DECOWELL®
德克威尔自动化

工业信号处理专家

AX-3000系列运动控制器

运动控制用户手册

德克威尔 • 工业智造可靠伙伴

网址：www.welllinkio.com
邮箱：sales@welllinkio.com
地址：南京市浦口区兰新路19号瑞创智造园13号楼

全国服务热线:400-096-9016

前 言

资料简介

感谢您购买德克威尔 AX-3000 运动控制器！

AX-3000 系列运动控制器是高性能 AX3000，拥有小体积、模块化、高性能 等优点。其拥有强大的运动控制能力，可带 32 个伺服轴，同时提供各种应用库，包括基于 EtherCAT 的运动控制库，运动控制包括点对点运动，凸轮运动，插补运动等。另外集成了 Modbus TCP Master/Slave、Modbus RTU Master/Slave，且硬件上包含了 RS485、USB 等串口。支持本体高速计数、拓展 EX 系列 IO 等。

本手册主要对产品的运动控制进行详细说明

目 录

安全注意事项

安全声明

01. 在安装、操作、维护产品时，请先阅读并遵守本安全注意事项。

02. 为保障人身和设备安全，在安装、操作和维护产品时，请遵循产品上的标识及手册中说明的所有安全注意事项。

03. 手册中的“提示”、“注意”、“警告”和“危险”事项，并不代表所应遵循的所有安全事项，只作为所有安全注意事项的补充。

04. 本产品应在符合设计规格要求的环境下使用，否则可能造成故障，因未遵循相关规定引发的功能异常或部件损坏等不在产品质量保证范围之内。

05. 因违规操作产品引发的人身安全事故、财产损失等，德克威尔不承担任何法律责任。

安全等级定义

提示

该标记表示 “对操作的描述进行必要的补充或说明”。

注意

该标记“未按要求操作造成的危险，会导致人身轻度或中度伤害和设备损坏”。

警告

该标记表示“由于没有按要求操作造成的危险，可能导致人身伤亡”。

控制系统设计时警告

01.应用时请务必设计安全电路，保证当外部电源掉电或扩展模块故障时，控制系统依然能安全工作；

02.输出电路中由于超过额定负载电流或者负载短路等导致长时间过电流时，模块可能冒烟或着火，应在外部设置保险丝或断路器等安全装置。

控制系统设计时注意

01.务必在扩展模块的外部电路中设置紧急制动电路、保护电路、正反转操作的互锁电路和防止机器损坏的位置上限、下限互锁开关；

02.为使设备能安全运行，对于重大事故相关的输出信号，请设计外部保护电路和安全机构；

03.扩展模块的继电器、晶体管等输出单元损坏时，会使其输出无法控制为 ON 或 OFF 状态；

04.扩展模块设计应用于室内、过电压等级 II 级的电气环境，其电源系统级应有防雷保护装置，确保雷击过电压不施加于扩展模块的电源输入端或信号输入端、控制输出端等端口，避免损坏设备。

一、PLCopen 规范简介

IEC61131 是一部针对通用可编程控制器 PLC 的国际标准，当时是由欧洲几家 PLC 技术领先的公司首先发起拟定的一个行业标准，其中的 IEC 61131-3 是针对 PLC 编程的国际标准规格，先后定义了 6 种编程语言标准。

PLCopen 是总部在欧洲的 IEC 61131-3 推广团体，是一个全球性的会员组织，该组织中的若干知名 PLC 产品厂家对其中的一些技术细节作了完善，目的是为了使得不同 PLC 厂家的产品应用编程通用化，消除产品技术差异和壁垒，在用户选用用不同品牌 PLC 时，不必另外学习相应的编程方法。

在中国，与其对应的国家推荐标准 GBT15969.3 先后于 1995、2005 年发布和升级，作为 PLC 设备厂家的设计推荐标准。在中国也有相应的 PLCopen 推广组织。

PLCopen 规范除了对通用逻辑控制指令、程序结构、各种语言所含的关键字进行了规范建议外，还对运动控制功能块 MC 的技术规格进行了约定，包括 MC 功能模块的命名、具体功能、输入输出变量定义、相关时序逻辑等，最大限度地保证用户编程技术的互通。

AX3000 控制器选用的是德国 3S 公司的 CODESYS 编程平台，该平台完整支持 PLCopen 规范，用户可以引用许多标准的功能函数库；高级语言的编程方式，易于 PLC 厂家和用户开发自己专有的功能块和指令库，借用已有的类似控制程序，形成行业特点的“工艺包”，可显著提高用户编程效率。

二、运动控制程序组成

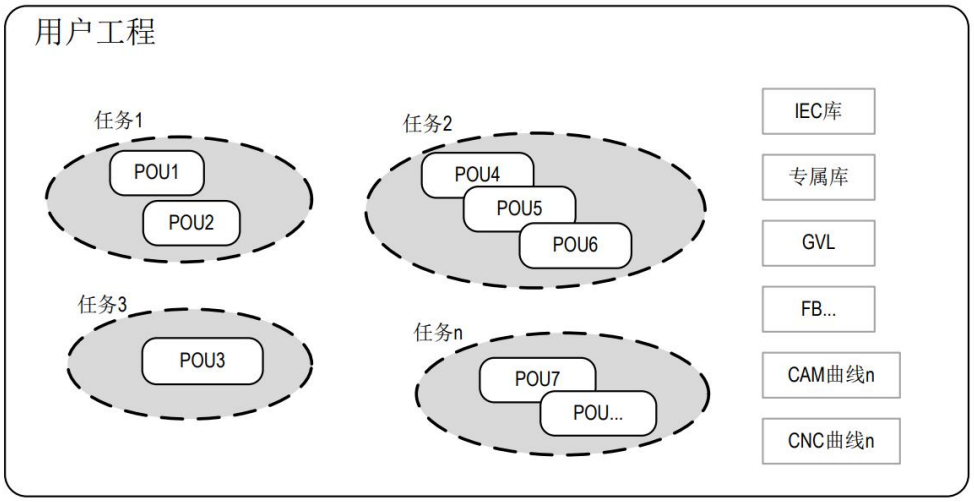
2.1 用户程序结构

AX3000 是基于多任务操作系统开发的控制器，系统运行的多个功能模块以多任务执行的方式，对于用户程序，也可分为多个任务组成，根据用户设定的任务优先级，分别执行。

编写 AX3000 的用户程序时，用户可根据应用系统中不同处理的业务类型和紧急程度，分成若干个程序组织单元组成，可将每个任务指定不同的执行触发条件，或相应的执行时间间隔（也称执行周期），这样可以让应用系统的控制响应达到最佳状态。

2.1.1 用户程序组成

正如前面介绍，AX3000 可采用多任务的执行模式，即可以“同时”执行几个任务，每个任务可以有若干个用户程序组织单元（简称 POU），典型的构成举例如下图：



用户工程是由若干个 POU 组成，按照 POU 执行特性要求，分为若干个任务组，配置其执行特性，没有列入任务配置的 POU 将不会被执行的。

另外用户工程中，还有一些支撑用户程序的对象，如库函数、全局变量 GVL、功能块 FB、凸轮定义 CAM 曲线、多轴插补轨迹定义 CNC 曲线等等，作为用户程序的组成部分。

2.1.2 任务类型

任务配置是将用户程序按执行要求，分成若干个任务组，每个任务组可以设置不同的执行触发条件、执行时间间隔、优先级等。

AX3000 常见的任务有：EtherCAT 任务、HSIO 高速中断任务、主循环任务等，其中，运动控制相关的用户程序主体，是安排在 EtherCAT 任务下执行。

其中的 EtherCAT 任务，在 AX3000 中是一个最为重要的任务，运动控制功能的实时处理，都是在这个任务中完成的，它是一个执行时间间隔短、优先级最高的一个时钟中断型的任务，一旦满足时间条件，它可以无条件地中断其他的任务，开始执行 EtherCAT 任务，直到该任务配置下所有的 POU 执行完毕，才会退出。

2.1.3 用户程序由多个 POU 组成的好处

不同执行周期的处理程序，应放在不同的 POU 中编写。比如按 EtherCAT 周期执行的 POU、外部中断程序 POU、按 20ms 时间处理的程序 POU，就必需分成独立的 POU 来编写；

为提高程序的可读性，按不同控制工艺段、不同的操作对象、不同的物理结构部件等，分别用不同的 POU 来 处理，每个 POU 分别命名为易于理解的名称；

如同 C 语言编程，将一个反复调用的处理程序做成一个独立的 POU，方便本工程调用；甚至还方便其他工程沿用；

多人合作编程时，每个编程人员各自编写调试自己负责的工艺段的 POU，最后合成为一个用户程序项目；

CodeSys 编程软件支持 6 种编程语言，根据所需的处理逻辑类型不同，某种的语言可能更方便，而一般情况下，每个 POU 只能用一种编程语言来进行编写，一个项目中若需要同时使用多种编程语言的话，分为多个 POU 来编写也是一个比较好的对策。

2.1.4 用户程序如何同时做到逻辑控制与运动控制

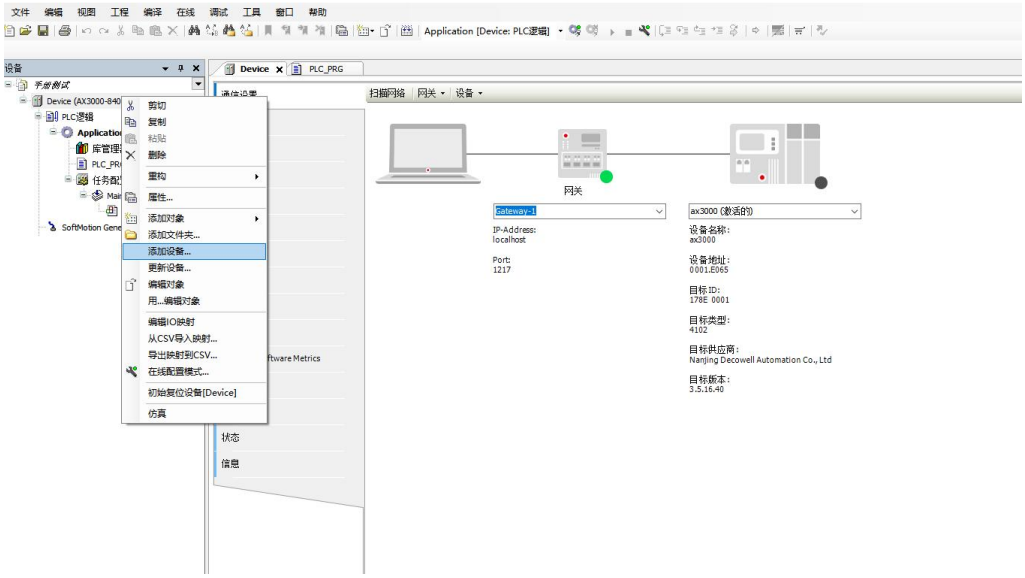
应用系统的同步控制、轨迹控制，往往都有高实时性要求，而逻辑控制的及时性要求则相对较低，在 AX3000 用户程序中，可以将运动控制（MC）的 POU 放在 EtherCAT 任务周期中执行，而逻辑控制 POU 就放在普通任务周期中执行即可。若将特定的程序变量声明为全局变量，就可以在运动控制中，实现与逻辑控制的协调动作。

对于主要以伺服驱动器+马达为控制对象的单轴 MC 控制，需要有伺服的使能、原点回归、定位控制、速度控制、力矩控制、停止与复位；对于多轴同步 MC 控制的应用，如凸轮控制、轨迹插补控制等，控制器提供有对应的 MC 功能块来完成这些操作。因此，功能块是运动控制编程中常用的控制命令，就如建筑中的采用预制件 代替砂石水泥，以提高施工效率一样。

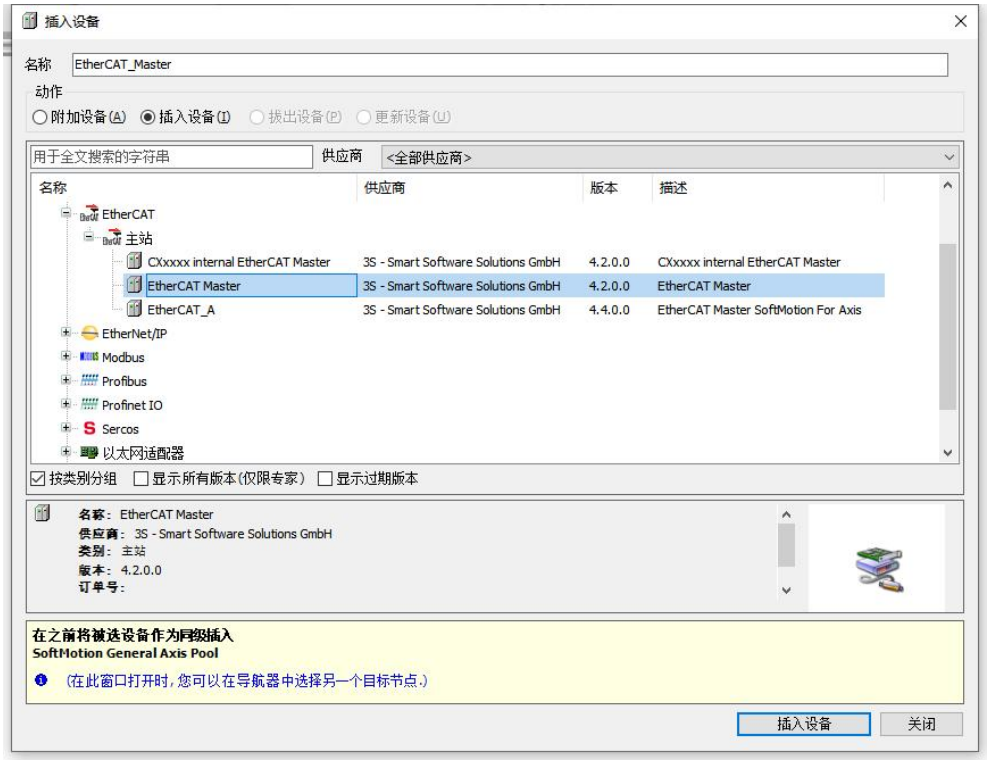
用户程序可以根据应用系统的控制逻辑进行，控制功能块的运行触发、终止执行等，同时可对功能块的执行 状态、是否出错等进行判断；在 PLCopen 规范中，还引入了轴状态数据结构，控制器系统为用户已经配置的 每一个伺服轴建立了一个对应的数据结构，并自动在每个 EtherCAT 周期中及时对其状态进行更新，用户程序可以通过访问该数据结构的变量，就可对伺服轴的运行状态进行监控，将状态变量作为逻辑控制的依据， 这样就使得逻辑控制与运动控制在一个用户程序中得以轻松实现。

2.2 一个简单的用户程序编写调试过程

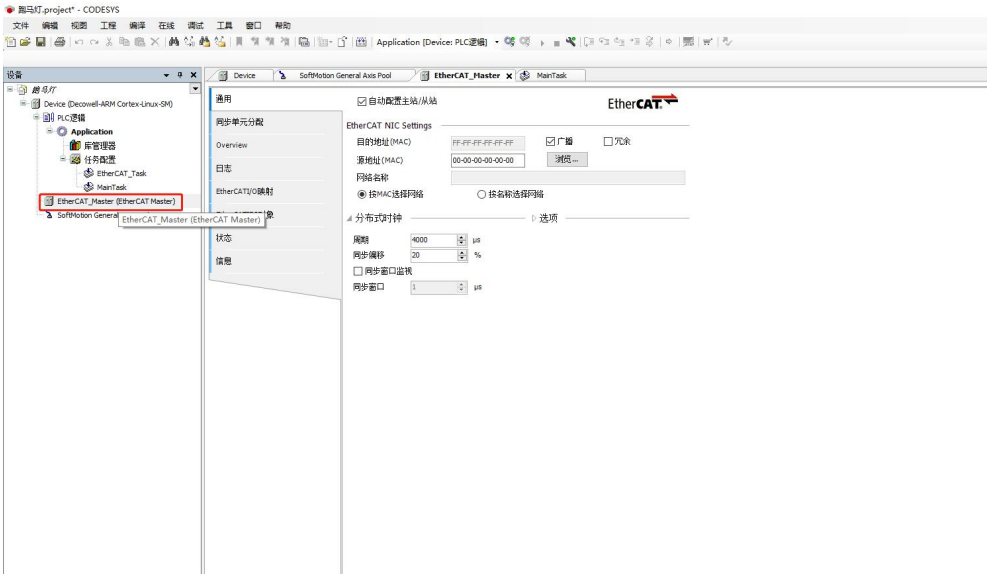
运行 CodeSys 编程软件，新建一个用户工程，在如下图的画面中，右击 Device（AX3000-8400-X），选择添加设备。



在添加设备弹框中选择现场总线-EtherCat-主站-EtherCat Master，然后点击添加设备

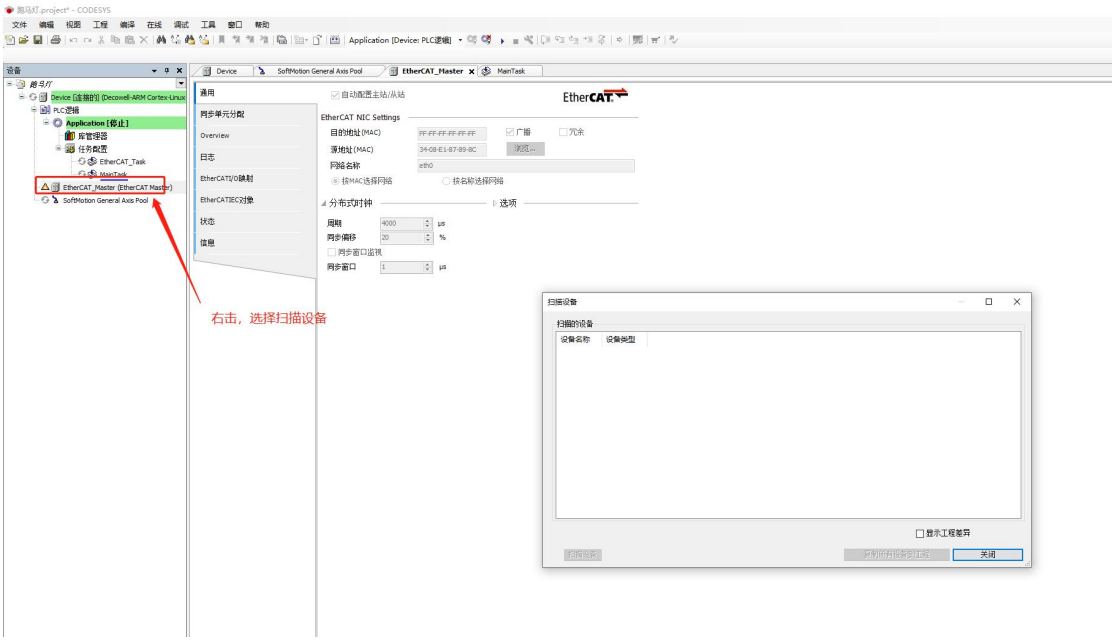


连接上 PLC 之后，双击 EtherCat_Master，在 EtherCat_Master 页面的通用中浏览源地址

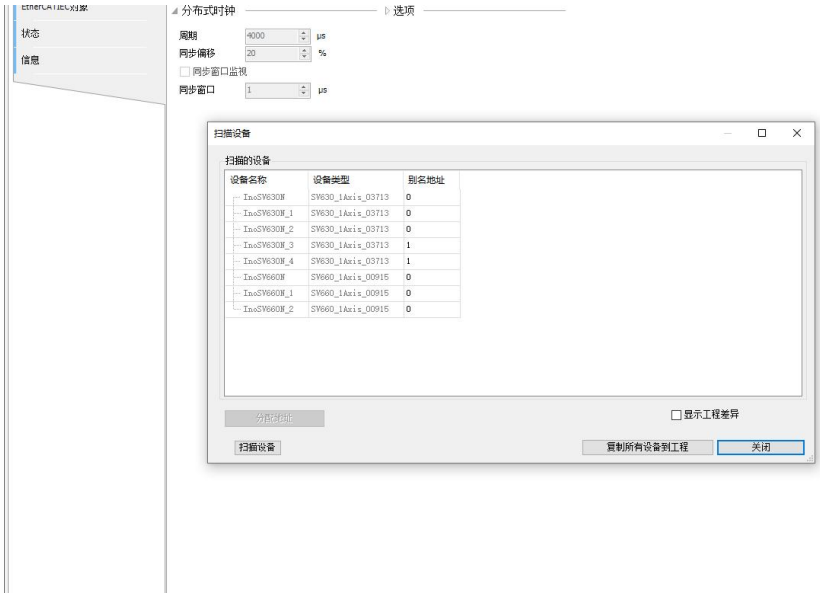


在选择网络适配器中选择 eth0，点击确定

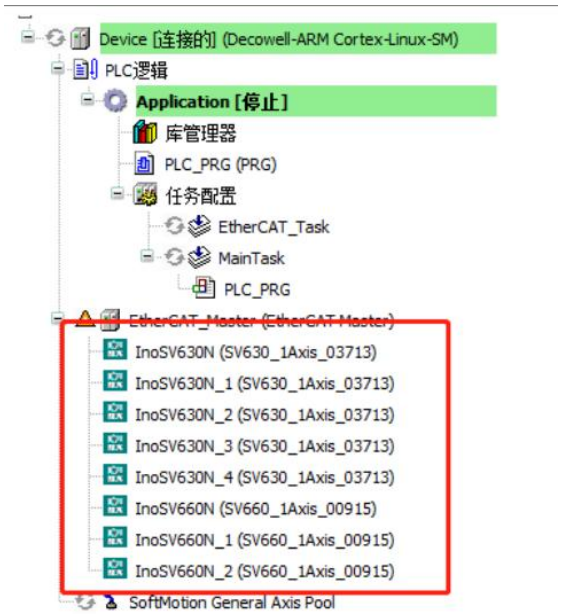
使用 Codesys 登录 AX3000，然后右击 EtherCat_Master 扫描设备



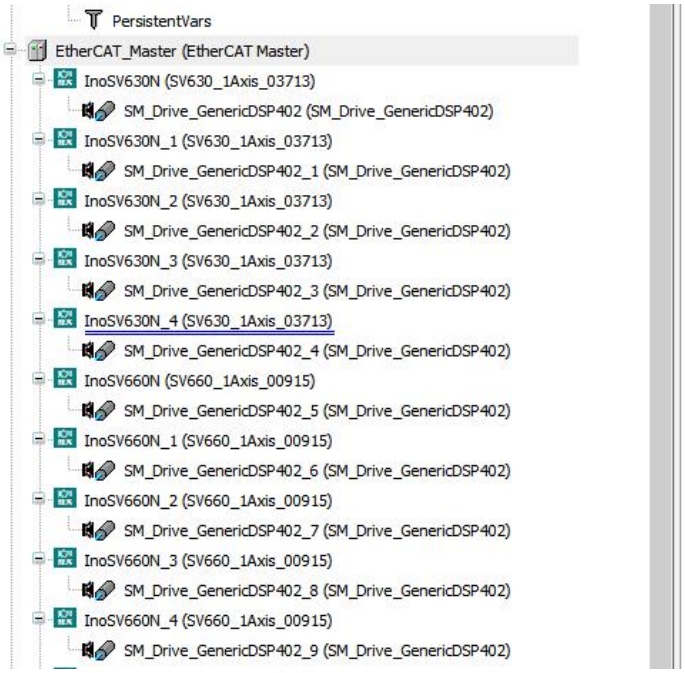
扫描设备框出现组态到的伺服轴设备，点击复制所有设备到工程



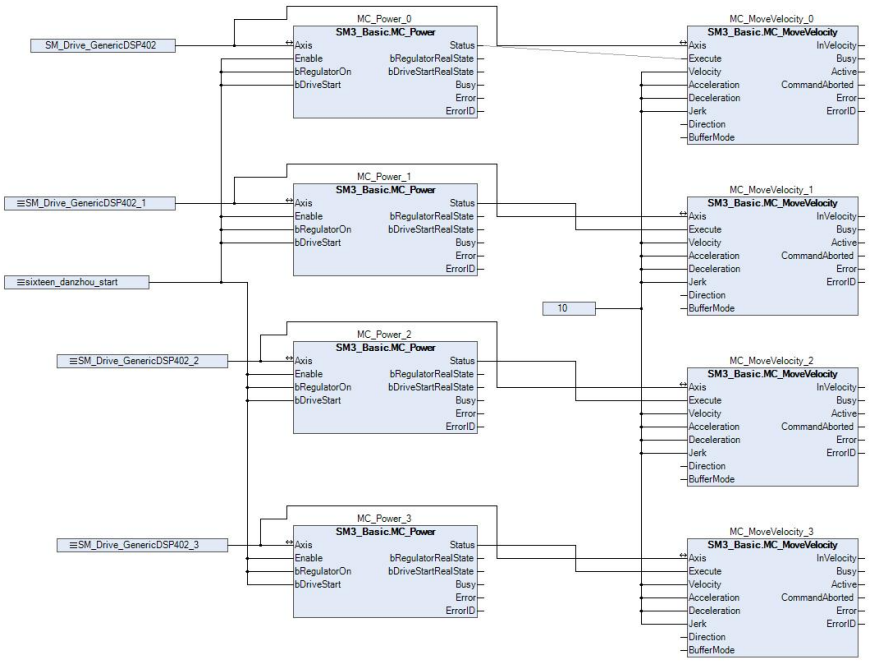
EtherCat_Master 下显示组态到的设备信息



退出登录，在伺服下右击添加设备

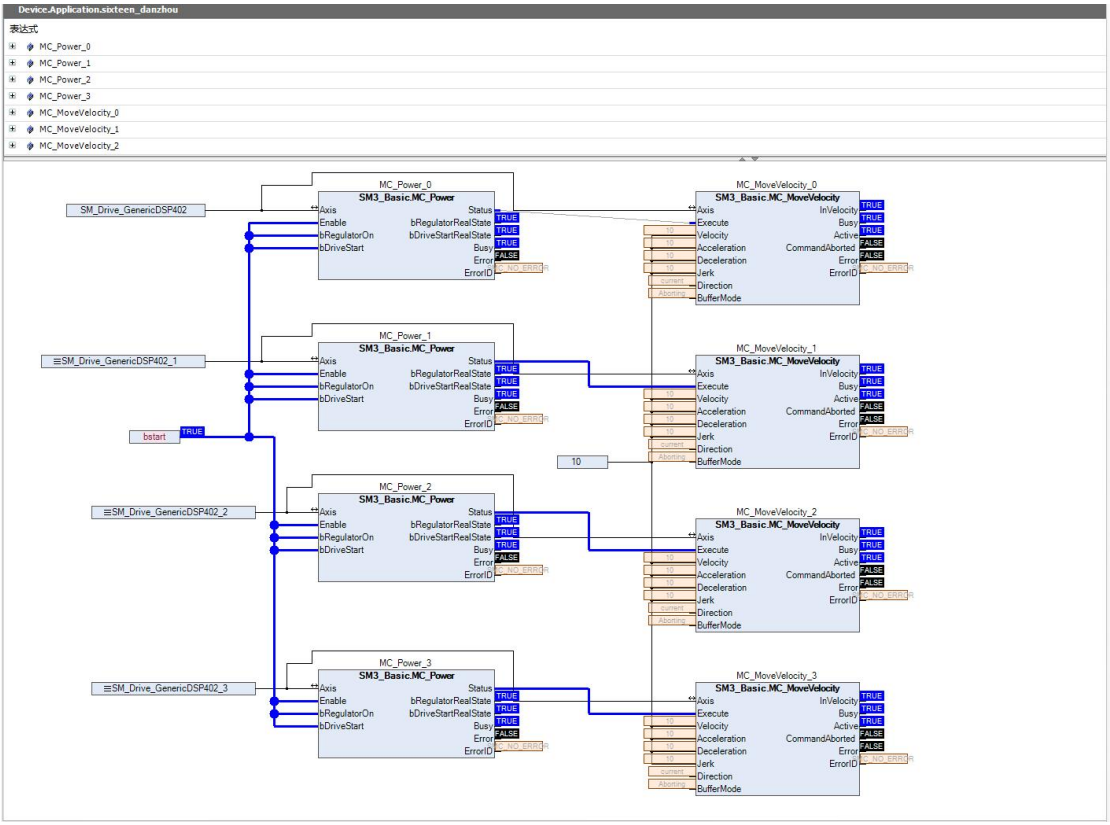


编写程序



将任务放到 EtherCAT_Task 任务下

运行程序



三、用户程序应用编程

3.1 单轴 MC 定位的运动控制编程

3.1.1 运控应用的编程要点提醒

AX3000 控制器与伺服轴配合的运动控制是基于 EtherCAT 总线网络来实现的，与以往的硬件输出的脉冲方式不同，完全是采用软件方式实现，具体来说是在每个很短的 EtherCAT 总线周期中进行一次计算、发布一次控制命令，来实现对伺服的控制。因此，需要注意如下要点：

- 用户 MC 控制程序，是以 EtherCAT 任务周期执行的，MC 相关的 POU 应配置在 EtherCAT 任务下执行，多数 MC 功能块若放在低优先级的 Main 任务的 POU 中，无法正常运行；
- MC 功能块的执行，需要通信中的通信数据对象来传递，因此 PDO 配置表中要有必需的配置项；若遗漏了配置相关的数据对象，伺服可能无法正常运行，且不会有出错报警；
- 控制器可通过 SDO 的配置对伺服的功能码进行初始化设定操作，使得伺服的运行模式（一般为 CSP 模式）、 伺服电机编码器模式、电子齿轮比等，以确保控制命令与物理运行位置的对应；对伺服的初始化，还可提高设备的调试效率、部件更换后不会出错；
- 对于伺服轴的控制，要遵循轴状态转移的规则和逻辑，根据轴当前的状态和希望的运动，使用合适的 MC 功能块进行控制；
- 用户程序中使用的是 MC 功能块的实例，一个 MC 实例，只能用于一个伺服轴的控制，若同时用于几个伺服轴的控制，会导致控制混乱；
- 一个正在运转的伺服轴，必需有一个 MC 功能块对其运行进行监控，哪怕是 MC_Stop 也是一种监控，避免因程序逻辑的跳转导致无 MC 功能块监控的状态，系统会停机报错，这样的错误不容易检查；
- 注意调试的安全处理。若伺服系统采用增量式编码器，正常运行之前需要有归零操作，伺服驱动器的 DI 信号输入端口，可接入原点位置信号，对于在有限范围内运动（如丝杆），调试前应有极限与安全保护信号。

3.1.2 单轴控制常用的 MC 功能块

MC 功能块（FB）也称为 MC 指令，准确地讲，用户程序中使用的是 MC 功能块的对象实例，伺服轴通过 MC 对象实例来进行控制，例如：

MC_Power1：MC_Power；// 声明实例 MC_Power1

MC_Power1（Axis=Axis1）

单轴的控制，一般用于定位的控制，即伺服电机拖动外部机构运动到指定的位置；有时也需要伺服以指定的速度或力矩运行等，在单轴控制中，常用到如下的 MC 功能块：

控制操作	需要使用的 MC 指令	说明
伺服使能	MC_Power	运行该指令，使伺服轴使能，才能进行后续的运行控制
伺服点动运行	MC_Jog	伺服电机的点动运行，常用于低速试车，用于检验设备或调整伺服电机位置
相对定位	MC_MoveRelative	以当前位置为参考，运行指定距离
相对叠加定位	MC_MoveAdditive	在伺服当前运行指令的基础上，再相对运行指定距离
绝对定位	MC_MoveAbsolute	命令伺服运行到指定的坐标点

控制操作	需要使用的 MC 指令	说明
速度控制	MC_MoveVelocity	命令伺服以指定的速度运行
力矩控制	SMC_SetTorque	命令伺服以指定的力矩运行
伺服暂停	MC_Halt	命令伺服暂停运行，若 MC_Movexxx 再次触发，伺服可以再运行
紧急停机	MC_Stop	命令伺服紧急停机，只有 stop 命令复位后，触发 MC_Movexxx，伺服才可以再运行
告警复位	MC_Reset	当伺服出现告警停机后，运行该指令进行复位，
改变伺服运行模式	MC_ControlMode	使用该指令可以让伺服选择“位置”、“速度”或“力矩”模式
伺服原点回归	MC_Home	命令伺服开始原点回归操作，应用系统的原点信号、两侧极限信号等都接在伺服的 DI 端口
控制器原点回归	SMC_Homing	控制系统开始原点回归操作，应用系统的原点信号、两侧极限信号等都接在控制器的 DI 端口

3.1.3 MC 指令与 PDO/SDO 配置

AX3000 在执行用户程序的伺服轴 MC 控制命令时，需要将执行 MC 指令时与伺服交互所需的信息项目，添加在通信 PDO/SDO 配置表中，以便完成所需的控制功能。

MC 指令	所需的 TPDO 对象	所需的 RPDO 对象
MC_Power MC_Halt MC_Stop MC_Reset MC_Home SMC_Homing	ControlWord（控制字）	StatusWord（状态字） Errorcode（错误码）
MC_Jog	TargetPosition（目标位置）	Position actual value（当前轴位置） Following error actual value（当前跟踪误差）
MC_MoveRelative		
MC_MoveAdditive		
MC_MoveAbsolute		
MC_MoveVelocity	Target velocity（目标速度） Max profile velocity（最大轮廓速度）	
SMC_SetTorque	Target torque（目标力矩）	Torque actual value（当前力矩）
MC_ControlMode	Modes of operation（运行模式）	16#6060=8: 周期同步位置 CSP 16#6060=9: 速度模式 16#6060=10: 力矩模式

上述 TPDO、RPDO 是进行单轴控制所需的基本配置项。

在 MC 控制中，伺服多数情况下为位置模式，尤其是在基于 EtherCAT 总线的应用系统中，为“周期同步位置 模式”，因此编程时在 SDO 配置中，一般将伺服设置为该运行模式。以 IS620N 为例，一般在 SDO 中初始化如下项目：

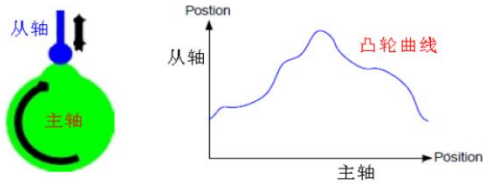
对伺服的初始化操作	需要使用 SDO 对象	说明
设为周期同步位置模式	Modes of operation（16#6060）	设为 8
设电子齿轮比	16#6091-1: 16#6091-2:	建议设为 1:1 （不推荐采用功能码设定）

设马达编码器类型	16#0201（IS620N 功能码）	0: 增量；1: 绝对有限长； 2: 绝对无限长
设允许最大转速	Max profile velocity（16#607F）	适用于速度模式、力矩模式
设允许最大力矩	Max torque（16#6072）	适用于速度模式、力矩模式
设回零模式	Homing method（16#6098）	
设回零速度	Homing speeds（16#6099）	
探针功能	Touchprobe function（16#60B8）	

3.2 多轴 CAM 凸轮同步的运动控制编程

凸轮运动是借用了机械凸轮与挺杆相对运动特性概念，指控制器按特定的相对位置非线性关系，令伺服从轴跟随主轴，进行的连续同步运动，以满足设备所需的运动特性的一种运动，在定长切割、追剪控制、飞剪控制、 多色套印等有同步要求的应用中，被大量使用。

电子凸轮曲线的主从轴位置关系如下图，水平轴为主轴位置，而纵轴为从轴位置：



AX3000 采用的是软件方式，实现的凸轮运动控制特性，也就是利用软件数字方式的“凸轮表”来代替机械凸轮， 因此也称电子凸轮控制。相比于机械凸轮，有如下特点：

- 凸轮形状制作容易：采用凸轮表、凸轮曲线或数组描述凸轮；
- 凸轮形状容易多样：支持多个凸轮表选择、运行中可动态切换；
- 凸轮形状修改容易：允许凸轮表关键点运行中动态修改；
- 多个凸轮从轴：允许有多个凸轮从轴；
- 凸轮挺杆：允许有多个凸轮挺杆、多个设置区间；

◆ 凸轮离合器：凸轮运行中，用户程序可使之进入与退出凸轮运行；

◆ 电子凸轮特有功能：支持虚拟主轴，相位偏移，输出叠加

AX3000 的凸轮运算采用纯软件方式执行，若在 CAM 运行状态，每次进入 EtherCAT 任务，都会计算一次从轴的下一目标点，因此相比硬件凸轮运算，具有更好的功能灵活性。

电子凸轮的控制有三个要素：

① 主轴：被用于同步控制的参考轴；

② 从轴：根据主轴位置，按照所需非线性特性进行跟随运动的伺服轴；

③ 凸轮表：描述主轴 – 从轴相对位置与范围、周期性等的数据表或凸轮曲线。用户编写程序需要设计凸轮表，指定主轴与从轴，运行中在合适时刻触发凸轮运行，就可以使得从轴进入凸轮运行了。

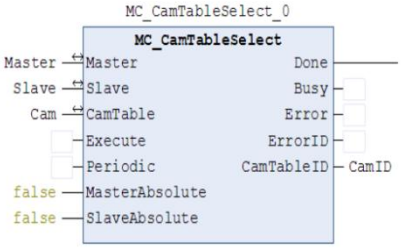
电子凸轮控制的基本指令功能块：

控制操作	需要使用的 MC 指令	说明
凸轮表选择	MC_CamTableSelect	运行该指令，关联主轴、从轴凸轮表三者关系
进入凸轮运行	MC_CamIn	让从轴进入凸轮运行
退出凸轮运行	MC_CamOut	让从轴退出凸轮运行
修正凸轮相位	MC_Phasing	主轴相位修改

3.2.1 凸轮运行的主要功能块说明

1） 凸轮表选择 MC_CamTableSelect 功能块

该功能块用于关联主轴、从轴、凸轮表三者的关系，并设定凸轮运行的周期、主轴从轴的位置模式（绝对位置或相对位置）等，该指令为管理型指令，即触发该指令，只执行一次之后，相关主从轴就可以按该特性一直运行下去了；若需要更换凸轮表，或改变主从轴，就需要再触发执行该功能块一次。



MC_CamTableSelect(
Master:= , // 凸轮主轴
Slave:= , // 凸轮从轴
CamTable:= ,// 凸轮表
Execute:= , // 指令触发变量，上升沿有效
Periodic:= , // 设定凸轮周期性
MasterAbsolute:= ,// 主轴位置模式
SlaveAbsolute:= , // 从轴位置模式
Done=> , //
Busy=> ,
Error=> ,
ErrorID=> ,
CamTableID=>);
	MC_CamIn(
	Master:= , // 凸轮主轴
	Slave:= , // 凸轮从轴
	Execute:= ,// 执行触发，上升沿有效
	MasterOffset:= ,// 主轴位置偏置
	SlaveOffset:= , // 从轴位置偏置
	MasterScaling:= ,// 主轴放大比率
	SlaveScaling:= , // 从轴放大比率
	StartMode:= , // 从轴触发位置模式
	CamTableID:= , // 凸轮表指针
	VelocityDiff:= , // 速度偏差
	Acceleration:= , // 加速度
	Deceleration:= , // 减速度
	Jerk:= , // 加加速度
	TappetHysteresis:= , // 挺杆回差
	InSync=> , // 同步指示
	Busy=> , // 正在运行
	CommandAborted=> , // 指令被中断
	Error=> , // 出错
	ErrorID=> ,// 错误号
	EndOfProfile=> , // 已执行到凸轮尾
	Tappets=> ,// 挺杆有效
	);

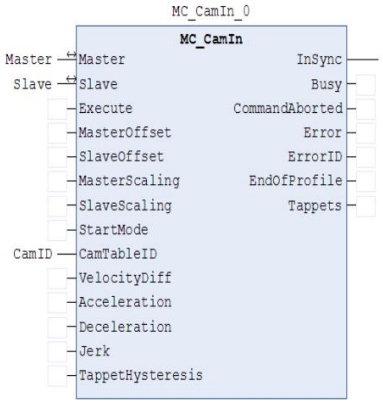
该功能块的输出输出变量及其说明如下：

MC_CamTableSelect

	类型	初始值	描述	
VAR_IN_OUT				
Master	AXIS_REF		映射到主轴	主轴名
Slave	AXIS_REF		映射到从轴	从站名
CamTable	MC_CAM_REF		映射为cam表格描述	待选择的CAM曲线表名称
VAR_INPUT				
Execute	BOOL	FALSE	此变量的一个上升沿将会启动功能块的处理	
Periodic	BOOL	TRUE	TRUE = 周期， FALSE = 非周期	
MasterAbsolute	BOOL	TRUE	TRUE = 绝对坐标， FALSE = 相对坐标	
SlaveAbsolute	BOOL	TRUE	TRUE = 绝对坐标， FALSE = 相对坐标	
VAR_OUTPUT				
Done	BOOL	FALSE	TRUE，如果cam表格选择结束	
Busy	BOOL	FALSE	TRUE，如果功能块的处理没有完成	
Error	BOOL	FALSE	功能块内部发生错误信号	
ErrorID	SMC_ERROR	0	错误ID	
CamTableID	MC_CAM_ID		cam表格的定义	选择并解析后的CAM表ID

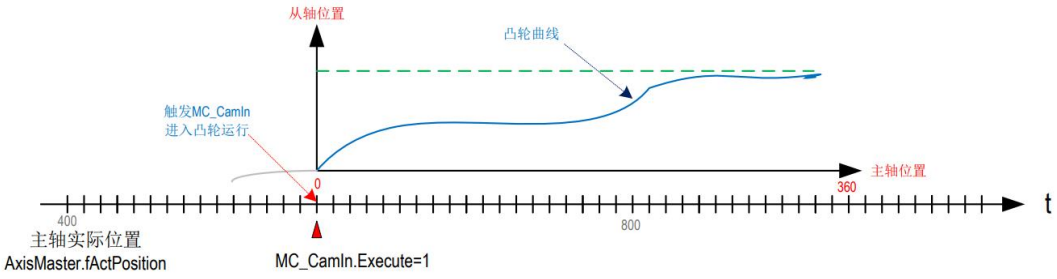
2） 凸轮运行的 MC_CamIn 功能块

当运行了 MC_CamTableSelect 功能块后，再该功能块可使得 Slave 从轴进行凸轮运行状态（Synchronized_ Motion，即 Axis.nAxisState=6），并在每次进入 EtherCAT 任务后，系统均执行一次该功能块，根据主轴当前位置和凸轮表，计算从轴的下一个目标点。若事先没有运行 MC_CamTableSelect 功能块，触发该功能块 就会报错。



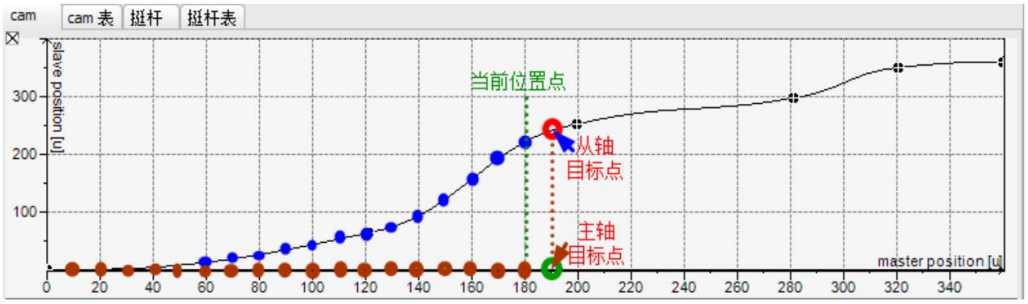

```
MC_CamOut(
    Slave:= , // 凸轮从轴
    Execute:= ,// 触发变量，上升沿有效
    Done=> , // 执行完成
    Busy=> , // 正在执行中
    Error=> , // 出错
    ErrorID=> // 错误号
);
```

本指令的功能是，使凸轮从轴进入与凸轮主轴同步运行状态，根据主轴当前位置、凸轮表的位置关系，控制凸轮从轴调整到对应的目标点，该指令的执行对主轴没有任何影响。

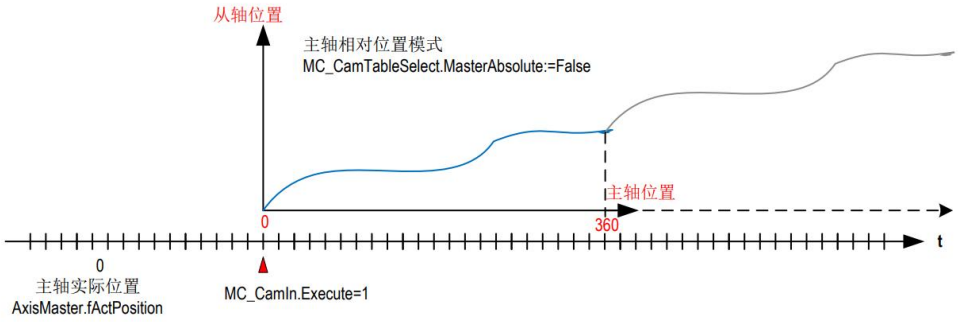


一旦 MC_CamIn 被触发，从轴按照凸轮表的位置对应关系，跟随主轴的位置运动，注意是位置对应，而不是速度对应。

进入凸轮运行后，系统每次 EtherCAT 中断都会解析 CAM 凸轮表，根据主轴当前位置计算从轴的下一个目标点，然后将下一目标位置发送给从轴，令其运行 ：



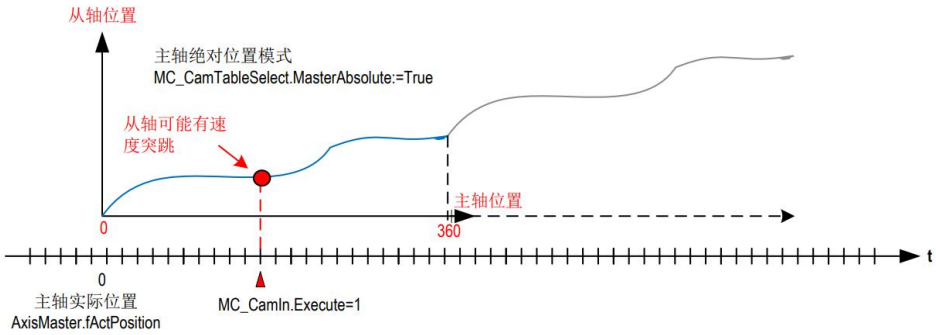
3.2.2 当主轴和从轴都为相对位置模式时的运行特点



当主轴为相对位置模式，在进入 CAM 时，凸轮运算模块会以当前位置作为主轴的起点 X=0，进行运算；

当从轴为相对位置模式，在进入 CAM 时，凸轮运算模块会以当前位置作为从轴的起点 Y0 进行运算，此后的 CAM 输出结果在此基础上进行叠加。

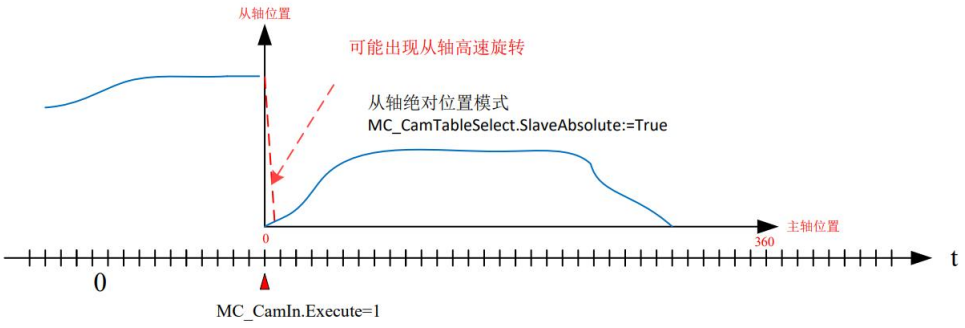
3.2.3 当主轴为绝对位置模式，从轴为相对位置模式



当主轴为绝对位置模式，在进入 CAM 时，凸轮运算模块会以当前主轴位置运算得到从轴位置，因此：

- 进入 CAM 运行时从轴位置调整时的高速转动，导致设备冲击或损坏；
- 若当前位置超出 CAM 表有效范围，从轴不会运动，并会告警；
- 若 CAM 表为周期模式，当本周期执行完毕，即开始下一 CAM 周期的连续运行。

3.2.4 当主轴为相对对位置模式，从轴为绝对位置模式



当从轴为绝对位置模式，进入 CAM 运行时会调整到 CAM 要求的位置，若偏差比较大，会出现高速运动的自动调整。

根据应用特点采取对策：

◆ 对于必需对准操作的设备，如定长切割的旋转刀作为凸轮从轴必需采用绝对位置，编程时注意在旋转刀第一 次旋切动作之前，要进行旋转刀的归零位操作；

◆ 合理设置凸轮表的主轴位置范围，在下一周期开始时，避免凸轮出现反方向的位置调整；

◆ 运行 SMC_GetCamSlaveSetPosition，将凸轮进入点的从轴位置设为从轴的当前坐标。

◆ 对于可以采用相对位置模式的应用，尽量采用相对位置模式：

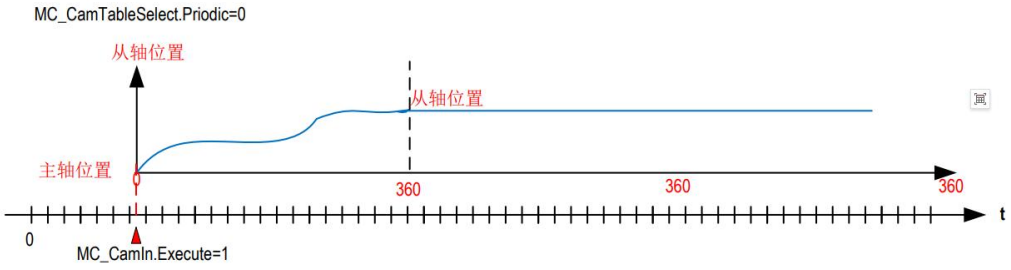
MC_CamTableSelect.SlaveAbsolute:=False; 或设 MC_CamIn.StartMode:=1;（相对模式）

注意：

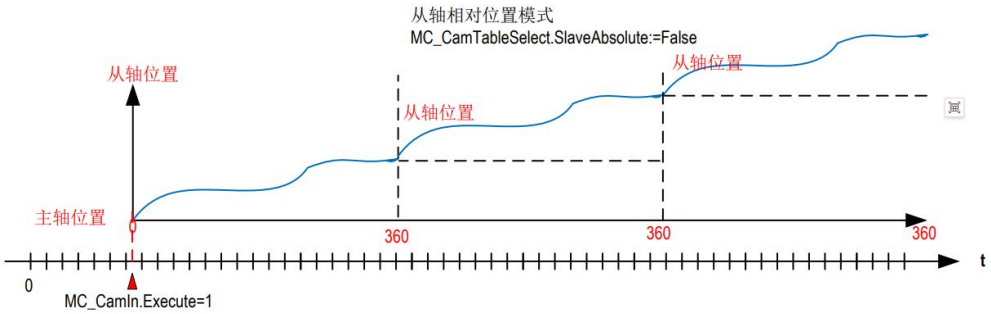
当从轴设为“有限长”的绝对模式时，在作回零调整时，若可以左转回零，也可以右转回零，控制器会选择 一个比较近的方向回零。设计凸轮表的范围时，尤其要注意不要让凸轮表的范围超过了实际需要运行的范围， 否则就可以出现伺服从轴的瞬间高速回转调整，形成机械冲击。

3.3 凸轮表的周期模式特点

下图为单周期凸轮运行的效果。当凸轮表选择为单周期模式（Periodic:=0），在运行完一个凸轮表周期后， 从轴即脱离凸轮运行状态。

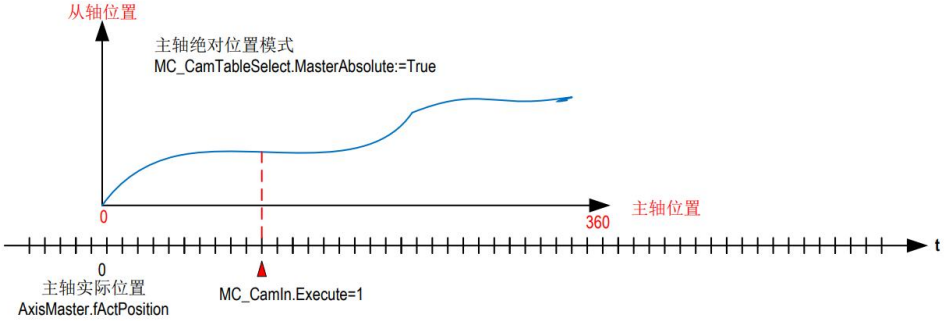


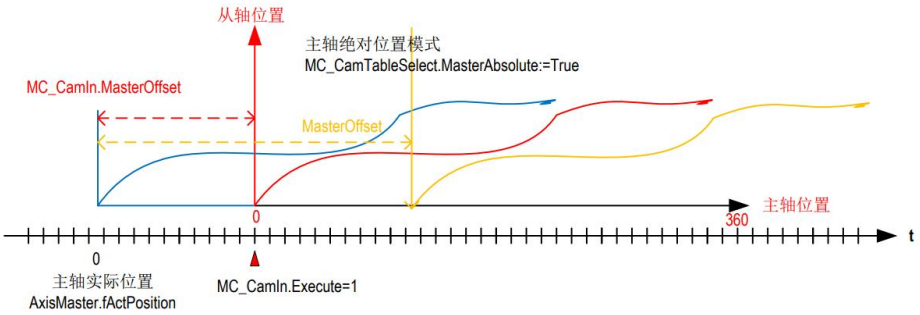
当凸轮表选择为周期模式（Periodic:=1），在运行完一个凸轮表周期后，从轴又开始下一凸轮周期的运行， 直到有用户程序命令其退出凸轮运行状态，如下图：



上图为周期性凸轮运行的效果，一旦完成凸轮表中主轴位置范围，即自动开始下一凸轮周期的运动。

3.3.1 CamIn 运行的 Offset 功能



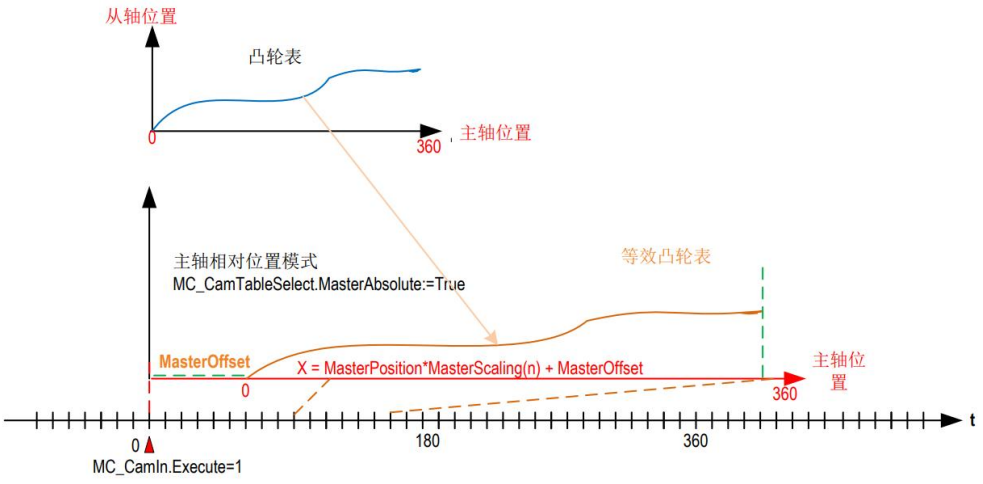


通过给凸轮主轴设置合适的偏移 Offset 值，可以修改凸轮从轴的起始点。根据主轴当前的实际　　位置，来计算偏移值，可以使其从凸轮表的 0 点开始：

MasterOffset := 0 - AxisMaster.fActPosition

3.3.2 凸轮运行中的主轴 MasterScaling 计算

默认情况下，系统对 MasterScaling=1，若用户程序修改了该变量，则：



给凸轮主轴设置比例 SCALE 值，可以对主轴的位置进行线性缩放，使之与凸轮表的对应位置关系符合所期望的要求。

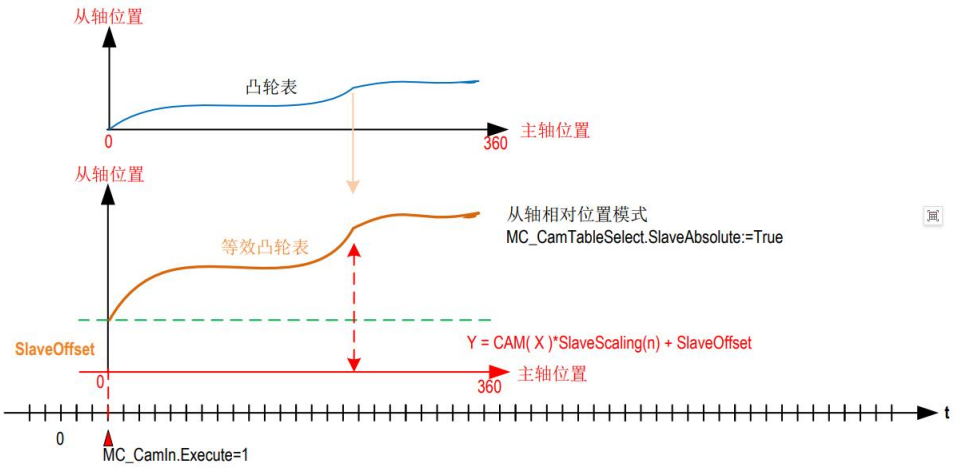
若考虑了主轴的偏移设定值，在凸轮表中主轴（X）的计算位置将是：

X = MasterPosition * MasterScaling(n) + MasterOffset

该参数可用于设备加工工件的尺寸微调。

3.3.3 凸轮运行中的从轴 SlaveScaling 计算

默认情况下，系统对 SlaveScaling=1，若用户程序修改了该变量，则：



给凸轮从轴设置比例 SlaveScaling 值，可以对从轴的位置进行线性缩放，使凸轮控制的输出符合所期望的从轴运动位置要求。

若考虑了从轴的偏移设定值，凸轮从轴（Y）的输出位置是：

Y = CAM(X)*SlaveScaling(n) + SlaveOffset

该参数可用于设备加工工件的尺寸微调。

3.3.4 凸轮运行中的 Offset、Scale 使用特性与注意事项

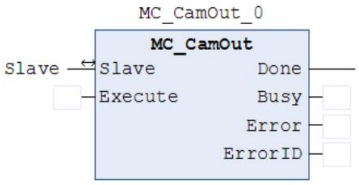
主轴位置模式、从站位置模式，除应用系统特别要求外，建议尽量采用相对模式，这样编程简单，出现机械系统冲击的可能性比较小；

凸轮表主轴起止范围、Offset、Scale 等设定值，是可以弥补 CAM 表的设计偏差，建议尽量参考默认设定， 这样便于调试和维护，运行出错的机会也可以减少；

当 CAM 凸轮表周期执行完毕/或退出/或切换 CAM 表后，再次执行 MC_CamIn 重新进入时，系统会清除内存中的 Offset、Scale 等设定值，恢复为默认值，需要留意。

3.3.5 凸轮运行状态的退出 MC_CamOut 功能块

当从轴正在凸轮运行，触发执行该功能块可使得 Slave 从轴退出凸轮运行状态，进入连续运行状态（Continuios_Motion，即 Axis.nAxisState=5），该指令的执行对主轴没有任何影响。



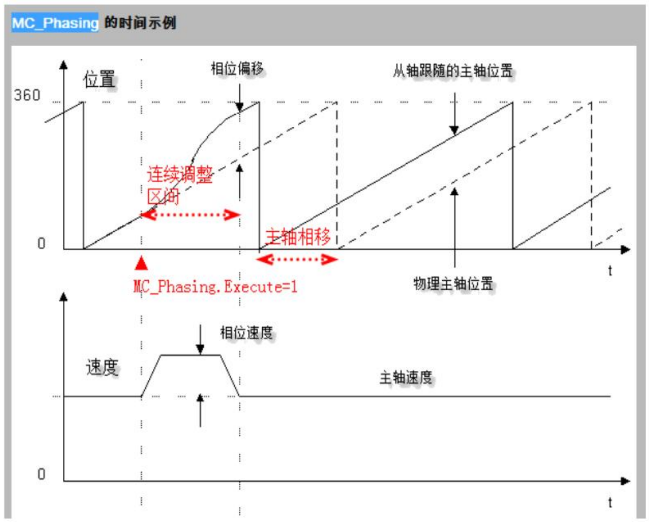

```
VAR i: INT;
CAM: MC_CAM_REF := (
   byType:=2, (* non-equidistant *)
   byVarType:=2, (* UINT *)
   nElements:=128,
   xStart:=0,
   xEnd:=360);
Table: SMC_CAMTable_UINT_128_2 := (
    fEditorMasterMin := 0, fEditorMasterMax := 360,
    fTableMasterMin := 0, fTableMasterMax := 6000,
    fEditorSlaveMin := 0, fEditorSlaveMax := 360,
    fTableSlaveMin := 0, fTableSlaveMax := 6000);
END_VAR
```

注意，执行该指令，从轴脱离凸轮运行状态，但其运行速度将仍以退出凸轮状态瞬间的速度继续运行，就如 机械离合器断开之后，设备从动轮仍会靠惯性继续运转一样。此时必须有其他的 MC 功能块接替对从轴进行运动控制，比如 MC_Movexxx、MC_Halt、MC_Stop 等等都可以。

3.4 凸轮主轴相位调整 MC_Phasing 功能块

在有些设备的 CAM 同步运行中，有时会需要对凸轮从轴与主轴的相对相位需要作修正处理，可以采用 MC_ Phasing 指令来实现。

该指令执行时，修改的是 CAM 从轴位置的运算结果，可以设置因相位调整产生的速度、加速度约束；调整时从轴的运行速度和位置保持连续；调整完成后，保持该相位差持续运行



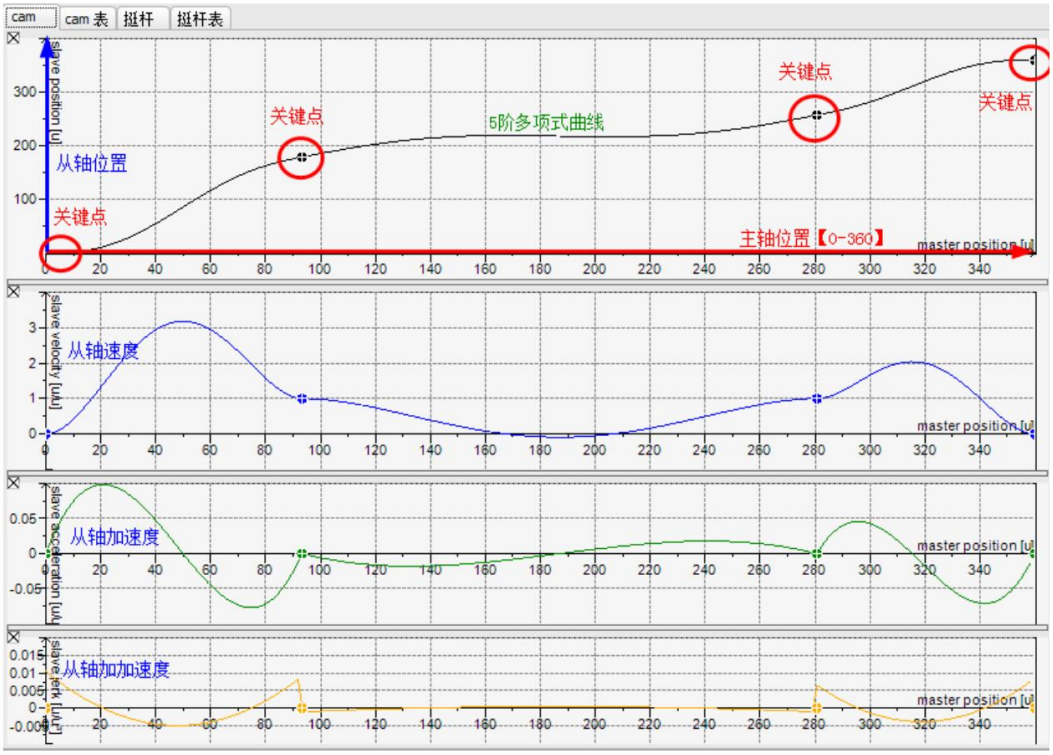
该指令在一些具有色标的同步控制中，需要调整色标在被剪切工件段中的位置，就需要用到该指令。

3.5 凸轮表设计与其数据结构

在编写凸轮运行的用户程序时，凸轮表是其中的一个编写项，它决定了凸轮运行的特性，可以采用图形和表格的方式输入。

3.5.1 凸轮表的特点

如下图为图形方式的 CAM 凸轮表，横坐标为主轴位置轴，轴的长度为凸轮运行的行程，共有 4 条坐标曲线，纵轴分别为从轴位置、从轴的速度、从轴的加速度、从轴的加加速度曲线。编程调试时，往往更关注位置曲线、 速度曲线，调试平稳性时还会关注加速度曲线。



凸轮曲线有如下特点：

◆ 在主从位置曲线坐标中，垂直轴即为从轴可能运动的范围；其他 3 个曲线的垂直轴则是从站与主轴的速度之比、 从轴与主站的加速度之比；

◆ 凸轮曲线在垂直方向是一个单调曲线，即主轴的每一个坐标，只能对应从轴唯一的坐标值；凸轮执行时，主 轴坐标是按由小到大的方向运动；

◆ 凸轮曲线可以有若干个关键点，两个关键点之间的线型可以分别设定直线或 5 次曲线，系统会为每条 5 次曲线作最佳优化，以尽量减小速度和加速度的突变；

◆ 水平轴（主轴）的起止坐标默认从 0 开始，360 结束，用户可以根据实际的物理行程进行修改。

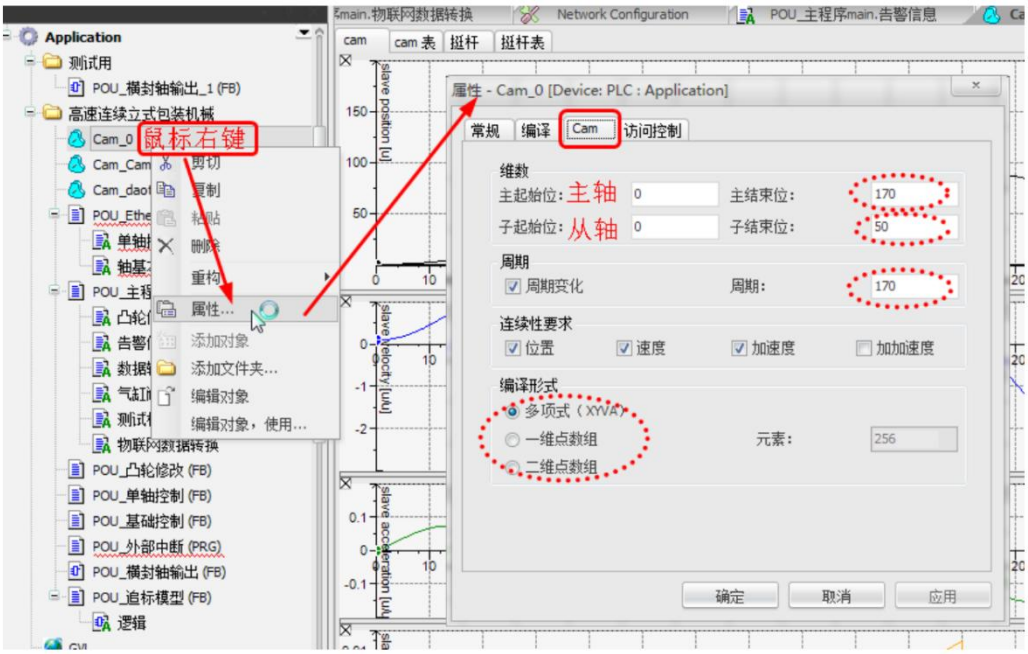
3.5.2 CAM 凸轮表的输入方法

① 当新建一个凸轮表时，系统会自动设置一个最简单的凸轮曲线，用户在此基础上进行修改，形成自己所需的 CAM 曲线表

② 用户可以增减凸轮曲线的关键点个数、修改关键点的坐标；

③ 用户可以修改任意两个相邻关键点之间的线型，或 5 次曲线，或直线；

④ 凸轮曲线中的关键点之间，系统会默认按 5 次曲线进行联接，这样可以确保运行时速度的连续性，减小机械冲击；



凸轮曲线中的关键点，往往与控制对象的机械运动要求相关，例如：

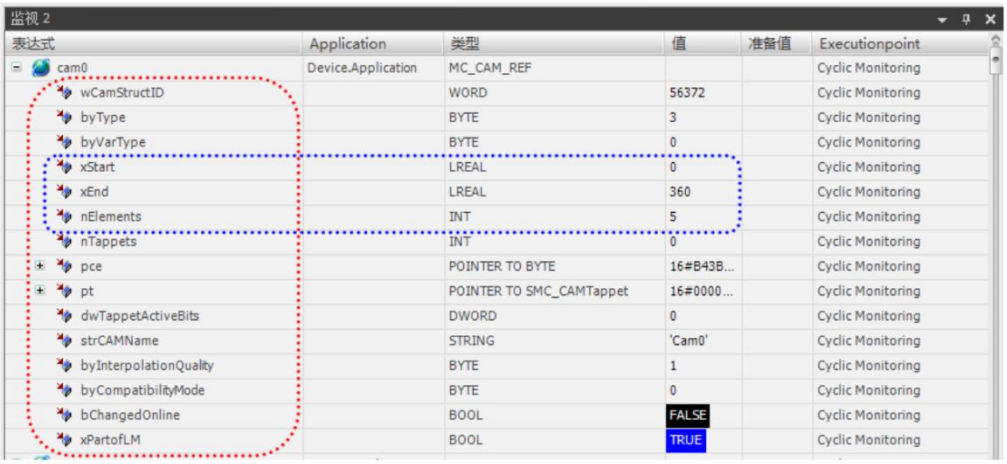
① 对于追剪应用，主轴的坐标范围建议对应运行区间的物理行程，便于分析；

② 从轴往返行程起止点、同步运行区间的起始位置点、脱离同步位置点等，就是重要的关键点；

③ 比例同步区间，凸轮曲线的线段应该为直线，其他区间则为 5 次曲线

3.5.3 CAM 凸轮表的内部数据结构与数组

在 CodeSys 中，对每一个 CAM 表，都有描述该 CAM 表的数据结构，描述该 CAM 表的特征数据。下图为“CAM0” 凸轮表的描述数据结构，请注意其结构各变量名称：



监视 2

表达式	Application	类型	值	准备值	Executionpoint
cam0	Device.Application	MC_CAM_REF			Cyclic Monitoring
wCamStructID		WORD	56372		Cyclic Monitoring
byType		BYTE	3		Cyclic Monitoring
byVarType		BYTE	0		Cyclic Monitoring
xStart		LREAL	0		Cyclic Monitoring
xEnd		LREAL	360		Cyclic Monitoring
nElements		INT	5		Cyclic Monitoring
nTappets		INT	0		Cyclic Monitoring
pce		POINTER TO BYTE	16#B43B...		Cyclic Monitoring
pt		POINTER TO SMC_CAMTappet	16#0000...		Cyclic Monitoring
dwTappetActiveBits		DWORD	0		Cyclic Monitoring
strCAMName		STRING	'Cam0'		Cyclic Monitoring
byInterpolationQuality		BYTE	1		Cyclic Monitoring
byCompatibilityMode		BYTE	0		Cyclic Monitoring
bChangedOnline		BOOL	FALSE		Cyclic Monitoring
xPartofLM		BOOL	TRUE		Cyclic Monitoring

CodeSys 在内部有一个数据结构来描述 CAM 凸轮表的特征：如果我们手动编写一个 CAM 表也是可以的，如下：虽然我们不需要手动编写 CAM 表，但我们可以通过数据结构的访问操作，对所需的 CAM 特性数据进行修改。

注意：我们在声明 CAM0 凸轮表时，系统自动默认声明了全局变量类型的 CAM0 数据结构，同时声明了 CAM0_A[i]数组

例如在用户程序中，修改 CAM0 凸轮表关键点个数或坐标：

```
CAM0. nElements:=20; // 将关键点个数改为 20 个
  CAM0. xEnd:=500;      // 将主轴的结束点改为 500.          // 例如在用户程序中，修改其中 2 个关键点的坐标:
    CAM0_A[3].dx:=30;
    CAM0_A[3].dy:=45;
    CAM0_A[3].dv:=1;
    CAM0_A[3].da:=0;

    CAM0_A[4].dx:=60;
    CAM0_A[4].dy:=75;
    CAM0_A[4].dv:=1;
    CAM0_A[4].da:=0;
```

在线修改 CAM 凸轮表的方法

所谓“在线修改 CAM 曲线”，是指用户编写的程序在执行过程中，根据控制特性的需要，对 CAM 曲线的关键点坐标，进行的修改。

修改的内容，一般是修改关键点坐标，但也可以修改关键点的个数、修改主轴的距离范围等。

提醒：在进入凸轮运行之前，修改凸轮表，不宜运行中修改，避免出现预想不到的运动结果 需要修改 CAM 凸轮表应用场合：

① 一般情况下，OEM 客户使用调试验证成功的凸轮表；

② 若有几种加工对象或模式，可以考虑预设多个凸轮表，根据用户的需要，自动进行切换；

③ 有些设备，要求其适应范围更宽，例如包装设备，若要求其适用的包装长度在 10cm~25cm 范围，且对应运行速度作自动适应改变时，可能必需在线修改 CAM 凸轮表。

3.5.4 CAM 凸轮表的引用与动态切换

CAM 凸轮表在控制器内部是用一个数组来保存，可以通过特定的 MC_CAM_REF 变量类型来指向，例如声明：

凸轮表 p: MC_CAM_REF;

可以给该变量赋值，也可认为是将其指向某具体的凸轮表：

凸轮表 p:= Cam0; // 指向所需的凸轮表

```
凸轮表 p: MC_CAM_REF; // 凸轮表指针;
TableID: uint; // 凸轮表选择命令，可由 HMI 设置;
Case TableID of
0: 凸轮表 p := 凸轮表 A;
1: 凸轮表 p := 凸轮表 B;
2: 凸轮表 p := 凸轮表 C;
End_case

MC_CamTableSelect_0(          // 凸轮关系
        Master:= 虚主轴 ,
        Slave:=          凸轮从轴 ,
        CamTable:=       凸轮表 p,
        Execute:= ReSelect, // 上升沿触发凸轮表选择
        Periodic:=       TRUE,
        MasterAbsolute:=FALSE,
        SlaveAbsolute:= FALSE);
```

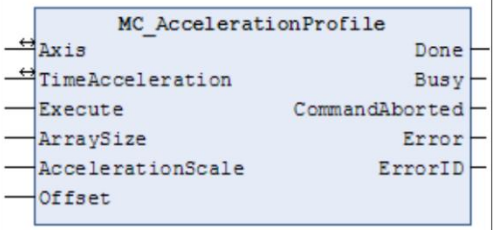
上面的例程，利用该 MC_CAM_REF 变量的赋值运算，就可以实现多个凸轮表的切换操作了。

四、MC 指令详解

4.1 单轴指令

4.1.1 MC_AccelerationProffle

1）指令格式

指令	名称	图形表现	ST 表现
MC_ AccelerationProfile	加速度轮廓指令		MC_AccelerationProfile(Axis:= , TimeAcceleration:= , Execute:= , ArraySize:= , AccelerationScale:= , Offset:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

2）相关变量：

输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例
TimeAcceleration	轴加速时间和加速度描述	MC_TA_REF	-	-	轴加速时间和加速度数据描述，加速数据由多组数据组成

输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE,FALSE	FALSE	输入的一个上升沿将启动功能块的处理
ArraySize	动态数组	INT	数据范围	0	运行轮廓中使用的数组个数
AccelerationScale	综合因子	LREAL	“正数”+”0”	1	MC_TA_REF 中加速度或减速度的比例因子
Offset	偏移	LREAL	-	0	加减速度的总体偏移值

输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE,FALSE	FALSE	轴指令执行完成，置为 TRUE
Busy	指令正在执行	BOOL	TRUE,FALSE	FALSE	当前指令正在执行中，置为 TRUE
CommandAbort	指令被中断	BOOL	TRUE,FALSE	FALSE	当前指令被中断，置为 TRUE
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ERROR	参阅 SMC_ERROR	0	异常发生时，输出错误代码

3）功能说明

◆ 本功能块为时间段和加减速度的轮廓运动模型,运行模式为 Discrete Motion, 按用户在 TimeAcceleration 变 量中设定的数据运行。

◆ 本功能块运行状态为 Standstill 中 ， 指令运行时的状态为 Discrete Motion, 其他状态无法运行。

◆ 启动指令为 Execute 的上升沿启动 ， 本指令在 Discrete Motion 重复运行速度都是在上一次的叠加 ， 容易引起系统故障。

◆ TimeAcceleration 为 MC_TA_REF 数据类型 ；

MC_TA_REF 具体描述如下 ：

成员	类型	初始值	描述
Number_of_pairs	INT	0	轮廓路径的段数
IsAbsolute	BOOL	TRUE	绝对运动 (TRUE) 和相对运动选择
MC_TA_Array	ARRAY[1..N] OF SMC_TA	-	时间和加速值的数组

SMC_TA 具体描述如下 ：

成员	类型	初始值	描述
delta_time	TIME	TIME#0ms	加速度段的时间
acceleration	LREAL	0	当前的加速度值

注:设置的加速度体现在速度的变化上,所有的加速度变化按 S 曲线的方式变化 ， 从最终的结果变化为起始加速度为 A。

终止加速度为[B](A+B)/2 的加速度数据体现在最终的速度上;

4）时序图

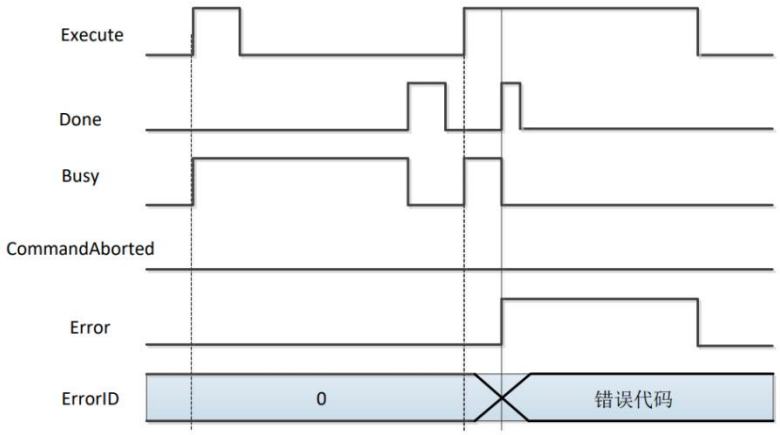
条件 MC_TA_Array 通过其他方式已经设置;

轴必须处于 Standstill 状态指令才能运行;

功能块的 Execute 必须有上升沿的条件;

功能块的 Done 表示指令正常执行完成；

功能块的 Busy 表示当前功能块正在执行中；



错误说明

错误的出现为轴状态不是在 Standstill 中启动指令或指令系统中的参数错误，出现轴错误只能清除错误后才开始运行。

【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.1.2 MC_Halt

1）指令格式

指令	名称	图形表现	ST 表现
MC_Halt	轴正常停止命令	MC_Halt MC_Halt Axis, Execute, Deceleration, Jerk Done, Busy, CommandAborted, Error, ErrorID	MC_Halt(Axis:= , Execute:= , Deceleration:= , Jerk:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE,FALSE	FALSE	输入的一个上升沿将启动功能块的处理
Deceleration	减速度	LREAL	“正数”+”0”	0	功能块的减速度 (u/s^2)
Jerk	跃度	LREAL	“正数”+”0”	0	指定跃度［指令单位 /s^3］

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE,FALSE	FALSE	轴指令执行完成，置为 TRUE
Busy	指令正在执行	BOOL	TRUE,FALSE	FALSE	当前指令正在执行中，置为 TRUE
CommandAborted	指令被中断	BOOL	TRUE,FALSE	FALSE	当前指令被中断，置为 TRUE
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ERROR	参阅 SMC_ERROR	0	异常发生时，输出错误代码

3）功能说明

◆ 本功能块为在正常运行的情况下停止一个轴的运动，而其它一个运动轴指令再运行时可以终止本指令的运行。

◆ 本功能块运行状态为运行状态(Motion)才能运行 ， 其他状态无法运行。

◆ 启动指令为 Execute 的上升沿启动 ；

◆ 指令运行中的状态为 Discrete Motion, 运行完成后状态为 Standstill。

◆ 运动指令被在 MC_Halt 和 MC_Stop 打断时，轴变量 bAvoidReversalOnHaltStop 置 TRUE 可以对加速度进行调整，避免出现停止过程中速度反向现象。在暂停（Halt）或停止（Stop）时，若断点处的速度值很小，而加速度值较大，且加加速度值较小，则可能会导致速度不能直接降为 0，而是先降为 0 然后进行反向加速运动再反向减速为 0。这是由于加速度过大而加加速度又太小，以当前最大加加速度将加速度降为 0 时的速度值要大于断点速度值，从而速度必须反向才能与加速度同时降为 0。这种现象会出现在二次和二次平滑曲线中。

4)时序图

轴必须处于运行状态(Motion)指令才能运行；

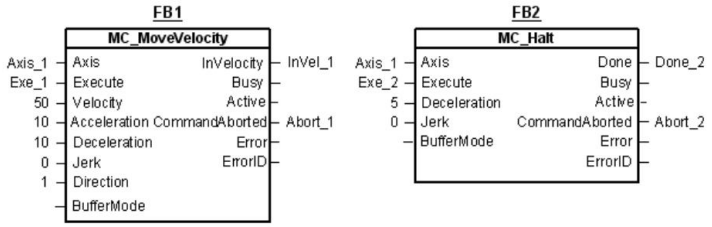
功能块的 Execute 必须有上升沿的条件；

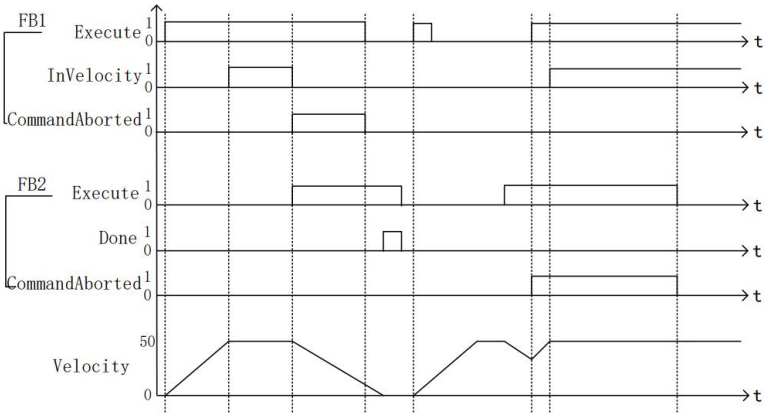
功能块的 Done 表示指令正常执行完成；

功能块的 Busy 表示当前功能块正在执行中；

功能块的 CommandAborted 表示指令被其他运动控制指令中断，此时标志位为 TRUE；

例程:在执行 MC_MoveVelocity 指令和 MC_Halt 指令在不同的时序操作中对应的标志位的变化 ；对 CommandAborted 的处理描述如下图的时序描述。





5） 错误说明

错误的出现为轴状态不是在 Standstill 中启动指令或指令系统中的参数错误，出现轴错误只能清除错误后才开始运行。

【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.1.3 MC_Home

1）指令格式

指令	名称	图形表现	ST 表现
MC_Home	轴回零指令	MC_Home MC_Home Axis Execute Position Done Busy CommandAborted Error ErrorID	MC_Home(Axis:= Axis, Execute:= , Position:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE,FALSE	FALSE	输入的一个上升沿将启动功能块的处理
Position	轴到达位置	LREAL	数据范围	0	代表轴位置的回零位置

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE,FALSE	FALSE	轴指令执行完成，置为 TRUE
Busy	指令正在执行	BOOL	TRUE,FALSE	FALSE	当前指令正在执行中，置为 TRUE
CommandAbort	指令被中断	BOOL	TRUE,FALSE	FALSE	当前指令被中断，置为 TRUE
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ERROR	参阅 SMC_ERROR	0	异常发生时，输出错误代码

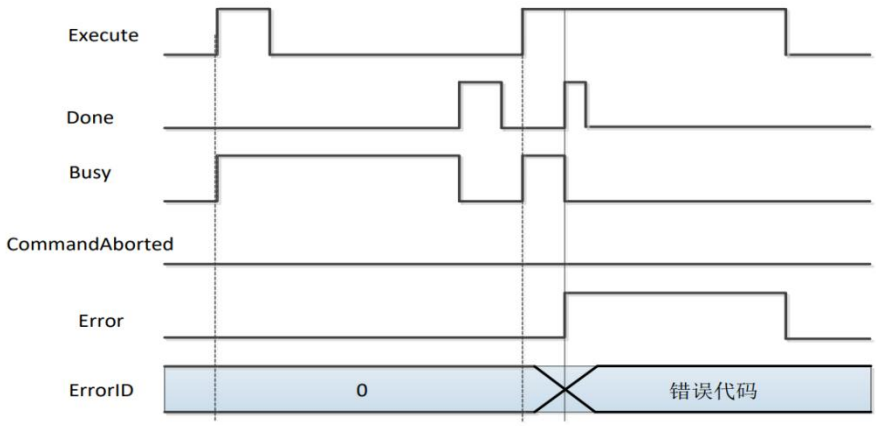
3） **功能说明**

◆ 本功能块为回零操作 ， Position 数据位轴的零点位置。

◆ 本功能块运行状态为 Standstill 中 ， 指令运行时的状态为 homing， 其他状态无法运行。

◆ 启动指令为 Execute 的上升沿启动指令。

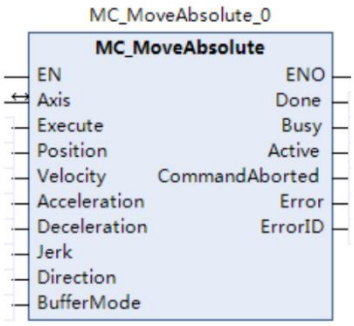
4） **时序图**



4.1.4 MC_MoveAbsolute

指定绝对坐标的目标位置，进行定位。

1）指令格式

指令	名称	图形表现	ST 表现
MC_ MoveAbsolute	轴绝对位置控制指令		MC_MoveAbsolute_0(Axis:= Axis, Execute:= , Position:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Direction:= , BufferMode:= , Done=> , Busy=> , Active=> , CommandAborted=> , Error=> , ErrorID=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE,FALSE	FALSE	输入的一个上升沿将启动功能块的处理
Position	轴到达位置	LREAL	数据范围	0	此位置为轴的绝对位置数据
Velocity	运行速度	LREAL	数据范围	0	轴运行到目标位置的最大速度
Acceleration	加速度	LREAL	数据范围	0	速度变大时加速度值
Deceleration	减速度	LREAL	数据范围	0	速度变小时减速度值
Jerk	跃度	LREAL	数据范围	0	曲线加减速的斜率变化值
Direction	指令极性	MC_ DIRECTION	Negative,shortest Positive, current,fastest	shortest	Negative: 反向移动； Shortest: 根据最短路径选择方向； Positive: 正向移动； Current: 按当前方向移动； Fastest: 自动选择最快的方向移动； (此功能轴在旋转模式下有效)
BufferMode	缓存模式选择	MC_ BUFFER_ MODE	0: Aborting 1: Buffered 2: BlendingLow 3: BlendingPrevious 4: BlendingNext 5: BlendingHigh	0	指定多个实例启动运动指令时的动作。 0：中断 1：等待 2：以低速合并 3：以前一个速度合并 4：以后一个速度合并 5：以高速合并

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE,FALSE	FALSE	轴指令执行完成，置为 TRUE
Busy	指令正在执行	BOOL	TRUE,FALSE	FALSE	当前指令正在执行中，置为 TRUE
Active	控制	BOOL	TRUE,FALSE	FALSE	在进行控制时为 TRUE

输出变量	名称	数据类型	有效范围	初始值	描述
CommandAbort	指令被中断	BOOL	TRUE,FALSE	FALSE	当前指令被中断，置为 TRUE
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ ERROR	参阅 SMC_ ERROR	0	异常发生时，输出错误代码

3）功能说明

◆ 本功能块为轴绝对定位指令 ， Position 数据为轴的绝对位置。

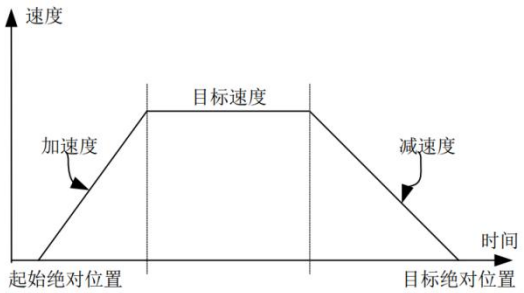
◆ 本功能块运行状态为 Standstill 中 ， 指令运行时的状态为 Discrete Motion, 一个完整的运行过程一定要控制轴的不同运动状态。

◆ 启动指令为 Execute 的上升沿启动 ， 本指令在 Discrete Motion 可以重复上升沿有效 ， 每次都可以刷新最新的 Position 位置。

◆ Acceleration 或 Deceleration 为零，指令运行都为异常状态，但轴的状态为 Discrete Motion；

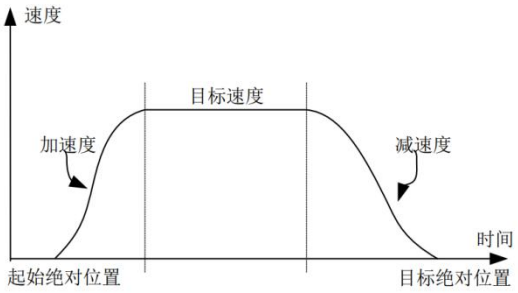
◆ 梯形加减速动作

Velocity、Acceleration 和 Deceleration 有数据；而 Jerk 为 0；



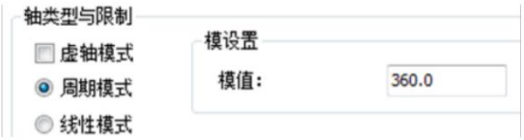
◆ S 曲线加减速动作

Velocity、Acceleration、 Deceleration 和 Jerk 有数据；

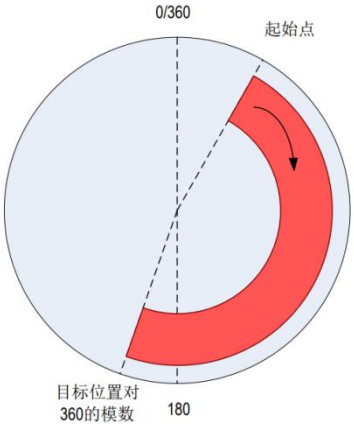


◆ 轴在周期模式下绝对定位

① 轴旋转周期设置为 360、Direction 设置为 Positive。

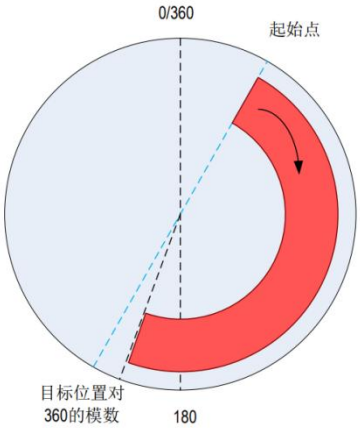


当 Position 对 360 的模值（Position/360 取余，比如 Position 为 380 则对 360 的模值为 20，Position 为 350 则对 360 模值为 350）>起始绝对位置时，此时轴朝正向运行 Position 对 360 的模值-起始绝对位置的距离。

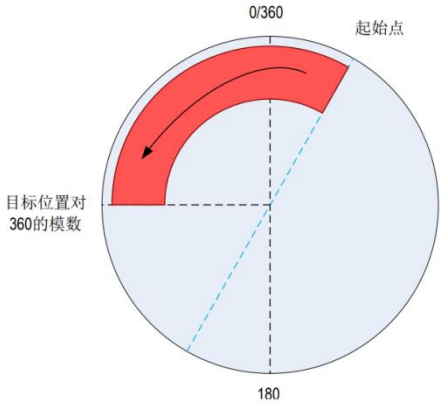


当 Position 对 360 的模值（Position/360 取余，比如 Position 为 380 则对 360 的模值为 20）< 起始绝对位置时，此时轴朝正向运行 360-起始绝对位置 + Position 对 360 的模值的距离。

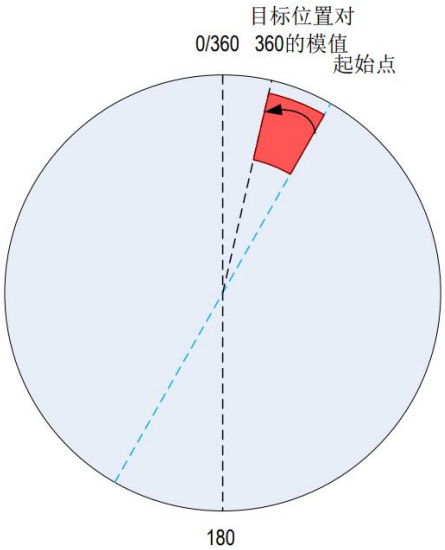
② 轴旋转周期设置为 360、Direction 设置为 shortest 或者 fastest。Position 对 360 的模值为 XPosition 当=0<XPosition-起始绝对位置<180 时,轴朝正向运行 XPosition-起始绝位置的距离。



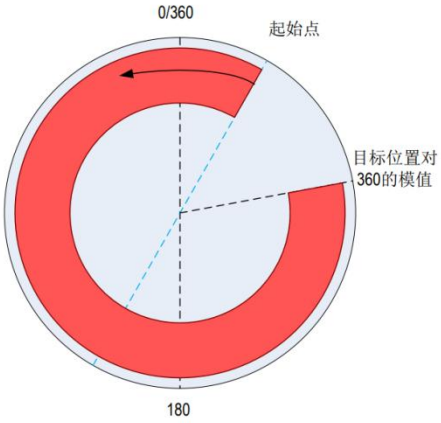
当 180<XPosition- 起始绝对位置时 ，轴朝反向运行 360-XPosition+ 起始绝对位置的距离。



当 XPosition<起始绝对位置时 ， 轴朝反向运行起始绝对位置 -XPosition 的距离。

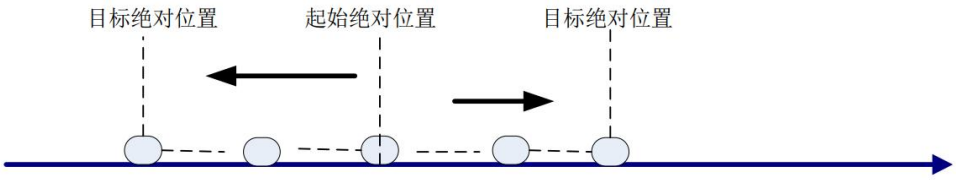


③ 轴旋转周期设置为 360、Direction 设置为 shortest 或者 Negative。Position 对 360 的模值为 XPosition 轴朝反向运行起始绝对位置+360-XPosition 的距离。



◆ 轴在线性模式下绝对定位

当目标绝对位置>起始位置，则正向移动目标绝对位置-起始位置距离，当目标位置<起始位置时则反向移动起始位置-目标位置的距离。线性模式下设定的运行方向不决定轴运行方向。



4）注意事项

* Direction(方向选择) 选择“2: Current (指定为当前方向)”时，沿着前一动作的指令方向进行动作。因此， 根据指令组合的不同，可能与前一动作的运动指令的输入发出指令的方向不一致。

* 相对位移（输入绝对位移与当前位移差）不为 0 但 Velocity（目标速度）为 0 时，指令运行异常。

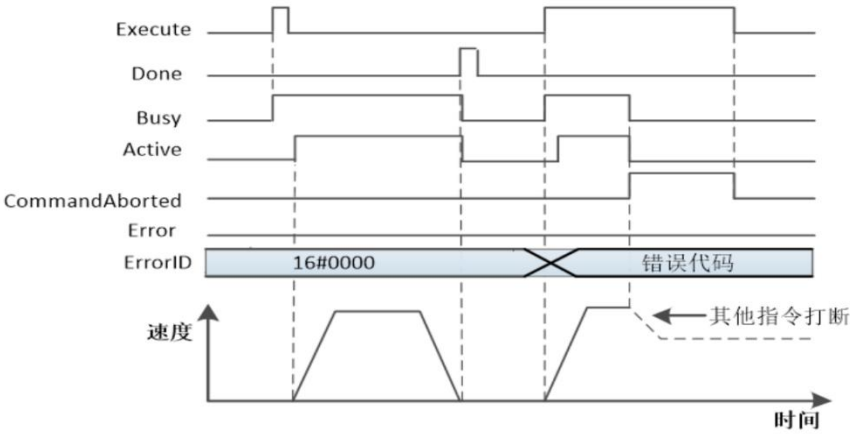
* 相对位移（输入绝对位移与当前位移差）不为 0 且 Velocity（目标速度）也不为 0 时，若输入 Acceleration（加 速度）、Deceleration（减速度）或 Jerk（跃度）为 0，则会赋默认初值。相对位移（输入绝对位移与当前位移差）与 Velocity（目标速度）均为 0 时，指令置 Done。

5）时序图

◆ 在启动 Execute（启动）的同时，Busy（执行中）变为 TRUE。在下一个周期 Active（控制中）变为 TRUE。

◆ 到达 Position（目标位置）定位完成时，Done（完成）变为 TRUE。

◆ 利用其它指令中止本指令时，CommandAborted（执行中断）变为 TRUE，Busy（执行中）、Active（控制中）变为 FALSE。



◆ 重启运动指令

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

◆ 其他指令执行中本指令启动

对当前正在执行的指令启动本指令，可切换或缓存到本指令。

本指令的多个实例启动时的动作由 BufferMode(缓存模式选择) 指定

缓存模式选择		说明
中断（Aborting）		立即中止当前正在执行的指令，切换为本指令。 轴的动作方向因指令切换而反转时，根据轴参数中的“反转时动作”进行反转。
等待（Buffered）		功能块在最后一个命令运动终止后立即启动。这里不进行混合。 当达到结束条件（Done、InVelocity、InEndVelocity、InGear、InSync、EndOfProfile 等）时，新运动以先前运动的速度开始。 如果之前的运动是 MC_MoveAbsolute 或 MC_MoveRelative，那么新的运动会在静止状态下开始。
合并（Blending）		以当前正在执行的指令到达目标位置时的速度 (中继速度) 为启动速度，连续使已缓存的本指令动作。变更当前正在执行的指令的动作，确保以中继速度到达目标位置。中继速度的指定方法分为如下 4 种。
	以低速合并（BlendingLow）	功能块在最后一个命令运动终止后立即启动。 轴在运动之间不会停止，而是以两个运动命令中较低的速度通过第一个运动的结束位置。
	以前一个速度合并（BlendingPrevious）	功能块在最后一个命令运动终止后立即启动。 轴在运动之间不停止，而是以第一个运动命令的速度通过第一个运动的结束位置。
	以后一个速度合并（BlendingNext）	功能块在最后一个命令运动终止后立即启动。 轴在运动之间不停止，而是以第二个运动命令的速度通过第一个运动的结束位置。
	以高速合并（BlendingHigh）	功能块在最后一个命令运动终止后立即启动。 轴不会在运动之间停止，而是以两个运动命令的较高速度通过第一个运动的结束位置。

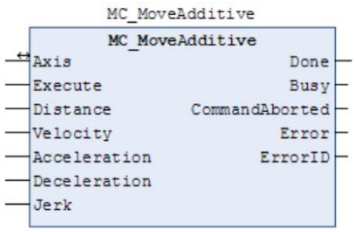
◆ 重启运动指令

对本指令进行多个实例启动运动指令时，可选择中断、等待、合并。

4.1.5 MC_MoveAdditive

轴在原来指令位置上再叠加 Distance 指定的数据，用于运动轴控制过程的在线叠加位置。本指令在 Discrete Motion 状态下可以在任何时刻添加 MC_MoveAddtive 的相关执行过程； 在 Continuous Motion 中只能在指 令执行的某段中；在 standstill 状态下相当于 MC_MoveRelative 指令。

1）指令格式

指令	名称	图形表现	ST 表现
MC_ MoveAdditive	叠加绝对运动指令		MC_MoveAdditive(Axis:= , Execute:= , Distance:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

2） 相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE,FALSE	FALSE	输入的一个上升沿将启动功能块的处理
Distance	轴到达位置	LREAL	数据范围	0	此数据为叠加位置数据
Velocity	运行速度	LREAL	数据范围	0	轴运行到目标位置的最大速度
Acceleration	加速度	LREAL	数据范围	0	速度变大时加速度值
Deceleration	减速度	LREAL	数据范围	0	速度变小时减速度值
Jerk	跃度	LREAL	数据范围	0	曲线加减速的斜率变化值

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE,FALSE	FALSE	轴指令执行完成，置为 TRUE
Busy	指令正在执行	BOOL	TRUE,FALSE	FALSE	当前指令正在执行中，置为 TRUE
CommandAbort	指令被中断	BOOL	TRUE,FALSE	FALSE	当前指令被中断，置为 TRUE
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ ERROR	参阅 SMC_ ERROR	0	异常发生时，输出错误代码

3） 功能说明

◆ 本功能块为叠加位置指令 ， Distance 数据为轴的叠加数据；

◆ 本功能块运行状态如果为 Discrete Motion，使用情况会把其他指令的 CommandAbort 置位；

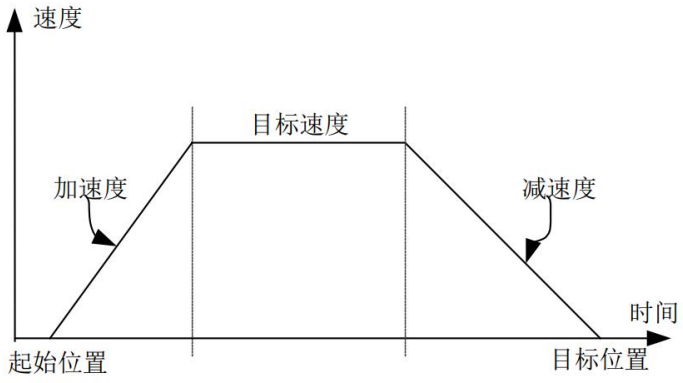
◆ 在 standstill 状态，本指令能独立运行，实现相对定位需求；

◆ Acceleration 或 Deceleration 为零，指令运行都为异常状态，但轴的状态为 Discrete Motion；

◆ 启动指令为 Execute 的上升沿启动。

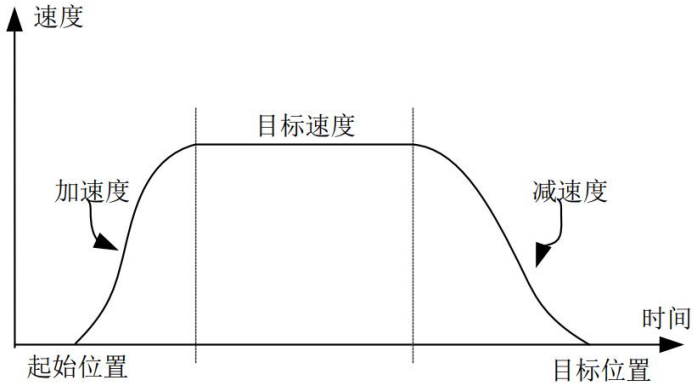
梯形加减速动作

Velocity、Acceleration 和 Deceleration 有数据；而 Jerk 为 0；



S 曲线加减速动作

Velocity、Acceleration、 Deceleration 和 Jerk 有数据；



4） 时序图

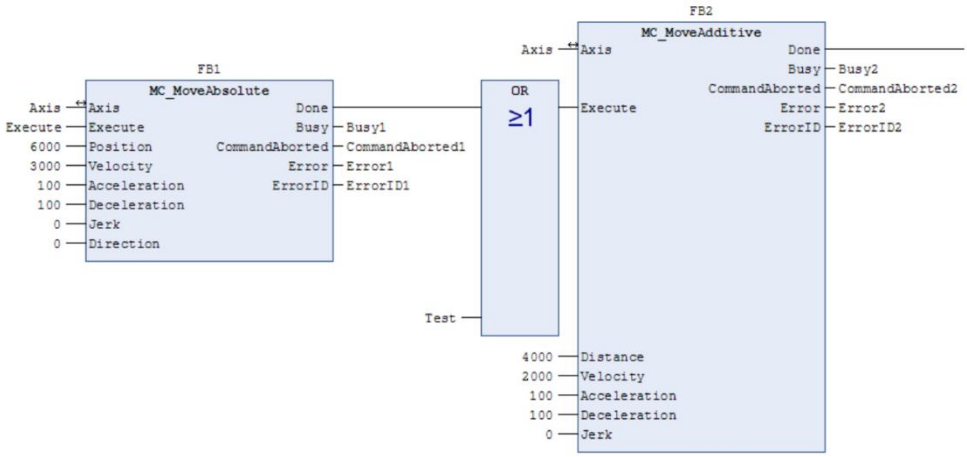
轴必须处于 Standstill 状态指令才能运行；

功能块的 Execute 必须有上升沿的条件；

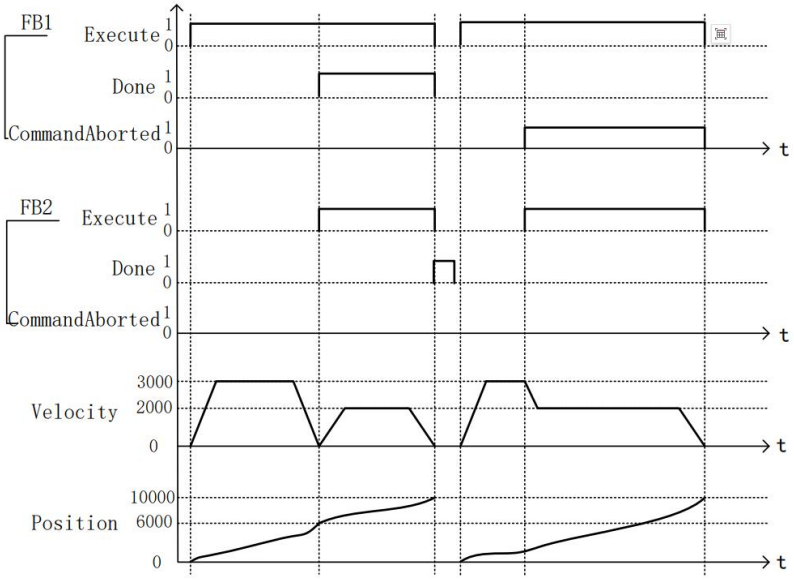
功能块的 Done 表示指令正常执行完成；

功能块的 Busy 表示当前功能块正在执行中；

◆ 举例说明



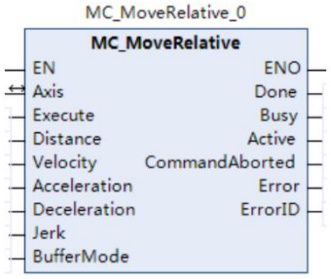
◆ 时序操作说明 ：



4.1.6 MC_MoveRelative

指定自当前位置起的移动距离，进行定位。

1）指令格式

指令	名称	图形表现	ST 表现
MC_ MoveRelative	轴相对定位指令		MC_MoveRelative_0(Axis:= Axis, Execute:= , Distance:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , BufferMode:= , Done=> , Busy=> , Active=> , CommandAborted=> , Error=> , ErrorID=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE,FALSE	FALSE	输入的一个上升沿将启动功能块的处理
Distance	运动相对位置	LREAL	数据范围	0	此数据为运动的相对位置
Velocity	运行速度	LREAL	数据范围	0	轴运行到目标位置的最大速度
Acceleration	加速度	LREAL	数据范围	0	速度变大时加速度值
Deceleration	减速度	LREAL	数据范围	0	速度变小时减速度值
Jerk	跃度	LREAL	数据范围	0	曲线加减速的斜率变化值
BufferMode	缓存模式选择	MC_BUF-FER_MODE	0: Aborting 1: Buffered 2: BlendingLow 3: BlendingPrevious 4: BlendingNext 5: BlendingHigh	0	指定多个实例启动运动指令时的动作。 0：中断 1：等待 2：以低速合并 3：以前一个速度合并 4：以后一个速度合并 5：以高速合并

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE,FALSE	FALSE	轴指令执行完成，置为 TRUE
Busy	指令正在执行	BOOL	TRUE,FALSE	FALSE	当前指令正在执行中，置为 TRUE
Active	控制	BOOL	TRUE,FALSE	FALSE	在进行控制时为 TRUE
CommandAbort	指令被中断	BOOL	TRUE,FALSE	FALSE	当前指令被中断，置为 TRUE
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE

输出变量	名称	数据类型	有效范围	初始值	描述
ErrorID	错误代码	SMC_ ERROR	参阅 SMC_ ERROR	0	异常发生时，输出错误代码

3）功能说明

◆ 指定自指令当前位置起的移动距离，进行定位。

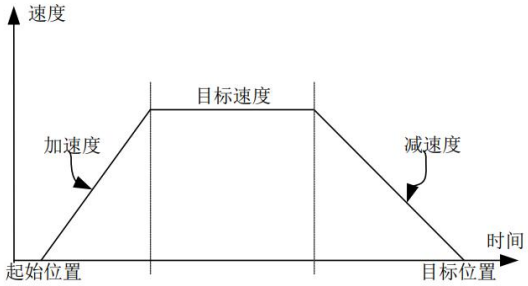
◆ 本功能块运行状态为 Standstill 中,指令运行时的状态为 Discrete Motion。

◆ 启动指令为 Execute 的上升沿启动,本指令可以重复上升沿有效，每次都可以刷新最新的 Position 位置。

◆ 可在输入变量中指定 Velocity(目标速度)、Acceleration(加速度)、Deceleration(减速度)、Jerk(跃度)。

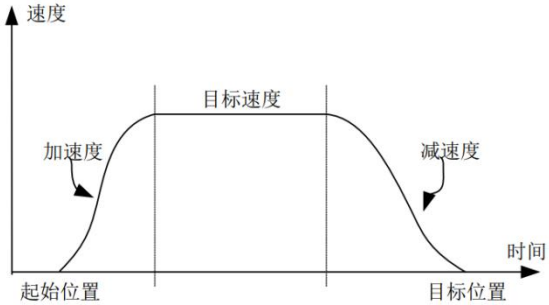
◆ 梯形加减速动作

Velocity、Acceleration 和 Deceleration 有数据；而 Jerk 为 0；



◆ S 曲线加减速动作

Velocity、Acceleration、 Deceleration 和 Jerk 有数据；



4）注意事项

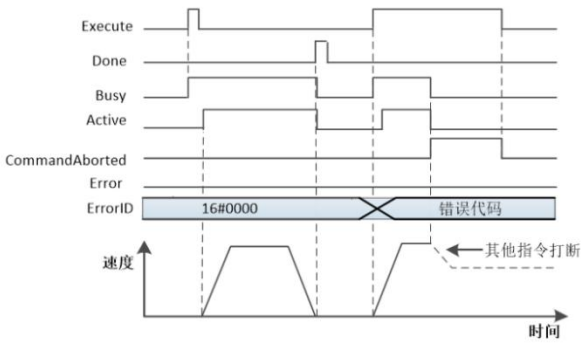
◆ 输入变量 Distance（相对位移）不为 0 但 Velocity（目标速度）为 0 时，指令运行异常。

◆ 输入变量 Distance（相对位移）不为 0 且 Velocity（目标速度）也不为 0 时，若输入 Acceleration（加速度）、 Deceleration（减速度）或 Jerk（跃度）为 0，则会赋默认初值。

◆ 输入变量 Distance（相对位移）与 Velocity（目标速度）均为 0 时，指令置 Done。

5）时序图

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



◆ 重启运动指令

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

重启时的 Distance（移动距离）的起点，为重启时的指令当前位置。

◆ 其他指令执行中的本指令启动

对当前正在执行的指令启动本指令，可切换或缓存到本指令。

本指令的多个实例启动时的动作由 BufferMode（缓存模式选择）指定。

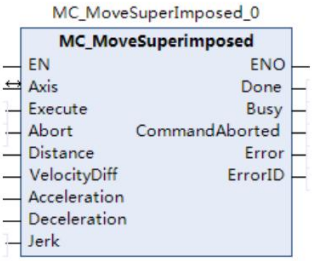
缓存模式选择		说明
中断（Aborting）		立即中止当前正在执行的指令，切换为本指令。 轴的动作方向因指令切换而反转时，根据轴参数中的“反转时动作”进行反转。
等待（Buffered）		功能块在最后一个命令运动终止后立即启动。这里不进行混合。 当达到结束条件（Done、InVelocity、InEndVelocity、InGear、InSync、EndOfProfile 等）时，新运动以先前运动的速度开始。 如果之前的运动是 MC_MoveAbsolute 或 MC_MoveRelative，那么新的运动会在静止状态下开始。
合并（Blending）		以当前正在执行的指令到达目标位置时的速度（中继速度）为启动速度，连续使已缓存的本指令动作。变更当前正在执行的指令的动作，确保以中继速度到达目标位置。中继速度的指定方法分为如下 4 种。
	以低速合并（BlendingLow）	功能块在最后一个命令运动终止后立即启动。 轴在运动之间不会停止，而是以两个运动命令中较低的速度通过第一个运动的结束位置。
	以前一个速度合并（BlendingPrevious）	功能块在最后一个命令运动终止后立即启动。 轴在运动之间不停止，而是以第一个运动命令的速度通过第一个运动的结束位置。
	以后一个速度合并（BlendingNext）	功能块在最后一个命令运动终止后立即启动。 轴在运动之间不停止，而是以第二个运动命令的速度通过第一个运动的结束位置。
	以高速合并（BlendingHigh）	功能块在最后一个命令运动终止后立即启动。 轴不会在运动之间停止，而是以两个运动命令的较高速度通过第一个运动的结束位置。

◆ 本指令执行中的其他指令启动对本指令进行多个实例启动运动指令时，可选择中断、等待、合并。

4.1.7 MC_MoveSuperImposed

轴在原来指令速度和位置的基础上叠加速度和位置的数据在运行的指令上，对整个原来的指令执行时间模型上没有影响。

1）指令格式

指令	名称	图形表现	ST 表现
MC_ MoveSuperImposed	叠加相对运动指令		MC_MoveSuperimposed_0(Axis:= Axis, Execute:= , Abort:= , Distance:= , VelocityDiff:= , Acceleration:= , Deceleration:= , Jerk:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE,FALSE	FALSE	输入的一个上升沿将启动功能块的处理
Abort	中止条件	BOOL	TRUE,FALSE	FALSE	中止正在进行的运动并重置所有输出
Distance	轴到达位置	LREAL	数据范围	0	此数据为叠加位置数据
VelocityDiff	叠加速度	LREAL	数据范围	0	轴运行叠加速度
Acceleration	加速度	LREAL	数据范围	0	速度变大时加速度值
Deceleration	减速度	LREAL	数据范围	0	速度变小时减速度值
Jerk	跃度	LREAL	数据范围	0	曲线加减速的斜率变化值

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE,FALSE	FALSE	轴指令执行完成，置为 TRUE
Busy	指令正在执行	BOOL	TRUE,FALSE	FALSE	当前指令正在执行中，置为 TRUE
CommandAbort	指令被中断	BOOL	TRUE,FALSE	FALSE	当前指令被中断，置为 TRUE
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ ERROR	参阅 SMC_ ERROR	0	异常发生时，输出错误代码

3）功能说明

◆ 本功能块为叠加位置和速度指令 ， VelocityDiff 和 Distance 分别为叠加在其他指令上的速度和位置；

◆ 在运动模式下 MC_MoveSuperImposed 可以叠加在任何其他指令；

◆ MC_MoveSuperImposed 也可以被 MC_MoveSuperImposed 中止；

◆ 在状态 StandStill 下，功能块 MC_MoveSuperimposed 的动作类似于 MC_MoveRelative；

◆ 启动指令为 Execute 的上升沿启动。

◆ 本功能块为轴绝对定位指令，Position 数据为轴的绝对位置。

◆ 启动指令为 Execute 的上升沿启动。

4）注意事项

如果 MC_MoveSuperImposed 的一个实例处于活动状态并且调用了 MC_MoveSuperImposed 类型的第二个实例，则第二个实例会报告错误；如果 MC_MoveSuperImposed 的实例处于活动状态并在 Execute 的新上升沿再次启动（可能使用不同的输入），则活动的叠加运动将中止并由新的叠加运动替换，而原始运动功能块仍保持活动状态。

Abort 引脚功能为中止叠加运动，Abort 触发会清除已经执行的叠加运动，若已叠加位置较大，则会导致期望位置突变,从而导致伺服报错，请谨慎使用。

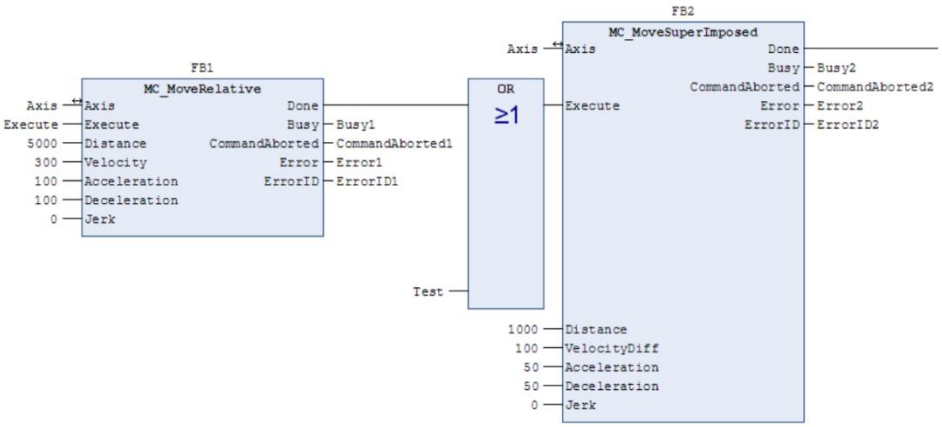
5）时序图

功能块的 Execute 必须有上升沿的条件；

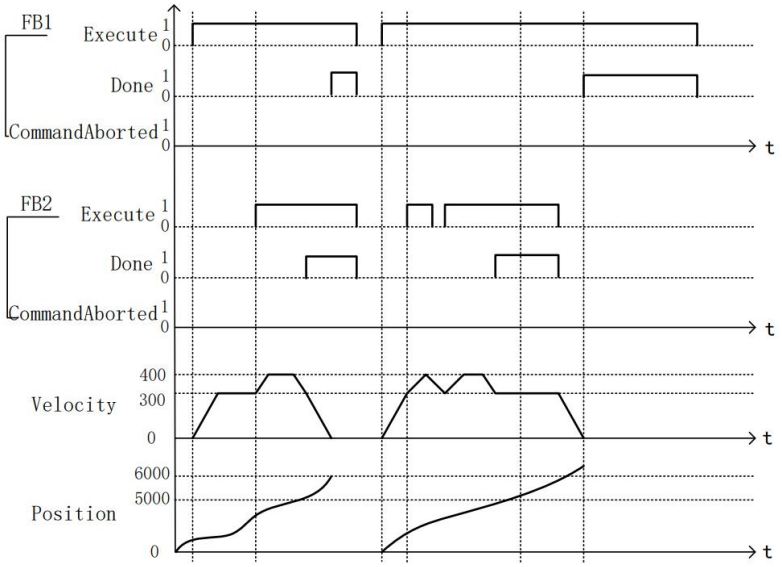
功能块的 Done 表示指令正常执行完成；

功能块的 Busy 表示当前功能块正在执行中；

◆ 举例说明



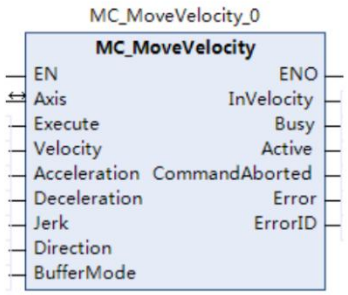
◆ 时序操作说明 ：



4.1.8 MC_MoveVelocity

该功能块可以实现驱动器速度模式（CSV）和位置模式（CSP）下对轴的速度控制，在轴使能之后，通过输入引脚 Velocity 设置轴运行速度，并且上升沿触发即可运行该指令。

1）指令格式

指令	名称	图形表现	ST 表现
MC_MoveVelocity	速度控制指令		MC_MoveVelocity_0(Axis:= Axis, Execute:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Direction:= , BufferMode:= , InVelocity=> , Busy=> , Active=> , CommandAborted=> , Error=> , ErrorID=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE,FALSE	FALSE	输入的一个上升沿将启动功能块的处理
Velocity	速度设定值	LREAL	数据范围	0	此数据为本指令的速度运行值
Acceleration	加速度	LREAL	数据范围	0	速度变大时加速度值
Deceleration	减速度	LREAL	数据范围	0	速度变小时减速度值
Jerk	跃度	LREAL	数据范围	0	曲线加减速的斜率变化值

输入变量	名称	数据类型	有效范围	初始值	描述
Direction	运行方向	MC_Direction	positive, negative, current	current	为运行方向的指令操作
BufferMode	缓存模式选择	MC_BUFFER_MODE	0: Aborting 1: Buffered 2: BlendingLow 3: BlendingPrevious 4: BlendingNext 5: BlendingHigh	0	指定多个实例启动运动指令时的动作。 0：中断 1：等待 2：以低速合并 3：以前一个速度合并 4：以后一个速度合并 5：以高速合并

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
InVelocity	达到设定速度的标志	BOOL	TRUE,FALSE	FALSE	设定的运行速度达到，置为 TRUE
Busy	指令正在执行	BOOL	TRUE,FALSE	FALSE	当前指令正在执行中，置为 TRUE
Active	控制	BOOL	TRUE,FALSE	FALSE	在进行控制时为 TRUE
CommandAbort	指令被中断	BOOL	TRUE,FALSE	FALSE	当前指令被中断，置为 TRUE
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ERROR	参阅 SMC_ERROR	0	异常发生时，输出错误代码

3）功能说明

◆ 进行基于位置控制的模拟速度控制。

◆ 在 Execute（启动）的上升沿，开始速度控制的动作。

◆ 通过 Direction(方向选择) 指定移动方向。Direction(方向选择) 为“指定为正方向”时正方向移动；

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

4）注意事项

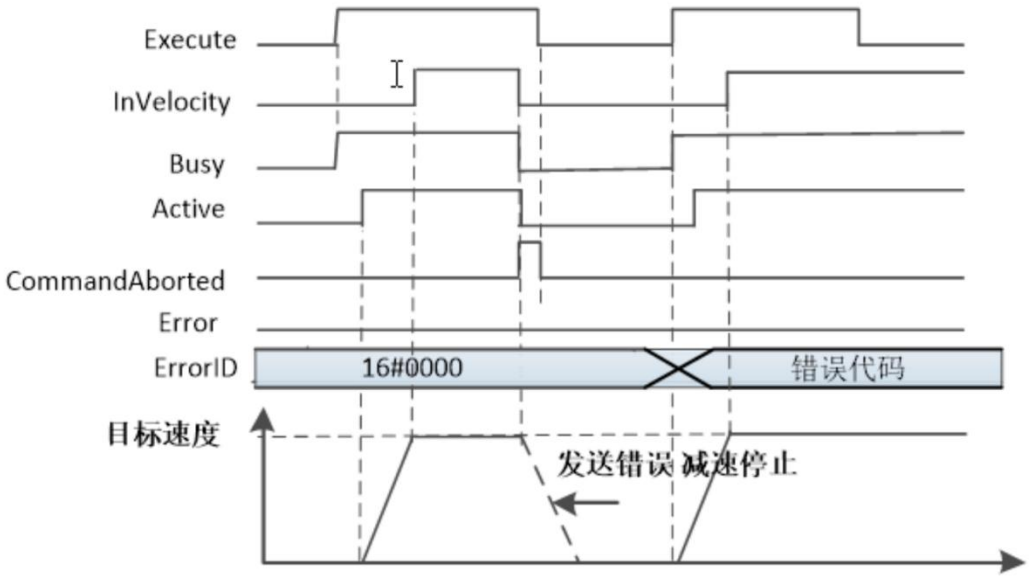
Direction(方向选择) 选择“MC_Direction (指定为当前方向)”时，沿着前一动作的指令方向进行动作。因此， 根据指令组合的不同，可能与前一动作的运动指令的输入发出指令的方向不一致。 输入变量 Velocity（目标速度）为 0 时，若输入 Acceleration（加速度）、Deceleration（减速度）或 Jerk（跃 度）为 0，则会赋默认初值。

5）时序图

在启动 Execute（启动）的同时，Busy（执行中）变为 TRUE。在下一个周期 Active（控制中）变为 TRUE。达到 Velocity（目标速度）时，InVelocity（达到目标速度）变为 TRUE。

利用其它指令中止本指令时，CommandAborted（执行中断）变为 TRUE，Busy（执行中）、Active（控制 中）、InVelocity（达到目标速度）变为 FALSE。

◆ 举例说明



◆ 其他指令执行中的本指令启动

对当前正在执行的指令启动本指令，可切换或缓存到本指令。

本指令的多个实例启动时的动作由 BufferMode(缓存模式选择) 指定

缓存模式选择		说明
中断 （Aborting）		立即中止当前正在执行的指令，切换为本指令。 轴的动作方向因指令切换而反转时，根据轴参数中的“反转时动作”进行反转。
等待 （Buffered）		功能块在最后一个命令运动终止后立即启动。这里不进行混合。 当达到结束条件（Done、InVelocity、InEndVelocity、InGear、InSync、EndOfProfile 等）时，新运动以先前运动的速度开始。 如果之前的运动是 MC_MoveAbsolute 或 MC_MoveRelative，那么新的运动会在静止状态下开始。
合并（Blending）		以当前正在执行的指令到达目标位置时的速度 (中继速度) 为启动速度，连续使已缓存的本指令动作。变更当前正在执行的指令的动作，确保以中继速度到达目标位置。中继速度的指定方法分为如下 4 种。
	以低速合并 （BlendingLow）	功能块在最后一个命令运动终止后立即启动。 轴在运动之间不会停止，而是以两个运动命令中较低的速度通过第一个运动的结束位置。
	以前一个速度合并 （BlendingPrevious）	功能块在最后一个命令运动终止后立即启动。 轴在运动之间不停止，而是以第一个运动命令的速度通过第一个运动的结束位置。
	以后一个速度合并 （BlendingNext）	功能块在最后一个命令运动终止后立即启动。 轴在运动之间不停止，而是以第二个运动命令的速度通过第一个运动的结束位置。
	以高速合并 （BlendingHigh）	功能块在最后一个命令运动终止后立即启动。 轴不会在运动之间停止，而是以两个运动命令的较高速度通过第一个运动的结束位置。

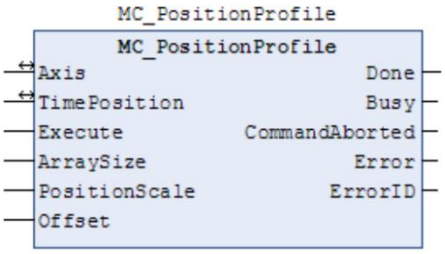
◆ 本指令执行中的其他指令启动

只有对其他指令的 BufferMode（缓存模式选择）指定中断和等待时，才能在执行本指令时，利用其它指令多 个实例启动运动指令。

指定等待时，如果本指令的输出变量“InVelocity（达到目标速度）”变为 TRUE，则开始多个实例启动指令的动作。

4.1.9 MC_PostionProfile

1）指令格式

指令	名称	图形表现	ST 表现
MC_ PositionProfile	位置轮廓指令		MC_PositionProfile(Axis:= , TimePosition:= , Execute:= , ArraySize:= , PositionScale:= , Offset:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

输入输出变量	名称	数据类型	有效范围	初始值	描述
TimePosition	轴位置运行时间和位置描述	MC_TP_REF	-	-	轴位置运行时间和位置数据描述，数据由多组数据组成

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE,-FALSE	FALSE	输入的一个上升沿将启动功能块的处理
ArraySize	动态数组	INT	数据范围	0	运行轮廓中使用的数组个数
PositionScale	综合因子	LREAL	“正数”+”0”	1	MC_TP_REF 中位置的比例因子
Offset	偏移	LREAL	-	0	位置的总体偏移值

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE,-FALSE	FALSE	轴指令执行完成，置为 TRUE
Busy	指令正在执行	BOOL	TRUE,-FALSE	FALSE	当前指令正在执行中，置为 TRUE
CommandAbort	指令被中断	BOOL	TRUE,-FALSE	FALSE	当前指令被中断，置为 TRUE
Error	错误	BOOL	TRUE,-FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ER-ROR	参阅 SMC_ERROR	0	异常发生时，输出错误代码

3）功能说明

◆ 本功能块为时间段和位置的轮廓运动模型，运行模式为 Discrete Motion，按用户在 TimePosition 变量中设定的数据运行。

◆ 本功能块运行状态为 Standstill 中，指令运行时的状态为 Discrete Motion，其他状态无法运行。

◆ 启动指令为 Execute 的上升沿启动，本指令在 Discrete Motion 重复运行。

◆ TimePosition 为 MC_TP_REF 数据类型；

MC_TP_REF 具体描述如下：

成员	类型	初始值	描述
Number_of_pairs	INT	0	轮廓路径的段数
IsAbsolute	BOOL	TRUE	绝对运动 (TRUE) 和相对运动选择
MC_TP_Array	ARRAY[1..N] OF SMC_TP	-	时间和位置的数组

SMC_TP 具体描述如下：

成员	类型	初始值	描述
delta_time	TIME	TIME#0ms	位置段的时间
position	LREAL	0	当前的位置值

注：按设置的位置数据对应的速度有变化时都按 S 曲线进行相关调整。

4）时序图

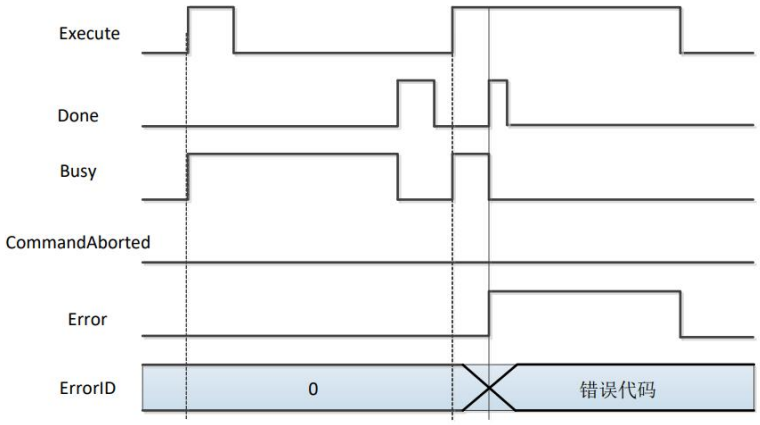
条件 MC_TP_Array 通过其他方式已经设置才能运行位置轮廓曲线指令；

轴必须处于 Standstill 状态指令才能运行；

功能块的 Execute 必须有上升沿的条件；

功能块的 Done 表示指令正常执行完成；

功能块的 Busy 表示当前功能块正在执行中；



5）错误说明

错误的出现为轴状态不是在 Standstill 中启动指令或指令系统中的参数错误，出现轴错误只能清除错误后才开始运行。

【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.1.10 MC_Power

1）指令格式

指令	名称	图形表现	ST 表现
MC_Power	轴使能指令	MC_Power Axis → Status Enable → bRegulatorRealState bRegulatorOn → bDriveStartRealState bDriveStart → Busy Error ErrorID	MC_Power(Axis:= , Enable:= , bRegulatorOn:= , bDriveStart:= , Status=> , bRegulatorRealState=> , bDriveStartRealState=> , Busy=> , Error=> , ErrorID=>);

2）相关变量

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	有效	BOOL	TRUE,FALSE	FALSE	设置为 TRUE 则功能块开始处理
bRegulatorOn	使能状态	BOOL	TRUE,FALSE	FALSE	设置为 TRUE 则设置轴为使能状态
bDriveStart	允许驱动	BOOL	TRUE,FALSE	FALSE	设置为 TRUE 以关闭功能块的紧急停止处理

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Status	可运行状态	BOOL	TRUE,FALSE	FALSE	如果轴已经准备好运动，置为 TRUE
bRegulatorRealState	轴使能信号状态	BOOL	TRUE,FALSE	FALSE	当轴使能处于有效状态时，置为 TRUE
bDriveStartRealState	允许驱动状态	BOOL	TRUE,FALSE	FALSE	如果轴没有被快速停止机制中断，置为 TRUE
Busy	执行中	BOOL	TRUE,FALSE	FALSE	如果功能块的处理没有完成，置为 TRUE
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ ERROR	参阅 SMC_ ERROR	0	异常发生时，输出错误代码

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

3）功能说明

◆ 只有在输入 Enable 为 TRUE 的时候，其它的输入才会被功能块处理。

◆ 如果功能块 MC_Power 已经被调用，并且 bRegulatorOn=FALSE，那么功能块将设置相关轴的轴状态（nAxisState）为 power_off 状态，表明驱动器还没有做好运动准备。

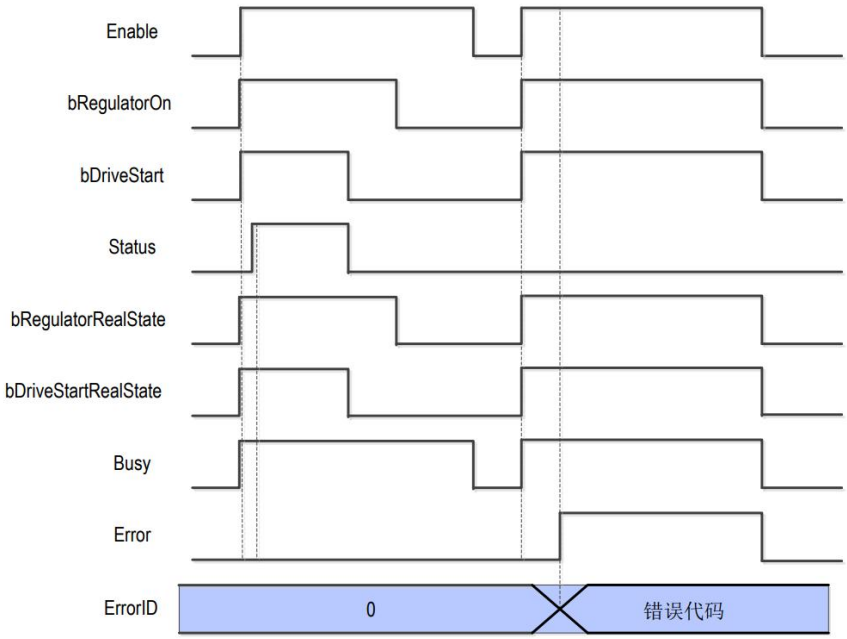
◆ 如果功能块 MC_Power 已经被调用，并且 bRegulatorOn=TRUE，如果此时轴没有错误发生，那么功能块将 设置相关轴的轴状态（nAxisState）为 standstill 状态；如果有错误发生，将输出相应的错误状态。

◆ 如果 Enable，bRegulatorOn 以及 bDriveStart 都为 TRUE，但是输出 Status 在一定时间后仍为 FALSE，那么输出 Error 将会被置位。当在使能状态情况下产生一个硬件问题，可能会发生这种情况。

如果使能信号丢失（通常在操作模式下），相关轴的 nAxisState 将会被置位 ErrorStop 状态。

4）时序图

将 Enable 设为 TRUE,bRegulatorOn 设为 TRUE,bDriveStart 设为 TRUE，表示正在受理指令的 Busy 变为 TRUE，然后轴进入使能 ON 状态，Status 状态变为 TRUE。



5）错误说明

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

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

【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.1.11 MC_ReadActialPostion

指令读取驱动器运行的实际位置，保存在自己定义的变量单元中。

1）指令格式

指令	名称	图形表现	ST 表现
MC_ ReadActualPosition	实际位置读取指令	MC_ReadActualPosition MC_ReadActualPosition Axis Valid Enable Busy Error ErrorID Position	MC_ReadActualPosition(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , Position=>);

2） 相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE,FALSE	FALSE	为 TRUE 状态读取伺服的当前位置

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Valid	位置数据可获取标志	BOOL	TRUE,FALSE	FALSE	能正确的获取驱动器的位置，置为 TRUE
Busy	指令正在执行	BOOL	TRUE,FALSE	FALSE	当前指令正在执行中，置为 TRUE
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ ERROR	参阅 SMC_ ERROR	0	异常发生时，输出错误代码
Position	获取到的轴位置	LREAL	轴位置	0	指令读出来的轴位置数据

3）功能说明

通过本指令读取驱动器中的实际位置指令，指令为 Enable 电平使能效应。指令可以重复多次使用，互不影响。

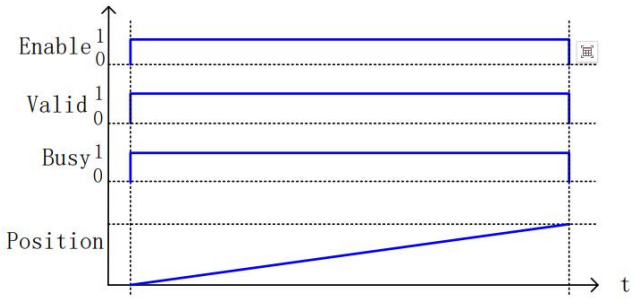
4）时序图

功能块的 Enable 必须为 TRUE 的条件；

功能块的 Valid 表示读出的 Position 为有效的数据值；

功能块的 Busy 表示当前功能块正在执行中；

时序操作说明 ：



4.1.12 MC_ReadAxisError

指令读取轴的错误情况，保存在自己定义的变量单元中。

1）指令格式

指令	名称	图形表现	ST 表现
MC_ReadAxisEr-ror	读轴的错误状态	MC_ReadAxisError MC_ReadAxisError Axis → Valid Enable → Busy Error ErrorID AxisError AxisErrorID SWEndSwitchActive	MC_ReadAxisError(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , AxisError=> , AxisErrorID=> , SWEndSwitchActive=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE,FALSE	FALSE	为 TRUE 状态读取伺服的当前位置

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Valid	错误数据可获取标志	BOOL	TRUE,FALSE	FALSE	能获取轴的错误数据，置为 TRUE
Busy	指令正在执行	BOOL	TRUE,FALSE	FALSE	当前指令正在执行中，置为 TRUE
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ ERROR	参阅 SMC_ ERROR	0	异常发生时，输出错误代码
AxisError	轴错误标示	BOOL	TRUE,FALSE	FALSE	读出轴为错误，对应的标示置位
AxisErrorID	轴错误码	DWORD	-	0	读出轴为错误码
SWEndSwitchActive	软限位开关有效	BOOL	TRUE,FALSE	FALSE	在指令读取中，检查软限位开关状态

3）功能说明

通过 MC_ReadAxisError 读取驱动器中的错误码，指令为 Enable 电平使能效应。指令可以重复多次使用， 互不影响。

4）时序图

功能块的 Enable 必须为 TRUE 的条件；

功能块的 Valid 表示读出的 AxisError 和 AxisErrorID 为有效的数据值；

功能块的 Busy 表示当前功能块正在执行中；

4.1.13 MC_ReadBoolParameter

指令读取驱动轴的位参数，保存在自己定义的变量单元中。

1）指令格式

指令	名称	图形表现	ST 表现
MC_ ReadBoolParameter	读取轴的位参数	MC_ReadBoolParameter MC_ReadBoolParameter Axis → Valid Enable → Busy ParameterNumber → Error ErrorID Value	MC_ReadBoolParameter(Axis:= , Enable:= , ParameterNumber:= , Valid=> , Busy=> , Error=> , ErrorID=> , Value=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE,FALSE	FALSE	为 TRUE 状态读取伺服的当前位置
ParameterNumber	轴参数的序号	DINT	-	0	访问轴参数的索引和子索引和序号

注 :ParameterNumber (DINT) = -DWORD_TO_DINT(SHL(USINT_TO_DOWRD(usiDataLength), 24)(对象字 典中的数据长度) + SHL(UINT_TO_DWORD(uiIndex), 8) (对象字典中的索引 -16BIT) + usisubIndex(对象字典中的子索引 -8BIT))

usiDataLength: 按字节数填写；1 字节为 16#01；2 字节为 16#02；4 字节为 16#04 等。

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Valid	位置数据可获取标志	BOOL	TRUE,FALSE	FALSE	能正确的获取驱动器的位置，置为 TRUE
Busy	指令正在执行	BOOL	TRUE,FALSE	FALSE	当前指令正在执行中，置为 TRUE
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ ERROR	参阅 SMC_ ERROR	0	异常发生时，输出错误代码
Value	获取到的轴位状态	BOOL	TRUE,FALSE	FALSE	指令读出来的轴位状态

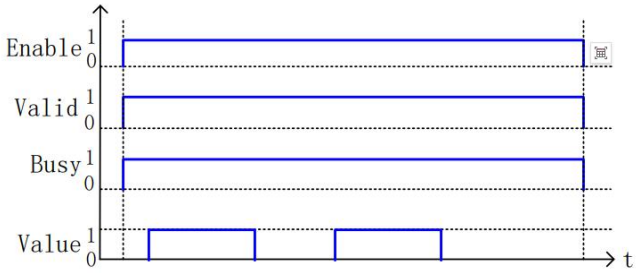
3）功能说明

通过 MC_ReadBoolParam 读取驱动器中的位数据状态，指令为 Enable 电平使能效应。指令可以重复多次 使用，互不影响。

4）时序图

功能块的 Enable 必须为 TRUE 的条件。 功能块的 Valid 表示读出的 Valid 为有效的位状态数据。 功能块的 Busy 表示当前功能块正在执行中。

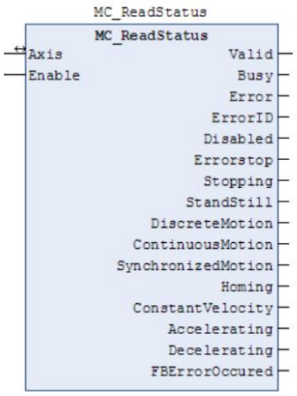
◆ 时序操作说明 :



4.1.14 MC_ReadStatus

指令读取轴的状态数据，保存在自己定义的变量单元中。

1）指令格式

指令	名称	图形表现	ST 表现
MC_ReadStatus	读轴的状态		MC_ReadStatus(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , Disabled=> , Errorstop=> , Stopping=> , StandStill=> , DiscreteMotion=> , ContinuousMotion=> , SynchronizedMotion=> , Homing=> , ConstantVelocity=> , Accelerating=> , Decelerating=> , FBErrorOccured=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE,FALSE	FALSE	为 TRUE 状态读取伺服的当前位置

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Valid	错误数据可获取标志	BOOL	TRUE,FALSE	FALSE	能获取轴的错误数据，置为 TRUE
Busy	指令正在执行	BOOL	TRUE,FALSE	FALSE	当前指令正在执行中，置为 TRUE
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ ERROR	参阅 SMC_ ERROR	0	异常发生时，输出错误代码
Disabled	轴没使能状态	BOOL	TRUE,FALSE	FALSE	轴在没使能状态为 TRUE;
Errorstop	轴错误状态	BOOL	TRUE,FALSE	FALSE	轴在错误运行状态为 TRUE;
Stoping	轴停止过程状态	BOOL	TRUE,FALSE	FALSE	轴在停止过程中为 TRUE

输出变量	名称	数据类型	有效范围	初始值	描述
StandStill	轴标准状态	BOOL	TRUE,FALSE	FALSE	轴在标准（能运行）状态中为 TRUE
DiscreteMotion	轴离散运动状态	BOOL	TRUE,FALSE	FALSE	轴在离散运动状态中为 TRUE
ContinuousMotion	轴连续运动状态	BOOL	TRUE,FALSE	FALSE	轴在连续运动状态中为 TRUE
SynchronizedMotion	轴同步运行状态	BOOL	TRUE,FALSE	FALSE	轴在同步运动状态中为 TRUE
Homing	轴回原点状态	BOOL	TRUE,FALSE	FALSE	轴在回原点状态中为 TRUE
ConstantVelocity	轴运行速度到达	BOOL	TRUE,FALSE	FALSE	轴达到运行速度中为 TRUE
Accelerating	轴加速过程状态	BOOL	TRUE,FALSE	FALSE	轴加速过程状态为 TRUE
Dccelerating	轴减速过程状态	BOOL	TRUE,FALSE	FALSE	轴减速过程状态为 TRUE
FBErrorOccured	轴功能块错误出现标志	BOOL	TRUE,FALSE	FALSE	轴功能块错误标志为 TRUE

3）功能说明

◆ 通过 MC_ReadStatus 对应轴的各种状态，指令为 Enable 电平使能效应。指令可以重复多次使用，互不影响。

◆ 功能块的 Enable 必须为 TRUE 的条件。

◆ 功能块的 Valid 表示读出的后面个状态标志的各种数据。

◆ 功能块的 Busy 表示当前功能块正在执行中。

4.1.15 MC_ReadParameter

指令读取驱动轴的参数，保存在自己定义的变量单元中。

1）指令格式

指令	名称	图形表现	ST 表现
MC_ ReadParameter	读取轴的参数	MC_ReadParameter MC_ReadParameter Axis, Enable, ParameterNumber → Valid, Busy, Error, ErrorID, Value	MC_ReadParameter(Axis:= , Enable:= , ParameterNumber:= , Valid=> , Busy=> , Error=> , ErrorID=> , Value=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE,FALSE	FALSE	为 TRUE 状态读取伺服的当前位置
ParameterNumber	轴参数的序号	DINT	-	0	访问轴参数的索引和子索引和序号

注 :ParameterNumber (DINT) = -DWORD_TO_DINT(SHL(USINT_TO_DOWRD(usiDataLength), 24)(对象字典中的数据长度)

+ SHL(UINT_TO_DWORD(uiIndex), 8) (对象字典中的索引 -16BIT)

+ usisubIndex(对象字典中的子索引 -8BIT)

usiDataLength: 按字节数填写；1 字节为 16#01；2 字节为 16#02；4 字节为 16#04 等。

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Valid	位置数据可获取标志	BOOL	TRUE,FALSE	FALSE	能正确的获取驱动器的位置，置为 TRUE
Busy	指令正在执行	BOOL	TRUE,FALSE	FALSE	当前指令正在执行中，置为 TRUE
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ ERROR	参阅 SMC_ ERROR	0	异常发生时，输出错误代码
Value	获取到的轴参数	LREAL	-	0	指令读出来的轴参数

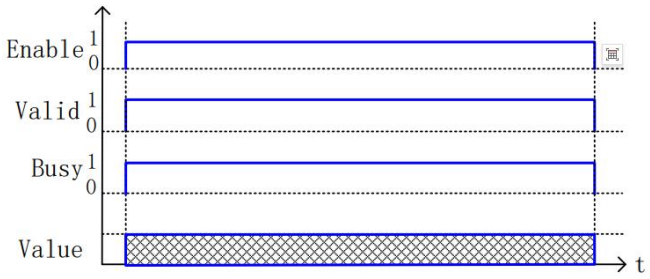
3）功能说明

通过 MC_ReadParam 读取驱动器中的位数据状态，指令为 Enable 电平使能效应。指令可以重复多次使用， 互不影响。

4）时序图

功能块的 Enable 必须为 TRUE 的条件； 功能块的 Valid 表示读出的 Valid 为有效的位状态数据； 功能块的 Busy 表示当前功能块正在执行中；

时序操作说明 ：



4.1.16 MC_Reset

1）指令格式

指令	名称	图形表现	ST 表现
MC_Reset	轴错误状态复位指令	MC_Reset MC_Reset Axis Done Execute Busy Error ErrorID	MC_Reset(Axis:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE,FALSE	FALSE	输入的一个上升沿将启动功能块的处理

◆ 输出变量

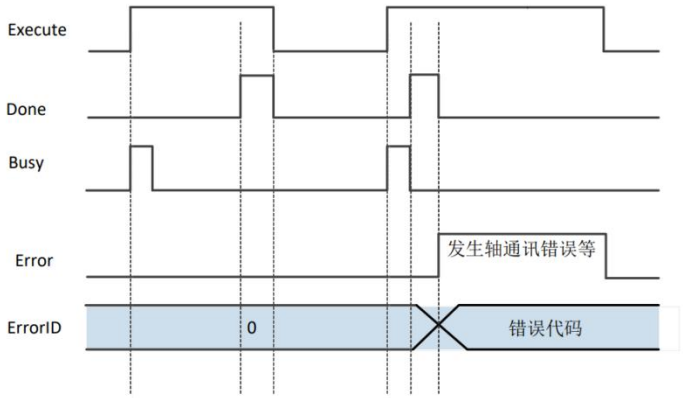
输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE,FALSE	FALSE	轴指令执行完成，置为 TRUE
Busy	指令正在执行	BOOL	TRUE,FALSE	FALSE	当前指令正在执行中，置为 TRUE
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ ERROR	参阅 SMC_ ERROR	0	异常发生时，输出错误代码

3）功能说明

◆ 本功能块在轴通讯正常的情况下 ， 把轴状态处于 errorstop 变为 Standstill, 把轴的异常状态变为正常可运行的状态 ；

◆ 当出现轴 errorstop 无法复位 ,Axis.bCommunication 为 FLASE 状态,必须要重新建立主站和从站轴的通讯 ； 在指令中的 Busy 标志位接通的时间非常短,使用时请注意 ；

4）时序图



4.1.17 MC_stop

1）指令格式

指令	名称	图形表现	ST 表现
MC_Stop	轴停止命令	MC_Stop MC_Stop Axis, Execute, Deceleration, Jerk → Done, Busy, Error, ErrorID	MC_Stop(Axis:= , Execute:= , Deceleration:= , Jerk:= , Done=> , Busy=> , Error=> , ErrorID=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE,FALSE	FALSE	输入的一个上升沿将启动功能块的处理
Deceleration	减速度	LREAL	“正数”+”0”	0	功能块的减速度 (u/s^2)
Jerk	跃度	LREAL	“正数”+”0”	0	指定跃度［指令单位 /s^3］

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE,FALSE	FALSE	轴指令执行完成，置为 TRUE

输出变量	名称	数据类型	有效范围	初始值	描述
Busy	指令正在执行	BOOL	TRUE,FALSE	FALSE	当前指令正在执行中，置为 TRUE
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ ERROR	参阅 SMC_ ERROR	0	异常发生时，输出错误代码

3）功能说明

◆ 本功能块为在正常运行的情况下停止一个轴的运动，当轴处于 stopping 状态对本轴的任何指令都是无效的。

◆ 当轴状态处于 stopping 时 ,Execute 为 Flase 状态 ,Done 输出状态为 True, 轴状态变为 Standstill;

◆ 本功能块运行状态为运行状态 (Motion) 才能运行 , 其他状态无法运行。

◆ 启动指令为 Execute 的上升沿启动。

◆ 在 MC_Stop 有效执行的过程 Busy 有效时 , 再一次启动 MC_Stop 指令系统处于 Errorstop 状态

◆ 在 Halt 和 Stop 状态时，通过轴变量 bAvoidReversalOnHaltStop 可以对加速度进行调整，避免出现速度逆转现象，具体分析参考 MC_Halt 指令。

4）时序图

轴必须处于运行状态 (Motion) 指令才能运行。

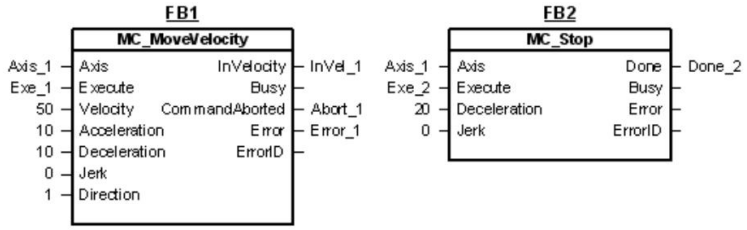
功能块的 Execute 必须有上升沿的条件。

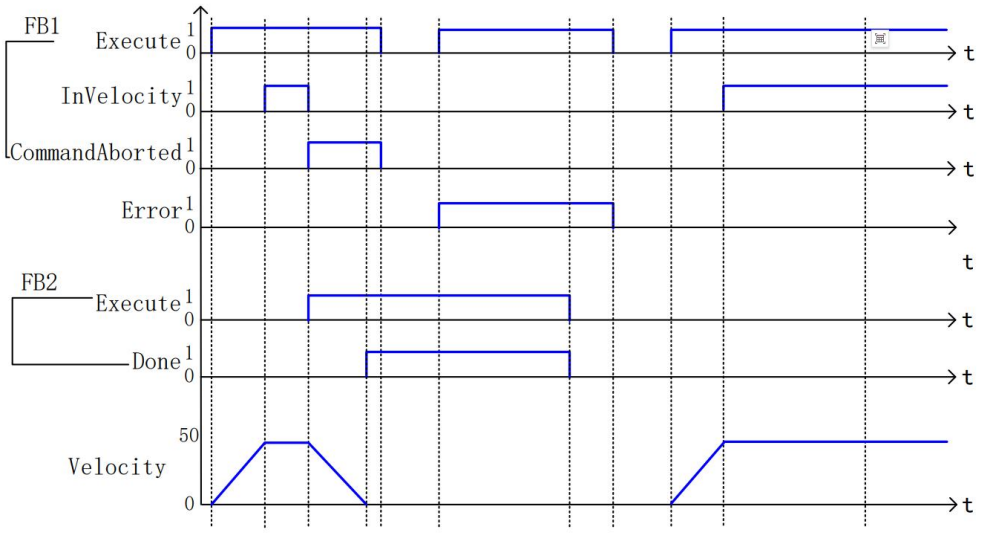
功能块的 Done 表示指令正常执行完成。

功能块的 Busy 表示当前功能块正在执行中。

功能块的 CommandAborted 表示指令被其他运动控制指令中断，此时标志位为 TRUE。

例程:在执行 MC_MoveVelocity 指令和 MC_Stop 指令在不同的时序操作中对应的标志位的变化。对 CommandAborted 的处理描述如下图的时序描述。





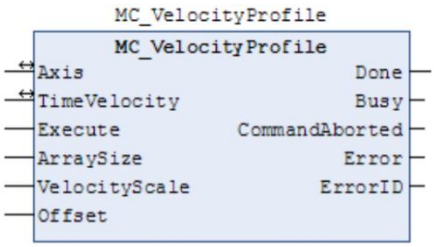
错误说明:

MC_Stop 有重复性的指令运行时,错误标志 Error 为 True,ErrorID 为 SMC_MS_AXI 错误 ;

【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.1.18 MC_VelocityProfile

1）指令格式

指令	名称	图形表现	ST 表现
MC_ VelocityProfile	速度轮廓指令		MC_VelocityProfile(Axis:= , TimeVelocity:= , Execute:= , ArraySize:= , VelocityScale:= , Offset:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

输入输出变量	名称	数据类型	有效范围	初始值	描述
TimeVelocity	轴速度运行时间和速度描述	MC_TV_REF	-	-	轴速度运行时间和速度数据描述，由多组数据组成

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE,FALSE	FALSE	输入的一个上升沿将启动功能块的处理
ArraySize	动态数组	INT	数据范围	0	运行轮廓中使用的数组个数
VelocityScale	速度因子	LREAL	“正数”，”0”	1	速度的比例因子
Offset	偏移	LREAL	-	0	速度值的总体偏移值

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE,-FALSE	FALSE	指令执行完成，置为 TRUE
Busy	指令正在执行	BOOL	TRUE,-FALSE	FALSE	当前指令正在执行中，置为 TRUE
CommandAbort	指令被中断	BOOL	TRUE,-FALSE	FALSE	当前指令被中断，置为 TRUE
Error	错误	BOOL	TRUE,-FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ER-ROR	参阅 SMC_ERROR	0	异常发生时，输出错误代码

3）功能说明

◆ 本功能块为时间段和速度的轮廓运动模型，运行模式 Continuous Motion，按用户在 TimeVelocity 变量中设定的数据运行。

◆ 本功能块运行状态为 Standstill 中 ，指令运行时的状态为 Discrete Motion，其他状态无法运行。

◆ 启动指令为 Execute 的上升沿启动 ， 本指令在 Discrete Motion 重复运行。

◆ TimeVelocity 为 MC_TV_REF 数据类型 ；

MC_TV_REF 具体描述如下 ：

成员	类型	初始值	描述
Number_of_pairs	INT	0	轮廓路径的段数
IsAbsolute	BOOL	TRUE	绝对运动 (TRUE) 和相对运动选择
MC_TV_Array	ARRAY[1..N] OF SMC_TV	-	时间和速度的数组

SMC_TV 具体描述如下 ：

成员	类型	初始值	描述
delta_time	TIME	TIME#0ms	速度值段的时间
Velocity	LREAL	0	当前记录的速度值

注 ： 整个速度的过程为 S 曲线加减速的方式，每段轮廓都速度都为叠加的计算方式 ； 在指令重复运行时，速度也为叠加方式，在指令使用时避免速度超限的出现；重复运行一定把本轴的状态重回到 Standstill 状态。

4）时序图

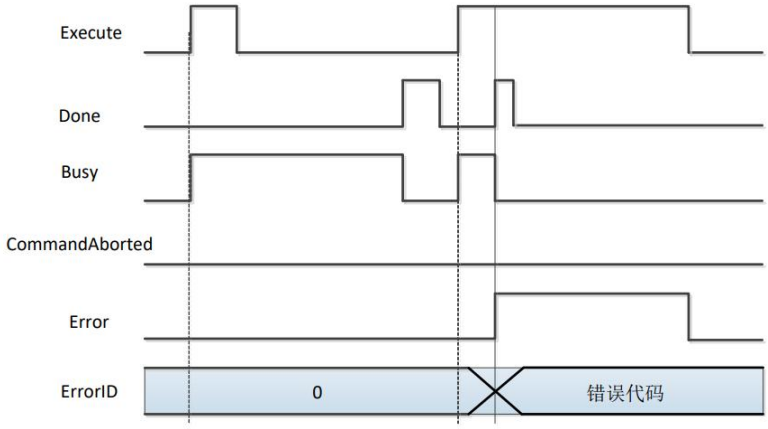
条件 MC_TV_Array 通过其他方式已经设置才能运行位置轮廓曲线指令。

轴必须处于 Standstill 状态指令才能运行。

功能块的 Execute 必须有上升沿的条件。

功能块的 Done 表示指令正常执行完成。

功能块的 Busy 表示当前功能块正在执行中。



5）错误说明

错误的出现为轴状态不是在 Standstill 中启动指令或指令系统中的参数错误，出现轴错误只能清除错误后才开始运行。

【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.1.19 MC_WriteBoolParameter

指令设置驱动轴的位参数。

1）指令格式

指令	名称	图形表现	ST 表现
MC_ WriteBoolParameter	设置轴的位参数	MC_WriteBoolParameter MC_WriteBoolParameter Axis Done Execute Busy ParameterNumber Error Value ErrorID	MC_WriteBoolParameter(Axis:= , Execute:= , ParameterNumber:= , Value:= , Done=> , Busy=> , Error=> , ErrorID=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE,FALSE	FALSE	为上升沿操作驱动一次设置操作
ParameterNumber	轴参数的序号	DINT	-	0	访问轴参数的索引和子索引和序号
Value	设定值	BOOL	TRUE,FALSE	FALSE	设置位参数值

注 :ParameterNumber(DINT) =-DWORD_TO_DINT(SHL(USINT_TO_DOWRD(usiDataLength),24)(对象字典中的数据长度)

+ SHL(UINT_TO_DWORD(uiIndex),8) (对象字典中的索引-16BIT)

+ usisubIndex(对象字典中的子索引-8BIT)

usiDataLength: 按字节数填写；1 字节为 16#01；2 字节为 16#02；4 字节为 16#04 等。

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	设置操作成功	BOOL	TRUE,FALSE	FALSE	设置操作成功置为 TRUE
Busy	指令正在执行	BOOL	TRUE,FALSE	FALSE	当前指令正在执行中，置为 TRUE
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ ERROR	参阅 SMC_ ERROR	0	异常发生时，输出错误代码

3）功能说明

通过 MC_WriteBoolParameter 设置轴的位参数，指令为 Execute 上升沿触发。指令可以重复多次使用，互不影响。

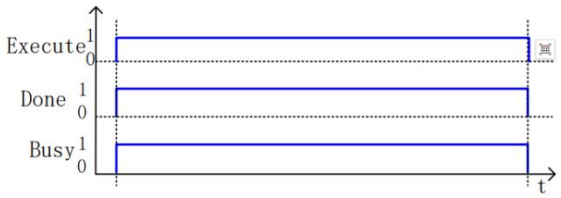
4）时序图

功能块的 Execute 必须为上升沿触发条件。

功能块的 Done 表示设置操作成功。

功能块的 Busy 表示当前功能块正在执行中。

◆ 时序操作说明 ：



4.1.20 MC_WriteParameter

指令修改驱动轴的参数，保存在自己定义的变量单元中。

1）指令格式

指令	名称	图形表现	ST 表现
MC_ WriteParameter	设置轴参数	MC_WriteParameter MC_WriteParameter Axis, Execute, ParameterNumber, Value → Done, Busy, Error, ErrorID	MC_WriteParameter(Axis:= , Execute:= , ParameterNumber:= , Value:= , Done=> , Busy=> , Error=> , ErrorID=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE,FALSE	FALSE	为上升沿操作驱动一次设置操作
ParameterNumber	轴参数的序号	DINT	-	0	访问轴参数的索引和子索引和序号
Value	设定值	LREAL	-	-	设置位参数值

注:ParameterNumber(DINT) =-DWORD_TO_DINT(SHL(USINT_TO_DOWRD(usiDataLength),24)(对象字典中的数据长度)

+ SHL(UINT_TO_DWORD(uiIndex),8) (对象字典中的索引-16BIT)

+ usisubIndex(对象字典中的子索引-8BIT)

usiDataLength:按字节数填写；1 字节为 16#01；2 字节为 16#02；4 字节为 16#04 等。

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	设置操作成功	BOOL	TRUE,FALSE	FALSE	设置操作成功置为 TRUE
Busy	指令正在执行	BOOL	TRUE,FALSE	FALSE	当前指令正在执行中，置为 TRUE
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ ERROR	参阅 SMC_ ERROR	0	异常发生时，输出错误代码

3）功能说明

通过 MC_ WriteParameter 设置轴的位参数，指令为 Execute 上升沿触发。指令可以重复多次使用，互不影响。

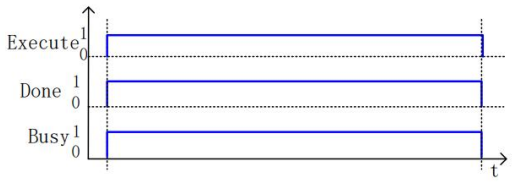
4）时序图

功能块的 Execute 必须为上升沿触发条件。

功能块的 Done 表示设置操作成功。

功能块的 Busy 表 示当前功能块正在执行中。

◆ 时序操作说明 ：



4.1.21 MC_AbortTrigger

功能块终止输入锁存相关事件的关联特性，和 MC_Touchprobe 配套使用。

1）指令格式

指令	名称	图形表现	ST 表现
MC_AbortTrigger	功能块终止事件关联	MC_AbortTrigger MC_AbortTrigger Axis TriggerInput Execute Done Busy Error ErrorID	MC_AbortTrigger(Axis:= , TriggerInput:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例
TruggerInput	触发信号	TRIGGER_REF	-	-	触发信号、触发属性等描述

◆ TRIGGER_REF 说明 ：

结构体	元素	数据类型	初始值	描述
TRIGGER_REF	iTriggerNumber	INT	-1	在驱动器模式下，锁定功能中的哪一个。 0：探针 1 上升沿锁存 1：探针 1 下降沿锁存 2：探针 2 上升沿锁存 3：探针 2 下降沿锁存 详细参考《IS620N 系列伺服设计维护使用手册》
	bFastLatching	BOOL	TRUE	指定锁存触发的模式： TRUE：驱动器模式 FALSE：控制器模式
	bInput	BOOL	-	当 bFastLatching=FLASE 时，由控制器 Input 信号触发
	bActive	BOOL	-	触发的有效信号

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE,FALSE	FALSE	为上升沿操作驱动一次设置操作

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	设置操作成功	BOOL	TRUE,FALSE	FALSE	设置操作成功置为 TRUE
Busy	指令正在执行	BOOL	TRUE,FALSE	FALSE	当前指令正在执行中，置为 TRUE
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ERROR	参阅 SMC_ERROR	0	异常发生时，输出错误代码

3）功能说明

通过 MC_AbortTrigger 功能块是把触发信号或属性和相关的触发指令终止其关联操作。

功能块的 Execute 必须为上升沿触发条件。

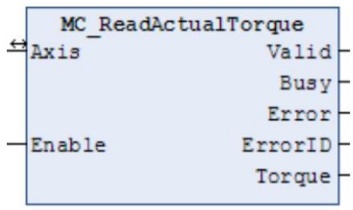
功能块的 Done 表示设置操作成功。

功能块的 Busy 表示当前功能块正在执行中。

4.1.22 MC_ReadActualTorque

指令读取驱动器运行的当前力矩值，读取的当前力矩值保存在自己定义的变量单元中。

1）指令格式

指令	名称	图形表现	ST 表现
MC_ReadActualTorque	当前力矩值读取指令		MC_ReadActualTorque0(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , Torque=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE,FALSE	FALSE	为 TRUE 状态读取伺服的当前位置

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Valid	当前力矩值可获取标志	BOOL	TRUE,FALSE	FALSE	能正确的获取驱动器的力矩值，置为 TRUE
Busy	指令正在执行	BOOL	TRUE,FALSE	FALSE	当前指令正在执行中，置为 TRUE
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ ERROR	参阅 SMC_ ERROR	0	异常发生时，输出错误代码
Torque	获取的当前力矩值	LREAL	力矩值	0	指令读出来的当前力矩数据

3）功能说明

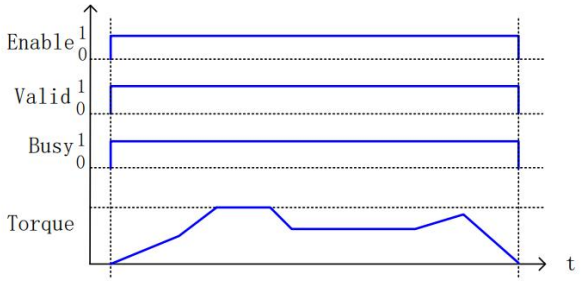
通过 MC_ReadActualTorque 读取驱动器中的当前力矩值指令，指令为 Enable 电平使能效应。指令可以重 复多次使用,互不影响。

4）时序图

功能块的 Enable 必须为 TRUE 的条件。

功能块的 Valid 表示读出的 Torque 为有效的数据值。

功能块的 Busy 表示当前功能块正在执行中。



4.1.23 MC_ReadActualVelocity

指令读取驱动器运行的当前速度值，读取的当前速度值保存在自己定义的变量单元中。

1）指令格式

指令	名称	图形表现	ST 表现
MC_ ReadActualVelocity	当前速度读取指令	MC_ReadActualVelocity Axis → Valid Busy Error Enable → ErrorID Velocity	MC_ReadActualVelocity0(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , Velocity=>);

2） 相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE,FALSE	FALSE	为 TRUE 状态读取当前轴速度

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Valid	当前速度值可获取标志	BOOL	TRUE,FALSE	FALSE	能正确的获取驱动器的速度值，置为 TRUE
Busy	指令正在执行	BOOL	TRUE,FALSE	FALSE	当前指令正在执行中，置为 TRUE
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ ERROR	参阅 SMC_ ERROR	0	异常发生时，输出错误代码
Velocity	获取的当前速度值	LREAL	速度值	0	指令读出来的当前速度数据

3）功能说明

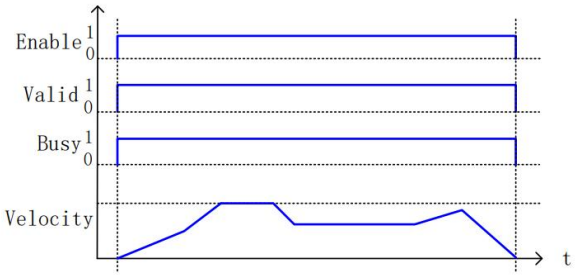
通过 MC_ReadActualVelocity 读取驱动器中的当前速度值指令，指令为 Enable 电平使能效应。指令可以重 复多次使用，互不影响。

4）时序图

功能块的 Enable 必须为 TRUE 的条件。

功能块的 Valid 表示读出 Velocity 为有效的数据值。

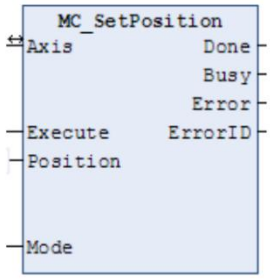
功能块的 Busy 表示当前功能块正在执行中。



4.1.24 MC_SetPosition

将指令中的位置数据设为当前轴的位置数据，对设置位置数据操作不会产生任何位移移动，用于产生坐标系 的位移。

1）指令格式

指令	名称	图形表现	ST 表现
MC_ SetPosition	设置轴的位置参数		MC_SetPosition0(Axis:= , Execute:= , Position:= , Mode:= , Done=> , Busy=> , Error=> , ErrorID=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE,FALSE	FALSE	为上升沿操作驱动一次设置操作
Position	轴位置数据	LREAL	-	0	位置数据
Mode	设定值	BOOL	TRUE,FALSE	FALSE	位置模式；TRUE：相对位置 (RELATIVE)； FALSE：绝对位置 (ABSOLUTE)；

◆ 输出变量

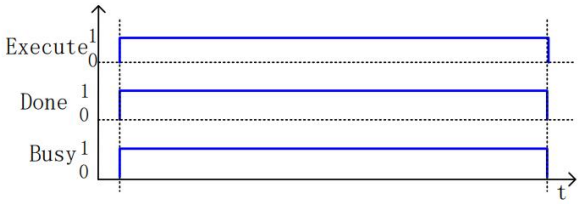
输出变量	名称	数据类型	有效范围	初始值	描述
Done	设置操作成功	BOOL	TRUE,FALSE	FALSE	设置操作成功置为 TRUE
Busy	指令正在执行	BOOL	TRUE,FALSE	FALSE	当前指令正在执行中，置为 TRUE
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ERROR	参阅 SMC_ERROR	0	异常发生时，输出错误代码

3）功能说明

通过 MC_SetPosition 设置轴的位置参数，不产生任何位移，但形成了坐标偏移；指令为 Execute 上升沿触发； 指令可以重复多次使用，互不影响。

4） 时序图

功能块的 Execute 必须为上升沿触发条件。功能块的 Done 表示设置操作成功。 功能块的 Busy 表示当前功能块正在执行中。



4.1.25 MC_TouchProbe

指令通过外部信号触发，保存当前轴的位置数据。

1）指令格式

指令	名称	图形表现	ST 表现
MC_TouchProbe	启用外部锁定	MC_TouchProbe MC_TouchProbe Axis → Done TriggerInput → Busy Execute → Error WindowOnly → ErrorID FirstPosition → RecordedPosition LastPosition → CommandAborted	MC_TouchProbe(Axis:= , TriggerInput:= , Execute:= , WindowOnly:= , FirstPosition:= , LastPosition:= , Done=> , Busy=> , Error=> , ErrorID=> , RecordedPosition=> , CommandAborted=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例
TruggerInput	触发信号	TRIGGER_REF	-	-	触发信号或触发属性等关联属性

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE,FALSE	FALSE	为上升沿操作驱动一次设置操作
WindowOnly	触发窗口	BOOL	TRUE,FALSE	FALSE	
FirstPosition	触发开始位置	LREAL	-	0	指定接收触发的开始位置
LastPosition	触发结束位置	LREAL	-	0	指定接收触发的结束位置

◆ 输出变量

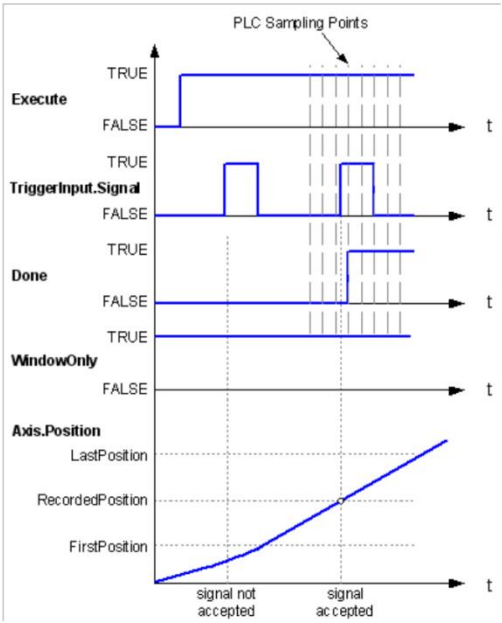
输出变量	名称	数据类型	有效范围	初始值	描述
Done	设置操作成功	BOOL	TRUE,FALSE	FALSE	设置操作成功置为 TRUE
Busy	指令正在执行	BOOL	TRUE,FALSE	FALSE	当前指令正在执行中，置为 TRUE
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ ERROR	参阅 SMC_ ERROR	0	异常发生时，输出错误代码
RecordedPosition	触发记录位置	LREAL	-	0	触发发生时当前的位置
CommandAbort	指令被中断	BOOL	TRUE,FALSE	FALSE	当前指令被中断，置为 TRUE

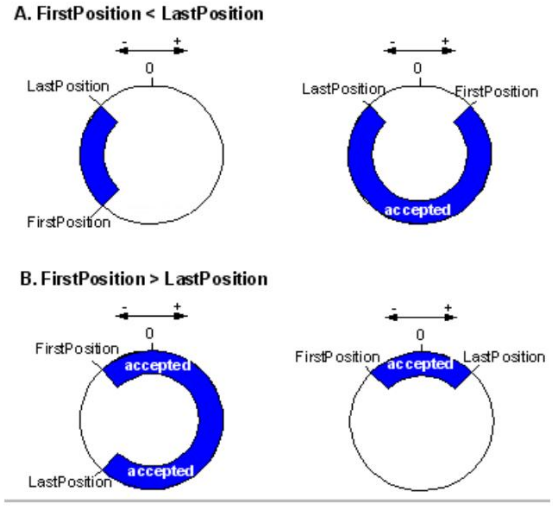
3）功能说明

◆ 通过 MC_TouchProbe 功能块 TruggerInput 的信号触发时记录运行轴的当前位置。

◆ Execute 上升沿执行执行 。

◆ 驱动器锁存时：驱动器采集到锁存信号会在记录位置，等到控制器。

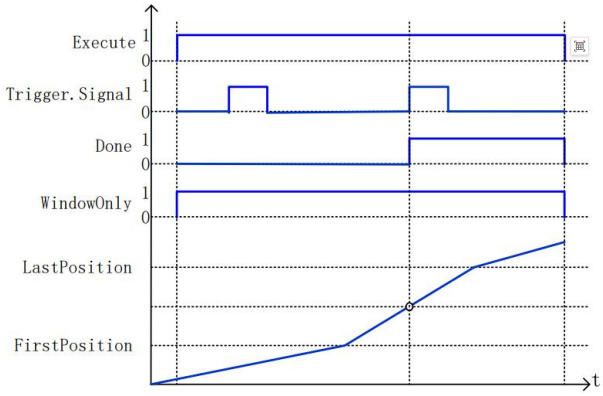




4）时序图

功能块的 Execute 必须为上升沿触发条件。

功能块的 Done 表示设置操作成功。

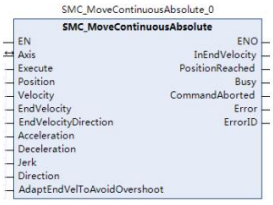


在 Position 为循环计数单元，触发信号的位置可以重复使用。

4.1.26 SMC_MoveContinuousAbsolute

轴按绝对位置连续运行（单位按轴设置），绝对位置由 Distance 指定，最后的运行速度 EndVelocity 运行。

1）指令格式

指令	名称	图形表现	ST 表现
MC_MoveContinousAbsolute	轴绝对位置连续控制指令		

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE,FALSE	FALSE	输入的一个上升沿将启动功能块的处理
Position	运动绝对位置	LREAL	数据范围	0	此数据为运动的绝对位置
Velocity	运行速度	LREAL	数据范围	0	轴运行到目标位置的最大速度
EndVelocity	运行结束速度	LREAL	数据范围	0	指令执行完成后的运行速度
EndVelocity-Direction	结束速度的方向	MC_Direction	positive, negative, current;	Current	可以使用：positive, negative, current; 不可使用 : shortest, fastest
Acceleration	加速度	LREAL	数据范围	0	速度变大时加速度值
Deceleration	减速度	LREAL	数据范围	0	速度变小时减速度值
Jerk	跃度	LREAL	数据范围	0	曲线加减速的斜率变化值
Direction	运行方向	shortest	数据范围	shortest	对于线性 / 直线轴：positive, negative; 对于旋转 / 圆周轴：positive, negative, current, shortest, fastest
AdaptEndVel ToAvoidOvershoot	末端速度调整标志	BOOL	TRUE,FALSE	FALSE	TRUE: 调整末端速度以避免过冲 FALSE: 无处理

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
InEndVelocity	指令位置到达	BOOL	TRUE,FALSE	FALSE	轴指令执行位置到达，置为 TRUE
Busy	指令正在执行	BOOL	TRUE,FALSE	FALSE	当前指令正在执行中，置为 TRUE
CommandAbort	指令被中断	BOOL	TRUE,FALSE	FALSE	当前指令被中断，置为 TRUE
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE

输出变量	名称	数据类型	有效范围	初始值	描述
ErrorID	错误代码	SMC_ERROR	参阅 SMC_ERROR	0	异常发生时，输出错误代码

3）功能说明

◆ 本功能块为轴绝对定位指令 ， Distance 数据为轴的绝对位置。

◆ 本功能块运行状态为 Standstill 中 ， 指令运行时的状态为 Discrete Motion, 一个完整的运行过程一定要控制轴的不同运动状态。

◆ 启动指令为 Execute 的上升沿启动 ， 本指令在 Discrete Motion 可以重复上升沿有效 ， 每次都可以刷新最新的 Position 位置。

◆ Acceleration 或 Deceleration 为零，指令运行都为异常状态，但轴的状态为 Discrete Motion。

◆ AdaptEndVelToAvoidOvershoot 为 TRUE，则将调整有效末端速度以避免过冲。在某些情况下，给定的 EndVelocity 可能会导致位置超调。例如，如果剩余距离太短而无法从当前速度和加速度达到最终速度，则轴可能会向相反方向转动，从而导致位置超调。另一种情况是终点速度与要运行的距离方向相反。若要以这个速度到达目标位置，轴首先会超过距离，反向并加速到终点速度，从而再次导致位置超调。下面是可以避免过冲的两种情况：

如果剩余距离太短而无法达到所需的 EndVelocity。在这种情况下，以达到末速度为主，不考虑目标位置限制，从而实际运行的位移会大于目标位置，即先到达目标位置（DistanceTravelled=TRUE），再到达目标速 度（InTenVelocity=TRUE）。

如果终点速度的方向与运动的方向相反。在这种情况下，将目标位置对应的末端速度设置为零，再从 0 运行至期望末速度。请注意，运动的方向不是由当前的速度和加速度决定的，而是由距离目标位置的符号决定的。

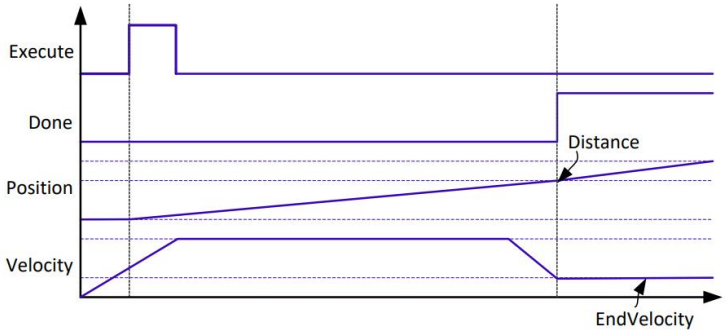
4）时序图

轴必须处于 Standstill 状态指令才能运行。

功能块的 Execute 必须有上升沿的条件。

功能块的 Done 表示指令正常执行完成。

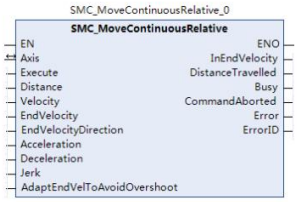
功能块的 Busy 表示当前功能块正在执行中。



4.1.27 SMC_MoveContinuousRelative

轴按相对位置连续运行（单位按轴设置），绝对位置由 Distance 指定，最后的运行速度 EndVelocity 运行。

1）指令格式

指令	名称	图形表现	ST 表现
MC_ MoveContinuousRelative	轴相对定位指令		SMC_MoveContinuousRelative_0(Axis:= Axis, Execute:= , Distance:= , Velocity:= , EndVelocity:= , EndVelocityDirection:= , Acceleration:= , Deceleration:= , Jerk:= , AdaptEndVelToAvoidOvershoot:= , InEndVelocity=> , DistanceTravelled=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE,FALSE	FALSE	输入的一个上升沿将启动功能块的处理
Distance	运动相对位置	LREAL	数据范围	0	此数据为运动的相对位置
Velocity	运行速度	LREAL	数据范围	0	轴运行到目标位置的最大速度
EndVelocity	运行结束速度	LREAL	数据范围	0	指令执行完成后的运行速度
EndVelocityDirection	结束速度的方向	MC_Direction	positive, negative, current;	Current	可以使用：positive, negative, current; 不可使用 : shortest, fastest
Acceleration	加速度	LREAL	数据范围	0	速度变大时加速度值
Deceleration	减速度	LREAL	数据范围	0	速度变小时减速度值
Jerk	跃度	LREAL	数据范围	0	曲线加减速的斜率变化值
AdaptEndVelTo AvoidOvershoot	末端速度调整标志	BOOL	TRUE,FALSE	FALSE	TRUE: 调整末端速度以避免过冲 FALSE: 无处理

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
InEndVelocity	指令位置到达	BOOL	TRUE,FALSE	FALSE	轴指令执行位置到达，置为 TRUE
Busy	指令正在执行	BOOL	TRUE,FALSE	FALSE	当前指令正在执行中，置为 TRUE
CommandAbort	指令被中断	BOOL	TRUE,FALSE	FALSE	当前指令被中断，置为 TRUE
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ ERROR	参阅 SMC_ ERROR	0	异常发生时，输出错误代码

3）功能说明

◆ 本功能块运行状态为 Standstill 中， 指令运行时的状态为 Discrete Motion，在指令执行中关注本轴的运行状态，避免打断本轴的其他指令或被其他指令打断本轴的执行。

◆ 启动指令为 Execute 的上升沿启动 ， 本指令在 Discrete Motion 可以重复上升沿有效 ， 每次都可以刷新最新的 Position 位置。

◆ Acceleration 或 Deceleration 为零，指令运行都为异常状态，但轴的状态为 Discrete Motion；

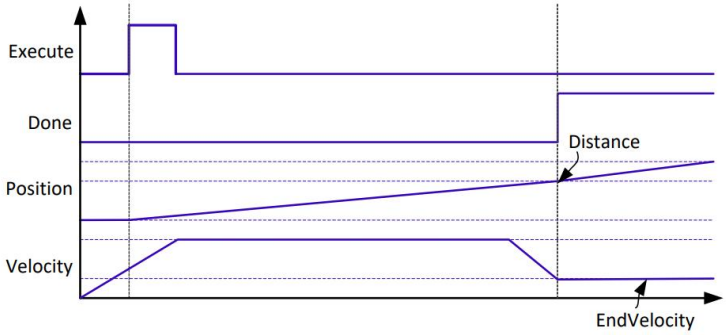
◆ AdaptEndVelToAvoidOvershoot 为 TRUE，则将调整有效末端速度以避免过冲。

4）时序图

功能块的 Execute 必须有上升沿的条件。

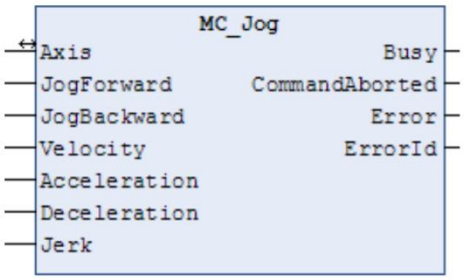
功能块的 Done 表示指令正常执行完成。

功能块的 Busy 表示当前功能块正在执行中。



4.1.28 SMC_Jog

1）指令格式

指令	名称	图形表现	ST 表现
MC_Jog	轴点动指令		MC_Jog(Axis:= , JogForward:= , JogBackward:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Busy=> , CommandAborted=> , Error=> , ErrorId=>);

2）相关变量

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
JogForward	正向有效	BOOL	TRUE,FALSE	FALSE	设置为 TRUE 则开始正向移动；设置为 FALSE 则停止正向移动
JogBackward	负向有效	BOOL	TRUE,FALSE	FALSE	设置为 TRUE 则开始反向移动；设置为 FALSE 则停止反向移动
Velocity	目标速度	LREAL	正数或“0”	0	指定目标速度。单位：[指令单位 /s]
Acceleration	加速度	LREAL	正数或“0”	0	指定加速度。单位：[指令单位 /s]
Deceleration	减速度	LREAL	正数或“0”	0	指定减速度。单位：[指令单位 /s]

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Busy	执行中	BOOL	TRUE,FALSE	FALSE	接收指令后，置为 TRUE
CommandAborted	执行中断	BOOL	TRUE,FALSE	FALSE	指令中止时，置为 TRUE
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ERROR	参阅 SMC_ERROR	0	异常发生时，输出错误代码

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

3）功能说明

◆ 根据指定 Velocity（目标速度）执行点动运行。

◆ 需要正向运行时，将 JogForward（正向运行有效）置为 TRUE；需要反向运行时，将 JogBackward（负向 运行有效）置为 TRUE。

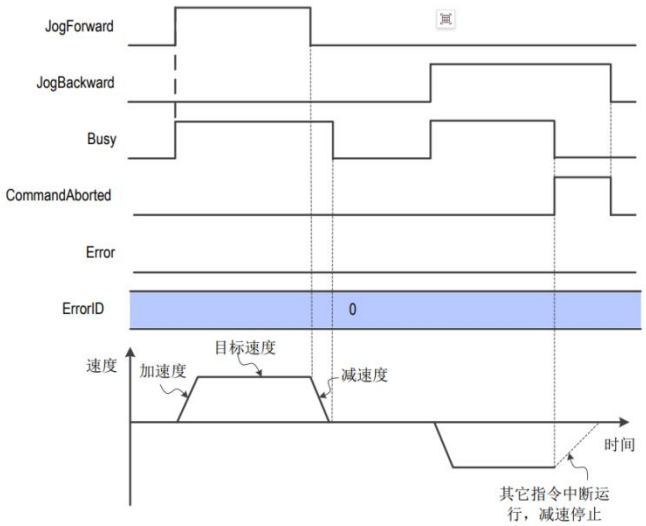
◆ 同时将 JogForward（正向运行有效）和 JogBackward（负向运行有效）置为 TRUE，将不会有运动发生。 如果 MC_Jog 指令的指令速度设置值超过轴参数中的点动最高速度，则以点动最高速度执行。

4）时序图

在启动 JogForward（正向运行有效）或 JogBackward（负向运行有效）的同时，Busy（执行中）变为 TRUE。

在 JogForward（正向运行有效）或 JogBackward（负向运行有效）的下降沿开始减速并停止轴的同时，Busy（执行中）变为 FALSE。

利用其它指令中止本指令时，CommandAborted（执行中断）变为 TRUE，Busy（执行中）变为 FALSE。

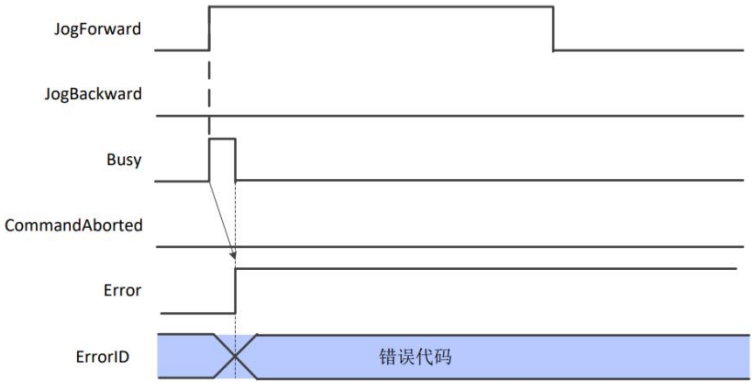


5）错误说明

在执行本指令中发生异常时，Error（错误）变成 TRUE，轴停止动作。

可查看 ErrorID（错误代码）的输出值，了解发生异常的原因。

◆ 发生异常时的时序图

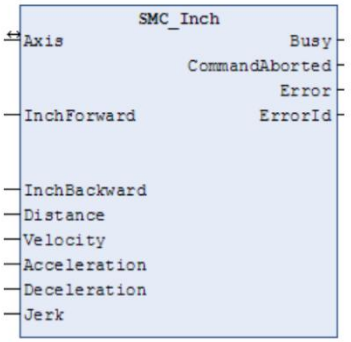


关于指令发生的异常，请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.1.29 SMC_Inch

轴单步运动控制，通过程序能实现一步接一步的单步控制。

1）指令格式

指令	名称	图形表现	ST 表现
SMC_Inch	轴相对定位指令		SMC_Inch0(Axis:= , InchForward:= , InchBackward:= , Distance:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Busy=> , CommandAborted=> , Error=> , ErrorId=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
InchForward	正向执行	BOOL	TRUE,FALSE	FALSE	如果 InchForward 为 TRUE，轴将按照给定的速度运动 (Velocity, Acceleration, Deceleration) 以正向直到到达距离。输入必须被指定为 FALSE 然后再为 TRUE 再次启动运动。 如果 InchForward 在到达位置之前被设置为 FALSE ，那么轴将立即减速到 0 并且 Busy 将会被设置为 FALSE。 如果输入 InchBackward 在仿真情况下被设置为 TRUE，那么将不会有运动产生。
InchBackward	反向执行	BOOL	TRUE,FALSE	FALSE	如果 InchBackward 为 TRUE，轴将会按照给定的速度值进行运动 (Velocity, Acceleration, Deceleration) 以正向运动到设定位置。然后输入必须被设置为 FALSE 然后再设置为 TRUE 启动另一个运动。 如果输入信号 InchForward 同时被设置为 TRUE，那么将不会有轴运动。
Distance	移动的距离	LREAL	数据范围	0	此数据为运动的距离
Velocity	运行速度	LREAL	数据范围	0	轴运行到目标位置的最大速度
Acceleration	加速度	LREAL	数据范围	0	速度变大时加速度值
Deceleration	减速度	LREAL	数据范围	0	速度变小时减速度值

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Busy	指令正在执行	BOOL	TRUE,FALSE	FALSE	当前指令正在执行中，置为 TRUE
CommandAbort	指令被中断	BOOL	TRUE,FALSE	FALSE	当前指令被中断，置为 TRUE

输出变量	名称	数据类型	有效范围	初始值	描述
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ ERROR	参阅 SMC_ ERROR	0	异常发生时，输出错误代码

3）功能说明

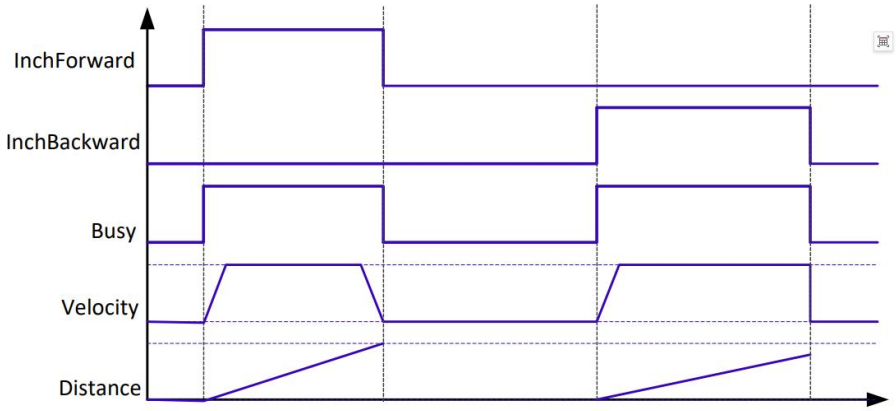
◆ 本功能块运行状态为 Standstill 中，指令运行时的状态为 Discrete Motion，在指令执行中关注本轴的运行状态，避免打断本轴的其他指令或被其他指令打断本轴的执行。

◆ Acceleration 或 Deceleration 为零，指令运行都为异常状态，但轴的状态为 Discrete Motion。

4）时序图

功能块的 InchForward/ InchBackward 必须有 TRUE/FALSE 的条件。

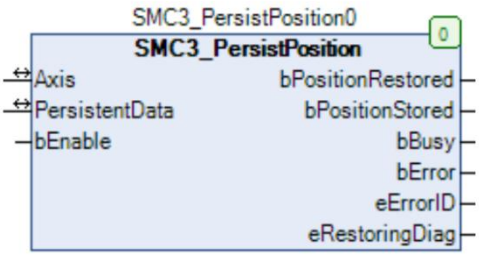
功能块的 Busy 表示当前功能块正在执行中。



4.1.30 SMC3_PersistPosition

该指令用来保持纪录实轴绝对值编码器的位置（断电重启控制器后，恢复断电前位置记录值）。如果伺服电 机使用的是绝对值编码器，使用该功能块配合使用。

1）指令格式

指令	名称	图形表现	ST 表现
SMC3_ PersistPosition	轴位置保持		SMC3_PersistPosition0(Axis:= , PersistentData:= , bEnable:= , bPositionRestored=> , bPositionStored=> , bBusy=> , bError=> , eErrorID=> , eRestoringDiag=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例
PersistentData	保持数据	SMC3_ PersistPosition_Data	-	-	存储位置信息的断电保持型数据结构

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
bEnable	执行	BOOL	TRUE,FALSE	FALSE	True 功能块执行，false 不执行功能块 若要在初始化期间还原上次存储的位置，则必须从应用程序启动时将该值置为 true

◆ 输出变量

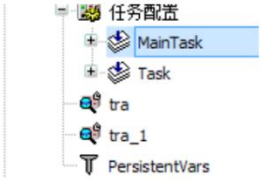
输出变量	名称	数据类型	有效范围	初始值	描述
bPositionRestored	位置恢复	BOOL	TRUE,FALSE	FALSE	TRUE, 轴重启后位置恢复完成
bPositionStored	位置保存	BOOL	TRUE,FALSE	FALSE	TRUE, 调用功能块后保存位置完成
bBusy	FB 执行中	BOOL	TRUE,FALSE	FALSE	TRUE，功能块没有执行完成
bError	错误	BOOL	TRUE,FALSE	FALSE	TRUE, 异常发生
eErrorID	错误代码	SMC_ERROR	-	SMC_NO_ERROR	异常发生时，输出错误代码
eRestoringDiag	恢复诊断	SMC3_ Persist-PositionDiag		SMC3_ PersistPositionDiag. SMC3_PPD_ RESTORING_OK	位置恢复中的诊断信息 SMC3_PPD_RESTORING_OK: 位置成功恢复 SMC3_PPD_AXIS_PROP_ CHANGED: 轴参数有更改，无法恢复位置 SMC3_PPD_DATA_STORED_ DURING_WRITING: 功能块从轴参数数据结构复制数据，而不是从 PersistentData 数据中复制。可能原因：非同步性持续变量、控制器崩溃死机

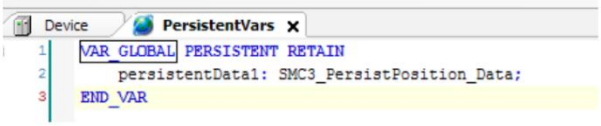
3）功能说明

◆ PLC 重启 bEnable 信号为 TRUE，则 bPositionRestroed 输出为 TRUE。

◆ 不支持虚轴跟逻辑轴。

① PersistentVars 中声明 SMC3_PersistPosition_Data 型数据





② PLC 主任务（EthCat 任务）中调用

◆ 申明部分：

VAR

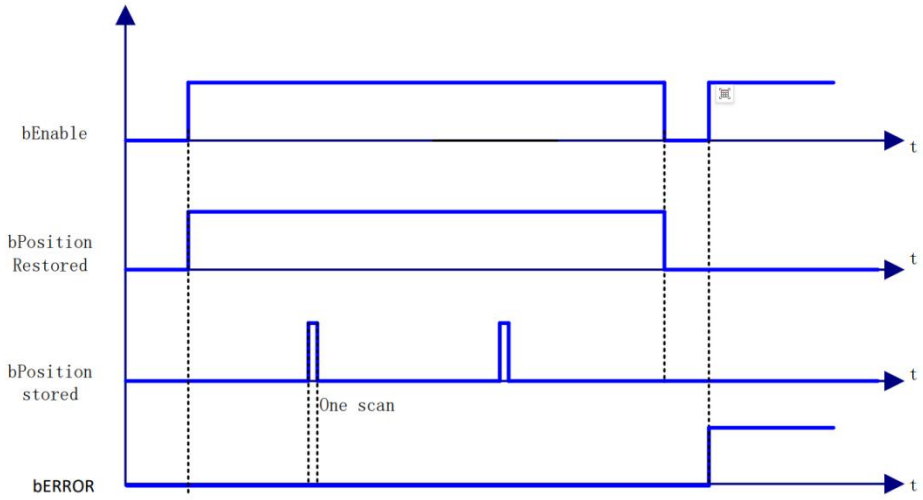
SMC3_PersistPosition_1:SMC3_PersistPosition;

END_VAR

程序部分：

```
//绝对值位置保存
SMC3_PersistPosition_1(Axis:=X_Axis , PersistentData:=persistentData1 ,bEnable:=TRUE );
```

4）时序图



5）错误说明

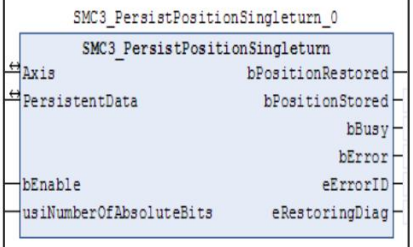
输入轴为虚拟轴或者逻辑轴会导致错误输出；轴有错误时会导致错误输出。

【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.1.31 SMC3_PersistPositionSingleturn

该指令用来保持纪录实轴绝对值编码器（单圈绝对值）的位置（断电重启控制器后，恢复断电前位置记录值）。如果伺服电机使用的是单圈绝对值编码器，使用该功能块配合使用。

1）指令格式

指令	名称	图形表现	ST 表现
SMC3_ PersistPositionSingleturn	轴位置保持		SMC3_PersistPositionSingleturn_0(Axis:= , PersistentData:= , bEnable:= , usiNumberOfAbsoluteBits:= , bPositionRestored=> , bPositionStored=> , bBusy=> , bError=> , eErrorID=> , eRestoringDiag=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例
PersistentData	保持数据	SMC3_ PersistPositionSingletrun_ Data	-	-	存储位置信息的断电保持型数据结构

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
bEnable	执行	BOOL	TRUE,FALSE	FALSE	True 功能块执行，false 不执行功能块 若要在初始化期间还原上次存储的位置，则必须从应用程序启动时将该值置为 true
usiNumberofAbsoluteBitesusiNumberofAbsoluteBites	位数	UINT	-	16	多少位的绝对值编码器（如 20 位，24 位编码器等等）

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
bPositionRestored	位置恢复	BOOL	TRUE,FALSE	FALSE	TRUE, 轴重启后位置恢复完成
bPositionStored	位置保存	BOOL	TRUE,FALSE	FALSE	TRUE, 调用功能快后保存位置完成
bBusy	FB 执行中	BOOL	TRUE,FALSE	FALSE	TRUE，功能块没有执行完成
bError	错误	BOOL	TRUE,FALSE	FALSE	TRUE, 异常发生
eErrorID	错误代码	SMC_ERROR	-	SMC_NO_ERROR	异常发生时，输出错误代码

输出变量	名称	数据类型	有效范围	初始值	描述
eRestoringDiag	恢复诊断	SMC3_ PersistPosi-tionDiag	-	SMC3_ PersistPositionDiag. SMC3_PPD_ RESTORING_OK	位置恢复中的诊断信息 SMC3_PPD_RESTORING_OK: 位置成功恢复 SMC3_PPD_AXIS_PROP_ CHANGED: 轴参数有更改，无法恢复位置 SMC3_PPD_DATA_STORED_ DURING_WRITING: 功能块从轴参数数据结构复制数据，而不是从 PersistentData 数据中复制。可能原因：非同步性持续变量、控制器崩溃死机

3）功能说明

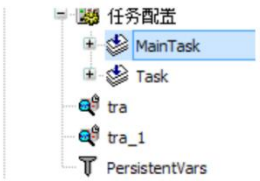
◆ PLC 重启 bEnable 信号为 TRUE，则 bPositionRestroed 输出为 TRUE。

◆ 不支持虚轴跟逻辑轴。

◆ PLC 重启后要恢复断电前“实际位置”需使用该功能块，并且为了记录断电前轴“实际位置”需将 SMC3_ PersistPositionSingleTurn_Data 配成持续型变量”。

使用方法（实轴编码器为多圈绝对值时）：

① PersistentVars 中声明 SMC3_PersistPositionSingleTurn_Data 型数据




```
VAR_GLOBAL PERSISTENT RETAIN
    persistentData2: SMC3_PersistPositionSingleTurn_Data;
END_VAR
```

② PLC 主任务（EthCat 任务）中调用

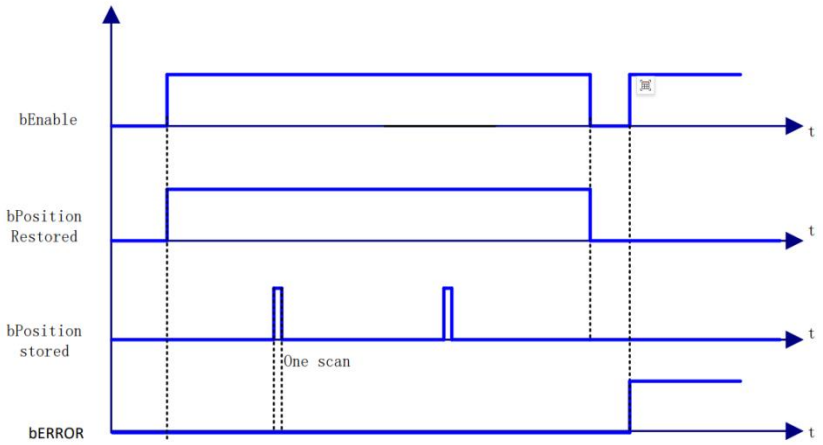
◆ 申明部分：

```
VAR
    SMC3_PersistPosition_2: SMC3_PersistPositionSingleTurn_Data;
END_VAR
```

◆ 程序部分：

```
//绝对值位置保存
SMC3_PersistPosition_2(Axis:=Y_Axis , PersistentData:=persistentData2 ,bEnable:=TRUE );
```

4）时序图



5）错误说明

输入轴为虚拟轴或者逻辑轴会导致错误输出；轴有错误时会导致错误输出。

【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.1.32 SMC_CheckAxisCommunication

该指令功能为：检查当前驱动器通讯状态。

1）指令格式

指令	名称	图形表现	ST 表现
SMC_ CheckLimits	轴限制检查	SMC_CheckAxisCommunication Axis, bEnable → bValid, bError, eErrorID, bOperational, eComState, wComState	SMC_CheckAxisCommunication0(Axis:= , bEnable:= , bValid=> , bError=> , eErrorID=> , bOperational=> , eComState=> , wComState=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
bEnable	执行	BOOL	TRUE,FALSE	FALSE	TRUE: 执行检查中

◆ 输出变量

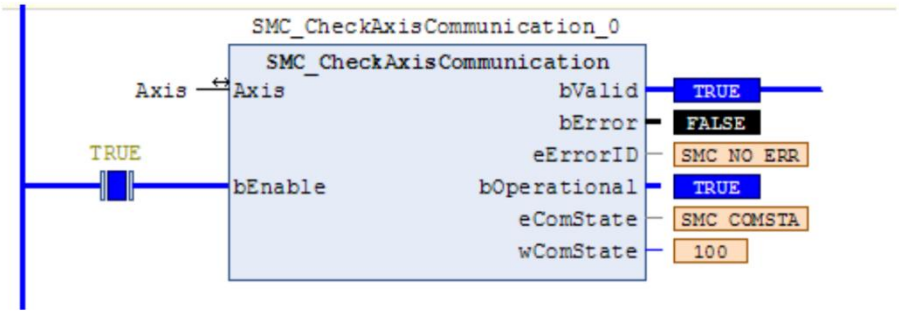
输出变量	名称	数据类型	有效范围	初始值	描述
bValid	执行中	BOOL	TRUE,FALSE	FALSE	True, 指令执行有效
bError	错误	BOOL	TRUE,FALSE	FALSE	True, 异常产生
eErrorID	错误代码	SMC_ERROR			参考 SMC_Error
bOperational	通讯正常	BOOL	TRUE,FALSE	FALSE	True, 通讯正常（代码为 100）可操作 False, 通讯不正常，不可对轴操作
eComState	通讯状态	SMC_ COMMUNI-CATIONSTATE	-	-	包含： SMC_COMSTATE_NOT_STARTED, 通讯没有启动 SMC_COMSTATE_VARIABLE_INITIALIZATION, 通讯变量初始化 SMC_COMSTATE_BASE_COM_INITIALIZATION, 基本端口初始化 SMC_COMSTATE_DRIVE_INITIALIZATION, 通讯驱动初始化 SMC_COMSTATE_DRIVE_WAITING_FOR_SYNC, 同步警告 SMC_COMSTATE_INITIALIZATION_DONE, 初始化完成 SMC_COMSTATE_OPERATIONAL, 通讯可正常使用 SMC_COMSTATE_REINITIALIZATION, 通讯重新初始化 SMC_COMSTATE_ERROR, 通讯错误 SMC_COMSTATE_UNKNOWN 通讯状态不知
wComState	通讯代码	WORD	-	-	与输入输出轴结构体变量中：Axis .wCommunicationState 值相同 表示当前通讯状态的代码，参考 AXIS_REF_SM3 参数 1013

3）功能说明

bEnable 为 TRUE, 无错误，bValid 输出 TRUE。执行轴通讯状态检查。

bValid 输出 TRUE 时检查轴通讯状态，当 eComState 输出为 SMC_COMSTATE_OPERATIONAL 时，bOperational 输出为 TRUE。

4） 样例程序



5） 错误说明

bExecute 上升沿时：

轴报错，Error 输出；

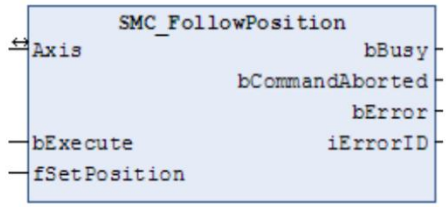
无效的轴输入，Error 输出。

【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.1.33 SMC_FollowPosition

指令功能为不做任何检查直接给轴设定位置。该指令与 MC_MoveAbsolute 有所不同,执行上升沿型号来后，每个任务周期都会给轴位置指令而不管轴的状态。（用户可用该指令写凸轮功能，而不使用 MC_CamIn 等指 令）。

1）指令格式

指令	名称	图形表现	ST 表现
SMC_ FollowPosition	轴位置给定		SMC_FollowPosition_0(Axis:= , bExecute:= , fSetPosition:=SET_POSITION , bBusy=> , bCommandAborted=> , bError=> , iErrorID=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
bExecute	执行	BOOL	TRUE,FALSE	FALSE	上升沿执行功能块
fSetPosition	设定位置	LREAL	-	0	轴设定的位置

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
bBusy	执行中	BOOL	TRUE,FALSE	FALSE	True- 指令执行中， （此时轴处于同步状态，与凸轮 MC_CamIn 指令运行时轴状态一样），可以用 MC_ Camout 指令清除 bBusy 状态
bCommandAborted	指令被中断	BOOL	TRUE,FALSE	FALSE	True- 轴被其他控制命令打断
bError	错误	BOOL	TRUE,FALSE	FALSE	True, 异常产生
iErrorID	错误代码	SMC_ERROR	-	-	参考 SMC_Error

3）功能说明

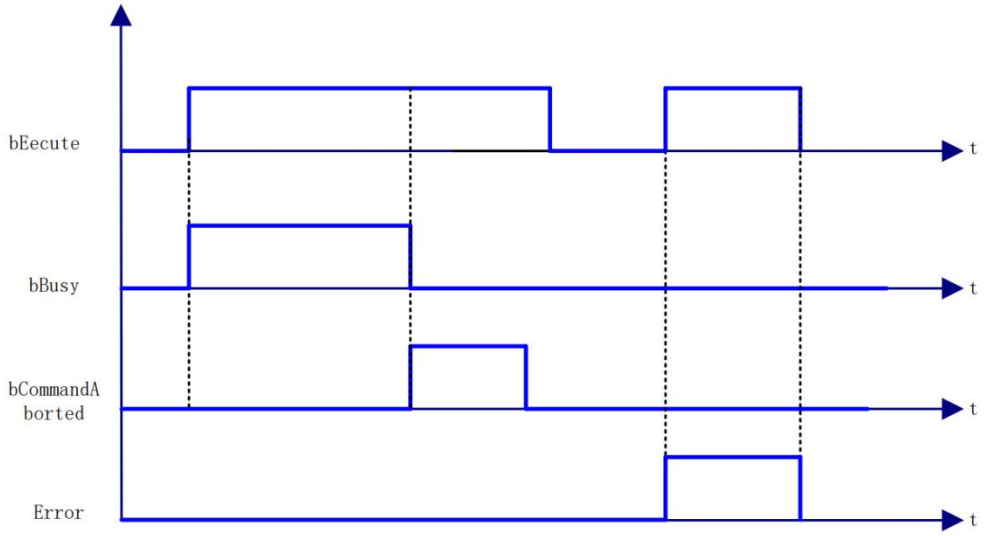
◆ SMC_FollowPosition 通过 bExecute 的上升沿启动之后，轴会每个任务周期给轴发送位置指令。

◆ bBusy 信号来时会轴的状态为同步运行与 MC_CamIn 指令生效时从轴状态一样，可以用 MC_CamOut 指令清除。

◆ 轴的的速度-由轴两个任务周期相差的位置增量自己计算出来，速度： ΔL / Δt ， ΔL 本任务周期 fSetVelocity 跟上个任务周期 fSetVelocity 差值、Δt 为扫描时间。

◆ bExecute 信号为 TRUE 时，当有其他控制命令中断该指令则 bBusy 由 TRUE 变为 FALSE。

4）时序图



5）错误说明

bExecute 上升沿时：

Axis 变量连接的为非 AXIS_REF_SM3 类型结构变量，Error 输出；

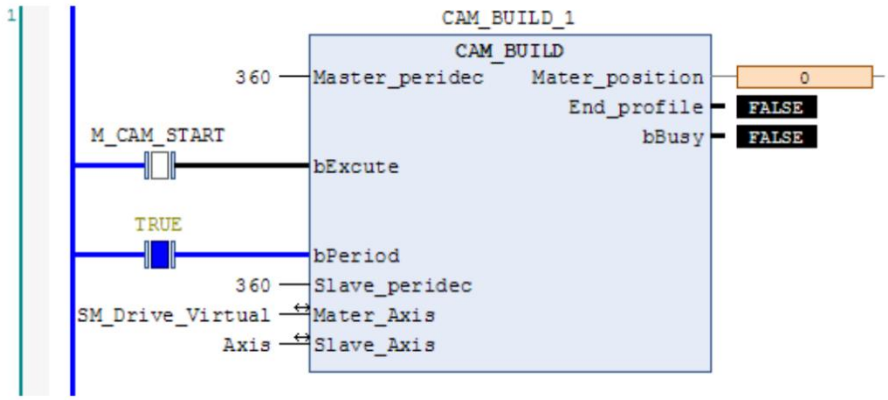
轴没使能，Error 输出。

指令运行中，轴出错，Error 输出

【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

6）样例说明

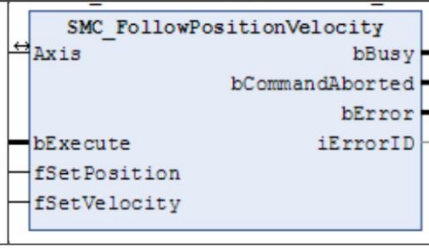
使用 SMC_FollowPosition 实现电子凸轮功能。



4.1.34 SMC_FollowPositionVelocity

该指令功能跟 SMC_FollowPosition 使用跟功能都一样，但是增加了速度设定。注意：速度设定要满足位置设定变化即：速度设定 = 相隔任务周期位置设定差值对时间的一次导数。比如： 两个相隔周期位置设定一致，则速度应该设为 0，否则会造成电机剧烈振动。

1）指令格式

指令	名称	图形表现	ST 表现
SMC_ FollowPositionVelocity	轴位置、速度给定		SMC_FollowPositionVelocity_0(Axis:= , bExecute:= , fSetPosition:= , fSetVelocity:= , bBusy=> bBusy, bCommandAborted=> , bError=> , iErrorID=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
bExecute	执行	BOOL	TRUE,FALSE	FALSE	上升沿执行功能块
fSetPosition	设定位置	LREAL	-	0	轴设定的位置
fSetVelocity	设定速度	LREAL	-	0	轴设定的位置

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
bBusy	执行中	BOOL	TRUE,FALSE	FALSE	True- 指令执行中， （此时轴处于同步状态，与凸轮 MC_CamIn 指令运行时轴状态一样），可以用 MC_ Camout 指令清除 bBusy 状态
bCommandAborted	指令被中断	BOOL	TRUE,FALSE	FALSE	True- 轴被其他控制命令打断
bError	错误	BOOL	TRUE,FALSE	FALSE	True, 异常产生
iErrorID	错误代码	SMC_ERROR	-	-	参考 SMC_Error

3）功能说明

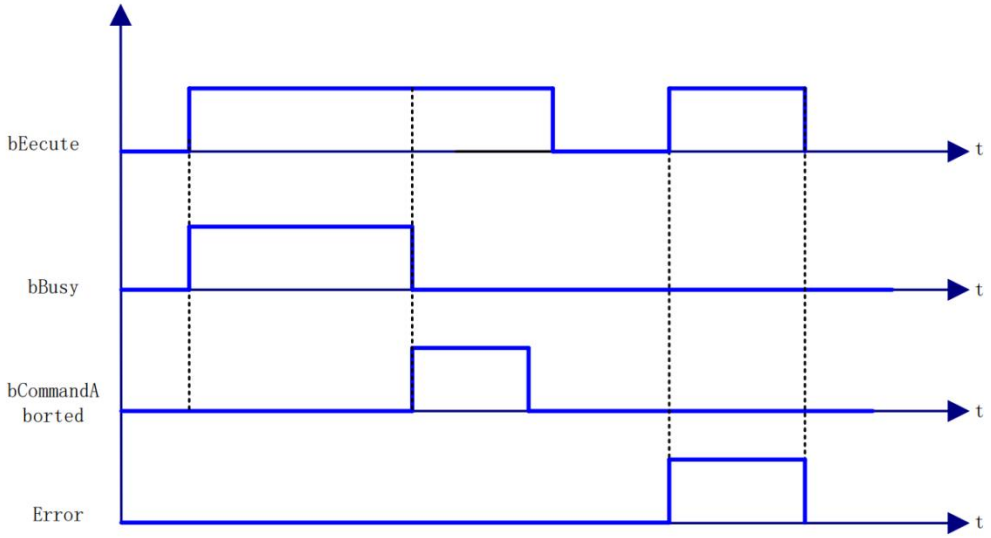
◆ SMC_FollowPositionVelocity 通过 bExecute 的上升沿启动之后，轴会每个任务周期给轴发送设定位置、设定速度指令。

◆ bBusy 信号来时会轴的状态为同步运行与 MC_CamIn 指令生效时从轴状态一样，可以用 MC_CamOut 指令清除。

◆ 轴的设定速度要与设定位置变化一致：fSetVelocity= ΔL / Δt ，ΔL 本任务周期 fSetVelocity 跟上个任务周期 fSetVelocity 差值、 Δt 为扫描时间。

◆ bExecute 信号为 TRUE 时，当有其他控制命令中断该指令则 bBusy 由 TRUE 变为 FALSE。

4）时序图



5）错误说明

bExecute 上升沿时：

Axis 变量连接的为非 AXIS_REF_SM3 类型结构变量，Error 输出；

轴没使能，Error 输出；

指令运行中，轴出错，Error 输出。

【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.1.35 SMC_FollowVelocity

指令功能为不做任何检查直接给轴设定速度。该指令与 MC_MoveVelocity 有所不同 ，执行上升沿型号来后，每个任务周期都会给轴速度指令。（MC_MoveVelocity 指令速度更改后必须刷新执行才能生效）

1）指令格式

指令	名称	图形表现	ST 表现
SMC_ FollowVelocity	轴速度给定	SMC_FollowVelocity Axis bExecute fSetVelocity bBusy bCommandAborted bError iErrorID	SMC_FollowVelocity_0(Axis:= , bExecute:= , fSetVelocity:= , bBusy=> , bCommandAborted=> , bError=> , iErrorID=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
bExecute	执行	BOOL	TRUE,FALSE	FALSE	上升沿执行功能块
fSetVelocity	设定速度	LREAL	-	0	轴设定的速度

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
bBusy	执行中	BOOL	TRUE,FALSE	FALSE	True- 指令执行中，（此时轴处于同步状态，与凸轮 MC_CamIn 指令运行时轴状态一样），可以用 MC_Camout 指令清除 bBusy 状态
bCommandAborted	指令被中断	BOOL	TRUE,FALSE	FALSE	True- 轴被其他控制命令打断（bExecute 为 True 时）
bError	错误	BOOL	TRUE,FALSE	FALSE	True, 异常产生
iErrorID	错误代码	SMC_ERROR	-	-	参考 SMC_Error

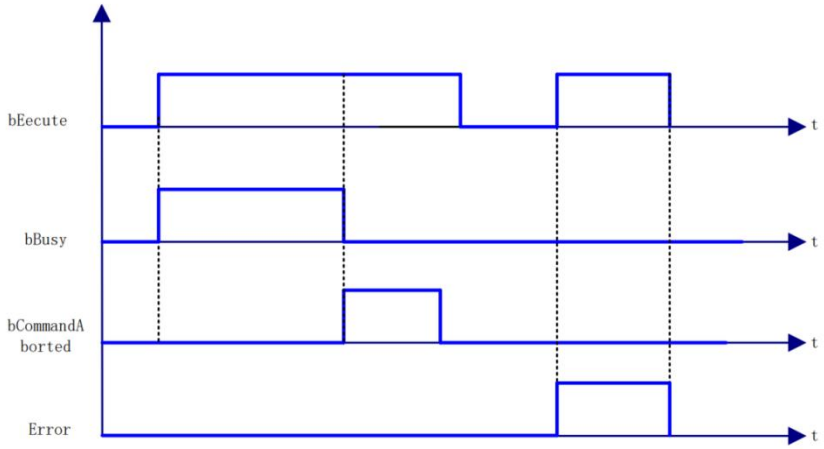
3）功能说明

◆ SMC_FollowVelocity 通过 bExecute 的上升沿启动之后，轴会每个任务周期给轴发送速度指令。

◆ bBusy 信号来时会轴的状态为同步运行 MC_CamIn 指令生效时从轴状态一样，可以用 MC_CamOut 指令清除。

◆ bExecute 信号为 TRUE 时，当有其他控制命令中断该指令则 bBusy 由 TRUE 变为 FALSE。

4）时序图



5）错误说明

bExecute 上升沿时：

Axis 变量连接的为非 AXIS_REF_SM3 类型结构变量，Error 输出；

轴没使能，Error 输出；

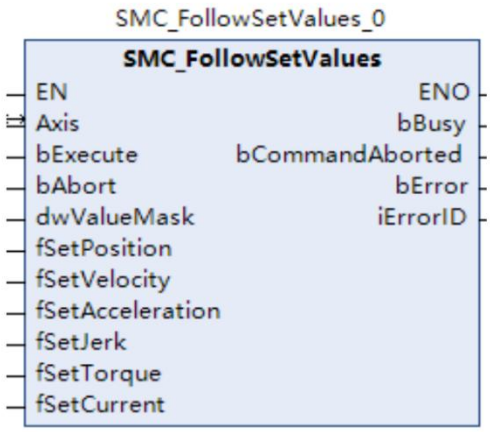
指令运行中，轴出错，Error 输出。

【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.1.36 SMC_FollowSetValues

跟其他的 SMC_Follow 功能一样都是直接给轴指令。但是该指令不仅包含其他 SMC_Follow 指令功能，还包括加速度、电流、转矩等控制信号，可认为是个综合版。通过 DwValueMask 值来选择所需指令。

1）指令格式

指令	名称	图形表现	ST 表现
SMC_ FollowSetValues	轴相关指令给定		SMC_FollowSetValues_0(Axis:= Axis, bExecute:= , bAbort:= , dwValueMask:= , fSetPosition:= , fSetVelocity:= , fSetAcceleration:= , fSetJerk:= , fSetTorque:= , fSetCurrent:= , bBusy=> , bCommandAborted=> , bError=> , iErrorID=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
bExecute	启动	BOOL	TRUE,FALSE	FALSE	上升沿执行功能块
bAbort	中止条件	BOOL	TRUE,FALSE	FALSE	中止正在进行的运动并重置所有输出
dwValueMask	控制管理	DWORD	-	0	Bite0:TRUE:fSetPosition 激活 FALSE: 忽略 Bite1:TRUE: fSetVelocity 激活 FALSE: 忽略 Bite2:TRUE: fSetAcceleration 激活 FALSE: 忽略 Bite3:TRUE: fSetJerk 激活 FALSE: 忽略 Bite4:TRUE: fSetTorque 激活 FALSE: 忽略 Bite5:TRUE: fSetCurrent 激活 FALSE: 忽略
fSetPosition	设定位置	LREAL	-	0	轴设定的位置（标定好的单位）
fSetVelocity	设定速度	LREAL	-	0	轴设定的速度（标定好的单位 /s）
fSetAcceleration	设定加速度	LREAL	-	0	轴设定的加速度（标定好的单位 /s2）
fSetJerk	设定跃度	LREAL	-	0	轴设定的跃度值（标定好的单位 /s3）
fSetTorque	设定转矩	LREAL	-	0	轴设定的跃度值（NM/N）
fSetCurrent	设定电流	LREAL	-	0	轴设定的电流值（A）

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
bBusy	执行中	BOOL	TRUE,FALSE	FALSE	接收指令后变为 TRUE
bCommandAborted	指令中断	BOOL	TRUE,FALSE	FALSE	当前指令被中断，置为 TRUE
bError	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE

输出变量	名称	数据类型	有效范围	初始值	描述
iErrorID	错误代码	SMC_ERROR	-	0	异常发生时，输出错误代码

3）功能说明

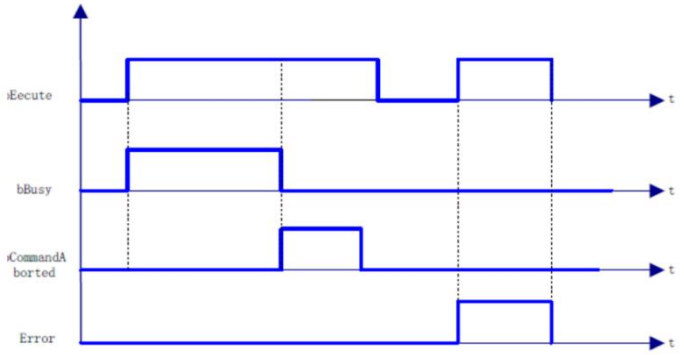
◆ SMC_FollowSetValues 通过 bExecute 的上升沿启动之后，轴会每个任务周期给轴发送选好的参数指令。

◆ bBusy 信号来时会轴的状态为同步运行与 MC_CamIn 指令生效时从轴状态一样，可以用 MC_CamOut 指 令清除。

◆ bExecute 信号为 TRUE 时，当有其他控制命令中断该指令则 bBusy 由 TRUE 变为 FALSE。

◆ 通过 DwValueMask 值来选择控制参数，比如 DwValueMask 为 1，则为每个任务周期发送位置与 SMC_ FollowPosition 指令功能一样。DwValueMask 为 2 则为单独速度指令输出。DwValueMask 为 3，则为位置速度指令输出。DwValueMask 为 7，则为位置、速度、加速度指令输出，等等。

4）时序图



4.1.37 SMC_SetControllerMode

设定伺服当前运行模式，默认为同步周期位置控制，控制模式相关设置请参考《IS620N 系列伺服设计维护手册》控制模式。

1）指令格式

指令	名称	图形表现	ST 表现
SMC_ SetControllerMode	设定轴控制模式	SMC_SetControllerMode Axis → bDone bBusy bError bExecute → nErrorID nControllerMode	SMC_SetControllerMode0(Axis:= , bExecute:= , nControllerMode:= , bDone=> , bBusy=> , bError=> , nErrorID=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

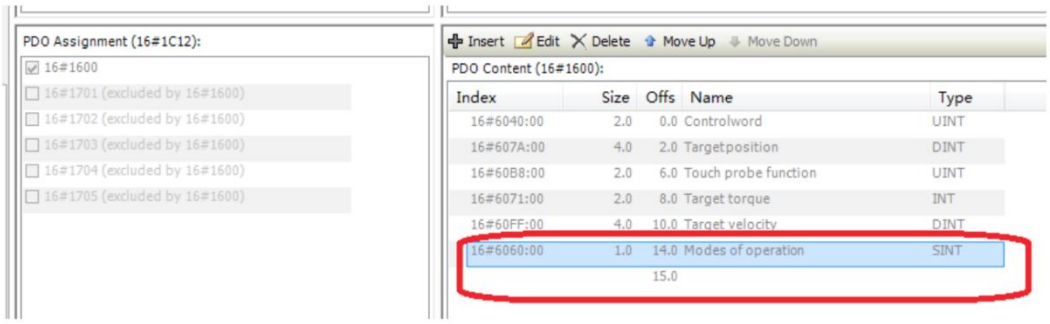
输入变量	名称	数据类型	有效范围	初始值	描述
bExecute	执行	BOOL	TRUE,FALSE	FALSE	上升沿执行功能块
nControllerMode	控制模式	SMC_CONTROLLER_MODE	-	SMC_Position	轴控制模式 1：转矩控制模式，SMC_torque 2：速度控制模式，SMC_Velocity 3：位置控制模式，SMC_Position 4: 电流控制模式，SMC_Current

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
bDone	模式设定完成	BOOL	TRUE,FALSE	FALSE	True, 模式设定完成
bBusy	执行中	BOOL	TRUE,FALSE	FALSE	True- 指令执行中，
bError	错误	BOOL	TRUE,FALSE	FALSE	True, 异常产生
iErrorID	错误代码	SMC_ERROR	-	-	参考 SMC_Error

3）功能说明

◆ SMC_SetControllerMode，通过 bExecute 的上升沿启动之后，给伺服驱动器控制模式指令，也可通过轴配置后 Axis.out.byModesofOpreation 值来设定控制模式（需在过程数据中添加对象字典 6060h）。



◆ 功能块使用需满足条件：

1：轴必须满足这些控制条件，比如虚轴不能使用该功能块。

2：各模式支持的同步周期必须保证一致

3：执行指令时轴必须在非“errorstop”，“stopping”，“homing”状态，否则会产生错误。

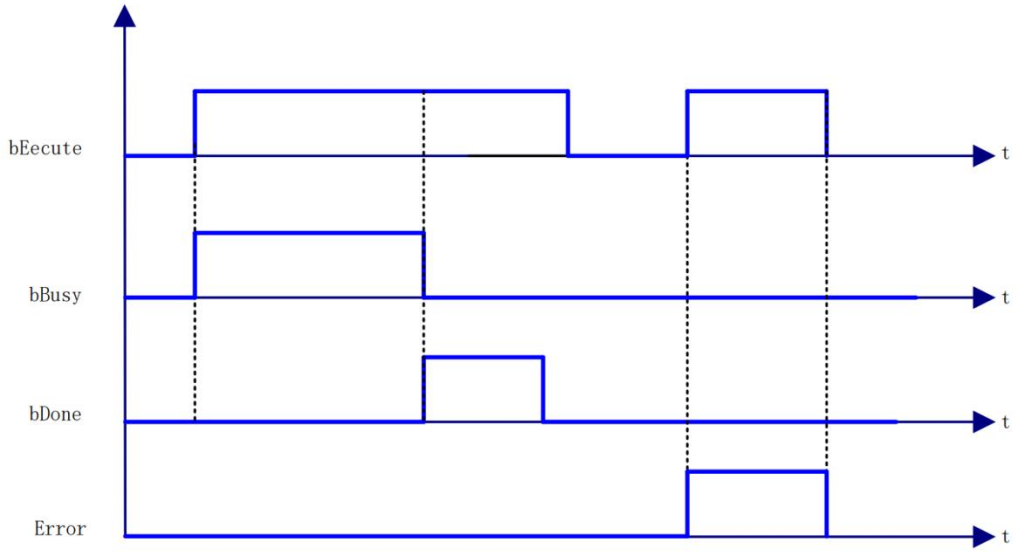
◆ 如果指令执行 1000 个任务周期后，轴仍然没有变为设定控制方式，则指令报错，bError 由 false 变为 true。

◆ 当轴的控制方式由低级向高级转变时 (torque -> velocity, torque->position, velocity->position)，功能块会 计算高级方式的设定值，例如当有转矩模式变为位置模式时，

功能块会根据当前轴实际位置叠加一个预期位 置距离（通过当前实际的速度和任务周期内的时间偏移来计算），来补偿实际和设定值之间的时间滞后。

◆ 指令执行后，当轴的实际控制方式变为设定控制方式 ,bDone 信号触发，在指令触发与 bDone 信号触发之 间的时间内轴仍然会运行，并且在这段时间内功能块会按照设定控制方式计算合适的设定值，但是如果一旦 bDone 信号触发而没有其他的控制指令继续给轴设定值，则轴会立即停止并报错，因此需使用 bDone 信号上升沿来触发 MC_Halt, MC_MoveVelocity or MC_MoveAbsolute 等指令来平滑控制轴。特别注意，当控制模式转换为力矩控制时，后需要接力矩控制指令（如 MC_TorqueControl、SMC_ SetTorque 等来平滑控制轴。

4）时序图



5）错误说明

bExecute 上升沿时：

轴无效

轴状态无效。

轴不满足控制方式。

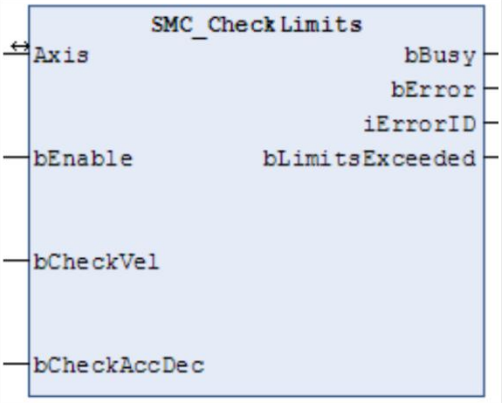
轴报错，Error 输出。

【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.1.38 SMC_CheckLimits

该指令功能为：检查当前驱动器设置值是否超出控制器配置的最大值。

1）指令格式

指令	名称	图形表现	ST 表现
SMC_ CheckLimits	轴限制检查		SMC_CheckLimits0(Axis:= , bEnable:= , bCheckVel:= , bCheckAccDec:= , bBusy=> , bError=> , iErrorID=> , bLimitsExceeded=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
bEnable	执行	BOOL	TRUE,FALSE	FALSE	TRUE: 执行检查中
bCheckVel	速度检查	BOOL	TRUE,FALSE	FALSE	TRUE: 执行速度检查，false: 不执行速度检查
bCheckAccDec	加减速检查	BOOL	TRUE,FALSE	FALSE	TRUE: 执行加减速检查，false: 不执行加减速检查

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
bBusy	执行中	BOOL	TRUE,FALSE	FALSE	True- 执行轴检查， False: 不执行轴检查
bError	错误	BOOL	TRUE,FALSE	FALSE	True, 异常产生
iErrorID	错误代码	SMC_ERROR	-	-	参考 SMC_Error

输出变量	名称	数据类型	有效范围	初始值	描述
bLimitsExceeded	检查限制输出	BOOL	TRUE,FALSE	FALSE	TRUE: 当前设定速度，或加减速超过 Axis.fSWMaxVelocity, Axis.fSWMaxAcceleration Axis.fSWMaxDeceleration

3）功能说明

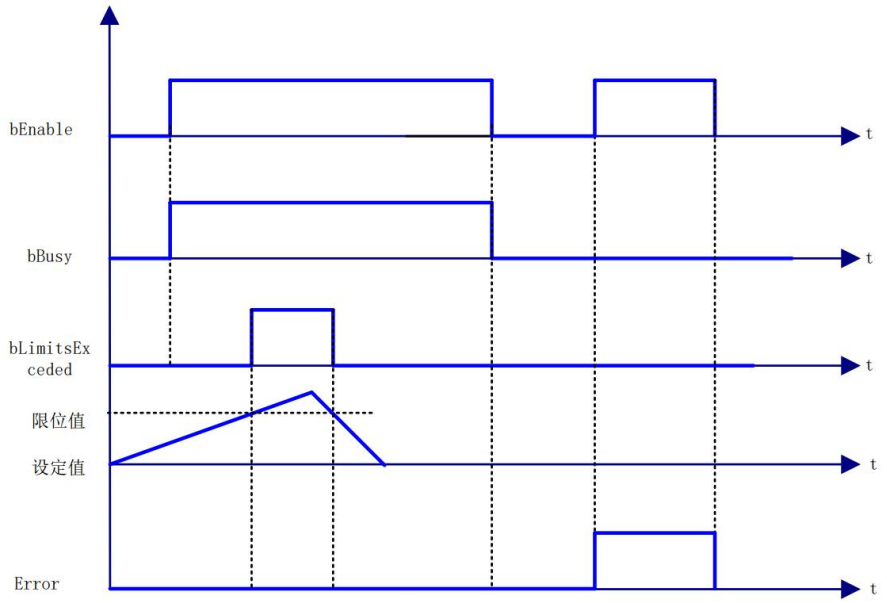
◆ bEnable 为 TRUE,bBusy 输出 TRUE。执行轴速度、加速度检查。

◆ 当前轴的设定速度或者加减速度超过 Axis.fSWMaxVelocity、Axis.fSWMaxAcceleration、

Axis.fSWMaxDeceleration 设定值，bLimitsExceeded 信号输出为 TRUE。

注意：该功能只是检查当前的指令速度或加减速超过设置的限值，并不能停止轴。

4）时序图



5）错误说明

bExecute 上升沿时：

轴报错，Error 输出。

无效的轴输入，Error 输出。

【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.1.39 SMC_GetMaxSetAccDec

该指令功能为：读取轴最大加减速度。

1）指令格式

指令	名称	图形表现	ST 表现
SMC_GetMaxSetAccDec	轴度最大加减速	SMC_GetMaxSetAccDec Axis, bEnable, dwTimeStamp → bValid, bBusy, fMaxAcceleration, dwTimeAtMax	SMC_GetMaxSetAccDec_0(Axis:= , bEnable:= , dwTimeStamp:= , bValid=> , bBusy=> , fMaxAcceleration=> , dwTimeAtMax=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
bEnable	执行	BOOL	TRUE,FALSE	FALSE	TRUE: 执行读取
dwTimeStamp		Dword	-	-	可选的时间戳输入; 可以用来查找最大值时发生的情况。

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
bValid	有效	BOOL	TRUE,FALSE	FALSE	True, 指令执行有效
bBusy	执行中	BOOL	TRUE,FALSE	FALSE	True, 异常产生
fMaxAcceleration	最大加减速值	LREAL	-	0	最大的加减速值（正为加速，负为减速，加减速绝对值最大值为最终值）
dwTimeAtMax	最大值对应时间戳	Dword	-	0	最大加减速时对应的 dwTimeStamp 值（例如加速度持续增加时，该值更随 dwTimeStamp，fMaxAcceleration 值也更新，一旦加速度达到最大值，则 fMaxAcceleration 记录最大值，同时最大值对应的 dwTimeStamp 也被记录）

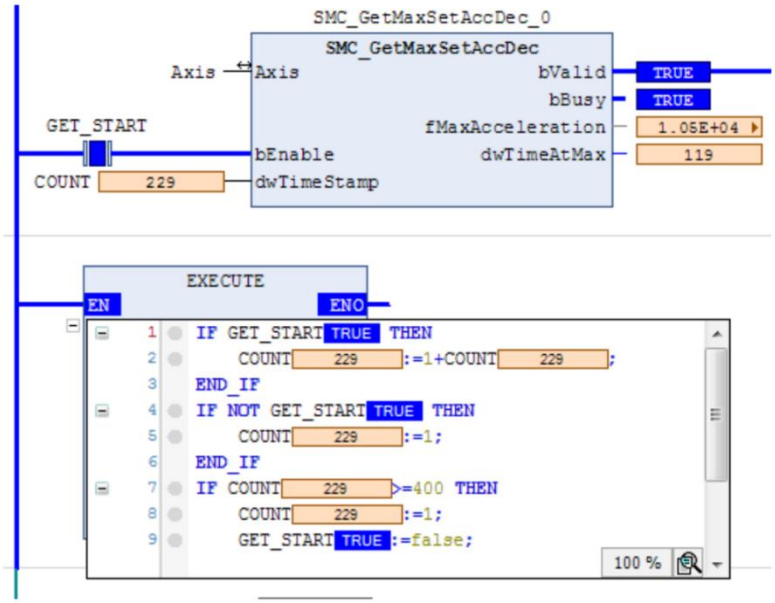
3）功能说明

◆ bEnable 为 TRUE，无错误，bValid 输出 TRUE。执行轴最大加减速检测。

◆ 当加减速绝对值大于前面记录值时，fMaxAcceleration 跟 dwTimeAtMax 会刷新。

◆ dwTimeAtMax 值为最大加减速时对应 dwTimeStamp 值，所以 dwTimeStamp 应设置为可变的，比如设置 为随着任务周期或者固定时间周期的计数值。（见样例程序）

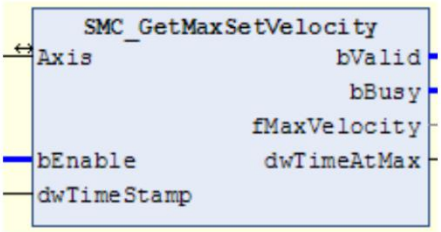
4）样列程序



4.1.40 SMC_GetMaxSetVelocity

该指令功能为：读取轴最大速度。

1）指令格式

指令	名称	图形表现	ST 表现
SMC_GetMax SetVelocity	轴度最大加减速		SMC_GetMaxSetVelocity_0(Axis:= , bEnable:= , dwTimeStamp:= , bValid=> , bBusy=> , fMaxVelocity=> , dwTimeAtMax=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
bEnable	执行	BOOL	TRUE,FALSE	FALSE	TRUE: 执行读取
dwTimeStamp	-	Dword	-	-	可选的时间戳输入；可以用来查找最大值时发生的情况。

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
bValid	有效	BOOL	TRUE,FALSE	FALSE	True, 指令执行有效
bBusy	执行中	BOOL	TRUE,FALSE	FALSE	True, 异常产生
fMaxVelocity	最大加速度值	LREAL	-	0	最大的速度值（正为正向，负为反向，绝对值最大值为最终值）
dwTimeAtMax	最大值对应时间戳	Dword	-	0	最大速度时对应的 dwTimeStamp 值（例如速度持续增加时，该值更随 dwTimeStamp，fMaxVelocity 值也更新，一旦速度达到最大值，则 fMaxVelocity 记录最大值，同时最大值对应的 dwTimeStamp 也被记录）

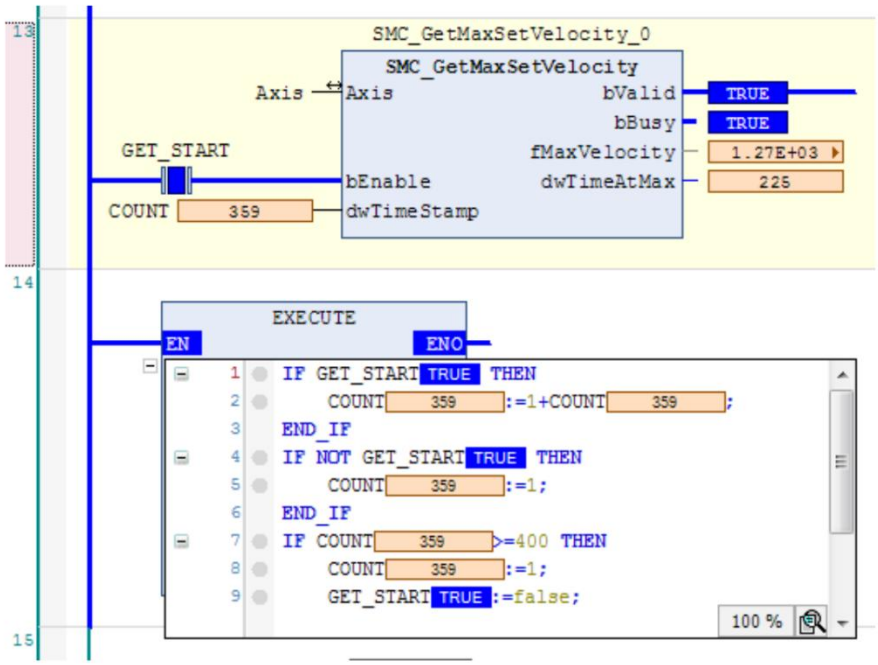
3）功能说明

◆ bEnable 为 TRUE，无错误，bValid 输出 TRUE。执行轴最大加减速检测。

◆ 当速度绝对值大于前面记录值时，fMaxVelocity 跟 dwTimeAtMax 会刷新。

◆ dwTimeAtMax 值为最大速度时对应 dwTimeStamp 值，所以 dwTimeStamp 应设置为可变的，比如设置为 随着任务周期或者固定时间周期的计数值。（见样例程序）

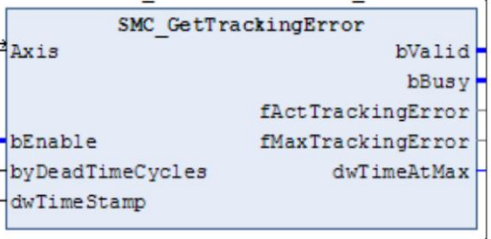
4）样列程序



4.1.41 MC_GetTrackingError

该指令功能为：测量当前或者最大滞后误差（ 指令与轴实际位置差 ）用于补偿死区时间。

1）指令格式

指令	名称	图形表现	ST 表现
SMC_GeTrackingError	轴滞后偏差读取		SMC_GetTrackingError(Axis:= , bEnable:= , byDeadTimeCycles:= , dwTimeStamp:= , bValid=> , bBusy=> , fActTrackingError=> , fMaxTrackingError=> , dwTimeAtMax=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
bEnable	执行	BOOL	TRUE,FALSE	FALSE	TRUE: 执行读取
byDeadTimeCycles	-	Byte	-	2	死区周期数，bEnable 触发延迟多少个 dwTimeStamp 值开始滞后检测。
dwTimeStamp	-	Dword	-	-	可选的时间戳输入；可以用来查找最大值时发生的情况。

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
bValid	有效	BOOL	TRUE,FALSE	FALSE	True, 指令执行有效
bBusy	执行中	BOOL	TRUE,FALSE	FALSE	True, 异常产生
fActTrackingError	当前滞后	LREAL	-	0	跟 byDeaTimeCycles 值相关的当前偏差检测
fMaxTrackingError	最大滞后	LREAL	-	0	当前的偏差值（指令位置，跟反馈位置偏差）
dwTimeAtMax	最大值对应时间戳	Dword	-	0	最大的偏差值（正为滞后，负为超前，绝对值最大值为最终值） 注意：byDeaTimeCycles 值会影响该值

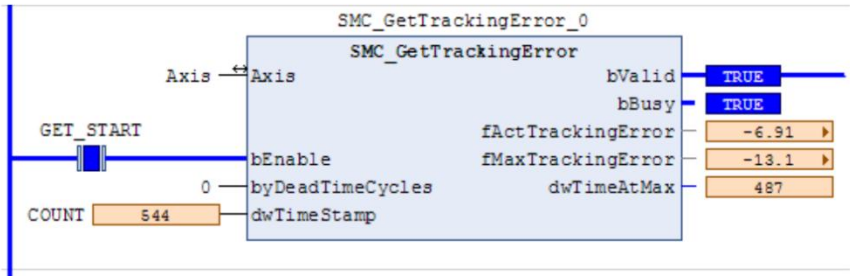
3）功能说明

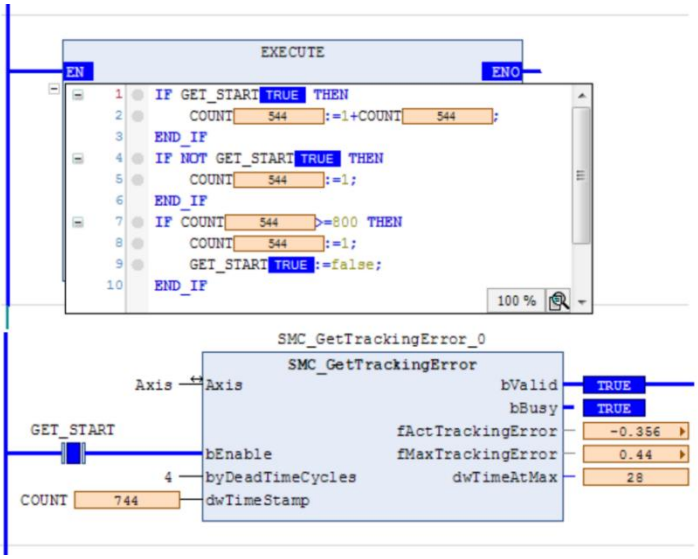
◆ bEnable 为 TRUE，bValid 输出 TRUE。执行轴滞后偏差检测。

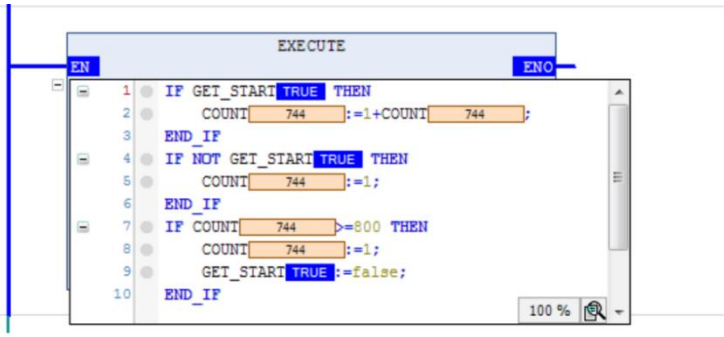
◆ 当偏差绝对值大于前面记录值时，fMaxTrackingError 跟 dwTimeAtMax 会刷新。

◆ dwTimeAtMax 值为最大偏差时对应 dwTimeStamp 值，所以 dwTimeStamp 应设置为可变的，比如设置为 随着任务周期或者固定时间周期的计数值。（见样例程序）

4）样列程序







4.1.42 SMC_InPosition

该指令功能为：监控当前轴设定位置值跟实际值之间的偏差 ， 通过设定的偏差窗口来确定轴是否在要求的偏差范围内。

1）指令格式

指令	名称	图形表现	ST 表现
SMC_InPosition	轴偏差监控	SMC_InPosition Axis bInPosition bBusy bTimeOut bEnable fPosWindow fPosTime fTimeOut	SMC_InPosition0(Axis:=Axis , bEnable:= , fPosWindow:= , fPosTime:= , fTimeOut:= , bInPosition=> , bBusy=> , bTimeOut=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
bEnable	执行	BOOL	TRUE,FALSE	FALSE	TRUE: 执行读取
fPosWindow	偏差窗口	LREAL	-	0	设定偏差监控的窗口，fPosWindow>Distance（指令位置跟反馈位置间偏差），则根据 fPosTime 时间输出 bInPosition 为 TRUE
fPosTime	触发时间	LREAL	-	0	偏差在窗口范围内时间，用来触发 bInPosition 单位为 S（秒）。
fPosTiOut	超时时间	LREAL	-	0	偏差超时 单位为 S（秒）。

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
bInPosition	偏差正常	BOOL	TRUE,FALSE	FALSE	True, 偏差在设置窗口范围内
bBusy	执行中	BOOL	TRUE,FALSE	FALSE	True, 执行中
bTimeOut	超时	LREAL	TRUE,FALSE	FALSE	跟 byDeaTimeCycles 值相关的当前偏差检测

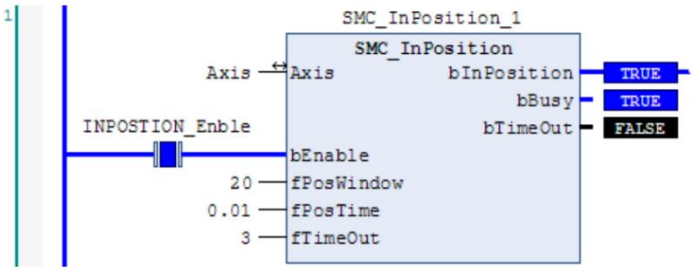
3）功能说明

◆ bEnable 为 TRUE，一旦检测到的偏差小于设置的窗口 fPosWindow 持续 fPosTime 秒则 bInPosition 触发为 TRUE。一旦检测到的偏差大于设置的窗口，bInPosition 立马输出为 FALSE。注意：fPosTime 时间设定需合 理否则会造成 bTimeOut 触发（比如一个凸轮周期为 2 秒的凸轮，连续偏差没有超过设定窗口的时间为 1.5 秒，fPosTime 如果设定的大于 1.5 秒则会造成 bInPosition 不会触发）。

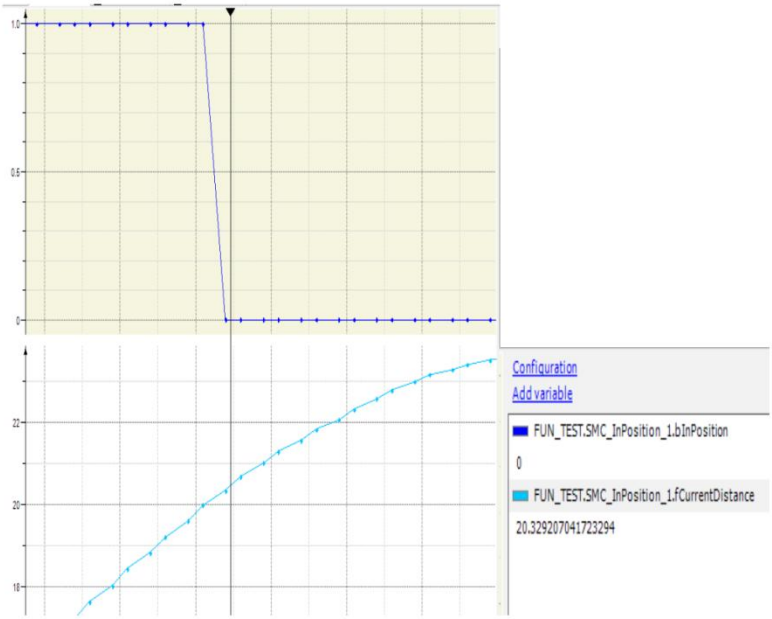
◆ bEnable 为 TRUE，bBusy 输出为 true。

◆ 偏差值可监控 SMC_InPosition 结构体中的数据 fCurrentDistance。bEnable 为 TRUE，超过 fPosTime 设定时间 bInPosition 仍然没有触发为 TRUE，则 bTimeOut 触发为 TRUE。

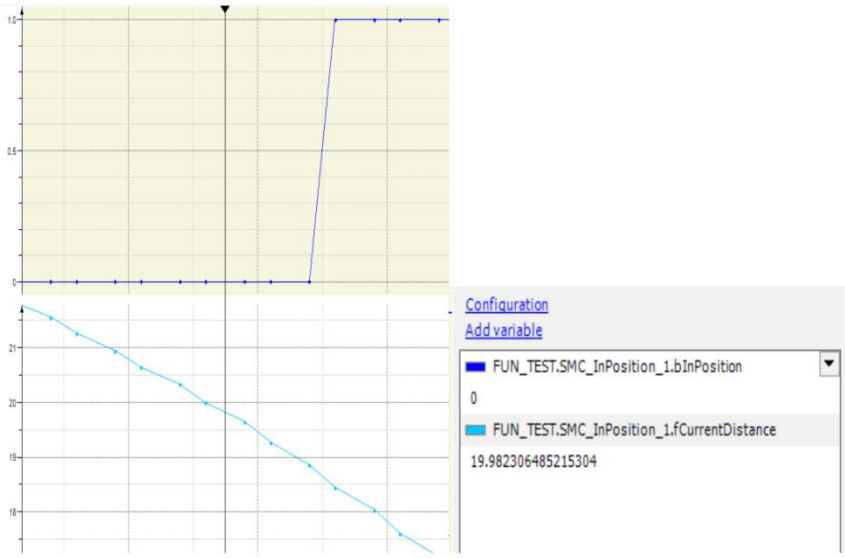
◆ 时序图样列程序



◆ 样例程序

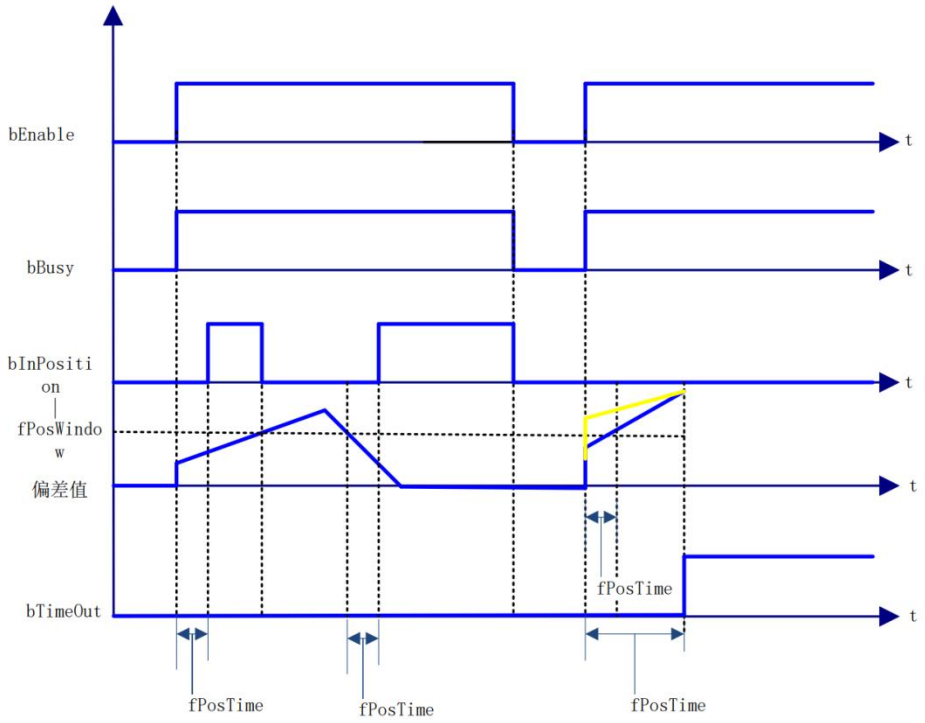


大于窗口设定 bInPosition 立即由 true 变为 FALSE



在设定窗口以内 4 个任务周期（2.5ms）后 bInPosition 变为 TRUE，跟程序设置 0.01S 相符

4）时序图



4.1.43 SMC_ReadSetPosition

该指令功能为：读取轴的指令位置（转换过后的用户单位）。

1）指令格式

指令	名称	图形表现	ST 表现
SMC_ ReadSetPosition	读轴指令位置	SMC_ReadSetPosition Axis Enable Valid Busy Error ErrorID Position	SMC_ReadSetPosition0(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , Position=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行	BOOL	TRUE,FALSE	FALSE	TRUE: 执行读取

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Valid	有效	BOOL	TRUE,FALSE	FALSE	True, 读取有效
Busy	执行中	BOOL	TRUE,FALSE	FALSE	True, 执行中
Error	错误	BOOL	TRUE,FALSE	FALSE	True, 异常产生
ErrorID	错误代码	SMC_ERROR	-	-	参考 SMC_Error
Position	指令位置	LREAL	-	0	当前任务周期的指令位置

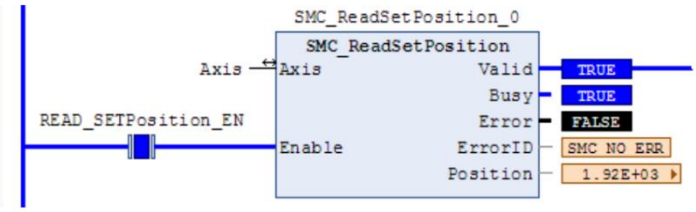
3）功能说明

Enable 为 TRUE，无错误则 Valid，Busy 输出为 TURE。

Position 输出的值为 Axis.fSetPosition 的值。

Enable 变为 FALSE，，则 Valid，Busy 输出为 FALSE。Position 停留在 FALSE 之前的值。

4）时序图样列程序



5）错误说明

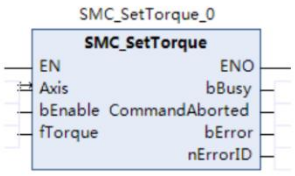
bExecute 上升沿时：轴报错，Error 输出；无效的轴输入，Error 输出。

【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.1.44 SMC_SetTorque

该指令功能为：设定轴转矩（转矩控制模式时有效）。

1）指令格式

指令	名称	图形表现	ST 表现
SMC_SetTorque	力矩设定		SMC_SetTorque_0(Axis:= Axis, bEnable:= , fTorque:= , bBusy=> , CommandAborted=> , bError=> , nErrorID=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
bEnable	执行	BOOL	TRUE,FALSE	FALSE	上升沿，设定轴力矩
fTorque	设定的力矩	LREAL	-	0	目标力矩值

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Busy	执行中	BOOL	TRUE,FALSE	FALSE	True, 执行中
CommandAborted	指令中断	BOOL	TRUE,FALSE	FALSE	当前指令被中断，置为 TRUE。
Error	错误	BOOL	TRUE,FALSE	FALSE	True, 异常产生
ErrorID	错误代码	SMC_ERROR	-	-	参考 SMC_Error

3）功能说明

◆ bEnable 上升沿，无错误则 bBusy 输出为 TURE。

◆ 该指令只是给轴设定转矩值用并不是转矩控制功能，轴控制模式在转矩控制模式下有效。

◆ 力矩设定指令只能在同步力矩模式下运行，在使能本指令时，需先利用 SMC_SetControllermode 系统将控制模式切换为同步力矩模式。

◆ 驱动器实际力矩受配置参数中设置的最大正负力矩限制。

◆ 要停止执行本指令，可使用 MC_Stop（强制停止）指令或 MC_ImmediateStop（急停）指令，停止后，驱动 器转换到同步位置模式。

4）错误说明

bExecute 上升沿时：

轴报错，Error 输出；无效的轴输入，Error 输出。

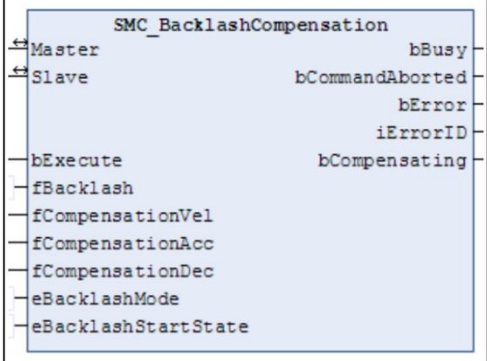
轴控制模式错误，Error 输出，错误代码 SMC_ST_WRONG_CONTROLLER_MODE

【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.1.44 SMC_BacklashCompensation

该指令功能为：用来补偿主从轴间隙，比如说皮带传送中虚轴为主轴，从轴为虚轴同步镜像，由于外部原因 导致从轴位置跟主轴存在间隙，可用该指令来补偿这种间隙。该指令功能与相位偏移指令（MC_Phasing）类似，其相位取决于主轴运行的方向。

1）指令格式

指令	名称	图形表现	ST 表现
SMC_ BacklashCom pensation	间隙补偿		SMC_BacklashCompensation0(Master:= , Slave:= , bExecute:= , fBacklash:= , fCompensationVel:= , fCompensationAcc:= , fCompensationDec:= , eBacklashMode:= , eBacklashStartState:= , bBusy=> , bCommandAborted=> , bError=> , iErrorID=> , bCompensating=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Master	主轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例
Slave	从轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
bExecute	执行	BOOL	TRUE,FALSE	FALSE	上升沿，设定偏移
fBacklash		LREAL	-	0	补偿间隙
fCompensationVel		LREAL	-	0	补偿时速度
fCompensationAcc		LREAL	-	0	补偿时加速度
fCompensationDec		LREAL	-	0	补偿时减速度
eBacklashMode		SMC_ BACKLASH_ MODE	-	SMC_BL_ AUTO	补偿模式: SMC_BL_AUTO: 主轴运行方向决定补偿方向 SMC_BL_POSITIVE: 正向补偿，独立于于主轴运行方向 SMC_BL_NEGATIVE: 反向补偿，独立于主轴运行方向 SMC_BL_OFF: 不补偿

输入变量	名称	数据类型	有效范围	初始值	描述
eBacklashStartState		SMC_ BACKLASH_ STARTSTATE	-	SMC_BL_ START_ NEGATIVE	描述该指令工作时轴的工作状态。 SMC_BL_START_NEGATIVE: 从轴在负方向牵引下运动，在负方向运动下不需要补偿，一旦正向运动以两倍 fBacklash 建立补偿 SMC_BL_START_POSITIVE: 从轴在正方向牵引下运动，在正方向运动下不需要补偿，一旦反向运动需要以两倍 fBacklash 建立补偿 SMC_BL_START_NONE: 在正的或反的方向运动会产生 fBacklash 值的距离补偿。

◆ 输出变量

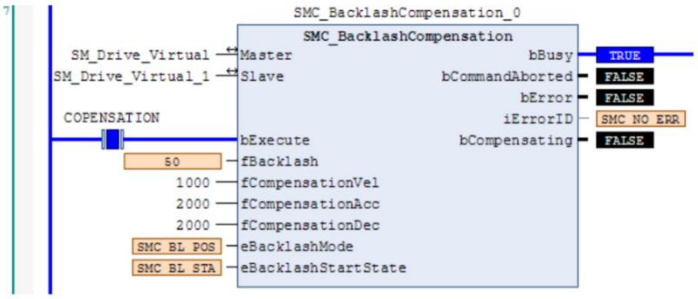
输出变量	名称	数据类型	有效范围	初始值	描述
bBusy	执行中	BOOL	TRUE,FALSE	FALSE	True, 执行中
bCommandAborted	指令被中断	BOOL	TRUE,FALSE	FALSE	True- 被其他控制命令打断
bError	错误	BOOL	TRUE,FALSE	FALSE	True, 异常产生
iErrorID	错误代码	SMC_ERROR	-	-	参考 SMC_Error
bCompsating	补偿中	BOOL	TRUE,FALSE	FALSE	

3）功能说明

◆ bEecute 上升沿，无错误则，bBusy 输出为 TURE，bCompsating 输出为 true，补偿完成后 bCompsating 输出为 false。

◆ 工作方式为：eBacklashMode- 补偿方向为“正”，eBacklashStartState 为“正”，fBacklash 为正值。在 bBusy 信号没来之前，最好主从轴位置一致否则 bEecute 上升沿来后，从轴会调整到主轴相位同步，bBusy 信号已经有的情况再刷新 bEecute 上升沿请遵守：

4）时序图样列程序



5）错误说明

bExecute 上升沿时：

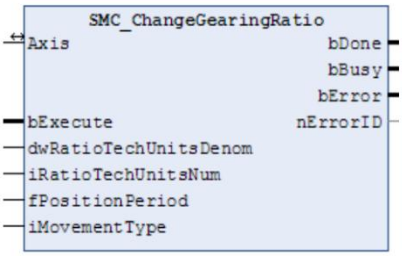
轴报错，Error 输出；无效的轴输入，Error 输出。

【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.1.45 SMC_ChangeGearingRatio

该指令功能为：用来改变用户设定电子齿轮比（脉冲转用户使用单位比）和驱动类型。注意：执行了功能块 后轴需要通过 SMC3_ReinitDrive 来重启来保证能够正确初始化设置变量

1）指令格式

指令	名称	图形表现	ST 表现
SMC_ ChangeGearingRatio	改变齿轮比		SMC_ChangeGearingRatio0(Axis:= , bExecute:= , dwRatioTechUnitsDenom:= , iRatioTechUnitsNum:= , fPositionPeriod:= , iMovementType:= , bDone=> , bBusy=> , bError=> , nErrorID=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例．齿轮比将被改的轴

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
bExecute	执行	BOOL	TRUE,FALSE	FALSE	上升沿，执行功能块
dwRatioTechUnitsDenom	-	DWORD	-	0	脉冲单位转换为应用单位（eg:mm）
iRatioTechUnitsNum	-	DINT	-	0	dwRatioTechUnitsDenom 值对应所需的应用单位
fPositionPeriod	-	LREAL	-	-	位置循环周期（模数值），只对旋转电机有效
iMovementType	-	INT	-	-	0: modulo axis（模数轴）, 1: finite axis（有限长轴）.

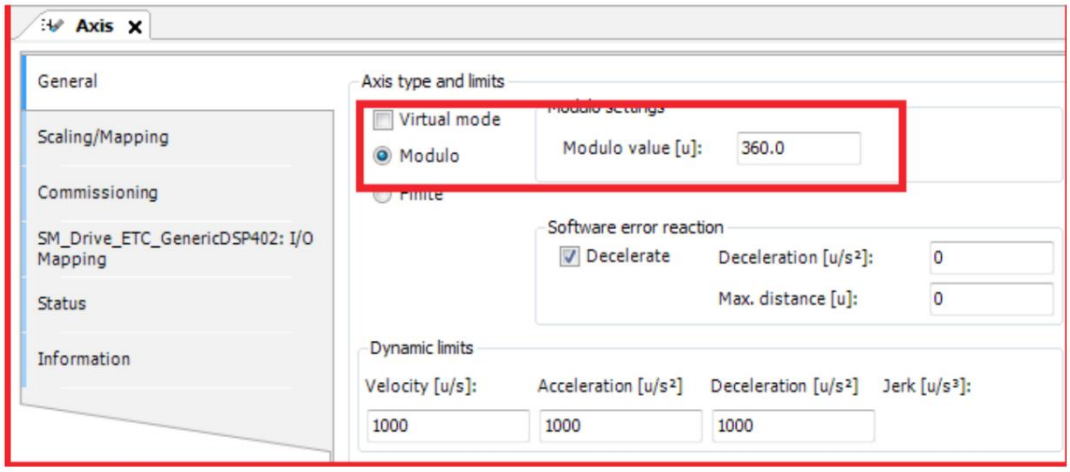
◆ 输出变量

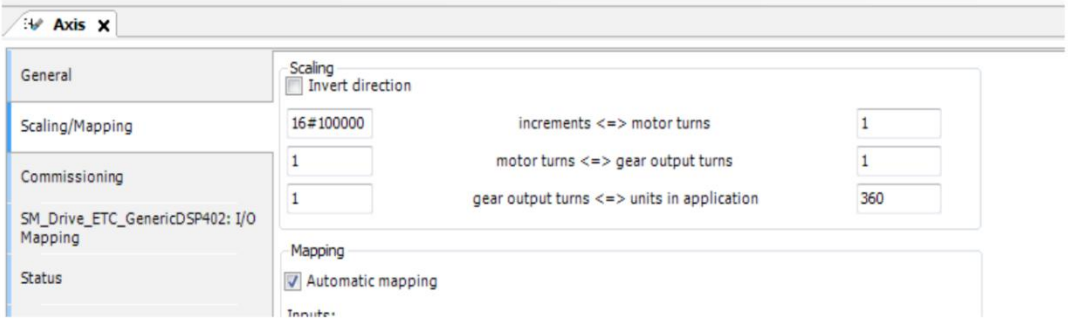
输出变量	名称	数据类型	有效范围	初始值	描述
bDone	完成	BOOL	TRUE,FALSE	FALSE	True, 执行设定完成
bBusy	执行中	BOOL	TRUE,FALSE	FALSE	True, 执行中
bError	错误	BOOL	TRUE,FALSE	FALSE	True, 异常产生
nErrorID	错误代码	SMC_ERROR	-	-	参考 SMC_Error

3）功能说明

◆ bEecute 上升沿，无错误则，bBusy 输出为 TURE，完成 bDone 输出为 true，bBusy 输出为 false。 比如 20 位编码器伺服电机加 10:1 减速比，驱动丝杠（10mm 节距），则电机转 10 圈，丝杆动距离 10mm，设置 dwRatioTechUnitsDenom 1048576*10，iRatioTechUnitsNum 为 10。

◆ 该功能块的作用为程序动态修改下图所示的部分：





4）错误说明

bExecute 上升沿时：

轴报错，Error 输出。

输入值无效，Error 输出 ， 错误代码 SMC_CGR_ZERO_VALUES

轴在指令控制运行中，Error 输出 ， 错误代码 SMC_CGR_DRIVE_POWERED

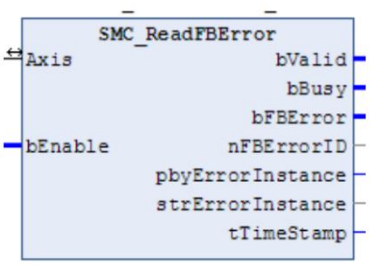
输入的模数值无效

【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.1.45 SMC_ReadFBError

该指令功能为：读功能块错误。

1）指令格式

指令	名称	图形表现	ST 表现
SMC_ ReadFBError	读功能块错误		SMC_ReadFBError(Axis:= , bEnable:= , bValid=> , bBusy=> , bFBError=> , nFBErrorID=> , pbyErrorInstance=> , strErrorInstance=> , tTimeStamp=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
bEnable	执行	BOOL	TRUE,FALSE	FALSE	TRUE: 执行读取

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
bValid	有效	BOOL	TRUE,FALSE	FALSE	True, 读取有效
bBusy	执行中	BOOL	TRUE,FALSE	FALSE	True, 执行中
bFBError	错误	BOOL	TRUE,FALSE	FALSE	True, 有 FB 错误产生
nFBErrorID	错误代码	SMC_ERROR	-	-	参考 SMC_Error
pbyErrorInstance	-	-	-	-	输出点的功能块报错
strErrorInstance	-	-	-	-	指向错误功能块（程序，子程序，功能块）
tTimeStamp	-	TIME	-	-	错误发生时的时间戳

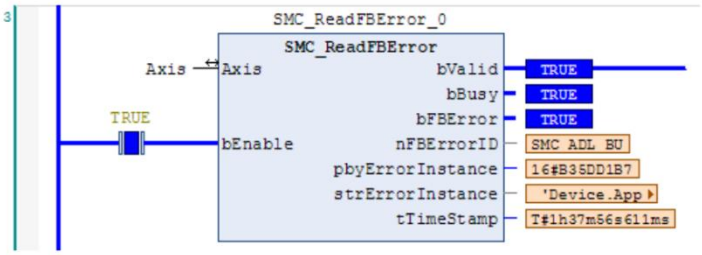
3）功能说明

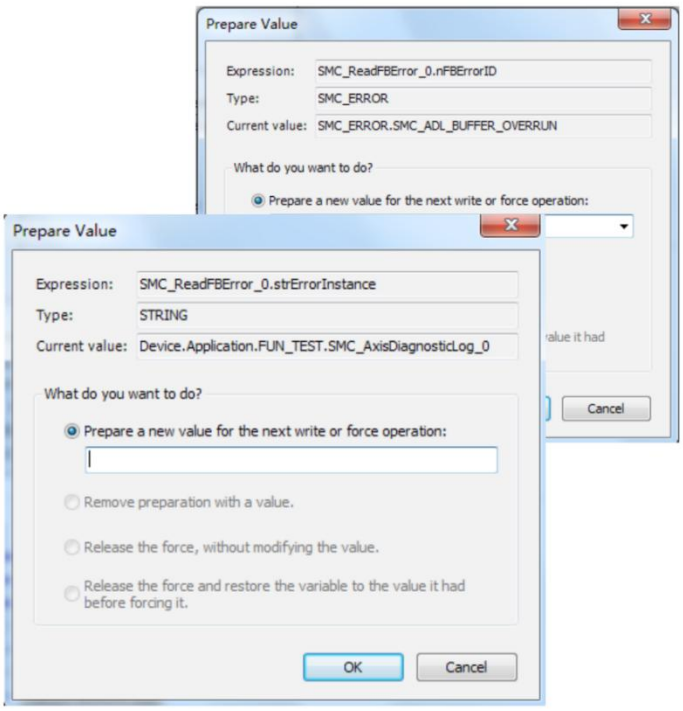
◆ Enable 为 TRUE，无错误则 Valid，Busy 输出为 TURE。

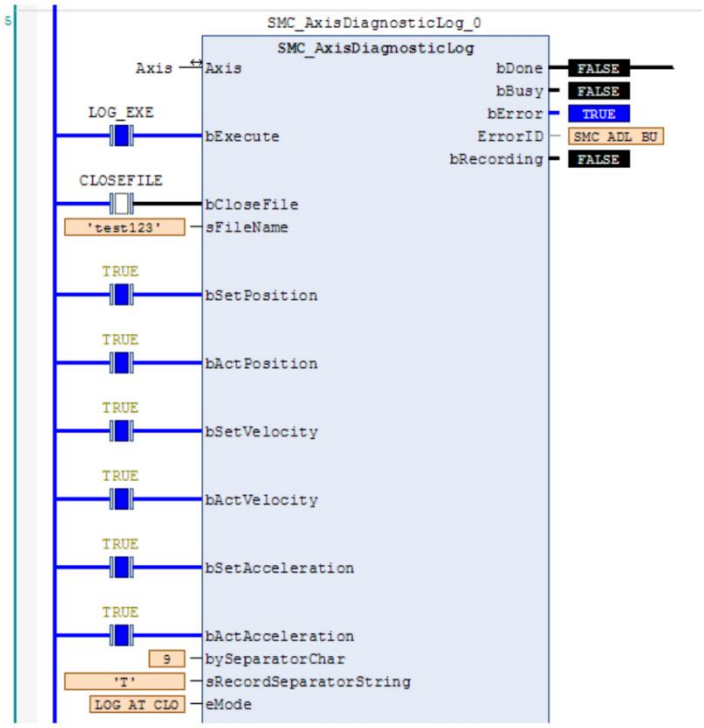
◆ 有功能块报警则，bFBError 输出为 true。

◆ Enable 变为 FALSE，，则 Valid，Busy 输出为 FALSE。

4）时序图样列程序


3
SMC_ReadFBError_0
SMC_ReadFBError
Axis
Axis
bValid
TRUE
bBusy
TRUE
TRUE
bFBError
TRUE
bEnable
nFBErrorID
SMC_ADL_BU
pbyErrorInstance
16#B35DD1B7
strErrorInstance
'Device.App
tTimeStamp
T#1h37m56s611ms


Prepare Value
Expression: SMC_ReadFBError_0.nFBErrorID
Type: SMC_ERROR
Current value: SMC_ERROR.SMC_ADL_BUFFER_OVERRUN
What do you want to do?
Prepare a new value for the next write or force operation:
Cancel
Prepare Value
Expression: SMC_ReadFBError_0.strErrorInstance
Type: STRING
Current value: Device.Application.FUN_TEST.SMC_AxisDiagnosticLog_0
What do you want to do?
Prepare a new value for the next write or force operation:
Remove preparation with a value.
Release the force, without modifying the value.
Release the force and restore the variable to the value it had before forcing it.
OK
Cancel



发生错误的功能块

5）错误说明

bExecute 上升沿时：

轴报错，Error 输出；

无效的轴输入，Error 输出。

【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.1.46 SMC_ClearFBError

该指令功能为：清除功能块的 FB 错误。

1）指令格式

指令	名称	图形表现	ST 表现
SMC_ ClearFBError	清除功能块的 FB 错误	SMC_ClearFBError pDrive POINTER TO AXIS_REF_SM3 — BOOL SMC_ClearFBError	TEST:=SMC_ ClearFBError(pDrive:=ADR(Axis));

2）相关变量

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
pDrive	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

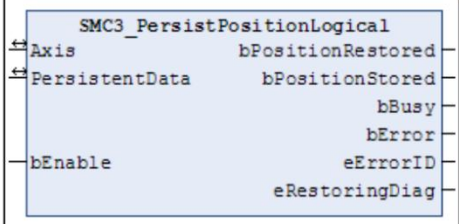
◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
SMC_ClearFBError	清除错误	BOOL	TRUE,FALSE	FALSE	True, 清除

4.1.47 SMC_PersistPositionLogical

该指令用来保持记录逻辑轴（在实轴或者虚轴右键“添加设备”来选择添加逻辑轴）的位置（断电重启控制器后，恢复断电前位置记录值）。

1)指令格式

指令	名称	图形表现	ST 表现
SMC3_ PersistPosition Logical	轴位置保持		SMC3_PersistPositionLogical0(Axis:= , PersistentData:= , bEnable:= , bPositionRestored=> , bPositionStored=> , bBusy=> , bError=> , eErrorID=> , eRestoringDiag=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_LOGICAL_ SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例
PersistentData	保持数据	SMC3_ PersistPositionLogical_ Data	-	-	存储位置信息的断电保持型数据结构

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
bEnable	执行	BOOL	TRUE,FALSE	FALSE	True 功能块执行，false 不执行功能块 若要在初始化期间还原上次存储的位置，则必须从应用程序启动时将该值置为 true

◆ 输出变量

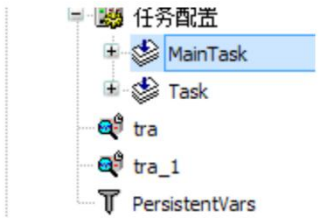
输出变量	名称	数据类型	有效范围	初始值	描述
bPosition Restored	位置恢复	BOOL	TRUE,FALSE	FALSE	TRUE, 轴重启后位置恢复完成
bPosition Stored	位置保存	BOOL	TRUE,FALSE	FALSE	TRUE, 调用功能快后保存位置完成
bBusy	FB 执行中	BOOL	TRUE,FALSE	FALSE	TRUE，功能块没有执行完成
bError	错误	BOOL	TRUE,FALSE	FALSE	TRUE, 异常发生
eErrorID	错误代码	SMC_ERROR	-	SMC_NO_ERROR	异常发生时，输出错误代码
eRestoring-Diag	恢复诊断	SMC3_ PersistPositionDiag	-	SMC3_ PersistPositionDiag. SMC3_PPD_ RESTORING_OK	位置恢复中的诊断信息 SMC3_PPD_RESTORING_OK: 位置成功恢复 SMC3_PPD_AXIS_PROP_ CHANGED: 轴参数有更改，无法恢复位置 SMC3_PPD_DATA_STORED_ DURING_WRITING: 功能块从轴参数数据结构复制数据，而不是从 PersistentData 数据中复制。 可能原因：非同步性持续变量、控制器崩溃死机

3）功能说明

◆ PLC 重启 bEnable 信号为 TRUE，则 bPositionRestroed 输出为 TRUE。 不支持虚轴跟实轴。

◆ PLC 重启后要恢复断电前“位置”需使用该功能块，并且为了记录断电前轴 “实际位置”需将 SMC3_ PersistPositionLogical_Data 配成持续型变量”。

◆ 使用方法（实轴编码器为多圈绝对值时）： PersistentVars 中声明 SMC3_PersistPositionLogical_Data 型数据




```
1  VAR_GLOBAL PERSISTENT RETAIN
2      persistentData3: SMC3_PersistPositionLogical_Data;
3  END_VAR
```

PLC 主任务（EthCat 任务）中调用

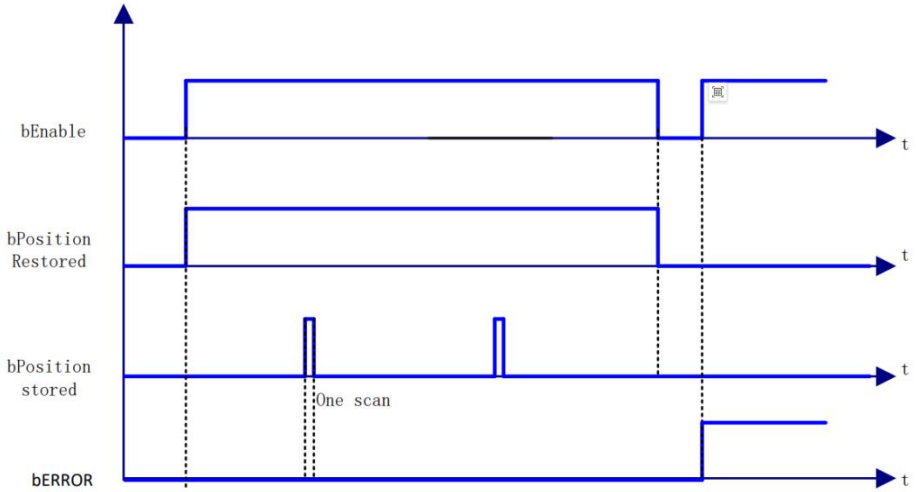
◆ 申明部分：

```
VAR
    SMC3_PersistPosition_3:SMC3_PersistPositionLogical;
END_VAR
```

◆ 程序部分：

```
//绝对值位置保存
SMC3_PersistPosition_1(Axis:=X_Axis , PersistentData:=persistentData1 ,bEnable:=TRUE );
```

◆ 时序图



4）错误说明

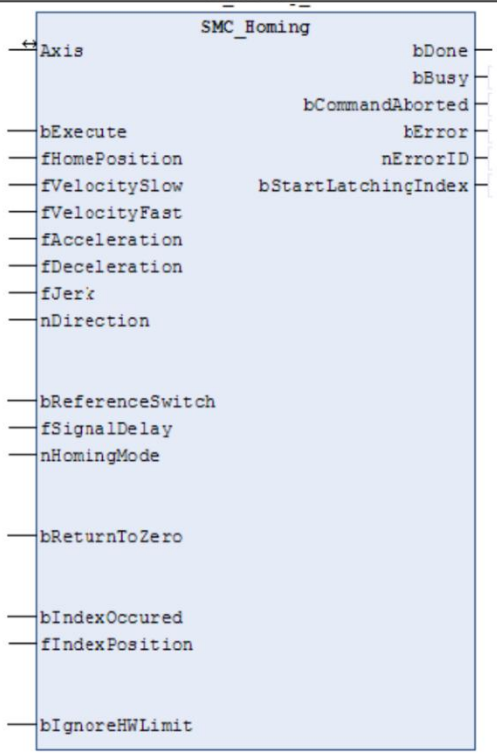
输入轴为虚拟轴或者实轴会导致错误输出；轴有错误时会导致错误输出。

【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.1.48 SMC_Homing

轴回零指令，与 MC_Home 有区别，MC_Home 在轴配置处设置回零方式，该指令为控制器控制的回零方式。

1）指令格式

指令	名称	图形表现	ST 表现
SMC_Homing	轴回零		SMC_Homing0(Axis:= , bExecute:= , fHomePosition:= , fVelocitySlow:= , fVelocityFast:= , fAcceleration:= , fDeceleration:= , fJerk:= , nDirection:= , bReferenceSwitch:= , fSignalDelay:= , nHomingMode:= , bReturnToZero:= , bIndexOccured:= , fIndexPosition:= , bIgnoreHWLimit:= , bDone=> , bBusy=> , bCommandAborted=> , bError=> , nErrorID=> , bStartLatchingIndex=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
bExecute	执行	BOOL	TRUE,FALSE	FALSE	True 功能块执行，false 不执行功能块
fHomePosition	原点设定位置	LREAL	-	0	回零后原点设定位置，单位为用户标定后的单位
fVelocitySlow	慢速	LREAL	-	0	离开参考开关后慢速设定速度
fVelocityFast	快速	LREAL	-	0	离开参考开关置位时，快速设定速度
fAcceleration	加速度	LREAL	-	0	加速度设定值
fDeceleration	减速度	LREAL	-	0	减速度设定值
fJerk	加速度导数	LREAL	-	0	Jerk in [u/s3]
nDirection	回零方向	MC_DIRECTION	-	negative	回零开始方向，参考 MC_DIRECTION
bReferenceSwitch	参考开关	BOOL	TRUE,FALSE	FALSE	连接参考开关，TRUE: 参开开关触发，FALSE: 参考开关闭合
fSignalDelay	延迟	LREAL	-	0	参考开关的传输时间，用来补偿死区时间。单位为秒。

输入变量	名称	数据类型	有效范围	初始值	描述
nHomingMode	回零模式	SMC_HOMING_MODE	-	-	参考 SMC_HOMING_MODE
bReturnTozero	返回零位	BOOL	TRUE,FALSE	FALSE	TRUE:回零完成后轴运行到位置零（注意:如果 fHomePosition=10，则回零完成后轴位置变为 10，bReturnTozero 为 ture 则回零完成后轴反向走 10 个单位到 0 位）
bIndexOccured		BOOL	TRUE,FALSE	FALSE	True, 标志脉冲记录，回零模式为 FAST_BSLOW_I_S_STOP，FAST_SLOW_I_S_STOP 时生效
fIndexPosition		LREAL	-	0	标志脉冲时记录的位置
bIgnoreHWLimit	忽略硬限位	BOOL	TRUE,FALSE	FALSE	TRUE, 设定硬件限位开关使能为 false，如果相同的物理开关用于硬件限位开关和参考开关，那么硬件控制将被设置为假

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
bDone		BOOL	TRUE,FALSE	FALSE	True, 回零完成
bBusy		BOOL	TRUE,FALSE	FALSE	True, 功能块正在生效中
bCommandAborted		BOOL	TRUE,FALSE	FALSE	True, 功能块被其他动作指令中断
Error		BOOL	TRUE,FALSE	FALSE	True, 错误发生
ErrorID		SMC_ERROR	-	0	错误代码，枚举型变量，查看帮助 smc_error 查看具体报警代码
bStartLatchingIndex		BOOL	TRUE,FALSE	FALSE	由“bIndexOccured”和“fIndexPosition”共同作用产生

◆ 回零模式（SMC_HOMING_MODE）

枚举名称	类型	初始值	描述
FAST_BSLOW_S_STOP	SMC_HOMING_MODE	0	按照设定方向以快速度走向原点开关，碰到原点开关后反向以慢速离开原点开关，离开后先执行 MC_setPosition 将当前位置设为 fHomePosition 设定值，然后执行 MC_stop
FAST_BSLOW_STOP_S	SMC_HOMING_MOD	1	按照设定方向以快速走向原点开关，碰到原点开关后反向以慢速离开原点开关，离开后先执行 MC_stop 将轴停止，然后执行 MC_setPosition，将当前位置设为 fHomePosition 设定值
FAST_BSLOW_I_S_STOP	SMC_HOMING_MOD	2	按照设定方向以快速走向原点开关，碰到原点开关后反向以慢速离开原点开关，bIndexOccured 信号到时先执行 MC_setPosition 再执行 MC_stop
FAST_SLOW_S_STOP	SMC_HOMING_MOD	4	按照设定方向以快速走向原点开关，碰到原点开关后以慢速离开原点开关，离开后先执行 MC_setPosition 将当前位置设为 fHomePosition 设定值，然后执行 MC_stop
FAST_SLOW_STOP_S	SMC_HOMING_MOD	5	按照设定方向以快速走向原点开关，碰到原点开关后以慢速离开原点开关，离开后先执行 MC_stop，然后执行 MC_setPosition 将当前位置设为 fHomePosition 设定值。
FAST_SLOW_I_S_STOP	SMC_HOMING_MOD	6	按照设定方向以快速走向原点开关，碰到原点开关后反向以慢速离开原点开关，bIndexOccured 信号到时先执行 MC_setPosition 再执行 MC_stop

3） 功能说明

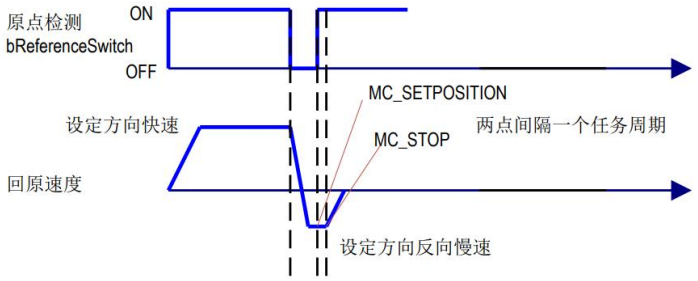
◆ SMC_HOMING 通过 bExecute 的上升沿启动之后，轴将会按照速度 fVelocityFast 并以 nDirection 定义的方 向开始运动，直到 bReferenceSwitch = FALSE。然后轴将会缓慢停止并按照相反的方向以速度 fVelocitySlow 离开参考开关。bReferenceSwitch = TRUE 后回零完成，

◆ 使能回零指令后 bReferenceSwitch 的状态为 ON->OFF->ON，在 OFF->ON 的上升沿回零完成，设置参考位置。

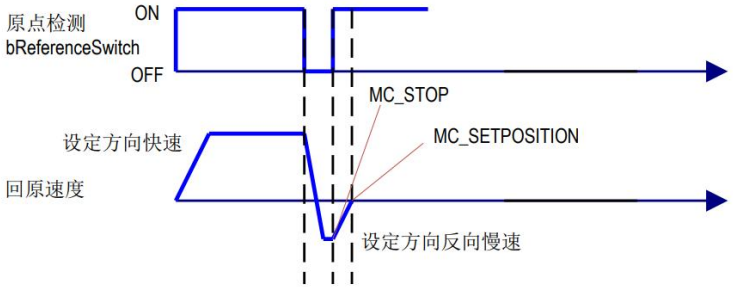
◆ 参考位置 =fHomePostion+（（fSignalDelay*1000+1 个 DC 时钟周期）/1000）*fVelocitySlow 实际就是补 偿了设置的 bReferenceSwitch 采样延迟和一个通讯周期位移延迟。

◆ 如果 bReturnToZero=TRUE，bReferenceSwitch 的状态在 OFF->ON 的上升沿将参考位置设置为 fHomePostion+（（fSignalDelay*1000+1 个 DC 时钟周期）/1000）*fVelocitySlow，然后按照速度 fVelocityFast 运行到 0 位置。

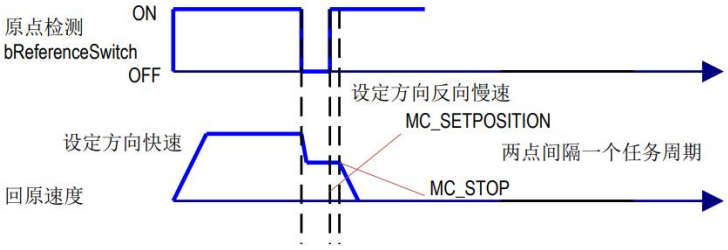
注意：Done 完成信号后，轴位置设定为：fHomePosition。设定的时机跟 nHomingMode 有关（详情参考 SMC_HOMING_MODE）。下图为几种回零模式： ① 回零模式“0”时：



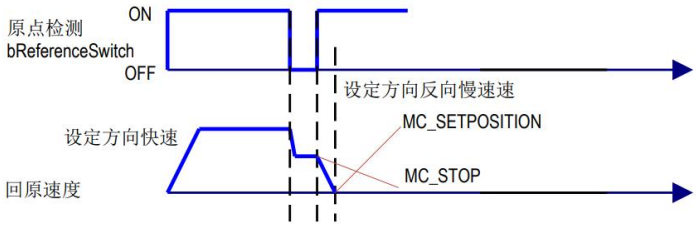
② 回零模式“1”时



③ 回零模式“4”时

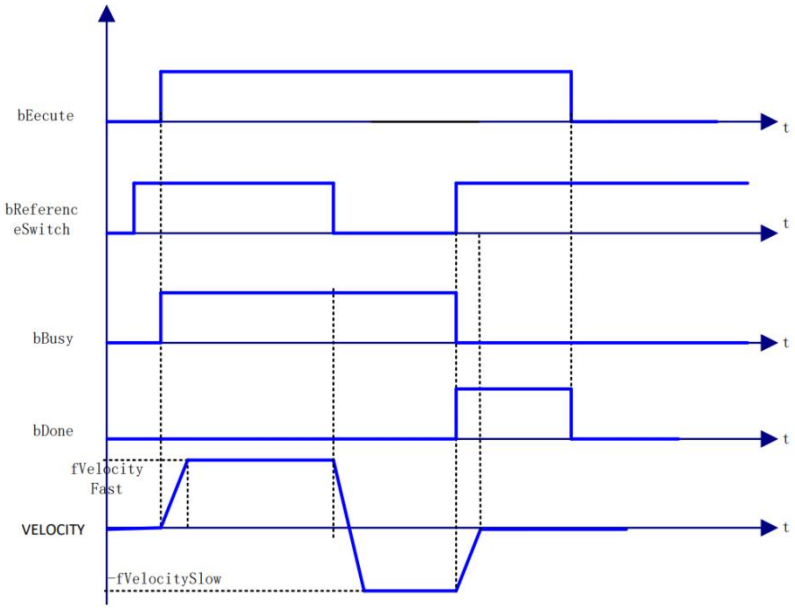


④ 回零模式“5”时

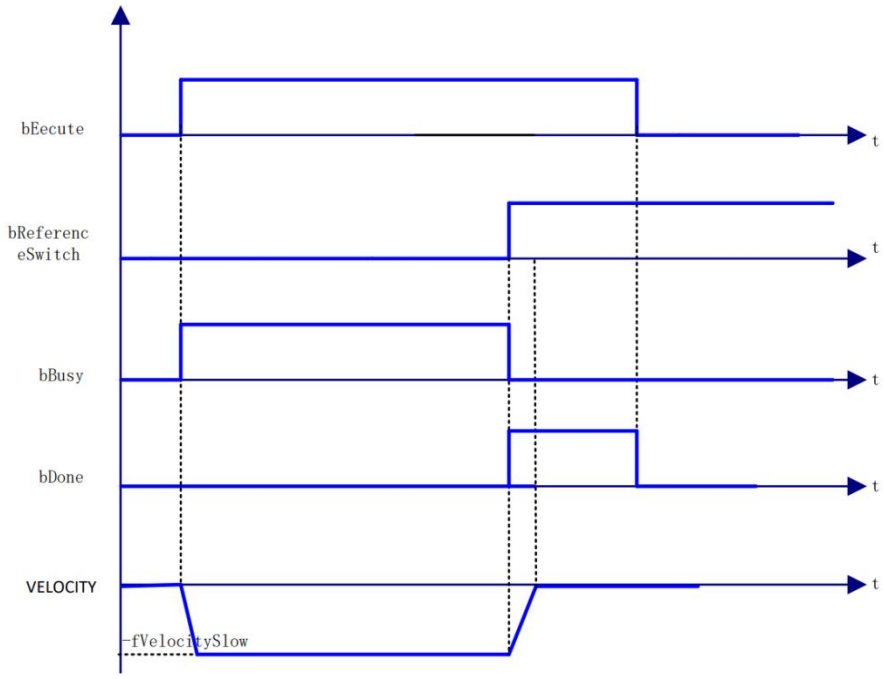


4）时序图

① 指令执行时 bReferenceSwitch TRUE 时



② 指令执行时 bReferenceSwitch FALSE 时



5）错误说明

输入轴类型出错。

轴有错误。

轴没有使能

速度或加速度无效。

【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.1.49 SMC_SetSoftwareLimits

设置上位机软件限位。

1）指令格式

指令	名称	FB/FC	LD 表现	ST 表现
SMC_ SetSoftwareLimits	设置软件限制	FB		

2）相关变量

◆输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
bExecute	启动	BOOL	TRUE,FALSE	FALSE	输入的一个上升沿将启动功能块的处理
SWL_Activated	激活软件限位	BOOL	TRUE,FALSE		TRUE: 激活软件限位
SWL_Positive	正限位	LREAL	-	0	正限位
SWL_Negative	负限位	LREAL	-	0	负限位
SWL_Error_Decelerate	减速配置	BOOL	TRUE,FALSE		无效设置，默认遇限位需减速
SWL_Error_Deceleration	超限位错误响应减速度	LREAL	-	0	遇正限位减速度
SWL_Error_MaxDistance	最大减速距离	LREAL	-	0	最大减速距离，仅在正限位使用

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	执行完成	BOOL	TRUE,FALSE	FALSE	TRUE：复位完成
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	DWORD	-	0	异常发生时，输出错误代码

3）功能说明

设置上位机的正负位置限制以及遇软限位报错响应。

4）注意事项

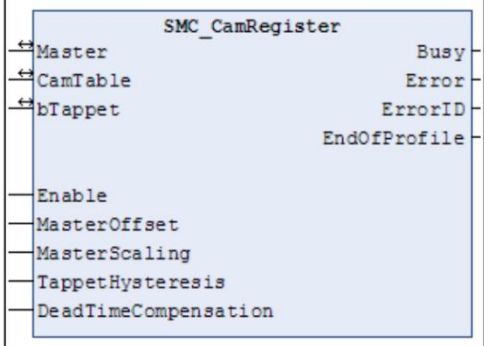
超限位响应的减速度是从三个参数中取最大值进行减速。这三个参数分别是 fSwLimitDeceleration（功能块 设置参数 SWL_Error_Deceleration）、后台动态限制配置参数 fSWMaxDeceleration，以及按最大减速距离 fSWErrorDistance 进行停止所计算的减速度。

4.2 轴组指令（主/从轴指令）

4.2.1 SMC_CamRegister

实现凸轮挺杆控制（凸轮开关）。凸轮编辑的时候可以不编辑主从轴曲线，只需配置挺杆表就可通过该功能块实现挺杆控制。

1）指令格式

指令	名称	图形表现	ST 表现
SMC_ CamRegister	凸轮挺杆控制		SMC_CamRegister0(Master:= , CamTable:= , bTappet:= , Enable:= , MasterOffset:=0 , MasterScaling:= 1, TappetHysteresis:= , DeadTimeCompensation:= , Busy=> , Error=> , ErrorID=> , EndOfProfile=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Master	主轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例
CamTable	凸轮表	MC_CAM_REF	-	-	映射到一个电子凸轮，即一个电子凸轮实例
bTappet	挺杆输出	ARRAY [1..MAX_ NUM_TAPPETS] OF BOOL	-	-	挺杆点的输出

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行	BOOL	TRUE,FALSE	FALSE	True 功能块执行，false 不执行功能块
Masteroffset	主轴偏移	LREAL	-	0	主轴偏移量
MasterScaling	主轴标尺	LREAL	-	1	主轴线性缩放因子
TappetHysteresis	挺杆阻尼	LREAL	-	0	挺杆控制阻尼系数
DeadTime Compensation	死区时间补偿	LREAL	-	0	死区补偿时间单位为 S，根据主轴当前速度线性补偿挺杆输出，可为正值，可为负值

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Busy	执行中	BOOL	TRUE,FALSE	FALSE	TRUE, 功能块执行中
Error	错误	BOOL	TRUE,FALSE	FALSE	TRUE, 异常发生
ErrorID	错误代码	SMC_ERROR	-	SMC_NO_ERROR	异常发生时，输出错误代码
EndofProfile	曲线周期完成	BOOL	TRUE,FALSE	FALSE	True, 主轴位置大于等于设定周期

3）功能说明

◆ Enable 信号为 TRUE，如果没有错误输出则 Busy 输出为 TRUE，执行挺杆控制。

◆ 该控制功能块跟电子凸轮中的从轴没有关系，只需配置主轴周期与挺杆表。

◆ “bTappet”为一维布尔型结构体（MAX_NUM_TAPPETS=512），且 bTappet[i] 对应第 i 个挺杆点的输出 ◆ DeadTimeCompensation 单位为 S/ 秒，设置为正值时则超前输出挺杆信号，设置为负值时则滞后输出挺杆 信号。比如说设置为 0.02 秒 Ethcat 任务周期设置为 4ms 则根据主轴线性速度 v，挺杆输出位置为 P，则挺杆在主 轴设定位置 =P-V*0.02 处输出挺杆值。反之如果设置为 -0.02 秒则主轴设定位置大于等于 P 后滞后五个周期 挺杆信号输出。

◆ 该功能块使用样例：

变量声明：

```
VAR
    TPP:ARRAY[1..MAX_NUM_TAPPETS] OF BOOL;
    SMC_CamRegister0: SMC_CamRegister;
END_VAR
```

程序部分：

```
SMC_CamRegister0(
Master:=Virtual_X ,
CamTable:=Cam,
bTappet:=TPP ,
Enable:=TRUE ,
MasterOffset:=0 ,
MasterScaling:= 1,
TappetHysteresis:= 0,
DeadTimeCompensation:=0 ,
Busy=> ,
Error=> ,
ErrorID=> ,
EndOfProffle=> );
```

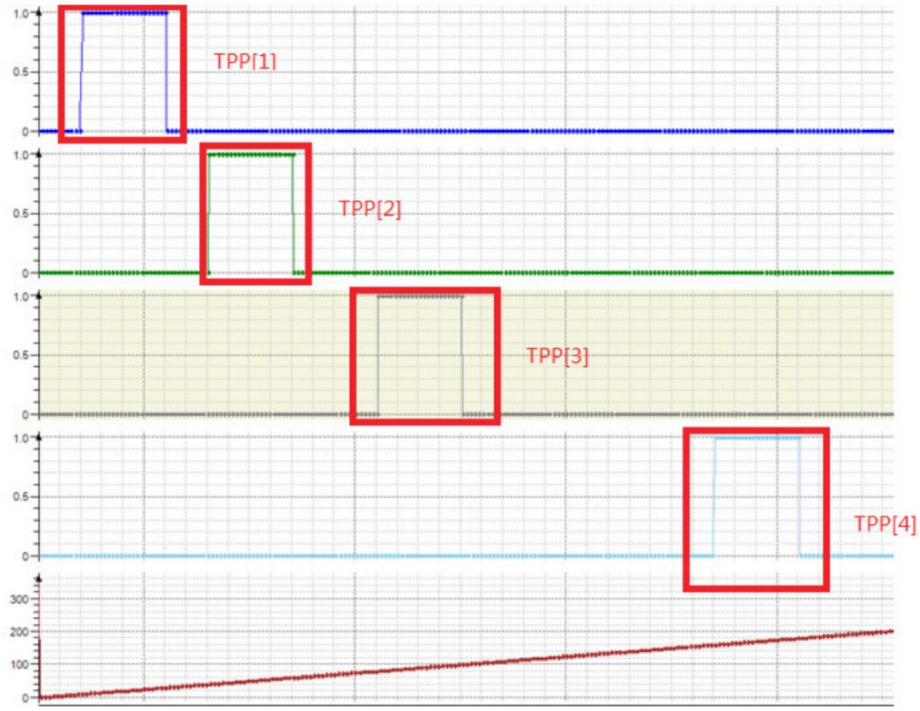
Cam 编辑如下图：

cam | cam 表 | 挺杆 | 挺杆表

	Track ID	X	positive pass	negative pass
	1			
		10	打开	无
		30	关闭	无
	2			
		40	打开	无
		60	关闭	无
	3			
		80	打开	无
		100	关闭	无
	4			
		160	打开	无
		180	关闭	无

启动 Virtual_X 轴：

监控曲线如下图：

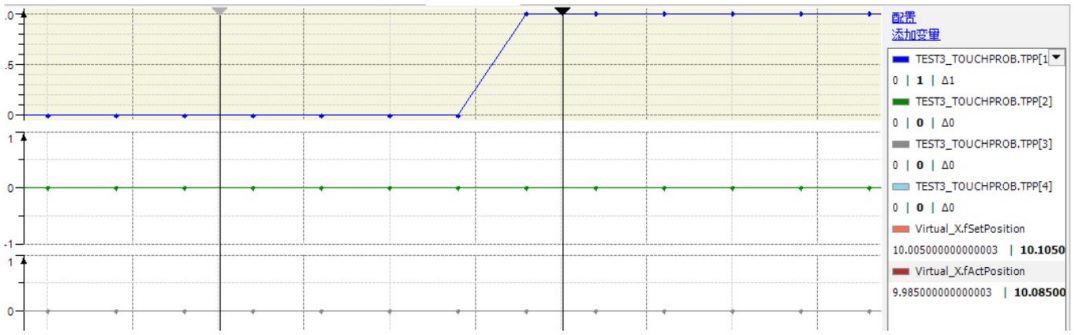


当死区补偿时间设置为 -0.02 秒时

```
SMC_CamRegister0(
Master:=Virtual_X ,
CamTable:=Cam,
bTappet:=TPP ,
Enable:=TRUE ,
MasterOffset:=0 ,
```

```
MasterScaling:= 1,
TappetHysteresis:= 0,
DeadTimeCompensation:=-0.02 ,
Busy=> ,
Error=> ,
ErrorID=> ,
EndOfProffle=> );
```

挺杆输出滞后五个任务周期（任务周期为 4ms），如下图所示：



4）错误说明

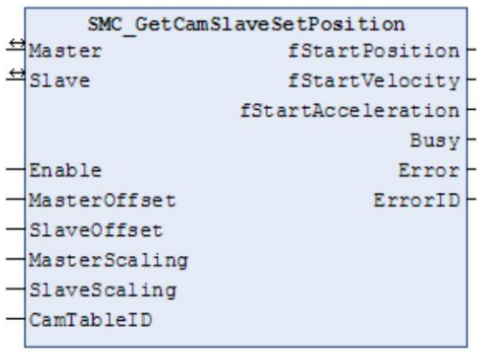
轴有错误、轴没有使能、偏移值或者标尺值设置超过主轴范围。

【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.2.2 SMC_GetCamSlaveSetPosition

读取凸轮表从轴位置、速度、加速度信息。

1）指令格式

指令	名称	图形表现	ST 表现
SMC_GetCam SlaveSet Position	获取凸轮从轴位置		SMC_GetCamSlaveSetPosition0(Master:= , Slave:= , Enable:= , MasterOffset:= , SlaveOffset:= , MasterScaling:= , SlaveScaling:= , CamTableID:= fStartPosition=> , fStartVelocity=> , fStartAcceleration=> , Busy=> , Error=> , ErrorID=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Master	主轴	AXIS_REF	-	-	映射到一个轴
Slave	从轴	AXIS_REF	-	-	映射到一个轴

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行	BOOL	TRUE,FALSE	FALSE	True 功能块执行，false 不执行功能块
Masteroffset	主轴偏移	LREAL	-	0	凸轮表主轴偏移
Slaveoffset	从轴偏移	LREAL	-	0	凸轮表从轴偏移
MasterScaling	主轴缩放	LREAL	-	1	凸轮表主轴缩放因子
SlaveScaling	从轴缩放	LREAL	-	1	凸轮表从轴轴缩放因子
CamTableID	凸轮 ID	MC_CAM_ID	-	-	凸轮表 ID

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
fStartPosition	从轴位置	LREAL	-	0	根据凸轮表跟当前主轴信息所获得的从轴位置
fStartVelocity	从轴速度	LREAL	-	0	根据凸轮表跟当前主轴信息所获得的从轴速度
fStartAcceleration	从轴加速度	LREAL	-	0	根据凸轮表跟当前主轴信息所获得的从轴加速度
busy	执行中	BOOL	TRUE,FALSE	FALSE	TRUE, 表示功能块正在执行中
Error	错误	BOOL	TRUE,FALSE	FALSE	TRUE, 异常发生
ErrorID	错误代码	SMC_ERROR	-	SMC_NO_ERROR	异常发生时，输出错误代码

3）功能说明

该指令计算的输出值为：Y=（cam（(凸轮起始主轴查表位置 +Masteroffset)* MasterScaling）+slaveoffset）*SlaveScaling，Cam 为凸轮表函数。例如：凸轮起始主轴位置为 0，主从轴缩放为 1，Masteroffset 为 100，slaveoffset 为 0，则功能块输出为凸轮表在 100 所对应的从轴位置。该功能块读取从轴位置只需要凸轮表构建成功，对于主从轴是否运转并没有要求。

示例：

声明：

```
SMC_GetCamSlaveSetPosition0: SMC_GetCamSlaveSetPosition;
ENABLE: BOOL;
    MC_CamTableSelect0: MC_CamTableSelect;
```

程序：

```
MC_CamTableSelect0(
    Master:=Virtual_X ,
    Slave:=Virtual_Y ,
    CamTable:=Cam ,
    Execute:= ,
```

```
        Periodic:=TRUE ,
        MasterAbsolute:=0 ,
        SlaveAbsolute:=0 ,
        Done=> ,
        Busy=> ,
        Error=> ,
        ErrorID=> ,
        CamTableID=> );
    SMC_GetCamSlaveSetPosition0(
        Master:= Virtual_X,
        Slave:= Virtual_Y,
        Enable:=ENABLE ,
        MasterOffset:= 100,
        SlaveOffset:=0 ,
        MasterScaling:=1 ,
        SlaveScaling:= 1,
        CamTableID:=MC_CamTableSelect0.CamTableID,
        fStartPosition=> ,
        fStartVelocity=> ,
        fStartAcceleration=> ,
        Busy=> ,
        Error=> ,
        ErrorID=> );
```

Enable	BOOL	TRUE
MasterOffset	LREAL	100
SlaveOffset	LREAL	0
MasterScaling	LREAL	1
SlaveScaling	LREAL	1
CamTableID	MC_CAM_ID	
fStartPosition	LREAL	33.580246913580254

4）错误说明

Error 输出为 True，则指令错误输出；

参考 ErrorID,SMC_ERROR 确定错误原因。

【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.2.3 SMC_GetTappetValue

与 MC_CamIn 指令配合使用，获取当前挺杆输出值。

1）指令格式

指令	名称	图形表现	ST 表现
SMC_GetTappetValue	获取挺杆输出值	SMC_GetTappetValue Tappets → bTappet iID bInitValue bSetInitValueAtReset	SMC_GetTappetValue0(Tappets:= , iID:=, bInitValue:= , bSetInitValueAtReset:= , bTappet=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Tappets	挺杆	SMC_TappetData	-	-	映射到一个挺杆

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
iID	挺杆组号	INT	-	0	挺杆的组 ID
bInitValue	初始值	BOOL	-	-	功能块第一次调用时挺杆初始化值
bSetInitValueAtReset		BOOL	-	-	TRUE,MC_CamIn 功能块重启时挺杆输出值将被初始化为 bInitValue 设定值 FALSE, 挺杆输出值将会保持当 MC_CamIn 功能块重启时。

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
bTappet	挺杆输出	BOOL	-	FALSE	挺杆值

3）功能说明

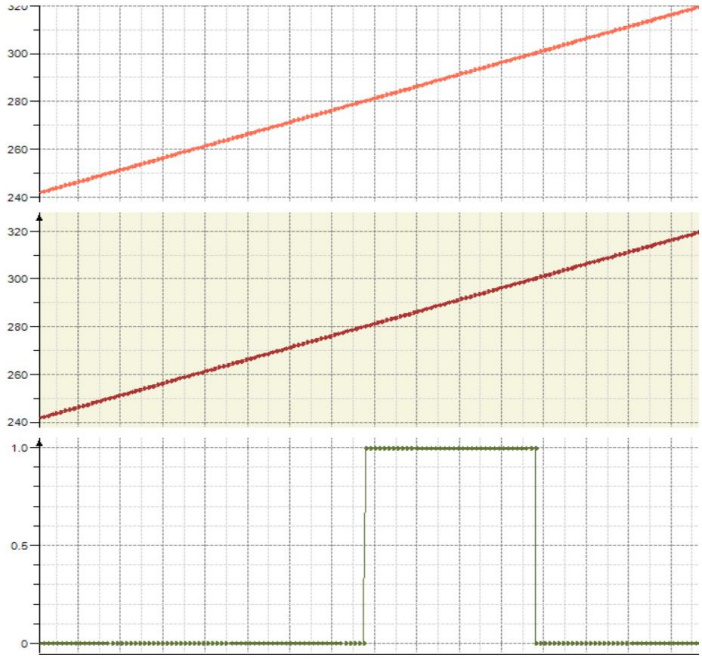
◆ 该功能块需要与 MC_CamIn 指令配合使用。

◆ 该功能块与 SMC_CamRegister 功能一样都是读取挺杆输出，但是两者存在冲突，所以在同一个凸轮挺杆表中只使用一种。

使用示例：

```
MC_CamIn0(
    Master:=Virtual_X ,
    Slave:= Virtual_Y,
    Execute:= ,
    MasterOffset:= 0,
```

```
        SlaveOffset:= 0,
        MasterScaling:=1 ,
        SlaveScaling:= 1,
        StartMode:= 1,
        CamTableID:= MC_CamTableSelect0.CamTableID,
        VelocityDiff:= ,
        Acceleration:= ,
        Deceleration:= ,
        Jerk:= ,
        TappetHysteresis:= ,
        InSync=> ,
        Busy=> ,
        CommandAborted=> ,
        Error=> ,
        ErrorID=> ,
        EndOfProffle=> ,
        Tappets=> );
    SMC_GetTappetValue0(
        Tappets:= MC_CamIn0.Tappets,
        iID:=2,
        bInitValue:= false,
        bSetInitValueAtReset:=true ,
        bTappet=> );
```



4）错误说明

轴有错误 ；

轴没有使能 ；

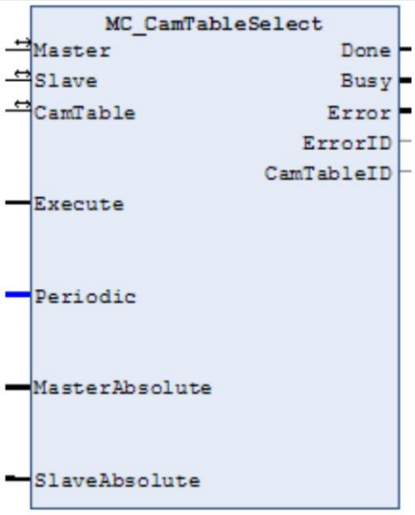
CamTable ID 没有指向。

【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.2.4 MC_CamTableSelect

指定凸轮表，与 MC_CamIn 指令配合使用。

1）指令格式

指令	名称	图形表现	ST 表现
MC_ CamTableSelect	凸轮表指定		MC_CamTableSelect0(Master:= , Slave:= , CamTable:=), Execute:= , Periodic:= , MasterAbsolute:= , SlaveAbsolute:= , Done=> , Busy=> , Error=> , ErrorID=> , CamTableID=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Master	主轴	AXIS_REF_SM3	-	-	映射到主轴，即 AXIS_REF_SM3 的一个实例
Slave	从轴	AXIS_REF	-	-	映射到从轴，即 AXIS_REF_SM3 的一个实例
CamTable	选择表	MC_CAM_REF	-	-	映射到 CAM 表格描述，即 MC_CAM_REF 的一个实例

使用注意事项：

主轴和从轴不能指定为同一轴，否则会有报错输出，CamTable 所对应的凸轮表编辑需要正确无误，否则也会导致指令报错。主轴、从轴可以为实轴也可以是虚轴。

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行	BOOL	TRUE,FALSE	FALSE	上升沿信号，执行指令
Periodic	重复模式	BOOL	TRUE,FALSE	FALSE	指定是反复执行指定的凸轮表还是只执行一次 TRUE: 重复 False: 不重复

输入变量	名称	数据类型	有效范围	初始值	描述
MasterAbsolute	主轴绝对模式	BOOL	TRUE,FALSE	FALSE	指定主轴跟踪距离坐标系是按绝对位置还是相对位置 1：绝对位置，0: 相对位置
SlaveAbsolute	从轴绝对模式	BOOL	TRUE,FALSE	FALSE	与 MC_CamIn 指令中的 StartMode 综合指定从轴当前指令位置为凸轮表输出的绝对（当前主轴位置对应的凸轮表输出值）还是相对（凸轮表输出值叠加指令开始时从轴位置） 1：绝对位置，0: 相对位置。

使用注意事项：

MasterAbsolute、SlaveAbsolute 选择不当可能会导致电子凸轮输出跳变，所以设定前请确定设定凸轮曲线工作方式。

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	完成	BOOL	TRUE,FALSE	FALSE	选择完成时为 TRUE,
Busy	执行中	BOOL	TRUE,FALSE	FALSE	选择中没有完成时为 TRUE
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ERROR	参阅 SMC_ERROR	0	异常发生时，输出错误代码
CamTableID	生效 CAMID	MC_CAM_ID	-	-	选择生效的 Cam_ID，与 MC_CamIn 指令中的 CamTableID 配合使用

使用注意事项：

Error 发生时请对照 ErrorID 查看帮助里面 SMC_ERROR .

3）功能说明

◆ 本指令指定电子凸轮运行所需凸轮表，所以在使用本指令之前先要将凸轮表编辑好（凸轮编辑器编辑或者在线编辑好）。

◆ Excute 上升沿，执行指定凸轮表，亦可凸轮表更新后刷新指定凸轮表。

◆ Done 信号输出为 TRUE 时，则输出变量“CamTableID”生成并且生效。

◆ 指令执行中，Busy 信号输出 TRUE，Done 信号 TRUE、Busy 信号为 FALSE。

◆ MasterAbsolute、SlaveAbsolute、Periodic 参数具体作用参考 MC_CamIn 指令详情

4）错误说明

◆ 主轴和从轴不能指定为同一轴，否则会有报错输出。。

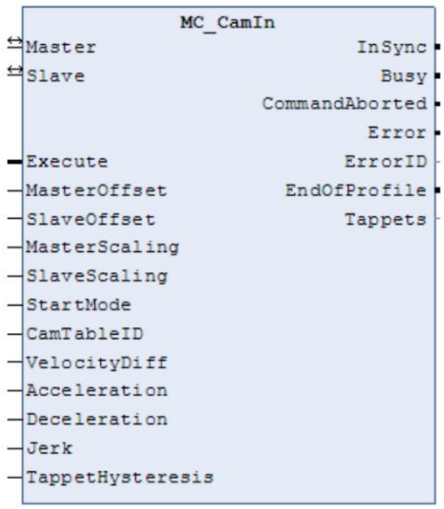
◆ CamTable 所对应的凸轮表编辑需要正确无误否则会错误输出。

【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.2.5 MC_Camin

使用指定的凸轮表开始执行电子凸轮动作，可根据应用需求指定主从轴偏置值、缩放比和工作模式。

1）指令格式

指令	名称	图形表现	ST 表现
MC_CamIn	凸轮动作开始		MC_CamIn0(Master:= , Slave:= , Execute:= , MasterOffset:= , SlaveOffset:= , MasterScaling:= , SlaveScaling:= , StartMode:= , CamTableID:= , VelocityDiff:= , Acceleration:= , Deceleration:= , Jerk:= , TappetHysteresis:= , InSync=> , Busy=> , CommandAborted=> , Error=> , ErrorID=> , EndOfProfile=> , Tappets=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Master	主轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例
Slave	从轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

使用注意事项：

主轴和从轴不能指定为同一轴，否则会有报错输出。

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行凸轮功进入能块	BOOL	TRUE,FALSE	FALSE	上升沿，执行电子凸轮
MasterOffset	主轴偏置	LREAL	负数，正数，0	0	以指定的偏移值移动主轴的相位
SlaveOffset	从轴偏执	LREAL	负数，正数，0	0	以指定的偏移值移动从轴的相位
MasterScaling	主轴预编译比例	LREAL	>0.0	1	以指定的比例放大 / 缩小主轴的相位
SlaveScaling	从轴预编译比例	LREAL	>0.0	1	以指定的比例放大 / 缩小从轴轴的相位
StartMode	从轴相对凸轮输出模式	MC_ StartMode	0~4	absolute	0: absolute 绝对位置 :1: relative 相对位置 : 2: ramp_in (斜坡切入) 3: ramp_in_pos (正向斜坡切入) 4: ramp_in_neg 反向斜坡切入
CamTableID	表格编号	MC_CAM_ ID	>0	-	定义 cam 表格的使用，与 MC_ CamTableSelect 的输出点 CamTableID 配合使用
VelocityDiff	耦合速度	LREAL	>0.0	0	与 ramp_in 不同的最大速度
Acceleration	加速度	LREAL	>0.0	0	ramp_in 时加速度

输入变量	名称	数据类型	有效范围	初始值	描述
Deceleration	减速度	LREAL	>0.0	0	ramp_in 时减速度
Jerk	跃度	LREAL	>0.0	0	ramp_in 的加速度
TappetHysteresis	挺杆系数范围	LREAL	>0.0	0	挺杆的阻尼系数

使用注意事项：

主轴和从轴不能指定为同一轴，否则会有报错输出。

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
InSync	凸轮生效	BOOL	TRUE,FALSE	FALSE	主轴和从轴建立凸轮关系后，InSync 被置位，当指令的执行条件 OFF 时，InSync 被复位。
Busy	同步运行中	BOOL	TRUE,FALSE	FALSE	Execute 输入上升沿时，置位 TRUE， TRUE 时表示凸轮关系耦合中，需用 Cam_out 指令复位，指令执行条件复位不能复位该状态。
CommandAborted	指令中断	BOOL	TRUE,FALSE	FALSE	从轴被其他控制指令中断输出为 TRUE
Error	错误	BOOL	TRUE,FALSE	FALSE	如果检测到有错误，Error 位被置位；当指令的执行条件 OFF 时，Error 位被复位。
ErrorID	错误代码	SMC_ ERROR	参阅 SMC_ ERROR	0	异常发生时，输出错误代码
EndOfProfile	曲线完成	BOOL	-	FALSE	如果 MC_CamTableSelect 指令执行时 Periodic 参数为 0（非周期），当凸轮曲线执行完一次后，EndOfProfile 位被置位，当指令的执行条件 OFF 时，EndOfProfile 位被复位。
Tappets		SMC_ TappetData	-	-	关联一个凸轮挺杆，可被 MC_GetTappetValue 指令读出

3）功能说明

◆ Execute 上升沿、轴没报错、选择凸轮表正确则本指令启动；

◆ 在一个凸轮系统中，要调用一条凸轮曲线，先调用 MC_CamTableSelect 指令选择相应的凸轮表，再执行 MC_CamIn；如要更换凸轮曲线，则再调用 MC_CamTableSelect 指令重新选择凸轮表。

◆ 需使用 Camout 指令断开主从轴的凸轮耦合关系。

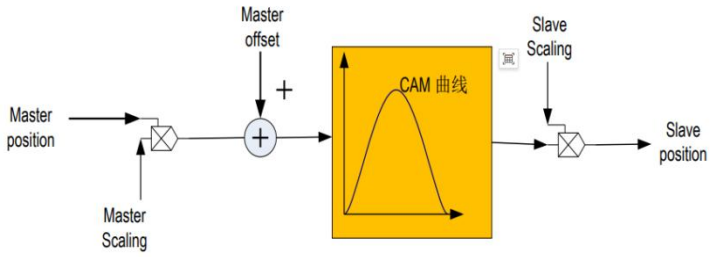
◆ 该指令执行时 ，该指令的从轴再执行其它运动指令时，从轴和主轴之间的凸轮关系会解除，并且 CommandAborted 输出为 TRUE。

4）指令详情

下面对指令详细说明：

◆ 指令启动条件 在主轴停止中、位置控制中、速度控制中、同步控制中任意状态下都可以启动本指令注意：凸轮从轴位置设定值要在软件限位值以内，否则会导致错误输出指令。

◆ 凸轮曲线中齿合点的计算方法如下：



由上图得出计算方法如下

Position_Slave = SlaveScaling*CAM(MasterScaling*MasterPosition + MasterOffset) + SlaveOffset

该公式中的主轴位置、从轴位置并不是代表实际物理轴的位置，而是凸轮函数曲线相关的主从轴位置。主从轴位置跟主从实轴位置之间的关系有详细描述。

注意：主轴位置、从轴位置均指的是凸轮函数曲线所需的主从轴位置，而不是主轴、从轴物理实轴的位置。

◆ 周期模式与 EndOfProffle 的关系：

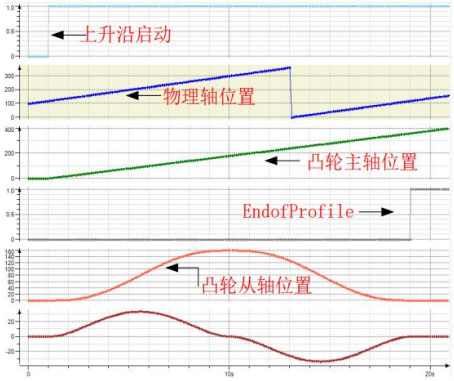
周期模式非周期模式决定了主轴到终止位置后电子凸轮是否要再次进行。

非周期模式：MC_CamTableSelect 指令 Periodic 选择 False；

非周期模式时，凸轮完成 EndofProffle 信号输出为 True，执行输入 FALSE 则 EndofProffle 输出 FALSE。此时凸轮只执行一个主轴周期。

注意：主轴周期指的是电子凸轮主轴位置从起始位置到终止位置的范围。

① 周期模式：MC_CamTableSelect 指令 Periodic 选择 TRUE

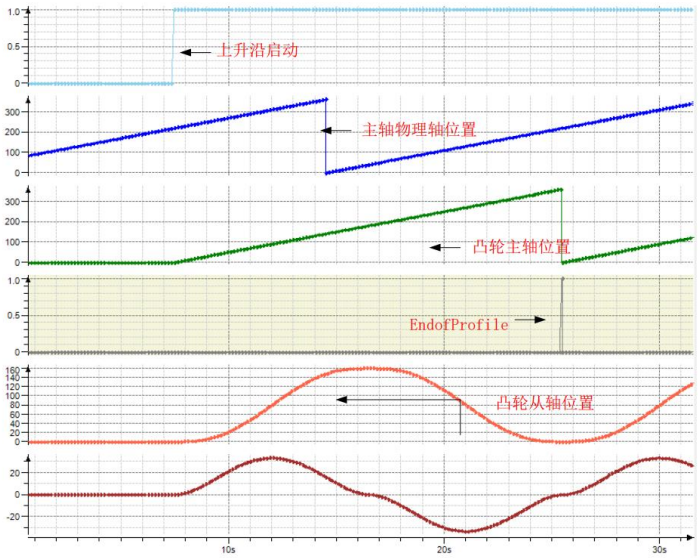


此时凸轮一个主轴周期完成后会接着执行下一个周期，且 EndofProffle 信号 TRUE 输出只维持一个任务周期。

注意： 当凸轮主轴位置大于等于凸轮终止位置时 EndofProffle 信号输出为 TRUE，并且凸轮主轴位置更新为凸轮起 始位置 + 大于终止位置部分。例如：电子凸轮主轴起始位置为 0、终止位置为 360、主从轴缩放设置为 1、主 从轴偏移值设为 0、任务周期 2ms，主轴速度为 100。当某个任务周期凸轮主轴位置为 359.99，那么下个周 期 EndofProffle 输出为 True 且主轴位置变为 359.99+100*0.002-360=0.19。

周期模式下设计的凸轮曲线起始位置跟终止位置最好能保持平滑过渡，否则会产生跳变

比如说起始速度为 0，终止速度不为 0，会导致主轴在周期完成和新周期开始处产生跳变。



StartMode 与 MC_CamTableSlect 中主从轴绝对相对模式的关系：

绝对模式：在新的电子凸轮循环开始时，电子凸轮的计算与当前从轴的位置无关。如果从轴相对于主轴起始 位置不同于从轴相对于主轴的终止位置，其将造成一个跳变。

相对模式：新的电子凸轮会根据当前从轴的位置进行改变；也就是说，从轴在上一个电子凸轮循环结束的位置， 会被现在的电子凸轮运动作为“从轴偏移”进行位置相加的计算。但是，如果在电子凸轮定义中，与主轴起 始位置对应的从轴位置不是 0，其将造成一个跳变。

斜坡输入：通过增加一个补偿运动（根据极限值 VelocityDiff、加速度、减速度得出的运动）来防止电子凸 轮在开始时的潜在跳变。因此，只要从轴是旋转的方式，正向斜坡输入选项则只进行正向补偿，而反向斜坡 输入则只进行反向补偿。对于线性运动的从轴，补偿方向可以自动实现，也就是说，正向斜坡输入和反向斜 坡输入可以被用斜坡输入的方式进行解释。）

关系表如下表所示：

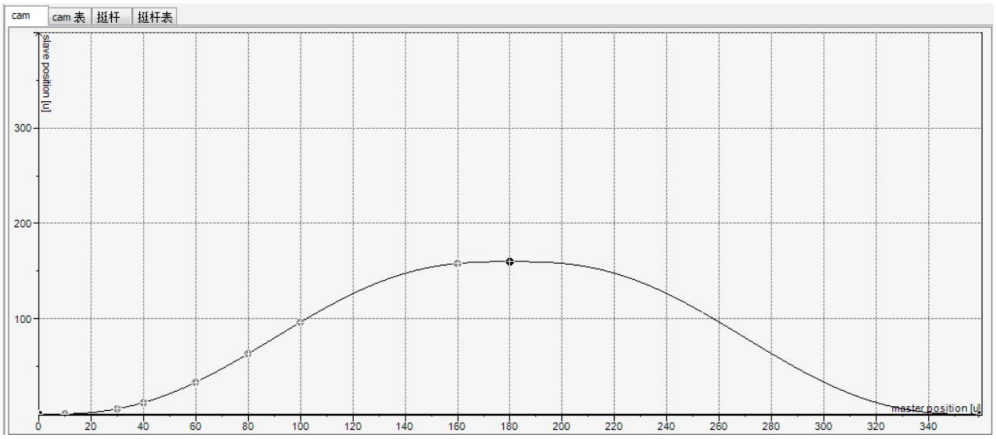
MC_CamTableSelect.MasterAbsolute	主轴模式
absolute	绝对模式
relative	相对模式

MC_CamIn.StartMode	MC_CamTableSelect.SlaveAbsolute	从轴模式
absolute	TRUE	绝对模式
absolute	FALSE	相对模式
relative	TRUE	相对模式
relative	FALSE	相对模式
ramp_in	TRUE	斜坡切入绝对模式
ramp_in	FALSE	斜坡切入相对模式

MC_CamIn.StartMode	MC_CamTableSelect.SlaveAbsolute	从轴模式
ramp_in_pos	TRUE	正向斜坡切入绝对模式
ramp_in_pos	FALSE	正向斜坡切入相对模式
ramp_in_neg	TRUE	反向斜坡切入绝对模式
ramp_in_neg	FALSE	反向斜坡切入相对模式

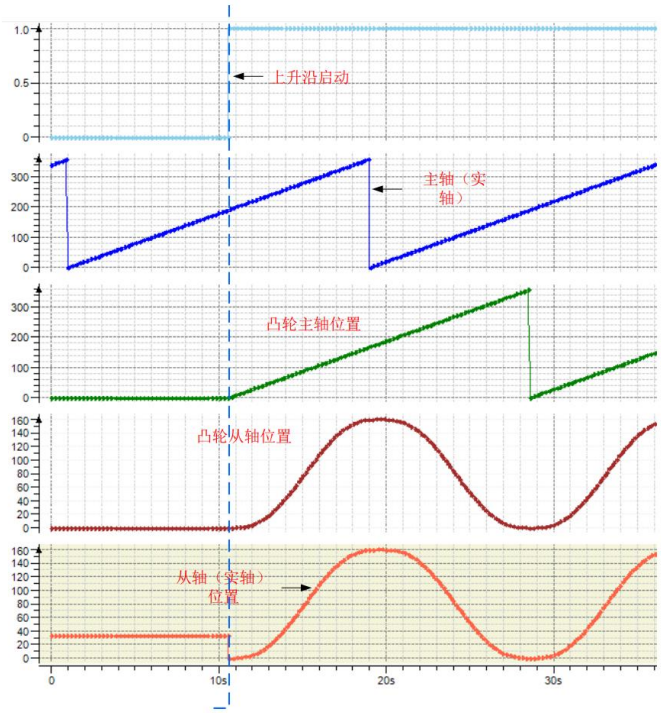
详细关系描述如下：

凸轮主轴范围（0-360）、凸轮从轴范围为（0-180）、周期模式、主从轴偏移值 0、主从轴缩放比 1。设计的凸轮表如下图所示：



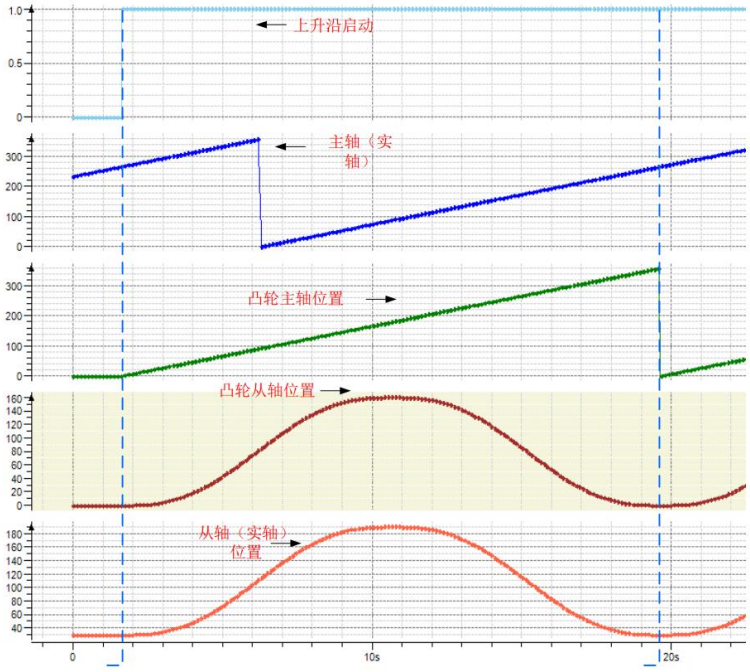
StartMode 为 0（绝对模式）

当 MC_CamTableSlect 指令 MasterAbsolute 设置为 FALSE，SlaveAbsolute 设置为 TRUE 时，此时主轴工 作在相对模式，从轴工作在绝对模式。当 Excute 上升沿 ，凸轮启动时，凸轮主轴从凸轮表“起始位置”（0） 开始，凸轮从轴按照上述的“凸轮表齿合公式”计算输出，从轴实轴指令位置等于齿合计算输出值。比如说凸轮从轴起始位置为 0，凸轮启动时从轴实轴位置为 20，则启动开始从轴实轴的位置指令为 0 产生跳变。 注意：该情况下从轴（实轴）开始位置不在凸轮从轴开始位置则会产生跳变。



当 MC_CamTableSlect 指令 MasterAbsolute 设置为 FALSE，SlaveAbsolute 设置为 FALSE 时，此时主轴工 作在相对模式，从轴工作在相对模式。当 Excute 上升沿 ，凸轮启动时，凸轮主轴从凸轮表“起始位置”（0） 开始，凸轮从轴按照上述的“凸轮表齿合公式”计算输出，从轴实轴指令位置等于齿合计算输出值（凸轮从 轴位置）+ 启动时从轴实轴位置。

比如说凸轮启动时从轴实轴位置为 20，凸轮表从轴起始位置为 0，则启动凸轮时从轴实轴位置指令为 20，后 续为 20+ 凸轮表计算值，最高峰值为 20+ 凸轮表计算最大值（此处为 180）=200。



5）错误说明

◆ 指令设置信息与 Camslect 指令设置信息不相符合。

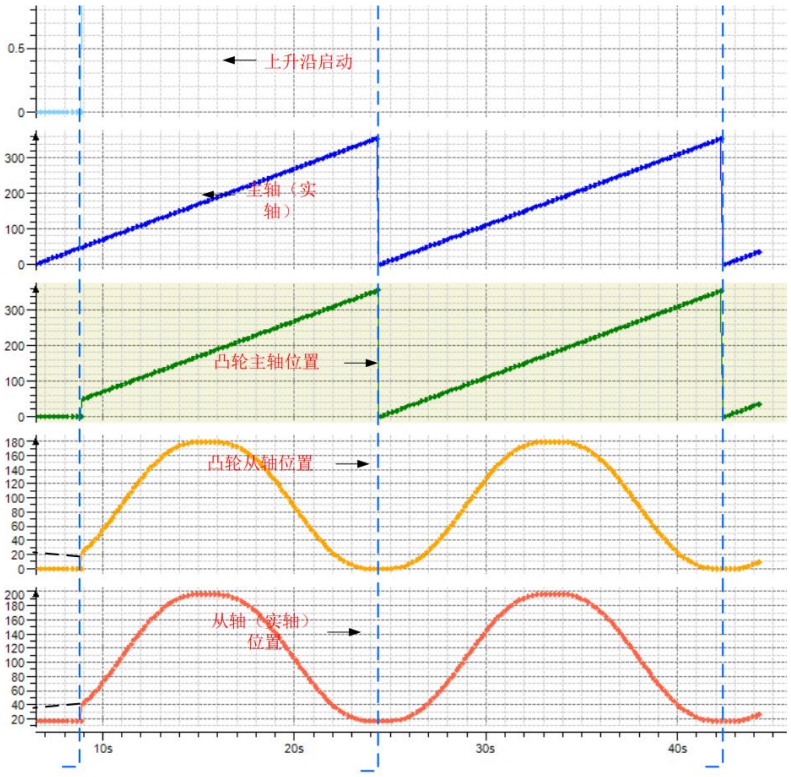
◆ 周没有使能情况。

当 MC_CamTableSlect 指令 MasterAbsolute 设置为 TRUE，SlaveAbsolute 设置为 FALSE 时，此时主轴工 作在绝对模式，从轴工作在相对模式。当 Excute 上升沿 ，凸轮启动时，凸轮主轴从当前“主轴实轴位置开始”， 从轴实轴位置指令 = 凸轮表齿合计算值（凸轮从轴位置）+ 启动时从轴位置。

注意：

1 如果该情况下主轴（实轴）开始位置不在凸轮主轴开始位置则会产生跳变。

2 主轴位置应在凸轮主轴位置范围内。

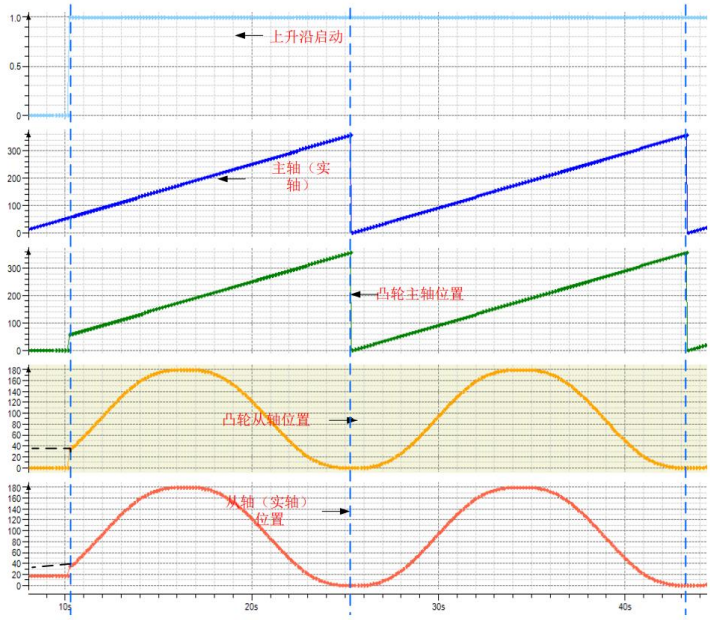


当 MC_CamTableSlect 指令 MasterAbsolute 设置为 TRUE，SlaveAbsolute 设置为 TRUE 时，此时主轴工 作在绝对模式，从轴工作在绝对模式。当 Excute 上升沿 ，凸轮启动时，凸轮主轴从当前“主轴实轴位置开始”， 从轴实轴位置指令 = 凸轮表齿合计算值（凸轮从轴位置）。

注意：

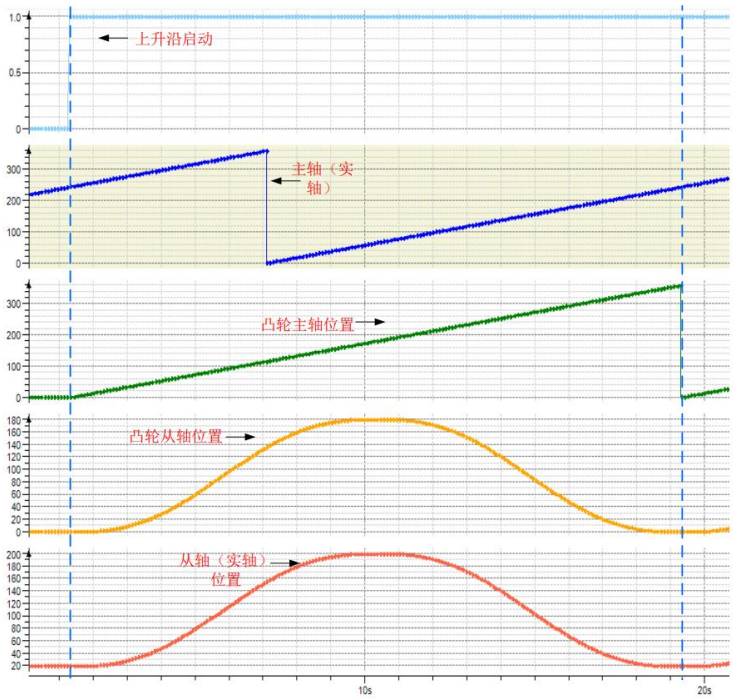
1 如果该情况下主轴（实轴）开始位置不在凸轮主轴开始位置、从轴位置不在凸轮从轴开始位置则会产生跳变。

2 主轴位置应在凸轮主轴位置范围内。



StartMode 为 1（相对模式）

当 MC_CamTableSlect 指令 MasterAbsolute 设置为 FALSE，SlaveAbsolute 设置为 TRUE 或者为 False 时， 此时主轴工作在相对对模式，从轴工作在相对模式。当 Excute 上升沿 ，凸轮启动时，凸轮主轴从 “凸轮表起 始位置开始”，从轴实轴位置指令 = 凸轮表齿合计算值 + 凸轮表齿合计算值（凸轮从轴位置）。

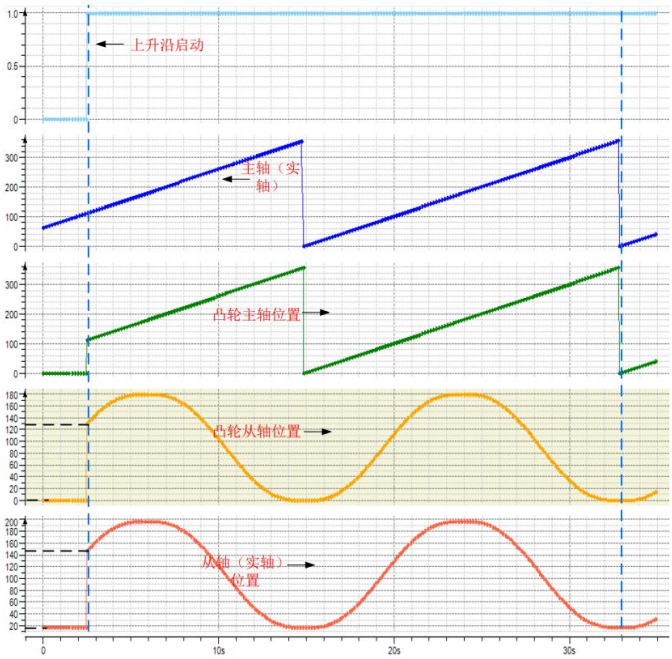


当 MC_CamTableSlect 指令 MasterAbsolute 设置为 TRUE，SlaveAbsolute 设置为 TRUE 或者为 False 时， 此时主轴工作在绝对模式，从轴工作在相对模式。当 Excute 上升沿 ， 凸轮启动时，凸轮主轴从 “主轴当前位 置开始”，从轴实轴位置指令 = 启动时从轴位置 + 凸轮表齿合计算值（凸轮从轴位置）。

注意：

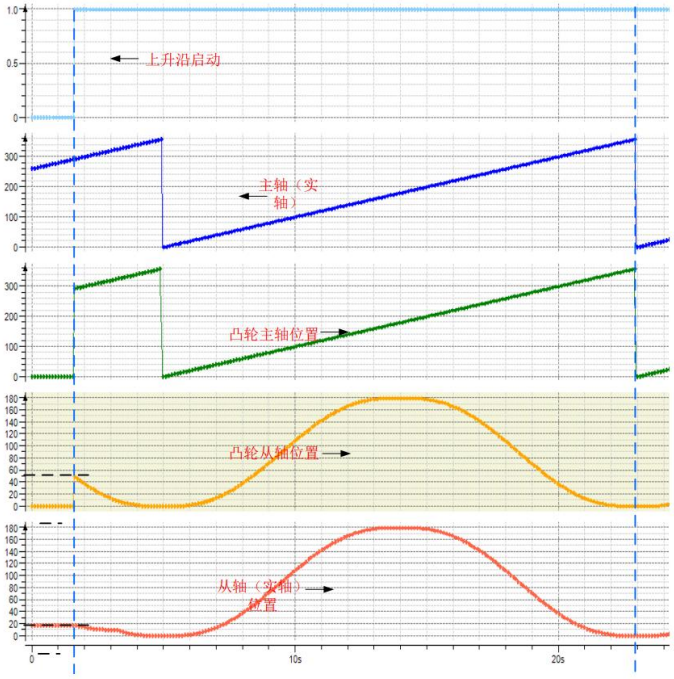
1 如果该情况下主轴（实轴）开始位置不在凸轮主轴开始位置则会产生跳变。

2 主轴位置应在凸轮主轴位置范围内

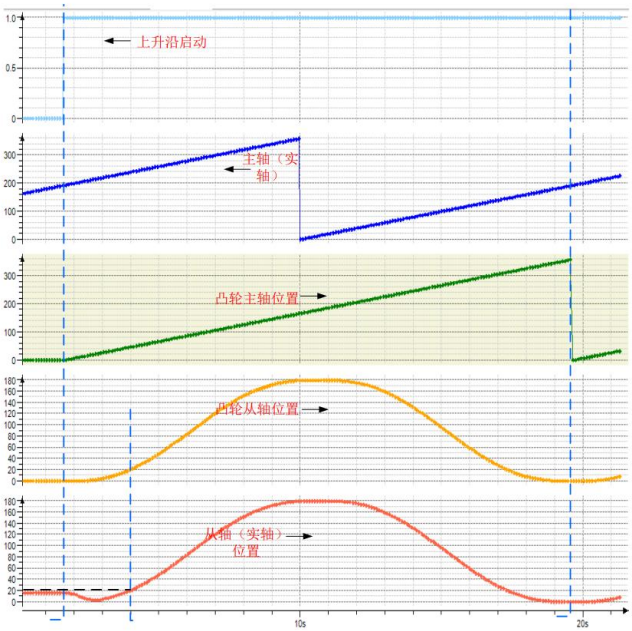


StartMode 为 2（rampin 斜坡切入模式）

当 MC_CamTableSlect 指令 MasterAbsolute 设置为 TRUE，SlaveAbsolute 设置为 TRUE 时，此时主轴工 作在绝对模式，从轴工作在绝对模式。当 Excute 上升沿 ， 凸轮启动时，凸轮主轴从 “主轴当前位置开始”， 从轴通过设定的 VelocityDiff、Acceleration、Deceleration 增加一个补偿运动避免切入时的潜在跳变。从轴实轴位置指令 = 凸轮表齿合计算值（凸轮从轴位置）+f(VelocityDiff,Acceleration,Deceleration)。

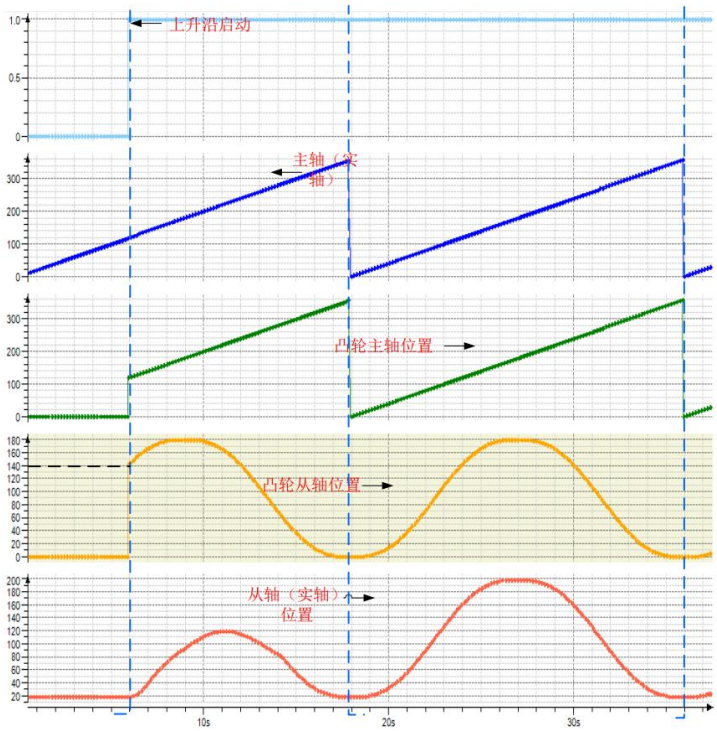


当 MC_CamTableSlect 指令 MasterAbsolute 设置为 FALSE，SlaveAbsolute 设置为 TRUE 时，此时主轴工 作在相对模式，从轴工作在绝对模式。当 Excute 上升沿 ，凸轮启动时，凸轮主轴从“ 凸轮主轴起始位置开始”， 从轴通过设定的 VelocityDiff、Acceleration、Deceleration 增加一个补偿运动避免切入时的潜在跳变。从轴实轴位置指令 = 凸轮表齿合计算值（凸轮从轴位置）+f(VelocityDiff,Acceleration,Deceleration)。

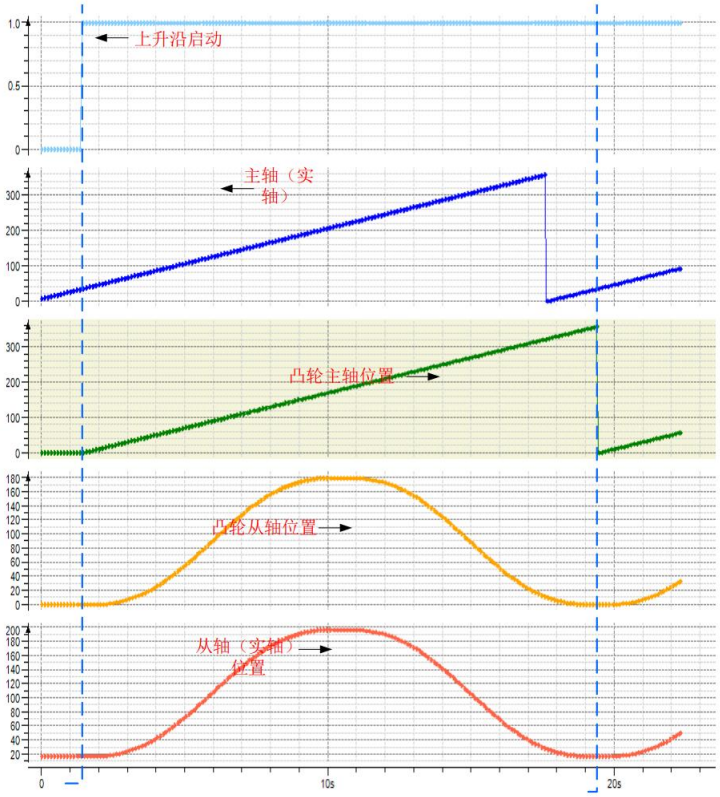


当 MC_CamTableSlect 指令 MasterAbsolute 设置为 TRUE，SlaveAbsolute 设置为 FALSE 时，此时主轴工 作在绝对模式，从轴工作在相对模式。当 Excute 上升沿 ，凸轮启动时，凸轮主轴从 “主轴当前位置开始”， 从轴通过设定的 VelocityDiff、Acceleration、Deceleration 增加一个补偿运动避免切入时的潜在跳变。从轴 实轴位置指令 = 从轴当前位置 + 凸轮表齿合计算值（凸轮从轴位置）+f(VelocityDiff,Acceleration,Decelerati on)。

注意：该方式下第一个主轴周期内凸轮曲线与设计曲线可能存在较大变化



当 MC_CamTableSlect 指令 MasterAbsolute 设置为 FALSE，SlaveAbsolute 设置为 FALSE 时，此时主轴工 作在相对模式，从轴工作在相对模式。当 Excute 上升沿 ，凸轮启动时，凸轮主轴从“ 凸轮主轴起始位置开始”， 从轴通过设定的 VelocityDiff、Acceleration、Deceleration 增加一个补偿运动避免切入时的潜在跳变。从轴实轴位置指令 = 从轴当前位置 + 凸轮表齿合计算值（凸轮从轴位置）+f(VelocityDiff,Acceleration,Decelerati on)。 注意：该方式下第一个主轴周期内凸轮曲线与设计曲线可能存在较大变化。



StartMode 为 3、4（正向斜坡切入 ramp_in_pos、反向斜坡切入 ramp_in_neg）

当从轴工作为“旋转模式”ramp_in_pos 只会沿着轴正向运动方向补偿，ramp_in_neg 只朝轴反向运动 方向补偿，当轴工作为线性模式 ramp_in_pos、ramp_in_neg、ramp_in 三种模式都是为补偿方向自动调整， 也就是说如果轴设置为线性模式工作则 ramp_in_pos、ramp_in_neg、ramp_in 三种启动模式工作情况是一 样的。

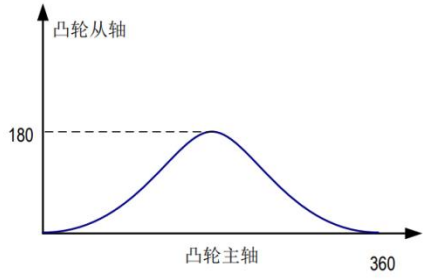
缩放比、主从轴偏移：

参照上述凸轮齿合计算公式可知：输入量 MasterOffset 和 MasterScaling 按照下面公式变换主轴位置， 并且电子凸轮会用变换后的位置 X 进行计算：

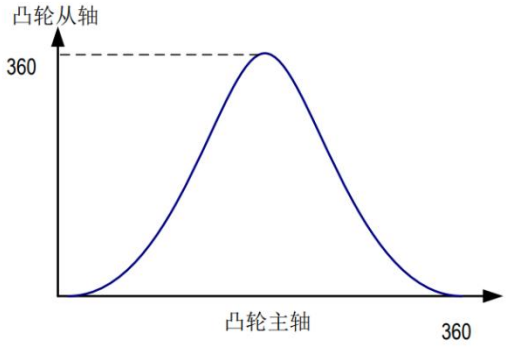
X = MasterScaling*MasterPosition + MasterOffset

因此，如果 MasterScaling 的值大于 1，电子凸轮将在较高速度下运行；如果值小于 1，电子凸轮将在较低速度运行。

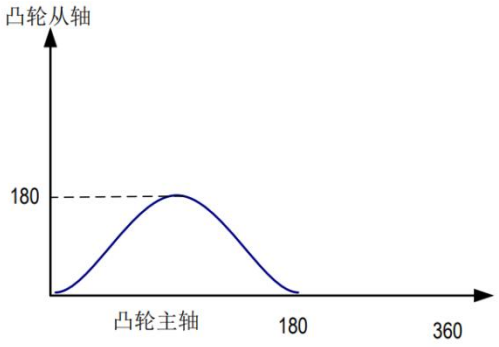
SlaveOffset 参数会使电子凸轮纵向（从轴方向）移动，SlaveScaling 参数则会在从 轴方向拉伸电子凸轮。根据下述公式说明可知，第一步电子凸轮先进行拉伸动作， 然后再进行移动： Y = SlaveScaling*CAM(X) + SlaveOffset A SlaveScaling >1 ，拉伸电子凸轮，从轴运动范围增大；同理，SlaveScaling。



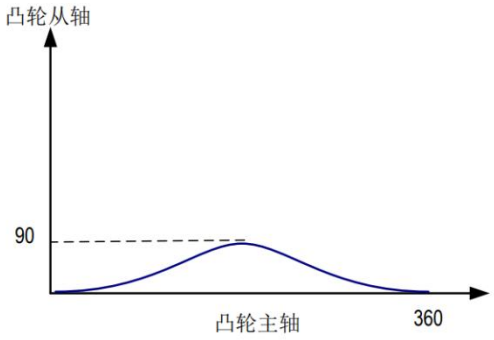
当 MasterScaling=1.0、SlaveScaling=2.0、MasterOffset=0、SlaveOffset=0，此时凸轮曲线如下图所示：



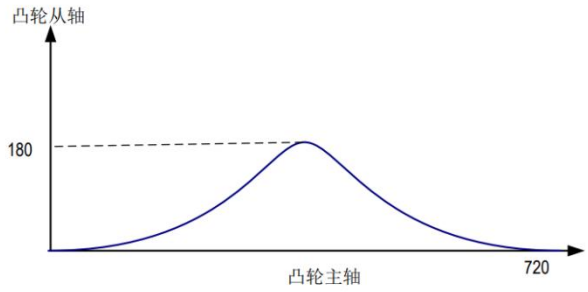
当 MasterScaling=2.0、SlaveScaling=1.0、MasterOffset=0、SlaveOffset=0，此时凸轮曲线如下图所示：



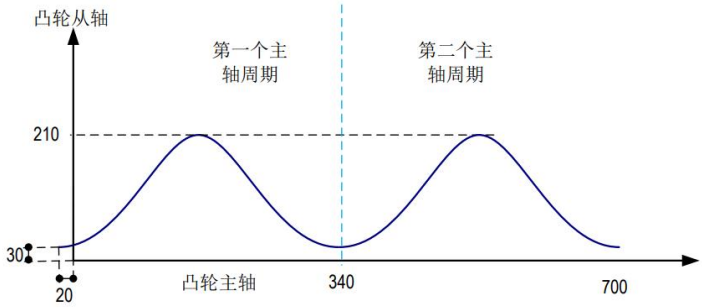
当 MasterScaling=1.0、SlaveScaling=0.5、MasterOffset=0、SlaveOffset=0，此时凸轮曲线如下图所示：



当 MasterScaling=0.5、SlaveScaling=1、MasterOffset=0、SlaveOffset=0，此时凸轮曲线如下图所示：



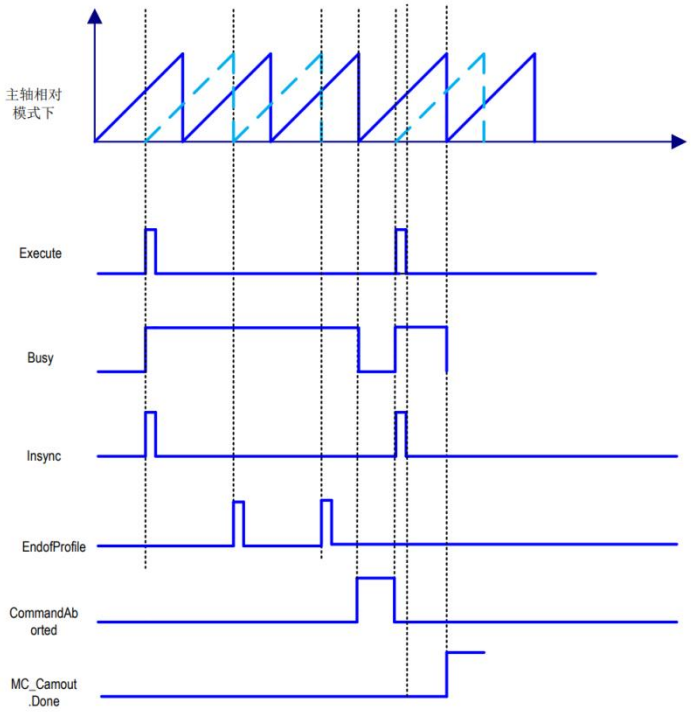
当 MasterScaling=1、SlaveScaling=1、MasterOffset=20、SlaveOffset=30，此时凸轮曲线如下图所示：



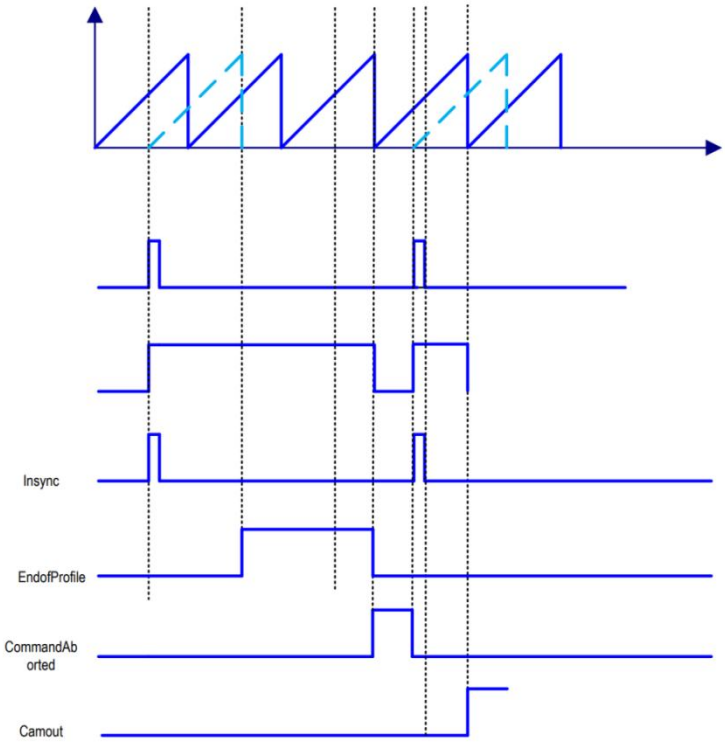
6） 时序图

周期模式（MC_CamTableSelect.Periodic 设为 TRUE）如下图：

注意：MC_Camout 指令只切断主从轴的凸轮耦合关系，如果断开时候从轴速度不为 0，则从轴不会自动减 速为 0，需配合使用 MC_STOP 指令 .



非周期模式（MC_CamTableSelect.Periodic 设为 FALSE）如下图：



◆ 电子凸轮重启：

基本上，两个电子凸轮在任何时候都可以切换，但是需要考虑一些情况：在电子凸轮编辑器中，从轴的位置 定义为电子凸轮函数的计算输出，电子凸轮函数是以在主轴范围内的一个主轴位置为

计算条件，由此可以用 下述简单的公式来进行表达说明：SlavePosition = CAM(MasterPosition) 因为主轴驱动的实际周期一般与电 子凸轮定义的主轴范围是不同的，所以为满足电子凸轮函数的正确输入，主轴位置必须被按比例调整到函数 定义域内：SlavePosition = CAM(MasterScale*MasterPosition + MasterOffset) 类似的情况，如果一个电子 凸轮在绝对值模式下启动，产生了一个向上的跳变，函数输出（也就是虚拟从轴位置）也会按比例被修正：SlavePosition = SlaveScale*CAM(MasterPosition)+ SlaveOffset 最坏情况下，这两种比例修正都必须被采 用，所以，事实上从轴位置（SlavePosition）是由更为复杂的公式计算得出的：Slaveposition =SlaveScale*CAM(MasterScale*Masterposition + MasterOffset) + SlaveOffset 在每个电子凸轮周期结束时，比例和偏移可以改变以获得更为合适的参数。遗憾的是，电子凸轮的 MC_ CamIn 模块的重新启动，将会删除它的内存并且包括比例值和偏移值。因此，所定义的电子凸轮函数会适 应于一般各种不同的从轴数值。基于这个原因，建议只有在需要处理另外一个不同的电子凸轮时，才去重启 MC_CamIn-FB。

注：电子凸轮切换请参考运动控制功能篇。

电子凸轮样例参考运动控制功能篇：

挺杆功能请参考运动控制功能篇

7） 错误说明

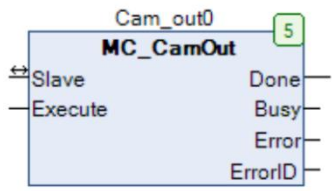
启动本指令检测到异常时，Error(错误) 变为 TRUE。

可查看 ErrorID(错误代码)的输出值，阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.2.6 MC_CamOut

断开从轴的凸轮耦合关系。注意：执行该指令后从轴会按照分离前的速度继续运行，所以需要跟 MC_Stop 等 指令配合使用。

1）指令格式

指令	名称	图形表现	ST 表现
MC_CamOut	断开凸轮耦合		MC_CamOut(Slave:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);

2）相关变量

◆ 输入输出

输入输出变量	名称	数据类型	有效范围	初始值	描述
Slave	从轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行指令	BOOL	-	-	上升沿信号执行指令

◆ 输出

输出变量	名称	数据类型	有效范围	初始值	描述
Done	完成	BOOL	TRUE,FALSE -	- FALSE	完成与主轴的凸轮耦合断开
Busy	执行中	BOOL	TRUE,FALSE	FALSE	指令执行中
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ERROR	参阅 SMC_ERROR	0	异常发生时，输出错误代码

3）功能说明

◆ 执行本指令解除从轴的凸轮耦合关系。

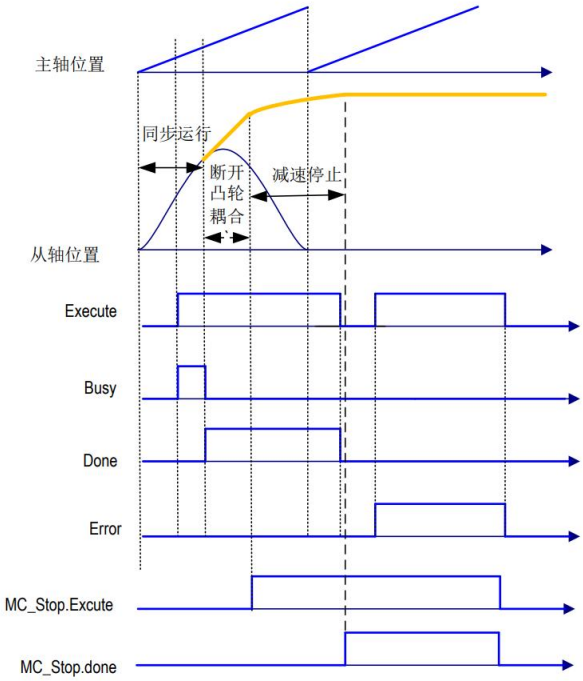
◆ Excute 上升沿时执行从轴的凸轮耦合关系断开

◆ 凸轮关系断开后，从轴并不一定是停止的。

◆ 如果从轴在执行该指令前速度不为 0，则指令 DONE 信号完成后凸轮耦合关系断开但是从轴任然按照切出去 前速度运行。

◆ 从轴没有凸轮耦合关系执行该执行，ERROR 输出。

4）时序图



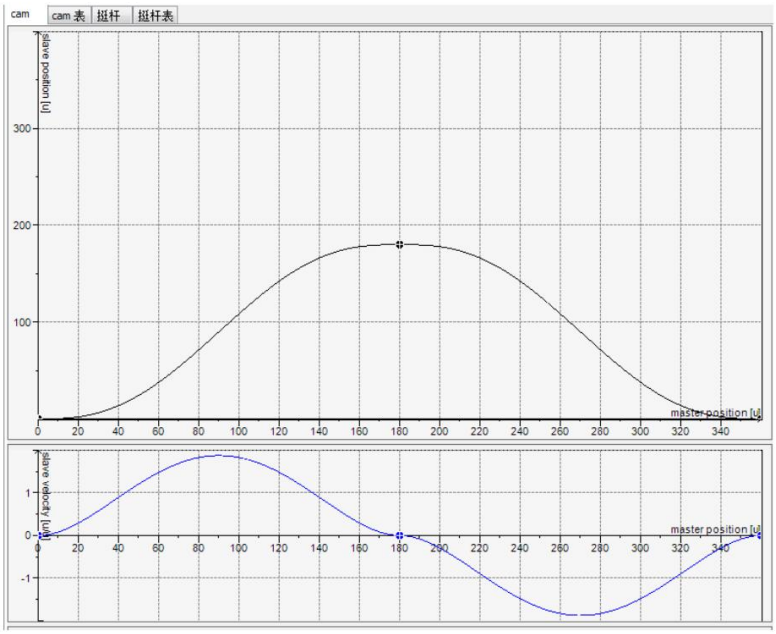
5）使用范例

本范例应用凸轮相关指令，介绍凸轮关系的建立、运行及脱离时轴的相关运动状态

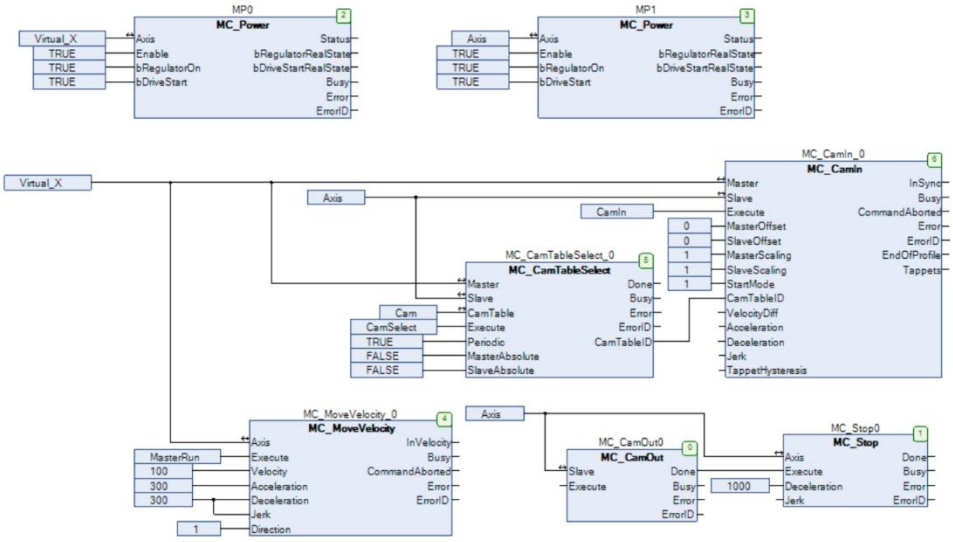
凸轮编辑器建立如下凸轮表（cam）：

cam | cam 表 | 挺杆 | 挺杆表

	X	Y	V	A	J	节类型	min(P...	max(P...	max(|V...	max(|A...
	0	0	0	0	0					
						Poly5	0	180	1.8749...	0.0320...
	180	180	0	0	0					
						Poly5	0	180	1.8749...	0.0320...
	360	0	0	0	0					



程序：



上电后主从轴自动使能，MasterRun 置为 TRUE 则主轴以 100 速度运行 CamSelect 置为 True，选择凸轮表，然后 CamIn 置为 True 启动电子凸轮。需要断开电子凸轮时将 MC_CamOut0.Execute 置为 True。

6）错误说明

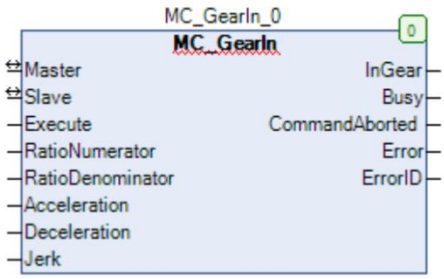
启动本指令发生异常时，Error 输出为 True。

可查看 ERRORID，参考“附录 C 错误代码说明”以了解 SMC_ERROR 错误代码。

4.2.7 MC_GearIn

设定从轴与主轴间的齿轮比，进行电子齿轮动作。

1）指令格式

指令	名称	图形表现	ST 表现
MC_GearIn	电子齿轮功能块		MC_GearIn0(Master:= , Slave:= , Execute:= , RatioNumerator:= , RatioDenominator:= , Acceleration:= , Deceleration:= , Jerk:= , InGear=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Master	主轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例
Slave	从轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行	BOOL	TRUE- FALSE	-FALSE	上升沿，开始执行指令
RatioNumerator	齿轮比分子	DINT	正数、负数 -	1-	齿轮比分子
RatioDenominator	齿轮比分母	UDINT	正数	1	齿轮比分母
Acceleration	加速度	LREAL	正数	-	指定加速度
Deceleration	减速度	LREAL	正数	-	指定减速度
Jerk	越度	LREAL	正数或 0	-	加加速度

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
InGear	齿轮比到达	BOOL	TRUE- FALSE	- FALSE	True, 从轴达到目标速度
Busy	执行中	BOOL	- TRUE,FALSE	FALSE	True，正在执行指令
CommandAborted	中断	BOOL	TRUE,FALSE	FALSE	True，被其他控制指令中断
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ERROR	参阅 SMC_ ERROR	0	异常发生时，输出错误代码

3）功能说明

◆ 通过 Slave（从轴）指定动作对象轴，指定 RatioNumerator（齿轮比分子）、RatioDenominator（齿轮比分母）、ReferenceType(位置类型选择) 、Acceleration(加速度) 、Deceleration(减速度)，进行齿轮动作。

◆ 可对 Master(主轴) 指定指令位置和反馈位置、最新的指令位置。

◆ Execute 上升沿开始电子齿轮动作

◆ 开始动作后，Slave（从轴）以 Master（主轴）速度乘以齿轮比得到的速度为目标速度，进行加减速动作。

◆ 执行电子齿轮后要解除耦合必须通过 GearOut 指令。

◆ 该指令为速度电子齿轮功能，加速过程中造成的同步距离丢失不会自动补偿。

◆ 指令执行中 Busy 信号为 TRUE 时，如果从轴目标速度没有达到此时 Execute 新的上升沿不会影响它。

◆ 指令执行中 Busy 信号为 TRUE 时，如果从轴目标速度达到此时 Execute 新的上升沿不会影响它。

◆ 到达目标速度，InGear 为 TRUE，此后从轴移动量 = 主轴移动量 * RatioNumerator/RatioDenominator。

◆ 如果主轴速度实时变化，请注意慎重使用该指令。

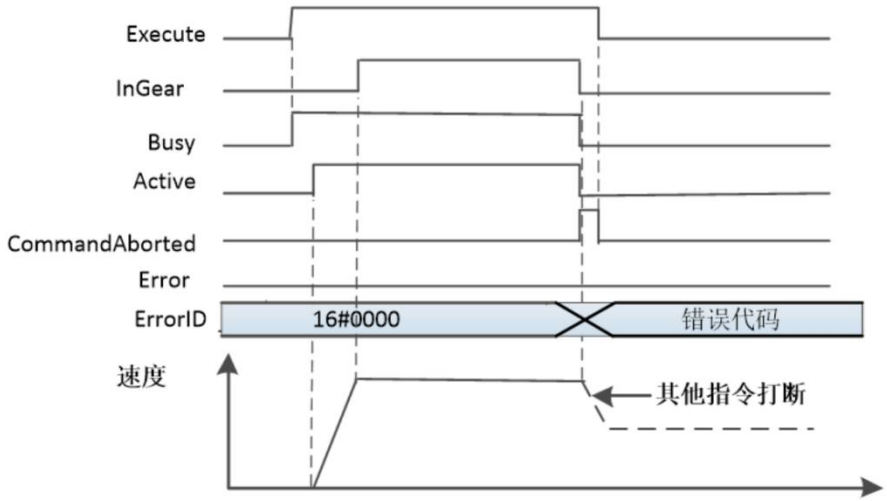
4）注意事项

◆ 执行指令过程中请不要使用 MC_SetPosition 指令以免引起电机急速运转产生事故。

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

5） 时序图

在启动 Execute（启动）的同时，Busy（执行中）变为 TRUE。在下一个周期 Active（控制中）变为 TRUE。 达到目标速度时，InGear（达到齿轮比）变为 TRUE。 利用其它指令中止本指令时，CommandAborted（执行中断）变为 TRUE，Busy（执行中）、Active（控制 中）、InGear（达到齿轮比）变为 FALSE。 要中途结束电子齿轮动作，可使用 MC_GearOut（齿轮动作解除）指令、或 MC_Stop（强制停止）指令。



◆ 其他指令执行中的本指令启动

对当前正在执行的指令启动本指令，可切换或缓存到本指令。本指令的多个实例启动时的动作由 BufferMode（ 缓存模式选择 ） 指定。

缓存模式选择	说明
中断 （Aborting）	立即中止当前正在执行的指令，切换为本指令。 轴的动作方向因指令切换而反转时，根据轴参数中的“反转时动作”进行反转。

缓存模式选择	说明
等待 （Buffered）	功能块在最后一个命令运动终止后立即启动。这里不进行混合。 当达到结束条件（Done、InVelocity、InEndVelocity、InGear、InSync、EndOfProfile 等）时，新运动以先前运动的速度开始。 如果之前的运动是 MC_MoveAbsolute 或 MC_MoveRelative，那么新的运动会在静止状态下开始。

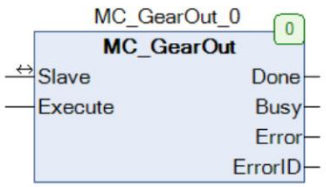
◆ 本指令执行中的其他指令启动

可对从轴以中断方式执行指令的多个实例启动。 此时，中止齿轮动作，开始多个实例启动的动作。无法以非中断方式多个实例启动指令。

4.2.8 MC_GearOut

终止执行中的 MC_GearIn,MC_GearInPos 指令。

1）指令格式

指令	名称	图形表现	ST 表现
MC_GearOut	电子齿轮耦合断开		MC_GearOut0(Slave:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Slave	从轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行	BOOL	TRUE- FALSE	FALSE	上升沿，开始执行指令

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	完成	BOOL	TRUE- FALSE	FALSE	True, 从轴与主轴电子齿轮耦合断开
Busy	执行中	BOOL	- TRUE,FALSE	FALSE	True，指令正在执行中
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ERROR	参阅 SMC_ERROR	0	异常发生时，输出错误代码

3）功能说明

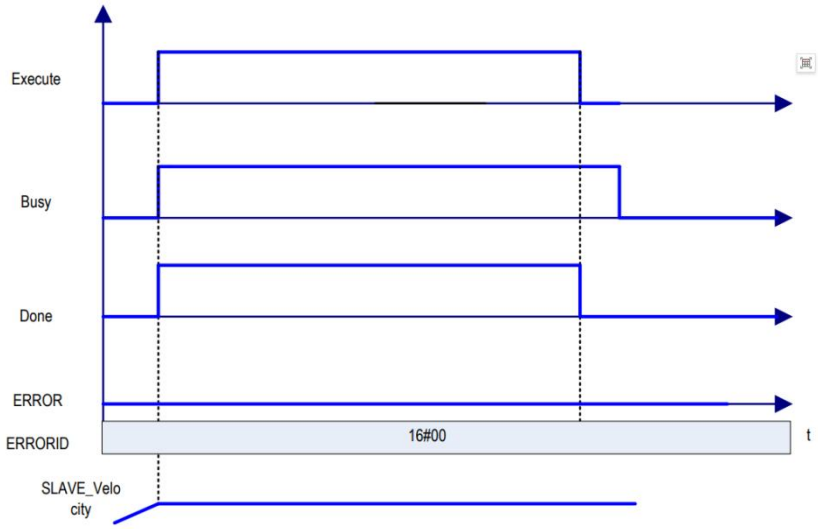
◆ Execute 上升沿 ， 执行切出电子齿轮动作。

◆ Excute 为 TRUE，ERROR 为 FALSE 则 Busy 输出为 TRUE，Done 输出为 TRUE。

◆ 切出电子齿轮动完成后此时从轴的速度为切出前的速度,所以需配合以 MC_Stop 指令停止从轴。

◆ Execute 下降沿则，Done 为 FALSE.

◆ MC_Stop 指令执行复位 Busy 信号



4）错误说明

相关参数设置有错，会导致指令报警。

轴没有使能会导致指令报警。

【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.2.9 MC_Phasint

指定主从轴之间的相位偏差。

1）指令格式

指令	名称	图形表现	ST 表现
MC_ Phasing	主从轴相位偏移	MC_Phasing_0 MC_Phasing Master, Slave, Execute, PhaseShift, Velocity, Acceleration, Deceleration, Jerk Done, Busy, CommandAborted, Error, ErrorID	MC_Phasing0(Master:= , Slave:= , Execute:= , PhaseShift:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Master	主轴	AXIS_REF_SM3	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例
Slave	从轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◆ 输入相关变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	指令执行	BOOL	TRUE- FALSE	FALSE	上升沿，开始执行指令
PhaseShift	主从轴相位偏差值	LREAL	-	0	主从轴相位偏差值,正数代表从轴滞后。
Velocity	速度	LREAL	-	0	执行相位偏移时最大速度值
Acceleration	加速度	LREAL	-	0	执行相位偏移时最大加速度值

输入变量	名称	数据类型	有效范围	初始值	描述
Deceleration	减速度	LREAL	-	0	执行相位偏移时最大减速度值
Jerk	速度二次导数	LREAL	-	0	执行相位偏移时最大 Jerk 值

◆ 输出相关变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	完成	BOOL	TRUE- FALSE	FALSE	True, 如果相位偏移完成
Busy	指令处理中	BOOL	TRUE,FALSE	FALSE	True，指令正在处理中
CommandAborted	指令中断	BOOL	TRUE,FALSE	FALSE	被其它控制指令中断
Error	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ERROR	参阅 SMC_ERROR	0	异常发生时，输出错误代码

3）功能说明

◆ Execute 上升沿执行相位偏移，从轴自动计算一条平滑曲线，完成从轴对主轴的相位偏移，主从轴相位差为 输入信号的 PhaseShift 值，正值为从轴滞后于主轴。

◆ 完成偏移后 Done 信号输出为 True。

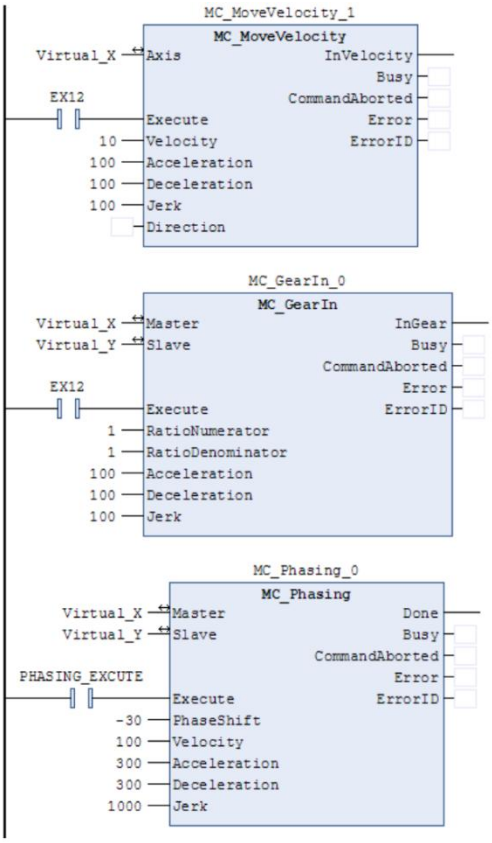
◆ 根据设定的 PhaseShift、Velocity、Acceleration、Deceleration 对主从轴相位差进行补偿。

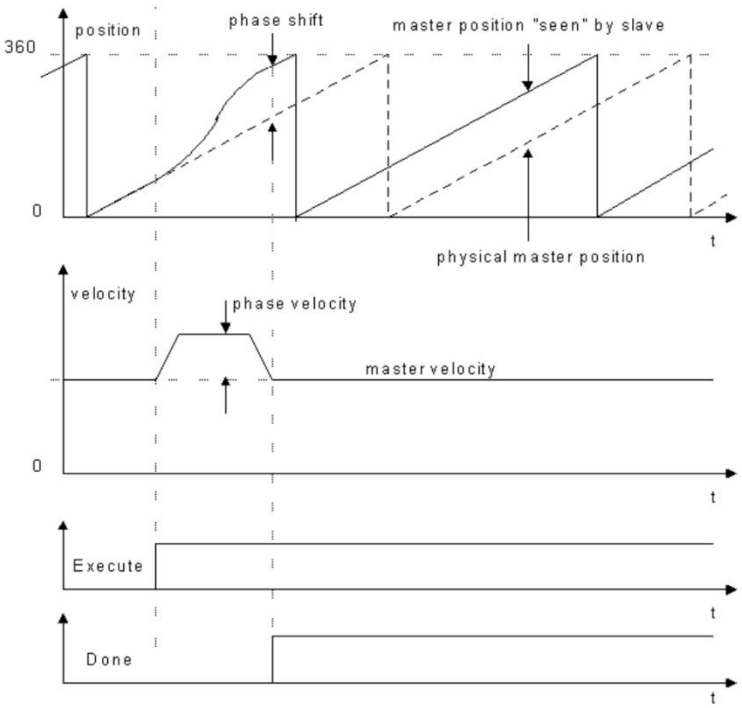
◆ 主从轴相位差达到 PhaseShift 时，Done 信号输出。

◆ 执行指令时主轴指令位置与反馈位置不变，从轴进行调整，完成后，从轴跟主轴之间相位差为 PhaseShift。

◆ 该指令最终结果为轴给定值之间的相位偏移，所以实轴的实际反馈值与最终的偏移可能不一致。

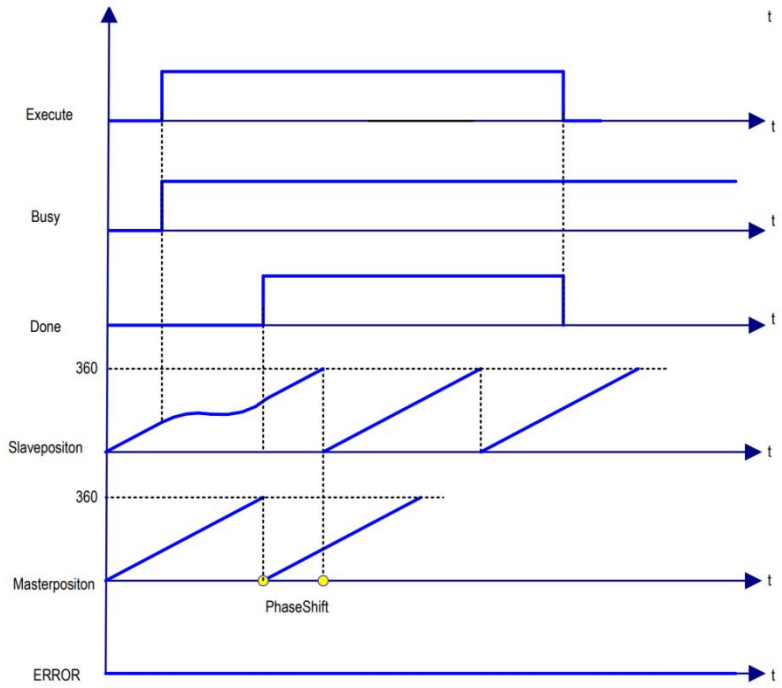
◆ 该指令跟 MC_GearIn 指令配合使用，如下：主轴为 Virtual_x，从轴为 Virtual_y,EX12 上升沿执行主轴速度 控制以及主从轴电子齿轮动作，然后执行相位偏移。此外可以跟电子凸轮配合使用，此时从轴作为“电子凸轮主轴”以达到电子凸轮主轴相位偏移的效果。


MC_MoveVelocity_1
MC_MoveVelocity
Virtual_X
Axis
InVelocity
Busy
EX12
CommandAborted
Execute
Error
10
Velocity
ErrorID
100
Acceleration
100
Deceleration
100
Jerk
Direction
MC_GearIn_0
MC_GearIn
Virtual_X
Master
InGear
Virtual_Y
Slave
Busy
CommandAborted
EX12
Error
Execute
ErrorID
1
RatioNumerator
1
RatioDenominator
100
Acceleration
100
Deceleration
100
Jerk
MC_Phasing_0
MC_Phasing
Virtual_X
Master
Done
Virtual_Y
Slave
Busy
CommandAborted
PHASING_EXCUTE
Error
Execute
ErrorID
-30
PhaseShift
100
Velocity
300
Acceleration
300
Deceleration
1000
Jerk


position
phase shift
master position "seen" by slave
360
0
t
physical master position
velocity
phase velocity
master velocity
0
t
Execute
t
Done
t

4）时序图

以主从轴都按 360 周期运动，Execute 信号上升沿执行调整，调整完成后从轴与主轴之间相位偏差为 PhaseShift 设定的值。



5）错误说明

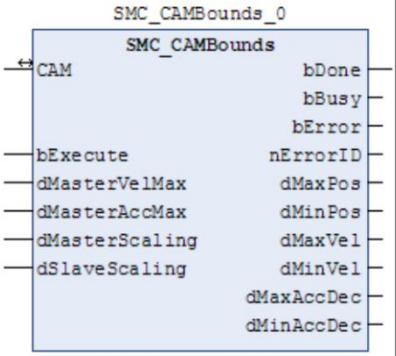
◆ 启动指令时，Error 输出为 TRUE，则有错误发生。

◆ 查看 ErrorID，查看帮助中 SMC_ERROR 确定报警信息，请阅读“附录 C 错误代码说明”了解相关错误代码说明。

4.2.10 SMC_CAMBounds

当从轴与主轴凸轮耦合后可以通过该功能块来计算出从轴最大位置，速度，加速度。主轴在输入最大速度、加减速限制下运动。该指令在设计凸轮表时候可以检查曲线是否正确，前提 知道主轴最大加减速，速度等。

1）指令格式

指令	名称	图形表现	ST 表现
SMC_CAMBounds	凸轮上下限		SMC_CAMBounds_0(CAM:= , bExecute:= , dMasterVelMax:= , dMasterAccMax:= , dMasterScaling:= , dSlaveScaling:= , bDone=> , bBusy=> , bError=> , nErrorID=> , dMaxPos=> , dMinPos=> , dMaxVel=> , dMinVel=> , dMaxAccDec=> , dMinAccDec=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
CAM	凸轮	MC_CAM_REF	-	-	映射到凸轮，即 MC_CAM_REF 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
bExecute	指令执行	BOOL	TRUE- FALSE	FALSE	上升沿，开始执行指令
dMasterVelMax	最大速度	LREAL	-	1	绝对模式下主轴最大速度。
dMasterAccMax	最大加速度	LREAL	-	0	绝对模式下主轴最大加速度
dMasterScaling	标尺因子	LREAL	-	1	主轴凸轮应用中标尺因子
dSlaveScaling	标尺因子	LREAL	-	1	从轴凸轮应用中标尺因子

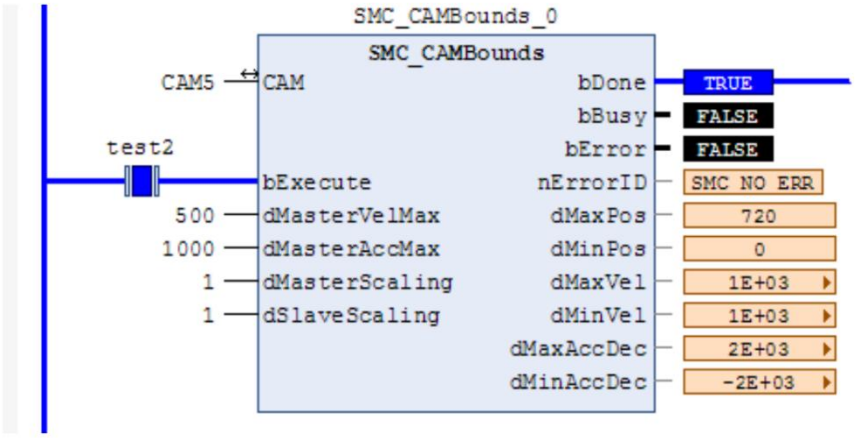
◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
bDone	完成	BOOL	TRUE- FALSE	FALSE	True, 如果计算完成
bBusy	指令处理中	BOOL	TRUE,FALSE	FALSE	True，指令正在处理中
bError	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
nErrorID	错误代码	SMC_ERROR	参阅 SMC_ERROR	0	异常发生时，输出错误代码
dMaxPos	最大位置	LREAL	-	0	根据凸轮表计算出从轴最大位置
dMinPos	最小位置	LREAL	-	0	根据凸轮表计算出从轴最小位置
dMaxVel	最大速度	LREAL	-	0	计算出最大速度
dMinVel	最小速度	LREAL	-	0	计算出最小速度
dMaxAccDec	最大加速度	LREAL	-	0	计算出最大加速度
dMinAccDec	最小加速度	LREAL	-	0	计算出最小加速度

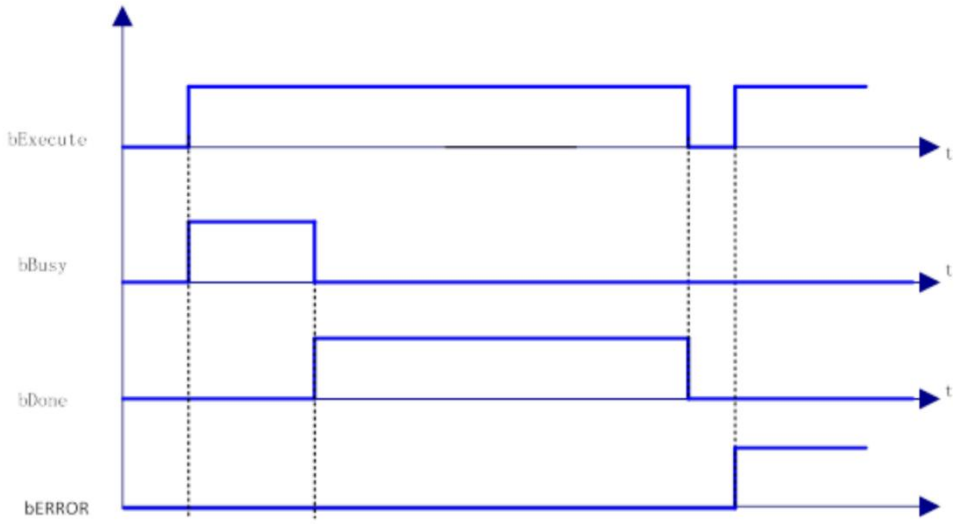
3） 功能说明

◆ bExecute 上升沿来综合输入变量的“dMasterVelMax”，“dMasterAccMax”，“dMasterScaling”，“dSlaveScaling”值与凸轮表数据计算出从轴“最大位置”最小位置等值。例如：主轴周期 360，凸轮表为 一条斜率为 2 的直线，计算出的结果如下图所示： 主轴以

绝对模式运行或者主轴设置为周期模式，模值设置为主轴周期都可使用该指令计算。 凸轮表为 XYVA（多项式模式时有效），一维数组、二维数组等无效



4）时序图



5）错误说明

凸轮表格式不是多项式模式。

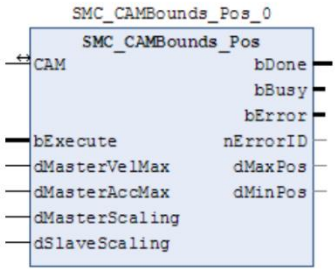
凸轮表 MC_CAM_REF 设定值与实际凸轮表不匹配。

【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.2.11 SMC_CAMBounds_Pos

当从轴与主轴凸轮耦合后可以通过该功能块来计算出从轴最大位置，与最小位置。该功能块与 SMC_ CAMBounds 相比少了最大加速度等计算 ， 其他功都一致。

1）指令格式

指令	名称	图形表现	ST 表现
SMC_ CAMBounds_Pos	凸轮位置上下限		SMC_CAMBounds_Pos0(CAM:= , bExecute:= , dMasterVelMax:= , dMasterAccMax:= , dMasterScaling:= , dSlaveScaling:= , bDone=> , bBusy=> , bError=> , nErrorID=> , dMaxPos=> , dMinPos=>);

2）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
CAM	凸轮	MC_CAM_REF	-	-	映射到凸轮，即 MC_CAM_REF 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
bExecute	指令执行	BOOL	TRUE- FALSE	FALSE	上升沿，开始执行指令
dMasterVelMax	最大速度	LREAL	-	1	绝对模式下主轴最大速度。
dMasterAccMax	最大加速度	LREAL	-	0	绝对模式下主轴最大加速度
dMasterScaling	标尺因子	LREAL	-	1	主轴凸轮应用中标尺因子
dSlaveScaling	标尺因子	LREAL	-	1	从轴凸轮应用中标尺因子

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
bDone	完成	BOOL	TRUE- FALSE	FALSE	True, 如果计算完成
bBusy	指令处理中	BOOL	TRUE,FALSE	FALSE	True，指令正在处理中
bError	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
nErrorID	错误代码	SMC_ERROR	参阅 SMC_ERROR	0	异常发生时，输出错误代码
dMaxPos	最大位置	LREAL	-	0	根据凸轮表计算出从轴最大位置
dMinPos	最小位置	LREAL	-	0	根据凸轮表计算出从轴最小位置

3）功能说明

◆ bExecute 上升沿来综合输入变量的“dMasterVelMax”，“dMasterAccMax”，“dMasterScaling”，“dSlaveScaling”值与凸轮表数据计算出从轴“最大位置”最小位置。

◆ 主轴以绝对模式运行或者主轴设置为周期模式，模值设置为主轴周期都可使用该指令计算。

◆ 凸轮表为 XYVA（多项式模式时有效），一维数组、二维数组等无效

4）错误说明

凸轮表格式不是多项式模式；凸轮表 MC_CAM_REF 设定值与实际凸轮表不匹配。

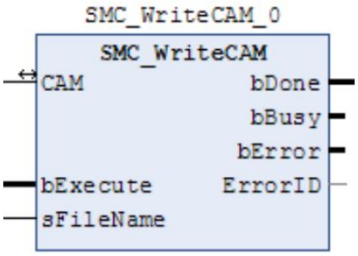
【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

4.2.12 SMC_WriteCAM

程序运行时用于将编辑的凸轮表存储为一个文件。使之可以被 MC_CamIn 等指令使用。生成的文件包含内容参考“Cam Format”。

该指令可以用来跟 SMC_ReadCAM 配合使用。

SMC_ReadCAM 配合使用。

指令	名称	图形表现	ST 表现
SMC_WriteCAM	凸轮上下限		

1）相关变量

◆ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
CAM	凸轮	MC_CAM_REF	-	-	映射到凸轮，即 MC_CAM_REF 的一个实例

◆ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
bExecute	指令执行	BOOL	TRUE-FALSE	FALSE	上升沿，开始执行指令
sFileName	文件名	STRING	-	‘’	包含凸轮描述的以 ASCII 格式定义的文件名，可以通过帮助里面”Cam Format”来查看具体描述。

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
bDone	完成	BOOL	TRUE- FALSE	FALSE	True, 如果凸轮写进文件完成
bBusy	指令处理中	BOOL	TRUE,FALSE	FALSE	True，指令执行没有完成
bError	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
nErrorID	错误代码	SMC_ERROR	参阅 SMC_ERROR	0	异常发生时，输出错误代码

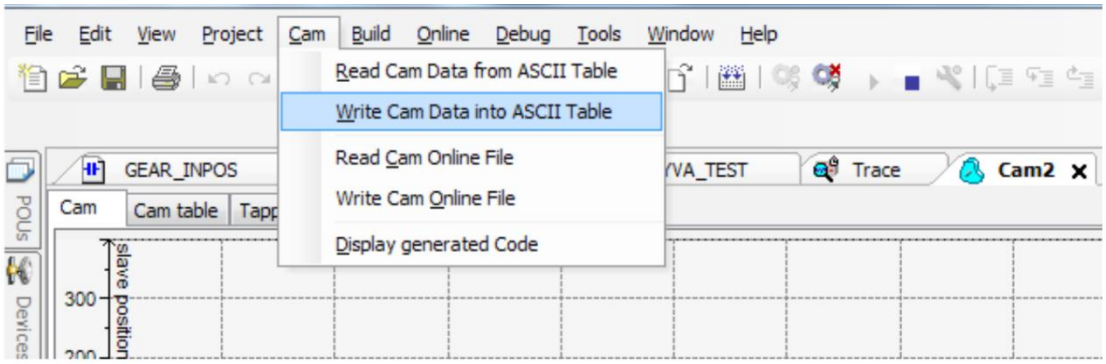
2）功能说明

◆ bExecute 上升沿，指令执行 – 将“CAM”连接的凸轮信息存储到文件名为“sFileName”连接的文件中。

◆ 存储成功并完成 bDone 信号输出为 true。

◆ 存储的凸轮表信息受硬件内存限制。

◆ 注意：该功能在程序运行中的执行，凸轮表信息也可以手动存储到离线信息



3）错误说明

◆ 该指令只能完成 XYVA 多项式模式的凸轮表，一维，二维等会造成错误输出

◆ sFileName 连接的文件名不存在或者信息错误。

【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

五、版本修订说明

版本号	修订章节	修订内容	修订日期	修订人
V1.0		初版	2024/4	胡坤

附录　错误代码说明

SMC_ERROR：记录了运动控制功能块返回的错误序号。

错误代码	产生源	变量名称	错误原因描述
0	所有	SMC_NO_ERROR	没有错误
1	驱动器接口	SMC_DI_GENERAL_COMMUNICATION_ERROR	通讯错误（例如 Sercos 环断裂）
2	驱动器接口	SMC_DI_AXIS_ERROR	轴错误
3	驱动器接口	SMC_DI_FIELDBUS_LOST_SYNCRONICITY	总线 DC 同步丢失
10	驱动器接口	SMC_DI_SWLIMITS_EXCEEDED	软限位被激活， bSWLimitEnable 使能后， 轴的当前位置不在 fSWLimitPositive 和 fSWLimitNegative 范围
11	驱动器接口	SMC_DI_HWLIMITS_EXCEEDED	硬件限位开关被激活
13	驱动器接口	SMC_DI_HALT_OR_QUICKSTOP_NOT_SUPPORTED	驱动器状态停止或者不支持快速停止
14	驱动器接口	SMC_DI_VOLTAGE_DISABLED	驱动器没有使能
15	驱动器接口	SMC_DI_IRREGULAR_ACTPOSITION	驱动器当前给予的位置格式不正确。检查通讯。
16	驱动器接口	SMC_DI_POSITIONLAGERROR	位置滞后错误。在设置和当前位置超过限制值
17	驱动器接口	SMC_DI_HOMING_ERROR	驱动器回零错误
20	所有运动控制创建的模块	SMC_REGULATOR_OR_START_NOT_SET	控制器没有使能或者抱闸没有打开
21	轴在错误的控制模式下	SMC_WRONG_CONTROLLER_MODE	轴不是一个正确的控制方式
30	驱动器接口	SMC_FB_WASNT_CALLED_DURING_MOTION	运动控制创建的模块在运动结束之前没有被调用
31	所有模块	SMC_AXIS_IS_NO_AXIS_REF	给出的 AXIS_REF 变量不是 AXIS_REF 类型
32	轴位于错误控制模式下	SMC_AXIS_REF_CHANGED_DURING_OPERATION	AXIS_REF- 变量的返回值在模块激活前被处理
33	驱动器接口	SMC_FB_ACTIVE_AXIS_DIABLED	轴在移动时没有被激活 (MC_Power. bRegulatorOn)
34	所有运动控制创建的模块	SMC_AXIS_NOT_READY_FOR_MOTION	在当前状态下轴不能处理当前命令
35	所有运动控制创建的模块	SMC_AXIS_ERROR_DURING_MOTION	轴在运动过程中发生错误

40	虚拟驱动器	SMC_VD_MAX_VELOCITY_EXCEEDED	达到最大速度 (fMaxVelocity)
41	虚拟驱动器	SMC_VD_MAX_ACCELERATION_EXCEEDED	达到最大加速度 (fMaxAcceleration)
42	虚拟驱动器	SMC_VD_MAX_DECELERATION_EXCEEDED	达到最大减速度 (fMaxDeceleration)
50	SMC_Homing	SMC_3SH_INVALID_VELACC_VALUES	无效的速度或者加速度值
51	SMC_Homing	SMC_3SH_MODE_NEEDS_HWLIMIT	模块需要使用结束限位开关（安全用途）
70	SMC_SetControllerMode	SMC_SCM_NOT_SUPPORTED	模式不支持
71	SMC_SetControllerMode	SMC_SCM_AXIS_IN_WRONG_STATE	在当前模式下使用的控制模式不支持
75	SMC_SetTorque	SMC_ST_WRONG_CONTROLLER_MODE	轴不是一个正确的控制模式，需要在转矩模式下使能此功能块
80	SMC_ResetAxisGroup	SMC_RAG_ERROR_DURING_STARTUP	在轴组启动时发生错误
90	SMC_ ChangeGearingRatio	SMC_CGR_ZERO_VALUES	不正确的变量
91	SMC_ ChangeGearingRatio	SMC_CGR_DRIVE_POWERED	驱动器控制模式下不能更改传动比
92	SMC_ ChangeGearingRatio	SMC_CGR_INVALID_POSPERIOD	不合适的位置周期 (<=0)
110	MC_Power	SMC_P_FTASKCYCLE_EMPTY	轴在扫描周期内不包含任何信息 (fTaskCycle = 0)
120	MC_Reset	SMC_R_NO_ERROR_TO_RESET	轴没有错误复位
121	MC_Reset	SMC_R_DRIVE_DOESNT_ANSWER	轴没有执行错误复位
122	MC_Reset	SMC_R_ERROR_NOT_RESETTABLE	错误不能被复位
123	MC_Reset	SMC_R_DRIVE_DOESNT_ANSWER_IN_TIME	与轴之间的通讯没有回应

错误代码	产生源	变量名称	错误原因描述
130	MC_ReadParameter, MC_ ReadBoolParameter	SMC_RP_PARAM_UNKNOWN	参数序号位置
131	MC_ReadParameter, MC_ ReadBoolParameter	SMC_RP_REQUESTING_ERROR	在将参数传送到驱动器过程中发生错误。参阅功能块实例 ReadDriveParameter 的错误 (SM_ DriveBasic.lib)
140	MC_WriteParameter, MC_ WriteBoolParameter	SMC_WP_PARAM_INVALID	参数序号位置或者不允许进行写操作
141	MC_WriteParameter, MC_ WriteBoolParameter	SMC_WP_SENDING_ERROR	参阅模块实例 WriteDriveParameter 的错误 (Drive_Basic.lib)
170	MC_Home	SMC_H_AXIS_WASNT_STANDSTILL	轴不是标准状态
171	MC_Home	SMC_H_AXIS_DIDNT_START_HOMING	在执行回零时发生错误
172	MC_Home	SMC_H_AXIS_DIDNT_ANSWER	通讯错误
173	MC_Home	SMC_H_ERROR_WHEN_STOPPING	执行回零错误停止。查阅是否设置减速度。
180	MC_Stop	SMC_MS_UNKNOWN_STOPPING_ERROR	停止时发生完未知错误
181	MC_Stop	SMC_MS_INVALID_ACCDEC_VALUES	不合适的速度或者加速度值
182	MC_Stop	SMC_MS_DIRECTION_NOT_APPLICABLE	Direction=shortest 不可用
183	MC_Stop	SMC_MS_AXIS_IN_ERRORSTOP	轴位于错误停止状态。停止不能被处理。
184	MC_Stop	SMC_BLOCKING_MC_STOP_WASNT_CALLED	一个 MC_Stop 的实例，锁定轴 (Execute=TRUE)，不能进行调用。请调用 MC_ Stop(Execute=FALSE).
201	MC_MoveAbsolute	SMC_MA_INVALID_VELACC_VALUES	不合适的速度或者加速度值
202	MC_MoveAbsolute	SMC_MA_INVALID_DIRECTION	方向错误
226	MC_MoveRelative	SMC_MR_INVALID_VELACC_VALUES	不合适的速度或者加速度值

227	MC_MoveRelative	SMC_MR_INVALID_DIRECTION	方向错误
251	MC_MoveAdditive	SMC_MAD_INVALID_VELACC_VALUES	不合适的速度或者加速度值
252	MC_MoveAdditive	SMC_MAD_INVALID_DIRECTION	方向错误
276	MC_MoveSuperImposed	SMC_MSI_INVALID_VELACC_VALUES	不合适的速度或者加速度值
277	MC_MoveSuperImposed	SMC_MSI_INVALID_DIRECTION	方向错误
301	MC_MoveVelocity	SMC_MV_INVALID_ACCDEC_VALUES	不合适的速度或者加速度值
302	MC_MoveVelocity	SMC_MV_DIRECTION_NOT_APPLICABLE	Direction=shortest/fastest 不支持
325	MC_PositionProfile	SMC_PP_ARRAYSIZE	错误排列尺寸
326	MC_PositionProfile	SMC_PP_STEP0MS	步时间 = t#0s
350	MC_VelocityProfile	SMC_VP_ARRAYSIZE	错误排列尺寸
351	MC_VelocityProfile	SMC_VP_STEP0MS	步时间 = t#0s
375	MC_AccelerationProfile	SMC_AP_ARRAYSIZE	错误排列尺寸
376	MC_AccelerationProfile	SMC_AP_STEP0MS	步时间 = t#0s
400	MC_TouchProbe	SMC_TP_TRIGGEROCCUPIED	触发条件已经被激活
401	MC_TouchProbe	SMC_TP_COULDNT_SET_WINDOW	驱动器接口不支持窗口功能
402	MC_TouchProbe	SMC_TP_COMM_ERROR	通讯错误
410	MC_AbortTrigger	SMC_AT_TRIGGERNOTOCCUPIED	触发条件已经被终止
426	SMC_ MoveContinuousRelative	SMC_MCR_INVALID_VELACC_VALUES	不合适的速度或者加速度值
427	SMC_ MoveContinuousRelative	SMC_MCR_INVALID_DIRECTION	方向错误
451	SMC_ MoveContinuousAbsolute	SMC_MCA_INVALID_VELACC_VALUES	不合适的速度或者加速度值
452	SMC_ MoveContinuousAbsolute	SMC_MCA_INVALID_DIRECTION	方向错误
453	SMC_ MoveContinuousAbsolute	SMC_MCA_DIRECTION_NOT_APPLICABLE	Direction= fastest 不可用
600	SMC_CamRegister	SMC_CR_NO_TAPPETS_IN_CAM	CAM 中不包含任何挺杆
601	SMC_CamRegister	SMC_CR_TOO_MANY_TAPPETS	挺杆组 ID 达到 MAX_NUM_TAPPETS

错误代码	产生源	变量名称	错误原因描述
602	SMC_CamRegister	SMC_CR_MORE_THAN_32_ACCESSES	在一个 CAM_REF 中多于 32 个接口
625	MC_CamIN	SMC_CI_NO_CAM_SELECTED	没有 CAM 被选中
626	MC_CamIN	SMC_CI_MASTER_OUT_OF_SCALE	主轴超出范围
627	MC_CamIN	SMC_CI_RAMPIN_NEEDS_VELACC_VALUES	针对 ramp_in 功能块速度和加速度必须被精确指定
628	MC_CamIN	SMC_CI_SCALING_INCORRECT	比例变量 fEditor/TableMasterMin/Max 不正确
640	SMC_CAMBounds, SMC_CamBounds_Pos	SMC_CB_NOT_IMPLEMENTED	给予的 CAM 格式的功能块不支持
675	MC_GearIn	SMC_GI_RATIO_DENOM	RatioDenominator = 0
676	MC_GearIn	SMC_GI_INVALID_ACC	加速度不合适
677	MC_GearIn	SMC_GI_INVALID_DEC	加速度不合适
725	MC_Phase	SMC_PH_INVALID_VELACCDEC	速度，加速度，减速度不合适
726	MC_Phase	SMC_PH_ROTARYAXIS_PERIOD0	旋转轴 fPositionPeriod = 0
750	All modules using MC_CAM_REF as input	SMC_NO_CAM_REF_TYPE	给予的 CAM 不是类型 MC_CAM_REF
751	MC_CamTableSelect	SMC_CAM_TABLE_DOES_NOT_COVER_MASTER_SCALE	如果从 CamTable 中获取的数据不是通过数据转化得到的主轴区域 (xStart and xEnd) 。
775	MC_GearInPos	SMC_GIP_MASTER_DIRECTION_CHANGE	在从轴耦合过程中主轴改变旋转方向
800	SMC_BacklashCompensation	SMC_BC_BL_TOO_BIG	齿轮返回比 (fBacklash) 太大 (>position periode/2)
1000	CNC 需要授权的功能块	SMC_NO_LICENSE	目标没有进行 CNC 的授权。
1001	SMC_Interpolator	SMC_INT_VEL_ZERO	路径不能被处理因为速度 = 0.
1002	SMC_Interpolator	SMC_INT_NO_STOP_AT_END	上一个路径对象 Vel_End > 0.
1003	SMC_Interpolator	SMC_INT_DATA_UNDERRUN	警告： GEOINFO- 列表在 DataIn 进行处理，但是列表最后没有被设置。理由：忘记在 DataIn 中设置 EndOfList 或者 SMC_Interpolator 比路径编译模块处理速度快
1004	SMC_Interpolator	SMC_INT_VEL_NONZERO_AT_STOP	停止速度 > 0.
1005	SMC_Interpolator	SMC_INT_TOO_MANY_RECURSIONS	使用太多 SMC_Interpolator 调用 SoftMotion-错误。
1006	SMC_Interpolator	SMC_INT_NO_CHECKVELOCITIES	Input-OutQueue DataIn 没有作为 SMC_CHeckVelocities 的最后处理模块
1007	SMC_Interpolator	SMC_INT_PATH_EXCEEDED	内部 / 数值错误错误
1008	SMC_Interpolator	SMC_INT_VEL_ACC_DEC_ZERO	速度，加速度或者减速度为空或者太低。
1009	SMC_Interpolator	SMC_INT_DWIPOTIME_ZERO	FB 调用 dwIpoTime = 0
1050	SMC_Interpolator2Dir	SMC_INT2DIR_BUFFER_TOO_SMALL	数据缓冲区太小
1051	SMC_Interpolator2Dir	SMC_INT2DIR_PATH_FITS_NOT_IN_QUEUE	路径没有完全包含在队列中
1100	SMC_CheckVelocities	SMC_CV_ACC_DEC_VEL_NONPOSITIVE	速度，减速度或者加速度值不为正向
1120	SMC_Controlaxisbypos	SMC_CA_INVALID_ACCDEC_VALUES	变量 of fGapVelocity / fGapAcceleration / fGapDeceleration 不是正值
1200	SMC_NCDecoder	SMC_DEC_ACC_TOO_LITTLE	加速度值不允许
1201	SMC_NCDecoder	SMC_DEC_RET_TOO_LITTLE	减速度值不允许
1202	SMC_NCDecoder	SMC_DEC_OUTQUEUE_RAN_EMPTY	低于 Queue 的数据被读取并且为空。
1203	SMC_NCDecoder	SMC_DEC_JUMP_TO_UNKNOWN_LINE	因为行号未知所以跳转的行号不能执行
1204	SMC_NCDecoder	SMC_DEC_INVALID_SYNTAX	语法错误
1205	SMC_NCDecoder	SMC_DEC_3DMODE_OBJECT_NOT_SUPPORTED	这些对象不支持 3D 模式
1300	SMC_GCodeViewer	SMC_GCV_BUFFER_TOO_SMALL	缓冲区太小
1301	SMC_GCodeViewer	SMC_GCV_BUFFER_WRONG_TYPE	缓冲区元素类型错误
1302	SMC_GCodeViewer	SMC_GCV_UNKNOWN_IPO_LINE	当前插补行不能被找到
1500	使用 SMC_CNC_REF 的所有功能块	SMC_NO_CNC_REF_TYPE	给定的 CNC 程序不是类型 SMC_CNC_REF

错误代码	产生源	变量名称	错误原因描述
1501	所有使用 SMC_OUTQUEUE 的功能块	SMC_NO_OUTQUEUE_TYPE	给定的 OutQueue 不是类型 SMC_OUTQUEUE
1600	CNC 功能块	SMC_3D_MODE_NOT_SUPPORTED	这个功能块只在 2D 路径中可用
2000	SMC_ReadNCFile	SMC_RNCF_FILE_DOESNT_EXIST	文件不存在
2001	SMC_ReadNCFile	SMC_RNCF_NO_BUFFER	没有缓冲分配
2002	SMC_ReadNCFile	SMC_RNCF_BUFFER_TOO_SMALL	缓冲区太小
2003	SMC_ReadNCFile	SMC_RNCF_DATA_UNDERRUN	换种区中低缓冲数据被读取，为空
2004	SMC_ReadNCFile	SMC_RNCF_VAR_COULDNT_BE_REPLACED	占位符变量不能被替换
2005	SMC_ReadNCFile	SMC_RNCF_NOT_VARLIST	输入的 pvl 不能指向 SMC_VARLIST 对象
2050	SMC_ReadNCQueue	SMC_RNCQ_FILE_DOESNT_EXIST	文件不能打开
2051	SMC_ReadNCQueue	SMC_RNCQ_NO_BUFFER	没有缓冲区定义
2052	SMC_ReadNCQueue	SMC_RNCQ_BUFFER_TOO_SMALL	缓冲区太小
2053	SMC_ReadNCQueue	SMC_RNCQ_UNEXPECTED_EOF	未知文件结尾
2100	SMC_AxisDiagnosticLog	SMC_ADL_FILE_CANNOT_BE_OPENED	文件不能被打开
2101	SMC_AxisDiagnosticLog	SMC_ADL_BUFFER_OVERRUN	超过范围的缓冲； WriteToFile 必须更经常的调用
2200	SMC_ReadCAM	SMC_RCAM_FILE_DOESNT_EXIST	文件不能打开
2201	SMC_ReadCAM	SMC_RCAM_TOO_MUCH_DATA	保存到 CAM 数据太多
2202	SMC_ReadCAM	SMC_RCAM_WRONG_COMPILE_TYPE	错误编译模式
2203	SMC_ReadCAM	SMC_RCAM_WRONG_VERSION	文件版本错误
2204	SMC_ReadCAM	SMC_RCAM_UNEXPECTED_EOF	未知的文件结尾
3001	SMC_WriteDriveParamsToFile	SMC_WDPF_CHANNEL_OCCUPIED	SMC_WDPF_TIMEOUT_PREPARING_LIST
3002	SMC_WriteDriveParamsToFile	SMC_WDPF_CANNOT_CREATE_FILE	文件不能被创建
3003	SMC_WriteDriveParamsToFile	SMC_WDPF_ERROR_WHEN_READING_PARAMS	读取文件参数的时候错误
3004	SMC_WriteDriveParamsToFile	SMC_WDPF_TIMEOUT_PREPARING_LIST	准备参数列表时时间错误
5000	SMC_Encoder	SMC_ENC_DENOM_ZERO	译码器参考的转换因子 (dwRatioTechUnitsDenom) 为 0。
5001	SMC_Encoder	SMC_ENC_AXISUSEDBYOTHERFB	其他模块正在处理译码轴。
5002	驱动器接口	SMC_ENC_FILTER_DEPTH_INVALID	过滤器选择不合适

本手册如有参数更新，恕不另行通知。

南京德克威尔自动化有限公司
Nanjing Decowell Automation Co., Ltd.

全国服务热线

400-0969016

地址：南京市浦口区兰新路19号瑞创智造园13号楼

网址：www.welllinkio.com

邮箱：sales@welllinkio.com

德克威尔微信公众号

抖音官方账号