HCFA
禾川股份

Q 系列可编程逻辑控制器

运动控制指令手册

ATC/MQXM2110

2021 年 11 月 3 日

禾川自动化中心 ATC

目录

轴管理功能块 5

MC_Power（FB） 5

SMC_SetControllerMode(FB) 7

MC_SetPosition(FB) 9

MC_Reset(FB) 10

MC_TouchProbe(FB) 12

MC_AbortTrigger(FB) 15

MC_ReadStatus(FB) 16

MC_ReadActualPosition(FB) 18

MC_ReadActualVelocity(FB) 19

MC_ReadActualTorque(FB) 21

MC_ReadAxisError(FB) 22

MC_ReadParameter（FB） 24

MC_ReadBoolParameter（FB） 25

MC_WriteParameter（FB） 27

MC_WriteBoolParameter（FB） 28

SMC_CheckAxisCommunication（FB） 30

SMC_ReadFBError(FB) 31

SMC_ClearFBError（FUN） 33

SMC3_PersistPosition（FB） 33

SMC3_PersistPositionSingleturn（FB） 35

SMC_ChangeGearingRatio（FB） 37

SMC3_ReInitDrive（FB） 39

SMC3_ETC_WriteParameter_CoE（FB） 40

SMC3_ETC_ReadParameter_CoE（FB） 42

ETC_CO_SdoWrite（FB） 43

ETC_CO_SdoRead（FB） 45

单轴运动控制 47

MC_Home（FB） 47

SMC_Homing（FB） 50

MC_Stop（FB） 52

MC_Halt（FB） 53

MC_Jog（FB） 55

MC_MoveAbsolute（FB） 56

MC_MoveRelative（FB） 58

MC_MoveVelocity（FB） 60

SMC_SetTorque(FB) 61

MC_MoveSuperImposed（FB） 63

MC_MoveAdditive（FB） 65

多轴运动控制 67

MC_GearIn（FB） 67

MC_GearOut（FB） 69

MC_GearInPos（FB） 70

MC_Phasing（FB） 72

MC_CamIn(FB) 74

MC_CamTableSelect（FB） 76

MC_CamOut（FB） 78

SMC_GetCamSlaveSetPosition（FB） 80

SMC_FollowPosition（FB） 82

SMC_FollowPositionVelocity（FB） 83

SMC_FollowVelocity（FB） 85

SMC_FollowSetValues（FB） 87

SMC_GetTappetValue（FB） 89

参考 90

MC_Direction（ENUM） 90

SMC_Controller_Mode（ENUM） 90

SMC_Homing_Mode(ENUM) 91

SMC_CommunicationState(ENUM) 91

MC_StartMode(ENUM) 91

TRIGGER_REF（STRUCT） 92

SMC3_PersistPositionDiag (ENUM) 92

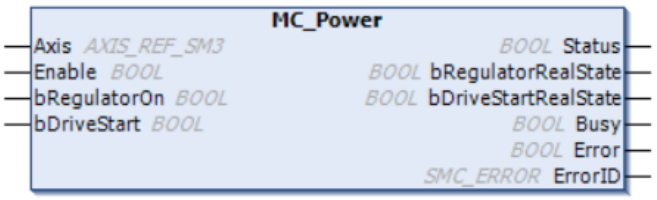
轴状态转移说明（ENUM） 93

SMC_Error(ENUM) 94

轴管理功能块

MC_Power（FB）

将轴进行使能。

名称	MC_Power（轴使能）		
支持的模式	CSP	CSV	CST
图形表现		ST 表现	
		MC_Power(Axis:= , Enable:= , bRegulatorOn:= , bDriveStart:= , Status=> , bRegulatorRealState=> , bDriveStartRealState=> , Busy=> , Error=> , ErrorID=>);	

变量

（1） 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

（2） 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Enable	有效	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块 FALSE：关闭功能块
bRegulatorOn	使能	BOOL	TRUE、FALSE	FALSE	TRUE:打开轴使能 FALSE：关闭轴使能
bDriveStart	驱动开启	BOOL	TRUE、FALSE	FALSE	TRUE：允许轴运动 FALSE：禁止轴运动

（3） 输出变量

输出变量	名称	数据类型	有效范围	内容
Status	状态	BOOL	TRUE、FALSE	TRUE:轴可以执行运动控制功能块
bRegulatorRealState	轴使能信号状态	BOOL	TRUE、FALSE	TRUE:轴使能有效
bDriveStartRealState	驱动开启状态	BOOL	TRUE、FALSE	TRUE:允许轴运动
Busy	执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR	0	功能块产生异常时输出对应的故障码

（4） 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
Status	✧ 轴可以执行运动控制功能块	✧ 轴不可运动 ✧ Error 变为 TRUE
bRegulatorRealState	✧ 轴使能有效	✧ 轴使能失效
bDriveStartRealState	✧ 允许轴运动	✧ 轴不允许运行
Busy	✧ Enable 的上升沿	✧ Enable 变为 FALSE 时 ✧ Error 变为 TRUE 时
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时

要点说明

- 将 Enable 设为 TRUE 后，MC_Power 进入执行状态，此时其他的输入引脚被 MC_Power 处理。
- 将 Enable 设为 FALSE 后，MC_Power 将不会再执行任何功能块，此时修改 MC_Power 其他的输入引脚将不会产生任何效果。
- bRegulatorOn（使能）引脚控制轴使能，bRegulatorOn 为 TRUE 时，MC_Power 会尝试使能轴，轴使能成功之后 MC_Power 的 bRegulatorRealState 引脚输出为 TRUE。bRegulatorOn 为 FALSE 时，MC_Power 会断开轴使能，bRegulatorRealState 引脚输出为 FALSE。
- bDriveStart（驱动开启）引脚允许轴运动，bDriveStart 为 TRUE 时，MC_Power 允许轴运动，bDriveStartRealState 引脚输出为 TRUE；bDriveStart 为 FALSE 时，MC_Power 禁止轴运动，bDriveStartRealState 引脚输出为 FALSE。
- 在轴运动时将 MC_Power 的 bRegulatorOn 或者 bDriveStart 关闭，此时正在执行的功能块 CommandAborted 变为 TRUE，轴切换为不可运动状态，所有运动控制都停止。
- 当 bRegulatorRealState 为 TRUE、bDriveStartRealState 为 TRUE， 且轴没有故障产生，Status 引脚将输出为 TRUE，轴状态将转换为 StandStill，可以执行其他运动控制功能块。

➢ ※注：断开轴使能时，请勿同时关闭 MC_Power 的 bEnable 和 bRegulatorOn，此操作会出现轴无法断使能的情况，建议不要关闭 MC_Power 的 Enable 引脚。

SMC_SetControllerMode(FB)

设置驱动器的控制模式。

名称	SMC_SetControllerMode(更改控制模式)		
支持的模式	CSP	CSV	CST
图形表现		ST 表现	
		SMC_SetControllerMode(Axis:= , bExecute:= , nControllerMode:= , bDone=> , bBusy=> , bError=> , nErrorID=>);	

变量

(1) 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

(2) 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
bExecute	启动	BOOL	TRUE、FALSE	FALSE	输入一个上升时开始执行功能块
nControllerMode	控制模式	SMC_CONTROLLER_MODE		SMC_Position	选择控制模式，参考 SMC_Controller_Mode

(3) 输出变量

输出变量	名称	数据类型	有效范围	内容
bDone	模式设置完成	BOOL	TRUE、FALSE	TRUE：功能块执行完成
bBusy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
bError	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
nErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码

（4） 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
bDone	✧ 完成时	✧ bExecute 为 TRUE 时，与 bExecute 的 FALSE 同时 ✧ bExecute 为 FALSE 时，1 个周期后
bBusy	✧ bExecute 的上升沿	✧ bDone 为 TRUE 时 ✧ bError 为 TRUE 时 ✧ bCommandAborted 变为 TRUE 时
bError	✧ 功能块执行中产生故障时	✧ bExecute 为 TRUE 时，与 bExecute 的 FALSE 同时 ✧ bExecute 为 FALSE 时，1 个周期后

要点说明

- bExecute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数，重新执行功能块。
- 通过 nControllerMode 可以设置驱动器的控制模式，具体的模式转换参考 SMC_Controller_Mode。
- 轴必须为实体轴，该功能块修改的是轴 byControllerMode（控制模式的命令值）参数，并且会等待 byRealControllerMode（控制模式的实际命令）参数返回。
- 必须支持指定的控制模式，具体可查看轴对应驱动器的说明手册。
- 对应模式的通讯数据需要先配置好，如转矩模式需要配置目标转矩等通讯参数。
- 功能块被调用时，轴不能处于 ErrorStop、Homing、Stopping 状态，否则将会报错（SMC_SCM_AXIS_IN_WRONG_STATE）。
- 在功能块执行的 1000 个周期后，如果依旧没有变成设定的控制模式，则这个功能块会中断并且返回一个错误。
- 当功能块输出为 bDone 之后，需要立即使用对应模式支持的功能块接管轴的运动控制，否则轴的运动会处于不受监管状态，产生故障。

MC_SetPosition(FB)

将轴的功能块位置和反馈位置修改为功能块输入的位置数据。

名称	MC_SetPosition（设置位置）		
支持的模式	CSP	CSV	CST
图形表现		ST 表现	
		MC_SetPosition(Axis:=, Execute:= , Position:= , Mode:= , Done=> , Busy=> , Error=> , ErrorID=>);	

变量

(1) 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

(2) 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块
Position	目标位置	LREAL	负数、正数、“0”	0	指定设定的目标位置，单位为【指令单位】
Mode	模式选择	BOOL	TRUE、FALSE	FALSE	TRUE：相对 FALSE：绝对

(3) 输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	TRUE：功能块执行完成
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码

(4) 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
Done	✧ 位置偏移完成时	✧ Execute 为 TRUE 时，与 Execute 的 FALSE 同时 ✧ Execute 为 FALSE 时，1 个周期后
Busy	✧ Execute 的上升沿	✧ Done 为 TRUE 时 ✧ Error 为 TRUE 时
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时

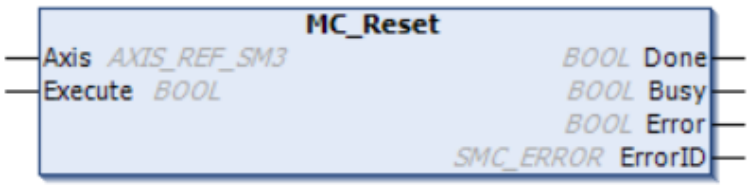
要点说明

- Execute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数，重新执行功能块。
- 本功能块不修改实际编码器位置，只是对当前坐标系产生坐标偏移。
- Mode 引脚为 TRUE 时，为相对模式，MC_SetPosition 会将 Position 引脚的值与轴当前相加作为坐标系偏移之后的位置。
- Mode 引脚为 FALSE 时，为绝对模式，MC_SetPosition 会将 Position 引脚的值直接作为偏移之后的位置。

➢ ※注：此功能块产生的偏移会在控制器断电上电后丢失，所以不可替代 MC_Home 功能块。

MC_Reset(FB)

复位轴状态，用于将轴从 ErrorStop 变为 StandStill 状态，或者复位驱动器故障。

名称	MC_Reset（轴复位）		
支持的模式	CSP	CSV	CST
图形表现		ST 表现	
		MC_Reset(Axis:=, Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);	

变量

(1) 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

(2) 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块

(3) 输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	TRUE：功能块执行完成
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码

(4) 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
Done	✧ 完成时	✧ Execute 为 TRUE 时，与 Execute 的 FALSE 同时 ✧ Execute 为 FALSE 时，1 个周期后
Busy	✧ Execute 的上升沿	✧ Done 为 TRUE 时 ✧ Error 为 TRUE 时 ✧ CommandAborted 变为 TRUE 时
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时

要点说明

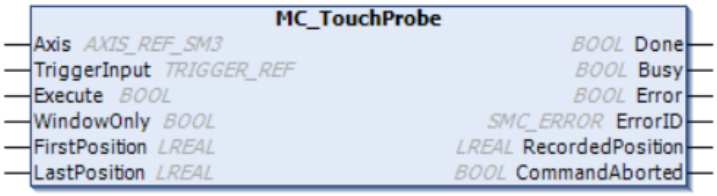
- Execute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数，重新执行功能块。
- 本功能块在轴通讯正常的情况下，把轴处于 Errorstop 变成 Standstill，把轴的异常状态变为正常可运行的状态，同时也会复位驱动器故障状态。
- 如果轴在其他状态下使用 MC_Reset 功能块会报错，轴不会报错，并且不会打断当前轴状态。

无法复位通信相关的故障。

➢ ※注：通信相关的故障需要再主站与从站重新建立连接，通信正常后，调用 SMC3_ReInitDrive 功能块复位。本功能块对于所有种类的轴都可以使用。

MC_TouchProbe(FB)

通过触发探针信号记录信号触发时轴的位置。

名称	MC_TouchProbe（启用外部锁定）		
支持的模式	CSP	CSV	CST
图形表现		ST 表现	
		MC_TouchProbe(Axis:= , TriggerInput:= , Execute:= , WindowOnly:= , FirstPosition:= , LastPosition:= , Done=> , Busy=> , Error=> , ErrorID=> , RecordedPosition=> , CommandAborted=>);	

变量

(1) 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例
TriggerInput	触发脉冲输入	TRIGGER_REF	关联到上升沿信号，参考 TRIGGER_REF

(2) 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块
WindowOnly	窗口有效	BOOL	TRUE、FALSE	FALSE	如果为 TRUE，触发信号只在规定范围内有效
FirstPosition	开始位置	LREAL	-	0	指定接收触发信号的开始位置
LastPosition	结束位置	LREAL	-	0	指定接收触发信号的结束位置

(3) 输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	TRUE：功能块执行完成
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码
RecordedPosition	记录位置	LREAL	0	触发信号发生时记录的位置
CommandAborted	功能块执行中断	BOOL	TRUE、FALSE	TRUE:功能块被中止

(4) 输出变量的转化时序

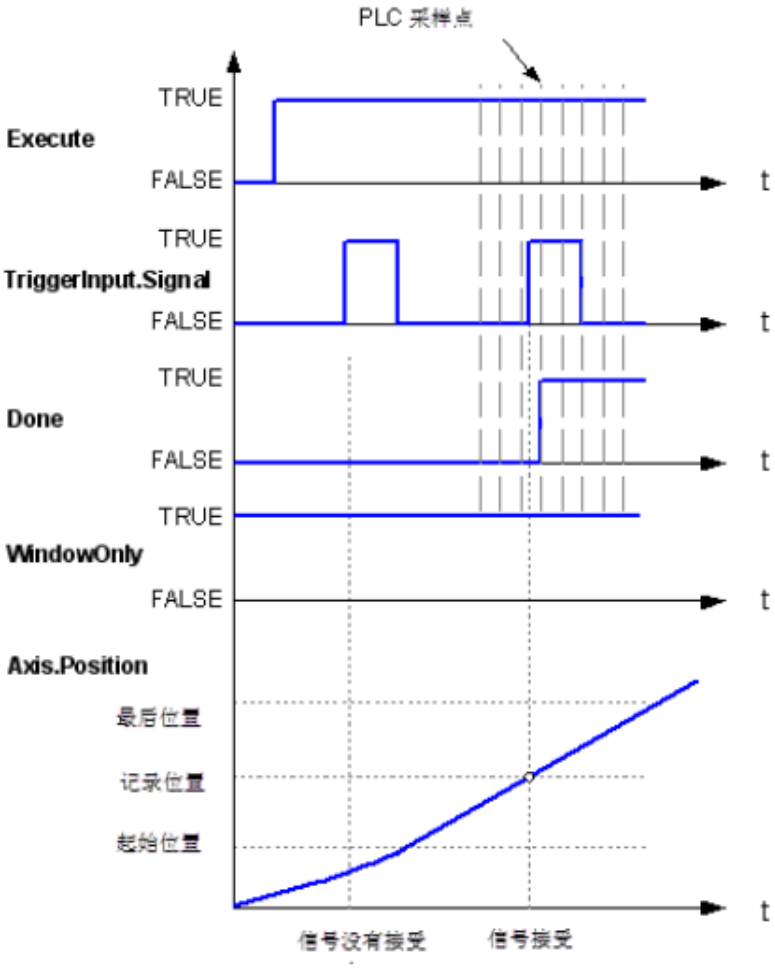
变量	变为 TRUE 时	变为 FALSE 时
Done	✧ 完成时	✧ Execute 为 TRUE 时，与 Execute 的 FALSE 同时 ✧ Execute 为 FALSE 时，1 个周期后
Busy	✧ Execute 的上升沿	✧ Done 为 TRUE 时 ✧ Error 为 TRUE 时 ✧ CommandAborted 变为 TRUE 时
CommandAborted	✧ 由其他功能块进行运动功能块多重启动（中断），本功能块中止时 ✧ 因发生异常，本功能块中止时 ✧ 正在发生异常时，启动了本功能块 ✧ 正在执行 MC_Stop 功能块时，启动了本功能块	✧ Execute 为 TRUE 时，与 Execute 的 FALSE 同时 ✧ Execute 为 FALSE 时，1 个周期后
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时

要点说明

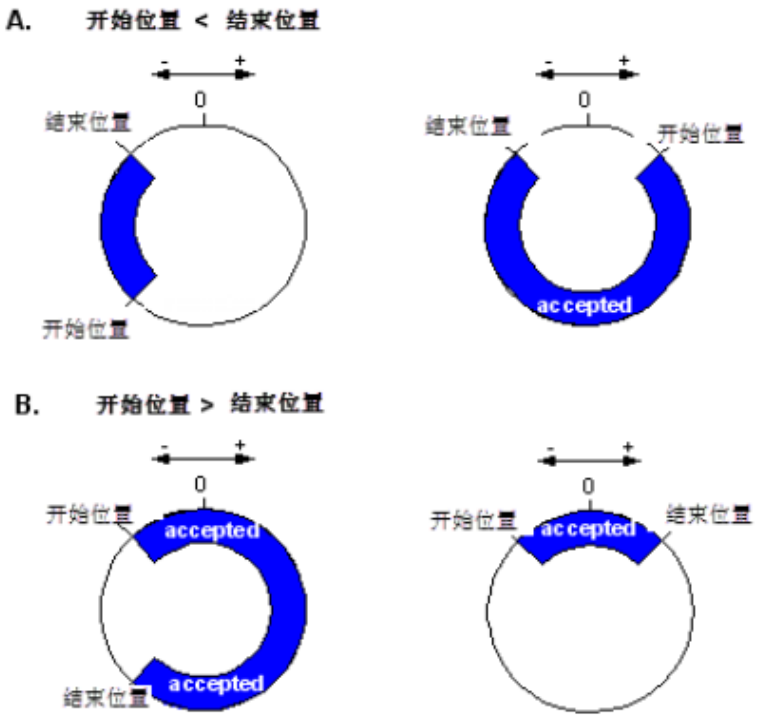
- Execute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数，重新执行功能块。
- 本功能块通过 TriggerInput（触发脉冲输入）、WindowOnly（窗口有效）、FirstPosition（开始位置）、LastPosition（结束位置）作为触发设定，当信号触发时，触发时轴的位置将输出到 RecordedPosition（记录位置）上。

➢ ※注：获取轴当前位置的方法有两种，建议使用 16#60B8 定义的探针功能，具体使用方法请参考伺服手册。

时序图



模态轴的窗口示例



MC_AbortTrigger(FB)

与 MC_AbortTrigger 配合使用，断开与 TriggerInput 变量的关联。

名称	MC_AbortTrigger（禁用外部锁定）		
支持的模式	CSP	CSV	CST
图形表现		ST 表现	
		MC_AbortTrigger(Axis:=, TriggerInput:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);	

变量

(1) 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例
TriggerInput	触发脉冲输入	TRIGGER_REF	关联到上升沿信号，参考 TRIGGER_REF

(2) 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	输入一个上升时开始指

(3) 输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	TRUE：功能块执行完成
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码

(4) 输出变量的转化时序

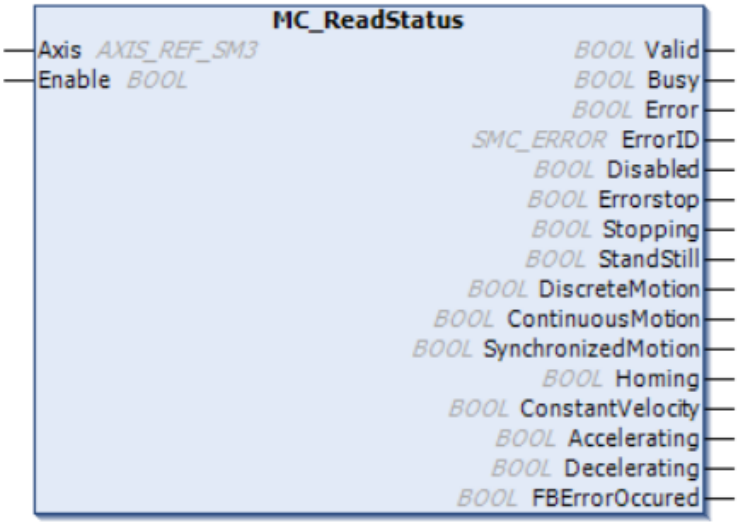
变量	变为 TRUE 时	变为 FALSE 时
Done	✧ 完成时	✧ Execute 为 TRUE 时，与 Execute 的 FALSE 同时 ✧ Execute 为 FALSE 时，1 个周期后
Busy	✧ Execute 的上升沿	✧ Done 为 TRUE 时 ✧ Error 为 TRUE 时
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时

要点说明

- Execute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数，重新执行功能块。
- 本功能块通过可终止轴与 TriggerInput（触发脉冲输入）变量的关联，与 MC_AbortTrigger 配合使用。

MC_ReadStatus(FB)

读取轴当前的状态信息。

名称	MC_ReadStatus（读取轴状态）		
支持的模式	CSP	CSV	CST
图形表现		ST 表现	
		MC_ReadStatus(Axis:=, Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , Disabled=> , Errorstop=> , Stopping=> , StandStill=> , DiscreteMotion=> , ContinuousMotion=> , SynchronizedMotion=> , Homing=> , ConstantVelocity=> , Accelerating=> , Decelerating=> , FBErrorOccured=>);	

变量

（1） 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

(2) 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Enable	有效	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块 FALSE：关闭功能块

(3) 输出变量

输出变量	名称	数据类型	有效范围	内容
Valid	轴状态数据可获取标志	BOOL	TRUE、FALSE	TRUE:输出有效
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
CommandAborted	功能块执行中断	BOOL	TRUE、FALSE	TRUE:功能块被中止
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码
Disabled	轴未使能标志	BOOL	TRUE、FALSE	TRUE:轴没使能
Errorstop	轴错误标志	BOOL	TRUE、FALSE	TRUE:轴在错误运行状态
Stoping	轴停止标志	BOOL	TRUE、FALSE	TRUE:轴状态为 Stopping
StandStill	轴等待标志	BOOL	TRUE、FALSE	TRUE:轴状态为 StandStill
DiscreteMotion	轴离散状态标志	BOOL	TRUE、FALSE	TRUE:轴状态为 DiscreteMotion
ContinuousMotion	轴连续状态标志	BOOL	TRUE、FALSE	TRUE:轴状态为 ContinuousMotion
SynchronizedMotion	轴同步状态标志	BOOL	TRUE、FALSE	TRUE:轴状态为 SynchronizedMotion
Homing	轴回原状态标志	BOOL	TRUE、FALSE	TRUE:轴状态为 Homing
ConstantVelocity	轴匀速状态标志	BOOL	TRUE、FALSE	TRUE:轴运行速度到达
Accelerating	轴正在加速标志	BOOL	TRUE、FALSE	TRUE:轴在加速过程中
Decelerating	轴正在减速标志	BOOL	TRUE、FALSE	TRUE:轴在减速过程中
FBErrorOccured	功能块错误标志	BOOL	TRUE、FALSE	TRUE:发生函数块错误。它还没有被 SMC_ClearFBError 清除

(4) 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
Valid	✧ 输出有效	✧ Enable 为 TRUE 时，与 Enable 的 FALSE 同时 ✧ Enable 为 FALSE 时，1 个周期后
Busy	✧ Enable 的上升沿	✧ Done 为 TRUE 时 ✧ Error 为 TRUE 时 ✧ CommandAborted 变为 TRUE 时
CommandAborted	✧ 由其他功能块进行运动功能块多重启动（中断），本功能块中止时 ✧ 因发生异常，本功能块中止时 ✧ 正在发生异常时，启动了本功能块 ✧ 正在执行 MC_Stop 功能块时，启动了本功能块	✧ Enable 为 TRUE 时，与 Enable 的 FALSE 同时 ✧ Enable 为 FALSE 时，1 个周期后
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时
Disabled	✧ 轴没使能时	✧ 轴使能时
Errorstop	✧ 轴在错误运行状态	✧ 轴不在错误运行状态
Stoping	✧ 轴状态为 Stoping 时	✧ 轴状态不为 Stoping 时
StandStill	✧ 轴状态为 StandStill 时	✧ 轴状态不为 StandStill 时
DiscreteMotion	✧ 轴状态为 DiscreteMotion 时	✧ 轴状态不为 DiscreteMotion 时
ContinuousMotion	✧ 轴状态为 ContinuousMotion 时	✧ 轴状态不为 ContinuousMotion 时

SynchronizedMotion	✧ 轴状态为 SynchronizedMotion 时	✧ 轴状态不为 SynchronizedMotion 时
Homing	✧ 轴状态为 Homing 时	✧ 轴状态不为 Homing 时
ConstantVelocity	✧ 轴匀速运动时	✧ 轴没有匀速运动时
Accelerating	✧ 轴在加速时状态	✧ 轴不在加速时状态
Decelerating	✧ 轴在减速时状态	✧ 轴不在减速时状态
FBErrorOccured	✧ 功能块错误时	✧ 功能块没有错误时

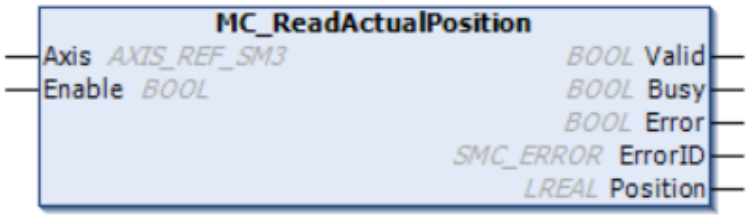
要点说明

- 功能块的 Enable 为 TRUE 时，可以实时更新轴状态信息；

  功能块的 Enable 为 FALSE 时，会保留之前更新的轴状态信息。

MC_ReadActualPosition(FB)

读取轴当前实际位置。

名称	MC_ReadActualPosition（读取轴实际位置）		
支持的模式	CSP	CSV	CST
图形表现		ST 表现	
		MC_ReadActualPosition(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , Position=>);	

变量

（1） 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

（2） 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Enable	有效	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块 FALSE：关闭功能块

（3） 输出变量

输出变量	名称	数据类型	有效范围	内容
Valid	轴状态数据可获取标志	BOOL	TRUE、FALSE	TRUE:输出有效
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码
Position	读取到的轴实际位置	LREAL		功能块读取出的轴实际位置数据

（4） 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
Valid	✧ 输出有效	✧ Enable 为 TRUE 时，与 Enable 的 FALSE 同时 ✧ Enable 为 FALSE 时，1 个周期后
Busy	✧ Enable 的上升沿	✧ Done 为 TRUE 时 ✧ Error 为 TRUE 时
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时

要点说明

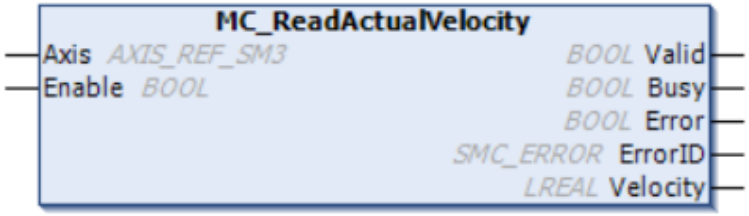
- 本功能块可以读取轴的实际位置：

  当 Enable 为 TRUE 时，功能块实时更新 Position 数据；

  当 Enable 为 FALSE 时，保留上一次更新的数据。

MC_ReadActualVelocity(FB)

读取轴当前实际速度。

名称	MC_ReadActualVelocity（读取轴实际速度）		
支持的模式	CSP	CSV	CST
图形表现		ST 表现	
		MC_ReadActualVelocity(Axis:=, Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , Velocity=>);	

变量

（1） 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，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	内容
Enable	有效	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块 FALSE：关闭功能块

（2） 输出变量

输出变量	名称	数据类型	有效范围	内容
Valid	轴状态数据可获取标志	BOOL	TRUE、FALSE	TRUE:输出有效
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码
Velocity	读取到的轴实际速度	LREAL		功能块读取出的轴实际速度数据

（3） 输变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
Valid	✧ 输出有效	✧ Enable 为 TRUE 时，与 Enable 的 FALSE 同时 ✧ Enable 为 FALSE 时，1 个周期后
Busy	✧ Enable 的上升沿	✧ Done 为 TRUE 时 ✧ Error 为 TRUE 时
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时

要点说明

- 本功能块可以读取驱动器中的实际速度：

  当 Enable 为 TRUE 时，功能块实时更新 Velocity 数据；

  当 Enable 为 FALSE 时，保留上一次更新的数据。

MC_ReadActualTorque(FB)

读取轴当前实际转矩。

名称	MC_ReadActualTorque（读取轴实际转矩）		
支持的模式	CSP	CSV	CST
图形表现		ST 表现	
		MC_ReadActualTorque(Axis:=, Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , Torque=>);	

变量

（1） 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

（2） 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Enable	有效	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块 FALSE：关闭功能块

（3） 输出变量

输出变量	名称	数据类型	有效范围	内容
Valid	轴状态数据可获取标志	BOOL	TRUE、FALSE	TRUE:输出有效
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码
Torque	读取到的轴实际转矩	LREAL		功能块读取出的轴实际转矩数据（单位 Nm）

（4） 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
Valid	✧ 输出有效	✧ Enable 为 TRUE 时，与 Enable 的 FALSE 同时 ✧ Enable 为 FALSE 时，1 个周期后
Busy	✧ Enable 的上升沿	✧ Done 为 TRUE 时 ✧ Error 为 TRUE 时
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时

要点说明

- 本功能块可以读取驱动器中的实际转矩：

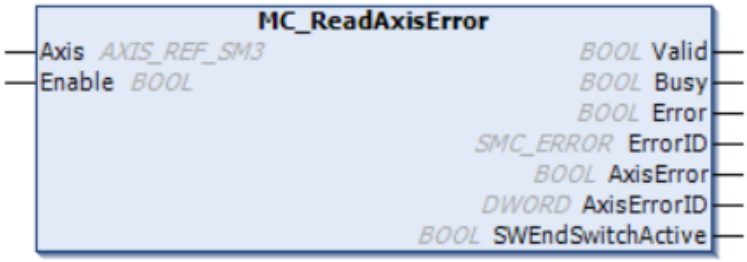
  当 Enable 为 TRUE 时，功能块实时更新 Torque 数据；

  当 Enable 为 FALSE 时，保留上一次更新的数据。
- 需要配置实际转矩相关的通讯数据，才可读取到正确的转矩值。

➢ ※注：这个功能块读取到的是轴的实际输出转矩值，并非额定转矩的百分比。

MC_ReadAxisError(FB)

读取轴错误状态。

名称	MC_ReadAxisError（读取轴错误状态）		
支持的模式	CSP	CSV	CST
图形表现		ST 表现	
		MC_ReadAxisError(Axis:=, Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , AxisError=> , AxisErrorID=> , SWEndSwitchActive=>);	

变量

（1） 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

（2） 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Enable	有效	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块 FALSE：关闭功能块

（3） 输出变量

输出变量	名称	数据类型	有效范围	内容
Valid	轴状态数据可获取标志	BOOL	TRUE、FALSE	TRUE:输出有效
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码
AxisError	轴错误标志	BOOL	TRUE、FALSE	当轴发生错误时为 TRUE
AxisErrorID	轴错误代码	DWORD		读取轴的错误代码
SWEndSwitchActive	软限位开关有效	BOOL	TRUE、FALSE	TRUE:软限位触发

（4） 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
Valid	✧ 输出有效	✧ Enable 为 TRUE 时，与 Enable 的 FALSE 同时 ✧ Enable 为 FALSE 时，1 个周期后
Busy	✧ Enable 的上升沿	✧ Done 为 TRUE 时 ✧ Error 为 TRUE 时
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时

要点说明

- 本功能块可以读取驱动器中的错误代码：

  当 Enable 为 TRUE 时，功能块实时更新轴错误相关的数据；

  当 Enable 为 FALSE 时，保留上一次更新的数据。

MC_ReadParameter（FB）

读取轴的参数。

名称	MC_ReadParameter （读取轴的参数）		
支持的模式	CSP	CSV	CST
图形表现		ST 表现	
		MC_ReadParameter(Axis:=, Enable:= , ParameterNumber:= , Valid=> , Busy=> , Error=> , ErrorID=> , Value=>);	

变量

（1） 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

（2） 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Enable	有效	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块 FALSE：关闭功能块
ParameterNumber	轴参数序号	DINT	-	0	轴参数对应的序号，ParameterNumber（轴参数序号）可查看 AXIS_REF_SM3，也可读取对应总线中定义的参数，如 COE 中的状态字

（3） 输出变量

输出变量	名称	数据类型	有效范围	内容
Valid	轴参数数据可获取标志	BOOL	TRUE、FALSE	TRUE：可获取轴参数数据
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码
Value	读取到的轴参数	LREAL	-	功能块读取出的轴参数

（4） 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
Valid	✧ 输出有效	✧ Enable 为 TRUE 时，与 Enable 的 FALSE 同时 ✧ Enable 为 FALSE 时，1 个周期后
Busy	✧ Enable 的上升沿	✧ Done 为 TRUE 时 ✧ Error 为 TRUE 时
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时

要点说明

本功能块在 Enable 为 TRUE 时可以读取对应 ParameterNumber（轴参数序号）的参数，也可读取现场总线协议中定义的参数，如禾川 X3EB 驱动中的 16#6041 状态字。

➢ ※注：读取总线协议参数时，需要在程序里做如下操作

diParameterNumeber 声明为 DINT 类型变量，添加至功能块的输入引脚 ParameterNumeber。

usiDataLength（写想读取的对象字典中的数据长度）声明为 USINT 类型变量。

uiIndex（写想读取的对象字典中的索引）声明为 UINT 类型变量。

usisubIndex（写想读取的对象字典中的子索引）声明为 DINT 类型变量。

举例读取禾川驱动器对象字典中的回原模式：

```
diParameterNumeber :=-DWORD_TO_DINT(SHL(USINT_TO_DWORD(1),24)+SHL(UINT_TO_DWORD(16#6098),8)+16#00);
```

MC_ReadBoolParameter（FB）

读取轴的 BOOL 型参数。

名称	MC_ReadBoolParameter（读取轴的 BOOL 参数）		
支持的模式	CSP	CSV	CST
图形表现		ST 表现	
		MC_ReadBoolParameter(Axis:=, Enable:= , ParameterNumber:= , Valid=> , Busy=> , Error=> , ErrorID=> , Value=>);	

变量

（1） 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

（2） 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Enable	有效	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块 FALSE：关闭功能块
ParameterNumber	轴参数序号	DINT		0	轴参数对应的序号，ParameterNumber（轴参数序号）可查看 AXIS_REF_SM3，也可读取对应总线中定义的参数，如 COE 中的状态字

（3） 输出变量

输出变量	名称	数据类型	有效范围	内容
Valid	轴参数数据可获取标志	BOOL	TRUE、FALSE	TRUE：可获取轴参数数据
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码
Value	读取到的轴 BOOL 参数	BOOL	TRUE、FALSE	功能块读取出的轴参数

（4） 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
Valid	✧ 输出有效	✧ Enable 为 TRUE 时，与 Enable 的 FALSE 同时 ✧ Enable 为 FALSE 时，1 个周期后
Busy	✧ Enable 的上升沿	✧ Done 为 TRUE 时 ✧ Error 为 TRUE 时
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时

要点说明

本功能块在 Enable 为 TRUE 时可以读取对应 ParameterNumber（轴参数序号）的参数，也可读取现场总线协议中定义的参数，如禾川 X3EB 驱动中的 16#2104 组中 DI、DO 的 BOOL 值。

➢ ※注：读取总线协议参数时，需要在程序里做如下操作

diParameterNumeber 声明为 DINT 类型变量，添加至功能块的输入引脚 ParameterNumeber。

usiDataLength（写想读取的对象字典中的数据长度）声明为 USINT 类型变量。

uiIndex（写想读取的对象字典中的索引）声明为 UINT 类型变量。

usisubIndex（写想读取的对象字典中的子索引）声明为 DINT 类型变量。

diParameterNumeber :=-DWORD_TO_DINT(SHL(USINT_TO_DWORD(usidatalength),24)+SHL(UINT_TO_DWORD(uiindex),8)+usisubindex);

具体操作可参考 MC_ReadParameter 读取总线协议参数。

MC_WriteParameter（FB）

修改轴的参数。

名称	MC_WriteParameter（修改轴的参数）		
支持的模式	CSP	CSV	CST
图形表现		ST 表现	
		MC_WriteParameter(Axis:=, Execute:= , ParameterNumber:= , Value:= , Done=> , Busy=> , Error=> , ErrorID=>);	

变量

（1） 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

（2） 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	有效	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块 FALSE：关闭功能块
ParameterNumber	轴参数序号	DINT	-	0	输入想要修改的轴参数的索引和子索引
Value	参数设定	LREAL	-	-	设定参数值

（3） 输出变量

输出变量	名称	数据类型	有效范围	内容
Done	参数设置成功标志	BOOL	TRUE、FALSE	TRUE:参数设置成功
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码

（4） 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
Done	✧ 设置成功	✧ Execute 为 TRUE 时，与 Execute 的 FALSE 同时 ✧ Execute 为 FALSE 时，1 个周期后
Busy	✧ Execute 的上升沿	✧ Done 为 TRUE 时 ✧ Error 为 TRUE 时
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时

要点说明

- Execute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数，重新执行功能块。
- 本功能块可以修改对应 ParameterNumber（轴参数序号）的参数，也可修改现场总线协议中定义的参数，如禾川 X3EB 驱动中的 16#6041 状态字。

➢ ※注：修改总线协议参数时，需要在程序里做如下操作

diParameterNumeber 声明为 DINT 类型变量，添加至功能块的输入引脚 ParameterNumeber。

usiDataLength（写想读取的对象字典中的数据长度）声明为 USINT 类型变量。

uiIndex（写想读取的对象字典中的索引）声明为 UINT 类型变量。

usisubIndex（写想读取的对象字典中的子索引）声明为 DINT 类型变量。diParameterNumeber :=-DWORD_TO_DINT(SHL(USINT_TO_DWORD(usidatalength),24)+SHL(UINT_TO_DWORD(uiindex),8)+usisubindex);

举例修改禾川驱动器对象字典中的回原模式：

```
diParameterNumeber :=-DWORD_TO_DINT(SHL(USINT_TO_DWORD(1),24)+SHL(UINT_TO_DWORD(16#6098),8)+16#00);
```

MC_WriteBoolParameter（FB）

修改轴的 BOOL 型参数。

名称	MC_WriteBoolParameter（修改轴的 BOOL 参数）		
支持的模式	CSP	CSV	CST
图形表现		ST 表现	
		MC_WriteBoolParameter(Axis:=, Execute:= , ParameterNumber:= , Value:= , Done=> , Busy=> , Error=> , ErrorID=>);	

变量

（1） 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

（2） 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	有效	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块 FALSE：关闭功能块
ParameterNumber	轴参数序号	DINT		0	输入想要修改的轴参数的索引和子索引
Value	参数设定	BOOL	TRUE、FALSE		设定参数值

（3） 输出变量

输出变量	名称	数据类型	有效范围	内容
Done	参数设置成功标志	BOOL	TRUE、FALSE	TRUE:参数设置成功
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码

（4） 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
Done	✧ 设置成功	✧ Execute 为 TRUE 时，与 Execute 的 FALSE 同时 ✧ Execute 为 FALSE 时，1 个周期后
Busy	✧ Execute 的上升沿	✧ Done 为 TRUE 时 ✧ Error 为 TRUE 时
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时

要点说明

- Execute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数，重新执行功能块。
- 本功能块可以修改对应 ParameterNumber（轴参数序号）的参数，也可修改现场总线协议中定义的参数，如禾川 X3EB 驱动中的 16#6041 状态字。

➢ ※注：修改总线协议参数时，需要在程序里做如下操作

diParameterNumeber 声明为 DINT 类型变量，添加至功能块的输入引脚 ParameterNumeber。

usiDataLength（写想读取的对象字典中的数据长度）声明为 USINT 类型变量。

uiIndex（写想读取的对象字典中的索引）声明为 UINT 类型变量。

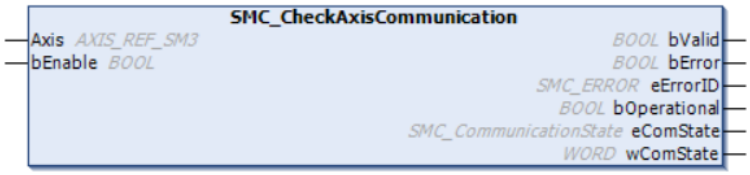
usisubIndex（写想读取的对象字典中的子索引）声明为 DINT 类型变量。diParameterNumeber :=-

DWORD_TO_DINT(SHL(USINT_TO_DWORD(usidatalength),24)+SHL(UINT_TO_DWORD(uiindex),8)+usisubindex);

具体操作可参考 MC_ReadParameter 读取总线协议参数。

SMC_CheckAxisCommunication（FB）

获取轴的当前通信状态。

名称	SMC_CheckAxisCommunication(获取轴当前通信状态)		
支持的模式	CSP	CSV	CST
图形表现		ST 表现	
		SMC_CheckAxisCommunication(Axis:=, bEnable:= , bValid=> , bError=> , eErrorID=> , bOperational=> , eComState=> , wComState=>);	

变量

（1） 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

（2） 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
bEnable	有效	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块 FALSE：关闭功能块

（3） 输出变量

输出变量	名称	数据类型	有效范围	内容
bValid	轴参数数据可获取标志	BOOL	TRUE、FALSE	TRUE:可获取轴参数数据
bError	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
bErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码
bOperational	通讯正常标志	BOOL	TRUE、FALSE	TRUE:通信正常,可对轴进行操作
eComState	通讯状态	SMC_CommunicationState		通信状态，参考 SMC_CommunicationState
wComState	通讯代码	WORD		用于区分当前的通讯状态， 与轴结构变量 wCommunicationState 相同

（4） 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
bValid	✧ 输出有效	✧ bEnable 为 TRUE 时，与 bEnable 的 FALSE 同时 ✧ bEnable 为 FALSE 时，1 个周期后
bError	✧ 功能块执行中产生故障时	✧ 解除了异常时

要点说明

本功能块可以获取轴的通讯状态：

- 当 bEnable 为 TRUE 时，功能块实时更新通讯状态数据。

  当 bEnable 为 FALSE 时，保留上一次更新的数据。

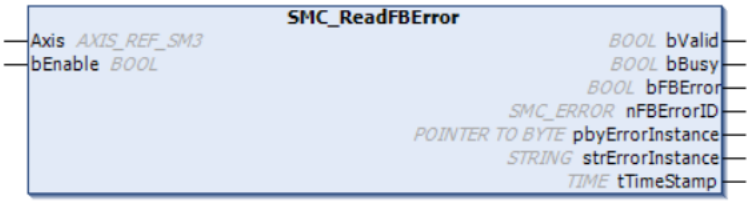
- wComState =100 时 bOperational 为 TRUE，轴通讯正常，可以执行其他功能块操作。

  wComState<100 时，轴通讯初始化中，尚未通讯完成，轴无法运行。

  WComState>100 时，轴通讯出现故障，并且无法通过 MC_Reset 恢复，需在总线重新建立连接后调用 SMC3_ReInitDrive 功能块重新初始化轴。

SMC_ReadFBError(FB)

读取 MC、SMC 功能块错误。

名称	SMC_ReadFBError(读取功能块错误)		
支持的模式	CSP	CSV	CST
图形表现		ST 表现	
		SMC_ReadFBError(Axis:= , bEnable:= , bValid=> , bBusy=> , bFBError=> , nFBErrorID=> , pbyErrorInstance=> , strErrorInstance=> , tTimeStamp=>);	

变量

（1） 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

（2） 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
bEnable	有效	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块 FALSE：关闭功能块

（3） 输出变量

输出变量	名称	数据类型	有效范围	内容
bValid	轴参数数据可获取标志	BOOL	TRUE、FALSE	可获取轴参数数据时，变成 TRUE
bBusy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
bFBError	错误	BOOL	TRUE、FALSE	TRUE:有功能块发生异常
nFBErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码
pbyErrorInstance	错误实例	POINTER TO BYTE		指向报错的功能块
strErrorInstance	错误实例	STRING		发生异常功能块的路径、名称
tTimeStamp	发生错误时间	TIME		功能块异常产生时的时间戳

（4） 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
bValid	✧ 输出有效	✧ bEnable 为 TRUE 时，与 bEnable 的 FALSE 同时 ✧ bEnable 为 FALSE 时，1 个周期后
bBusy	✧ bEnable 的上升沿	✧ bDone 为 TRUE 时 ✧ bError 为 TRUE 时
bFBError	✧ 功能块执行中产生故障时	✧ 解除了异常时

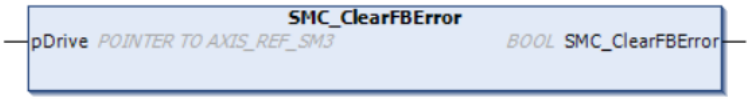
要点说明

- 将 Enable 设为 TRUE 后，SMC_ReadFBError 进入执行状态，此时功能块检测使用此轴的功能块是否有报错情况。
- 将 Enable 设为 FALSE 后，SMC_ReadFBError 将不会再读取功能块报错情况，会保留之前更新的功能块状态信息。如果 Enable 为 FALSE 时有功能块报错，在 Enable 设为 TRUE 时，轴报错信息会被显示出来，但 Enable 重新设为 FALSE 时，输出的报错信息将被复位。
- 清除 SMC_ReadFBError 所读取的功能块报错信息需要使用 SMC_ClearFBError，具体操作参考 SMC_ClearFBError。

➢ ※注：如果有多个相同的 Axis 的功能块报错，SMC_ReadFBError 将只保存第一个报错的功能块信息。

SMC_ClearFBError（FUN）

清除 MC、SMC 功能块错误。

名称	SMC_ClearFBError（清除功能块错误）		
支持的模式	CSP	CSV	CST
图形表现		ST 表现	
		SMC_ClearFBError(pDrive:=ADR( ));	

变量

输入变量

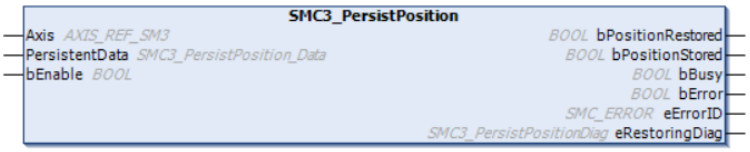
输入变量	名称	数据类型	内容
pDrive	指向轴	POINTER TO AXIS_REF	指针指向 AXIS_REF_SM3

要点说明

- 本功能块是一个功能块异常复位函数，在需要复位功能块异常时调用该函数。
- pDrive 引脚需要取 AXIS_REF_SM3 型变量的地址。

SMC3_PersistPosition（FB）

保存记录实轴绝对值编码器的位置（多圈），不支持虚轴和逻辑轴。

名称	SMC_ PersistPosition（保存位置）		
支持的模式	CSP	CSV	CST
图形表现		ST 表现	
		SMC3_PersistPosition(Axis:=, PersistentData:=, bEnable:= , bPositionRestored=> , bPositionStored=> , bBusy=> , bError=> , eErrorID=> , eRestoringDiag=>);	

变量

（1） 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例
PersistentData	保持数据	SMC3_PersistPosition_Data	用于位置保存

（2） 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
bEnable	有效	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块 FALSE：关闭功能块

（3） 输出变量

输出变量	名称	数据类型	有效范围	内容
bPositionRestored	位置恢复标志	BOOL	TRUE、FALSE	TRUE:轴断电前保存的位置恢复成功
bPositionStored	位置保存标志	BOOL	TRUE、FALSE	TRUE:轴当前位置已保存
bBusy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
bError	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
nErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码
eRestoringDiag	恢复诊断信息	SMC3_PersistPositionDiag	SMC3_PPD_RESTORING_OK	功能块的诊断信息，具体含义参考 SMC3_PersistPositionDiag

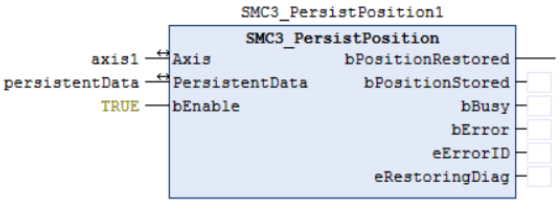
要点说明

- 本功能块仅用于实轴，并且控制器上电后必须优先执行本功能块，功能块 Enable 引脚建议默认为 TRUE。
- 输出变量 bPositionRestored 的状态为 TRUE 时代表上电后功能块加载数据正确。
- 本功能块可以掉电保持 Axis.iTurn 和 Axis.PosOffsetForResiduals 的数据，需要把 SMC3_PersistPosition_Data 配置成掉电保持型变量。
- 本功能块上电后恢复位置数据时，轴的缩放比例等参数必须与断电前一致，所以本功能块与 SMC_ChangeGearingRatio 一起使用时，可能会有问题，需要注意执行顺序。
- SMC3_PersistPosition_Data 配置为掉电保持型变量教程：

  先声明一个 SMC3_PersistPosition_Data 类型实例 persistentData，变量类型为 RETAIN PERSISTENT。

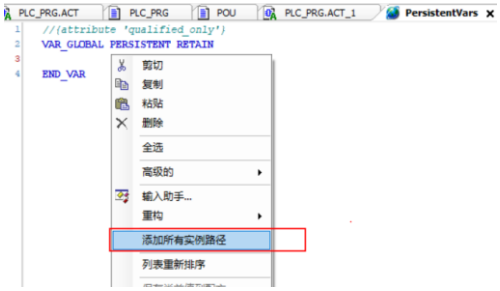
```
VAR PERSISTENT RETAIN
    persistentData: SMC3_PersistPosition_Data;
END_VAR
```

将其添加到 SMC3_PersistPosition 的 PersistentData 引脚上。



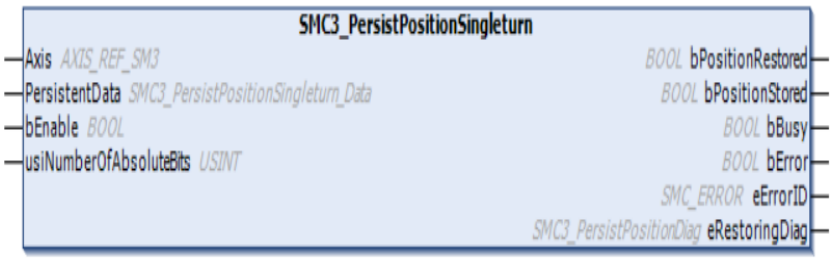
创建一个 PersistentVars 变量（持续变量）

重新编译程序，编译无误后在“PersistentVars”的空白处右键单击，选择添加所有实例路径。



SMC3_PersistPositionSingleturn（FB）

该功能块用来保存记录实轴绝对值编码器（单圈）的位置。

名称	SMC_PersistPositionSingleturn（保存位置）		
支持的模式	CSP	CSV	CST
图形表现		ST 表现	
		SMC3_PersistPositionSingleturn(Axis:=, PersistentData:= , bEnable:= , usiNumberOfAbsoluteBits:= , bPositionRestored=> , bPositionStored=> , bBusy=> , bError=> , eErrorID=> , eRestoringDiag=>);	

变量

（1） 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例
PersistentData	保持数据	SMC3_PersistPositionSingleturn_Data	用于位置保存

（2） 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
bEnable	有效	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块 FALSE：关闭功能块
usiNumberOfAbsoluteBits	绝对编码器数据位	USINT		16	绝对编码器数据位（8-32）

（3） 输出变量

输出变量	名称	数据类型	有效范围	内容
bPositionRestored	位置恢复标志	BOOL	TRUE、FALSE	TRUE:轴断电前保存的位置恢复成功
bPositionStored	位置保存标志	BOOL	TRUE、FALSE	TRUE:轴当前位置已保存
bBusy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
bError	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
nErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码
eRestoringDiag	恢复诊断信息	SMC3_PersistPositionDiag	SMC3_PPD_RESTORING_OK	关于加载位置的动态诊断信息，参考 SMC3_PersistPositionDiag

要点说明

- 本功能块仅用于实轴，并且控制器上电后必须优先执行本功能块，功能块 Enable 引脚建议默认为 TRUE
- 输出变量 bPositionRestored 的状态为 TRUE 时代表上电后功能块加载数据正确。
- 本功能块上电后恢复位置数据时，轴的缩放比例等参数必须与断电前一致，所以本功能块与 SMC_ChangeGearingRatio 一起使用时，可能会有问题，需要注意执行顺序。

➢ ※注：可参考 SMC_ChangeGearingRatio。

SMC_ChangeGearingRatio（FB）

修改电子齿轮比和轴类型。

名称	SMC_ChangeGearingRatio（修改电子齿轮比）		
支持的模式	CSP	CSV	CST
图形表现		ST 表现	
		SMC_ChangeGearingRatio(Axis:= , bExecute:= , dwRatioTechUnitsDenom:= , iRatioTechUnitsNum:= , fPositionPeriod:= , iMovementType:= , bDone=> , bBusy=> , bError=> , nErrorID=>);	

变量

(1) 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

(2) 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
bExecute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块
dwRatioTechUnitsDenom		DWORD		0	将驱动器增量转换为指令单位的转换因子的分母
iRatioTechUnitsNum		DINT		0	将驱动器增量转换为指令单位的转换因子的分子
fPositionPeriod	位置周期	LREAL		0	仅对模态轴有用，位置周期
iMovementType	轴类型选择	SMC_MOVEMENTTYPE	0、1	rotary	0:rotary 模态轴 1:linear 线性轴

(3) 输出变量

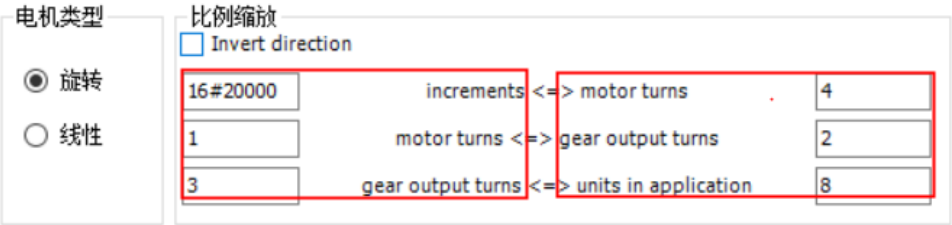
输出变量	名称	数据类型	有效范围	内容
bDone	完成	BOOL	TRUE、FALSE	TRUE：功能块执行完成
bBusy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
bError	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
nErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码

(4) 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
bDone	✧ 复位完成时	✧ bExecute 为 TRUE 时，与 bExecute 的 FALSE 同时 ✧ bExecute 为 FALSE 时，1 个周期后
bBusy	✧ bExecute 的上升沿	✧ bDone 为 TRUE 时 ✧ bError 为 TRUE 时
bError	✧ 功能块执行中产生故障时	✧ 解除了异常时

要点说明

- bExecute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数，重新执行功能块。
- 轴的齿轮比和轴类型可以通过本功能块修改，bDone 为 TRUE 后，可能需要通过 SMC3_ReInitDrive 重新启动轴，以确保所有变量都成功初始化。
- 下面对 dwRatioTechUnitsDenom 和 iRatioTechUnitsNum 作简单说明：



dwRatioTechUnitsDenom 由左边三个参数相乘，iRatioTechUnitsNum 由右边三个参数相乘，请注意，如果求出的 dwRatioTechUnitsDenom 与 iRatioTechUnitsNum 之间可以通分，软件会自动进行通分。

dwRatioTechUnitsDenom：=increments（16#20000） * motor turns（1） * gear output turns（3）；

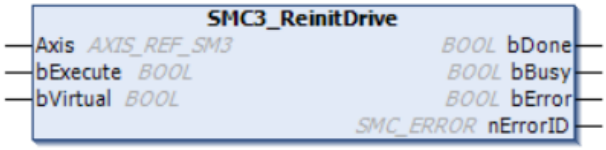
iRatioTechUnitsNum：= motor turns（4） * gear output turns（2） * units in application（8）；

dwRatioTechUnitsDenom	DWORD	6144			Parameter number 1051
iRatioTechUnitsNum	DINT	1			Parameter number 1052

所求得参数可在轴结构体中查看（AXIS_REF_SMC3）。

SMC3_ReInitDrive（FB）

重新启动轴。

名称	SMC3_ReInitDrive（重启轴）		
支持的模式	CSP	CSV	CST
图形表现		ST 表现	
		SMC3_ReinitDrive(Axis:= , bExecute:= , bVirtual:= , bDone=> , bBusy=> , bError=> , nErrorID=>);	

变量

(1) 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

(2) 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
bExecute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块
bVirtual	虚拟	BOOL	TRUE、FALSE	FALSE	TRUE：虚轴

(3) 输出变量

输出变量	名称	数据类型	有效范围	内容
bDone	完成	BOOL	TRUE、FALSE	TRUE：功能块执行完成
bBusy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
bError	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
nErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码

(4) 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
bDone	复位完成时	bExecute 为 TRUE 时，与 bExecute 的 FALSE 同时 bExecute 为 FALSE 时，1 个周期后
bBusy	bExecute 的上升沿	bDone 为 TRUE 时 bError 为 TRUE 时
bError	功能块执行中产生故障时	解除了异常时

要点说明

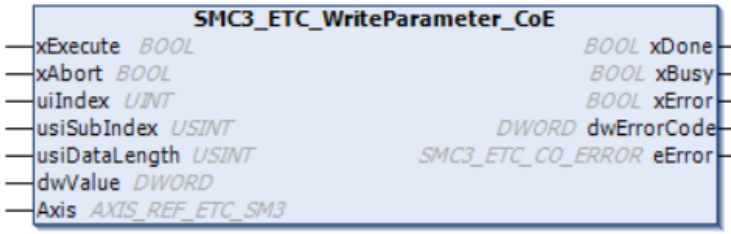
- bExecute 为 TRUE 时，轴会重新初始化，初始化完成之前轴无法进行任何操作。
- 如果 bVirtual 为 TRUE，轴会被判断为虚轴，将作为虚轴存在，对现场总线设备没有影响，可以正常运行，但不会与真实设备进行通讯。
- 功能块运行时轴状态为 Power_off。

功能块运行前轴状态	功能块完成后轴状态	备注
Power_Off	Power_Off	
Errorstop	Power_Off	
Standstill	Standstill	
Homing	Standstill	功能块（SMC3_ReInitDrive）Execute 为 TRUE 时，MC_Home 的输出引脚 Done 直接为 TRUE
Continuous Motion	Standstill	原本的运动控制功能块会报错 报错内容为：“SMC_AXIS_NOT_READY_FOR_MOTION”
Synchronized Motion	Standstill	
Discrete Motion	Standstill	
Stopping	Standstill	MC_Stop 会报错“SMC_REGULATOR_OR_START_NOT_SET”

SMC3_ETC_WriteParameter_CoE（FB）

修改 EtherCAT 参数。

名称	SMC3_ETC_WriteParameter_CoE（修改 EtherCAT 参数）		
支持的模式	CSP	CSV	CST
图形表现		ST 表现	




```
SMC3_ETC_WriteParameter_CoE(
xExecute:= ,
xAbort:= ,
uiIndex:= ,
usiSubIndex:= ,
usiDataLength:= ,
dwValue:= ,
Axis:= ,
xDone=> ,
xBusy=> ,
xError=> ,
dwErrorCode=> ,
eError=> );
```

变量

(1) 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

(2) 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
xExecute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块
xAbort	中断	BOOL	TRUE、FALSE	FALSE	TRUE ：中断功能块
uiIndex	索引	UINT		0	对象字典的索引
usiSubIndex	子索引	USINT		0	对象字典的子索引
usiDataLength	数据长度	USINT		0	对象字典的数据长度
dwValue	参数设定	DWORD		0	修改数值

(3) 输出变量

输出变量	名称	数据类型	有效范围	内容
xDone	参数设置成功标志	BOOL	TRUE、FALSE	TRUE:功能块执行完成
xBusy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
xError	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
dwErrorCode	错误代码	DWORD	0	通过 CANopen 转换 SDO 的时候发生错误
eError	错误	SMC3_ETC_CO_ERROR		功能块报警

(4) 输出变量的转化时序

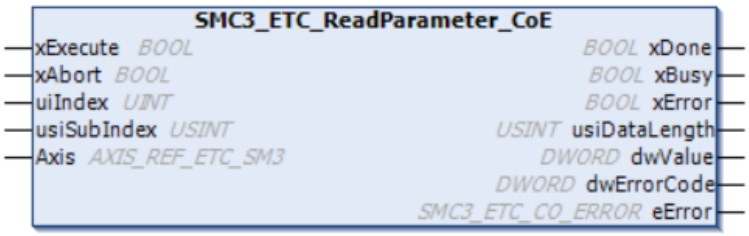
变量	变为 TRUE 时	变为 FALSE 时
xDone	✧ 复位完成时	✧ xExecute 为 TRUE 时，与 xExecute 的 FALSE 同时 ✧ xExecute 为 FALSE 时，1 个周期后
xBusy	✧ xExecute 的上升沿	✧ xDone 为 TRUE 时 ✧ bError 为 TRUE 时
xError	✧ 功能块执行中产生故障时	✧ 解除了异常时

要点说明

- xExecute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数，重新执行功能块。
- 本功能块可以修改现场总线协议中定义的参数，以修改对象字典 16#6098（回原模式）为例，uiIndex（索引）赋值为 16#6098，usiSubIndex（子索引）赋值为 16#00，usiDataLength（数据长度）赋值为 1，dwValue（设定参数）赋值为 30，此时 xExecute 设置为 TRUE，功能块 xDone 变为 TRUE 时，回原模式修改成功。

SMC3_ETC_ReadParameter_CoE（FB）

获取 EtherCAT 参数。

名称	SMC3_ETC_ReadParameter_CoE（获取 EtherCAT 参数）		
支持的模式	CSP	CSV	CST
图形表现		ST 表现	
		SMC3_ETC_ReadParameter_CoE(xExecute:= , xAbort:= , uiIndex:= , usiSubIndex:= , Axis:= , xDone=> , xBusy=> , xError=> , usiDataLength=> , dwValue=> , dwErrorCode=> , eError=>);	

变量

(1) 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

(2) 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
xExecute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块
xAbort	中断	BOOL	TRUE、FALSE	FALSE	TRUE ：中断功能块
uiIndex	索引	UINT		0	对象字典的索引
usiSubIndex	子索引	USINT		0	对象字典的子索引

(3) 输出变量

输出变量	名称	数据类型	有效范围	内容
xDone	参数读取成功标志	BOOL	TRUE、FALSE	TRUE:功能块执行完成
xBusy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
xError	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
dwErrorCode	错误代码	DWORD	0	通过 CANopen 转换 SDO 的时候发生错误
eError	错误	SMC3_ETC_CO_ERROR		功能块报警
usiDataLength	数据长度	USINT		对象字典的数据长度
dwValue	参数设定	DWORD		读取数值

(4) 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
xDone	✧ 复位完成时	✧ xExecute 为 TRUE 时，与 xExecute 的 FALSE 同时 ✧ xExecute 为 FALSE 时，1 个周期后
xBusy	✧ xExecute 的上升沿	✧ xDone 为 TRUE 时 ✧ xError 为 TRUE 时
xError	✧ 功能块执行中产生故障时	✧ 解除了异常时

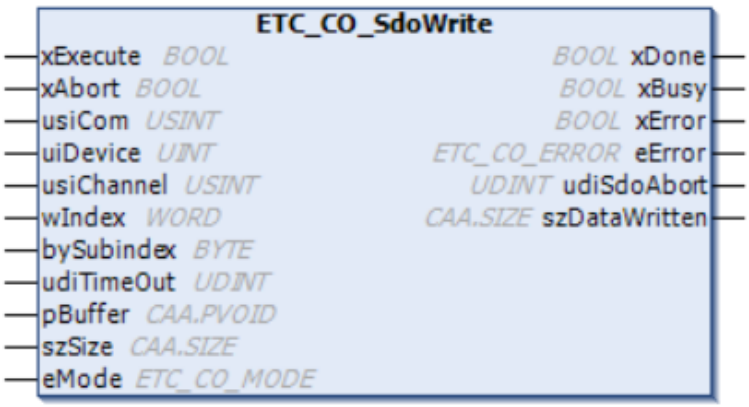
要点说明

- xExecute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数，重新执行功能块。
- 本功能块可以读取现场总线协议中定义的参数，以读取对象字典 16#6080（最大电机速度）为例，uiIndex（索引）赋值为 16#6080，usiSubIndex（子索引）赋值为 16#00，此时 bExecute 设置为 TRUE，功能块 bDone 变为 TRUE 时，最大电机速度读取成功， usiDataLength（数据长度）、dwValue（数值）显示当前参数值。

ETC_CO_SdoWrite（FB）

修改从站参数。

名称	ETC_CO_SdoWrite（修改从站参数）		
支持的模式	CSP	CSV	CST
图形表现		ST 表现	




```
ETC_CO_SdoWrite(
xExecute:= ,
xAbort:= ,
usiCom:= ,
uiDevice:= ,
usiChannel:= ,
wIndex:= ,
bySubindex:= ,
udiTimeOut:= ,
pBuffer:= ,
szSize:= ,
eMode:= ,
xDone=> ,
xBusy=> ,
xError=> ,
eError=> ,
udiSdoAbort=> ,
szDataWritten=> );
```

变量

(1) 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
xExecute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块
xAbort	中断	BOOL	TRUE、FALSE	FALSE	TRUE ：中断功能块
usiCom	主站索引	USINT		1	主站索引 1：第一个主站
uiDevice	从站地址	UINT		0	用于访问 EtherCAT 从站地址
usiChannel	通道	USINT		1	不需要设置，这个是自动设置的
uiIndex	索引	UINT		0	对象字典的索引
usiSubIndex	子索引	USINT		0	对象字典的子索引
udiTimeOut	操作延时	UDINT		0	设置延时时间，单位 ms
pBuffer	缓冲区	CAA.PVOID		0	用于写入数据的指向缓冲区的指针
szSize	数据大小	CAA.SIZE		0	传输数据大小，单位 byte
eMode	模式选择	ETC_CO_MODE		0	传输模式：ETC_CO_AUTO；ETC_CO_EXPEDITED；ETC_CO_SEGMENTED（ethercat 不支持）

(2) 输出变量

输出变量	名称	数据类型	有效范围	内容
xDone	参数设置成功标志	BOOL	TRUE、FALSE	TRUE:功能块执行完成
xBusy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
xError	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
eError	错误	ETC_CO_ERROR		功能块报警
udiSdoAbort	SDO 中断	UDINT		从设备接收到的中止代码
szDataWritten	数据大小	CAA.SIZE		传输数据大小，单位 byte

(3) 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
xDone	✧ 复位完成时	✧ xExecute 为 TRUE 时，与 xExecute 的 FALSE 同时 ✧ xExecute 为 FALSE 时，1 个周期后
xBusy	✧ xExecute 的上升沿	✧ xDone 为 TRUE 时 ✧ xError 为 TRUE 时
xError	✧ 功能块执行中产生故障时	✧ 解除了异常时

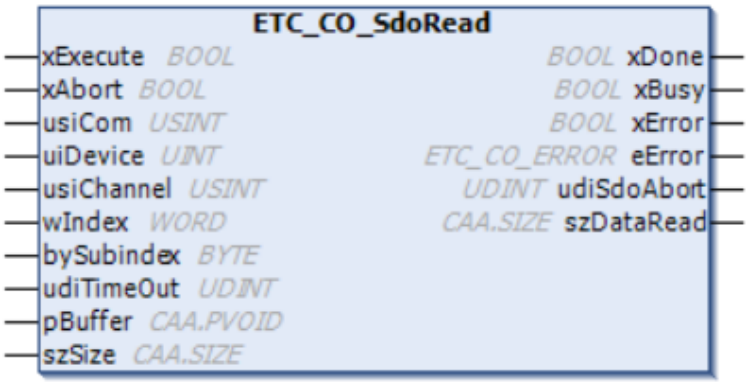
要点说明

- xExecute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数，重新执行功能块。
- 这个功能块用于写入 EtherCAT 从站参数，要写入参数需要用 uiIndex 和 usiSubIndex 指定，pBuffer 需要用到指针。
- szSize 单位大小为 byte，即 szSize=1 为 8 位。

- eMode 中 ETC_CO_EXPEDITED 传输速度比 ETC_CO_AUTO 更快。
- uiDevice 的地址查询如下图所示：

ETC_CO_SdoRead（FB）

读取从站参数。

名称	ETC_CO_SdoRead（读取从站参数）		
支持的模式	CSP	CSV	CST
图形表现		ST 表现	
		ETC_CO_SdoRead(xExecute:= , xAbort:= , usiCom:= , uiDevice:= , usiChannel:= , wIndex:= , bySubindex:= , udiTimeOut:= , pBuffer:= , szSize:= , xDone=> , xBusy=> , xError=> , eError=> , udiSdoAbort=> , szDataRead=>);	

变量

(1) 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
xExecute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块
xAbort	中断	BOOL	TRUE、FALSE	FALSE	TRUE ：中断功能块
usiCom	主站索引	USINT		1	主站索引 1：第一个主站
uiDevice	从站地址	UINT		0	用于访问 EtherCAT 从站地址
usiChannel	通道	USINT		1	不需要设置，这个是自动设置的
uiIndex	索引	UINT		0	对象字典的索引
usiSubIndex	子索引	USINT		0	对象字典的子索引
udiTimeOut	操作延时	UDINT		0	设置延时时间，单位 ms
pBuffer	缓冲区	CAA.PVOID		0	用于写入数据的指向缓冲区的指针
szSize	数据大小	CAA.SIZE		0	传输数据大小，单位 byte

(2) 输出变量

输出变量	名称	数据类型	有效范围	内容
xDone	参数设置成功标志	BOOL	TRUE、FALSE	TRUE:功能块执行完成
xBusy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
xError	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
eError	错误	ETC_CO_ERROR		功能块报警
udiSdoAbort	SDO 中断	UDINT		从设备接收到的中止代码
szDataWritten	数据大小	CAA.SIZE		传输数据大小，单位 byte

(3) 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
xDone	✧ 复位完成时	✧ xExecute 为 TRUE 时，与 xExecute 的 FALSE 同时 ✧ xExecute 为 FALSE 时，1 个周期后
xBusy	✧ xExecute 的上升沿	✧ xDone 为 TRUE 时 ✧ xError 为 TRUE 时
xError	✧ 功能块执行中产生故障时	✧ 解除了异常时

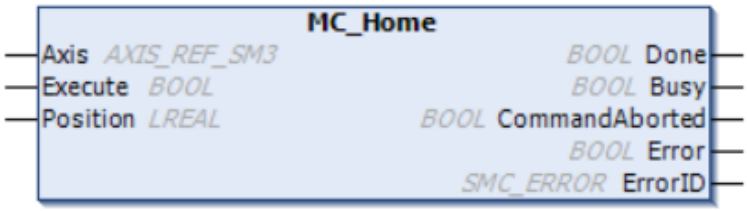
要点说明

- xExecute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数，重新执行功能块。
- 这个功能块用于读取 EtherCAT 从站参数，要读取参数需要用 uiIndex 和 usiSubIndex 指定，pBuffer 需要用到指针。
- szSize 单位大小为 byte，即 szSize=1 为 8 位。
- uiDevice 的地址查询如下图所示：

单轴运动控制

MC_Home（FB）

控制轴进行驱动器回原。

名称	MC_Home（轴回原）		
支持的模式	CSP		
图形表现		ST 表现	
		MC_Home(Axis:= , Execute:= , Position:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);	

变量

(1) 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

(2) 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块
Position	原点设定位置	LREAL	负数、正数、“0”	0	回原完成后，原点的设定位置

(3) 输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	TRUE：功能块执行完成
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
CommandAborted	功能块执行中断	BOOL	TRUE、FALSE	TRUE:功能块被中止
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码

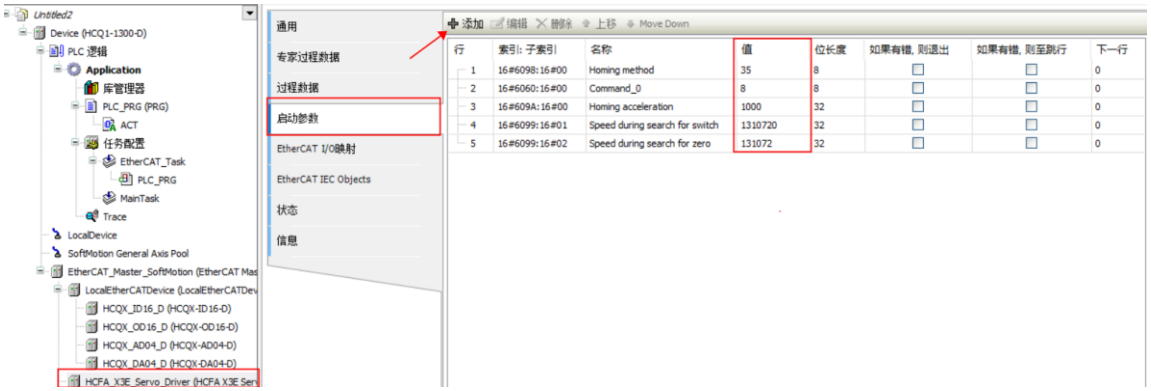
(4) 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
Done	✧ 回原完成时	✧ Execute 为 TRUE 时，与 Execute 的 FALSE 同时 ✧ Execute 为 FALSE 时，1 个周期后
Busy	✧ Execute 的上升沿	✧ Done 为 TRUE 时 ✧ Error 为 TRUE 时 ✧ CommandAborted 变为 TRUE 时
CommandAborted	✧ 由其他功能块进行运动功能块多重启动（中断），本功能块中止时 ✧ 因发生异常，本功能块中止时 ✧ 正在发生异常时，启动了本功能块 ✧ 正在执行 MC_Stop 功能块时，启动了本功能块	✧ Execute 为 TRUE 时，与 Execute 的 FALSE 同时 ✧ Execute 为 FALSE 时，1 个周期后
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时

要点说明

- Execute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数，重新执行功能块。
- 轴在 Standstill 状态才可执行 MC_Home。
- MC_Home 执行时，无法执行其他常规运动功能块，可使用 MC_Stop 停止回原。
- 在使用前需要配置好对应的回原参数：16#6098、16#6099、16#609A（具体回原参数详见驱动器手册）。

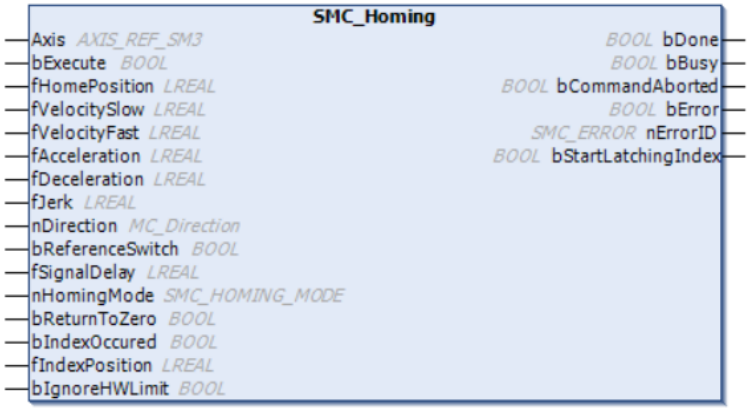
➢ ※注：回原参数建议在启动参数里添加，以禾川 X3EB 驱动器为例，回原参数需要注意伺服 P09.13 参数，P09.13 参数右起第三位可以设置速度单位类型，为 1 时速度单位为用户指令/S，当为 0 时速度单位为转每分钟（rpm）。下图添加的 SDO 参数及数值仅做参考，具体设置请根据实际回原要求设置。



SMC_Homing（FB）

控制轴进行控制器回原。

名称	SMC_Homing（轴回原）		
支持的模式	CSP		
图形表现		ST 表现	




```
SMC_Homing(
Axis:= ,
bExecute:= ,
fHomePosition:= ,
fVelocitySlow:= ,
fVelocityFast:= ,
fAcceleration:= ,
fDeceleration:= ,
fJerk:= ,
nDirection:= ,
bReferenceSwitch:= ,
fSignalDelay:= ,
nHomingMode:= ,
bReturnToZero:= ,
bIndexOccured:= ,
fIndexPosition:= ,
bIgnoreHWLimit:= ,
bDone=> ,
bBusy=> ,
bCommandAborted=> ,
bError=> ,
nErrorID=> ,
bStartLatchingIndex=> );
```

变量

(1) 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

(2) 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
bExecute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块
fHomePosition	原点设定位置	LREAL	负数、正数、“0”	0	回原完成后，原点的设定位置
fVelocitySlow	慢速度	LREAL	正数	0	回原慢速运行的速度

fVelocityFast	快速度	LREAL	正数	0	回原快速运行的速度
fAcceleration	加速度	LREAL	正数或“0”	0	加速度，单位为【指令单位/s^2】
fDeceleration	减速度	LREAL	正数或“0”	0	减速度，单位为【指令单位/s^2】
fJerk	加加速度	LREAL	正数或“0”	0	指定加加速度，单位为【指令单位/S^3】
nDirection	方向选择	MC_Direction		negative	回原的方向
bReferenceSwitch	原点参考开关	BOOL	TRUE、FALSE	TRUE	轴不在原点参考开关（bReferenceSwitch）上时，原点参考开关为 TRUE，轴碰到原点参考开关时，原点参考开关为 FALSE，轴离开参考后，原点参考开关为 TRUE，回原完成
fSignalDelay	信号延迟	LREAL		0	原点参考开关的传输延时，单位为 S
nHomingMode	回原模式	SMC_Homing_Mode		FAST_BSLOW_S_STOP	回原模式选择，参考 SMC_Homing_Mode
bReturnToZero	返回原位	BOOL	TRUE、FALSE	FALSE	TRUE：回原完成后，若 fHomePosition 设置不为 0，轴会按照 fVelocityFast 的速度运行至 0 的位置
bIndexOccured		BOOL	TRUE、FALSE	FALSE	TRUE：回原模式为 FAST_BSLOW_I_S_STIO 或 FAST_SLOW_I_S_STOP 时生效
fIndexPosition		LREAL		0	标志脉冲时记录的位置
bIgnoreHWLimit	忽略硬件限位	BOOL	TRUE、FALSE	FALSE	TRUE:回原时忽略轴的硬限位功能

(3) 输出变量

输出变量	名称	数据类型	有效范围	内容
bDone	完成	BOOL	TRUE、FALSE	TRUE：功能块执行完成
bBusy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
bCommandAborted	功能块执行中断	BOOL	TRUE、FALSE	TRUE:功能块被中止
bError	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
nErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码
bStartLatchingIndex	启动记录脉冲	BOOL	TRUE、FALSE	取决于 bIndexOccured 以及 fIndexPosition.的设置

(4) 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
bDone	✧ 回原完成时	✧ bExecute 为 TRUE 时，与 bExecute 的 FALSE 同时 ✧ bExecute 为 FALSE 时，1 个周期后
bBusy	✧ bExecute 的上升沿	✧ bDone 为 TRUE 时 ✧ bError 为 TRUE 时 ✧ CommandAborted 变为 TRUE 时
bCommandAborted	✧ 由其他功能块进行运动功能块多重启动（中断），本功能块中止时 ✧ 因发生异常，本功能块中止时 ✧ 正在发生异常时，启动了本功能块 ✧ 正在执行 MC_Stop 功能块时，启动了本功能块	✧ bExecute 为 TRUE 时，与 bExecute 的 FALSE 同时 ✧ bExecute 为 FALSE 时，1 个周期后
bError	✧ 功能块执行中产生故障时	✧ 解除了异常时

要点说明

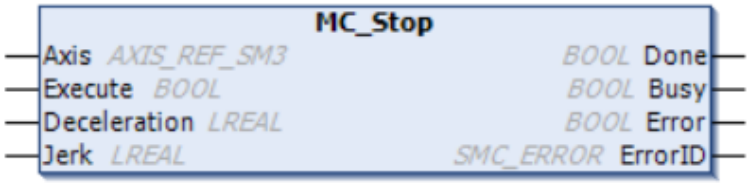
- bExecute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数，重新执行功能块。
- 该功能块是控制器控制的回原方式，与 MC_Home 有本质不同：

  MC_Home 为驱动器控制的回原，精度更高，并且会修改驱动器实际位置；

  SMC_Homing 不会修改驱动器实际位置，仅是通过 MC_SetPosition 进行位置设定，断电后可能会丢失原点位置。
- 当 bExecute 的上升沿启动之后，SMC_Homing 被执行，只要原点参考开关 bReferenceSwitch 为 TRUE，轴以 fVelocityFast 的速度在 nDirection 指定的方向移动。当原点参考开关 bReferenceSwitch 为 FALSE，轴以慢速 fVelocitySlow 离开原点参考开关。当原点参考开关 bReferenceSwitch 再次为 TRUE，按照选择的回原模式 nHomingMode 动作，具体参考 SMC_Homing_Mode。

MC_Stop（FB）

停止一个轴的运动，并将轴切换为 Stopping 状态。

名称	MC_Stop（轴停止）		
支持的模式	CSP	CSV	
图形表现		ST 表现	
		MC_Stop(Axis:=, Execute:= , Deceleration:= , Jerk:= , Done=> , Busy=> , Error=> , ErrorID=>);	

变量

(1) 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

(2) 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块
Deceleration	减速度	LREAL	正数或“0”	0	减速度，单位为【指令单位/s^2】
Jerk	加加速度	LREAL	正数或“0”	0	指定加加速度，单位为【指令单位/S^3】

(3) 输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	TRUE：功能块执行完成
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码

要点说明

- Execute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数，重新执行功能块。
- 本功能块是在正常运行的情况下停止一个轴的运动，并且将轴切换为 Stopping 状态，当轴处于 Stopping 状态任何运动功能块都是无效的。
- 即使 MC_Stop 停止轴运动已完成，Done 已变为 TRUE，只要不将 Execute 设置为 FALSE,那么轴仍旧会保持为 Stopping 状态。
- 多个 MC_Stop 互相无法打断，所以建议保证程序中一个轴只由一个 MC_Stop 控制，常规停止建议使用 MC_Halt。

➢ ※注：Mc_Stop 在转矩模式下不可用！

MC_Halt（FB）

停止一个轴的运动。

名称	MC_Halt(轴停止)		
支持的模式	CSP	CSV	
图形表现		ST 表现	
		MC_Halt(Axis:= , Execute:= , Deceleration:= , Jerk:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);	

变量

(1) 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

(2) 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块
Deceleration	减速度	LREAL	正数或“0”	0	减速度，单位为【指令单位/s^2】
Jerk	加加速度	LREAL	正数或“0”	0	指定加加速度，单位为【指令单位/S^3】

(3) 输出变量

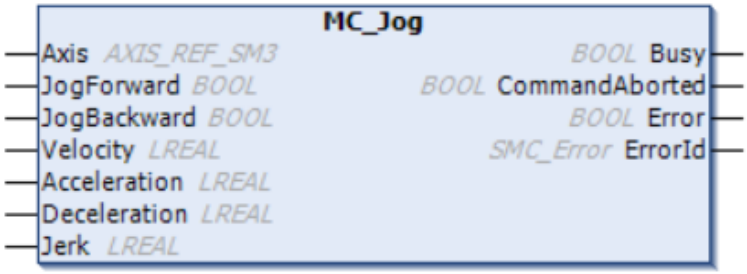
输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	TRUE：功能块执行完成
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
CommandAborted	功能块执行中断	BOOL	TRUE、FALSE	TRUE:功能块被中止
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码

要点说明

- Execute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数，重新执行功能块。
- 本功能块是在正常运行的情况下停止一个轴的运动， 停止过程中如有其他运动功能块触发，会打断本功能块的执行。
- 功能块执行中的状态为 Discrete_motion，运行完成后状态为 Standstill。
- MC_Halt 不同于 MC_Stop，MC_Halt 允许被打断。

MC_Jog（FB）

控制轴进行正反点动。

名称	MC_Jog（轴点动控制）		
支持的模式	CSP	CSV	
图形表现		ST 表现	
		MC_Jog(Axis:=, JogForward:= , JogBackward:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Busy=> , CommandAborted=> , Error=> , ErrorId=>);	

变量

(1) 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

(2) 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
JogForward	正向点动	BOOL	TRUE、FALSE	FALSE	TRUE：轴向正方向移动 FALSE：轴停止向正方向移动
JogBackward	反向点动	BOOL	TRUE、FALSE	FALSE	TRUE：轴向反方向移动 FALSE：轴停止向反方向移动
Velocity	目标速度	LREAL	正数	0	指定目标速度。单位为【指令单位/S】
Acceleration	加速度	LREAL	正数或“0”	0	加速度，单位为【指令单位/s^2】
Deceleration	减速度	LREAL	正数或“0”	0	减速度，单位为【指令单位/s^2】
Jerk	加加速度	LREAL	正数或“0”	0	指定加加速度，单位为【指令单位/S^3】

(3) 输出变量

输出变量	名称	数据类型	有效范围	内容
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
CommandAborted	功能块执行中断	BOOL	TRUE、FALSE	TRUE:功能块被中止
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码

(4) 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
Busy	✧ JogForward 或 JogBackward 为 TRUE	✧ Error 为 TRUE 时 ✧ CommandAborted 变为 TRUE 时
CommandAborted	✧ 由其他功能块进行运动功能块多重启动（中断），本功能块中止时 ✧ 因发生异常，本功能块中止时 ✧ 正在发生异常时，启动了本功能块 ✧ 正在执行 MC_Stop 功能块时，启动了本功能块	✧ Execute 为 TRUE 时，与 Execute 的 FALSE 同时 ✧ Execute 为 FALSE 时，1 个周期后
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时

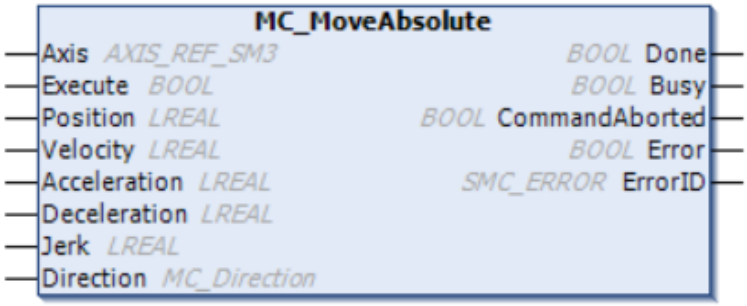
要点说明

- 控制轴点动运行，正向点动由 JogForward 控制，设置为 TRUE 轴即会按照设定的速度、加速度进行正向点动。反向点动由 JogBackward 控制，设置为 TRUE 轴即会按照设定的速度、加速度进行反向点动。
- 点动运行时，如果 JogForward 或者 JogBackwar 从 TRUE 变为 FALSE，轴便会立即按照设定的减速度减速停止。
- 点动过程中轴处于 Continuous Motion 状态。
- 如果 JogForward 和 JogBackward 同时设置为 TRUE，轴将停止运动或不开始运动且不会报错。

MC_MoveAbsolute（FB）

控制轴以绝对目标位置进行定位运动。

名称	MC_MoveAbsolute（轴绝对位置控制）		
支持的模式	CSP		
图形表现		ST 表现	




```
MC_MoveAbsolute(
Axis:=,
Execute:= ,
Position:= ,
Velocity:= ,
Acceleration:= ,
Deceleration:= ,
Jerk:= ,
Direction:= ,
Done=> ,
Busy=> ,
CommandAborted=> ,
Error=> ,
```

变量

(1) 输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	AXIS_REF_SM3		指定轴，即 AXIS_REF_SM3 的一个实例

(2) 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块
Position	目标位置	LREAL	负数、正数、“0”	0	指定绝对坐标的目标位置，单位为【指令单位】
Velocity	目标速度	LREAL	正数	0	指定目标速度，单位为【指令单位/S】
Acceleration	加速度	LREAL	正数或“0”	0	加速度，单位为【指令单位/s^2】
Deceleration	减速度	LREAL	正数或“0”	0	减速度，单位为【指令单位/s^2】
Jerk	加加速度	LREAL	正数或“0”	0	指定加加速度，单位为【指令单位/S^3】
Direction	方向选择	MC_Direction	fastest,current, positive,shortest, negative	shortest	参考 MC_Direction

(3) 输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	TRUE：功能块执行完成
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
CommandAborted	功能块执行中断	BOOL	TRUE、FALSE	TRUE:功能块被中止
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码

(4) 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
Done	✧ 定位完成时	✧ Execute 为 TRUE 时，与 Execute 的 FALSE 同时 ✧ Execute 为 FALSE 时，1 个周期后
Busy	✧ Execute 的上升沿	✧ Done 为 TRUE 时 ✧ Error 为 TRUE 时 ✧ CommandAborted 变为 TRUE 时
CommandAborted	✧ 由其他功能块进行运动功能块多重启动（中断），本功能块中止时 ✧ 因发生异常，本功能块中止时 ✧ 正在发生异常时，启动了本功能块 ✧ 正在执行 MC_Stop 功能块时，启动了本功能块	✧ Execute 为 TRUE 时，与 Execute 的 FALSE 同时 ✧ Execute 为 FALSE 时，1 个周期后
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时

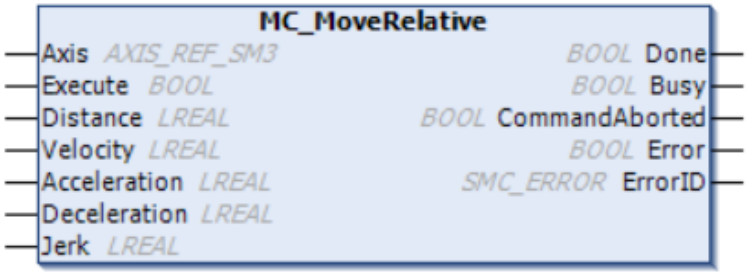
要点说明

- 本功能块为绝对定位功能块，Position 数据为轴的绝对位置，本功能块执行时轴的状态为 Discrete Motion。
- Execute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数，重新执行功能块。
- Acceleration 或 Deceleration 为零，功能块启动（Execute）会报错，轴的状态为 Standstill 。
- Direction(方向选择)，参考 MC_Direction。

MC_MoveRelative（FB）

控制轴以相对目标位置进行定位运动。

名称	MC_MoveRelative（相对定位）		
支持的模式	CSP		
图形表现		ST 表现	




```
MC_MoveRelative(
Axis:=,
Execute:= ,
Distance:= ,
Velocity:= ,
Acceleration:= ,
Deceleration:= ,
Jerk:= ,
Done=> ,
Busy=> ,
CommandAborted=> ,
Error=> ,
```

变量

(1) 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

(2) 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块
Distance	目标相对位置	LREAL	负数、正数、“0”	0	指定相对坐标的目标位置，单位为【指令单位】
Velocity	目标速度	LREAL	正数	0	指定目标速度，单位为【指令单位/S】
Acceleration	加速度	LREAL	正数或“0”	0	加速度，单位为【指令单位/s^2】

Deceleration	减速度	LREAL	正数或“0”	0	减速度，单位为【指令单位/s^2】
Jerk	加加速度	LREAL	正数或“0”	0	指定加加速度，单位为【指令单位/S^3】

(3) 输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	TRUE：功能块执行完成
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
CommandAborted	功能块执行中断	BOOL	TRUE、FALSE	TRUE:功能块被中止
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码

(4) 输出变量的转化时序

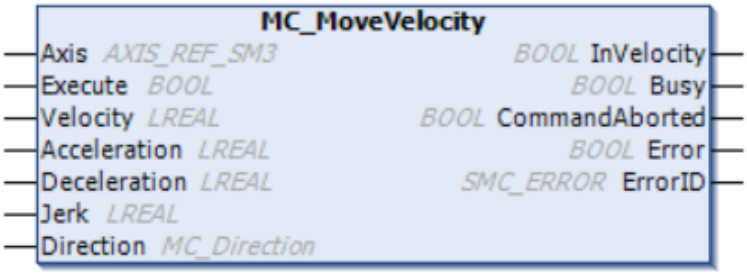
变量	变为 TRUE 时	变为 FALSE 时
Done	✧ 定位完成时	✧ Execute 为 TRUE 时，与 Execute 的 FALSE 同时 ✧ Execute 为 FALSE 时，1 个周期后
Busy	✧ Execute 的上升沿	✧ Error 为 TRUE 时 ✧ CommandAborted 变为 TRUE 时
CommandAborted	✧ 由其他功能块进行运动功能块多重启动（中断），本功能块中止时 ✧ 因发生异常，本功能块中止时 ✧ 正在发生异常时，启动了本功能块 ✧ 正在执行 MC_Stop 功能块时，启动了本功能块	✧ Execute 为 TRUE 时，与 Execute 的 FALSE 同时 ✧ Execute 为 FALSE 时，1 个周期后
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时

要点说明

- 本功能块为相对定位功能，Distance 为轴要移动的相对距离。
- 功能块执行时轴处于 Discrete Motion 状态。
- Execute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数。
- Acceleration 或 Deceleration 为零，指定运行时会报错，轴的状态为 Standstill 。
- 在 Execute 得到上升沿动作的同时，Busy 变为 TRUE。到达目标位后，定位完成，Done 变为 TRUE，Busy 变为 FALSE。
- 当本功能块被中断时，CommandAborted 变为 TRUE，Busy 变为 FALSE。

MC_MoveVelocity（FB）

控制轴进行匀速运动。

名称	MC_MoveVelocity（速度控制）		
支持的模式	CSP	CSV	
图形表现		ST 表现	
		MC_MoveVelocity(Axis:= , Execute:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Direction:= , InVelocity=> , Busy=> , CommandAborted=> ,	

变量

(1) 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

(2) 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块
Velocity	目标速度	LREAL	正数	0	指定目标速度，单位为【指令单位/S】
Acceleration	加速度	LREAL	正数或“0”	0	加速度，单位为【指令单位/s^2】
Deceleration	减速度	LREAL	正数或“0”	0	减速度，单位为【指令单位/s^2】
Jerk	加加速度	LREAL	正数或“0”	0	指定加加速度，单位为【指令单位/S^3】
Direction	方向选择	MC_Direction	fastest,current, positive,shortest, negative	shortest	参考 MC_Direction

(3) 输出变量

输出变量	名称	数据类型	有效范围	内容
InVelocity	达到了设定速度	BOOL	TRUE、FALSE	TRUE：到达了设定的目标速度
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
CommandAborted	功能块执行中断	BOOL	TRUE、FALSE	TRUE:功能块被中止
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码

(4) 输出变量的转化时序

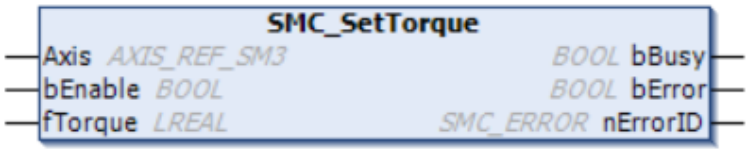
变量	变为 TRUE 时	变为 FALSE 时
InVelocity	✧ 目标速度到达设定值	✧ 速度没有到达目标速度的设定值 ✧ Error 为 TRUE 时 ✧ CommandAborted 变为 TRUE 时
Busy	✧ Execute 的上升沿	✧ Error 为 TRUE 时 ✧ CommandAborted 变为 TRUE 时
CommandAborted	✧ 由其他功能块进行运动功能块多重启动（中断），本功能块中止时 ✧ 因发生异常，本功能块中止时 ✧ 正在发生异常时，启动了本功能块 ✧ 正在执行 MC_Stop 功能块时，启动了本功能块	✧ Execute 为 TRUE 时，与 Execute 的 FALSE 同时 ✧ Execute 为 FALSE 时，1 个周期后
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时

要点说明

- 本功能块控制轴以设定的速度不间断地运动。
- 功能块执行时轴处于 Continuous Motion 状态。
- Execute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数。
- 轴的状态为 Standstill 时，MC_MoveVelocity 的加速度或减速度为 0，功能块会报错，轴状态为 Standstill。
- 轴的状态为 Continuous Motion 时，MC_MoveVelocity 的加速度或减速度为 0，功能块和轴都会报错，轴状态为 Errorstop。
- 功能块执行时 Execute 引脚设置为 FALSE 并不会停止功能块执行，轴仍旧会按照设定的参数继续执行，如要停止 MC_MoveVelocity，可以执行 MC_Halt 进行停止。

SMC_SetTorque(FB)

控制轴输出设定的转矩（转矩模式可用）。

名称	SMC_SetTorque(转矩设定)		
支持的模式	CST		
图形表现		ST 表现	
		SMC_SetTorque(Axis:=, bEnable:= , fTorque:= , bBusy=> , bError=> , nErrorID=>);	

变量

(1) 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

(2) 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
bEnable	有效	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块 FALSE：关闭功能块
fTorque	设定的转矩	LREAL		0	单位：Nm

(3) 输出变量

输出变量	名称	数据类型	有效范围	内容
bBusy	功能块执行中	BOOL	TRUE、FALSE	收到功能块时变为 TRUE
bError	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
nErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码

(4) 输出变量的转化时序

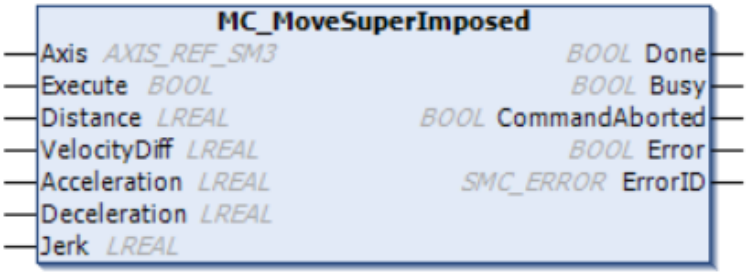
变量	变为 TRUE 时	变为 FALSE 时
bBusy	✧ bEnable 为 TRUE 时	✧ Done 为 TRUE 时 ✧ Error 为 TRUE 时 ✧ CommandAborted 变为 TRUE 时
bError	✧ 功能块执行中产生故障时	✧ 解除了异常时

要点说明

- 本功能块是给轴设定转矩目标值，转矩目标值在转矩模式才会生效，故使用本功能块要先保证轴已切换为转矩模式。
- 可通过 SMC_SetControllerMode 功能块来切换轴控制模式。
- 该功能块是给对象字典中的 6071h 赋值，要求 RXPDO 中包含 6071h。

MC_MoveSuperImposed（FB）

叠加位置、速度运动功能。
在轴原本的控制功能块的基础上进行位置和速度叠加，不会打断或者影响原本控制轴运动的功能块。

名称	Mc_MoveSuperImposed（叠加相对运动）		
支持的模式	CSP		
图形表现		ST 表现	
		MC_MoveSuperImposed(Axis:=, Execute:= , Abort:= , Distance:= , VelocityDiff:= , Acceleration:= , Deceleration:= , Jerk:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);	

变量

（1） 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

（2） 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块
Abort	中断	BOOL	TRUE、FALSE	FALSE	输入一个上升时中断
Distance	目标相对位置	LREAL	负数、正数、“0”	0	指定相对坐标的目标位置，单位为【指令单位】
VelocityDiff	叠加速度	LREAL	正数	0	指定叠加速度，单位为【指令单位/S】
Acceleration	加速度	LREAL	正数或“0”	0	加速度，单位为【指令单位/s^2】
Deceleration	减速度	LREAL	正数或“0”	0	减速度，单位为【指令单位/s^2】
Jerk	加加速度	LREAL	正数或“0”	0	指定加加速度，单位为【指令单位/S^3】

（3） 输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	TRUE：功能块执行完成

Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
CommandAborted	功能块执行中断	BOOL	TRUE、FALSE	TRUE:功能块被中止
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码

（4） 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
Done	✧ 定位完成时	✧ Execute 为 TRUE 时，与 Execute 的 FALSE 同时 ✧ Execute 为 FALSE 时，1 个周期后
Busy	✧ Execute 的上升沿	✧ Done 为 TRUE 时 ✧ Error 为 TRUE 时 ✧ CommandAborted 变为 TRUE 时
CommandAborted	✧ 由其他功能块进行运动功能块多重启动（中断），本功能块中止时 ✧ 因发生异常，本功能块中止时 ✧ 正在发生异常时，启动了本功能块 ✧ 正在执行 MC_Stop 功能块时，启动了本功能块	✧ Execute 为 TRUE 时，与 Execute 的 FALSE 同时 ✧ Execute 为 FALSE 时，1 个周期后
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时

要点说明

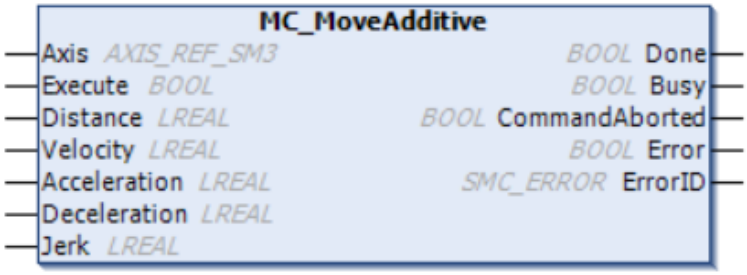
- 本功能块为叠加位置和速度功能块，Distance 为叠加在执行中的功能块基础上的距离，VelocityDiff 为叠加在执行中的功能块基础上的速度。
- Execute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数。在上一次叠加还没完成前提下，新的叠加数据载入功能块后，最终轴位置为定位功能块的目标位置+最后一次叠加的叠加距离。
- 如果单独执行本功能块，相当于 MC_MoveRelative。
- 本功能块在使用时，必须注意功能块摆放顺序，本功能块必须摆放在原本控制轴的运动控制功能块之后。如果被叠加的轴同时作为也其他轴的主轴，那么在调用到同步运动控制功能块之前，必须保证主轴的 MC_MoveSuperImposed 已被扫描执行过。
- 与 MC_MoveAdditive 的区别：
  ① 本功能块不会覆盖原功能块的速度等参数，只是在原功能块运动的基础上，叠加一个相对定位功能。MC_MoveAdditive 会打断原功能块，在原功能块的基础上加上设定的位置，然后按照 MoveAdditive 引脚中设定的速度进行定位运动。
  ② 本功能块多次重复触发，叠加位置不会累积。MC_MoveAdditive 多次触发叠加位置会累积，如每次叠加 100，假如若干同步周期内，多次重复触发 MC_MoveAdditive，那么最终叠加的位置为：100*N（N=重复触发次数）。
  ③ MC_MoveAdditive 功能块可以摆放在原本控制轴的运动控制功能块前面。

MC_MoveAdditive（FB）

叠加位置运动功能。

在轴原本的控制功能块的基础上进行位置叠加，并覆盖原速度加速度等参数。

名称	MC_MoveAdditive（叠加绝对运动功）		
支持的模式	CSP		
图形表现		ST 表现	




```
MC_MoveAdditive(
Axis:=,
Execute:= ,
Distance:= ,
Velocity:= ,
Acceleration:= ,
Deceleration:= ,
Jerk:= ,
Done=> ,
Busy=> ,
CommandAborted=> ,
Error=> ,
ErrorID=> );
```

变量

（1） 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

（2） 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块
Distance	目标绝对位置	LREAL	负数、正数、“0”	0	指定相对坐标的目标位置，单位为【指令单位】
Velocity	叠加速度	LREAL	正数	0	指定叠加速度，单位为【指令单位/S】
Acceleration	加速度	LREAL	正数或“0”	0	加速度，单位为【指令单位/s^2】
Deceleration	减速度	LREAL	正数或“0”	0	减速度，单位为【指令单位/s^2】
Jerk	加加速度	LREAL	正数或“0”	0	指定加加速度，单位为【指令单位/S^3】

（3） 输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	TRUE：功能块执行完成
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中

CommandAborted	功能块执行中断	BOOL	TRUE、FALSE	TRUE:功能块被中止
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代

（4） 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
Done	✧ 定位完成时	✧ Execute 为 TRUE 时，与 Execute 的 FALSE 同时 ✧ Execute 为 FALSE 时，1 个周期后
Busy	✧ Execute 的上升沿	✧ Done 为 TRUE 时 ✧ Error 为 TRUE 时 ✧ CommandAborted 变为 TRUE 时
CommandAborted	✧ 由其他功能块进行运动功能块多重启动（中断），本功能块中止时 ✧ 因发生异常，本功能块中止时 ✧ 正在发生异常时，启动了本功能块 ✧ 正在执行 MC_Stop 功能块时，启动了本功能块	✧ Execute 为 TRUE 时，与 Execute 的 FALSE 同时 ✧ Execute 为 FALSE 时，1 个周期后
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时

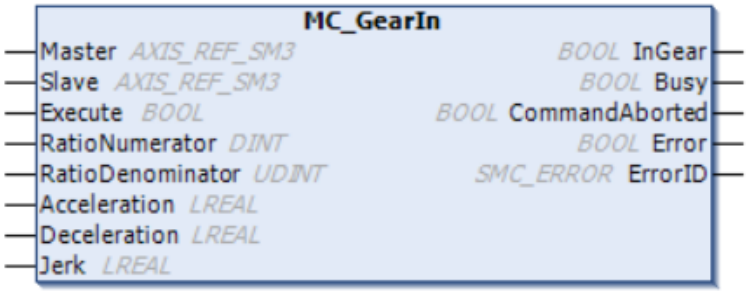
要点说明

- 本功能块为叠加位置，Distance 为叠加在原位置功能块基础上的相对距离，Velocity 为本功能块执行后，轴的运行速度。
- MC_MoveAdditive 在叠加时会中断其他可以被中断的功能块。
- Execute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数并在上次触发叠加的位置基础上叠加新的位置。
- 如果单独执行本功能块，相当于 MC_MoveRelative。
- 与 MC_MoveSuperImposed 的区别：参见 MC_MoveSuperImposed。

MC_GearIn（FB）

电子齿轮耦合功能。
控制从轴跟随主轴按照设定的速度比同步运行。

名称	MC_GearIn（电子齿轮耦合）		
支持的模式	CSP	CSV	
图形表现		ST 表现	




```
MC_GearIn(
Master:= ,
Slave:= ,
Execute:= ,
RatioNumerator:= ,
RatioDenominator:= ,
Acceleration:= ,
Deceleration:= ,
Jerk:= ,
InGear=> ,
Busy=> ,
CommandAborted=> ,
Error=> ,
ErrorID=> );
```

变量

（1） 输入输出变量

输入输出变量	名称	数据类型	内容
Master	主轴	AXIS_REF_SM3	指定主轴
Slave	从轴	AXIS_REF_SM3	指定从轴

（2） 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块
RatioNumerator	齿轮比分子	DINT	负数、正数	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”	0	加速度，单位为【指令单位/s^2】
Deceleration	减速度	LREAL	正数或“0”	0	减速度，单位为【指令单位/s^2】
Jerk	加加速度	LREAL	正数或“0”	0	指定加加速度，单位为【指令单位/S^3】

（3） 输出变量

输出变量	名称	数据类型	有效范围	内容
InGear	达到齿轮比	BOOL	TRUE、FALSE	TRUE:从轴跟随主轴同步完成
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
CommandAborted	功能块执行中断	BOOL	TRUE、FALSE	TRUE:功能块被中止
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码

（4） 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
InGear	✧ 从轴跟主轴同步完成	✧ 执行 MC_GearOut 或者其他运动控制功能打断 MC_GearIn ✧ Execute 为 FALSE 时，1 个周期后
Busy	✧ Execute 的上升沿	✧ Error 为 TRUE 时 ✧ CommandAborted 变为 TRUE 时
CommandAborted	✧ 由其他功能块进行运动功能块多重启动（中断），本功能块中止时 ✧ 因发生异常，本功能块中止时 ✧ 正在发生异常时，启动了本功能块 ✧ 正在执行 MC_Stop 功能块时，启动了本功能块	✧ Execute 为 TRUE 时，与 Execute 的 FALSE 同时 ✧ Execute 为 FALSE 时，1 个周期后
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时

要点说明

- 本功能块为电子齿轮耦合功能，RatioNumerator 和 RatioDenominator 分别为电子齿轮比的分子和分母，速度同步关系可参考如下公式：

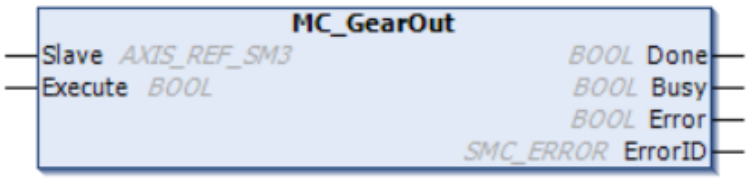
$$从轴速度 = \frac{RatioNumerator}{RatioDenominator} * 主轴速度$$

- Execute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数，重新执行功能块。
- 功能块执行时从轴处于 Synchronized Motion 状态。
- 此功能块只针对从轴，主轴并不会因为有从轴与其耦合而产生任何影响。
- 功能块的 Acceleration 与 Deceleration 参数只在功能块刚执行从轴追随主轴速度之时生效，一旦从轴与主轴速度进入同步，InGear 引脚为 TRUE，功能块将不做加减速度计算，直接将主轴速度按照齿轮比换算后给予从轴。
- 功能块执行中修改 RatioNumerator 和 RatioDenominator 等参数不会生效。

MC_GearOut（FB）

电子齿轮解除耦合功能。

此功能将从轴脱离与主轴的同步状态。

名称	Mc_GearOut(电子齿轮解耦)		
支持的模式	CSP	CSV	
图形表现		ST 表现	
		MC_GearOut(Slave:=, Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);	

变量

（1） 输入输出变量

输入输出变量	名称	数据类型	内容
Slave	从轴	AXIS_REF_SM3	指定从轴

（2） 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块

（3） 输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	TRUE：功能块执行完成
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码

（4） 输出变量的转化时序

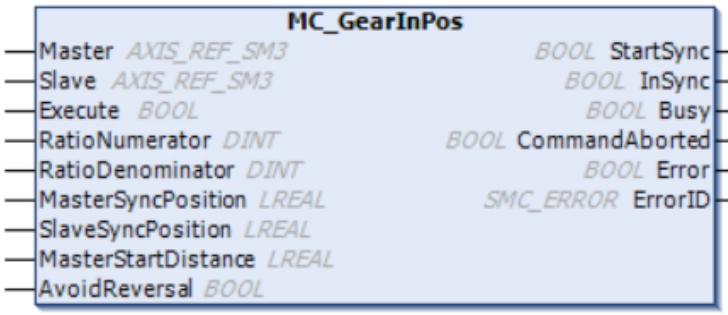
变量	变为 TRUE 时	变为 FALSE 时
Done	✧ 解耦完成时	✧ Execute 为 TRUE 时，与 Execute 的 FALSE 同时 ✧ Execute 为 FALSE 时，1 个周期后
Busy	✧ Execute 的上升沿	✧ Done 为 TRUE 时 ✧ Error 为 TRUE 时 ✧ CommandAborted 变为 TRUE 时
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时

要点说明

- Execute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数，重新执行功能块。
- 齿轮解耦功能块，功能是从轴和主轴分离，对 MC_GearIn、MC_GearInPos 都有效。
- 本功能块为电子齿轮耦合解除功能， 将从轴于与主轴的同步状态脱离，不在与主轴进行速度同步。
- 功能块执行完成后，从轴会处于 Continuous Motion 状态，从轴会以解除耦合时的速度持续运行，直到被其他运动控制功能块接管。因此建议如要解除耦合后停止从轴运行，可以在 MC_GearOut 执行成功后，调用 MC_Halt 来停止从轴运行，或者直接使用 MC_Halt 打断齿轮耦合。

MC_GearInPos（FB）

通过设置指定的主轴位置、从轴位置和主轴开始同步距离来完成切入电子齿轮动作。

名称	MC_GearInPos（指定位置耦合电子齿轮）		
支持的模式	CSP	CSV	
图形表现		ST 表现	
		MC_GearInPos(Master:=, Slave:=, Execute:= , RatioNumerator:= , RatioDenominator:= , MasterSyncPosition:= , SlaveSyncPosition:= , MasterStartDistance:= , AvoidReversal:= , StartSync=> , InSync=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);	

变量

（1） 输入输出变量

输入输出变量	名称	数据类型	内容
Master	主轴	AXIS_REF_SM3	指定主轴
Slave	从轴	AXIS_REF_SM3	指定从轴

（2） 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块
RatioNumerator	齿轮比分子	DINT	负数、正数	1	齿轮比分子
RatioDenominator	齿轮比分母	UDINT	正数	1	齿轮比分母
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			在这个距离上从轴进入同步，如果 MasterStartDistance 为 0，从轴立即启动
AvoidReversal	禁止反转	BOOL	TRUE、FALSE	FALSE	FALSE：如果从轴物理位置超前，会进行反转 TRUE：只在模态轴下使用。如果反转不能被避免，那么轴将错误停止动作

（3） 输出变量

输出变量	名称	数据类型	有效范围	内容
StartSync	开始耦合处理	BOOL	TRUE、FALSE	电子齿轮耦合开始时，变成 TRUE
InSync	耦合中	BOOL	TRUE、FALSE	电子齿轮耦合完成时，变成 TRUE
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
CommandAborted	功能块执行中断	BOOL	TRUE、FALSE	TRUE:功能块被中止
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR	0	发生异常时，输出错误代码

（4） 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
StartSync	✧ 电子齿轮开始耦合	✧ Error 为 TRUE 时 ✧ CommandAborted 变为 TRUE 时
InSync	✧ 电子齿轮完成耦合时	✧ Error 为 TRUE 时 ✧ CommandAborted 变为 TRUE 时
Busy	✧ Execute 的上升沿	✧ Error 为 TRUE 时 ✧ CommandAborted 变为 TRUE 时
CommandAborted	✧ 由其他功能块进行运动功能块多重启动（中断），本功能块中止时 ✧ 因发生异常，本功能块中止时 ✧ 正在发生异常时，启动了本功能块 ✧ 正在执行 MC_Stop 功能块时，启动了本功能块	✧ Execute 为 TRUE 时，与 Execute 的 FALSE 同时 ✧ Execute 为 FALSE 时，1 个周期后
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时

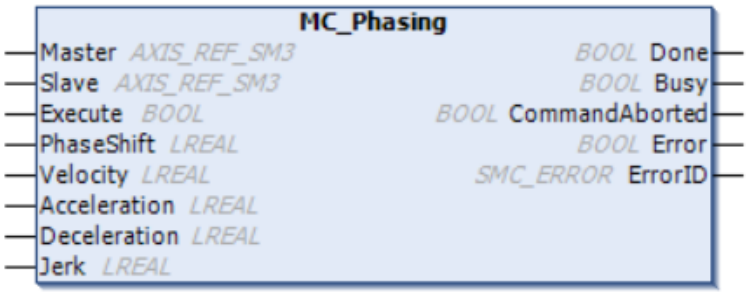
要点说明

- Execute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数，重新执行功能块。功能块开始动作后，从轴以主轴的速度乘以齿轮比所得到的速度为目标速度，进行加减速度。
- 从轴进入同步的位置等于 MasterSyncPosition-MasterStartDistance。

MC_Phasing（FB）

修改主轴与从轴之间的相位偏移

名称	MC_Phasing（主从轴相位偏移）		
支持的模式	CSP		
图形表现		ST 表现	




```
MC_Phasing(
Master:=,
Slave:=,
Execute:= ,
PhaseShift:= ,
Velocity:= ,
Acceleration:= ,
Deceleration:= ,
Jerk:= ,
Done=> ,
Busy=> ,
CommandAborted=> ,
Error=> ,
ErrorID=> );
```

变量

（1） 输入输出变量

输入输出变量	名称	数据类型	内容
Master	主轴	AXIS_REF_SM3	指定主轴
Slave	从轴	AXIS_REF_SM3	映射到从轴

（2） 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块
PhaseShift	相位偏移量	LREAL		0	主从轴之间的偏移量
Velocity	目标速度	LREAL	正数	0	指定目标速度。单位为【指令单位/S】
Acceleration	加速度	LREAL	正数或“0”	0	加速度，单位为【指令单位/s^2】
Deceleration	减速度	LREAL	正数或“0”	0	减速度，单位为【指令单位/s^2】
Jerk	加加速度	LREAL	正数或“0”	0	指定加加速度。单位为【指令单位/S^3】

（3） 输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	TRUE：功能块执行完成
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
CommandAborted	功能块执行中断	BOOL	TRUE、FALSE	TRUE:功能块被中止
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR		发生异常时，输出错误代码

（4） 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
Done	✧ 完成时	✧ Execute 为 TRUE 时，与 Execute 的 FALSE 同时 ✧ Execute 为 FALSE 时，1 个周期后
Busy	✧ Execute 的上升沿	✧ Done 为 TRUE 时 ✧ Error 为 TRUE 时 ✧ CommandAborted 变为 TRUE 时
CommandAborted	✧ 由其他功能块进行运动功能块多重启动（中断），本功能块中止时 ✧ 因发生异常，本功能块中止时 ✧ 正在发生异常时，启动了本功能块 正在执行 MC_Stop 功能块时，启动了本功能块	✧ Execute 为 TRUE 时，与 Execute 的 FALSE 同时 ✧ Execute 为 FALSE 时，1 个周期后
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时

要点说明

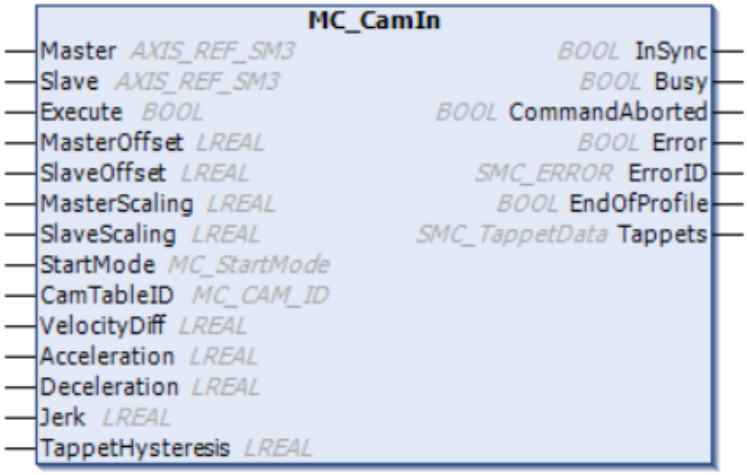
- 本功能块调整从轴相对于主轴的相位偏移，从轴会根据 PhaseShift 设定的相位偏移以及当前实际的相位偏移，依据 Velocity,Acceleration,Deceleration 等参数进行调整。
- PhaseShift 设置的是主轴与从轴的相位差，如果 PhaseShift 为正数，代表主轴相位超前从轴相位。
- 相位调整不会打断原本的同步运动，相位调整完成后，如果没有让从轴解除耦合状态，主从轴之间会始终保持这个相位差。
- 功能块执行时从轴处于 Synchronized Motion 状态，此功能块只针对从轴，主轴并不会因为有从轴与其耦合而产生任何影响。
- Execute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数，重新执行功能块，如果旧的相位偏差没有到位，则会以新相位偏差为准。

MC_CamIn(FB)

电子凸轮耦合功能。

根据凸轮表指定的位置关系以及主轴位置，控制从轴运动。

名称	MC_CamIn(主从轴耦合)		
支持的模式	CSP		
图形表现		ST 表现	




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

变量

（1） 输入输出变量

输入输出变量	名称	数据类型	内容
Master	主轴	AXIS_REF_SM3	映射到主轴
Slave	从轴	AXIS_REF_SM3	映射到从轴

（2） 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块
MasterOffset	主轴位置偏移	LREAL	负数、正数、“0”	0	以指定的偏移值移动主轴的相位
SlaveOffset	从轴位置偏移	LREAL	负数、正数、“0”	0	以指定的偏移值移动从轴的相位
MasterScaling	主轴放大比率	LREAL	>0.0	1	指定主轴比例
SlaveScaling	从轴放大比率	LREAL	>0.0	1	指定从轴比例
StartMode	从轴触发模式	MC_StarMode		absolute	参考 MC_StarMode
CamTableID	生效的 Cam_ID	MC_CAM_ID			定义 cam 表格的使用，与 MC_CamTableSelect 的输出点 CamTableID 配合使用
VelocityDiff	叠加速度	LREAL	正数	0	指定叠加速度，单位为【指令单位/S】
Acceleration	加速度	LREAL	正数或“0”	0	加速度，单位为【指令单位/s^2】
Deceleration	减速度	LREAL	正数或“0”	0	减速度，单位为【指令单位/s^2】
Jerk	加加速度	LREAL	正数或“0”	0	指定加加速度，单位为【指令单位/S^3】
TappetHysteresis	挺杆滞后	LREAL			挺杆的阻尼系数

（3） 输出变量

输出变量	名称	数据类型	有效范围	内容
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR		发生异常时，输出错误代码
EndOfProfile	曲线完成	BOOL	TRUE、FALSE	TRUE：凸轮执行结束，如果 Mc_CamTableSelect 的 Periodic 参数选择 TRUE（周期），此引脚会在凸轮主轴越过凸轮表结束点的那一个周期为 TRUE 如果 Mc_CamTableSelect 的 Periodic 参数选择 FALSE（非周期），则凸轮运行完成后会保持为 TRUE
Tappets	挺杆	SMC_TAPPETDATA		关联一个凸轮挺杆，可被 MC_GetTappetValue 读出

（4） 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
Busy	✧ Execute 的上升沿	✧ Done 为 TRUE 时 ✧ Error 为 TRUE 时 ✧ CommandAborted 变为 TRUE 时
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时

要点说明

- Execute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数，重新执行功能块。
- 该功能块是耦合主从轴在凸轮表的位置关系，在使用本功能块前，需要先绘制凸轮曲线并先执行 MC_CamTableSelect 选定凸轮表，然后再执行 MC_CamIn 进行凸轮耦合。
- 如要更换凸轮曲线关系，需要在合适的时机重新执行 MC_CamTableSelect 功能块选择新的凸轮表，然后重新执行 MC_CamIn。

- 凸轮耦合后，主轴与从轴有一个凸轮关系中的坐标系位置 X(对应凸轮表中的主轴位置)和 Y(凸轮表中主轴位置对应的从轴位置)，对应凸轮表中的位置。这两个位置由实际主轴从轴位置经过一系列计算变换得来，变化关系可以简写如下：

$$X = MasterScaling * MasterPos + MasterOffset;$$

$$Y = CAM(X);$$

$$SlavePos = SlaveScaling * CAM(X) + SlaveOffset;$$

MasterPos：主轴实际位置 、 SlavePos：从轴目标位置 、 CAM(X)：根据 X 的值与凸轮关系曲线，计算出 Y 的值。

StartMode 支持设置：Absolute 以及 Relative 两种模式，此模式设定主要针对从轴。在 MC_CamTableSelect 中也有两个对应的引脚来制定主轴以及从轴的绝对与相对，具体耦合关系如下：

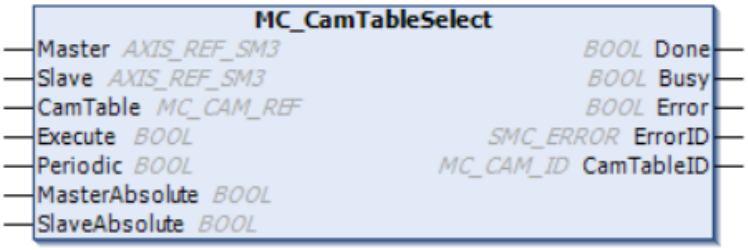
MC_CamTableSelect.MasterAbsolute	主轴最终坐标关系
TRUE	绝对坐标
FALSE	相对坐标

MC_CamTableSelect.SlaveAbsolute	MC_CamIn.StartMode	**从轴最终坐标位置**
TRUE	absolute	绝对坐标
TRUE	relative	相对坐标
FALSE	absolute	相对坐标
FALSE	relative	相对坐标

MC_CamTableSelect（FB）

选定电子凸轮表。

名称	MC_CamTableSelect（选择凸轮表）		
支持的模式	CSP		
图形表现		ST 表现	




```
MC_CamTableSelect(
Master:=,
Slave:=,
CamTable:= ,
Execute:= ,
Periodic:= ,
MasterAbsolute:= ,
SlaveAbsolute:= ,
Done=> ,
Busy=> ,
Error=> ,
ErrorID=> ,
CamTableID=> );
```

变量

(1) 输入输出变量

输入输出变量	名称	数据类型	内容
Master	主轴	AXIS_REF_SM3	映射到主轴
Slave	从轴	AXIS_REF_SM3	映射到从轴
CamTable	选择凸轮表	MC_CAM_REF	映射为 cam 表格描述

(2) 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块
Periodic	周期模式	BOOL	TRUE、FALSE	TRUE	设定凸轮周期性 TRUE：周期 FALSE：不周期
MasterAbsolute	主轴绝对模式	BOOL	TRUE、FALSE	TRUE	主轴位置模式 TRUE：绝对位置 FALSE：相对位置
SlaveAbsolute	从轴绝对模式	BOOL	TRUE、FALSE	TRUE	从轴位置模式 TRUE：绝对位置 FALSE：相对位置

(3) 输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	TRUE：功能块执行完成
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR		发生异常时，输出错误代码
CamTableID	生效的 Cam_ID	MC_CAM_ID		选择生效的 Cam_ID，与 MC_CamIn 功能块中的 CamTableID 配合使用

(4) 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
Done	✧ 完成时	✧ Execute 为 TRUE 时，与 Execute 的 FALSE 同时 ✧ Execute 为 FALSE 时，1 个周期后
Busy	✧ Execute 的上升沿	✧ Done 为 TRUE 时 ✧ Error 为 TRUE 时
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时

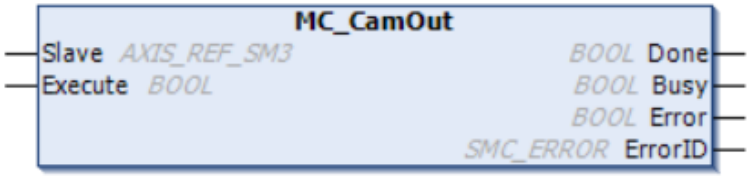
要点说明

- 该功能块用于关联主轴、从轴、电子凸轮表三者的关系，并设定凸轮运行是否为周期模式，主轴、从轴坐标系为绝对还是相对等，若需要更换凸轮表，或改变主从轴，还需要再触发执行该功能块一次。
- MasterAbsolute、SlaveAbsolute、Periodic 参数具体作用参考 MC_CamIn 功能块。
- 本功能块不会控制从轴或者主轴运动。
- Execute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数，重新执行功能块。

MC_CamOut（FB）

电子凸轮解除耦合功能。

此功能将从轴脱离与主轴的同步状态。

名称	MC_CamOut(解除耦合)		
支持的模式	CSP	CSV	
图形表现		ST 表现	
		MC_CamOut(Slave:=, Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);	

变量

（1） 输入输出变量

输入输出变量	名称	数据类型	内容
Slave	从轴	AXIS_REF_SM3	映射到从轴

（2） 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块

（3） 输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	TRUE：功能块执行完成
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR		发生异常时，输出错误代码

（4） 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
Done	✧ 解耦完成时	✧ Execute 为 TRUE 时，与 Execute 的 FALSE 同时 ✧ Execute 为 FALSE 时，1 个周期后
Busy	✧ Execute 的上升沿	✧ Done 为 TRUE 时 ✧ Error 为 TRUE 时
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时

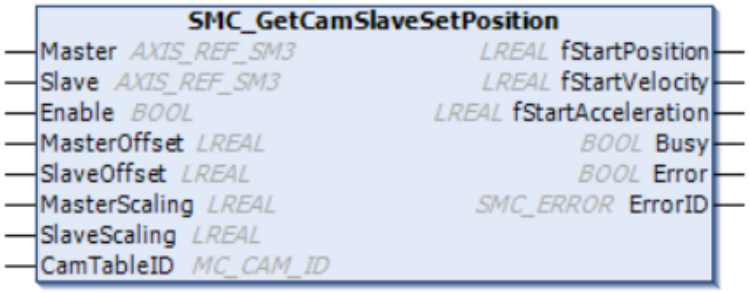
要点说明

- Execute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数，重新执行功能块。
- 齿轮解耦功能块，功能是从轴和主轴分离。
- 本功能块为电子凸轮耦合解除功能， 将从轴于与主轴的同步状态脱离，不再与主轴进行位置同步。
- 功能块执行完成后，从轴会处于 Continuous Motion 状态，从轴会以解除耦合时的速度持续运行，直到被其他运动控制功能块接管。因此建议如要解除耦合后停止从轴运行，可以在 MC_GearOut 执行成功后，调用 MC_Halt 来停止从轴运行，或者直接使用 MC_Halt 打断凸轮耦合。

SMC_GetCamSlaveSetPosition（FB）

根据主轴位置获取凸轮表关系中从轴的位置、速度、加速度信息。

名称	SMC_GetCamSlaveSetPosition(读取凸轮从轴位置)		
支持的模式	CSP		
图形表现		ST 表现	




```
SMC_GetCamSlaveSetPosition(
Master:=,
Slave:=,
Enable:= ,
MasterOffset:= ,
SlaveOffset:= ,
MasterScaling:= ,
SlaveScaling:= ,
CamTableID:=,
fStartPosition=> ,
fStartVelocity=> ,
fStartAcceleration=>
Busy=> ,
Error=> ,
ErrorID=> );
```

变量

（1） 输入输出变量

输入输出变量	名称	数据类型	内容
Master	主轴	AXIS_REF_SMC3	映射到主轴
Slave	从轴	AXIS_REF_SMC3	映射到从轴

（2） 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
bEnable	有效	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块 FALSE：关闭功能块
MasterOffset	主轴位置偏移	LREAL	负数、正数、“0”	0	以指定的偏移值移动主轴的相位
SlaveOffset	从轴位置偏移	LREAL	负数、正数、“0”	0	以指定的偏移值移动从轴的相位
MasterScaling	主轴放大比率	LREAL	>0.0	1	指定主轴比例
SlaveScaling	从轴放大比率	LREAL	>0.0	1	指定从轴比例
CamTableID	生效的 Cam_ID	MC_CAM_ID			定义 cam 表格的使用，与 MC_CamTableSelect 的输出点 CamTableID 配合使用

（3） 输出变量

输出变量	名称	数据类型	有效范围	内容
fStartPosition	位置	LREAL		根据凸轮表格和当前主轴位置定义的从轴位置
fStartVelocity	速度	LREAL		根据凸轮表格和当前主轴位置定义的从轴速度
fStartAcceleration	加速度	LREAL		根据凸轮表格和当前主轴位置定义的从轴加速度
Busy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
Error	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
ErrorID	错误代码	SMC_ERROR		发生异常时，输出错误代码

（4） 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
Busy	✧ Enable 为 TRUE 时	✧ Done 为 TRUE 时 ✧ Error 为 TRUE 时
Error	✧ 功能块执行中产生故障时	✧ 解除了异常时

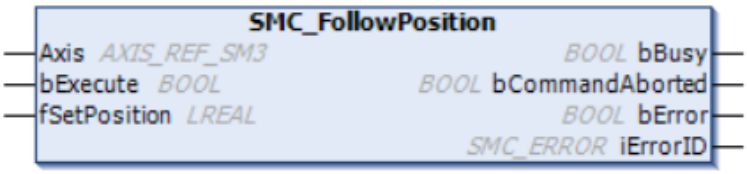
要点说明

- 本功能块不会控制从轴或主轴运动，可以在主从轴静止时使用，只需要关联好主从轴以及电子凸轮。
- 本功能块根据凸轮表关系，通过输入 MasterOffset 和 SlaveOffset，计算出从轴对应的位置速度加速度等参数。
- 在 Enable 为 TRUE 时，其他输入参数写入一次，输出参数就会更新一次读取到的参数。如果读取参数的时候主轴速度为 0，那么输出参数 fStartVelocity、fStartAcceleration 显示为 0，如果主轴速度不为 0，则输出参数 fStartVelocity、fStartAcceleration 根据 cam 表规划显示相应数值。

SMC_FollowPosition（FB）

位置跟随功能。

该功能块将目标位置直接作为轴下个周期的目标位置，而不执行任何检查，执行上升沿信号后，每个任务周期都会给轴位置功能块而不管轴的状态。

名称	SMC_FollowPosition（位置跟随）		
支持的模式	CSP		
图形表现		ST 表现	
		SMC_FollowPosition(Axis:=, bExecute:= , fSetPosition:= , bBusy=> , bCommandAborted=> , bError=> , iErrorID=>);	

变量

（1） 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

（2） 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
bExecute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块
fSetPosition	设定目标位置	LREAL		0	设定轴的目标位置

（3） 输出变量

输出变量	名称	数据类型	有效范围	内容
bBusy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
bCommandAborted	功能块执行中断	BOOL	TRUE、FALSE	TRUE:功能块被中止
bError	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
iErrorID	错误代码	SMC_ERROR	-	发生异常时，输出错误代码

（4） 输出变量的转化时序

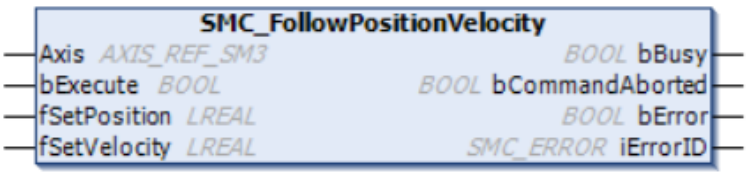
变量	变为 TRUE 时	变为 FALSE 时
bBusy	✧ bExecute 的上升沿	✧ bDone 为 TRUE 时 ✧ bError 为 TRUE 时 ✧ bCommandAborted 变为 TRUE 时
bCommandAborted	✧ 由其他功能块进行运动功能块多重启动（中断），本功能块中止时 ✧ 因发生异常，本功能块中止时 ✧ 正在发生异常时，启动了本功能块 ✧ 正在执行 MC_Stop 功能块时，启动了本功能块	✧ bExecute 为 TRUE 时，与 bExecute 的 FALSE 同时 ✧ bExecute 为 FALSE 时，1 个周期后
bError	✧ 功能块执行中产生故障时	✧ 解除了异常时

要点说明

- bExecute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数，重新执行功能块。
- 本功能块用于直接指定轴每个同步周期要运动到的目标位置，不会做速度、加速度等检查，故使用时要注意处理好 fSetPosition 的值，实现轴运行速度的线性变化。
- 功能块执行时，轴处于 Synchronized Motion。
- 如要停止本功能块的执行，可以通过 MC_Halt、MC_GearOut、MC_CamOut 等功能打断。

SMC_FollowPositionVelocity（FB）

该功能块跟 SMC_FollowPosition 使用跟功能都一样，但是增加了速度设定，速度设定要满足位置设定变化，速度设定=相隔任务周期位置设定差值对时间的一次导数。

名称	SMC_FollowPositionVelocity(位置速度跟随)		
支持的模式	CSP	CSV	
图形表现		ST 表现	
		SMC_FollowPositionVelocity(Axis:=, bExecute:= , fSetPosition:= , fSetVelocity:= , bBusy=> , bCommandAborted=> , bError=> , iErrorID=>);	

变量

(1) 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

(2) 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
bExecute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块
fSetPosition	设定目标位置	LREAL	-	0	设定轴的目标位置
fSetVelocity	设定目标速度	LREAL	-	0	设定轴的目标速度

(3) 输出变量

输出变量	名称	数据类型	有效范围	内容
bBusy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
bCommandAborted	功能块执行中断	BOOL	TRUE、FALSE	TRUE:功能块被中止
bError	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
iErrorID	错误代码	SMC_ERROR	-	发生异常时，输出错误代码

(4) 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
bBusy	✧ bExecute 的上升沿	✧ bDone 为 TRUE 时 ✧ bError 为 TRUE 时 ✧ bCommandAborted 变为 TRUE 时
bCommandAborted	✧ 由其他功能块进行运动功能块多重启动（中断），本功能块中止时 ✧ 因发生异常，本功能块中止时 ✧ 正在发生异常时，启动了本功能块 ✧ 正在执行 MC_Stop 功能块时，启动了本功能块	✧ bExecute 为 TRUE 时，与 bExecute 的 FALSE 同时 ✧ bExecute 为 FALSE 时，1 个周期后
bError	✧ 功能块执行中产生故障时	✧ 解除了异常时

要点说明

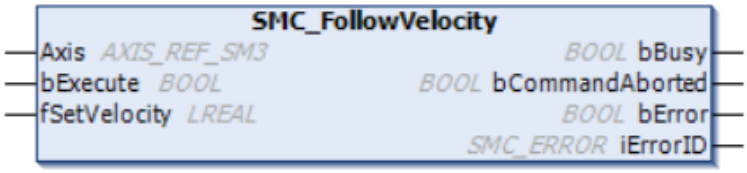
- bBusy 为 TRUE 时，轴状态为 Synchronized Motion,可以用 MC_CamOut 解除状态。
- 通过 bExecute 上升沿启动之后，会每个周期给轴发生位置、速度功能块。
- $fSetVelocity = \frac{\Delta fSetPoistion}{\Delta t}$ 轴设定目标速度要与设定目标位置变化一致。

  fSetPoistion 为本任务周期跟上个任务周期 fSetPoistion 差值，△t 为扫描时间。

SMC_FollowVelocity（FB）

速度跟随功能。

该功能块将目标速度直接作为轴的当前目标速度，而不执行任何检查，执行上升沿信号后，每个任务周期都会给轴速度功能块而不管轴的状态。

名称	SMC_FollowVelocity(速度跟随)		
支持的模式	CSP	CSV	
图形表现		ST 表现	
		SMC_FollowVelocity(Axis:=, bExecute:= , fSetVelocity:= , bBusy=> , bCommandAborted=> , bError=> , iErrorID=>);	

变量

(1) 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

(2) 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
bExecute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块
fSetVelocity	设定速度	LREAL		0	设定轴的速度

(3) 输出变量

输出变量	名称	数据类型	有效范围	内容
bBusy	功能块执行中	BOOL	TRUE、FALSE	收到功能块时变为 TRU
bCommandAborted	功能块执行中断	BOOL	TRUE、FALSE	TRUE:功能块被中止
bError	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
iErrorID	错误代码	SMC_ERROR		发生异常时，输出错误代码

(4) 输出变量的转化时序

变量	变为 TRUE 时	变为 FALSE 时
bBusy	✧ Execute 的上升沿	✧ Done 为 TRUE 时 ✧ Error 为 TRUE 时 ✧ CommandAborted 变为 TRUE 时
bCommandAborted	✧ 由其他功能块进行运动功能块多重启动（中断），本功能块中止时 ✧ 因发生异常，本功能块中止时 ✧ 正在发生异常时，启动了本功能块 正在执行 MC_Stop 功能块时，启动了本功能块	✧ Execute 为 TRUE 时，与 Execute 的 FALSE 同时 ✧ Execute 为 FALSE 时，1 个周期后
bError	✧ 功能块执行中产生故障时	✧ 解除了异常时

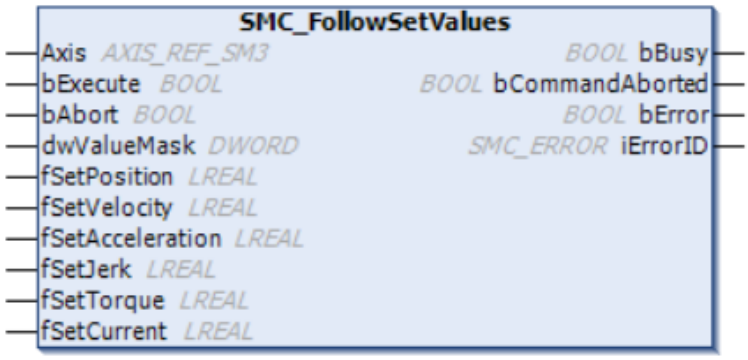
要点说明

- bExecute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数，重新执行功能块。
- 本功能块用于直接指定轴当前的目标速度，不会做速度、加速度等检查，故使用时要注意处理好 fSetVelocity 的值，通过编程实现 fSetVelocity 的线性变化。
- 功能块执行时，轴处于 Synchronized Motion。
- 如要停止本功能块的执行，可以通过 MC_Halt、MC_GearOut、MC_CamOut 等功能打断。

SMC_FollowSetValues（FB）

该功能块实现将设置位置、速度、加速度、转矩、电流写入轴中不执行任何检查。

名称	SMC_FollowSetValues（设定值跟随）		
支持的模式	CSP	CSV	CST
图形表现		ST 表现	




```
SMC_FollowSetValues(
Axis:=,
bExecute:= ,
bAbort:= ,
dwValueMask:= ,
fSetPosition:= ,
fSetVelocity:= ,
fSetAcceleration:= ,
fSetJerk:= ,
fSetTorque:= ,
fSetCurrent:= ,
bBusy=> ,
bCommandAborted=> ,
bError=> ,
iErrorID=> );
```

变量

(1) 输入输出变量

输入输出变量	名称	数据类型	内容
Axis	轴	AXIS_REF_SM3	指定轴，即 AXIS_REF_SM3 的一个实例

(2) 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
bExecute	启动	BOOL	TRUE、FALSE	FALSE	TRUE：启用功能块
bAbort	中断	BOOL	TRUE、FALSE	FALSE	输入一个上升时中断
dwValueMask	选择控制参数	DWORD		0	Bit0：TRUE 时激活 fSetPosition，FALSE 忽略 Bit1：TRUE 时激活 fSetVelocity，FALSE 忽略 Bit 2：TRUE 时激活 fSetAcceleration，FALSE 忽略 Bit 3：TRUE 时激活 fSetJerk，FALSE 忽略 Bit 4：TRUE 时激活 fSetTorque，FALSE 忽略 Bit 5：TRUE 时激活 fSetCurrent，FALSE 忽略
fSetPosition	设定目标位置	LREAL	负数、正数、“0”	0	指定绝对坐标的目标位置，单位为【指令单位】
fSetVelocity	设定目标速度	LREAL	正数或“0”	0	设定轴的速度
fSetAcceleration	设定目标加速度	LREAL	正数或“0”	0	指定加速度，单位为【指令单位/S^2】

fSetJerk	设定目标加加速度	LREAL	正数或“0”	0	指定加加速度，单位为【指令单位/S^3】
fTSetTorque	设定目标转矩	LREAL	正数或“0”	0	单位【Nm】
fTSetCurrent	设定目标电流	LREAL	正数或“0”	0	单位【A】

(3) 输出变量

输出变量	名称	数据类型	有效范围	内容
bBusy	功能块执行中	BOOL	TRUE、FALSE	TRUE:功能块运行中
bCommandAborted	功能块执行中断	BOOL	TRUE、FALSE	TRUE:功能块被中止
bError	错误	BOOL	TRUE、FALSE	TRUE:功能块产生异常，已停止执行
iErrorID	错误代码	SMC_ERROR		发生异常时，输出错误代码

(4) 输出变量的转化时序

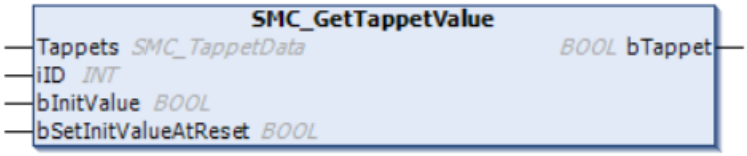
变量	变为 TRUE 时	变为 FALSE 时
bBusy	✧ bExecute 的上升沿	✧ Done 为 TRUE 时 ✧ Error 为 TRUE 时 ✧ CommandAborted 变为 TRUE 时
bCommandAborted	✧ 由其他功能块进行运动功能块多重启动（中断），本功能块中止时 ✧ 因发生异常，本功能块中止时 ✧ 正在发生异常时，启动了本功能块 ✧ 正在执行 MC_Stop 功能块时，启动了本功能块	✧ Execute 为 TRUE 时，与 Execute 的 FALSE 同时 ✧ Execute 为 FALSE 时，1 个周期后
bError	✧ 功能块执行中产生故障时	✧ 解除了异常时

要点说明

- bBusy 为 TRUE 时，轴状态为 Synchronized Motion,可以用 MC_CamOut 解除状态。
- 通过 bExecute 上升沿启动之后，会每个周期给轴发生位置、速度功能块。
- 通过 dwValueMask 值来选择控制参数，当 dwValueMask 为 1（2#1）时，则本功能块与 SMC_FollowPosition 功能一样，当 dwValueMask 为 2（2#10）时，则本功能块与 SMC_FollowVelocity 功能一样，当 dwValueMask 为 3（2#11）时，则本功能块与 SMC_FollowPositionVelocity 功能一样。

SMC_GetTappetValue（FB）

与 MC_CamIn 配合使用，读取当前挺杆输出值。

名称	SMC_GetTappetValue(读取挺杆输出值)		
支持的模式	CSP		
图形表现		ST 表现	
		SMC_GetTappetValue(Tappets:=, iID:= , bInitValue:= , bSetInitValueAtReset:= , bTappet=>);	

变量

(1) 输入输出变量

输入输出变量	名称	数据类型	内容
Tappets	挺杆	SMC_TappetData	映射到一个挺杆

(2) 输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
iID	挺杆编号	INT		0	要评估的挺杆的组 ID
bInitValue	挺杆初始值	BOOL	TRUE、FALSE	FALSE	
bSetInitValueAtRest	设置初始值	BOOL	TRUE、FALSE	FALSE	TRUE：MC_CamIn 功能块重启时挺杆的输出值将会被设置为 bInitValue FALSE：挺杆的值在 MC_CamIn 功能块重启的时候将会被保持

(3) 输出变量

输出变量	名称	数据类型	有效范围	内容
bTappet	挺杆输出值	BOOL	TRUE、FALSE	挺杆输出值

要点说明

- Execute 的上升沿为启动功能块，功能块执行中可以重新触发上升沿，每次上升沿都会重新载入功能块的输入参数，重新执行功能块。
- 该功能块需要与 MC_CamIn 功能块配合使用。
- 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，所以在同一个凸轮表中只使用一种。

参考

MC_Direction（ENUM）

指定移动方向，需要注意的是并不是所有的移动方向都可以应用于所有的功能块和轴类型(模态/线性)。当“轴类型与限位”中选择“模态轴”，在 Direction 中指定开始定位的方向。“轴类型与限位”中选择“线性轴”时，将忽略 Direction。

名称	序号	内容
fastest	3	选择最快的方向移动（仅模态）
current	2	按当前方向移动
positive	1	正方向移动
shortest	0	根据最短路径选择方向（仅模态）
negative	-1	负方向移动

SMC_Controller_Mode（ENUM）

轴的控制模式。

名称	初始值	内容
SMC_cocontrol	0	不支持
SMC_torque	1	转矩模式
SMC_velocity	2	速度模式
SMC_position	3	位置模式
SMC_current	4	电流模式

SMC_Homing_Mode(ENUM)

回原模式选择。

名称	类型	初始值	描述
FAST_BSLOW_S_STOP	SMC_Homing_Mode	0	轴按照设定方向以快速度走向原点参考开关（bRefenrenceSwitch），碰到原点参考开关后以慢速度反转离开，离开后先执行 MC_SetPosition，然后执行 MC_Stop
FAST_BSLOW_STOP_S	SMC_Homing_Mode	1	轴按照设定方向以快速度走向原点参考开关，碰到原点参考开关后以慢速度反转离开，离开后先执行 MC_ Stop，然后执行 MC_SetPosition
FAST_BSLOW_I_S_STOP	SMC_Homing_Mode	2	轴按照设定方向以快速度走向原点参考开关，碰到原点参考开关后以慢速度反转离开，等待脉冲信号，离开后先执行 MC_SetPosition，然后执行 MC_Stop
FAST_SLOW_S_STOP	SMC_Homing_Mode	4	轴按照设定方向以快速度走向原点参考开关，碰到原点参考开关后以慢速度离开，离开后先执行 MC_SetPosition，然后执行 MC_Stop
FAST_SLOW_STOP_S	SMC_Homing_Mode	5	轴按照设定方向以快速度走向原点参考开关，碰到原点参考开关后以慢速度离开，离开后先执行 MC_ Stop，然后执行 MC_SetPosition
FAST_SLOW_I_S_STOP	SMC_Homing_Mode	6	轴按照设定方向以快速度走向原点参考开关，碰到原点参考开关后以慢速度反转离开，等待脉冲信号，离开后先执行 MC_ Stop，然后执行 MC_SetPosition

SMC_CommunicationState(ENUM)

SMC_COMSTATE_NOT_STARTED	通讯未启动
SMC_COMSTATE_VARIABLE_INITIALIZATION	通讯变量初始化
SMC_COMSTATE_BASE_COM_INITIALIZATION	通讯基本端口初始化
SMC_COMSTATE_DRIVE_INITIALIZATION	通讯驱动初始化
SMC_COMSTATE_DRIVE_WAITING_FOR_SYNC	通讯驱动等待同步
SMC_COMSTATE_INITIALIZATION_DONE	通讯初始化完成
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	通讯状态未知

MC_StartMode(ENUM)

名称	序号	内容
Absolute	0	绝对位置
Relative	1	相对位置
Ramp_in	2	斜坡切入
Ramp_in_pos	3	正向斜坡切入
Ramp_in_neg	4	反向斜坡切入

TRIGGER_REF（STRUCT）

名称	数据类型	初始值	内容
iTriggerNumeber	INT	-1	输入不同数值可选择不同探针功能
bFastLatching	BOOL	TRUE	TRUE：使用 16#60B8 定义的探针功能 FALSE：使用 PLC 探针功能，会受扫描周期影响不建议使用
bInput	BOOL		仅在使用 PLC 探针功能时生效
bActive	BOOL		锁存功能执行标志

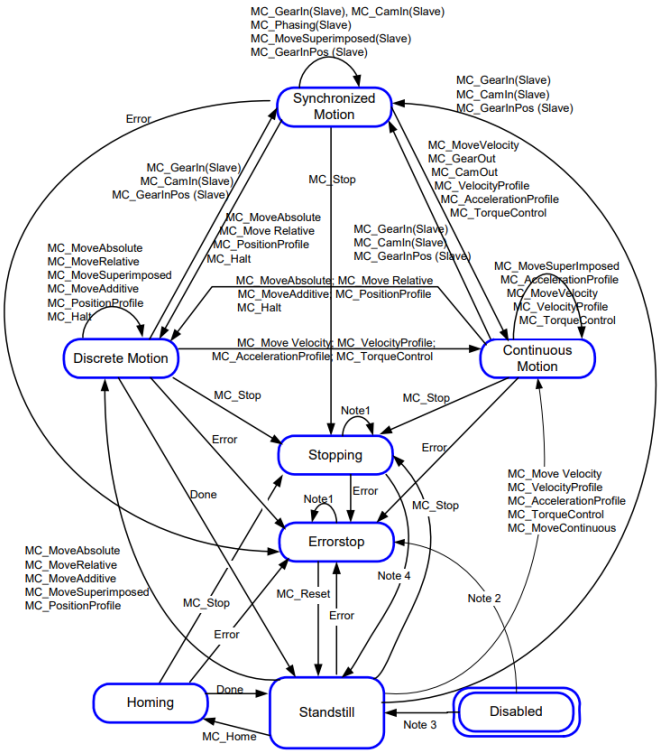
探针要点说明参考伺服手册。

SMC3_PersistPositionDiag (ENUM)

SMC3_PPD_RESTORING_OK	位置恢复成功
SMC3_PPD_AXIS_PROP_CHANGED	至少一个轴的属性 （比例，位置周期，运动类型 （模态/直线））发生改变，所以不能恢复位置
SMC3_PPD_DATA_STORED_DURING_WRITING	SMC3_PPD_DATA_STORED_DURING_WRITING： 数据显示当功能块从轴结构体中复制数据 PersistentData 被保存。 可能原因：1、非持久性变量 2、功能块控制无效

轴状态转移说明（ENUM）

根据 PLC OPEN 规定，轴状态可做如下转换：



Note1	在 Errorstop 或 Stopping 状态时，所有的功能块都可以被调用，但是他们不会并不会执行，除了 MC_Reset 可以把状态转换为 Standstill。如果错误发生在轴状态为 Stopping 时，轴状态会转换成 Errorstop
Note2	MC_Power 的 Enable 引脚为 TRUE 且轴发生错误
Note3	MC_Power 的 Enable 引脚为 TRUE 且轴没有发生错误
Note4	MC_Stop 的 Done 引脚为 TRUE 且 MC_Stop 的 Execute 引脚为 FALSE

➢ ※注：codesys 通过枚举体 SMC_AXIS_STATE 列出了 PLC OPEN 所规定的轴状态，具体如下：

0: power_off（在 PLC OPEN 的定义中为 Disabled）

1: errorstop　2: stopping　3: standstill　4: discrete_motion　5: continuous_motion　6: synchronized_motion　7: homing

下图是以重命名为 axis1 的轴做轴状态控制的样例程序：

```
IF axis2.nAxisState=1 THEN  //nAxisState为1时表示Errorstop
    MC_MoveVelocity1.execute:=FALSE;
    wstate:="errorstop";
    MC_Stop1.Execute:=TRUE;
END_IF
```

SMC_Error(ENUM)

枚举提 SMC_ERROR 包含 SoftMotion 功能块可以返回的所有错误号。SMC_ErrorString 生成错误字符串输出。

Name	Initial	Comment
SMC_NO_ERROR	0	No error
SMC_DI_GENERAL_COMMUNICATION_ERROR	1	Communication error. For example, sercos ring has broken
SMC_DI_AXIS_ERROR	2	Axis error
SMC_DI_FIELDBUS_LOST_SYNCRONICITY	3	Fieldbus has lost synchronism
SMC_DI_SWLIMITS_EXCEEDED	10	Position outside of permissible range of SWLimit
SMC_DI_HWLIMITS_EXCEEDED	11	Hardware end switch is active
SMC_DI_LINEAR_AXIS_OUTOFRANGE	12	This error occurs if a linear axis has more than 2^15 32-bit overflows of the position in increments
SMC_DI_HALT_OR_QUICKSTOP_NOT_SUPPORTED	13	Drive status Halt or Quickstop is not supported
SMC_DI_VOLTAGE_DISABLED	14	Drive has no power
SMC_DI_IRREGULAR_ACTPOSITION	15	This error is no longer used
SMC_DI_POSITIONLAGERROR	16	Position lag error. Difference between set and current position exceeds the given limit
SMC_DI_HOMING_ERROR	17	Homing error reported by axis
SMC_DI_LICENSING_ERROR	18	A problem with the license occurred
SMC_REGULATOR_OR_START_NOT_SET	20	Controller enable not done or brake applied
SMC_WRONG_CONTROLLER_MODE	21	Axis in wrong controller mode
SMC_INVALID_ACTION_FOR_LOGICAL	25	Invalid action at logical axis
SMC_FB_WASNT_CALLED_DURING_MOTION	30	Motion creating module has not been called again before end of the motion
SMC_AXIS_IS_NO_AXIS_REF	31	Type of given AXIS_REF variable is not AXIS_REF
SMC_AXIS_REF_CHANGED_DURING_OPERATION	32	AXIS_REF variable has been exchanged while the module was active
SMC_FB_ACTIVE_AXIS_DISABLED	33	Axis disabled while being moved. MC_Power.bRegulatorOn
SMC_AXIS_NOT_READY_FOR_MOTION	34	Axis in its current state cannot execute a motion command, because the axis doesn’t signal currently that it follows the target values
SMC_AXIS_ERROR_DURING_MOTION	35	
SMC_VD_MAX_VELOCITY_EXCEEDED	40	Maximum velocity fMaxVelocity exceeded
SMC_VD_MAX_ACCELERATION_EXCEEDED	41	Maximum acceleration fMaxAcceleration exceeded
SMC_VD_MAX_DECELERATION_EXCEEDED	42	Maximum deceleration fMaxDeceleration exceeded
SMC_3SH_INVALID_VELACC_VALUES	50	Invalid velocity or acceleration values
SMC_3SH_MODE_NEEDS_HWLIMIT	51	Mode requests for safety reasons use of end switches
SMC_FRC_NO_FREE_HANDLE	60	No free handle has been sent to open file
SMC_SCM_NOT_SUPPORTED	70	Mode not supported

Name	Initial	Comment
SMC_SCM_AXIS_IN_WRONG_STATE	71	In current mode, controller mode cannot be changed
SMC_SCM_INTERRUPTED	72	
SMC_ST_WRONG_CONTROLLER_MODE	75	Axis not in correct controller mode
SMC_RAG_ERROR_DURING_STARTUP	80	Error at startup of the axis group
SMC_RAG_ERROR_AXIS_NOT_INITIALIZED	81	The axis is not yet in the required state.
SMC_PP_WRONG_AXIS_TYPE	85	
SMC_PP_NUMBER_OF_ABSOLUTE_BITS_INVALID	86	
SMC_CGR_ZERO_VALUES	90	Invalid values
SMC_CGR_DRIVE_POWERED	91	Gearing parameters must not be changed as long as the drive is under control
SMC_CGR_INVALID_POSPERIOD	92	Invalid position period (<= 0)
SMC_CGR_POSPERIOD_NOT_INTEGRAL	93	
SMC_P_FTASKCYCLE_EMPTY	110	Axis contain no information on cycle time (fTaskCycle = 0)
SMC_R_NO_ERROR_TO_RESET	120	Axis without error
SMC_R_DRIVE_DOESNT_ANSWER	121	Axis does not perform error-reset
SMC_R_ERROR_NOT_RESETTABLE	122	Error could not be reset
SMC_R_DRIVE_DOESNT_ANSWER_IN_TIME	123	Communication with the axis did not work
SMC_RP_PARAM_UNKNOWN	130	Parameter number unknown
SMC_RP_REQUESTING_ERROR	131	Error during transmission to the drives. See error number in function block instance ReadDriveParameter
SMC_RP_DRIVE_PARAMETER_NOT_MAPPED	132	No assignment for drive parameters available
SMC_RP_PARAM_CONVERSION_ERROR	133	Conversion of the value to / from the drive parameters failed Unknown SoftMotionparameters
SMC_WP_PARAM_INVALID	140	Parameter number unknown or writing not allowed
SMC_WP_SENDING_ERROR	141	See error number in function block instance WriteDriveParameter
SMC_WP_DRIVE_PARAMETER_NOT_MAPPED	142	No assignment for drive parameters available
SMC_WP_PARAM_CONVERSION_ERROR	143	Conversion of the value to / from the drive parameters failed Unknown SoftMotionparameters
SMC_H_AXIS_WASNT_STANDSTILL	170	Axis has not been in standstill state
SMC_H_AXIS_DIDNT_START_HOMING	171	Error at start of homing action
SMC_H_AXIS_DIDNT_ANSWER	172	Error at start of homing action
SMC_H_ERROR_WHEN_STOPPING	173	Error at stop after homing. Deceleration may not be set
SMC_H_AXIS_IN_ERRORSTOP	174	Drive is in errorstop status. Homing cannot be executed
SMC_MS_UNKNOWN_STOPPING_ERROR	180	Unknown error at stop
SMC_MS_INVALID_ACCDEC_VALUES	181	Invalid velocity or acceleration values
SMC_MS_DIRECTION_NOT_APPLICABLE	182	Direction = shortest not applicable

Name	Initial	Comment
SMC_MS_AXIS_IN_ERRORSTOP	183	Drive is in errorstop status. Stop cannot be executed
SMC_BLOCKING_MC_STOP_WASNT_CALLED	184	Instance of MC_Stop blocking the axis by Execute = TRUE has not been called yet. MC_Stop (Execute = FALSE) has to be called
SMC_MS_AXIS_ALREADY_STOPPING	185	Cannot interrupt an ongoing MC_Stop
SMC_UNKNOWN_TASK_INTERVAL	200	
SMC_MA_INVALID_VELACC_VALUES	201	Invalid velocity or acceleration values
SMC_MA_INVALID_DIRECTION	202	Direction error
SMC_MR_INVALID_VELACC_VALUES	226	Invalid velocity or acceleration values
SMC_MR_INVALID_DIRECTION	227	Direction error
SMC_MAD_INVALID_VELACC_VALUES	251	Invalid velocity or acceleration values
SMC_MAD_INVALID_DIRECTION	252	Direction error
SMC_MSI_INVALID_VELACC_VALUES	276	Invalid velocity or acceleration values
SMC_MSI_INVALID_DIRECTION	277	Direction error
SMC_LOGICAL_NO_REAL_AXIS	300	no longer used; only for compatibility
SMC_MV_INVALID_ACCDEC_VALUES	301	Invalid velocity or acceleration values
SMC_MV_DIRECTION_NOT_APPLICABLE	302	Direction = shortest/fastest not applicable
SMC_PP_ARRAYSIZE	325	erroneous array size
SMC_PP_STEP0MS	326	Step time = t#0s
SMC_VP_ARRAYSIZE	350	erroneous array size
SMC_VP_STEP0MS	351	Step time = t#0s
SMC_AP_ARRAYSIZE	375	erroneous array size
SMC_AP_STEP0MS	376	Step time = t#0s
SMC_TP_TRIGGEROCCUPIED	400	Trigger already active
SMC_TP_COULDNT_SET_WINDOW	401	DriveInterface does not support the window function
SMC_TP_COMM_ERROR	402	Communication error
SMC_AT_TRIGGERNOTOCCUPIED	410	Trigger already de-allocated
SMC_MCR_INVALID_VELACC_VALUES	426	Invalid velocity or acceleration values
SMC_MCR_INVALID_DIRECTION	427	Invalid direction
SMC_MCA_INVALID_VELACC_VALUES	451	Invalid velocity or acceleration values
SMC_MCA_INVALID_DIRECTION	452	Invalid direction
SMC_MCA_DIRECTION_NOT_APPLICABLE	453	Direction = fastest not applicable
SMC_SDL_INVALID_AXIS_STATE	475	
SMC_SDL_INVALID_VELACC_VALUES	476	
SMC_CR_NO_TAPPETS_IN_CAM	600	cam does not contain any tappets
SMC_CR_TOO_MANY_TAPPETS	601	Tappet group ID exceeds MAX_NUM_TAPPETS

Name	Initial	Comment
SMC_CR_MORE_THAN_32_ACCESSES	602	More than 32 accesses on one CAM_REF
SMC_CI_NO_CAM_SELECTED	625	No cam selected
SMC_CI_MASTER_OUT_OF_SCALE	626	Master axis out of valid range
SMC_CI_RAMPIN_NEEDS_VELACC_VALUES	627	Velocity and acceleration values must be specified for ramp_in function
SMC_CI_SCALING_INCORRECT	628	Scaling variables fEditor/TableMasterMin/Max are not correct
SMC_CI_TOO_MANY_TAPPETS_PER_CYCLE	629	Too many tappets became active during one cycle
SMC_CB_NOT_IMPLEMENTED	640	Function block for the given cam format is not implemented
SMC_GI_RATIO_DENOM	675	RatioDenominator = 0
SMC_GI_INVALID_ACC	676	Acceleration invalid
SMC_GI_INVALID_DEC	677	Deceleration invalid
SMC_GI_MASTER_REGULATOR_CHANGED	678	Status Enable/Disable of the master has changed without permission
SMC_GI_INVALID_JERK	679	
SMC_PH_INVALID_VELACCDEC	725	Velocity and acceleration/deceleration values invalid
SMC_PH_ROTARYAXIS_PERIOD0	726	Rotation axis with fPositionPeriod = 0
SMC_NO_CAM_REF_TYPE	750	Type of given cam is not MC_CAM_REF
SMC_CAM_TABLE_DOES_NOT_COVER_MASTER_SCALE	751	Master area, xStart and xEnd, from CamTable is not covered by curve data
SMC_CAM_TABLE_EMPTY_MASTER_RANGE	752	
SMC_CAM_TABLE_INVALID_MASTER_MINMAX	753	
SMC_CAM_TABLE_INVALID_SLAVE_MINMAX	754	
SMC_GIP_MASTER_DIRECTION_CHANGE	775	During coupling of slave axis, master axis has changed direction of rotation
SMC_GIP_SLAVE_REVERSAL_CANNOT_BE_AVOIDED	776	
SMC_GIP_AVOID_REVERSAL_FOR_FINITE_AXIS	777	
SMC_BC_BL_TOO_BIG	800	Gear backlash fBacklash too large (> position periode/2)
SMC_QPROF_DIVERGES	825	
SMC_QPROF_INVALID_PARAMETER	826	
SMC_QPROF_NO_RESULT	827	
SMC_QPROF_INVALID_NEW_LBD	828	
SMC_QPROF_BAD_NEGOTIATION	829	
SMC_QPROF_INVALID_INTERVAL	830	
SMC_QPROF_NOT_ENOUGH_PHASES	831	
SMC_SRT_NOT_STANDSTILL_OR_POWEROFF	850	Action only permitted in standstill or power_off
SMC_SRT_INVALID_RAMPTYPE	851	Invalid ramp type
SMC_SMT_NOT_STANDSTILL_OR_POWEROFF	852	Action only permitted in standstill or power_off
SMC_SMT_INVALID_MOVEMENTTYPE_OR_POSITIONPERIOD	853	Invalid motion type or position period
SMC_SMT_AXIS_NOT_VIRTUAL	854	Function block only applicable to virtual axis

Name	Initial	Comment
SMC_NO_LICENSE	1000	Target is not licensed for CNC
SMC_INT_VEL_ZERO	1001	
SMC_INT_NO_STOP_AT_END	1002	Path cannot be processed because set velocity=0
SMC_INT_DATA_UNDERRUN	1003	Warning: GEOINFO-List processed in DataIn but end of list not reached Reason: • EndOfList of the queue in DataIn not be set • SMC_Interpolator faster than path generating function blocks
SMC_INT_VEL_NONZERO_AT_STOP	1004	Velocity at Stop > 0
SMC_INT_TOO_MANY_RECURSIONS	1005	Too much SMC_Interpolator recursions. SoftMotion-Error
SMC_INT_NO_CHECKVELOCITIES	1006	SMC_CheckVelocities is not the last processed function block, that accesses to the OutQueue-data by poqDataIn
SMC_INT_PATH_EXCEEDED	1007	Internal or numeric error
SMC_INT_VEL_ACC_DEC_ZERO	1008	Velocity and acceleration / deceleration is null or to low
SMC_INT_DWIPOTIME_ZERO	1009	FB called with dwIpoTime=0
SMC_INT_JERK_NONPOSITIVE	1010	Jerk invalid because jerk must be positive
SMC_INT_QPROF_DIVERGES	1011	Internal error. Calculation algorithm incorrectly
SMC_INT_INVLALID_VELOCITY_MODE	1012	Invalid velocity mode
SMC_INT_TOO_MANY_AXES_INTERPOLATED	1013	
SMC_INT_DEGENERATE_SEGMENT	1014	The segment is numerically degenerate: it is very short and at the end of the queue. It should be ignored.
SMC_INT2DIR_BUFFER_TOO_SMALL	1050	Warning: poqDataIn of OutQueue is created too small. Adherence of stops not guaranteed
SMC_INT2DIR_PATH_FITS_NOT_IN_QUEUE	1051	Path does not go completely in queue
SMC_XINT_INVALID_DIRECTION	1070	
SMC_XINT_NOINTERSECTION	1071	
SMC_WAR_INT_OUTQUEUE_TOO_SMALL	1080	Warning: poqDataIn of OutQueue is created too small. Adherence of stops not guaranteed
SMC_WAR_END_VELOCITIES_INCORRECT	1081	Warning: Final velocities inconsistent
SMC_CV_ACC_DEC_VEL_NONPOSITIVE	1100	Velocity and acceleration/deceleration values non-positive
SMC_CA_INVALID_ACCDEC_VALUES	1120	Values of fGapVelocity / fGapAcceleration / fGapDeceleration non-positive
SMC_DEC_ACC_TOO_LITTLE	1200	Acceleration value impermissible
SMC_DEC_RET_TOO_LITTLE	1201	Deceleration value impermissible
SMC_DEC_OUTQUEUE_RAN_EMPTY	1202	Data underrun. Queue has been read and is empty
SMC_DEC_JUMP_TO_UNKNOWN_LINE	1203	Jump to line cannot be executed because line number is unknown
SMC_DEC_INVALID_SYNTAX	1204	Syntax invalid
SMC_DEC_3DMODE_OBJECT_NOT_SUPPORTED	1205	Objects are not supported in 3D mode
SMC_DEC_NEGATIVE_PERIOD	1206	Negative value as a period of additional axes invalid

Name	Initial	Comment
SMC_DEC_DIMENSIONS_EXCLUSIVE_AU	1207	Not both axes A and U are interpolated. PA and PU are mutually exclusive
SMC_DEC_DIMENSIONS_EXCLUSIVE_BV	1208	Not both axes B and V are interpolated. PB and PV are mutually exclusive
SMC_DEC_DIMENSIONS_EXCLUSIVE_CW	1209	Not both axes C and W are interpolated. PC and PW are mutually exclusive
SMC_GCV_BUFFER_TOO_SMALL	1300	Buffer too small
SMC_GCV_BUFFER_WRONG_TYPE	1301	Buffer elements have wrong type
SMC_GCV_UNKNOWN_IPO_LINE	1302	Current line of the Interpolator could not be found
SMC_NO_CNC_REF_TYPE	1500	Given CNC program is not of type SMC_CNC_REF
SMC_NO_OUTQUEUE_TYPE	1501	Given OutQueue is not of type SMC_OUTQUEUE
SMC_GEOINFO_BUFFER_MISALIGNED	1502	4-byte aligned buffer part is not used in pbyBuffer
SMC_3D_MODE_NOT_SUPPORTED	1600	Function block only works with 2D paths
SMC_SAA_SMOOTHAREA_TOO_LARGE	1700	Range for smoothing too large
SMC_SAA_SP_INVALID_INPUT	1701	Invalid input dSmoothingPart]0..1]
SMC_SA_QUEUE_NOT_IN_BUFFER	1800	SMC_SegmentAnalyzer detects that OutQueue buffer is full but not completed. The function block can only operate when OutQueue fits the buffer completely.
SMC_SA_QUEUE_CHANGED_DURING_OP	1801	OutQueue buffer changed while the function block is operating on them
SMC_OS_INVALID_PARAMETER	1820	
SMC_BSSP_IPO_NOT_ACTIVE	1830	Position cannot be saved. Interpolator is inactive
SMC_BS_SAVEDPOS_NOT_REACHED	1831	Saved position has not been found so far. It is probably on a different path
SMC_BS_NO_POS_STORED	1832	Structure passed in ePos contains no saved position. SMC_BlockSearchSavePos is not executed or in a wrong way.
SMC_INVALID_FEATURE_FLAG	1900	
SMC_SMB_HFUN_NOT_SUPPORTED	1901	
SMC_SMB_ONLY_3DMODE	1902	
SMC_SMB_ERROR_COMPUTING_SPLINE	1903	
SMC_SMM_INVALID_PARAM_NUMBER	1910	
SMC_RNCF_FILE_DOESNT_EXIST	2000	File does not exist
SMC_RNCF_NO_BUFFER	2001	No buffer allocated
SMC_RNCF_BUFFER_TOO_SMALL	2002	Buffer too small
SMC_RNCF_DATA_UNDERRUN	2003	Data underrun. Buffer has been read, is empty
SMC_RNCF_VAR_COULDNT_BE_REPLACED	2004	Placeholder variable could not be replaced
SMC_RNCF_NOT_VARLIST	2005	Input pvl does not point to a SMC_VARLIST object
SMC_RNCF_NO_STRINGBUFFER	2006	Input pStringBuffer is not used or does not point to a variable of type SMC_StringBuffer
SMC_RNCF_STRINGBUFFER_OVERRUN	2007	In the CNC program more different strings are used as reserved in pStringBuffer
SMC_RNCQ_FILE_DOESNT_EXIST	2050	File could not be opened
SMC_RNCQ_NO_BUFFER	2051	No buffer defined

Name	Initial	Comment
SMC_RNCQ_BUFFER_TOO_SMALL	2052	Buffer too small
SMC_RNCQ_UNEXPECTED_EOF	2053	Unexpected end of file
SMC_ADL_FILE_CANNOT_BE_OPENED	2100	File could not be opened
SMC_ADL_BUFFER_OVERRUN	2101	Buffer overrun. WriteToFile must be called more frequently
SMC_RCAM_FILE_DOESNT_EXIST	2200	File could not be opened
SMC_RCAM_TOO_MUCH_DATA	2201	Saved cam too big
SMC_RCAM_WRONG_COMPILE_TYPE	2202	Wrong compilation mode
SMC_RCAM_WRONG_VERSION	2203	File has wrong version
SMC_RCAM_UNEXPECTED_EOF	2204	Unexpected end of file
SMC_WDPF_CHANNEL_OCCUPIED	3001	SMC_WDPF_TIMEOUT_PREPARING_LIST
SMC_WDPF_CANNOT_CREATE_FILE	3002	File could not be created
SMC_WDPF_ERROR_WHEN_READING_PARAMS	3003	Error at reading the parameters
SMC_WDPF_TIMEOUT_PREPARING_LIST	3004	Timeout during preparing the parameter list
SMC_ENC_DENOM_ZERO	5000	Nominator of the conversion factor dwRatioTechUnitsDenom of the Encoder reference is 0
SMC_ENC_AXISUSEDBYOTHERFB	5001	Other module trying to process motion on the Encoder axis
SMC_ENC_FILTER_DEPTH_INVALID	5002	Filter depth is invalid
SMC_PCCQ_POINTBUFFERTOOSMALL	10000	Buffer pBuffer too small
SMC_PCCQ_INPUTBUFFERFULLBUTNOTFINALIZED	10001	The function block must be applied to a path that fits completely in the buffer
SMC_AXIS_GROUP_WRONG_STATE	11000	The axis group is in the wrong state for the requested operation
SMC_AXIS_GROUP_TOO_MANY_AXES	11001	More than the maximum allowed number of axes has been added to an axis group
SMC_AXIS_GROUP_INVALID_DYNLIMITS	11002	The dynamic limits of a single axis are invalid (fSWMaxVelocity/Acceleration/Deceleration/Jerk)
SMC_AXIS_GROUP_INVALID_COORD_SYSTEM	11004	The given coordinate system is invalid for the requested operation
SMC_AXIS_GROUP_SINGLE_AXIS_ERROR	11005	An axis of the axis group is in state error
SMC_MOVE_DIRECT_INVALID_BUFFER_MODE	11006	The given buffer mode is not supported
SMC_MOVE_PTP_INVALID_DYNAMIC_FACTOR	11007	The dynamic factor is not in the range]0 .. 1]
SMC_MOVE_PTP_INVALID_DYNAMICS	11008	The dynamic limits for the PTP movement are invalid
SMC_AXIS_GROUP_AXIS_NOT_PART_OF_GROUP	11009	The given axis is not part of the axis group
SMC_AXIS_GROUP_NOT_SUPPORTED	11010	The requested operation is not supported
SMC_AXIS_GROUP_KINEMATICS_NOT_SET	11011	No kinematic configuration has been configured
SMC_AXIS_GROUP_WRONG_NUMBER_OF_AXES	11012	The number of axes is not equal to the number of axes needed by the kinematic transformation
SMC_AXIS_GROUP_INTERRUPTED_BY_SINGLE_AXIS	11013	A coordinated movement has been interrupted by a single axis movement.
SMC_AXIS_GROUP_FOLLOW_SETVALUES	11014	An error occurred while following the computed set values

Name	Initial	Comment
SMC_AXIS_GROUP_TOO_MANY_DEPENDENCIES	11015	An axis group cannot depend on more than one master axis group
SMC_AXIS_GROUP_MUTUAL_DEPENDENCY	11016	An axis group A may not depend on another axis group the depends on A
SMC_AXIS_GROUP_DEPENDENCY_IN_DIFFERENT_TASK	11017	A dependant axis group must be executed in the same task
SMC_AXIS_GROUP_AXIS_IN_DIFFERENT_TASK	11018	An axis belonging to an axis group must be executed in the same task
SMC_AXIS_GROUP_PCS_STILL_IN_USE	11019	A second activation of the function block occurred while the PCS was still used by buffered motion commands.
SMC_AXIS_GROUP_CMD_ABORTED_DUE_TO_ERROR	11020	An error ocurred in a previous motion command.
SMC_AXIS_GROUP_INVALID_PARAMETER	11021	A parameter of an administrative function block is invalid
SMC_AXIS_GROUP_UNSUPPORTED_RAMPTYPE	11022	One of the axes of the group uses an unsupported ramp type. Only trapez and quadratic are supported.
SMC_KERNEL_PTP_INTERNAL_ERROR	11100	Internal error in the kernel
SMC_KERNEL_PTP_INVALID_TASKCYCLETIME	11101	Task cycle time not positive
SMC_TRAFO_INVALID_PARAMETERS	12000	Invalid parameters
SMC_TRAFO_INVALID_CONSTELLATION	12001	The input constellation (typ. cartesian values) are incompatible with the configuration. Typically this means that the given position is outside of the working area of the kinematic transformation.

用我们的工作

创造美好的生活

浙江禾川科技股份有限公司
电话：0570-7117888

杭州研发中心
技术支持热线:400126969