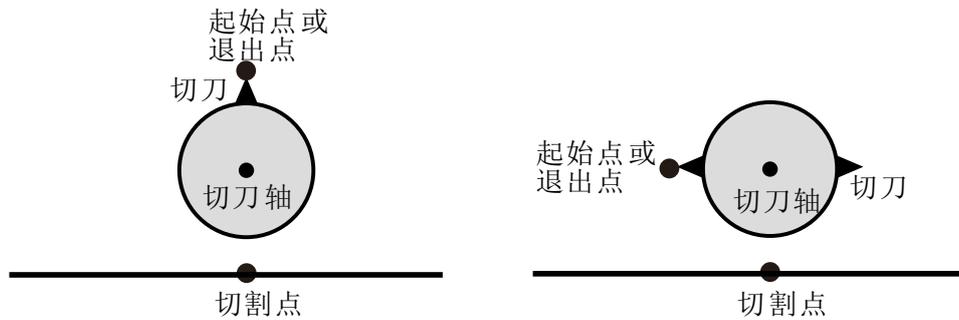
### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	执行完成时	Done为TRUE, Execute从TRUE变为FALSE时 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE
Busy	Execute的上升沿	Done从FALSE变为TRUE时 Error从FALSE变为TRUE时
Error	指令输入变量值不在允许的范围	Error为TRUE, Execute从TRUE变为FALSE时

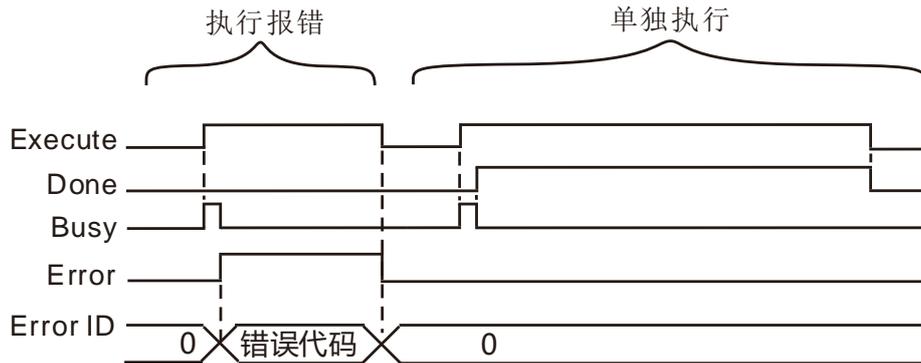
### ■ 功能说明

#### ● 基本功能说明

- 根据设置的旋切机构参数，切刀轴和进给轴解除控制器计算的旋切关系。切刀轴和进给轴解除旋切关系后停止的位置成为退出点，也是切刀轴和进给轴建立旋切关系前的起始点。下图所示为切刀轴有1把刀和2把刀时的退出点。
- 切刀轴和进给轴建立旋切关系后，进给轴运转时执行该指令，该指令执行后，切刀轴会跟随进给轴运转到建立旋切关系前的位置停止，防止切刀和进给轴机构所带的物质有接触，同时该指令的Done变为TRUE。
- 切刀轴和进给轴建立旋切关系后，主轴停止时执行该指令，切刀轴不会运转，也不会立即和进给轴脱离旋切关系，而是等待切刀轴运转至建立旋切关系前的位置停止并脱离旋切关系。
- 切刀轴和进给轴建立旋切关系后，才可以执行该指令，否则执行该指令会报错。



### ■ 输出变量变化时序说明



#### ● 执行报错

该指令输入变量值不在允许范围内时，该指令 Execute（启动）由 FALSE 变为 TRUE 时，同时 Busy（执行中）变为 TRUE，下一个周期 Error（错误）变为 TRUE，同时 Busy（执行中）变为 FALSE，ErrorID（错误码）输出对应的错误码，可通过 ErrorID（错误码）的值，查找发生问题原因。该指令 Execute（启动）由 TRUE 变为 FALSE 时，同时 Error（错误）变为 FALSE，ErrorID（错误码）的值变为 0。

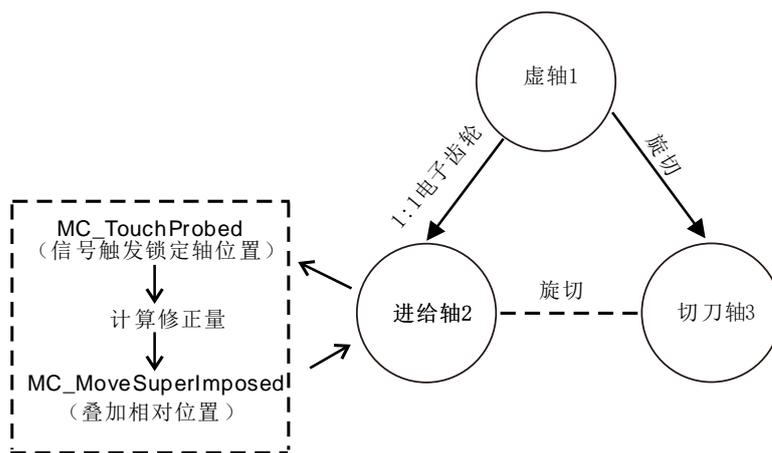
#### ● 单独执行

Execute（启动）由 FALSE 变为 TRUE 时，Busy（执行中）同时变为 TRUE，切刀轴运转至建立旋切关系前的位置停止，Done（完成）变为 TRUE，旋切关系解除。当 Execute（启动）变为 FALSE 时，Done（完成）同时变为 FALSE。

### ■ 旋切示例程序

#### ● 实现功能

如下图所示该程序通过虚拟伺服轴 1 和进给轴 2 建立 1:1 的电子齿轮关系，虚拟伺服轴 1 和切刀轴 3 建立旋切关系，进给轴 2 和切刀轴 3 建立间接的旋切关系，原本应该为进给轴 2 和切刀轴 3 建立旋切关系。这样做的好处在于进给轴 2 需要实时修正时，方便进轴 2 的位置修正。该示例程序中没有涉及位置修正，需要位置修正时，可以参考下图虚线方框处的操作方法。当切刀轴需要和进给轴解除旋切关系并停止时，执行 MC\_RotaryKnife\_Out 指令。



● 轴参数设置

进给轴轴 2 参数设置如下图所示，“工作每转的工作行程”=2\*3.14\*20（进给轴半径）=125.6。



虚拟伺服轴和进给轴为 1:1 的电子齿轮关系，虚拟伺服轴轴 1 参数和进给轴轴参数相同，如下图所示。



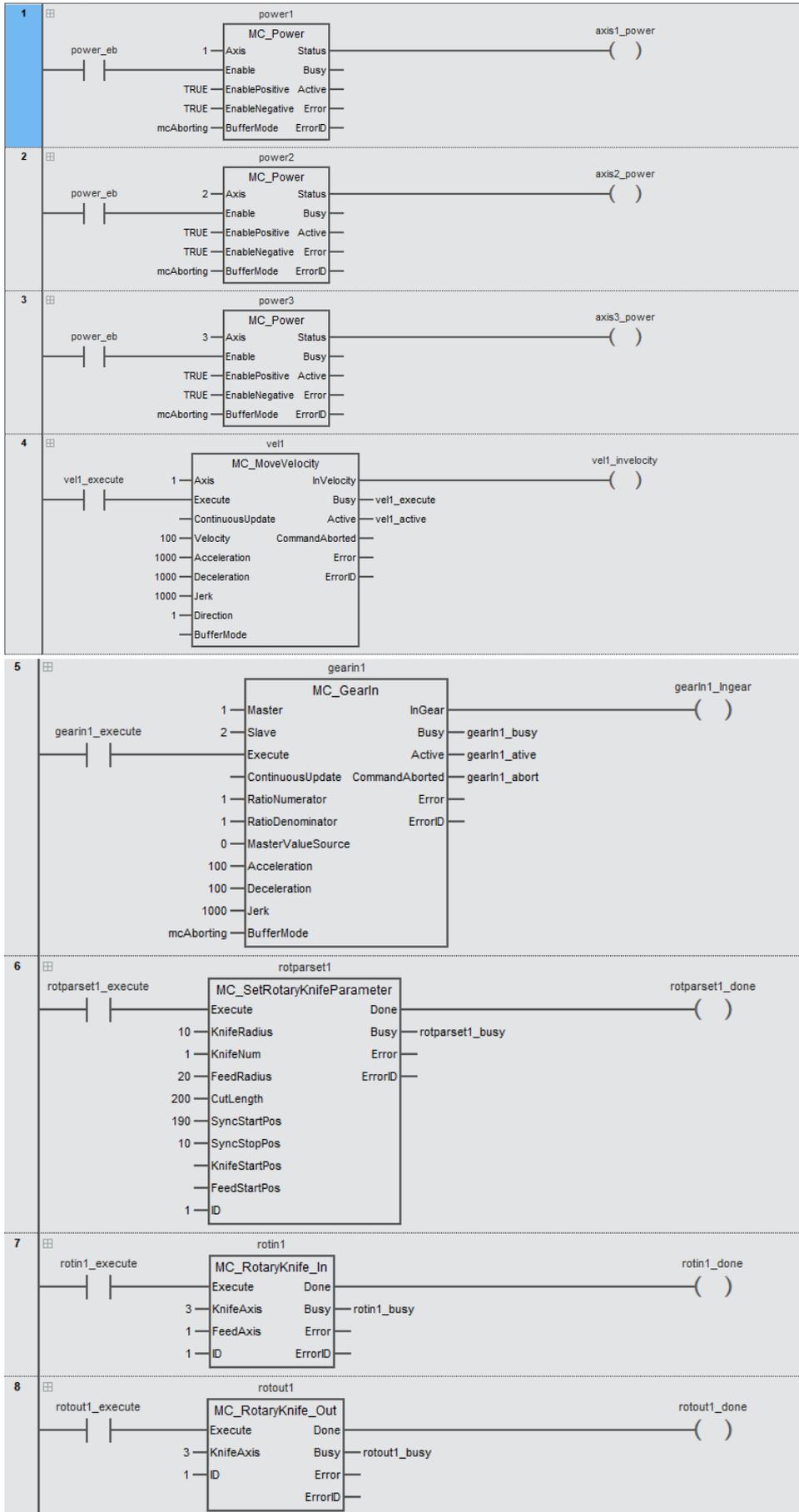
切刀轴轴 3 参数设置如下图所示，“工作每转的工作行程” $=2*3.14*10$ （切刀轴半径） $=62.8$ 。



### ● 变量表

类别	名称	分配到	数据类型	初始值	注释
VAR	power_eb		BOOL		
VAR	power1		MC_Power		
VAR	axis1_power		BOOL		
VAR	power2		MC_Power		
VAR	axis2_power		BOOL		
VAR	power3		MC_Power		
VAR	axis3_power		BOOL		
VAR	vel1		MC_MoveVelocity		
VAR	vel1_execute		vel1_execute		
VAR	vel1_invelocity		BOOL		
VAR	vel1_busy		BOOL		
VAR	vel1_active		BOOL		
VAR	gearin1		MC_GearIn		
VAR	gearin1_execute		BOOL		
VAR	gearIn1_Ingear		BOOL		
VAR	gearIn1_busy		BOOL		
VAR	gearIn1_ative		BOOL		
VAR	gearIn1_abort		BOOL		
VAR	rotparset1		MC_SetRotaryKnifeParameter		
VAR	rotparset1_execute		BOOL		
VAR	rotparset1_done		BOOL		
VAR	rotparset1_busy		BOOL		
VAR	rotin1		MC_RotaryKnife_In		
VAR	rotin1_execute		BOOL		
VAR	rotin1_done		BOOL		
VAR	rotin1_busy		BOOL		
VAR	rotout1		MC_RotaryKnife_Out		
VAR	rotout1_execute		BOOL		
VAR	rotout1_done		BOOL		
VAR	rotout1_busy		BOOL		

● 梯形图(LD)



- 结构化文本(ST)

```
power1(  
  Axis:=1 ,  
  Enable:=power_eb ,  
  EnablePositive:=TRUE ,  
  EnableNegative:=TRUE ,  
  BufferMode:=mcAborting ,  
  Status=>axis1_power  
);  
  
power2(  
  Axis:=2 ,  
  Enable:=power_eb ,  
  EnablePositive:=TRUE ,  
  EnableNegative:=TRUE ,  
  BufferMode:=mcAborting ,  
  Status=>axis2_power  
);  
  
power3(  
  Axis:=3 ,  
  Enable:=power_eb ,  
  EnablePositive:=TRUE ,  
  EnableNegative:=TRUE ,  
  BufferMode:=mcAborting ,  
  Status=>axis3_power  
);  
  
vel1(  
  Axis:=1 ,  
  Execute:= vel1_execute ,  
  Velocity:=30 ,  
  Acceleration:=100 ,  
  Deceleration:=100 ,  
  Jerk:=1000 ,  
  Direction:=1 ,  
  BufferMode:=mcAborting ,  
  InVelocity=> vel1_invelocity ,  
  Busy=> vel1_busy ,  
  Active=> vel1_active ,  
  CommandAborted=> vel1_abort  
);  
  
gearin1(  
  Master:=1 ,  
  Slave:=2 ,  
  Execute:=gearin1_execute ,  
  RatioNumerator:= 1 ,  
  RatioDenominator:=1 ,  
  MasterValueSource:=0 ,  
  Acceleration:=100 ,  
  Deceleration:=100 ,  
  Jerk:=1000 ,
```

```

    BufferMode:=mcAborting ,
    InGear=>gearIn1_Ingear ,
    Busy=>gearIn1_busy ,
    Active=>gearIn1_busy ,
    CommandAborted=>gearIn1_abort
);

rotparset1(
    Execute:=rotparset1_execute ,
    KnifeRadius:=10 ,
    KnifeNum:=1 ,
    FeedRadius:=20 ,
    CutLength:=200 ,
    SyncStartPos:=190 ,
    SyncStopPos:=10 ,
    ID:=1 ,
    Done=>rotparset1_done ,
    Busy=>rotparset1_busy
);

rotin1(
    Execute:=rotin1_execute ,
    KnifeAxis:=3 ,
    FeedAxis:=1 ,
    ID:=1 ,
    Done=>rotin1_done ,
    Busy=>rotin1_busy
);

rotout1(
    Execute:=rotout1_execute ,
    KnifeAxis:=3 ,
    ID:=1 ,
    Done=> rotout1_done ,
    Busy=> rotout1_busy
);

```

#### ● 程序说明

- 轴使能后，gearin1\_execute变为TRUE时，虚拟伺服轴1和进给轴轴2建立1:1的电子齿轮关系。
- rotparset1\_execute变为TRUE时，设置旋切相关的参数，参数设置完成后，rotin1\_execute变为TRUE时，虚拟伺服轴1和切刀轴3建立旋切关系。
- vel1\_execute变为TRUE时，虚拟伺服轴1执行速度指令，进给轴轴2和切刀轴轴3跟随虚拟伺服轴1运转。为进给轴轴2和切刀轴轴3间接建立旋切关系，这样做的目的是为了切长准确，方便对进给轴进行位置修正。
- rotout1\_execute变为TRUE时，切刀轴运转至建立旋切关系前的位置停止，rotout1\_done变为TRUE，旋切关系解除，切刀轴不再跟随进给轴运转。



## 第七章 轴组指令



## 7.1 MC\_AxesGroupAddAxis (指定轴在轴组中的编号)

指定轴在指定轴组中的编号，为执行轴组运动指令做准备。所属库：MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_AxesGroupAddAxis	指定轴在轴组中编号	FB		<pre>MC_AxesGroupAddAxis_Instance (   AxesGroup:= 参数 ,   Axis := 参数 ,   Execute := 参数 ,   IdentNum:= 参数 ,   Done =&gt; 参数 ,   Busy =&gt; 参数 ,   Error =&gt; 参数 ,   ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
AxesGroup	轴组编号	USINT	由机种决定	不可缺省	轴组编号
Axis	轴号	USINT	由机种决定	不可缺省	指定控制轴的轴号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
IdentNum	轴组中的逻辑轴编号	USINT	1~8	不可缺省	轴组中的逻辑轴编号

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成后变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	指令执行完成后	<ul style="list-style-type: none"> <li>◆ Done为TRUE，Execute由TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时，Done变为TRUE一个周期后，变为FALSE</li> </ul>
Busy	Execute 由FALSE变为TRUE时	<ul style="list-style-type: none"> <li>◆ Done变为TRUE时</li> <li>◆ Error变为TRUE时</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Execute由TRUE变为FALSE时</li> </ul>

### ■ 功能说明

#### ● 基本功能说明

- 该指令用于将指定轴分配到指定的轴组中，并和轴组中逻辑轴编号建立对应关系，为轴组运动指令控制指定轴做准备。指定轴由输入变量Axis指定，指定的轴组由输入变量AxesGroup指定，轴组中的逻辑轴编号由输入变量IdentNum指定。该指令执行后，指定轴和轴组中逻辑轴建立对应关系。
- 轴组中有固定的逻辑轴编号，编号范围为1~8，轴组运动指令执行时，会为每个逻辑轴指定对应的位置。通过该指令，编号为1的可以指定为软件中配置的轴1，也可以指定为轴2。指定为轴1时，逻辑轴1控制轴1运

动；指定为轴2时，逻辑轴1控制轴2运动。

- 下表所示为执行8个该指令，设定逻辑轴编号和指定轴的对应关系。轴组运动指令控制时，都是控制逻辑轴，逻辑轴再根据逻辑轴编号和指定轴的对应关系，控制指定轴。

情形1:

<b>AxesGroup的值</b>	<b>1</b>							
IdentNum的值	1	2	3	4	5	6	7	8
Axis的值	1	2	3	4	5	6	7	8
<b>AxesGroup的值</b>	<b>1</b>							
IdentNum的值	1	2	3	4	5	6	7	8
Axis的值	2	1	4	3	7	6	8	5

- **指令完成时机**

指定轴和轴组中逻辑轴编号建立对应关系后，该指令完成，Done 由 FALSE 变为 TRUE。

- **重启该指令**

该指令在轴组没有运行时可以重新执行，Execute 再次由 FALSE 变为 TRUE 时，重新执行。轴组运行后，执行该指令会报错。轴组运行后，如果要重启该指令，需要先通过 MC\_AxesGroupEnable 指令解除轴组运行，再执行该指令。

- **其它指令执行时启动该指令**

轴组没有运行时可以执行该指令，通过执行 MC\_AxesGroupEnable 指令解决轴组运行状态。其它运动指令执行时，执行该指令对其它指令执行无影响。

- **该指令执行时启动其它指令**

该指令执行时，执行 MC\_AxesGroupDeleteAxis 或者 MC\_AxesGroupDeleteAllAxis 指令解除指定轴和轴组中逻辑轴编号对应关系。执行其它指令对该指令执行无影响。

- **异常排除**

该指令执行时，如果指令输入变量值不在允许的范围时，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。

## 7.2 MC\_AxesGroupDeleteAxis (删除逻辑轴在轴组中编号)

删除指定逻辑轴在指定轴组中的编号。所属库：MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_AxesGroupDeleteAxis	删除逻辑轴在轴组中编号	FB		<pre>MC_AxesGroupDeleteAxis_Instance (   AxesGroup:= 参数 ,   Execute := 参数 ,   IdentNum:= 参数 ,   Done =&gt; 参数 ,   Busy =&gt; 参数 ,   Error =&gt; 参数 ,   ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
AxesGroup	轴组编号	USINT	由机种决定	不可缺省	轴组编号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
IdentNum	轴组中的逻辑轴编号	USINT	1~8	不可缺省	轴组中的逻辑轴编号

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成后变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	指令执行完成后	<ul style="list-style-type: none"> <li>◆ Done为TRUE，Execute由TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时，Done变为TRUE一个周期后，变为FALSE</li> </ul>
Busy	Execute 由FALSE变为TRUE时	<ul style="list-style-type: none"> <li>◆ Done变为TRUE时</li> <li>◆ Error变为TRUE时</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Execute由TRUE变为FALSE时</li> </ul>

### ■ 功能说明

#### ● 基本功能说明

- 该指令用于将指定逻辑轴从指定的轴组中删除。指定的轴组由输入变量AxesGroup指定，轴组中的逻辑轴编号由输入变量IdentNum指定。该指令执行后，指定的逻辑轴不能再通过轴组运动指令控制。
- 指定轴和轴组中逻辑轴编号通过MC\_AxesGroupAddAxis指令建立对应关系后，该关系会一直存在，即使指定轴执行其它单轴指令后，该关系仍存在。可以通过该指令取消建立的关系。

#### ● 指令完成时机

指定逻辑轴从指定的轴组中删除后，该指令完成，Done 由 FALSE 变为 TRUE。

- **重启该指令**

该指令在轴组没有运行时可以重新执行，Execute 再次由 FALSE 变为 TRUE 时，重新执行。轴组运行后，执行该指令会报错。轴组运行后，如果要重启该指令，需要先通过 MC\_AxesGroupEnable 指令解除轴组运行，再执行该指令。

- **其它指令执行时启动该指令**

轴组没有运行时可以执行该指令，通过执行 MC\_AxesGroupEnable 指令解决轴组运行状态。其它运动指令执行时，执行该指令对其它指令执行无影响。

- **该指令执行时启动其它指令**

该指令执行时，执行 MC\_AxesGroupAddAxis 指令，逻辑轴编号在轴组中重新建立，执行其它指令对该指令执行无影响。

- **异常排除**

该指令执行时，如果指令输入变量值不在允许的范围时，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。

## 7.3 MC\_AxesGroupDeleteAllAxis（删除所有逻辑轴在轴组中编号）

删除所有逻辑轴在轴组中编号。所属库：MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_AxesGroupDeleteAllAxis	删除所有逻辑轴在轴组中编号	FB		<pre>MC_AxesGroupDeleteAllAxis_Instance (   AxesGroup:= 参数 ,   Execute := 参数 ,   Done =&gt; 参数 ,   Busy =&gt; 参数 ,   Error =&gt; 参数 ,   ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
AxesGroup	轴组编号	USINT	由机种决定	不可缺省	轴组编号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成后变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考 <a href="#">“指令错误代码描述”</a>

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	指令执行完成后	<ul style="list-style-type: none"> <li>◆ Done为TRUE，Execute由TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时，Done变为TRUE一个周期后，变为FALSE</li> </ul>
Busy	Execute 由FALSE变为TRUE时	<ul style="list-style-type: none"> <li>◆ Done变为TRUE时</li> <li>◆ Error变为TRUE时</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Execute由TRUE变为FALSE时</li> </ul>

### ■ 功能说明

#### ● 基本功能说明

- 该指令用于将所有逻辑轴从指定的轴组中删除。指定的轴组由输入变量AxesGroup指定。该指令执行后，所有逻辑轴不能再通过轴组运动指令控制。
- 指定轴和轴组中逻辑轴编号通过MC\_AxesGroupAddAxis指令建立对应关系后，该关系会一直存在，即使指定轴执行其它单轴指令后，该关系仍存在。可以通过该指令取消建立的关系。

#### ● 指令完成时机

所有轴从指定的轴组中删除后，该指令完成，Done 由 FALSE 变为 TRUE。

#### ● 重启该指令

该指令在轴组没有运行时可以重新执行，Execute 再次由 FALSE 变为 TRUE 时，重新执行。轴组运行后，执行该指令会报错。轴组运行后，如果要重启该指令，需要先通过 MC\_AxesGroupEnable 指令解除轴组运行，再执行该指令。

- **其它指令执行时启动该指令**

轴组没有运行时可以执行该指令，通过执行 MC\_AxesGroupEnable 指令解除轴组运行状态。其它运动指令执行时，执行该指令对其它指令执行无影响。

- **该指令执行时启动其它指令**

该指令执行时，执行 MC\_AxesGroupAddAxis 指令，逻辑轴编号在轴组中重新建立，执行其它指令对该指令执行无影响。

- **异常排除**

该指令执行时，如果指令输入变量值不在允许的范围时，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。

## 7.4 MC\_AxesGroupEnable (轴组运行和解除运行)

控制轴组进入运行状态或者解除轴组的运行状态。所属库: MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_AxesGroupEnable	轴组运行和解除运行	FB		<pre>MC_AxesGroupAddAxis_Instance (   AxesGroup:= 参数 ,   Enable := 参数 ,   DirectVelocity := 参数 ,   DirectAcceleration:= 参数 ,   DirectDeceleration := 参数 ,   DirectJerk := 参数 ,   LinerVelocity := 参数 ,   LinerAcceleration := 参数 ,   LinerDeceleration := 参数 ,   LinerJerk := 参数 ,   Status =&gt; 参数 ,   Busy =&gt; 参数 ,   CommandAborted =&gt; 参数 ,   Error =&gt; 参数 ,   ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
AxesGroup	轴组编号	USINT	由机种决定	不可缺省	轴组编号
Enable	有效	BOOL	TRUE或FALSE	FALSE	设为TRUE, 控制轴组进入运行状态 设为FALSE, 解除轴组的运行状态
MoveDirectVelocity	快速定位速度	ARRAY [1..8] OF LREAL	正数	不可缺省	快速定位速度
MoveDirectAcceleration	快速定位加速度	ARRAY [1..8] OF LREAL	正数	不可缺省	快速定位加速度
MoveDirectDeceleration	快速定位减速度	ARRAY [1..8] OF LREAL	正数	不可缺省	快速定位减速度
MoveDirectJerk	快速定位跃度	ARRAY [1..8] OF LREAL	正数	不可缺省	快速定位跃度
LinerVelocity	直线插补速度	LREAL	正数	不可缺省	保留
LinerAcceleration	直线插补加速度	LREAL	正数	不可缺省	保留
LinerDeceleration	直线插补减速度	LREAL	正数	不可缺省	保留
LinerJerk	直线插补跃度	LREAL	正数	不可缺省	直线插补跃度

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Status	轴组运行状态	BOOL	TRUE或FALSE	该输出变量为TRUE时, 表示轴组状态进入运行状态 该输出变量为FALSE时, 表示轴组状态处于解除运行状态
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时, 输出错误代码 值的含义请参考“ <a href="#">指令错误代码描述</a> ”

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	指令执行完成后	◆ Done为TRUE, Enable由TRUE变为FALSE时
Busy	Enable 由FALSE变为TRUE时	◆ Done变为TRUE时 ◆ Error变为TRUE时 ◆ CommandAborted 变为TRUE时
CommandAborted	指令的执行被其它指令中断时	◆ CommandAborted 为TRUE, Enable从TRUE变为FALSE时
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	◆ Enable由TRUE变为FALSE时

## ■ 功能说明

### ● 基本功能说明

- 该指令用于控制指定的轴组进入运行状态或者解除轴组的运行状态。输入变量Enable位为TRUE时，控制指定的轴组进入运行状态；为FALSE时，解除轴组的运行状态。该指令的输出变量Status为TRUE表示轴组状态进入运行状态，轴状态处于DiscreteMotion（定位动作中）状态；为FALSE时，表示轴组的运行状态解除。
- 轴组中单个轴的状态在standstill（停止状态）时，才可以执通过该指令输入变量Enable由FALSE变为TRUE执行该指令。
- 无论轴组中的轴是否运转，该指令输入变量Enable位由TRUE变为FALSE时都会解除轴组的运行状态。轴组运动指令控制轴运转时，轴速度会在一个周期降为0，轴组内各个轴的状态变为standstill（停止状态）。轴组运动指令控制轴运转时，如果需要解除轴组运行，建议先停止轴组的运动，然后再解除轴运行状态。
- 该指令执行后，轴组运动指令才可以执行。
- 该指令执行后，轴组中轴执行非轴组相关的运动指令时，该指令会被打断，执行运动指令的轴的状态由对应的运动指令决定，其它轴状态变为standstill（停止状态）。

### ● 指令完成时机

该指令的输出变量 Status 为 TRUE 表示轴组状态进入运行状态；为 FALSE 时，表示轴组的运行状态解除。

### ● 重启该指令

输入变量 Enable 位为 TRUE 时，控制指定的轴组进入运行状；为 FALSE 时，解除轴组的运行状态。

### ● 其它指令执行时启动该指令

参考该指令基本功能说明部分。

### ● 该指令执行时启动其它指令

参考该指令基本功能说明部分。

### ● 异常排除

该指令执行时，如果指令输入变量值不在允许的范围时，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。

## 7.5 MC\_AxesGroupPause (轴组运动暂停)

对正在执行的轴组运动进行减速停止，暂停轴组运动。所属库：MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_AxesGroupPause	轴组运动暂停	FB		<pre>MC_AxesGroupPause_Instance (   AxesGroup:= 参数 ,   Execute := 参数 ,   Deceleration:= 参数,   Jerk:= 参数,   Done =&gt; 参数 ,   Busy =&gt; 参数 ,   CommandAborted =&gt; 参数 ,   Error =&gt; 参数 ,   ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
AxesGroup	轴组编号	USINT	由机种决定	不可缺省	轴组编号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
Deceleration	减速度	LREAL	正数	不可缺省	保留
Jerk	跃度	LREAL	正数	不可缺省	保留

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成后变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	指令执行完成后	◆ Done为TRUE，Enable由TRUE变为FALSE时
Busy	Enable 由FALSE变为TRUE时	◆ Done变为TRUE时 ◆ Error变为TRUE时 ◆ CommandAborted 变为TRUE时
CommandAborted	指令的执行被其它指令中断时	◆ CommandAborted 为TRUE，Enable从TRUE变为FALSE时
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	◆ Enable由TRUE变为FALSE时

## ■ 功能说明

### ● 基本功能说明

- 该指令用于对正在执行的轴组运动进行减速停止，暂停轴组运动。该指令执行后，使用正在执行轴组运动指令的输入变量的值进行减速停止。该指令执行完成后，轴状态为Discrete Motion（定位动作中）。该指令输入变量Deceleration、Jerk为保留参数，用户输入数值无效。
- 轴组运动指令MC\_MoveDirectAbsolute、MC\_MoveDirectRelative、MC\_MoveLinearAbsolute、MC\_MoveLinearRelative、MC\_MoveCircularAbsolute与MC\_MoveCircularRelative执行时，MC\_AxesGroupPause指令才可以执行，其它情况下执行该指令会报错。
- 该指令执行后，如果要继续轴组运动指令的执行，需要通过执行指令MC\_AxesGroupResume解除暂停。
- 如果想终止轴组指令的执行或者轴组中的轴需要执行其它运动指令，需要先执行MC\_AxesExit指令，再将MC\_AxesGroupEnable 指令Enable设为FALSE。建议执行顺序：先执行MC\_AxesGroupPause（轴组运动暂停）指令，再执行MC\_AxesGroupExit指令，再将MC\_AxesGroupEnable 指令Enable设为FALSE。

### ● 指令完成时机

该指令执行后，下达暂停命令后，该指令完成，Done 由 FALSE 变为 TRUE。该指令的 Done 为 TRUE，不代表轴已经停止。轴是否停止可以通过轴结构体中的轴命令速度或者轴反馈速度判断。

### ● 重启该指令

参考该指令基本功能说明部分。

### ● 其它指令执行时启动该指令

参考该指令基本功能说明部分。

### ● 该指令执行时启动其它指令

参考该指令基本功能说明部分。

### ● 异常排除

该指令执行时，如果指令输入变量值不在允许的范围时，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。

## 7.6 MC\_AxesGroupResume (轴组运动暂停解除)

轴组运动暂停解除，从暂停处继续轴组运动中的剩余部分。所属库：MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_AxesGroupResume	轴组运动暂停解除	FB		<pre>MC_AxesGroupResume_Instance (   AxesGroup:= 参数 ,   Execute := 参数 ,   Done =&gt; 参数 ,   Busy =&gt; 参数 ,   CommandAborted =&gt; 参数 ,   Error =&gt; 参数 ,   ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
AxesGroup	轴组编号	USINT	由机种决定	不可缺省	轴组编号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成后变为TRUE。
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE。
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE。
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE。
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码。 值的含义请参考“ <a href="#">指令错误代码描述</a> ”。

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	指令执行完成后。	◆ Done为TRUE, Enable由TRUE变为FALSE时。
Busy	Enable 由FALSE变为TRUE时。	◆ Done变为TRUE时。 ◆ Error变为TRUE时。 ◆ CommandAborted 变为TRUE时。
CommandAborted	指令的执行被其它指令中断时。	◆ CommandAborted 为TRUE, Enable从TRUE变为FALSE时。
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常。	◆ Enable由TRUE变为FALSE时

## ■ 功能说明

### ● 基本功能说明

- 该指令用于解除轴组运动的暂停，从暂停处继续轴组运动中的剩余部分。该指令执行后，轴组运动使用正在执行轴组运动指令的输入变量的值进行运动。
- MC\_AxesGroupPause指令执行后，该指令才可以执行，否则执行该指令会报错。

### ● 指令完成时机

该指令执行后，下达暂停命令后，该指令完成，Done 由 FALSE 变为 TRUE。

### ● 重启该指令

参考该指令基本功能说明部分。

### ● 其它指令执行时启动该指令

参考该指令基本功能说明部分。

### ● 该指令执行时启动其它指令

该指令执行时，非轴组相关的运动指令可以打断该指令。

### ● 异常排除

该指令执行时，如果指令输入变量值不在允许的范围时，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。

## 7.7 MC\_AxesGroupExit (轴组运动急停)

对正在执行的轴组运动进行急停，一个周期停止。所属库：MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_AxesGroupExit	轴组运动急停	FB		<pre>MC_AxesGroupExit_Instance (   AxesGroup:= 参数 ,   Execute := 参数 ,   Deceleration:= 参数 ,   Jerk:= 参数 ,   BufferMode:= 参数 ,   Done =&gt; 参数 ,   Busy =&gt; 参数 ,   Active=&gt; 参数 ,   CommandAborted =&gt; 参数 ,   Error =&gt; 参数 ,   ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
AxesGroup	轴组编号	USINT	由机种决定	不可缺省	轴组编号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
Deceleration	减速度	LREAL	正数	不可缺省	保留
Jerk	跃度	LREAL	正数	不可缺省	保留
BufferMode	缓存模式	MC_Buffer_Mode	保留	—	保留

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成后变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Active	正在控制轴	BOOL	TRUE或FALSE	指令正在控制轴
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	指令执行完成后	<ul style="list-style-type: none"> <li>◆ Done为TRUE，Execute由TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时，Done变为TRUE一个周期后，变为FALSE</li> </ul>
Busy	Execute 由FALSE变为TRUE时	<ul style="list-style-type: none"> <li>◆ Done变为TRUE时</li> <li>◆ Error变为TRUE时</li> </ul>
Active	指令开始控制轴时	<ul style="list-style-type: none"> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>
CommandAborted	指令的执行被中断时	<ul style="list-style-type: none"> <li>◆ CommandAborted为TRUE，Execute由TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时，该指令被其它指令中断时，CommandAborted变为TRUE一个周期后，变为FALSE</li> </ul>

Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	◆ Execute由TRUE变为FALSE时
-------	--	------------------------

## ■ 功能说明

### ● 基本功能说明

- 该指令用于对正在执行的轴组运动进行急停。该指令执行后，一个周期停止，轴状态处于DiscreteMotion（定位动作中）状态。轴组中的轴运转时，不建议使用该指令。建议先执行MC\_AxesGroupPause（有减速过程）指令，再执行MC\_AxesGroupExit指令，再将MC\_AxesGroupEnable 指令Enable设为FALSE。
- 轴组状态在运行状态时（通过MC\_AxesGroupEnable控制轴组状态进入运行状态），该指令才可以执行，其它情况下执行该指令会报错。
- 该指令执行后，如果想解轴组的运行状态，可以通过将MC\_AxesGroupEnable 指令Enable设为FALSE解除轴组的运行状态。

### ● 指令完成时机

该指令执行后，一个周期停止，该指令完成，Done 由 FALSE 变为 TRUE。

### ● 重启该指令

参考该指令基本功能说明部分。

### ● 其它指令执行时启动该指令

参考该指令基本功能说明部分。

### ● 该指令执行时启动其它指令

参考该指令基本功能说明部分。

### ● 异常排除

该指令执行时，如果指令输入变量值不在允许的范围时，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。

## 7.8 MC\_AxesGroupSetOverride (轴组速度比例设置)

该指令用于更改轴组运动指令的目标速度比例。所属库：MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_AxesGroupSetOverride	轴组速度比例设置	FB		<pre>MC_AxesGroupSetOverride_Instance _AxesGroupSetOverride (   AxesGroup := 参数 ,   Enable:= 参数 ,   VelFactor:= 参数 ,   AccFactor:= 参数 ,   JerkFactor:= 参数 ,   Enabled =&gt; 参数 ,   Busy =&gt; 参数 ,   Error =&gt; 参数 ,   ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
AxesGroup	轴组编号	USINT	由机种决定	不可缺省	轴组编号
Enable	有效	BOOL	TRUE或FALSE	FALSE	设为TRUE时，目标速度比例值生效；设为FALSE时，目标速度比例值变为“100%”
VelFactor	目标速度比例值	LREAL	0~500	100	目标速度比例值（单位：%）
AccFactor	加速度比例值	LREAL	正数	不可缺省	保留
JerkFactor	跃度比例值	LREAL	正数	不可缺省	保留

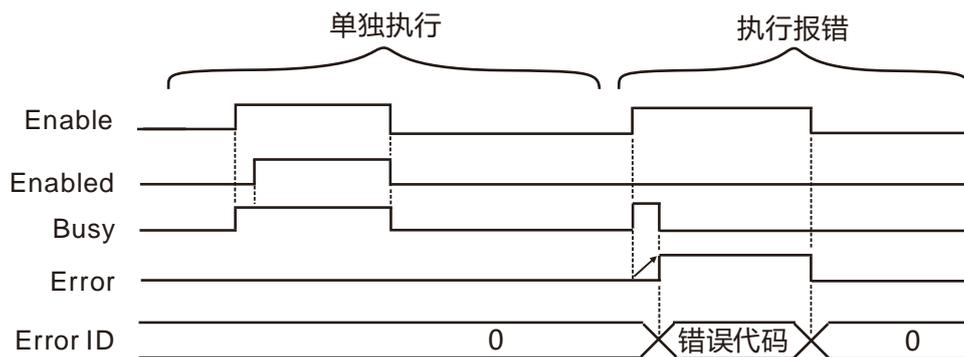
### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Enabled	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Enabled	当开始执行指令时	<ul style="list-style-type: none"> <li>当Enable变为FALSE时</li> <li>当Error变为TRUE时</li> </ul>
Busy	当 Enable 为TRUE时	<ul style="list-style-type: none"> <li>当Enable变为FALSE时</li> <li>当Error为TRUE时</li> </ul>
Error	当指令输入变量不合法或指令执行过程中出错时	<ul style="list-style-type: none"> <li>当Enable由TRUE变为FALSE时</li> </ul>

## ■ 输出变量变化时序说明



### ● 单独执行

Enable (有效) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE, 下一个周期 Enabled (有效) 变为 TRUE, 该指令 VelFactor 参数值开始生效。当 Enable (有效) 变为 FALSE 时, Busy (执行中) 和 Enabled 同时变为 FALSE。

### ● 执行报错

该指令输入变量值不在允许范围内时, 该指令 Enable (有效) 由 FALSE 变为 TRUE 时, 同时 Busy (执行中) 变为 TRUE, 下一个周期 Error (错误) 变为 TRUE, 同时 Busy (执行中) 变为 FALSE, ErrorID (错误码) 输出对应的错误码, 可通过 ErrorID (错误码) 的值, 查找发生问题原因。该指令 Enable (有效) 由 TRUE 变为 FALSE 时, 同时 Error (错误) 变为 FALSE, ErrorID (错误码) 的值变为 0。

## ■ 功能说明

### ● 基本功能说明

- 该指令用于更改轴组运动指令的目标速度比例。该指令的Enable为TRUE时, 可以通过变更该指令VelFactor (目标速度比例) 的值实时更改轴组运动指令的目标速度。VelFactor的单位为百分比, 如“50”表示“50%”。VelFactor参数值范围为0~500, 超出范围时执行该指令, 该指令会报错。
- 轴组指令执行时, 可以通过该指令更改目标速度比例的指令如下表所示。

MC_MoveLinearRelative (相对值直线插补)	MC_MoveDirectRelative (相对值快速定位)
MC_MoveLinearAbsolute (绝对值直线插补)	MC_MoveDirectAbsolute (绝对值快速定位)
MC_MoveCircularRelative (相对值圆弧插补)	
MC_MoveCircularAbsolute (绝对值圆弧插补)	

- 该指令执行后, 新的目标速度如下:

$$\text{变更后的目标速度} = \text{当前控制轴的轴组指令的目标速度} \times \text{该指令速度比例值}$$

- 该指令的Enable为TRUE时, 更改该指令输入变量VelFactor的值, 根据轴组运动控制指令的输入变量进行加速或减速到目标速度。如MC\_MoveLinearRelative指令执行时, 更改该指令输入变量VelFactor的值, 则根据MC\_MoveLinearRelative指令输入变量Acceleration、Deceleration的设定值的进行加速或者减速到目标速度。
- 将VelFactor设置为“0”时, 目标速度变为“0”。希望保持运动指令执行, 但又想暂停运动指令控制轴时, 可以将该指令VelFactor设为“0”实现。此种情况下, 轴状态不会发生变化。
- 该指令的Enable从TRUE变为FALSE时, 轴组速度从当前速度变化到轴组运动指令的目标速度, 相当于该指令执行时VelFactor的值为100的情形。轴组的速度变化根据当前控制轴的轴组指令的输入变量进行加速或减速到目标速度。如MC\_MoveLinearRelative指令执行时, 该指令VelFactor的值为200并且Enable为TRUE, 该指令的Enable从TRUE变为FALSE时, 则根据MC\_MoveLinearRelative指令输入变量Acceleration、Deceleration的设定值的进行加速或者减速到MC\_MoveLinearRelative指令设定的目标速度。

- **重启该指令**

该指令的 Enable 为 TRUE 时持续执行。

- **异常排除**

该指令执行时，如果指令输入变量值不在允许的范围或者不满足指令的执行条件，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。

## 7.9 MC\_MoveLinearRelative (相对值直线插补)

控制多个轴同时启动和停止，进行直线插补动作，指定距离为相对值。所属库：MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_MoveLinearRelative	相对值直线插补	FB		<pre> MC_MoveLinearRelative_Instance (   AxesGroup := 参数,   Execute := 参数,   Distance:= 参数,   Velocity:= 参数,   Acceleration:= 参数,   Deceleration:= 参数,   Jerk:= 参数,   CoordSystem:= 参数,   BufferMode:= 参数,   TransitionMode:= 参数,   TransitionParameter:= 参数,   Done =&gt; 参数,   Busy =&gt; 参数,   Active=&gt; 参数,   CommandAborted =&gt; 参数,   Error =&gt; 参数,   ErrorID=&gt; 参数 ); </pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
AxesGroup	轴组编号	USINT	由机种决定	不可缺省	轴组编号
Execute	启动	BOOL	TRUE或FALSE	FALSE	设为TRUE时，目标速度比例值生效；设为FALSE时，目标速度比例值变为“100%”
Distance	移动距离	LREAL	正数、负数、0	0	指定以当前位置为参考点的移动距离*1 (单位: 行程单位) *2
Velocity	目标速度	LREAL	正数	不可缺省	指定目标速度*1 (单位: 行程单位/秒) *2
Acceleration	加速度	LREAL	正数	不可缺省	指定加速度*1 (单位: 行程单位/秒 <sup>2</sup> ) *2
Deceleration	减速度	LREAL	正数	不可缺省	指定减速度*1 (单位: 行程单位/秒 <sup>2</sup> ) *2
Jerk	跃度	LREAL	正数	不可缺省	指定跃度*1 (单位: 行程单位/秒 <sup>3</sup> ) *2
CoordSystem	坐标系	INT	保留	保留	保留
BufferMode	缓存模式	MC_Buffer_Mode	1: mcBuffered 3: mcBlendingPrevious	不可缺省	设定两个指令之间中继模式*3 1: 等待 3: 以当前指令目标速度中继
TransitionMode	过渡模式	MC_Transition_Mode	0: mcTMNone 2: TMCons-tantVelocity 3: mcTMCorner-Distance	不可缺省	设定当前控制轴的插补指令轨迹和缓存插补指令轨迹的连接方式 0: 位置到达时，速度降为0 2: 以指定的速度过渡 3: 以设定的附加角过渡
TransitionParameter	附加角	LREAL	正数、0	0	TransitionMode选择以附加角过渡时，设定过渡参数

\*1: Velocity、Acceleration、Deceleration、Jerk的关系请参考“[运动控制指令相关参数说明](#)”章节。

\*2: 指令单位的详细介绍请参考“[运动控制指令参数单位](#)”章节。

\*3: BufferMode的详细介绍请参考“[运动控制指令的多重启动时缓存模式说明 \(BufferMode\)](#)”章节。

## ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成时变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Active	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

## ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	定位完成时	<ul style="list-style-type: none"> <li>◆ Done为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE</li> </ul>
Busy	Execute的上升沿	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
Active	指令开始控制轴时	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
CommandAborted	指令的执行被其它指令中断时	<ul style="list-style-type: none"> <li>◆ CommandAborted 为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令被其它指令中断时, CommandAborted变为TRUE, 一个周期后, 变为FALSE</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令执行过程中遇到异常, Error变为TRUE, 一个周期后, 变为FALSE</li> </ul>

## ■ 功能说明

### ● 基本功能说明

该指令用于控制多个轴同时启动和停止，进行直线插补动作，最多可以 8 个轴同时动作。各个轴移动的距离通过输入变量 Distance 单独设定，输入变量 Velocity 为所有轴的合成速度，输入变量 Acceleration、Deceleration 为所有轴的合成加速度和减速度。该指令在 Execute 由 FALSE 变为 TRUE 时，按照输入变量设定的值进行运动。

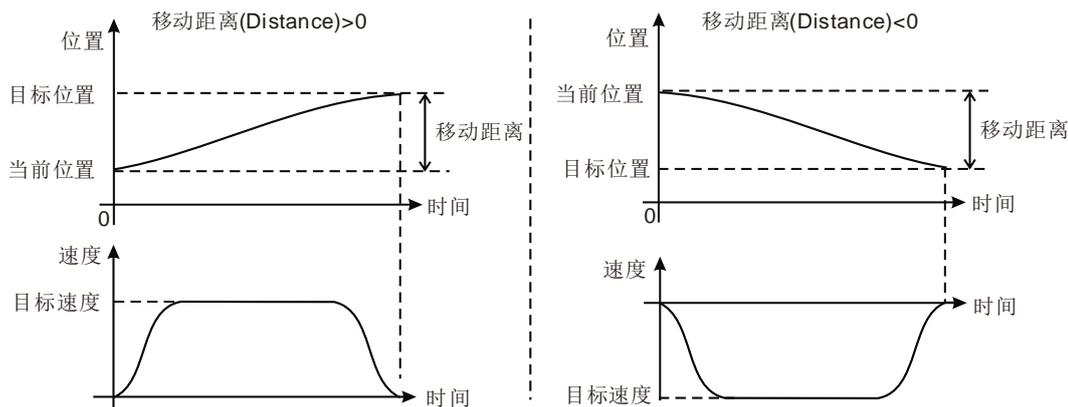
### ● 目标速度

输入变量 Velocity 为所有轴的合成速度，合成速度和各个轴的速度关系：合成速度的平方和= 各轴速度的平方和。该指令输入变量 Acceleration、Deceleration 为所有轴的合成加速度和减速度，合成加速度、减速度和各个轴加速度、减速度关系：合成加（减）速度平方和 = 各轴加（减）速度的平方和。

### ● 移动距离

- 轴的目标位置由轴组中各个轴的当前位置和Distance的值共同决定，即目标位置= 当前位置+Distance。Distance的值可以为正数、0、负数。Velocity、Acceleration、Deceleration的值只能为正数。
- 轴静止时执行该指令，执行该指令，Distance的值大于0和小于0时的位置曲线以及速度曲线如下图所示。Distance的值大于0时，轴正向运转；Distance的值小0时，则轴反向运转。

- Distance的值等于0时, 该指令不控制轴运转, 该指令的Done在执行该指令一个扫描周期后变为TRUE。



- **指令完成时机**

轴组中各个轴的命令位置到达设定的距离时, 该指令完成, Done 由 FALSE 变为 TRUE。

- **重启该指令**

当指令执行完成后, Execute 再次由 FALSE 变为 TRUE 时, 该指令可以重新执行; 当指令正在执行中, Execute 再次由 FALSE 变为 TRUE 时, 对指令的执行不会产生影响, 指令仍按照未执行完成的输入变量执行指令。

- **其它指令执行时启动该指令**

其它轴组运动指令执行时, 启动该指令, 该指令和其它运动指令如何中继由该指令输入变量 BufferMode (缓存模式)、TransitionMode (过渡模式) 的值共同决定。详细说明如下表所示。

BufferMode (缓存模式)	TransitionMode (过渡模式)	含义
mcBuffered (等待)	mcTMNone(位置到达时, 速度降为0)	等待当前轴组指令执行完成时 (目标位置到达时, 速度降为0), 切换为缓存轴组指令控制轴。
mcBlendingPrevious (以当前指令的目标速度中继)	mcTMConstantVelocity (以指定的速度过渡) 该模式一般应用于曲线拟合或者小线段连续插补, 使用时建议将多个指令使用同一个执行位一起触发执行, 过渡速度为第一个执行的轴组指令的目标速度。	当前轴组指令目标位置到达时, 轴组速度为当前轴组指令的目标速度, 然后以当前轴组指令的目标速度为过渡速度进行过渡, 尔后再以缓存轴组指令设定的输入参数执行。
	mcTMCornerDistance (以设定的附加角过渡)	当前轴组指令目标位置完成时刻为当前轴组指令的目标位置减去mcTMCornerDistance设定的附加角位置。当前轴组指令执行完成后, 立即执行缓存轴组指令, 缓存轴组指令先以mcTMCornerDistance设定的附加角运动, 然后再以缓存轴组指令设定的参数运动。

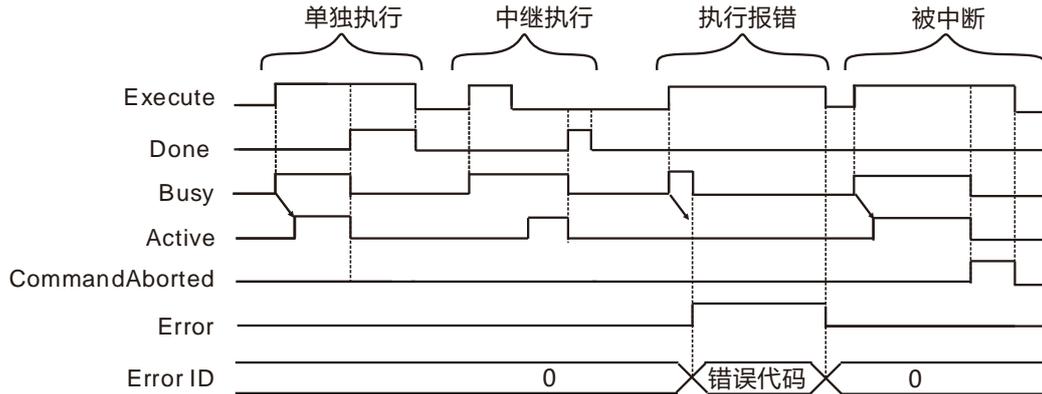
- **该指令执行时启动其它指令**

该指令执行时启动其它轴组运动指令, 其它轴组运动指令和该指令如何中继由其它运动指令的 BufferMode (缓存模式)、TransitionMode (过渡模式) 的值共同决定。其它运动指令和该指令缓存时, 其它指令执行的时机为该指令 Done (完成)变为 TRUE 时。如果其它运动指令无 BufferMode 引脚, 一般为中断该指令。该指令执行时, 执行其它非轴组相关的运动指令, 会打断该指令, 指定轴的加速或者减速方式由执行指令的输入变量决定; 轴组中其它轴进入 standstill (停止) 状态, 轴速度会一个周期降为 0。

- **异常排除**

该指令执行时, 如果指令输入变量值不在允许的范围或者不满足指令的执行条件, 该指令 Error 变为 TRUE, ErrorID 中有对应的错误码, 可以通过该错误码判断错误原因。该指令执行时, 如遇到异常 (如轴报警等), 该指令也会报错, 轴会立即停止。

## ■ 输出变量变化时序说明



### ● 单独执行

Execute (启动) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE, 下一个周期 Active (控制中) 变为 TRUE。当达到 Distance (移动距离)、定位完成时, Done (完成) 变为 TRUE, Busy (执行中) 和 Active (控制中) 同时变为 FALSE。当 Execute (启动) 变为 FALSE 时, Done (完成) 同时变为 FALSE。

### ● 中继执行

其它指令控制轴时, 执行该指令 (BufferMode 的值不为 mcAborting), 当 Execute (启动) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE, Active (控制中) 变为 TRUE 的时机为前一个指令完成时。

### ● 执行报错

该指令输入变量值不在允许范围内时, 该指令 Execute (启动) 由 FALSE 变为 TRUE 时, 同时 Busy (执行中) 变为 TRUE, 下一个周期 Error (错误) 变为 TRUE, 同时 Busy (执行中) 变为 FALSE, ErrorID (错误码) 输出对应的错误码, 可通过 ErrorID (错误码) 的值, 查找发生问题原因。该指令 Execute (启动) 由 TRUE 变为 FALSE 时, 同时 Error (错误) 变为 FALSE, ErrorID (错误码) 的值变为 0。

### ● 被中断

该指令执行后被其它指令中断时, 该指令 CommandAborted (中断) 变为 TRUE, Busy (执行中) 和 Active (控制中) 同时变为 FALSE; 当 Execute (执行中) 变为 FALSE 时, CommandAborted (中断) 同时变为 FALSE。

## 7.10 MC\_MoveLinearAbsolute (绝对值直线插补)

控制多个轴同时启动和停止，进行直线插补动作，指定位置为绝对值。所属库：MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_MoveLinearAbsolute	绝对值直线插补	FB		<pre> MC_MoveLinearAbsolute_Instance (   AxesGroup := 参数 ,   Execute := 参数 ,   Position := 参数 ,   Velocity := 参数 ,   Acceleration := 参数 ,   Deceleration := 参数 ,   Jerk := 参数 ,   CoordSystem := 参数 ,   BufferMode := 参数 ,   TransitionMode := 参数 ,   TransitionParameter := 参数 ,   Done =&gt; 参数 ,   Busy =&gt; 参数 ,   Active =&gt; 参数 ,   CommandAborted =&gt; 参数 ,   Error =&gt; 参数 ,   ErrorID =&gt; 参数 ); </pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
AxesGroup	轴组编号	USINT	由机种决定	不可缺省	轴组编号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
Position	绝对位置	LREAL	正数、负数、0	0	指定以0点位置为参考点的绝对位置 (单位: 行程单位) *2
Velocity	目标速度	LREAL	正数	不可缺省	指定目标速度*1 (单位: 行程单位/秒) *2
Acceleration	加速度	LREAL	正数	不可缺省	指定加速度*1 (单位: 行程单位/秒 <sup>2</sup> ) *2
Deceleration	减速度	LREAL	正数	不可缺省	指定减速度*1 (单位: 行程单位/秒 <sup>2</sup> ) *2
Jerk	跃度	LREAL	正数	不可缺省	指定跃度*1 (单位: 行程单位/秒 <sup>3</sup> ) *2
CoordSystem	坐标系	INT	保留	保留	保留
BufferMode	缓存模式	MC_Buffer_Mode	1: mcBuffered 3: mcBlendingPrevious	不可缺省	设定两个指令之间中继模式*3 1: 等待 3: 以当前指令目标速度中继
TransitionMode	过渡模式	MC_Transition_Mode	0: mcTMNone 2: mcTMConstantVelocity 3: mcTMCornerDistance	不可缺省	设定当前控制轴的插补指令轨迹和缓存插补指令轨迹的连接方式 0: 位置到达时, 速度降为0 2: 以指定的速度过渡 3: 以设定的附加角过渡
TransitionParameter	附加角	LREAL	正数、0	0	TransitionMode选择以附加角过渡时, 设定过渡参数

\*1: Velocity、Acceleration、Deceleration、Jerk的关系请参考“[运动控制指令相关参数说明](#)”章节。

\*2: 指令单位的详细介绍请参考“[运动控制指令参数单位](#)”章节。

\*3: BufferMode的详细介绍请参考“[运动控制指令的多重启动时缓存模式说明 \(BufferMode\)](#)”章节。

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成后变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Active	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

## ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	定位完成时	<ul style="list-style-type: none"> <li>◆ Done为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE</li> </ul>
Busy	Execute的上升沿	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
Active	指令开始控制轴时	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
CommandAborted	指令的执行被其它指令中断时	<ul style="list-style-type: none"> <li>◆ CommandAborted 为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令被其它指令中断时, CommandAborted变为TRUE, 一个周期后, 变为FALSE</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令执行过程中遇到异常时, Error变为TRUE, 一个周期后, 变为FALSE</li> </ul>

## ■ 功能说明

### ● 基本功能说明

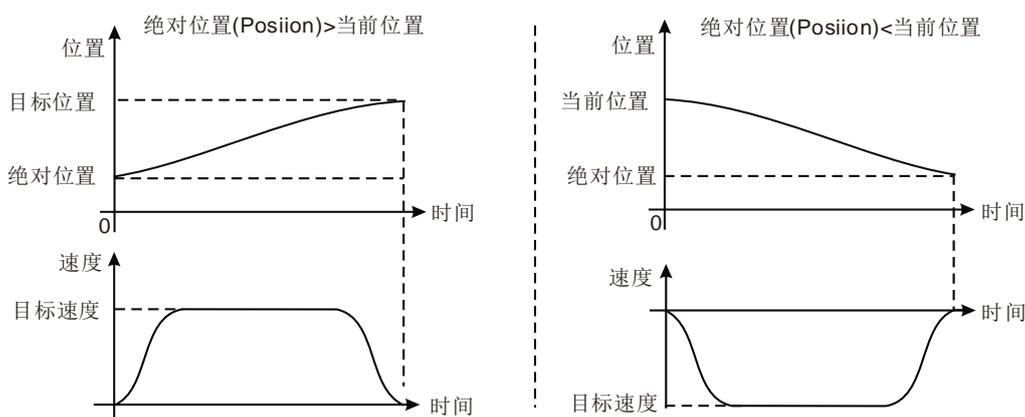
该指令用于控制多个轴同时启动和停止，进行直线插补动作，最多可以 8 个轴同时动作。各个轴移动的距离通过输入变量 Distance 单独设定，输入变量 Velocity 为所有轴的合成速度，输入变量 Acceleration、Deceleration 为所有轴的合成加速度和减速度。该指令在 Execute 由 FALSE 变为 TRUE 时，按照输入变量设定的值进行运动。

### ● 目标速度

输入变量 Velocity 为所有轴的合成速度，合成速度和各个轴的速度关系：合成速度的平方和= 各轴速度的平方和。该指令输入变量 Acceleration、Deceleration 为所有轴的合成加速度和减速度，合成加速度、减速度和各个轴加速度、减速度关系：合成加（减）速度 = 各轴加（减）速度的平方和。

### ● 绝对位置

- 输入变量Position的值表示以0点位置为参考点的绝对位置。该值可以为正数、0、负数。Velocity、Acceleration、Deceleration、Jerk的值只能为正数。
- 轴静止时执行该指令，执行该指令，Position的值大于当前位置和小于当前位置时的位置曲线以及速度曲线如下图所示。Position的值大于当前位置时，轴正向运转；Position的值小于当前位置时，则轴反向运转。
- Position的值等于当前位置时，该指令不控制轴运转，该指令的Done在执行该指令一个扫描周期后变为TRUE。



- **指令完成时机**

轴组中各个轴的命令位置到达设定的位置时，该指令完成，Done 由 FALSE 变为 TRUE。

- **重启该指令**

当指令执行完成后，Execute 再次由 FALSE 变为 TRUE 时，该指令可以重新执行；当指令正在执行中，Execute 再次由 FALSE 变为 TRUE 时，对指令的执行不会产生影响，指令仍按照未执行完成的输入变量执行指令。

- **其它指令执行时启动该指令**

其它轴组运动指令执行时，启动该指令，该指令和其它运动指令如何中继由该指令输入变量 BufferMode（缓存模式）、TransitionMode（过渡模式）的值共同决定。详细说明如下表所示。

BufferMode（缓存模式）	TransitionMode（过渡模式）	含义
mcBuffered (等待)	mcTMNone(位置到达时，速度降为0)	等待当前轴组指令执行完成时（目标位置到达时，速度降为0），切换为缓存轴组指令控制轴。
mcBlendingPrevious (以当前指令的目标速度中继)	mcTMConstantVelocity (以指定的速度过渡) 该模式一般应用于曲线拟合或者小线段连续插补，使用时建议将多个指令使用同一个执行位一起触发执行，过渡速度为第一个执行的轴组指令的目标速度。	当前轴组指令目标位置到达时，轴组速度为当前轴组指令的目标速度，然后以当前轴组指令的目标速度为过渡速度进行过渡，尔后再以缓存轴组指令设定的输入参数执行。
	mcTMCornerDistance (以设定的附加角过渡)	当前轴组指令目标位置完成时刻为当前轴组指令的目标位置减去mcTMCornerDistance设定的附加角位置。当前轴组指令执行完成后，立即执行缓存轴组指令，缓存轴组指令先以mcTMCornerDistance设定的附加角运动，然后再以缓存轴组指令设定的参数运动。

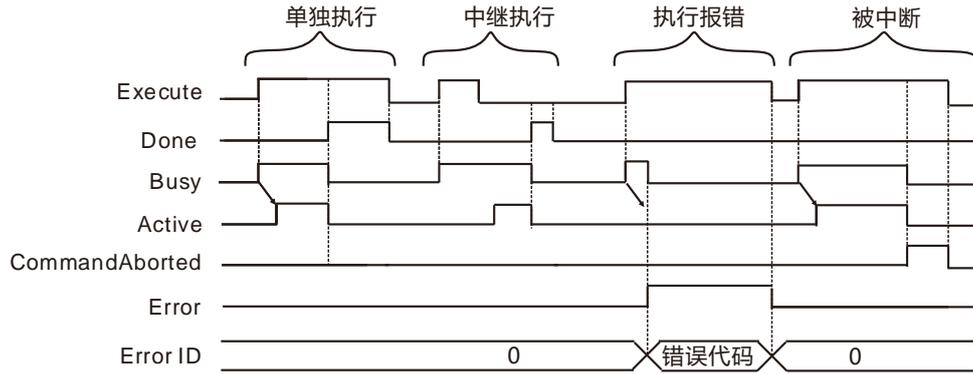
- **该指令执行时启动其它指令**

该指令执行时启动其它轴组运动指令，其它轴组运动指令和该指令如何中继由其它运动指令的 BufferMode（缓存模式）、TransitionMode（过渡模式）的值共同决定。其它运动指令和该指令缓存时，其它指令执行的时机为该指令 Done（完成）变为 TRUE 时。如果其它运动指令无 BufferMode 引脚，一般为中断该指令。该指令执行时，执行其它非轴组相关的运动指令，会打断该指令，指定轴的加速或者减速方式由执行指令的输入变量决定；轴组中其它轴进入 standstill（停止）状态，轴速度会一个周期降为 0。

- **异常排除**

该指令执行时，如果指令输入变量值不在允许的范围或者不满足指令的执行条件，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。该指令执行时，如遇到异常（如轴报警等），该指令也会报错，轴会立即停止。

## ■ 输出变量变化时序说明



### ● 单独执行

Execute（启动）由 FALSE 变为 TRUE 时，Busy（执行中）同时变为 TRUE，下一个周期 Active（控制中）变为 TRUE。当达到 Distance（移动距离）定位完成时，Done（完成）变为 TRUE，Busy（执行中）和 Active（控制中）同时变为 FALSE。当 Execute（启动）变为 FALSE 时，Done（完成）同时变为 FALSE。

### ● 中继执行

其它指令控制轴时，执行该指令（BufferMode 的值不为 mcAborting），当 Execute（启动）由 FALSE 变为 TRUE 时，Busy（执行中）同时变为 TRUE，Active（控制中）变为 TRUE 的时机为前一个指令完成时。

### ● 执行报错

该指令输入变量值不在允许范围内时，该指令 Execute（启动）由 FALSE 变为 TRUE 时，同时 Busy（执行中）变为 TRUE，下一个周期 Error（错误）变为 TRUE，同时 Busy（执行中）变为 FALSE，ErrorID（错误码）输出对应的错误码，可通过 ErrorID（错误码）的值，查找发生问题原因。该指令 Execute（启动）由 TRUE 变为 FALSE 时，同时 Error（错误）变为 FALSE，ErrorID（错误码）的值变为 0。

### ● 被中断

该指令执行后被其它指令中断时，该指令 CommandAborted（中断）变为 TRUE，Busy（执行中）和 Active（控制中）同时变为 FALSE；当 Execute（执行中）变为 FALSE 时，CommandAborted（中断）同时变为 FALSE。

## 7.11 MC\_MoveDirectRelative (相对值单独定位)

控制多个轴同时启动，停止由各个轴指定的移动距离决定，移动距离为相对值。所属库：MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_MoveDirectRelative	相对值单独定位	FB		<pre>MC_MoveDirectRelative_Instance (   AxesGroup := 参数 ,   Execute := 参数 ,   Distance:=参数 ,   CoordSystem:= 参数 ,   BufferMode:= 参数 ,   TransitionMode:= 参数 ,   TransitionParameter:= 参数 ,   Done =&gt; 参数 ,   Busy =&gt; 参数 ,   Active=&gt; 参数 ,   CommandAborted =&gt;参数 ,   Error =&gt; 参数 ,   ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
AxesGroup	轴组编号	USINT	由机种决定	不可缺省	轴组编号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
Distance	移动距离	LREAL	正数、负数、0	0	指定以当前位置为参考点的移动距离 (单位: 行程单位) *1
CoordSystem	坐标系	INT	保留	保留	保留
BufferMode	缓存模式	MC_Buffer_Mode	1: mcBuffered	1	设定两个指令之间中继模式*2 1: 等待
TransitionMode	过渡模式	MC_Transition_Mode	0: mcTMNone	0	设定当前控制轴的插补指令轨迹和缓存插补指令轨迹的连接方式 0: 位置到达时, 速度降为0
TransitionParameter	附加角	LREAL	正数、0	0	TransitionMode选择以附加角过渡时, 设定过渡参数

\*1: 指令单位的详细介绍请参考“[运动控制指令参数单位](#)”章节。

\*2: BufferMode的详细介绍请参考“[运动控制指令的多重启动时缓存模式说明 \(BufferMode\)](#)”章节。

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成后变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Active	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时, 输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	定位完成时	<ul style="list-style-type: none"> <li>◆ Done为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE</li> </ul>
Busy	Execute的上升沿	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
Active	指令开始控制轴时	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
CommandAborted	指令的执行被其它指令中断时	<ul style="list-style-type: none"> <li>◆ CommandAborted 为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令被其它指令中断时, CommandAborted变为TRUE, 一个周期后, 变为FALSE</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令执行过程中遇到异常时, Error变为TRUE, 一个周期后, 变为FALSE</li> </ul>

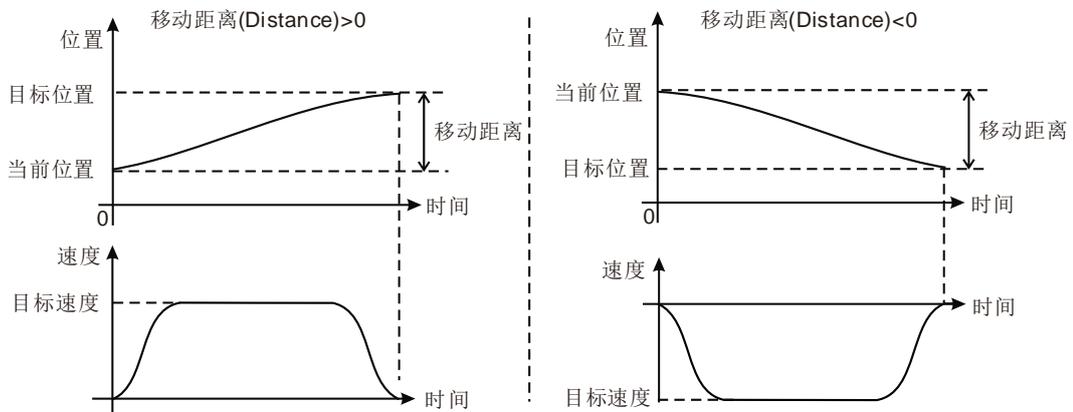
## ■ 功能说明

### ● 基本功能说明

该指令用于控制多个轴单独运动, 最多可以控制 8 个轴, 所有轴同时启动, 停止由各个轴指定的移动距离决定, 移动距离为该指令执行时各轴当前位置的相对值。各个轴移动的距离通过该指令输入变量 Distance 单独设定, 各个轴的目标速度、加速度、减速度、跃度由 MC\_AxesGroupEnable 指令的输入变量 Velocity、Acceleration、Deceleration、Jerk 设定。该指令在 Execute 由 FALSE 变为 TRUE 时, 按照输入变量设定的值进行运动。

### ● 移动距离

- 轴的目标位置由轴组中各个轴的当前位置和Distance的值共同决定, 即目标位置= 当前位置+Distance。Distance的值可以为正数、0、负数。Velocity、Acceleration、Deceleration、Jerk的值只能为正数。
- 轴静止时执行该指令, 执行该指令, Distance的值大于0和小于0时的位置曲线以及速度曲线如下图所示。Distance的值大于0时, 轴正向运转; Distance的值小0时, 则轴反向运转。
- Distance的值为0时, 该指令不控制轴运转, 该指令的Done在执行该指令一个扫描周期后变为TRUE。
- Distance的值等于0时, 该指令不控制轴运转, 该指令的Done在执行该指令一个扫描周期后变为TRUE。



### ● 指令完成时机

轴组中各个轴的命令位置到达设定的距离时, 该指令完成, Done 由 FALSE 变为 TRUE。

### ● 重启该指令

当指令执行完成后，Execute 再次由 FALSE 变为 TRUE 时，该指令可以重新执行；当指令正在执行中，Execute 再次由 FALSE 变为 TRUE 时，对指令的执行不会产生影响，指令仍按照未执行完成的输入变量执行指令。

- **其它指令执行时启动该指令**

其它轴组运动指令执行时，启动该指令，该指令 BufferMode（缓存模式）只能选择 mcBuffered（等待），TransitionMode（过渡模式）只能选择 mcTMNone（位置到达时，速度降为 0）。

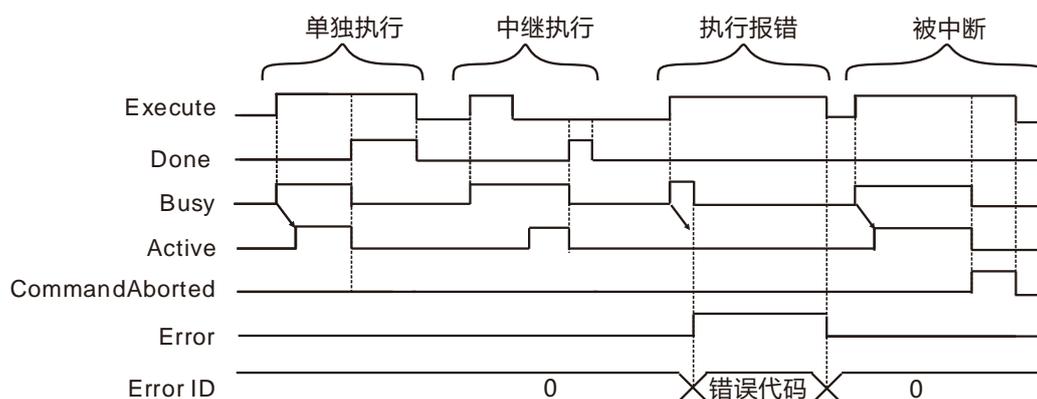
- **该指令执行时启动其它指令**

该指令执行时启动其它轴组运动指令，其它轴组运动指令和该指令如何中继由其它运动指令的 BufferMode（缓存模式）、TransitionMode（过渡模式）的值共同决定。其它运动指令和该指令缓存时，其它指令执行的时机为该指令 Done（完成）变为 TRUE 时。如果其它运动指令无 BufferMode 引脚，一般为中断该指令。该指令执行时，执行其它非轴组相关的运动指令，会打断该指令，指定轴的加速或者减速方式由执行指令的输入变量决定；轴组中其它轴进入 standstill（停止）状态，轴速度会一个周期降为 0。

- **异常排除**

该指令执行时，如果指令输入变量值不在允许的范围或者不满足指令的执行条件，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。该指令执行时，如遇到异常（如轴报警等），该指令也会报错，轴会立即停止。

## ■ 输出变量变化时序说明



- **单独执行**

Execute（启动）由 FALSE 变为 TRUE 时，Busy（执行中）同时变为 TRUE，下一个周期 Active（控制中）变为 TRUE。当达到 Distance（移动距离）、定位完成时，Done（完成）变为 TRUE，Busy（执行中）和 Active（控制中）同时变为 FALSE。当 Execute（启动）变为 FALSE 时，Done（完成）同时变为 FALSE。

- **中继执行**

其它指令控制轴时，执行该指令（BufferMode 的值不为 mcAborting），当 Execute（启动）由 FALSE 变为 TRUE 时，Busy（执行中）同时变为 TRUE，Active（控制中）变为 TRUE 的时机为前一个指令完成时。

- **执行报错**

该指令输入变量值不在允许范围内时，该指令 Execute（启动）由 FALSE 变为 TRUE 时，同时 Busy（执行中）变为 TRUE，下一个周期 Error（错误）变为 TRUE，同时 Busy（执行中）变为 FALSE，ErrorID（错误码）输出对应的错误码，可通过 ErrorID（错误码）的值，查找发生问题原因。该指令 Execute（启动）由 TRUE 变为 FALSE 时，同时 Error（错误）变为 FALSE，ErrorID（错误码）的值变为 0。

- **被中断**

该指令执行后被其它指令中断时，该指令 CommandAborted（中断）变为 TRUE，Busy（执行中）和 Active（控制中）同时变为 FALSE；当 Execute（执行中）变为 FALSE 时，CommandAborted（中断）同时变为 FALSE。

## 7.12 MC\_MoveDirectAbsolute (绝对值单独定位)

控制多个轴同时启动，停止由各个轴的当前位置和指定的绝对位置决定，指定位置为绝对值。所属库：MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_MoveDirectAbsolute	绝对值单独定位	FB		<pre>MC_MoveDirectAbsolute_Instance (   AxesGroup := 参数 ,   Execute := 参数 ,   Position := 参数 ,   CoordSystem := 参数 ,   BufferMode := 参数 ,   TransitionMode := 参数 ,   TransitionParameter := 参数 ,   Done =&gt; 参数 ,   Busy =&gt; 参数 ,   Active =&gt; 参数 ,   CommandAborted =&gt; 参数 ,   Error =&gt; 参数 ,   ErrorID =&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
AxesGroup	轴组编号	USINT	由机种决定	不可缺省	轴组编号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
Position	绝对位置	LREAL	正数、负数、0	0	指定以0点位置为参考点的绝对位置 (单位: 行程单位) *2
Velocity	目标速度	LREAL	正数	不可缺省	指定目标速度*1 (单位: 行程单位/秒) *2
Acceleration	加速度	LREAL	正数	不可缺省	指定加速度*1 (单位: 行程单位/秒 <sup>2</sup> ) *2
Deceleration	减速度	LREAL	正数	不可缺省	指定减速度*1 (单位: 行程单位/秒 <sup>2</sup> ) *2
Jerk	跃度	LREAL	正数	不可缺省	指定跃度*1 (单位: 行程单位/秒 <sup>3</sup> ) *2
CoordSystem	坐标系	INT	保留	保留	保留
BufferMode	缓存模式	MC_Buffer_Mode	1: mcBuffered	1	设定两个指令之间中继模式*3 1: 等待
TransitionMode	过渡模式	MC_Transition_Mode	0: mcTMNone	0	设定当前控制轴的插补指令轨迹和缓存插补指令轨迹的连接方式 0: 位置到达时, 速度降为0
TransitionParameter	附加角	LREAL	正数、0	0	保留

\*1: Velocity、Acceleration、Deceleration、Jerk的关系请参考“[运动控制指令相关参数说明](#)”章节。

\*2: 指令单位的详细介绍请参考“[运动控制指令参数单位](#)”章节。

\*3: BufferMode的详细介绍请参考“[运动控制指令的多重启动时缓存模式说明 \(BufferMode\)](#)”章节。

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成后变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Active	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE

CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

## ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	定位完成时	<ul style="list-style-type: none"> <li>◆ Done为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE</li> </ul>
Busy	Execute的上升沿	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
Active	指令开始控制轴时	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
CommandAborted	指令的执行被其它指令中断时	<ul style="list-style-type: none"> <li>◆ CommandAborted 为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令被其它指令中断时, CommandAborted变为TRUE, 一个周期后, 变为FALSE</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令执行过程中遇到异常时, Error变为TRUE, 一个周期后, 变为FALSE</li> </ul>

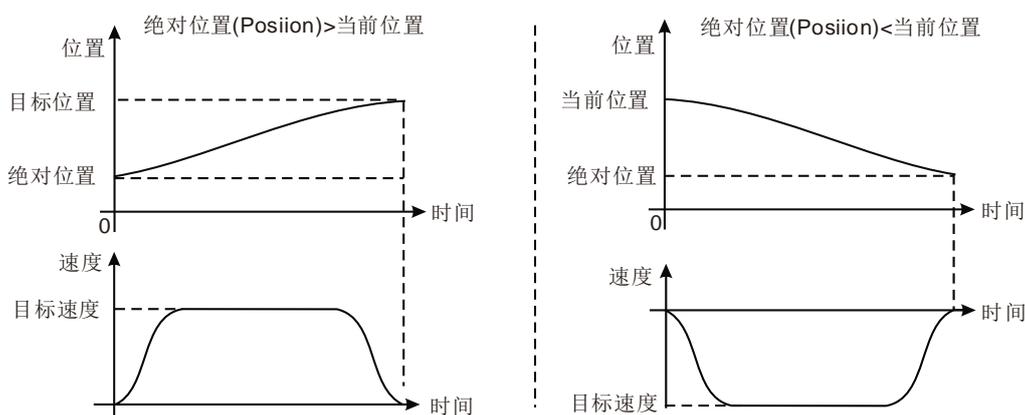
## ■ 功能说明

### ● 基本功能说明

该指令用于控制多个轴单独运动，最多可以控制 8 个轴，所有轴同时启动，停止由各个轴指定的绝对位置决定，绝对位置表示以 0 点位置为参考点的绝对位置。各个轴的绝对位置通过该指令输入变量 Position 单独设定，各个轴的目标速度、加速度、减速度、跃度由 MC\_AxesGroupEnable 指令的输入变量 Velocity、Acceleration、Deceleration、Jerk 设定。该指令在 Execute 由 FALSE 变为 TRUE 时，按照输入变量设定的值进行运动

### ● 绝对位置

- 输入变量Position的值表示以0点位置为参考点的绝对位置。该值可以为正数、0、负数。Velocity、Acceleration、Deceleration、Jerk的值只能为正数。
- 轴静止时执行该指令，执行该指令，Position的值大于当前位置和小于当前位置时的位置曲线以及速度曲线如下图所示。Position的值大于当前位置时，轴正向运转；Position的值小于当前位置时，则轴反向运转。
- Position的值等于当前位置时，该指令不控制轴运转，该指令的Done在执行该指令一个扫描周期后变为TRUE。



- **指令完成时机**

轴组中各个轴的命令位置到达设定的位置时，该指令完成，Done 由 FALSE 变为 TRUE。

- **重启该指令**

当指令执行完成后，Execute 再次由 FALSE 变为 TRUE 时，该指令可以重新执行；当指令正在执行中，Execute 再次由 FALSE 变为 TRUE 时，对指令的执行不会产生影响，指令仍按照未执行完成的输入变量执行指令。

- **其它指令执行时启动该指令**

其它轴组运动指令执行时，启动该指令，该指令 BufferMode（缓存模式）只能选择 mcBuffered（等待），TransitionMode（过渡模式）只能选择 mcTMNone（位置到达时，速度降为 0）。

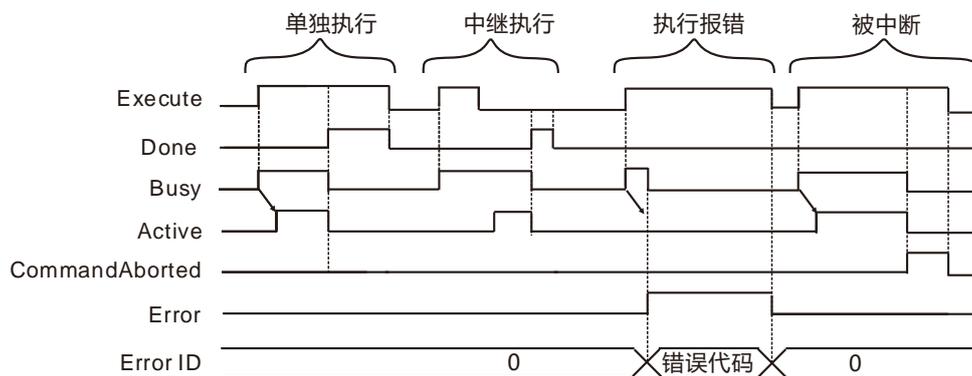
- **该指令执行时启动其它指令**

该指令执行时启动其它轴组运动指令，其它轴组运动指令和该指令如何中继由其它运动指令的 BufferMode（缓存模式）、TransitionMode（过渡模式）的值共同决定。其它运动指令和该指令缓存时，其它指令执行的时机为该指令 Done（完成）变为 TRUE 时。如果其它运动指令无 BufferMode 引脚，一般为中断该指令。该指令执行时，执行其它非轴组相关的运动指令，会打断该指令，指定轴的加速或者减速方式由执行指令的输入变量决定；轴组中其它轴进入 standstill（停止）状态，轴速度会一个周期降为 0。

- **异常排除**

该指令执行时，如果指令输入变量值不在允许的范围或者不满足指令的执行条件，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。该指令执行时，如遇到异常（如轴报警等），该指令也会报错，轴会立即停止。

- **输出变量变化时序说明**



- **单独执行**

Execute（启动）由 FALSE 变为 TRUE 时，Busy（执行中）同时变为 TRUE，下一个周期 Active（控制中）变为 TRUE。当达到 Distance（移动距离）、定位完成时，Done（完成）变为 TRUE，Busy（执行中）和 Active（控制中）同时变为 FALSE。当 Execute（启动）变为 FALSE 时，Done（完成）同时变为 FALSE。

- **中继执行**

其它指令控制轴时，执行该指令（BufferMode 的值不为 mcAborting），当 Execute（启动）由 FALSE 变为 TRUE 时，Busy（执行中）同时变为 TRUE，Active（控制中）变为 TRUE 的时机为前一个指令完成时。

- **执行报错**

该指令输入变量值不在允许范围内时，该指令 Execute（启动）由 FALSE 变为 TRUE 时，同时 Busy（执行中）变为 TRUE，下一个周期 Error（错误）变为 TRUE，同时 Busy（执行中）变为 FALSE，ErrorID（错误码）输出对应的错误码，可通过 ErrorID（错误码）的值，查找发生问题原因。该指令 Execute（启动）由 TRUE 变为 FALSE 时，同时 Error（错误）变为 FALSE，ErrorID（错误码）的值变为 0。

- **被中断**

该指令执行后被其它指令中断时，该指令 CommandAborted（中断）变为 TRUE，Busy（执行中）和 Active（控制中）同时变为 FALSE；当 Execute（执行中）变为 FALSE 时，CommandAborted（中断）同时变为 FALSE。

## 7.13 MC\_MoveCircularRelative (相对值圆弧插补)

控制多个轴同时启动和停止，两个轴进行圆弧插补，其它轴跟随圆弧运动同时启动和停止。所属库：MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_MoveCircularRelative	相对值圆弧插补	FB	<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">MC_MoveCircularRelative_Instance</p> <p style="text-align: center;"><b>MC_MoveCircularRelative</b></p> <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p>— AxesGroup</p> <p>— Execute</p> <p>— CircMode</p> <p>— AuxPoint</p> <p>— EndPoint</p> <p>— MultiTurn</p> <p>— PathChoice</p> <p>— Velocity</p> <p>— Acceleration</p> <p>— Deceleration</p> <p>— Jerk</p> <p>— CoordSystem</p> <p>— BufferMode</p> <p>— TransitionMode</p> <p>— TransitionParameter</p> </div> <div style="width: 45%;"> <p>Done</p> <p>Busy</p> <p>Active</p> <p>CommandAborted</p> <p>Error</p> <p>ErrorID</p> </div> </div> </div>	<pre> MC_MoveCircularRelative_Instance (   AxesGroup := 参数 ,   Execute := 参数 ,   CircMode:= 参数 ,   AuxPoint:= 参数 ,   EndPoint:= 参数 ,   MultiTurn:= 参数 ,   PathChoice:= 参数 ,   Velocity:= 参数 ,   Acceleration:= 参数 ,   Deceleration:= 参数 ,   Jerk:= 参数 ,   CoordSystem:= 参数 ,   BufferMode:= 参数 ,   TransitionMode:= 参数 ,   TransitionParameter:= 参数 ,   Done =&gt; 参数 ,   Busy =&gt; 参数 ,   Active=&gt; 参数 ,   CommandAborted =&gt; 参数 ,   Error =&gt; 参数 ,   ErrorID=&gt; 参数 );                     </pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
AxesGroup	轴组编号	USINT	由机种决定	不可缺省	轴组编号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
CircMode	圆弧插补模式	INT	0~5	0	设定圆弧插补的模式 0: XY平面指定圆心坐标 1: XZ平面指定圆心坐标 2: YZ平面指定圆心坐标 3: XY平面指定半径 4: XZ平面指定半径 5: YZ平面指定半径
AuxPoint	圆心坐标或半径	ARRAY [1..2] OF LREAL	负数、正数、0	0	指定圆心坐标画圆弧时，该参数表示圆心坐标； 指定半径画圆时，Auxpoint[1]表示半径，AuxPoint[2]无作用。
EndPoint	终点坐标	ARRAY [1..8] OF LREAL	负数、正数、0	0	各轴以当前点为参考点相对位置。
MultiTurn	圈数	UINT	正数、0	0	设置画圆的圈数
PathChoice	路径选择	INT	0、1	0	圆弧路径的方向 0: 顺时针方向 1: 逆时针方向
Velocity	目标速度	LREAL	正数	不可缺省	指定目标速度*1 (单位: 行程单位/秒) *2
Acceleration	加速度	LREAL	正数	不可缺省	指定加速度*1

					(单位: 行程单位/秒 <sup>2</sup> ) *2
Deceleration	减速度	LREAL	正数	不可缺省	指定减速度*1 (单位: 行程单位/秒 <sup>2</sup> ) *2
Jerk	跃度	LREAL	正数	不可缺省	指定跃度*1 (单位: 行程单位/秒 <sup>3</sup> ) *2
CoordSystem	坐标系	INT	保留	保留	保留
BufferMode	缓存模式	MC_Buffer_Mode	1: mcBuffered 3: mcBlending Previous	不可缺省	设定两个指令之间中继模式*3 1: 等待
TransitionMode	过渡模式	MC_Transition_Mode	0: mcTMNone 3: mcTMCorner- Distance	不可缺省	设定当前控制轴的插补指令轨迹和缓存 插补指令轨迹的连接方式 0: 位置到达时, 速度降为0
TransitionParameter	附加角	LREAL	正数、0	0	TransitionMode选择以附加角过渡 时, 设定过渡参数

\*1: Velocity、Acceleration、Deceleration、Jerk的关系请参考“[运动控制指令相关参数说明](#)”章节。

\*2: 指令单位的详细介绍请参考“[运动控制指令参数单位](#)”章节。

\*3: BufferMode的详细介绍请参考“[运动控制指令的多重启动时缓存模式说明 \(BufferMode\)](#)”章节。

## ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成后变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Active	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时, 输出错误代码 值的含义请参考“ <a href="#">指令错误代码描述</a> ”

## ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	定位完成时	<ul style="list-style-type: none"> <li>◆ Done为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE</li> </ul>
Busy	Execute的上升沿	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
Active	指令开始控制轴时	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
CommandAborted	指令的执行被其它指令中断时	<ul style="list-style-type: none"> <li>◆ CommandAborted 为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令被其它指令中断时, CommandAborted变为TRUE, 一个周期后, 变为FALSE</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令执行过程中遇到异常时, Error变为TRUE, 一个周期后, 变为FALSE</li> </ul>

## ■ 功能说明

### ● 基本功能说明

该指令用于控制多个轴同时启动和停止，两个轴进行圆弧插补，其它轴跟随圆弧运动同时启动和停止。该指令在 Execute 由 FALSE 变为 TRUE 时，按照输入变量设定的值进行运动。

### ● 目标速度

输入变量 Velocity 为所有轴的合成速度，合成速度和各个轴的速度关系：合成速度的平方和= 各轴速度的平方和。该指令输入变量 Acceleration、Deceleration 为所有轴的合成加速度和减速度，合成加速度、减速度和各个轴加速度、减速度关系：合成加（减）速度 = 各轴加（减）速度的平方和。

### ● 指令完成时机

轴组中各个轴的命令位置到达设定的距离时，该指令完成，Done 由 FALSE 变为 TRUE。

### ● 重启该指令

当指令执行完成后，Execute 再次由 FALSE 变为 TRUE 时，该指令可以重新执行；当指令正在执行中，Execute 再次由 FALSE 变为 TRUE 时，对指令的执行不会产生影响，指令仍按照未执行完成的输入变量执行指令。

### ● 其它指令执行时启动该指令

其它轴组运动指令执行时，启动该指令，该指令和其它运动指令如何中继由该指令输入变量 BufferMode（缓存模式）、TransitionMode（过渡模式）的值共同决定。详细说明如下表所示。

BufferMode（缓存模式）	TransitionMode（过渡模式）	含义
mcBuffered (等待)	mcTMNone(位置到达时，速度降为0)	等待当前轴组指令执行完成时（目标位置到达时，速度降为0），切换为缓存轴组指令控制轴。
mcBlendingPrevious (以当前指令的目标速度中继)	mcTMCornerDistance (以设定的附加角过渡)	当前轴组指令目标位置完成时刻为当前轴组指令的目标位置减去mcTMCornerDistance设定的附加角位置。当前轴组指令执行完成后，立即执行缓存轴组指令，缓存轴组指令先以mcTMCornerDistance设定的附加角运动，然后再以缓存轴组指令设定的参数运动。

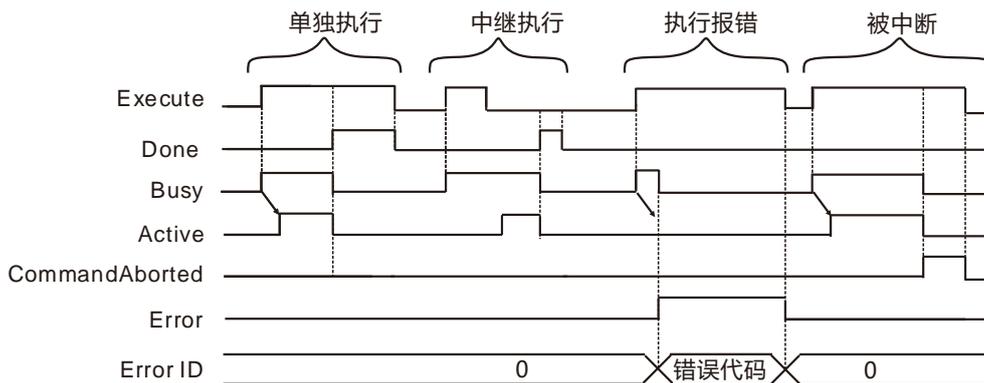
### ● 该指令执行时启动其它指令

该指令执行时启动其它轴组运动指令，其它轴组运动指令和该指令如何中继由其它运动指令的 BufferMode（缓存模式）、TransitionMode（过渡模式）的值共同决定。其它运动指令和该指令缓存时，其它指令执行的时机为该指令 Done（完成）变为 TRUE 时。如果其它运动指令无 BufferMode 引脚，一般为中断该指令。该指令执行时，执行其它非轴组相关的运动指令，会打断该指令，指定轴的加速或者减速方式由执行指令的输入变量决定；轴组中其它轴进入 standstill（停止）状态，轴速度会一个周期降为 0。

### ● 异常排除

该指令执行时，如果指令输入变量值不在允许的范围或者不满足指令的执行条件，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。该指令执行时，如遇到异常（如轴报警等），该指令也会报错，轴会立即停止。

## ■ 输出变量变化时序说明



### ● 单独执行

Execute (启动) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE, 下一个周期 Active (控制中) 变为 TRUE。当达到 Distance (移动距离)、定位完成时, Done (完成) 变为 TRUE, Busy (执行中) 和 Active (控制中) 同时变为 FALSE。当 Execute (启动) 变为 FALSE 时, Done (完成) 同时变为 FALSE。

### ● 中继执行

其它指令控制轴时, 执行该指令 (BufferMode 的值不为 mcAborting), 当 Execute (启动) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE, Active (控制中) 变为 TRUE 的时机为前一个指令完成时。

### ● 执行报错

该指令输入变量值不在允许范围内时, 该指令 Execute (启动) 由 FALSE 变为 TRUE 时, 同时 Busy (执行中) 变为 TRUE, 下一个周期 Error (错误) 变为 TRUE, 同时 Busy (执行中) 变为 FALSE, ErrorID (错误码) 输出对应的错误码, 可通过 ErrorID (错误码) 的值, 查找发生问题原因。该指令 Execute (启动) 由 TRUE 变为 FALSE 时, 同时 Error (错误) 变为 FALSE, ErrorID (错误码) 的值变为 0。

### ● 被中断

该指令执行后被其它指令中断时, 该指令 CommandAborted (中断) 变为 TRUE, Busy (执行中) 和 Active (控制中) 同时变为 FALSE; 当 Execute (执行中) 变为 FALSE 时, CommandAborted (中断) 同时变为 FALSE。

## 7.14 MC\_MoveCircularAbsolute (绝对值圆弧插补)

控制多个轴同时启动和停止，两个轴进行圆弧插补，其它轴跟随圆弧运动同时启动和停止。所属库：MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_MoveCircularAbsolute	绝对值圆弧插补	FB		<pre> MC_MoveCircularAbsolute_Instance (   AxesGroup := 参数 ,   Execute := 参数 ,   CircMode:= 参数 ,   AuxPoint:= 参数 ,   EndPoint:= 参数 ,   MultiTurn:= 参数 ,   PathChoice:= 参数 ,   Velocity:= 参数 ,   Acceleration:=参数 ,   Deceleration:=参数 ,   Jerk:= 参数 ,   CoordSystem:= 参数 ,   BufferMode:= 参数 ,   TransitionMode:= 参数 ,   TransitionParameter:= 参数 ,   Done =&gt; 参数 ,   Busy =&gt; 参数 ,   Active=&gt; 参数 ,   CommandAborted =&gt; 参数 ,   Error =&gt; 参数 ,   ErrorID=&gt; 参数 );                     </pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
AxesGroup	轴组编号	USINT	由机种决定	不可缺省	轴组编号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
CircMode	圆弧插补模式	INT	0~5	0	设定圆弧插补的模式 0: XY平面指定圆心坐标 1: XZ平面指定圆心坐标 2: YZ平面指定圆心坐标 3: XY平面指定半径 4: XZ平面指定半径 5: YZ平面指定半径
AuxPoint	圆心坐标或半径	ARRAY [1..2] OF LREAL	负数、正数、0	0	指定圆心坐标画圆弧时，该参数表示圆心坐标； 指定半径画圆时，Auxpoint[1]表示半径，AuxPoint[2]无作用。
EndPoint	终点坐标	ARRAY [1..8] OF LREAL	负数、正数、0	0	圆弧终点各轴的绝对对位置。
MultiTurn	圈数	UINT	正数、0	0	设置画圆的圈数
PathChoice	路径选择	INT	0、1	0	圆弧路径的方向 0: 顺时针方向 1: 逆时针方向
Velocity	目标速度	LREAL	正数	不可缺省	指定目标速度*1 (单位: 行程单位/秒) *2

Acceleration	加速度	LREAL	正数	不可缺省	指定加速度* <sup>1</sup> (单位: 行程单位/秒 <sup>2</sup> ) * <sup>2</sup>
Deceleration	减速度	LREAL	正数	不可缺省	指定减速度* <sup>1</sup> (单位: 行程单位/秒 <sup>2</sup> ) * <sup>2</sup>
Jerk	跃度	LREAL	正数	不可缺省	指定跃度* <sup>1</sup> (单位: 行程单位/秒 <sup>3</sup> ) * <sup>2</sup>
CoordSystem	坐标系	INT	保留	保留	保留
BufferMode	缓存模式	MC_Buffer_Mode	1: mcBuffered 3: mcBlending Previous	不可缺省	设定两个指令之间中继模式* <sup>3</sup> 1: 等待 3: 以当前指令目标速度中继
TransitionMode	过渡模式	MC_Transition_Mode	0: mcTMNone 3: mcTMCorner- Distance	不可缺省	设定当前控制轴的插补指令轨迹和缓存 插补指令轨迹的连接方式 0: 位置到达时, 速度降为0 3: 以附加角过渡
TransitionParameter	附加角	LREAL	正数、0	0	TransitionMode选择以附加角过渡 时, 设定过渡参数

\*1: Velocity、Acceleration、Deceleration、Jerk的关系请参考“[运动控制指令相关参数说明](#)”章节。

\*2: 指令单位的详细介绍请参考“[运动控制指令参数单位](#)”章节。

\*3: BufferMode的详细介绍请参考“[运动控制指令的多重启动时缓存模式说明 \(BufferMode\)](#)”章节。

## ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成后变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Active	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时, 输出错误代码 值的含义请参考“ <a href="#">指令错误代码描述</a> ”

## ■ 输出变量刷新时期

名称	变为TRUE的时机	变为FALSE的时机
Done	定位完成时	<ul style="list-style-type: none"> <li>◆ Done为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE</li> </ul>
Busy	Execute的上升沿	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
Active	指令开始控制轴时	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
CommandAborted	指令的执行被其它指令中断时	<ul style="list-style-type: none"> <li>◆ CommandAborted 为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令被其它指令中断时, CommandAborted变为TRUE, 一个周期后, 变为FALSE</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Execute从TRUE变为FALSE时</li> <li>◆ 指令已执行且Execute变为FALSE, 该指令执行过程中遇到异常时, Error变为TRUE, 一个周期后, 变为FALSE</li> </ul>

## ■ 功能说明

### ● 基本功能说明

该指令用于控制多个轴同时启动和停止，两个轴进行圆弧插补，其它轴跟随圆弧运动同时启动和停止，终点位置为以 0 点位置为参考点的绝对目标位置。该指令在 Execute 由 FALSE 变为 TRUE 时，按照输入变量设定的值进行运动。

### ● 目标速度

输入变量 Velocity 为所有轴的合成速度，合成速度和各个轴的速度关系：合成速度的平方和= 各轴速度的平方和。该指令输入变量 Acceleration、Deceleration 为所有轴的合成加速度和减速度，合成加速度、减速度和各个轴加速度、减速度关系：合成加（减）速度 = 各轴加（减）速度的平方和。

### ● 指令完成时机

轴组中各个轴的命令位置到达设定的位置时，该指令完成，Done 由 FALSE 变为 TRUE。

### ● 重启该指令

当指令执行完成后，Execute 再次由 FALSE 变为 TRUE 时，该指令可以重新执行；当指令正在执行中，Execute 再次由 FALSE 变为 TRUE 时，对指令的执行不会产生影响，指令仍按照未执行完成的输入变量执行指令。

### ● 其它指令执行时启动该指令

其它轴组运动指令执行时，启动该指令，该指令和其它运动指令如何中继由该指令输入变量 BufferMode（缓存模式）、TransitionMode（过渡模式）的值共同决定。详细说明如下表所示。

BufferMode（缓存模式）	TransitionMode（过渡模式）	含义
mcBuffered (等待)	mcTMNone(位置到达时，速度降为0)	等待当前轴组指令执行完成时（目标位置到达时，速度降为0），切换为缓存轴组指令控制轴。
mcBlendingPrevious (以当前指令的目标速度中继)	mcTMCornerDistance (以设定的附加角过渡)	当前轴组指令目标位置完成时刻为当前轴组指令的目标位置减去mcTMCornerDistance设定的附加角位置。当前轴组指令执行完成后，立即执行缓存轴组指令，缓存轴组指令先以mcTMCornerDistance设定的附加角过度运动，然后再以缓存轴组指令设定的参数运动。

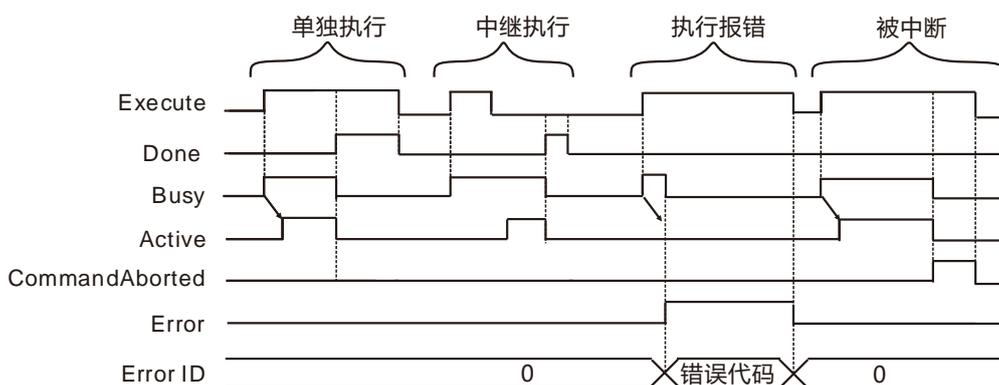
### ● 该指令执行时启动其它指令

该指令执行时启动其它轴组运动指令，其它轴组运动指令和该指令如何中继由其它运动指令的 BufferMode（缓存模式）、TransitionMode（过渡模式）的值共同决定。其它运动指令和该指令缓存时，其它指令执行的时机为该指令 Done（完成）变为 TRUE 时。如果其它运动指令无 BufferMode 引脚，一般为中断该指令。该指令执行时，执行其它非轴组相关的运动指令，会打断该指令，指定轴的加速或者减速方式由执行指令的输入变量决定；轴组中其它轴进入 standstill（停止）状态，轴速度会一个周期降为 0。

### ● 异常排除

该指令执行时，如果指令输入变量值不在允许的范围或者不满足指令的执行条件，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。该指令执行时，如遇到异常（如轴报警等），该指令也会报错，轴会立即停止。

## ■ 输出变量变化时序说明



- **单独执行**

Execute (启动) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE, 下一个周期 Active (控制中) 变为 TRUE。当达到 Distance (移动距离)、定位完成时, Done (完成) 变为 TRUE, Busy (执行中) 和 Active (控制中) 同时变为 FALSE。当 Execute (启动) 变为 FALSE 时, Done (完成) 同时变为 FALSE。

- **中继执行**

其它指令控制轴时, 执行该指令 (BufferMode 的值不为 mcAborting), 当 Execute (启动) 由 FALSE 变为 TRUE 时, Busy (执行中) 同时变为 TRUE, Active (控制中) 变为 TRUE 的时机为前一个指令完成时。

- **执行报错**

该指令输入变量值不在允许范围内时, 该指令 Execute (启动) 由 FALSE 变为 TRUE 时, 同时 Busy (执行中) 变为 TRUE, 下一个周期 Error (错误) 变为 TRUE, 同时 Busy (执行中) 变为 FALSE, ErrorID (错误码) 输出对应的错误码, 可通过 ErrorID (错误码) 的值, 查找发生问题原因。该指令 Execute (启动) 由 TRUE 变为 FALSE 时, 同时 Error (错误) 变为 FALSE, ErrorID (错误码) 的值变为 0。

- **被中断**

该指令执行后被其它指令中断时, 该指令 CommandAborted (中断) 变为 TRUE, Busy (执行中) 和 Active (控制中) 同时变为 FALSE; 当 Execute (执行中) 变为 FALSE 时, CommandAborted (中断) 同时变为 FALSE。

## 7.15 MC\_AxesGroupReadActualPosition (轴组中各个轴反馈位置读取)

该指令用于周期性读取轴组中各个轴的反馈位置。所属库: MotionControl\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
MC_AxesGroupReadActualPos	轴组中各个轴反馈位置读取	FB		<pre>MC_AxesGroupReadActualPos_Instance ( Axis := 参数 , Enable := 参数 , CoordSystem:= 参数 , Valid =&gt; 参数 , Busy =&gt; 参数 , Error =&gt; 参数 , ErrorID=&gt; 参数 , Position=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
AxesGroup	轴组编号	USINT	由机种决定	不可缺省	轴组编号
Enable	有效	BOOL	TRUE或FALSE	FALSE	设为TRUE, 读取指定轴反馈位置 设为FALSE, 停止读取指定轴反馈位置
CoordSystem	坐标系	INT	保留	—	保留

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Valid	有效	BOOL	TRUE或FALSE	指令执行完成后变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时, 输出错误代码 值的含义请参考“ <a href="#">指令错误代码描述</a> ”
Position	反馈位置	ARRAY [1..8] OF LREAL	负数、正数、0	根据指定轴的反馈位置通过轴“设置参数”换算后的位置 单位: 行程单位

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Valid	Position (反馈位置) 的值可以读取时	<ul style="list-style-type: none"> <li>◆ Enable变为FALSE时</li> <li>◆ Error变为TRUE时</li> </ul>
Busy	当Enable 为TRUE时	<ul style="list-style-type: none"> <li>◆ Enable变为FALSE时</li> <li>◆ Error为TRUE时</li> </ul>
Error	指令输入变量值不在允许的范围时。	<ul style="list-style-type: none"> <li>◆ Error为TRUE, Enable由TRUE变为FALSE时</li> </ul>

### ■ 功能说明

#### ● 基本功能说明

该指令用于周期读取指定轴组中各轴的反馈位置, 每个同步时钟周期更新一次; 该指令输出变量 Position 是一个一维数组, 数组中每个成员对应轴组中一个轴的位置, 最多可以读取轴组中 8 个轴的位置。该指令输入变量 Enable 为 TRUE 时, Position 的值每个同步时钟周期更新一次; Enable 为 FALSE 时, 轴反馈位置 Position 的值停止更新, 并保持最后 Enable 为 TRUE 时的值不变。

- **反馈位置计算方法**

反馈位置 (Position) 的值为通过轴位置换算至机构中的位置, 单位为行程单位。指定轴为伺服轴时, 该指令读取位置来源为电机反馈给驱动器的脉冲数; 虚拟伺服轴时, 该指令读取位置来源为虚拟伺服轴的命令位置, 不需要换算。指定轴为伺服轴或者编码器轴时, 反馈位置 (Position) 的值通过上述来源经过机构参数换算后得到, 机构参数可以在软件中 【运动控制】 → 【轴设置】 → 【基本设置】 中查看, 转换关系计算方法如下图所示, 此转换都是控制器进行运算, 不需要用户运算。

$$\text{反馈位置 (Position)} = \frac{\text{电机反馈给驱动器的脉冲数} \times \text{工作每转的工作行程} \times \text{减速机输出转速}}{\text{或编码器接收脉冲数} \times \text{电机每转的脉冲数} \times \text{减速机输入转速}}$$

- **反馈位置更新周期**

该指令读取轴的反馈位置, 一个同步周期更新一次, 同一个同步周期内不同时刻, 该指令读取到反馈位置相同。

- **MC\_Home、MC\_SetPosition 和 MC\_HomeWithParm 对实际位置的影响**

MC\_Home、MC\_SetPosition 和 MC\_HomeWithParm 指令执行后, 轴命令位置和反馈位置会跟随这些指令的设置位置变化, 这些指令执行后, 该指令读取到的位置和轴结构体中的反馈位置一致。

- **异常排除**

该指令执行时, 如果输入变量不合法, 该指令 Error 变为 TRUE, ErrorID 中有对应的错误码, 可以通过该错误码判断错误原因。

## 7.16 轴组运动指令示例程序

### ■ 轴组运动指令执行需要的步骤

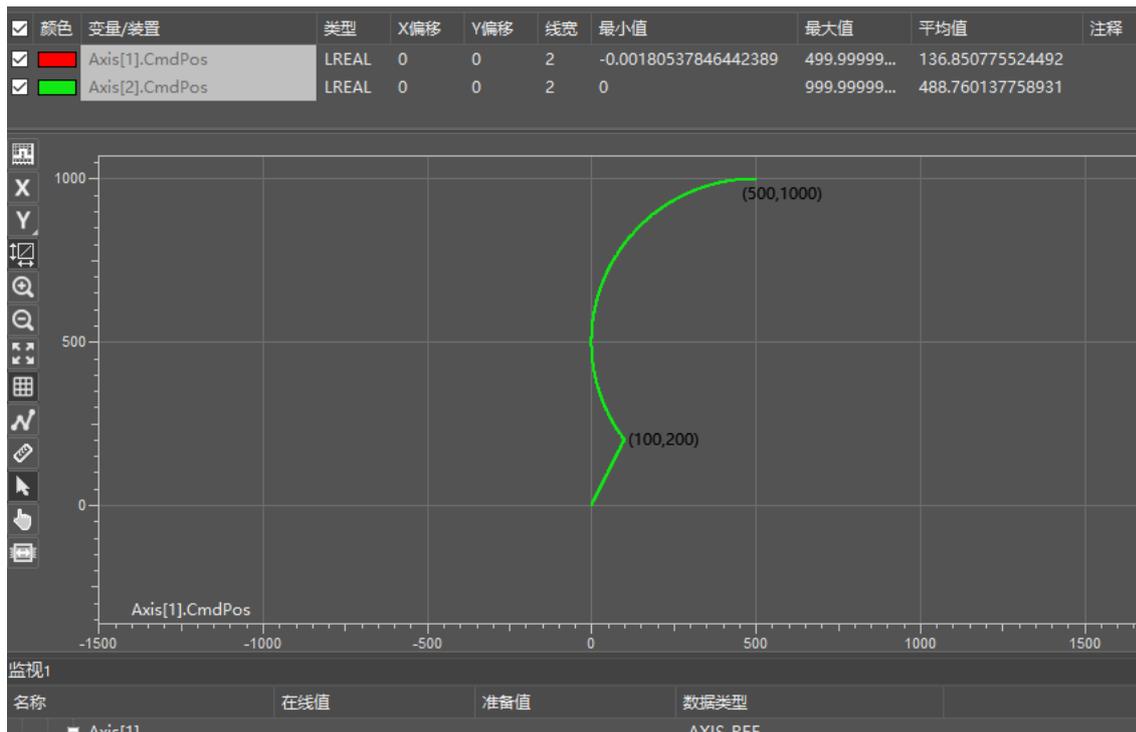
步骤	步骤描述	备注
步骤1	设置各个轴的轴参数	在软件中设定
步骤2	轴使能	伺服轴执行MC_Power指令，虚拟伺服轴不需要
步骤3	添加轴到轴组	轴组中的逻辑轴和软件中的配置的轴建立对应关系
步骤4	轴组使能	轴组使能
步骤5	轴组运动指令执行	轴组运动指令执行

### ■ 轴参数设置

轴1和轴2轴参数设置如下图所示，轴1和轴2轴参数设置相同。



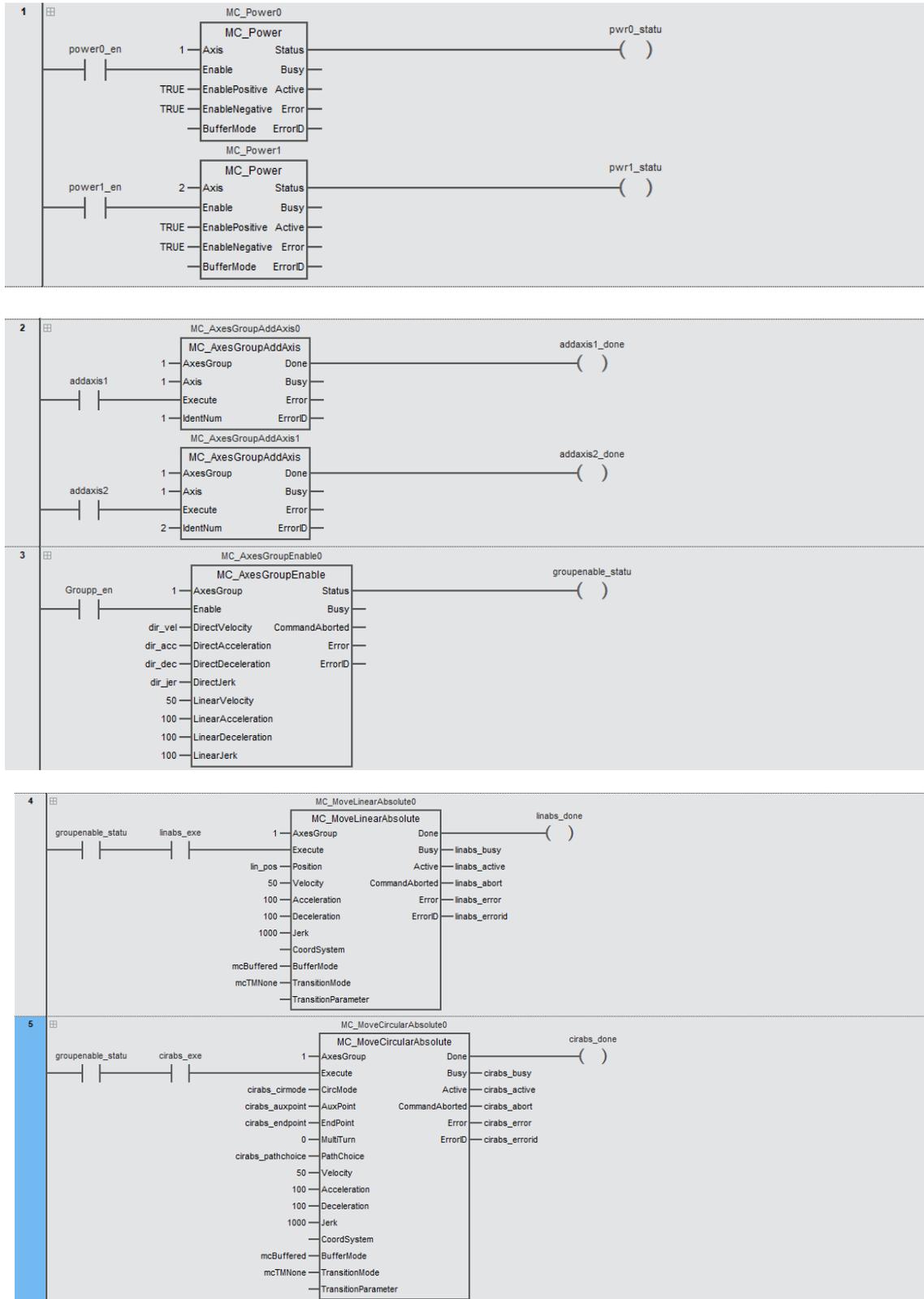
### ■ 轴组运动轨迹曲线图



## ● 变量表

类别	名称	分配到	数据类型	初始值	注释
VAR	MC_Power0		MC_Power		
VAR	MC_Power1		MC_Power		
VAR	power0_en		BOOL		
VAR	power1_en		BOOL		
VAR	pwr0_statu		BOOL		
VAR	pwr1_statu		BOOL		
VAR	MC_AxesGroupAddAxis0		MC_AxesGroupAddAxis		
VAR	MC_AxesGroupAddAxis1		MC_AxesGroupAddAxis		
VAR	addaxis1		BOOL		
VAR	addaxis2		BOOL		
VAR	addaxis1_done		BOOL		
VAR	addaxis2_done		BOOL		
VAR	MC_AxesGroupEnable0		MC_AxesGroupEnable		
VAR	Groupp_en		BOOL		
VAR	dir_vel		ARRAY [1..8] OF LREAL	[8(50)]	
VAR	dir_acc		ARRAY [1..8] OF LREAL	[8(100)]	
VAR	dir_dec		ARRAY [1..8] OF LREAL	[8(100)]	
VAR	dir_jer		ARRAY [1..8] OF LREAL	[8(1000)]	
VAR	lin_vel		LREAL		
VAR	lin_acc		LREAL		
VAR	groupenable_statu		BOOL		
VAR	linabs_exe		BOOL		
VAR	MC_MoveLinearAbsolute0		MC_MoveLinearAbsolute		
VAR	lin_pos		ARRAY [1..8] OF LREAL	[100,200]	
VAR	linabs_done		BOOL		
VAR	linabs_busy		BOOL		
VAR	linabs_active		BOOL		
VAR	linabs_abort		BOOL		
VAR	linabs_error		BOOL		
VAR	linabs_errorid		WORD		
VAR	MC_MoveCircularAbsolute0		MC_MoveCircularAbsolute		
VAR	cirabs_exe		BOOL		
VAR	cirabs_auxpoint		ARRAY [1..2] OF LREAL	[400,300]	
VAR	cirabs_endpoint		ARRAY [1..8] OF LREAL	[500,1000]	
VAR	cirabs_cirmode		INT		
VAR	cirabs_pathchoice		INT		
VAR	cirabs_done		BOOL		
VAR	cirabs_busy		BOOL		
VAR	cirabs_active		BOOL		
VAR	cirabs_abort		BOOL		
VAR	cirabs_error		BOOL		
VAR	cirabs_errorid		WORD		

■ 梯形图(LD)



## ■ 结构化文本(ST)

```
MC_Power0(  
  Axis:=1 ,  
  Enable:= power0_en,  
  EnablePositive:= TRUE,  
  EnableNegative:= TRUE,  
  BufferMode:=0 ,  
  Status=> pwr0_statu  
);
```

```
MC_Power1(  
  Axis:=2 ,  
  Enable:= power1_en,  
  EnablePositive:= TRUE,  
  EnableNegative:= TRUE,  
  BufferMode:=0 ,  
  Status=>pwr0_statu  
);
```

```
MC_AxesGroupAddAxis0(  
  AxesGroup:=1 ,  
  Axis:=1 ,  
  Execute:= addaxis1 ,  
  IdentNum:=1 ,  
  Done=> addaxis1_done  
);
```

```
MC_AxesGroupAddAxis1(  
  AxesGroup:=1 ,  
  Axis:=2 ,  
  Execute:= addaxis2,  
  IdentNum:=2 ,  
  Done=> addaxis2_done  
);
```

```
MC_AxesGroupEnable0(  
  AxesGroup:=1 ,  
  Enable:= Groupp_en ,  
  DirectVelocity:= dir_vel,  
  DirectAcceleration:= dir_acc ,  
  DirectDeceleration:= dir_dec ,  
  DirectJerk:= dir_jer ,  
  LinearVelocity:= 50,  
  LinearAcceleration:=100 ,  
  LinearDeceleration:= 100,  
  LinearJerk:= 1000,  
  Status=> groupenable_statu  
);
```

```
MC_MoveLinearAbsolute0(  
  AxesGroup:= 1,  
  Execute:= groupenable_statu AND linabs_exe,  
  Position:= lin_pos ,
```

```
Velocity:= 50,  
Acceleration:=100 ,  
Deceleration:= 100,  
Jerk:= 1000,  
BufferMode:=mcBuffered ,  
TransitionMode:=mcTMNone ,  
Done=> linabs_done,  
Busy=> linabs_busy,  
Active=>linabs_active,  
CommandAborted=>linabs_abort,  
Error=>linabs_error,  
ErrorID=>linabs_errorid  
);
```

```
MC_MoveCircularAbsolute0(  
  AxesGroup:=1 ,  
  Execute:= groupenable_statu AND cirabs_exe ,  
  CircMode:= cirabs_cirmode,  
  AuxPoint:= cirabs_auxpoint ,  
  EndPoint:= cirabs_endpoint,  
  MultiTurn:= 0,  
  PathChoice:= cirabs_pathchoice ,  
  Velocity:= 50,  
  Acceleration:=100 ,  
  Deceleration:= 100,  
  Jerk:= 1000,  
  BufferMode:=mcBuffered ,  
  TransitionMode:=mcTMNone ,  
  Done=>cirabs_done,  
  Busy=>cirabs_busy,  
  Active=>cirabs_active,  
  CommandAborted=>cirabs_abort,  
  Error=>cirabs_error,  
  ErrorID=>cirabs_errorid  
);
```



## 第八章 CNC介绍



## 8.1 CNC简介

M500系列控制器（不包括M500S系列），作为高阶的运动控制器，支持基本的CNC功能，如支持G0、G1、G2/G3、G4、G90/G91、G50/G51/G52等；另外支持M代码和其它任务中程序变量进行逻辑处理，如通过CNC代码可以控制气缸动作等。

多条CNC代码执行时，控制器会预读多条CNC代码，其中一行CNC代码执行完毕后，速度可以不为0，继续执行下一行CNC代码，提高加工效率。此功能可以通过G51或者G52代码实现。

CNC代码可以在软件中由用户自行编辑，也可以由CAD相关软件转换为DXF后导入到软件内。CNC代码编辑后，软件中会显示出对应CNC代码的运动轨迹。软件中的CNC代码须下载到控制后才可以执行。

### ■ 支持 CNC 功能的机种

M500系列控制器（不包括M500S系列）支持CNC功能，其它系列不支持。

### ■ CNC 代码单独书写格式

CNC程序中包括G代码和M代码，对应的代码单独书写时的含义及输入格式如下表所示：

支持代码	代码名称	输入形式*	最多轴数
G0	快速定位	格式： N& G0 X& Y& Z& A& B& C& P& Q& 范例： N0 G0 X10.1 Y10 Z10 A10 B10 C10 P10 Q10	8轴
G1	直线插补	格式： N& G1 X& Y& Z& A& B& C& P& Q& E& E& F& 范例： N0 G1 X10.1 Y10 Z10 A10 B10 C10 P10 Q10 E100 E-100 F10	8轴
G2	顺时针圆弧插补、螺旋插补	格式1： N& G2 X& Y& Z& A& B& C& P& Q& I& J& T& E& E& F& 范例： N0 G2 X500 Y1000 I400 J300 E100 E-100 F10 格式2： N& G2 X& Y& Z& A& B& C& P& Q& R& T& E& E& F& 范例： N0 G2 X500 Y1000 R500 E100 E-100 F10	8轴
G3	逆时针圆弧插补、螺旋插补	格式1： N& G3 X& Y& Z& A& B& C& P& Q& I& J& T& E& E& F& 范例： N0 G3 X100 Y200 I0 J-500 E100 E-100 F10 格式2： N& G3 X& Y& Z& A& B& C& P& Q& R& T& E& E& F& 范例： N0 G3 X100 Y200 R500 E100 E-100 F10	8轴
G4	时间延时	格式： N& G4 K& 范例： N0 G4 K2	和轴数量 无直接关 系
G17	设定为XY平面	格式： N& G17 范例： N0 G17	
G18	设定为ZX平面	格式： N& G18 范例： N0 G18	
G19	设定为YZ平面	格式： N& G19 范例： N0 G19	
G90	绝对值模式	格式： N& G90 范例： N0 G90	
G91	相对值模式	格式： N& G91 范例： N0 G91	
G50	精确停止	格式： N& G50 范例： N0 G50	
G51	圆弧交接	格式： N& G51 范例： N0 G51	
G52	圆滑交接	格式： N& G52 范例： N0 G52	
M0~M99	M代码	格式1： N& M& 范例： N0 M1 格式2： N& M& D& 范例： N0 M1 D6.2	

\*&表示G代码中对应的参数值，参数值的单位请参考具体G代码的说明

## ■ CNC 代码组合书写格式

G代码	G代码/M代码	输入形式*
G17/G18/G19	G2/G3	范例: N0 G17 G2
G0/G1/G2/G3/G4	M0~M99	范例: N0 G1 X10.1 Y10 M10 D12.5

\*1: G17/G18/G19和G2/G3代码组合使用时, G17/G18/G19要放同一行的在前面。

\*2: G代码和M代码组合使用时, M代码要放在同一行的后面。

## ■ CNC 代码可以不输入的参数

- G0 代码中 X& Y& Z& A& B& C& P& Q& 中的一个或多个可以不输入, 没有输入的参数不执行。
- G1 代码中 G1 X& Y& Z& A& B& C& P& Q& E& E& F& 中的一个或多个可以不输入, 没有输入的参数不执行(E、F 采取缺省值)。
- G2、G3 代码后 X& Y& Z& A& B& C& P& Q& E& E& F& 中的一个或多个可以不输入, I&、J&、或者 R&必须输入。如指定为 XY 平面时, X 和 Y 对应的参数必须要有; 如指定为 XZ 平面时, X 和 Z 对应的参数必须要有; 如指定为 YZ 平面时, Y 和 Z 对应的参数必须要有。
- M 代码中 D 参数可以不输入。

## ■ CNC 代码参数使用装置表示方法

- 如 G 代码中的相关参数不是固定值, 需要由外部指定时, 对应的参数可以使用装置表示, 表示方法为 ML 装置编号的两边为\$符号, ML 装置编号对应的%不用输入。范例: N00 G1 X\$ML10\$ Y\$ML11\$ Z\$ML12\$, 如%ML10、%ML11、%ML12 的分别为值为 100、200、300 时, 上述使用装置的 G 代码和 N00 G1 X100 Y200 Z300 的执行效果相同。
- X、Y、Z、A、B、C、P、Q、E、F、I、J、R、T、K 后的参数可以使用%ML 装置, 使用装置时, 注意对应参数的数据类型, 每个参数对应的数据类型可以在【G 代码功能详细介绍】→【参数说明】中查看, 可以在软件中定义相应的数据类型的变量, 分配到中指定 CNC 代码中使用的装置。

## ■ CNC 代码默认值

- 默认绝对值模式, CNC 代码中没有 G90 和 G91 时, 默认为绝对值模式。CNC 代码中的位置为绝对值模式或者相对值模式可以通过执行 G90 和 G91 切换。
- 默认圆弧插补平面: CNC 代码中没有 G17/G18/G19 时, 圆弧插补默认平面为 XY 平面。可通过 G17/G18/G19 切换平面。
- CNC 代码之间的缓存模式默认: 默认为等待模式。可通过 G50/G51/G52 切换缓存模式。
- G0 代码中相关默认值: G0 中各个轴的速度、加速度、跃度默认值为 1000, 单位为行程单位。上述参数值可以通过 MC\_SetMoveDirectParm 指令设置。
- G1、G2、G3 代码中相关默认值: 相关代码的速度、加速度、减速度、跃度默认值为 1000, 单位为行程单位。上述参数默认值可以通过 MC\_SetMoveLinearParm 指令设置。相关代码的速度、加速度、减速度可以通过 CNC 代码中 F、E 参数设置。

## ■ CNC 代码参数的继承

- 绝对模式和相对模式继承: 执行过 G90 后, 如果后面的 G 代码没有执行 G91, 则后面的 G 代码中的位置模式都为绝对模式, 即继承 G90 的执行; 如果执行 G90 后, 后面继续执行 G91, 则 G91 后面 G 代码中的位置模式都为相对模式, 即继承 G91 的执行。如下所示的 G 代码, N0~N6 之间的 G 代码中的位置为相对模式, N7 行后的 G 代码中的位置为绝对模式。
- 圆弧插补平面继承: 如下所示的 G 代码, N1 行有执行 G17, 则 N9 行的圆弧插补仍继承 N1 行的 G17 平面。
- CNC 代码之间的缓存模式继承: 如下所示的 G 代码, N2 行有执行 G50, 则其它两行 G 代码之间缓存模式继承 N2 行的 G50 缓存模式。
- G0、G1、G2、G3 代码中参数值继承: 相同 G 代码中的速度、加速度参数会继承。如下所示的 G 代码, N3 行 G0 有设置 E(加速度)和 F(速度), N4 行中的 G0 没有写 E 和 F 参数值, 但仍会继承 N3 行中设置的 E、F 参数值, 即 N4 行中 G0 对应的 E 和 F 参数值和 N3 行中的相同。同理, N6 行 G1 的 E 和 F 参数值和 N5 行的相同。N11 行 G2 的 E 和 F 参数值和 N9 行 G2 的参数相同。

N0 G91

N1 G17

```
N2 G50
N3 G0 X100 Y100 E100 E-100 F100
N4 G0 X200 Y200
N5 G1 X1000 Y1000 E100 E-100 F100
N6 G1 X2000 Y2000
N7 G90
N8 G1 X100 Y200
N9 G2 X500 Y1000 I400 J300 E100 E-100 F10
N10 G1 X100 Y200
N11 G2 X500 Y1000 I400 J300
```

## 8.2 G代码功能详细介绍

### 8.2.1 G90 (绝对值模式)

- **格式:**  
N& G90
- **书写范例:**  
N0 G90
- **参数说明:**  
格式中 N 后的&代表 CNC 代码中的行号。
- **代码功能说明:**  
设定 CNC 代码中的位置为绝对值模式（圆心坐标位置除外），CNC 代码中的位置表示以 0 位置为参考点的位置。该代码执行后，后续 CNC 代码中的位置为绝对值模式（圆心坐标位置除外），可以通过执行 G91 代码切换为相对值模式。

### 8.2.2 G91 (相对模式)

- **格式:**  
N& G91
- **书写范例:**  
N& G91
- **参数说明:**  
格式中 N 后的&代表 CNC 代码中的行号。
- **代码功能说明:**  
设定 CNC 代码中的位置为相对值模式，CNC 代码中的位置表示以执行 CNC 代码前的位置为参考点的相对值位置。该代码执行后，后续 CNC 代码中的位置为为相对值模式，可以通过执行 G90 代码切换为绝对值模式。

### 8.2.3 G0 (快速定位)

- **格式:**  
N& G0 X& Y& Z& A& B& C& P& Q&
- **书写范例:**  
N0 G0 X100.1 Y100.2 Z200.1 A100.5 B100.2 C100.2 P100.2 Q100.2

- **参数说明：**

G0 代码中相关参数后的数值含义如下表所示：

输入变量	数据类型	范围	缺省值	描述
N后的数值	UINT	0~65535	不可缺省	CNC代码中行号
X后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴X轴的目标位置。 (单位: 行程单位/秒 <sup>2</sup> ) *1
Y后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴Y轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> ) *1
Z后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴Z轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> ) *1
A后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴A轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> ) *1
B后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴B轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> ) *1
C后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴C轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> ) *1
P后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴P轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> ) *1
Q后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴Q轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> ) *1

\*1: 指令单位的详细介绍请参考“[运动控制指令参数单位](#)”章节。

- **代码功能说明：**

- 该代码用于控制多个轴单独运动，最多可以控制8个轴，所有轴同时启动，停止由CNC代码中各个轴指定的位置决定，各个轴指定的位置可以设定为绝对值模式或者相对值模式，通过G90和G91设定。各个轴的目标速度、加速度、减速度、跃度由MC\_SetMoveDirectParm指令的输入变量Velocity、Acceleration、Deceleration、Jerk设定。
- 该代码中的X、Y、Z、A、B、C、P、Q表示轴组中的逻辑轴，和MC\_AxesGroupAddAxis指令中输入变量IdentNum的值1~8对应。逻辑轴X、Y、Z、A、B、C、P、Q和软件中配置轴的对应关系由MC\_AxesGroupAddAxis指令中IdentNum的值和Axis的值决定。

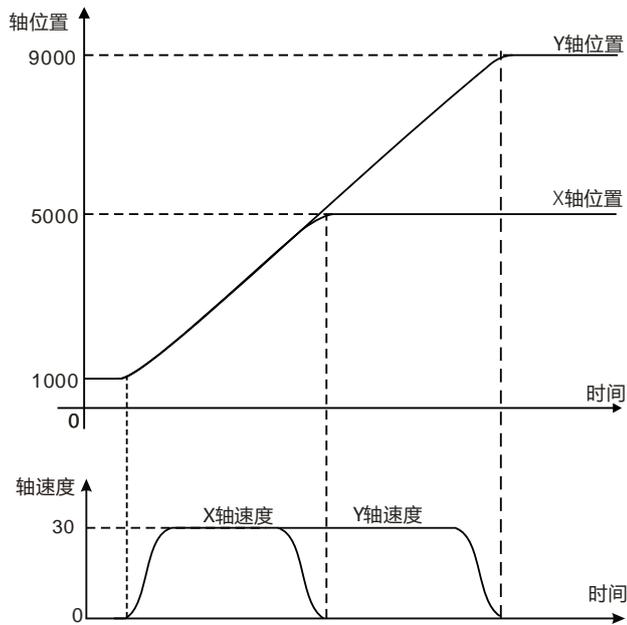
- **绝对值模式范例：**

- X、Y轴的初始位置都为1000，轴参数都为默认值。将要执行的G代码如下：

```
N0 G90
```

```
N1 G0 X5000 Y9000
```

- G代码执行后，X轴和Y轴对应的位置和速度曲线如下图所示：



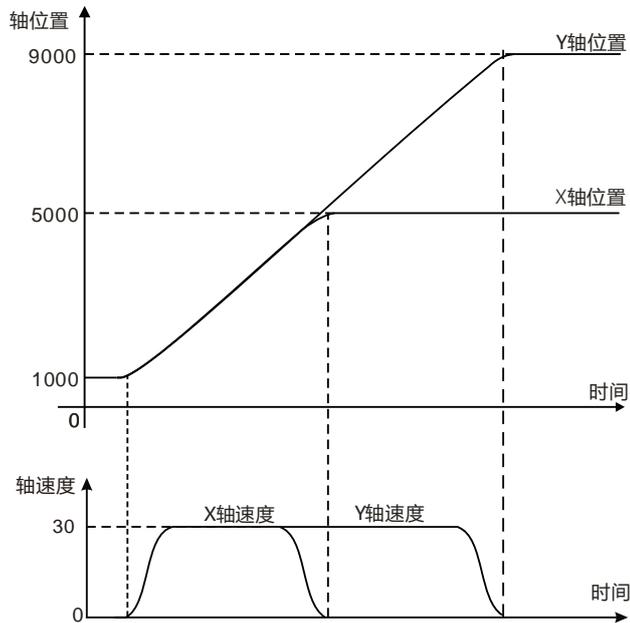
- **相对值模式范例:**

- X、Y轴的初始位置都为1000，轴参数都为默认值。将要执行的G代码如下:

```
N00 G91
```

```
N01 G0 X4000 Y8000
```

- G代码执行后，X轴和Y轴对应的位置和速度曲线如下图所示:



## 8.2.4 G1 (直线插补)

- **格式:**

```
N& G1 X& Y& Z& A& B& C& P& Q& E& E& F&
```

- **书写范例:**

```
N0 G1 X100.1 Y100.2 Z200.1 A100.5 B100.2 C100.2 P100.2 Q100.2 E10 E-20 F10
```

- **参数说明:**

输入变量	数据类型	范围	缺省值	描述
N后的数值	UINT	0~65535	不可缺省	CNC代码中行号
X后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴X轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> ) *2
Y后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴Y轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> ) *2
Z后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴Z轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> ) *2
A后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴A轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> ) *2
B后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴B轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> ) *2
C后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴C轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> ) *2
P后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴P轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> ) *2
Q后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴Q轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> ) *2
E后的数值	LREAL	正数	不可缺省	指定加速度*1 (单位: 行程单位/秒 <sup>2</sup> ) *2
E后的数值	LREAL	负数	不可缺省	指定减速度*1 (单位: 行程单位/秒 <sup>2</sup> ) *2
F后的数值	LREAL	正数	不可缺省	指定加速度*1 (单位: 行程单位/秒 <sup>2</sup> ) *2

\*1:

Velocity、Acceleration、Deceleration、Jerk的关系请参考“[运动控制指令相关参数说明](#)”章节。

\*2: 指令单位的详细介绍请参考“[运动控制指令参数单位](#)”章节。

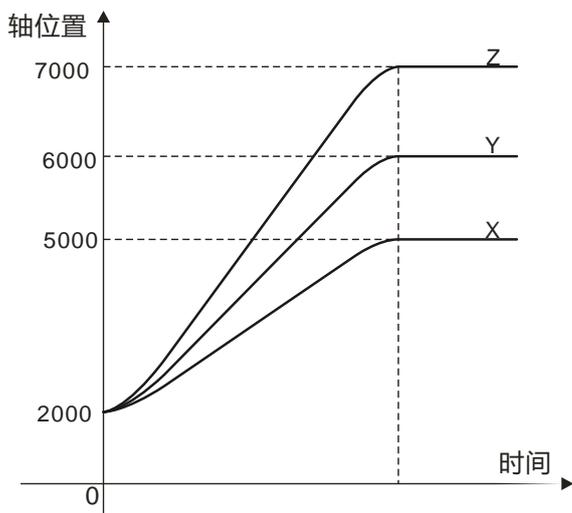
#### ● 代码功能说明:

- 该代码用于控制多个轴同时启动和停止, 进行直线插补动作, 最多可以8个轴同时动作。各个轴的目标位置通过G代码中的输入变量设定, 输入变量F为所有轴的合成速度, 输入变量E为所有轴的合成加速度和减速度, E后面的数值为正s数表示加速度, 为负数表示减速度。
- 该代码中的X、Y、Z、A、B、C、P、Q表示轴组中的逻辑轴, 和MC\_AxesGroupAddAxis指令中输入变量IdentNum的值1~8对应。逻辑轴X、Y、Z、A、B、C、P、Q和软件中配置轴的对应关系由MC\_AxesGroupAddAxis指令中IdentNum的值和Axis的值决定。
- 输入变量F为所有轴的合成速度, 合成速度的平方为代码中各轴速度的平方和。
- 该代码使用时, 需要控制哪些轴, 输入X、Y、Z、A、B、C、P、Q中的对应轴即可, 不需要控制的可以不写。

#### ● 绝对值模式范例:

- X、Y、Z轴的初始位置都为2000, 轴参数都为默认值。将要执行的G代码如下:
 

```
N00 G90
N01 G1 X5000 Y6000 Z7000
```
- G代码执行后, X、Y、Z轴位置 and 时间的对应关系如下所示:



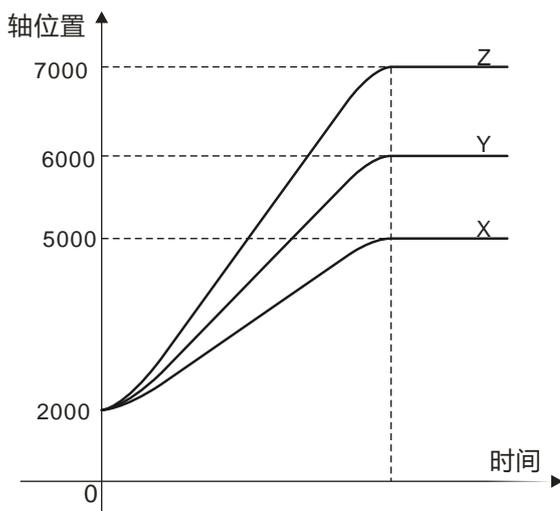
- **相对值模式范例：**

- X、Y、Z轴的初始位置都为2000，轴参数都为默认值。将要执行的G代码如下：

```
N00 G90
```

```
N01 G1 X3000 Y4000 Z5000
```

- G代码执行后，X、Y、Z轴位置和时间对应关系如下所示：



## 8.2.5 G2 (顺时针圆弧/螺旋插补)

- **格式：**

格式 1: N& G2 X& Y& Z& A& B& C& P& Q& E& E& F& I& J& T&

格式 2: N& G2 X& Y& Z& A& B& C& P& Q& E& E& F& R& T&

- **书写范例：**

格式 1 范例: N0 G2 X500 Y1000 I400 J300 E100 E100 F10

格式 1 说明: XY 平面指定圆心坐标的圆弧插补。

格式 2 范例: N2 G2 X500 Y1000 R500 E100 E100 F10

格式 2 说明: XY 平面指定半径的圆弧插补。

- **参数说明：**

输入变量	数据类型	范围	缺省值	描述
N后的数值	UINT	0~65535	不可缺省	CNC代码中行号
X后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴X轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> ) *2
Y后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴Y轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> ) *2
Z后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴Z轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> ) *2
A后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴A轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> ) *2
B后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴B轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> ) *2
C后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴C轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> ) *2
P后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴P轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> ) *2
Q后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴Q轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> ) *2
I/I/J后的数值	LREAL	正数、负数	0	圆心坐标 XY平面X方向相对于起点的相对值 XZ平面X方向相对于起点的相对值 YZ平面Y方向相对于起点的相对值 详细请参考代码功能说明→圆心坐标说明部分
J/K/K后的数值	LREAL	正数、负数	0	圆心坐标 XY平面Y方向相对于起点的相对值 XZ平面Z方向相对于起点的相对值 YZ平面Z方向相对于起点的相对值 详细请参考代码功能说明→圆心坐标说明部分
R后的数值	LREAL	正数、负数、0	0	圆弧的半径
T后的数值	ULINT	正数	0	圆的圈数
E后的数值	LREAL	正数	不可缺省	指定加速度*1 (单位: 行程单位/秒 <sup>2</sup> ) *2
E后的数值	LREAL	负数	不可缺省	指定减速度*1 (单位: 行程单位/秒 <sup>2</sup> ) *2
F后的数值	LREAL	正数	不可缺省	指定目标速度*1 (单位: 行程单位/秒) *2

\*1: Velocity、Acceleration、Deceleration、Jerk的关系请参考“[运动控制指令相关参数说明](#)”章节。

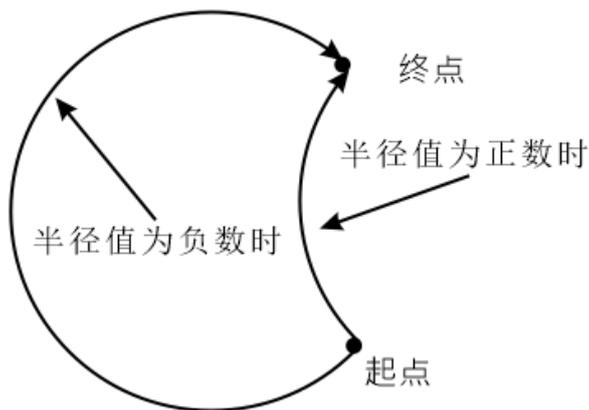
\*2: 指令单位的详细介绍请参考“[运动控制指令参数单位](#)”章节。

#### ● 代码功能说明：

- 该指令用于控制多个轴同时启动和停止，两个轴进行顺时针圆弧插补，其它轴跟随圆弧运动同时启动和停止。如X、Y、Z三个轴同时动作时，则为螺旋插补。
- 顺时针圆弧插补的两个轴为X、Y、Z中的两个轴，具体哪两个轴进行圆弧插补由指定的平面决定，默认为XY平面。如通过G17指定为XY平面时，则X轴和Y轴进行圆弧插补；通过G18指定为XZ平面时，则X轴和Z轴进行圆弧插补；通过G19指定为YZ平面时，则Y轴和Z轴进行圆弧插补。
- T为圆的圈数，T为1时，圆弧的起点和终点相同时，该代码的运行轨迹为一个整圆；圆弧的起点和终点不同时，该代码的运行轨迹为圆弧。T的值大于1时，该代码的运行轨迹为圆弧和T指定的圆的圈数。
- 该代码中的X、Y、Z、A、B、C、P、Q表示轴组中的逻辑轴，和MC\_AxesGroupAddAxis指令中输入变量IdentNum的值1~8对应。逻辑轴X、Y、Z、A、B、C、P、Q和软件中配置轴的对应关系由

MC\_AxesGroupAddAxis指令中IdentNum的值和Axis的值决定。

- 输入变量F为所有轴的合成速度，合成速度的平方为代码中各轴速度的平方和。
- 该代码使用时，E（加速度）和F（目标速度）可以省略不写，不写时，会继承前面G代码中E（加速度）和F（目标速度）的设定值，如果所有G代码都没有指定E和F，可以通过MC\_SetMovelinearParm指令设定使用的E（加速度）和F（目标速度）。
- 该代码使用时，需要控制哪些轴，输入X、Y、Z、A、B、C、P、Q中的对应轴即可，不需要控制的可以不写。
- 指定半径进行圆弧插补时，R（半径）参数后的值为正数时，该代码的运行轨迹为较短的的圆弧；R（半径）参数后的值为负数时，该代码的运行轨迹为较长的的圆弧。示意图如下图所示。



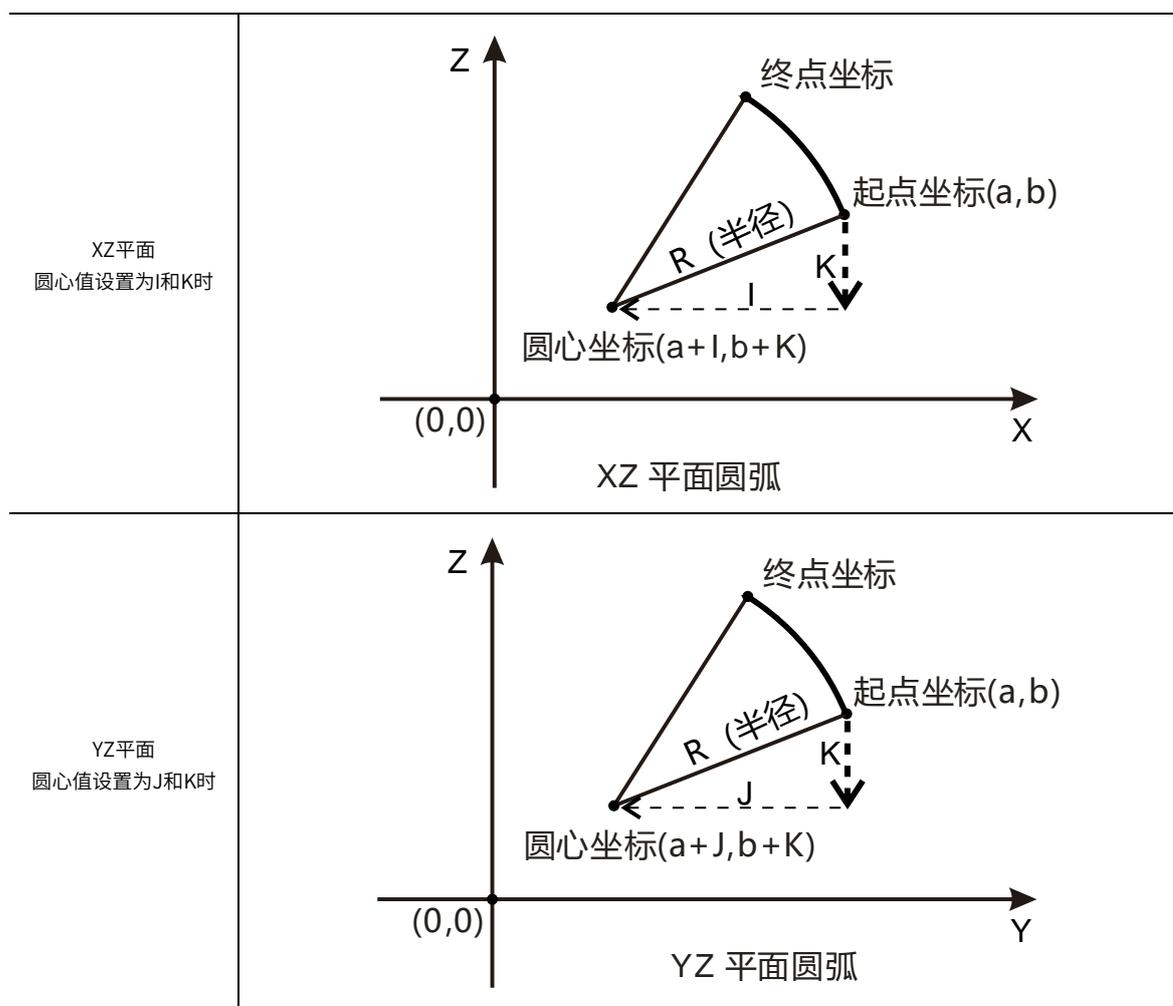
➤ 圆心坐标说明

指定圆心进行圆弧插补时，圆心坐标表示相对于指定平面中两个方向相对于圆弧起点坐标的相对值。无论是相对值模式（通过G91设定），还是绝对值模式（通过G90设定），圆心坐标都是相对于圆弧起点的相对值。

选择不同的平面时，圆心值参数不同，平面可以通过G17、G18、G19选择。XY平面时，圆心值使用I和J表示；XZ平面时，圆心值使用I和K表示；YZ平面时，圆心值使用J和K表示。

下表所示为不同平面时，圆心坐标和起点坐标的关系如下图所示。

平面	圆心坐标和起点坐标的关系
XY平面 圆心值设置为I和J时	

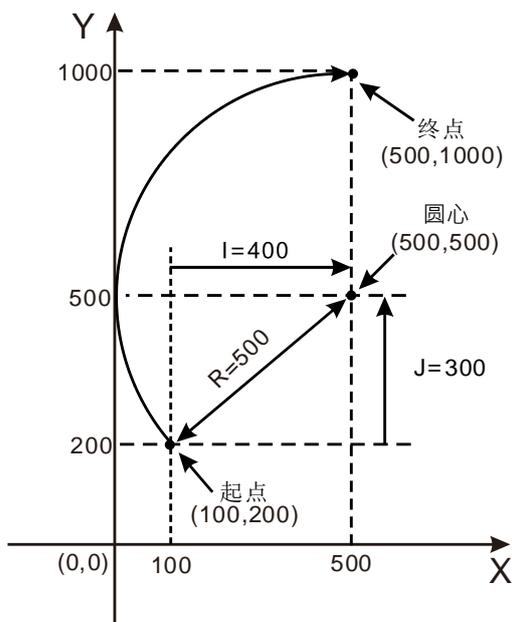


### ■ 范例 1: 绝对模式下指定圆心或者指定半径时的圆弧插补

- X轴和Y轴的当前位置为100和200，即下图中圆弧的的起点位置(100,200)，圆弧的终点位置为(500,1000)，即下图中的圆弧的终点，圆心位置为(500,500)，但程序中书写的圆心位置为(400,300)，因为圆心位置为相对于起点的相对值。绝对模式下，圆弧起点位置和终点位置为绝对位置，圆心位置为相对于起点的相对值。
- 指定圆心时的G代码书写方法
 

```
N0 G90
N1 G17
N2 G2 X500 Y1000 I400 J300 E100 E-100 F10
```
- 指定半径时的G代码书写方法
 

```
N0 G90
N1 G17
N2 G2 X500 Y1000 R500 E100 E-100 F10
```
- 上面G代码执行后，XY平面对应的圆弧轨迹如下图所示。



### ■ 范例 2：相对模式下指定圆心或者指定半径时的圆弧插补

- X轴和Y轴的当前位置为100和200，即下图中圆弧的的起点位置(100,200)，圆弧的终点位置为(500,1000)，即下图中的终点位置，但程序中书写的终点位置为(400,800)，因为当前模式为相对模式，程序中书写的终点位置为相对于起点的相对位置。程序中书写的圆心位置为(400,300)，圆心坐标也是相对于起点的相对值。

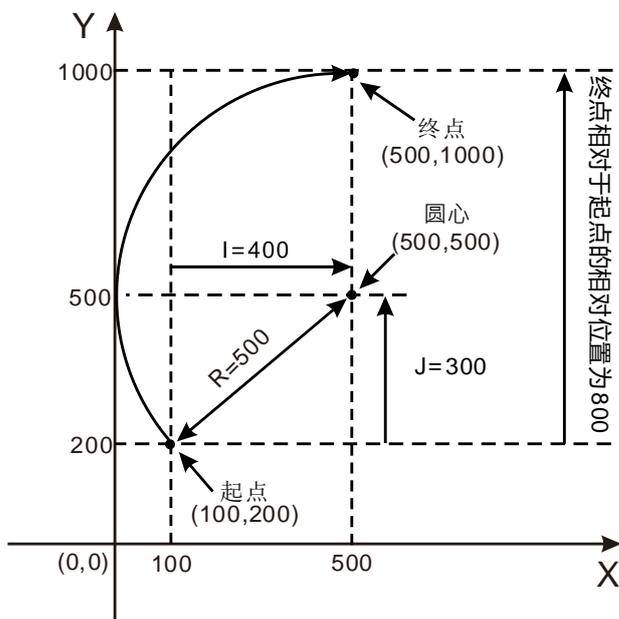
#### ● 指定圆心时的 G 代码书写方法

```
N0 G91
N1 G17
N2 G2 X400 Y800 I400 J300 E100 E-100 F10
```

#### ● 指定半径时的 G 代码书写方法

```
N0 G91
N1 G17
N2 G2 X400 Y800 R500 E100 E-100 F10
```

- 上面 G 代码执行后，XY 平面对应的圆弧轨迹如下图所示。



### ■ 范例 3: 绝对模式下指定圆心时的整圆插补:

- 轴组中 X 轴和 Y 轴的当前位置为 100 和 200, 即下图中圆弧的起点位置(100,200), 圆弧的终点位置和起点位置为同一位置, 即下图中的圆弧的终点, 圆心位置为 (400,300)。该种模式下, 圆弧起点位置和终点位置为绝对位置, 圆心位置为相对于起点的相对值。

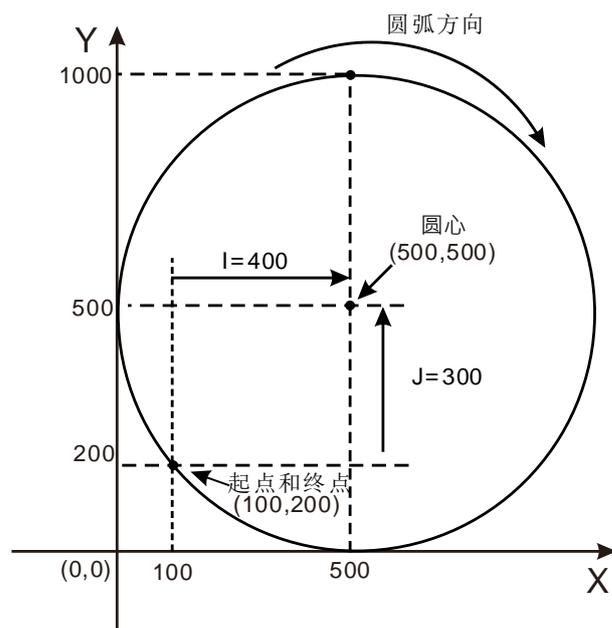
- 指定圆心时的 G 代码书写方法

N0 G90

N1 G17

N2 G2 X100 Y200 I400 J300 E100 E100 F10

- 上面 G 代码执行后, XY 平面对应的圆弧轨迹如下图所示。



## 8.2.6 G3 (逆时针圆弧/螺旋插补)

- 格式:

格式 1: N& G3 X& Y& Z& A& B& C& P& Q& E& E& F& I& J& T&

格式 2: N& G3 X& Y& Z& A& B& C& P& Q& E& E& F& R& T&

- 书写范例:

格式 1 范例: N0 G3 X100 Y200 I0 J-500 E100 E100 F10

格式 1 说明: XY 平面指定圆心坐标的圆弧插补。

格式 2 范例: N2 G3 X100 Y200 R500 E100 E100 F10

格式 2 说明: XY 平面指定半径的圆弧插补。

- 参数说明:

格式中的&代表相关参数的数值, 代码中相关参数后的数值含义如下表所示:

输入变量	数据类型	范围	缺省值	描述
N后的数值	UINT	0~65535	不可缺省	CNC代码中行号。
X后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴X轴的目标位置。 (单位: 行程单位/秒 <sup>2</sup> )。*2
Y后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴Y轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> )。*2

Z后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴Z轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> )。*2
A后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴A轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> )。*2
B后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴B轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> )。*2
C后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴C轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> )。*2
P后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴P轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> )。*2
Q后的数值	LREAL	正数、负数、0	0	轴组中逻辑轴Q轴的目标位置 (单位: 行程单位/秒 <sup>2</sup> )。*2
I/I/J后的数值	LREAL	正数、负数	0	圆心坐标。 XY平面X方向相对于起点的相对值。 XZ平面X方向相对于起点的相对值。 YZ平面Y方向相对于起点的相对值。 详细请参考代码功能说明→圆心坐标说明部分。
J/K/K后的数值	LREAL	正数、负数	0	圆心坐标。 XY平面Y方向相对于起点的相对值。 XZ平面Z方向相对于起点的相对值。 YZ平面Z方向相对于起点的相对值。 详细请参考代码功能说明→圆心坐标说明部分。
R后的数值	LREAL	正数、负数、0	0	圆弧的半径。
T后的数值	ULINT	正数	0	圆的圈数。
E后的数值	LREAL	正数	不可缺省	指定加速度。*1 (单位: 行程单位/秒 <sup>2</sup> )。*2
E后的数值	LREAL	正数、负数	不可缺省	指定减速度。*1 (单位: 行程单位/秒 <sup>2</sup> )。*2
F后的数值	LREAL	正数	不可缺省	指定目标速度。*1 (单位: 行程单位/秒)。*2

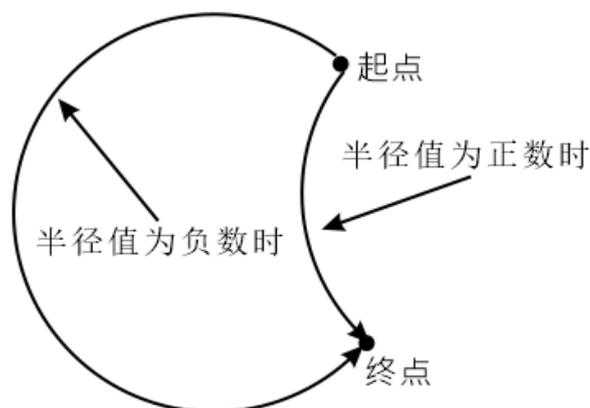
\*1: Velocity、Acceleration、Deceleration、Jerk的关系请参考“[运动控制指令相关参数说明](#)”章节。

\*2: 指令单位的详细介绍请参考“[运动控制指令参数单位](#)”章节。

#### ● 代码功能说明:

- 该指令用于控制多个轴同时启动和停止，两个轴进行逆时针圆弧插补，其它轴跟随圆弧运动同时启动和停止。如X、Y、Z三个轴同时动作时，则为螺旋插补。
- 逆时针圆弧插补的两个轴为X、Y、Z中的两个轴，具体哪两个轴进行圆弧插补由指定的平面决定，默认为XY平面。如通过G17指定为XY平面时，则X轴和Y轴进行圆弧插补；通过G18指定为XZ平面时，则X轴和Z轴进行圆弧插补；通过G19指定为YZ平面时，则Y轴和Z轴进行圆弧插补。
- T为圆的的圈数，T为1时，圆弧的起点和终点相同时，该代码的运行轨迹为一个整圆；圆弧的起点和终点不同时，该代码的运行轨迹为圆弧。T为大于1时，该代码的运行轨迹为圆弧和T指定的圆的圈数。
- 指定圆心进行圆弧插补时，圆心坐标表示相对于指定平面中两个方向相对于圆弧起点坐标的相对值。无论是相对值模式（通过G91设定），还是绝对值模式（通过G90设定），圆心坐标都是相对于圆弧起点的相对值。
- 该代码中的X、Y、Z、A、B、C、P、Q表示轴组中的逻辑轴，和MC\_AxesGroupAddAxis指令中输入变量IdentNum的值1~8对应。逻辑轴X、Y、Z、A、B、C、P、Q和软件中配置轴的对应关系由MC\_AxesGroupAddAxis指令中IdentNum的值和Axis的值决定。
- 输入变量F为所有轴的合成速度，合成速度的平方和为代码中各轴速度的平方和。

- 该代码使用时，E（加速度）和F（目标速度）可以省略不写，不写时，会继承前面G代码中E（加速度）和F（目标速度）的设定值，如果所有G代码都没有指定E和F，可以通过MC\_SetMovelinearParm指令设定使用的E（加速度）和F（目标速度）。
- 该代码使用时，需要控制哪些轴，输入X、Y、Z、A、B、C、P、Q中的对应轴即可，不需要控制的轴可以不写
- 指定半径进行圆弧插补时，R（半径）参数后的值为正数时，该代码的运行轨迹为较短的的圆弧；R（半径）参数后的值为负数时，该代码的运行轨迹为较长的的圆弧。示意图如下图所示。



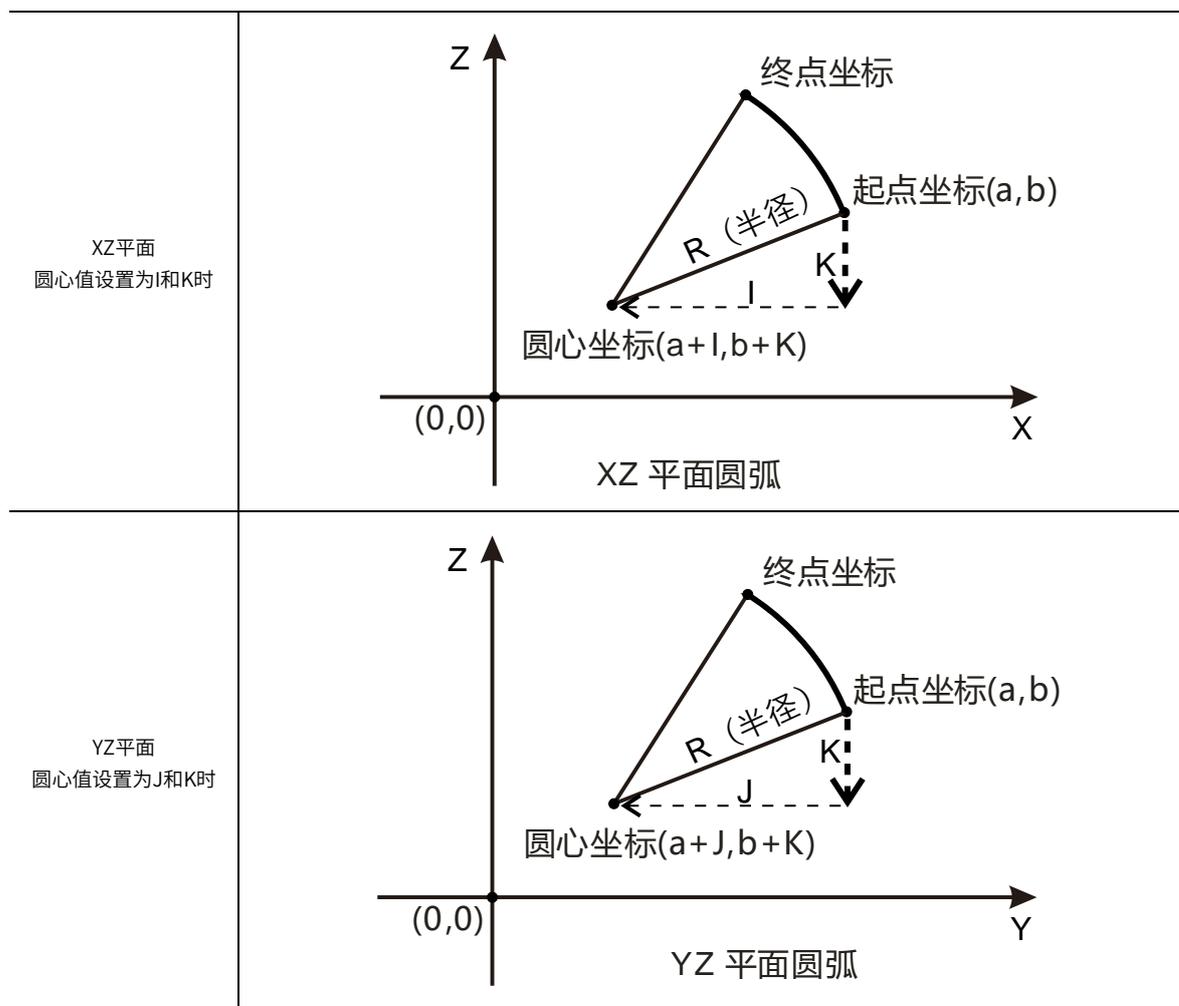
#### ➤ 圆心坐标说明

指定圆心进行圆弧插补时，圆心坐标表示相对于指定平面中两个方向相对于圆弧起点坐标的相对值。无论是相对值模式（通过G91设定），还是绝对值模式（通过G90设定），圆心坐标都是相对于圆弧起点的相对值。

选择不同的平面时，圆心值参数不同，平面可以通过G17、G18、G19选择。XY平面时，圆心值使用I和J表示；XZ平面时，圆心值使用I和K表示；YZ平面时，圆心值使用J和K表示。

下表所示为不同平面时，圆心坐标和起点坐标的关系如下图所示。

平面	圆心坐标和起点坐标的关系
XY平面 圆心值设置为I和J时	

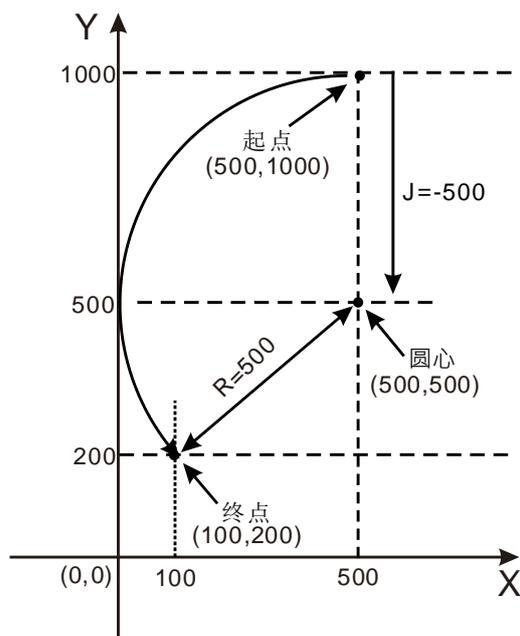


#### ■ 范例 1: 绝对模式下指定圆心或者指定半径时的圆弧插补

- X 轴和 Y 轴的当前位置为 500 和 1000，即下图中圆弧的的起点位置(500,1000)，圆弧的终点位置为 (100,200)，圆心位置为 (500,500)，但程序中书写的圆心位置为 (0,-500)，因为程序中书写的圆心位置为相对于起点的相对值。绝对模式下，圆弧起点位置和终点位置为绝对位置，圆心位置为相对于起点的相对值。
- 指定圆心时的 G 代码书写方法
 

```
N0 G91
N1 G17
N2 G3 X100 Y200 I0 J-500 E100 E100 F10
```
- 指定半径时的 G 代码书写方法
 

```
N0 G91
N1 G17
N2 G3 X100 Y200 R500 E100 E100 F10
```
- 上面 G 代码执行后，XY 平面对应的圆弧轨迹如下图所示。



### ■ 范例 2：相对模式下指定圆心或者指定半径时的圆弧插补

- X轴和Y轴的当前位置为500和1000，即下图中圆弧的起点位置(500,1000)，圆弧的终点位置为(100,200)，圆心位置为(500,500)，但程序中书写的终点位置为(-400,-800)，圆心位置为(0,-500)，相对模式下，程序中书写的圆弧终点位置和圆心位置为相对于起点的相对位置。

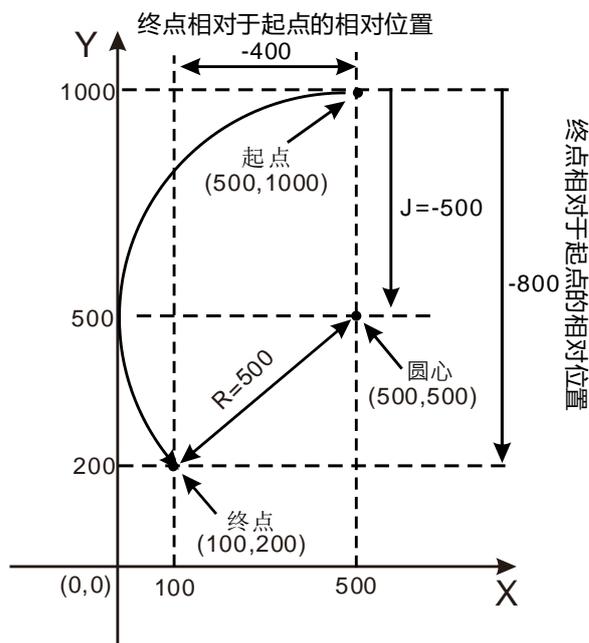
#### ● 指定圆心时的 G 代码书写方法

```
N0 G90
N1 G17
N2 G3 X-400 Y-800 I0 J-500 E100 E100 F10
```

#### ● 指定半径时的 G 代码书写方法

```
N0 G90
N1 G17
N2 G3 X-400 Y-800 R500 E100 E100 F10
```

- 上面 G 代码执行后，XY 平面对应的圆弧轨迹如下图所示。



### ■ 范例 3: 绝对模式下指定圆心的整圆插补:

- X 轴和 Y 轴的当前位置为 100 和 200，即下图中圆弧的起点位置(100,200)，圆弧的终点位置和起点位置为同一位置，即圆弧的起点和终点位置都为(100,200)。圆心位置为 (500,500)，但程序中书写的圆心位置为 (400,300)，因为程序中书写的圆心位置为相对于起点的相对值。绝对模式下，圆弧起点位置和终点位置为绝对位置，圆心位置为相对于起点的相对值。

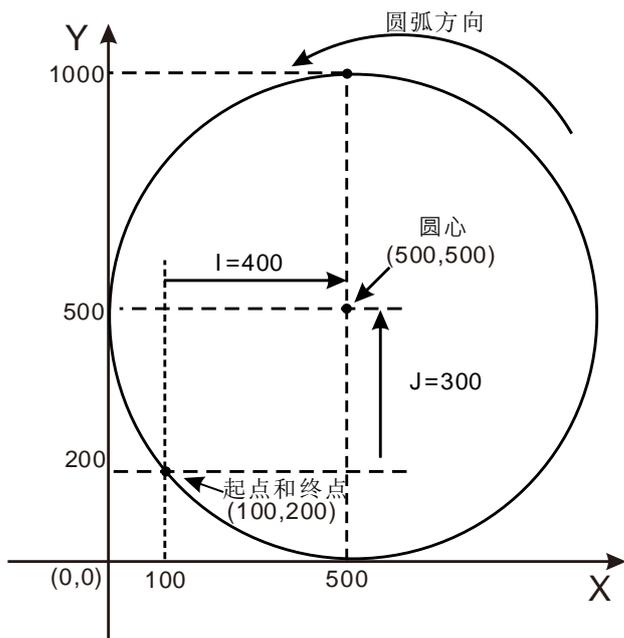
#### ● 指定圆心时的 G 代码书写方法

N0 G90

N1 G17

N3 G3 X100 Y200 I400 J300 E100 E100 F10

- 上面 G 代码执行后，XY 平面对应的圆弧轨迹如下图所示。



## 8.2.7 G17/G18/G19 (指定圆弧插补平面)

#### ● 格式:

N& G&

#### ● 书写范例:

格式 1 范例: N0 G17

格式 2 范例: N0 G18

格式 3 范例: N0 G19

#### ● 参数说明:

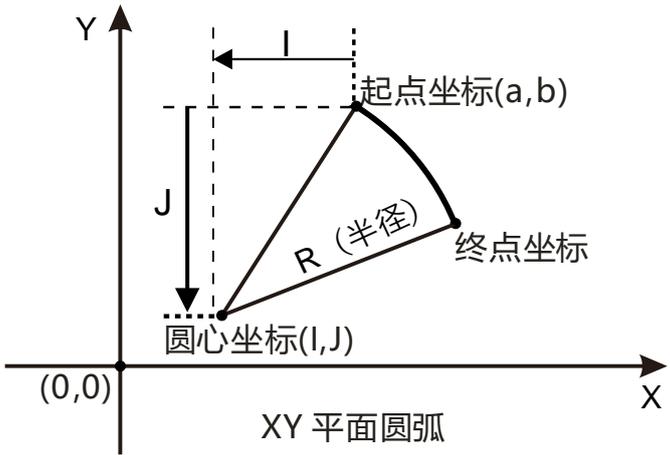
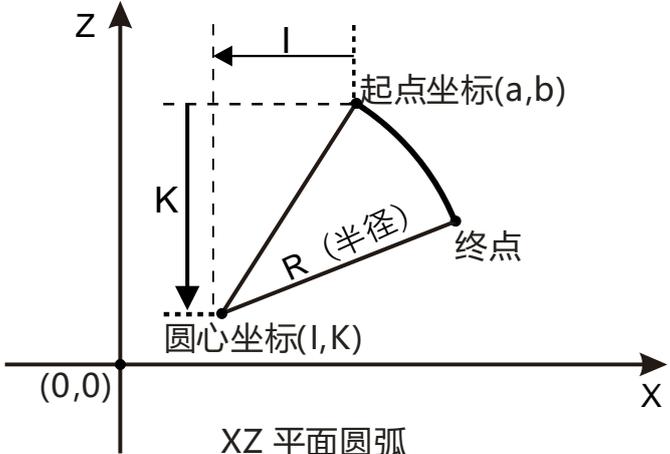
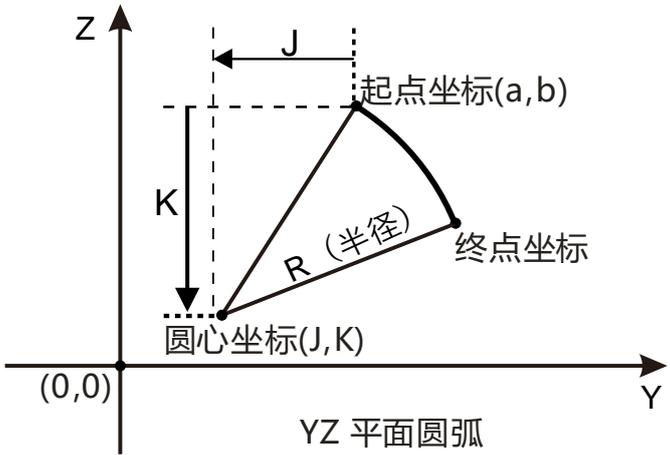
格式中 N 后的 & 代表 CNC 代码中的行号。

格式中 G 后的 & 代表对应的 G 代码，可以填写 17、18 或者 19

#### ● 代码功能说明:

G17、G18 或者 G19 用于设置圆弧操作的平面。G17 指定圆弧平面为 XY 平面，G18 指定圆弧平面为 XZ 平面，G19 指定圆弧平面为 YZ 平面。

不同平面示意图如下表所示:

平面	不同平面时圆心坐标的表示方法
XY平面 (G17) 圆心值使用I和J表示	 <p>XY 平面圆弧</p>
XZ平面 (G18) 圆心值使用I和K表示	 <p>XZ 平面圆弧</p>
YZ平面 (G19) 圆心值使用J和K表示	 <p>YZ 平面圆弧</p>

### 8.2.8 G4 (延时指令)

- 格式：  
N& G4 K&
- 书写范例：  
N0 G4 K2

- **参数说明:**

格式中的&代表相关参数的数值，代码中相关参数后的数值含义如下表所示：

输入变量	数据类型	范围	缺省值	描述
N后的数值	UINT	0~65535	不可缺省	CNC代码中行号。
K后的数值	LREAL	0.001~100000	0	设定的延时时间。 (单位:秒)

- **代码功能说明:**

该代码用于设定指定的延时时间。设定的延时时间到后再执行下一条G代码。

## 8.2.9 G50 (无过渡曲线模式)

- **格式:**

N& G50

- **书写范例:**

N0 G50

- **参数说明:**

格式中 N 后的&代表 CNC 代码中的行号。

- **代码功能说明:**

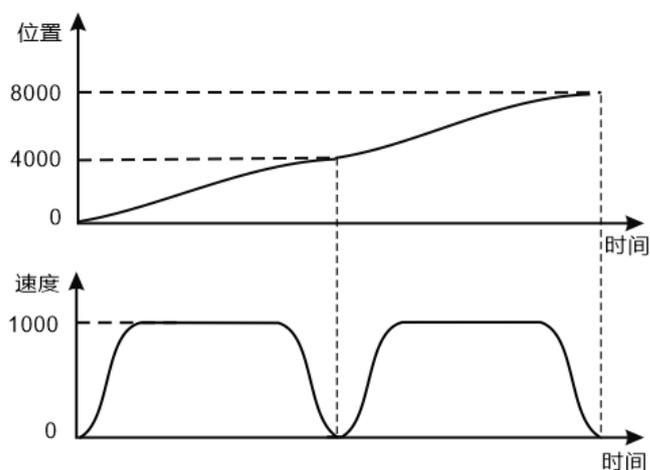
- 该代码用于设置CNC代码相邻两个代码之间的过渡模式，该代码的功能表示上一个CNC代码位置到达时，速度也同时降为0，然后再启动下一行CNC代码的执行。
- CNC代码两个代码之间的过渡模式默认为G50，可以通过G51和G52代码切换，所有G代码之间的过渡都可以使用G50。

- **功能范例:**

N00 G50

N01 G1 X4000 Y4000 E5000 E-5000 F1000

N02 G1 X8000 Y8000 E5000 E-5000 F1000



## 8.2.10 G51 (指定附加角过渡模式)

- **格式:**

N& G51 D&

- **书写范例:**

N0 G51 D100

- **参数说明:**

格式中的&代表相关参数的数值，代码中相关参数后的数值含义如下表所示：

输入变量	数据类型	范围	缺省值	描述
N后的数值	UINT	0~65535	不可缺省	CNC代码中行号
D后的数值	LREAL	正数、0	0	指定的过渡距离

- **代码功能说明:**

- 该代码用于设置CNC代码相邻两个代码之间的过渡模式，该代码的功能表示上一行CNC代码到达终点位置时，速度不降为0的情况下执行下一行CNC代码，上一行的终点位置和下一行CNC代码的起点位置之间为圆弧过渡，过渡时插补速度不降为0。
- 仅执行插补运动G代码时可以指定过渡模式为G51，其它G代码不可以指定为G51。CNC代码两个代码之间的过渡可以通过G50和G51代码切换。
- 相邻两个代码之间使用G51过渡模式时，过渡开始的起点为上一行G代码中的终点位置减去该代码中指定的过渡距离，过渡终点为上一行G代码中的终点位置加上该代码中指定的过渡距离。详细请参考范例中的说明。
- 相邻两个代码之间使用G51过渡模式时，G代码的速度以G51代码后第一条G代码中指定的合成速度为准，后续的G代码如果没有再使用G51，后续的G代码即使通过F设定合成速度，仍使用G51代码后第一条G代码中指定的合成速度。如果想G代码执行过程中变更速度，可以通过更改MC\_CoorMotion指令输入变量VelOverride的值更改合成速度。
- 多条CNC代码执行时，相邻两条CNC代码之间的过渡模式可以选择G51的情形如下表所示：

不同的G代码过渡
G1和G1之间过渡时
G1和G2或者G2和G1过渡时
G1和G3或者G3和G1过渡时
G2和G3或者G3和G2过渡时
G2和G2或者G3和G3过渡时

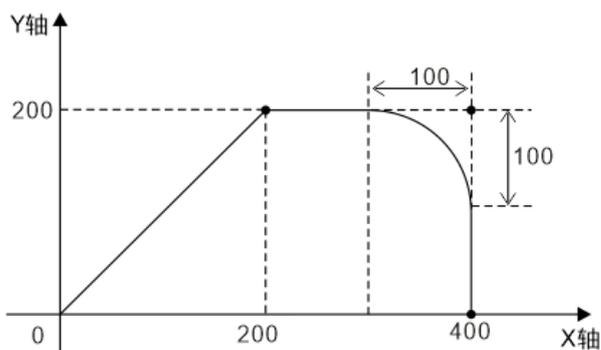
- **功能范例 1:**

N00 G1 X200 Y200

N01 G51 D100

N02 G1 X400 Y200

N03 G1 X400 Y0

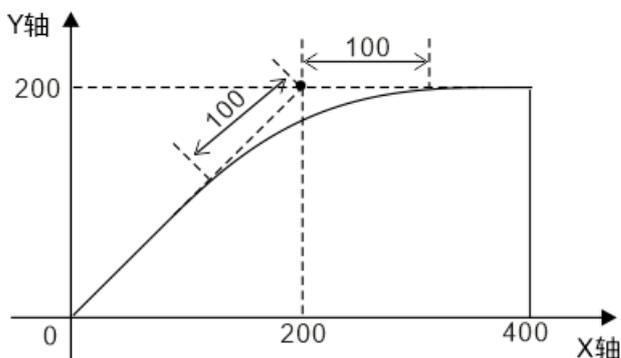


● 功能范例 2:

N00 G51 D100

N01 G1 X200 Y200

N02 G1 X400 Y200



### 8.2.11 G52 (以恒定的速度过渡模式)

● 格式:

N& G52

● 书写范例:

N0 G52

● 参数说明:

格式中 N 后的 & 代表 CNC 代码中的行号。

● 代码功能说明:

- 该代码用于设置 CNC 代码相邻两个代码之间的过渡模式，该代码的功能表示上一行 CNC 代码到达终点位置时，速度不降为 0 的情况下执行下一行 CNC 代码，以设定的速度（上一行 CNC 代码的合成速度）进行过渡。
- 仅执行 G1、G2、G3 代码时可以指定过渡模式为 G52，其它 G 代码不可以指定为 G52。CNC 代码两个代码之间的过渡可以通过 G51 和 G52 代码切换。
- 该模式一般应用于多个 CNC 代码之间的增量位置小，通过多行 CNC 代码位置拟合曲线的情况下使用。
- 相邻两个代码之间使用 G52 过渡模式时，G 代码的速度以 G52 代码后第一条 G 代码中指定的合成速度为准，后续的 G 代码如果没有再使用 G52，后续的 G 代码即使通过 F 设定合成速度，仍使用 G52 代码后第一条 G 代码中指定的合成速度。如果想 G 代码执行过程中变更速度，可以通过更改 MC\_CoorMotion 指令输入变量 VelOverride 的值更改合成速度。
- 多条 CNC 代码执行时，相邻两条 CNC 代码之间的过渡模式可以选择 G52 的情形如下表所示：

不同的G代码过渡	条件
G1和G1之间过渡时	可以执行
G1和G2或者G2和G1过渡时	直线和圆弧相切时才可以执行
G1和G3或者G3和G1过渡时	直线和圆弧相切时才可以执行
G2和G3或者G3和G2过渡时	圆弧和圆弧相切时才可以执行
G2和G2或者G3和G3过渡时	圆弧和圆弧相切时才可以执行

## 8.2.12 M代码

- **格式:**

N& M& D&

- **书写范例:**

书写范例 1: N0 M0 D123.12

书写范例 2: N1 G1 X100.5 Y100.5 E100 E-100 F30.5 M0 D123.12

- **参数说明:**

格式中 N 后的&代表 CNC 代码中的行号:

输入变量	数据类型	范围	缺省值	描述
N后的数值	UINT	0~65535	不可缺省	CNC代码中行号。
M后的数值	USINT	0~99	0	M代码的编号
D后的数值	LREAL	正数、负数、0	0	指定M代码执行后对应的数值

- **代码功能说明:**

- M代码一般用于G代码和其它任务中程序变量进行交互。如可以通过M代码触发控制器本体输出点开或者关（本体输出点开和关可以控制气缸动作等），也可以通过M代码判断G代码执行到何处。
- M代码和M代码的D参数并无具体含义，用户在使用时，可以对其赋予具体的含义。使用M代码时，执行到M代码所在行时，对应的M代码为TRUE，对应的D参数会赋值为M代码中的指定值。M代码的状态和D参数的值可以通过MC\_GetMCodeStatus指令读取。M代码的状态需要通过MC\_ResetMCode指令复位，M代码的值变为TRUE后，不会自动变为FALSE。
- M代码在CNC代码中可以单独一行，也可以和其它G代码在同一行。
- M代码和G代码在同一行时，M代码要放在G代码的后面。当执行到M代码所在的行时(该行对应的G代码开始执行时)，M代码对应的状态为TRUE，D参数值为M代码指定的值。该行G代码执行完毕后，继续执行下一行CNC代码。详细参考范例1。
- M代码单独一行时，当执行到M代码所在的行时(该行对应的M代码开始执行时)，M代码对应的状态为TRUE，D参数值为M代码指定的值。该行M代码后的CNC代码停止执行，需要复位该M代码后才可以继续执行，可以通过MC\_ResetMCode指令复位对应的M代码。详细参考范例2。

- **范例 1: M 代码和 G 代码在同一行范例**

- 对应的CNC代码

N1 G1 X100.5 Y100.5 E100 E-100 F30.5

N2 G1 X300.5 Y300.5 E100 E-100 F30.5 M0 D123.12

N3 G1 X500.5 Y500.5 E100 E-100 F30.5

- CNC代码执行到N2行（N2行刚开始执行）时，M0的值为TRUE，D参数的值为123.2。N2行G1代码执行完毕后，继续执行N3行的G代码。

● 范例 2: M 代码和 G 代码不在同一行范例

➤ 对应的CNC代码

N1 G1 X100.5 Y100.5 E100 E-100 F30.5

N2 M1 D123.12

N3 G1 X300.5 Y300.5 E100 E-100 F30.5

N4 G1 X500.5 Y500.5 E100 E-100 F30.5

- CNC代码执行到N2行时，M1的值为TRUE，D参数的值为123.12，N3和N4行的CNC代码不执行，M1通过MC\_ResetMCode指令复位后，继续执行N3和N4行的G代码。

## 8.3 MC\_SetMoveLinearParm (CNC插补运动默认参数设定)

该指令用于设置CNC插补运动时需要的默认运动参数。所属库: MotionControl\_Part2

适用机种: M500系列控制器 (不包括M500S系列)

指令	名称	FB/FUN	梯形图样式	ST样式
MC_SetMoveLinearParm	CNC插补运动默认参数设定	FB		<pre>MC_SetMoveLinearParm_Instance (   AxesGroup:= 参数 ,   Execute := 参数 ,   Velocity := 参数 ,   Acceleration:= 参数 ,   Deceleration := 参数 ,   Jerk := 参数 ,   Done =&gt; 参数 ,   Busy =&gt; 参数 ,   Error =&gt; 参数 ,   ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
AxesGroup	轴组编号	USINT	1~8	不可缺省	轴组编号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
Velocity	插补速度	LREAL	正数	不可缺省	插补速度
Acceleration	插补加速度	LREAL	正数	不可缺省	插补加速度
Deceleration	插补减速度	LREAL	正数	不可缺省	插补减速度
Jerk	插补跃度	LREAL	正数	不可缺省	插补跃度

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成后变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时, 输出错误代码的含义请参考“ <a href="#">指令错误代码描述</a> ”

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	指令执行完成后	<ul style="list-style-type: none"> <li>◆ Done为TRUE, Execute由TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE</li> </ul>
Busy	Execute 由FALSE变为TRUE时	<ul style="list-style-type: none"> <li>◆ Done变为TRUE时</li> <li>◆ Error变为TRUE时</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Execute由TRUE变为FALSE时</li> </ul>

## ■ 功能说明

### ● 基本功能说明

该指令用于设置 CNC 中 G1、G2、G3 中使用的默认速度、加速度、减速度、跃度参数，通过该指令输入变量 Velocity、Acceleration、Deceleration、Jerk 进行设置。G1、G2、G3 中没有通过 E 和 F 指定插补速度、加速度和减速度时，使用该指令中设定的插补速度、加速度和减速度。G1、G2、G3 中 Jerk（跃度）的值只能通过该指令设定。

### ● 指令完成时机

该指令的输出变量 Done 为 TRUE 表示参数设置完成。

### ● 异常排除

该指令执行时，如果指令输入变量值不在允许的范围时，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。

## 8.4 MC\_SetMoveDirectParm (CNC单独定位运动参数设定)

该指令用于设置CNC单独定位时需要的运动参数。所属库：MotionControl\_Part2

适用机种：M500系列控制器（不包括M500S系列）

指令	名称	FB/FUN	梯形图样式	ST样式
MC_SetMoveDirectParm	CNC单独定位运动参数设定	FB		<pre>MC_SetMoveDirectParm_Instance (   AxesGroup:= 参数 ,   Execute := 参数 ,   Velocity := 参数 ,   Acceleration:=参数 ,   Deceleration := 参数 ,   Jerk := 参数 ,   Done =&gt; 参数 ,   Busy =&gt; 参数 ,   Error =&gt; 参数 ,   ErrorID=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
AxesGroup	轴组编号	USINT	1~8	不可缺省	轴组编号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
Velocity	快速定位速度	ARRAY [1..8] OF LREAL	正数	不可缺省	快速定位速度
Acceleration	快速定位加速度	ARRAY [1..8] OF LREAL	正数	不可缺省	快速定位加速度
Deceleration	快速定位减速度	ARRAY [1..8] OF LREAL	正数	不可缺省	快速定位减速度
Jerk	快速定位跃度	ARRAY [1..8] OF LREAL	正数	不可缺省	快速定位跃度

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成后变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	指令执行完成后	<ul style="list-style-type: none"> <li>◆ Done为TRUE，Execute由TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时，Done变为TRUE一个周期后，变为FALSE</li> </ul>
Busy	Execute 由FALSE变为TRUE时	<ul style="list-style-type: none"> <li>◆ Done变为TRUE时</li> <li>◆ Error变为TRUE时</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Execute由TRUE变为FALSE时</li> </ul>

## ■ 功能说明

### ● 基本功能说明

该指令用于设置 CNC 中 G0 中使用的速度、加速度、减速度、跃度参数，通过该指令输入变量 Velocity、Acceleration、Deceleration、Jerk 进行设置。G0 中使用的位置在 G0 中设定。CNC 中使用 G0 时，先使用该指令设置各个轴需要的运动参数。

### ● 指令完成时机

该指令的输出变量 Done 为 TRUE 表示参数设置完成。

### ● 异常排除

该指令执行时，如果指令输入变量值不在允许的范围时，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。



## ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	指令执行完成后。	<ul style="list-style-type: none"> <li>◆ Done为TRUE, Execute由TRUE变为FALSE时。</li> <li>◆ 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE。</li> </ul>
Busy	Execute 由FALSE变为TRUE时。	<ul style="list-style-type: none"> <li>◆ Done变为TRUE时。</li> <li>◆ Error变为TRUE时。</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常。	<ul style="list-style-type: none"> <li>◆ Execute由TRUE变为FALSE时</li> </ul>

## ■ 功能说明

### ● 基本功能说明

- 该指令用于将轴组中各轴的命令位置设置为该指令设定的绝对位置。该指令执行后, 轴的命令位置和反馈位置都变化为设定的位置。该指令的执行不会让轴产生移动, 也不会让指定轴的物理位置发生变化, 一般用于重新定义位置坐标或者坐标偏移。该指令为控制器内部运算实现, 对伺服轴操作时不会改变驱动器的位置。
- 如通过该指令设定轴组中X轴、Y轴的当前位置为2000, CNC代码为G90 G1 X3000 Y5000,则执行G代码后, X轴从2000位置移动到3000位置, 移动的距离1000, Y轴从2000位置移动到5000位置, Y轴的移动的距离3000。
- 使用MC\_CoorMotion指令执行CNC代码时, 该指令会将轴组中各轴的反馈位置设置为轴组中各轴的的起点位置, 一般不需要执行该指令。需要将轴组中各轴的的起点位置设置为用户指定的位置时, 需要执行该指令。

### ● 指令完成时机

该指令的输出变量 Done 为 TRUE 表示参数设置完成。

### ● 异常排除

该指令执行时, 如果指令输入变量值不在允许的范围时, 该指令 Error 变为 TRUE, ErrorID 中有对应的错误码, 可以通过该错误码判断错误原因。

## 8.6 MC\_CoorMotion (CNC执行)

该指令用于执行控制器内的CNC代码。所属库：MotionControl\_Part2

适用机种：M500系列控制器（不包括M500S系列）

指令	名称	FB/FUN	梯形图样式	ST样式
MC_CoorMotion	CNC执行指令	FB		<pre>MC_CoorMotion_Instance ( Execute := 参数 , Pause:= 参数 , Stop:= 参数 , VelOverride:= 参数 , PreReadMaxNum:= 参数 , NCFile:= 参数 , AxesGroup:= 参数 , Mode:= 参数 , Res:= 参数 , Done =&gt; 参数 , Busy =&gt; 参数 , Active=&gt; 参数 , CommandAborted =&gt; 参数 , Error =&gt; 参数 , ErrorID=&gt; 参数 , CurrentLine=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
AxesGroup	轴组编号	USINT	1~8	不可缺省	轴组编号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
Pause	暂停	BOOL	TRUE或FALSE	FALSE	暂停轴组中所有轴的运动，按照CNC中执行命令的减速度减速停止
Stop	急停	BOOL	TRUE或FALSE	FALSE	急停轴组中所有轴的运动，一个周期停止
VelOverride	速度超调值	LREAL	0~500	0	速度超调值
PreReadMaxNum	预读CNC指令数量	UINT	1~5	不可缺省	预读CNC指令数量
NCFile	CNC编号	UINT	1~64	不可缺省	CNC编号
AxesGroup	轴组编号	USINT	1~8	不可缺省	轴组编号
Mode	模式	INT	保留	保留	保留
Res	保留	LREAL	保留	保留	保留

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成后变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Active	控制中	BOOL	TRUE或FALSE	指令控制轴时变为TRUE
CommandAborted	中断	BOOL	TRUE或FALSE	指令被中断时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时，输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”
CurrentLine	当前执行行数	UDINT	0~65535	CNC代码当前执行行数

## ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	指令执行完成后	<ul style="list-style-type: none"> <li>◆ Done为TRUE, Execute由TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE</li> </ul>
Busy	Execute 由FALSE变为TRUE时	<ul style="list-style-type: none"> <li>◆ Done变为TRUE时</li> <li>◆ Error变为TRUE时</li> <li>◆ CommandAborted变为TRUE时</li> </ul>
Active	指令开始控制轴时	<ul style="list-style-type: none"> <li>◆ Done从FALSE变为TRUE时</li> <li>◆ Error从FALSE变为TRUE时</li> <li>◆ CommandAborted从FALSE变为TRUE时</li> </ul>
CommandAborted	指令的执行被中断时	<ul style="list-style-type: none"> <li>◆ CommandAborted为TRUE, Execute由TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时, 该指令被其它指令中断时, CommandAborted变为TRUE一个周期后, 变为FALSE</li> </ul>
Error	指令输入变量值不在允许的范围、不满足指令的执行条件或者指令执行过程中遇到异常	<ul style="list-style-type: none"> <li>◆ Execute由TRUE变为FALSE时</li> </ul>

## ■ 功能说明

### ● 基本功能说明

- 该指令用于执行软件中下载到控制器的CNC代码。该指令执行前轴组内各轴状态必须处于StandStill状态, 否则指令执行报错。
- 该指令执行前, 请先将各个轴通过MC\_AddAxisToGroup指令将各个轴添加到轴组, 然后再执行该指令。哪几个轴为一组由MC\_AddAxisToGroup指令中AxesGroup的值决定。可以多个轴组同时执行, 如将1、2、3轴组成一个轴组, 4、5、6轴组成一个轴组, 使用两个MC\_CoorMotion指令, 两个指令中AxesGroup指定不同的值来实现。
- CNC代码的速度和加速度可以在CNC代码中指定, 也可以通过MC\_SetMoveDirectParm和MC\_SetMoveLinearParm指令指定。
- NCFile输入变量用于指定执行的CNC编号, 该编号可以通过软件中建立的CNC编号查看。
- Pause输入变量用于暂停CNC代码的执行, 该输入变量设置为TRUE后, CNC代码按照CNC代码中设置的减速速度减速停止, 正在执行的CNC代码暂停; 该输入变量设置为FALSE后, CNC代码按照代码中设置的的速度、加速度等继续执行。CNC代码暂停时, 轴组中各个轴的状态为不变。
- Stop输入变量用于急停CNC代码的执行, 该输入变量为TRUE时, 轴组中的所有轴会立即停止(一个周期命令速度变为0), 轴状态进入standstill(停止)状态, 该指令的输出变量Done变为TRUE。CNC代码执行时, 如果需要中断当前CNC代码的执行, 建议先将Pause引进设置为TRUE, 再将Stop输入变量设置为TRUE。
- MC\_CoorMotion指令和其它运动指令的目标速度一起控制目标速度。可以通过变更该指令VelFactor(目标速度比例)的值实时变更目标速度。VelFactor的单位为1%。如“50”表示“50%”。VelFactor参数值范围为0~500, 超出范围时执行该指令, 该指令会报错。
- 可以通过将VelFactor的值设置为0实现暂停的功能。
- 该指令的输出变量CurrentLine用于显示AxesGroup指定的CNC代码执行过的行数。

### ● 指令完成时机

该指令在 Execute 由 FALSE 变为 TRUE 后执行，执行由 NCFfile 指定的 CNC 代码，CNC 代码执行完毕时，Done 由 FALSE 变为 TRUE。CNC 代码执行过程中，该指令输入变量 Stop 由 FALSE 变为 TRUE 时，CNC 代码停止执行，该指令的输出变量 Done 变为 TRUE，轴状态进入 standstill（停止）状态。

- **重启该指令**

当指令执行完成后，Execute 再次由 FALSE 变为 TRUE 时，该指令可以重新执行；当指令正在执行中，Execute 再次由 FALSE 变为 TRUE 时，对指令的执行不会产生影响，指令仍按照未执行完成的输入变量执行指令。

- **其它指令执行时启动该指令**

轴组内各轴状态必须处于 StandStill 状态时才可以执行该指令。执行其它运动指令时执行该指令，该指令会报错。

- **该指令执行时启动其它指令**

该指令执行时，执行轴组相关的运动指令会报错，执行其它非轴组相关的运动指令，会打断该指令，指定轴的加速或者减速方式由执行指令的输入变量决定；轴组中其它轴进入 standstill（停止）状态，轴速度会一个周期降为 0。

- **异常排除**

该指令执行时，如果指令输入变量值不在允许的范围时，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。

## 8.7 MC\_GetMCodeStatus (M代码状态读取)

该指令用于读取M代码的状态及M代码对应的数值。所属库：MotionControl\_Part2

适用机种：M500系列控制器（不包括M500S系列）

指令	名称	FB/FUN	梯形图样式	ST样式
MC_GetMCodeStatus	M代码状态读取	FB		<pre>MC_GetMCodeStatus_Instance (   AxesGroup:= 参数 ,   Enable := 参数 ,   MCodeNum := 参数 ,   Valid =&gt; 参数 ,   Busy =&gt; 参数 ,   Error =&gt; 参数 ,   ErrorID=&gt; 参数 ,   Status=&gt; 参数 ,   Value=&gt; 参数 );</pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
AxesGroup	轴组编号	USINT	1~8	不可缺省	轴组编号
Enable	有效	BOOL	TRUE或FALSE	FALSE	设为TRUE, 读取M代码的状态 设为FALSE, 停止读取M代码的状态
MCodeNum	指定的M代码编号	USINT	正数	不可缺省	指定的M代码编号

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Valid	有效	BOOL	TRUE或FALSE	该指令读取M代码的状态时, 该参数值状态为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时, 输出错误代码 值的含义请参考“ <a href="#">指令错误代码描述</a> ”
Status	M代码的状态	BOOL	TRUE或FALSE	M代码的状态
Value	M代码对应的数值	LREAL	正数	M代码对应的数值

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Valid	M代码的状态可以读取时	<ul style="list-style-type: none"> <li>◆ Enable变为FALSE时</li> <li>◆ Error变为TRUE时</li> </ul>
Busy	Enable 由FALSE变为TRUE时	<ul style="list-style-type: none"> <li>◆ Done变为TRUE时</li> <li>◆ Error变为TRUE时</li> </ul>
Error	指令输入变量值不在允许的范围	<ul style="list-style-type: none"> <li>◆ Enable由TRUE变为FALSE时</li> </ul>

### ■ 功能说明

#### ● 基本功能说明

- 该指令用于读取指定M代码的状态及M代码对应的数值。M代码的编号由输入变量MCodeNum指定。该指令输入变量Enable为TRUE时, 输出变量Status和Value的值每个同步时钟周期更新一次; Enable为FALSE时, 输出变量Status和Value的值停止更新, 并保持最后Enable为TRUE时的值不变。

- M代码的状态及M代码对应的数值在CNC代码执行时，根据CNC代码中的M代码的执行进行变化。
- G代码和M代码不在同一行时，如下写法：

```
N1 G1 X100.5 Y100.5
```

```
N2 M10 D66.6
```

```
N3 G1 X200.8 Y300.8
```

CNC 代码执行完 N2 行的 M 代码后，编号为 10 的 M 代码的状态为 TRUE，编号为 10 的 M 代码对应的数值为 66.6，该指令输出变量 Status 的状态为 TRUE，Value 的值为 66.6。G 代码和 M 代码不在同一行时，M 代码所在行执行完毕后，CNC 代码暂停执行，需要通过 MC\_ResetMcode 指令复位 M 代码的状态后再继续执行。

- G代码和M代码在同一行时，如下写法：

```
N1 G1 X100.5 Y100.5 M10 D66.6
```

```
N2 G1 X200.8 Y300.8
```

CNC 代码执行完 N1 行的 G 代码和 M 代码后，编号为 10 的 M 代码的状态为 TRUE，编号为 10 的 M 代码对应的数值为 66.6，该指令输出变量 Status 的状态为 TRUE，Value 的值为 66.6。G 代码和 M 代码在同一行时，M 代码所在行执行完毕后，会继续执行后面的 CNC 代码。输出变量 Status 状态可以通过 MC\_ResetMcode 指令复位。

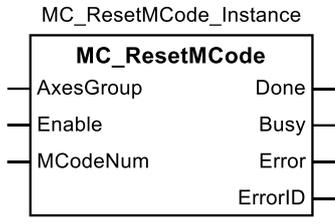
- **异常排除**

该指令执行时，如果指令输入变量值不在允许的范围时，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。

## 8.8 MC\_ResetMCode (M代码复位)

该指令用于复位指定编号的M代码。所属库: MotionControl\_Part2

适用机种: M500系列控制器 (不包括M500S系列)

指令	名称	FB/FUN	梯形图样式	ST样式
MC_ResetMCode	M代码复位	FB	 <pre> MC_ResetMCode_Instance   MC_ResetMCode   - AxesGroup      Done   - Enable         Busy   - MCodeNum       Error                   ErrorID </pre>	<pre> MC_ResetMCode_Instance (   AxesGroup:= 参数,   Enable := 参数,   MCodeNum := 参数 ,   Done=&gt; 参数 ,   Busy =&gt; 参数 ,   Error =&gt; 参数 ,   ErrorID=&gt; 参数 ); </pre>

### ■ 输入变量

输入变量	名称	数据类型	设定范围	缺省值	说明
AxesGroup	轴组编号	USINT	1~8	不可缺省	轴组编号
Execute	启动	BOOL	TRUE或FALSE	FALSE	检测到该参数的上升沿时执行该指令
MCodeNum	指定的M代码编号	USINT	正数	不可缺省	指定的M代码编号

### ■ 输出变量

名称	名称	数据类型	输出范围	功能
Done	完成	BOOL	TRUE或FALSE	指令执行完成后变为TRUE
Busy	执行中	BOOL	TRUE或FALSE	指令执行时变为TRUE
Error	错误	BOOL	TRUE或FALSE	指令执行发生异常时变为TRUE
ErrorID	错误代码	WORD	0~65535	指令执行异常时, 输出错误代码值的含义请参考“ <a href="#">指令错误代码描述</a> ”

### ■ 输出变量刷新时机

名称	变为TRUE的时机	变为FALSE的时机
Done	指令执行完成后	<ul style="list-style-type: none"> <li>◆ Done为TRUE, Execute由TRUE变为FALSE时</li> <li>◆ 指令执行且Execute为FALSE时, Done变为TRUE一个周期后, 变为FALSE</li> </ul>
Busy	Execute 由FALSE变为TRUE时	<ul style="list-style-type: none"> <li>◆ Done变为TRUE时</li> <li>◆ Error变为TRUE时</li> </ul>
Error	指令输入变量值不在允许的范围内	<ul style="list-style-type: none"> <li>◆ Execute由TRUE变为FALSE时</li> </ul>

### ■ 功能说明

#### ● 基本功能说明

- 该指令用于复位指定编号的M代码。M代码的状态在M代码执行后变为TRUE, 可以通过该指令将M代码的状态复位为FALSE。
- G代码和M代码不在同一行时, 如下写法

N1 G1 X100.5 Y100.5

N2 M10 D66.6

N3 G1 X200.8 Y300.8

CNC 代码执行完 N2 行的 M 代码后，编号为 10 的 M 代码的状态为 TRUE。G 代码和 M 代码不在同一行时，M 代码所在行执行完毕后，CNC 代码暂停执行，需要通过该指令复位 M 代码的状态后再继续执行。

➤ G代码和M代码在同一行时，如下写法:

N1 G1 X100.5 Y100.5 M10 D66.6

N2 G1 X200.8 Y300.8

CNC 代码执行完 N1 行的 G 代码和 M 代码后，编号为 10 的 M 代码的状态为 TRUE，可以通过该指令复位 M 代码的状态。G 代码和 M 代码在同一行时，M 代码所在行执行完毕后，会继续执行后面的 CNC 代码。

- **异常排除**

该指令执行时，如果指令输入变量值不在允许的范围时，该指令 Error 变为 TRUE，ErrorID 中有对应的错误码，可以通过该错误码判断错误原因。

## 8.9 CNC代码调用执行示例程序

### ■ 调用 CNC 代码需要的步骤

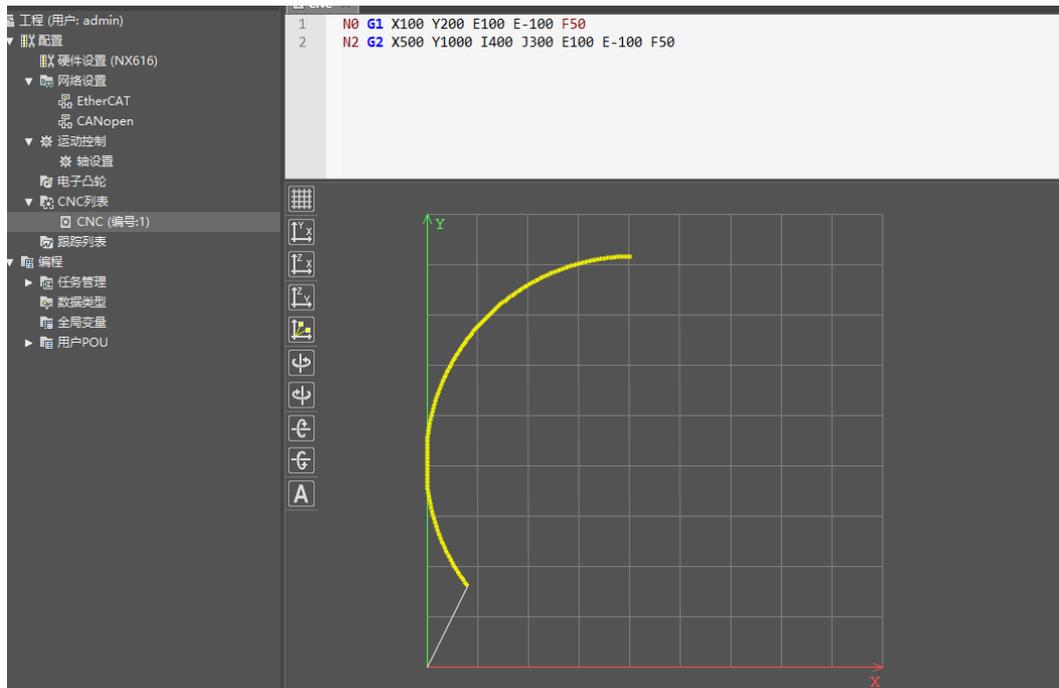
步骤	步骤描述	备注
步骤1	设置各个轴的轴参数	在软件中设定
步骤2	轴使能	伺服轴执行MC_Power指令, 虚拟伺服轴不需要
步骤3	添加轴到轴组	轴组中的逻辑轴和软件中的配置的轴建立对应关系
步骤4	执行CNC执行指令调用CNC代码执行	执行CNC代码

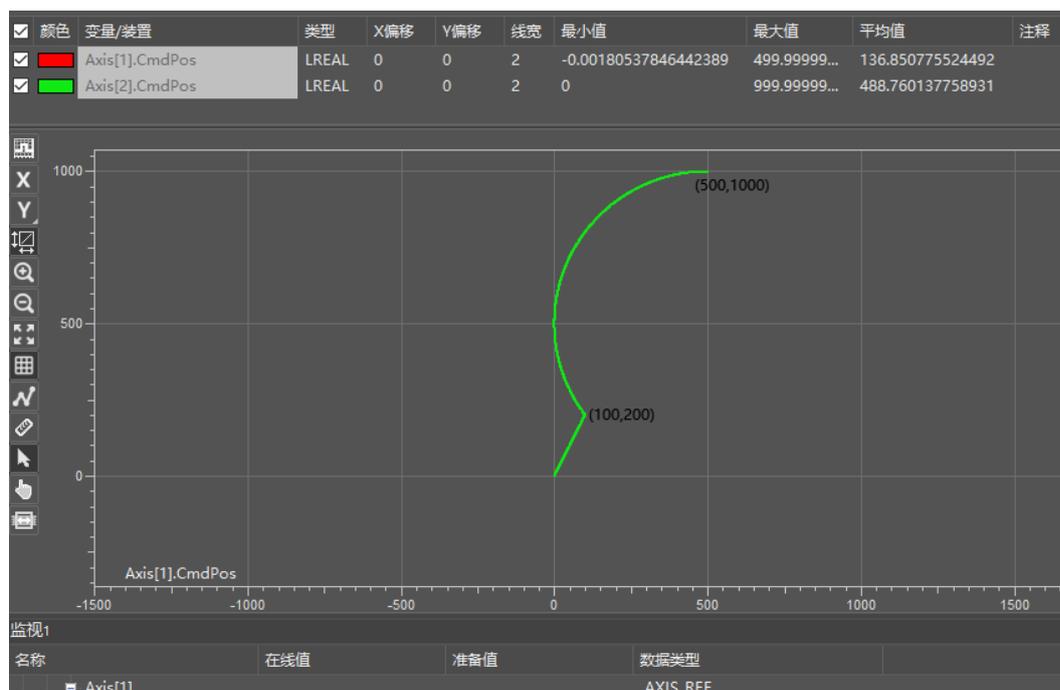
### ■ 轴参数设置

轴1和轴2轴参数设置如下图所示, 轴1和轴2轴参数设置相同。



### ■ NC 代码轨迹曲线图

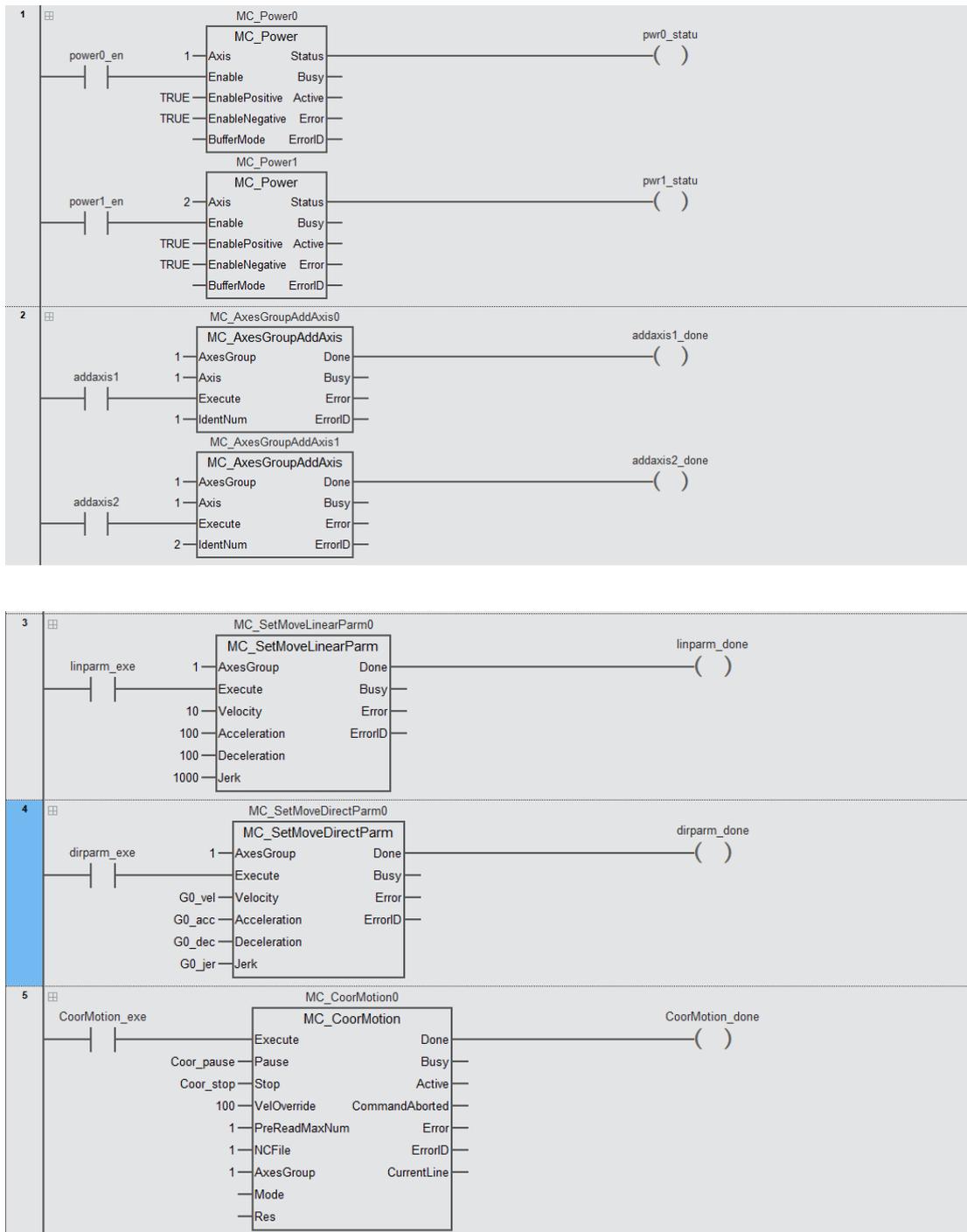




## ● 变量表

类别	名称	分配到	数据类型	初始值	注释
VAR	MC_Power0		MC_Power		
VAR	MC_Power1		MC_Power		
VAR	power0_en		BOOL		
VAR	power1_en		BOOL		
VAR	pwr0_statu		BOOL		
VAR	pwr1_statu		BOOL		
VAR	MC_AxesGroupAddAxis0		MC_AxesGroupAddAxis		
VAR	MC_AxesGroupAddAxis1		MC_AxesGroupAddAxis		
VAR	addaxis1		BOOL		
VAR	addaxis2		BOOL		
VAR	addaxis1_done		BOOL		
VAR	addaxis2_done		BOOL		
VAR	MC_SetMoveLinearParm0		MC_SetMoveLinearParm		
VAR	linparm_exe		BOOL		
VAR	linparm_done		BOOL		
VAR	MC_SetMoveDirectParm0		MC_SetMoveDirectParm		
VAR	dirparm_exe		BOOL		
VAR	G0_vel		ARRAY [1..8] OF LREAL	[8(50)]	
VAR	G0_acc		ARRAY [1..8] OF LREAL	[8(100)]	
VAR	G0_dec		ARRAY [1..8] OF LREAL	[8(100)]	
VAR	G0_jer		ARRAY [1..8] OF LREAL	[8(1000)]	
VAR	dirparm_done		BOOL		
VAR	MC_CoorMotion0		MC_CoorMotion		
VAR	CoorMotion_exe		BOOL		
VAR	Coor_pause		BOOL		
VAR	Coor_stop		BOOL		
VAR	CoorMotion_done		BOOL		

■ 梯形图语言范例程序



## ■ ST 语言范例程序

```
MC_Power0(  
  Axis:=1 ,  
  Enable:= power0_en,  
  EnablePositive:= TRUE,  
  EnableNegative:= TRUE,  
  BufferMode:=0 ,  
  Status=> pwr0_statu  
);
```

```
MC_Power1(  
  Axis:=2 ,  
  Enable:= power1_en,  
  EnablePositive:= TRUE,  
  EnableNegative:= TRUE,  
  BufferMode:=0 ,  
  Status=>pwr0_statu  
);
```

```
MC_AxesGroupAddAxis0(  
  AxesGroup:=1 ,  
  Axis:=1 ,  
  Execute:= addaxis1 ,  
  IdentNum:=1 ,  
  Done=> addaxis1_done  
);
```

```
MC_AxesGroupAddAxis1(  
  AxesGroup:=1 ,  
  Axis:=2 ,  
  Execute:= addaxis2 ,  
  IdentNum:=2 ,  
  Done=> addaxis2_done  
);
```

```
MC_SetMovelinearParm0(  
  AxesGroup:=1 ,  
  Execute:= linparm_exe ,  
  Velocity:= 10 ,  
  Acceleration:=100 ,  
  Deceleration:=100 ,  
  erk:=1000 ,  
  Done=> linparm_done  
);
```

```
MC_SetMoveDirectParm0(  
  AxesGroup:=1 ,  
  Execute:= dirparm_exe ,  
  Velocity:= G0_vel ,  
  Acceleration:=G0_acc ,  
  Deceleration:=G0_dec ,  
  Jerk:=G0_jer ,  
  Done=> dirparm_exe
```

```
);  
  
MC_CoorMotion0(Execute:=CoorMotion_exe ,  
  Pause:=Coor_pause ,  
  Stop:=Coor_stop ,  
  VelOverride:=100 ,  
  PreReadMaxNum:=1 ,  
  NCFile:=1 ,  
  AxesGroup:=1 ,  
  Done=>CoorMotion_done  
);
```

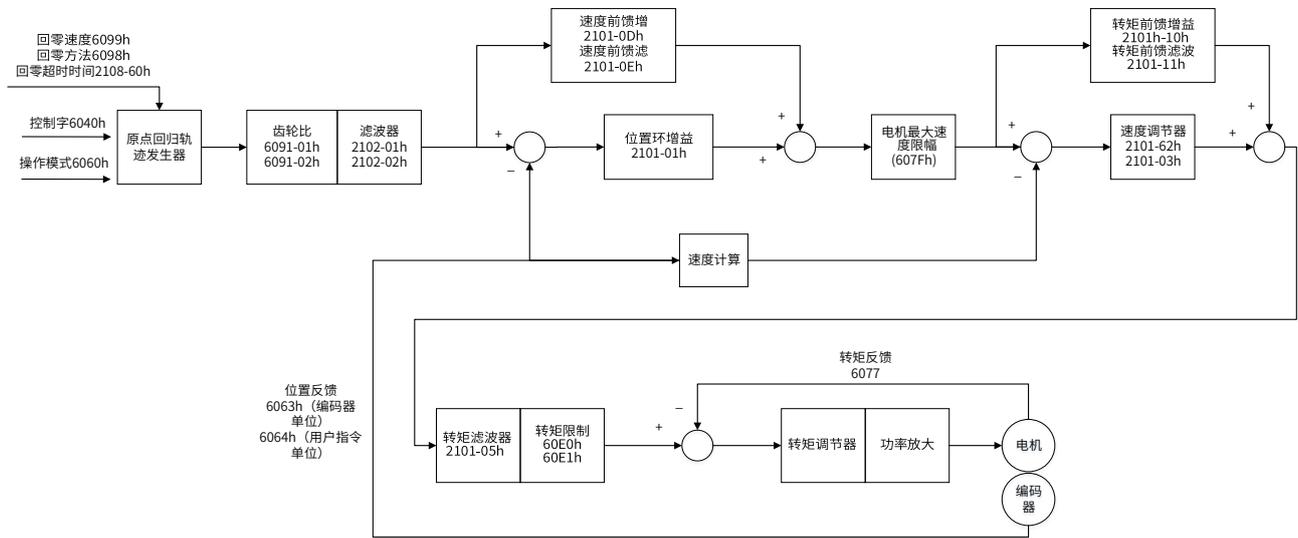


# 附录

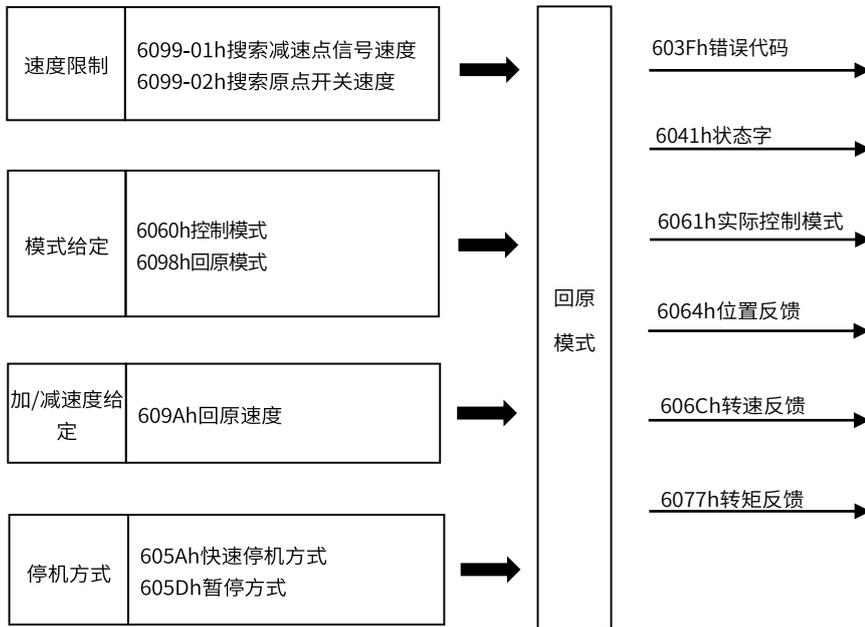


## 附录1 原点返回模式

根据原点开关信号、限位开关信号和编码器Z信号，CiA402协议定义了31种回原方式。启用此模式时，将对象6060H设置为6。此模式适用于EtherCAT伺服驱动器。



图附录1-1 原点回归模式控制框图



图附录1-2 原点模式输入输出

## 附录1.1 原点返回模式中的控制字设定 (60400010h)

选择原点返回模式时，控制字（6040h）各个位的意义如表附录1-1所示，其中背景用深颜色标注的是原点返回模式专用的控制命令。

表附录1-1 原点返回模式的控制字说明

Bit	名称	说明
0	Switch on	使能伺服时必须设置为 1。
1	Enable voltage	使能伺服时必须设置为 1。
2	快速停机	使能伺服时必须设置为 1，设置为 0 则快速停机。
3	Operation enable	使能伺服时必须设置为 1。
4	回原使能	0: 无效，1: 有效。有效时启动回原点流程，在回原点全程必须保持为有效，切换到无效则停止回原点流程
5、6	原点模式预留	暂无
7	故障复位	在 0→1 变化时执行一次故障复位，如需多次复位，则需要产生多次 0→1 变化。此位置 1 时，其它控制指令
8	暂停	0: 无效，1: 有效。有效时减速停止回原点流程。
9	原点模式预留	暂无
10	预留	暂无
11~15	厂家自定义	暂无

## 附录1.2 原点返回模式的状态字定义 (60410010h)

选择原点返回模式时，状态字（6041h）各个位的意义如表附录1-2 所示。其中背景用深颜色标注的是原点返回模式专用的状态。

表附录1-2 原点返回模式的状态字说明

Bit	名称	说明
0	Ready to switch on	0: 无效， 1: 有效。有效时表示可以使能伺服
1	Switched on	0: 无效， 1: 有效。有效时表示可以使能伺服
2	Operation enabled	0: 无效， 1: 有效。有效时表示伺服已使能
3	伺服故障	0: 无故障，1: 有故障
4	Voltage enabled	0: 无效，1: 有效。有效时表示可以使能伺服
5	快速停机	0: 快速停机有效， 1: 快速停机无效
6	Switch on disabled	0: 无效， 1: 有效。有效时表示可以使能伺服
7	警告	0: 无警告，1: 有警告
8	厂家自定义	暂无
9	远程控制	0: 无效， 1: 有效。有效时表示控制字已生效
10	位置到达	60400010h bit 8 (暂停)=0, 0: 位置未到达，1: 位置到达； 60400010h bit 8 (暂停)=1, 0: 减速中，1: 速度为 0
11	内部软限位状态	0: 没有到达软限位， 1: 到达软限位
12	回原点完成输出	0: 回原点未完成，1: 回原点完成
13	回原点错误	0: 无错误，1: 回原点发生错误
14	厂家自定义	暂无

15	回原完成	0: 无效, 1: 已完成回原点。 对于绝对值系统: P06.47设2, P09.14 的十六进制值右起第2位设置为1, 回原成功后会存储 bit15的值 (掉电保持), 将P20.06设置为7可清除存储值
----	------	--

### 附录1.3 原点返回模式相关的参数

表附录1-3所示, 列出了原点返回模式涉及到的字典对象。

表附录1-3 原点返回模式相关的字典对象

索引	子索引	名称	访问类型	数据类型	默认值
603Fh		错误代码	ro	unsigned16	0
6040h		控制字	rw	unsigned16	0
6041h		状态字	ro	unsigned16	0
6060h		控制模式	rw	integer8	0
6061h		控制模式显示	ro	integer8	0
6062h		用户位置指令	ro	integer32	0
6063h		电机位置反馈	ro	integer32	0
6064h		用户位置反馈	ro	integer32	0
6065h		用户位置偏差过大阈值	rw	unsigned32	100000000
6067h		位置到达阈值	rw	unsigned32	100
6068h		位置到达时间	rw	unsigned16	1
606Bh		用户速度指令值	ro	integer32	0
606Ch		用户实际速度反馈	ro	integer32	0
607Ch		原点偏置	rw	integer32	0
607Dh	01h	软限位: 最小位置限制	rw	integer32	-2147483648
	02h	软限位: 最大位置限制	rw	integer32	2147483647
6098h		回原模式	rw	integer8	0
6099h	01h	回原模式中搜索减速度信号速度	rw	unsigned32	218453
	02h	回原模式中搜索原点开关信号速度	rw	unsigned32	21845
609Ah		回原加速度	rw	unsigned32	1310720

### 附录1.4 原点返回模式简单使用教程 (以X3E伺服举例)

1. 设置X3E伺服驱动器参数, 配置回原DI相关参数 (第4组参数: 数字输入输出, 其中P6.28=0)

表附录1-4 运行原点返回模式时的伺服驱动器参数配置

参数地址	设定值	说明
P00.01 (2100-02h)	7	CANopen/EtherCAT模式
P09.00 (2109-01h)	1	CANopen从站地址 (默认为1)
P09.18 (2109-12h)	1	EtherCAT从站地址 (默认为1)
P09.13 (2109-0Eh)	5	波特率 (默认为500K, EtherCAT不用设置, 固定100M)

2. 上位控制器连接伺服驱动器, 设置CANopen通讯参数、组态PDO参数等, EtherCAT不用设置。

表附录1-5 原点回归模式启动及运行流程

地址	名称	值设定 (10 进制数值)
60600008h	控制模式	6
60980008h	回原模式	1~35
60400010h 控制字	报警清除	任意数 → 128 (上升沿有效, 如能清除)
	回原	6→7→15→ 31(BIT4上升沿触发回原)

60990120h	回原模式中搜索减速点信号速度	默认值: 218453 (指令单位/s)
60990220h	回原模式中搜索原点开关信号速度	默认值: 21845 (指令单位/s)
609A0020h	回原加速度	默认值: 1310720 (指令单位/s <sup>2</sup> )

## 附录1.5 原点回归模式介绍

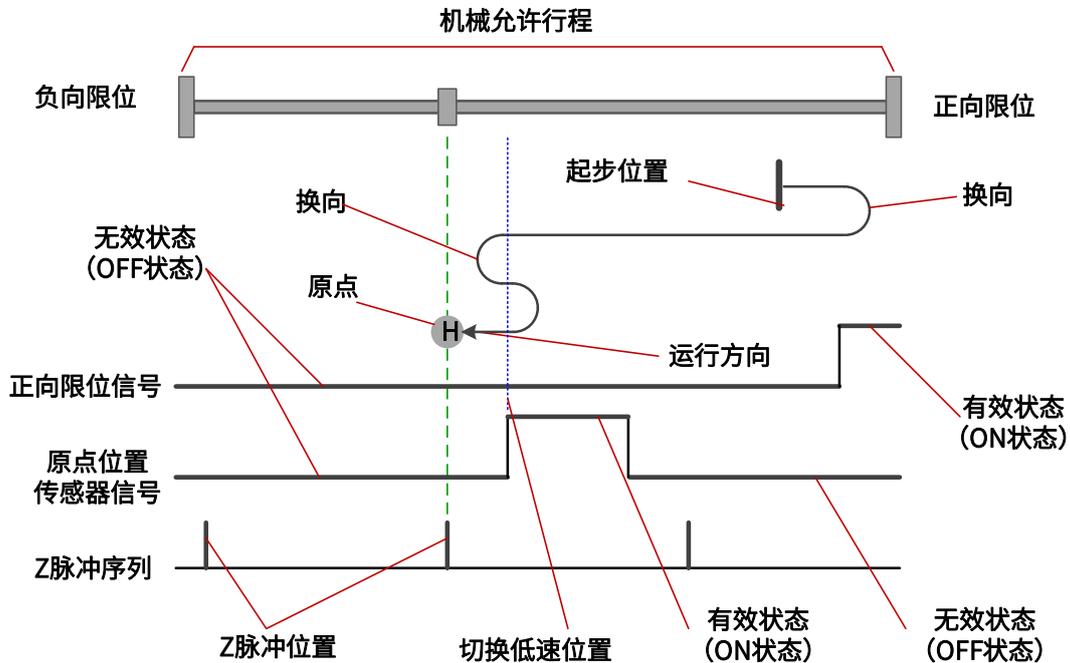
CiA402内部定义了31种回原方式（适用于CANopen/EtherCAT），如表附录1-6所述。

以下描述中以HSW表示原点位置传感器信号，以NL表示负向限位信号，以PL表示正向限位信号。ON表示信号的有效状态，OFF表示信号的无效状态。OFF→ON表示信号从无效状态到有效状态的跳变沿，ON→OFF表示信号从有效状态到无效状态的跳变沿。下面分别介绍各种原点模式运行轨迹和信号状态变化，各种回原点模式图示中的图标意义如图附录1-3所示。

表附录1-6 原点回归模式启动及运行流程

回原方式	说明
0	无
1	起步朝负向运行，以负向运行时遇到 NL 的 OFF→ON 状态时换低速运行，然后回退找最近的 Z 脉冲位置作为原点
2	起步朝正向运行，正向运行时遇到 PL 的 OFF→ON 状态时换低速运行，然后回退找最近的 Z 脉冲位置作为原点
3	起步时 HSW 无效则朝正向运行，否则朝负向运行。朝负向运行时遇到 HSW 的 ON→OFF 状态时换低速运行，然后继续负向运行找最近的 Z 脉冲位置作为原点
4	起步时 HSW 无效则朝正向运行，否则朝负向运行。朝正向运行时遇到 HSW 的 OFF→ON 状态时换低速运行，然后继续正向运行找最近的 Z 脉冲位置作为原点
5	起步时 HSW 无效则朝负向运行，否则朝正向运行。朝正向运行时遇到 HSW 的 ON→OFF 状态时换低速运行，然后继续正向运行找最近的 Z 脉冲位置作为原点
6	起步时 HSW 无效则朝负向运行，否则朝正向运行。朝负向运行时遇到 HSW 的 ON→OFF 状态时换低速运行，然后继续负向运行找最近的 Z 脉冲位置作为原点
7	起步时 HSW 无效则朝正向运行，否则朝负向运行。朝负向运行时遇到 HSW 的 ON→OFF 状态时换低速运行，然后继续负向运行找最近的 Z 脉冲位置作为原点
8	起步时 HSW 无效则朝正向运行，否则朝负向运行。朝正向运行时遇到 HSW 的 OFF→ON 状态时换低速运行，然后继续正向运行找最近的 Z 脉冲位置作为原点
9	起步时都是朝正向运行，不论 HSW 有效或无效。朝负向运行时遇到 HSW 的 OFF→ON 状态时换低速运行，然后继续负向运行找最近的 Z 脉冲位置作为原点
10	起步时都是朝正向运行，不论 HSW 有效或无效。朝正向运行时遇到 HSW 的 ON→OFF 状态时换低速运行，然后继续正向运行找最近的 Z 脉冲位置作为原点
11	起步时 HSW 无效则朝负向运行，否则朝正向运行。朝正向运行时遇到 HSW 的 ON→OFF 状态时换低速运行，然后继续正向运行找最近的 Z 脉冲位置作为原点
12	起步时 HSW 无效则朝负向运行，否则朝正向运行。朝负向运行时遇到 HSW 的 OFF→ON 状态时换低速运行，然后继续负向运行找最近的 Z 脉冲位置作为原点
13	起步时都是朝负向运行，不论 HSW 有效或无效。朝正向运行时遇到 HSW 的 OFF→ON 状态时换低速运行，然后继续正向运行找最近的 Z 脉冲位置作为原点
14	起步时都是朝负向运行，不论 HSW 有效或无效。朝负向运行时遇到 HSW 的 ON→OFF 状态时换低速运行，然后继续负向运行找最近的 Z 脉冲位置作为原点
15	保留
16	保留
17	类似方式 1，但不找 Z 脉冲，以负向运行时遇到 NL 的 OFF→ON 状态位置作为原点
18	类似方式 2，但不找 Z 脉冲，以正向运行时遇到 PL 的 OFF→ON 状态位置作为原点
19	类似方式 3，但不找 Z 脉冲，以负向运行时遇到 HSW 的 ON→OFF 状态位置作为原点
20	类似方式 4，但不找 Z 脉冲，以正向运行时遇到 HSW 的 OFF→ON 状态位置作为原点
21	类似方式 5，但不找 Z 脉冲，以正向运行时遇到 HSW 的 ON→OFF 状态位置作为原点
22	类似方式 6，但不找 Z 脉冲，以负向运行时遇到 HSW 的 OFF→ON 状态位置作为原点

23	类似方式 7, 但不找 Z 脉冲, 以朝负向运行时遇到 HSW 的 ON→OFF 状态位置作为原点
24	类似方式 8, 但不找 Z 脉冲, 以朝正向运行时遇到 HSW 的 OFF→ON 状态位置作为原点
25	类似方式 9, 但不找 Z 脉冲, 以朝负向运行时遇到 HSW 的 OFF→ON 状态位置作为原点
26	类似方式 10, 但不找 Z 脉冲, 以朝正向运行时遇到 HSW 的 ON→OFF 状态位置作为原点
27	类似方式 11, 但不找 Z 脉冲, 以朝正向运行时遇到 HSW 的 ON→OFF 状态位置作为原点
28	类似方式 12, 但不找 Z 脉冲, 以朝负向运行时遇到 HSW 的 OFF→ON 状态位置作为原点
29	类似方式 13, 但不找 Z 脉冲, 以朝正向运行时遇到 HSW 的 OFF→ON 状态位置作为原点
30	类似方式 14, 但不找 Z 脉冲, 以朝负向运行时遇到 HSW 的 ON→OFF 状态位置作为原点
31	保留
32	保留
33	起步时朝负向找最近的 Z 脉冲位置作为原点
34	起步时朝正向找最近的 Z 脉冲位置作为原点
35	以当前位置为原点



图附录1-3 原点模式图示中各种图标的意义

一般的, 建议将原点模式3~6、19~22, 应用在HSW的OFF/ON状态正好将整个机械允许行程范围分成两部分的情形, 因为这8种模式下, 无论何时遇到NL还是PL, 都是停止并报警, 并不会自动反向寻找原点。

建议将原点模式7~14、23~30, 应用在HSW的ON状态正好将整个机械允许行程范围分成三部分的情形, 此时ON状态区间只占据整个机械允许行程范围很小一部分 (即ON状态是短时暂态)。

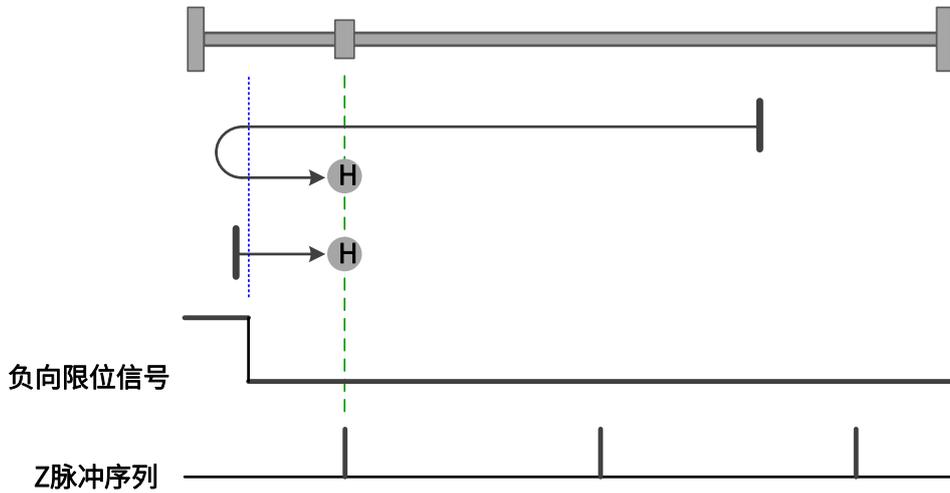
以上只是建议, 并不是强制要求。

### ■ 模式 1，寻找负限位和 Z 脉冲

起步时如果NL无效，则以高速朝负向运行，遇到NL的OFF→ON状态之后减速停止，然后换低速朝正向运行。在低速朝正向运行时遇到NL的ON→OFF状态之后，继续朝正向找最近的Z脉冲位置作为原点。

起步时如果NL有效，则以低速朝正向运行。在朝正向遇到NL的ON→OFF状态之后，继续正向找最近的Z脉冲位置作为原点。

如图附录1-4所示，参见表附录1-6。



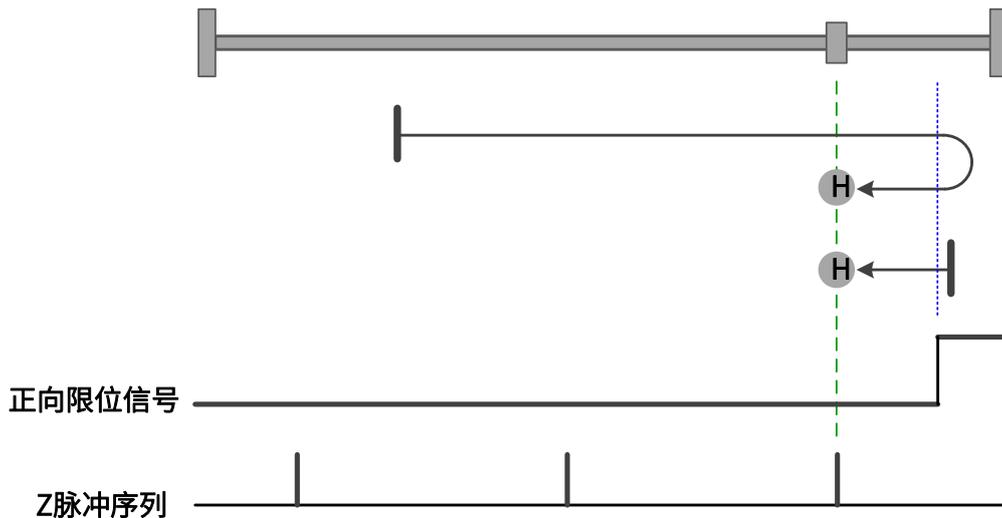
图附录1-4 原点模式1轨迹及信号状态

### ■ 模式 2，寻找正限位和 Z 脉冲

起步时如果PL无效，则以高速朝正向运行，遇到PL的OFF→ON状态之后减速停止，然后换低速朝负向运行。在低速朝负向运行时遇到PL的ON→OFF状态之后，继续朝负向找最近的Z脉冲位置作为原点。

起步时如果PL有效，则以低速朝负向运行。在朝负向运行时遇到PL的ON→OFF状态之后，继续朝负向找最近的Z脉冲位置作为原点。

如图附录1-5所示，参见表附录1-6。



图附录1-5 原点模式2轨迹及信号状态

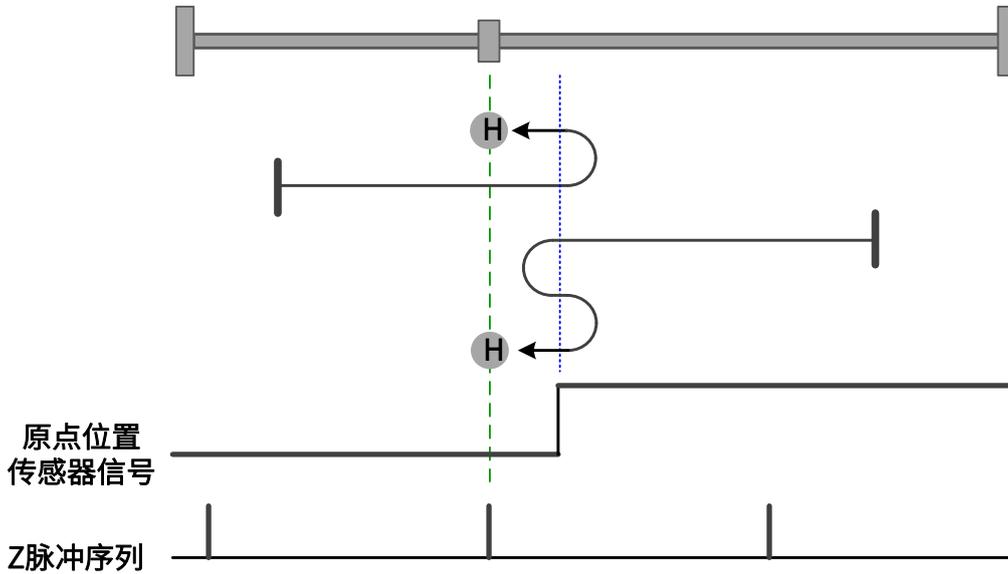
### ■ 模式 3，寻找朝负向运行时 HSW 的 ON→OFF 位置和 Z 脉冲

起步时 HSW 无效则以高速朝正向运行，在正向运行时遇到 HSW 的 OFF→ON 状态之后减速停止，然后换低速朝负向运行。在低速负向运行时遇到 HSW 的 ON→OFF 状态之后，继续朝负向找最近的 Z 脉冲位置作为原点。

起步时 HSW 有效则以高速朝负向运行。在负向运行时遇到 HSW 的 ON→OFF 状态之后减速停止，然后高速回退到 HSW 有效的位置之后再减速停止，此后换低速朝负向运行。在低速负向运行时遇到 HSW 的 ON→OFF 状态之后，继续朝负向找最近的 Z 脉冲位置作为原点。

这种模式下，无论遇到 NL 还是 PL 的 ON 状态，都是停止回原点流程并报警。

如图附录 1-6 所示，参见表附录 1-6。



图附录 1-6 原点模式 3 轨迹及信号状态

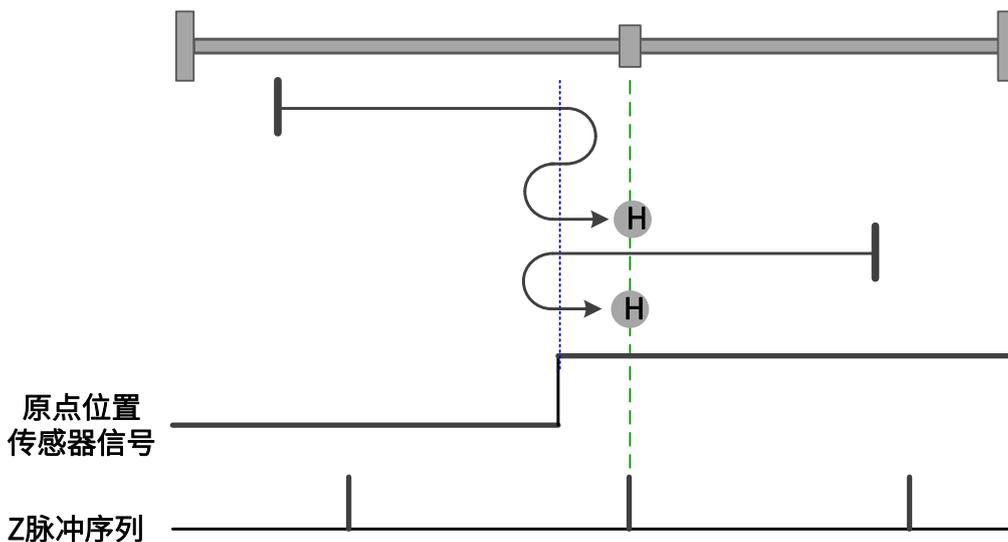
### ■ 模式 4，寻找朝正向运行时 HSW 的 OFF→ON 位置和 Z 脉冲

起步时 HSW 无效则以高速朝正向运行，在正向运行时遇到 HSW 的 OFF→ON 状态之后减速停止，然后高速回退到 HSW 无效的位置之后再减速停止，此后换低速朝正向运行。在低速正向运行时遇到 HSW 的 OFF→ON 状态之后，继续朝正向找最近的 Z 脉冲位置作为原点。

起步时 HSW 有效则以高速朝负向运行。在负向运行时遇到 HSW 的 ON→OFF 状态之后减速停止，然后换低速朝正向运行。在低速正向运行时遇到 HSW 的 OFF→ON 状态之后，继续朝正向找最近的 Z 脉冲位置作为原点。

这种模式下，无论遇到 NL 还是 PL 的 ON 状态，都是停止回原点流程并报警。

如图附录 1-7 所示，参见表附录 1-6。



图附录 1-7 原点模式 4 轨迹及信号状态

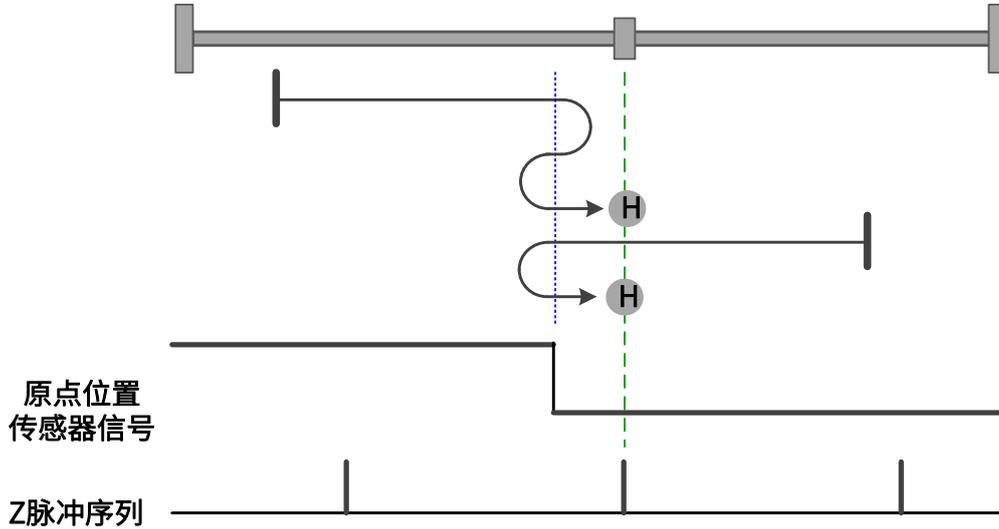
■ 模式 5，寻找朝正向运行时 HSW 的 ON→OFF 位置和 Z 脉冲

起步时 HSW 无效则以高速朝负向运行。在负向运行时遇到 HSW 的 OFF→ON 状态之后减速停止，然后换低速朝正向运行。在低速正向运行时遇到 HSW 的 ON→OFF 状态之后，继续朝正向找最近的 Z 脉冲位置作为原点。

起步时 HSW 有效则以高速朝正向运行，在正向运行时遇到 HSW 的 ON→OFF 状态之后减速停止，然后高速回退到 HSW 有效的位置之后再减速停止，此后换低速朝正向运行。在低速正向运行时遇到 HSW 的 ON→OFF 状态之后，继续朝正向找最近的 Z 脉冲位置作为原点。

这种模式下，无论遇到 NL 还是 PL 的 ON 状态，都是停止回原点流程并报警。

如图附录 1-8 所示，参见表附录 1-6。



图附录 1-8 原点模式 5 轨迹及信号状态

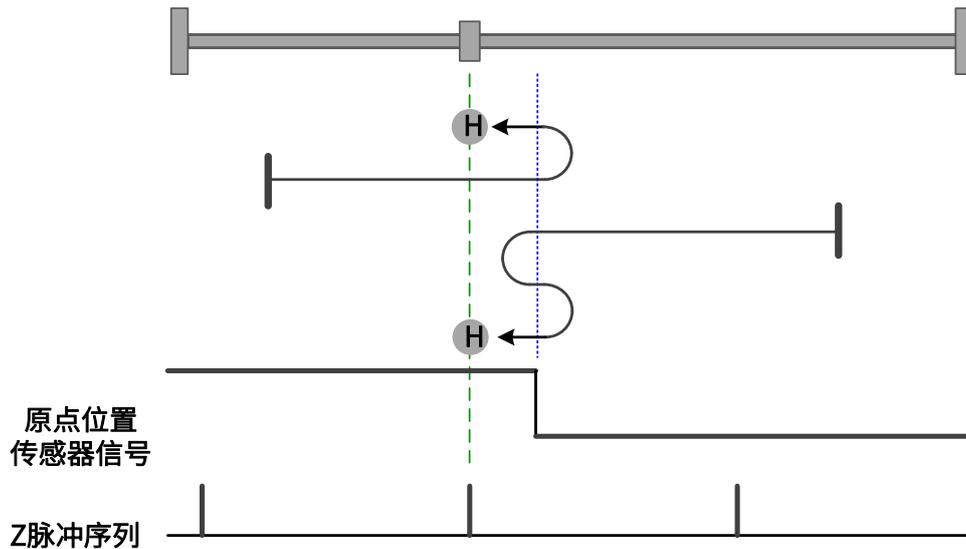
■ 模式 6，寻找朝负向运行时 HSW 的 OFF→ON 位置和 Z 脉冲

起步时 HSW 无效则以高速朝负向运行。在负向运行时遇到 HSW 的 OFF→ON 状态之后减速停止，然后高速回退到 HSW 无效的位置之后再减速停止，此后换低速朝负向运行。在低速负向运行时遇到 HSW 的 OFF→ON 状态之后，继续朝负向找最近的 Z 脉冲位置作为原点。

起步时 HSW 有效则以高速朝正向运行，在正向运行时遇到 HSW 的 ON→OFF 状态之后减速停止，然后换低速朝负向运行。在低速负向运行时遇到 HSW 的 OFF→ON 状态之后，继续朝负向找最近的 Z 脉冲位置作为原点。

这种模式下，无论遇到 NL 还是 PL 的 ON 状态，都是停止回原点流程并报警。

如图附录 1-9 所示，参见表附录 1-6。



图附录 1-9 原点模式 6 轨迹及信号状态

### ■ 模式 7，寻找朝负向运行时 HSW 的 ON→OFF 位置和 Z 脉冲，遇正限位自动反向

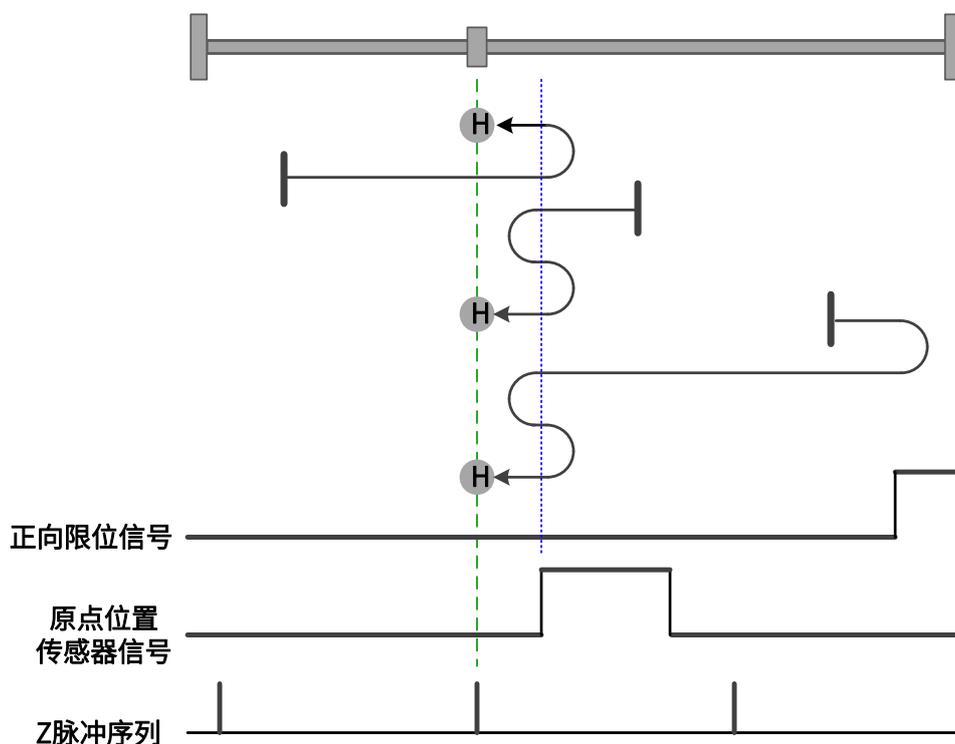
起步时 HSW 无效且位于原点位置传感器所在位置的右侧，则以高速朝正向运行，遇到 PL 的 ON 状态时减速停止，然后以高速朝负向运行。在负向运行时遇到 HSW 的 ON→OFF 状态之后减速停止，然后高速回退到 HSW 有效的位置之后再减速停止（如果 HSW 有效的区间很窄，则可能进入另一侧 HSW 无效的位置区间），此后换低速朝负向运行。在低速负向运行时遇到 HSW 的 ON→OFF 状态之后，继续朝负向找最近的 Z 脉冲位置作为原点。

起步时 HSW 无效且位于原点位置传感器所在位置的左侧，则以高速朝正向运行，在正向运行时遇到 HSW 的 OFF→ON 状态之后减速停止，然后换低速朝负向运行。在低速负向运行时遇到 HSW 的 ON→OFF 状态之后，继续朝负向找最近的 Z 脉冲位置作为原点。

起步时 HSW 有效则以高速朝负向运行。在负向运行时遇到 HSW 的 ON→OFF 状态之后减速停止，然后高速回退到 HSW 有效的位置之后再减速停止（如果 HSW 有效的区间很窄，则可能进入另一侧 HSW 无效的位置区间），此后换低速朝负向运行。在低速负向运行时遇到 HSW 的 ON→OFF 状态之后，继续朝负向找最近的 Z 脉冲位置作为原点。

这种模式下，朝正向运行第一次遇到 PL 的 ON 状态时自动反向；遇到 NL 的 ON 状态，或者再次遇到 PL 的 ON 状态，则停止回原点流程并报警。

如图附录 1-10 所示，参见表附录 1-6。



图附录 1-10 原点模式 7 轨迹及信号状态

### ■ 模式 8，寻找朝正向运行时 HSW 的 OFF→ON 位置和 Z 脉冲，遇正限位自动反向

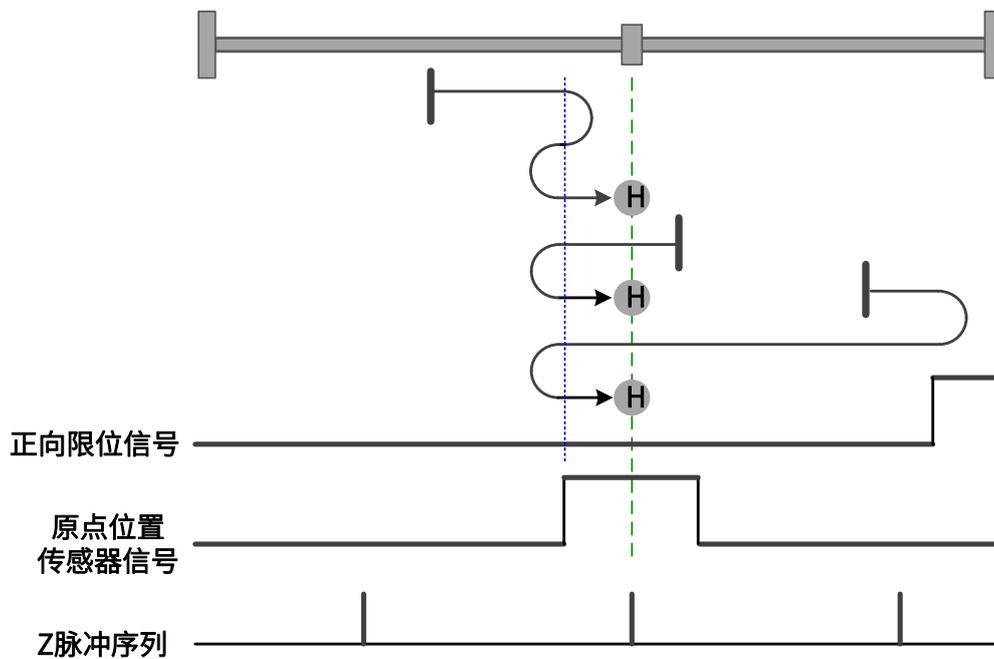
起步时 HSW 无效且位于原点位置传感器所在位置的右侧，则以高速朝正向运行，遇到 PL 的 ON 状态时减速停止，然后以高速朝负向运行。在负向运行时遇到 HSW 的 ON→OFF 状态之后减速停止，然后换低速朝正向运行。在低速正向运行时遇到 HSW 的 OFF→ON 状态之后，继续朝正向找最近的 Z 脉冲位置作为原点。

起步时 HSW 无效且位于原点位置传感器所在位置的左侧，则以高速朝正向运行，在正向运行时遇到 HSW 的 OFF→ON 状态之后减速停止，然后高速回退到 HSW 无效的位置之后再减速停止，此后换低速朝正向运行。在低速正向运行时遇到 HSW 的 OFF→ON 状态之后，继续朝正向找最近的 Z 脉冲位置作为原点。

起步时 HSW 有效则以高速朝负向运行。在负向运行时遇到 HSW 的 ON→OFF 状态之后减速停止，然后换低速朝正向运行。在低速正向运行时遇到 HSW 的 OFF→ON 状态之后，继续朝正向找最近的 Z 脉冲位置作为原点。

这种模式下，朝正向运行第一次遇到 PL 的 ON 状态时自动反向；遇到 NL 的 ON 状态，或者再次遇到 PL 的 ON 状态，则停止回原点流程并报警。

如图附录 1-11 所示，参见表附录 1-6。



图附录1-11 原点模式8轨迹及信号状态

■ 模式 9，寻找朝负向运行时 HSW 的 OFF→ON 位置和 Z 脉冲，遇正限位自动反向

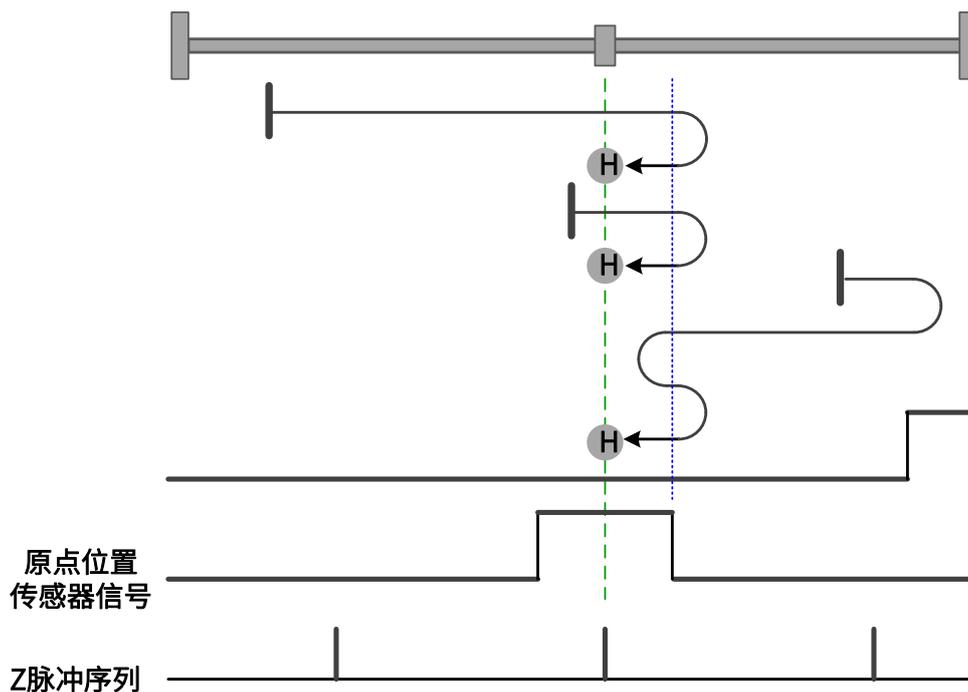
起步时 HSW 无效且位于原点位置传感器所在位置的正当侧，则以高速朝正向运行，遇到 PL 的 ON 状态时减速停止，然后以高速朝负向运行。在负向运行时遇到 HSW 的 OFF→ON 状态之后减速停止，然后高速回退到 HSW 无效的位置之后再减速停止，此后换低速朝负向运行。在低速负向运行时遇到 HSW 的 OFF→ON 状态之后，继续朝负向找最近的 Z 脉冲位置作为原点。

起步时 HSW 无效且位于原点位置传感器所在位置的负向侧，则以高速朝正向运行，在正向运行时遇到 HSW 的 ON→OFF 状态之后减速停止，然后换低速朝负向运行。在低速负向运行时遇到 HSW 的 OFF→ON 状态之后，继续朝负向找最近的 Z 脉冲位置作为原点。

起步时 HSW 有效则以高速朝正向运行。在正向运行时遇到 HSW 的 ON→OFF 状态之后减速停止，然后换低速朝负向运行。在低速负向运行时遇到 HSW 的 OFF→ON 状态之后，继续朝负向找最近的 Z 脉冲位置作为原点。

这种模式下，朝正向运行第一次遇到 PL 的 ON 状态时自动反向；遇到 NL 的 ON 状态，或者再次遇到 PL 的 ON 状态，则停止回原点流程并报警。

如图附录1-12所示，参见表附录1-6。



图附录1-12 原点模式9轨迹及信号状态

### ■ 模式 10，寻找朝正向运行时 HSW 的 ON→OFF 位置和 Z 脉冲，遇正限位自动反向

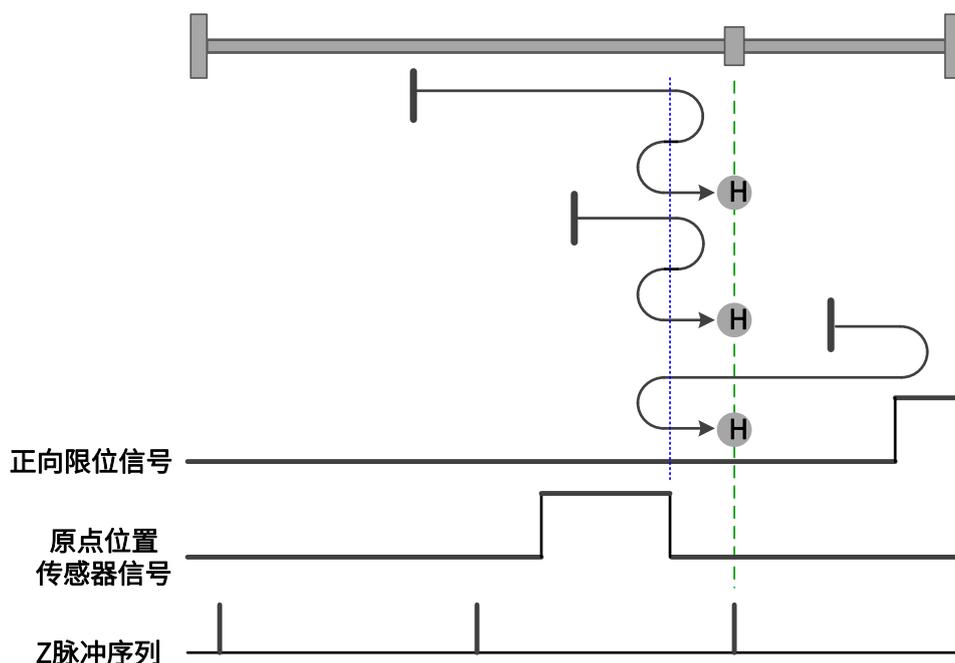
起步时 HSW 无效且位于原点位置传感器所在位置的右侧，则以高速朝正向运行，遇到 PL 的 ON 状态时减速停止，然后以高速朝负向运行。在负向运行时遇到 HSW 的 OFF→ON 状态之后减速停止，然后换低速朝正向运行。在低速正向运行时遇到 HSW 的 ON→OFF 状态之后，继续朝正向找最近的 Z 脉冲位置作为原点。

起步时 HSW 无效且位于原点位置传感器所在位置的左侧，则以高速朝正向运行，在正向运行时遇到 HSW 的 ON→OFF 状态之后减速停止，然后高速回退到 HSW 有效的位置之后再减速停止（如果 HSW 有效的区间很窄，则可能进入另一侧 HSW 无效的位置区间），此后换低速朝正向运行。在低速正向运行时遇到 HSW 的 ON→OFF 状态之后，继续朝正向找最近的 Z 脉冲位置作为原点。

起步时 HSW 有效则以高速朝正向运行。在正向运行时遇到 HSW 的 ON→OFF 状态之后减速停止，然后高速回退到 HSW 有效的位置之后再减速停止（如果 HSW 有效的区间很窄，则可能进入另一侧 HSW 无效的位置区间），此后换低速朝正向运行。在低速正向运行时遇到 HSW 的 ON→OFF 状态之后，继续朝正向找最近的 Z 脉冲位置作为原点。

这种模式下，朝正向运行第一次遇到 PL 的 ON 状态时自动反向；遇到 NL 的 ON 状态，或者再次遇到 PL 的 ON 状态，则停止回原点流程并报警。

如图附录 1-13 所示，参见表附录 1-6。



图附录 1-13 原点模式 10 轨迹及信号状态

### ■ 模式 11，寻找朝正向运行时 HSW 的 ON→OFF 位置和 Z 脉冲，遇负限位自动反向

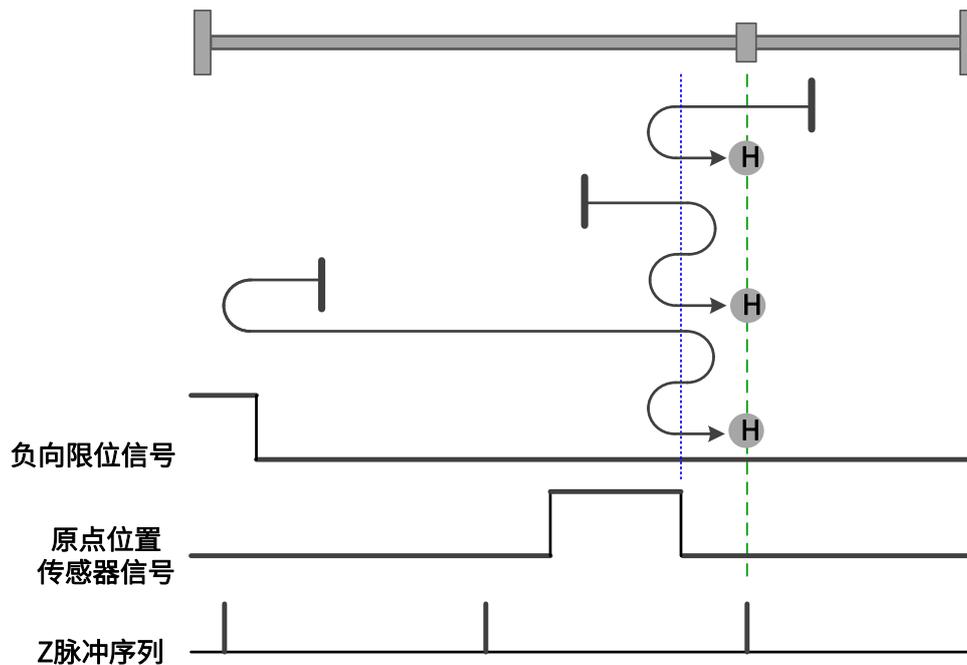
起步时 HSW 无效且位于原点位置传感器所在位置的右侧，则以高速朝负向运行，在负向运行时遇到 HSW 的 OFF→ON 状态之后减速停止，然后换低速朝正向运行。在低速正向运行时遇到 HSW 的 ON→OFF 状态之后，继续朝正向找最近的 Z 脉冲位置作为原点。

起步时 HSW 无效且位于原点位置传感器所在位置的左侧，则以高速朝负向运行，遇到 NL 的 ON 状态时减速停止，然后以高速朝正向运行。在正向运行时遇到 HSW 的 ON→OFF 状态之后减速停止，然后高速回退到 HSW 有效的位置之后再减速停止（如果 HSW 有效的区间很窄，则可能进入另一侧 HSW 无效的位置区间），此后换低速朝正向运行。在低速正向运行时遇到 HSW 的 ON→OFF 状态之后，继续朝正向找最近的 Z 脉冲位置作为原点。

起步时 HSW 有效则以高速朝正向运行。在正向运行时遇到 HSW 的 ON→OFF 状态之后减速停止，然后高速回退到 HSW 有效的位置之后再减速停止（如果 HSW 有效的区间很窄，则可能进入另一侧 HSW 无效的位置区间），此后换低速朝正向运行。在低速正向运行时遇到 HSW 的 ON→OFF 状态之后，继续朝正向找最近的 Z 脉冲位置作为原点。

这种模式下，朝负向运行第一次遇到 NL 的 ON 状态时自动反向；遇到 PL 的 ON 状态，或者再次遇到 NL 的 ON 状态，则停止回原点流程并报警。

如图附录 1-14 所示，参见表附录 1-6。



图附录1-14 原点模式11轨迹及信号状态

■ 模式 12，寻找朝负向运行时 HSW 的 OFF→ON 位置和 Z 脉冲，遇负限位自动反向

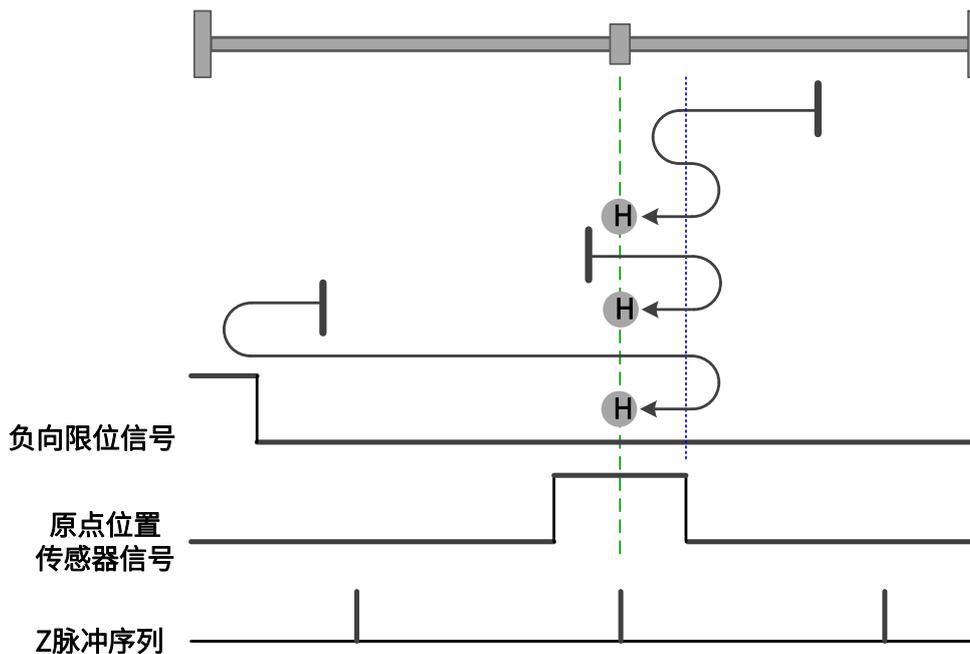
起步时HSW无效且位于原点位置传感器所在位置的右侧，则以高速朝负向运行，在负向运行时遇到HSW的OFF→ON状态之后减速停止，然后高速回退到HSW无效的位置之后再减速停止，此后换低速朝负向运行。在低速负向运行时遇到HSW的OFF→ON状态之后，继续朝负向找最近的Z脉冲位置作为原点。

起步时HSW无效且位于原点位置传感器所在位置的左侧，则以高速朝负向运行，遇到NL的ON状态时减速停止，然后以高速朝正向运行。在正向运行时遇到HSW的ON→OFF状态之后减速停止，然后换低速朝负向运行。在低速负向运行时遇到HSW的OFF→ON状态之后，继续朝负向找最近的Z脉冲位置作为原点。

起步时HSW有效则以高速朝正向运行。在正向运行时遇到HSW的ON→OFF状态之后减速停止，然后换低速朝负向运行。在低速负向运行时遇到HSW的OFF→ON状态之后，继续朝负向找最近的Z脉冲位置作为原点。

这种模式下，朝负向运行第一次遇到NL的ON状态时自动反向；遇到PL的ON状态，或者再次遇到NL的ON状态，则停止回原点流程并报警。

如图附录1-15所示，参见表附录1-6。



图附录1-15 原点模式12轨迹及信号状态

### ■ 模式 13, 寻找朝正向运行时 HSW 的 OFF→ON 位置和 Z 脉冲, 遇负限位自动反向

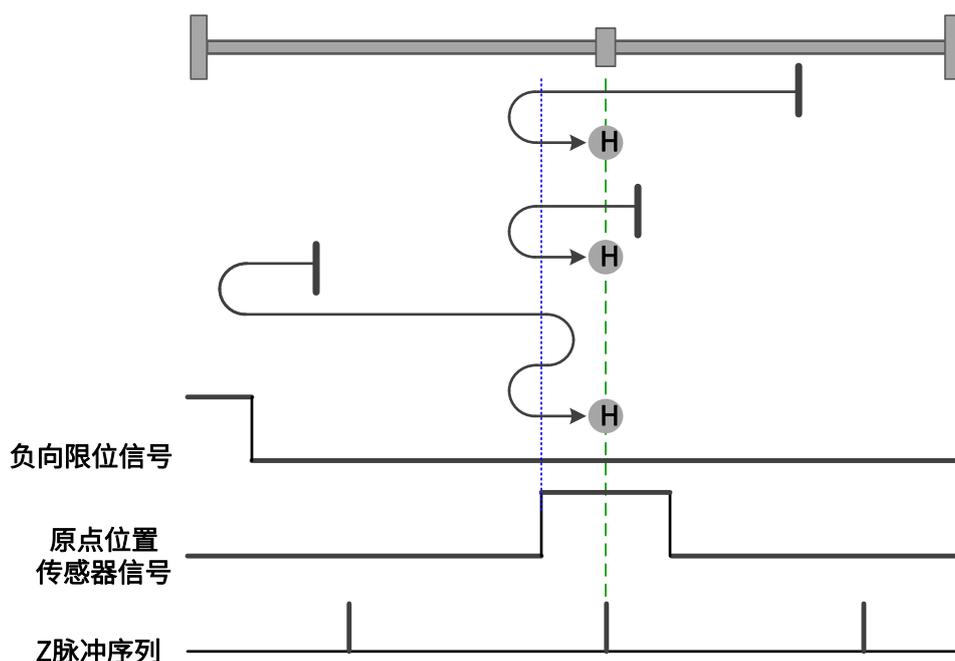
起步时HSW无效且位于原点位置传感器所在位置的右侧, 则以高速朝负向运行, 在负向运行时遇到HSW的ON→OFF状态之后减速停止, 然后换低速朝正向运行。在低速正向运行时遇到HSW的OFF→ON状态之后, 继续朝正向找最近的Z脉冲位置作为原点。

起步时HSW无效且位于原点位置传感器所在位置的左侧, 则以高速朝正向运行, 遇到NL的ON状态时减速停止, 然后以高速朝负向运行。在负向运行时遇到HSW的OFF→ON状态之后减速停止, 然后高速回退到HSW无效的位置之后再减速停止, 此后换低速朝正向运行。在低速正向运行时遇到HSW的OFF→ON状态之后, 继续朝正向找最近的Z脉冲位置作为原点。

起步时HSW有效则以高速朝正向运行。在负向运行时遇到HSW的ON→OFF状态之后减速停止, 然后换低速朝正向运行。在低速正向运行时遇到HSW的OFF→ON状态之后, 继续朝正向找最近的Z脉冲位置作为原点。

这种模式下, 朝负向运行第一次遇到NL的ON状态时自动反向; 遇到PL的ON状态, 或者再次遇到NL的ON状态, 则停止回原点流程并报警。

如图附录1-16所示, 参见表附录1-6



图附录1-16 原点模式13轨迹及信号状态

### ■ 模式 14, 寻找朝负向运行时 HSW 的 ON→OFF 位置和 Z 脉冲, 遇负限位自动反向

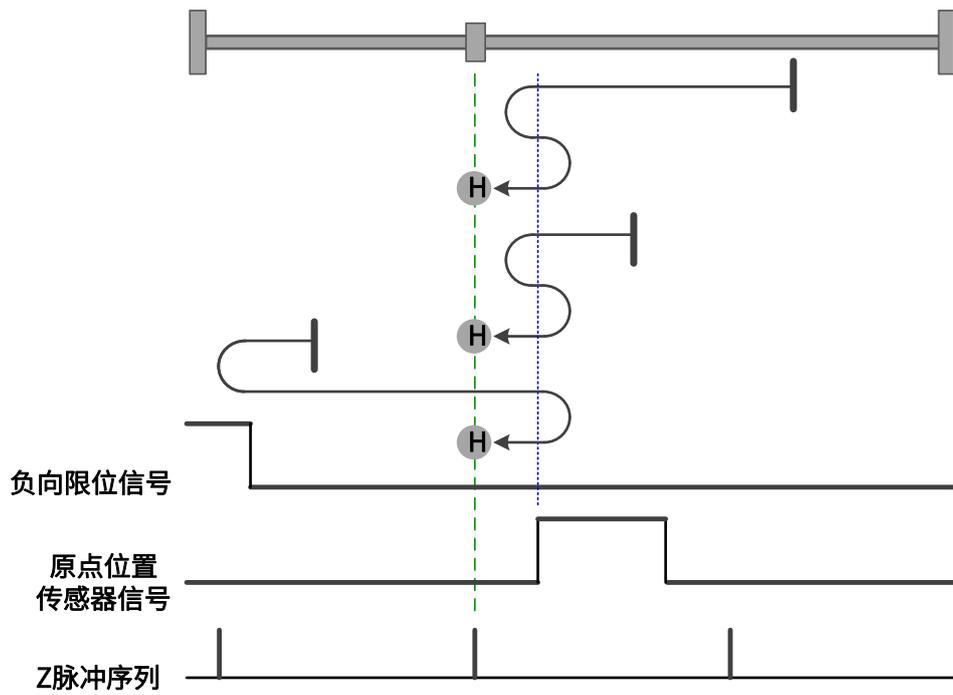
起步时HSW无效且位于原点位置传感器所在位置的右侧, 则以高速朝负向运行, 在负向运行时遇到HSW的ON→OFF状态之后减速停止, 然后高速回退到HSW有效的位置之后再减速停止 (如果HSW有效的区间很窄, 则可能进入另一侧HSW无效的位置区间), 此后换低速朝负向运行。在低速负向运行时遇到HSW的ON→OFF状态之后, 继续朝负向找最近的Z脉冲位置作为原点。

起步时HSW无效且位于原点位置传感器所在位置的左侧, 则以高速朝正向运行, 遇到NL的ON状态时减速停止, 然后以高速朝负向运行。在负向运行时遇到HSW的OFF→ON状态之后减速停止, 然后换低速朝负向运行。在低速负向运行时遇到HSW的ON→OFF状态之后, 继续朝负向找最近的Z脉冲位置作为原点。

起步时HSW有效则以高速朝负向运行。在负向运行时遇到HSW的ON→OFF状态之后减速停止, 然后高速回退到HSW有效的位置之后再减速停止 (如果HSW有效的区间很窄, 则可能进入另一侧HSW无效的位置区间), 此后换低速朝负向运行。在低速负向运行时遇到HSW的ON→OFF状态之后, 继续朝负向找最近的Z脉冲位置作为原点。

这种模式下, 朝负向运行第一次遇到NL的ON状态时自动反向; 遇到PL的ON状态, 或者再次遇到NL的ON状态, 则停止回原点流程并报警。

如图附录1-17所示, 参见表附录1-6。



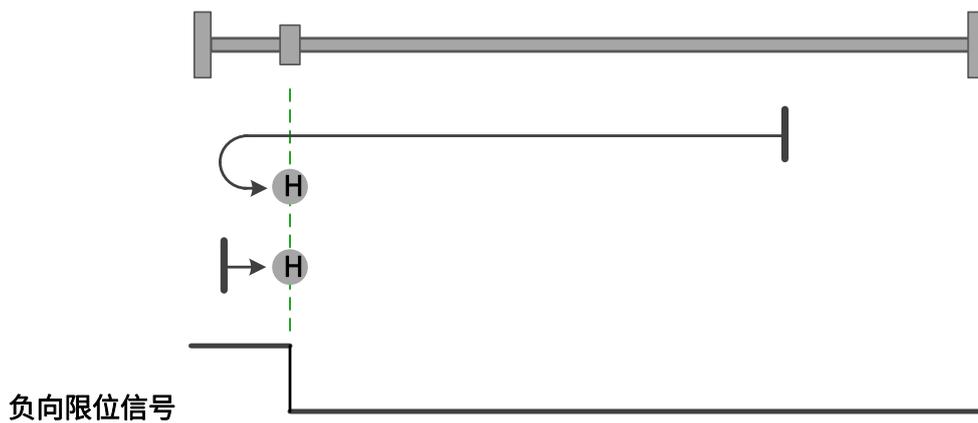
图附录1-17 原点模式14轨迹及信号状态

- 模式 15，保留，请不要设置。
- 模式 16，保留，请不要设置。
- 模式 17，寻找负限位

起步时如果NL无效，则以高速朝负向运行，遇到NL的OFF→ON状态之后减速停止，然后换低速朝正向运行。在低速朝正向运行遇到NL的ON→OFF状态时减速停止，以停止位置作为原点。

起步时如果NL有效，则以低速朝正向运行。在正向运行遇到的NL的ON→OFF状态时减速停止，以停止位置作为原点。

如图附录1-18所示，参见表附录1-6。



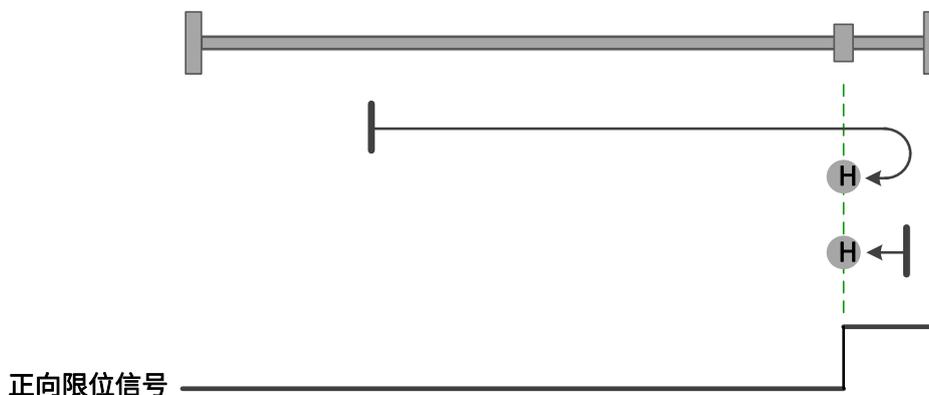
如图附录1-18 原点模式17轨迹及信号状态

### ■ 模式 18, 寻找正限位

起步时如果PL无效,则以高速朝正向运行,遇到PL的OFF→ON状态之后减速停止,然后换低速朝负向运行。在低速朝负向运行遇到PL的ON→OFF状态时减速停止,以停止位置作为原点。

起步时如果PL有效,则以低速朝负向运行。在低速朝负向运行遇到PL的ON→OFF状态时减速停止,以停止位置作为原点。

如图附录1-19所示,参见表附录1-6。



图附录1-19 原点模式18轨迹及信号状态

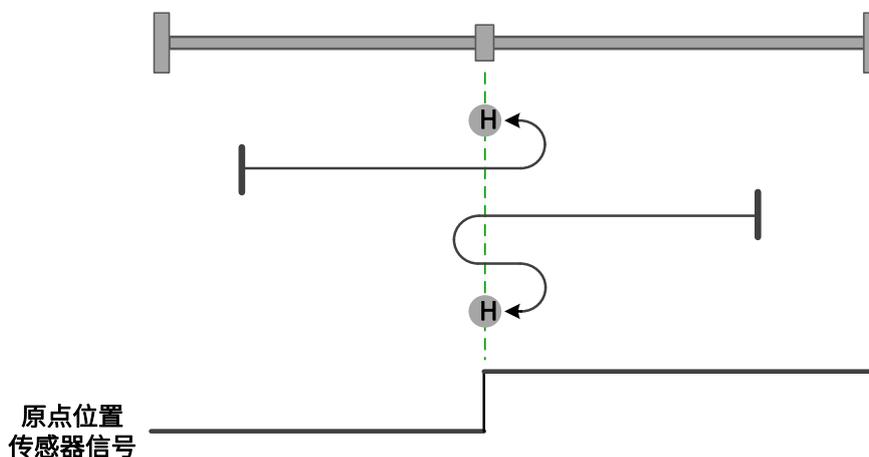
### ■ 模式 19, 寻找朝负向运行时 HSW 的 ON→OFF 位置

起步时HSW无效则以高速朝正向运行,在正向运行时遇到HSW的OFF→ON状态之后减速停止,然后换低速朝负向运行。在低速负向运行遇到HSW的ON→OFF状态时减速停止,以停止位置作为原点。

起步时HSW有效则以高速朝负向运行。在负向运行时遇到HSW的ON→OFF状态之后减速停止,然后高速回退到HSW有效的位置之后再减速停止,此后换低速朝负向运行。在低速负向运行遇到HSW的ON→OFF状态时减速停止,以停止位置作为原点。

这种模式下,无论遇到NL还是PL的ON状态,都是停止回原点流程并报警。

如图附录1-20所示,参见表附录1-6。



图附录1-20 原点模式19轨迹及信号状态

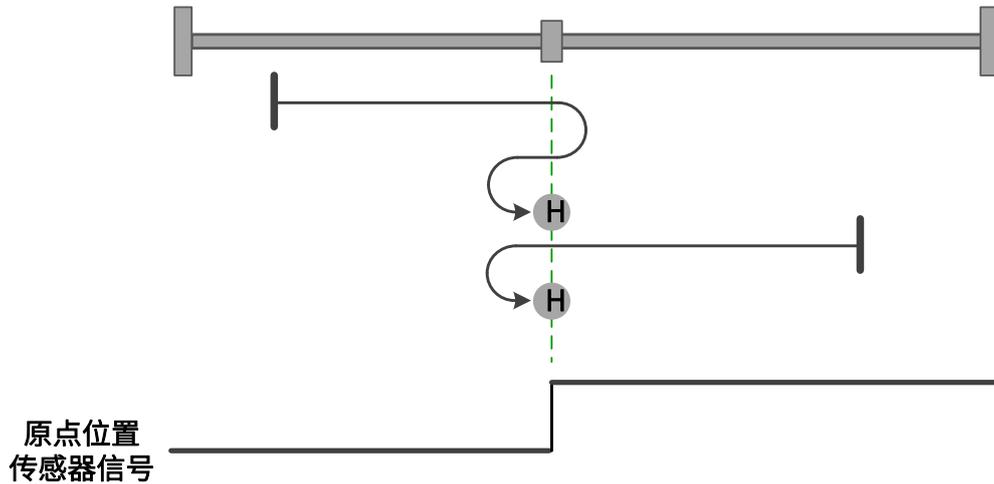
### ■ 模式 20，寻找朝正向运行时 HSW 的 OFF→ON 位置

起步时 HSW 无效则以高速朝正向运行，在正向运行时遇到 HSW 的 OFF→ON 状态之后减速停止，然后高速回退到 HSW 无效的位置之后再减速停止，此后换低速朝正向运行。在低速正向运行遇到 HSW 的 OFF→ON 状态时减速停止，以停止位置作为原点。

起步时 HSW 有效则以高速朝负向运行。在负向运行时遇到 HSW 的 ON→OFF 状态之后减速停止，然后换低速朝正向运行。在低速正向运行遇到 HSW 的 OFF→ON 状态时减速停止，以停止位置作为原点。

这种模式下，无论遇到 NL 还是 PL 的 ON 状态，都是停止回原点流程并报警。

如图附录 1-21 所示，参见表附录 1-6。



图附录 1-21 原点模式 20 轨迹及信号状态

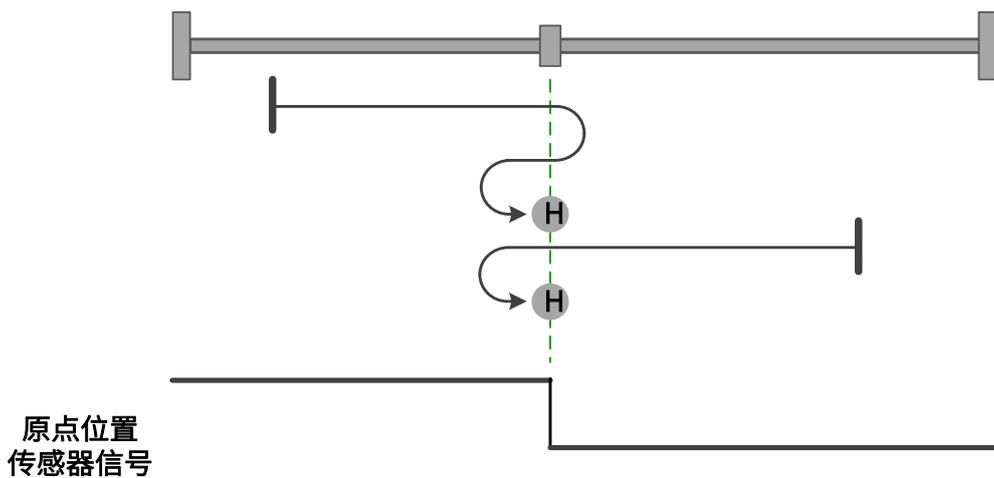
### ■ 模式 21，寻找朝正向运行时 HSW 的 ON→OFF 位置

起步时 HSW 无效则以高速朝负向运行。在负向运行时遇到 HSW 的 OFF→ON 状态之后减速停止，然后换低速朝正向运行。在低速正向运行遇到 HSW 的 ON→OFF 状态时减速停止，以停止位置作为原点。

起步时 HSW 有效则以高速朝正向运行，在正向运行时遇到 HSW 的 ON→OFF 状态之后减速停止，然后高速回退到 HSW 有效的位置之后再减速停止，此后换低速朝正向运行。在低速正向运行遇到 HSW 的 ON→OFF 状态时减速停止，以停止位置作为原点。

这种模式下，无论遇到 NL 还是 PL 的 ON 状态，都是停止回原点流程并报警。

如图附录 1-22 所示，参见表附录 1-6。



图附录 1-22 原点模式 21 轨迹及信号状态

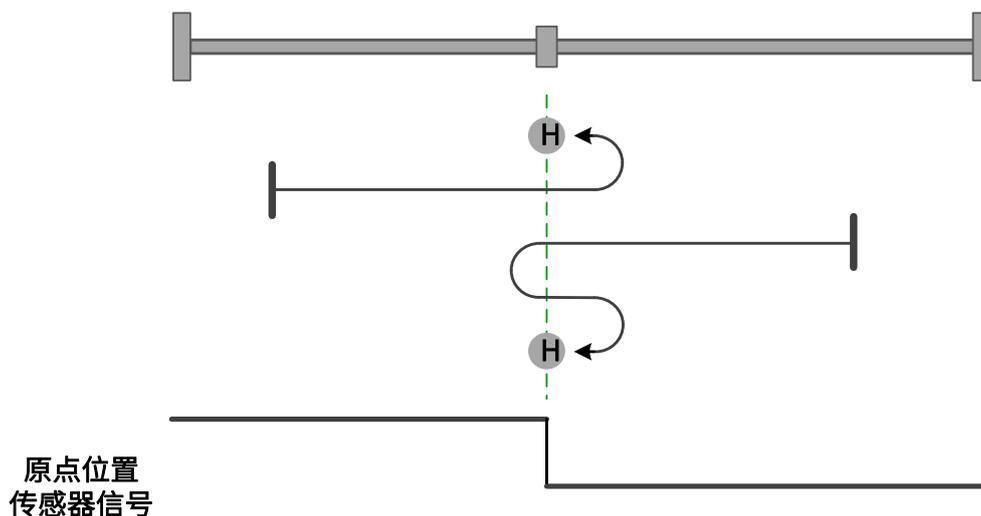
### ■ 模式 22，寻找朝负向运行时 HSW 的 OFF→ON 位置

起步时 HSW 无效则以高速朝负向运行。在负向运行时遇到 HSW 的 OFF→ON 状态之后减速停止，然后高速回退到 HSW 无效的位置之后再减速停止，此后换低速朝负向运行。在低速负向运行遇到 HSW 的 OFF→ON 状态时减速停止，以停止位置作为原点。

起步时 HSW 有效则以高速朝正向运行，在正向运行时遇到 HSW 的 ON→OFF 状态之后减速停止，然后换低速朝负向运行。在低速负向运行遇到 HSW 的 OFF→ON 状态时减速停止，以停止位置作为原点。

这种模式下，无论遇到 NL 还是 PL 的 ON 状态，都是停止回原点流程并报警。

如图附录 1-23 所示，参见表附录 1-6。



图附录 1-23 原点模式 22 轨迹及信号状态

### ■ 模式 23，寻找朝负向运行时 HSW 的 ON→OFF 位置，遇正限位自动反向

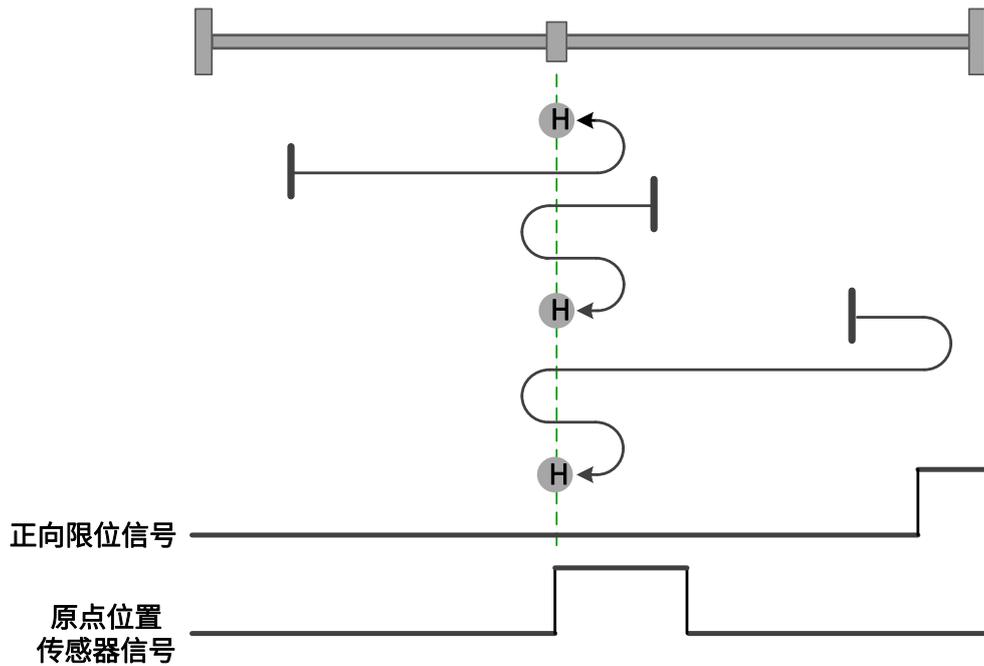
起步时 HSW 无效且位于原点位置传感器所在位置的右侧，则以高速朝正向运行，遇到 PL 的 ON 状态时减速停止，然后以高速朝负向运行。在负向运行时遇到 HSW 的 ON→OFF 状态之后减速停止，然后高速回退到 HSW 有效的位置之后再减速停止（如果 HSW 有效的区间很窄，则可能进入另一侧 HSW 无效的位置区间），此后换低速朝负向运行。在低速负向运行遇到 HSW 的 ON→OFF 时减速停止，以停止位置作为原点。

起步时 HSW 无效且位于原点位置传感器所在位置的左侧，则以高速朝正向运行，在正向运行时遇到 HSW 的 OFF→ON 状态之后减速停止，然后换低速朝负向运行。在低速负向运行遇到 HSW 的 ON→OFF 状态时减速停止，以停止位置作为原点。

起步时 HSW 有效则以高速朝负向运行。在负向运行时遇到 HSW 的 ON→OFF 状态之后减速停止，然后高速回退到 HSW 有效的位置之后再减速停止（如果 HSW 有效的区间很窄，则可能进入另一侧 HSW 无效的位置区间），此后换低速朝负向运行。在低速负向运行遇到 HSW 的 ON→OFF 时减速停止，以停止位置作为原点。

这种模式下，朝正向运行第一次遇到 PL 的 ON 状态时自动反向；遇到 NL 的 ON 状态，或者再次遇到 PL 的 ON 状态，则停止回原点流程并报警。

如图附录 1-24 所示，参见表附录 1-6。



图附录1-24 原点模式23轨迹及信号状态

■ **模式 24，寻找朝正向运行时 HSW 的 OFF→ON 位置，遇正限位自动反向**

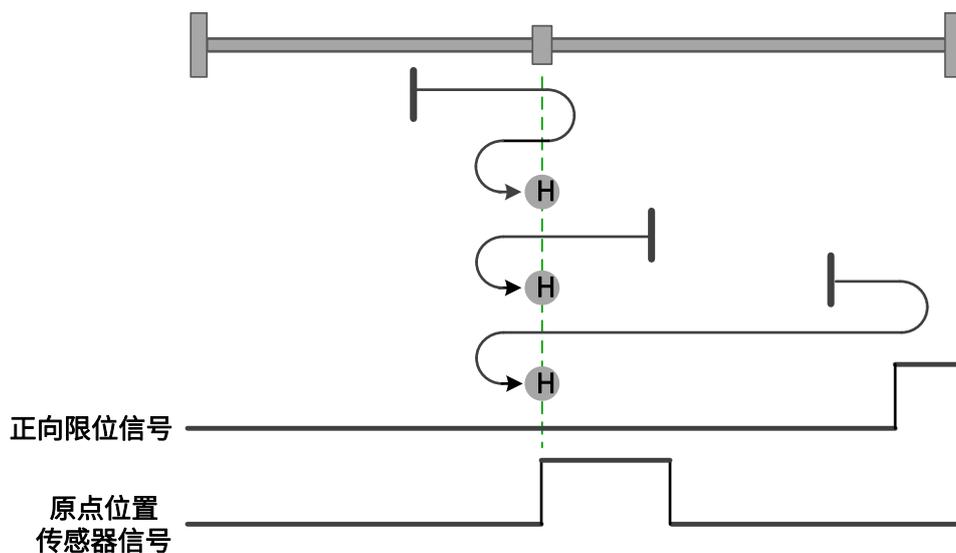
起步时HSW无效且位于原点位置传感器所在位置的右侧，则以高速朝正向运行，遇到PL的ON状态时减速停止，然后以高速朝负向运行。在负向运行时遇到HSW的ON→OFF状态之后减速停止，然后换低速朝正向运行。在低速正向运行遇到HSW的OFF→ON状态时减速停止，以停止位置作为原点。

起步时HSW无效且位于原点位置传感器所在位置的左侧，则以高速朝正向运行，在正向运行时遇到HSW的OFF→ON状态之后减速停止，然后高速回退到HSW无效的位置之后再减速停止，此后换低速朝正向运行。在低速正向运行遇到HSW的OFF→ON状态时减速停止，以停止位置作为原点。

起步时HSW有效则以高速朝负向运行。在负向运行时遇到HSW的ON→OFF状态之后减速停止，然后换低速朝正向运行。在低速正向运行遇到HSW的OFF→ON状态时减速停止，以停止位置作为原点。

这种模式下，朝正向运行第一次遇到PL的ON状态时自动反向；遇到NL的ON状态，或者再次遇到PL的ON状态，则停止回原点流程并报警。

如图附录1-25所示，参见表附录1-6。



图附录1-25 原点模式24轨迹及信号状态

### ■ 模式 25, 寻找朝负向运行时 HSW 的 OFF→ON 位置, 遇正限位自动反向

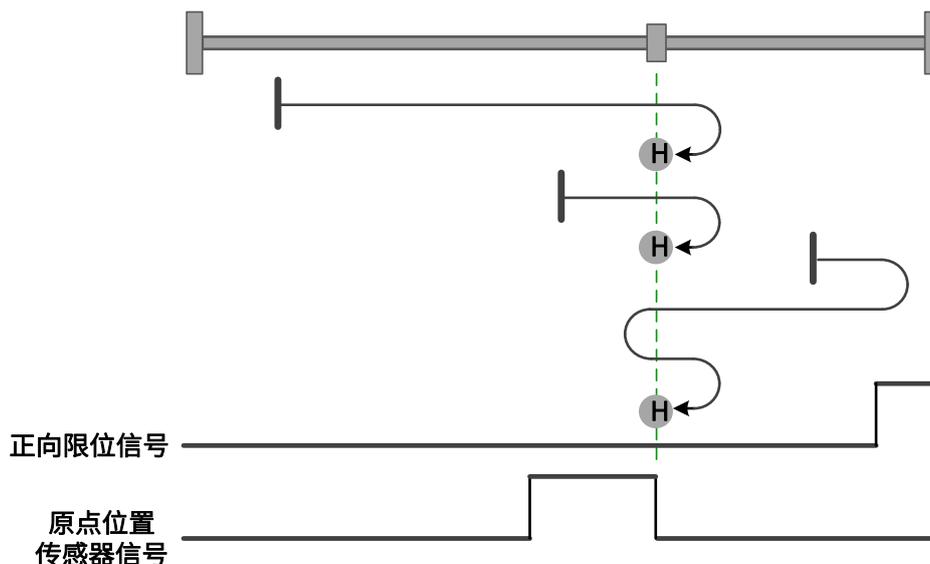
起步时 HSW 无效且位于原点位置传感器所在位置的正当侧, 则以高速朝正向运行, 遇到 PL 的 ON 状态时减速停止, 然后以高速朝负向运行。在负向运行时遇到 HSW 的 OFF→ON 状态之后减速停止, 然后高速回退到 HSW 无效的位置之后再减速停止, 此后换低速朝负向运行。在低速负向运行遇到 HSW 的 OFF→ON 状态时减速停止, 以停止位置作为原点。

起步时 HSW 无效且位于原点位置传感器所在位置的负向侧, 则以高速朝正向运行, 在正向运行时遇到 HSW 的 ON→OFF 状态之后减速停止, 然后换低速朝负向运行。在低速负向运行遇到 HSW 的 OFF→ON 状态时减速停止, 以停止位置作为原点。

起步时 HSW 有效则以高速朝正向运行。在正向运行时遇到 HSW 的 ON→OFF 状态之后减速停止, 然后换低速朝负向运行。在低速负向运行遇到 HSW 的 OFF→ON 状态时减速停止, 以停止位置作为原点。

这种模式下, 朝正向运行第一次遇到 PL 的 ON 状态时自动反向; 遇到 NL 的 ON 状态, 或者再次遇到 PL 的 ON 状态, 则停止回原点流程并报警。

如图附录 1-26 所示, 参见表附录 1-6。



图附录 1-26 原点模式 25 轨迹及信号状态

### ■ 模式 26, 寻找朝正向运行时 HSW 的 ON→OFF 位置, 遇正限位自动反向

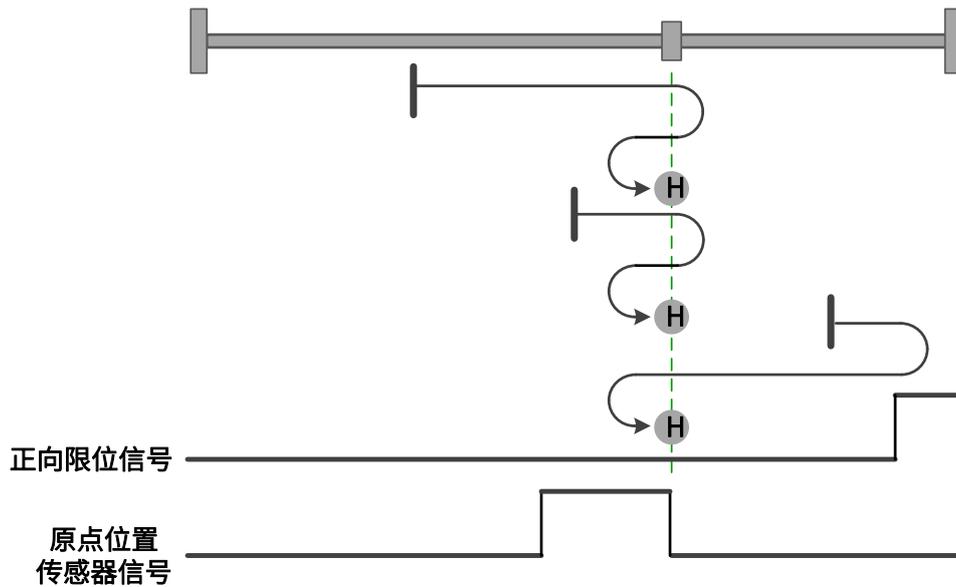
起步时 HSW 无效且位于原点位置传感器所在位置的正当侧, 则以高速朝正向运行, 遇到 PL 的 ON 状态时减速停止, 然后以高速朝负向运行。在负向运行时遇到 HSW 的 OFF→ON 状态之后减速停止, 然后换低速朝正向运行。在低速正向运行遇到 HSW 的 ON→OFF 状态时减速停止, 以停止位置作为原点。

起步时 HSW 无效且位于原点位置传感器所在位置的负向侧, 则以高速朝正向运行, 在正向运行时遇到 HSW 的 ON→OFF 状态之后减速停止, 然后高速回退到 HSW 有效的位置之后再减速停止 (如果 HSW 有效的区间很窄, 则可能进入另一侧 HSW 无效的位置区间), 此后换低速朝正向运行。在低速正向运行遇到 HSW 的 ON→OFF 状态时减速停止, 以停止位置作为原点。

起步时 HSW 有效则以高速朝正向运行。在正向运行时遇到 HSW 的 ON→OFF 状态之后减速停止, 然后高速回退到 HSW 有效的位置之后再减速停止 (如果 HSW 有效的区间很窄, 则可能进入另一侧 HSW 无效的位置区间), 此后换低速朝正向运行。在低速正向运行遇到 HSW 的 ON→OFF 状态时减速停止, 以停止位置作为原点。

这种模式下, 朝正向运行第一次遇到 PL 的 ON 状态时自动反向; 遇到 NL 的 ON 状态, 或者再次遇到 PL 的 ON 状态, 则停止回原点流程并报警。

如图附录 1-27 所示, 参见表附录 1-6。



图附录1-27 原点模式26轨迹及信号状态

■ 模式 27，寻找朝正向运行时 HSW 的 ON→OFF 位置，遇负限位自动反向

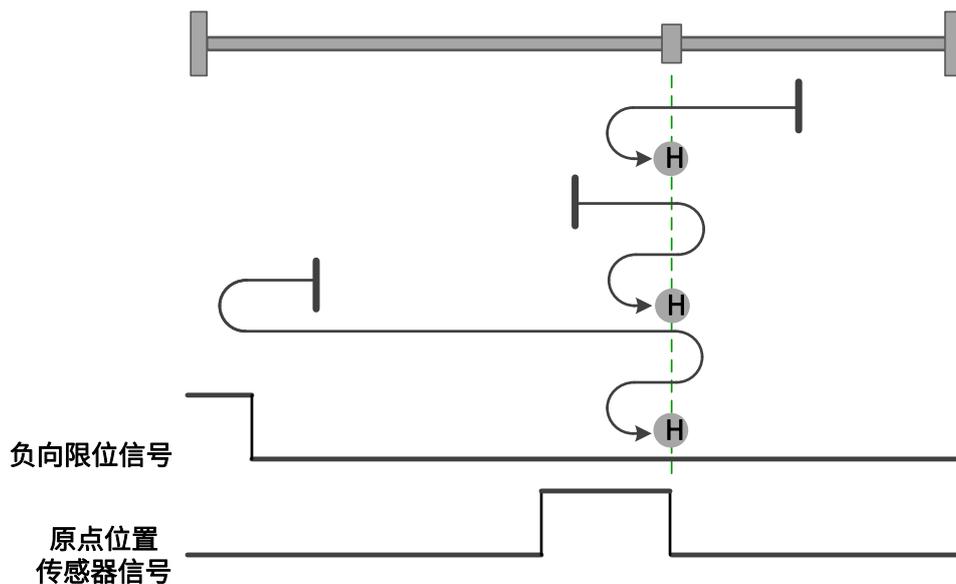
起步时HSW无效且位于原点位置传感器所在位置的正向侧，则以高速朝负向运行，在负向运行时遇到HSW的OFF→ON状态之后减速停止，然后换低速朝正向运行。在低速正向运行遇到HSW的ON→OFF状态时减速停止，以停止位置作为原点。

起步时HSW无效且位于原点位置传感器所在位置的负向侧，则以高速朝负向运行，遇到NL的ON状态时减速停止，然后以高速朝正向运行。在正向运行时遇到HSW的ON→OFF状态之后减速停止，然后高速回退到HSW有效的位置之后再减速停止（如果HSW有效的区间很窄，则可能进入另一侧HSW无效的位置区间），此后换低速朝正向运行。在低速正向运行遇到HSW的ON→OFF状态时减速停止，以停止位置作为原点。

起步时HSW有效则以高速朝正向运行。在正向运行时遇到HSW的ON→OFF状态之后减速停止，然后高速回退到HSW有效的位置之后再减速停止（如果HSW有效的区间很窄，则可能进入另一侧HSW无效的位置区间），此后换低速朝正向运行。在低速正向运行遇到HSW的ON→OFF状态时减速停止，以停止位置作为原点。

这种模式下，朝负向运行第一次遇到NL的ON状态时自动反向；遇到PL的ON状态，或者再次遇到NL的ON状态，则停止回原点流程并报警。

如图附录1-28所示，参见表附录1-6。



图附录1-28 原点模式27轨迹及信号状态

### ■ 模式 28, 寻找朝负向运行时 HSW 的 OFF→ON 位置, 遇负限位自动反向

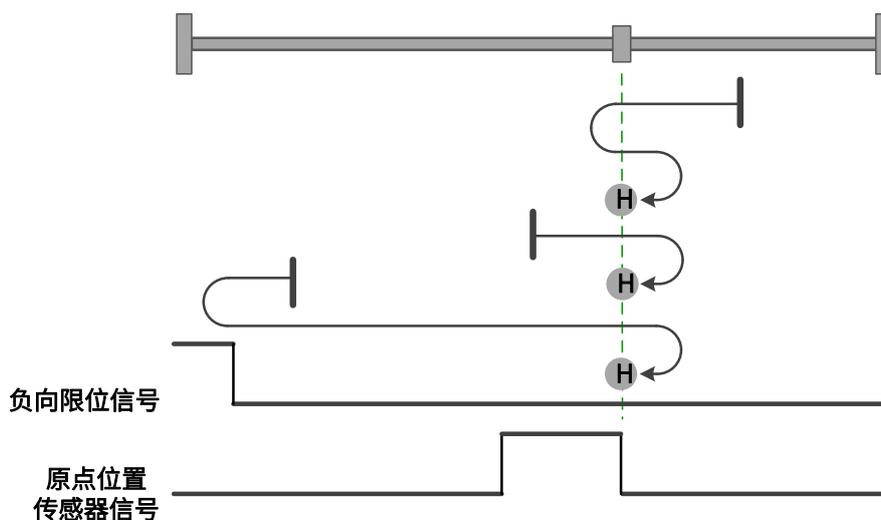
起步时 HSW 无效且位于原点位置传感器所在位置的正向侧, 则以高速朝负向运行, 在负向运行时遇到 HSW 的 OFF→ON 状态之后减速停止, 然后高速回退到 HSW 无效的位置之后再减速停止, 此后换低速朝负向运行。在低速负向运行遇到 HSW 的 OFF→ON 状态时减速停止, 以停止位置作为原点。

起步时 HSW 无效且位于原点位置传感器所在位置的负向侧, 则以高速朝负向运行, 遇到 NL 的 ON 状态时减速停止, 然后以高速朝正向运行。在正向运行时遇到 HSW 的 ON→OFF 状态之后减速停止, 然后换低速朝负向运行。在低速负向运行遇到 HSW 的 OFF→ON 状态时减速停止, 以停止位置作为原点。

起步时 HSW 有效则以高速朝正向运行。在正向运行时遇到 HSW 的 ON→OFF 状态之后减速停止, 然后换低速朝负向运行。在低速负向运行遇到 HSW 的 OFF→ON 状态时减速停止, 以停止位置作为原点。

这种模式下, 朝负向运行第一次遇到 NL 的 ON 状态时自动反向; 遇到 PL 的 ON 状态, 或者再次遇到 NL 的 ON 状态, 则停止回原点流程并报警。

如图附录 1-29 所示, 参见表附录 1-6。



图附录 1-29 原点模式 28 轨迹及信号状态

### ■ 模式 29, 寻找朝正向运行时 HSW 的 OFF→ON 位置, 遇负限位自动反向

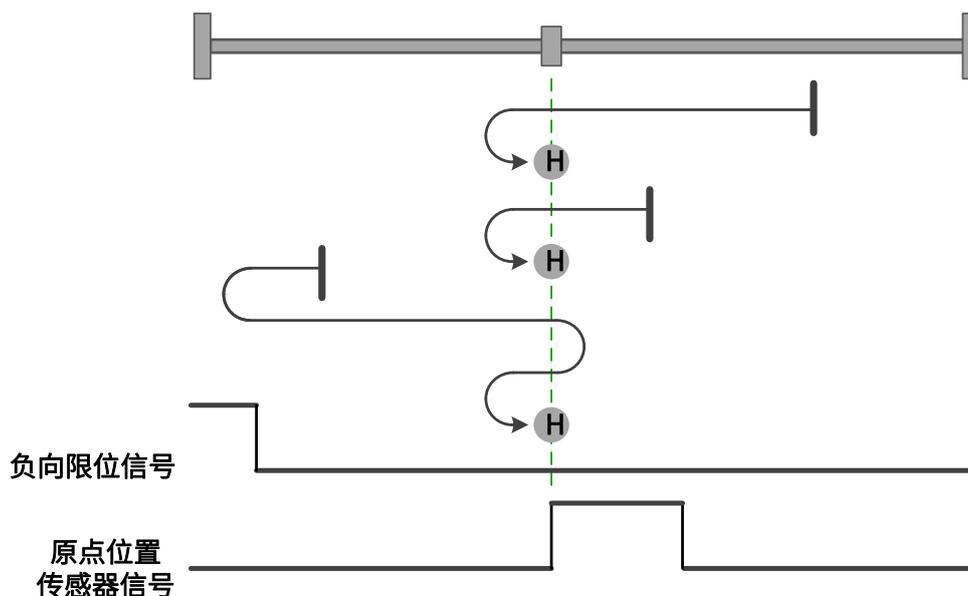
起步时 HSW 无效且位于原点位置传感器所在位置的正向侧, 则以高速朝负向运行, 在负向运行时遇到 HSW 的 ON→OFF 状态之后减速停止, 然后换低速朝正向运行。在低速正向运行遇到 HSW 的 OFF→ON 状态时减速停止, 以停止位置作为原点。

起步时 HSW 无效且位于原点位置传感器所在位置的负向侧, 则以高速朝负向运行, 遇到 NL 的 ON 状态时减速停止, 然后以高速朝正向运行。在正向运行时遇到 HSW 的 OFF→ON 状态之后减速停止, 然后高速回退到 HSW 无效的位置之后再减速停止, 此后换低速朝正向运行。在低速正向运行遇到 HSW 的 OFF→ON 状态时减速停止, 以停止位置作为原点。

起步时 HSW 有效则以高速朝负向运行。在负向运行时遇到 HSW 的 ON→OFF 状态之后减速停止, 然后换低速朝正向运行。在低速正向运行遇到 HSW 的 OFF→ON 状态时减速停止, 以停止位置作为原点。

这种模式下, 朝负向运行第一次遇到 NL 的 ON 状态时自动反向; 遇到 PL 的 ON 状态, 或者再次遇到 NL 的 ON 状态, 则停止回原点流程并报警。

如图附录 1-30 所示, 参见表附录 1-6。



图附录1-30 原点模式29轨迹及信号状态

■ 模式 30，寻找朝负向运行时 HSW 的 ON→OFF 位置，遇负限位自动反向

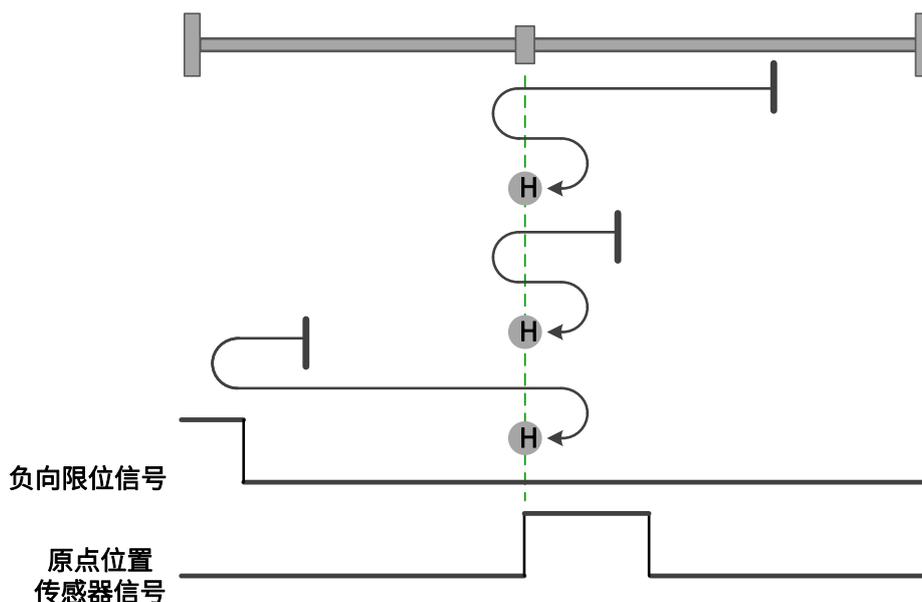
起步时HSW无效且位于原点位置传感器所在位置的正向侧，则以高速朝负向运行，在负向运行时遇到HSW的ON→OFF状态之后减速停止，然后高速回退到HSW有效的位置之后再减速停止（如果HSW有效的区间很窄，则可能进入另一侧HSW无效的位置区间），此后换低速朝负向运行。在低速负向运行遇到HSW的ON→OFF状态时减速停止，以停止位置作为原点。

起步时HSW无效且位于原点位置传感器所在位置的负向侧，则以高速朝负向运行，遇到NL的ON状态时减速停止，然后以高速朝正向运行。在正向运行时遇到HSW的OFF→ON状态之后减速停止，然后换低速朝负向运行。在低速负向运行遇到HSW的ON→OFF状态时减速停止，以停止位置作为原点。

起步时HSW有效则以高速朝负向运行。在负向运行时遇到HSW的ON→OFF状态之后减速停止，然后高速回退到HSW有效的位置之后再减速停止（如果HSW有效的区间很窄，则可能进入另一侧HSW无效的位置区间），此后换低速朝负向运行。在低速负向运行遇到HSW的ON→OFF状态时减速停止，以停止位置作为原点。

这种模式下，朝负向运行第一次遇到NL的ON状态时自动反向；遇到PL的ON状态，或者再次遇到NL的ON状态，则停止回原点流程并报警。

如图附录1-31所示，参见表附录1-6。



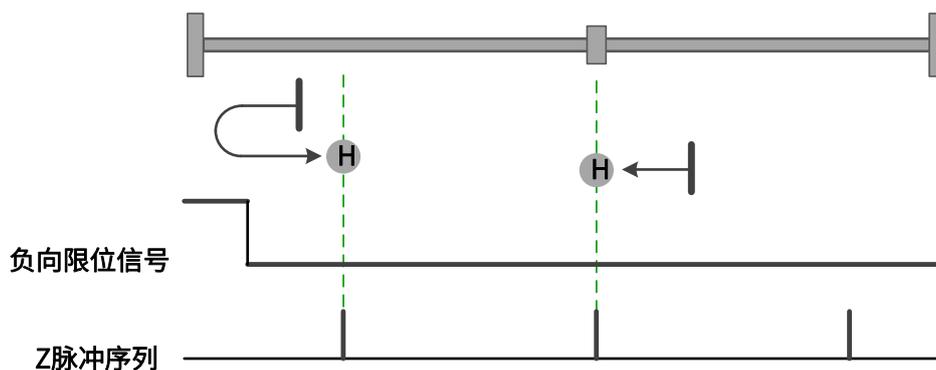
图附录1-31 原点模式30轨迹及信号状态

- 模式 31，保留，请不要设置。
- 模式 32，保留，请不要设置。
- 模式 33，寻找负向运行时最近的 Z 脉冲

起步时以低速朝负向找最近的Z脉冲位置作为原点。如果朝负向运行在找到Z脉冲之前就遇到NL的ON状态，则减速停止，然后朝正向运行找最近的Z脉冲位置作为原点。

这种模式下，朝负向运行第一次遇到NL的ON状态时自动反向；遇到PL的ON状态，或者再次遇到NL的ON状态，则停止回原点流程并报警。

如图附录1-32所示，参见表附录1-6。



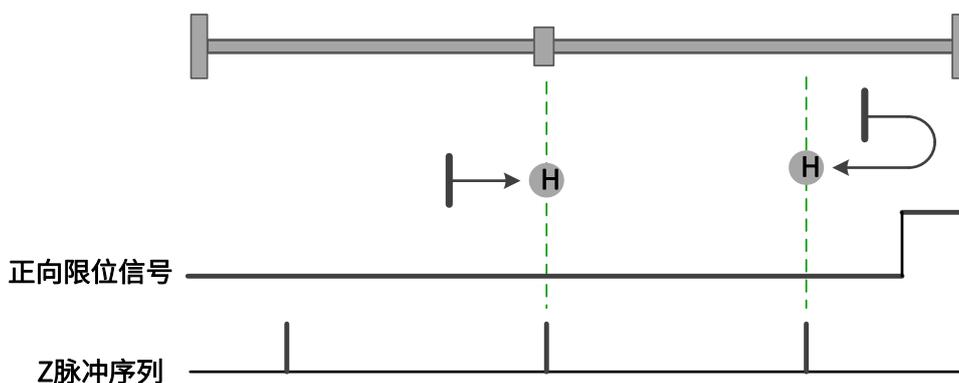
图附录1-32 原点模式33轨迹及信号状态

- 模式 34，寻找正向运行时最近的 Z 脉冲

起步时以低速朝正向找最近的Z脉冲位置作为原点。如果朝正向运行在找到Z脉冲之前就遇到PL的ON状态，则减速停止，然后朝负向运行找最近的Z脉冲位置作为原点。

这种模式下，朝正向运行第一次遇到PL的ON状态时自动反向；遇到NL的ON状态，或者再次遇到PL的ON状态，则停止回原点流程并报警。

如图附录1-33所示，参见表附录1-6。



图附录1-33 原点模式34轨迹及信号状态

## 附录2 指令错误代码描述

当指令执行出错时，指令ErrorID（错误代码）有对应的值，通过下表查询ErrorID值的含义及处理方法。

错误代码		ErrorID值含义	处理方法
十六进制	十进制		
1001	4097	指令操作的轴未在软件中配置	检查指令操作的轴有在软件中配置
1002	4098	编码器轴不能执行此操作	编码器轴不能执行此操作
1003	4099	轴类型或捕获模式不正确	检查是否是编码器轴及轴类型
1004	4100	轴范围超出	请根据控制器支持轴范围设定
1005	4101	加速度超出范围	检查参数数值是否在合法范围
1006	4102	减速度超出范围	检查参数数值是否在合法范围
1007	4103	加加速度超出范围	检查参数数值是否在合法范围
1008	4104	速度超出范围	检查参数数值是否在合法范围
1009	4105	位置超出范围	MC_MoveAbsolute输入Position的数值不能超过轴配置中模数的数值
100A	4106	方向超出范围	检查参数数值是否在合法范围
100B	4107	交接模式（BufferMode）超出范围	检查参数数值是否在合法范围
100C	4108	速度超调值超出范围	检查参数数值是否在合法范围
100D	4109	参考类型输入错误	检查参数数值是否在合法范围
100E	4110	指令不能执行	MC_Home、MC_TorqueControlWithVelocity指令执行时，此指令不能执行
100F	4111	指令不能执行	轴在standstill状态时，可以执行此指令
1010	4112	MC_Reset执行失败	检查轴类型及驱动器状态
1011	4113	指令执行会导出致软件限位	修改指令执行位置
1012	4114	MC_Home 指令Posiotn写入失败	检查参数数值是否在合法范围
1013	4115	轴类型错误	此轴不能执行MC_Home指令
1014	4116	捕获指令捕获触发位输入错误	检查参数数值是否在合法范围
1015	4117	捕获指令获触发位已使用	修改TriggerInput的值为尚未使用的触发位
1016	4118	捕获功能窗口功能错误	修改窗口设置上下限范围
1017	4119	一个轴同时只能执行一个捕获指令	检查参数数值是否在合法范围
1018	4120	捕获指令模式设定错误	检查参数数值是否在合法范围
1019	4121	捕获指令不能执行	轴断线不能执行捕获指令
1100	4352	电子凸轮ID超出范围	检查参数数值是否在合法范围
1101	4353	主轴超出范围	检查参数数值是否在合法范围
1102	4354	啮合模式超出范围	检查参数数值是否在合法范围
1103	4355	主轴位置缩放比超出范围	检查参数数值是否在合法范围
1104	4356	从轴位置缩放比超出范围	检查参数数值是否在合法范围
1105	4357	主轴位置来源选择超出范围	检查参数数值是否在合法范围
1106	4358	主从轴轴号冲突	主从轴号不能相同
1107	4359	电子齿轮分子超出范围	检查参数数值是否在合法范围
1108	4360	电子齿轮分母超出范围	检查参数数值是否在合法范围
1109	4361	凸轮啮合功能激活位置超出范围	检查参数数值是否在合法范围
110A	4362	主轴位置偏移量超过主轴凸轮周期范围	合法范围为主轴周期范围的负值到正值之间（主轴凸轮周期=主轴最大值-主轴最小值）

错误代码		ErrorID值含义	处理方法
十六进制	十进制		
110B	4363	从轴位置偏移量超过从轴凸轮周期范围	合法范围为从轴周期范围的负值到正值之间（从轴凸轮周期=从轴最大值-从轴最小值）
110C	4364	凸轮曲线未在软件中建立	检查CamTable对应的凸轮曲线在软件中是否建立
110D	4365	旋切轴半径超出范围	检查参数数值是否在合法范围
110E	4366	进给轴半径超出范围	检查参数数值是否在合法范围
110F	4367	旋切长度超出范围	检查参数数值是否在合法范围
1110	4368	同步区开始位置超出范围	检查参数数值是否在合法范围
1111	4369	同步区结束位置超出范围	检查参数数值是否在合法范围
1112	4370	同步区同步位置超出范围	检查参数数值是否在合法范围
1113	4371	旋切编号超出范围	检查参数数值是否在合法范围
1114	4372	旋切刀数超出范围	检查参数数值是否在合法范围
1115	4373	旋切内部状态非法	修改旋切初始化参数
1116	4374	旋切功能未初始化	请先执行旋切功能初始化功能
1200	4608	预读深度超出范围	检查参数数值是否在合法范围
1201	4609	合成速度超调超出范围	检查参数数值是否在合法范围
1202	4610	GCode文件编码超出范围	检查参数数值是否在合法范围
1203	4611	GCode文件为空	检查文件内容是否合法
1204	4612	GCode文件解析错误	检查文件内容是否合法
1205	4613	轴组编号超出范围	检查参数数值是否在合法范围
1206	4614	模式超出范围	检查参数数值是否在合法范围
120A	4618	GCode指令格式错误	检查并确保GCode指令的格式符合要求
1301	4865	SDO 超时	检查通讯口连接是否正常，波特率是否一致
1302	4866	SDO输入变量错误	检查参数数值是否在合法范围
1303	4867	SDO 其它错误	检查从站是否处于正常工作状态。
1401	5121	运动方向限制	检查MC_Power指令EnablePositive和EnableNegative设置
1403	5123	不能执行MC_HaltSuperimposed	执行MC_MoveSuperimposed指令时，方可执行MC_HaltSuperimposed指令
1501	5377	状态机限制不能执行此功能	请参考指令状态机执行规则
1502	5378	BufferMode缓存超出	运动指令BufferMode只支持一级运动交接，请使用前一指令Busy触发后一指令启动，以此实现多级运动交接
1503	5379	指令不能支持Buffer 功能	该指令无法进行BufferMode操作
1504	5380	轴异常去使能	请检查驱动器状态
1601	5633	轴类型设置超出范围	修改轴配置界面中轴类型
1602	5634	驱动器报警	清除伺服报警再进行控制
1603	5635	驱动器通讯超时	检查控制器和伺服之间连线是否正常
1604	5636	软件限位超出	检查轴运动状态及软件限位设置是否合理
1606	5638	位置跟随检测超出	检查轴运动状态及位置跟随检测设置是否合理
4000	16384	扩展模块编号超出范围	将扩展模块编号(参数ModularID)设置在允许范围内
4001	16385	主机与扩展模块通讯超时	检查并确保主机与扩展模块连接正常
4002	16386	主机与扩展模块通讯错误	1.检查并确保主机与扩展模块连接正常 2.避免将产品置于强干扰的环境中

错误代码		ErrorID值含义	处理方法
十六进制	十进制		
4100	16640	脉冲输出通道编号设置错误	正确设置脉冲输出通道编号(参数ChannelNum)
4101	16641	脉冲输出通道已被使用	检查脉冲通道使用情况, 避免重复使用
4120	16672	脉冲输入边沿检测设置错误	正确设置脉冲输入边沿检测(参数EdgeSelect)
4121	16673	脉冲输入模式设置错误	正确设置脉冲输入模式(参数SignalMode)
4122	16674	脉冲输入计数方式设置错误	正确设置脉冲输入计数方式(参数CounterDirection)
4123	16675	脉冲输入倍频模式设置错误	正确设置脉冲输入计数倍频方式(参数Multi_Frequency)
4200	16896	Stop过程中再次触发Stop	修改Stop执行时机
4201	16897	在MC_StopAtPhase未控制轴运动的情形下触发Stop	修改Stop执行时机
4202	16898	参数RoundPhase设置错误	修改并确保参数RoundPhase的值大于0
4203	16899	参数StopPhase设置错误	修改并确保参数StopPhase的值大于0且小于RoundPhase
4220	16928	参数TorqueRamp设置错误	修改并确保参数TorqueRamp的值大于0
4240	16960	参数ScaleDen或ScaleNum设置错误	修改并确保参数ScaleDen、ScaleNum均大于0
4241	16961	参数UnitsPerTurn设置错误	修改并确保参数UnitsPerTurn大于0
4242	16962	参数AxisType设置错误	修改并确保参数AxisType在允许的范围内
4243	16963	参数Modulo设置错误	修改并确保参数Modulo大于0
4260	16992	参数Lag设置错误	修改并确保参数Lag大于或等于0
4261	16993	参数HoldTime设置错误	修改并确保参数HoldTime大于或等于0
4280	17024	参数Index(凸轮关键点编号)设置错误	修改并确保参数Index大于0且不大于该凸轮表中的关键点总数量
4281	17025	参数Index(凸轮挺杆编号)设置错误	修改并确保参数Index大于0且不大于该凸轮表中的挺杆总数量
4282	17026	参数MasterPos(主轴位置) 设置错误	修改并确保参数MasterPos大于0且不大于该凸轮表主轴位置最大值
4283	17027	挺杆数量超出最大值	在挺杆数量未达最大值时执行
4290	17040	参数MaxNum超出范围	修改并确保参数MaxNum大于0且不大于20
4291	17041	参数Precision超出范围	修改并确保参数Precision大于或者等于0
4500	17664	从站NodeID超出范围	确保从站NodeID在允许的范围内
4501	17665	参数Mode设置错误	修改并确保参数Mode在允许的范围内
4502	17666	诊断的从站不存在(未配置)	对已配置的从站执行诊断功能
4503	17667	本机CANOpen端口不是主站模式	在主站模式下执行诊断功能
4510	17680	从站NodeID超出范围	确保从站NodeID在允许的范围内
4511	17681	诊断的从站不存在(未配置)	对已配置的从站执行诊断功能
5000	20480	轴组状态机不允许执行	在允许的状态机下执行
5001	20481	参数IdentNum设置错误	修改并确保参数IdentNum在允许的范围内
5002	20482	参数IdentNum设定值在该轴组中已被使用	修改并确保参数IdentNum与该轴组中已使用的其它IdentNum不重复
5003	20483	参数Axis指定的轴已属于其它轴组	修改参数Axis
5004	20484	轴组未建立	在轴组已建立情形下执行本功能
5005	20485	属于轴组的各轴的状态不允许执行本功能	在各轴均为Standstill情形下执行本功能
5006	20486	参数TransitionMode设置错误	修改并确保参数TransitionMode在允许的范围内
5007	20487	参数TransitionParameter设置错误	修改并确保参数TransitionParameter在允许的范围内
5008	20488	参数BufferMode与参数TransitionMode冲突	修改并确保参数值在允许的范围内

错误代码		ErrorID值含义	处理方法
十六进制	十进制		
5009	20489	参数CircMode设置错误	修改并确保参数CircMode在允许的范围内
5010	20496	参数PathChoice设置错误	修改并确保参数PathChoice在允许的范围内
5100	20736	MCode的编号错误	确保MCode的编号在允许的范围内
5200	20992	参数MoveMode设置错误	修改并确保参数MoveMode在允许的范围内
5201	20993	参数Mode设置错误	修改并确保参数Mode在允许的范围内
5300	21248	参数HomeMode设置错误	修改并确保参数HomeMode在允许的范围内
5301	21249	参数VelocityFast设置错误	修改并确保参数VelocityFast在允许的范围内
5302	21250	参数VelocitySlow设置错误	修改并确保参数VelocitySlow在允许的范围内
5303	21251	参数Acceleration设置错误	修改并确保参数Acceleration在允许的范围内
5304	21252	参数Deceleration设置错误	修改并确保参数Deceleration在允许的范围内
5305	21253	参数Jerk设置错误	修改并确保参数Jerk在允许的范围内



HCFA



HCFA\_ATC