XINJE

XS series PLCopen controller

User manual [Instruction] (XS Studio)

Wuxi XINJE Electric Co., Ltd.

Data No. PS05 20230109EN 1.0

Basic Description

- Thank you for purchasing the Xinje XS series PLCopen standard controller.
- This manual mainly introduces the relevant usage of XS series PLCopen standard controller instructions.
- Before using the product, please read this manual carefully and program it on the premise of fully understanding the content of the manual.
- Please deliver this manual to the end user.

User Information

- Only operators with a certain level of electrical knowledge can perform wiring and other operations on the product. If there are any areas where the use is unknown, please
- Consult our company's technical department.
- The examples listed in manuals and other technical materials are for user understanding and reference only, and do not guarantee certain actions.
- When using this product in combination with other products, please confirm whether it meets relevant specifications, principles, etc.
- When using this product, please confirm whether it meets the requirements and is safe.
- Please set up backup and safety functions on your own to avoid potential machine malfunctions or losses caused by the malfunction of this product.

Declaration of Responsibility

- Although the content in the manual has been carefully checked, errors are inevitable and we cannot guarantee complete consistency.
- We will regularly check the content in the manual and make corrections in subsequent versions. We welcome your valuable feedback.
- The content described in the manual is subject to change without prior notice.

Related manuals

Please refer to the following manual for hardware related and software applications of the XS series PLC.

- User Manual for XS Series PLCopen Standard Controller [Hardware]
- User Manual for XS Series PLCopen Standard Controller [Software]

Jan. 2023

Catalog

1. BASIC INSTRUCTIONS 1

1-1. BIT LOGIC INSTRUCTIONS 1

1-1-1. Basic bit logic instructions 1

1-1-2. Set priority and reset priority trigger instructions 5

1-1-3. Data unit type 6

1-2. TIMER 7

1-2-1. Pulse timer TP 7

1-2-2. Power on delay timer TON 7

1-2-3. Power off delay timer TOF 8

1-2-4. RTC 8

1-3. COUNTER 9

1-3-1. Up counter CTU 9

1-3-2. Down counter CTD 9

1-3-3. Up/down counter CTUD 10

1-4. DATA PROCESSING INSTRUCTIONS 11

1-4-1. Selection commands 11

1-4-2. Compare Instructions 14

1-4-3. Shift instruction 15

1-5. OPERATION INSTRUCTIONS 21

1-5-1. Assignment instruction 21

1-5-2. Arithmetic operation instructions 21

1-5-3. Mathematical operation instructions 25

1-5-4. Address operation instruction 30

1-5-5. Data conversion instructions 33

2. SPECIAL INSTRUCTIONS 40

2-1. XSA SERIES HIGH SPEED COUNT INSTRUCTIONS 40

2-1-1. Function overview 40

2-1-2. Function block 40

2-1-3. Parameter settings 54

2-2. XS SERIES HIGH SPEED COUNT INSTRUCTIONS 56

2-2-1. Function overview 56

2-2-2. Function block 56

2-2-3. Parameter configuration 60

2-2-4. Application example 61

2-3. EXTERNAL INTERRUPT AND COMPARE CONSISTENT INTERRUPT INSTRUCTIONS 62

2-3-1. Function overview 62

2-3-2. Function block 62

2-3-3. Parameter configuration 66

2-3-4. Application example 67

2-4. PID INSTRUCTIONS 69

2-4-1. Command format 69

2-4-2. Related variables 69

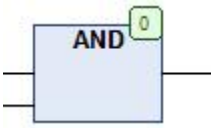
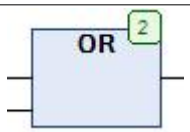
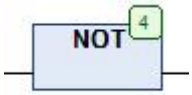
2-4-3. Function description 70

2-4-4. Application example *72*
2-5. SYSTEM LIBRARY 79
2-5-1. Function overview *79*
2-5-2. Function block introduction *79*
2-5-3. Parameter configuration *89*
2-6. ECAT_FROMTO 91
2-6-1. Function overview *91*
2-6-2. Function block introduction *91*
2-6-3. Parameter configuration *93*
2-6-4. Application *95*
3. MOTION INSTRUCTIONS 96
3-1. SINGLE AXIS 96
3-1-1. Single axis instruction overview *96*
3-1-2. Single axis instructions *97*
3-1-3. Single axis function application *144*
3-2. AXIS GROUP FUNCTION 145
3-2-1. Axis group instruction *145*
3-2-2. Axis group instructions *146*
3-2-3. Axis group function application *199*
3-3. CAM FUNCTION 201
3-3-1. CAM instruction list *201*
3-3-2. CAM instructions *202*
3-3-3. CAM function application *229*
APPENDIX 232

1. Basic instructions

1-1. Bit logic instructions

1-1-1. Basic bit logic instructions

Instruction	Instruction icon	Function
AND		Operator AND
OR		Operator OR
NOT		Operator NOT
XOR		Operator XOR

Basic bit logic instructions include AND, OR, NOT, and XOR. In XS Studio, functions can be divided into bit logic operation and Boolean logic operation.

- Bit logic operation: perform Boolean logic operation on corresponding bits of two integer data one by one, and return compatible integer results.
- Boolean logic operation: perform logical operation on two Boolean type data.

1-1-1-1. Bit AND

Function: The bit AND operation instruction compares the corresponding bits of two integers. When the corresponding bits of two numbers are both 1, the corresponding result bit returned is "1". When the corresponding bits of two integers are both "0" or one of the bits is "0", the corresponding result bit is returned as "0".

AND logic relationship table

Input 1	Input 2	Result
0	0	0
0	1	0
1	0	0
1	1	1

Example: Create a POU and declare two integers

The variables iVar1 and iVar2 are assigned values of 1 and 85, and bit AND operations are performed on these two variables to output the results to iResult. The specific implementation code is as follows:

```
VAR
iVar1:INT:=1;
iVar2:INT:=85;
iResult:INT;
```

```
END_VAR
```

```
iResult:=iVar1 and iVar2;
```

The decimal number 1 corresponds to a binary number of 0000 0001, and the decimal number 85 corresponds to a binary number of 0101 0101. According to the definition of bit AND operation, each independent bit is subjected to an AND operation one by one, resulting in the final result of 0000 0001, which is a decimal value of 1.

1-1-1-2. Boolean AND

Function: Boolean AND operation is used to calculate the AND results of two Boolean expression. When the result of two Boolean expression is true, it is returned as true. If one of them is false, it is returned as false.
Example: Create a POU and use the Boolean AND operation to determine the return value of the operation. The code is as follows:

```
VAR
   B1:BOOL;
   B2:BOOL;
   B3:BOOL;
END_VAR
```

```
B3:=B1 AND B2;
```

When both B1 and B2 are true, the running result of the program is B3 to be true.

1-1-1-3. Bit OR

Function: The bit OR operation instruction compares the corresponding bits of two integers. When one of the corresponding bits of two numbers is "1" or both are "1", the corresponding result bit is returned as "1". When the corresponding bits of two integers are both "0", the corresponding result is returned as "0".

OR logic relationship table

Input 1	Input 2	Result
0	0	0
0	1	1
1	0	1
1	1	1

Example: Create a POU, perform bit OR on variables iVar1 and iVar2, and perform bit OR on these two variables to output the results to iResult. The specific implementation code is as follows:

```
VAR
iVar1:INT:=1;
iVar2:INT:=85;
iResult:INT;
END_VAR
```

```
iResult:=iVar1 OR iVar2;
```

The final running result of the program is 85.

1-1-1-4. Boolean OR

Function: Boolean OR operation instruction is used to calculate the OR result of two Boolean expression. When one of the two Boolean expression returns true, the result is true. When the result of two Boolean expression is false, the result is false.
Example: Create a POU and use the Boolean OR operation to determine the return value of the operation. The code is as follows:

```
VAR
bResult:BOOL;
bVar1:BOOL;
iVar1:INT:=30;
END_VAR
```

```
bResult:=bVar1 OR (iVar1 <80);
```

Due to the initial value of iVar1 being 30, the condition of iVar1<80 is true, while the initial value of bVar1 is "0", so it is false. The logical result of one true and one false OR is true. Therefore, the condition on the right side of the equation is true, and the running result of the program is bResult being true.

1-1-1-5. Bit NOT

Function: Negate the logical string, changing the current value from "0" to "1" or from "1" to "0". The bit 'not' operation instruction is to take variables or constants one by one as not.

NOT logic relationship table

Input	Result
0	1
1	0

Example: Create a POU and use the bit NOT operation to determine the return value of the operation. The specific code is as follows:

```
VAR
byVar1:BYTE:=1;
byVar2:BYTE;
END_VAR
```

```
byVar2:=NOT byVar1;
```

Due to the value of byVar1 being 1, converting it to binary yields 0000 0001. After performing bit negation, the result is 1111 1110. The final output result is 254.

1-1-1-6. Boolean NOT

Function: Boolean "Not" operation instruction is used to calculate the result of a single Boolean expression. When the input is true, the result is false. When the input is false, the result is true.

Example: Create a POU and use the Boolean 'not' operation to determine the return value of the operation. The specific code is as follows:

```
VAR
bResult:BOOL;
bVar1:BOOL;
iVar1:INT:=30;
END_VAR
```

```
bResult:=NOT ( 80 < iVar1 );
```

80<30 this proposition is false. After using the NOT instruction to negate the Boolean expression, the result is true, so the final bResult is true.

1-1-1-7. Bit XOR

Function: Compare the corresponding bits of two integers using the bit XOR operation instruction. When the corresponding bit of two integers is one "1" and the other is "0", the corresponding result bit returned is "1". When the corresponding bits of two integers are both "1" or "0", the corresponding result bit is returned as "0".
Example: Create a POU, perform bit XOR on variables iVar1 and iVar2, and output the results. The specific implementation code is as follows:

```
VAR
iVar1:INT:=1;
iVar2:INT:=85;
iResult:INT;
END_VAR
```

```
iResult:=iVar1 XOR iVar2;
```

The decimal number 1 corresponds to a binary number of 0000 0001, and the decimal number 85 corresponds to a binary number of 0101 0101. According to the definition of the bit XOR operation instruction, the result is 84.
Bit XOR operation. Only when the input state of one contact is "1" and the input state of the other contact is "0", the output is "1". If both contact states are "1" or "0" at the same time, the output is "0".

1-1-1-8. Boolean XOR

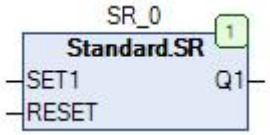
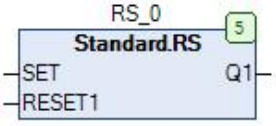
Function: Boolean XOR operation instruction is used to calculate the results of two Boolean expression. Only when one expression is true and the other is false, the result returned by the expression is true. When both expressions evaluate to true or false, the returned result is false.
Example: Create a POU and use the Boolean XOR operation instruction to determine whether the return value is TRUE or FALSE. The specific implementation code is as follows:

```
VAR
bResult:BOOL;
bVar1:BOOL;
iVar1:INT:=30;
END_VAR
```

```
bResult:=bVar1 XOR (iVar1 <80);
```

The running result of the program is TRUE.

1-1-2. Set priority and reset priority trigger instructions

Instruction	Instruction icon	Function
SR		Set priority trigger: Set bistable trigger, set priority
RS		Reset priority trigger: Reset bistable trigger, reset priority

In a relay system, several pairs of contacts of a relay act simultaneously. In PLC, instructions are executed one by one, and the execution of instructions is sequential, without any "simultaneous" instructions. So the setting and reset commands for coil format have priority. The set and reset inputs of SR trigger and RS trigger are in the same instruction, and the set and reset inputs are executed by the one below the instruction input. The SR trigger is a "set priority" type trigger. When the set signal (SET1) and reset signal (RESET) are both 1, the trigger ultimately enters the set state. The RS trigger is a "reset priority" type trigger. When the set signal (SET) and reset signal (RESET1) are both 1, the trigger ultimately enters the reset state.

1-1-2-1. Set priority trigger SR

Function: Set bistable trigger, with set priority. Logical relationship: Q1=(NOT RESET AND Q1) OR SET1, where SET1 is the set signal and RESET is the reset signal.
Syntax: When SET1 is "1", regardless of whether RESET is "1" or not, Q1 output is "1". When SET1 is "0", if Q1 output is "1", once RESET is "1", Q1 output will immediately reset to "0". If Q1 output is "0", regardless of whether RESET is "1" or "0", Q1 output remains "0".

SR status table

SET1	RESET	Q1 output
0	0	Maintain the original state
1	0	1
0	1	0
1	1	1

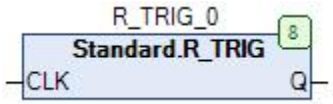
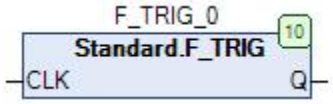
1-1-2-2. Reset priority trigger RS

Function: Reset the bistable trigger, with reset priority. Logical relationship: Q1=NOT RESET1 AND (Q1 OR SET), where SET is the set signal and RESET1 is the reset signal.
Syntax: When RESET1 is "1", regardless of whether SET is "1" or not, Q1 output is "0". When RESET1 is "0", if Q1 output is "0", once SET is "1", Q1 output is immediately set to "1". If Q1 output is "1", regardless of whether SET is "1" or "0", Q1 output remains "1".

RS status table

SET	RESET1	Q1 output
0	0	Maintain the original state
1	0	1
0	1	0
1	1	0

1-1-3. Data unit type

Instruction	Instruction icon	Function
R_TRIG		Rising edge trigger
F_TRIG		Falling edge trigger

The edge detection instruction is used to detect changes in the rising edge (signal from 0-->1) and falling edge (signal from 1-->0) of the BOOL signal. In each scanning cycle, the signal state is compared with its state in the previous scanning cycle. If different, it indicates a jump edge. Therefore, the signal state from the previous cycle must be stored in order to be compared with the new signal state.

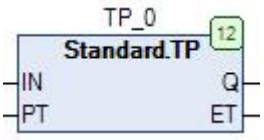
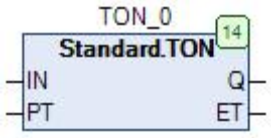
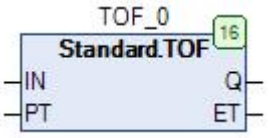
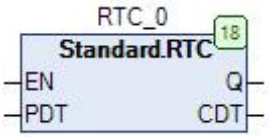
1-1-3-1. Rising edge detection R_TRIG

Function: Used to detect the rising edge.
Syntax: When CLK changes from "0" to "1", the rising edge detector starts, and the Q output first changes from "1" to "0", lasting for one PLC operation cycle. If CLK remains at "1" or "0" continuously, Q output remains at "0".

1-1-3-2. Falling edge detection F_TRIG

Function: Used to detect the falling edge.
Syntax: When CLK changes from "1" to "0", the falling edge detector starts, and the Q output first changes from "1" to "0", lasting for one PLC operation cycle. If CLK remains at "1" or "0" continuously, Q output remains at "0".

1-2. Timer

Command	Command icon	Function
TP		Pulse Timer: Once IN becomes true, then Q becomes true, and time will start counting in milliseconds in ET until its value equals PT, then Q is FALSE
TON		Power on delay timer: Once IN becomes TRUE, the time will start counting in milliseconds in ET until its value equals PT, then Q is TRUE
TOF		Power off delay timer: When IN changes from TRUE to FALSE and ET is equal to PT, Q is FALSE, otherwise it is TRUE
RTC		Real-time clock: start at a given time and return the date and time

1-2-1. Pulse timer TP

Function: Pulse timing.
Syntax: When the input IN of the timer changes from "0" to "1", the timer starts. Regardless of how the input IN of the timer changes, the actual running time of the timer is the user-defined PT time. When the timer is running, the output signal of its output Q is "1". The output terminal ET provides a timing time for the output terminal Q. The timing starts from T#0s and ends at the set PT time. When the PT time expires, ET will maintain a timed time until IN becomes "0". If the input IN has already changed to "0" before reaching the PT timing time, input ET changes to T#0s, the timing of the PT. To reset the timer, simply set PT=T#0s.

1-2-2. Power on delay timer TON

Function: Power on delay timing.
Syntax: When the input IN of the timer changes from "0" to "1", the timer starts. When the timing time PT is reached and the input signal IN is always maintained at "1", the output signal of the output Q is "1". If the input IN signal changes from "1" to "0" before the timer's timing time is reached, the timer resets, and timer restarts at the rising edge of the next IN signal. The output end ET provides a timing time, with a delay starting from T#0s and ending at the set PT time. When PT arrives, ET will maintain a timed time until IN becomes "0". If the input IN becomes "0" before reaching the PT timing time, the output ET immediately becomes T#0s. To restart the timer, you can set PT=T#0s or IN=FALSE.

1-2-3. Power off delay timer TOF

Function: Power off delay timing.
Syntax: When the input IN of the timer changes from "0" to "1", the Q output signal of the timer is "1". When the start input of the timer changes to "0", the timer starts. As long as the timer is running, its output Q remains "1". When the timing time is reached, the output Q is reset. Before the timing time is reached, if the input of the timer returns to "1", the timer is reset, The Q output signal at the output end remains at "1". The output end ET provides a timing time, with a delay starting from T#0s and ending at the set timing time PT. When the PT time expires, ET will maintain a timed time until the input IN returns to "1". If the input IN becomes "1" before reaching the PT timing time, the output ET immediately becomes T#0s. To reset the timer, it is possible to set PT=T#0s.

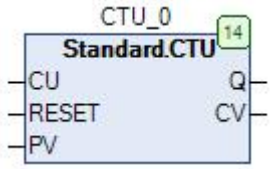
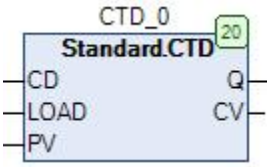
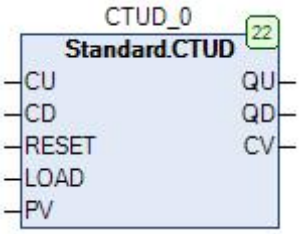
1-2-4. RTC

Function: Start at a given time and return the current date and time.
Syntax: RTC (EN, PDT, Q, CDT) means that when EN is "0", the output variable Q and CDT are "0", and the relevant time is DT#1970-01-01-00:00:00. Once EN is "1", the time given by PDT will be set and counted in seconds. Once EN is reset to FALSE, CDT will be reset to the initial value DT#1970-01-01-00:00:00. Please note that the PDT time is only valid on the rising edge. The RTC timer parameter table is as follows:

Standard Timer Command Parameter Table

Name	Definition	Data type	Explanation
EN	Input variables	BOOL	Start enable
PDT	Input variables	DATE_AND_TIME	Set the time and date to start
Q	Output variables	BOOL	Status output
CDT	Output variables	DATE_AND_TIME	Current count time and date status

1-3. Counter

Command name	Command icon	Function
CTU		Up counter: If RESET is TRUE, initialize to 0. When the rising edge of CU is always increased by 1, once CV>=PV, Q will be set to TRUE
CTD		Down counter: If LOAD is TURE, CV will be set to the starting value given by PV. Then set the load to FALSE manually, and the CV value will decrease by 1 every time the rising edge of the CD is approached. If the CV value decreases to 0, Q will be set to TRUE
CTUD		Up/down counter

1-3-1. Up counter CTU

When the signal at the input end of the counter CU changes from state "0" to state "1", the current calculated value is increased by 1 and displayed through the output end CV. On the first call (resetting the input RESET signal state to "0"), the count at the input PV end is the default value. When the count reaches the upper limit of 32767, the counter will not increase again and the CU will no longer function.

When the signal status of the reset input RESET is "1", the CV and Q of the counter are both "0". As long as the input RESET status is "1", the rising edge will no longer have any effect on the CU. When the CV value is greater than or equal to PV, the output Q is "1". At this point, CV can still continue to accumulate, and output Q continues to be output "1".

Incremental function block. The input variable CU, RESET, and output variable Q are of Boolean type, while the input variable PV and output variable CV are WORD type. CV will be initialized to 0 if RESET is reset to true. If CU has a rising edge that changes from FALSE to TRUE and CV increases by 1, Q will return TRUE, so CV will be greater than or equal to the upper limit PV.

1-3-2. Down counter CTD

When the CD signal at the input end of the counter is changed from "0" to state "1", the current count value is reduced by 1, and the CV displays the current value on the output end. When the first call is made (the load input signal needs to be initialized, and it needs to be changed from "0" to state "1" and then to state "0" to make the function block effective), the count at the input PV end is the default value. When the count reaches 0, the count value will no longer decrease, CD no longer work.

When the load input signal LOAD is "1", the count value will be set to the PV default value. As long as the load

input signal LOAD status is "1", the CD rising edge of the input end will not work. When the CV value is less than or equal to 0, the output Q is "1".

1-3-3. Up/down counter CTUD

When the CU signal at the counting input end changes from "0" to state "1", the current counting value is incremented by 1 and displayed on the output CV.
When the signal status of the CD at the subtract count input changes from "0" to "1", the current count value is subtracted by 1 and displayed on the output CV. If both inputs have rising edges, the current count value will remain unchanged.
When the counting value reaches the upper limit value 32767, the rising edge of the counting input CU no longer works. Therefore, even if there is a rising edge on the counting input CU, its value will not increase. Similarly, when the count value reaches the lower limit value of 0, the subtracting input CD will not be effective. Therefore, even if the subtracting input CD shows a rising edge, the count value will not decrease.
When the CV value is greater than or equal to the PV value, the output QU is "1". When the CV value is less than or equal to 0, the output QD is "1".

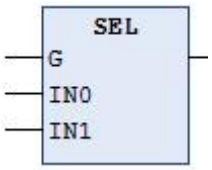
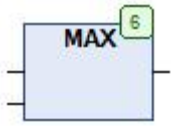
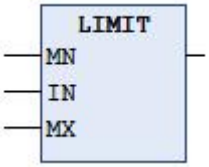
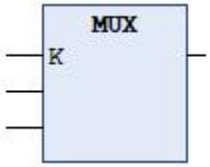
Example: To create a POU, use the up/down counter CTUD. When bUp has a rising edge signal, the count value increases, and when bDown has a rising edge signal, the count value decreases. bReset is used for data reset, and the specific code is as follows:

```
VAR
bUp: BOOL;
bDown: BOOL;
bReset: BOOL;
bLoad: BOOL;
CTUD_0: CTUD;
END_VAR
```

```
CTUD_0(
CU:= bUp,
CD:= bDown,
RESET:=bReset ,
LOAD:= bLoad,
PV:= ,QU=> ,QD=> ,CV=> );
```

1-4. Data Processing Instructions

1-4-1. Selection commands

Command	Command icon	Function
SEL		Binary choice: When the selection switch is FALSE, the output is the first input data, and when the selection switch is TRUE, the output is the second data
MAX		Take the maximum value
MIN		Take the minimum value
LIMIT		Limit value: If the IN value is higher than the Max upper limit, the result is Max. If the value of IN is lower than the lower limit of Min, the result is Min
MUX		Multiple Choice: MUX selects the Kth value from a set of values. The first value is K=0. If K is greater than the number of other inputs (n), XS Studio passes the last value

1-4-1-1. Binary choice instruction SEL

Function: select one of the two input data as the output through the selection switch. When the switch is FALSE, the output is the first input data, and when the switch is TRUE, the output is the second data.

Grammar: Its textual language syntax format is as follows:

```
OUT := SEL(G, IN0, IN1)
```

The parameter G must be a Boolean variable. If G is FALSE, the result of the returned value is IN0. If G is TRUE, the result of the returned value is IN1. The parameter description is detailed in the table below:

Binary choice instruction SEL

Name	Definition	Data type	Explanation
G	Input variables	BOOL	Input selection bit
IN0	Input variables	Any type	Input data 0
IN1	Input variables	Any type	Input data 1
Return value	Output variables	Any type	Output data

Example: To create a POU, when the input value bInput is FALSE, the output is 3, and vice versa, when it is TRUE, the output is 4. The specific implementation program is as follows:

```
VAR
iVar1:INT:=3;
```

```
iVar2:INT:=4;
iOutVar: INT;
bInput: BOOL;
END_VAR
```

```
iOutVar:=SEL(bInput,iVar1,iVar2);
```

After running the program, the output result is 3.

1-4-1-2. Take the maximum value MAX

Function: Maximum function. Select the maximum value as the output from multiple input data.

Grammar: The textual language syntax format is as follows:

OUT := MAX(IN0, …,INn)

IN0, INn, and OUT can be any data type, and their parameter descriptions are detailed in the following table:

Take the maximum value MAX

Name	Definition	Data type	Explanation
IN0	Input variables	Any type	Input data 0
INn	Input variables	Any type	Input data 1
Return value	Output variables	Any type	Output data

Create a POU, the input value of iOutVar being the larger one of iVar1 and iVar2. The specific implementation program is as follows:

```
VAR
iVar1:INT:=30;
iVar2:INT:=60;
iOutVar: INT;
END_VAR
```

```
iOutVar:=MAX(iVar1,iVar2);
```

After running the program, the output result is 60.

1-4-1-3. Take the minimum value MIN

Function: Minimum value function. Select the minimum value as the output from multiple input data.

Grammar: The textual language syntax format is as follows:

OUT := MIN(IN0, …,INn)

IN0, INn, and OUT can be any data type, and their parameter descriptions are detailed in the following table:

Take the minimum value MIN

Name	Definition	Data type	Explanation
IN0	Input variables	Any type	Input data 0
INn	Input variables	Any type	Input data 1
Return value	Output variables	Any type	Output data

Example: To create a POU, the input value of iOutVar is the smaller one of iVar1 and iVar2. The specific implementation program is as follows:

```
VAR
iVar1:INT:=30;
iVar2:INT:=60;
iOutVar: INT;
END_VAR
```

```
iOutVar:=MIN(iVar1,iVar2);
```

After running the program, the output result is 30.

1-4-1-4. Limit value LIMIT

Function: Limit value output. Determine whether the input data is between the minimum and maximum values. If the input data is between the two, directly output the input data as output data. If the input data is greater than the maximum value, the maximum value is taken as the output value. If the input data is less than the minimum value, the minimum value is used as the output value.
Grammar: Its textual language syntax format is as follows:,

```
OUT := LIMIT(Min, IN, Max)
```

IN, Min, Max, and return values can be any data type, and their parameter descriptions are detailed in the table below:

Limit value LIMIT

Name	Definition	Data type	Explanation
Min	Input variables	BOOL	Input data 0
IN	Input variables	Any type	Input data n
Max	Input variables	Any type	Input data n
Return value	Output variables	Any type	Output data

Example: Create a POU, use the limit value instruction to ensure that the output value is within the range of 30-80 regardless of the input value. The specific implementation program is as follows:

```
VAR
iVar:INT:=90;
iOutVar: INT;
END_VAR
```

```
iOutVar:=limit(30,iVar,80);
```

The minimum input value is 30, the maximum input value is 80, and the actual input value is 90, which is greater than the maximum value. Therefore, the final output is based on the maximum value of 80, so the final result is 80.

1-4-1-5. Multiple choice MUX

Function: Multiplexer operation. Select one of multiple input data as the output through the control value.
Grammar: Its textual language syntax format is as follows:

```
OUT := MUX(K, IN0,...,INn)
```

IN0,…, INn and the return value can be any variable type. But K must be BYTE, WORD, DWORD, LWORD, SINT, USINT, INT, UINT, DINT, LINT, ULINT, or UDINT. MUX selects the Kth data output from the variable group. The parameter description is detailed in the table below:

Multiple choice MUX

Name	Definition	Data type	Explanation
K	Input variables	Integer type	Control value
IN0	Input variables	Any type	Input data 0
INn	Input variables	Any type	Input data n
Return value	Output variables	Any type	Output data

Example: To create a POU, use multiple choice instruction to select the final output data based on the input control value iVar. The specific implementation program is as follows:

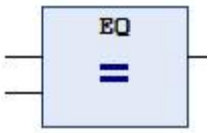
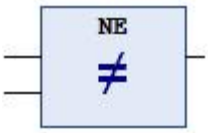
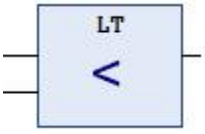
```
VAR
iVar:INT:=1;
iOutVar: INT;
END_VAR
```

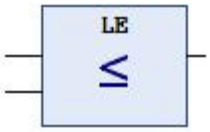
```
iOutVar:=MUX(iVar,30,40,50,60,70,80);
```

The final output result is 40, as the data sorting starts from the 0th element and accumulates.

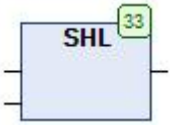
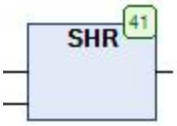
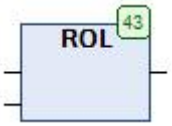
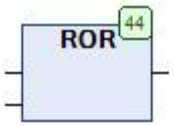
If the data exceeds the range, the final data will be output based on the last data. In the example, setting the value of iVar to 10 will result in a final output of 80. If iVar is -1, the final output value is still 80.

1-4-2. Compare Instructions

Command name	Command icon	Function
EQ		Equal to
NE		Not equal to
GT		Greater than
GE		Greater than or equal to
LT		Smaller than

Command name	Command icon	Function
LE		Smaller than or equal to

1-4-3. Shift instruction

Command name	Command icon	Function
SHL		Shift left
SHR		Shift right
ROL		Rotate left
ROR		Rotate right

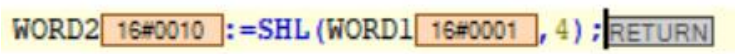
1-4-3-1. Shift left SHL

Function: Shift the operand to the left bit by bit, without processing the left out bit, and automatically fill in the right empty bit with 0.

Syntax: The instruction can shift the data in input IN by n bits to the left, and assign the output result to OUT. Shifting a binary number by one bit to the left is equivalent to multiplying the original number by 2. If n is greater than the width of the data type, BYTE, Word, and DWORD values will be filled to zero. The syntax format of textual language is as follows:

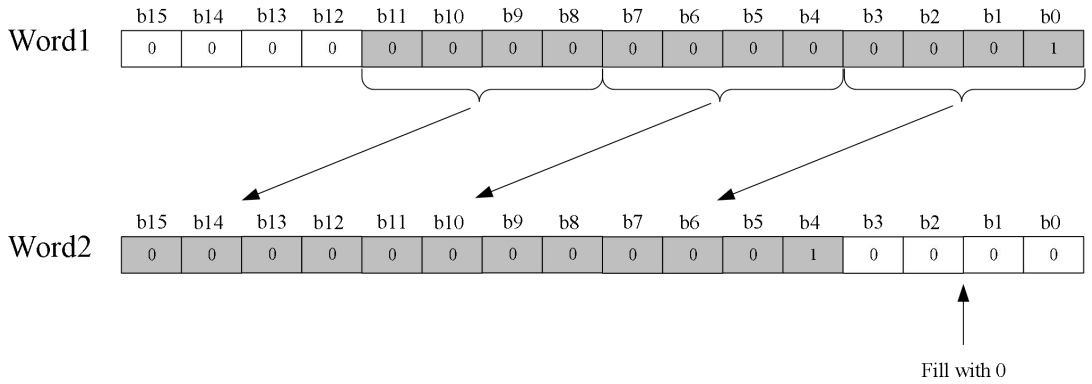
```
OUT:= SHL (IN, n)
```

For example, the shift left instruction is used to shift the current value of WORD1(an WORD type input variable) to the left by 4 bits, , and the output result is assigned to Word2.



Shift left program example

As mentioned above, Word1 is 0001 in hexadecimal, and after moving it 4 bits to the left, the final output result is 16#0010. The process is shown in the following figure. Fill in the empty space of the lower 4 bits with 0.



The process of shift left by 4 bits

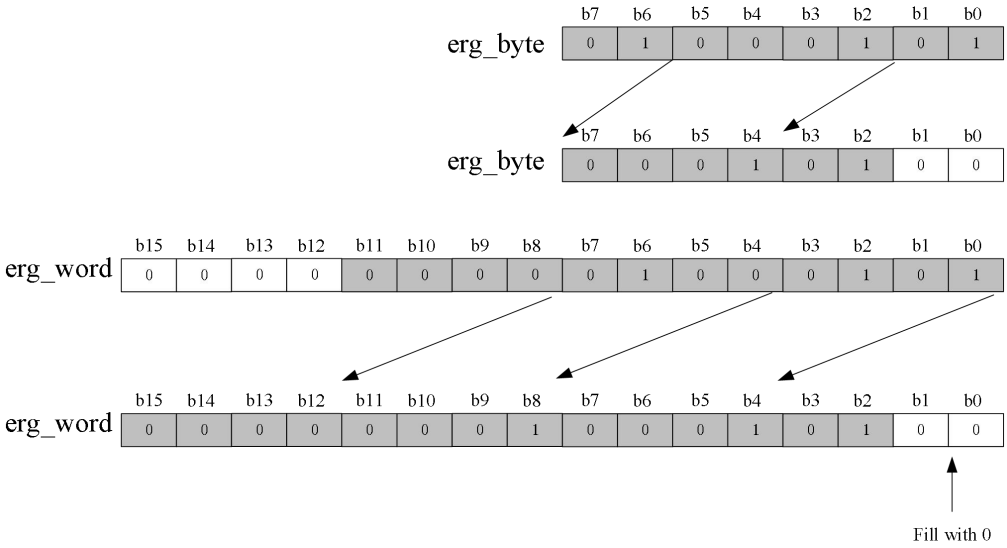
The total number of bits in a shift operation is influenced by the data type of the input variable. If the input variable is a constant, the data type with the smallest length will be taken. The data type of the output variable does not affect the arithmetic operation, and the difference between the two is identified through the following example.

Please compare the bit left shift operation of hexadecimal numbers below. Although the input variable values in byte and word form are equal, depending on the data type of the input variable (BYTE or WORD), erg_byte and erg_word will yield different results.

```
VAR
in_byte : BYTE:=16#45;
in_word : WORD:=16#45;
erg_byte : BYTE;
erg_word : WORD;
n: BYTE :=2;
END_VAR
```

```
erg_byte:=SHL(in_byte,n); (* result is 16#14 *)
erg_word:=SHL(in_word;n); (* result is 16#0114 *)
```

When the BYTE type variables b6 and b7 shift left by 2 bits and overflow, the final data is hexadecimal 14. When the b6 and b7 bits of the WORD type variable are shifted to the left by two bits and enter the high byte b8 and b9 bits, this bit will continue to be retained, and the final result is a hexadecimal value of 114. The process is shown in the following figure:



BYTE and WORD variables shifted left by bit

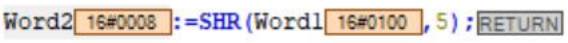
1-4-3-2. Shift right SHR

Function: Shift the operand bit by bit to the right, without processing the right out bit, and automatically fill in the left empty bit with 0.

Syntax: The instruction can shift the data in input IN by n bits to the right, assign the output result to OUT, and shift the binary number by one bit to the right is equivalent to dividing the original number by 2. If n is greater than the width of the data type, BYTE, Word, and DWORD values will be filled in as zero. If a signed data type is used, the arithmetic shift will be supplemented by the highest order. The textual language syntax format is as follows:

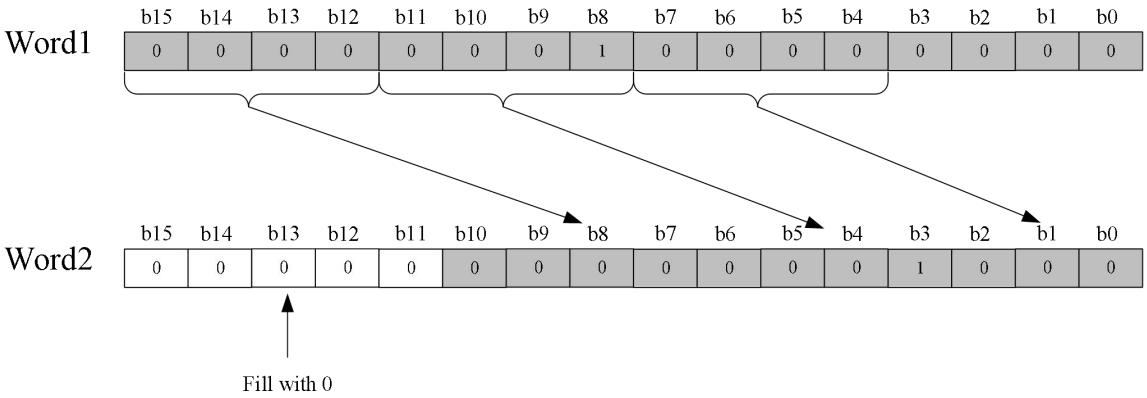
```
OUT:= SHR (IN, n)
```

Example: Use the bit by bit right shift instruction to shift the current value Word1(WORD type input variable) to the right by 5 bits, and assign the output result to Word2.



Shift right by bit example

Word1 is a hexadecimal value of 0100, and after moving 5 bits to the right, the final output result is 16#0008. Due to the fact that WORD type variables belong to unsigned data types and have valid values ranging from 0 to 65535, after shifting 5 bits to the right, there is no sign bit, and the high 5 bits are filled with 0. The shift process is shown in the following figure:



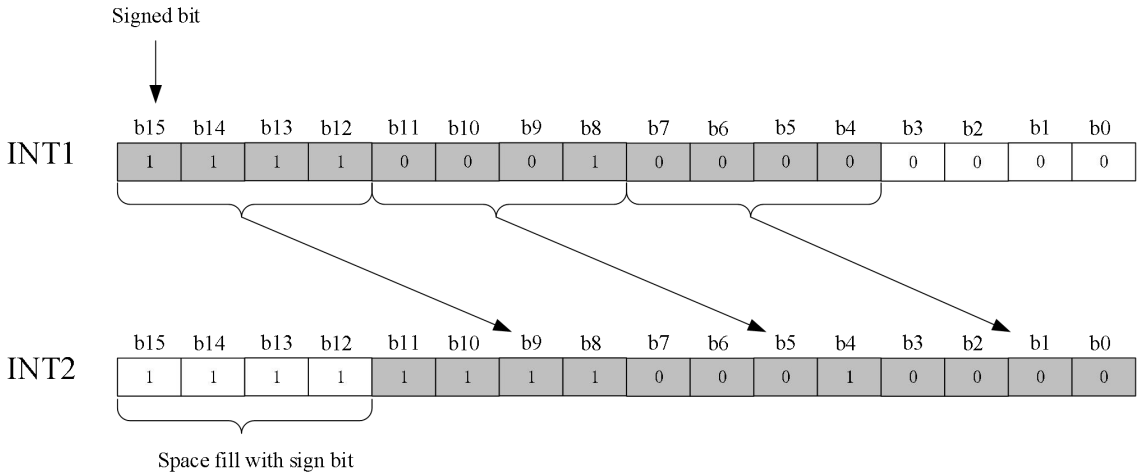
The process of shift right by 5 bits

The previous example was the right shift of unsigned bit data. If encountering signed integer data, the high bit right shift requires filling in the signed bit. As shown in the following example:
For example, the current value of the INT type input variable iINT1 is shifted to the right by 4 bits, and the output result is assigned to iINT2.

```
INT2 16#FF10 :=SHR(INT1 16#F100 ,4);RETURN
```

Signed integer data shift right example

As mentioned above, since INT is signed bit data with valid values ranging from -32768 to 32767, iINT1 is hexadecimal signed data F100, and the highest bit b15 is the sign bit. After moving 4 bits to the right, data needs to be supplemented. Due to the sign bit of the source data being 1, the high 4 bits are supplemented with 1 (i.e. sign bits). Therefore, the final result of program operation is 16#FF10. The specific shift process is shown in the following figure:



1-4-3-3. Rotate left ROL

Function: Rotate the operands to the left, and the bits moved from the left are directly added to the lowest bit on the right.
Syntax: Allowed data types: BYTE, WORD, DWORD. This instruction can be used to rotate all the contents of the input IN bit by bit to the left, and the vacated bits are filled with the signal state of the shifted bits. The input parameter n provides a numerical value representing the number of bits to rotate, and OUT is the result of the rotation operation. The textual language syntax format is as follows:

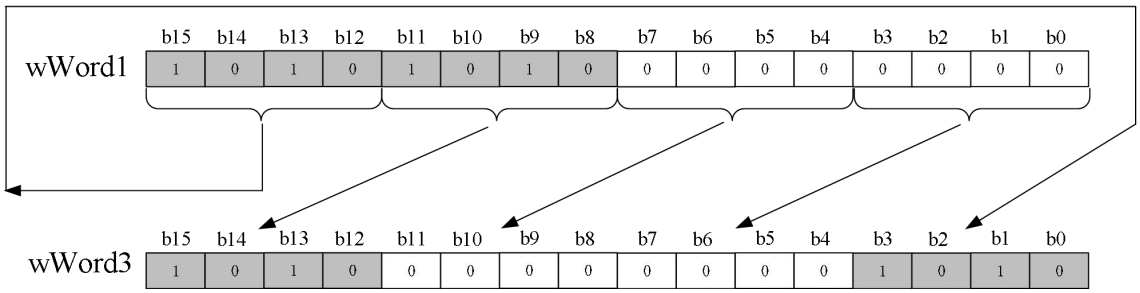
```
OUT:= ROL (IN, n)
```

Example: Create a POU and compare the differences between bitwise left shift and cyclic left shift. Move the hexadecimal WORD type variable Word1 by the same number of digits using two different left shift methods, and compare the results.

```
Word2 16#A000 :=SHL(Word1 16#AA00 ,4);
Word3 16#A00A :=ROL(Word1 16#AA00 ,4);
```

Bitwise left shift and rotate left shift comparison program

As shown in the following figure, after using the rotated right, the output of Word3's b0 to b3 bits does not fill with 0, but rather fills in the 1010 of b12 to b15 in the input data Word1 to b0 to b3 bits.



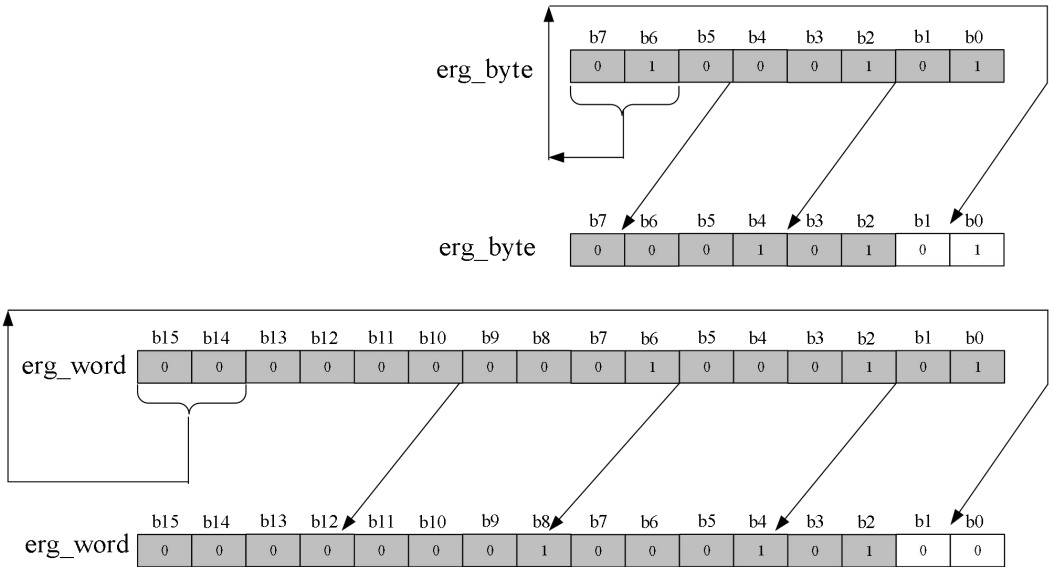
The process of rotate left shift by 4 bits

The total number of bits in a rotate shift instruction is also affected by the data type of the input variable. If the input variable is a constant, the data type with the smallest length will be taken. The data type of the output variable does not affect the arithmetic operation, and the difference between the two is identified through the following example.

Example: please compare the rotate left shift operation of hexadecimal numbers below. Although the input variable values in byte and word form are equal, depending on the data type of the input variable (BYTE or WORD), erg_byte and erg_word will yield different results.

```
VAR
in_byte: BYTE:=16#45;
in_word: WORD:=16#45;
erg_byte : BYTE;
erg_word : WORD;
n: BYTE :=2;
END_VAR
```

```
erg_byte:=ROL(in_byte,n); (* result is 16#15 *)
erg_word:=ROL(in_word,n); (* result is 16#0114 *)
```



BYTE and WORD variable rotate left shift comparison

As shown in the figure, when the b6 and b7 bits of the BYTE type variable are moved to the left by 2 bits and then moved to the b0 and b1 bits in the output data, the final data is hexadecimal 15. When the b6 and b7 bits of the WORD type variable are shifted to the left by two bits, they are moved to bits b8 and b9 in the output data. The b14 and b15 bits of the original data are 0, and after being shifted to the left, they are moved to bits b0 and b1 in the output data. Therefore, the final result is a hexadecimal 114.

1-4-3-4. Rotate right ROR

Function: Rotate the operands bit by bit to the right, and the bits moved out from the right are directly added to the highest position on the left.

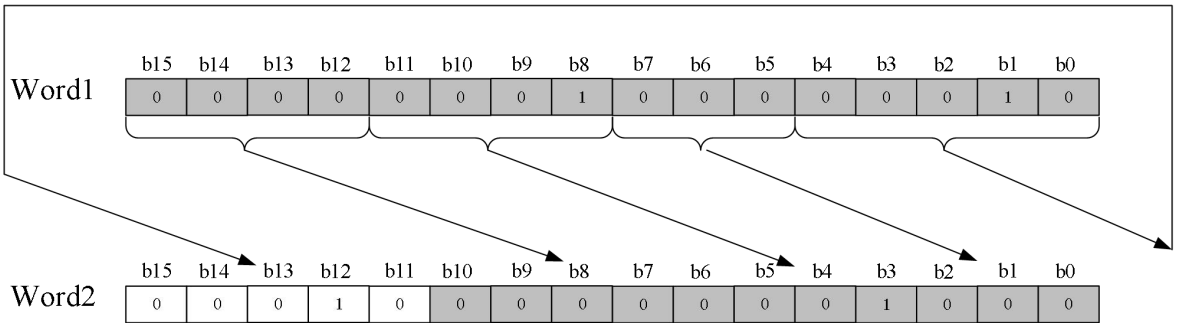
Syntax: Allowed data types: BYTE, WORD, DWORD. Using this instruction, all contents in the input IN can be rotated bit by bit to the right, and the vacated bits are filled with the signal state of the shifted bits. The input parameter n provides a numerical value to represent the number of digits to rotate, and OUT is the result of the rotation operation. The textual language syntax format is as follows:

```
OUT: = ROR (IN, n)
```

For example, the current value of the WORD type input variable Word1 is rotated 5 bits to the right by using the rotate right shift instruction, and the output result is assigned to Word2.

```
Word2 16#1008 :=ROR(Word1 16#0102 ,5);
```

Rotate right program example

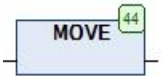


Rotate right by 5 bits process

The final running result of the program is 1008 in hexadecimal, and the program moves the lower 5 bits b0~b4 of Word1 to b11~b15 in Word2.

1-5. Operation instructions

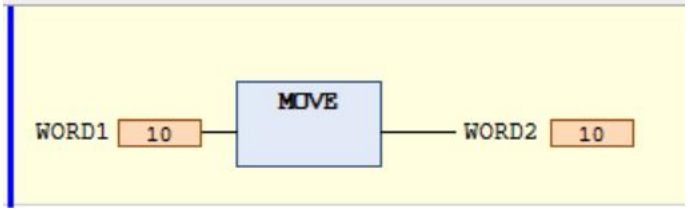
1-5-1. Assignment instruction

Command	Icon	Function
MOVE		Assignment

1-5-1-1. Assignment instruction MOVE

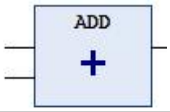
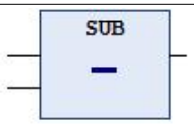
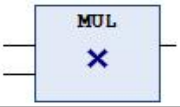
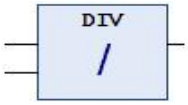
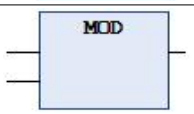
Function: Assign the value of a constant or variable to another variable.

Example: Create a POU and assign the data from the WORD variable WORD1 to WORD2. The specific implementation program is as follows:



MOVE instruction example

1-5-2. Arithmetic operation instructions

Command	Icon	Function
ADD		Addition
SUB		Subtraction
MUL		Multiplication
DIV		Division
MOD		Remainder

1-5-2-1. Addition operation ADD

Function: Addition operation instruction, adding two (or more) variables or constants. Two time variables can also be added, resulting in another time variable.
Syntax: The instruction can add the value of the input variable IN0 to the value of INn, and assign the result to OUT. The addition operation instruction supports the following variable types: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, (L) REAL, TIME, and constants. The textual language syntax format is as follows:

```
OUT := IN0 +…+INn
```

Example 1: Create a POU, declare two integer variables iVar1 and iVar2, assign iVar1 to 2014, and then make the value of iVar2 the sum of iVar1 and iVar1. The specific code is as follows:

```
VAR
iVar1:INT：=2014;
iVar2:INT;
END_VAR
```

```
iVar2:=iVar1+iVar1;
```

The running result of the program is iVar2 equal to 4028.

Example 2: In practical engineering, it is often necessary to record the number of operations, using the ST programming language. When the number accumulates to 10, the cumulative variable is cleared to zero.
The following is a program implementation using ST language. Accumulate the addend iCounter by triggering the function block along the rising edge.

```
VAR
bCalStart: BOOL;
FB_StartTrigR_TRIG:R_TRIG;
iCounter:word;
END_VAR
```

```
FB_StartTrigR_TRIG(CLK:=bCalStart);
IF FB_StartTrigR_TRIG.Q THEN
iCounter:=iCounter+1;
END_IF
IF iCounter=10 THEN
iCounter:=0;
END_IF
```

Note: TIME variables can also use the addition function, where two TIME variables are added to obtain a new time.

Example: t#45s + t#50s = t#1m35 s.

The selected output data type should be able to store the output results, otherwise it may cause data errors.

1-5-2-2. Subtraction operation SUB

Function: Subtraction operation instruction, subtracting two variables or constants.
Syntax: The instruction can subtract the value of IN0 by the input variable IN1 and assign the result to OUT. The subtraction operation instruction supports the following variable types: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, (L) REAL, TIME, and constants. The textual language syntax format is as follows:

```
OUT := IN0 –IN1
```

For example: create a POU, declare two Floating-point arithmetic variables rVar1 and rVar2, assign values of 3.14 and 10 respectively, and declare an rResult variable, whose value is the value obtained by subtracting rVar1 from rVar2. The specific code is as follows:

```
VAR
rVar1:REAL:=3.14;
rVar2:REAL:=10;
rResult:REAL;
END_VAR
```

```
rResult:=rVar2-rVar1;
```

The running result of the program is rResult equal to 6.86.

Note:

① TIME type variables can also use the subtraction function, where two TIME variables are subtracted to obtain a new time.

Example: t#1m35s - t#50s = t#45s, but the time result cannot have a negative value.

② TOD type variables can also use the subtraction function, where two TOD types are subtracted to obtain a new TIME type data.

Example: TOD#45:40:30- TOD#22:30:20=T#1390m10s0ms, but the time result cannot have a negative value.

1-5-2-3. Multiplication operation MUL

Function: Multiplication operation instruction, multiplying two (or more) variables or constants.
Syntax: The instruction can perform a multiplication operation on the value of the input variable IN0 until the value of INn, and assign its product to OUT. The multiplication instruction supports the following variable types: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, (L) REAL, TIME, TOD, and constants. The textual language syntax format is as follows:

```
OUT := IN0 *…*INn
```

Example: Create a POU, declare two integer variables iVar1 and iVar2, assign values of 10 and 2 respectively, and then declare an integer variable iResult, so that the result is the product of iVar1 and iVar2. The specific implementation code is as follows:

```
VAR
iVar1:INT:=10;
iVar2:INT:=2;
iResult:INT;
END_VAR
```

```
iResult:=iVar1*iVar2;
```

The running result of the program is iResult equal to 20.

1-5-2-4. Division operation DIV

Function: Division operation instruction, dividing two variables or constants.
Syntax: The instruction can divide the input variable IN0 by the value of IN1, and assign its quotient value to OUT. The division operation instruction supports the following variable types: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, and constant. The textual language syntax format is as follows:

```
OUT := IN0 / IN1
```

Example: Create a POU, declare two integer variables iVar1 and iVar2, assign values of 10 and 2 respectively, and then declare an integer variable iResult, which is the value obtained by dividing iVar1 by iVar2. The specific implementation code is as follows:

```
VAR
iVar1:INT:=10;
iVar2:INT:=2;
iResult:INT;
END_VAR
```

```
iResult:=iVar1/iVar2;
```

The running result of the program is iResult equal to 5.
Note: When using the DIV instruction, instructions such as CheckDivByte, CheckDivWord, CheckDivDWord, and CheckDivReal can be used to check whether the divisor is zero, avoiding the phenomenon of divisor being zero.

1-5-2-5. Remainder operation MOD

Function: Divide variables or constants to obtain remainder, and the result is the remainder after dividing two numbers, which is an integer data.
Syntax: The MOD instruction can assign the remainder of the input variables IN0 and IN1 to OUT, and typically uses this instruction to create equations with a remainder within a specific range. The remainder operation instruction supports the following variable types: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, and constants. The textual language syntax format is as follows:

```
OUT := IN0 MOD IN1;
```

Example: Create a POU, declare two integer variables iVar1 and iVar2, assign values of 44 and 9 respectively, and then declare an integer variable iResult to make its value to be the one after the remainder operation of iVar1 and iVar2. The specific implementation code is as follows:

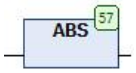
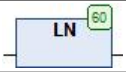
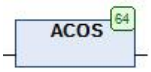
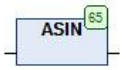
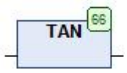
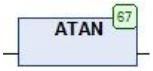
```
VAR
iVar1:INT:=44;
iVar2:INT:=9;
iResult:INT;
```

```
END_VAR
```

```
iResult:=iVar1 MOD iVar2;
```

The running result of the program is iResult equal to 8.

1-5-3. Mathematical operation instructions

Command	Icon	Function
ABS		Absolute value
SQRT		Square root
EXP		Exponent
LN		Natural logarithm
LOG		Common logarithm
SIN		Sine
COS		Cosine
ACOS		Arccosine
ASIN		Arcsine
TAN		Tangent
ATAN		Arctangent

1-5-3-1. Absolute value ABS

Function: This function instruction is used to calculate the absolute value of a number. It has nothing to do with the sign of positive and negative numbers.

Syntax: The absolute value operation instruction supports the following variable types: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, and constants. The textual language syntax format is as follows:

```
OUT := ABS (IN);
```

Example: ABS function example.

```
VAR
iVar1:INT:=-44;
iResult:INT;
END_VAR
```

```
iResult:=abs(iVar1);
```

The running result of the program is iResult equal to 44.

1-5-3-2. Square root SQRT

Function: The square root of non negative real numbers.
Syntax: The input variable IN can be BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, and constant, but the output must be of type REAL or LREAL. The textual language syntax format is as follows:

```
OUT := SQRT(IN);
```

Example: SQRT function example.

```
VAR
rVar1:REAL:=16;
rResult:REAL;
END_VAR

rResult:=SQRT(rVar1);
```

The running result of the program is rResult equal to 4.

1-5-3-3. Exponent EXP

Function: return the power of e (the base of Natural logarithm). e is a constant of 2.71828.
Syntax: The input variable IN can be BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, and constant, but the output must be type REAL or LREAL. The textual language syntax format is as follows:

```
OUT := EXP(IN);
```

Example: EXP function example.

```
VAR
rVar1:REAL:=2;
rResult:REAL;
END_VAR

rResult:=EXP(rVar1);
```

The running result of the program is rResult equal to 7.389056.

1-5-3-4. Natural logarithm LN

Function: returns the Natural logarithm of a number. The base of Natural logarithm is the constant term e (2.71828182845904).
Syntax: The input variable IN can be BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, and constant, but the output must be type REAL or LREAL. The textual language syntax format is as follows:

```
OUT := LN (IN);
```

Example: LN function example.

```
VAR
```

```
    rVar1:REAL:=45;
    rResult:REAL;
    END_VAR

    rResult:=LN(rVar1);
```

The running result of the program is rResult equal to 3.80666.

1-5-3-5. Logarithm with a base of 10 LOG

Function: Returns the logarithm of a base of 10.
Syntax: The input variable IN can be BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, and constant, but the output must be type REAL or LREAL. The textual language syntax format is as follows:

```
    OUT := LOG(IN);
```

Example: LOG function example.

```
    VAR
    rVar1:REAL:=314.5;
    rResult:REAL;
    END_VAR

    rResult:=LOG(rVar1);
```

The running result of the program is rResult equal to 2.49762.

1-5-3-6. Sine function SIN

Function: Sine function.
Syntax: The input variable IN can be BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, and constant, but the output must be type REAL or LREAL. The textual language syntax format is as follows:

```
    OUT := SIN(IN);
```

Example: SIN function example.

```
    VAR
    rVar1:REAL:=0.5;
    rResult:REAL;
    END_VAR

    rResult:=SIN(rVar1);
```

The running result of the program is rResult equal to 0.479426.

1-5-3-7. Cosine function COS

Function: Cosine function.
Syntax: The input variable IN can be BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, and constant, but the output must be type REAL or LREAL. The textual language syntax format is as follows:

```
OUT := COS(IN);
```

Example: COS function example.

```
VAR
rVar1:REAL:=0.5;
rResult:REAL;
END_VAR

rResult:=COS(rVar1);
```

The running result of the program is rResult equal to 0.877583.

1-5-3-8. Arccosine functionACOS

Syntax: The input variable IN can be BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, and constant, but the output must be type REAL or LREAL. The textual language syntax format is as follows:

```
OUT := ACOS(IN);
```

Example: ACOS function example.

```
VAR
rVar1:REAL:=0.5;
rResult:REAL;
END_VAR

rResult:=ACOS(rVar1);
```

The running result of the program is rResult equal to 1.0472.

1-5-3-9. Arcsin function ASIN

Function: Sine radian (Arcsine function).
Syntax: The input variable IN can be BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, and constant, but the output must be type REAL or LREAL. The textual language syntax format is as follows:

```
OUT := ASIN(IN);
```

Example: ASIN function example.

```
VAR
rVar1:REAL:=0.5;
rResult:REAL;
END_VAR
```

```
rResult:=ASIN(rVar1);
```

The running result of the program is rResult equal to 0.523599.

1-5-3-10. Tangent function TAN

Function: Tangent function.

Syntax: The input variable IN can be BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, and constant, but the output must be type REAL or LREAL. The textual language syntax format is as follows:

OUT:=TAN (IN);

Example: TAN function example.

```
VAR
rVar1:REAL:=0.5;
rResult:REAL;
END_VAR

rResult:= TAN (rVar1);
```

The running result of the program is rResult equal to 0.546302.

1-5-3-11. Arctangent function ATAN

Function: Tangent radian (Arctangent function).

Syntax: The input variable IN can be BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, and constant, but the output must be type REAL or LREAL. The textual language syntax format is as follows:

```
OUT := ATAN(IN);
```

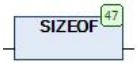
Example: ATAN function example.

```
VAR
rVar1:REAL:=0.5;
rResult:REAL;
END_VAR

rResult:= ATAN (rVar1);
```

The running result of the program is rResult equal to 0.463648.

1-5-4. Address operation instruction

Command	Icon	Function
SIZEOF		Data type size
ADR		Address operator
BITADR		Bit address operator

1-5-4-1. Data type size SIZEOF

Function: Execute this function to determine the number of bytes required for the given data type. Simply put, its function is to return the number of memory bytes occupied by an object or type.
Syntax: The return value of SIZEOF is an unsigned value, and the return value of type will be used to find the size of variable IN0. The output value of OUT is in bytes, and IN0 can be any data type. The textual language syntax format is shown below. The type of return value is an implicit data type, which will be determined according to the actual data value. See the following table for details:

```
OUT := SIZEOF(IN0);
```

Return data type of SIZEOF

SIZEOF return value	Implicit data type
0 <= size of x < 256	USINT
256 <= size of x < 65536	UINT
65536 <= size of x < 4294967296	UDINT
4294967296 <= size of x	ULINT

Example: Using the SIZEOF instruction to retrieve the memory size occupied by an array, the program is as follows:
Example of ST language:

```
VAR
arr1:ARRAY[0..4] OF INT;
var1:INT;
END_VAR
```

```
var1 := SIZEOF(arr1);
```

The program assigns the result to var1, and ultimately var1 is equal to 10. Because the arr1 array consists of 5 INT integer elements and the result unit of SIZEOF is BYTE, the program runs with a total of 10 BYTEs, indicating that arr1 occupies 10 bytes of memory.

1-5-4-2. Address operator ADR

Function: Obtain the memory address of the input variable and output it. This address can be used as a pointer within the program or passed as a pointer to a function.
Syntax: The ADR operator returns an address variable with a DWORD value, and IN0 can be of any data type. The textual language syntax format is as follows:

```
OUT :=ADR(IN0);
```

The return value of ADR is only the memory address of the variable. The memory address can store data with a length of 1 BYTE. The content operator "^" can be used to extract the content in the corresponding address, such as obtaining the memory address of var_int1 is assigned to a pointer variable, and the specific content in its corresponding address is extracted by using the "^" operator and assigned to var_int2. The implementation program is as follows:

```
pt := ADR(var_int1);
var_int2:= pt^;
```

Example 1: Using the ADR instruction to retrieve an array, the program is as follows:
Example of ST language:

```
VAR
arr1:ARRAY[0..4] OF INT;
dwVar:DWORD;
END_VAR

dwVar:=ADR(arr1);
```

Example 2: Using the ADR instruction to retrieve an array, the program is as follows:
Example of ST language:

```
VAR
arr1:ARRAY[0..4] OF INT;
dwVar:DWORD;
END_VAR

dwVar:=ADR(arr1);
```

Example 3: An example of using the ADR instruction to retrieve an array is as follows:
Example of ST language:

```
VAR
pt:POINTER TO INT;
var_int1:INT;
var_int2:INT;
END_VAR

pt := ADR(var_int1);
var_int2:=pt^;
```

1-5-4-3. Bit address operator BITADR

Function: Returns the bit address information offset of the allocation variable.
Syntax: The BITADR operator returns an address variable with a DWORD value, and IN0 can be of any data type. The textual language syntax format is as follows:

```
OUT :=BITADR(IN0);
```

The implementation program is as follows, where BITADR returns a bit offset numerical address in the DWORD variable type. Note that the offset value depends on whether the option type address can be obtained from the target system. The maximum DWORD defines the memory area as follows:

BITADR offset addresses for each address area

Address area	Start address	Explanation
Memory	16x40000000	%M
Input	16x80000000	%I
Output	16xC0000000	%Q

An example of using the BITADR instruction to retrieve bit address information is shown in the following program. Example of ST language:

```
VAR
var1 AT %IX2.3:BOOL;
bitoffset: DWORD;
END_VAR

bitoffset:=BITADR(var1);
```

The running result is 80000013 in hexadecimal, where "2" in %IX2.3 represents 2 Bytes and ". 3" represents the 4th Bit, so its address is equal to 2*8+4=20. Convert the decimal 20 to the hexadecimal 14. Because the first address corresponding to Zone I is stored starting from 80000000, it is not difficult to understand that the actual address corresponding to hexadecimal 14 is 16#80000013. The schematic diagram is shown in the table below:

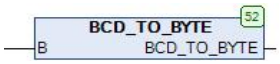
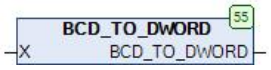
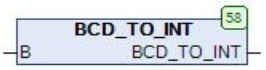
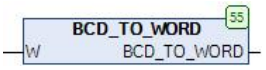
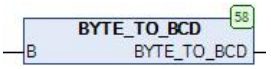
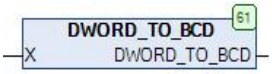
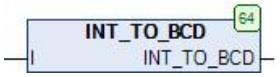
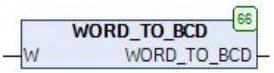
BITADR example explanation

offset value	data content
16#80000000	
16#80000001	
16#80000002	
...	
...	
16#80000011	
16#80000012	
16#80000013	bitoffset

1-5-5. Data conversion instructions

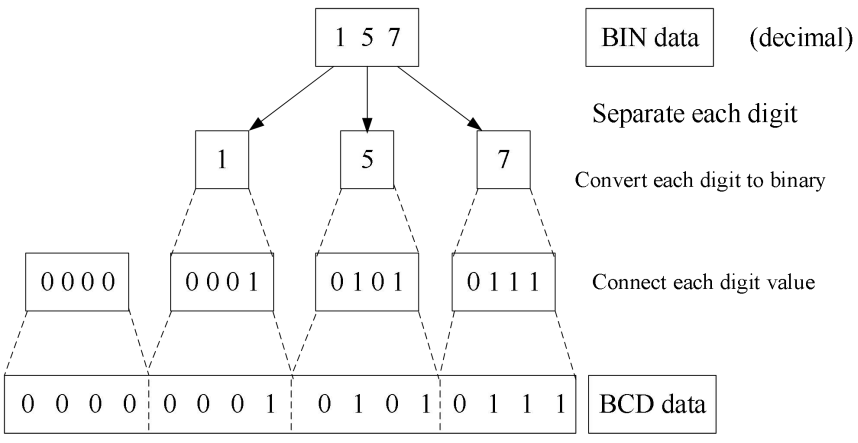
Before using this instruction, you need to add the util.library.
Syntax:<TYPE1>_TO_< TYPE2>
It is strictly prohibited to implicitly convert "larger" data types to "smaller" data types for use, because information may be lost when converting from larger data types to smaller data types.
If the converted value exceeds the storage range of the target data type, the high bytes of this number will be ignored. Example: Convert INT Type to BYTE type, or DINT Type to WORD type.
For <TYPE>_TO_STRING conversion, the string is generated from the left. If the length of the defined string is less than the length of<TYPE>, the right part will be truncated.

BCD code and integer conversion instruction

Command name	Command icon	Function
BCD_TO_BYTE		BCD convert to BYTE
BCD_TO_DWORD		BCD convert to DWORD
BCD_TO_INT		BCD convert to INT
BCD_TO_WORD		BCD convert to WORD
BYTE_TO_BCD		BYTE convert to BCD
DWORD_TO_BCD		DWORD convert to BCD
INT_TO_BCD		INT convert to BCD
WORD_TO_BCD		WORD convert to BCD

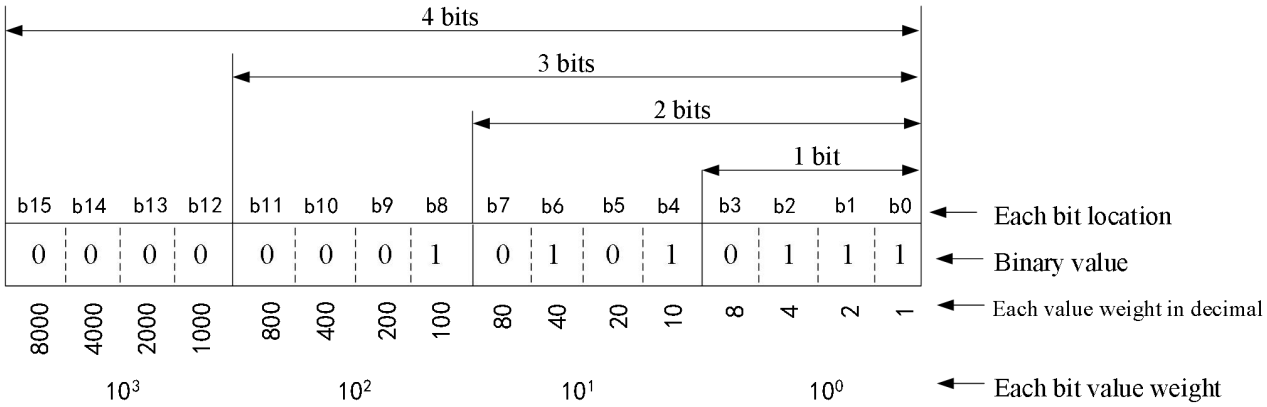
1-5-5-1. BCD code and integer data conversion

BCD (Binary Coded Decimal... BCD) refers to the use of 4-bit binary numbers to represent the values of each digit in a decimal number in parallel. For example, in BIN data, BCD data 0000 0001 0101 0111 (343) is used to represent the decimal number "157" in the following way.



BCD example

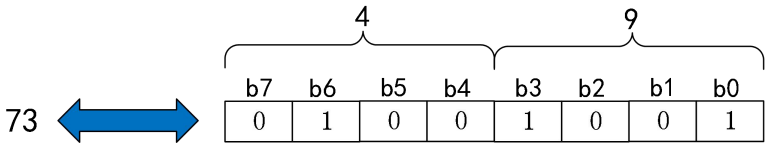
When BCD data is stored in 16 bits memory, it can handle values ranging from 0 to 9999 (the maximum value of 4 bits). The weights of each bit are shown in the following figure:



Weights of each value represented by BCD in decimal system

Example: Using the ST programming language, convert BCD code 73 to integer data.

```
i:=BCD_TO_INT(73);
```

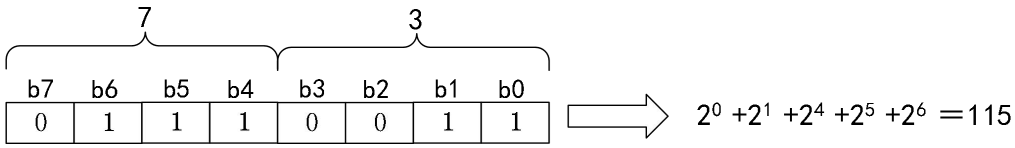


Application of convert BCD to INT

As shown in the figure, use BCD_TO_ INT instruction performs conversion, as the result of converting 73 to binary is 01001001, the result of converting it to BCD is 49.

Example: Using ST programming language to convert integer data 73 into BCD code.

```
i:=INT_TO_BCD(73);
```



Application of convert INT to BCD

As shown in the above figure, using INT_TO_BCD instruction for conversion. The program converts 73 in decimal system to BCD code, and the result is 01110011, so the final BCD code Decimal representation result is 115.

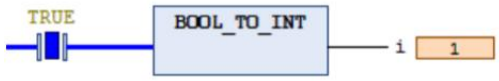
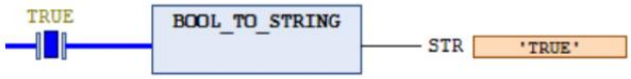
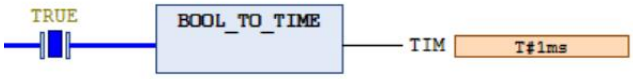
1-5-5-2. BOOL_TO_<TYPE> boolean type conversion data

Function: convert Boolean data type to other data types.

Support data types: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, TIME, DATE, TOD, DT, and STRING.

- When the output is digital type: if the input is TRUE, the output is 1. If the input is FALSE, the output is 0.
- When outputting as a string type: If the input is TRUE, the string 'TRUE' is output. If the input is FALSE, the output is the string 'FALSE'.

BOOL_TO_<TYPE> conversion example

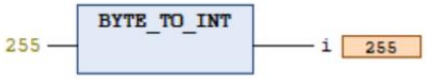
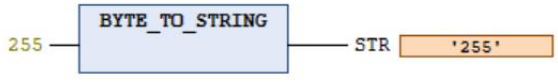
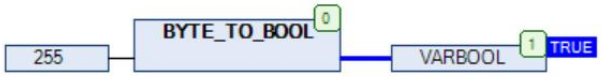
Command name	Command example	Function
BOOL_TO		BOOL convert to INT
		BOOL convert to STRING
		BOOL convert to TIME
		BOOL convert to TOD

1-5-5-3. BYTE_TO_<TYPE> byte type conversion data

Function: convert byte type to other data types. Support data types: BOOL, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, TIME, DATE, TOD, DT, and STRING.

- When the output is BOOL: When the input is not equal to 0, the output is TRUE. When the input is equal to 0, the output is FALSE.
- When the output is TIME or TOD: the input will be converted in milliseconds.
- When the output is DATE or DT: the input will be converted in seconds.

BYTE_TO_<TYPE> conversion example

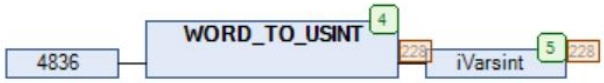
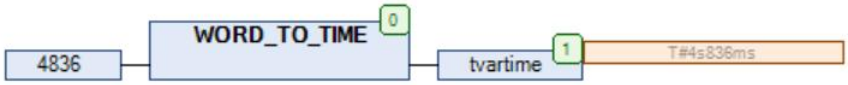
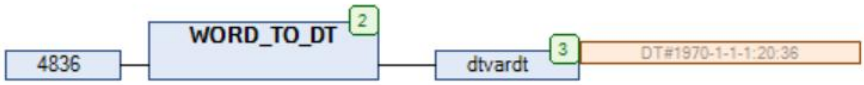
Comand name	Command example	Function
BYTE_TO		BYTE convert to INT
		BYTE convert to STRING
		BYTE convert to TIME
		BYTE convert to BOOL

1-5-5-4. < Integer data > _TO_<TYPE> integer type conversion instruction

Function: Convert integer type data to other data types.
Support data types: BOOL, BYTE, SINT, WORD, DWORD, USINT, INT, UINT, DINT, UDINT, REAL, TIME, DATE, TOD, DT, and STRING.

- When the output is BOOL: When the input is not equal to 0, the output is TRUE. When the input is equal to 0, the output is FALSE.
- When the output is TIME or TOD, the input will be converted in milliseconds.
- When the output is DATE or DT, the input will be converted in seconds.

< Integer data > _TO_<TYPE> Example of integer type conversion

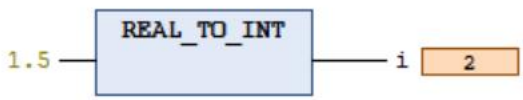
Command name	Command example	Function
WORD_TO		Word convert to USINT
		Word convert to Time
		Word convert to DT

1-5-5-5. REAL_TO_<TYPE> Real type conversion instruction

Function: convert Floating-point arithmetic numbers to other types of data. When converting a Floating-point arithmetic number to other types of data, first round the value to an integer value, and then convert it to a new quantity type.
Support data types: BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, TIME, DATE, TOD, DT, and STRING.

- When the output is BOOL: When the input is not equal to 0, the output is TRUE. When the input is equal to 0, the output is FALSE.
- When the output is TIME or TOD: the input will be converted in milliseconds.
- When the output is DATE or DT: the input will be converted in seconds.

REAL_TO_<TYPE> Example of real type conversion

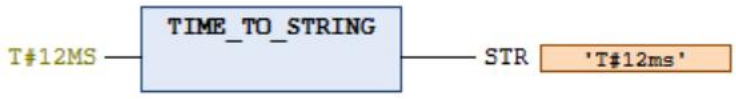
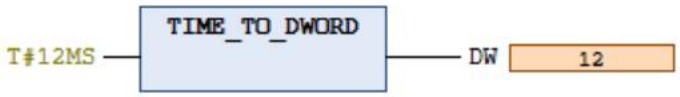
Command name	Command example	Function
REAL_TO		REAL convert to INT

1-5-5-6. TIME_TO_<TYPE> time type conversion instruction

Function: Convert time-based data to other types of data, and store the time in milliseconds internally as a DWORD type (starting from 00:00 am for the TIME_OF_DAY variable).
Support data types: BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, TIME, DATE, TOD, DT, and STRING.

- When the output is BOOL: When the input is not equal to 0, the output is TRUE. When the input is equal to 0, the output is FALSE.

TIME_TO_<TYPE> Example of time type conversion

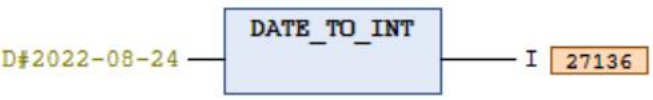
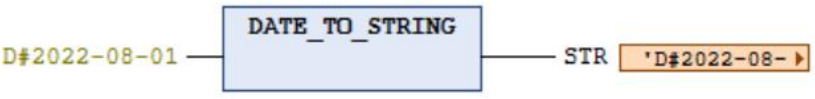
Command name	Command example	Function
TIME_TO		TIME convert to STRING
		TIME convert to DWORD

1-5-5-7. DATE_TO_<TYPE> Date type conversion instruction

Function: Convert date type into other types, store dates in seconds internally, starting from January 1, 1970.
Support data types: BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, TIME, DATE, TOD, DT, and STRING.

- When the output is BOOL: When the input is not equal to 0, the output is TRUE. When the input is equal to 0, the output is FALSE.

DATE_TO_<TYPE> Example of date type conversion instruction

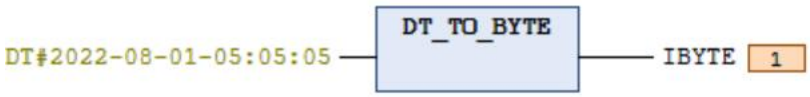
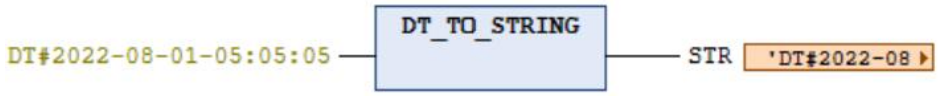
Command name	Command example	Function
DATE_TO		DATE convert to INT
		DATE convert to STRING

1-5-5-8. DT_TO_<TYPE> Date time type conversion instruction

Function: Convert date time data into other types of data, with dates stored internally in seconds, starting from January 1, 1970.
Support data types: BOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, TIME, DATE, TOD, DT, and STRING.

- When the output is BOOL: When the input is not equal to 0, the output is TRUE. When the input is equal to 0, the output is FALSE.

DT_TO_<TYPE> Date time type conversion example

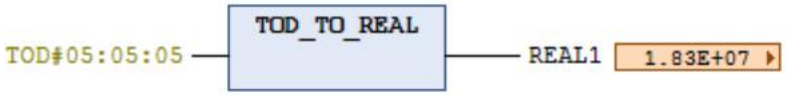
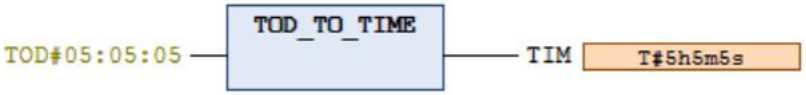
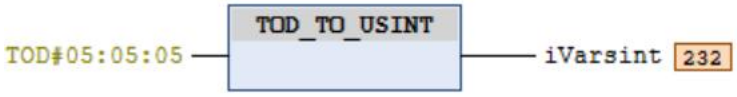
Command name	Command example	Function
DT_TO		DT convert to BYTE
		DT convert to STRING

1-5-5-9. TOD_TO_<TYPE> Time type conversion instruction

Function: Convert time-based data into other types of data, and convert dates internally in milliseconds.
Support data types: BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, TIME, DATE, TOD, DT, and STRING.

- When the output is BOOL: When the input is not equal to 0, the output is TRUE. When the input is equal to 0, the output is FALSE.

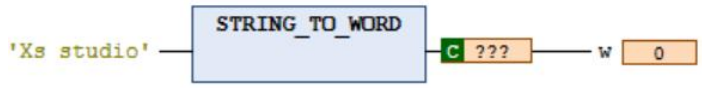
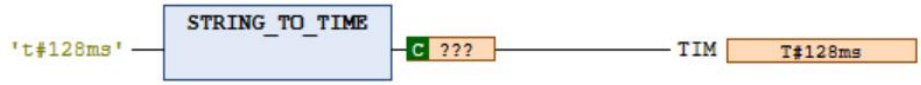
TOD_TO_<TYPE> Example of time type conversion

Command name	Command example	Function
TOD_TO		TOD convert to REAL
		TOD convert to TIME
		TOD convert to USINT

1-5-5-10. STRING_TO_<TYPE> Character type conversion instruction

Function: Convert strings to other types of data. String variables must contain a valid target variable value, otherwise the conversion result is 0.
Support data types: BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, TIME, DATE, TOD, DT, and STRING.

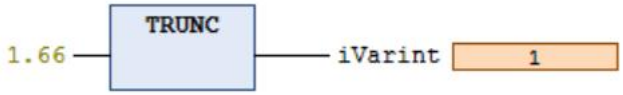
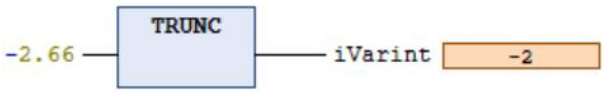
STRING_TO_<TYPE> Example of character type conversion

Command name	Command example	Function
STRING_TO		STRING convert to WORD
		STRING convert to TIME

1-5-5-11. Rounding TRUNC

Function: Truncate the decimal part of the data and only retain the integer part.
Support data types: input is REAL type, output is INT, WORD, DWORD type.

TRUNC example

Command name	Command example	Function
TRUNC		Round positive numbers
		Rounding negative numbers

Notes:
① When changing from a larger data type to a smaller data type, information may be lost.
② This instruction only truncates the integer part. If you want to round it up, you can use REAL_TO_INT instruction.

2. Special instructions

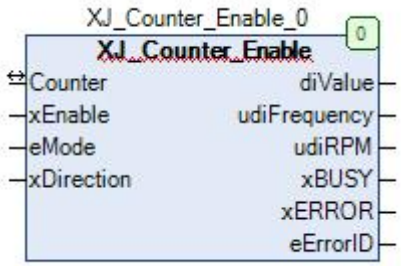
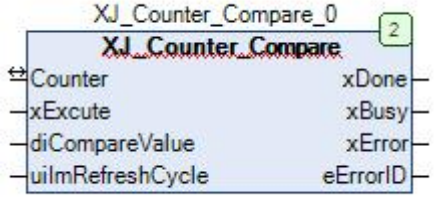
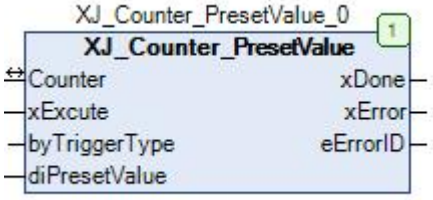
2-1. XSA series high speed count instructions

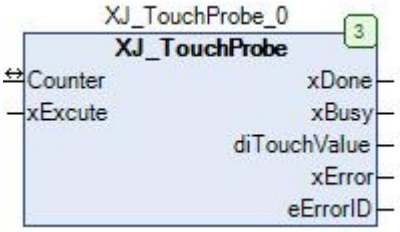
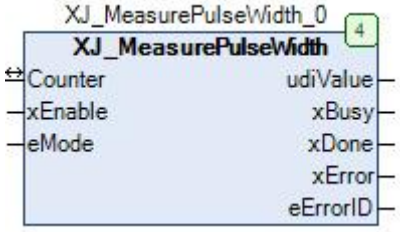
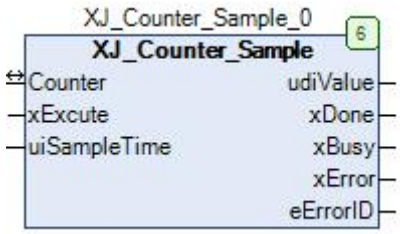
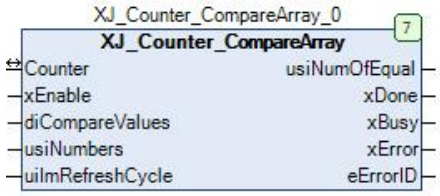
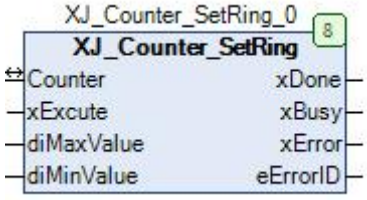
2-1-1. Function overview

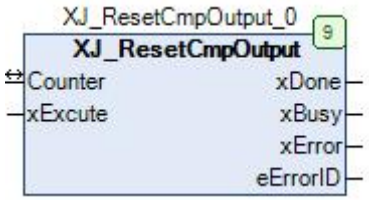
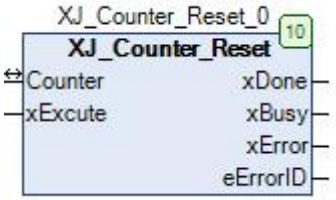
The XSA series PLC has a high-speed counting function, which enables the measurement of high-speed input signals such as measurement sensors and rotary encoders by selecting different counters. Its maximum measurement frequency can reach 1MHz. At present, the instruction library and high-speed IO interface are only supported by XSA330.
The XSDH and XSLH series will be supported in the future. For firmware versions below V1.1.0, please refer to Chapter 2-2 for the XSDH and XSLH series instruction libraries.

2-1-2. Function block

2-1-2-1. Command format

Command	Name	Graph	ST language
XJ_Counter_Enable	Enable the high speed counter		XJ_Counter_Enable(Counter:= , xEnable:= , eMode:= , xDirection:= , diValue=> , udiFrequency=> , udiRPM=> , xBUSY=> , xERROR=> , eErrorID=>);
XJ_Counter_Compare	Compare consistent output		XJ_Counter_Compare(Counter:= , xExcute:= , diCompareValue:= , uiImRefreshCycle:= , xDone=> , xBusy=> , xError=> , eErrorID=>);
XJ_Counter_PresetValue	Preset value write in		XJ_Counter_PresetValue(Counter:= , xExcute:= , byTriggerType:= , diPresetValue:= , xDone=> , xError=> , eErrorID=>);

Command	Name	Graph	ST language
XJ_TouchProbe	Probe		XJ_TouchProbe(Counter:= , xExcute:= , xDone=> , xBusy=> , diTouchValue=> , xError=> , eErrorID=>);
XJ_MeasurePulseWidth	Read the pulse width measurement value of the counter		XJ_MeasurePulseWidth(Counter:= , xEnable:= , eMode:= , udiValue=> , xBusy=> , xDone=> , xError=> , eErrorID=>);
XJ_Counter_Sample	Counter sample		XJ_Counter_Sample(Counter:= , xExcute:= , uiSampleTime:= , udiValue=> , xDone=> , xBusy=> , xError=> , eErrorID=>);
XJ_Counter_CompareArray	Multiple segments compare		XJ_Counter_CompareArray(Counter:= , xEnable:= , diCompareValues:= , usiNumbers:= , uiImRefreshCycle:= , usiNumOfEqual=> , xDone=> , xBusy=> , xError=> , eErrorID=>);
XJ_Counter_SetRing	Set ring counter		XJ_Counter_SetRing(Counter:= , xExcute:= , diMaxValue:= , diMinValue:= , xDone=> , xBusy=> , xError=> , eErrorID=>);

Command	Name	Graph	ST language
XJ_ResetCmpOutput	Reset the compare consistent output port		XJ_ResetCmpOutput(Counter:= , xExcute:= , xDone=> , xBusy=> , xError=> , eErrorID=>);
XJ_Counter_Reset	Clear the error		XJ_Counter_Reset(Counter:= , xExcute:= , xDone=> , xBusy=> , xError=> , eErrorID=>);

Note: The reason why the graphical representation and ST representation of some instructions are marked with red wavy lines is that VAR_IN_OUT'Counter' must be assigned a value, then the program can be compiled and downloaded.

2-1-2-2. Enable the high speed counter【XJ_Counter_Enable】

(1) Input variables

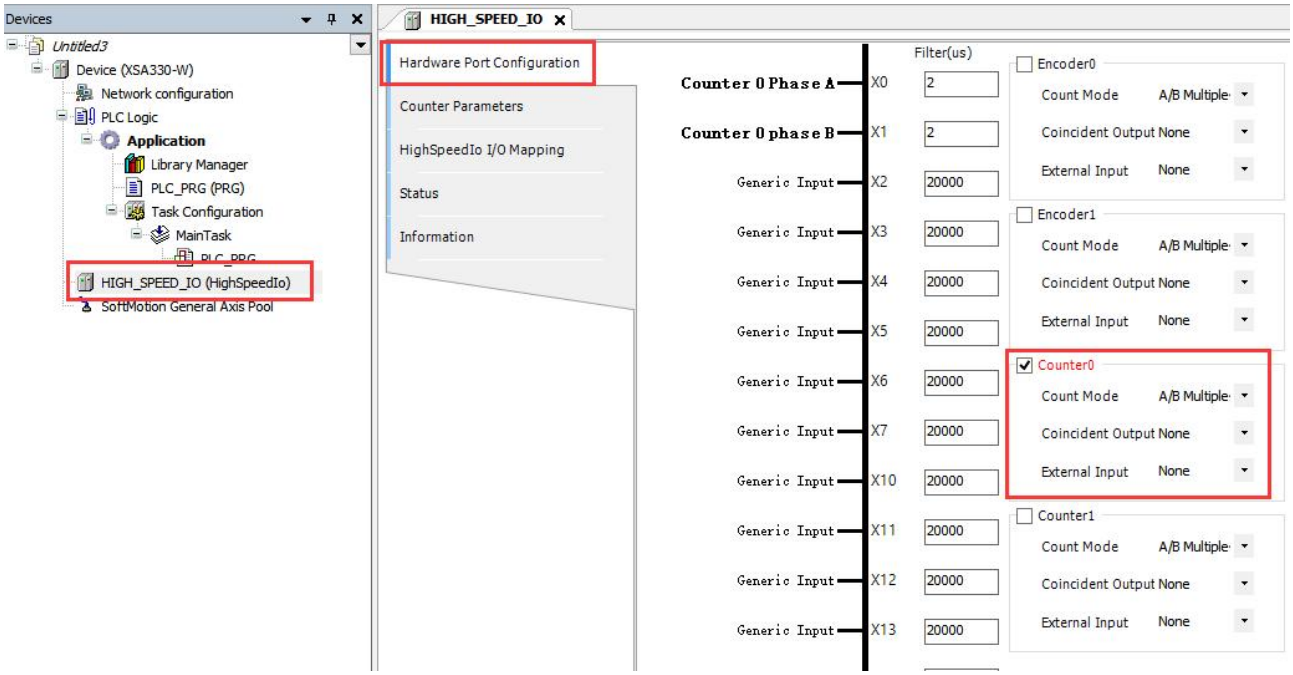
Input variable	Name	Data type	Effective range	Initial value	Description
Counter	High speed counting input port	XJ_COUNTER_REF	-	-	After configuring the high-speed counter input in the high-speed IO interface, it will be automatically instantiated. Select the corresponding instantiation name of the high-speed counter
xEnable	Enable	BOOL	TRUE, FALSE	FALSE	Normally ON the enable to count
eMode	Counting mode	HSC_EDGE_MODE	PosEdge, NegEdge, BothEdge	PosEdge	PosEdge: rising edge counting NegEdge: falling edge counting BothEdge: double edge counting
xDirection	Direction	BOOL	TRUE, FALSE	FALSE	FALSE: up counting TRUE: down counting

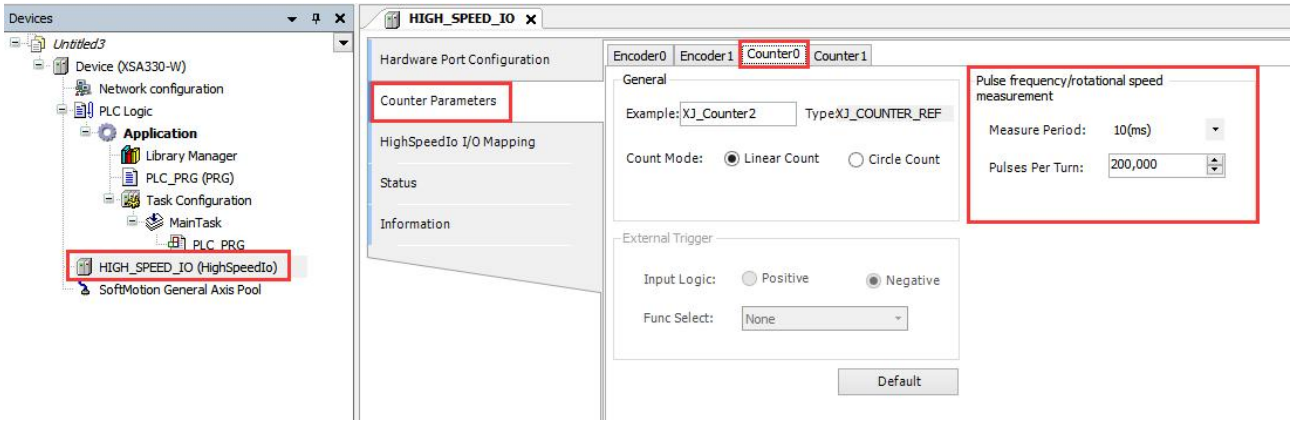
(2) Output variables

Output variables	Name	Data type	Effective range	Initial value	Description
diValue	High speed counting value	DINT	Data range	0	High speed counting value
udiFrequency	Pulse frequency measurement value	UDINT		0	Unit: Hz, if it is a low frequency, it can be used in conjunction with the measurement period through the interface

udiRPM	Rotation speed per minute	UDINT		0	Unit: r/min, used in conjunction with the number of pulses per turn in the interface configuration
xBUSY	Busy	BOOL	TRUE,FALSE	FALSE	
xError	Error flag	BOOL	TRUE,FALSE	FALSE	
eErrorID	Error type	HSIO_ERROR	-	0	

To use high-speed counting, it is necessary to check and configure the corresponding high-speed counter in the hardware parameters, set the counting method and pulse frequency/rotation speed measurement related parameters for the counter parameters.





Note:

(1) Please assigning a value to VAR_IN_OUT'Counter' when calling XJ_Counter_Enable, then can the program be compiled and downloaded.

(2) Notes when setting xDirection:

- When single-phase or AB phase counting, counting up/down based on the xDirection terminal status.
- When in P+D mode, the direction terminal FALSE and the xDirection terminal FALSE are counting up. Direction terminal FALSE, xDirection terminal TRUE, represents the subtraction count. The direction terminal is TRUE, while the xDirection terminal is FALSE, indicating a subtractive count. The direction terminal is true, and the xDirection terminal is true, indicating the addition count.

(3) The meaning of measurement cycle is to collect the number of pulses during this period for calculation:

- The measurement cycle is 1000ms, and the minimum measurable frequency is 1Hz. The

measurement cycle is 1ms, and the minimum measurable frequency is 1kHz.

- The measurement cycle is 10ms, and the minimum measurable frequency is 100Hz. The measurement cycle is 100ms, and the minimum measurable frequency is 10Hz. The maximum measurement frequency is the frequency that can be received by high-speed counting.

(4) The unit of rotation speed per minute is r/min, and the measured values may fluctuate back and forth due to hardware influence.

(5) The difference between linear counting and circular counting:

- Linear counting: (DownLimitValue, UpLimitValue)
- Circular counting: [iRingDownValue, iRingUpValue]
- Counter 0-3: The linear counting range is (-2147483648, 2147483647), not include -2147483648 and 2147483647.
- Ring counting range is [-2147483648, 2147483647], include -2147483648 and 2147483647.

(6) When Counter0/Counter1 selects AB phase counting, only A-phase frequency is displayed. Set in single phase, AB phase interface.

(7) If the count value reaches near the boundary value, the counter overflow may not report an error due to the influence of the scanning cycle.

(8) When downloading the program, the counter value is not cleared to zero.

(9) Please enable the XJ_Counter_Enable before using XJ_Counter_Sample function block.

2-1-2-3. Compare consistent output【XJ_Counter_Compare】

Set a compare consistent output. If high-speed counting interrupt function is required, it needs to call XJ_EnableInterrupt to open the interrupt.

Note: Multi segment comparison and single segment comparison are executed in the triggering order, and those triggered later will not interrupt those triggered earlier. The later triggered execution will take effect after the first triggered execution is completed.

(1) Input variable

Input variables	Name	Data type	Effective range	Initial value	Description
Counter	High speed counter input port	XJ_COUNTER_REF	-	-	High speed counter, it is necessary to define the high-speed counting input terminal (see functional description)
xExecute	Trigger	BOOL	TRUE,FALSE	FALSE	Trigger
diCompareValue	appoint the compare value	DINT		0	
uiImRefreshCycle	Hardware direct output time	UINT		0	Unit: 100us, maximum output time is 3000ms.

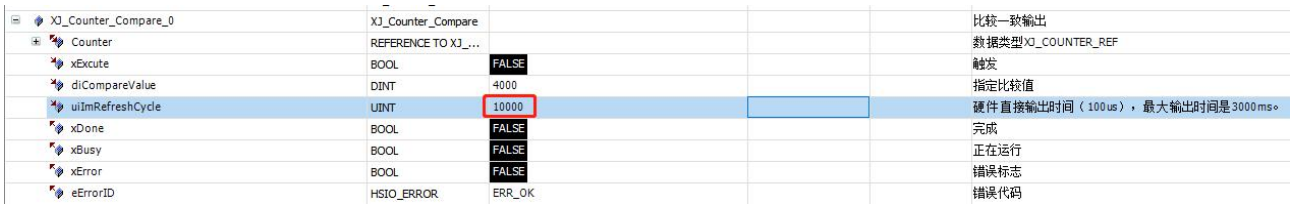
(2) Output variables

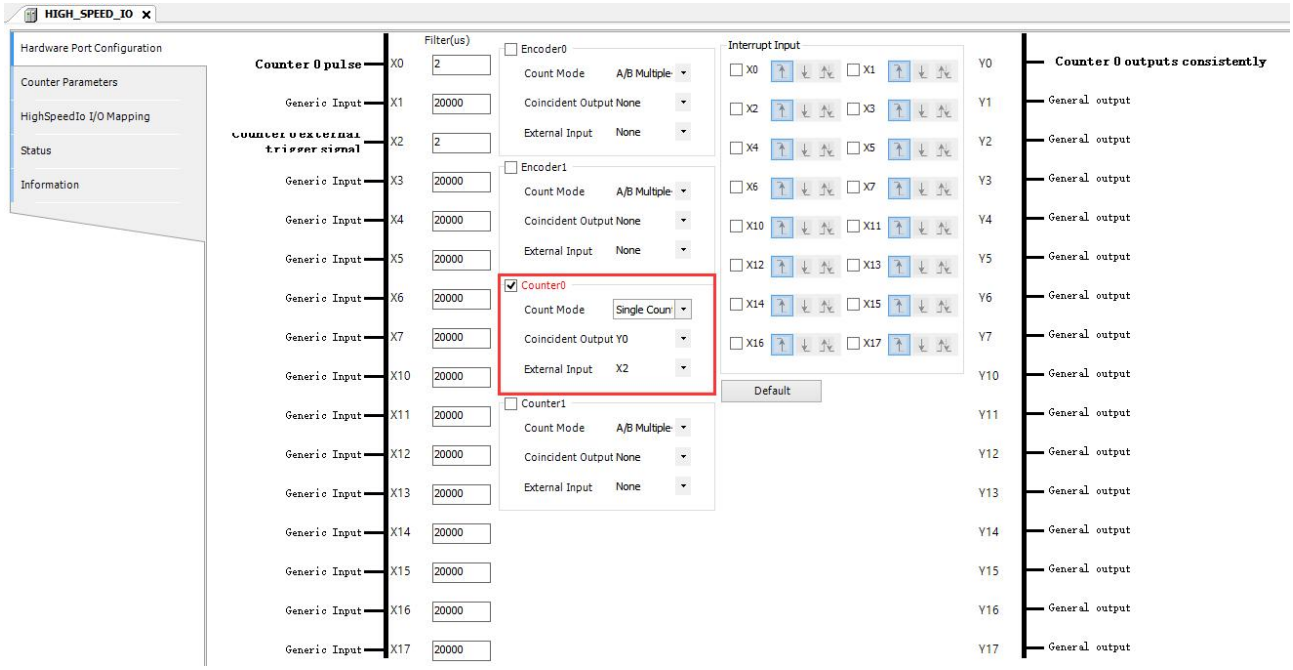
Output variables	Name	Data type	Effective range	Initial value	Description
xDone	Completed flag	BOOL	TRUE,FALSE	FALSE	After the instruction execution is completed, the flag is TRUE
xBusy	Running	BOOL	TRUE,FALSE	FALSE	
xError	Error flag	BOOL	TRUE,FALSE	FALSE	
eErr1orID	Error type	HSIO_ERROR	-	0	

Note:

(1) The control of Y0-Y7 can be configured in the background, and each counter can freely select Y0-Y7, with a certain delay in output. The output time is 100us-3000ms. If ImRefreshCycle is 0, call XJ_Reset CmpOutput to lower the output.
(2) ImRefresh=1: Hardware immediately outputs, Y0-Y7 cannot be freely selected, there is no delay, and the output time is set through ImRefreshCycle. The output time is 0-3000ms.
(3) XJ_EnableInterrupt needs to be called in advance to open compare consistent interrupt.
(4) After hot reset and cold reset, maintain the previous comparison value.

Set compare consistent output and call XJ_EnableInterrupt to open the interrupt.
If hardware output is not set, it is necessary to turn on compare consistent interrupt. If hardware output is set, compare consistent interrupt may not be turned on.
For example, if the output time is set to 1 second, the hardware output of Y will be 1 second after comparing the values, and the software status of Y will remain unchanged, it will not display any output. Execute interrupt program after external trigger input.





2-1-2-4. Preset value write in【XJ_Counter_PresetValue】

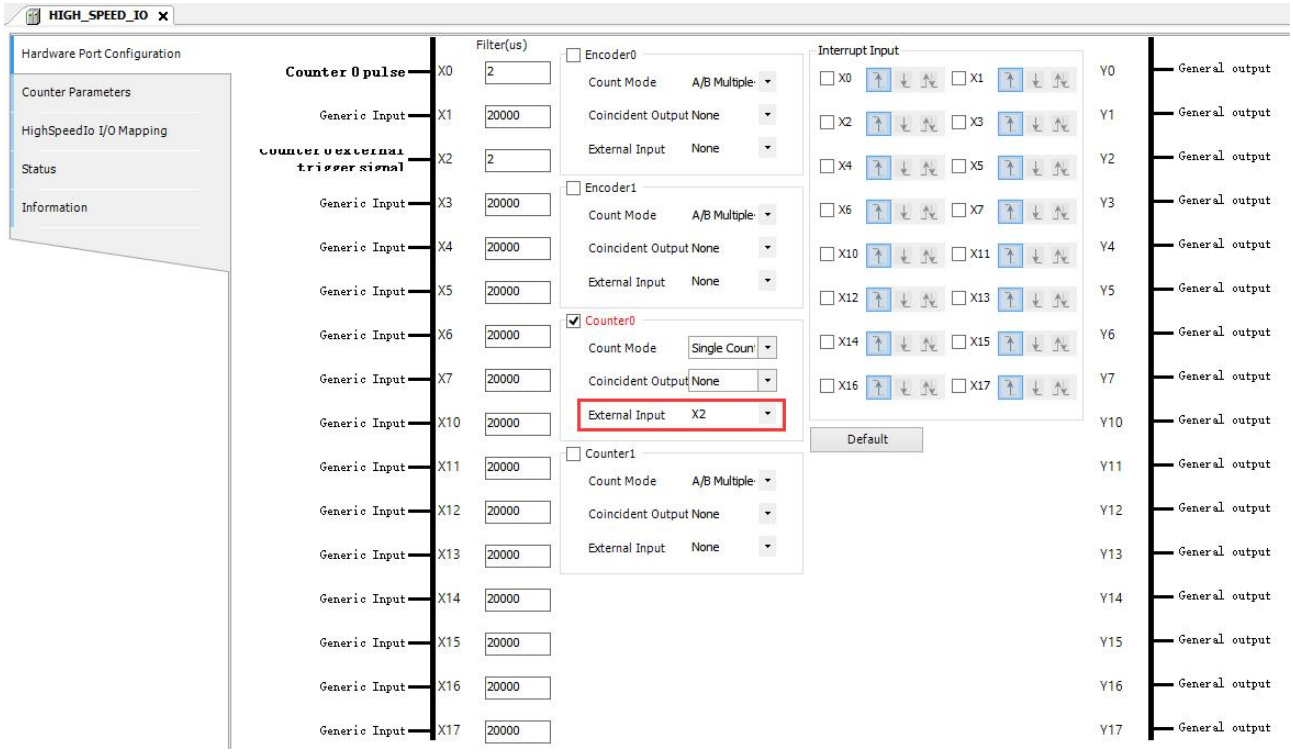
(1) Input variables

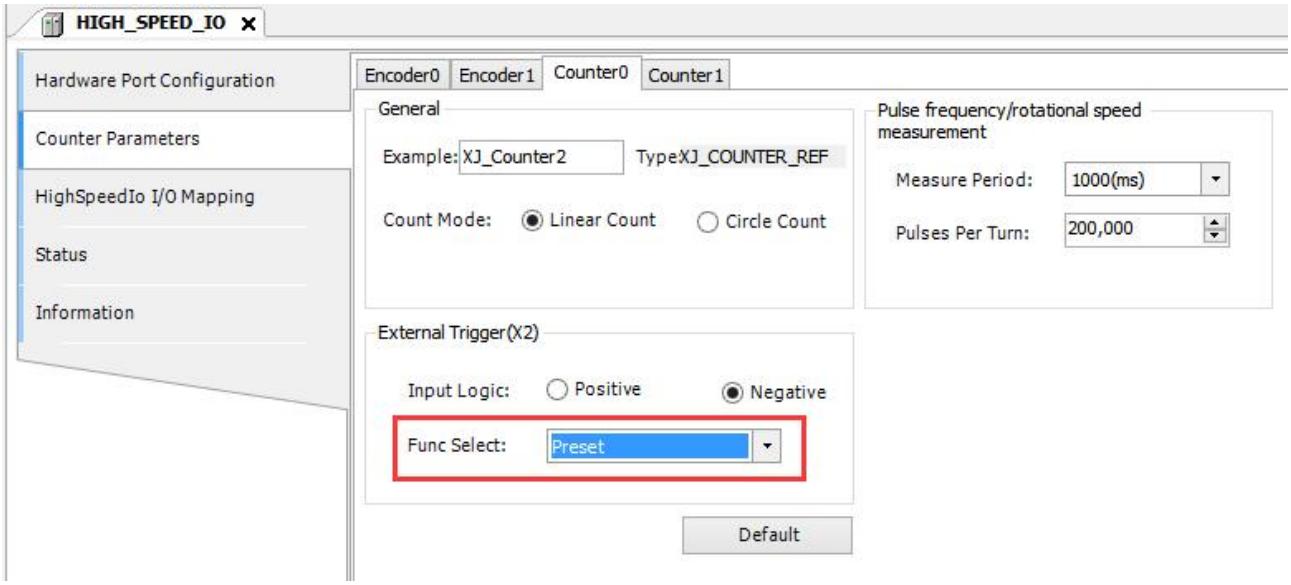
Input variables	Name	Data type	Effective range	Initial value	Description
Counter	High speed counter input port	XJ_COUNTER_REF	-	-	High speed counter, it is necessary to define the high-speed counting input terminal (see functional description)
xExecute	Trigger	BOOL	TRUE,FALSE	FALSE	Trigger
byTrigger Type	Trigger type	HSC_PSV_TIGGLE		0	0: Rising edge triggered write; 1: External input triggering; 2: When comparing consistent outputs, it is preset and triggered at the rising edge.
diPresetValue	Preset value	DINT	Data range	0	Write high-speed count preset value

(2) Output variables

Output variables	Name	Data type	Effective range	Initial value	Description
xDone	Completed flag	BOOL	TRUE,FALSE	FALSE	After completing the write, the flag is TRUE
xError	Error flag	BOOL	TRUE,FALSE	FALSE	
eErrorID	Error type	HSIO_ERROR	-	0	

Note: If the value displayed for ErrorID is 2, it is because the CounterID range is not between 0 and 3.
0: Rising edge triggered writing. After the instruction is triggered, the preset value can be written into the high-speed count.
1: External input triggering. After the command is triggered, wait for the set external input terminal signal, and if there is a signal, the preset value can be written in.





2: When comparing consistent outputs, it is preset and triggered at the rising edge.

2-1-2-5. Probe【XJ_TouchProbe】

(1) Input variables

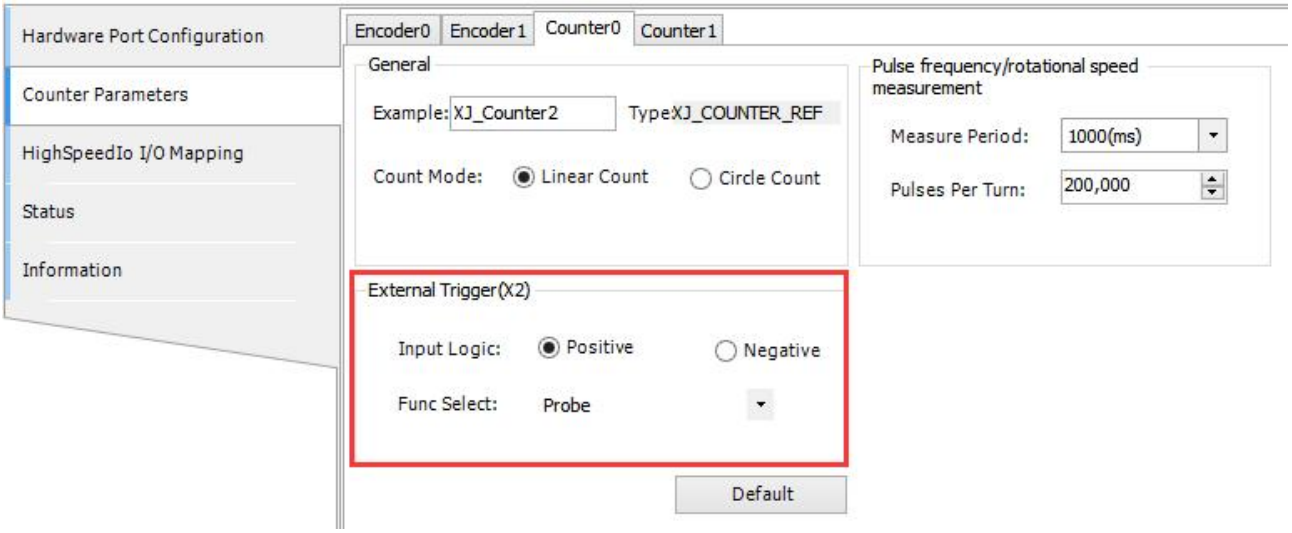
Input variables	Name	Data type	Effective range	Initial value	Description
Counter	High speed counter input port	XJ_COUNTER_REF	-	-	High speed counter, it is necessary to define the high-speed counting input terminal (see functional description)
xExcute	Trigger	BOOL	TRUE,FALSE	FALSE	Trigger

(2) Output variables

Output variables	Name	Data type	Effective range	Initial value	Description
xDone	Completed flag	BOOL	TRUE,FALSE	FALSE	After completing the write, the flag is TRUE
xBusy	Running	BOOL	TRUE,FALSE	FALSE	
diTouchVaule	Latch value	DINT		0	
xError	Error flag	BOOL	TRUE,FALSE	FALSE	
eErrorID	Error type	HSIO_ERROR	-	0	

Set the terminal for the probe function. After triggering this terminal, the current high-speed count value can be locked, as shown in the following figure. The positive logic, takes effect after the rising edge is triggered.

XJ_TouchProbe_0	XJ_TouchProbe				
Counter	REFERENCE TO XJ_COUNTER_REF				数据类型XJ_COUNTER_REF
xExcute	BOOL	FALSE			触发
xDone	BOOL	FALSE			完成标志
xBusy	BOOL	FALSE			正在运行
diTouchValue	DINT	0			锁存值
xError	BOOL	FALSE			错误标志
eErrorID	HSIO_ERROR	ERR_OK			错误代码



2-1-2-6. Read the pulse width measurement value of the counter【XJ_MeasurePulseWidth】

(1) Input variables

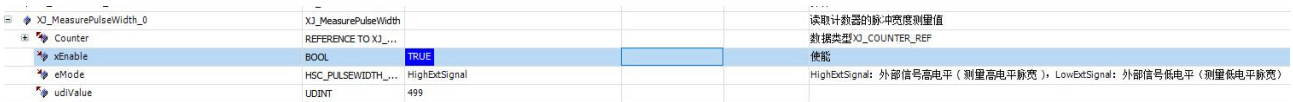
Input variables	Name	Data type	Effective range	Initial value	Description
Counter	High speed counter input port	XJ_COUNTER_REF	-	-	High speed counter, it is necessary to define the high-speed counting input terminal (see functional description)
xExecute	Trigger	BOOL	TRUE,FALSE	FALSE	Trigger
eMode	Measure high/low level pulse width	HSC_PULSEWIDTH_TYPE	0,1	0	0: External signal high level (measuring high level pulse width) 1: External signal low-level (measuring low-level pulse width)

(2) Output variables

Output variables	Name	Data type	Effective range	Initial value	Description
udiValue	Measuring value	UDINT		0	
xDone	Completed flag	BOOL	TRUE,FALSE	FALSE	After completing the write, the flag is TRUE
xBusy	Running	BOOL	TRUE,FALSE	FALSE	
xError	Error flag	BOOL	TRUE,FALSE	FALSE	
eErrorID	Error type	HSIO_ERROR	-	0	

Note: Measure the duration of high and low levels in us microseconds.

Example: If the frequency is 1kHz, it is equivalent to sending a pulse every 1ms, and if there is a high or low level every 0.5ms, the measured high or low level is about 500us.



2-1-2-7. Counter sample【XJ_Counter_Sample】

(1) Input variables

Input variables	Name	Data type	Effective range	Initial value	Description
Counter	High speed counter input port	XJ_COUNTER_REF	-	-	High speed counter, it is necessary to define the high-speed counting input terminal (see functional description)
eExecute	Trigger	BOOL	TRUE,FALSE	FALSE	Trigger
uiSampleTime	Sampl time	UINT		0	Sample time (10ms~65535ms)

(2) Output variables

Output variables	Name	Data type	Effective range	Initial value	Description
udiValue	Sample value	UDINT		0	
xDone	Completed flag	BOOL	TRUE,FALSE	FALSE	After completing the write, the flag is TRUE
xBusy	Running	BOOL	TRUE,FALSE	FALSE	
xError	Error flag	BOOL	TRUE,FALSE	FALSE	
eErrorID	Error type	HSIO_ERROR	-	0	

Note:

(1) XJ_Counter_Sample can run after XJ_Counter_Enable is enabled.

(2) After the command is triggered, the sampled signal count value will not change in real time, but will be displayed after the command is completed, displaying the high-speed count value collected within the set time.

2-1-2-8. Multiple segments compare【XJ_Counter_CompareArray】

Compare consistent output: Up to 100 comparison values can be set, and the Done signal is the output.
Note: Multi-segment comparison and single segment comparison are executed in the triggering order, and those triggered later will not interrupt those triggered earlier. The later triggered execution will take effect after the first triggered execution is completed.

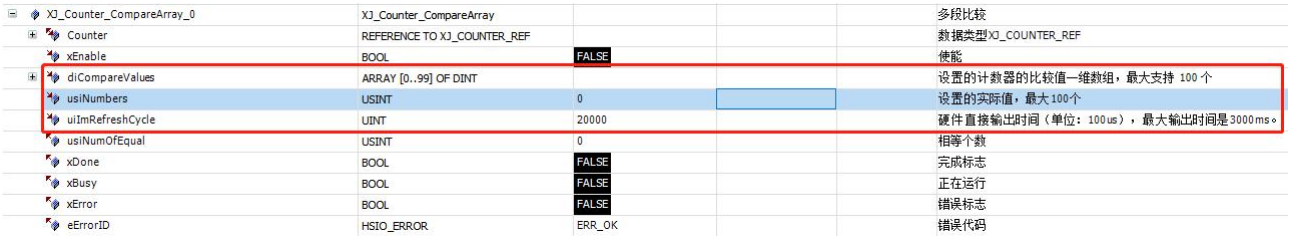
(1) Input variables

Input variables	Name	Data type	Effective range	Initial value	Description
Counter	High speed counter input port	XJ_COUNTER_REF	-	-	High speed counter, it is necessary to define the high-speed counting input terminal (see functional description)
xEnable	Enable	BOOL	TRUE,FALSE	FALSE	Enable
diCompareValues	Comparison value one-dimensional array	ARRAY		0	A one-dimensional array of comparison values for the set counters, supporting a maximum of 100
usiNumbers	set actual value	USINT		0	The actual value set, allowed to be set to 1, with a maximum of 100
uiImRefreshcycle	hardware direct output time	UINT		0	The unit is 100us, and the maximum output time is 3000ms. For example, setting 10000 means 1000ms

(2) Output variables

Output variables	Name	Data type	Effective range	Initial value	Description
usiNumOfEqual	Equal numbers	USINT		0	
xDone	Completed flag	BOOL	TRUE,FALSE	FALSE	After completing the write, the flag is TRUE
xBusy	Running	BOOL	TRUE,FALSE	FALSE	
xError	Error flag	BOOL	TRUE,FALSE	FALSE	
eErrorID	Error type	HSIO_ERROR	-	0	

- Multiple comparison values can be set, and when the comparison value is reached, it will enter the task execution instruction of external events, with hardware output.
- Difference from single segment comparison: It is necessary to open a compare consistent interrupt, otherwise the instruction will report an error.
- As shown in the figure below, set the corresponding parameters as needed:



2-1-2-9. Ring counting【XJ_Counter_SetRing】

(1) Input variables

Input variables	Name	Data type	Effective range	Initial value	Description
Counter	High speed counter input port	XJ_COUNTER_REF	-	-	High speed counter, it is necessary to define the high-speed counting input terminal (see functional description)
xExcute	Trigger	BOOL	TRUE,FALSE	FALSE	Trigger
diMaxValue	Maximum value	DINT		0	Maximum value
diMinValue	Minimum value	DINT		0	Minimum value

(2) Output variables

Output variables	Name	Data type	Effective range	Initial value	Description
xDone	Completed flag	BOOL	TRUE,FALSE	FALSE	After completing the write, the flag is TRUE
xBusy	Running	BOOL	TRUE,FALSE	FALSE	
xError	Error flag	BOOL	TRUE,FALSE	FALSE	
eErrorID	Error type	HSIO_ERROR	-	0	

- The current count value needs to be within the set value range, otherwise the command will report an error. After the instruction is executed, no instruction can switch back to linear counting. If switching back to linear technology is required, the counting range needs to be changed to a linear counting range.

- The minimum value for linear counting is -2147483647, and the maximum value is 2147483646. For example, adding one signal input after increasing the count to 2147483646 will change to -2147483647, and then continue counting upwards. It will not reach the minimum value of double word signed numbers -2147483648 and the maximum value of 2147483647.

- There are two configuration methods for ring counting: firstly, when not logged in, in the "High speed IO" interface - "Counter parameter settings", you can click "Ring counting" and set the range. The second is to use the XJ_Counter_SetRing instruction setting, which can directly switch linear counting to circular counting during login.

2-1-2-10. Reset port of compare consistent output【XJ_ResetCmpOutput】

(1) Input variables

Input variables	Name	Data type	Effective range	Initial value	Description
Counter	High speed counter input port	XJ_COUNTER_REF	-	-	High speed counter, it is necessary to define the high-speed counting input terminal (see functional description)
xExcute	Trigger	BOOL	TRUE,FALSE	FALSE	Trigger

(2) Output variables

Output variables	Name	Data type	Effective range	Initial value	Description
xDone	Completed flag	BOOL	TRUE,FALSE	FALSE	After completing the setting, the flag is TRUE
xBusy	Running	BOOL	TRUE,FALSE	FALSE	
xError	Error flag	BOOL	TRUE,FALSE	FALSE	
eErrorID	Error type	HSIO_ERROR	-	0	

Release all compare consistent output ports. At this time, Y, which was originally set as a compare consistent output, can be used as a normal Y. As for whether the Y-point is used as a compare consistent output or a regular output, it depends on which port instruction is triggered first for the compare consistent output command and reset the compare consistent output command.

2-1-2-11. Clear the error【XJ_Counter_Reset】

Clear errors related to high-speed counting instructions.

(1) Input variables

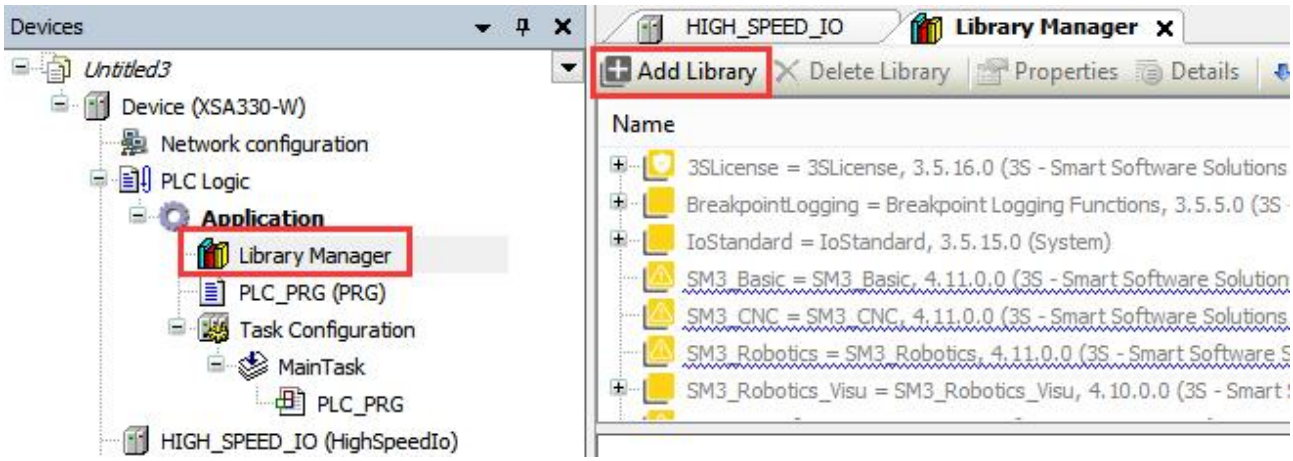
Input variables	Name	Data type	Effective range	Initial value	Description
Counter	High speed counter input port	XJ_COUNTER_REF	-	-	High speed counter, it is necessary to define the high-speed counting input terminal (see functional description)
xExcute	Trigger	BOOL	TRUE,FALSE	FALSE	Trigger

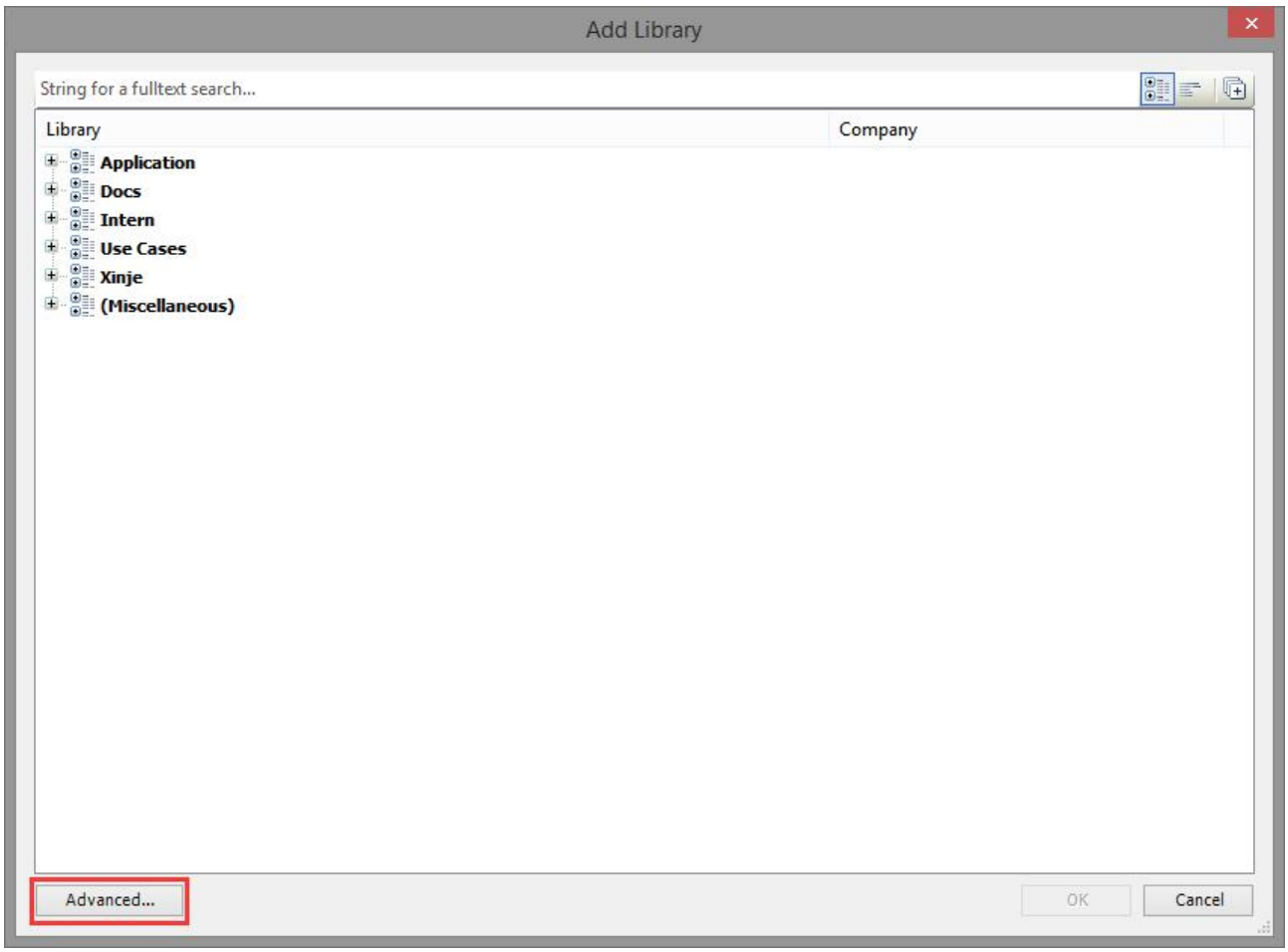
(2) Output variables

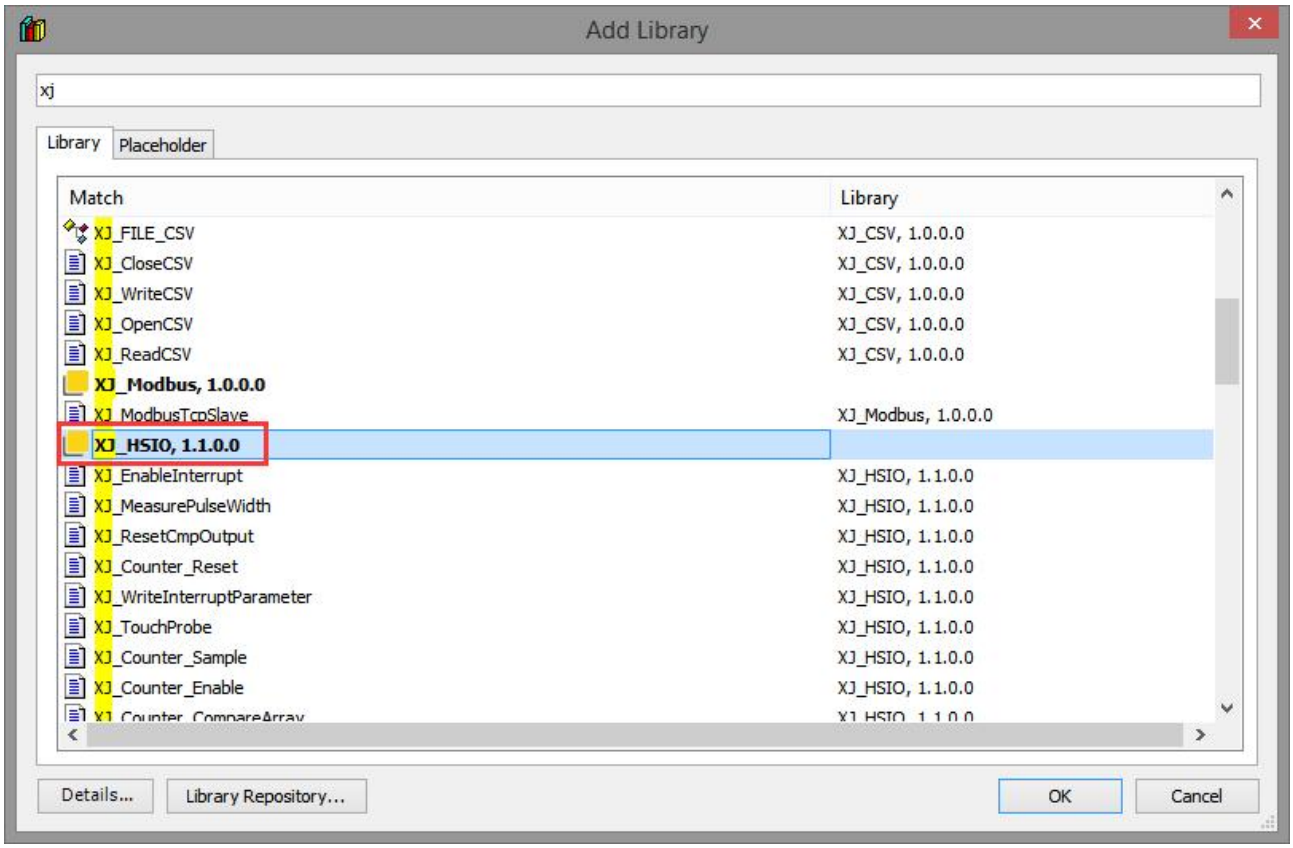
Output variables	Name	Data type	Effective range	Initial value	Description
xDone	Completed flag	BOOL	TRUE,FALSE	FALSE	After completing the setting, the flag is TRUE
xBusy	Running	BOOL	TRUE,FALSE	FALSE	
xError	Error flag	BOOL	TRUE,FALSE	FALSE	
eErrorID	Error type	HSIO_ERROR	-	0	

2-1-3. Parameter settings

Add Library File: Library Manager - Add Library - Advanced - Add XJ_HSIO.




Add Library
String for a fulltext search...
Library
Company
Application
Docs
Intern
Use Cases
Xinje
(Miscellaneous)
Advanced...
OK
Cancel


Add Library
xj
Library
Placeholder
Match
Library
XJ_FILE_CSV
XJ_CSV, 1.0.0.0
XJ_CloseCSV
XJ_CSV, 1.0.0.0
XJ_WriteCSV
XJ_CSV, 1.0.0.0
XJ_OpenCSV
XJ_CSV, 1.0.0.0
XJ_ReadCSV
XJ_CSV, 1.0.0.0
XJ_Modbus, 1.0.0.0
XJ_ModbusTcpSlave
XJ_Modbus, 1.0.0.0
XJ_HSIO, 1.1.0.0
XJ_EnableInterrupt
XJ_HSIO, 1.1.0.0
XJ_MeasurePulseWidth
XJ_HSIO, 1.1.0.0
XJ_ResetCmpOutput
XJ_HSIO, 1.1.0.0
XJ_Counter_Reset
XJ_HSIO, 1.1.0.0
XJ_WriteInterruptParameter
XJ_HSIO, 1.1.0.0
XJ_TouchProbe
XJ_HSIO, 1.1.0.0
XJ_Counter_Sample
XJ_HSIO, 1.1.0.0
XJ_Counter_Enable
XJ_HSIO, 1.1.0.0
Details...
Library Repository...
OK
Cancel

2-2. XS series high speed count instructions

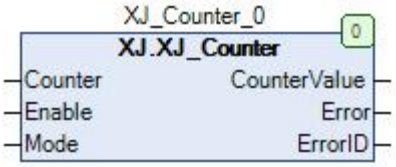
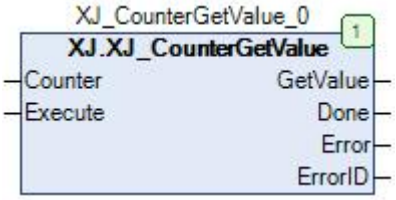
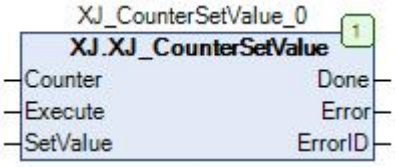
2-2-1. Function overview

The XS series PLC has a high-speed counting function, which enables the measurement of high-speed input signals such as measurement sensors and rotary encoders by selecting different counters. Its maximum measurement frequency can reach 200kHz.

Note: The instruction library in this chapter is only supported for firmware versions XSDH and XSLH series below V1.1.0.

2-2-2. Function block

2-2-2-1. Commands

Instruction	Name	Graph	ST language
XJ_Counter	High speed count		XJ_Counter(Counter:= , Enable:= , Mode:= , CounterValue=> , Error=> , ErrorID=>);
XJ_CounterGetValue	Read high speed counter		XJ_CounterGetValue(Counter:= , Execute:= , GetValue=> , Done=> , Error=> , ErrorID=>);
XJ_CounterSetValue	Write high speed counter		XJ_CounterSetValue(Counter:= , Execute:= , SetValue:= , Done=> , Error=> , ErrorID=>);

2-2-2-2. 【XJ_Counter】

(1) Input variables

Input variables	Name	Data type	Effective range	Initial value	Description
Counter	Counter	COUNTER_REF	-	-	High speed counter, which specifies the input terminal and initial value of high-speed counting
Enable	Enable	BOOL	TRUE,FALSE	FALSE	Normally open the enable to start counting
Mode	Count mode	Mode	AB_Mode，Single_Mode	FALSE	High speed count mode: MODE=XJ.AB_Mode is AB phase high speed count. MODE=XJ.Single_Mode is single phase high speed count.

(2) Output variables

Output variables	Name	Data type	Effective range	Initial value	Description
CounterValue	Count value	DINT	Data range	0	High speed count value
Error	Error flag	BOOL	TRUE,FALSE	FALSE	
ErrorID	Error type	UINT	-	0	

2-2-2-3. 【XJ_CounterGetValue】

(1) Input variables

Input variables	Name	Data type	Effective range	Initial value	Description
Counter	Counter	COUNTER_REF	-	-	High speed counter, which specifies the input terminal and initial value of high-speed counting
Execute	Enable	BOOL	TRUE,FALSE	FALSE	Triggered by the rising edge, reading the current high-speed count value

(2) Output variables

Output variables	Name	Data type	Effective range	Initial value	Description
GetValue	Read value	DINT	Data range	0	Present count value
Done	Completion	BOOL	TRUE,FALSE	FALSE	After completing the read, the flag is TRUE
Error	Error flag	BOOL	TRUE,FALSE	FALSE	
ErrorID	Error type	UINT	-	0	

2-2-2-4. 【XJ_CounterSetValue】

(1) Input variables

Input variables	Name	Data type	Effective range	Initial value	Description
Counter	Counter	COUNTER_REF	-	-	High speed counter, which specifies the input terminal and initial value of high-speed counting
Execute	Enable	BOOL	TRUE,FALSE	FALSE	Trigger by rising edge, write high-speed count value, write SetValue in CounterValue
SetValue	Write in value	DINT	Data range	0	Write high-speed counting set value

(2) Output variables

Output variables	Name	Data type	Effective range	Initial value	Description
Done	Completion	BOOL	TRUE,FALSE	FALSE	After completing the write, the flag bit is TRUE
Error	Error flag	BOOL	TRUE,FALSE	FALSE	
ErrorID	Error type	UINT	-	0	

Note: If the value displayed for ErrorID is 2, it is because the CounterID range is not between 0 and 3.

(3) Function description

- The high-speed counting function has three functional blocks, namely the high-speed counting function block, the read high-speed counting function block, and the write high-speed counting function block. XS3 series high-speed input can only receive differential signal (DIFF), but cannot receive Open collector signal (OC). Please be sure to select differential signal encoder. XSDH series high-speed input is to receive Open collector signal (OC).
- Counter is COUNTER_REF data type:
  COUNTER_REF specific description is as follows:

Member	Name	Data type	Effective range	Initial value	Description
CounterID	Counter terminal	INT	0,1,2,3	0	Select high-speed counter input port
CounterValue	Counter initial value	DINT	data range	0	Setting the initial value of the counter

- The XS3 and XSDH series high-speed counting functions have two modes, namely single phase incremental mode and AB phase mode.

(1) Mode= Single_Mode

In this mode, the input pulse signal is counted, and the count value increases with the rising edge of each pulse signal.

(2) Mode=AB_Mode

In this mode, the high-speed counting value is incremented or decremented based on the pulse signal (A phase and B phase) with a phase difference of 90°. The default counting mode is 4 times frequency.

- XS series high-speed counting input port allocation

XS3-26T4								
	Single phase mode				AB phase mode			
CounterID	0	1	2	3	0	1	2	3
Max frequency	200k	200k	200k	200k	200k	200k	200k	200k
X0+	U+				A+			
X0-	U-				A-			
X1+					B+			
X1-					B-			
X2								
X3+		U+				A+		
X1-		U-				A-		
X4+						B+		
X2-						B-		
X5								
X6+			U+				A+	
X6-			U-				A-	
X7+							B+	
X7-							B-	
X10								
X11+				U+				A+
X11-				U-				A-
X12+								B+
X12-								B-
X13								

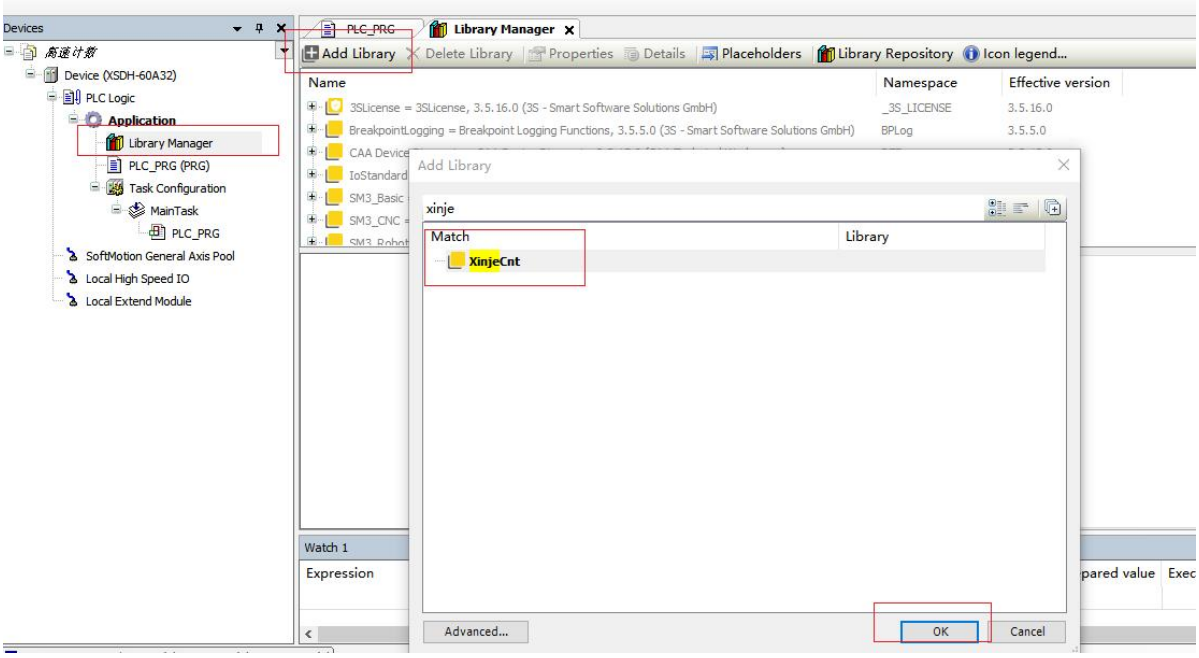
XSDH-60A32-E								
	Single phase mode				AB phase mode			
CounterID	0	1	2	3	0	1	2	3
Max frequency	200k	200k	200k	200k	100k	100k	100k	100k
X0	U				A			
X1					B			
X2								
X3		U				A		
X4						B		
X5								
X6			U				A	
X7							B	
X10								
X11				U				A
X12								B
X13								

XSLH-30A32								
	Single phase mode				AB phase mode			
CounterID	0	1	2	3	0	1	2	3
Max frequency	200k	200k	80k	80k	100k	100k	50k	50k
X0+	U+				A+			
X0-	U-				A-			
X1+					B+			
X1-					B-			
X2								
X3+		U+				A+		
X1-		U-				A-		
X4+						B+		
X2-						B-		
X5								
X6			U				A	
X7							B	
X10								
X11				U				A
X12								B
X13								
X14								
X15								

2-2-3. Parameter configuration

Add library file:

Add "XinjeCnt" in the "Library Manager", and after adding it, you can use the high-speed counting function.



2-2-4. Application example

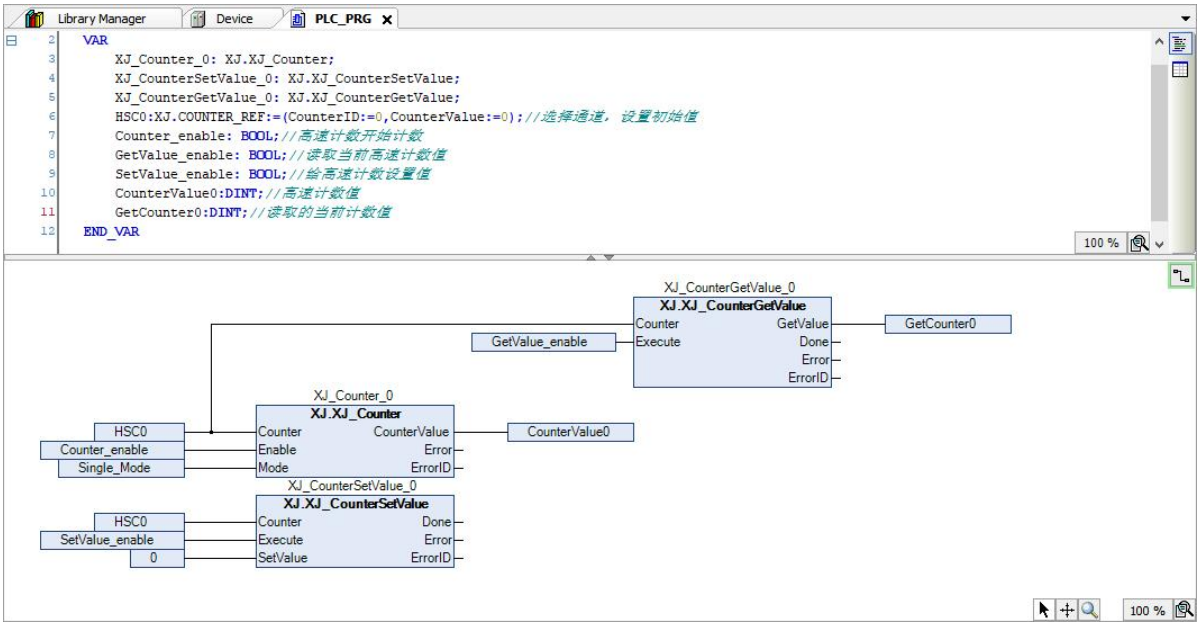
Example 1: Use the first channel of high-speed count and read the current count value in the count to modify the current high-speed count value.

Program operation:

(1) Install the required libraries according to the steps in sections 2-1-3.

(2) Write a high-speed counting program.

Programming: Use the function blocks "XJ.XJ_Counter", "XJ.XJ_CounterGetValue", and "XJ.XJ_CounterSetValue" to set the high-speed counting port, high-speed counting mode, and high-speed counting value used in the program.



2-3. External interrupt and compare consistent interrupt instructions

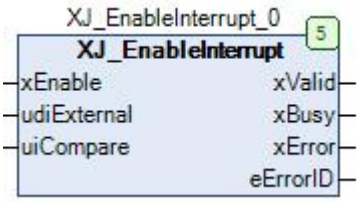
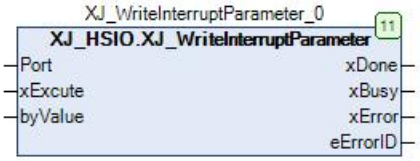
2-3-1. Function overview

The XSA series PLC supports X-terminal interrupt, and the same terminal supports rising and falling edge interrupts. In Codesys, interrupts are used through external events in the task type. Like X2R_TRIG represents X2 rising edge interrupt, X2F_TRIG represents the falling edge interrupt, and the number and type of interrupts supported by each model can be found in the "External event" option.

Note: XSDH and XSLH models currently do not support this instruction library for external interrupts. Please refer to the "XS Series PLCopen Standard Controller User Manual [Software Chapter]" manual for external interrupts.

2-3-2. Function block

2-3-2-1. Instruction format

Command	Name	Graph	ST language
XJ_EnableInterrupt	Open external interrupts and compare consistent interrupts		XJ_EnableInterrupt(xEnable:= , udiExternal:= , uiCompare:= , xValid=> , xBusy=> , xError=> , eErrorID=>);
XJ_WriteInterruptParameter	Interrupt parameter writing		XJ_WriteInterruptParameter(Port:= , xExcute:= , byValue:= , xDone=> , xBusy=> , xError=> , eErrorID=>);

2-3-2-2. Open external interrupts and compare consistent interrupts【XJ_EnableInterrupt】

(1) Input variables

Input variables	Name	Data type	Effective range	Initial value	Description
xEnable	Enable	BOOL	TRUE,FALSE	FALSE	Enable
udiExternal	Open external input interrupt	UINT		0	
uiCompare	Open compare consistent interrupt	UINT		0	For example, 4 means CounterID:=2

(2) Output variables

Output variables	Name	Data type	Effective range	Initial value	Description
xValid	Interruption take effect	BOOL	TRUE,FALSE	FALSE	After the interrupt takes effect, the flag bit is TRUE
xBusy	Running	BOOL	TRUE,FALSE	FALSE	
xError	Error flag	BOOL	TRUE,FALSE	FALSE	

eErrorID	Error type	HSIO_ERROR	-	0	

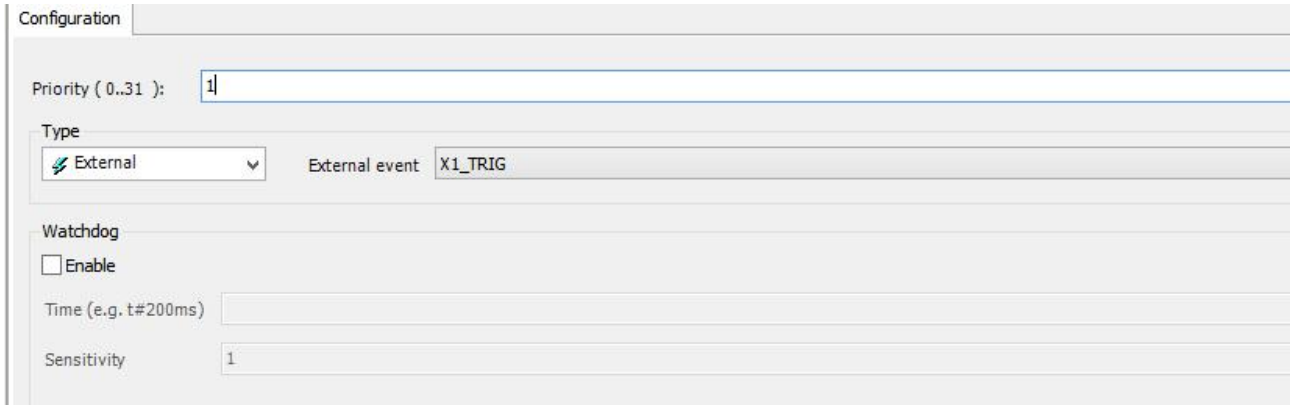
- **Open external input interrupt:**

① Select external interrupt terminal, such as X1

② Open XJ_EnableInterrupt to add interrupt task, where 1 represents X0, 2 represents X1, 3 represents X0, X1, and so on. As shown in the following figure, external interrupt tasks are executed when the rising edge of X1 is effective.

That is, the udiExternal opens the external output interrupt and displays it as a binary address, and then gives a value. For example, if you fill in binary 2#11 or decimal 10#3, then open X0 and X1. In the "Hardware Parameter Configuration", you also need to check the required X0 and X1, which can be selected as "edge" or "double edge". If there is a signal from X0 and X1, the task configured as "X0" or "X1" will execute the program.

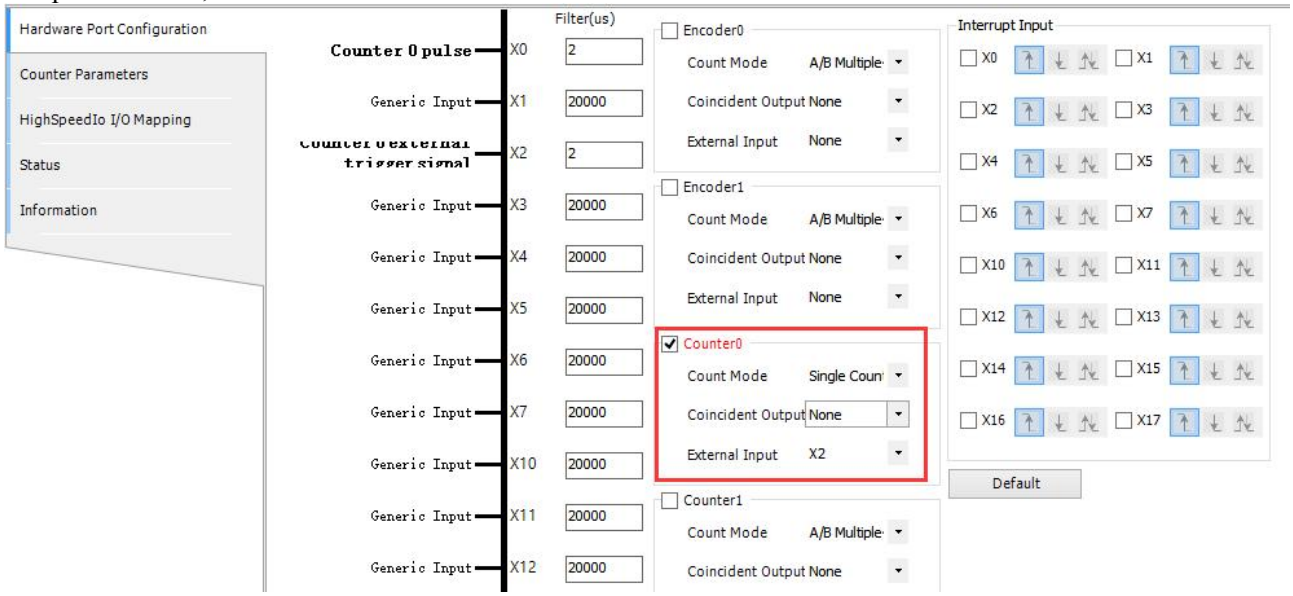
XJ_EnableInterrupt_0	XJ_EnableInterrupt				打开外部中断和比较一致中断
xEnable	BOOL	FALSE			使能
udiExternal	UINT	2			打开外部输入中断
uiCompare	UINT	4			打开比较一致中断
xValid	BOOL	FALSE			中断生效
xBusy	BOOL	FALSE			正在运行
xError	BOOL	FALSE			错误标志
eErrorID	HSIO_ERROR	ERR_OK			错误代码



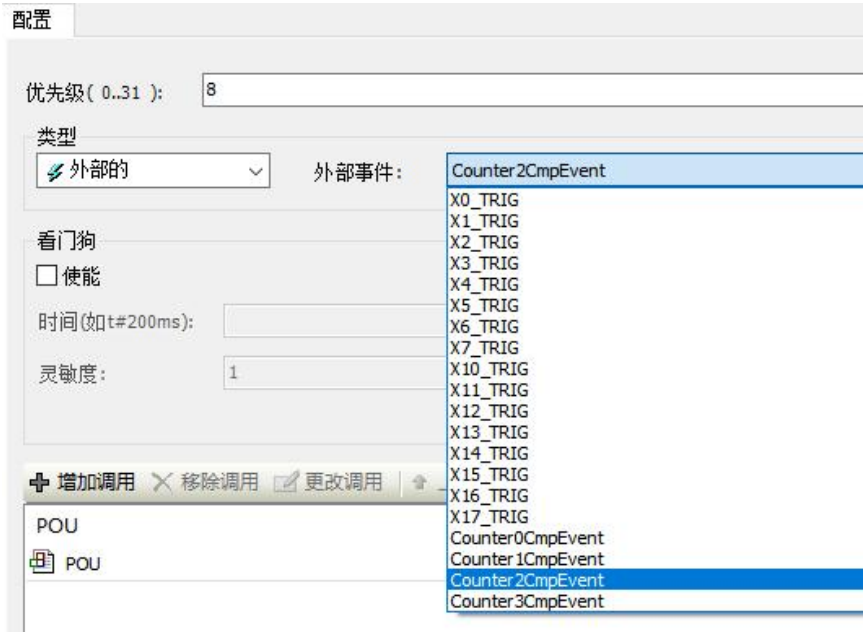
The port needs to be configured on both the interface and instructions to take effect.

- **Open compare consistent interrupt:**

Set the counter ID that needs to be opened, such as 3 indicating CounterID:=0 and CounterID:=1. In the case, M210 and CounterID:=2 are used, so opening a compare consistent interrupt to write 4. After reaching the comparison value, it will enter the task execution instruction of external events.



XJ_EnableInterrupt_0	XJ_EnableInterrupt		打开外部中断和比较一致中断
xEnable	BOOL	FALSE	使能
udiExternal	UINT	0	打开外部输入中断
uiCompare	UINT	4	打开比较一致中断
xValid	BOOL	FALSE	中断生效
xBusy	BOOL	FALSE	正在运行
xError	BOOL	FALSE	错误标志
eErrorID	HSIO_ERROR	ERR_OK	错误代码



2-3-2-3. Write in interrupt parameters【XJ_WriteInterruptParameter】

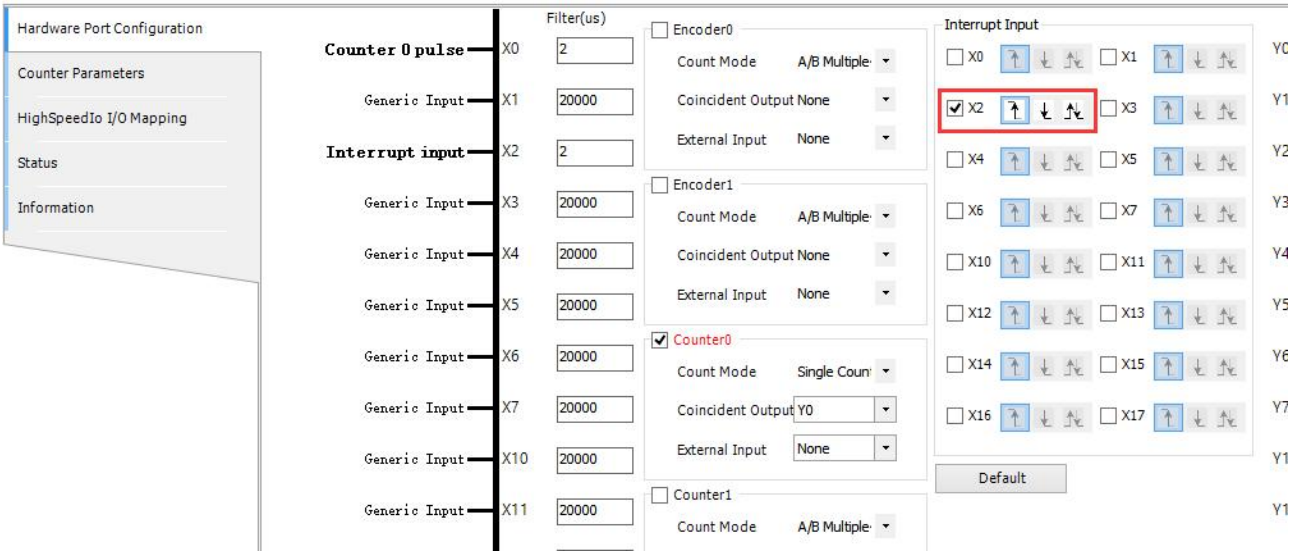
(1) Input variables

Input variables	Name	Data type	Effective range	Initial value	Description
Port	Port no.	UINT		0	Binary, such as 2#11 representing X0 and X1
xExcute	Trigger	BOOL	TRUE,FALSE	FALSE	
byValue	Value	BYTE		0	

(2) Output variables

Output variables	Name	Data type	Effective range	Initial value	Description
xDone	Completed flag	BOOL	TRUE,FALSE	FALSE	After completing the setting, the flag bit is TRUE
xBusy	Running	BOOL	TRUE,FALSE	FALSE	
xError	Error flag	BOOL	TRUE,FALSE	FALSE	
eErrorID	Error type	HSIO_ERROR	-	0	

External interrupt parameters can be configured on the interface or modified using the command.



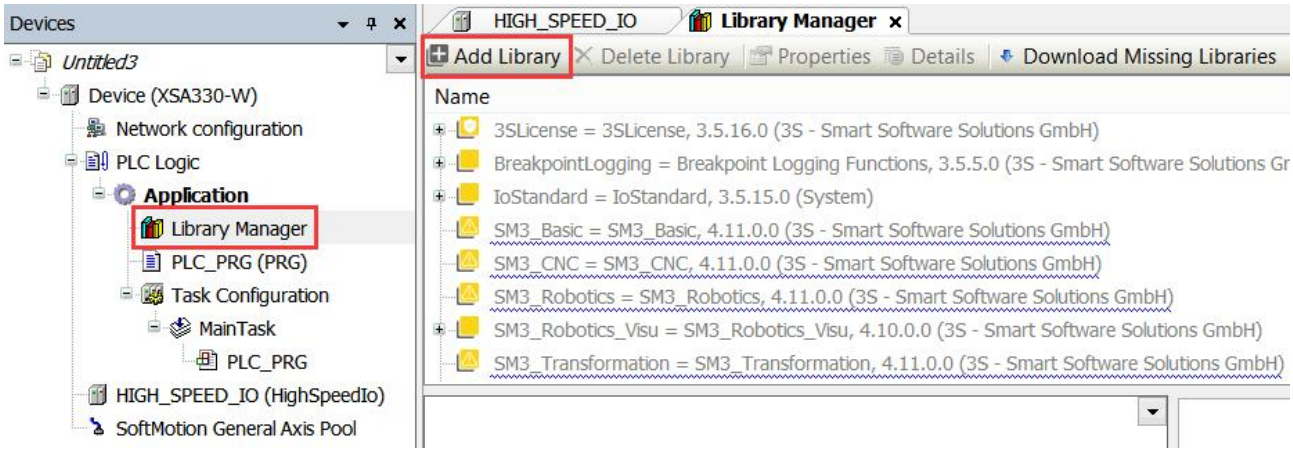
XJ_WriteInterruptParameter_0	XJ_HSIO.XJ_WriteInterruptParameter				中断参数写入
Port	UINT	4			端口号二进制如：2#11代表X0、X1
xExcute	BOOL	FALSE			触发
byValue	BYTE	0			值(0:上升沿，1为下降沿，2为上升沿+下降沿)
xDone	BOOL	FALSE			完成标志
xBusy	BOOL	FALSE			正在运行
xError	BOOL	FALSE			错误代码
eErrorID	HSIO_ERROR	ERR_OK			错误标志

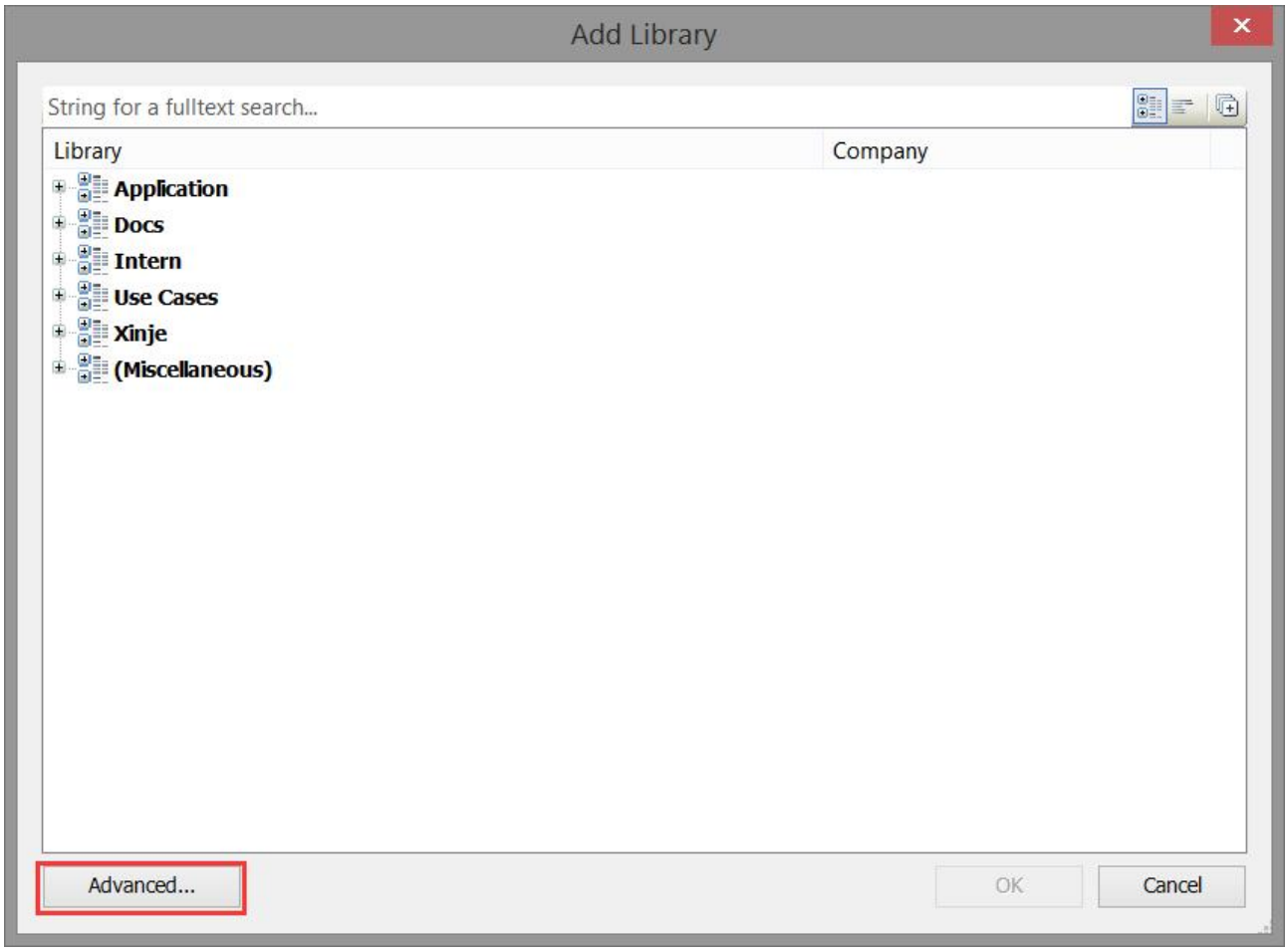
- XJ_WriteInterruptParameter executed before XJ_EnableInterrupt, then XJ_WriteInterruptParameter parameters are valid.
- XJ_EnableInterrupt executed before XJ_WriteInterruptParameter, background interrupt parameters are valid. XJ_WriteInterruptParameter executed again, then XJ_WriteInterruptParameter parameters are valid.
- When executing this command, it is also necessary to open external input interrupts.

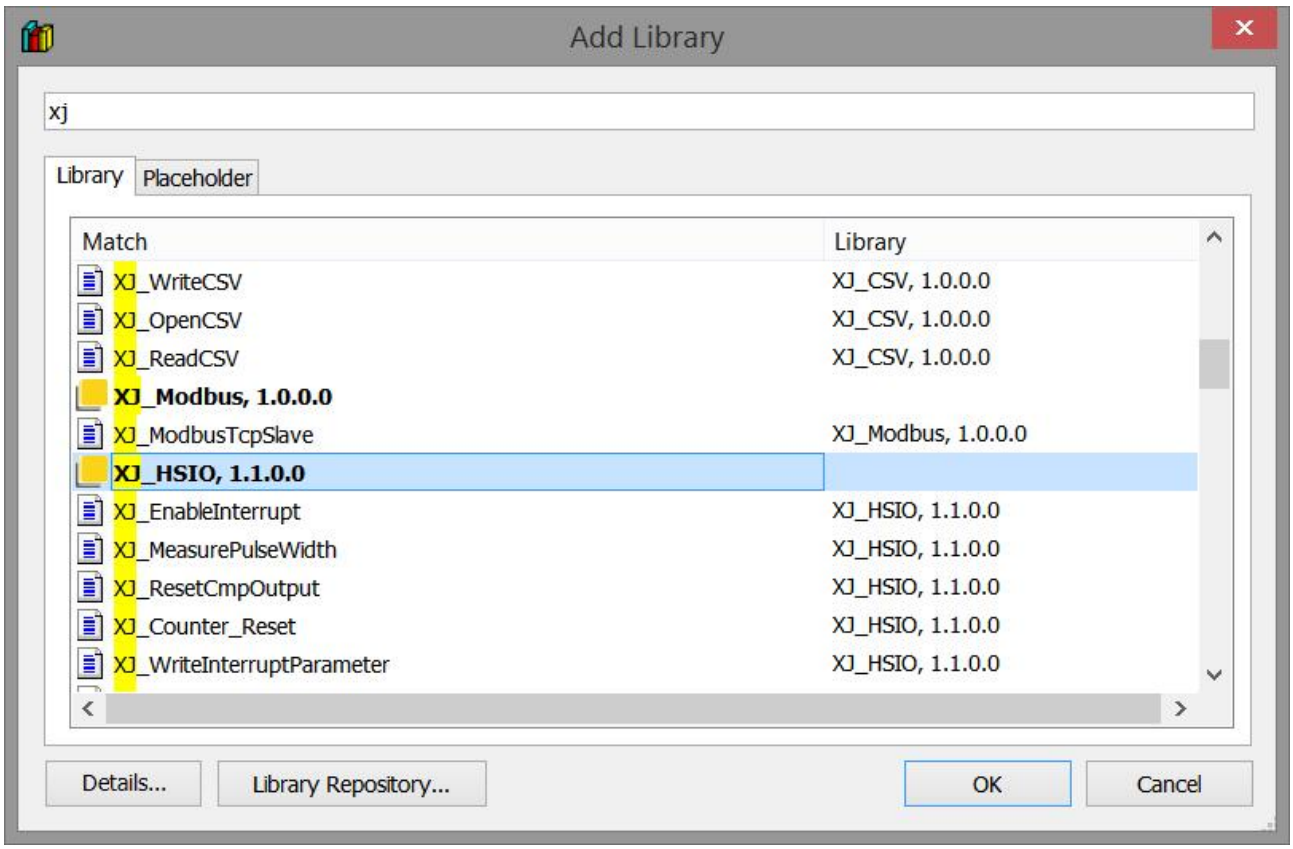
XJ_EnableInterrupt_0	XJ_EnableInterrupt				打开外部中断和比较一致中断
xEnable	BOOL	FALSE			使能
udiExternal	UINT	4			打开外部输入中断
uiCompare	UINT	4			打开比较一致中断
xValid	BOOL	FALSE			中断生效
xBusy	BOOL	FALSE			正在运行
xError	BOOL	FALSE			错误标志
eErrorID	HSIO_ERROR	ERR_OK			错误代码

2-3-3. Parameter configuration

Add Library File: Library Manager - Add Library - Advanced - Add XJ_HSIO.

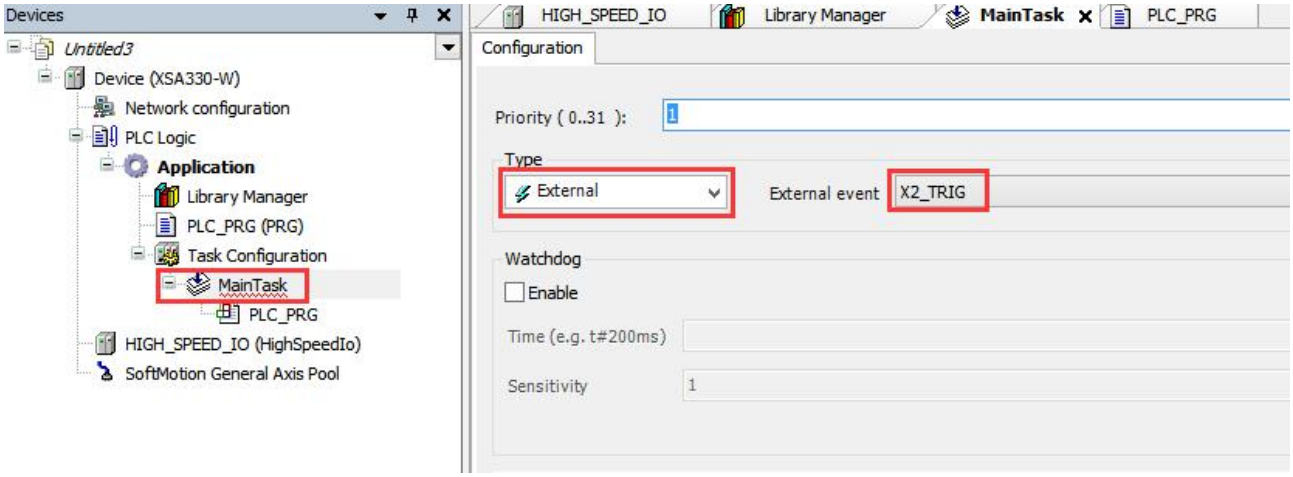




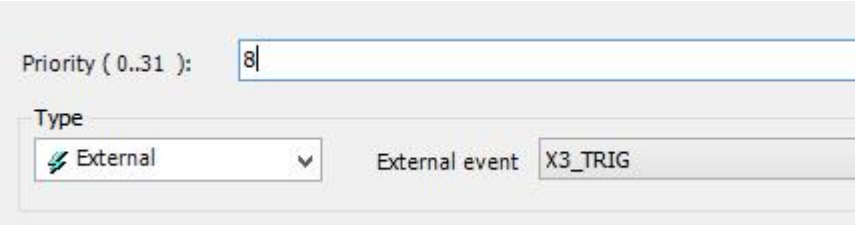


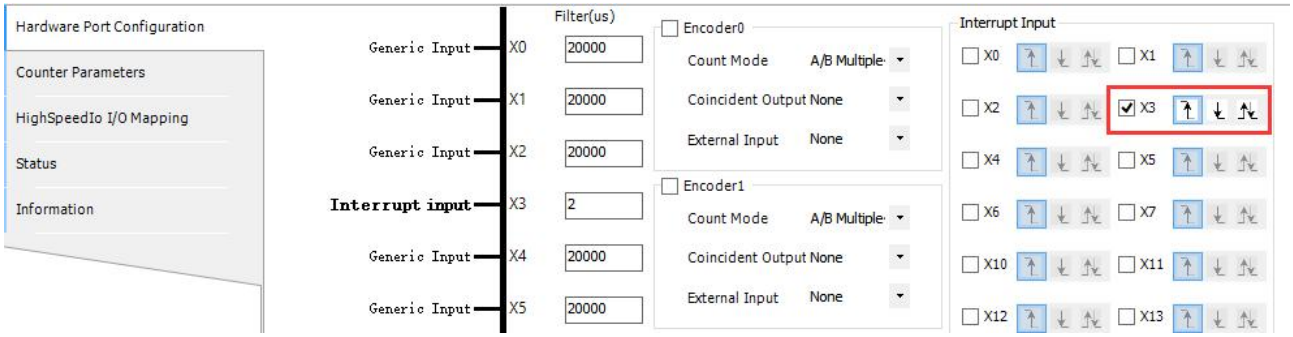
2-3-4. Application example

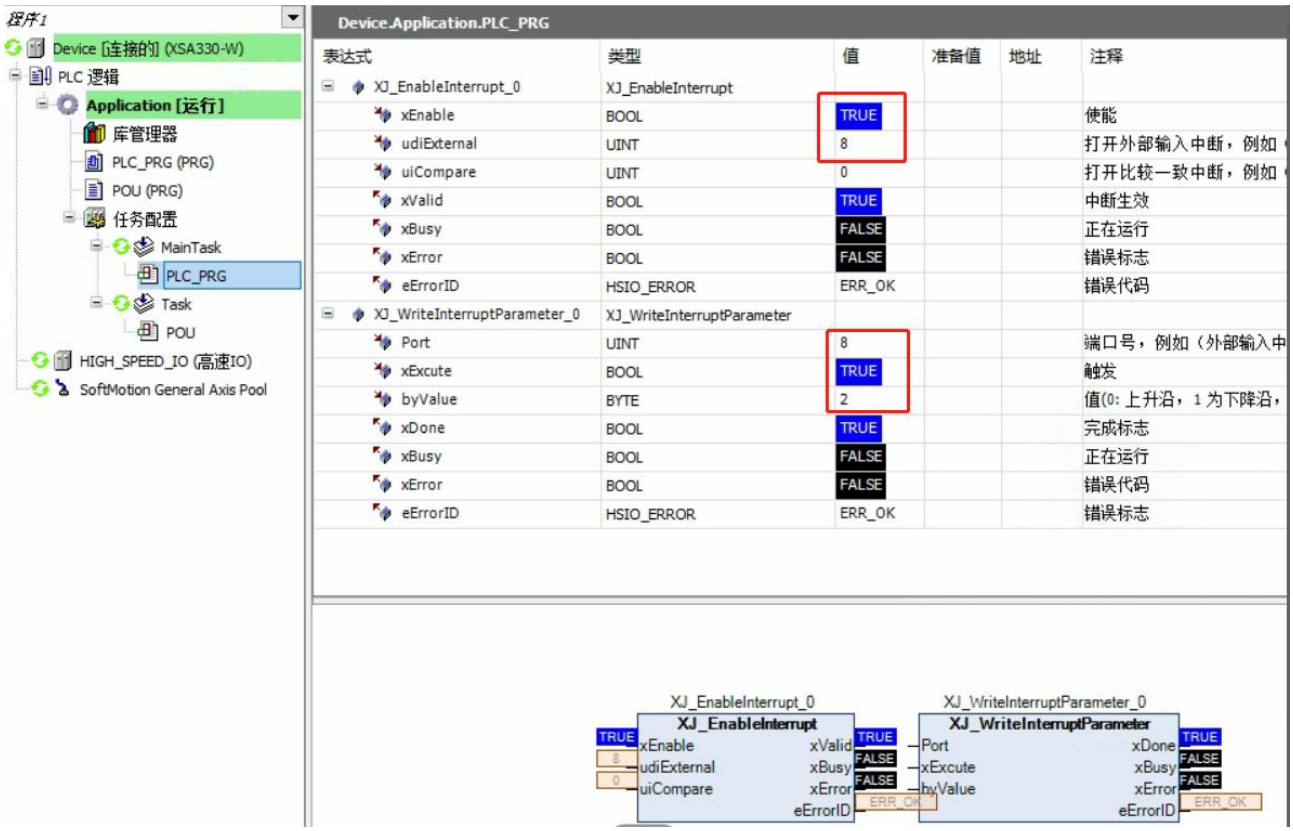
Double click “Task”, the type to "External" in the pop-up interface - use terminal X for external interrupts, and you can also set the priority of external interrupt events.



Example 1: Use the 【XJ_EnnableInterrupt】【XJ_WriteInterruptParameter】instructions. Set X3 as an external interrupt input, take its bilateral edge signal, which can be configured in the hardware parameter configuration interface or using XJ_WriteInterruptParameter instruction. Execute the self adding 1 instruction in the POU program under another task (configured as external, X3_TRIG) once the edge signal of X3 is given. The parameter configuration and instructions are shown in the following figure.


Priority (0..31):
8
Type
External
External event
X3_TRIG


Hardware Port Configuration
Counter Parameters
HighSpeedIo I/O Mapping
Status
Information
Filter(us)
Generic Input X0 20000
Generic Input X1 20000
Generic Input X2 20000
Interrupt input X3 2
Generic Input X4 20000
Generic Input X5 20000
Encoder0
Count Mode A/B Multiple
Coincident Output None
External Input None
Encoder1
Count Mode A/B Multiple
Coincident Output None
External Input None
Interrupt Input
X0 X1 X2 X3 X4 X5 X6 X7 X10 X11 X12 X13

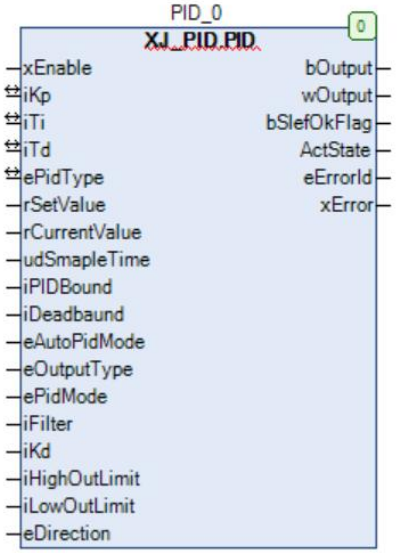

程序1
Device [连接的] (XSA330-W)
PLC 逻辑
Application [运行]
库管理器
PLC_PRG (PRG)
POU (PRG)
任务配置
MainTask
PLC_PRG
Task
POU
HIGH_SPEED_IO (高速IO)
SoftMotion General Axis Pool
Device.Application.PLC_PRG
表达式 类型 值 准备值 地址 注释
XJ_EnableInterrupt_0 XJ_EnableInterrupt
xEnable BOOL TRUE 使能
udiExternal UINT 8 打开外部输入中断，例如
uiCompare UINT 0 打开比较一致中断，例如
xValid BOOL TRUE 中断生效
xBusy BOOL FALSE 正在运行
xError BOOL FALSE 错误标志
eErrorID HSIO_ERROR ERR_OK 错误代码
XJ_WriteInterruptParameter_0 XJ_WriteInterruptParameter
Port UINT 8 端口号，例如（外部输入中
xExcute BOOL TRUE 触发
byValue BYTE 2 值(0: 上升沿，1 为下降沿，
xDone BOOL TRUE 完成标志
xBusy BOOL FALSE 正在运行
xError BOOL FALSE 错误代码
eErrorID HSIO_ERROR ERR_OK 错误标志
XJ_EnableInterrupt_0
XJ_EnableInterrupt
xEnable
udiExternal
uiCompare
xValid
xBusy
xError
eErrorID
XJ_WriteInterruptParameter_0
XJ_WriteInterruptParameter
Port
xExcute
byValue
xDone
xBusy
xError
eErrorID


Device.Application.POU
表达式 类型 值
A INT 1
1 A 1 :=A 1 +1; RETURN

2-4. PID instructions

Attention: The PID function block is copyrighted and can only be used on Xinje's PLC! Simulation not available!

2-4-1. Command format

Command	Name	Graph	ST language
XJ_PID.PID	PID instruction		XJ_PID.PID(xEnable:= , iKp:= , iTi:= , iTd:= , ePidType:= , rSetValue:= , rCurrentValue:= , udSmapleTime:= , iPIDBound:= , iDeadbaund:= , eAutoPidMode:= , eOutputType:= , ePidMode:= , iFilter:= , iKd:= , iHighOutLimit:= , iLowOutLimit:= , eDirection:= , bOutput=> , wOutput=> , bSlefOkFlag=> , ActState=> , eErrorId=> , xError=>);

2-4-2. Related variables

VAR_INPUT	Name	Data type	Effective range	Initial value	Description
xEnable	Enable	BOOL	TRUE/FALSE	FALSE	Must be set to TRUE to activate the processing of the function block
iKp	Proportional gain	INT	0~32767	0	Proportional gain
iTi	Integral time	INT	0~32767	0	Integral time (*100ms)
iTd	Differential time	INT	0~32767	0	Differential time (*10ms)
ePidType	Tuning mode	PIDTYPE	-	-	0: Manual mode; 1: Oscillation Self-tuning mode
rSetValue	Target value	REAL	Data range	0	*Target setting value
rCurrentValue	Feedback value	REAL	Data range	0	*Present feedback value
udSmapleTime	Sample time	UDINT	Data range	1000	* Sampling time (ms) Suggest setting between 1000 and 2000

VAR_INPUT	Name	Data type	Effective range	Initial value	Description
iPIDBound	Operational range	INT	0~32767	0	PID operation range
iDeadbaund	Control Deadband	INT	0~32767	0	PID control dead zone
eAutoPidMode	Self-tuning PID control mode	PID CONTRLMODE	-	-	Self-tuning PID mode 0: PID control; 1: PI control; 2: P control
eOutputType	Output mode	OUTPUTTYPE	-	-	0: IO channel output; 1: Digital output
ePidMode	Control mode	PIDMODE	-	-	Control mode: general mode, advanced mode
iFilter	Input filtering constant	INT	0~100	0	Input filtering constant, available in advanced mode from 0 to 100
iKd	Differential gain	INT	-	50	Differential gain, default is 50, available in advanced mode
iHighOutLimit	Output upper limit setting value	INT	0~32767	0	Output upper limit setting value (default: 4095)
iLowOutLimit	Output lower limit setting value	INT	0~32767	0	Output lower limit setting value (default: 0)
eDirection	Reverse action/forward action	DIRECTIONPARAMETER	-	-	ATHWART: reverse action (heating); POSITIVE: Positive action (cooling)
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
bOutpu	Boolean output	BOOL	TRUE/FALSE	FALSE	IO output is TRUE
wOutput	Digital output	INT	TRUE/FALSE	FALSE	Digital output is true
bSlefOkFlag	Self-tuning completion flag	BOOL	TRUE/FALSE	FALSE	The completion of Self-tuning is TRUE
ActState	Present mode	AUTOPIDSTATE	-	-	Present mode
eErrorId	Error code	ERRID_PID	-	-	When normal, the value is 0. When an exception occurs, an error code is output
xError	Error flag	BOOL	TRUE/FALSE	FALSE	True when an exception occurs

2-4-3. Function description

- Direction of action

Positive action: The action where the output value MV of the operation increases with the increase of the measured value PV, usually used for cooling control.
Reverse action: An action in which the output value MV decreases as the measured value PV increases, usually used for heating control.

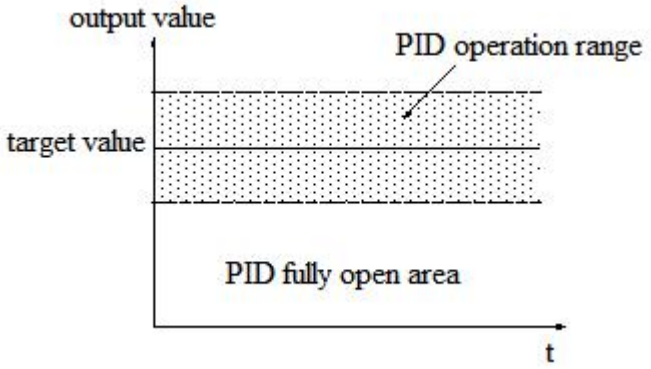
- Advanced mode setting: enable advanced mode, and users can set parameters such as filtering time, differential gain, and output upper and lower limits.

- Sample time

The system samples the current value at a certain time interval and compares it with the output value. This time interval is the sampling time T. When DA is output, T has no limit. When the port outputs, T must be greater than 1 PLC program scan cycle. The value of T should be within the range of 100-1000 PLC scanning cycles.

- PID operation range

When the system is running, it is initially in the PID fully open stage, which approaches the target value at the fastest speed (default is 4095). When it reaches the operating range of the PID, the parameters Kp, TI, and TD begin to take control. As shown in the following figure:

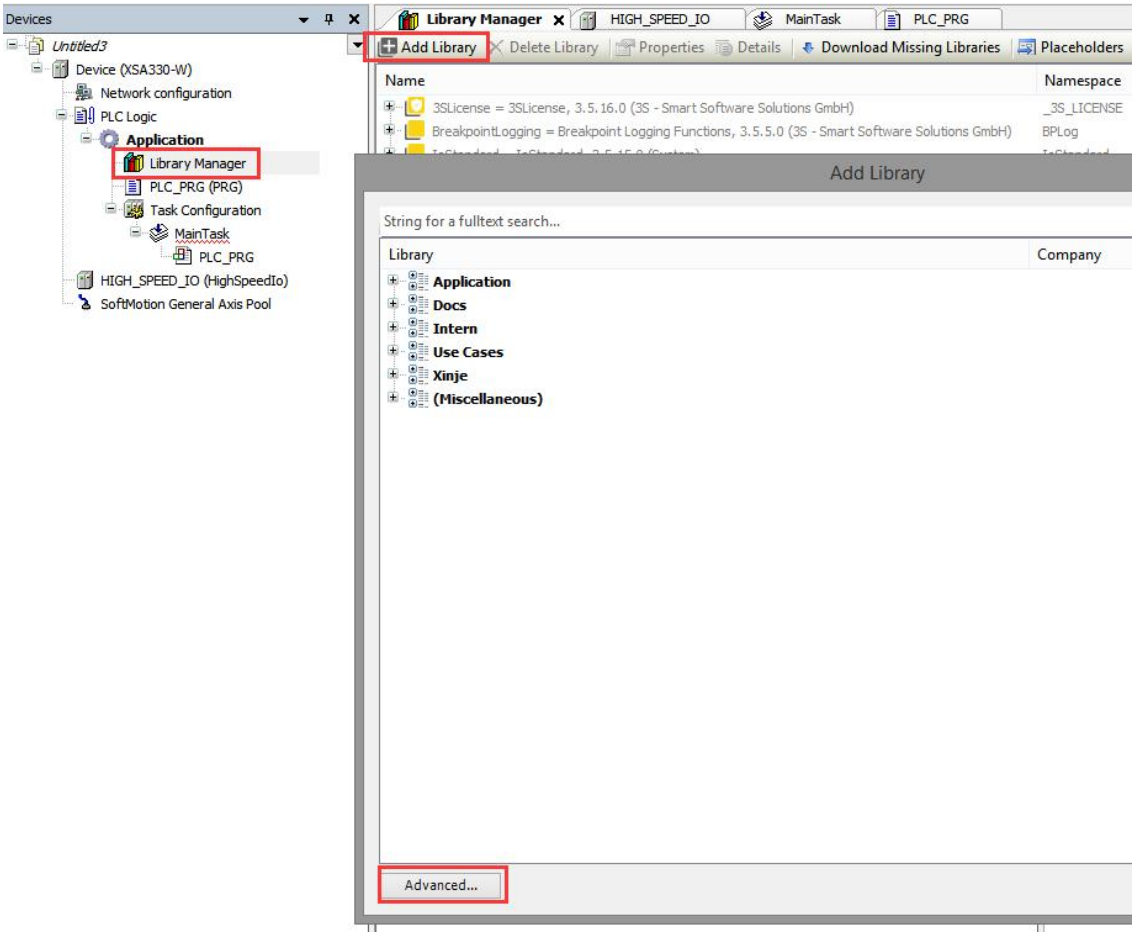


When the user is not clear about the specific set value of PID parameters, the Self-tuning mode can be selected to enable the system to automatically find the best control parameters (proportional gain Kp, integration time Ti, differential time TD).

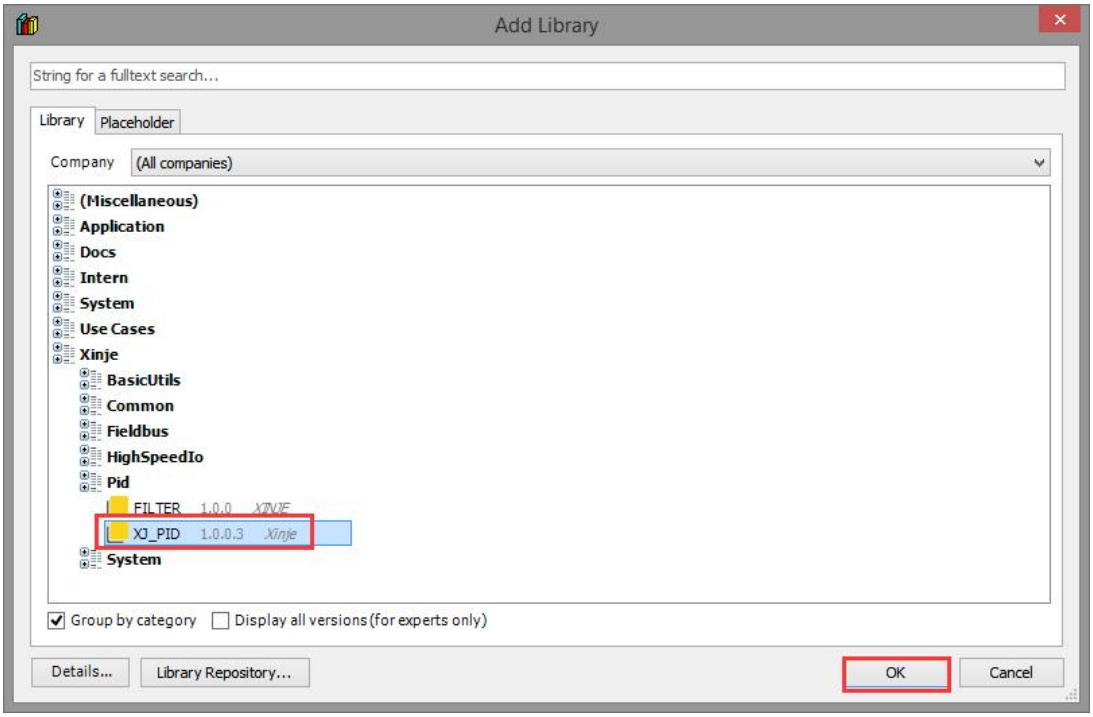
- Control object applicable to Self-tuning mode: temperature and pressure. Unsuitable control objects: liquid level and flow rate, etc.
- Self-tuning is the process of extracting PID parameters. Sometimes Self-tuning can not find the best parameters once, and it requires multiple Self-tuning. Oscillation in the process is normal. After Self-tuning is completed and the best parameters are found, it is necessary to switch to manual PID. If the control object is unstable during the manual PID process and cannot be controlled at a constant target value, it may be caused by poor parameter adjustment, and it is necessary to adjust the PID parameters again to achieve stable control.
- At the beginning of Self-tuning in critical oscillation method, the user needs to set the PID control cycle (sampling time) in advance. Reference value: Generally, slow response systems can be set to 1000ms, while fast response systems can be set to 10ms-100ms.
- By using the critical oscillation method for Self-tuning, the system can start from any state. For temperature control objects, the current measured temperature does not need to be consistent with the ambient temperature. It can be below or above the target temperature.
- For Self-tuning mode, it is necessary to set "ePidType" to TRUE, and select the pid mode (p, pi, pid). When the setting is completed, switch xEnable status to TRUE, PLC will enter the Self-tuning status, ActState status is AutoPidBusy, when "ActState" is AutoPidDone, it means Self-tuning is successful, and "bSlefOkFlag" is true. After maintaining a cycle, PID status will automatically switch to CommonPID, at this time, PID controls the control system in manual mode with Self-tuning parameters.
- Switching of output values. If the output value required by the user is a digital quantity, set the eOutputType value to 1. If the output value required by the user is a switching quantity, set the eOutputType value to 0.

2-4-4. Application example

Double click Library manager → add library →advanced.

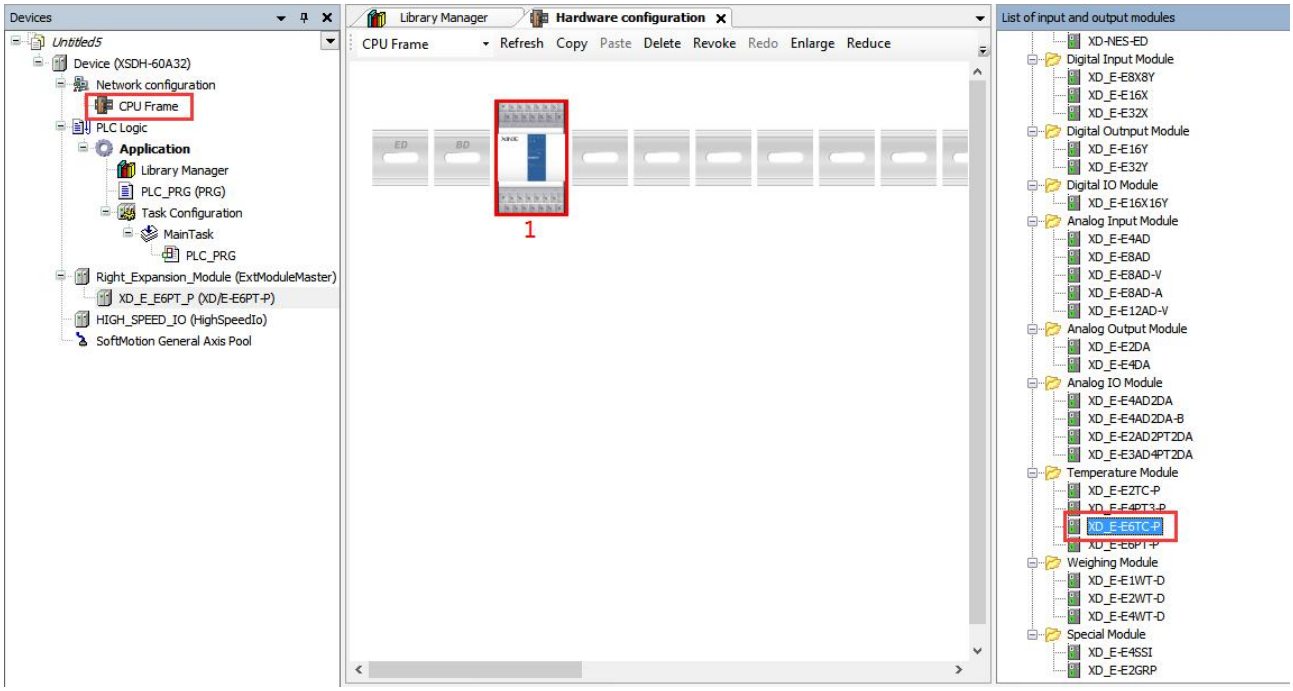


Add XJ_PID.

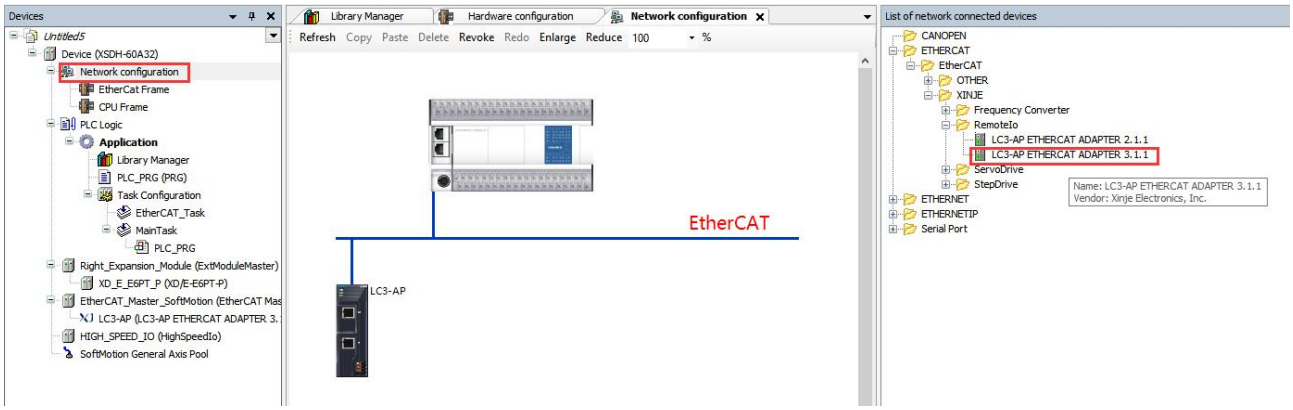


Add temperature expansion module.

- Double click CPU Frame, select position 1, then double click XD-E6TC-P.

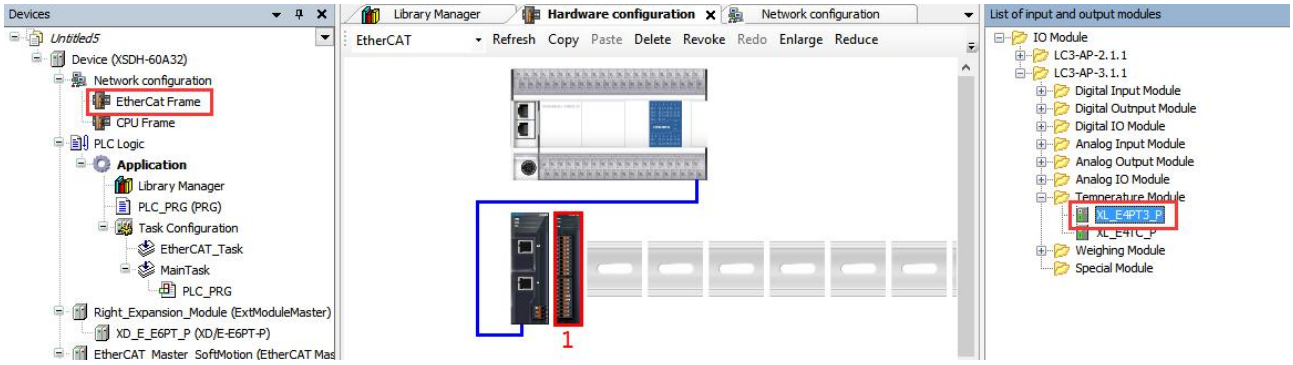


- Remote IO module: double click Network configuration, double click LC3-AP Ethercat adapter 3.1.1.

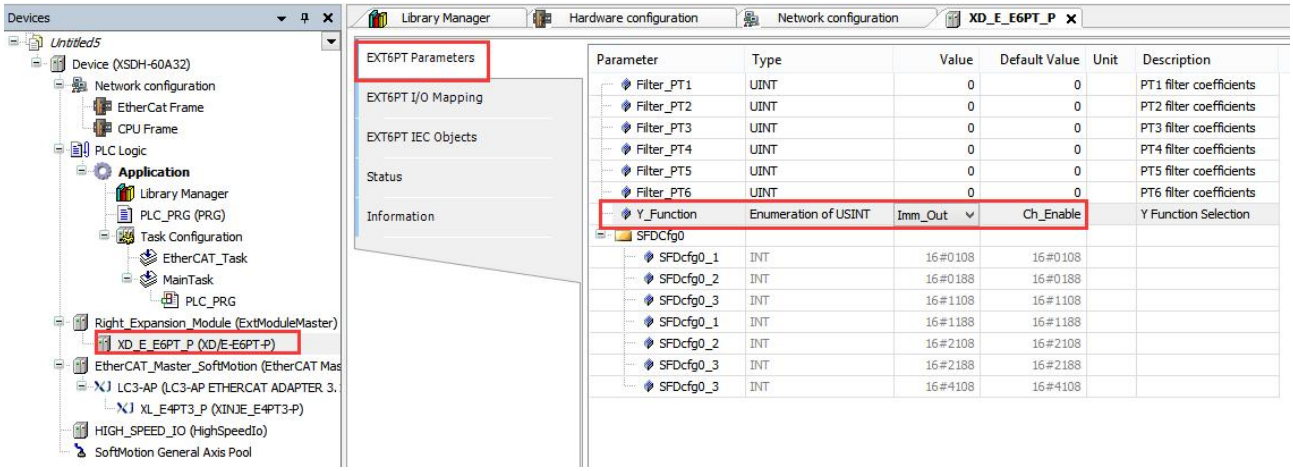


Note: Currently, only LC3-AP V3.1 and above support EC_From TO instruction.

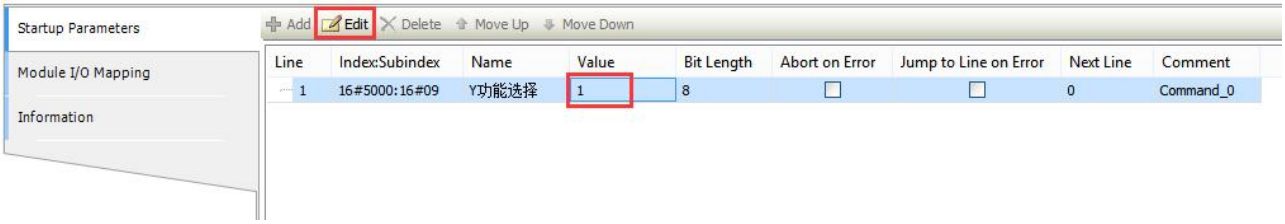
Double click EtherCat Frame, select position 1, double click XL-E4PT3-P.



If a switch quantity control relay is needed, the module on the main body needs to set the Y_Function to 'Imm Out'.

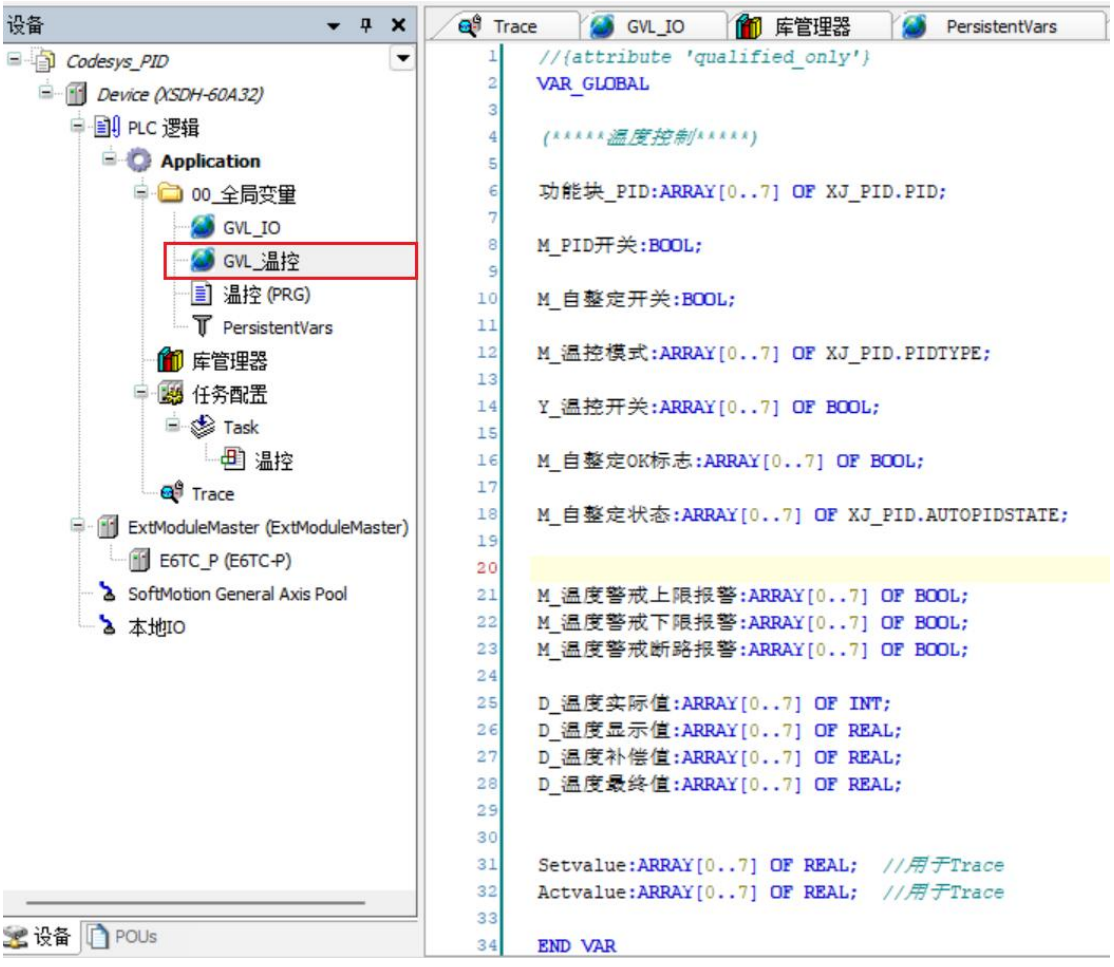


The module on remote IO needs to add the corresponding startup parameter to the default value of 1.

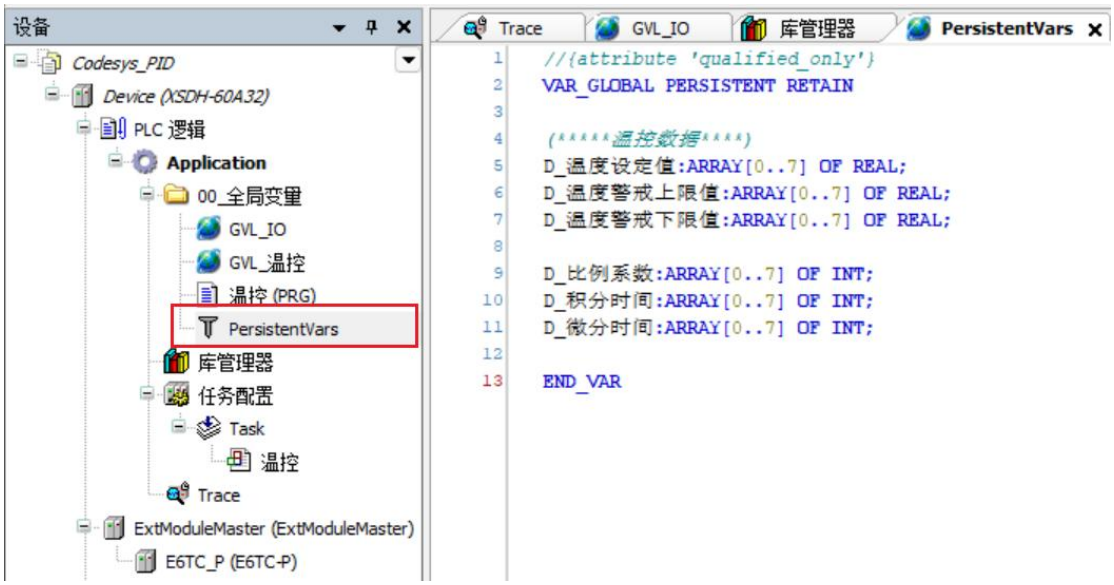


After modifying the module, the parameters need to be powered on again to take effect.

Define the required Global variable.



8. Define the required power-off retention variables so that data will not be lost after the PLC restarts.



9. Make the PID program.

① Convert the actual temperature value obtained from the temperature module into the correct temperature display value. For example, if the transmitted data is 289, it needs to be divided by 10 to convert it into a Floating-point number of 28.9, representing 28.9°, which is convenient for subsequent personnel to understand and maintain. Due to the possibility of different on-site conditions requiring additional compensation, an additional compensation value is added to obtain a final value. Use this final value to adjust the PID.

```
//注：在手动模式下，写入参数需要将PID功能块使能关掉。

(*****温度实际值转换*****)
FOR a:=0 TO 2 BY 1 DO

    D_温度显示值[a] :=TO_REAL(D_温度实际值[a]) / 10;
    D_温度最终值[a] := D_温度显示值[a] + D_温度补偿值[a];

    Setvalue[a]:=D_温度设定值[a];  //用于Trace采集
    Actvalue[a]:=D_温度最终值[a];  //用于Trace采集

END_FOR
```

② Temperature alarm: When the actual temperature is too high or too low, as well as when there is an open circuit, an alarm is required.

```
(*****温度超限报警*****)
FOR b:=0 TO 2 BY 1 DO
    //温度低下限报警
    IF D_温度最终值[b] < D_温度警戒下限值[b] THEN
        M_温度警戒下限报警[b]:= TRUE;
    ELSE
        M_温度警戒下限报警[b]:= FALSE;
    END_IF

    //温度超上限报警
    IF D_温度最终值[b] > D_温度警戒上限值[b] THEN
        M_温度警戒上限报警[b]:= TRUE;
    ELSE
        M_温度警戒上限报警[b]:= FALSE;
    END_IF

    //温度断路报警
    IF D_温度显示值[b] = -0.1 THEN
        M_温度警戒断路报警[b]:= TRUE;
    ELSE
        M_温度警戒断路报警[b]:= FALSE;
    END_IF

END_FOR
```

③ One key Self-tuning, when the "M_ Self-tuning switch" is on (it can be instantaneous, and it is forbidden to be always on), all PID function blocks are in Self-tuning mode, which is convenient for batch debugging.

```
(*****启用自整定模式*****)
IF M_自整定开关 THEN
    M_温控模式[0]:= 1;
    M_温控模式[1]:= 1;
    M_温控模式[2]:= 1;
END_IF
```

④ To enable Enable the PID function block, simply set ON M_PID. (Note: When modifying PID parameters, be sure to turn off the PID enable!)

```
IF NOT M_温度警戒上限报警[0] THEN
  功能块_PID[0](
    xEnable:= M_PID开关,  //启用PID功能块
    iKp:= D_比例系数[0],  //比例系数，范围0~32767，单位%
    iTi:= D_积分时间[0],  //积分时间，范围0~32767，单位100ms
    iTd:= D_微分时间[0],  //微分时间，范围0~32767,单位10ms
    rSetValue:=  D_温度设定值[0],  //要维持的温度值
    rCurrentValue:= D_温度最终值[0],  //温度传感器检测到的实际值
    ePidType:= M_温控模式[0], //0:手动模式；1：自整定模式
    udSmapleTime:= 1000,  //采样时间，建议设置在1000~2000
    iPIDBound:= ,  //PID运算范围，0~32767
    iDeadbaund:= ,  //PID控制死区，0~32767
    eAutoPidMode:= 0,//自整定模式，0：PID控制；1：PI控制；2：P控制
    eOutputType:= XJ_PID.OUTPUTTYPE.BoolType,  //输出模式，开关量或模拟量
    ePidMode:= 0,  //控制模式：通用模式，高级模式
    iFilter:= ,  //采样温度滤波参数，0~100，单位%，高级模式模式下可用
    iKd:= ,  //微分增益，默认50，高级模式模式下可用
    iHighOutLimit:= ,  //输出上限设定值，0~32767，高级模式模式下可用
    iLowOutLimit:= ,  //输出下限设定值，0~32767，高级模式模式下可用
    eDirection:= 0,  //0：逆动作（加热）；1：正动作（冷却）
    bOutput=> Y_温控开关[0],  //开关量输出
    wOutput=> ,  //模拟量输出
    bSlefOkFlag=> M_自整定OK标志[0] ,
    ActState=> M_自整定状态[0],  //自整定状态
    eErrorId=> ,
    xError=>  );
END_IF
```

⑤ The PID Self-tuning time is about 15~25 minutes. After the Self-tuning is completed, iKp, iTi and iTd will all have values. ePidype will automatically switch to "ManualType", the bSelfOKFlag flag will be on, and ActState will display "AutoPIDrea" to facilitate the next Self-tuning.

```
(*****PID*****)
PID[0](
    xEnable TRUE := M_PID开关 TRUE,   //启用PID功能块
    iKp 31929 := D_比例系数[0] 31929,   //比例系数，范围0~32767，单位%
    iTi 1636 := D_积分时间[0] 1636,   //积分时间，范围0~32767，单位100ms
    iTd 4091 := D_微分时间[0] 4091,   //微分时间，范围0~32767,单位10ms
    rSetValue 85 := D_温度设定值[0] 85,   //要维持的温度值
    rCurrentValue 91.2 := D_温度最终值[0] 91.2,   //传感器检测到的实际值
    ePidType 0 := M_温控模式[0] ManulType, //0:手动模式；1：自整定模式
    udSmapleTime 1000 := 1000,   //采样时间
    iPIDBound:= ,   //PID运算范围，0~32767
    iDeadbaund:= ,   //PID控制死区，0~32767
    eAutoPidMode PIDMode := 0,//自整定模式，0：PID控制；1：PI控制；2：P控制
    eOutputType BoolType := XJ_PID.OUTPUTTYPE.BoolType,   //输出模式，开关量或模拟量
    ePidMode Common := 0,   //控制模式：通用模式，高级模式
    iFilter:= ,   //采样温度滤波参数，0~100，单位%，高级模式模式下可用
    iKd:= ,   //微分增益，默认50，高级模式模式下可用
    iHighOutLimit:= ,   //输出上限设定值，0~32767，高级模式模式下可用
    iLowOutLimit:= ,   //输出下限设定值，0~32767，高级模式模式下可用
    eDirection Athwart := 0,   //0：逆动作（加热）；1：正动作（冷却）
    bOutput FALSE => Y_温控开关[0] FALSE,   //开关量输出
    wOutput=> ,   //模拟量输出
    bSlefOkFlag TRUE => M_自整定OK标志[0] TRUE ,
    ActState AutoPIDRea => M_自整定状态[0] AutoPIDRea,   //自整定状态
    eErrorId=> ,
    xError=>  ,
    PreI_Item 0 => PreI_Item[0] 0,
    PreD_Item 293 => PreD_Item[0] 293,
    PrePID_Item 0 => PrePID_Item[0] 0,
    Overshoot 7 => Overshoot[0] 7)
    ;
```

10. The PID accuracy is around 2% of the set temperature. After actual measurement, the set temperature is 85 °C. With PID adjustment, the temperature has been maintained at 85±1 °C

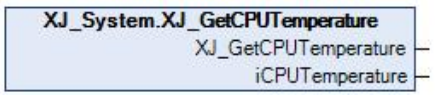
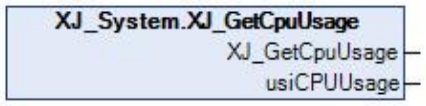
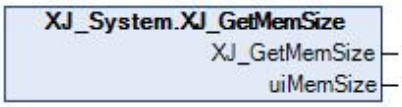
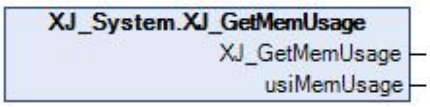
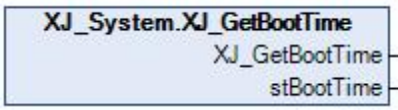
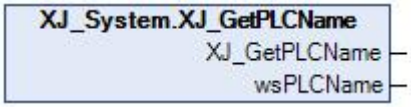
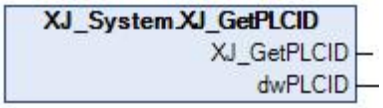
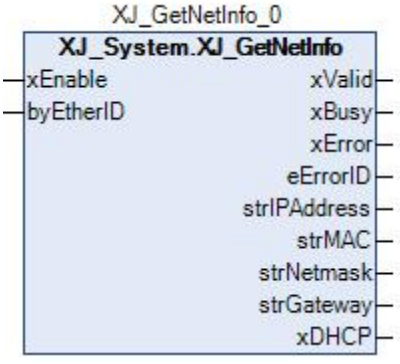
2-5. System library

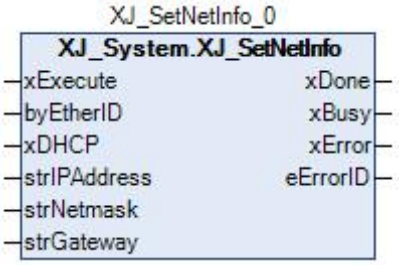
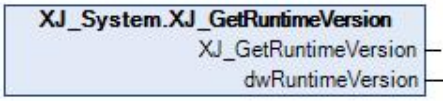
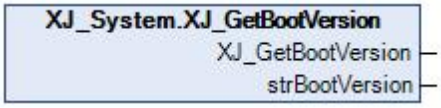
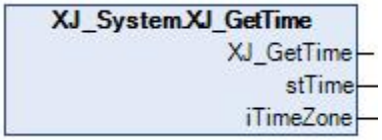
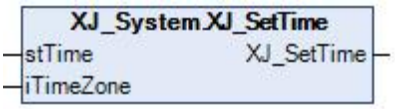
2-5-1. Function overview

Users can read or write system parameters of the controller through instructions. Currently only XSA330-W supports this function.

2-5-2. Function block introduction

2-5-2-1. Command format

Command	Name	Graph representation	ST language
XJ_GetCPUFrequency	CPU dominant frequency		XJ_GetCPUFrequency(uiCPUFrequency=>)
XJ_GetCPUTemperature	CPU temperature		XJ_GetCPUTemperature(iCPUTemperature=>)
XJ_GetCPUUsage	CPU occupancy		XJ_GetCpuUsage(usiCPUUsage=>)
XJ_GetMemSize	Memory size		XJ_GetMemSize(uiMemSize=>)
XJ_GetMemUsage	Memory occupancy		XJ_GetMemUsage(usiMemUsage=>)
XJ_GetBootTime	On time		XJ_GetBootTime(stBootTime=>)
XJ_GetPLCName	PLC name		XJ_GetPLCName(wsPLCName=>)
XJ_GetPLCID	PLCID		XJ_GetPLCID(dwPLCID=>)
XJ_GetPLCVersion	PLC firmware version		XJ_GetPlcVersion(strPLCVersion=>)
XJ_GetNetInfo	Obtain network port information		XJ_GetNetInfo(xEnable:= , byEtherID:= , xValid=> , xBusy=> , xError=> , eErrorID=> , strIPAddress=> , strMAC=> , strNetmask=> , strGateway=> , xDHCP=>);

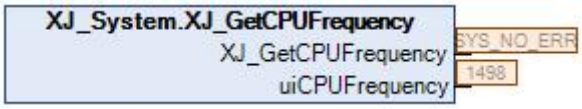
Command	Name	Graph representation	ST language
XJ_SetNetInfo	Set network port information		XJ_SetNetInfo(xExecute:= , byEtherID:= , xDHCP:= , strIPAddress:= , strNetmask:= , strGateway:= , xDone=> , xBusy=> , xError=> , eErrorID=>);
XJ_GetRuntimeVersion	Runtime version		XJ_GetRuntimeVersion(dwRuntimeVersion=>)
XJ_GetBootVersion	BOOT version		XJ_GetBootVersion(strBootVersion=>)
XJ_GetTime	Obtain date and time		XJ_GetTime(stTime=> , iTimeZone=>)
XJ_SetTime	Set date and time		XJ_SetTime(stTime:= , iTimeZone:=)

2-5-2-2. CPU dominant frequency【XJ_GetCPUFrequency】

(1) Output variables

Output variables	Name	Data type	Effective range	Initial value	Description
GetCPUFrequency	Error code	SYS_XJ_ERR	-	0	Error code
CPUFrequency	CPU dominant frequency	UINT	-	0	Unit is M

The execution results are as follows:

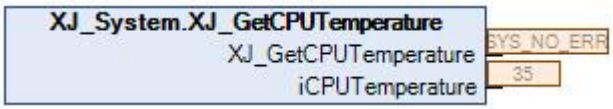


2-5-2-3. CPU temperature【XJ_GetCPUTemperature】

(1) Output variables

Output variables	Name	Data type	Effective range	Initial value	Description
GetCPUTemperature	Error code	SYS_XJ_ERR	-	0	Error code
CPUTemperature	CPU temperature	INT	-	0	CPU temperature

The execution results are as follows:

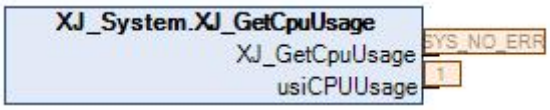


2-5-2-4. CPU occupancy【XJ_GetCPUUsage】

(1) Output variables

Output variables	Name	Data type	Effective range	Initial value	Description
GetCPUUsage	Error code	SYS_XJ_ERR	-	0	Error code
CPUUsage	CPU occupancy	USINT	-	0	Return percentage, 0~100

The execution results are as follows:

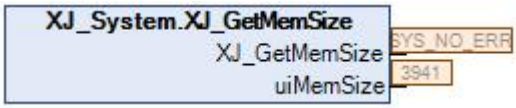


2-5-2-5. Memory size【XJ_GetMemSize】

(1) Output variables

Output variables	Name	Data type	Effective range	Initial value	Description
GetMemSize	Error code	SYS_XJ_ERR	-	0	Error code
MemSize	Memory size	UINT	-	0	Unit is MB

The execution results are as follows:

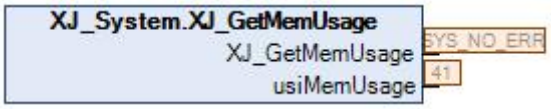


2-5-2-6. Memory occupancy【XJ_GetMemUsage】

(1) Output variables

Output variables	Name	Data type	Effective range	Initial value	Description
GetMemUsage	Error code	SYS_XJ_ERR	-	0	Error code
MemUsage	Memory occupancy	USINT	-	0	Return percentage, 0~100

The execution results are as follows:

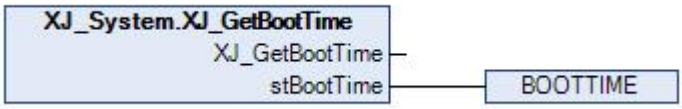


2-5-2-7. On time【XJ_GetBootTime】

(1) Output variables

Output variables	Name	Data type	Effective range	Initial value	Description
GetBootTime	Error code	SYS_XJ_ERR	-	0	Error code
BootTime	On time	SYS_XJ_TIME	-	0	structural morphology

The execution results are as follows:




```
PROGRAM POU
VAR
    // 获取网口信息
    XJ_GetNetInfo_0: XJ_System.XJ_GetNetInfo;
    PLCID: DWORD;
    BOOTTIME: XJ_SYSTEM.SYS_XJ_TIME;
    AA: BOOL;
    NETID: BYTE;
    XJ_SetNetInfo_0: XJ_System.XJ_SetNetInfo;
    runtimeversion: DWORD;
    STTIME: XJ_SYSTEM.SYS_XJ_TIME;
    timezone: INT;
    SetTIME: XJ_System.SYS_XJ_TIME;
    itimezone: INT;
END_VAR
```

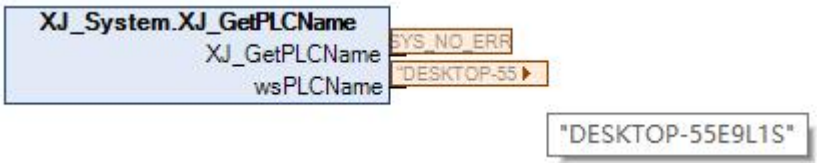
BOOTTIME	XJ_SYSTEM.SYS_XJ_TIME				
Year	UINT	2022			年
Month	UINT	11			月
Day	UINT	19			日
Hour	UINT	1			时
Minute	UINT	55			分
Second	UINT	43			秒
Milliseconds	UINT	0			微妙
DayOfWeek	UINT	6			周

2-5-2-8. PLC name【XJ_GetPLCName】

(1) Output variables

Output variables	Name	Data type	Effective range	Initial value	Description
GetPLCName	Error code	SYS_XJ_ERR	-	0	Error code
PLCName	PLC name	WSTRING	-	0	PLC name

The execution results are as follows:

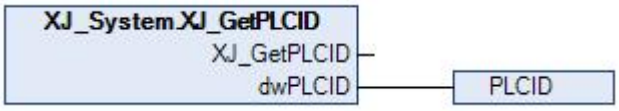


2-5-2-9. PLCID【XJ_GetPLCID】

(1) Output variables

Output variables	Name	Data type	Effective range	Initial value	Description
GetPLCID	Error code	SYS_XJ_ERR	-	0	Error code
PLCID	PLCID	DWORD	-	0	PLCID information, hexadecimal display

The execution result is as follows, and the PLCID information is displayed in hexadecimal:




```
PROGRAM POU
VAR
    // 获取网口信息
    XJ_GetNetInfo_0: XJ_System.XJ_GetNetInfo;
    PLCID: DWORD;
    BOOTTIME: XJ_SYSTEM.SYS_XJ_TIME;
    AA: BOOL;
    NETID: BYTE;
    XJ_SetNetInfo_0: XJ_System.XJ_SetNetInfo;
    runtimeversion: DWORD;
    STTIME: XJ_SYSTEM.SYS_XJ_TIME;
    timezone: INT;
    SetTIME: XJ_System.SYS_XJ_TIME;
    itimezone: INT;
END_VAR
```

表达式	类型	值	准
XJ_GetNetInfo_0	XJ_System.XJ_GetNetInfo		
PLCID	DWORD	16#17070200	
BOOTTIME	XJ_SYSTEM.SYS_XJ_TIME		
AA	BOOL	FALSE	
NETID	BYTE	16#00	
XJ_SetNetInfo_0	XJ_System.XJ_SetNetInfo		
runtimeversion	DWORD	16#03050F28	
STTIME	XJ_SYSTEM.SYS_XJ_TIME		
timezone	INT	16#0000	
SetTIME	XJ_System.SYS_XJ_TIME		
itimezone	INT	16#0000	

2-5-2-10. Firmware version【XJ_GetPLCVersion】

(1) Output variables

Output variables	Name	Data type	Effective range	Initial value	Description
GetPLCVersion	Error code	SYS_XJ_ERR	-	0	Error code
PLCVersion	PLC firmware version	STRING	-	0	PLC firmware version

The execution result is as follows:

2-5-2-11. Obtain network port information【XJ_GetNetInfo】

(1) Input variables

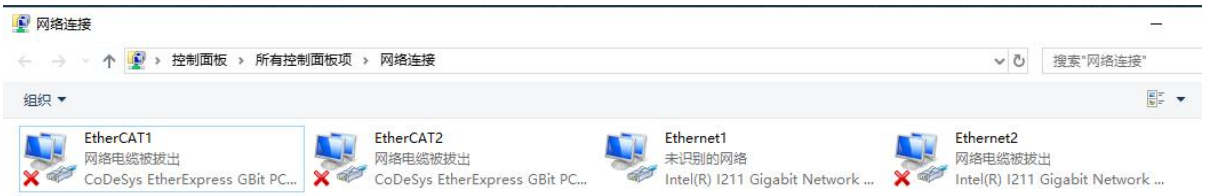
Input variables	Name	Data type	Effective range	Initial value	Description
Enable	Enable	BOOL	TRUE,FALSE	FALSE	Enable
EtherID	Network port ID	BYTE	-	0	Corresponding network port no. on the display screen

(2) Output variables

Output variables	Name	Data type	Effective range	Initial value	Description
Valid	Output valid	BOOL	TRUE,FALSE	FALSE	Output valid
Busy	Executing	BOOL	TRUE,FALSE	FALSE	Executing
Error	Error flag	BOOL	TRUE,FALSE	FALSE	Error flag
ErrorID	Error code	SYS_XJ_ERR	-	0	Error code
IPAddress	IP address	STRING	-	0	IP address
MAC	MAC address	STRING	-	0	MAC address
Netmask	Subnet address	STRING	-	0	Subnet address
Gateway	Gateway Information	STRING	-	0	Gateway Information
DHCP	DHCP	BOOL	TRUE，FALSE	FALSE	Is it automatic property

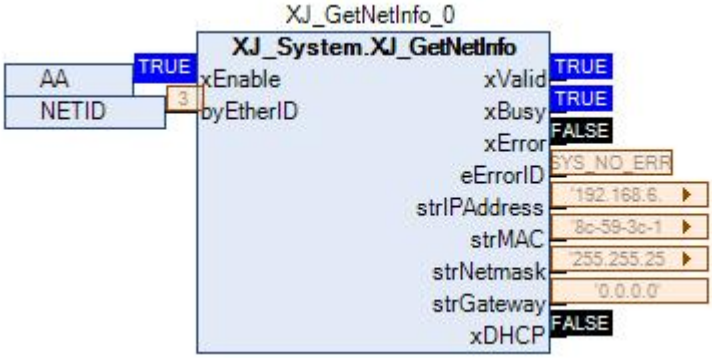
	properties				acquisition

The network port IDs of the industrial computer are arranged in 0-3 order from left to right in the network adapter, as shown in the figure Ethernet1, and the IDs are written in 2.




```
PROGRAM POU
VAR
    // 获取网口信息
    XJ_GetNetInfo_0: XJ_System.XJ_GetNetInfo;
    PLCID: DWORD;
    BOOTTIME: XJ_SYSTEM.SYS_XJ_TIME;
    AA: BOOL;
    NETID: BYTE;
END_VAR
```

The execution result is as follows：



XJ_GetNetInfo_0	XJ_System.XJ_GetNetInfo			获取网口信息
xEnable	BOOL	TRUE		功能块使能
byEtherID	BYTE	3		网口Id
xValid	BOOL	TRUE		输出有效
xBusy	BOOL	TRUE		正在执行
xError	BOOL	FALSE		错误
eErrorID	SYS_XJ_ERR	SYS_NO_ERR		错误ID
strIPAddress	STRING(17)	'192.168.6.6'		IP地址
strMAC	STRING(17)	'8c-59-3c-10-a9-...		MAC地址
strNetmask	STRING(17)	'255.255.255.0'		子网地址
strGateway	STRING(17)	'0.0.0.0'		网关信息
xDHCP	BOOL	FALSE		是否为自动获取属性

2-5-2-12. Set network port information【XJ_SetNetInfo】

(1) Input variables

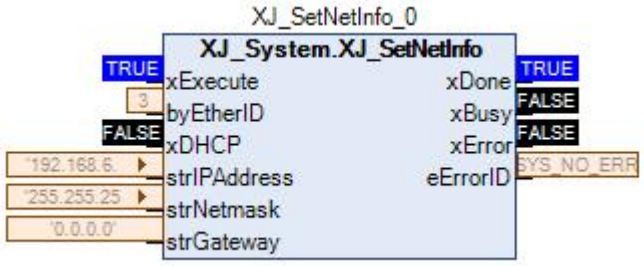
Input variables	Name	Data type	Effective range	Initial value	Description
Excute	Excute	BOOL	TRUE,FALSE	FALSE	Execute the function block at the rising edge
EtherID	Network port ID	BYTE	-	0	Corresponding network port on the display screen
DPCH	DPCH	BOOL	TRUE,FALSE	FALSE	Is it automatic property

	property				acquisition
IPAddress	IP address	STRING	-	0	IP address
Netmask	Subnet address	STRING	-	0	Subnet address
Gateway	Gateway information	STRING	-	0	Gateway information

(2) Output variables

Output variables	Name	Data type	Effective range	Initial value	Description
Done	Execution completed	BOOL	TRUE,FALSE	FALSE	Execution completed
Busy	Executing	BOOL	TRUE,FALSE	FALSE	Executing
Error	Error flag	BOOL	TRUE,FALSE	FALSE	Error flag
ErrorID	Error code	SYS_XJ_ERR	-	0	Error code

The modification will only take effect after powering off and restarting the PLC or RTE.



2-5-2-13. Runtime version【XJ_GetRuntimeVersion】

(1) Output variables

Output variables	Name	Data type	Effective range	Initial value	Description
GetRuntimeVersion	Error code	SYS_XJ_ERR	-	0	Error code
RuntimeVersion	Runtim version	DWORD	-	0	Runtim version

The runtime version is viewed in hexadecimal.

2-5-2-14. BOOT version【XJ_GetBootVersion】

(1) Output variables

Output variables	Name	Data type	Effective range	Initial value	Description
GetBootVersion	Error code	SYS_XJ_ERR	-	0	Error code
BootVersion	BOOT version	STRNG	-	0	BOOT version

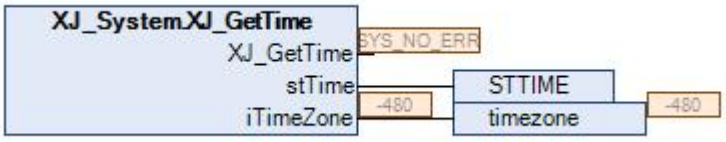
The execution results are as follows:

2-5-2-15. Obtain date and time【XJ_GetTime】

(1) Output variables

Output variables	Name	Data type	Effective range	Initial value	Description
GetTime	Error code	SYS_XJ_ERR	-	0	Error code
Time	Time	SYS_XJ_TIME	-	0	Structural morphology
TimeZone	Get time zone	INT	-	-	Unit is minute

The execution results are as follows:




```
PROGRAM POU
VAR
    // 获取网口信息
    XJ_GetNetInfo_0: XJ_System.XJ_GetNetInfo;
    PLCID: DWORD;
    BOOTTIME: XJ_SYSTEM.SYS_XJ_TIME;
    AA: BOOL;
    NETID: BYTE;
    XJ_SetNetInfo_0: XJ_System.XJ_SetNetInfo;
    runtimeversion: DWORD;
    STTIME: XJ_SYSTEM.SYS_XJ_TIME;
    timezone: INT;
    SetTIME: XJ_System.SYS_XJ_TIME;
    itimezone: INT;
END_VAR
```

STTIME	XJ_SYSTEM.SYS_XJ_TIME				
Year	UINT	2022			年
Month	UINT	11			月
Day	UINT	18			日
Hour	UINT	13			时
Minute	UINT	9			分
Second	UINT	8			秒
Milliseconds	UINT	909			微秒
DayOfWeek	UINT	5			周

2-5-2-16. Set date and time【XJ_SetTime】

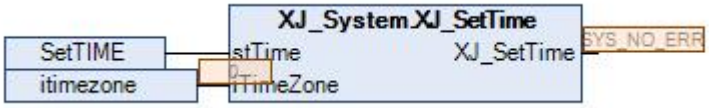
(1) Input variables

Input variables	Name	Data type	Effective range	Initial value	Description
Time	Set time	SYS_XJ_TIME	-	0	Set time
TimeZone	Set time zone	INT	-	-480	Unit is minute

(2) Output variables

Output variables	Name	Data type	Effective range	Initial value	Description
SetTime	Error code	SYS_XJ_ERR	-	0	Error code

The change in time takes effect immediately after writing. Due to the lack of conduction conditions, the time remains unchanged after writing the time. The execution result is as follows:




```
PROGRAM POU
VAR
    // 获取网口信息
    XJ_GetNetInfo_0: XJ_System.XJ_GetNetInfo;
    PLCID: DWORD;
    BOOTTIME: XJ_SYSTEM.SYS_XJ_TIME;
    AA: BOOL;
    NETID: BYTE;
    XJ_SetNetInfo_0: XJ_System.XJ_SetNetInfo;
    STTIME: XJ_SYSTEM.SYS_XJ_TIME;
    runtimeversion: DWORD;
    timezone: INT;
    SetTIME: XJ_System.SYS_XJ_TIME;
    itimezone: INT;
END_VAR
```

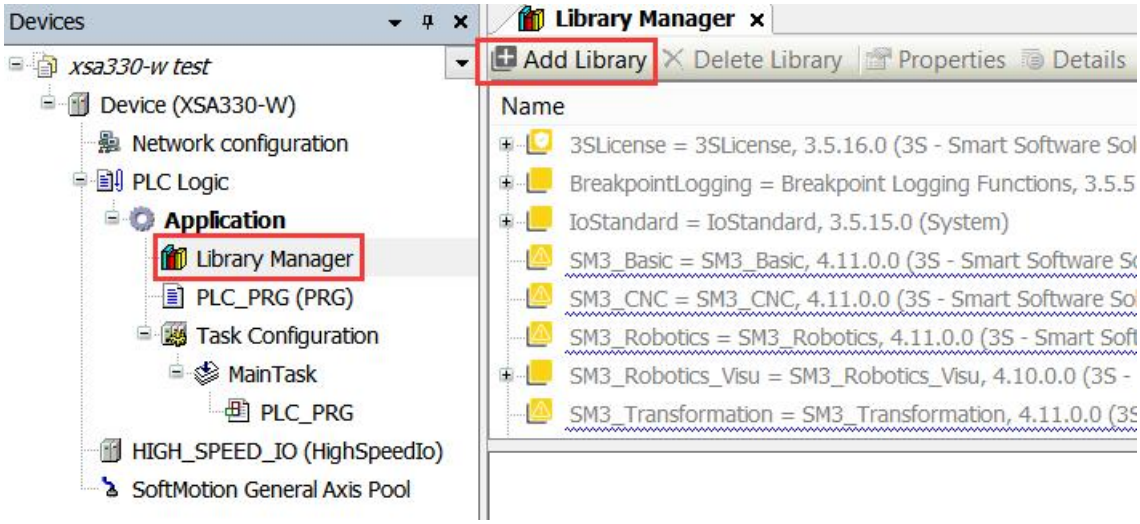
SetTIME	XJ_System.SYS_XJ_TIME				
Year	UINT	2022			年
Month	UINT	11			月
Day	UINT	19			日
Hour	UINT	13			时
Minute	UINT	30			分
Second	UINT	10			秒
Milliseconds	UINT	855			微妙
DayOfWeek	UINT	5			周

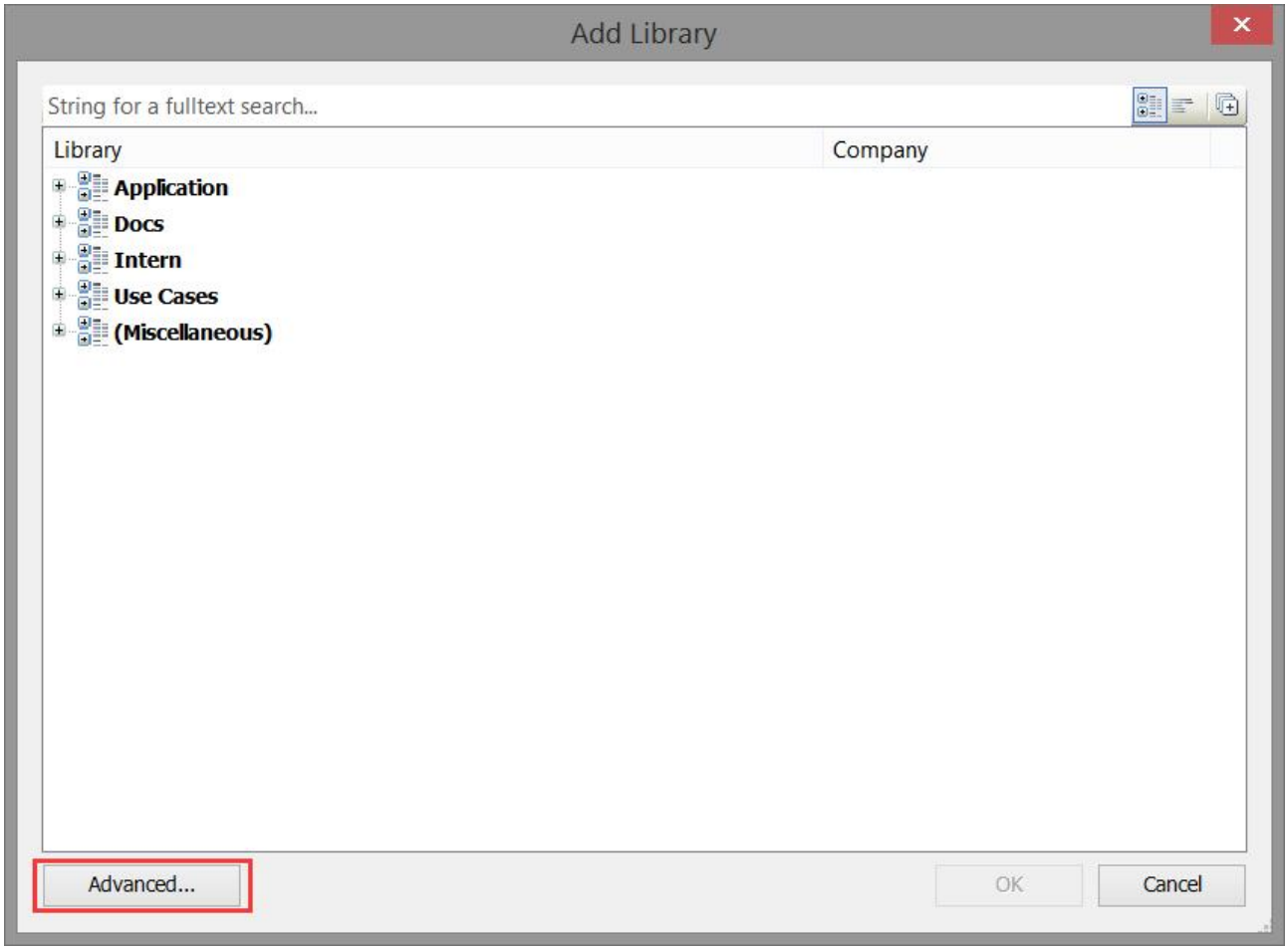
The obtained time is consistent with the set time.

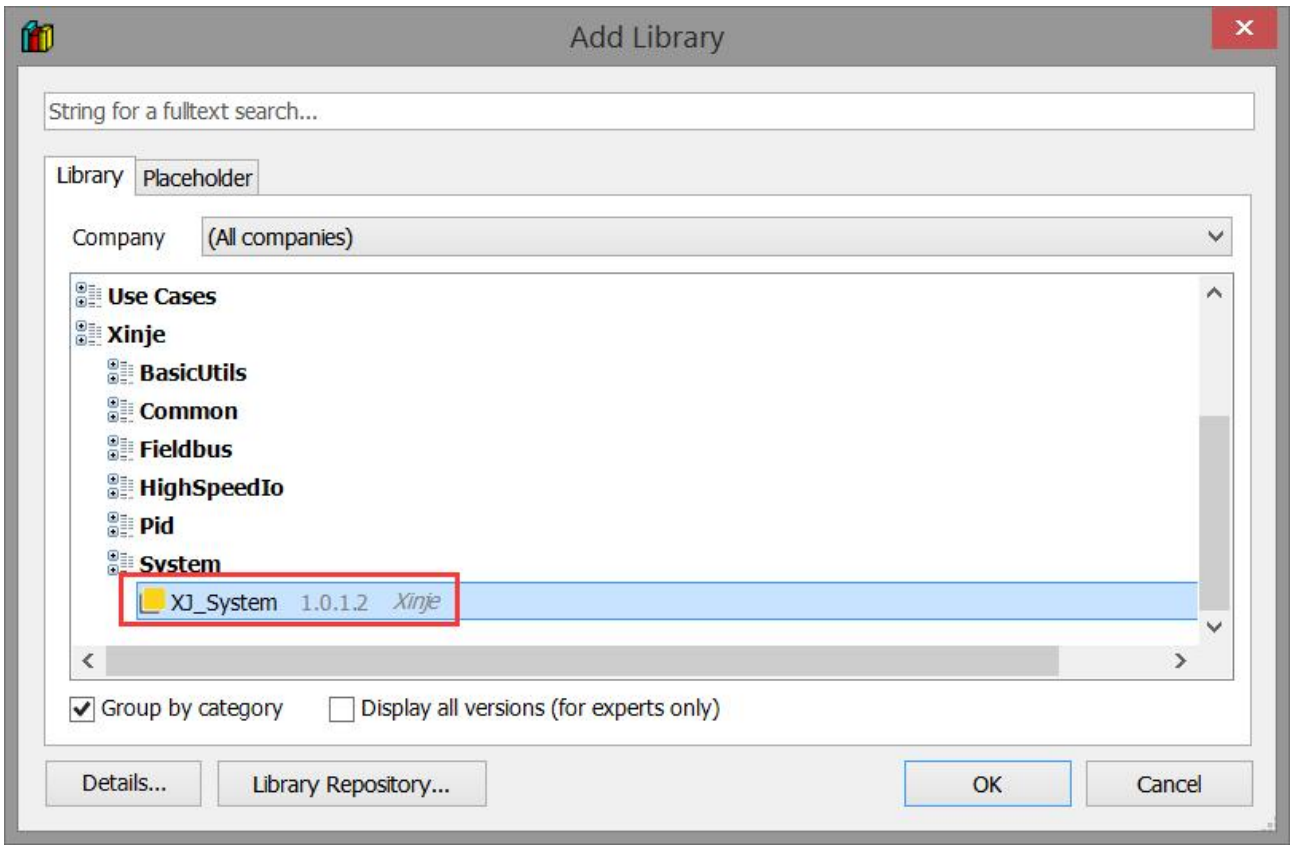
STTIME	XJ_SYSTEM.SYS_XJ_TIME				
Year	UINT	2022			年
Month	UINT	11			月
Day	UINT	19			日
Hour	UINT	13			时
Minute	UINT	30			分
Second	UINT	11			秒
Milliseconds	UINT	938			微妙
DayOfWeek	UINT	6			周

2-5-3. Parameter configuration

Add Library File: Library Manager - Add Library - Advanced - Add XJ_System.






Add Library
String for a fulltext search...
Library
Placeholder
Company
(All companies)
Use Cases
Xinje
BasicUtils
Common
Fieldbus
HighSpeedIo
Pid
System
XJ_System 1.0.1.2 Xinje
Group by category
Display all versions (for experts only)
Details...
Library Repository...
OK
Cancel

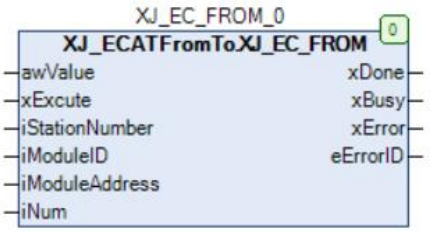
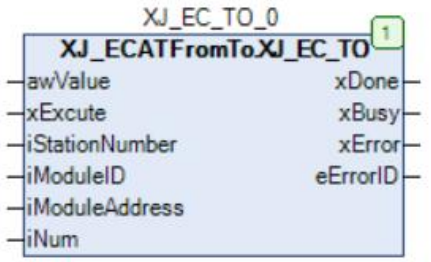
2-6. ECAT_FROMTO

2-6-1. Function overview

Used for reading and writing module parameters on remote IO, for temperature modules and weighing modules.

2-6-2. Function block introduction

2-6-2-1. Instruction format

Command	Name	Graphical representation	ST language
XJ_ECATFromTo.XJ_EC_FROM	Remote IO read		XJ_ECATFromTo.XJ_EC_FROM(awValue:= , xExcute:= , iStationNumber:= , iModuleID:= , iModuleAddress:= , iNum:= , xDone=> , xBusy=> , xError=> , eErrorID=>);
XJ_ECATFromTo.XJ_EC_TO	Remote IO write		XJ_ECATFromTo.XJ_EC_TO(awValue:= , xExcute:= , iStationNumber:= , iModuleID:= , iModuleAddress:= , iNum:= , xDone=> , xBusy=> , xError=> , eErrorID=>);

2-6-2-2. Remote IO read【XJ_ECATFromTo.XJ_EC_FROM】

(1) Input variables

VAR_INPUT	Name	Data type	Effective range	Initial value	Description
awValue	Read value	POINTER TO WORD	-	-	The longest array length [0..100]
xExcute	Rising edge trigger	BOOL	-	-	
iStationNumber	Station no.	UINT		-	ECAT station no., such as 1001
iModuleID	Module no.	INT	0-15	-	One remote IO with a maximum of 16 modules
iModuleAddress	Module address	DWORD	-	-	Refer to the module address in the expansion module manual
iNum	Register numbers	INT	-	-	Number of consecutive Word reads

(2) Output variables

VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
xDone	Boolean output	BOOL	TRUE,FALSE	FALSE	Read completed is TRUE
xBusy	Boolean output	BOOL	TRUE,FALSE	FALSE	Reading is TRUE
xError	Boolean output	BOOL	TRUE,FALSE	FALSE	True when an exception occurs
eErrorID	Present mode	EC_FT_ERRID	-	-	When normal, the value is 0. When an exception occurs, an error code is output

2-6-2-3. Remote IO write【XJ_ECATFromTo.XJ_EC_TO】

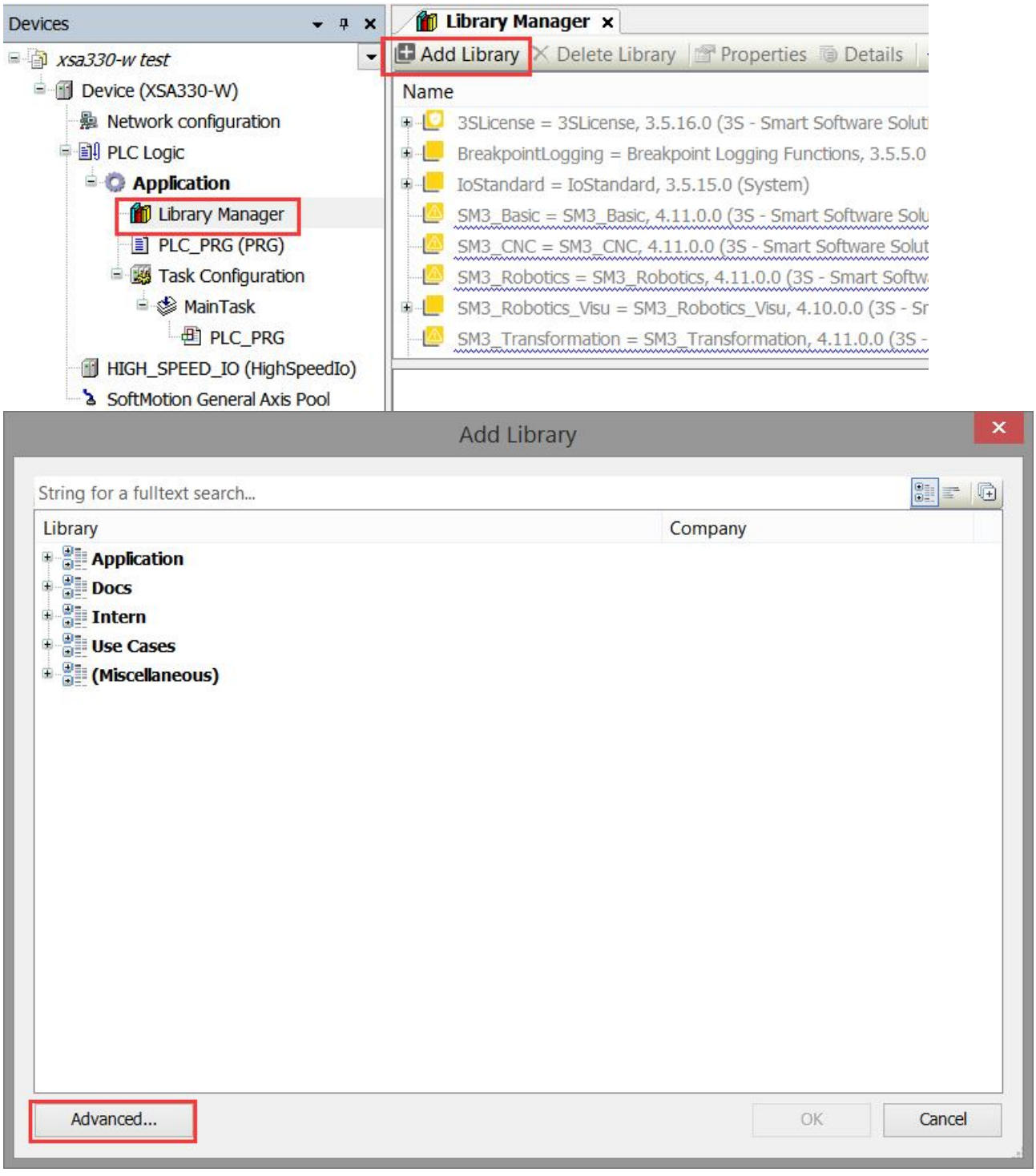
(1) Input variables

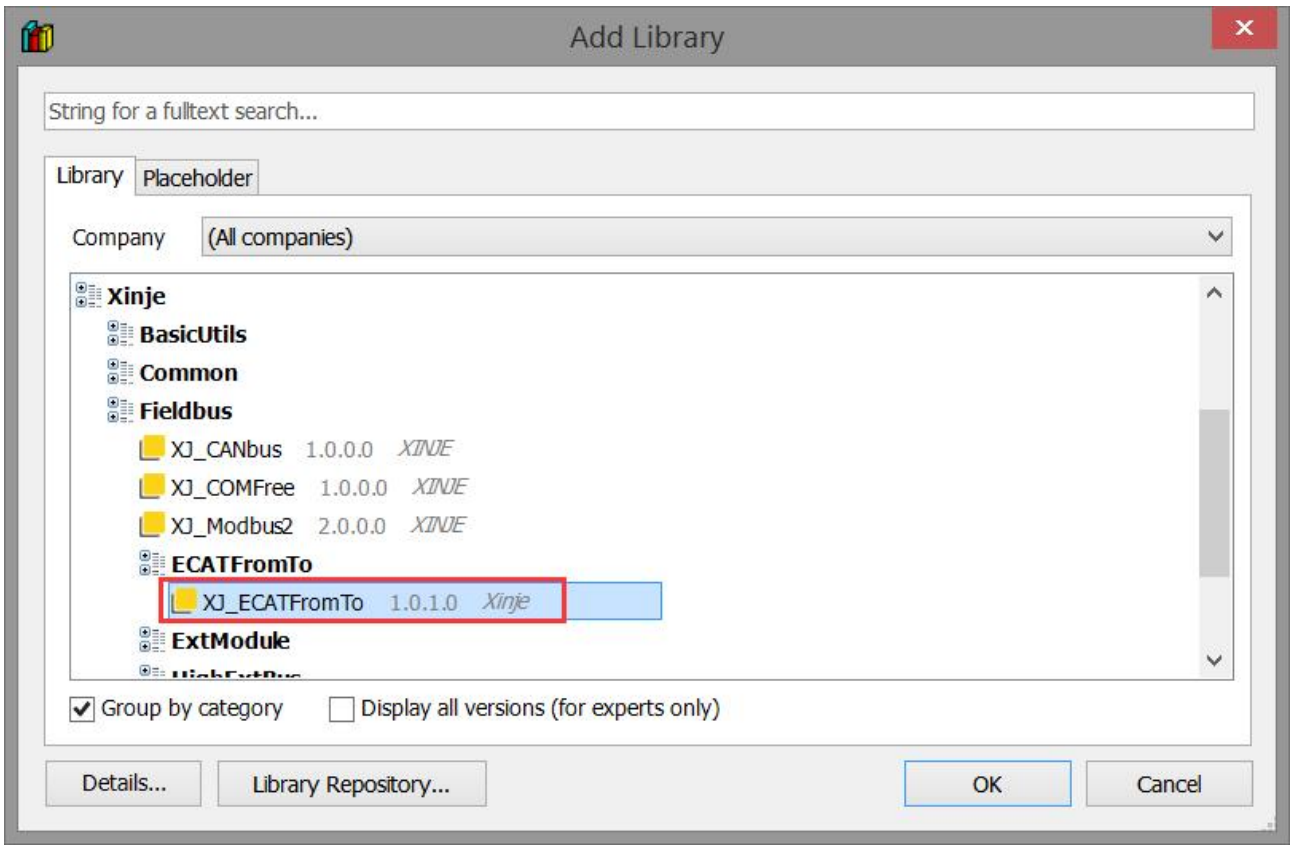
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
awValue	write in value	POINTER TO WORD	-	-	The longest array length [0..100]
xExcute	Rising edge trigger	BOOL	-	-	
iStationNumber	Station no.	UINT		-	ECAT station no., such as 1001
iModuleID	Module no.	INT	0-15	-	One remote IO with a maximum of 16 modules
iModuleAddress	Module address	DWORD	-	-	Refer to the module address in the expansion module manual
iNum	Register numbers	INT	-	-	Number of consecutive Word writes

(2) Output variables

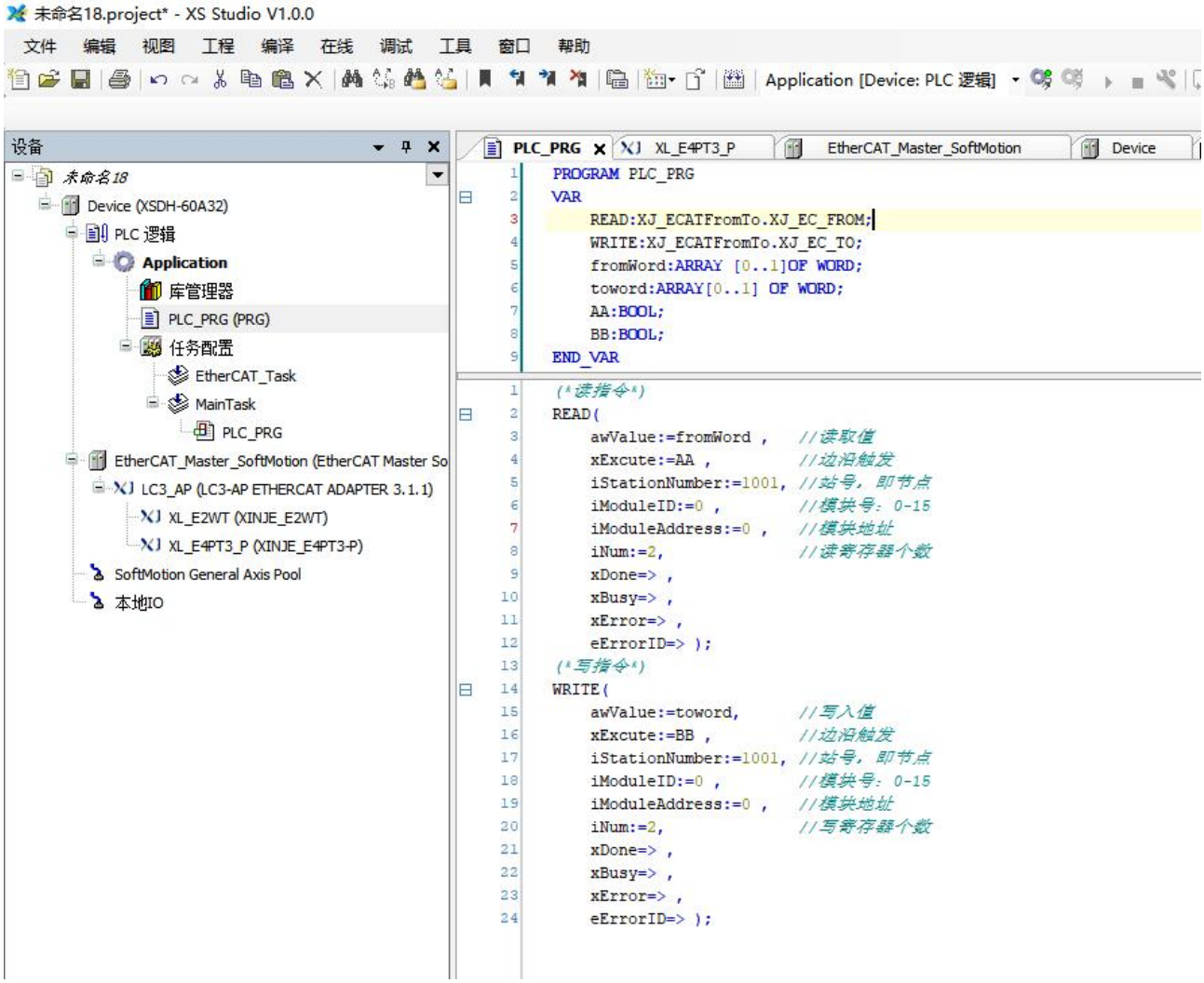
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
xDone	Boolean output	BOOL	TRUE,FALSE	FALSE	Write completed is TRUE
xBusy	Boolean output	BOOL	TRUE,FALSE	FALSE	Writing is TRUE
xError	Boolean output	BOOL	TRUE,FALSE	FALSE	True when an exception occurs
eErrorID	Present mode	EC_FT_ERRID	-	-	When normal, the value is 0. When an exception occurs, an error code is output

2-6-3. Parameter configuration




Add Library
String for a fulltext search...
Library
Placeholder
Company
(All companies)
Xinje
BasicUtils
Common
Fieldbus
XJ_CANbus 1.0.0.0 XINJE
XJ_COMFree 1.0.0.0 XINJE
XJ_Modbus2 2.0.0.0 XINJE
ECATFromTo
XJ_ECATFromTo 1.0.1.0 Xinje
ExtModule
Group by category
Display all versions (for experts only)
Details...
Library Repository...
OK
Cancel

2-6-4. Application



3. Motion instructions

3-1. Single axis

3-1-1. Single axis instruction overview

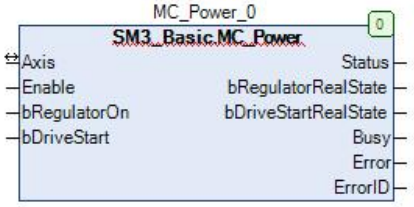
Command	Function
MC_Power	Put the axis into a runnable state
MC_Reset	Reset axis internal related errors
MC_Stop	Stop controller movement
MC_Halt	Pause the execution of functional blocks in progress
MC_Home	Homing
MC_Jog	Jog run
MC_MoveAbsolute	Implement a control axis to reach the specified absolute position
MC_MoveAdditive	Accelerate to move an additional distance at a given speed
MC_MoveRelative	Move the axis to a relative position from the current axis position
MC_MoveSuperImposed	Based on the previous motion, the superimposed velocity and acceleration run an additional distance
MC_MoveVelocity	The shaft continues to operate at a specified speed
MC_PositionProfile	Perform movements according to time-location planning
MC_VeloctyProfile	Perform movements according to time-velocity planning
MC_AccelerationProfile	Perform movements according to time-acceleration planning
MC_ReadActualPosition	Reads the current position of the current relevant axis
MC_ReadActualTorque	Reads the current torque of the current relevant axis
MC_ReadActualVlocity	Reads the current speed of the current related axis
MC_ReadAxisError	Obtain the error code
MC_ReadBoolParameter	Obtain parameter values based on parameter serial number
MC_ReadParameter	Obtain the parameter value based on the parameter ID
SMC_ReadSetPosition	Reads the set position of the current axis
SMC_ReadFBError	Read the historical error information of the function block
MC_WriteBoolParameter	Modify the parameter values of a specific Boolean variable specified by the user
MC_WriteParameter	Modify a special parameter specified by a user
SMC_ClearFBError	Clear historical error information of the function block
SMC_ErrorString	Read the error description corresponding to the error code
SMC3_ReinitDrive	Re-invoke the drive/axis
SMC3_ETC_WriteParameter_CoE	Set COE parameters for the axis
MC_TouchProbe	Probe instruction

3-1-2. Single axis instructions

3-1-2-1. Axis enable [MC_Power]

(1) Instruction overview

It is used to enable a specified axis to enter or exit the runable state, also called axis enable.

Instruction	Name	Graphic representation	ST language
MC_Power	Axis enable		MC_Power(Axis:= , Enable:= , bRegulatorOn:= , bDriveStart:= , Status=> , bRegulatorRealState=> , bDriveStartRealState=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Enable	Valid	BOOL	TRUE,FALSE	FALSE	Must be set to TRUE to activate processing of function blocks
bRegulatorOn	Enable	BOOL	TRUE,FALSE	FALSE	Must be set to TRUE to enable the function block
bDriveStart	Enable drive	BOOL	TRUE,FALSE	FALSE	Must be set to TRUE to turn off emergency stop processing for function blocks
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Status	Can run	BOOL	TRUE,FALSE	FALSE	TRUE if the axis is ready
bRegulatorRealState	Enable valid	BOOL	TRUE,FALSE	FALSE	Active status of the axis enablement
bDriveStartRealState	Drive can be used	BOOL	TRUE,FALSE	FALSE	The drive is not interrupted by the quick stop mechanism is TRUE
Busy	Executing	BOOL	TRUE,FALSE	FALSE	The processing of the function block did not complete is TRUE
Error	Error	BOOL	TRUE,FALSE	FALSE	TRUE if an exception occurs
ErrorID	Error code	SMC_ERROR	-	0	In normal cases, the value is 0. When an exception occurs, an error code is displayed

(3) Function description

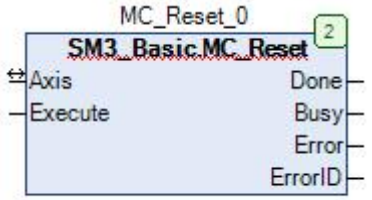
- Processes the input only if Enable is TRUE.
- Call MC_Power with bRegulatoron = FALSE to set nAxisState of the reference axis to state (disabled) off. Then the axis can't move.
- If there are no errors on the disabled axis, call MC_Power with bRegulatorOn = TRUE, sets its AxisState to pause.
- If an error is detected, set the axis status to errorstop.
- If the input Enable, bRegulatoron and bDrivestart are TRUE, but the output Status remains FALSE for a long time, there may be a drive power-level hardware problem. If the power supply fails (also

during operation), the nAxisState of the reference axis is set to errorstop.

3-1-2-2. Axis reset [MC_Reset]

(1) Instruction overview

Clear the axis error.

Instruction	Name	Graphic representation	ST language
MC_Reset	Axis reset		MC_Reset(Axis:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Execute	Valid	BOOL	TRUE,FALSE	FALSE	The rising edge of the input value initiates the execution of the function block
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	Done	BOOL	TRUE,FALSE	FALSE	TRUE if the reset is performed
Busy	Executing	BOOL	TRUE,FALSE	FALSE	TRUE when the function block execution has not finished
Error	Error	BOOL	TRUE,FALSE	FALSE	Function block execution error
ErrorID	Error code	SMC_ERROR	-	0	Error recognition

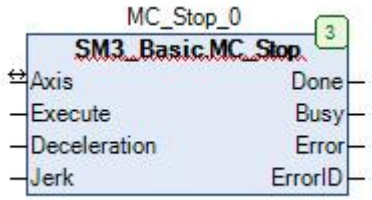
(3) Function description

- When the shaft communication is normal, change the axis state from errorstop to Standstill, turns the abnormal state of the axis to the normal running state.
- The function block returns the error SMC_R_NO_ERROR_TO_RESET when called in a state other than errorstop.

3-1-2-3. Stop controller motion [MC_Stop]

(1) Instruction overview

Deceleration stop the axis.

Instruction	Name	Graphic representation	ST language
MC_Stop	Forced stop		MC_Stop(Axis:= , Execute:= , Deceleration:= , Jerk:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Execute	Valid	BOOL	TRUE/FALSE	FALSE	Trigger the axis motion at rising edge
Deceleration	Deceleration speed	LREAL	Positive value	0	Deceleration speed. the unit is [command unit/s^2]
Jerk	Jerk	LREAL	Positive value	0	Jerk. the unit is [command unit/s^3]
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	Done	BOOL	TRUE/FALSE	FALSE	Becomes TRUE upon completion of execution
Busy	Executing	BOOL	TRUE/FALSE	FALSE	Becomes TRUE after receiving instruction
Error	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	Error code	SMC_ERROR	-	0	Error recognition

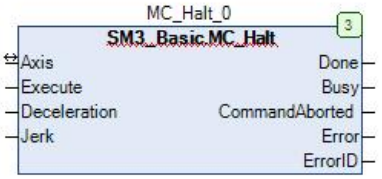
(3) Function description

- MC_Stop Sets the axis to the stop state. Therefore, the motion of the currently running function block instance is aborted. The only exception is a running Mc_Stop instance, which will not be aborted. Instead, the MC_Stop instance that was just started returns an error.)
- As long as the axis is in the stopped state, no other instance can perform motion on it. If the axis reaches the speed value of zero, the Done output is set to TRUE. As long as the Execute input is TRUE, the axis remains stop. If Execute goes to FALSE and the Done output is TRUE, the axis goes to a standstill.

3-1-2-4. Pasue the motion [MC_Halt]

(1) Instruction overview

Slow down to stop the movement being performed by the axis, and the stopped movement can resume the unfinished part of the execution.

Instruction	Name	Graphic representation	ST language
MC_Halt	Pause the motion		MC_Halt(Axis:= , Execute:= , Deceleration:= , Jerk:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis
VAR_INPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Execute	Valid	BOOL	TRUE/FALSE	FALSE	Start execution at rising edge
Deceleration	Deceleration speed	LREAL	Positive value	0	Deceleration speed. the unit is [command unit/s^2]
Jerk	Jerk	LREAL	Positive value	0	Jerk. the unit is [command unit/s^3]
VAR_OUTPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Done	Done	BOOL	TRUE/FALSE	FALSE	Becomes TRUE upon completion of execution
Busy	Executing	BOOL	TRUE/FALSE	FALSE	TRUE when the function block execution has not finished
CommandAborted	Instruction interrupted	BOOL	TRUE/FALSE	FALSE	TRUE if the command has been terminated by another command
Error	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	Error code	SMC_ERROR	-	0	Error recognition

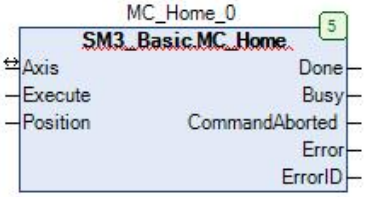
(3) Function description

- This function block stops referencing axes in a controlled manner. If operations of other function blocks are running at this time, the operation is aborted. The axis enters discrete motion until it reaches velocity 0. If the Done output of MC_Halt is set, the axis comes to a standstill. As long as MC_Halt is active, a new motion command can be issued to interrupt the execution of MC_Halt. Unlike MC_Stop, MC_Halt can be interrupted.

3-1-2-5. Axis homing [MC_Home]

(1) Instruction overview

This command executes the pulse motor homing action, and the specific homing process is determined by the homing mode designed by the bus driver.

Instruction	Name	Graphic representation	ST language
MC_Home	axis homing		MC_Home(Axis:= , Execute:= , Position:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis
VAR_INPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Execute	Valid	BOOL	TRUE/FALSE	FALSE	Execute at the rising edge
Position	Axis reached location	LREAL	Data range	0	Represents the homing position of the axis
VAR_OUTPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Done	Completed	BOOL	TRUE/FALSE	FALSE	Change to TRUE upon completion of execution
Busy	Executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
CommandAborted	Command is interrupted	BOOL	TRUE/FALSE	FALSE	True if the command has been terminated by another command
Error	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	Error code	SMC_ERROR	-	0	Error recognition

(3) Function

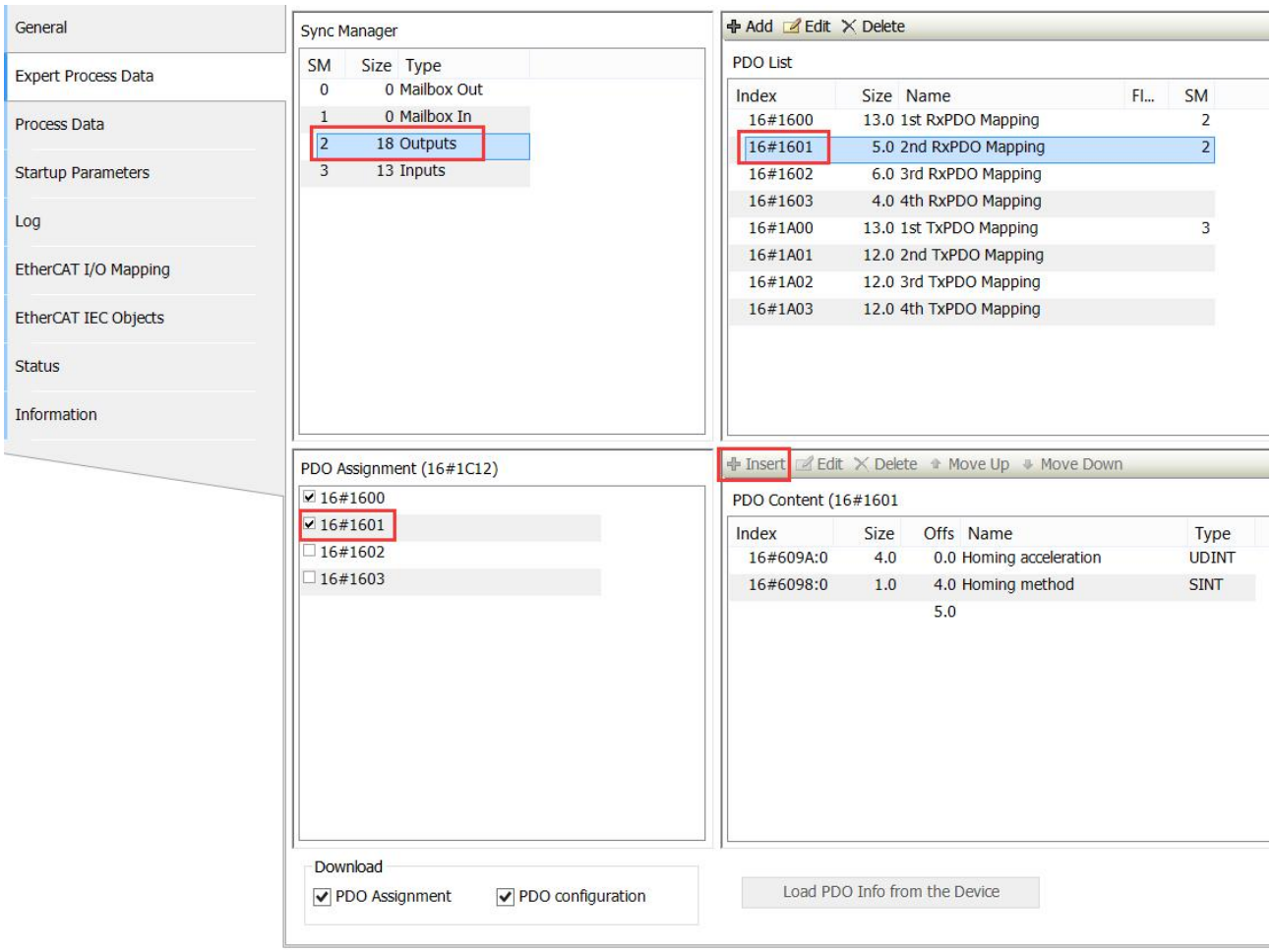
- This function block is for homing operation, and the position is the zero position of the axis.
- The running status of this function block is in Standstill, and the state of the instruction running is homing. Other states cannot be run.
- Start the instruction at the rising edge of Execute.
- Before executing the homing, it is necessary to configure the bus driver homing parameters, such as homing mode, speed, acceleration, etc. Please refer to the driver manual. For example, DS5C1 series servo driver user manual chapter 6.5 HM mode.
- The general bus driver needs to set the index and sub index data as shown in the table below for returning to zero:

Index	Sub index	Explanation
0x6098	00h	Homing mode

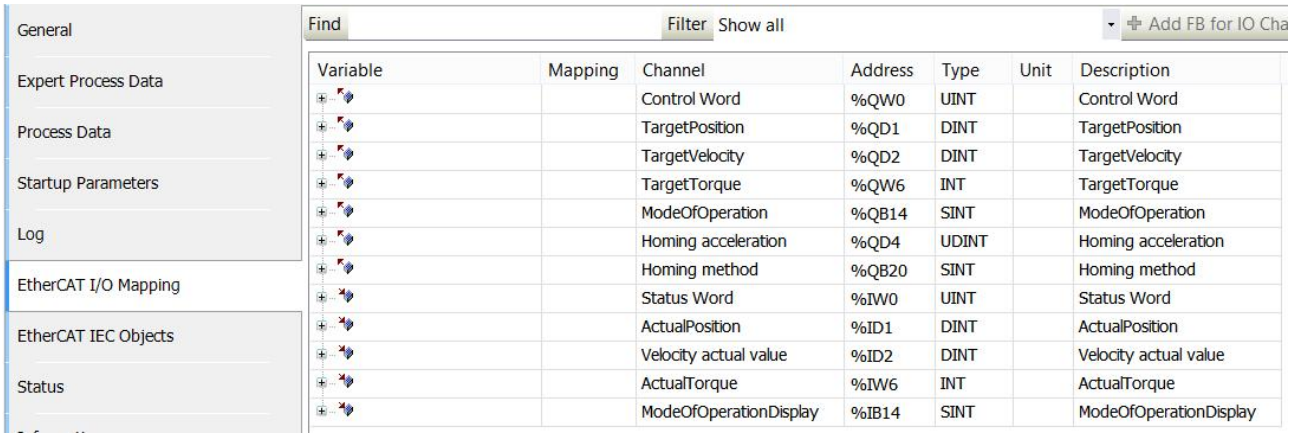
Index	Sub index	Explanation
0x6099	01h	The speed of the process from starting to homing to finding zero point, with a higher value to reduce the homing time
0x6099	02h	The speed of the process from finding zero point to homing completed, with lower values to improve accuracy
0x609A	00h	Homing acceleration

(4) Application example

Example 1: Taking the Xinje DS5C servo as an example, the specified axis homing in homing mode 1. P5-22 is the positive limit setting address, with a default value of 1, which corresponds to the servo terminal SI1. P5-23 is the negative limit setting address, with a default value of 2, which corresponds to the servo terminal SI2. P5-27 sets the address for the origin, with a default value of 3, which corresponds to the servo terminal SI3.
Make the program:
Select [expert settings], select 1601 in [Expert process data] – [PDO assignment], add 6098h, 609Ah in 1601.

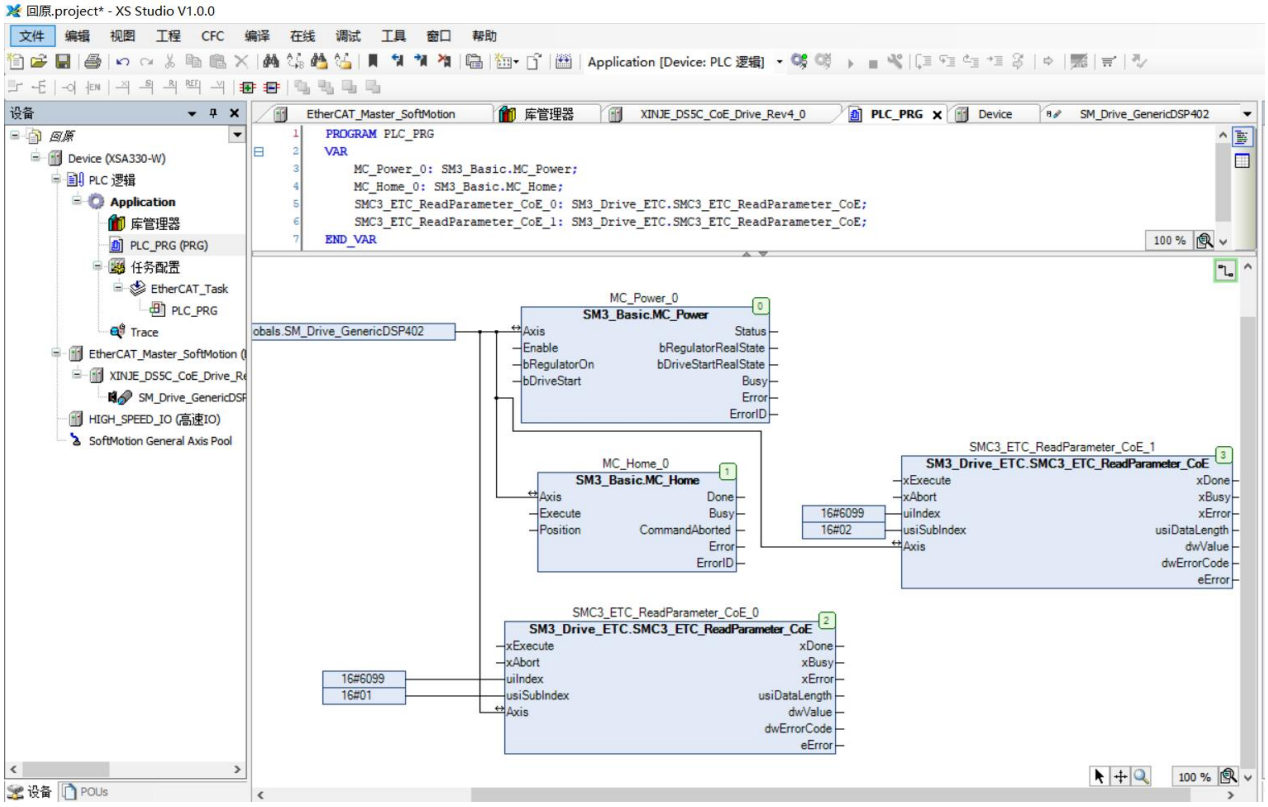


The new added parameters can be seen in [EtherCAT I/O mapping].

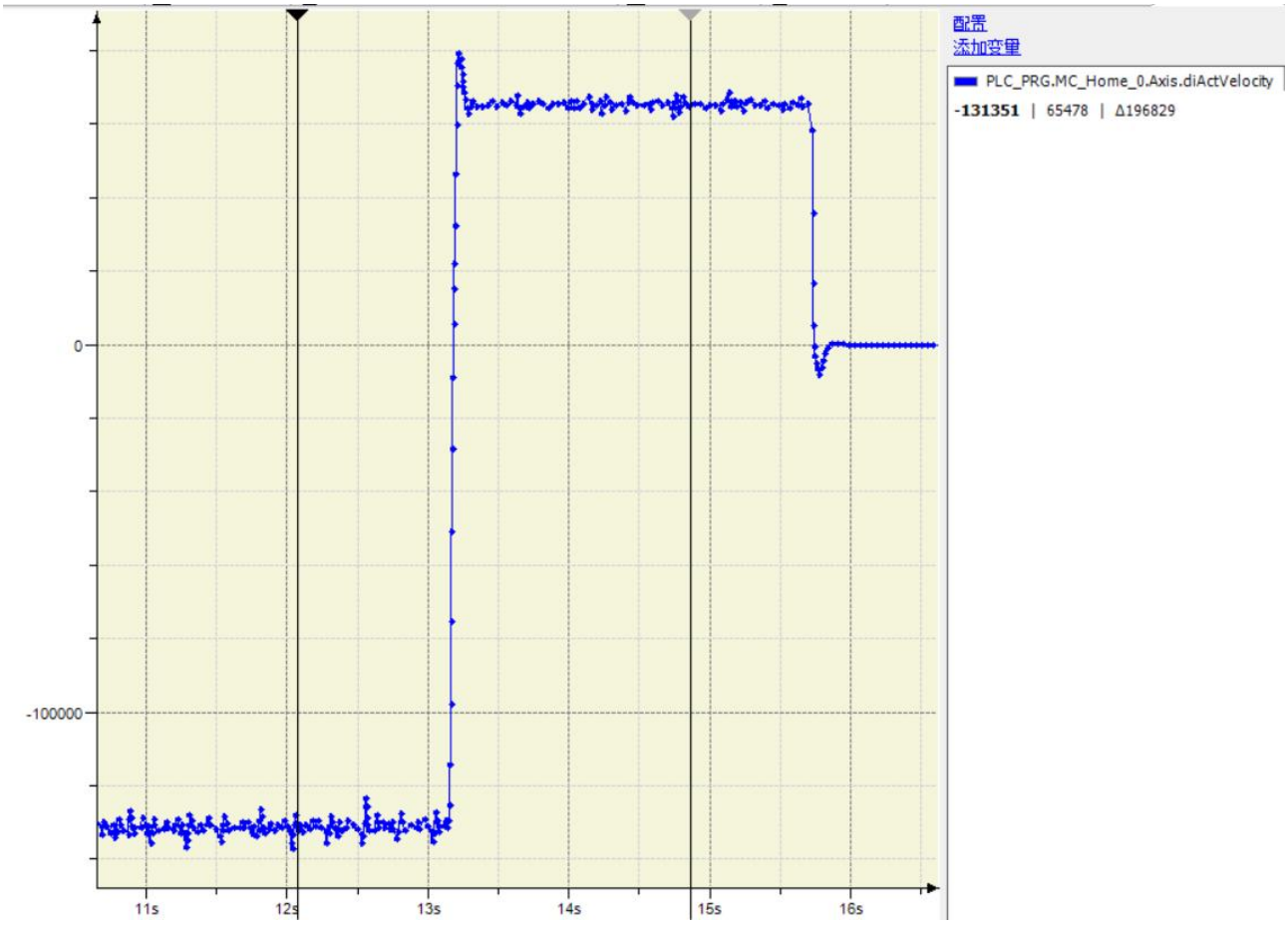


Variable	Mapping	Channel	Address	Type	Unit	Description
		Control Word	%QW0	UINT		Control Word
		TargetPosition	%QD1	DINT		TargetPosition
		TargetVelocity	%QD2	DINT		TargetVelocity
		TargetTorque	%QW6	INT		TargetTorque
		ModeOfOperation	%QB14	SINT		ModeOfOperation
		Homing acceleration	%QD4	UDINT		Homing acceleration
		Homing method	%QB20	SINT		Homing method
		Status Word	%IW0	UINT		Status Word
		ActualPosition	%ID1	DINT		ActualPosition
		Velocity actual value	%ID2	DINT		Velocity actual value
		ActualTorque	%IW6	INT		ActualTorque
		ModeOfOperationDisplay	%IB14	SINT		ModeOfOperationDisplay

Read the homing speed in 6099h through SMC3_ETC_ReadParameter_CoE, set homing mode 1 in [Ethercat I/O mapping], homing acceleration set to 13107200, after enabling through MC_Power, set ON the function block MC_Home, then give SI1 signal.



Check the homing speed through Trace.



3-1-2-6. Jog run [MC_Jog]

(1) Instruction overview

This command is used to manually control the axis movement in the specified direction.

Instruction	Name	Graphic representation	ST language
MC_Jog	Jog run	MC_Jog_0 SM3_Basic.MC_Jog Axis JogForward JogBackward Velocity Acceleration Deceleration Jerk Busy CommandAborted Error ErrorId	MC_Jog(Axis:= , JogForward:= , JogBackward:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Busy=> , CommandAborted=> , Error=> , ErrorId=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
JogForward	Forward jog run	BOOL	TRUE/FALSE	FALSE	If JogForward is TRUE, the axis will move forward with the given parameters (Velocity, Acceleration, Deceleration, and Jerk), and if JogBackward is also TRUE, the axis will not move
JogBackward	Reverse jog run	BOOL	TRUE/FALSE	FALSE	If JogBackward is TRUE, the axis will move reverse with the given parameters (Velocity, Acceleration, Deceleration, and Jerk), and if JogForward is both TRUE, the axis will not move
Velocity	Target speed	LREAL	0, positive number	0	Maximum speed [u/s]
Acceleration	Target acceleration speed	LREAL	positive number	0	Acceleration value [u/s^2]
Deceleration	Target deceleration speed	LREAL	positive number	0	Deceleration value [u/s^2]
Jerk	Target jerk speed	LREAL	positive number	0	Jerk value [u/s^3]
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Busy	Executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
CommandAborted	Instruction interrupted	BOOL	TRUE/FALSE	FALSE	True if the command has been terminated by another command
Error	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	Error code	SMC_ERROR	-	0	Error indentification

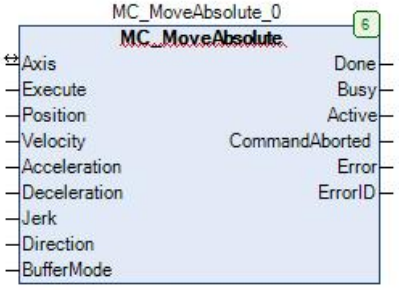
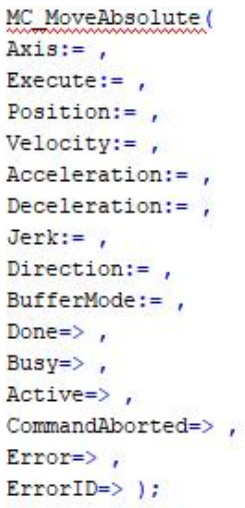
(3) Function description

- Perform jog operation according to the specified Velocity (target speed).
- When forward operation is required, set JogForward (valid for forward operation) to TRUE; When reverse running is required, set JogBackward (valid for negative running) to TRUE.
- Setting both JogForward (positive running valid) and JogBackward (negative running valid) to TRUE will prevent any movement from occurring. If MC_Jog command speed setting exceeds the maximum jog speed in the axis parameters, it will be executed at the maximum jog speed.

3-1-2-7. Absolute position [MC_MoveAbsolute]

(1) Instruction overview

This command is used to move the control axis to the specified absolute position according to the set parameters.

Instruction	Name	Graphic representation	ST language
MC_MoveAbsolute	absolute position		


```
MC_MoveAbsolute(
Axis:= ,
Execute:= ,
Position:= ,
Velocity:= ,
Acceleration:= ,
Deceleration:= ,
Jerk:= ,
Direction:= ,
BufferMode:= ,
Done=> ,
Busy=> ,
Active=> ,
CommandAborted=> ,
Error=> ,
ErrorID=> );
```

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis
VAR_INPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Execute	Valid	BOOL	TRUE/FALSE	FALSE	Starting execution at the rising edge
Position	Position	LREAL	Data range	0	Target position of motion
Velocity	Target speed	LREAL	0, positive number	0	Maximum speed [u/s]
Acceleration	Target acceleration speed	LREAL	positive number	0	Acceleration value [u/s^2]
Deceleration	Target deceleration speed	LREAL	positive number	0	Deceleration value [u/s^2]
Jerk	Target jerk speed	LREAL	positive number	0	Jerk value [u/s^3]
Direction	Direction	MC_DIRECTION	3: fastest 2: current 1: Positive 0: shortest -1: Negative	0	Fastest: Automatically select the fastest direction to move Current: Move in the current direction Positive: Forward run Shortest: Select direction based on the shortest path

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
					Negative: Reverse run
BufferMode	Buffer mode	MC_BUFFER_MODE	-	0	If the function block is Busy, only BufferMode=Aborting is allowed
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	Completed	BOOL	TRUE/FALSE	FALSE	TRUE after execution is completed
Busy	Executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Active	In control	BOOL	TRUE/FALSE	FALSE	Change to TRUE in control
CommandAborted	Instruction interrupted	BOOL	TRUE/FALSE	FALSE	True if the command has been terminated by another command
Error	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	Error code	SMC_ERROR	-	0	Error indentification

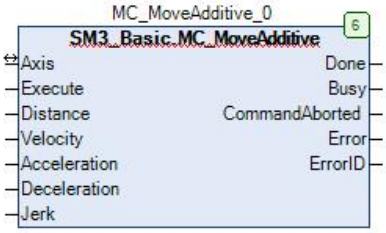
(3) Function description

- This function block moves the shaft to an absolute position and uses the values of speed, deceleration, acceleration, and jerk. If not set, the instruction will end at speed 0 after execution. (i.e. standstill state)
- The start command start at the rising edge of Execute.

3-1-2-8. Position overlay [MC_MoveAdditive]

(1) Instruction overview

The axis is overlaid with the data specified by Distance on the original command position, which is used for online stacking position during the motion axis control process.

Instruction	Name	Graphic representation	ST language
MC_MoveAdditive	Position overlay		MC_MoveAdditive(Axis:= , Execute:= , Distance:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis
VAR_INPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Execute	Valid	BOOL	TRUE/FALSE	FALSE	Starting execution at the rising edge
Distance	Position	LREAL	Data range	0	This data is stacked position data
Velocity	Target speed	LREAL	0, positive number	0	Maximum speed [u/s]
Acceleration	Target acceleration speed	LREAL	positive number	0	Acceleration value [u/s^2]
Deceleration	Target deceleration speed	LREAL	positive number	0	Deceleration value [u/s^2]
Jerk	Target jerk speed	LREAL	positive number	0	Jerk value [u/s^3]
VAR_OUTPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Done	Completed	BOOL	TRUE/FALSE	FALSE	TRUE after execution is completed
Busy	Executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
CommandAborted	Instruction interrupted	BOOL	TRUE/FALSE	FALSE	True if the command has been terminated by another command
Error	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	Error code	SMC_ERROR	-	0	Error indentification

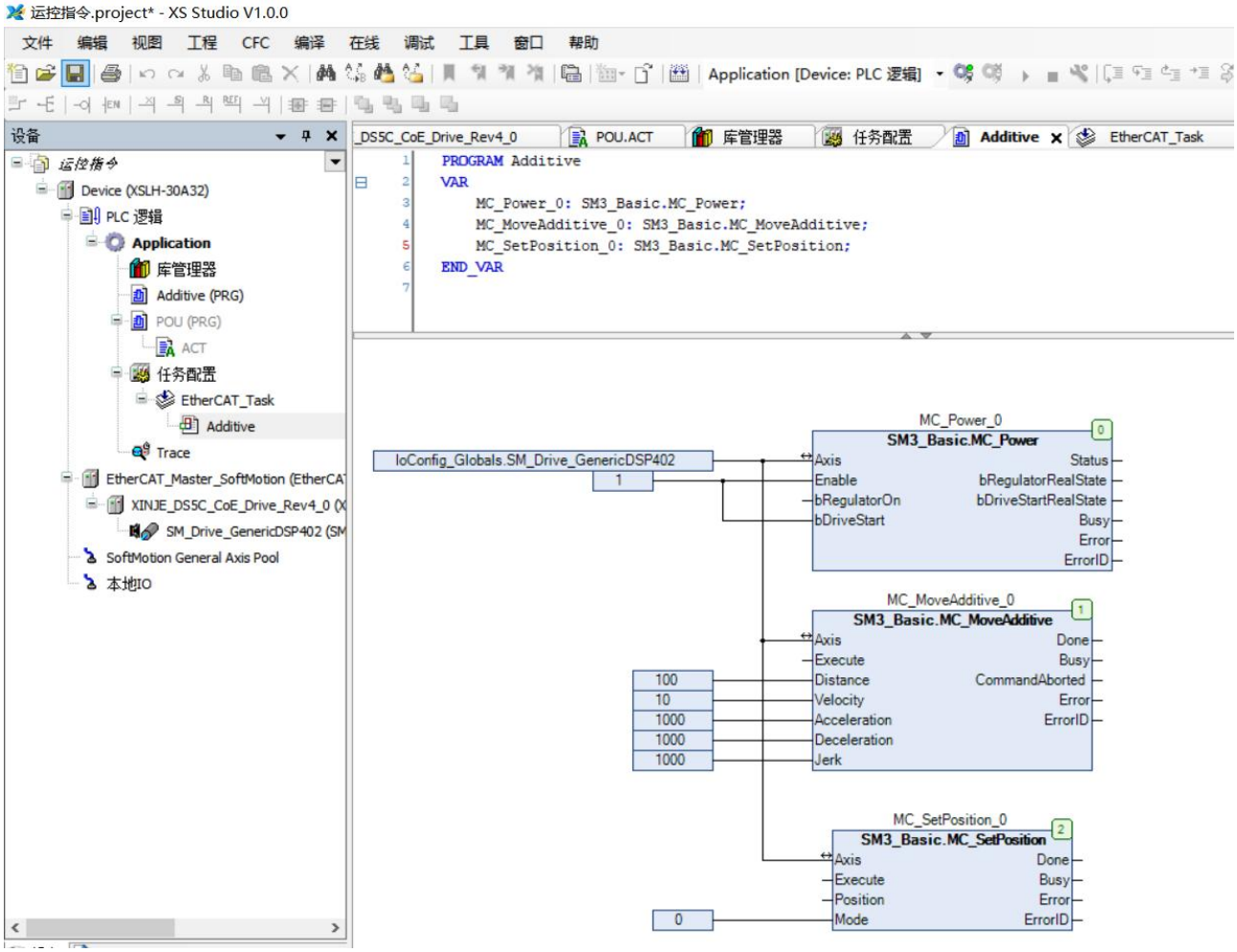
(3) Function description

- This function block is a stacking position command, and the Distance data is the stacking data of the axis;
- If the running status of this function block is Discrete Motion, the CommandAbort of other instructions will be set ON during usage;
- In the standstill state, this instruction can run independently to achieve relative positioning requirements;
- Acceleration or Deceleration is zero, and the instruction operation is in an abnormal state, but the state of the axis is Discrete Motion;
- The start command will start at the rising edge of Execute.

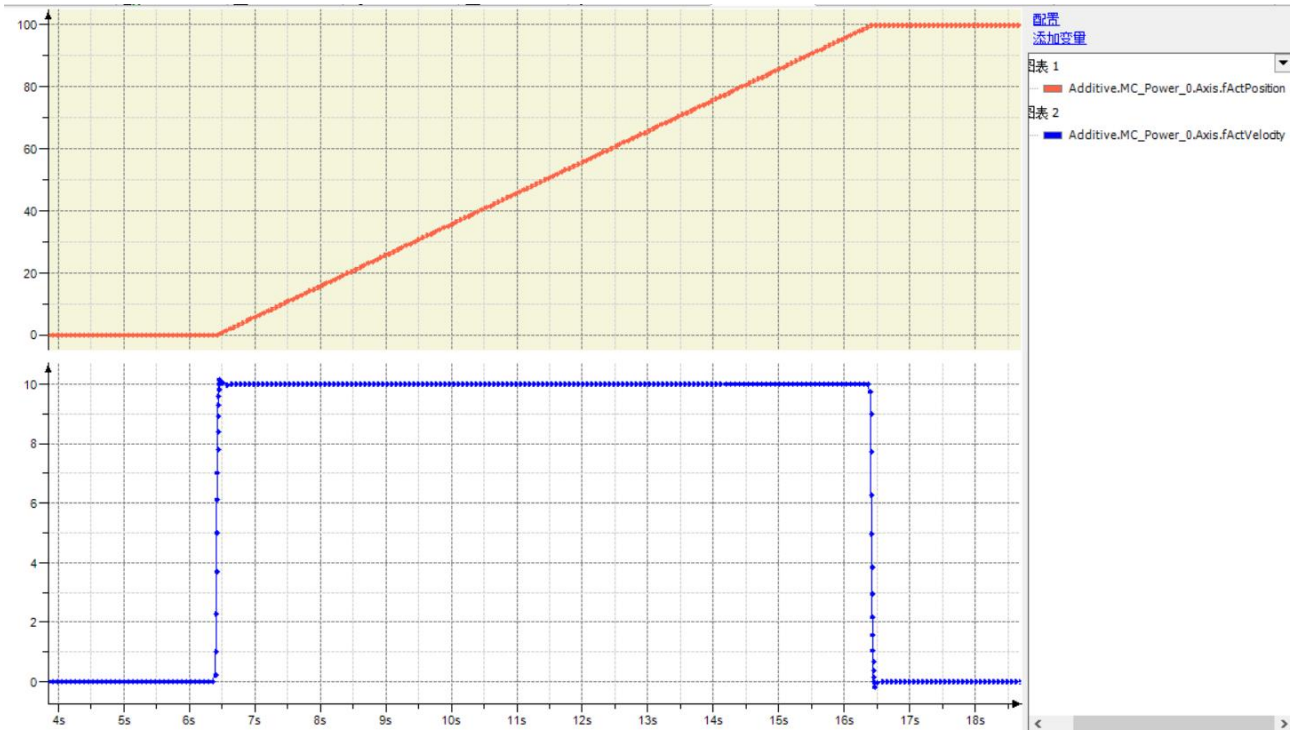
(4) Application example

Example 1: It can be directly used and run through "MC_MoveAdditive" after "MC_Power" is enabled.

① Programming: You can first set the position to 0 through "MC_SetPosition", enable it, and then set ON MC_MoveAdditive, can be configured to run at a speed of 10 and at a position of 100.

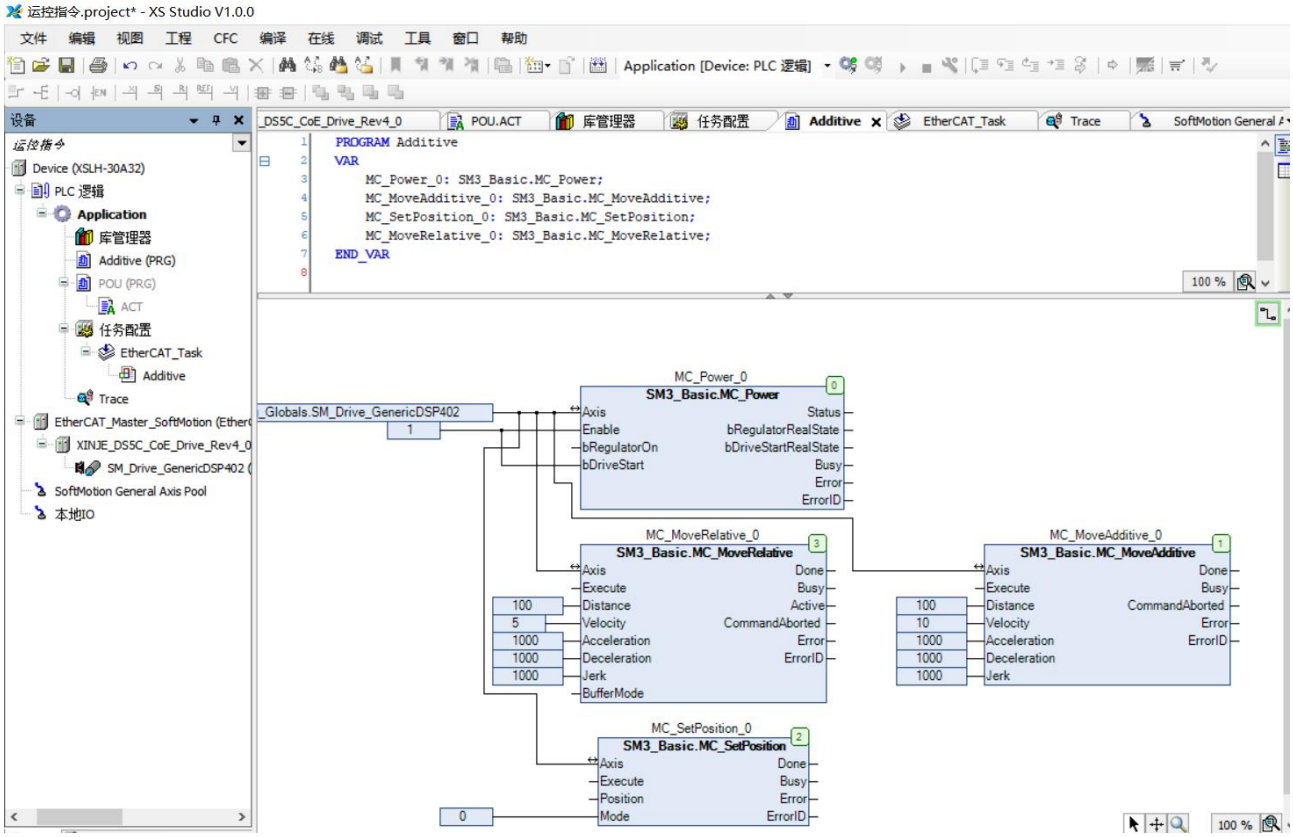


② The position and speed of operation can be seen through Trace.

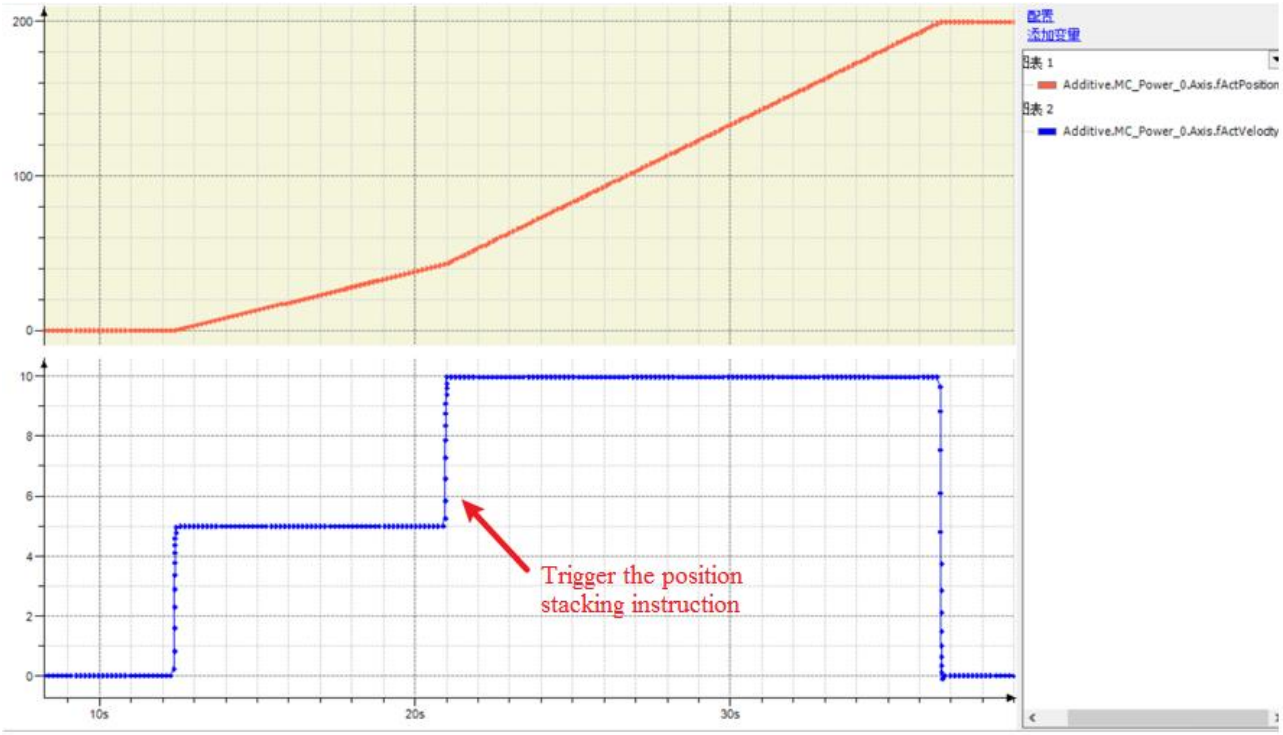


Example 2: After "MC_Power" is enabled, execute "MC_MoveRelative" first, and then "MC-MoveAdditive".

① Write a program: "MC_MoveRelative" is executed at a speed of 5 and a target position of 100. It can be turned on during operation or after waiting for the execution of "MC_MoveRelative" to complete. The speed set for "MC_MoveAdditive" is 10 and the target position is 100. The execution speed after turning on is 10 and the final position is 100. That is, the position is superimposed, and the acceleration is not superimposed, only the velocity is reset.



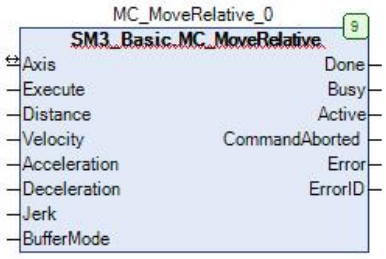
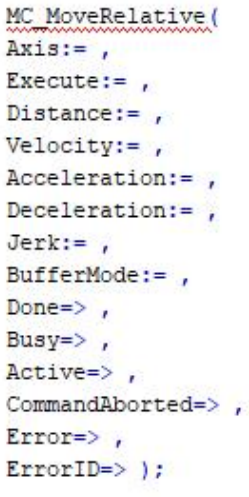
② The current speed and position can be monitored through trace.



3-1-2-9. Relative position [MC_MoveRelative]

(1) Instruction overview

The axis runs in relative position, which is specified by Distance.

Instruction	Name	Graphic representation	ST language
MC_MoveRelative	Relative position		

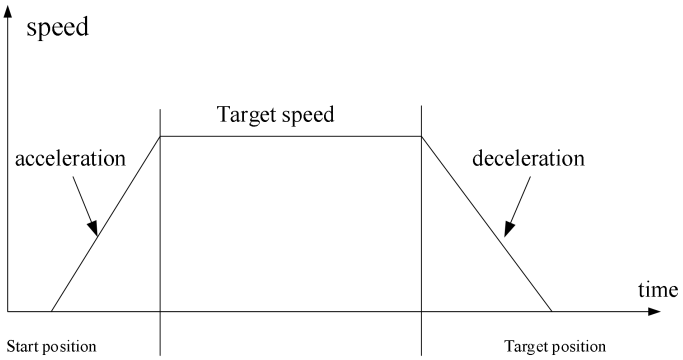

```
MC_MoveRelative(
Axis:= ,
Execute:= ,
Distance:= ,
Velocity:= ,
Acceleration:= ,
Deceleration:= ,
Jerk:= ,
BufferMode:= ,
Done=> ,
Busy=> ,
Active=> ,
CommandAborted=> ,
Error=> ,
ErrorID=> );
```

(2) Related variables

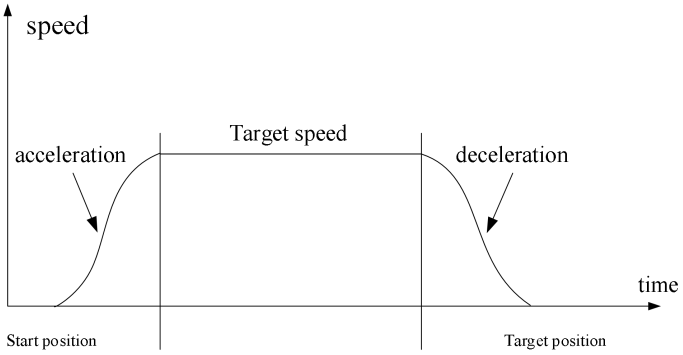
VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis
VAR_INPUT	Name	Data type	Effective range	Initial value	Description

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Execute	Valid	BOOL	TRUE/FALSE	FALSE	Starting execution at the rising edge
Distance	Position	LREAL	Data range	0	The relative distance between the target and its current position
Velocity	Target speed	LREAL	0, positive number	0	Maximum speed [u/s]
Acceleration	Target acceleration speed	LREAL	positive number	0	Acceleration value [u/s^2]
Deceleration	Target deceleration speed	LREAL	positive number	0	Deceleration value [u/s^2]
Jerk	Target jerk speed	LREAL	positive number	0	Jerk value [u/s^3]
BufferMode	Buffer mode	MC_BUFFER_MODE	-	0	If the function block is Busy, only BufferMode=Aborting is allowed
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	Completed	BOOL	TRUE/FALSE	FALSE	TRUE after execution is completed
Busy	Executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Active	In control	BOOL	TRUE/FALSE	FALSE	Change to TRUE in control
CommandAborted	Instruction interrupted	BOOL	TRUE/FALSE	FALSE	True if the command has been terminated by another command
Error	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	Error code	SMC_ERROR	-	0	Error recognition

- The running status of this function block is in Standstill, and the state of the instruction during execution is Discrete Motion. Pay attention to the running status of this axis during instruction execution to avoid interrupting other instructions on this axis or being interrupted by other instructions.
- The start command will start at the rising edge of Execute. This command is valid for repeated rising edges in Discrete Motion, and can refresh the latest Position every time.
- Acceleration or Deceleration is zero, and the instruction operation is in an abnormal state, but the state of the axis is Discrete Motion;
- Trapezoidal acceleration and deceleration action
  Velocity, Acceleration and Deceleration have value but Jerk is 0.



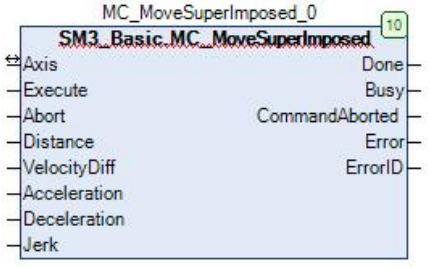
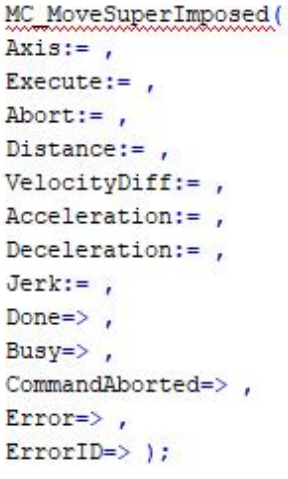
- S curve acceleration and deceleration action
  Velocity, Acceleration, Deceleration and Jerk have value.



3-1-2-10. Superimposed relative motion command [MC_MoveSuperImposed]

(1) Instruction overview

The speed and position data of the axis are superimposed on the running command based on the original command speed and position.

Instruction	Name	Graphic representation	ST language
MC_MoveSuperImposed	Superimp-osed relative motion command		


```
MC_MoveSuperImposed(
Axis:= ,
Execute:= ,
Abort:= ,
Distance:= ,
VelocityDiff:= ,
Acceleration:= ,
Deceleration:= ,
Jerk:= ,
Done=> ,
Busy=> ,
CommandAborted=> ,
Error=> ,
ErrorID=> );
```

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Execute	Valid	BOOL	TRUE/FALSE	FALSE	Starting execution at the rising edge
Abort	Abort	BOOL	TRUE/FALSE	FALSE	Abort ongoing motion and reset all outputs
Distance	Position	LREAL	Data range	0	This data is stacked position data
VelocityDiff	Stacking velocity	LREAL	0, positive number	0	Maximum velocity of superimposed motion [u/s]
Acceleration	Target acceleration speed	LREAL	positive number	0	Acceleration value [u/s^2]
Deceleration	Target deceleration speed	LREAL	positive number	0	Deceleration value [u/s^2]
Jerk	Target jerk speed	LREAL	positive number	0	Jerk value [u/s^3]
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	Completed	BOOL	TRUE/FALSE	FALSE	TRUE after execution is completed
Busy	Executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
CommandAborted	Instruction interrupted	BOOL	TRUE/FALSE	FALSE	True if the command has been terminated by another command
Error	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	Error code	SMC_ERROR	-	0	Error recognition

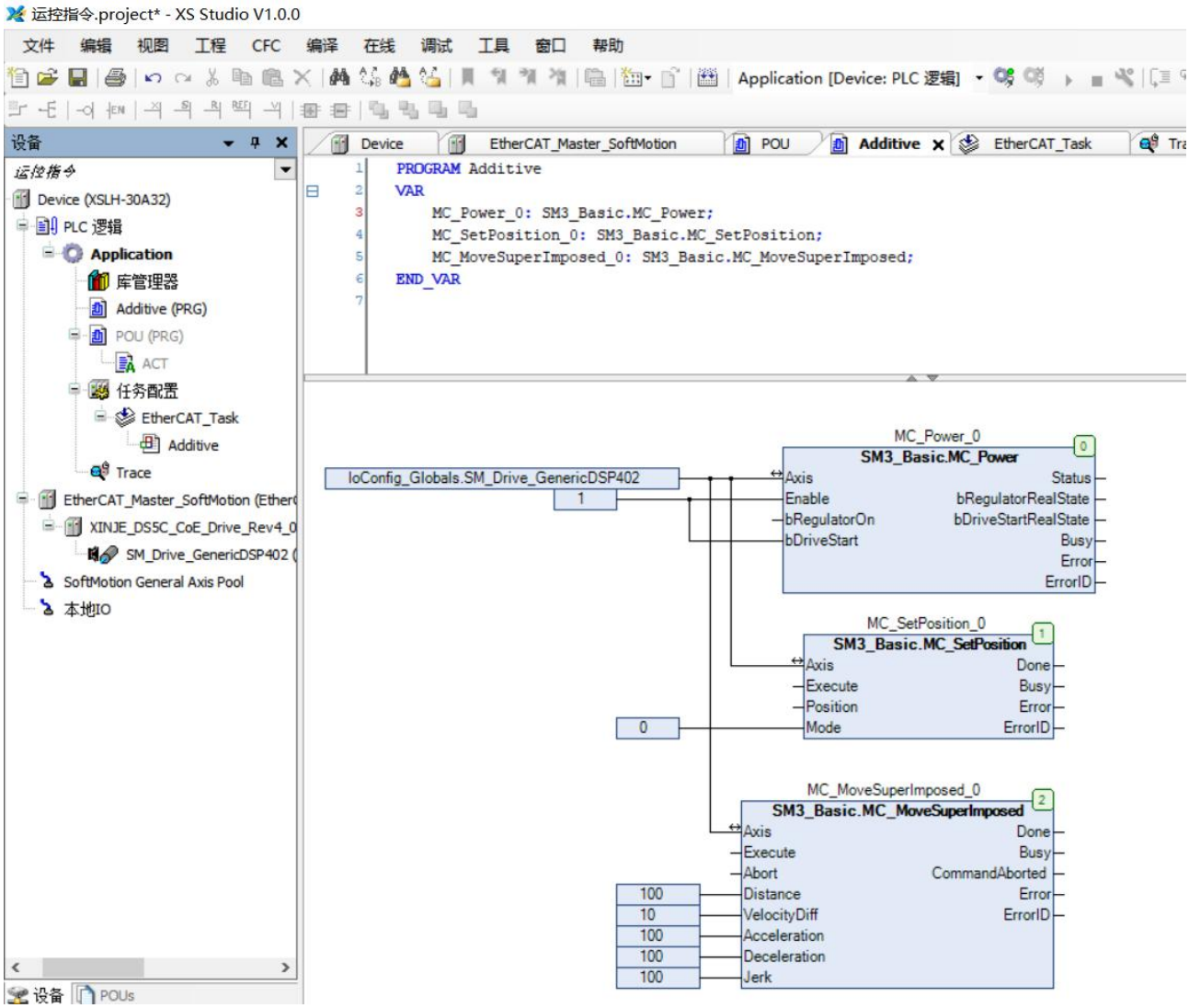
(3) Function description

- This function block is for stacking position and speed commands, while VelocityDiff and Distance represent the speed and position stacked on other commands, respectively;
- MC_ MoveSuperImposed can be superimposed on any other instruction;
- MC_ MoveSuperImposed can also be aborted by MC_ MoveSuperImposed;
- In the StandStill state, the function block MC_ MoveSuperimposed is similar to MC_ MoveRelative;
- The start command will start at the rising edge of Execute.

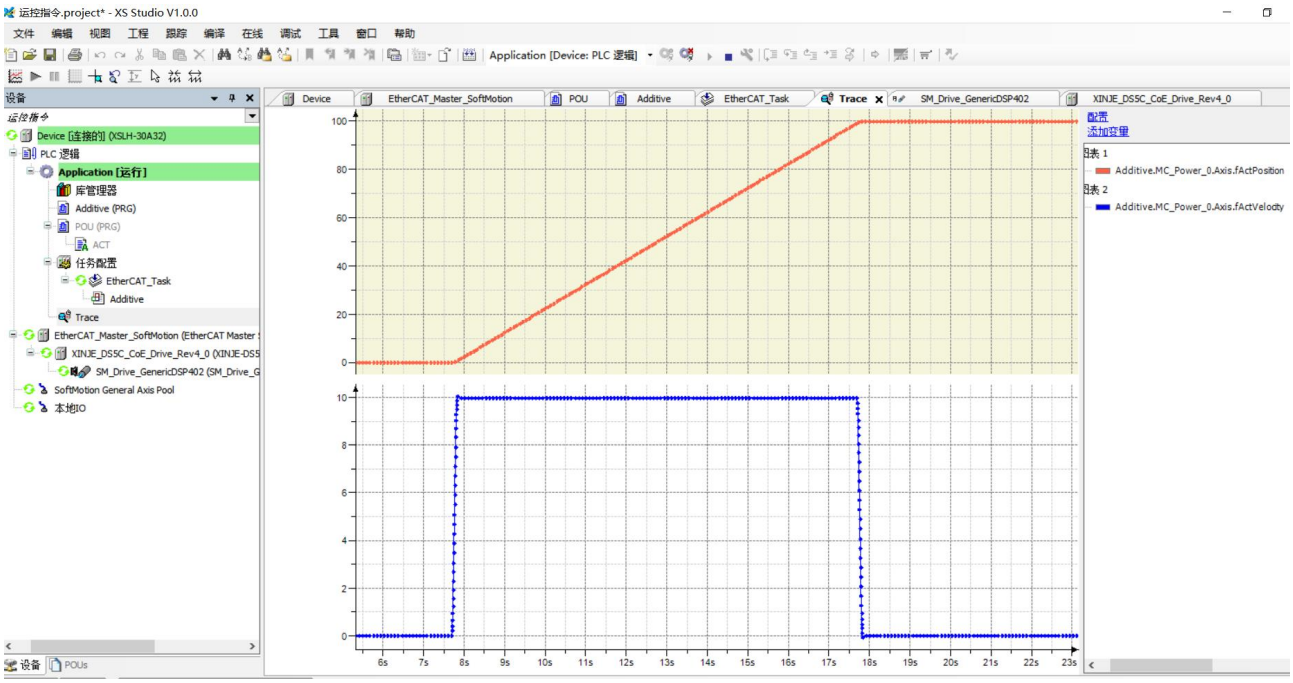
(4) Application

Example 1: It can directly run through "MC_MoveSuperImposed" after "MC_Power" is enabled.

Programming: You can first set the position to 0 through "MC_SetPosition", enable it, and then conduct MC_ MoveSuperImplied, can be configured with a running speed of 10 and a running position of 100.

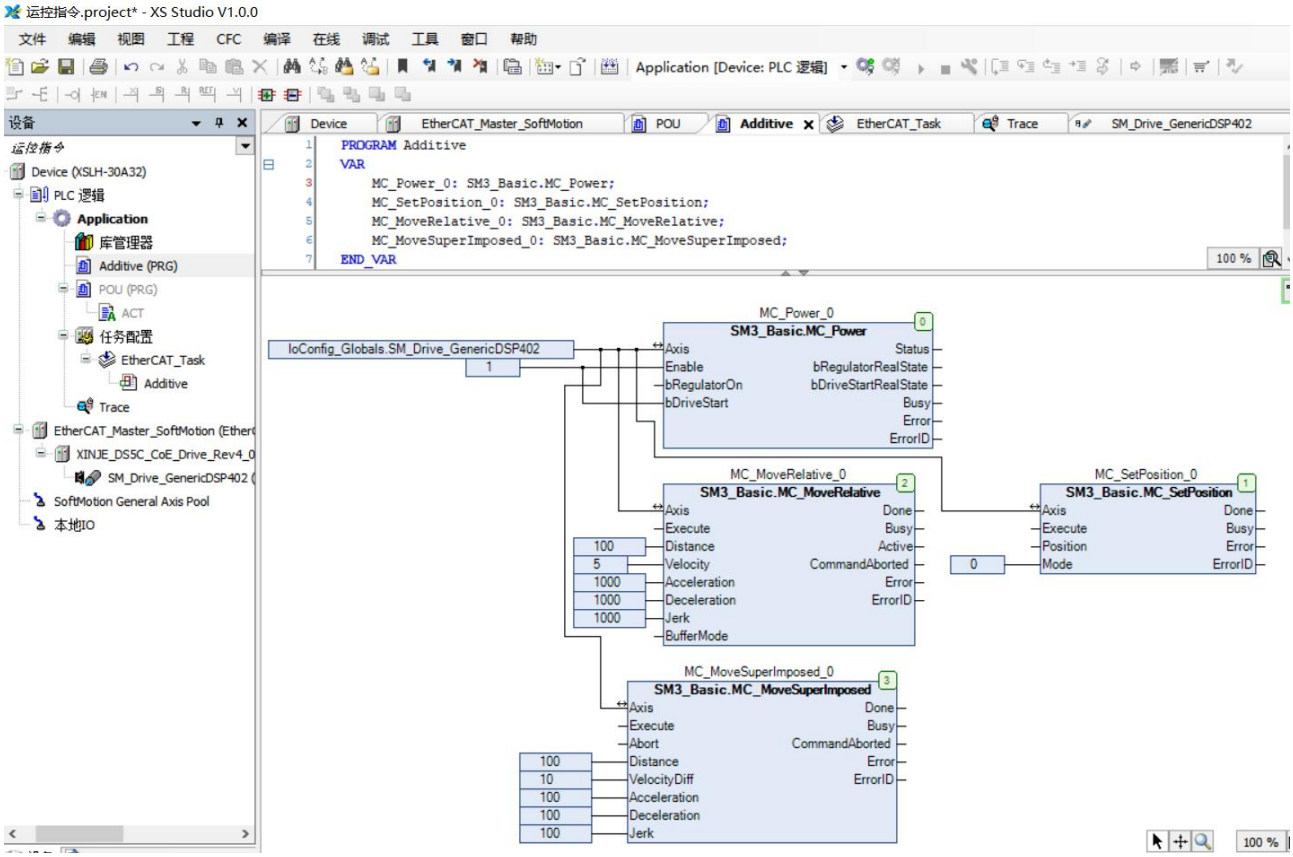


The current speed and position can be monitored through trace.

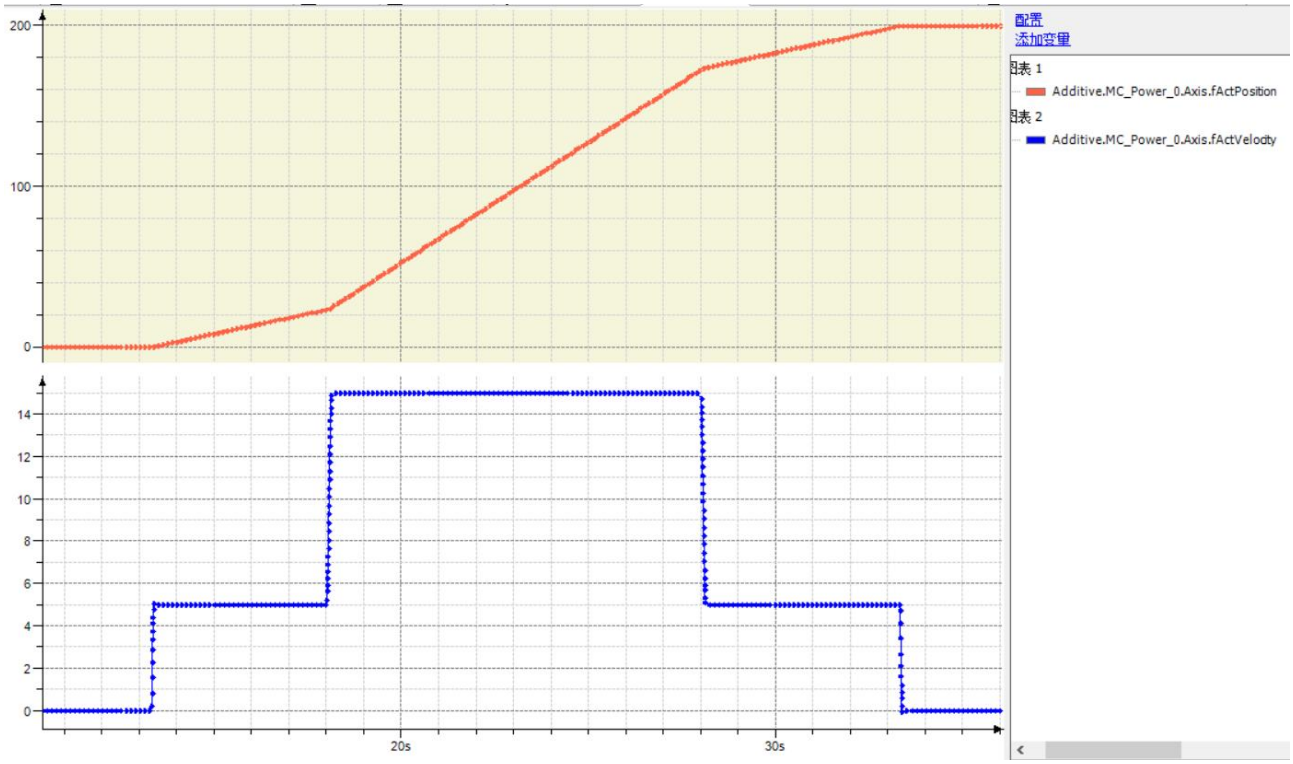


Example 2: After "MC_Power" is enabled, execute "MC_MoveRelative" first, and then "MC_MoveSuperImposed".

Programming: "MC_MoveRelative" is executed at a speed of 5u/s and a target position of 100u. MC_MoveSuperImposed can be turned on during operation or after waiting for the execution of "MC_MoveRelative" to complete. The speed set for "MC_MoveSuperImposed" is 10u/s and the target position is 100u. The position after turning on is the sum of the target positions set in the two instructions. MC_MoveSuperImposed instruction overlays both position and speed.



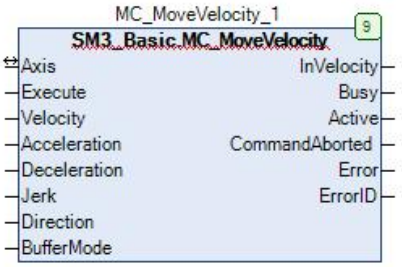
The current speed and position can be monitored through trace.



3-1-2-11. Speed control [MC_MoveVelocity]

(1) Instruction overview

This function block moves infinitely at a specified speed.

Instruction	Name	Graphic representation	ST language
MC_MoveVelocity	Speed control command		MC_MoveVelocity(Axis:= , Execute:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Direction:= , BufferMode:= , InVelocity=> , Busy=> , Active=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis
VAR_INPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Execute	Valid	BOOL	TRUE/FALSE	FALSE	Starting execution at the rising edge
Velocity	Speed	LREAL	0, positive value	0	Speed operation value [u/s]
Acceleration	Target acceleration speed	LREAL	0, positive value	0	Acceleration value [u/s^2]
Deceleration	Target deceleration speed	LREAL	0, positive value	0	Deceleration value [u/s^2]
Jerk	Target jerk speed	LREAL	0, positive value	0	Jerk value [u/s^3]
Direction	Direction	MC_DIRECTION	3: fastest 2: current 1: Positive 0: shortest -1: Negative	0	Fastest: Automatically select the fastest direction to move Current: Move in the current direction Positive: move forward Shortest: Select direction based on the shortest path Negative: move reverse
BufferMode	Buffer mode	MC_BUFFER_MODE	-	0	If the function block is Busy, only

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
					BufferMode=Aborting is allowed
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
InVelocity	Flag for reaching the set speed	BOOL	TRUE/FALSE	FALSE	After reaching the set speed, it is set to TRUE
Busy	Executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Active	In control	BOOL	TRUE/FALSE	FALSE	Change to TRUE in control
Command Aborted	Command is aborted	BOOL	TRUE/FALSE	FALSE	True if the command has been terminated by another command
Error	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	Error code	SMC_ERROR	-	0	Error recognition

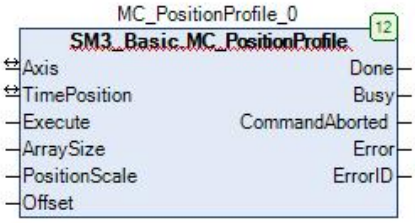
(3) Function description

- Change the Velocity parameter to control the speed of the drive.
- The execution of the function block must have a condition of rising edge.
- The InVelocity of the function block indicates that the operating speed of the instruction has reached the set value.
- The Busy of the function block indicates that the current function block is currently executing.

3-1-2-12. Position profile [MC_PositionProfile]

(1) Instruction overview

Users can plan their own "time - position" data table, and the controller will complete the motion according to the planned data.

Instruction	Name	Graphic representation	ST language
MC_PositionProfile	Position profile command		MC_PositionProfile(Axis:= , TimePosition:= , Execute:= , ArraySize:= , PositionScale:= , Offset:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis
TimePosition	Data table	MC_TP_REF	-	-	User planned time location data table
VAR_INPUT	Name	Data type	Effective range	Initial value	Description

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Execute	Valid	BOOL	TRUE/FALSE	FALSE	Starting execution at the rising edge
ArraySize	Dynamic array	INT	Positive value	0	Number of arrays used in the running profile
PositionScale	Comprehensive factor	LREAL	Data range	1	Overall Position Scale Factor
Offset	Offset	LREAL	Data range	0	Position offset
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	Completed	BOOL	TRUE/FALSE	FALSE	TRUE after execution is completed
Busy	Executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
CommandAborted	Command is aborted	BOOL	TRUE/FALSE	FALSE	True if the command has been terminated by another command
Error	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	Error code	SMC_ERROR	-	0	Error recognition

(3) Function description

- This function block is a profile motion model for time periods and positions, running in Discrete Motion mode according to the data set by the user in the TimePosition variable.
- The running status of this function block is in Standstills, and the state of the instruction running is Discrete Motion. Other states cannot be run.
- The command starts at the rising edge of Execute, and this command runs repeatedly in Discrete Motion.
- TimePosition is MC_ TP_ REF data type;

The specific description of MC_TP_REF is as follows:

Member	Type	Initial value	Description
Number_of_pairs	INT	0	Number of segments in the contour path
IsAbsolute	BOOL	TRUE	Absolute motion (TRUE) and relative motion selection
MC_TP_ArrayARRAY[1..N] OF SMC_TP			Array of time and location

The specific description of SMC_TP is as follows;

Member	Type	Initial value	Description
delta_time	TIME	TIME#0ms	Time of position segment
position	LREAL	0	Current position value

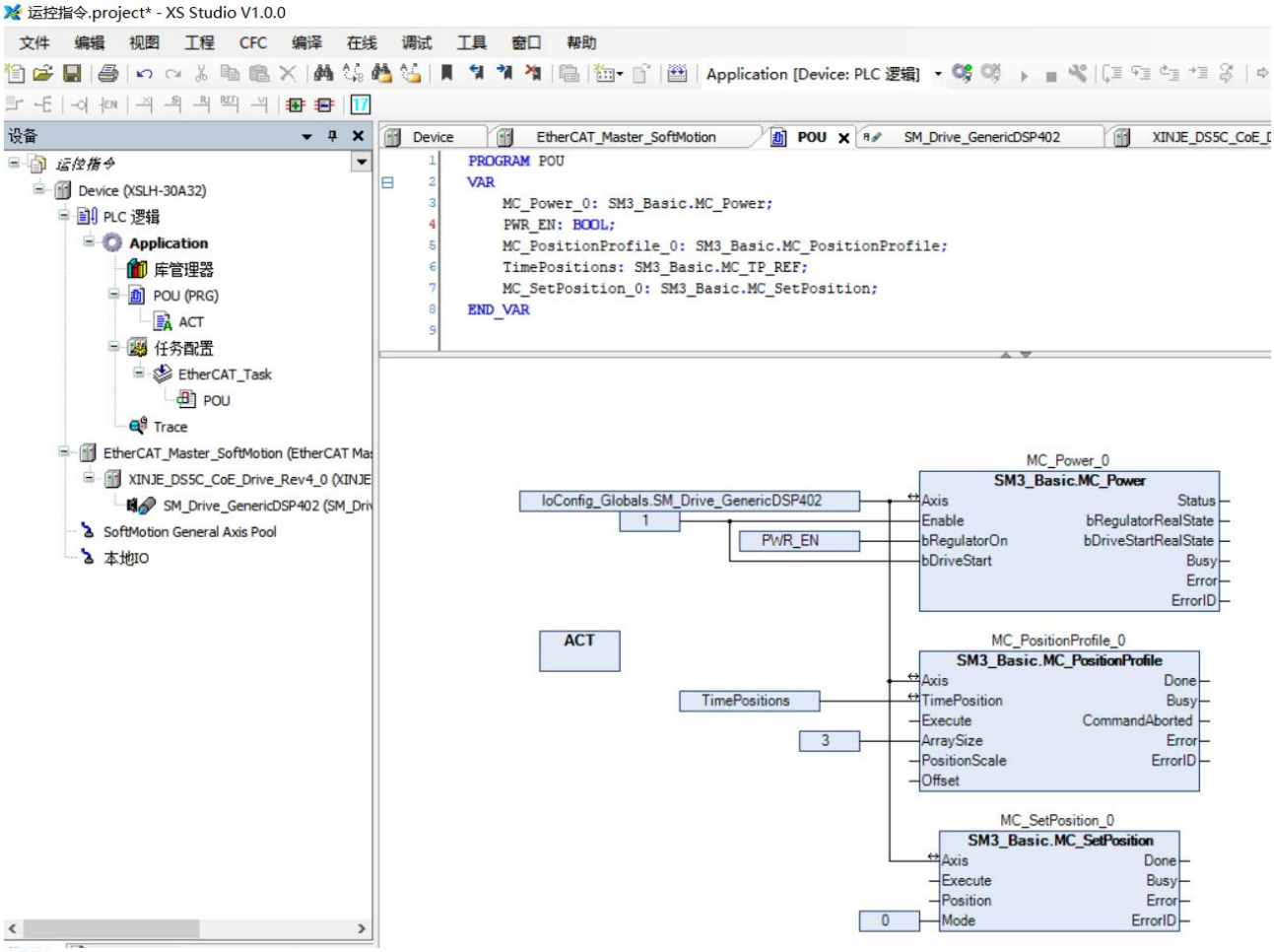
Note: When the speed corresponding to the set position data changes, relevant adjustments are made according to the S-curve.

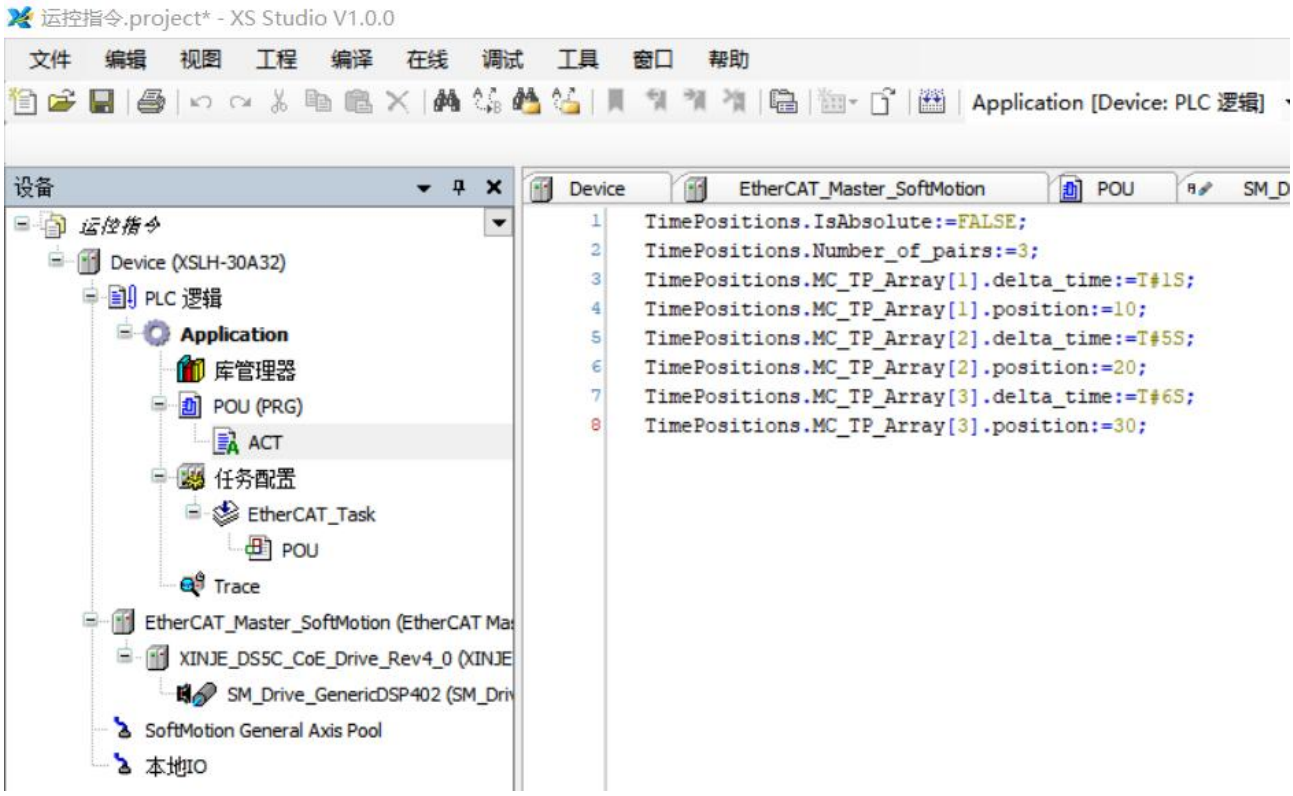
(4) Application

Example 1: Set the execution of three segments and set them as relative motion. The first segment runs at a position of 10 within 1 second, the second segment runs at a position of 20 within 5 seconds, and the third segment runs at a position of 30 within 6 seconds. The total duration of these three segments is 12 seconds, and the total distance of execution is 60.

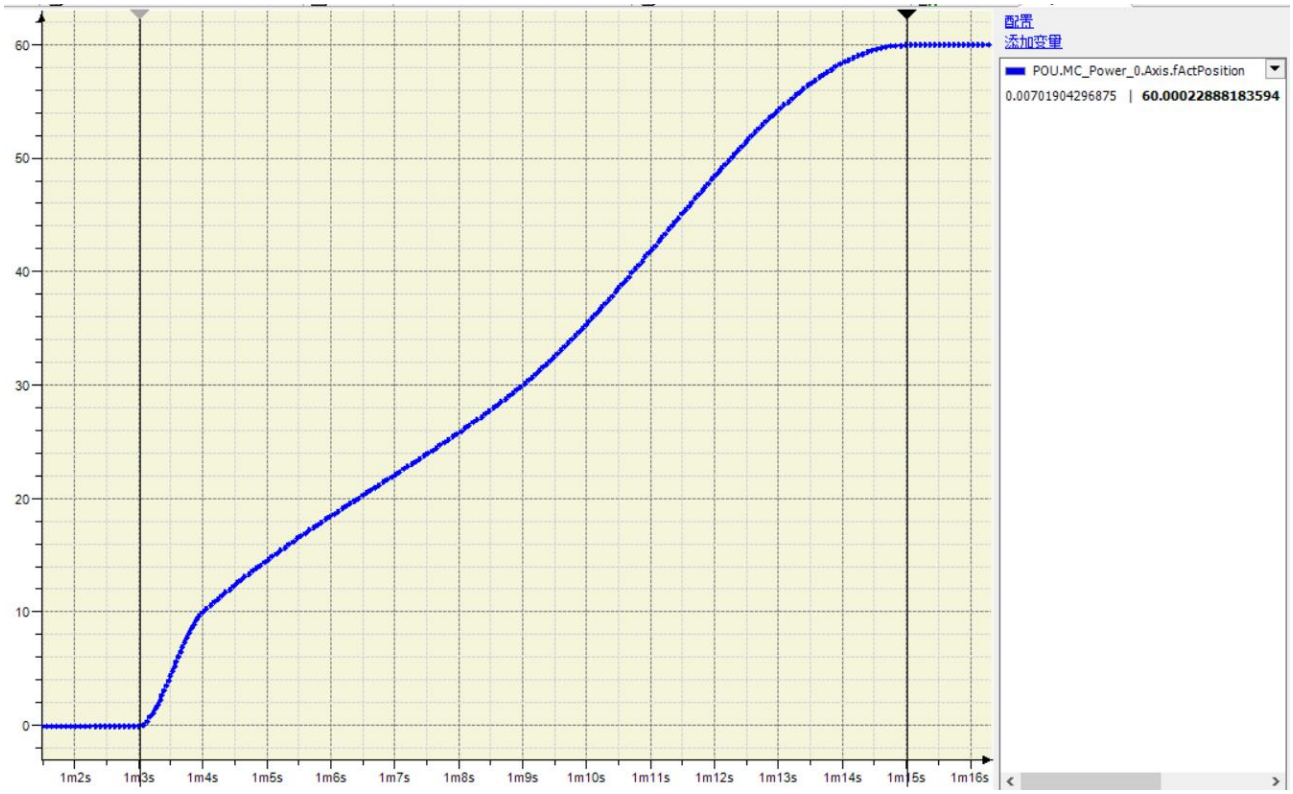
Write a program using the "MC_Power" and "MC_PositionProfile" instructions, enable it, and then set and run the three segments as the command MC_ PositionProfile. Set the position to 0 through the command

MC_SetPosition before executing. In ACT (Action Properties of POU), set the time and position.





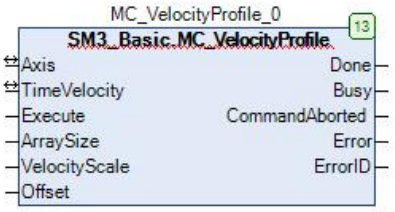
You can set 'Trace' to view the execution time and location.


配置
添加变量
POU.MC_Power_0.Axis.fActPosition
0.00701904296875 | 60.00022888183594
60
50
40
30
20
10
0
1m2s
1m3s
1m4s
1m5s
1m6s
1m7s
1m8s
1m9s
1m10s
1m11s
1m12s
1m13s
1m14s
1m15s
1m16s

3-1-2-13. Speed profile [MC_VelocityProfile]

(1) Instruction overview

Similar to MC_PositionProfile, MC_VelocityProfile plans motion by defining "time - velocity" data.

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

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis
TimePosition	Data table	MC_TP_REF	-	-	User planned time - speed data table
VAR_INPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Execute	Valid	BOOL	TRUE/FALSE	FALSE	Starting execution at the rising edge
ArraySize	Dynamic array	INT	Positive value	0	Number of arrays used in the running profile
VelocityScale	Speed factor	LREAL	Data range	1	Scale factor of speed
Offset	Offset	LREAL	Data range	0	Speed offset
VAR_OUTPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Done	Completed	BOOL	TRUE/FALSE	FALSE	TRUE after execution is completed
Busy	Executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
CommandAborted	Command is aborted	BOOL	TRUE/FALSE	FALSE	True if the command has been terminated by another command
Error	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	Error code	SMC_ERROR	-	0	Error recognition

(3) Function description

- This function block is a contour motion model for time periods and speeds, running in Continuous Motion mode according to the data set by the user in the TimeVelocity variable.
- The running status of this function block is in Standstills, and the state of the instruction running is Discrete Motion. Other states cannot be run.
- The command will start at the rising edge of Execute, this instruction runs repeatedly in Discrete Motion.
- TimeVelocity is MC_TV_REF data type:

The detail description of MC_TP_REF is as follows:

Member	Type	Initial	Description

		value	
Number_of_pairs	INT	0	Number of segments in the contour path
IsAbsolute	BOOL	TRUE	Absolute motion (TRUE) and relative motion selection
MC_TP_ArrayARRAY[1..N] OF SMC_TP			Array of time and location

The detail description of SMC_TP is as follows:

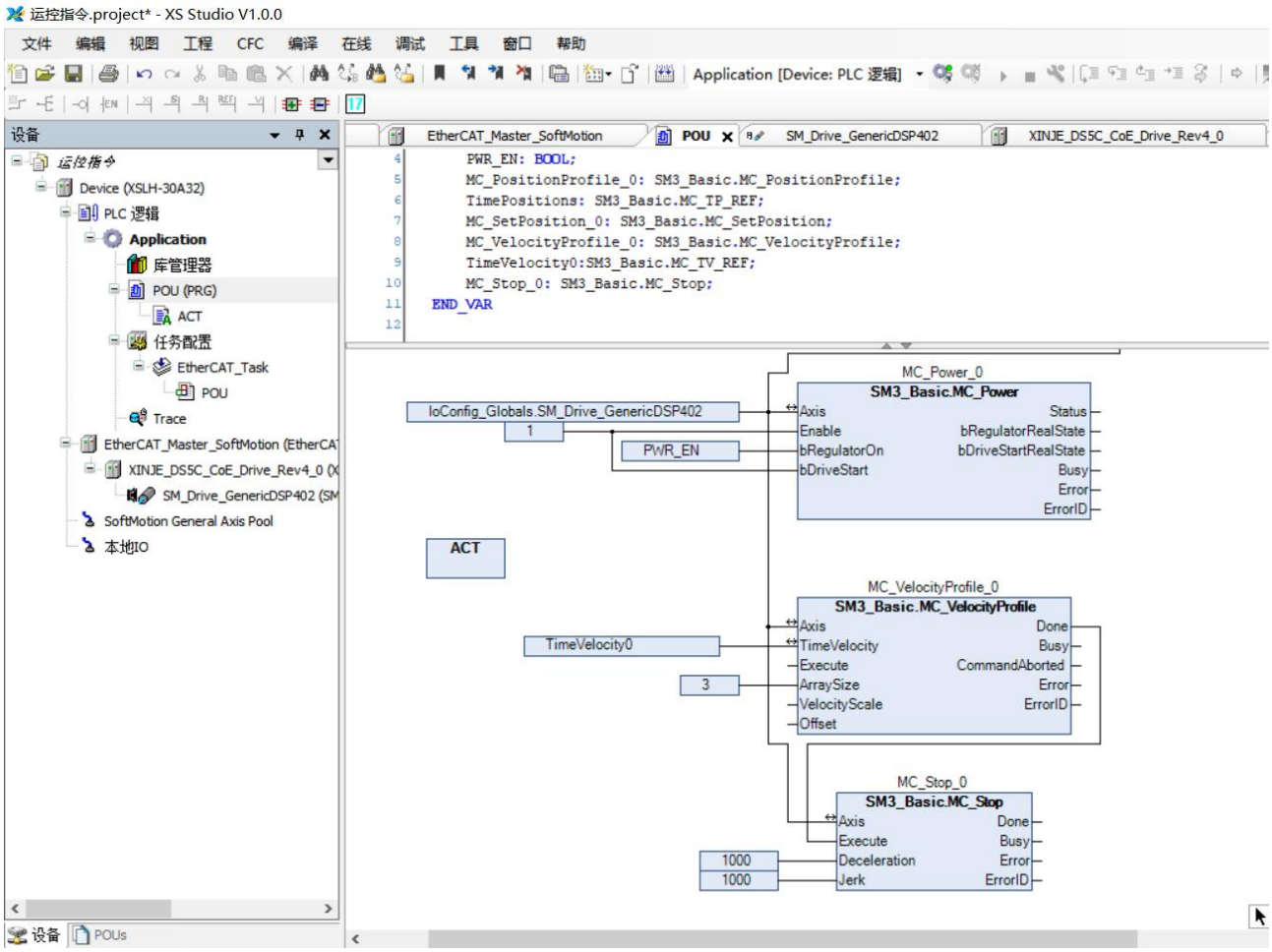
Member	Type	Initial value	Description
delta_time	TIME	TIME#0ms	Time of position segment
position	LREAL	0	Current position value

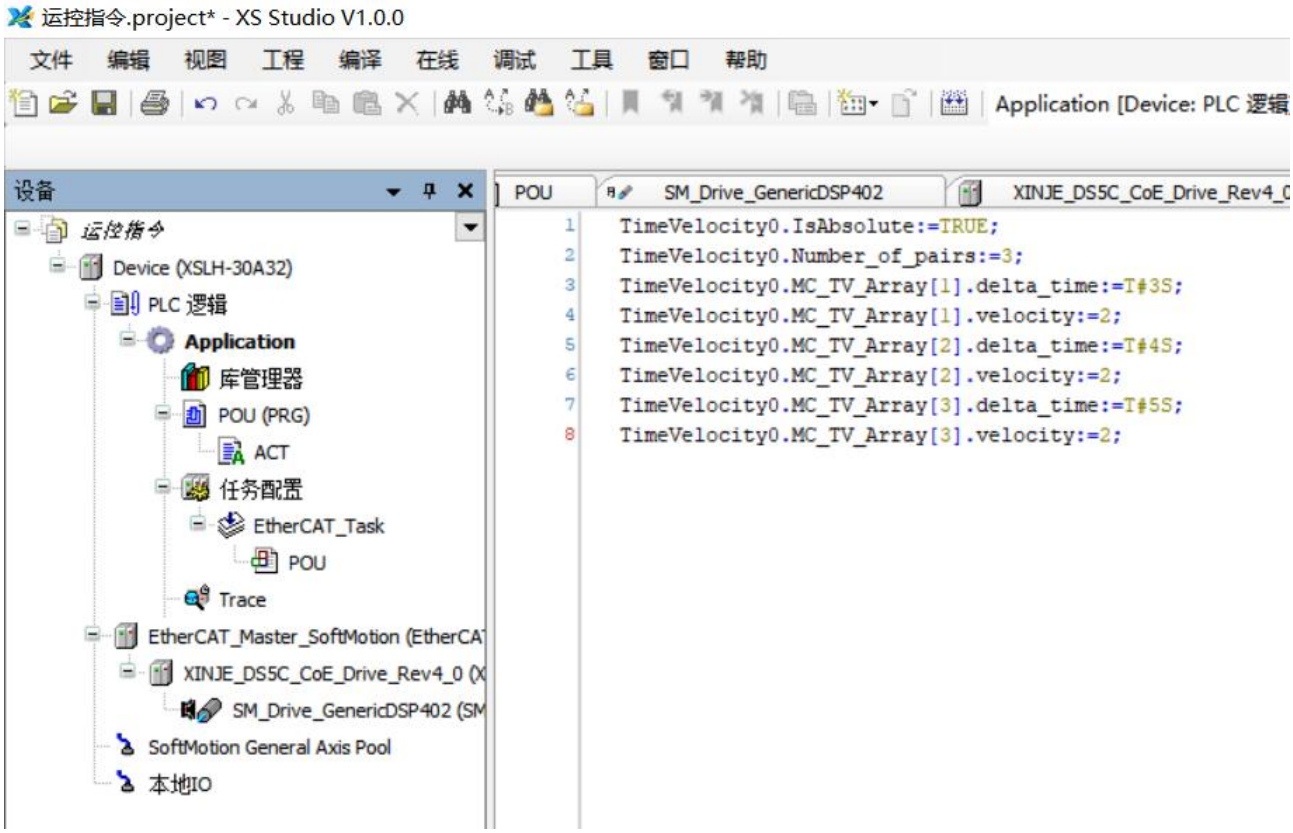
Note: The entire speed process is calculated using an S-curve acceleration and deceleration method, and the speed of each contour segment is calculated using a superposition method; When instructions are repeatedly run, the speed is also stacked to avoid exceeding the speed limit during instruction usage; Repeated operation must return the state of this axis to the Standstill state.

(4) Application

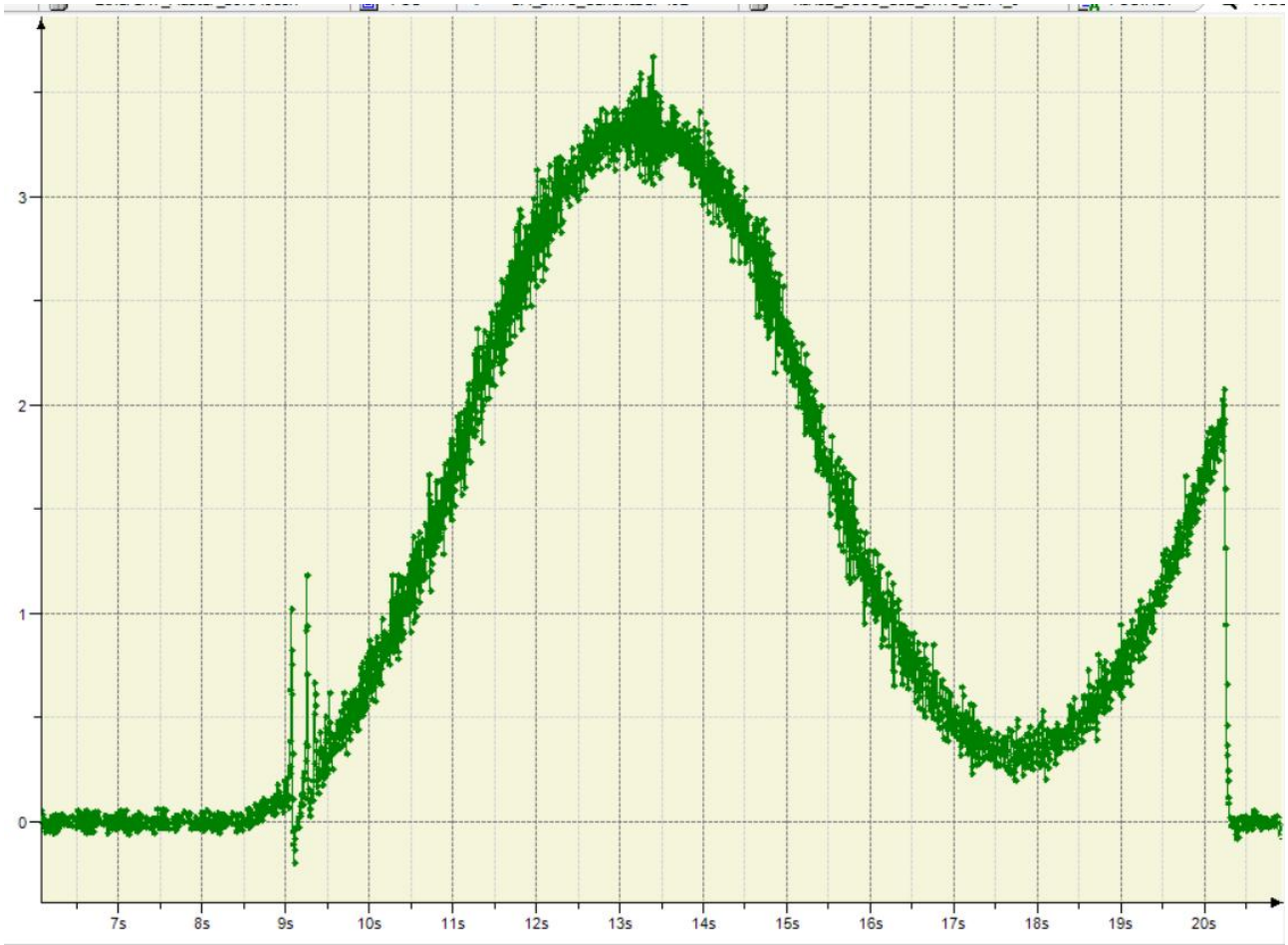
Example 1: Set the execution of three segments, with absolute motion. The first segment reaches a speed of 2 after 3 seconds, the second segment reaches a speed of 2 after 4 seconds, and the third segment reaches a speed of 2 after 5 seconds. The total duration of these three segments is 12 seconds, and the final execution speed is 2.

Make the program by using “MC_Power”, “MC_VelocityProfile”, “MC_Stop” instructions. After enabled, run the three segments as the command setting of MC_VelocityProfile. In ACT (POU action), set the time and speed.





You can view time and speed in Trace.

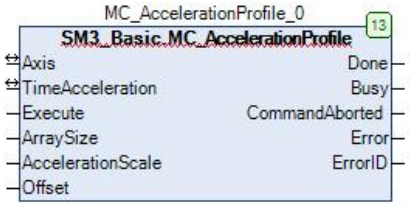


Note: when MC_VelocityProfile is Done, the speed is not 0. So you need to add STOP or other operation and control commands later.

3-1-2-14. Acceleration profile [MC_AccelerationProfile]

(1) Instruction overview

Simliar toMC_PositionProfile instruction, MC_AccelerationProfile plans motion by defining "time - acceleration" data.

Instruction	Name	Graphic representation	ST language
MC_AccelerationProfile	Acceleration profile comand		MC_AccelerationProfile(Axis:= , TimeAcceleration:= , Execute:= , ArraySize:= , AccelerationScale:= , Offset:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis
TimeAcceleration	Data table	MC_TA_REF	-	-	Reference time/acceleration description
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Execute	Valid	BOOL	TRUE/FALSE	FALSE	Starting execution at the rising edge
ArraySize	Dynamic array	INT	Positive value	0	Number of arrays used in the running profile
AccelerationScale	Comprehensive factor	LREAL	Data range	1	Scale factor for acceleration or deceleration
Offset	Offset	LREAL	Data range	0	Acceleration offset
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	Completed	BOOL	TRUE/FALSE	FALSE	TRUE after execution is completed
Busy	Executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
CommandAborted	Command is aborted	BOOL	TRUE/FALSE	FALSE	True if the command has been terminated by another command
Error	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	Error code	SMC_ERROR	-	0	Error recognition

(3) Function description

- This function block is a contour motion model for the time period and acceleration/deceleration. The running mode is Discrete Motion, which runs according to the data set in the user's TimeAcceleration variable.
- The running status of this function block is in Standstills, and the state of the instruction running is Discrete Motion. Other states cannot be run.
- The command will start at the rising edge of Execute, and the repeated running speed of this command in Discrete Motion is the superposition of the previous one, which can easily cause system

faults.

- TimeVelocity is MC_TV_REF data type.

The specific description of MC_TP_REF is as follows:

Member	Type	Initial value	Description
Number_of_pairs	INT	0	Number of segments in the contour path
IsAbsolute	BOOL	TRUE	Absolute motion (TRUE) and relative motion selection
MC_TP_ArrayARRAY[1..N] OF SMC_TP			Array of time and location

The specific description of SMC_TP is as follows:

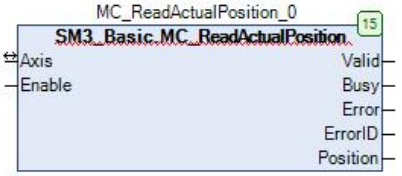
Member	Type	Initial value	Description
delta_time	TIME	TIME#0ms	Time of position segment
position	LREAL	0	Current position value

Note: The set acceleration is reflected in the change in velocity, and all acceleration changes follow the S-curve. The acceleration data from the final result change to [starting acceleration is A, ending acceleration is B] (A+B)/2 is reflected in the final velocity.

3-1-2-15. Read actual position [MC_ReadActualPosition]

(1) Instruction overview

Used to read the current actual position value of the axis.

Instruction	Name	Graphic representation	ST language
MC_ReadActualPosition	read the actual position		MC_ReadActualPosition(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , Position=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Enable	Valid	BOOL	TRUE/FALSE	FALSE	Read the current position of the servo for the true state
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Valid	Obtain flag	BOOL	TRUE/FALSE	FALSE	If the output value is valid, then it is TRUE

Busy	Executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	Error code	SMC_ERROR	-	0	Error recognition
Position	Obtained axis position	LREAL	Data range	0	Axis position data read from instructions

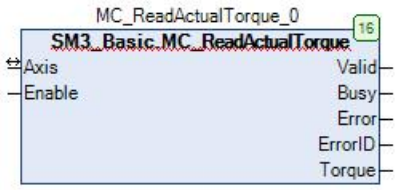
(3) Function description

- Read the actual position command in the driver through this command, which is the Enable level enable effect. Instructions can be used multiple times without affecting each other.

3-1-2-16. Read current torque [MC_ReadActualTorque]

(1) Instruction overview

Used to read the actual torque value of the shaft.

Instruction	Name	Graphic representation	ST language
MC_ReadActualTorque	Read the current torque value		MC_ReadActualTorque(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , Torque=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Enable	Valid	BOOL	TRUE/FALSE	FALSE	Must be set to TRUE to activate the processing of the function block
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Valid	Obtain flag	BOOL	TRUE/FALSE	FALSE	If the output value is valid, then it is TRUE
Busy	Executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	Error code	SMC_ERROR	-	0	Error recognition
Torque	Current torque value obtained	LREAL	Data range	0	The current torque data read by the command

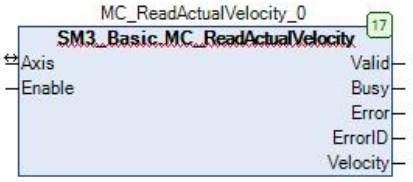
(3) Function description

- Reads the current torque value instruction in the driver through MC_ReadActualTorque, which is the enable effect of the Enable level. Instructions can be used multiple times without affecting each other.

3-1-2-17. Read the current speed [MC_ReadActualVelocity]

(1) Instruction overview

Used to read the actual speed value of the axis.

Instruction	Name	Graphic representation	ST language
MC_ReadActualVelocity	Read the current speed		MC_ReadActualVelocity(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , Velocity=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Enable	Valid	BOOL	TRUE/FALSE	FALSE	Must be set to TRUE to activate the processing of the function block
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Valid	Obtain flag	BOOL	TRUE/FALSE	FALSE	If the output value is valid, then it is TRUE
Busy	Executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	Error code	SMC_ERROR	-	0	Error recognition
Velocity	Current speed value obtained	LREAL	Data range	0	The current speed data read by the instruction

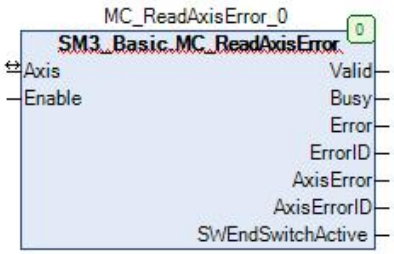
(3) Function description

- Reads the current speed value instruction in the driver through MC_ReadActualVelocity, which is the effect of the Enable level. Instructions can be used multiple times without affecting each other.

3-1-2-18. Read axis error status [MC_ReadAxisError]

(1) Instruction overview

Read the axis error.

Instruction	Name	Graphic representation	ST language
SMC_AxisReadSettingsualVelocity	Read the axis error status		MC_ReadAxisError(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , AxisError=> , AxisErrorID=> , SWEndSwitchActive=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Enable	Valid	BOOL	TRUE /FALSE	FALSE	Must be set to TRUE to activate the processing of the function block
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Valid	Obtain flag	BOOL	TRUE/FALSE	FALSE	If the output value is valid, then it is TRUE
Busy	Executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	Error code	SMC_ERROR	-	0	Error recognition
AxisError	Axis error flag	BOOL	TRUE/FALSE	FALSE	Axis error flag
AxisErrorID	Axis error code	DWORD	Positive value, 0	0	Read axis error code
SWEndSwitchActive	Soft limit	BOOL	TRUE/FALSE	FALSE	If the soft limit is exceeded, it is TRUE

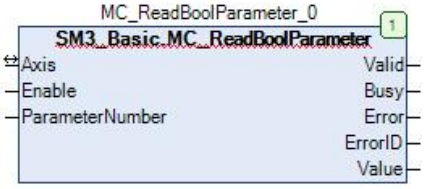
(3) Function description

- Reads the error code in the driver through MC_ReadAxisError, and the instruction is Enable level effect. Instructions can be used multiple times without affecting each other.

3-1-2-19. Read the axis bit parameter [MC_ReadBoolParameter]

(1) Instruction overview

Read the value of the specified BOOL type variable.

Instruction	Name	Graphic representation	ST language
MC_ReadBoolParameter	Read the axis bit parameters		MC_ReadBoolParameter(Axis:= , Enable:= , ParameterNumber:= , Valid=> , Busy=> , Error=> , ErrorID=> , Value=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis
VAR_INPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Enable	Valid	BOOL	TRUE/FALSE	FALSE	Must be set to TRUE to activate the processing of the function block
ParameterNumber	Axis parameter number	DINT	Positive value, 0	0	Parameter number
VAR_OUTPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Valid	Obtain flag	BOOL	TRUE/FALSE	FALSE	If the output value is valid, then it is TRUE
Busy	Executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	Error code	SMC_ERROR	-	0	Error recognition
Value	Parameter value	BOOL	TRUE/FALSE	FALSE	Read the value of the parameter

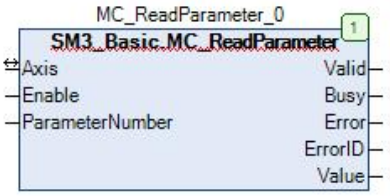
(3) Function description

- Reads the bit data status in the driver through MC_ ReadBoolParam, and the instruction is Enable level effect. Instructions can be used multiple times without affecting each other.

3-1-2-20. Read the axis parameter [MC_ReadParameter]

(1) Instruction overview

Used to read the specified parameter value.

Instruction	Name	Graphic representation	ST language
MC_ReadParameter	Read the axis parameter		MC_ReadParameter(Axis:= , Enable:= , ParameterNumber:= , Valid=> , Busy=> , Error=> , ErrorID=> , Value=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis
VAR_INPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Enable	Valid	BOOL	TRUE/FALSE	FALSE	Must be set to TRUE to activate the processing of the function block
ParameterNumber	Axis parameter number	DINT	Positive value, 0	0	Parameter number
VAR_OUTPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Valid	Obtain flag	BOOL	TRUE/FALSE	FALSE	If the output value is valid, then it is TRUE
Busy	Executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	Error code	SMC_ERROR	-	0	Error recognition
Value	Parameter value	LREAL	Data range	0	Read the value of the parameter

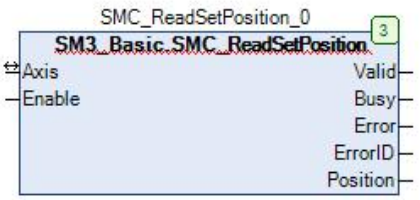
(3) Function description

- Reads the bit data status in the driver through MC_ReadParam, and the instruction is Enable level effect. Instructions can be used multiple times without affecting each other.

3-1-2-21. Read axis instruction position [SMC_ReadSetPosition]

(1) Instruction overview

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

Instruction	Name	Graphic representation	ST language
SMC_ReadSetPosition	Read axis command position		SMC_ReadSetPosition(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , Position=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Enable	Valid	BOOL	TRUE/FALSE	FALSE	Must be set to TRUE to activate the processing of the function block
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Valid	Obtain flag	BOOL	TRUE/FALSE	FALSE	If the output value is valid, then it is TRUE
Busy	Executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	Error code	SMC_ERROR	-	0	Error recognition
Position	Position	LREAL	Data range	0	Set position value

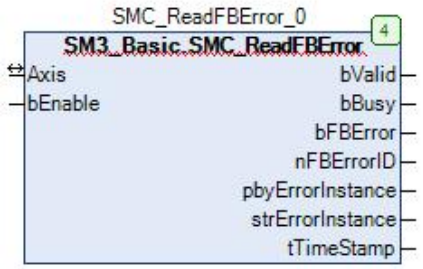
(3) Function description

- Enable is true, valid if there are no errors, and Busy output is TURE.
- The output value of Position is the value of Axis.fSetPosition.
- If Enable is changed to FALSE, then Valid and Busy outputs as FALSE. Position remains at the value before FALSE.

3-1-2-22. Read function block error [SMC_ReadFBError]

(1) Instruction overview

Read error messages of the axis function block.

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

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
bEnable	Valid	BOOL	TRUE/FALSE	FALSE	Must be set to TRUE to activate the processing of the function block
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
bValid	Obtain flag	BOOL	TRUE/FALSE	FALSE	If the output value is valid, then it is TRUE
bBusy	Executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
bFBError	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
nFBErrorID	Error code	SMC_ERROR	-	0	Error recognition
pbyErrorInstance	Error pointer	POINTER TO BYTE	-	-	Pointer pointing to error reporting function block
strErrorInstance	Error pointer	STRING	-	-	-
tTimeStamp	Time stamp	TIME	-	0	The timestamp at which the error occurred

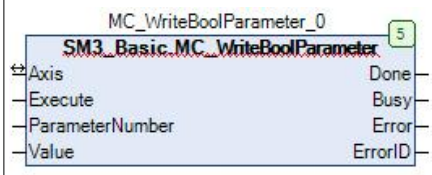
(3) Function description

- Enable is true, valid if there are no errors, and Busy output is TURE.
- If there is a function block alarm, the output of bFBError is true.
- If Enable is changed to FALSE, then Valid and Busy outputs as FALSE.

3-1-2-23. Set axis bit parameter [MC_WriteBoolParameter]

(1) Instruction overview

Write parameter values of type BOOL.

Instruction	Name	Graphic representation	ST language
MC_WriteBoolParameter	Set the axis bit parameter		MC_WriteBoolParameter(Axis:= , Execute:= , ParameterNumber:= , Value:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis
VAR_INPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Execute	Valid	BOOL	TRUE/FALSE	FALSE	The rising edge of the input value will initiate the execution of the function block
ParameterNumber	Parameter number	DINT	Positive number, 0	0	Parameter ID
Value	Parameter value	BOOL	TRUE/FALSE	FALSE	-
VAR_OUTPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Done	Completed	BOOL	TRUE/FALSE	FALSE	True if the parameter value has been successfully written
Busy	Executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	Error code	SMC_ERROR	-	0	Error recognition

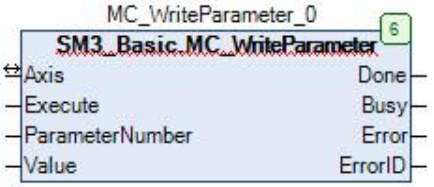
(3) Function description

- Sets the bit parameter of the axis through MC_ WriteBoolParameter, and the instruction is triggered at the rising edge of Execute. Instructions can be used multiple times without affecting each other.

3-1-2-24. Set the axis parameter [MC_WriteParameter]

(1) Instruction overview

Write the specified parameter value.

Instruction	Name	Graphic representation	ST language
MC_WriteParameter	Set the axis parameter		MC_WriteParameter(Axis:= , Execute:= , ParameterNumber:= , Value:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Execute	Valid	BOOL	TRUE/FALSE	FALSE	The rising edge of the input value will initiate the execution of the function block
ParameterNumber	Parameter number	DINT	Positive value, 0	0	Parameter ID
Value	Parameter value	LREAL	Data range	-	Write the value that needs to be set
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	Completed	BOOL	TRUE/FALSE	FALSE	True if the parameter value has been successfully written
Busy	Executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	Error code	SMC_ERROR	-	0	Error recognition

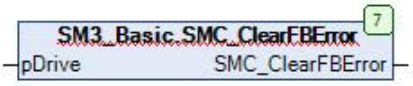
(3) Function description

- Set the axis parameter through MC_WriteParameter, the instruction is triggered at the rising edge of Execute. Instructions can be used multiple times without affecting each other.

3-1-2-25. Clear the error [SMC_ClearFBError]

(1) Instruction overview

Clear the historical error information of the function block.

Instruction	Name	Graphic representation	ST language
SMC_ClearFBError	Clear error		SMC_ClearFBError(pDrive:=)

(2) Related variables

VAR_INPUT	Name	Data type	Effective range	Initial value	Description
pDrive	Axis pointer	POINTR TO AXIS_REF_SM3	-	-	Mapping to the axis
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
SMC_ClearFBError	Valid	BOOL	TRUE/FALSE	FALSE	Clear error is true

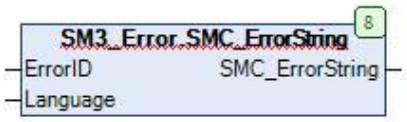
(3) Function description

- When an error occurs on the axis, after calling the reset function block to reset the axis, it is necessary to call the function block to clear the historical error status of the axis. The rising edge of Execute will trigger the execution of this instruction.

3-1-2-26. Read the error [SMC_ErrorString]

(1) Instruction overview

Read the error description information corresponding to the error code.

Instruction	Name	Graphic representation	ST language
SM3_Error.SMC_ErrorString	Read the error		SMC_ErrorString(ErrorID:= , Language:=)

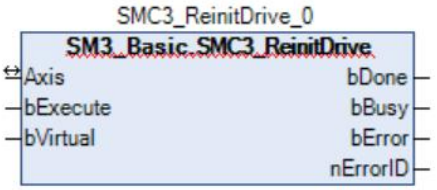
(2) Related variables

VAR_INPUT	Name	Data type	Effective range	Initial value	Description
ErrorID	Error code	SMC_ERROR	-	-	Error recognition
Language	Language	-	-	-	Required language
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
SMC_ErrorString	Error	STRING(100)	-	-	Error description information

3-1-2-27. Recall driver/axis [SMC3_ReinitDrive]

(1) Instruction overview

This function block calls the driver/axis again. This means that the startup phase runs again and the application cannot control the driver until the function block is set to bDone:=TRUE.

Instruction	Name	Graphic representation	ST language
SMC3_ReinitDrive	Recall driver/axis		SMC3_ReinitDrive(Axis:= , bExecute:= , bVirtual:= , bDone=> , bBusy=> , bError=> , nErrorID=>);

(2) Related variables

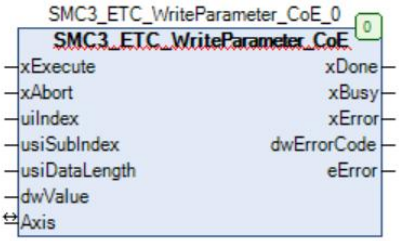
VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
bExecute	Valid	BOOL	TRUE/FALSE	FALSE	The rising edge of the input value will initiate the execution of the function block
bVirtual	Language	BOOL	TRUE/FALSE	FALSE	Input determines whether an axis actually exists or is simulated
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
bDone	Completed	BOOL	TRUE/FALSE	FALSE	TRUE after execution is completed
bBusy	Executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
bError	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
nErrorID	Error code	SMC_ERROR	-	0	Error recognition

(3) Function description

- Used to reset the drive and reconnect with the controller.
- Input bvirtual to determine whether an axis truly exists or is simulated. If the input bVirtual is set to TRUE, the axis is set to virtual mode. Then, it will be replaced by simulations similar to virtual drive devices. This has no impact on the fieldbus device, it will continue to operate as usual, but will not receive messages from or send messages to the actual device.

3-1-2-28. Set the axis COE parameter [SMC3_ETC_WriteParameter_CoE]

(1) Instruction overview

Instruction	Name	Graphic representation	ST language
SMC3_ETC_WriteParameter_CoE	Set the axis COE parameter		SMC3_ETC_WriteParameter_CoE(xExecute:= , xAbort:= , uiIndex:= , usiSubIndex:=, usiDataLength:= , dwValue:= , Axis:=SM_Drive_GenericDSP402 , xDone=> , xBusy=> , xError=> , dwErrorCode=> , eError=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
xExecute	Valid	BOOL	TRUE/FALSE	FALSE	The rising edge of the input value will initiate the execution of the function block
XAbort	Abort	BOOL	TRUE/FALSE	FALSE	Terminate an ongoing write request
uiIndex	Object index	UINT	-	-	Object index, eg. 16#6060
usiSubInde	Object subindex	USINT	-	-	Object subindex, eg. 0
usiDataLength	Write in data length	USINT	-	-	The length of data written in bytes (1-4)
dwValue	Write value	DWORD	-	-	Write in value DWORD
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
bDone	Completed	BOOL	TRUE/FALSE	FALSE	TRUE after execution is completed
bBusy	Executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
bError	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
nErrorID	Error code	SMC_ERROR	-	0	Error recognition

(3) Application

Modify the value of 607D:

```
VAR
  SMC3_ETC_WriteParameter_CoE_0:  SMC3_ETC_WriteParameter_CoE;
```

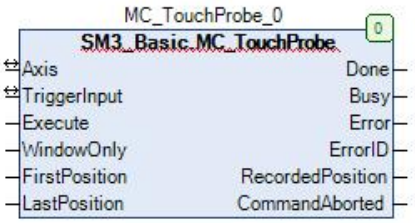
```
    SMC3_ETC_WriteParameter_CoE_1:    SMC3_ETC_WriteParameter_CoE;
    WRITE: BOOL;
    SMC3_ReinitDrive: SMC3_ReinitDrive;
END_VAR
```

```
SMC3_ETC_WriteParameter_CoE_0(
    xExecute: = WRITE, //trigger at the rising edge
    xAbort: = ,
    uiIndex: =16#607D , //object index such as 16#6060
    usiSubIndex: =1 , //object subindex such as 0
    usiDataLength: =4 , // the length of data written in bytes (1-4)
    dwValue: = 16#8000, //write in value DWORD, can customize DWORD variables
    Axis: =SM_Drive_GenericDSP402 , // SoftMotion axis
    xDone=> ,
    xBusy=> ,
    xError=> ,
    dwErrorCode=> ,
    eError=> );
SMC3_ETC_WriteParameter_CoE_1(
    xExecute: = WRITE,
    xAbort: = ,
    uiIndex: =16#607D ,
    usiSubIndex: =2 ,
    usiDataLength: =4 ,
    dwValue: = 16#8000,
    Axis: =SM_Drive_GenericDSP402 ,
    xDone=> ,
    xBusy=> ,
    xError=> ,
    dwErrorCode=> ,
    eError=> );
```

3-1-2-29. Probe [MC_TouchProbe]

(1) Instruction overview

This function block is used to record the position of the axis when a triggering event occurs.

Instruction	Name	Graphic representation	ST language
MC_TouchProbe	Enable probe		MC_TouchProbe(Axis:= , TriggerInput:= , Execute:= , WindowOnly:= , FirstPosition:= , LastPosition:= , Done=> , Busy=> , Error=> , ErrorID=> , RecordedPosition=> , CommandAborted=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis
TruggerInput	Trigger signal	TRIIGGER_ REF	-	-	Reference trigger signal source
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Execute	Valid	BOOL	TRUE/FALSE	FALSE	The rising edge of the input value will initiate the execution of the function block
WindowOnly	Trigger window	BOOL	TRUE/FALSE	FALSE	
FirstPosition	Parameter value	LREAL	-	0	The starting position for receiving triggering events
LastPosition	Parameter value	LREAL	-	0	The final location to receive the triggering event
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	Completed	BOOL	TRUE/FALSE	FALSE	True if the parameter value has been successfully written
Busy	Executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	Error code	SMC_ERROR	-	0	Error recognition
RecordedPosition	Trigger record location	LREAL	-	0	Current position when triggered
CommandAborted	Instruction interrupted	BOOL	TRUE/FALSE	FALSE	The current instruction has been interrupted and set to TRUE

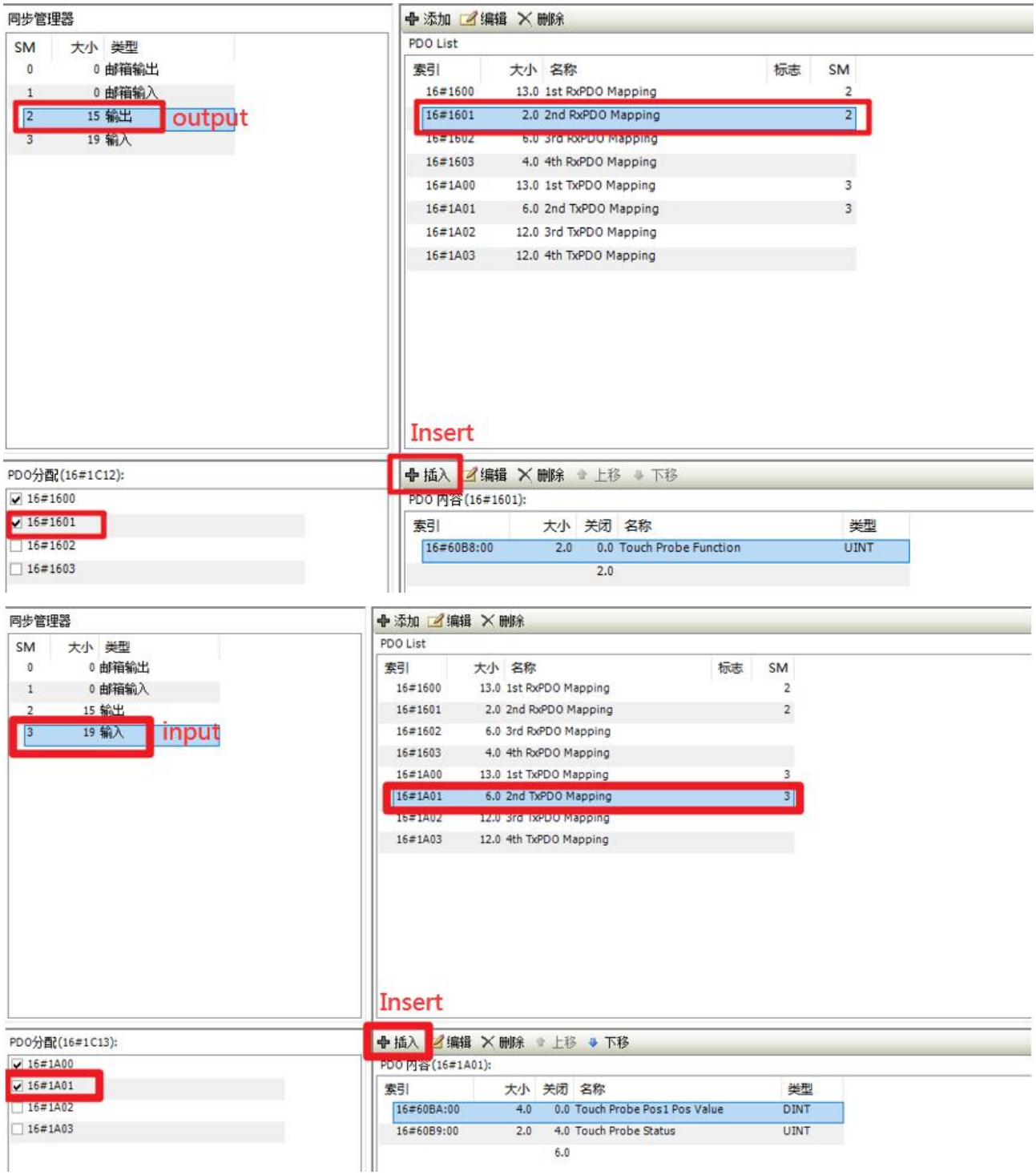
(3) Function description

- Record the current position of the running axis when the TruggerInput signal is triggered through the function block MC_TouchProbe.
- When the driver is locked: the driver will collect the locking signal at the recording position and wait until the controller.

(4) Application

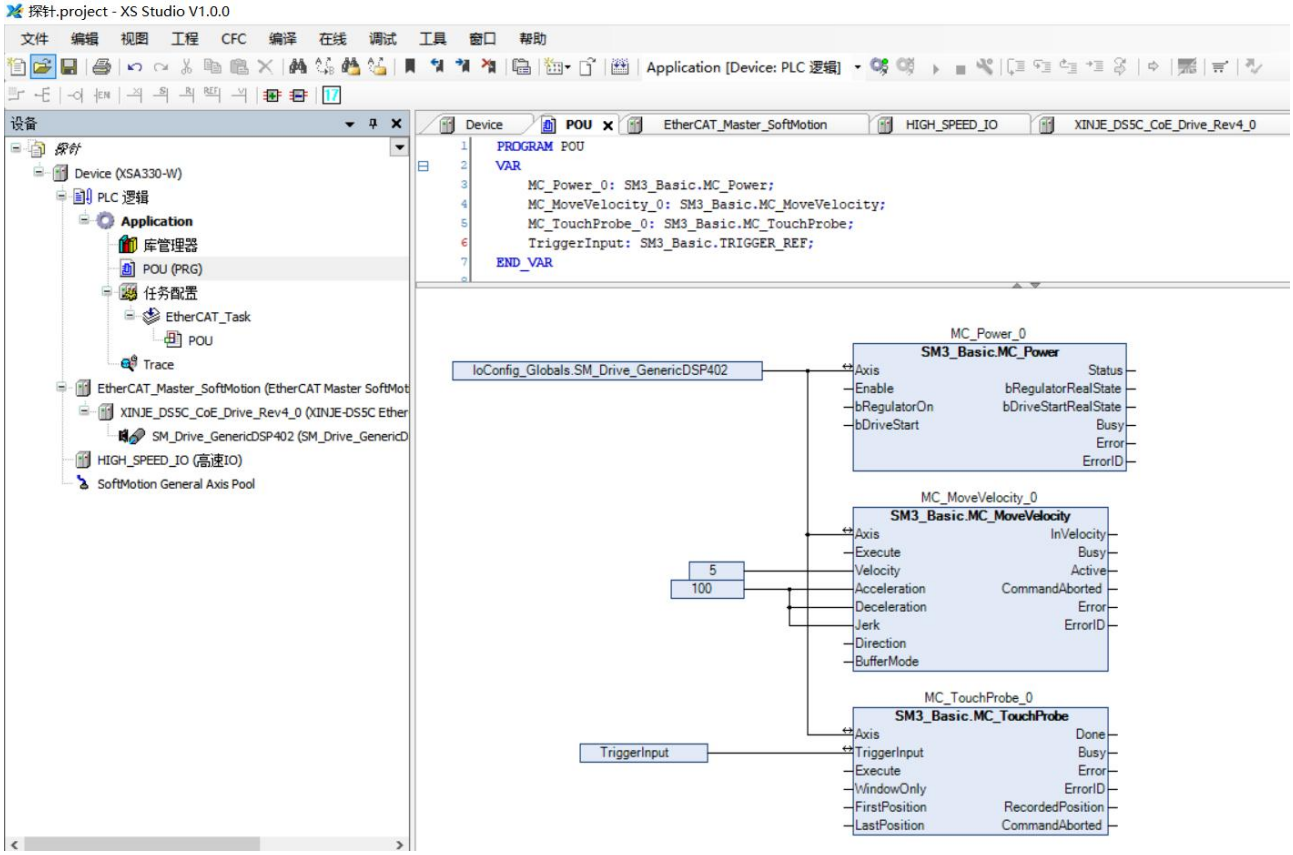
Example 1: Taking Xinje DS5C servo as an example, to achieve locking position function of probe 1 at the rising edge.

Programming:
(1) Check 'Enable Expert Settings', select 1601 and 1A01 in PDO Allocation in Expert Process Data, add 60B8h in 1601, and 60BAh and 60B9h in 1a01.

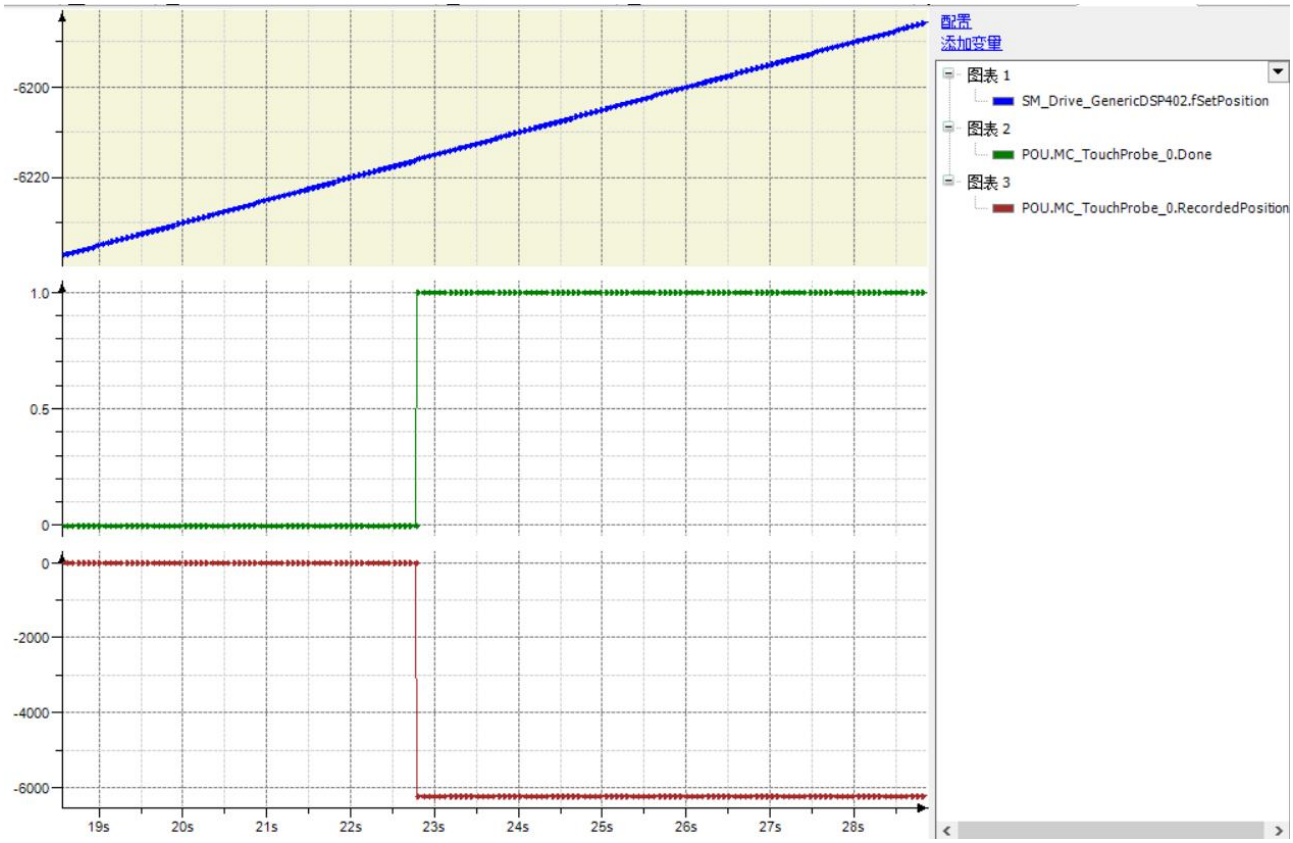


(2) You can see the newly added parameters in "EthercatI/O Mapping".

(3) Set bFastLatching to true, bInput to false, and iTriggerNumber to 0 in the input pin TriggerInput of the function block "MC_TouchProbe". After the function block is turned on, it will open the rising edge of probe 1. After conduction, it can be seen that the value in 60B8h is 17, which is the rising edge of probe 1.



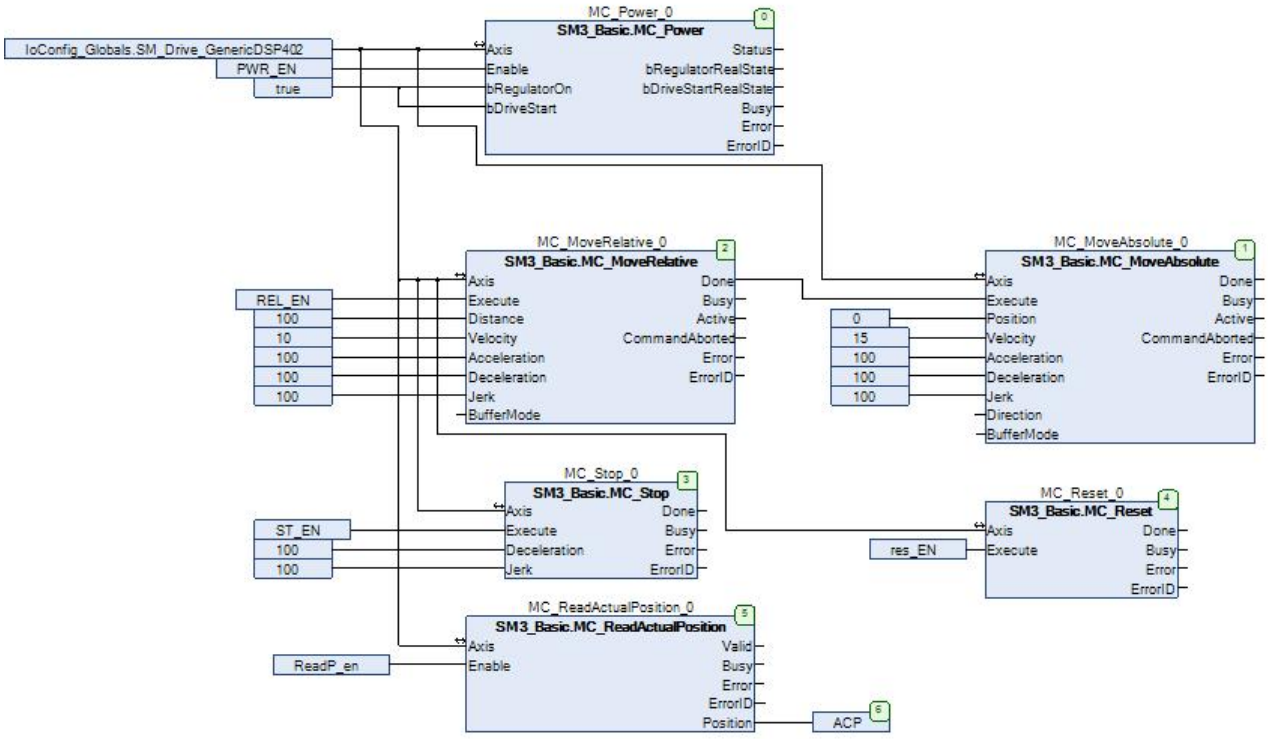
(4) Can be found in 'Trace' that after MC_TouchProbe command done, it will immediately latch the current position.


配置
添加变量
图表 1
SM_Drive_GenericDSP402.fSetPosition
图表 2
POU.MC_TouchProbe_0.Done
图表 3
POU.MC_TouchProbe_0.RecordedPosition
-6200
-6220
1.0
0.5
0
0
-2000
-4000
-6000
19s
20s
21s
22s
23s
24s
25s
26s
27s
28s

3-1-3. Single axis function application

Example 1: To achieve axis 0 movement in two segments to reach the designated position of 0 Pluse, the first segment operates at a speed of 10 Pluse/S and an acceleration of 100 Pluse/S^2 for relative motion. Then, relative to the starting position, a further 100 Pluse is run. After reaching the relative position of the target, an absolute motion is run at a speed of 15 Pluse/S and an acceleration of 100 Pluse/S^2 for absolute motion. The motion reaches the target position of 0 Pluse. During the movement process, the real-time position can be read, and the movement can also be stopped. If an error occurs, the axis can also be reset.

```
PROGRAM POU
VAR
    //功能块定义  define the function block
    MC_Power_0: SM3_Basic.MC_Power;
    MC_MoveRelative_0: SM3_Basic.MC_MoveRelative;
    MC_MoveAbsolute_0: SM3_Basic.MC_MoveAbsolute;
    MC_Stop_0: SM3_Basic.MC_Stop;
    MC_Reset_0: SM3_Basic.MC_Reset;
    MC_ReadActualPosition_0: SM3_Basic.MC_ReadActualPosition;
    PWR_EN: BOOL;//使能  enable
    ST_EN: BOOL;//停止  stop
    res_EN: BOOL;//复位  reset
    ReadP_en: BOOL;//实时位置读取  read real-time position
    ACP: LREAL;//读取的位置  read position
END_VAR
```



3-2. Axis group function

3-2-1. Axis group instruction

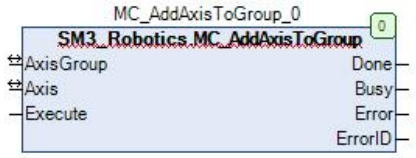
Instruction	Function
MC_AddAxisToGroup	add the axis to the axis group
MC_RemoveAxisFromGroup	remove the axis from the axis group
MC_UngroupAllAxes	delete all the axes in the axis group
MC_GroupEnable	enable the axis group
MC_GroupDisable	disable the axis group
MC_GroupReset	reset the axis group
MC_GroupSetPosition	set the axis group position
MC_SetCoordinateTransform	coordinate transformation
MC_SetDynCoordTransform	connect two axis groups
MC_GroupContinue	axis group continue
MC_GroupHalt	axis group halt
MC_GroupInterrupt	axis group interrupt
MC_GroupStop	axis group stop
MC_GroupSetOverride	change speed, acceleration, or active and controlled actions
MC_SetKinTransform	kinematic coordinate system conversion
MC_MoveCircularAbsolute	circular move to absolute position
MC_MoveCircularRelative	circular move to relative position
MC_MoveDirectAbsolute	move to absolute position
MC_MoveDirectRelative	move to relative position
MC_MoveLinearAbsolute	linear move to absolute position
MC_MoveLinearRelative	linear move to relative position
MC_GroupReadActualPosition	read actual position
MC_GroupReadActualVelocity	read actual speed
MC_GroupReadConfiguration	read parameters
MC_GroupReadError	read error
MC_GroupReadStatus	read status
SMC_StartupAxisGroup	startup the axis group
SMC_GroupPower	power on the axis group
SMC_GroupInterruptAt	break assignment
SMC_GroupEnableResumeAfterError	Resume motion after axis error
SMC_GroupJog	axis group jog run
SMC_GroupWait	axis group wait

3-2-2. Axis group instructions

3-2-2-1. Add axis to axis group [MC_AddAxisToGroup]

(1) Instruction overview

Add an axis to the axis group using command mode.

Instruction	Name	Graphical representation	ST language
MC_AddAxisToGroup	Add axis to axis group		MC_AddAxisToGroup(AxisGroup:= , Axis:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
Axis	axis	AXIS_REF_SM3	-	-	specified axis
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Execute	valid	BOOL	TRUE/FALSE	FALSE	The rising edge of the input value will initiate the execution of the function block
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	completed	BOOL	TRUE/FALSE	FALSE	TRUE when instruction execution is completed
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Error recognition

(3) Function description

- The instruction is used to add axis to the specified axis group and bind them together in the relationship of the axis group.
- When the Done variable of the instruction becomes TRUE, it indicates that the axis has been successfully added to the axis group. Note that setting Execute to FALSE does not remove the axis from the axis group. If you need to remove the axis from the axis group, you need to use MC_ RemoveAxisFromGroup instruction.
- This command can only be executed when the axis group is in the GroupDisabled state. If this command is executed after the axis group is enabled, an error will be reported.

3-2-2-2. Remove axis from the axis group [MC_RemoveAxisFromGroup]

(1) Instruction overview

Remove the axis from the axis group.

Instruction	Name	Graphical representation	ST language
MC_RemoveAxisFromGroup	remove axis from the axis group		MC_RemoveAxisFromGroup(AxisGroup:= , Axis:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
Axis	axis	AXIS_REF_SM3	-	-	specified axis
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Execute	valid	BOOL	TRUE/FALSE	FALSE	The rising edge of the input value will initiate the execution of the function block
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	completed	BOOL	TRUE/FALSE	FALSE	TRUE when instruction execution is completed
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Error recognition

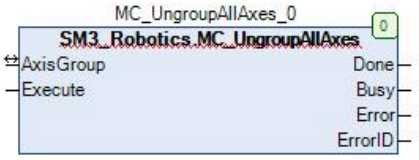
(3) Function description

- This command can only be executed when the axis group is not enabled. If this command is executed after the axis group is enabled, an error will be reported.
- When the Done variable of the instruction becomes TRUE, it indicates that the axis has been successfully removed from the axis group.

3-2-2-3. Ungroup all the axes [MC_UngroupAllAxes]

(1) Instruction overview

Remove all the axes contained in a certain axis group and dissolve the axis group.

Instruction	Name	Graphical representation	ST language
MC_UngroupAllAxes	ungroup the axis group		MC_UngroupAllAxes(AxisGroup:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Execute	Valid	BOOL	TRUE/FALSE	FALSE	The rising edge of the input value will initiate the execution of the function block
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	completed	BOOL	TRUE/FALSE	FALSE	TRUE when instruction execution is completed
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Error recognition

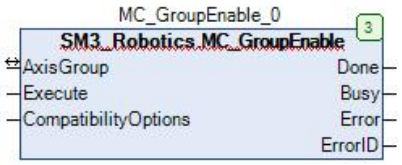
(3) Function description

- Remove all axes from the axis group. If the status of the axis group is not GroupDisabled, an error will be generated and the axis will not be removed from the axis group.

3-2-2-4. Enable the axis group [MC_GroupEnable]

(1) Instruction overview

Enable the axis group.

Instruction	Name	Graphical representation	ST language
MC_GroupEnable	Enable the axis group		MC_GroupEnable(AxisGroup:= , Execute:= , CompatibilityOptions:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Execute	valid	BOOL	TRUE/FALSE	FALSE	The rising edge of the input value will initiate the execution of the function block
CompatibilityOptions	compatibility option	SMC_AXIS_GROUP_COMPATIBILITY_OPTIONS	-	-	Parameters that exist for compatibility with previous versions
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	completed	BOOL	TRUE/FALSE	FALSE	TRUE when instruction execution is completed
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Error recognition

(3) Function description

- Before calling the motion control and other commands of the axis group, SMC_GroupPower or MC_Power needs to be called to enable the axis group.
- When the Done variable of the instruction becomes TRUE, it indicates that the axis group has successfully switched to the Standby state.
- The types of axes that can be specified in the axis group can only be "servo axis" and "virtual servo axis". If other axis types are specified, an exception will occur.
- When executing this command, all axes under the axis group must be in a stop state.

- If there are axes that already belong to other axis groups and have already been enabled, MC_GroupEnable cannot be executed, the command will cause an unexpected error.
- The conditions for invalidating the axis group include: executing MC_GroupDisable command, switch to program mode to stop running, and start MC trial run.

3-2-2-5. Disable the axis group [MC_GroupDisable]

(1) Instruction overview

The axis group is switched to the Disabled state, and motion control of the axis group cannot be performed in this state.

Instruction	Name	Graphical representation	ST language
MC_GroupEnable	Disable the axis group		MC_GroupDisable(AxisGroup:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Execute	valid	BOOL	TRUE/FALSE	FALSE	The rising edge of the input value will initiate the execution of the function block
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	completed	BOOL	TRUE/FALSE	FALSE	TRUE when instruction execution is completed
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Error recognition

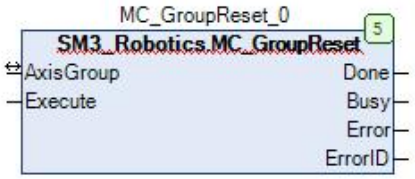
(3) Function description

- This command switches the specified axis group to the GroupDisable state.
- When the axis group status changes to GroupDisable, the cache instructions for the specified AxesGroup will be cleared.

3-2-2-6. Axis group reset [MC_GroupReset]

(1) Instruction overview

Release the abnormal state of the axis group and axis.

Instruction	Name	Graphical representation	ST language
MC_GroupReset	axis group reset		MC_GroupReset(AxisGroup:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Execute	valid	BOOL	TRUE/FALSE	FALSE	The rising edge of the input value will initiate the execution of the function block
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	completed	BOOL	TRUE/FALSE	FALSE	TRUE when instruction execution is completed
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Error recognition

(3) Function description

- On the rising edge of Execute, handle the exceptions of the AxesGroup specified axis group in the GroupEnable state and the exceptions of the axis to which the axis group belongs. The exceptions that can be resolved include "minor faults" in the axis and axis group, "abnormal monitoring information", and driver error reset.
- Regardless of whether the axis is in servo ON or servo OFF state, abnormal release processing can be executed.
- For axes with driver errors, the driver error reset process should be executed first, and then the exception relief process should be executed.
- The driver error reset processing can choose to clear the driver error or remain unchanged within the axis parameter [driver error reset monitoring time]. Driver error reset is performed simultaneously on all axes belonging to the axis group.
- The exception object that can be resolved is the exception that occurs when the rising edge of Execute is activated. Cannot perform exception resolution on exceptions that occur during the exception resolution process.

- If the command is executed during an incorrect deceleration stop of the axis group, it cannot be executed because abnormal release cannot be performed before the axis stops. In addition, if an abnormal error occurs on the axis itself in the axis group, it cannot be resolved through this command.

3-2-2-7. Set axis group position [MC_GroupSetPosition]

(1) Instruction overview

Used to set the command positions of each axis in the axis group.

Instruction	Name	Graphical representation	ST language
MC_GroupSetPosition	set axis group position		MC_GroupSetPosition(AxisGroup:= , Execute:= , Position:= , Relative:= , CoordSystem:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
VAR_INPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Execute	valid	BOOL	TRUE/FALSE	FALSE	The rising edge of the input value will initiate the execution of the function block
Position	position	SMC_POS_REF	data range	0	axis target position
Relative	position mode	BOOL	TRUE/FALSE	FALSE	Relative position mode=True, absolute position mode=False (default)
CoordSystem	Applied coordinate system	SMC_COORD_SYSTEM	-	-	Applied coordinate system
VAR_OUTPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Done	completed	BOOL	TRUE/FALSE	FALSE	TRUE when instruction execution is completed
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Error recognition

(3) Function description

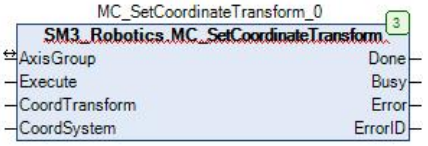
- Set the coordinate position under the specified coordinate system in the axis group.

- The instruction is executed in the GroupStandby state of the axis group, and cannot be executed in the dynamic coordinate system or executed with MC_GroupContinue instruction simultaneously.

3-2-2-8. Coordinate transform [MC_SetCoordinateTransform]

(1) Instruction overview

Used to convert different command coordinates of reference coordinate systems.

Instruction	Name	Graphical representation	ST language
MC_SetCoordinateTransform	Coordinate transform		MC_SetCoordinateTransform(AxisGroup:= , Execute:= , CoordTransform:= , CoordSystem:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Execute	valid	BOOL	TRUE/FALSE	FALSE	The rising edge of the input value will initiate the execution of the function block
CoordTransform	coordinate transform	MC_COORD_REF	-	-	Coordinate transformation, i.e. the product coordinate system (PCS1 or PCS2) or machine coordinate system (MCS) represented by the World Coordinate System (WCS)
CoordSystem	Applied coordinate system	SMC_COORD_SYSTEM	-	-	Target coordinate system, allowing for conversion of PCS_1, PCS_2 and MCS
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	completed	BOOL	TRUE/FALSE	FALSE	TRUE when instruction execution is completed
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Error recognition

(3) Function description

- Set the world coordinate system (WCS) and product/machine coordinate system (PCS */MCS) in the axis group for coordinate system conversion.
- When PCS is a dynamic coordinate system (PCS moves relative to WCS), MC_SetDynCoordTransform needs to be used.
- This command only performs coordinate system conversion and is independent of motion control commands.

3-2-2-9. Dynamic coordinate system conversion [MC_SetDynCoordTransform]

(1) Instruction overview

When the specified coordinate system moves relative to WCS, this command needs to be called to achieve coordinate system conversion.

Instruction	Name	Graphical representation	ST language
MC_SetDynCoordTransform	Dynamic coordinate system conversion		MC_SetDynCoordTransform(AxisGroup:= , MasterAxisGroup:= , Execute:= , CoordTransform:= , CoordSystem:= , Done=> , Busy=> , InUse=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
MasterAxisGroup	master axis group	AXIS_GROUP_REF_SM3	-	-	specified master axis group
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Execute	valid	BOOL	TRUE/FALSE	FALSE	The rising edge of the input value will initiate the execution of the function block
CoordTransform	Coordinate system to be converted	MC_COORD_REF	-	-	The tool coordinate system of the spindle group is relative to the coordinates and direction of PCS
CoordSystem	Coordinate System	SMC_COORD_SYSTEM	-	-	The PCS coordinate system to be converted (PCS1 or PCS2)
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	completed	BOOL	TRUE/FALSE	FALSE	TRUE when instruction execution is completed
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the

					function block has not yet ended
InUse	Reference system	BOOL	-	-	Indicates the dynamic coordinate system that the axis group still needs to reference
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Error recognition

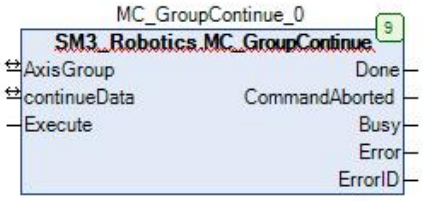
(3) Function description

- Usually used when using dynamic PCS, such as MC_TrackConveyorBelt or MC_TrackRotaryTable is called together.
- Instruction SMC_SetDynCoordTransformEX provides a more universal interface.

3-2-2-10. Axis group continue running [MC_GroupContinue]

(1) Instruction overview

Release the interrupt status of the axis group and continue executing unfinished commands.

Instruction	Name	Graphical representation	ST language
MC_GroupContinue	axis group continue running		MC_GroupContinue(AxisGroup:= , continueData:= , Execute:= , Done=> , CommandAborted=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
continueData	Continuing motion data	SMC_AXIS_GROUP_CONTINUE_DATA	-	-	Axis group position during motion interruption
VAR_INPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Execute	valid	BOOL	TRUE/FALSE	FALSE	Execute current instruction
VAR_OUTPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Done	completed	BOOL	TRUE/FALSE	FALSE	TRUE when instruction execution is completed
CommandAborted	Instruction interrupted	BOOL	TRUE/FALSE	FALSE	Module execution interrupted is true
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution

					error
ErrorID	error code	SMC_ERROR	-	0	Error recognition

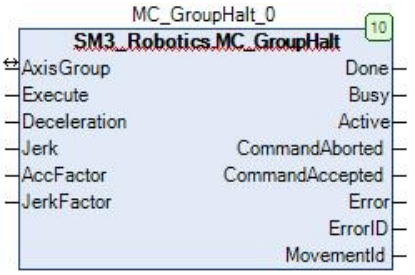
(3) Function description

- Release the interrupt status of the axis group and continue executing unfinished commands.

3-2-2-11. Axis group halt [MC_GroupHalt]

(1) Instruction overview

Used to pause the current axis group motion.

Instruction	Name	Graphical representation	ST language
MC_GroupHalt	axis group halt		MC_GroupHalt(AxisGroup:= , Execute:= , Deceleration:= , Jerk:= , AccFactor:= , JerkFactor:= , Done=> , Busy=> , Active=> , CommandAborted=> , CommandAccepted=> , Error=> , ErrorID=> , MovementId=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Execute	valid	BOOL	TRUE/FALSE	FALSE	Execute current instruction
Deceleration	deceleration	LREAL	positive number	0	Maximum resultant deceleration [command unit/s^2]
Jerk	jerk speed	LREAL	positive number	0	Maximum resultant jerk speed[command unit/s^3]
AccFactor	acceleration factor	LREAL	0-1	1	Acceleration factor, the maximum velocity of each axis multiplied by this acceleration factor, with a value between [0,1]
JerkFactor	jerk factor	LREAL	0-1	1	Jerk speed factor, multiply the maximum velocity of each axis by this jerk factor, and the value is between [0,1]
VAR_OUTPUT	Name	Data type	Effective	Initial	Description

			range	value	
Done	completed	BOOL	TRUE/FALSE	FALSE	TRUE when instruction execution is completed
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Active	in control	BOOL	TRUE/FALSE	FALSE	Change to TRUE in control
CommandAborted	Instruction interrupted	BOOL	TRUE/FALSE	FALSE	Module execution interrupted is true
CommandAccepted	motion accepted	BOOL	TRUE/FALSE	FALSE	True when the module successfully calls the axis group
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Error recognition
MovementId	motion flag	SMC_Movement_Id	TRUE/FALSE	FALSE	True when the motion is being executed or completed

3-2-2-12. Axis group interruption [MC_GroupInterrupt]

(1) Instruction overview

Interrupt the currently moving axis group, which can be done through MC_GroupContinue instruction continues to execute unfinished motion instructions.

Instruction	Name	Graphical representation	ST language
MC_GroupInterrupt	axis group interruption		MC_GroupInterrupt(AxisGroup:= , continueData:= , Execute:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=> , mvtIdInterruptPosition=>);

(2) Related variables

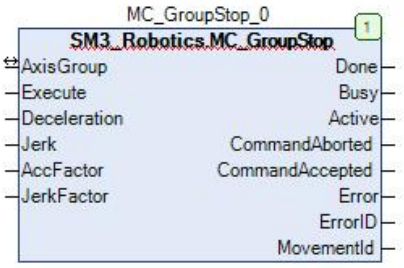
VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
continueData	Continuing motion data	SMC_AXIS_GROUP_CONTINUE_DATA	-	-	Motion information when axis group motion is interrupted
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Execute	valid	BOOL	TRUE/FALSE	FALSE	Execute current instruction
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	completed	BOOL	TRUE/FALSE	FALSE	TRUE when instruction execution is completed
CommandAborted	Instruction	BOOL	TRUE/FALSE	FALSE	Module execution interrupted

	interrupted				is true
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Error recognition
mvtIdInterruptPosition	interrupt position ID	SMC_Movement_Id	-	-	The Movement ID corresponding to the interrupt location

3-2-2-13. Axis group stop [MC_GroupStop]

(1) Instruction overview

Stop the movement of the axis group.

Instruction	Name	Graphical representation	ST language
MC_GroupStop	axis group stop		MC_GroupStop(AxisGroup:= , Execute:= , Deceleration:= , Jerk:= , AccFactor:= , JerkFactor:= , Done=> , Busy=> , Active=> , CommandAborted=> , CommandAccepted=> , Error=> , ErrorID=> , MovementId=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Execute	valid	BOOL	TRUE/FALSE	FALSE	Execute current instruction
Deceleration	deceleration	LREAL	positive number	0	Maximum resultant deceleration [command unit/s^2]
Jerk	jerk speed	LREAL	positive number	0	Maximum resultant jerk speed [command unit/s^3]
AccFactor	acceleration factor	LREAL	0-1	1	Acceleration factor, the maximum velocity of each axis multiplied by this acceleration factor, with a value between [0,1]

JerkFactor	jerk factor	LREAL	0-1	1	Jerk speed factor, multiply the maximum velocity of each axis by this jerk factor, and the value is between [0,1]
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	completed	BOOL	TRUE/FALSE	FALSE	TRUE when instruction execution is completed
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Active	in control	BOOL	TRUE/FALSE	FALSE	Change to TRUE in control
CommandAborted	Instruction interrupted	BOOL	TRUE/FALSE	FALSE	Module execution interrupted is true
CommandAccepted	motion accepted	BOOL	TRUE/FALSE	FALSE	True when the module successfully calls the axis group
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Error recognition
MovementId	motion flag	SMC_Movement_Id	TRUE/FALSE	FALSE	True when the motion is being executed or completed

3-2-2-14. Kinematic coordinate transformation [MC_SetKinTransform]

(1) Instruction overview

Set the kinematic transformation of the axis group from the ACS coordinate system to the MCS coordinate system.

Instruction	Name	Graphical representation	ST language
MC_SetKinTransform	Kinematic coordinate transformation		MC_SetKinTransform(AxisGroup:= , Execute:= , KinTransform:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Execute	valid	BOOL	TRUE/FALSE	FALSE	The rising edge of the input value will initiate the execution of the function block
KinTransform	Kinematic	TRAFO.MC_KIN_REF_	-	-	Kinematic transformation

	transformation values	SM3			
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	completed	BOOL	TRUE/FALSE	FALSE	TRUE when instruction execution is completed
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Error recognition

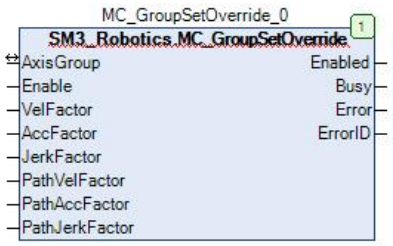
(3) Function description

- Set up kinematic transformations between the Axis Coordinate System (ACS) and the Machine Coordinate System (MCS).
- After executing this function block, the tool offset will be reset.

3-2-2-15. Set axis group overshoot value [MC_GroupSetOverride]

(1) Instruction overview

When the axis group is in the Moving state, change the motion speed of the axis group.

Instruction	Name	Graphical representation	ST language
MC_GroupSetOverride	set the axis group overshoot value		MC_GroupSetOverride(AxisGroup:= , Enable:= , VelFactor:= , AccFactor:= , JerkFactor:= , PathVelFactor:= , PathAccFactor:= , PathJerkFactor:= , Enabled=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Execute	valid	BOOL	TRUE/FALSE	FALSE	Execute current instruction
VelFactor	speed factor	LREAL	0-1	1	Speed factor, the maximum speed of each axis multiplied by this speed factor, with a value between [0, 1]

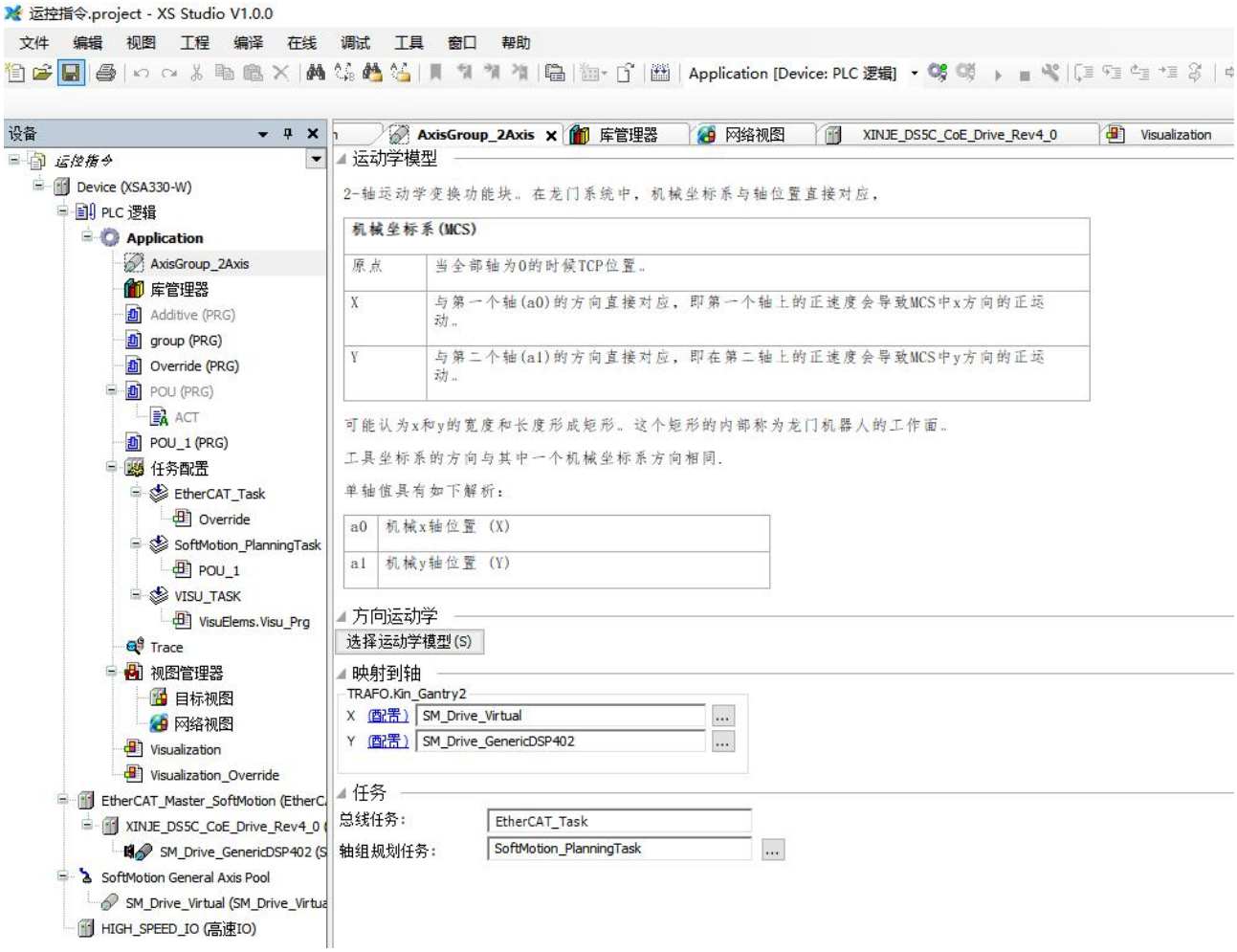
AccFactor	acceleration factor	LREAL	0-1	1	Acceleration factor, the maximum velocity of each axis multiplied by this acceleration factor, with a value between [0,1]
JerkFactor	jerk speed factor	LREAL	0-1	1	jerk velocity factor, multiply the maximum velocity of each axis by this jerk factor, and the value is between [0,1]
PathVelFactor	Resultant velocity factor	LREAL	0-1	1	Resultant velocity factor, the maximum resultant velocity of the entire axis group's motion trajectory multiplied by this velocity factor, with a value between [0,1]
PathAccFactor	resultant acceleration factor	LREAL	0-1	1	Resultant acceleration factor, the maximum resultant acceleration of the entire axis group motion trajectory multiplied by this resultant acceleration factor, with a value between [0,1]
PathJerkFactor	resultant jerk speed factor	LREAL	0-1	1	The resultant jerk factor is the maximum resultant jerk speed of the entire axis group's motion trajectory multiplied by this resultant jerk factor, with a value between [0,1]
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Enabled	completed	BOOL	TRUE/FALSE	FALSE	When the overshoot factor setting completed is TRUE
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Error recognition

(3) Function description

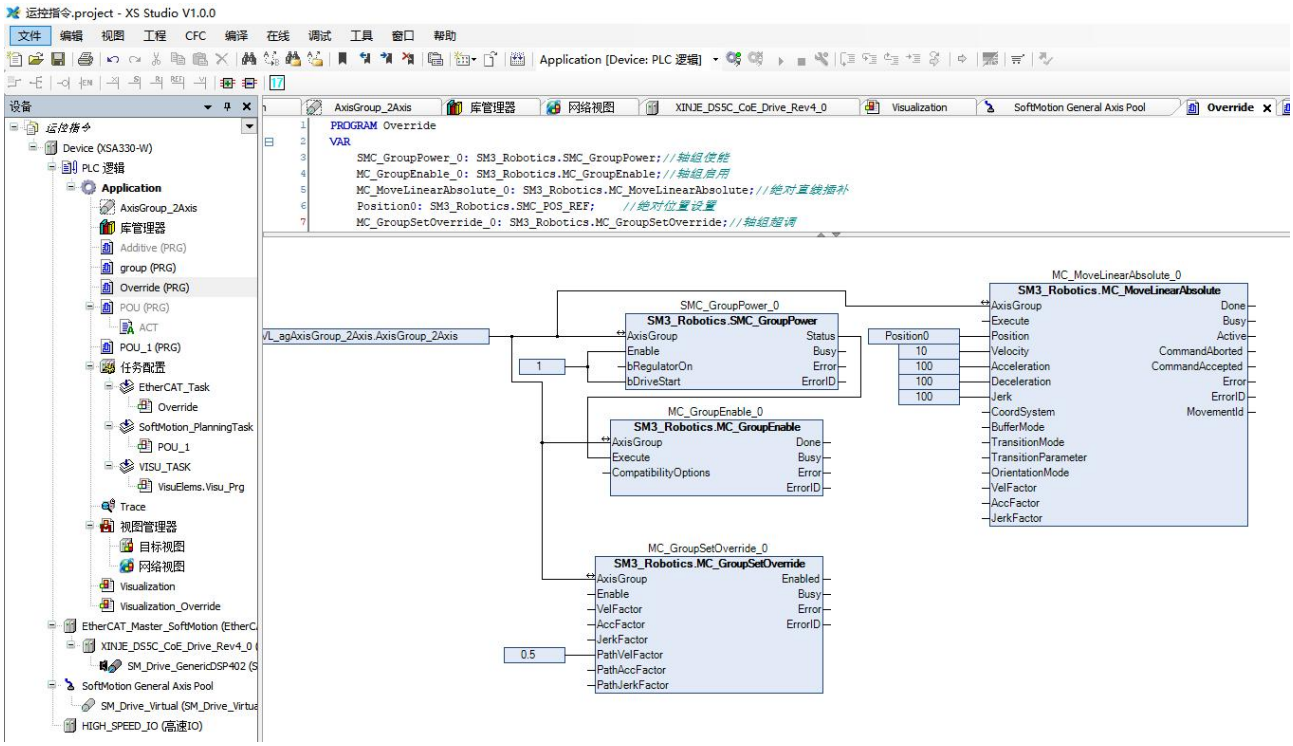
- The axis group can change the speed, acceleration, and acceleration of movement by executing this command in the Moving state.
- The given overshoot factor remains valid until a new value is set. For example, for an axis group in CP motion (even following motion), setting VelFactor or PathVelFactor to 0 will cause a sudden stop in the motion trajectory. If MC_GroupStop is currently in an Active state, an error will be returned.
- MC_GroupStop instruction is not affected by the overshoot factor.
- Re-enable the axis group, and the overshoot factor remains valid.
- The default value of the overshoot factor is 1.
- Reducing AccFactor or JerkFactor can cause position overshoot and may cause damage to the equipment.

(4) Application

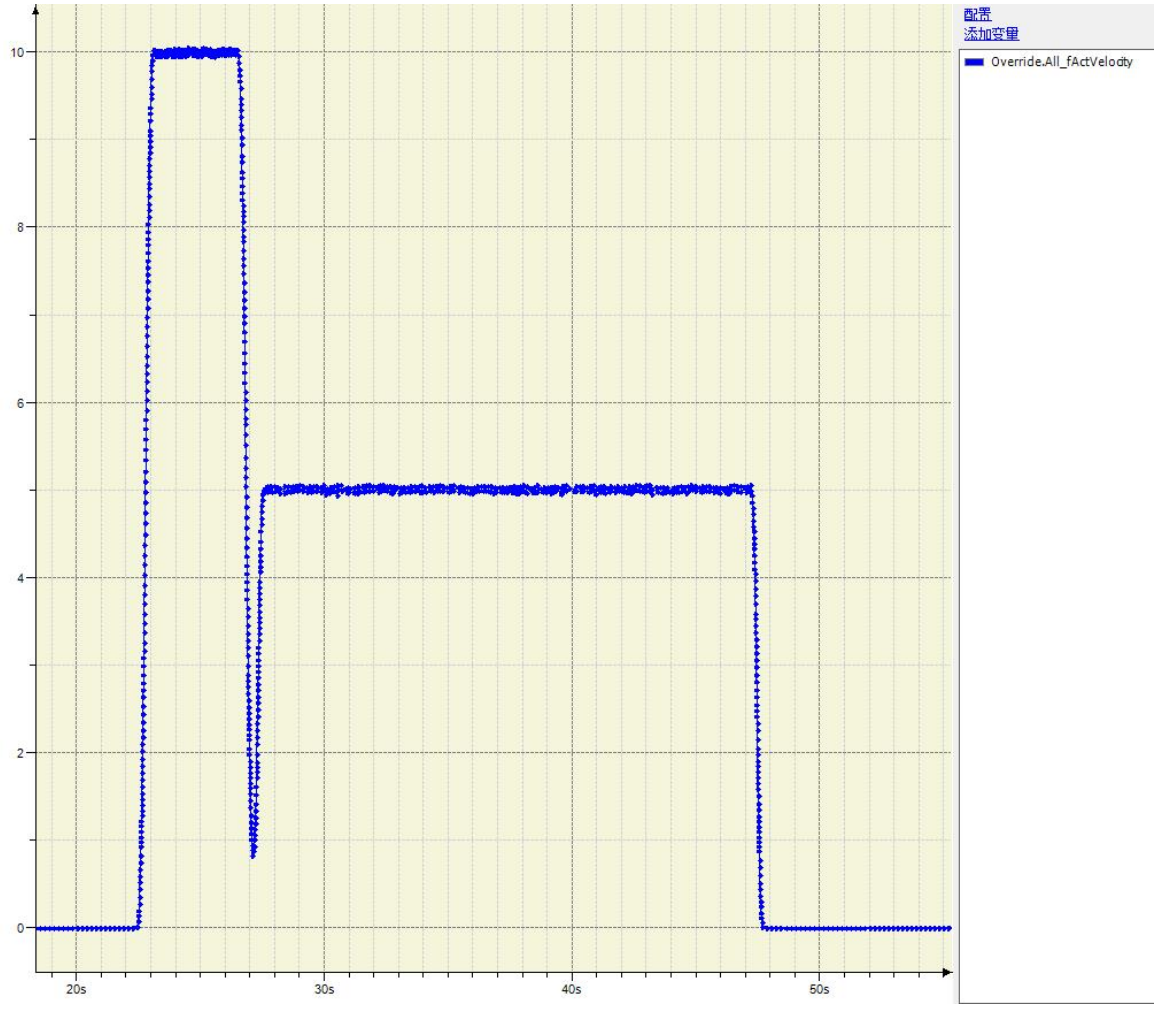
Example 1: Select the kinematic model of the two axes of the gantry, select the virtual axis for the X-axis, and select the Ethercat real axis for the Y-axis. During the execution of the "MC-MoveLinearAbsolute" absolute linear interpolation process, use the "MC_GroupSetOverride" axis group overshoot command to change the axis group speed.



Programming: Set the current positions of both axes to 0, set the target position to (100, 100), perform absolute linear interpolation motion with a starting point of (0,0), and set the axis group speed to 10u/s. Use the axis group overshoot command to change the axis group speed to 5u/s.



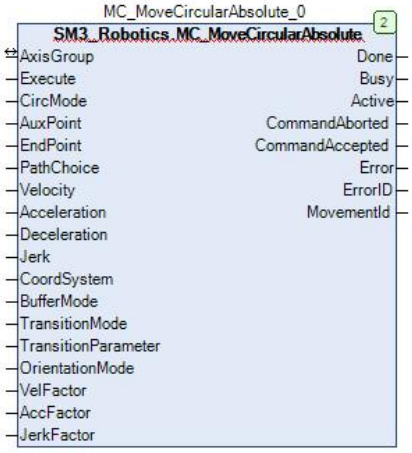
In Trace, the current speed of the axis group can be seen. After conducting the axis group overshoot command, the speed changes from 10u/s to 5u/s.



3-2-2-16. Absolute arc interpolation [MC_MoveCircularAbsolute]

(1) Instruction overview

Control the axis group to perform arc interpolation motion in absolute position mode.

Instruction	Name	Graphical representation	ST language
MC_MoveCircularAbsolute	absolute arc interpolation		MC_MoveCircularAbsolute(AxisGroup:= , Execute:= , CircMode:= , AuxPoint:= , EndPoint:= , PathChoice:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , CoordSystem:= , BufferMode:= , TransitionMode:= , TransitionParameter:= , OrientationMode:= , VelFactor:= , AccFactor:= , JerkFactor:= , Done=> , Busy=> , Active=> , CommandAborted=> , CommandAccepted=> , Error=> , ErrorID=> , MovementId=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
VAR_INPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Execute	valid	BOOL	TRUE/FALSE	FALSE	Execute current instruction
CircMode	arc mode	SMC_CIRC_MODE	Border/Center/Radius	-	Specify the method of arc interpolation: Border: Three point arc Center: Center arc Radius: Radius arc
AuxPoint	Auxiliary point	SMC_POS_REF	-	-	Specify auxiliary points in the coordinate system. Refer to CircMode
EndPoint	end point	SMC_POS_REF	-	-	Specify the end position in the coordinate system
PathChoice	direction	MC_CIRC_PATHCHOICE	CLOCKWISE/COUNTER_CLOCKWISE	-	motion direction: CLOCKWISE: clockwise COUNTER_CLOCKWISE: counterclockwise

Velocity	speed	LREAL	0, positive number	0	Max resultant velocity [command unit/s]
Deceleration	deceleration	LREAL	positive number	0	Max resultant deceleration[command unit/s^2]
Jerk	jerk speed	LREAL	positive number	0	Max resultant jerk speed [command unit/s^3]

VAR_INPUT	Name	Data type	Effective range	Initial value	Description
CoordSystem	reference coordinate system	SMC_COORD_SYSTEM	SMC_COORD_SYSTEM	-	reference coordinate system
BufferMode	buffer mode	MC_BUFFER_MODE	-	0	Action when specifying multiple start motion commands
TransitionMode	Corner transition mode	MC_TRANSITION_MODE	TMNone/ TMStartVelocity/ TMCornerDistance	-	TMNone: No mixing TMStartVelocity: Speed based mixing TMCornerDistance: Distance based mixing
TransitionParameter	Corner transition parameters	array [0..(SMC_RCNST.MAX_TRANS_PARAMS - 1)] OF LREAL	0, positive number	0	Corner transition parameters
OrientationMode	interpolation positioning mode	SMC_ORIENTATION_MODE	GreatCircle/ Axis	-	GreatCircle: Move along the shortest path from the starting position to the target position. In this mode, even if the starting and ending positions are within the specified area, the implemented path may still leave this area. Axis: The positioning axis moves within the specified area from the start position to the end position, and not all kinematic transformations support this mode
VelFactor	speed factor	LREAL	0-1	1	Speed factor, the maximum speed of each axis multiplied by this speed factor, with a value between [0, 1]
AccFactor	acceleration factor	LREAL	0-1	1	Acceleration factor, the maximum velocity of each axis multiplied by this acceleration factor, with a value between [0,1]
JerkFactor	jerk speed factor	LREAL	0-1	1	Jerk factor, the maximum velocity of each axis multiplied by this jerk factor, with a value between [0,1]

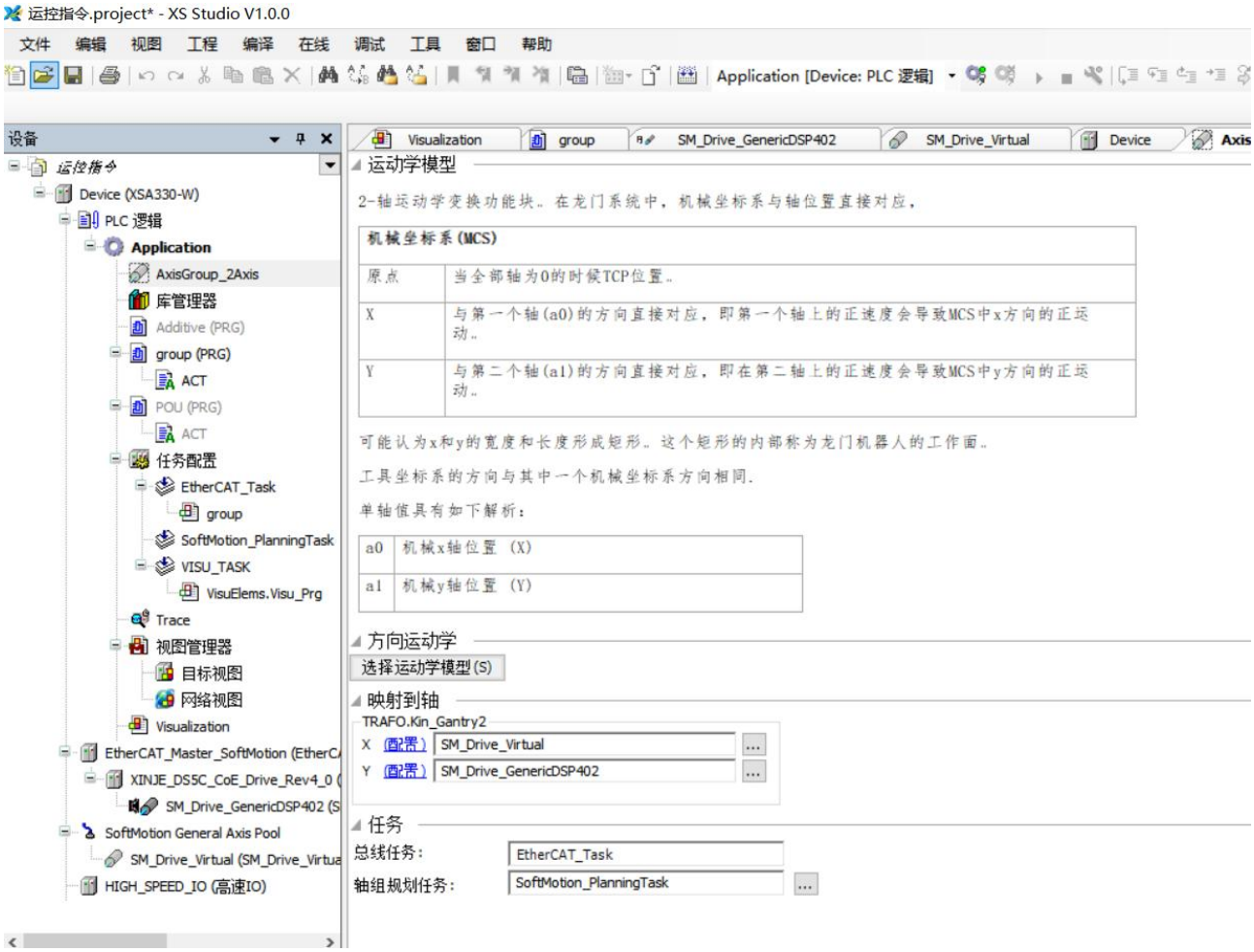
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	completed	BOOL	TRUE/FALSE	FALSE	TRUE when instruction execution is completed
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Active	in control	BOOL	TRUE/FALSE	FALSE	Change to TRUE in control
CommandAborted	Instruction interrupted	BOOL	TRUE/FALSE	FALSE	Module execution interrupted is true
CommandAccepted	Motion reception	BOOL	TRUE/FALSE	FALSE	True when the module successfully calls the axis group
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Error recognition
MovementId	motion flag	SMC_Movement_Id	TRUE/FALSE	FALSE	Valid when CommandAccepted or Done is TRUE

(3) Function description

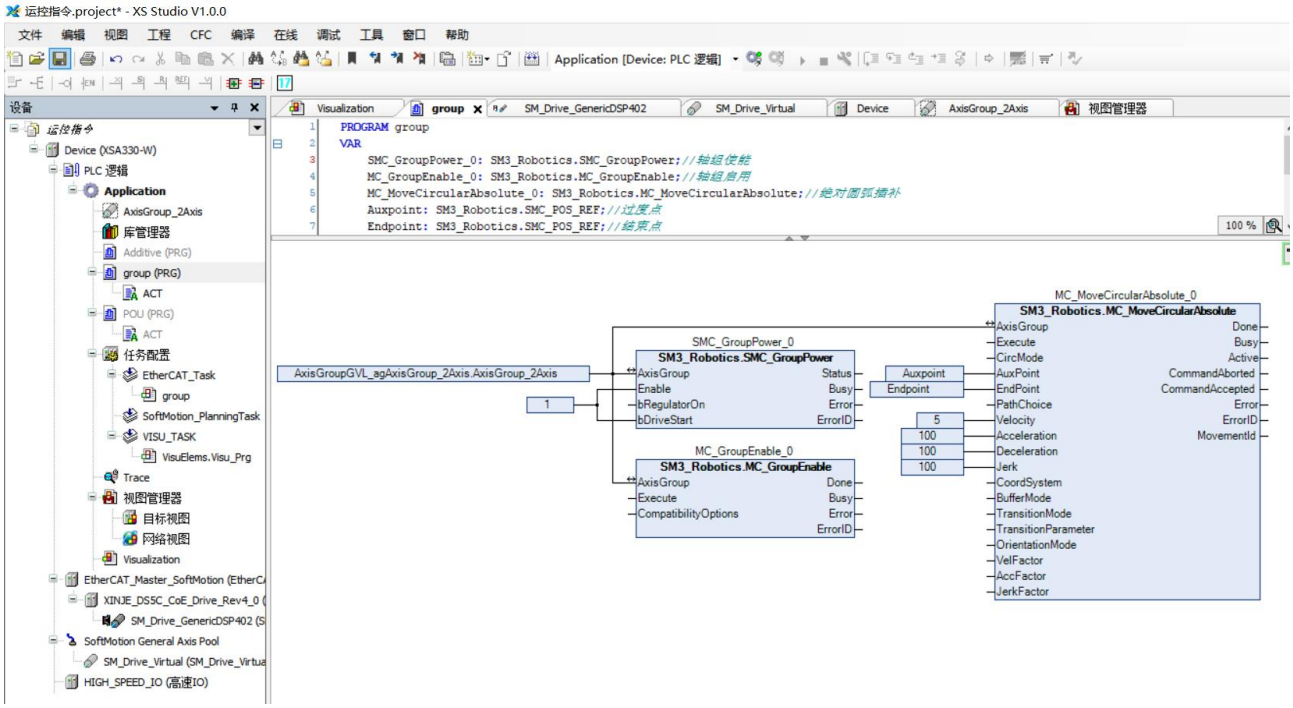
- The rising edge of the Execute triggers the command movement, while the falling edge has no effect on the command movement.
- When using the center arc mode, input the values of parameters AuxPoint [1] and AuxPoint [2] as the distance between the center of the circle and the starting point of the arc; When using the radius arc mode, AuxPoint [1] represents the radius value, and AuxPoint [2] is invalid.
- The numerical values of EndPoint [1]~EndPoint [8] represent the endpoint coordinates of each axis

(4) Application

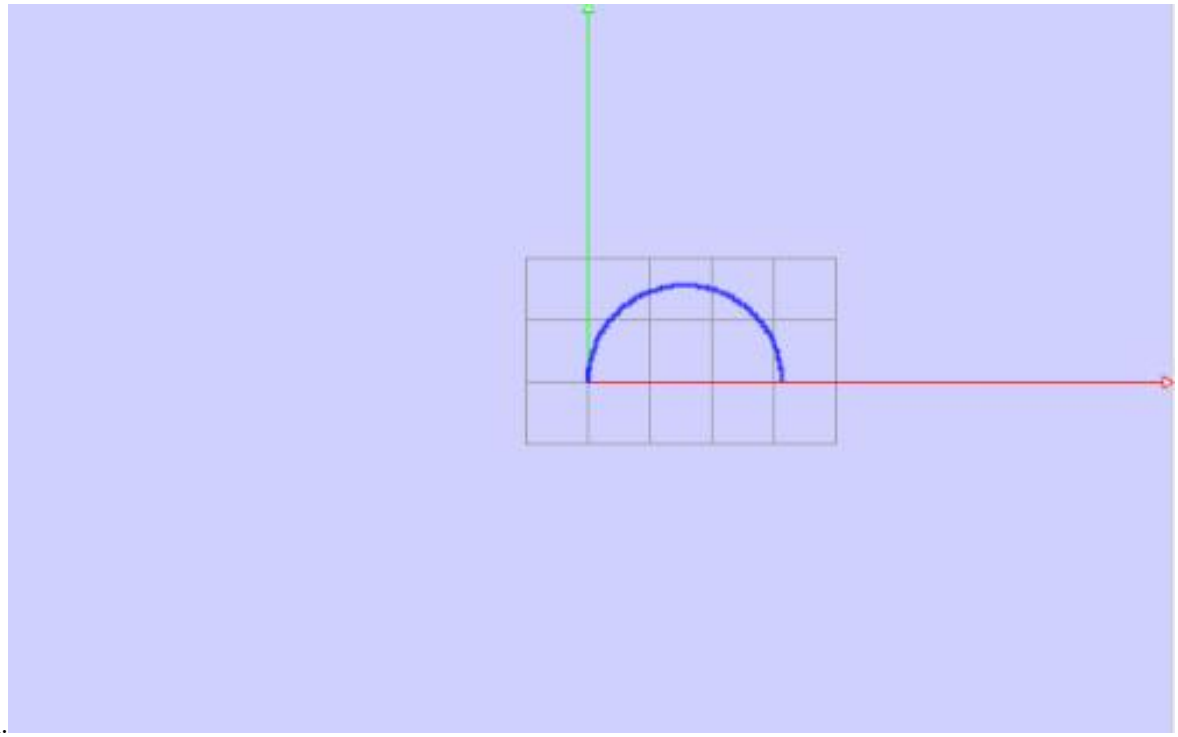
Example 1: Select the kinematic model of the two axes of the gantry, select the imaginary axis for the X-axis, and select the Ethercat real axis for the Y-axis to perform an absolute arc interpolation.



Programming: Set the current positions of both axes to 0, select the arc mode as three points to determine an arc, set AuxPoint to (50, 50), EndPoint to (100, 0), set the speed of both axes to 5, and draw an arc absolutely at the origin (0, 0).



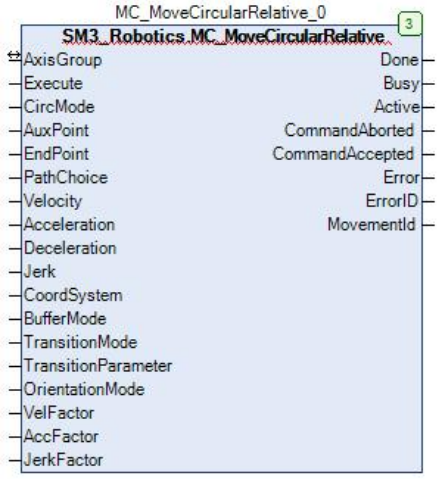
In the view, the arc trajectory can be

seen. 

3-2-2-17. Relative arc interpolation [MC_MoveCircularRelative]

(1) Instruction overview

Control the axis group to perform circular interpolation motion in relative position mode.

Instruction	Name	Graphical representation	ST language
MC_MoveCircularRelative	Relative arc interpolation		MC_MoveCircularRelative(AxisGroup:= , Execute:= , CircMode:= , AuxPoint:= , EndPoint:= , PathChoice:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , CoordSystem:= , BufferMode:= , TransitionMode:= , TransitionParameter:= , OrientationMode:= , VelFactor:= , AccFactor:= , JerkFactor:= , Done=> , Busy=> , Active=> , CommandAborted=> , CommandAccepted=> , Error=> , ErrorID=> , MovementId=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
VAR_INPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Execute	valid	BOOL	TRUE/FALSE	FALSE	Execute current instruction
CircMode	arc mode	SMC_CIRC_MODE	Border/Center/Radius	-	Specify the method of arc interpolation: Border: Three point arc Center: Center arc Radius: Radius arc
AuxPoint	Auxiliary point	SMC_POS_REF	-	-	Specify auxiliary points in the coordinate system. Refer to CircMode
EndPoint	end point	SMC_POS_REF	-	-	Specify the end position in the coordinate system
PathChoice	direction	MC_CIRC_PATHCHOICE	CLOCKWISE/COUNTER_CLOCKWISE	-	motion direction: CLOCKWISE: clockwise COUNTER_CLOCKWISE: counterclockwise

Velocity	speed	LREAL	0, positive number	0	Max resultant speed [command unit/s]
Deceleration	deceleration	LREAL	positive number	0	Max resultant deceleration [command unit/s^2]
Jerk	jerk speed	LREAL	positive number	0	Max resultant jerk speed [command unit/s^3]
CoordSystem	reference coordinate system	SMC_COORD_SYSTEM	SMC_COORD_SYSTEM	-	reference coordinate system
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
BufferMode	buffer mode	MC_BUFFER_MODE	-	0	specify the multiple start motion commands action
TransitionMode	Corner transition mode	MC_TRANSITION_MODE	TMNone/ TMStartVelocity/ TMCornerDistance	-	TMNone: no mixing TMStartVelocity: mixed based on speed TMCornerDistance: mixed based on distance
TransitionParameter	Corner transition parameters	array [0..(SMC_RCNST.MAX_TRANS_PARAMS - 1)] OF LREAL	0, positive number	0	Corner transition parameters
OrientationMode	Interpolation positioning mode	SMC_ORIENTATION_MODE	GreatCircle/ Axis	-	GreatCircle: Move along the shortest path from the starting position to the target position. In this mode, even if the starting and ending positions are within the specified area, the implemented path may still leave this area. Axis: The positioning axis moves within the specified area from the start position to the end position, and not all kinematic transformations support this mode
VelFactor	speed factor	LREAL	0-1	1	Speed factor, the maximum speed of each axis multiplied by this speed factor, with a value between [0, 1]
AccFactor	acceleration factor	LREAL	0-1	1	Acceleration factor, the maximum velocity of each axis multiplied by this acceleration factor, with a value between [0,1]
JerkFactor	jerk factor	LREAL	0-1	1	Jerk factor, the maximum velocity of each axis multiplied by this jerk factor, with a value between [0,1]

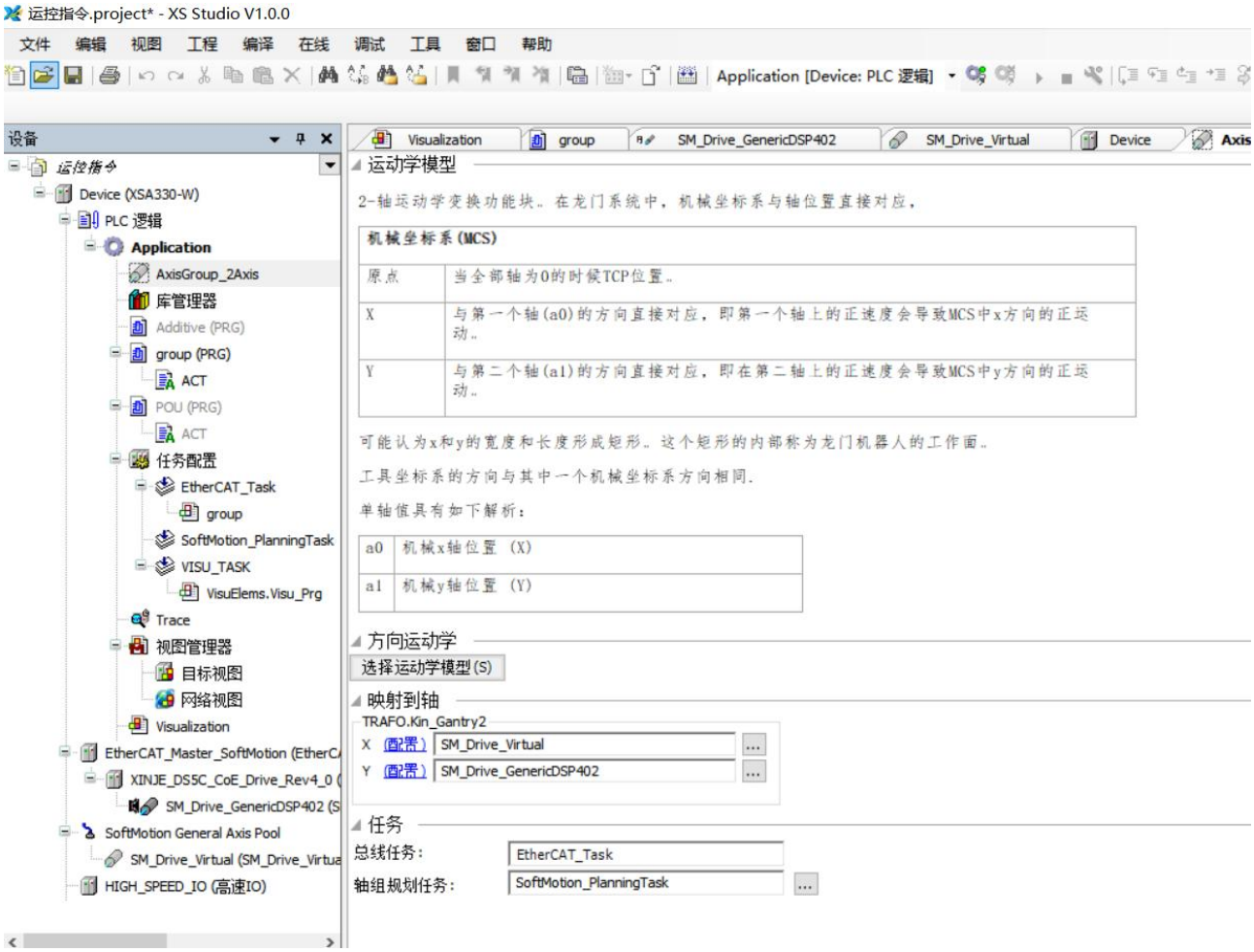
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	completed	BOOL	TRUE/FALSE	FALSE	TRUE when instruction execution is completed
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Active	in control	BOOL	TRUE/FALSE	FALSE	Change to TRUE in control
CommandAborted	Instruction interrupted	BOOL	TRUE/FALSE	FALSE	Module execution interrupted is true
CommandAccepted	Motion reception	BOOL	TRUE/FALSE	FALSE	True when the module successfully calls the axis group
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Error recognition
MovementId	motion flag	SMC_Movement_Id	TRUE/FALSE	FALSE	Valid when CommandAccepted or Done is TRUE

(3) Function description

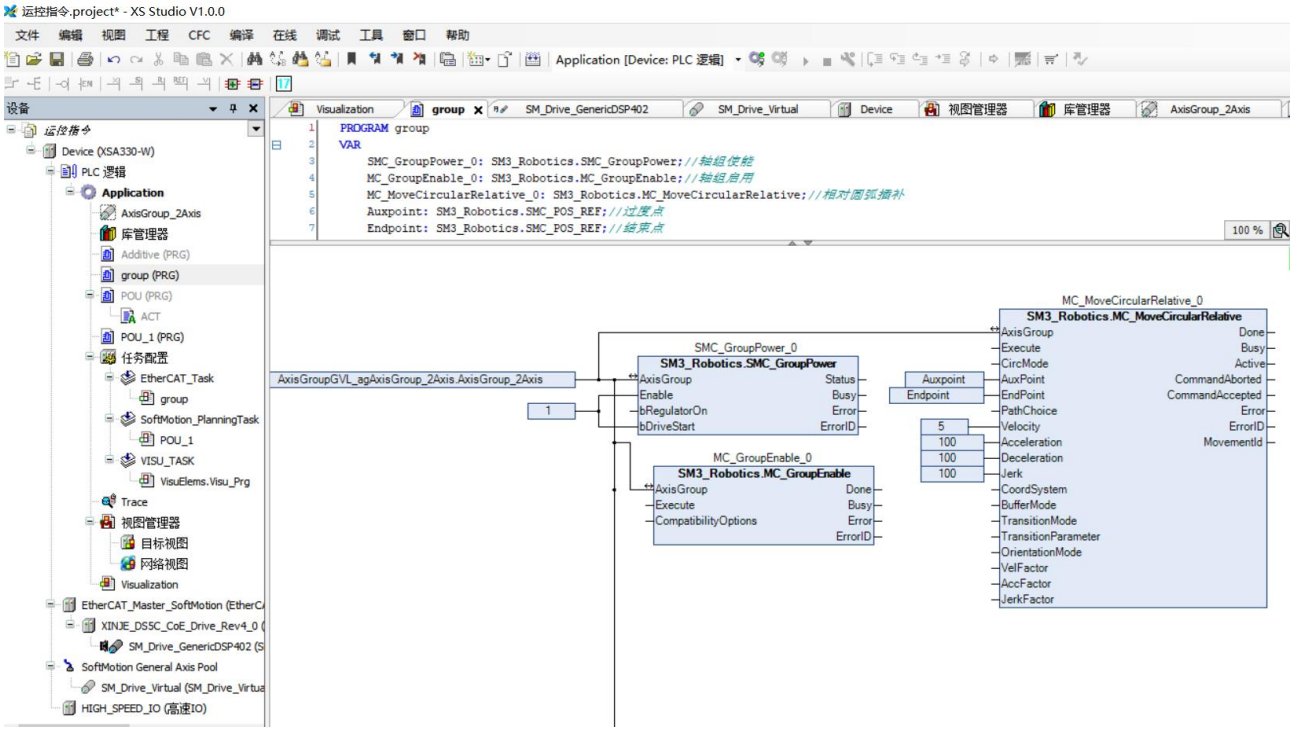
- The rising edge of the Execute triggers the command movement, while the falling edge has no effect on the command movement.
- When using the center arc mode, input the values of parameters AuxPoint [1] and AuxPoint [2] as the distance between the center of the circle and the starting point of the arc. When using the radius arc mode, AuxPoint [1] represents the radius value, and AuxPoint [2] is invalid.
- The numerical values of EndPoint [1]~EndPoint [8] represent the endpoint coordinates of each axis.

(4) Application

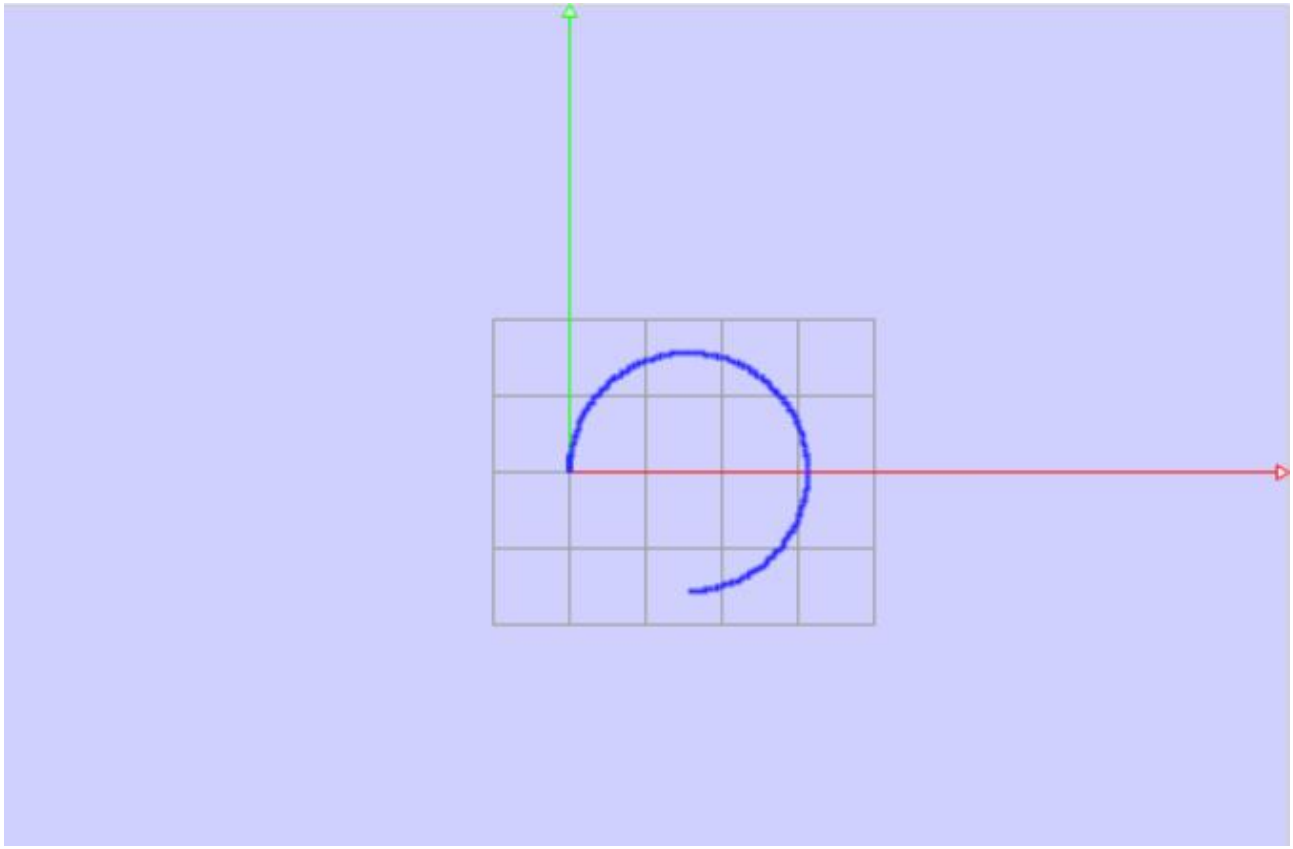
Example 1: Select the kinematic model of the two axes of the gantry, select the imaginary axis for the X-axis, and select the Ethercat real axis for the Y-axis to perform a relative arc interpolation.



Programming: Set the current positions of both axes to 0, select the arc mode as three points to determine an arc, set the auxiliary point AuxPoint to (50, 50), set the EndPoint to (50, -50), set the speed of both axes to 5, and draw an arc relative to the current point (0, 0).



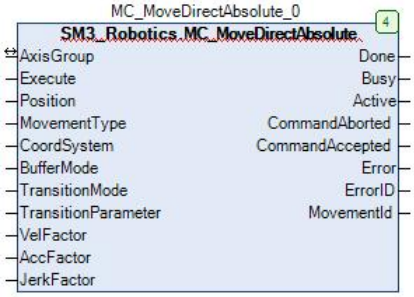
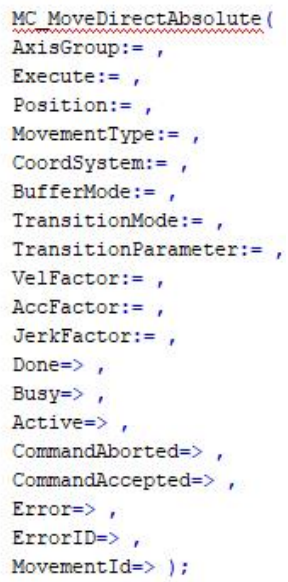
In the view, the arc trajectory can be seen.



3-2-2-18. Absolute position quick positioning [MC_MoveDirectAbsolute]

(1) Instruction overview

Control all axes within the axis group to run at the specified speed to the absolute position endpoint.

Instruction	Name	Graphical representation	ST language
MC_MoveDirectAbsolute	Absolute position quick positioning		


```
MC_MoveDirectAbsolute(
AxisGroup:= ,
Execute:= ,
Position:= ,
MovementType:= ,
CoordSystem:= ,
BufferMode:= ,
TransitionMode:= ,
TransitionParameter:= ,
VelFactor:= ,
AccFactor:= ,
JerkFactor:= ,
Done=> ,
Busy=> ,
Active=> ,
CommandAborted=> ,
CommandAccepted=> ,
Error=> ,
ErrorID=> ,
MovementId=> );
```

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
VAR_INPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Execute	valid	BOOL	TRUE/FALSE	FALSE	Execute current instruction
Position	position	SMC_POS_REF	-	-	Absolute target position in the specified reference coordinate system
MovementType	PTP motion mode	SMC_PTP_MOVEMENT_TYPE	-	-	Fast (0): PTP motion mode with time priority Path_ Invariant: PTP motion with fixed path
CoordSystem	reference coordinate system	SMC_COORD_SYSTEM	SMC_COORD_SYSTEM		reference coordinate system
BufferMode	buffer mode	MC_BUFFER_MODE	-	0	specify the multiple start motion commands action
TransitionMode	Corner transition mode	MC_TRANSITION_MODE	TMNone/ TMStartVelocity /TMCornerDistance	-	TMNone: no mixing TMStartVelocity: mixed based on speed TMCornerDistance: mixed based on distance
Transition	Corner	array	0, positive	0	Corner transition parameters

Parameter	transition parameters	[0..(SMC_RCNST.MAX_TRANS_PARAMS - 1)] OF LREAL	number		
VelFactor	speed factor	LREAL	0-1	1	Speed factor, the maximum speed of each axis multiplied by this speed factor, with a value between [0, 1]
AccFactor	acceleration factor	LREAL	0-1	1	Acceleration factor, the maximum velocity of each axis multiplied by this acceleration factor, with a value between [0,1]
JerkFactor	jerk factor	LREAL	0-1	1	Jerk factor, the maximum velocity of each axis multiplied by this jerk factor, with a value between [0,1]
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	completed	BOOL	TRUE/FALSE	FALSE	TRUE when instruction execution is completed
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Active	in control	BOOL	TRUE/FALSE	FALSE	Change to TRUE in control
CommandAborted	Instruction interrupted	BOOL	TRUE/FALSE	FALSE	Module execution interrupted is true
CommandAccepted	Motion reception	BOOL	TRUE/FALSE	FALSE	True when the module successfully calls the axis group
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Error recognition
MovementId	motion flag	SMC_Movement_Id	TRUE/FALSE	FALSE	True when the motion is being executed or completed

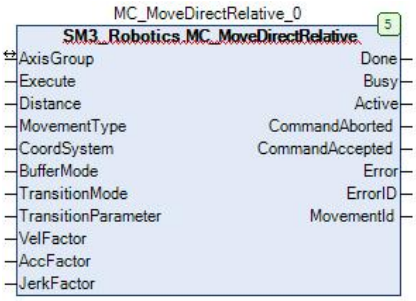
(3) Function description

- In motion, each axis is a relatively independent absolute position motion, and its motion trajectory is uncertain.

3-2-2-19. Relative position quick positioning [MC_MoveDirectRelative]

(1) Instruction overview

Control all axes within the axis group to run at the specified speed to the end point of the relative position.

Instruction	Name	Graphical representation	ST language
MC_MoveDirectRelative	relative position quick positioning		MC_MoveDirectRelative(AxisGroup:= , Execute:= , Distance:= , MovementType:= , CoordSystem:= , BufferMode:= , TransitionMode:= , TransitionParameter:= , VelFactor:= , AccFactor:= , JerkFactor:= , Done=> , Busy=> , Active=> , CommandAborted=> , CommandAccepted=> , Error=> , ErrorID=> , MovementId=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
VAR_INPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Execute	valid	BOOL	TRUE/FALSE	FALSE	Execute current instruction
Distance	relative position	SMC_POS_REF	-	-	Relative target position in the specified reference coordinate system
MovementType	PTP motion mode	SMC_PTP_MOVEMENT_TYPE	-	-	Fast (0): PTP motion mode with time priority Path_ Invariant: PTP motion with fixed path
CoordSystem	reference coordinate system	SMC_COORD_SYSTEM	SMC_COORD_SYSTEM		reference coordinate system
BufferMode	buffer mode	MC_BUFFER_MODE	-	0	specify the multiple start motion commands action
TransitionMode	Corner transition mode	MC_TRANSITION_MODE	TMNone/TMStartVelocity/TMCornerDistance		TMNone: no mixing TMStartVelocity: mixed based on speed TMCornerDistance: mixed based on distance
Transition Parameter	Corner transition parameters	array [0..(SMC_RCNST.MAX_TRANS_PARA	0, positive number	0	Corner transition parameters

		MS - 1)] OF LREAL			
VelFactor	speed factor	LREAL	0-1	1	Speed factor, the maximum speed of each axis multiplied by this speed factor, with a value between [0, 1]
AccFactor	acceleration factor	LREAL	0-1	1	Acceleration factor, the maximum velocity of each axis multiplied by this acceleration factor, with a value between [0,1]

VAR_INPUT	Name	Data type	Effective range	Initial value	Description
JerkFactor	jerk factor	LREAL	0-1	1	Jerk factor, the maximum velocity of each axis multiplied by this jerk factor, with a value between [0,1]
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	completed	BOOL	TRUE/FALSE	FALSE	TRUE when instruction execution is completed
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Active	in control	BOOL	TRUE/FALSE	FALSE	Change to TRUE in control
Command Aborted	Instruction interrupted	BOOL	TRUE/FALSE	FALSE	Module execution interrupted is true
Command Accepted	Motion reception	BOOL	TRUE/FALSE	FALSE	True when the module successfully calls the axis group
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Error recognition
MovementId	motion flag	SMC_Movement_Id	TRUE/FALSE	FALSE	True when the motion is being executed or completed

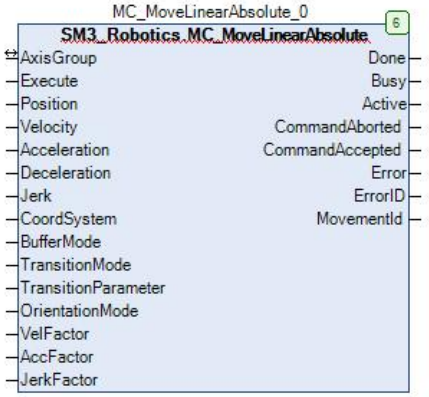
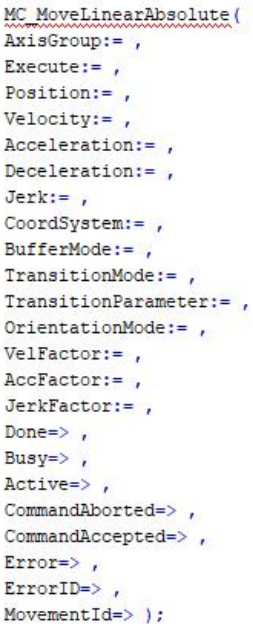
(3) Function description

- In motion, each axis is a relatively independent absolute position motion, and its motion trajectory is uncertain.

3-2-2-20. Absolute position linear interpolation [MC_MoveLinearAbsolute]

(1) Instruction overview

The linear interpolation motion of the absolute position mode of the control axis group in the specified coordinate system.

Instruction	Name	Graphical representation	ST language
MC_MoveLinearAbsolute	absolute position linear interpolation		


```
MC_MoveLinearAbsolute(
AxisGroup:= ,
Execute:= ,
Position:= ,
Velocity:= ,
Acceleration:= ,
Deceleration:= ,
Jerk:= ,
CoordSystem:= ,
BufferMode:= ,
TransitionMode:= ,
TransitionParameter:= ,
OrientationMode:= ,
VelFactor:= ,
AccFactor:= ,
JerkFactor:= ,
Done=> ,
Busy=> ,
Active=> ,
CommandAborted=> ,
CommandAccepted=> ,
Error=> ,
ErrorID=> ,
MovementId=> );
```

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
VAR_INPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Execute	valid	BOOL	TRUE /FALSE	FALSE	Execute current instruction
Position	position	SMC_POS_REF	-	-	Absolute target position in the specified reference coordinate system
Velocity	speed	LREAL	0, positive number	0	Max resultant speed [command unit/s]
Acceleration	acceleration	LREAL	positive number	0	Max resultant acceleration [command unit /s^2]
Deceleration	deceleration	LREAL	positive number	0	Max resultant deceleration [command unit /s^2]
Jerk	jerk speed	LREAL	positive number	0	Max resultant jerk speed [command unit /s^3]
CoordSystem	reference coordinate system	SMC_COORD_SYSTEM	SMC_COORD_SYSTEM	-	reference coordinate system
BufferMode	buffer mode	MC_BUFFER_MODE	-	0	specify the multiple start motion commands action

TransitionMode	Corner transition mode	MC_TRANSITION_MODE	TMNone/TMStartVelocity/TMCornerDistance	-	TMNone: no mixing TMStartVelocity: mixed based on speed TMCornerDistance: mixed based on distance
TransitionParameter	Corner transition parameters	array [0..(SMC_RCNST.MAX_TRANS_PARAMS - 1)] OF LREAL	0, positive number	0	corner transition parameters

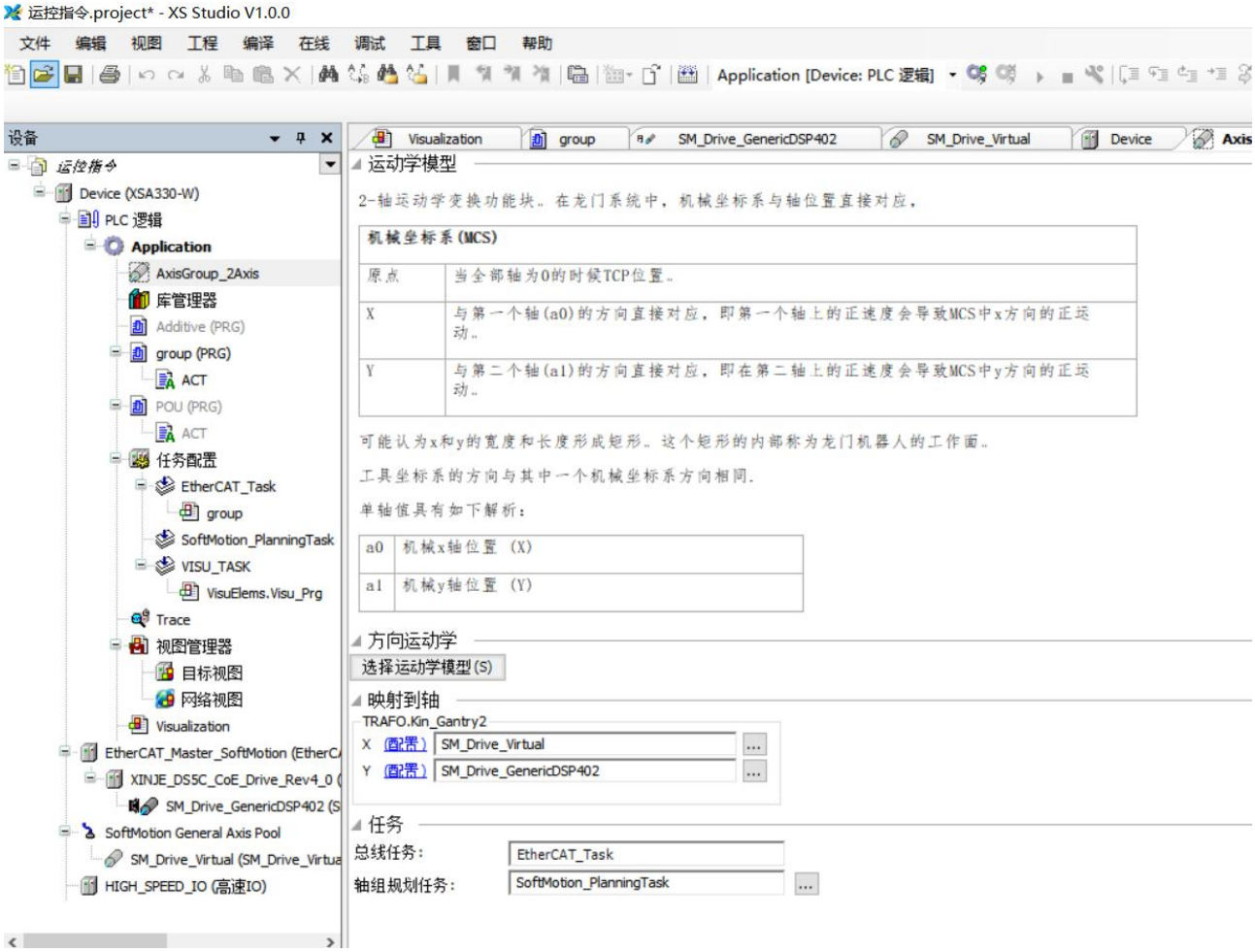
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
OrientationMode	interpolation positioning mode	SMC_ORIENTATION_MODE	GreatCircle/ Axis	-	GreatCircle: Move along the shortest path from the starting position to the target position. In this mode, even if the starting and ending positions are within the specified area, the implemented path may still leave this area. Axis: The positioning axis moves within the specified area from the start position to the end position, and not all kinematic transformations support this mode
VelFactor	speed factor	LREAL	0-1	1	Speed factor, the maximum speed of each axis multiplied by this speed factor, with a value between [0, 1]
AccFactor	acceleration factor	LREAL	0-1	1	Acceleration factor, the maximum velocity of each axis multiplied by this acceleration factor, with a value between [0,1]
JerkFactor	jerk factor	LREAL	0-1	1	Jerk factor, the maximum velocity of each axis multiplied by this jerk factor, with a value between [0,1]
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	completed	BOOL	TRUE/FALSE	FALSE	TRUE when instruction execution is completed
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Active	in control	BOOL	TRUE/FALSE	FALSE	Change to TRUE in control
CommandAborted	Instruction interrupted	BOOL	TRUE/FALSE	FALSE	Module execution interrupted is true
CommandAccepted	Motion reception	BOOL	TRUE/FALSE	FALSE	True when the module successfully calls the axis group
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Error recognition
MovementId	motion flag	SMC_Movemen	TRUE/FALSE	FALSE	True when the motion is being executed

		t_Id			or completed

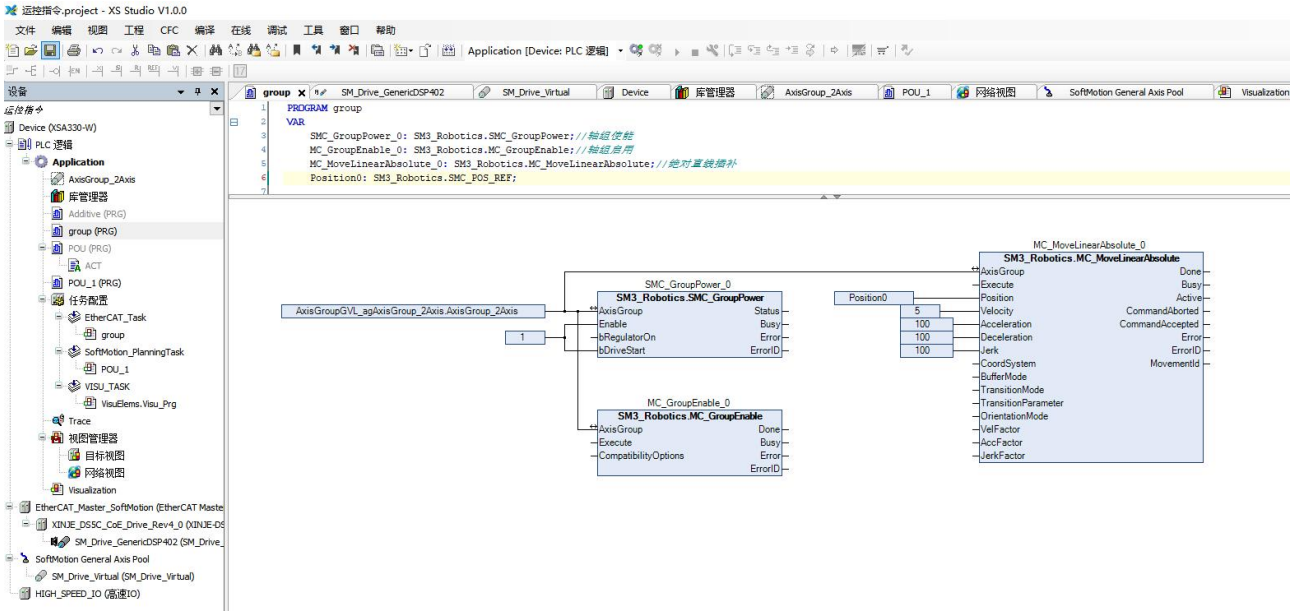
(3) Function description

- Control the linear interpolation motion of the axis group in the absolute position mode under the specified coordinate system.
- The relationship between the parameters BufferMode, TransitionMode, and TransitionParameter is explained as follows:
  (1) When BufferMode selects mcBuffered mode, TransitionMode only supports mcTMNone mode;
  (2) When BufferMode selects mcBlendingPrevious mode, TransitionMode can be selected.
- McTMConstantVelocity and mcTMCornerDistance modes.

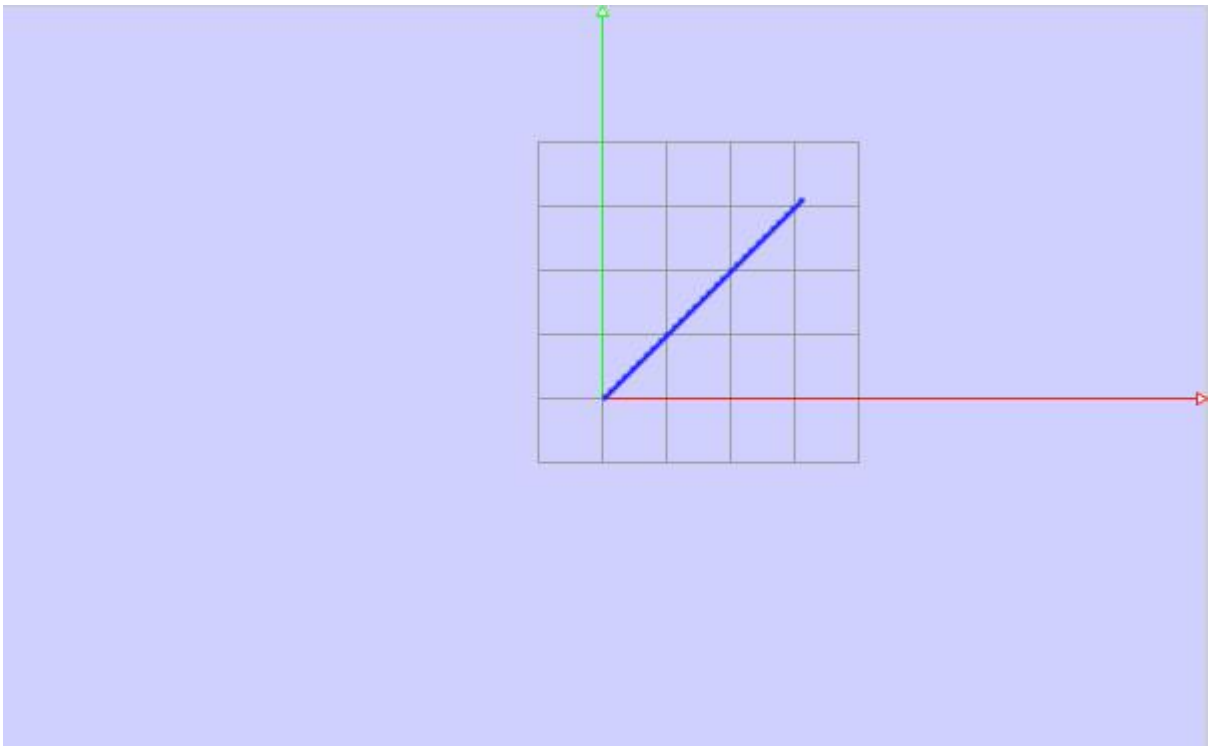
Example 1: Select the kinematic model of the two axes of the gantry, select the imaginary axis for the X-axis, and select the Ethercat real axis for the Y-axis to perform an absolute position interpolation.



Programming: Set the current positions of both axes to 0, set the position to (100, 100), and perform absolute linear interpolation motion starting from the starting point (0,0).



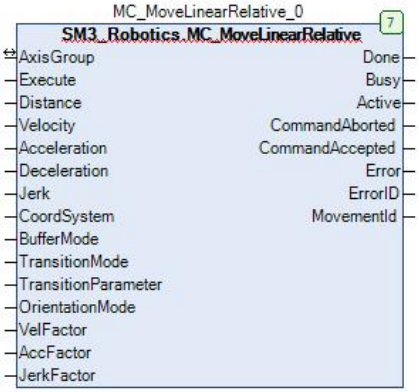
In the view, the straight line trajectory can be seen.



3-2-2-21. Relative position linear interpolation [MC_MoveLinearRelative]

(1) Instruction overview

The linear interpolation motion of the relative position mode of the control axis group in the specified coordinate system.

Instruction	Name	Graphical representation	ST language
MC_MoveLinearRelative	relative position linear interpolation		MC_MoveLinearRelative(AxisGroup:= , Execute:= , Distance:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , CoordSystem:= , BufferMode:= , TransitionMode:= , TransitionParameter:= , OrientationMode:= , VelFactor:= , AccFactor:= , JerkFactor:= , Done=> , Busy=> , Active=> , CommandAborted=> , CommandAccepted=> , Error=> , ErrorID=> , MovementId=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
VAR_INPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Execute	valid	BOOL	TRUE/FALSE	FALSE	Execute current instruction
Distance	relative position	SMC_POS_REF	-	-	Relative target position in the specified reference coordinate system
Velocity	speed	LREAL	0, positive number	0	Max resultant speed [command unit/s]
Acceleration	acceleration	LREAL	positive number	0	Max resultant acceleration [command unit /s^2]
Deceleration	deceleration	LREAL	positive number	0	Max resultant deceleration [command unit /s^2]
Jerk	jerk speed	LREAL	positive number	0	Max resultant jerk speed [command unit /s^3]
CoordSystem	reference coordinate system	SMC_COORD_SYSTEM	SMC_COORD_SYSTEM	-	reference coordinate system
BufferMode	buffer mode	MC_BUFFER_	-	0	specify the multiple start motion

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
		MODE			commands action
TransitionMode	Corner transition mode	MC_TRANSITION_MODE	TMNone/TMStartVelocity/TMCornerDistance	-	TMNone: no mixing TMStartVelocity: mixed based on speed TMCornerDistance: mixed based on distance

VAR_INPUT	Name	Data type	Effective range	Initial value	Description
TransitionParameter	Corner transition parameters	array [0..(SMC_RCNST.MAX_TRANS_PARAMS - 1)] OF LREAL	0, positive number	0	Corner transition parameters
OrientationMode	Interpolation positioning mode	SMC_ORIENTATION_MODE	GreatCircle /Axis	-	GreatCircle: Move along the shortest path from the starting position to the target position. In this mode, even if the starting and ending positions are within the specified area, the implemented path may still leave this area. Axis: The positioning axis moves within the specified area from the start position to the end position, and not all kinematic transformations support this mode
VelFactor	speed factor	LREAL	0-1	1	Speed factor, the maximum speed of each axis multiplied by this speed factor, with a value between [0, 1]
AccFactor	acceleration factor	LREAL	0-1	1	Acceleration factor, the maximum velocity of each axis multiplied by this acceleration factor, with a value between [0,1]
JerkFactor	jerk factor	LREAL	0-1	1	Jerk factor, the maximum velocity of each axis multiplied by this jerk factor, with a value between [0,1]
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	completed	BOOL	TRUE/FALSE	FALSE	TRUE when instruction execution is completed
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended

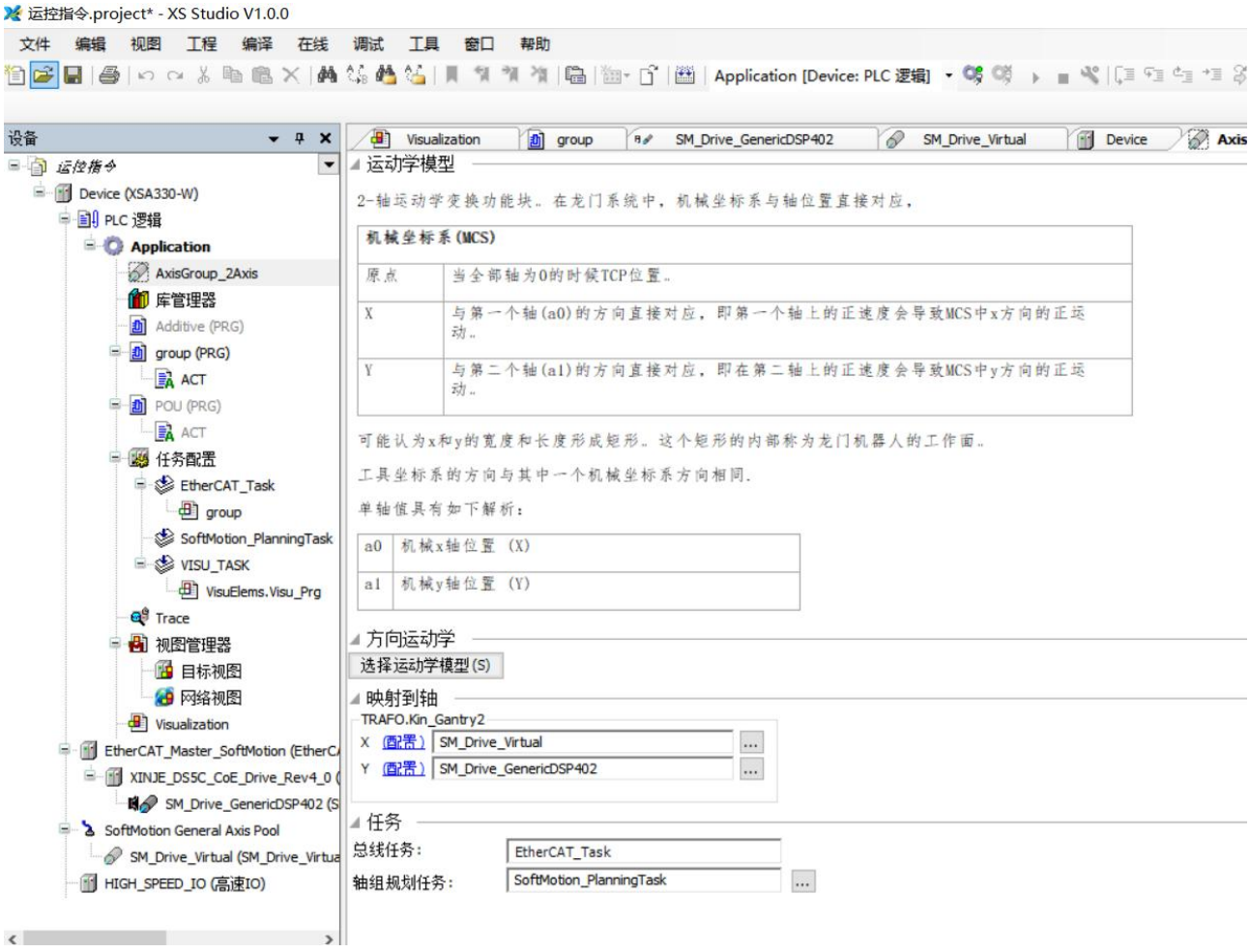
Active	in control	BOOL	TRUE/FALSE	FALSE	Change to TRUE in control
CommandAborted	Instruction interrupted	BOOL	TRUE/FALSE	FALSE	Module execution interrupted is true
CommandAccepted	Motion reception	BOOL	TRUE/FALSE	FALSE	True when the module successfully calls the axis group
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Error recognition
MovementId	motion flag	SMC_Movement_Id	TRUE/FALSE	FALSE	True when the motion is being executed or completed

(3) Function description

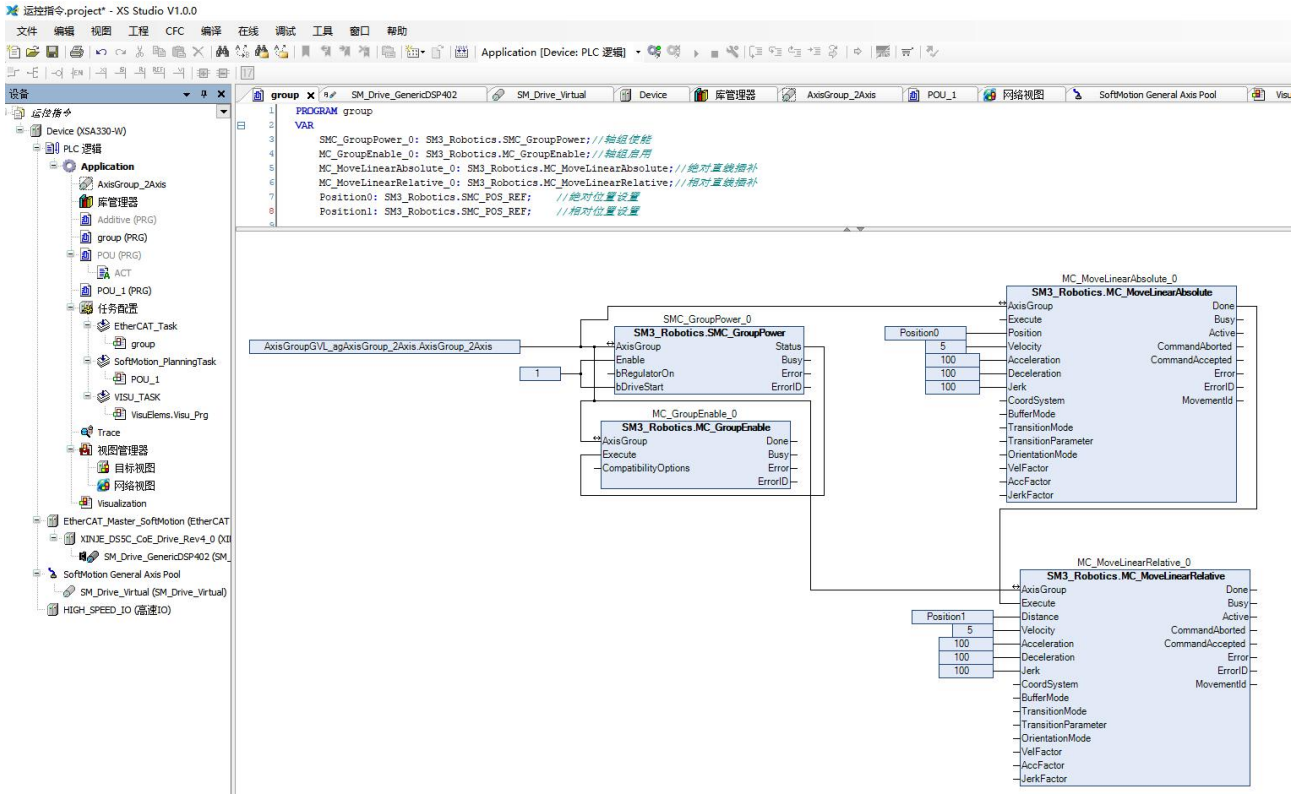
- The parameter TransitionParameter is only valid when the TransitionMode is mcTMCornerDistance.

(4) Application

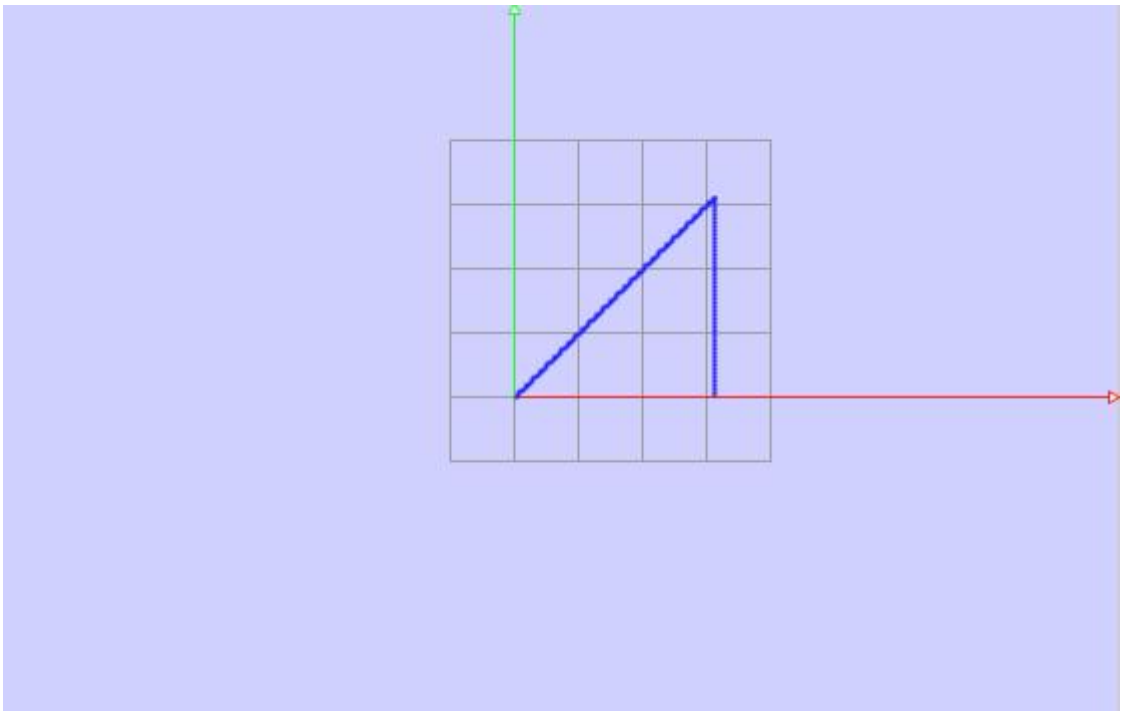
Example 1: Select the kinematic model of the two axes of the gantry, select the imaginary axis for the X-axis, and select the Ethercat real axis for the Y-axis to perform a relative position interpolation.



Programming: Set the current positions of both axes to 0, that is, the starting position is (0, 0), the absolute linear interpolation position is set to (100, 100), and then the relative linear interpolation position is set to (0, -100).



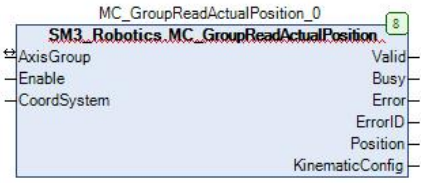
In the view, the straight line trajectory can be seen.



3-2-2-22. Read the feedback position of the axis group [MC_GroupReadActualPosition]

(1) Instruction overview

Read the feedback position of the axis group in the specified coordinate system.

Instruction	Name	Graphical representation	ST language
MC_GroupReadActualPosition	read axis group feedback position		MC_GroupReadActualPosition(AxisGroup:= , Enable:= , CoordSystem:= , Valid=> , Busy=> , Error=> , ErrorID=> , Position=> , KinematicConfig=>);

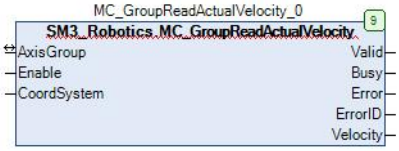
(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Enable	valid	BOOL	TRUE/FALSE	FALSE	Must be set to TRUE to activate the processing of the function block
CoordSystem	Apply coordinate system	SMC_COORD_SYSTEM	-	-	Apply coordinate system
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Valid	obtain flag	BOOL	TRUE/FALSE	FALSE	If the output value is valid, then it is TRUE
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	error recognization
Position	position	SMC_POS_REF	-	-	axis group actual position
KinematicConfig	kinematic configuration	TRAFO.CONFIGDATA	-	-	The kinematic configuration of the current position. Only set when the coordinate system is Cartesian (i.e. not set in ACS)

3-2-2-23. Read the feedback speed of the axis group [MC_GroupReadActualVelocity]

(1) Instruction overview

Read the feedback speed of the axis group in the specified coordinate system.

Instruction	Name	Graphical representation	ST language
MC_GroupReadActualVelocity	read axis group feedback speed		MC_GroupReadActualVelocity(AxisGroup:= , Enable:= , CoordSystem:= , Valid=> , Busy=> , Error=> , ErrorID=> , Velocity=>);

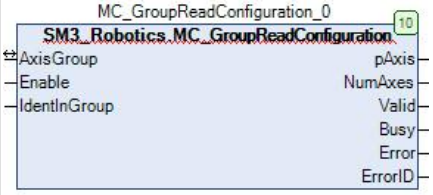
(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Enable	valid	BOOL	TRUE/FALSE	FALSE	Must be set to TRUE to activate the processing of the function block
CoordSystem	Apply coordinate system	SMC_COORD_SYSTEM	-	-	Apply coordinate system
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Valid	obtain flag	BOOL	TRUE/FALSE	FALSE	If the output value is valid, then it is TRUE
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	error recognization
Velocity	speed	SMC_POS_REF	-	-	The current feedback speed of the axis group. If the Cartesian coordinate system is selected; Velocity. c contains Cartesian velocities: (X, Y, Z) is the velocity vector, and (A, B, C) is the angular velocity around the x, y, and z axes, respectively

3-2-2-24. Read axis group configuration parameters [MC_GroupReadConfiguration]

(1) Instruction overview

Read the configuration parameters such as axis and quantity contained in the axis group.

Instruction	Name	Graphical representation	ST language
MC_GroupReadConfiguration	read axis group configuration parameters		MC_GroupReadConfiguration(AxisGroup:= , Enable:= , IdentInGroup:= , pAxis=> , NumAxes=> , Valid=> , Busy=> , Error=> , ErrorID=>);

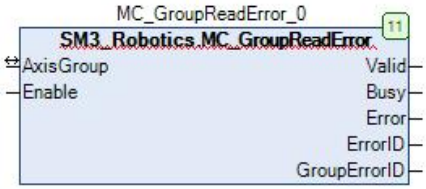
(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
VAR_INPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Enable	valid	BOOL	TRUE/FALSE	FALSE	Must be set to TRUE to activate the processing of the function block
IdentInGroup	axis number	IDENT_IN_GROUP_REF_SM3	0, positive number	0	Enter the number of the corresponding axis in the axis group
VAR_OUTPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
pAxis	reference axis	POINTER TOAXIS_REF_SM3	-	-	selected reference axis
NumAxes	axis quantity	UDINT	0, positive number	0	axis quantity in the axis group
Valid	obtain flag	BOOL	TRUE/FALSE	FALSE	If the output value is valid, then it is TRUE
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	error recognization

3-2-2-25. Read axis group error [MC_GroupReadError]

(1) Instruction overview

Obtain error information for the axis group.

Instruction	Name	Graphical representation	ST language
MC_GroupReadError	read axis group error		MC_GroupReadError(AxisGroup:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , GroupErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Enable	valid	BOOL	TRUE/FALSE	FALSE	Must be set to TRUE to activate the processing of the function block
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Valid	obtain flag	BOOL	TRUE/FALSE	FALSE	If the output value is valid, then it is TRUE
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Single axis error code indication
GroupErrorID	error code	SMC_ERROR	-	0	Axis group error code indication

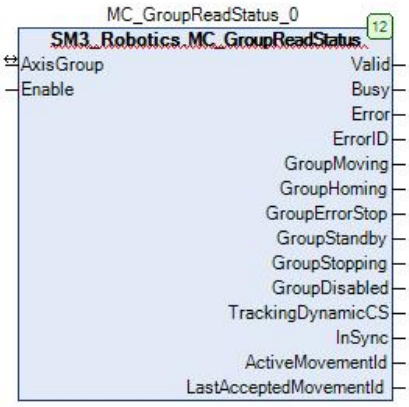
(3) Function description

- Obtain error information for single axis and axis group, such as reading hard limit (or soft limit) or single axis error.

3-2-2-26. Read the current operating status of the axis group [MC_GroupReadStatus]

(1) Instruction overview

Used to obtain the current motion state of the axis group.

Instruction	Name	Graphical representation	ST language
MC_GroupReadStatus	read axis group current motion status		MC_GroupReadStatus(AxisGroup:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , GroupMoving=> , GroupHoming=> , GroupErrorStop=> , GroupStandby=> , GroupStopping=> , GroupDisabled=> , TrackingDynamicCS=> , InSync=> , ActiveMovementId=> , LastAcceptedMovementId=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Enable	valid	BOOL	TRUE/FALSE	FALSE	Must be set to TRUE to activate the processing of the function block
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Valid	obtain flag	BOOL	TRUE/FALSE	FALSE	If the output value is valid, then it is TRUE
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Single axis error code indication
GroupErrorID	error code	SMC_ERROR	-	0	Axis group error code indication
GroupMoving	moving	BOOL	TRUE/FALSE	FALSE	Axis group in motion is TRUE
GroupHoming	homing	BOOL	TRUE/FALSE	FALSE	Axis group homing is TRUE
GroupErrorStop	error stop	BOOL	TRUE/FALSE	FALSE	Axis group error stop is true
GroupStandby	ready to move	BOOL	TRUE/FALSE	FALSE	Axis group motion preparation status is TRUE
GroupStopping	stopping	BOOL	TRUE/FALSE	FALSE	True when the axis group movement stops
GroupDisabled	not enable axis group	BOOL	TRUE/FALSE	FALSE	Axis group invalid status is TRUE
TrackingDynamicCS	Currently in	BOOL	TRUE/FALSE	FALSE	True when currently using a

	a dynamic coordinate system				dynamic coordinate system
InSync	On the path or already in place	BOOL	TRUE/FALSE	FALSE	True in continuous interpolation motion when the obtained position belongs to the specified path; In normal point position and interpolation motion, it is true when the current position is equal to the target position
ActiveMovementId	motion segment number	SMC_Movement_Id	0, positive number	-	The identifier of the activity movement. Id=0 indicates no active movement

VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
LastAcceptedMovementId	motion segment number	SMC_Movement_Id	0, positive number	0	The identifier of the last accepted motion. Id=0 indicates that no action has been taken yet

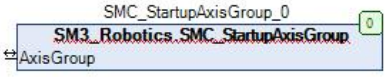
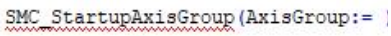
(3) Function description

- Calling this command can continuously obtain information such as the motion status of the axis group, whether it is in place, and the motion segment number.

3-2-2-27. Startup the axis group [SMC_StartupAxisGroup]

(1) Instruction overview

Used for testing and debugging axis groups.

Instruction	Name	Graphical representation	ST language
SMC_StartupAxisGroup	startup the axis group		

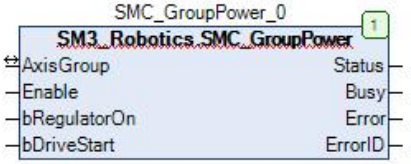
(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group

3-2-2-28. Enable the axis group [SMC_GroupPower]

(1) Instruction overview

Enable all axes under the axis group, equivalent to calling MC_Power for all axes under the axis group.

Instruction	Name	Graphical representation	ST language
SMC_GroupPower	enable the axis group		SMC_GroupPower(AxisGroup:= , Enable:= , bRegulatorOn:= , bDriveStart:= , Status=> , Busy=> , Error=> , ErrorID=>);

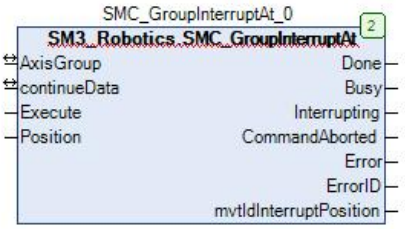
(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Enable	valid	BOOL	TRUE/FALSE	FALSE	Enter TRUE to start the module running
bRegulatorOn	enable	BOOL	TRUE/FALSE	FALSE	Set TRUE to enable the axis group
bDriveStart	enable drive	BOOL	TRUE/FALSE	FALSE	Must be set to TRUE to turn off emergency stop processing for the function block
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Status	can run	BOOL	TRUE/FALSE	FALSE	True when the axis group is ready
Valid	obtain flag	BOOL	TRUE/FALSE	FALSE	If the output value is valid, then it is TRUE
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	error code

3-2-2-29. Break Assignment [SMC_GroupInterruptAt]

(1) Instruction overview

Interrupt the currently moving axis group.

Instruction	Name	Graphical representation	ST language
SMC_GroupInterruptAt	break assignment		SMC_GroupInterruptAt(AxisGroup:= , continueData:= , Execute:= , Position:= , Done=> , Busy=> , Interrupting=> , CommandAborted=> , Error=> , ErrorID=> , mvtIdInterruptPosition=>);

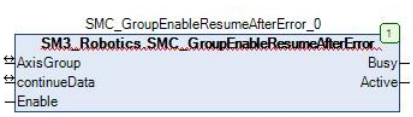
(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
continueData	Continuing motion data	SMC_AXIS_GROUP_CONTINUE_DATA	-	-	Motion information when axis group motion is interrupted
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Execute	valid	BOOL	TRUE/FALSE	FALSE	Execute current instruction
Position	position	SMC_GroupInterruptPosition	-	-	The location where the ongoing movement will be interrupted
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	completed	BOOL	TRUE/FALSE	FALSE	TRUE when instruction execution is completed
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Interrupting	interrupting	BOOL	TRUE/FALSE	FALSE	Signal indicating that the function block is currently being interrupted
CommandAborted	Instruction interrupted	BOOL	TRUE/FALSE	FALSE	Module execution interrupted is true
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Error code
mvtIdInterruptPosition	Interrupt location ID	-	-	-	MovemenID corresponding to the interrupt location

3-2-2-30. Reboot after error reset [SMC_GroupEnableResumeAfterError]

(1) Instruction overview

Restore the axis group state that was interrupted due to an error.

Instruction	Name	Graphical representation	ST language
SMC_GroupEnableResumeAfterError	restart after error reset		SMC_GroupEnableResumeAfterError(AxisGroup:= , continueData:= , Enable:= , Busy=> , Active=>);

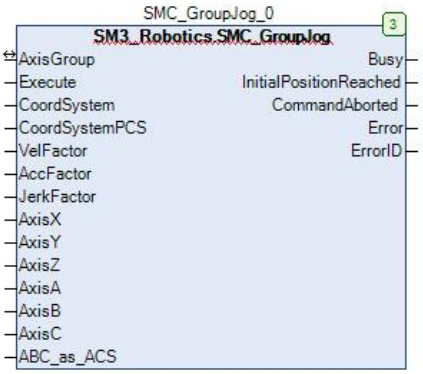
(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
continueData	Continuing motion data	SMC_AXIS_GROUP_CONTINUE_DATA	-	-	Axis group position during motion interruption
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Enable	valid	BOOL	TRUE/FALSE	FALSE	Enable command function
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Active	in call	BOOL	TRUE/FALSE	FALSE	Writing resume data

3-2-2-31. Axis group jog run [SMC_GroupJog]

(1) Instruction overview

Control the axis group to perform Jog motion in the specified coordinate system.

Instruction	Name	Graphical representation	ST language
SMC_GroupJog	axis group jog run		SMC_GroupJog(AxisGroup:= , Execute:= , CoordSystem:= , CoordSystemPCS:= , VelFactor:= , AccFactor:= , JerkFactor:= , AxisX:= , AxisY:= , AxisZ:= , AxisA:= , AxisB:= , AxisC:= , ABC_as_ACS:= , Busy=> , InitialPositionReached=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_REF_SM3	-	-	specified axis group
VAR_INPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Execute	valid	BOOL	TRUE/FALSE	FALSE	Execute current instruction
CoordSystem	reference coordinate system	SMC_COORD_SYSTEM	-	-	reference coordinate system
CoordSystemPCS	PCS coordinate system	SMC_COORD_SYSTEM	-	-	The internal coordinate system used for jogging is PCS, and SMC_SetDynCoordTransformEx is used when needed to make changes
VelFactor	speed factor	LREAL	0-1	1	Speed factor, the maximum speed of each axis multiplied by this speed factor, with a value between [0, 1]
AccFactor	acceleration factor	LREAL	0-1	1	Acceleration factor, the maximum velocity of each axis multiplied by this acceleration factor, with a value between [0,1]
JerkFactor	jerk factor	LREAL	0-1	1	Jerk factor, the maximum velocity of each axis multiplied by this jerk factor, with a value between [0,1]
AxisX	Axis X	IAxisRef	-	0	The X-axis in the coordinate system is set to 0 if it is not used
AxisY	Axis Y	IAxisRef	-	0	The Y-axis in the coordinate system is set to

					0 if it is not used
AxisZ	Axis Z	IAxisRef	-	0	The Z-axis in the coordinate system is set to 0 if it is not used
AxisA	Axis A	IAxisRef	-	0	Control the a-coordinate (rotation around the z-axis) or the axis of the first tool axis. If not applicable, set to 0

VAR_INPUT	Name	Data type	Effective range	Initial value	Description
AxisB	Axis B	IAxisRef	-	0	Control the b coordinate (rotation around the y axis) or the axis of the second tool axis. If not applicable, set to 0
AxisC	Axis C	IAxisRef	-	0	Control the c-coordinate (rotation around the z axis) or the axis of the third tool axis. If not applicable, set to 0
ABC_as_ACS	Start coordinate conversion	BOOL	TRUE/FALSE	FALSE	If true, the positions of AxisA, AxisB, and AxisC will be interpreted as the target position of the tool's kinematic axis, otherwise represented as the ZYZ direction. If CoordSystem is set to ACS, it will be ignored. If true, the kinematic transformation of the axis group must be Kin_ Coupled type and supports SMC_ORIENTATION_MODE.Axis
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
InitialPosition Reached	Received location	BOOL	TRUE/FALSE	FALSE	-
CommandAborted	Instruction interrupted	BOOL	TRUE/FALSE	FALSE	Module execution interrupted is true
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	error code

(3) Function description

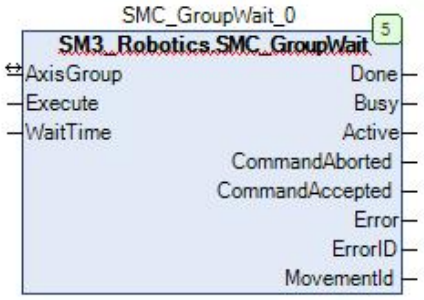
- When the axis group undergoes JOG motion (jog), it will move according to the position and direction provided by the configured axis.
- The command will convert the input coordinate system into the target coordinate system for output, such as JOG motion in a machine coordinate system using MCS.
- Set ABC_as_ACS parameters is using a hybrid transformation of the reference axis, where X/Y/Z is the position of the Cartesian coordinate system and A/B/C is the position of the tool coordinate system.
- When the product coordinate system is used for real-time changes in position, it can be achieved by configuring the input coordinate system CoordSystemPCS, and the real-time changes of the axis group in the coordinate system need to be marked.

- To restart this FB without changing CoordSystemPCS, it is necessary to add another move command, such as MC_GroupHalt. If this usage principle is not followed, the error SMC_ AXIS_ GROUP_ PCS_ STILL_ IN_USE will be returned.

3-2-2-32. Axis group wait [SMC_GroupWait]

(1) Instruction overview

Set the delay waiting for the axis group.

Instruction	Name	Graphical representation	ST language
SMC_GroupJog	axis group wait		SMC_GroupWait(AxisGroup:= , Execute:= , WaitTime:= , Done=> , Busy=> , Active=> , CommandAborted=> , CommandAccepted=> , Error=> , ErrorID=> , MovementId=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
AxisGroup	axis group	AXIS_GROUP_ REF_SM3	-	-	specified axis group
VAR_INPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Execute	valid	BOOL	TRUE/FALSE	FALSE	Execute current instruction
WaitTime	wait time	LREAL	0, positive number	0	The time to wait on the path, in seconds
VAR_OUTPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Done	completed	BOOL	TRUE/FALSE	FALSE	TRUE when instruction execution is completed
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Active	in control	BOOL	TRUE/FALSE	FALSE	Change to TRUE in control
CommandAborted	Instruction interrupted	BOOL	TRUE/FALSE	FALSE	Module execution interrupted is true
CommandAccepted	motion accepted	BOOL	TRUE/FALSE	FALSE	True when the module successfully calls the axis group
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	error code
MovementId	motion flag	SMC_Movement_Id	TRUE/FALSE	FALSE	True when the motion is being

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
					executed or completed

(3) Function description

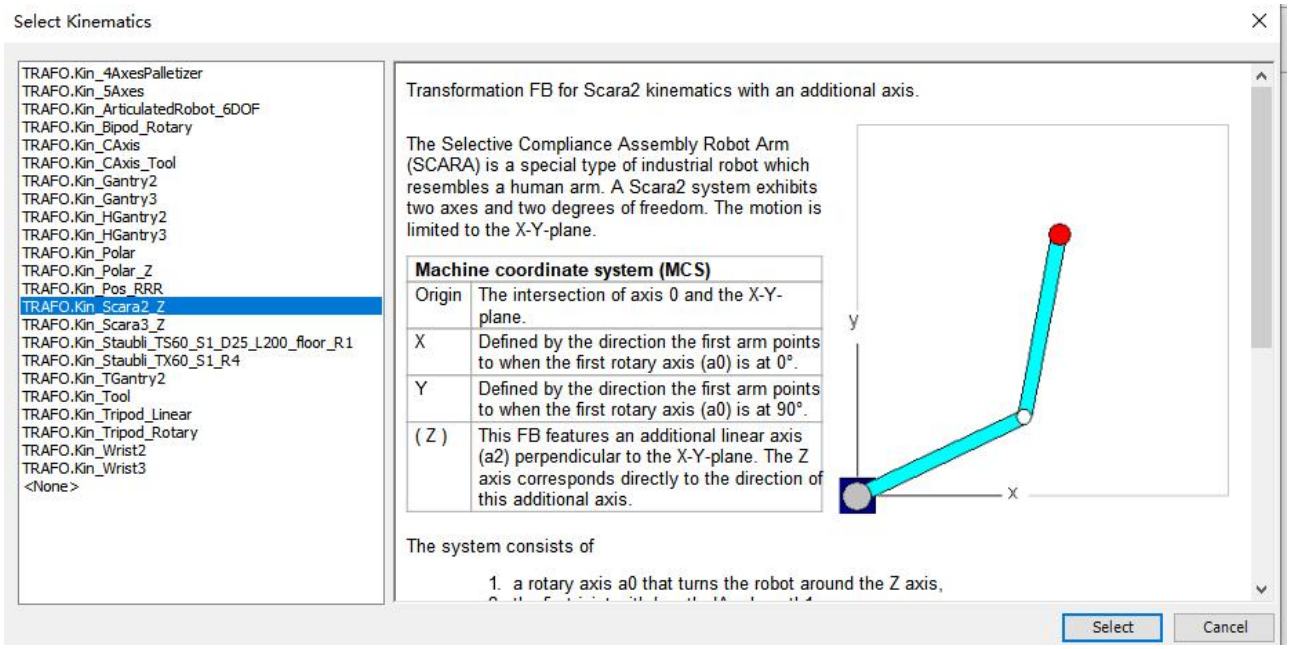
- If the waiting time between two adjacent instructions and the time configured for the task are the same, the actual waiting time between two movements through SMC_ GroupWait instruction has become fewer, for example, the execution of the next instruction usually starts from the next cycle of the task cycle, but if there is a SMC_GroupWait in the middle, then After the delay of the SMC_GroupWait instruction ends, it will immediately start the next motion instruction without waiting for the start of the next cycle of the task cycle.
- If there is a waiting command after tracking the movement, the axis group will track the endpoint of the previous movement within the specified time.
- If each axis is not in a Standstill state but is not controlled by the axis group and waits to be called, the axis group will report an error SMC_AXIS_GROUP_IDLE_WAIT_AXES_MOVING.
- Unlike updating every cycle of the task cycle, If the time of the SMC_ GroupWait instruction is different from a multiple of the bus cycle time, it can also immediately follow the next instruction after the waiting time, making subsequent movements smoother to start.
- Due to technical reasons, the waiting time can be increased by up to one cycle in the following situations: - non-tracking -> waiting -> tracking - tracking -> waiting -> non-tracking - tracking -> waiting -> PTP-tracking.

3-2-3. Axis group function application

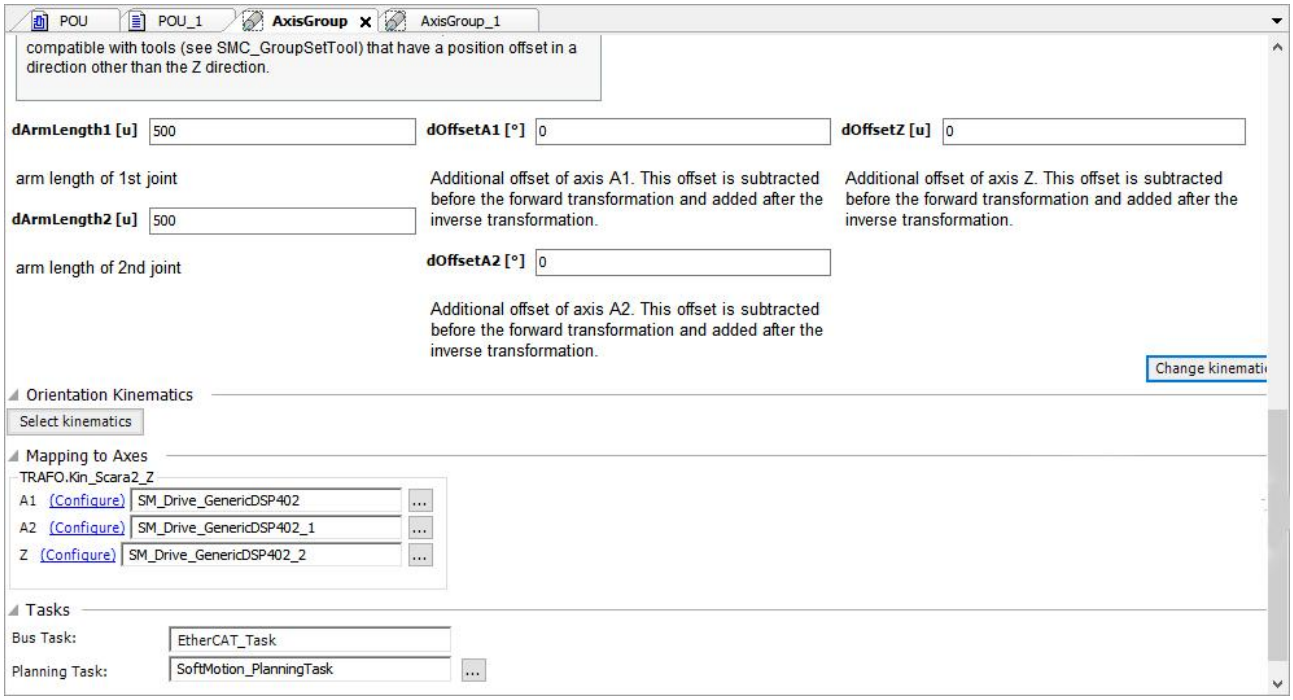
Example: This routine introduces the implementation of motion control for a SCARA system with two nodes.

Program operation:

(1) Right click on "Application" in the engineering equipment bar, select "Add Object" - "AxisGroup", name it AxisGroup, and open it. Select the SCARA kinematic model through the "Kinematic Model" interface.

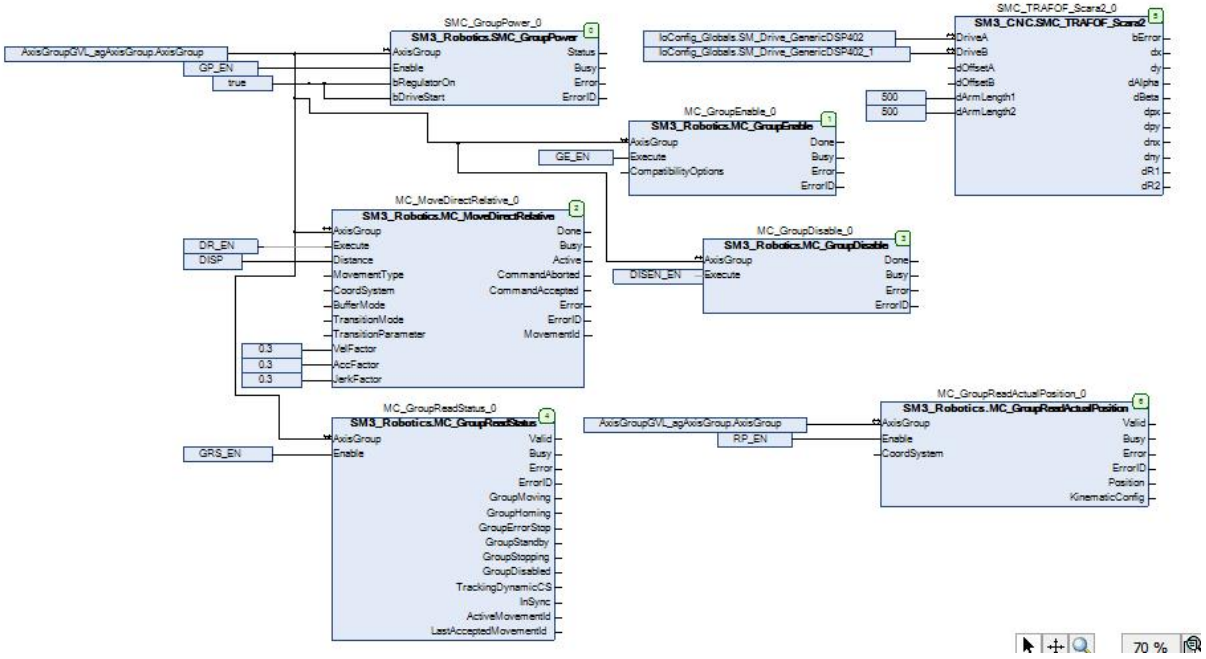


(2) Configure kinematic parameters: Set the length of the large and small arms to 500, and have zero angle offset. Axis parameter configuration, at mapping axis positions.



(3) Write kinematic instructions and configure motion parameters based on motion trajectory

```
PROGRAM POU
VAR
    SMC_GroupPower_0: SM3_Robotics.SMC_GroupPower;
    GP_EN: BOOL;//伺服使能 servo enable
    GE_EN: BOOL;//轴组启动 axis group startup
    MC_MoveDirectRelative_0: SM3_Robotics.MC_MoveDirectRelative;
    DR_EN: BOOL;//移动到相对位置 move to relative position
    DISP: SM3_Robotics.SMC_POS_REF;//相对位置设置 set relative position
    DIST: SM3_Robotics.SMC_POS_REF;
    MC_GroupEnable_0: SM3_Robotics.MC_GroupEnable;
    MC_GroupReadStatus_0: SM3_Robotics.MC_GroupReadStatus;
    GRS_EN: BOOL;//读取轴组状态 read axis group status
    Kin_HGantry2_0: TRAFO.Kin_HGantry2;
    SMC_TRAFOF_Gantry2_0: SM3_CNC.SMC_TRAFOF_Gantry2;
    SMC_TRAFOF_Scara2_0: SM3_CNC.SMC_TRAFOF_Scara2;
    MC_GroupReadActualPosition_0: SM3_Robotics.MC_GroupReadActualPosition;
    RP_EN: BOOL;//读取实时位置 read real-time position
    MC_GroupDisable_0: SM3_Robotics.MC_GroupDisable;
    DISEN_EN: BOOL;//轴组失能 axis group disable
END VAR
```



3-3. CAM function

3-3-1. CAM instruction list

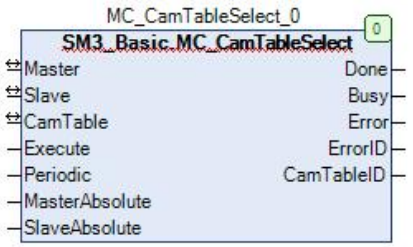
Instruction	Description
MC_CamTableSelect	Connect the selected CAM table to the actual cam table
MC_CamIn	Cam binding
MC_CamOut	Cam unbinding
MC_GearIn	Set the gear ratio of the main and slave axis and activate the electronic gears
MC_GearInPos	Set the synchronization distance and gear ratio of the master and slave axis and activate the electronic gear
MC_GearOut	Disconnect the main and slave axis electronic gears
MC_Phasing	Phase offset
SMC_CAMBounds	Spindle coupling slave axis
SMC_CAMBounds_Pos	Slave axis coupling spindle
SMC_CamEditor	Display cam table in visualization
SMC_CamRegister	Only read tappet information
SMC_GetCamSlaveSetPosition	Obtain cam table slave axis position
SMC_GetTappetValue	Evaluate output tappets
SMC_ReadCAM	Read the cam table
SMC_WriteCAM	Write the cam table

3-3-2. CAM instructions

3-3-2-1. Cam table designation [MC_CamTableSelect]

(1) Instruction overview

Used to select the cam table to be executed, which needs to be used together with MC_CamIn command.

Instruction	Name	Graphical representation	ST language
MC_CamTableSelect	cam table designation		MC_CamTableSelect(Master:= , Slave:= , CamTable:= , Execute:= , Periodic:= , MasterAbsolute:= , SlaveAbsolute:= , Done=> , Busy=> , Error=> , ErrorID=> , CamTableID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Master	Master axis	AXIS_REF	-	-	map to the master axis
Slave	slave axis	AXIS_REF	-	-	map to the slave axis
CamTable	cam table	MC_CAM_REF	-	-	map to cam table description
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Execute	valid	BOOL	TRUE/FALSE	FALSE	Rising edge signal, execute current command
Periodic	repetitive mode	BOOL	TRUE/FALSE	FALSE	TRUE: Execute the specified cam table periodically and repeatedly FALSE: Execute the cam table only once
MasterAbsolute	master axis absolute mode	BOOL	TRUE/FALSE	FALSE	TRUE: represents absolute coordinates FALSE: represents relative coordinates
SlaveAbsolute	slave axis absolute mode	BOOL	TRUE/FALSE	FALSE	TRUE: represents absolute coordinates FALSE: represents relative coordinates
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	completed	BOOL	TRUE/FALSE	FALSE	TRUE when instruction execution is completed
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Error recognition
CamTableID	effective CAMID	MC_CAM_ID	-	-	Select effective Cam_ID, used together with CamTableID in MC_CamIn

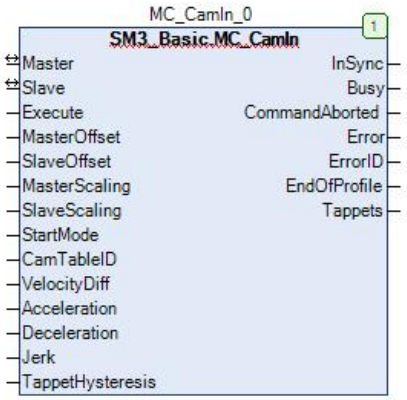
(3) Function description

- This instruction is implemented by the "SM3_Basic" library.
- This command is used to specify the cam table required for electronic cam operation, so before using this command, the cam table must be edited (by the cam editor or online editing).
- Execute the specified cam table at the rising edge of Execute, or refresh the specified cam table after updating the cam table.
- When the Done signal output is TRUE, the output variable "CamTableID" is generated and takes effect.
- The main and slave axis cannot be specified as the same axis, otherwise there will be an error output.

3-3-2-2. CAM binding [MC_CamIn]

(1) Instruction overview

Master slave axis binding, can set the engagement mode and related speed of the slave axis.

Instruction	Name	Graphical representation	ST language
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

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Master	master axis	AXIS_REF	-	-	map to master axis
Slave	slave axis	AXIS_REF	-	-	map to slave axis
VAR_INPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Execute	valid	BOOL	TRUE/FALSE	FALSE	Rising edge signal, execute current command
MasterOffset	master axis offset	LREAL	-	0	master table offset
SlaveOffset	slave axis offset	LREAL	-	0	slave table offset
MasterScaling	master axis scale	LREAL	-	1	Scale factor of the main configuration file
SlaveScaling	slave axis scale	LREAL	-	1	Scale factor for slave configuration files
StartMode	engagement method of slave axis	MC_StartMode	0-4	absolute	0: absolute position 1: relative position 2: ramp_in (ramp in)

					3: ramp_in_pos (positive ramp in) 4: ramp_in_neg (negative ramp im)
CamTableID	cam table	MC_CAM_ID	-	-	Define the using of cam table, output point of MC_CamTableSelect, used together with CamTableID
VelocityDiff	speed	LREAL	Negative number, 0, positive number	0	Maximum velocity of superimposed motion [u/s]
Acceleration	target acceleration	LREAL	Negative number, 0, positive number	0	acceleration [u/s^2]
Deceleration	target deceleration	LREAL	Negative number, 0, positive number	0	deceleration [u/s^2]
Jerk	target jerk speed	LREAL	Negative number, 0, positive number	0	jerk speed [u/s^3]
TappetHysteresis	tappet hysteresis	LREAL	-	-	tappet hysteresis

VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
InSync	cam take effect	BOOL	TRUE/FALSE	FALSE	TRUE indicates that the slave axis is synchronized with the spindle based on the cam table
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
CommandAborted	Instruction interrupted	BOOL	TRUE/FALSE	FALSE	Module execution interrupted is true
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Error recognition
EndOfProfile	curve completed	BOOL	TRUE/FALSE	FALSE	Pulse output: The cycle of cam contour ends
Tappets	tappet table	SMC_TappetData	-	-	for tappet signal processed by SMC_GetTappetValue

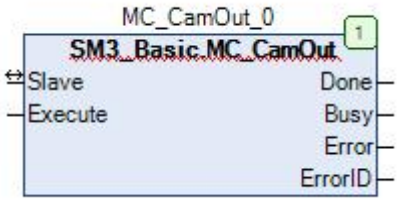
(3) Function description

- This instruction is implemented by the "SM3_Basic" library.
- This command can perform synchronous cam action as the cam table for the phase (spindle) and displacement (slave axis).
- There are two production methods for the cam table specified by this instruction:
  (1) Compile using a cam editor;
  (2) Build a cam table data structure through programming.
- In a cam system, to call a cam curve, first call MC_CamTableSelect command to select the corresponding cam table and then executes MC_CamIn. If you want to replace the cam curve, then call MC_CamTableSelect again to reselect the cam table.
- Need to use MC_CamOut command to release the cam coupling relationship between the master and slave axis.
- When the command is executed, the cam relationship between the slave axis and the spindle will be released when the slave axis of the command executes other motion commands, and Command-Aborted variable of MC_CamIn is output as TRUE.

3-3-2-3. Cam unbinding [MC_CamOut]

(1) Instruction overview

Release the cam coupling relationship between the specified slave axis and its corresponding spindle.

Instruction	Name	Graphical representation	ST language
MC_CamOut	release the cam action		MC_CamOut(Slave:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Slave	slave axis	AXIS_REF	-	-	map to the axis
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Execute	valid	BOOL	TRUE/FALSE	FALSE	Rising edge signal, execute current command
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	completed	BOOL	TRUE/FALSE	FALSE	TRUE when instruction execution is completed
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Error recognition

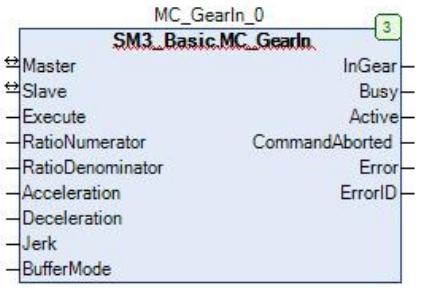
(3) Function description

- This instruction is implemented by the "SM3_Basic" library.
- This command is used to release the cam coupling relationship between the specified slave axis and its corresponding spindle.
- Release the slave axis cam coupling relationship at the rising edge of Execute.
- After the cam relationship is disconnected, the slave axis may not necessarily stop. If the speed of the slave shaft is not 0 before executing the command, then after the command DONE signal is completed, the cam coupling relationship is disconnected, but the slave axis still operates at the speed before cutting out.
- If the command is executed without a cam coupling relationship for the slave axis, an ERROR error will occur.

3-3-2-4. CAM action start [MC_GearIn]

(1) Instruction overview

Set the gear ratio of the master and slave axes and activate the electronic gears.

Instruction	Name	Graphical representation	ST language
MC_GearIn	cam action start		MC_GearIn(Master:= , Slave:= , Execute:= , RatioNumerator:= , RatioDenominator:= , Acceleration:= , Deceleration:= , Jerk:= , BufferMode:= , InGear=> , Busy=> , Active=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Master	master axis	AXIS_REF	-	-	map to master axis
Slave	slave axis	AXIS_REF	-	-	map to slave axis
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Execute	valid	BOOL	TRUE/FALSE	FALSE	Rising edge signal, execute current command
RatioNumerator	numerator of gear ratio	DINT	positive number	1	offset on the master table
RatioDenominator	denominator of gear ratio	UDINT	positive number	1	offset on the slave table
Acceleration	target acceleration	LREAL	positive number	0	acceleration [u/s^2]
Deceleration	target deceleration	LREAL	positive number	0	deceleration [u/s^2]
Jerk	target jerk speed	LREAL	positive number	0	jerk speed [u/s^3]
BufferMode	buffer mode	MC_BUFFER_MODE	-	-	-
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
InGear	gear ratio reached	BOOL	TRUE/FALSE	FALSE	The slave station moves at the given master station speed ratio, and the Target speed reached by the slave

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
					axis is TRUE
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Active	in control	BOOL	TRUE/FALSE	FALSE	Change to TRUE in control
CommandAborted	instruction interrupted	BOOL	TRUE/FALSE	FALSE	Module execution interrupted is true
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Error recognition

(3) Function description

- Start the cam action at the rising edge of Execute
- To decouple after executing the electronic gear, the GearOut command must be used.
- This command is a speed electronic gear function, and the loss of synchronization distance caused during the acceleration process will not be automatically compensated.
- When the Busy signal is TRUE during command execution, repeatedly triggering the rising edge of Execute will not restart the command regardless of whether the target speed of the slave axis has been reached.
- When reaching the target speed, InGear is TRUE, and at this time, the slave axis movement quantity=the spindle movement quantity * RatioNumerator/RatioDenominator.
- If the spindle speed changes in real time, please use this command with caution.

Note: Do not use MC_SetPosition during instruction execution to prevent accidents caused by rapid motor operation.

3-3-2-5. Position specified gear action [MC_GearInPos]

(1) Instruction overview

Set the electronic gear ratio between the main axis and the slave axis for electronic gear action.
Specify the spindle position and slave axis position of starting synchronization, and the starting synchronization distance of the spindle, and use this to complete the action of entering the electronic gear.

Instruction	Name	Graphical representation	ST language

MC_GearInPos	position specified gear action		MC_GearInPos(Master:= , Slave:= , Execute:= , RatioNumerator:= , RatioDenominator:= , MasterSyncPosition:= , SlaveSyncPosition:= , MasterStartDistance:= , BufferMode:= , AvoidReversal:= , StartSync=> , InSync=> , Busy=> , Active=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Master	master axis	AXIS_REF	-	-	map to master axis
Slave	slave axis	AXIS_REF	-	-	map to slave axis
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Execute	valid	BOOL	TRUE/FALSE	FALSE	Rising edge signal, execute current command
RatioNumerator	numerator of gear ratio	DINT	positive number	1	master axis offset
RatioDenominator	denominator of gear ratio	UDINT	positive number	1	slave axis offset
MasterSyncPosition	Spindle synchronization position	LREAL	Negative number, 0, positive number	0	Main position for axis synchronous operation
SlaveSyncPosition	slave axis synchronization position	LREAL	Negative number, 0, positive number	0	Slave position for axis synchronous operation
MasterStartDistance	Execute synchronous spindle position	LREAL	Negative number, 0, positive number	0	The main distance of the gears in the program (start the slave axis to enter synchronization)
BufferMode	buffer mode	MC_BUFFER_MODE	-	-	-
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
AvoidReversal	Prohibit reverse run	BOOL	TRUE/FALSE	FALSE	FALSE: Indicates that slave axis reversal is physically possible and acceptable. TRUE: Indicates that slave axis reversal is physically impossible,

					Otherwise, it may cause danger. It is only applicable to modular driven axis. If reverse rotation cannot be avoided, the axis will stop due to an error
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
StartSync	Start synchronization	BOOL	TRUE/FALSE	FALSE	If the electronic gear starts processing, it is true
InSync	Arrival synchronization	BOOL	TRUE/FALSE	FALSE	Electronic gear command completed
Busy	Executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Active	In control	BOOL	TRUE/FALSE	FALSE	Change to TRUE in control
CommandAborted	Instruction interrupted	BOOL	TRUE/FALSE	FALSE	Module execution interrupted is TRUE
Error	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	Error code	SMC_ERROR	-	0	Error recognition

(3) Function description

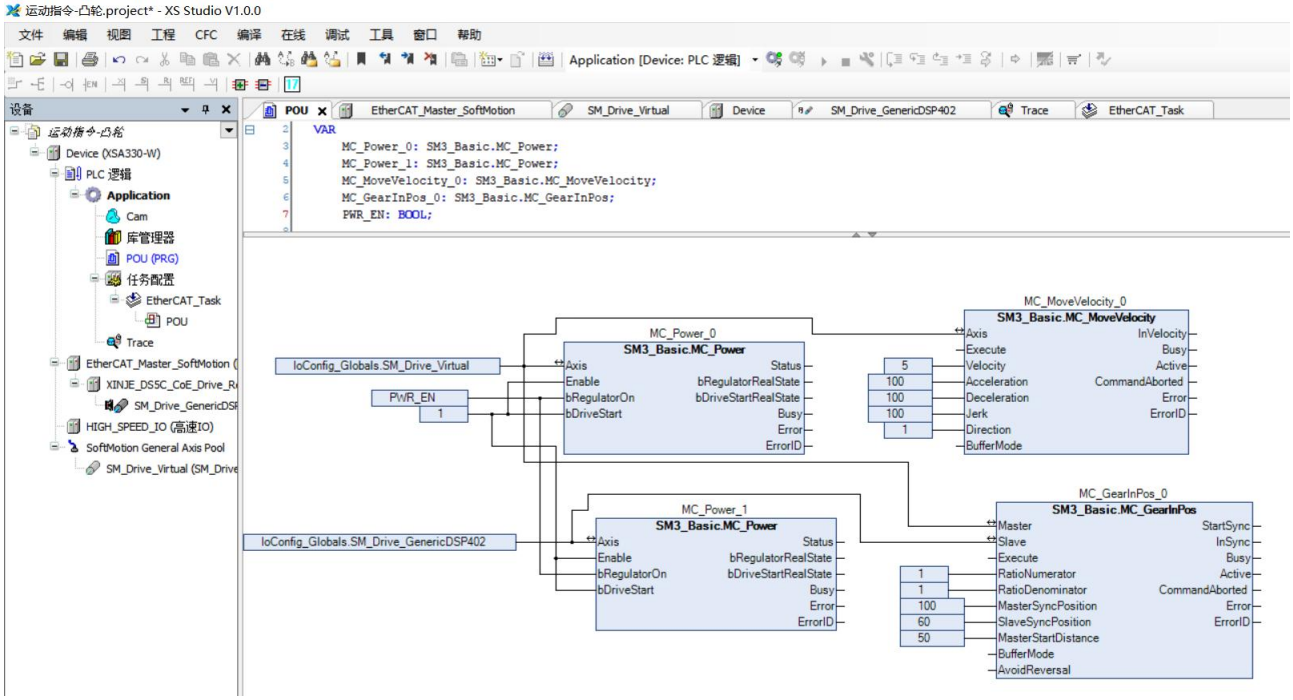
- This instruction is implemented by the "SM3_Basic" library.
- After starting the action, the slave axis target speed is the speed obtained by multiplying the spindle speed by the gear ratio, and the acceleration and deceleration actions are carried out.
- The process from the start of synchronization to the end of synchronization in this function block is essentially an electronic cam that slave axis follows the spindle within the synchronization interval. At this time, based on the spindle range (MasterSyncPosition-MasterStartDistance, MasterSyncPosition) and the slave axis range (current position, SlaveSyncPosition), the command will automatically design a cam curve based on the set gear ratio and the above three parameters, When performing synchronization, slave axis will follow the spindle to complete the cam movement.
- If the master-slave axis operates in linear mode, it is necessary to ensure that the above parameters are set reasonably, otherwise the gear action cannot be carried out correctly. Therefore, it is recommended to use this command when the master-slave axis is in modulus axis mode. For example, both the master and slave axis linear working modes move in a forward direction, if the master axis position MasterSyncPosition-MasterStartDistance or the slave axis position>SlaveSyncPosition, the electronic gear action cannot be cut in.
- At the same time as synchronization is completed (InSync is true), the target speed is reached, and then the slave axis movement quantity=the spindle movement quantity RatioNumerator/RatioDenominator
- For AvoidReversal: If the slave axis is a modal axis and the spindle speed (the multiple relationship of gear ratio) is not relative to the velocity relationship of the slave axis, then MC_GearInPos will try to avoid reversing of the slave axis. It attempts to "stretch" the motion of the slave axis by adding 5 slave axis cycles. If this stretching is invalid, an error will occur and slave axis will error stop. If the slave axis speed is associated with the spindle speed (which is a multiple of the gear ratio), an error will occur and the axis will error stop. If the slave axis is a linear axis mode axis, an error will occur at the rising edge of Execute.

(4) Application

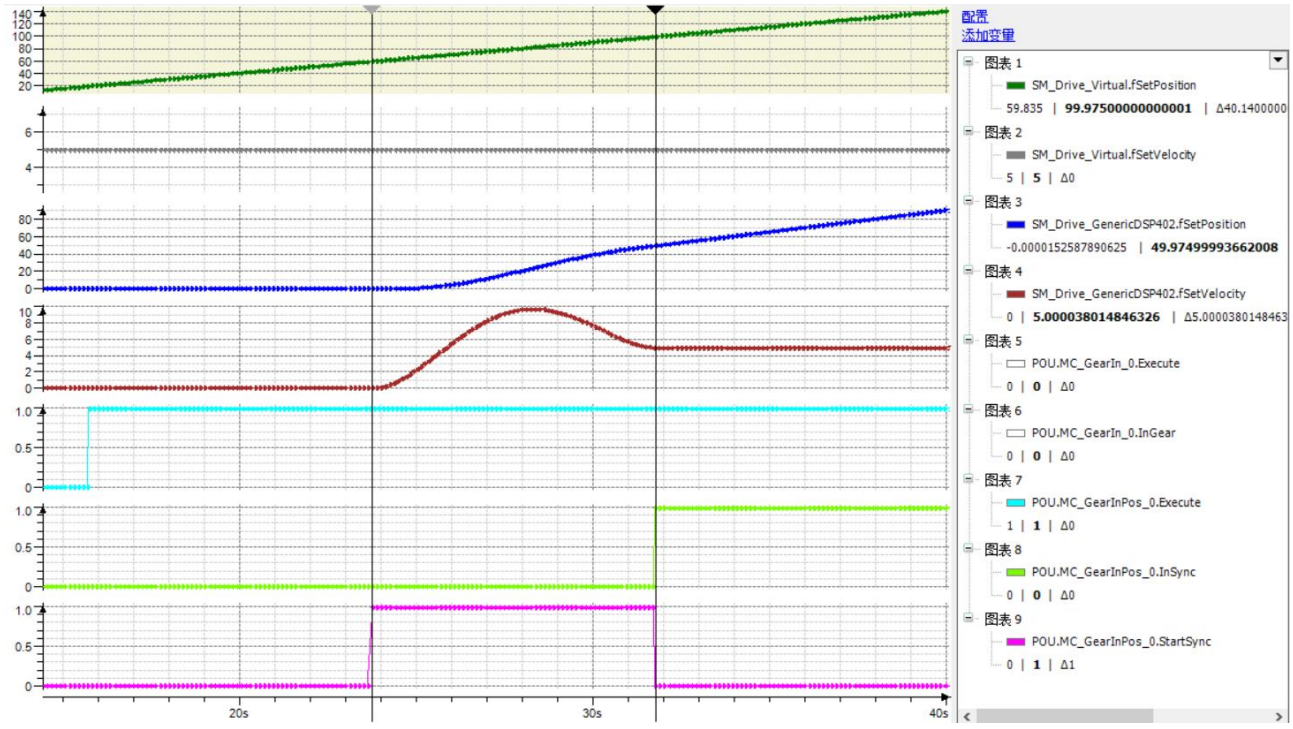
Example 1: Set the electronic gear ratio between the main axis and the slave axis to perform electronic gear

action. Specify the spindle position and slave axis position to start synchronization, and the starting synchronization distance of the spindle, and use this to complete the action of entering the electronic gear. Select the virtual axis for the main axis, select the Ethercat real axis from the slave axis, set the main axis speed to 5u/s, and the direction is positive.

(1) Programming: The starting positions of both axes are set to 0, the spindle is set to a speed of 5 u/s using the function block "MC_MoveVelocity", and the direction is positive. The electronic gear ratio of the two axes is set to 1:1 using "MC_GearInPos", the spindle synchronization position is 100u, the slave axis synchronization position is 50u, and the spindle starting synchronization distance is 40u.



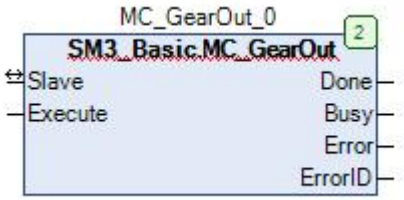
(2) You can use 'Trace' to view the current status. As can be seen from the cursor, synchronization starts at 60u for the spindle position. After synchronization is completed, the spindle position is at 100u, the slave axis position is at 50u, and the slave axis speed is at 5u/s.



3-3-2-6. Gear action release [MC_GearOut]

(1) Instruction overview

Disconnect the electronic gear coupling between the slave axis and spindle.

Instruction	Name	Graphical representation	ST language
MC_GearOut	gear action release		MC_GearOut(Slave:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Slave	slave axis	AXIS_REF	-	-	map to the axis
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Execute	valid	BOOL	TRUE/FALSE	FALSE	Rising edge signal, execute current command
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Done	completed	BOOL	TRUE/FALSE	FALSE	If the coupling between the slave axis and the spindle electronic gear is disconnected, it is true
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Error recognition

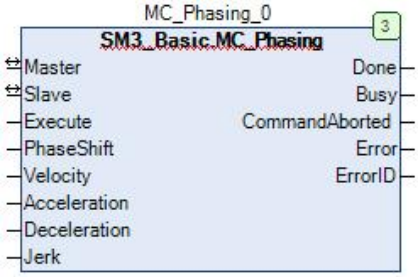
(3) Function description

- This instruction is implemented by the "SM3_Basic" library.
- After cutting out the electronic gear, the speed from the slave axis is the speed before cutting out, so it is necessary to cooperate with MC_Stop command stops the slave axis.
- Specify the action object axis through Slave, specify Deceleration, and abort the executing command MC_GearIn (gear action start), MC_ GearInPos (position specified gear action) command.
- This instruction has no effect for the master axis action of command MC_GearIn and MC_GearInPos.

3-3-2-7. Master slave axis phase offset [MC_Phasing]

(1) Instruction overview

Specify the phase deviation between the master and slave axes.

Instruction	Name	Graphical representation	ST language
MC_Phasing	master slave axis phase offset		MC_Phasing(Master:= , Slave:= , Execute:= , PhaseShift:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Master	master axis	AXIS_REF	-	-	map to master axis
Slave	slave axis	AXIS_REF	-	-	map to the axis
VAR_INPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Execute	valid	BOOL	TRUE/FALSE	FALSE	Rising edge signal, execute current command
PhaseShift	Phase compensation	LREAL	Negative number, 0, positive number	0	Specify the phase compensation amount for the spindle
Velocity	speed	LREAL	0, positive number	0	Maximum speed value when performing phase shift [command unit/s]
Acceleration	acceleration	LREAL	0, positive number	0	Maximum acceleration during phase shift execution [command unit /s^2]
Deceleration	deceleration	LREAL	0, positive number	0	Maximum deceleration during phase shift execution [command unit /s^2]
Jerk	jerk speed	LREAL	0, positive number	0	Maximum jerk speed during phase shift execution [command unit /s^3]
VAR_OUTPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Done	completed	BOOL	TRUE/FALSE	FALSE	If the electronic gear coupling between the slave axis and the spindle is disconnected, it is true
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended

CommandAborted	instruction interrupted	BOOL	TRUE/FALSE	FALSE	Module execution interrupted is true
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Error recognition

(3) Function description

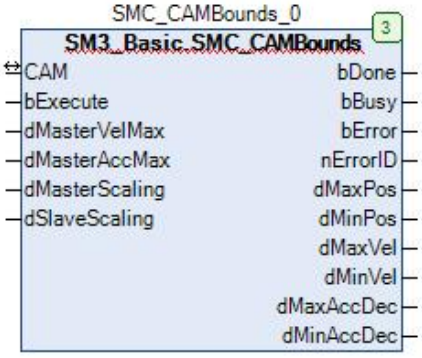
- Perform phase shift at the rising edge of Execute, slave axis automatically calculates a smooth curve to complete the phase shift from the slave axis to the main axis. The phase difference between the main and slave axes is the PhaseShift value of the input signal, and a positive value is the lag of the slave axis to the main axis.
- After completing the offset, the Done signal output is true.
- Compensate for the phase difference between the master and slave axes based on the set PhaseShift, Velocity, Acceleration, and Deceleration.
- When the phase difference between the master and slave axes reaches PhaseShift, the Done signal is output.
- When executing the command, the spindle command position and feedback position remain unchanged, and the slave axis is adjusted. After completion, the phase difference between the slave axis and the spindle is PhaseShift. The final result of this command is the phase offset between the given values of the axis, so the actual feedback value of the real axis may not be consistent with the final offset.

3-3-2-8. CAM range [SMC_CAMBounds]

(1) Instruction overview

When the slave axis is coupled with the spindle cam, the maximum position, speed, and acceleration of the slave axis can be calculated using this function block.

The spindle moves under maximum input speed and acceleration/deceleration restrictions. This command can check whether the curve is correct when designing the cam table, provided that the maximum acceleration, deceleration, speed, etc. of the spindle are known.

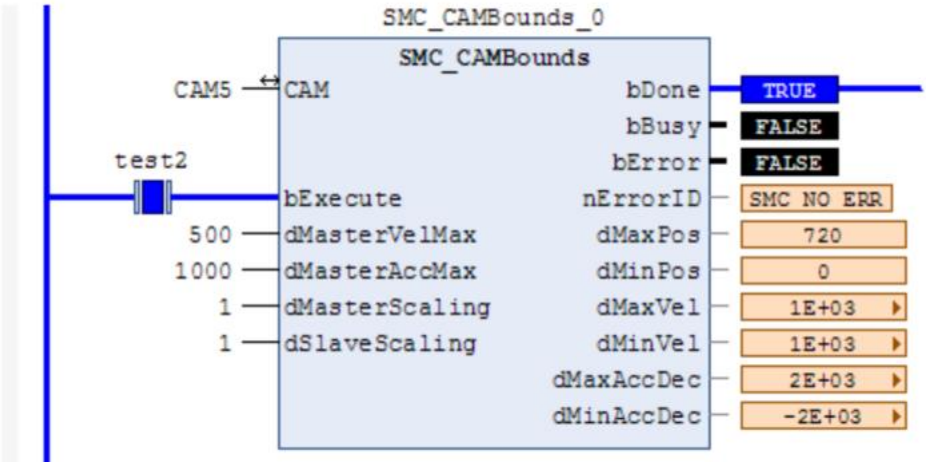
Instruction	Name	Graphical representation	ST language
SMC_CAMBounds	CAM range		SMC_CAMBounds(CAM:= , bExecute:= , dMasterVelMax:= , dMasterAccMax:= , dMasterScaling:= , dSlaveScaling:= , bDone=> , bBusy=> , bError=> , nErrorID=> , dMaxPos=> , dMinPos=> , dMaxVel=> , dMinVel=> , dMaxAccDec=> , dMinAccDec=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Cam	CAM	MC_CAM_REF	-	-	map to the cam
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
bExecute	valid	BOOL	TRUE/FALSE	FALSE	Rising edge signal, execute current command
dMasterVelMax	max speed	LREAL	-	1	Maximum spindle speed in absolute mode
dMasterAccMax	max acceleration	LREAL	-	0	Maximum spindle acceleration in absolute mode
dMasterScaling	Scale Factor	LREAL	-	1	Scale factor for spindle cam application
dSlaveScaling	Scale Factor	LREAL	-	1	Scale factor for slave cam application
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
bDone	Completed	BOOL	TRUE/FALSE	FALSE	If the coupling between the slave axis and the spindle electronic gear is disconnected, it is true
bBusy	Executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
bError	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
nErrorID	Error code	SMC_ERROR	-	0	Error recognition
dMaxPos	Max position	LREAL	-	0	Calculate the maximum position of the slave axis based on the cam table
dMinPos	Min position	LREAL	-	0	Calculate the minimum position of the slave axis based on the cam table
dMaxVel	Max speed	LREAL	-	0	Calculate the maximum speed of the slave axis
dMinVel	Min speed	LREAL	-	0	Calculate the minimum speed of the slave axis
dMaxAccDec	Max acceleration	LREAL	-	0	Calculate the maximum acceleration of the slave axis
dMinAccDec	Min acceleration	LREAL	-	0	Calculate the minimum acceleration of the slave axis

(3) Function description

- Based on the rising edge of bExecute, the "dMasterVelMax", "dMasterAccMax", "dMasterScaling", and "dSlaveScaling" values of the input variables are integrated with the cam table data to calculate the equivalent value of the "maximum position" and "minimum position" of the slave axis. For example, if the spindle cycle is 360 and the cam table is a straight line with a slope of 2, the calculated result is shown in the following figure:



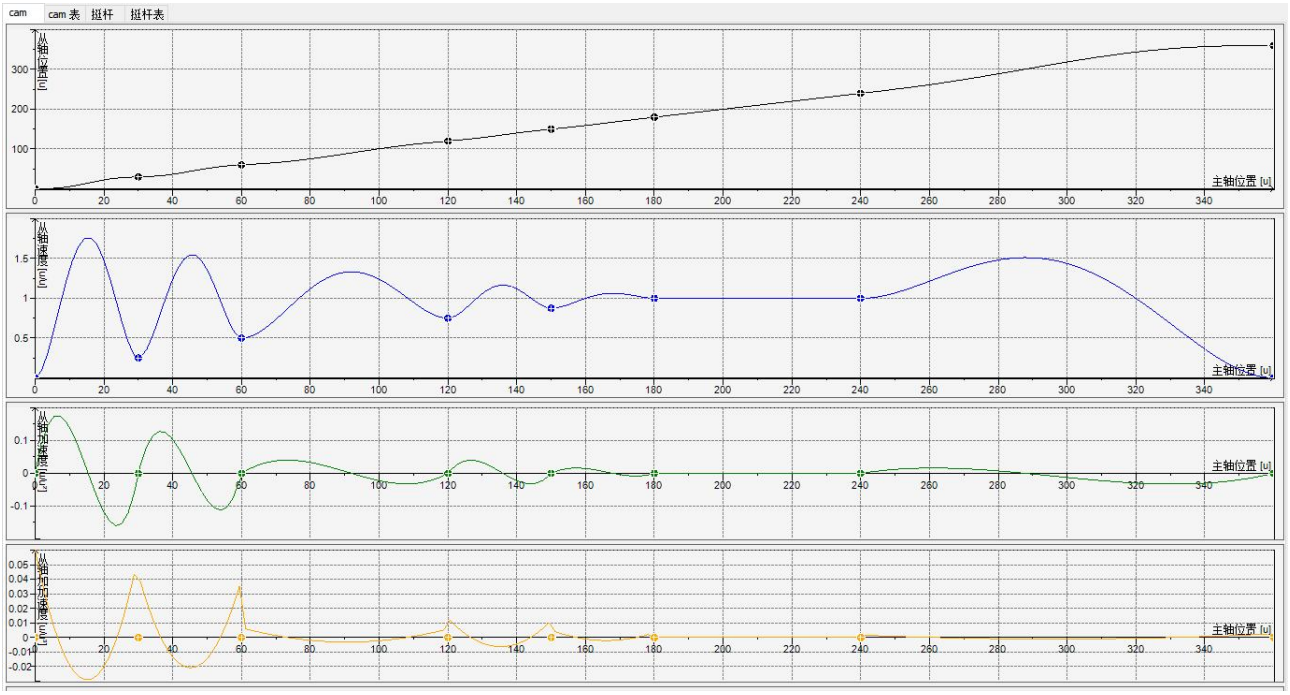
- The spindle can be calculated using this command when running in absolute mode or when the spindle is set to cycle mode, and the modulus is set to spindle cycle.
- The cam table is XYVA (valid in polynomial mode), and one-dimensional arrays, two-dimensional arrays, etc. are invalid.

(4) Application

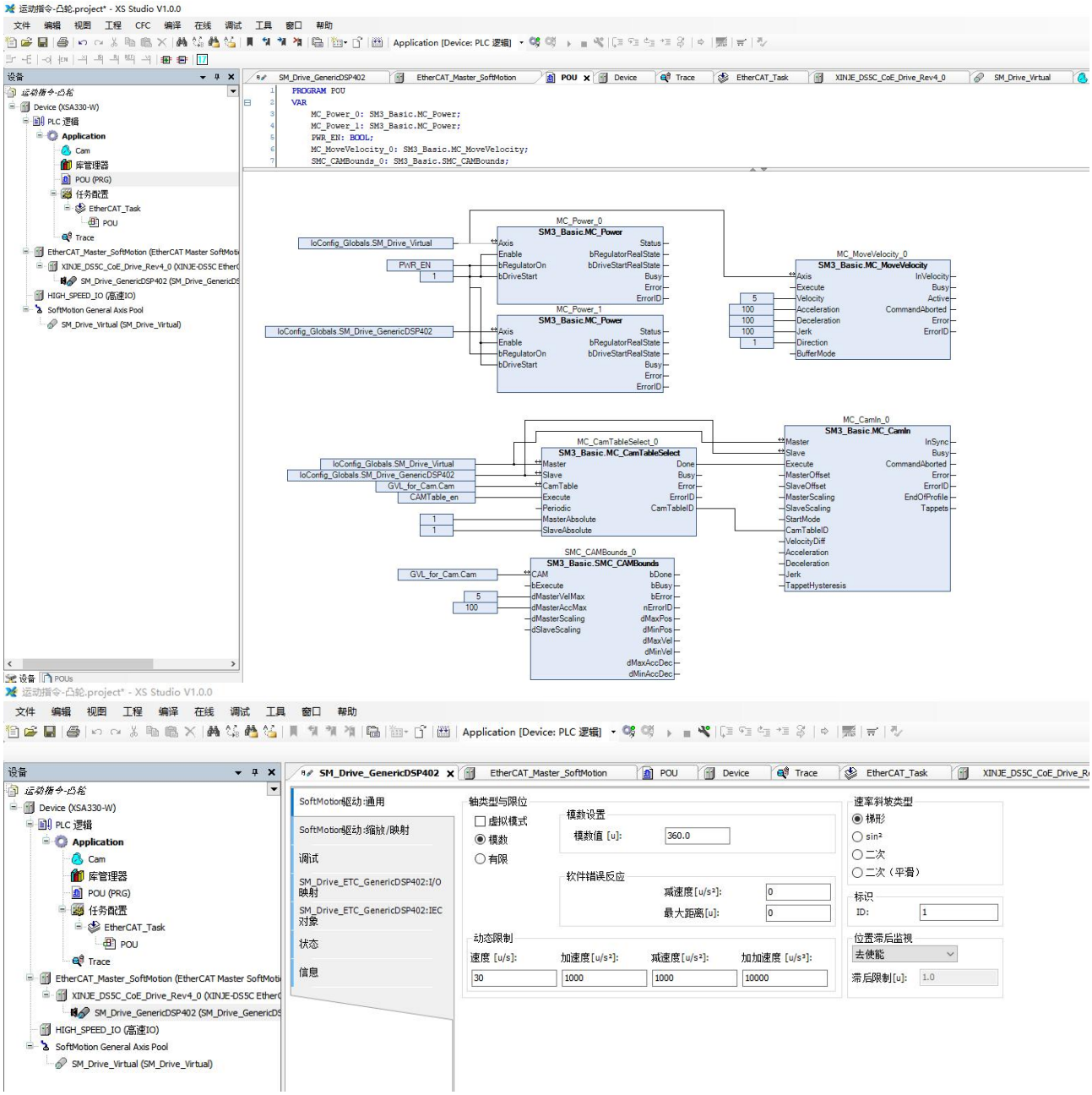
Example: Establish an electronic cam table, and calculate the maximum position, speed, acceleration/deceleration of the slave axis through "SMC_CAMBounds" after coupling the slave axis and the spindle cam. Select the virtual axis for the main axis and the Ethercat real axis for the slave axis.

Program:

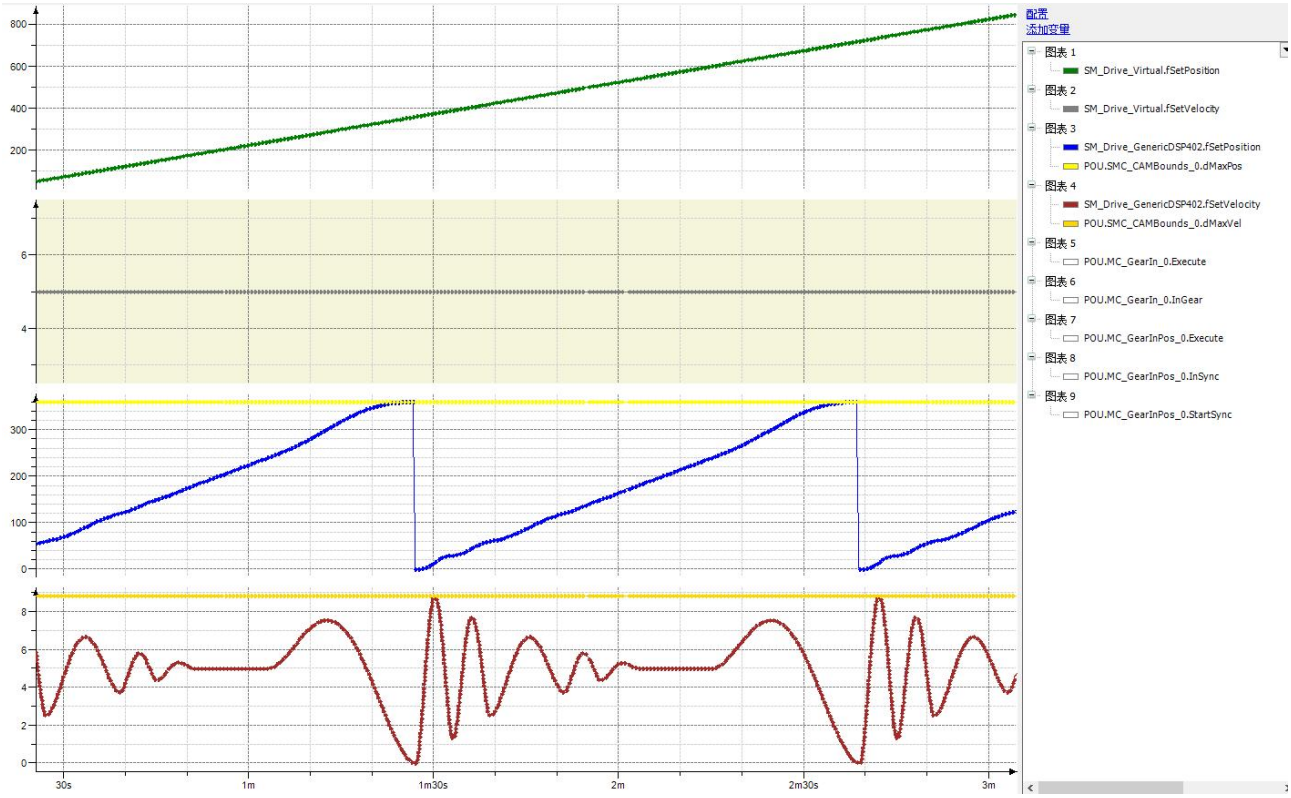
Right click on "Application" in the project device bar, select "Add Object" - "Cam Table", name it Cam, and open it to add the corresponding cam table parameters as shown in the following figure.



Select the cam table by instruction MC_CamTableSelect, select the coupling mode as absolute, and configure the electronic cam module MC_CamIn. Set the spindle running speed to 5 and acceleration to 100 using "MC-MoveVelocity", and calculate the maximum position, speed, and acceleration of the slave station using "SMC_CAMBounds". The master station runs synchronously with the master slave bounding, and sets the slave axis running mode to modulus - running between 0 and 360.



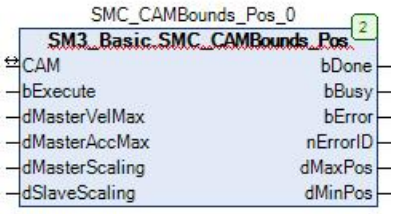
By monitoring the position and speed of the slave axis through "Trace", it can be seen that the maximum position and speed matched during operation.


800
600
400
200
6
4
300
200
100
0
8
6
4
2
0
30s
1m
1m30s
2m
2m30s
3m
配置
添加变量
图表 1
SM_Drive_Virtual.fSetPosition
图表 2
SM_Drive_Virtual.fSetVelocity
图表 3
SM_Drive_GenericDSP402.fSetPosition
POU.SMC_CAMBounds_0.dMaxPos
图表 4
SM_Drive_GenericDSP402.fSetVelocity
POU.SMC_CAMBounds_0.dMaxVel
图表 5
POU.MC_GearIn_0.Execute
图表 6
POU.MC_GearIn_0.InGear
图表 7
POU.MC_GearInPos_0.Execute
图表 8
POU.MC_GearInPos_0.InSync
图表 9
POU.MC_GearInPos_0.StartSync

3-3-2-9. CAM position range [SMC_CAMBounds_Pos]

(1) Instruction overview

When the slave axis is coupled with the spindle cam, the maximum and minimum positions of the slave axis can be calculated using this function block. Compare to SMC_CAMBounds, this function block lacks calculations such as maximum acceleration, and all other work is consistent.

Instruction	Name	Graphical representation	ST language
SMC_CAMBounds_Pos	CAM position range		SMC_CAMBounds_Pos(CAM:= , bExecute:= , dMasterVelMax:= , dMasterAccMax:= , dMasterScaling:= , dSlaveScaling:= , bDone=> , bBusy=> , bError=> , nErrorID=> , dMaxPos=> , dMinPos=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Cam	CAM	MC_CAM_REF	-	-	map to the cam
VAR_INPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
bExecute	Valid	BOOL	TRUE/FALSE	FALSE	Rising edge signal, execute current command
dMasterVelMax	Max speed	LREAL	-	1	Maximum spindle speed in absolute mode
dMasterAccMax	Max acceleration	LREAL	-	0	Maximum spindle acceleration in absolute mode
dMasterScaling	Scale Factor	LREAL	-	1	Scale factor for spindle cam application
dSlaveScaling	Scale Factor	LREAL	-	1	Scale factor for slave axis cam application
VAR_OUTPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
bDone	Completed	BOOL	TRUE/FALSE	FALSE	If the coupling between the slave axis and the spindle electronic gear is disconnected, it is true
bBusy	Executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
bError	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
nErrorID	Error code	SMC_ERROR	-	0	Error recognition
dMaxPos	Max position	LREAL	-	0	Calculate the maximum position of the slave axis based on the cam table
dMinPos	Min position	LREAL	-	0	Calculate the minimum position of the slave axis based on the cam table

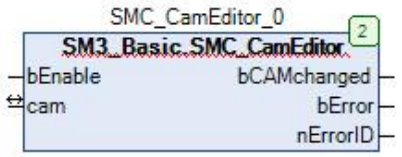
(3) Function description

- Based on the rising edge of bExecute, the "dMasterVelMax", "dMasterAccMax", "dMasterScaling", and "dSlaveScaling" values of the input variables are combined with the cam table data to calculate the "maximum position" and "minimum position" of the slave axis.
- The spindle can be calculated using this command when running in absolute mode or when the spindle is set to cycle mode, and the modulus is set to spindle cycle.
- The cam table is XYVA (valid in polynomial mode), and one-dimensional arrays, two-dimensional arrays, etc. are invalid.

3-3-2-10. Display cam table in visualization [SMC_CamEditor]

(1) Instruction overview

This function block is used to display cam table in visualization. This feature block does not work without TargetVideo or WebVideo.

Instruction	Name	Graphical representation	ST language
SMC_CamEditor	display cam table in visualization		SMC_CamEditor(bEnable:= , bCAMchanged=> , bError=> , nErrorID=> , cam:=);

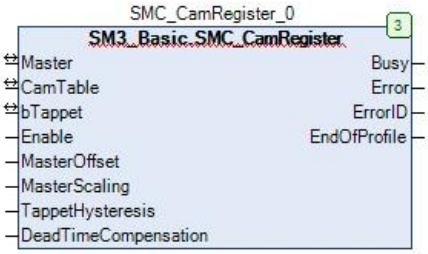
(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Cam	CAM	MC_CAM_REF	-	-	map to the cam
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
bEnable	valid	BOOL	TRUE/FALSE	FALSE	Modify the cam table when it is true
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
bCAMchanged	Completed	BOOL	TRUE/FALSE	FALSE	The cam in the visualization has changed. Then it is true
bError	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
nErrorID	Error code	SMC_ERROR	-	0	Error recognition

3-3-2-11. CAM tappet control [SMC_CamRegister]

(1) Instruction overview

Implement cam tappet control (cam switch). When editing the cam, it is possible not to edit the curves of the master and slave axes, and simply configure the tappet table to achieve tappet control through this function block.

Instruction	Name	Graphical representation	ST language
SMC_CamRegister	CAM tappet control		SMC_CamRegister(Master:= , CamTable:= , bTappet:= , Enable:= , MasterOffset:= , MasterScaling:= , TappetHysteresis:= , DeadTimeCompensation:= , Busy=> , Error=> , ErrorID=> , EndOfProfile=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Master	Master axis	AXIS_REF	-	-	map to the master axis
CamTable	CAM table	MC_CAM_REF	-	-	Mapping to an electronic cam, i.e. an electronic cam instance
bTappet	Tappet output	ARRAY [1..MAX_NUM_TAPPETS]OF BOOL	-	-	tappet point output
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
Enable	Executing	BOOL	TRUE/FALSE	FALSE	Execute current instruction is TRUE
MasterOffset	master axis position	LREAL	-	0	Offset on main table
MasterScaling	master axis scale	LREAL	-	1	Scale factor of the main configuration file
TappetHysteresis	tappet hysteresis	LREAL	-	-	tappet hysteresis size
DeadTimeCompensation	Dead time compensation	LREAL	-	0	Deadband time compensation in seconds. The expected spindle position will be calculated using linear extrapolation method
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
Busy	Executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
Error	Error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	Error code	SMC_ERROR	-	0	Error recognition

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
EndOfProfile	Curve cycle completion	BOOL	TRUE/FALSE	FALSE	True if the spindle position is greater than or equal to the set cycle

(3) Function description

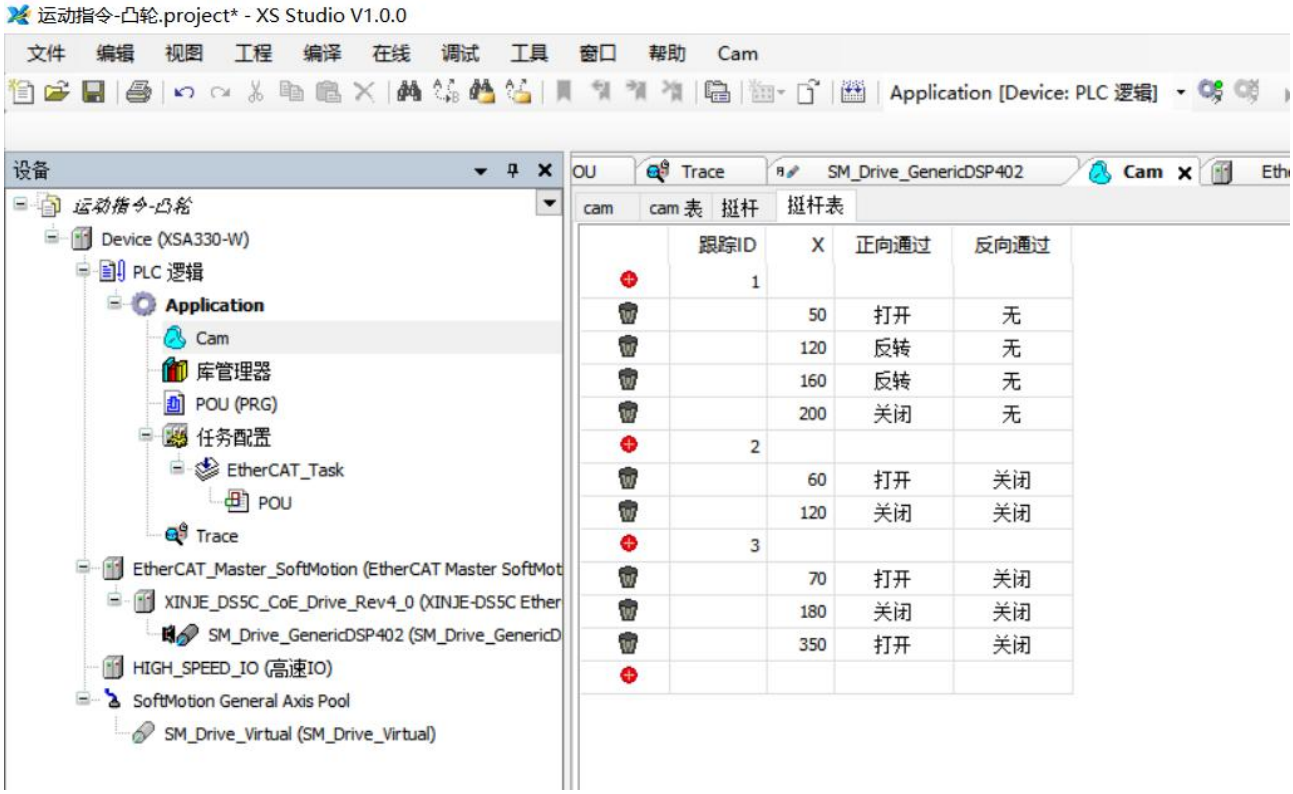
- Enable signal is TRUE, and if there is no error output, Busy output is TRUE to execute tappet control.
- The control function block is not related to the slave axis in the electronic cam, only the spindle cycle and tappet table need to be configured.
- bTappet is a one-dimensional Boolean structure (MAX_NUM_TAPPETS=512), and bTappet[i] corresponds to ith tappet output DeadTimeCompensation, the unit is second. When set to a positive value, it is ahead the tappet signal, and when set to a negative value, it is behind the tappet signal.

(4) Application

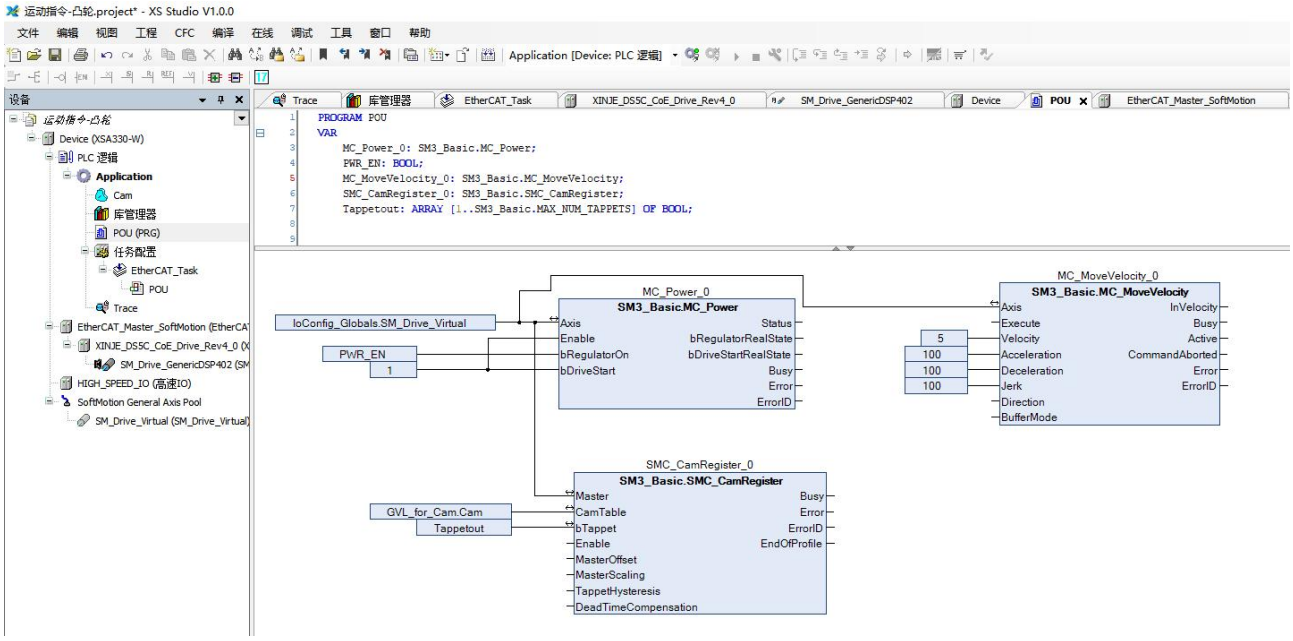
Example: Set 3 tappets in the tappet table in the cam table. Set the spindle speed to 5 and the acceleration to 100, and run the spindle through the trajectory set in the cam table. The main axis is selected as the virtual axis, and the slave axis is selected as the real axis.

Programming

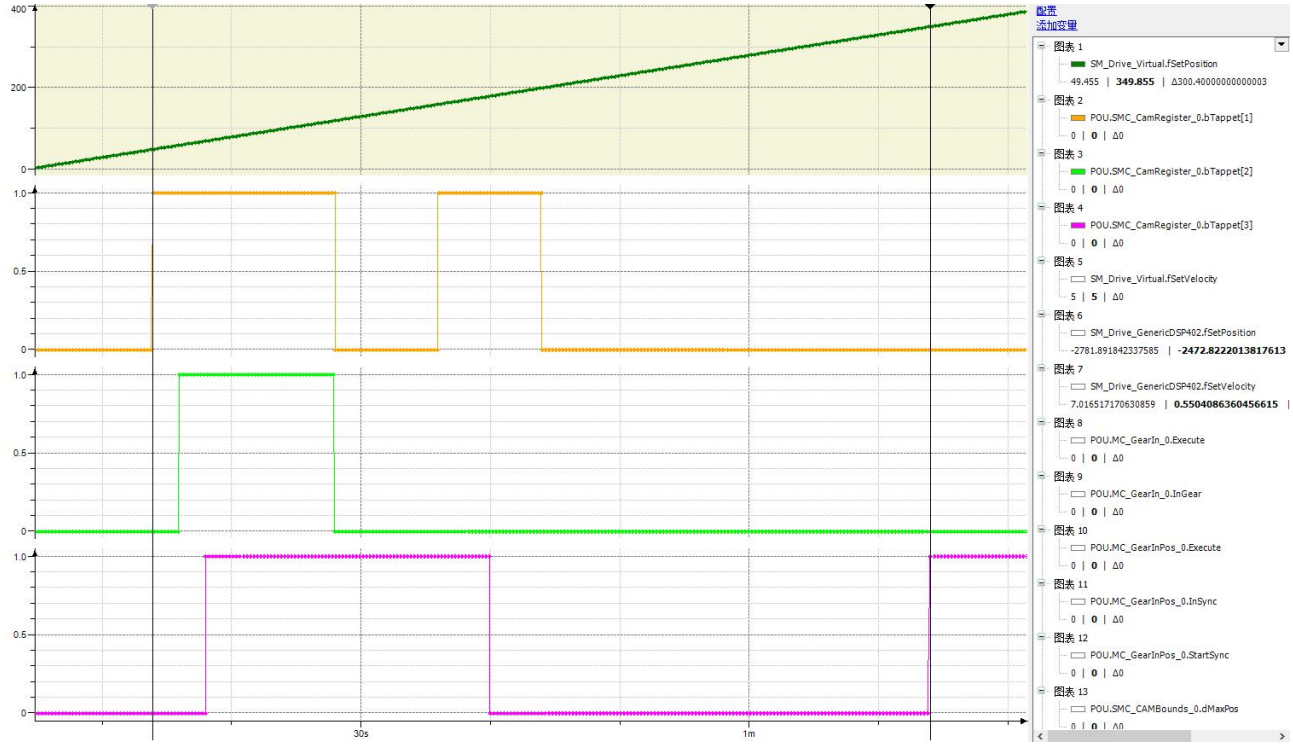
Right click on "Application" in the project device bar, select "Add Object" - "Cam Table", name it Cam, and open it to add the corresponding tappet parameters as shown in the following figure.



Add enabling module MC_ Power in the main program, spindle running speed module MC_MoveVelocity. By using the function block "SMC_CamRegister", the output status of all IDs in the tappet table can be monitored.



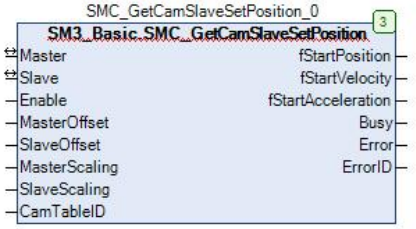
You can see all the current tappet output states of the spindle in 'Trace'.



3-3-2-12. Get cam slave axis position [SMC_GetCamSlaveSetPosition]

(1) Instruction overview

Read the cam table slave axis position, speed, and acceleration information.

Instruction	Name	Graphical representation	ST language
SMC_GetCamSlaveSetPosition	obtain the cam slave axis position		SMC_GetCamSlaveSetPosition(Master:= , Slave:= , Enable:= , MasterOffset:= , SlaveOffset:= , MasterScaling:= , SlaveScaling:= , CamTableID:= , fStartPosition=> , fStartVelocity=> , fStartAcceleration=> , Busy=> , Error=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Master	master axis	AXIS_REF	-	-	map to the master axis
Slave	slave axis	AXIS_REF	-	-	map to the slave axis
VAR_INPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
Execute	valid	BOOL	TRUE/FALSE	FALSE	execute current command at the rising edge signal
MasterOffset	master axis offset	LREAL	-	0	master table offset
SlaveOffset	slave axis offset	LREAL	-	0	slave table offset
MasterScaling	master scale	LREAL	-	1	Scale factor of the main configuration file
SlaveScaling	slave axis	LREAL	-	1	Scale factor of the slave configuration file
CamTableID	cam table	MC_CAM_ID	-	-	define the using of cam, use together with MC_CamTable Select output point CamTableI
VAR_OUTPUT	**Name**	**Data type**	**Effective range**	**Initial value**	**Description**
fStartPosition	slave axis position	LREAL	-	0	The position of the slave axis obtained from the cam table and the current spindle information
fStartVelocity	slave axis speed	LREAL	-	0	The slave axis speed obtained from the cam table and the current spindle information
fStartAcceleration	slave acceleration	LREAL	-	0	The slave axis acceleration obtained from the cam table and the current spindle information
Busy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
					block has not yet ended
Error	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
ErrorID	error code	SMC_ERROR	-	0	Error recognition

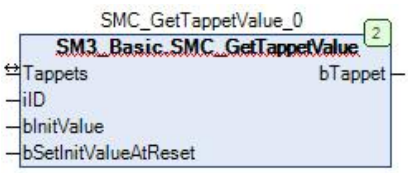
(3) Function description

- The output value calculated by this instruction is: Y= (cam((Cam starting spindle lookup position +Masteroffset)* MasterScaling)+slaveoffset)* SlaveScaling, Cam is a function of the cam table. For example, if the starting spindle position of the cam is 0, the master-slave axis is scaled to 1, the Masteroffset is 100, and the slaveoffset is 0, the function block outputs the position of the slave axis corresponding to the cam table at 100.
- This function block only requires the successful construction of the cam table to read the position of the slave axis, and there is no requirement for the operation of the master slave axis.

3-3-2-13. Obtain tappet output value [SMC_GetTappetValue]

(1) Instruction overview

To read the current tappet status, it is necessary to use with MC_CamIn command.

Instruction	Name	Graphical representation	ST language
SMC_GetTappetValue	Obtain the tappet output value		SMC_GetTappetValue(Tappets:= , iID:= , bInitValue:= , bSetInitValueAtReset:= , bTappet=>);

(2) Related variables

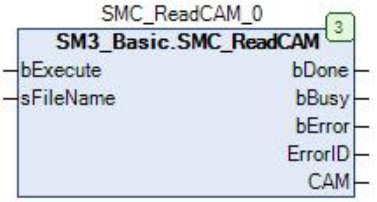
VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
Tappet	tappet	SMC_Tappet Data	-	-	map to a tappet
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
iID	tappet ID	INT	positive value, 0	0	tappet ID
bInitValue	initial value	BOOL	TRUE/FALSE	FALSE	The initial value of the tappet to be allocated on the first call
bSetInitValueAtReset	tappet reset	BOOL	TRUE/FALSE	FALSE	If it is true, the output value of the tappet will be set to the initial value when the CamIn function block restarts. If it is FALSE, the value of the tappet will be maintained when the CamIn function block is restarted
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
bTappet	tappet value	BOOL	TRUE/FALSE	FALSE	tappet value

(3) Function description

- This function block needs to use together with MC_CamIn.

3-3-2-14. Read the cam table [SMC_ReadCAM]

(1) Instruction overview

Instruction	Name	Graphical representation	ST language
SMC_ReadCAM	read the cam table		SMC_ReadCAM(bExecute:= , sFileName:= , bDone=> , bBusy=> , bError=> , ErrorID=> , CAM=>);

(2) Related variables

VAR_INPUT	Name	Data type	Effective range	Initial value	Description
bExecute	execute	BOOL	TRUE/FALSE	FALSE	execute the function block at the rising edge
sFileName	file name	STRING(255)	-	-	The name of the file, which contains a cam description that defines ASCII format
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
bDone	completed	BOOL	TRUE/FALSE	FALSE	True if Cam has been read in
bBusy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
bError	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
nErrorID	error code	SMC_ERROR	-	0	Error recognition
CAM	-	MC_CAM_REF	STRUCT(nElements : = 0, nTappets : = 0)	-	-

(3) Function description

- This function block is designed for loading cams during runtime, and make it available for modules MC_CamTableSelect and MC_CamIn. The cam to be loaded must be created before the cam editor, and saved in "*.CAM" file (refer to SMC_WriteCAM).
- This function block needs to use together with MC_CamIn.
- The size of the loadable cam is limited by the global constant gc_SMC_FILE_MAXCAMEL (Number of Elements) and gc_SMC_FILE_ MAXCAMTAP (number of cam switch actions).

3-3-2-15. Modify the cam table [SMC_WriteCAM]

(1) Instruction overview

Used to store the cam table created in the cam editor in a file at runtime.

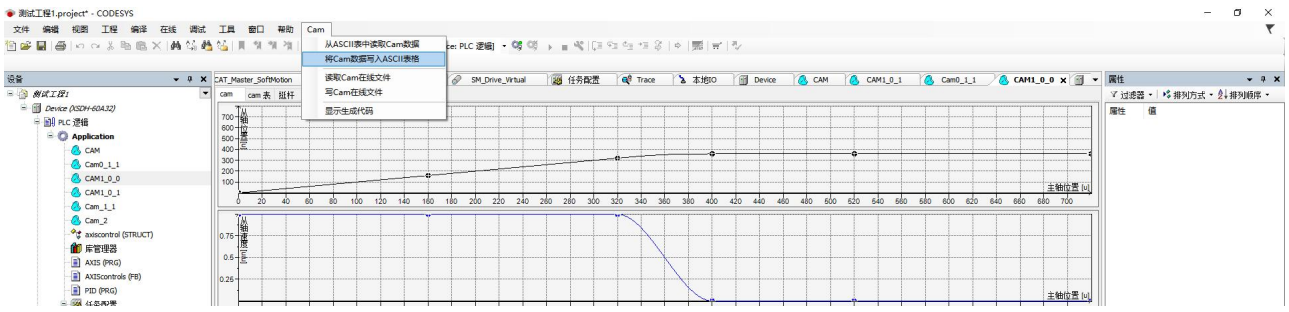
Instruction	Name	Graphical representation	ST language
SMC_WriteCAM	modify the cam table	SMC_WriteCAM_0 SM3_Basic.SMC_WriteCAM CAM bExecute sFileName bDone bBusy bError ErrorID	SMC_WriteCAM(CAM:= , bExecute:= , sFileName:= , bDone=> , bBusy=> , bError=> , ErrorID=>);

(2) Related variables

VAR_IN_OUT	Name	Data type	Effective range	Initial value	Description
CAM	cam	MC_CAM_REF	-	-	map to the cam
VAR_INPUT	Name	Data type	Effective range	Initial value	Description
bExecute	execute	BOOL	TRUE/FALSE	FALSE	execute the function block at the rising edge
sFileName	file name	STRING(255)			The name of the file, which contains the defined cam description in ASCII format
VAR_OUTPUT	Name	Data type	Effective range	Initial value	Description
bDone	completed	BOOL	TRUE/FALSE	FALSE	True if the cam is written into the file
bBusy	executing	BOOL	TRUE/FALSE	FALSE	True when the execution of the function block has not yet ended
bError	error	BOOL	TRUE/FALSE	FALSE	Function block execution error
nErrorID	error code	SMC_ERROR	-	0	Error recognition

(3) Function description

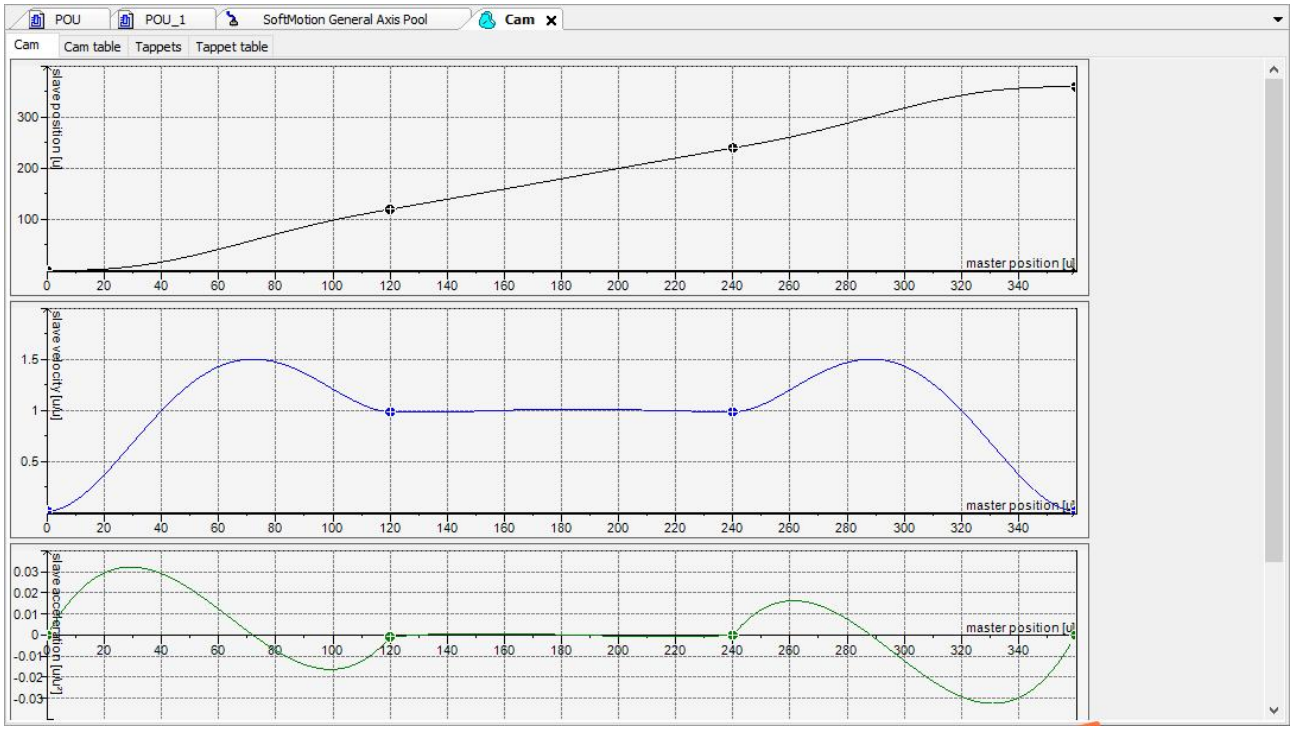
- Store the CAM information connected with “CAM” in the file “sFileName” at the rising edge of bExecut.
- Successfully stored and completed bDone signal output is true.
- The stored cam table information is limited by hardware memory.
- Attention: This function can be executed during program operation, and cam table information can also be manually stored in offline information.



3-3-3. CAM function application

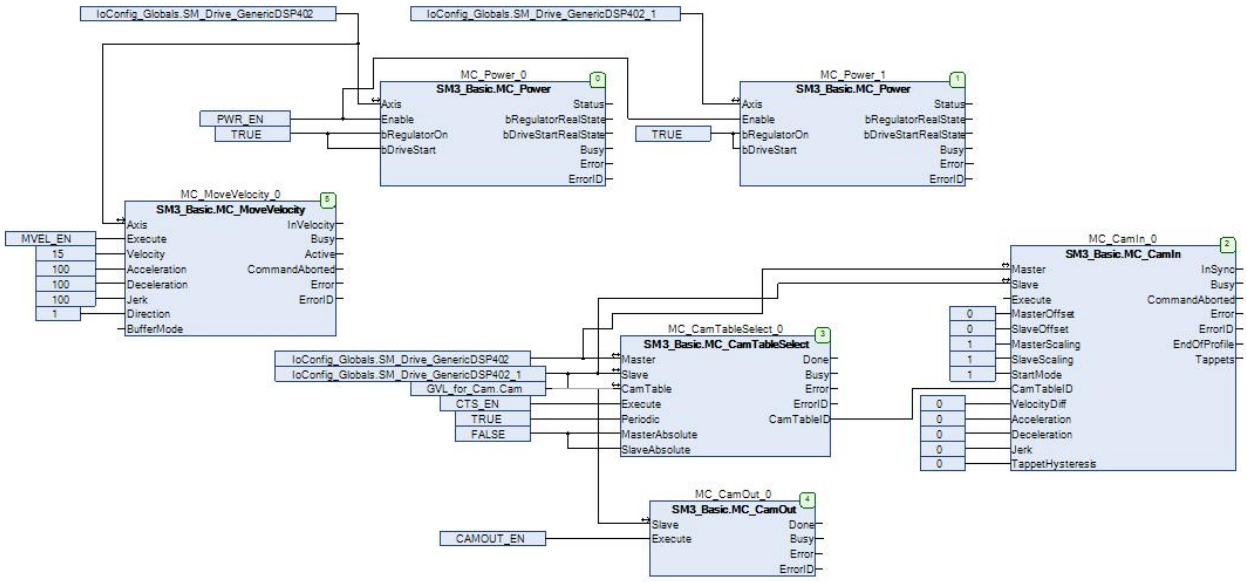
Example 1: Using an electronic cam to achieve spindle running speed mode, with a speed of 15 Plus/S, an acceleration of 100 Plus/S^2, and a positive direction. The slave station runs according to the cam table.
Program operation:

(1) Right click on "Application" in the engineering equipment bar, select "Add Object" - "Cam Table", name it Cam, and open it to add the corresponding cam table parameters as shown in the following figure.



(2) Add the enable module MC_Power, spindle running speed module MC_MoveVelocity in the main program POU_1, calling the cam table module MC_CamTableSelect, configured the electronic cam module MC_CamIn and unbinding electronic cam module MC_CamOut.

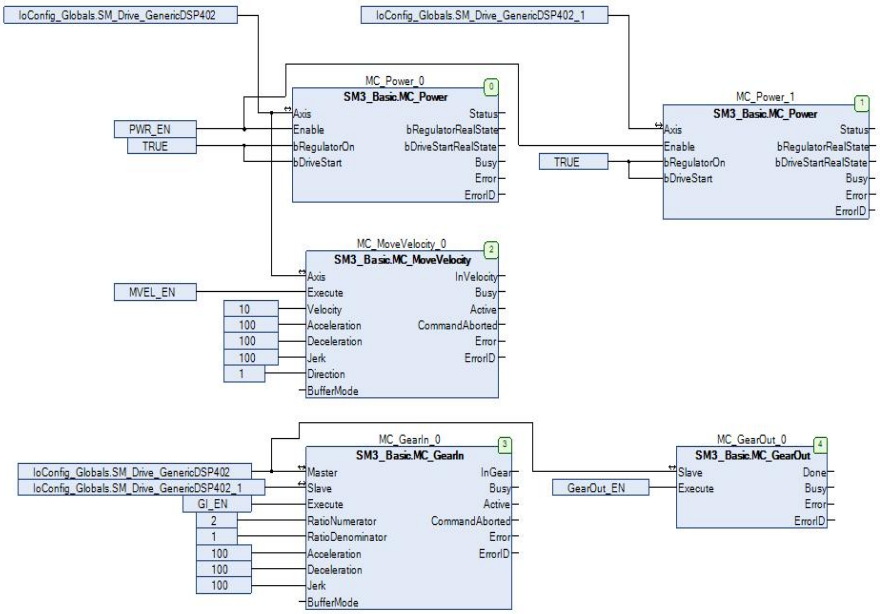
```
PROGRAM POU_1
VAR
    MC_Power_0: SM3_Basic.MC_Power;
    PWR_EN: BOOL;//主从使能 master slave enabled
    MC_Power_1: SM3_Basic.MC_Power;
    MC_CamIn_0: SM3_Basic.MC_CamIn;
    MC_CamTableSelect_0: SM3_Basic.MC_CamTableSelect;
    CTS_EN: BOOL;//凸轮表加载 load the cam table
    MC_CamOut_0: SM3_Basic.MC_CamOut;
    CAMOUT_EN: BOOL;//凸轮解绑 unbinding cam
    MC_MoveVelocity_0: SM3_Basic.MC_MoveVelocity;
    MVEL_EN: BOOL;//主轴速度模式 master axis speed mode
END_VAR
```



Note: If the slave axis is in motion during CAMOUT execution, and after the command is executed, the slave axis will continue to run at its original speed, stop the instruction through MC_Stop, MC_Halt.

Example 2: Using electronic gears to achieve spindle running speed mode, with a speed of 15 Plus/S, an acceleration of 100 Plus/S^2, and a positive direction. The slave station moves according to the electronic gear ratio of 2. So at this point, the speed of the slave station is 30 Plus/S.

```
PROGRAM POU_2
VAR
    MC_Power_0: SM3_Basic.MC_Power;
    MC_Power_1: SM3_Basic.MC_Power;
    PWR_EN: BOOL;//使能 enable
    MC_GearIn_0: SM3_Basic.MC_GearIn;
    MC_GearOut_0: SM3_Basic.MC_GearOut;
    MC_MoveVelocity_0: SM3_Basic.MC_MoveVelocity;
    MVEL_EN: BOOL;//速度模式 speed mode
    GearOut_EN: BOOL;//断开电子齿轮 disconnect the electronic cam
    GI_EN: BOOL;//绑定电子齿轮 binding the electronic cam
    SMC_GroupPower_0: SM3_Robotics.SMC_GroupPower;
END_VAR
```


IoConfig_Globals.SM_Drive_GenericDSP402
IoConfig_Globals.SM_Drive_GenericDSP402_1
MC_Power_0
SM3_Basic.MC_Power
Axis
Enable
bRegulatorOn
bDriveStart
Status
bRegulatorRealState
bDriveStartRealState
Busy
Error
ErrorID
PWR_EN
TRUE
MC_Power_1
SM3_Basic.MC_Power
TRUE
MC_MoveVelocity_0
SM3_Basic.MC_MoveVelocity
Axis
Execute
Velocity
Acceleration
Deceleration
Jerk
Direction
BufferMode
InVelocity
Busy
Active
CommandAborted
Error
ErrorID
MVEL_EN
10
100
100
100
1
MC_GearIn_0
SM3_Basic.MC_GearIn
Master
Slave
Execute
RatioNumerator
RatioDenominator
Acceleration
Deceleration
Jerk
BufferMode
InGear
Busy
Active
CommandAborted
Error
ErrorID
IoConfig_Globals.SM_Drive_GenericDSP402
IoConfig_Globals.SM_Drive_GenericDSP402_1
GI_EN
2
1
100
100
100
MC_GearOut_0
SM3_Basic.MC_GearOut
Slave
Execute
Done
Busy
Error
ErrorID
GearOut_EN

Appendix

Error code	Reasons	Description
0	SMC_NO_ERROR	no error
1	SMC_DI_GENERAL_COMMUNICATION_ERROR	Communication error. The fieldbus slave station is no longer operational
2	SMC_DI_AXIS_ERROR	axis error
3	SMC_DI_FIELDBUS_LOST_SYNCRONICITY	Loss of synchronization of fieldbus
10	SMC_DI_SWLIMITS_EXCEEDED	The position exceeds the SWLimit setting range
11	SMC_DI_HWLIMITS_EXCEEDED	Abnormal hardware position limit
12	SMC_DI_LINEAR_AXIS_OUTOFRANGE	Linear axis incremental position overflow
13	SMC_DI_HALT_OR_QUICKSTOP_NOT_SUPPORTED	Drive status Halt or Quickstop not supported
14	SMC_DI_VOLTAGE_DISABLED	Drive does not have power
15	SMC_DI_IRREGULAR_ACTPOSITION	This error is no longer used
16	SMC_DI_POSITIONLAGERROR	Excessive position error
17	SMC_DI_HOMING_ERROR	homing error
18	SMC_DI_LICENSING_ERROR	License issues
20	SMC_REGULATOR_OR_START_NOT_SET	Axis status cannot execute motion control commands
21	SMC_WRONG_CONTROLLER_MODE	Axis in incorrect controller mode
25	SMC_INVALID_ACTION_FOR_LOGICAL	Invalid logical axis action
30	SMC_FB_WASNT_CALLED_DURING_MOTION	The motion command is running and cannot be interrupted
31	SMC_AXIS_IS_NO_AXISS_REF	AXIS_REF variable type error
32	SMC_AXIS_REF_CHANGED_DURING_OPERATION	AXIS_REF variable has been swapped while the module is active
33	SMC_FB_ACTIVE_AXIS_DISABLED	Axis disabled during movement. MC_Power.bRegulatorOn
34	SMC_AXIS_NOT_READY_FOR_MOTION	Axis is unable to execute motion commands in its current state because Axis has not issued a signal indicating that it is following the target value
35	SMC_AXIS_ERROR_DURING_MOTION	Drive encountered an error while running
40	SMC_VD_MAX_VELOCITY_EXCEEDED	Over the max speed fMaxVelocity
41	SMC_VD_MAX_ACCERATION_EXCEEDED	Over the max acceleration fMaxAcceleration
42	SMC_VD_MAX_DEDRATION_EXCEEDED	Over the max deceleration fMaxDeceleration
50	SMC_3SH_INVALID_VELACC_VALUES	Invalid speed or acceleration value
51	SMC_3SH_MODE_NEEDS_HWLIMIT	Mode request using terminal switches for security reasons
67	SMC_MAC_TOO_MANY_TASK	There are too many tasks to generate axes using SDO
68	SMC_MAC_ATOMIC_ADD_FAILED	Atomic addition failed
69	SMC_SDO_INVALID_DATALENGTH	SDO reading resulted in invalid data length (>4)
70	SMC_SCM_NOT_SUPPORTED	not supported mode
71	SMC_SCM_AXIS_IN_Error_STATE	In the current mode, the controller mode cannot be changed
72	SMC_SCM_INTERRUPTED	SMC_SetControllerMode is interrupted by MC_Stop or errorstop
75	SMC_ST_WRONG_CONTROLLER_MODE	The axis is not in the correct controller mode. Abandoned and no longer returned by SMC_ SetTorque
80	SMC_RAG_ERROR_DURRING_STARTUP	axis group startup error
81	SMC_RAG_ERROR_AXIS_NOT_INITIALIZED	The axis has not reached the target state yet
85	SMC_PP_WRONG_AXIS_TYPE	Function block does not support virtual or

Error code	Reasons	Description
		logical axes
86	SMC_PP_NUMBER_OF_ABSOLUTE_BITS_INVALID	The absolute number of digits is invalid and must be within the range of 8~32
90	SMC_CGR_ZERO_VALUES	invalid value
91	SMC_CGR_DRIVE_POWERED	As long as the drive is in control mode, the transmission parameters must not be changed
92	SMC_CGR_INVALID_POSPERIOD	modulus cycle invalid (<=0 or >half of the bus bandwidth)
93	SMC_CGR_POSPERIOD_NOT_INTEGRAL	The modulus period expressed in increments is not an integer, but the modulus is processed by driver
110	SMC_P_FTASKCYCLE_EMPTY	Axis does not contain information about cycle time (fTaskCycle=0)
120	SMC_R_NO_ERROR_TO_RESET	Error free axis
121	SMC_R_DRIVE_DOESNT_ANSWER	Axis does not execute error reset
122	SMC_R_ERROR_NOT_RESETABLE	Unable to reset error
123	SMC_R_DRIVE_DOESNT_ANSWER_IN_TIME	Communication with axis is not working
124	SMC_R_CANNOT_RESET_COMMUNICATION_ERROR	If there is a communication error, it cannot be reset
130	SMC_RP_PARAM_UNKNOWN	Unknown parameter number
131	SMC_RP_REQUESTING_ERROR	Error transferring to drive. Please refer to the error number in the function block instance ReadDriveParameter
132	SMC_RP_DRIVE_PARAMETER_NOT_MAPPED	No available drive parameter assignments
133	SMC_RP_PARAM_CONVERSION_ERROR	Failed to convert value to drive parameter/from drive parameter to value. Unknown soft motion parameters
140	SMC_WP_PARAM_INVALID	Parameter number unknown or not allowed to be written
141	SMC_WP_SENDING_ERROR	Please refer to the error number in the function block instance WriteDriveParameter
142	SMC_WP_DRIVE_PARAMETER_NOT_MAPPED	No available drive parameter assignments
143	SMC_WP_PARAM_CONVERSION_ERROR	Failed to convert value to drive parameter/from drive parameter to value. Unknown soft motion parameters
170	SMC_H_AXIS_WASNT_STANDSTILL	The axis is not in a stationary state
171	SMC_H_AXIS_DIDNT_START_HOMING	Drive not started homing
172	SMC_H_AXIS_DIDNT_ANSWER	Drive not responding when homing is completed
173	SMC_H_ERROR_WHEN_STOPPING	Error stopping after reset. No need to set deceleration
174	SMC_H_AXIS_IN_ERRORSTOP	The drive is in an error stopped state. Unable to perform homing.
180	SMC_MS_UNKNOWN_STOPPING_ERROR	Unknown error while stopping
181	SMC_MS_INVALID_ACCDEC_VALUES	Invalid speed or acceleration value
182	SMC_MS_DIRECTION_NOT_APPLICABLE	Direction=shortest not applicable
183	SMC_MS_AXIS_IN_ERRORSTOP	The drive is in an error stopped state. Unable to execute stop
184	SMC_BLOCKING_MC_STOP_WASNT_CALLED	Not call through Execute=TRUE to block axis instance MC_Stop. Must call MC_Stop (Execute=FALSE)
185	SMC_MS_AXIS_ALREADY_STOPPING	Unable to interrupt ongoing MC_ Stop
200	SMC_UNKNOWN_TASK_INTERVAL	Unable to determine task interval for bus task
201	SMC_MA_INVALID_VELACC_VALUES	Invalid speed or acceleration value
202	SMC_MA_INVALID_DIRECTION	Direction error

Error code	Reasons	Description
226	SMC_MR_INVALID_VELACC_VALUES	Invalid speed or acceleration value
227	SMC_MR_INVALID_DIRECTION	Direction error
251	SMC_MAD_INVALID_VELACC_VALUES	Invalid speed or acceleration value
252	SMC_MAD_INVALID_DIRECTION	Direction error
276	SMC_MSI_INVALID_VELACC_VALUES	Invalid speed or acceleration value
277	SMC_MSI_INVALID_DIRECTION	Direction error
278	SMC_MSI_INVALID_EXECUTION_ORDER	Except interruption, do not call main mvtFB after SMC_MoveSupplicated
300	SMC_LOGICAL_NO_REAL_AXIS	No longer in use, for compatibility only
301	SMC_MV_INVALID_ACCDEC_VALUES	Invalid speed or acceleration value
302	SMC_MV_DIRECTION_NOT_APPLICABLE	Direction=shortest/fastest not applicable
325	SMC_PP_ARRAYSIZE	Wrong array size
326	SMC_PP_STEP0MS	Step time =t#0s
350	SMC_VP_ARRAYSIZE	Wrong array size
351	SMC_VP_STEP0MS	Step time =t#0s
375	SMC_AP_ARRAYSIZE	Wrong array size
376	SMC_AP_STEP0MS	Step time =t#0s
400	SMC_TP_TRIGGEROCCUPIED	Trigger activated
401	SMC_TP_COULDNT_SET_WINDOW	DriveInterface not support window function
402	SMC_TP_COMM_ERROR	Communication error
410	SMC_AT_TRIGGERNOTOCCUPIED	Trigger unassigned
426	SMC_MCR_INVALID_VELACC_VALUES	Invalid speed or acceleration value
427	SMC_MCR_INVALID_DIRECTION	Invalid direction
451	SMC_MCA_INVALID_VELACC_VALUES	Invalid speed or acceleration value
452	SMC_MCA_INVALID_DIRECTION	Invalid direction
453	SMC_MCA_DIRECTION_NOT_APPLICABLE	Direction=fastest, not applicable
475	SMC_SDL_INVALID_AXIS_STATE	SMC_ChangeDynamicLimits can only be called in a static state. or power_off
476	SMC_SDL_INVALID_VELACC_VALUES	Invalid speed, acceleration, deceleration, or jerk
600	SMC_CR_NO_TAPPETS_IN_CAM	Cam not set tappet point
601	SMC_CR_TOO_MANY_TAPPETS	tappets ID exceeded MAX_NUM_TAPPETS
602	SMC_CR_MORE_THAN_32_ACCESSES	a CAM_REF over 32 access times
625	SMC_CI_NO_CAM_SELECTED	not select cam
626	SMC_CI_MASTER_OUT_OF_SCALE	Spindle out of valid range
627	SMC_CI_RAMPIN_NEEDS_VELACC_VALUES	Must be ramp_ Must specify the velocity and acceleration values for ramp_in function
628	SMC_CI_SCALING_INCORRECT	Scale the variable fEditor/TableMasterMin/Max incorrect
629	SMC_CI_TOO_MANY_TAPPETS_PER_CYCLE	Too many tappets are active in one cycle
640	SMC_CB_NOT_IMPLEMENTED	Function block of given cam format is not implemented
675	SMC_GI_RATIO_DENOM	RatioDenominator = 0
676	SMC_GI_INVALID_ACC	Invalid acceleration
677	SMC_GI_INVALID_DEC	Invalid deceleration
678	SMC_GI_MASTER_REGULATOR_CHANGED	The status of the main device 'enabled/disabled' has been changed without permission
679	SMC_GI_INVALID_JECK	Jerk invalid
725	SMC_PH_INVALID_VELACCDEC	Invalid speed and acceleration/deceleration values
726	SMC_PH_ROTARYAXIS_PERIOD0	fPositionPeriod=0 modulus axis
750	SMC_NO_CAM_REF_TYPE	The given cam type is not MC_CAM_REF
751	SMC_CAM_TABLE_DOES_NOT_COVER_MASTER_SCALE	The curve data does not include the main regions xStart and xEnd in CamTable
752	SMC_CAM_TABLE_EMPTY_MASTER_RANGE	The main range of the Cam data table is

Error code	Reasons	Description
		empty
753	SMC_CAM_TABLE_INVALID_MASTER_MINMAX	Cam data master has invalid maximum and minimum values
754	SMC_CAM_TABLE_INVALID_SLAVE_MINMAX	Cam data slave device has invalid maximum and minimum values
775	SMC_GIP_MASTER_DIRECTION_CHANGE	During the slave axis coupling, the spindle changed its direction of rotation
776	SMC_GIP_SLAVE_REVERSAL_CANNOT_BE_AVOIDED	Input AvoidReversal has been set, but it cannot be avoided from slave reversing
777	SMC_GIP_AVOID_REVERSAL_FOR_FINITE_AXIS	Input AvoidReversal cannot be limited slave axis setting
778	SMC_GIP_MASTERSTARTDISTANCE_MUST_BE_ZERO_BUFFERED	If BufferMode hasn't been interrupted, then MasterStartInstance cannot be positive
779	SMC_GIP_CANNOT_START_SYNC	Unable to start synchronization. This situation may occur when GearInPos is commanded to act as a buffer or hybrid motion, and the main controller is in a stationary state when activated
800	SMC_BC_BL_TOO_BIG	Gear return ratio (fBacklash) too large (>position periode/2)
825	SMC_QPROF_DIVERGES	Internal error: quadratic trajectory calculation failed
826	SMC_QPROF_INVALID_PARAMETER	Internal error: quadratic trajectory calculation failed
827	SMC_QPROF_NO_RESULT	Internal error: quadratic trajectory calculation failed
828	SMC_QPROF_INVALID_NEW_LBD	Internal error: quadratic trajectory calculation failed
829	SMC_QPROF_BAD_NEGOTIATION	Internal error: quadratic trajectory calculation failed
830	SMC_QPROF_INVALID_INTERVAL	Internal error: quadratic trajectory calculation failed
831	SMC_QPROF_NOT_ENOUGH_PHASES	Internal error: quadratic trajectory calculation failed
832	SMC_TG_INTERNAL_ERROR	Internal error: quadratic trajectory calculation failed
850	SMC_SRT_NOT_STANDSTALL_OR_POWEROFF	Allow action only when stationary or powered off
851	SMC_SRT_INVALID_RAMPTYPE	Invalid slope type
852	SMC_SMT_NOT_STANDSTALL_OR_POWEROFF	Allow action only when stationary or powered off
853	SMC_SMT_INVALID_MOVEMENTTYPE_OR_POSITIONPERIOD	Invalid motion type or position period
854	SMC_SMT_AXIS_NOT_VIRTUAL	Function block only applies to virtual axes
1000	SMC_NO_LICENSE	Target not licensed by CNC
1001	SMC_INT_VEL_ZERO	Unable to process path because set speed=0
1002	SMC_INT_NO_STOP_AT_END	The end speed of the last object in the path <>0
1003	SMC_INT_DATA_UNDERRUN	Warning: The geographic information list was processed in DataIn, but did not reach the end of the list. Reason: Unable to set EndOfList SMC_Interpolator of queue in DataIn faster than path generation function blocks
1004	SMC_INT_VEL_NONZERO_AT_STOP	speed when stop>0
1005	SMC_INT_TOO_MANY_RECURSIONS	SMC_Interpolator excessive recursion. Soft

Error code	Reasons	Description
		motion error
1006	SMC_INT_NO_CHECKVELOCITIES	SMC_CheckVelocity is not the last processed function block, it accessed data outside the queue through poqDataIn
1007	SMC_INT_PATH_EXCEEDED	Internal or numerical error
1008	SMC_INT_VEL_ACC_DEC_ZERO	Speed and acceleration/deceleration are zero or too low
1009	SMC_INT_DWIPOTIME_ZERO	Use dwIpoTime=0 to call FB
1010	SMC_INT_JERK_NONPOSITIVE	Jerk invalid, as Jerk must be positive
1011	SMC_INT_QPROF_DIVERGES	Internal error. The calculation of quadratic velocity distribution does not converge
1012	SMC_INT_INVLALID_VELOCITY_MODE	Invalid speed mode
1013	SMC_INT_TOO_MANY_AXES_INTERPOLATED	The number of inserted axes exceeds the allowed number of axes. You are using a restricted version
1014	SMC_INT_DEGENERATE_SEGMENT	This segment is numerically degenerate: it is very short and located at the end of the queue. It should be ignored
1015	SMC_HIGH_CURVATURE_SPLINE	The calculation of interpolation points failed because the curvature of the spline curve is too high. Try changing the path to avoid sharp corners
1050	SMC_INT2DIR_BUFFER_TOO_SMALL	Warning: The poqDataIn of the created output queue is too small, cannot guarantee compliance with stopping
1051	SMC_INT2DIR_PATH_FITS_NOT_IN_QUEUE	The path is not completely in the queue
1070	SMC_XINT_INVALID_DIRECTION	Direction input has invalid value
1071	SMC_XINT_NOINTERSECTION	Unable to determine the position of the given x position on the CNC path
1080	SMC_WAR_INT_OUTQUEUE_TOO_SMALL	Warning: The poqDataIn of the created output queue is too small, cannot guarantee compliance with station
1081	SMC_WAR_END_VELOCITIES_INCORRECT	Warning: Inconsistent final speed
1100	SMC_CV_ACC_DEC_VEL_NONPOSITIVE	Speed and acceleration/deceleration values: non positive
1120	SMC_CA_INVALID_ACCDEC_VALUES	fGapVelocity/fGapAcceleration/FGAPDegregation values: non positive
1130	SMC_TOK_COMPLETE_TOKEN_AT_END_OF_INPUT	Input has been exhausted, but there are still unfinished marks
1131	SMC_TOK_NOT_A_VALID_TOKEN	Input does not match any token type
1132	SMC_TOK_MUBILITY_INPUT	Ambiguous input, there may be multiple token types
1133	SMC_TOK_STRING_TOO_LONG_FOR_TOKEN	The string is too long to store in the token. (String text, variable name, or identifier too long)
1134	SMC_TOK_INVALID_NUMLIT	Invalid numeric text
1150	SMC_PRS_FUNC_DECL_TOO_MORY_ARGMENTS	Too few parameters provided for the function in G code
1151	SMC_PRS_FUNC_DECL_TOO_MANY_ARGMENTS	Too many parameters provided for the function in G code
1152	SMC_PRS_FUNC_DECL_WRONG_ARGUMENT_TYPE	Check the parameter types that match the function declaration. Return error
1153	SMC_PRS_LOCAL_VAR_NOT_FOUND	Cannot find local variable on the stack
1154	SMC_PRS_INVALID_STRING	Unable to read string from token
1155	SMC_PRS_TOO_MANY_CLOSING_BRACKETS	More closing parentheses than opening parentheses
1156	SMC_PRS_TOO_MANY_OPENING_BRACKETS	More start parentheses than end parentheses

Error code	Reasons	Description
1157	SMC_PRS_NO_SUCH_INFIX_OPERATOR	Unable to find infix operator
1158	SMC_PRS_NO_SUCH_PREFIX_OPERATOR	Unable to find prefix operator
1159	SMC_PRS_OPERATOR_INVALID_PRECEDENCE	Obtain two operators with equal priority but unequal associativity
1160	SMC_PRS_NOT_A_TERM	Expected validity period
1161	SMC_PRS_EXPRESSION_INVALID_SEQUENCE	Invalid tag sequence found in expression
1162	SMC_PRS_TOO_MANY_TERMS	Obtained more terms than expected
1163	SMC_PRS_STACK_OVERFLOW	Unable to parse expression because the stack is too small
1164	SMC_PRS_VAR_NAME_READY_USED	The name of a subroutine parameter or variable is already used for other subroutine parameters or parameters
1165	SMC_PRS_INCOMPLETE_SENTENCE_IN_TOKEN_QUEUE	The input token queue of the g-code parser does not contain a complete g-code statement
1167	SMC_PRS_TOO_MANY_SUBPROGRAMS	Cannot store subroutine declaration because symbol table capacity has been exceeded
1168	SMC_PRS_TOO_MANY_SUBPROGRAM_PARAMETERS	The maximum number of subroutine parameters has been exceeded
1169	SMC_PRS_SUBPROGRAM_LOOKUP_FAILED	Failed to find subroutine declaration. Perhaps the subroutine name is incorrect, or the search path for the subroutine is incomplete
1170	SMC_PRS_VAR_NOT_FOUND	Variable not found in symbol table
1171	SMC_PRS_TOKEN_TYPE_UNKNOWN	Unknown token type
1172	SMC_PRS_GOT_NO_TERM	No terminology after G-address
1173	SMC_PRS_INVALID_VAR_TYPE	Unknown variable type (not LREAL, BOOL, or string)
1174	SMC_PRS_UNEXPECTED_TOKEN	Different types of tokens (such as operators or identifiers) are required here
1175	SMC_PRS_ID3_EXPECTED	An identifier with a length of 3 or longer is required here
1176	SMC_PRS_ID_TOO_LONG	Identifier too long (over 80 characters)
1177	SMC_PRS_GADDRESS_EXPECTED	G address is required here (such as "G", "F", "X")
1178	SMC_PRS_NWORD_EXPECTED	An N character is required here (such as' N10 ')
1179	SMC_PRS_NWORD_INVALID_SENTURE_NUMBER	The sentence number is not within the range of [0,..., 16 # FFFFFFFF]
1180	SMC_PRS_NWORD_SENTENCE_NUMBER_NO_NUMBER_LITERAL	The sentence number must be a numerical literal. (For example, expressions are not allowed.)
1181	SMC_PRS_USE_OF_RESERVED_KEYWORD	Identifier is a reserved keyword and cannot be used here
1182	SMC_PRS_SUBPROGRAMS_SIGNATURE_MISMATCH	A different signature is used or the subroutine declaration has been read error
1183	SMC_PRS_INITIAL_VALUE_HAS_WRONG_TYPE	The initial value of this local variable has the wrong type
1184	SMC_PRS_TOO_MANY_LOCAL_VARIABLES	The maximum number of subroutine parameters has been exceeded
1185	SMC_PRS_SUBPROG_MUST_BE_FIRST_SENTENCE	The subroutine declaration must be the first sentence in the g code file
1186	SMC_PRS_ONLY_ONE_subgram_PER_FILE_ALLOWED	Only one subroutine is allowed in the G code file.
1187	SMC_PRS_LET_AFTER_REGULAR_SENTENCE	LET declaration is not allowed after regular G-code statements
1188	SMC_PRS_UNMATCHED_END_SUBPROGRAM	END_SUBPROGRAM. statement does not

Error code	Reasons	Description
		match subroutine declaration
1189	SMC_PRS_UNEXPECTED_TOKENS_AFTER_SUBPROGRAM	After END_SUBPROGRAM., no more tokens allowed
1190	SMC_PRS_MISSING_END_SUBPROGRAM	The subroutine does not terminat with END_SUBPROGRAM.
1200	SMC_DEC_ACC_TOO_LITTLE	Acceleration value not allowed
1201	SMC_DEC_RET_TOO_LITTLE	Deceleration value not allowed
1202	SMC_DEC_OUTQUEUE_RAN_EMPTY	Insufficient data. Queue read and empty
1203	SMC_DEC_JUMP_TO_UNKNOWN_LINE	Unable to perform jump to line because the line number is unknown
1204	SMC_DEC_INVALID_SYNTAX	Invalid syntax
1205	SMC_DEC_3DMODE_OBJECT_NOT_SUPPORTED	Object not supported in 3D mode
1206	SMC_DEC_NEGATIVE_PERIOD	Invalid negative value as additional axis period
1207	SMC_DEC_DIMENSIONS_EXCLUSIVE_AU	Interpolation is not performed on both axes A and U. PA and PU are mutually exclusive
1208	SMC_DEC_DIMENSIONS_EXCLUSIVE_BV	Interpolation is not performed on both axes B and V. PB and PV are mutually exclusive
1209	SMC_DEC_DIMENSIONS_EXCLUSIVE_CW	Interpolation is not performed on both axes C and W. PC and PW are mutually exclusive
1210	SMC_DEC_DCS_NOT_ALL_OF_ABC_GIVEN	For G54/G55/G56, if the directional mode is not equal to SMC_ORI_CONVENTION.ADDAXES, then all A, B, C or none must be given
1211	SMC_DEC_DCS_2D_NOT_IN_XY_PLANE	The Z-axis of the decoder CS must be equal to the Z-axis of the machine CS
1212	SMC_DEC_CIRCLE_NON_UNIFORM_SCALING	Scaling the circle or ellipse that has twist command
1213	SMC_DEC_ROTATION_AFFECTS_SCALING	The new relative rotation of DCS (G55) will affect scaling
1214	SMC_DEC_DCS_NOT_ALL_OF_IJK_GIVEN	For G54/G55/G56, all I, J, K or none must be given
1250	SMC_IPR_LOCAL_VAR_UNKNOWN_TYPE	Unable to read local variable due to invalid type
1251	SMC_IPR_LVALUE_WRONG_TYPE	The variable type that should be written is incorrect
1252	SMC_IPR_EVAL_STACK_OVERFLOW	Unable to evaluate expression because the evaluation stack is too small
1253	SMC_IPR_NOT_A_NUMBER	The numerical term becomes NaN
1254	SMC_IPR_DIVISION_BY_ZERO	Divide by zero
1255	SMC_IPR_INVALID_SCALING_FACTORS	Invalid Scale Factor
1256	SMC_IPR_LVALUE_WRONG_TYPE	When the subroutine returns, the interpreter stack is empty. internal error
1257	SMC_IPR_INTERPRETER_STACK_OVERFLOW	The interpreter stack is too small. There are too many local variables in the G code or too many active subroutines
1258	SMC_IPR_INVALID_INTERPRETER_STACK_BUFFER	The given buffer of the interpreter stack is 0 or less than 1024 bytes
1259	SMC_IPR_BUFFER_TOO_SMALL	The buffer of the output queue is too small
1280	SMC_DNCCS_NO_DATA	There is no available data at all
1281	SMC_DNCCS_TOO_MANY_CALLSTACKS	There are too many call stacks between the current state of iObjNo and the interpreter
1282	SMC_DNCCS_INVALID_PROGRAM_INDEX	Invalid program index
1283	SMC_DNCCS_TOO_MANY_PROGRAM	Too many different programs
1284	SMC_DNCCS_WRONG_TASK	FB called the wrong task
1300	SMC_GCV_BUFFER_TOO_SMALL	buffer too small
1301	SMC_GCV_BUFFER_WRONG_TYPE	Wrong type of buffer element

Error code	Reasons	Description
1302	SMC_GCV_UNKNOWN_IPO_LINE	Unable to find the current row of the interpolator
1400	SMC_CNC_INTERNAL_ERROR	Internal error in CNC
1401	SMC_PATH_MAX_HPOINTS_EXCEEDED	The path element cannot be saved beyond H switching points of MAX_IPOSWITCHES. Adjust the switching point using different "O" values, or reduce the number of H-points for each path element
1410	SMC_TRC_G75_NOT_ALLOWED	G75 is not allowed during tool radius correction
1411	SMC_TRC_QUEUE_FULL_NON_CARTESIAN	The queue is full, but there are no other Cartesian elements
1412	SMC_TRC_SPLINE3D_5_NOT_SUPPORTED	Tool radius correction does not support motion type SPLINE3D_5
1414	SMC_TRC_PLANE_CHANGE_NOT_ALLOWED	Do not allow plane changes during tool radius correction
1450	SMC_NAV_MAX_SUBSPROGRAM_NESTING_EXCEEDED	The maximum nesting level for subroutine call has been exceeded
1451	SMC_NAV_RETURN_FROM_MAIN	Do not allow main G code programs to return statements
1452	SMC_NAV_SUBPROGRAM_DECLARATION_NOT_FOUND	The subroutine declaration was not found in the subroutine CNC file
1453	SMC_NAV_NOT_ENOUGH_SPACE_FOR_COMPLETE_SENTENCE	Unable to add the next G code statement to the output statement queue because there is not enough space
1500	SMC_NO_CNC_REF_TYPE	The given CNC program is not SMC_ CNC_ REF type
1501	SMC_NO_OUTQUEUE_TYPE	The given output queue is not SMC_ OutQueue type
1502	SMC_GEOINFO_BUFFER_MISALIGNED	Not using 4-byte aligned buffer section in pbyBuffer
1600	SMC_3D_MODE_NOT_SUPPORTED	Function block only applies to 2D paths
1700	SMC_SAA_SMOOTHAREA_TOO_LARGE	Smooth range too large
1701	SMC_SAA_SP_INVALID_INPUT	Invalid input dSmoothingPart]0..1]
1800	SMC_SA_QUEUE_NOT_IN_BUFFER	SMC_SegmentAnalyzer detected that the out of queue buffer is full but not complete. The function block can only run when OutQueue is fully suitable for the buffer
1801	SMC_SA_QUEUE_CHANGED_DURING_OP	When the function block operates on it, the out of queue buffer has changed
1820	SMC_OS_INVALID_PARAMETER	Invalid input value inside dSplittingParameter
1830	SMC_BSSP_IPO_NOT_ACTIVE	Unable to save location. Inserter not active
1831	SMC_BS_SAVEDPOS_NOT_REACHED	No saved location has been found so far. It may have taken a different path
1832	SMC_BS_NO_POS_STORED	The structure passed in ePos does not contain a saved location. SMC_BlockSearchSavePos did not executed or executed in an incorrect manner
1900	SMC_INVALID_FEATURE_FLAG	The functional symbol must be within the range of {1,..., 31}
1901	SMC_SMB_HFUN_NOT_SUPPORTED	Function block does not support the h function
1902	SMC_SMB_ONLY_3DMODE	Function block only works in 3D mode
1903	SMC_SMB_ERROR_COMPUTING_SPLINE	Calculate the internal error of a spline curve
1910	SMC_SMM_INVALID_PARAM_NUMBER	WadAdditionalParameter too large
1950	SMC_INVALID_PARAMETER	One of the input values is invalid

Error code	Reasons	Description
1951	SMC_BLENDING_NOT_SUPPORTED_BY_PREVIOUS_MOVEMENT	The previous move does not support mixing
1952	SMC_BUFFERED_NOT_SUPPORTED_BY_PREVIOUS_MOVEMENT	The previous move does not support buffered moves
1953	SMC_INHERIT_OWNER_ACTIVE_MOVEMENT_NOT_CALLED	Cannot inherit owner because activity move has not been called in this loop. internal error
1954	SMC_MOVING_WITHOUT_ACTIVE_MOVEMENT	The axis is moving, but there is no active move function block. Losing subsequent movements in one unstoppable movement after another
1955	SMC_BUFFER_MODE_NOT_SUPPORTED	Command movement does not support configured buffering mode
1956	SMC_ERROR_IN_A_PREVIOUS_MOVEMENT	An error occurred during the previous move
1957	SMC_MORE_THAN_ONE_MOVEMENT_PER_INSTANCE	Only one busy buffer/mixed move allowed per FB instance
2000	SMC_RNCF_FILE_DOESNT_EXIST	file does not exist
2001	SMC_RNCF_NO_BUFFER	Unallocated buffer
2002	SMC_RNCF_BUFFER_TOO_SMALL	buffer too small
2003	SMC_RNCF_DATA_UNDERRUN	Insufficient data. Buffer read, it is empty
2004	SMC_RNCF_VAR_COULDNT_BE_REPLACED	Unable to replace placeholder variable
2005	SMC_RNCF_NOT_VARLIST	Input pvl does not point to SMC_ VARLIST Object
2006	SMC_RNCF_NO_STRINGBUFFER	Input pStringBuffer does not be used or pointed to SMC_StringBuffer type variables
2007	SMC_RNCF_STRINGBUFFER_OVERRUN	In CNC programs, more different strings are used than the string buffer can accommodate
2008	SMC_RNCF_SUBPROGRAM_FILE_NOT_FOUND	Unable to find file for subroutine
2050	SMC_RNCQ_FILE_DOESNT_EXIST	could not open file
2051	SMC_RNCQ_NO_BUFFER	Undefined buffer
2052	SMC_RNCQ_BUFFER_TOO_SMALL	buffer too small
2053	SMC_RNCQ_UNEXPECTED_EOF	Unexpected end of file
2100	SMC_ADL_FILE_CANNOT_BE_OPENED	could not open file
2101	SMC_ADL_BUFFER_OVERRUN	Buffer overflow. WriteToFile must be called more frequently
2200	SMC_RCAM_FILE_DOESNT_EXIST	could not open file
2201	SMC_RCAM_TOO_MUCH_DATA	The saved cam is too large
2202	SMC_RCAM_WRONG_COMPILE_TYPE	Incorrect compilation mode
2203	SMC_RCAM_WRONG_VERSION	The version of the file is incorrect
2204	SMC_RCAM_UNEXPECTED_EOF	Unexpected end of file
3001	SMC_WDPF_CHANNEL_OCCUPIED	This error is outdated and reserved for compatibility only. SMC_WDPF_TIMEOUT_PREPARING_LIST
3002	SMC_WDPF_CAN_CREATE_FILE	This error is outdated and reserved for compatibility only. could not create file
3003	SMC_WDPF_ERROR_WHEN_READING_PARAMS	This error is outdated and reserved for compatibility only. Error reading parameters
3004	SMC_WDPF_TIMEOUT_PREPARING_LIST	This error is outdated and reserved for compatibility only. Timed out when preparing parameter list
5000	SMC_ENC_DENOM_ZERO	The conversion coefficient of encoder reference dwRatioTechUnitsDenom named is 0
5001	SMC_ENC_AXISUSEDBYOTHERFB	Attempting to handle other modules moving on the encoder axis
5002	SMC_ENC_FILTER_DEPTH_INVALID	Invalid filter depth
6000	SMC_INTERNAL_ERROR_INIT_MOVEMENT	Internal error when initializing new move

Error code	Reasons	Description
6001	SMC_INVALID_AXIS_TYPE	Invalid axis type, not finite or modular (internal error)
10000	SMC_PCCQ_POINTBUFFERTOOSMALL	Buffer pBuffer too small
10001	SMC_PCCQ_INPUTBUFFERFULLBUTNOTFINALIZED	The function block must be applied to a path that is completely suitable for the buffer
11000	SMC_AXIS_GROUP_WRONG_STATE	The axis group is in an error state for the requested operation
11001	SMC_AXIS_GROUP_TOO_MANY_AXES	Axis group has been added with more than the maximum allowed number of axes
11002	SMC_AXIS_GROUP_INVALID_DYNLIMITS	Invalid dynamic limit for a single axis (fSWMaxVelocity/ acceleration/deceleration/jerk)
11004	SMC_AXIS_GROUP_INVALID_COORD_SYSTEM	The given coordinate system is invalid for the requested operation
11005	SMC_AXIS_GROUP_SINGLE_AXISS_ERROR	The axis of the axis group is in the wrong state
11006	SMC_MOVE_INVALID_BUFFER_MODE	The given buffer mode is not supported
11007	SMC_MOVE_INVALID_DYNAMIC_FACTOR	The dynamic coefficient is not within the range of [0.. 1]
11008	SMC_MOVE_INVALID_DYNAMICS	Invalid dynamic restrictions for movement
11009	SMC_AXIS_GROUP_AXIS_NOT_PART_OF_GROUP	The given axis is not part of an axis group
11010	SMC_AXIS_GROUP_NOT_SUPPORTED	The requested operation is not supported
11011	SMC_AXIS_GROUP_KINEMATICS_NOT_SET	No kinematic configurations have been configured yet
11012	SMC_AXIS_GROUP_WRONG_NUMBER_OF_AXES	The number of axes is not equal to the number of axes required for motion transformation
11013	SMC_AXIS_GROUP_INTERRUPTED_BY_SINGLE_AXIS	Coordinated movement interrupted by single axis movement
11014	SMC_AXIS_GROUP_FOLLOW_SETVALUES	Error tracking calculated settings
11015	SMC_AXIS_GROUP_TOO_MANY_dependency	One axis group cannot rely on multiple axis groups
11016	SMC_AXIS_GROUP_MUTUAL_DEPENDENCY	Axis group A may not depend on another axis group that depends on A
11017	SMC_AXIS_GROUP_DEPENDENCY_IN_DIFFERENT_TASK	The dependent axis group must be executed in the same task
11018	SMC_AXIS_GROUP_AXISS_IN_DIFFERENT_TASK	Axes belonging to an axis group must be executed in the same task
11019	SMC_AXIS_GROUP_PCS_STILL_IN_USE	When PCS is still being used by buffered motion commands, the function block undergoes a second activation
11020	SMC_AXIS_GROUP_CMD_ABORTED_DUE_TO_ERROR	Error in previous motion command
11021	SMC_AXIS_GROUP_INVALID_PARAMETER	Invalid parameters for management function block
11022	SMC_AXIS_GROUP_UNSUPPORTED_RAMPTYPE	One axis in the group uses an unsupported ramp type. Only supports trapezoidal and quadratic types
11023	SMC_MOVE_INVALID_TRANSITION_PARAMETER	Invalid conversion parameter
11024	SMC_AXIS_GROUP_INTERNAL_ERROR	Internal error in the state machine of the axis group
11025	SMC_AXIS_GROUP_CPTR_CANNOT_FOLLOW	CP tracking: unable to maintain path; Attempting to reduce dynamics on the path and/or entering CP tracking
11026	SMC_AXIS_GROUP_CONTINUE_WRONG_POSITION	The current position does not allow continuation
11027	SMC_AXIS_GROUP_CONTINUE_BUFFER_TOO_SM	The buffer in ContinueData is too small;

Error code	Reasons	Description
	ALL	Using large external move queue buffer
11028	SMC_AXIS_GROUP_CONTINUE_WRONG_CHECKSUM	The parity and error in continue data, continue data is not written by MC_GroupInterrupt, but is modified later
11029	SMC_AXIS_GROUP_IDLE_WAIT_AXES_MOVING	Command SMC_GroupWait, simultaneously, the axis moves independently of the axis group (single axis movement or similar)
11030	SMC_AXIS_INVERSE_TRAFO_EXCEEDING_POSLIMITS	The inverse transformation will cause the axis value to exceed the configured limit
11031	SMC_AXIS_GROUP_CANNOT_ADD_SAME_AXIS	Prohibit adding the same axis to the axis group twice
11032	SMC_AXIS_GROUP_TRANSFORMATION_NOT_SET	Do not set conversion
11033	SMC_AXIS_GROUP_NON_RESUMABLE_ERROR	Unable to recover path after current axis group error
11034	SMC_AXIS_GROUP_CONTINUE_DATA_NOT_WRITTEN	The continuation data for MC_GroupContinue was not written correctly. (MC_GroupInterrupt not called/unrecoverable axis group error)
11100	SMC_KERNEL_PTP_INTERNAL_ERROR	Internal errors in the kernel
11101	SMC_KERNEL_PTP_INVALID_TASKCYCLETIME	Task cycle time is not positive
12000	SMC_TRAFO_INVALID_PARAMETERS	The converted parameter value is invalid
12001	SMC_TRAFO_INVALID_CONSTELLATION	The requested location is outside the converted work area
12002	SMC_TRAFO_INVALID_COUNPING	Tools cannot work together with the positioning components of the machine. Alternatively, the positioning component is not implemented ISMPocationKinetics2, or this tool is unable to process directional images for positioning
12003	SMC_TRAFO_NOT_INITIALIZED	FB realized |ISMkineticWithinitialization|, but not initialized yet
12004	SMC_TRAFO_NO_TOOL_WITH_OFFSET_ALLOWED	Special errors in the positioning section Kin_4Axis_Palletzer, it realized ISMPositionKinetics_Offset2. But in reality, it's not possible to handle offsets <>0
12005	SMC_AXIS_GROUP_TOOL_OFFSET_INCOMPATIBLE_WITH_KINEMATICS	The configured tool offset is not compatible with the used kinematics
12100	SMC_CP_CACHE_FULL	CP planner's cache size is too small
12101	SMC_CP_EVAL_ERROR	Calculation of position on path element failed (internal error)
12102	SMC_CP_NON_CONTINUABLE_STATE	Reached non continuous state (internal error)
12103	SMC_CP_MAX_LENGTH_EXCEEDED	Exceeded maximum trajectory length (internal error)
12104	SMC_CP_ACCELERATION_TOO_HIGH	Path acceleration too high (invalid state, internal error)
12105	SMC_CP_MAX_ITERATIONS_EXCEEDEDSMC_CP_MAX_ITERATIONS_EXCEEDED	Exceeded maximum number of iterations (internal error)
12106	SMC_CP_NO_TRAJECTORY	Unable to calculate trajectory (internal error)
12107	SMC_CP_OUT_QUEUE_FULL	Output queue full (internal error)
12108	SMC_CP_QUEUE_UNDERRUN	Insufficient queue operation: there are no remaining elements in the queue
12109	SMC_CP_INVALID_QUEUE	Invalid queue (invalid size or pointer)
12110	SMC_CP_Blend_INTERNAL_ERROR	Internal error while mixing two CP movements
12111	SMC_CP_CIRCLE_COLLINEAR_POINTS	The three points defined a circle are collinear
12112	SMC_CP_CIRCLE_CENTER_NOT_ON_BISECTOR	The center point is not on the vertical

Error code	Reasons	Description
		bisector of the starting and ending points
12113	SMC_CP_CIRCLE_RADIUS_ZERO	Zero radius
12114	SMC_CP_CIRCLE_DISTANCE_LARGER_THAN_DIAMETER	The distance between the start and end points is greater than the diameter (when trying to create a circle using SMC_Circ_Mode.radius)
12115	SMC_CP_CIRCLE_START_AND_ENDPOINT_EQUAL	Equal starting and ending points (SMC_Circ_Mode.Radius)
12116	SMC_CP_BLENDING_DEGENERATE_SPLINE	Mixed spline degradation (too short or irregular)
12117	SMC_CP_ELEMENT_TOO_SHORT	Unable to create path element because it is too short
12118	SMC_CP_CAN_NOT_CUT_PATH	Unable to cut path to abort movement (internal error)
12119	SMC_CP_INVALID_ANGULAR_VEL_ACC	The given angular velocity or acceleration is invalid (quaternions with non zero real parts, internal error)
12120	SMC_CP_INVALID_ORIENTATION	Invalid given direction (non orthogonal matrix or non identity quaternion, internal error)
12121	SMC_CP_TIME_BUDGET_EXCEEDED	Exceeded the calculated given time budget (internal error)
12122	SMC_AXIS_GROUP_AXIS_LIMIT_VIOLATED	Axis restrictions have been violated. If the CP moves too close to the singularity or the path has very high curvature, this situation may occur
12125	SMC_CP_CONFIGS_DIFFER	The kinematic configuration of the starting position is different from the ending position. CP motion will pass through singularities
12126	SMC_CP_BUS_TASK_NOT_CALLED	This error is outdated and reserved for compatibility only. If the command CPhalt/-stop is used, some information must be transferred to the bus task. We expect this to occur before the next cycle of the slow task
12127	SMC_CP_NO_ROOT_IN_INTERVAL_FOUND	Unable to find the stop trajectory in the interval. (Internal error)
12128	SMC_CP_KIN_DOES_NOT_SUPPORT_AXIS_ORIENTATION_IPO	Kinematics does not support continuous path movement direction mode 'axis'
12129	SMC_CP_AXIS_ORIENTATION_IPO_NOT_SUPPORTED_FOR_CPTR	CP movement with dynamic PCS (tracking) does not support axis direction interpolation mode
12130	SMC_CP_INVALID_PATH_ELEM	Invalid path element created (internal error)
12131	SMC_CP_TRANSITION_NOT_SMOOTH	The conversion between two path elements is not G2 continuous and did not stop (internal error)
12132	SMC_CP_AXIS_ORIENTATION_IPO_CONFIG_DIFFERS	The axis direction interpolation mode was used, but there is singularity in positional kinematics between the start and end positions
12133	SMC_CP_AXIS_ORIENTATION_IPO_OFFSET2_NOT_IMPLEMENTED	Position kinematics has configurations but does not implement interfacesIsmPositionKinetics_Offset2. In this case, axis direction interpolation is not possible
12134	SMC_CP_ROTARY_AXIS_PERIOD_MISMATCH	The target position of the rotation axis will not be reached within the command cycle. (For example, the instruction target position

Error code	Reasons	Description
		is -170°, but it will reach the target at position 190°.) This means that the selected target position interpolation is not compatible with the instruction position
12135	SMC_CP_ROTARY_AXIS_RANGE_VIOLATION	During CP movement, the rotation axis violated the allowed axis range. CP movement is not within the work area
12136	SMC_CP_COMPUTE_TARGET_DISCONTINUITY	Due to the discontinuity in the end position (internal error), the target trajectory cannot be calculated
12137	SMC_CP_TRACTION_NOT_SMOOTH	The trajectory is not smooth, and the phase end state is not equal to the current state (internal error)
12138	SMC_CP_ERROR_CREATING_PARAM_TRANSFORM	Unable to create parameter conversion (internal error)
12139	SMC_CP_PTP_DATA_NUMBER_OF_AXES	Error generating path invariant PTP element (internal error)
12140	SMC_CP_NEGATIVE_PATH_VELOCITY	Error calculating new trajectory: Path velocity after stage 1 (positive acceleration of length dTau1) is negative (internal error)
12141	SMC_CP_TRANSITIONING_FROM_SINGLE_AXIS_MOVEMENT_NOT_SUPPORTED	Transition from single axis movement to continuous path movement is not supported
12142	SMC_CP_PLANNER_NO_PROGRESS	CP-planner has not made any progress. This is an internal error that can trigger very sharp corners (such as mixing short elements or small angular distances)
12143	SMC_CP_INTERNAL_EVAL_CACHE_ERROR	There is an internal error in the evaluation cache of the CP planner

WUXI XINJE ELECTRIC CO., LTD.

Address: 816 Jianshe West Road, Binhu District, Wuxi City, Jiangsu Province, China

Tel: 0510-85134136 Fax: 0510-85111290

Website: www.xinje.com Email: sales@xinje.com