AX Series
Programmable Controller
Programming Manual

SHENZHEN INVT ELECTRIC CO., LTD.

Preface

Thank you for choosing AX series programmable controller (programmable controller for short).

This manual contains the information necessary to use the programmable controller. Please read this manual carefully before using the product. Then you can fully understand the functions, performance, and system build-up, which helps to give full play to the advanced performance.

Target audience

Personnel with electrical professional knowledge (such as qualified electrical engineers or personnel with equivalent knowledge)

Applicable product

AX70 programmable controller

AX71 programmable controller

Online support

You can also obtain product documentation and technical support from INVT website: invt.com

If the product is ultimately used for military affairs or weapon manufacture, comply with the export control regulations in the Foreign Trade Law of the People's Republic of China and complete related formalities.

The manual is subject to change without prior notice.

Contents

Preface ..... **i**

Target audience i

Applicable product i

Online support i

Contents ..... **ii**

1 Basic commands **1**

1.1 Description of commands in standard.lib 1

1.1.1 Strings 1

1.1.2 Bistable function blocks 4

1.1.3 Triggers 5

1.1.4 Counters 6

1.1.5 Timers 8

1.2 Description of commands in Util.lib 11

1.2.1 BCD conversion 11

1.2.2 Bit/byte functions 12

1.2.3 Mathematical auxiliary function blocks 13

1.2.4 Regulators 15

1.2.5 Signal generator 19

1.2.6 Operation function block 22

1.2.7 Analog value processing 24

2 Motion control commands **26**

2.1 Description of the SM3_Basic library 26

2.1.1 MC_Power 26

2.1.2 MC_Halt 27

2.1.3 MC_Home 28

2.1.4 MC_MoveAbsolute 30

2.1.5 MC_AccelerationProfile 33

2.1.6 MC_MoveAdditive 35

2.1.7 MC_MoveRelative 37

2.1.8 MC_MoveSuperImposed 40

2.1.9 MC_MoveVelocity 43

2.1.10 MC_PositionProfile 45

2.1.11 MC_ReadActualPosition 46
2.1.12 MC_ReadBoolParameter 48
2.1.13 MC_ReadAxisError 49
2.1.14 MC_ReadStatus 50
2.1.15 MC_ReadParameter 52
2.1.16 MC_Reset 54
2.1.17 MC_Stop 55
2.1.18 MC_VelocityProfile 57
2.1.19 MC_WriteBoolParameter 59
2.1.20 MC_WriteParameter 61
2.1.21 MC_AbortTrigger 62
2.1.22 MC_ReadActualTorque 64
2.1.23 MC_ReadActualVelocity 65
2.1.24 MC_SetPosition 66
2.1.25 MC_TouchProbe 68
2.1.26 MC_MoveContinuousAbsolute 70
2.1.27 MC_MoveContinuousRelative 72
2.1.28 MC_Jog 74
2.1.29 MC_Inch 76
2.1.30 SMC3_PersistPosition 78
2.1.31 SMC3_PersistPositionSingleturn 80
2.1.32 SMC3_PersistPosition Logical 82
2.1.33 SMC_Homing 84
2.1.34 MC_Camin 89
2.1.35 MC_Camout 97
2.1.36 MC_CamTableSelect 99
2.1.37 MC_GearIn 101
2.1.38 MC_GearOut 103
2.1.39 MC_GearInPos 105
2.1.40 MC_Phasing 109
3 INVT commands 112
3.1 Communication commands 112
3.1.1 ModbusRTU master node 112
3.1.2 ModbusRTU slave node 115
3.1.3 ModbusTCP master node 116

3.1.4 ModbusTCP slave node 119
3.2 Pulse commands 120
3.2.1 MC_InitSys_Invt 120
3.2.2 MC_Axis 121
3.2.3 MC_Power_Invt 123
3.2.4 MC_Stop_Invt 124
3.2.5 MC_Reset_Invt 125
3.2.6 MC_JOG_Invt 126
3.2.7 MC_JOGP_Invt 127
3.2.8 MC_MoveAbsolute_Invt 129
3.2.9 MC_MoveRelative_Invt 130
3.2.10 MC_MoveVelocity_Invt 132
3.2.11 MC_ReadActualPosition_Invt 133
3.2.12 MC_ActiveSpeed_Invt 134
3.2.13 MC_HOME_Invt 135
3.2.14 AMC_Reset_Invt 136
3.2.15 AMC_Stop_Invt 137
3.2.16 AMC_ActiveSpeed_Invt 139
3.2.17 AMC_MoveAbsolute_Invt 140
3.2.18 AMC_MoveRelative_Invt 142
3.2.19 AMC_ArcDel_Invt 144
3.2.20 MC_SendData_Invt 146
3.2.21 MC_GetVersion_Invt 147
3.2.22 MC_SetPosition_Invt 147
3.2.23 MC_ReadParameter_Invt 148
3.2.24 MC_ReadStatus_Invt 150
3.3 Enhanced commands 152
3.3.1 FB_ReinitDrv 152
3.3.2 FB_AxisEasyControl 154
3.3.3 FB_Relative_INVT 156

1 Basic commands

1.1 Description of commands in standard.lib

By default, Invtmatic Studio provides the library **standard.lib**, including all standard functions and function modules compliant with IEC61131-3. Though the development system can identify operators, standard modules are definitely imported to a project as a library.

1.1.1 Strings

1.1.1.1 LEN

This function is used to obtain the length of a character string. The input variable STR is of the STRING type, and the return value is of the INT type.

Example in FBD:

Example in ST:

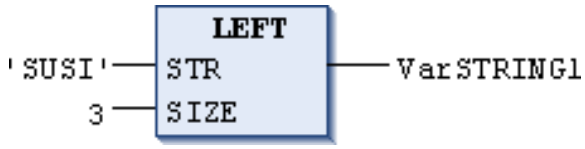
```
VarINT1 := LEN ('SUSI');
```

1.1.1.2 LEFT

This function is used to obtain certain characters from the left of a source character string. The input variable STR is of the STRING type, the input variable SIZE is of the INT type, and the return value is of the STRING type.

LEFT (STR, SIZE) is used to obtain the characters with the length specified by SIZE, starting from the left of the character string STR.

Example in FBD:



Example in ST:

```
VarSTRING1 := LEFT ('SUSI',3);
```

1.1.1.3 RIGHT

This function is used to obtain certain characters from the right of a source character string. The input variable STR is of the STRING type, the input variable SIZE is of the INT type, and the return value is of the STRING type.

RIGHT (STR, SIZE) is used to obtain the characters with the length specified by SIZE, starting from the right of the character string STR.

Example in FBD:

Example in ST:

```
VarSTRING1 := RIGHT ('SUSI',3);
```

1.1.1.4 MID

This function is used to obtain certain characters from a source character string. The input variable STR is of the STRING type, the input variables LEN and POS are of the INT type, and the return value is of the STRING type.

MID (STR, LEN, POS) is used to obtain the characters with the length specified by LEN, starting from the character with the position specified by POS of the character string STR.

Example in FBD:

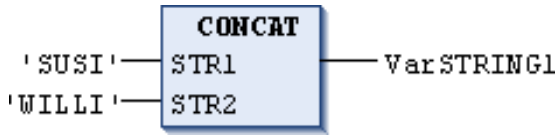
Example in ST:

```
VarSTRING1 := MID ('SUSI',2,2);
```

1.1.1.5 CONCAT

This function is used to combine two character strings in series. The input variables STR1 and STR2, and the return value are of the STRING type.

Example in FBD:



Example in ST:

```
VarSTRING1 := CONCAT ('SUSI','WILLI');
```

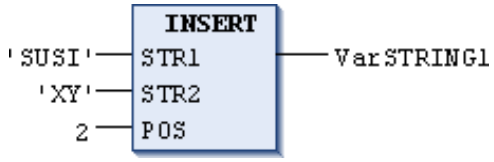
1.1.1.6 INSERT

This function is used to insert another character string at a specified position into a source character string.

The input variables STR1 and STR2 are of the STRING type, the input variable POS is of the INT type, and the return value is of the STRING type.

INSERT(STR1, STR2, POS) is used to insert the character string STR2 next to the position specified by POS into the character string STR1.

Example in FBD:



Example in ST:

```
VarSTRING1 := INSERT ('SUSI','XY',2);
```

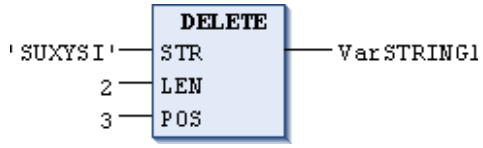
1.1.1.7 DELETE

This function is used to delete specified characters from a specified position of a source character string.

The input variable STR is of the STRING type, the input variables LEN and POS are of the INT type, and the return value is of the STRING type.

DELETE (STR, L, POS) is used to delete certain characters from the character string STR, while L specifies the length of characters to be deleted and POS specifies the character deletion start position.

Example in FBD:



Example in ST:

```
Var1 := DELETE ('SUXYSI',2,3);
```

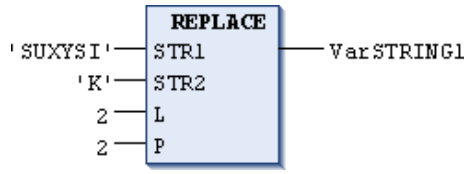
1.1.1.8 REPLACE

This function is used to replace certain characters at a specified position of a source character string with another given character string.

The input variables STR1 and STR2 are of the STRING type, the input variables L and P are of the INT type, and the return value is of the STRING type.

REPLACE(STR1, STR2, L, P) is used to replace certain characters with the character string STR2 for the character string STR1, while L specifies the length of characters to be replaced and P specifies the character replacing start position.

Example in FBD:



Example in ST:

```
VarSTRING1 := REPLACE ('SUXYSI','K',2,2);
```

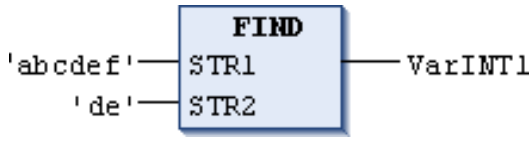
1.1.1.9 FIND

This function is used to search a character string for certain characters. The input variables STR1 and STR2 are of the

STRING type, and the return value is of the INT type.

FIND(STR1, STR2) is used to find where STR2 occurs in STR1 for the first time. If STR2 is not found in STR1, the message is displayed: "OUT:=0"

Example in FBD:



Example in ST:

```
arINT1 := FIND ('abcdef','de');
```

1.1.2 Bistable function blocks

1.1.2.1 SR

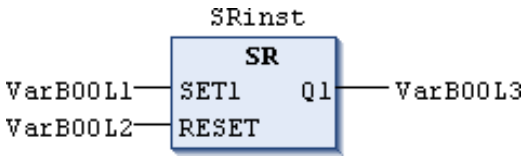
This function module gives priority to setting. Q1 = SR (SET1, RESET) equals Q1 = (NOT RESET AND Q1) OR SET1.

The input variables SET1 and RESET, and the output variable Q1 are of the BOOL type.

Declaration example:

```
SRInst : SR ;
```

Example in FBD:



Example in ST:

```
SRInst(SET1:= VarBOOL1 , RESET:=VarBOOL2 );

VarBOOL3 := SRInst.Q1 ;
```

1.1.2.2 RS

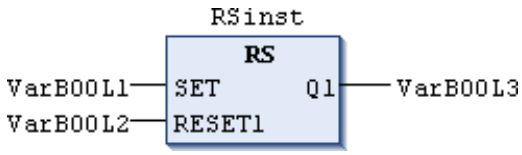
This function module gives priority to resetting. Q1 = RS (SET, RESET1) equals Q1 = NOT RESET1 AND (Q1 OR SET).

The input variables SET1 and RESET, and the output variable Q1 are of the BOOL type.

Declaration example:

```
RSInst : RS ;
```

Example in FBD:



Example in ST:

```
RSInst(SET:= VarBOOL1 , RESET1:=VarBOOL2 );

VarBOOL3 := RSInst.Q1 ;
```

1.1.3 Triggers

1.1.3.1 R_TRIG

This function block is used to detect the rise edge.

For input, `CLK: BOOL;` The boolean input signal is used to detect the rise edge.

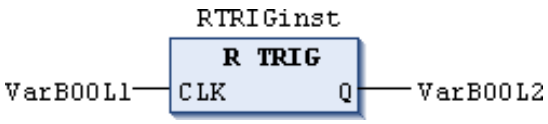
For output, `Q: BOOL;` If CLK has detected the rise edge, the output is True.

Only if the value of the input variable CLK is False, the values of the output variable Q and the medium auxiliary variable M are always False. When the value of CLK changes to True, the value of Q changes to True first and then that of M is set to True. In a word, every time when the function block is invoked, Q returns False if CLK changes from the rise edge to the fall edge.

Declaration example:

```
RTRIGInst : R_TRIG ;
```

Example in FBD:



Example in ST:

```
RTRIGInst(CLK:= VarBOOL1);

VarBOOL2 := RTRIGInst.Q;
```

1.1.3.2 F_TRIG

This function block is used to detect the fall edge.

For input, `CLK: BOOL;` The boolean input signal is used to detect the fall edge.

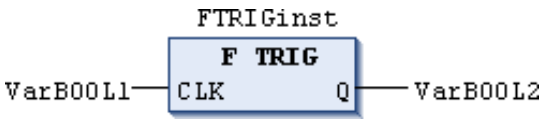
For output, `Q: BOOL;` If CLK has detected the fall edge, the output is True.

Only if the value of the input variable CLK is True, the values of the output variable Q and the medium auxiliary variable M are always False. When the value of CLK changes to False, the value of Q changes to True first and then that of M is set to True. In a word, every time when the function block is invoked, Q returns False if CLK changes from the fall edge to the rise edge.

Declaration example:

```
FTRIGInst : F_TRIG ;
```

Example in FBD:



Example in ST:

```
FTRIGInst(CLK:= VarBOOL1);

VarBOOL2 := FTRIGInst.Q;
```

1.1.4 Counters

1.1.4.1 CTU

This function block is used to count up.

For input:

`CU: BOOL;` The rise edge triggers the counting up of CV.

`RESET: BOOL;` When the value is TRUE, CV is reset to 0.

`PV: WORD;` It indicates the upper limit of CV counting.

For output:

`Q: BOOL;` When CV reaches the counting upper limit PV, the value is TRUE.

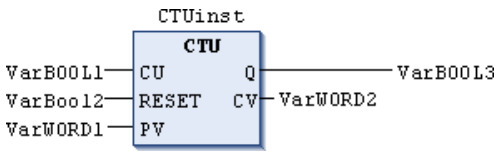
`CV: WORD;` The value is continuously added by 1 until it reaches CV.

If the value of RESET is TRUE, the counting value CV is initialized to 0. If there is a rise edge when CU changes from FALSE to TRUE, the value of CV is added by 1. If CV is greater than or equal to the counting upper limit PV, the value of Q is TRUE.

Declaration example:

```
CTUInst :CTU ;
```

Example in FBD:



Example in ST:

```
CTUInst(CU:= VarBOOL1, RESET:=VarBOOL2 , PV:= VarWORD1);

VarBOOL3 := CTUInst.Q ;

VarWORD2 := CTUInst.CV;
```

1.1.4.2 CTD

This function block is used to count down

For input:

`CD: BOOL;` The rise edge triggers the counting down of CV.

`LOAD: BOOL;` When the value is TRUE, CV is set to the upper limit PV.

`PV: WORD;` It indicates the initial value for the counting down of CV.

For output:

`Q: BOOL;` When CV reaches 0, the value is TRUE.

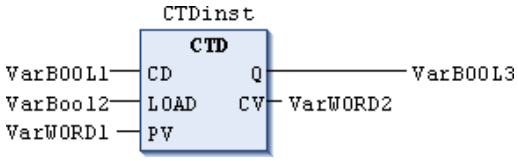
`CV: WORD;` The value is continuously reduced by 1 until it reaches 0.

If the value of LOAD is TRUE, the counting value CV is initialized to the counting upper limit PV. If there is a rise edge when CD changes from FALSE to TRUE and CV is greater than 0, the value of CV is reduced by 1 (that is, the value of CV cannot be less than 0). If CV is 0, the value of Q is TRUE.

Declaration example:

```
CTDInst:CTD ;
```

Example in FBD:



Example in ST:

```
CTDInst(CD:= VarBOOL1, LOAD:=VarBOOL2 , PV:= VarWORD1);

VarBOOL3 := CTDInst.Q ;

VarWORD2 := CTDInst.CV;
```

1.1.4.3 CTUD

This function block is used to count up and down.

For input:

`CU: BOOL;` The rise edge triggers the counting up of CV.

`CD: BOOL;` The rise edge triggers the counting down of CV.

`RESET: BOOL;` When the value is TRUE, CV is reset to 0.

`LOAD: BOOL;` When the value is TRUE, CV is set to the upper limit PV.

`PV: WORD;` It indicates the initial value for the counting up or down of CV.

For output:

`QU: BOOL;` When CV reaches the counting upper limit PV, the value is TRUE.

`QD BOOL;` When CV reaches 0, the value is TRUE.

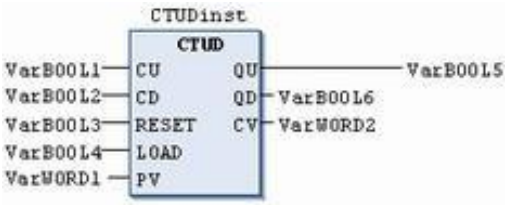
`CV: WORD;` The value is continuously reduced by 1 until it reaches 0.

If there is a rise edge when CU changes from FALSE to TRUE, the value of CV is added by 1. If there is a rise edge when CD changes from FALSE to TRUE and CV is greater than 0, the value of CV is reduced by 1. If CV is greater than or equal to PV, the value of QU is TRUE. If CV is 0, the value of QD is TRUE.

Declaration example:

```
CTUDInst :CUTD ;
```

Example in FBD:



Example in ST:

```
CTUDInst(CU := VarBOOL1, CD:= VarBOOL2, RESET := VarBOOL3, LOAD:=VarBOOL4 , PV:= VarWORD1);
                                VarBOOL5 := CTUDInst.QU ;
                                VarBOOL6 := CTUDInst.QD ;
                                VarWORD2 := CTUDInst.CV;
```

1.1.5 Timers

1.1.5.1 TP

This function block serves as a timer.

For input:

`IN: BOOL;` The rise edge triggers timing on the ET end.

`PT: TIME;` It indicates the upper limit of ET timing period.

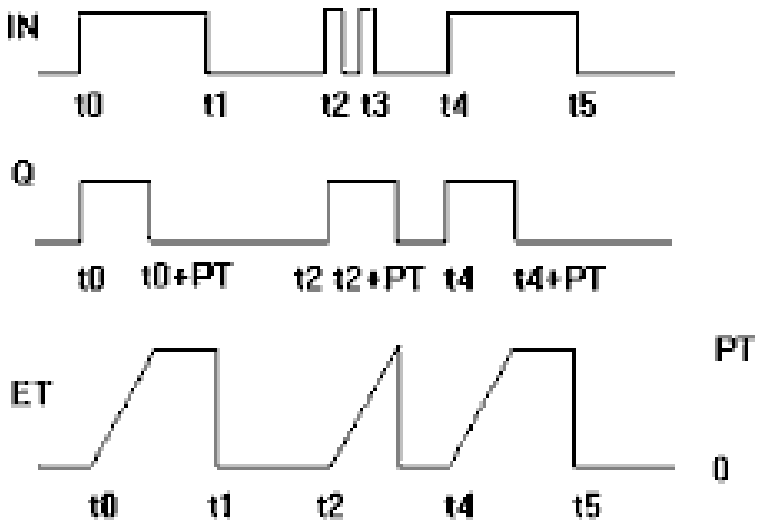
For output:

`Q: BOOL;` When ET is performing timing, the value is TRUE.

`ET: TIME;` It indicates the present status of time.

If the value of IN is FALSE, the value of Q is FALSE, and the value of ET is 0. If the value of IN is TRUE, timing starts on the ET end, which is calculated in millisecond and does not stop until ET is equal to PT. Once when ET is equal to PT, this constant is kept. If IN is TRUE and ET is less than or equal to PT, Q is TURE; otherwise, Q is FALSE.

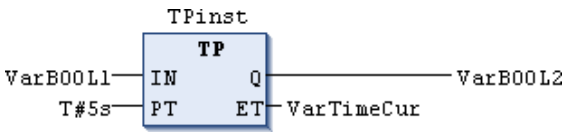
Therefore, Q is TRUE in the timing period defined by PT. The timing sequence diagram of TP is as follows:



Declaration example:

```
TPInst : TP ;
```

Example in FBD:



Example in ST:

```
TPInst(IN := VarBOOL1, PT:= T#5s);

VarBOOL2 :=TPInst.Q;
```

1.1.5.2 TON

This function block serves as a timer that is enabled with a delay.

For input:

`IN: BOOL;` The rise edge triggers timing on the ET end.

`PT: TIME;` It indicates the upper limit (that is, the delay time) of ET timing period.

For output:

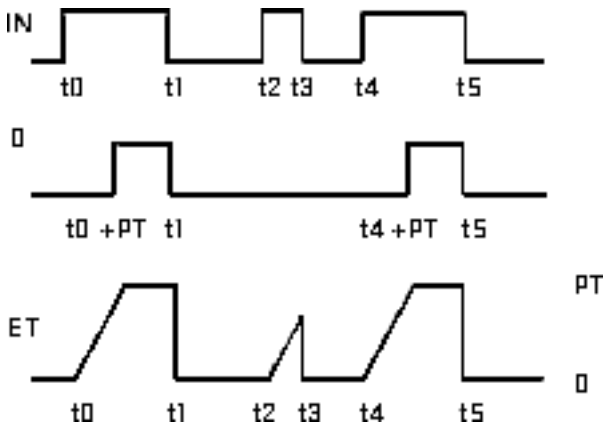
`Q: BOOL;` When the timing of ET reaches the upper limit PT, a rise edge is output.

`ET: TIME;` It indicates the present status of time.

TP(IN, PT, Q, ET): IN is an input variable of the BOOL type and PT is an input variable of the TIME type. Q is an output variable of the BOOL type and ET is an output variable of the TIME type. If the value of IN is FALSE, the value of Q is FALSE, and the value of ET is 0.

If the value of IN is TRUE, timing starts on the ET end, which is calculated in millisecond and does not stop until ET is equal to PT. Once when ET is equal to PT, this constant is kept. If IN is TRUE and ET is equal to PT, Q is TURE. Otherwise, Q is FALSE. Therefore, Q has a rise edge after the delay (that is, the time defined by PT) is completed.

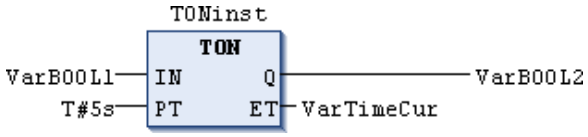
The timing sequence diagram of TON is as follows:



Declaration example:

```
TONInst : TON ;
```

Example in FBD:



Example in ST:

```
TONInst(IN := VarBOOL1, PT:= T#5s);
```

1.1.5.3 TOF

This function block serves as a timer that is disabled with a delay.

For input:

`IN: BOOL;` The fall edge triggers timing on the ET end.

`PT: TIME;` It indicates the upper limit (that is, the delay time) of ET timing period.

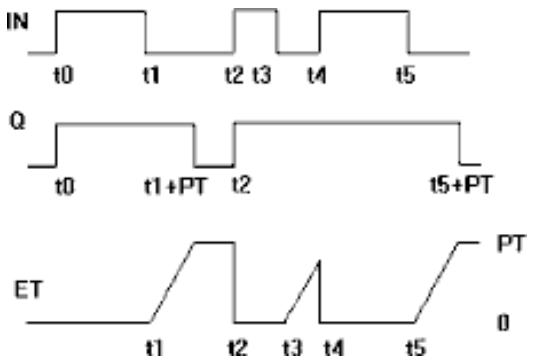
For output:

`Q: BOOL;` When the timing of ET reaches the upper limit PT, a fall edge is output.

`ET: TIME;` It indicates the present status of time.

TOF(IN, PT, Q, ET): If IN is TRUE, Q is TRUE. If the value of IN is FALSE, timing starts on the ET end, which is calculated in millisecond and does not stop until ET is equal to PT. Once when ET is equal to PT, this constant is kept. If IN is FALSE and ET is equal to PT, Q is TURE; otherwise, Q is TRUE. Therefore, Q has a fall edge after the delay is completed.

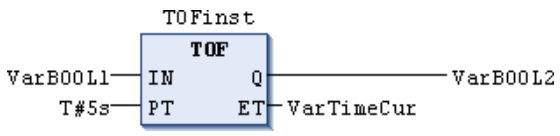
The timing sequence diagram of TOF is as follows:



Declaration example:

```
TOFInst :TOF ;
```

Example in FBD:



Example in ST:

```
TOFInst(IN := VarBOOL1, PT:= T#5s);

VarBOOL2 :=TOFInst.Q;
```

1.1.5.4 RTC

This function block serves as a real-time timer.

For input:

`EN: BOOL;` The rise edge triggers timing on the CDT end.

`PDT:DATE_AND_TIME;` It indicates the start date and time of timing.

For output:

`Q: BOOL;` When the CDT end starts timing, the output is TRUE.

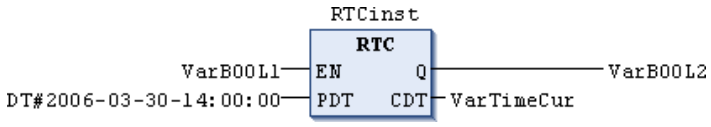
`CDT: DATE_AND_TIME;` It indicates the present date and time of timing.

VarBOOL2:=RTC(EN, PDT, Q, CDT): When EN is FALSE, Q is FALSE, and CDT is DT#1970-01-01-00:00:00. Once when EN changes to TRUE (having a rise edge) and keeps the value, the CDT end starts counting up, using PDT as the initial value, accurate to seconds. When EN is reset to FALSE, CDT is reset to the initial value DT#1970-01-01-00:00:00.

Declaration example:

```
RTCInst:RTC ;
```

Example in FBD:



Example in ST:

```
RTCInst (EN:=VarBOOL1, PDT:=DT#2006-03-30-14:00:00, Q=>VarBOOL2, CDT=>VarTimeCur);
```

1.2 Description of commands in Util.lib

The library Util.lib contains a variety of function blocks for BCD conversion, bit/byte functions, mathematical auxiliary functions, and analog value processing.

1.2.1 BCD conversion

1.2.1.1 BCD_TO_INT

This function is used to convert the bytes in BCD format into integers.

The input variable is of the BYTE type, and the output variable is of the INT type.

If the byte to be converted is not in the BCD format, the output is -1.

Example in ST:

```
i:=BCD_TO_INT(73); (* Result is 49 *)
k:=BCD_TO_INT(151); (* Result is 97 *)
l:=BCD_TO_INT(15); (* Output -1, because it is not in BCD format *)
```

1.2.1.2 INT_TO_BCD

This function is used to convert integers into the bytes in BCD formats.

The input variable is of the INT type, and the output variable is of the BYTE type.

If the integer cannot be converted into the BCD format, the output is 255.

Example in ST:

```
i:=INT_TO_BCD(49); (* Result is 73 *)
k:=BCD_TO_INT(97); (* Result is 151 *)
l:=BCD_TO_INT(100); (* Error! Output: 255 *)
```

1.2.2 Bit/byte functions

1.2.2.1 EXTRACT

In this function, the input variable X is of the DWORD type, and the input variable N is of the BYTE type. The output variable is of the BOOL type, and the output is the digit at bit N of the input variable X, in which N is counted from bit 0.

Example in ST:

```
FLAG:=EXTRACT(X:=81, N:=4);
(* Result : TRUE, because 81 is binary 1010001, so the 4th bit is 1 *)
FLAG:=EXTRACT(X:=33, N:=0);
(* Result : TRUE, because 33 is binary 100001, so the bit '0' is 1 *)
```

1.2.2.2 PACK

This function is used to combine the eight input variables B0, B1,.... and B7 of the BOOL type into a data value of the BYTE type.

The function UNPACK has a tight relationship with the function PACK.

1.2.2.3 PUTBIT

In this function, the input variables X, N, and B are of the DWORD type, BYTE type, and BOOL type respectively.

PUTBIT is used to set the digit at bit N of X to B, in which N is counted from bit 0.

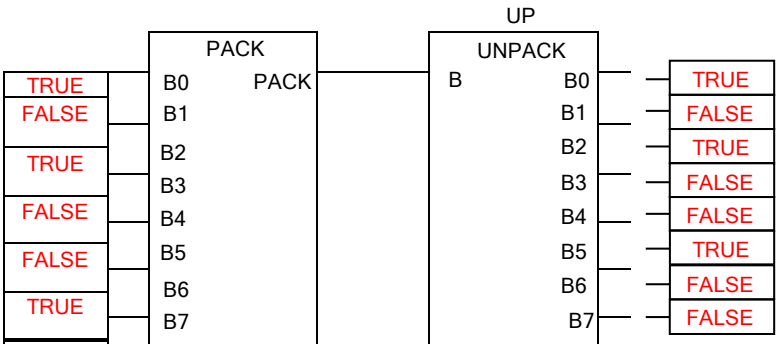
Example in ST:

```
var1:=38; (* binary 100110 *)
var2:=PUTBIT(A,4,TRUE); (* Result: 54 = 2#110110 *)
var3:=PUTBIT(A,1,FALSE); (* Result: 36 = 2#100100 *)
```

1.2.2.4 UNPACK

This function is used to split the input variable B of the BYTE type into eight output variables B0, B1,.... and B7 of the BOOL type. This function is opposite to the function PACK.

Example in FBD:



1.2.3 Mathematical auxiliary function blocks

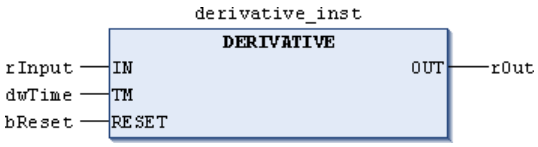
1.2.3.1 DERIVATIVE

This function block is used to determine local approximate derivatives.

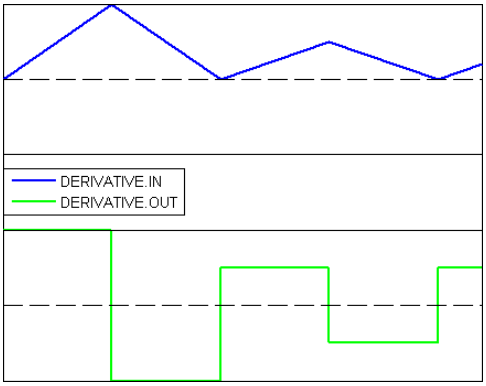
The input variables IN and TM are of the REAL and DWORD types respectively, with the time accurate to milliseconds. RESET is of the BOOL type, and this function block is reset when the value of RESET is TRUE. The output variable OUT is of the REAL type.

To obtain a most accurate result, DERIVATIVE approximately uses the last four values to decrease inaccuracy generated by input parameters.

Example in FBD:



DERIVATIVE input and output:



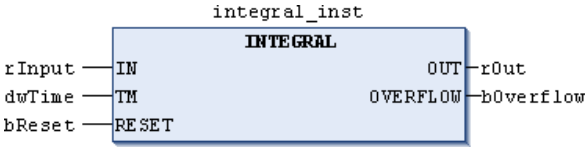
1.2.3.2 INTEGRAL

This function block is used for approximate integral.

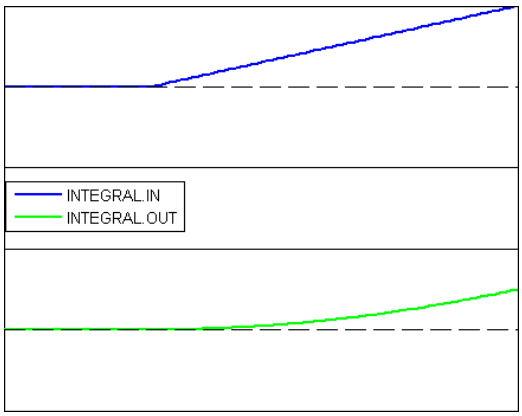
Similar to DERIVATIVE, the input variables IN and TM are of the REAL and DWORD types respectively, with the time accurate to milliseconds. RESET is of the BOOL type, and this function block is reset when the value of RESET is TRUE. The output variable OUT is of the REAL type.

The integral is the approximate value of the two step functions, and the average value of the data is an approximate integral.

Example in FBD:



INTEGRAL input and output:

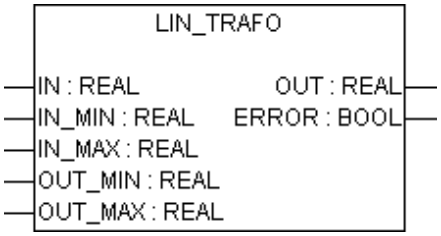


1.2.3.3 LIN_TRAFO

This function block converts the real numbers in the range determined by the upper limit and lower limit into real numbers in the range determined by the other upper limit and lower limit.

The following expression is based on such conversion:

(IN - IN_MIN) : (IN_MAX - IN) = (OUT - OUT_MIN) : (OUT_MAX - OUT)



Input variables:

Variable	Data type	Description
IN	REAL	Input variable
IN_MIN	REAL	Lower limit in the variable range
IN_MAX	REAL	Upper limit in the variable range
OUT_MIN	REAL	Lower limit in the output range
OUT_MAX	REAL	Upper limit in the output range

Output variables:

Variable	Data type	Description
OUT	REAL	Output value
ERROR	BOOL	An error occurred: TRUE, if IN_MIN = IN_MAX, or IN is out of the specified input range.

Application example:

A temperature is provided by a voltage value (input IN). However, the temperature will be converted into temperature values (output OUT). The range of the input (voltage) value is restricted by IN_MIN=0 and IN_MAX=10. The range of the output (temperature) value is restricted by OUT_MIN=-20 and OUT_MAX=40.

Therefore, the input of a 5V voltage value will cause the output of a 10°C temperature.

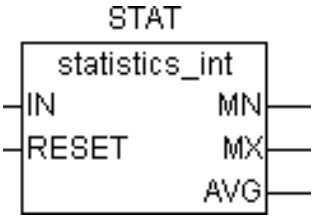
1.2.3.4 STATISTICS_INT

This function block is used to calculate data for standard data collection.

The input variable IN is of the INT type. When the input variable RESET of the BOOL type is TRUE, all data values are initialized again.

Among the output variables, MN is the minimum value of IN, MX is the maximum value of IN, and AVG is the average value. The three output variables are of the INT type.

Example in FBD:



1.2.3.5 STATISTICS_REAL

This function block is similar to STATISTICS_INT. The only difference is that the input variable IN and output variables MN, MX, and AVG of this function block are of the REAL type.

1.2.3.6 VARIANCE

This function block is used to calculate the variance of input data.

The input variables IN and RESET are of the REAL and BOOL types respectively, and the output variable OUT is of the REAL type.

This function block is used to calculate the variance of input data. When RESET=TRUE, VARIANCE is reset.

The standard deviation can be easily obtained by calculating the square root of variance.

1.2.4 Regulators

1.2.4.1 PD

This function block is used to regulate proportions and differentials.

Input variables:

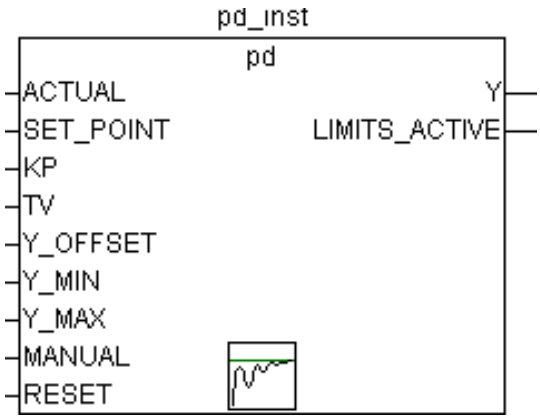
Variable	Data type	Description
ACTUAL	REAL	Actual value of the control variable.
SET_POINT	REAL	Description value and command value.
KP	REAL	Proportional coefficient used to represent the proportional gain of the P- part.
TV	REAL	Differential time used to represent the time calculated in seconds of the D- part. For example, "0.5" indicates 500 seconds.
Y_MANUAL	REAL	Used to define the output value Y when MANUAL=TRUE.
Y_OFFSET	REAL	Offset value of the operation value Y.
Y_MIN, Y_MAX	REAL	Lower limit and upper limit of the operation value Y. If Y reaches a limit value, LIMITS_ACTIVE is set to TRUE and Y is kept within the

Variable	Data type	Description
		formulated range. This function block works only when Y_MIN < Y_MAX.
MANUAL	BOOL	If it is TRUE, manual operating is activated, and the output value is defined through Y_MANUAL.
RESET	BOOL	Setting the value to TRUE will reset the controller. During re-initialization, Y is equal to Y_OFFSET.

Output variables:

Variable	Data type	Description
Y	REAL	Operation value, defined by the function block (see the following).
LIMITS_ACTIVE	BOOL	When the value is TRUE, Y reaches the given limit value (Y_MIN or Y_MAX).

Example in FBD:



Y_OFFSET, Y_MIN, and Y_MAX are used to convert numbers in specified ranges.

MANUAL can be used to enable or disable manual operating. RESET is used to reset the controller.

During normal operating (MANUAL = RESET = LIMITS_ACTIVE = FALSE), the controller calculates the deviation value SET_POINT- ACTUAL and stores the time-related derivatives de/dt as internal variables.

The output value Y can be obtained by using the following:

$$\mathrm{Y} = \mathrm{KP} \cdot (\Delta + \mathrm{TV}\frac{\delta\Delta}{\delta t}) + \mathrm{Y_OFFSET}$$

In the formula, Δ=SET_POINT- ACTUAL

Therefore, except for the P-part and the present deviation (D-part) of the controller, all the others have an impact on the calculation output.

In addition, Y is restricted to the range defined by Y_MIN and Y_MAX. If Y reaches a limit value, LIMITS_ACTIVE is set to TRUE. If there is no calculation limit value, Y_MIN and Y_MAX must be set to 0.

Once MANUAL=TRUE, Y is written into Y_MANUAL.

A P adjustment can be achieved by setting TV=0.

1.2.4.2 PID

This function block is used to regulate proportions, integrals, and differentials.

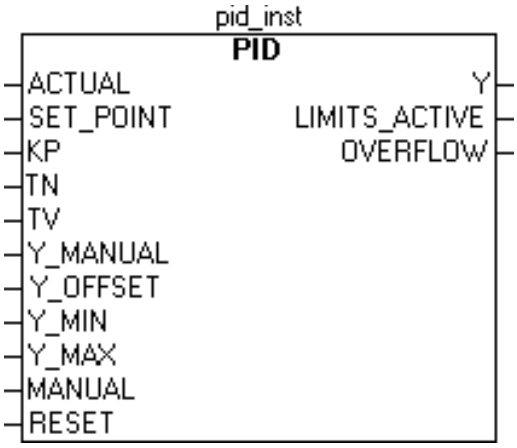
Input variables:

Variable	Data type	Description
ACTUAL	REAL;	Actual value of the control variable.
SET_POINT	REAL;	Expected value, command variable.
KP	REAL;	Proportional coefficient. The value cannot be 0 for the unity gain in the P- part; otherwise, the function block does not perform any calculations.
TN	REAL;	Reset time. The unit gain in the i part is fixed to seconds. For example, "0.5" is 500 milliseconds, the value must be greater than 0; otherwise, the function block does not perform any calculations. A smaller TN value obtains a greater integral part, including the variable value. A greater TN value obtains a smaller integral part.
TV	REAL;	When the differential functions, the unit gain in the D- part is fixed to seconds. For example, "0.5" is 500 milliseconds.
Y_MANUAL	REAL;	The output value is Y when MANUAL = TRUE.
Y_OFFSET	REAL;	Offset operation variable Y.
Y_MIN, Y_MAX	REAL;	A smaller resp value indicates a higher upper limit of the operation variable Y. If Y exceeds a limit value, LIMITS_ACTIVE is set to TRUE and Y is kept within the formulated range. Only when Y_MIN < Y_MAX, the control takes effect.
MANUAL	BOOL	If it is TRUE, manual operating is activated, and the operation variable is defined through Y_MANUAL.
RESET	BOOL	During initialization in which Y is equal to Y_OFFSET, setting the value to TRUE will reset the controller.

Output variables:

Variable	Data type	Description
Y	REAL;	Operation variable value, defined by the function block (see the following).
LIMITS_ACTIVE	BOOL	The value TRUE indicates that Y is out of the range defined by Y_MIN and Y_MAX.
OVERFLOW	BOOL	The value TRUE indicates overflow.

Example in FBD:



Y_OFFSET, Y_MIN, and Y_MAX are used to convert numbers in specified ranges.

MANUAL can be used to enable or disable manual operating. RESET is used to reset the controller.

During normal operating (MANUAL = RESET = LIMITS_ACTIVE = FALSE), the controller calculates the deviation value SET_POINT- ACTUAL and stores the time-related derivatives de/dt as internal variables.

The output value Y can be obtained by using the following:

$$\mathrm{Y} = \mathrm{KP} \cdot (\Delta + \frac{1}{TN}\int e\mathrm{dt} + \mathrm{TV}\,\frac{\delta\Delta}{\delta t}) + \mathrm{Y_OFFSET}$$

In the formula, Δ=SET_POINT - ACTUAL

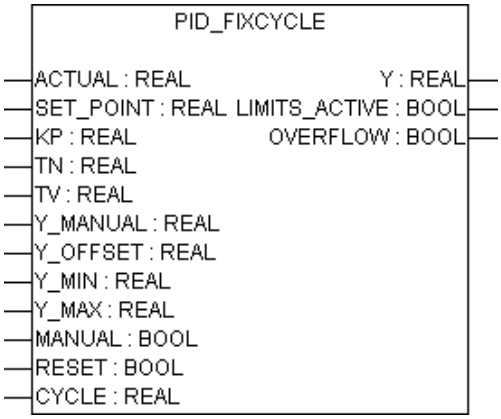
Therefore, except for the P-part and the present deviation (D-part) of the controller, all the others have an impact on the calculation output.

The PID controller can be easily converted into a PI controller by setting TV=0.

Incorrect controller parameter settings may cause overflow if the incorrect integral part becomes larger. Therefore, for safety purpose, the output can invoke OVERFLOW, in which the value is TRUE. This happens only when the control system is unstable due to incorrect parameter settings. At the same time, the controller is suspended and can be reactivated only through re-initialization.

1.2.4.3 PID_FIXCYCLE

Example in FBD:



The function of this function module is the same as that of the PID controller. The difference is that its cycle time is set by CYCLE (seconds) instead of being automatically measured by an internal function.

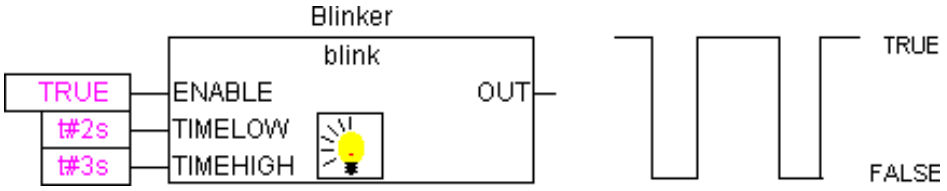
1.2.5 Signal generator

1.2.5.1 BLINK

This function block is used to generate a pulse signal. The input variable ENABLE is of the BOOL type, and the input variables TIMELOW and TIMEHIGH are of the TIME type. The output variable OUT is of the BOOL type.

If ENABLE is set to TRUE, BLINK is enabled. The value of OUT is TRUE within the time period specified by TIMEHIGH, while it is FALSE within the time period specified by TIMELOW.

Example in CFC:



1.2.5.2 FREQ_MEASURE

This function block is used to measure the (average) frequency value (Hz) of the Boolean input signal. Measuring periods can be specified. A measuring period is the interval between two signal rising edges.

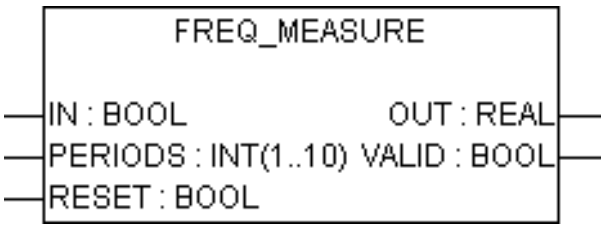
Input variables:

Variable	Data type	Description
IN	BOOL	Input signal.
PERIODS	INT	Sequence number of a period, which is the interval between two signal rising edges. Options: 1 to 10
RESET	BOOL	Used to reset all parameters to 0.

Output variables:

Variable	Data type	Description
OUT	REAL	Result frequency [Hz]
VALID	BOOL	The value is FALSE until the first measuring ends or if the period is greater than 3*OUT (indicating an input fault).

Example in FBD:



1.2.5.3 GEN

This function block is used to generate standard oscillation periods.

The input variable MODE can predefine the GEN_MODE type; BASE is of the BOOL type; PERIOD is of the TIME type; CYCLES and AMPLITUDE are of the INT type; RESET is of the BOOL type.

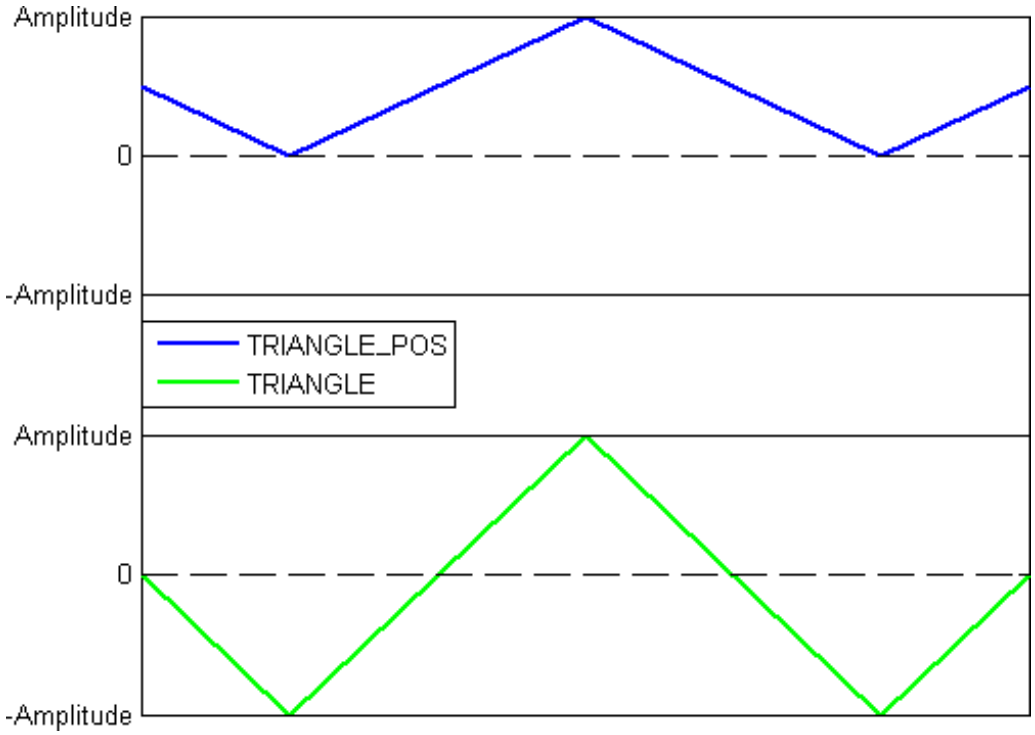
MODE is used to define oscillation generation mode. Among them, the enumerated values TRIANGLE and TRIANGLE_POS are triangle waves; SAWTOOTH_RISE indicates incremental sawtooth waves; SAWTOOTH_FALL indicates decremental sawtooth waves; RECTANGLE indicates square waves; SINUS and COSINUS indicate sine waves and cosine waves respectively.

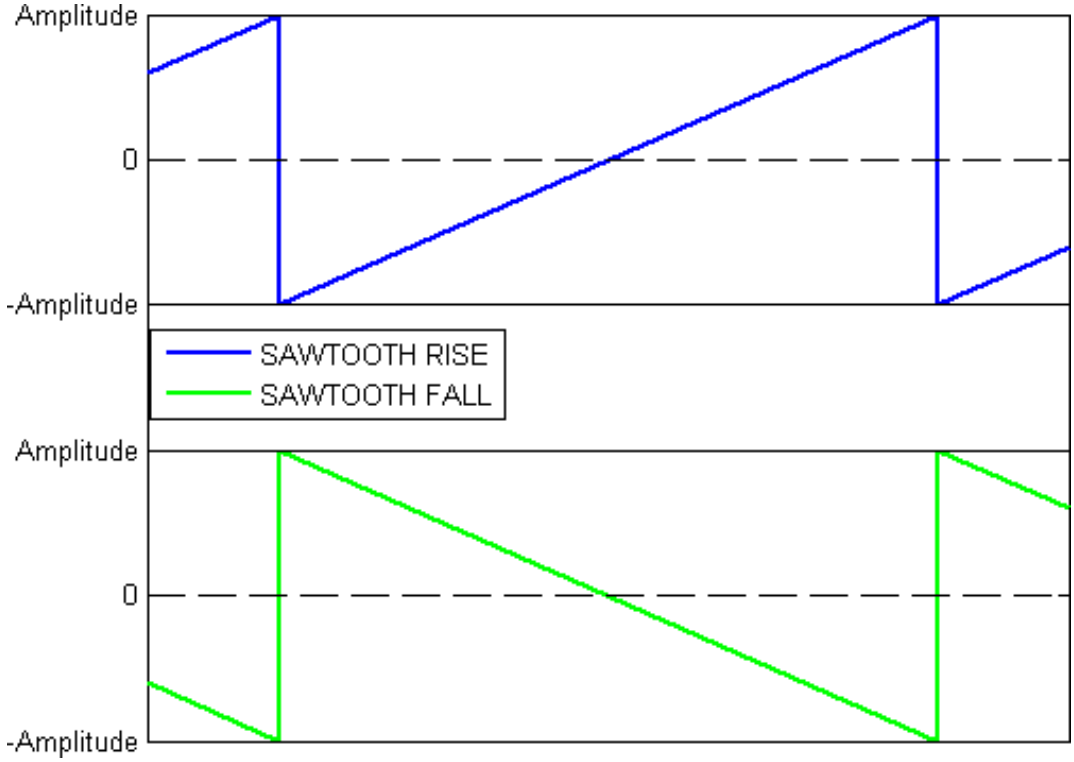
BASE is used to define whether to use the set time to define the cyclic period (BASE=TRUE) or use a specific period value that represents the number of times the function block is invoked to define the cyclic period (BASE=FALSE).

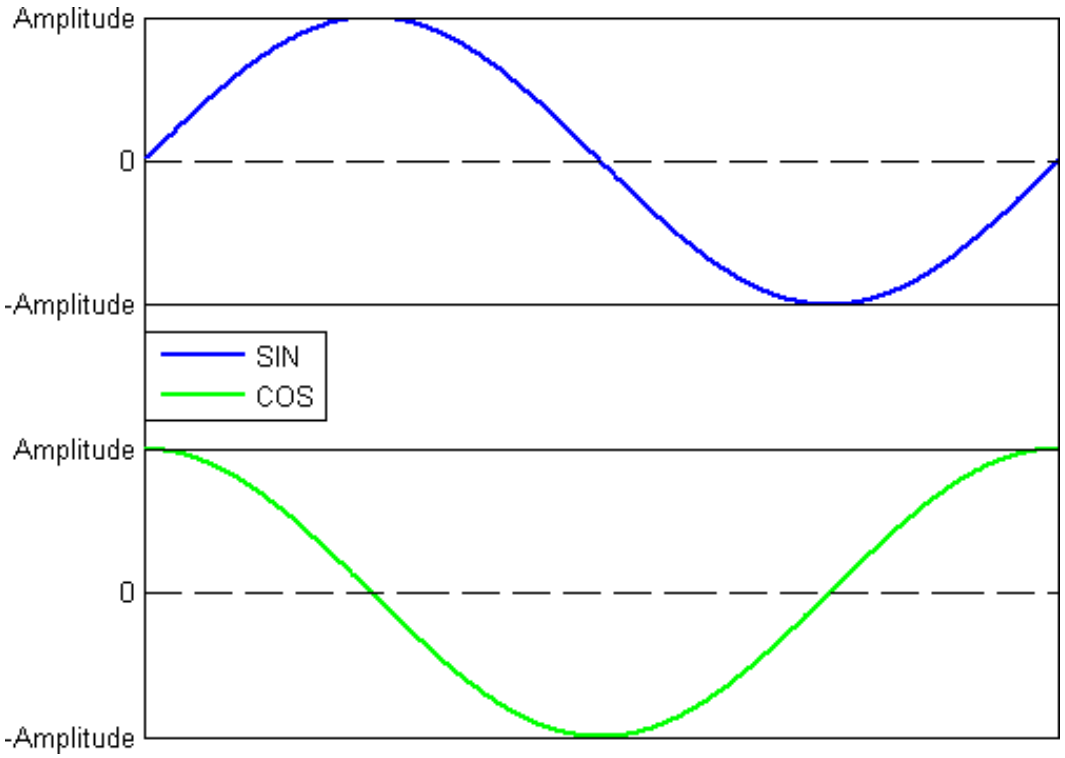
PERIOD or CYCLS is used to define the corresponding cyclic period.

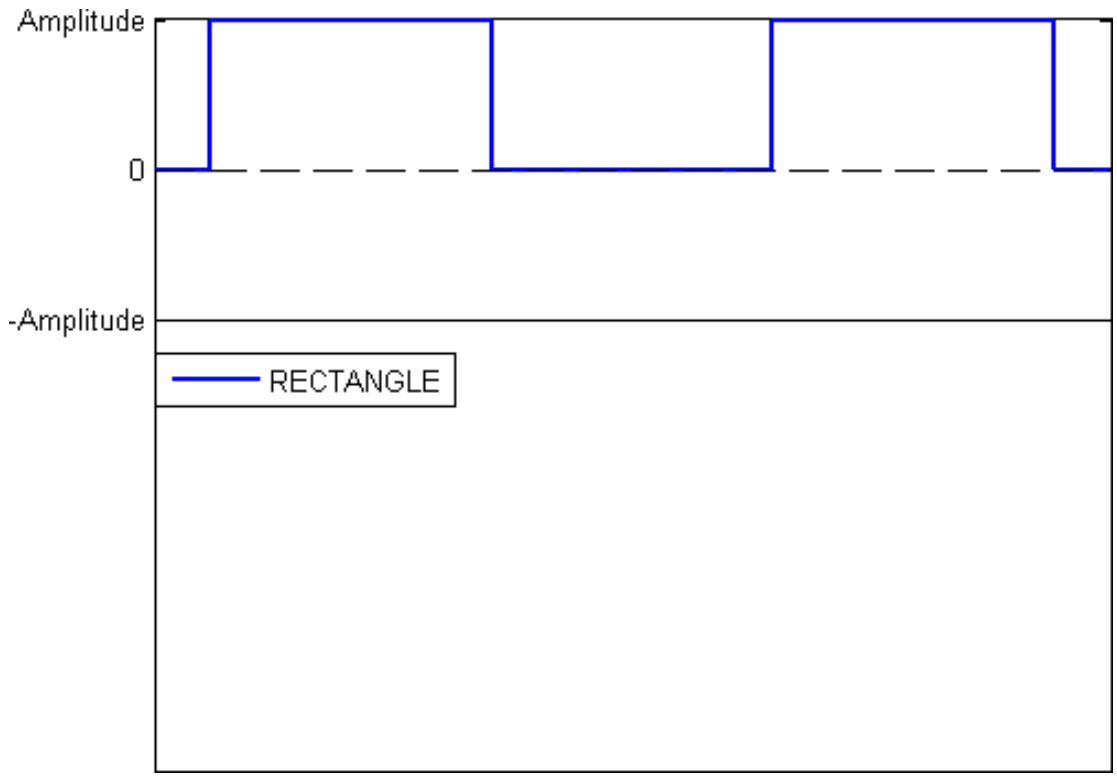
AMPLITUDE is used to define the amplitude generated.

When RESET=TRUE, the signal generator is reset to 0.

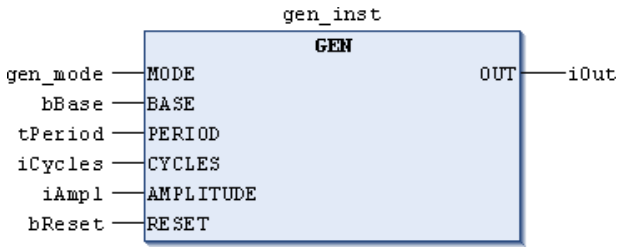



Amplitude
0
-Amplitude
SAWTOOTH RISE
SAWTOOTH FALL
Amplitude
0
-Amplitude


Amplitude
0
-Amplitude
SIN
COS
Amplitude
0
-Amplitude



Example in CFC:



1.2.6 Operation function block

1.2.6.1 CHARCURVE

This function block is used to express values in linear manner.

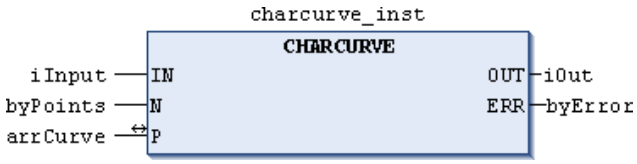
The input variable IN is of the INT type, used to set the value to be processed; N is of the BYTE type, used to set the number of points. P is a predefined POINT-type value based on two integers (X and Y), and array P[0..10] is used to generate characteristic lines.

The output variable OUT is of the INT type, used to output processed data; ERR is of the BYTE type, used to display errors.

The points P[0]..P[N-1] in the array must be sorted according to their X values; otherwise, ERR receives the value 1. If the value of IN is not between P[0].X and P[N-1].X, ERR=2, and OUT contains the corresponding limit value P[0].Y or P[N-1].Y .

If the value of N is out of the range 2-11, ERR=4.

Example in FBD:



Example in ST:

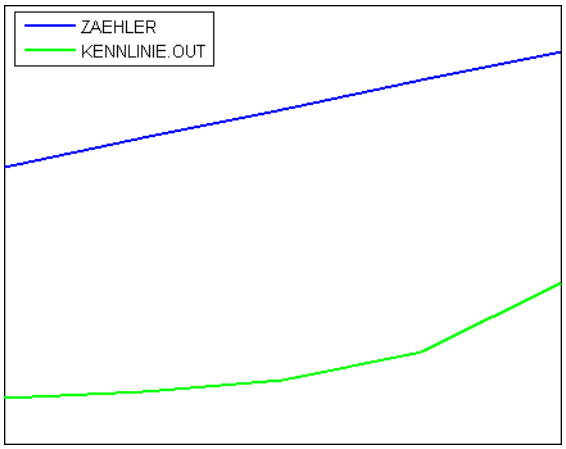
First, define array P.

```
VAR
...
CHARACTERISTIC_LINE:CHARCURVE;
KL:ARRAY[0..10] OF POINT:=[(X:=0,Y:=0),(X:=250,Y:=50),
(X:=500,Y:=150),(X:=750,Y:=400),7((X:=1000,Y:=1000))];
COUNTER:INT;
...
END_VAR
```

Then, supply CHARCURVE with a constantly increasing value, such as:

```
COUNTER:=COUNTER+10;
CHARACTERISTIC_LINE(IN:=COUNTER,N:=5,P:=KL);
```

Track the display effect:



1.2.6.2 RAMP_INT

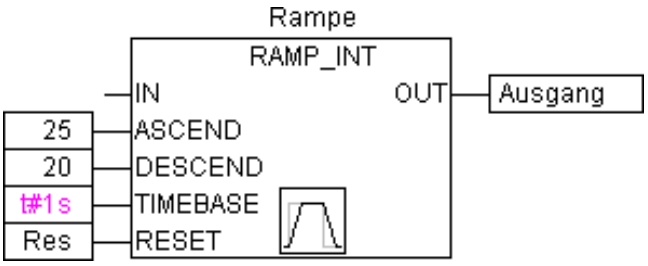
This function block is used to limit the rising or falling rates of the input values.

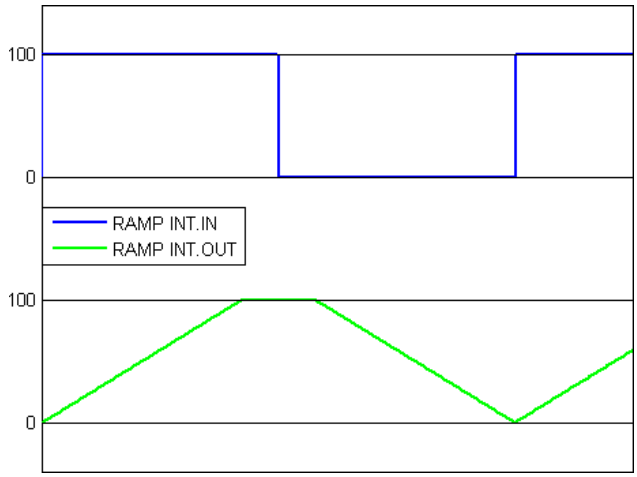
The input variables IN, ASCEND and DESCEND are of the INT type, among which, IN is the input value, while ASCEND and DESCEND are the maximum increment value and decrement value in a given time respectively. TIMEBASE is a TIME type, used to set a given time. When the value of RESET is TRUE, RAMP_INT is reinitialized.

The output variable OUT is of the INT type, and it contains the numbers of which the rising rate and falling rate have been restricted.

When the value of TIMEBASE is t#0S, ASCEND and DESCEND are irrelevant with the time interval, but kept the same.

Example in CFC:





1.2.6.3 RAMP_REAL

With similar functions, RAMP_REAL differs from RAMP_INT only in the following: The input variables IN, ASCEND, and DESCEND, and the output variable OUT of RAMP_REAL are of the REAL type.

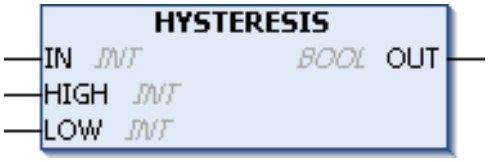
1.2.7 Analog value processing

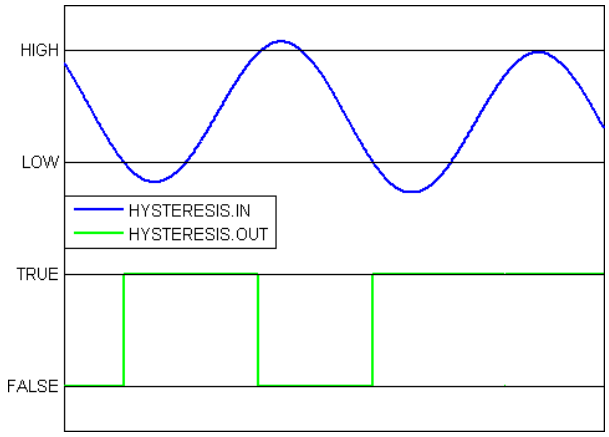
1.2.7.1 HYSTERESIS

This function block contains three INT-type variables IN, HIGH and LOW. The output variable OUT is of the BOOL type.

If IN is lower than the lower limit LOW, the value of OUT is TRUE. If IN is higher than the lower limit HIGH, the value of OUT is FALSE.

Example in FBD:





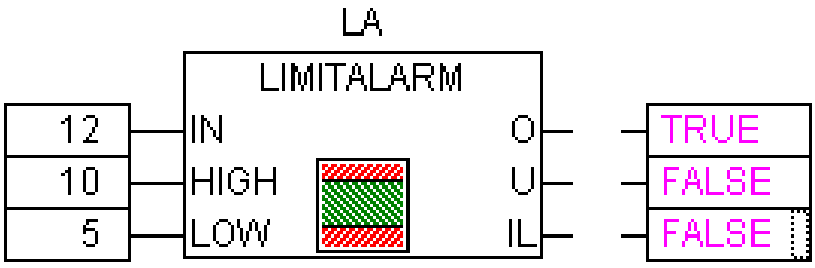
1.2.7.2 LIMITALARM

This function block is used to check whether input values are within the set ranges and which limits are exceeded.

The input variables IN, HIGH and LOW are all of the INT type. The output variables O, U, and IL are all of the BOOL type.

When IN reaches the input upper limit HIGH, O is set to TRUE; when IN is lower than LOW, U is set to TRUE. When IN is between LOW and HIGH, IL is set to TRUE.

Example in FBD:



2 Motion control commands

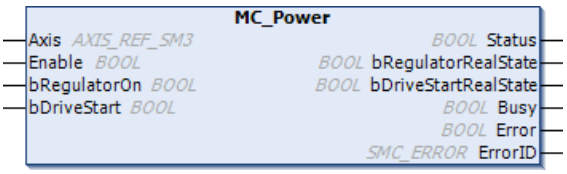
2.1 Description of the SM3_Basic library

2.1.1 MC_Power

MC_Power: used to enable the servo drive.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

MC_Power	Axis enabling		`MC_Power(`
			`Axis:= ,`
			`Enable:= ,`
			`bRegulatorOn:= ,`
			`bDriveStart:= ,`
			`Status=> ,`
			`bRegulatorRealState=> ,`
			`bDriveStartRealState=> ,`
			`Busy=> ,`
			`Error=> ,`
			`ErrorID=> );`

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Enable	Execution condition	BOOL	TRUE, FALSE	FALSE	A rising edge of the input will start the processing of the function block.
bRegulatorOn	Execution condition	BOOL	TRUE, FALSE	FALSE	
bDriveStart	Execution condition	BOOL	TRUE, FALSE	FALSE	

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Status	Enabling	BOOL	TRUE, FALSE	FALSE	It becomes TRUE when the Enabled state is entered.
bRegulatorRealState		BOOL	TRUE,	FALSE	It becomes TRUE after

Output variable	Name	Data type	Valid range	Initial value	Description
			FALSE		bRegulatorOn is set to TRUE.
bDriveStartRealState		BOOL	TRUE, FALSE	FALSE	It becomes TRUE after bDriveStart is set to TRUE.
Busy	Executing	BOOL	TRUE, FALSE	FALSE	It becomes TRUE after the command is accepted.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It becomes TRUE when a fault occurs.
ErrorID	Error ID	SMC_ERROR			When an error occurs, the error ID is output.

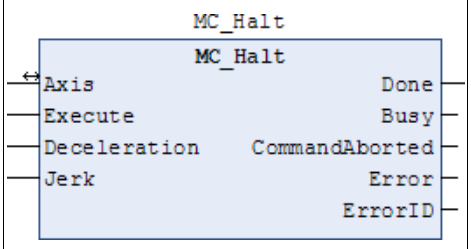
(3) Function description

When Enable is set to TRUE, the axis specified by Axis enters the operable state. Setting the axis status to operable can implement axis control. When Enable is set to FALSE, the axis specified by Axis exits the operable state. After exiting the operable state, the axis does not accept any command, and therefore axis control cannot be implemented. In addition, the axis abnormally responds to motion commands, but the axis can execute the MC_Power and MC_Reset commands.

2.1.2 MC_Halt

MC_Halt: used to stop the motion of a specified axis.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
MC_Halt	Command to stop an axis normally		MC_Halt(Axis:= , Execute:= , Deceleration:= , Jerk:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Execute	Execution condition	BOOL	TRUE, FALSE	FALSE	A rising edge of the input will start the processing of the function block.
Deceleration	Deceleration	LREAL	Positive	0	Function block deceleration

Input variable	Name	Data type	Valid range	Initial value	Description
			number or 0		speed (μ/S^2)
Jerk	Execution condition	LREAL	Positive number or 0	0	Specified jump [Command unit/S^3]

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Command execution completion	BOOL	TRUE, FALSE	FALSE	It is set to TRUE after the axis command is executed completely.
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.
CommandAborted	Command interruption	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is interrupted.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.

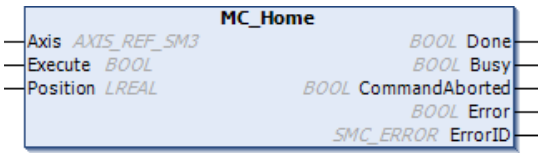
(3) Function description

Starting this function block can stop the motion of an axis, but the execution of this function block can be terminated when another motion axis command is started. This function block can be executed only when the axis is in running state. This function block is started at the rising edge of the input variable execution condition. The axis status changes from Discrete Motion during function block execution and to Standstill after the function block execution.

2.1.3 MC_Home

MC_Home: used to determine the home position of an axis.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
MC_Home	Axis homing command		MC_Home(Axis:= , Execute:= , Position:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Execute	Execution condition	BOOL	TRUE, FALSE	FALSE	A rising edge of the input will start the processing of the function block.
Position	Position that the axis reaches	LREAL	Data range	0	Home position of the axis.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Command execution completion	BOOL	TRUE, FALSE	FALSE	It is set to TRUE after the axis command is executed completely.
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.
CommandAborted	Command interruption	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is interrupted.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.

(3) Function description

This function block is used for homing and it is started at the rising edge of the input variable execution condition. The position that the axis of the input variable reaches is the Home position. This function block can be executed only when the axis is in the Standstill state. In addition, the servo homing mode must be set before the execution, and the axis must be in the Homing state during the execution.

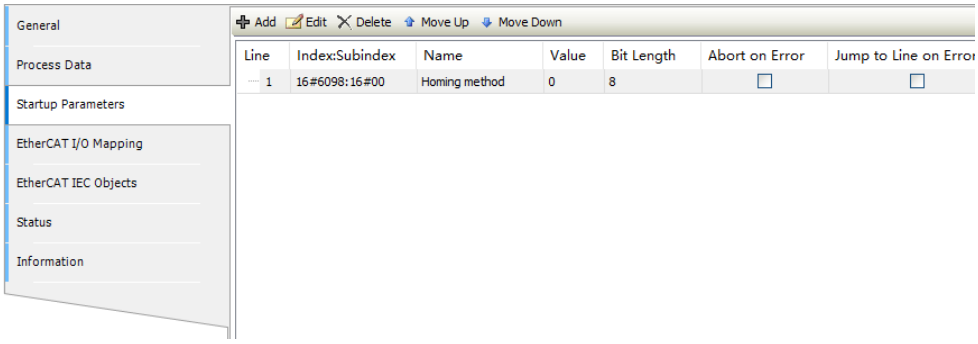
There are two methods for setting the homing mode:

Method 1: Manually setting servo function codes

Method 2: Setting startup parameters of AX series slave nodes. If communication modes are used, index and sub-index data must be set.

Item	Index	Sub-index	Description
Homing method	0x6098		Set parameters according to specific servo manuals.
Origin finding speed	0x6099	0x01	Generally the speed is defined relatively high, reducing the zero return time.
Zero finding speed	0x6099	0x02	Generally the speed is defined relatively low.
ACC/DEC for homing	0x609A		Acceleration or deceleration during homing.
Homing timeout time	0x2005	0x24	If the homing time exceeds the specified time, the system reports "Err.601".

The setting interface is similar to the following:

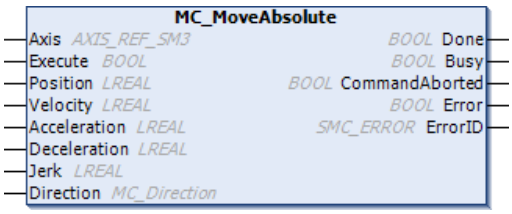


2.1.4 MC_MoveAbsolute

MC_MoveAbsolute: used to specify the destination position of absolute coordinates for positioning.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

MC_Move Absolute	Axis absolute position control command		`MC_MoveAbsolute(`
			`Axis:= ,`
			`Execute:= ,`
			`Position:= ,`
			`Velocity:= ,`
			`Acceleration:= ,`
			`Deceleration:= ,`
			`Jerk:= ,`
			`Direction:= ,`
			`Done=> ,`
			`Busy=> ,`
			`CommandAborted=> ,`
			`Error=> ,`
			`ErrorID=> );`

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Execute	Execution condition	BOOL	TRUE, FALSE	FALSE	A rising edge of the input will start the processing of the function block.
Position	Position that the axis reaches	LREAL	Data range	0	Absolute position of the axis.
Velocity	Running speed	LREAL	Data range	0	Max. speed at which the axis runs to reach the destination position.
Acceleration	Acceleration	LREAL	Data range	0	Acceleration when the speed increases.
Deceleration	Deceleration	LREAL	Data range	0	Deceleration when the speed reduces.
Jerk	Jump	LREAL	Data range	0	Slope change value of the curve acceleration or deceleration.
Direction	Command polarity	MC_DIRECTION	Negative, Shortest, Positive, Current, Fastest	Shortest	Negative: Move reversely. Shortest: Select a direction depending on the shortest distance. Positive: Move forward. Current: Move at the current direction. Fastest: Automatically choose to move at fastest manner. (Valid only in rotary mode)

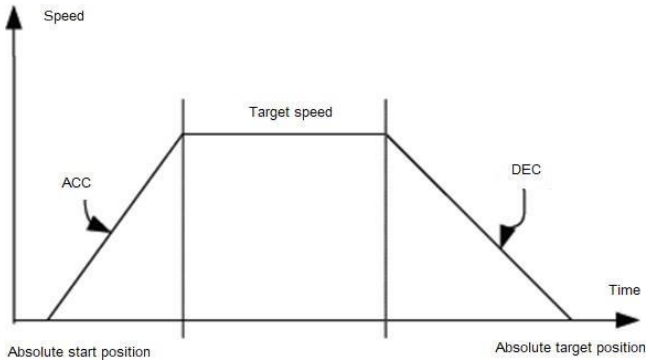
✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Command execution completion	BOOL	TRUE, FALSE	FALSE	It is set to TRUE after the axis command is executed completely.
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.
CommandAborted	Command interruption	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is interrupted.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.

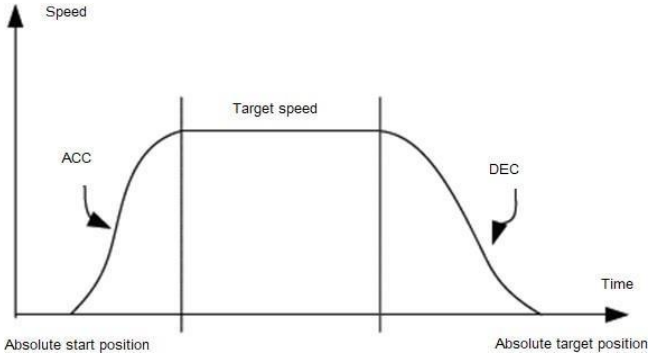
(3) Function description

This function block is the axis absolute-position command. Before executing this function block, the axis is in the Standstill state. After the function block is started at the rising edge, the axis is in the Discrete Motion state and moves to the specified position. When Jerk is 0, the axis performs trapezoidal acceleration/deceleration movement; when Velocity, Acceleration, Deceleration and Jerk are not empty, the axis performs S-curve acceleration/deceleration movement.

✧ Trapezoidal acceleration/deceleration action



✧ S-curve acceleration/deceleration action

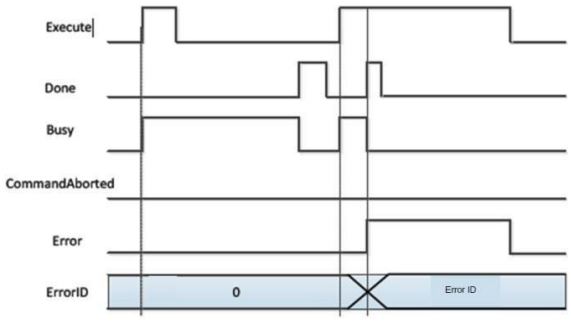


(4) Timing diagram

✧ The axis must be in the Standstill state.

✧ The function block is started at the rising edge.

✧ For the function block, when Done is TRUE, the execution is completed; otherwise, Busy is TRUE.

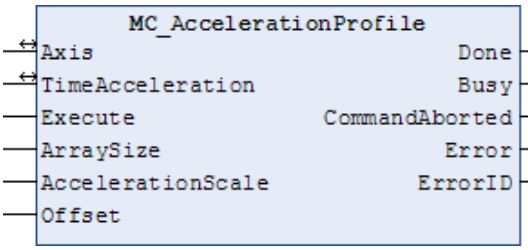


2.1.5 MC_AccelerationProfile

MC_AccelerationProfile: indicates the motion model of the time segment and acceleration/deceleration profile.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

MC_ Acceleration Profile	Acceleration profile command		`MC_AccelerationProfile(`
			`Axis:= ,`
			`TimeAcceleration:= ,`
			`Execute:= ,`
			`ArraySize:= ,`
			`AccelerationScale:= ,`
			`Offset:= ,`
			`Done=> ,`
			`Busy=> ,`
			`CommandAborted=> ,`
			`Error=> ,`
			`ErrorID=> );`

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.
TimeAcceleration	Acceleration time and description of the axis	MC_TA_ REF			Acceleration time and data description of the axis. The acceleration data consists of multiple groups of data.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Execute	Execution condition	BOOL	TRUE, FALSE	FALSE	A rising edge of the input will start the processing of the function block.
ArraySize	Dynamic arrays	INT	Data range	0	Number of arrays used in the motion profile.
AccelerationScale	Comprehensive factor	LREAL	Positive number or 0	1	Scale factor of acceleration or deceleration in MC_TA_REF.
Offset	Offset	LREAL		0	Overall offset value of acceleration and deceleration.

✧ Output variable

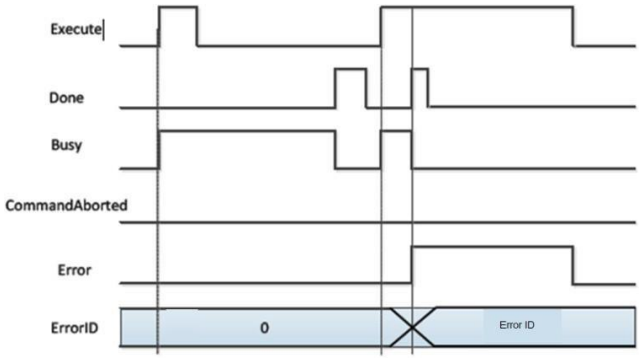
Output variable	Name	Data type	Valid range	Initial value	Description
Done	Command execution	BOOL	TRUE, FALSE	FALSE	It is set to TRUE after the axis command is executed

Output variable	Name	Data type	Valid range	Initial value	Description
	completion				completely.
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.
CommandAborted	Command interruption	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is interrupted.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.

(3) Function description

This function block is used to specify the motion model of the time segment and acceleration/deceleration profile. During the function block execution, the axis is in the Discrete Motion state, and it uses the data in TimeAcceleration. The axis must be in the Standstill state before the function block execution and in the Discrete Motion state during the execution. This function block is started at the rising edge. The execution of this function block superimposes the speeds of the axis that is in the Discrete Motion state, which may cause system faults.

(4) Timing diagram

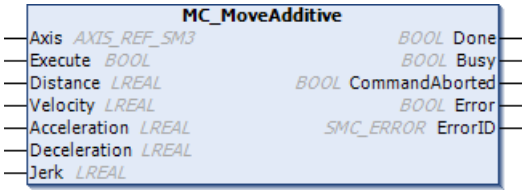


2.1.6 MC_MoveAdditive

MC_MoveAdditive: used for positioning when a specified distance is superimposed to the original position of an axis.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

MC_ MoveAdditive	Absolute motion superimpostion command		`MC_MoveAdditive(`
			`Axis:= ,`
			`Execute:= ,`
			`Distance:= ,`
			`Velocity:= ,`
			`Acceleration:= ,`
			`Deceleration:= ,`
			`Jerk:= ,`
			`Done=> ,`
			`Busy=> ,`
			`CommandAborted=> ,`
			`Error=> ,`
			`ErrorID=> );`

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Execute	Execution condition	BOOL	TRUE, FALSE	FALSE	A rising edge of the input will start the processing of the function block.
Distance	Position that the axis reaches	LREAL	Data range	0	Superimposed position data of the axis.
Velocity	Running speed	LREAL	Data range	0	Max. speed at which the axis runs to reach the destination position.
Acceleration	Acceleration	LREAL	Data range	0	Acceleration when the speed increases.
Deceleration	Deceleration	LREAL	Data range	0	Deceleration when the speed reduces.
Jerk	Jump	LREAL	Data range	0	Slope change value of the curve acceleration or deceleration.

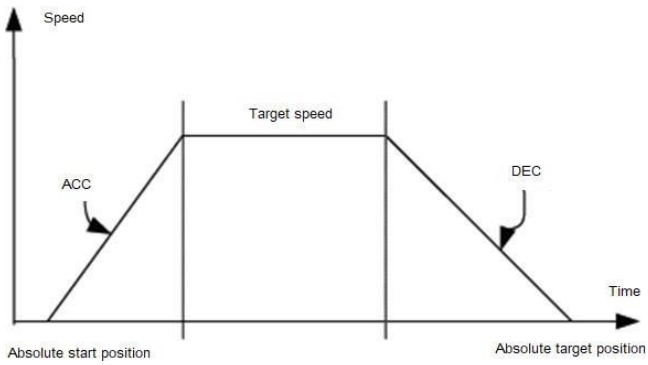
✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Command execution completion	BOOL	TRUE, FALSE	FALSE	It is set to TRUE after the axis command is executed completely.
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.
CommandAborted	Command interruption	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is interrupted.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.

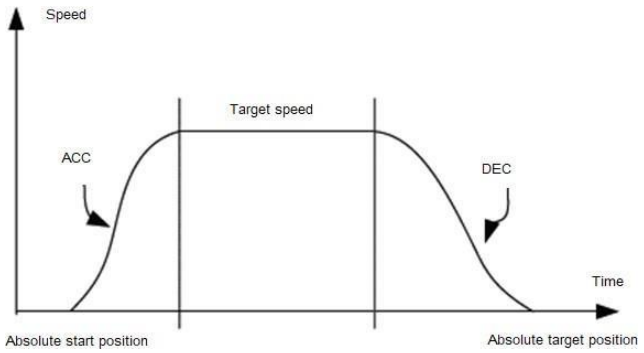
(3) Function description

The startup command is Execute, the rising edge triggers the function block, and Distance specifies the superimposed data of the axis. If the running state of this function block is Discrete Motion, the CommandAbort values of other commands are set; in the standstill state, this command can run independently to achieve relative positioning requirements; if Acceleration or Deceleration is zero, the command execution is abnormal, but the axis is in the Discrete Motion state; when Jerk is 0, the axis performs trapezoidal acceleration or deceleration motion; when Velocity, Acceleration, Deceleration and Jerk are not empty, it performs S-curve acceleration or deceleration motion.

✧ Trapezoidal acceleration/deceleration action

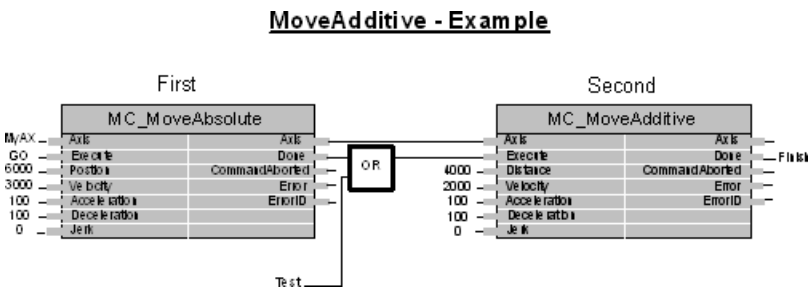


✧ S-curve acceleration/deceleration action

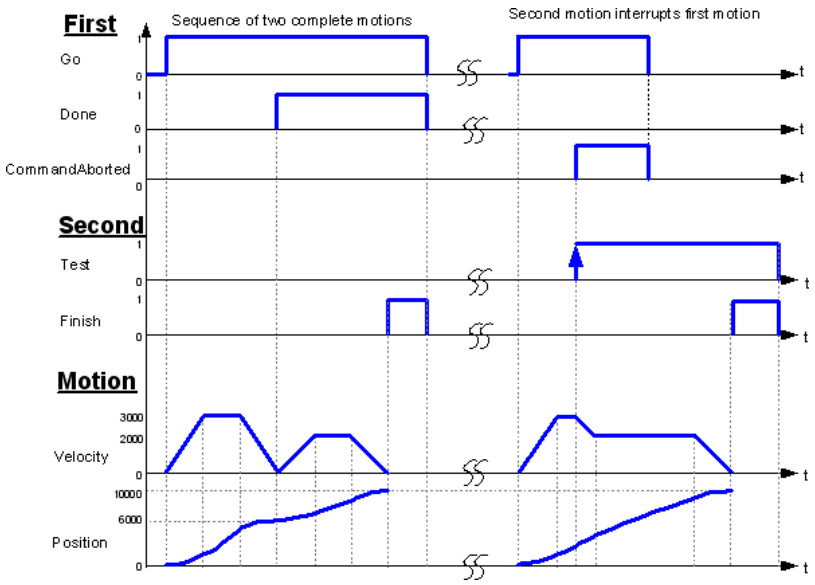


(4) Timing diagram

✧ Example



✧ Timing description

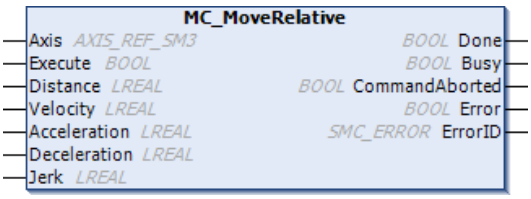


2.1.7 MC_MoveRelative

MC_Move Relative: used for positioning by specify the moving distance from the current position.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

MC_ MoveRelative	Axis relative positioning command		`MC_MoveRelative(`
			`Axis:= ,`
			`Execute:= ,`
			`Distance:= ,`
			`Velocity:= ,`
			`Acceleration:= ,`
			`Deceleration:= ,`
			`Jerk:= ,`
			`Done=> ,`
			`Busy=> ,`
			`CommandAborted=> ,`
			`Error=> ,`
			`ErrorID=> );`

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Execute	Execution condition	BOOL	TRUE, FALSE	FALSE	A rising edge of the input will start the processing of the function block.
Distance	Relative position of motion	LREAL	Data range	0	Relative position of motion.
Velocity	Running speed	LREAL	Data range	0	Max. speed at which the axis runs to reach the destination position.
Acceleration	Acceleration	LREAL	Data range	0	Acceleration when the speed increases.
Deceleration	Deceleration	LREAL	Data range	0	Deceleration when the speed reduces.
Jerk	Jump	LREAL	Data range	0	Slope change value of the curve acceleration or deceleration.

✧ Output variable

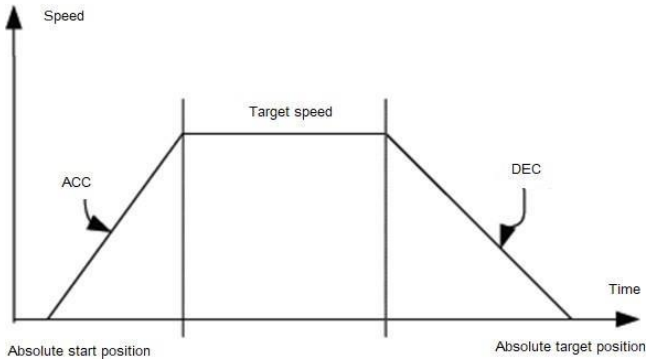
Output variable	Name	Data type	Valid range	Initial value	Description
Done	Command execution completion	BOOL	TRUE, FALSE	FALSE	It is set to TRUE after the axis command is executed completely.
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.
CommandAborted	Command interruption	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is interrupted.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.

(3) Function description

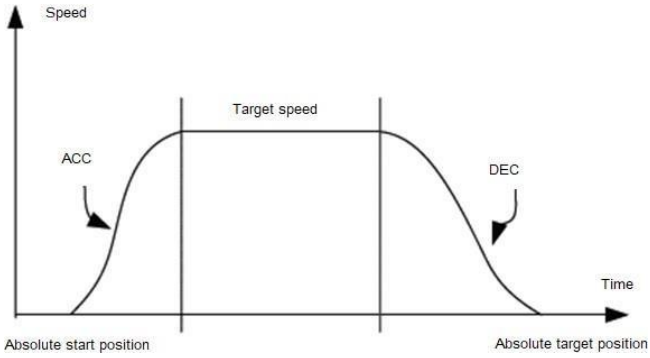
The axis must be in the Standstill state before the function block execution and in the Discrete Motion state during the execution. Pay attention to the axis status during the execution to prevent other commands from interrupting the command execution of the axis. The startup command is Execute, and the rising edge triggers the function block. The

startup command can repeatedly make the rising edge valid when the axis is in the Discrete Motion state, which always refreshes the position. When Acceleration or Deceleration is 0, the command execution is abnormal, but the axis is in the Discrete Motion state.

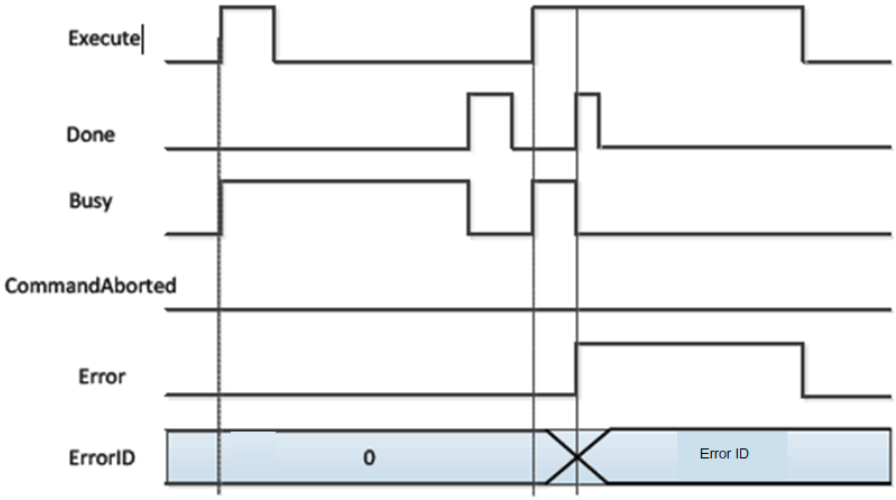
✧ Trapezoidal acceleration/deceleration action



✧ S-curve acceleration/deceleration action



(4) Timing diagram



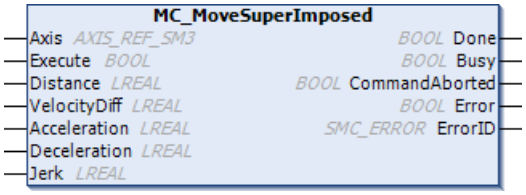
After the Execute command is executed, the rising edge triggers the function block. When Busy is set, the function block is being executed. After the execution is completed, Done is set.

2.1.8 MC_MoveSuperImposed

MC_MoveSuperImposed: used to superimpose speed and position data on the speed and position data in the running command, which brings no change to the entire original command execution time model.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

MC_Move SuperImposed	Relative motion superimposition command		`MC_MoveSuperImposed(`
			`Axis:= ,`
			`Execute:= ,`
			`Distance:= ,`
			`VelocityDiff:= ,`
			`Acceleration:= ,`
			`Deceleration:= ,`
			`Jerk:= ,`
			`Done=> ,`
			`Busy=> ,`
			`CommandAborted=> ,`
			`Error=> ,`
			`ErrorID=> );`

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Execute	Execution condition	BOOL	TRUE, FALSE	FALSE	A rising edge of the input will start the processing of the function block.
Distance	Relative position of motion	LREAL	Data range	0	The data is a relative position.
VelocityDiff	Superimposition speed	LREAL	Data range	0	Superimposition speed for axis running.
Acceleration	Acceleration	LREAL	Data range	0	Acceleration when the speed increases.
Deceleration	Deceleration	LREAL	Data range	0	Deceleration when the speed reduces.
Jerk	Jump	LREAL	Data range	0	Slope change value of the curve acceleration or deceleration.

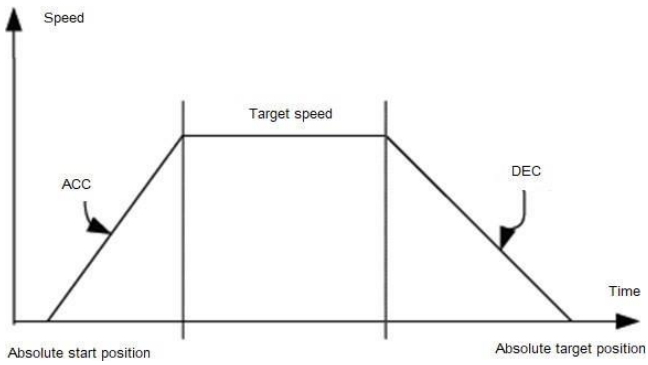
✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Command execution completion	BOOL	TRUE, FALSE	FALSE	It is set to TRUE after the axis command is executed completely.
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.
CommandAborted	Command interruption	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is interrupted.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.

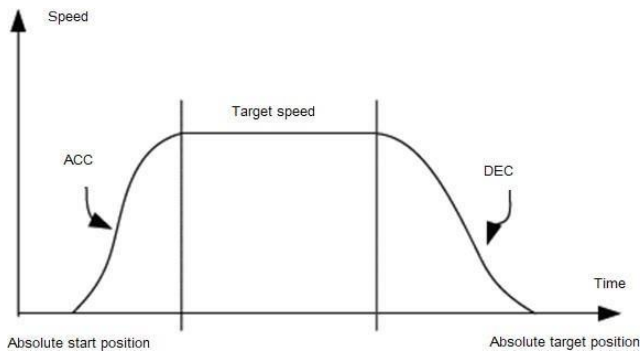
(3) Function description

This function block is the position and speed superimposition command. The startup command is Execute, and the rising edge triggers the function block. VelocityDiff and Distance are superimposed to the speed and position of other commands. In the motion mode, MC_MoveSuperImposed can be superimposed onto any other command. This function block can solve the error compensation for the clearance between the belt and gear, which can ensure motion consistency. To execute the function block, you need to set the parameter superimposition position.

✧ Trapezoidal acceleration/deceleration action

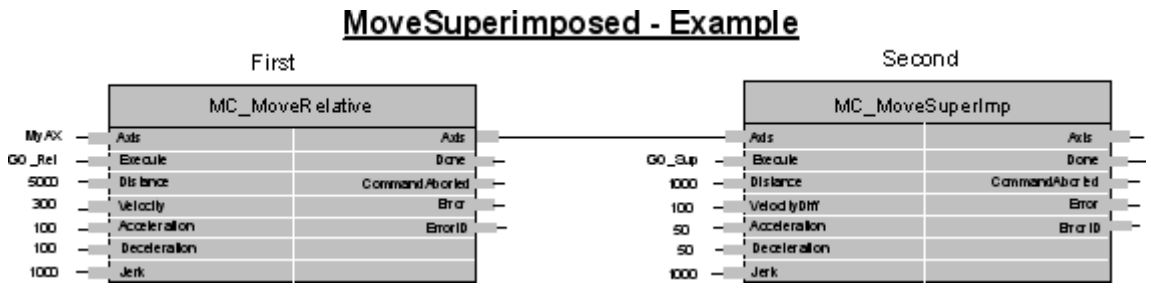


✧ S-curve acceleration/deceleration action

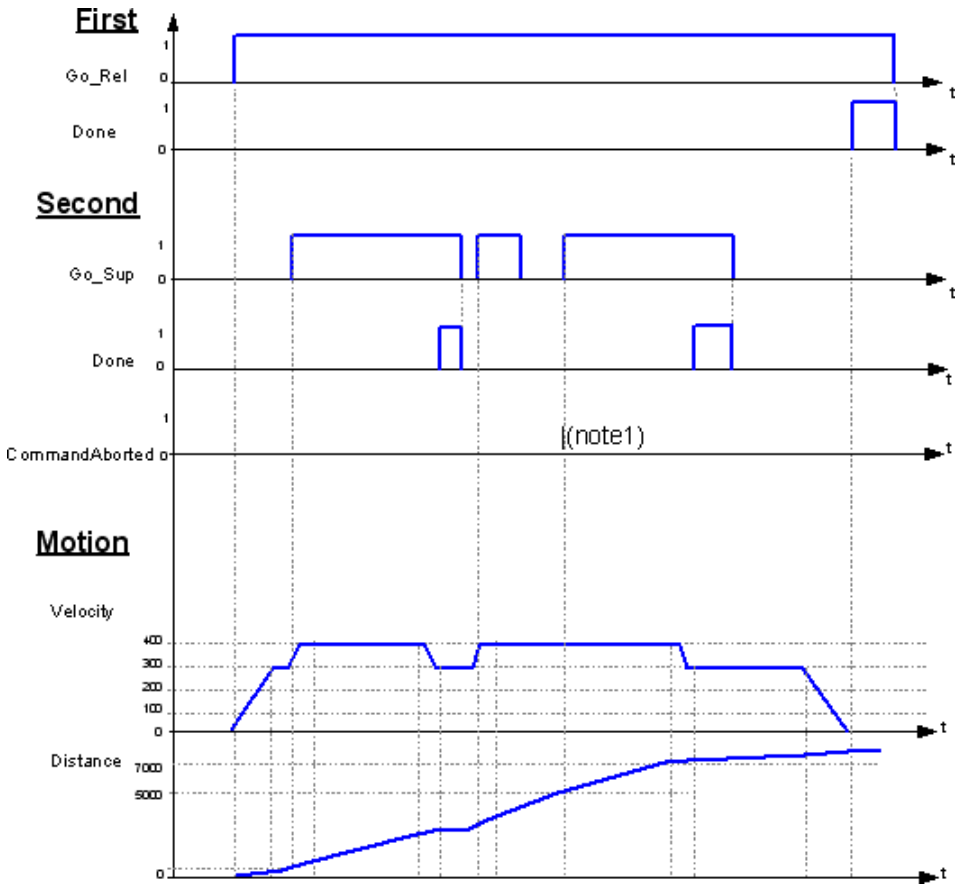


(4) Timing diagram

✧ Example



✧ Timing description

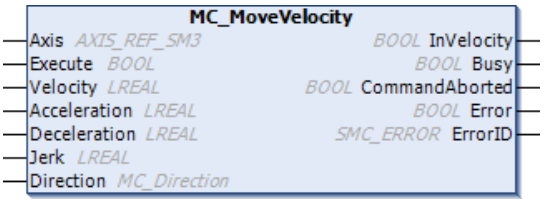


2.1.9 MC_MoveVelocity

MC_MoveVelocity: used to simulate speed control by using the servo drive position control mode.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

MC_ MoveVelocity	Speed control command		`MC_MoveVelocity(`
			`Axis:= ,`
			`Execute:= ,`
			`Velocity:= ,`
			`Acceleration:= ,`
			`Deceleration:= ,`
			`Jerk:= ,`
			`Direction:= ,`
			`InVelocity=> ,`
			`Busy=> ,`
			`CommandAborted=> ,`
			`Error=> ,`
			`ErrorID=> );`

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Execute	Execution condition	BOOL	TRUE, FALSE	FALSE	A rising edge of the input will start the processing of the function block.
Velocity	Running speed	LREAL	Data range	0	Specified speed for running.
Acceleration	Acceleration	LREAL	Data range	0	Acceleration when the speed increases.
Deceleration	Deceleration	LREAL	Data range	0	Deceleration when the speed reduces.
Jerk	Jump	LREAL	Data range	0	Slope change value of the curve acceleration or deceleration.
Direction	Running direction	MC_Direction	positive, negative, current	current	Running direction.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Command	BOOL	TRUE, FALSE	FALSE	It is set to TRUE after the axis

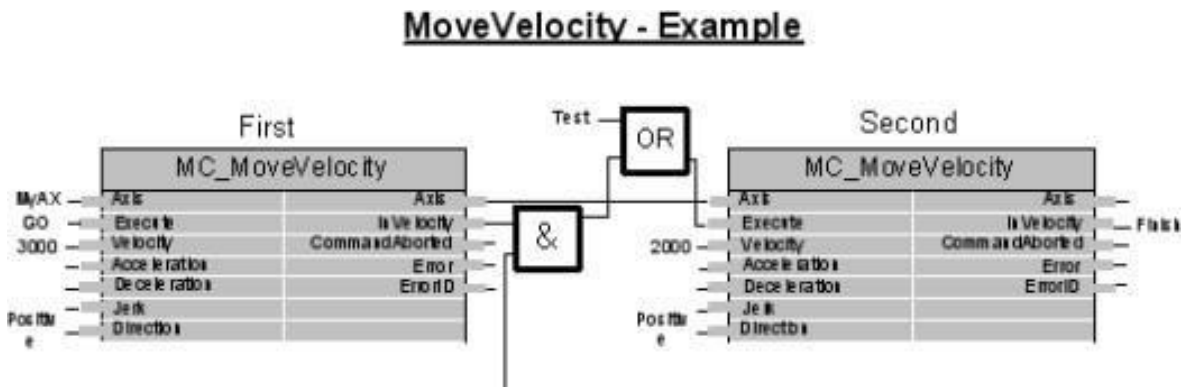
Output variable	Name	Data type	Valid range	Initial value	Description
	execution completion				command is executed completely.
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.
CommandAborted	Command interruption	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is interrupted.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.

(3) Function description

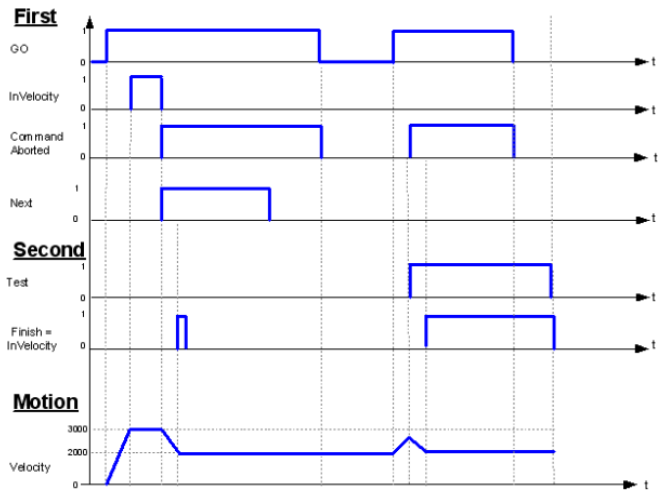
The startup command is Execute, and the rising edge triggers this function block. The drive performs speed control according to the value of Velocity. InVelocity indicates that the running speed in the function block has reached the specified value.

(4) Timing diagram

✧ Example



✧ Timing description

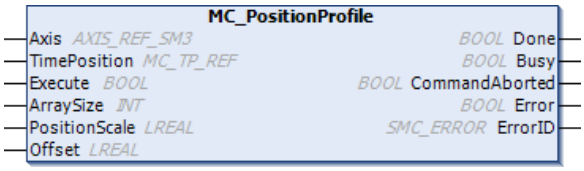


2.1.10 MC_PositionProfile

MC_PositionProfile: indicates the motion model of the time segment and position profile.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

MC_ PositionProfile	Position profile command		`MC_PositionProfile(`
			`Axis:= ,`
			`TimePosition:= ,`
			`Execute:= ,`
			`ArraySize:= ,`
			`PositionScale:= ,`
			`Offset:= ,`
			`Done=> ,`
			`Busy=> ,`
			`CommandAborted=> ,`
			`Error=> ,`
			`ErrorID=> );`

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.
TimePosition	Running time and position description	MC_TP_REF			Running time and position data description of the axis. The data consists of multiple groups of data.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Execute	Execution condition	BOOL	TRUE, FALSE	FALSE	A rising edge of the input will start the processing of the function block.
ArraySize	Array size	INT	Data range	0	Number of arrays used in the motion profile.
PositionScale	Comprehensive factor	LREAL	Positive number or 0	0	Position scaling factor in MC_TP_REF.
Offset	Offset	LREAL		0	Overall offset value of the position.

✧ Output variable

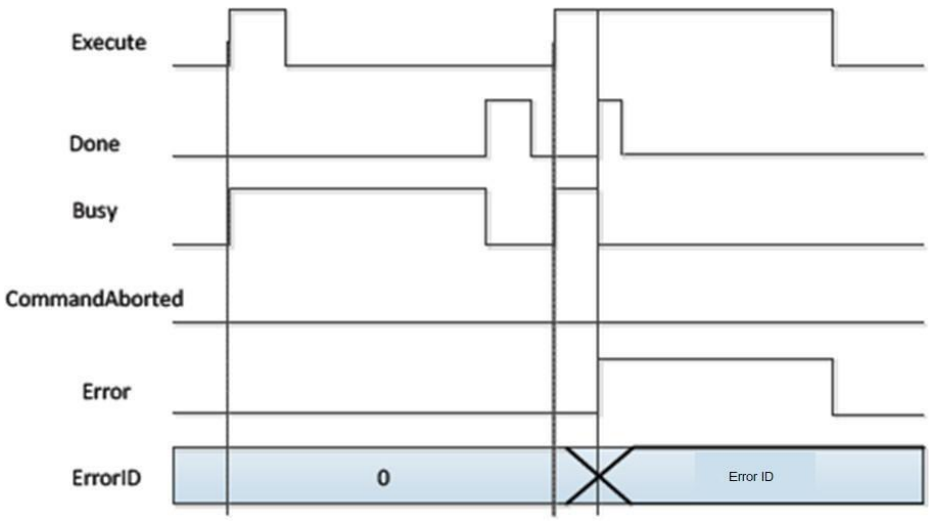
Output variable	Name	Data type	Valid range	Initial value	Description
Done	Command execution	BOOL	TRUE, FALSE	FALSE	It is set to TRUE after the axis command is executed

Output variable	Name	Data type	Valid range	Initial value	Description
	completion				completely.
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.
CommandAborted	Command interruption	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is interrupted.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.

(3) Function description

This function block is used to specify the motion model of the time segment and position profile, using the data in TimePosition. Before executing this function block, the axis is in the Standstill state. This function block is started at the rising edge. The axis is in the Discrete Motion state during the function block execution.

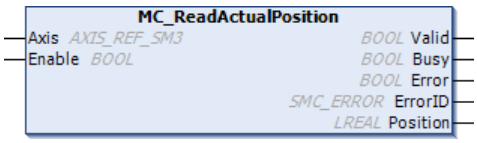
(4) Timing diagram



2.1.11 MC_ReadActualPosition

MC_ReadActualPosition: used to read the actual position of the drive and save it to a user-defined variable.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
MC_ ReadActual Position	Actual position reading command		MC_ReadActualPosition(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , Position=>);

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Enable	Execution condition	BOOL	TRUE, FALSE	FALSE	A rising edge of the input will start the processing of the function block.

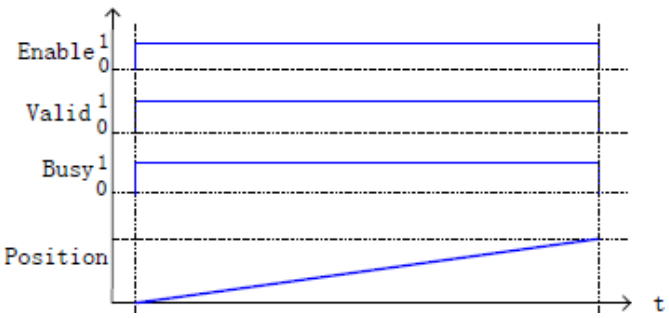
✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Valid	Obtainable flag of position data	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the drive position can be obtained correctly.
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.
Position	Obtained axis position	LREAL	Axis position	0	Axis position data that is read.

(3) Function description

This function block is started at the rising edge and it can read the axis position value. When Valid is TRUE, the read position value is valid. This function block can be repeatedly invoked, and the invoking does not affect the other.

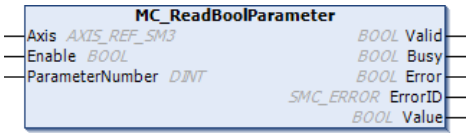
(4) Timing diagram



2.1.12 MC_ReadBoolParameter

MC_ReadBoolParameter: used to read the bit parameters of the drive axis and saves them to user-defined variables.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
MC_ ReadBoolParameter	Axis bit parameter reading		MC_ReadBoolParameter(Axis:= , Enable:= , ParameterNumber:= , Valid=> , Busy=> , Error=> , ErrorID=> , Value=>);

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Enable	Execution condition	BOOL	TRUE, FALSE	FALSE	When it is set to TRUE, this function block is started.
ParameterNumber	Axis parameter number	DINT		0	Access index, sub-index, and number of the axis parameter.

Note:

ParameterNumber (DINT) = -DWORD_TO_DINT(SHL(USINT_TO_DOWRD(usiDataLength), 24) (Data length in the object dictionary)

+ SHL(UINT_TO_DWORD(uiIndex), 8) (Index in the object dictionary-16 bits)

+ usisubIndex (Sub-index in the object dictionary-8 bits)

usiDataLength: Fill in according to the number of bytes: Byte 1 is 16#01; byte 2 is 16#02; byte 4 is 16#04 and so on.

✧ Output variable

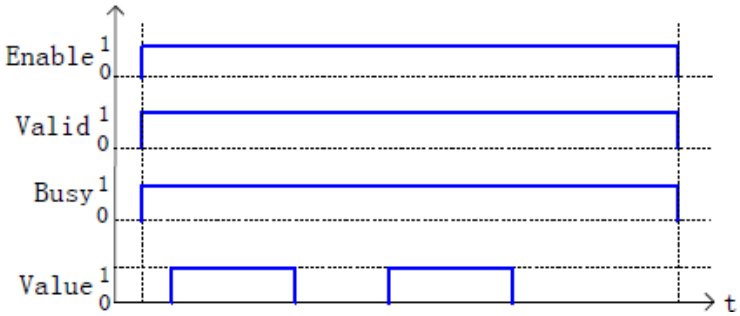
Output variable	Name	Data type	Valid range	Initial value	Description
Valid	Obtainable flag of position data	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the drive position can be obtained correctly.
Busy	Command	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis

Output variable	Name	Data type	Valid range	Initial value	Description
	being executed				command is being executed.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.
Value	Obtained axis position.	BOOL	TRUE, FALSE	FALSE	The axis error code is read.

(3) Function description

Bit data status is read from the drive by executing MC_ReadBoolParam, which is valid when Enable is TRUE. The function block can be repeatedly executed, without mutual affecting. When Valid is TRUE, the bit status data is valid; when Busy is TRUE, the function block is being executed.

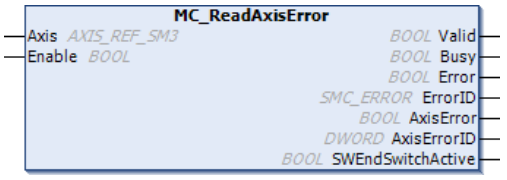
(4) Timing diagram



2.1.13 MC_ReadAxisError

MC_ReadAxisError: used to read axis error information and save it to user-defined variables.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
MC_ Read AxisErro	Axis error reading		MC_ReadAxisError(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , AxisError=> , AxisErrorID=> , SWEndSwitchActive=>);

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Enable	Execution condition	BOOL	TRUE, FALSE	FALSE	When it is set to TRUE, this function block is started.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Valid	Error data obtaining flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the drive position can be obtained correctly.
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR	TRUE, FALSE	FALSE	When an error occurs, the error ID is output.
AxisError	Axis error flag	BOOL	TRUE, FALSE	FALSE	When an error is read, the corresponding flag is set.
AxisErrorID	Axis error code	DWORD		0	The axis error code is read.
SWEndSwitchActive	Software limit switch status	BOOL	TRUE, FALSE	FALSE	The software limit switch status is checked during command reading.

(3) Function description

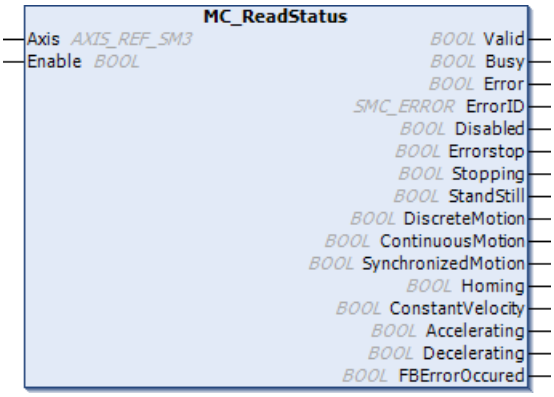
This function block is used to read axis error information, and it is valid when Enable is TRUE. When Valid is TRUE, AxisError and AxisErrorID are valid data values; when Busy is TRUE, the current function block is being executed. This function block can be repeatedly executed without mutual affecting.

2.1.14 MC_ReadStatus

MC_Read Status: used to read axis status data and save it to user-defined variables.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

MC_ ReadStatus	Axis status reading		`MC_ReadStatus(`
			`Axis:= ,`
			`Enable:= ,`
			`Valid=> ,`
			`Busy=> ,`
			`Error=> ,`
			`ErrorID=> ,`
			`Disabled=> ,`
			`Errorstop=> ,`
			`Stopping=> ,`
			`StandStill=> ,`
			`DiscreteMotion=> ,`
			`ContinuousMotion=> ,`
			`SynchronizedMotion=> ,`
			`Homing=> ,`
			`ConstantVelocity=> ,`
			`Accelerating=> ,`
			`Decelerating=> ,`
			`FBErrorOccured=> );`

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Enable	Execution condition	BOOL	TRUE, FALSE	FALSE	When it is set to TRUE, this function block is started.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Valid	Error data obtaining flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the drive position can be obtained correctly.
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR	TRUE, FALSE	FALSE	When an error occurs, the error ID is output.
Disabled	Axis disabled	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis is disabled.
Errorstop	Axis error status	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis is running abnormally.

Output variable	Name	Data type	Valid range	Initial value	Description
Stoping	Axis in stop process	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis is in the stop process.
StandStill	Standard status of axis	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis is in the StandStill state (able to run).
DiscreteMotion	Discrete motion status of axis	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis is in the DiscreteMotion state.
ContinuousMotion	Continuous motion status of axis	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis is in the ContinuousMotion state.
SynchronizedMotion	Synchronous running status of axis	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis is in the SynchronizedMotion state.
Homing	Homing status of axis	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis is in the Homing state.
ConstantVelocity	Axis running speed reached	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis reaches the running speed.
Accelerating	Acceleration status of axis	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis is in the Accelerating state.
Dccelerating	Deceleration status of axis	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis is in the Dccelerating state.
FBErrorOccured	Axis function block error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis function block encounters an error.

(3) Function description

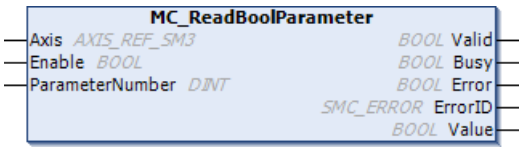
Axis status is read by executing MC_ReadStatus, which is valid when Enable is TRUE. This function block can be repeatedly executed, without mutual affecting.

To execute the function block, set Enable to TRUE. When Valid is TRUE, the axis status data is valid; when Busy is TRUE, the function block is being executed.

2.1.15 MC_ReadParameter

MC_ReadParameter: used to read drive axis parameters of the drive axis and saves them to user-defined variables.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
MC_ ReadParameter	Axis parameter reading		MC_ReadParameter(Axis:= , Enable:= , ParameterNumber:= , Valid=> , Busy=> , Error=> , ErrorID=> , Value=>);

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Enable	Execution condition	BOOL	TRUE, FALSE	FALSE	When it is set to TRUE, this function block is started.
ParameterNumber	Axis parameter number	DINT		0	Access index, sub-index, and number of the axis parameter.

Note:

ParameterNumber (DINT) = -DWORD_TO_DINT(SHL(USINT_TO_DOWRD(usiDataLength), 24) (Data length in the object dictionary)

+ SHL(UINT_TO_DWORD(uiIndex), 8) (Index in the object dictionary-16 bits)

+ usisubIndex (Sub-index in the object dictionary-8 bits)

usiDataLength: Fill in according to the number of bytes: Byte 1 is 16#01; byte 2 is 16#02; byte 4 is 16#04 and so on.

✧ Output variable

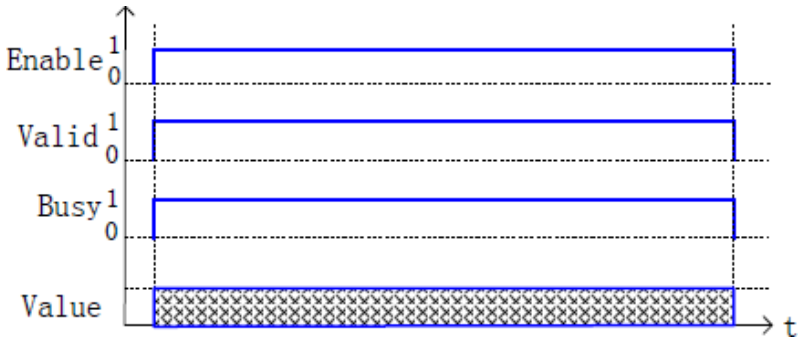
Output variable	Name	Data type	Valid range	Initial value	Description
Valid	Obtainable flag of position data	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the drive position can be obtained correctly.
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.

Output variable	Name	Data type	Valid range	Initial value	Description
Value	Obtained axis position.	BOOL	TRUE, FALSE	FALSE	The axis error code is read.

(3) Function description

Bit data status is read from the drive by executing MC_ReadBoolParam, which is valid when Enable is TRUE. The function block can be repeatedly executed, without mutual affecting. When Valid is TRUE, the bit status data is valid; when Busy is TRUE, the function block is being executed.

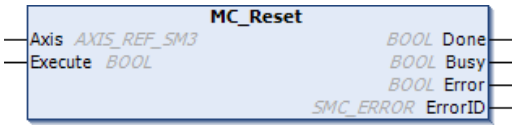
(4) Timing diagram



2.1.16 MC_Reset

MC_Reset: used to reset all errors of an axis.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
MC_Reset	Axis error reset command		MC_Reset(Axis:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Execute	Execution condition	BOOL	TRUE, FALSE	FALSE	The rising edge will start the processing of the function block.

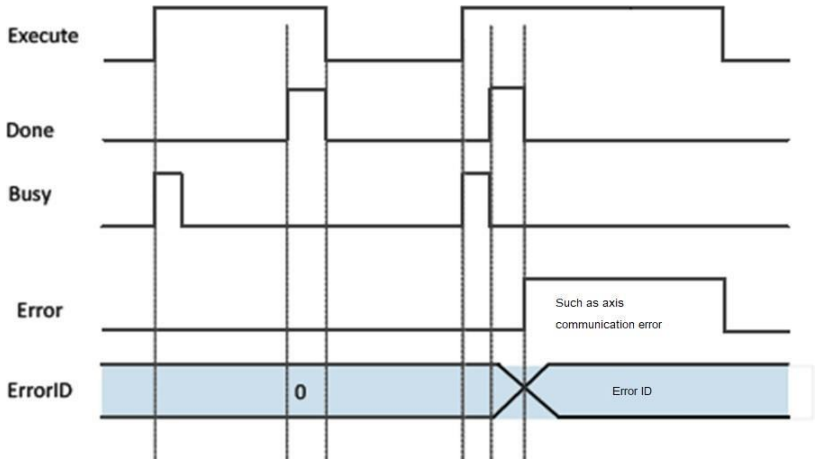
✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Command execution completion	BOOL	TRUE, FALSE	FALSE	It is set to TRUE after the axis command is executed completely.
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.

(3) Function description

This function can change the axis status from Errorstop to Standstill when the axis is in normal communication. If the axis cannot be reset from the Errostop state and Axis.bCommunication is FALSE, you must re-establish the communication between the master and slave axes.

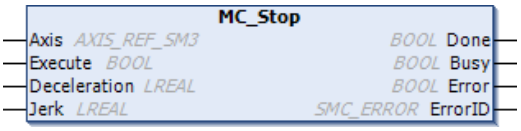
(4) Timing diagram



2.1.17 MC_Stop

MC_Stop: used to instruct an axis to decelerate to stop.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
MC_Stop	Axis stop command		MC_Stop(Axis:= , Execute:= , Deceleration:= , Jerk:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Execute	Execution condition	BOOL	TRUE, FALSE	FALSE	The rising edge will start the processing of the function block.
Deceleration	Deceleration	LREAL	Positive number or 0	0	Function block deceleration speed (μ/S^2)
Jerk	Jump	LREAL	Positive number or 0	0	Specified jump [Command unit/S^3]

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Command execution completion	BOOL	TRUE, FALSE	FALSE	It is set to TRUE after the axis command is executed completely.
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.

(3) Function description

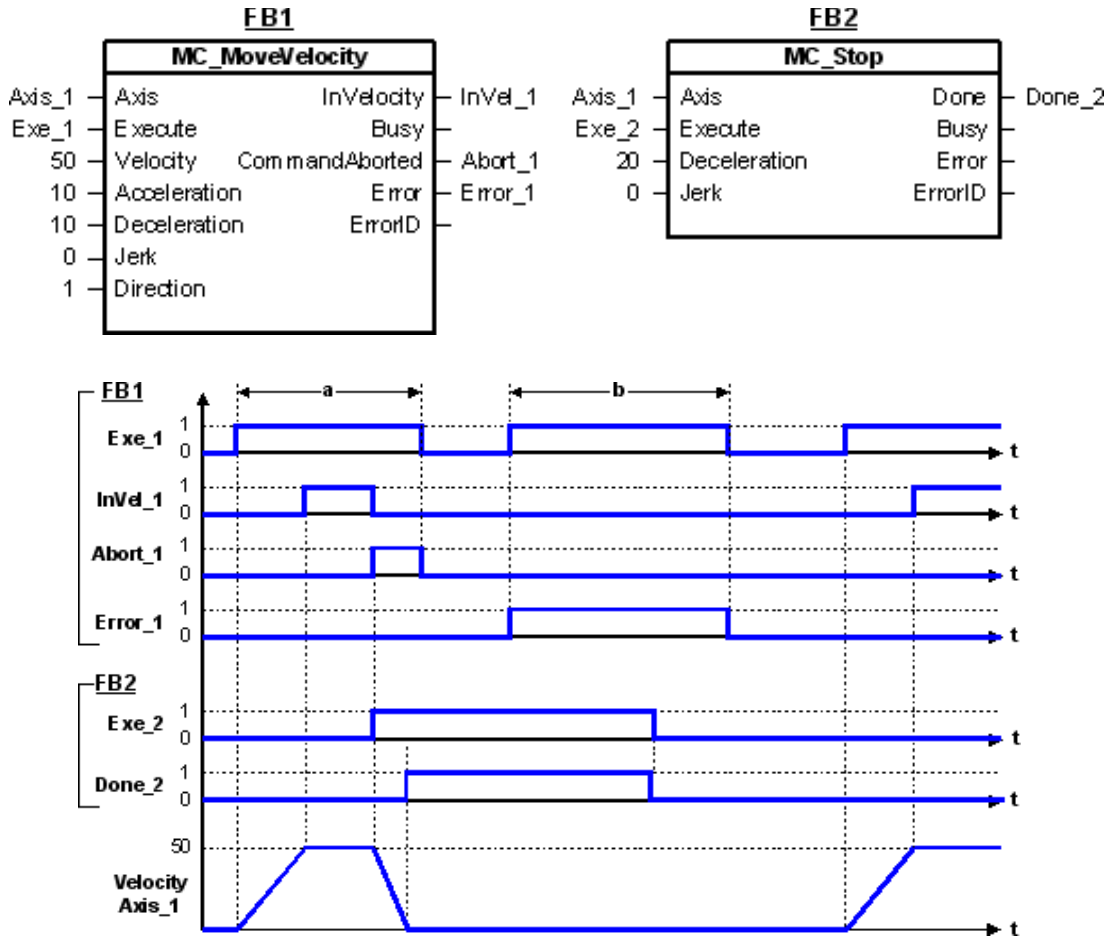
This function block is used to stop the motion of an axis that is in normal running. It does not take effect to the axis when it is in the Stopping state.

If the axis is in the Stopping state, Execute is Flase, and Done is TRUE, and the axis status changes to Standstill. The function block can be executed only when the axis is in the Motion state. The function block is started at the rising edge. If Busy is TRUE when MC_Stop is in the execution process, the restart of MC_Stop will cause the axis to enter the Errorstop state.

(4) Timing diagram

✧ Example

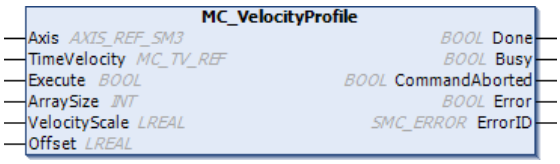
Flag bit difference in executing MC_MoveVelocity and MC_Stop:



2.1.18 MC_VelocityProfile

MC_VelocityProfile: indicates the motion model of the time segment and speed profile.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
MC_ VelocityProfile	Speed profile command		MC_VelocityProfile(Axis:= , TimeVelocity:= , Execute:= , ArraySize:= , VelocityScale:= , Offset:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.
TimeVelocity	Running	MC_TV_REF			Runing time and speed data

I/O variable	Name	Data type	Valid range	Initial value	Description
	time and description of axis				description of the axis. The data consists of multiple groups of data.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Execute	Execution condition	BOOL	TRUE, FALSE	FALSE	The rising edge will start the processing of the function block.
ArraySize	Dynamic arrays	INT		0	Number of arrays used in the motion profile.
VelocityScale	Speed factor	LREAL	Positive number or 0	1	Speed scaling factor.
Offset	Offset	LREAL		0	Overall offset value of the speed.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Command execution completion	BOOL	TRUE, FALSE	FALSE	It is set to TRUE after the axis command is executed completely.
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.

(3) Function description

This function block is used to specify the motion model of the time segment and speed profile. The axis running mode is Continuous Motion, and the function block uses the data in TimeVelocity. The axis must be in the Standstill state before the function block execution and in the Discrete Motion state during the execution. This function block is started at the rising edge. This function block can be repeatedly executed when the axis is in the Discrete Motion state. TimeVelocity is of the MC_TV_REF data type.

MC_TV_REF is described as follows:

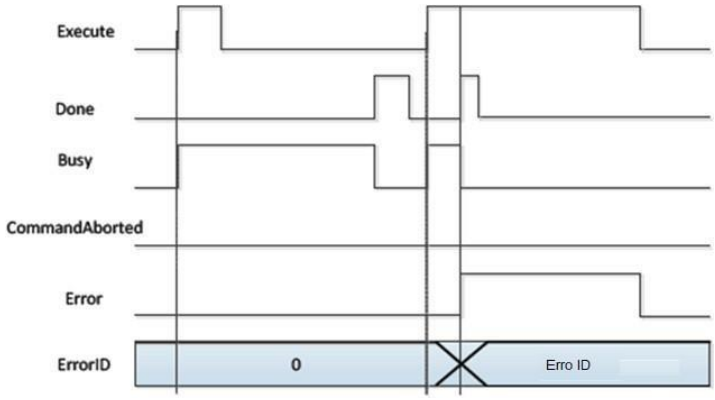
Member	Type	Initial value	Description
Number_of_pairs	INT	0	Number of profile path segments.
IsAbsolute	BOOL	TRUE	Absolute motion (TRUE) or relative motion (FALSE).
MC_TV_Array	ARRAY[1..N] OF SMC_TV		Data arrays of time and speed.

SMC_TV is described as follows:

Member	Type	Initial value	Description
delta_time	TIME	TIME#0ms	Time of a speed segment.
Velocity	LREAL	0	Speed that is recorded currently.

Note: The entire speed process represents the S curve with acceleration and deceleration, and the speed of each profile segment is calculated by superimposition; during repeated running, the speed is also superimposed to avoid the occurrence of speed limit exceeding; before repeated running, the axis status must be set to Standstill.

(4) Timing diagram



2.1.19 MC_WriteBoolParameter

MC_WriteBoolParameter: used to set the bit parameters of the drive axis.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

MC_ WriteBool Parameter	Axis bit parameter setting		`MC_WriteBoolParameter(`
			`Axis:= ,`
			`Execute:= ,`
			`ParameterNumber:= ,`
			`Value:= ,`
			`Done=> ,`
			`Busy=> ,`
			`Error=> ,`
			`ErrorID=> );`

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Enable	Execution condition	BOOL	TRUE, FALSE	FALSE	When it is set to TRUE, this function block is started.

Input variable	Name	Data type	Valid range	Initial value	Description
ParameterNumber	Axis parameter number	DINT		0	Access index, sub-index, and number of the axis parameter.
Value	Setting	BOOL	TRUE, FALSE	FALSE	Used to set the bit parameters.

Note:

ParameterNumber (DINT) = -DWORD_TO_DINT(SHL(USINT_TO_DOWRD(usiDataLength), 24) (Data length in the object dictionary)

+ SHL(UINT_TO_DWORD(uiIndex), 8) (Index in the object dictionary-16 bits)

+ usisubIndex (Sub-index in the object dictionary-8 bits)

usiDataLength: Fill in according to the number of bytes: Byte 1 is 16#01; byte 2 is 16#02; byte 4 is 16#04 and so on.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Setting result	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the setting operation succeeds.
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.

(3) Function description

Axis bit parameters are set by executing MC_WriteBoolParameter, which is started at the rising edge. This function block can be repeatedly executed, without mutual affecting.

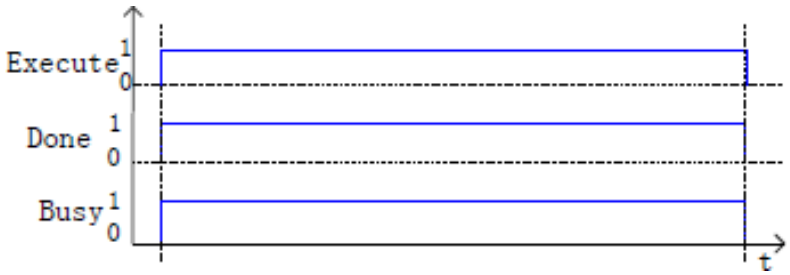
(4) Timing diagram

The function block can be triggered only at the rising edge.

When Done is TRUE, the setting operation is successful.

When Busy is TRUE, the function block is being executed.

✧ Timing description



2.1.20 MC_WriteParameter

MC_WriteParameter: used to set the bit parameters of the drive axis.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
MC_ Write Parameter	Axis parameter setting		MC_WriteParameter(Axis:= , Execute:= , ParameterNumber:= , Value:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Enable	Execution condition	BOOL	TRUE, FALSE	FALSE	When it is set to TRUE, this function block is started.
ParameterNumber	Axis parameter number	DINT		0	Access index, sub-index, and number of the axis parameter.
Value	Setting	BOOL	TRUE, FALSE	FALSE	Used to set the bit parameters.

Note:

ParameterNumber (DINT) = -DWORD_TO_DINT(SHL(USINT_TO_DOWRD(usiDataLength), 24) (Data length in the object dictionary)

+ SHL(UINT_TO_DWORD(uiIndex), 8) (Index in the object dictionary-16 bits)

+ usisubIndex (Sub-index in the object dictionary-8 bits)

usiDataLength: Fill in according to the number of bytes: Byte 1 is 16#01; byte 2 is 16#02; byte 4 is 16#04 and so on.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Setting result	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the setting operation succeeds.
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.

Output variable	Name	Data type	Valid range	Initial value	Description
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.

(3) Function description

Axis bit parameters are set by executing MC_WriteParameter, which is started at the rising edge. This function block can be repeatedly executed, without mutual affecting.

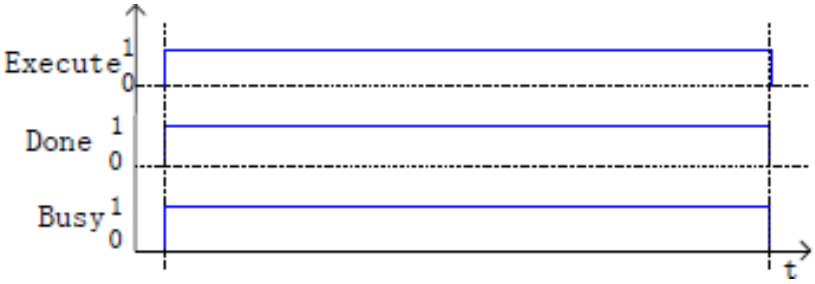
(4) Timing diagram

The Execute of the function block must be the rising edge trigger condition.

When Done is TRUE, the setting operation is successful.

When Busy is TRUE, the function block is being executed.

✧ Timing description



2.1.21 MC_AbortTrigger

MC_AbortTrigger: used to terminate the association features of input latch related events, in conjunction with MC_Touchprobe.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
MC_ AbortTrigger	Event association terminating		MC_AbortTrigger(Axis:= , TriggerInput:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.
TriggerInput	Trigger signal	TRIGGER_REF			Description such as trigger signal and attributes.

✧ TRIGGER_REF description:

I/O variable	Name	Data type	Initial value	Description
TRIGGER_REF	iTriggerNumber	INT	-1	Used to select a function to latch in the drive mode. 0: Rising edge latching for probe 1 1: Falling edge latching for probe 1 2: Rising edge latching for probe 2 3: Falling edge latching for probe 2
	bFastLatching	BOOL	TRUE	Used to specify the latching trigger mode. TRUE: Drive mode FALSE: Controller mode
	bInput	BOOL		When bFastLatching=FALSE, the controller inputs a signal for trigger.
	bActive	BOOL		Valid signal for trigger.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Execute	Execution condition	BOOL	TRUE, FALSE	FALSE	The rising edge trigger will start the processing of the function block.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Setting result	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the setting operation succeeds.
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.

(3) Function description

The MC_AbortTrigger function block is used to terminate the association between the trigger signal/attribute and the related trigger command. The Execute of the function block must be the rising edge trigger condition. When Done is TRUE, the setting operation is successful; when Busy is TRUE, the function block is being executed.

2.1.22 MC_ReadActualTorque

MC_ReadActualTorque: used to read the actual torque of the drive and save it to a user-defined variable.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
MC_ ReadActual Torque	Actual torque reading command		MC_ReadActualTorque(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , Torque=>);

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Enable	Execution condition	BOOL	TRUE, FALSE	FALSE	When it is set to TRUE, this function block is started.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Valid	Actual torque obtaining flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the drive torque can be obtained correctly.
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR	-	0	When an error occurs, the error ID is output.
Torque	Actual torque obtaining	LREAL	Torque	0	Actual torque data that is read.

(3) Function description

Actual torque data is read by executing MC_ReadActualTorque, which is valid when the electrical level is enabled. This function block can be repeatedly executed, without mutual affecting.

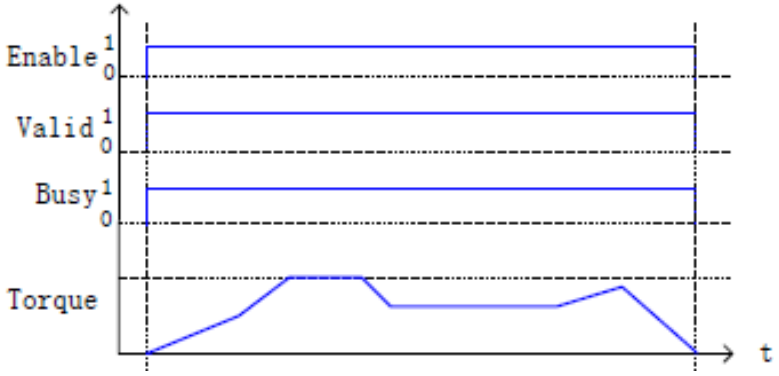
(4) Timing diagram

Enable must be TRUE.

When Valid=TRUE, the read torque is valid.

When Busy is TRUE, the function block is being executed.

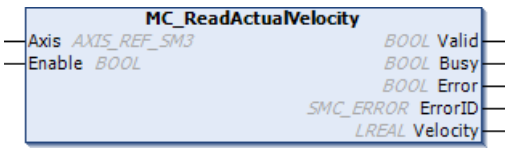
✧ Timing description



2.1.23 MC_ReadActualVelocity

MC_ReadActualVelocity: used to read the actual speed of the drive and save it to a user-defined variable.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
MC_ ReadActualVelocity	Actual speed reading command		MC_ReadActualVelocity(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , Velocity=>);

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Enable	Execution condition	BOOL	TRUE, FALSE	FALSE	When it is set to TRUE, this function block is started.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Valid	Actual torque obtaining flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the drive torque can be obtained correctly.
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.

Output variable	Name	Data type	Valid range	Initial value	Description
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR	-	0	When an error occurs, the error ID is output.
Velocity	Actual speed obtaining	LREAL	Speed	0	Actual speed data that is read.

(3) Function description

Actual speed data is read by executing MC_ReadActualVelocity, which is valid when the electrical level is enabled. This function block can be repeatedly executed, without mutual affecting.

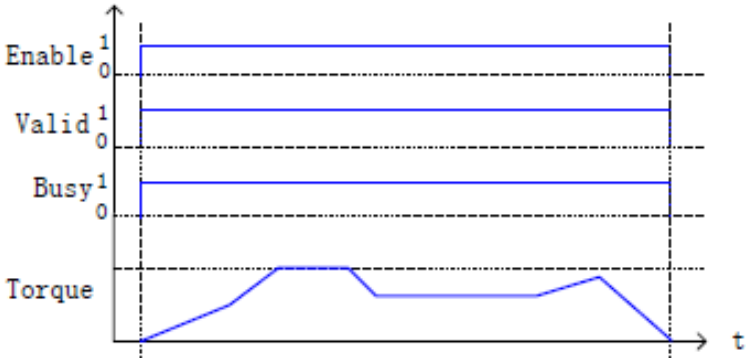
(4) Timing diagram

Enable must be TRUE.

When Valid=TRUE, the read torque is valid.

When Busy is TRUE, the function block is being executed.

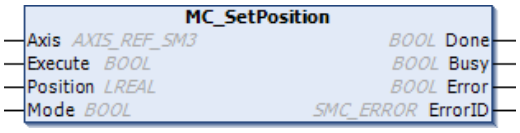
✧ Timing description



2.1.24 MC_SetPosition

MC_SetPosition: used to set the position data in the command as the position data of an axis, without causing any movement for setting position data. It is designed for shifting the coordinate system of an axis.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
MC_ SetPosition	Position change		MC_SetPosition(Axis:= , Execute:= , Position:= , Mode:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Execute	Execution condition	BOOL	TRUE, FALSE	FALSE	The rising edge will start the processing of the function block.
Position	Axis position data	LREAL	-	0	Position data.
Mode	Setting	BOOL	TRUE, FALSE	FALSE	Position mode. TRUE: relative FALSE: absolute

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR	-	0	When an error occurs, the error ID is output.
Velocity	Actual speed obtaining	LREAL	Speed	0	Actual speed data that is read.

(3) Function description

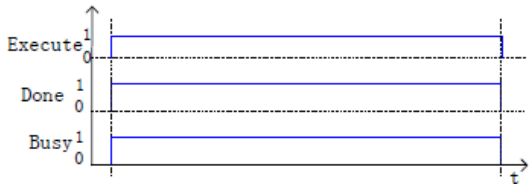
Axis position parameters are set by executing MC_SetPosition, without any movement caused but with cordinate system offset caused. This function block is started at the rising edge and it can be repeatedly executed without mutual affecting.

(4) Timing diagram

The Execute of the function block must be the rising edge trigger condition.

When Done is TRUE, the setting operation is successful. When Busy is TRUE, the function block is being executed.

✧ Timing description

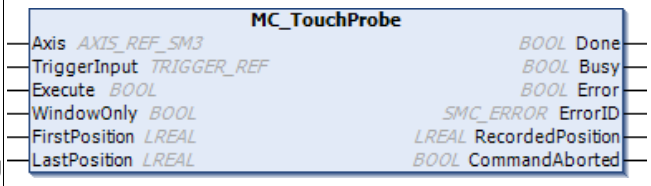


2.1.25 MC_TouchProbe

MC_TouchProbe: used to save the axis position when an external signal triggers the command.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

MC_ TouchProbe	External locking enabling		`MC_TouchProbe(`
			`Axis:= ,`
			`TriggerInput:= ,`
			`Execute:= ,`
			`WindowOnly:= ,`
			`FirstPosition:= ,`
			`LastPosition:= ,`
			`Done=> ,`
			`Busy=> ,`
			`Error=> ,`
			`ErrorID=> ,`
			`RecordedPosition=> ,`
			`CommandAborted=> );`

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.
TriggerInput	Trigger signal	TRIGGER_ REF			Association attributes such as trigger signal and attributes.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Execute	Execution condition	BOOL	TRUE, FALSE	FALSE	The rising edge trigger will start the processing of the function block.
WindowOnly	Trigger window	BOOL	TRUE, FALSE	FALSE	
FirstPosition	Trigger start position	LREAL		0	Used to specify the start position for receiving trigger.
LastPosition	Trigger end position	LREAL		0	Used to specify the end position for receiving trigger.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Setting success	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when setting succeeds.
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.

Output variable	Name	Data type	Valid range	Initial value	Description
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.
RecordedPosition	Trigger recording position	LREAL			Position where the trigger occurs.
CommandAbort	Command interruption	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is interrupted.

(3) Function description

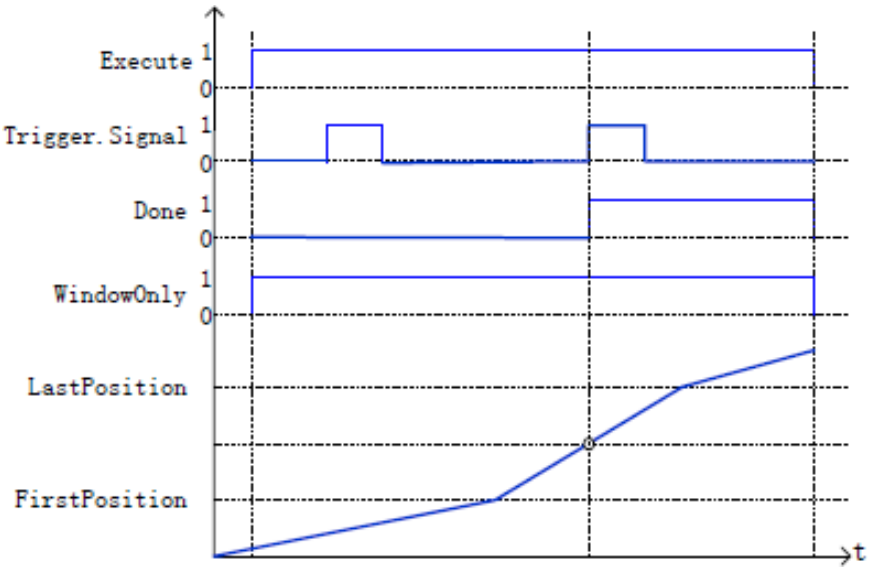
The actual position of the axis is recorded when the MC_TouchProbe function block is triggered through TriggerInput. When the rising edge executes drive latching, the latching signal collected by the drive is in the recording position.

(4) Timing diagram

The Execute of the function block must be the rising edge trigger condition.

When Done is TRUE, the setting operation is successful.

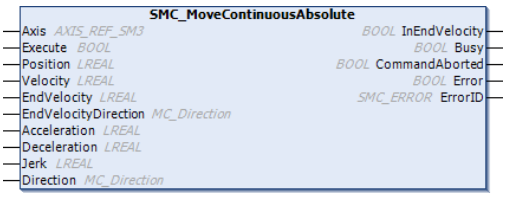
✧ Timing description



2.1.26 MC_MoveContinuousAbsolute

MC_MoveContinuousAbsolute: used to specify that an axis runs at the continous absolute speed (the unit is axis depended). The absolute position is specified by Distance and running end speed is specified by EndVelocity.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
MC_ MoveContinuousAbsolute	Axis absolute position continuous control command		SMC_MoveContinuousAbsolute(Axis:= , Execute:= , Position:= , Velocity:= , EndVelocity:= , EndVelocityDirection:= , Acceleration:= , Deceleration:= , Jerk:= , Direction:= , InEndVelocity=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Execute	Execution condition	BOOL	TRUE, FALSE	FALSE	The rising edge will start the processing of the function block.
Distance	Relative position of motion	LREAL	Data range	0	Relative position of motion.
Velocity	Running speed	LREAL	Data range	0	Max. speed at which the axis runs to reach the destination position.
EndVelocity	Running end speed	LREAL	Data range	0	Running speed after command execution.
EndVelocity-Direction	Direction of running at end speed	MC_Direction	positive, negative, current;	Current	Options: positive, negative, current; Not allowed: shortest, fastest
Acceleration	Acceleration	LREAL	Data range	0	Acceleration when the speed increases.
Deceleration	Deceleration	LREAL	Data range	0	Deceleration when the speed reduces.

Input variable	Name	Data type	Valid range	Initial value	Description
Direction	Running direction	shortest	Data range	shortest	For linear/straight-line axes: positive, negative; For rotary/circular axes: positive, negative, current, shortest, fastest

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
InEndVelocity	Command position reaching	BOOL	TRUE, FALSE	FALSE	It is set to TRUE after the position in the command is reached.
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.
CommandAbort	Command interruption	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is interrupted.

(3) Function description

This function block is the axis absolute position command, in which Distance specifies the axis absolute position. The axis must be in the Standstill state before the function block execution and in the Discrete Motion state during the execution. The axis status must be controlled throughout the complete running process. The startup command is Execute, and the rising edge triggers the function block. The startup command can repeatedly make the rising edge valid when the axis is in the Discrete Motion state, which always refreshes the position. When Acceleration or Deceleration is 0, the command execution is abnormal, but the axis is in the Discrete Motion state.

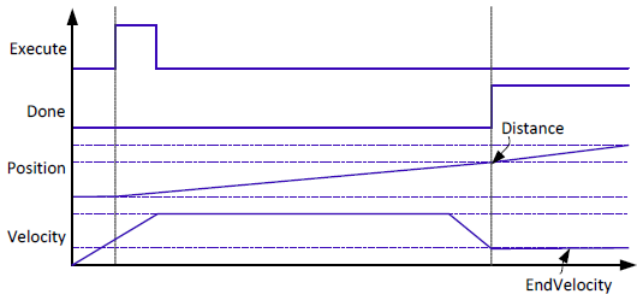
(4) Timing diagram

The function block can be executed only when the axis is in the Standstill state.

The Execute of the function block must be the rising edge trigger condition.

When Done is TRUE, the setting operation is successful. When Busy is TRUE, the function block is being executed.

✧ Timing description



2.1.27 MC_MoveContinuousRelative

MC_MoveContinuous Relative: used to specify that an axis runs at the continuous absolute speed (the unit is axis depended). The absolute position is specified by Distance and running end speed is specified by EndVelocity.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
MC_ MoveContinuousRelative	Axis absolute position continuous control command		SMC_MoveContinuousRelative (Axis:= , Execute:= , Distance:= , Velocity:= , EndVelocity:= , EndVelocityDirection: = , Acceleration:= , Deceleration:= , Jerk:= , InEndVelocity=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Execute	Execution condition	BOOL	TRUE, FALSE	FALSE	The rising edge will start the processing of the function block.
Distance	Relative position of motion	LREAL	Data range	0	Relative position of motion.
Velocity	Running speed	LREAL	Data range	0	Max. speed at which the axis runs to reach the destination position.
EndVelocity	Running end speed	LREAL	Data range	0	Running speed after command execution.
EndVelocity-Direction	Direction of running at end speed	MC_Direction	Positive, Negative, current	Current	Options: positive, negative, current; Not allowed: shortest, fastest
Acceleration	Acceleration	LREAL	Data range	0	Acceleration when the speed increases.
Deceleration	Deceleration	LREAL	Data range	0	Deceleration when the speed reduces.

Input variable	Name	Data type	Valid range	Initial value	Description
Direction	Running direction	shortest	Data range	shortest	For linear/straight-line axes: positive, negative; For rotary/circular axes: positive, negative, current, shortest, fastest

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
InEndVelocity	Command position reaching	BOOL	TRUE, FALSE	FALSE	It is set to TRUE after the position in the command is reached.
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.
CommandAbort	Command interruption	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is interrupted.

(3) Function description

The axis must be in the Standstill state before the function block execution and in the Discrete Motion state during the execution. Pay attention to the axis status during the execution to prevent other commands from interrupting the command execution of the axis. The startup command is Execute, and the rising edge triggers the function block. The startup command can repeatedly make the rising edge valid when the axis is in the Discrete Motion state, which always refreshes the position. When Acceleration or Deceleration is 0, the command execution is abnormal, but the axis is in the Discrete Motion state.

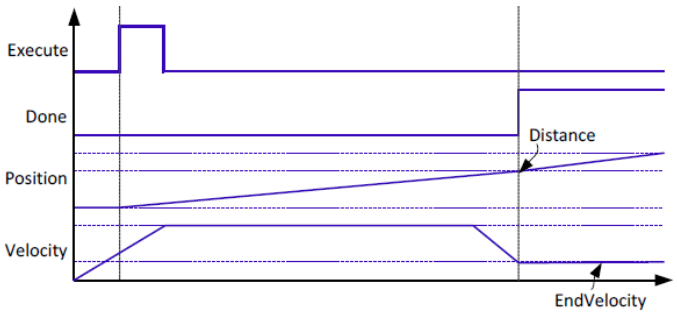
(4) Timing diagram

The function block can be executed only when the axis is in the Standstill state.

The Execute of the function block must be the rising edge trigger condition.

When Done is TRUE, the setting operation is successful. When Busy is TRUE, the function block is being executed.

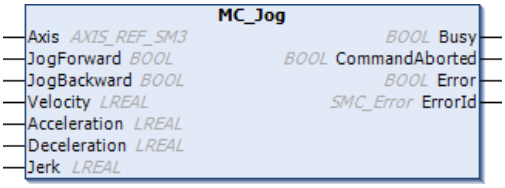
✧ Timing description



2.1.28 MC_Jog

MC_Jog: used to instruct an axis to jog at a specified speed.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
MC_Jog	Axis jogging command		MC_Jog(Axis:= , JogForward:= , JogBackward:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Busy=> , CommandAborted=> , Error=> , ErrorId=>);

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
JogForward	Valid at forward jogging	BOOL	TRUE, FALSE	FALSE	If it is TRUE, the axis moves forward. If it is FALSE, the axis stops moving forward.
JogBackward	Valid at reverse jogging	BOOL	TRUE, FALSE	FALSE	If it is TRUE, the axis moves reversely. If it is FALSE, the axis stops moving reversely.
Velocity	Target speed	LREAL	Positive number or 0	0	Target speed. Unit: [Command unit/s]
Acceleration	Acceleration	LREAL	Positive number or 0	0	Acceleration. Unit: [Command unit/s]
Deceleration	Deceleration	LREAL	Positive number or 0	0	Deceleration. Unit: [Command unit/s]
Jerk	Jump	LREAL	Data range	0	Slope change value of the curve acceleration or deceleration.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.
CommandAborted	Command interruption	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is interrupted.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.

(3) Function description

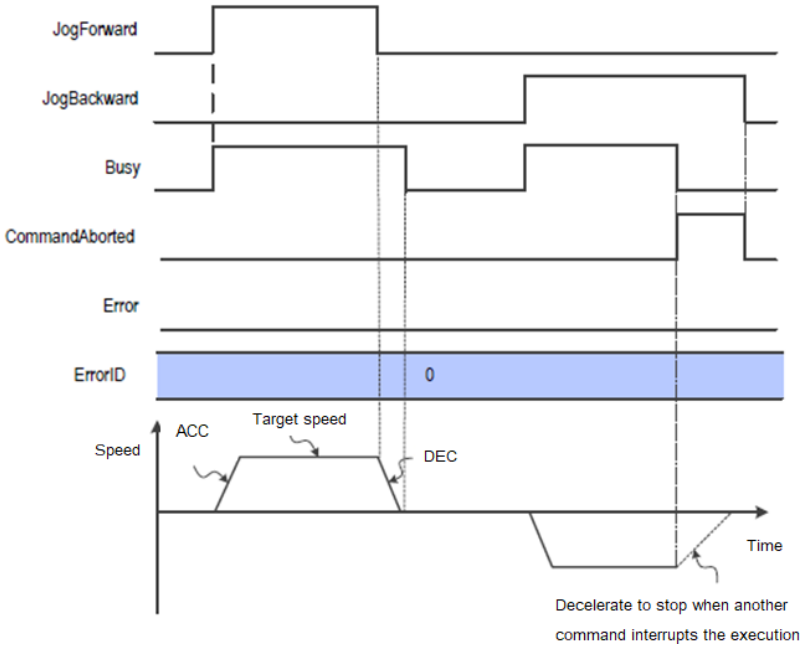
The function block is used to instruct the axis to jog at the target speed. When the axis needs to run forward, set JogForward to TRUE; when the axis needs to run forward, set JogBackward to TRUE. When both JogForward and JogBackward are set to TRUE at the same time, the axis does not move. If the speed value in MC_Jog exceeds the max. jogging speed in the axis parameters, the axis moves at the max. jogging speed.

(4) Timing diagram

When JogForward or JogBackward is set to TRUE, the value of Busy changes to TRUE; when the falling edge of JogForward or JogBackward starts deceleration and stops the axis also, the value of Busy changes to FALSE.

If another command is used to terminate the execution of this function block, the value of CommandAborted changes to TRUE, and the value of Busy changes to FALSE.

✧ Timing description

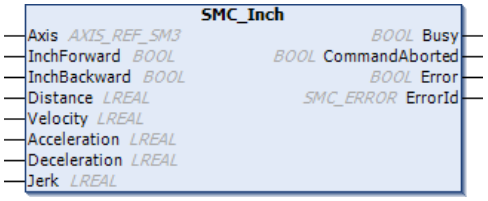


2.1.29 MC_Inch

MC_Inch: used to cause a gradual motion on an axis, which is carried out step by step.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

MC_Inch	Axis relative positioning command		`SMC_Inch(`
			`Axis:= ,`
			`InchForward:= ,`
			`InchBackward:= ,`
			`Distance:= ,`
			`Velocity:= ,`
			`Acceleration:= ,`
			`Deceleration:= ,`
			`Jerk:= ,`
			`Busy=> ,`
			`CommandAborted=> ,`
			`Error=> ,`
			`ErrorId=> );`

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
InchForward	Forward inching	BOOL	TRUE, FALSE	FALSE	If InchForward is TRUE, the axis runs at the given speed in the forward direction until it reaches the destination. The input must be set to FALSE and then TRUE to restart the running. If InchForward is set to FALSE before the destination is reached, the axis decelerates to 0 at once, and Busy is set to FALSE. If InchBackward is set to TRUE in simulation mode, the axis does not move.
InchBackward	Reverse inching	BOOL	TRUE, FALSE	FALSE	If InchBackward is TRUE, the axis runs at the given speed in the reverse direction until it reaches the destination. The input must be set to FALSE and then TRUE to restart the running. If both InchBackward and InchForward

Input variable	Name	Data type	Valid range	Initial value	Description
					are set to TRUE at the same time, the axis does not move.
Distance	Moving distance	LREAL	Data range	0	Moving distance.
Velocity	Running speed	LREAL	Data range	0	Max. speed at which the axis runs to reach the destination position.
Acceleration	Acceleration	LREAL	Data range	0	Acceleration when the speed increases.
Deceleration	Deceleration	LREAL	Data range	0	Deceleration when the speed reduces.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.
CommandAborted	Command interruption	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is interrupted.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.

(3) Function description

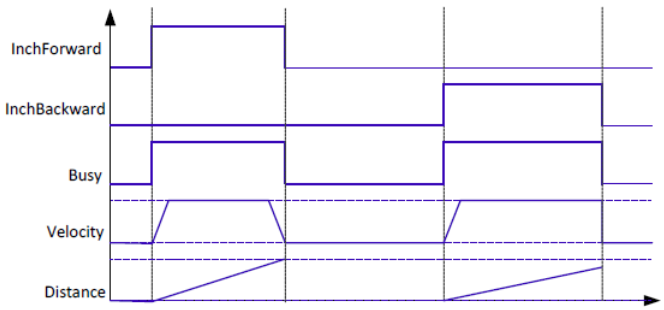
The axis must be in the Standstill state before the function block execution and in the Discrete Motion state during the execution. Pay attention to the axis status during the execution to prevent other commands from interrupting the command execution of the axis. When Acceleration or Deceleration is 0, the command execution is abnormal, but the axis is in the Discrete Motion state.

(4) Timing diagram

InchForward and InchBackward must be set to TRUE or FALSE.

When Busy is TRUE, the function block is being executed.

✧ Timing description

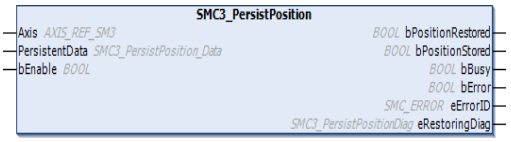


2.1.30 SMC3_PersistPosition

SMC3_PersistPosition: used to persist the axis position of a multi-turn absolute encoder with real axis. (The controller that is restarted due to power failure uses the position recorded before the power failure.) If the servo motor uses an absolute encoder, use this function block.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

SMC3_PersistPosition	Axis position persisting		`SMC3_PersistPosition(`
			`Axis:= ,`
			`PersistentData:= ,`
			`bEnable:= ,`
			`bPositionRestored=> ,`
			`bPositionStored=> ,`
			`bBusy=> ,`
			`bError=> ,`
			`eErrorID=> ,`
			`eRestoringDiag=> );`

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.
PersistentData	Data to persist	SMC3_PersistPosition_Data			Structure of position data stored at power failure.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Enable	Enabling	BOOL	TRUE, FALSE	FALSE	TRUE indicates executing the function block, while FALSE indicates not executing the function block.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
bPositionRestored	Position restoring	BOOL			TRUE indicates the position data is restored after the axis restart.
bPositionStored	Position data storing	BOOL			TRUE indicates the position data is stored after the function block is invoked.
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.

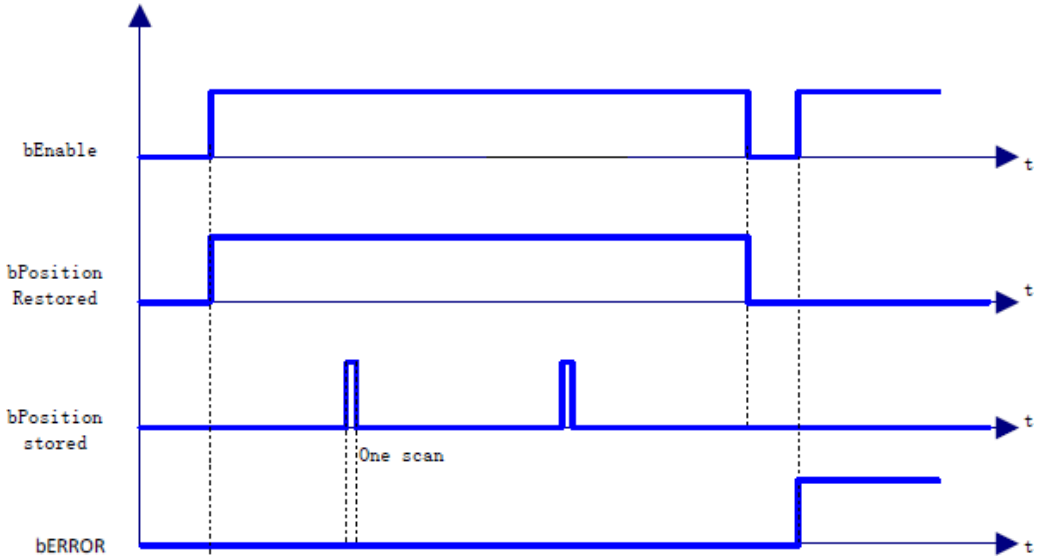
Output variable	Name	Data type	Valid range	Initial value	Description
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.
eRestoringDiag	Diagnosis restoring	SMC3_Persist-PositionDiag			Diagnosis information for position restoring SMC3_PPD_RESTORING_OK: Position restoring success SMC3_PPD_AXIS_PROP_CHANGED: Failed to restore the position due to axis parameter changes. SMC3_PPD_DATA_STORED_DURING_WRITING: The function block copies data from the axis data structure but not from PersistentData. Possible causes: Asynchronous persistent variables, and controller crash

(3) Function description

When the PLC is restarted and bEnable is TRUE, bPositionRestroed is TRUE.

(4) Timing diagram

When Busy is TRUE, the function block is being executed.



2.1.31 SMC3_PersistPositionSingleturn

SMC3_PersistPositionSingleturn: used to persist the axis position of a single-turn absolute encoder with real axis. (The controller that is restarted due to power failure uses the position recorded before the power failure.) If the servo motor uses a single-turn absolute encoder, use this function block.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

SMC3_Persisit Position Singleturn	Axis position persisting		`SMC3_PersistPositionSingleturn(`
			`Axis:= ,`
			`PersistentData:= ,`
			`bEnable:= ,`
			`usiNumberOfAbsoluteBits:= ,`
			`bPositionRestored=> ,`
			`bPositionStored=> ,`
			`bBusy=> ,`
			`bError=> ,`
			`eErrorID=> ,`
			`eRestoringDiag=> );`

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.
PersistentData	Data to persist	SMC3_PersistPosition_Data			Structure of position data stored at power failure.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Enable	Enabling	BOOL	TRUE, FALSE	FALSE	TRUE indicates executing the function block, while FALSE indicates not executing the function block.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
bPositionRestored	Position restoring	BOOL			TRUE indicates the position data is restored after the axis restart.
bPositionStored	Position data storing	BOOL			TRUE indicates the position data is stored after the function block is invoked.
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.

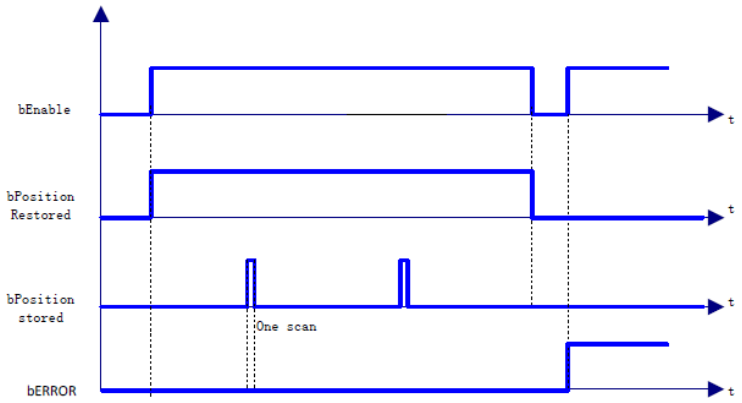
Output variable	Name	Data type	Valid range	Initial value	Description
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.
eRestoringDiag	Diagnosis restoring	SMC3_Persist-PositionDiag			Diagnosis information in position restoring SMC3_PPD_RESTORING_OK: Position restoring success SMC3_PPD_AXIS_PROP_CHANGED: Failed to restore the position due to axis parameter changes. SMC3_PPD_DATA_STORED_DURING_WRITING: The function block copies data from the axis data structure but not from PersistentData. Possible causes: Asynchronous persistent variables, and controller crash

(3) Function description

When the PLC is restarted and bEnable is TRUE, bPositionRestroed is TRUE.

(4) Timing diagram

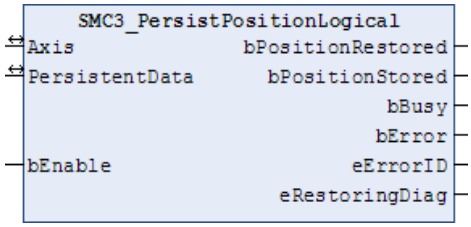
When Busy is TRUE, the function block is being executed.



2.1.32 SMC3_PersistPosition Logical

used to persist the axis position of a single-turn absolute encoder with real axis. (The controller that is restarted due to power failure uses the position recorded before the power failure.) If the servo motor uses a single-turn absolute encoder, use this function block.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
SMC3_PersisitPosition Logical	Axis position persisting		SMC3_PersistPositionLogical(Axis:= , PersistentData:= , bEnable:= , bPositionRestored=> , bPositionStored=> , bBusy=> , bError=> , eErrorID=> , eRestoringDiag=>);

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.
PersistentData	Data to persist	SMC3_PersistPosition_Data			Structure of position data stored at power failure.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Enable	Enabling	BOOL	TRUE, FALSE	FALSE	TRUE indicates executing the function block, while FALSE indicates not executing the function block.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
bPositionRestored	Position restoring	BOOL			TRUE indicates the position data is restored after the axis restart.
bPositionStored	Position data storing	BOOL			TRUE indicates the position data is stored after the function block is invoked.
Busy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is being executed.

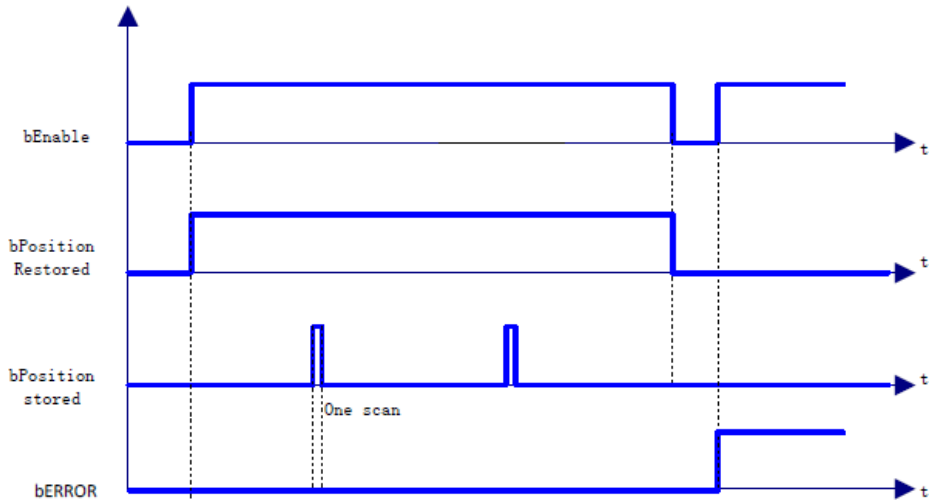
Output variable	Name	Data type	Valid range	Initial value	Description
Error	Error flag	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.
eRestoringDiag	Diagnosis restoring	SMC3_Persist-PositionDiag			Diagnosis information in position restoring SMC3_PPD_RESTORING_OK: Position restoring success SMC3_PPD_AXIS_PROP_CHANGED: Failed to restore the position due to axis parameter changes. SMC3_PPD_DATA_STORED_DURING_WRITING: The function block copies data from the axis data structure but not from PersistentData. Possible causes: Asynchronous persistent variables, and controller crash

(3) Function description

When the PLC is restarted and bEnable is TRUE, bPositionRestroed is TRUE.

(4) Timing diagram

When Busy is TRUE, the function block is being executed.

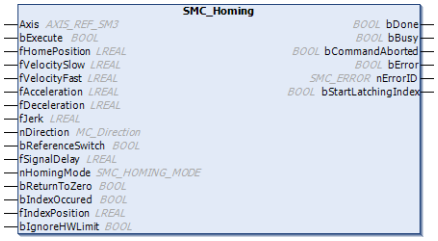


2.1.33 SMC_Homing

Axis homing command, different from MC_Home. MC_Home specifies the homing mode controlled by the servo controller, while SMC_Homing specifies the homing mode controlled by the PLC.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

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

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
bExecute	Enabling	BOOL	TRUE, FALSE	FALSE	TRUE indicates executing the function block, while FALSE indicates not executing the function block.
fHomePosition	Home position	LREAL		0	Home position after zeroing, using the unit after user calibration.
fVelocitySlow	Low speed	LREAL		0	Used to drive out of the reference switch.
fVelocityFast	High speed	LREAL		0	Used until the reference switch is found.

Input variable	Name	Data type	Valid range	Initial value	Description
fAcceleration	Acceleration	LREAL		0	Acceleration setting
fDeceleration	Deceleration	LREAL		0	Deceleration setting
fJerk	Jerk value	LREAL		0	
nDirection	Homing direction	MC_DIRECTION		Negative	Homing start direction. Refer to MC_DIRECTION.
bReferenceSwitch	Reference switch	BOOL	TRUE, FALSE	FALSE	Reference switch status. TRUE: The reference switch is open. FALSE: The reference switch is closed.
fSignalDelay	Delay	LREAL		0	Reference switch transmission time, used to compensate for the deadzone time. Unit: second
nHomingMode	Homing mode	SMC_HOMING_MODE			Refer to SMC_HOMING_MODE.
bReturnTozero	Returning to position zero	BOOL	TRUE, FALSE	FALSE	TRUE: The axis moves to position zero after homing. Note: If fHomePosition=10, the axis position is 10 after homing, and when bReturnTozero is TRUE, the axis reversely moves by 10 units to position zero.
bIndexOccured		BOOL	TRUE, FALSE	FALSE	TRUE: Index pulse is detected. It is valid at the homing modes FAST_BSLOW_I_S_STOP and FAST_SLOW_I_S_STOP.
fIndexPosition		LREAL		0	Position where the index occurs.
bIgnoreHWLimit	Ignoring hardware position limit	BOOL	TRUE, FALSE	FALSE	TRUE: The hardware position limit switch is disabled. If the same physical switch is used both the hardware position limit switch and reference switch, hardware control is set to false.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
bDone		BOOL	TRUE, FALSE	FALSE	TRUE, homing completed
Busy		BOOL	TRUE, FALSE	FALSE	TRUE: The function block is being executed.
bCommandAborted		BOOL	TRUE, FALSE	FALSE	TRUE: The function block is interrupted by other action commands.
Error		BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
ErrorID		SMC_ERROR		0	When an error occurs, the error ID is output.
bStartLatchingIndex		BOOL	TRUE, FALSE	FALSE	Generated by bIndexOccured and fIndexPosition.

The homing modes are described as follows:

Mode	Type	Initial value	Description
FAST_BSLOW_S_STOP	SMC_HOMING_ MODE	0	The axis follows the set direction to the home switch at a high speed, and leaves the home switch at a low speed in the reverse direction after touching the home switch. After leaving, the controller executes MC_setPosition to set the present position to the setting of fHomePosition, and then executes MC_stop.
FAST_BSLOW_STOP_S	SMC_HOMING_ MOD	1	The axis follows the set direction to the home switch at a high speed, and leaves the home switch at a low speed in the reverse direction after touching the home switch. After leaving, the controller executes MC_stop to stop the axis, and then executes MC_setPosition to set the present position to the setting of fHomePosition.
FAST_BSLOW_I_S_STOP	SMC_HOMING_ MOD	2	The axis follows the set direction to the home switch at a high speed, and leaves the home switch at a low speed in the reverse direction after touching the home switch. When receiving the

Mode	Type	Initial value	Description
			bIndexOccured signal, the controller executes MC_setPosition and then MC_stop.
FAST_SLOW_S_STOP	SMC_HOMING_ MOD	4	The axis follows the set direction to the home switch at a high speed, and leaves the home switch at a low speed after touching the home switch. After leaving, the controller executes MC_setPosition to set the present position to the setting of fHomePosition, and then executes MC_stop.
FAST_SLOW_STOP_S	FAST_SLOW_STOP_S SMC_HOMING_ MOD	5	The axis follows the set direction to the home switch at a high speed, and leaves the home switch at a low speed after touching the home switch. After leaving, the controller executes MC_stop to stop the axis, and then executes MC_setPosition to set the present position to the setting of fHomePosition.
FAST_SLOW_I_S_STOP	SMC_HOMING_MOD	6	The axis follows the set direction to the home switch at a high speed, and leaves the home switch at a low speed in the reverse direction after touching the home switch. When receiving the bIndexOccured signal, the controller executes MC_setPosition and then MC_stop.

(3) Function description

After SMC_HOMING is started at the rising edge of bExecute, the axis moves at the speed specified by fVelocityFast in the direction specified by nDirection, which does not end until bReferenceSwitch = FALSE. The axis slowly stops and leaves the reference switch at the speed specified by fVelocitySlow in the reverse direction. When bReferenceSwitch = TRUE, homing is completed.

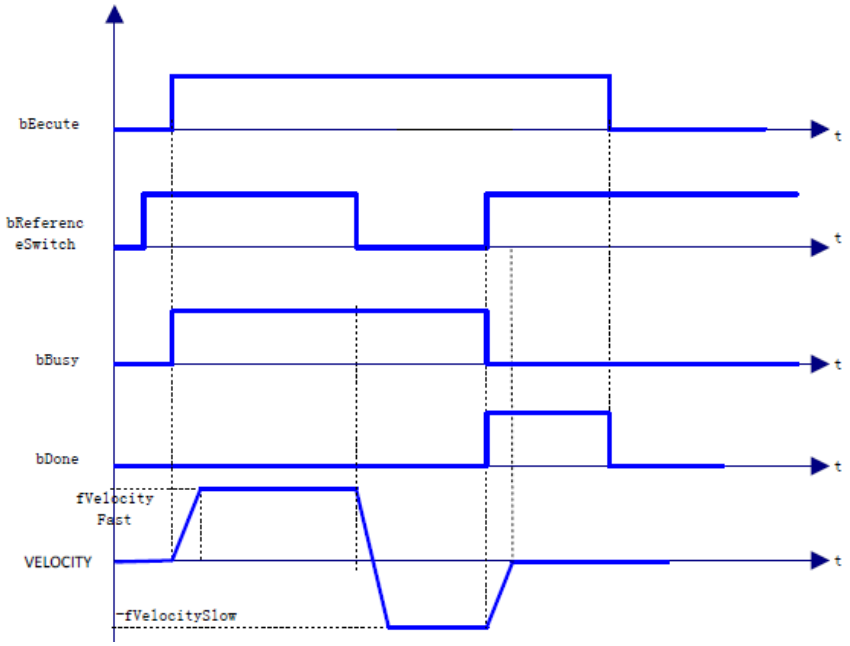
After the homing command is enabled, the status change sequence of bReferenceSwitch is ON->OFF->ON, the homing is completed on the rising edge of OFF->ON, and the reference position is set. Reference position = fHomePostion + [(fSignalDelay*1000 + 1 DC cycle)/1000] * fVelocitySlow, which actually compensates for the bReferenceSwitch sampling delay and one-communication-cycle displacement delay

If bReturnToZero=TRUE, the reference position is set to {fHomePostion + [(fSignalDelay*1000 + 1 DC cycle)/1000] * fVelocitySlow} at the rising edge of OFF->ON of bReferenceSwitch, the axis moves to position zero at the speed specified by fVelocityFast.

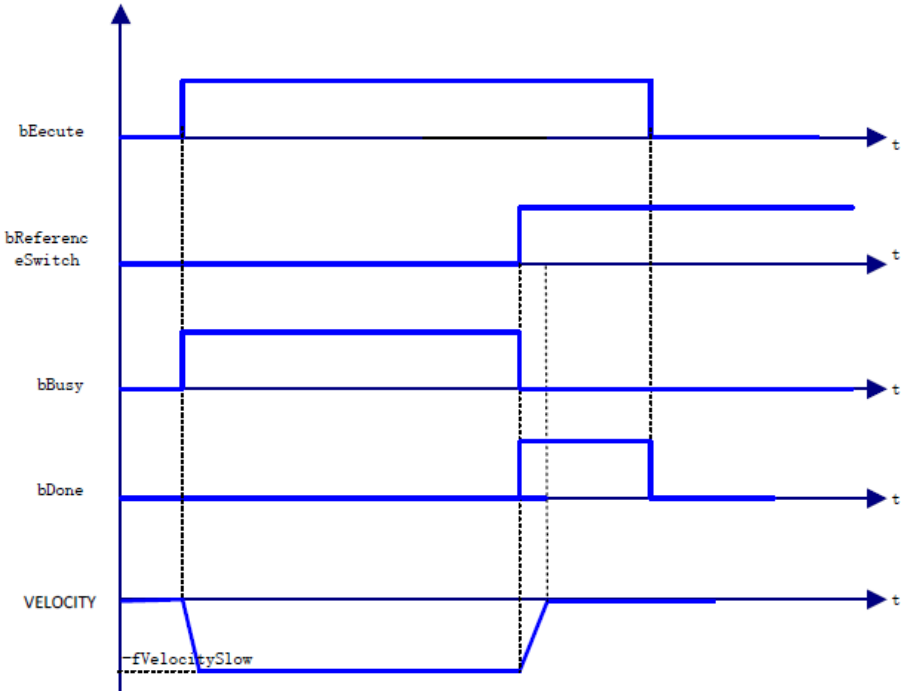
Note: After the Done signal is completed, the axis position is set to fHomePosition. The setting time is related to nHomingMode. (For details, see SMC_HOMING_MODE.)

(4) Timing diagram

a. When bReferenceSwitch=TRUE:



b. When bReferenceSwitch=FALSE:

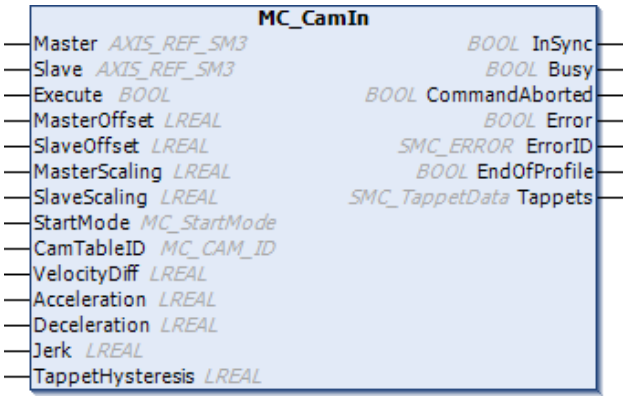


2.1.34 MC_Camin

MC_Camin: used to designate a cam table to start the execution of the e-cam actions, and specify the offset value, scaling ratio and working mode of the master and slave axes according to application requirements.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

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

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Master	Master axis	AXIS_REF	-	-	Reference to axis, that is, an instance of AXIS_REF_SM3.
Slave	Slave axis	AXIS_REF			Reference to axis, that is, an instance of AXIS_REF_SM3.

Note: The master axis and slave axis must be different axes. Otherwise, errors may be reported.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Execute	Cam function entry	BOOL	TRUE, FALSE	FALSE	The rising edge starts the execution of the function block.
MasterOffset	Master axis offset	LREAL	Negative number, positive number, or 0	0	The phase of the master axis is moved by the specified offset value.
SlaveOffset	Slave axis offset	LREAL	Negative number, positive	0	The phase of the slave axis is moved by the specified offset value.

Input variable	Name	Data type	Valid range	Initial value	Description
			number, or 0		
MasterScaling	Pre-compiling scaling factor of the master axis	LREAL	>0.0	1	The phase of the master axis is scaled up or down by the specified value.
SlaveScaling	Pre-compiling scaling factor of the slave axis	LREAL	>0.0	1	The phase of the slave axis is scaled up or down by the specified value.
StartMode	Output mode of the slave axis in relative to cam	MC_ StartMode		absolute	0: Absolute position 1: Relative position 2: ramp_in (ramp switching in) 3: ramp_in_pos (forward ramp switching in) 4: ramp_in_neg (reverse ramp switching in)
CamTableID	Table ID	MC_ CAM_ID			Used to define a cam table, in conjunction with output points of MC_CamTableSelect.
VelocityDiff		LREAL			Max. speed, different from ramp_in
Acceleration		LREAL			Acceleration for ramp_in
Deceleration		LREAL			Deceleration for ramp_in
Jerk		LREAL			Jerk for ramp_in
TappetHysteresis		LREAL			Damping factor of the tappet

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
InSync	Cam taking effect	BOOL	TRUE, FALSE	FALSE	After the master and slave axes establish a cam relationship, InSync is set. When the execution condition is OFF, InSync is reset.
Busy	Synchronous running	BOOL	TRUE, FALSE	FALSE	When Execute inputs the rising edge, it is set to TRUE, which indicates that the cam relationship is being coupled, Cam_out is used for reset. The command execution condition reset cannot reset the status.

Output variable	Name	Data type	Valid range	Initial value	Description
CommandAborted	Command interruption	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the slave axis is interrupted by other control commands.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	Error is set when an error is detected. Error is reset when the command execution condition is OFF.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.
EndOfProfile	Profile completion	BOOL		FALSE	If Periodic is 0 when MC_CamTableSelect is executed, EndOfProfile is set after the cam profile is completed for one time, and EndOfProfile is reset when the command execution condition is OFF.
Tappets		SMC_ TappetData			Associated cam tappet, which can be read by MC_GetTappetValue.

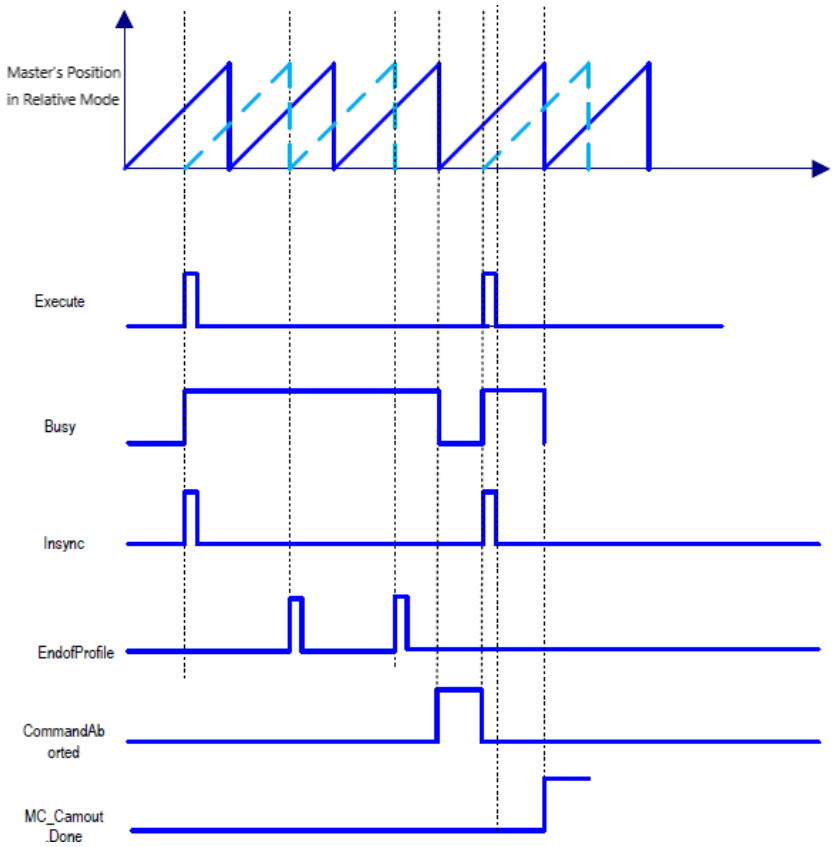
(3) Function description

Under the condition that correct cam tables are selected and axes do not encounter errors, the Execute rising edge triggers the function block. In a cam motion system, to invoke a cam profile, invoke the MC_CamTableSelect command to select the corresponding cam table, and then execute MC_CamIn; to change the cam profile, invoke the MC_CamTableSelect command to reselect a cam table. You need to use the Camout command to unbind the cam coupling relationship between the master axis and slave axis. When the command is being executed, if another command is applied to the slave axis at this time, the cam coupling relationship between the master axis and slave axis is unbound, and Command-Aborted outputs TRUE.

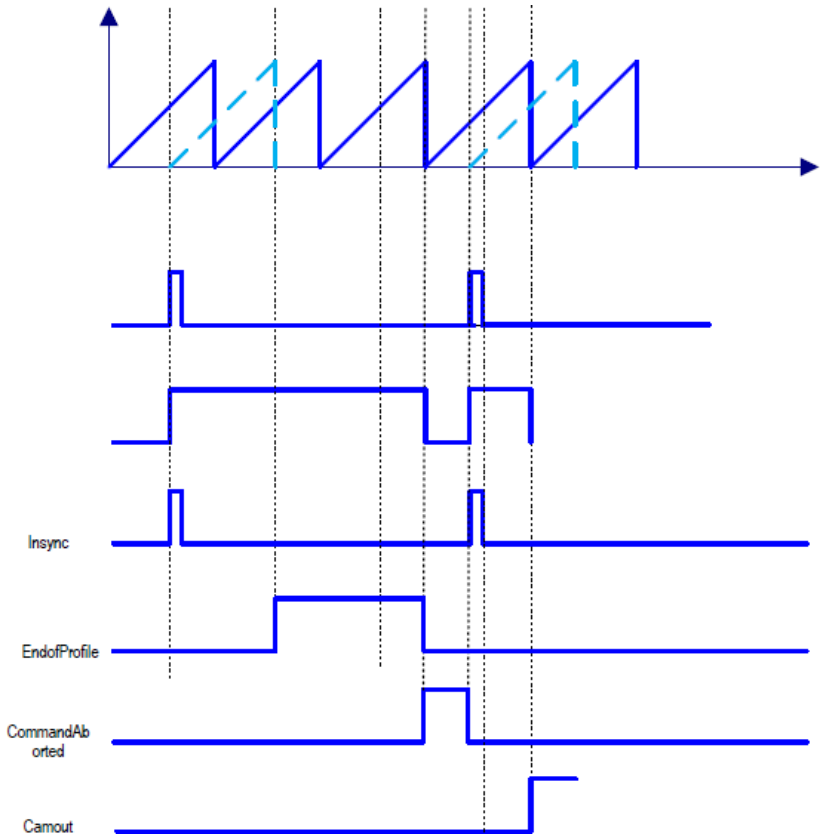
(4) Timing diagram

✧ Cyclic mode (MC_CamTableSelect.Periodic is TRUE):

Note: The MC_Camout command only unbinds the cam coupling relationship between the master axis and slave axis. If the slave axis speed is not 0 during the unbinding, the slave axis does not automatically decelerate to 0, which indicates using MC_STOP is required.



✧ Non-cyclic mode (MC_CamTableSelect.Periodic is FALSE):

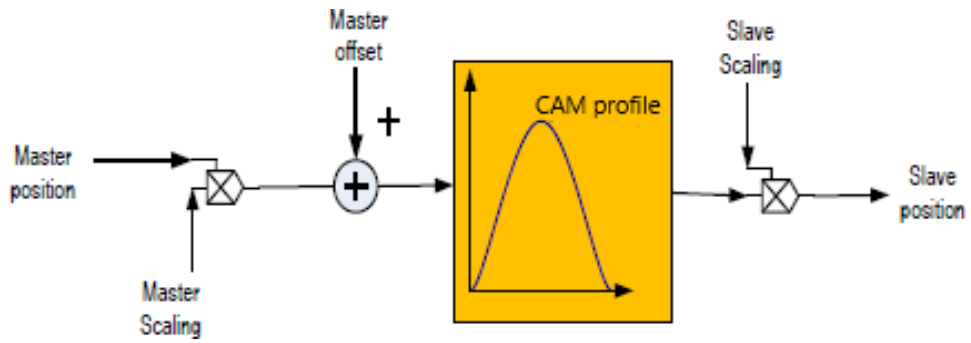


(5) Function block description

✧ Command start condition

The command can be started in any state during master axis stop, position control, speed control, and synchrization control.

The calculation method of the engaging points in the cam profile is as follows:



The following formula is obtained according to the figure:

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

The positions of the master and slave axes in the formula do not represent the actual physical axis positions, but the positions of the master and slave axes related to the cam function curve.

The relationship between the master/slave axis positions and the master/slave real axis position is described in detail.

Note: The positions of the master and slave axes refer to the positions of the master and slave axes required by the cam function curve, but not the physical real axis positions of the master and slave axes.

✧ Relationship between the cyclic mode and EndOfProfile:

Whether the cyclic or non-cyclic mode determines whether the e-cam needs to be performed again after the master axis reaches the end position.

In non-cyclic mode: Periodic is False in the MC_CamTableSelect command.

In non-cyclic mode, when the cam is completed, EndofProfile outputs TRUE; when Execute inputs FALSE, EndofProfile outputs FALSE. At this time, the cam only runs one master axis cycle.

Note: The master axis cycle indicates the range from the start position to the end position of the master axis of the e-cam.

✧ Relationship between the cyclic mode and EndOfProfile:

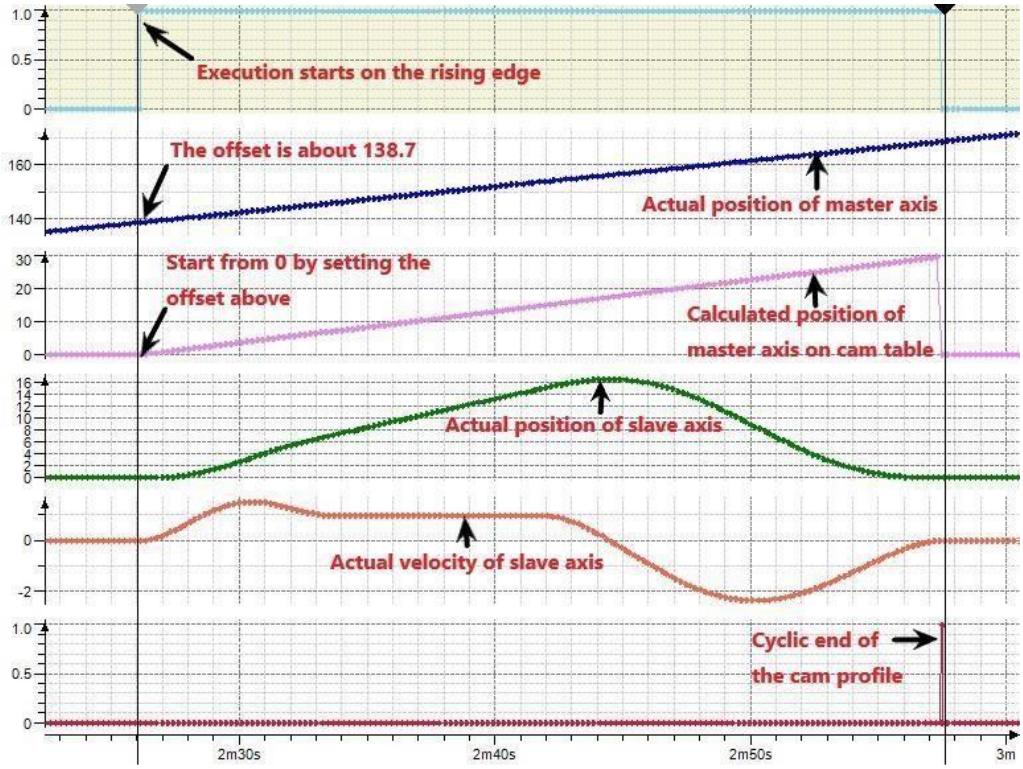
Whether the cyclic or non-cyclic mode determines whether the e-cam needs to be performed again after the master axis reaches the end position.

In non-cyclic mode: Periodic is False in the MC_CamTableSelect command.

In non-cyclic mode, when the cam is completed, EndofProfile outputs TRUE; when Execute inputs FALSE, EndofProfile outputs FALSE. At this time, the cam only runs one master axis cycle.

Note: The master axis cycle indicates the range from the start position to the end position of the master axis of the e-cam.

In cyclic mode: Periodic is TRUE in the MC_CamTableSelect command.

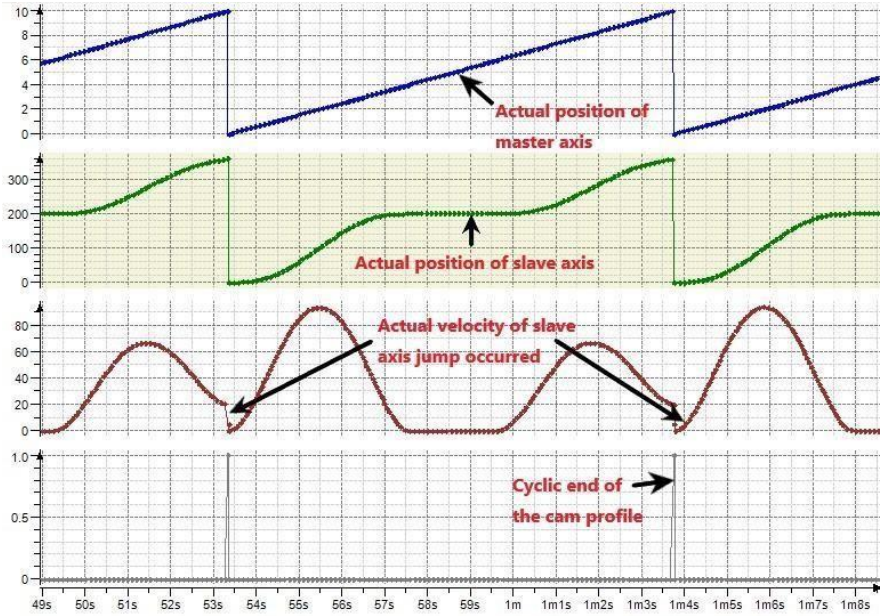


At this time, after completing one master axis cycle, the cam starts the next cycle, and the TRUE output of the EndofProfile signal only maintains one cycle.

Note: When the cam master-axis position is greater than or equal to the cam end position, the EndofProfile signal outputs TRUE, and the cam master-axis position is updated to (Cam start position + Actual position - End position). For example: The start position and end position of the cam master axis are 0 and 360, the master-slave axis scaling is set to 1, the master-slave axis offset value is set to 0, the task cycle is 2ms, and the master axis speed is 100. When the cam master-axis position in a certain task cycle is 359.99, the output of EndofProfile in the next cycle is TRUE and the master axis position becomes 359.99+100*0.002-360=0.19.

The start position and end position of the cam profile designed in cyclic mode need to maintain a smooth transition; otherwise, jumping may be caused.

For example, if the start speed is 0 and the end speed is not 0, jumping is caused when the master axis transits from the end of the cycle and the beginning of the new cycle.



The master/slave axis absolute/relative mode relationship in StartMode and MC_CamTableSlect is as follows:

Absolute mode: At the beginning of a new e-cam cycle, the calculation of the e-cam has no relationship with the present slave axis position. If the start position of the slave axis relative to the master axis is different from the end position of the slave axis relative to the master axis, jumping is caused.

Relative mode: The new e-cam cycle changes according to the present position of the slave axis; that is, the position of the slave axis at the end of the previous e-cam cycle is considered as "slave axis offset" in the present e-cam movement, therefore added. However, if the position of the slave axis corresponding to the start position of the master axis is not 0 in the e-cam definition, jumping is caused.

Ramp input: Potential jumping at the beginning of the e-cam is prevented by adding a compensation movement. (The movement is based on VelocityDiff, acceleration, and deceleration. Therefore, as long as the slave axis is rotating, the forward ramp input can only use forward compensation, and the reverse ramp input can only use reverse compensation. For the slave axis in linear motion, the compensation direction can be realized automatically, that is, the forward ramp input and the reverse ramp input can be interpreted by the ramp input.)

The relationship table is as follows:

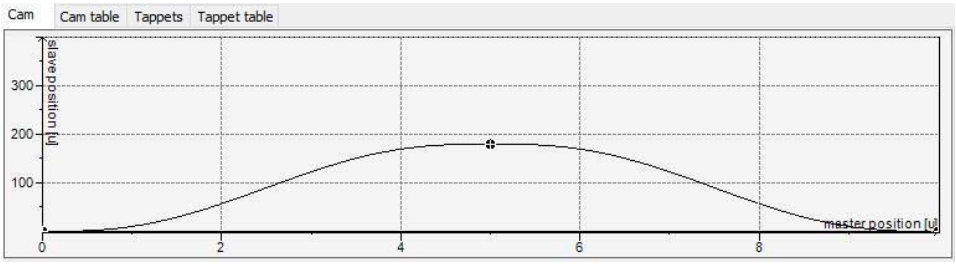
MC_CamTableSelect.MasterAbsolute	Master axis mode
absolute	Absolute mode
relative	Relative mode

MC_CamIn.StartMode	MC_CamTableSelect.SlaveAbsolute	Slave axis mode
absolute	TRUE	Absolute mode
absolute	FALSE	Relative mode
relative	TRUE	Relative mode
relative	FALSE	Relative mode
ramp_in	TRUE	Absolute mode of ramp switching in
ramp_in	FALSE	Relative mode of ramp switching in
ramp_in_pos	TRUE	Absolute mode of forward ramp switching in

MC_CamIn.StartMode	MC_CamTableSelect.SlaveAbsolute	Slave axis mode
ramp_in_pos	FALSE	Relative mode of forward ramp switching in
ramp_in_neg	TRUE	Absolute mode of reverse ramp switching in
ramp_in_neg	FALSE	Relative mode of reverse ramp switching in

The relationship is described as follows:

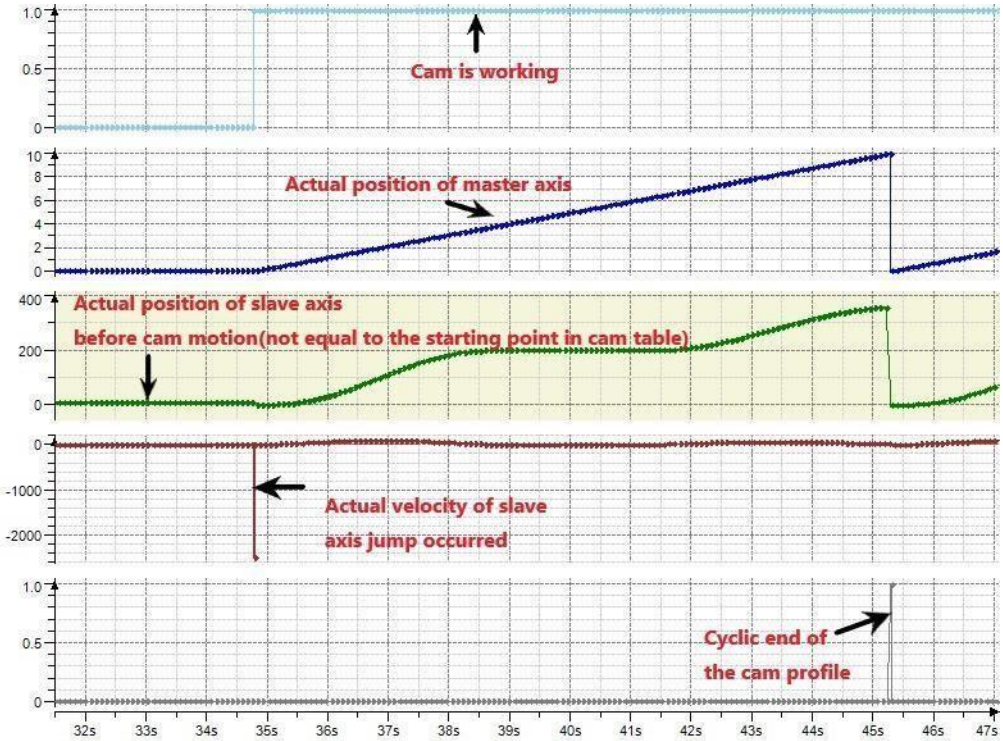
Cam master-axis range: 0-360; cam slave-axis range: 0-180; cyclic mode; master/slave axis offset value: 0; master/slave axis scaling ratio: 1 The designed cam table is shown in the following figure.



StartMode=0 (Absolute mode)

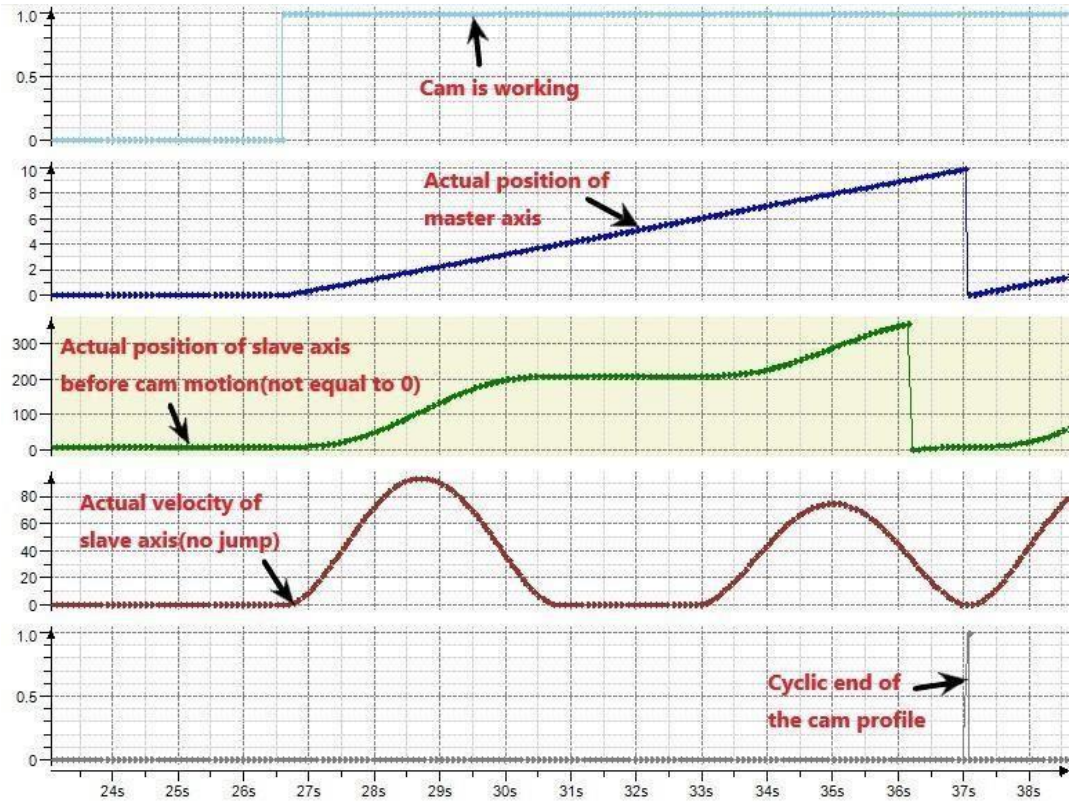
In MC_CamTableSlect, when MasterAbsolute is set to FALSE and SlaveAbsolute is set to TRUE, the master axis is working in relative mode and the slave axis is working in absolute mode. When the Execute rising edge starts the cam, the master axis of the cam starts from the "start position" (0) in the cam table, the cam slave axis is calculated and output according to the above-mentioned "cam table engaging formula". The real axis command position of the slave axis equals the output value of the engaging calculation. For example, if the start position of the cam slave axis is 0, and the real axis position of the slave axis is 20 when the cam is started, the real axis position command of the slave axis is 0 at the start, which causes jumping.

Note: In this case, jumping occurs when the slave axis (real axis) start position is not the slave axis start position of the cam.



When MC_CamTableSlect instruction MasterAbsolute is set to FALSE and SlaveAbsolute is set to FALSE, the master axis is working in relative mode and the slave axis is working in relative mode. When the Execute rising edge starts the cam, the master axis of the cam starts from the "start position" (0) in the cam table, the cam slave axis is calculated and output according to the above-mentioned "cam table engaging formula". The real axis command position of the slave axis is equal to [Output value of engaging calculation, or cam slave-axis position) plus (Real axis position of the slave axis at startup).

For example, when the cam is started, if the real axis position of the slave axis is 20, and the slave axis start position in the cam table is 0, then the real axis command position of the slave axis is 20 when the cam is started, the position in the following is 20 plus the calculated value of the cam table, and the highest value is 20 plus the max. calculated value (180) of the cam table, that is, 200.



2.1.35 MC_Camout

MC_Camout: used to unbind the cam coupling relationship of the slave axis. Note: After executing this command, the slave axis continues to run at the speed used before the unbinding. Therefore, this command needs to be used in conjunction with commands such as MC_Stop.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
MC_Camout	Cam coupling unbinding	MC_CamOut Slave AXIS_REF_SM3 — BOOL Done Execute BOOL — BOOL Busy BOOL Error SMC_ERROR ErrorID	MC_CamOut(Slave:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Slave	Slave axis	AXIS_REF			Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Execute	Cam function exit	BOOL	TRUE, FALSE	FALSE	The rising edge starts the execution of the function block.

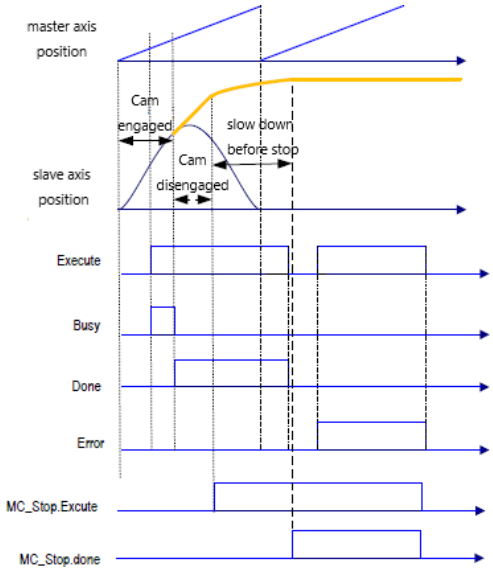
✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Completion	BOOL	TRUE, FALSE	FALSE	The cam coupling relationship with the master slave has been unbound.
Busy	Synchronous running	BOOL	TRUE, FALSE	FALSE	The command is being executed.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	Error is set when an error is detected.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.

(3) Function description

The command is used to unbind the cam coupling relationship of the slave axis. At the rising edge, the cam coupling relationship of the slave axis is unbound. After the unbinding, the salve axis may stop or may not. If the slave axis speed is not 0 before the command is executed, the cam coupling relationship is unbound after the DONE signal is completed, but the slave axis still runs at the speed before the switching out. If the slave axis does not have a cam coupling relationship, ERROR is output.

(4) Timing diagram

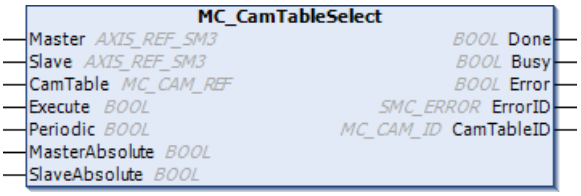


2.1.36 MC_CamTableSelect

MC_CamTableSelect: used to select cam tables, in conjunction with MC_CamIn.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

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

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Master	Master axis	AXIS_REF			Reference to axis, that is, an instance of AXIS_REF_SM3.
Slave	Slave axis	AXIS_REF			Reference to axis, that is, an instance of AXIS_REF_SM3.
CamTable	Table selection	MC_CAM_REF			Mapped to cam table description, that is, an instance of MC_CAM_REF.

Note: The master axis and slave axis must be different axes. Otherwise, errors may be reported. The cam table specified by CamTable must be correct; otherwise, errors may be reported. The master and slave axes may be virtual axes and may be virtual axes.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Execute	Enabling	BOOL	TRUE, FALSE	FALSE	The rising edge starts the execution of the function block.
Periodic	Repeated mode	BOOL	TRUE, FALSE	FALSE	Used to specify whether the cam table is executed only once or repeatedly. TRUE: Repeatedly FALSE: Not repeatedly
MasterAbsolute	Master axis absolute mode	BOOL	TRUE, FALSE	FALSE	Used to specify whether the coordinate system of master axis tracking uses an absolute or

Input variable	Name	Data type	Valid range	Initial value	Description
					relative position. 1: Absolute position 0: Relative position
SlaveAbsolute	Slave axis absolute mode	BOOL	TRUE, FALSE	FALSE	Used with StartMode in MC_CamIn to specify whether the present command position of the slave axis is the absolute or relative position output of the cam table. 1: Absolute position 0: Relative position

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Completion	BOOL	TRUE, FALSE	FALSE	The cam coupling relationship with the master slave has been unbound.
Busy	Synchronous running	BOOL	TRUE, FALSE	FALSE	The command is being executed.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	Error is set when an error is detected.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.
CamTableID	Effective cam ID	MC_CAM_ID			Used to select the effective cam ID, which is used together with CamTableID in MC_CamIn.

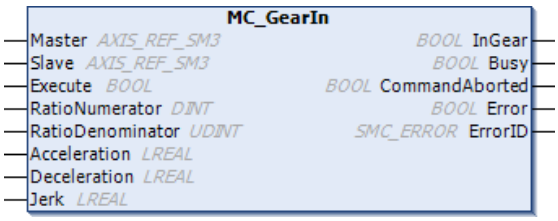
(3) Function description

The command specifies the cam table required for e-cam running. Therefore, before using this command, you must edit the cam table (with a cam editor or online). The specified cam table can be executed at the rising edge or refreshed after cam table update. When the Done signal is TRUE, the variable CamTableID is output and takes effect. During command execution, Busy is TRUE; when Done is TRUE, Busy is FALSE. For details about MasterAbsolute, SlaveAbsolute, and Periodic, see MC_CamIn.

2.1.37 MC_GearIn

MC_GearIn: used to set the gear ratio between the slave axis and the master axis to perform electronic gearing.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
MC_GearIn	E-gear function block		MC_GearIn(Master:= , Slave:= , Execute:= , RatioNumerator:= , RatioDenominator:= , Acceleration:= , Deceleration:= , Jerk:= , InGear=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Master	Master axis	AXIS_REF			Reference to axis, that is, an instance of AXIS_REF_SM3.
Slave	Slave axis	AXIS_REF			Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Execute	Enabling	BOOL	TRUE, FALSE	FALSE	The function block is started at the rising edge.
RatioNumerator	Numerator of the gear ratio	DINT	Positive number, negative number	1	Numerator of the gear ratio
RatioDenominator	Denominator of the gear ratio	UDINT	Positive number	1	Denominator of the gear ratio
Acceleration	Acceleration	LREAL	Positive number or 0		Acceleration
Deceleration	Deceleration	LREAL	Positive number or 0		Deceleration
Jerk	Jump	LREAL	Positive number or 0		Jerk value

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
InGear	Gear ratio reached	BOOL	TRUE, FALSE	FALSE	TRUE: The slave axis reaches the target speed.
Busy	Synchronous running	BOOL	TRUE, FALSE	FALSE	The command is being executed.
CommandAborted	Interruption	BOOL	TRUE, FALSE	FALSE	TRUE: The function block is interrupted by another control command.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	Error is set when an error is detected.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.

(3) Function description

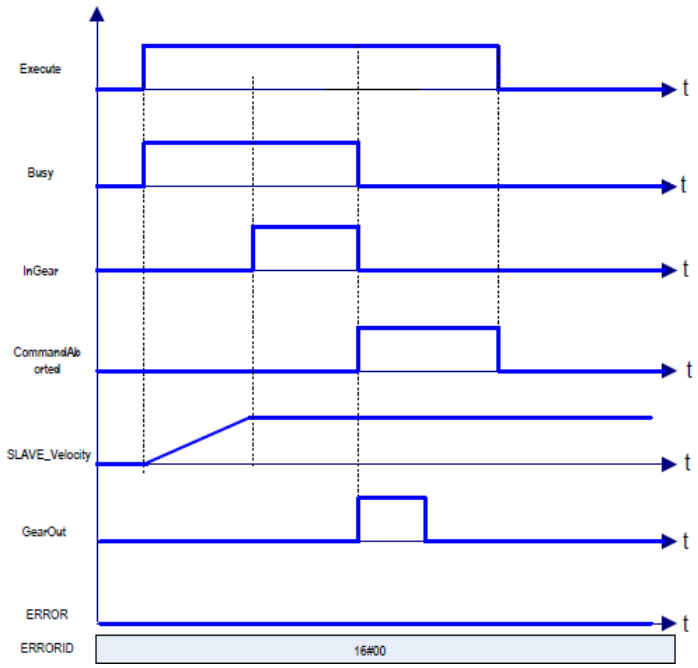
The e-gear action is started at the rising edge. To unbind the coupling after executing the e-gear, the GearOut command must be used. This command is a speed e-gear function, and the synchronization distance loss caused during acceleration will not be automatically compensated. When the Busy signal is TRUE during command execution, if the slave axis target speed is not reached, the new rising edge of Execute will not affect it. When the Busy signal is TRUE during command execution, if the slave axis target speed is reached, the new rising edge of Execute will not affect it. When the target speed is reached, InGear is TRUE, and then:

Slave axis movement amount = Master axis movement amount * RatioNumerator/RatioDenominator

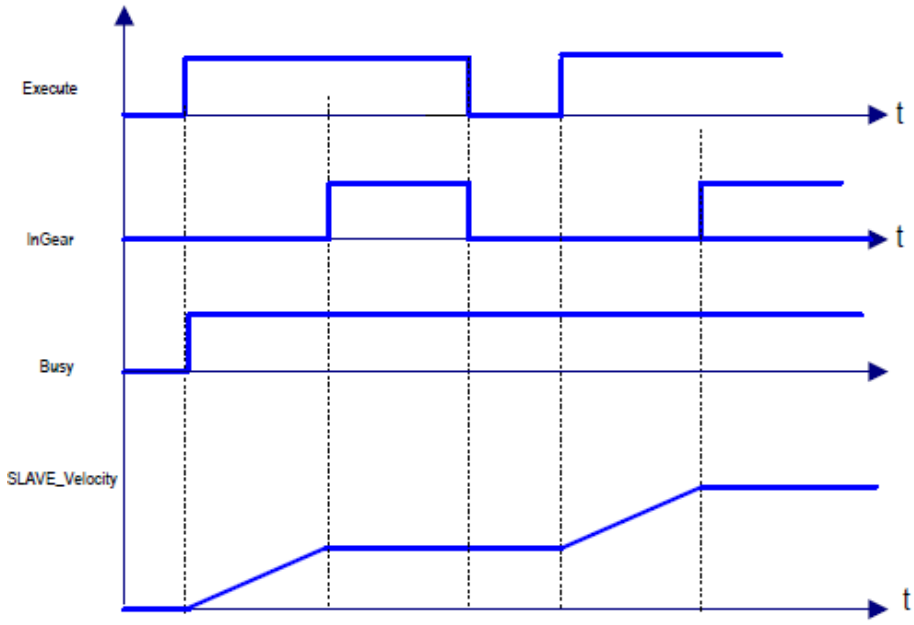
If the master axis speed changes in real time, exercise caution before using this command.

Note: Do not use the MC_SetPosition command during command execution to avoid accidents caused by the rapid motor running.

(4) Timing diagram



The timing diagram of the restart after gear ratio parameter change is as follows:



2.1.38 MC_GearOut

MC_GearOut: used to terminate the MC_GearIn and MC_GearInPos commands that are being executed.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
MC_GearOut	E-gear coupling disconnection		MC_GearOut(Slave:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Slave	Slave axis	AXIS_REF			Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Execute	Enabling	BOOL	TRUE, FALSE	FALSE	The function block is started at the rising edge.

✧ Output variable

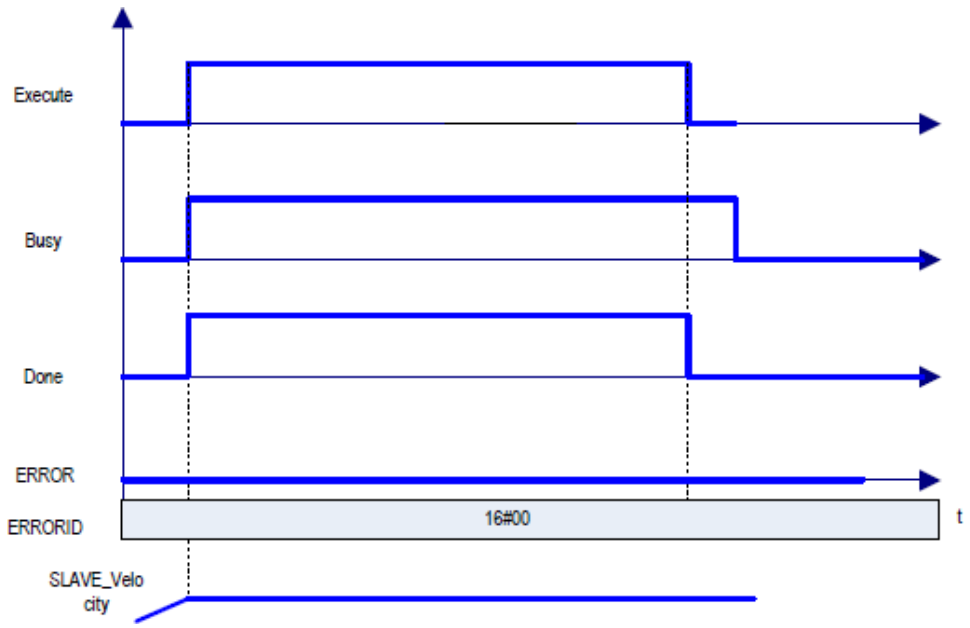
Output variable	Name	Data type	Valid range	Initial value	Description
Done	Completion	BOOL	TRUE, FALSE	FALSE	TRUE: The e-gear coupling of the slave axis is disconnected.
Busy	Synchronous running	BOOL	TRUE, FALSE	FALSE	The command is being executed.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	Error is set when an error is detected.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.

(3) Function description

The e-gear action switching out is started at the rising edge. If Execute is TRUE and ERROR is False, Busy is TRUE and Done is TRUE.

After the e-gear action is switched out, the slave axis speed used before the switching out is used. Therefore, the slave axis is stopped in conjunction with the MC_Stop command. At the falling edge, Done is FALSE.

(4) Timing diagram

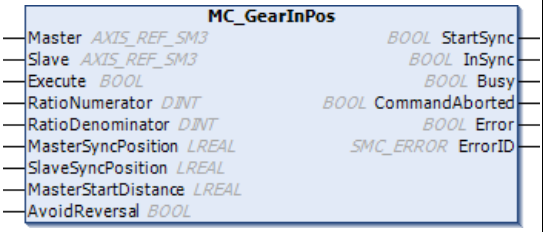


2.1.39 MC_GearInPos

MC_GearInPos: used to set the e-gear ratio between the slave axis and the master axis to perform electronic gearing. It specifies the master axis position, slave axis position, and master axis distance from the synchronization start to switch in e-gear actions.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

MC_GearInPos	E-gear coupling switching -in position		`MC_GearInPos(`
			`Master:= ,`
			`Slave:= ,`
			`Execute:= ,`
			`RatioNumerator:= ,`
			`RatioDenominator:= ,`
			`MasterSyncPosition:=`
			`,`
			`SlaveSyncPosition:= ,`
			`MasterStartDistance:=`
			`,`
			`AvoidReversal:= ,`
			`StartSync=> ,`
			`InSync=> ,`
			`Busy=> ,`
			`CommandAborted=> ,`
			`Error=> ,`
			`ErrorID=> );`

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Master	Master axis	AXIS_REF			Reference to axis, that is, an instance of AXIS_REF_SM3.
Slave	Slave axis	AXIS_REF			Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Execute	Enabling	BOOL	TRUE, FALSE	FALSE	The function block is started at the rising edge.
RatioNumerator	Numerator of the gear ratio	DINT	TRUE, FALSE		Numerator of the master/slave speed ratio
RatioDenominator	Denominator of the gear ratio	DINT			Denominator of the master/slave speed ratio
MasterSyncPosition	Master axis synchronization position	LREAL			Master axis position when the master/slave axis gear ratios are coupled.
SlaveSyncPosition	Slave axis synchronization position	LREAL			Slave axis position when the master/slave axis gear ratios are coupled.

Input variable	Name	Data type	Valid range	Initial value	Description
MasterStartDistance	Master axis position of synchronization execution	LREAL			According to this position value, -MasterSyncPosition, and the SlaveSyncPosition value, a smooth curve is calculated to make the slave axis gear synchronized with the master axis gear when the slave axis is at SlaveSyncPosition. The master axis range of the curve is [MasterStartDistance, MasterSyncPosition].
AvoidReversal	Disabling reverse running	BOOL	TRUE, FALSE	FALSE	Set to FALSE if the physical position of the slave axis leads. Set to TRUE if the slave axis cannot implement reverse running physically or the reverse running may cause danger. It is applicable only to modal axes. If reverse running cannot be avoided, the axis will stop due to exceptions.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
StartSync	Coupling start	BOOL	TRUE, FALSE	FALSE	TRUE: The e-gear coupling is started.
InSync	Coupling	BOOL	TRUE, FALSE	FALSE	TRUE: The e-gear coupling is completed, and the master/slave axis gear ratios are being coupled.
Busy	Synchronous running	BOOL	TRUE, FALSE	FALSE	The command is being executed.
CommandAborted	Command interruption	BOOL	TRUE, FALSE	FALSE	TRUE: The function block is interrupted by another control command.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	Error is set when an error is detected.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.

(3) Function description

The command is started at the rising edge.

After the action starts, the slave axis accelerates or decelerates at the target speed that is the master axis speed multiplied by the gear ratio.

The essential of the process from the synchronization start to the end is an e-cam where the slave axis follows the master axis in the synchronization interval. At this time, the command automatically designs a cam profile according to the master axis range (MasterSyncPosition-MasterStartDistance, MasterSyncPosition), the slave axis range (current position, SlaveSyncPosition), and the gear ratios. When synchronization is performed, the slave axis follows the master axis to complete the cam action.

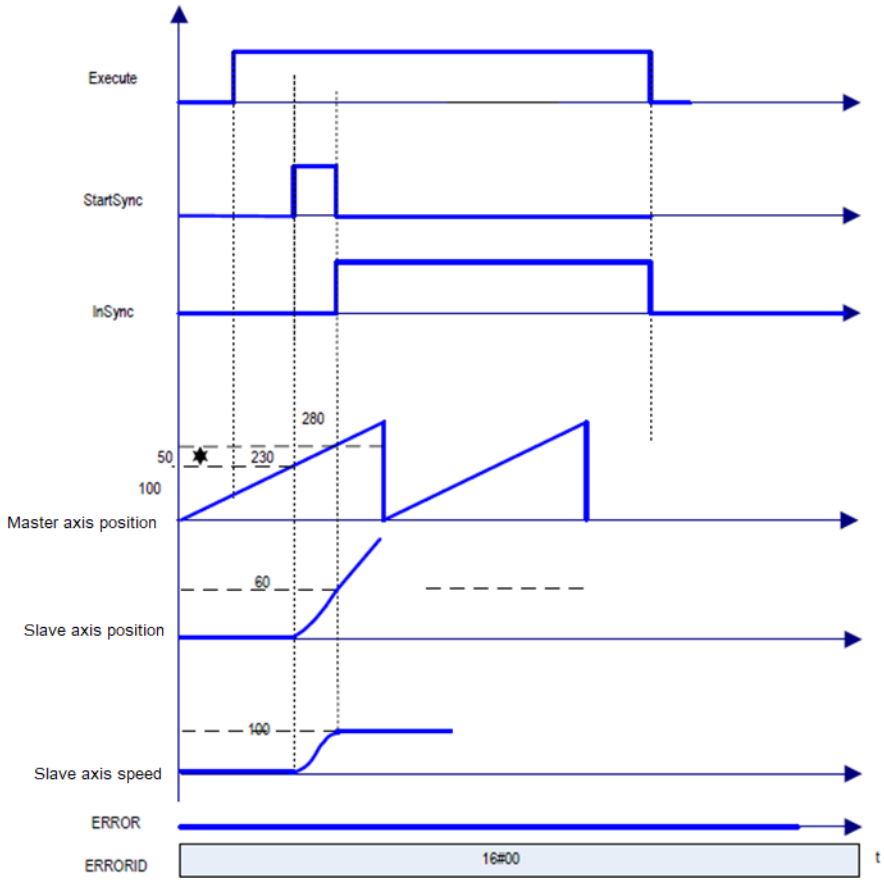
Note: If the master and slave axes work in linear mode, ensure that the above-mentioned parameters are set properly; otherwise, the gear action cannot be performed correctly. Therefore, it is recommended that the master and slave axes work in cyclic mode when this command is used.

For example: Both the master and slave axes move forward in linear mode. If the master axis position > MasterSyncPosition-MasterStartDistance, or the slave axis position > SlaveSyncPosition, when the command is executed, the e-gear movement cannot be switched in.

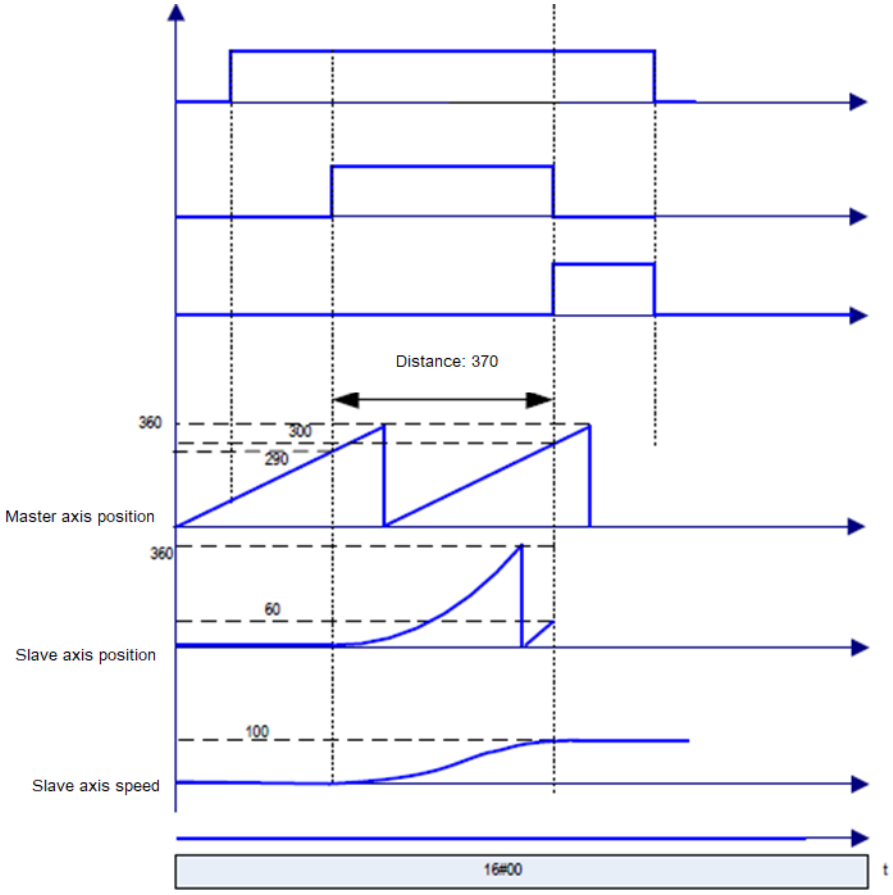
The timing diagram instances with different parameters are provided:

When the master axis works in cyclic mode (360 cycles) and the slave axis works in cyclic mode (360 cycles):

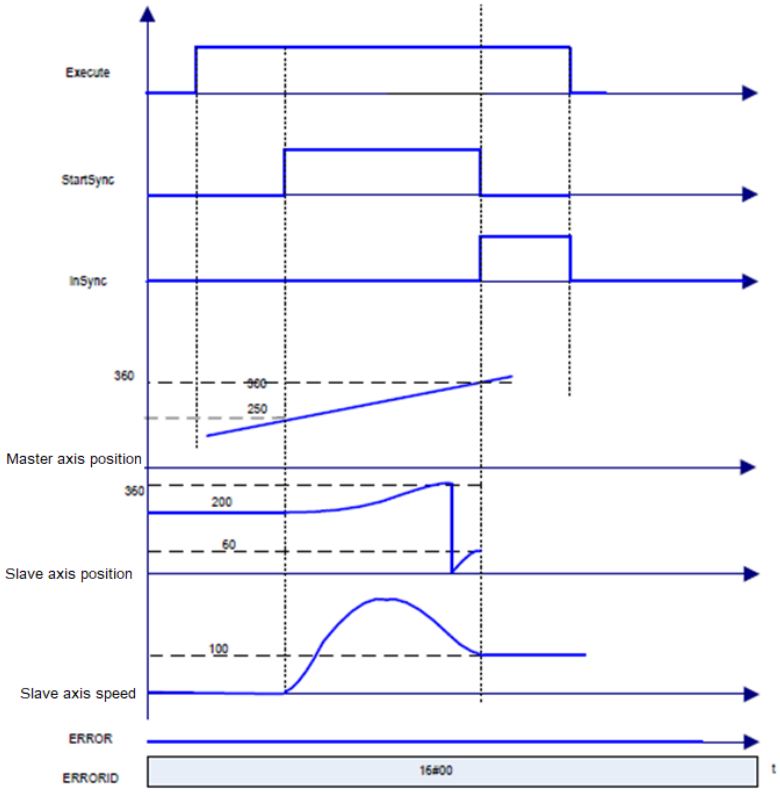
a) MasterSyncPosition=280, MasterStartDistance=50, SlaveSyncPosition=60, Master axis speed=50, AvoidReversal=FALSE



b) MasterSyncPosition=300, MasterStartDistance=370, SlaveSyncPosition=60, Master axis speed=50, AvoidReversal=FALSE



c) MasterSyncPosition=300, MasterStartDistance=50, SlaveSyncPosition=60, Master axis speed=50, AvoidReversal=FALSE, Slave axis start position > 60

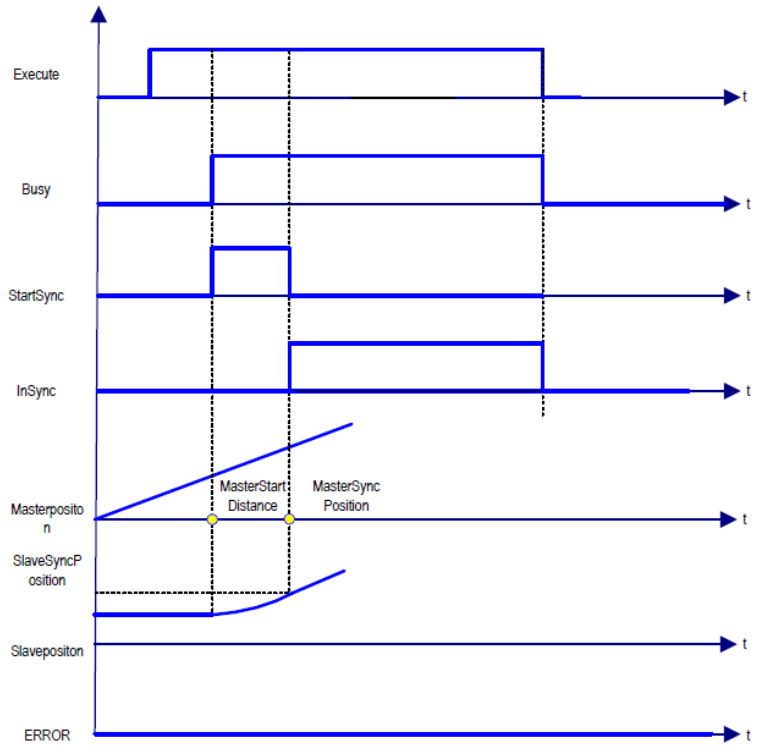


When the synchronization is completed, InGear is TRUE, the target speed is reached also, and then:

Slave axis movement amount = Master axis movement amount * RatioNumerator/ RatioDenominator

For AvoidReversal: If the slave axis is a modal axis and the master axis speed (a gear ratio multiple) is not relative to the slave axis speed, MC_GearInPos will try to avoid the reversal of the slave axis. It attempts to "stretch" the movement of the slave axis by adding 5 slave axis cycles. If the "stretch" is invalid, an error occurs and the slave axis stops abnormally. If the slave axis speed is related to the master axis speed (a gear ratio multiple), an error occurs and the slave axis stops abnormally. If the slave axis is a modal axis in linear mode, an error occurs when Execute inputs the rising edge.

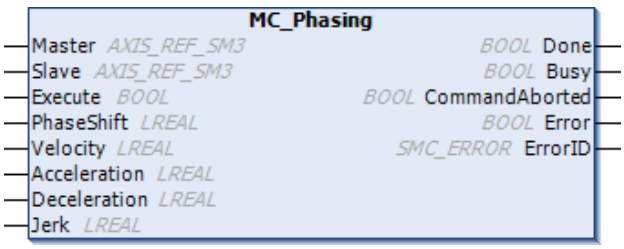
(4) Timing diagram



2.1.40 MC_Phasing

MC_Phasing: used to specify the phase difference between the master axis and slave axis.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
MC_Phasing	Electronic gear (e-gear) coupling disconnection		MC_Phasing(Master:= , Slave:= , Execute:= , PhaseShift:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Master	Master axis	AXIS_REF			Reference to axis, that is, an instance of AXIS_REF_SM3.
Slave	Slave axis	AXIS_REF			Reference to axis, that is, an instance of AXIS_REF_SM3.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Execute	Enabling	BOOL	TRUE, FALSE	FALSE	The function block is started at the rising edge.
PhaseShift	Phase difference between the master axis and slave axis	LREAL		0	Phase difference between the master axis and slave axis. A positive number indicates the slave axis lags.
Velocity	Speed	LREAL		0	Max. speed at phase shift execution
Acceleration	Acceleration	LREAL		0	Max. acceleration at phase shift execution
Deceleration	Deceleration	LREAL		0	Max. deceleration at phase shift execution
Jerk	Jerk	LREAL		0	Max. jerk at phase shift execution

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Completion	BOOL	TRUE, FALSE	FALSE	TRUE: The e-gear coupling of the slave axis is disconnected.
Busy	Synchronous running	BOOL	TRUE, FALSE	FALSE	The command is being executed.
CommandAborted	Command interruption	BOOL	TRUE, FALSE	FALSE	TRUE: The function block is interrupted by another control command.
Error	Error flag	BOOL	TRUE, FALSE	FALSE	Error is set when an error is detected.
ErrorID	Error ID	SMC_ERROR		0	When an error occurs, the error ID is output.

(3) Function description

The phase shift is executed at the rising edge of Execute. The slave axis automatically calculates a smooth curve, completing the phase shift relative to the master axis. The master/slave axis phase difference is the value of PhaseShift in the input signal. When the value is a positive number, the slave axis lags behind the master axis.

After the phase shift is completed, Done is TRUE.

The master/slave axis phase difference is compensated according to PhaseShift, Velocity, Acceleration, and Deceleration.

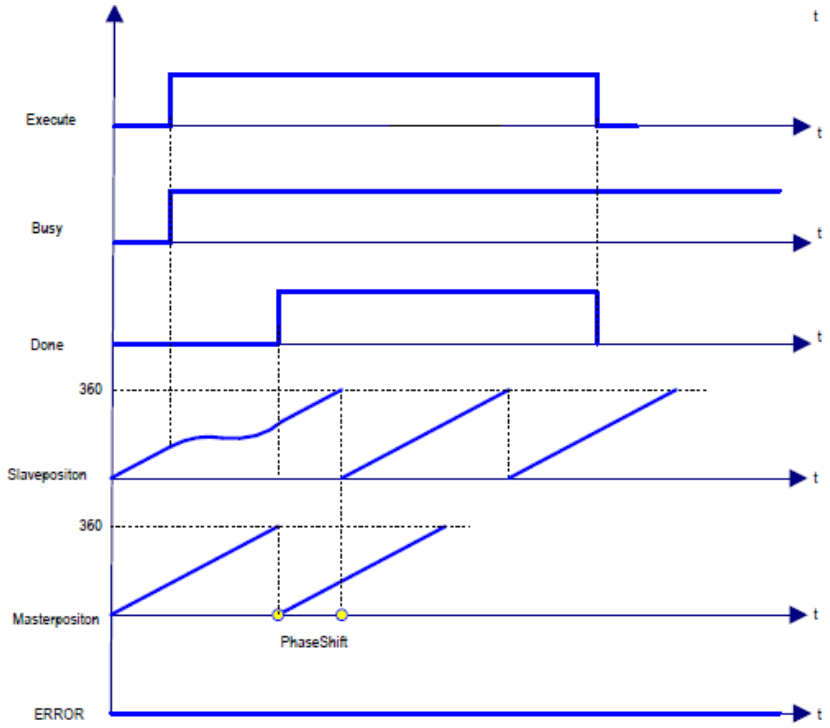
When the master/slave axis phase difference reaches PhaseShift, the Done signal is output.

During the command execution, if the master axis command position and feedback position remain unchanged, the slave axis is adjusted. Then the master/slave axis phase difference is PhaseShift.

The final result of this command is the phase shift between the given axis values, and therefore the actual feedback value of a real axis may be inconsistent with the final shift.

(4) Timing diagram

The master and slave axes move in 360 cycles, and the adjustment is performed at the rising edge of the Execute signal. After the adjustment is completed, the phase shift between the slave axis and the master axis is the value of PhaseShift.



3 INVT commands

3.1 Communication commands

3.1.1 ModbusRTU master node

The following describes the variables in the ModbusRTU master node command library.

✧ Variable definitions

Category	Variable name	Type		Function	Remarks
ModbusRTU_ Master_Init_COM1	Execute1	INPUT	BOOL	Indicates whether to activate the initialization of serial port.	0: No 1: Yes
	Baud1		DINT	Baud rate.	Example: 115200
	Databits1		INT	Data bits.	Example: 8 bits (7 bits disallowed in ASCII)
	Stopbits1		INT	Stop bits.	Example: 1 stop bit; 2 stop bits
	Parity1		INT	Read/write flag.	0: No parity 1: Even parity 2: Odd parity
	Slave1		UINT	Slave node ID.	1-128
	Timeout1		DINT	Timeout time.	Example: 1000
	bDone1	OUTPUT	BOOL	Completion flag.	0: The command is being executed. 1: The command has been completed.
	Error1		BOOL	Error flag.	0: No error 1: An error occurred.
	ErrorID1		INT	Error ID.	See the ModbusRTU error ID table.
ModbusRTU_ Master_Fun_COM1	xExecute1	INPUT	BOOL	Indicates whether to activate the read/write function.	0: No 1: Yes
	Fun_Code1		INT	Function code.	0X01, 0x03, 0x05, 0x06, 0x0f, 0x10
	Addr1		UINT	IP address.	0x0000-0xFFFF
	DataCount1		UINT	Data count.	Read: 1-250 Write: 1-240

Category	Variable name	Type		Function	Remarks
	DataPtr1		POINTER TO INT	Data pointer.	Points to the address where the data to read or write is stored.
	Error1	OUTPUT	BOOL	Error flag.	0: No error 1: An error occurred.
	ErrorID1		INT	Error ID.	See the ModbusRTU error ID table.

When serial port 2 serves as the ModbusRTU_Master master node, the variable count is the same, with the only difference that the number at the variable name end changes from 1 to 2, for example, ModbusRTU_Master_Init_COM2.

✧ Use instructions

Step 1 Set data for connecting the ModbusRTU master node to the slave node.

Category	Variable	Function	Example
ModbusRTU_Master_Init_COM1	Execute1	Indicates whether to enable the slave node.	Enable := TRUE
	Baud1	Baud rate	Baud1 := 19200
	Databits1	Data bits.	Port :=8
	Stopbits	Stop bits.	Unit := 1
	Parity1	Parity bit	Parity1:=2
	Slave1	Slave node ID.	Slave1:= 12
	Timeout1	Timeout time.	Delay Time := 1000

When defining the ModbusRTU slave node to be connected, refer to the preceding table for configuration. The following is an example (in ST) for reference:

```
PROGRAM PLC_PRG
VAR
    Execute1_1:BOOL :=FALSE;
    Baud1_1:DINT:=19200;
    Databits1_1:INT:=8;  Stopbits1_1:INT:=1;
    Parity1_1:INT:=2;    Slave1_1:BYTE:=1;
    Timeout1_1:DINT:=1000;
    xExecute1_1:BOOL:=FALSE;
    Fun_Code1_1:INT:=3;
    Addr1_1:UINT:=4;
    DataCount1_1 : UINT:=10;  DataPtr1_1:ARRAY[0..9] OF INT;
    ModbusRTU_Master_Init_COM1 :ModbusRTU_Master_Init_COM1;  ModbusRTU_Master_Fun_COM1:ModbusRTU_Master_Fun_COM1;
```

Variable definitions

```
ModbusRTU_Master_Init_COM1(
    Execute1:=Execute1_1 ,
    Baud1:= Baud1_1,
    Databits1:= Databits1_1,
    Stopbits1:= stopbits1_1,
    Parity1:= Parity1_1,
    Slave1:= Slave1_1,
    Timeout1:= Timeout1_1,
    bDone1=> ,
    Error1=> ,
    ErrorID1=> );
ModbusRTU_Master_Fun_COM1(
    xExecute1:= xExecute1_1,
    Fun_Code1:= Fun_Code1_1,
    Addr1:= Addr1_1,
    DataCount1:= DataCount1_1,
    DataPtr1:= ADR(DataPtr1_1),
    Error1=> ,
    ErrorID1=> );
```

Parameters for the slave node to connect

Function parameters

Figure 3.1 Parameter setting example for connecting the ModbusRTU master node to the slave node

Step 2 After setting the parameters for connecting the ModbuRTU master node to the slave node, set the communication function.

The setting parameters are described in the following table, and a setting example is also provided.

Variable	Function	Example
xExecute1	Indicates whether to enable RTU communication.	RW:= TRUE
Fun_Code1	Function code	Fun_Code1:=0x03
Addr1	R/W register start address.	Addr := 2001
DataCount1	R/W register count.	Conut := 12
DataPtr1	Points to the address where the data to read or write is stored.	ADR (DATE_RTU1)

```
1   ModbusRTU_Master_Init_COM1_1(
2       Execute1:= Execute1_1,
3       Baud1:= Baud1_1,
4       Databits1:= Databits1_1,
5       Stopbits1:= Stopbits1_1,
6       Parity1:= Parity1_1,
7       Slave1:= Slave1_1,
8       Timeout1:= Timeout1_1,
9       bDone1=> ,
10      Error1=> ,
11      ErrorID1=> );
12
13      ModbusRTU_Master_Fun_COM1_1(
14      xExecute1:= xExecute1_1,
15      Fun_Code1:= Fun_Code1_1,
16      Addr1:= Addr1_1,
17      DataCount1:= DataCount1_1,
18      DataPtr1:= ADR(DataPtr1_1),
19      Error1=> ,
20      ErrorID1=> );
```

Figure 3.2 Parameter setting example for communication between the ModbusRTU master node and the slave node

3.1.2 ModbusRTU slave node

The following describes the variables in the ModbusRTU slave node command library.

✧ Variable definitions

Category	Variable name	Type		Function	Remarks
ModbusRTU_Slave1	Execute1	INPUT	BOOL	Indicates whether to activate the initialization of serial port.	0: No 1: Yes
	Baud1		DINT	Baud rate	Example: 115200
	Databits1		INT	Data bits.	Example: 8 bits, 7 bits
	Stopbits1		INT	Stop bits.	Example: 1 stop bit; 2 stop bits
	Parity1		INT	Read/write flag.	0: No parity 1: Even parity 2: Odd parity
	Slave_Addr1		UINT	Slave node ID.	1-128
	Enable1		BOOL	Indicates whether to	0: No 1: Yes

Category	Variable name	Type		Function	Remarks
				activate the read/write function.	
	Done1	OUTPUT	BOOL	Completion flag.	0: Incompleted 1: Completed
	ErrorID1		BYTE	Error ID.	See the ModbusRTU error ID table.

✧ Use instructions

Step 1 Set serial port parameters, and establish the connection between the ModbusRTU master node and slave node.

Category	Variable	Function	Example
ModbusRTU_Slave1	Execute1	Indicates whether to enable the slave node.	Enable := TRUE
	Baud1	Baud rate	Baud1 := 19200
	Databits1	Data bits.	Port :=8
	Stopbits	Stop bits.	Unit := 1
	Parity1	Parity bit	Parity1:=2
	Timeout1	Timeout time.	Delay Time := 1000
	Slave_Addr1	Slave node ID.	Slave1:= 12

Configure the slave node by referring to the ModbusRTU master node serial port parameters and the parameters in the preceding table. (The variable Slave_Addr1 in the preceding table corresponds to Slave1 of the master node.)

Step 2 Exchange data between the ModbusRTU master node and slave node.

Set Execute1 to activate the ModbusRTU slave node. If the master node function code is 0x03 reading the holding register, or the master node function code is 0x10 writing multiple registers, define a storage zone in the variable area, of which the size cannot be smaller than the size of data that the ModbusTCP master node will write. If the master node function code is 0x0F writing multiple coils, the operation is similar.

3.1.3 ModbusTCP master node

The following describes the variables in the ModbusTCP master node command library.

✧ Variable definitions

Variable name	Type		Function	Remarks
Enable	INPUT	BOOL	Indicates whether to activate the ModbusTCP function.	0: No 1: Yes
IP		STRING	Slave node IP address.	Example: 192.168.1.13
Port		DINT	Slave node port number.	Example: 502
Unit		INT	Slave node ID.	Non-negative integer

Variable name	Type		Function	Remarks
DelayTime		INT	Response delay.	Non-negative integer
Fun_Enable		BOOL	Indicates whether to enable the function code.	0: No 1: Yes
fun_code		BYTE	Function code.	0x03: Read multiple registers. 0x10: Write multiple registers.
Addr		UINT	R/W register address.	Example: 2000, 2001
Count		INT	R/W register count.	Up to 120 in a batch.
CoilSingleData		INT	Indicates whether to write a single coil.	0 or 1
BitPtr		POINTER TO BOOL	Bit data pointer.	Used to save the bit data to read and write.
DataPtr		POINTER TO INT	R/W pointer.	Used to store the location information of the data read or to store the data to be written to registers.
Done	OUTPUT	BOOL	Completion flag.	0: The command is being executed. 1: The command has been completed.
Error		BOOL	Error flag.	0: No error 1: An error occurred.
ErrorID		INT	Error ID.	See the ModbusTCP error ID table.

✧ Use instructions

Step 1 Perform settings for connecting the ModbusTCP master node to the slave node.

In the project monitoring state, set the parameters for connecting to the ModbusTCP slave node. See the following table.

Variable	Function	Example
Enable	Indicates whether to enable the slave node.	Enable := TRUE
IP address	ModbusTCP slave node IP address.	IP := ‘192.168.1.13’
Port	ModbusTCP slave node port number.	Port := ‘502’
Unit	ModbusTCP slave node ID.	Unit := 3
Delay Time	Function start timeout time.	Delay Time := 1000

When the master node accesses a single slave node, the preceding variables need to be assigned. The following provides an example (in FDB):

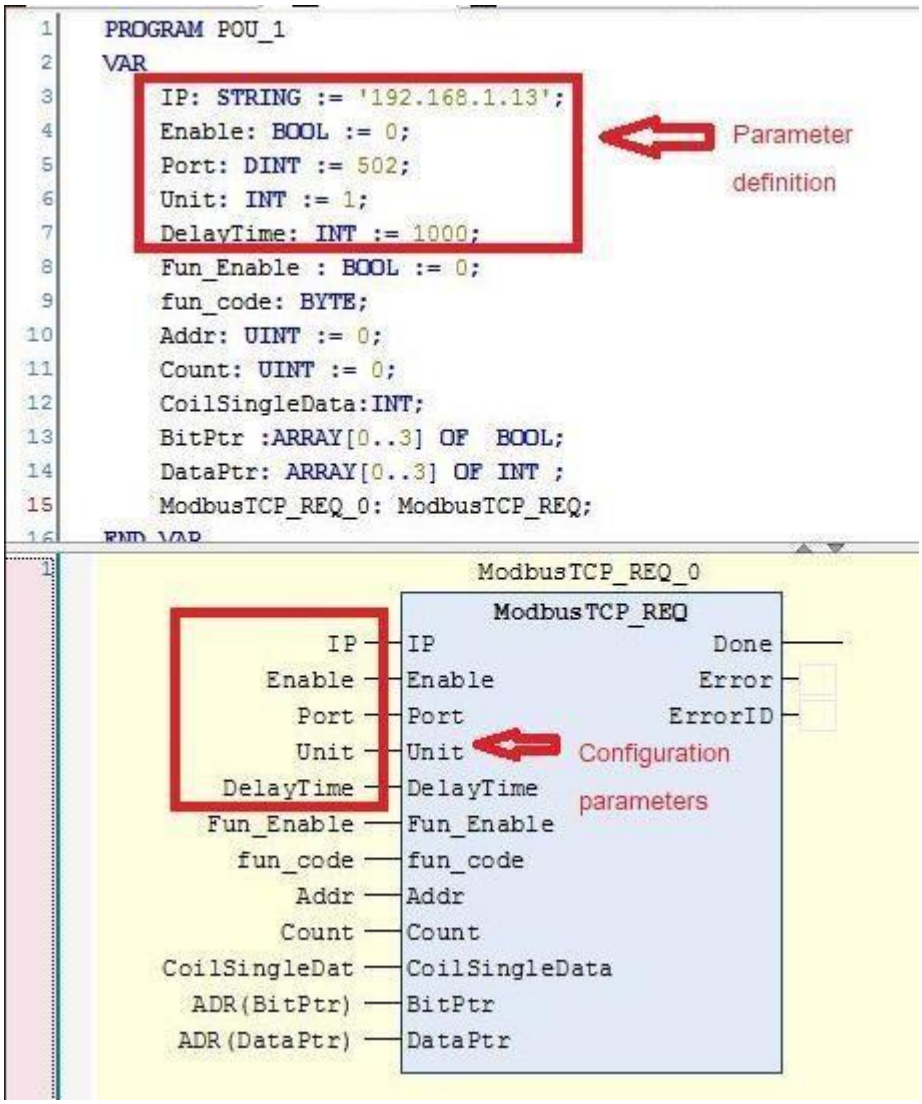


Figure 3.3 Parameter setting example for connecting the ModbusTCP master node to the slave node

The function block in the preceding figure represents an independent connection between the ModbusTCP master node and slave node. If you want to add another ModbusTCP master/slave node connection, create a new function block and set the parameters by referring to the example.

Step 2 After setting the parameters for connecting the ModbuTCP master node to the slave node, set the communication function. The setting parameters are described in the following table, and a setting example is also provided.

Variable	Function	Example
Fun_Enable	Indicates whether to enable the function code.	Fun_Enable:= TRUE
fun_code	Function of reading/writing multiple register coils.	Fun_code := 3
Addr	R/W register start address.	Addr := 2001
Count	R/W register count.	Conut := 12
DataPtr	Points to the address where the data to read or write is stored.	ADR (DATE_TCP)

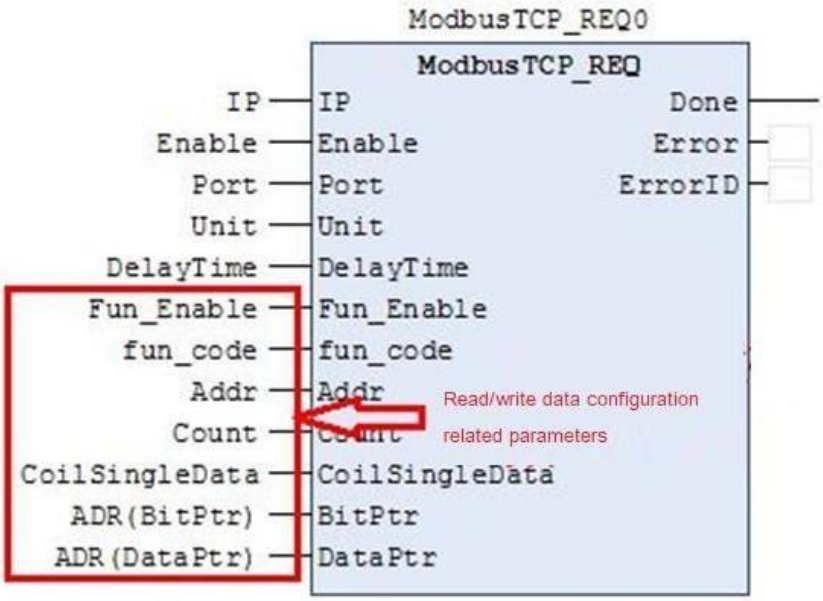


Figure 3.4 Parameter setting example for communication between the ModbusTCP master node and the slave node

In the preceding figure, each operation block represents a ModbusTCP request. As shown in the figure, the connection is defined for one ModbusTCP master node and slave node. The first and third operation blocks indicate (0x03) reading the holding registers from different slave nodes, while the second and fourth operation blocks indicate writing a certain amount of data to the registers of different slave nodes.

If you want to add more communication requirements on the same ModbusTCP master node and slave node, you can create the required operation blocks for the connection, and set communication parameters by referring to the example.

3.1.4 ModbusTCP slave node

The following describes the variables in the ModbusTCP slave node command library.

✧ Variable definitions

Variable	Type		Function	Remarks
Enable	INPUT	BOOL	Indicates whether to activate the ModbusTCP_Slave function.	0: No 1: Yes
Port		DINT	Slave node port number.	The default value 502 is used.
Unit		INT	Slave node ID.	Slave node ID (1-247)
Done	OUTPUT	BOOL	Completion flag.	0: The command is being executed. 1: The command has been executed.
IP		STRING	Slave node IP address.	Local IP address (change disallowed).
Error		BOOL	Error flag	0: No error 1: An error occurred.
ErrorID		INT	Error ID.	See the ModbusTCP error ID table.

✧ Use instructions

Step 1 The ModbusTCP master node reads data from the ModbusTCP slave node.

Set Enable to 1 to activate the ModbusTCP slave node. If the master node function code is 0x03 reading holding register, set InputSize, set the data arrays with the size specified by InputSize to store the data to read, and assign data array addresses to the Inputs pointer. If the master node function code is 0x01 reading coils, the operation is similar.

Step 2 The ModbusTCP master node writes data to the ModbusTCP slave node.

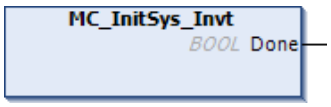
Set Enable to 1 to make the ModbusRTU slave node in activated state. If the master node function code is 0x10 writing multiple registers, define a storage zone in the variable area, of which the size cannot be smaller than the size of data that the ModbusTCP master node will write. If the master node function code is 0x0F writing multiple coils, the operation is similar.

3.2 Pulse commands

3.2.1 MC_InitSys_Invt

MC_InitSys_Invt: initialization command of the motion control module axis.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
MC_InitSys_Invt	Axis initialization command		`MC_InitSys_Invt_0(Done=> );`

(2) Related variables

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Mode	Execution condition	BOOL	TRUE, FALSE	TRUE	TRUE: Pulse type. FALSE: Bus type. You must restart the controller after the setting change. Otherwise, unknown situation may occur during running.
Period	Execution condition	UINT	-	1	Running cycle. For the pulse type, the value must be 1. For the bus type, the value must be greater than 0. You must restart the controller after the setting change. Otherwise, unknown situation may occur during running.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Command execution	BOOL	TRUE, FALSE	FALSE	TRUE: Initialization completed. FALSE: Initialization incompleted.

Output variable	Name	Data type	Valid range	Initial value	Description
	completion				
Error	Axis error flag	BOOL	TRUE, FALSE	FALSE	TRUE: An axis error occurs. FALSE: No axis error occurs.
ErrorID	Error ID	UINT	-	0	Default value: 0 Value > 100: Alarm ID. Value < 100: Error ID.

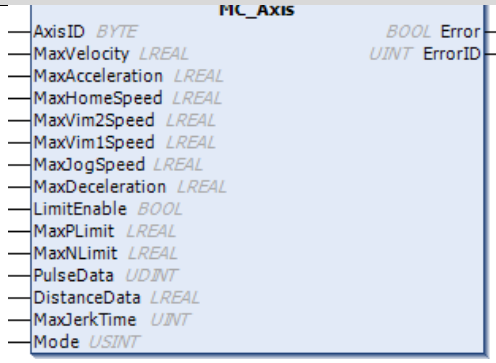
(3) Function description

The function block is used to initialize a motion control module. When Done is TRUE, the initialization is completed. Otherwise, the initialization is not completed.

3.2.2 MC_Axis

MC_Axis: axis defintion command.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
MC_Axis	Axis definition command		MC_Axis(AxisID:= , MaxVelocity:= , MaxAcceleration:= , MaxHomeSpeed:= , MaxVim2Speed:= , MaxVim1Speed:= , MaxJogSpeed:= , MaxDeceleration:= , LimitEnable:= , MaxPLimit:= , MaxNLimit:= , PulseData:= , DistanceData:= , MaxJerkTime:= , Mode:= , Error=> , ErrorID=>);

(2) Related variables

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
AxisID	Axis ID	BYTE	0-3	0	Axis ID (0-3).
MaxVelocity	Max. running speed of axis	LREAL	1-5000	2000	Max. running speed of the present axis. Unit: mm/min
MaxAcceleration	Max. acceleration of axis	LREAL	Positive number or 0	1800000	Max. acceleration of the present axis. Unit: mm/min^2
MaxHomeSpeed	Max. homing speed of axis	LREAL	1-5000	1000	Max. homing speed of the present axis. Unit: mm/min
MaxVim2Speed	Max. homing speed at	LREAL	1-50	20	Max. running speed at step 2 of the present axis. Unit: mm/min

Input variable	Name	Data type	Valid range	Initial value	Description
	step 2 of axis				
MaxVim1Speed	Max. homing speed at step 1 of axis	LREAL	50-600	200	Max. homing speed at step 1 of the present axis. Unit: mm/min
MaxJogSpeed	Max. jogging speed of axis	LREAL	1-5000	1000	Max. jogging speed of the present axis. Unit: mm/min
MaxDeceleration	Max. deceleration speed of axis	LREAL	Positive number or 0	1800000	Max. deceleration of the present axis. Unit: mm/min^2
LimitEnable	Software position limit flag	BOOL	TRUE, FALSE	TRUE	TRUE: Enable the software position limit function. FALSE: Disable the software position limit function. (Not considered in single-axis speed mode)
MaxPLimit	Max. forward limit position	LREAL	Positive number or 0	999999.999	Max. forward limit position of the present axis. Unit: mm
MaxNLimit	Max. reverse limit position	LREAL	Negative number or 0	-999999.999	Max. reverse limit position of the present axis. Unit: mm
PulseData	Number of pulses required for one rotation of axis	UDINT	Positive number or 0	10000	Number of pulses required for one rotation of the present axis. Unit: pulse
DistanceData	Distance of one revolution of axis	LREAL	Positive number or 0	10	Distance of one revolution of the present axis. Unit: mm
MaxJerkTime	Duration of the max. jerk of axis	UINT	10-400	100	Duration of the max. jerk of the present axis.
Mode	Axis control mode	USINT	0-2	2	Control mode of the present axis. 0: "Pulse + Sign" mode 1: FWD/REV pulse train mode 2: Quadrature encoding pulse mode

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Error	Axis error flag	BOOL	TRUE, FALSE	FALSE	TRUE: An axis error occurs. FALSE: No axis error occurs.
ErrorID	Error ID	UINT	——	0	Default value: 0 Value > 100: Alarm ID. Value < 100: Error ID.

(3) Function description

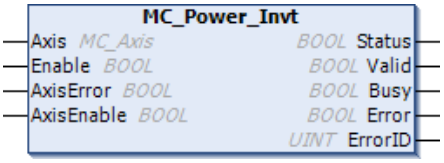
It is used to declare the axis ID and control parameters.

3.2.3 MC_Power_Invt

MC_Power_Invt: axis enabling command.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

MC_Power_Invt	Axis enabling command		`MC_Power_Invt(`
			`Axis:= ,`
			`Enable:= ,`
			`AxisError:= ,`
			`AxisEnable:= ,`
			`Status=> ,`
			`Valid=> ,`
			`Busy=> ,`
			`Error=> ,`
			`ErrorID=> );`

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	MC_Axis	-	-	Used to specify the axis ID (0-3) and axis control settings.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Enable	Execution condition	BOOL	TRUE, FALSE	FALSE	Function block enabling bit, triggered at high level.
AxisError	Axis alarm flag	BOOL	TRUE, FALSE	FALSE	TRUE: Report an alarm. FALSE: Do not report an alarm.
AxisEnable	Axis enabling flag	BOOL	TRUE, FALSE	FALSE	TRUE: The axis has been enabled. FALSE: The axis is not enabled.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Status	Ready to run	BOOL	TRUE, FALSE	FALSE	TRUE: Ready to run FALSE: Not ready to run
Valid	Axis enabling flag	BOOL	TRUE, FALSE	FALSE	TRUE: The axis has been enabled. FALSE: The axis is not enabled.
Busy	Executing	BOOL	TRUE, FALSE	FALSE	TRUE: The function block is being executed. FALSE: The function block is not executed.
Error	Axis error flag	BOOL	TRUE, FALSE	FALSE	TRUE: An axis error occurs. FALSE: No axis error occurs.
ErrorID	Error ID	UINT	-	0	Default value: 0 Value > 100: Alarm ID. Value < 100: Error ID.

(3) Function description

It is used to enable the motion control module axis. During the control process, the motion module uses the axis to perform related operation control functions only after the axis does not generate any alarm and the servo enabling is completed. The output reflects the function block running status, whether errors occur, and error information if any.

3.2.4 MC_Stop_Invt

MC_Stop_Invt: axis stop command.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
MC_Stop_Invt	Axis stop command		MC_Stop_Invt(Axis:= , Enable:= , Deceleration:= , JerkTime:= , Mode:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	MC_Axis	-	-	Used to specify the axis ID (0-3) and axis control settings.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Enable	Execution condition	BOOL	TRUE, FALSE	FALSE	Function block enabling bit, triggered at the rising edge.
Deceleration	Deceleration	LREAL	Positive number or 0	1800000	Deceleration. Unit: mm/min² (When the deceleration is set improperly, the deceleration in the axis control function is used.)
JerkTime	Jerk duration	UINT	0-400	0	Jerk duration.
Mode	Deceleration stop mode	BOOL	TRUE, FALSE	TRUE	TRUE: Decelerate to stop. FALSE: Stop immediately.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Command execution completion	BOOL	TRUE, FALSE	FALSE	TRUE: Initialization completed. FALSE: Initialization incompleted.
Busy	Executing	BOOL	TRUE, FALSE	FALSE	TRUE: The function block is being executed. FALSE: The function block is not executed.
Error	Axis error flag	BOOL	TRUE, FALSE	FALSE	TRUE: An axis error occurs. FALSE: No axis error occurs.
ErrorID	Error ID	UINT	-	0	Default value: 0 Value > 100: Alarm ID. Value < 100: Error ID.

(3) Function description

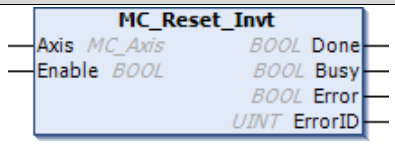
It is used by the control module. It supports single-axis stop during the control process, and supports T-type and S-type acceleration and deceleration planning. JerkTime>0 corresponds to S-type acceleration and deceleration planning, while JerkTime=0 corresponds to T-type acceleration and deceleration planning. It supports the deceleration stop and immediate stop control modes.

3.2.5 MC_Reset_Invt

MC_Reset_Invt: axis reset command.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

MC_Reset_Invt	Axis reset command		`MC_Reset_Invt(`
			`Axis:= ,`
			`Enable:= ,`
			`Done=> ,`
			`Busy=> ,`
			`Error=> ,`
			`ErrorID=> );`

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	MC_Axis	-	-	Used to specify the axis ID (0-3) and axis control settings.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Enable	Execution condition	BOOL	TRUE, FALSE	FALSE	Function block enabling bit, triggered at the rising edge.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Command execution completion	BOOL	TRUE, FALSE	FALSE	TRUE: Initialization completed. FALSE: Initialization incompleted.
Busy	Executing	BOOL	TRUE, FALSE	FALSE	TRUE: The function block is being executed. FALSE: The function block is not executed.
Error	Axis error flag	BOOL	TRUE, FALSE	FALSE	TRUE: An axis error occurs. FALSE: No axis error occurs.
ErrorID	Error ID	UINT	-	0	Default value: 0 Value > 100: Alarm ID. Value < 100: Error ID.

(3) Function description

It is used by the axis reset control module for single-axis control. If the stop function is performed in any situation, you must use this module to cancel the stop state.

3.2.6 MC_JOG_Invt

MC_JOG_Invt: jogging command.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

MC_JOG_Invt	Jogging command		`MC_JOG_Invt(`
			`Axis:= ,`
			`JogDirection:= ,`
			`RunStart:= ,`
			`Velocity:= ,`
			`Acceleration:= ,`
			`Deceleration:= ,`
			`Busy=> ,`
			`CommandAborted=> ,`
			`Error=> ,`
			`ErrorID=> );`

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	MC_Axis	-	-	Used to specify the axis ID (0-3) and axis control settings.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
JogDirection	Motion direction	BOOL	TRUE, FALSE	FALSE	TRUE: Forward motion FALSE: Reverse motion
RunStart	Execution condition	BOOL	TRUE, FALSE	FALSE	Function block enabling bit, triggered at the rising edge.
Velocity	Speed	LREAL	Positive number or 0	500	Jogging speed. Unit: mm/min
Acceleration	Acceleration	LREAL	Positive number or 0	1800000	Acceleration. Unit: mm/min²

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Busy	Executing	BOOL	TRUE, FALSE	FALSE	TRUE: The function block is being executed. FALSE: The function block is not executed.
CommandAborted	Command interruption flag	BOOL	TRUE, FALSE	FALSE	TRUE: Interrupted FALSE: Not interrupted
Error	Axis error flag	BOOL	TRUE, FALSE	FALSE	TRUE: An axis error occurs. FALSE: No axis error occurs.
ErrorID	Error ID	UINT	-	0	Default value: 0 Value > 100: Alarm ID. Value < 100: Error ID.

(3) Function description

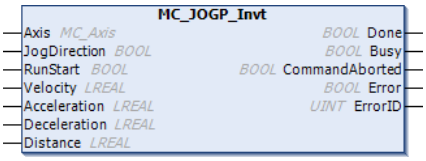
It is used to enable a single axis to jog. It allows you to set the jogging speed, acceleration and deceleration of the axis. The output returns the function block execution status, whether an error occurs, and error information if any.

3.2.7 MC_JOGP_Invt

MC_JOGP_Invt: axis point-to-point (PTP) motion command.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

MC_JOGP_Invt	PTP motion command		`MC_JOGP_Invt(`
			`Axis:= ,`
			`JogDirection:= ,`
			`RunStart:= ,`
			`Velocity:= ,`
			`Acceleration:= ,`
			`Deceleration:= ,`
			`Distance:= ,`
			`Done=> ,`
			`Busy=> ,`
			`CommandAborted=> ,`
			`Error=> ,`
			`ErrorID=> );`

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	MC_Axis	-	-	Used to specify the axis ID (0-3) and axis control settings.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
JogDirection	Motion direction	BOOL	TRUE, FALSE	FALSE	TRUE: Forward motion FALSE: Reverse motion
RunStart	Execution condition	BOOL	TRUE, FALSE	FALSE	Function block enabling bit, triggered at the rising edge.
Velocity	Speed	LREAL	Positive number or 0	500	PTP motion speed. Unit: mm/min
Acceleration	Acceleration	LREAL	Positive number or 0	1800000	Acceleration. Unit: mm/min²
Distance	Moving distance	LREAL	Data range	0.5	Moving distance. Unit: mm

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Command execution completion	BOOL	TRUE, FALSE	FALSE	TRUE: Initialization completed. FALSE: Initialization incompleted.
Busy	Executing	BOOL	TRUE, FALSE	FALSE	TRUE: The function block is being executed. FALSE: The function block is not executed.
CommandAborted	Command interruption flag	BOOL	TRUE, FALSE	FALSE	TRUE: Interrupted FALSE: Not interrupted

Output variable	Name	Data type	Valid range	Initial value	Description
Error	Axis error flag	BOOL	TRUE, FALSE	FALSE	TRUE: An axis error occurs. FALSE: No axis error occurs.
ErrorID	Error ID	UINT	-	0	Default value: 0 Value > 100: Alarm ID. Value < 100: Error ID.

(3) Function description

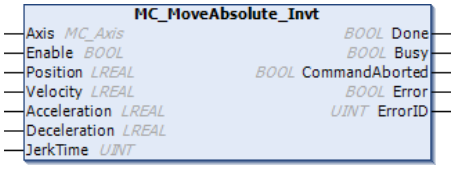
It is used to enable a single axis to perform PTP motion. It allows you to set the motion speed, acceleration and deceleration of the axis. The axis can be positioned to 0.001mm through the function. The output returns the function block execution status, whether an error occurs, and error information if any.

3.2.8 MC_MoveAbsolute_Invt

MC_MoveAbsolute_Invt: single-axis absolute position control command.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

MC_Move Absolute_Invt	Single-axis absolute position control command		`MC_MoveAbsolute_Invt(`
			`Axis:= ,`
			`Enable:= ,`
			`Position:= ,`
			`Velocity:= ,`
			`Acceleration:= ,`
			`Deceleration:= ,`
			`JerkTime:= ,`
			`Done=> ,`
			`Busy=> ,`
			`CommandAborted=> ,`
			`Error=> ,`
			`ErrorID=> );`

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	MC_Axis	-	-	Used to specify the axis ID (0-3) and axis control settings.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Enable	Execution condition	BOOL	TRUE, FALSE	FALSE	Function block enabling bit, triggered at the rising edge.
Position	Distance	LREAL	Data range	0	Distance
Velocity	Speed	LREAL	Positive number	1000	Speed. Unit: mm/min
Acceleration	Acceleration	LREAL	Positive number or 0	1800000	Acceleration. Unit: mm/min²
JerkTime	Jerk duration	UINT	0-400	0	Jerk duration.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Command execution completion	BOOL	TRUE, FALSE	FALSE	TRUE: Initialization completed. FALSE: Initialization incompleted.
Busy	Executing	BOOL	TRUE, FALSE	FALSE	TRUE: The function block is being executed. FALSE: The function block is not executed.
CommandAborted	Command interruption flag	BOOL	TRUE, FALSE	FALSE	TRUE: Interrupted FALSE: Not interrupted
Error	Axis error flag	BOOL	TRUE, FALSE	FALSE	TRUE: An axis error occurs. FALSE: No axis error occurs.
ErrorID	Error ID	UINT	-	0	Default value: 0 Value > 100: Alarm ID. Value < 100: Error ID.

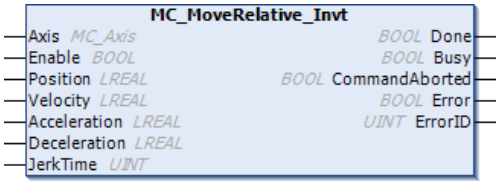
(3) Function description

It is used for single-axis absolute position control. It supports T-type and S-type acceleration and deceleration planning. JerkTime>0 corresponds to S-type acceleration and deceleration planning, while JerkTime=0 corresponds to T-type acceleration and deceleration planning. The output returns the function block execution status, whether an error occurs, and error information if any.

3.2.9 MC_MoveRelative_Invt

MC_MoveRelative_Invt: single-axis relative position control command.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
MC_MoveRelative_Invt	Single-axis relative-position control command		MC_MoveRelative_Invt(Axis:= , Enable:= , Position:= , Velocity:= , Acceleration:= , Deceleration:= , JerkTime:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	MC_Axis	-	-	Used to specify the axis ID (0-3) and axis control settings.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Enable	Execution condition	BOOL	TRUE, FALSE	FALSE	Function block enabling bit, triggered at the rising edge.
Position	Distance	LREAL	Data range	0	Distance
Velocity	Speed	LREAL	Positive number	1000	Speed. Unit: mm/min
Acceleration	Acceleration	LREAL	Positive number or 0	1800000	Acceleration. Unit: mm/min²
JerkTime	Jerk duration	UINT	0-400	0	Jerk duration.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Command execution completion	BOOL	TRUE, FALSE	FALSE	TRUE: Initialization completed. FALSE: Initialization incompleted.
Busy	Executing	BOOL	TRUE, FALSE	FALSE	TRUE: The function block is being executed. FALSE: The function block is not executed.
CommandAborted	Command interruption flag	BOOL	TRUE, FALSE	FALSE	TRUE: Interrupted FALSE: Not interrupted
Error	Axis error flag	BOOL	TRUE, FALSE	FALSE	TRUE: An axis error occurs. FALSE: No axis error occurs.
ErrorID	Error ID	UINT	-	0	Default value: 0 Value > 100: Alarm ID. Value < 100: Error ID.

(3) Function description

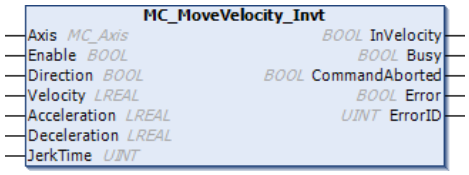
It is used for single-axis relative position control. It supports T-type and S-type acceleration and deceleration planning. JerkTime>0 corresponds to S-type acceleration and deceleration planning, while JerkTime=0 corresponds to T-type acceleration and deceleration planning. The output returns the function block execution status, whether an error occurs, and error information if any.

3.2.10 MC_MoveVelocity_Invt

MC_MoveVelocity_Invt: single-axis speed control command.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

MC_MoveVelocity_Invt	Single-axis speed control command		`MC_MoveVelocity_Invt(`
			`Axis:= ,`
			`Enable:= ,`
			`Direction:= ,`
			`Velocity:= ,`
			`Acceleration:= ,`
			`Deceleration:= ,`
			`JerkTime:= ,`
			`InVelocity=> ,`
			`Busy=> ,`
			`CommandAborted=> ,`
			`Error=> ,`
			`ErrorID=> );`

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	MC_Axis	-	-	Used to specify the axis ID (0-3) and axis control settings.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Enable	Execution condition	BOOL	TRUE, FALSE	FALSE	Function block enabling bit, triggered at the rising edge.
Direction	Motion direction	BOOL	TRUE, FALSE	TRUE	Direction. TRUE: Forward FALSE: Reverse
Velocity	Speed	LREAL	Positive number or 0	500	Speed. Unit: mm/min
Acceleration	Acceleration	LREAL	Positive number or 0	1800000	Acceleration. Unit: mm/min²
JerkTime	Jerk duration	UINT	0-400	0	Jerk duration.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
InVelocity	Flag indicating the speed value is	BOOL	TRUE, FALSE	FALSE	TRUE: The speed value is reached for the first time.

Output variable	Name	Data type	Valid range	Initial value	Description
	reached for the first time				
Busy	Executing	BOOL	TRUE, FALSE	FALSE	TRUE: The function block is being executed. FALSE: The function block is not executed.
CommandAborted	Command interruption flag	BOOL	TRUE, FALSE	FALSE	TRUE: Interrupted FALSE: Not interrupted
Error	Axis error flag	BOOL	TRUE, FALSE	FALSE	TRUE: An axis error occurs. FALSE: No axis error occurs.
ErrorID	Error ID	UINT	-	0	Default value: 0 Value > 100: Alarm ID. Value < 100: Error ID.

(3) Function description

It is used for single-axis speed control. It is the flag indicating no need to complete homing in the control process. It supports T-type and S-type acceleration and deceleration planning. JerkTime>0 corresponds to S-type acceleration and deceleration planning, while JerkTime=0 corresponds to T-type acceleration and deceleration planning. The output returns the function block execution status, whether an error occurs, and error information if any.

3.2.11 MC_ReadActualPosition_Invt

MC_ReadActualPosition_Invt: axis actual position reading command.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

MC_ReadActual Position_Invt	Axis actual position reading command		`MC_ReadActualPosition_Invt(`
			`Axis:= ,`
			`Enable:= ,`
			`Busy=> ,`
			`Value=> ,`
			`Error=> ,`
			`ErrorID=> );`

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	MC_Axis	-	-	Used to specify the axis ID (0-3) and axis control settings.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Enable	Execution condition	BOOL	TRUE, FALSE	FALSE	Function block enabling bit, triggered at the rising edge.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Busy	Executing	BOOL	TRUE, FALSE	FALSE	TRUE: The function block is being executed. FALSE: The function block is not executed.
Value	Position value	LREAL	Data range	0	Position parameter return value.
Error	Axis error flag	BOOL	TRUE, FALSE	FALSE	TRUE: An axis error occurs. FALSE: No axis error occurs.
ErrorID	Error ID	UINT	-	0	Default value: 0 Value > 100: Alarm ID. Value < 100: Error ID.

(3) Function description

It is used to read the actual position of an axis. The input variable Axis determines which axis is read. The output variable Value returns the actual coordinate position of the axis.

3.2.12 MC_ActiveSpeed_Invt

MC_ActiveSpeed_Invt: axis actual speed reading command.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

MC_ActiveSpeed_Invt	Axis actual speed reading command		`MC_ActiveSpeed_Invt(`
			`Axis:= ,`
			`Enable:= ,`
			`Busy=> ,`
			`Error=> ,`
			`ErrorID=> ,`
			`ActiveSpeed=> );`

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	MC_Axis	-	-	Used to specify the axis ID (0-3) and axis control settings.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Enable	Execution condition	BOOL	TRUE, FALSE	FALSE	Function block enabling bit, triggered at high level.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Busy	Executing	BOOL	TRUE, FALSE	FALSE	TRUE: The function block is being executed. FALSE: The function block is not executed.
Error	Axis error flag	BOOL	TRUE, FALSE	FALSE	TRUE: An axis error occurs. FALSE: No axis error occurs.
ErrorID	Error ID	UINT	-	0	Default value: 0 Value > 100: Alarm ID. Value < 100: Error ID.
ActiveSpeed	Speed	LREAL	Data range	0	Speed parameter return value.

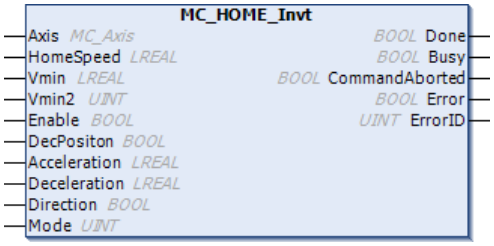
(3) Function description

It is used to read the actual speed of an axis. The input variable Axis determines which axis is read. The output variable ActiveSpeed returns the actual running speed of the axis.

3.2.13 MC_HOME_Invt

MC_HOME_Invt: axis homing command.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
MC_HOME_Invt	Axis homing command		MC_HOME_Invt(Axis:= , HomeSpeed:= , Vmin:= , Vmin2:= , Enable:= , DecPositon:= , Acceleration:= , Deceleration:= , Direction:= , Mode:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	MC_Axis	-	-	Used to specify the axis ID (0-3) and axis control settings.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
HomeSpeed	Homing speed	LREAL	Positive number	800	Homing speed

Input variable	Name	Data type	Valid range	Initial value	Description
Vmin	Speed 1	LREAL	50-600	200	Running speed at the deceleration block.
Vmin2	Speed 2	LREAL	1-50	30	Running speed when waiting for the Z signal of motor.
Enable	Execution condition	BOOL	TRUE, FALSE	FALSE	Function block enabling bit, triggered at the rising edge.
DecPositon	Deceleration block signal	BOOL	TRUE, FALSE	FALSE	Deceleration block signal
Acceleration	Acceleration	LREAL	Positive number or 0	1800000	Acceleration. Unit: mm/min²
Direction	Homing direction	BOOL	TRUE, FALSE	TRUE	Homing direction. TRUE: Forward FALSE: Reverse
Mode	Homing mode	UINT	-	1	Homing mode (It does not work currently.)

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Command execution completion	BOOL	TRUE, FALSE	FALSE	TRUE: The mechanical origin is found. FALSE: The mechanical origin is not found.
Busy	Executing	BOOL	TRUE, FALSE	FALSE	TRUE: The function block is being executed. FALSE: The function block is not executed.
Error	Axis error flag	BOOL	TRUE, FALSE	FALSE	TRUE: An axis error occurs. FALSE: No axis error occurs.
ErrorID	Error ID	UINT	-	0	Default value: 0 Value > 100: Alarm ID. Value < 100: Error ID.
ActiveSpeed	Speed	LREAL	Data range	0	Speed parameter return value.

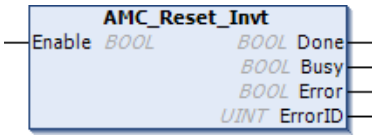
(3) Function description

It is used for an axis to find the zero point control function of the mechanism. The homing module can be customized according to the structure of the mechanism. Currently, only one homing method is supported. The homing of four axes is supported at the same time. The output returns the function block execution status, whether an error occurs, and error information if any.

3.2.14 AMC_Reset_Invt

AMC_Reset_Invt: multi-axis reset command.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
AMC_Reset_Invt	Multi-axis reset command		AMC_Reset_Invt(Enable:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Enable	Execution condition	BOOL	TRUE, FALSE	FALSE	Function block enabling bit, triggered at the rising edge.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Command execution completion	BOOL	TRUE, FALSE	FALSE	TRUE: The mechanical origin is found. FALSE: The mechanical origin is not found.
Busy	Executing	BOOL	TRUE, FALSE	FALSE	TRUE: The function block is being executed. FALSE: The function block is not executed.
Error	Axis error flag	BOOL	TRUE, FALSE	FALSE	TRUE: An axis error occurs. FALSE: No axis error occurs.
ErrorID	Error ID	UINT	-	0	Default value: 0 Value > 100: Alarm ID. Value < 100: Error ID.

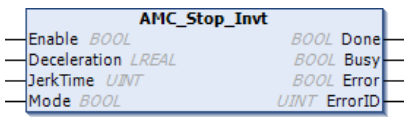
(3) Function description

It is used by the axis reset control module for multi-axis control. If the stop function is performed in any situation, you must use this module to cancel the stop state.

3.2.15 AMC_Stop_Invt

AMC_Stop_Invt: multi-axis stop command.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
AMC_Stop_Invt	Multi-axis stop command		AMC_Stop_Invt(Enable:= , Deceleration:= , JerkTime:= , Mode:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Enable	Execution condition	BOOL	TRUE, FALSE	FALSE	Function block enabling bit, triggered at the rising edge.
Deceleration	Deceleration	LREAL	Positive number or 0	1800000	Deceleration. Unit: mm/min² (When the deceleration is set improperly, the deceleration in the axis control function is used.)
JerkTime	Jerk duration	UINT	0-400	0	Jerk duration.
Mode	Stop mode	BOOL	TRUE, FALSE	TRUE	TRUE: Decelerate to stop. FALSE: Stop immediately.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Command execution completion	BOOL	TRUE, FALSE	FALSE	TRUE: The axis has stopped. FALSE: The axis is decelerating to stop.
Busy	Executing	BOOL	TRUE, FALSE	FALSE	TRUE: The function block is being executed. FALSE: The function block is not executed.
Error	Axis error flag	BOOL	TRUE, FALSE	FALSE	TRUE: An axis error occurs. FALSE: No axis error occurs.
ErrorID	Error ID	UINT	-	0	Default value: 0 Value > 100: Alarm ID. Value < 100: Error ID.

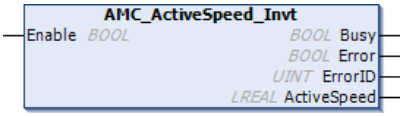
(3) Function description

It is used to stop the control module. It supports stopping specified axes during the control process, and supports T-type and S-type acceleration and deceleration planning. JerkTime>0 corresponds to S-type acceleration and deceleration planning, while JerkTime=0 corresponds to T-type acceleration and deceleration planning. It supports the deceleration stop and immediate stop control modes.

3.2.16 AMC_ActiveSpeed_Invt

AMC_ActiveSpeed_Invt: multi-axis composite speed reading command.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
AMC_ActiveSpeed_Invt	Multi-axis composite speed reading command		AMC_ActiveSpeed_Invt(Enable:= , Busy=> , Error=> , ErrorID=> , ActiveSpeed=>);

(2) Related variables

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Enable	Execution condition	BOOL	TRUE, FALSE	FALSE	Function block enabling bit, triggered at high level.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Busy	Executing	BOOL	TRUE, FALSE	FALSE	TRUE: The function block is being executed. FALSE: The function block is not executed.
Error	Axis error flag	BOOL	TRUE, FALSE	FALSE	TRUE: An axis error occurs. FALSE: No axis error occurs.
ErrorID	Error ID	UINT	-	0	Default value: 0 Value > 100: Alarm ID. Value < 100: Error ID.
ActiveSpeed	Speed	LREAL	Data range	0	Axis actual speed.

(3) Function description

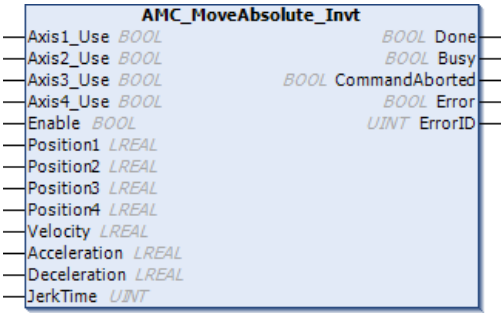
It is used to read the actual speeds of specified axes. The output variable ActiveSpeed returns the actual running speeds of the axes.

3.2.17 AMC_MoveAbsolute_Invt

AMC_MoveAbsolute_Invt: multi-axis absolute position control command.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

AMC_Move Absolute_Invt	Multi-axis absolute position control command		`AMC_MoveAbsolute_Invt(`
			`Axis1_Use:= ,`
			`Axis2_Use:= ,`
			`Axis3_Use:= ,`
			`Axis4_Use:= ,`
			`Enable:= ,`
			`Position1:= ,`
			`Position2:= ,`
			`Position3:= ,`
			`Position4:= ,`
			`Velocity:= ,`
			`Acceleration:= ,`
			`Deceleration:= ,`
			`JerkTime:= ,`
			`Done=> ,`
			`Busy=> ,`
			`CommandAborted=> ,`
			`Error=> ,`
			`ErrorID=> );`

(2) Related variables

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Axis1_Use	Use flag of axis 1	BOOL	TRUE, FALSE	FALSE	Use flag of axis 1. TRUE: Used FALSE: Not used
Axis2_Use	Use flag of axis 2	BOOL	TRUE, FALSE	FALSE	Use flag of axis 2. TRUE: Used FALSE: Not used
Axis3_Use	Use flag of axis 3	BOOL	TRUE, FALSE	FALSE	Use flag of axis 3. TRUE: Used FALSE: Not used
Axis4_Use	Use flag of axis 4	BOOL	TRUE, FALSE	FALSE	Use flag of axis 4. TRUE: Used FALSE: Not used
Enable	Execution condition	BOOL	TRUE, FALSE	FALSE	Function block enabling bit, triggered at the rising edge.
Position1	Distance of axis 1	LREAL	Data range	0	Distance of axis 1
Position2	Distance of axis 2	LREAL	Data range	0	Distance of axis 2
Position3	Distance of axis 3	LREAL	Data range	0	Distance of axis 3
Position4	Distance of axis 4	LREAL	Data range	0	Distance of axis 4
Velocity	Speed	LREAL	Positive number or 0	500	Speed. Unit: mm/min

Input variable	Name	Data type	Valid range	Initial value	Description
Acceleration	Acceleration	LREAL	Positive number or 0	1800000	Acceleration. Unit: mm/min²
JerkTime	Jerk duration	UINT	0-400	0	Jerk duration.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Command execution completion	BOOL	TRUE, FALSE	FALSE	Completion flag. TRUE: The specified position is reached. FALSE: The specified position is not reached.
Busy	Executing	BOOL	TRUE, FALSE	FALSE	TRUE: The function block is being executed. FALSE: The function block is not executed.
CommandAborted	Command interruption flag	BOOL	TRUE, FALSE	FALSE	TRUE: Interrupted. FALSE: Not interrupted.
Error	Axis error flag	BOOL	TRUE, FALSE	FALSE	TRUE: An axis error occurs. FALSE: No axis error occurs.
ErrorID	Error ID	UINT	-	0	Default value: 0 Value > 100: Alarm ID. Value < 100: Error ID.

(3) Function description

The multi-axis absolute position mode supports simultaneous movement of four axes. The absolute position is specified by Position. You need to set the corresponding movement direction, speed, acceleration, and other information and make Enable to start the function block by changing the value from FALSE to TRUE, and specify the axes to move from the present positions to the absolute positions.

The multi-axis absolute position mode has the same principle as the single-axis absolute position mode. The difference is that the multi-axis absolute position mode can control multiple axes to move to absolute positions at the same time.

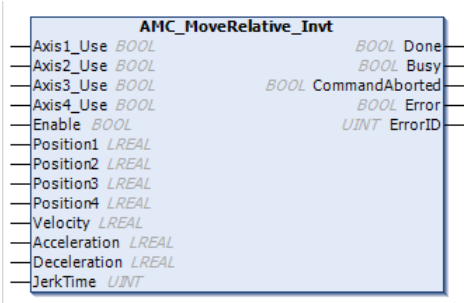
The multi-axis absolute position mode supports T-type and S-type acceleration and deceleration planning.

3.2.18 AMC_MoveRelative_Invt

AMC_MoveRelative_Invt: multi-axis relative position control command.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

AMC_Move Relative_Invt	Multi-axis relative position control command		`AMC_MoveRelative_Invt(`
			`Axis1_Use:= ,`
			`Axis2_Use:= ,`
			`Axis3_Use:= ,`
			`Axis4_Use:= ,`
			`Enable:= ,`
			`Position1:= ,`
			`Position2:= ,`
			`Position3:= ,`
			`Position4:= ,`
			`Velocity:= ,`
			`Acceleration:= ,`
			`Deceleration:= ,`
			`JerkTime:= ,`
			`Done=> ,`
			`Busy=> ,`
			`CommandAborted=> ,`
			`Error=> ,`
			`ErrorID=> );`

(2) Related variables

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Axis1_Use	Use flag of axis 1	BOOL	TRUE, FALSE	FALSE	Use flag of axis 1. TRUE: Used FALSE: Not used
Axis2_Use	Use flag of axis 2	BOOL	TRUE, FALSE	FALSE	Use flag of axis 2. TRUE: Used FALSE: Not used
Axis3_Use	Use flag of axis 3	BOOL	TRUE, FALSE	FALSE	Use flag of axis 3. TRUE: Used FALSE: Not used
Axis4_Use	Use flag of axis 4	BOOL	TRUE, FALSE	FALSE	Use flag of axis 4. TRUE: Used FALSE: Not used
Enable	Execution condition	BOOL	TRUE, FALSE	FALSE	Function block enabling bit, triggered at the rising edge.
Position1	Distance of axis 1	LREAL	Data range	0	Distance of axis 1
Position2	Distance of axis 2	LREAL	Data range	0	Distance of axis 2
Position3	Distance of axis 3	LREAL	Data range	0	Distance of axis 3
Position4	Distance of	LREAL	Data range	0	Distance of axis 4

Input variable	Name	Data type	Valid range	Initial value	Description
	axis 4				
Velocity	Speed	LREAL	Positive number or 0	500	Speed. Unit: mm/min
Acceleration	Acceleration	LREAL	Positive number or 0	1800000	Acceleration. Unit: mm/min²
JerkTime	Jerk duration	UINT	0-400	0	Jerk duration.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Command execution completion	BOOL	TRUE, FALSE	FALSE	Completion flag. TRUE: The specified position is reached. FALSE: The specified position is not reached.
Busy	Executing	BOOL	TRUE, FALSE	FALSE	TRUE: The function block is being executed. FALSE: The function block is not executed.
CommandAborted	Command interruption flag	BOOL	TRUE, FALSE	FALSE	Interrupted.
Error	Axis error flag	BOOL	TRUE, FALSE	FALSE	TRUE: An axis error occurs. FALSE: No axis error occurs.
ErrorID	Error ID	UINT	-	0	Default value: 0 Value > 100: Alarm ID. Value < 100: Error ID.

(3) Function description

The multi-axis relative position mode supports simultaneous movement of four axes. The relative position is specified by Position. You need to set the corresponding movement direction, speed, acceleration, and other information and make Enable to start the function block by changing the value from FALSE to TRUE, and specify the axes to move from the present positions to the relative positions.

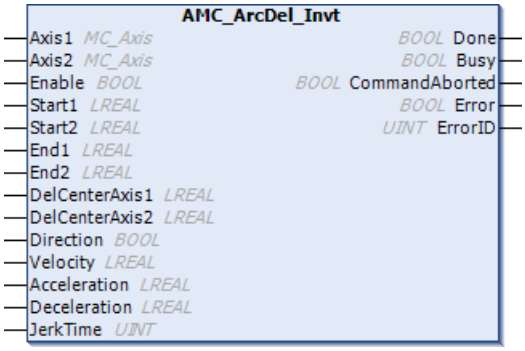
The multi-axis relative position mode has the same principle as the single-axis relative position mode. The difference is that the multi-axis relative position mode can control multiple axes to move to relative positions at the same time.

The multi-axis relative position mode supports T-type and S-type acceleration and deceleration planning.

3.2.19 AMC_ArcDel_Invt

AMC_ArcDel_Invt: planar arc interpolation command.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
AMC_ArcDel_Invt	Plane arc interpolation command		AMC_ArcDel_Invt(Axis1:= , Axis2:= , Enable:= , Start1:= , Start2:= , End1:= , End2:= , DelCenterAxis1:= , DelCenterAxis2:= , Direction:= , Velocity:= , Acceleration:= , Deceleration:= , JerkTime:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Related variables

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Axis1	Axis 1	MC_AXIS	-	-	Used to specify the ID (0-3) and control settings of axis 1.
Axis2	Axis 2	MC_AXIS	-	-	Used to specify the ID (0-3) and control settings of axis 2.
Enable	Execution condition	BOOL	TRUE, FALSE		Function block enabling bit, triggered at the rising edge.
Start1	Start position of axis 1	LREAL	Data range	0	Start position of axis 1
Start2	Start position of axis 2	LREAL	Data range	0	Start position of axis 2
End1	End position of axis 1	LREAL	Data range	0	End position of axis 1
End2	End position of axis 2	LREAL	Data range	0	End position of axis 2
DelCenterAxis1	Distance from the start of axis 1 to the circle center	LREAL	Data range	0	Distance from the start of axis 1 to the circle center
DelCenterAxis2	Distance from the start of axis	LREAL	Data range	0	Distance from the start of axis 2 to the circle center

Input variable	Name	Data type	Valid range	Initial value	Description
	2 to the circle center				
Direction	Running direction	BOOL	TRUE, FALSE	FALSE	TRUE: Clockwise FALSE: Counterclockwise
Velocity	Speed	LREAL	Positive number or 0	500	Speed. Unit: mm/min
Acceleration	Acceleration	LREAL	Positive number or 0	1800000	Acceleration. Unit: mm/min²
JerkTime	Jerk duration	UINT	0-400	0	Jerk duration.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Command execution completion	BOOL	TRUE, FALSE	FALSE	Completion flag. TRUE: The specified position is reached. FALSE: The specified position is not reached.
Busy	Executing	BOOL	TRUE, FALSE	FALSE	TRUE: The function block is being executed. FALSE: The function block is not executed.
CommandAborted	Command interruption flag	BOOL	TRUE, FALSE	FALSE	TRUE: Interrupted. FALSE: Not interrupted.
Error	Axis error flag	BOOL	TRUE, FALSE	FALSE	TRUE: An axis error occurs. FALSE: No axis error occurs.
ErrorID	Error ID	UINT	-	0	Default value: 0 Value > 100: Alarm ID. Value < 100: Error ID.

(3) Function description

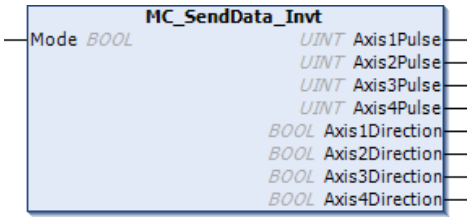
It is used for planar arc interpolation. It supports T-type and S-type acceleration and deceleration planning. JerkTime>0 corresponds to S-type acceleration and deceleration planning, while JerkTime=0 corresponds to T-type acceleration and deceleration planning. It supports clockwise and counterclockwise running. Using this module needs to specify the start and end positions and the distance from the start to the circle center. Only three planes XY/XZ/ZY are allowed to perform circular interpolation, and errors are reported if additional axis arc interpolation is involved. The output returns the function block execution status, whether an error occurs, and error information if any.

3.2.20 MC_SendData_Invt

MC_SendData: command of sending data to the underlayer.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

MC_SendData _Invt	Data sending command		`MC_SendData_Invt_0(`
			`Mode:= ,`
			`Axis1Pulse=> ,`
			`Axis2Pulse=> ,`
			`Axis3Pulse=> ,`
			`Axis4Pulse=> ,`
			`Axis1Direction=> ,`
			`Axis2Direction=> ,`
			`Axis3Direction=> ,`
			`Axis4Direction=> );`

(2) Related variables

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Mode	Execution condition	BOOL	TRUE, FALSE	TRUE	TRUE: Pulse type FALSE: Bus type

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Axis1Pulse	Pulse feedback of axis 1	UINT	-	-	Pulse feedback of axis 1
Axis2Pulse	Pulse feedback of axis 2	UINT	-	-	Pulse feedback of axis 2
Axis3Pulse	Pulse feedback of axis 3	UINT	-	-	Pulse feedback of axis 3
Axis4Pulse	Pulse feedback of axis 4	UINT	-	-	Pulse feedback of axis 4
Axis1Direction	Direction of axis 1	BOOL	TRUE, FALSE	-	Direction of axis 1. TRUE: Forward FALSE: Reverse
Axis2Direction	Direction of axis 2	BOOL	TRUE, FALSE	-	Direction of axis 2. TRUE: Forward FALSE: Reverse
Axis3Direction	Direction of axis 3	BOOL	TRUE, FALSE	-	Direction of axis 3. TRUE: Forward FALSE: Reverse

Output variable	Name	Data type	Valid range	Initial value	Description
Axis4Direction	Direction of axis 4	BOOL	TRUE, FALSE	-	Direction of axis 4. TRUE: Forward FALSE: Reverse

(3) Function description

It is used to convert axis control data to pulse data and send it to the FPGA for processing. An application program can invoke this function block only once since multiple invoking will cause position deviation.

3.2.21 MC_GetVersion_Invt

MC_GetVersion_Invt: command for obtaining the motion control library version.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
MC_Get Version_Invt	Command for obtaining the motion control library version		`MC_GetVersion_Invt(Version=> );`

(2) Related variables

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Version	Motion control library version	STRING	-	-	Motion control library version

(3) Function description

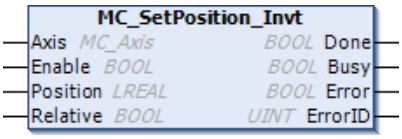
It is used to obtain the motion control library version. The output variable Version returns the version number.

3.2.22 MC_SetPosition_Invt

MC_SetPosition_Invt: axis position setting command.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

MC_SetPosition _Invt	Axis position setting command		`MC_SetPosition_Invt(`
			`Axis:= ,`
			`Enable:= ,`
			`Position:= ,`
			`Relative:= ,`
			`Done=> ,`
			`Busy=> ,`
			`Error=> ,`
			`ErrorID=> );`

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	MC_Axis	-	-	Used to specify the axis ID (0-3) and axis control settings.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Enable	Execution condition	BOOL	TRUE, FALSE	FALSE	Function block enabling bit, triggered at the rising edge.
Position	Position value	LREAL	Positive number or 0	0	Position value
Relative	Relative position flag of a logic address	BOOL	TRUE, FALSE	FALSE	TRUE: Relative position FALSE: Absolute position

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Done	Completion flag	BOOL	TRUE, FALSE	FALSE	Completion flag.
Busy	Executing	BOOL	TRUE, FALSE	FALSE	TRUE: The function block is being executed. FALSE: The function block is not executed.
Error	Axis error flag	BOOL	TRUE, FALSE	FALSE	TRUE: An axis error occurs. FALSE: No axis error occurs.
ErrorID	Error ID	UINT	-	0	Default value: 0 Value > 100: Alarm ID. Value < 100: Error ID.

(3) Function description

It is used to set the axis actual position, which is determined by Axis, Position, and Relative.

3.2.23 MC_ReadParameter_Invt

MC_ReadParameter_Invt: axis parameter reading command.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

MC_ReadParameter_Invt	Axis parameter reading command		`MC_ReadParameter_Invt(`
			`Axis:= ,`
			`Enable:= ,`
			`ParameterNumber:= ,`
			`Valid=> ,`
			`Value=> ,`
			`Error=> ,`
			`ErrorID=> );`

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	MC_Axis	-	-	Used to specify the axis ID (0-3) and axis control settings.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Enable	Execution condition	BOOL	TRUE, FALSE	FALSE	Function block enabling bit, triggered at the rising edge.
ParameterNumber	Parameter number	DINT	-	100	Parameter number

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Valid	Execution validity flag	BOOL	TRUE, FALSE	FALSE	Execution validity flag
Value	Parameter return value	LREAL	-	0	Parameter return value
Error	Axis error flag	BOOL	TRUE, FALSE	FALSE	Axis error flag. TRUE: An axis error occurs. FALSE: No axis error occurs.
ErrorID	Error ID	UINT	-	0	Error ID. Default value: 0 Value > 100: alarm Value <100: error

(3) Function description

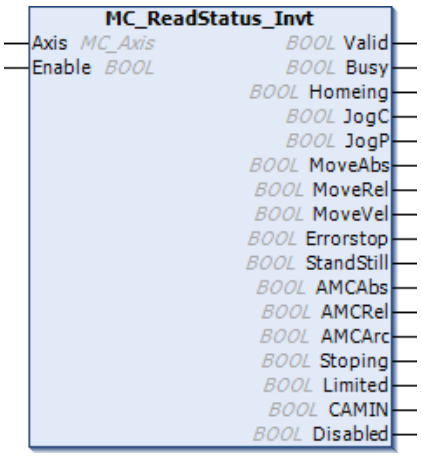
It is used to read the parameters of a specified axis.

3.2.24 MC_ReadStatus_Invt

MC_ReadStatus_Invt: axis motion reading command.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

MC_ReadStatus _Invt	Axis motion reading command		`MC_ReadStatus_Invt(`
			`Axis:= ,`
			`Enable:= ,`
			`Valid=> ,`
			`Busy=> ,`
			`Homeing=> ,`
			`JogC=> ,`
			`JogP=> ,`
			`MoveAbs=> ,`
			`MoveRel=> ,`
			`MoveVel=> ,`
			`Errorstop=> ,`
			`StandStill=> ,`
			`AMCAbs=> ,`
			`AMCRel=> ,`
			`AMCArc=> ,`
			`Stoping=> ,`
			`Limited=> ,`
			`Disabled=> );`

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	MC_Axis	-	-	Used to specify the axis ID (0-3) and axis control settings.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
Enable	Execution condition	BOOL	TRUE, FALSE	FALSE	Function block enabling bit, triggered at high level.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
Valid	Execution validity flag	BOOL	TRUE, FALSE	FALSE	Execution validity flag
Busy	Executing	BOOL	TRUE, FALSE	FALSE	TRUE: The function block is being executed. FALSE: The function block is not executed.
Homeing	Single-axis homing status flag	BOOL	TRUE, FALSE	FALSE	TRUE: In single-axis homing motion FALSE: No single-axis homing motion
JogC	Manual single-axis	BOOL	TRUE, FALSE	FALSE	TRUE: In manual single-axis

Output variable	Name	Data type	Valid range	Initial value	Description
	running flag				continuous motion FALSE: No manual single-axis continuous motion
JogP	Single-axis jogging flag	BOOL	TRUE, FALSE	FALSE	TRUE: In manual single-axis jogging FALSE: No manual single-axis jogging
MoveAbs	Single-axis absolute position running flag	BOOL	TRUE, FALSE	FALSE	TRUE: In single-axis absolute position running FALSE: No single-axis absolute position running
MoveRel	Single-axis relative position running flag	BOOL	TRUE, FALSE	FALSE	TRUE: In single-axis relative position running FALSE: No single-axis relative position running
MoveVel	Flag of running at the single-axis speed	BOOL	TRUE, FALSE	FALSE	TRUE: In the running at the single-axis speed FALSE: No running at the single-axis speed
Errorstop	Alarm flag	BOOL	TRUE, FALSE	FALSE	TRUE: Report an alarm and stop. FALSE: Do not report an alarm.
StandStill	Flag of keeping standstill	BOOL	TRUE, FALSE	FALSE	TRUE: Keep standstill. FALSE: In running.
AMCAbs	Multi-axis absolute position running flag	BOOL	TRUE, FALSE	FALSE	TRUE: In multi-axis absolute position running FLASE: No multi-axis absolute position running
AMCRel	Multi-axis relative position running flag	BOOL	TRUE, FALSE	FALSE	TRUE: In multi-axis relative position running FLASE: No multi-axis relative position running
AMCArc	Multi-axis arc interpolation running flag	BOOL	TRUE, FALSE	FALSE	TRUE: In multi-axis arc interpolation running FLASE: No multi-axis arc interpolation running
Stoping	Flag of decelerating to stop	BOOL	TRUE, FALSE	FALSE	TRUE: Decelerate to stop. FALSE: Do not decelerate to stop.
Limited	Limit flag	BOOL	TRUE, FALSE	FALSE	TRUE: Being limited. FALSE: Not limited.

Output variable	Name	Data type	Valid range	Initial value	Description
					(If limiting has occurred, the value is reset to FALSE.)
CAMIN	E-cam running flag	BOOL	TRUE, FALSE	FALSE	TRUE: In e-cam running FALSE: No e-cam running
Disabled	Axis availability flag	BOOL	TRUE, FALSE	FALSE	TRUE: The axis is available. FALSE: The axis is not available.

(3) Function description

It is used to read the motion of a specified axis.

3.3 Enhanced commands

3.3.1 FB_ReinitDrv

FB_ReinitDrv: automatic reconnection command for servo disconnection.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST
FB_ReinitDrv	Automatic reconnection command for servo disconnection		FB_ReinitDrv(apAxis:= , pEtherCATMaster SoftMotion:= , bEnable:= , bForceReinitETC := , bForceReinitDrv := , bReinitSuccess= > , iAxisCnt=> , bBusy=> , bError=> , iErrorID=>);

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
apAxis	Axis pointer array	ARRAY[1..16] OF POINTER TO AXIS_REF_SM3	-	NULL	Axis pointer array, supporting assignment of up to 16 axis pointers. A pointer without any value assigned points to the address 0 by default.
pEtherCATMasterSoftMotion	EtherCAT master node pointer	POINTER TO IODrvEtherCAT	-	NULL	EtherCAT master node pointer

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
bEnable	Enabling	BOOL	TRUE, FALSE	FALSE	TRUE indicates executing the function block, while FALSE indicates not executing the function block.
bForceReinitEtherCAT	Forcible EtherCAT master node reset	BOOL	TRUE, FALSE	FALSE	EtherCAT communication is forcibly reset at the rising edge. It is used to manually enable communication reset when a function block does not detect an axis failure.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
bReinitSuccess	Reset success	BOOL	TRUE, FALSE	FALSE	TRUE: Communication reset success, and the axis is operational. FALSE: Communication reset failure.
iAxisCnt	Axis count	INT	1-16	0	Number of axes connected.
bBusy	Command being executed	BOOL	TRUE, FALSE	FALSE	TRUE indicates executing the function block, while FALSE indicates not executing the function block.
bError	Fault	BOOL	TRUE, FALSE	FALSE	An error occurred when executing the function block.
iErrorID	Error ID	INT	-	0	Error ID.

(3) Function description

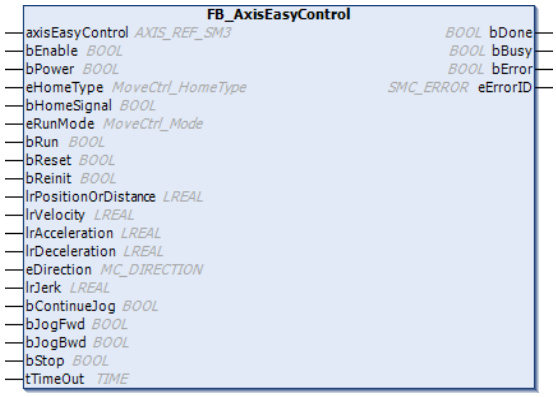
It is an automatic reconnection command for servo disconnection. It is used to implement the real-time detection of the servo communication status and the automatic reconnection after the communication is disconnected. After the communication is successful, relevant information is returned and the servo alarm is cleared. The function block supports the input of up to 16 axes.

3.3.2 FB_AxisEasyControl

FB_AxisEasyControl: single-axis easy control command.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

FB_AxisEasyControl	Single-axis easy control command		`FB_AxisEasyControl(`
			`axisEasyControl:= ,`
			`bEnable:= ,`
			`bPower:= ,`
			`eHomeType:= ,`
			`bHomeSignal:= ,`
			`eRunMode:= ,`
			`bRun:= ,`
			`bReset:= ,`
			`bReinit:= ,`
			`lrPositionOrDistance:= ,`
			`lrVelocity:= ,`
			`lrAcceleration:= ,`
			`lrDeceleration:= ,`
			`eDirection:= ,`
			`lrJerk:= ,`
			`bContinueJog:= ,`
			`bJogFwd:= ,`
			`bJogBwd:= ,`
			`bStop:= ,`
			`tTimeOut:= ,`
			`bDone=> ,`
			`bBusy=> ,`
			`bError=> ,`
			`eErrorID=> );`

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
axisEasyControl	Axis	AXIS_REF_SM3	-	-	Reference to axis.

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
bEnable	Enabling command	BOOL	TRUE, FALSE	FALSE	It is set to TRUE to execute the command.
bPower	Power on	BOOL	TRUE, FALSE	FALSE	It is set to TRUE for power on.
eHomeType	Homing method	MoveCtrl_HomeType	Data range	0	It is used to set the homing method.
bHomeSignal	Homing signal	BOOL	TRUE, FALSE	FALSE	External homing signal.
eRunMode	Run mode	MoveCtrl_Mode	Data range	0	It is used to set the run mode.
bRun	Running	BOOL	TRUE, FALSE	FALSE	The running mode is triggered at the rising edge.

Input variable	Name	Data type	Valid range	Initial value	Description
bReset	Axis reset	BOOL	TRUE, FALSE	FALSE	Axis reset
bReinit	Axis re-initialization	BOOL	TRUE, FALSE	FALSE	It is set to TRUE to re-initialize the axis.
lrPositionOrDistance	Position or distance	LREAL	Data range	0	It is set to the distance or position to move.
lrVelocity	Speed	LREAL	Data range	0	Running speed.
lrAcceleration	Acceleration	LREAL	Data range	0	Acceleration value.
lrDeceleration	Deceleration	LREAL	Data range	0	Deceleration value.
eDirection	Homing direction	MC_DIRECTION	Data range		Homing initial direction
lrJerk	Jump	LREAL	Data range	0	Slope change value of the curve acceleration or deceleration.
bContinueJog	Continuous jogging	BOOL	TRUE, FALSE	FALSE	It is set to TRUE to perform continuous jogging.
bJogFwd	Forward jogging	BOOL	TRUE, FALSE	FALSE	It is set to TRUE to perform forward jogging.
bJogBwd	Reverse jogging	BOOL	TRUE, FALSE	FALSE	It is set to TRUE to perform reverse jogging.
bStop	Axis stop	BOOL	TRUE, FALSE	FALSE	It is set to TRUE to stop the axis.
tTimeOut	Timeout time.	TIME	Data range	30S	When the motion timeout time is exceeded, the axis is stopped.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
bDone	Command execution completion	BOOL	TRUE, FALSE	FALSE	It is set to TRUE after the command is executed completely.
bBusy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the command is being executed.
bError	Fault	BOOL	TRUE,	FALSE	It is set to TRUE when an

Output variable	Name	Data type	Valid range	Initial value	Description
			FALSE		error occurs.
eErrorID	Error ID	SMC_ERROR	-	0	Error ID.

(3) Function description

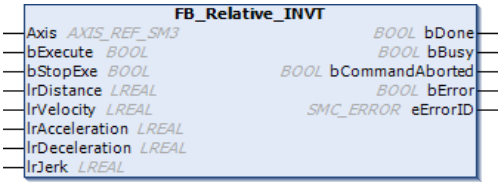
To quickly invoke and use single-axis motion control functions, the function block encapsulates the power-on function, homing function, jogging function, relative motion, absolute motion, specific-speed running, stop, and reset functions of the Codesys axis module. If you want to use single-axis motion control functions, you only need to invoke this function block, improving application program development efficiency.

3.3.3 FB_Relative_INVT

FB_Relative_INVT: relative motion suspension command.

(1) Command format

Command	Name	Expressed in diagram	Expressed in ST

FB_Relative_INVT	Relative motion suspension command		`FB_Relative_INVTt(`
			`Axis:= ,`
			`bExecute:= ,`
			`bStopExe:= ,`
			`lrDistance:= ,`
			`lrVelocity:= ,`
			`lrAcceleration:= ,`
			`lrDeceleration:= ,`
			`lrJerk:= ,`
			`bDone=> ,`
			`bBusy=> ,`
			`bCommandAborted=> ,`
			`bError=> ,`
			`eErrorID=> );`

(2) Related variables

✧ I/O variable

I/O variable	Name	Data type	Valid range	Initial value	Description
Axis	Axis	AXIS_REF_SM3	-	-	Motion axis

✧ Input variable

Input variable	Name	Data type	Valid range	Initial value	Description
bExecute	Enabling	BOOL	TRUE, FALSE	FALSE	It is triggered at the rising edge.
bStopExe	External suspension	BOOL	TRUE, FALSE	FALSE	Set it to TRUE to suspend the running and set it to FALSE to resume the running.
lrDistance	Destination position of relative motion	LREAL	Data range	0	Relative position data of the axis.
lrVelocity	Speed	LREAL	Positive	0	Max. speed at which the axis

Input variable	Name	Data type	Valid range	Initial value	Description
			number or 0		runs to reach the destination position.
lrAcceleration	Acceleration	LREAL	Positive number	0	Acceleration when the speed increases.
lrDeceleration	Deceleration	LREAL	Positive number	0	Deceleration when the speed reduces.
lrJerk	Jump	LREAL	Positive number or 0	0	Slope change value of the curve acceleration or deceleration.

✧ Output variable

Output variable	Name	Data type	Valid range	Initial value	Description
bDone	Command execution completion	BOOL	TRUE, FALSE	FALSE	It is set to TRUE after the command is executed completely.
bBusy	Command being executed	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the command is being executed.
bCommandAborted	Command interruption	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when the axis command is interrupted.
bError	Fault	BOOL	TRUE, FALSE	FALSE	It is set to TRUE when an error occurs.
eErrorID	Error ID	SMC_ERROR	Data range	0	A non-zero value indicates that an exception occurs.

(3) Function description

The function block implements the relative motion function on the Codesys platform, which can be suspended. During axis running, when bStopExe is set to TRUE, the running is suspended; when bStopExe is reset to FALSE again, the axis moves to the position specified by lrDistance.

Service line:86-755-23535967 E-mail:overseas@invt.com.cn Website: invt.com.cn

The products are owned by **Shenzhen INVT Electric Co.,Ltd.**

Two companies are commissioned to manufacture: (For product code, refer to the 2nd/3rd place of S/N on the name plate.)

Shenzhen INVT Electric Co.,Ltd. (origin code: 01)
Address: INVT Guangming Technology Building, Songbai Road, Matian, Guangming District, Shenzhen, China

INVT Power Electronics (Suzhou) Co.,Ltd. (origin code: 06)
Address: No. 1 Kunlun Mountain Road, Science & Technology Town, Gaoxin District, Suzhou, Jiangsu, China

Industrial Automation: ■HMI ■PLC ■VFD ■Servo System ■Elevator Intelligent Control System ■Rail Transit Traction System

Energy & Power: ■UPS ■DCIM ■Solar Inverter ■SVG ■New Energy Vehicle Powertrain System ■New Energy Vehicle Charging System ■New Energy Vehicle Motor

66001-00756

Copyright© INVT.
Manual information may be subject to change without prior notice.

202107 (V1.1)