HMC series controller

Motion Control Commands Manual

Thank you very much for purchasing our HMC series products.

This manual describes the use and maintenance of the HMC series controller, basic programming instruction, etc. Please read this manual carefully before installing, wiring, using, maintaining, and checking the product.

Please keep this manual in a safe place and deliver it to the end user.

Statement

The contents of this user manual are subject to change without prior notice.

If you find any suspicion, error, or omission in the content of this user manual, please contact us to change it.

If there are any error or missing pages in this user manual, we will replace them for you.

HMC series controller motion control commands manual

Publisher: Guangzhou Auctech Automation Technology Ltd

ADD: Hongshi Business Building, 11 Kehua Road, SCI-TECH Industry Park, Taihe Town, Baiyun District, Guangzhou, CHINA

Telephone: +86 20 84898493

Change Log

Revision	Change Information	Originator	Date
V1.0	New Release	mxh	2021-05
V1.1	Modify format	mxh	2021-08
V1.2	Organize the content and revise the format	lsw	2022-09
V1.3	Update HMC series controller related manual tables and notes	czm	2023-02
V1.4	Update templates and fonts throughout the manual	czm	2023-06
V1.5	Change company address and related information	czm	2023-09

HMC Series Controller Related Manuals

The following table shows the information, please select the manual according to your need

Serial number	Manual Name	Description
1	HMC Series Controller and IO Module Selection Manual	About the basic function type of controller products to understand the description
2	HMC Series Controller Software Getting Started Manual	Software acquisition, installation, getting started tutorial
3	HMC S3 Series Controller User Manual	Explanation on the basic use of S3 series controller, etc.
4	HMC S4 Series Controller User's Manual	Explanation on the basic use of S4 series controllers, etc.
5	HMC G300 Series Controller User Manual	About the basic use and functions of the G300 series controllers
6	HMC series controller programming basic Command manual	Understanding of the concept and function of basic controller programming Commands
7	HMC series controller motion control Command manual (this book)	Understanding of basic concepts and functions of motion control commands

*Note: All the above information can be found on the official website: http://www.auctech.com.cn.

Table of Contents

Section 1 Safety Precautions

■ Safety Commands

- Please read and follow these safety precautions when installing, operating, or maintaining the product.
- For personal and equipment safety, please follow all safety precautions described in the markings and manuals on the product when installing, operating, and maintaining the product.
- The "Caution", "Warning" and "Danger" items in the manual do not represent all safety precautions to be observed, but only in addition to all other safety precautions.
- This product should be used in an environment that meets design specifications, otherwise it may cause a malfunction due to failure to comply with the relevant safety precautions.
- The product quality warranty does not cover abnormal function or damage to parts caused by the regulations.
- We will not bear any legal responsibility for personal safety accidents and property damage caused by illegal operation of the product.

Security Level Definition	
⚠ Danger	Indicates a potentially hazardous situation which, if not avoided,could result in death or serious injury. Additionally, there maybe severe property damage.
⚠ Caution	If not used in accordance with the regulations, may cause fires, serious personal injury, or even death!
⚠ Warning	Failure to use in accordance with the regulations may result in moderate personal injury or minor injury, as well as the occurrence of equipment damage!

When products arrive and are stored	
⚠ Warning	● If the product and product accessories are damaged when opening the box, please do not install them and contact our company or your supplier immediately. ● Check carefully whether the arriving product and the ordered product model match, and whether the product and product accessories are included.
⚠ Caution	● Do not stack multiple of this product on top of each other as this may cause injury or malfunction. ● Do not store in places exposed to direct sunlight, places where the ambient temperature exceeds the temperature conditions for storage, places where the relative humidity exceeds the humidity condition for

	storage, places where there is a large temperature difference, places where there is high condensation, places near corrosive gases, places where there are flammable gases, places where there is a large amount of dust, dirt, salt or metal dust, places where water, oil or medicine drip, places where vibration or shock can affect the main body of product; otherwise it can lead to fire, Electric shock or machine damage. ● Do not hold the cable or motor shaft for weight holding, as this may result in injury or malfunction.

When designing the system	
⚠ Danger	● If the rated load of current is exceeded or the load is short-circuited for a long period of time resulting in over-current, the product may start smoking or catch fire. safety devices such as fuses, or circuit breakers should be set externally.
⚠ Warning	● Be sure to design safety circuits to ensure that the product system will still work safely if the external power supply is lost, or the product fails. ● For safe operation of the equipment, please design external protection circuits and safety mechanisms for output signals related to major accidents.
⚠ Caution	● Be sure to install emergency brake circuits, protection circuits, interlock circuits for forward and reverse operation, and position upper and lower limit interlock switches to prevent damage to the machine in the external circuit of the product. ● The product may shut down all outputs after detecting abnormalities in its own system; when part of the controller circuit fails, it may cause its output to be uncontrolled. To ensure normal operation, a suitable external control circuit needs to be designed. ● If the output unit such as relay or transistor of the product is damaged, the output will not be controlled to the ON or OFF state. ● The product is designed to be used in indoor, over voltage class II electrical environments, and its power system level should have lightning protection devices to ensure that lightning over voltage is not applied to the product's power input or signal input, control output and other ports to avoid damage to equipment.

When the product is installed	
⚠ Danger	● Only maintenance professionals with adequate electrical knowledge and training related to electrical equipment should install this product. ● For the product with open equipment, please install in the control cabinet with door lock (product cabinet shell protection > IP20), only operators with sufficient electrical knowledge and training related to electrical equipment can open the product cabinet.
⚠ Warning	● When disassembling the product, the external power supply used for the system must be completely disconnected before performing the operation. Failure to disconnect all power supplies may result in electric shock or product failure and malfunction. ● While dissembling the product, the power and the power indicator must be turned off for at least 5 minutes, before disassembling the driver. Otherwise, the residual voltage may cause electric shock. ● Do not use the product in the following places: places with dust, oil fumes, conductive dust, corrosive gases, combustible gases; places exposed to high temperature, condensation, wind, and rain; places with vibration and shock. Electric shock, fire, and misuse can also cause damage and deterioration of the product!
⚠ Caution	● Avoid metal shavings and wire tips falling into the ventilation holes of the product during installation, this may cause fire, malfunction, and misoperation. ● After installation, ensure that there is no foreign matter on the ventilation surfaces, otherwise it may lead to poor heat dissipation and cause fire, malfunction and misoperation. ● When installing, make a tight connection to the respective connector and lock the product connection hook firmly. If the products are not installed properly, it may lead to misoperation, malfunction and dislodgement.

When wiring products	
⚠ Danger	● Only maintenance professionals with adequate electrical knowledge and training related to electrical equipment should perform the wiring of this product.

⚠ Warning	● During wiring operations, the external supply power used by the system must be completely disconnected before operation. Failure to disconnect all of them may result in electric shock or equipment malfunction or misoperation. ● When powering up and running after the wiring operation, the terminal cover that comes with the product must be installed. Failure to install the terminal cover may result in electric shock. ● Check the type of interface to be connected before connecting the cable correctly. If the wrong interface is connected or the wiring is incorrect, it may cause the product or external equipment to malfunction. ● The cable terminals should be well insulated to ensure that the insulation distance between the cables is not reduced after the cables are installed to the terminal block. Otherwise, it will lead to electric shock or equipment damage. ● Avoid metal shavings and wire tips falling into the ventilation holes of the controller when wiring, which may cause fire, malfunction, and misoperation! ● The bolts on the terminal blocks should be tightened within the specified torque range. Untightened terminal bolts may result in short circuit, fire, or malfunction. Over-tightening the bolts may damage the bolts and the product, resulting in dislodgement, short circuit, fire, or false operation.
⚠ Caution	● The specification and installation method of the external wiring of the equipment should meet the requirements of local power distribution regulations. ● To ensure the safety of the equipment and the operator, the equipment needs to be reliably grounded using cables of sufficient wire size. ● For connections using connectors and external devices, press fit, crimp, or properly solder using the tool specified by the manufacturer. A poor connection may result in a short circuit, fire, or malfunction. ● If the product is labeled to prevent foreign objects from entering the product during wiring, such as the wiring head. Do not remove this label during wiring operations. Before starting system operation, be sure to remove the label to facilitate heat dissipation. ● Please do not bundle the control and communication cables with the main circuit or power supply cables, etc. The alignment should be more than 100mm apart, otherwise the noise may lead to misoperation. ● For applications with serious interference, please use shielded cables for input or output of high frequency signals to improve the system's

	anti-interference capability.

Before powering on the product	
Danger	● Before powering on, please make sure the product is well installed, wired firmly and the motor unit is allowed to restart. ● Before powering on, please confirm that the power supply meets the product requirements to avoid causing damage to the product or starting a fire. ● It is strictly forbidden to open the product cabinet door or product protective cover, touch any terminals of the product, disassemble any device or parts of the product in the energized state, otherwise there is a risk of electric shock. ● Make sure that no one is around the product, the motor, or the machinery before powering it on, as this may result in injury or death!
Warning	● After the wiring operation and parameter setting are completed, please conduct a test run of the machine to confirm that it can operate safely, otherwise it may lead to injury or equipment damage! ● Before powering on, please make sure that the rated voltage of the product is the same as the power supply voltage. If the power supply voltage is used incorrectly, there is a risk of fire!

When operating and maintaining	
Danger	● Only maintenance professionals with adequate electrical knowledge and training on electrical equipment can perform the operation and maintenance of the products. ● Do not touch the terminals when the power is on, as this may cause electric shock or malfunction. ● When the motor or equipment is running, please never touch its rotating parts, otherwise it may lead to serious personal safety accidents.
Warning	● When cleaning the product or retightening the bolts on the terminal block or the connector mounting bolts, the external supply power used by the system must be completely disconnected. Failure to do so may result in electric shock. ● When disassembling the product or connecting or removing the communication cable, the external supply power used by the system must be completely disconnected first. Failure to disconnect all of them may result in electric shock or false operation. ● While dissembling the product, the power and the power indicator must be

	turned off for at least 5 minutes, before disassembling the driver. Otherwise, the residual voltage may cause electric shock.
⚠ Caution	● For online modification, forced output, RUN, STOP, etc., you must read the user's manual and confirm its safety before performing the relevant operations. ● Be sure to disconnect the power before loading and unloading expansion cards, modules, and other components!

When the product is scrapped	
⚠ Caution	● Please dispose of them as industrial waste; when disposing of batteries, do so separately according to the ordinances established by each region to avoid property damage or human injury! ● End-of-life products should be treated and recycled in accordance with industrial waste treatment standards to avoid polluting the environment.

Section 2 Introduction to motion control commands

2. 1 Concept of motion control commands

During project implementation, motion control functions need to be implemented and motion control commands need to be used, while the motion control commands for MC function blocks are based on the technical specifications of PLCopen's motion control function blocks.

2. 1. 1 PLCopen specification

In addition to the specification recommendations for general logic control Commands, program structures, and keywords contained in various languages, the PLCopen specification also defined the technical specifications of the motion control function block MC, including the naming of MC function blocks, specific functions, definitions of input and output variables, and related timing logic, to ensure maximum interoperability of user programming techniques.

2. 1. 2 Types of motion control commands

The types of motion control commands contain three types: general commands, single-axis commands, and axis group commands.

Command Type	Outline
General Commands	Commands for general status, operation or monitoring of various data
Single axis commands	Single-axis motion control command, monitoring axis status command
Axis group commands	Axis group coordinated action control command, monitoring axis group status command

1) Start and status of motion control commands

The input variables for motion control commands are "Execute" and Enable, while the output variables for status include "Busy", "Done", "Command Aborted", "Error", etc.

2) Exception handling

When there is an exception in the motion control command, the output variable “Error” changes to TRUE and the exception code outputs to Error ID (error code).

3) Motion control commands can be written with exception handlers in 2 ways:

a. Exception handling as command

The output variable Error (error) or ErrorID (error code) of the command can be used to handle exceptions for each command where an exception occurs.

The following is an example of determining that "Hardware end switch is active" occurs for the command with the instance name "MC_Power". When "bNoAxisErr" becomes TRUE, the programmer executes exception handling. (As shown in Figure 1)

b. Exception handling by type

The exception status of the motion control system variables can be used to perform exception handling for each type of exception.

The following is an example of determining the "Slave communication exception" for the axis of "Axis". When "bConnectErr" changes to TRUE, exception processing is performed. (Figure 2)

PLC_PRG ×

```
PROGRAM PLC_PRG
VAR
    bPwr_Enable     : BOOL;
    bPwr_Status     : BOOL;
    bPwr_Busy       : BOOL;
    bPwr_Err        : BOOL;
    bNoAxisError    : BOOL;

    Pwr_ErrID       : SMC_ERROR;
    fbMC_Power      : MC_Power;
```

```
bPwr_Enable := TRUE;

fbMC_Power(
    Axis:= Axis,
    Enable:= bPwr_Enable,
    bRegulatorOn:= bPwr_Enable,
    bDriveStart:= bPwr_Enable,
    Status=> ,
    bRegulatorRealState=> ,
    bDriveStartRealState=> ,
    Busy=> bPwr_Busy,
    Error=> bPwr_Err,
    ErrorID=> Pwr_ErrID);

IF bPwr_Err AND Pwr_ErrID = SMC_ERROR.SMC_DI_HWLIMITS_EXCEEDED THEN
    bNoAxisError := TRUE;
END_IF
```

Figure 1

```
IF Axis.wCommunicationState <> 100 THEN
    bConnectError := TRUE;
ELSE
    bConnectError := FALSE;
END_IF
```

Figure 2

4) Change of input variables when executing motion control command (command restart takes effect)

When the input variable of the same command is changed during the execution, change Execute to TRUE again to take effect for the changed value.

5) Execute command multi-start by selecting cache mode

By selecting the cache mode, additional instances of commands can be executed in the action, i.e., multiple starts.

The time at which the action starts can be specified by setting an input called Buffer Mode selection.

The following modes are available in the Buffer Mode selection:

Cache Mode	Action
Interrupting Aborting	Can interrupt other actions to execute new Commands immediately

Cached Buffered	will wait until the output variables Done and InVelocity of the executing command become TRUE
Blending Low	Passes the end position of the first move at the lower speed of the two move commands.
Blending Previous	The function block starts immediately after the end of the last commanded move. The axis does not stop between moves but passes the end position of the first move at the speed of the first move command.
Blending Next	The function block starts immediately after the end of the last commanded move. The axis does not stop between moves but passes the end position of the first move at the speed of the second move command.
Blending High	The function block starts immediately after the last commanded move is finished. The axis does not stop between moves but passes the end position of the first move with the higher speed of the two move commands.

2. 1. 3 Precautions for synchronous control of the master and slave axes

The following are the precautions and abnormal conditions that occur when the spindle and slave axis of synchronous control undergo a rapid change in speed.

1. When the speed changes sharply

When the speed of the spindle and slave axis changes sharply at the beginning of the synchronous action or during the action, the action of the slave axis will change sharply, and excessive force may be applied to the device.

Please note that the speed of the spindle and slave axis may change sharply under the following conditions.

- When the following four commands are executed by the master and slave axes

a. MC_Stop(Stop) command

b. MC_SetPosition (change current position) command

c. MC_Reset(reset) command

d. MC_MoveVelocity (velocity control) command

Please set the input parameters and start time of the above command correctly or start the above command after the synchronous control is released to prevent the slave axis from moving sharply.

- When the immediate stop input signal and limit stop input signal of the master axis and slave axis change to TRUE
- When the master axis and slave axis change from servo ON to servo OFF,When the master axis and slave axis are vertical, the speed may change sharply when the servo is turned off.

Take countermeasures such as setting a brake on the spindle or slave axis, or turning the servo OFF state after the synchronization control ends, to prevent the slave axis from moving rapidly.

- When switching the control mode of the Servo drive

Note the change in speed when the command is activated.

Please set the input parameters of the command appropriately.

2. Conditions for antibiotic abnormalities

When the following four conditions occur in the spindle and slave axis at the start of synchronous operation or during synchronous operation, "Spindle position reading error" or "Slave axis position reading error" occurs in the slave axis.

At the same time, the output variable "CommandAborted (execution interrupt)" of the synchronous control command becomes TRUE.

a. When EtherCAT process data communication is not established

b. When "Slave communication error" occurs in the state "EtherCAT communication not established".

c. When the "Cannot calculate the current position of the absolute encoder" exception occurs

d. When the slave station is disengaged, and the synchronization control command multiple start is executed for the slave axis.

e. In the case of multi-start, even if any one of the above four conditions occurs for the spindle and slave axis, the multi-start will be accepted normally, and the buffer state will be entered.

2. 2 Basic knowledge of motion control commands

The interface definition of the MC function block and the program specification are explained.

2. 2. 1 Names of motion control commands

All motion control commands of the MC function block start with the word "MC".

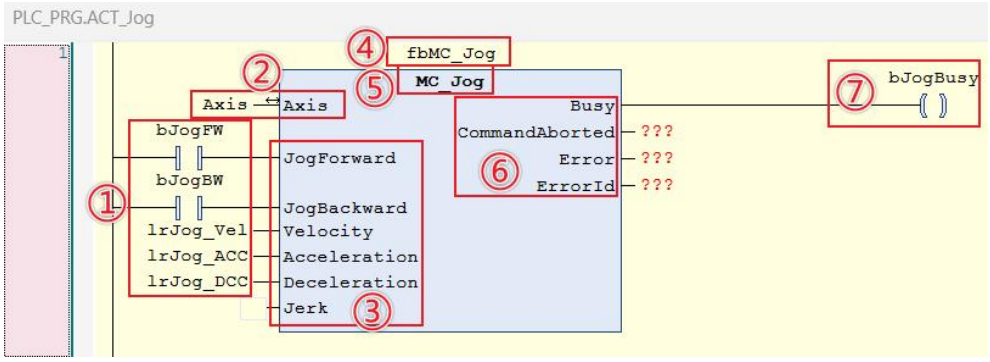
2. 2. 2 Language of motion control commands

All motion control commands of the MC function block can be used in the following two programming languages, and are generally used in Structured Text (ST).

（1） Ladder Diagram (LD)

Commands for writing motion controller in ladder language. The following is an example of the MC_Jog (jog) command:

① Input parameter
② Input output variables
③ Input variables
④ Instantiated Function Block Name
⑤ Main function Block Name
⑥ Output variables
⑦ Output parameter



a. Specify the axis variable name of the servo drive, etc. to be controlled to the input/output variable Axis.

b. Specify the action conditions such as target position and target speed to each input variable.

c. Outputs the status of the command and the status of the servo driver to each output variable.

d. When each input parameter is omitted, it becomes the initial value of each input variable.

（2） Structured Text (ST)

Specifies the command instance name. The following is an example description of the MC_Jog (jogging) command.

```
MC_J(
    Axis:= Axis.
    JogForward:= bJogFW.
    JogBackward:= bJogBW.
    Velocity:= lrJog_Velocity.
    Acceleration:= lrJog_ACC.
    Deceleration:= lrJog_DCC.
    Jerk:= .
    Busy=> bJogBusy.
    CommandAborted=> .
    Error=> .
    ErrorId=> ).
```

2. 2. 3 Configuration of motion control commands

The following explains in which tasks the motion control commands can be configured and what actions will occur at which locations in the program.

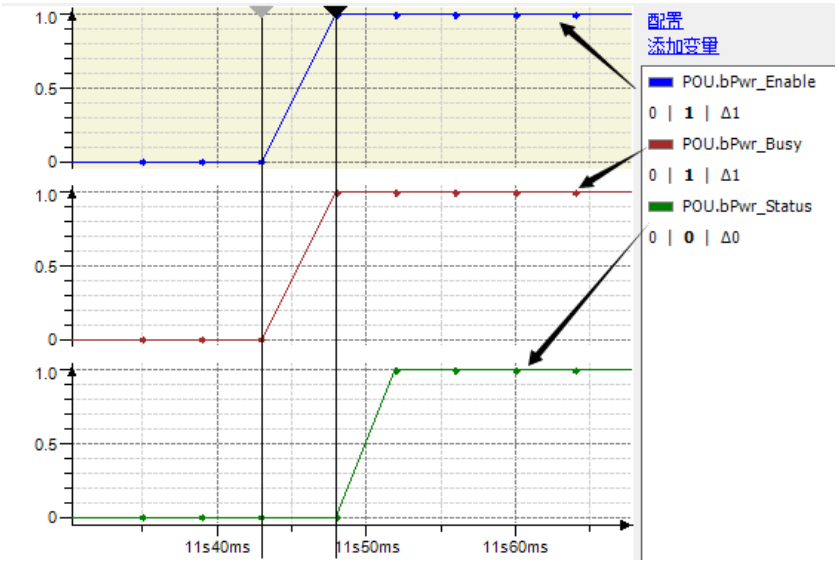
- Within functional block definition

Motion control commands can also be used within user-created function block definitions.

- For Enable type motion control command

Enable, Busy, and Status startup status diagrams. The following MC_Power example illustrates.

```
MC_Power(
    Axis:= Axis,                    //enable axis
    Enable:= bPwr_Enable,           //Servo ON
    bRegulatorOn:= bPwr_Enable.
    bDriveStart:= bPwr_Enable.
    Status=> bPwr_Status.
    bRegulatorRealState => .
    bDriveStartRealState => .
    Busy => bPwr_Busy.
    Error=> bPwr_Err.
    ErrorID=> Pwr_ErrID)
```



- For Execute type motion control command

Abs_Execute, Abs_Busy, Abs_Active, Abs_CommandAborted, Abs_Done startup status charts.

```
MC_MoveAbs(
    Axis:= Axis.
    Execute:= Abs_Execute.
    Position:= 100000.
    Velocity:= 100000.
    Acceleration:= 100000.
    Deceleration:= 100000.
    Jerk:= 100000.
    Direction:= MC_Direction.fastest.
    BufferMode:= .
    Done=> Abs_Done.
```

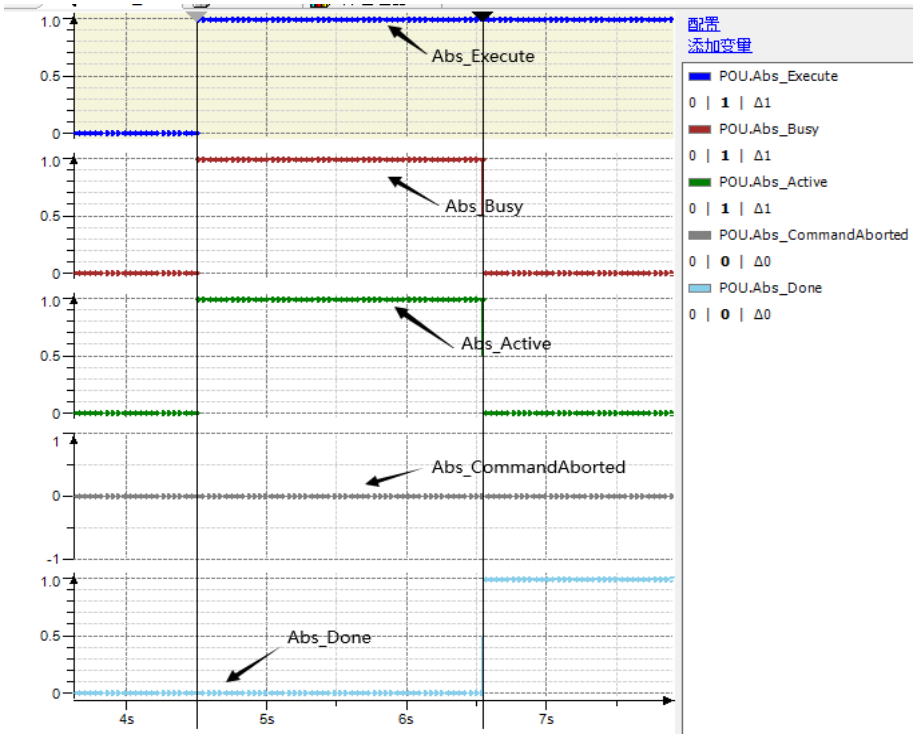
Busy => Abs_Busy.

Active=> Abs_Active.

CommandAborted=> Abs_CommandAborted.

Error=> .

ErrorID=>).



- Data type

CodeSys data types, specifying their upper and lower bounds, and memory:

Data Type	Lower limit	Upper limit	Memory
BYTE	0	255	8-bit
WORD	0	65535	16-bit
DWORD	0	4294967295	32-bit
LWORD	0	264-1	64-bit
SINT	-128	127	8-bit
USINT	0	255	8-bit
INT	-32768	32767	16-bit
UINT	0	65535	16-bit
DINT	-2147483648	2147483647	32-bit
UDINT	0	4294967295	32-bit
LINT	-263	263-1	64-bit
ULINT	0	264-1	64-bit

Section 3 Summary of Variables and Commands

3. 1 Variable Summary

The variables used by the MC function module are divided into two categories:

The system-defined variables used by MC function modules are called motion control system variables.

There is also a category of motion control commands that receive the independent variables of the command as input and the execution status of the command as output. Several of the input variables of the motion control commands are enumerated body variables selected from the options.

- Motion control system variables

Level 1	Content
Axis variables	Monitoring the status of each axis and setting contents of some axes
Axis group variables	The status of each axis group and the setting of some axis group parameters can be monitored

- Motion control command variables

Category	Outline
Input Variables	The independent variables of the command
Output Variables	Monitoring of command execution status
Input and output variables	Designation of the object of command execution

3. 1. 1 Axis variables

The axis variables are the interface definitions for the SoftMotion CiA402 axes, and the following are descriptions of some common interfaces:

Variable Name	Data Type	Name	Explanation
nAxisStatus (Axis state)	SMC_AXIS_STATE (ENUM)	0: power_off	Not Enabled
		1: errorstop	Error Stop
		2: stopping	Error stop in progress
		3: standstill	Enabling the ready state
		4: discrete_motion	Discrete Motion
		5: continuous_motion	Continuous motion
		6: synchronized_motion	Synchronized Movement
		7: homing	Origin copying in progress

nDirection (Direction)	MC_DIRECTION (ENUM)	-1: Negative	Negative directional motion
		0: shortest	Orientation selection based on shortest distance (modulus axis only)
		1: Positive	Positive directional motion
		2: current	Maintain the current direction to reach the target (module axis only).
		3: fastest	Automatic direction selection to get to the target position as fast as possible (modulo axes only)
Act	lreal	fActPosition	Feedback Location
	lreal	fActVelocity	Feedback speed
	lreal	fActTorque	Feedback torque
	lreal	fActJerk	Feedback leap
	lreal	fActAcceleration	Feedback acceleration
	lreal	fActCurrent	Feedback current

Variable Name	Data Type	Name	Explanation
Set	Lreal	fSetPosition	Distribution Location
	Lreal	fSetVelocity	Despatch speed
	Lreal	fSetTorque	Downward torque
	Lreal	fSetJerk	Downward leap
	Lreal	fSetAcceleration	Send down acceleration
	Lreal	fSetCurrent	Downstream current
Limit	Bool	bSWLimitEnable	Limit switch, true open
	Bool	bHWLimitEnable	Limit switch, true open
	Lreal	fSWLimitPositive	Positive limit value
	Lreal	fSWLimitNegative	Negative limit value
Maximum value	Lreal	fMaxCurrent	Maximum current
	Lreal	fMaxTorque	Maximum torque
	Lreal	fMaxJerk	Maximum Leap
	Lreal	fMaxVelocity	Maximum speed
	Lreal	fMaxPositionLag	Maximum position

3. 1. 2 Axis group variables

The axis group variables are the interface definitions for the SMC_AXIS_GROUP axis, and the following is an introduction to some common interfaces:

Variable Name	Data Type	Name	Explanation
Status (Axis group status)	SMC_AXIS_GROUP_STATE (ENUM)	Disable	Not Enabled
		Standby	Emotional state
		Moving	In Motion
		Homing	Origin reset in progress
		Stopping	Stop in progress
		ErrorStop	Error stop in progress

3. 2 Commands Summary

3. 2. 1 Input Variables for Motion Control Commands

The following are descriptions of the input variables, valid ranges of input values, and initial values for motion control commands.

- Input Variable Form

Input Variables	Name	Data Type	Effective range	Initial Value	Content
Execute	Start	BOOL	TRUE, FALSE	FALSE	At the rising edge of Execute, other input variables are imported. If you want to update the input value, change the input value first, and then start Execute again. The output variables are valid when Execute becomes TRUE even after the Command is executed. After that, the output variables other than Error and ErrorID are invalid on the falling edge of Execute. When Execute becomes FALSE before Command execution is completed, the output variables are valid for at least one cycle.

Input Variables	Name	Data Type	Effective range	Initial Value	Content
Enable	Effective	BOOL	TRUE, FALSE	FALSE	The function of the command becomes effective when it changes to TRUE and invalid when it changes to FALSE. When Enable is TRUE, other input variables are imported by cycle. When Enable is changed to FALSE, output variables other than Error and ErrorID are invalid.
Buffer Mode	Cache mode selection	MC_ BUFFER _MODE	0: Aborting 1: Buffered 2: BlendingLow 3: BlendingPrevious 4: BlendingNext 5: BlendingHigh	0	Specifies the action to be taken when multiple start motion control commands are given. 0: Interrupt 1: Cache 2: Merge at low speed 3: Previous one speed merge 4: Later a speed merge 5: Merge at high speed
Velocity	Target speed	LREAL	Positive numbers	0	Specify the target speed.
Acceleration	Acceleration	LREAL	Positive number or "0"	0	Specifies the acceleration.
Deceleration	Deceleration	LREAL	Positive number or "0"	0	Specify the deceleration rate.
Jerk	Jumpiness	LREAL	Positive number or "0"	0	Specify the degree of leap.
Distance	Travel distance	LREAL	Negative numbers, positive numbers, "0"	0	Specifies the distance to move from the current position of the command.
		ARRAY [0..3]OF LREAL	Negative numbers, positive numbers, "0"	0	Specifies the target position for linear interpolation.
Position	Target Location	LREAL	Negative numbers, positive numbers, "0"	0	Specifies the target position in absolute coordinates.
		ARRAY [0..3] OF LREAL	Negative numbers, positive numbers, "0"	0	Specifies the target position for linear interpolation.
VelFactor	Speed Hypertonic Value	LREAL	0～500	100	Specify the overshoot value of the speed. The valid range of the overshoot value is "0.01 to 500.00". "500.00 or more" is treated as "500", and "0.01 or less (including negative numbers)" is treated as "0.01 " is processed. Only when "0" is specified, the action is treated as "0". Unit is [%].

Input Variables	Name	Data Type	Effective range	Initial Value	Content
AccFactor (Reserved)	Acceleration and deceleration Hypertonic Value	LREAL	0～500	100	(Reserved)
JerkFactor (Reserved)	Jumpiness Hypertonic Value	LREAL	0～500	100	(Reserved)
Periodic	Repeat Mode	BOOL	TRUE, FALSE	FALSE	Specifies whether the specified cam table is executed repeatedly or only 1 time. TRUE: Repeat FALSE: No repetition
Master Start Distance	Spindle tracking Distance	LREAL	Negative numbers, positive numbers, "0"	0	Specify the position of the spindle when the slave axis starts cam action. When StartMode is specified as [Absolute positioning], the absolute position of the spindle is specified. When relative positioning is specified, the relative amount from StartPosition (cam table start position) is specified.
MasterScaling	Spindle factor	LREAL	Positive value (>0.0)	1.0	Enlarges/reduces the phase of the spindle by the specified ratio.
SlaveScaling	Slave coefficient	LREAL	Positive value (>0.0)	1.0	Enlarges/reduces the displacement from the axis by the specified ratio.
MasterOffset	Spindle offset	LREAL	Negative numbers, positive numbers, "0"	0	Shifts the phase of the spindle by the specified offset value.
SlaveOffset	Offset from axis	LREAL	Negative numbers, positive numbers, "0"	0	Moves the displacement from the axis by the specified offset value.
Continuous (Reserved)	Continuous method selection	BOOL	TRUE, FALSE	FALSE	(Reserved)
Ratio Numerator	Gear More than molecules	DINT	Positive or negative numbers	10000	Specify the numerator of the electronic gear between the main shaft and the slave shaft.
Ratio Denominator	Gear Ratio of denominators	UDINT	Positive numbers	10000	Specify the denominator of the electronic gear between the main shaft and the slave shaft.

Input Variables	Name	Data Type	Effective range	Initial Value	Content
MasterSync Position	Spindle synchronization Location	LREAL	Negative numbers, positive numbers, "0"	0	Specifies the spindle synchronization position in absolute coordinates.
SlaveSync Position	From the shaft Synchronization Location	LREAL	Negative numbers, positive numbers, "0	0	Specifies the synchronization position of the slave axis for absolute coordinates.
SlaveDistance	Travel distance from axis	LREAL	Negative numbers, positive numbers, "0"	0	Specifies the distance to move from the axis.
Execution Mode (Reserved)	Execution mode selection	MC_ EXECUTIO N_MODE	0: Immediately. Queued	0	(Reserved)
Permitted Deviation	Inter-axis deviation Allowable value	LREAL	Positive number or "0"	0	Specify the maximum allowable position deviation of the master and slave axes.
CoordSystem	Coordinate system	MC_ COORD_ SYSTEM	0: ACS 1: MCS 2: WCS 3: PCS_1 4: PCS_2 5: TCS	0	Specify the coordinate system 0: Axis coordinate system (ACS) 1: Machine Coordinate System (MCS) 2: World Coordinate System (WCS) 3: Workpiece coordinate system1 4: Workpiece coordinate system2 5: Tool coordinate system
Transition Mode	Transition Mode (Switching mode)	MC_ TRANSITIO N_MODE	0: TMNone TMStartVelocity TMCornerDistance	0	Specify the path of the action TMNone (no mix) TMStartVelocity (Speed-based mixing) TMCornerDistance (Distance-based mixing)

Input Variables	Name	Data Type	Effective range	Initial Value	Content
CircMode	Circular interpolation mode	MC_CIRC_MODE	0: Border 1: Center 2: Radius	0	Specifying the method of circular interpolation 0: Designated by point 1: Center point designation 2: Radius designation
AuxPoint	Auxiliary Points	SMC_POS_REF	a. c. v.	0	a: Axis coordinates c: Cartesian position v: value of the array, interpretation depends on the coordinate system used
EndPoint	Endpoint	SMC_POS_REF	a. c. v.	0	a: Axis coordinates c: Cartesian position v: value of the array, interpretation depends on the coordinate system used
PathChoice	Path Selection	MC_CIRC_PATHCHOICE	0: CLOCKWISE 1：COUNTER_CLOCKWISE	0	Specify the path direction 0: CW 1: CCW
EnableMask	Valid for track section	WORD	16#0000～FFFF	0	Specifies validity or invalidity by track segment. For up to 16 track segments, the value of bit 0 specifies valid or invalid for track segment number 0, and the value of bit 15 specifies valid or invalid for track segment number 15. 0: invalid 1: valid

3. 2. 2 Output Variables of Motion Control Commands

The following is a table of output variables for motion control commands:

Output Variables	Name	Data Type	Effective range	Content
Done	Completion	BOOL	TRUE, FALSE	The command becomes TRUE when execution is completed. Currently, the output variables Active, Error, and CommandAborted are FALSE. When the Command completes, Done is TRUE for at least one cycle when the input variable Execute is FALSE. When Execute is TRUE, Done remains TRUE until Execute becomes FALSE.
Busy	Under implementation	BOOL	TRUE, FALSE	The command is received and becomes TRUE.
Enabled	Effective	BOOL	TRUE, FALSE	Change to TRUE in the control.
Command Aborted	Execution Interruptions	BOOL	TRUE, FALSE	The command cannot be started when an abnormality occurs in the target axis or axis group. Similarly, the Command cannot be started during deceleration and stop. If another Command is started or an exception other than this Command occurs, the Command is interrupted. In this case, the output variables Done, Active, and Error become FALSE. When an interrupt occurs while the input variable Execute is FALSE, CommandAborted is TRUE for at least one cycle. When Execute or Enable is TRUE, CommandAborted remains TRUE until Execute or Enable becomes FALSE.

Output Variables	Name	Data Type	Effective range	Content
Error	Error	BOOL	TRUE, FALSE	TRUE if an exception occurs due to an input variable error or command processing.
ErrorID	Error Code	WORD		When an exception occurs, an error code is output. 16#0000 means normal.
Status	Runnable	BOOL	TRUE, FALSE	Becomes TRUE when it enters the runnable state.
EndOfProfile	Cam cycle completion	BOOL	TRUE, FALSE	Changes to TRUE after executing the end of the cam table.
Index	Index	UINT	Positive number or "0"	Outputs the index number of the cam data.
StartSync	Tracking in progress	BOOL	TRUE, FALSE	Becomes TRUE when the acceleration/deceleration action starts for synchronization.
Recorded Position	Locking Location	LREAL	Negative numbers, positive numbers, "0"	Outputs the locked position.
InVelocity	Reaching target speed	BOOL	TRUE, FALSE	InVelocity reaches the target speed BOOL TRUE, FALSE up.
InSync	In sync	BOOL	TRUE, FALSE	Becomes TRUE when synchronized with the spindle, or when the slave axis reaches the slave synchronization position.
InGear	Gear Ratio Arrival	BOOL	TRUE, FALSE	Changes to TRUE when the target speed is reached from the axis.
Valid	Effective	BOOL	TRUE, FALSE	Change to TRUE in the control.
Position	Feedback current position	LREAL	Negative numbers, positive numbers, "0"	Outputs the current value of the feedback position.
bInPosition	In place	BOOL	TRUE, FALSE	The current position of the feedback of all constituent axes becomes TRUE when it is within the in-place width of the target position.
InOperation	In Motion	BOOL	TRUE, FALSE	becomes TRUE in the command action.

3. 2. 3 Input and Output Variables of Motion Control Commands

The following is a table of input and output variables for motion control commands:

Input and output variables	Name	Data Type	Content
Axis	Shaft	AXIS_REF	Designated axis
AxisGroup	Shaft set	AXIS_GROUP_REF	Designated axis group
Slave	From the shaft	AXIS_REF	Specify the slave axis
Master	Main shaft	AXIS_REF	Specify spindle
CamTable	Cam table	MC_CAM_REF	Specify the array variable of the cam data structure MC_CAM_REF type as the cam table.
TriggerInput	Trigger input conditions	TRIGGER_REF	Set trigger conditions
NumAxes	Shaft group composition shaft	UDINT	Specify the axis number of the rewritten constituent axis
Switches	Switch	MC_ CAMSWITCH_REF	Specify the array variable of the switch structure MC_CAMSWITCH_REF type as the ON/OFF mode data of the switch. The array element number indicates the switch number.
Outputs	Output Signal	MC_OUTPUT_REF	Specify the array variable of the output signal MC_OUTPUT_REF type as the output object of the ON/OFF moment of the digital output calculated based on the ON/OFF mode data of the switch. The array element number indicates the track segment number. The actual digital output is set to ON/OFF by designating this array variable as the input and output variable of the NX_AryDOutTimeStamp Command.
TrackOptions	Track Section Options	MC_TRACK_REF	The track segment option structure MC_TRACK_REF An array variable of the type specified as the action condition of the switch. The array element number indicates the track segment number.

3. 2. 4 Commands

Motion control commands are divided into the following 3 types.

Category	Outline
General Commands	MC Function Module General Commands
Axis commands	MC function module executes commands for single-axis control
Axis group commands	MC function module executes commands for coordinated multi-axis control

- General Commands

The following is a description of the MC function module general commands:

Control operation	MC commands to be used	Description
Write MC settings	MC_Write	Write to MC settings

- Axis commands

The MC/SMC function blocks commonly used for single-axis control are shown in the following table:

Control operation	MC commands to be used	Description
Servo Enable	MC_Power	Run this command to enable the servo axis for subsequent operation control
Servo Point Dynamic operation	MC_Jog	Point operation of servo motor, often used for low-speed test run, for checking equipment or adjusting servo motor position
Relative positioning	MC_MoveRelative	Run the specified distance with the current position as reference
Relative Overlay Positioning	MC_MoveAdditive	In addition to the servo's current operation command, the specified distance is run relative to the servo.
Absolute positioning	MC_MoveAbsolute	The command servo runs to the specified coordinate point
Speed Control	MC_MoveVelocity	Command Servo to run at the specified speed
Torque control	MC_MoveTorque	Commands the servo to run at the specified torque
Servo pause	MC_Halt	Command the servo to pause, if MC_Movexxx is triggered again, the servo can run again.
Emergency shutdown	MC_Stop	Only after the stop command is reset and MC_ Movexxx is triggered, the servo can run again.

Control operation	MC commands to be used	Description
Alarm reset	MC_Reset	When the servo has an alarm stop, run this command to reset
Change servo Operation mode	MC_SetControlMode	This command allows the servo to select "position", "speed" or "torque" mode.
Servo origin return	MC_Home	Command the servo to start the home return operation, the home signal of the application system, both side limit signals, etc. are connected to the DI port of the servo
Controller Return to the origin	SMC_Homing	The control system starts home return operation, and the home signal of the application system and the limit signals on both sides are connected to the DI port of the controller
Speed Control	MC_MoveVelocity	Command Servo to run at the specified speed
Torque control	MC_SetTorque	Commands the servo to run at the specified torque
Location Setting	MC_SetPosition	Current position change

- MC commands and PDO/SDO configuration

Relationship between commonly used MC commands and PDO/SDO usage:

MC Commands	Name	Required TPDO objects	Required RPDO objects
MC_SetTorque	Downward torque	16#6040: Target torque (Target torque)	16#6077: Torque actual value (Current torque)
MC_SetControlMode	Settings Operation mode	16#6060: Modes of operation (operation mode) 16#6060=8: Cycle synchronous position CSP 16#6060=9: Speed mode 16#6060=10: Torque mode	16#6061: Mode of Operation Display (Current operation mode)

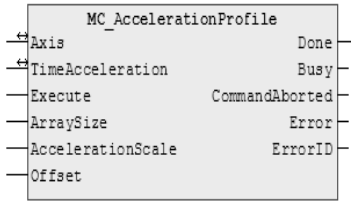
Section 4 Common MC commands in detail

4. 1 Single-axis commands

4. 1. 1 Acceleration Profile Commands: MC_AccelerationProfile

- **Acceleration profile module**

（1）Command format

Command	Name	Graphical representation	ST Performance
MC_AccelerationProfile	Acceleration Contour command		MC_AccelerationProfile(Axis:= , TimeAcceleration:= , Execute:= , ArraySize:= , AccelerationScale:= , Offset:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

（2）Relevant variables

Scope	Name	Chinese	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_ REF_SM3		Mapped to axis, an instance of AXIS_REF_SM3
	Time Acceleration	Axis acceleration Time and addition Speed description	MC_ TA_REF		Axis acceleration time and acceleration data are described, and the acceleration data consists of multiple data sets.
Input	Execute	Implementation conditions	BOOL		A rising edge of the input will start the function The place of the block.
	ArraySize	Dynamic arrays	INT		The number of arrays used in the run profile.
	Acceleration Scale	Combined factor	LREAL	1	Scale factor of acceleration or deceleration in MC_TA_REF.
	Offset	Offset	LREAL		The overall offset value of the acceleration and deceleration.

Scope	Name	Chinese	Type	Initial	Comment
Output	Done	Command Execution complete	BOOL		The axis Command is executed and set to TRUE
	Busy	Command Being implemented	BOOL		The current Command is being executed, set to TRUE
	Command Aborted	Command is interrupted	BOOL		Current Command is interrupted, set to TRUE
	Error	Error	BOOL		Set to TRUE when an exception occurs
	ErrorID	Error Code	SMC_ ERROR		Output error code when exception occurs

（3）Function description

This function block models the contour motion for time periods and acceleration/deceleration.

TimeAcceleration is MC_TA_REF data type:

a. MC_TA_REF is described as follows.

Members	Type	Initial Value	Description
Number_of_pairs	INT	0	Number of segments of the contour path
IsAbsolute	BOOL	TRUE	Absolute motion (TRUE) and relative motion selection
MC_TA_Array	ARRAY[1..N] OSMC_TA		Arrays of time and acceleration values

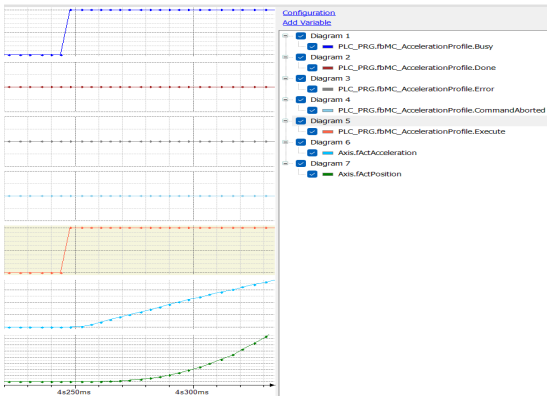
b. SMC_TA is described as follows .

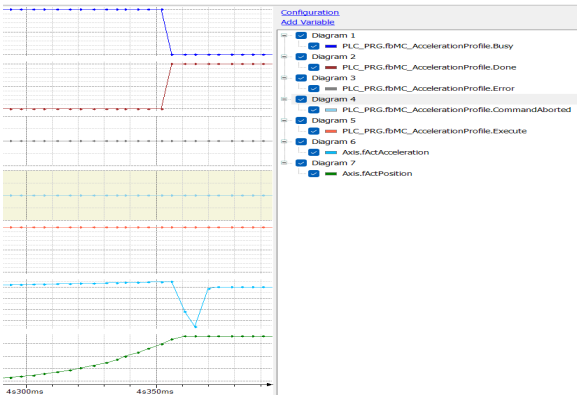
Members	Type	Initial Value	Description
delta_time	TIME	TIME#0ms	Time of acceleration section
acceleration	LREAL	0	Current acceleration value

c. Time-series diagram:

Execute of the function block must have a rising edge condition; Done of the function block indicates that the Command has completed normal execution.

Busy of a function block indicates that the function block is currently being executed.





4. 1. 2 Axis normal pause command: MC_Halt

- **Axis normal pause module**

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_Halt	Axis normal pause command	MC_Halt MC_Halt Axis → Done Execute → Busy Deceleration → CommandAborted Jerk → Error ErrorID	MC_Halt(Axis:= , Execute:= , Deceleration:= , Jerk:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

（2） Relevant variables

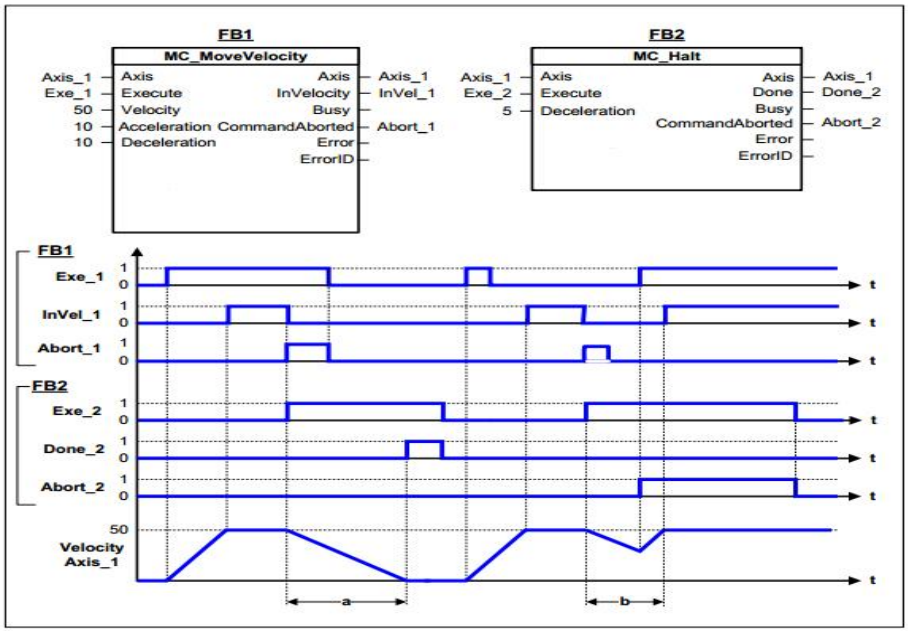
Scope	Name	Chinese	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_REF_SM3		mapped to the axis, an instance of AXIS_REF_SM3.
Input	Execute	Implementation conditions	BOOL	FALSE	A rising edge of the input will initiate the processing of the function block.
	Deceleration	Implementation conditions	LREAL	0	Deceleration of the functional block (u/S^2)
	Jerk	Jumpiness	LREAL	0	Specify the leap [command unit /S^3].

Input	Execute	Implementation conditions	BOOL	FALSE	A rising edge of the input will initiate the processing of the function block.
	Deceleration	Implementation conditions	LREAL	0	Deceleration of the functional block (u/S^2)
	Jerk	Jumpiness	LREAL	0	Specify the leap [command unit /S^3].
Output	Done	Command Execution Completion	BOOL	FALSE	The axis Command is executed and set to TRUE.
	Busy	The command is being Execution	BOOL	FALSE	The current Command is being executed, set to TRUE
	Command Aborted	Command Interrupted	BOOL	FALSE	The current Command is interrupted and set to TRUE.
	Error	Error	BOOL	FALSE	Set to TRUE when the exception occurs.
	ErrorID	Error Code	SMC_ERROR	0	Outputs an error code when an exception occurs.

（3） Function description

This function block stops the referenced axis in a controlled manner. If an operation of another function block is run at this time, the operation will stop. The axis discrete_motion state until it reaches speed 0. If the completion output of MC_Halt is set, the state of the axis will change to stationary. As long as MC_Halt is active, a new motion command can be issued to interrupt the execution of MC_Halt. unlike MC_Stop, MC_Halt can also be overridden.

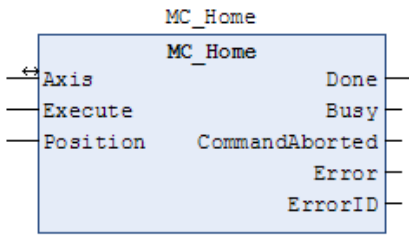
Time-series diagram:



4. 1. 3 Axis return to zero command: MC_Home

- **Back to zero**

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_Home	Axis return to zero command		MC_Home(Axis:= Axis, Execute:= , Position:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

（2） Relevant variables amount

Scope	Name	Chinese	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_REF_SM3		mapped to the axis, an instance of AXIS_REF_SM3.
Input	Execute	Implementation conditions	BOOL	**FALSE**	A rising edge of the input will start the function block.
	Position	Axis arrival position	LREAL	0	Represents the zero-return position of the axis position.
Output	Done	Command execution completed	BOOL	**FALSE**	The axis Command is executed and set to TRUE
	Busy	Command is being executed	BOOL	**FALSE**	The current Command is being executed, set to TRUE.
	Command Aborted	Command is interrupted	BOOL	**FALSE**	Current Command is interrupted, set to TRUE
	Error	Error	BOOL	**FALSE**	Set to TRUE when the exception occurs.
	ErrorID	Error Code	SMC_ERROR	0	When the abnormality occurs, the Outputs an error code.

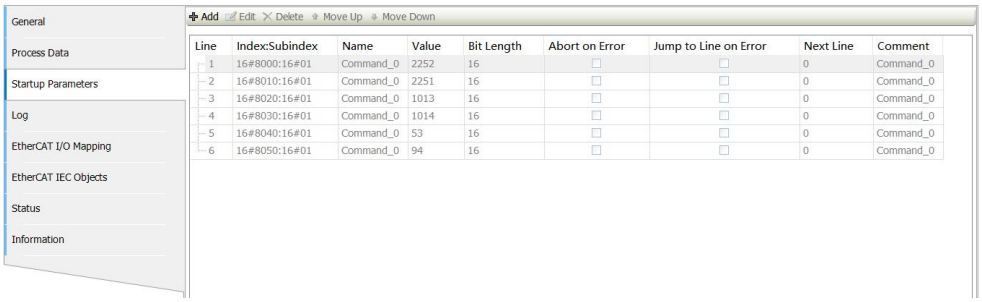
（3） Function description

This function block is for the drive to return to zero, if the requirement of return to zero accuracy is very high, please use MC_Home, MC_Home need to configure the servo's return to zero mode, return to zero speed, etc. first. The program triggers MC_Home, the PLC just changes the servo module into zero return mode, starts zero return and waits for the servo to return to zero by itself, feeds back the completion signal, and then switches the mode to the control mode before zero return.

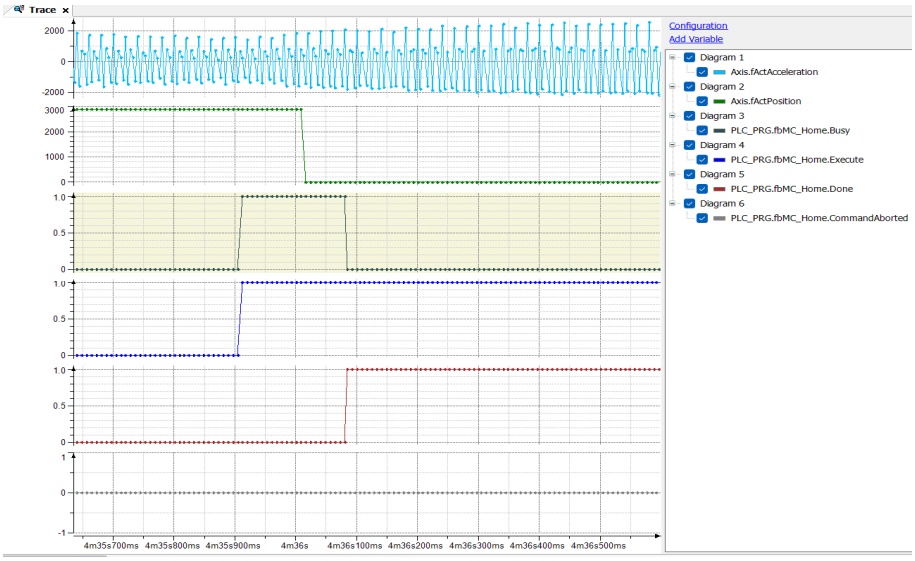
Servo settings are performed in two ways:

a. The return mode of the servo parameters must be set when using each servo axis to return at the home position; the setting mode can be set by the servo host computer or manually by setting the function code of the servo.

b. The corresponding function codes can also be configured via the start-up parameters of the CodeSys slave.SoftMotion CiA402 axis is set to return to zero in the following way:



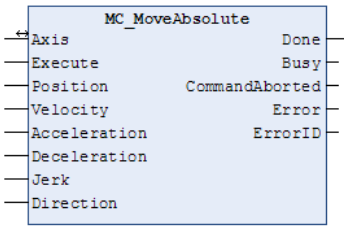
c. Time-series diagram:



4. 1. 4 Axis absolute position control command: MC_MoveAbsolute

- **Axis absolute position motion**

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_ MoveAbsolute	Axis absolute position Control commands		MC_MoveAbsolute(Axis:= , Execute:= , Position:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Direction:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_ REF_SM 3		Mapped to axis, an instance of AXIS_REF_SM3
Input	Execute	Implementation conditions	BOOL		A rising edge of the input will initiate the processing of the function block.
	Position	Axis arrival position	LREAL		This position is the absolute position data of the axis.
	Velocity	Running speed	LREAL		The maximum speed at which the axis runs to the target position.
	Acceleration	Acceleration	LREAL		The acceleration value when the speed becomes large.
	Deceleration	Deceleration	LREAL		The deceleration value when the speed becomes smaller.
	Jerk	Jumpiness	LREAL		The value of the slope change of the acceleration and deceleration of the curve.

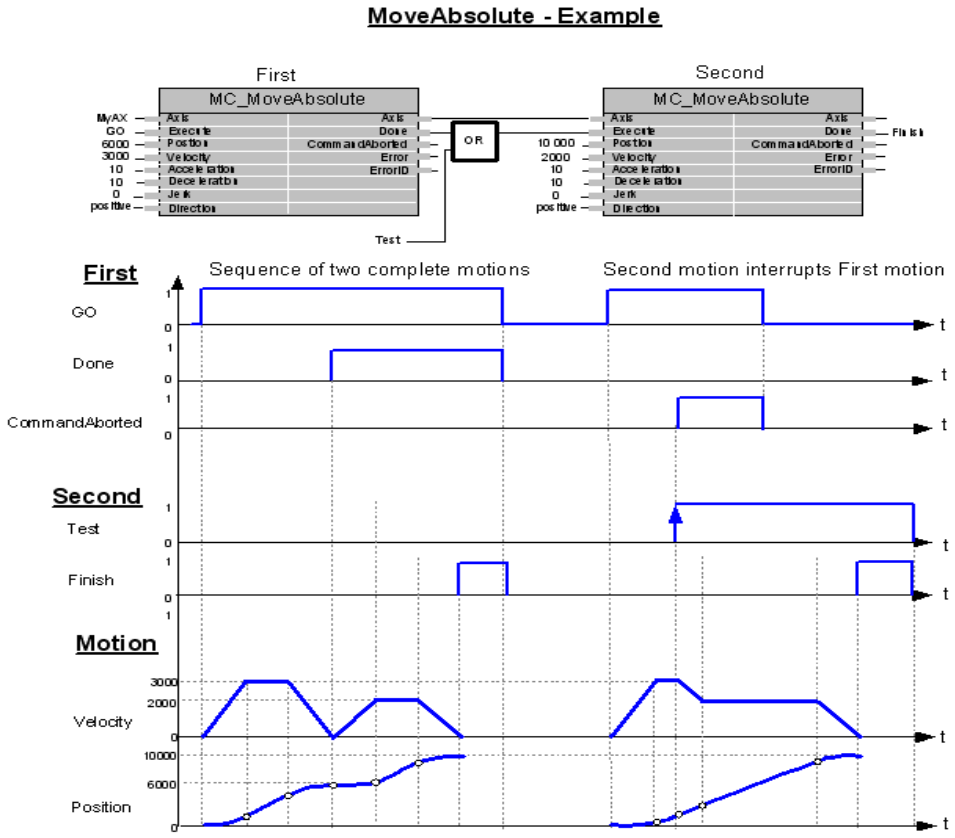
Scope	Name	Chinese Name	Type	Initial	Comment
Input	Direction	Command polarity	MC_ Direction	shortest	Negative:Reverse movement. Shortest:Choose the direction according to the shortest path. Positive:Positive movement. Current:Move in the current direction. Fastest:Automatically select the fastest direction to move; (This function is effective in the rotation mode.
Input	BufferMode		MC_ BUFFER _MODE		Defines the chronological sequence of the FB relative to the previous block. If the function block is Busy, then only BufferMode=Aborting is allowed.
Output	Done	Command Execution Completion	BOOL		The axis Command is executed and set to TRUE.
	Busy	The command is being Execution	BOOL		The current command is being executed, set is TRUE.
	Active	Command Activation	BOOL		Indicates that the FB has control on the axis
	Command Aborted	Command is interrupted	BOOL		The current Command is interrupted and set to TRUE.
	Error	Error	BOOL		Set to TRUE when the exception occurs.
	ErrorID	Error Code	SMC_ER ROR		Outputs an error code when an exception occurs.

（3） Function description

This function block moves the axis to the absolute position and uses velocity, deceleration, acceleration, and Jerk values. If no other operations are pending, the execution ends with velocity 0.

Time-series diagram:

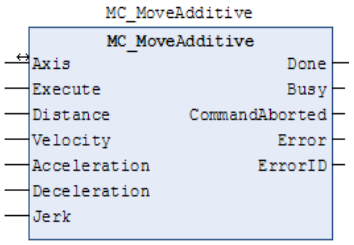
Using MC_MoveAbsolute The following diagram shows two possibilities of combining two instances (first and second) of the MC_MoveAbsolute type. On the left side of the diagram, the second instance is called after the first one. If the first instance has reached the specified position 6000 and the input velocity is 0, the completed output will cause the second instance to move the axis to position 10000. in the right part of the diagram, the execution is started by the second instance, while the first instance is still running. The motion caused by the first is interrupted and aborted by the test signal transmitted during the first constant velocity phase. The second one turns directly to position 10000, even though it has not yet reached position 6000



4. 1. 5 Superimposed absolute motion command: MC_MoveAdditive

The axis is superimposed on the original command position with the data specified by Distance for the online superimposed position of the motion axis control process.

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_ MoveAdditive	Superimposed absolute Motion Commands		MC_MoveAdditive(Axis:= , Execute:= , Distance:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	**AXIS_ REF_SM3**		Mapped to axis, an instance of AXIS_REF_SM3
Input	Execute	Execution Conditions	BOOL	**FALSE**	A rising edge of the input will initiate the processing of the function block.
	Distance	Axis arrival position	LREAL	0	This data is the superimposed position data.
	Velocity	Running speed	LREAL	0	The maximum speed at which the axis runs to the target position.
	Acceleration	Acceleration	LREAL	0	The acceleration value when the speed becomes large.
	Deceleration	Deceleration	LREAL	0	The deceleration value when the speed becomes smaller.
	Jerk	Jumpiness	LREAL	0	The value of the slope change of the acceleration and deceleration of the curve.
Output	Done	Command Execution Completion	BOOL	FALS E	The axis Command is executed and set to TRUE.

Scope	Name	Chinese Name	Type	Initial	Comment
	Busy	The command is being Execution	BOOL	FALSE	The current Command is being executed, set to TRUE
	Command Aborted	Command is interrupted	BOOL	FALSE	The current Command is interrupted and set to TRUE.
	Error	Error	BOOL	FALSE	Set to TRUE when the exception occurs.
	ErrorID	Error Code	SMC_ERROR	0	Outputs an error code when an exception occurs.

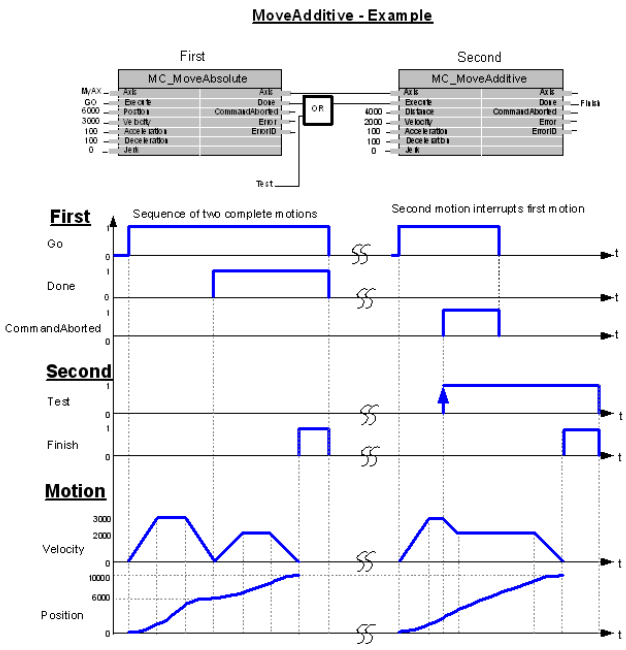
（3） Function description

This function block causes a control motion that adds the specified distance to the last specified target position. As a result, the axis is in discrete_motion mode. The current target position may come from a previously aborted MC_MoveAdditive motion.

If the function block is running in continuous_motion mode, the specified distance is added to the current position during processing.

Time-series diagram:

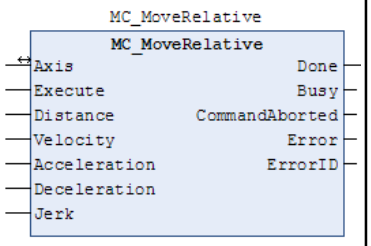
Using MC_MoveAdditive The figure shows how the first instance of type MC_MoveAbsolute and the second instance of type MC_MoveAdditive are combined. The axes are in discrete motion mode. On the left side of the diagram, second instance is called after the first instance. If the first instance has reached the specified position 6000, the velocity is 0 and the setup is complete, second will move the axis to position 10000. In the right part of the diagram, second instance starts executing while the first is still running. first's motion is interrupted and aborted by the test signal transmitted by the first instance during the constant speed phase. The distance 4000 plus the last recommended position 60000. then second instance moves the axis to the result position 10000.



4. 1. 6 Axis Relative Positioning Command: MC_MoveRelative

The axes run in relative positions (units are set by axis), and the relative positions are specified by Distance.

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_MoveRelative	Axis relative positioning command		MC_MoveRelative(Axis:= , Execute:= , Distance:= , Velocity:= 10, Acceleration:= , Deceleration:= , Jerk:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Comment
Inout	Axis	Shaft	AXIS_REF_SM3	Mapping to an axis, an instance of AXIS_REF_SM3.
Input	Execute	Execution Conditions	BOOL	A rising edge of the input will start the function The processing of the block.
	Distance	Movement relative position	LREAL	This data is the relative position of the motion.
	Velocity	Running speed	LREAL	The maximum speed at which the axis runs to the target position.
	Acceleration	Acceleration	LREAL	The acceleration value when the speed becomes large.
	Deceleration	Deceleration	LREAL	The deceleration value when the speed becomes smaller.
	Jerk	Jumpiness	LREAL	The value of the slope change of the acceleration and deceleration of the curve.
	BufferMode		MC_BUFFER_MODE	Defines the chronological sequence of the FB relative to the previous block. If the function block is Busy, then only BufferMode=Aborting is allowed.

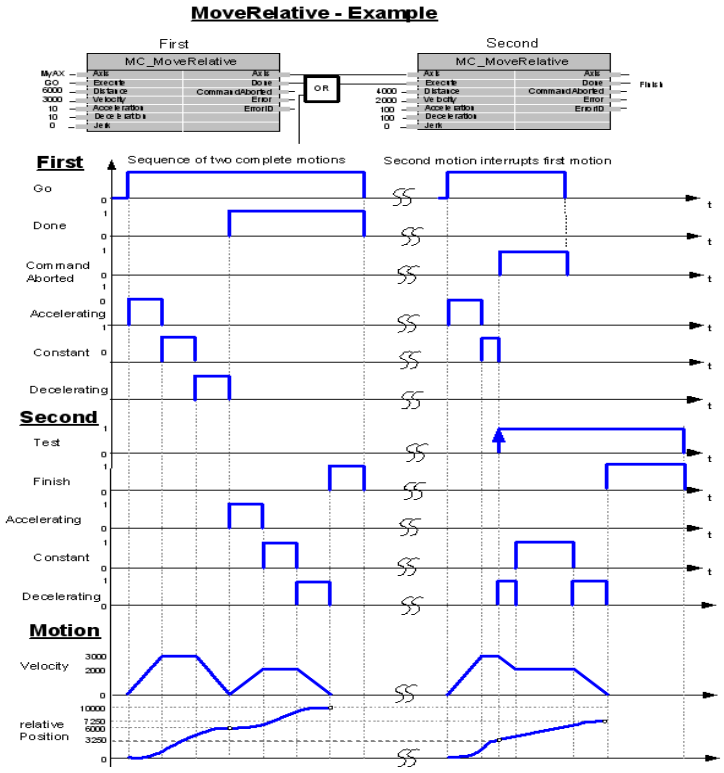
Output	Done	Command execution completed	BOOL	The axis Command is executed and set to TRUE.
	Busy	Command is being executed	BOOL	The current Command is being executed, set to TRUE.
	Active	Command Controls	BOOL	Indicates that the FB has control on the axis
	Command Aborted	Command is interrupted	BOOL	CommandAborted
	Error	Error	BOOL	Error

（3） Function description

This function block commands a controlled motion at the specified distance. The motion ends with a pause (unless mixed to a subsequent motion).

Time-series diagram:

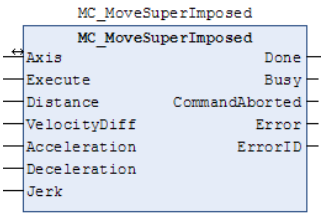
Using MC_MoveRelative; the diagram shows how the two instances (First and Second) of type MC_MoveRelative relative to each other are combined. On the left side of the diagram, Second is called after First. If First reaches the specified position at a distance of 6000, then the velocity is 0. First is set to complete. Then the second instance will cause the axis to move to position 10000. In the right part of the diagram, Second starts executing while First is still running. First's movement is interrupted and aborted by the Test signal transmitted by First during the constant velocity phase. The distance 4000 is added to the actual position 3250. then Second moves the axis to the resulting position 7250.



4. 1. 7 Superimposed relative motion command: MC_MoveSuperImposed

The axes stack acceleration and position data on top of the original command speed and position on the running Command, and there is no change in the execution time model for the entire original Command.

（1） Related Commands

Command	Name	Graphical representation	ST Performance
MC_MoveSuperImposed	Superimposed relative Motion Commands		MC_MoveSuperImposed(Axis:= , Execute:= , Distance:= , VelocityDiff:= , Acceleration:= , Deceleration:= , Jerk:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

（2） Relevant variables

Scope	Name	Chinese Name	Name	Comment
Inout	Axis	Shaft	AXIS_REF_SM3	Mapped to the axis, an instance of AXIS_REF_SM3
Input	Execute	Execution Conditions	BOOL	A rising edge of the input will initiate the processing of the function block.
	Abort		BOOL	
	Distance	Axis arrival position	LREAL	This data is the superimposed position data.
	VelocityDiff	Overlay Speed	LREAL	Axis running stack acceleration.
	Acceleration	Acceleration	LREAL	The acceleration value when the speed becomes large.
	Deceleration	Deceleration	LREAL	The deceleration value when the speed becomes smaller.
	Jerk	Jumpiness	LREAL	The value of the slope change of the acceleration and deceleration of the curve.
Output	Done	Command execution completed	BOOL	The axis Command is executed and set to TRUE.
	Busy	Command is being executed	BOOL	The current Command is being executed, set to TRUE.
	Command Aborted	Command Interrupted	BOOL	The current Command is interrupted and set to TRUE.
	Error	Error	BOOL	Set to TRUE when the exception occurs.
	ErrorID	Error Code	SMC_ERROR	Outputs an error code when an exception occurs.

（3） Function description

The function block causes additional movement on top of the continuous (original) movement of the axis. The original movement is not aborted, but the movements at the given distance are superimposed.

If MC_MoveSuperImposed is active, a further abort command on the same axis interrupts both commands, MC_MoveSuperImposed and the original command. A further buffer or blend command does not interrupt the superimposed motion. The superimposed motion continues at the same time.

The function block MC_MoveSuperImposed must be called after the function block of the original motion.

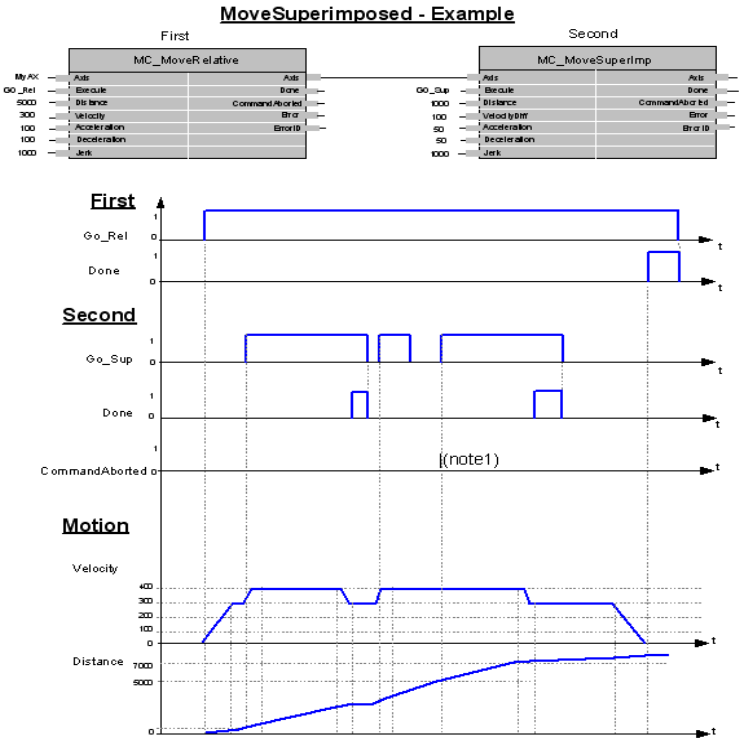
If the move command is called after MC_MoveSuperImposed, the error SMC_MSI_INVALID_EXECUTION_ORDER is returned.

If the MC_MoveSuperImposed instance is active and a second type MC_MoveSuperImposed instance is called, an error is reported by the second trial. If the MC_MoveSuperImposed instance is active and has a new rising edge (possibly with a different input) starting at execution, the active superimposed motion is aborted and replaced by the new superimposed motion while the original motion function block remains active.

In the stationary state of the axis, MC_MoveSuperImposed behaves like MC_MoveRelative.

The Acceleration, Deceleration and Jerk inputs are additional values (not absolute values) that are added to the original motion. The original function block will always complete its motion for the same duration and does not consider any instances of MC_MoveSuperImposed that act simultaneously.

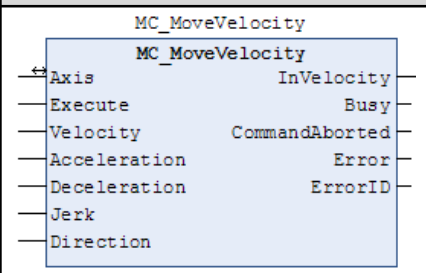
MC_MoveSuperImposed is active on the slave axis, while MC_Phasing operates on the main axis.



4. 1. 8 Velocity Control Command: MC_MoveVelocity

This module controls the specified speed sent in the position control mode.

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_MoveVelocity	Speed Control commands		MC_MoveVelocity(Axis:= , Execute:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Direction:= , InVelocity=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_REF_ SM3		Mapped to axis, an instance of AXIS_REF_SM3
Input	Execute	Implementation conditions	BOOL		A rising edge of the input will start the function The processing of the block.
	Velocity	Speed Set value	LREAL		This data is the speed transport value for this command.
	Acceleration	Acceleration	LREAL		The acceleration value when the speed becomes large.
	Deceleration	Deceleration	LREAL		The deceleration value when the speed becomes smaller.
	Jerk	Jumpiness	LREAL		The value of the slope change of the acceleration and deceleration of the curve.
	Direction	Running direction	MC_Direction	current	is the command operation in the run direction.

Scope	Name	Chinese Name	Type	Initial	Comment
Input	BufferMode		MC_BUFFER_MODE		Defines the chronological sequence of the FB relative If the function block is Busy, then only BufferMode=Aborting is allowed.
Output	InVelocity	The sign of reaching the set speed	BOOL		The set running speed is reached and set to TRUE.
	Busy	Command is being executed	BOOL		The current Command is being executed, set to TRUE.
	Active		BOOL		Indicates that the FB has control on the axis
Output	Command Aborted	Command Interrupted	BOOL	FALSE	The current Command is interrupted and set to TRUE.
	Error	Error	BOOL		Set to TRUE when the exception occurs.
	ErrorID	Error Code	SMC_ERROR		Outputs an error code when an exception occurs.

（3）Function description

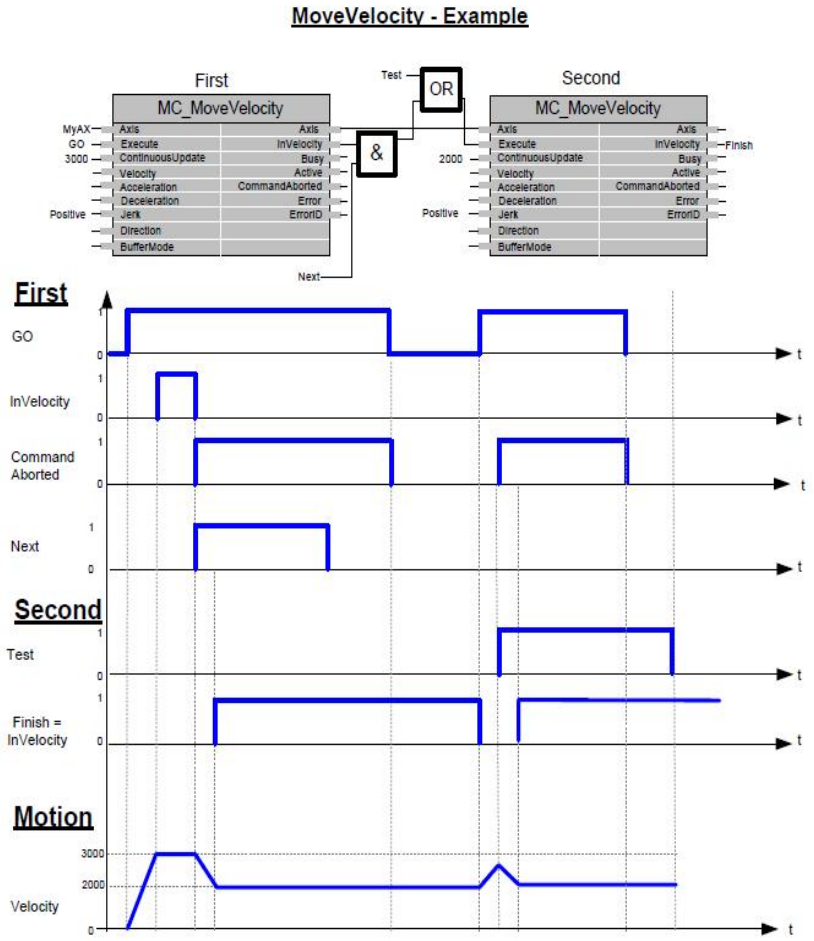
This function block causes an endless motion at the specified speed.

Time-series diagram:

Use MC_MoveVelocity

The following illustration shows how the two instances (first and second) of the function block MC_MoveVelocity are combined. In the left part of the illustration, the second instance is called after one trial. If the specified velocity 3000 is reached first, the first output velocity is ANDed with the next signal. this causes the second to move at a velocity of 2000.

In the right part of the illustration, the execution is started by First, which aborts the previously executed second trial, so the output InVelocity of the second instance is set to error. Although the first is still accelerating to reach the speed 3000, it is interrupted and aborted as the test signal starts to execute the second. Now, after the second will decelerate the speed to 2000, the rate of the second instance is set to true.



4. 1. 9 Position Profile Command: MC_PositionProfile

This function block is designed for command time position locking motion profiles.

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_PositionProfile	Location Contour command	MC_PositionProfile Axis TimePosition Execute ArraySize PositionScale Offset Done Busy CommandAborted Error ErrorID	MC_PositionProfile(Axis:= , TimePosition:= , Execute:= , ArraySize:= , PositionScale:= , Offset:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

（2）Relevant variables

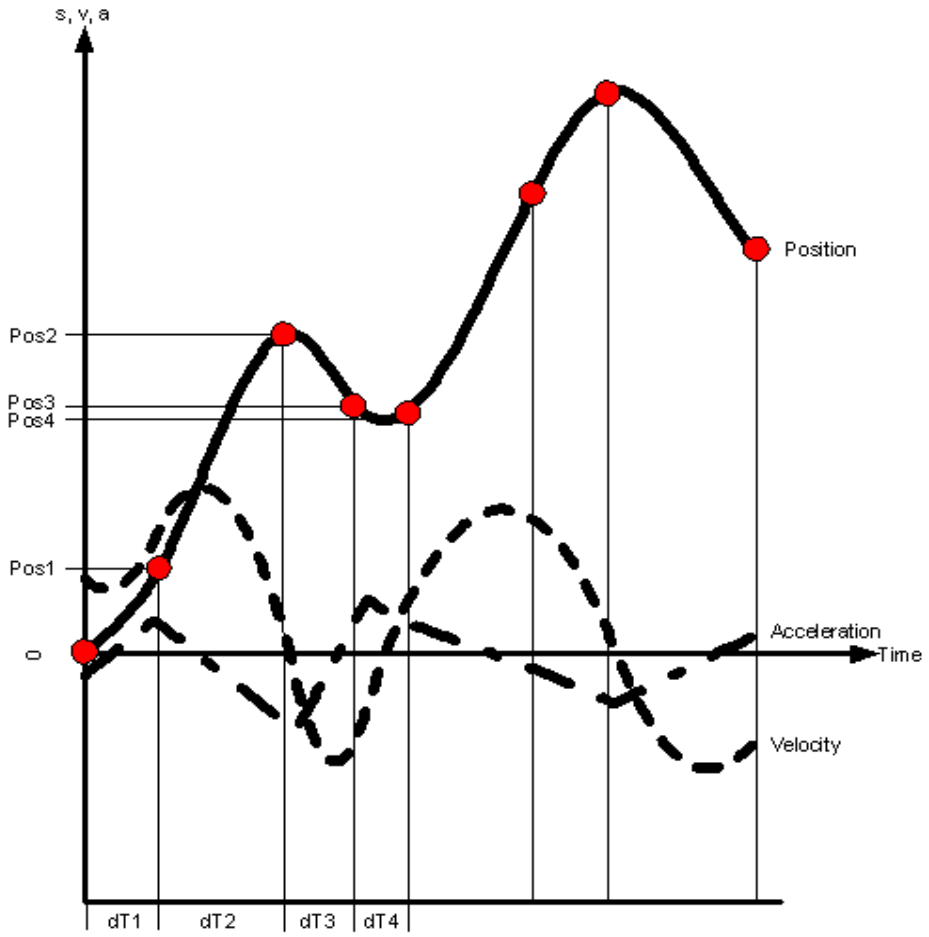
Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	**AXIS_REF_SM3**		Mapped to the axis, an instance of AXIS_REF_SM3
	Time Position	Axis position running time and position tracing	MC_ TP_REF		Axis position run time and position data are described, and the data consists of multiple data sets.
Input	Execute	Implementation conditions	BOOL		A rising edge of the input will initiate the processing of the function block.
	Array Size	Dynamic arrays	INT		The number of arrays used in the run profile.
	Position Scale	Combined factor	LREAL	1	Scale factor of the position in MC_TP_REF
	Offset	Offset	LREAL		The overall offset value of the position.
Output	Done	Command execution completed	BOOL		The axis Command is executed and set to TRUE.
	Busy	Command is being executed	BOOL		The current Command is being executed, set to TRUE.
	Command Aborted	Command is interrupted	BOOL		The current Command is interrupted and set to TRUE.
	Error	Error	BOOL		Set to TRUE when the exception occurs.
	ErrorID	Command execution completed	SMC_ ERROR		Outputs an error code when an exception occurs.

（3）Function description

This function block is designed for command time position locking motion profiles.

Tip:

Transitions between different profiles can be managed. Alternatively, the MC_PositionProfile can be used with a cam function block coupled to the virtual host.

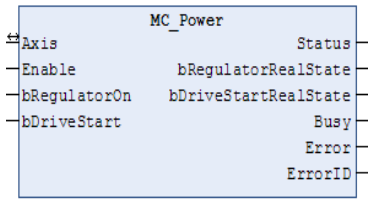


deltaTime	absPos
dT1	Pos1
dT2	Pos2
dT3	Pos3
dT4	Pos4

4. 1. 10 Axis enable command: MC_Power

- **Enable Function Block**

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_Power	Axis enable command		MC_Power(Axis:= , Enable:= , bRegulatorOn:= , bDriveStart:= , Status=> , bRegulatorRealState=> , bDriveStartRealState=> , Busy=> , Error=> , ErrorID=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Comment
Inout	Axis	Shaft	AXIS_REF_SM3	Mapped to the axis, an instance of AXIS_REF_SM3
Input	Enable	Effective	BOOL	Set to TRUE to start processing of the function block.
	bRegulatorOn	Enabled state	BOOL	Set to TRUE to set the axis to the enable state.
	bDriveStart	Allow drive	BOOL	Set to TRUE to turn off the emergency stop processing of the function block.
Output	Status	Operable state	BOOL	Set to TRUE if the axis is ready to move.
	bRegulatorRealState	Axis enable signal status	BOOL	Set to TRUE when the axis enable is active.
	bDriveStartRealState	Allow drive status	BOOL	Set to TRUE if the axis is not interrupted by the fast stop mechanism.
	Busy	Under implementation	BOOL	Set to TRUE if the processing of the function block is not completed.
	Error	Error	BOOL	Set to TRUE when the exception occurs.
	ErrorID	Error Code	SMC_	Outputs an error code when an

			ERROR	exception occurs.

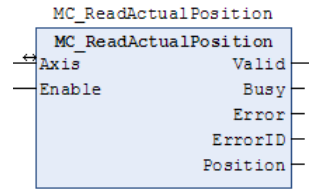
（3） Function description

MC_Power is designed to control the power phase ("on" or "off").

4. 1. 11 Read Actual Position Command MC_ReadActualPosition

The command reads the actual location where the drive is running and saves it in a variable cell defined by itself.

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_ ReadActualPosition	Actual Location Read command		MC_ReadActualPosition(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , Position=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_ REF_SM 3		Mapped to axis, an instance of AXIS_REF_SM3
Input	Enable	Implementation conditions	BOOL	**FALSE**	Reads the current position of the servo for the TRUE state.
Output	Valid	Location data obtainable flags	BOOL	**FALSE**	Set to TRUE if the drive position is correctly obtained.
	Busy	Command is being executed	BOOL	**FALSE**	The current Command is being executed, set to TRUE.
	Error	Error	BOOL	**FALSE**	Set to TRUE when the exception occurs.
	ErrorID	Error Code	SMC_ ERROR	0	Outputs an error code when an exception occurs.
	Position	Acquired axis position	LREAL	0	The axis position data read out by the command.

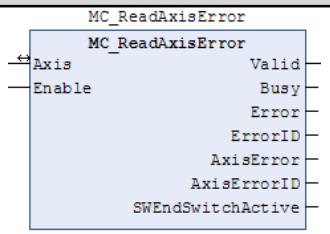
（3） Function description

This function block returns the current position of the referenced axis.

4. 1. 12 Read Axis Error Status Command MC_ReadAxisError

The command reads the error condition of the axis and saves it in a variable cell defined by itself.

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_ReadAxisError	Read axis of Error Status		MC_ReadAxisError(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , AxisError=> , AxisErrorID=> , SWEndSwitchActive=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_ REF_SM 3		Mapped to axis, an instance of AXIS_REF_SM3
Input	Enable	Implementation conditions	BOOL	FALSE	Reads the current position of the servo for the TRUE state.
Output	Valid	Wrong data Accessibility	BOOL	FALSE	Can get the error data of the axis, set to TRUE.
	Busy	Command is being executed	BOOL	FALSE	The current Command is being executed, set to TRUE.
	Error	Error	BOOL	FALSE	Set to TRUE when an exception occurs
	ErrorID	Error Code	SMC_ ERROR	0	Outputs an error code when an exception occurs.
	AxisError	Axis error labeling	BOOL	FALSE	The readout axis is an error, and the corresponding marker is set.
	AxisErrorID	Axis Error Code	DWORD	0	Reads the axis as an error code.
	SWEnd SwitchActive	Soft limit switches Effective	BOOL	FALSE	In command reading, check the soft limit switch status.

（3） Function description

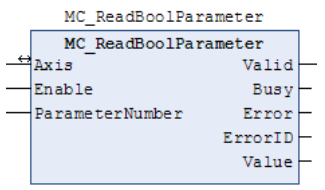
This function block is used to describe general axis errors that are not related to the function block.

4. 1. 13 Read Bit Parameter Command for Axis

MC_ReadBoolParameter

The command reads the bit parameters of the drive axis and saves them in the variable cell defined by itself.

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_ ReadBoolParameter	Read the axis of Bit Parameters		MC_ReadBoolParameter(Axis:= , Enable:= , ParameterNumber:= , Valid=> , Busy=> , Error=> , ErrorID=> , Value=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Comment
Inout	Axis	Shaft	AXIS_ REF_SM 3	Mapped to an axis, an instance of AXIS_REF_SM3.
Input	Enable	Implementation conditions	BOOL	Reads the current position of the servo for the TRUE state.
Input	Parameter Number	Serial number of the axis parameter	DINT	Access the index and subindexes and serial numbers of the axis parameters. Note:ParameterNumber(DINT)= -DWORD_TO_DINT(SHL(USINT_ TO_DOWRD(usiDataLength,24) (length of data in object dictionary) + SHL25)(UINT_TO_DWORD(uiIndex), 8) (index in object dictionary - 16BIT) + usisubIndex (subindex in object dictionary - 8BIT)) usiDataLength: by bytes Fill in; 1 byte for 16#01; 2 bytes for 16#02; 4 bytes for 16#04, etc.
Output	Valid	Location Data Obtainable flags	BOOL	Set to TRUE if the drive position is correctly obtained.
	Busy	Command is being executed	BOOL	The current Command is being executed, set to TRUE.
	Error	Error	BOOL	Set to TRUE when the exception occurs.
	ErrorID	Error Code	SMC_ ERROR	Outputs an error code when an exception occurs.

（3） Function description

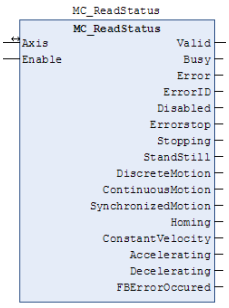
MC_ReadBoolParameter returns the value of the soft motion parameter specified in the input parameter number.

Boolean variable values contain read values.

4. 1. 14 Read axis status command MC_ReadStatus

Read the status data of the axes

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_ReadStatus	Read the state of the axis		MC_ReadStatus(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , Disabled=> , Errorstop=> , Stopping=> , StandStill=> , DiscreteMotion=> , ContinuousMotion=> , SynchronizedMotion=> , Homing=> , ConstantVelocity=> , Accelerating=> , Decelerating=> , FBErrorOccured=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_REF_SM3		Mapped to axis, an instance of AXIS_REF_SM3
Input	Enable	Implementation conditions	BOOL	**FALSE**	Reads the current position of the servo for the TRUE state.
Output	Valid	Wrong data Obtainable flags	BOOL	**FALSE**	Can get the error data of the axis, set to TRUE.
	Busy	Command is being executed	BOOL	**FALSE**	The current Command is being executed, set to TRUE.
	Error	Error	BOOL	**FALSE**	Set to TRUE when an exception occurs
	ErrorID	Error Code	SMC_ERROR	0	Outputs an error code when an exception occurs.
	Disabled	Shaft not Enabled state	BOOL	**FALSE**	The axis in the unenabled state is TRUE.

Scope	Name	Chinese Name	Type	Initial	Comment
	Errorstop	Axis error status	BOOL	**FALSE**	The axis is TRUE in the error run state.
	Stopping	Axis stop Process Status	BOOL	**FALSE**	Axis in the stopping process for TRUE.

	StandStill	Axis standard state	BOOL	**FALSE**	The axis is TRUE in the standard (capable of running) state.
	Discrete Motion	Axis discrete Movement Status	BOOL	**FALSE**	The axis is TRUE in the discrete motion state.
Output	Continuous Motion	Axis Continuous Movement Status	BOOL	**FALSE**	The axis is TRUE in the continuous motion state
	Synchronized Motion	Axis synchronization Operation Status	BOOL	**FALSE**	TRUE for the axis in synchronous motion
	Homing	Axis back to home state	BOOL	**FALSE**	The axis is TRUE in the home state
	Constant Velocity	Axis operation Speed Arrival	BOOL	**FALSE**	The axis reaches TRUE in the operating speed
	Accelerating	Axis acceleration Process Status	BOOL	**FALSE**	Axis acceleration process status is TRUE
	Decelerating	Shaft reduction Process Status	BOOL	**FALSE**	The status of the axis deceleration process is TRUE
	FBError Occured	Axis function block error appearance flag	BOOL	**FALSE**	Axis function block error flag is TRUE

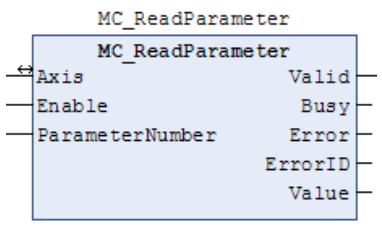
（3） Function description

This function block returns the detailed status of the axes to see what motion is currently in progress.

4. 1. 15 Read parameter command for axis MC_ReadParameter

The command reads the parameters of the drive axis.

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_ ReadParameter	Reading the parameters of the axes		MC_ReadParameter(Axis:= , Enable:= , ParameterNumber:= , Valid=> , Busy=> , Error=> , ErrorID=> , Value=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Comment
Inout	Axis	Shaft	AXIS_REF_SM3	Mapping to an axis, an instance of AXIS_REF_SM3.
Input	Enable	Implementation conditions	BOOL	Reads the current position of the servo for the TRUE state.
	Parameter Number	Serial number of the axis parameter	DINT	Accesses the index and subindex and serial number of the axis parameter. Note :ParameterNumber(DINT)= -DWORD_TO_DINT(SHL(USINT_TO_DOWRD(usiDataLength),24)(degree of data in the object dictionary) + SHL(UINT_TO_ DWORD(uiIndex), 8) (sol-16BIT in object dictionary)+usisubIndex(subindex-8BIT in object dictionary)usiDataLength. Filled by number of bytes; 1 byte for 16#01; 2 bytes for 16#02; 4 bytes for 16#04, etc.
Output	Valid	Location Data Obtainable flags	BOOL	It is possible to correctly obtain the location of the drive and set it to TRUE
	Busy	Command is being executed	BOOL	The current Command is being executed, set to TRUE.
	Error	Error	BOOL	Set to TRUE when the exception occurs.
	ErrorID	Error Code	SMC_ERROR	Outputs an error code when an exception occurs.
	Value	Acquired axis parameters	LREAL	The axis parameters read out by the command.

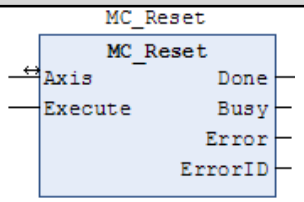
（3） Function description

MC_ReadParameter returns the value of the soft motion parameter specified in the input parameter number. The returned variable value is converted to lreal if necessary.

4. 1. 16 Axis error state reset command MC_Reset

- **Replication Module**

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_Reset	Axis error status Reset command		MC_Reset(Axis:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	**AXIS_ REF_SM3**		mapped to the axis, an instance of AXIS_REF_SM3
Input	Execute	Implementation conditions	**BOOL**		A rising edge of the input will initiate Processing of function blocks.
Output	Done	Command execution completed	**BOOL**		The execution of the axis command is completed. Set to TRUE.
	Busy	Command is being executed	**BOOL**		The current Command is being executed, set to TRUE.
	Error	Error	**BOOL**		Set to TRUE when the exception occurs.
	ErrorID	Error Code	**SMC_ ERROR**	**SMC_NO _ERROR**	Outputs an error code when an exception occurs.

（3） Function description

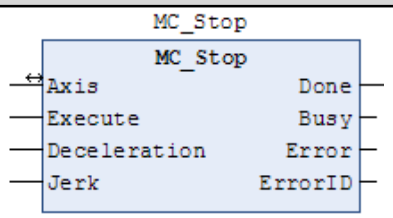
This function block brings the axis from a status error stop to a status stop state by resetting (acknowledging) all errors. Both drive errors and software errors are reset.

The SMC_R_NO_ERROR_TO_RESET error is returned by the function block when called in a state other than error stop.

4. 1. 17 Axis stop command MC_Stop

- **Stop Module**

（1）Command format

Command	Name	Graphical representation	ST Performance
MC_Stop	Axis stop command		MC_Stop(Axis:= , Execute:= , Deceleration:= , Jerk:= , Done=> , Busy=> , Error=> , ErrorID=>);

（2）Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_ REF_SM3		Mapped to the axis, an instance of AXIS_REF_SM3
Input	Execute	Execution Conditions	BOOL	FALSE	A rising edge of the input will start the function The processing of the block.
	Deceleration	Deceleration	LREAL	0	Deceleration of the functional block (u/S^2)
	Jerk	Jumpiness	LREAL	0	Specified jump degree [command unit/S^3]
Output	Done	Command execution completed	BOOL	FALSE	The axis Command is executed and set to TRUE.
	Busy	Command is being executed	BOOL	FALSE	The current Command is being executed, set to TRUE.
	Error	Error	BOOL	FALSE	Set to TRUE when the exception occurs.
	ErrorID	Error Code	SMC_ ERROR	0	Outputs an error code when an exception occurs.

（3）Function description

MC_Stop puts the axis in the stop state. Therefore, the motion of the currently running function block instance is aborted. (The only exception is running an instance of MC_Stop, which has not been aborted. Instead, the MC_Stop instance that just started returns an error.

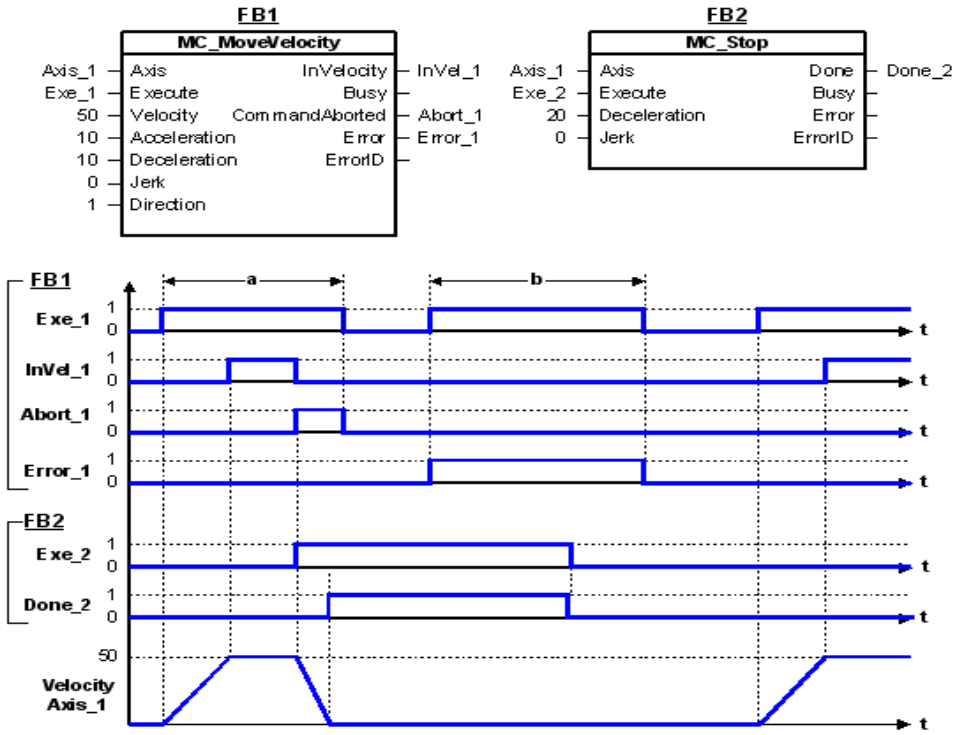
If the axis is stopped, no other instance can execute motion on the axis. If the axis reaches a speed value

of zero, the finished output is set to TRUE. If the execution input is TRUE, the axis remains in the stopped state. If the execution is wrong and the finished output is TRUE, the axis goes to standstill.

Time-series diagram:

Use MC_Stop

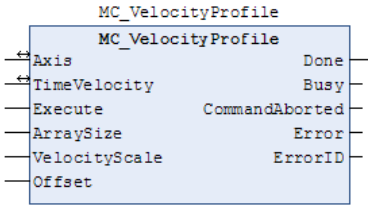
The illustration shows the combination of the FB2 instance of this type of MC_Stop and the FB1 instance of the MC_MoveVelocity type. The rotation axis is tilted down by FB2. The axis does not execute any motion commands as long as FB2. execution is real. fb1 outputs an error message indicating that the FB2 instance is active.



4. 1. 18 Velocity Profile Command MC_VelocityProfile

This function block is designed for command time speed locking motion profiles.

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_VelocityProfile	Speed profile command		MC_VelocityProfile(Axis:= , TimeVelocity:= , Execute:= , ArraySize:= , VelocityScale:= , Offset:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_REF_SM3		Mapped to axis, an instance of AXIS_REF_SM3
	TimeVelocity	Axis speed run time and speed tracing	MC_TV_REF		Axis speed run time and speed data description, consisting of multiple data sets.
Input	Execute	Implementation conditions	BOOL		A rising edge of the input will initiate the processing of the function block.
	ArraySize	Dynamic arrays	INT		The number of arrays used in the run profile.
	VelocityScale	Speed factor	LREAL	1	Scaling factor for speed.
	Offset	Offset	LREAL		Overall offset value of velocity values
Output	Done	Command execution completed	BOOL		The Command is executed and set to TRUE.
	Busy	Command is being executed	BOOL		The current Command is being executed, set to TRUE.
	Command Aborted	Command is interrupted	BOOL		The current Command is interrupted and set to TRUE.
	Error	Error	BOOL		Set to TRUE when the exception occurs.
	ErrorID	Error Code	SMC_ERROR		Outputs an error code when an exception occurs.

（3） Function description

This function block is designed for command time speed locking motion profiles.

Tip: The

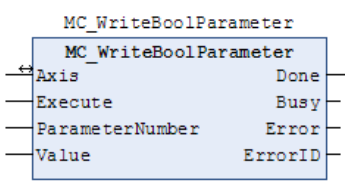
Conversions between different profiles can be managed. Alternatively, in MC_VelocityProfile, a cam function block coupled with a virtual host can be used.

Warning: MC_MoveSuperimposed is not supported except for MC_VelocityProfile.

4. 1. 19 Setting the bit parameter of the axis command MC_WriteBoolParameter

The command sets the bit parameter of the drive axis.

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_WriteBoolParameter	Set the axis of Bit Parameter Command		MC_WriteBoolParameter(Axis:= , Execute:= , ParameterNumber:= , Value:= , Done=> , Busy=> , Error=> , ErrorID=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Comment
Inout	Axis	Shaft	AXIS_REF_SM3	Mapping to an axis, an instance of AXIS_REF_SM3.
Input	Execute	Implementation conditions	BOOL	Drive a setup operation for rising edge operation.
	Parameter Number	Axis parameters The serial number of	DINT	Accesses the index and subindex and ordinal number of the axis parameter. Note: ParameterNumber(DINT) = -DWORD_TO_DINT(SHL(USINT_TO_DOWRD(usiDataLength),24)(length of data in object dictionary) + SHL(UINT_TO_DWORD(uiIndex), 8)(index in object dictionary - 16BIT) + usisubIndex(subindex in object dictionary - 8BIT) usiDataLength. Fill in by the number of bytes; 1 byte for 16#01; 2 bytes for 16#02; 4 bytes for 16#04, etc.
	Value	Set value	BOOL	Sets the bit parameter value.

Scope	Name	Chinese Name	Type	Comment
Output	Done	Settings Successful operation	BOOL	Set the operation success to TRUE.
	Busy	Command Being implemented	BOOL	The current Command is being executed, set to TRUE.
	Error	Error	BOOL	Set to TRUE when the exception occurs.
	ErrorID	Error Code	SMC_ ERROR	Outputs an error code when an exception occurs.

（3） Function description

MC_WriteBoolParameter modifies the value of the soft motion parameter specified in the parameter number. The boolean variable value contains the writing value.

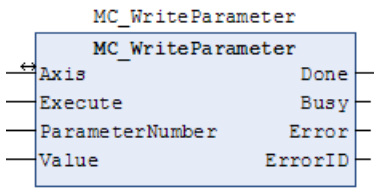
Tip:

EtherCAT Coe and CAN, CANopen code for parameter numbering.

4. 1. 20 Setting the axis parameter MC_WriteParameter

MC_WriteParameter modifies the value of the soft motion parameter specified in the parameter number.

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_ WriteParameter	Setting axis parameters		MC_WriteParameter(Axis:= , Execute:= , ParameterNumber:= , Value:= , Done=> , Busy=> , Error=> , ErrorID=>);

（2）Relevant variables

Scope	Name	Chinese Name	Type	Comment
Inout	Axis	Shaft	AXIS_REF_SM3	Mapped to an axis, an instance of AXIS_REF_SM3.
Input	Execute	Implementation conditions	BOOL	Drive a setup operation for rising edge operation.
	Parameter Number	Serial number of the axis parameter	DINT	Accesses the index and subindex and ordinal number of the axis parameter. Note:ParameterNumber(DINT) = -DWORD_TO_DINT(SHL(USINT_TO_DOWRD(usiDataLength),24)(length of data in object dictionary) + SHL(UINT_TO_DWORD(uiIndex), 8) (index in the object dictionary - 16BIT) + usisubIndex (subsoil in the object dictionary - 8BIT) usiDataLength: filled in by the number of bytes; 1 byte for 16#01; 2 bytes for 16#02; 4 bytes for 16#04, etc.
	Value	Set value	LREAL	Sets the bit parameter value.
Output	Done	Setup operation successful	BOOL	Set the operation success to TRUE.
	Busy	Command is being executed	BOOL	The current Command is being executed, set to TRUE.
	Error	Error	BOOL	Set to TRUE when the exception occurs.
	ErrorID	Error Code	SMC_ERROR	Outputs an error code when an exception occurs.

（3）Function description

MC_WriteParameter modifies the value of the soft motion parameter specified in the parameter number.

Tip:

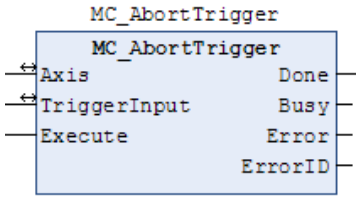
EtherCAT Coe and CAN, CANopen code for parameter numbering.

diParameterNumber := -DWORD_TO_DINT(SHL(TO_DOWRD(usiDataLength), 24)
+ SHL(TO_DWORD(uiIndex), 8)
+ usisubIndex).

4. 1. 21 Function block termination event association MC_AbortTrigger

This function block is designed for the abort function block connected to a trigger event (e.g., MC_TouchProbe).

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_AbortTrigger	Function block termination Event Associated Commands		MC_AbortTrigger(Axis:= , TriggerInput:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_REF_SM3		Mapped to the axis, an instance of AXIS_REF_SM3
	TriggerInput	Trigger signal	TRIGGER_REF		Description of trigger signals, trigger attributes, etc.
Input	Execute		BOOL	**FALSE**	Drive a setup operation for rising edge operation.
Output	Done		BOOL	**FALSE**	Set the operation success to TRUE.
	Busy		BOOL	**FALSE**	The current Command is being executed, set to TRUE.
	Error		BOOL	**FALSE**	Set to TRUE when the exception occurs.
	ErrorID		SMC_ERROR	0	Outputs an error code when an exception occurs.

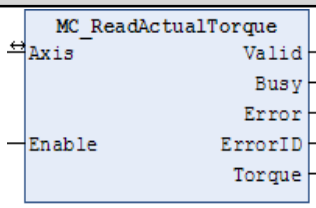
（3） Function description

This function block is designed for the abort function block connected to a trigger event (e.g., MC_TouchProbe).

4. 1. 22 Current Torque Read Command MC_ReadActualTorque

Reads the current torque value of the drive

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_ReadActualTorque	Current torque Value read command		MC_ReadActualTorque0(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , Torque=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS _REF_SM3		Mapped to an axis, an instance of AXIS_REF_SM3.
Input	Enable	Implementation conditions	BOOL	FALSE	Reads the current position of the servo for the TRUE state.
Output	Valid	Current torque value Obtainable flags	BOOL	FALSE	Can get the correct torque value of the driver, set to TRUE.
	Busy	Command is being executed	BOOL	FALSE	The current Command is being executed, set to TRUE.
	Error	Error	BOOL	FALSE	Set to TRUE when the exception occurs.
	ErrorID	Error Code	SMC_ERRO R	0	Outputs an error code when an exception occurs.
	Torque	The current torque value obtained	LREAL	0	The current torque data read by the command.

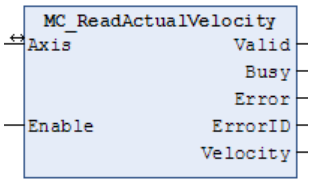
（3） Function description

The function block will return the current torque or force value if hold true is enabled. Once the data output torque is in effect, the validity will be set to "true". If Enable is reset, the data will lose its validity and the validity will be reset whether or not new data is available.

4. 1. 23 Current velocity read command MC_ReadActualVelocity

The command reads the current speed value of the drive operation.

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_ReadActualVelocity	Current speed Read command		MC_ReadActualVelocity0(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , Velocity=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_ REF_SM3		Mapped to an axis, an instance of AXIS_REF_SM3.
Input	Enable	Implementation conditions	BOOL	FALSE	Can correctly get the speed value of the drive, set to TRUE.
Output	Valid	Current speed value Obtainable flags	BOOL	FALSE	Can correctly get the speed value of the drive, set to TRUE.
	Busy	Command is being executed	BOOL	FALSE	The current Command is being executed, set to TRUE.
	Error	Error	BOOL	FALSE	Set to TRUE when the exception occurs.
	ErrorID	Error Code	SMC_ERROR	0	Outputs an error code when an exception occurs.
	Velocity	The current speed value obtained	LREAL	0	The current speed data read out by the command.

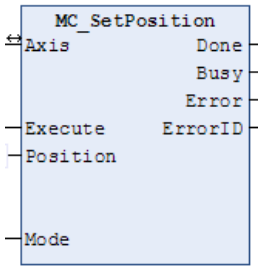
（3） Function description

The function block will return the value of the current velocity as long as the enable remains "true". If enabled is reset, the data will lose its validity, and the validity will be reset regardless of whether new data is available.

4. 1. 24 Parameter Command MC_SetPosition for setting the axis

Set the position data in the command to the position data of the current axis.

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_ SetPosition	Parameter commands for setting axes		MC_SetPosition0(Axis:= , Execute:= , Position:= , Mode:= , Done=> , Busy=> , Error=> , ErrorID=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_ REF_SM3		Mapped to an axis, an instance of AXIS_REF_SM3.
Input	Execute	Implementation conditions	BOOL		Drive a setup operation for rising edge operation.
	Position	Axis position data	LREAL		Location data.
	Mode	Set value	BOOL		Location model. TRUE: Relative position (RELATIVE). FALSE: Absolute position (ABSOLUTE).
Output	Done	Setup operation successful	BOOL		Set the operation success to TRUE.
	Busy	Command is being executed	BOOL	FALSE	The current Command is being executed, set to TRUE.
	Error	Error	BOOL		Set to TRUE when the exception occurs.
	ErrorID	Error Code	SMC_ERROR		Outputs an error code when an exception occurs.

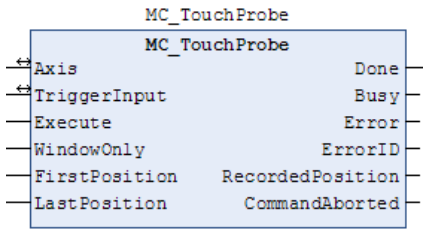
（3） Function description

This function block is used to transfer the coordinate system of the axis. Thus, the function block will manipulate the set position as well as the actual position of the axis with the same values without causing any movement. This recalibration can be used, for example, for the reference case. The function block can be called during the movement without changing the commanded position, which will then be in the moved coordinate system.

4. 1. 25 Enable external locking MC_TouchProbe

The command is triggered by an external signal and saves the position data of the current axis.

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_TouchProbe	Enable external locking		MC_TouchProbe(Axis:= , TriggerInput:= , Execute:= , WindowOnly:= , FirstPosition:= , LastPosition:= , Done=> , Busy=> , Error=> , ErrorID=> , RecordedPosition=> , CommandAborted=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_REF_SM3		Mapped to axis, an instance of AXIS_REF_SM3
	TriggerInput	Trigger signal	TRIGGER_REF		Associated attributes such as trigger signal or trigger attribute.
Input	Execute	Implementation conditions	BOOL	FALSE	Drive one setup operation for rising edge operation.
	WindowOnly	Trigger Window	BOOL	FALSE	TRUE: Trigger events will be accepted only within the specified window (definition hereunder).
	FirstPosition	Trigger start position	LREAL	0	Specifies the start position of the receive trigger.
	LastPosition	Trigger end position	LREAL	0	Specifies the end position of the receive trigger.
Output	Done	Setup operation successful	BOOL	FALSE	Set the operation success to TRUE.

Scope	Name	Chinese Name	Type	Initial	Comment
Output	Busy	Command is being executed	BOOL	FALSE	The current Command is being executed, set to TRUE.
	Error	Error	BOOL	FALSE	Set to TRUE when the exception occurs.
	ErrorID	Error Code	SMC_ ERROR	0	Outputs an error code when an exception occurs.
	Recorded Position	Triggerrecord location	LREAL	0	The current position at the time the trigger occurs.
	Command Aborted	Command is interrupted	BOOL		The current Command is interrupted and set to TRUE.

（3） Function description

This function block is designed to record the axis position when the trigger event is raised.

Note:

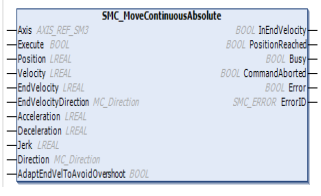
The function block is used for single shot operations, i.e., only the first event followed on the rising edge is executed to apply to the recording. Other events will be ignored.

A function block instance should fully represent a probe command.

4. 1. 26 Axis absolute position continuous control SMC_MoveContinuousAbsolute

This function block performs absolute motion at a given final velocity.

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_MoveContinousAbsolute	Axis absolute position Continuous control commands		SMC_MoveContinuousAbsolute(Axis:= , Execute:= , Position:= , Velocity:= , EndVelocity:= , EndVelocityDirection:= , Acceleration:= , Deceleration:= , Jerk:= , Direction:= , AdaptEndVelToAvoidOvershoot:= , InEndVelocity=> , PositionReached=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_REF_SM3		Mapping to an axis, an instance of AXIS_REF_SM3.
Input	Execute	Implementation conditions	BOOL		A rising edge of the input will initiate the processing of the function block.
	Position	Movement relative position	LREAL		This data is the relative position of the motion.
	Velocity	Running speed	LREAL		The maximum speed at which the axis runs to the target position.
	EndVelocity	End of run speed	LREAL		The speed of operation after the Command execution is completed.
	EndVelocity Direction	End speed direction	MC_Direction	current	You can use: positive,negative, current. Not available: shortest, fastest
	Acceleration	Acceleration	LREAL		Acceleration value when speed becomes large
	Deceleration	Deceleration	LREAL		Deceleration value when speed becomes smaller
Input	Jerk	Jumpiness	LREAL		Value of the jerk
	Direction	Running direction	MC_Direction	shortest	For linear/rectilinear axes: positive, negative; for rotational/circular axes: positive, negative, current, shortest, fastest
	AdaptEnd VelToAvoid Overshoot		BOOL		
Output	InEndVelocity	Command Location Arrival	BOOL		Axis Command execution position reached, set to TRUE
	Position Reached		BOOL		The current Command is being executed, set to TRUE

Scope	Name	Chinese Name	Type	Initial	Comment
	Busy	The command is being Execution	BOOL		The current Command is interrupted and set to TRUE.
	Command Aborted	Command Interrupted	BOOL		Set to TRUE when the exception occurs.
	Error	Error	BOOL		Outputs an error code when an exception occurs.
	ErrorID	Error Code	SMC_ ERROR		The axis Command execution position is reached and set to TRUE.

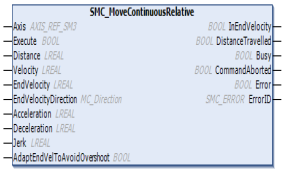
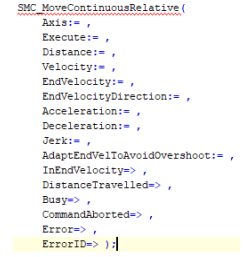
（3） Function description

This function block performs absolute motion at the given final velocity. In contrast to MC_MoveAbsolute, it does not reach the target position with zero velocity, but with the specified EndVelocity. It is designed to be generated by motion. After reaching the target position, the function block sets the "InEndVelocity" output and maintains this velocity until interrupted by another motion.

4. 1. 27 Axis Relative Positioning Command SMC_MoveContinuousRelative

This function block performs relative motion at a given final velocity.

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_ MoveContinuousRelative	Axis Relative Positioning commands		

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_ REF_SM3		Mapping to an axis, an instance of AXIS_REF_SM3.
Input	Execute	Implementation	BOOL		A rising edge of the input

		conditions			will initiate the processing of the function block.
	Distance	Movement relative position	LREAL		This data is the relative position of the motion.
	Velocity	Running speed	LREAL		The maximum speed at which the axis runs to the target position.
	EndVelocity	End of run speed	LREAL		The speed of operation after the Command execution is completed.
	EndVelocity Direction	End speed direction	MC _Direction	current	You can use positive, negative, current. Not available: shortest, fastest
	Acceleration	Acceleration	LREAL		The acceleration value when the speed becomes large.
	Deceleration	Deceleration	LREAL		The deceleration value when the speed becomes smaller.
	Jerk	Jumpiness	LREAL		Specify the leap [command unit /S^3].
	AdaptEnd VelToAvoid Overshoot		BOOL		

Scope	Name	Chinese Name	Type	Initial	Comment
Output	InEndVelocity		BOOL		The axis Command execution position is reached and set to TRUE.
	Distance Travelled		BOOL		TRUE: Commanded distance has been travelled. Axis runs with commanded velocity “EndVelocity” (or possibly slower if Adapt End Vel ToAvoid Overshoot has been set).

	Busy		BOOL		The current Command is being executed, set to TRUE.
	Command Aborted		BOOL		The current Command is interrupted and set to TRUE.
	Error		BOOL		Set to TRUE when the exception occurs.
	ErrorID		SMC _ERROR		Outputs an error code when an exception occurs.

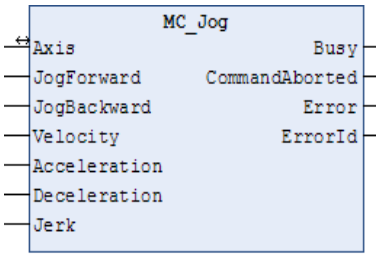
（3） Function description

This function block performs relative motion at the given final velocity. In contrast to MC_MoveRelative, it does not arrive at the target position with zero velocity, but with the specified EndVelocity. It is designed to be generated by motion. After reaching the target position, the function block sets the "InEndVelocity" output and maintains this velocity until interrupted by another motion.

4. 1. 28 Axis point command MC_Jog

- **Pointing module**

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_Jog	Axis point command		MC_Jog(Axis:= , JogForward:= , JogBackward:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Busy=> , CommandAborted=> , Error=> , ErrorId=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_ REF_SM 3		Mapping to an axis, an instance of AXIS_REF_SM3.

Input	JogForward	Positive validity	BOOL		Set to TRUE to start forward movement. Set to FALSE to stop forward movement.
	JogBackward	Negatively effective	BOOL		set to TRUE to start the reverse movement. Set to FALSE to stop the reverse movement.
	Velocity	Target speed	LREAL		Specify the target speed. Unit: [command unit /s]
	Acceleration	Acceleration	LREAL		Specifies the acceleration. Unit: [command unit /s]
	Deceleration	Deceleration	LREAL		Specify the deceleration rate. Unit: [command unit/s]
	Jerk	Jumpiness	LREAL		Specify the leap [command unit /S^3].

Scope	Name	Chinese Name	Type	Initial	Comment
Output	Busy	Under implementation	BOOL	FALSE	After receiving the command, it is set to TRUE.
	Command Aborted	Execution interruptions	BOOL	FALSE	Set to TRUE when the command is aborted.
	Error	Error	BOOL		Set to TRUE when the exception occurs.
	ErrorId	Error Code	SMC_Error		Outputs an error code when an exception occurs.

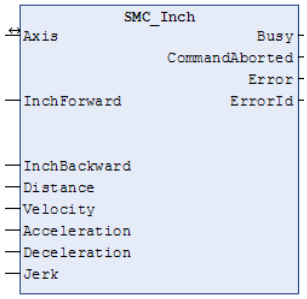
（3） Function description

MC_Jog causes continuous motion on the axis.

4. 1. 29 Axis inching command SMC_Inch

- **Inching module**

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC_Inch	Axis inching command		SMC_Inch0(Axis:= , InchForward:= , InchBackward:= , Distance:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Busy=> , CommandAborted=> , Error=> , ErrorId=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_REF_SM3		Mapping to an axis, an instance of AXIS_REF_SM3.
Input	InchForward	Positive Execution	BOOL		If InchForward is TRUE, the The axis will move at the given velocity (Velocity, Acceleration, Deceleration) in a positive direction until the distance is reached. The input must be specified as FALSE and then TRUE to start the motion again. If InchForward is set to FALSE before reaching the position, then the axis will immediately decelerate to 0 and Busy will be set to FALSE.
	InchBackward	Reverse Execution	BOOL		If InchBackward is TRUE, the axis will move according to the given velocity value (Velocity, Acceleration. Deceleration) to move forward to the set position. The input must then be set to FALSE and then to TRUE to start another motion. If the input signal InchForward is set at the same time.

	Distance	Distance traveled	LREAL		This data is the distance of movement.
	Velocity	Running speed	LREAL		The maximum speed at which the axis runs to the target position.
	Acceleration	Acceleration	LREAL		The acceleration value when the speed becomes large.
	Deceleration	Deceleration	LREAL		The deceleration value when the speed becomes smaller.
	Jerk	Jumpiness	LREAL		Specify the leap [command unit /S^3].
Output	Busy	Command is being executed	BOOL	**FALSE**	The current Command is being executed, set to TRUE.
	Command Aborted	Command is interrupted	BOOL	**FALSE**	The current Command is interrupted and set to TRUE.
	Error	Error	BOOL		Set to TRUE when the exception occurs.
	ErrorId	Error Code	SMC_ ERROR		Outputs an error code when an exception occurs.

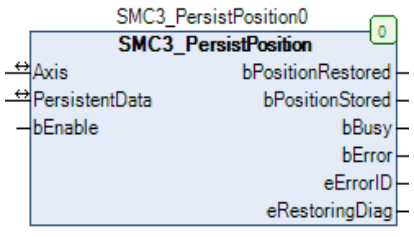
（3） Function description

The function block produces a gradual movement on the axis and is executed progressively.

4. 1. 30 Axis position hold SMC3_PersistPosition

This function block is used to hold the axis position of a multi-turn absolute encoder with a real axis.

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC3_PersistPosition	Axis Position Hold		SMC3_PersistPosition0(Axis:= , PersistentData:= , bEnable:= , bPositionRestored=> , bPositionStored=> , bBusy=> , bError=> , eErrorID=> , eRestoringDiag=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_ REF_SM3		Mapping to an axis, an instance of AXIS_REF_SM3 .

	PersistentData	Maintain Data	SMC3_Persist Position_Data		Power-off hold data structure for storing position information.
Input	bEnable	Execution	BOOL	**FALSE**	True The function block is executed, false does not execute the function block if the last stored position is to be restored during initialization.

Scope	Name	Chinese Name	Type	Initial	Comment
Output	bPosition Restored	Location Recovery	BOOL		TRUE, Position recovery is complete after axis restart.
	bPosition Stored	Location Save	BOOL		TRUE,Save position completion after calling function block.
	bBusy	FB Under implementation	BOOL		TRUE, the execution of the function block is not completed.
	bError	Error	BOOL	FALSE	TRUE,Exception occurred.
	eErrorID	Error Code	SMC _ERRO R	SMC_ NO_ERROR	Output error code when exception occurs
	eRestoring Diag	Recovery Diagnosis	SMC3 _Persist Position Diag	SMC3_ Persist PositionDiag. SMC3_ PPD_ RESTORING _OK	Diagnostic letter SMC3_ in position recovery PPD_RESTORING_OK: Position successfully recovered SMC3_PPD_AXIS_PRO P_ CHANGED: The axis parameters have been changed and the position cannot be recovered SMC3_PPD_DATA_STO RED_DURING_WRITIN

					G: Function block copies data from axis parameter data structure instead of PersistentData data. Possible causes: Non-synchronous persistent variables, controller crash dead.

（3） Function description

This function block is used to hold the axis position of a multi-turn absolute encoder with a real axis.

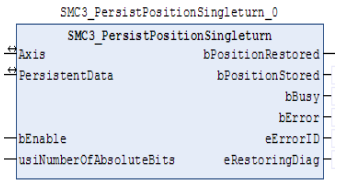
This function block assumes a multi-turn encoder travel of 2^32.

For multi-turn encoders with smaller ranges, use SMC3_PersistPositionSingleturn.

4. 1. 31 Axis Position Hold SMC3_PersistPositionSingleturn

This function block is used to hold the axis position of an absolute encoder with a limited range of real axes.

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC3_ PersistPositionSingleturn	Axis position Maintain		SMC3_PersistPositionSingleturn_0(Axis:= , PersistentData:= , bEnable:= , usiNumberOfAbsoluteBits:= , bPositionRestored=> , bPositionStored=> , bBusy=> , bError=> , eErrorID=> , eRestoringDiag=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_ REF_SM3		Mapping to an axis, an instance of AXIS_REF_SM3.
	Persistent Data	Maintain data	SMC3_ Persist PositionSing leturn_Data		A power-off hold data structure for storing position information.
Input	bEnable	Execution	BOOL	FALSE	True - the function block is executed.

					False - not executed. The value must be set to true from application startup if the last stored location is to be restored during initialization.
	usiNumberOf AbsoluteBits	Digits	USINT	16	How many bits absolute encoders (e.g. 20-bit, 24-bit encoders, etc.)

Scope	Name	Chinese Name	Type	Initial	Comment
Output	bPosition Restored	Position Recovery	BOOL		TRUE, after axis restart Location recovery is complete.
	bPosition Stored	Location saving	BOOL		TRUE,Call function Quickly after the save position is completed.
Output	bBusy	FB execution in progress	BOOL		TRUE, the execution of the function block is not completed.
	bError	Error	BOOL	FALSE	TRUE,Exception occurred.
	eErrorID	Error Code	SMC _ERROR	SMC_NO _ERROR	When the abnormality occurs, the Outputs an error code.
	eRestoring Diag	Recovery Diagnosis	SMC3_ Persist Position Diag	SMC3_ Persist Position Diag. SMC3_ PPD_ RESTO RING_OK	Diagnostic message in location recovery SMC3_PPD_RESTORING_ OK: Location successfully recovered SMC3_PPD_AXIS_PROP_ CHANGED. The axis parameters have been changed and the position cannot be restored SMC3_PPD_DATA_STORE D_ DURING_ WRITING:The function block copies data from the axis parameter data structure instead of PersistentData data.

					Possible causes: Non Synchronous continuous variability, controller crash Crash dead.

（3） Function description

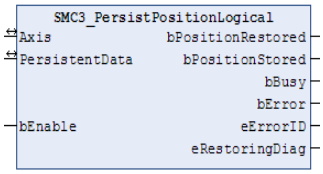
This function block is used to maintain the axis position of absolute encoders with a limited range of true axes. Strictly speaking, this function block is used not only for single-turn encoders, but also for multi-turn encoders with an encoder range of 2 and less than 2^32.

Range of two encoders - 2^k . . .2^k - 1 and the range 0. . 2^(k+1) are supported.

4. 1. 32 Logical axis position hold command SMC3_PersistPositionLogical

This function block is used to maintain the axis position of the logical axis.

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC3_ PersistPosition Logical	Logic axis position Hold command		SMC3_PersistPositionLogical0(Axis:= , PersistentData:= bEnable:= , bPositionRestored=> , bPositionStored=> , bBusy=> , bError=> , eErrorID=> , eRestoringDiag=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_REF_ LOGICAL_SM 3		Mapped to axis, an instance of AXIS_REF_SM3
	Persistent Data	Maintain Data	SMC3_ PersistPosition Logical_Data		A power-off hold data structure for storing position information.
Input	bEnable	Execution	BOOL	**FALSE**	True function blocks are executed, false does not execute function blocks to restore the last stored location during initialization, the value must be set to true from application startup.

Scope	Name	Chinese Name	Type	Initial	Comment
Output	bPosition Restored	Location Recovery	BOOL		TRUE,The position is restored after the axis is restarted.
	bPosition Stored	Location Save	BOOL		TRUE, Position recovery is complete after axis restart.
	bBusy	FB Under implementation	BOOL		TRUE, the execution of the function block is not completed.
	bError	Error	BOOL	FALSE	TRUE, Exception occurred.
	eRestoring Diag	Recovery Diagnosis	SMC3_ Persist Position Diag	SMC3_PersistPositionDiag. SMC3_PPD _ RESTORING _OK	Diagnostic information in position recovery SMC3_ PPD_RESTORING_OK:P osition successfully restored SMC3_PPD _AXIS_PROP_ CHANGED:The axis parameter has been changed. Unable to recover location SMC3_PPD_D ATA_STORED_ DURING_ WRITING:Function block copies data from the axis parameter data structure instead of from Persistent Replication in Data data. Possible cause: Non-synchronous persistence Variables, controllers Crash and die.

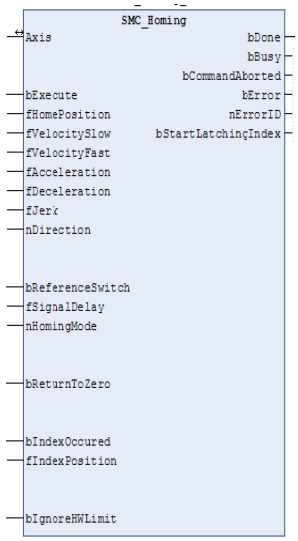
（3） Function description

This function block is used to maintain the axis position of the logical axis.

4. 1. 33 Axis return to zero command SMC_Homing

- Axis return to zero command

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC_Homing	Axis return to zero command		SMC_Homing0(Axis:= , bExecute:= , fHomePosition:= , fVelocitySlow:= , fVelocityFast:= , fAcceleration:= , fDeceleration:= , fJerk:= , nDirection:= , bReferenceSwitch:= , fSignalDelay:= , nHomingMode:= , bReturnToZero:= , bIndexOccured:= , fIndexPosition:= , bIgnoreHWLimit:= , bDone=> , bBusy=> , bCommandAborted=> , bError=> , nErrorID=> , bStartLatchingIndex=>);

（2）Variable description

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_REF_SM3		mapped to the axis, an instance of AXIS_REF_SM3.
Input	bExecute	Execution	BOOL	FALSE	True function blocks are executed, false function blocks are not executed
	fHome Position	Origin setting Location	LREAL	0	The home set position after returning to zero, and the unit is the user calibrated unit.
	fVelocity Slow	Slow	LREAL	0	Slow setting speed after leaving the reference switch.
	fVelocity Fast	Fast	LREAL	0	Quick set speed when leaving the reference switch set.
	fAcceleration	Acceleration	LREAL	0	Acceleration setting value.
	fDeceleration	Deceleration	LREAL	0	Deceleration setting value.
	fJerk	Acceleration derivative	LREAL	0	Jerk in [u/s3]

Scope	Name	Chinese Name	Type	Initial	Comment
Input	nDirection	Back to zero direction	MC_ Direction	negative	Back to zero start direction, refer to MC_DIRECTION.
	bReferenceSwitch	Reference Switch	BOOL	TRUE	Connect the reference switch. TRUE:Refer to the open switch trigger. FALSE:Reference switch is closed.
	fSignal Delay	Delay	LREAL	0	The transmission time of the reference switch to compensate for the dead time. The unit is seconds.
	nHoming Mode	Back to zero mode	SMC_ HOMING _MODE	FAST_BSLOW _S_STOP	Reference SMC_ HOMING_MODE.
	bReturn ToZero	Return to zero position	BOOL	FALSE	TRUE: the axis runs to position zero after the return to zero is completed (note: if fHomePosition=10, the axis position becomes 10 after the return to zero is completed, and bReturnTozero is ture then the axis goes 10 units backwards to position zero after the return to zero is completed)

Scope	Name	Chinese Name	Type	Initial	Comment
	bIndex Occured		BOOL	FALSE	True, flag pulse recording, zero return mode is FAST_ BSLOW_I_S_ STOP, FAST_SLOW Effective when _I_S_STOP.
	fIndex Position		LREAL	0	The position recorded at the time of the flag pulse.
	bIgnore HWLimit	Ignore hard limits	BOOL	FALSE	TRUE, set hardware limit switch enable to false, if the same physical switch is used for hardware limit switch and reference switch, then hardware control will be set to false.
Output	bDone		BOOL	FALSE	True, return to zero is complete.
	bBusy		BOOL		True, the function block is in effect.
	bCommandAborted		BOOL	FALSE	True,The function block is interrupted by other action Commands.
	bError		BOOL	FALSE	True, the error occurred.
	nErrorID		SMC _ERROR	0	Error code, enumerated type Variables, see help smc_ error to see the specific alarm code.
	bStart LatchingIndex		BOOL		by "bIndexOccured" and "fIndexPosition" work together to produce.

（3） Function description

Several return-to-zero modes:

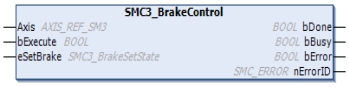
Name	Initial	Comment
FAST_BSLOW_S_STOP	0	Reverses and moves from the reference switch at a slow speed. Execution position: Stop.
FAST_BSLOW_STOP_S	1	Reverses and moves from the reference switch at a slow speed. Stop; execution position.
FAST_BSLOW_I_S_STOP	2	Reverses and moves from the reference switch at a slow speed. Waiting for the index pulse; execution position: stop.
FAST_SLOW_S_STOP	4	Move from the reference switch at a slow speed. Execution position: Stop.
FAST_SLOW_STOP_S	5	Move from the reference switch at a slow speed. Stop; execution position.
FAST_SLOW_I_S_STOP	6	Move from reference switch at slow speed; wait for index pulse; execution position: stop

Defines the lifting order. Used in function block SMC_Homing.

4. 1. 34 Brake switch command SMC3_BrakeControl

The brake can be turned off or on, or it can be switched to automatic mode.

（1） Command description

Command	Name	Graphical representation	ST Performance
SMC3_ BrakeControl	Brakes The switch command of		SMC3_BrakeControl (Axis:=. bExecute:=. eSetBrate:=. bDone=>. bBusy=>. bError=>. nErrorID=>).

（2）Variable description

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_ REF_SM3		
Input	bExecute	Trigger	BOOL	FALSE	TRUE: Activates FB, else inactive.
	eSetBrake	Setting the brake status	SMC3_ BrakeSetState	SMC_ BRAKE _AUTO	Sets the brake state. The state of the placement brake.
	fTimeOut	Set timeout time	LREAL		If the time since enabling the FB is larger than this value, bTimeOut is set. This mechanism is deactivated, if fTimeOut = 0.
Output	bDone	Completion	BOOL		TRUE: Signal has been sent to the drive
	bBusy	Busy	BOOL		TRUE: FB is not idle
	bError	Error	BOOL	FALSE	TRUE: Error has occured within the function block.
	nErrorID	Error ID	SMC_ ERROR	0	Error identification

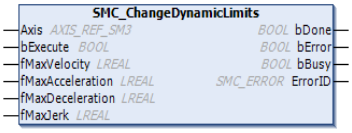
(3) Function description

If the drive and its SoftMotion drive support this function block, it determines the behavior of the mechanical brake. By using this function block in an application, the brake can be turned off or on, or it can be switched to automatic mode, where opening and closing are handled by the drive itself. This function can be used if the automatic mode cannot be used due to the special requirements of the application where the brake must be controlled manually (i.e., manual movement of the components). The function block does not assume any further monitoring or other intelligent functions (i.e., delayed opening of the brake).

4. 1. 35 Setting dynamic limits of axes command SMC_ChangeDynamicLimits

Sets the dynamic limits of the axis (velocity, acceleration, acceleration). Callable only when the axis is powered off or at standstill.

（1）Command description

Command	Name	Graphical representation	ST Performance
SMC_ ChangeDynamicLimits	Set the axis of motion State limit command		SMC3_ChangeDynamicLimits (Axis:=. bExecute:=. fMaxVelocity:=. fMaxAcceleration:=. fMaxDeceleratio:=. fMaxJerk:=. bDone=>. bError=>. bBusy=>. ErrorID=>).

（2） Variable description

Scope	Name	Chinese Name	Type	Comment
Inout	Axis	Shaft	AXIS_ REF_SM3	
Input	bExecute	Trigger	BOOL	Rising edge: Changes the limits.
	fMaxVelocity	Maximum speed	LREAL	The maximum velocity in [u/s]. Must be positive.
	fMaxAcceleration	Maximum acceleration	LREAL	The maximum acceleration in [u/s Australia. Must be positive.
	fMax Deceleration	Maximum deceleration	LREAL	The maximum deceleration in [u/s Australia. Must be positive.
	fMaxJerk	Maximum Leap	LREAL	The maximum jerk in [u/s ¼. Must be positive.
Output	bDone	Completion	BOOL	TRUE: New limits have has been set.
	bError	Error	BOOL	TRUE: An error has occurred.
	bBusy	Busy	BOOL	TRUE: The function block is in operation.
	ErrorID	Error ID	SMC_ ERROR	Error identification

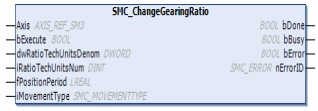
（3） Function description

Sets the dynamic limits of the axis (velocity, acceleration, acceleration). Callable only when the axis is powered off or at standstill.

4. 1. 36 Modify Ratio and Drive Type Command SMC_ChangeGearingRatio

With the help of this function block, the transmission ratio and transmission type can be modified.

（1） Command description

Command	Name	Graphical representation	ST Performance
SMC_ Change GearingRatio	Modify the transmission ratio and Transmission type command		SMC3_ChangeGearingRatio (Axis:=.bExecute:=. dwRatioTechUnitsDenom:=. iRatioTechUnitsNum:=. fPositionPeriod:=. iMovementType:=. bDone=>. bError=>. bBusy=>. nErrorID=>).

（2） Variable description

Scope	Name	Chinese Name	Type	Comment
Inout	Axis	Shaft	AXIS_REF_SM3	Axis
Input	bExecute	Trigger	BOOL	Rising edge: Starts the execution of the function block.
	dwRatioTech UnitsDenom	Conversion factor denominator	DWORD	Denominator of the conversion factor to convert drive increments into technical units.
	iRatioTech UnitsNum		DINT	Numerator of the conversion factor to convert drive increments into technical units.
	fPosition Period		LREAL	Position period, modulo value (only for modulo axes)

Scope	Name	Chinese Name	Type	Comment
	iMovement Type		SMC_ MOVEMENTTY PE	0: Modulo axis. 1: Finite axis
Output	bDone		BOOL	TRUE: Homing is done.
	bBusy		BOOL	TRUE: Execution of function block has not been finished yet.
	bError		BOOL	TRUE: Error has occurred within the function block.
	nErrorID		SMC_ERROR	Error identification

（3） Function description

With the help of this function block, the transmission ratio and transmission type can be modified.

After execution, the axis must be restarted via SMC3_ReinitDrive to ensure that all variables are initialized in the correct way.

4. 1. 37 Setting the gradient of the axis setting to a new value Command SMC_SetCustomRampType

Sets the axis ramp type to a new value. Unlike SMC_SetRampType, a user defined RampType (implementing SMC_TG_iramType) is possible. Usually, iramType1 is equal to iramType2. It is only available when the axis is at rest or powered off. FB is executed synchronously on the rising edge of bExecute. After the call, bDone or bError will be set.

（1） Command description

Command	Name	Graphical representation	ST Performance
SMC_ SetCustom RampType	Set the axis to the Gradient settings for the new value command		SMC_SetCustomRampType (Axis:=. bExecute:=. iRampType1:=. iRampType2:=. bDone=>. bError=>. eErrorID=>).

（2） Variable description

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_ REF_SM3		
Input	bExecute	Trigger	BOOL	FALSE	
	iRampType1	Ramp Type 1	SMC_TG_ IRAMPTYPE		The new ramp type for the first ramp of the trajectories, usually the acceleration-phase.
	iRampType2	Ramp Type 2	SMC_TG_ IRAMPTYPE		The new ramp type for the second ramp of the trajectories, usually the deceleration phase.
Output	bDone	Completion	BOOL		
	bError	Error	BOOL		
	eErrorID	Error ID	**SMC_ERROR**	**SMC_NO_ ERROR**	

（3） Function description

Sets the axis ramp type to a new value. Unlike SMC_SetRampType, a user defined RampType (implementing SMC_TG_iramType) is possible. Usually, iramType1 is equal to iramType2. It is only available when the axis is at rest or powered off. FB is executed synchronously on the rising edge of bExecute. After the call, bDone or bError will be set.

4. 1. 38 Set the virtual axis movement type to linear or modal command SMC_SetMovementType

Sets the movement type of the virtual axis to linear or modal. Only available when the axis is at rest or powered off. FB is executed synchronously on the rising edge of bExecute. After the call, bDone or bError will be set.

（1） Command description

Command	Name	Graphical representation	ST Performance
SMC_ SetMovementType	Set the movement type of the virtual axis to linear or modal command		SMC_SetMovementType (Axis:=. bExecute:=. iMovementType:=. fPositionPeriod:=. bDone=>. bError=>. eErrorID=>).

（2） Variable description

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_ REF_SM 3		Axis
Input	bExecute	Trigger	BOOL	FALSE	
	iMovement Type	Linear Type	INT	0	The new drive type, 0 for modulo, 1 for linear
	fPosition Period	Positioning cycle	LREAL	1	The new position period for modulo movement, must be positive

Output	bDone	Completion	BOOL	FALSE	TRUE: Success
	bError	Error	BOOL	FALSE	TRUE: Error has occured within function block.
	eErrorID	Error ID	SMC_ERROR	SMC_NO_ERROR	Error identification

（3） Function description

Sets the axis ramp type to a new value. Unlike SMC_SetRampType, a user defined RampType (implementing SMC_TG_iramType) is possible. Usually, iramType1 is equal to iramType2. It is only available when the axis is at rest or powered off. FB is executed synchronously on the rising edge of bExecute. After the call, bDone or bError will be set.

4. 1. 39 Setting the axis fade type to a new value Command SMC_SetRampType

Sets the fade type of the axis to a new value. Only available when the axis is at rest or powered off. FB is executed synchronously on the rising edge of bExecute. After the call, bDone or bError will be set.

（1） Command description

Command	Name	Graphical representation	ST Performance
SMC_SetRampType	Set the axis gradient type to the new value command		SMC_SetRampType (Axis:=. bExecute:=. eRampType:=. bDone=>. bError=>. eErrorID=>).

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_REF_SM3		Axis
Input	bExecute	Trigger	BOOL	FALSE	
	eRampType	Ramp Type	SMC_RAMPTYPE	trapez	Defines the new ramp type.
Input	bDone	Completion	BOOL	FALSE	TRUE: Success
	bError	Error	BOOL	FALSE	TRUE: Error has occured within function block.
	eErrorID	Error ID	SMC_ERROR	SMC_NO_ERROR	Error identification

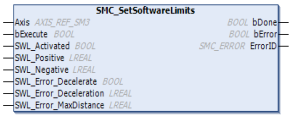
（3） Function description

Sets the fade type of the axis to a new value. Only available when the axis is at rest or powered off. FB is executed synchronously on the rising edge of bExecute. After the call, bDone or bError will be set.

4. 1. 40 Setting the axis position limits and the corresponding error response command SMC_SetSoftwareLimit

Set the position limits of the axes and the corresponding error response. If axes are used for coordinated motion, the corresponding axis group must be re-enabled to consider the new limits for coordinated motion calculations.

（1） Command description

Command	Name	Graphical representation	ST Performance
SMC_ SetSoftwareLimit	Set axis position limits and corresponding error responses		SMC_SetSoftwareLimit (Axis:=.bExecute:=. SWL_Activated:=. SWL_Positive:=. SWL_Negative:=. SWL_Error_Decelerate:=. SWL_Error_Deceleration:=. SWL_Error_MaxDistance=. bDone=>. bError=>. ErrorID=>).

（2） Relevant variables

Scope	Name	Chinese Name	Type	Comment
Inout	Axis	Shaft	AXIS_ REF_SM 3	Axis to change
Input	bExecute	Trigger	BOOL	Rising edge: Rising edge: Starts the execution of the function block. Applies the Changes.
	SWL_Activate d	Activation Limits	BOOL	TRUE: Activates the position limits.
	SWL_Positive	Positive Switch	LREAL	The positive limit switch ([u])
	SWL_Negative	Negative Switch	LREAL	The negative limit switch ([u])

Scope	Name	Chinese Name	Type	Comment
Input	SWL_ Error_ Decelerate		BOOL	Deprecated, the axis decelerates always if a deceleration is configured (`SWL_Error_Deceleration, SWL_Error_MaxDistance) or in The maximum deceleration (see SMC_ ChangeDynamicLimits . fMaxDeceleration or drive configuration).
Input	SWL_ Error_ Deceleration	Deceleration	LREAL	The (positive) deceleration in [u/s]
	SWL_ Error_ MaxDistance	Maximum deceleration distance	LREAL	The maximum braking distance in [u], only applied if positive.
Output	bDone	Completion	BOOL	TRUE: Success
	bError	Error	BOOL	TRUE: Error has occured within the function block.
	ErrorID	Error ID	SMC_ ERROR	Error identification.

（3） Function description

Set the position limits of the axes and the corresponding error response. If axes are used for coordinated motion, the corresponding axis group must be re-enabled to take into account the new limits for coordinated motion calculations.

Note:

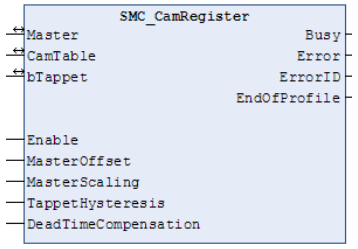
The effective error deceleration is the maximum of the values derived from the maximum deceleration of the axis configuration, the SWL_Error_Deceleration value, the SWL_Error_MaxDistance and the current velocity (SWL_Error_MaxDistance is considered only if the value is positive.)

4. 2 Axis group commands (master/slave commands)

4. 2. 1 Cam tappet control command SMC_CamRegister

This function block represents a tappet control unit that works in negating raw path information and reading only tappet information MC_CAM_REF (e.g., MC_CamIn)

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC_CamRegister	Cam tappet control		SMC_CamRegister0(Master:= , CamTable:= , bTappet:= , Enable:= , MasterOffset:=0 , MasterScaling:= 1, TappetHysteresis:= , DeadTimeCompensation:= , Busy=> , Error=> , ErrorID=> , EndOfProfile=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Master	Main shaft	AXIS_ REF_SM3		Mapped to axis, an instance of AXIS_REF_SM3
	CamTable	Cam table	MC_ CAM_REF		Mapping to an e-cam, an e-cam instance.
	bTappet	Tappet output	ARRAY[1.. MAX_NUM _ TAPPETS] OF BOOL		Output of the tappet point.
Input	Enable	Execution	BOOL	**FALSE**	True function blocks are executed, false does not execute function blocks.
	MasterOffset	Spindle offset	LREAL	0	Spindle offset.
	MasterScaling	Spindle scale	LREAL	1	Linear scaling factor of the principal axis.
	Tappet Hysteresis	Tappet damping	LREAL	0	Tappet control damping factor.

Scope	Name	Chinese Name	Type	Initial	Comment
Input	DeadTime Compensation	Dead time compensation	LREAL	0	Deadband compensation time in S, linearly compensating the tappet output according to the eed of thcurrent spe spindle, can be positive or negative.
Output	Busy	Under implementation	BOOL	**FALSE**	TRUE, function block execution in progress
	Error	Error	BOOL	**FALSE**	TRUE, exception occurred
	ErrorID	Error Code	SMC_ ERROR	**0**	Output error code when exception occurs
	EndOfProfile	Curve cycle completion	BOOL	**FALSE**	True, the spindle position is greater than or equal to the set period

（3） Function description

This function block represents a tappet control unit that works in negating raw path information and reading only tappet information MC_CAM_REF (e.g. MC_CamIn)

4. 2. 2 Get cam slave position command SMC_GetCamSlaveSetPosition

Reads cam gauge slave position, speed, and acceleration information.

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC_ GetCam SlaveSetPosition	Get Cams From axis position	SMC_GetCamSlaveSetPosition Master fStartPosition Slave fStartVelocity fStartAcceleration Busy Enable Error MasterOffset ErrorID SlaveOffset MasterScaling SlaveScaling CamTableID	SMC_GetCamSlaveSetPosition0(Master:= , Slave:= , Enable:= , MasterOffset:= , SlaveOffset:= , MasterScaling:= , SlaveScaling:= , CamTableID:= fStartPosition=> , fStartVelocity=> , fStartAcceleration=> , Busy=> , Error=> , ErrorID=>);

（2）Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Master	Main shaft	**AXIS_ REF_SM3**		Mapping to a spindle
	Slave	From the shaft	**AXIS_ REF_SM3**		Mapping to a slave axis
Input	Enable	Execution	BOOL		True function blocks are executed, false does not execute function blocks.
	MasterOffset	Spindle offset	LREAL		Cam table spindle offset.
	SlaveOffset	Offset from axis	LREAL		Cam table offset from the shaft.
	MasterScaling	Spindle scaling	LREAL	1	Cam table spindle scaling factor.
	SlaveScaling	Zoom from axis	LREAL	1	Cam table from the shaft axis scaling factor.
	CamTableID	Cam ID	MC_ CAM_ID		Cam table ID.
Output	fStart Position	From axis position	LREAL		The slave position obtained from the cam table and the current spindle information.
	fStart Velocity	Slave axis speed	LREAL		The slave speed obtained from the cam table and the current spindle information.
	fStart Acceleration	Slave acceleration	LREAL		The slave acceleration obtained from the cam table and the current spindle information.
	Busy	Under implementation	BOOL		TRUE, means the function block is being executed.
	Error	Error	BOOL		TRUE, Exception occurred.
	ErrorID	Error Code	SMC_ ERROR		Outputs an error code when an exception occurs.

（3）Function description

This function block calculates the current target position of the axis (slave) in case the axis is coupled to

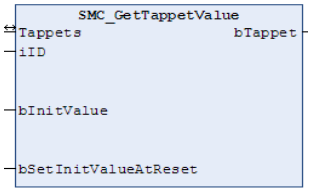
the motion of another axis (spindle) by means of a cam. The slave station does not move or is otherwise affected. This block can be used to determine the starting position, velocity and acceleration of a slave station coupled to a spindle via a cam table.

Since the corresponding values are calculated in one cycle, there is no need to complete the output.

4. 2. 3 Get tappet output value command SMC_GetTappetValue

This function block evaluates the output Tappets of the function block MC_CamIn and contains the current tappet status.

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC_GetTappetValue	Get Tappet Output value		SMC_GetTappetValue0(Tappets:= , iID:=, bInitValue:= , bSetInitValueAtReset:= , bTappet=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Comment
Inout	Tappets		SMC_TappetData	Mapping to a tappet.
Input	iID	Tappet set number	INT	Group ID of the tappet.
	bInitValue	Initial Value	BOOL	The tappet initialization value at the first call of the function block.
	bSetInitValue AtReset		BOOL	TRUE,MC_CamIn The tappet output value will be initialized to the bInitValue setting when the function block is restarted FALSE,the tappet output value will be kept when MC When the _CamIn function block is restarted.
Output	bTappet	Tappet output	BOOL	Tappet value.

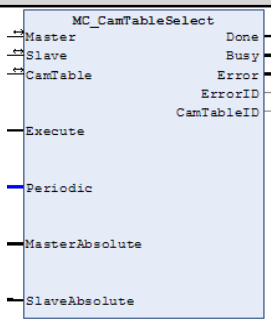
（3） Function description

This function block evaluates the output Tappets of the function block MC_CamIn and contains the current tappet status.

4. 2. 4 Cam Table Designation Command MC_CamTableSelect

This function block is designed to select the cam table by setting up a connection to the relevant table.

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_ CamTableSelect	Cam table designation		MC_CamTableSelect0(Master:= , Slave:= , CamTable:=), Execute:= , Periodic:= , MasterAbsolute:= , SlaveAbsolute:= , Done=> , Busy=> , Error=> , ErrorID=> , CamTableID=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Master	Main shaft	AXIS_ REF_SM3		Mapping to the main axis, an instance of AXIS_REF_SM3.
	Slave	From the shaft	AXIS_ REF_SM3		mapped to the slave axis, an instance of AXIS_REF_SM3.
	CamTable	Select Table	MC_ CAM_REF		Mapped to a CAM table description, an instance of MC_CAM_REF.
Input	Execute	Execution	BOOL	FALSE	Rising edge signal to execute the command.
	Periodic	Repeat Mode	BOOL	TRUE	Specify whether the specified cam table should be executed repeatedly or only once TRUE: Repeat False: No duplication
	Master Absolute	Main shaft Absolute Mode	BOOL	TRUE	Specify whether the spindle tracking distance coordinate system is by absolute or relative position 1: Absolute position 0: Relative position

Scope	Name	Chinese Name	Type	Initial	Comment
Input	Slave Absolute	From the shaft Absolute Mode	BOOL	TRUE	The StartMode in the MC_CamIn command specifies whether the current command position of the slave axis is absolute (the cam table output value corresponding to the current spindle position) or relative (the cam table output value is superimposed on the slave axis position at the start of the command) to the cam table output. 1: absolute position, 0: relative position.
Output	Done	Completion	BOOL	**FALSE**	Select TRUE for completion.
	Busy	Under implementation	BOOL	**FALSE**	TRUE when there is no completion in the selection.
	Error	Error	BOOL	**FALSE**	Set to TRUE when the exception occurs.
	ErrorID	Error Code	SMC_ ERROR	0	Outputs an error code when an exception occurs.
	CamTableID	Effective CAMID	MC_CA M_ID		Select the effective Cam_ID, the same as in the MC_CamIn command The CamTableID is used in conjunction with the

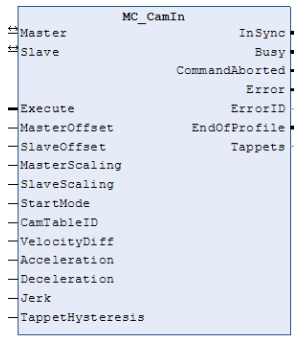
（3） Function description

This function block is designed to select a cam table by setting up a connection to the relevant table.

4. 2. 5 Cam action start command MC_Camin

The function block implements the selected cam plate.

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_CamIn	Start of cam action		MC_CamIn0(Master:= , Slave:= , Execute:= , MasterOffset:= , SlaveOffset:= , MasterScaling:= , SlaveScaling:= , StartMode:= , CamTableID:= , VelocityDiff:= , Acceleration:= , Deceleration:= , Jerk:= , TappetHysteresis:= , InSync=> , Busy=> , CommandAborted=> , Error=> , ErrorID=> , EndOfProfile=> , Tappets=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Master	Main shaft	AXIS_ REF_SM3		Mapping to an axis, an instance of AXIS_REF_SM3.
	Slave	From the shaft	AXIS_ REF_SM3		Mapping to an axis, an instance of AXIS_REF_SM3.
Input	Execute	Perform cam work into the energy block	BOOL		Rising edge, execution of electronic cam
	MasterOffset	Spindle Offset	LREAL		Shifts the phase of the spindle by the specified offset value.
	SlaveOffset	Paranoia from the axis	LREAL		Shifts the phase of the slave axis by the specified offset value.
Input	MasterScaling	Spindle pre-programming Translation ratio	LREAL	1	Enlarges/reduces the phase of the spindle by a specified ratio
	SlaveScaling	Pre-programmed from axis Translation ratio	LREAL	1	Enlarges/reduces the phase of the slave axis by a specified ratio

Scope	Name	Chinese Name	Type	Initial	Comment
Input	StartMode	Slave relative cam output mode	MC_ StartMode	absolute	0: Absolute absolute position. 1：relative relative position. 2: ramp_in (ramp cut in) 3: ramp_in_pos (forward slope cut) 4: ramp_in_neg reverse ramp cut-in
	CamTableID	Form Number	MC_ CAM_ID		Defines the use of cam tables in conjunction with the output point CamTableID of MC_ CamTableSelect.
	VelocityDiff		LREAL		A different maximum speed than ramp_in.
	Acceleration		LREAL		Acceleration at ramp_in.
	Deceleration		LREAL		Decreases the speed when ramp_in.
	Jerk		LREAL		The acceleration of ramp_in.
	Tappet Hysteresis		LREAL		Damping factor of tappet.
Output	InSync		BOOL		InSync is set after the cam relationship between the master and slave axes is established and is reset when the execution condition of the command is OFF.
	Busy		BOOL		When the rising edge of Execute input is set to TRUE, TRUE means that the cam relationship is coupled and needs to be reset with Cam_out Command.

Scope	Name	Chinese Name	Type	Initial	Comment
Output	Command Aborted		BOOL		The output of the slave axis interrupted by other control Commands is TRUE.
	Error		BOOL		If an error is detected, the Error bit is set. The Error bit is reset when the execution condition of the order is OFF.
	ErrorID		SMC _ERROR		Output error code when exception occurs
Output	EndOfProfile		BOOL		If the Periodic parameter is 0 (non-periodic) when the MC_CamTableSelect Command is executed, the EndOfProfile bit is set when the cam curve has been executed once, and the EndOfProfile bit is reset when the execution condition of the Command is OFF.
	Tappets		SMC_ TappetData		Associating a cam tappet, which can be read out by MC_GetTappetValue command

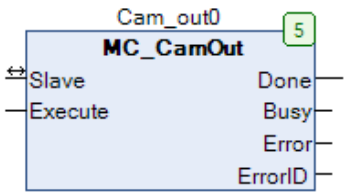
（3） Function description

The function block implements the selected cam plate.

4. 2. 6 Disconnecting Cam Coupling Command MC_CamOut

This function block immediately separates the slave axis from the spindle.

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_CamOut	Disconnecting cam coupling		MC_CamOut(Slave:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Slave	From the shaft	AXIS_REF_SM3		Mapping to an axis, an instance of AXIS_REF_SM3.
Input	Execute	Execution Commands	BOOL	FALSE	The rising edge signal executes the command.
Input	Done	Completion	BOOL	FALSE	Complete disconnection of the cam coupling from the spindle.
	Busy	Under implementation	BOOL	FALSE	Command execution in progress.
	Error	Error	BOOL	FALSE	Set to TRUE when the exception occurs.
	ErrorID	Error Code	SMC_ERROR	0	Outputs an error code when an exception occurs.

（3） Function description

This function block immediately separates the slave axis from the spindle.

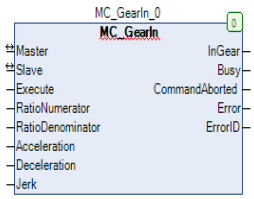
Note:

Assuming that this command is followed by another command, e.g. MC_Stop, MC_GearIn. If there is no further command, the default condition should be: keep the last speed.

4. 2. 7 Electronic Gear Function Block Command MC_GearIn

Set the gear ratio between slave shaft and spindle for electronic gear action.

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_GearIn	Electronic Gear Function blocks		MC_GearIn0(Master:= , Slave:= , Execute:= , RatioNumerator:= , RatioDenominator:= , Acceleration:= , Deceleration:= , Jerk:= , InGear=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Master	Main shaft	AXIS_ REF_SM3		Mapping to an axis, an instance of AXIS_REF_SM3.
	Slave	From the shaft	AXIS_ REF_SM3		Mapping to an axis, an instance of AXIS_REF_SM3.
Input	Execute	Execution	BOOL		Rising edge, start executing the command.
	Ratio Numerator	Gear ratio molecule	DINT	1	Gear ratio molecule.
	Ratio Denominator	Gear score master	UDINT	1	The gear is more than the mother.
	Acceleration	Acceleration	LREAL		Specifies the acceleration.
	Deceleration	Deceleration	LREAL		Specify the deceleration rate.
	Jerk	Yue Du	LREAL		Add acceleration.
	BufferMode		MC_BUFFE R_MODE		True, the target speed is reached from the axis.
Output	InGear	Gear ratio arrives	BOOL		True, the command is being executed
	Busy	Under implementation	BOOL		True, interrupted by other control Commands
	Active		BOOL		Set to TRUE when an exception occurs

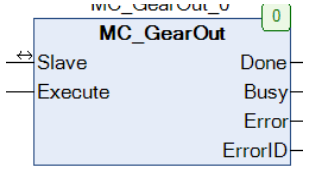
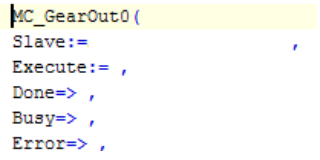
Scope	Name	Chinese Name	Type	Initial	Comment
	Command Aborted	Interruptions	BOOL	FALSE	Output error code when exception occurs
	Error	Error	BOOL		True, from axis to target speed
	ErrorID	Error Code	SMC_ERROR		True, the command is being executed.

（3） Function description

4. 2. 8 Electronic Gear Coupling Disconnect Command MC_GearOut

This function block separates the slave shaft from the spindle.

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_GearOut	Electronic Gear Coupling disconnection		

（2） Relevant variables

Scope	Name	Chinese Name	Type	Comment
Inout	Slave	From the shaft	AXIS_REF_SM3	Mapped to an axis, an instance of AXIS_REF_SM3.
Input	Execute	Execution	BOOL	Rising edge, start executing the command.
Output	Done	Completion	BOOL	True, the slave shaft is disconnected from the spindle electronic gear coupling.
	Busy	Under implementation	BOOL	True, the command is being executed.
	Error	Error	BOOL	Set to TRUE when the exception occurs.
	ErrorID	Error Code	SMC_ERROR	Outputs an error code when an exception occurs.

（3） Function description

This function block separates the slave shaft from the spindle.

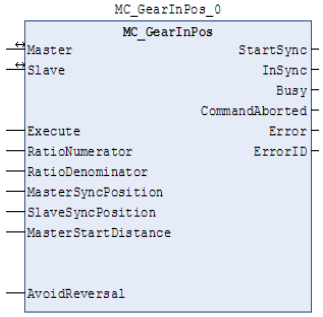
Note:

Assuming this command is followed by another command, e.g. MC_Stop, MC_GearIn. If there is no further command, the default condition is: keep the last speed.

4. 2. 9 Specifying the position to cut into the electronic gear coupling MC_GearInPos

Considering the specific position relationship, MC_GearInPos combines the slave axis with the mainaxis.

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_GearInPos	Specify the position to cut into the electronic Gear Coupling		MC_GearInPos0(Master:= Slave:= Execute:= , RatioNumerator:= , RatioDenominator:= , MasterSyncPosition:= SlaveSyncPosition:= , MasterStartDistance:= AvoidReversal:= , StartSync=> , InSync=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Master	Main shaft	AXIS_REF_SM3		Mapped to an axis, an instance of AXIS_REF_SM3.
	Slave	From the shaft	AXIS_REF_SM3		Mapping to an axis, an instance of AXIS_REF_SM3.
Input	Execute	Command Execution	BOOL		Rising edge, start of command execution
	Ratio Numerator	Gear Ratio Molecular	DINT	1	Numerator of master-slave speed ratio
	Ratio Denominator	Gear Ratio Denominator	DINT	1	Denominator of master-slave speed ratio

	Master SyncPosition	Spindle synchronization position	LREAL		Spindle position when coupling master to slave gear ratio.
	SlaveSync Position	Slave axis synchronization position	LREAL		Slave shaft position when coupling master to slave gear ratio.

Scope	Name	Chinese Name	Type	Initial	Comment
Input	Master StartDistance	Execution of synchronized spindle position	LREAL		From the axis according to this position value and -M asterSyncPosition and SlaveSyncPosition values calculate a smooth curve to synchronize the slave axis with the spindle gear at SlaveSyncPosition, and main axis curve range is [MasterStartDistance, MasterSyncPosition].
Input	BufferMode		MC_ BUFFER _MODE		Defines the chronological sequence of the FB relative to the previous block. only the BufferModes Aborting, Buffered and Blending Previous are supported. BlendingPrevious means that the configured velocity (including the direction) of the previous movement is used as the blending If the function block is Busy, then only BufferMode=Aborting is allowed.
	AvoidReversal		BOOL		Set to FALSE if reversal is performed if the slave axis physically overruns its position. Set to TRUE if reversal is not physically

					possible from the axis or causes a hazard. Only applicable with modal axes. If reversal cannot be avoided, then the axis will stop incorrectly.

Scope	Name	Chinese Name	Type	Initial	Comment
Output	StartSync	Start coupling processing	BOOL		True, start electronic gear coupling process
	InSync	In Coupling	BOOL		True, the electronic gear coupling process is completed, the master-slave gear ratio coupling in
	Busy	Command In process	BOOL		True, the command is being processed
	Active		BOOL		Interrupted by other control Commands
	Command Aborted	Command interrupt	BOOL	FALSE	Set to TRUE when an exception occurs
	Error	Error	BOOL		Output error code when exception occurs
	ErrorID	Error Code	SMC_ ERROR		True, start electronic gear coupling process

（3） Function description

Considering the specific position relationship, MC_GearInPos combines the slave axis with the mainaxis.

a. Important:

In some cases, it may not be possible to avoid the reversal of the slave station. (Even if the velocity of the slave station is not opposite to the master. An example is a slave with a small positive velocity and a large negative acceleration. It may not be possible to reduce the acceleration fast enough to prevent the velocity from becoming negative.

b. Timing chart:

After the buffer mode is aborted, any previous motion will continue until the master starts in the correct direction across the position master position. At this point, the output begins synchronization setup. When a stop command is executed on the slave axis before synchronization occurs, it will inhibit synchronization and the function block issue command has been implemented = TRUE.

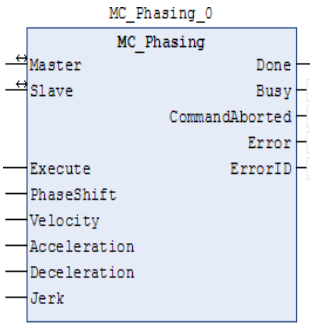
If the main start is not specified (expressed as a non-positive value), the function block calculates the set start distance point so that the synchronization starts from the current main position.

Using buffered mode buffering or mixed preprocessing, the input master start point cannot be provided (i.e., it must be set to 0). The reason for this is that once the last movement is completed, MC_GearInPos must take over the axis and continue the movement; it cannot wait until the master sync position has been reached. (The error SMC_GIP_CANNOT_START_SYNC is reported if the master is at a standstill while completing the last movement.

4. 2. 10 Master-Slave Phase Offset Command MC_Phasing

Specifies the phase deviation between the master and slave axes.

（1） Command format

Command	Name	Graphical representation	ST Performance
MC_Phasing	Master-slave axis phase shift		MC_Phasing0(Master:= , Slave:= , Execute:= , PhaseShift:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Master	Main shaft	AXIS_ REF_SM3		Mapped to axis, an instance of AXIS_REF_SM3
	Slave	From the shaft	AXIS_ REF_SM3		Mapping to an axis, an instance of AXIS_REF_SM3.
Input	Execute	Command Execution	BOOL	FALSE	Rising edge, start executing the command.
	PhaseShift	Master-slave axis phase deviation value	LREAL	0	Master-slave axis phase deviation value, positive number represents slave axis lag.
	Velocity	Speed	LREAL	0	Maximum speed value when performing phase shift.
	Acceleration	Acceleration	LREAL	0	Maximum acceleration value when performing phase shift.
	Deceleration	Deceleration	LREAL	0	Maximum deceleration value when performing phase shift.

Scope	Name	Chinese Name	Type	Initial	Comment
	Jerk	Second derivative of velocity	LREAL		Maximum Jerk value when performing a phase shift.
	Done	Completion	BOOL	**FALSE**	True, if the phase shift is complete
	Busy	Command In process	BOOL	**FALSE**	True, the command is being processed
	Command Aborted	Command interrupt	BOOL	**FALSE**	Interrupted by other control Commands
	Error	Error	BOOL	**FALSE**	Set to TRUE when an exception occurs
	ErrorID	Error Code	SMC_ ERROR	0	Output error code when exception occurs

（3） Function description

Specifies the phase deviation between the master and slave axes.

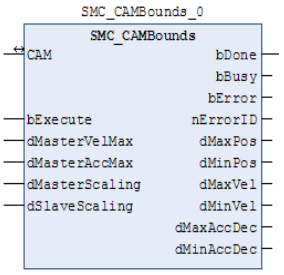
MC_Phasing will abort any ongoing motion of the slave axis. In this respect, it differs from the function blocks MC_PhasingAbsolute and MC_PhasingRelative defined by the PLCopen Motion Control 2.0 standard. these function blocks do not abort ongoing slave motion, but rather superimpose phase transfers, such as ongoing MC_CamIn or MC_GearIn motion.

To implement this behavior with MC_Phasing, use an additional virtual axis as the slave axis for MC_Phasing and use this virtual slave as the master MC_CamIn or MC_GearIn. Alternatively, MC_MoveSuperimposed can be used on the main axis for MC_CamIn or MC_GearIn motion.

4. 2. 11 Cam upper and lower limit commands SMC_CAMBounds

This function block calculates the maximum position, velocity and acceleration values of a slave coupled in absolute mode to a master that moves according to the specified maximum velocity and acceleration/deceleration.

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC_CAMBounds	Cam upper and lower limits		SMC_CAMBounds_0(CAM:= , bExecute:= , dMasterVelMax:= , dMasterAccMax:= , dMasterScaling:= , dSlaveScaling:= , bDone=> , bBusy=> , bError=> , nErrorID=> , dMaxPos=> , dMinPos=> , dMaxVel=> , dMinVel=> , dMaxAccDec=> , dMinAccDec=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	CAM	Cams	MC_CAM_REF		Reference to cam.
Input	bExecute	Command Execution	BOOL		Rising edge, start executing the command.
	dMasterVelMax	Maximum speed	LREAL	1	Maximum spindle speed in absolute mode.
	dMasterAccMax	Maximum acceleration	LREAL	0	Maximum spindle acceleration in absolute mode.
	dMasterScaling	Scale factor	LREAL	1	Scale factor in spindle cam application.
	dSlaveScaling	Scale factor	LREAL	1	Scale factor in slave cam application.
Output	bDone	Completion	BOOL		True, if the calculation is complete.
	bBusy	Command processing in progress	BOOL	**FALSE**	True, the command is being processed.

Scope	Name	Chinese Name	Type	Initial	Comment
Output	bError	Error	BOOL		Set to TRUE when the exception occurs.
	nErrorID	Error Code	SMC_ERROR		Outputs an error code when an exception occurs.
	dMaxPos	Maximum position	LREAL		Calculate the maximum slave shaft position from

					the cam table.
	dMinPos	Minimum position	LREAL		Calculate the minimum slave shaft position from the cam table.
	dMaxVel	Maximum speed	LREAL		Calculate the maximum speed.
	dMinVel	Minimum speed	LREAL		Calculate the minimum speed.
	dMaxAccDec	Maximum acceleration	LREAL		Calculate the maximum acceleration.
	dMinAccDec	Minimum acceleration	LREAL		Calculate the minimum acceleration.

（3） Function description

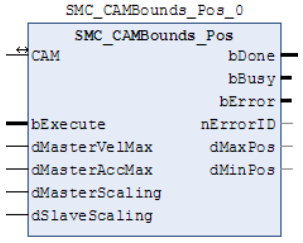
This function block calculates the maximum position, velocity and acceleration values of a slave coupled in absolute mode to a master that moves according to the specified maximum velocity and acceleration/deceleration.

This module is especially helpful when cam discs are created and modified in online mode and compliance with the maximum values must be checked in advance.

4. 2. 12 Cam position upper/lower limit command SMC_CAMBounds_Pos

This function block calculates the maximum position, velocity and acceleration values of a slave coupled in absolute mode to a master that moves according to the specified maximum velocity and acceleration/deceleration.

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC_ CAMBounds_Pos	Cam position upper and lower limits		SMC_CAMBounds_Pos0(CAM:= , bExecute:= , dMasterVelMax:= , dMasterAccMax:= , dMasterScaling:= , dSlaveScaling:= , bDone=> , bBusy=> , bError=> , nErrorID=> , dMaxPos=> , dMinPos=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	CAM	Cams	MC_CAM_REF		Mapped to cam, an instance of MC_CAM_REF
Input	bExecute	Command Execution	BOOL		True, if the calculation is complete.
	dMasterVelMax	Maximum speed	LREAL	1	True, the command is being processed.
	dMaster AccMax	Maximum acceleration	LREAL	0	Set to TRUE when the exception occurs.
	dMasterScaling	Scale factor	LREAL	1	Outputs an error code when an exception occurs.
	dSlaveScaling	Scale factor	LREAL	1	Calculate the maximum slave shaft position from the cam table.
Output	bDone	Completion	BOOL		True, if the calculation is complete.
	bBusy	Command In process	BOOL	FALSE	True, the command is being processed.

Scope	Name	Chinese Name	Type	Initial	Comment
Output	bError	Error	BOOL		Set to TRUE when the exception occurs.
	nErrorID	Error Code	SMC_ERROR		Outputs an error code when an exception occurs.
	dMaxPos	Maximum position	LREAL		Calculate the maximum slave shaft position from the cam table.
	dMinPos	Minimum position	LREAL		Calculate the minimum slave shaft position from the cam table.

（3） Function description

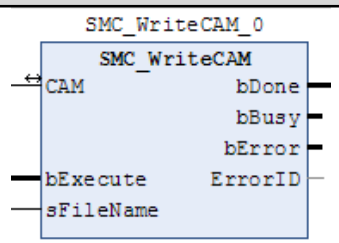
With SMC_CAMBounds this function block calculates only the maximum and minimum position values of the slaves coupled in absolute mode, and this master position value is moved according to the specified maximum speed and acceleration/deceleration.

This module is especially helpful when cam discs are created and modified in online mode and compliance with the maximum values must be checked in advance.

4. 2. 13 Cam up/down command SMC_WriteCAM

This function block is designed to store the cam table created in the Cam Editor to a file at runtime

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC_WriteCAM	Cam upper and lower limits		SMC_WriteCAM(CAM:= , bExecute:= , sFileName:= , bDone=> , bBusy=> , bError=> , ErrorID=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	CAM	Cams	MC_CAM_REF		Mapped to cam, an instance of MC_CAM_REF
Input	bExecute	Command Execution	BOOL	FALSE	Rising edge, start executing the command.
	sFileName	File name	STRING (255)		The file name defined in ASCII format containing the cam description can be found in the help "Cam Format" to see the specific description.
Output	bDone	Completion	BOOL	FALSE	True, if the cam writes to the file to complete.
	bBusy	Command In process	BOOL	FALSE	True, the Command execution is not completed.
	bError	Error	BOOL	FALSE	Set to TRUE when the exception occurs.
	ErrorID	Error Code	SMC_ERROR	0	Outputs an error code when an exception occurs.

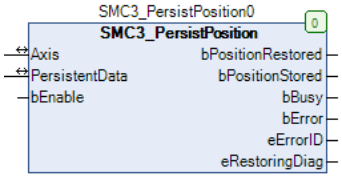
（3） Function description

This function block is designed to store the cam table created in the Cam editor to a file at runtime.

4. 2. 14 Axis position hold SMC3_PersistPosition

This function block is used to hold the axis position of a multi-turn absolute encoder with a real axis.

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC3_PersistPosition	Axis position Maintain		SMC3_PersistPosition0(Axis:= , PersistentData:= , bEnable:= , bPositionRestored=> , bPositionStored=> , bBusy=> , bError=> , eErrorID=> , eRestoringDiag=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_ REF_SM3		Mapping to an axis, an instance of AXIS_REF_SM3.
	Persistent Data	Maintain Data	SMC3_ Persist Position_ Data		A power-off hold data structure for storing position information.
Input	bEnable	Execution	BOOL	FALSE	True function blocks are executed, false does not execute function blocks to restore the last stored location during initialization, the value must be set to true from application startup.
Output	bPosition Restored	Location Recovery	BOOL		TRUE, position recovery completed after axis restart
	bPosition Stored	Location Save	BOOL		TRUE, save the location after calling the function fast.
	bBusy	FB Under implementation	BOOL		TRUE, the execution of the function block is not completed.
	bError	Error	BOOL	FALSE	TRUE, Exception occurred.

	eErrorID	ErrorCode	SMC_ ERROR	SMC_NO _ERROR	Outputs an error code when an exception occurs.

Scope	Name	Chinese Name	Type	Initial	Comment
Output	eRestoring Diag	Recovery Diagnosis	SMC3_ Persist Position Diag	SMC3_ Persist PositionDiag. SMC3_ PPD_ RESTORING_O K	Diagnostic letter SMC3_PPD_R in position recovery ESTORING_OK. Location successfully recovered SMC3_PPD_AXIS_ PROP_CHANGED: Axis parameters have changed, no Law Recovery Location SMC3_PPD_DATA _STORED_DURIN G_WRITING: Function block to copy data from axis parameter data structure instead of PersistentData numbers. According to the replication in. Possible causes: non-identical

（3） Function description

This function block is used to hold the axis position of a multi-turn absolute encoder with a real axis.

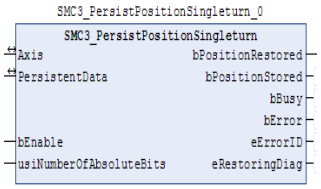
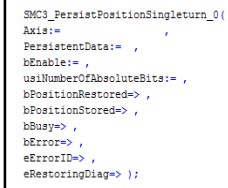
This function block assumes a multi-turn encoder with a range of 2^32.

For multi-turn encoders with smaller ranges, use SMC3_PersistPositionSingleturn.

4. 2. 15 Axis Position Hold SMC3_PersistPositionSingleturn

This function block is used to hold the axis position of an absolute encoder with a limited range of real axes.

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC3_ PersistPositionSingleturn	Axis position Maintain		

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_ REF_SM3		
	Persistent Data	Maintain data	SMC3_Persis tPositionSingl eturn_Data		
Input	bEnable	Execution	BOOL	FALS E	True function blocks are executed, false does not execute function blocks PLC reboot needs to be true to restore the location stored before reboot.
	usiNumberOf AbsoluteBits	Digits	USINT	16	How many bits of absolute Value encoder (e.g. 20-bit, 24-bit editor (Coders, etc.)
Output	bPosition Restored	Position Recovery	BOOL		TRUE, Position recovery is complete after axis restart.
	bPosition Stored	Location saving	BOOL		TRUE, save bit after calling function fast The placement is complete.

Scope	Name	Chinese Name	Type	Initial	Comment
Output	bBusy	FB Under implementation	BOOL		TRUE, the execution of the function block is not completed.
	bError	Error	BOOL	FALSE	TRUE, Exception occurred.
Output	eErrorID	Error Code	SMC _ERRO R	SMC_ NO_ERROR	Outputs an error code when an exception occurs.
	eRestoring Diag	Recovery Diagnosis	SMC3_ Persist Position Diag	SMC3_ PersistPosition Diag.SMC3_ PPD_RESTORIN G_OK	Position recovery in Diagnostic information.

（3） Function description

This function block is used to hold the axis position of an absolute encoder with a limited range of real axes.

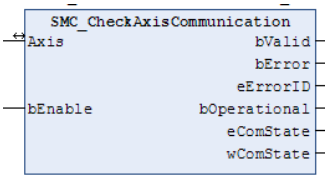
Strictly speaking, this function block is not only used for single-turn encoders, but also for multi-turn encoders with an encoder range of 2 and less than 2^32.

Range of two encoders - 2^k . . .2^k - 1 and the range 0 . . 2^(k+1) are supported.

4. 2. 16 Axis Limit Check SMC_CheckAxisCommunication

Checks the current communication status of the axis and returns the current communication status of the axis.

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC_CheckLimits	Axis limitation Check		SMC_CheckAxisCommunication0(Axis:= , bEnable:= , bValid=> , bError=> , eErrorID=> , bOperational=> , eComState=> , wComState=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Comment
Inout	Axis	Shaft	AXIS_ REF_SM3	Mapping to an axis, an instance of AXIS_REF_SM3.
Input	bEnable	Execution	BOOL	TRUE: Execution check in progress.
Output	bValid	Under implementation	BOOL	True, the command is valid.
	bError	Error	BOOL	True, the exception is generated.

	eErrorID	Error Code	SMC_ ERROR	Refer to SMC_Error.
	bOperational	Communication is normal	BOOL	True, communication is normal (code 100) operable. False, the communication is not normal and the axis cannot be operated.
Output	eComState	Communication Status	SMC_ CommunicationState	Contains: SMC_COMSTATE_NOT_STARTED, communication not initiated. SMC_COMSTATE_VARIABLE_INITIALIZATION, initialization of communication variables. SMC_COMSTATE_BASE_COM_INITIALIZATION, basic port initialization. SMC_COMSTATE_DRIVE_INITIALIZATION, communication driver initialization. SMC_COMSTATE_DRIVE_WAITING_FOR_SYNC, synchronization warning. SMC_COMSTATE_INITIALIZATION_DONE, initialization complete. SMC_COMSTATE_OPERATIONAL, communication is functional. SMC_COMSTATE_REINITIALIZATION, communication re-initialization. SMC_COMSTATE_ERROR, communication error. SMC_COMSTATE_UNKNOWN communication status unknown;
	wComState	Communication Code	WORD	with the input and output axis structure variables in: The Axis.wCommunicationState value is the same. The code indicating the current communication status, refer to AXIS_REF_SM3 parameter 1013.

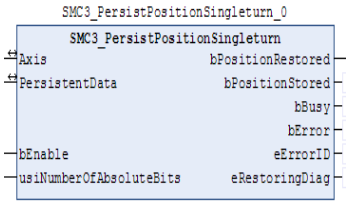
（3） Function description

Returns the current communication status of the axis.

4. 2. 17 Axis Position Hold SMC3_PersistPositionSingleturn

This function block is used to hold the axis position of an absolute encoder with a limited range of real axes.

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC3_ PersistPositionSingleturn	Axis position Maintain		SMC3_PersistPositionSingleturn_0(Axis:= , PersistentData:= , bEnable:= , usiNumberOfAbsoluteBits:= , bPositionRestored=> , bPositionStored=> , bBusy=> , bError=> , eErrorID=> , eRestoringDiag=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_ REF_SM3		Mapped to an axis, an instance of AXIS_REF_SM3.
	Persistent Data	Maintain data	SMC3_ PersistPositionSingletur_ Data		Mapped to the record position structure as SMC3_ PersistPosition_Data (the structure variable must be a power-down hold type)
Input	bEnable	Execution	BOOL	FALSE	True function blocks are executed, false does not execute function blocks; PLC needs to be true after restart to restore the stored position before restart.
	usiNumberOf AbsoluteBits	Digits	USINT	16	How many bits absolute encoders (e.g. 20-bit, 24-bit encoders, etc.)
Output	bPosition Restored	Position Recovery	BOOL		TRUE, Position recovery is complete after axis restart.
	bPosition Stored	Location saving	BOOL		TRUE, save the location after calling the function fast.
Output	bBusy	FB Under implementation	BOOL		TRUE, the execution of the function block is not completed.
	bError	Error	BOOL	FALSE	TRUE, exception occurred

Scope	Name	Chinese Name	Type	Initial	Comment
	eErrorID	Error Code	SMC_ ERROR	SMC_ NO_ ERROR	Outputs an error code when an exception occurs.
	eRestoring Diag	Recovery Diagnosis	SMC3_ Persist Position Diag	SMC3_ Persist Position Diag. SMC3_ PPD_RE STORIN G_OK	Diagnostic information in location recovery.

（3） Function description

This function block is used to hold the axis position of an absolute encoder with a limited range of real axes.

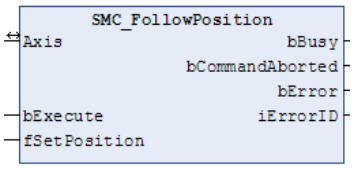
Strictly speaking, this function block is not only used for single-turn encoders, but also for multi-turn encoders with an encoder range of 2 and less than 2^32.

Range of two encoders - 2^k . . .2^k - 1 and the range 0 . . 2^(k+1) are supported.

4. 2. 18 Axis Position Giving Command SMC_FollowPosition

This function block sets the position to the axis without performing any checks

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC_FollowPosition	Axis position Given		SMC_FollowPosition_0(Axis:= , bExecute:= , fSetPosition:=SET_POSITION , bBusy=> , bCommandAborted=> , bError=> , iErrorID=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Comment
Inout	Axis	Shaft	AXIS_ REF_SM 3	Mapped to an axis, an instance of AXIS_REF_SM3.
Input	bExecute	Execution	BOOL	Rising edge execution function block.
	fSetPosition	Set position	LREAL	The position of the axis setting.
Output	bBusy	Under implementation	BOOL	True, the command is in execution, (the axis is in synchronous state, the same as when the cam MC_CamIn Command is running), you can use MC_ Camout Command to clear the bBusy state.

	bCommand Aborted	Command Interrupted	BOOL	True, axis is interrupted by other control commands
	bError	Error	BOOL	True, exception generated
	iErrorID	Error Code	SMC_ ERROR	Reference SMC_Error

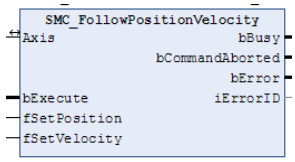
（3） Function description

This function block sets the position to the axis without performing any checks

4. 2. 19 Axis position and velocity giving command SMC_FollowPositionVelocity

This function block employs the equivalent of SMC_FollowPosition, with the possible exception of the defined speed.

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC_ FollowPositionVelocity	Axis position, Speed given		SMC_FollowPositionVelocity_0(Axis:= , bExecute:= , fSetPosition:= , fSetVelocity:= , bBusy=> bBusy, bCommandAborted=> , bError=> , iErrorID=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Comment
Inout	Axis	Shaft	AXIS_ REF_SM 3	Mapping to an axis, an instance of AXIS_REF_SM3.
Input	bExecute	Execution	BOOL	Rising edge execution function block.
	fSetPosition	Set position	LREAL	The position of the axis setting.
	fSetVelocity	Set speed	LREAL	The position of the axis setting.
Output	bBusy	Under implementation	BOOL	True, the command is executing, (at this time the axis is in synchronous state, the same as the axis state when the cam MC_CamIn Command is running), you can clear the bBusy state with MC_ Camout Command.
	bCommand Aborted	Command is interrupted	BOOL	True, axis is interrupted by other control commands.
	bError	Error	BOOL	True, the exception is generated.
	iErrorID	Error Code	SMC_ ERROR	True- The Command is executing, (at

				this time the axis is in synchronous state, the same as the axis state when the cam MC_CamIn Command is running), you can clear the bBusy state with MC_ Camout Command.

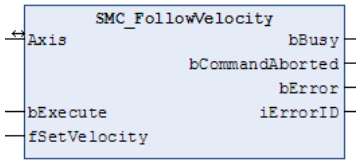
（3） Function description

This function block similar with SMC_FollowPosition, but it can define speed.

4. 2. 20 4Axis velocity giving command SMC_FollowVelocity

This function block writes the set speed to the axis without performing any checks.

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC_FollowVelocity	Axis speed Given		SMC_FollowVelocity_0(Axis:= , bExecute:= , fSetVelocity:= , bBusy=> , bCommandAborted=> , bError=> , iErrorID=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Comment
Inout	Axis	Shaft	AXIS_ REF_SM 3	Mapped to an axis, an instance of AXIS_REF_SM3.
Input	bExecute	Execution	BOOL	Rising edge execution function block.
	fSetVelocity	Set position	LREAL	The speed set by the axis.
Output	bBusy	Under implementation	BOOL	True- The Command is executing, (at this time the axis is in synchronous state, the same as the axis state when the cam MC_ CamIn Command is running), you can clear the bBusy state with the MC_ Camout Command.
	bCommand Aborted	Command is interrupted	BOOL	True- Axis is interrupted by other control commands (when bExecute is True)
	bError	Error	BOOL	True, the exception is generated.
	iErrorID	Error Code	SMC _ERROR	Refer to SMC_Error.

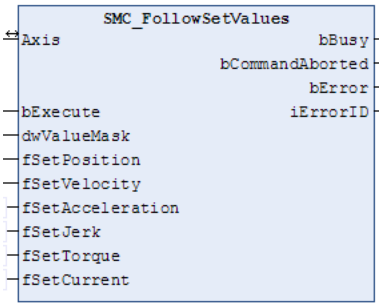
（3） Function description

This function block writes the set speed to the axis without performing any checks.

4. 2. 21 Axis-related Commands given by the Command SMC_FollowSetValues

Writes the set value to the axis without doing any checking

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC_FollowSetValues	Axis Related Command Given		SMC_FollowSetValues_0(Axis:= , bExecute:= , dwValueMask:= , fSetPosition:= , fSetVelocity:= , fSetAcceleration:= , fSetJerk:= , fSetTorque:= , fSetCurrent:= , bBusy=> , bCommandAborted=> , bError=> , iErrorID=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Comment
Inout	Axis	Shaft	AXIS_REF_SM3	Mapped to an axis, an instance of AXIS_REF_SM3.
Input	bExecute	Execution	BOOL	Rising edge execution function block.
	bAbort		BOOL	Abort the execution of the function block (e.g., to avoid an error when restarting with a different axis)
	dwValueMask	Control Management	DWORD	Bite0: TRUE: fSetPosition active FALSE: ignore Bite1: TRUE: fSetVelocity active FALSE: ignore Bite2: TRUE: fSetAcceleration active FALSE: Ignore Bite3: TRUE: fSetJerk active FALSE: ignore Bite4: TRUE: fSetTorque active FALSE: ignore

				Bite5: TRUE: fSetCurrent active FALSE: ignore
	fSetPosition	Set position	LREAL	Position of the axis setting (calibrated units)
	fSetVelocity	Set speed	LREAL	Axis set speed (calibrated units /s)
	fSetAcceleration	Settings Acceleration	LREAL	Acceleration of the axis setting (calibrated units /s2)
	fSetJerk	Settings Jumpiness value	LREAL	Axis-set leap value (calibrated units /s3)
	fSetTorque	Setting torque	LREAL	Leap value for axis setting (NM/N)
Input	fSetCurrent	Set current	LREAL	Current value for axis setting (A)
Output	bBusy	Under implementation	BOOL	True- The Command is being executed, (the axis is in synchronous state, the same as when the cam MC_CamIn Command is running), you can use MC_Camout Command to clear the bBusy state
	bCommand Aborted	Command Interrupted	BOOL	True- The axis is interrupted by other control commands
	bError	Error	BOOL	True, the exception is generated.
	iErrorID	Error Code	SMC _ERROR	Refer to SMC_Error.

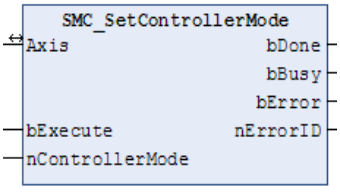
（3） Function description

Writes the set value to the axis without doing any checking.

4. 2. 22 Set axis control mode command SMC_SetControllerMode

This function block can be used to switch to other controller modes if supported by the driver.

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC_ SetControllerMode	Setting axis Control Mode		SMC_SetControllerMode0(Axis:= , bExecute:= , nControllerMode:= , bDone=> , bBusy=> , bError=> , nErrorID=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS _REF_SM3		Mapped to the axis, an instance of AXIS_REF_SM3.
Input	bExecute	Execution	BOOL	FALSE	Rising edge execution function block
	nControllerMode	Control Mode	SMC_CONTROLLER_MODE	SMC _position	Axis control mode 1: Torque control mode, SMC_torque 2: Speed control mode, SMC_Velocity 3: Position control mode, SMC_Position 4: Current control mode, SMC_Current
Output	bDone	Mode Settings Completion	BOOL		True, the mode setting is completed.
	bBusy	Under implementation	BOOL		True- Command execution is in progress.
	bError	Error	BOOL		True, the exception is generated.
	nErrorID	Error Code	SMC _ERROR		Refer to SMC_Error.

（3） Function description

This function block can be used to switch to other controller modes if supported by the driver.

- **Prerequisites required:**

a. The axes must support the desired control mode. To check this, refer to the function documentation PDF of the corresponding soft motion driver library.

b. The required cyclic I/O data must be plotted (e.g., torque mode: set torque object).

c. When this function block is called, the axis must not be in the state error STOP, STOP, or WHIRRING. Otherwise, the error SMC_SCM_AXIS_IN_WRONG_STATE will be reported.

- **Behavior at the time of the event:**

a. This function block sets the parameter of the controller module (command value of the operating mode) of the axis and waits until the parameter of the real controller module (actual value of the operating mode) reflects this value.

b. If this does not happen within 1000 cycles, the function block will be aborted, and an error will occur.

c. When switching from a lower level to a higher-level control mode (e.g. Torque -> Speed, Torque-> Position, Speed-> Position), the function block calculates the set value of the higher level signal. For example, when switching from torque mode to position mode, adjustments are made. The function block uses the actual position of the axis and adds the expected position distance to compensate for the time difference between the actual value and the set value based on the actual speed and the time shift in the cycle (Axis.fSetActTimeLag cycle). The latter value depends mainly on the system (controller hardware, fieldbus, drives) and may need to be adjusted.

- **What function blocks must be called after SMC_SetControllerMode outputs bDone?**

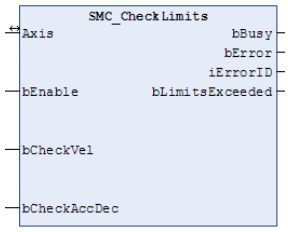
Once the controller mode is reflected by the controller mode, the function block sets its completion output and stops any operation. This means that the application program is immediately responsible for calculating the correct set value from this cycle. For example, an axis being controlled by torque should be switched to position control.

When the function block is started, the axis is still moving SMC_SetControllerMode. during the switching time, the function block calculates the correct value for the set position. However, once the setup is complete, any function block will not continue to provide the setup value and the axis will stop immediately and report an error. So, for example, function blocks for MC_Halt, MC_MoveVelocity or MC_MoveAbsolute may be triggered by the completion output of SMC_ControllerMode and take over control of the axis.

4. 2. 23 Axis limit check command SMC_CheckLimits

This function block can be used to check if the current setpoint of the drive exceeds the maximum value configured in the controller. The result of the check will be displayed by output limit.

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC_CheckLimits	Shaft limitation check		SMC_CheckLimits0(Axis:= , bEnable:= , bCheckVel:= , bCheckAccDec:= , bBusy=> , bError=> , iErrorID=> , bLimitsExceeded=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_REF_SM3		
Input	bEnable	Execution	BOOL		TRUE: Execution check in progress.

	bCheckVel	Speed Check	BOOL	TRUE	TRUE: performs a speed check. false: Do not perform speed checks.
	bCheckAccDec	Acceleration and deceleration check	BOOL	**FALSE**	TRUE:Perform acceleration and deceleration check. false:No acceleration/deceleration check is performed.
Output	bBusy	Under implementation	BOOL		True: performs axis checks. False: do not perform axis checking.
	bError	Error	BOOL		True, the exception is generated.
	iErrorID	Error Code	SMC_ ERROR		Reference SMC_Error
	bLimits Exceeded	Check limits Output	BOOL		TRUE: The current set speed or acceleration/deceleration exceeds Axis.fSWMaxVelocity. Axis.fSWMaxAcceleration Axis.fSWMaxDeceleration

（3） Function description

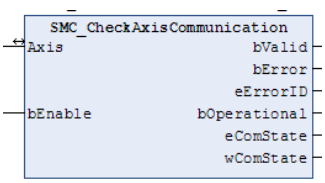
This function block can be used to check if the current setpoint of the drive exceeds the maximum value configured in the controller.

The check result will be displayed by output b limit.

4. 2. 24 Axis Limit Check SMC_CheckAxisCommunication

Detects the current axis status and returns the current communication status of the axis.

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC_CheckLimits	Shaft limitation check		SMC_CheckAxisCommunication0(Axis:= , bEnable:= , bValid=> , bError=> , eErrorID=> , bOperational=> , eComState=> , wComState=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Comment
Inout	Axis	Shaft	AXIS_ REF_SM3	Mapping to an axis, an instance of AXIS_REF_SM3.
Input	bEnable	Execution	BOOL	TRUE: Execution check in progress
Output	bValid	Under implementation	BOOL	True, command execution is valid
	bError	Error	BOOL	True, exception generated
	eErrorID	Error Code	SMC_ERROR	Reference SMC_Error
	bOperational	Communication is normal	BOOL	True, communication is normal (code 100) operable. False, the communication is not normal, and the axis cannot be operated.
	eComState	Communication Status	SMC_ Communication State	True, the command is valid.
	wComState	Communication Code	WORD	True, the exception is generated.

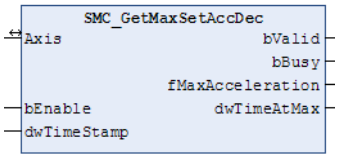
（3） Function description

Detects the current axis status and returns the current communication status of the axis.

4. 2. 25 Axis degree maximum acceleration/deceleration command SMC_GetMaxSetAccDec

This function block can be used to measure the maximum absolute value of shaft acceleration (or deceleration)

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC_GetMaxSetAccDec	Axiality Maximum acceleration and deceleration		SMC_GetMaxSetAccDec_0(Axis:= , bEnable:= , dwTimeStamp:= , bValid=> , bBusy=> , fMaxAcceleration=> , dwTimeAtMax=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_REF_SM3		Mapped to axis, an instance of AXIS_REF_SM3
Input	bEnable	Execution	BOOL	FALSE	An optional timestamp input. It can be used to check. Find the largest The situation that occurs when the value.

Scope	Name	Chinese Name	Type	Initial	Comment
Input	dwTime Stamp	bValid	DWORD		True, the command is valid.
Output	bValid	bBusy	BOOL	FALSE	True, the exception is generated.
	bBusy	fMaxAcceleration	BOOL		True, the command is valid.
	fMax Acceleratin	dwTimeAtMax	LREAL	0	Maximum acceleration and deceleration values (positive for acceleration, negative for deceleration, absolute value of acceleration and deceleration; maximum value is the final value)
	dwTime At Max	bValid	DWORD	0	The dwTimeStamp value corresponding to the maximum acceleration

					and deceleration (e.g., plus As the speed continues to increase, the value increases more with dwTimeStamp, the fMaxAcceleration value is also updated, the Once the acceleration reaches its maximum value, fMaxAcceleration records the maximum value, while the dwTimeStamp corresponding to the maximum value is also recorded)

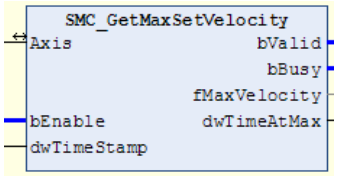
（3） Function description

This function block can be used to measure the maximum absolute value of the axis acceleration (or deceleration). If bEnable is true, it will be reset to 0. If bEnable is wrong, the measurement will be taken. With dwTimeStamp you can provide a call counter, which is locked to the output at dwTimeAtMax with a new maximum value.

4. 2. 26 Axis degree maximum acceleration/deceleration command SMC_GetMaxSetVelocity

This function block can be used to measure the maximum value of the axis speed.

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC_ etMax SetVelocity	Maximum acceleration and deceleration of axial degrees		SMC_GetMaxSetVelocity(Axis:= , bEnable:= , dwTimeStamp:= , bValid=> , bBusy=> , fMaxVelocity=> , dwTimeAtMax=>);

（2）Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_REF_SM3		Mapped to axis, an instance of AXIS_REF_SM3
Input	bEnable	Execution	BOOL	FALSE	Optional timestamp input: can be used to find out what happened at the time of the maximum value.
	dwTimeStamp	Effective	DWORD		Optional time stamp input: can be used to find out when the maximum occurred.
Output	bValid	Under implementation	BOOL	FALSE	True, the command is valid.
	bBusy	Maximum Acceleration value	BOOL		True, the exception is generated.
	fMaxVelocity	Maximum value for Time Stamp	LREAL	0	Maximum velocity value (positive for forward, negative for reverse, absolute maximum value is the final value)
	dwTimeAtMax	Effective	DWORD	0	The dwTimeStamp value corresponding to the maximum speed (e.g., the value changes as the speed continues to increase) with dwTimeStamp, fMaxVelocity The value is also updated once the speed reaches the maximum value, then fMaxVelocity records the maximum value, while the dwTimeStamp corresponding to the maximum value (Also recorded)

（3）Function description

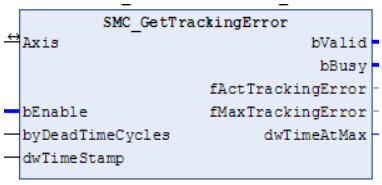
This function block can be used to measure the maximum value of the axis speed. If bEnable is true, it will be reset to 0. If bEnable is wrong, the measurement will be taken. With dwTimeStamp you can provide a call counter, which is locked to the output at dwTimeAtMax with a new maximum value.

4. 2. 27 Axis lag deviation read command SMC_GetTrackingError

This function block can be used to measure current and maximum delay errors to compensate for dead time, which can be generated via fieldbus communication and will be given by the number of cycles (by DeadTimeCycles).

Like SMC_GetMaxSetVelocity timestamp (dwTimeSatamp) can be used to measure the maximum value of time.

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC_ GeTrackingError	Axial lag Deviation reading		SMC_GetTrackingError(Axis:= , bEnable:= , byDeadTimeCycles:= , dwTimeStamp:= , bValid=> , bBusy=> , fActTrackingError=> , fMaxTrackingError=> , dwTimeAtMax=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_ REF_SM 3		Mapping to an axis, an instance of AXIS_REF_SM3.
Input	bEnable	Execution	BOOL	**FALSE**	TRUE: Execute the read.
	byDead TimeCycles		BYTE	2	The number of dead cycles, bEnable triggers a delay of how many dwTimeStamp values to start the lag detection.
	dwTimeStamp		DWORD		Optional timestamp input: can be used to find out what happened at the time of the maximum value.
Output	bValid	Effective	BOOL		True, the command is valid.
	bBusy	Under implementation	BOOL		True, the exception is generated.
	fActTracking Error	Current lag	LREAL	0	The current deviation detection associated with the byDeaTimeCycles value.

	fMaxTracking Error	Maximum lag	LREAL	0	The current deviation value (command position, deviation from the feedback position).
	dwTimeAtMax	The maximum value corresponds to the timestamp	DWORD	0	Maximum deviation value (positive for lag, negative for overrun, absolute value Maximum value is the final value) Note: byDeaTimeCycles value affects this value.

（3） Function description

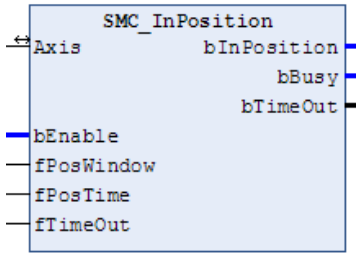
This function block can be used to measure current and maximum delay errors to compensate for dead time, which can be generated via fieldbus communication and will be given by the number of cycles (by DeadTimeCycles).

Similar to SMC_GetMaxSetVelocity timestamp (dwTimeSatamp) can be used to measure the maximum value of time.

4. 2. 28 Axis deviation monitoring SMC_InPosition

This function block monitors the offset between the nominal and actual positions of the axes (drag error). This tests whether the drag error is within the specified value for the specified period (position window).

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC_InPosition	Axis deviation monitoring		SMC_InPosition0(Axis:=Axis , bEnable:= , fPosWindow:= , fPosTime:= , fTimeOut:= , bInPosition=> , bBusy=> , bTimeOut=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Comment
Inout	Axis	Shaft	AXIS_ REF_SM3	Mapped to an axis, an instance of AXIS_REF_SM3.

Input	bEnable	Execution	BOOL	Set the window for deviation monitoring, fPosWindow>Distance (Deviation between command position and feedback position), then bInPosition is TRUE according to the fPosTime time.
	fPos Window	Deviation Window	LREAL	The deviation in the window range time used to trigger bInPosition. in S (seconds).
	fPosTime	Trigger time	LREAL	Deviation timeout. Unit is S (seconds).
	fTimeOut	Timeout time	LREAL	Set the window for deviation monitoring, fPosWindow>Distance (deviation between command position and feedback position), then output bInPosition as TRUE according to fPosTime time.
Output	bInPosition	Normal deviation	BOOL	True, the deviation is within the setting window.
	bBusy	Under implementation	BOOL	True, execution in progress.
	bTimeOut	Timeout	BOOL	The current deviation detection associated with the byDeaTimeCycles value.

（3） Function description

This function block monitors the offset between the nominal and actual positions of the axes (drag error). This tests whether the drag error is within the specified range of values for the specified period (position window). This tests whether the drag error is within the specified range of values for the specified period (position window).

When using logical axes, you can use function blocks to compare the true and logical axes. This tests whether the logical axis (the real axis plot compensated by the communication time) is close to the real axis value. The deviation from the real position of the logical axis is not the deviation between the nominal position and the real position but is used as the nominal position of the real axis.

- The function block SMC_InPosition can be used for this test under the following conditions:

a. The movement type must be the same (limited/modular).

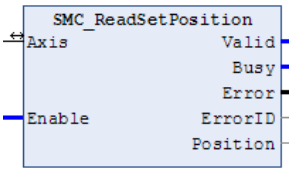
b. In the case of "Modulo", the duration of the position must be the same.

c. The same offset must be defined (e.g., set by MC_SetPosition).

4. 2. 29 Read axis command position SMC_ReadSetPosition

This function block can be used to read the current set position of the drive

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC_ReadSetPosition	Reading axis refers to Order Position		SMC_ReadSetPosition0(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , Position=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_ REF_SM3		Mapped to an axis, an instance of AXIS_REF_SM3.
Input	Enable	Execution	BOOL	FALSE	TRUE: Enables the execution of the function block.
Output	Valid	Effective	BOOL	FALSE	TRUE: Parameter is available.
	Busy	Under implementation	BOOL	FALSE	TRUE: Execution of function block has not been finished yet.
	Error	Error	BOOL	FALSE	TRUE: Error has occurred within the function block.
	ErrorID	Error Code	SMC _ERROR	0	Error identification
	Position	Command Location	LREAL	0	Position of drive

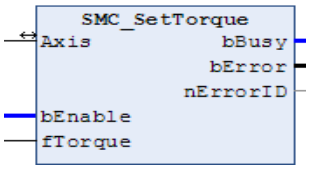
（3） Function description

This function block can be used to read the current set position of the drive.

4. 2. 30 Torque setting command SMC_SetTorque

If the drive is in controller mode "Torque", this function block can be used to create a torque.

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC_SetTorque	Torque setting		SMC_SetTorque0(Axis:= , bEnable:= , fTorque:= , bBusy=> , bError=> , nErrorID=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_ REF_SM3		Mapped to the axis, an instance of AXIS_REF_SM3
Input	bEnable	Execution	BOOL	FALSE	Rising edge, setting axis torque.
	fTorque	Set torque	LREAL		The unit is 0.1%.
Output	bBusy	Under implementation	BOOL	FALSE	True, execution in progress.
	bError	Error	BOOL	FALSE	True, the exception is generated.
	nErrorID	Error Code	SMC_ ERROR	0	Refer to SMC_Error.

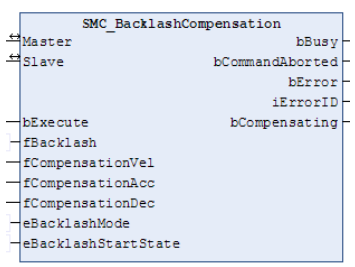
（3） Function description

If the drive is in controller mode "Torque", this function block can be used to create a torque.

4. 2. 31 Gap Compensation Command SMC_BacklashCompensation

This function block can be used to compensate for bounces that occur in belt drives or gearboxes.

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC_ BacklashCompensation	Gap compensation		SMC_BacklashCompensation0(Master:= , Slave:= , bExecute:= , fBacklash:= , fCompensationVel:= , fCompensationAcc:= , fCompensationDec:= , eBacklashMode:= , eBacklashStartState:= , bBusy=> , bCommandAborted=> , bError=> , iErrorID=> , bCompensating=>);

(2) Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Master	Main shaft	AXIS_REF_SM3		Mapped to axis, an instance of AXIS_REF_SM3
	Slave	From the shaft	AXIS_REF_SM3		Mapped to the axis, an instance of AXIS_REF_SM3
Input	bExecute	Execution	BOOL		Rising edge, set offset
	fBacklash		LREAL		Compensation gap
	fCompensationVel		LREAL		Speed at compensation
	fCompensationAcc		LREAL		Acceleration at compensation
	fCompensation Dec		LREAL		Deceleration during compensation
	fCompensation Jerk		LREAL		Compensation model: SMC_BL_AUTO. The direction of spindle operation determines the direction of compensation. SMC_BL_POSITIVE: forward compensation, independent of the direction of spindle operation. SMC_BL_NEGATIVE: Reverse compensation, independent of the main Direction of axis operation. SMC_BL_OFF: no compensation;

Scope	Name	Chinese Name	Type	Initial	Comment
Input	eBacklashMode		SMC_BACKLASH_MODE	SMC_BL_AUTO	Describes the operating state of the axis when this command is operating. SMC_BL_START_NE GATIVE: from axis on the negative side Movement under traction, in negative No compensation is needed under the directional motion, once the forward motion starts with Two times fBacklash build Compensation: SMC_BL_START_POSITIVE: The slave axis moves in the positive direction with traction in the positive direction. No compensation is required for reverse motion, once the reverse motion is established with twice the fBacklash. SMC_BL_START_NONE: Distance compensation for fBacklash values generated by motion in the positive or negative direction.

Scope	Name	Chinese Name	Type	Initial	Comment
	eBacklash StartState		SMC_BACKLASH_STARTSTATE	SMC_BL_START_NONE	Rising edge, set offset.
Output	bBusy	Under implementation	BOOL		True, execution in progress.
	bCommand Aborted	Command Interrupted	BOOL		True- Interrupted by other control commands.
	bError	Error	BOOL		True, abnormal birth

	iErrorID	Error Code	SMC _ERROR		Reference SMC_Error
	bCompensating	Compensation in progress	BOOL		TRUE: While compensating a backlash.

（3） Function description

This function block can be used to compensate for bounces that occur in belt drives or gearboxes.

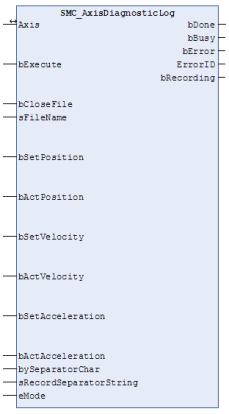
For this purpose, a (usually virtual) spindle is mirrored on a real slave axis whose bounce should be compensated. The function block works similarly to the phase function block, the phase depends on the direction of the spindle If the spindle starts in a certain direction, the slave axis makes the same plus an additional movement in this direction (distance: fBacklash): If the spindle is inverted, the slave axis also turns and covers an additional double distance fBacklash in this new direction.

When the module starts working, make sure that the spindle and slave axis are in the same position, otherwise the slave axis will be located on the spindle.

4. 2. 32 Axis parameters are written to the file SMC_AxisDiagnosticLog

This function block is used to periodically write a set of parameter values belonging to one axis of the file.

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC_ AxisDiagnosticLog	Axis parameters Write to file		SMC_AxisDiagnosticLog(Axis:= , bExecute:= , bCloseFile:= , sFileName:= , bSetPosition:= , bActPosition:= , bSetVelocity:= , bActVelocity:= , bSetAcceleration:= , bActAcceleration:= , bySeparatorChar:= , sRecordSeparatorString:= , eMode:= , bDone=> , bBusy=> , bError=> , ErrorID=> , bRecording=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_ REF_SM3		mapped to the axis, an instance of AXIS_REF_SM3
Input	bExecute	Execution	BOOL		Rising edge, executing function blocks.
	bCloseFile	Close file	BOOL		TRUE, the command closes the file immediately.
	sFileName	File name	STRING (255)		The name of the stored file (before the path.)
Input	bSetPosition	Record set position	BOOL		TRUE, the set position is recorded when the command is executed
	bActPosition	Record the actual location	BOOL		TRUE, the actual position is recorded when the command is executed
	bSetVelocity	Record set speed	BOOL		TRUE, the set speed is recorded when the command is executed

Scope	Name	Chinese Name	Type	Initial	Comment
Input	bActVelocity	Several rounds of actual speed	BOOL		TRUE, the actual speed is recorded when the command is executed
	bSet Acceleration	Record set acceleration	BOOL		TRUE, the set acceleration is recorded when the command is executed
	bAct Acceleration	Recording actual acceleration	BOOL		TRUE, the actual speed is recorded when the command is executed
	bySeparator Char		BYTE	9	ASCII code value, written between two different values
	sRecord Separator String		STRING (3)	'RN'	End of date written string
	eMode		SMC_ LOGGER MODE	LOG_ CONTI NUOUS	log_continuous: even Continue logging to file log_at_close: continuous Record to buffer (10kbyte). When bclosefile is true Writes the data in the buffer to the file.
Output	bDone	Completion	BOOL		True, save is complete.
	bBusy	Under implementation	BOOL	FALSE	True, execution in progress.
	bError	Error	BOOL		True, the exception is generated.
	ErrorID	Error Code	SMC_ ERROR		Refer to SMC_Error.
	bRecording	On the Record	BOOL	FALSE	TRUE: Module records.

（3） Function description

This function block is used to periodically write a set of parameter values belonging to one axis of the file. This output file is well suited for diagnostic purposes. Since it usually takes some time to write data on the data medium, this block stores the collected data in a buffer of size 10 kByte and does not write the data until the operation called WriteToFile is performed by the module.

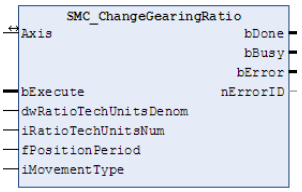
This action call should be placed in a slower (~50 ms) lower priority task to prevent interference with the actual action task as well as the motion itself. Once the buffer is exceeded, the module will create an error output.

4. 2. 33 Change Gearing Ratio SMC_ChangeGearingRatio

With the help of this function, gear ratios and drive types can be prevented from being modified.

After execution, the axis must be restarted by SMC3_ReinitDrive to ensure that all variables are initialized in the correct way.

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC_ChangeGearingRatio	Change Gear Ratio		SMC_ChangeGearingRatio0(Axis:= , bExecute:= , dwRatioTechUnitsDenom:= , iRatioTechUnitsNum:= , fPositionPeriod:= , iMovementType:= , bDone=> , bBusy=> , bError=> , nErrorID=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Comment
Inout	Axis	Shaft	AXIS _REF_SM3	Mapped to axis, an instance of AXIS_REF_SM3. The gear ratio will be changed by the shaft.
Input	bExecute	Execution	BOOL	Rising edge, execution function block
	dwRatioTechUnits Denom		DWORD	Conversion of pulse units to application units (eg:mm)
	iRatioTech UnitsNum		DINT	The dwRatioTechUnitsDenom value corresponds to the required application unit
	fPositionPeriod		LREAL	Position cycle time (modulus value), valid only for rotating motors

	iMovementType		SMC_ MOVEMENTTY PE	0: Modulo axis. 1: Finite axis
Output	bDone	Completion	BOOL	True, the execution of the setting is complete
	bBusy	Under implementation	BOOL	True, in progress
	bError	Error	BOOL	True, exception generated
	nErrorID	Error Code	SMC_ERROR	Reference SMC_Error

（3） Function description

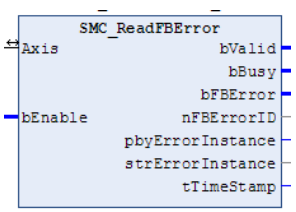
With the help of this function, gear ratios and drive types can be prevented from being modified.

After execution, the axis must be restarted by SMC3_ReinitDrive to ensure that all variables are initialized in the correct way.

4. 2. 34 Read Function Block Error Command SMC_ReadFBError

This function block is used to read the oldest information about function block errors.

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC_ReadFBError	Read function block error		SMC_ReadFBError(Axis:= , bEnable:= , bValid=> , bBusy=> , bFBError=> , nFBErrorID=> , pbyErrorInstance=> , strErrorInstance=> , tTimeStamp=>);

（2） Relevant variables

Scope	Name	Chinese Name	Type	Initial	Comment
Inout	Axis	Shaft	AXIS_REF_SM3		Mapped to axis, an instance of AXIS_REF_SM3
Input	bEnable	Execution	BOOL	FALSE	TRUE: Execute the read.
Output	bValid	Effective	BOOL	FALSE	True, the read is valid.
	bBusy	Under implementation	BOOL		True, execution in progress.
	bFBError	Error	BOOL	FALSE	True, there are FB errors generated.
	nFBErrorID	Error Code	SMC_ERROR	0	Reference SMC_Error
	pbyErrorInstance		POINTER TO BYTE		The function block of the output point reports an error.
	strErrorInstance		STRING		Pointing to the wrong function block (program, subroutine, function block)
	tTimeStamp		TIME		The timestamp when the error occurred.

（3） Function description

This function block is used to read the oldest information about function block errors.

4. 2. 35 Read Function Block Error Command SMC_ClearFBError

This function is used to delete the oldest function block error message.

（1） Command format

Command	Name	Graphical representation	ST Performance
SMC_ClearFBError	Read function Block Error		TEST:=SMC_ ClearFBError (pDrive:=ADR(Axis)).

（2） Relevant variables

Scope	Name	Chinese Name	Type	Ccomment
Return	SMC_ ClearFBError	Shaft	BOOL	Mapped to axis, an instance of AXIS_REF_SM3
Input	pDrive	Clear errors	POINTER TO AXIS_REF_SM3	True, Clear

（3） Function description

This function is used to delete the oldest function block error message.

Revision: V1.5

HMC

AUCTECH AUTOMATION

Guangzhou Auctech Automation Technology Ltd

ADD: Hongshi Business Building, 11 Kehua Road, SCI-TECH Industry Park,
Taihe Town, Baiyun District, Guangzhou, CHINA

Tel	Website	Mail
+86 20 84898493	www.auctech.com.cn	info@auctech.com.cn