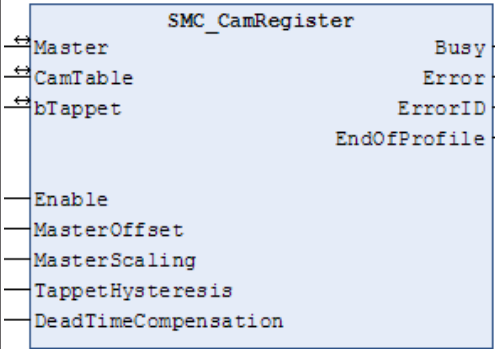


6.2 轴组指令（主 / 从轴指令）

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		<pre>SMC_CamRegister0(Master:= , CamTable:= , bTappet:= , Enable:= , MasterOffset:=0 , MasterScaling:= 1, TappetHysteresis:= , DeadTimeCompensation:= , Busy=> , Error=> , ErrorID=> , EndOfProfile=>);</pre>

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：

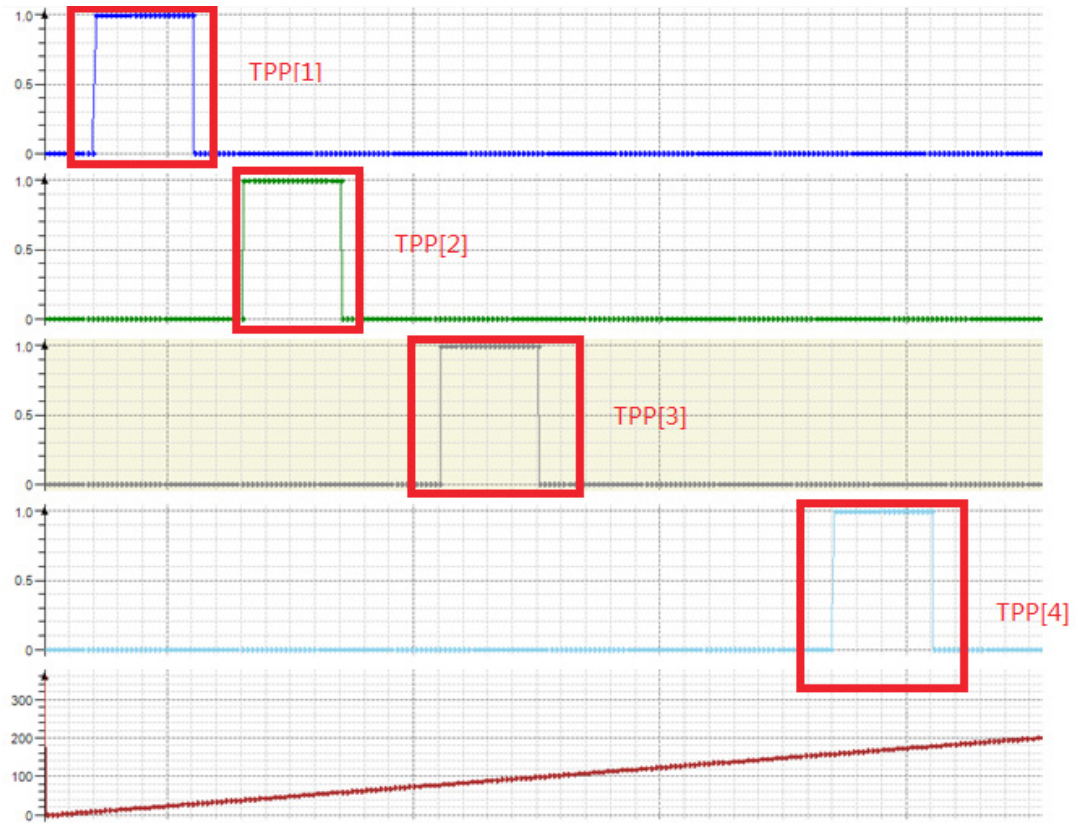
```
SMC_CamRegister0(
    Master:=Virtual_X ,
    CamTable:=Cam,
    bTappet:=TPP ,
    Enable:=TRUE ,
    MasterOffset:=0 ,
    MasterScaling:= 1,
    TappetHysteresis:= 0,
    DeadTimeCompensation:=0 ,
    Busy=> ,
    Error=> ,
    ErrorID=> ,
    EndOfProfile=> );
```

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：

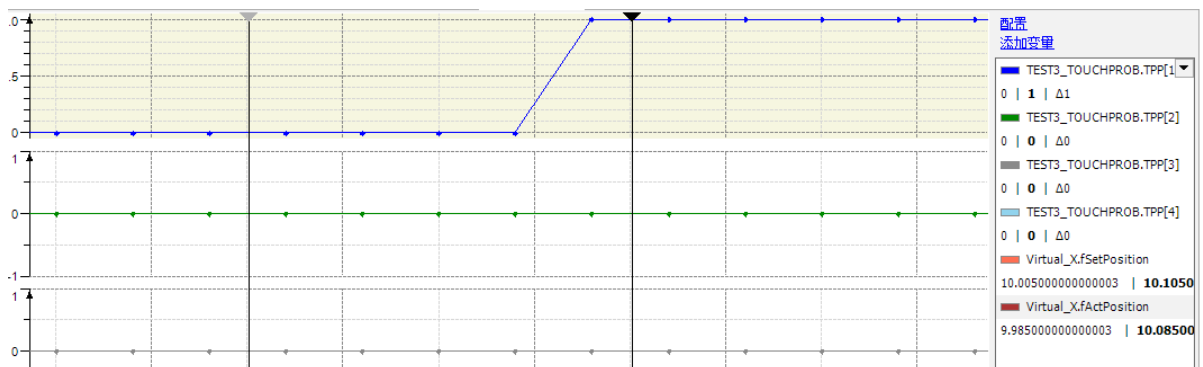
监控曲线如下图：



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=> ,
    EndOfProfile=> );
```

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

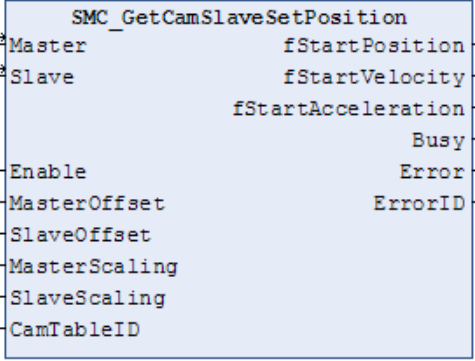
轴有错误、轴没有使能、偏移值或者标尺值设置超过主轴范围。

【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。

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_GetCamSlaveSetPosition	获取凸轮从轴位置		<pre>SMC_GetCamSlaveSetPosition0(Master:= Slave:= Enable:= MasterOffset:= SlaveOffset:= MasterScaling:= SlaveScaling:= CamTableID:= fStartPosition=> fStartVelocity=> fStartAcceleration=> Busy=> Error=> ErrorID=>);</pre>

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： $Y = (\text{cam}((\text{凸轮起始主轴查表位置} + \text{Masteroffset}) * \text{MasterScaling}) + \text{slaveoffset}) * \text{SlaveScaling}$ ，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；

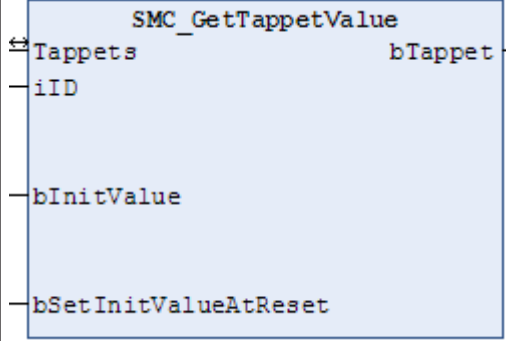
参考 ErrorID,SMC_ERROR 确定错误原因。

【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。

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		<pre>SMC_GetTappetValue0(Tappets:= , iID:= , bInitValue:= , bSetInitValueAtReset:= , bTappet=>);</pre>

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。

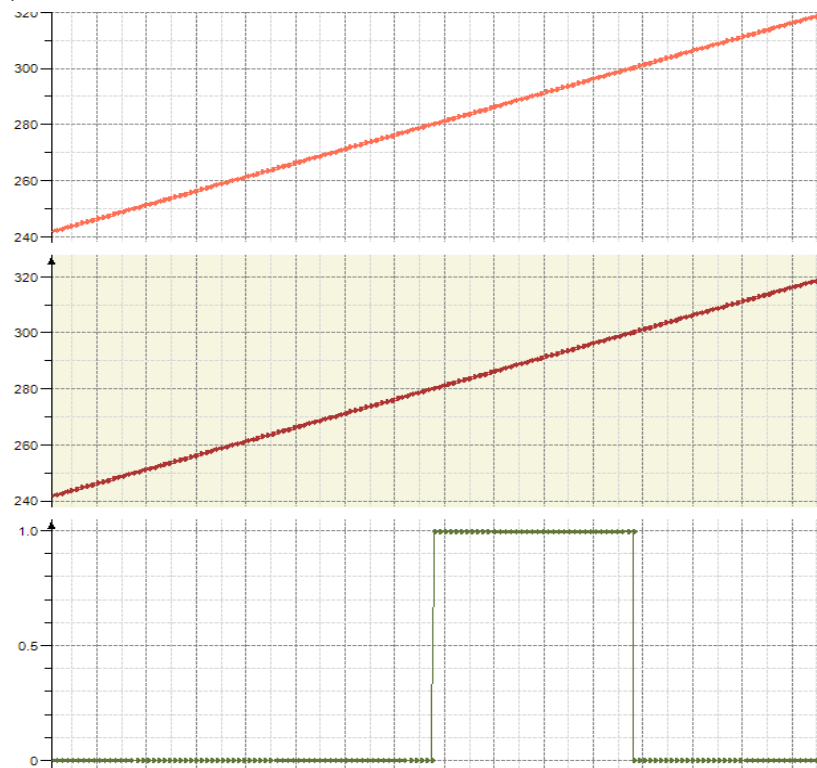
使用示例：

```
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=> ,
EndOfProfile=> ,
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；

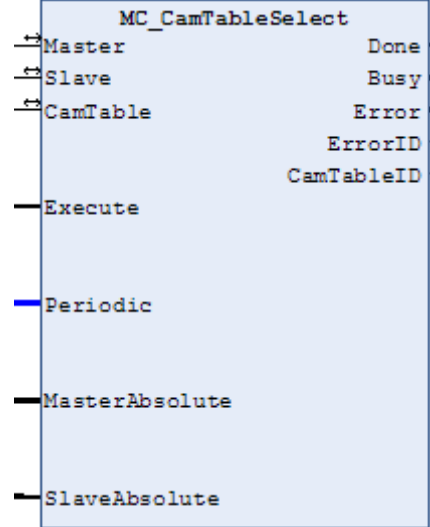
CamTable ID 没有指向。

【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。

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		<pre>MC_CamTableSelect0(Master:= Slave:= CamTable:=), Execute:= Periodic:= MasterAbsolute:= SlaveAbsolute:= Done=> Busy=> Error=> ErrorID=> CamTableID=>);</pre>

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
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

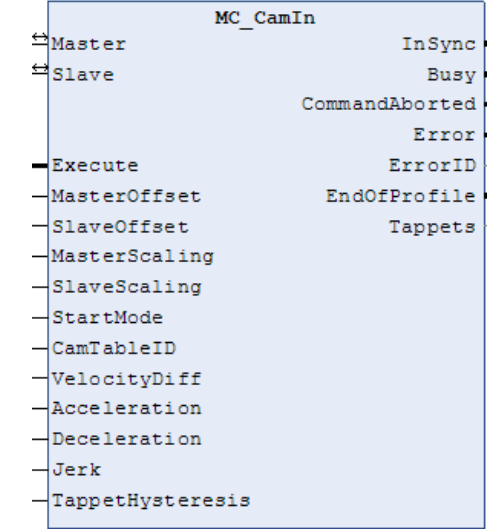
4) 错误说明

- ◆ 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。

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		<pre>MC_CamIn0(Master:= , Slave:= , Execute:= , MasterOffset:= , SlaveOffset:= , MasterScaling:= , SlaveScaling:= , StartMode:= , CamTableID:= , VelocityDiff:= , Acceleration:= , Deceleration:= , Jerk:= , TappetHysteresis:= , InSync=> , Busy=> , CommandAborted=> , Error=> , ErrorID=> , EndOfProfile=> , Tappets=>);</pre>

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		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			定义 cam 表格的使用，与 MC_CamTableSelect 的输出点 CamTableID 配合使用
VelocityDiff		LREAL			与 ramp_in 不同的最大速度
Acceleration		LREAL			ramp_in 时加速度
Deceleration		LREAL			ramp_in 时减速度

输入变量	名称	数据类型	有效范围	初始值	描述
Jerk		LREAL			ramp_in 的加速度
TappetHysteresis		LREAL			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 Command-Aborted 输出为 TRUE。

4) 指令详情

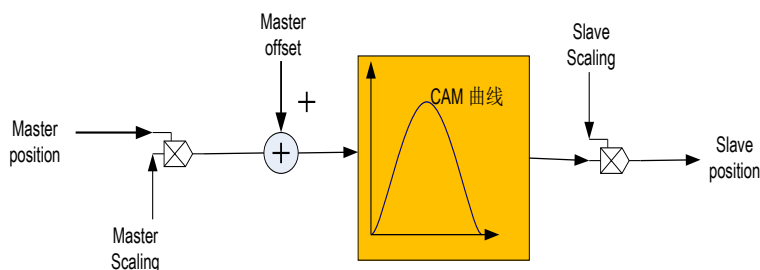
下面对指令详细说明：

◆ 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

$$\text{Position_Slave} = \text{SlaveScaling} * \text{CAM}(\text{MasterScaling} * \text{MasterPosition} + \text{MasterOffset}) + \text{SlaveOffset}$$

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 EndOfProfile 的关系：

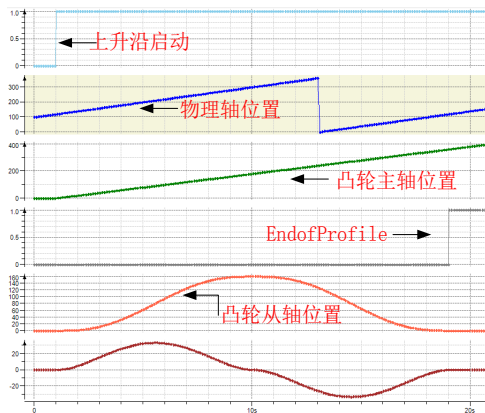
周期模式非周期模式决定了主轴到终止位置后电子凸轮是否要再次进行。

非周期模式：MC_CamTableSelect 指令 Periodic 选择 False：

非周期模式时，凸轮完成 EndofProfile 信号输出为 True，执行输入 FALSE 则 EndofProfile 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



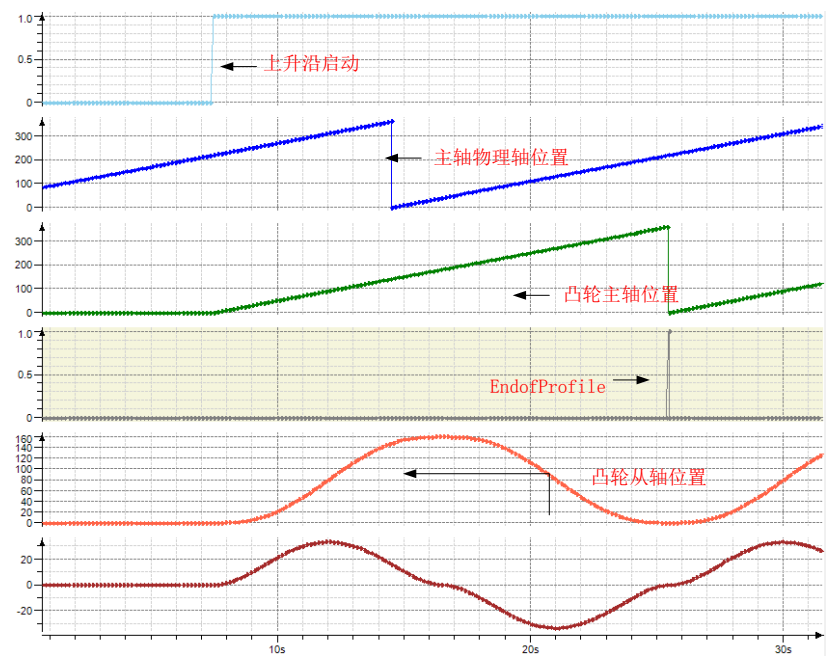
此时凸轮一个主轴周期完成后会接着执行下一个周期，且 EndofProfile 信号 TRUE 输出只维持一个任务周期。

注意：

当凸轮主轴位置大于等于凸轮终止位置时 EndofProfile 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 EndofProfile 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。)

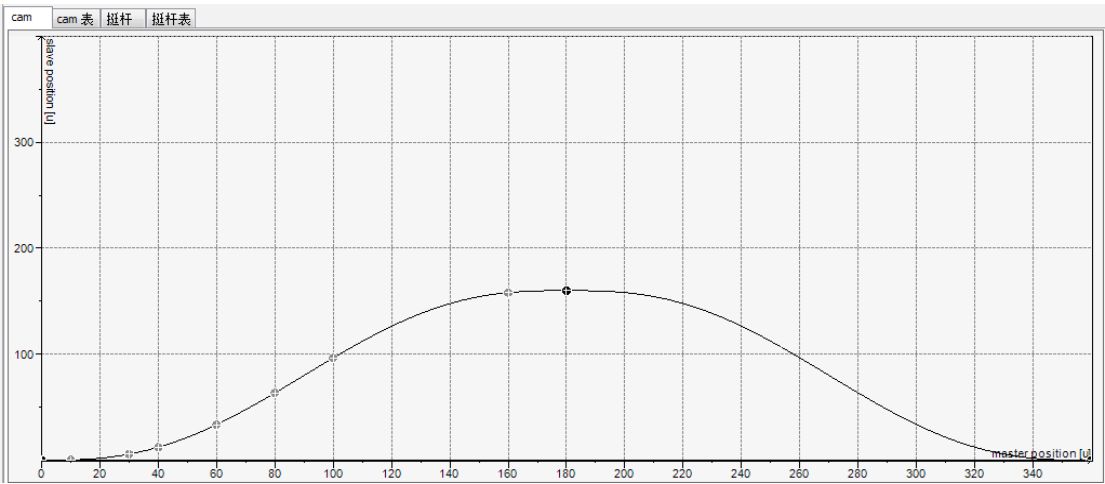
关系表如下表所示:

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
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:

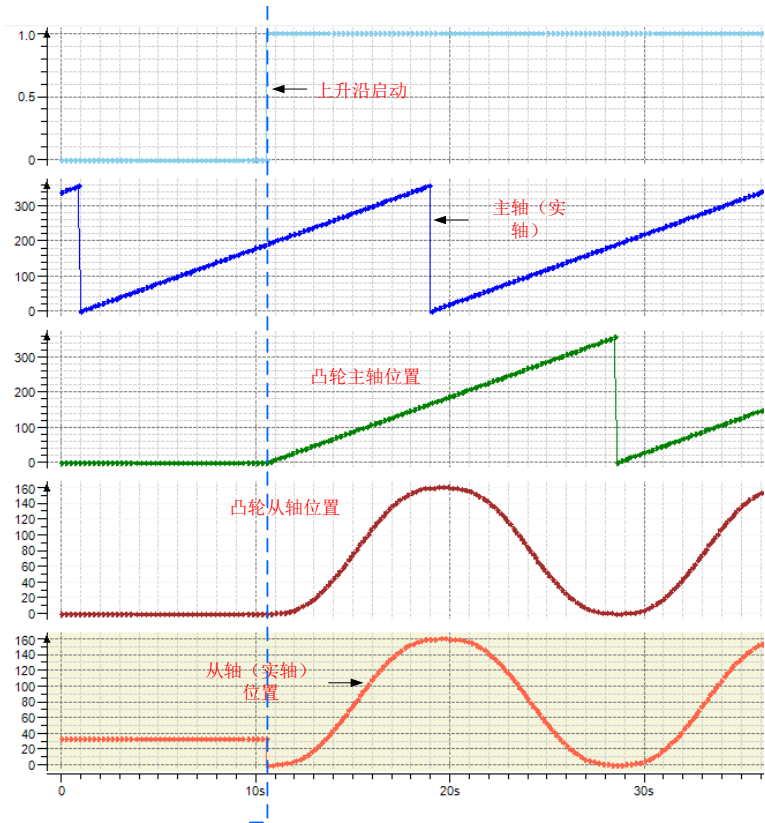
凸轮主轴范围（0-360）、凸轮从轴范围为（0-180）、周期模式、主从轴偏移值 0、主从轴缩放比 1。设计的凸轮表如下图所示：



② StartMode 为 0（绝对模式）

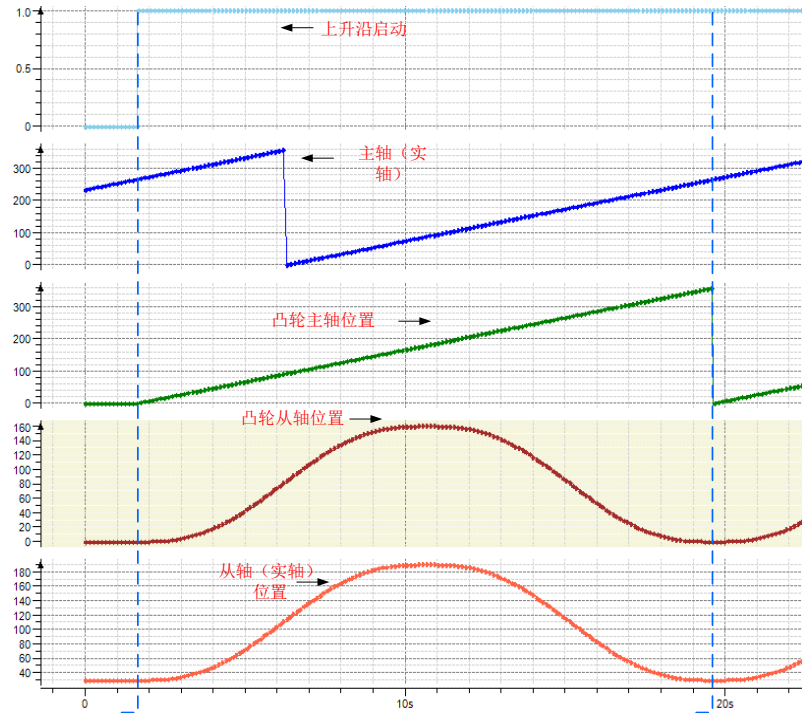
当 MC_CamTableSlect 指令 MasterAbsolute 设置为 FALSE，SlaveAbsolute 设置为 TRUE 时，此时主轴工作在相对模式，从轴工作在绝对模式。当 Excute 上升沿，凸轮启动时，凸轮主轴从凸轮表“起始位置”（0）开始，凸轮从轴按照上述的“凸轮表齿合公式”计算输出，从轴实轴指令位置等于齿合计算输出值。比如说凸轮从轴起始位置为 0，凸轮启动时从轴实轴位置为 20，则启动开始从轴实轴的位置指令为 0 产生跳变。

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。



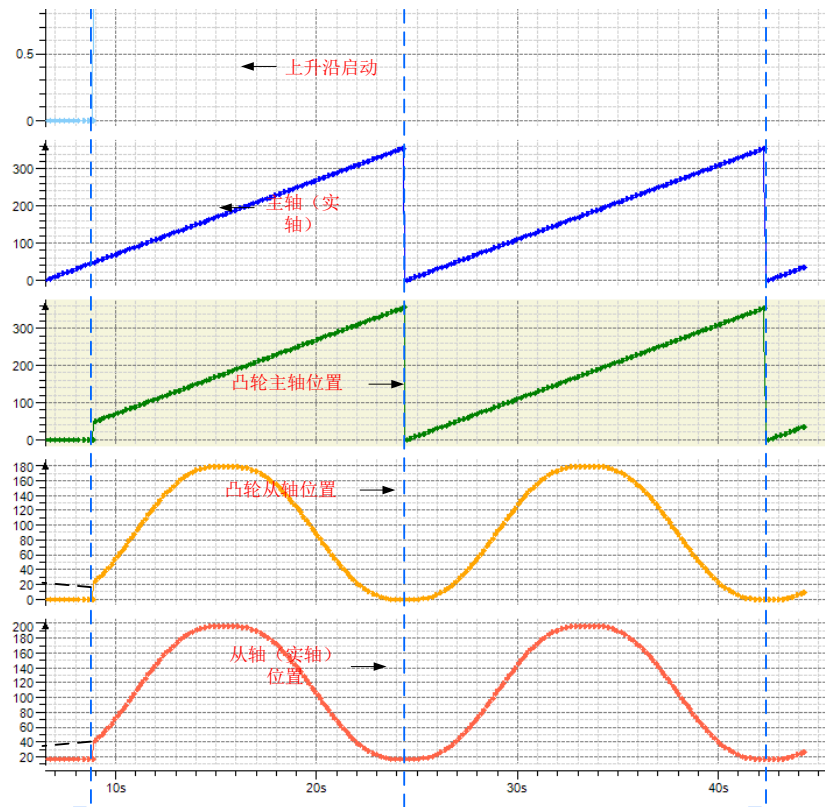
5) 错误说明

- ◆ 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

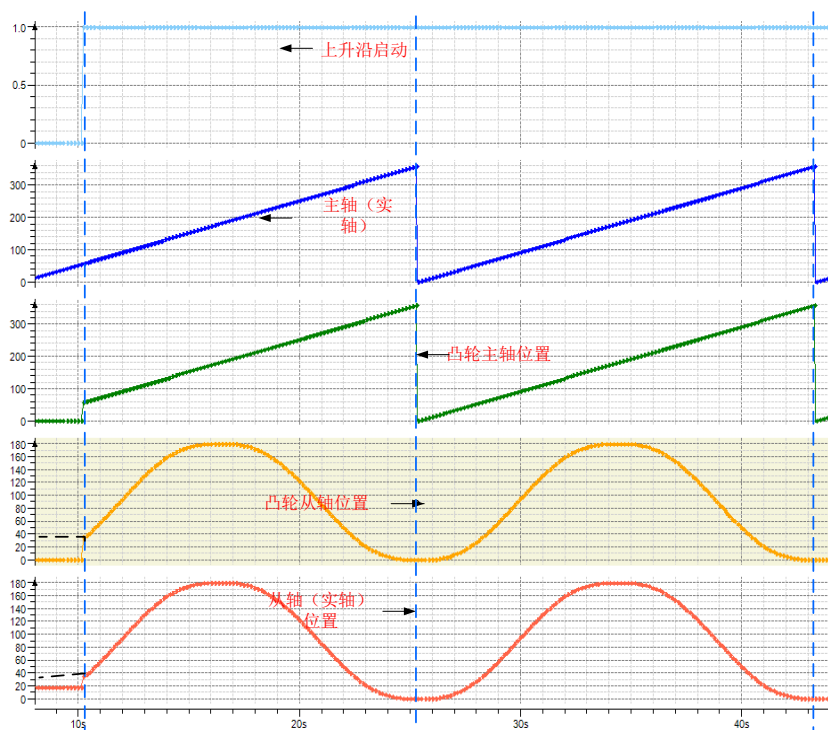


当 MC_CamTableSlect 指令 MasterAbsolute 设置为 TRUE, SlaveAbsolute 设置为 TRUE 时, 此时主轴工作在绝对模式, 从轴工作在绝对模式。当 Excute 上升沿, 凸轮启动时, 凸轮主轴从当前“主轴实轴位置开始”, 从轴实轴位置指令 = 凸轮表齿合计算值 (凸轮从轴位置)。

注意:

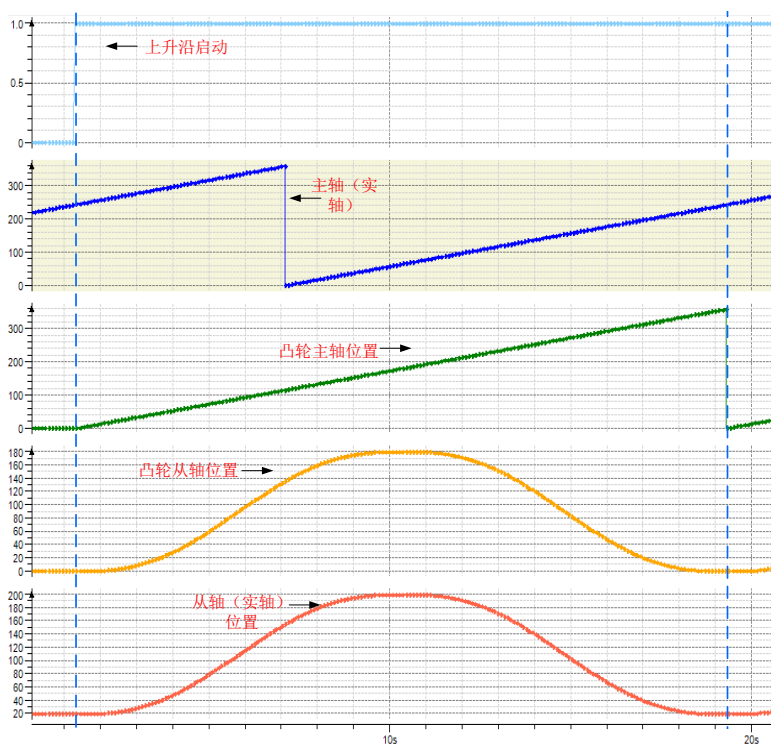
1 如果该情况下主轴 (实轴) 开始位置不在凸轮主轴开始位置、从轴位置不在凸轮从轴开始位置则会产生跳变。

2 主轴位置应在凸轮主轴位置范围内



2) 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)。

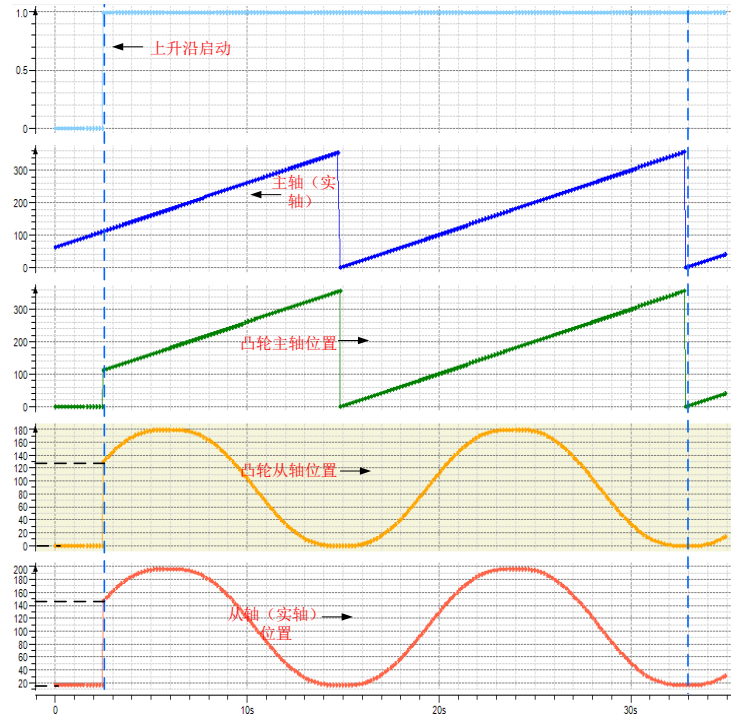


当 MC_CamTableSlect 指令 MasterAbsolute 设置为 TRUE, SlaveAbsolute 设置为 TRUE 或者为 False 时, 此时主轴工作在绝对模式, 从轴工作在相对模式。当 Excute 上升沿, 凸轮启动时, 凸轮主轴从 “主轴当前位置开始”, 从轴实轴位置指令 = 启动时从轴位置 + 凸轮表齿合计算值 (凸轮从轴位置)。

注意:

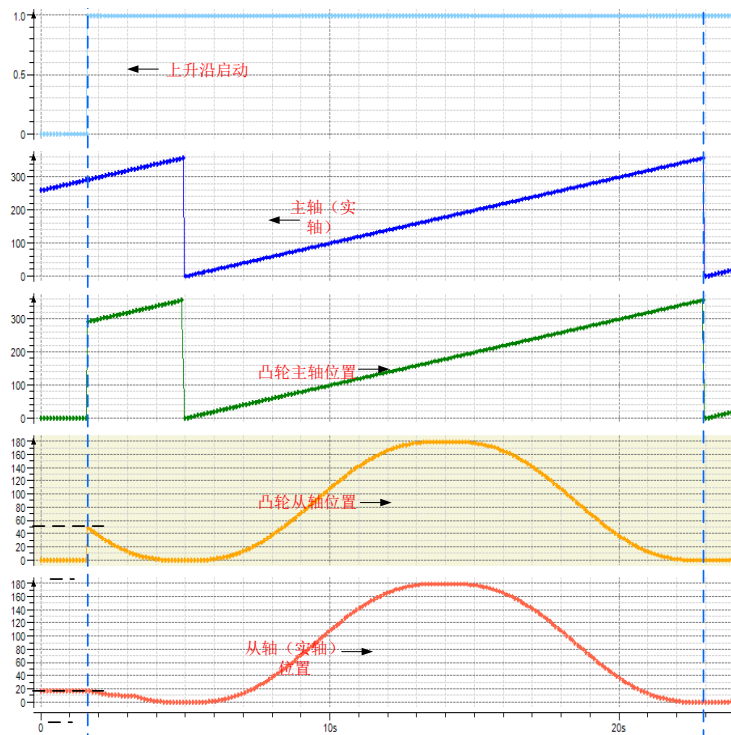
1 如果该情况下主轴 (实轴) 开始位置不在凸轮主轴开始位置则会产生跳变。

2 主轴位置应在凸轮主轴位置范围内

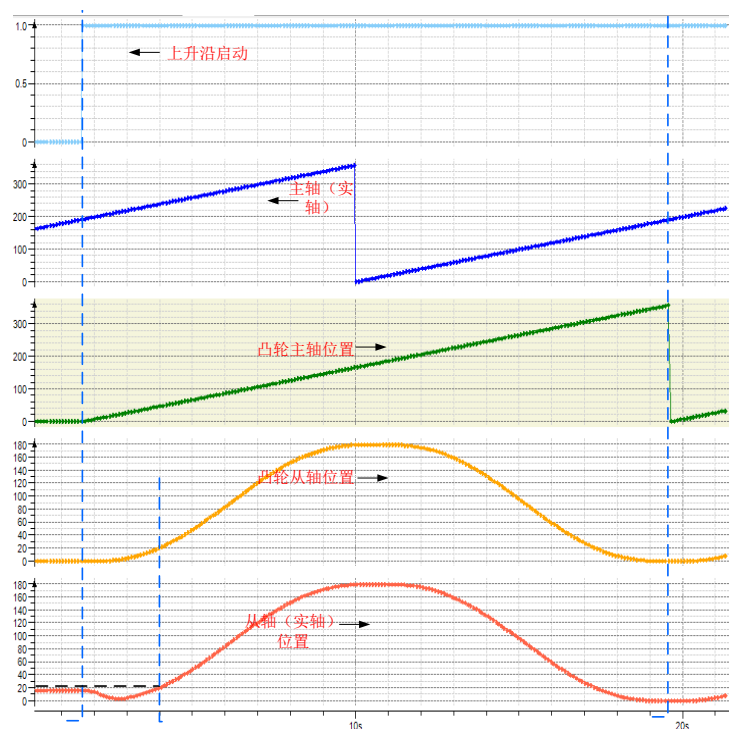


3) StartMode 为 2 (rampin 斜坡切入模式)

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”, 从轴通过设置的 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)。

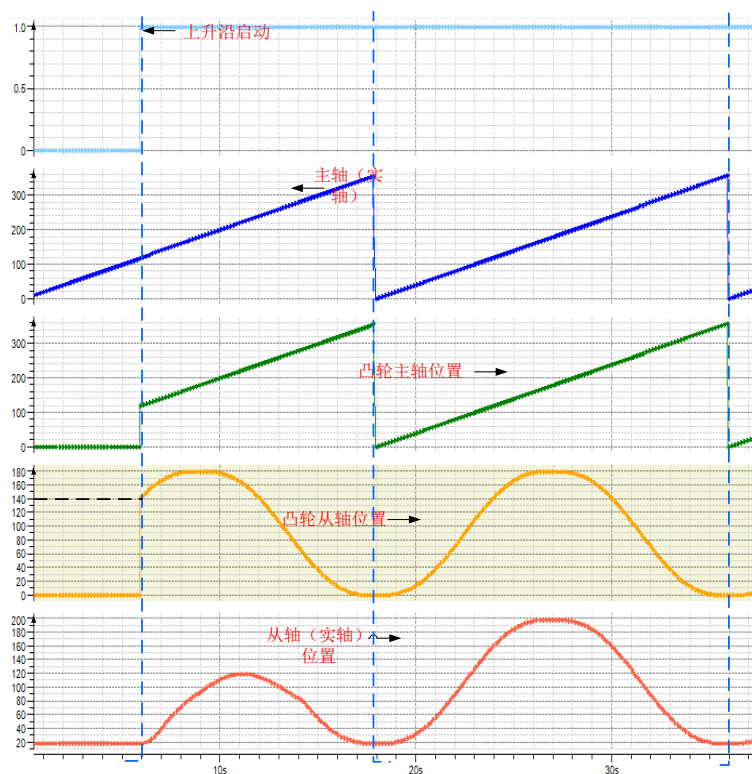


当 MC_CamTableSlect 指令 MasterAbsolute 设置为 FALSE，SlaveAbsolute 设置为 TRUE 时，此时主轴工作在相对模式，从轴工作在绝对模式。当 Excute 上升沿，凸轮启动时，凸轮主轴从“凸轮主轴起始位置开始”，从轴通过设置的 VelocityDiff、Acceleration、Deceleration 增加一个补偿运动避免切入时的潜在跳变。从轴实轴位置指令 = 凸轮表齿合计算值（凸轮从轴位置）+f(VelocityDiff,Acceleration,Deceleration)。



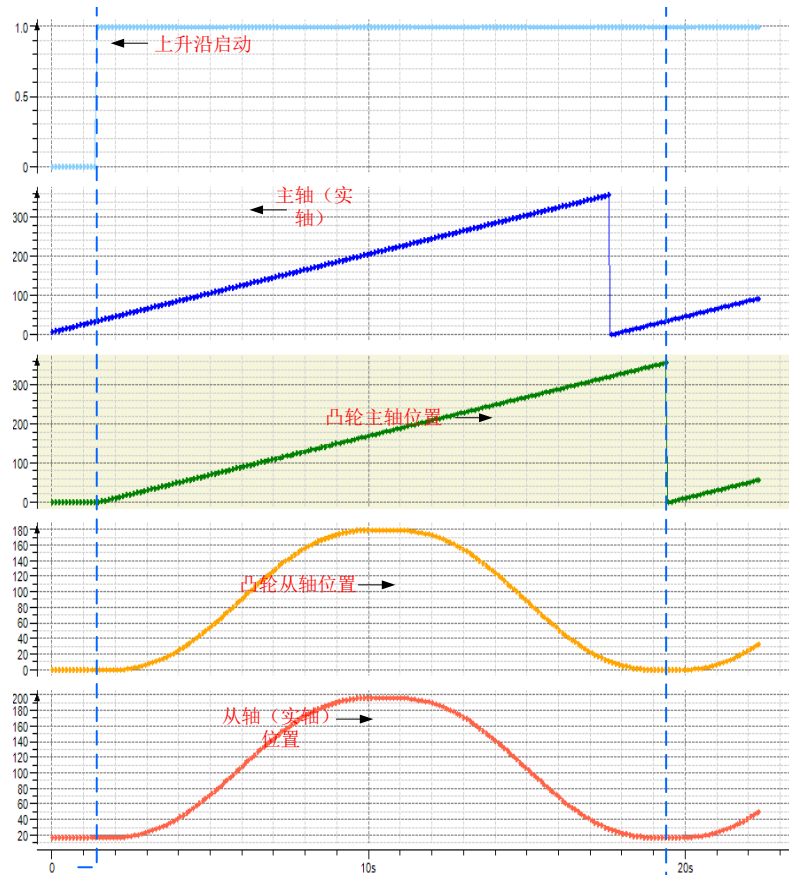
当 MC_CamTableSlect 指令 MasterAbsolute 设置为 TRUE，SlaveAbsolute 设置为 FALSE 时，此时主轴工作在绝对模式，从轴工作在相对模式。当 Excute 上升沿，凸轮启动时，凸轮主轴从“主轴当前位置开始”，从轴通过设置的 VelocityDiff、Acceleration、Deceleration 增加一个补偿运动避免切入时的潜在跳变。从轴实轴位置指令 = 从轴当前位置 + 凸轮表齿合计算值（凸轮从轴位置）+f(VelocityDiff,Acceleration,Deceleration)。

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,Deceleration)。

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



4) 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：

$$X = \text{MasterScaling} * \text{MasterPosition} + \text{MasterOffset}$$

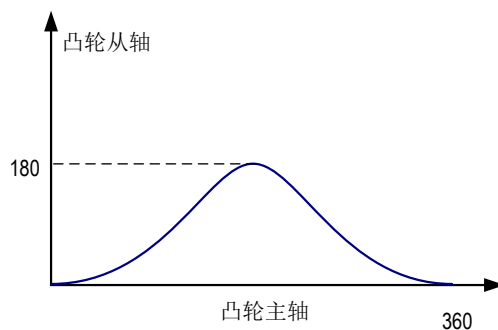
因此，如果 MasterScaling 的值大于 1，电子凸轮将在较高速度下运行；如果值小于 1，电子凸轮将在较低速度运行。

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：

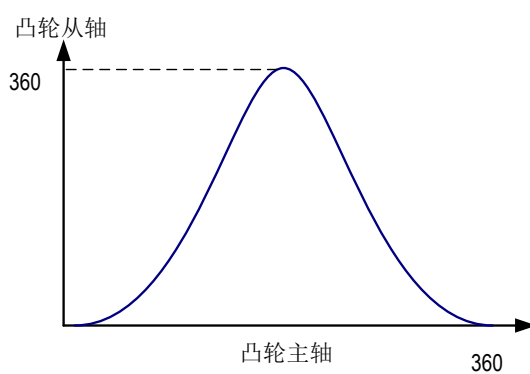
$$Y = \text{SlaveScaling} * \text{CAM}(X) + \text{SlaveOffset}$$

A SlaveScaling > 1，拉伸电子凸轮，从轴运动范围增大；同理，SlaveScaling < 1，缩小电子凸轮从轴运动范围。

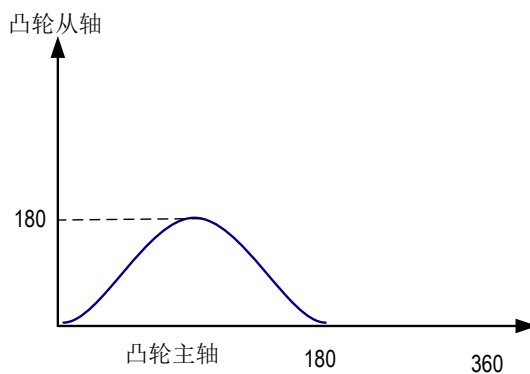
当 $\text{MasterScaling}=1.0$ 、 $\text{SlaveScaling}=1.0$ 、 $\text{MasterOffset}=0$ 、 $\text{SlaveOffset}=0$ ，此时凸轮曲线为规划的凸轮曲线如下图所示：



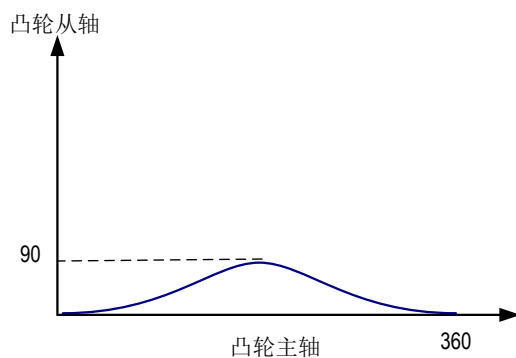
当 $\text{MasterScaling}=1.0$ 、 $\text{SlaveScaling}=2.0$ 、 $\text{MasterOffset}=0$ 、 $\text{SlaveOffset}=0$ ，此时凸轮曲线如下图所示：



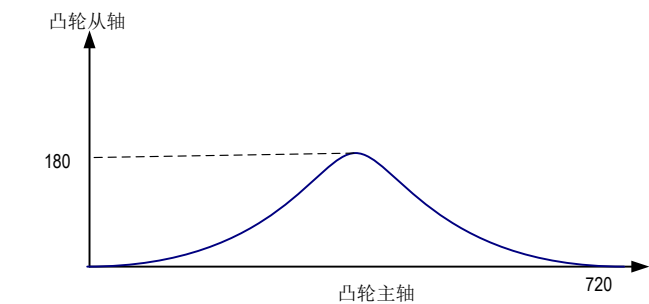
当 $\text{MasterScaling}=2.0$ 、 $\text{SlaveScaling}=1.0$ 、 $\text{MasterOffset}=0$ 、 $\text{SlaveOffset}=0$ ，此时凸轮曲线如下图所示：



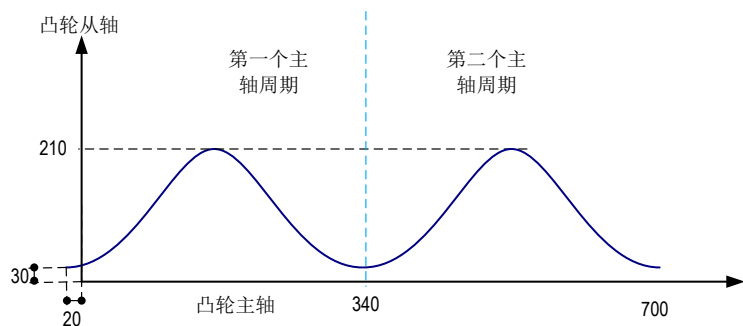
当 $\text{MasterScaling}=1.0$ 、 $\text{SlaveScaling}=0.5$ 、 $\text{MasterOffset}=0$ 、 $\text{SlaveOffset}=0$ ，此时凸轮曲线如下图所示：



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



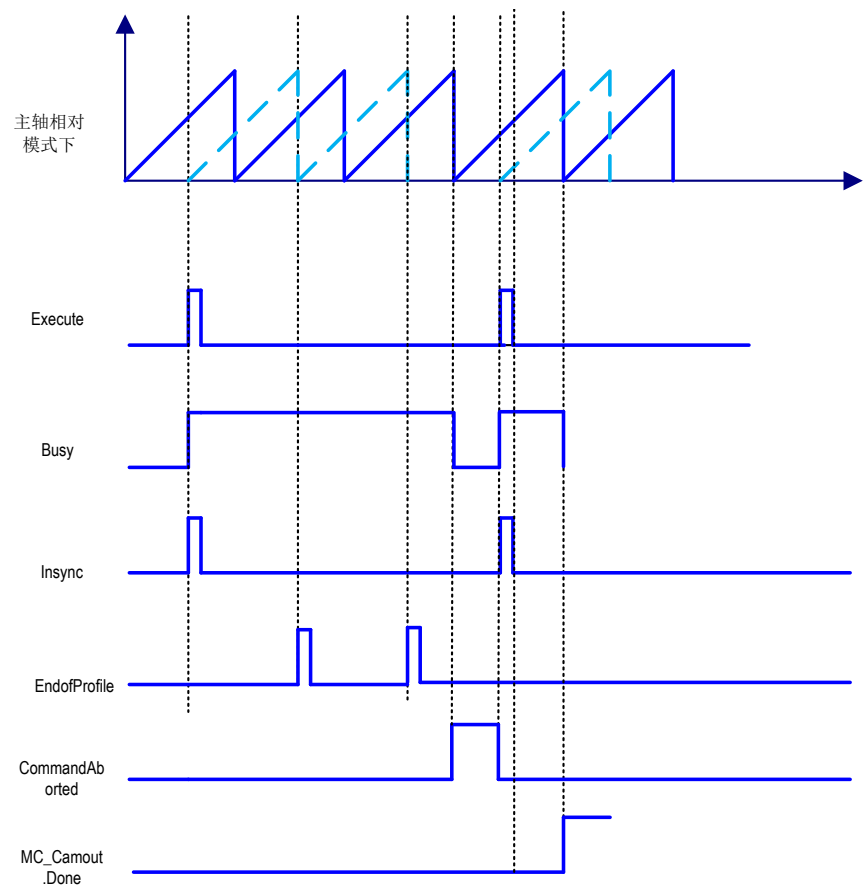
当 MasterScaling=1、SlaveScaling=1、MasterOffset=20、SlaveOffset=30，此时凸轮曲线如下图所示：



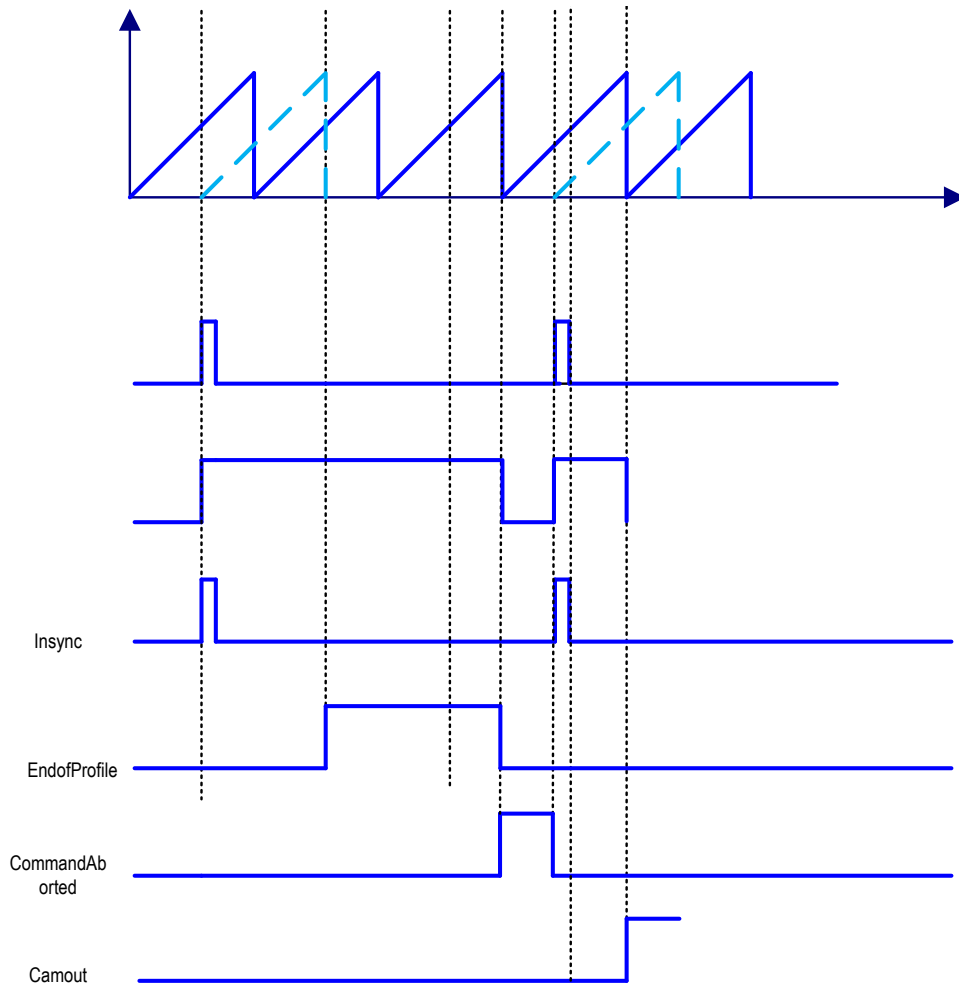
6) 时序图：

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： $\text{SlavePosition} = \text{CAM}(\text{MasterPosition})$ 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： $\text{SlavePosition} = \text{CAM}(\text{MasterScale} * \text{MasterPosition} + \text{MasterOffset})$ 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： $\text{SlavePosition} = \text{SlaveScale} * \text{CAM}(\text{MasterPosition}) + \text{SlaveOffset}$ 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：

$$\text{Slaveposition} = \text{SlaveScale} * \text{CAM}(\text{MasterScale} * \text{Masterposition} + \text{MasterOffset}) + \text{SlaveOffset}$$

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

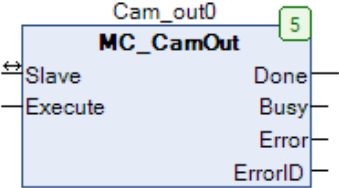
启动本指令检测到异常时，Error(错误) 变为 TRUE。

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。

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		<pre>MC_CamOut(Slave:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);</pre>

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	-	-	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。

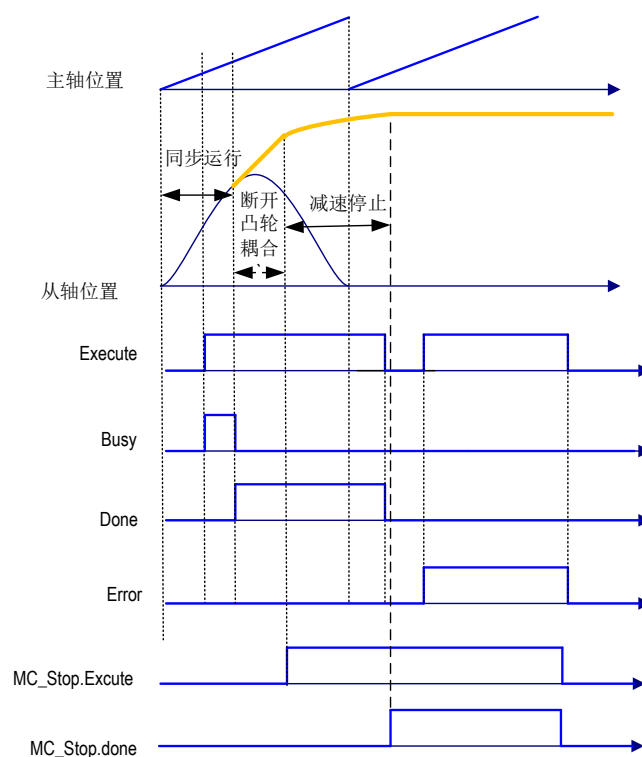
Excute 上升沿时执行从轴的凸轮耦合关系断开

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

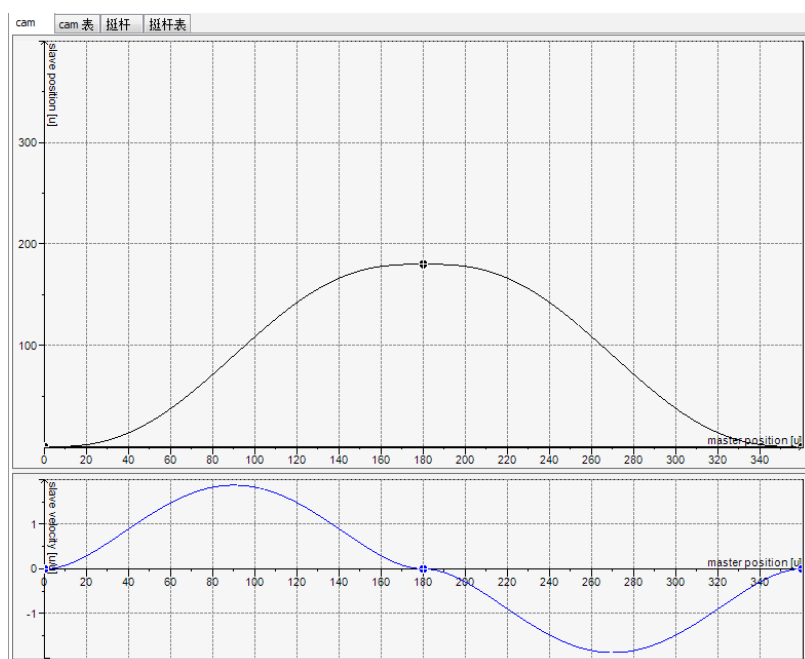


5) 使用范例

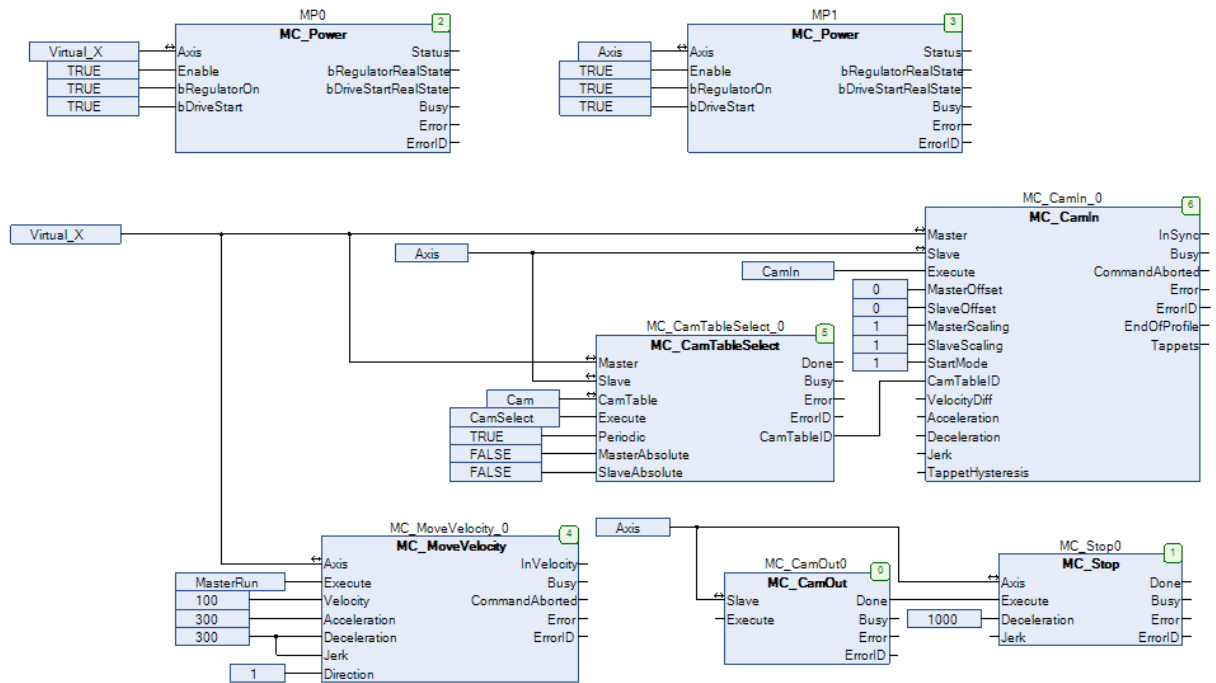
本范例应用凸轮相关指令，介绍凸轮关系的建立、运行及脱离时轴的相关运动状态

凸轮编辑器建立如下凸轮表 (cam) :

cam	cam 表	扭杆	扭杆表								
		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。

注意：

凸轮表在线修改等请参考运动控制功能篇

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。

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		<pre>MC_GearIn0(Master:= Slave:= Execute:= RatioNumerator:= RatioDenominator:= Acceleration:= Deceleration:= Jerk:= InGear=> Busy=> CommandAborted=> Error=> ErrorID=>);</pre>

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- FAKSE	-FAKSE	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	正数或 0		指定加速度
Deceleration	减速度	LREAL	正数或 0		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- FAKSE	- 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

Execute 上升沿，开始电子齿轮动作。

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。

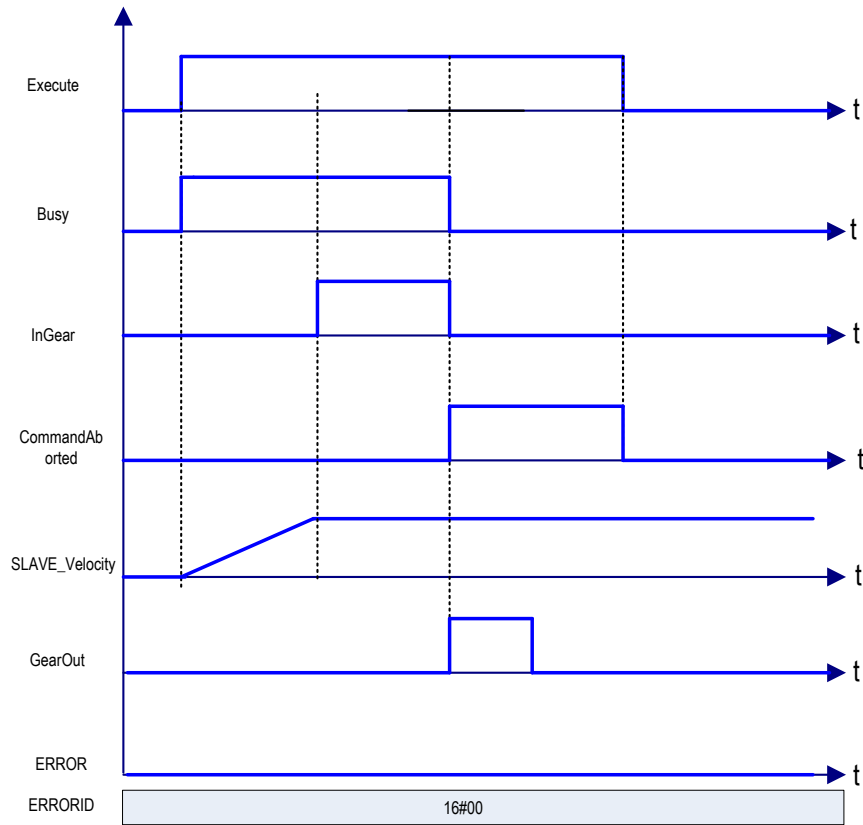
指令执行中 **Busy** 信号为 **TRUE** 时，如果从轴目标速度达到此时 **Execute** 新的上升沿不会影响它。

到达目标速度，**InGear** 为 **TRUE**，此后从轴移动量 = 主轴移动量 * $\text{RatioNumerator} / \text{RatioDenominator}$ 。

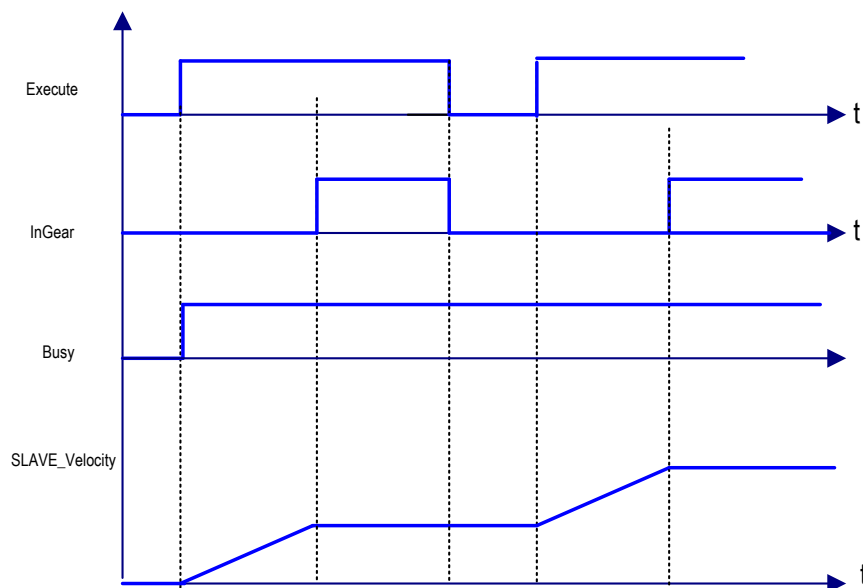
如果主轴速度实时变化的，情况下请注意慎重使用该指令。

注意：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。

◆ 时序图：



变更齿轮比参数后重启指令时序图如下：



4) 错误说明

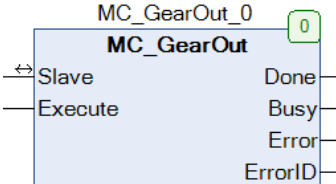
启动指令，**ERROR** 为 **TRUE**，则有错误输出。

相关错误根据 **ERRORID** 参考 **SMC_ERROR**。请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

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		<pre>MC_GearOut0(Slave:= Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);</pre>

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	-	-	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- FAKSE	FAKSE	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- FAKSE	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

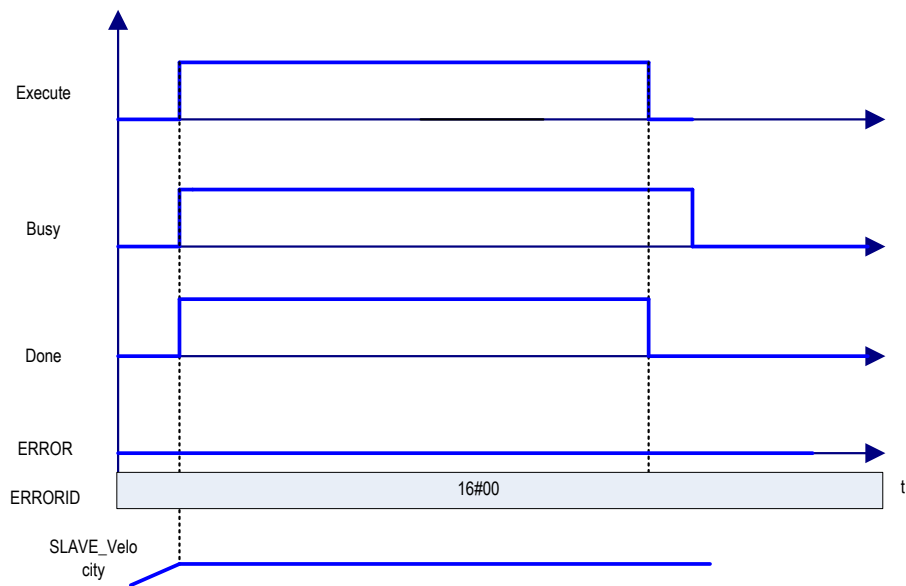
Execute 上升沿，执行切出电子齿轮动作。

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。

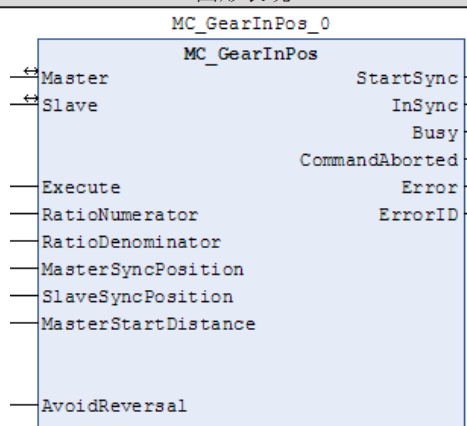
【注意】：请阅读 [“附录 C 错误代码说明”](#) 以了解相关错误代码说明。

MC_GearInPos

设定主轴与从轴之间的电子齿轮比，进行电子齿轮动作。

指定开始同步的主轴位置、从轴位置、主轴开始同步距离，并以此来完成切入电子齿轮动作。

1) 指令格式

指令	名称	图形表现	ST 表现
MC_GearInPos	指定位置 切入电子 齿轮耦合		<pre>MC_GearInPos0(Master:= Slave:= Execute:= RatioNumerator:= RatioDenominator:= MasterSyncPosition:= SlaveSyncPosition:= MasterStartDistance:= AvoidReversal:= StartSync=> InSync=> Busy=> CommandAborted=> Error=> ErrorID=>);</pre>

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	指令执行	BOOL	TRUE- FAKSE	-FALSE	上升沿，开始执行指令
RatioNumerator	齿轮比分子	DINT	-	1-	主从轴速度比的分子
RatioDenominator	齿轮比分母	DINT		1	主从轴速度比的分母
MasterSyncPosition	主轴同步位置	LREAL			主从轴齿轮比耦合时主轴位置
SlaveSyncPosition	从轴同步位置	LREAL			主从轴齿轮比耦合时从轴位置
MasterStartDistance	执行同步主轴 位置	LREAL			从轴按照该位置值和 -MasterSyncPosition 以及 SlaveSyncPosition 值计算一条平滑曲线使从轴在 SlaveSyncPosition 时跟主轴齿轮同步，曲线主轴范围为 [MasterStartDistance, MasterSyncPosition]
AvoidReversal	禁止反转	BOOL	TRUE- FAKSE	FALSE	设置为 FALSE，如果从轴物理位置超前的情况下进行反转。设置为 TRUE 如果从轴物理上不能实现反转或者导致危险。只在模态轴下适用。如果反转不能被避免，那么轴将错误停止。

◆ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
StartSync	开始耦合处理	BOOL	TRUE- FALSE	ALSE	True, 开始电子齿轮耦合处理
InSync	耦合中	BOOL	TRUE- FALSE	FALSE	True, 电子齿轮耦合处理完成，主从轴齿轮比耦合中
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 上升沿信号，开始执行指令。

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。

该功能块同步开始到同步结束的过程本质为同步区间内从轴跟随主轴的一个电子凸轮、此时根据主轴范围（MasterSyncPosition- MasterStartDistance, MasterSyncPosition），从轴范围（当前位置，SlaveSyncPosition），指令会根据设置齿轮比以及上述三个参数自动设计一条凸轮曲线，执行同步时从轴跟随主轴完成凸轮动作。

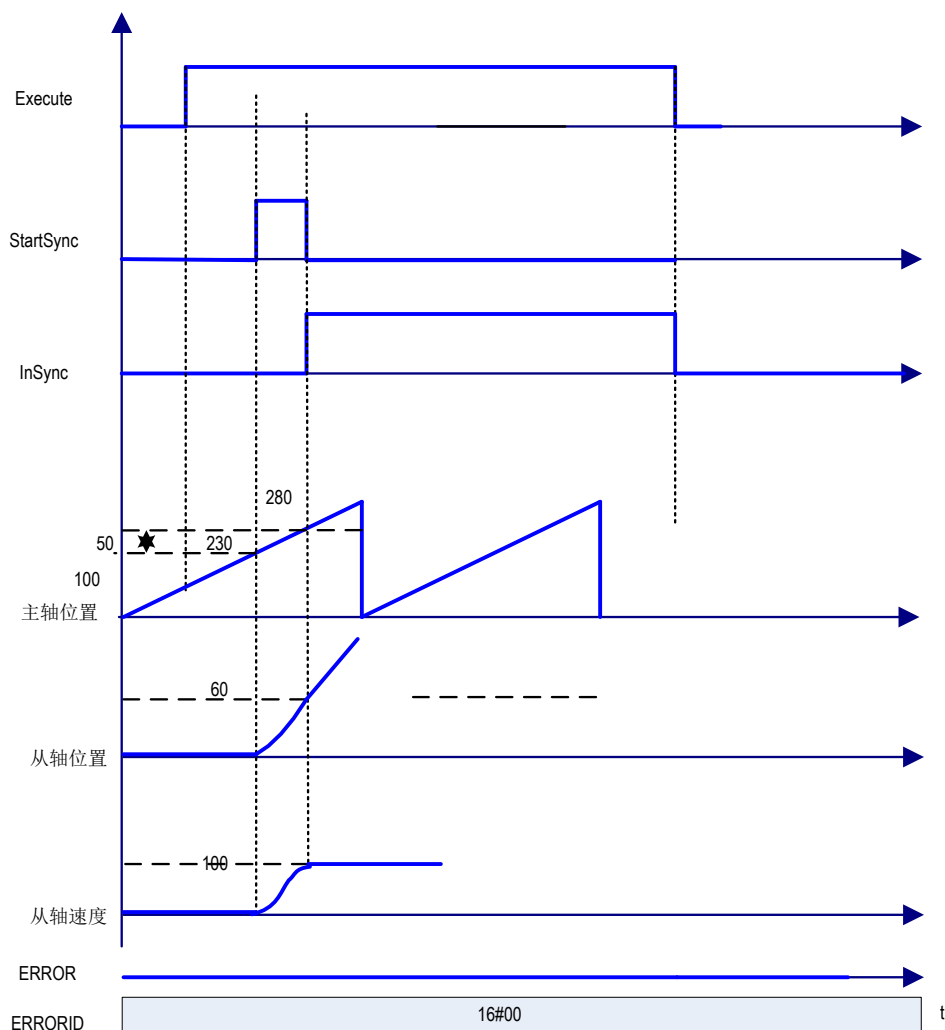
注意如果主从轴工作在线性模式需保证需保证上述几个参数设置合理否则齿轮动作无法正确进行，故而建议使用该指令时主从轴工作在周期模式。

例如：主从轴线性工作模式都向正向运动，如果执行指令时主轴位置 > MasterSyncPosition- MasterStartDistance，或者从轴位置 > SlaveSyncPosition 则无法切入电子齿轮动作。

下面给出了几种不同参数下的样例时序图：

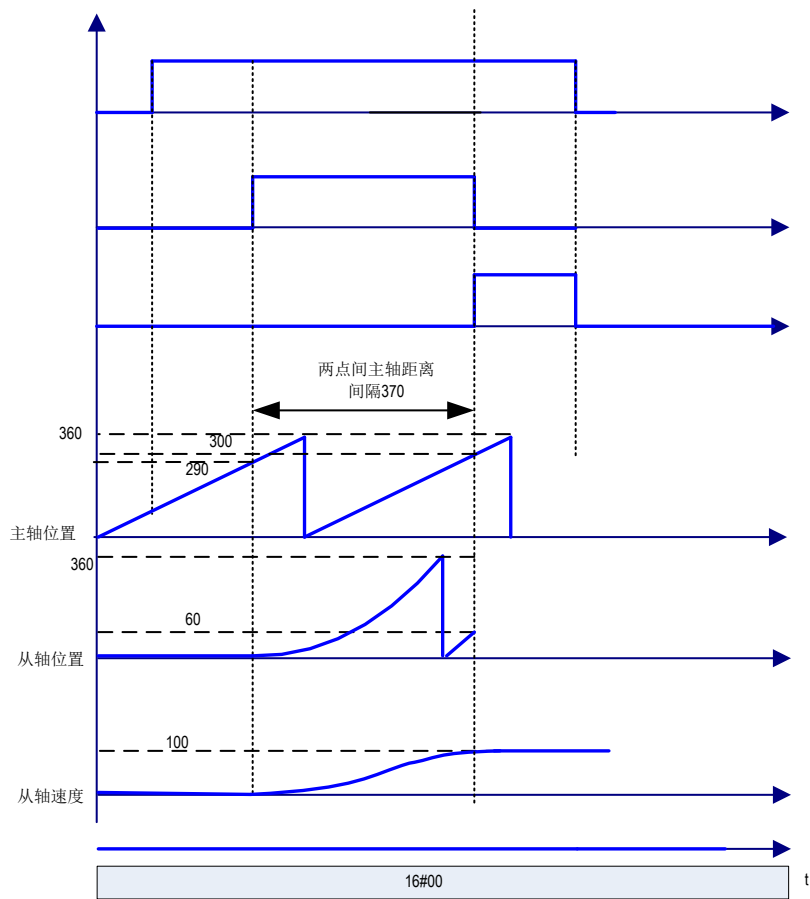
当主轴工作在周期模式（360 循环）、从轴工作在周期模式（360 循环）：

- ① MasterSyncPosition=280、MasterStartDistance=50、SlaveSyncPosition=60，主轴速度为 50、AvoidReversal=FALSE

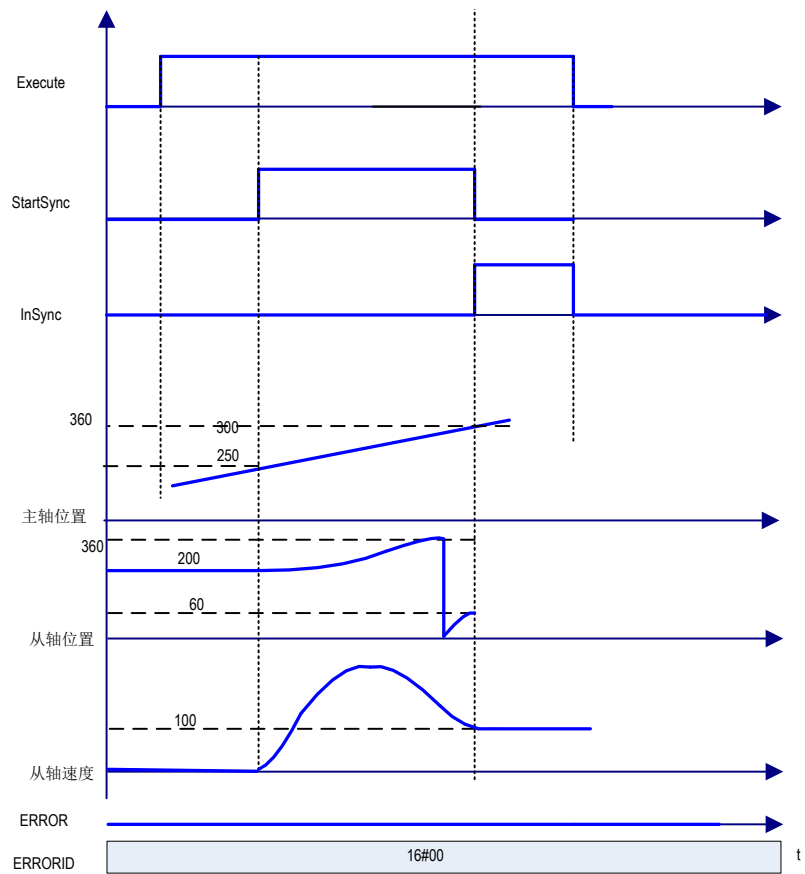


- ② MasterSyncPosition=300、MasterStartDistance=370、SlaveSyncPosition=60，主轴速度为 50、

AvoidReversal=FALSE



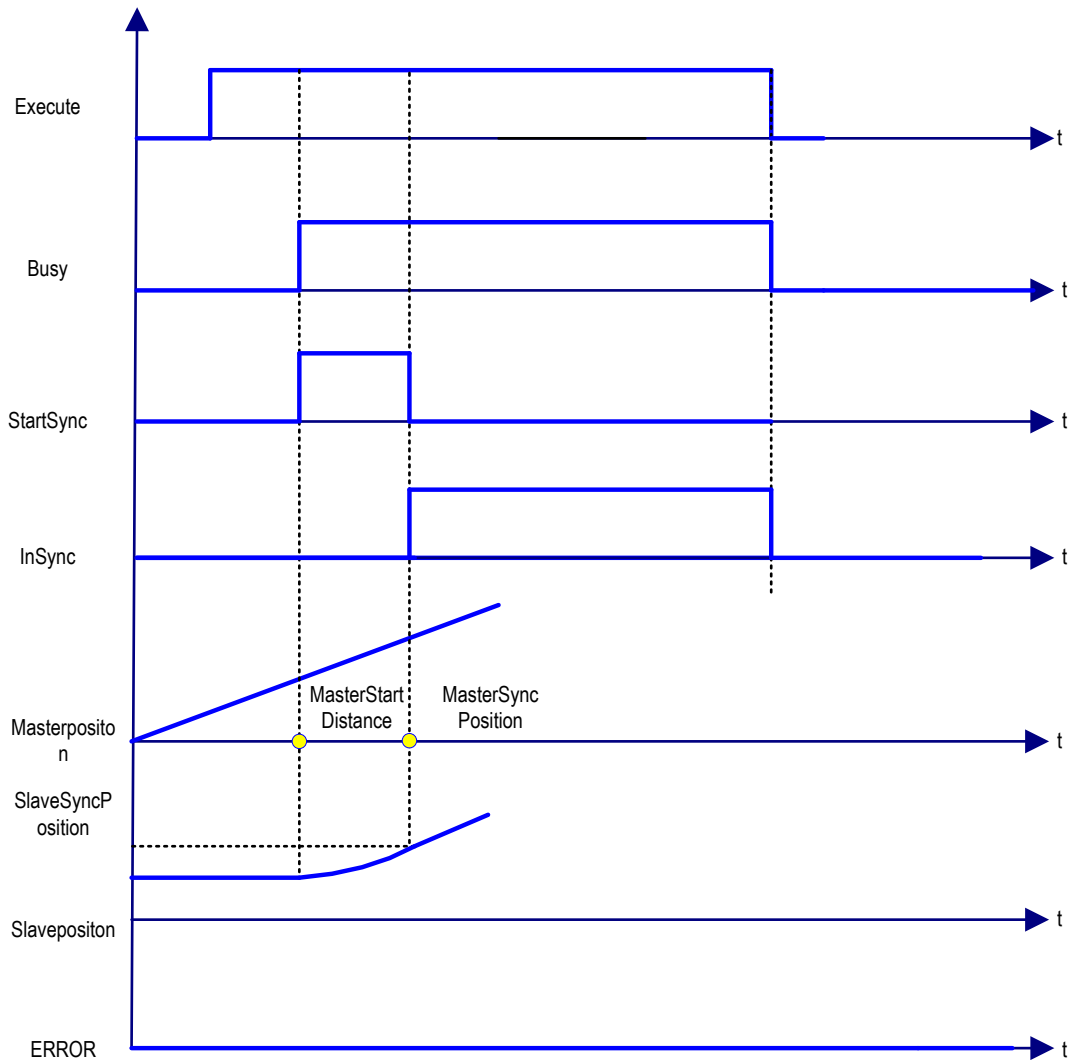
- ③ MasterSyncPosition=300、MasterStartDistance=50、SlaveSyncPosition=60，主轴速度为 50、AvoidReversal=FALSE、从轴起始位置大于 60



同步完成（InSync 为 TRUE）的同时达到目标速度，此后从轴移动量 = 主轴移动量 * RatioNumerator/ RatioDenominator。

对于 AvoidReversal: 如果从轴是模态轴并且主轴速度（齿轮比的倍数关系）不是相对于从轴的速度关系，那么 MC_GearInPos 会尝试着避免从轴的反转。它试图通过增加 5 个从轴周期“拉伸”从轴的运动。如果这个“拉伸”无效，那么会有错误出现并且从轴错误停止。如果从轴速度关联主轴速度（是齿轮比的倍数），那么会有错误出现，并且轴错误停止。如果从轴是线性轴模式轴，那么一个错误会产生在 Execute 输入上升沿处理时。

4) 时序图



5) 错误说明

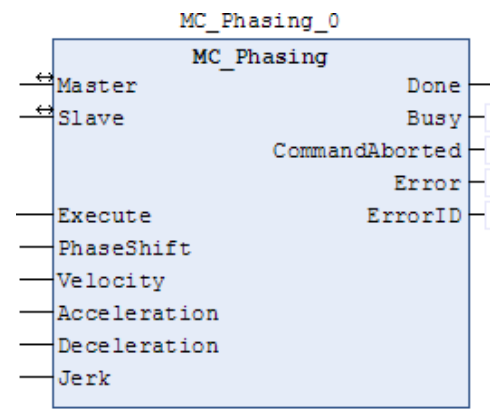
- ◆ 相关参数设置有错，会导致指令报警。
- ◆ 轴没有使能会导致指令报警。

【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。

MC_Phasing

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		<pre>MC_Phasing0(Master:= , Slave:= , Execute:= , PhaseShift:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);</pre>

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- FAKSE	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
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**。

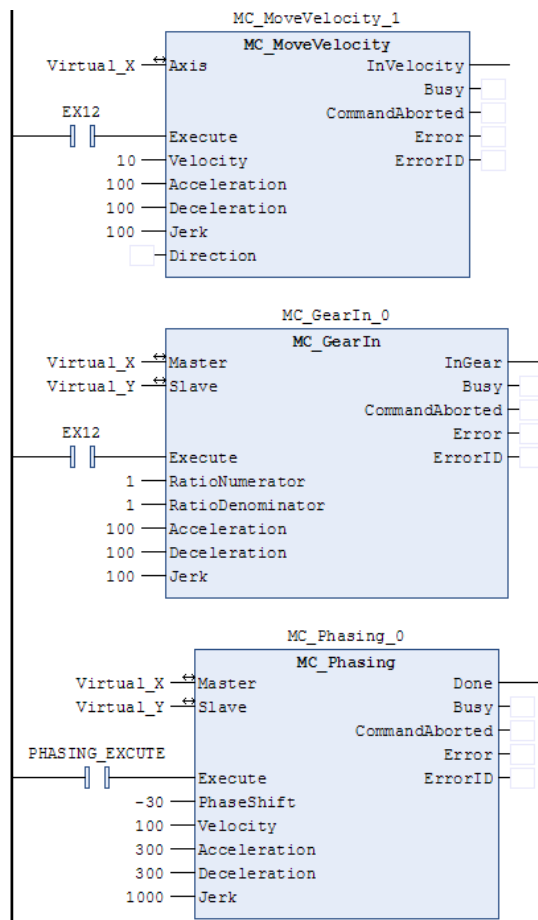
根据设定的 **PhaseShift**、**Velocity**、**Acceleration**、**Deceleration** 对主从轴相位差进行补偿。

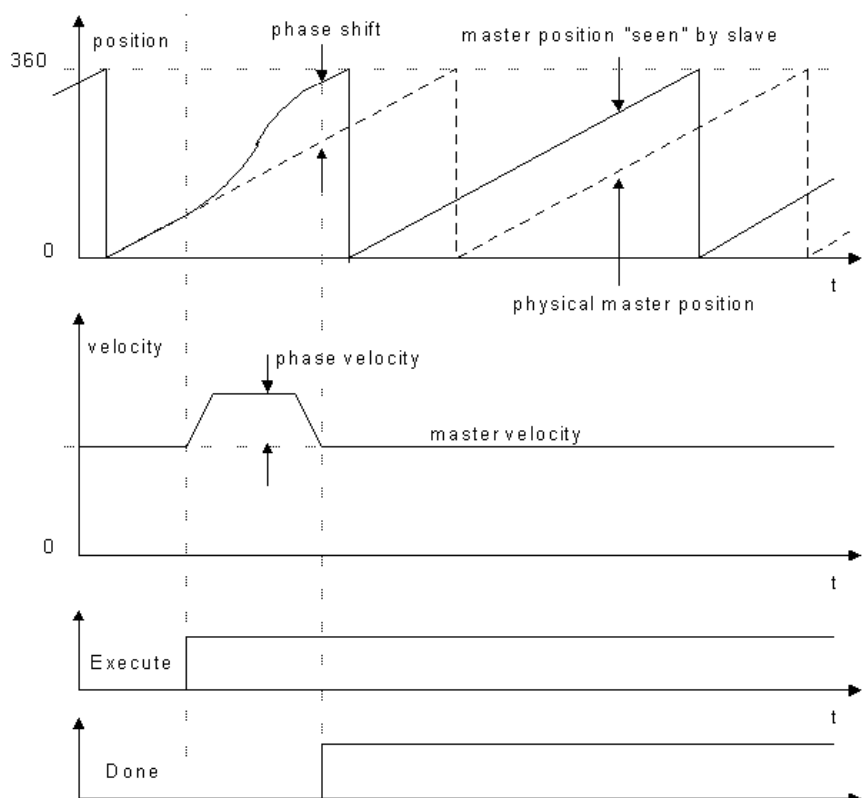
主从轴相位差达到 **PhaseShift** 时，**Done** 信号输出。

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。

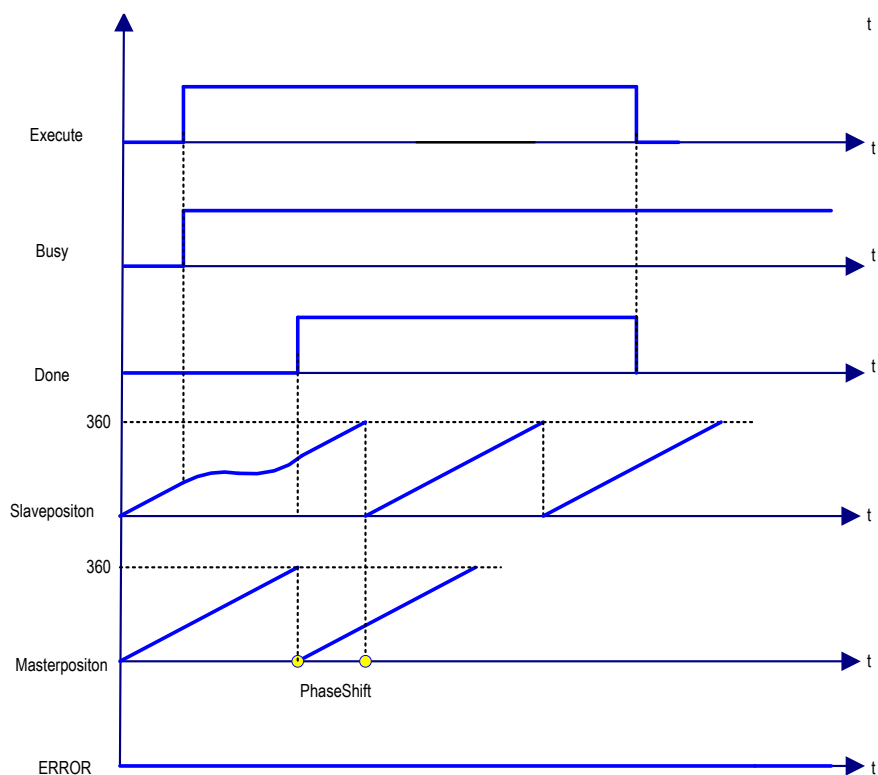
该指令跟 **MC_GearIn** 指令配合使用，如下：主轴为 **Virtual_x**，从轴为 **Virtual_y**，EX12 上升沿执行主轴速度控制以及主从轴电子齿轮动作，然后执行相位偏移。此外可以跟电子凸轮配合使用，此时从轴作为“电子凸轮主轴”以达到电子凸轮主轴相位偏移的效果。





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。

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		<pre> SMC_CAMBounds_0(CAM:= , bExecute:= , dMasterVelMax:= , dMasterAccMax:= , dMasterScaling:= , dSlaveScaling:= , bDone=> , bBusy=> , bError=> , nErrorID=> , dMaxPos=> , dMinPos=> , dMaxVel=> , dMinVel=> , dMaxAccDec=> , dMinAccDec=>); </pre>

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- FAKSE	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

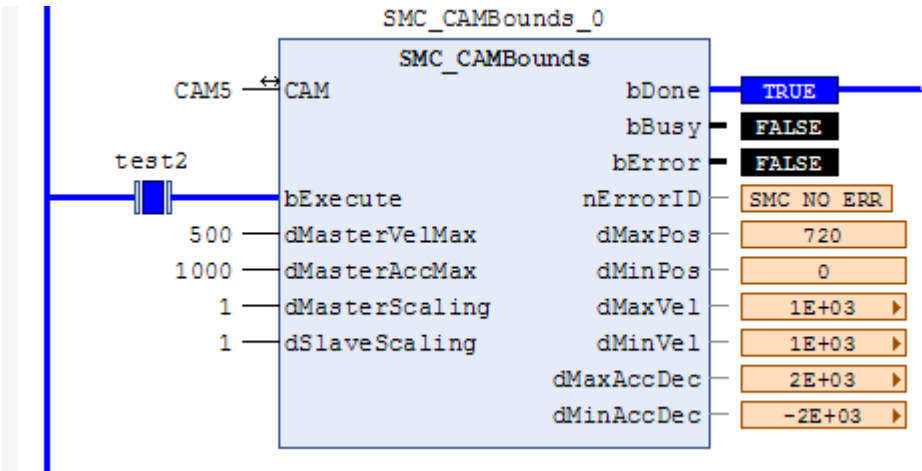
输出变量	名称	数据类型	有效范围	初始值	描述
bDone	完成	BOOL	TRUE- FALSE	FALSE	True, 如果计算完成
bBusy	指令处理中	BOOL	TRUE,FALSE	FALSE	True, 指令正在处理中
bError	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
nErrorID	错误代码	SMC_ERROR	参阅 SMC_ERROR	0	异常发生时，输出错误代码
dMaxPos	最大位置	LREAL		0	根据凸轮表计算出从轴最大位置
dMinPos	最小位置	LREAL		0	根据凸轮表计算出从轴最小位置
dMaxVel	最大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

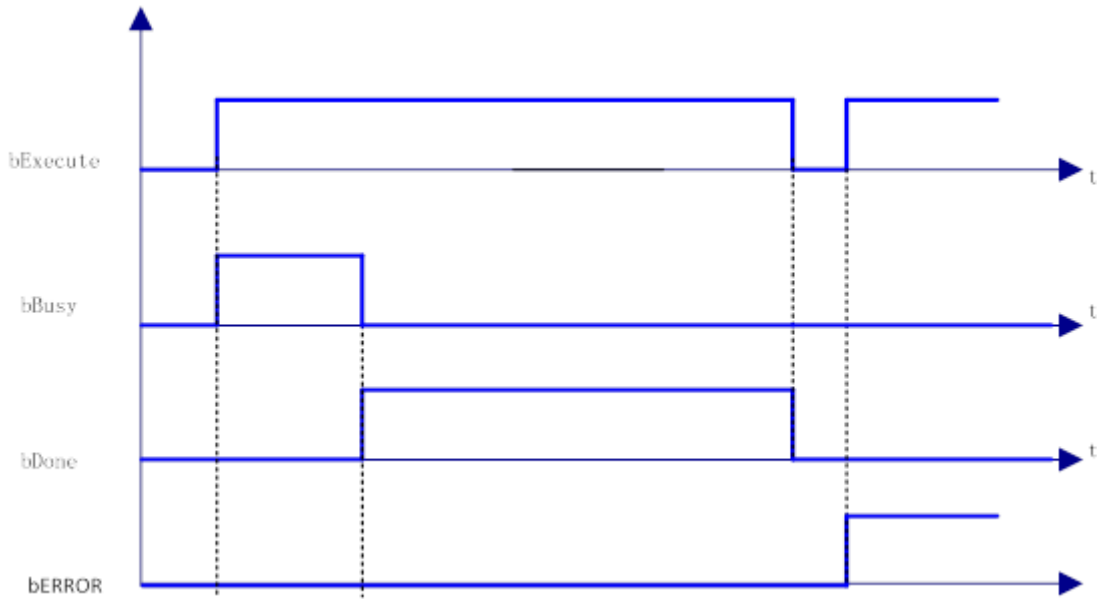
bExecute 上升沿来综合输入变量的“dMasterVelMax”，“dMasterAccMax”，“dMasterScaling”，“dSlaveScaling”值与凸轮表数据计算出从轴“最大位置”最小位置等值。例如：主轴周期 360，凸轮表为一条斜率为 2 的直线，计算出的结果如下图所示：

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。

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		<pre>SMC_CAMBounds_Pos0(CAM:= , bExecute:= , dMasterVelMax:= , dMasterAccMax:= , dMasterScaling:= , dSlaveScaling:= , bDone=> , bBusy=> , bError=> , nErrorID=> , dMaxPos=> , dMinPos=>);</pre>

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- FAKSE	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。

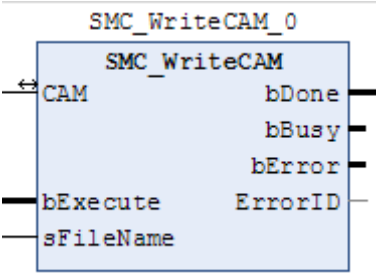
【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

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。

1) 指令格式

指令	名称	图形表现	ST 表现
SMC_WriteCAM	凸轮上下限		

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

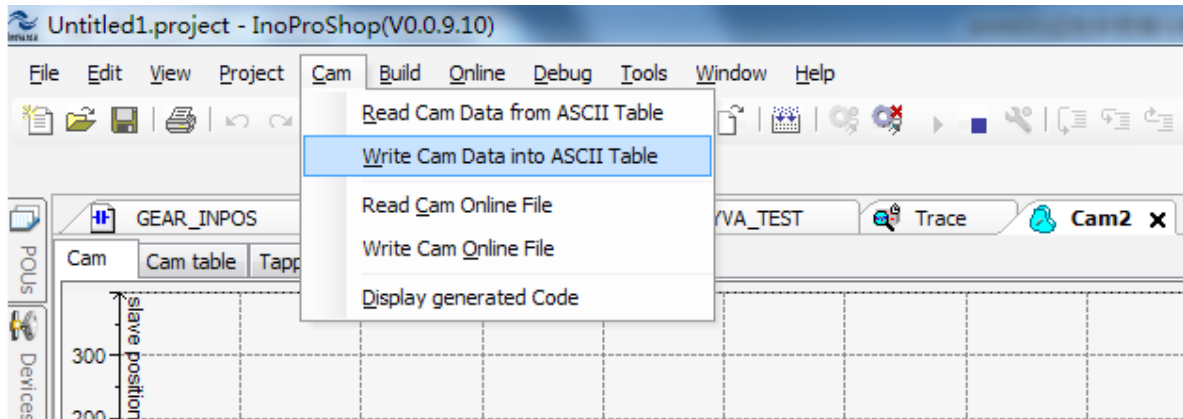
输入变量	名称	数据类型	有效范围	初始值	描述
bExecute	指令执行	BOOL	TRUE- FAKSE	FALSE	上升沿，开始执行指令
sFileName	文件名	STRING		“	包含凸轮描述的以 ASCII 格式定义的文件名，可以通过帮助里面” Cam Format” 来查看具体描述。

◆ 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

3) 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



4) 错误说明

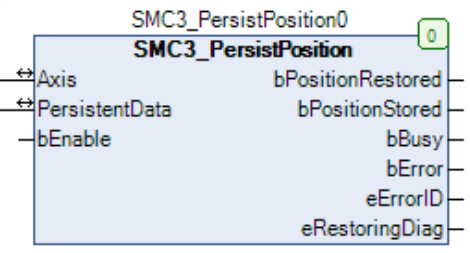
- ◆ 该指令只能完成 XYVA 多项式模式的凸轮表，一维，二维等会造成错误输出
- ◆ sFileName 连接的文件名不存在或者信息错误。

【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。

SMC3_PersistPosition

该指令用来保持记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		<pre>SMC3_PersistPosition0(Axis:= PersistentData:= bEnable:= bPositionRestored=> bPositionStored=> bBusy=> bError=> eErrorID=> eRestoringDiag=>);</pre>

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
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

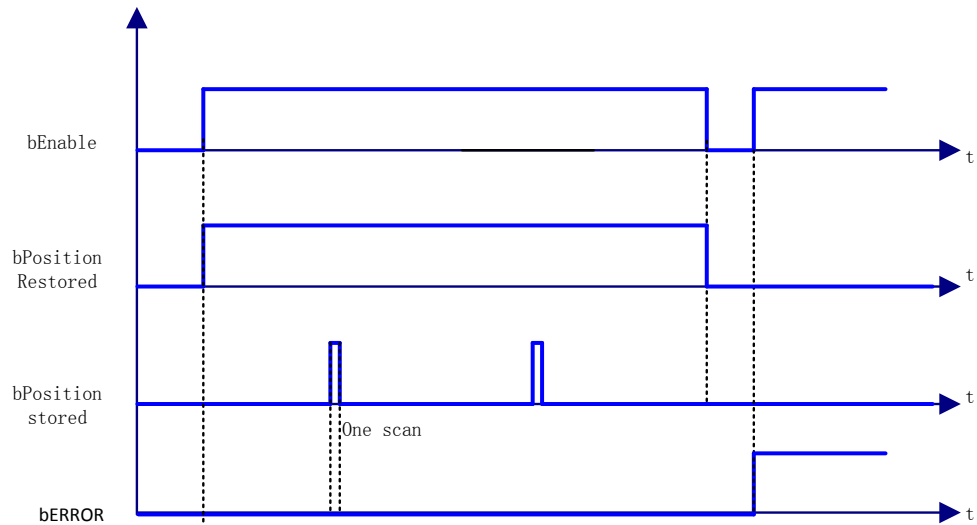
输入变量	名称	数据类型	有效范围	初始值	描述
bEnable	执行	BOOL	TRUE,FALSE	FALSE	True 功能块执行，false 不执行功能块 若要在初始化期间还原上次存储的位置，则必须从应用程序启动时将该值置为 true

◆ 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_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。
- ◆ 不支持虚轴跟逻辑轴。
- ◆ 时序图



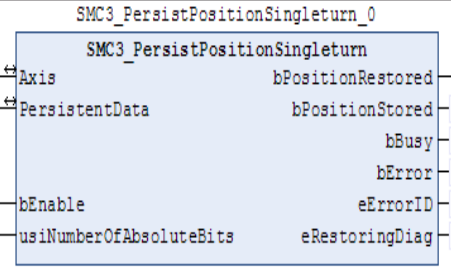
4) 错误说明

- ◆ 输入轴为虚拟轴或者逻辑轴会导致错误输出。
 - ◆ 轴有错误。
- 【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。

SMC3_PersistPositionSingleturn

该指令用来保持纪录实轴单圈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）。

1) 指令格式

指令	名称	图形表现	ST 表现
SMC3_PersistPositionSingleturn	轴位置保持		<pre>SMC3_PersistPositionSingleturn_0(Axis:= , PersistentData:= , bEnable:= , usiNumberOfAbsoluteBits:= , bPositionRestored=> , bPositionStored=> , bBusy=> , bError=> , eErrorID=> , eRestoringDiag=>);</pre>

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
PersistentData	保持数据	SMC3_PersistPositionSingletrun_Data			映射到记录位置结构，为 SMC3_PersistPosition_Data 的一个映射（该结构变量必须为断电保持型）

◆ 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 PLC 重启后需要为 true 才能恢复重启前存储的位置。
usiNumberOfAbsoluteBites	位数	UINT		16	多少位的绝对值编码器（如 20 位，24 位编码器等）

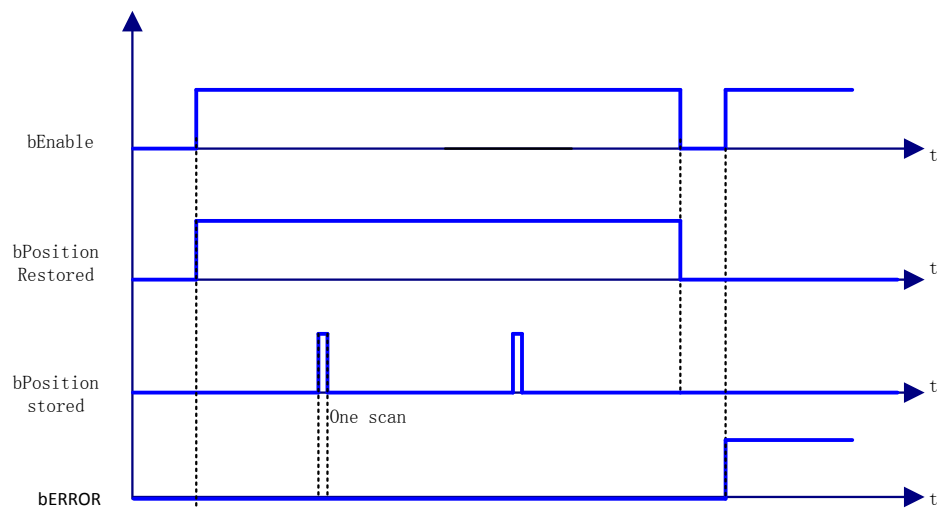
◆ 输出变量

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
eRestoringDiag	恢复诊断	SMC3_PersistPositionDiag		SMC3_PersistPositionDiag. SMC3_PPD_RESTORING_OK	位置恢复中的诊断信息

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。

◆ 时序图



4) 错误说明

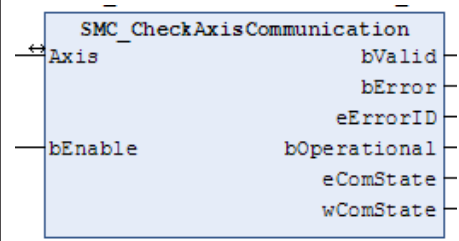
输入轴为虚拟轴或者逻辑轴会导致错误输出。
轴有错误。

【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。

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		<pre>SMC_CheckAxisCommunication0(Axis:= , bEnable:= , bValid=> , bError=> , eErrorID=> , bOperational=> , eComState=> , wComState=>);</pre>

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

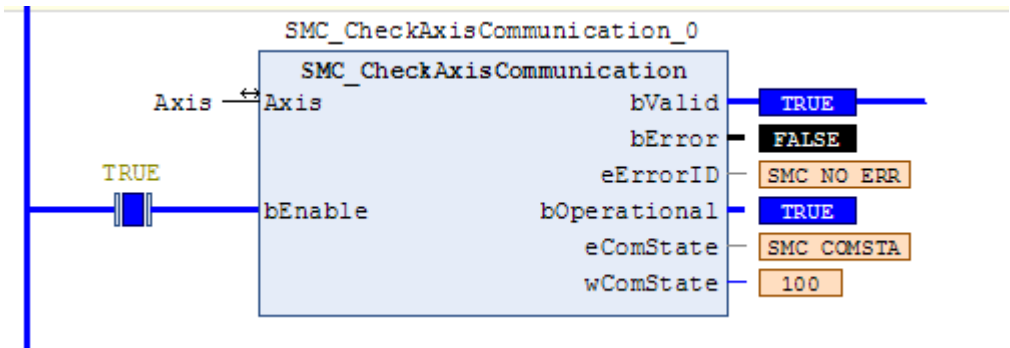
输出变量	名称	数据类型	有效范围	初始值	描述
bValid	执行中	BOOL	TRUE,FALSE	FALSE	True, 指令执行有效
bError	错误	BOOL	TRUE,FALSE	FALSE	True, 异常产生
eErrorID	错误代码	SMC_ERROR			参考 SMC_Error
bOperational	通讯正常	BOOL	TRUE,FALSE	FALSE	True, 通讯正常（代码为 100）可操作 False, 通讯不正常，不可对轴操作
eComState	通讯状态	SMC_COMMUNICATIONSTATE			包含： SMC_COMSTATE_NOT_STARTED, 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。

◆ 样例程序



4) 错误说明

bExecute 上升沿时:

轴报错, Error 输出;

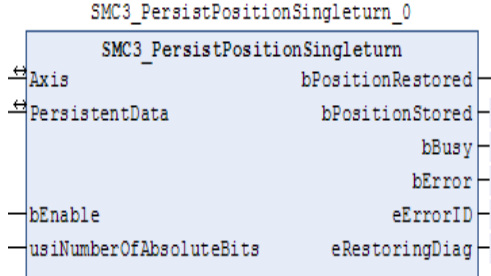
无效的轴输入, Error 输出。

【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。

SMC3_PersistPositionSingleturn

该指令用来保持纪录实轴单圈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）。

1) 指令格式

指令	名称	图形表现	ST 表现
SMC3_PersistPositionSingleturn	轴位置保持		<pre>SMC3_PersistPositionSingleturn_0(Axis:= PersistentData:= bEnable:= usiNumberOfAbsoluteBits:= bPositionRestored=> bPositionStored=> bBusy=> bError=> eErrorID=> eRestoringDiag=>);</pre>

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
PersistentData	保持数据	SMC3_PersistPositionSingleturn_Data			映射到记录位置结构，为 SMC3_PersistPosition_Data 的一个映射 (该结构变量必须为断电保持型)

◆ 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 PLC 重启后需要为 true 才能恢复重启前存储的位置
usiNumberOfAbsoluteBites	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

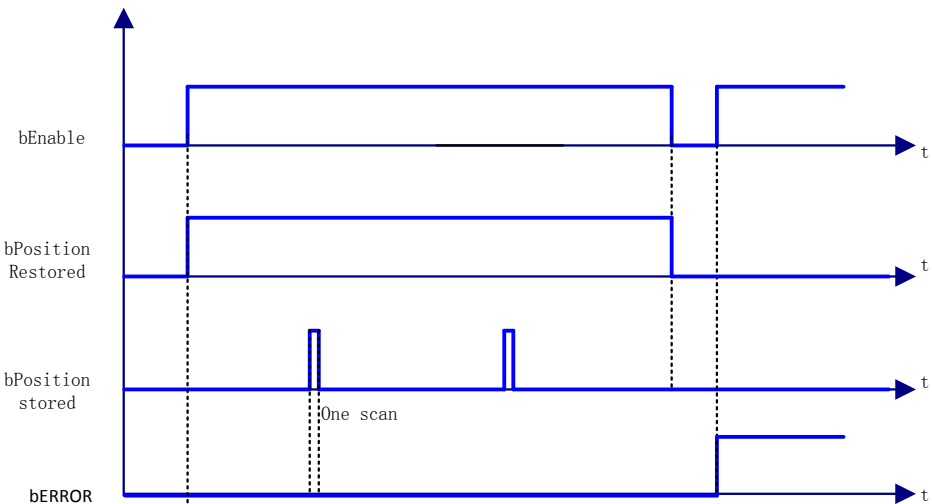
输出变量	名称	数据类型	有效范围	初始值	描述
bPositionRestored	位置恢复	BOOL	TRUE,FALSE	FALSE	TRUE, 轴重启后位置恢复完成
bPositionStored	位置保存	BOOL	TRUE,FALSE	FALSE	TRUE, 调用功能块后保存位置完成
bBusy	FB 执行中	BOOL	TRUE,FALSE	FALSE	TRUE, 功能块没有执行完成
bError	错误	BOOL	TRUE,FALSE	FALSE	TRUE, 异常发生
eErrorID	错误代码	SMC_ERROR		SMC_NO_ERROR	异常发生时，输出错误代码
eRestoringDiag	恢复诊断	SMC3_PersistPositionDiag		SMC3_PersistPositionDiag. SMC3_PPD_RESTORING_OK	位置恢复中的诊断信息

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。

◆ 时序图



4) 错误说明

输入轴为虚拟轴或者逻辑轴会导致错误输出；

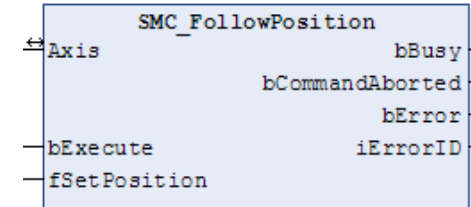
轴有错误。

【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。

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		<pre>SMC_FollowPosition_0(Axis:= , bExecute:= , fSetPosition:=SET_POSITION , bBusy=> , bCommandAborted=> , bError=> , iErrorID=>);</pre>

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

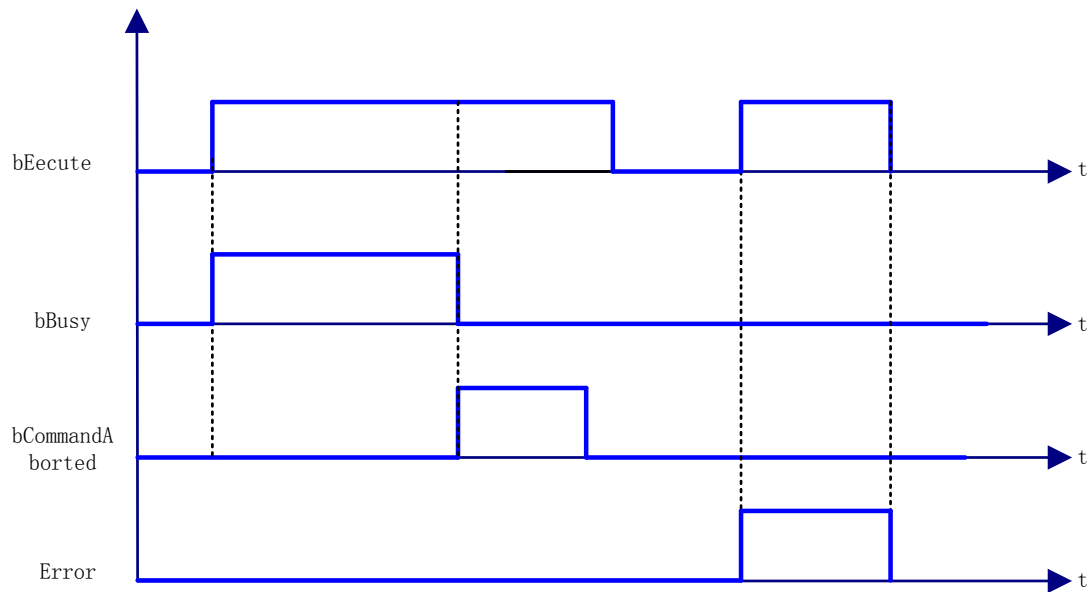
SMC_FollowPosition 通过 bExecute 的上升沿启动之后，轴会每个任务周期给轴发送位置指令。

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： $\Delta L / \Delta t$ ， 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。

◆ 时序图



4) 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；

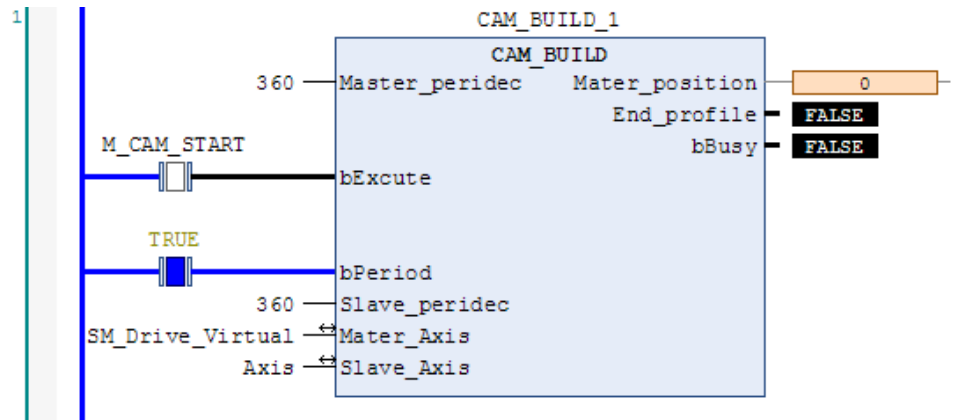
轴没使能，Error 输出。

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。

5) 样例说明

使用 SMC_FollowPosition 实现电子凸轮功能。



功能块变量定义部分：

```
FUNCTION_BLOCK CAM_BUILD
VAR_INPUT// 输入变量定义
    Master_peridec:REAL; // 主轴周期
    bExcute:BOOL; // 指令执行
    bPeriod:BOOL; // 凸轮周期执行，false 单周期执行
    Slave_peridec:REAL; // 从轴周期
END_VAR
VAR_OUTPUT// 输出变量定义
    Mater_position:LREAL;// 主轴位置（指令执行开始后计算的主轴位置）
    End_profile:BOOL; // 曲线完成输出标志位
    bBusy:BOOL; // 执行中
END_VAR
VAR// 功能块中间变量定义
    SMC_FollowPosition_0: SMC_FollowPosition;
    SET_POSITION: LREAL;
    SET_POSITIONOLD: LREAL;
    Mater_positionOLD:LREAL;
    bExcute_old:BOOL;
    INC:LREAL;
    Y:LREAL;
    X5:LREAL;
    X4:LREAL;
    X3:LREAL;
    X2:LREAL;
    X1:LREAL;
    MC_Stop0: MC_Stop;
    STOP:BOOL;
    COUNTNUM:DINT;
    SET_INC:LREAL;
    YOLD:LREAL;
    SMC_FollowPositionVelocity_0: SMC_FollowPositionVelocity;
    K:REAL;
    K_OUT:REAL;
    MC_CamOut_0: MC_CamOut;
END_VAR
VAR_IN_OUT// 输入输出变量定义
    Mater_Axis:AXIS_REF_SM3;
    Slave_Axis:AXIS_REF_SM3;
END_VAR
程序部分：
IF bExcute AND NOT bExcute_old THEN // 上升沿初始化参数
    Mater_position:=0;
    Mater_positionOLD:=Mater_Axis.fActPosition;
    End_profile:=FALSE;
    SET_POSITION:=Slave_Axis.fActPosition;
    SET_POSITIONOLD:=Slave_Axis.fActPosition;
    COUNTNUM:=0;
    YOLD:=0;
    K:=0;
ELSE
```

```

IF bExcute_old THEN
    INC:=Mater_Axis.fActPosition-Mater_positionOLD;// 主轴任务周期增量
    IF INC<0 THEN // 主轴编码位置过零点（轴设置为 modulo- 模数模式时）
        INC:=Mater_Axis.fActPosition-Mater_positionOLD+Mater_Axis.fPositionPeriod;
    END_IF
    Mater_position:=INC+Mater_position;// 当前主轴位置
    Mater_positionOLD:=Mater_Axis.fActPosition;
    //***** 判断曲线完成 *****//
    IF Mater_position>=Master_peridec THEN
        End_profile:=TRUE;
    ELSE
        End_profile:=FALSE;
    END_IF
    IF bPeriod THEN
        IF Mater_position>=Master_peridec THEN
            Mater_position:=Mater_position-Master_peridec;
        END_IF
    END_IF
END_IF
END_IF
IF bExcute_old THEN
    X1:=(Mater_position/Master_peridec);
    X2:=X1*X1;
    X3:=X2*X1;
    X4:=X3*X1;
    X5:=X4*X1;
    Y:=(6*X5-15*X4+10*X3)*Slave_peridec;// 从轴位置，曲线
    K:=(30*X4-60*X3+30*X2)*Slave_peridec/Master_peridec;// 曲线斜率
    SET_INC:=Y-YOLD;
    IF SET_INC<0 THEN
        SET_INC:=Slave_peridec-YOLD+Y;
    END_IF
    YOLD:=Y;
    IF bPeriod THEN
        SET_POSITION:=SET_POSITION+SET_INC;
    ELSE
        IF End_profile THEN
            SET_POSITION:=SET_POSITIONOLD+Slave_peridec;
        ELSE
            SET_POSITION:=SET_POSITION+SET_INC;
        END_IF
    END_IF
    IF SET_POSITION>=Slave_Axis.fPositionPeriod THEN
        SET_POSITION:=SET_POSITION-Slave_Axis.fPositionPeriod;
    END_IF
END_IF
SMC_FollowPosition_0(
    Axis:=Slave_Axis,
    bExecute:=bExcute,
    fSetPosition:=SET_POSITION ,
    bBusy=>bBusy ,
    bCommandAborted=> ,

```

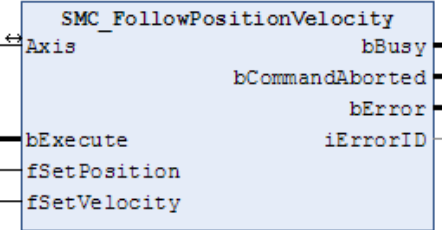
```
        bError=> ,
        iErrorID=> );
    IF NOT bExcute AND bExcute_old THEN
        STOP:=TRUE;
    END_IF
    MC_CamOut_0(
        Slave:=Slave_Axis,
        Execute:= STOP,
        Done=> ,
        Busy=> ,
        Error=> ,
        ErrorID=> );
    MC_Stop0(
        Axis:=Slave_Axis,
        Execute:= MC_CamOut_0.Done OR MC_CamOut_0.Error ,
        Deceleration:=20000 ,
        Jerk:= 20000,
        Done=> ,
        Busy=> ,
        Error=> ,
        ErrorID=> );
    IF MC_Stop0.Done OR MC_Stop0.Error THEN
        STOP:=FALSE;
    END_IF
    IF NOT bExcute_old THEN
        End_profile:=FALSE;
    END_IF
    bExcute_old:=bExcute;
```

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		<pre>SMC_FollowPositionVelocity_0(Axis:= bExecute:= , fSetPosition:= , fSetVelocity:= , bBusy=> bBusy, bCommandAborted=> , bError=> , iErrorID=>);</pre>

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

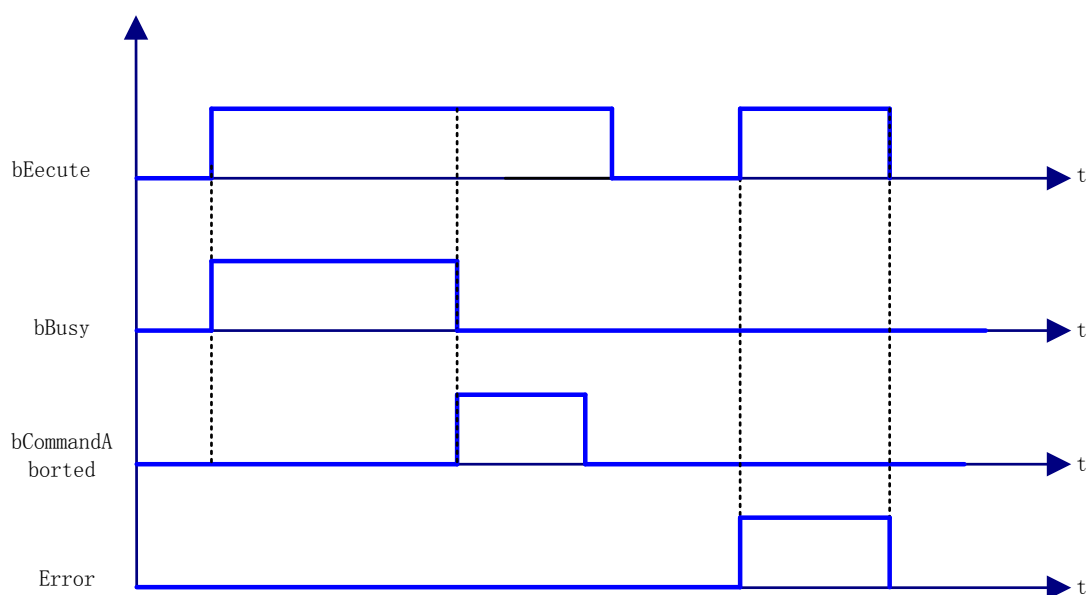
SMC_FollowPositionVelocity 通过 bExecute 的上升沿启动之后，轴会每个任务周期给轴发送设定位置、设定速度指令。

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： $fSetVelocity = \Delta L / \Delta t$ ， 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。

◆ 时序图



4) 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；

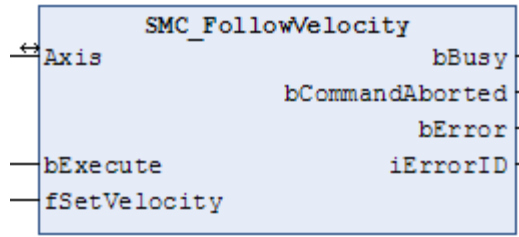
指令运行中，轴出错，Error 输出。

【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。

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		<pre>SMC_FollowVelocity_0(Axis:= , bExecute:= , fSetVelocity:= , bBusy=> , bCommandAborted=> , bError=> , iErrorID=>);</pre>

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	设定位置	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

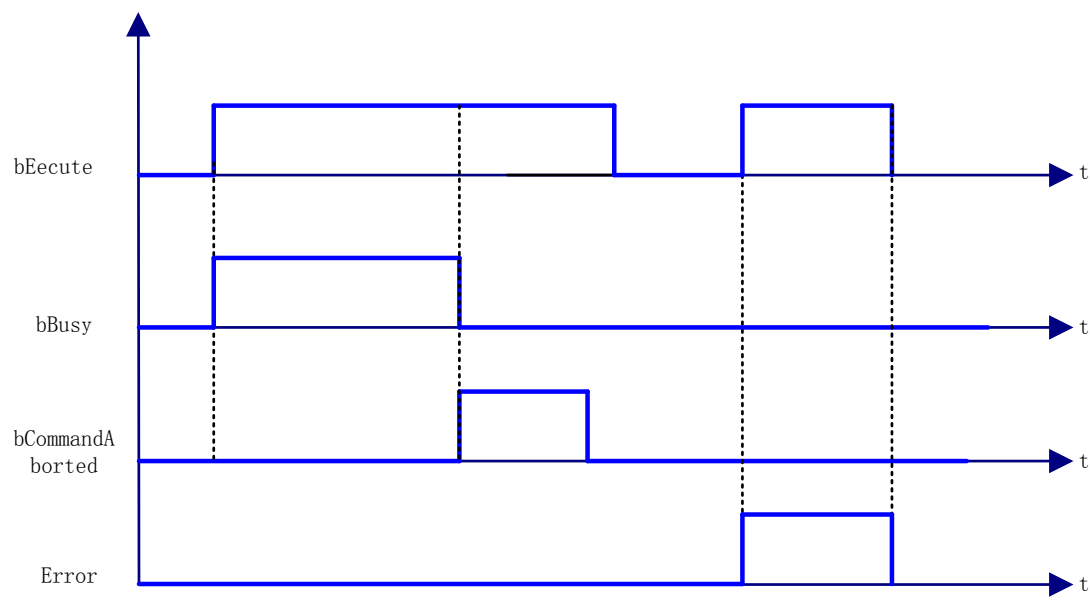
3) 功能说明

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。

◆ 时序图



4) 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。

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		<pre> SMC_FollowSetValues_0(Axis:= , bExecute:= , dwValueMask:= , fSetPosition:= , fSetVelocity:= , fSetAcceleration:= , fSetJerk:= , fSetTorque:= , fSetCurrent:= , bBusy=> , bCommandAborted=> , bError=> , iErrorID=>); </pre>

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

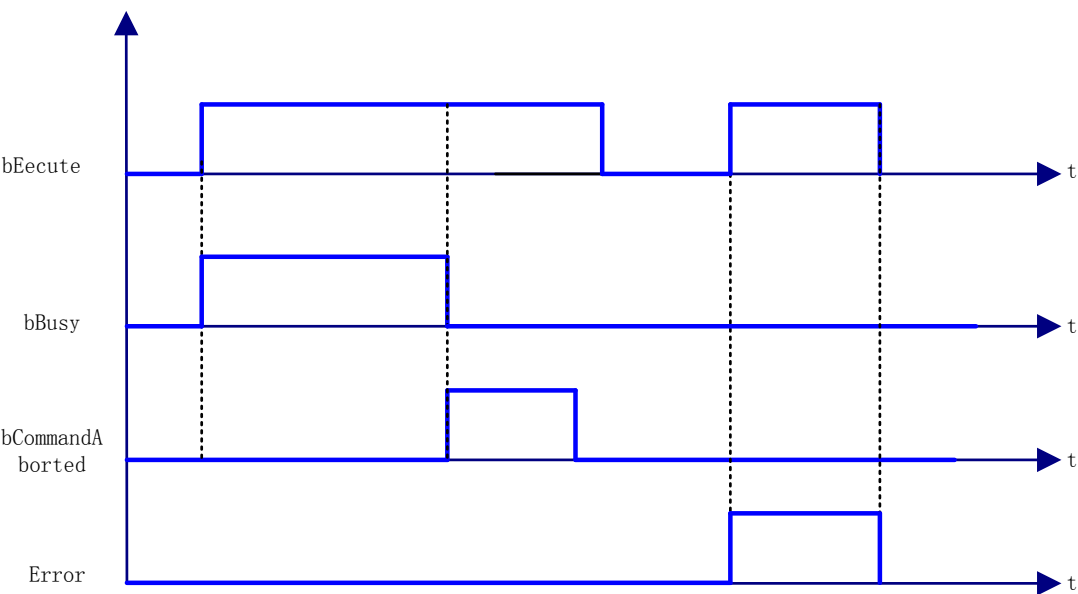
输入变量	名称	数据类型	有效范围	初始值	描述
bExecute	执行	BOOL	TRUE,FALSE	FALSE	上升沿执行功能块
DwValueMask	控制管理	DWORD		0	Bite0:TRUE:fSetPosition 激活 FALSE: 忽略 Bite1:TRUE: fSetVelocity 激活 FALSE: 忽略 Bite2:TRUE: fSetAcceleration 激活 FALSE: 忽略 Bite3:TRUE: fSetJerk 激活 FALSE: 忽略 Bite4:TRUE: fSetTorque 激活 FALSE: 忽略 Bite5:TRUE: fSetCurrent 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 /s ² ）
fSetJerk	设定跃度值	LREAL		0	轴设定的跃度值（标定好的单位 /s ³ ）
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	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_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。
- ◆ 时序图



4) 错误说明

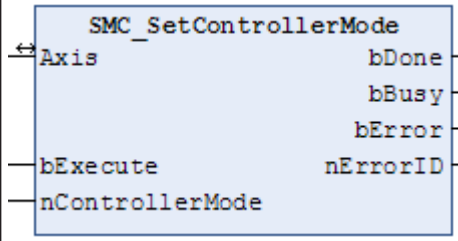
bExecute 上升沿时：
Axis 变量连接的为非 AXIS_REF_SM3 类型结构变量，Error 输出
轴没使能，Error 输出。
指令运行中，轴出错，Error 输出。

【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。

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		<pre>SMC_SetControllerMode0(Axis:= , bExecute:= , nControllerMode:= , bDone=> , bBusy=> , bError=> , nErrorID=>);</pre>

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）。

PDO Assignment (16#1C12):

☒ 16#1600

☐ 16#1701 (excluded by 16#1600)

☐ 16#1702 (excluded by 16#1600)

☐ 16#1703 (excluded by 16#1600)

☐ 16#1704 (excluded by 16#1600)

☐ 16#1705 (excluded by 16#1600)

+

Insert

Edit

Delete

Move Up

Move Down

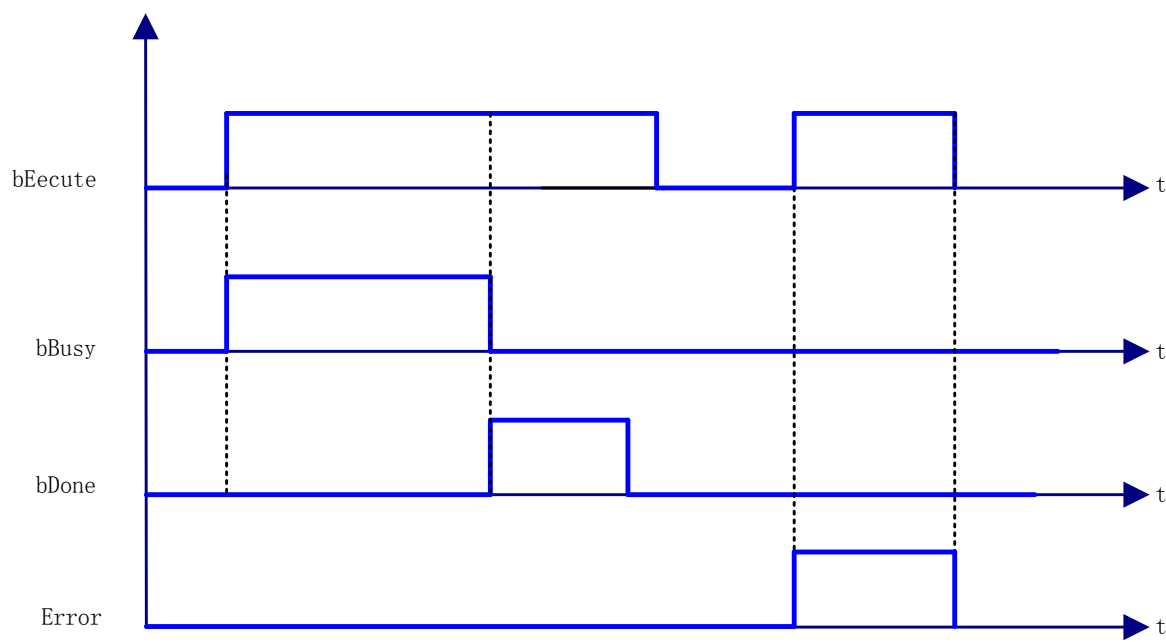
PDO Content (16#1600):

Index	Size	Offs	Name	Type
16#6040:00	2.0	0.0	Controlword	UINT
16#607A:00	4.0	2.0	Targetposition	DINT
16#60B8:00	2.0	6.0	Touch probe function	UINT
16#6071:00	2.0	8.0	Target torque	INT
16#60FF:00	4.0	10.0	Target velocity	DINT
16#6060:00	1.0	14.0	Modes of operation	SINT

15.0

- ◆ 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（参考《IS620N 系列伺服设计维护手册》7.3.3 各模式支持通讯周期）
 - 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。

4) 时序图



5) 错误说明

bExecute 上升沿时:

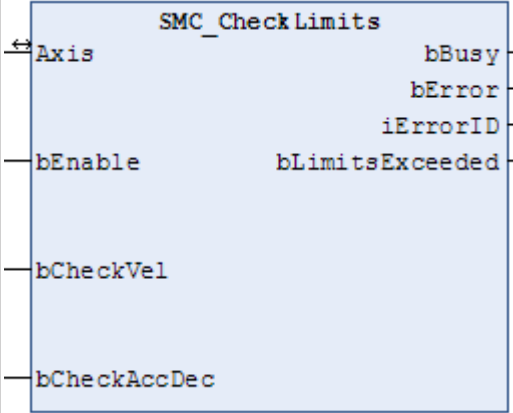
- 轴无效
- 轴状态无效。
- 轴不满足控制方式。
- 轴报错，Error 输出。

【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。

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		<pre>SMC_CheckLimits0(Axis:= , bEnable:= , bCheckVel:= , bCheckAccDec:= , bBusy=> , bError=> , iErrorID=> , bLimitsExceeded=>);</pre>

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
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

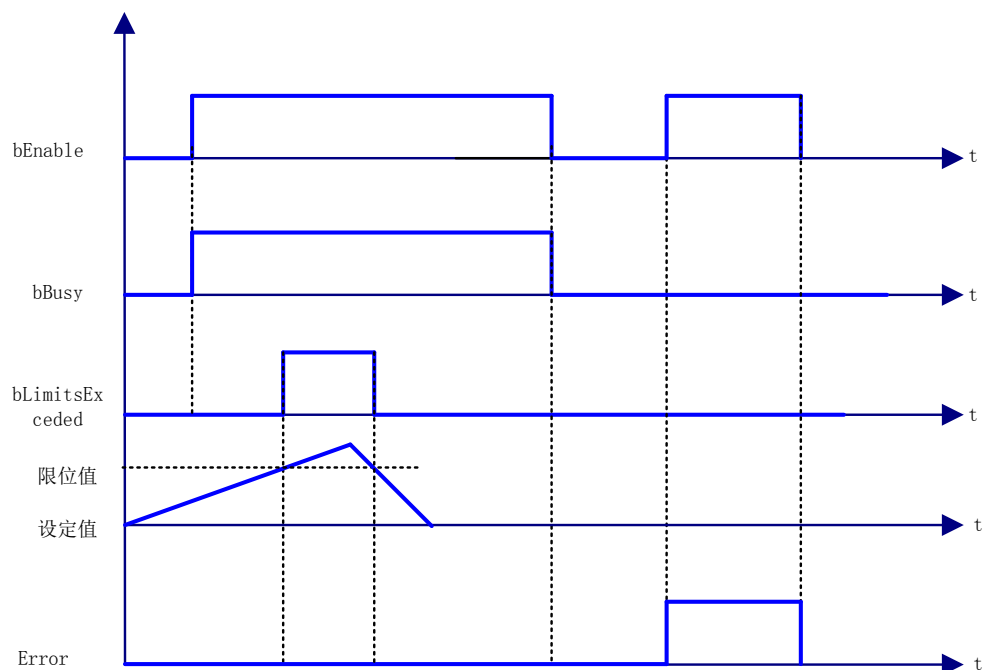
3) 功能说明

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。

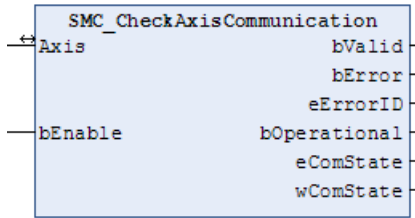
无效的轴输入，Error 输出。

【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。

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		<pre>SMC_CheckAxisCommunication0(Axis:= bEnable:= , bValid=> , bError=> , eErrorID=> , bOperational=> , eComState=> , wComState=>);</pre>

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

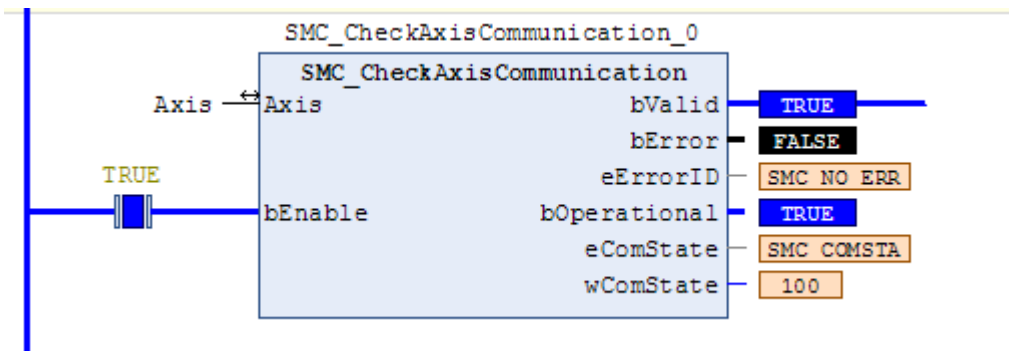
输出变量	名称	数据类型	有效范围	初始值	描述
bValid	执行中	BOOL	TRUE,FALSE	FALSE	True, 指令执行有效
bError	错误	BOOL	TRUE,FALSE	FALSE	True, 异常产生
eErrorID	错误代码	SMC_ERROR			参考 SMC_Error
bOperational	通讯正常	BOOL	TRUE,FALSE	FALSE	True, 通讯正常（代码为 100）可操作 False, 通讯不正常，不可对轴操作
eComState	通讯状态	SMC_ COMMUNI CATIONSTATE			包含： SMC_COMSTATE_NOT_STARTED, 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。

◆ 样例程序



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。

【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。

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		<pre>SMC_GetMaxSetAccDec_0(Axis:= , bEnable:= , dwTimeStamp:= , bValid=> , bBusy=> , fMaxAcceleration=> , dwTimeAtMax=>);</pre>

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）

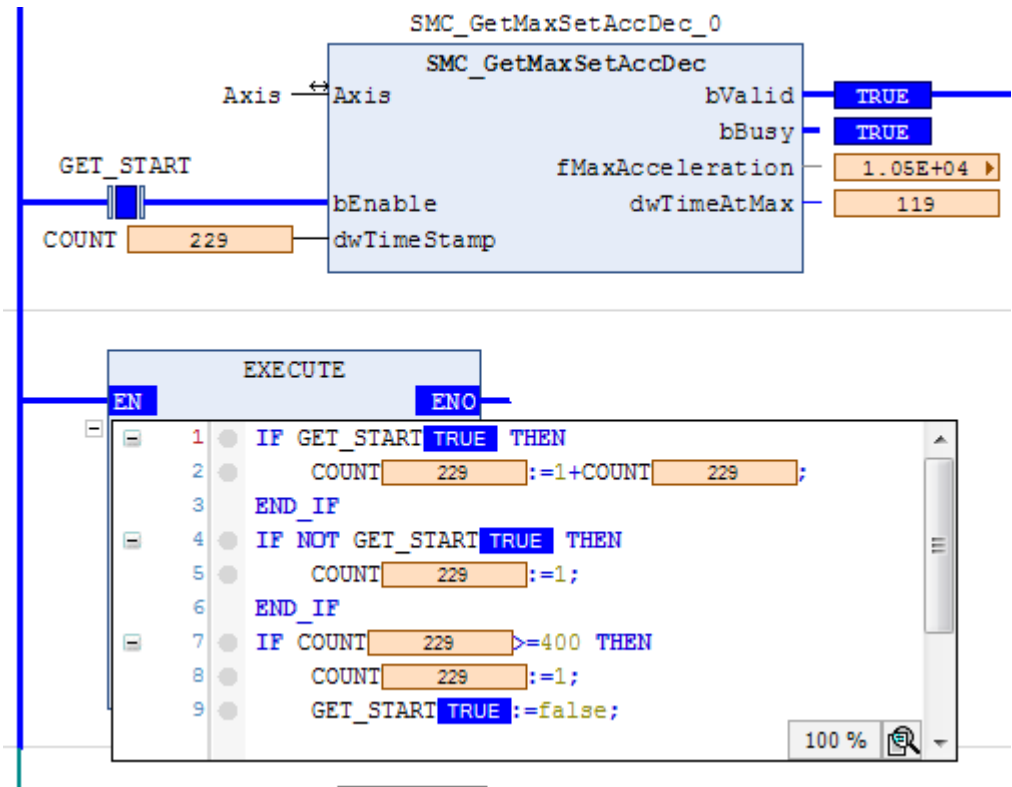
3) 功能说明

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)

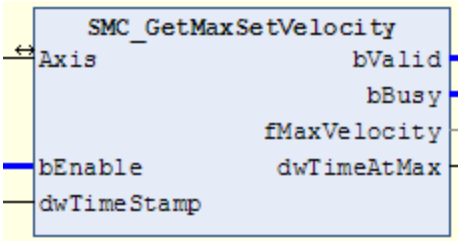
◆ 样例程序



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_GetMaxSetVelocity	轴度最大加减速		<pre>SMC_GetMaxSetVelocity_0(Axis:= , bEnable:= , dwTimeStamp:= , bValid=> , bBusy=> , fMaxVelocity=> , dwTimeAtMax=>);</pre>

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）

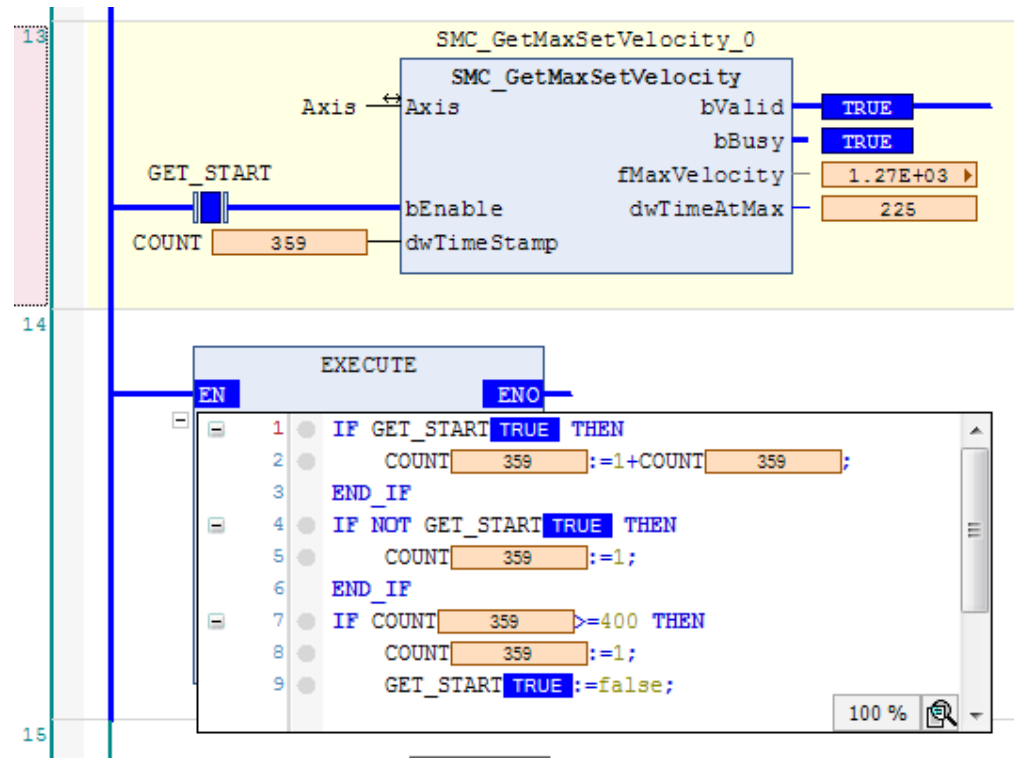
3) 功能说明

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)

◆ 样例程序



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_ GetTrackingError	轴滞后 偏差读取		<pre> SMC_GetTrackingError(Axis:= , bEnable:= , byDeadTimeCycles:= , dwTimeStamp:= , bValid=> , bBusy=> , fActTrackingError=> , fMaxTrackingError=> , dwTimeAtMax=>); </pre>

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

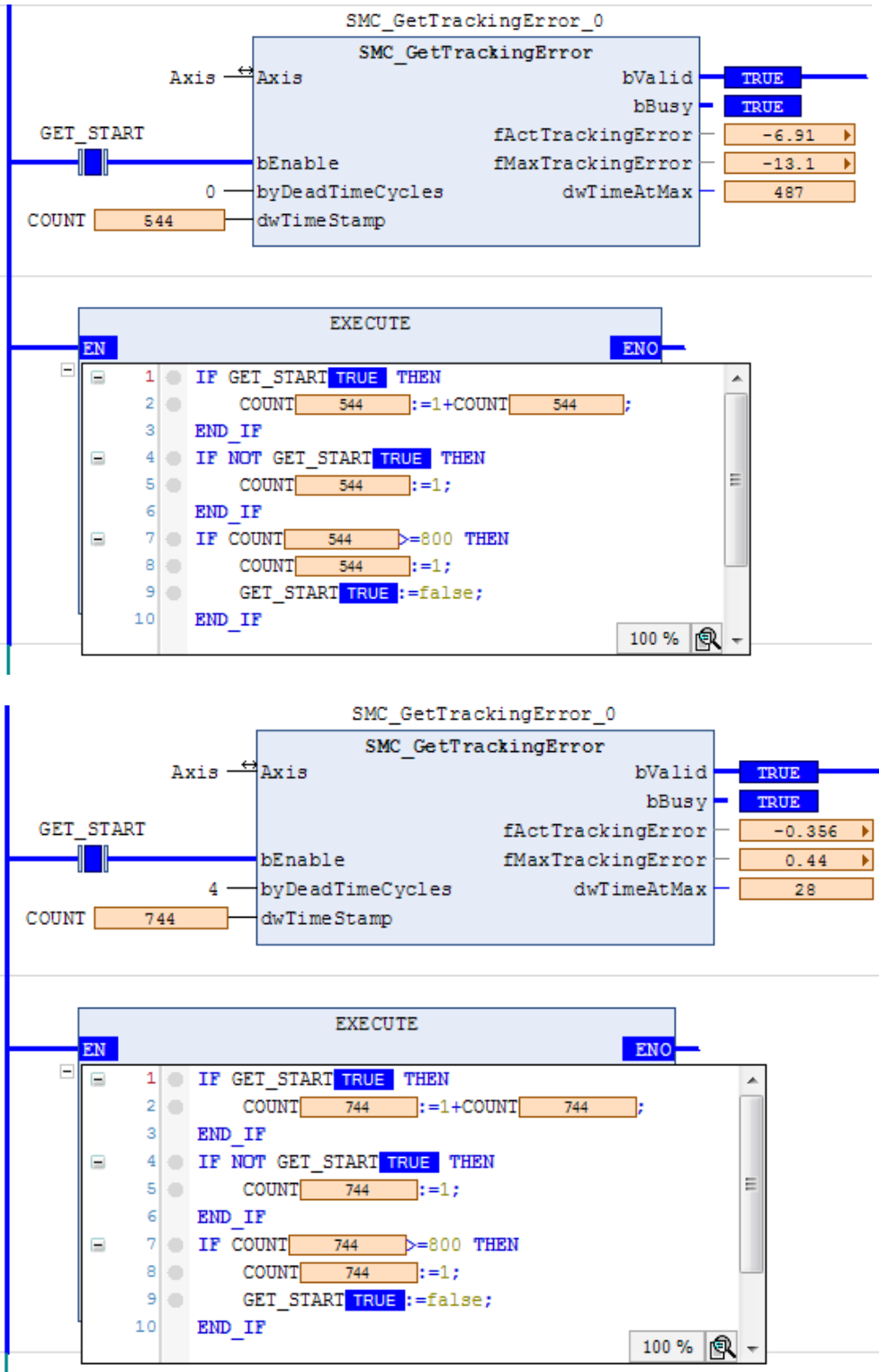
3) 功能说明

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）

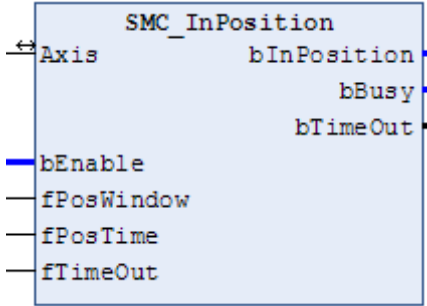
◆ 样例程序



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		<pre>SMC_InPosition0(Axis:=Axis , bEnable:= , fPosWindow:= , fPosTime:= , fTimeOut:= , bInPosition=> , bBusy=> , bTimeOut=>);</pre>

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

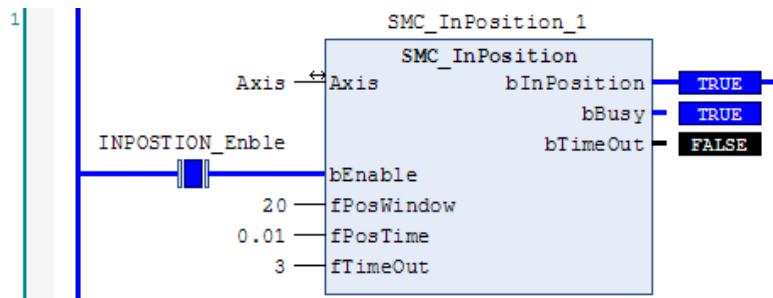
bEnable 为 TRUE，一旦检测到的偏差小于设置的窗口 fPosWindow 持续 fPosTime 秒则 bInPosition 触发为 TRUE。一旦检测到的偏差大于设置的窗口，bInPosition 立马输出为 FALSE。注意：fPosTime 时间设定需合理否则会造成 bTimeOut 触发（比如一个凸轮周期为 2 秒的凸轮，连续偏差没有超过设定窗口的时间为 1.5 秒，fPosTime 如果设定的大于 1.5 秒则会造成 bInPosition 不会触发）。

bEnable 为 TRUE，bBusy 输出为 true。

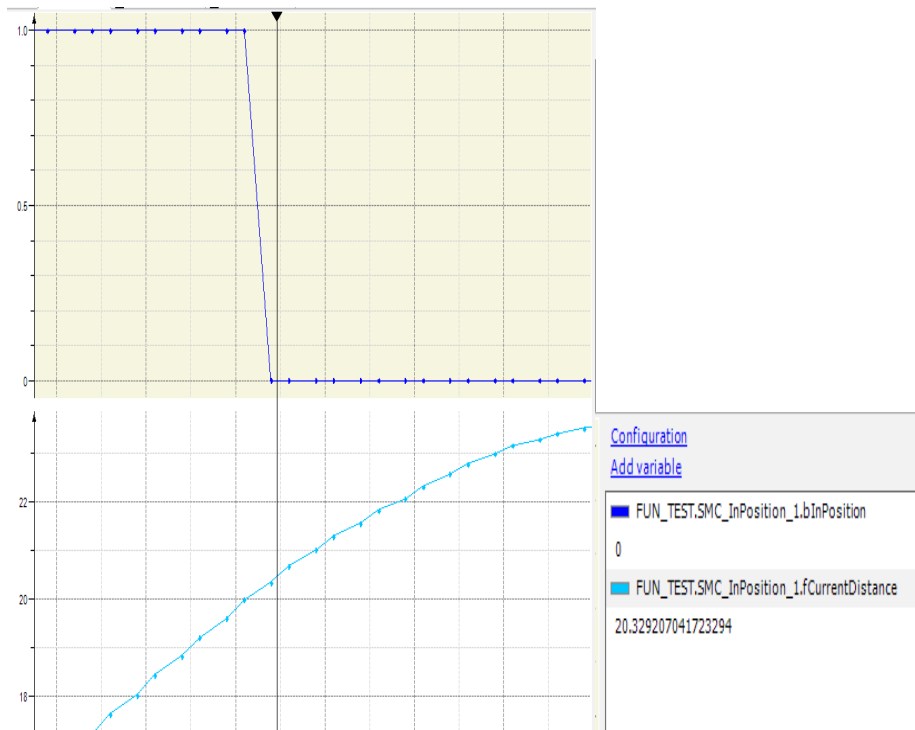
偏差值可监控 SMC_InPosition 结构体中的数据 fCurrentDistance。

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。

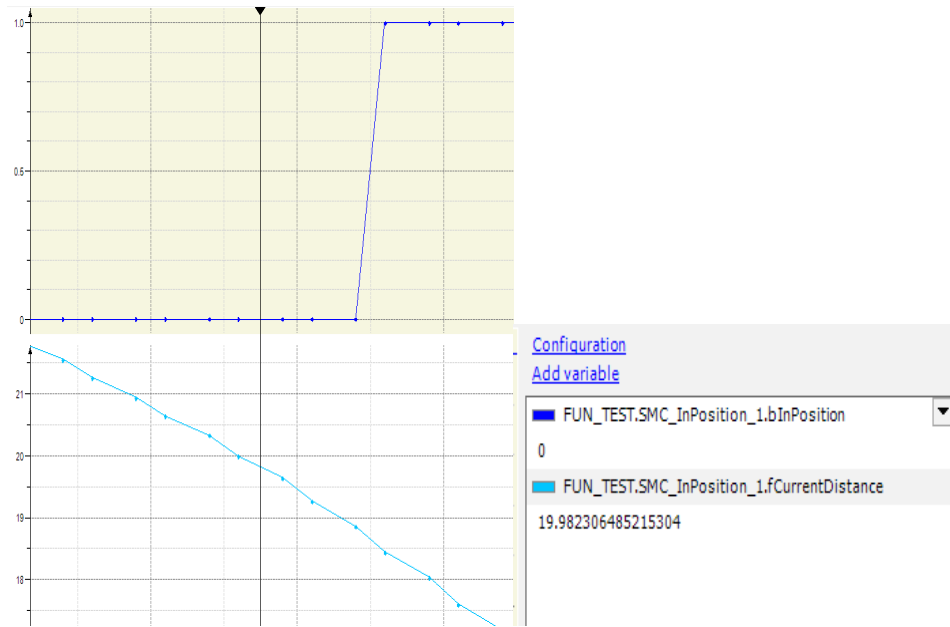
◆ 时序图样列程序



◆ 样例程序

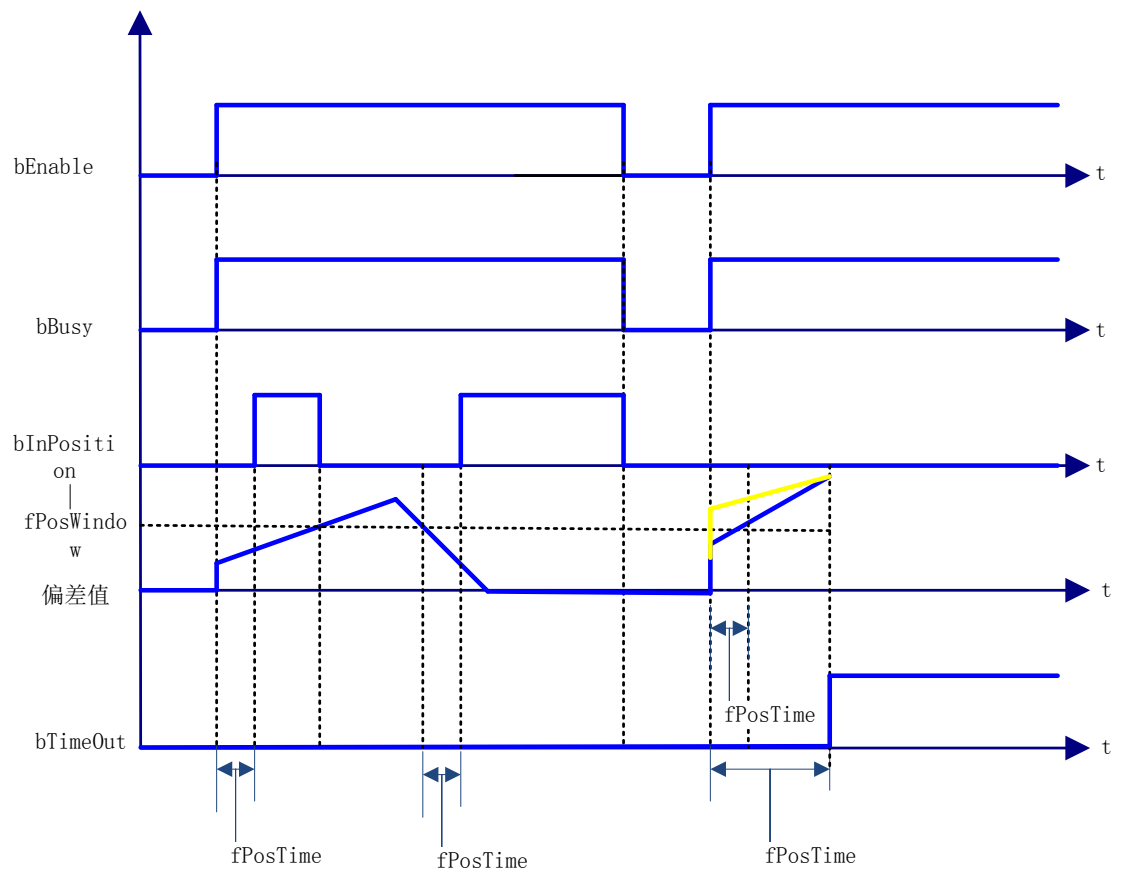


大于窗口设定 bInPosition 立即由 true 变为 FALSE



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

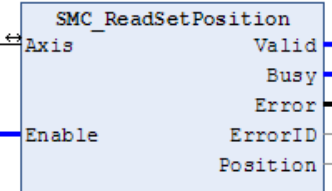
4) 时序图



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		<pre>SMC_ReadSetPosition0(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , Position=>);</pre>

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

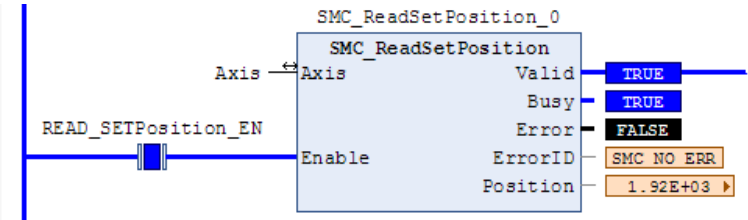
3) 功能说明

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。

◆ 时序图样列程序



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。

【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。

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		<pre>SMC_SetTorque0(Axis:= , bEnable:= , fTorque:= , bBusy=> , bError=> , nErrorID=>);</pre>

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	单位为 0.1%

◆ 输出变量

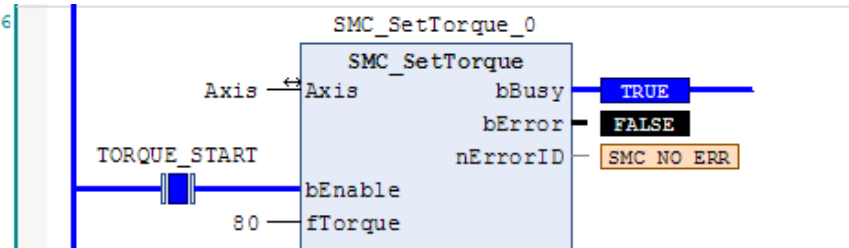
输出变量	名称	数据类型	有效范围	初始值	描述
Busy	执行中	BOOL	TRUE,FALSE	FALSE	True, 执行中
Error	错误	BOOL	TRUE,FALSE	FALSE	True, 异常产生
ErrorID	错误代码	SMC_ERROR			参考 SMC_Error

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。

◆ 时序图样列程序



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

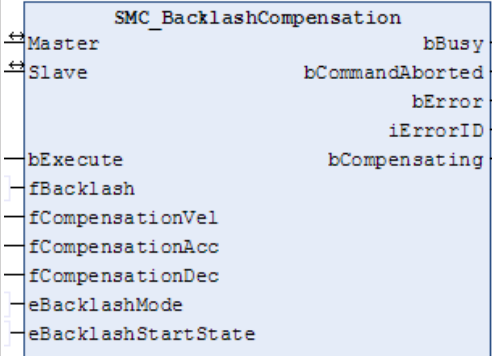
【注意】：请阅读“附录 C 错误代码说明”以了解相关错误代码说明。

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_BacklashCompensation	间隙补偿		<pre>SMC_BacklashCompensation0(Master:= , Slave:= , bExecute:= , fBacklash:= , fCompensationVel:= , fCompensationAcc:= , fCompensationDec:= , eBacklashMode:= , eBacklashStartState:= , bBusy=> , bCommandAborted=> , bError=> , iErrorID=> , bCompensating=>);</pre>

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
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
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

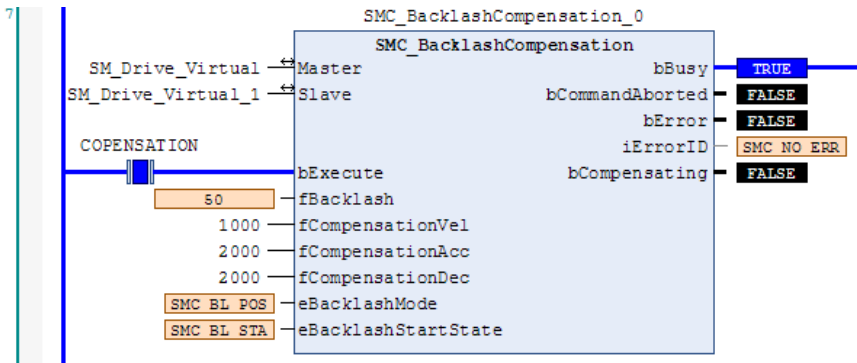
输出变量	名称	数据类型	有效范围	初始值	描述
bBusy	执行中	BOOL	TRUE,FALSE	FALSE	True, 执行中
bCommandAborted	指令被中断	BOOL	TRUE,FALSE	FALSE	True- 被其他控制命令打断
bError	错误	BOOL	TRUE,FALSE	FALSE	True, 异常产生
iErrorID	错误代码	SMC_ERROR			参考 SMC_Error
bCompsating	补偿中	BOOL	TRUE,FALSE	FALSE	

3) 功能说明

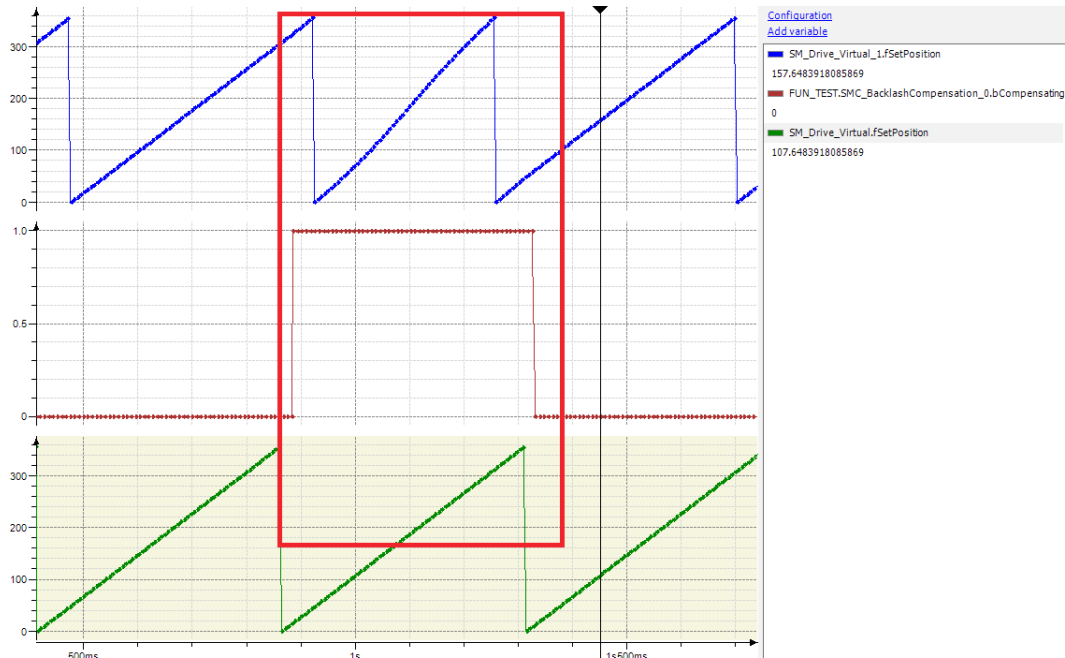
bEecute 上升沿，无错误则，bBusy 输出为 TURE，bCompsating 输出为 true，补偿完成后 bCompsating 输出为 false。

工作方式：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：

◆ 时序图样列程序



◆ 样例程序



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。
【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。

SMC_AxisDiagnosticLog

该指令功能为：周期性的将轴的一个参数写入文件。

1) 指令格式

指令	名称	图形表现	ST 表现
SMC_AxisDiagnosticLog	轴参数 写入文件		<pre> SMC_AxisDiagnosticLog(Axis:= , bExecute:= , bCloseFile:= , sFileName:= , bSetPosition:= , bActPosition:= , bSetVelocity:= , bActVelocity:= , bSetAcceleration:= , bActAcceleration:= , bySeparatorChar:= , sRecordSeparatorString:= , eMode:= , bDone=> , bBusy=> , bError=> , ErrorID=> , bRecording=>); </pre>

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
bClosefile	关闭文件	BOOL	TRUE,FALSE	FALSE	TRUE, 指令立马关闭文件
sFileName	文件名	STRING(80)		“ ”	存储的文件名（路径之前）
bSetPosition	记录设定位置	BOOL	TRUE,FALSE	FALSE	TRUE, 执行指令时记录设定位置
bActPosition	记录实际位置	BOOL	TRUE,FALSE	FALSE	TRUE, 执行指令时记录实际位置
bSetVelocity	记录设定速度	BOOL	TRUE,FALSE	FALSE	TRUE, 执行指令时记录设定速度
bActVelocity	几轮实际速度	BOOL	TRUE,FALSE	FALSE	TRUE, 执行指令时记录实际速度
bSetAcceleration	记录设定加速度	BOOL	TRUE,FALSE	FALSE	TRUE, 执行指令时记录设定加速度
bActAcceleration	记录实际加速度	BOOL	TRUE,FALSE	FALSE	TRUE, 执行指令时记录实际加速度
bySeparatorChar		BYTE		9	ASCII 代码值，写在两个不同值之间

输入变量	名称	数据类型	有效范围	初始值	描述
sRecordSe- paratorString				‘\$R\$N’	日期结束时写的字符串
eMode		SMC_ LOGGERMODE	LOG_ CONTINUOUS		log_continuous: 连续记录到文件 log_at_close: 连续记录到缓冲区 (10kbyte)。当 bclosefile 为 true 将缓冲区的数据写入文件

◆ 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, 保存完成
bBusy	执行中	BOOL	TRUE,FALSE	FALSE	True, 执行中
bError	错误	BOOL	TRUE,FALSE	FALSE	True, 异常产生
ErrorID	错误代码	SMC_ERROR			参考 SMC_Error
bRecording	记录中	BOOL	TRUE,FALSE	FALSE	True, 参数正在保存记录中

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) 错误说明

bExecute 上升沿时：

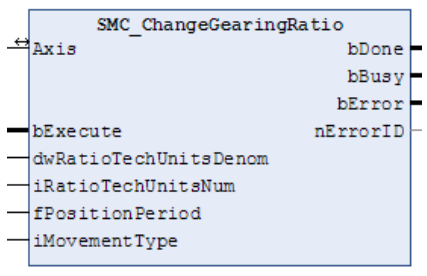
轴报错，Error 输出；无效的轴输入，Error 输出。

【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。

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		<pre>SMC_ChangeGearingRatio0(Axis:= , bExecute:= , dwRatioTechUnitsDenom:= , iRatioTechUnitsNum:= , fPositionPeriod:= , iMovementType:= , bDone=> , bBusy=> , bError=> , nErrorID=>);</pre>

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。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

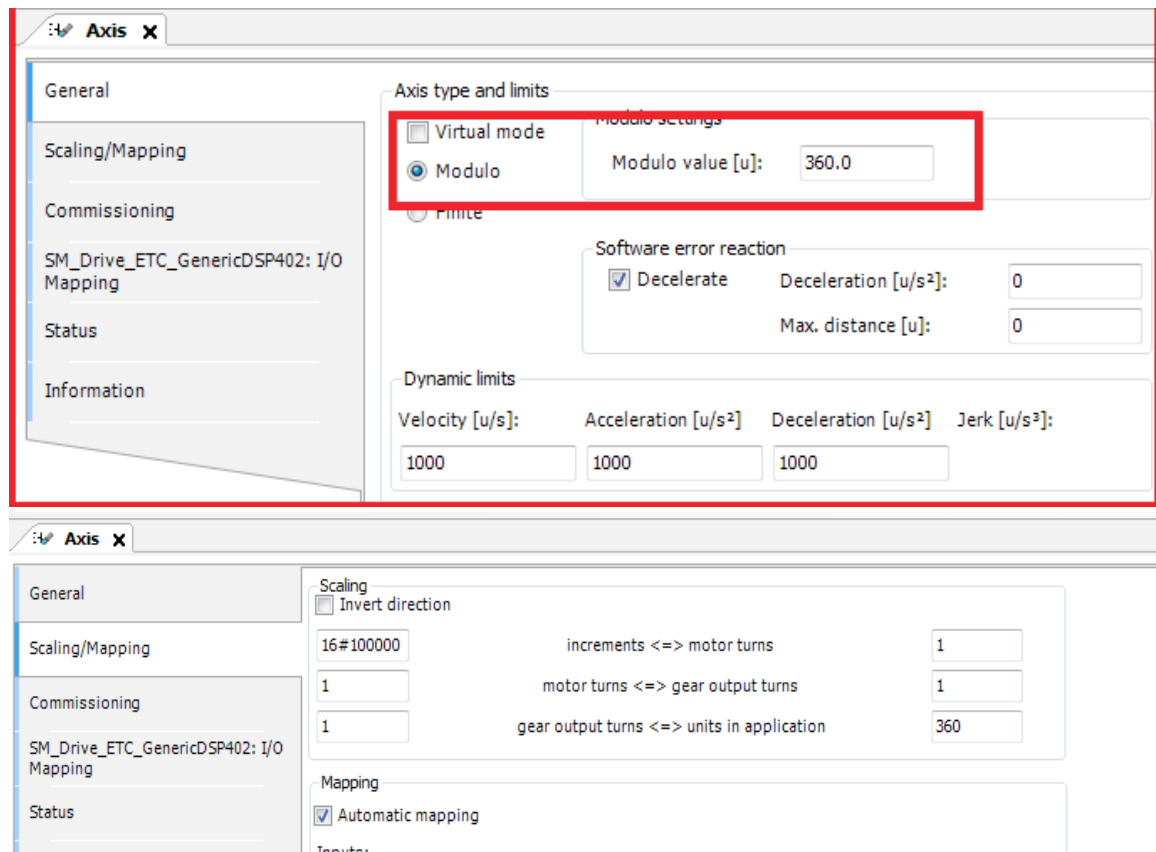
输出变量	名称	数据类型	有效范围	初始值	描述
bDone	完成	BOOL	TRUE,FALSE	FALSE	True, 执行设定完成
bBusy	执行中	BOOL	TRUE,FALSE	FALSE	True, 执行中
bError	错误	BOOL	TRUE,FALSE	FALSE	True, 异常产生
nErrorID	错误代码	SMC_ERROR			参考 SMC_Error

3) 功能说明

bExecute 上升沿，无错误则，bBusy 输出为 TRUE，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，丝杠动距离 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：

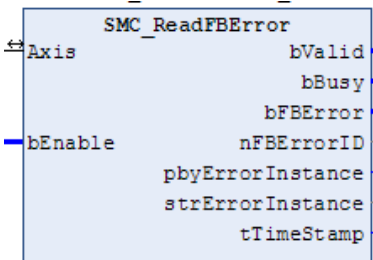
- ◆ 轴报错，Error 输出。
- ◆ 输入值无效，Error 输出，错误代码 SMC_CGR_ZERO_VALUES
- ◆ 轴在指令控制运行中，Error 输出，错误代码 SMC_CGR_DRIVE_POWERED
- ◆ 输入的模数值无效（eg: <0），Error 输出，错误代码 SMC_CGR_INVALID_POSPERIOD

【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。

SMC_ReadFBError

该指令功能为：MC,SMC 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		<pre>SMC_ReadFBError(Axis:= , bEnable:= , bValid=> , bBusy=> , bFBError=> , nFBErrorID=> , pbyErrorInstance=> , strErrorInstance=> , tTimeStamp=>);</pre>

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

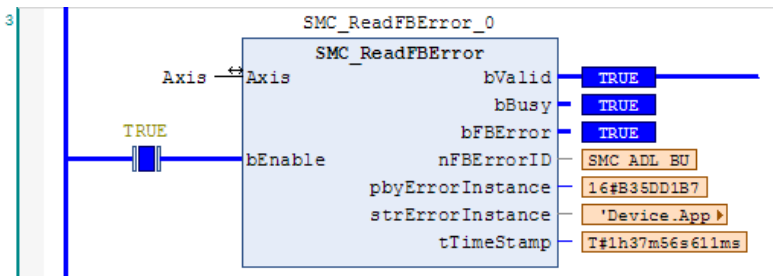
3) 功能说明

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

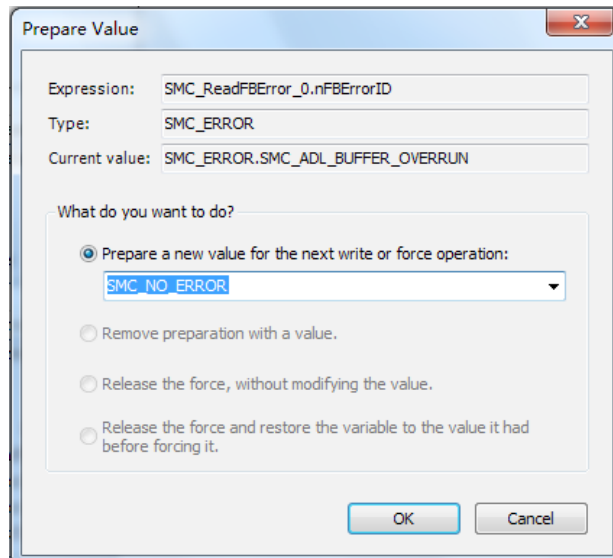
有功能块报警则，bFBError 输出为 true。

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

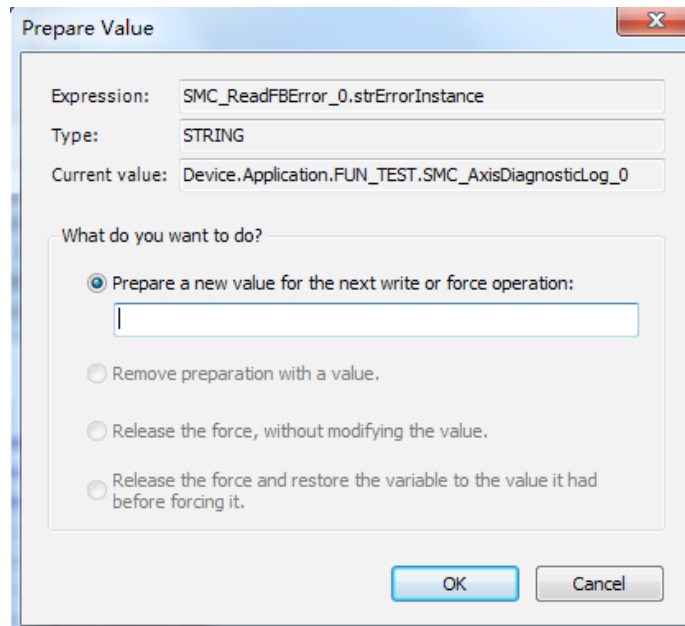
◆ 时序图样列程序

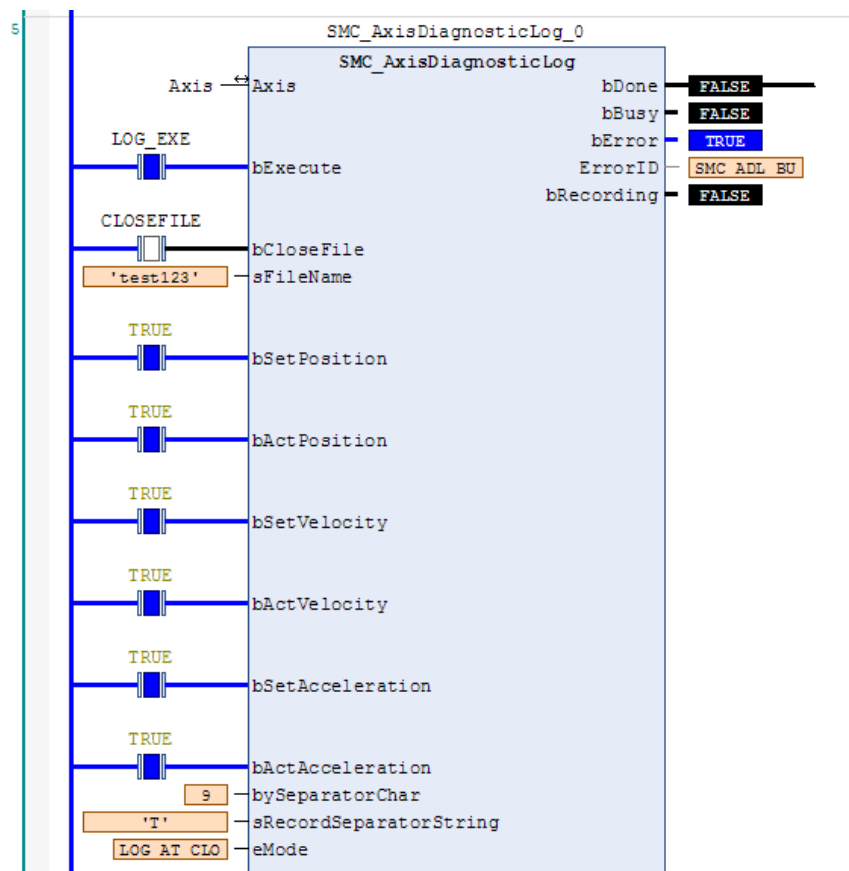


◆ 样例程序



◆ 错误 ID





发生错误的功能块

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。

【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。

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	读功能块错误		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	-	-	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