COTRUST

M226ES motion controller user manual

◎V1.00

COTRUST TECHNOLOGIES CO., LTD.

Preface

Copyright Notice

Trademark Declaration

Disclaimers

Preface

Thank you for purchasing the COTRUST CTMC series M226ES PLC product from COTRUST TECHNOLOGIES CO., LTD.

Before using COTRUST CTMC series M226ES PLC, please read this manual to better understand the product's characteristics, apply it more safely, and fully utilize the rich features of this product.

Introduction to Manual Content

This manual mainly introduces the installation, debugging, and technical specifications of the COTRUST CTMC series M226ES PLC. The content includes:

Chapter	Name	Summary
Chapter 1	summary	Introduce the functional features, appearance, input-output wiring, system architecture, and other information of M226ES.
Chapter 2	CPU Technical Specifications	Introduce the technical specifications of M226ES.
Chapter 3	install	Introduce the installation requirements, installation dimensions, and installation steps of M226ES.
Chapter 4	Introduction to Programming Software and Working Components	Introduce the configuration software/programming software that is compatible with M226ES, hardware connections and communication settings, as well as programming operation instructions for the upper computer software.
Chapter 5	Communication Use Cases	Introduce communication application examples of M226ES.
Chapter 6	Axis configuration and electronic cam	Introduce the configuration of each axis, electronic cam, PLCopen library instructions, and the function of returning to the original.
Chapter 7	appendix	Introduce M226ES product ordering information, as well as after-sales maintenance information, etc.

Safety precautions

The personnel responsible for product installation and operation must undergo strict training, comply with relevant industry safety regulations, strictly follow the equipment precautions and special safety instructions provided in this manual, and operate the equipment according to the correct operating methods.

To prevent personal injury or equipment damage, the following instructions are made for the matters that must be followed. The potential harm and degree of damage caused by incorrect use of this product are indicated by relevant symbols.

	**Warning** The danger caused by failure to operate as required may result in personal injury or death.
	**Attention** The danger caused by failure to operate as required may result in mild or moderate personal injury and equipment damage.
	**Tips** Provide necessary supplements or explanations to the description of the operation.

Version History

List the updated content of the manual version upgrade and insert hyperlinks, which can directly redirect to specific modification areas for easy viewing of the modified content.

Release date	Edition	Revised content
September 2024	V0.10	• Temporary version

Catalogue

1 Summary

This chapter mainly provides an overview of the functions, inputs, and outputs of the COTRUST CTMC series products.

1.1 Product Overview

The CTMC series M226ES is a product launched through in-depth market demand research.

This machine integrates 16 EtherCAT bus axes and 10 pulse axes. Equipped with two Ethernet ports, supporting cascading (switch function), remote programming, debugging, monitoring, and data exchange. Comes with two RS485 channels, 24 point input and 16 point output (transistor drain type). Can be extended with an expansion board (supporting combinations of analog/RS485/CAN). The bus can expand up to 7 CTH200 series modules (digital input/output, temperature module, PID module, weighing module, motion control module).

1.1.1 Product Features

Stable and reliable

• User programs and data are permanently saved, Built in real-time clock.

• Multiple password protection, with a one-way download function for core programs, ensuring permanent confidentiality.

Strong communication ability

• Supports multiple communication protocols such as Ethernet communication, MPI, PPI, Freeport, Modbus, Profinet, etc

• Support remote programming, debugging, monitoring, and data exchange.

• Support network port firmware upgrade, convenient and fast.

High speed and large capacity

• The maximum data space can reach 64KB.

• Equipped with 10 high-speed input counters.

• 10 channels * 200kHz high-speed pulse output.

Rich operation and control functions

• Supports 16 axis EtherCAT bus axis and 10 channel pulse axis. Compatible with PLCopen standard instruction library, supporting functions such as single axis, shaft group, interpolation, electronic cam, electronic gear, etc.

• Support the development of the Follow Up Clipper command by COTRUST.

1.1.2 Product appearance

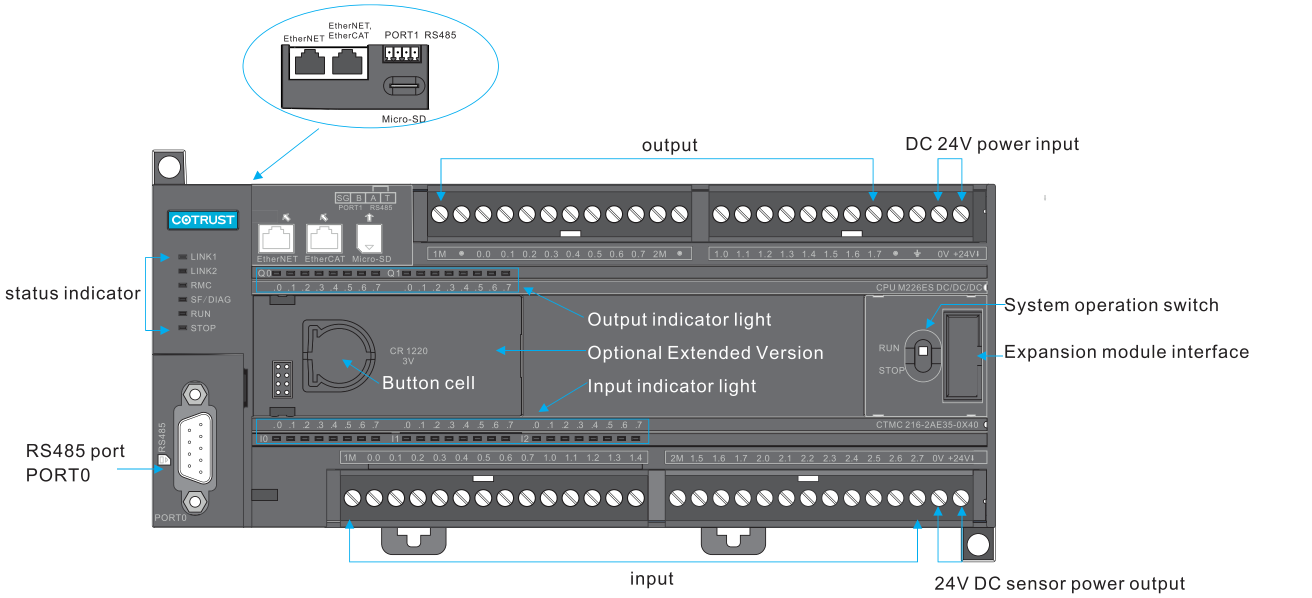


Fig1-1 Appearance diagram

1.2 Terminal wiring diagram

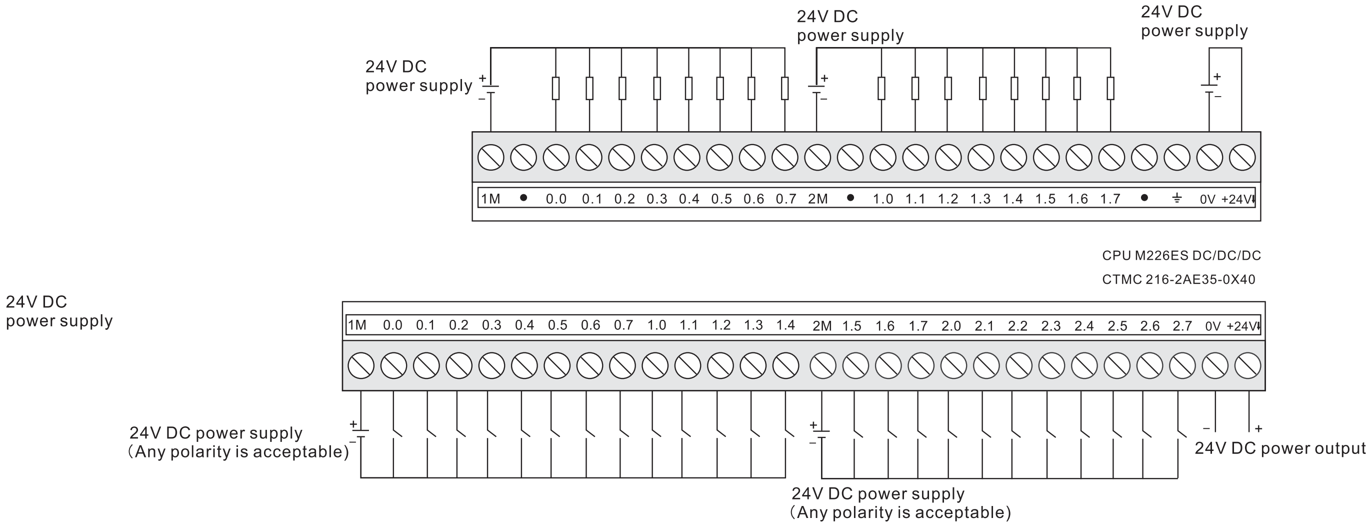


Fig1-2 Terminal wiring diagram

1.3 Communication interface terminal

Table1-1 RS485 communication interface (PORT0/FPORT0)

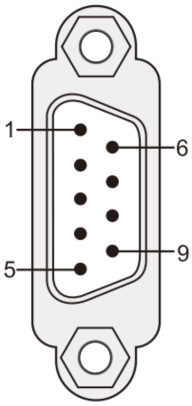
Connector	Pin number	PORT0/FPORT0（RS485）
	1	Chassis Ground
	2	+24V ground
	3	RS485 signal B/+
	4	Send Request RTS
	5	+5V ground
	6	+5V，100Ω
	7	+24V
	8	RS485 signal A/-
	9	NC
	Connector housing	Chassis Ground

Table1-2 RS485 (PORT1/FPORT1) terminal block

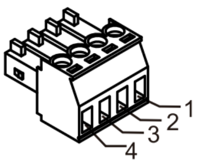
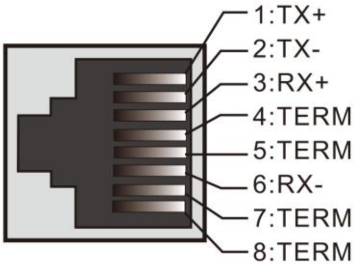
Connector	Pin number		PORT1/FPORT1（RS485）
	one	SG	Signal Ground
	two	B	RS485 signal B/+
	three	A	RS485 signal A/-
	four	T	Terminal resistance (effective with 3 short circuits)

Table1-3 EtherNET and EtherNET/EtherCAT communication interfaces

RJ45 communication port	Tag number	signal	Definition of Signal
	1	TX+	Data Send+
	2	TX-	Data Send-
	3	RX+	Receive Data+
	4	TERM	--
	5	TERM	--
	6	RX-	Receive Data-
	7	TERM	--
	8	TERM	--

1.4 System architecture

The architecture of CTMC M226ES automatic control system is shown in the following figure:

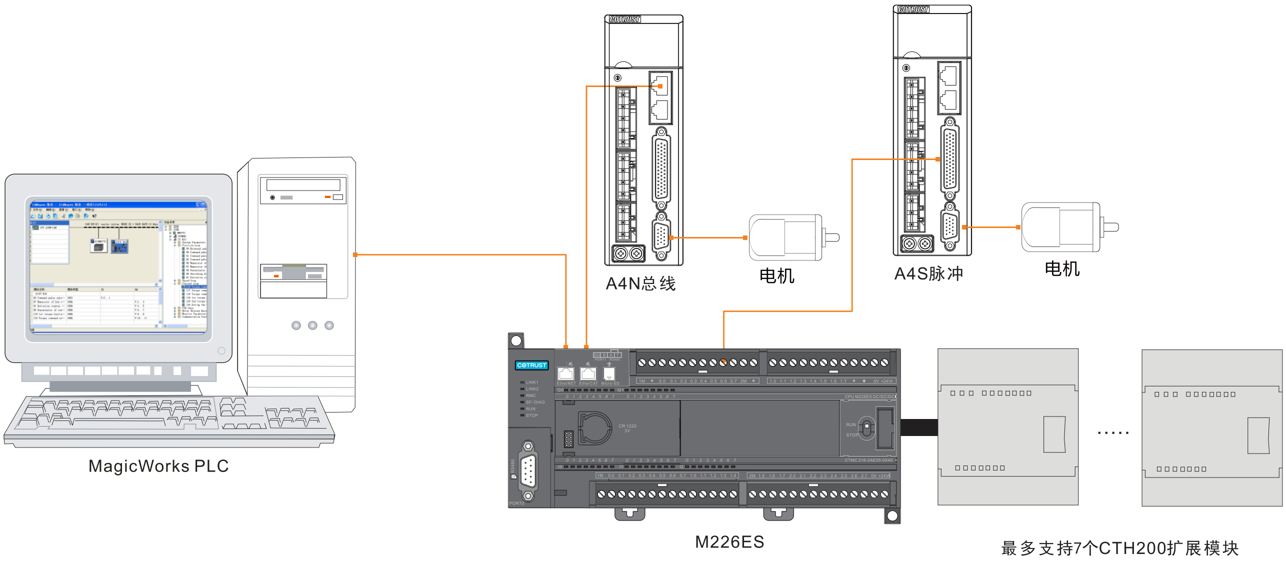


Fig1-3 system architecture

CPU Technical Specifications

2

This chapter mainly introduces the technical specifications of CTMC M226ES series products.

2.1 Basic performance parameters

Table2-1 Specification parameters

physical characteristics	
Dimensions (length x width x height)	196×80×62mm
Power consumption	20W (all communication interfaces communicate normally, with 7 modules)
Memory characteristics	
Program memory	Basic 200KB. Confidential library 1:48KB. Confidential Library 2:48KB
Data storage device	V Memory: 64KB, permanently retained
Real time clock characteristics	
Power down holding time (supercapacitor)	RTC: Approximately 112 hours (typical value)
Power down holding time (external battery)	RTC: Approximately 200 days (typical value)
General characteristics	
timer	The total is 2048 1ms: 36 pieces 10ms: 752 pieces 100ms: 1260 pieces <Note>Check the specific timer number in Chapter 2.5 Data storage device
Total number of counters	2048
accumulator	4
Memory storage location	8192 bits
Memory storage bit power down retention time	Permanently maintain
Local storage area (L)	64 bytes independent
Sequential control relay (S)	8192 bits
Time interruption	Two 1ms resolutions
Edge interruption	10 rising edges/or 10 falling edges (I0.0~I0.7, I1.0, I1.1)
Boolean operation execution time	≤0.05μs
Floating point operation execution time	≤1.5μs
Real time clock	built-in
Number of BD expansion board interfaces	1, supporting RS485 expansion board, CAN expansion board, analog expansion board

System indicator light	LINK1 indicator light (green): ON=Ethernet connection, flashing=transmission, OFF=disconnected	
	LINK2 indicator light (green): ON=EtherCAT connection, flashing=transmission, OFF=disconnection	
	RMC indicator light (green): ON=CPU successfully communicates with remote server, OFF: CPU fails to communicate with remote server or communication with remote server is prohibited	
	SF indicator light (red): ON=system error, OFF=normal DIAG indicator light (orange): ON=program control, OFF=normal	
	RUN indicator light (green): ON=running, OFF=stopping	
	STOP indicator light (orange): ON=stop, OFF=run	
RUN/STOP dialing code	RUN/STOP	
Battery interface	Support, power supply below 2.0V, alarm position SM195.0	
Programming card	support	
Memory card	TF card	
Function Description		
Basic Instruction Set	CTH200 Basic Instruction Set	
Programming tools	MagicWorks PLC	
Programming interface	RS485, Ethernet	
Online programming	support	
programming language	Statement table STL, ladder diagram LD, C language	
POU	Type/Quantity	Main program: 1 Subroutine: 1024 (0~1023) Interrupt program: 256 (0~255)
	Nesting Depth	From the main program: 8 subroutine levels Interrupt program: 8 subroutine levels
	FB/FC/DB functionality	Supports FC, but does not support FB/DB functionality
Power specifications		
Rated input voltage	24V DC	
Impulse current	12A at 28.8V DC	
Isolation (on-site and logical)	nothing	
Holding time (power failure)	More than 10ms	
Expansion bus+5V power output	1A	
BD expansion board bus+5V power output	200mA (with switch control)	
Communication+5V power output	10mA 100 Ω resistor	
Sensor+24V power output	Voltage range: 20.4~28.8V DC Rated current: 300mA Ripple noise: related to input power supply Isolation (sensors and logic): Not isolated	

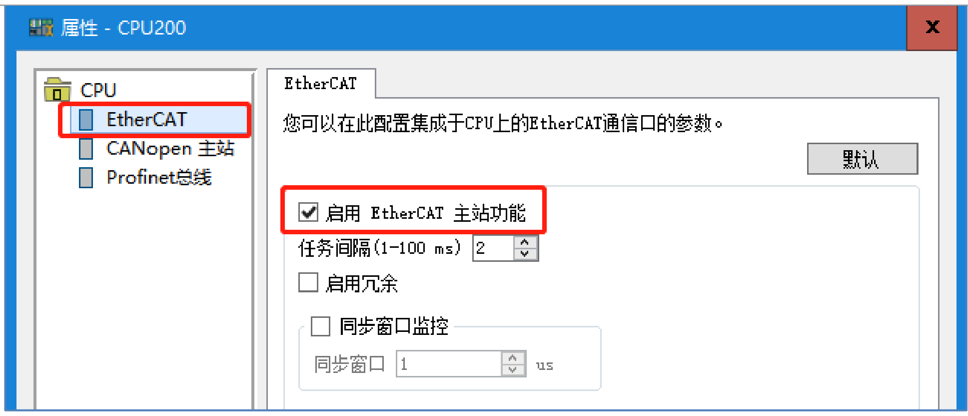
Power alarm	Power insufficient 15V alarm, alarm position SM195.1

2.2 CPU communication specifications

Table2-2 Integrated communication function

RS485 communication	
communication interface	2 RS485 communication ports PORT0: PPI/Freeport, Standard RS485 Level PORT1: PPI/Freeport, Standard RS485 Level After plugging in the RS485 expansion board and restarting, PORT0 is fixed as PPI and cannot be switched, and the free port FP0 is configured to the RS485 expansion board.
PPI baud rate (baud)	9.6kbps, 19.2kbps, and 187.5kbps
Free port baud rate (baud)	1.2kbps~115.2kbps
Maximum cable length per section	Using isolated relays: 1000 meters at a baud rate of 187.5kbps and 1200 meters at 38.4kbps
	Unused isolation relay: 50 meters
Maximum number of sites	32 stations per segment, 126 stations per network
Maximum number of main stations	32
Point to point (PPI master station mode)	Yes (NETR/NETW), up to 8 connections, with a maximum communication of 200 bytes per connection
MPI connection	(1PG/1OP) 8 in total, 2 reserved
quarantine	nothing
Ethernet communication	
communication interface	Two standard Ethernet ports, one of which is an EtherNET/EtherCAT shared port
Communication standards	Compliant with IEEE802.3
transmission speed	10Mbps/100Mbps adaptive
Adaptive Cross Connection	support
hardware interface	RJ45
Protocol type	Supports UDP_PPI, Modbus TCP, Socket, S7 protocol, and Profinet slave protocol, supports Ethernet communication between PLCs, and supports MiCo Ethernet programming. Note: For an example of using the Profinet slave protocol, please refer to the "COTRUST Built in Profinet Slave Function Manual"

collocation method	Download network blocks through MagicWorks for configuration
application interface	Supports MiCo services, UDP/PPI, TCP/Modbus, Socket programming, EtherCAT 402 axis
maximum connection	8 UDP/PPI without master-slave distinction
	8 TCP/Modbus without master-slave distinction
	16 Socket connections, 8 each for UDP and TCP
Maximum single transmission data	UDP/PPI can transmit up to 200 bytes at a time
	TCP/Modbus can transmit up to 240 bytes at a time
	Socket can transmit up to 512 bytes at a time
Network port startup time	5-10 seconds, affected by network environment.
indicator light	RMC ON: Communication between CPU and remote server is successful, OFF: Communication failure between CPU and remote server or prohibition of communication with remote server
	LINK1 ON: EtherNet network port hardware is connected flash: There is data exchange OFF: EtherNet network port hardware has been disconnected
Length of communication cable	100M. Cable type: CAT5e standard or above
IP address reset function	Press the running switch 5 times in a row within 2 seconds
Upgrade firmware functionality	Use MagicWorks PLC to remotely upgrade firmware from the company server via Ethernet
EtherNET/EtherCAT shared port	
communication interface	1 EtherCAT communication master interface, EtherCAT interface supports up to 16 slave connections
Baud rate	100Mbps
Protocol type	EtherCAT interface protocol
Maximum cable length per section	100m
Supported functions	Configure distributed clock, configure startup parameters, configure PDO parameters and mapping, configure bus cycle period Configure startup to check supplier ID and product ID
Indicator light (green)	ON: The network port hardware is connected flash: There is data exchange OFF: The network port hardware has been disconnected
quarantine	Communication port isolation
collocation method	The function switching of EtherNET/EtherCAT communication port is achieved through the EtherCAT communication port parameter configuration in the MagicWorks PLC hardware configuration (as shown in the figure below). Checking "Activate EtherCAT Master Station Function" will enable EtherCAT communication, otherwise it will enable EtherNET switch function.

<table>
<tr><td></td><td colspan="7">
</td></tr>
<tr><td colspan="8">CAN bus</td></tr>
<tr><td>communication interface</td><td colspan="7">1 CAN bus interface (implemented through CAN expansion board CTH2-CAN-01S2-EB)</td></tr>
<tr><td>communication protocol</td><td colspan="7">CANopen, CANfree protocol</td></tr>
<tr><td colspan="8">CANopen protocol</td></tr>
<tr><td>Communication speed (kbps)</td><td>1000</td><td>800</td><td>500</td><td>250</td><td>125</td><td>50</td><td>20</td></tr>
<tr><td>Maximum communication distance (m)</td><td>25</td><td>50</td><td>100</td><td>250</td><td>500</td><td>1000</td><td>2500</td></tr>
<tr><td>Maximum number of sites</td><td colspan="7">127</td></tr>
<tr><td>Site address range</td><td colspan="7">1-127</td></tr>
<tr><td>Maximum number of subordinate stations</td><td colspan="7">32</td></tr>
<tr><td>collocation method</td><td colspan="7">CAN configuration block, EDS file</td></tr>
<tr><td rowspan="3">Maximum Number of Visits</td><td></td><td colspan="3">Byte count</td><td colspan="3">Memory starting address</td></tr>
<tr><td>input</td><td colspan="3">64</td><td colspan="3">IB64</td></tr>
<tr><td>output</td><td colspan="3">64</td><td colspan="3">QB64</td></tr>
<tr><td rowspan="3">Maximum analog access</td><td></td><td colspan="3">Number of channels</td><td colspan="3">Memory starting address</td></tr>
<tr><td>input</td><td colspan="3">162</td><td colspan="3">AIW256</td></tr>
<tr><td>output</td><td colspan="3">162</td><td colspan="3">AQW256</td></tr>
</table>

2.3 CPU input/output specifications

Table2-3 I/O characteristics

Local digital input points	24
Input type	Leakage type/source type
Local digital output points	16
type of output	Leakage type
Digital I/O mapping area	I memory 16KB, Q memory 16KB (including CAN communication dedicated image area)
Simulate I/O image area	I memory 16KB, Q memory 16KB including CAN communication dedicated image area)
Allow maximum expansion of I/O modules	7
Pulse capture input	I won't support it
High-speed counter	Total: 10
	Single phase counter: 10 × 200kHz
	Dual phase counter: 10 × 100kHz

Note: Load conditions required for high-speed counting of 200k: current greater than 3mA and voltage below 30V.

Table2-4 Digital input and output specifications

Digital input characteristics	
Local integrated digital input points	24
Input type	Leakage type/source type
Rated voltage	24V DC
Input voltage range	20.4~28.8V DC
Surge voltage	35V DC, lasting for 0.5s
Logic 1 signal (minimum) Logic 0 signal (maximum)	15V DC，3mA 5V DC，1mA
Connect 2-wire proximity switch sensor (BERO) Allowable leakage current (maximum)	1mA
Input filtering	Configurable, supports the following parameters: 0.2us，0.4us，0.8us，1.6us，3.2us，6.4us，12.8us； 0.2ms，0.4ms，0.8ms，1.6ms，3.2ms，6.4ms，12.8ms； Default is 6.4ms

Isolation (on-site and logical) Isolation group		500V AC, 1 minute
Simultaneously connected inputs		twenty-four
Maximum cable length shield Unshielded		500 meters (standard input) 50 meters (high-speed counter input) 300 meters (standard input)
Digital output characteristics		
Local integrated digital output points		16
type of output		Leakage type
Output rated voltage		DC：24V
Output voltage range		DC：5～30V
Maximum current per common terminal		6A
Output current (maximum)		0.5A
Leakage current (maximum)		10uA
Surge current		8A，100ms
Lamp load		5W
On resistance		Typical value 0.15 Ω, maximum 0.32 Ω
Output delay (maximum) Disconnect to Connect/Connect to Disconnect		Disconnected to connected: 0.2us (16 Q points) Connect to disconnect: 2us (16 Q points)
High speed pulse output		10 channels * 200kHz (Pulse/Dir)
Output frequency (maximum)		200kHz（Q0.0~Q1.7）
Mechanical lifespan (no load)		No
Contact life (rated load)		No
Simultaneously connected outputs		16
Two outputs in parallel		Not support
Maximum cable length	shield Unshielded	500 meters 150 meters

2.4 System Interruption Event

Table2-5 System interrupt number

interrupt number	Interrupt Name	Trigger condition	Remarks
0	I0.0 rising edge interruption	I0.0 detected rising edge	--

1	I0.0 falling edge interruption	I0.0 detected a falling edge	--
2	I0.1 rising edge interruption	I0.1 detected rising edge	--
3	I0.1 falling edge interruption	I0.1 detected a falling edge	--
4	I0.2 rising edge interruption	I0.2 detected rising edge	--
5	I0.2 falling edge interruption	I0.2 detected a falling edge	--
6	I0.3 rising edge interruption	I0.3 detected rising edge	--
7	I0.3 falling edge interruption	I0.3 detected a falling edge	--
8	PORT0 receives data	PORT0 received a byte of data	--
9	PORT0 transmission completed	PORT0 has sent the data in the buffer	--
10	Timer Interrupt 0	The timing 0 signal given by FPGA arrives	--
11	Timer Interrupt 1	The timing 1 signal given by FPGA arrives	--
12	HSC0 set value interrupt	Count value=set value	--
13	HSC1 set value interrupt	Count value=set value	--
14	HSC1 direction interruption	Change of counting direction	--
15	HSC1 external reset interrupt	External reset signal is valid	--
16	HSC2 set value interrupt	Count value=set value	--
17	HSC2 direction interruption	Change of counting direction	--
18	HSC2 external reset interrupt	External reset signal is valid	--
19	PTO 0 output completed	PTO given pulse output completed	Unsupported
20	PTO 1 output completed	PTO given pulse output completed	Unsupported
21	T32	T32 time is up	--
22	T96	T96 time is up	--
23	PORT0 information reception completed	PORT0 completes the reception according to the set conditions	--
24	PORT1 information reception completed	PORT1 completes the reception according to the set conditions	--
25	PORT1 character reception	PORT1 received a byte of data	--
26	PORT1 transmission completed	PORT1 has sent the data from the buffer	--
27	HSC0 direction interruption	Change of counting direction	--
28	HSC0 external reset interrupt	External reset signal is valid	--
29	HSC4 set value interrupt	Count value=set value	--
30	HSC4 direction interruption	Change in counting direction	--
31	HSC4 external reset interrupt	External reset signal is valid	--
32	HSC3 set value interrupt	Count value=set value	--

33	HSC5 set value interrupt	Count value=set value	--
38	HSC3 direction interruption	Change in counting direction	--
39	HSC5 direction interruption	Change in counting direction	--
40	HSC3 external reset interrupt	External reset signal is valid	--
41	HSC5 external reset interrupt	External reset signal is valid	--
42	HSC0 captures interrupts	External capture signal is effective	--
43	HSC1 captures interrupts	External capture signal is effective	--
44	HSC2 captures interrupts	External capture signal is effective	--
45	HSC3 Capture Interrupt	External capture signal is effective	--
46	HSC4 Capture Interrupt	External capture signal is effective	--
48	I0.4 rising edge interruption	I0.4 detected rising edge	--
49	I0.4 falling edge interruption	I0.4 detected a falling edge	--
50	I0.5 rising edge interruption	I0.5 detected rising edge	--
51	I0.5 falling edge interruption	I0.5 detected a falling edge	--
52	I0.6 rising edge interruption	I0.6 detected rising edge	--
53	I0.6 falling edge interruption	I0.6 detected a falling edge	--
54	I0.7 rising edge interruption	I0.7 detected rising edge	--
55	I0.7 falling edge interruption	I0.7 detected a falling edge	--
56	I1.0 rising edge interruption	I1.0 detected rising edge	--
57	I1.0 falling edge interruption	I1.0 detects a falling edge	--
58	I1.1 Rising edge interruption	I1.1 Detected rising edge	--
59	I1.1 Downward edge interruption	I1.1 Detected falling edge	--
60	HSC6 set value interrupt	Count value=set value	--
61	HSC6 direction interruption	Change in counting direction	--
62	HSC6 external reset interrupt	External reset signal is valid	--
63	HSC6 captures interrupts	External capture signal is effective	--
64	HSC7 set value interrupt	Count value=set value	--
65	HSC7 direction interruption	Change of counting direction	--
66	HSC7 external reset interrupt	External reset signal is valid	--
67	HSC7 captures interrupt	External capture signal is effective	--
68	HSC8 set value interrupt	Count value=set value	--
69	HSC8 direction interruption	Change of counting direction	--

70	HSC8 external reset interrupt	External reset signal is valid	--
71	HSC8 Capture Interrupt	External capture signal is effective	--
72	HSC9 set value interrupt	Count value=set value	--
73	HSC9 direction interruption	Change of counting direction	--
74	HSC9 external reset interrupt	External reset signal is valid	--
75	HSC9 capture interrupt	External capture signal is effective	--

2.5 Data memory

Table2-6 Data storage device

Model	M226ES (Order Number: CTMC 216-2AE35-0X40)
Digital Input Mapping Area (I)	
Address range	I0.0~I16383.7
Byte address range	IB0~IB16383
Word address range	IW0~IW16382
Double letter address range	ID0~ID16380
Digital output image area (Q)	
Address Range	Q0.0~I16383.7
Byte address range	QB0~IB16383
Word address range	QW0~QW16382
Double letter address range	QD0~QD16380
Access Properties	Support immediate access/direct access/indirect access
Maintain attributes	Does not support power down maintenance
Analog Input Mapping Area (AI)	
Word length of each channel	16 bits
Word address range	AIW0~AIW8190
Analog output image area (AQ)	
Word length of each channel	16 bits
Word address range	AQW0~AQW8190
Access Properties	Support immediate access/direct access/indirect access
Maintain attributes	Does not support power down maintenance
Variable memory area (V)	
Capacity (in bytes)	64K

Address Range	V0.0~V65535.7
Byte address range	VB0~VB65535
Word address range	VW0~VW65534
Double letter address range	VD0~VD65532
Access Properties	Support direct/indirect access
Maintain attributes	Can be configured to maintain full or partial power failure
Special Memory Area (SM)	
capacity	64K
Address Range	SM0.0~SM65535.7
Byte address range	SMB0~SMB65535
Word address range	SMW0~SMW65534
Double letter address range	SMD0~SMD65532
Access Properties	The first 30 bytes of the access attribute are read-only and support direct/indirect access
Special Register SM	
SMB0	System status bit
SMB1	Instruction execution status bit
SMB2	Free mouth receives characters
SMB3	Free port verification error
SMB4	Interrupt queue overflow, runtime program error, interrupt enabled, forced free port transmitter
SMB5	I/O status
SMB6	CPU identification register
SMB7	retain
SMB8~SMB21	I/O module identification and error register
SMW22~SMW26	Scanning time
SMB30，SMB130	Free port control register
SMB31~SMB33	retain
SMB34	The time interval value of timed interrupt 0
SMB35	The time interval value of timed interrupt 1
SMB66~SMB85	retain
SMB86~SMB94 SMB186~SMB194	Free mouth receiving information control
SMW98	I/O expansion bus communication error
SMB110	Used to activate the forced upload protection function

SMB111	Programming Card Coverage Options 0: Covering program blocks and data blocks. 1: Do not cover program blocks, cover data blocks. 2: Covering program blocks, not data blocks. 3: Do not cover program blocks and data blocks. 4: Same as when it is 0
SMB112	Display CPU hardware platform information. 0x82 represents -2 platform. 0x85 represents V5 platform. 0x86 represents V6 platform.
SMB113	Always display the current CPU model correctly 16#25： M226ES
SMB114	Expansion board module type 0x19：2AI/1AQ（CTH2-AMS-03S1-EB） 0x1E：4AI/2AQ（CTH2-AMS-06S1-EB / CTH2-AMS-06S2-EB） 0x8A：RS485（CTH2-485-01S1-EB） 0x8D：CAN V2（CTH2-CAN-01S2-EB）
SMB115	0x0: The module has no errors. 0x255: Expansion board access error. Other: Internal diagnosis of expansion board
SMW116~SMW122	Expansion board analog input mapping area
SMW124~SMW126	Expansion board analog output mapping area
SMB131	retain
SMB166~SMB185	retain
SMB195	SMB195.0:1- Battery not installed or low power. 0- The battery level is normal SMB195.1:1- Power supply below 15V SMB195.6:0- Mico remote not connected. 1. MiCo remote connected
SMB197	SMB197.0:1- CANFree has been started
SMB198	retain
SMB200~SMB549	The status of the intelligent module can be found in Table 2-7 for details on the storage of intelligent expansion modules from SMB200 to SMB549
SMB550~SMB649	For CANopen communication, please refer to Table 2-8CAN Station Status Table for details
SMB2390	SM2390.0: EtherCAT initialization completion bit. 0- Not initialized completed, 1- Initial initialization completed SM2390.1: EtherCAT redundant activation bit. 0-Redundancy not activated, 1- Redundancy activated. At this time, the connection needs to be checked SM2390.2~SM390.7: Reserved
SMB2391	retain
SMD2392	EtherCAT task cycle execution time
SMD2396	EtherCAT task cycle interval time
SMB2400	The number of EtherCAT slave stations found
SMB2401	EtherCAT error: 0: No errors 1: Configuration parameter error 2: No sub station found 3: State switching error 4: An error occurred while writing the configuration 5: Wrong number of stations 6: Station mismatch
SMB2402	Station 1 status: 0: No connection 1: Initialization status 2: Pre operation status 4: Safe operation 8: Operation status

	16 # 80: Product ID mismatch 16 # 81: Manufacturer ID mismatch 16 # 2: SDO write error Other: Incorrect Status
SMB2403~SMB2417	From station 2 status to station 16 status. The status word indicates that the content is consistent with SMB2402
SMB2418~SMB65535	retain
Internal memory area (M)	
Capacity (bits)	8192 bits (1024 bytes)
Address Range	M0.0~M1023.7
Byte address range	MB0~MB1023
Word address range	MW0~MW1022
Double letter address range	MD0~MD1020
Access Properties	Support direct/indirect access
Maintain attributes	Can be configured to maintain full or partial power failure
Local variable area (L)	
Capacity (in bytes)	64 bytes
Address Range	L0.0~L63.7
Byte address range	LB0~LB63
Word address range	LW0~LW62
Double letter address range	LD0~LD60
Access Properties	Support direct access
Maintain attributes	Maintain subroutines at the same call location from startup to shutdown, without power failure
Accumulator Register (AC)	
capacity	4 of them
Address Range	I won't support it
Byte address range	AC0~AC3
Word address range	AC0~AC3
Double letter address range	AC0~AC3
Access Properties	Support direct access
Maintain attributes	Does not support power down maintenance
Sequential control relay (S)	
Capacity (bits)	8192 bits (1024 bytes)
Address Range	S0.0~S1023.7
Byte address range	SB0~SB1023

Word address range	SW0~SW1022			
Double letter address range	SD0~SD1020			
Access Properties	Support direct/indirect access			
Maintain attributes	Does not support power down maintenance			
Timer (T)				
Type	**Resolving power**	**Number**	**Number**	**Maximum timing value**
Memory connected delay timer TONR	1ms	2	T0，T64	32.767s
	10ms	8	T1~T4，T65~T68	327.67s
	100ms	54	T5~T31，T69~T95	3276.7s
Turn on delay timer TON/off delay timer TOF	1ms	34	T32，T96，T256~T287	32.767s
	10ms	744	T33~T36，T97~T100，T288~T1023	327.67s
	100ms	1206	T37~T63，T101~T255，T1024~T2047	3276.7s
Access Properties	The timing register can be accessed directly/indirectly, while the status bit can only be accessed directly			
Maintain attributes	Can configure the power down hold attribute for the current timing value, but the status bit cannot be held			
Counter (C)				
number	two thousand and forty-eight			
restart at	Counting Up/Counting Down/Counting Up/Counting Down			
Maximum count value	thirty-two thousand seven hundred and sixty-seven			
Access Properties	The counting register can be accessed directly/indirectly, while the status bit can only be accessed directly			
Maintain attributes	The power down hold attribute of the current count value can be configured, but the status bit cannot be held			

Table2-7 Information table for reserved storage intelligent expansion module from SMB200 to SMB549

0-slot intelligent module	**1 slot intelligent module**	**2-slot intelligent module**	**3-channel intelligent module**	**4-channel intelligent module**	**5-slot intelligent module**	**6-slot intelligent module**	**Function Description**
SMB200～215	SMB250～265	SMB300～315	SMB350～365	SMB400～415	SMB450～465	SMB500～515	Module name (16 ASCII characters)
SMB216～219	SMB266～269	SMB316～319	SMB366～369	SMB416～419	SMB466～469	SMB516～519	S/W revision number (4 ASCII characters)
SMW220	SMW270	SMW320	SMW370	SMW420	SMW470	SMW520	error code
SMB222～249	SMB272～299	SMB322～349	SMB372～399	SMB422～449	SMB472～499	SMB522～549	Information related to specific module types

Table2-8 CAN Station Status Table

	Byte count	Address allocation	Description of Status Values
SMB550~ SMB649 (Status)	one hundred	SMB550: CAN communication status of the main station (CPU itself)	0x00: Initialization 0x01: Disconnected 0x04: Stop 0x05: Running 0x7f: Pre run 0xff: Configuration data error
		SMB551-SMB582: CAN communication status of the 1st to 32nd slave stations (arranged in ascending order of Node ID)	0x00: Initialization 0x01: Disconnected 0x04: Stop 0x05: Running 0x7f: Pre run 0x7: Configuration parameter error
		SMB583-SMB614: Operation alarm status of the 1st to 32nd slave stations (arranged in ascending order of Node ID)	Bit7: Comparison result between configuration IO information and actual situation of 277C 0- No errors 1- There is an error bit6-bit0： New diagnostic message for the 277C extended IO module: bit0 represents the first module, and bit6 represents the seventh module 0- No alarm 1- There is an alarm (The above is the definition of 277C slave station, other slave stations are defined separately)
		SMB615-SMB649	retain

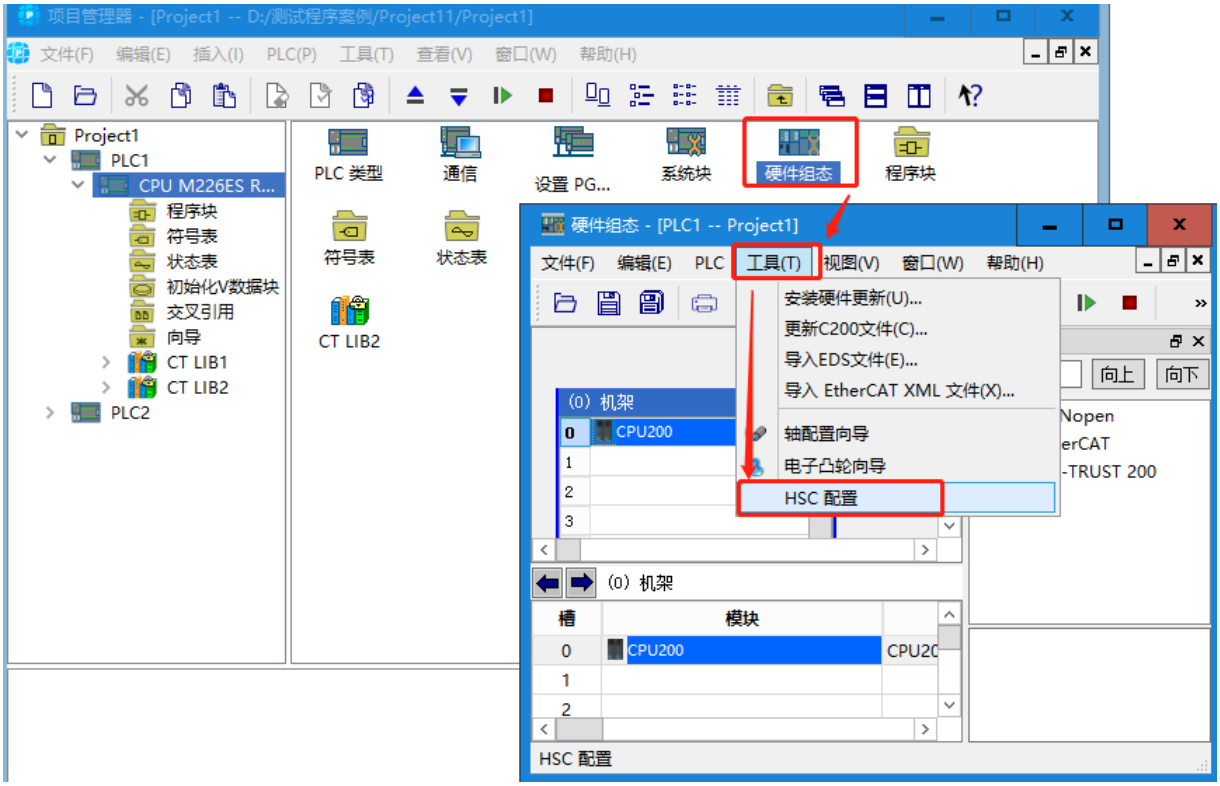
Tips

SMB550-SMB649 is a read-only storage area that users cannot write to.

2.6 High-speed counter

M226ES comes with 10 high count counters (HSC0~HSC9), all of which support 5 counting modes.

The high-speed counting input can be configured arbitrarily. Configure the HSC input point in Magic Works PLC [Hardware Configuration] → [Tools] → [HSC Configuration]. Note: The same input point cannot be applied to multiple high-speed counters, it can only be applied to one high-speed counter. Any unconfigured input can be arbitrarily configured as any input signal of the counter.



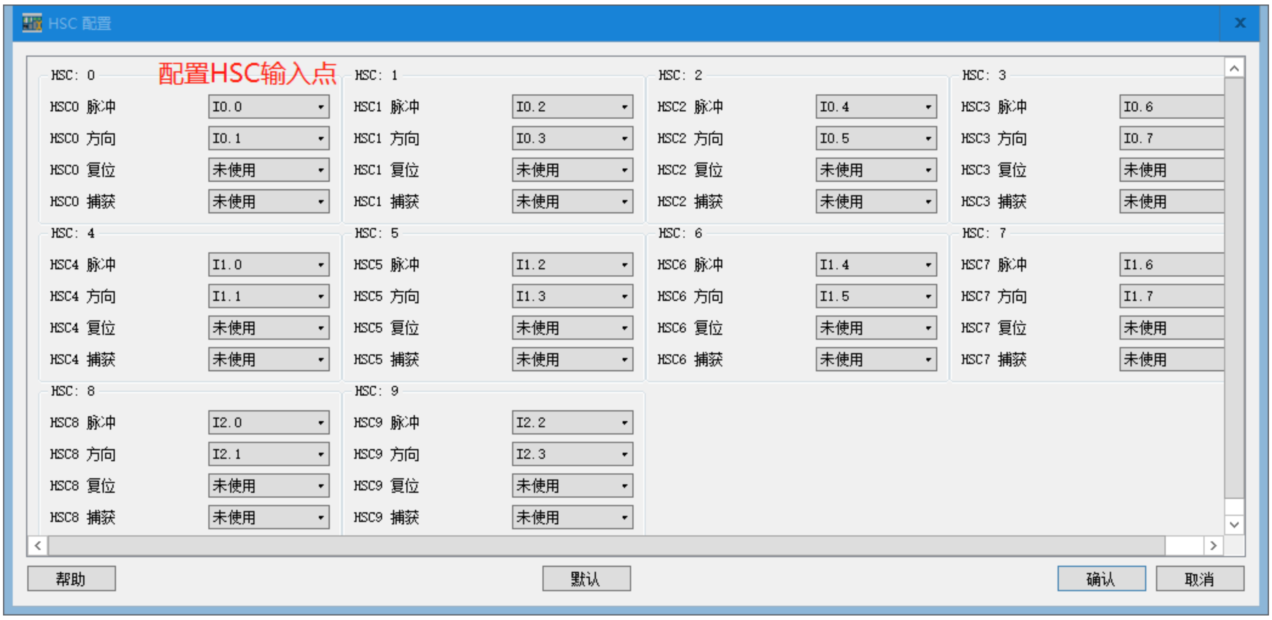


Table2-9 Counting mode

Counting mode		Configurable input ports			
Mode 0	Single phase counter with internal directional control	Clock	--	reset	capture
Mode 1	Single phase counter with external direction control	Clock	direction	reset	capture
Mode 2	Two phase counter with up and down counting clock	Increase Count	down counter	reset	capture
Mode 3	A/B phase orthogonal counter	Clock A	Clock B	reset	capture
Mode 4	Internal counting	--	--	--	capture

Tips

1. The difference between internal counting and external counting is that internal counting is directly counted through logic, without the need for wiring or occupying point I. External counting is initiated by an external input signal.

2. All modes support capture function, which means locking the current count value through an external input signal. The HSC-200 instruction can set a falling edge latch or a rising edge latch. When the corresponding capture interrupt is opened during the latch, an interrupt can be generated.

3. When using the capture function, it is necessary to set the latch window through the HSC_SETPARAMETER command, which represents the pulse interval generated by two consecutive valid locks. If set to 0, there will be no window interval, and the current count value will be locked upon detecting the capture signal. If set to greater than 0, the next lock operation will only be initiated after the current effective lock is generated and the HSC count reaches or exceeds the set value. Within the range of less than the set pulse number, there will be no response to the capture signal.

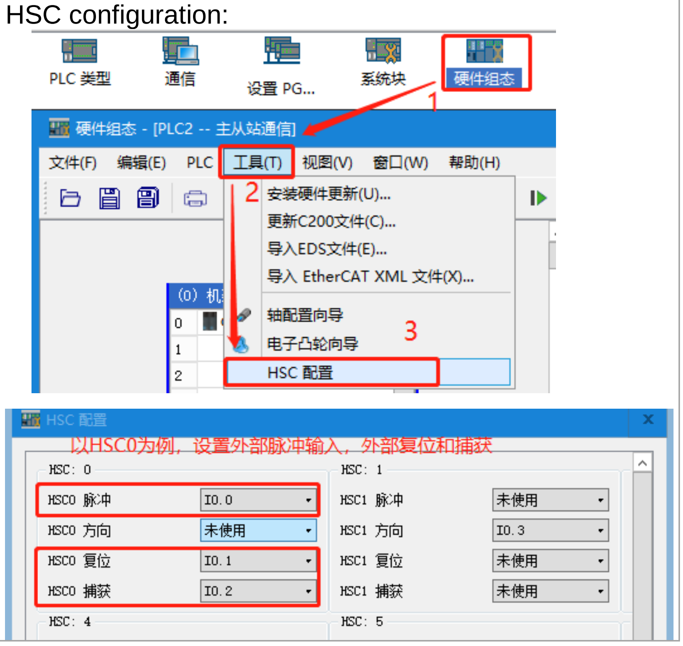
4. Forcing point I cannot trigger the capture function.
Counting unit: number of pulses.

2.6.1 Counting mode description

Mode 0 (single-phase counter with internal direction control)

Explanation: Mode 0 only has one pulse input for counting. The counting direction is set through the HSC-200 instruction, and the counter is reset and the count value is captured through external input.

<table>
<tr><th>Setting of counting direction</th><th>Pulse, reset, and capture settings</th></tr>
<tr><td>
Configure the counting direction through the CTRL control word of the HSC-200 instruction.
BIT0: Reset level, 0-High level reset, 1-Low level reset.
BIT1: Capture the effective edge of the trigger signal for the mode, with 0-falling edge being effective and 1-rising edge being effective.
BIT2: Orthogonal counting selection, 0-4x multiples, 1-1x multiples.
BIT3: Counting direction, 0-count down, 1-count up.
BIT4: Counting direction update, 0-no update, 1-update.
BIT5: Preset value update, 0-no update, 1-update.
BIT6: Current value updated, 0-no update, 1-update.
BIT7: Count enabled, 0-disable, 1-enable.
</td><td>
Configure pulse input, reset, and capture signals in HSC configuration:



</td></tr>
</table>

For example, taking the high-speed counter HSC0 as an example, I0.0 is used as the pulse input terminal, and the HSC-200 instruction determines the counting direction and preset value. When the count value (CV) reaches the preset value (PV), an interrupt will be triggered, which means that when CV=PV, the interrupt will be executed.

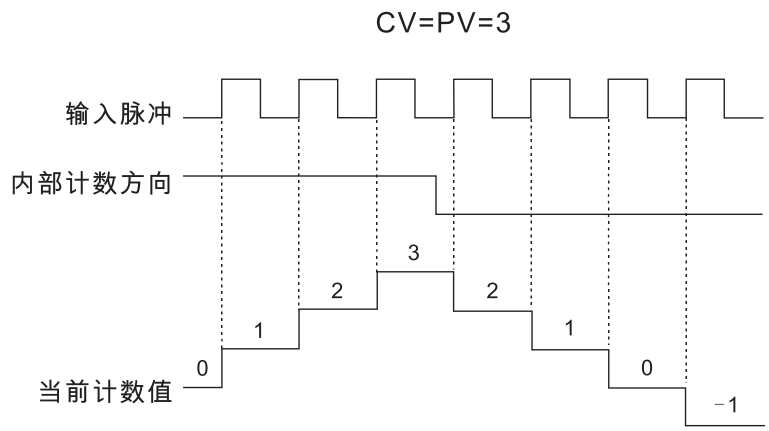
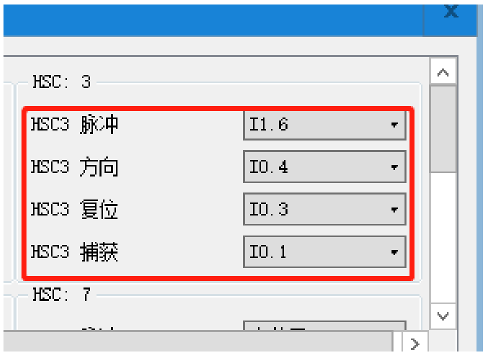


Fig2-1 Mode 0 Count Description

Mode 1 (single-phase counter with external direction control)

Explanation: One pulse input counts, and the external input pulse determines the counting direction. The counter is reset and the count value is captured through the external input. HSC-200 instruction sets preset values.

Configure pulse input, counting direction, reset, and capture signals in HSC configuration:



For example, taking the high-speed counter HSC3 as an example, I1.6 is used as the pulse input terminal, and I0.4 controls the counting direction. Assuming I0.4 high level is active, it counts up when I0.4 input high level and counts down when I0.4 input low level. When the count value (CV) reaches the preset value (PV), an interrupt will be triggered.

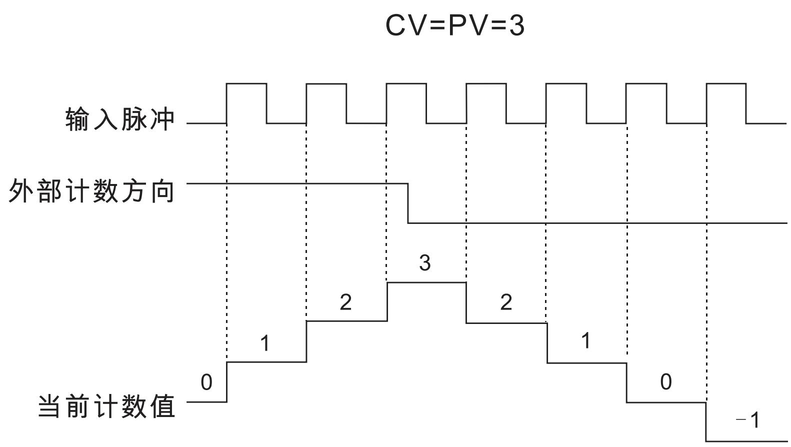
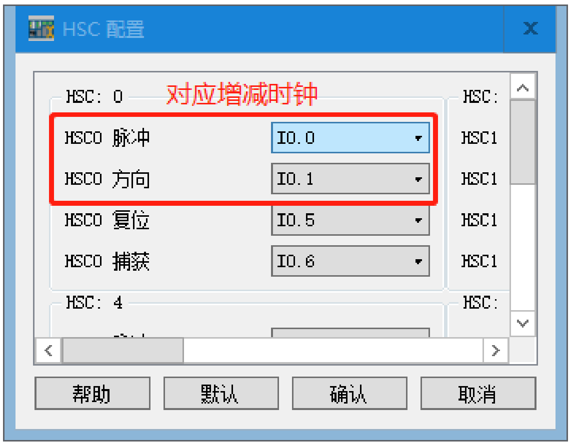


Fig2-2 Mode 1 Counting Description

Mode 2 (Two phase counter with up/down counting clock)

Explanation: Two pulse inputs, one is an up count input and the other is a down count input, reset the counter and capture the count value through external input.

Configure pulse input, counting direction, reset, and capture signals in the "HSC configuration". Configure the pulse and direction to correspond to two pulses, with "HSC pulse" corresponding to increasing clock count and "HSC direction" corresponding to decreasing clock count.



For example, taking HSC0 as an **example**, the high-speed counting mode is Mode 2. I0.0 is used as the clock up pulse input, I0.1 is used as the counter down pulse input, and the HSC-200 instruction sets the preset value of the counter.

When using Mode 2, if the time interval between the rising edge of the up count input and the rising edge of the down clock input is less than 0.3us, the high-speed counter will consider that the up count and down count occur simultaneously. If this happens, the current value remains unchanged and the counting direction remains unchanged. As long as the time interval between the rising edge of the clock up input and the rising edge of the clock down input is greater than 0.3us, the high-speed counter captures each event separately. In both of the above situations, no errors will occur and the counter will maintain the correct current value.

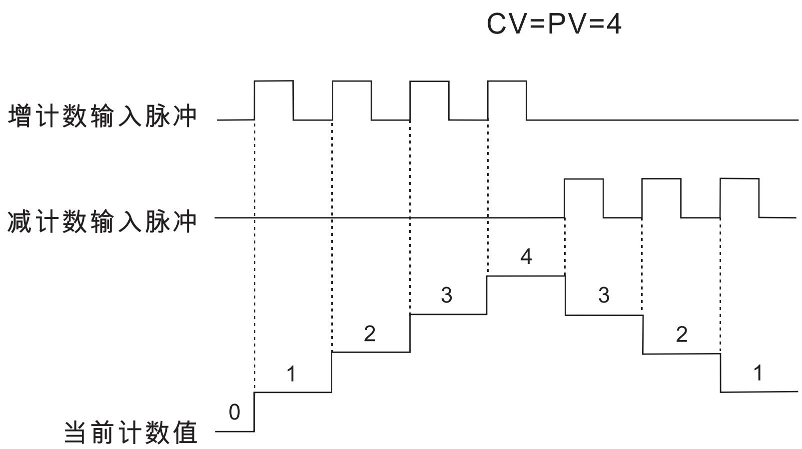
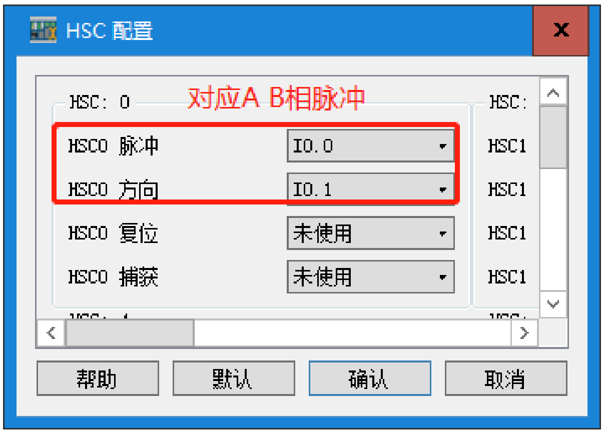


Fig2-3 Mode 2 Counting Instructions

Mode 3 (A/B phase orthogonal counter)

Explanation: Two pulse inputs, one is A-phase pulse input and the other is B-phase pulse input. A. B pulses differ by 90 degrees, which is a quarter cycle.

Configure AB phase pulses in "HSC configuration", where "HSC pulse" corresponds to A-phase pulses and "HSC direction" corresponds to B-phase pulses.



For example, the A-phase pulse input leads the B-phase pulse by 90 °, and the counting direction is increasing. The A-phase pulse input lags behind the B-phase pulse by 90 °, and the counting direction is countdown.

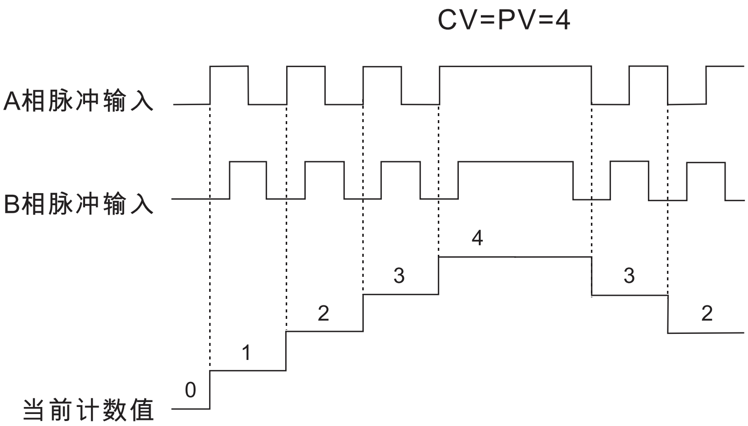


Fig2-4 Counting Explanation for Mode 3 (1x Orthogonal Mode)

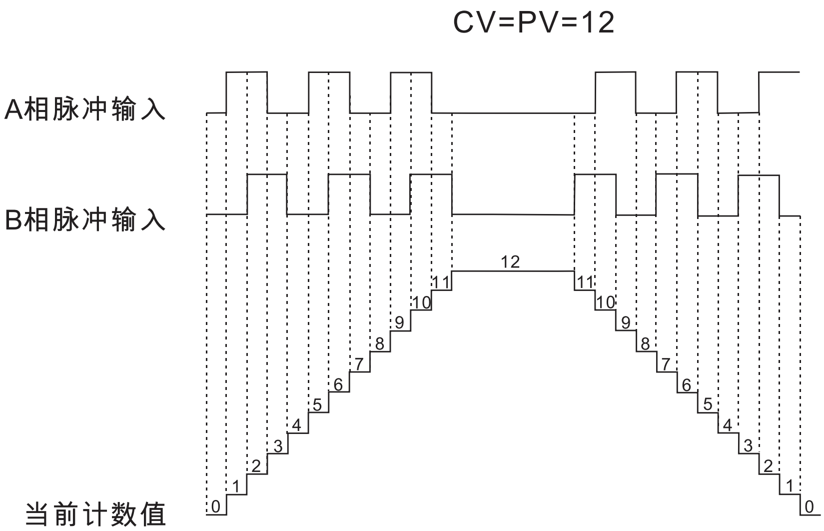


Fig2-5 Counting Explanation for Mode 3 (4x Orthogonal Mode)

Mode 4 (Internal Counting Mode)

One pulse input, HSC-200 instruction determines the counting direction, directly binds the internal pulse output channel. When using this mode, HSC defaults to counting axes with the same number, and the corresponding relationship is as follows:

HSC0 corresponds to Axis0, HSC1 corresponds to Axis1... HSC9 corresponds to Axis9.

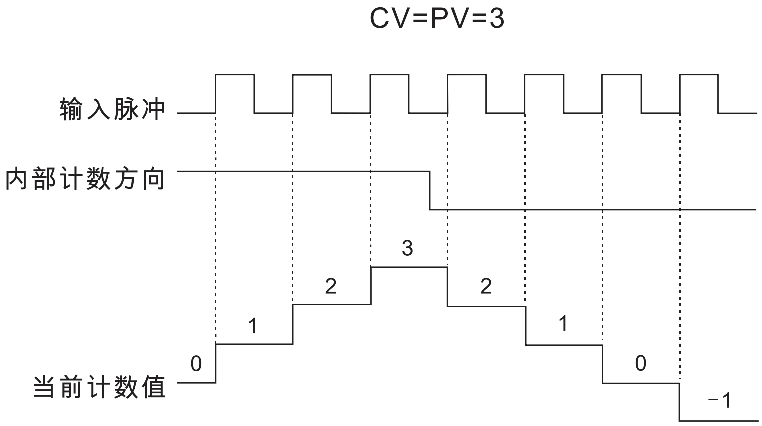


Fig2-6 Mode 4 (Internal Direct Counting Mode)

2.6.2 HSC library instruction

HSC-200 Set Counter

Set the counting direction, current value, preset value, etc. of the counter through the HSC-200 instruction.

Parameter Name	Input/Output	Parameter description	type	Numerical Range	remarks
CH_ADR	IN	Channel address	BYTE	0~9	0：HSC0 1：HSC1 **…..** 9：HSC9
CTRL	IN	Control word	BYTE	--	BIT0: Reset level, 0-High level reset, 1-Low level reset BIT1: Capture the effective edge of the trigger signal for the mode, with 0-falling edge being effective and 1-rising edge being effective. BIT2: Orthogonal counting selection, 0-4x multiples, 1-1x multiples. BIT3: Counting direction, 0-count down, 1-count up BIT4: Counting direction update, 0-no update, 1-update BIT5: Preset value update, 0-no update, 1-update BIT6: Current value updated, 0-no update, 1-update BIT7: Count enabled, 0-disable, 1-enable
PV_NUM	IN	Number of preset values	BYTE	0~100	--
PV_TABLE	IN	Preset value table	DWORD	--	Pointer, used to store preset values. Point to the first memory address where the PV table is stored, with each PV occupying 4 bytes (DW), and up to 100 PV values can be set. **be careful:** 1. When counting, as long as CV equals any value in PV_TABLE, the CV=PV interrupt can be triggered. 2. If the condition is met again after counting up and counting down, the interrupt CV=PV will still be triggered. If there is a 0 in the preset table, the initial value of the counter initialization will also be 0 and will not enter the interrupt.

					3. The output value of HSCGETPV is the PV value at the last time CV=PV. After triggering the HSC-200 instruction, the preset value table will update the HSCGETPV output value to zero until the next triggering condition is met and PV will be output.
CV	IN	Current value	DINT	--	Set initial CV value
STA	OUT	Return status	BYTE	0~255	0：OK 1: Parameter error 2: Access module error

HSC_SETMODE Set counter mode

Parameter Name	Input/Output	Parameter description	type	Numerical Range	remarks
MOD_ADR	IN	retain	BYTE	--	--
CH_ADR	IN	Channel address	BYTE	0~9	0：HSC0 1：HSC1 ..... 9：HSC9
MODE	IN	Counting mode	BYTE	0~4	0: Single phase counter with internal direction control 1: Single phase counter with external direction control 2: Two phase counter with up and down counting clock 3: A/B phase orthogonal counter 4: Internal direct connection counting mode
STA	OUT	Return status	BYTE	0~255	0：OK 1: Parameter error 2: Access module error

HSCGETCV retrieves the current count value

Note: When obtaining the current count value, it is also possible to read the high-speed counter value directly through addressing addresses HC0, HC1... HC9 without calling the HSCGETCV instruction.

Parameter Name	Input/Output	Parameter description	type	Numerical Range	remarks
MOD_ADR	IN	retain	BYTE	--	--
CH_ADR	IN	Channel address	BYTE	0~9	0：HSC0 1：HSC1 ..... 9：HSC9
CV	OUT	Current count value	DINT	--	Current count value
STA	OUT	Return status	BYTE	0~255	0：OK 1: Parameter error 2: Access module error

HSCGETSTA retrieves the current count status

Parameter Name	Input/Output	Parameter description	type	Numerical Range	remarks
MOD_ADR	IN	retain	BYTE	--	--
CH_ADR	IN	Channel address	BYTE	0~9	0：HSC0 1：HSC1 ….. 9：HSC9
HSC_STA	OUT	Counter status	BYTE	--	BIT0~BIT3: Current mode BIT4: Reserved BIT5: HSC current counting direction position, 1=increment count
STA	OUT	Return status	BYTE	0~255	0：OK 1: Parameter error 2: Access module error

HSCGETSPEED retrieves the current speed

Note: When obtaining the speed, the HSC_SETParameter instruction needs to be called first to set the time speed measurement window.

Parameter Name	Input/Output	Parameter description	type	Numerical Range	remarks
MOD_ADR	IN	retain	BYTE	--	--
CH_ADR	IN	Channel address	BYTE	0~9	0：HSC0 1：HSC1 ….. 9：HSC9
SPEED	OUT	Current speed	DWORD	--	Unit: Hz
STA	OUT	Return status	BYTE	0~255	0：OK 1: Parameter error 2: Access module error

HSCGETLOCK retrieves the current latch value

Parameter Name	Input/Output	Parameter description	type	Numerical Range	remarks
MOD_ADR	IN	retain	BYTE	--	--
CH_ADR	IN	Channel address	BYTE	0~9	0：HSC0 1：HSC1 ….. 9：HSC9
LOCKDATA	IN	Current latch value	DINT	--	Current latch value
STA	OUT	Return status	BYTE	0~255	0：OK 1: Parameter error 2: Access module error

HSC_CLEARLOCK clears the latch value

Parameter Name	Input/Output	Parameter description	type	Numerical Range	remarks
MOD_ADR	IN	retain	BYTE	--	--
CH_ADR	IN	Channel address	BYTE	0~9	0：HSC0 1：HSC1 ….. 9：HSC9
STA	OUT	Return status	BYTE	0~255	0：OK 1: Parameter error 2: Access module error

HSC_SETParameterSetting HSC parameters

Set parameters such as HSC filtering time, time measurement window, and latch window.

When using the capture function, it is necessary to set the latch window through the HSC_SETParameter instruction.

Parameter Name	Input/Output	Parameter description	type	Numerical Range	remarks
MOD_ADR	IN	retain	BYTE	--	--
CH_ADR	IN	Channel address	BYTE	0~9	0：HSC0 1：HSC1 ….. 9：HSC9
FILTER_PARA	IN	Filter parameters	BYTE	1-6	1-Frequency below 50kHz 2- Frequency below 80kHz 3- Frequency below 125kHz 4- Frequency below 150kHz 5- Frequency below 200kHz 6- Frequency below 250kHz Default is 5
TIME_WINDOW	IN	Time measurement window	WORD	0~65535	Default is 5ms. Unit: ms
LOCK_WINDOW	IN	Lock window	DWORD	--	The latch window represents the pulse interval generated by two consecutive valid locks. If set to 0, there will be no window interval, and the current count value will be locked upon detecting the capture signal. If set to greater than 0, the next lock operation will only be initiated after the current effective lock is generated and the HSC count reaches or exceeds the set value. Within the range of less than the set pulse number, there will be no response to the capture signal.
STA	OUT	Return status	BYTE	0~255	0：OK 1: Parameter error 2: Access module error

HSCGETPV retrieves the preset value for generating interrupts

• Call the HSCGETPV instruction to retrieve the PV value at the time of the most recent interrupt.

• When the program starts running, if the PV=CV interrupt has never been generated, the obtained PV value is 0.

• After executing the HSC-200 instruction, the PV value obtained before the first PV=CV interrupt is 0.

• If real-time acquisition of the PV value that generates interrupts is required, this instruction can be called within the interrupt program.

• Based on the preset interrupt values obtained from HSCGETPV, the corresponding logical action of immediate interrupt output can be executed to achieve different control effects of preset multi segment CV=PV.

Parameter Name	Input/Output	Parameter description	type	Numerical Range	remarks
MOD_ADR	IN	retain	BYTE	--	--
CH_ADR	IN	Channel address	BYTE	0~9	0：HSC0 1：HSC1 ..... 9：HSC9
STA	OUT	Return status	BYTE	0~255	0：OK 1: Parameter error 2: Access module error
PV	OUT	The PV value that generated the last PV==CV interrupt	DINT	--	--

HSC_SWINDOW_CEAR Clears Lock Window

When the HSC latch enters the window period, calling this instruction can clear the latch window, and HSC can immediately respond to the next capture signal.
Attention: The latch function will not be turned off. If HSC has not entered the window period, this instruction is invalid.

Parameter Name	Input/Output	Parameter description	type	Numerical Range	remarks
MOD_ADR	IN	retain	BYTE	--	--
CH_ADR	IN	Channel address	BYTE	0~9	0：HSC0 1：HSC1 ..... 9：HSC9
STA	OUT	Return status	BYTE	0~255	0：OK 1: Parameter error 2: Access module error

HSC-REMAP input signal remapping

◆ The HSC signal input I point remapping function maps an HSC input signal to any I point input on the PLC machine. If it is necessary to adjust the input pin of HSC or disable a certain pin, complete it by calling this instruction.

◆ I points can be reused, and when the same I point is used by multiple HSCs, it takes effect simultaneously. The functionality of the I point signal is processed according to the configuration of the corresponding HSC. If the I point is not reused, the repeatability of the I point needs to be confirmed during programming.

For example:

(1) If I0.0 is configured as a clock by HSC0, HSC1, and HSC2, then when a signal arrives at I0.0, all three HSCs will count.

(2) I0.1 is configured as direction by HSC0, reset by HSC1, and capture by HSC2. When the I0.1 signal arrives, HSC0 can reverse counting, HSC1 performs reset, and HSC2 performs capture operation.

◆ The four parameters CLK_A, DIR_CLK-B, RESET, and CATCH describe the offset of point I. The description method is: 8 * byte offset+bit offset

For example: I1.2, the parameter is 8 * 1+2=10. I0.7, then the parameter is 8 * 0+7=7

Parameter Name	Input/Output	Parameter description	type	Numerical Range	remarks
MOD_ADR	IN	retain	BYTE	--	--
CH_ADR	IN	Channel address	BYTE	0~9	0：HSC0 1：HSC1 ..... 9：HSC9
PIN_USING	IN	Enable signal usage	BYTE	0~23	Bit0: CLK_Signal is valid Bit1: DIR_CLK-B signal is valid Bit2: RESET signal is valid Bit3: CATCH signal is valid Bit4-Bit7: Reserved
CLK_A	IN	CLK (clock) or A-phase input point I	BYTE	0~23	
DIR_CLK_B	IN	DIR (pulse) or B-phase input point I	BYTE	0~23	
RESET	IN	Reset input point I	BYTE	0~23	
CATCH	IN	Capture input point I	BYTE	0~23	
STA	OUT	Return status	BYTE	0~7	0: Mapping successful 1: The parameter 'CHADR' is illegal 2: Module error 3: Illegal CLK_S parameter 4: The DIR_CL_B parameter is illegal 5: The RESET parameter is illegal 6: CATCH parameter is illegal 7: The current HSC mapping conflicts with point I being the same

2.6.3 Example of using high-speed counter

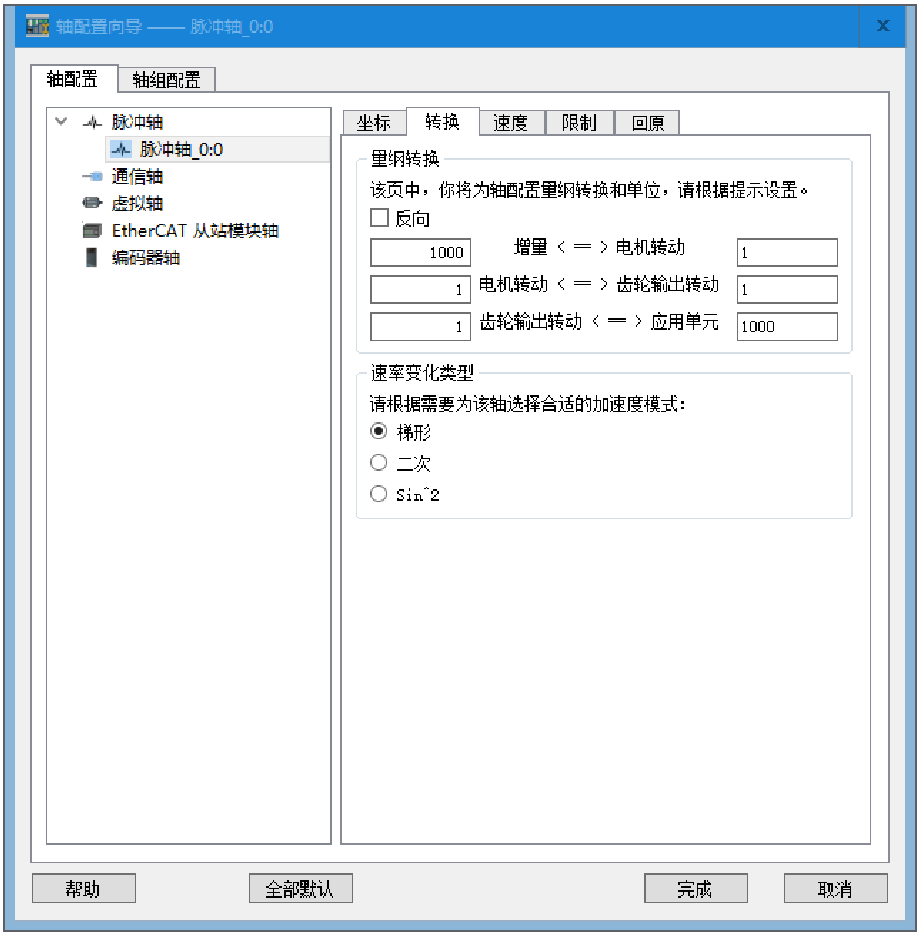
Example 1: Pulse Axis Combined Counting Mode 4 (Internal Counting) Example

Mode 4 is an internal counting mode that does not require configuring external input points, so there is no need to configure input points in the hardware configuration.

The example uses HSC0 mode 4 for counting explanation. Interrupt counting is performed based on the pulse output size, with a positive count of 2000 for interrupt plus one and a negative count of 2000 for interrupt minus one.

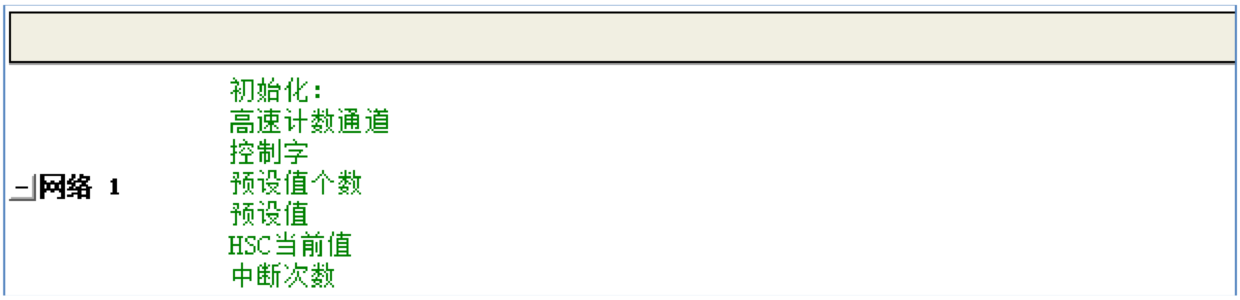
1. Set dimensional conversion

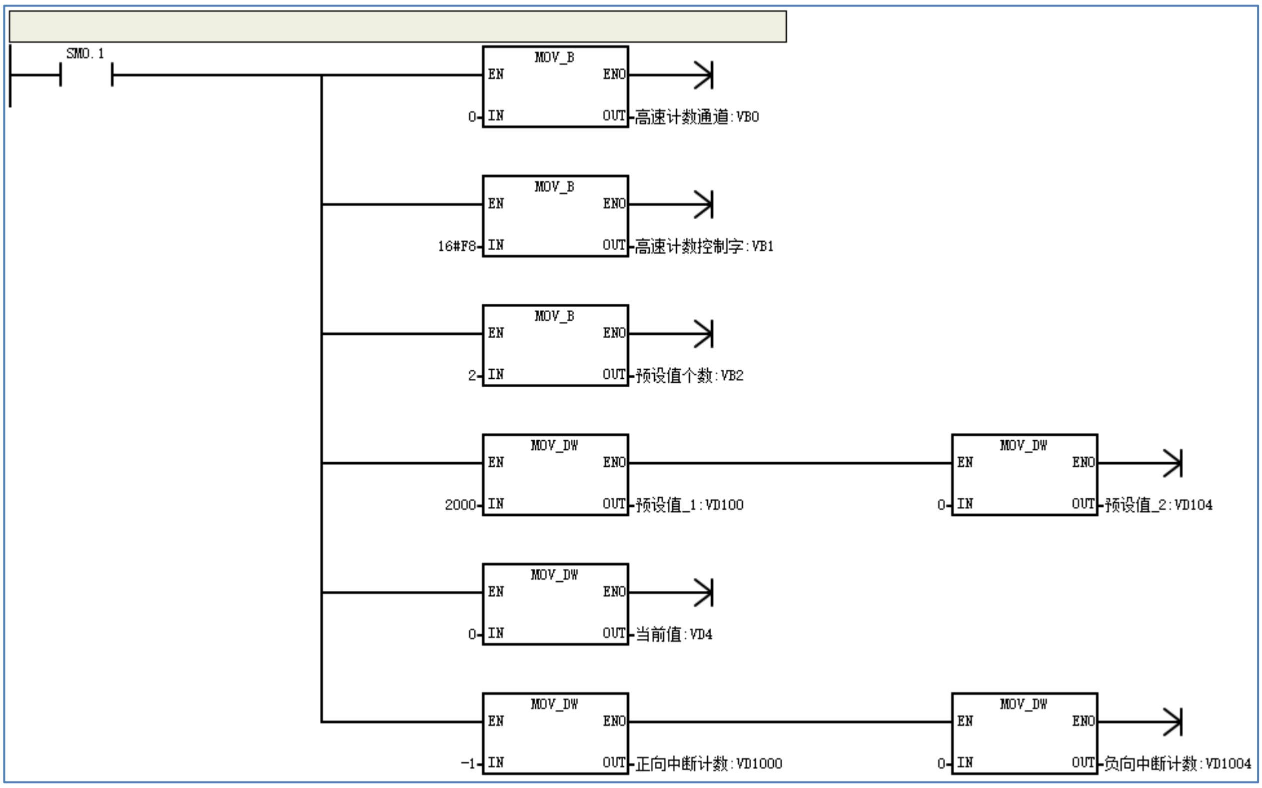
Hardware Configuration Axis Configuration Wizard Add Pulse Axis, set the pulse axis dimension conversion on the conversion page. When the application unit is 1000, the servo rotates once.



2. Programming

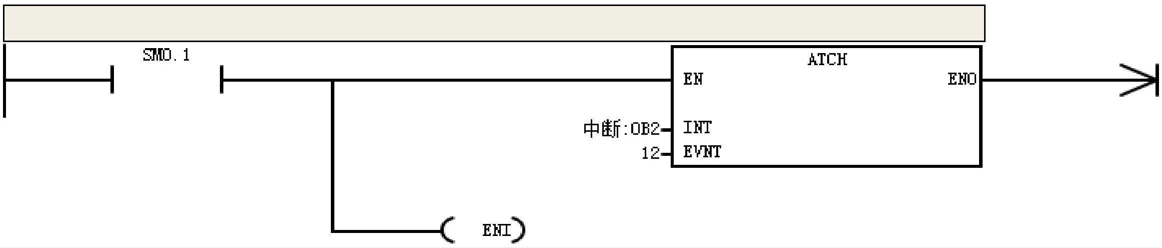
Main program:





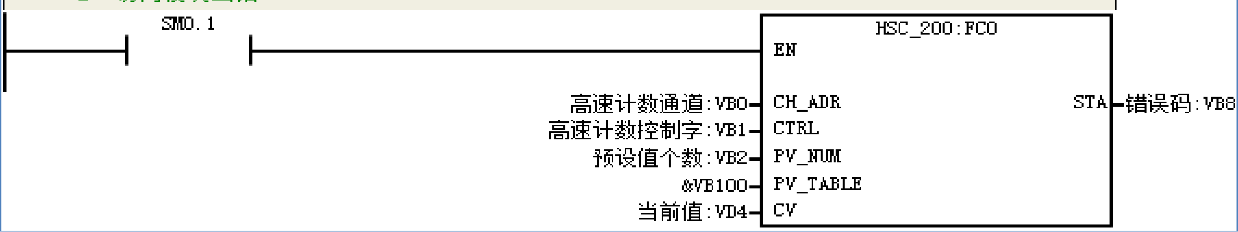
网络 2

绑定中断（只触发一次就行，不要重复触发）
****
INT：中断程序接口
EVNT：中断事件号
****
ENI：总中断使能



网络 3　控制字设置，必须调用，只调用一次就行，中途如果更改了参数，需要重新调用

CH_ADR：0-9对应HSC0-HSC9
****
CTRL：控制字
BIT0：复位电平，0-高电平复位，1-低电平复位
BIT1：捕获模式的触发信号有效边沿：0：下降沿有效，1：上升沿有效
BIT2：正交计数选择。00：4x倍数　01：1x倍数。
BIT3：计数方向，0-减计数，1-增计数
BIT4：计数方向更新，0-不更新，1-更新
BIT5：预设值更新，0-不更新，1-更新
BIT6：当前值更新，0-不更新，1-更新
BIT7：计数使能，0-不使能，1-使能
****
PV_NUM：预设值个数
****
PV_TABLE：预设值表（要用指针）
****
CV：当前值
****
STA：0：OK
1：参数错，
2：访问模块出错

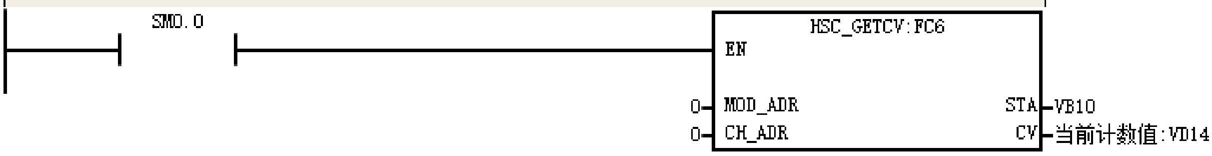


网络 4　　计数模式设置，必须调用，只调用一次就行，中途如果更改了参数，需要重新调用

MOD_ADR：　300系列用的参数，200不用。
****
CH_ADR：HSC通道地址。
****
MODE：　计数模式。
　　　　每一种模式都带外部捕获信号，可以通过HSC配置来使能捕获信号I点。
　　　　模式0：具有内部方向控制的单相计数器。
　　　　模式1：具有外部方向控制的单相计数器。
　　　　模式2：具有2个时钟输入的双相计数器。
　　　　模式3：A\B相正交计数器。
　　　　模式4：计数轴输出的脉冲值，hsc0-轴0
****
STA：0：OK
　　　1：参数错
　　　2：访问模块出错

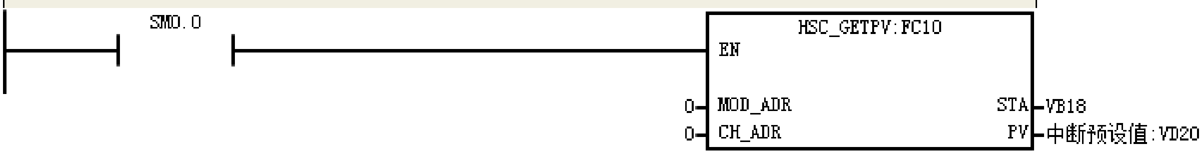
网络 5　　读取当前计数值

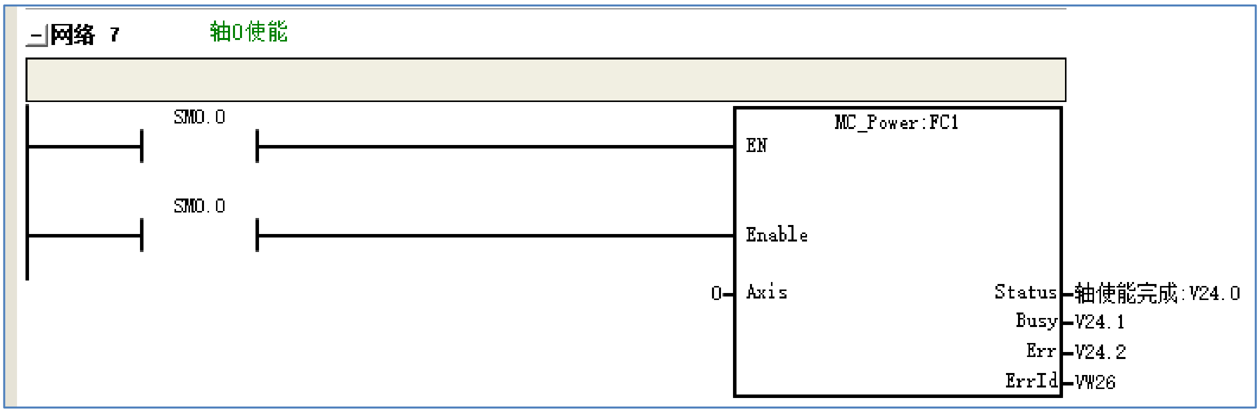
MOD_ADR：　300系列用的参数，200不用。
****
CH_ADR：HSC通道地址。
****
CV：　当前计数值
****
STA：
　0：OK
　1：参数错，
　2：访问模块出错

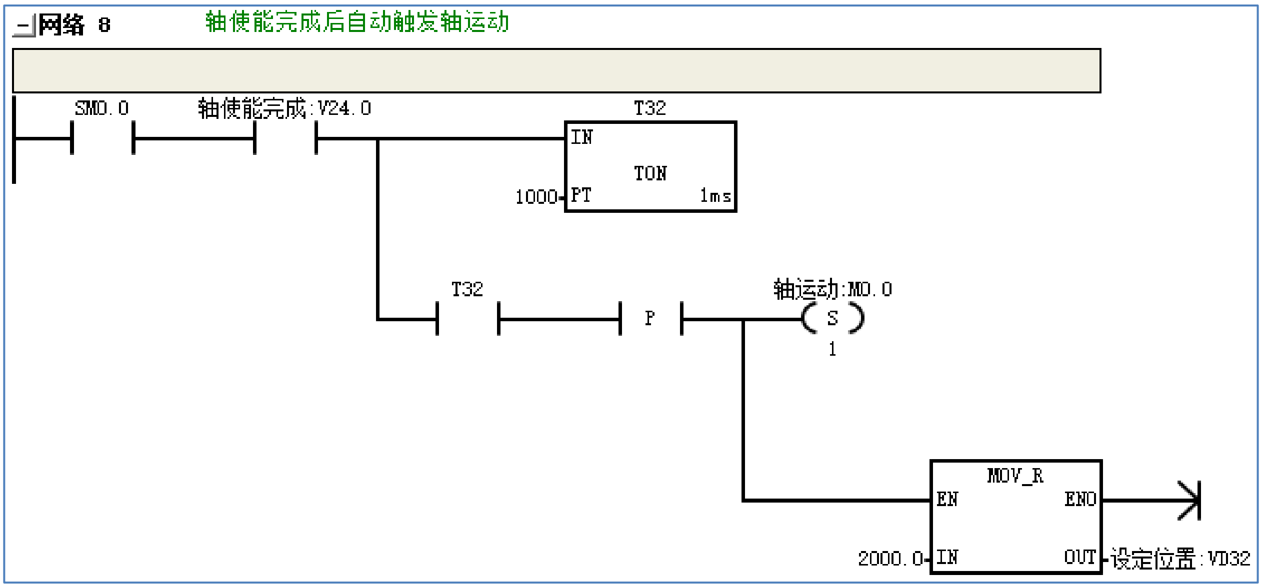


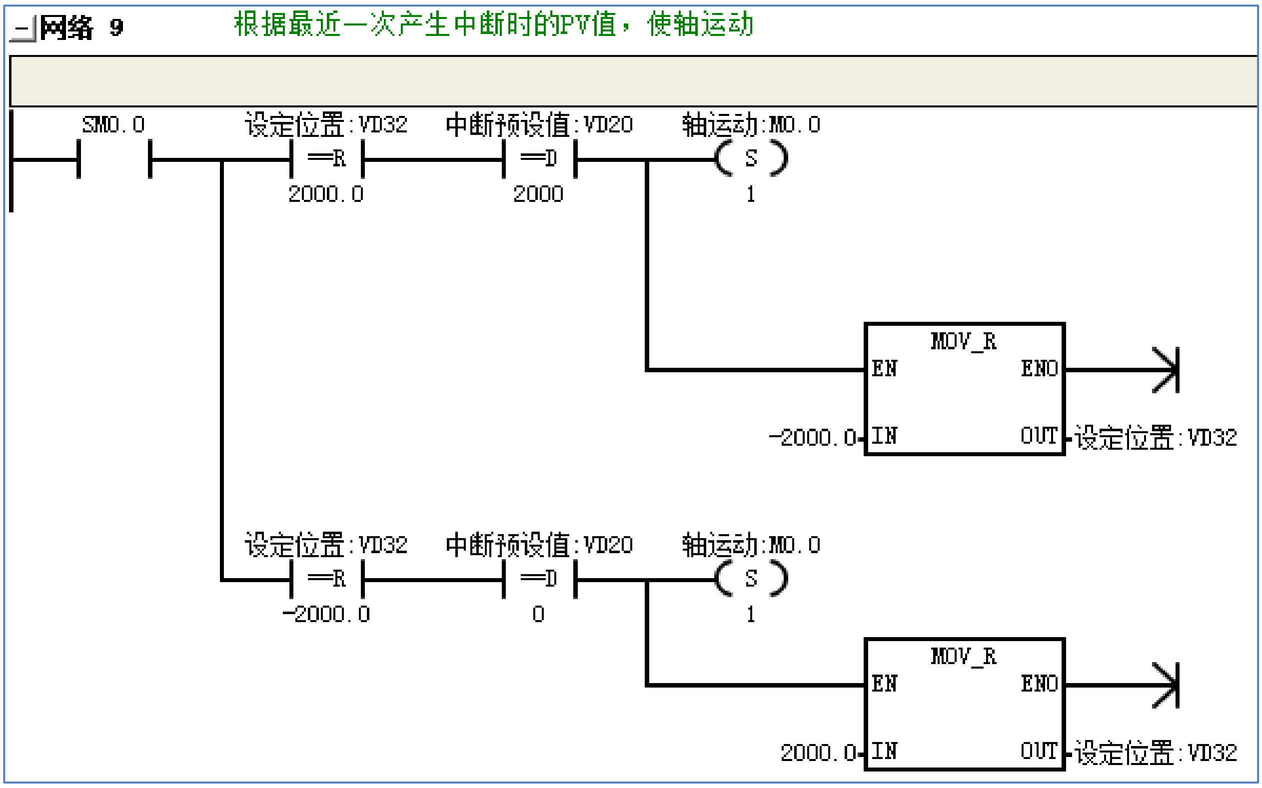
网络 6　　获取最近一次产生中断时的PV值

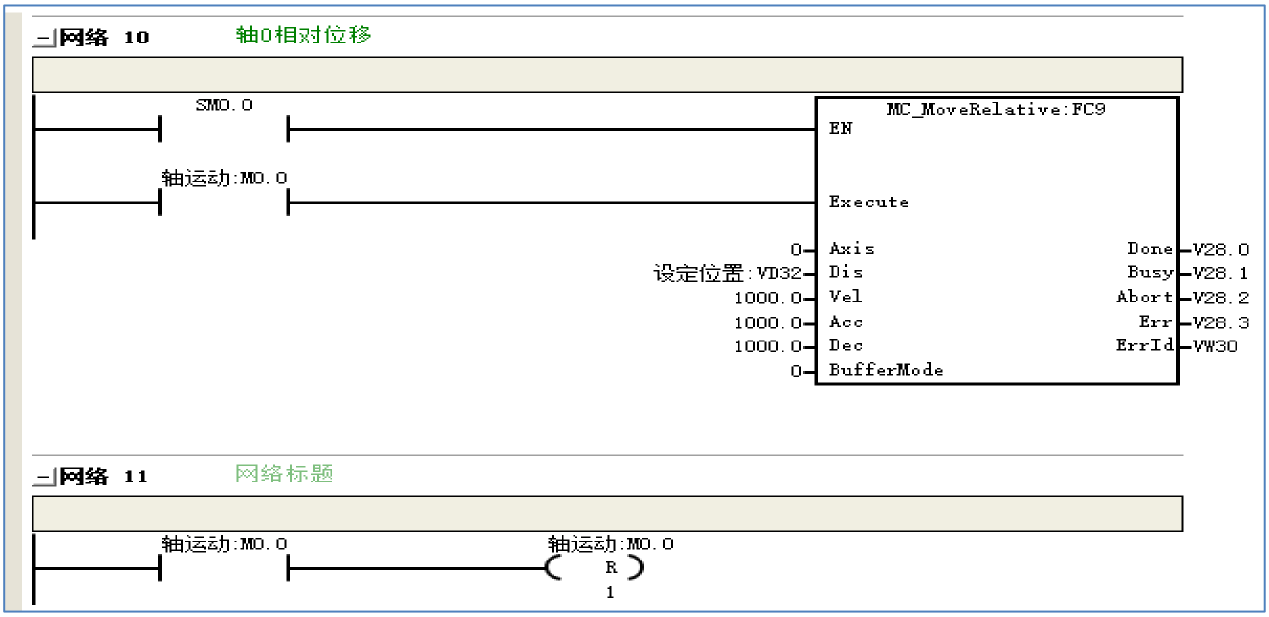
HSC_GETPV指令调用时，获取最近一次产生中断时的PV值。
程序开始运行的时候，如果从未产生过PV==CV中断，则获取到的PV值是0。
执行了HSC_200指令之后，在产生第一次PV==CV中断之前，获取到的PV值是0。
当存在多个PV值时，如果要实时检测产生中断PV值，在中断内调用该指令实时性最好。
****
MOD_ADR：　300系列PLC的参数，200系列PLC不用。
CH_ADR：HSC通道地址。
****
PV：　产生中断的PV值。
****
STA：
　0：OK
　1：参数错，
　2：访问模块出错



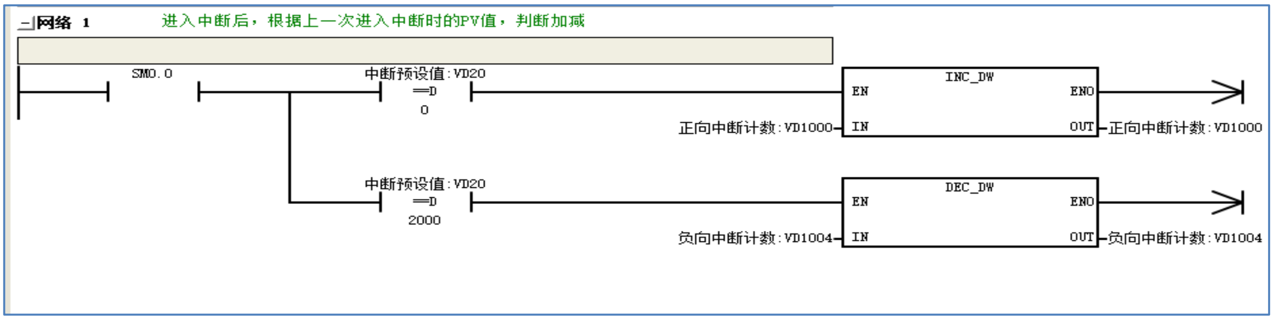

网络 7
轴0使能
SM0.0
SM0.0
MC_Power:FC1
EN
Enable
0
Axis
Status
Busy
Err
ErrId
轴使能完成:V24.0
V24.1
V24.2
VW26


网络 8
轴使能完成后自动触发轴运动
SM0.0
轴使能完成:V24.0
T32
IN
TON
1000
PT
1ms
T32
P
轴运动:M0.0
S
1
MOV_R
EN
ENO
2000.0
IN
OUT
设定位置:VD32


网络 9
根据最近一次产生中断时的PV值，使轴运动
SM0.0
设定位置:VD32
==R
2000.0
中断预设值:VD20
==D
2000
轴运动:M0.0
S
1
MOV_R
EN
ENO
-2000.0
IN
OUT
设定位置:VD32
设定位置:VD32
==R
-2000.0
中断预设值:VD20
==D
0
轴运动:M0.0
S
1
MOV_R
EN
ENO
2000.0
IN
OUT
设定位置:VD32

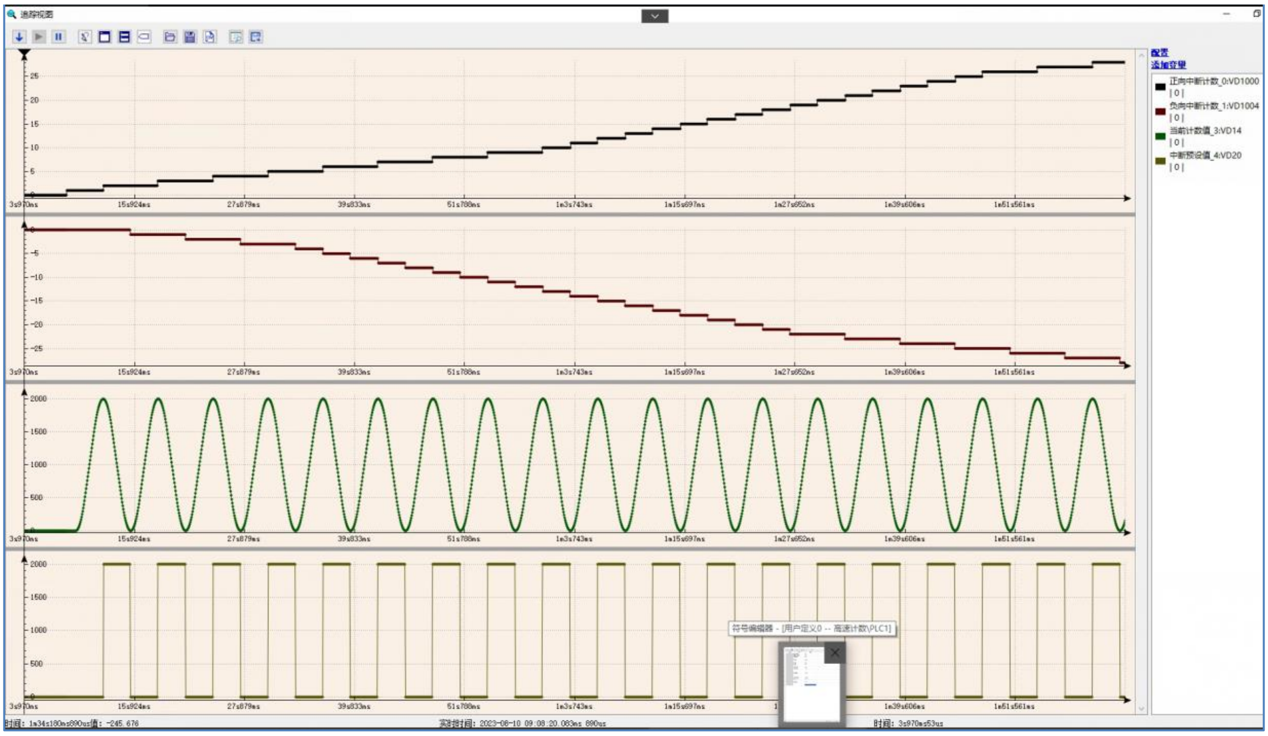


Associated interrupt subroutine:



3. View the running results in the trace view

In State Table ->Menu ->View ->Tracking or by clicking on the icon in the State Table toolbar, start the Tracking view and add pulse variables and count values.

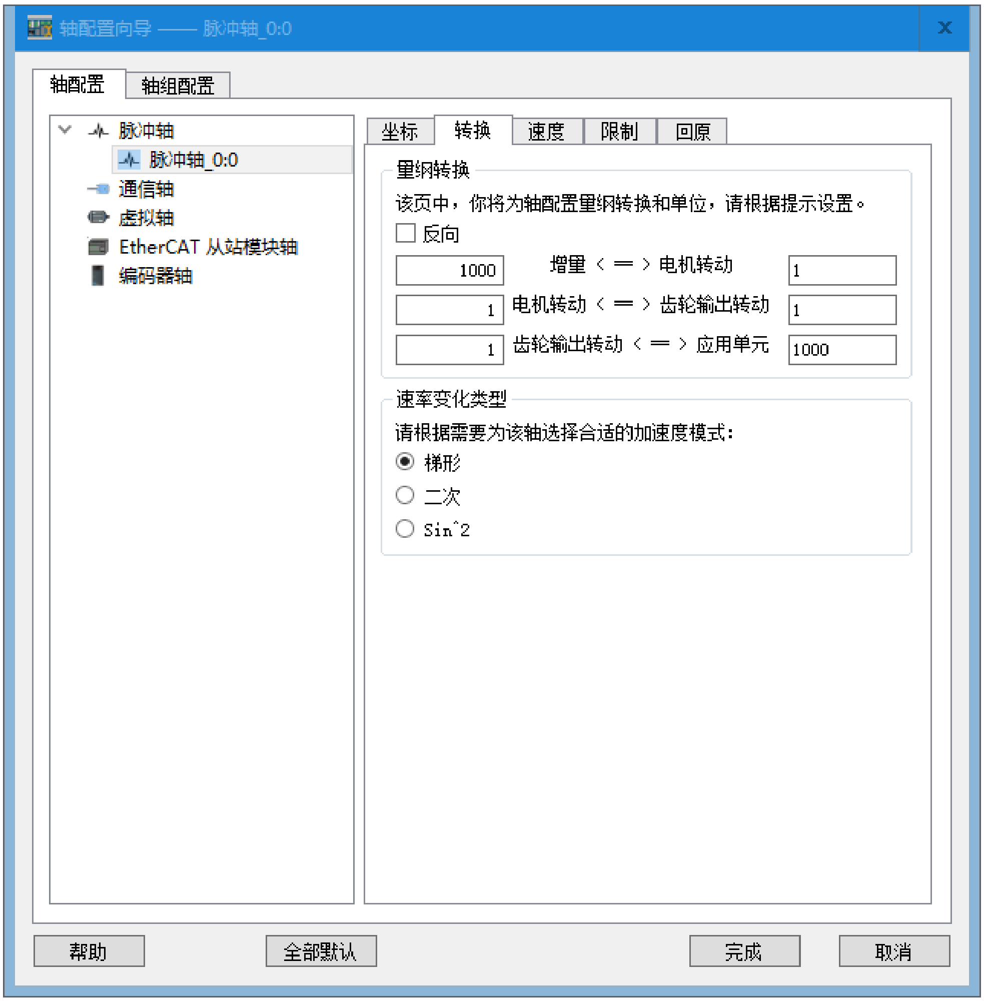


Example 2: Pulse axis combined with counting mode 4 (internal counting, multi segment PV values) Example

In this example, 10 values are set to demonstrate with internal counting. The servo dimension is 10 meters per circle, and HSC is used to count internal pulses. Each forward movement is 100 meters and the reverse movement is 90 meters, and HSC is used to determine the direction change.

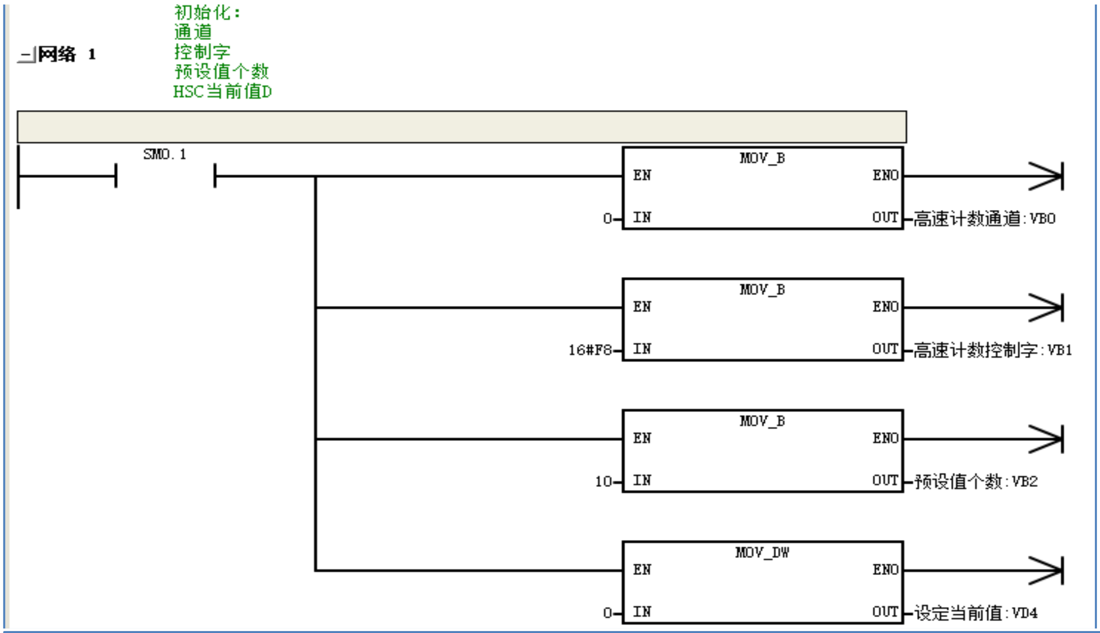
1. Set dimensional conversion

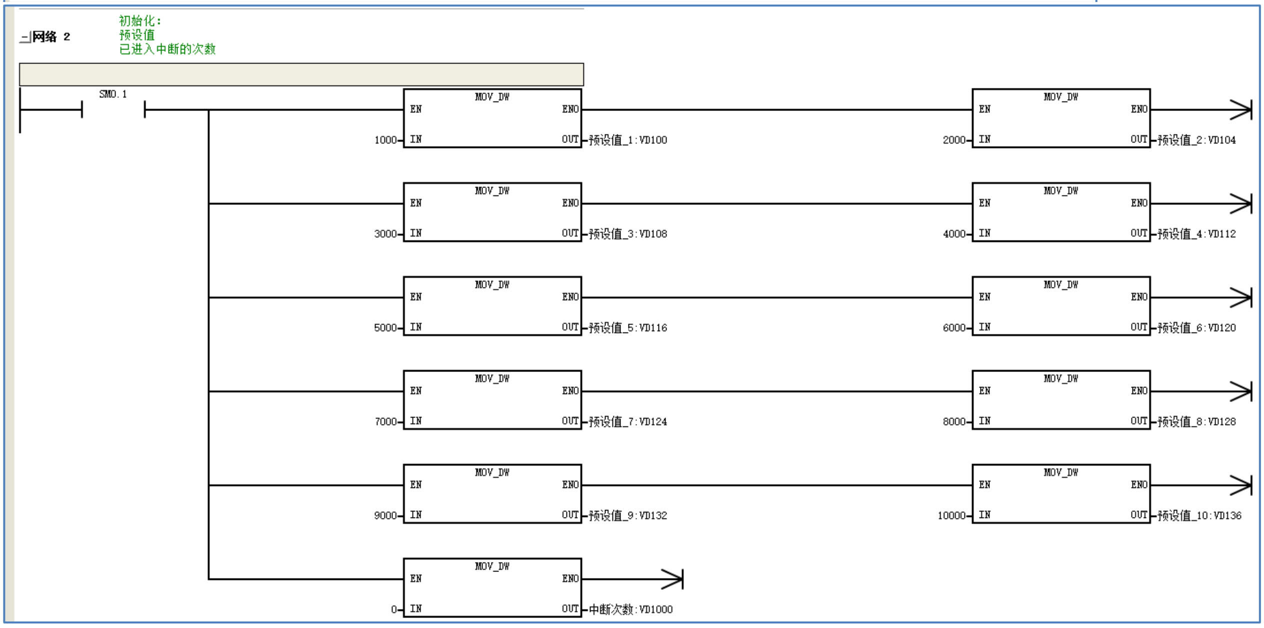
Hardware Configuration Axis Configuration Wizard Add Pulse Axis, set the pulse axis dimension conversion on the conversion page. When the application unit is 1000, the servo rotates once.

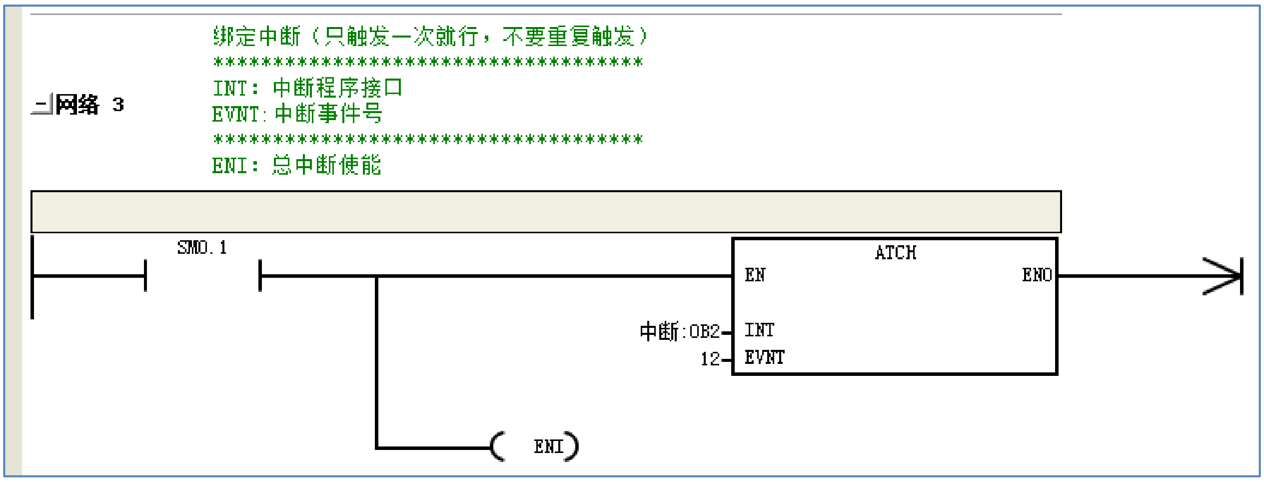


2. Program Editing

Main program:

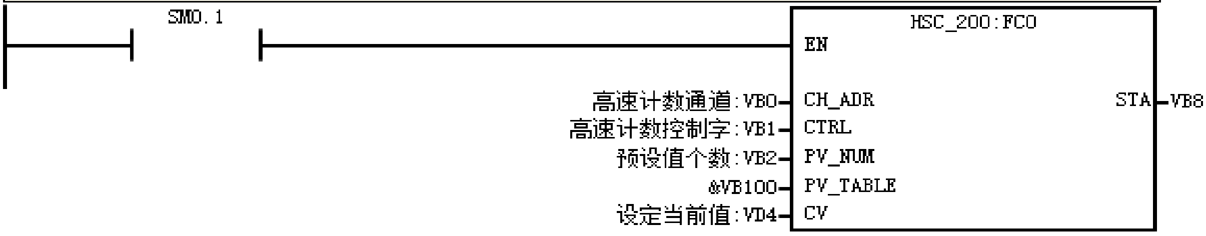

网络 1
初始化：
通道
控制字
预设值个数
HSC当前值D
SM0.1
MOV_B
EN ENO
0 IN OUT 高速计数通道:VB0
MOV_B
EN ENO
16#F8 IN OUT 高速计数控制字:VB1
MOV_B
EN ENO
10 IN OUT 预设值个数:VB2
MOV_DW
EN ENO
0 IN OUT 设定当前值:VD4


网络 2
初始化：
预设值
已进入中断的次数
SM0.1
MOV_DW
1000 IN OUT 预设值_1:VD100
2000 IN OUT 预设值_2:VD104
3000 IN OUT 预设值_3:VD108
4000 IN OUT 预设值_4:VD112
5000 IN OUT 预设值_5:VD116
6000 IN OUT 预设值_6:VD120
7000 IN OUT 预设值_7:VD124
8000 IN OUT 预设值_8:VD128
9000 IN OUT 预设值_9:VD132
10000 IN OUT 预设值_10:VD136
0 IN OUT 中断次数:VD1000


网络 3
绑定中断（只触发一次就行，不要重复触发）
****
INT：中断程序接口
EVNT：中断事件号
****
ENI：总中断使能
SM0.1
ATCH
EN ENO
中断:OB2 INT
12 EVNT
ENI

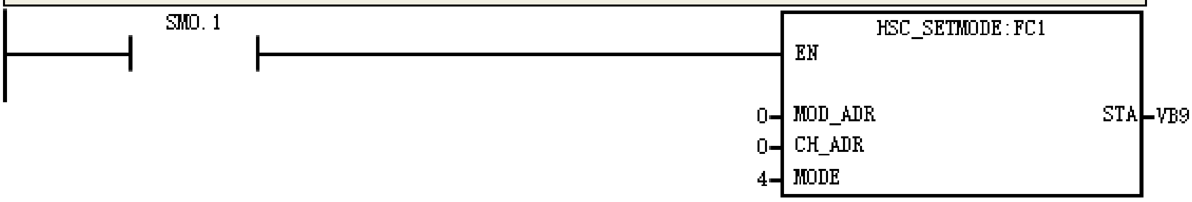
网络 4　　控制字设置，必须调用，只调用一次就行，中途如果更改了参数，需要重新调用

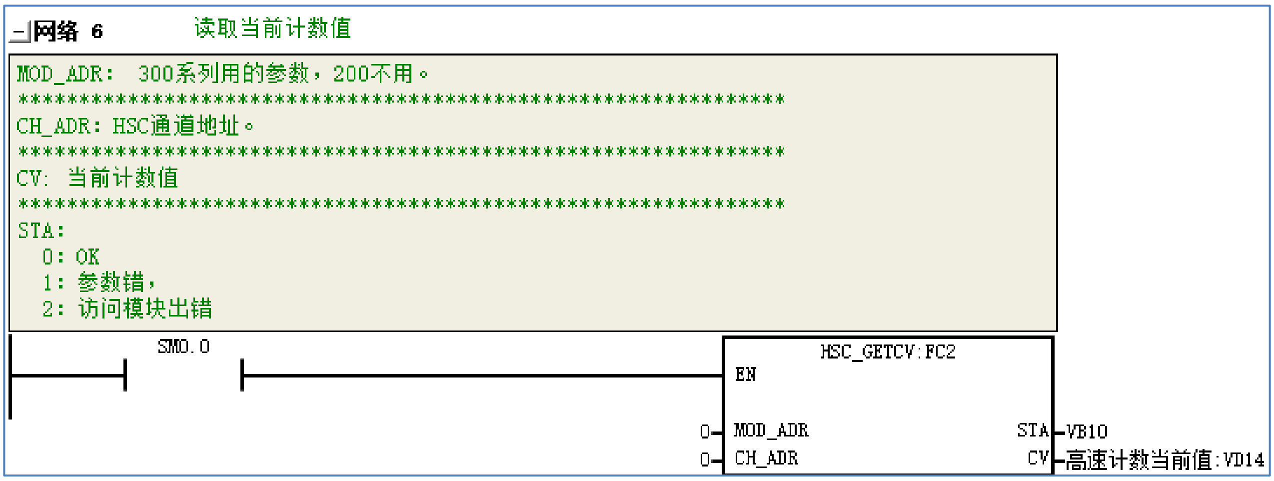
```
CH_ADR: 0-9对应HSC0-HSC9
*****************************************************************
CTRL:  控制字
       BIT0:  复位电平，0-高电平复位，1-低电平复位
       BIT1:   捕获模式的触发信号有效边沿：0：下降沿有效，1：上升沿有效
       BIT2:   正交计数选择。 00:  4x倍数   01：1x倍数。
       BIT3:  计数方向，0-减计数，1-增计数
       BIT4:   计数方向更新，0-不更新，1-更新
       BIT5:   预设值更新，0-不更新，1-更新
       BIT6:当前值更新，0-不更新，1-更新
       BIT7:计数使能，0-不使能，1-使能
*****************************************************************
PV_NUM:    预设值个数
*****************************************************************
PV_TABLE:预设值表（要用指针）
*****************************************************************
CV:    当前值
*****************************************************************
STA: 0: OK
      1: 参数错，
      2: 访问模块出错
```

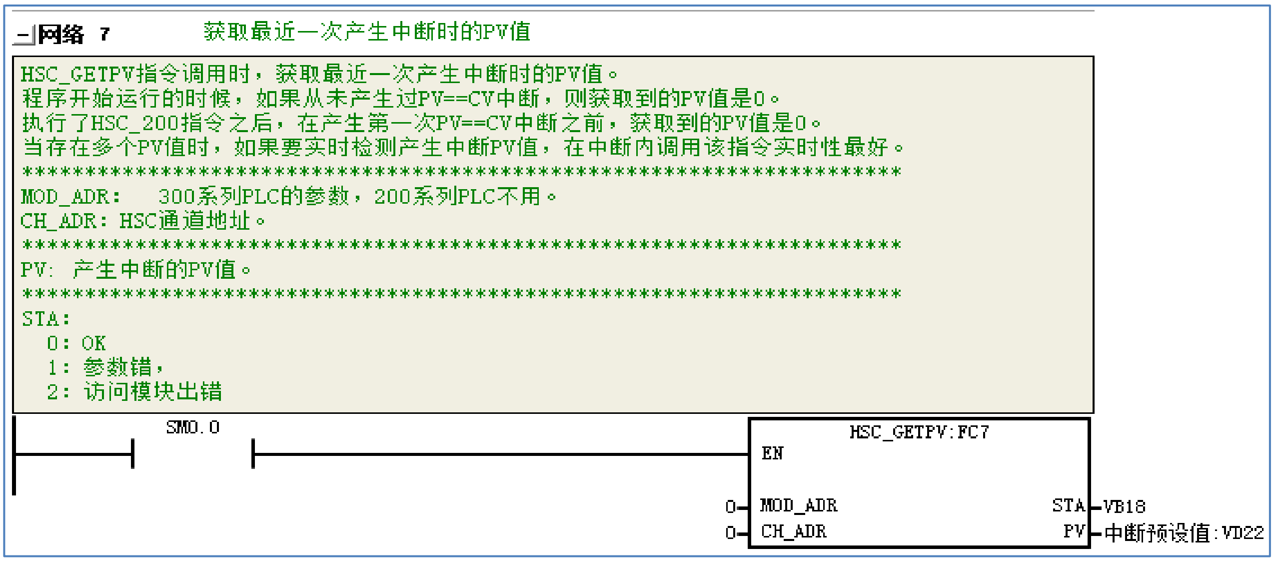


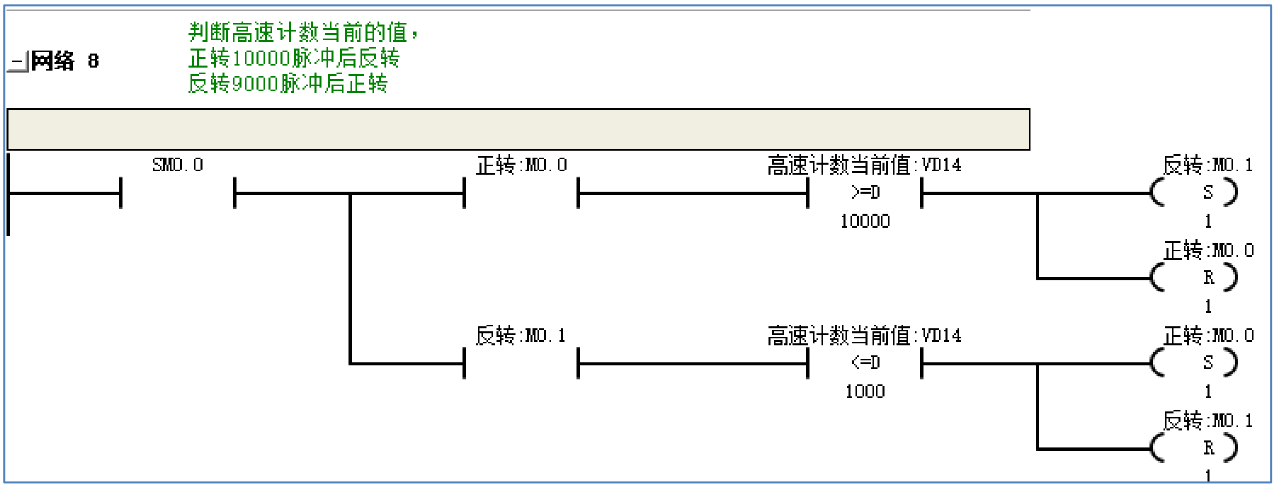
网络 5　　控制字设置，必须调用，只调用一次就行，中途如果更改了参数，需要重新调用

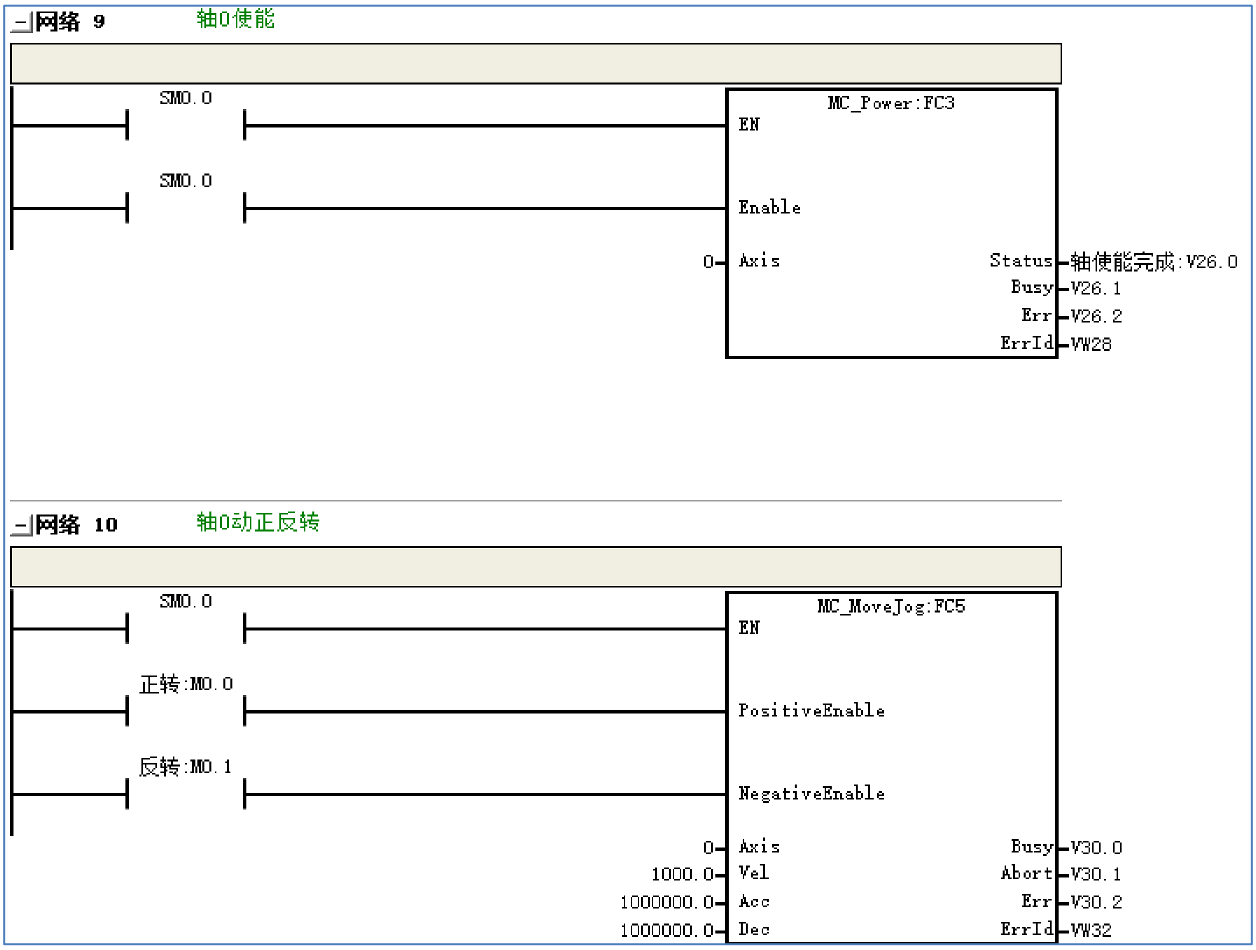
```
MOD_ADR:  300系列用的参数，200不用。
*****************************************************************
CH_ADR: HSC通道地址。
*****************************************************************
MODE:  计数模式。
         每一种模式都带外部捕获信号，可以通过HSC配置来使能捕获信号I点。
         模式0: 具有内部方向控制的单相计数器。
         模式1: 具有外部方向控制的单相计数器。
         模式2: 具有2个时钟输入的双相计数器。
         模式3: A\B相正交计数器。
         模式4: 计数轴输出的脉冲值，hsc0-轴0
*****************************************************************
STA: 0: OK
      1: 参数错
      2: 访问模块出错
```

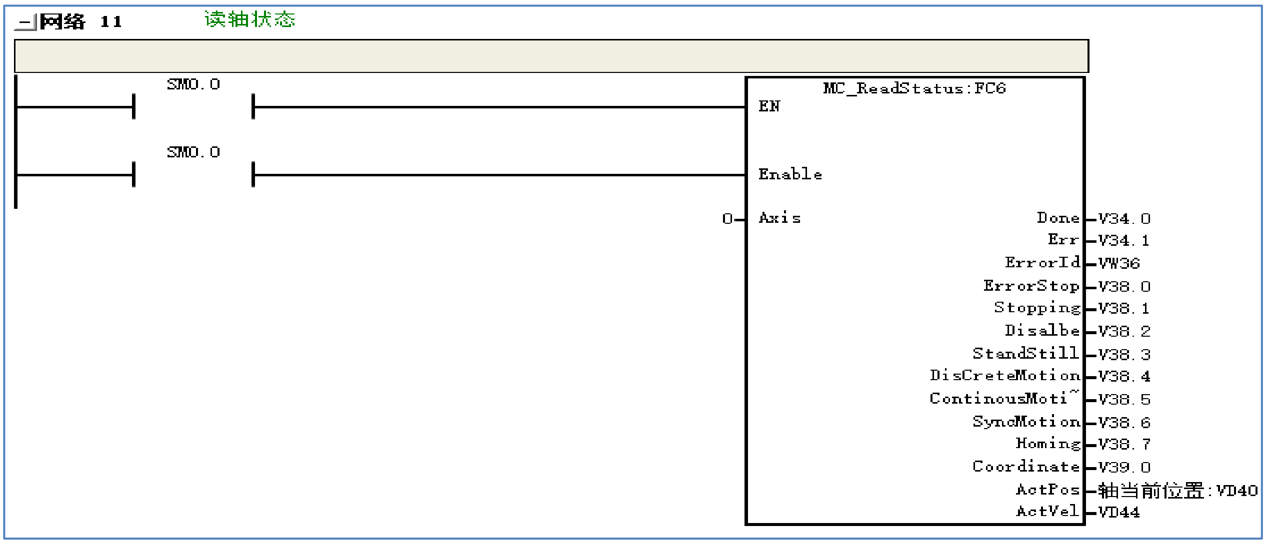


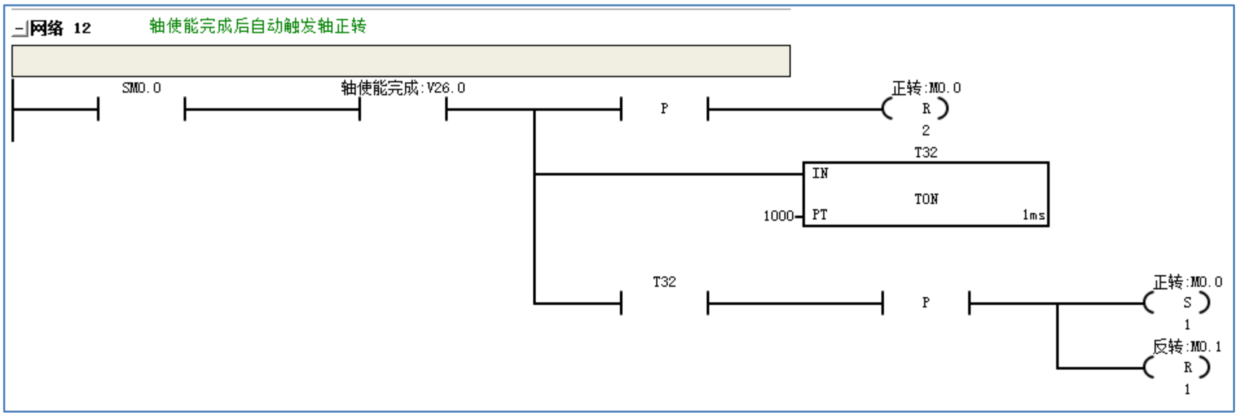

网络 6
读取当前计数值
MOD_ADR： 300系列用的参数，200不用。
****
CH_ADR：HSC通道地址。
****
CV： 当前计数值
****
STA：
0：OK
1：参数错，
2：访问模块出错
SM0.0
HSC_GETCV:FC2
EN
0 MOD_ADR
0 CH_ADR
STA VB10
CV 高速计数当前值:VD14


网络 7
获取最近一次产生中断时的PV值
HSC_GETPV指令调用时，获取最近一次产生中断时的PV值。
程序开始运行的时候，如果从未产生过PV==CV中断，则获取到的PV值是0。
执行了HSC_200指令之后，在产生第一次PV==CV中断之前，获取到的PV值是0。
当存在多个PV值时，如果要实时检测产生中断PV值，在中断内调用该指令实时性最好。
****
MOD_ADR： 300系列PLC的参数，200系列PLC不用。
CH_ADR：HSC通道地址。
****
PV： 产生中断的PV值。
****
STA：
0：OK
1：参数错，
2：访问模块出错
SM0.0
HSC_GETPV:FC7
EN
0 MOD_ADR
0 CH_ADR
STA VB18
PV 中断预设值:VD22

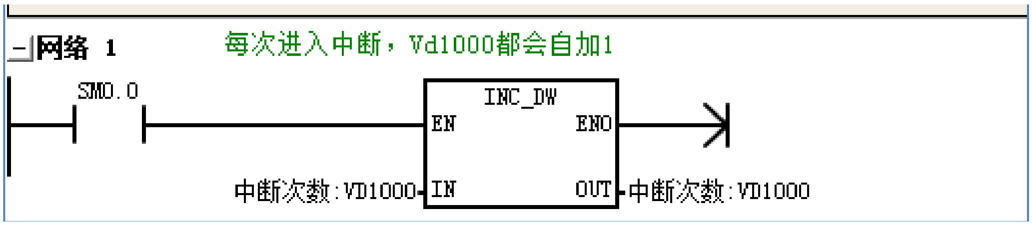

网络 8
判断高速计数当前的值，
正转10000脉冲后反转
反转9000脉冲后正转
SM0.0
正转:M0.0
高速计数当前值:VD14
>=D
10000
反转:M0.1
S
1
正转:M0.0
R
1
反转:M0.1
高速计数当前值:VD14
<=D
1000
正转:M0.0
S
1
反转:M0.1
R
1







Interrupt program:



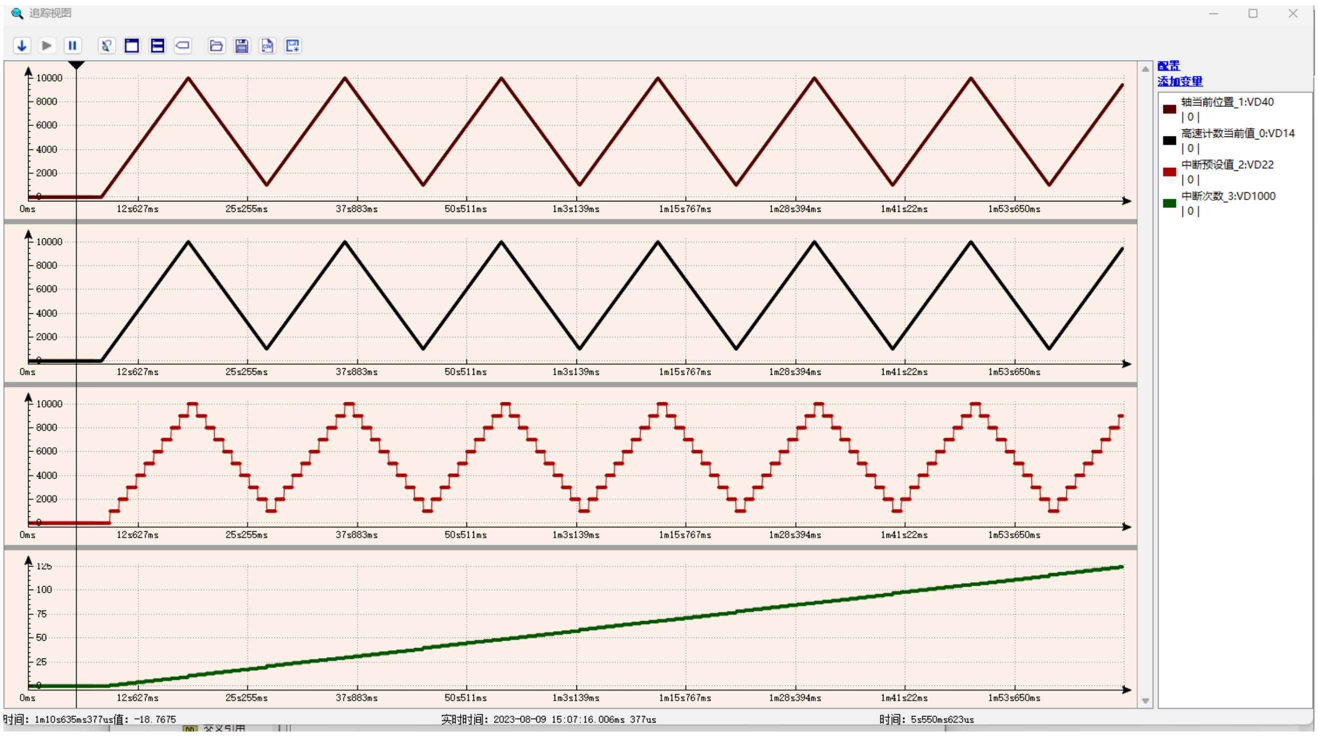
3. Operation results

• When counting, as long as CV equals any value in PV_TABLE, the CV=PV interrupt can be triggered.

• If the condition is met again after counting up and counting down, the interrupt CV=PV will still be triggered. If there is 0 in the preset table and the initial value of the counter initialization is also 0, the interrupt will not be entered at this time.

• Based on the preset interrupt value obtained from the HSCGETPV instruction, the corresponding logical action of immediate interrupt output can be executed to achieve different control effects of preset multi segment CV=PV.

• The output value of HSCGETPV is the PV value at the last time CV=PV. After triggering the HSC-200 instruction, the preset value table will update the HSCGETPV output value to zero until the next triggering condition is met and PV will be output.



Example 3: Encoder Axis Combined Counting Mode 3 (A/B Phase Orthogonal Counting) Example

This example takes the high-speed counter HSC0, mode 3 (A/B phase orthogonal counting) as an example to achieve the encoder axis reading of the current actual position and speed, as well as the use of multi segment position interruption.

一、preparation in advance

1. CTMC M226ES, firmware 3.11 and above

2. Library ct_plcopentlib (v2.9), software comes with built-in

3. Library ct-HSCtlib (v3.1), software comes with built-in

4. PLC programming software: Magicworks PLC V2.29

2、 Encoder shaft configuration and HSC hardware wiring configuration

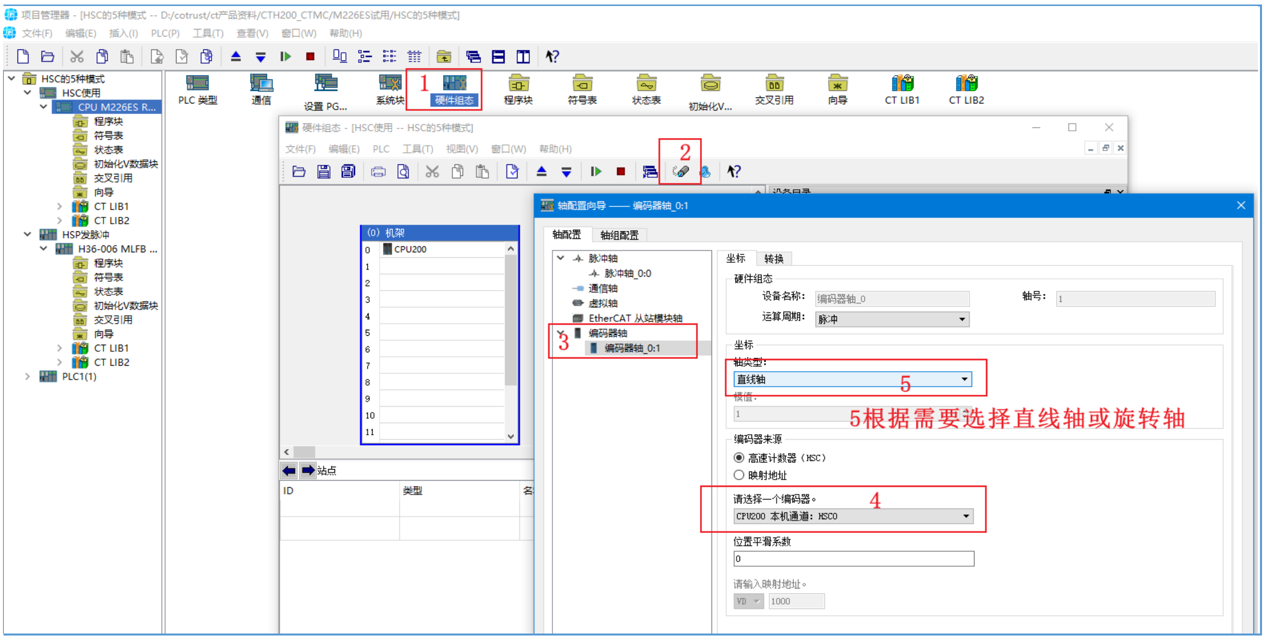
1. Add encoder shaft and set shaft parameters

• Axis type

Linear axis: with reciprocating motion. Rotation axis: only rotates in one direction.

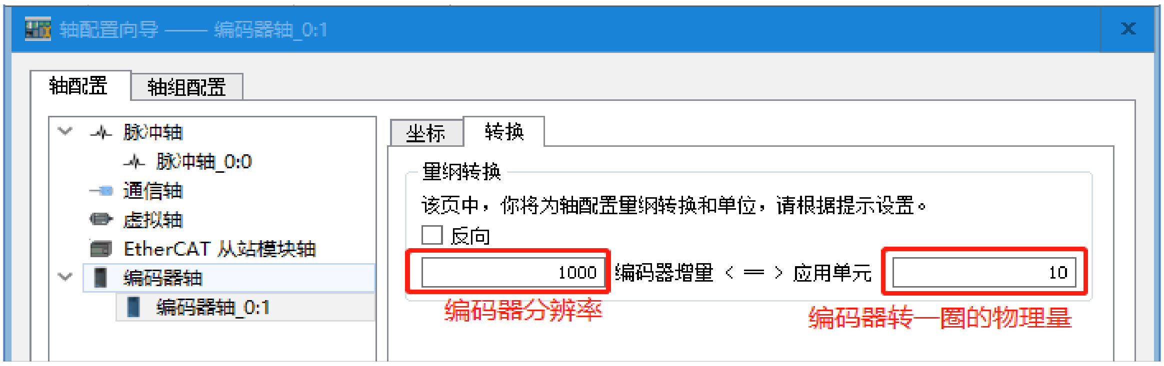
• Encoder channel: refers to the interface of the high count counter channel connected to the encoder on the local machine.

This example takes the linear axis and HSC0 as examples.



• Conversion - Encoder Increment: Encoder Resolution. Application unit: The physical quantity of one revolution of the actual encoder.

This example takes the encoder rotating once to 10mm as an example.



2. HSC hardware I-point wiring configuration

Configure HSC input points in Magic Works PLC [Hardware Configuration] [Tools] [HSC Configuration].

be careful:

(1). The configuration of point I for wiring is selected based on the actual signal line of the encoder.

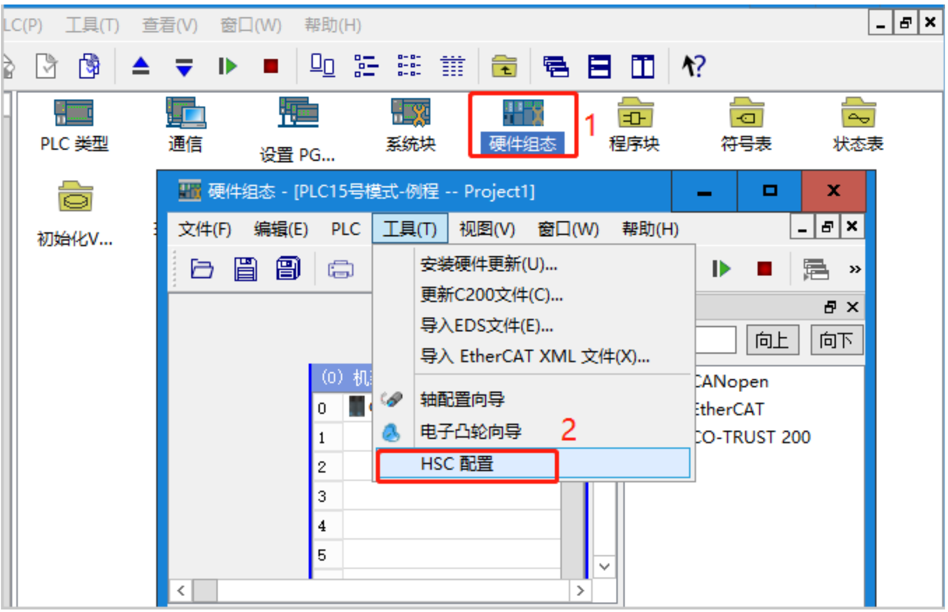
(2). The reset signal can be left unconnected, and can be reset to zero using the program call library ct_SSCtlib (v3.1).

(3). The same input point cannot be applied to multiple high-speed counters, it can only be applied to one high-speed counter.

(4). An unconfigured input can be arbitrarily configured as any input signal of the counter.

(5). The configured point requires the program to call the library ct-HSCtlib (v3.1) in order to activate the HSC function.

This example uses HSC0, AB connection line as an example to configure point I:



3、 Library instruction call

1. Enable HSC0 setting in the program

The HSC counting mode is as follows

Counting mode		Configurable input ports			
Mode 0	Single phase counter with internal directional control	Clock	--	reset	capture
Mode 1	Single phase counter with external direction control	Clock	direction	reset	capture
Mode 2	Two phase counter with up and down counting clock	Increase Clock	Clock reduction	reset	capture
Mode 3	A/B phase orthogonal counter	Clock A	Clock B	reset	capture
Mode 4	Internal counting	--	--	--	capture

HSC0-HSC9 all support the above 5 modes, and in this example, the AB phase is counted at 4 times.

Be careful:

(1) The difference between internal counting and external counting:

Internal counting is directly counted through logic, without the need for wiring or occupying point I. External counting is initiated by an external input signal.

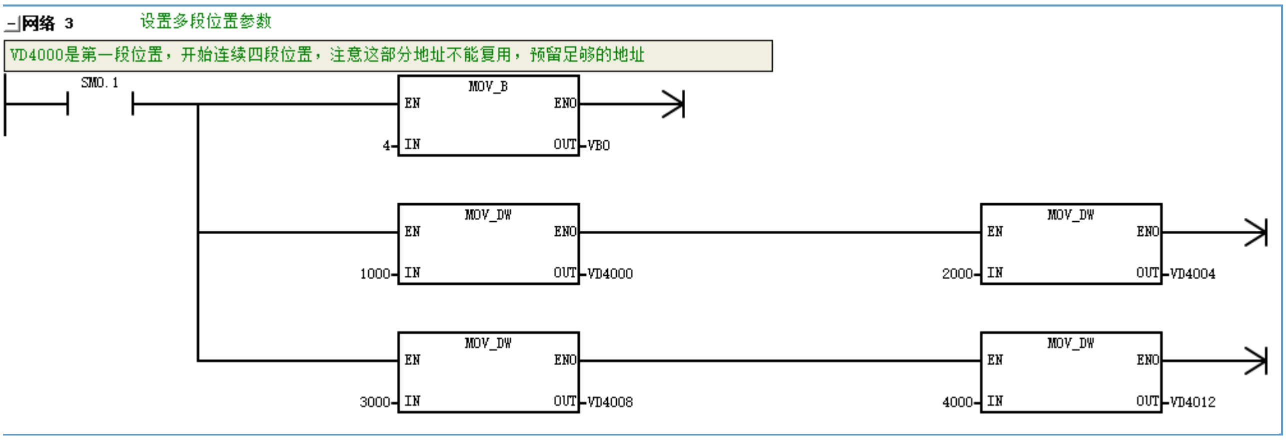
(2) The HSC-200 instruction CTRL_bit2 counts multiple according to the actual setting.

(3) If it is necessary to clear or change the CV value midway, after setting the required CV value in HSC-200, trigger it again to take effect.

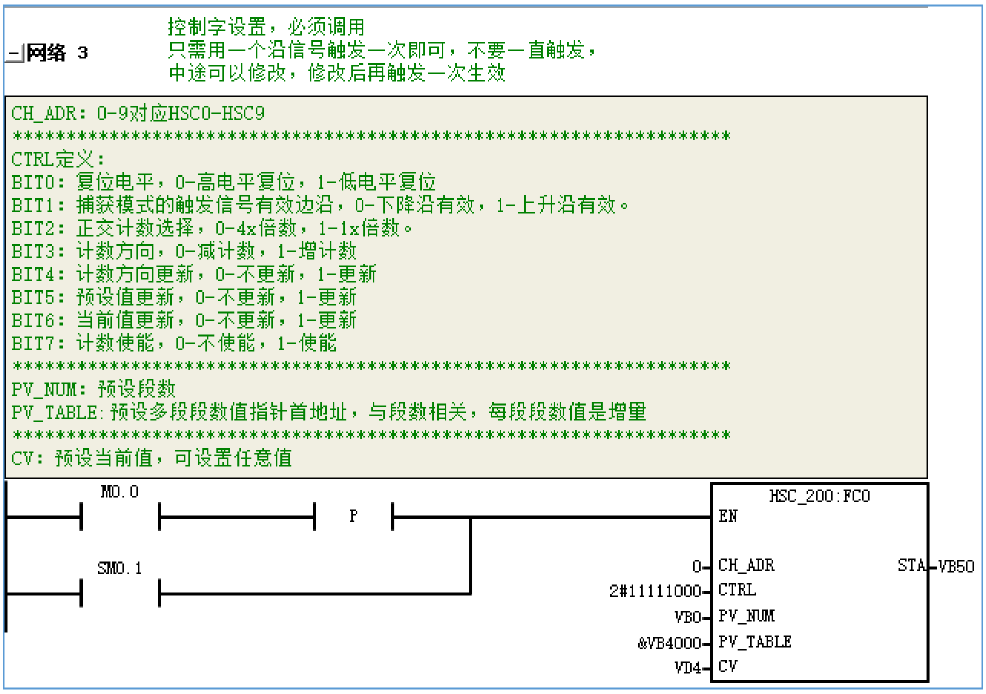
(4) The encoder shaft does not support the ct_plcopentlib (v2.9) library instruction MC_SetPosition to set the current position, so (2) is used to implement it.

2. HSC program editor

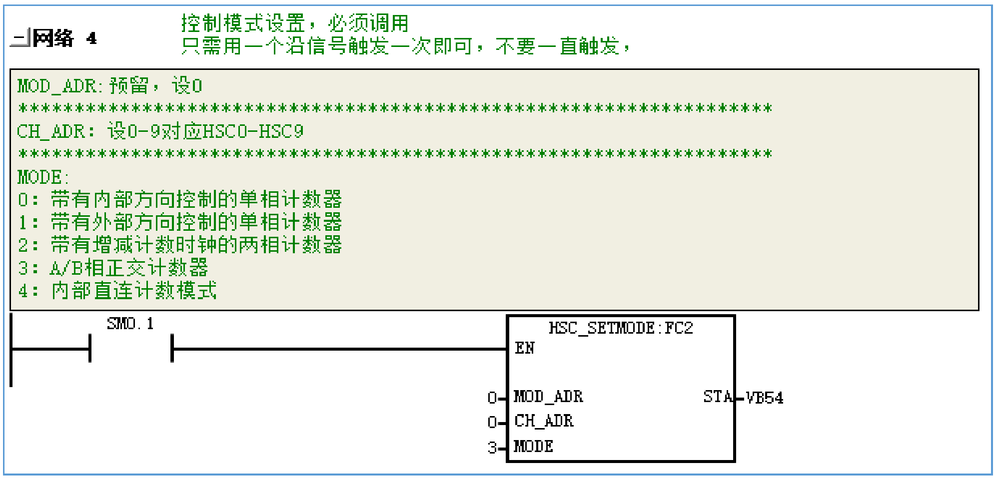
(1) HSC0 multi segment position parameter settings can be deleted without using this function.



(2) Initialize HSC0, call library instruction HSC-200

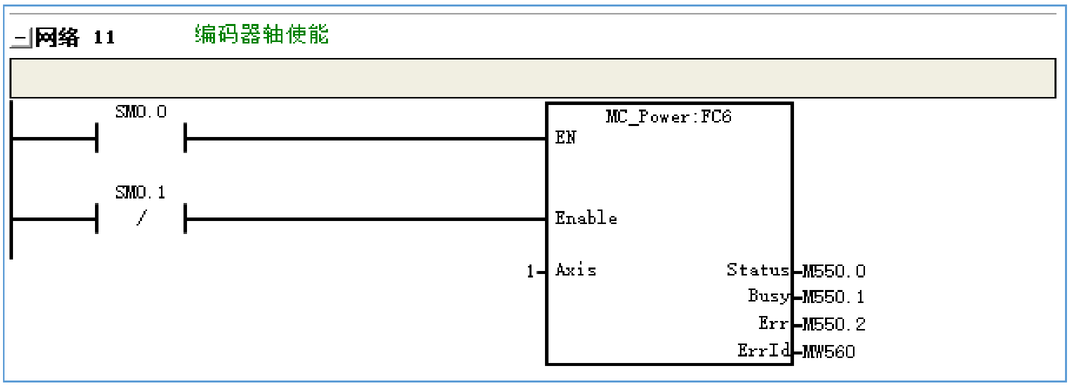


(3) HSC0 mode setting, call library instruction HSC_SETMODE

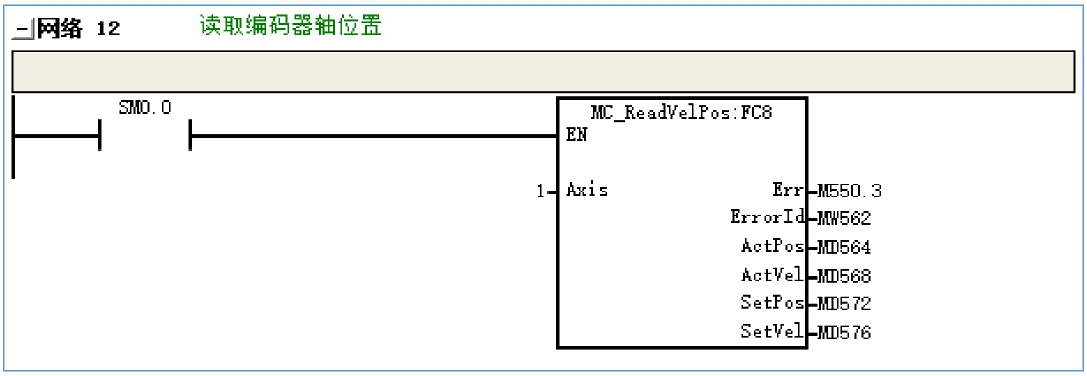


3. Use encoder axis function in the program

(1) To enable the encoder axis, the axis enable must be called. Call MC_Power from ct_plcopen_lib (v2.9)

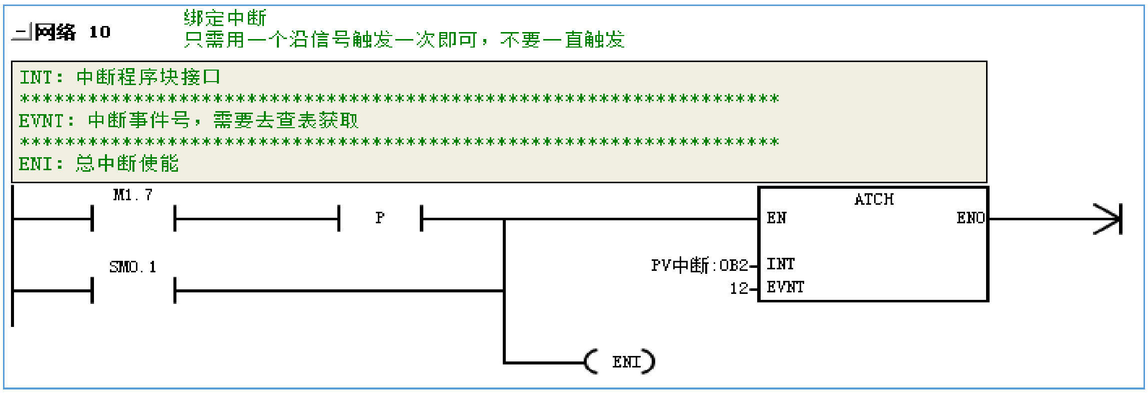


(2) Read the encoder axis position and speed, call MC_ReadVelPos from ct_plcopen_lib (v2.9), and read the actual position and speed are physical quantities after conversion, not pulse numbers.

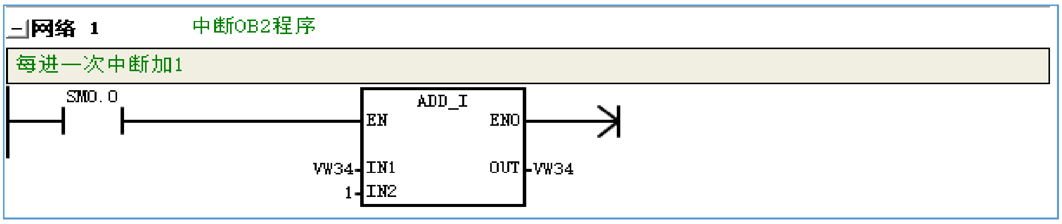


3、 Multi segment interrupt program

1. Bind interrupts and activate interrupt events



2. Interrupt program, handling interrupt events

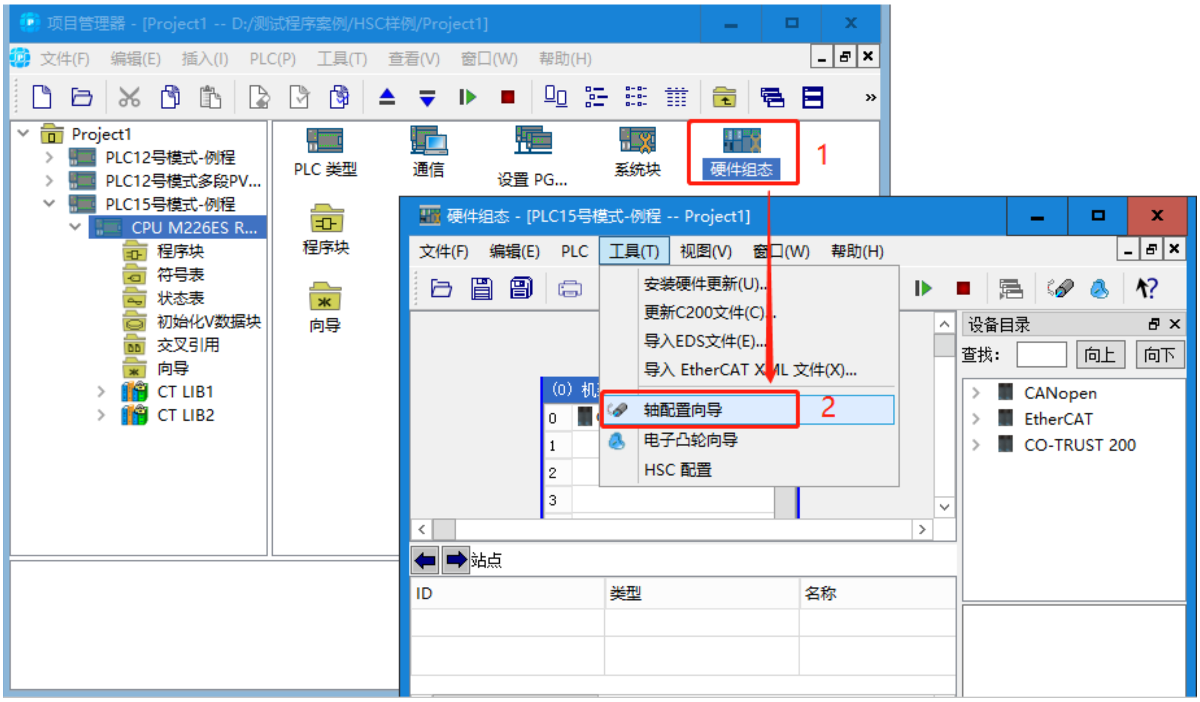


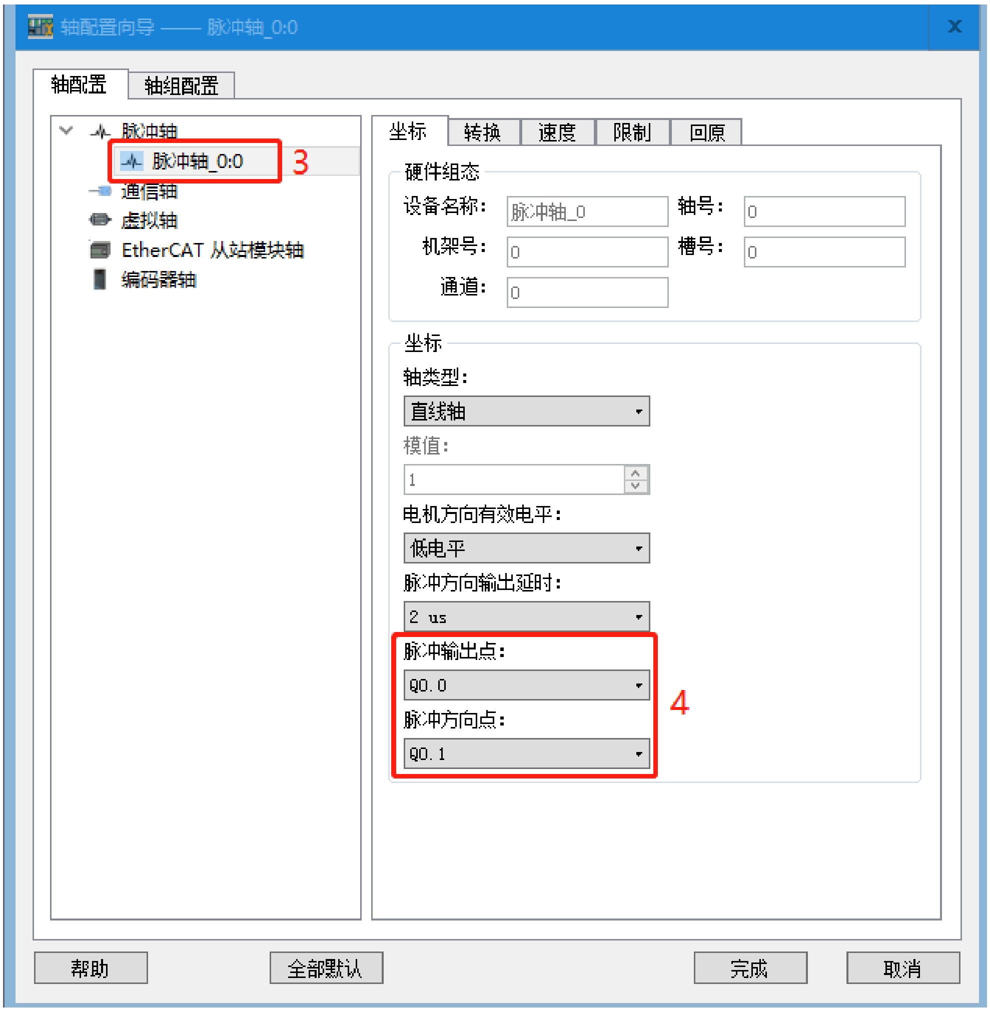
2.7 High speed pulse output

M226ES supports 10 * 200kHz (pulse/direction) high-speed pulse outputs.

Pulse and direction can be configured at any Q point. In the [Hardware Configuration] - [Axis Configuration

Wizard], add the pulse axis and configure the pulse and direction points.





3

Install

This chapter mainly introduces the installation of M226ES. Include installation instructions and dimensions for the product.

3.1 Installation precautions

Installation environment requirements:

- **Isolate PLC from heating devices, high voltage, and electronic noise**

When installing equipment components, devices that generate high voltage and high electronic noise should be separated from M226ES.

When arranging M226ES on the back panel of the control cabinet, it should be considered to place the electronic components in the lower temperature area of the control cabinet, as long-term operation of electronic components in high temperature environments will shorten their fault free time.

Consider wiring the back panel of the control cabinet, and try to avoid designing AC power supply lines, high-energy, high switching frequency DC signal lines, low-voltage signal lines, and communication cables in the same cable tray.

- **Leave appropriate space for heat dissipation and wiring**

The design of M226ES adopts natural convection heat dissipation method, and there must be at least 30mm of space above and below the module for normal heat dissipation. The distance between the front panel and the back panel should also be maintained at least 80mm.

When installing M226ES, sufficient space should be left for wiring and connecting communication cables.

> Be careful
>
> When installed vertically, the maximum allowable ambient temperature should be 10 ℃ lower than when installed horizontally, and the CPU should be installed below all expansion modules.
> Try to keep the wire as short as possible and ensure that the wire thickness meets the current requirements. The suitable wire thickness for terminal blocks is $2mm^2$ to $0.3mm^2$ (14AWG to 22AWG), and using shielded cables can achieve the best anti electronic noise characteristics. Generally, grounding the shielding layer can achieve the best effect.
> To prevent loose connections, it is important to ensure that the terminal block is securely plugged in and that the wires are firmly connected to the terminal block. To avoid damaging the terminal block, do not tighten the screws too tightly. The maximum torque for screw connections is 0.56N-m (5 inch pounds).

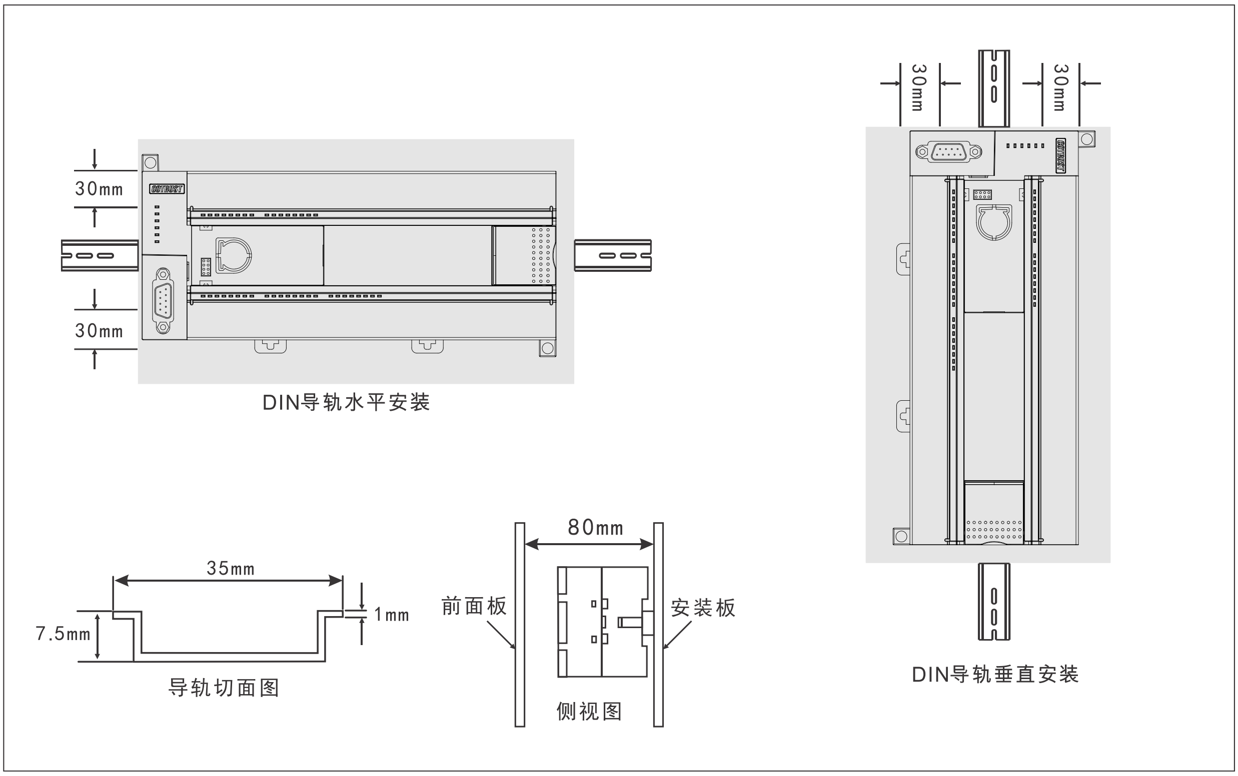


Fig3-1 installation diagram

- **Power budget**

M226ES has an internal power supply that provides 24V DC power to itself, expansion modules, and other electrical devices.

M226ES provides 5V DC logic power to all expansion modules in the system. You must pay special attention to the system configuration to ensure that the 5V DC power provided by the PLC can meet the needs of all the expansion modules you have selected. If your configuration requirements exceed the power supply capacity of the PLC, you need to remove some modules or choose a PLC with stronger power supply capacity.

M226ES also provides 24V DC sensor power supply, which can supply power to input points, relay coils on expansion modules, or other devices. If the power consumption of the device exceeds the sensor power supply budget, an external 24V DC power supply must be provided for the system.

If you are using an external 24V DC power supply, make sure that it is not connected in parallel with the sensor power supply on the PLC. To enhance electronic noise protection, it is recommended to connect the common terminals (M) of different power sources together.

The 24V DC sensor power supply of M226ES should not be supplied to the same point simultaneously with any external power supply.

	warning Connecting the external 24V power supply in parallel with the 24V sensor power supply of M226ES will cause conflicts between the two power supplies, with each power supply attempting to establish its own output potential. The result of this conflict can shorten the lifespan of the power supply or cause immediate damage to one or more power sources, leading to a series of uncertain operations in the PLC system. This uncertain operation can cause equipment damage or serious personal injury or even death.

3.2 Installation size

The installation dimension diagram of M226ES is as follows:

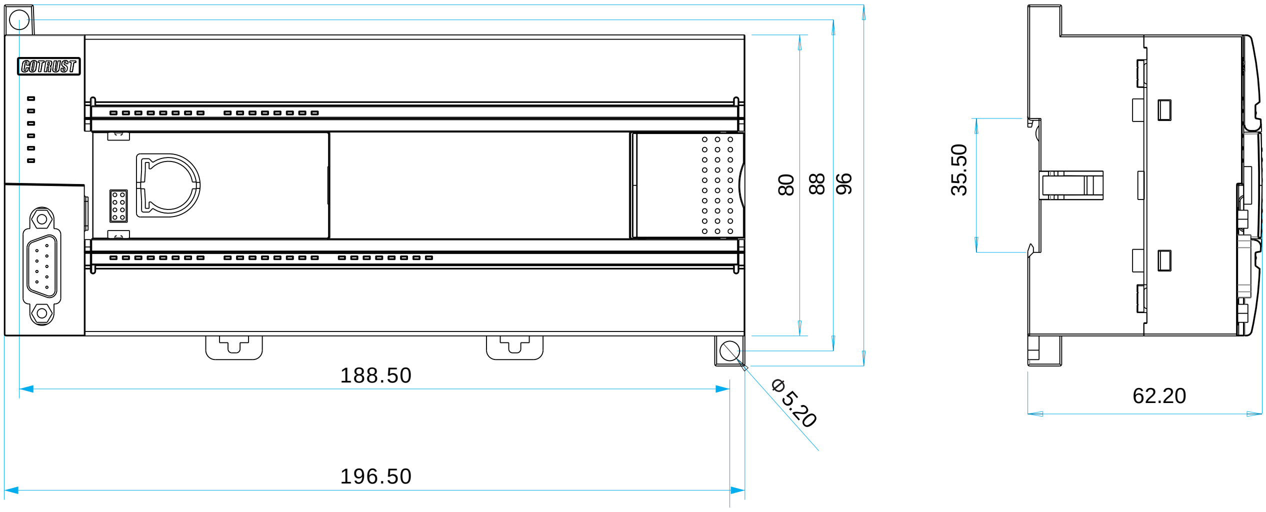


Fig3-2 CPU installation dimension diagram, unit: mm

3.3 Installation method

Installation precautions

Before installing and dismantling the PLC and its related equipment, ensure that the power supply to the PLC and its connected devices has been cut off.

⚠	warning If the PLC and its related equipment are installed or disassembled while live without cutting off all power sources, it may cause electric shock or equipment misoperation, resulting in equipment damage or serious personal injury or even death.

When replacing or installing a PLC, it is important to ensure that the correct or equivalent module is used. When replacing components, in addition to using the same module, it is also necessary to ensure that the installation direction and position are correct.

⚠	be careful If you install an incorrect module, the PLC program may produce incorrect functionality.

Installation and removal of CPU and expansion modules

Please install or remove the PLC according to the following method

• **Install panel**

1) Position and drill holes according to the size requirements in Figure 3-2.

2) Fix the module on the back panel with appropriate screws.

3) If an expansion module is used, connect the flat cable of the expansion module to the expansion port under the front cover.

• **Install DIN rail**

1) Fix the guide rail on the backboard with a spacing of 80mm.

2) Open the DIN clip at the bottom of the module and clip the back of the module onto the DIN rail.

3) If an expansion module is used, connect the flat cable of the expansion module to the expansion port under the front cover.

4) Rotate the module close to the DIN rail and close the DIN clamp.

5) Carefully check whether the DIN clamp on the module is tightly fitted to the DIN rail.

6) To avoid module damage, do not directly press the front of the module. You can press the installation hole part instead

- **Install terminal block**

1) Open the upper cover of the terminal block installation position.

2) Ensure that the pins on the module are aligned with the small holes on the edges of the terminal block.

3) Press the terminal block down into the module. Ensure that the terminal block is aligned and locked in place.

⚠	be careful When PLC is used in environments with high vibration or installed vertically, DIN rail blocks should be used. If the system is in a high vibration environment, using a backplane installation method can achieve a higher level of vibration protection.

- **Disassemble the CPU or expansion module**

Follow these steps to disassemble the CPU or expansion module:

1) Remove the power supply from the CPU.

2) Remove all wires and cables from the module.

3) If there are other expansion modules connected to the module you are removing, please open the front cover and unplug the expansion flat cable of the adjacent module.

4) Remove the installation screws or open the DIN clamp.

5) Remove the module, disassemble and install the terminal block.

- **Disassemble the terminal block**

1) Open the upper cover of the terminal block installation position.

2) Insert the screwdriver into the slot in the center of the terminal block.

3) Press down firmly and pry out the terminal block.

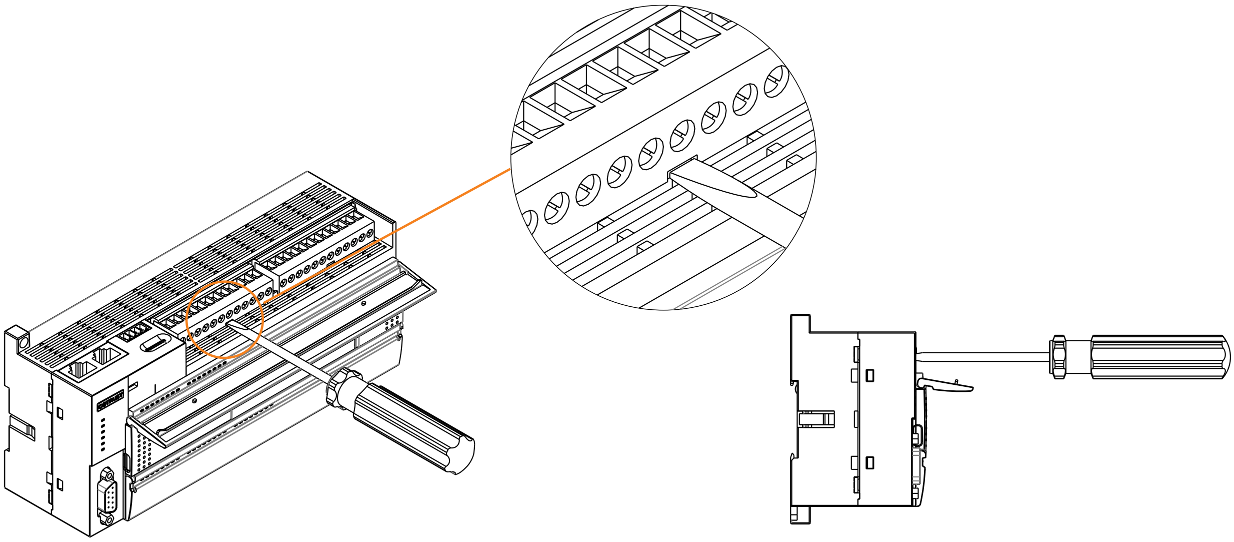


Fig3-3 Terminal strip disassembly

3.4 Grounding and wiring

Reasonable grounding and wiring of all electrical equipment is crucial to ensure optimal operating characteristics of your system and provide better electronic noise protection for your system.

Before grounding and wiring, it is necessary to ensure that the power supply of the equipment has been cut off, and also ensure that the power supply of the equipment related to the equipment has been cut off. When wiring M226ES PLC and its related equipment, it is necessary to ensure compliance with all valid electrical coding rules. All equipment installation and operation must comply with all valid national or regional standards. Maintain contact with authorities in the same region to determine which standards meet your specific needs.

	warning Attempting to ground or wire while energized may result in equipment damage or serious personal injury or even death.

When designing the grounding and wiring of M226ES, safety factors must be considered, otherwise it may cause equipment misoperation. Therefore, you should comply with all safety regulations to avoid personal injury and equipment damage.

	warning Controlling equipment may cause misoperation of the devices it controls. This kind of misoperation may lead to death or serious personal injury and equipment damage. Therefore, M226ES must have independent emergency stop function, electromechanical interlock or other redundant safety measures.

3.5 Suppression circuit

When using inductive loads, a suppression circuit needs to be added to limit the voltage rise when the output is turned off. The suppression circuit can protect the output point from premature damage due to high inductance switch current. In addition, the suppression circuit can also limit the electrical noise generated by inductive load switches.

> **be careful**
> The effectiveness of the suppression circuit depends on the application, and you should adjust its parameters to suit your specific application. Ensure that all device parameters are consistent with the actual application.

Transistor output and relay output for controlling DC loads

The transistor output has internal protection and can adapt to various applications. Due to the relay type output being able to connect both DC and AC loads, there is no internal protection.

Figure 3-4 shows an example of a DC load suppression circuit. In most applications, an additional diode A is sufficient, but if your application requires faster turn off speed, we recommend adding a Zener diode B. Ensure that the Zener diode can meet the current requirements of the output circuit.

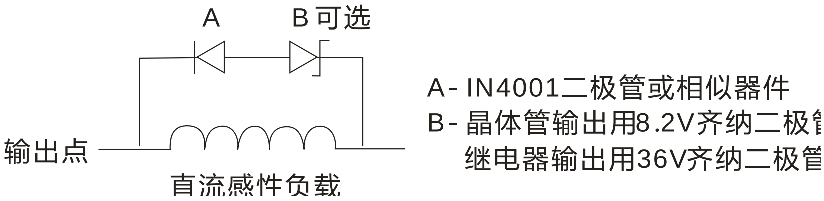


Fig3-4 Suppression circuit for DC load

Relay output for communication output and control of communication load

Communication output has internal protection for most applications, as relays can be used for DC or AC loads and therefore do not provide internal protection. Figure 3-5 shows an example of an AC load suppression circuit. In most applications, additional metal oxide variable resistors (MOVs) can limit peak voltage and protect the internal circuitry of M226ES. Ensure that the operating voltage of MOV is at least 20% higher than the normal line voltage.

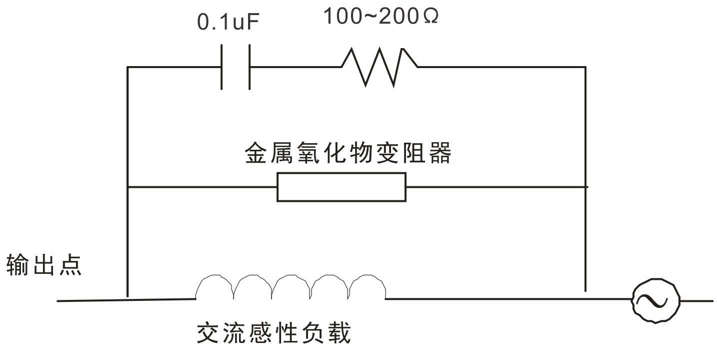


Fig3-5 Suppression circuit for AC load

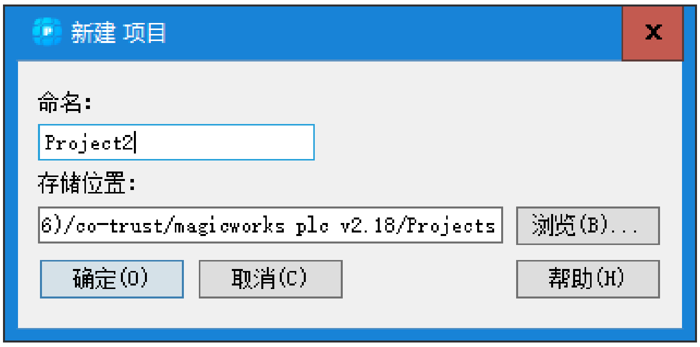
Simple Engineering Configuration

4

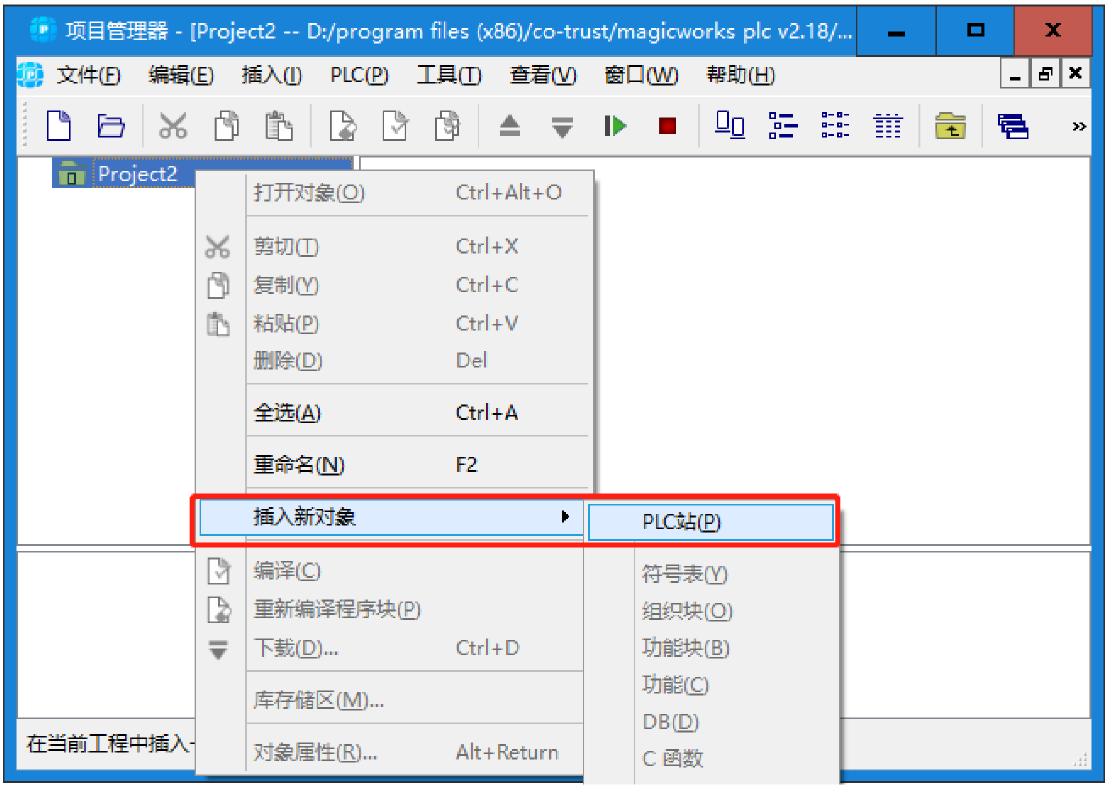
This chapter mainly introduces the upper computer software used to control M226ES and its basic usage.

4.1 New construction project

1. Start the MagicWorks PLC software, select "File" ->"New" in the "Project Manager" interface to create a new project, and name and select the storage address for the project in the pop-up window.



2. Right click on the project, select "Insert New Object" ->"PLC Station", and add a PLC station.



3. Select the PLC type as "M226ES" in the pop-up window and click "OK".

After completing the above steps, the establishment of the M226ES new PLC site has been completed in the programming software. For more detailed settings, please refer to the MagicWorks PLC User Manual. Download link: https://www.co-trust.com/Download/index.html

4.2 Establish communication

4.2.1 Select serial communication method

In the "Set PG/PC Interface" property dialog box, you can select and set the serial communication network supported by M226ES. The selected network will be used as an interface, and the default serial network communication interface that MagicWorks PLC can access is "PG/PC Cable (PPI)". The specific steps are as follows:

1) Double click the icon or click "Set PG/PC Interface" in the communication settings window.
2) Select the MagicWorks interface option, which defaults to "PC/PPI Cable (PPI)".

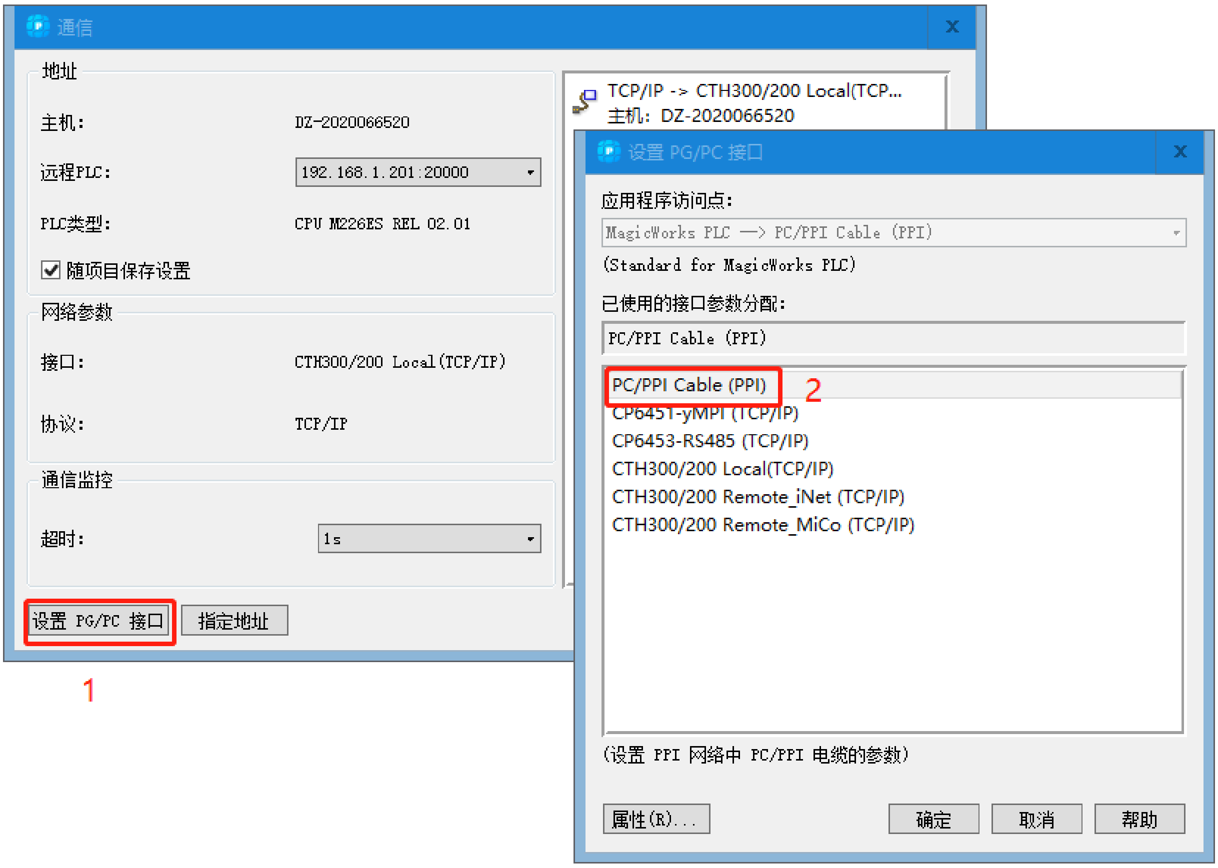
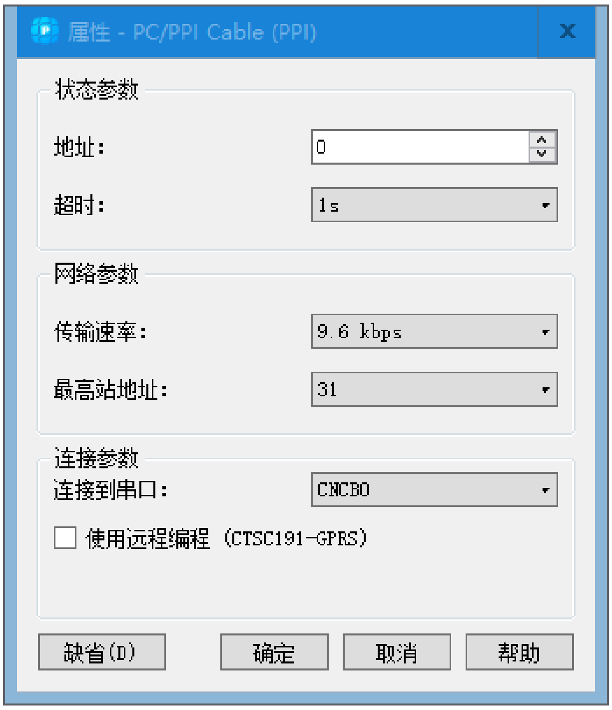


Fig4-1 MagicWorks PLC serial port network communication interface

Select "PC/PPI Cable (PPI)" in the "Set PG/PC Interface" dialog box, and then click the "Properties" button in the dialog box to set communication parameters.



1) Set the communication address of the PC/PPI cable to 0.

2) Select COM3 port.

3) The transmission baud rate is 9.6kbps.

Fig4-2 PC/PPI Cable (PPI) attribute setting interface

After setting up, click the "OK" button to return to the communication main interface, and then double-click "Double Click Refresh" to successfully search for the PLC.

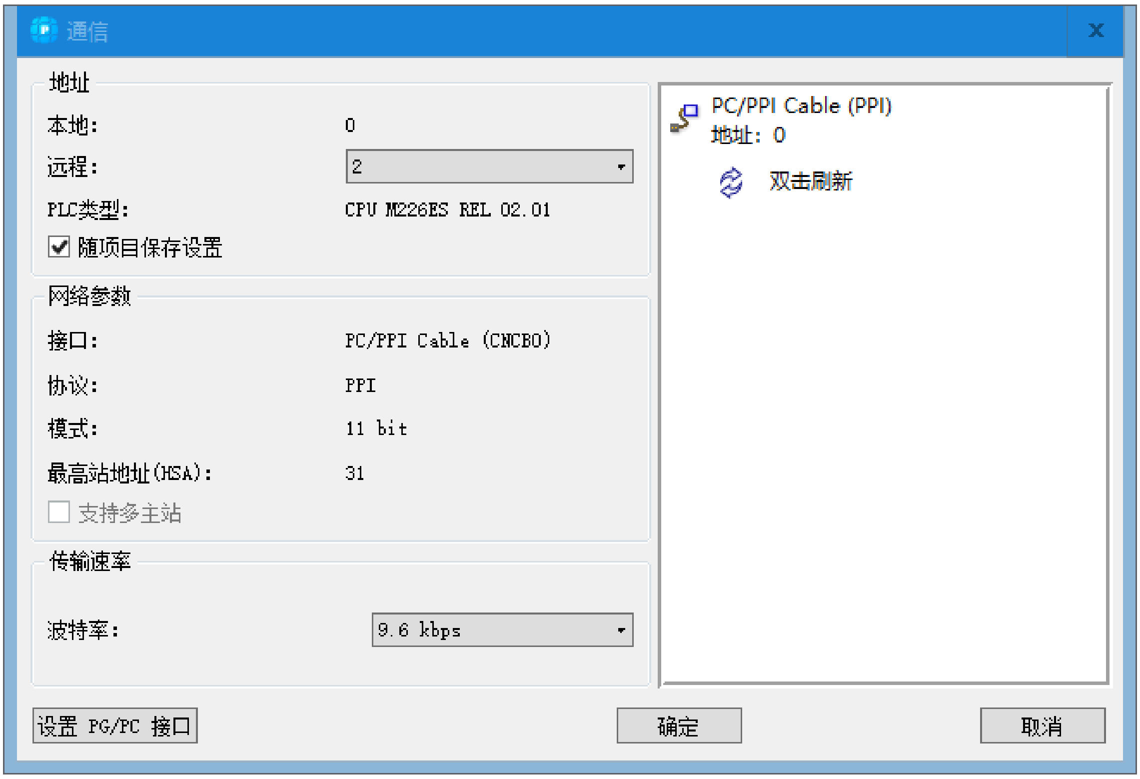


Fig4-3 Communication main interface

If communication is unsuccessful, please follow the steps below to investigate the specific cause:

1) Check the wiring

Ensure that the interface and cables are intact, please use the programming cable (CTS7-191-USB) provided by COTRUST Company, and pay attention to connecting the programming cable to the white port of the PLC (if the communication protocol has been changed before, please turn the PLC switch to STOP).

2) Check the driver

If using the programming cable provided by COTRUST, please install the corresponding version of the driver. The download link for the relevant driver installation package is: https://www.co-trust.com

3) Check communication settings

◆ Please select the PPI protocol (PC/PPI Cable).

◆ Please select the corresponding COM port to ensure that it is not occupied by other programs. The COM port should be within 10.

◆ Select the corresponding baud rate.

4.2.2 Choose Ethernet communication method

In the "Set PG/PC Interface" property dialog box, you can select and set the Ethernet communication network supported by M226ES. The selected network will be used as an interface, and the optional network interfaces are as follows:

- CTH300/200 Local (TCP/IP) series
- CTH300/200 Remote (TCP/IP) series

Taking CTH300/200 Local (TCP/IP) as an example, the specific steps are as follows:

(1) Double click the icon or click "Set PG/PC Interface" in the communication settings window

(2) Select 'CTH300/200 Local (TCP/IP)' from the listed network interface options

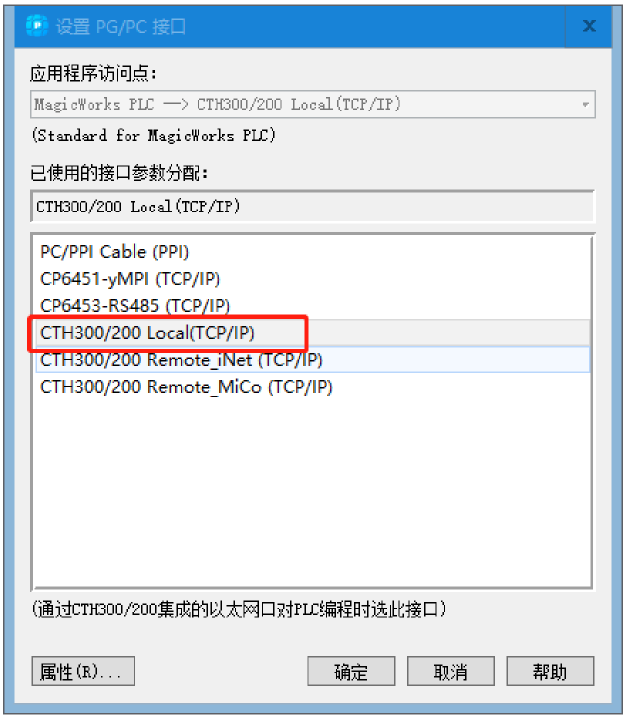


Fig4-4 MagicWorks PLC Ethernet Communication Interface

Establish communication with PLC using communication dialog box:

1、 Double click the refresh icon in the communication dialog box. MagicWorks PLC searches for and displays the icon of the connected CPU.

2、 After completing the search, select the station and click OK. If MagicWorks PLC cannot find the PLC, please first check if the cables are properly connected, then check the communication parameter settings, and repeat the above steps. After establishing communication with the PLC, set the computer IP to the same network segment as the PLC's IP, but the two IPs cannot be the same. As shown in the above figure, if the PLC's IP is 192.168.1.202, then the computer IP should also be 192.168.1.XXX (where XXX ranges from 1 to 255). This way, programming and downloading can be done. The specific process of setting the IP is as follows:

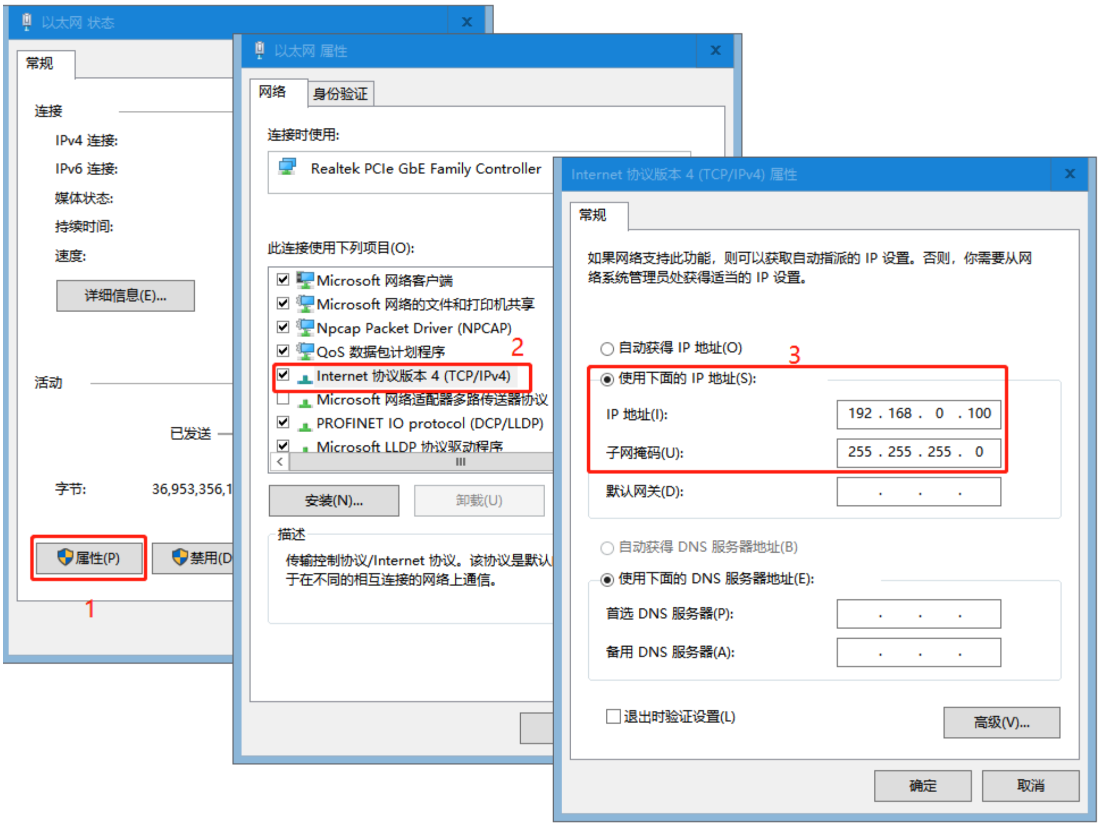


Fig4-5 Operating System Network Attribute Settings

4.2.3 Modify IP address in system block

After establishing communication with the PLC, if you need to modify the IP address of the CPU, please open the system block in MagicWorks PLC, modify the IP address, and select "Download System Block" in the download interface to make the modification effective. After successfully modifying the IP, you need to search for the CPU again to establish a connection. The modified IP is shown in the following figure:

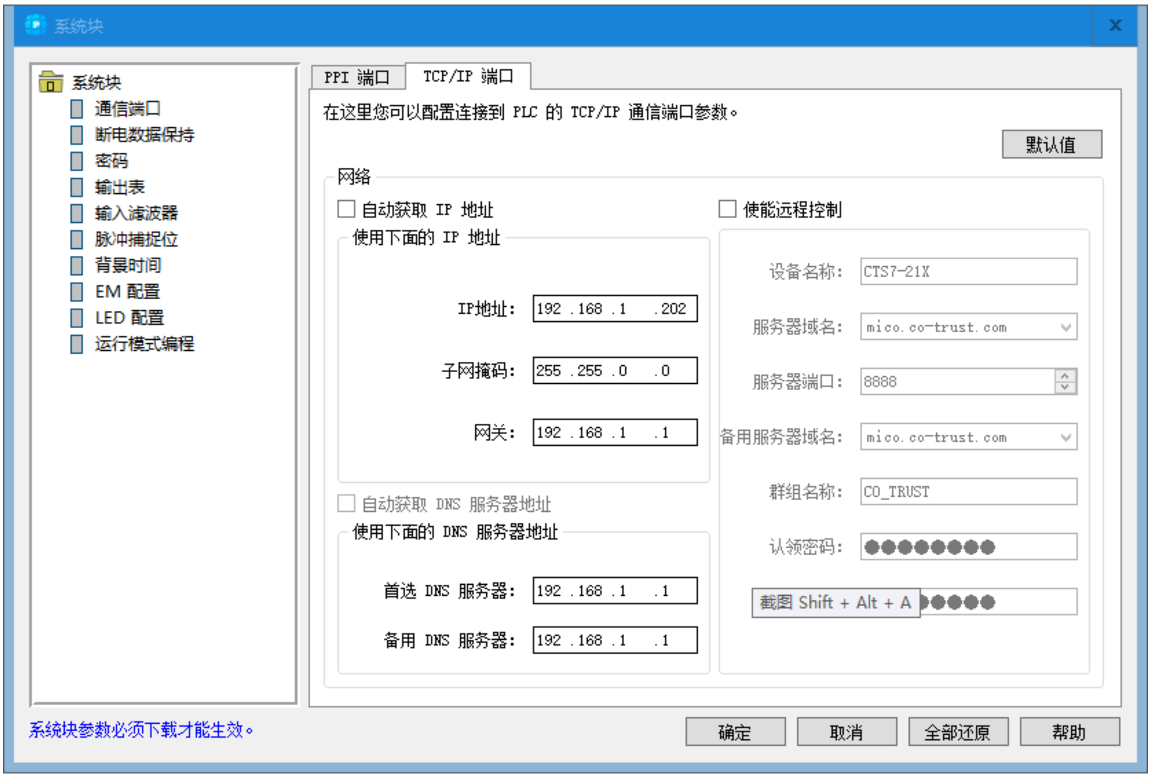


Fig4-6 System Block IP Setting Interface

Tips
If the automatic IP acquisition is unsuccessful, the system will assign a default IP of 192.168.1.202 to the PLC.

4.3 Create program segment

This section provides an example of the process of creating, downloading, and running a PLC program to help you understand and use M226ES. The following program example uses three instructions in a program segment, whose function is to shift the numerical byte transmitted to QB0 by one bit at 1-second intervals. In this example, a ladder diagram editor is used to input the program, and the process of compiling, downloading, and running the program is introduced.

The following is a ladder diagram and statement table program. The program comments listed in the statement table explain the logical relationship of the program in detail.

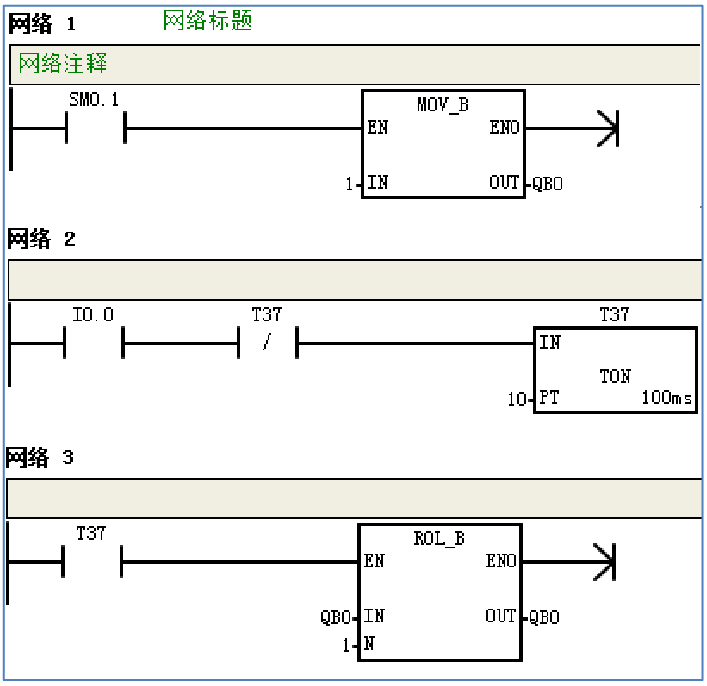


Fig4-7 M226ES Program Example

Program segment comments:

```
Network 1:
LD SM0.1//Enable first scan through SM0.1
MOVB 1, QB0//Move 1 to QB0
Network 2:
LD I0.0//Enable signal
AN T37//Timer signal cycle
TON T37, 10//Set timer T37 with a timing of 100ms x 10=1s
Network 3:
LD T37//Timer given enable pulse
RLB QB0, 1//QB0 moves one position to the left, and the PLC shows that the indicator lights
at each output point light up cyclically every second
```

4.3.1 Edit program

Select the PLC and set the relevant communication parameters in Maigicworks PLC (version 2.29 or above), then click on the program block icon to open the program editor, as shown in the following figure.

You can insert ladder diagram instructions from the instruction tree into the program editor by dragging or double clicking. There are shortcuts to some commands in the toolbar icons. After writing and compiling the program, you can download it to the M226ES series PLC.

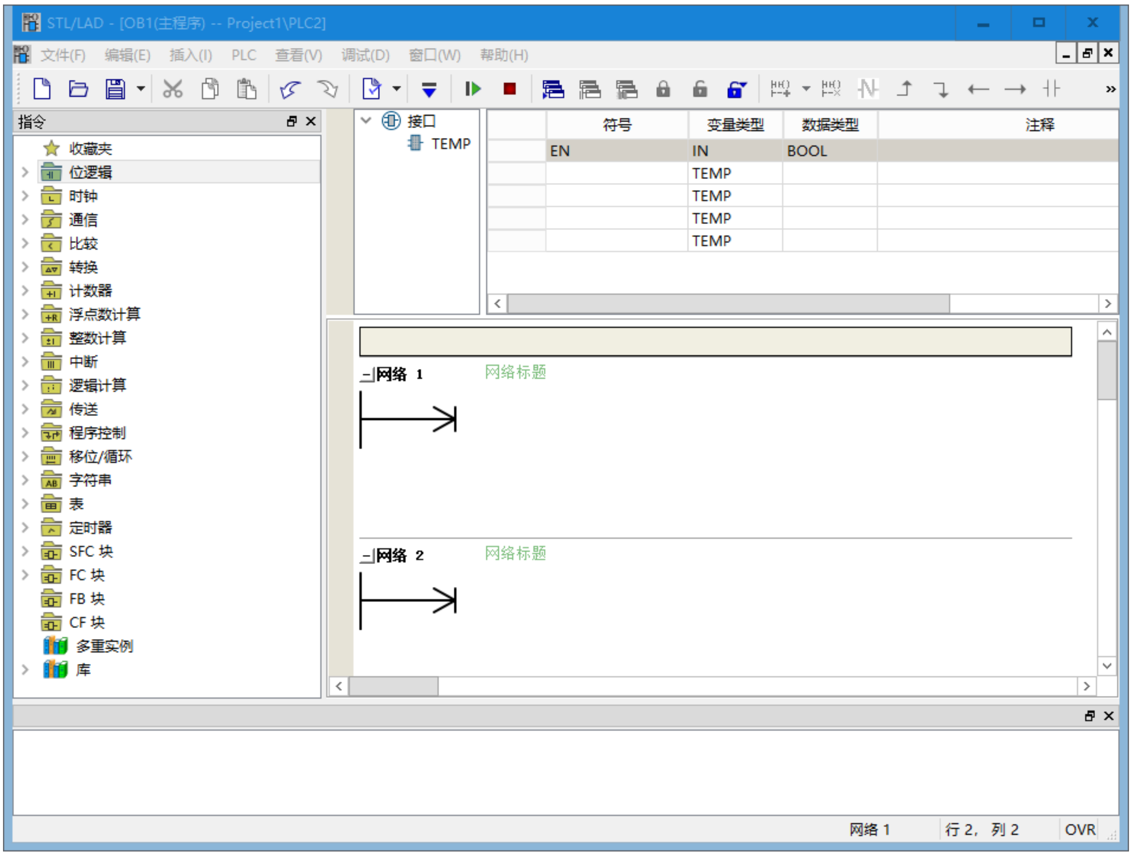


Fig4-8 Program Editor

4.3.2 Download program

Click on the download icon in the toolbar or select "PLC → Download" from the command menu to download the program, as shown in the following figure. Click 'OK' to download the program to the PLC. M226ES can perform read and write operations online, and even in running mode, it can continuously write instructions or parameter values without placing the PLC in stop mode.

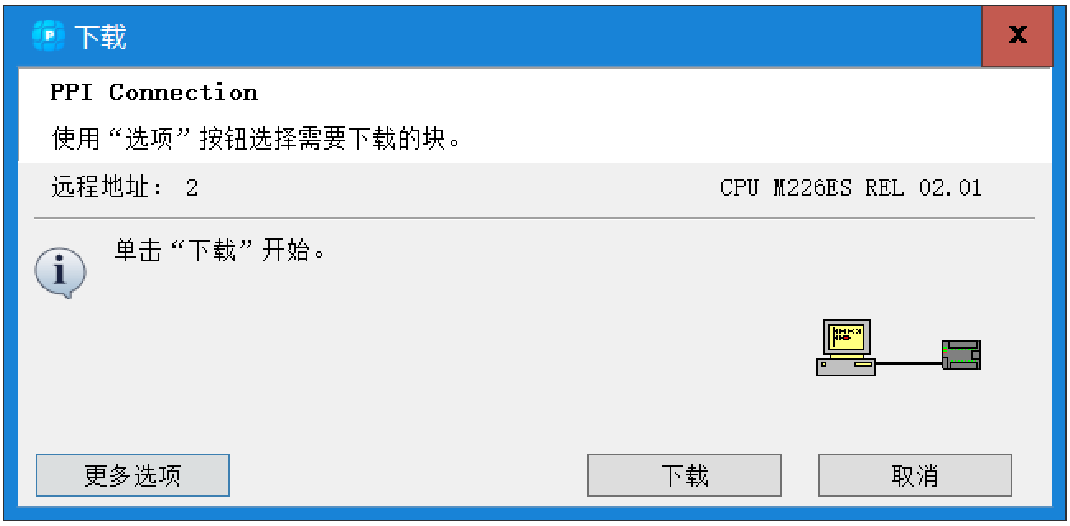


Fig4-9 Serial Communication Download Program (Image Update)

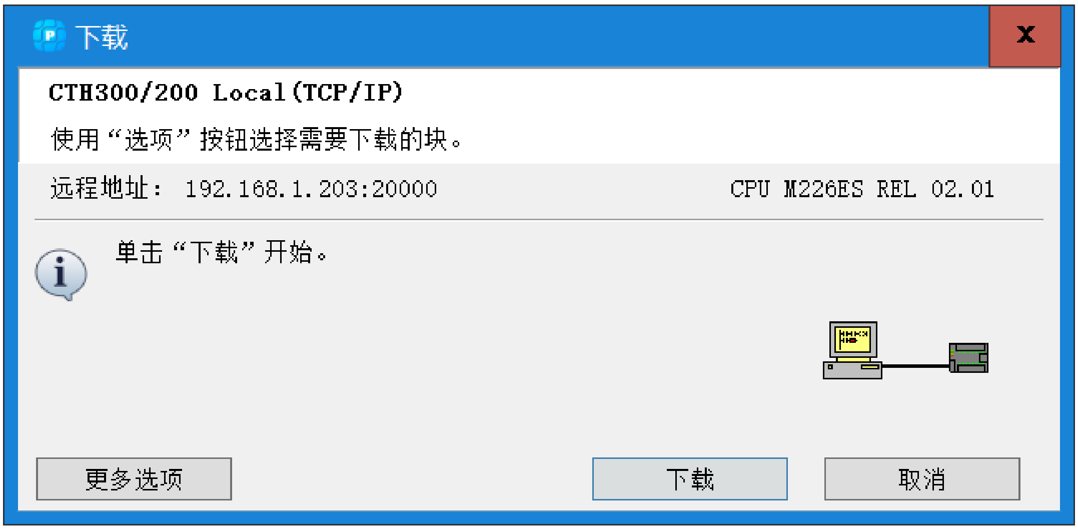


Fig4-10 Ethernet communication download program

4.3.1 Compile program

After the program segment input is completed, the program needs to be compiled:

1) Select the menu command "PLC → Compile" or click the "Compile" button

2) The output window at the bottom will display the compilation status. If there is a prompt for compilation error, please click on the window prompt to view the specific details. If the compilation is correct, the window will display as follows:

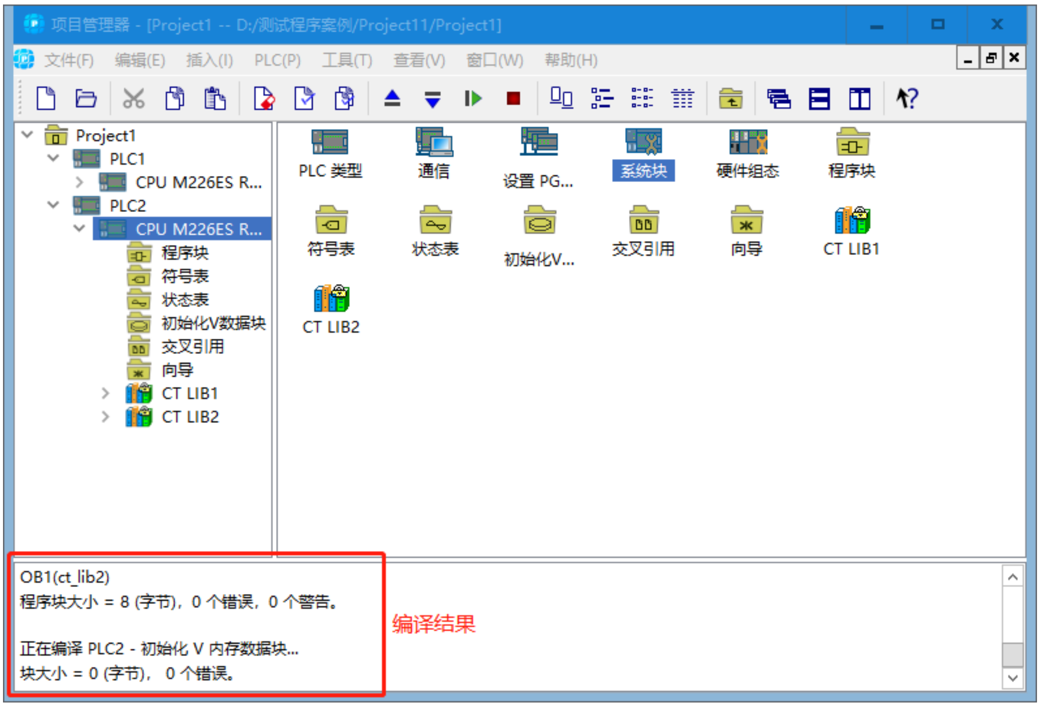


Fig4-11 Program compilation result window

Save project:

1) Select the menu command "File → Save As" from the menu bar.

2) Enter the project name in the Naming dialog box.

3) Select the storage location to download the program to the M226ESL series PLC.

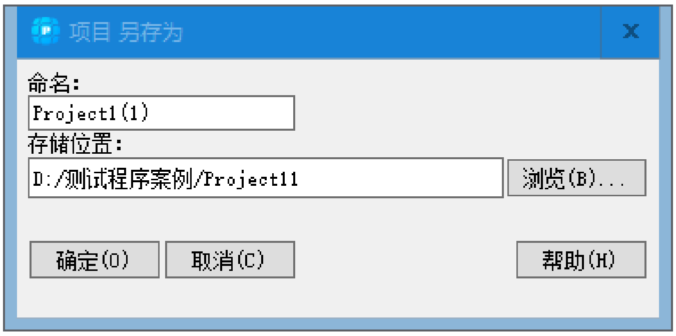


Fig4-12Storage program

4.3.2 Run PLC

When the mode switch of M226ES is set to RUN, MagicWorks PLC software can switch M226ES to RUN mode. When M226ES is in RUN mode, execute the program:

1) Click on the run icon in the toolbar or select PLC → RUN from the command menu.

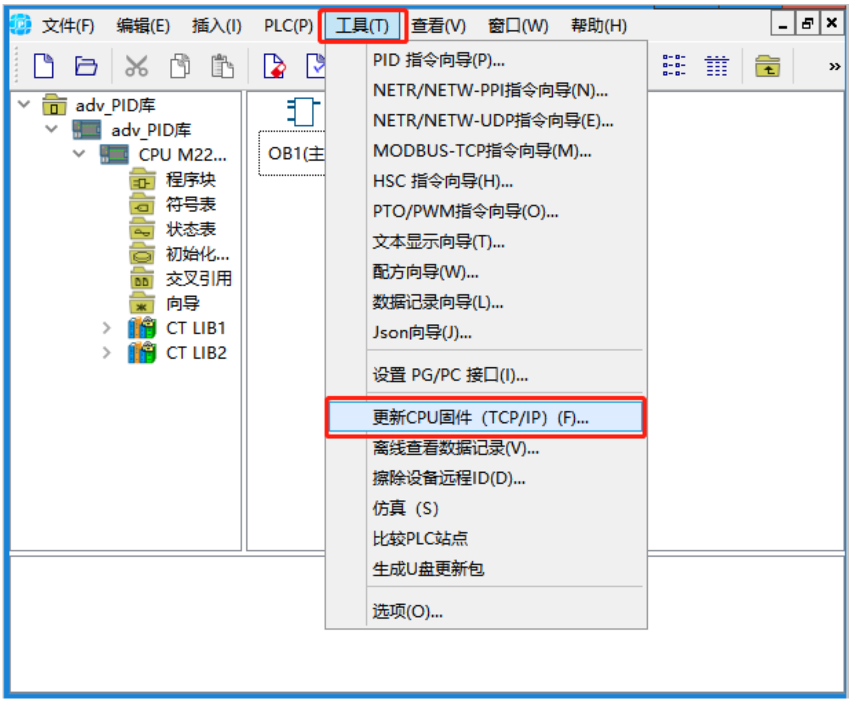
2) Click 'Yes' to enter the running mode. When M226ES enters run mode, the CPU will execute the program, and Q0.0-Q0.7 will light up cyclically from left to right at 1-second intervals, indicating that they are performing a left shift action on the values in Q memory.

Fig4-13 Run the program

You can monitor the program by selecting Project Tree → Status Table. MagicWorks PLC displays the execution results. To terminate the program run, you can click on the icon or select the menu command "PLC → Stop" to put M226ES into stop mode.

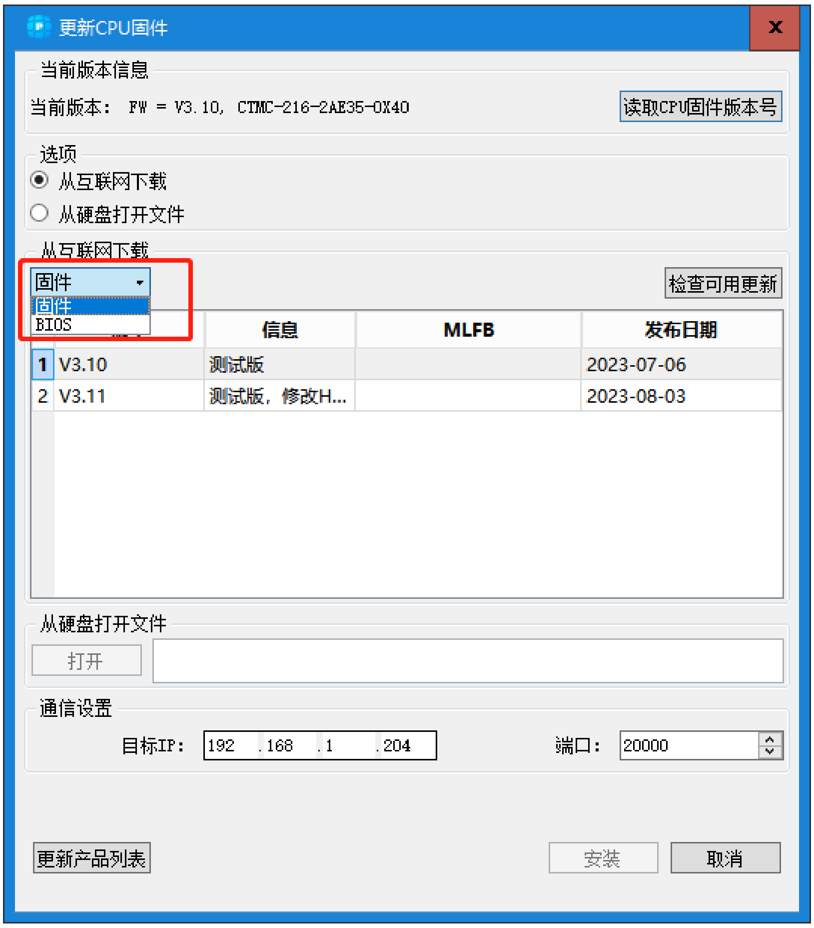
4.4 Update CPU firmware and BIOS online

Before performing online firmware and BIOS updates, it is necessary to ensure that the CPU is connected to the network, as shown in the following figure. In the programming software project manager interface, select "Tools" ->"Update CPU Firmware (TCP/IP)" to open the "Update CPU Firmware" interface.

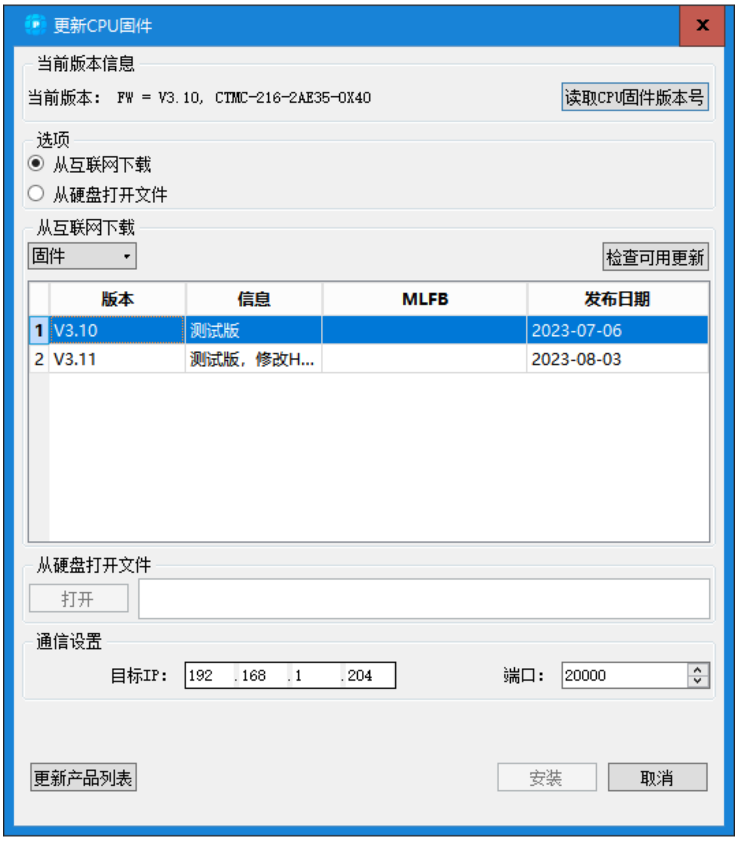


1、 Update firmware online

1. Select firmware

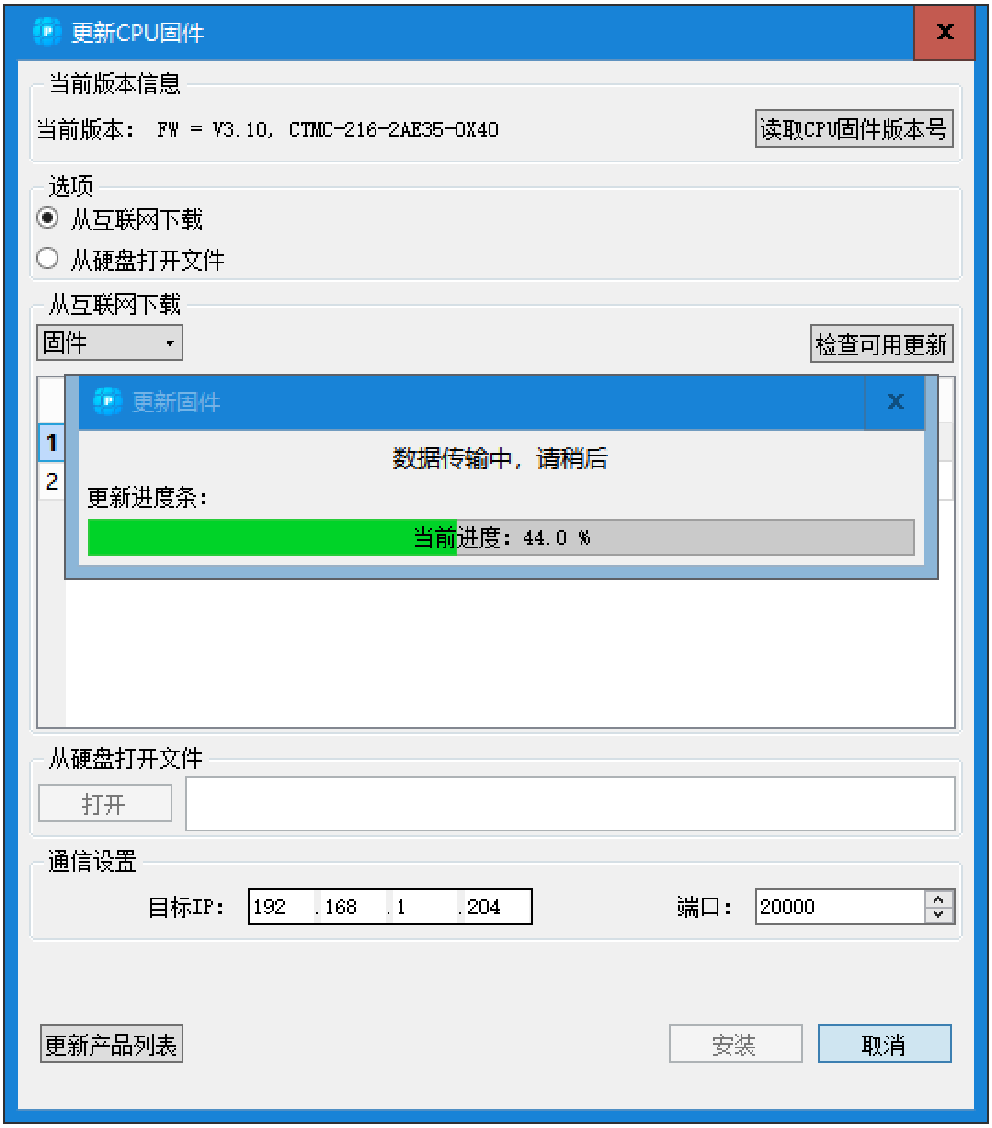


2. Select "Download from the Internet" in the "Update CPU Firmware" interface, click "Check for Available Updates", and the information window will display the firmware available for the current PLC, as shown in the following figure. Select the firmware to be updated and click "Install", and the firmware will be automatically downloaded from the Internet.



3. After the download is complete, the following pop-up window will appear. Select "Yes" and the CPU will automatically load the firmware.

4. Waiting for firmware to load.

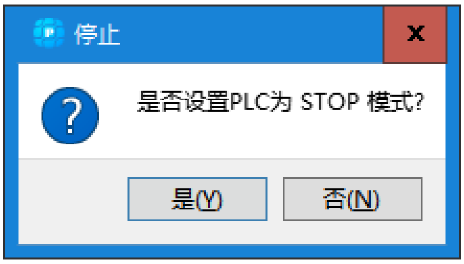


5. After the firmware is loaded, the following prompt will appear. Click "OK". Attention, do not operate or disconnect the power supply of the PLC before the automatic restart is completed!

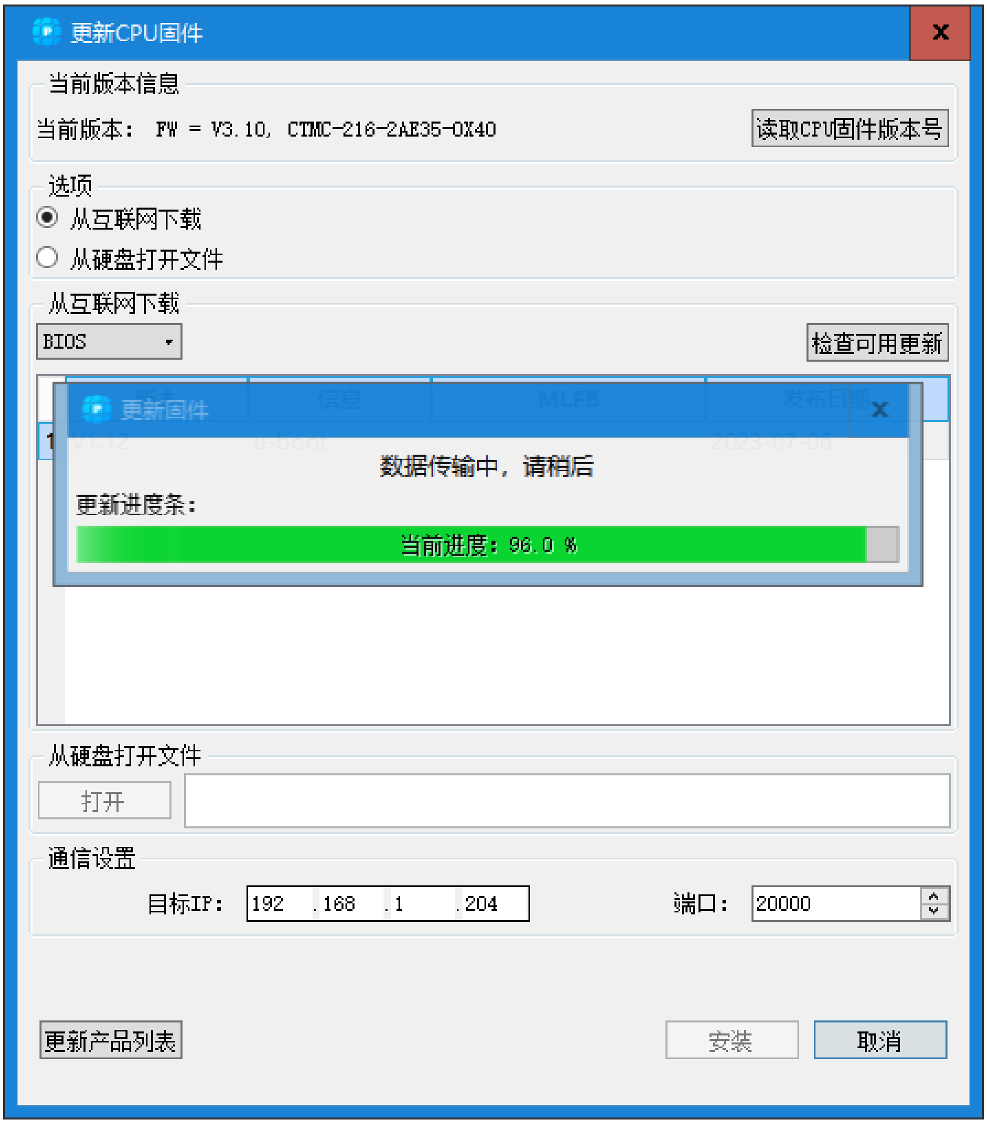
2、 Update BIOS

1. As shown in the figure below, select the BIOS to be updated (the program to boot the firmware) and click "Install". The BIOS will be automatically downloaded from the Internet.

After the download is complete, the following pop-up window will appear. Select "Yes" and the CPU will automatically load the BIOS.



Waiting for BIOS to load.



After BIOS loading is complete, the following prompt will appear. Click "OK". Attention, do not operate or disconnect the power supply of the PLC before the automatic restart is completed!


更新CPU固件
当前版本信息
当前版本： FW = V3.10, CTMC-216-2AE35-0X40
读取CPU固件版本号
选项
从互联网下载
从硬盘打开文件
从互联网下载
BIOS
检查可用更新
版本
信息
MLFB
发布日期
更新CPU固件
CPU固件下载成功，请等待PLC自动重启（当PLC指示灯停止闪烁时更新成功）。
确定
从硬盘打开文件
打开
通信设置
目标IP： 192 . 168 . 1 . 204
端口： 20000
更新产品列表
安装
取消

Communication Use Cases

5

This chapter provides a detailed introduction to the various functions of M226ES and provides specific examples in conjunction with the software.

5.1 UDP_PPI communication

This section will demonstrate the detailed operation steps of UDP_PPI communication (Ethernet communication port) through a specific example.

In the UDP_PPI communication network, when the CPU acts as a slave, communication is independent of the entire large cycle, that is, receive and return. When the CPU serves as the main station, it is controlled by user programs for transmission and reception.

M226ES can serve as a master or slave, using network read and write commands UDP_NETR/UDP_NETW or NETW/NETR to communicate with other devices within the local area network through UDP_PPI. Meanwhile, HMI can also serve as the UDP_PPI master station, with M226ES as the slave station for UDP_PPI communication.

Tips

Please refer to the MagicWorks PLC User Manual for instructions on how to use the UDP read-write command wizard. For relevant manuals and library files, please visit the official website of COTRUST at: https://www.co-trust.com

5.1.1 Example components

Table5-1 Example components of UDP_PPI communication

assembly	function
Programming equipment PG/PC	Install MagicWorks PLC software (V2.29 and above) to configure, program, and debug the PLC
PLC	Two M226ES, one serving as the UDP_PPI master station and the other as the UDP_PPI slave station, communicate with each other through the EtherNET interface for UDP_PPI communication
Standard Ethernet cable	·Connect the main station with programming devices ·Connect UDP_PPI master station and UDP_PPI slave station

5.1.2 network connections

1. Connect the programming device to the main station using a standard Ethernet cable, and program the UDP_PPI main station through the programming device:

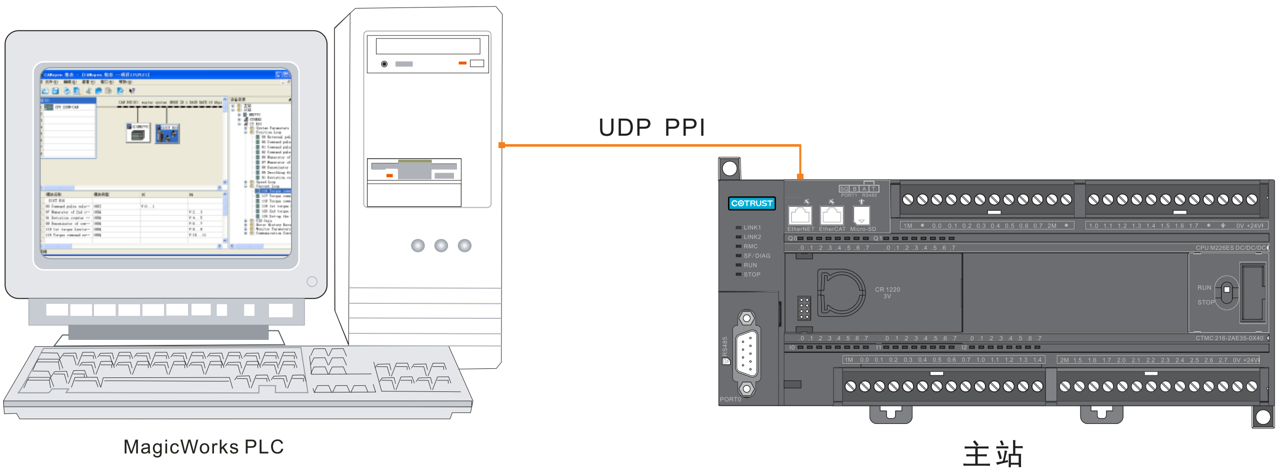


Fig5-1 UDP_PPI communication connection

2. Connect the UDP_PPI master station and UDP_PPI slave station using a standard Ethernet cable. The

UDP_PPI master station performs write operations to write data from a specified address to the UDP_PPI slave station, and then reads data from the slave station through UDP_PPI to achieve UDP_PPI communication

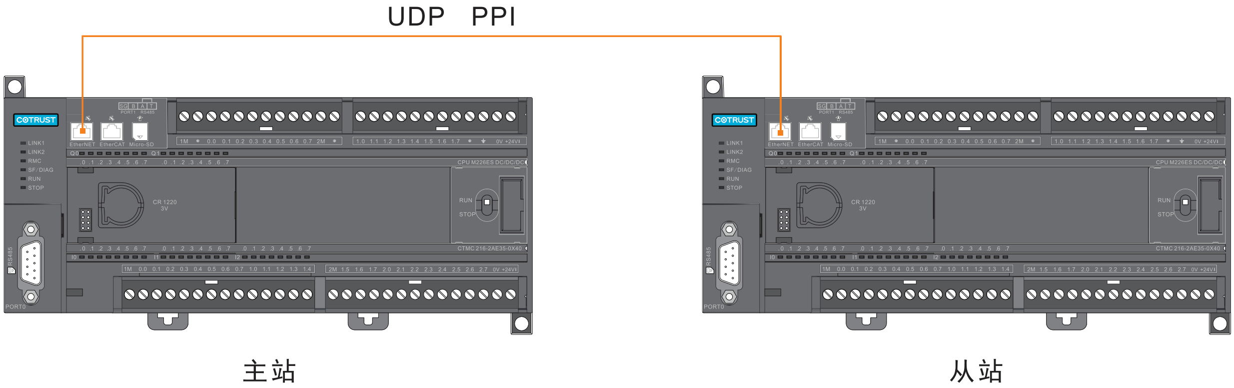


Fig5-2 Communication between main stations

5.1.3 Operation steps

Step 1: Connect the cable

Connect the PC, UDP_PPI master and slave stations using standard Ethernet cables.

Step 2: Set up communication

Create a new project in MagicWorks PLC, add PLC stations to the project, and connect M226ES to the PC for communication.

Step 3: Program the UDP_PPI master station

Taking UDP network read and write as an example, you can perform network read and write operations on UDP_PPI master station through one of the following two methods.

1. Program UDP_PPI master station through UDP network read and write commands UDP_NETR/UDP_NETW

Table5-2 TABLE parameters for UDP_NETR/UDP_NETW instruction:

bit7	bit6	bit5	bit4	bit3~0	Byte offset
D	0	E	0	error code	0
The first byte of the remote station IP address					1
The second byte of the remote station IP address					2
The third byte of the remote station IP address					3
The fourth byte of the remote station IP address					4
Port number high byte					5
Low byte of port number					6
The first byte of<I, Q, M, V, DB>(4 bytes in total) pointing to the target area pointer of the remote station					7
The second byte of the pointer to the target area of the remote station					8
The third byte of the pointer pointing to the target area of the remote station					9

The fourth byte of the pointer to the target area of the remote station	10
Data length	11
Data byte 0	12
Data byte 1	13
...	...
Data byte 199	211

D: Completed (function completed), 0=incomplete, 1=completed

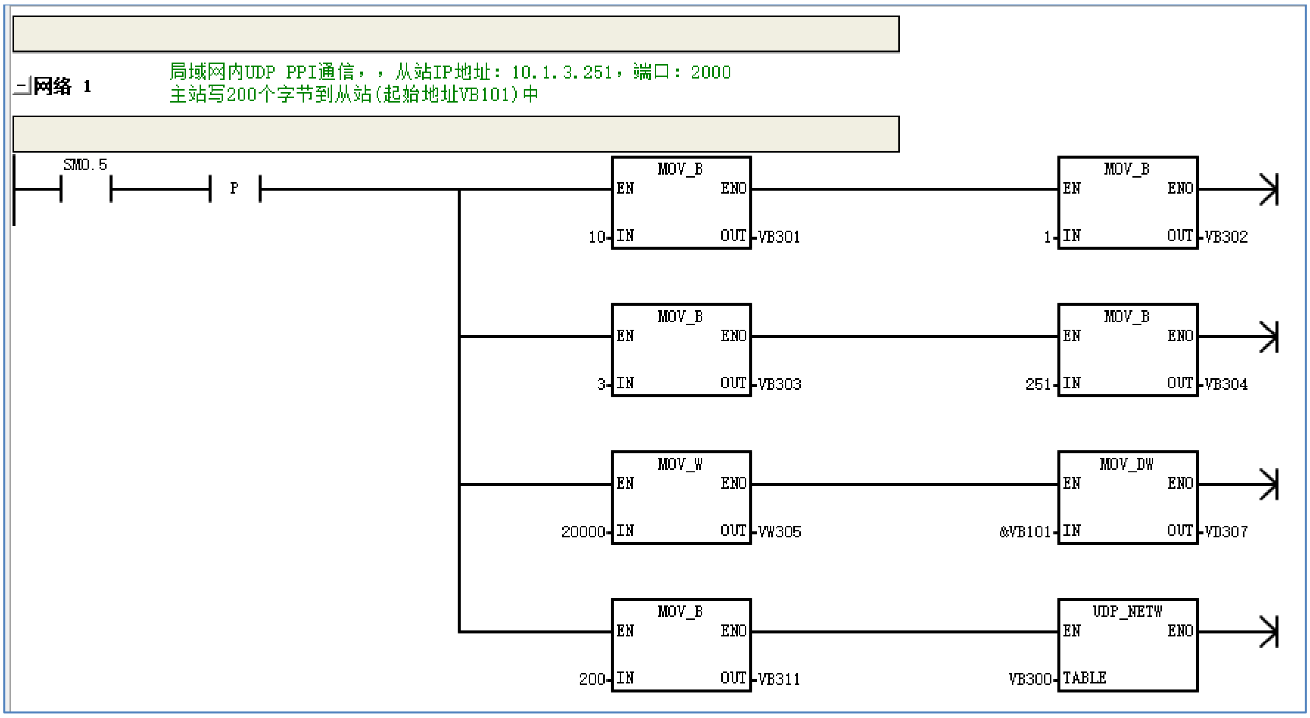
A: Active (function joined), 0=non active, 1=active

E: Error, 0=no error, 1=error

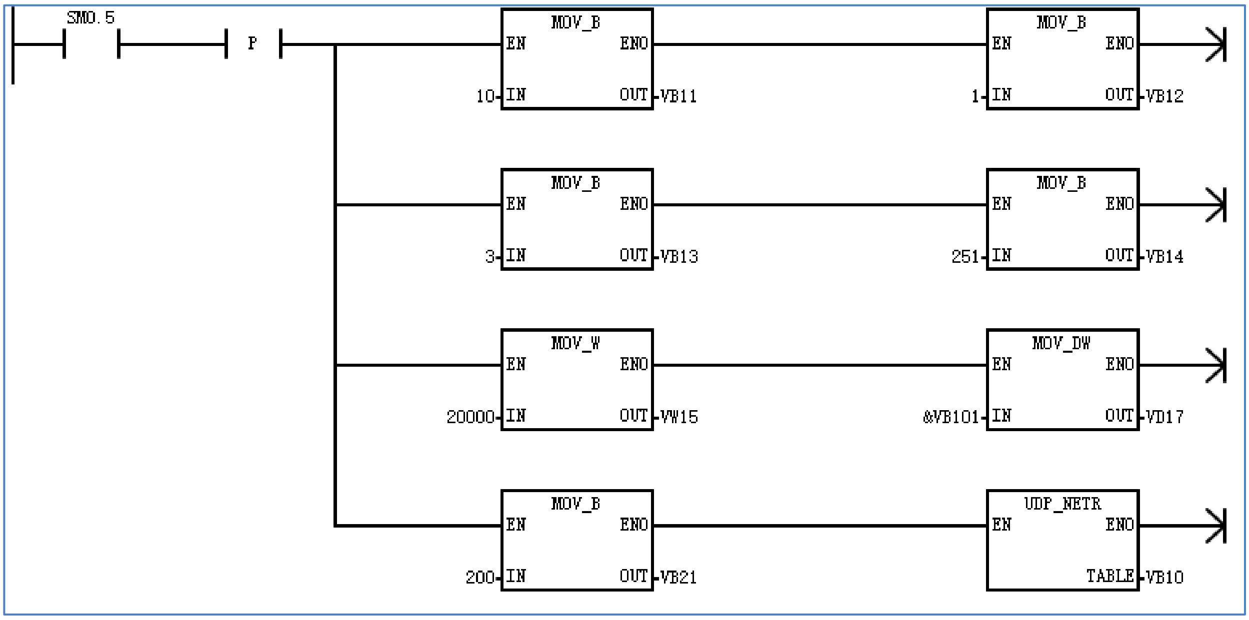
The lower four bits of the first byte in the TABLE parameter table are error codes, defined as follows:

error code	describe
0	No mistake
1	Timeout error. Remote station does not respond
2	Receiving error. There are checksum, frame, or checksum errors in the response
3	Offline error. Conflicts caused by duplicate station addresses or faulty hardware
4	Queue overflow error. More than 8 UDP_NETR/UDP_NETW boxes are activated
6	Illegal parameters. The UDP_NETR/UDP_NETW table contains an illegal or invalid value
7	No resources available. The remote station is busy (uploading or downloading sequences)
8	7th layer error. Violation of application protocol
9	Information error. Data address error or incorrect data length

Network 1: Write 200 bytes of data from the UDP_PPI master station (starting address: VB312) to the UDP_PPI slave station (IP: 10.1.3.251, starting address: VB101).

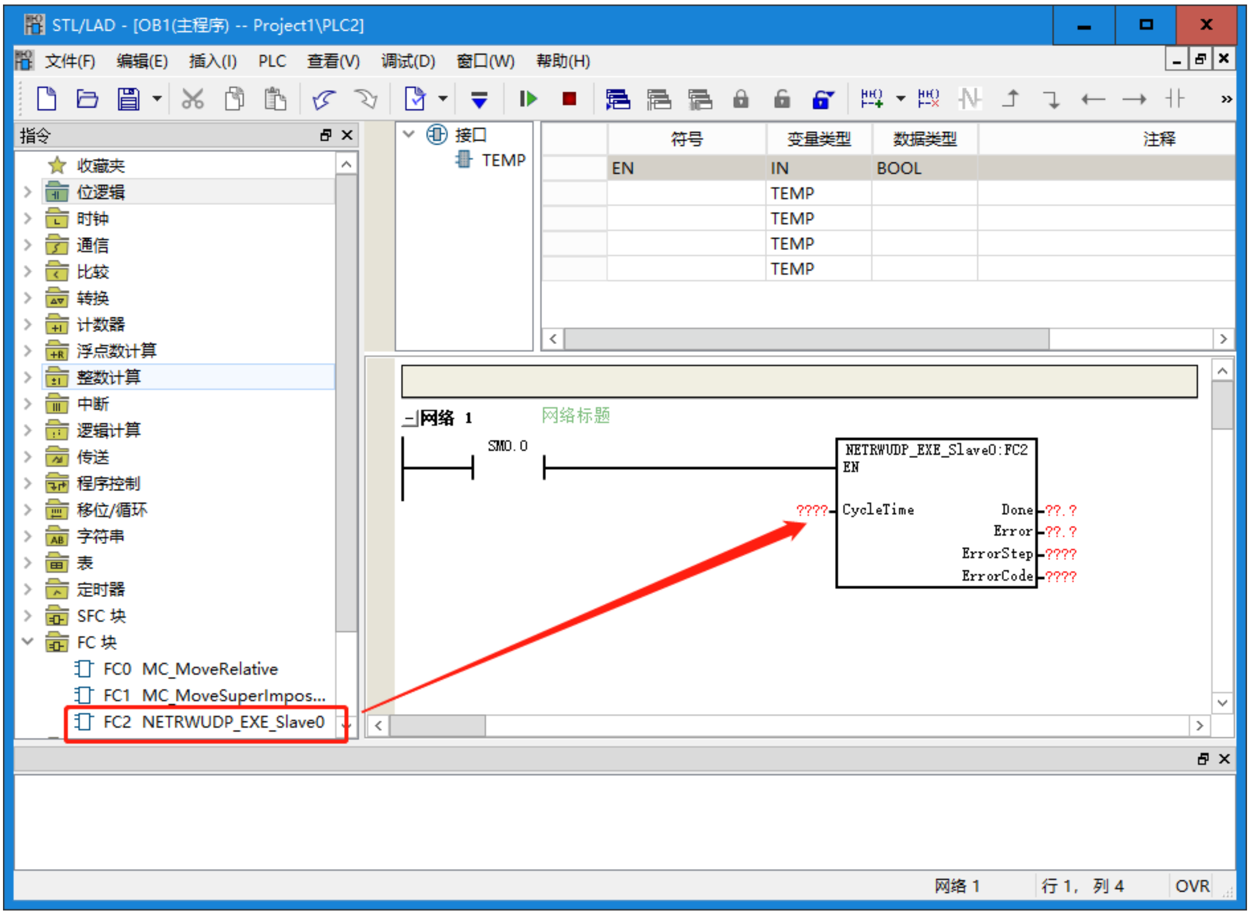


Network 2: Read 200 bytes of data from UDP_PPI slave station (IP: 10.1.3.251, starting address: VB101) into UDP_PPI master station (starting address: VB22).



2. Program UDP_PPI master station using the NETR/NETW-UDP command wizard

Expand M226ES in the project tree, select "Wizard" in the sub options, and then double-click "NETR/W-UDP Wizard" in the workspace on the right to open the wizard dialog box. You can then gradually configure the NETR/NETW-UDP command wizard. After configuration, call the FC in OB1 network, as shown in the following figure:



After completing the program editing according to method 1 or 2 above, compile and download the program to the UDP_PPI main station device (which must be called using SM0.0).

Step 4: UDP_PPI master station and UDP_PPI slave station interconnect and **communicate**

1) Connect the power supply to the UDP_PPI master and slave stations.

2) Connect the EtherNET communication ports of the UDP_PPI master and slave stations using standard Ethernet cables, and then they can perform UDP_PPI communication.

If you want to configure multiple read and write operations, it is recommended to use the Modbus UDP wizard of MagicWorks PLC. For specific operations, please refer to the MagicWorks PLC User Manual on the official website of COTRUST.

5.1.4 UDP_PPI communication slave address mapping

UDP_PPI communication uses ordinary network read-write, and multiple registers can be configured during read-write operations, mapping from station address to direct mapping.

5.2 Modbus TCP communication

This section demonstrates the detailed operation steps of Modbus TCP communication function through a specific example.

In Modbus TCP communication networks, the CPU acts as a slave and communication is independent of the entire large cycle, i.e. receive and return. The CPU serves as the main station, and its transmission and reception are controlled by user programs.

When using the EtherNET communication port of M226ES for Modbus TCP communication, no configuration is required to achieve Modbus TCP communication. If you need to change the IP address, please refer to section 4.2.3 to modify the IP address in the system block. When M226ES communicates with other slave stations as a Modbus TCP master, it is necessary to use the Modbus TCP wizard or Modbus TCP single read/write command library to configure the master station communication. The Modbus TCP communication port number is fixed at 502, and the port number is configured in the wizard.

	Tips Please refer to the MagicWorks PLC User Manual for instructions on using the TCP Read/Write Command Wizard. For relevant manuals and library files, please visit the official website of COTRUST at: https://www.co-trust.com

5.2.1 Example components

Table5-3 Example components for Modbus TCP communication

assembly	function
Programming equipment PG/PC	Install MagicWorks PLC software (V2.29 and above) to configure, program, and debug the PLC.
PLC	This example uses two PLCs, one as a Modbus TCP master and the other as a Modbus TCP slave, for Modbus TCP communication through the EtherNET interface.
Standard Ethernet cable	Connect the main station with programming devices. Connect Modbus TCP master and Modbus TCP slave.

5.2.2 network connections

Connect the programming device to the master station using a standard Ethernet cable, and then call the CT_Sodbus TCP library instruction provided by COTRUST on the programming device to program the Modbus master station:

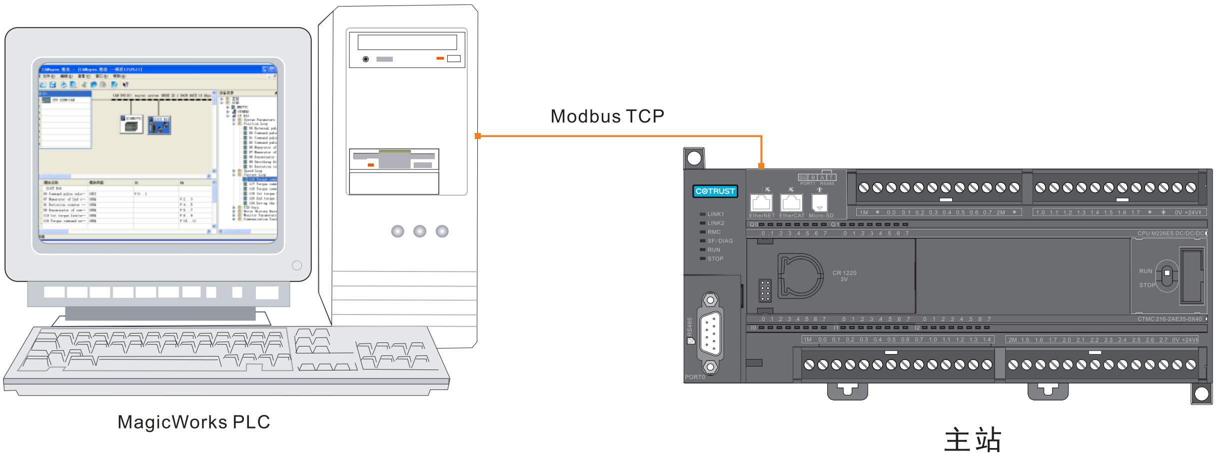


Fig5-3 Modbus TCP communication connection

Connect Modbus TCP master and Modbus TCP slave using standard Ethernet cables. The Modbus TCP master performs write operations, writes data from a specified address to the Modbus TCP slave, and then reads data from the slave through Modbus TCP to achieve Modbus TCP communication

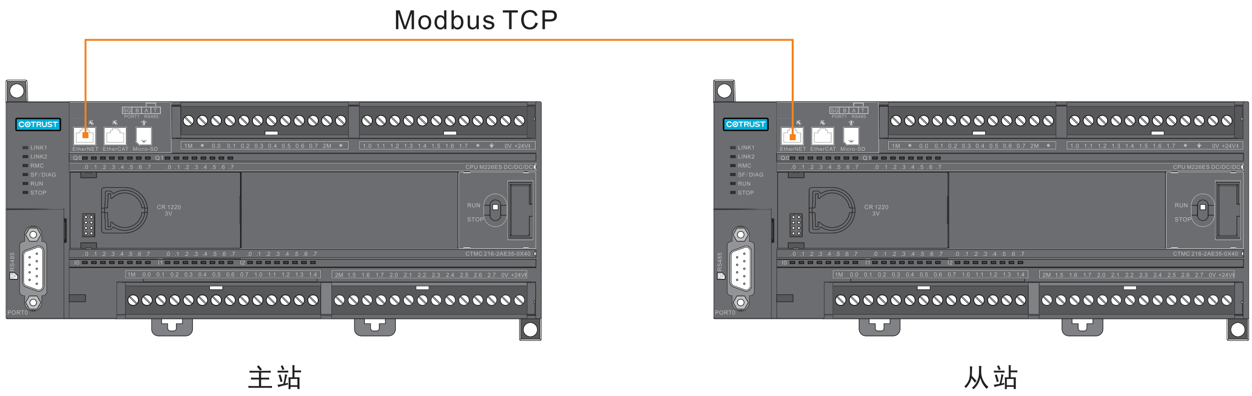


Fig5-4 Communication between master and slave stations

5.2.3 Operation steps

Step 1: Physical wiring

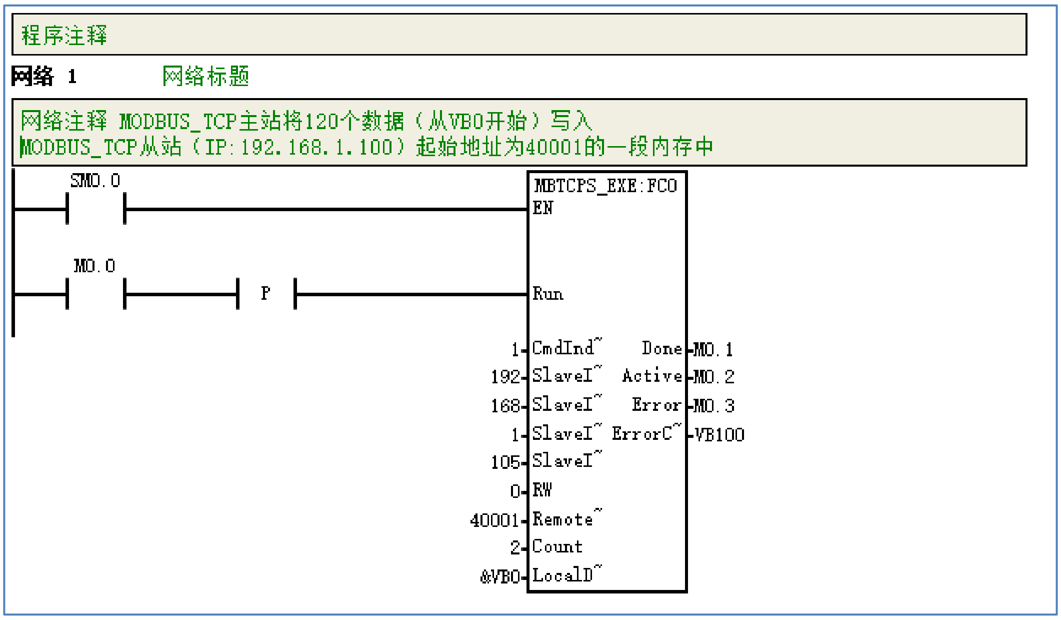
Connect PC, Modbus TCP master and slave using standard Ethernet cables

Step 2: Set up communication

Create a new project in MagicWorks PLC and add a PLC station to the project. Refer to section 4.2 to establish a communication connection with the PLC master station.

Step 3: Program the Modbus master station

1) In the MagicWorks PLC program block, program the Modbus TCP master station (communication can be configured through the Modbus TCP library or Modbus TCP wizard, and in this example, the Modbus TCP library is used for programming). The following instruction is the "MBTCPs_EXE" library instruction in the " Ct_Mbus_master_tcp_single " library file:



If you want to configure multiple read and write operations, it is recommended to use the Modbus TCP wizard of MagicWorks PLC. For specific operations, please refer to the MagicWorks PLC User Manual on the official website of COTRUST.

2) After programming is completed, compile and download the program to the Modbus TCP master device.

Table5-4 Definition of the "MBTCPs_EXE" library instruction:

Symbol	IN/OUT	data type	notes
RUN	IN	BOOL	Communication start bit, please use edge trigger
CmdIndex	IN	BYTE	The sequence number for calling MBTCPs_EXE must not be repeated at any point, with a range of 1-255
SlaveIP0	IN	BYTE	The first byte of the slave address, for example, if the slave address is 192.168.1.202, the first byte is 192
SlaveIP1	IN	BYTE	The second byte of the slave station address
SlaveIP2	IN	BYTE	The third byte of the slave station address
SlaveIP3	IN	BYTE	The fourth byte of the slave station address
RW	IN	BYTE	Read/write flag, read=0. Write=1
RemoteAddr	IN	DWORD	Remote data address (such as 40001)
Count	IN	WORD	Number of read-write units, effective range: 1-120 words or 1-1920 bits
LocalDataPtr	IN	DWORD	Local data pointer (such as&VB0)
Done	OUT	BOOL	Complete bit, 0=not yet completed or not started. 1=Operation completed
Active	OUT	BOOL	Activation bit, 0=not activated or completed. 1=Operation activated and not completed
Error	OUT	BOOL	Error bit, 0=no error, 1=there is an error
ErrorCode	OUT	BYTE	Error code, valid when the completion bit is 1, details are shown in the table below

Table5-5 The definition of the error code at the output of the Error Code is as follows:

error code	explain
0	No errors
1	The number of connections has reached its maximum value
2	Establishing connection
3	Receive timeout, no response from the slave station
4	The requested parameters are incorrect
5	Instruction not enabled
6	The connection is busy (such as activating multiple MBTCPs_EXE simultaneously)
7	Response error
8	Failed to create connection
9	Connection already exists
10	Connection does not exist
11	Unplug the network cable

Step 4: Modbus TCP master and Modbus TCP slave interconnect for **communication**

When configuring Modbus TCP slave, the slave IP must correspond to the IP programmed by the master. To change the slave IP, open the system block in MagicWorks PLC and modify the IP address, as shown in the following figure. In addition, when conducting ModBus TCP communication between PLCs, the slave port number is fixed at 502, which is the default specification of the system.

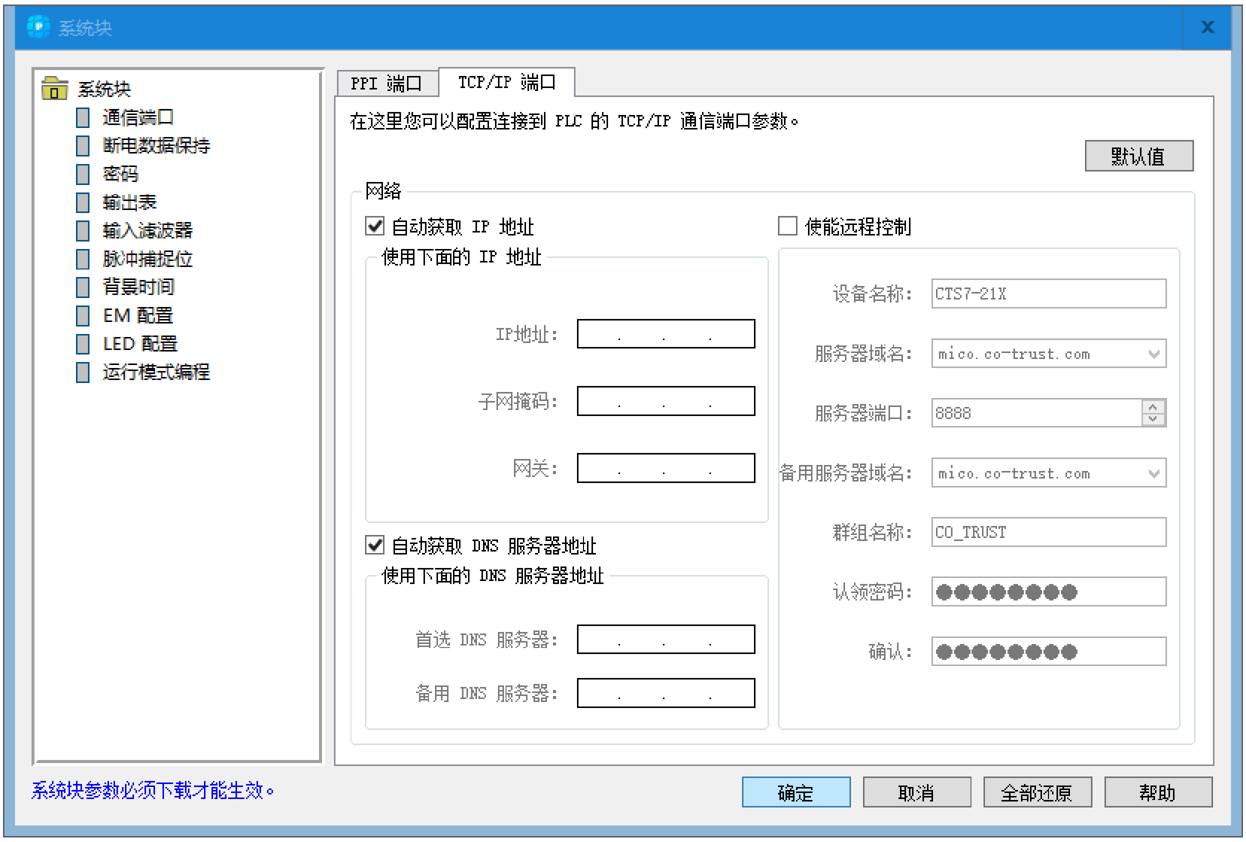


Fig5-5 System Block Settings

Step 5: Debugging

After following the above steps in sequence, connect the Modbus master and slave stations using a standard Ethernet cable, and then read the 120 data starting from 40001 from the Modbus TCP slave station through the status table of MagicWorks PLC software. If the content of this segment is consistent with the data in the corresponding memory of the Modbus TCP master station, it indicates successful Modbus TCP communication.

5.2.4 Modbus TCP Slave Address Mapping

Modbus addresses typically consist of 5 or 6 character values containing data types and offsets. The first or two characters determine the data type, and the last few characters are a corresponding value that matches the current data type.

Modbus TCP slaves support the following addresses (M-zone addresses are internally defined and only support being accessed as slaves):

Table5-6 The slave address mapping corresponding to the TCP protocol

Modbus slave address	Modbus slave address
000001	Q0.0
000002	Q0.1
000003	Q0.2
...	...

008191	Q1023.6
008192	Q1023.7
010001	I0.0
010002	I0.1
010003	I0.2
...	...
018191	I1023.6
018192	I1023.7
020001	M0.0
020002	M0.1
020003	M0.2
...	...
028191	M1023.6
028192	M1023.7
030001	AIW0
030002	AIW2
030003	AIW4
...	...
034096	AIW8190
040001	VW0
040002	VW0+2
...	...
04xxxx	VW0+2 x (xxxx-1)

5.3 CANfree communication

This section will demonstrate the detailed operation steps for implementing CANfree communication (Ethernet communication port) through a specific example of a charging station.

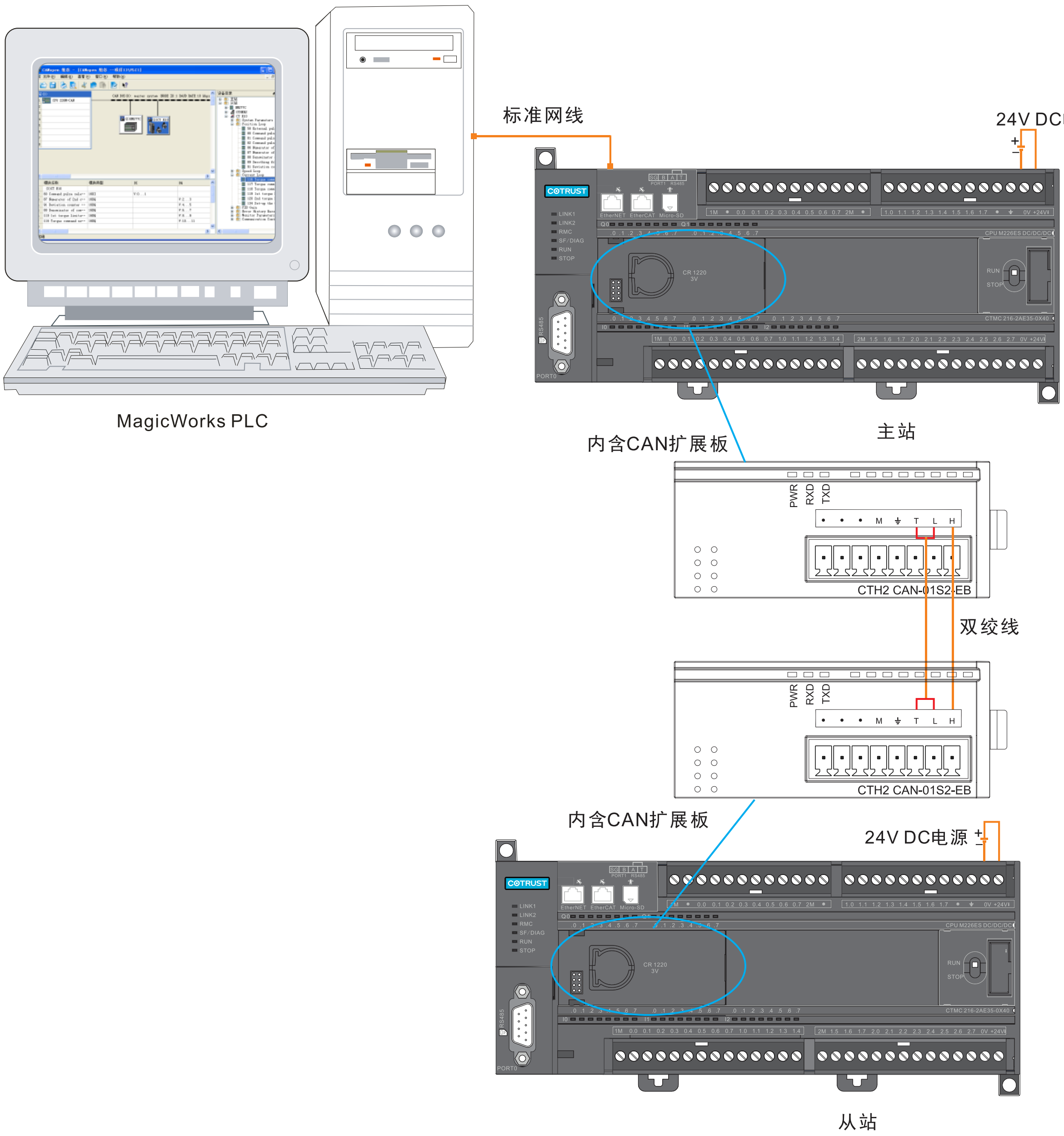
When conducting CAN communication, the controller needs to call the CAN communication command library to select the CAN communication port, complete the transmission and filtering of CAN communication message data. The CPU needs to be connected to a third-party charging module (CAN port) when implementing charging station applications.

5.3.1 Example components

Table5-7 Example components of CANfree communication

assembly	function
Programming equipment PG/PC	Install MagicWorks PLC version 2.28 and above, program and debug M226ES.
Master station, slave station	Two M226ES, one as the master station and one as the slave station.
Expansion board	Two CAN expansion boards, one of which can provide one CAN communication port. The specifications of the expansion board can be found in section 7.6.8 CAN expansion board specifications

5.3.2 network connections



5.3.3 Operation steps

Step 1: Connect all devices

Refer to the above network connection for system wiring:

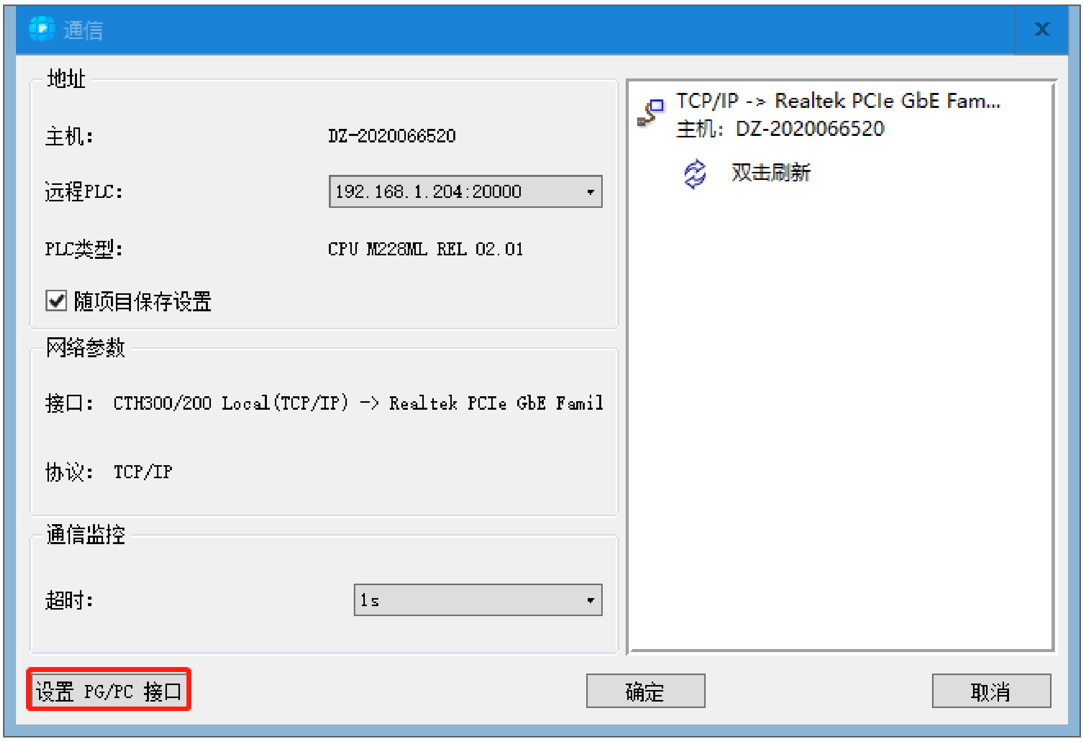
1) Connect the programming device to the CPU M226ES using a standard Ethernet cable.

2) Connect two CAN expansion boards using twisted pair cables.

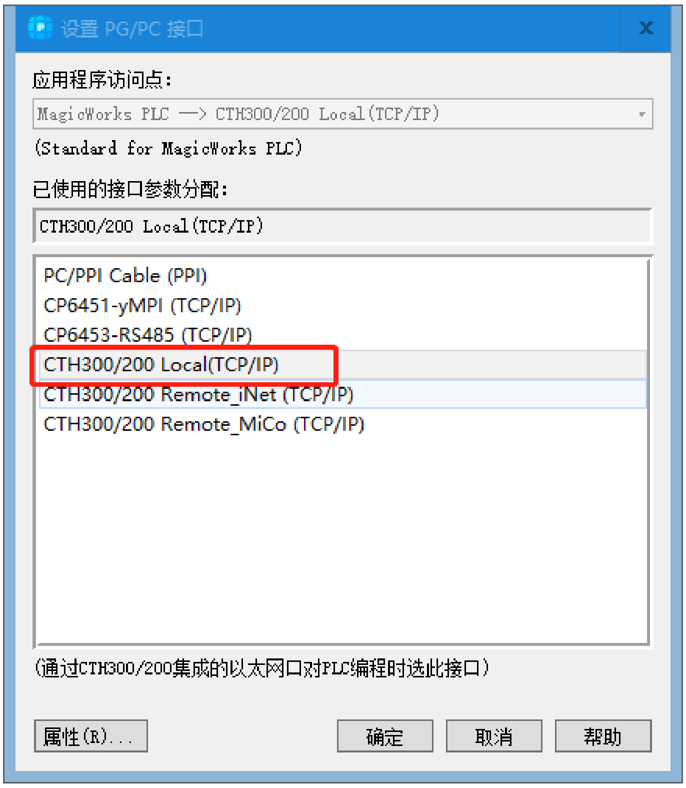
3) Check the wiring and ensure it is correct before connecting the power supply.

Step 2: Communication between the main station and programming devices

1. Open the MagicWorks PLC software, create a new project in MagicWorks PLC, keep the computer and PLC in the same network segment, open the communication interface, and click on Set PG/PC interface.



2. Select 'CTH300/200 Local (TCP/IP)' and then confirm.



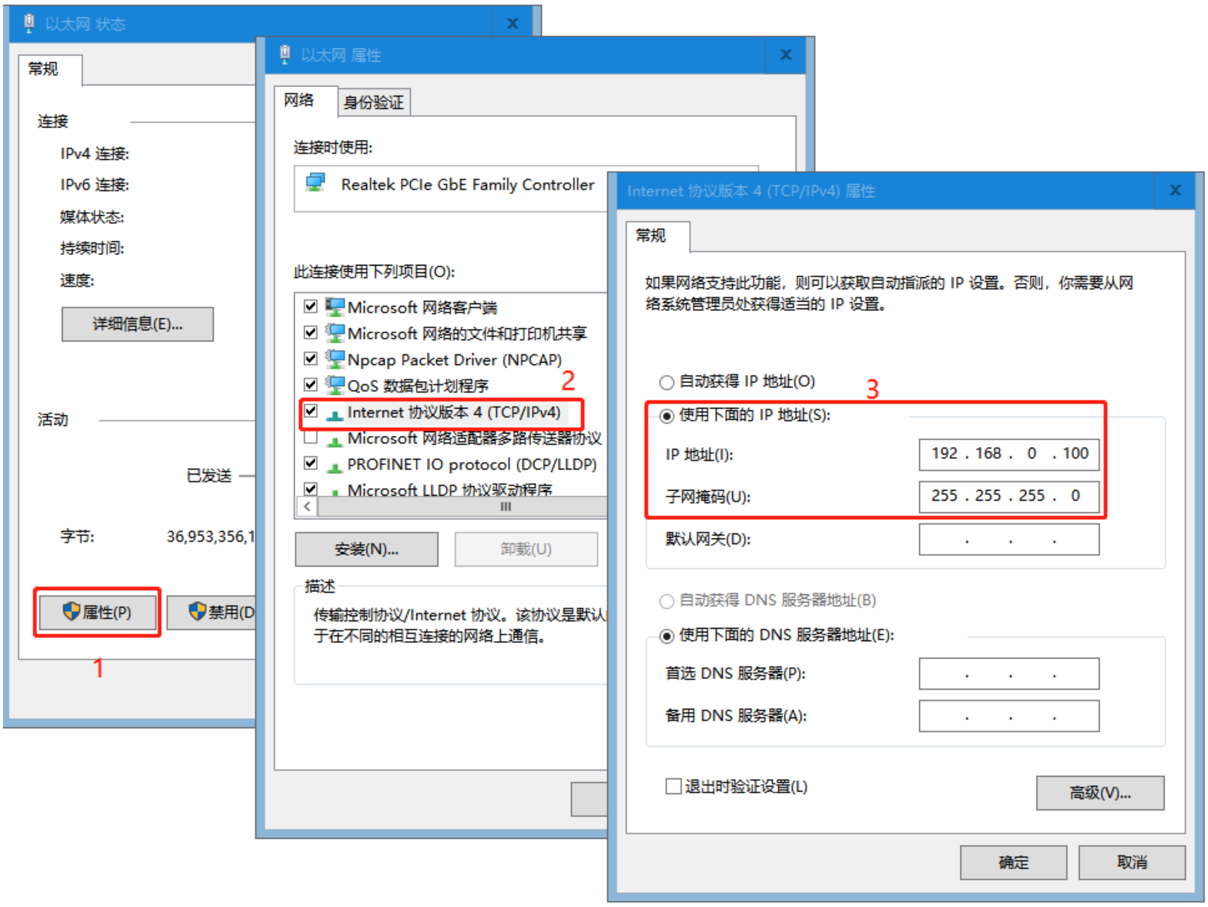
3. Double click to refresh on the communication interface, and M226ES will appear. Click M226ES to confirm.

4. If MagicWorks PLC cannot find M226ES, please first check if the wiring is properly connected, then check the communication parameter settings, and repeat the above steps. After establishing communication with M226ES, set the computer IP to be in the same network segment as M226ES, but the two IPs cannot be the same. The specific process of setting the computer IP is as follows:

1) Turn on the Ethernet properties of the PC,

2) Double click on the Internet TCP/IP protocol,

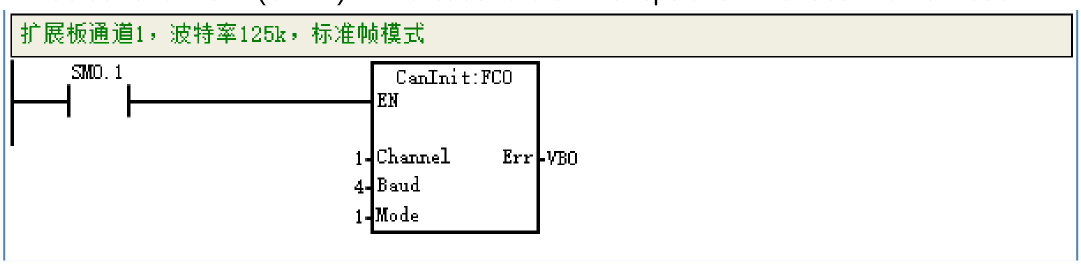
3) Change 'Automatically obtain IP address' to' Use the following IP address', and then fill in '192.168.0. X' in the IP address.



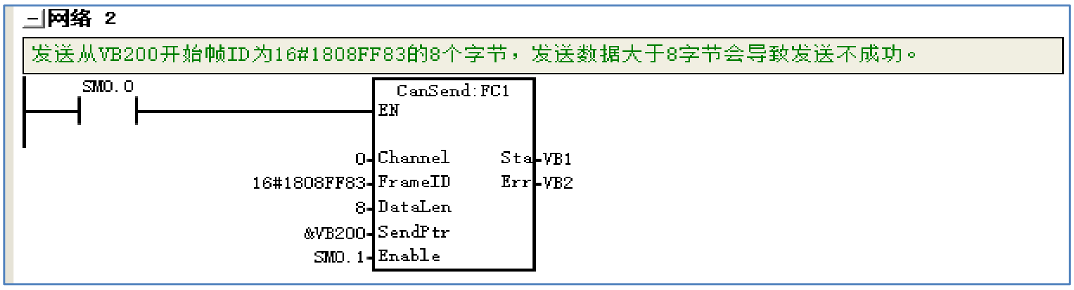
Step 3: Call the CAN communication instruction library canfree_ib (v1.3) for programming

Open the program block and call the CAN communication instruction library for programming.

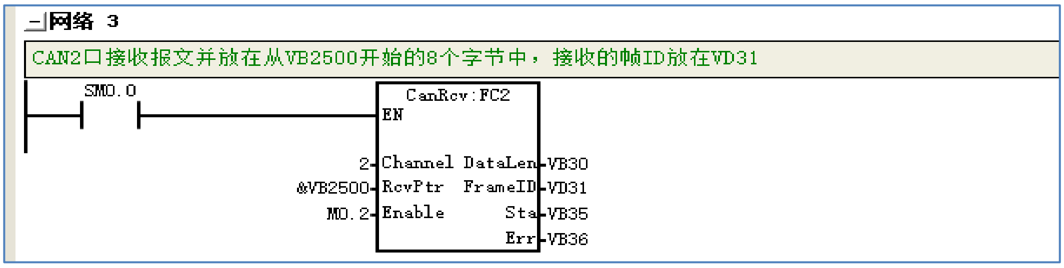
1、Select Channel 1 (CAN1) with a baud rate of 125kbps and Extended Frame Mode 1.



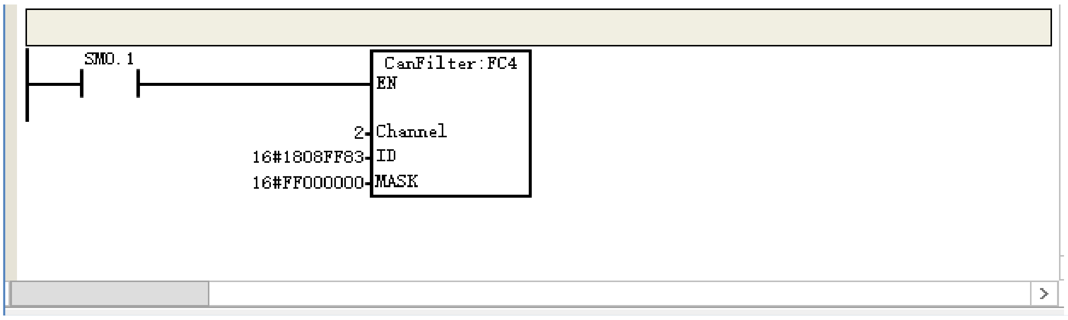
2. The CAN1 port of the main station sends an 8-byte message starting from VB200 (frame ID 16 # 1808FF83). Sending data larger than 8 bytes will result in unsuccessful transmission.



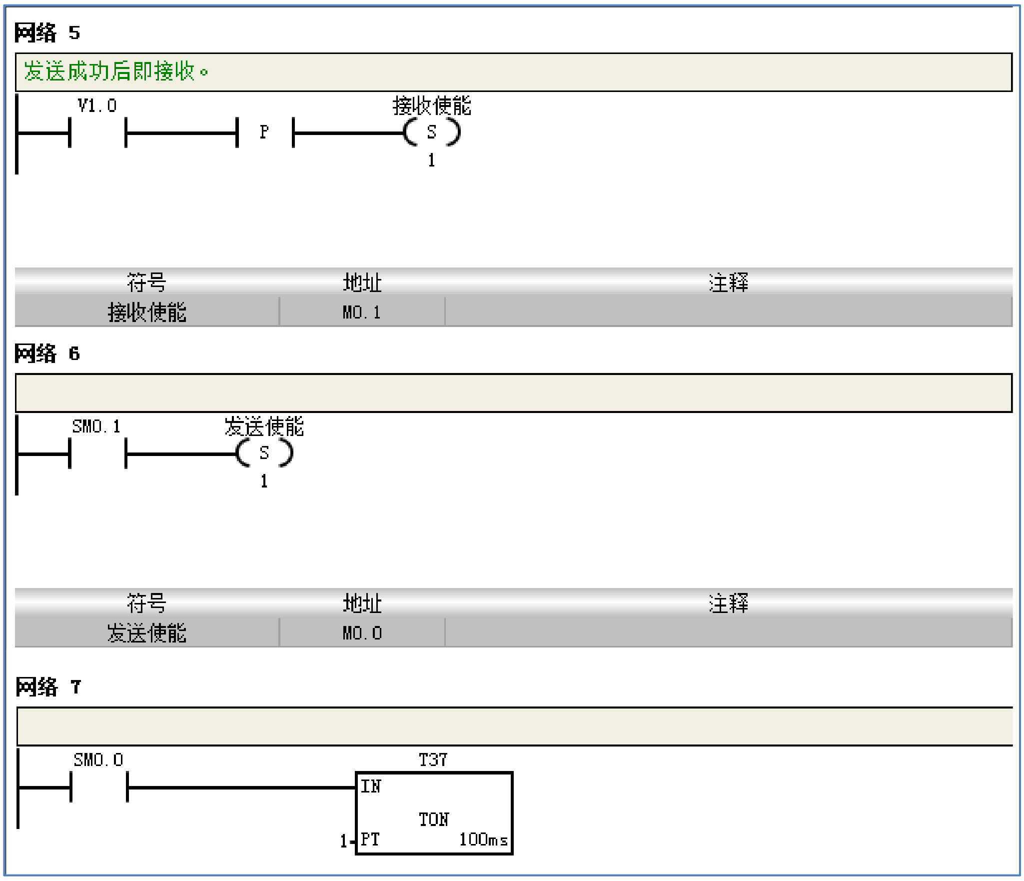
3. Receive messages from the CAN1 port and place them in the 8 bytes starting from VB2500. The received frame ID is placed in VD31.

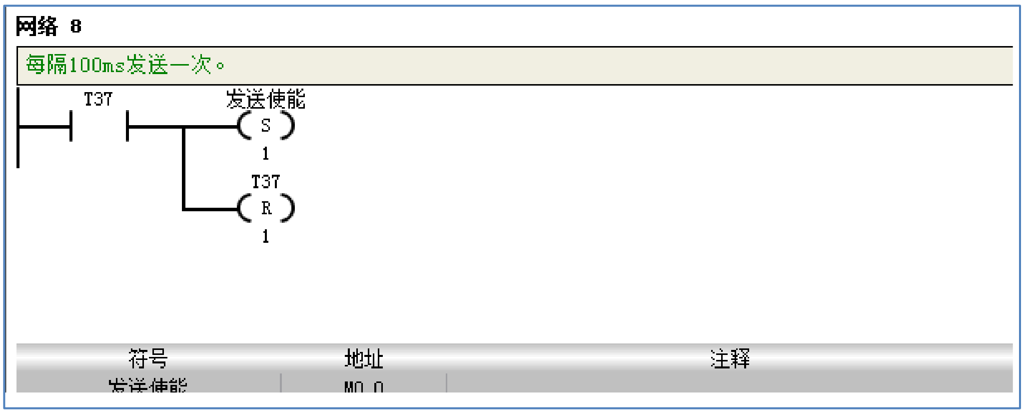


4、Filter the data received by CAN1 port, except for frame ID messages starting with 18, and filter out other frame ID messages.



5. Enable and schedule data transmission and reception operations.



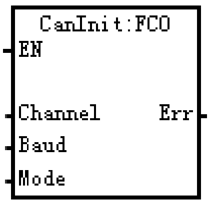


After saving the project, select the menu item "PLC" ->"Compile" ->"Download" to download the program to the PLC.

5.3.4 Detailed explanation of canfree_ib (v1.3) command

CANInt (CAN initialization)

Instruction Name: CANInt



Function: Initialize CAN free port.

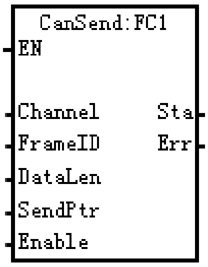
Parameter Description

Parameter Name	Input/output	type	Numerical Range	explain

EN	IN	BOOL	--	--
Channel	IN	BYTE	0-1	CAN channel selection 0: Channel 0 (local) 1: Channel 1 (expansion board)
Baud	IN	BYTE	0-7	Baud rate 0(1MHz)； 1（800kHz)； 2（500kHz)； 3（250kHz)； 4（125kHz)； 5（50kHz)； 6（20kHz)； 7（10kHz)
Mode	IN	BYTE	0-1	frame format 0: CAN_Standard Frame 1: CAN_S Extended Frame
Err	OUT	BYTE	0-4	Error code, please refer to the Err error code definition for details. 0: No error 4: Parameter error

CANSend (CAN Send Data)

Command Name: CANSend



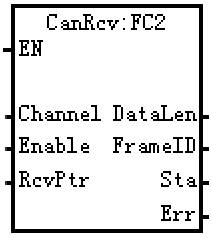
Function: CAN sends data. Enable this command, set Enable to 1 to send once

Parameter Description

Parameter Name	Input/output	type	Numerical Range	explain
EN	IN	BOOL	--	--
Channel	IN	BYTE	0-1	CAN channel selection 0: Channel 0 (local) 1: Channel 1 (expansion board)
FrameID	IN	DWORD	--	Frame ID for sending data
DataLen	IN	BYTE	--	Frame length for sending data
SendPtr	IN	DWORD	--	Send data buffer pointer
Enable	IN	BOOL	--	Enable position
Sta	OUT	BYTE	0-1	Sending status (0-incomplete, 1-completed) Set 1 upon completion of sending, automatically clear 0
Err	OUT	BYTE	0-4	Error code, please refer to Err error code definition for details Local CAN0/CAN1 channel: 0- No errors. 1. Sending timeout. 4- Parameter error CAN expansion board channel: 0- No error, 2- Sending buffer is full

CANRcv (CAN receive data)

Instruction Name: CANRcv



Function: CAN receives data. After receiving the data, it will be filled into the receiving buffer, and the receiving instruction is used to query whether there is data in the receiving buffer. Set Enable once to query the receive buffer.

Parameter Description

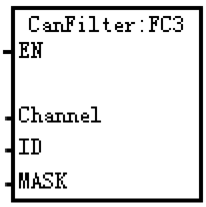
Parameter Name	Input/output	type	Numerical Range	explain
EN	IN	BOOL	--	--
Channel	IN	BYTE	0-1	CAN channel selection 0: Channel 0 (local) 1: Channel 1 (expansion board)
Enable	IN	BOOL		Enable position
RcvPtr	IN	DWORD	--	Receive data buffer pointer
FrameID	OUT	DWORD	--	Frame ID of received data
DataLen	OUT	BYTE	--	Frame length of received data
St	OUT	BYTE	0-1	Receiving status: 0- Receiving incomplete 1- Receiving completed
Err	OUT	BYTE	0-4	Error code, please refer to Err error code definition for details 0: No error. 3: The receiving buffer is empty

CAN data structure

Frame ID	Frame length len<=8	Frame Data
4bytes	1bytes	8bytes

CANFilter (CAN receive data message filtering)

Instruction Name: CANFilter



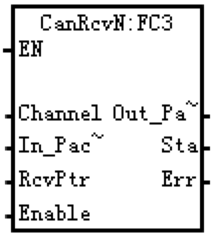
Function: CAN receive data message filtering. If only CAN messages with specific frame IDs are received, filtering instructions can be called. If the program does not call the filtering instruction, the default filtering ID and mask are both 0, which means it receives any CAN message.

Parameter Description

Parameter Name	Input/output	type	Numerical Range	explain
EN	IN	BOOL	--	--
Channel	IN	BYTE	0-1	CAN channel selection 0: Channel 0 (local) 1: Channel 1 (expansion board)
ID	IN	DWORD		The ID of the data message to be received
MASK	IN	DWORD	--	Mask control of ID

CanRcvN (receiving multiple frames of data)

Instruction Name: CanRcvN



Function: Read several frames of CAN data, output the actual number of frames of CAN data, store them in sequence in the buffer memory, offset by one frame with a 13 byte offset

Parameter Description

Parameter Name	Input/output	type	Numerical Range	explain
EN	IN	BOOL	--	--
Channel	IN	BYTE	0-1	CAN channel selection 0: Channel 0 (local) 1: Channel 1 (expansion board)
In_packets	IN	BYTE		Set the frame rate for a single reception
RcvPtr	IN	DWORD		Receive data memory address
Enable	IN_OUT	BOOL	--	Enable position
Out_packets	OUT	BYTE		The actual number of frames received
Sta	OUT	BYTE	0-1	Receiving status: 0- Incomplete. 1- Complete
Err	OUT	BYTE	0-4	Error code, please refer to Err error code definition for details

Err Error Code Definition

Incorrect coding	definition
0	No errors
one	Send Timeout
two	Buffer full (sent)
three	Buffer is empty (receive)
four	Parameter error
five	CANopen protocol reuse conflict

5.4 EtherCAT communication

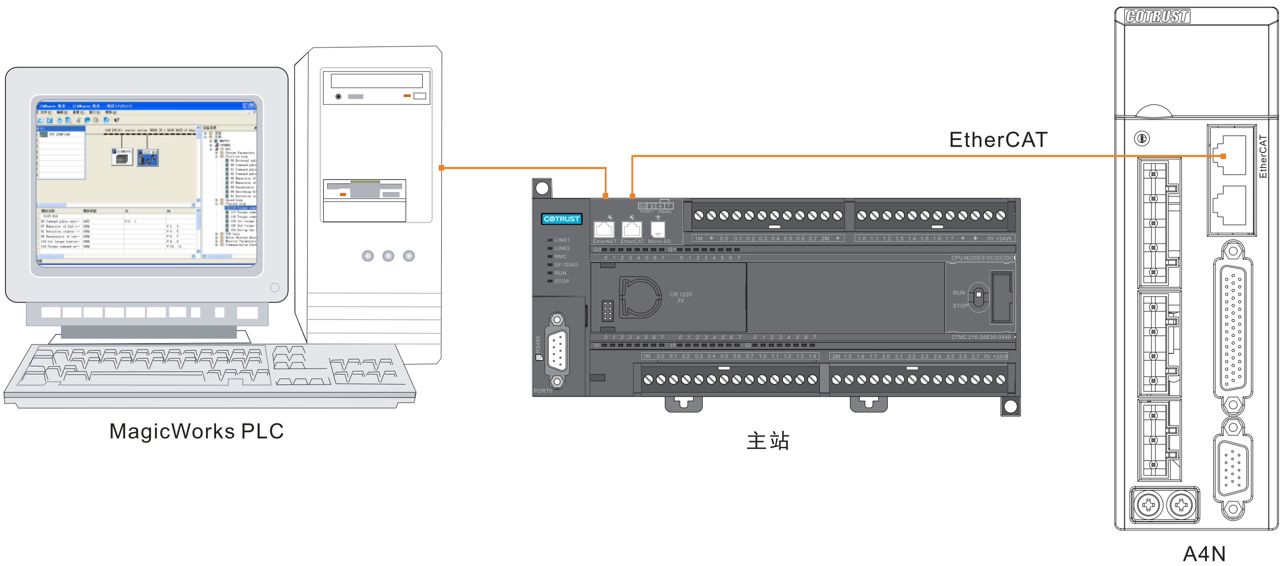
This section will demonstrate the EtherCAT communication function of M226ES through a specific example.

5.4.1 Example components

表5-1 Example components of EtherCAT communication

assembly	function
Programming equipment PG/PC	Install MagicWorks PLC software (V2.29 or higher) to configure, program, and debug M226ES.
M226ES main control module	The CPU executes user programs and communicates with other nodes in the Ethernet network through Ethernet interfaces.
EtherCAT Slave Device	A4N servo drive.
A4N servo motor	Connected to A4N servo drive.
2 standard network cables	Connect M226ES to programming devices Connect M226ES to A4N servo drive
Encoder cable	Connect A4N servo driver and motor

5.4.2 network connections



5.4.3 Operation steps

Step 1: Wiring

1) Connect PC to M226ES using standard Ethernet cable

2) Connect M226ES to A4N servo drive using standard Ethernet cable

3) Connect A4N servo drive and motor using encoder cable

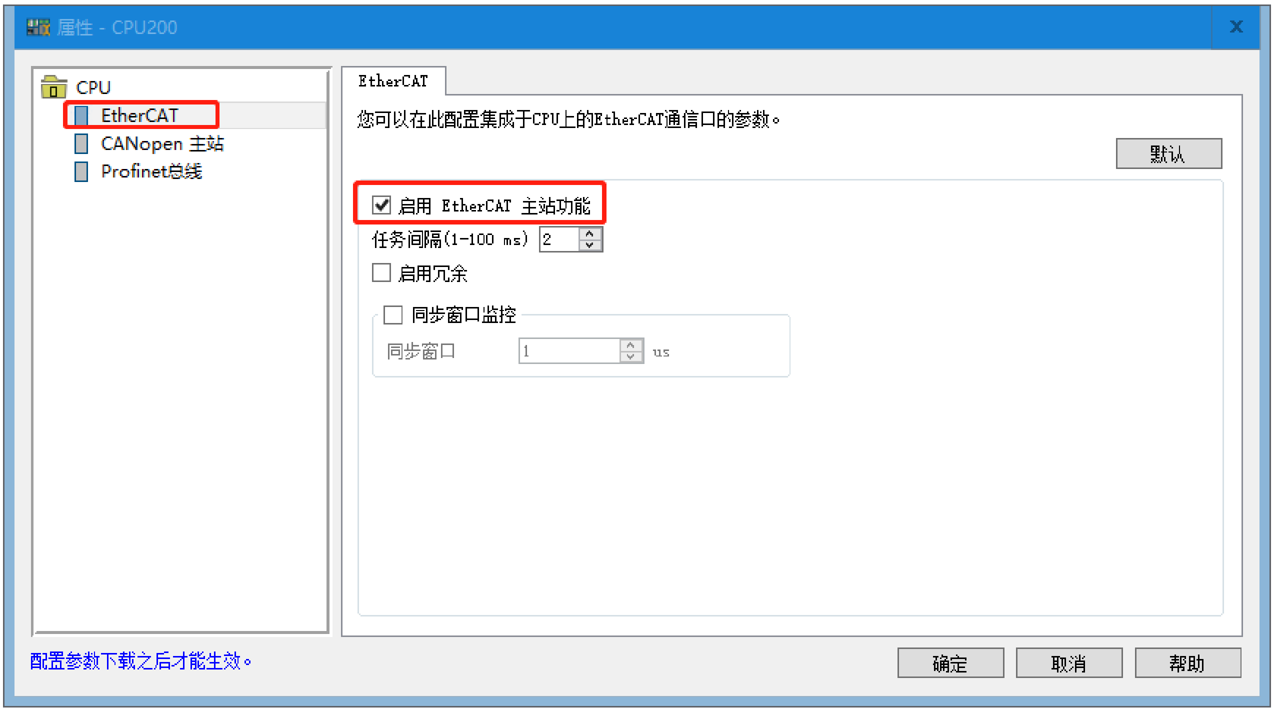
Step 2: Set up communication

Create a new project in MagicWorks PLC, add M226ES station to the project, and then refer to section 4.2 to establish communication

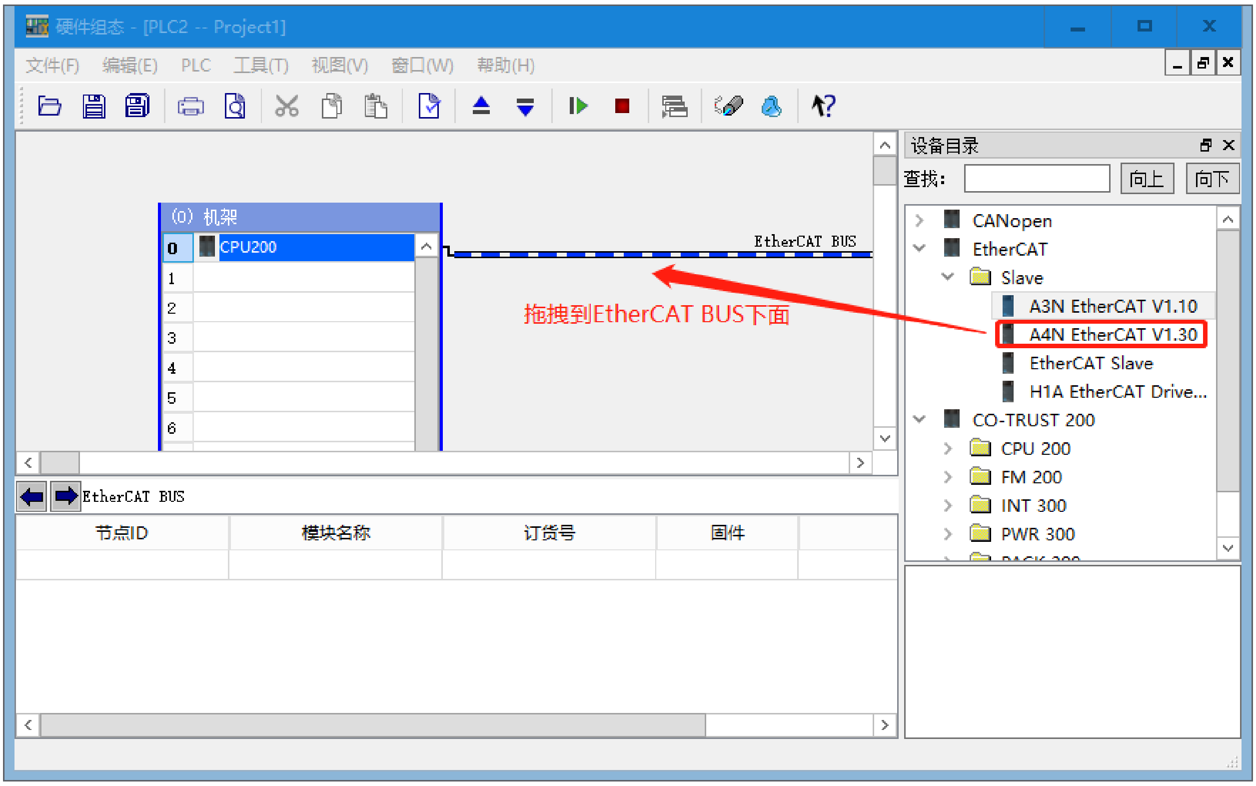
Step 3: Perform hardware configuration in MagicWorks PLC

Click to select the M226ES site in the MagicWorks PLC project view, and then double-click the hardware configuration icon in its right-hand work window to enter the hardware configuration interface.

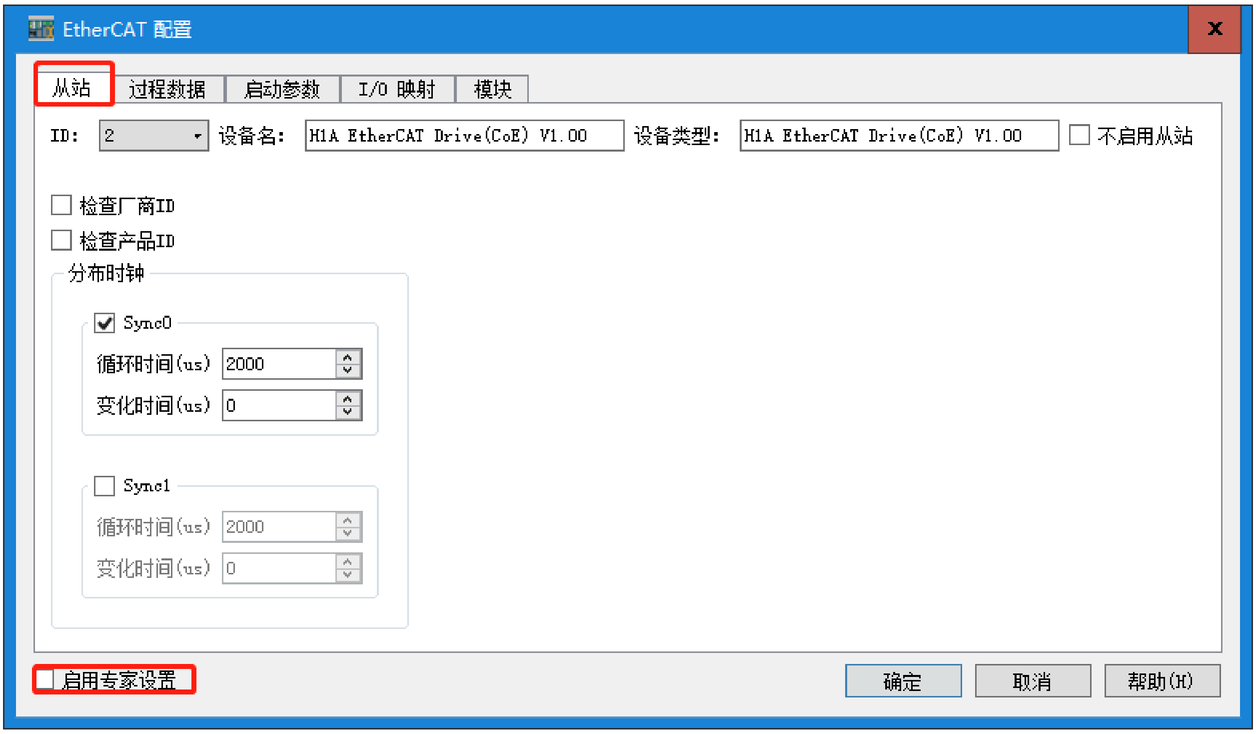
1) In the hardware configuration interface, expand the device directory to add the CTH200 CPU to the rack, then double-click the inserted CP200 to open its properties dialog box, select the "EtherCAT" tab in the left column, and check "Enable EtherCAT Master Station Function" to enable the EtherCAT function.



Expand the EtherCAT node in the device directory tree, open the slave node, select the slave device model A4N that matches your current actual device, drag and drop it into the EtherCAT bus area of the configuration interface, and the slave will be successfully added.

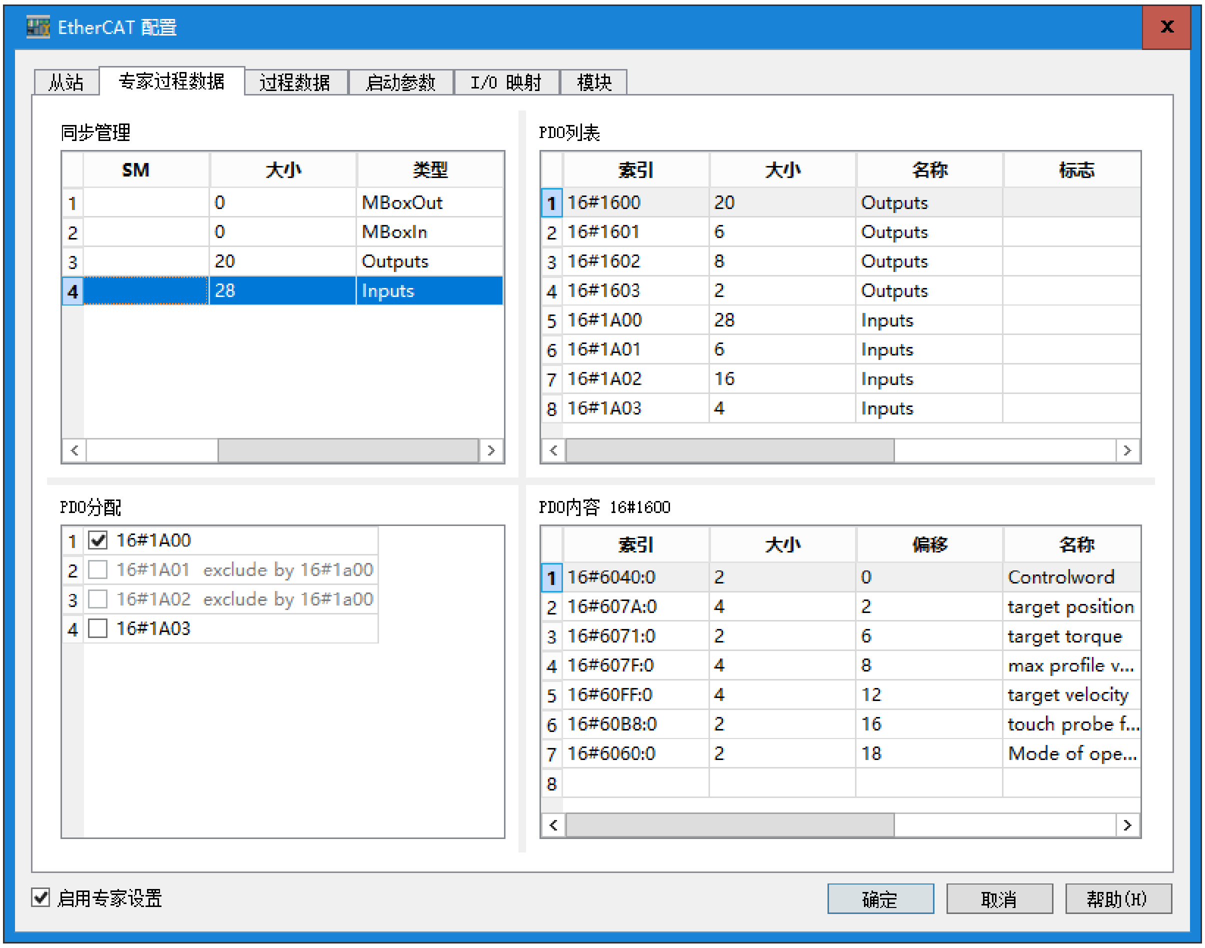


3) Double click the A4N slave icon on the EtherCAT bus to bring up the EtherCAT configuration dialog box as shown in the following figure. For specific parameter configuration, please refer to the following description. Select 'Enable Expert Settings' to configure expert process data.

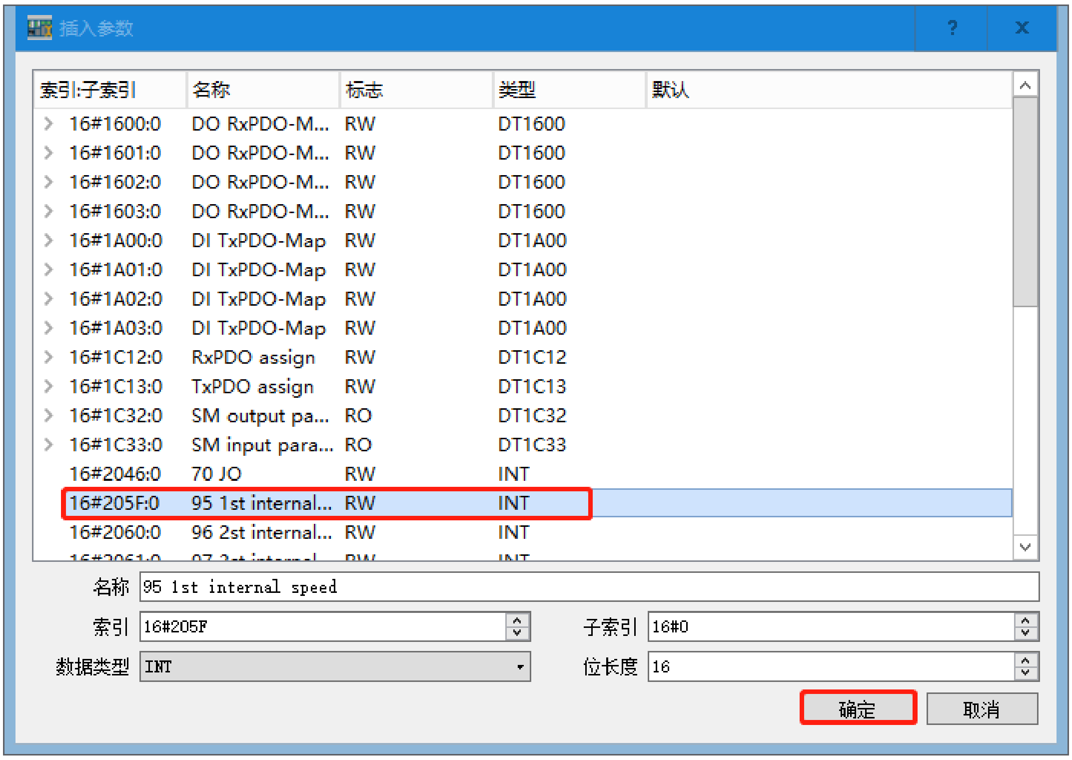


◆ Expert Process Data tab

Select input/output through synchronous management, then check the required input/output group in PDO allocation, select the corresponding group in PDO list, and finally add or remove parameters for the group in PDO content.

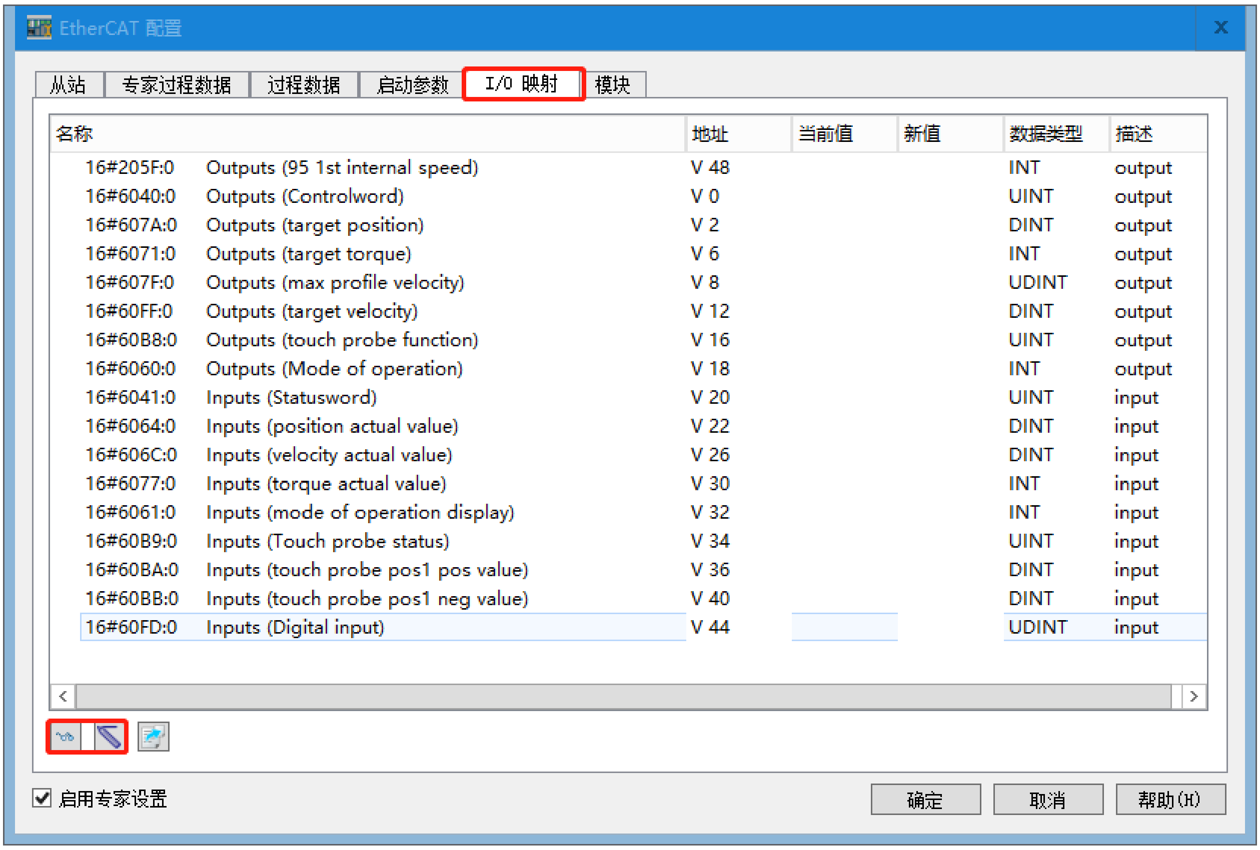


To add variables to a specific PDO parameter, select the parameter, right-click in the PDO Content area and select "Insert". The settings window for adding variables will then appear. Select the variable you want to add and click "OK", as shown in the following figure:



◆ I/O Mapping tab

The successfully configured I/O parameters will be displayed in the I/O mapping, and double-click the address column to edit their address.

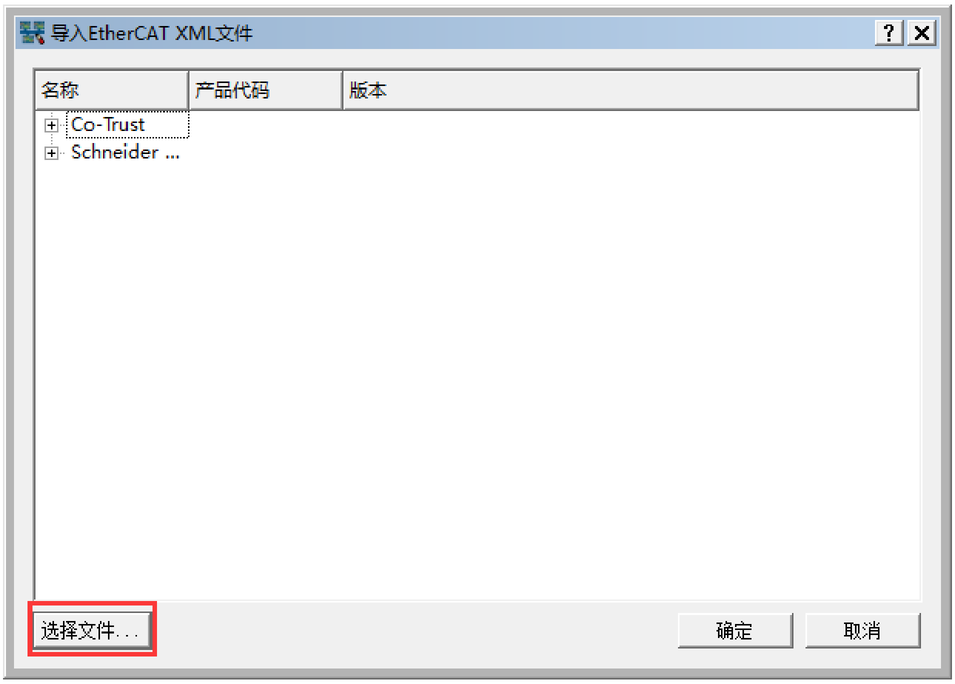


Click the monitoring button in the "I/O Mapping" tab of the EtherCAT configuration dialog box to start monitoring the configured parameters. Then enter the parameter values in the "New Value" column, and finally click the write button to successfully write the new values.

Step 4: Import third-party slaves

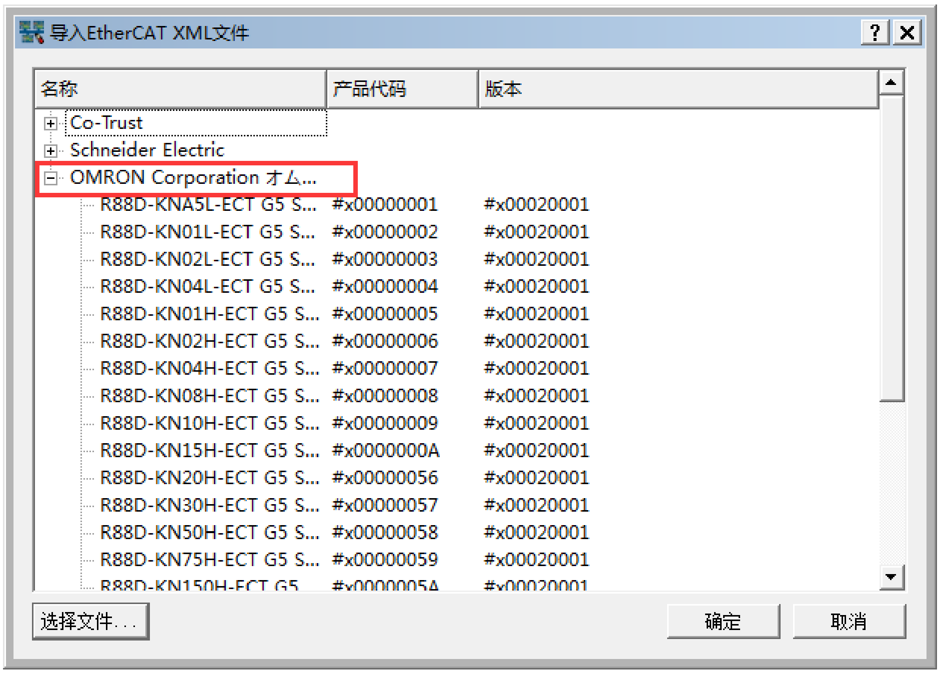
If a third-party EtherCAT slave is selected, the corresponding XML device description file needs to be imported first. Taking the Omron R88D-KN series servo as an example, please follow the steps below to import its XML file:

- Open "Hardware Configuration" ->"Tools" ->"Import EtherCAT XML File (X)" under Configuration Engineering, and the following interface will be displayed:



◆ Click on 'Select File' in the opened window and select the XML file in the corresponding storage path. The following window will pop up:

◆ After clicking "OK", the added XML file and its manufacturer will be listed in the previous interface:



◆ After clicking "OK", the relevant servo will be included in the EtherCAT slave list, and the window will display the manufacturer and related information of the product. Click "Finish" to end the import operation, and then users can configure and call third-party slave stations according to their actual needs.

	Tips The status of EtherCAT can be checked by viewing the EtherCAT registers SMB2390~SMB2417.

5.5 Socket communication

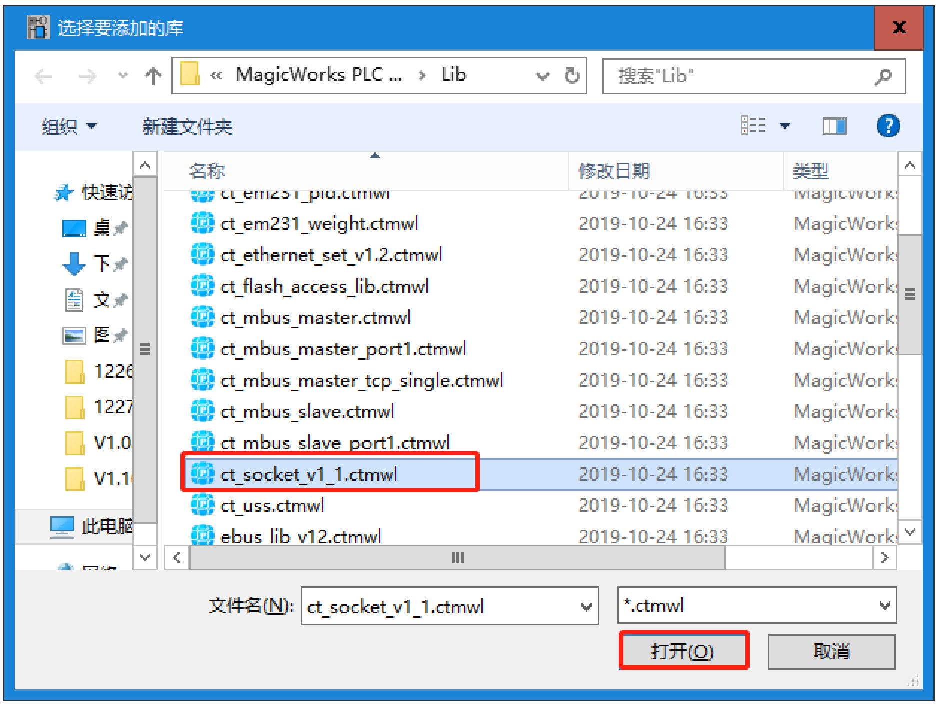
This section introduces socket communication through a specific example. The specific steps are as follows:

Step 1: Add Library Files

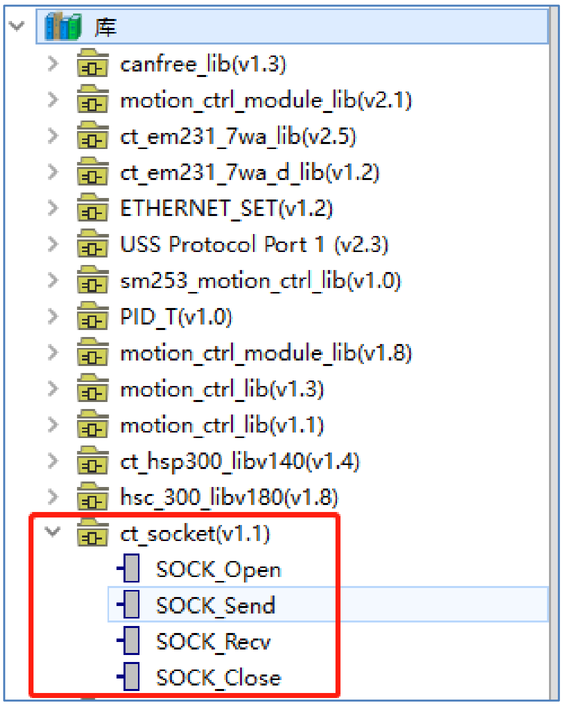
Select "File" - "Add/Remove Library" in the project manager interface or programming interface, and click

"Add" in the pop-up window to browse to the storage location of the library file:

Find the library file "ct_stocket w_1_1. ctmwl" as shown in the following figure, click the "Open" button, and then click "OK".



After successful installation, you can see the newly added "ct_stocket (v1.1)" under the "Libraries" section of the program block interface directory tree:



Step 2: Program the CPU

1) Create a TCP connection using the SOCK_Open instruction

Click on the Magicworks PLC program block and call SOCK_Open from the socket library to program.

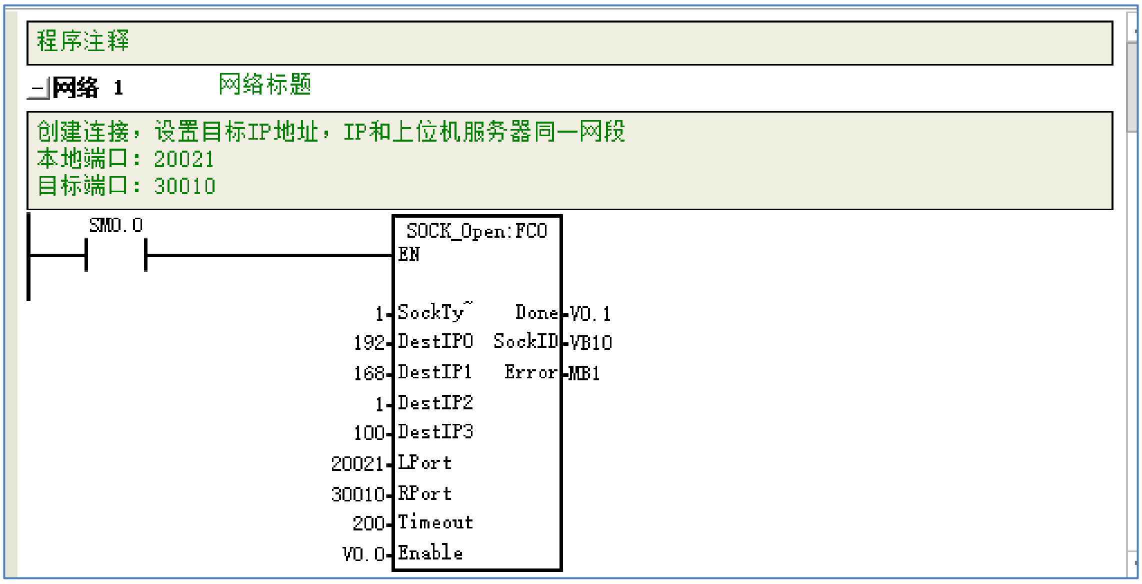
DestIP0: The first segment of the IP address of the corresponding upper computer server (such as 192)

DestIP1: The second segment of the IP of the corresponding upper computer server (such as 168)

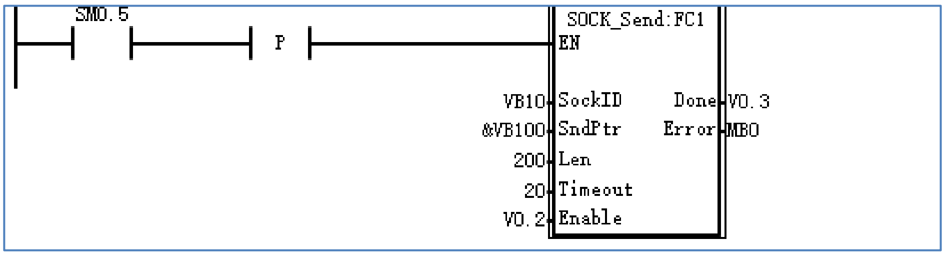
DestIP2: The third segment of the IP address of the corresponding upper computer server (such as 1)

DestIP3: The fourth segment of the IP address of the corresponding upper computer server (such as 100)

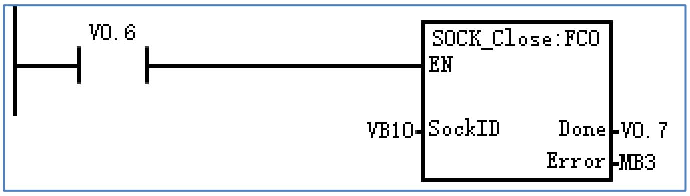
LPort: Local CPU port number
RPort: Port number of the upper computer



2) Use send/receive instructions to set the send/receive cache interval.



3) Stop connection command, which needs to be executed once when disconnecting and reconnecting.



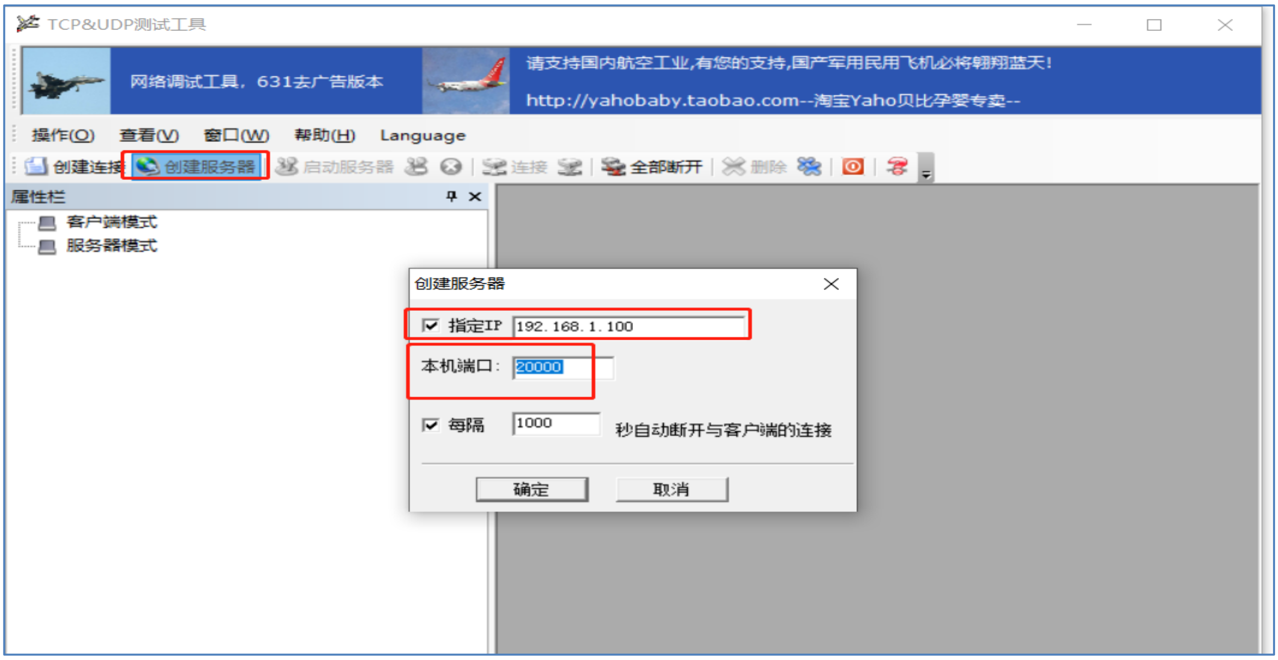
The specific use of instructions can be learned in detail by downloading the Magicworks PLC Instruction Library Manual from the official website of COTRUST.

Step 4: Use of PLC and testing tools

1. Click on "Create Server" and select the specified IP address.

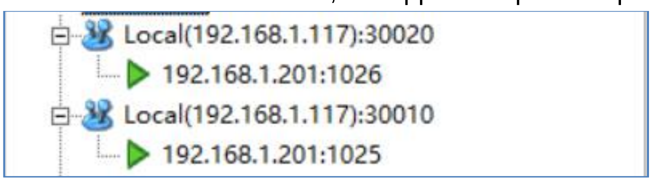
Specify IP: corresponds to the IP set in the SOCK_Open command

Local port: corresponding to the RPort set in the SOCK_Open command

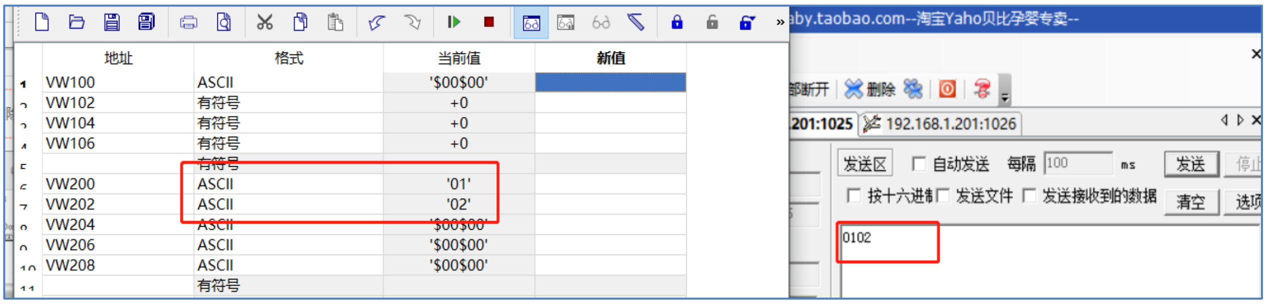


2. Click - Start Server

After successful connection, the upper computer displays a green triangle



3. Monitor and send data in ASCLL format for viewing



Attention to socket usage

When using testing tools, the computer IP must be consistent with the server IP.

The PLC network segment must be the same as the computer network segment

3. The port of the server cannot be 502, it can be 20000 like the local port, and must be greater than 0, ranging from 1 to 65535

4. The server can be connected to an external network. Only IPV4 is supported. In fact, IPV4 is the fourth version of the Internet communication protocol, not only applicable to the LAN, but also applicable to the WAN. Can connect to any IPV4 device on the public network that supports SOCKET.

5. A server must exist before opening SOCK - open.

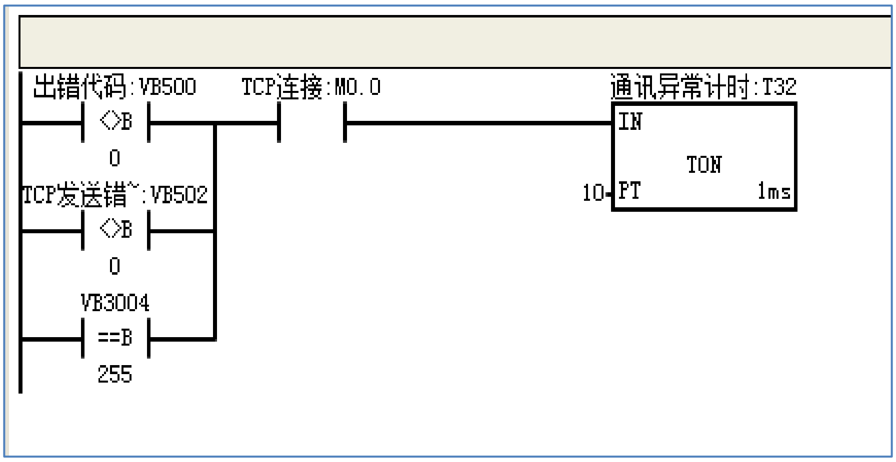
6. Opening the testing software will prompt a firewall asking for network approval, otherwise the server setup will fail.

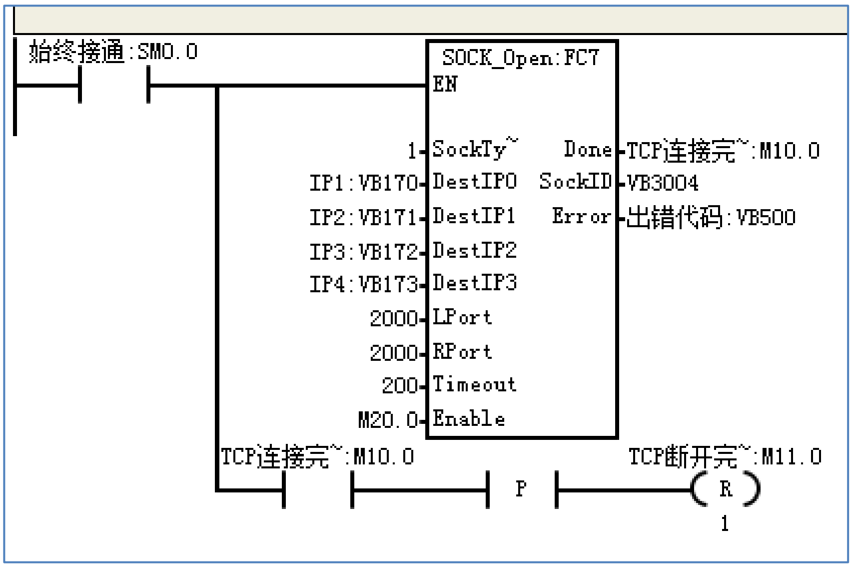
7. PLC only supports international code GB2312-80 for sending and receiving

8. The difference between UDP and TCP clients: UDP only handles sending, has small data, and is fast

Client can be 1-on-1, server 1-to-many

9. In case of communication interruption caused by interference or other reasons during the communication process, attention should be paid to automatic reconnection of the server. The reconnection criteria (error code not equal to 0, connection number equal to 255) are shown in the following figure. After communication interruption, reconnection needs to be delayed by about 3 seconds. If the time is too short, it may also cause the connection to fail.





5.6 Example of Siemens S7 protocol master-slave communication

M226ES supports Siemens S7 protocol master-slave station function, with Siemens PLC or screen or upper computer software supporting S7 protocol as the master station, and M226ES as the slave station, for S7 protocol communication, supporting up to 8 connections (regardless of master-slave).

This section will demonstrate the parameter configuration operation of M226ES with Siemens touch screen PLC and third-party support for Siemens S7 protocol touch screen and upper computer communication through specific examples.

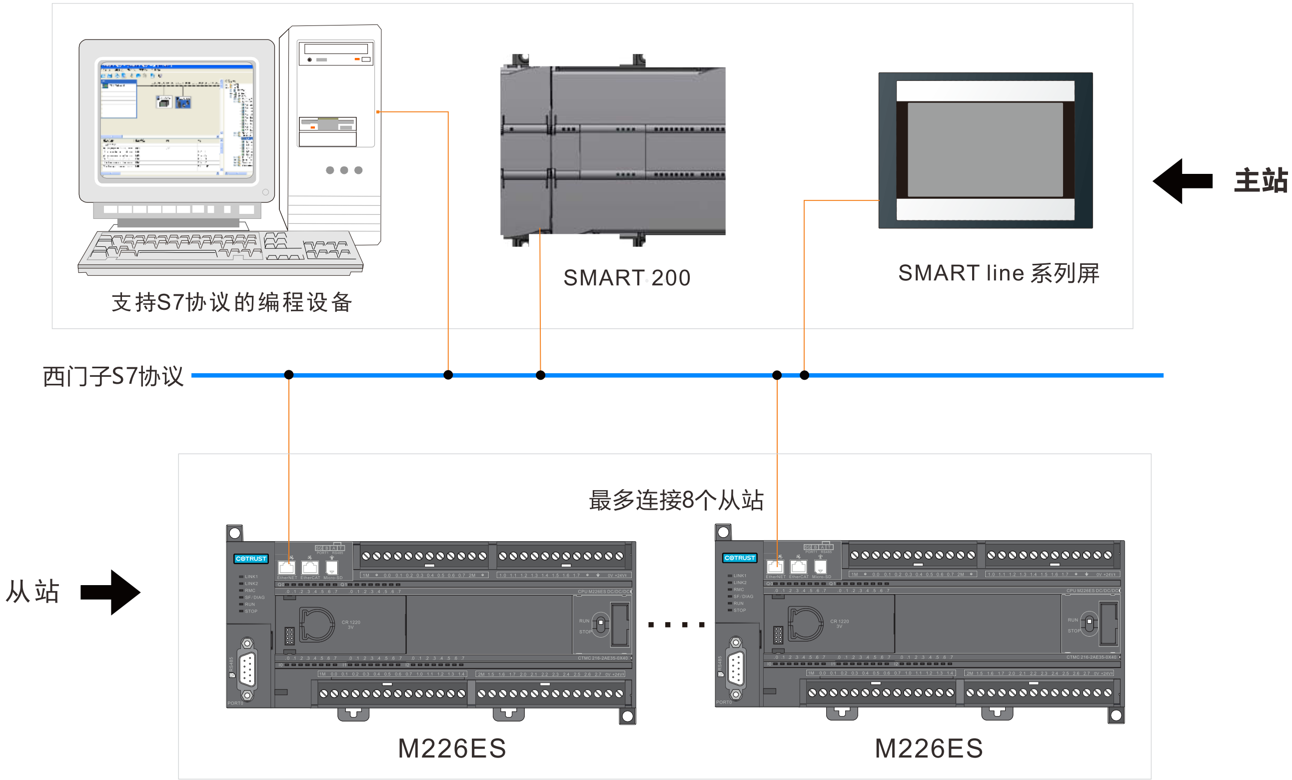
5.6.1 Example components

表5-1 Example components

assembly	function
Programming equipment	Install STEP 7-MicroWin SMART software to configure, program, and debug CTMC series CPUs.
CTMC series CPU	As a S7 protocol slave station, communicate with the S7 protocol master station
Siemens SMART200 PLC	As the S7 protocol master station, communicate with S7 protocol slave stations
Configuration King	As the S7 protocol master station, communicate with S7 protocol slave stations
Kunlun State Touch Screen	As the S7 protocol master station, communicate with S7 protocol slave stations
Siemens Touch Screen	SMART700IEV3, as the S7 protocol master station, communicates with S7 protocol slave stations
Weilun Tong touch screen	As the S7 protocol master station, communicate with S7 protocol slave stations

5.6.2 Network Architecture

The following figure shows a typical connection of CTMC series CPU with Siemens or other equipment or upper computer software for S7 protocol communication:



5.6.3 Network port communication between Siemens products and CTMC series CPUs

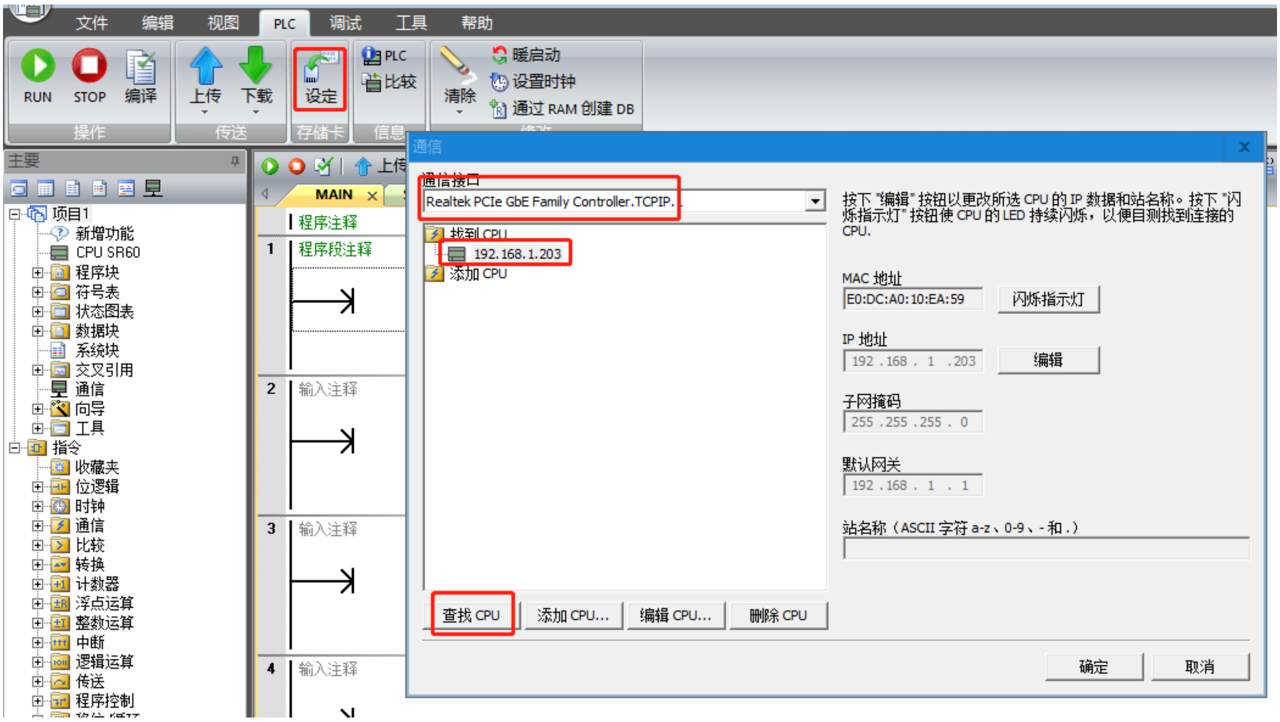
Example 1: S7 protocol communication between SMART200 and CTMC series CPU (CTMC series CPU acts as a slave)

This example uses Siemens SMART200 as the master station and CTMC series CPU as the slave station to perform S7 protocol communication.

<Note>: Siemens touch screens can also serve as the main station for S7 protocol communication with CTMC series CPUs.

1. Communication settings between SMART200 PLC and STEP 7-MicroWin SMART programming software

Open STEP 7-MicroWin SMART software, connect the SMART 200 PLC to the computer using RJ45 Ethernet cable (keep the computer and PLC in the same network segment), open the SMART software settings interface, select the local network card corresponding to the computer, and then click Find CPU to find the PLC.

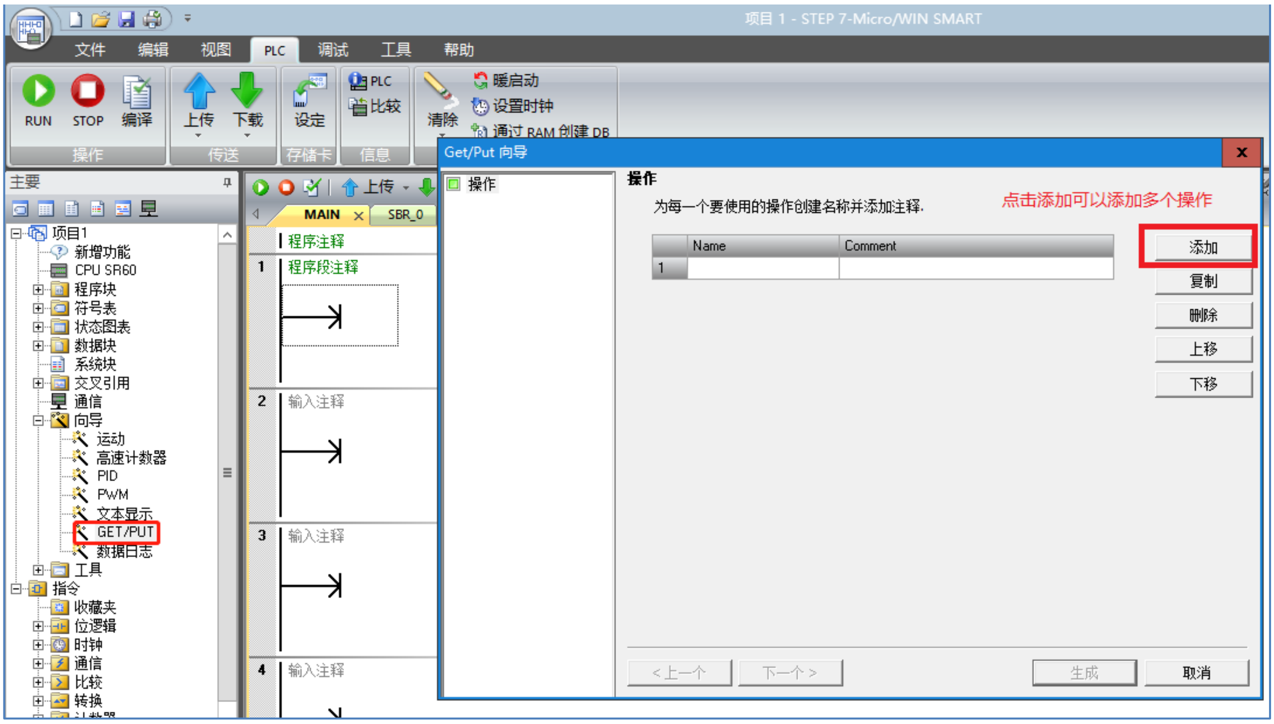


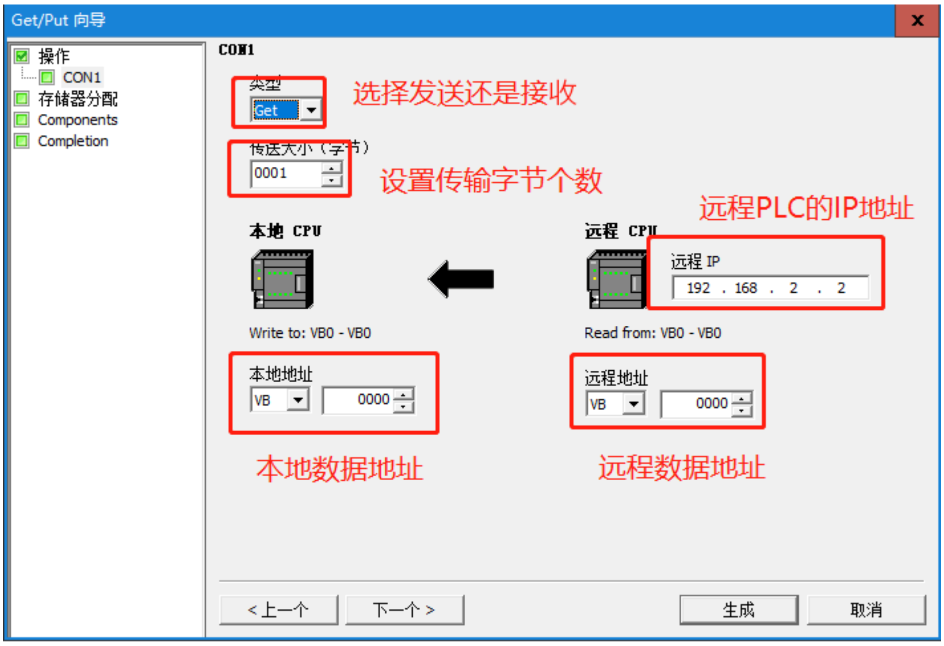
2. SMART200 PLC as the master station, CTMC series CPU as the slave station program setting

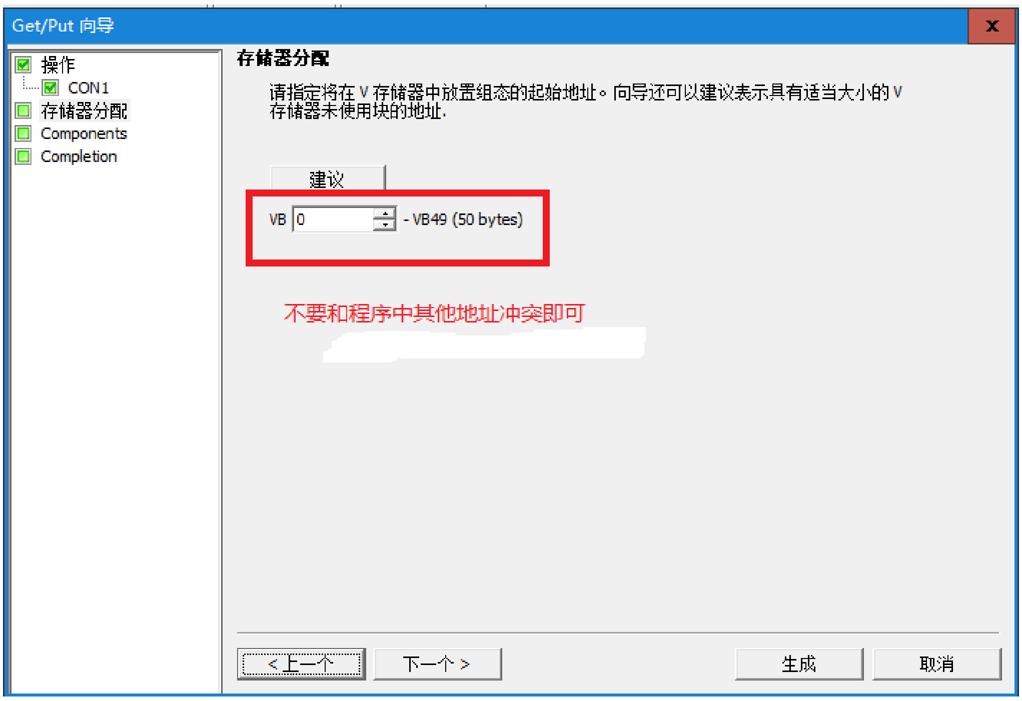
Main station configuration

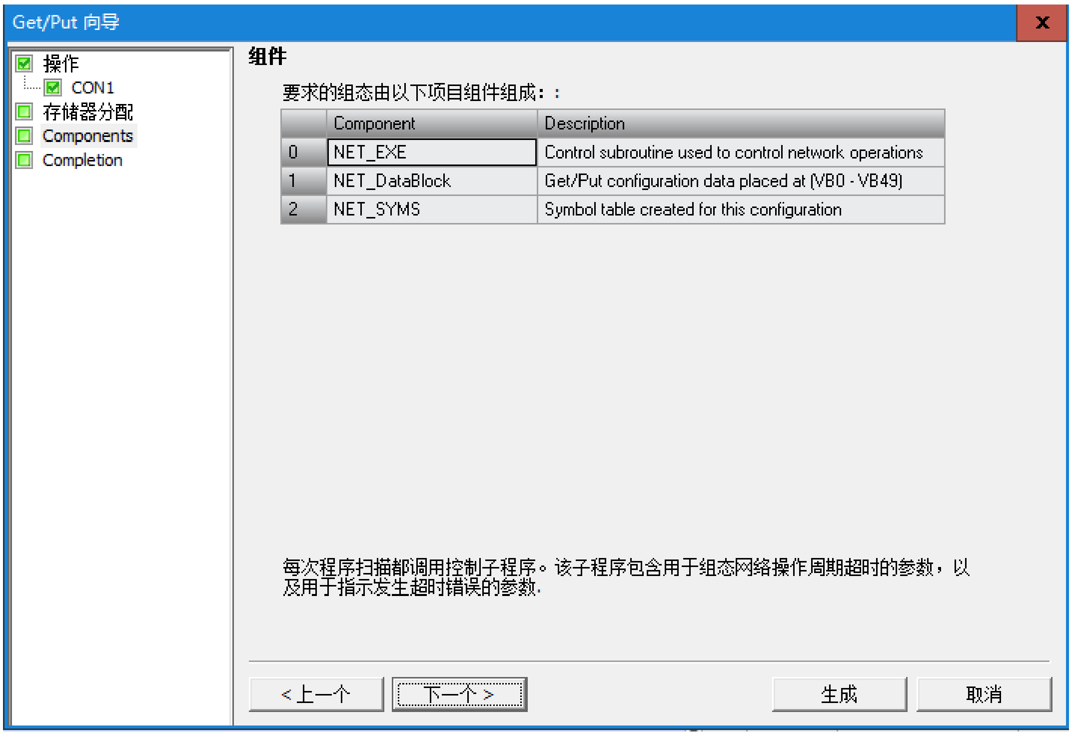
1）Establish GET/PUT Wizard

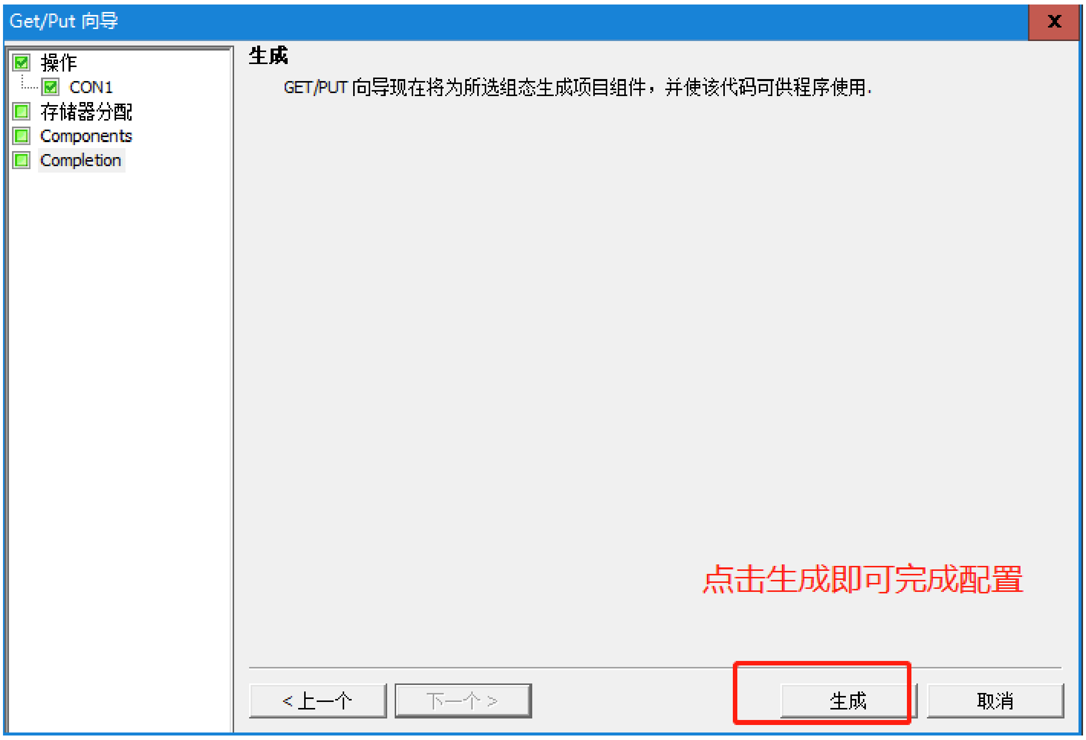
Open the GET/PUT wizard in STEP 7-MicroWin SMART software and make the following settings:



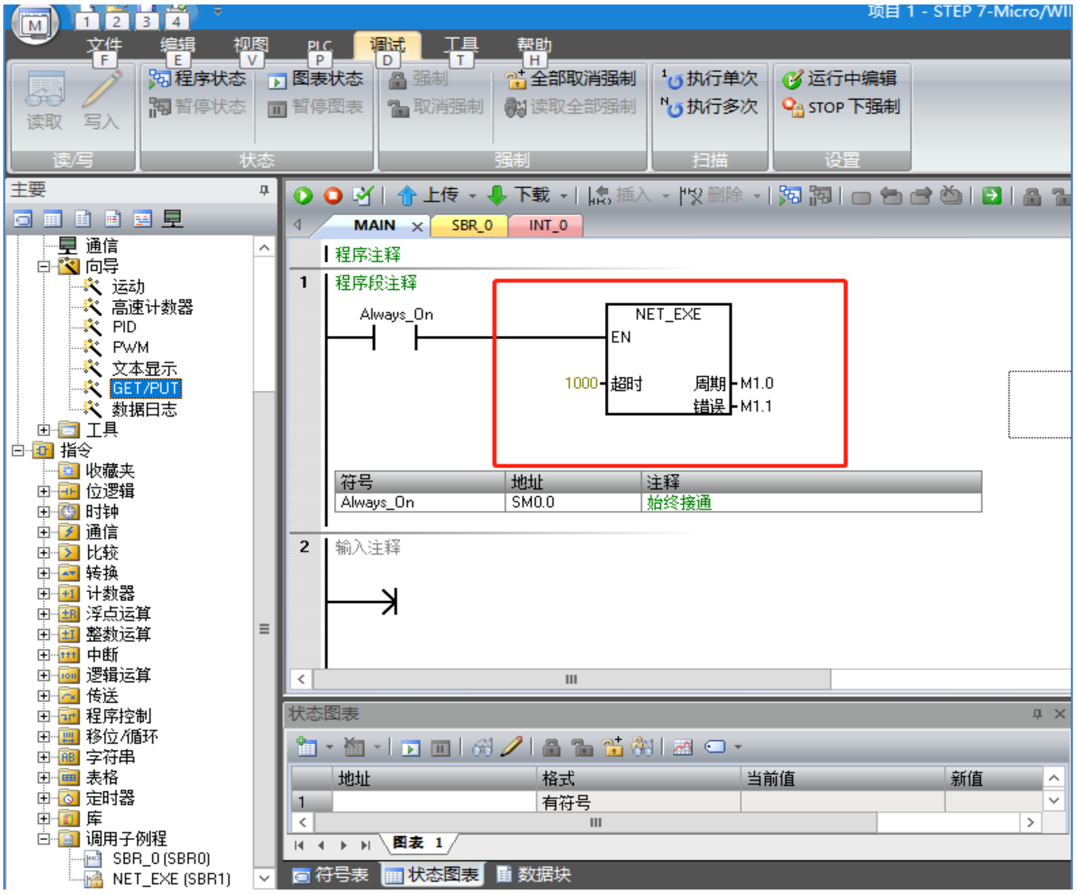

Get/Put 向导
操作
CON1
存储器分配
Components
Completion
CON1
类型
Get
选择发送还是接收
传送大小（字节）
0001
设置传输字节个数
远程PLC的IP地址
本地 CPU
远程 CPU
远程 IP
192 . 168 . 2 . 2
Write to: VB0 - VB0
Read from: VB0 - VB0
本地地址
VB 0000
远程地址
VB 0000
本地数据地址
远程数据地址
< 上一个
下一个 >
生成
取消

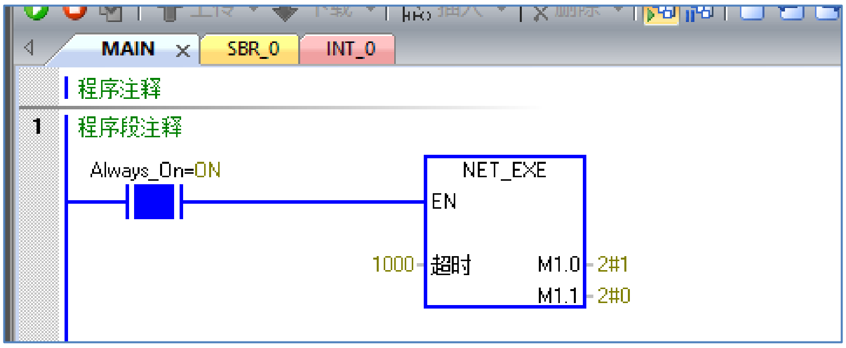

Get/Put 向导
操作
CON1
存储器分配
Components
Completion
存储器分配
请指定将在 V 存储器中放置组态的起始地址。向导还可以建议表示具有适当大小的 V 存储器未使用块的地址.
建议
VB 0 - VB49 (50 bytes)
不要和程序中其他地址冲突即可
< 上一个
下一个 >
生成
取消





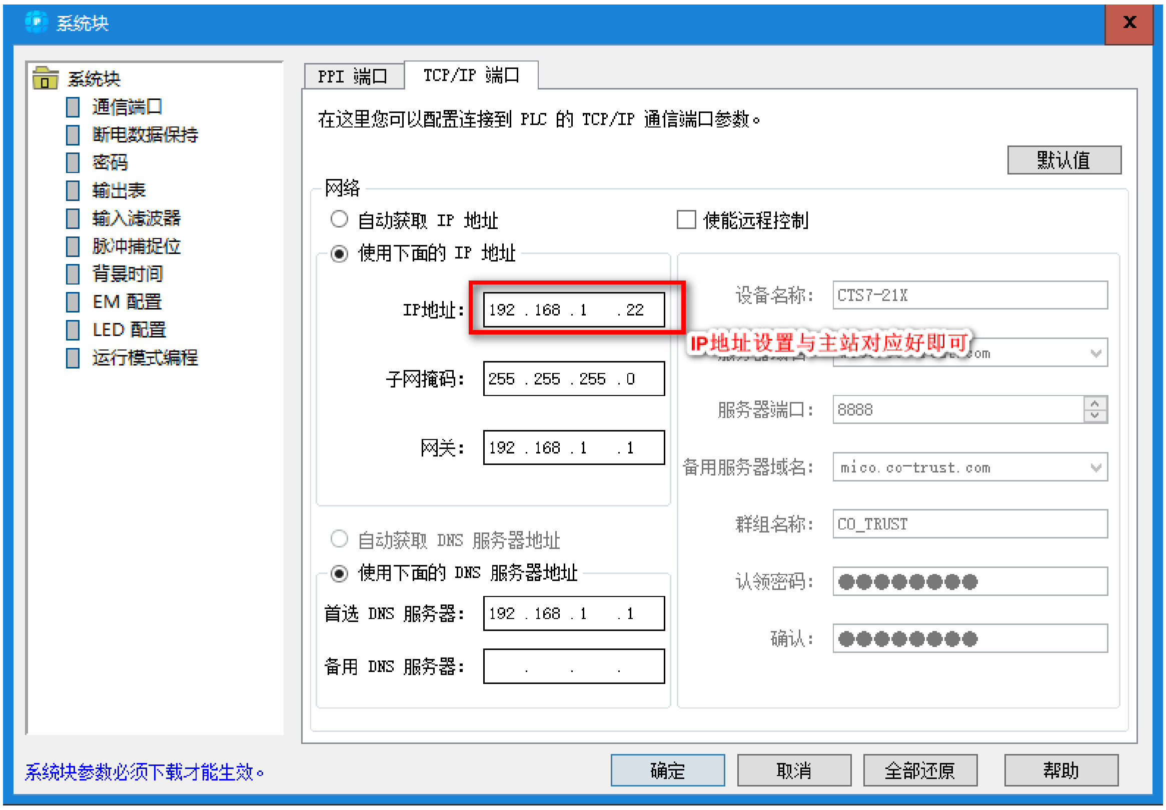
2) Use sm0.0 to call the wizard generated block in the SMART software program





Slave configuration

Simply configure the IP address in the system block of the CTMC series CPU, as shown below:



After setting up the S7 protocol master and slave stations through the above steps, S7 protocol communication between Siemens SMART200 and CTMC series CPUs can be achieved.

Example 2: Communication between CTMC series CPU and SMART200 using S7 protocol (CTMC series CPU as the main station)

The CTMC series CPU supports the Siemens S7 protocol, and the CTMC series CPU M226ES can directly serve as the master station. This section will demonstrate the parameter configuration operation for communication between the master station M226ES PLC and the slave station Siemens SMART200 PLC through specific examples.

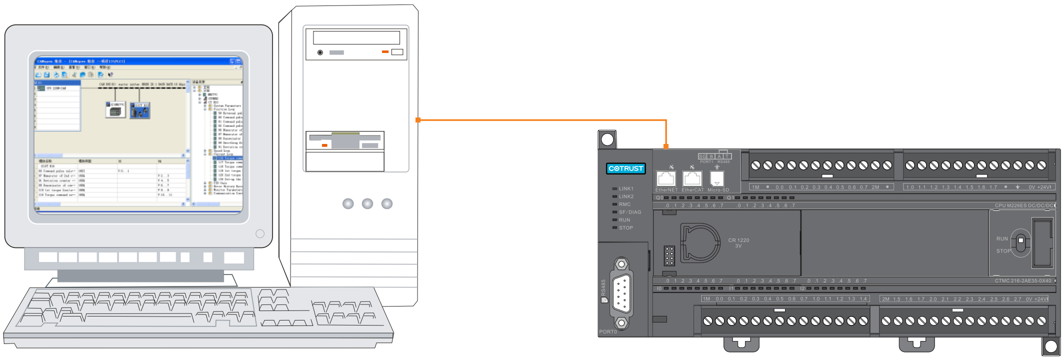
1. Example components

表5-1 Example components

assembly	function
Programming equipment	Magicworks PLC software is installed to configure, program, and debug the CTMC series CPU M226ES.
M226ES CPU	As the S7 protocol master station, communicate with S7 protocol slave stations
Siemens SMART200 PLC	As a S7 protocol slave station, communicate with the S7 protocol master station
Standard Ethernet cable	·Connect M226ES PLC with programming equipment ·Connect M226ES PLC (S7 master station) to Siemens SMART200 PLC (S7 slave station)

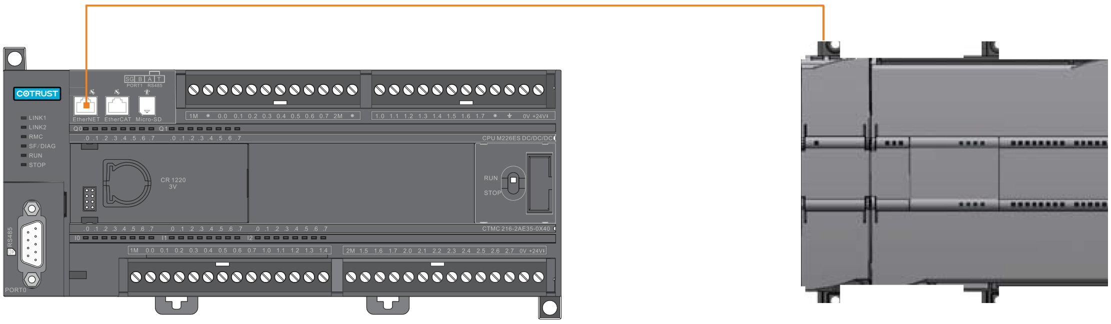
2. Network connection

Connect the programming device to the M226ES PLC using a standard Ethernet cable, and program the S7 main station (M226ES PLC) through the programming device:



MagicWorks PLC　　　　S7主站

Connect the S7 master station and S7 slave station using a standard Ethernet cable. The S7 master station will perform a write operation to write data from the specified address to the S7 slave station, and then read the data from the corresponding address through the S7 slave station to achieve S7 communication

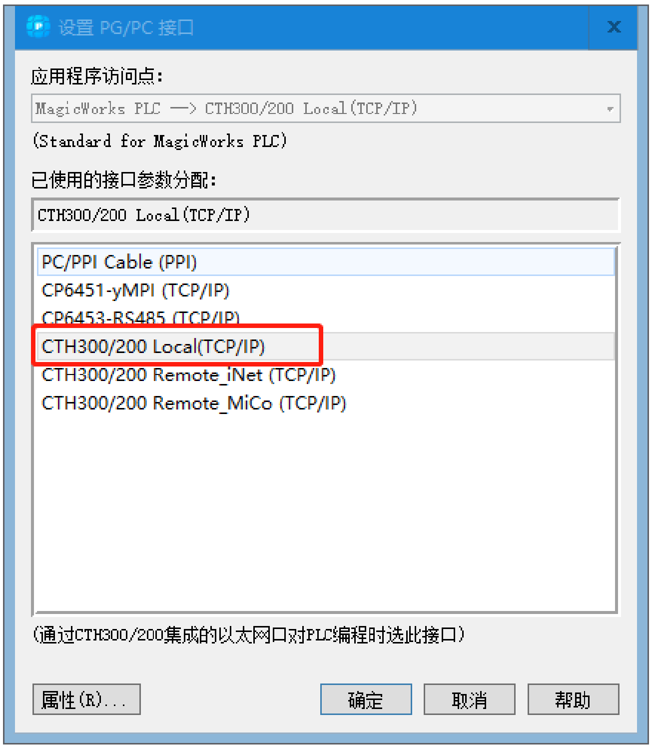


S7主站　　　　S7从站

1. Communication settings between main station M226ES PLC and Magicworks PLC programming software

Open the Magicworks PLC software, connect the M226ES PLC to the computer using an RJ45 Ethernet cable, keep the computer and PLC in the same network segment, open the communication interface, and click on Set PG/PC interface.

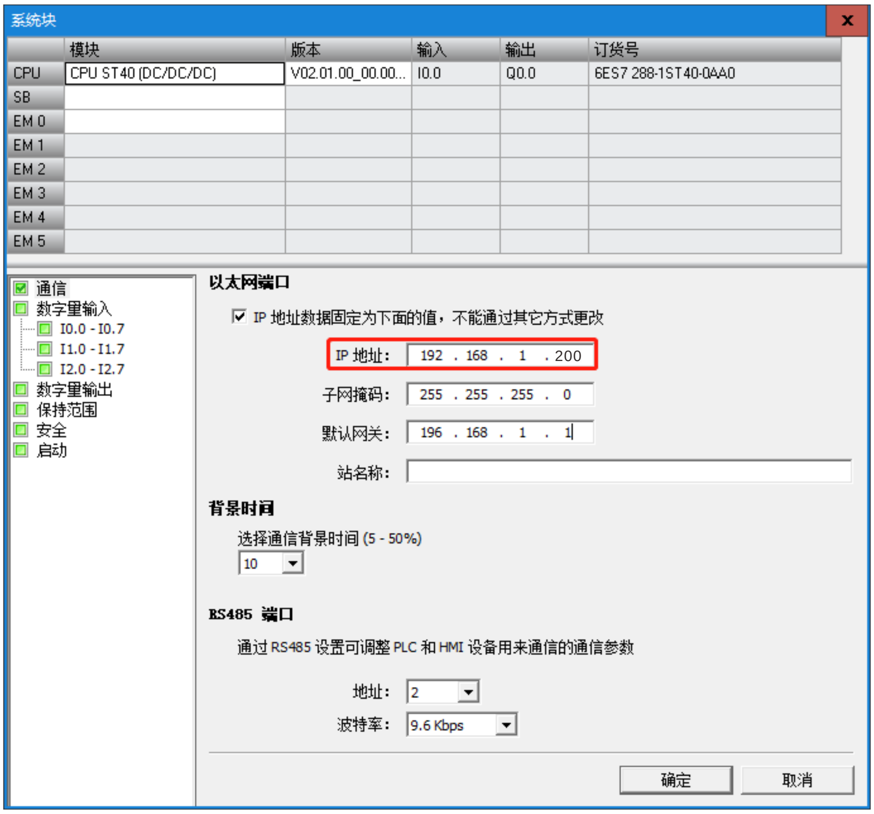
Select 'CTH300/200 Local (TCP/IP)' and then confirm.



Double click refresh on the communication interface to display CPU M226ES. Click on CPU M226ES to confirm.

Slave configuration

Open STEP7 MicroWin SMART, click on "System Fast" under the project tree, set the IP address of the slave station in the system block, and match the IP address with the master station.



Program 2 pairs of S7 main station (CPU M226ES)

Taking S7 network read and write as an example, you can perform network read and write operations on the S7 main station through the following methods.

Program the S7 master station using the S7 network read/write command S7_dead/S7w_write.

表5-2 Table of TABLE parameters for S7_dead/S7w_write instruction:

bit7	bit6	bit5	bit4	bit3~0	Byte offset
D	A	E	0	error code	0
The first byte of the remote station IP					one
Remote station IP second byte					two
The third byte of the remote station IP					three
The fourth byte of the remote station IP					four
Reserved (must be set to 0)					five
Reserved (must be set to 0)					six
The first byte of the pointer to the target area of the remote station					seven
The second byte of the pointer to the target area of the remote station					eight
The third byte of the pointer pointing to the target area of the remote station					nine
The fourth byte of the pointer to the target area of the remote station					ten
Length (not greater than 200)					eleven
The first byte of the pointer pointing to the target area of the local station					twelve
The second byte of the pointer pointing to the target area of the local station					thirteen
The third byte of the pointer pointing to the target area of the local station					fourteen
The fourth byte of the pointer pointing to the target area of the local station					fifteen

D: Completed bit, 0 incomplete, 1 completed

A: Activation bit, instruction in progress is 1, not executed is 0

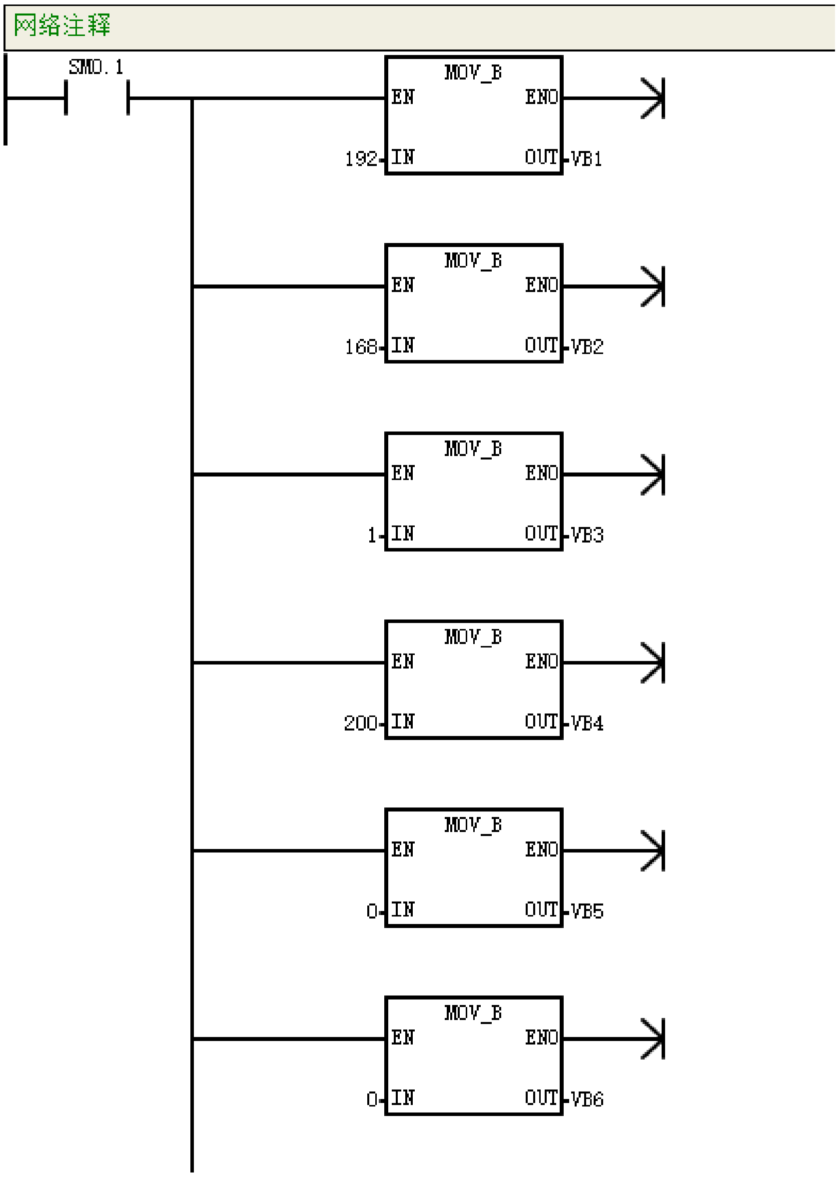
E: Bit, 0 is correct, 1 is incorrect

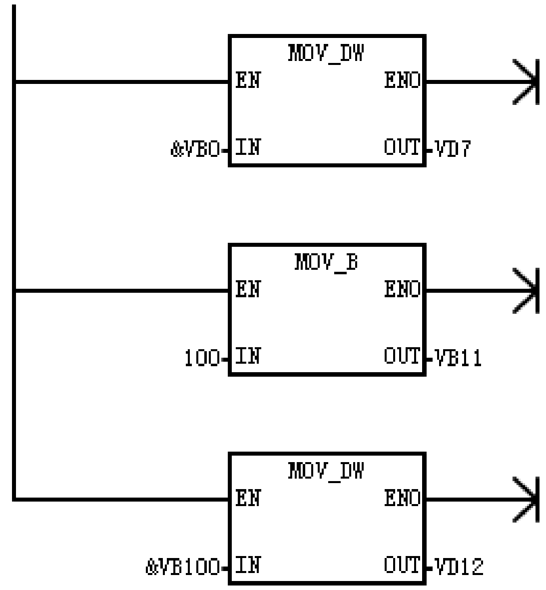
表5-3 **Error Code Table**

code	**explain**
0	No mistake
one	Timeout error: Remote station not responding
two	retain
three	Offline error: Conflict caused by duplicate site address or faulty hardware
four	Queue overflow error: 8 S7_ write/S7_ read boxes activated
five	Received error: The reply received is incorrect
six	Illegal parameter: S7w_write/S7_dead table contains an illegal or invalid value
seven	retain
eight	retain
nine	Information error: Incorrect data length

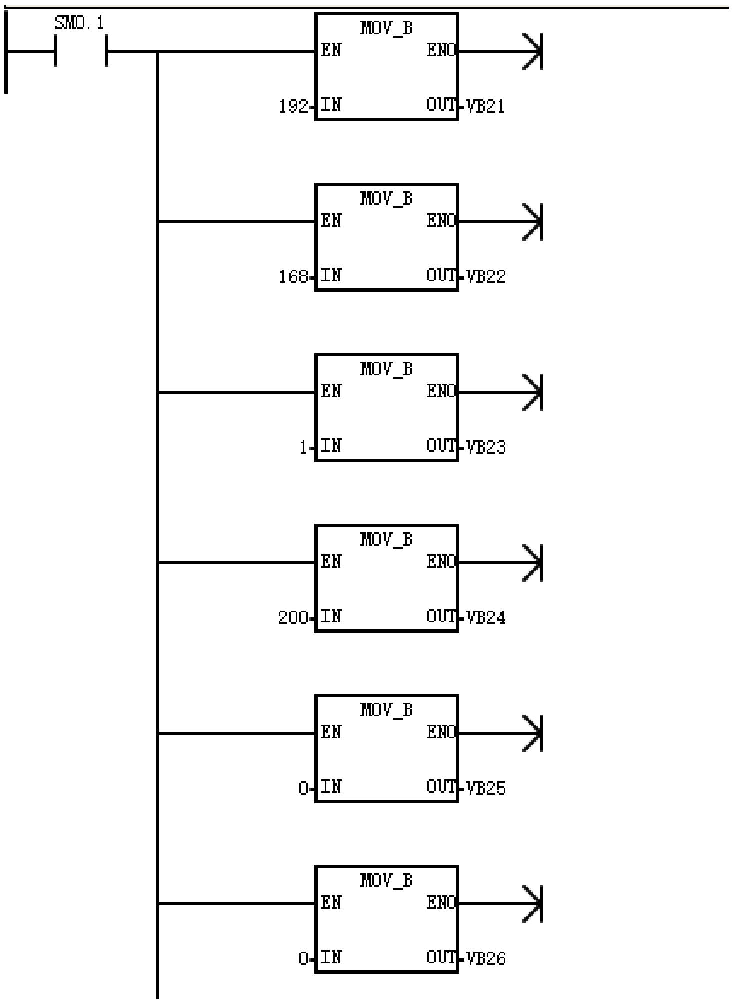
ten	The number of connections exceeds 8
eleven	The network cable is not plugged in

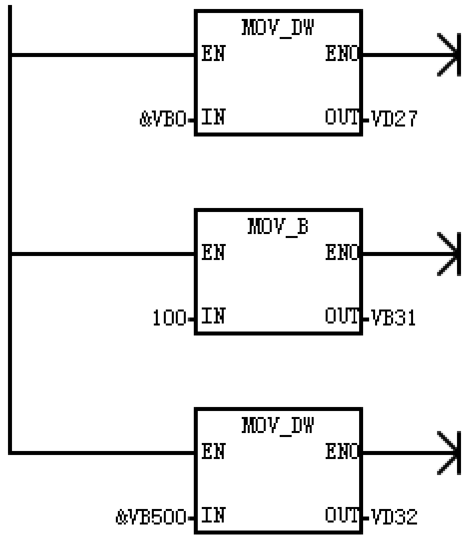
Network 1: Set the IP of the remote station to 192.168.1.200, the destination area pointer of the remote station to&VB0 (be sure to use a pointer), the read length to 100 bytes, and the destination area pointer of the local station to&VB100 (be sure to use a pointer) according to the TABLE parameter table of the S7_dead/S7w_write instruction. Configure the parameter table.



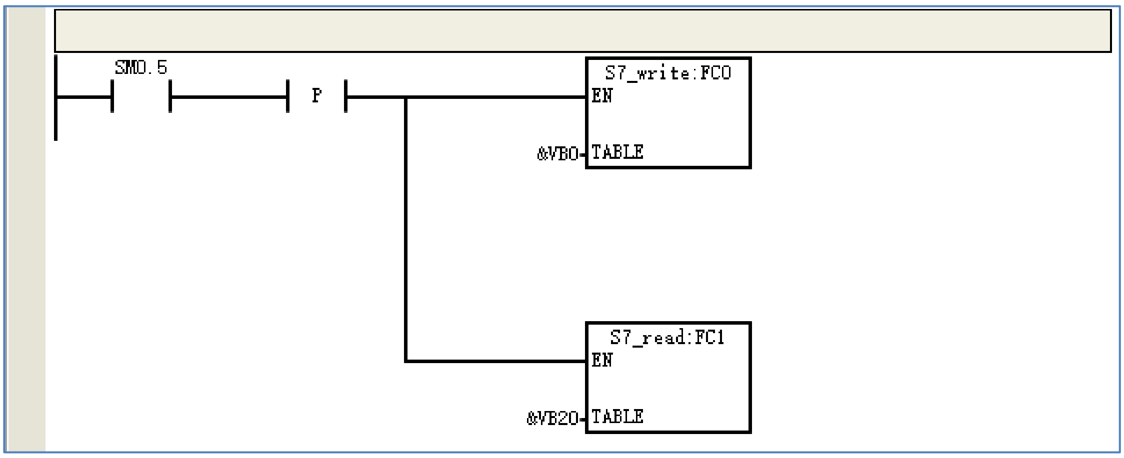


Network 2: Configure the parameter table according to the TABLE parameter table of the S7_dead/S7w_write instruction. Set the IP of the remote station to 192.168.1.200, the destination area pointer of the remote station to&VB0 (be sure to use a pointer), the write length to 100 bytes, and the destination area pointer of the local station to&VB100 (be sure to use a pointer).





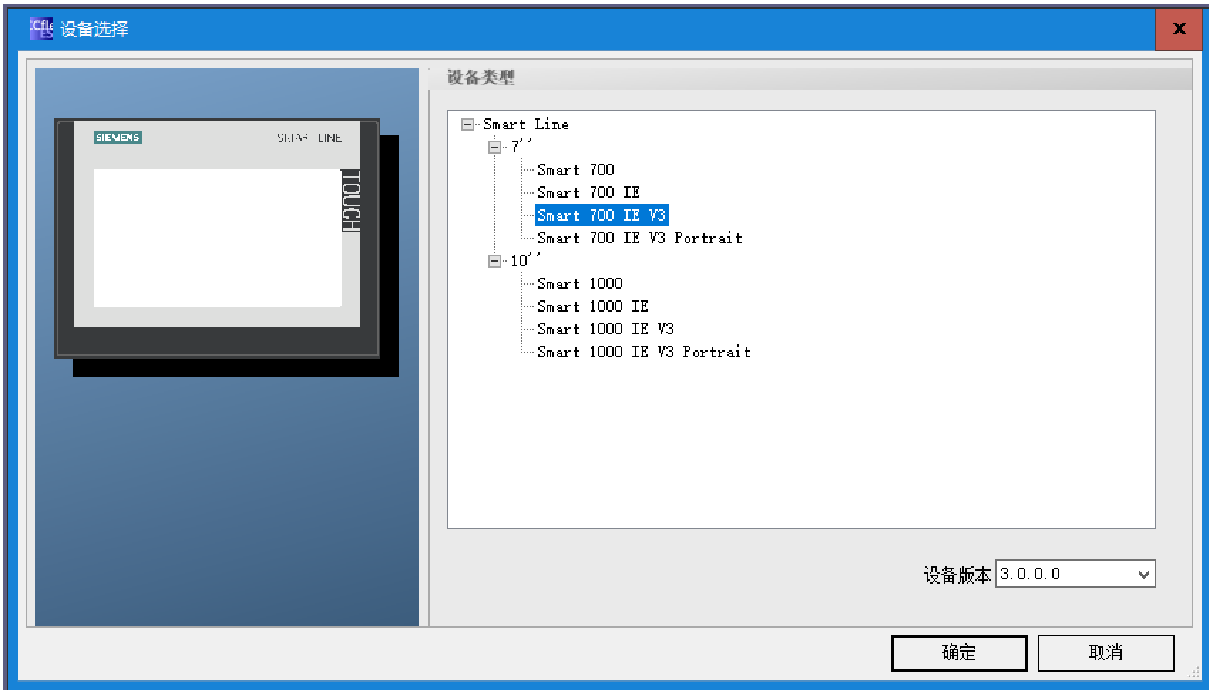
Network 3: Call the S7_dead/S7w_write instruction to read and write data through the parameters in the parameter table. The TABLE parameter of the instruction must use a pointer.

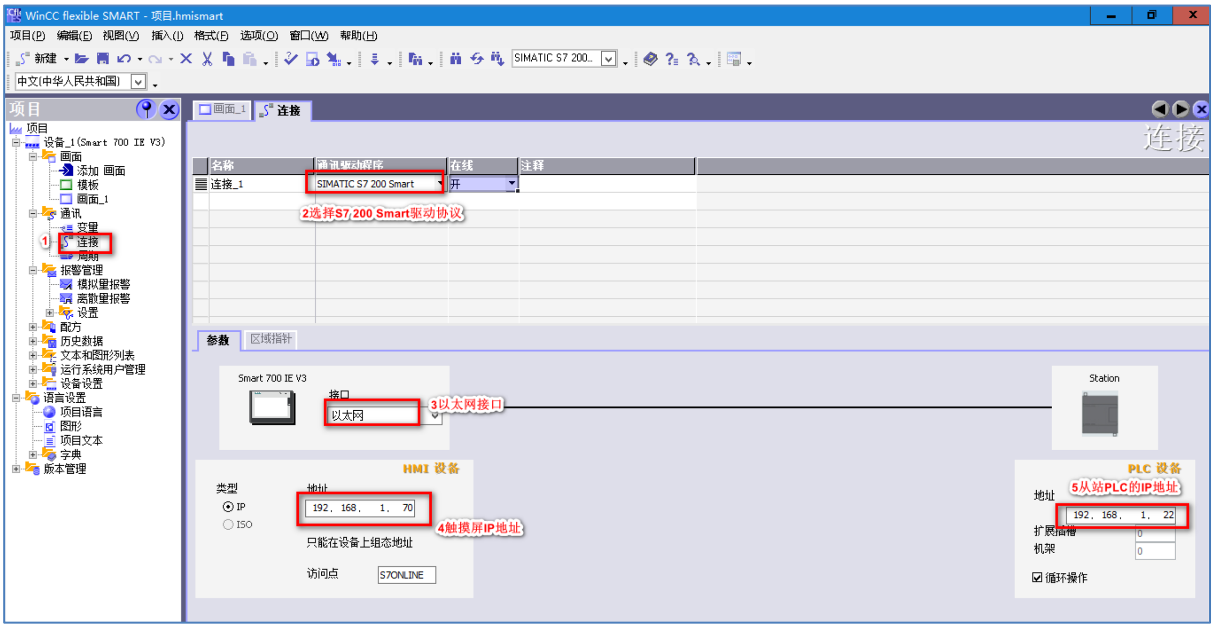


Example 3: S7 protocol communication between Siemens touch screen and CTMC series CPU (CTMC series CPU as slave)

This example uses Siemens touch screen as the master station and CTMC series CPU as the slave station to communicate with S7 protocol.

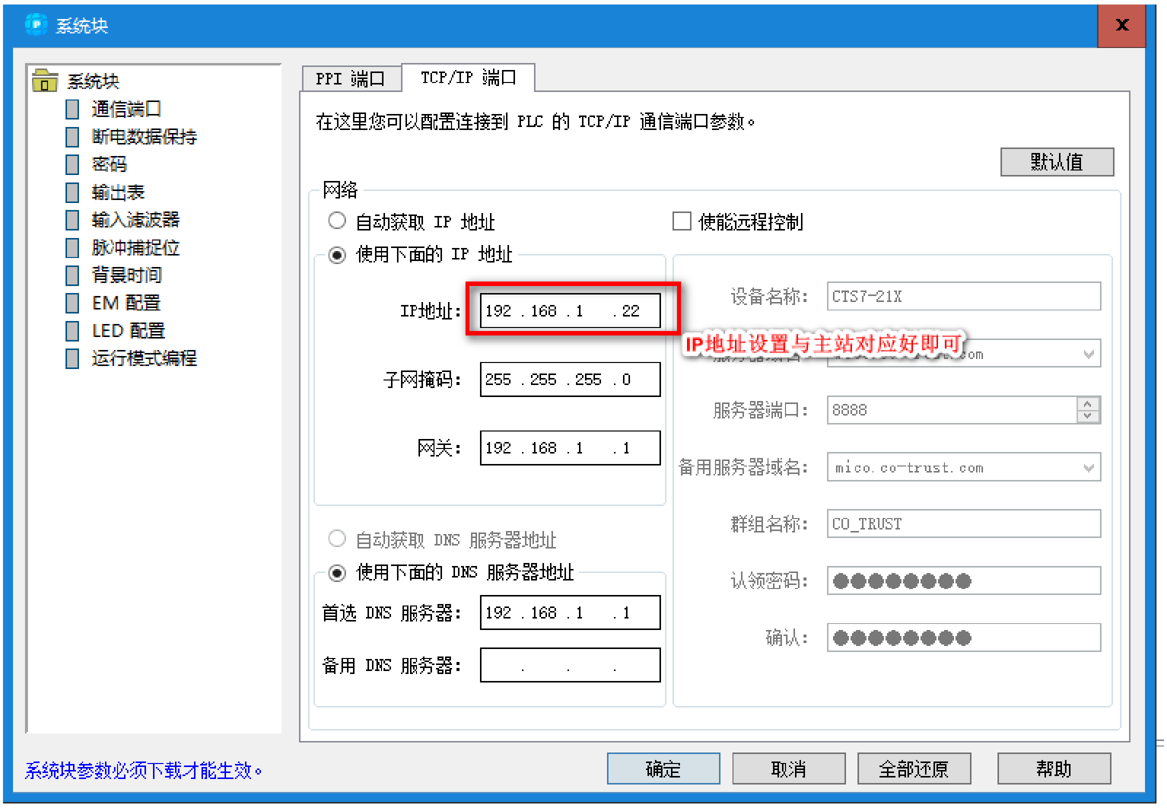
Main station configuration





Slave configuration

Simply configure the IP address in the system block of the CTMC series CPU:



Example 4: Communication between S7-1200/1500 and CTMC series PLC via S7 protocol (CTMC series CPU as slave)

The S7-1200/1500 CPU communicates with the CTMC series PLC using the S7 protocol (S7-1200/1500 serves as the master station). The COTRUST CTMC series PLC supports the Siemens S7 protocol slave station function, and the Siemens PLC or screen or upper computer software that supports the S7 protocol serves as the master station. The CTMC series CPU serves as the slave station for S7 protocol communication, supporting up to 8 connections (regardless of master-slave), with the COTRUST CPU only acting as the slave station.

Hardware and software requirements and completed communication tasks	
Hardware	● S7-1200 CPU hardware version V2.0 or higher ● CPU M226ES ● PC (with Ethernet card) ● TP Ethernet cable ● industrial switch
Software	● TIA portal V15 ● MagicWorks PLC (V2.29 or higher version)

Completed communication tasks:

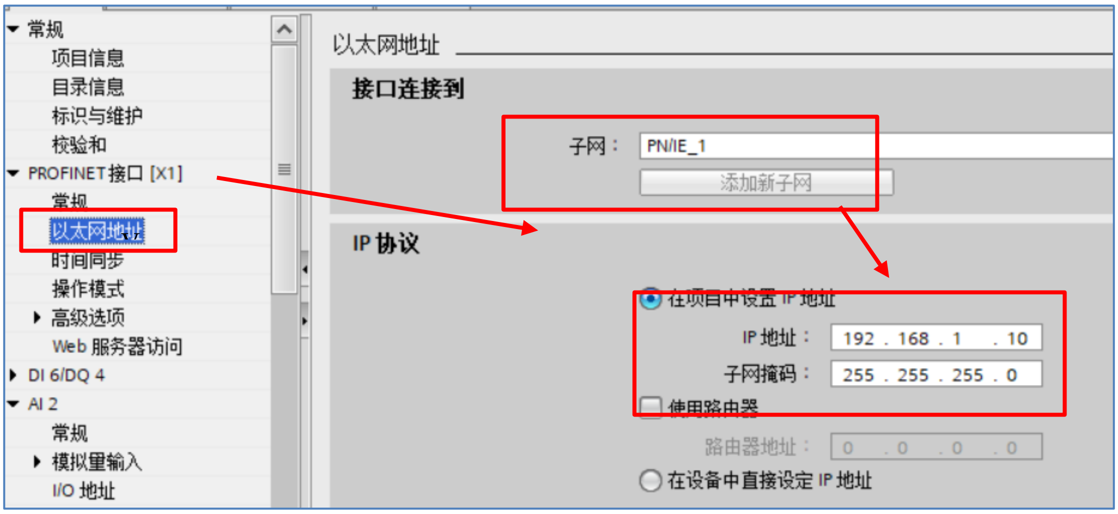
S7-1200 sends 100 bytes from the communication data area DB1 to the VB data area of M226ES, and reads the VB data area from M226ES and stores it in the DB1 data area of S7-1200.

10 consecutive bytes from MB0-MB9 in the M area of S7-1200 are sent to MB0-MB9 in the M area of M226ES, and S7-1200 reads 10 consecutive bytes from MB10-MB19 in the M226ES area to MB10-MB19 in the M area of S7-1200.

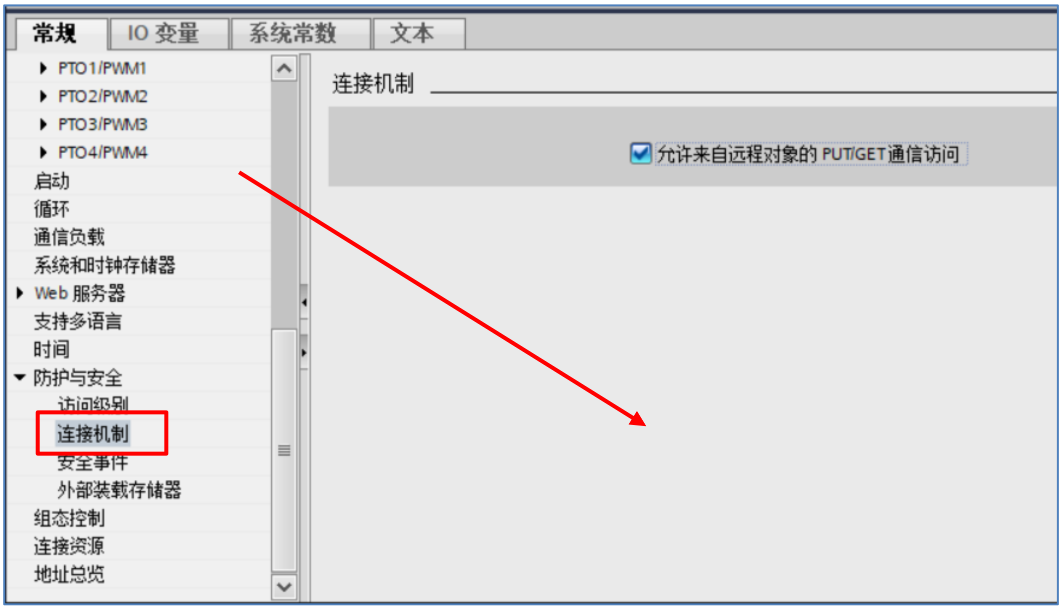
Configuration programming of S7-1200 main station

Step 1: Create a new project using TIA portal V15 software and complete hardware and network configuration

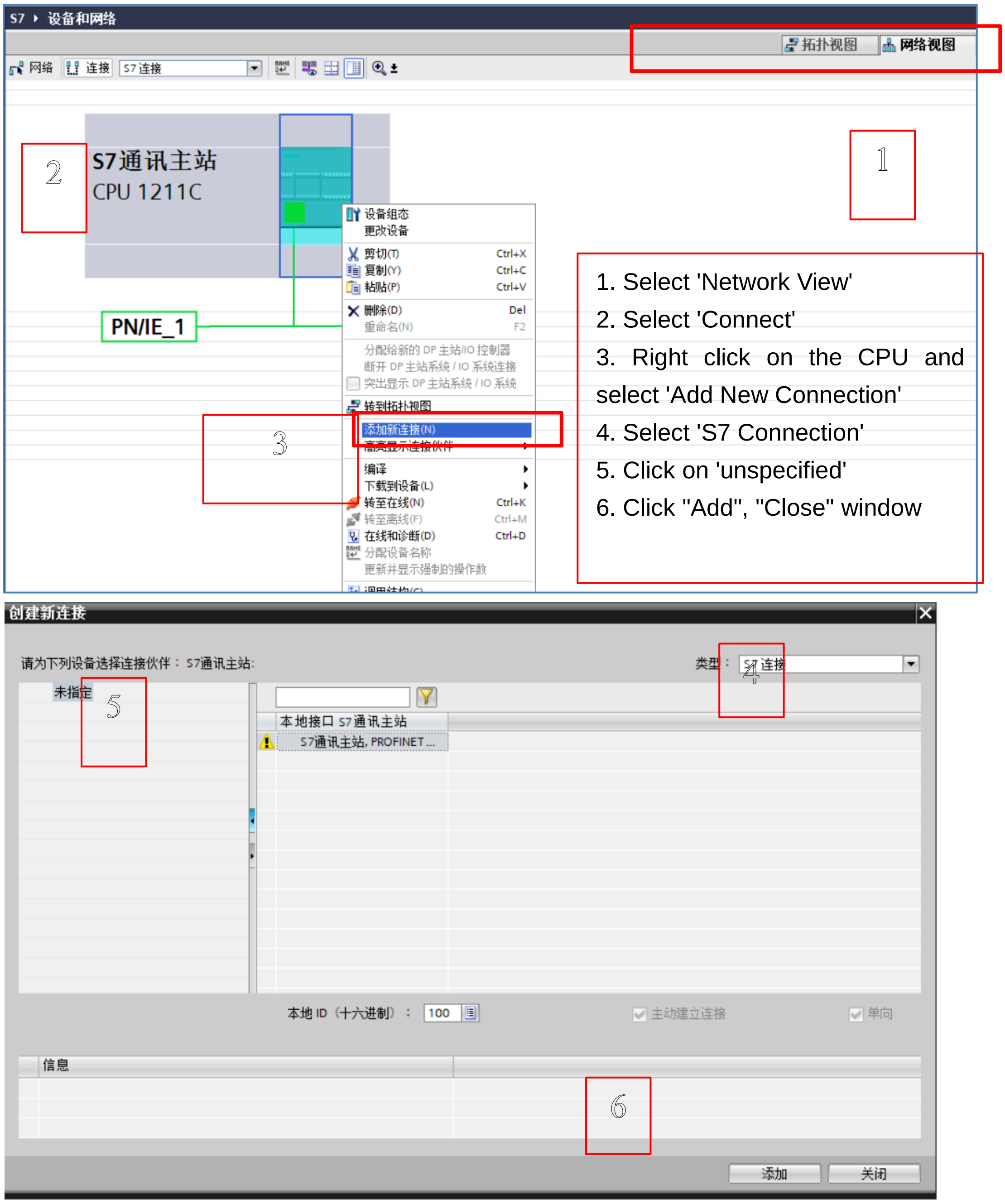
1) Create a new subnet PN/IE_1 and modify the IP address to be in the same network segment as the PC and M226ES



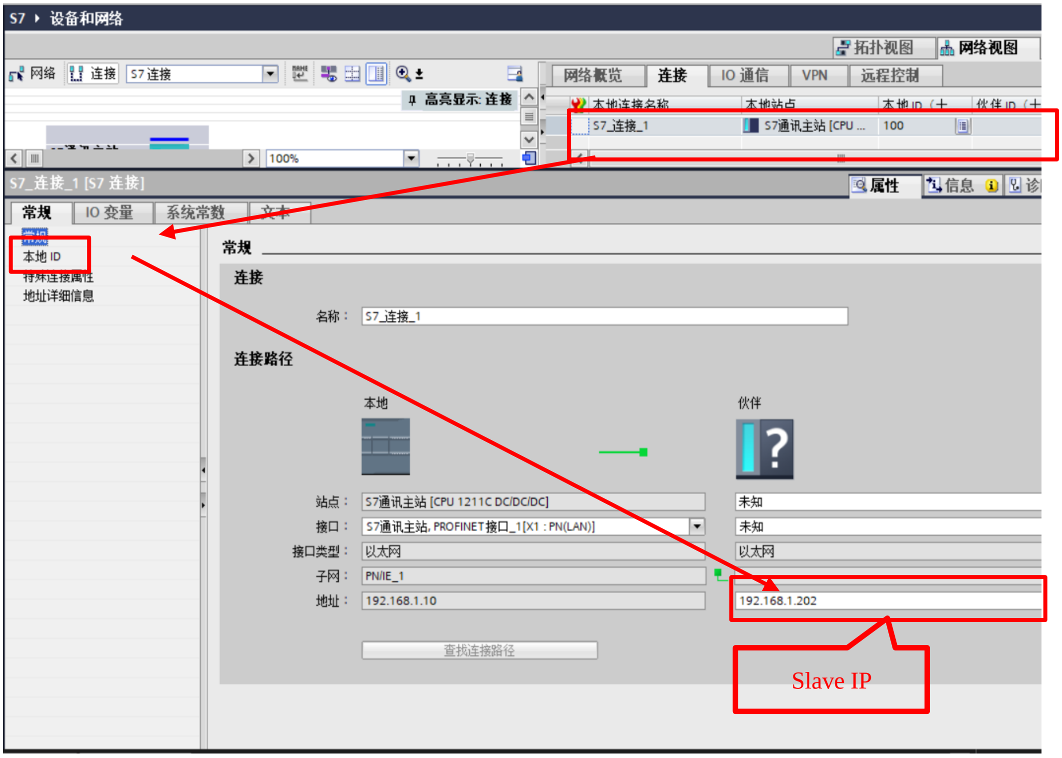
2) Check "Protection and Security">"Connection Mechanism">"Allow PTT/GET communication access from remote objects"



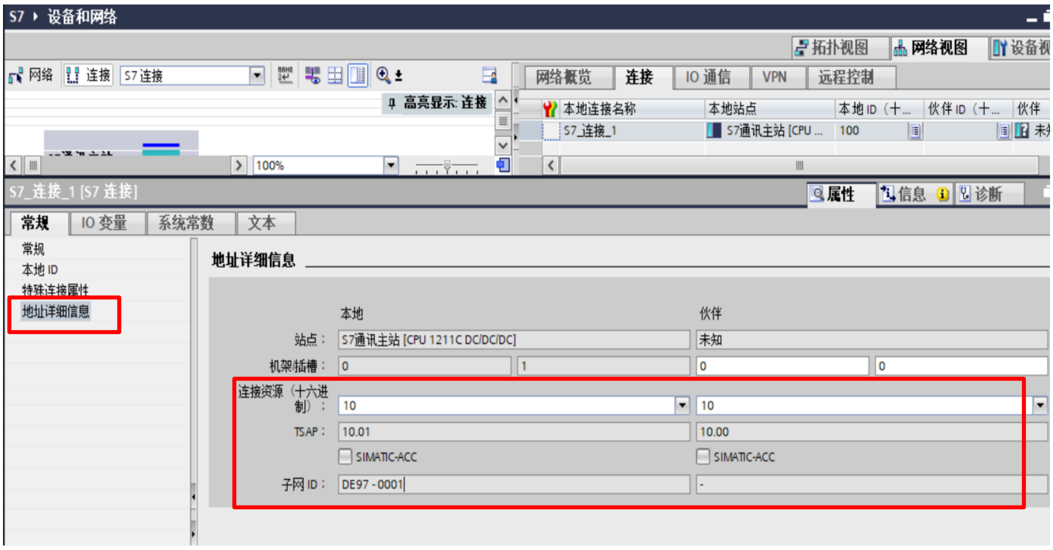
3) Under "Project Tree">"Device Configuration">"Network View", follow the steps 1-6 in the following picture to establish an S7 connection



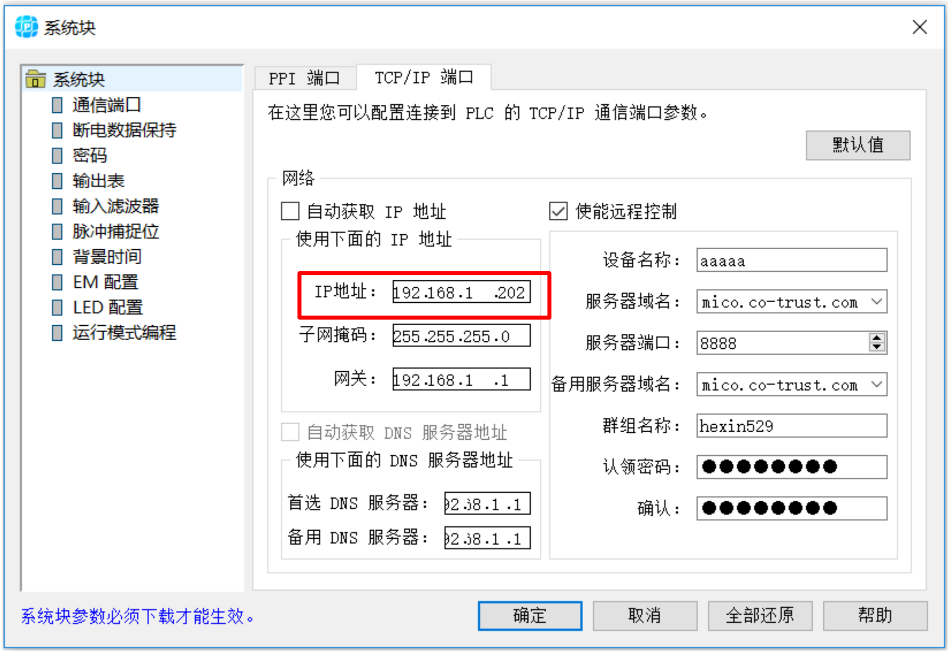
4) Fill in the connection parameters according to the steps in the following image



5) Set the TSAP address of the communication partner in the "Address Details" section

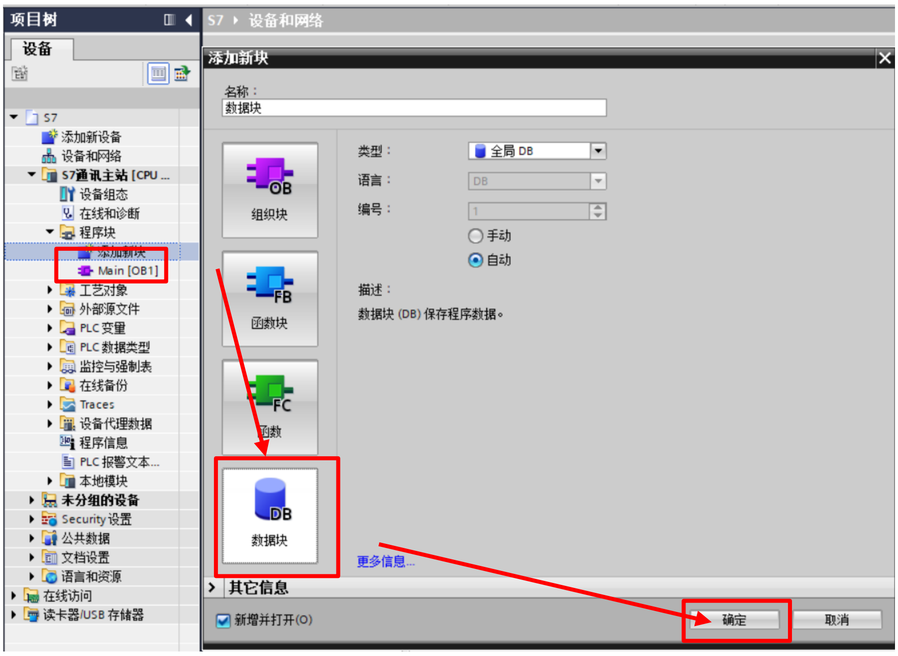


6) Simply configure the IP address in the system block of the CTMC series CPU, as shown below

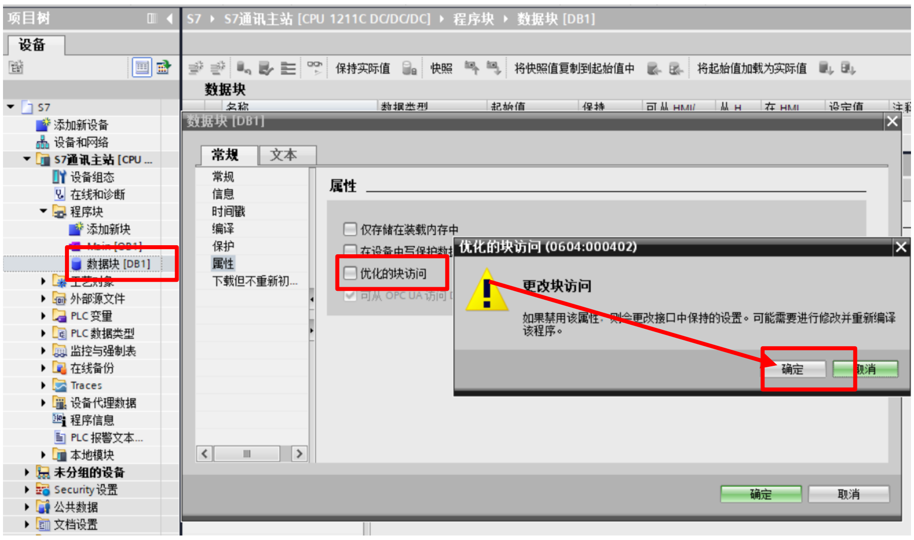


Step 2: Software Programming

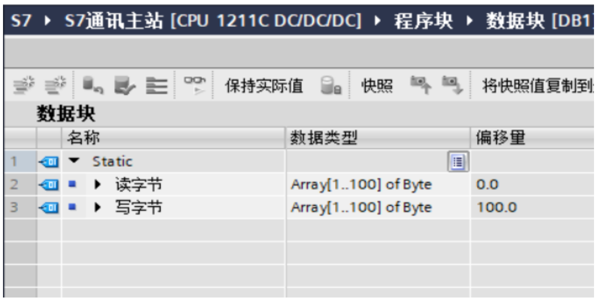
1) Create data block DB1 and define two arrays of 100 bytes each



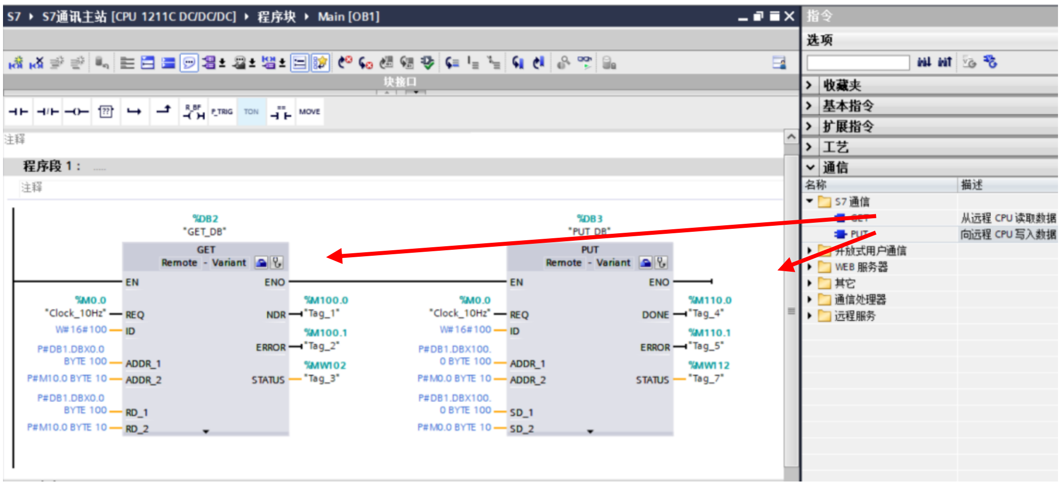
2) Under "Project Tree">"Data Blocks">"Properties", uncheck the option for optimizing block access.



3) Create write and read buffers for data block DB1, and define two sets of arrays of 100 bytes each



4) In OB1, under "Instructions" -->"S7 Communication", call the Get and Put communication instructions. The program is written as follows:



CALL 'GET',% DB2 //Call PUT, using background DB block: DB2

REQ :=% M0.0 //System clock 0.1 second pulse

ID :=W # 16 # 0100//Connection number, to be consistent with the connection configuration, the connection number used when creating the connection

DONE :=% M100.0//When it is 1, sending is complete

ERROR :=% M100.1 //When it is 1, a fault occurs

Status :=% MW102 //Status code

ADDR_1 :=P # DB1.DBX0.0 BYTE 100//Address for reading data from the communication partner data area

ADDR_2 :=P # M10.0 BYTE 10 //Address for reading data from the communication partner data area

SD_1 :=P # DB1.DBX0.0 BYTE 100 //Local receiving data address

SD_2 :=P # M10.0 BYTE 10 //Local send data area

CALL 'PUT',% DB3 //Call GET, using background DB block: DB3

REQ :=% M0.0 //System clock 0.1 second pulse

ID :=W # 16 # 0100//Connection number, to be consistent with the connection configuration, the connection number used when creating the connection

NDR :=% M110.0 //When it is 1, new data is received

Error :=% M110.1 //When it is 1, a fault occurs

Status :=% MW112 //Status code

ADDR_1 :=P # DB1.DBX100.0 BYTE 100//Address sent to communication partner data area

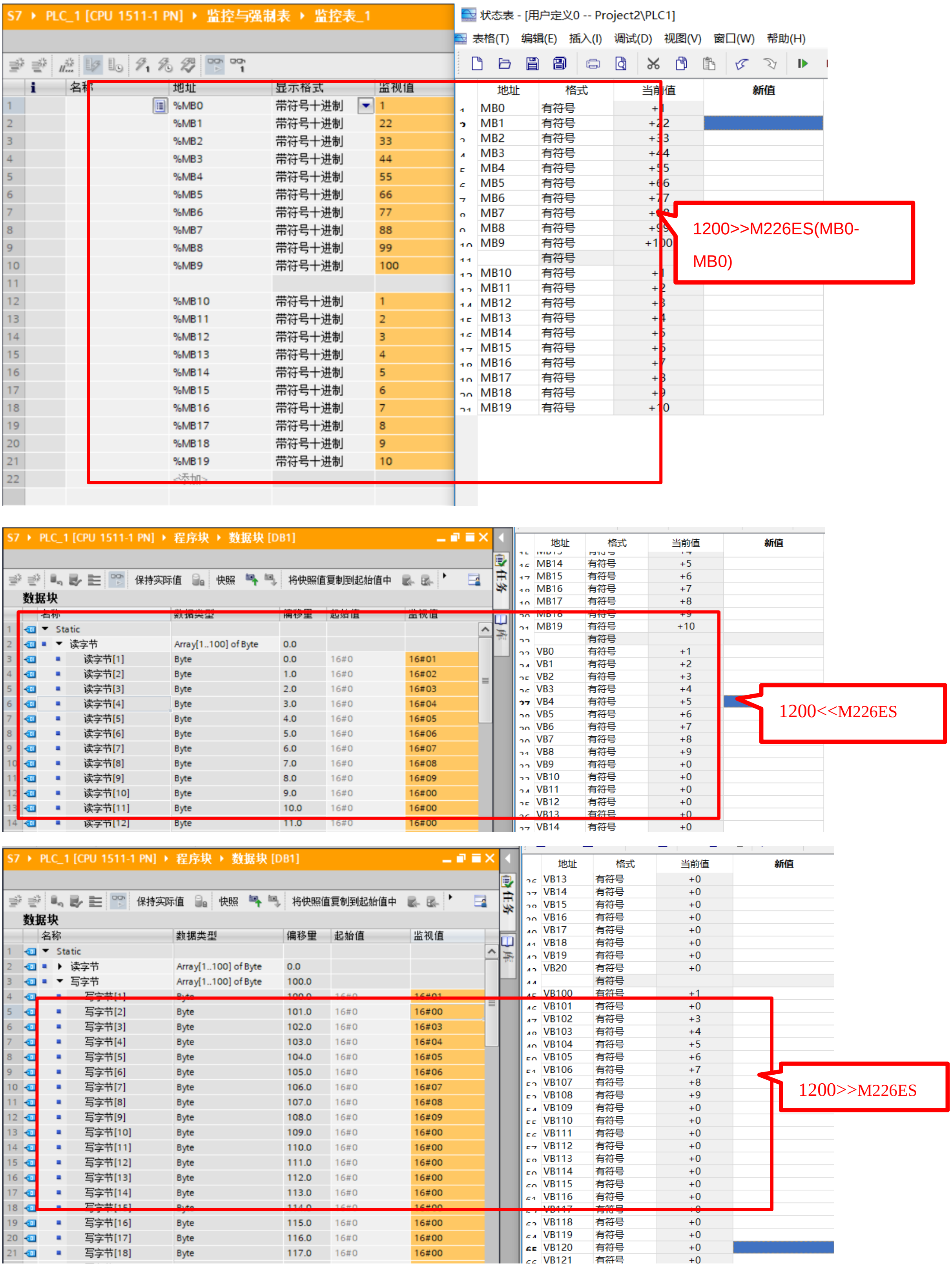
ADDR_1 :=P # M0.0 BYTE 10 //Address sent to communication partner data area

RD_1 :=P # DB1.DBX100.0 BYTE 100//Local send data area

RD_1 :=P # M0.0 BYTE 10 //Local send data area

Step 3: Monitoring Results

By programming on the S7-1200 side for S7 communication, data exchange between two CPUs is achieved. Monitoring results:



⚠	**be careful** The V area in M226ES corresponds to DB1, which is the communication partner data area used in the PUT instruction ADDR_1=P # DB1. DBX0.0 BYTE 100 is VB0~VB99 in M226ES. The communication partner data area ADDR_1=P # DB1.DBX10.0 used in the GET instruction is VB100~VB199 for BYTE 100 in M226ES.

5.6.4 Network port communication between other products and CTMC series CPUs

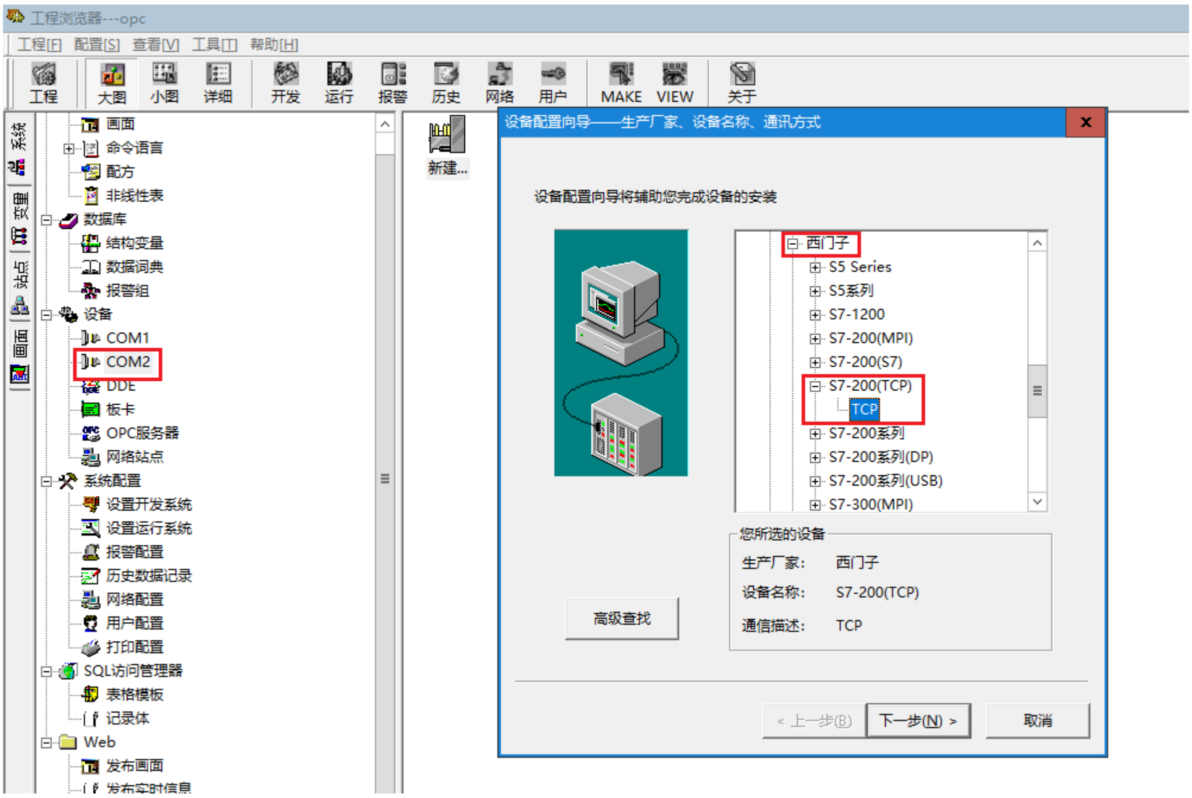
Example 1: S7 protocol communication between Kingview and CTMC series CPU (CTMC series CPU acts as a slave)

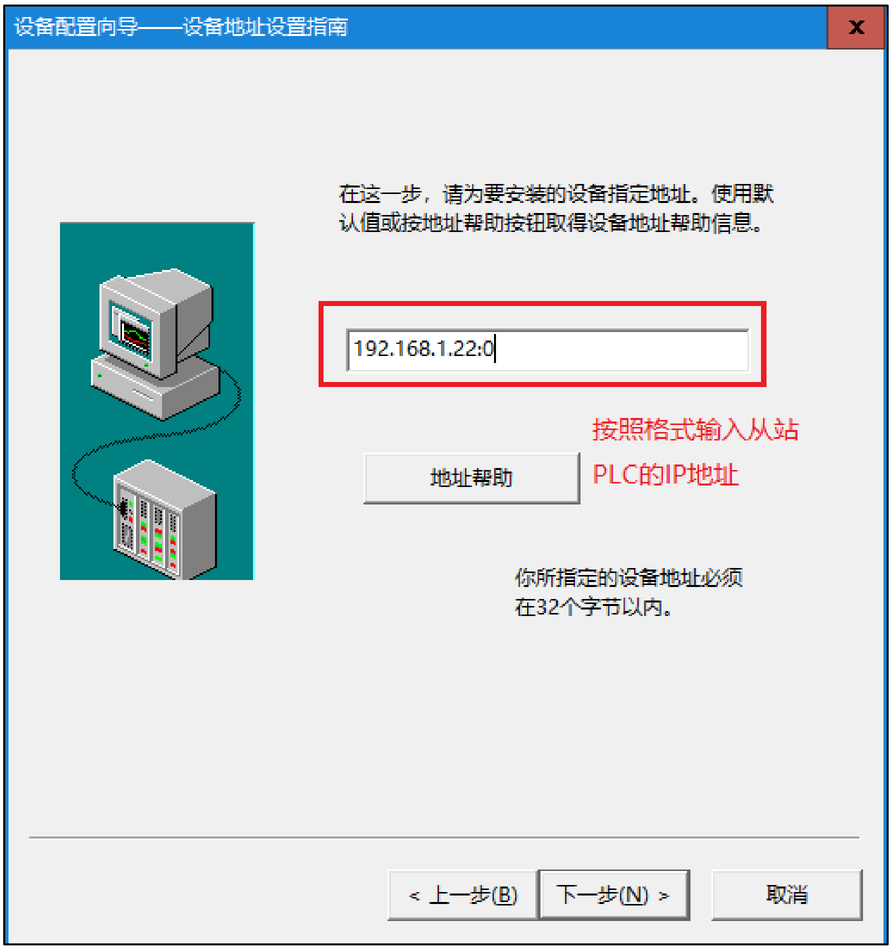
This example uses Kingview as the master station and CTMC series CPU as the slave station to communicate with S7 protocol.

Main station configuration

Refer to the following steps to configure the S7 protocol master station for the Kingview software:

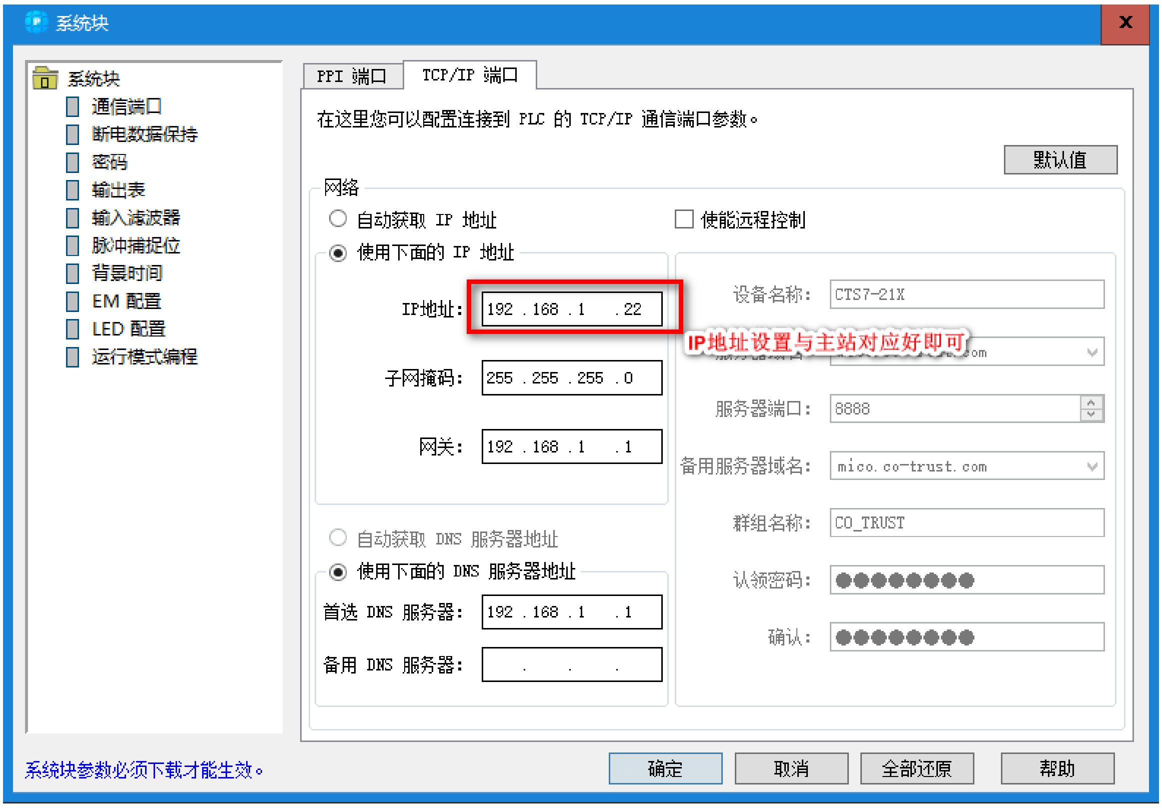
1) Select S7 communication protocol in the device configuration wizard for serial port





Slave configuration

Simply configure the IP address in the system block of the CTMC series CPU, as shown below:



Example 2: Communication between Kunlun Tong State Touch Screen and CTMC Series CPU via S7 Protocol (CTMC Series CPU acts as a slave)

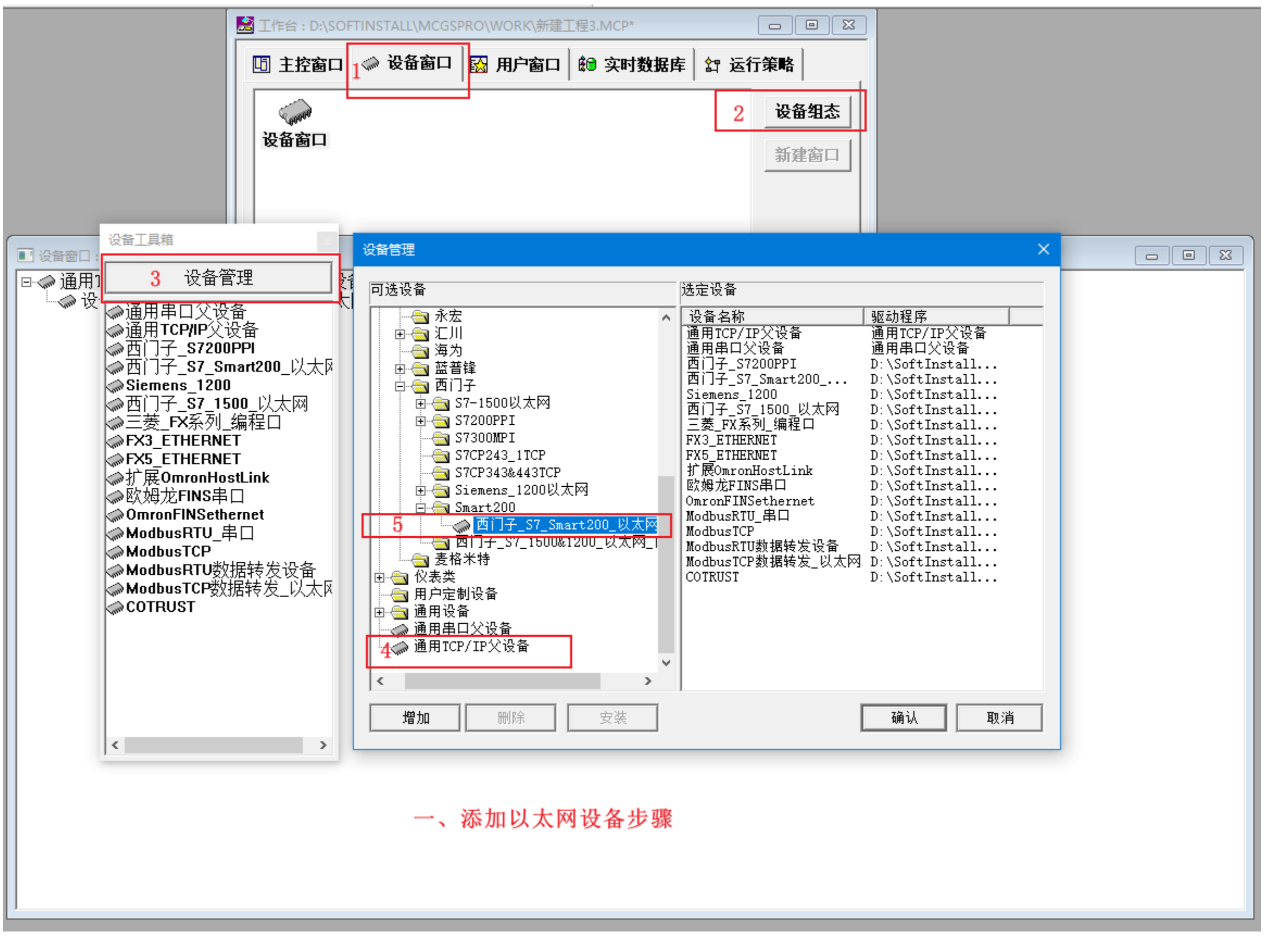
In this example, the Kunlun Tong state touch screen is used as the master station, and the CTMC series

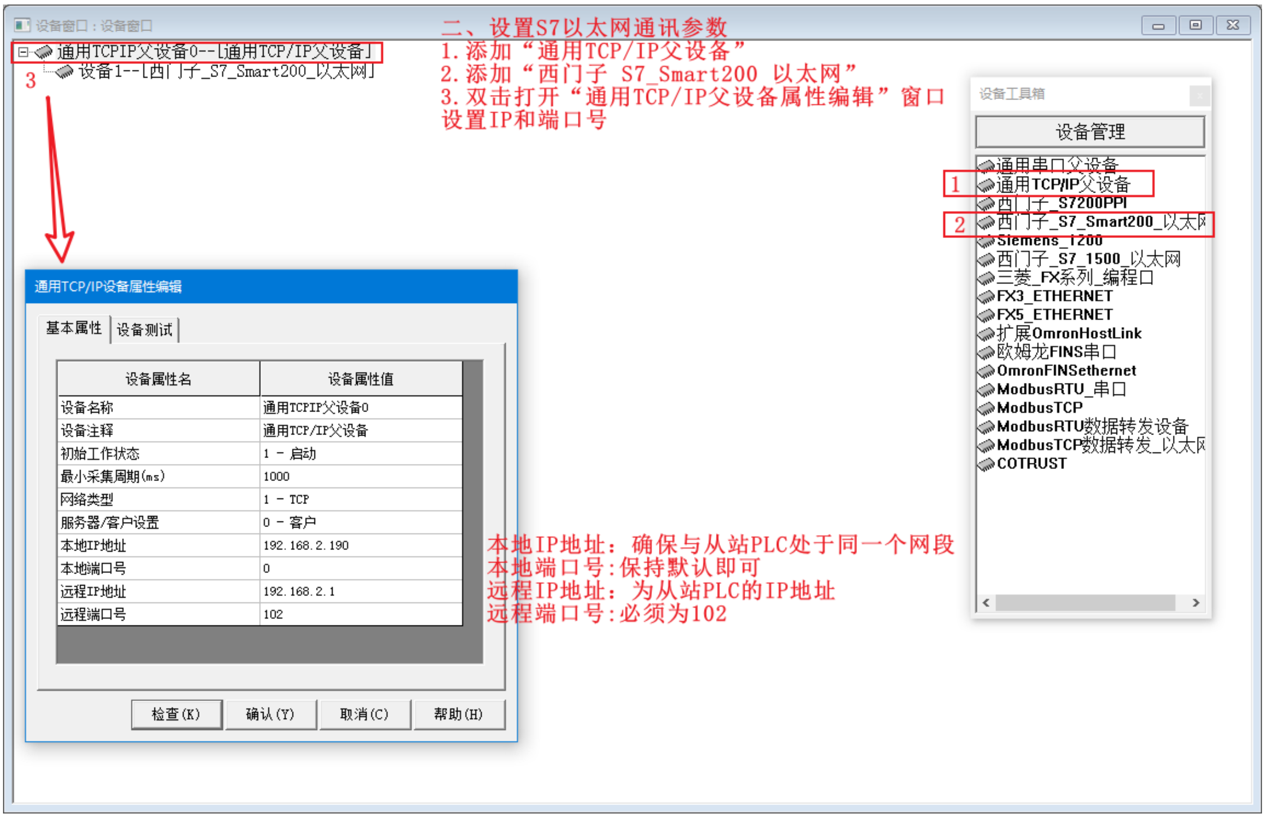
CPU is used as the slave station to communicate using the S7 protocol.

Note: The Kunlun communication upper computer can also serve as the main station for S7 protocol communication with CTMC series CPUs.

Main station configuration

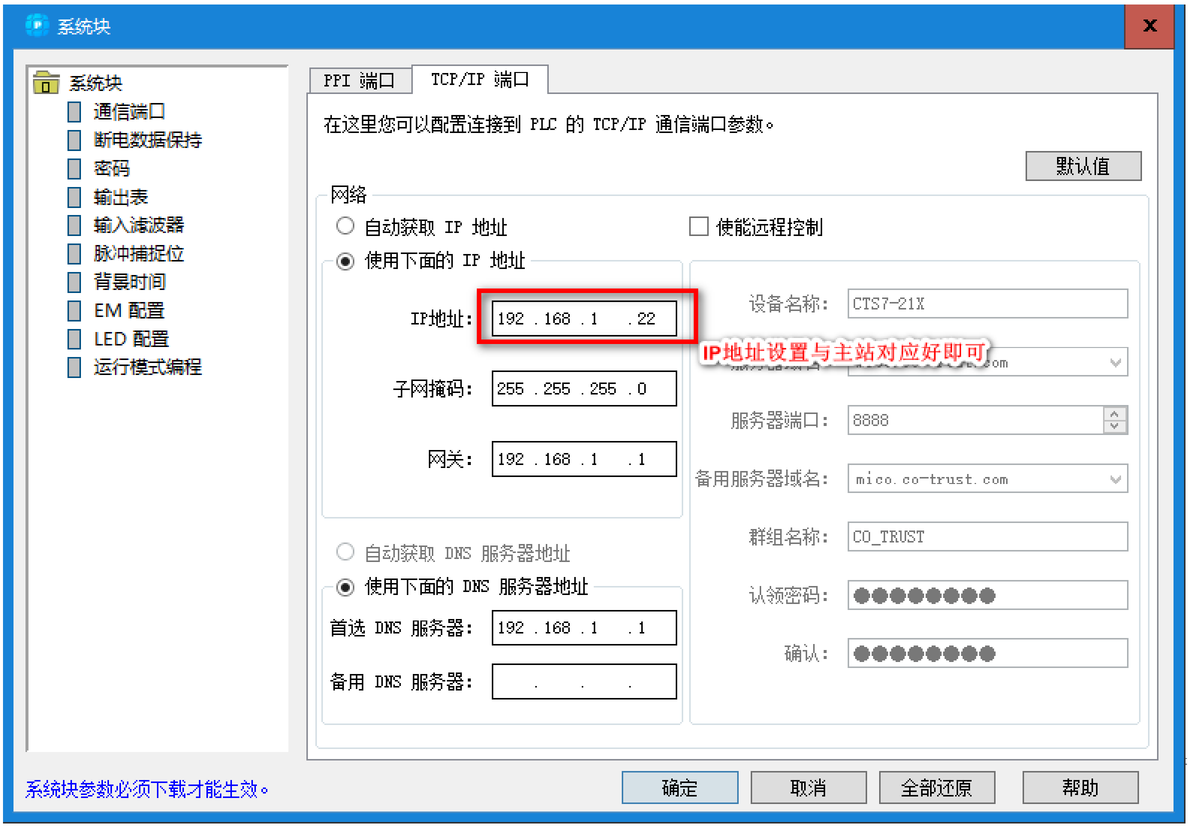
Refer to the following steps to configure the main station of the Kunlun Tong state touch screen with S7 protocol:





Slave configuration

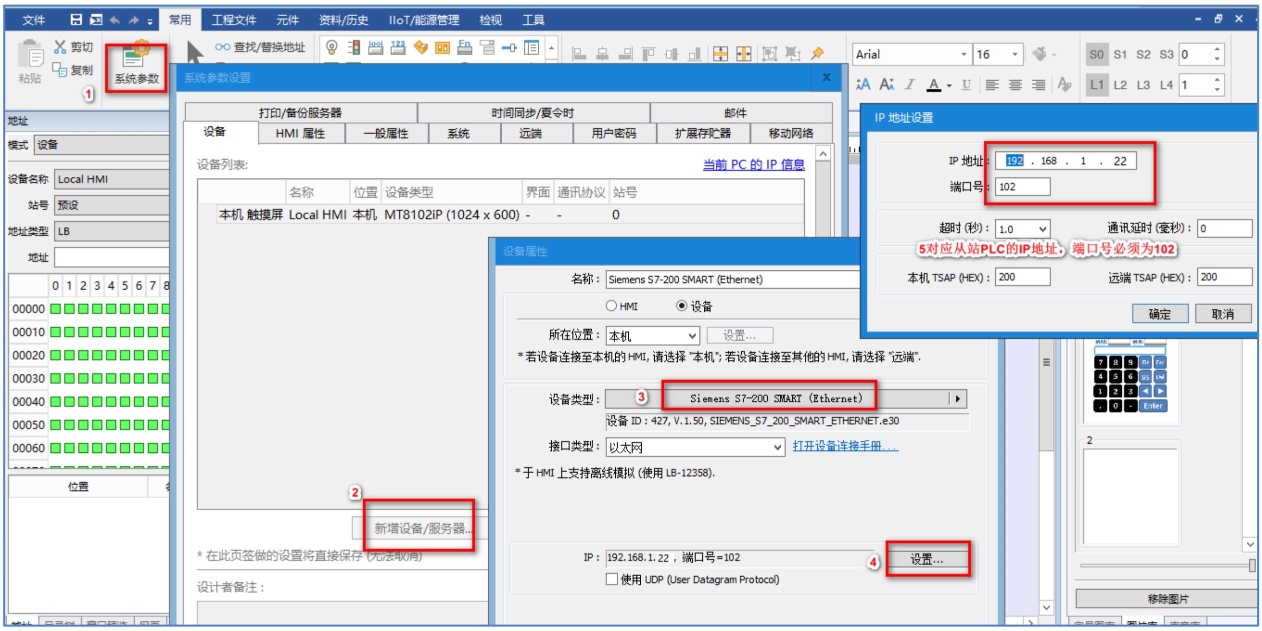
Simply configure the IP address in the system block of the CTMC series CPU, as shown below:



Example 3: S7 protocol communication between Weilun Tong touch screen and CTMC series CPU (CTMC series CPU acts as a slave)

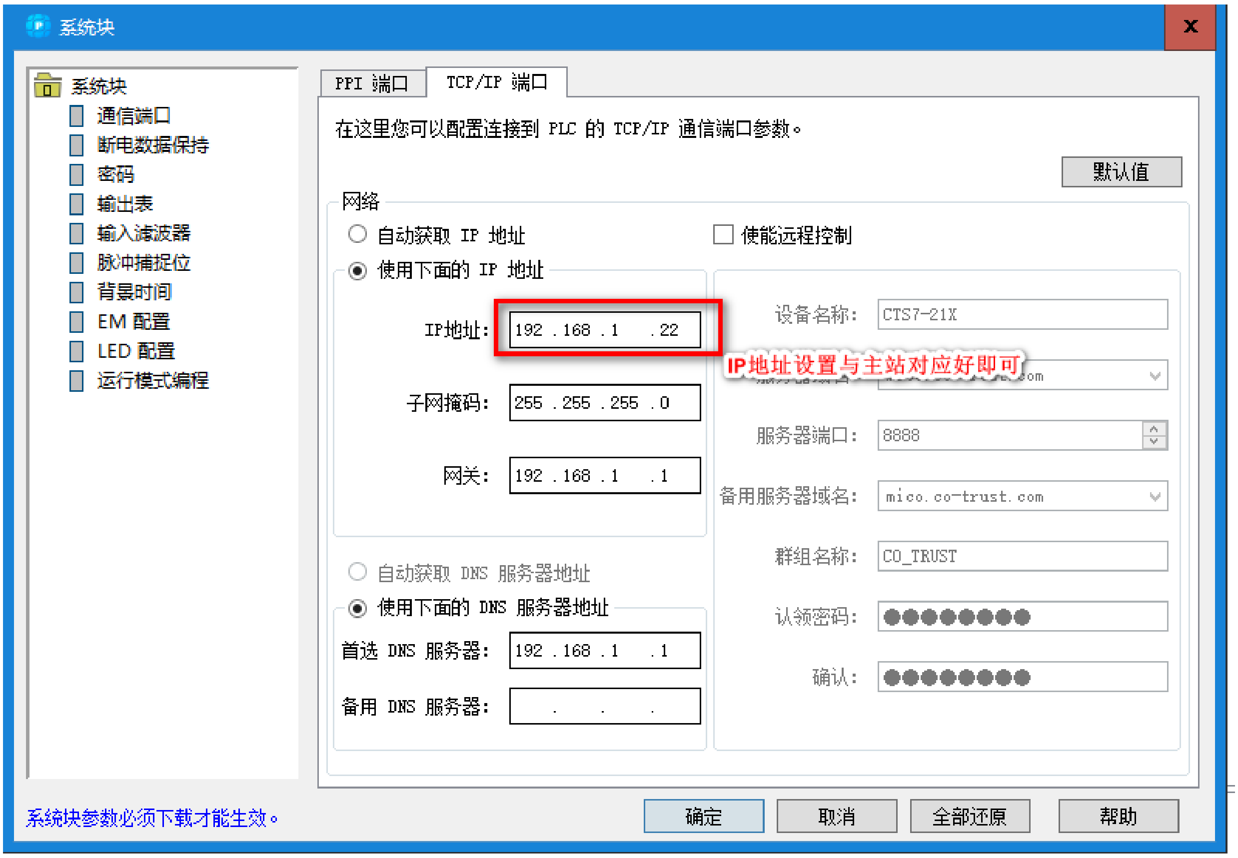
In this example, the Weilun Tong touch screen is used as the master station, and the CTMC series CPU is used as the slave station for S7 protocol communication.

Main station configuration



Slave configuration

Simply configure the IP address in the system block of the CTMC series CPU, as shown below:

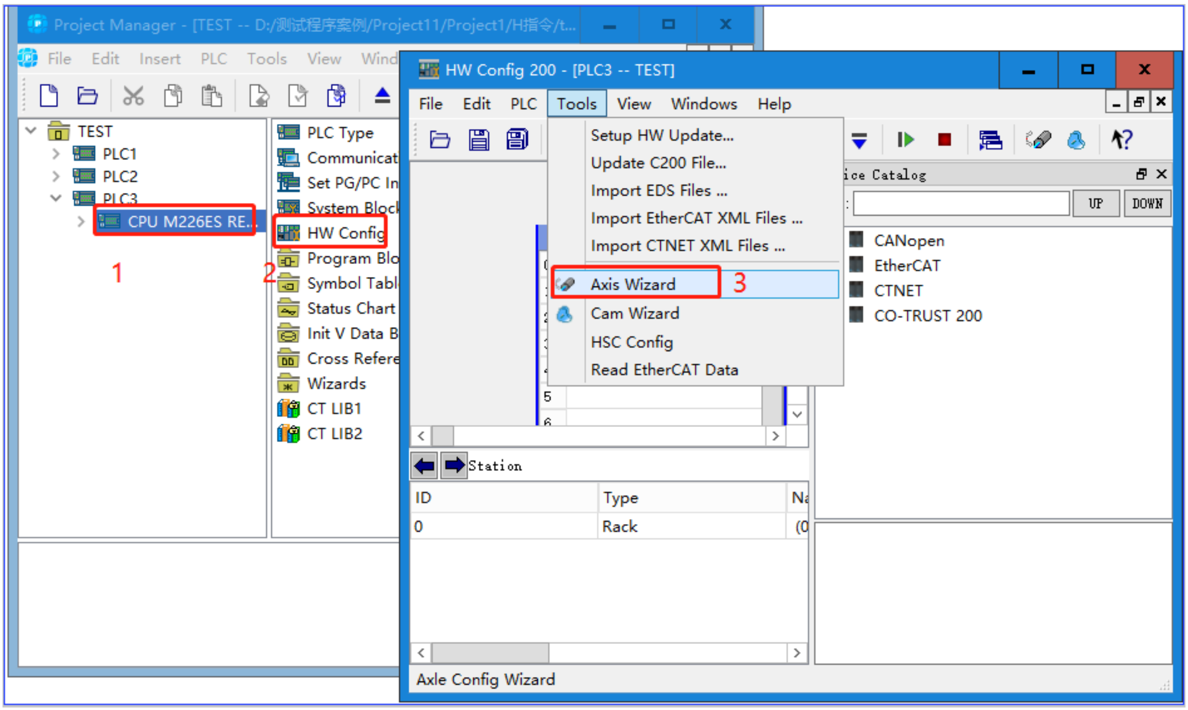


Axis configuration and electronic cam

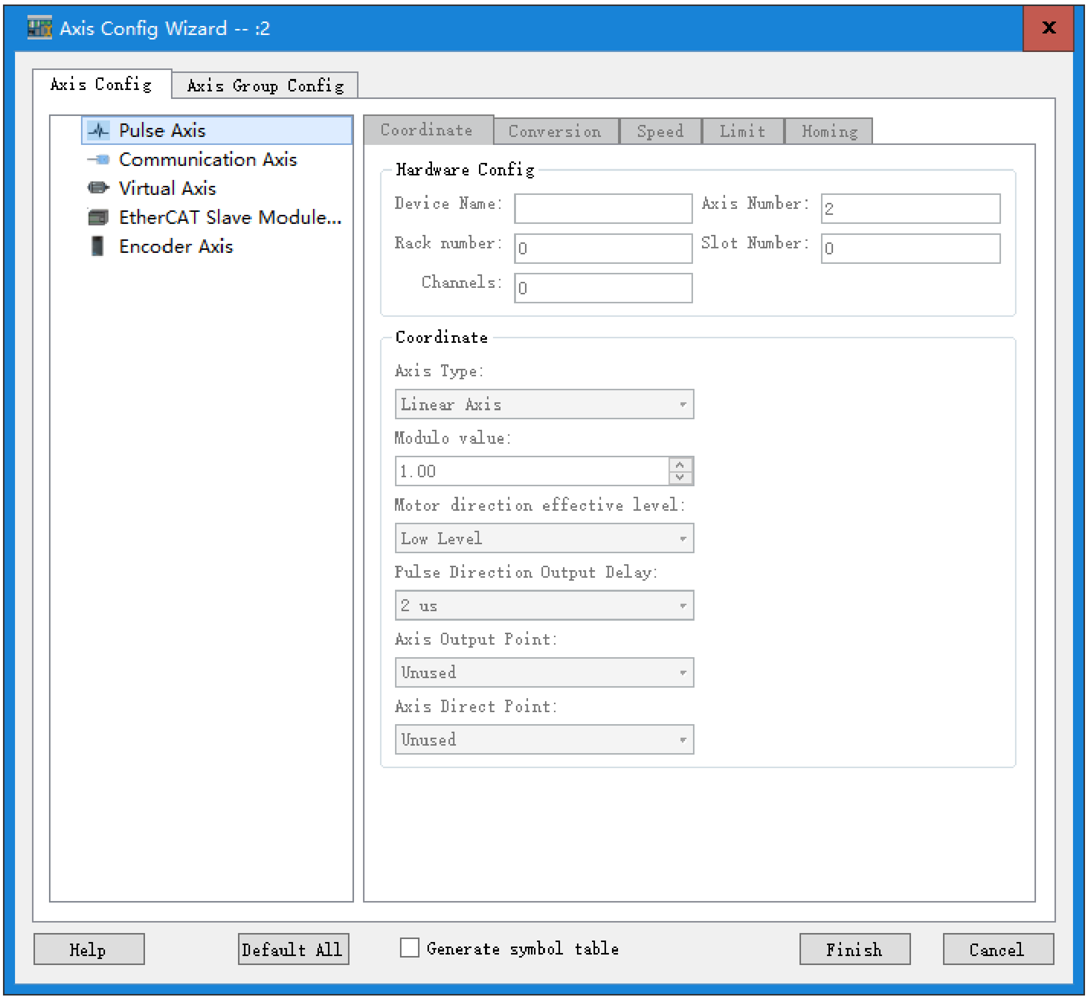
6

6.1 Axis Configuration Wizard

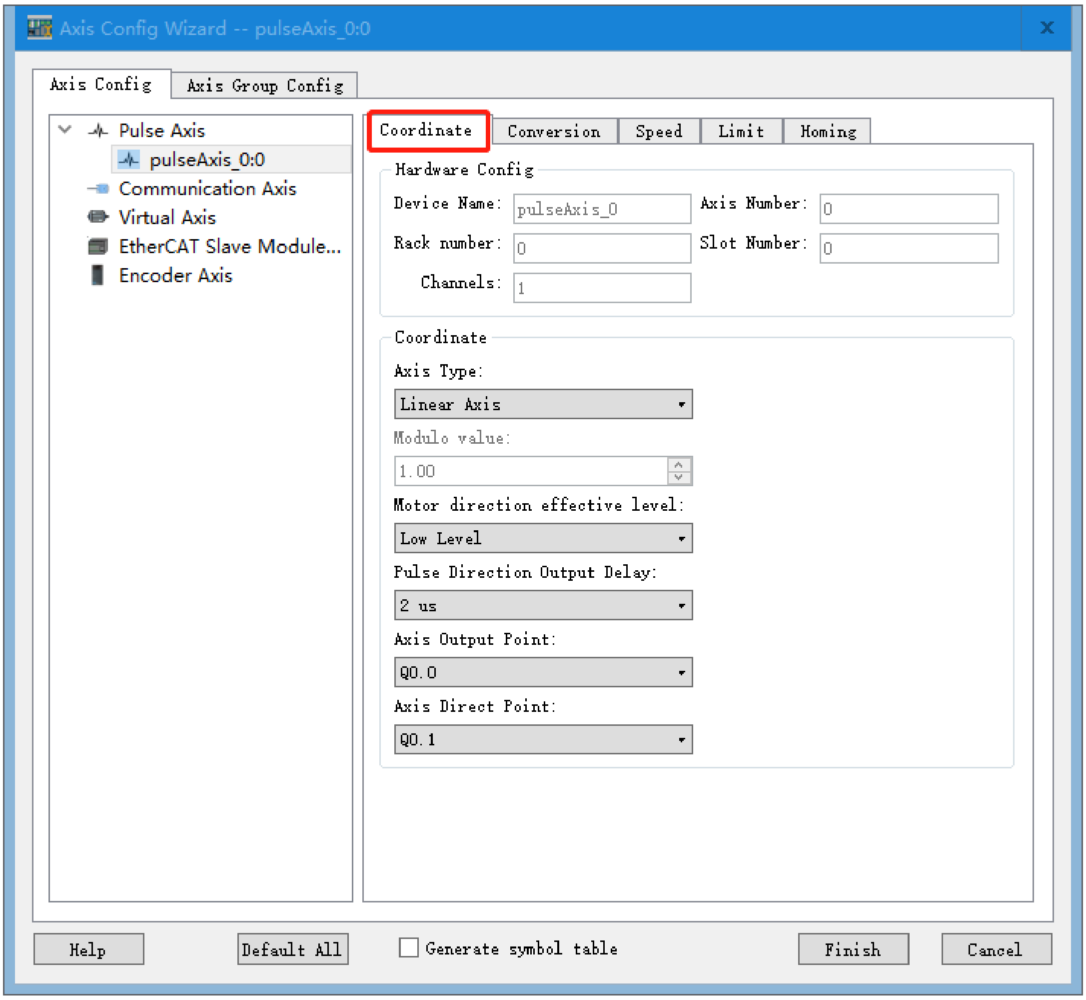
Select [Tools] → [Axis Configuration Wizard] in the hardware configuration or directly click the icon to configure the axis, as shown in the following figure.



The axis configuration wizard interface is shown below: Double click on the axis to add the desired axis.



[Coordinate]



Equipment name: the axis name set when adding the axis.

Axis number: The sequence ID when adding an axis, which cannot be modified and increases sequentially with the addition of the axis. If a certain axis is deleted, the newly added axis number will be the previous deleted axis number.

Rack number: corresponding to the rack where the hardware configuration axis is located.

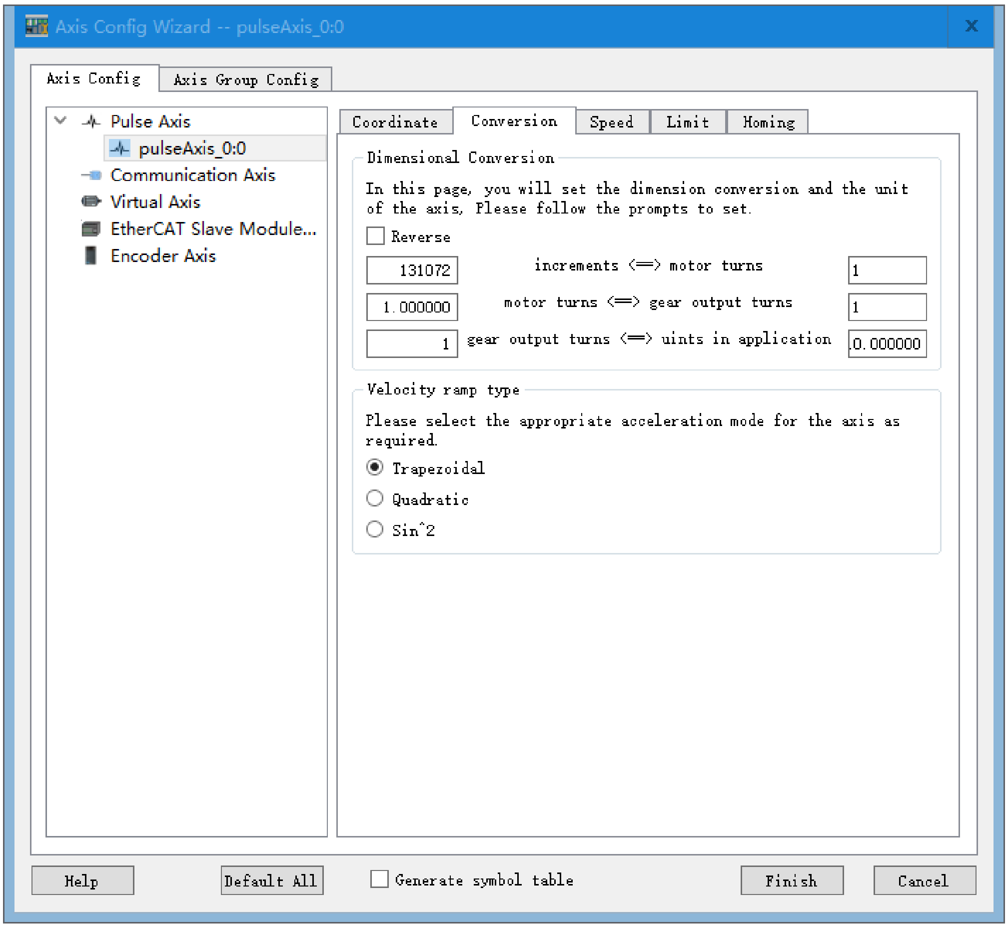
Slot number: corresponding to the slot number of the shaft on the rack.

Channel number: The channel number corresponding to this axis, ranging from 0 to 5.

Axis type: You can choose "linear axis" or "rotating axis". For reciprocating mechanisms of screw type, their stroke is limited, and it is necessary to know their absolute position within the range of screw stroke. In this case, it is better to choose "linear axis". If it is a unidirectional rotating shaft, using a linear mode can easily result in position counting overflow and calculation errors, so choosing a "rotating shaft" is better. All other options are default and not selectable.

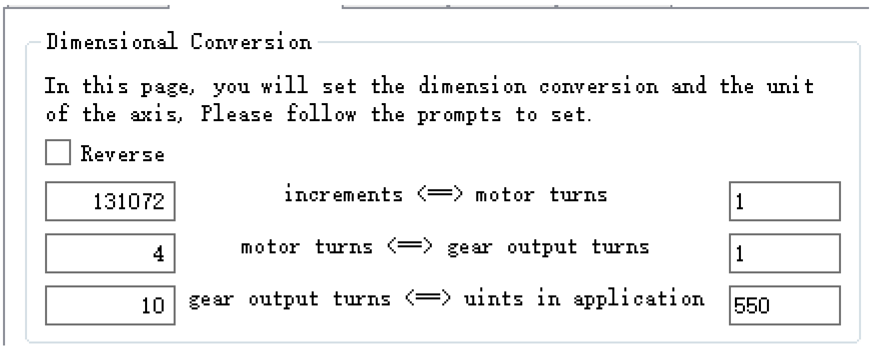
User coordinate source: Fixed optional "command output". The effective level of motor direction can be selected as "low level" or "high level".

【 Conversion 】

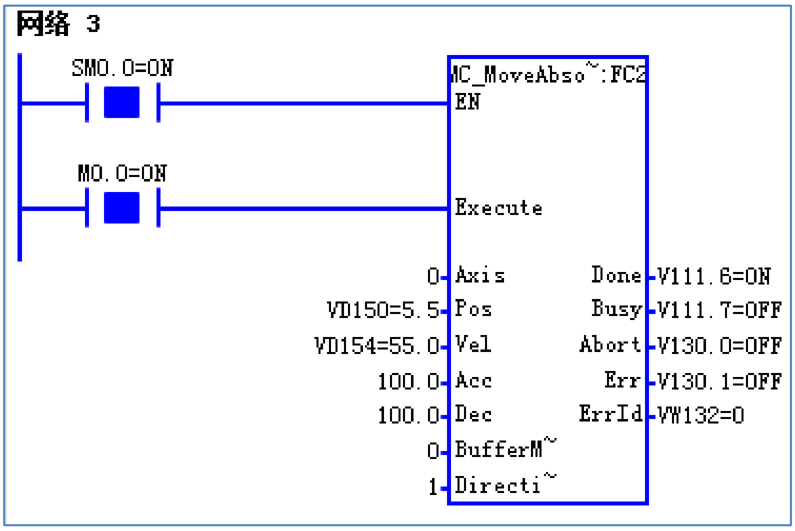


In the "Conversion" interface, you can set the type of program conversion and rate change according to the prompts. Set the "application unit" in the above figure to 360. When the program instruction requires the servo to run one unit, the gear end will select 1/360 turn (axis movement of 1 degree), the servo motor end will rotate 5 * 1/360 turn (motor rotation of 5 degrees), and so on. After setting the corresponding parameters (i.e. electronic gear ratio) according to the actual mechanical structure, distance commands can be input based on the physical unit of motion distance of the application system, making parameter control intuitive and easy to understand.

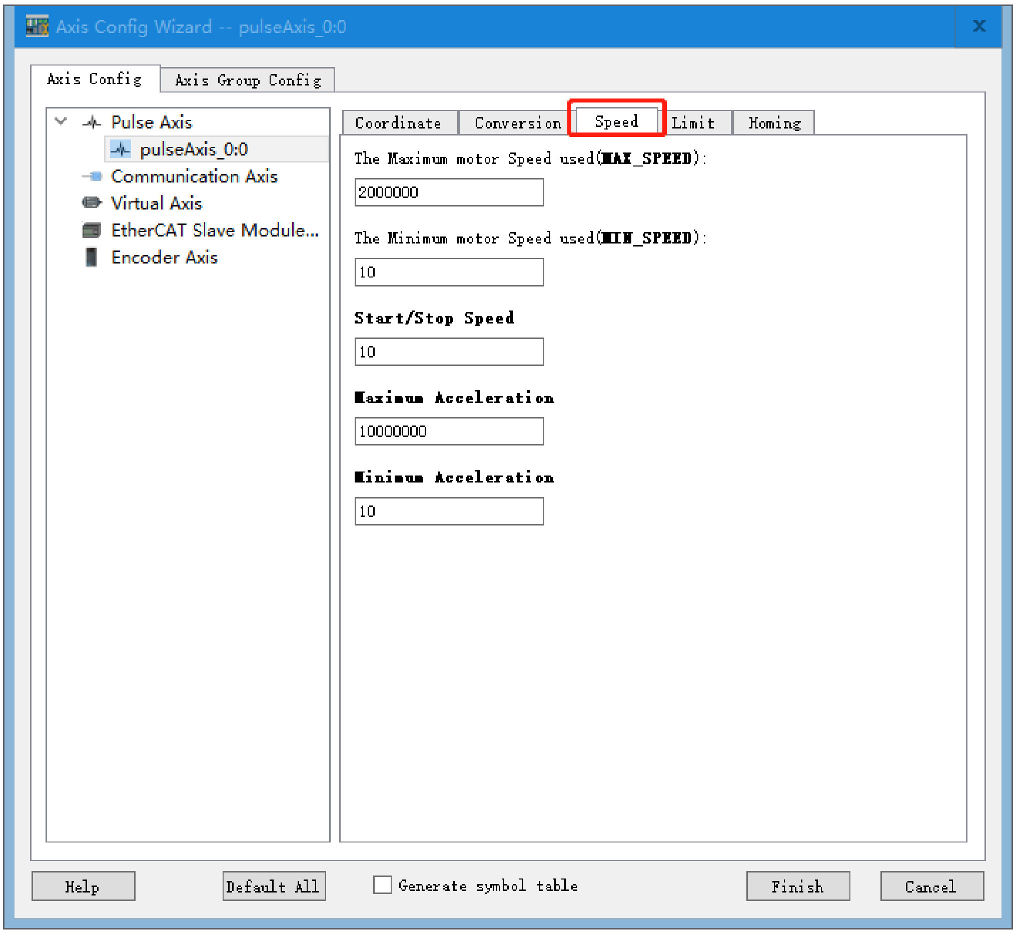
For example, after a 4:1 mechanical reduction by a reducer, the servo motor drives a lead screw with a lead of 5.5mm, which means that the lead screw rotates once and the lead screw slider moves 5.5mm. The dimensional conversion settings are as follows:



The set dimension can be used as the physical parameter dimension of the MC control command. Download the above settings into the program block and program to move the workpiece to the 5.5mm coordinate. The position naming unit is the equipment physical coordinate unit. The program example is as follows:



[Speed]

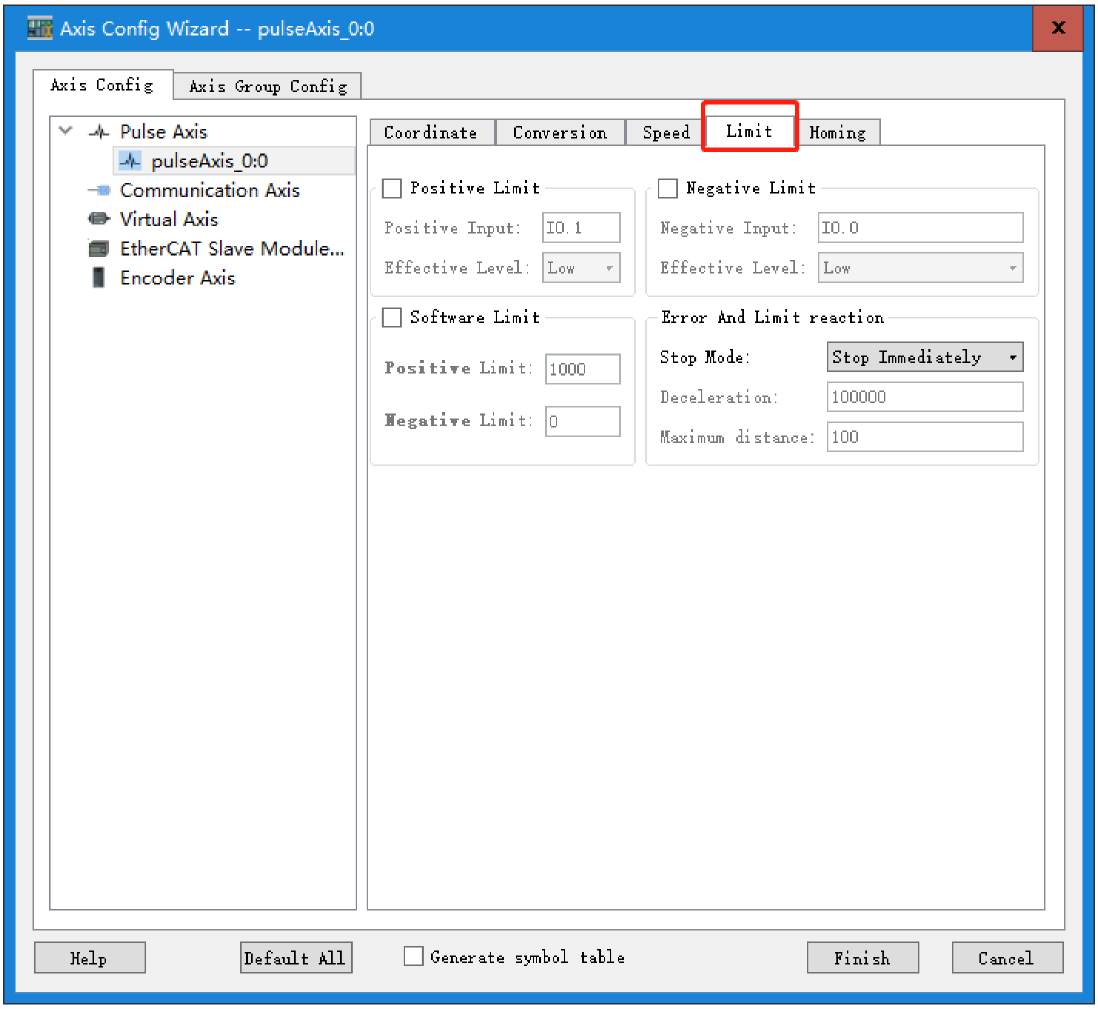


On this page, you can set the maximum/minimum speed, stop/start speed, and maximum and minimum acceleration/deceleration of the motor. During program operation, if the set speed exceeds the configuration setting, the library instruction will sound an alarm. The error code is as follows:

Error code	explain
5	The speed value is too small (less than the minimum speed)
6	The speed value is too high (greater than the maximum speed)
7	The acceleration value is too small
8	The acceleration value is too large

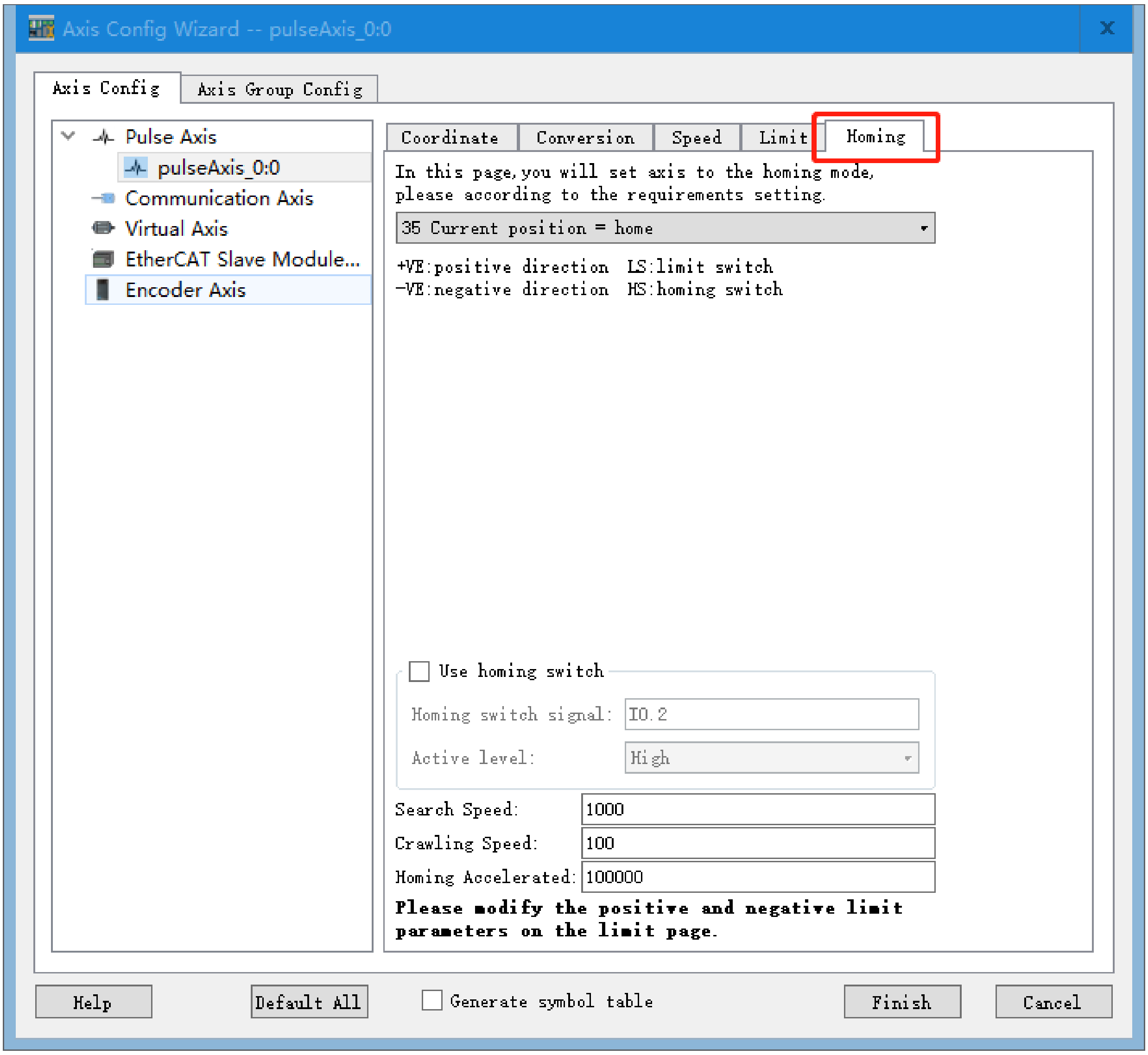
9	The deceleration value is too small
10	The deceleration value is too large

[**Limit**]



Checking the "positive limit" and "negative limit" can modify the positive/negative input settings, and the "effective level" can be selected as "high level" or "low level". Checking 'software limit' allows for modification and setting of positive/negative limit values. The "stop mode" for "error and limit response" can be selected as "immediate stop" or "deceleration stop".

【 **Homing** 】



On this page, the Homing mode of the axis can be configured. Only the pulse axis supports this configuration. Select the drop-down option to view the Homing mode. There are a total of 15 Homing modes to choose from.

6.1.1 Axis configuration

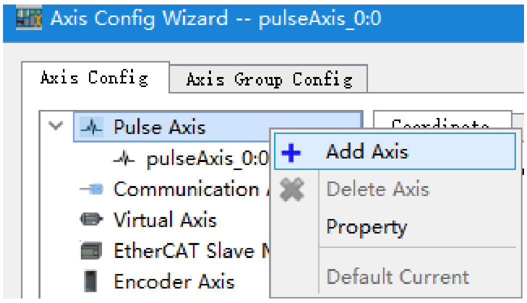
The upper limit of the total sum of all axes in the axis configuration is 64, supporting up to 6 local pulse axes and up to 16 EtherCAT axes. The number of EtherCAT axes=communication axis+EtherCAT slave axis. The virtual axis and encoder axis are calculated based on "64 pulse axis number EtherCAT axis".

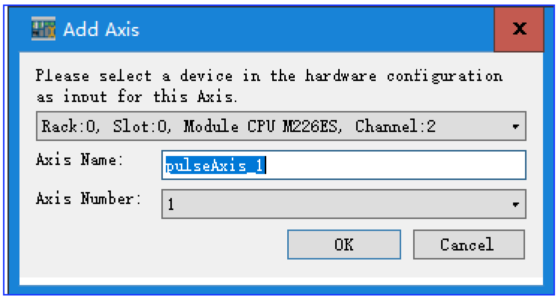
For example, if all 64 virtual axes are configured, or if 6 pulse axes are grouped, a maximum of 58 virtual axes can be supported. For example, if 4 pulse axes and 10 EtherCAT axes are configured, the remaining 50 axes can be all virtual axes or encoder axes.

In a single type of axis, the limit for pulse axes is 5-10, depending on the specific model's support. Supports up to 16 EtherCAT axes. The maximum number of virtual axes and encoder axes can reach 64.

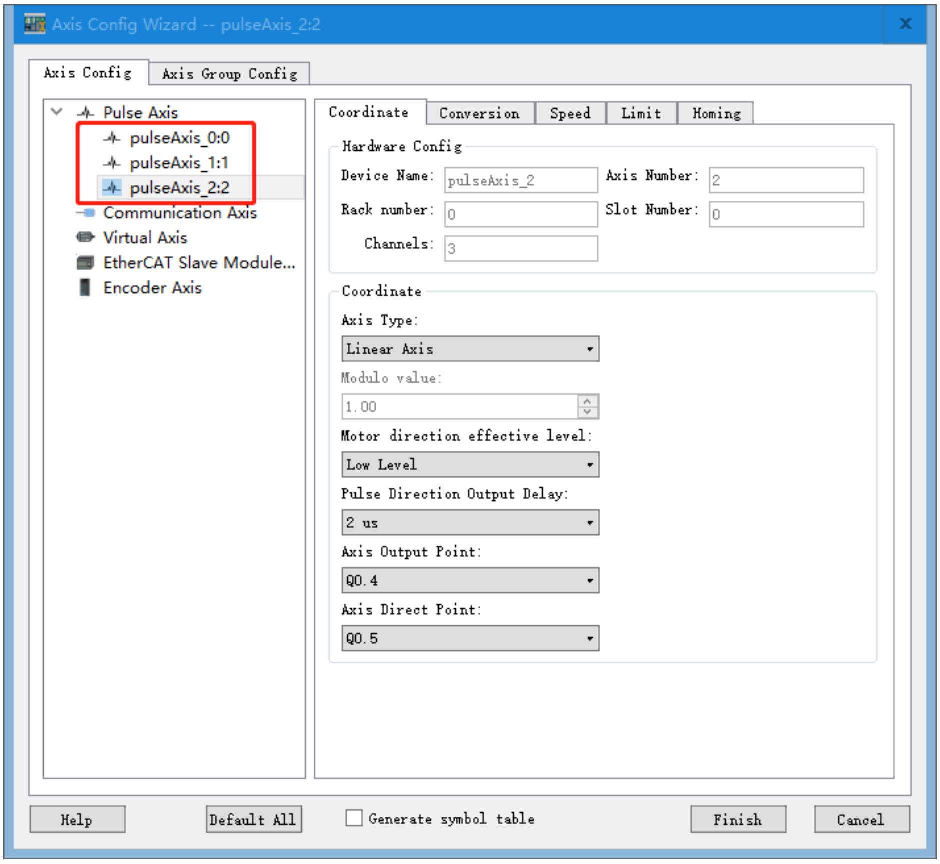
1. Pulse axis configuration

Select "Axis Wizard Configuration" in the "Tools" option, right-click on the pulse axis in the Axis Wizard Configuration interface to select "Add Axis", right-click on the added axis to select "Delete Axis". The axis name can be freely changed. Click "OK" to successfully add the axis, as shown below:

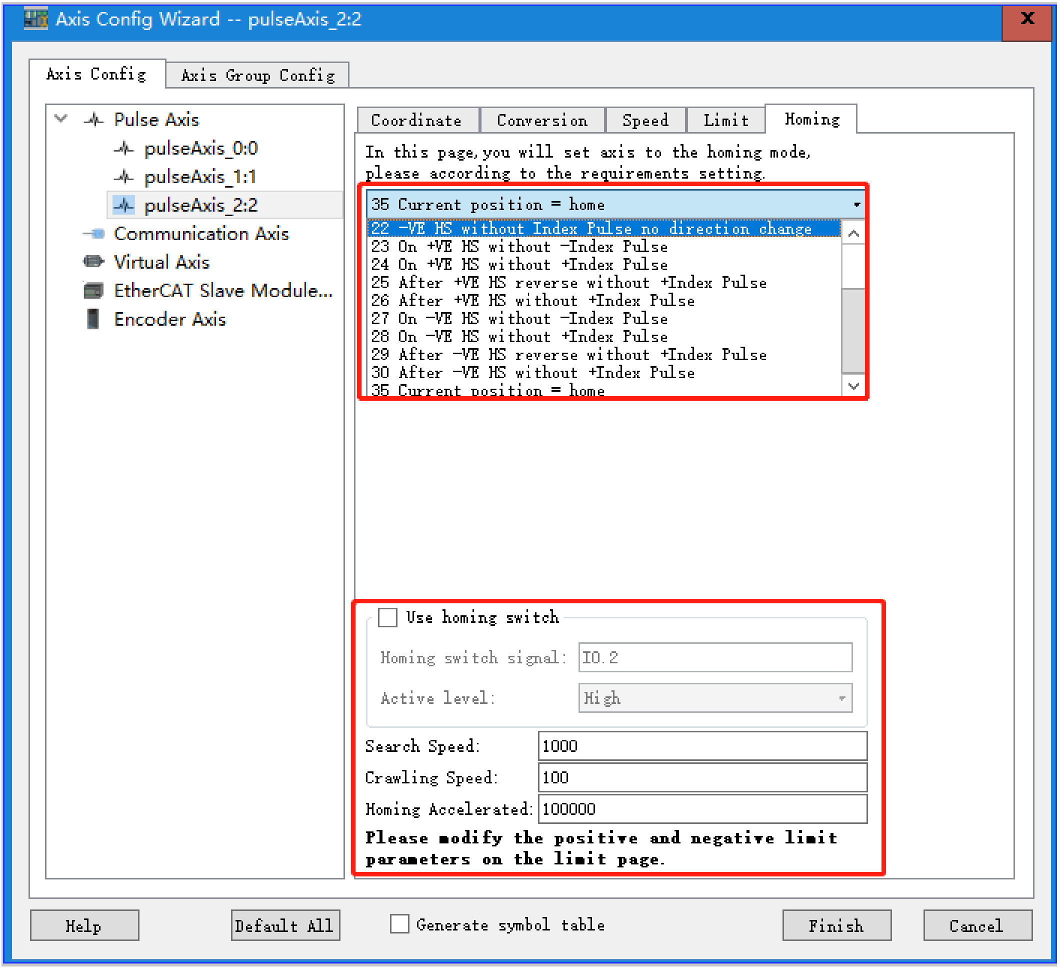




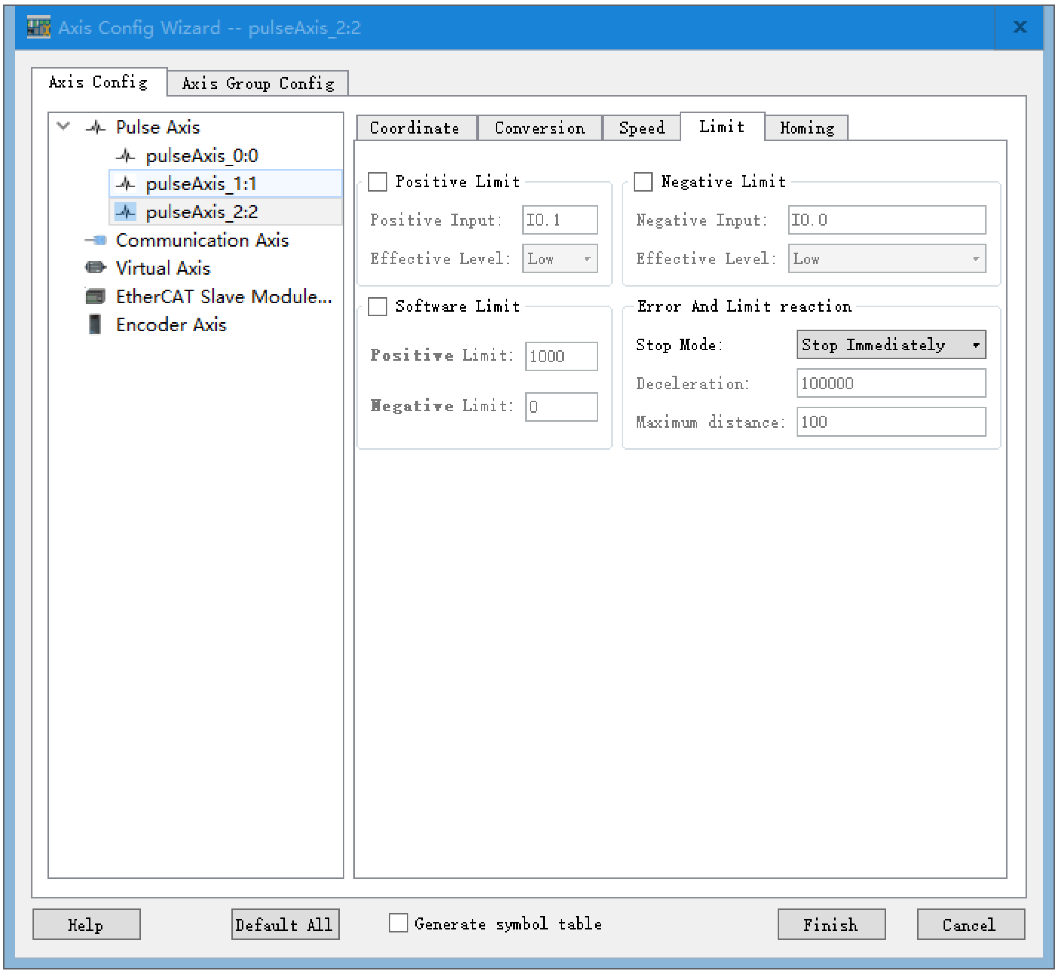
The added pulse axis interface is shown below:



If you need to use the Homing function, click on "Homing", select the Homing mode (there are a total of 17 Homing modes), check the "Use Homing Switch", set the Homing parameters, Homing effective level, search speed, crawling speed, and Homing acceleration.



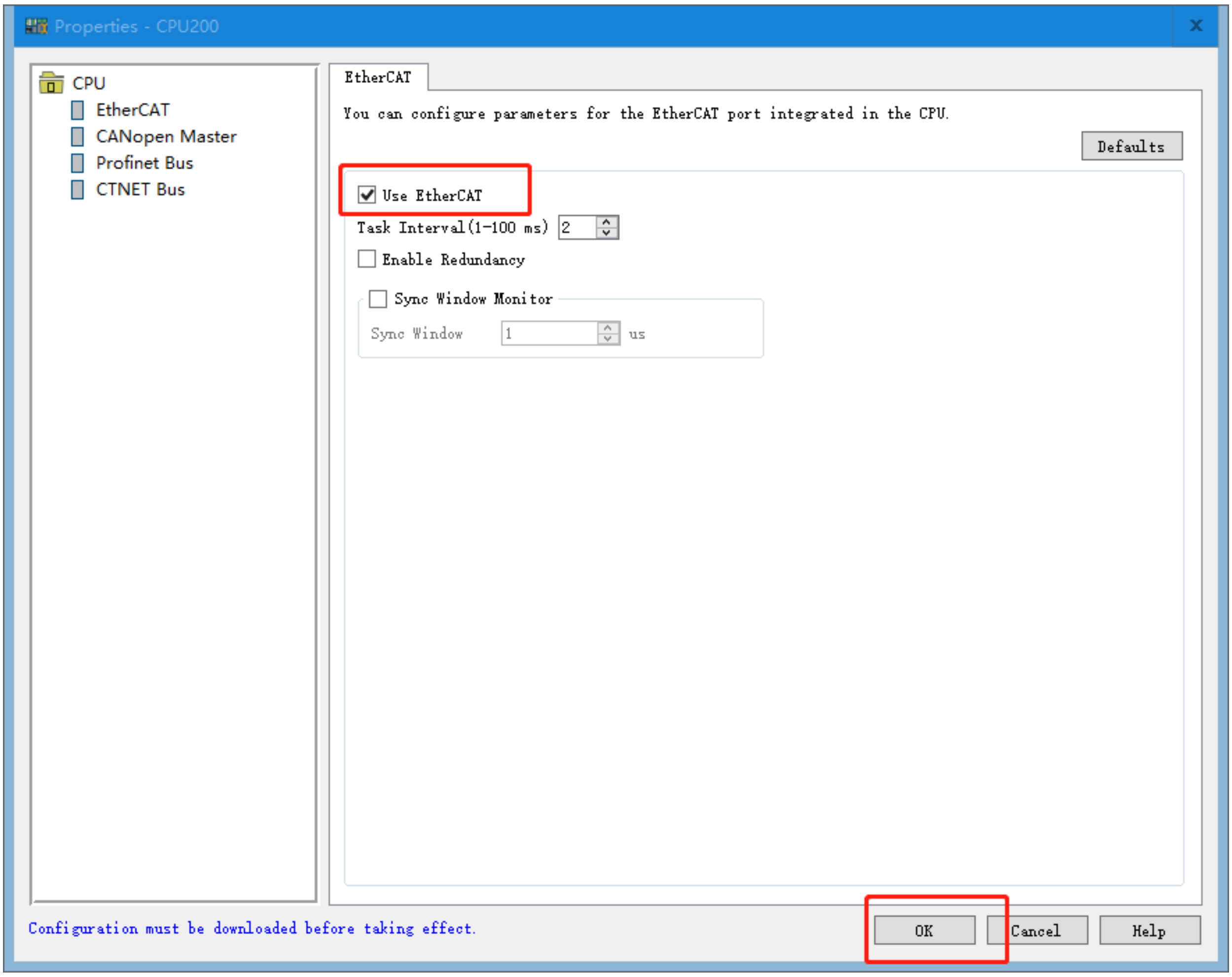
Set limit switches on the restriction page



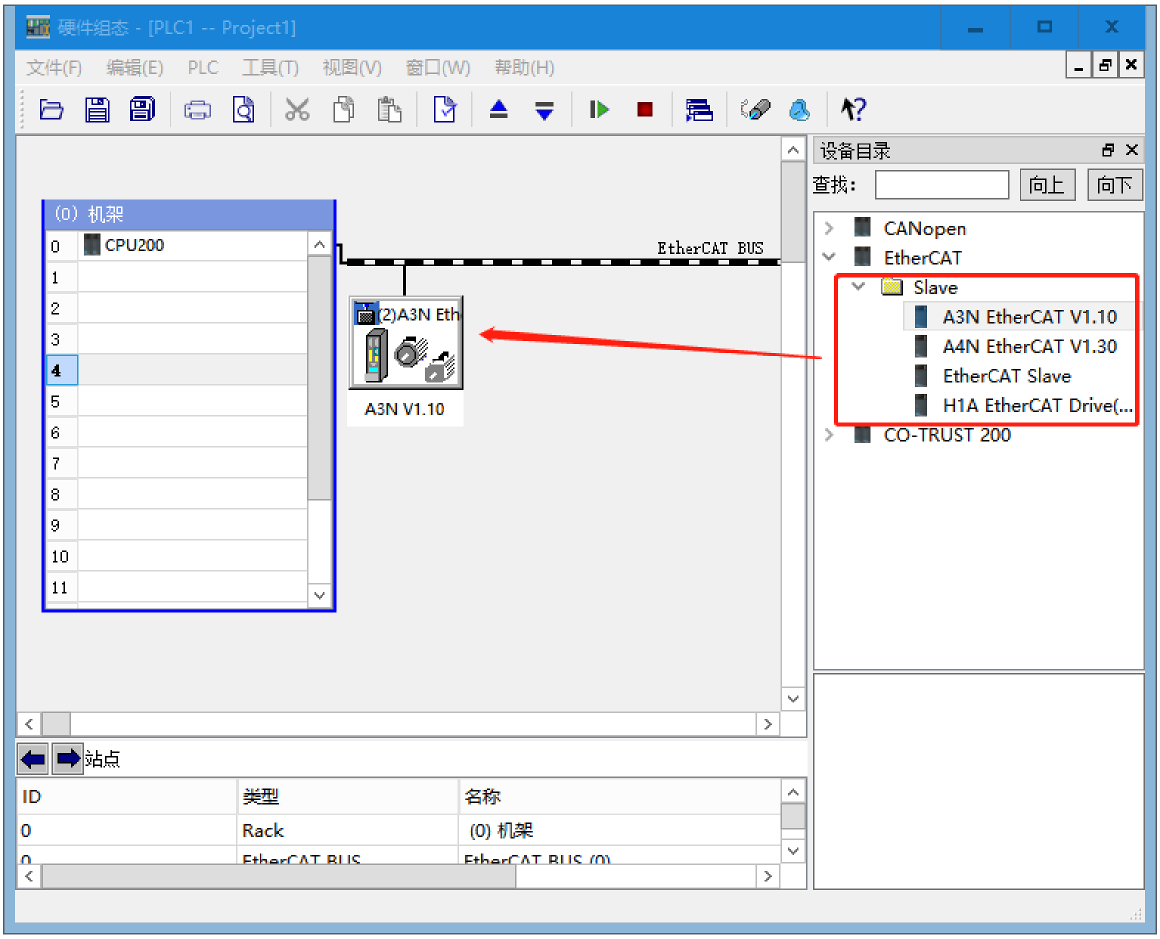
2. Communication axis configuration

Before adding a communication axis, it is necessary to add the corresponding EtherCAT slave in the hardware configuration interface, otherwise the communication axis cannot be added. The specific operation is as follows:

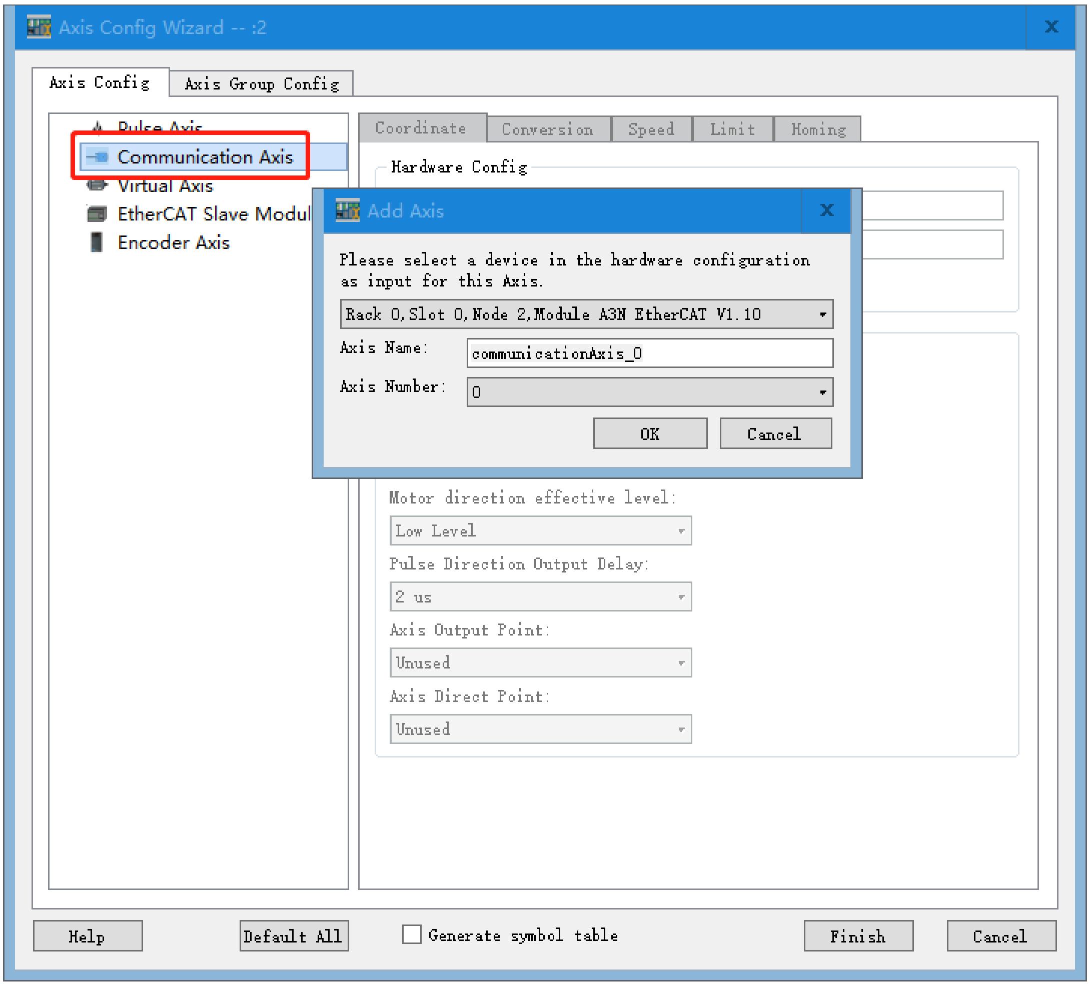
1) Right click on the CTH200 on the rack, select "Properties" → "EtherCAT", and click on " Use EtherCAT Master " on the page.



2) After confirmation, you can see the "EtherCAT BUS" on the hardware configuration interface. Click on "EtherCAT" on the right, select the slave station, and drag it to "EtherCAT BUS".

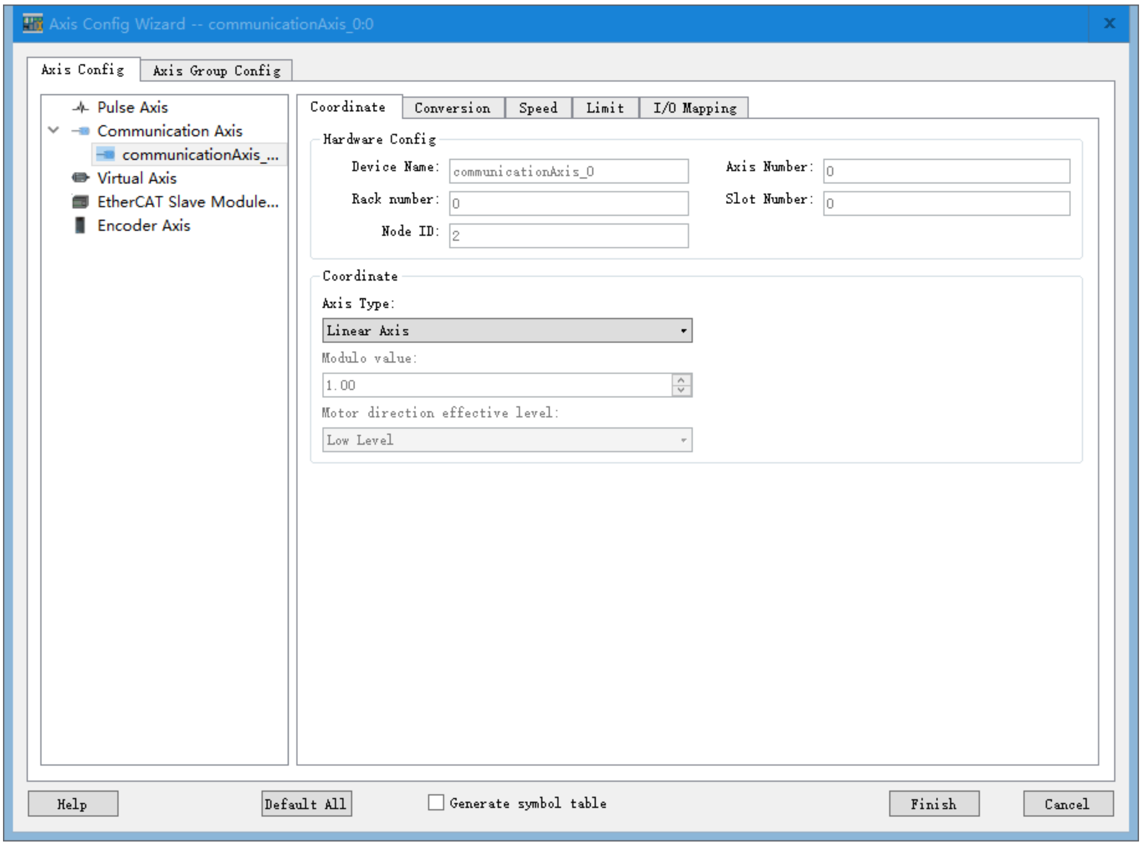


3) After completing the above configuration, select "Axis Wizard Configuration" in the "Tools" option, double-click "Communication Axis" in the Axis Wizard Configuration interface to add a communication axis. In the pop-up "Add Axis" interface, you can select the input device for the axis, and the optional device is the slave station added in the hardware configuration. The "axis name" can be modified, click "OK" to complete the addition.



Note: Only one communication axis can be added to a slave station.

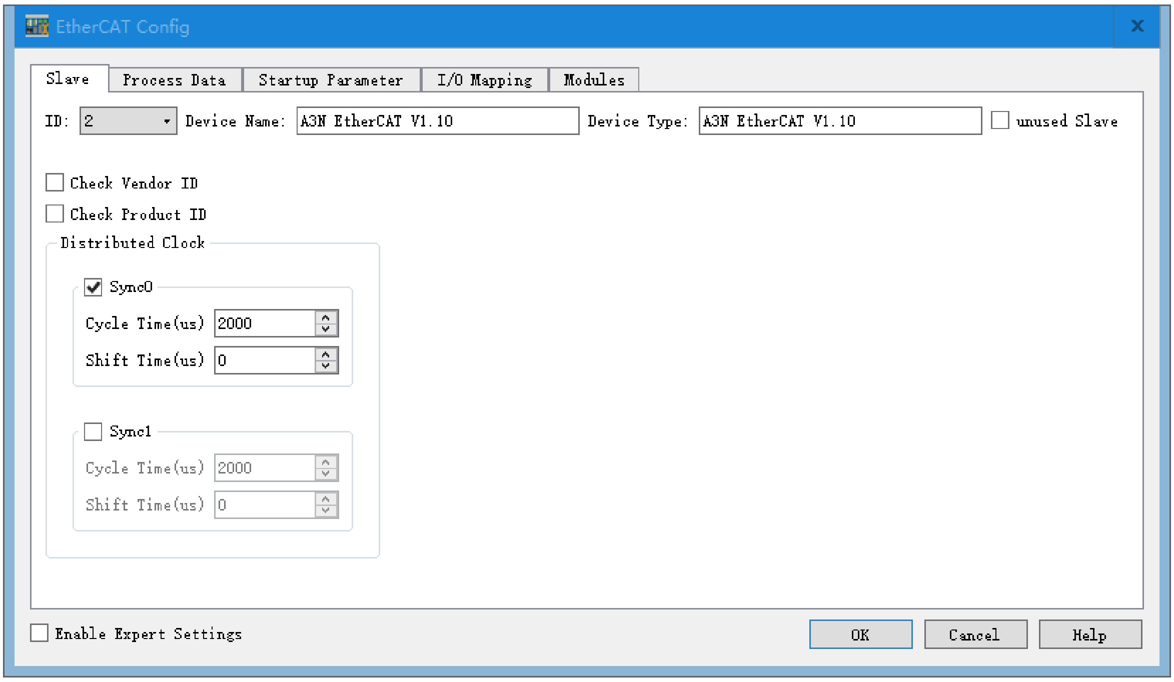
4) The added communication axis interface is shown below:



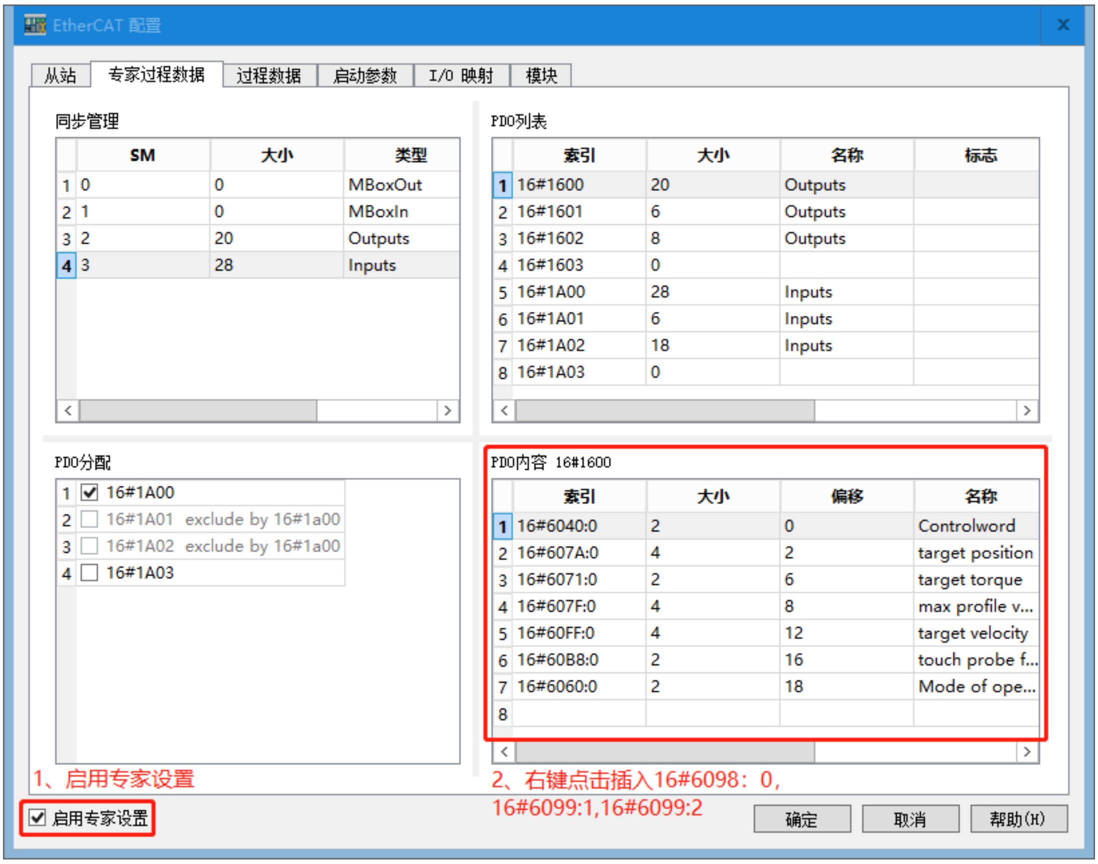
5) If it is necessary to use the driver return to the original (MC_Some command), when configuring the

communication axis return to the original function, it is necessary to configure the return to the original mode 16 # 6098:0, return to the original search speed 16 # 6099:1, and return to the original crawling speed 16 # 6099:2. The operation is as follows:

Double click the slave station attached to "EtherCAT BUS". The EtherCAT configuration interface appears.

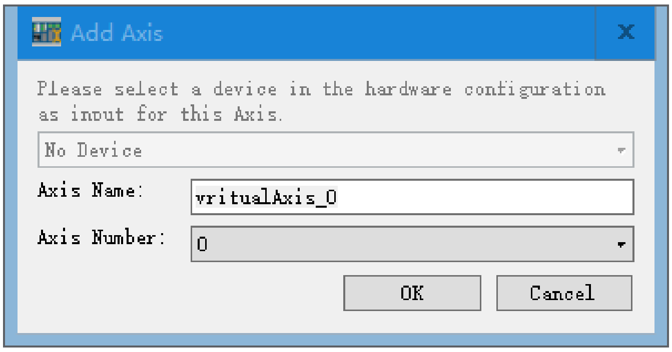


6) Click on 'Enable Expert Settings', right-click and insert parameters 16 # 6098:0 (Return to Origin Mode), 16 # 6099:1 (Return to Origin Search Speed), and 16 # 6099:2 (Return to Origin Crawl Speed).

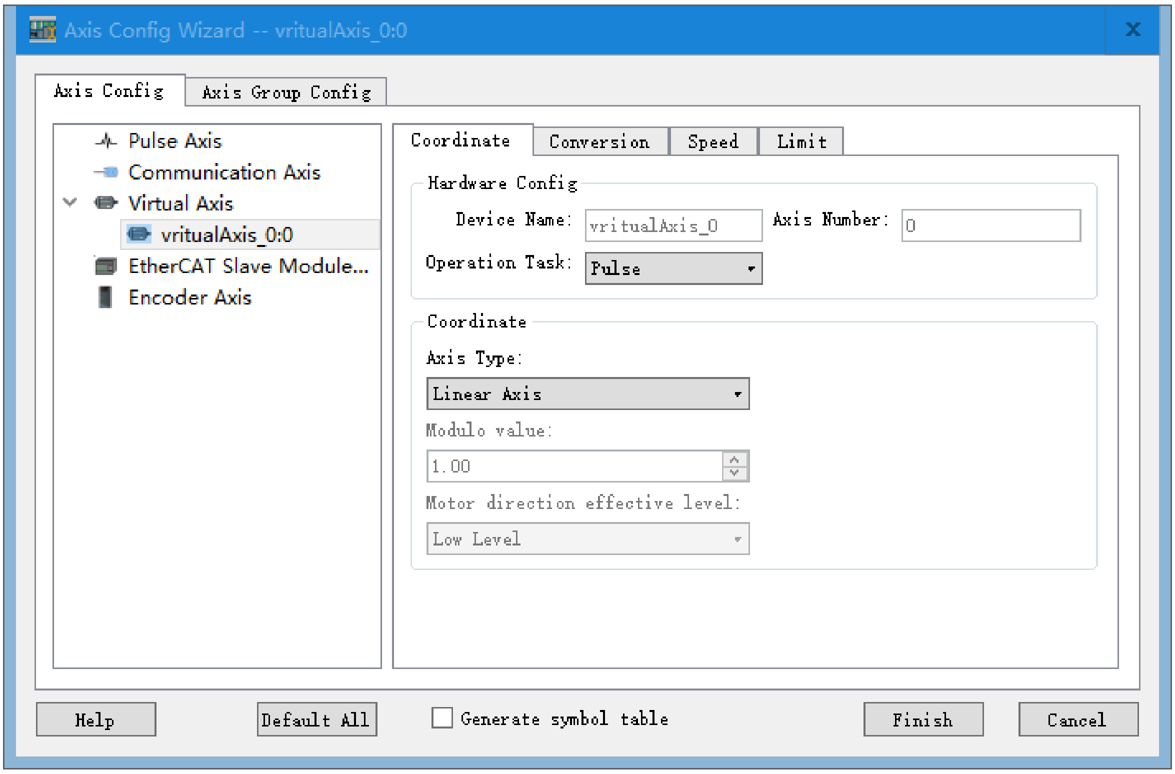


3. Virtual axis configuration

In the Axis Configuration Wizard interface, right-click on "Virtual Axis" and select "Add Axis". In the pop-up interface below, you can choose to modify the axis name and click "OK" to complete the virtual axis addition.

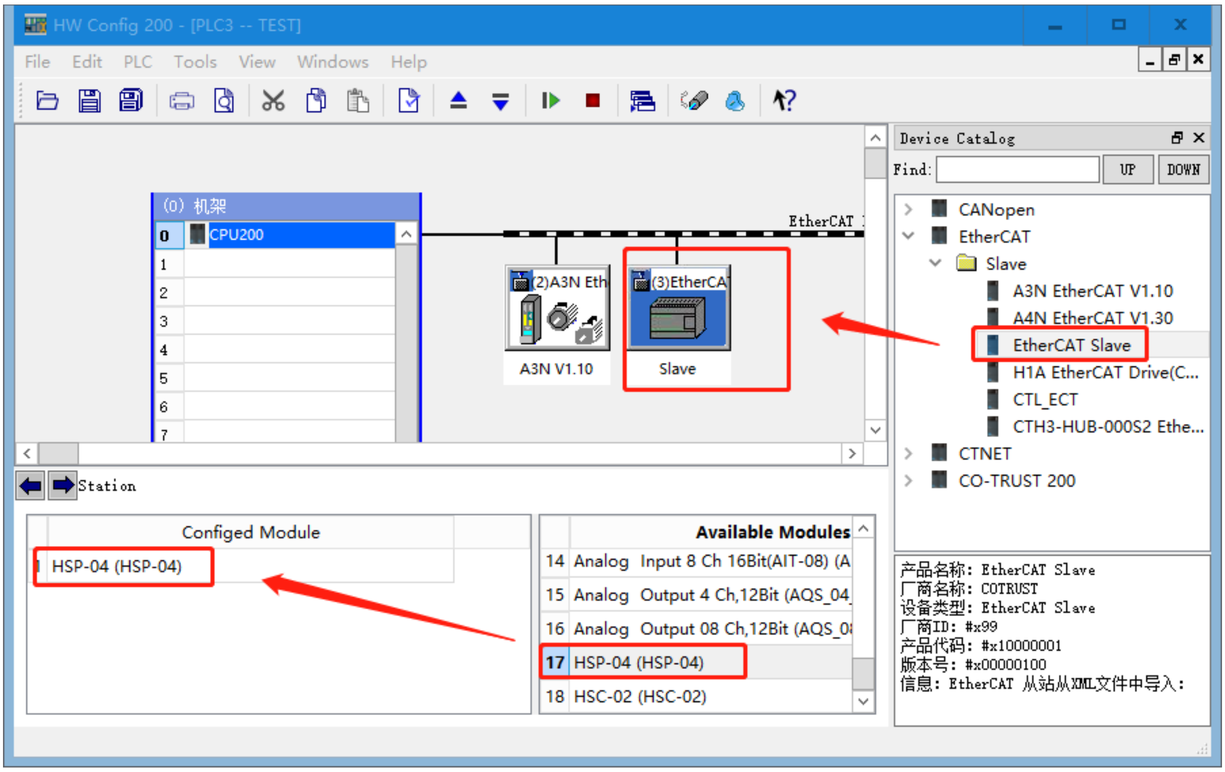


The added virtual axis interface is shown below:

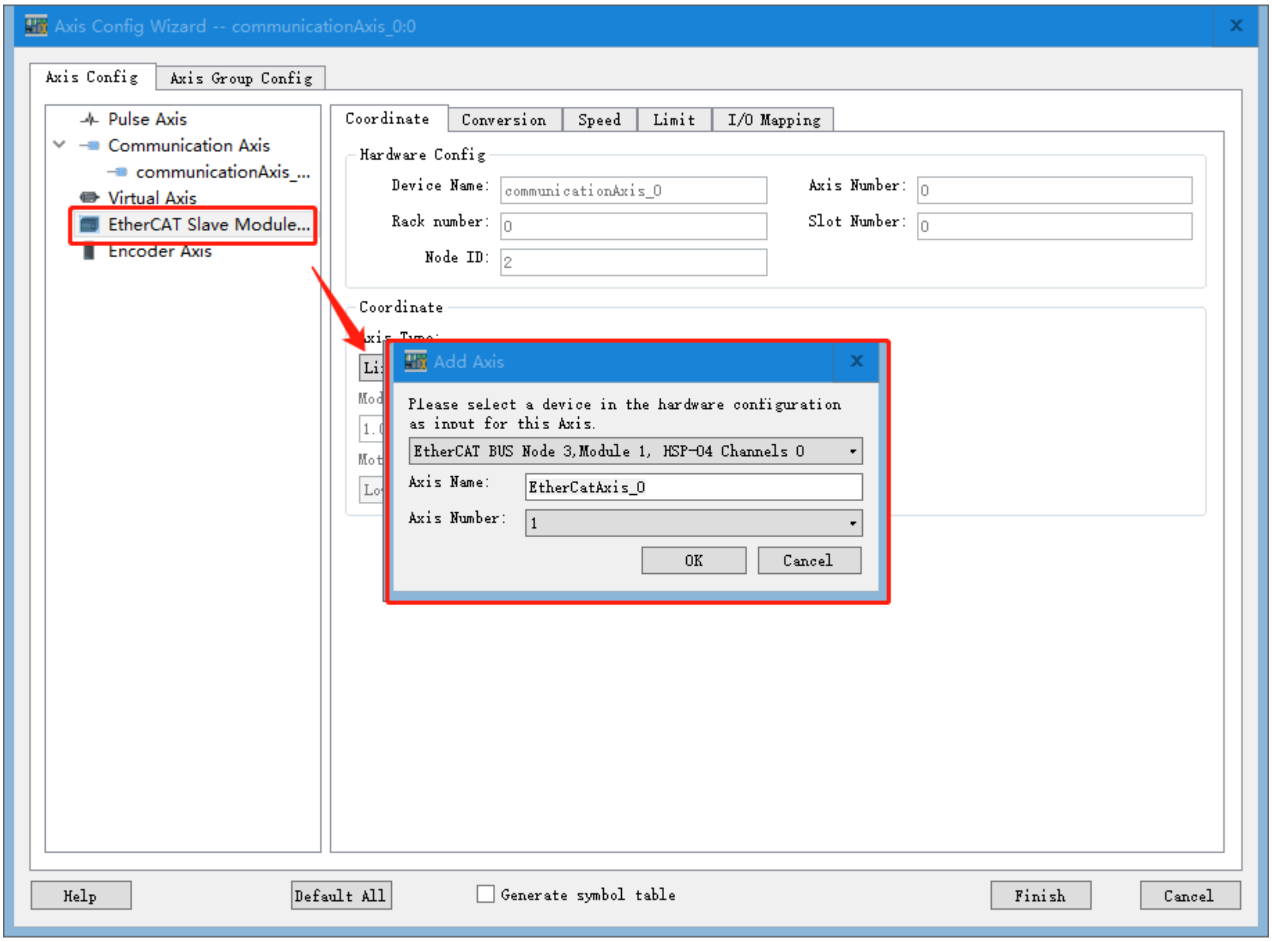


4. EtherCAT Slave Module Axis Configuration

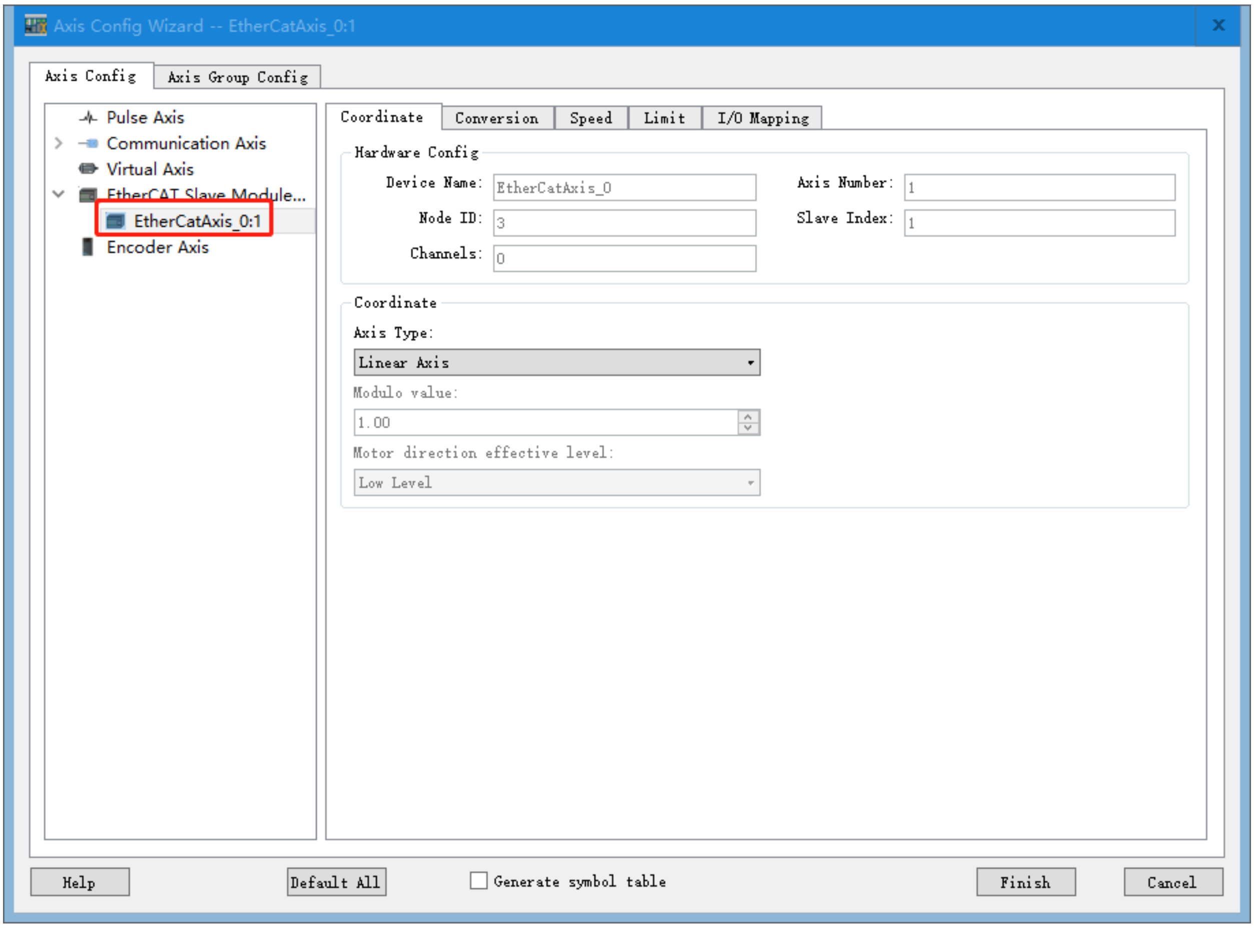
Before adding the EtherCAT slave module axis, it is necessary to first add the EtherCAT slave in the hardware configuration interface and select the HSP module for the added EtherCAT slave. As shown in the following figure:



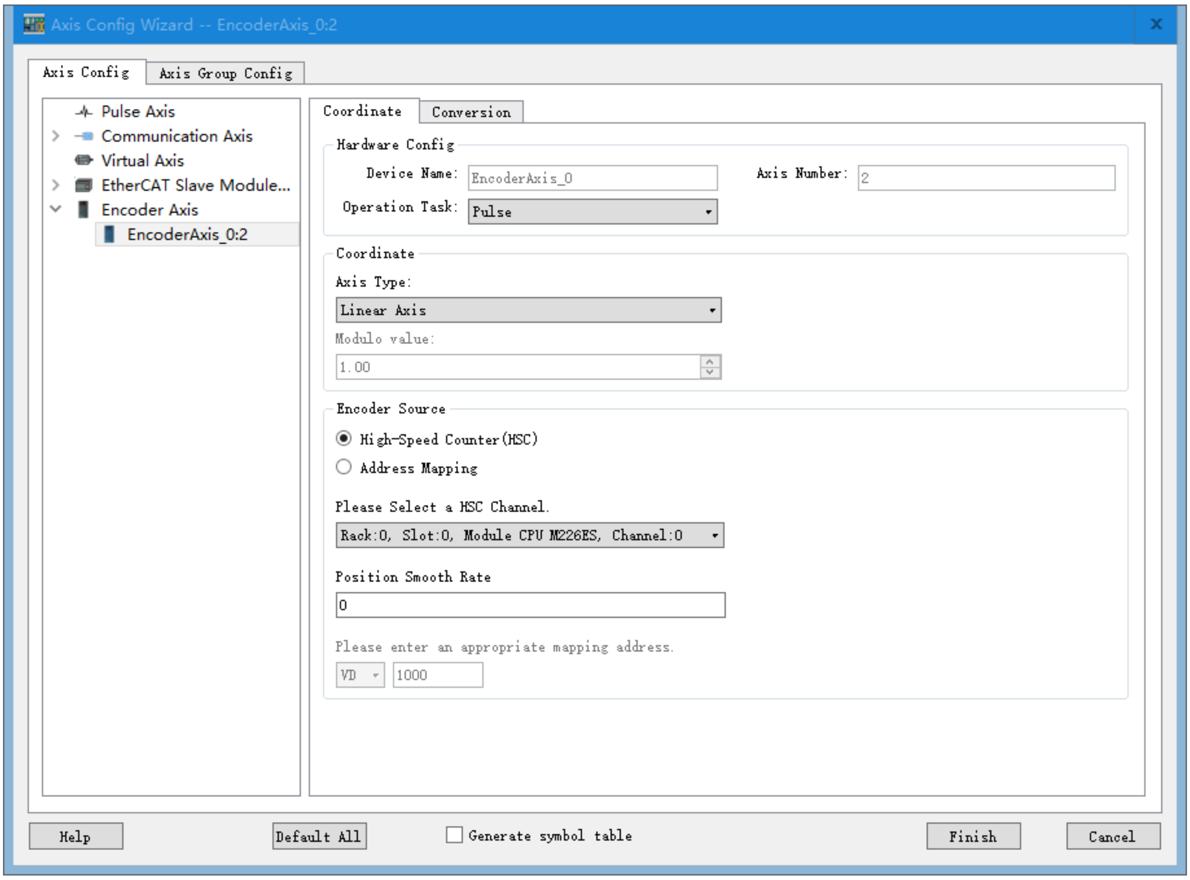
Right click on "EtherCAT Axis" in the "Axis Configuration Wizard" interface and select "Add Axis". The following interface will pop up for axis configuration. After configuration, click "OK" to complete the addition.



The interface of the added EtherCAT slave module axis is shown below:



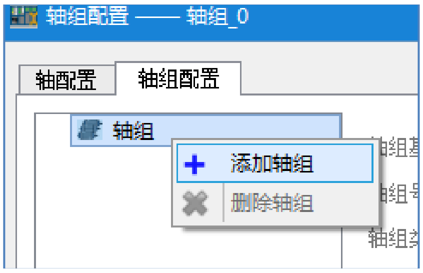
5. Encoder shaft configuration



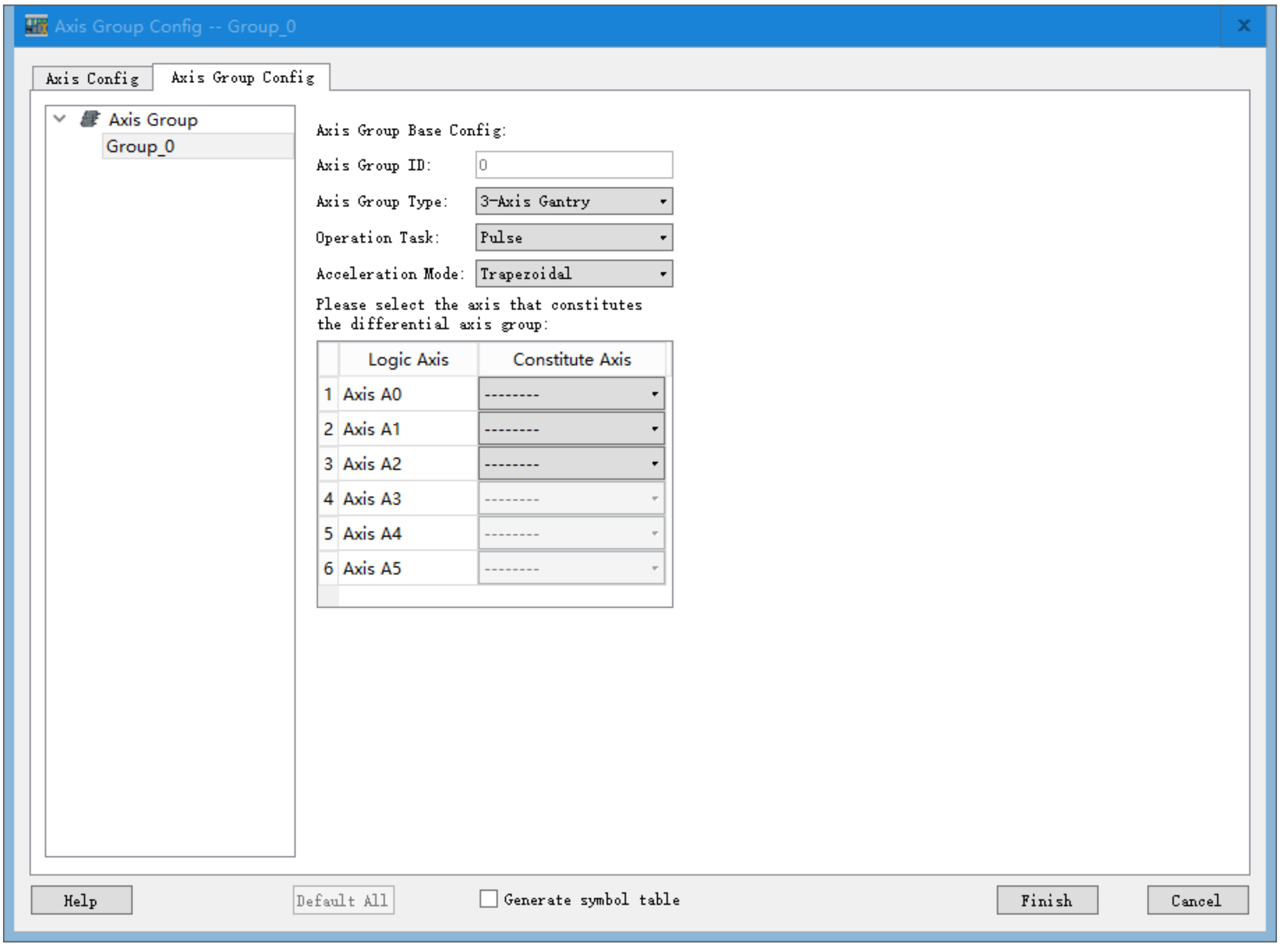
Select "Encoder Axis" on the above interface and right-click to add a new encoder axis. Select the corresponding encoder for the axis under the "Encoder Source" option, and choose to enter the corresponding mapping address (default V zone) to determine the encoder. The position smoothing coefficient is used to filter the encoder shaft. The smaller the smoothing filtering parameter setting, the smaller the speed fluctuation. Setting it to 1 or 0 means no filtering, and between 0 and 1 means filtering.

6.1.2 Axis group configuration

Axis group configuration: Two or three axes that are about to be configured can be freely combined in the form of axis groups. In the "Axis Group Configuration" interface, right-click on "Add Axis Group" to add axis groups, as shown in the following figure:



The interface for successfully adding axis groups is shown below:

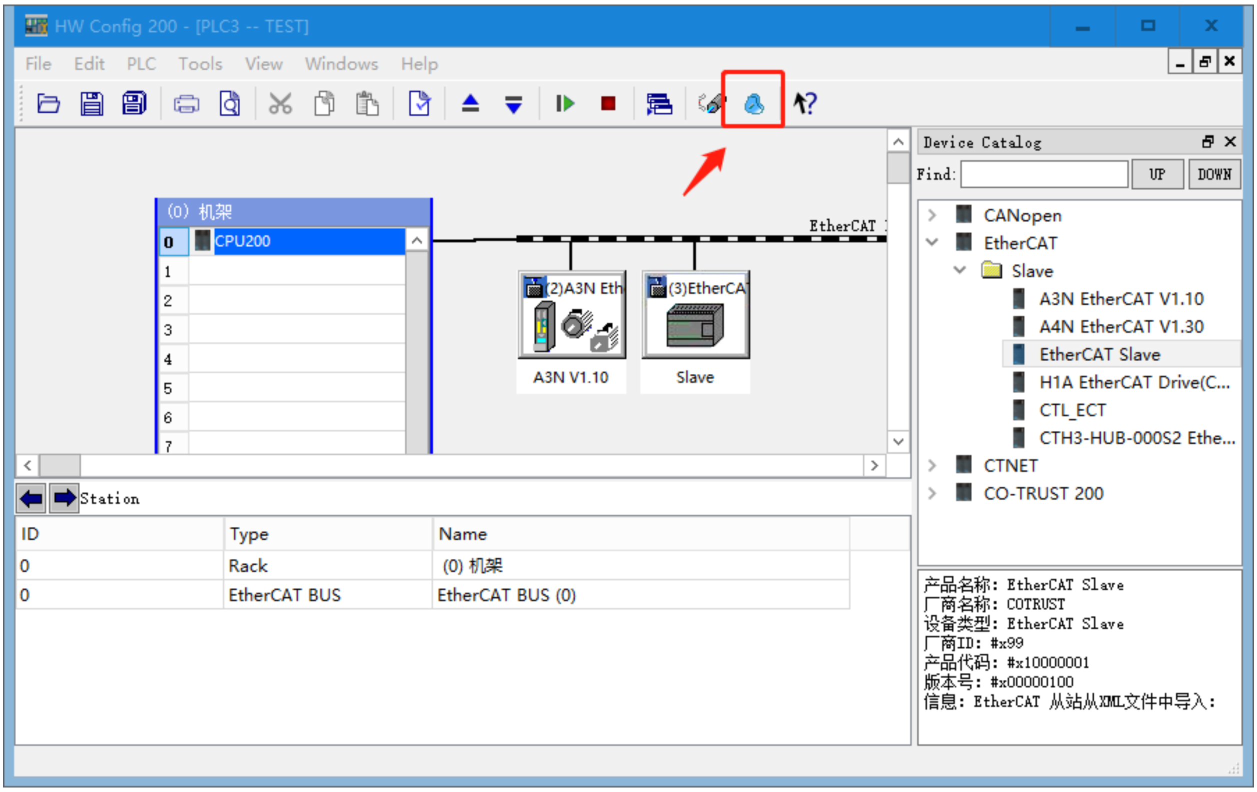


- Basic configuration of axis group ":" Axis group number "cannot be changed. The drop-down options for" Axis group type "can be selected as" 2-axis orthogonal "or" 3-axis orthogonal ", corresponding to two or three groups of" constituting axes "that can be selected respectively. The 'operation cycle' can be selected as' pulse '. The "acceleration mode" can be selected from "trapezoidal", "sin * 2", and "quadratic".
- Logical axis: default six groups, cannot be changed.
- Composition axis ":" 2-axis orthogonal "corresponds to two composition axes," 3-axis orthogonal "corresponds to three composition axes, and the drop-down options can only select" pulse axis "or" virtual axis ".

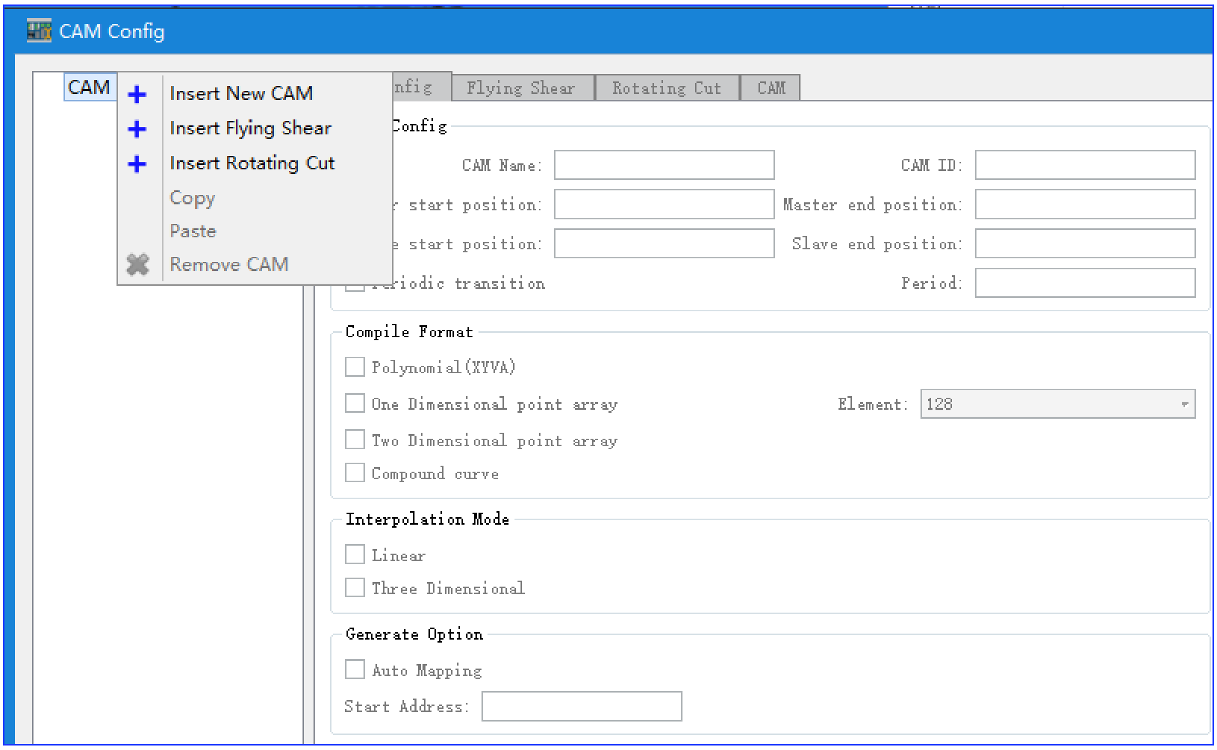
6.2 Electronic cam guide

In the electronic cam editor, electronic cam discs (or cam switch functions) can be implemented through graphics or lists. When generating code based on relevant applications, various global data structures (CAM data) will be created, which can be accessed by IEC programs.

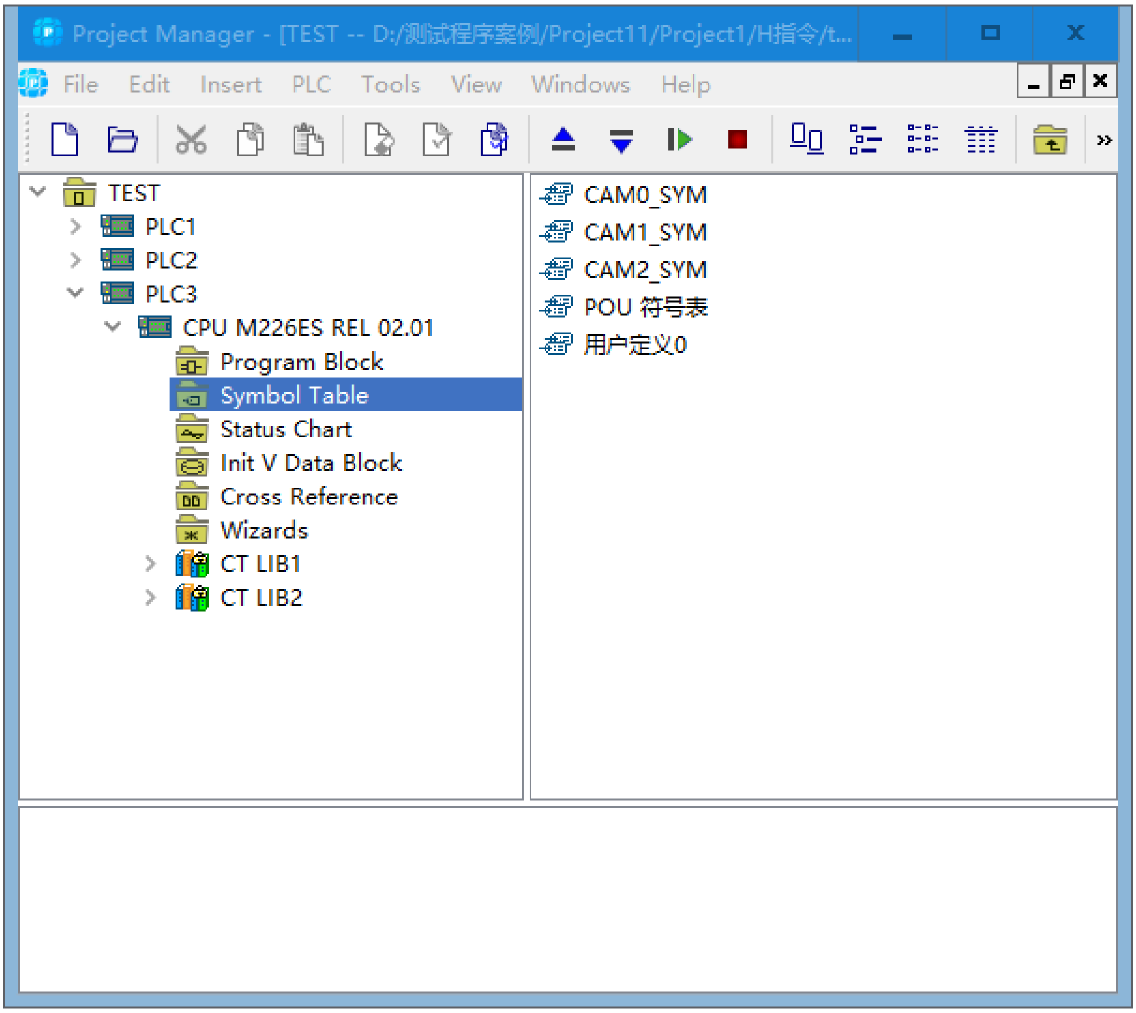
1. Select "Tools" and then "Electronic Cam Wizard" on the hardware configuration interface to configure the electronic cam.



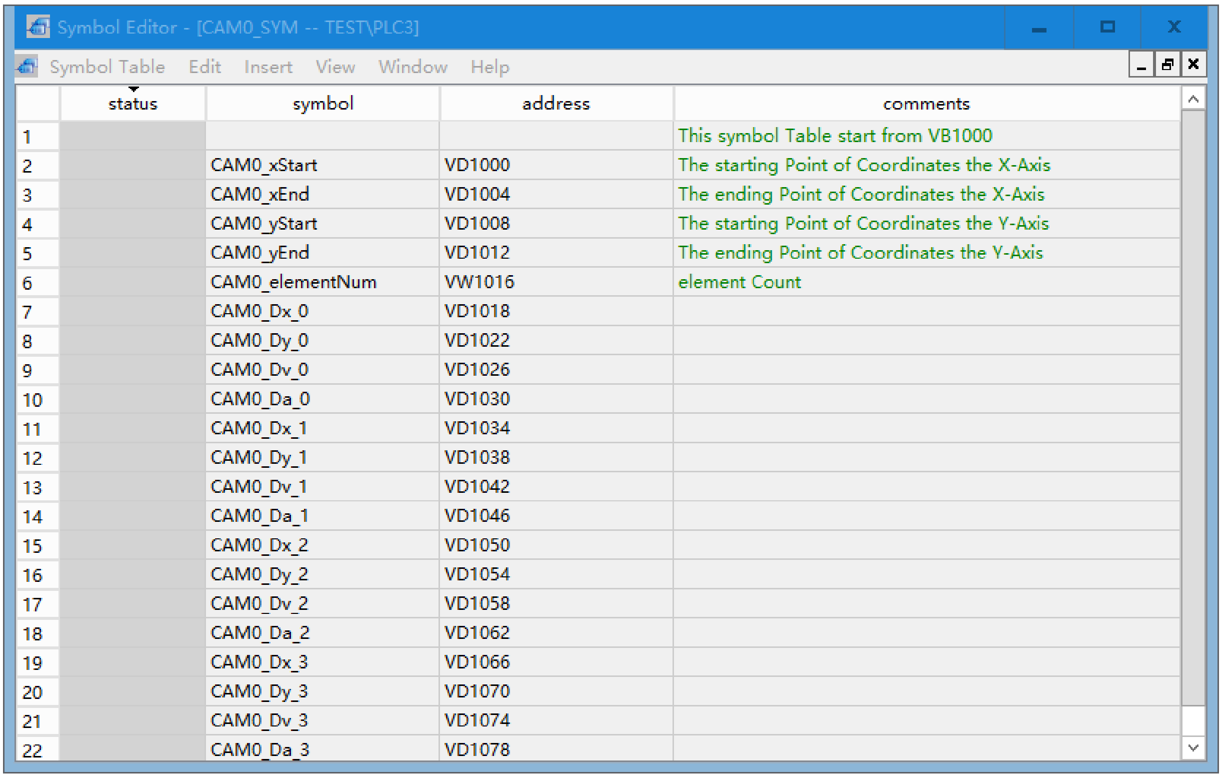
2. The configuration interface of the electronic cam guide is as follows. Select "CAM" and right-click to insert a new CAM, follow-up shear, or flying shear:



After successfully inserting CAM, follow-up cutting, and flying cutting, select "Symbol Table" in the project manager interface to see the newly added CAM.



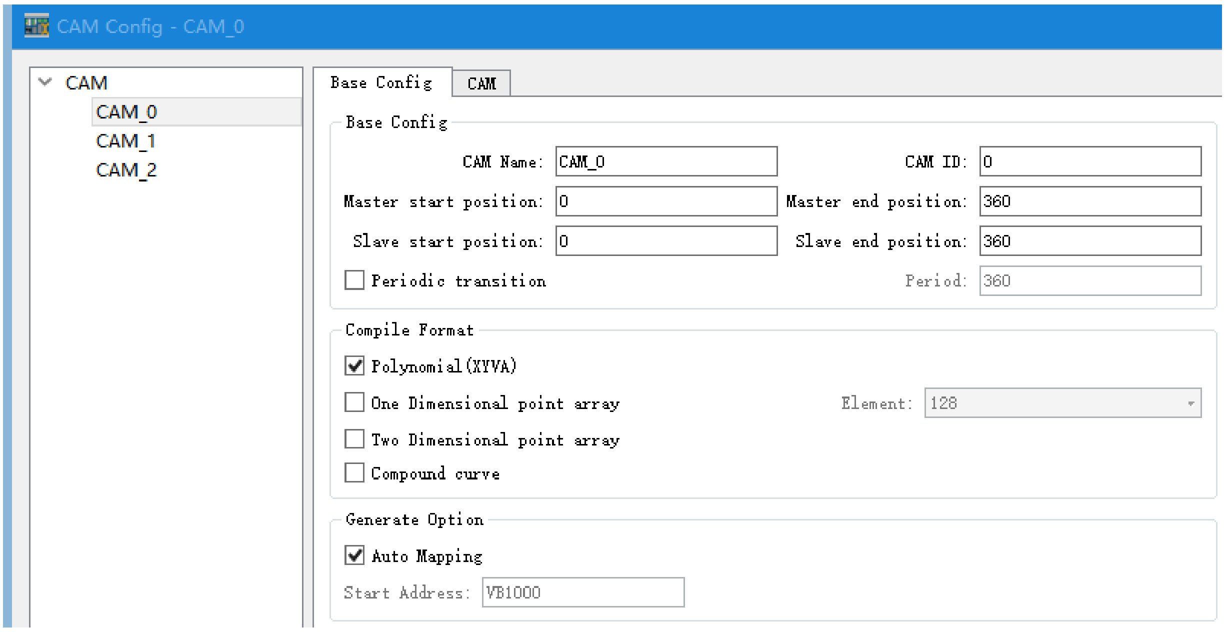
3. Click on "CAM" to display the following interface, where you can view the corresponding CAM symbol address.


Symbol Editor - [CAM0_SYM -- TEST\PLC3]

Symbol Table Edit Insert View Window Help

	status	symbol	address	comments
1				This symbol Table start from VB1000
2		CAM0_xStart	VD1000	The starting Point of Coordinates the X-Axis
3		CAM0_xEnd	VD1004	The ending Point of Coordinates the X-Axis
4		CAM0_yStart	VD1008	The starting Point of Coordinates the Y-Axis
5		CAM0_yEnd	VD1012	The ending Point of Coordinates the Y-Axis
6		CAM0_elementNum	VW1016	element Count
7		CAM0_Dx_0	VD1018	
8		CAM0_Dy_0	VD1022	
9		CAM0_Dv_0	VD1026	
10		CAM0_Da_0	VD1030	
11		CAM0_Dx_1	VD1034	
12		CAM0_Dy_1	VD1038	
13		CAM0_Dv_1	VD1042	
14		CAM0_Da_1	VD1046	
15		CAM0_Dx_2	VD1050	
16		CAM0_Dy_2	VD1054	
17		CAM0_Dv_2	VD1058	
18		CAM0_Da_2	VD1062	
19		CAM0_Dx_3	VD1066	
20		CAM0_Dy_3	VD1070	
21		CAM0_Dv_3	VD1074	
22		CAM0_Da_3	VD1078	

4. After inserting CAM, the configuration interface is shown below.



【 Basic Configuration 】

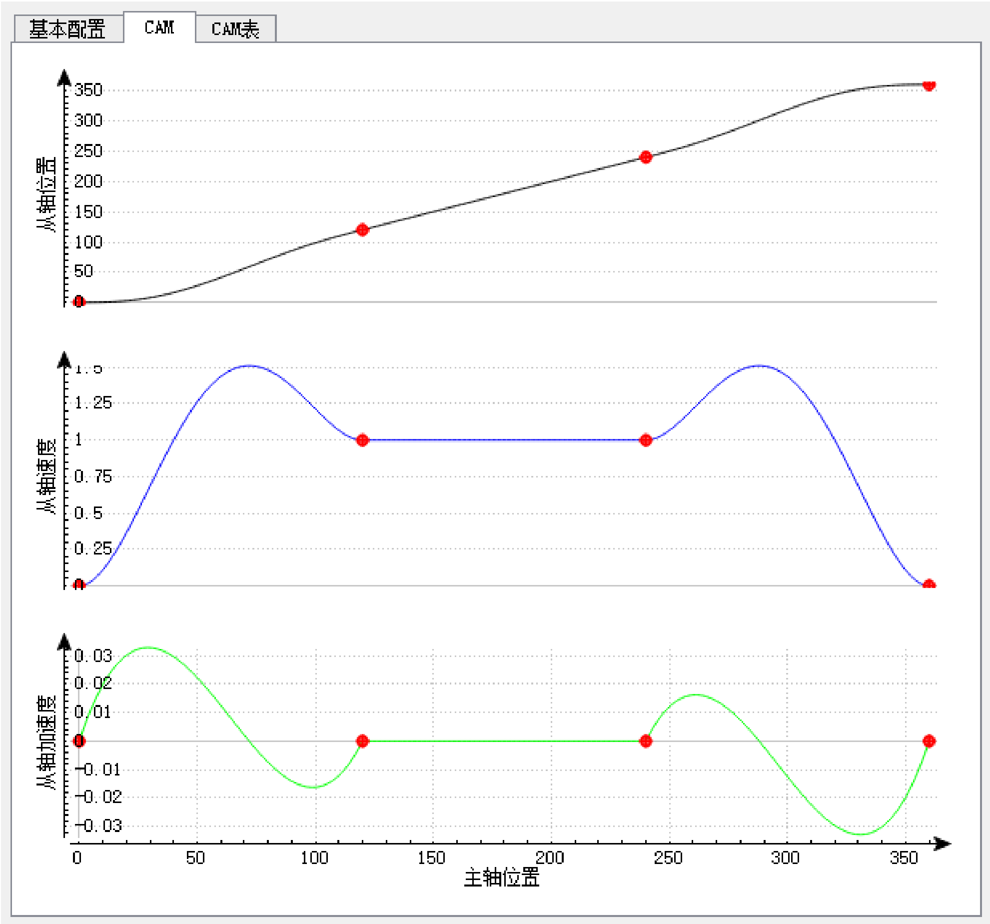
Basic Configuration: Users can manually change the "CAM Name" and the starting and ending positions of the master and slave axes (the spindle position corresponds to the horizontal axis in the CAM curve graph, and the slave axis position corresponds to the vertical axis in the CAM curve graph), check "Periodic Change", and the CAM curve graph will show periodic change, with the spindle end position being the cycle.

● Compilation type ": Check" Polynomial (XYVA) ", CAM will display the horizontal and vertical coordinates, velocity and acceleration states in the state table with the symbols x, y, a, v. Currently, the compilation types of" one-dimensional array "and" two-dimensional array "are not supported.

● Generate Options ": Check" Auto Mapping "to set the default starting address to VB1000. Uncheck to modify the starting address, which corresponds to the starting address in the symbol table.

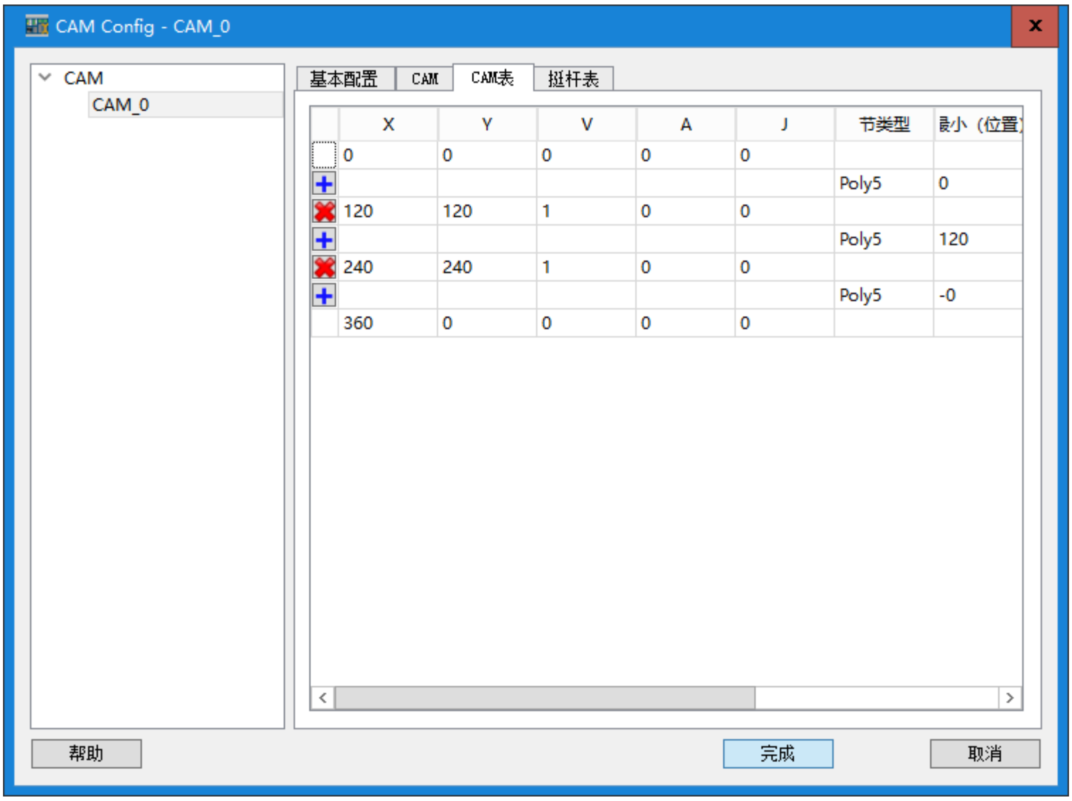
[CAM]

This page displays the relationship between "main axis position" and "sub axis position", "sub axis velocity", and "sub axis acceleration" in the form of a curve graph. The maximum value of the "main axis position" on the horizontal axis corresponds to the setting value of the "main axis end position" on the "basic configuration" interface, and the maximum value of the "sub axis position" on the vertical axis corresponds to the setting value of the "sub axis end position" on the "basic configuration" interface. Click on the red node to drag it.

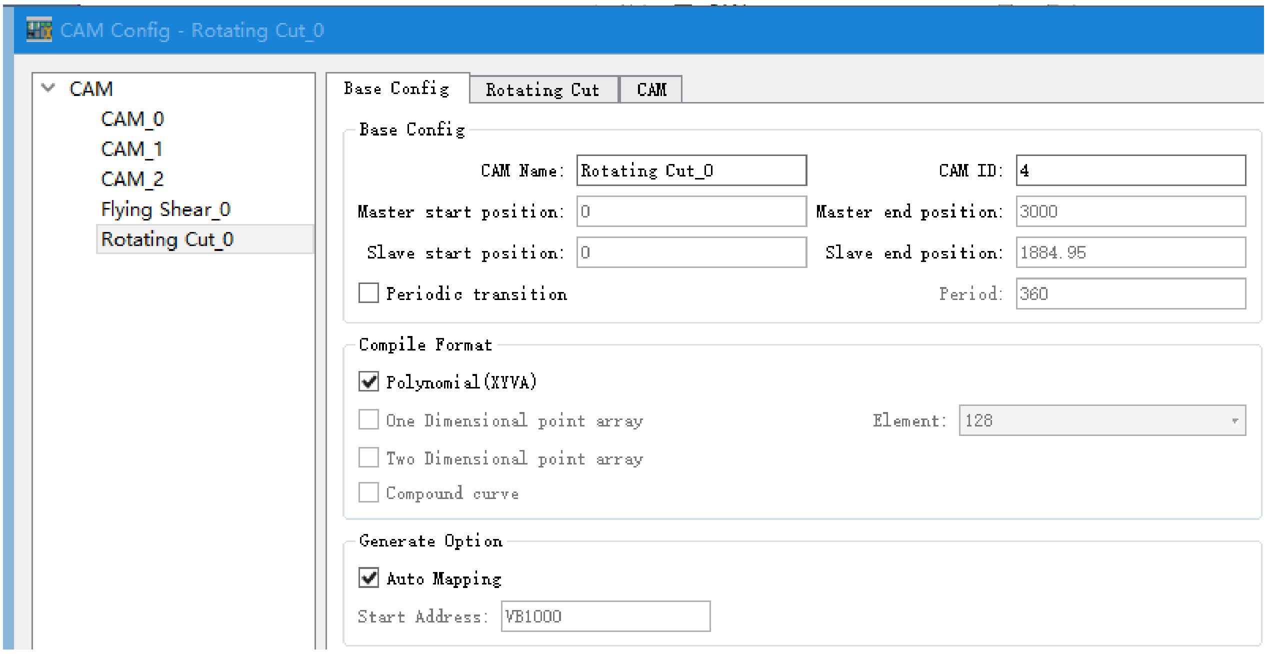


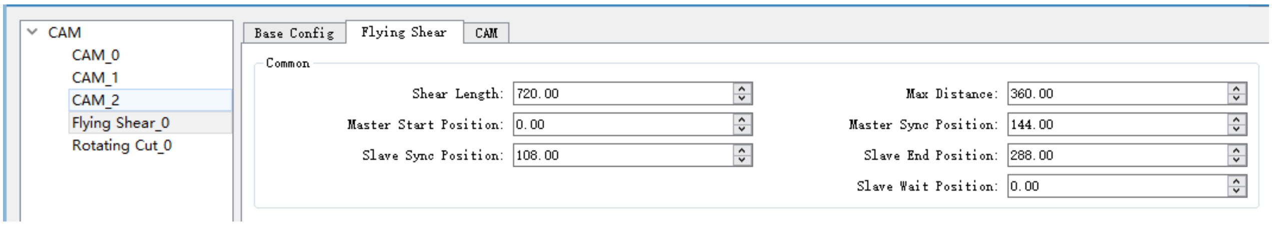
CAM Table

This page should display the information of the red nodes in the CAM page curve graph. Each red node corresponds to a set of data. Click "" to add a new node, and click "" to delete an already added node. Double click the values corresponding to "X, Y, V, A" to manually modify them.



5. After inserting the clip, the configuration interface is shown below.



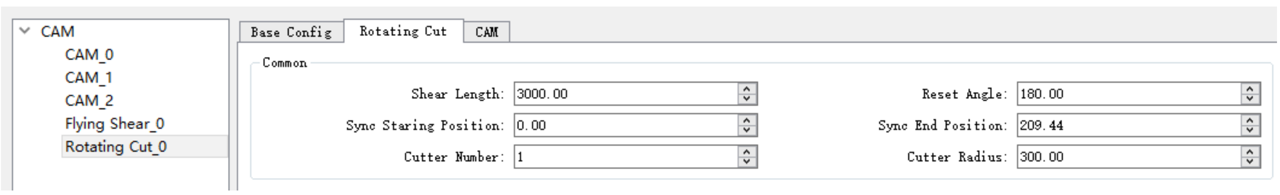


【 Basic Configuration 】

- Basic Configuration: Users can manually change the CAM name and CAM ID. The spindle position corresponds to the horizontal axis in the CAM curve graph, and the vertical axis in the CAM curve graph corresponds to the axial position. Check "Periodic variation", and the CAM curve graph will show periodic variation. The end position of the spindle is the periodic variation.
- Compilation type ": Check" Polynomial (XYVA) ", CAM will display the horizontal and vertical coordinates, velocity, and acceleration states in the state table with the symbols x, y, a, and v. It supports compilation types of" one-dimensional array "and" two-dimensional array ".
- Generate Options ": Check" Auto Mapping "to set the default starting address to VB1000. Uncheck to modify the starting address, which corresponds to the starting address in the symbol table.

⚠	be careful The automatically mapped V-zone address is not suitable for dynamically changing curves, and modifying the curve with the trimming and flying cutting instructions can cause an increase or decrease in the number of curve points, leading to memory conflicts. When using the trimming and flying cutting command, it is recommended that users customize the starting address and reserve sufficient memory range.

[Rotating cut configuration]



The parameters of the main axis and sub axis, as well as the cutting length of the material, can be set in the trimming configuration.

Cutting length: refers to the length of material that needs to be cut.

Maximum distance: refers to the maximum distance that can be moved from the axis.

Spindle starting position: The position where the spindle starts to transport the material forward.

Spindle synchronization position: The position where spindle synchronization begins, at which the spindle speed remains consistent with the slave shaft speed.

Axis synchronization position: The position from which axis synchronization begins, where the axis and spindle speeds are consistent.

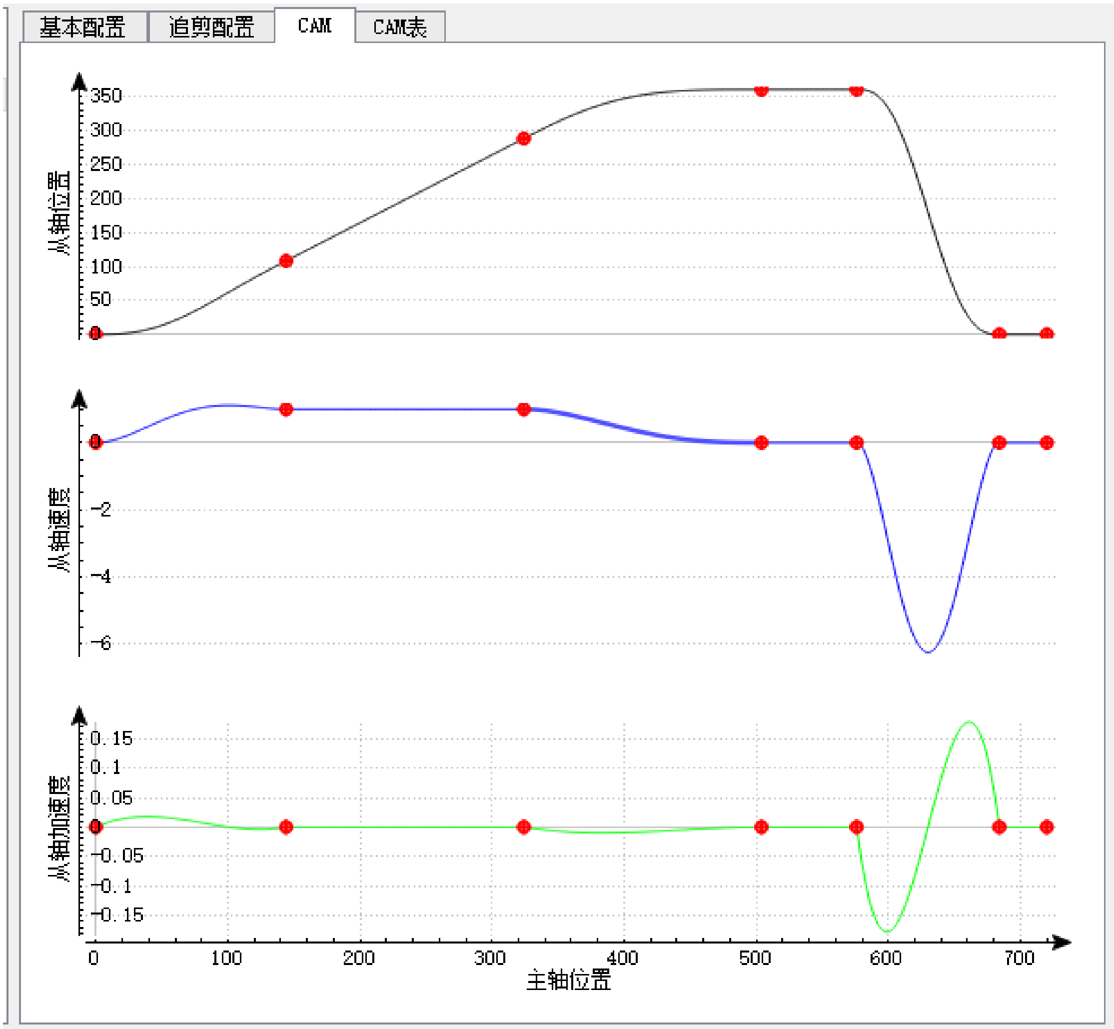
End position of axis synchronization: The position where axis synchronization ends.

Return from axis to main axis displacement: The displacement of the main axis during the process of returning to the origin in the opposite direction after walking from the axis to the set maximum distance.

Main axis displacement during waiting from the axis: The displacement of the main axis during waiting from the axis back to the origin.

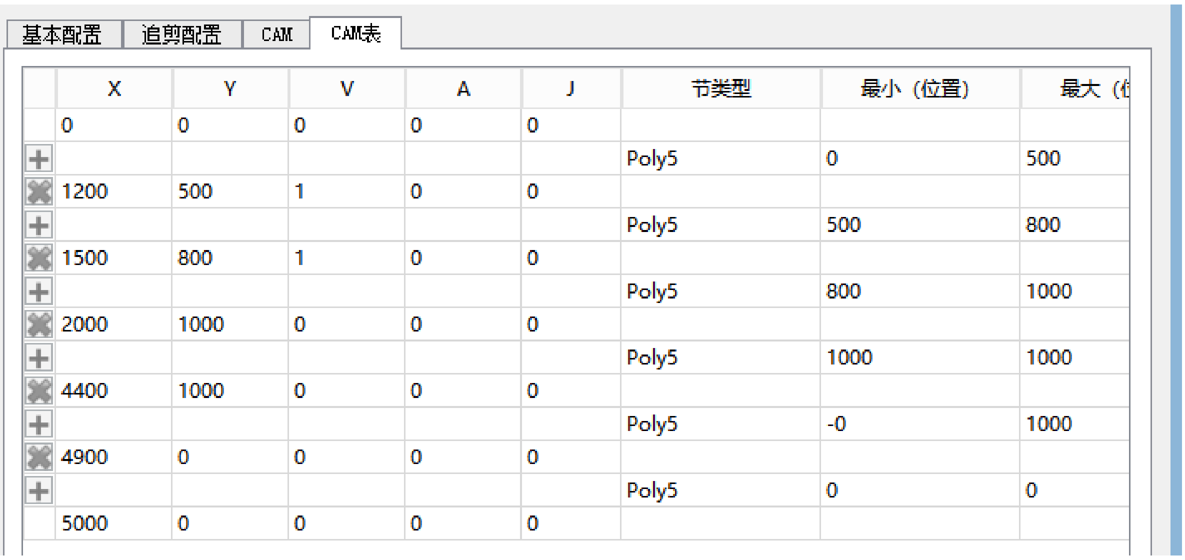
[CAM]

This page displays the relationship between "main axis position" and "sub axis position", "sub axis velocity", and "sub axis acceleration" in the form of a curve graph. The maximum value of the "main axis position" on the horizontal axis corresponds to the setting value of the "main axis end position" on the "basic configuration" interface, and the maximum value of the "sub axis position" on the vertical axis corresponds to the setting value of the "sub axis end position" on the "basic configuration" interface. The red nodes on the CAM curve of the flying shear cannot be dragged.



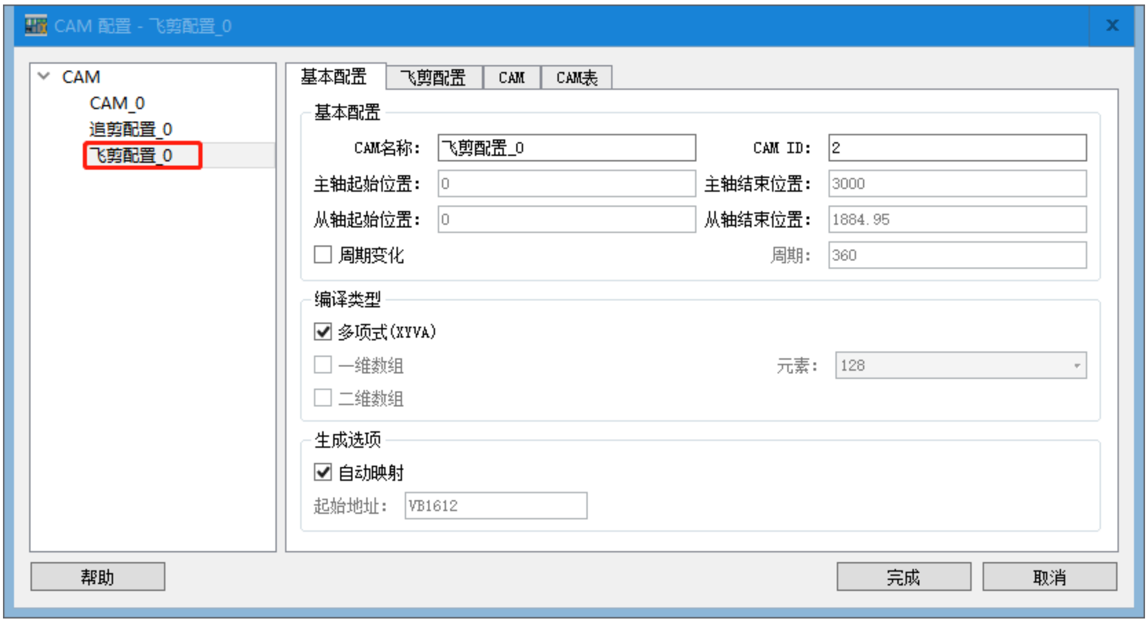
CAM Table

The CAM table parameters in the trimming process cannot be modified. This page should display the information of the red nodes in the CAM page curve graph, where each red node corresponds to a set of data.



X	Y	V	A	J	节类型	最小（位置）	最大（位
0	0	0	0	0			
					Poly5	0	500
1200	500	1	0	0			
					Poly5	500	800
1500	800	1	0	0			
					Poly5	800	1000
2000	1000	0	0	0			
					Poly5	1000	1000
4400	1000	0	0	0			
					Poly5	-0	1000
4900	0	0	0	0			
					Poly5	0	0
5000	0	0	0	0			

6. After inserting the flying shear, the configuration interface is shown below.



● Basic Configuration: Users can manually change the CAM name and CAM ID. The spindle position corresponds to the horizontal axis in the CAM curve graph, and the vertical axis in the CAM curve graph corresponds to the axial position. Check "Periodic variation", and the CAM curve graph will show periodic variation. The end position of the spindle is the periodic variation.

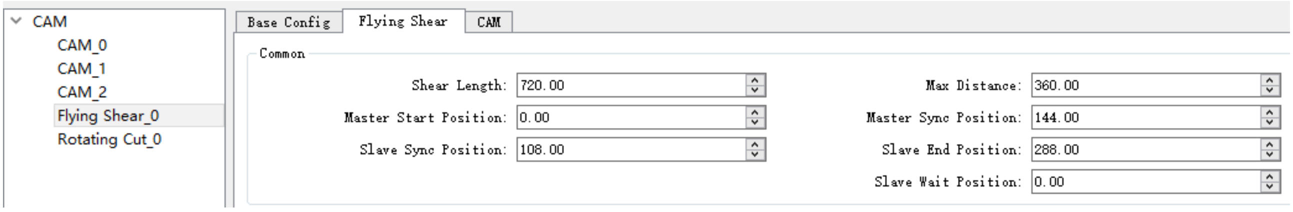
● Compilation type ": Check" Polynomial (XYVA) ", CAM will display the horizontal and vertical coordinates, velocity and acceleration states in the state table with the symbols x, y, a, v. Currently, the compilation types of" one-dimensional array "and" two-dimensional array "are not supported.

● Generate Options ": Check" Auto Mapping "to set the default starting address to VB1000. Uncheck to modify the starting address, which corresponds to the starting address in the symbol table.

⚠	be careful The automatically mapped V-zone address is not suitable for dynamically changing curves, and modifying the curve with the trimming and flying cutting instructions can cause an increase or decrease in the number of curve points, leading to memory conflicts. When using the trimming and flying cutting command, it is recommended that users customize the starting address and reserve sufficient memory range.

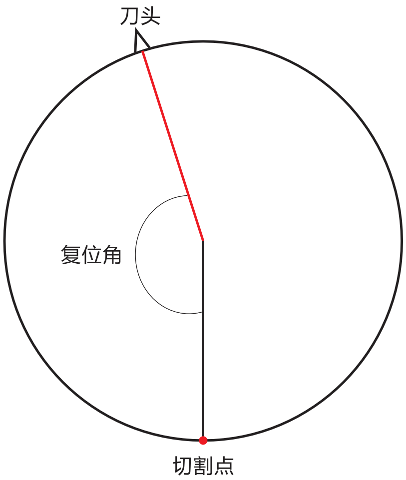
[Flying shear Configuration]

The parameters of the axis and the cutting length of the material can be set in the flying shear configuration.



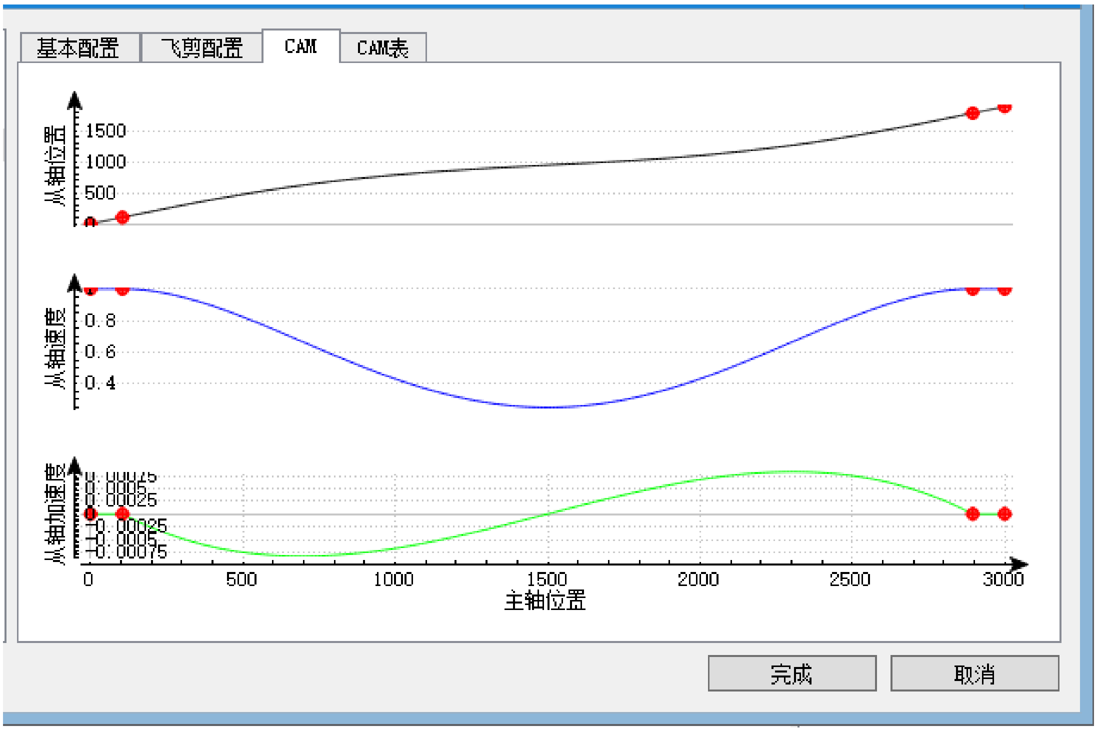
Cutting length: refers to the length of material that needs to be cut.
Reset angle: refers to the angle value at which the current synchronous angle edge reaches the cutting point after the cutter runs in the running direction from the origin after resetting.
Front synchronization angle: The angle of the synchronization area that the material passes through first is the front synchronization angle.
Post synchronization angle: The angle of the synchronization area that the material passes through is the post synchronization angle, and currently it is required that the size of the front and rear synchronization angles be consistent.
Number of cutting blades: the number of cutting blades on the axis. If the number of cutting blades is two or more, it is required that the cutting blades are installed at a uniform angle, and the angle between each pair of cutting blades is the same.
Blade radius: The distance from the axis center to the blade head is the blade radius.

The schematic diagram of the shaft structure is as follows:



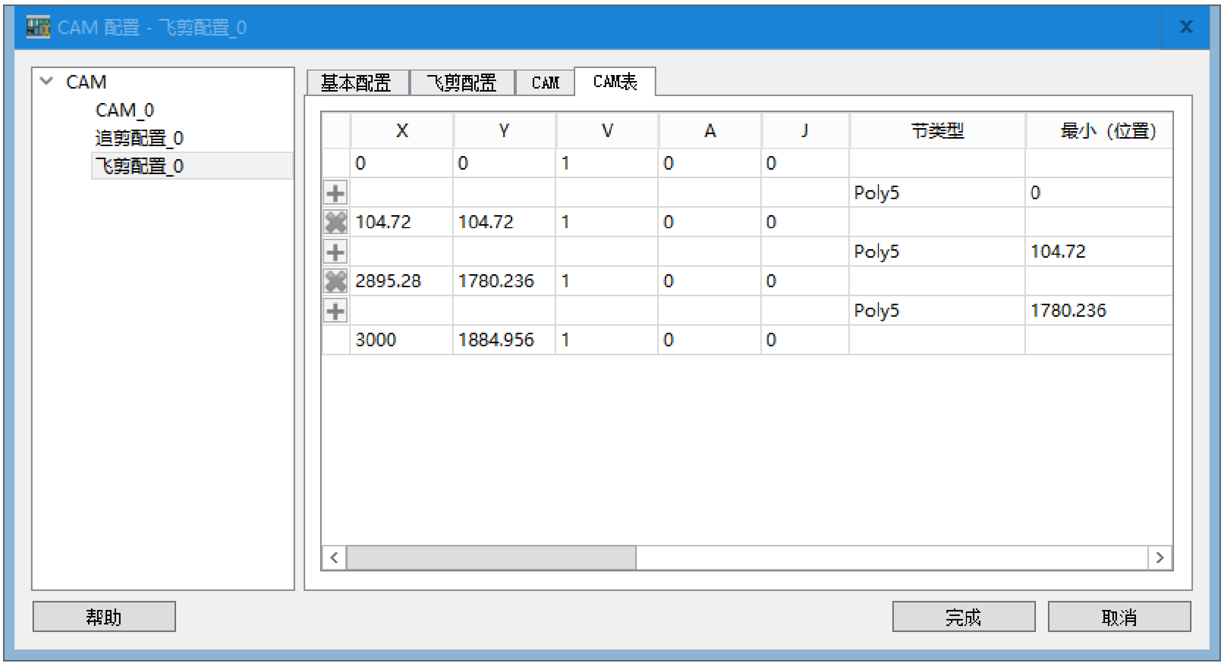
[CAM]

This page displays the relationship between "main axis position" and "sub axis position", "sub axis velocity", and "sub axis acceleration" in the form of a curve graph. The maximum value of the "main axis position" on the horizontal axis corresponds to the setting value of the "main axis end position" on the "basic configuration" interface, and the maximum value of the "sub axis position" on the vertical axis corresponds to the setting value of the "sub axis end position" on the "basic configuration" interface. Unlike regular camshafts, the red nodes on the CAM curve of the flying shear cannot be dragged.



CAM Table

The CAM table parameters in the flying shear cannot be modified. This page should display the information of the red nodes in the CAM page curve graph, where each red node corresponds to a set of data.



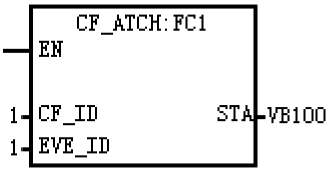
X	Y	V	A	J	节类型	最小（位置)
0	0	1	0	0		
					Poly5	0
104.72	104.72	1	0	0		
					Poly5	104.72
2895.28	1780.236	1	0	0		
					Poly5	1780.236
3000	1884.956	1	0	0		

6.3 Axis instruction plc_open_stm (V1.1) library instruction

Function Name	Instruction Name	M226ES
CF_ATCHC	Language block connection instruction	Unsupported
SMC_CtrlByActPos	Position follows the current position command	Unsupported

SMC_CtrlByPos	Position following instruction	Unsupported
SMC_GetCamMasterPos	Obtain the position of the cam spindle	Unsupported
SMC_GetCamSlavePos	Obtain camshaft position information	Unsupported
SMC_GetPosByEvent	Position capture command	support
SMC_Home	Special Origin Regression Instruction	support
SMC_PosToPulse	Position conversion instruction	support

CF_STCHC language block connection instruction - temporarily not supported

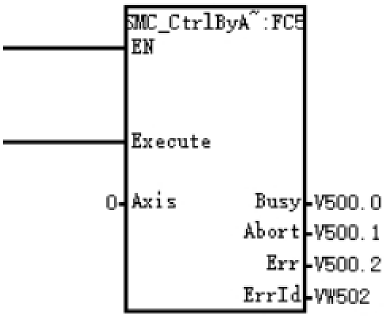


Function: This instruction adds the C language code block pointed to by CF-ID to the corresponding position for execution.

Parameter Description

Parameter Name	Input/Output	Parameter description	type	remarks
CF_ID	IN	Instruction execution conditions	BOOL	The ID number of the C language block, for example, CF-ID of CF2 is 2
EVE_ID	IN	Parameter number	WORD	0: EtherCAT cycle 1: Pulse axis period
STA	OUT	Parameter pointer	DWORD	0: Execution successful 1: Parameter error

SMC_TrlByActPos position follows current position instruction - temporarily not supported



Function: After calling this command, the axis position will be controlled by an external command, and the axis setting position will follow the current position.

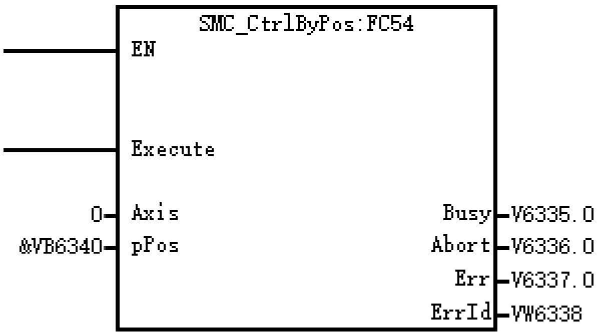
Parameter description:

Parameter Name	Input/Output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction. When the execution condition changes from ON to Off, cancel the current instruction
Axis	IN	Spindle ID number	BYTE	Axis number

Busy	OUT	Instruction processing in progress	BOOL	True, instruction is currently being processed
Abort	OUT	Command termination bit	BOOL	When the instruction is terminated during execution, the Abort bit is set. When the execution condition of the instruction is Off, the Abort bit is reset.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

SMC_TrlByPOS position following instruction - temporarily not supported

Function Name: SMC_TrlByPOS

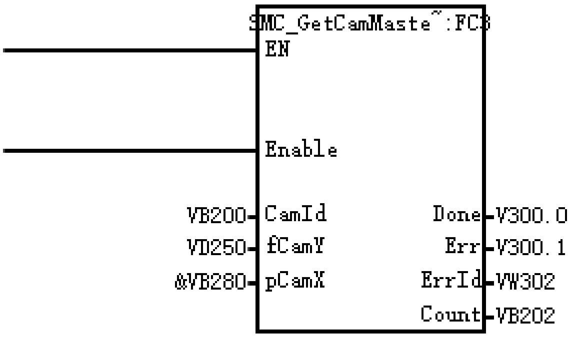


Function: After calling this command, the axis position will be controlled by an external command, and the axis setting position will follow the memory pointed to by pPos.

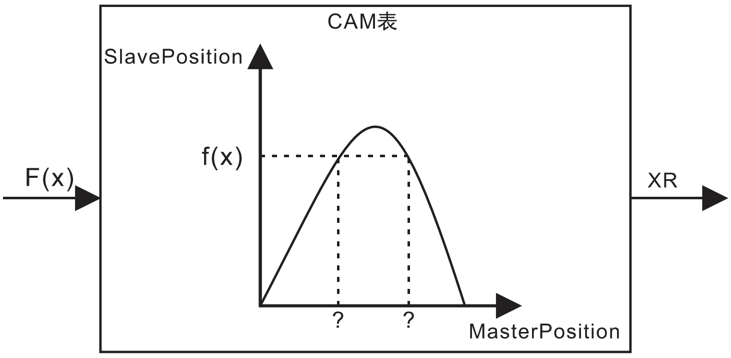
Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction. When the execution condition changes from ON to Off, cancel the current instruction
Axis	IN	Spindle ID number	BYTE	Axis number
pPos	IN	Position following value pointer	Pointer	
Busy	OUT	Instruction processing in progress	BOOL	True, instruction is currently being processed
Abort	OUT	Command termination bit	BOOL	When the instruction is terminated during execution, the Abort bit is set. When the execution condition of the instruction is Off, the Abort bit is reset.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

SMC_GetCamMasterPos obtains cam spindle position - temporarily not supported



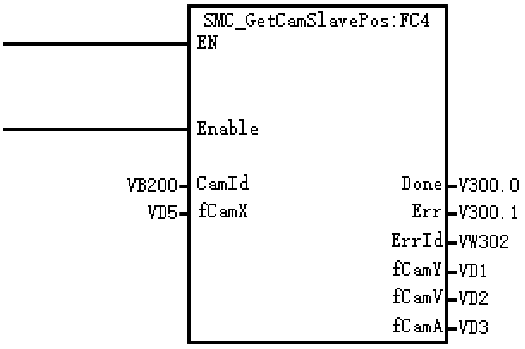
Function: This command is used to obtain the position of the cam spindle. Given the cam table and camshaft position values, calculate the corresponding camshaft spindle position value. The correct result can only be obtained after executing MC_CamTableSelect.



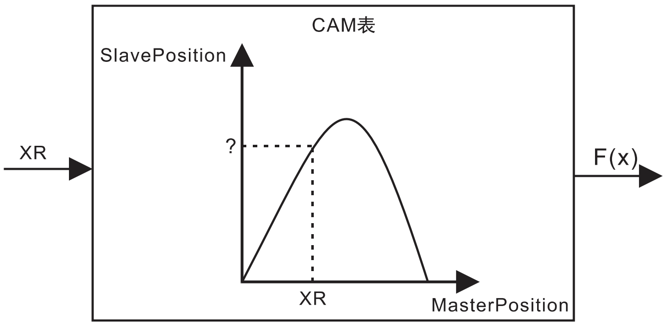
Parameter description:

Parameter Name	Input/Output	Parameter description	type	remarks
Enable	IN	Instruction execution conditions	BOOL	When the execution condition changes to On, execute the instruction
CamId	IN	ID number of cam table	BYTE	
fCamY	IN	Cam position value from the shaft	REAL	Cam position value from the shaft
pCamX	IN	Spindle position pointer	Pointer	There may be multiple solutions, and the solutions exist in the memory pointed to by pCamx. If the fitted curve in the cam table is a zero power curve and there are countless solutions, only the two endpoints of the line segment are taken. If it is a 5th power curve, the function block will be evaluated internally using the tangent method, with an accuracy of 6 decimal places.
Done	OUT	Complete position	BOOL	Instruction completed, Done set
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.
Count	OUT	The number of solutions	BYTE	

SMC_GetCamSlavePos retrieves camshaft position information - temporarily not supported



Function: This command is used to obtain camshaft position information. Given the cam table and cam spindle position values, calculate the corresponding cam slave shaft position values and speed ratios.

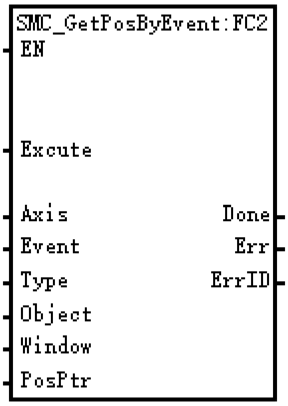


Parameter description:

Parameter Name	Input/Output	Parameter description	type	remarks
Enable	IN	Instruction execution conditions	BOOL	When the execution condition changes to On, execute the instruction.
CamId	IN	ID number of cam table	BYTE	
fCamX	IN	Position value of cam spindle	REAL	Position value of cam spindle
fCamY	OUT	Cam position value from the shaft	REAL	Cam position value from the shaft
fCamV	OUT	Cam shaft speed ratio	REAL	Cam shaft speed ratio
fCamA	OUT	Cam shaft acceleration ratio	REAL	Cam shaft acceleration ratio
Done	OUT	Complete position	BOOL	Instruction completed, Done set
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

SMC0tPosByEvent position capture command

Function Name: SMC_GetPosByEvent



Function: This instruction binds an IO interrupt. When the interrupt occurs, it reads the output pulse count (HSC or mapped memory value) of a certain axis and configures it according to the dimension of the axis. It converts this pulse value into a coordinate value and outputs it to the memory specified by the PosPtr pointer.

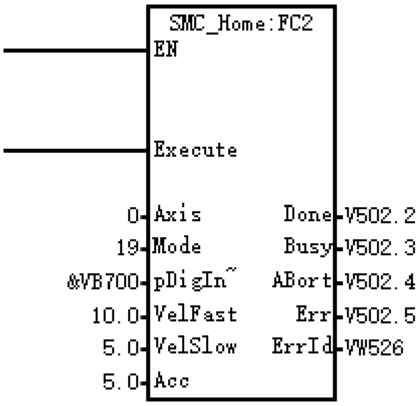
be careful

1. You only need to call once to bind interrupt events.
2. The ATC instruction (interrupt connection instruction) must be called first, and then the ENI instruction (interrupt enable instruction) must be called before the instruction interrupt can be generated.
3. This instruction function block is executed before the OB block bound to the ATC instruction, and its speed is faster than the values read from within the OB block.
4. If it is HSC, it must be initialized before this instruction in order to read the changing pulse values.

Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	Ascending edge execution
Axis	IN	Axis number	BYTE	
Event	IN	Event Interrupt Number	BYTE	Only supports local IO interrupts
Type	IN	type	BYTE	0: HSC, when an interrupt occurs, the current value of HSC is read. If HSC captures the interrupt, the HSC hardware capture register will be read. 1: Memory mapping, when an interrupt occurs, reads the mapped memory value.
Object	IN	read object	DWORD	If the Type parameter is 0, then this value represents the ID of HSC, ranging from 0 to 5. If the Type parameter is 1, then the value is a pointer to memory.
Window	IN	Detection window	REAL	The minimum interval position for two consecutive conversions
PosPtr	IN	Conversion results	REAL	Pointer for storing conversion results
Done	OUT	Complete position	BOOL	0: Instruction not completed 1: Instruction completed
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

SMC_Home Special Origin Regression Instruction



Function: This instruction is controlled by the controller to execute the return to origin action on the axis according to the given mode and speed parameters. The return to origin mode and speed are set in the instruction. Note that in general, the origin switch/limit signal of the axis is connected to the PLC system. After returning to the original, the controller will bias the axis coordinate to 0.0, and the 16 # 6064:0 (position actual value) of the axis itself will not change due to operation instructions.

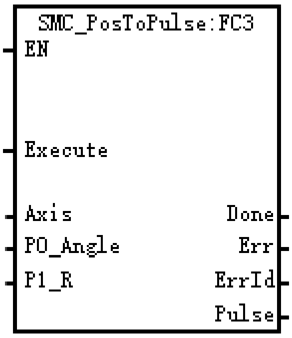
Parameter description:

Parameter Name	Input/Output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
Axis	IN	Axis ID number	BYTE	
Mode	IN	Homing mode	BYTE	Refer to Pulse Axis Return to Origin, 6.7.1 Homing mode description
pDigInput	IN	Input signal pointer	Pointer	The memory address pointed to by the position number of the origin switch, forward and reverse limit switches. In memory Bit0: Negative Switch Bit1: Forward switch Bit2: Origin switch Switch signal 1 is valid, 0 is invalid. For example, when pDigInput is equal to&IB0, I0.0: Negative Switch I0.1: Positive switch I0.2: Origin switch.
VelFast	IN	Fast Homing speed	REAL	The fast speed when searching for the origin, this parameter is always positive. (Unit: Unit/second)
VelSlow	IN	Slow Homing speed	REAL	The slow speed when searching for the origin, this parameter is always positive. (Unit: Unit/second)
Acc	IN	acceleration	REAL	The acceleration of the terminal actuator is always positive. (Unit: Unit/second/second)
Done	OUT	Complete position	BOOL	After returning to the original position, the Done position is set. When the execution condition of the instruction is Off, the Done bit is reset.
Busy	OUT	Instruction execution bit	BOOL	When the instruction is executed. Busy position is set. When the instruction is completed, Busy resets. When the execution condition of the instruction is Off, the Busy bit is reset.
Abort	OUT	Command termination bit	BOOL	When the instruction is terminated during execution, the Abort bit is set.

				When the execution condition of the instruction is Off, the Abort bit is reset.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

SMC_PosToPulse position conversion instruction

Function Name: SMC_PosToPulse

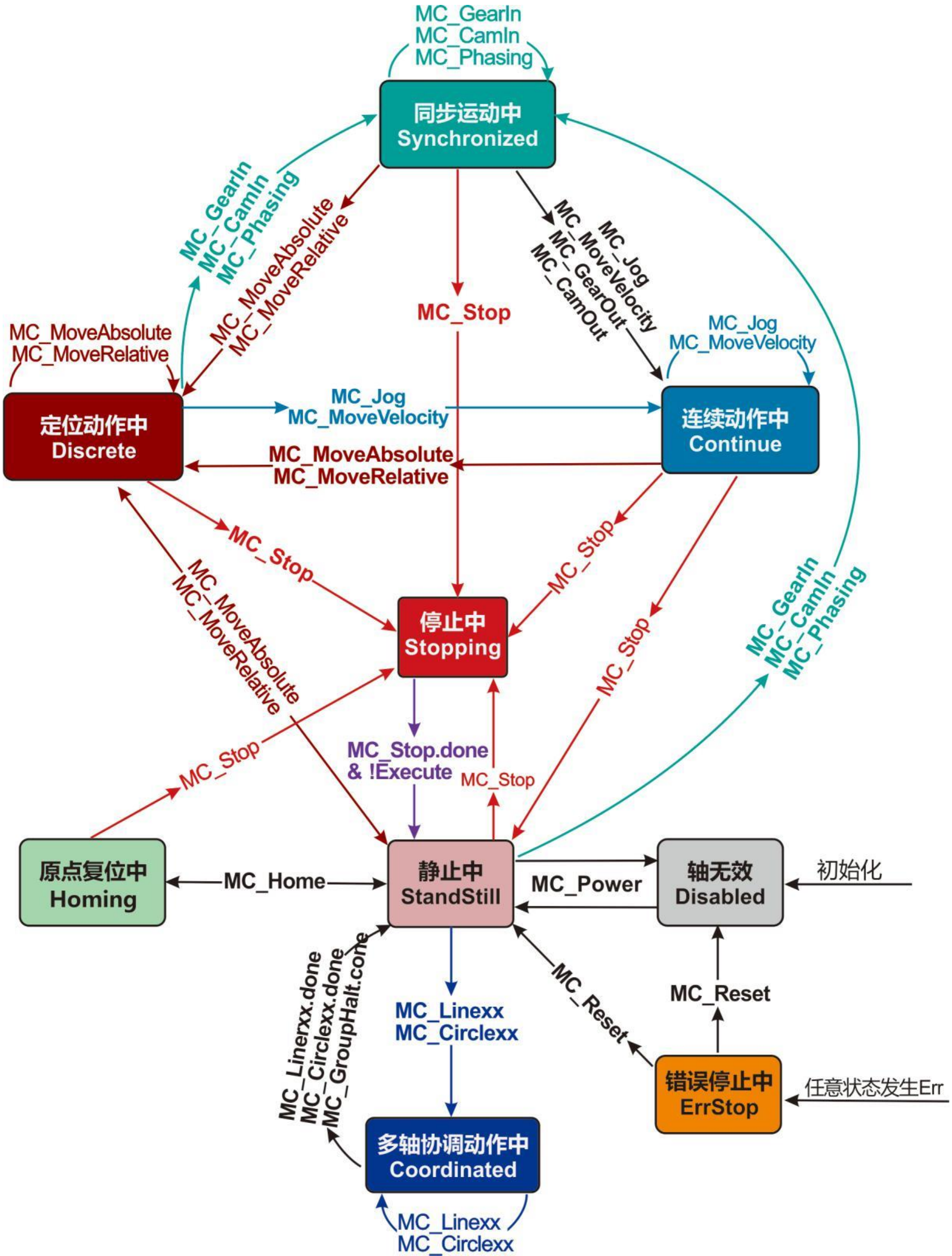


Function: This command converts two input points into actual pulse numbers based on their axial dimensions. Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	Ascending edge execution
Axis	IN	Axis number	BYTE	Need to use dimensional parameters and types
P0_Angle	IN	Event Interrupt Number	REAL	Position parameter 0, if it is a linear axis, it is position 0, if it is a rotational axis, it is angle.
P1_R	IN	type	REAL	Position parameter 1, if it is a linear axis, it is position 1, and if it is a rotational axis, it is radius.
Done	OUT	Complete position	BOOL	0: Instruction not completed 1: Instruction completed
Err	OUT	Error position	BOOL	
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.
Pulse	OUT	Number of pulses after conversion	DWORD	

6.4 Introduction to the Axis Instruction "ct_plcopentlib"

The Axis instruction is designed according to the PLCOpen standard, and the flowchart of the axis status instruction is shown in the following figure:



Ct_plcopentlib (v2.9) is divided into three types of instructions, namely single axis instructions, multi axis instructions, and axis group instructions. The following chapters will introduce these three types of axis instructions.

6.4.1 Single axis control command

Single axis control instructions are generally used for positioning control, that is, servo motors drag external mechanisms to a specified position.

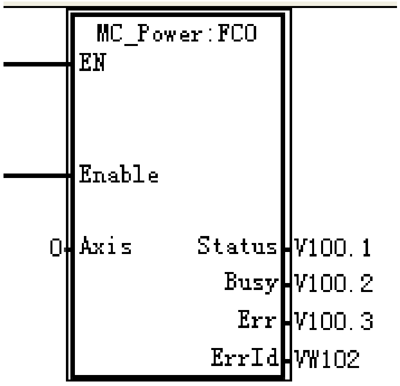
The commonly used MC function blocks for single axis control are as follows:

Function Name	Instruction Name	M226ES
MC_Power	Axis enable command	support
MC_ReadStatus	Read axis status command	support

MC_ReadSetPos	Read axis setting position command	support
MC_MoveAbsolute	Absolute displacement instruction	support
MC_MoveRelative	Relative displacement command	support
MC_MoveVelocity	Speed command	support
MC_MoveJog	Jog command	support
MC_Home	Origin Regression Instruction	support
MC_Stop	Stop command	support
MC_Halt	Interruptable stop command	support
MC_Reset	Reset error command	support
MC_SetPosition	Set current location command	support
MC_MoveFeed	Interrupt event triggers motion command	support
MC_ReadParameter	Read parameter instruction	support
MC_WriteParameter	Write parameter instructions	support
MC_TouchProbe	Probe instruction	Unsupported
MC_TorqueControl	Torque control command	Unsupported
MC_MoveFeed	Interrupt event triggers motion command	support
MC_MoveVelocityCSV	Speed command (CSV mode)	Unsupported
MC_Stop	Stop command	support
MC_Halt	Interruptable stop command	support
MC_HaltSuperImposed	Overlay Abort Instruction	Unsupported

MC-Power axis enable command

Function Name: MC-Power



Function: This command is used to enable or disable the corresponding axis.

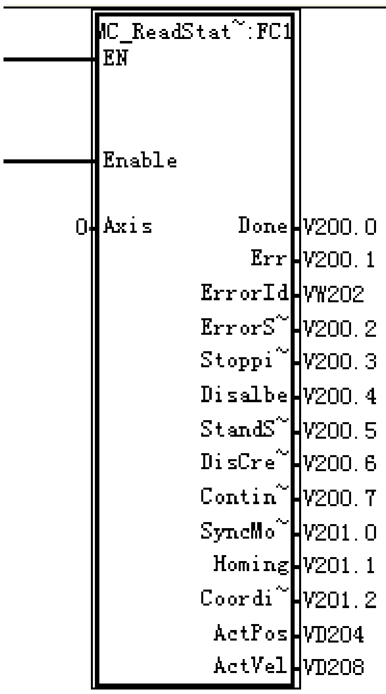
Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks

Enable	IN	Axis enablement	BOOL	Set to TRUE to enter the runnable state, set to False to release the runnable state.
Axis	IN	Axis ID number	BYTE	
Status	OUT	Status bit	BOOL	When entering the runnable state, it becomes TRUE. Entering an inoperable state becomes false.
Busy	OUT	Instruction execution bit	BOOL	When the instruction is executed. Busy position is set. When the command completes Busy reset. When the execution condition of the instruction is Off, the Busy bit is reset.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction changes from On to Off, the Error bit is reset.
ErrId	OUT	Error code	WORD	Error codes, see 6.4.4 error code.

MC_SeadStatus Read Axis Status Instruction

Function Name: MC_SeadStatus



Function: This command is used to read the values of the corresponding axis status parameters.

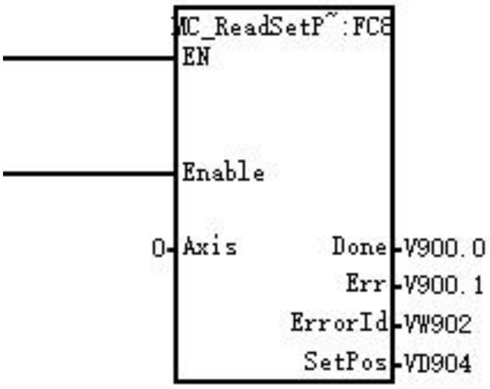
Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks
enable	IN	Instruction execution conditions	BOOL	When On, execute the instruction.
Axis	IN	Axis ID number	BYTE	Axis number
Done	OUT	Status bit	BOOL	After the instruction is executed successfully, the Done bit is set. When the execution condition of the instruction is Off, the Done bit is reset.
Error	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction changes from On to Off, the Error bit is reset.
ErrId	OUT	Error code	WORD	Error code When Error is TRUE, ErrId is the error of the instruction. When ErrStop is TRUE, ErrId is the current axis error.

				Error codes, see 6.4.4 error code.
ErrStop	OUT	Status indicator bit	BOOL	True, if the axis is in the ErrStop state
Stopping	OUT	Status indicator bit	BOOL	True, if the axis is in the state of Stopping
Disabled	OUT	Status indicator bit	BOOL	True, if the axis is in the disabled state
StandStill	OUT	Status indicator bit	BOOL	True, if the axis is in the StandStill state
DisCreteMotion	OUT	Status indicator bit	BOOL	True, if the axis is in the DisCreteMotion state
ContinousMotion	OUT	Status indicator bit	BOOL	True, if the axis is in the state of ContinuumMotion
SyncMotion	OUT	Status indicator bit	BOOL	True, if the axis is in the SyncMotion state
Homing	OUT	Status indicator bit	BOOL	True, if the axis is in the Homing state
Coordinate	OUT	Status indicator bit	BOOL	True, if the axis is in the Coordinate state
ActPos	OUT	Current coordinates	REAL	unit
ActVel	OUT	Current speed	REAL	Unit/second

MC_ ReadSet BIOS reads axis setting position command

Function Name: MC_SeadSetPhos



Function: This command is used to read the current set position of the axis.

Parameter Description

Parameter Name	Input/output	Parameter description	type	remarks
enable	IN	Instruction execution conditions	BOOL	When On, execute the instruction.
Axis	IN	Axis ID number	BYTE	Axis number
Done	OUT	Status bit	BOOL	After the instruction is executed successfully, the Done bit is set. When the execution condition of the instruction is Off, the Done bit is reset.
Error	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction changes from On to Off, the Error bit is reset.
ErrId	OUT	Error code	WORD	When Error is TRUE, ErrId is the error of the instruction. When ErrStop is TRUE, ErrId is the current axis error. Error codes, see 6.4.4 error code.

Setpos	OUT	Current setting location	REAL	unit

MC_SeadVelPos reads axis speed and position commands

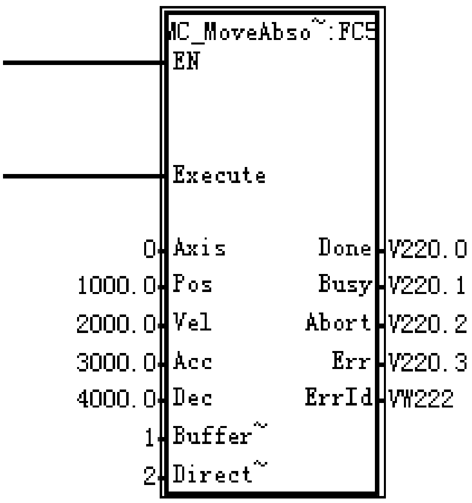
Function: This command is used to read the speed and position of the axis.

Parameter Description

Parameter Name	Input/Output	Parameter description	type	remarks
Axis	IN	Axis ID number	BYTE	Axis number
Err	OUT	Error position	BOOL	If an error is detected, the Err bit is set.
ErrId	OUT	Error code	WORD	When Err is TRUE, ErrId is the error of the instruction.
Actpos	OUT	Current actual location	REAL	unit
ActVel	OUT	Current actual speed	REAL	Unit/second
Setpos	OUT	Current setting location	REAL	unit
SetVel	OUT	Current set speed	REAL	Unit/second

MC_SoveAbsolute absolute displacement instruction

Function Name: MC_SoveAbsolute



Function: This command is used to control the terminal execution mechanism to accelerate and decelerate at a given speed to move to the target position relative to zero. Once this command is terminated during operation, the remaining unfinished distance will be discarded and a new command function will be executed.

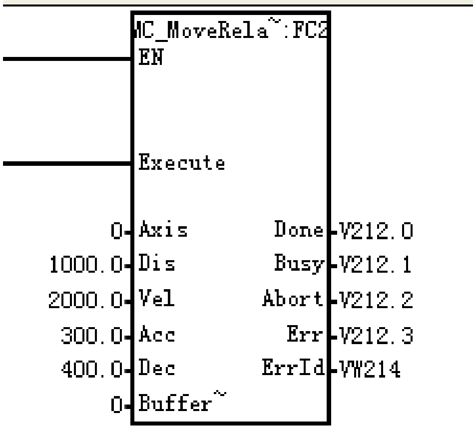
Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks

Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
Axis	IN	Axis ID number	BYTE	
Pos	IN	position	REAL	The target position of the terminal actuator relative to the zero point, unit: unit
Vel	IN	speed	REAL	The operating speed of the terminal actuator is always positive. (Unit: Unit/second)
Acc	IN	acceleration	REAL	The acceleration of the terminal actuator is always positive. (Unit: Unit/second/second)
Dec	IN	deceleration	REAL	The deceleration of the terminal actuator is always positive. (Unit: Unit/second/second)
BufferMode	IN	Abort mode	BYTE	Define the action of the axis: 0: Abort (directly interrupts the ongoing instruction) Other: Illegal
Direction	IN	direction	INT	Operation direction: 0: The direction with the shortest distance. 1: Positive direction. -1: In the opposite direction. 2: Continuing the current direction. This parameter is effective for the rotation axis and not for the linear axis.
Done	OUT	Complete position	BOOL	When the absolute displacement action is completed, the Done position is located. When the execution condition of the instruction is Off, the Done bit is reset.
Busy	OUT	Instruction execution bit	BOOL	When the instruction is executed. Busy position is set. When the command completes Busy reset. When the execution condition of the instruction is Off, the Busy bit is reset.
Abort	OUT	Command termination bit	BOOL	When the instruction is terminated during execution, the Abort position bit. When the execution condition of the instruction is Off, the Abort bit is reset.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

MC_SoveRelative relative displacement instruction

Function Name: MC_SoveRelative



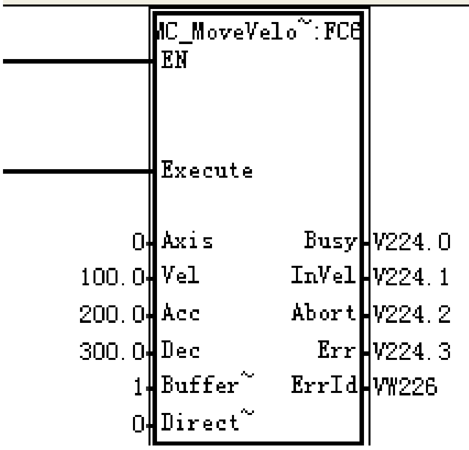
Function: This command is used to control the terminal execution mechanism to move a given distance at a given speed, with the current position as the reference point. Once this command is terminated during operation, the remaining unfinished distance will be discarded and a new command function will be executed.

Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
Axis	IN	Axis ID number	BYTE	
Dis	IN	distance	REAL	Based on the current position, the target distance that the terminal actuator needs to move is set to a negative value, indicating that the axis will be reversed. Unit: Unit.
Vel	IN	speed	REAL	The operating speed of the terminal actuator is always positive. (Unit: Unit/second)
Acc	IN	acceleration	REAL	The acceleration of the terminal actuator is always positive. (Unit: Unit/second/second)
Dec	IN	deceleration	REAL	The deceleration of the terminal actuator is always positive. (Unit: Unit/second/second)
BufferMode	IN	Abort mode	BYTE	0: Abort (directly interrupts the ongoing instruction) Other: Illegal
Done	OUT	Complete position	BOOL	When the absolute displacement action is completed, the Done bit is set. When the execution condition of the instruction is Off, the Done bit is reset.
Busy	OUT	Instruction execution bit	BOOL	When the instruction is executed. Busy position is set. When the command completes Busy reset. When the execution condition of the instruction is Off, the Busy bit is reset.
Abort	OUT	Command termination bit	BOOL	When the instruction is terminated during execution, the Abort bit is set. When the execution condition of the instruction is Off, the Abort bit is reset.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

MC_SoveVelocity Speed Command

Function Name: MC_SoveVelocity



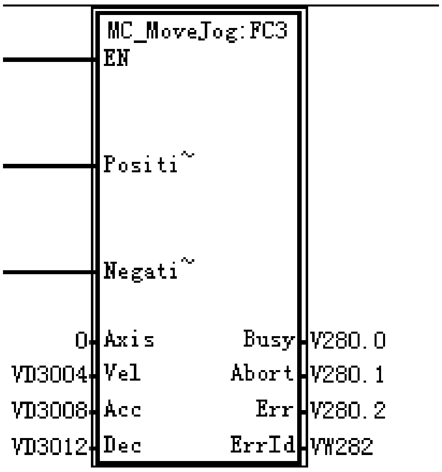
Function: This instruction is used to control the terminal execution mechanism to accelerate and decelerate to a given speed and run at a constant speed. When the terminal mechanism reaches the given speed, this instruction is executed, but the terminal mechanism continues to run at this speed.

Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
Axis	IN	Axis ID number	BYTE	
Vel	IN	speed	REAL	The operating speed of the terminal actuator is always positive. (Unit: Unit/second)
Acc	IN	acceleration	REAL	The acceleration of the terminal actuator is always positive. (Unit: Unit/second/second)
Dec	IN	deceleration	REAL	The deceleration of the terminal actuator is always positive. (Unit: Unit/second/second)
BufferMode	IN	Abort mode	BYTE	0: Abort (directly interrupts the ongoing instruction) Other: Illegal
Direction	IN	direction	INT	1: Positive direction -1: Negative direction
Busy	OUT	Instruction execution bit	BOOL	When the instruction is executed. Busy position is set. When the command completes Busy reset. When the execution condition of the instruction is Off, the Busy bit is reset.
Invel	OUT	Speed reaching position	BOOL	After reaching the target speed, the Invite bit is set. When instructed When the execution condition changes from On to Off, the Invite bit is reset.
Abort	OUT	Command termination bit	BOOL	When the instruction is terminated during execution, the Abort bit is set. When the execution condition of the instruction is Off, the Abort bit is reset.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

MC_SoveJog jog command

Function Name: MC_SoveJog



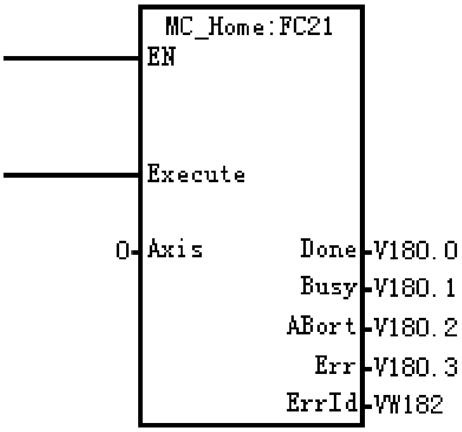
Function: Perform micro movements based on the specified Velocity. When positive micro movement is required, set PositiveEnable to TRUE, and when negative micro movement is required, set NegativeEnable to TRUE.

Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks
PositiveEnable	IN	Positive direction is effective	BOOL	If set to TRUE, start moving in the positive direction. Set to False to end the movement.
NegativeEnable	IN	Effective in the negative direction	BOOL	If set to TRUE, start moving in the negative direction. Set to False to end the movement.
Axis	IN	Axis ID number	DWORD	Axis number
Vel	IN	Target speed	REAL	Specify the target speed. The unit is [instruction unit/second].
Acc	IN	acceleration	REAL	The acceleration of the terminal actuator is always positive. (Unit: Unit/second/second)
Dec	IN	deceleration	REAL	The deceleration of the terminal actuator is always positive. (Unit: Unit/second/second) When set to 0, implement emergency stop.
Busy	OUT	Busy position	BOOL	The instruction is executing as TRUE and completing as FLASE.
Abort	OUT	Stop position	BOOL	When the instruction is terminated during execution, the Abort bit is set. When the execution condition of the instruction is Off, the Abort bit is reset.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

MC_Some Origin Regression Instruction

Function Name: MC_Some



Function: This command is used to control the axis to execute the return to origin action according to the mode and speed given by the axis parameters. The return to origin mode and speed are set in the axis parameter setting interface. Refer to Chapter 6.7 Homing function

Note: The configuration of the pulse axis and communication axis is different

● The pulse axis is set in the parameter setting interface to return to the original mode and return to the original speed. For details, see section 6.1.1 Axis configuration Pulse Axis Configuration.

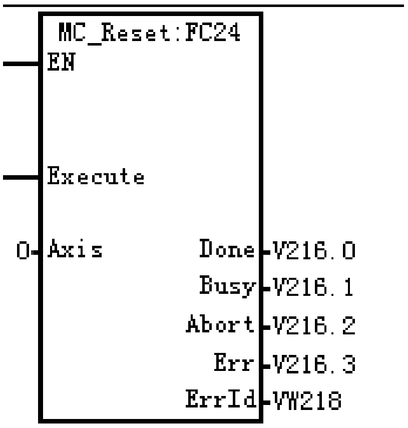
● For the communication axis, it is necessary to rewrite the value of the homing mode 16 # 6098:0 and configure it back to the original speed 16 # 6099:0, 16 # 6099:1. For specific operations, please refer to Chapter 6.1.1 Axis configurationCommunication Axis Configuration.

Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
Axis	IN	Axis ID number	BYTE	
Done	OUT	Complete position	BOOL	After the speed drops to 0, the Done position is set. When the execution condition of the instruction is Off, the Done bit is reset.
Busy	OUT	Busy position	BOOL	When the instruction is executed. Busy position is set. When the instruction is completed, Busy resets. When the execution condition of the instruction is Off, the Busy bit is reset.
Abort	OUT	Termination position	BOOL	When the instruction is terminated during execution, the Abort bit is set. When the execution condition of the instruction is Off, the Abort bit is reset.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

MC-Reset reset error command

Function Name: MC-Reset



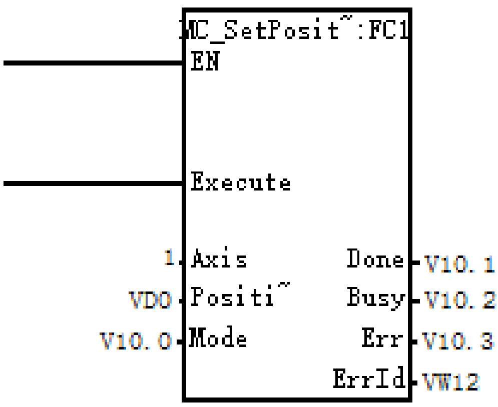
Function: Used to resolve axis anomalies. After resolving the anomaly, the axis status will transition from the Error Stop state to the Stand Still state.

Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
Axis	IN	Axis ID number	BYTE	
Done	OUT	Complete position	BOOL	After the axis state is reset to the ready to execute state, the Done bit is set. When the execution condition of the instruction changes from On to Off, the Done bit is reset
Busy	OUT	Instruction execution bit	BOOL	When the instruction is executed. Busy position is set. When the command completes Busy reset. When the execution condition of the instruction is Off, the Busy bit is reset.
Abort	OUT	Command termination bit	BOOL	When the instruction is terminated during execution, the Abort bit is set. When the execution condition of the instruction is Off, the Abort bit is reset.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

MC_SetPosition Set Current Position Instruction

Function Name: MC_SetPosition



Function: This command sets the value (unit) of the current coordinate.

be careful

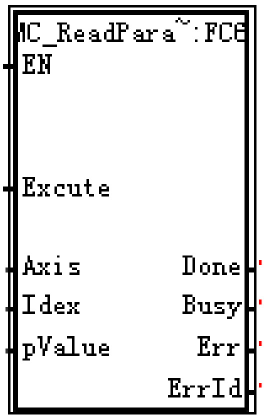
This instruction cannot be used when the axis is in motion, nor can it be used when the axis is the main axis of synchronous motion, otherwise unpredictable consequences may occur.

Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
Axis	IN	Axis ID number	BYTE	Axis number
Position	IN	Designated coordinate position	REAL	Designated coordinate position (unit) Value -1E+13<=Position<=1E+13
Mode	IN	Location mode	BOOL	0: Absolute position mode If this mode is specified, the absolute position of the axis is Position 1: Relative position mode If this mode is specified, the absolute position of the axis is the current position plus Position
Done	OUT	Status bit	BOOL	After the instruction is executed successfully, the Done bit is set. When the execution condition of the instruction is Off, the Done bit is reset.
Busy	OUT	Busy position	BOOL	Instruction is executing as TRUE, execution completed as FLASE
Error	OUT	Error position	OBOL	If an error is detected, the Error bit is set. When the execution condition of the instruction changes from On to Off, the Error bit is reset.
ErrId	OUT	Error code	WORD	Error codes, see 6.4.4 error code.

MC_SeadParameter Read Parameter Instruction

Function Name: MC_SeadParameter



Function: This command is used to read the values of the configuration parameters for the corresponding axis.

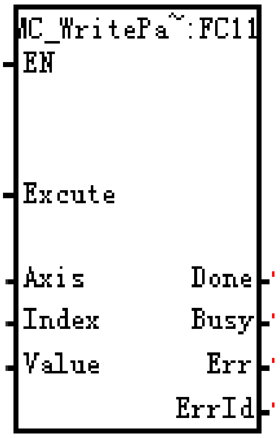
Parameter Description

Parameter Name	Input/output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
Index	IN	Parameter number	WORD	Refer to the parameter description table
pValue	OUT	Parameter pointer	DWORD	

Done	OUT	Status bit	BOOL	After the instruction is executed successfully, the Done bit is set. When the execution condition of the instruction is Off, the Done bit is reset.
Error	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction changes from On to Off, the Error bit is reset.
ErrId	OUT	Error code	WORD	Error codes, see 6.4.4 error code.

MC_TriteParameter Write Parameter Instruction

Function Name: MC_TriteParameter



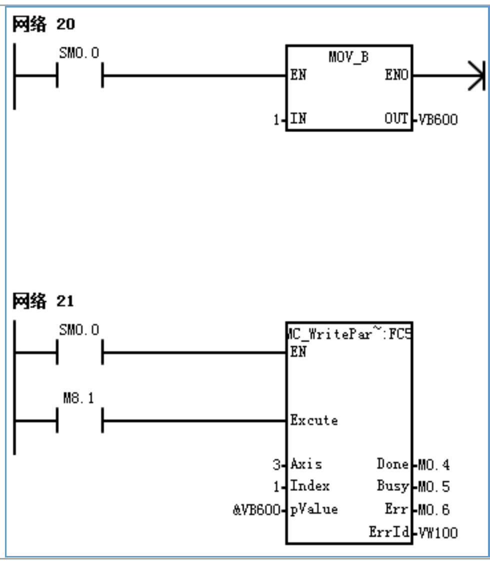
Function: This command is used to write the values of the corresponding axis configuration parameters. After successful writing, it is equivalent to modifying the hardware configuration and is permanently valid. This command can only be executed when the axis is not enabled.

Parameter Description

Parameter Name	Input/output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
Index	IN	Parameter number	WORD	Please refer to the **parameter description table** below
pValue	IN	Pointer to parameter value	DWORD	Parameter pointer
Done	OUT	Status bit	BOOL	After the instruction is executed successfully, the Done bit is set. When the execution condition of the instruction is Off, the Done bit is reset.
Error	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction changes from On to Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

Parameter Description Table

Index	parameter	pValue	explain
0	Dimensional conversion	PValue is a pointer to a dimensional description. See Appendix 3 for dimensional description.	
one	Acceleration and deceleration methods	PValue: A pointer pointing to the acceleration/deceleration mode. The acceleration and deceleration method is a BYTE	Axis 3 adopts an S-shaped acceleration and deceleration method

		Acceleration mode, 0: trapezoidal. 1: S-shaped. 2: Sin acceleration and deceleration	
two	speed	PValue: A pointer set to point to the speed. Refer to Appendix 4 for the directional structure	

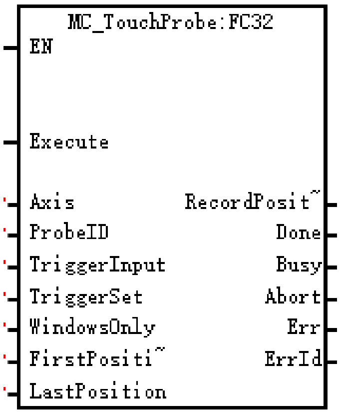
Appendix 3 Explanation of Data Structure for Quantity Conversion

name	type	explain
ScalPulseInc	REAL	Dimension conversion, pulse increment (hardware configuration dimension conversion left 1)
ScalMotorTurns	REAL	Dimensional conversion, number of motor turns (hardware configuration dimensional conversion on the right 1)
ScalMotorTurnsInc	REAL	Dimension conversion, incremental number of motor turns (hardware configuration dimension conversion, left 2)
ScalGearTurnsInc	REAL	Dimensional conversion, incremental number of gear turns (hardware configuration dimensional conversion, second on the right)
ScalGearTurnsInc	REAL	Dimensional conversion, incremental number of gear turns (hardware configuration dimensional conversion, left 3)
ScalAppUnitsc	REAL	Dimensional conversion, engineering units (hardware configuration dimensional conversion, left 3)

Appendix 4 Speed Setting Instructions

name	type	explain
VelMax	REAL	Maximum speed
VelMin	REAL	Minimum speed
VelStartStop	REAL	Start stop speed
AccMax	REAL	Maximum acceleration
AccMin	REAL	Minimum acceleration

MC_TouchProbe probe command - temporarily not supported



Function: Record the actual location when a triggering event occurs.

If communication axis is selected as the source for this instruction, an object dictionary with probe function needs to be added to PDO. Probe 1 is 0x60B8, 10x60B9, 0x60BA, 0x60BB. Probe 2 is 0x60B8, 0x60B9, 0x60BC, 0x60BD.

Parameter description:

Parameter Name	Input/Output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
Axis	IN	Axis ID number	BYTE	Axis number
ProbeID	IN	Probe ID	BYTE	0: PLC interrupt 1: Communication axis Probe1 2: Communication axis Probe2
TriggerInput	IN	Probe source	BYTE	Regarding PLC Interrupt: Interrupt Event Number, please refer to the table below for details - **Source** of **PLC Interrupt Probe** Regarding the communication axis: • Communication axis Probe1: After enabling the command, this setting will be mapped to bits 2-3 of drive 16 # 60B8:00 0: DI input 1: Z phase 2-3: Drive customization • Communication axis Probe2: After enabling the command, this setting will be mapped to bits 10-11 of drive 16 # 60B8:00 0: DI input 1: Z phase 2-3: Drive customization
TriggerSet	IN	Probe setting	BYTE	• For PLC interrupt: the location of the module that triggers the interrupt, for example, if the module is placed in rack number 1 and slot number 5, the address is 16 # 15. • For communication axis: After enabling the command, this setting will be mapped to the 4-5 bits of drive 16 # 60B8:00 0: Rising edge 1: Descending edge
WindowsOnly	IN	Window is valid	BOOL	Specify window 1 as valid and 0 as invalid.
FirstPosition	IN	Starting position	REAL	Specify the starting position for receiving triggers.
LastPosition	IN	Termination position	REAL	Specify the end position for receiving triggers.

Recordposition	OUT	Trigger record location	REAL	Current position when triggered
Done	OUT	Status bit	BOOL	After the instruction is executed successfully, the Done bit is set. When the execution condition of the instruction is Off, the Done bit is reset.
Busy	OUT	Busy position	BOOL	Instruction is executing as TRUE, execution completed as FLASE
Abort	OUT	Stop position	BOOL	When the instruction is terminated during execution, the Abort bit is set. When the execution condition of the instruction is Off, the Abort bit is reset.
Error	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction changes from On to Off, the Error bit is reset.
ErrId	OUT	Error code	WORD	Error codes, see 6.4.4 error code.

The following is an explanation of the interrupt event number table when the TriggerInput parameter in the probe instruction is a PLC interrupt, the validity of the Windows Only window, and the definition of the driver 16 # 60B8 bit.

1. Source of PLC interrupt probe: Interrupt event number table, please refer to section 2.4 System Interruption Event

2. Windows Only

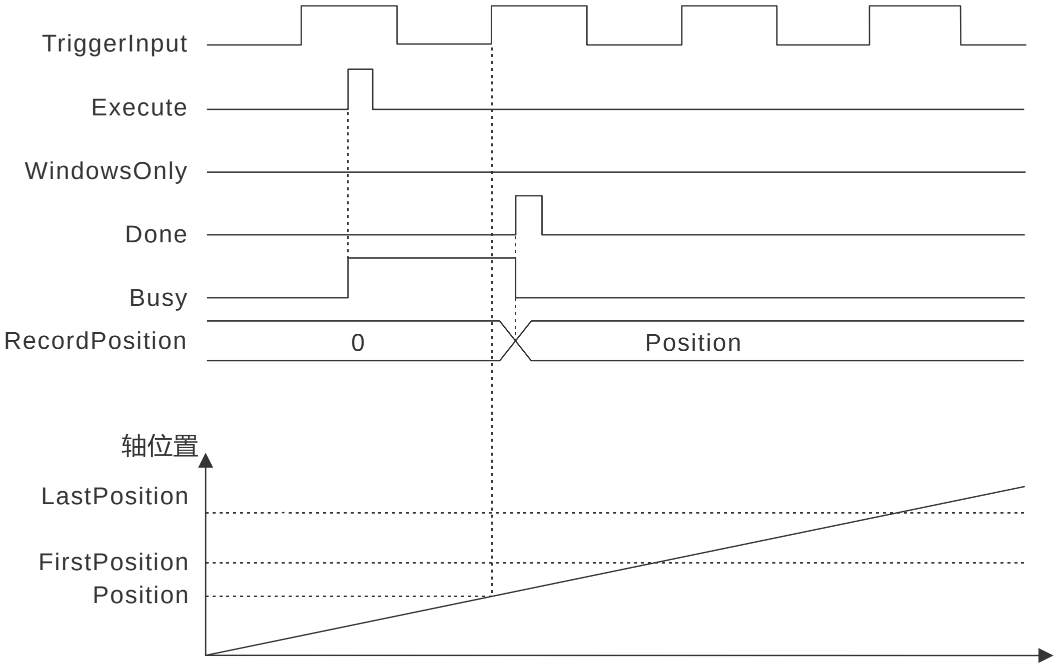
- In Windows Only, specify window 1 as valid and 0 as invalid.
- When disabled is specified, the trigger is detected at all axis positions.
- When Enable is specified, only the axis position detects triggers within the range of FirstPosition and LastPosition.

The moment when Windows Only changes from False to True and the time between the activation of the lock function cannot be locked.

The locking function takes time to activate, so the effective range of Windows Only is extremely short and cannot be locked. The critical value for locking the effective range depends on the performance of the servo drive and encoder input terminals and EtherCAT communication.

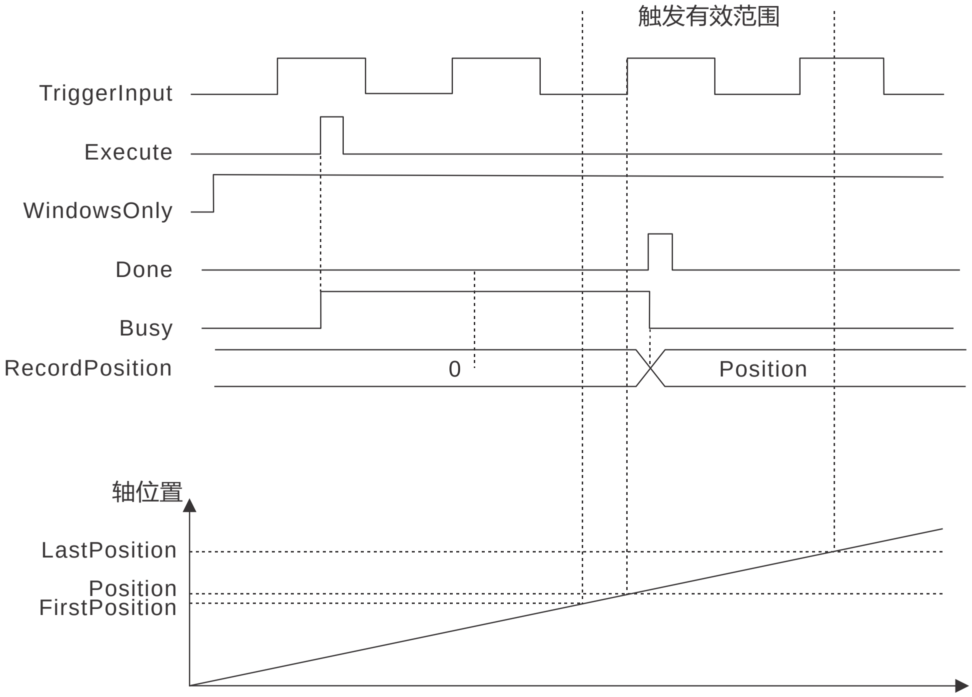
The actions vary depending on the specification of Windows Only, as shown in the sequence diagram below.

- When Windows Only specifies that the window is 0, after changing the Execute condition to TRUE, the initially triggered axis position is output to the Recordposition.



Invalid window timing diagram

- Windows Only specifies that the window is set to 1, and only detects trigger inputs within the window range to obtain axis positions



Window effective timing diagram

The range of FirstPosition and LastPosition under different counting modes

Linear mode

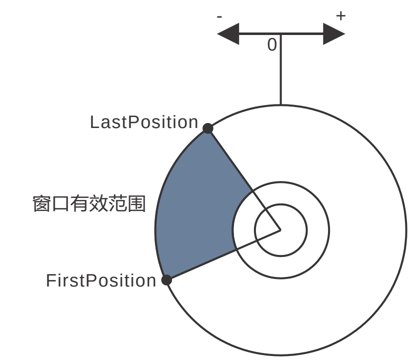
The effective range of the window in linear mode is shown in the following figure:

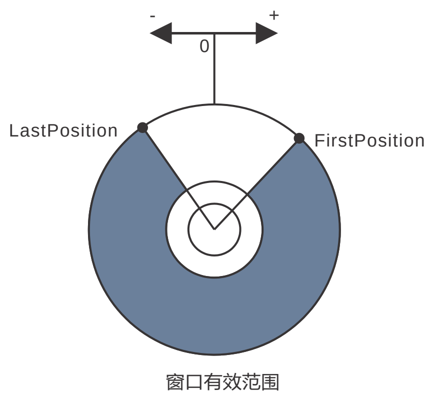
The effective range of the window includes FirstPosition and LastPosition

When FirstPostion>LastPostion, the LastPostion<=Recorded Postion<=firstPostion window is valid.

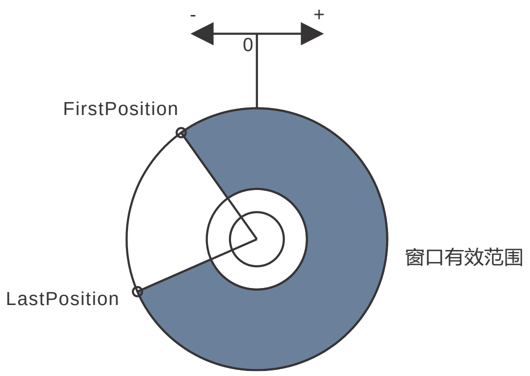
Rotation mode

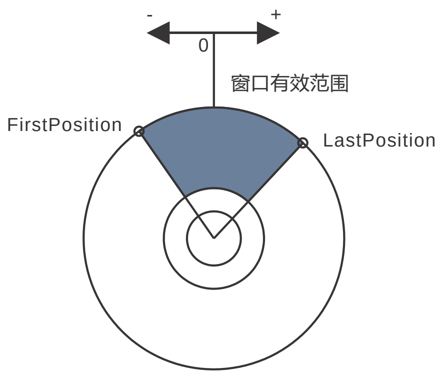
- FirstPosition ≤ LastPosition, or FirstPosition>LastPosition, both can be specified.
- When FirstPosition>LastPosition, the set value crosses the upper and lower limits of the ring counter.
- When the upper and lower limits of the ring counter are exceeded, an exception will occur.
- FirstPosition ≤ LastPosition





FirstPosition>LastPosition





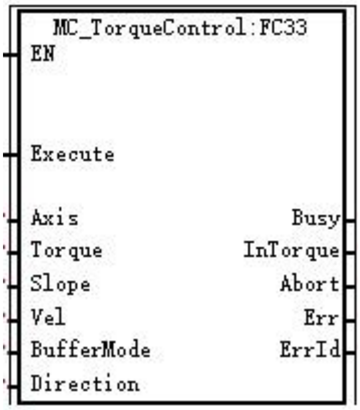
<Note>: When FirstPostion and LastPostion are set beyond the rotation axis modulus range, they will be corrected to within the modulus range.

3. Definition of bit 16 # 60B8 for driver:

Bit position	describe	
0	Probe 1 Enable 0: Probe 1 is not enabled 1: Probe 1 Enable	Bit0~Bit5: Probe 1 setting
one	Probe 1 trigger mode 0: Single trigger, only triggered when the trigger signal is valid for the first time. 1: Continuous triggering	
two	Probe 1 trigger signal selection 0: DI8 input signal 1: Z signal	
three	NA	
four	Probe 1 rising edge enable 0: Rising edge not locked 1: Rising edge latch	
five	Probe 1 descent edge enable 0: Falling edge not locked 1: Falling edge latch	
6~7	NA	

eight	Probe 2 Enable 0: Probe 2 is not enabled 1: Probe 2 Enable	Bit8~Bit13: Probe 2 settings
nine	Probe 2 trigger mode 0: Single trigger, only triggered when the trigger signal is valid for the first time. 1: Continuous triggering	
ten	Probe 2 trigger signal selection 0: DI9 input signal 1: Z signal	
eleven	NA	
twelve	Probe 2 rising edge enable 0: Rising edge not locked 1: Rising edge latch	
thirteen	Probe 2 descent edge enable 0: Falling edge not locked 1: Falling edge latch	
14~15	NA	

MC_TorqueControl torque control command - temporarily not supported



Function: Torque control command, this command controls the standard 402 axis to perform torque motion through torque mode. PDO needs to add three object dictionaries: 0x6071, 0x6060, and 0x6071. This instruction is only used for the communication axis and cannot be used for the pulse axis and virtual axis.

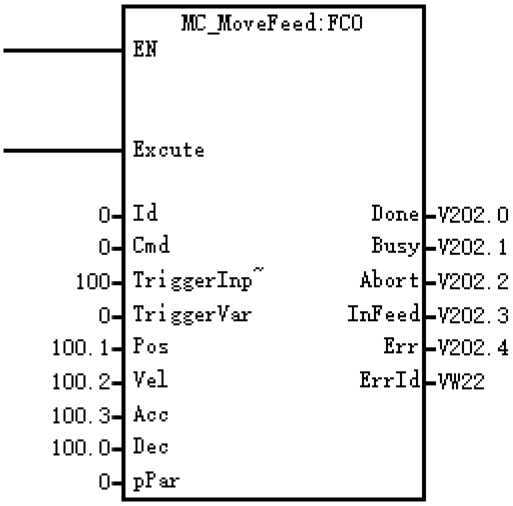
Parameter description:

Parameter Name	Input/Output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
Axis	IN	Axis ID number	BYTE	
Torque	IN	Target torque	REAL	Specify the target torque output to the servo drive in units of "0.1%". Specify the ratio based on the rated torque of "100.0%"
Slope	IN	Torque acceleration	REAL	The torque change rate from the current torque to the target torque, in%/s
Vel	IN	Speed limit	REAL	Limit speed, unit: unit/second
BufferMode	IN	Abort mode	BYTE	0: Abort (directly interrupts the ongoing instruction) Other: Illegal
Direction	IN	direction	INT	1: Positive direction -1: Negative direction
Busy	OUT	Instruction execution bit	BOOL	When the instruction is executed. Busy position is set. When the command completes Busy reset. When the execution condition of the instruction is Off, the Busy bit is reset.

InTorque	OUT	Torque reaching position	BOOL	After reaching the target torque, the InTorque bit is set. When the execution condition of the instruction changes from On to Off, the InTorque bit is reset.
Abort	OUT	Command termination bit	BOOL	When the instruction is terminated during execution, the Abort bit is set. When the execution condition of the instruction is Off, the Abort bit is reset.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

MC_SoveFeed interrupt event triggers motion command

Function Name: MC_SoveFeed



Function: After a specified input interrupt event occurs, the axis performs the action specified by CMD. Before the specified interrupt occurs, Execute becomes 0 to cancel the current instruction. After the exercise is completed, it needs to be triggered again to respond to the interrupt again.

Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
Id		Axis or Axis Group Id	BYTE	
CMD	IN	The action that occurred	BYTE	0: Stop. 1: Relative motion. 2: Absolute movement. 3: Speed control. 16 # 80: Axis group stops
TriggerInput	IN	Interruption event	BYTE	See Appendix 1 below
TriggerVar	IN	Interrupt event parameters	DWORD	
Pos	IN	position	REAL	See Appendix 2 below
Vel	IN	speed	REAL	
Acc	IN	acceleration	REAL	
Dec	IN	deceleration	REAL	

pPar	IN	Other parameter pointers	DWORD	reserve. When the above parameters are insufficient to describe the parameters, place the parameters in the memory indicated by pPar.
Done	IN	Complete position	BOOL	After the axis state is reset to the ready to execute state, the Done bit is set. When the instruction execution condition changes from On to Off, the Done bit resets
Busy	OUT	Instruction execution bit	BOOL	When the instruction is executed. Busy position is set. When the command completes Busy reset. When the execution condition of the instruction is Off, the Busy bit is reset.
Abort	OUT	Command termination bit	BOOL	When the instruction is terminated during execution, the Abort bit is set. When the execution condition of the instruction is Off, the Abort bit is reset.
InFeed	OUT	Exercise position	BOOL	When an interrupt occurs and triggers motion, InFeed is set. When the instruction is completed and enters InFeed, it is reset.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

Appendix 1 Interruption Trigger Events

interrupt number	Interrupt Name	Trigger condition	Special note
0	I0.0 rising edge interruption	I0.0 detected rising edge	M226ES not applicable, any value is invalid
one	I0.0 falling edge interruption	I0.0 detected a falling edge	
two	I0.1 rising edge interruption	I0.1 detected rising edge	
three	I0.1 falling edge interruption	I0.1 detected a falling edge	
four	I0.2 rising edge interruption	I0.2 detected rising edge	
five	I0.2 falling edge interruption	I0.2 detected a falling edge	
six	I0.3 rising edge interruption	I0.3 detected rising edge	
seven	I0.3 falling edge interruption	I0.3 detected a falling edge	
eight	PORT0 receives data	PORT0 received a byte of data	
nine	PORT0 transmission completed	PORT0 has sent the data in the buffer	
ten	Timer Interrupt 0	The timing 0 signal given by FPGA arrives	
eleven	Timer Interrupt 1	The timing 1 signal given by FPGA arrives	
twelve	HSC0 set value interrupt	Count value=set value	
thirteen	HSC1 set value interrupt	Count value=set value	
fourteen	HSC1 direction interruption	Change of counting direction	

fifteen	HSC1 external reset interrupt	External reset signal is valid	
sixteen	HSC2 set value interrupt	Count value=set value	
seventeen	HSC2 direction interruption	Change of counting direction	
eighteen	HSC2 external reset interrupt	External reset signal is valid	
twenty-one	T32	T32 time is up	
twenty-two	T96	T96 time is up	
twenty-three	PORT0 information reception completed	PORT0 completes the reception according to the set conditions	
twenty-four	PORT1 information reception completed	PORT1 completes the reception according to the set conditions	
twenty-five	PORT1 character reception	PORT1 received a byte of data	
twenty-six	PORT1 transmission completed	PORT1 has sent the data from the buffer	
twenty-seven	HSC0 direction interruption	Change of counting direction	
twenty-eight	HSC0 external reset interrupt	External reset signal is valid	
twenty-nine	HSC4 set value interrupt	Count value=set value	
thirty	HSC4 direction interruption	Change of counting direction	
thirty-one	HSC4 external reset interrupt	External reset signal is valid	
thirty-two	HSC3 set value interrupt	Count value=set value	
thirty-three	HSC5 set value interrupt	Count value=set value	
thirty-eight	HSC3 direction interruption	Change of counting direction	
thirty-nine	HSC5 direction interruption	Change of counting direction	
forty	HSC3 external reset interrupt	External reset signal is valid	
forty-one	HSC5 external reset interrupt	External reset signal is valid	
forty-two	HSC0 captures interrupts	External capture signal is effective	
forty-three	HSC1 captures interrupts	External capture signal is effective	
forty-four	HSC2 captures interrupts	External capture signal is effective	
forty-five	HSC3 Capture Interrupt	External capture signal is effective	
forty-six	HSC4 Capture Interrupt	External capture signal is effective	
forty-eight	I0.4 rising edge interruption	I0.4 detected rising edge	
forty-nine	I0.4 falling edge interruption	I0.4 detected a falling edge	
fifty	I0.5 rising edge interruption	I0.5 detected rising edge	
fifty-one	I0.5 falling edge interruption	I0.5 detected a falling edge	
fifty-two	I0.6 rising edge interruption	I0.6 detected rising edge	

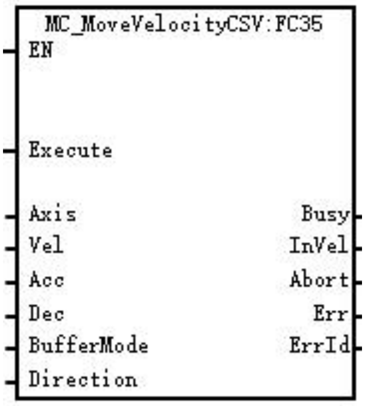
fifty-three	I0.6 falling edge interruption	I0.6 detected a falling edge	
fifty-four	I0.7 rising edge interruption	I0.7 detected rising edge	
fifty-five	I0.7 falling edge interruption	I0.7 detected a falling edge	
fifty-six	I1.0 rising edge interruption	I1.0 detected rising edge	
fifty-seven	I1.0 falling edge interruption	I1.0 detects a falling edge	
fifty-eight	I1.1 Rising edge interruption	I1.1 Detected rising edge	
fifty-nine	I1.1 Downward edge interruption	I1.1 Detected falling edge	
sixty	HSC6 set value interrupt	Count value=set value	
sixty-one	HSC6 direction interruption	Change of counting direction	
sixty-two	HSC6 external reset interrupt	External reset signal is valid	
sixty-three	HSC6 captures interrupts	External capture signal is effective	
sixty-four	HSC7 set value interrupt	Count value=set value	
sixty-five	HSC7 direction interruption	Change of counting direction	
sixty-six	HSC7 external reset interrupt	External reset signal is valid	
sixty-seven	HSC7 captures interrupt	External capture signal is effective	
sixty-eight	HSC8 set value interrupt	Count value=set value	
sixty-nine	HSC8 direction interruption	Change of counting direction	
sevent y	HSC8 external reset interrupt	External reset signal is valid	
sevent y-one	HSC8 Capture Interrupt	External capture signal is effective	
sevent y-two	HSC9 set value interrupt	Count value=set value	
sevent y-three	HSC9 direction interruption	Change of counting direction	
sevent y-four	HSC9 external reset interrupt	External reset signal is valid	
sevent y-five	HSC9 capture interrupt	External capture signal is effective	
one hundre d and twenty -eight	V memory rising edge interrupt	V Memory detected a valid rising edge	TriggerVar=memory offset address * 8+memory bits. If V0.0, this value is 0. The value of V127.7 is 1023 (127 * 8+7).
one hundre d and twenty -nine	V memory falling edge interrupt	V Memory detected a valid lower rising edge	

Appendix 2 Axis Motion

CMD	parameter	Detailed description

0: Stop	Id: Axis ID number Dec: Deceleration	Equivalent to executing the MC_Stop instruction after the interrupt is triggered.
1: Relative motion	Id: Axis ID number Vel: Speed Acc: Acceleration Dec: Deceleration Pos: Distance of Motion	Equivalent to executing MC_SoveRelative after the interrupt is triggered
2: Absolute motion	Id: Axis ID number Vel: Speed Acc: Acceleration Dec: Deceleration Pos: Position When the axis is a rotating axis: VEL>0 direction is positive rotation VEL<0 direction is reverse rotation	Equivalent to executing MC_SoveAbsolute after interrupt triggering
3: Speed motion	Id: Axis ID number Vel: Speed Acc: Acceleration Dec: Deceleration Vel>0.0 positive Vel<0.0 Reverse	Equivalent to executing MC_SoveVelocity after interrupt triggering
16 # 80: Axis group stops	Id: Axis group ID number Dec: Deceleration	Equivalent to executing MC_GroupHalt after interrupt triggering

MC_SoveVelocity CSV Speed Command (CSV Mode) - Currently not supported



Function: This instruction is used to control the terminal execution mechanism to accelerate and decelerate to a given speed and run at a constant speed. When the terminal mechanism reaches the given speed, this instruction is executed, but the terminal mechanism continues to run at this speed.

The function of this command is the same as MC_SoveVelocity, but it uses a periodic synchronous speed mode to control the servo axis. PDO needs to add three object dictionaries, 0x6060, 0x6061, and 0x60ff, which are only used for communication axes.

It should be noted that when calling this instruction, the MC_SoveSuperImosed instruction cannot be called to perform motion overlay actions.

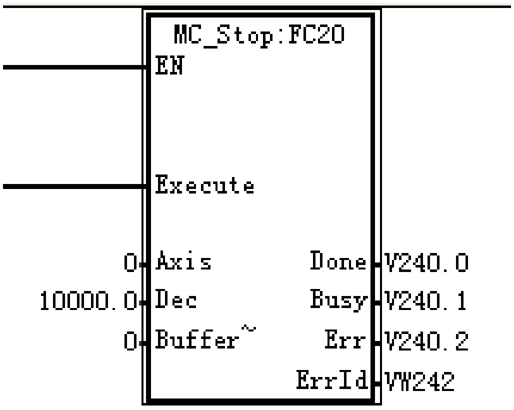
Parameter description:

Parameter Name	Input/Output	Parameter description	type	remarks

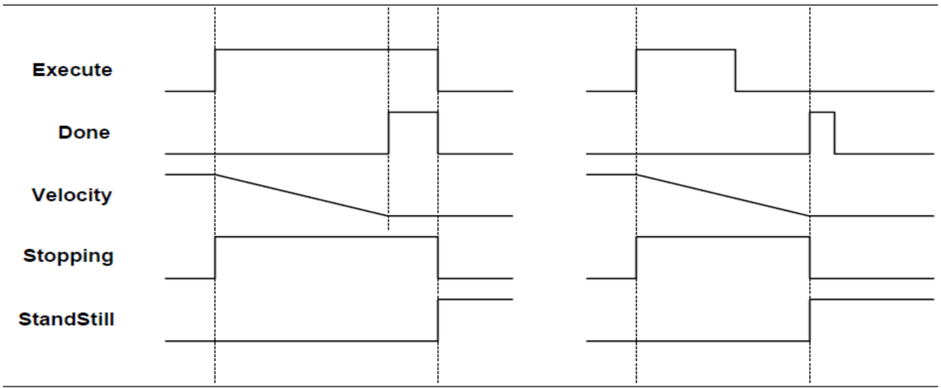
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
Axis	IN	Axis ID number	BYTE	
Vel	IN	speed	REAL	The operating speed of the terminal actuator is always positive. (Unit: Unit/second)
Acc	IN	acceleration	REAL	The acceleration of the terminal actuator is always positive. (Unit: Unit/second/second)
Dec	IN	deceleration	REAL	The deceleration of the terminal actuator is always positive. (Unit: Unit/second/second)
BufferMode	IN	Abort mode	BYTE	0: Abort (directly interrupts the ongoing instruction) Other: Illegal
Direction	IN	direction	INT	1: Positive direction -1: Negative direction
Busy	OUT	Instruction execution bit	BOOL	When the instruction is executed. Busy position is set. When the command completes Busy reset. When the execution condition of the instruction is Off, the Busy bit is reset.
Invel	OUT	Speed reaching position	BOOL	After reaching the target speed, the Invel bit is set. When the execution condition of the instruction changes from On to Off, the Invel bit is reset.
Abort	OUT	Command termination bit	BOOL	When the instruction is terminated during execution, the Abort bit is set. When the execution condition of the instruction is Off, the Abort bit is reset.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

MC_Stop stop command

Function Name: MC_Stop



Function: This instruction is used to control the terminal execution mechanism to decelerate at a given speed until it stops. This instruction cannot be terminated by any other instruction while running. After the speed drops to 0, the axis changes to Stopping state until Execute becomes 0, then it will switch to StandStill.

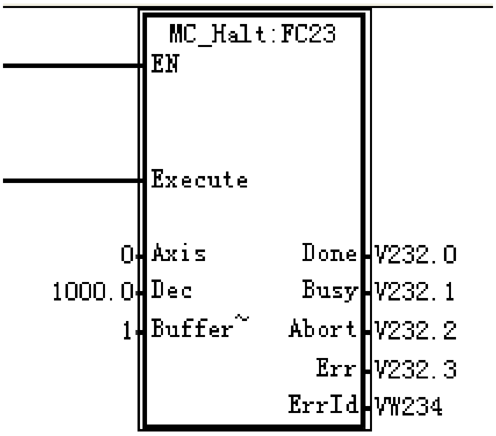


Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
Axis	IN	Axis ID number	BYTE	
Dec	IN	deceleration	REAL	The deceleration of the terminal actuator is always positive. (Unit: Unit/second)
BufferMode	IN	Abort mode	BYTE	0: Abort (directly interrupts the ongoing instruction) Other: Illegal
Done	OUT	Complete position	BOOL	After the speed drops to 0, the Done position is set. When the execution condition of the instruction is Off, the Done bit is reset.
Busy	OUT	Instruction execution bit	BOOL	When the instruction is executed. Busy position is set. When the command completes Busy reset. When the execution condition of the instruction is Off, the Busy bit is reset.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

MC-Halt interruptible stop command

Function Name: MC_Calt



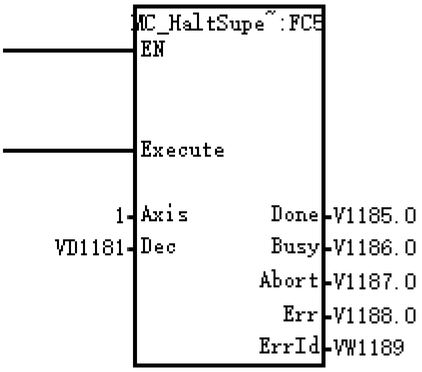
Function: This instruction is used to control the terminal execution mechanism to decelerate at a given speed until it stops. When this instruction runs, it can be terminated by other instructions.

Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks

Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
Axis	IN	Axis ID number	BYTE	
Dec	IN	deceleration	REAL	The deceleration of the terminal actuator is always positive. (Unit: Unit/second/second)
BufferMode	IN	Abort mode	BYTE	0: Abort (directly interrupts the ongoing instruction) Other: Illegal
Done	OUT	Complete position	BOOL	After the speed drops to 0, the Done position is set. When the execution condition of the instruction is Off, the Done bit is reset.
Busy	OUT	Instruction execution bit	BOOL	When the instruction is executed. Busy position is set. When the command completes Busy reset. When the execution condition of the instruction is Off, the Busy bit is reset.
Abort	OUT	Command termination bit	BOOL	When the instruction is terminated during execution, the Abort bit is set. When the execution condition of the instruction is Off, the Abort bit is reset.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

MC-HaltSuperExposed overlay abort instruction - temporarily not supported



Function: This instruction is used to control the terminal execution mechanism to suspend the stacking action without affecting the main motion.

Parameter description:

Parameter Name	Input/Output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
Axis	IN	Axis ID number	BYTE	
Dec	IN	deceleration	REAL	The deceleration of the terminal actuator is always positive. (Unit: Unit/second/second)
Done	OUT	Complete position	BOOL	When the absolute displacement action is completed, the Done bit is set. When the execution condition of the instruction is Off, the Done bit is reset.
Busy	OUT	Instruction execution bit	BOOL	When the instruction is executed. Busy position is set. When the command completes Busy reset. When the execution condition of the instruction is Off, the Busy bit is reset.
Abort	OUT	Command termination bit	BOOL	When the instruction is terminated during execution, the Abort bit is set. When the execution condition of the instruction is Off, the Abort bit is reset.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

6.4.2 Synchronous control instruction

Synchronous control is generally used for electronic gears, electronic camshafts, phase shift commands, etc.

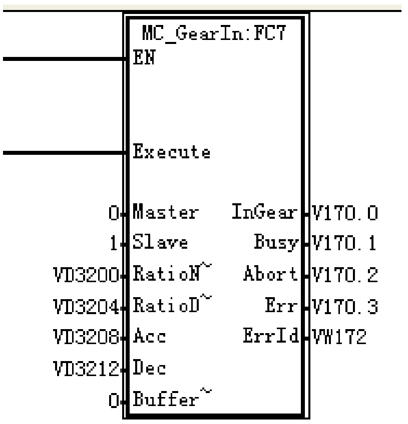
The MC function blocks commonly used in synchronous control instructions are as follows:

Function Name	Instruction Name	M226ES
MC_GearIn	Electronic gear coupling instruction	support
MC_GearOut	Electronic gear disengagement command	support
MC_Phasing	Phase offset instruction	support
MC_CamTableSelect	Cam table selection command	support

MC_CamIn	Electronic cam correlation command	support
MC_CamOut	Release the electronic cam association command	support
MC_MoveSuperImposed	Overlay relative displacement instruction	support
MC_MoveLink	Automatic trapezoidal cam command	Unsupported
MC_GearInPos	Electronic gear coupling instruction	support

MC_GearIn electronic gear coupling command

Function Name: MC_GearIn



Function: This command is used to establish the gear relationship between the master and slave axes. When establishing gear relationships, parameters such as gear ratios can be set. After establishing the gear relationship, the slave shaft follows the main shaft in a given proportional relationship to achieve synchronous control between the master and slave shafts.

After synchronization is completed, the axis speed is calculated as the spindle speed multiplied by RatioNum/RatioDen.

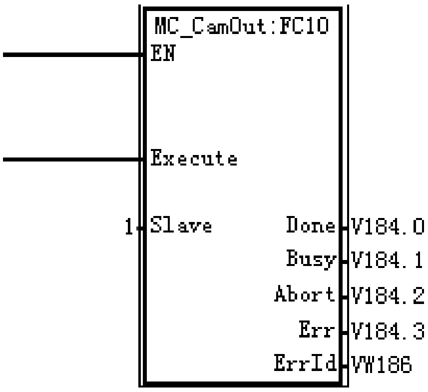
Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
Master	IN	Spindle ID number	BYTE	
Slave	IN	From axis ID number	BYTE	
RatioNum	IN	Molecules of electronic gears	REAL	Not 0
RetioDen	IN	The denominator of electronic gears	REAL	Not 0
Acc	IN	acceleration	REAL	
Dec	IN	deceleration	REAL	
BufferMode	IN	Abort mode	BYTE	0: Abort (directly interrupts the ongoing instruction) Other: Illegal
InGear	OUT	Coupling completion position	BOOL	Coupling completed to set InGear.
Abort	OUT	Command termination bit	BOOL	When the instruction is terminated during execution, the Abort bit is set.

				When the execution condition of the instruction is Off, the Abort bit is reset.
Busy	OUT	Instruction execution bit	BOOL	When the instruction is executed. Busy position is set.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

MC_GearOut electronic gear disengagement command

Function Name: MC_GearOut



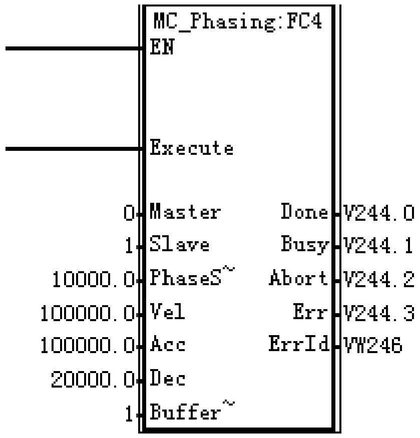
Function: This command is used to release the gear relationship between the master and slave shafts. After the relationship is released, the slave shaft will continue to operate at the speed before detachment.

Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
Slave	IN	From axis ID number	BYTE	
Done	OUT	Complete position	BOOL	When the command to cancel gear association is successfully executed, the Done position is set. When the instruction execution condition is Off, the Done bit is reset.
Abort	OUT	Command termination bit	BOOL	When the instruction is terminated during execution, the Abort bit is set.
Busy	OUT	Instruction execution bit	BOOL	When the instruction is executed. Busy position is set.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

MC-Phasing phase offset instruction

Function Name: MC-Phasing



Function: This command is used to adjust the phase difference between the main axis and the slave axis. After the command is called, the phase difference between the main axis and the slave axis is PhaseShift, that is, the slave axis position=main axis position - PhaseShift.

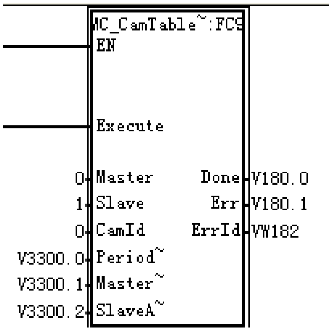
As shown in the figure below, both the master and slave axes move in a 360 cycle, and the Execute signal rises to perform adjustment. After adjustment, the phase deviation between the axis and the spindle is set to the value of PhaseShift.

Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
Master	IN	Spindle ID number	BYTE	
Slave	IN	From axis ID number	BYTE	
PhaseShift	IN	Phase offset	REAL	Phase offset
Vel	IN	speed	REAL	Phase adjustment speed.
Acc	IN	acceleration	BOOL	Phase adjustment acceleration.
Dec	IN	deceleration	BOOL	Phase adjustment deceleration.
BufferMode	IN	Abort mode	BYTE	0: Abort (directly interrupts the ongoing instruction) Other: Illegal
Done	OUT	Phase offset adjustment completed	BOOL	The phase adjustment is completed and Done is set. When the execution condition of the instruction is Off, the Done bit is reset
Busy	OUT	Instruction processing in progress	BOOL	True, instruction is currently being processed
Abort	OUT	Command termination bit	BOOL	When the instruction is terminated during execution, the Abort position bit. When the execution condition of the instruction is Off, the Abort bit is reset.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

MC_CamTableSelect cam table selection command

Function Name: MC_CamTableSelect



Function: This command is used to select the cam curve and specify the mode when associating the master and slave axes.

Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
Master	IN	Spindle ID number	BYTE	
Slave	IN	From axis ID number	BYTE	
CamId	IN	ID number of cam table	BYTE	
Periodic	IN	cycle	BOOL	When this parameter is 1, the slave shaft will periodically execute cam motion. When this parameter is 0, only one cycle of cam motion is executed from the shaft.
MasterAbsolute	IN	Spindle absolute position mode	BOOL	When this parameter is 1, the spindle is in absolute position mode. When this parameter is 0, the spindle is in relative position mode
SlaveAbsolute	IN	Absolute Axis Position Mode	BOOL	When this parameter is 1, the axis is in absolute position mode. When this parameter is 0, the axis is in relative position mode.
Done	OUT	Complete position	BOOL	When the cam parameters are successfully set, the Done position is set. When the instruction execution condition is Off, the Done bit is reset.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

In the basic configuration, the starting address of the cam data is specified. After configuration, a data block will be generated (function not yet implemented), or the starting address (V memory, symbol table) will be specified. The memory block after the starting address stores CAM table data.

When selecting polynomial mode for the cam table, the data is as follows:

Serial number	name	type	notes
one		STRUCT	
two	xStart	REAL	Starting point of X-axis coordinate
three	xEnd	REAL	X-axis coordinate endpoint

four	yStart	REAL	Y-axis coordinate starting point
five	yEnd	REAL	Y-axis coordinate endpoint
six	nElements	WORD	Number of elements
seven	dX0	REAL	The X coordinate of the 0th point
eight	dY0	REAL	The Y coordinate of the 0th point
nine	dV0	REAL	Speed at the 0th point
ten	dA0	REAL	Acceleration at point 0
eleven	dX1	REAL	The first point X coordinate
twelve	dY1	REAL	The Y coordinate of the first point
thirteen	dV1	REAL	The first point speed
fourteen	dA1	REAL	Acceleration at the first point
fifteen	dX2	REAL	The X coordinate of the second point
	...	...	...
		END_STRUCT	

The user can change the cam table data by modifying the data in this memory. After modifying the data, call MC_CamTableSelect to notify the PLC of the change of the cam table data. If the system is in the operation of the circular cam table at this time, the changed data will take effect in the next cam cycle. If it needs to take effect immediately, call the MC_CamIn command immediately after calling MC_CamTableSelect.

Cam offset and scaling ratio:

The position of the spindle input conversion is determined according to the following formula, and the converted X is used as the output of the cam:

Calculation formula: **X=MasterScaling * MasterPosition+MasterOffset**

Therefore, if the ratio of the spindle is greater than 1, the cam will operate at a higher speed. If the ratio value is less than 1,

The speed will decrease accordingly.

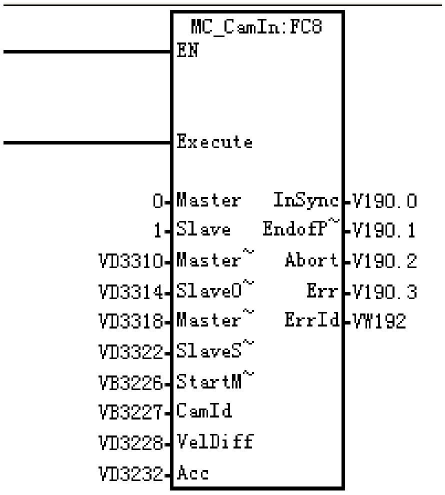
From axis SlaveOffset, SlaveScaling:

Calculation formula: **Y=SlaveScaling * CAM (X)+SlaveOffset**

If SlaveScaling>1 causes the stretching of the cam effect, the range from the shaft will increase. If SlaveScaling<1, it will cause a contraction.

MC_CamIn electronic cam correlation command

Function Name: MC_CamIn



Function: This command is used to establish the cam relationship between the master and slave axes. When establishing the cam relationship, the offset value, scaling ratio, and startup mode of the master and slave axes can be specified according to application requirements. When the instruction is executed, the slave shaft follows the main shaft movement according to the cam curve.

Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
Master	IN	Spindle ID number	BYTE	
Slave	IN	From axis ID number	BYTE	
MasterOffset	IN	Spindle offset	REAL	The position of the spindle cam is offset. Unit: Unit
SlaveOffset	IN	Offset from axis	REAL	Offset from the camshaft position. Unit: Unit
MasterScaling	IN	Spindle scaling	REAL	Spindle scaling configuration parameters. This parameter is not 0
SlaveScaling	IN	Scale from axis	REAL	Configure parameters for axis scaling. This parameter is not 0
StartMode	IN	Startup mode	BYTE	0: Forward jump immediately 1: Acceleration (shortest distance traveled) jump 2: Slope turning forward 3: Slope reversal Note: Forward and reverse rotation are only applicable when the axis is a rotating axis. When the StartMode of the linear axis is 0, 1 is a jump, and 2 and 3 are slope start. The four modes of this parameter are only for the slave station, and whether the slave axis is in absolute or relative mode is configured by the MC_CamTableSelect instruction.
CamId	IN	ID number of cam table	BYTE	
VelDiff	IN	Cam table coupling speed	REAL	
Acc	IN	Cam gauge acceleration	REAL	
InSync	OUT	Establishment of cam relationship	BOOL	After establishing a relationship between the main axis and the slave axis, InSync is set, When the execution condition of the instruction is Off, the InSync bit is reset.

EndOfProfile	OUT	End of cam relationship	BOOL	If the Perioc parameter is 0 when the MC_CamTableSelect instruction is executed, EndOfProfile is set after the cam curve is executed once. The Perioc parameter is 1. After the cam curve is executed once, EndOfProfile is set and reset after one cycle. When the instruction execution condition is Off, the EndOfProfile bit is reset.
Abort	OUT	Command termination bit	BOOL	When the instruction is terminated during execution, the Abort bit is set. When the execution condition of the instruction is Off, the Abort bit is reset.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

Cam offset and scaling ratio:

The position of the spindle input conversion is determined according to the following formula, and the converted X is used as the output of the cam:

X=MasterScaling*MasterPosition+MasterOffset

Therefore, if the ratio of the spindle is greater than 1, the cam will operate at a higher speed, and if the ratio value is less than 1, the speed will decrease accordingly.

From axis SlaveOffset, SlaveScaling:

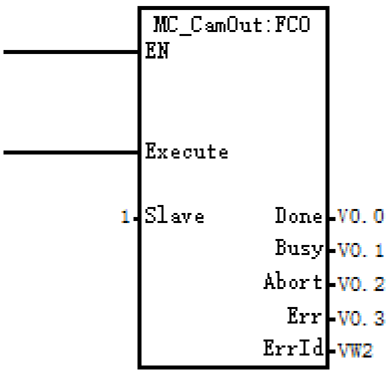
Calculation formula:

Y=SlaveScaling*CAM(X)+SlaveOffset

If SlaveScaling>1 causes the stretching of the cam effect, the range from the shaft will increase. If SlaveScaling<1, it will cause a contraction.

MC_CamOut releases electronic cam association command

Function Name: MC_CamOut



Function: This command is used to release the cam relationship between the master and slave shafts. After the relationship is released, the slave shaft will continue to move at the speed before detachment.

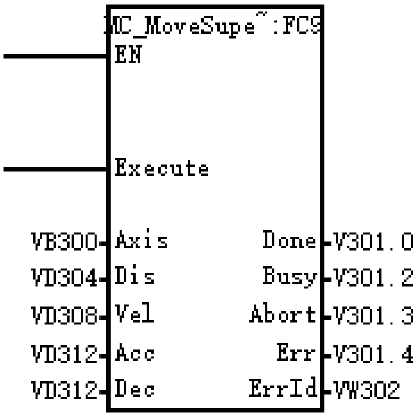
Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
Slave	IN	From axis ID number	BYTE	
Done	OUT	Complete position	BOOL	When the cancel cam association command is successfully executed, the Done position is set.

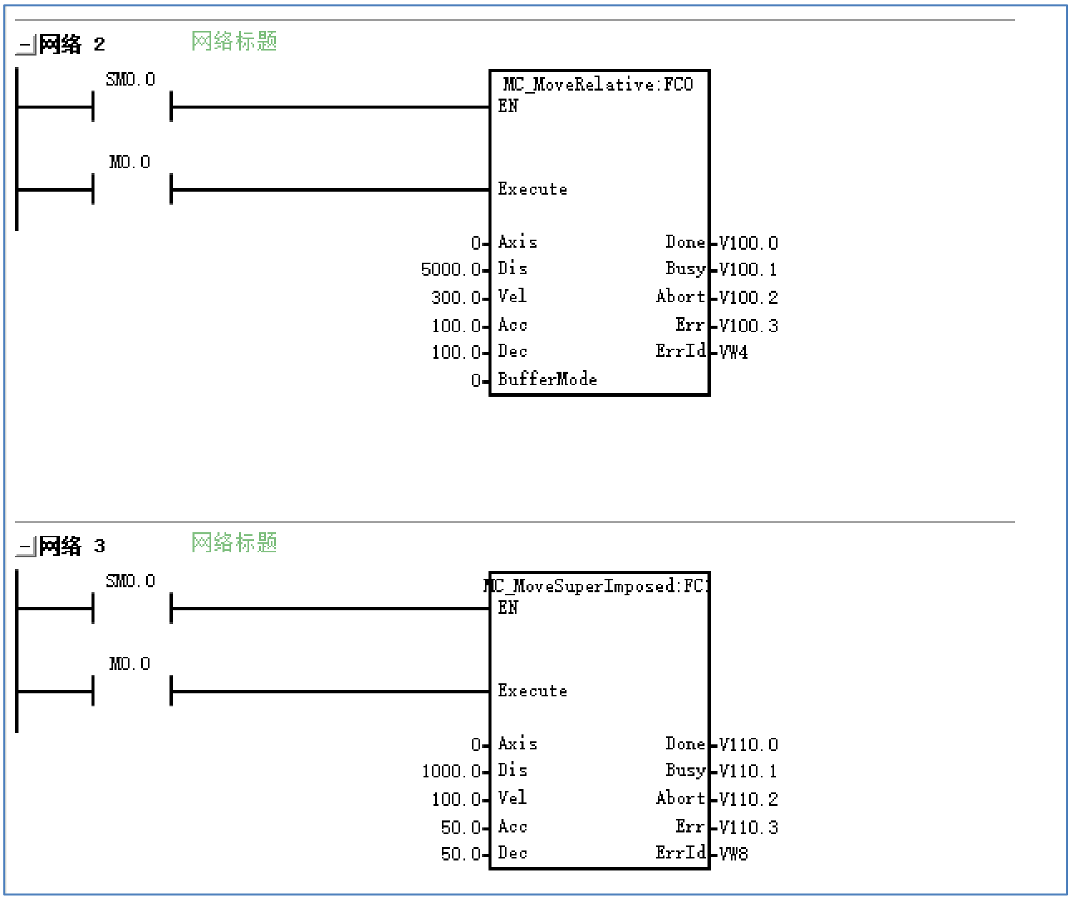
				When the instruction execution condition is Off, the Done bit is reset.
Abort	OUT	Command termination bit	BOOL	When the instruction is terminated during execution, the Abort bit is set.
Busy	OUT	Instruction execution bit	BOOL	When the instruction is executed. Busy position is set.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

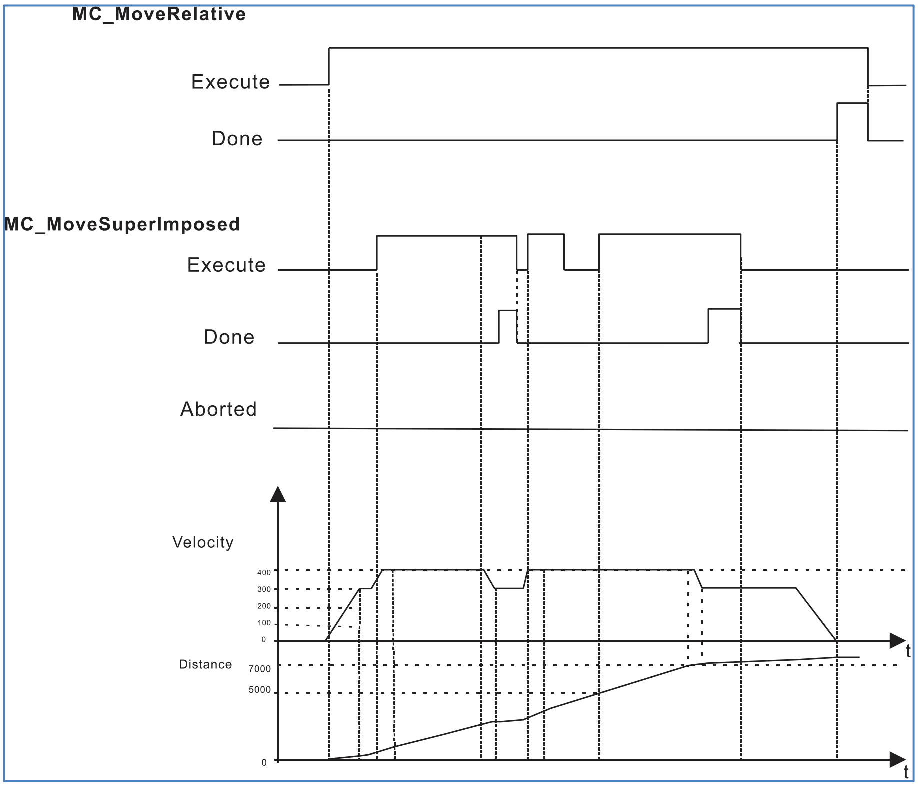
MC-MoveSuperExposed overlay relative displacement instruction

Function name: MC_SoveSuperExposed



Function: This command is used to control the terminal execution mechanism to independently add a given distance during acceleration and deceleration at a given speed in the current motion state. When this command is executed, it will not terminate the execution of the previous command. Both commands will be executed simultaneously, and distance, speed, acceleration, and deceleration will be superimposed in real time. When one of the instructions is executed, its speed, acceleration, and deceleration will no longer be superimposed, and the other instruction will still run independently. Here are some examples:



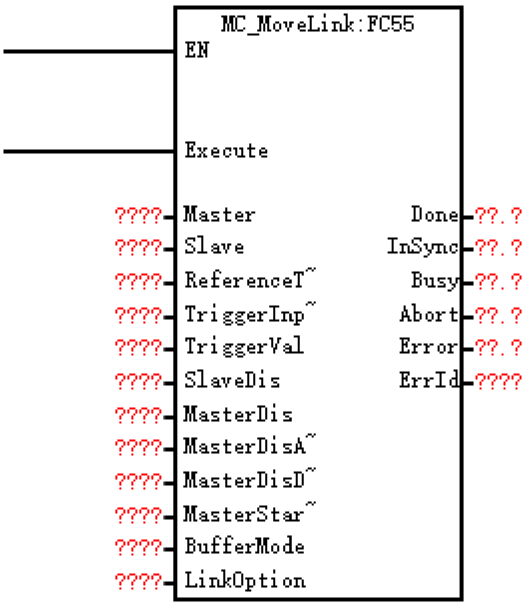


Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
Axis	IN	Axis ID number	BYTE	

Dis	IN	distance	REAL	Based on the current position, the target distance that the terminal actuator needs to move is set to a negative value, indicating that the axis will be reversed. Unit: Unit.
Vel	IN	speed	REAL	The operating speed of the terminal actuator is always positive. (Unit: Unit/second)
Acc	IN	acceleration	REAL	The acceleration of the terminal actuator is always positive. (Unit: Unit/second/second)
Dec	IN	deceleration	REAL	The deceleration of the terminal actuator is always positive. (Unit: Unit/second/second)
Done	OUT	Complete position	BOOL	When the absolute displacement action is completed, the Done bit is set. When the execution condition of the instruction is Off, the Done bit is reset.
Busy	OUT	Instruction execution bit	BOOL	When the instruction is executed. Busy position is set. When the command completes Busy reset. When the execution condition of the instruction is Off, the Busy bit is reset.
Abort	OUT	Command termination bit	BOOL	When the instruction is terminated during execution, the Abort bit is set. When the execution condition of the instruction is Off, the Abort bit is reset.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

MC_SoveLink (Automatic Trapezoidal Cam Command) - Not currently supported



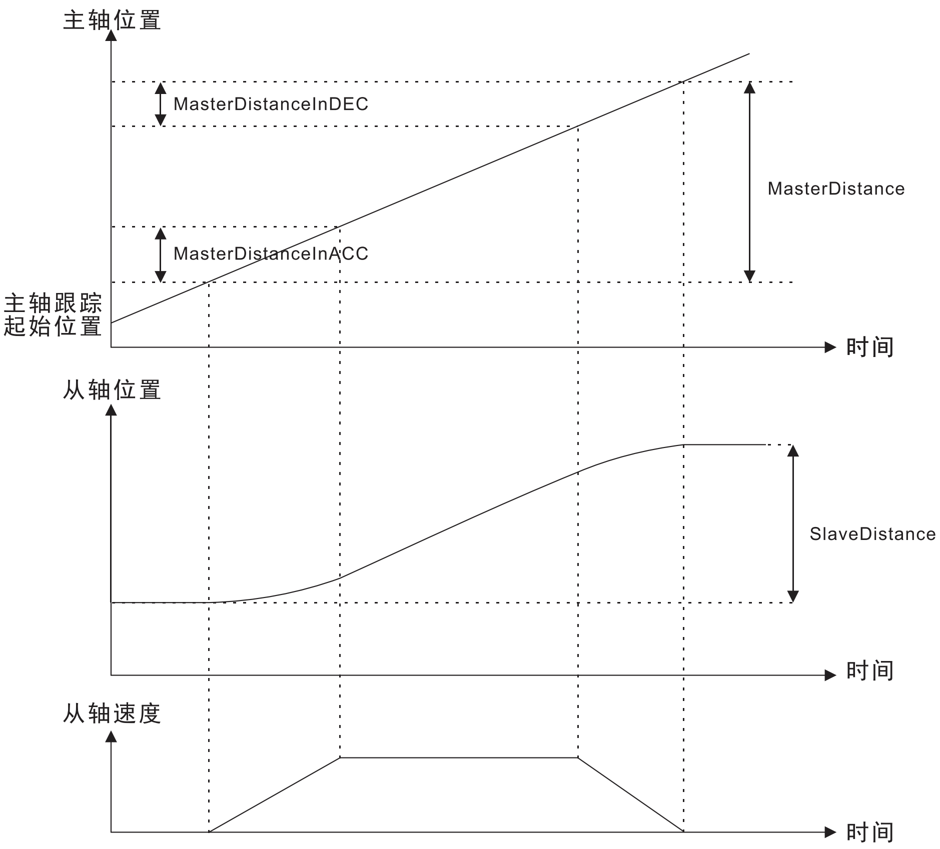
Function: This command is used for custom cam motion with adjustable acceleration and deceleration stages.

Parameter description:

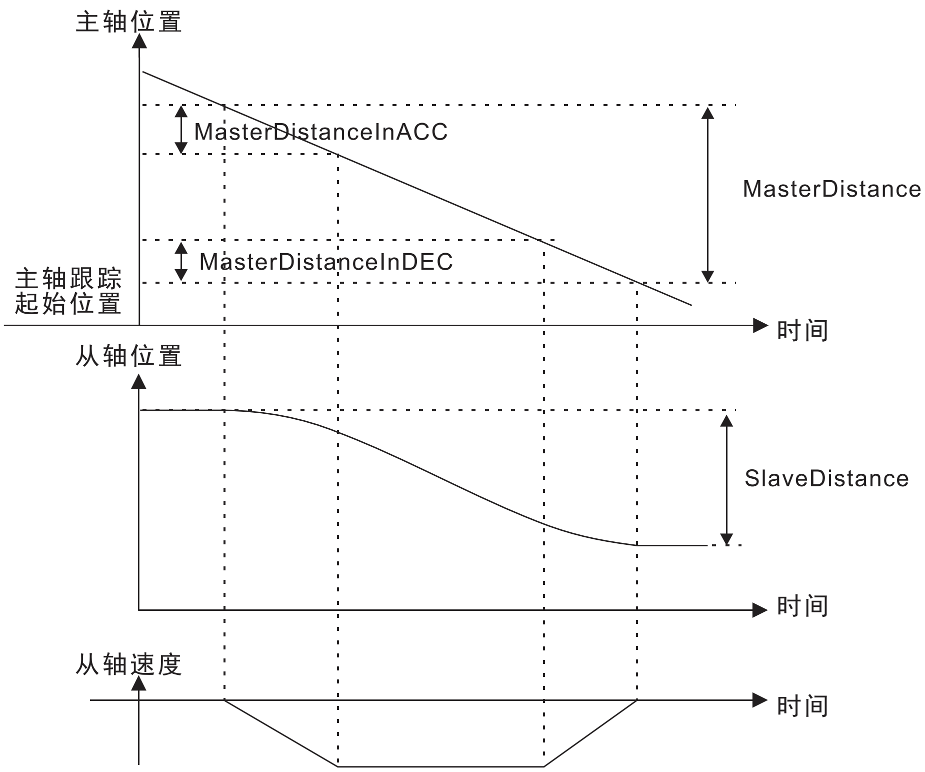
Parameter Name	Input/ Output	Parameter description	type	remarks

Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
Master	IN	Spindle ID number	BYTE	
Slave	IN	From axis ID number	BYTE	
ReferenceType	IN	Location type choice	BYTE	reserve.
TriggerInput	IN	Interruption event	BYTE	Refer to Appendix 1 for Interrupt Trigger Events
TriggerVar	IN	Interrupt event parameters	DWORD	
SlaveDis	IN	Move from axis distance	REAL	Specify the distance of movement from the axis. Unit: Instruction unit. Both positive and negative directions of movement are valid. Unit: Instruction unit.
MasterDis	IN	Main axis movement distance	REAL	Main axis movement distance, unit: command unit
MasterDisAcc	IN	Spindle acceleration movement distance	REAL	Spindle acceleration movement distance, unit: command unit
MasterDisDec	IN	Spindle deceleration movement distance	REAL	Spindle deceleration movement distance, unit: command unit
MasterStartDis	IN	Spindle tracking distance	REAL	Specify the spindle position that tracks the absolute coordinates of the spindle starting from the axis. Unit: Instruction unit.
BufferMode	IN	Abort mode	BYTE	0: Abort (directly interrupts the ongoing instruction) Other: Illegal
LinkOption	IN	Synchronization start condition selection	DWORD	Specify the conditions for synchronizing from axis to main axis. Bit0-Bit7 0: At startup 1: When a trigger is detected 2: When the spindle reaches the spindle tracking distance Bit8: Periodic
InSync	OUT	Establishment of cam relationship	BOOL	After establishing a relationship between the main axis and the slave axis, InSync is set, After the execution of the instruction is completed, insync is reset.
Busy	OUT	Instruction execution bit	BOOL	When the instruction is executed. Busy position is set. After the execution of the instruction is completed, Busy is reset.
Done	OUT	End of cam relationship	BOOL	The Perioc parameter is set to 0, and when the cam curve is executed once, Done is set. The Perioc parameter is 1. After the cam curve is executed once, Done is set and reset after one cycle. When the instruction execution condition is Off, the Done bit is reset.
Abort	OUT	Command termination bit	BOOL	When the instruction is terminated during execution, the Abort bit is set. When the execution condition of the instruction is Off, the Abort bit is reset.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

When the spindle moves in the positive direction:



When the spindle moves in the negative direction:



The relationship between the main axis and the amount of movement from the sub axis is as follows.

accelerate	**principal axis**	**Spindle acceleration movement distance**
	From the axis	从轴移动距离 X $\dfrac{\dfrac{\text{主轴加速移动距离}}{2}}{\dfrac{\text{主轴加速移动距离}}{2}+(\text{主轴移动距离}-\text{主轴加速移动距离}-\text{主轴减速移动距离})+\dfrac{\text{主轴减速移动距离}}{2}}$

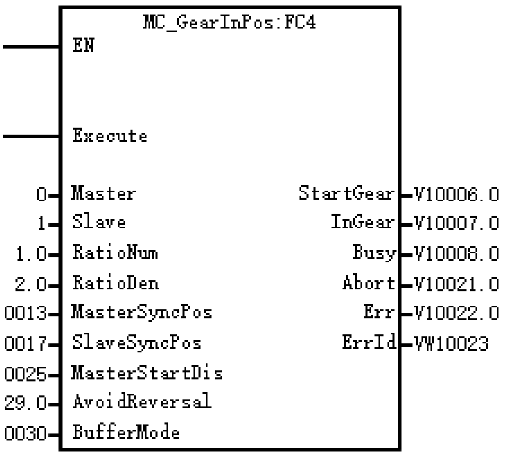
Constant speed	principal axis	Spindle movement distance - Spindle acceleration movement distance - Spindle deceleration movement distance
	From the axis	Off axis movement distance - Off axis movement distance during the acceleration of the main axis mentioned above - Off axis movement distance during the deceleration mentioned below
Slow down	principal axis	Spindle deceleration movement distance
	From the axis	$\text{从轴移动距离} \times \dfrac{\frac{\text{主轴减速移动距离}}{2}}{\frac{\text{主轴加速移动距离}}{2} + (\text{主轴移动距离-主轴加速移动距离-主轴减速移动距离}) + \frac{\text{主轴减速移动距离}}{2}}$

When the constant speed range of the spindle is negative, an abnormal constant speed movement occurs and the action stops.

In addition, when conveying at the same speed as the main shaft, please specify the amount of movement from the shaft as follows:

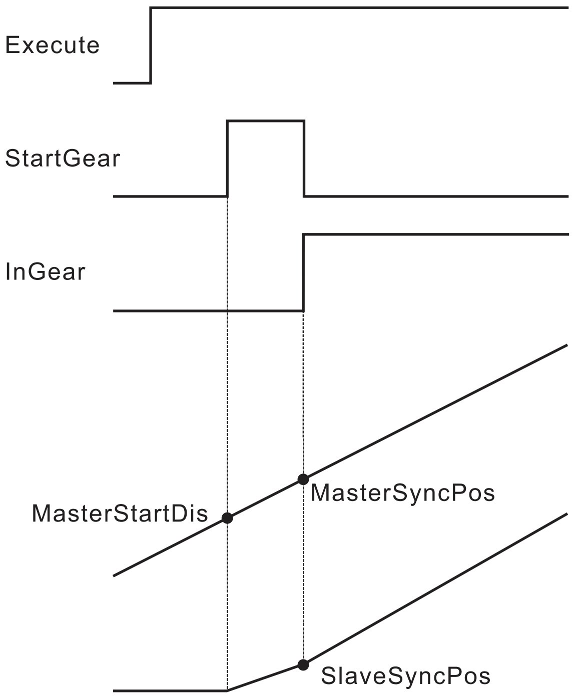
$$\text{从轴移动距离} = \frac{\text{主轴加速移动距离}}{2} + (\text{主轴移动距离-主轴加速移动距离-主轴减速移动距离}) + \frac{\text{主轴减速移动距离}}{2}$$

MC_GearInPhos (Electronic Gear Coupling Instruction)



Function: This command is used to establish the gear relationship between the master and slave axes. When establishing gear relationships, parameters such as gear ratios can be set. After establishing the gear relationship, the slave shaft follows the main shaft in a given proportional relationship to achieve synchronous control between the master and slave shafts.

After synchronization is completed, the axis speed is calculated as the spindle speed multiplied by RatioNum/RatioDen.



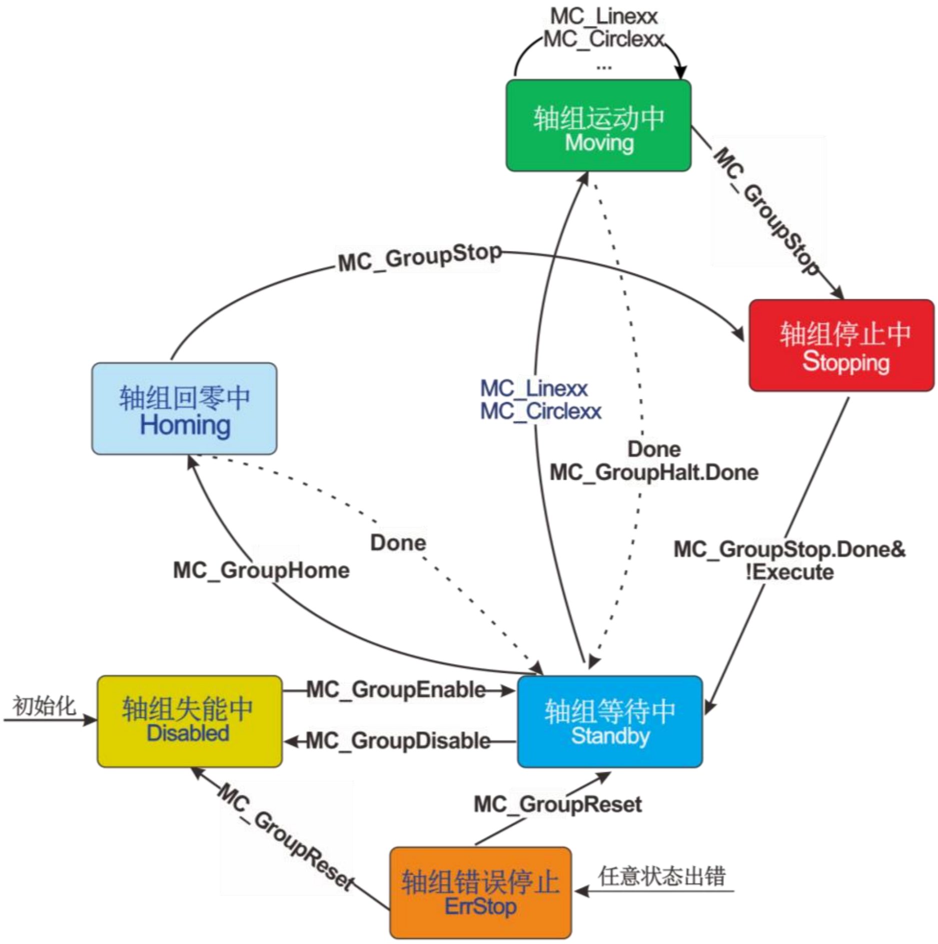
Parameter description:

Parameter Name	Input/ Output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
Master	IN	Spindle ID number	BYTE	
Slave	IN	From axis ID number	BYTE	
RatioNum	IN	Molecules of electronic gears	REAL	Not 0
RetioDen	IN	The denominator of electronic gears	REAL	Not 0
MasterSyncPos	IN	Spindle synchronization position	REAL	
SlaveSyncPos	IN	Synchronize position from axis	REAL	
MasterStartDis	IN	The spindle starts synchronizing distance	REAL	Calculate a smooth curve from the axis based on the MasterStartDis, MasterSyncPos, and SlaveSyncPos values to synchronize the axis with the main gear during SlaveSyncPos. Curve Range [MasterSyncPos MasterStartDis, MasterSyncPos]
AvoidRerverSal	IN	Do not reverse	BOOL	Set to False, if leading from the physical position of the axis Perform a reversal. The linear axis cannot be set to TRUE. If it is set to TRUE, it only applies to the rotation axis. If the linear axis is set to True, an error will be reported on the rising edge of execute (error id 85) For the rotation axis, if there is a possibility of reversal according to the parameter settings, try to avoid reversal by stretching one cycle from the axis. If reversal cannot be avoided after stretching, the axis will stop incorrectly (error ID 84)

BufferMode	IN	Abort mode	BYTE	0: Abort (directly interrupts the ongoing instruction) Other: Illegal
StartGear	IN	Start coupling processing	BOOL	TRUE, start electronic gear coupling processing
InGear	OUT	Coupling completion position	BOOL	Coupling completed to set InGear.
Abort	OUT	Command termination bit	BOOL	When the instruction is terminated during execution, the Abort bit is set. When the execution condition of the instruction is Off, the Abort bit is reset.
Busy	OUT	Instruction execution bit	BOOL	When the instruction is executed. Busy position is set.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

6.4.3 Axis group instruction

The motion control instructions for axis groups are generally used for multi axis linear interpolation, arc interpolation instructions, etc. The flowchart of the instruction status of the axis group is shown in the following figure:



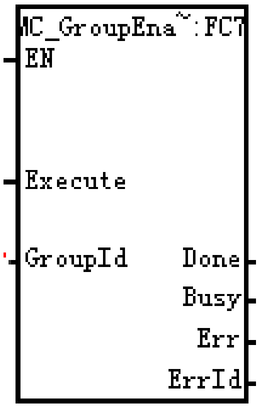
	Tips The MC_GroupHome, GroupStop and other functions in the figure have not been implemented yet. Therefore, the two axis configurations of Homing and Stopping are temporarily retained.

The MC function blocks commonly used in axis group instructions are as follows:

Function Name	Instruction Name	M226ES
MC_GroupEnable	Effective instructions for axis group	support
MC_GroupDisable	Invalid instruction for axis group	support
MC_GroupReset	Axis group error reset command	support
MC_MoveLinerRelative	Relative displacement linear interpolation instruction	support
MC_MoveLinerAbsolute	Absolute displacement linear interpolation instruction	support
MC_MoveCircularRalative	Relative displacement arc interpolation instruction	support
MC_MoveCircularAbolute	Absolute displacement arc interpolation instruction	support
MC_GroupHalt	Axis group stop command	support
MC_GroupReadState	Read axis group status command	support
MC_Helical	Spiral interpolation instruction	support
MC_GroupReset	Axis group error reset command	support

MC_GroupEnable axis group valid command

Function Name: MC_GroupEnable



Function: Make the shaft group effective. The axis group status has changed from GroupDisalbed to GroupStandby.

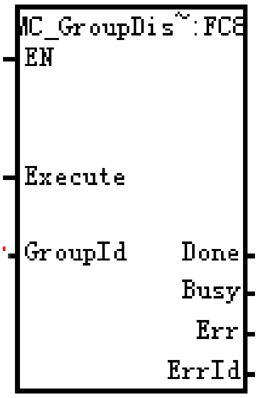
Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
GroupId	IN	Axis group ID number	BYTE	Axis group number
Busy	OUT	Busy position	BOOL	Instruction is executing as TRUE, execution completed as FLASE
Done	OUT	Status bit	BOOL	After the instruction is executed successfully, the Done bit is set.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.

ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

MC_GroupDisable axis group invalid command

Function Name: MC_GroupDisable



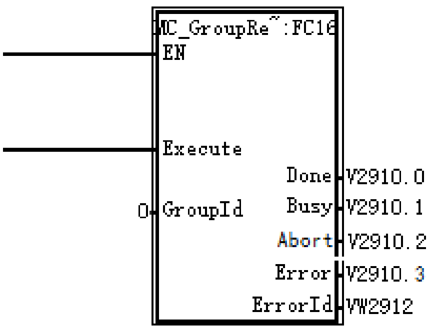
Function: Invalid axis group. Change the axis group status from GroupStandby to GroupDisalbed.

Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
GroupId	IN	Axis group ID number	BYTE	Axis group number
Done	OUT	Status bit	BOOL	After the instruction is executed successfully, the Done bit is set.
Busy	OUT	Busy position	BOOL	Instruction is executing as TRUE, execution completed as FLASE
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

MC_GroupReset axis group error reset command

Function Name: MC_GroupReset



Function: When the axis group is in ErrStop state, this command is called to clear errors in the axis group and axis, and switch the axis group state to Disabled or Standby.

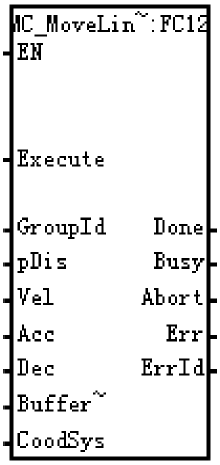
Parameter description:

Parameter	Input/ou	Parameter	type	remarks

Name	tput	description		
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
GroupId	IN	Axis group ID number	BYTE	Axis group number
Done	IN	Instruction completion bit	BOOL	Complete position
Busy	OUT	Instruction execution bit	BOOL	When the instruction is executed. Busy position is set. When the command completes Busy reset. When the execution condition of the instruction is Off, the Busy bit is reset.
Abort	OUT	Command termination bit	BOOL	Keep the interface, it will not appear
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

MC-MoveLinerRelative Relative Displacement Linear Interpolation Instruction

Function Name: MC_SoveLinerRelative



Function: This command performs linear interpolation based on relative displacement

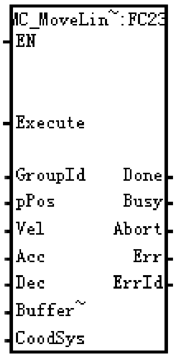
Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
GroupId	IN	Axis group ID number	BYTE	
pDis	IN	Relative position pointer	DWORD	Pointer to describe the distance to the target. The target distance is described as data of 6 REAL types. Representative (spatial displacement under MCS) X： REAL; Y： REAL; Z： REAL; A： REAL; B： REAL; C： REAL;

				Or (displacement of each axis under ACS) A0：REAL; A1：REAL; A2：REAL; A3：REAL; A4：REAL; A5：REAL; (Unit: Unit)
Vel	IN	speed	REAL	The operating speed of the terminal actuator is always positive. (Unit: Unit/second)
Acc	IN	acceleration	REAL	The acceleration of the terminal actuator is always positive. (Unit: Unit/second/second)
Dec	IN	deceleration	REAL	The deceleration of the terminal actuator is always positive. (Unit: Unit/second/second)
BufferMode	IN	Abort mode	BYTE	0: Abort (directly interrupts the ongoing command, disables transitions) 1: Buffered (the function block that starts buffering after the previous deceleration is completed) 2: BendingDevious (walks to the end of the previous segment at the current speed and starts executing the next segment at the same speed as the previous segment, disabling transitions) 16 # 12: TMCorner (additional angle transition, transitioning to the next section with an additional angle when starting deceleration in the previous section). For details, please refer to the specific introduction of BufferMode continuous interpolation.
CoordSystem	IN	Coordinate system	BYTE	0：ACS 1: MCS (currently only available in this type) 2：WCS 3：PCS_1 4：PCS_2
Done	OUT	Complete position	BOOL	When the absolute displacement action is completed, the Done bit is set. When the execution condition of the instruction is Off, the Done bit is reset.
Abort	OUT	Command termination bit	BOOL	When the instruction is terminated during execution, the Abort bit is set. When the execution condition of the instruction is Off, the Abort bit is reset.
Busy	OUT	Instruction execution bit	BOOL	When the instruction is executed. Busy position is set. When the command completes Busy reset. When the execution condition of the instruction is Off, the Busy bit is reset.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

MC-MoveLinerAbsolute Absolute Displacement Linear Interpolation Instruction

Function name: MC_SoveLinerAbsolute



Function: This command performs linear interpolation based on absolute displacement

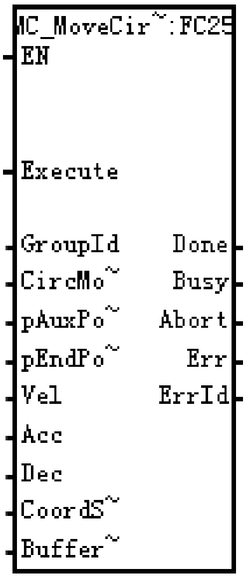
Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
GroupId	IN	Axis group ID number	BYTE	
pPos	IN	Absolute position pointer	DWORD	Pointer to describe the target location. The target location is described as data of 6 REAL types. Representative (spatial displacement under MCS) X：REAL； Y：REAL； Z：REAL； A：REAL； (Reserved) B：REAL； (Reserved) C：REAL； (Reserved) Or (displacement of each axis under ACS) A0：REAL； A1：REAL； A2：REAL； (Reserved for both axes) A3：REAL； (Reserved) A4：REAL； (Reserved) A5：REAL； (Reserved) (Unit: Unit)
Vel	IN	speed	REAL	The operating speed of the terminal actuator is always positive. (Unit: Unit/second)
Acc	IN	acceleration	REAL	The acceleration of the terminal actuator is always positive. (Unit: Unit/second/second)
Dec	IN	deceleration	REAL	The deceleration of the terminal actuator is always positive. (Unit: Unit/second/second)
BufferMode	IN	Abort mode	BYTE	0: Abort (directly interrupts the ongoing command, disables transitions) 1: Buffered (the function block that starts buffering after the previous deceleration is completed) 2: BendingDevious (walks to the end of the previous segment at the current speed and starts executing the next segment at the same speed as the previous segment, disabling transitions) 16 # 12: TMCorner (additional angle transition, transitioning to the next section with an additional angle when starting deceleration in the previous section). For details, please refer to the specific introduction of BufferMode continuous interpolation.

CoordSystem	IN	Coordinate system	BYTE	0: ACS 1: MCS (currently only available in this type) 2: WCS 3: PCS_1 4: PCS_2
Done	OUT	Complete position	BOOL	When the absolute displacement action is completed, the Done bit is set. When the execution condition of the instruction is Off, the Done bit is reset.
Abort	OUT	Command termination bit	BOOL	When the instruction is terminated during execution, the Abort bit is set. When the execution condition of the instruction is Off, the Abort bit is reset.
Busy	OUT	Instruction execution bit	BOOL	When the instruction is executed. Busy position is set. When the command completes Busy reset. When the execution condition of the instruction is Off, the Busy bit is reset.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

MC-MoveCircularRallative Relative Displacement Arc Interpolation Instruction

Function name: MC_SoveCircularRallative



Function: This command performs circular interpolation based on relative displacement.

Parameter description:

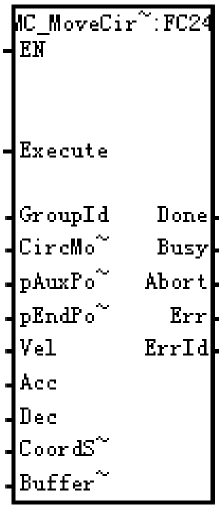
Parameter Name	Input/output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
GroupId	IN	Axis group ID number	BYTE	
CircMode	IN	Circular interpolation method	BYTE	Bit0~Bit3 0: Three point formula 1: Center point style. 8: Three-point full circle 9: Center point full circle Bit4: Interpolation direction (effective for center point

				and full circle) 0: Reverse (CCW) 1: Forward rotation (CW) The direction of the arc is selected according to the right-hand rule: the CCW direction is the direction along which the finger bends. When using the center point method, if the distance between the center point and the starting and ending points is not exactly the same (usually due to limited accuracy in programming the center point), it is projected onto the perpendicular bisector of the starting and ending points. Once the center point is too far away (more than 1% of the radius), an error will be returned.
pAuxPoint	IN	Reference position pointer	DWORD	Pointer to describe the reference position. In the three-point equation, pAuxPostion refers to the reference point passed through. When expressed as a center point, pAuxPostion refers to the center of the circle. The reference location is described as data of 6 REAL types. Representative (spatial displacement under MCS) X：REAL Y：REAL Z：REAL A: REAL (reserved) B: REAL (reserved) C: REAL (reserved) (Unit: Unit)
pEndPoint	IN	Target position pointer	DWORD	Pointer to describe the target location. The target location is described as data of 6 REAL types. Representative (spatial displacement under MCS) X：REAL Y：REAL Z：REAL A: REAL (reserved) B: REAL (reserved) C: REAL (reserved)
Vel	IN	speed	REAL	The operating speed of the terminal actuator is always positive. (Unit: Unit/second)
Acc	IN	acceleration	REAL	The acceleration of the terminal actuator is always positive. (Unit: Unit/second/second)
Dec	IN	deceleration	REAL	The deceleration of the terminal actuator is always positive. (Unit: Unit/second/second)
BufferMode	IN	Abort mode	BYTE	0: Abort (directly interrupts the ongoing command, disables transitions) 1: Buffered (the function block that starts buffering after the previous deceleration is completed) 2: BendingDevious (walks to the end of the previous segment at the current speed and starts executing the next segment at the same speed as the previous segment, disabling transitions) 16 # 12: TMCorner (additional angle transition, transitioning to the next section with an additional angle when starting deceleration in the previous

				section). For details, please refer to the specific introduction of BufferMode continuous interpolation.
CoordSystem	IN	Coordinate system	BYTE	0: ACS 1: MCS (currently only available in this type) 2: WCS 3: PCS_1 4: PCS_2
Done	OUT	Complete position	BOOL	When the absolute displacement action is completed, the Done bit is set. When the execution condition of the instruction is Off, the Done bit is reset.
Abort	OUT	Command termination bit	BOOL	When the instruction is terminated during execution, the Abort bit is set. When the execution condition of the instruction is Off, the Abort bit is reset.
Busy	OUT	Instruction execution bit	BOOL	When the instruction is executed. Busy position is set. When the command completes Busy reset. When the execution condition of the instruction is Off, the Busy bit is reset.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

MC-MoveCircularaAbolute Absolute Displacement Arc Interpolation Instruction

Function name: MC_SoveCircularaAbolute



Function: This command performs arc interpolation based on absolute displacement.

Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
GroupId	IN	Axis group ID number	BYTE	
CircMode	IN	Circular interpolation method	BYTE	Bit0-Bit3 0: Three point formula 1: Center point style. 8: Three-point full circle 9: Center point full circle Bit4: Interpolation direction (center point type is effective) 0: Reverse (CCW) 1: Forward rotation (CW) The direction of the arc is selected according to the right-hand rule: the CCW direction is the direction along which the finger bends. When it is a center point type, If the distance between the center point and the starting and ending points is not exactly the same (usually due to limited accuracy in programming the center point), project it onto the perpendicular bisector of the starting and ending points. Once the center point is too far away (more than 1% of the radius), an error will be returned.
pAuxPoint	IN	Reference position pointer	DWORD	Pointer to describe the reference position. When the three-point equation is used, pAuxPostion refers to the reference point passed through. When expressed as a center point, pAuxPostion refers to the center of the circle. The reference position distance is described as 6 types of REAL data. Representative (under MCS, PCS, WCS) X：REAL Y：REAL Z：REAL A: REAL (reserved) B: REAL (reserved) C: REAL (reserved) (Unit: Unit)
pEndPoint	IN	End position pointer	DWORD	Pointer to the end position of the description. The end position distance is described as 6 types of REAL data. X：REAL Y：REAL Z：REAL A: REAL (reserved) B: REAL (reserved) C: REAL (Reserved) (Unit: Unit)
Vel	IN	speed	REAL	The operating speed of the terminal actuator is always positive. (Unit: Unit/second)
Acc	IN	acceleration	REAL	The acceleration of the terminal actuator is always positive. (Unit: Unit/second)
Dec	IN	deceleration	REAL	The deceleration of the terminal actuator is always positive.

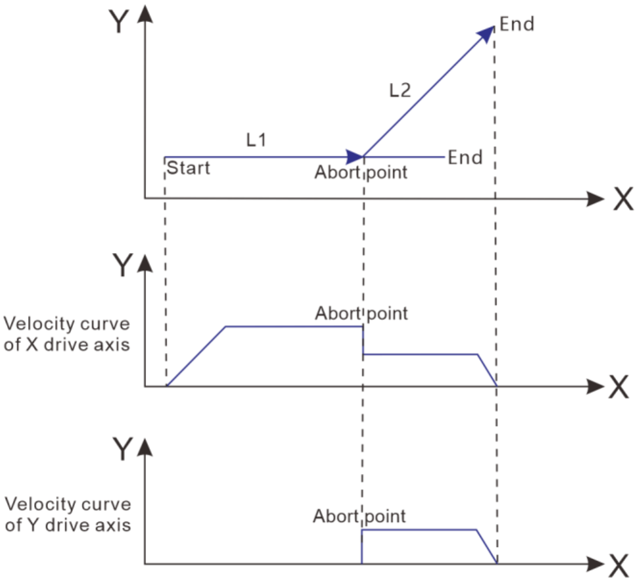
				(Unit: Unit/second)
BufferMode	IN	Abort mode	BYTE	0: Abort (directly interrupts the ongoing command, disables transitions) 1: Buffered (the function block that starts buffering after the previous deceleration is completed) 2: BendingDevious (walks to the end of the previous segment at the current speed and starts executing the next segment at the same speed as the previous segment, disabling transitions) 16 # 12: TMCorner (additional angle transition, transitioning to the next section with an additional angle when starting deceleration in the previous section). For details, please refer to the specific introduction of BufferMode continuous interpolation.
CoordSystem	IN	Coordinate system	BYTE	0：ACS； 1: MCS (currently only available). 2：WCS 3：PCS_1； 4：PCS_2
Done	OUT	Complete position	BOOL	When the absolute displacement action is completed, the Done bit is set. When the execution condition of the instruction is Off, the Done bit is reset.
Abort	OUT	Command termination bit	BOOL	When the instruction is terminated during execution, the Abort bit is set. When the execution condition of the instruction is Off, the Abort bit is reset.
Busy	OUT	Instruction execution bit	BOOL	When the instruction is executed. Busy position is set. When the command completes Busy reset. When the execution condition of the instruction is Off, the Busy bit is reset.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

Specific Introduction to BufferMode Continuous Interpolation

Value taking	describe	detailed
0	Abort	Directly interrupt the ongoing command and disable transitions
one	Buffered	The previous deceleration has been completed, and the instruction to execute buffering has begun
two	BendingPrevious	Go from the previous speed to the end of the previous speed and start executing the next speed according to the previous speed, disable transition
16#12	TMCorner	Additional angle transition, transitioning from the previous section to the next section with an additional angle when performing deceleration

Abort

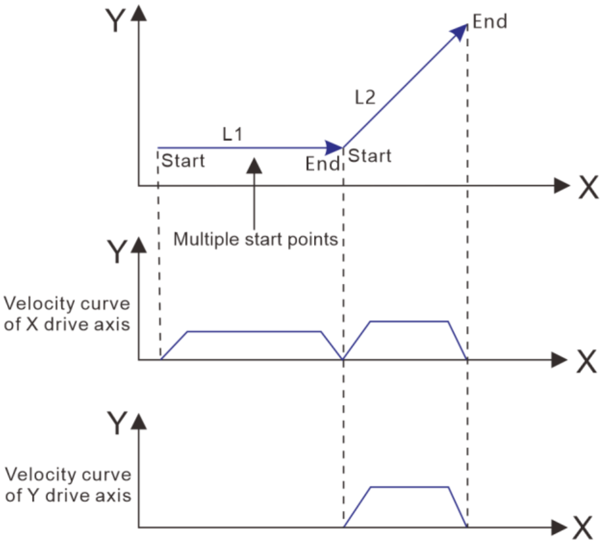
Line L1 executes the first interpolation instruction first, and triggers the second interpolation instruction before L1 finishes. If the buffer mode of the second interpolation instruction is set to "Abort", the second interpolation instruction immediately interrupts the first interpolation instruction and starts executing a new interpolation curve.



Buffered

line L1 first executes the first interpolation instruction, and triggers the second interpolation instruction before L1 completes. If the buffering mode of the second interpolation instruction is set to "Buffered", the interpolator will continue to execute the first interpolation instruction.

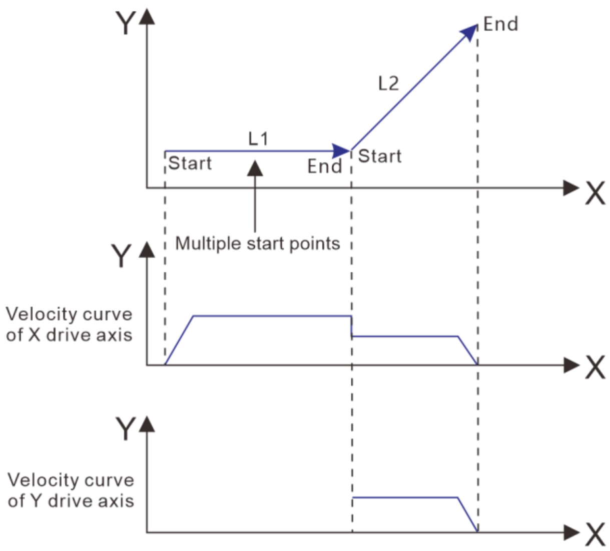
After the first interpolation instruction is executed and the Done signal is valid, the second interpolation instruction begins to execute, as shown in the following figure:



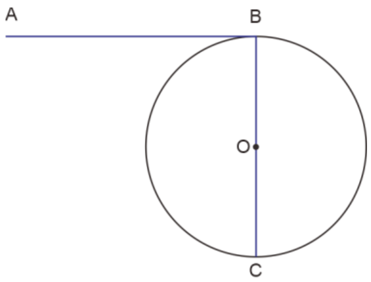
BendingPrevious

Line L1 executes the first interpolation instruction first, and triggers the second interpolation instruction before L1 completes. If the buffer mode of the second interpolation instruction is set to "BendingDevious", the interpolator will attempt to maintain the target speed of the first instruction and walk a complete straight line.

After the first interpolation instruction is executed and the Done signal output is valid, the second interpolation instruction will begin to execute. The switching point will maintain the same speed, and the speed distribution of the coordinate axis will be reassigned. As shown in the following figure:

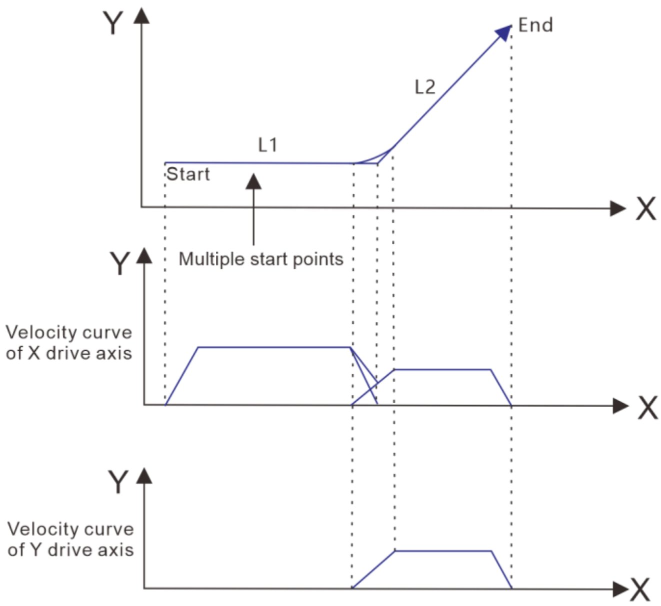


This mode is suitable for switching between straight lines and arcs, with the straight line located on the tangent of the arc. Using this mode can maintain the velocity continuity of the interpolation curve.



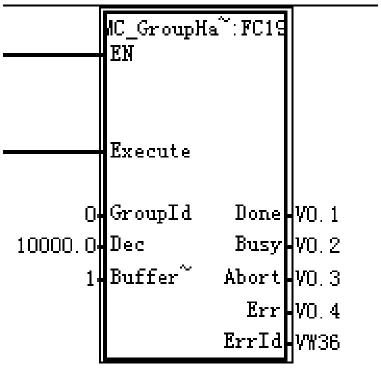
TMCorner

Line L1 first executes the first interpolation instruction and triggers the second interpolation instruction before L1 completes its journey. If the buffer mode of the second interpolation instruction is set to "TMCorner", when the interpolator detects that the first straight line L1 starts to decelerate, it will activate the second interpolation instruction and transition to the second interpolation in an additional angle manner, maintaining continuous velocity in all directions, as shown in the following figure:



MC_GroupHalt axis group stop command

Function Name: MC_GroupHalt



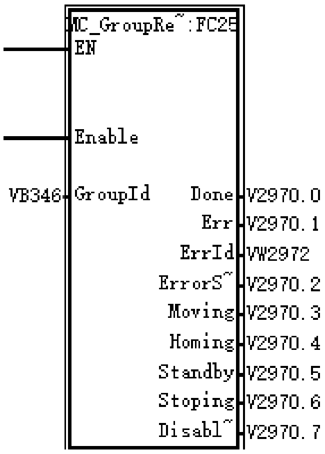
Function: Decelerate and stop all axes in the interpolation action.

Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
GroupId	IN	Axis group ID number	BYTE	Axis group number
Dec	IN	deceleration	REAL	The deceleration of the terminal actuator is always positive. (Unit: Unit/second), when set to 0, achieve emergency stop
BufferMode	IN	Abort mode	BYTE	0: Abort (directly interrupts the ongoing instruction) Other: Illegal
Done	OUT	Complete position	BOOL	When the instruction execution is completed, the Done bit is set. When the execution condition of the instruction is Off, the Done bit is reset.
Abort	OUT	Stop position	BOOL	When the instruction is terminated during execution, the Abort bit is set. When the execution condition of the instruction is Off, the Abort bit is reset.
Busy	OUT	Instruction execution bit	BOOL	When the instruction is executed. Busy position is set. When the command completes Busy reset. When the execution condition of the instruction is Off, the Busy bit is reset.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

MC_GroupReadState reads axis group status instruction

Function Name: MC_GroupReadState

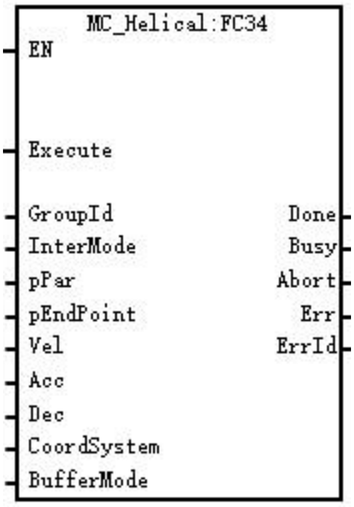


Function: This command is used to read the values of the corresponding axis group status parameters.

Parameter description:

Parameter Name	Input/output	Parameter description	type	remarks
Enable	IN	Instruction execution conditions	BOOL	When On, execute the instruction.
GroupId	IN	Axis group ID number	BYTE	Axis group number
Done	OUT	Status bit	BOOL	After the instruction is executed successfully, the Done bit is set.
Err	OUT	Error position	BOOL	If an error is detected, the Err bit is set. When the execution condition of the instruction changes from On to Off, the Err bit is reset.
ErrId	OUT	Error code	WORD	When Err is TRUE, ErrId is the error of the instruction. When ErrStop is TRUE, ErrId is the current error in the axis group. Error codes, see 6.4.4 error code.
ErrStop	OUT	Status indicator bit	BOOL	True, if the axis group is in the ErrStop state
Stopping	OUT	Status indicator bit	BOOL	True, if the axis group is in the state of Stopping
Moving	OUT	Status indicator bit	BOOL	True, if the axis group is in the moving state
Standby	OUT	Status indicator bit	BOOL	True, if the axis group is in the Standby state
Homing	OUT	Status indicator bit	BOOL	True, if the axis group is in the Homing state
Disabled	OUT	Status indicator bit	BOOL	True, if the axis group is in the disabled state

MC-Helical Spiral Interpolation Instruction

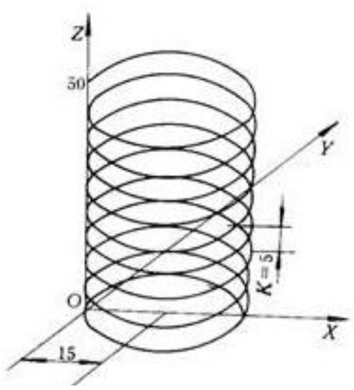


Function: This command performs interpolation in a spiral pattern.

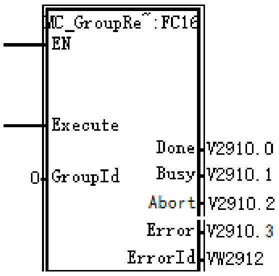
Parameter description:

Parameter Name	Input/Output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
GroupId	IN	Axis group ID number	BYTE	
InterMode	IN	Circular interpolation method	BYTE	0: Inverse circle of xy plane, relative position, ignoring the XY coordinates of the endpoint. 1: XY plane follows a circle, relative position, ignoring the XY coordinates of the endpoint. 0x10: Error reported when the XY coordinates of the endpoint are not on the spiral line, relative to the reverse circle of the xy plane. 0x11: Error reported when the XY coordinates of the endpoint are not on the spiral line due to the fact that the xy plane is circular and in relative position.
pPar	IN	Parameter pointer	DWORD	Pointer to parameter description PPar refers to the parameters of spiral interpolation, described as 6 REAL types 0: The horizontal coordinate of the center of the circle on the arc plane 1: The vertical coordinate of the center of the circle on the circular arc plane 2: The lead of a spiral. The distance that a spiral line moves along the axis of a cylinder by rotating it once. 3. 4-5: Reserved (Unit: Unit)
pEndPoint	IN	Target position pointer	DWORD	Pointer to describe the target location. The target location is described as data of 6 REAL types. Representative (spatial displacement under MCS (relative)) X： REAL Y： REAL Z： REAL A: REAL (reserved) B: REAL (reserved) C: REAL (reserved)
Vel	IN	speed	REAL	The operating speed of the terminal actuator is always

				positive. (Unit: Unit/second)
Acc	IN	acceleration	REAL	The acceleration of the terminal actuator is always positive. (Unit: Unit/second/second)
Dec	IN	deceleration	REAL	The deceleration of the terminal actuator is always positive. (Unit: Unit/second/second)
BufferMode	IN	Abort mode	BYTE	0: Abort (directly interrupts the ongoing command, disables transitions) 1: Buffered (the function block that starts buffering after the previous deceleration is completed) 2: BendingDevious (walks to the end of the previous segment at the current speed and starts executing the next segment at the same speed as the previous segment, disabling transitions) 16 # 12: TMCorner (additional angle transition, transitioning to the next section with an additional angle when performing deceleration in the previous section)
CoordSystem	IN	Coordinate system	BYTE	0：ACS 1: MCS (currently only available in this type) 2：WCS 3：PCS_1 4：PCS_2
Done	OUT	Complete position	BOOL	When the action execution is completed, the Done bit is set. When the execution condition of the instruction is Off, the Done bit is reset.
Abort	OUT	Command termination bit	BOOL	When the instruction is terminated during execution, the Abort bit is set. When the execution condition of the instruction is Off, the Abort bit is reset.
Busy	OUT	Instruction execution bit	BOOL	When the instruction is executed. Busy position is set. When the command completes Busy reset. When the execution condition of the instruction is Off, the Busy bit is reset.
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.



MC_GroupReset axis group error reset command



Function: When the axis group is in ErrStop state, this command is called to clear errors in the axis group and axis, and switch the axis group state to Disabled or Standby.

Parameter description:

Parameter Name	Input/Output	Parameter description	type	remarks
Execute	IN	Instruction execution conditions	BOOL	When the execution condition changes from Off to On, execute the instruction.
GroupId	IN	Axis group ID number	BYTE	Axis group number
Done	IN	Instruction completion bit	BOOL	Complete position
Busy	OUT	Instruction execution bit	BOOL	When the instruction is executed. Busy position is set. When the command completes Busy reset. When the execution condition of the instruction is Off, the Busy bit is reset.
Abort	OUT	Command termination bit	BOOL	Keep the interface, it will not appear
Err	OUT	Error position	BOOL	If an error is detected, the Error bit is set. When the execution condition of the instruction is Off, the Error bit is reset.
ErrId	OUT	error code	WORD	Error codes, see 6.4.4 error code.

6.4.4 error code

Error code description:

Error code	explain	Error code	explain
0	freedom from error	46	Overspeed operation (running at a speed exceeding the maximum speed)
1	Axis number value error	50	Communication error
2	Axis status error	51	Internal servo error
3	Command loss	52	There are no errors that need to be cleared from MC-Reset
4	Axis busy	53	Interruption of the motion process enables
5	The speed value is too small (less than the minimum speed)	60	RetioNumerator value error
6	The speed value is too high (greater than the maximum speed)	61	RetioMenominator value error
7	The acceleration value is too small	65	Cam table and MC_TableSelect are inconsistent
8	The acceleration value is too large	66	Cam table ID error (or not created)

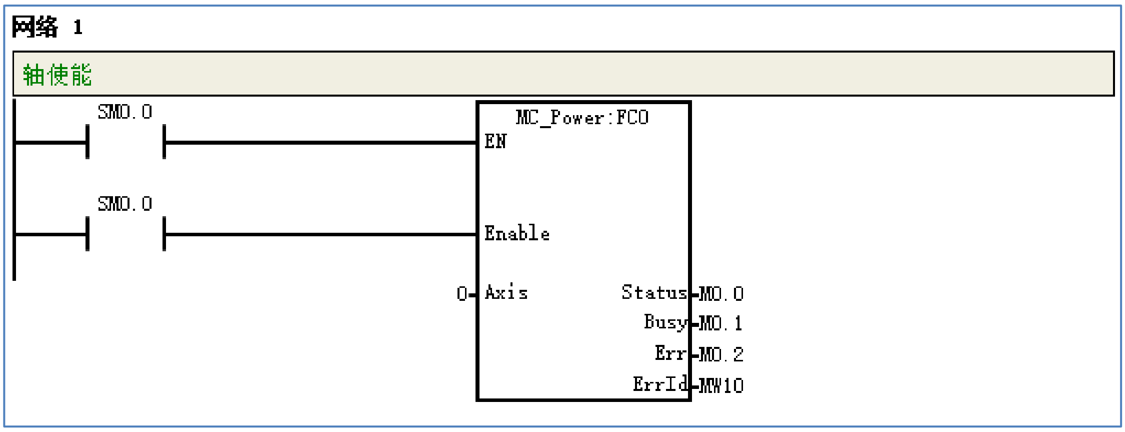
9	The deceleration value is too small	67	Cam table data error
10	The deceleration value is too large	68	MasterScaling value error
11	Wrong direction value	69	SlaveScaling value error
12	HSP axis value error	70	StartMode value error
13	HSP module number error	80	Reaching the left limit switch
15	Axis not ready	81	Reaching the right limit switch
16	Reset error failed	82	Reaching the negative software limit
17	GroupId error	83	Reaching the forward software limit
19	Distance pointer error	90	Cmd value error
20	Buffer Mode value error	91	TriggerInput value error
21	CoordSys value error	92	TriggerVar value error
22	Internal software error	93	MC_Seed execution failed
23	Memory request error	94	MC_Seed instruction is executing
24	CircMode setting error	95	There is no such interrupt in the module, or the interrupt is not enabled
25	Three points on the arc form a straight line	100	Axis errors in the axis group
26	The distance between two points and the center of the circle is not equal	101	Axis group status setting error
30	Homing mode lacks Z-phase configuration	102	A shaft in the shaft group is already under the control of another shaft group
31	Homing mode lacks forward switch configuration	1000	This configuration will only be supported in the future
32	Homing mode lacks negative switch configuration	2001~ 2022	Module error
33	Homing mode lacks origin switch configuration	3000	There are errors in the configuration
34	Homing mode with incorrect value	1600	No resources available
35	Internal servo return error	1601	Event interrupt ID not supported
43	Position parameter set too large	1602	Error in trimming parameters
44	Position parameter set too small	1603	Flying shear parameter error
45	The spindle coordinates are out of range	1605	Reset angle does not match synchronization area

6.5 Example of Electronic Cam Program

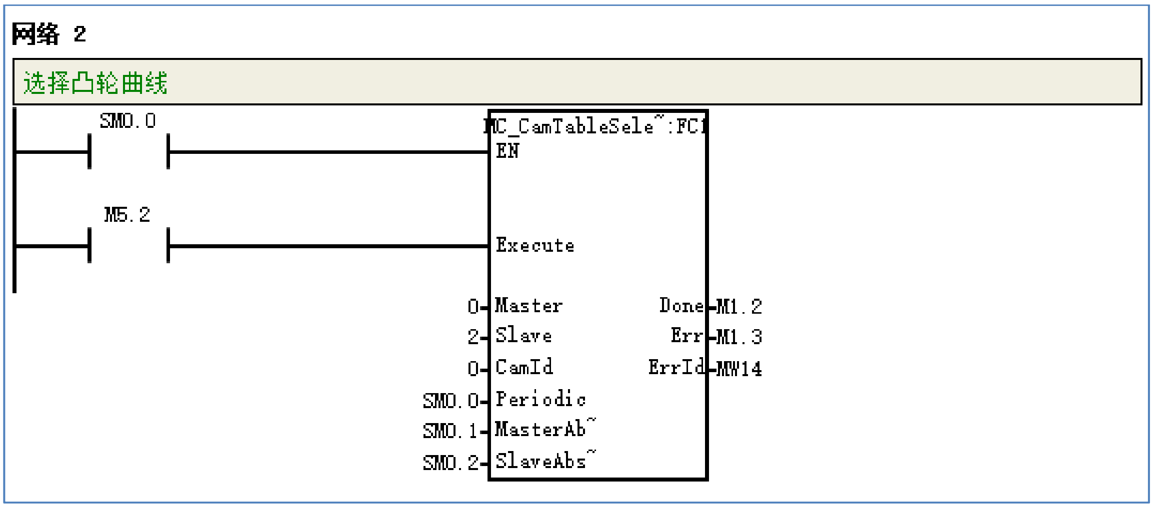
The following program is an example of the CPU using the axis instruction ct_plcopentlib to implement electronic cam function.

Network diagram of program implementation

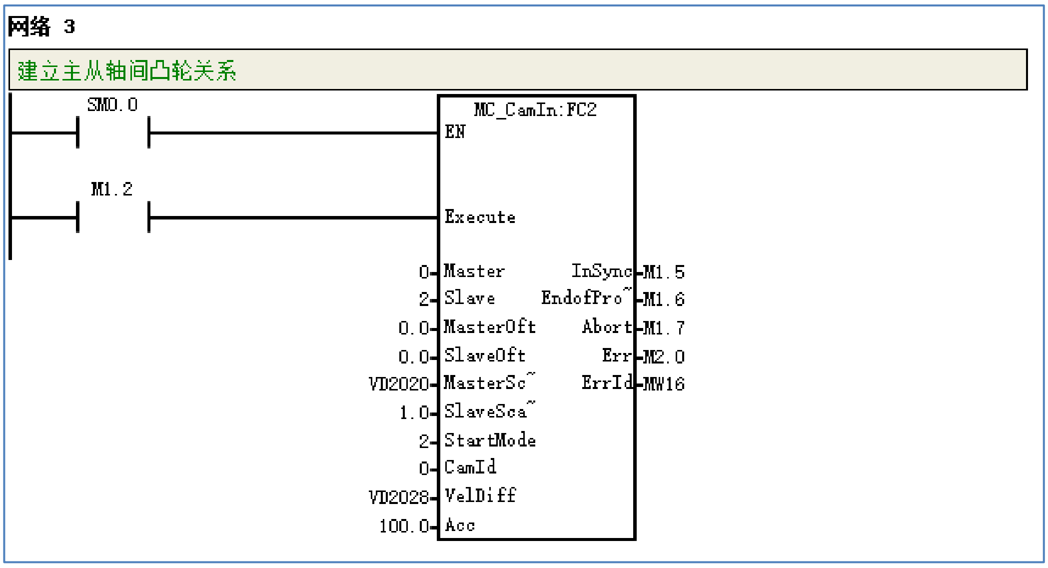
Network 1: Enable axis operation.



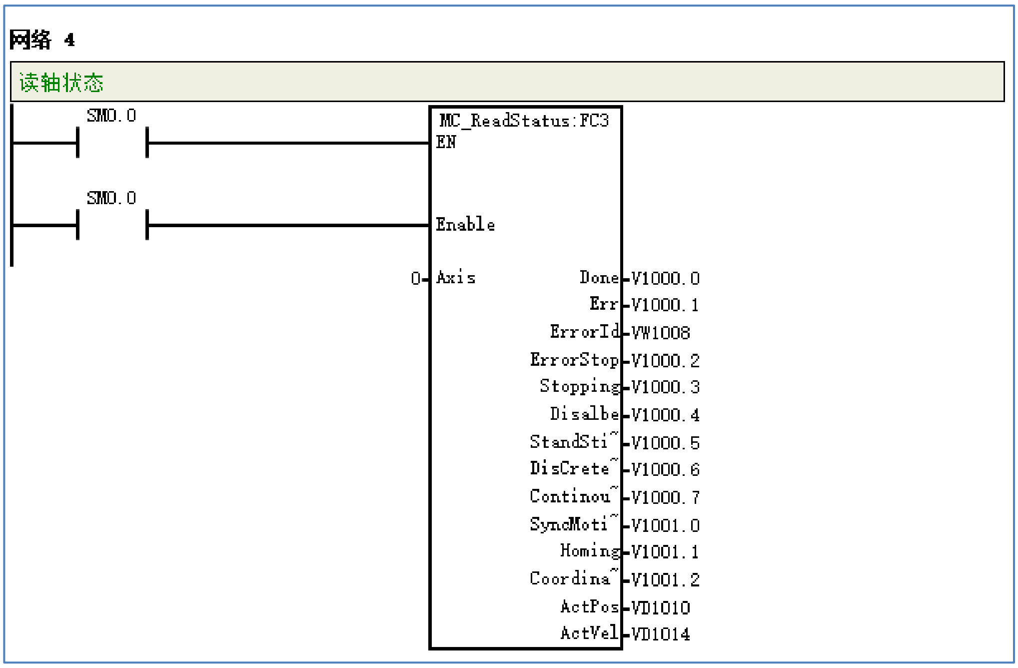
Network 2: Select the cam curve and set parameters such as master station, slave station information, and status information.



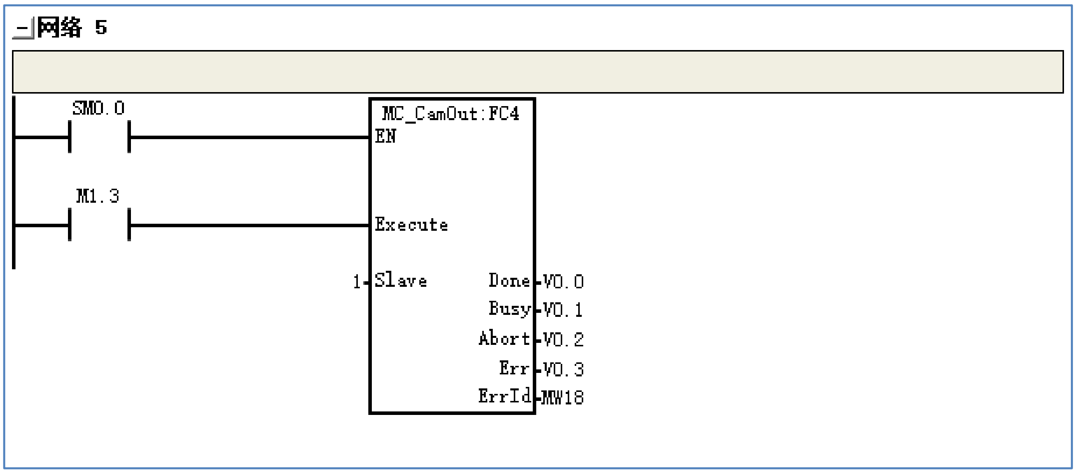
Network 3: Establish a cam relationship between the master and slave axes. When establishing the cam relationship, specify the offset value, scaling ratio, and startup mode of the master and slave axes according to application requirements. When the instruction is executed, the slave shaft follows the main shaft movement according to the cam curve.



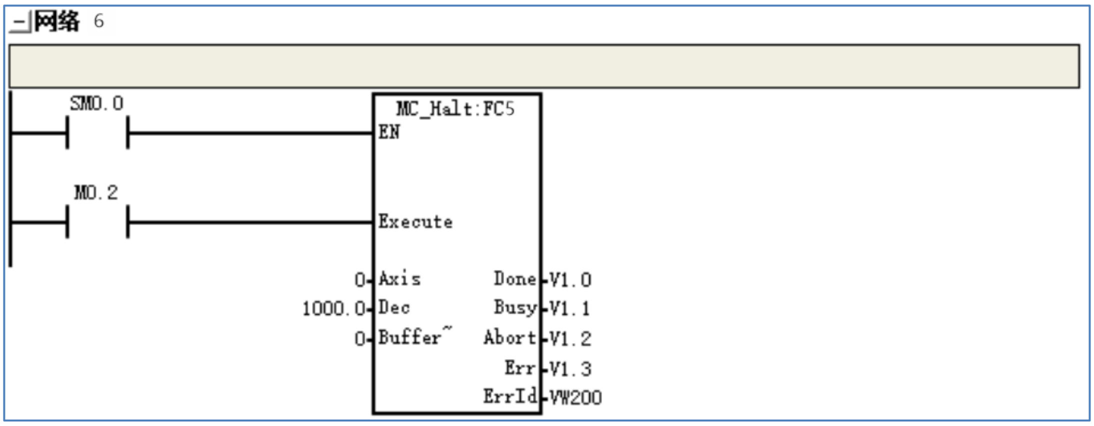
Network 4: Read the corresponding axis status.



Network 5: Release the cam relationship between the master and slave axes. After the relationship is released, the slave axis will continue to move at the speed before detachment.



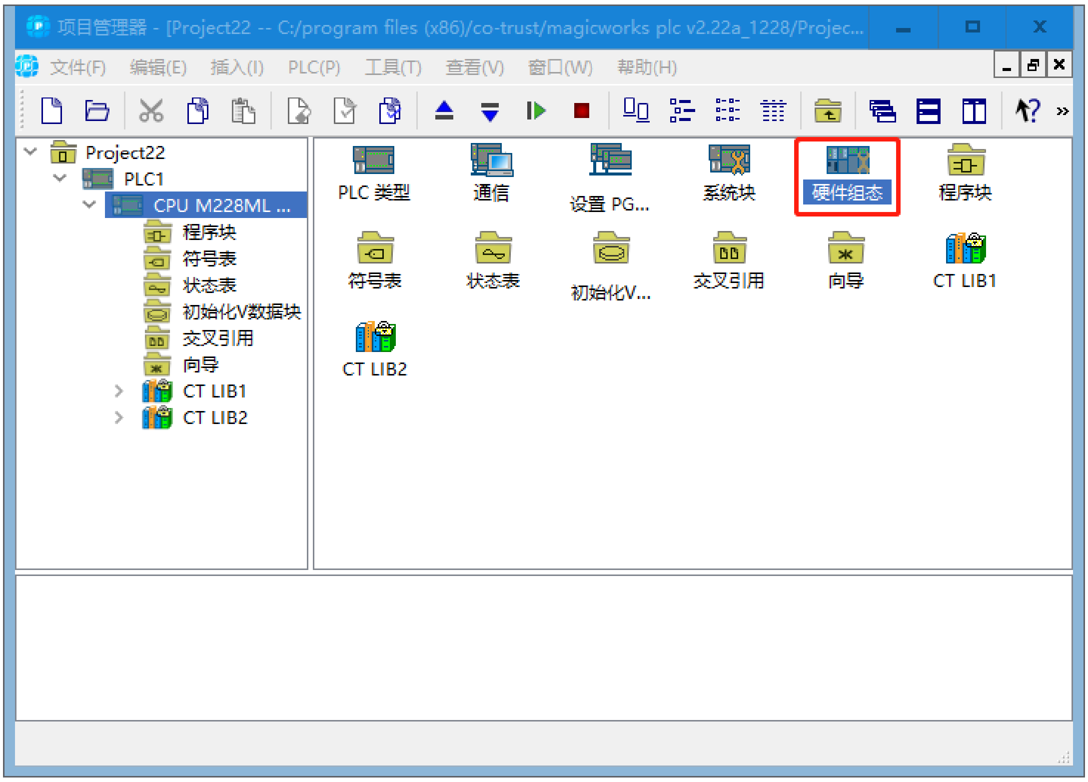
Network 6: The axis decelerates at a given rate until it stops.



6.6 Example of application of continuous interpolation function

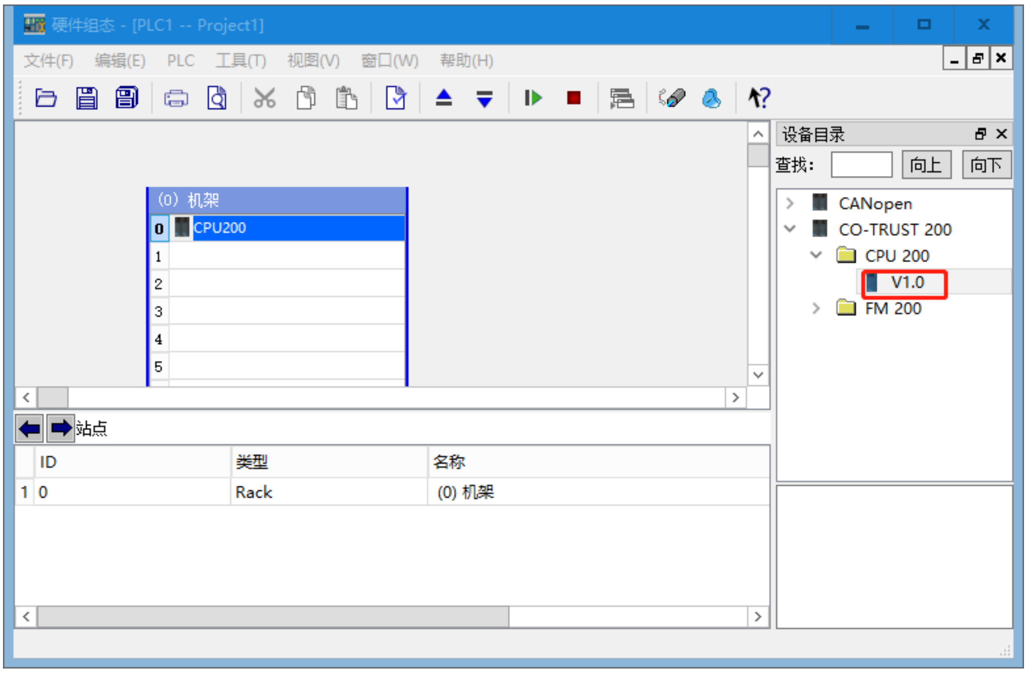
Taking CPU M226ES as an example, the continuous interpolation function will be introduced in detail.

Select "Hardware Configuration" in the project manager interface of M226ES station to configure the hardware, and then call the interpolation command in the main program to program. The interpolation instruction called in this example is the relative displacement linear interpolation instruction.



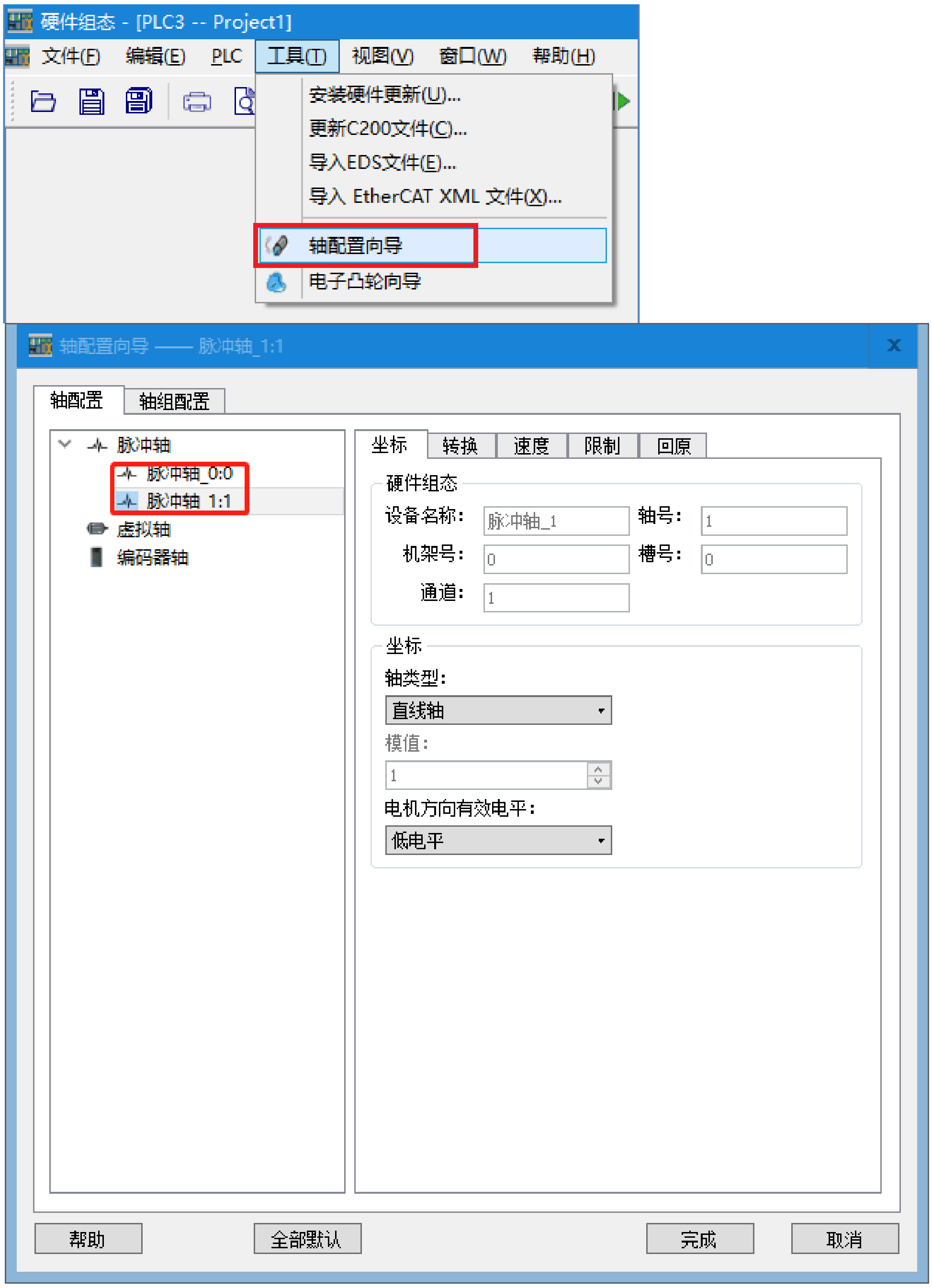
Step 1: Add the CPU module to the CPU.

In the hardware configuration interface, expand the "CPU 200" folder under the "CO-TTRUST 200" project tree, add the CPU module, and the CPU can only be placed in slot 0 of the rack, as shown in the following figure.



Step 2: Axis Configuration

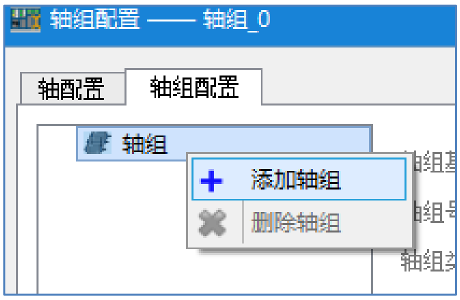
Select "Tools" - "Axis Configuration Wizard" on the hardware configuration interface, and add two pulse axes as shown in the following figure.



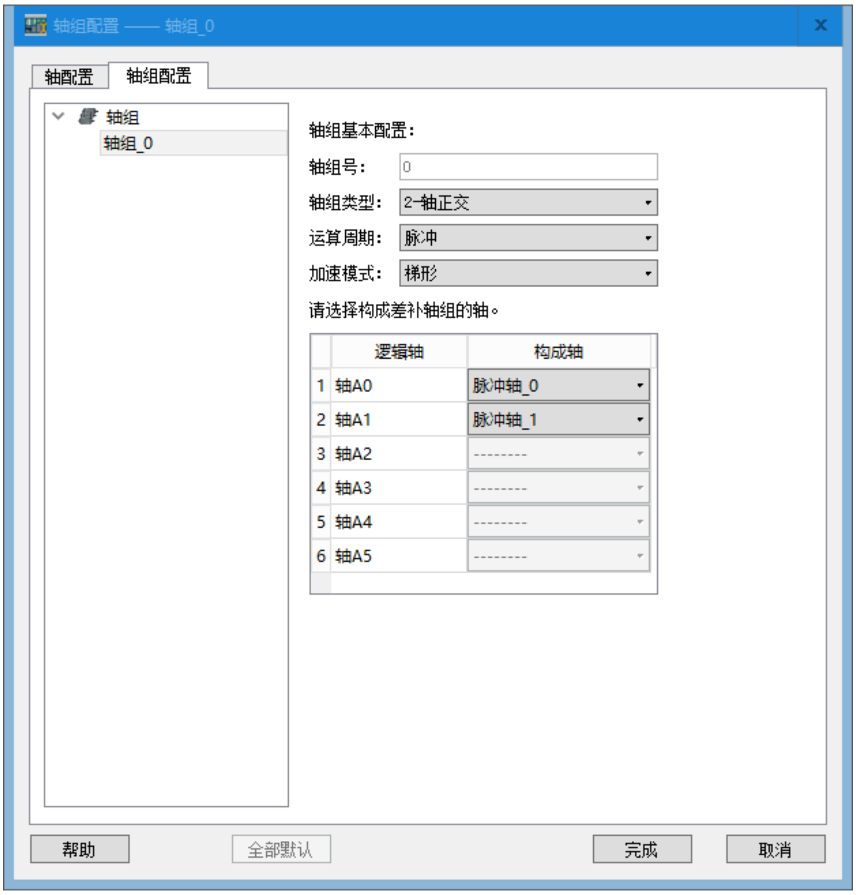
<Note>To use the interpolation function, two or more pulse axes need to be configured.

Step 3: Configure the axis group

Axis group configuration: Two or three axes that have been configured can be freely combined in the form of axis groups. In this example, there are two axes configured. In the "Axis Group Configuration" interface, right-click on "Add Axis Group" to add an axis group, as shown in the following figure:



The interface for successfully adding axis groups is shown below:



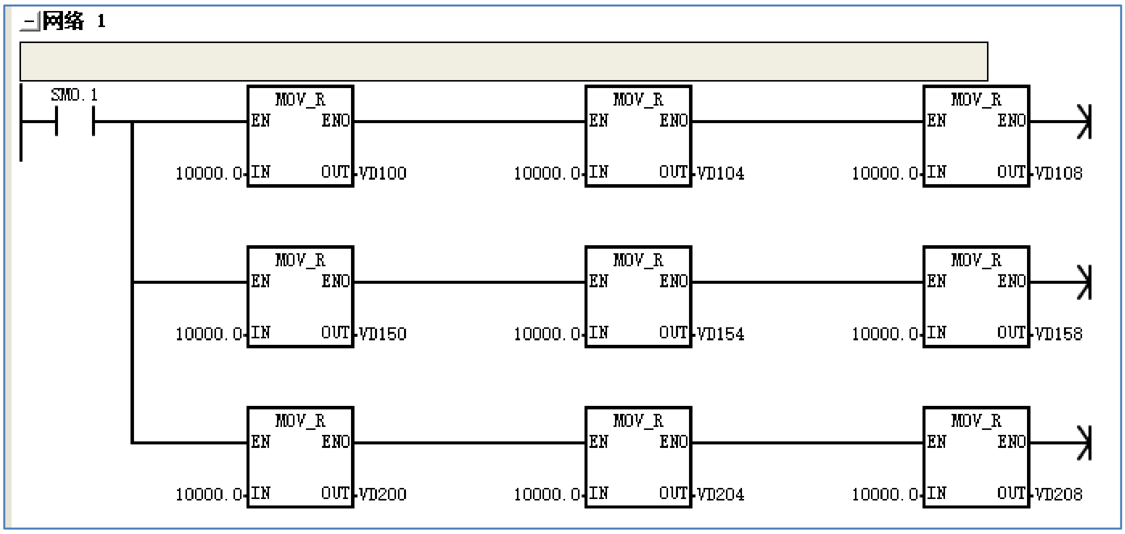
● Basic configuration of axis group ":" Axis group number "cannot be changed. The drop-down options for" Axis group type "can be selected as" 2-axis orthogonal "or" 3-axis orthogonal ", corresponding to two or three groups of" constituting axes "that can be selected respectively. The 'operation cycle' can be selected as' pulse '. The "acceleration mode" can be selected from "trapezoidal", "sin * 2", and "quadratic".

● Logical axis: default six groups, cannot be changed.

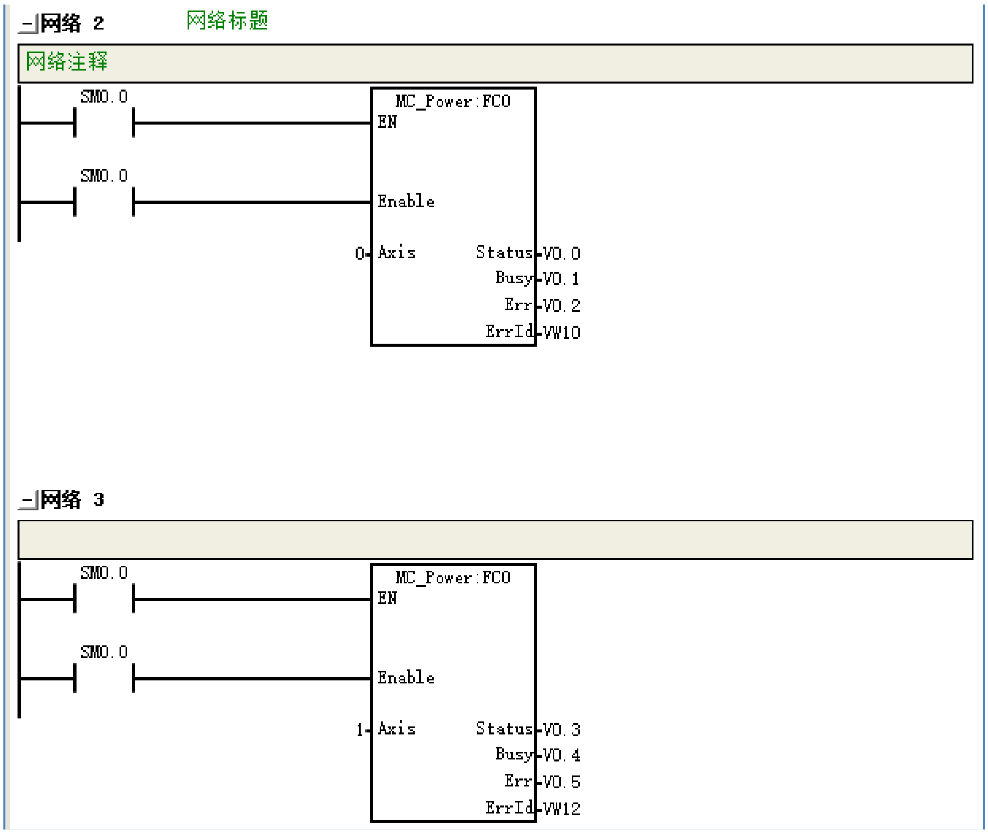
● Composition axis ":" 2-axis orthogonal "corresponds to two composition axes," 3-axis orthogonal "corresponds to three composition axes, and the drop-down options can select" pulse axis "or" virtual axis ".

Step 4: Call the interpolation instruction in the main program to program.

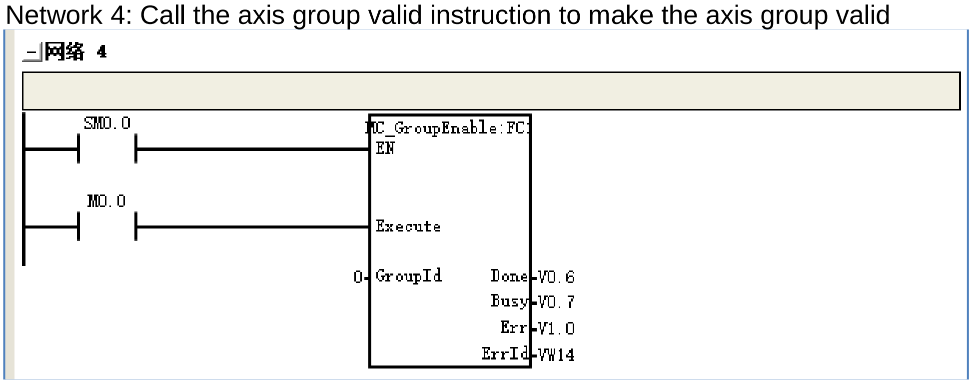
Network 1: Set the speed of interpolation instructions



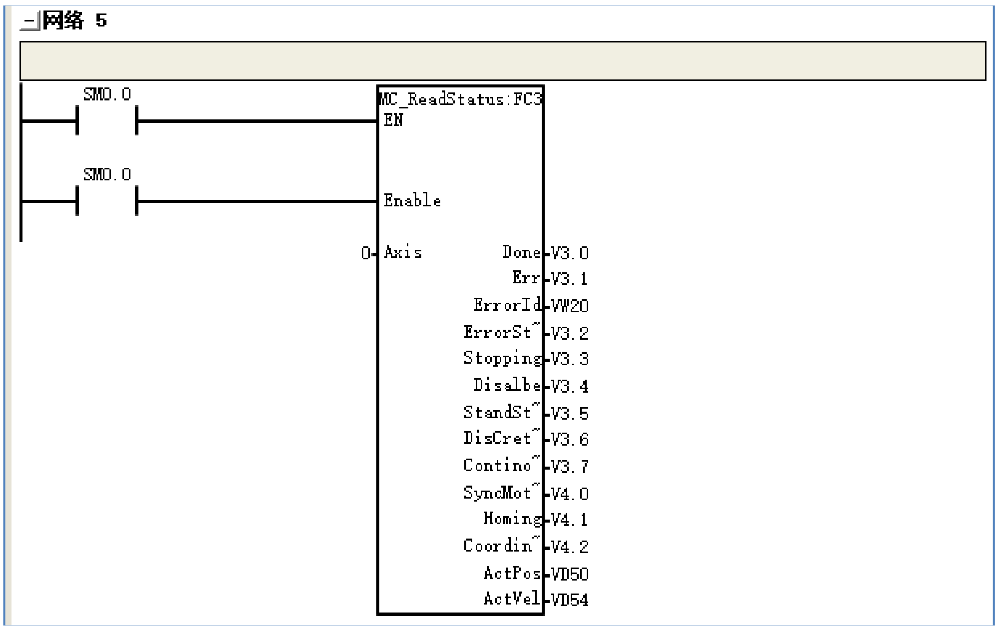
Network 2 and Network 3: Enable the two pulse axes separately.

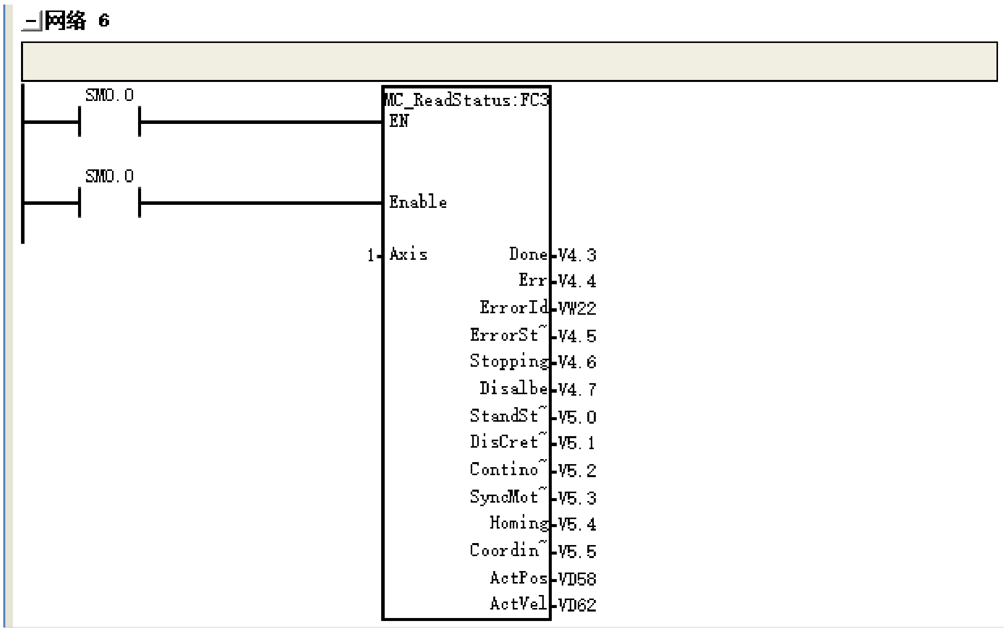


Network 4: Call the axis group valid instruction to make the axis group valid



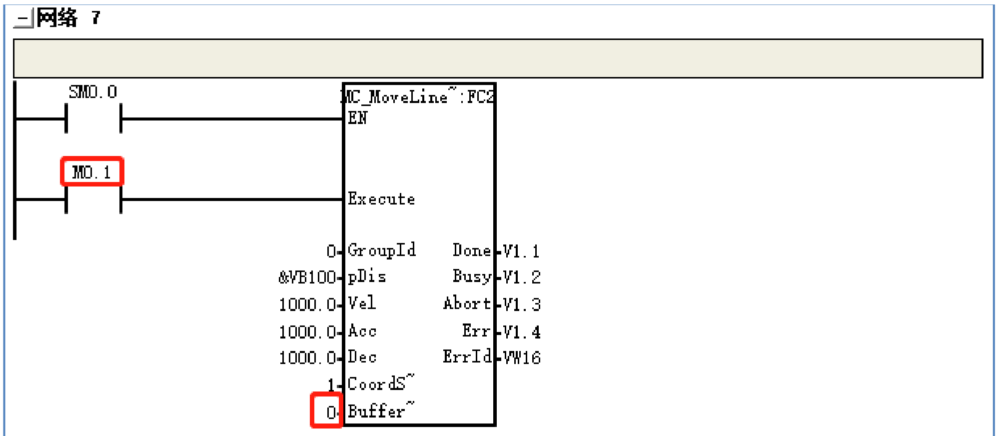
Network 5 and Network 6: Read the corresponding axis status.

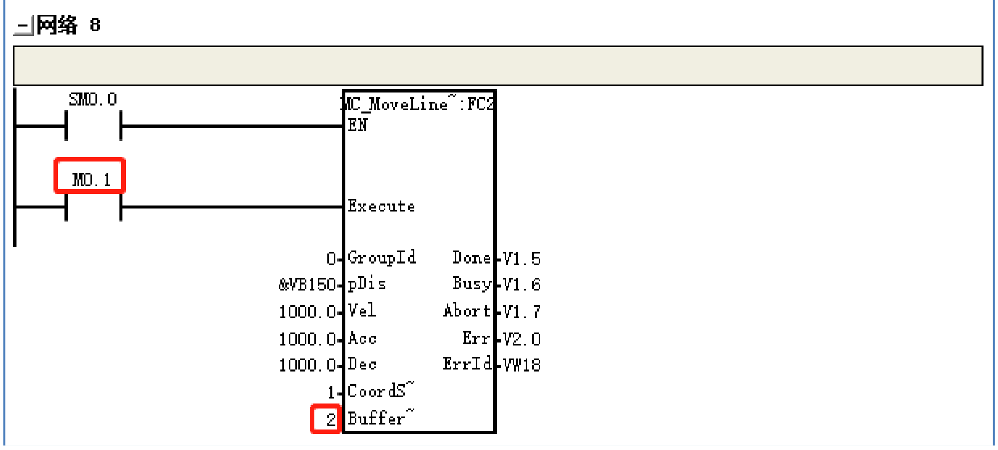


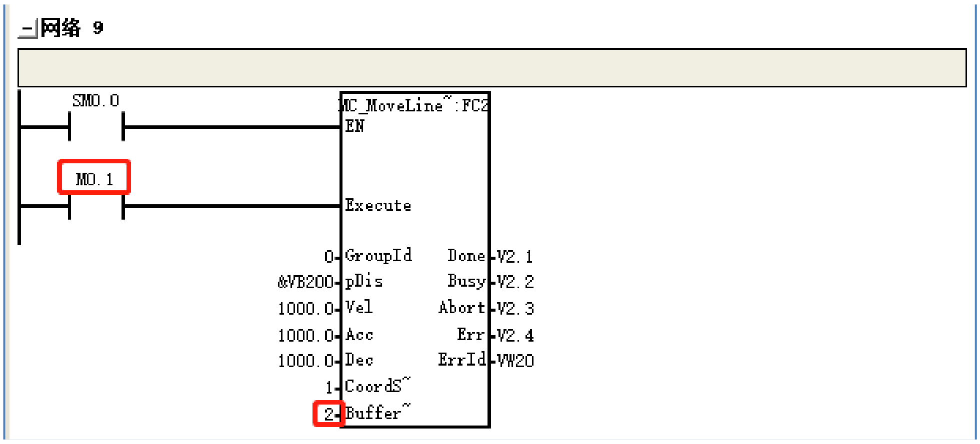


Network 7, 8, 9: Set parameters such as running speed, acceleration, deceleration, interpolation mode, etc. The instruction enable bit can be triggered simultaneously with the same bit (2 consecutive interpolation instructions are cached in the program), and then perform continuous linear interpolation motion in the order of the instructions

< **Note:** The maximum number of cached instructions for consecutive interpolation is 8. If you want to cache the 9th instruction, enable caching the 9th instruction after the first instruction is completed.







6.7 Homing function

This chapter introduces the pulse return mode of MC-Home. M226ES can use the following 15 return modes, which users can choose according to their accuracy requirements and practical application needs.

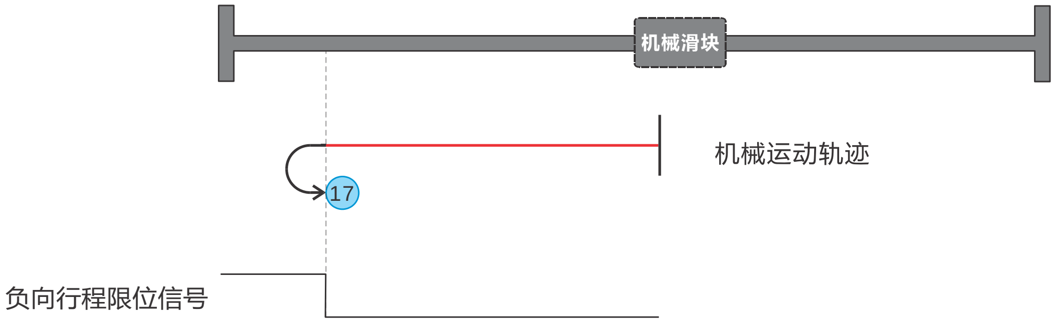
pattern	explain
17	Reference negative travel limit signal
18	Reference forward travel limit signal
19	Reference forward origin signal switch
20	Reference forward origin signal switch
21	Reference negative origin signal switch
22	Reference negative origin signal switch
23	Reference origin switch and positive limit origin mode
24	Reference origin switch and positive limit origin mode
25	Reference origin switch and positive limit origin mode
26	Reference origin switch and positive limit origin mode
27	Reference origin switch and negative limit for origin mode
28	Reference origin switch and negative limit for origin mode
29	Reference origin switch and negative limit for origin mode
30	Reference origin switch and negative limit for origin mode
35	Set the current position as the origin

6.7.1 Homing mode description

Regardless of the initial position of the machine, when the equipment (origin switch, positive travel limit switch, negative travel limit switch) is installed properly, the servo always finds a unique origin for the equipment. The purpose of returning to the origin is to unify the actual position of the axis with the position mark set in the program. The vertical line |"" in the following schematic diagrams represents the mechanical initial position, and the circle "" represents the origin position. **- "represents** search speed,**"** -

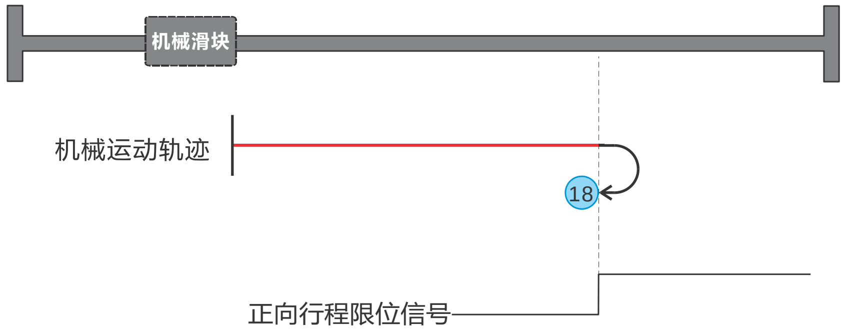
"represents crawling speed.

Homing mode 17: Reference negative limit falling edge



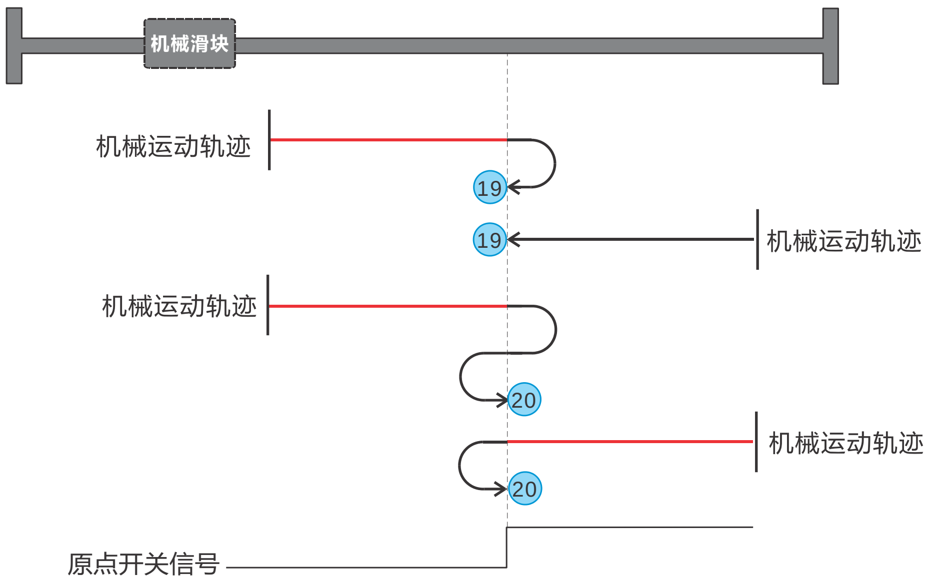
At the beginning of returning to the original position, the negative travel limit signal is 0, and the reverse high-speed return is carried out. When encountering the rising edge of the travel limit signal, the deceleration and reverse are carried out, and the low-speed operation is stopped until encountering the falling edge of the travel limit signal. This position is the origin position.

Homing mode 18: Reference positive limit falling edge



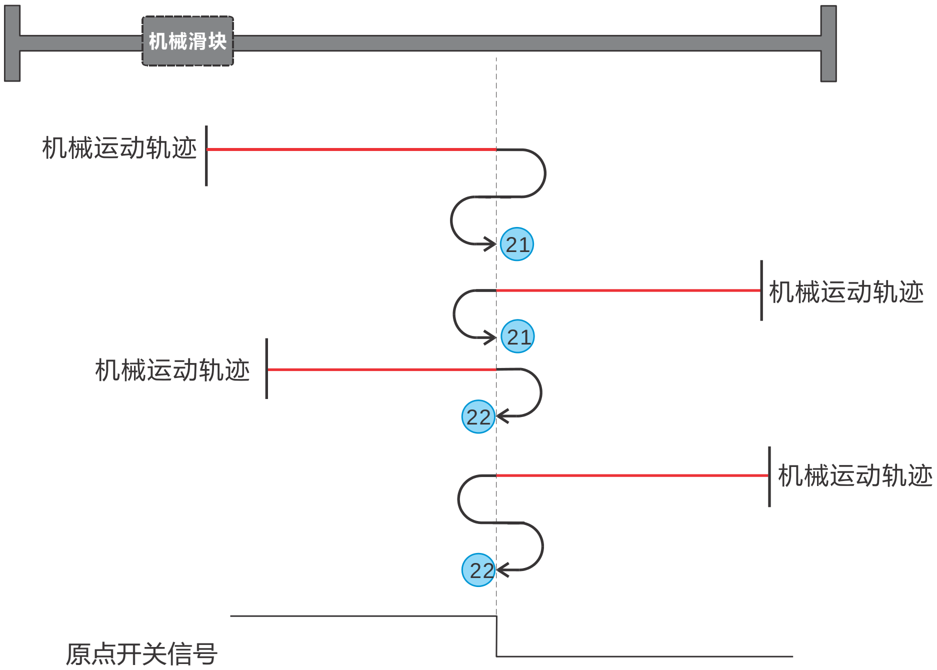
At the beginning of returning to the original position, the forward travel limit signal is 0, and the forward high-speed return is carried out. When encountering the rising edge of the travel limit signal, the deceleration and reverse are carried out, and the low-speed operation is stopped until encountering the falling edge of the travel limit signal. This position is the origin position.

Homing mode 19, 20: Reference forward origin signal switch



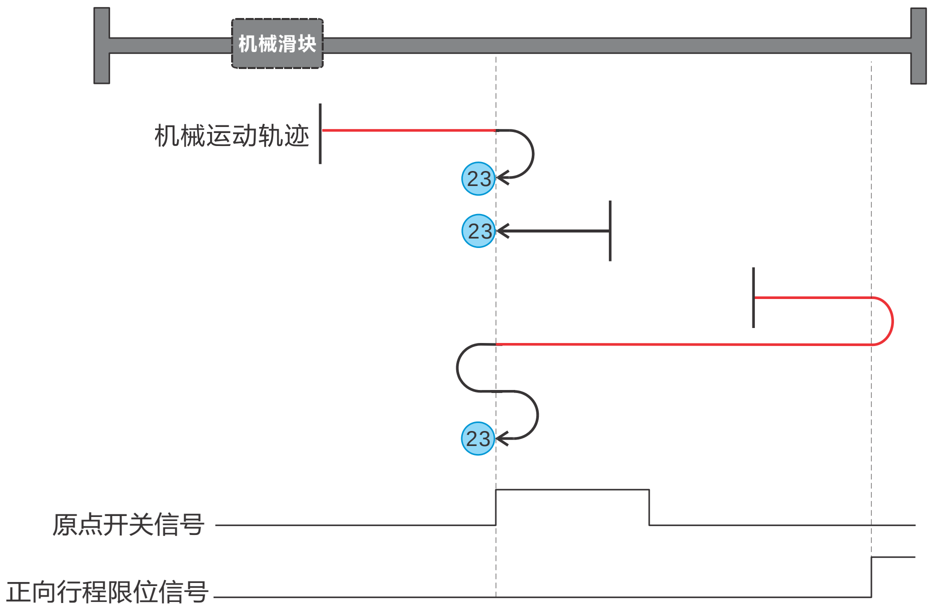
The starting direction of motion depends on the high and low states of the origin switch signal. The origin of mode 19 is at the falling edge position where the origin switch signal changes from high to low. The origin of Mode 20 is at the rising edge position where the origin switch signal changes from low to high.

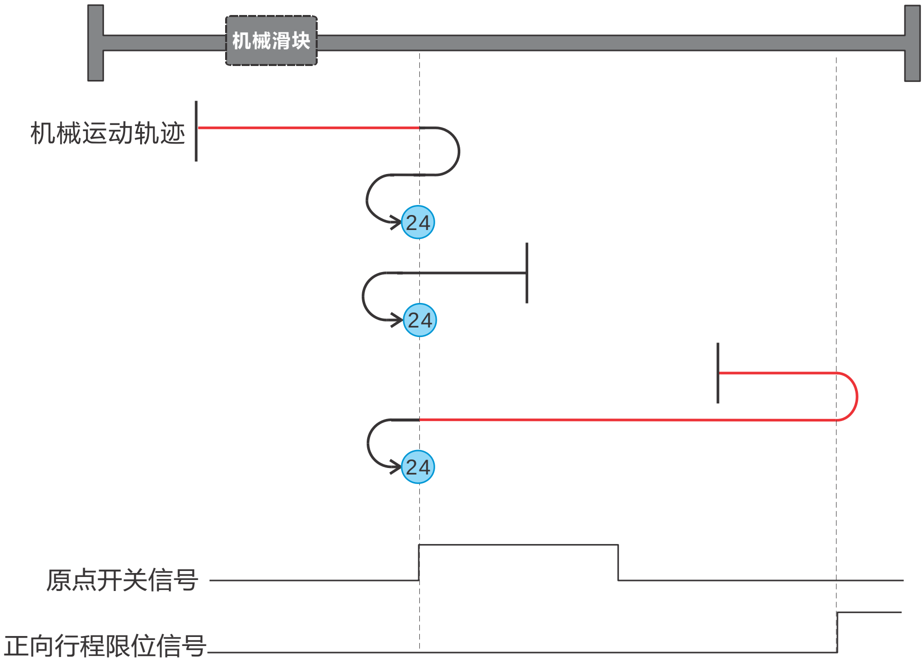
Homing mode 21, 22: Reference negative origin signal switch

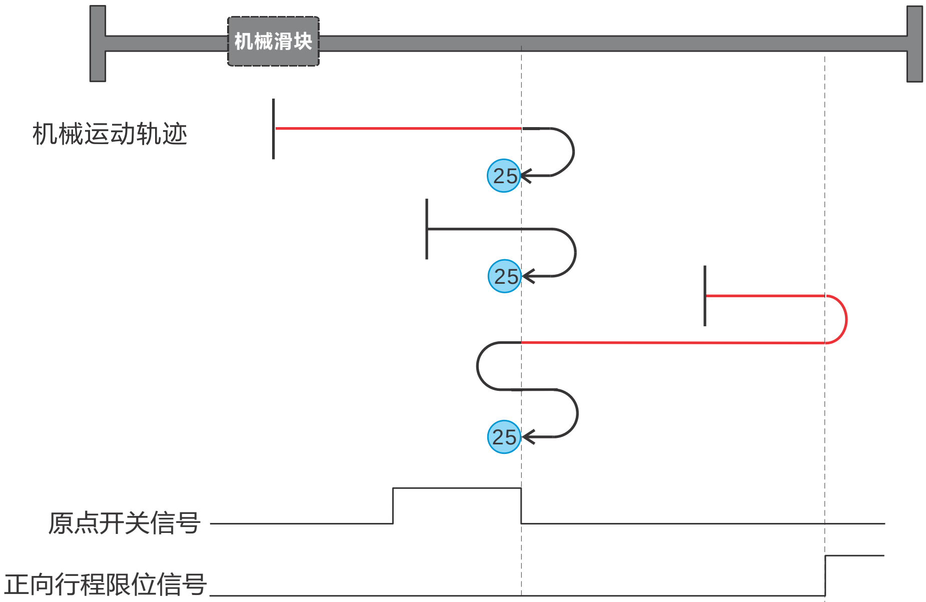


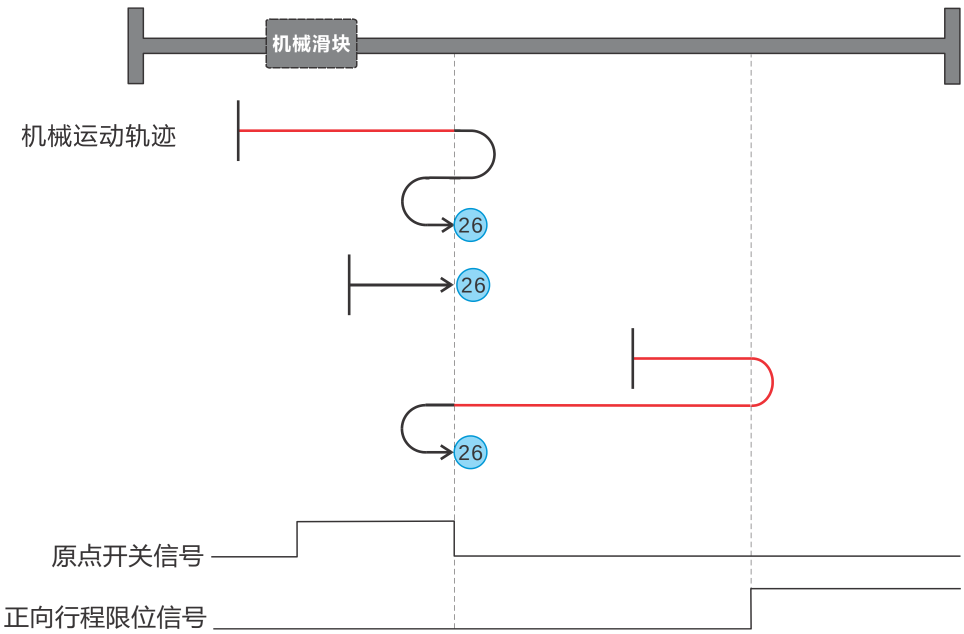
The starting direction of motion depends on the high and low states of the origin switch signal. The origin of mode 21 is at the falling edge position where the origin switch signal changes from high to low. The origin of mode 22 is at the rising edge position where the origin switch signal changes from low to high.

Homing mode 23~26: Reference origin signal switch and forward travel limit



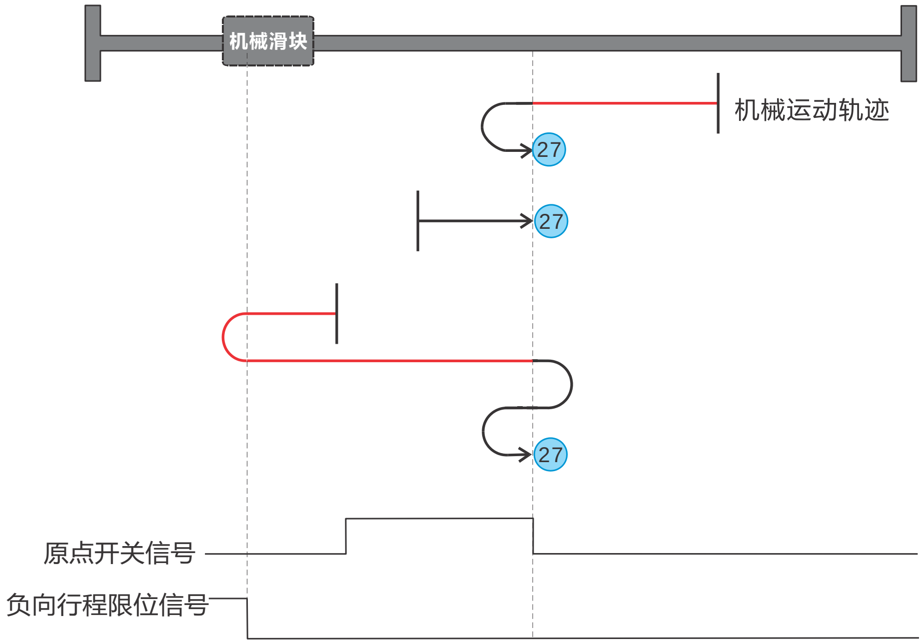

机械滑块
机械运动轨迹
24
24
24
原点开关信号
正向行程限位信号

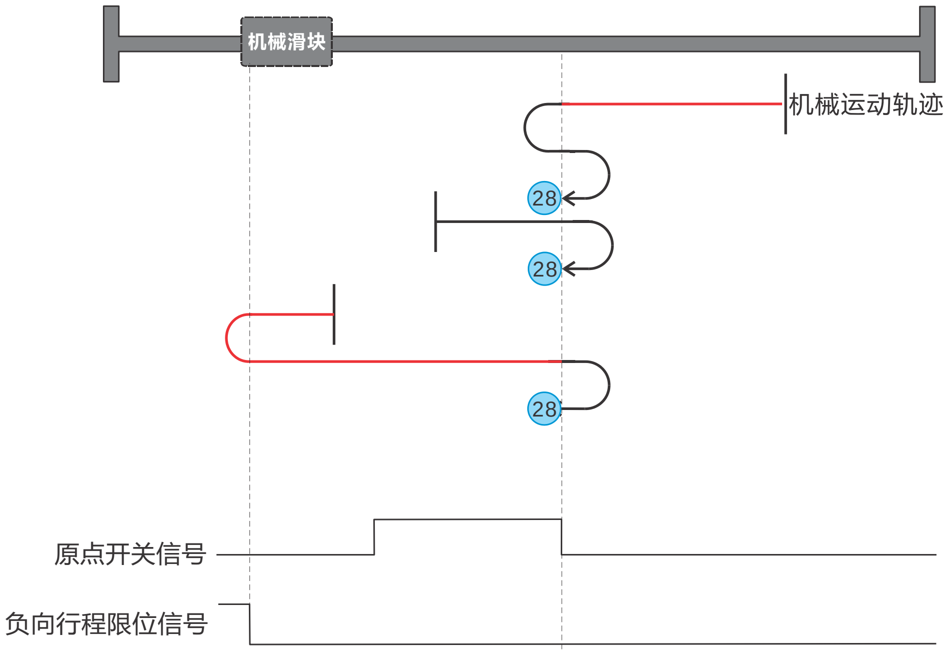

机械滑块
机械运动轨迹
25
25
25
原点开关信号
正向行程限位信号

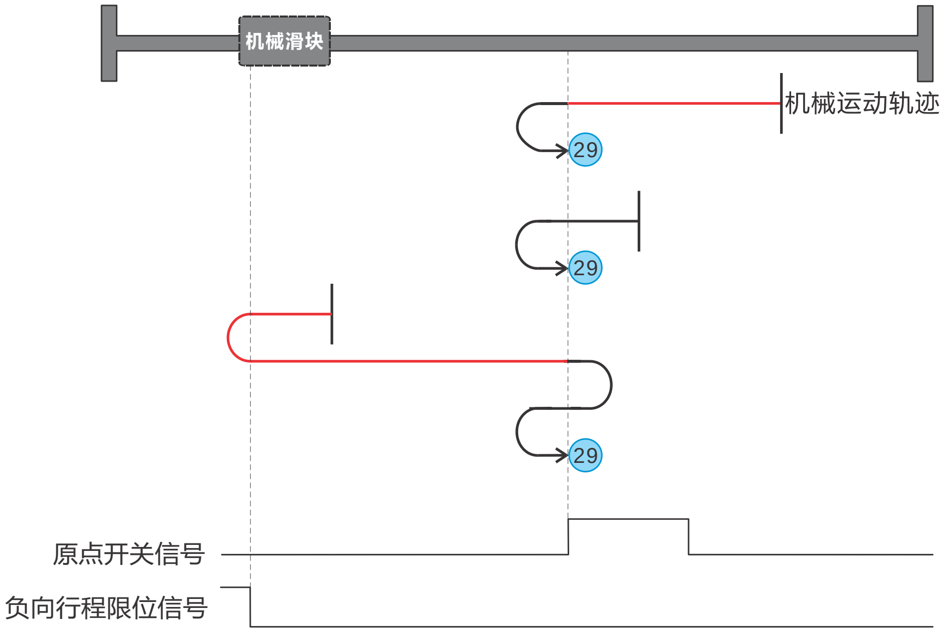

机械滑块
机械运动轨迹
26
26
26
原点开关信号
正向行程限位信号

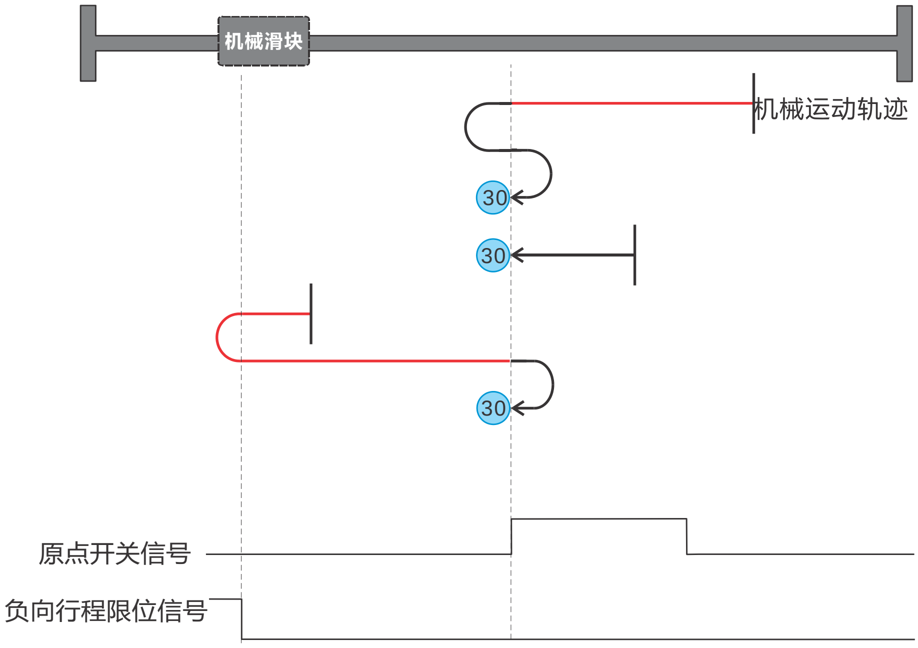
The starting direction of motion depends on the status of the origin switch signal and the forward travel limit signal. The origin of modes 23 and 26 is at the falling edge position where the origin switch signal changes from high to low. The origin of modes 24 and 25 is at the rising edge position where the origin switch signal changes from low to high.

Homing mode 27~30: Reference origin signal switch and negative travel limit







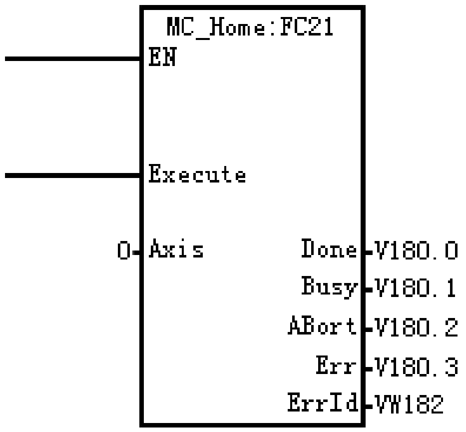
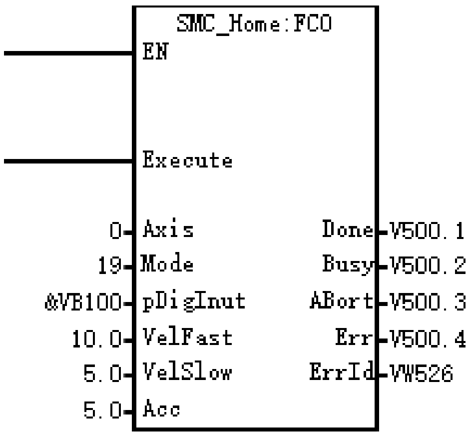


The starting direction of motion depends on the status of the origin switch signal and the negative travel limit signal. The origin of modes 27 and 30 is at the falling edge position where the origin switch signal changes from high to low. The origin of modes 28 and 29 is at the rising edge position where the origin switch signal changes from low to high.

Homing mode 35: Using the current position as the origin

The motor stops at the current position and completes the return to the original signal given.

6.7.2 The difference between MC-Home (Origin Regression) and SMC-Home (Special Origin Regression)

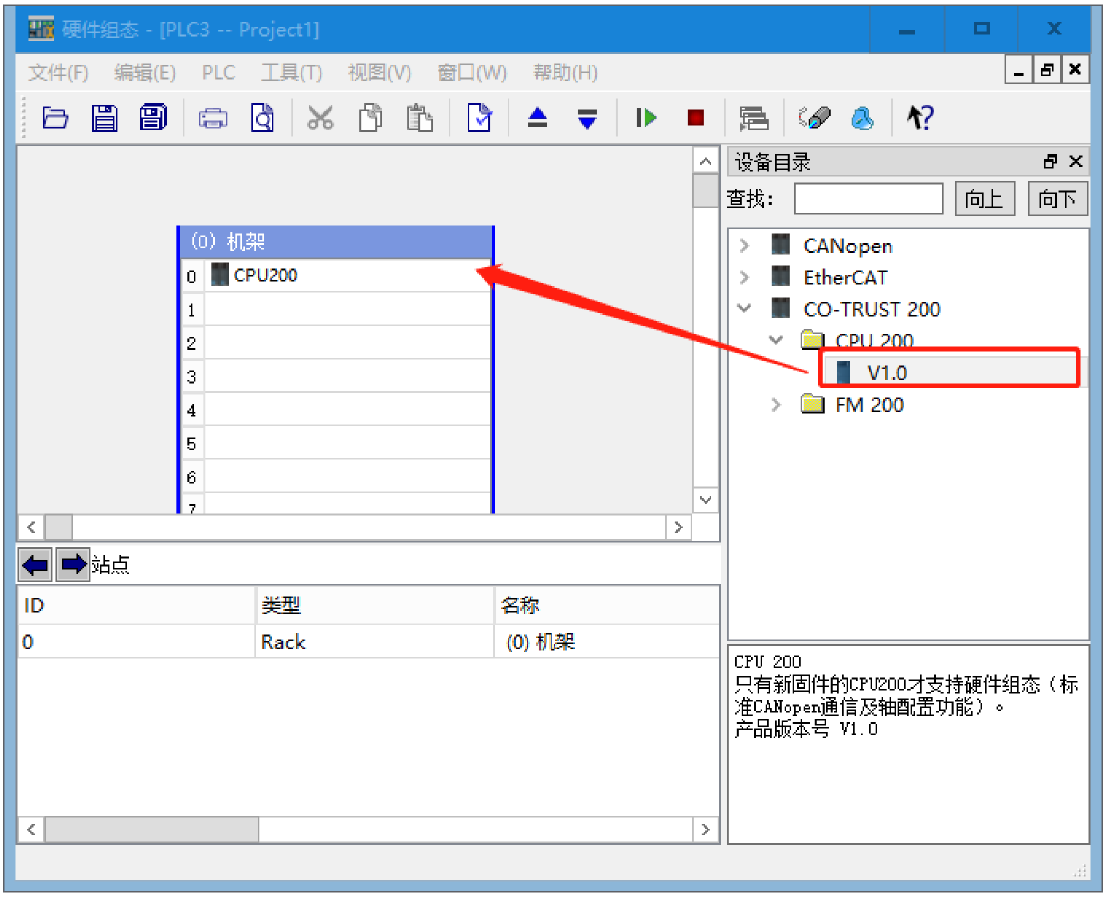
Origin regression	Special Far Point Regression
	
1. Connect the origin and positive/negative limit to the servo 2. Trigger the DI terminal of the servo to return to its original state 3. The servo itself has higher control accuracy MC-Home needs to first configure the servo's return mode, return speed, etc. The program triggers MC-Home, and the PLC simply changes the servo operation module to return to the original mode, starts the return to the original mode, waits for the servo to return to the original mode, feedbacks the completion signal, and then switches the mode to the control mode before returning to the original. Call MC-Power to enable servo. After the servo is enabled, call the MC_Home command, and the output Done is ON, indicating that the return to the original is complete	1. Connect the origin to the PLC 2. PLC controlled servo motion 3. PLC is affected by scanning cycles and has low accuracy 4. Connect the positive and negative limit switches, as well as the origin switch, to the PLC. After SMC_Home is activated by the rising edge of Execute, the axis will start moving at a speed of VelFast and in the direction defined by pDigInput Note: SMC_Home is controlled by the controller to return to the original position, and the current position after the return is completed is the position set by the controller, not equal to the servo encoder value. The position may change after re powering on or initializing the axis PDigInput: The memory address pointed to by the bit numbers of the origin switch, forward and reverse limit switches. Bit0: Negative Switch Bit1: Forward switch Bit2: Origin switch Switch signal 1 is valid, 0 is invalid. For example, when pDigInput is equal to&IB0, I0.0: Negative Switch I0.1: Positive switch I0.2: Origin switch.

6.7.3 Example 1: Pulse Axis 19 Homing Mode Example

This example takes the 19th as an example, and provides a simple explanation of axis control by combining axis enable, axis status acquisition, axis jog, alarm release, and return to the original.

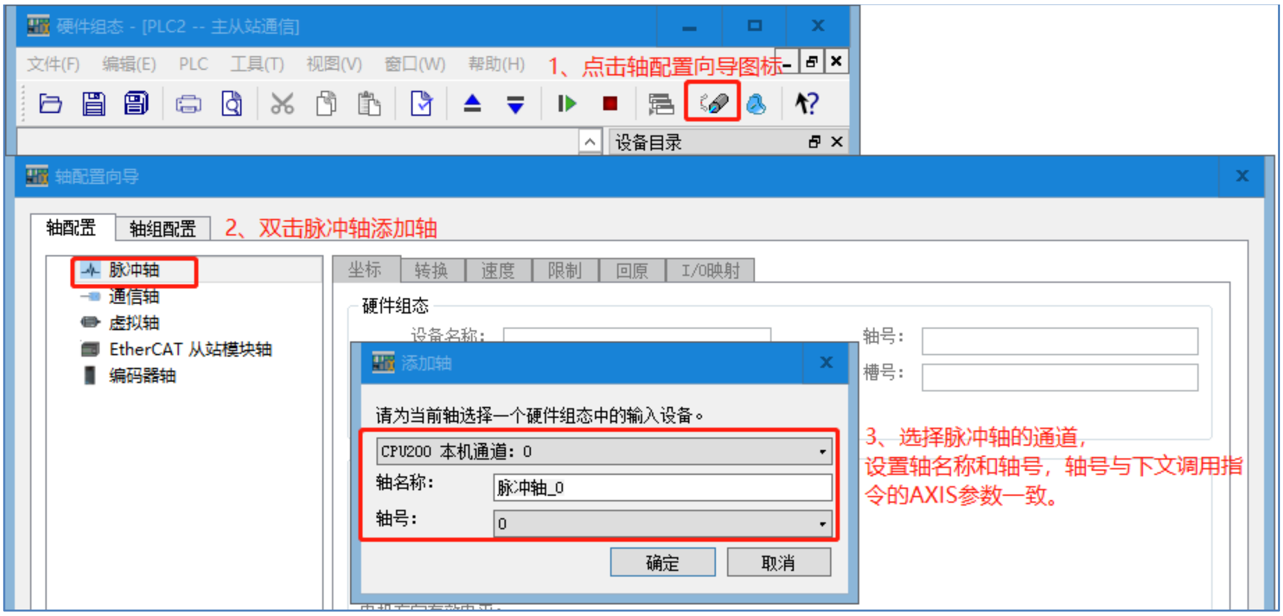
Step 1: Add CPU

Select "Hardware Configuration" in the project manager interface of the CPU site. In the hardware configuration interface, under the "CO-TTRUST 200" project tree, select "CPU 200" in sequence. The CPU can only be placed in slot 0 of the rack, as shown in the following figure:

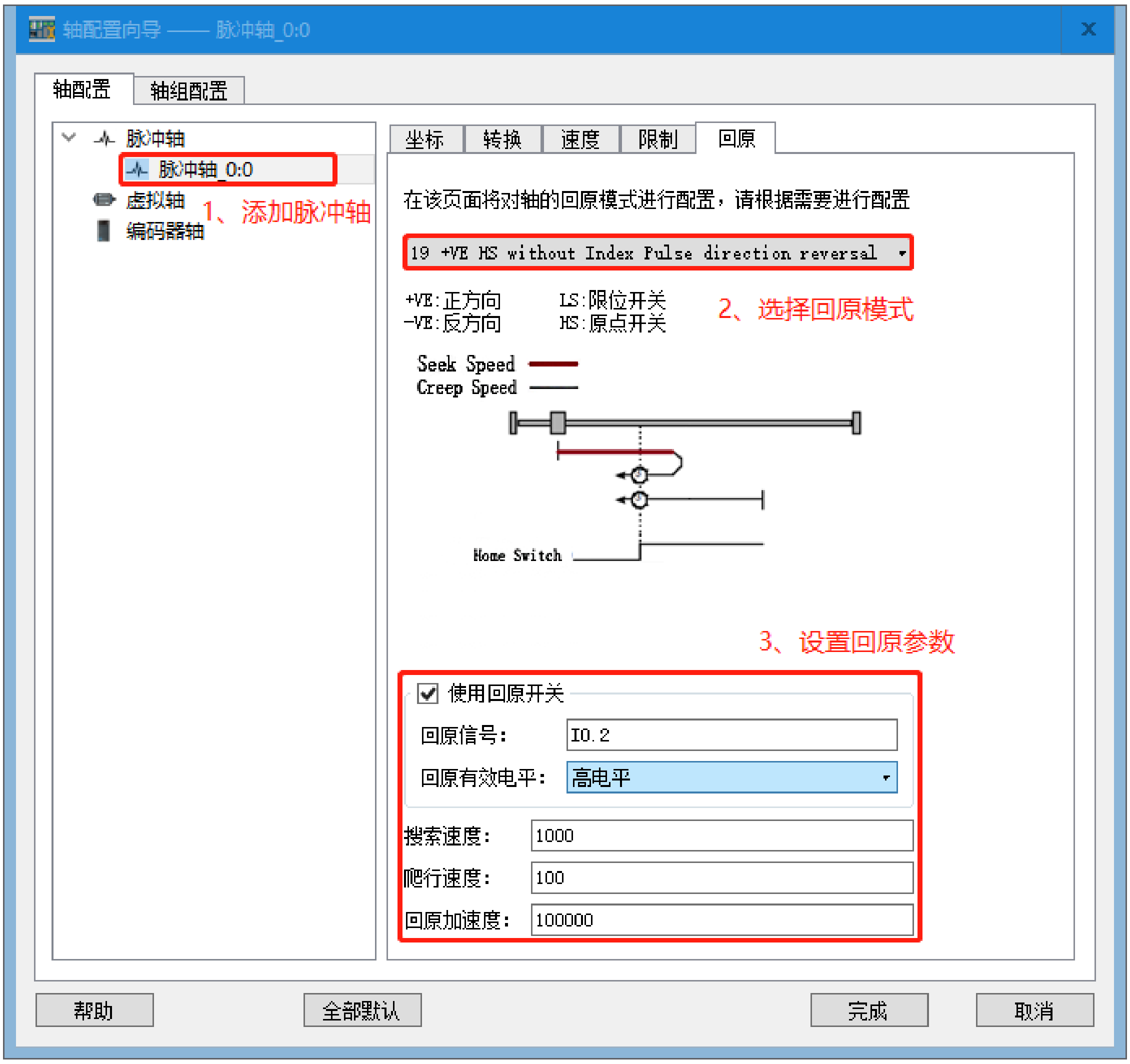


Step 2: Configure the axis

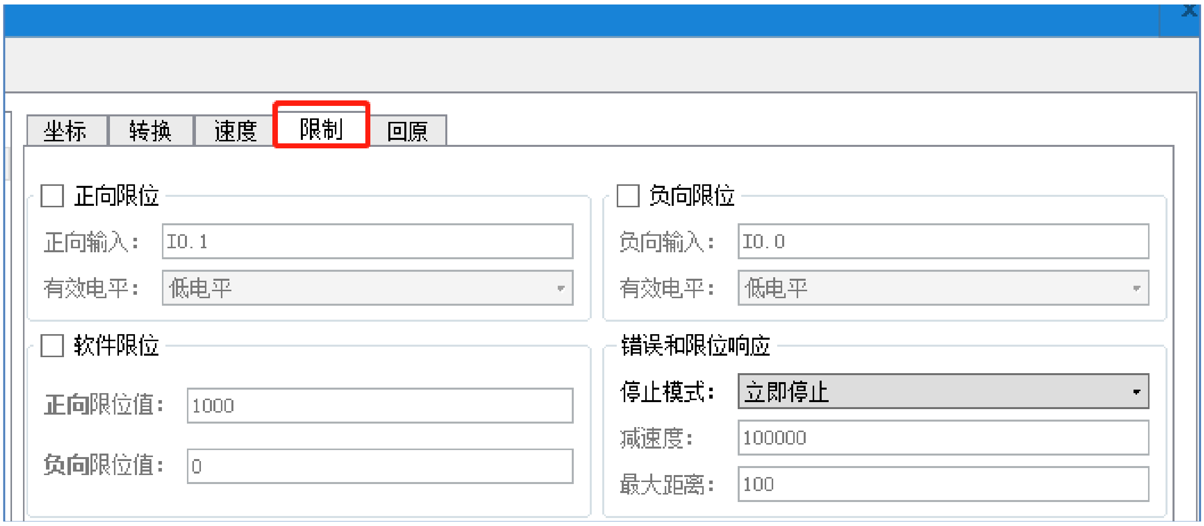
1. Enter the Axis Configuration Wizard and add a pulse axis to configure the axis, as shown in the following figure.



2. Click on 'Return to Origin' to configure the axis's return to origin mode on this page. Only the pulse axis supports return to origin configuration. Select the dropdown option to view the return to origin mode. In this example, the 19th return switch (referring to the forward origin signal switch) is selected. The 19th return switch only refers to the forward origin signal switch, so only the return signal and other parameters need to be set on the return interface.



3. If a limit switch is used to return to the original mode, set the limit signal in "Limit".



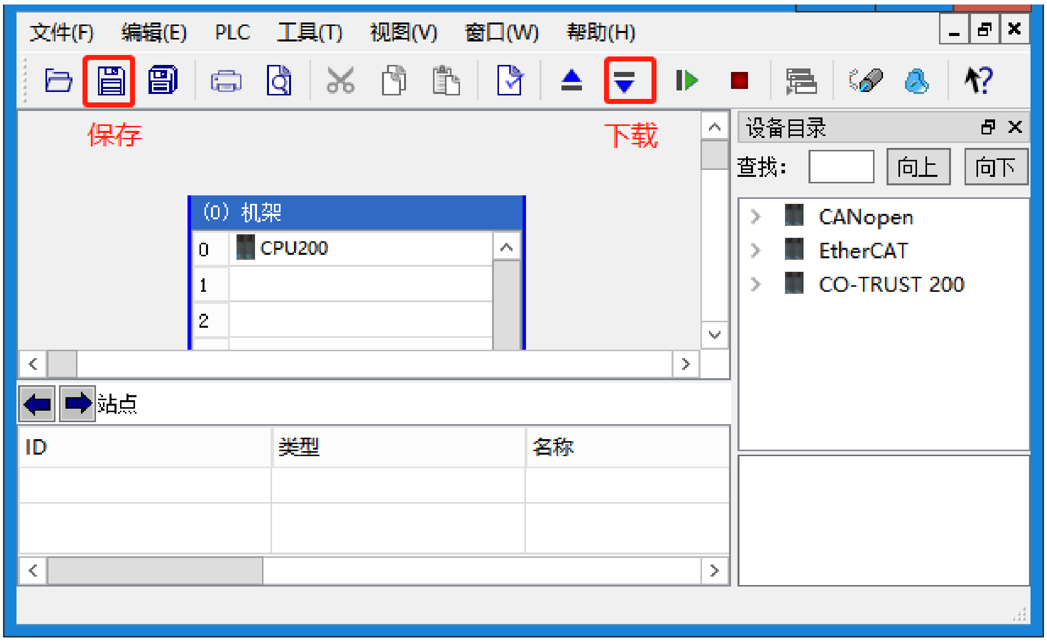
4. Dimensional conversion

PLC controls the action of servo motor by controlling the number of pulses sent, and servo driver controls the distance and speed of servo motor operation by receiving the number and pulse frequency. For example, as shown in the figure below, the PLC sends 131072 pulses (increments) to the servo motor, and the motor rotates once. If the PLC sends 131072 pulses within one minute, the servo motor completes one revolution at a speed of 1r/min. If 131072 pulses are sent within one second, the servo motor rotates one revolution at a speed of 60r/min.

The reduction ratio of 5:1 in the figure represents that for every 5 revolutions of the servo motor shaft, the output shaft of the reducer rotates 1 revolution. If no gearbox is used, the reduction ratio is 1:1.

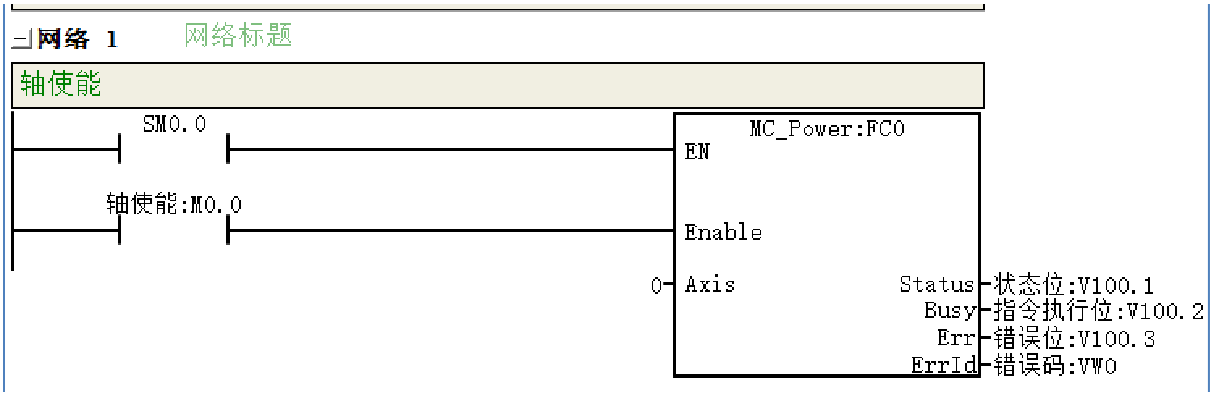
The application unit is set to 200, and the gear output is 1, indicating that the gear rotates one unit, while the application unit executes 200 units.

5. Save the configuration and download it to the PLC.



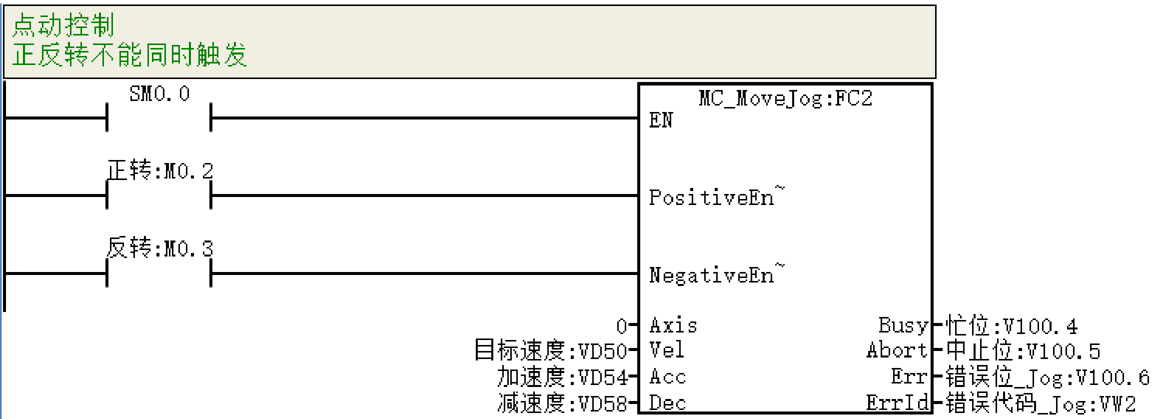
Step 3: Call the ct_plcopen_lib library instruction in the main program to program.

Network 1: Call MC-Power to enable the axis. If Enable is set to TRUE, it will enter the runnable state, and if it is set to False, it will release the runnable state. Axis ID number Axis is set to 1.



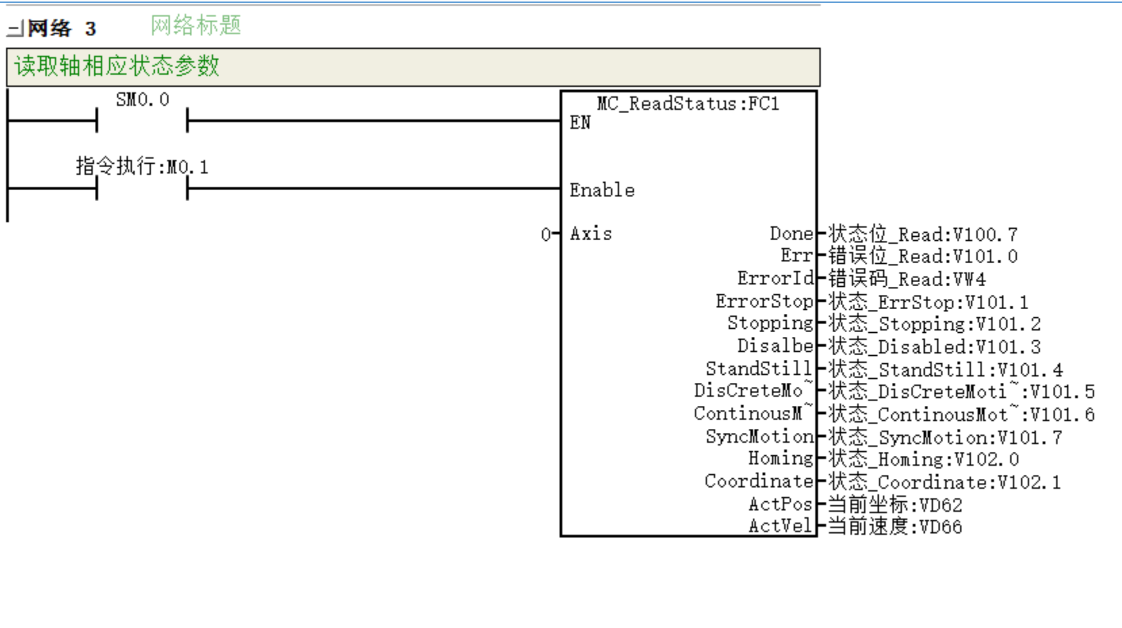
Network 2: Call the jog command to control the forward and reverse rotation of the axis

Trigger MC_SoveJog, Q0.0 will emit pulses at the target speed, and Q0.1 will be set to TRUE during forward rotation. If Vel=40, the speed is 1 turn per second. if Vel=200, the speed is 5 turns per second.



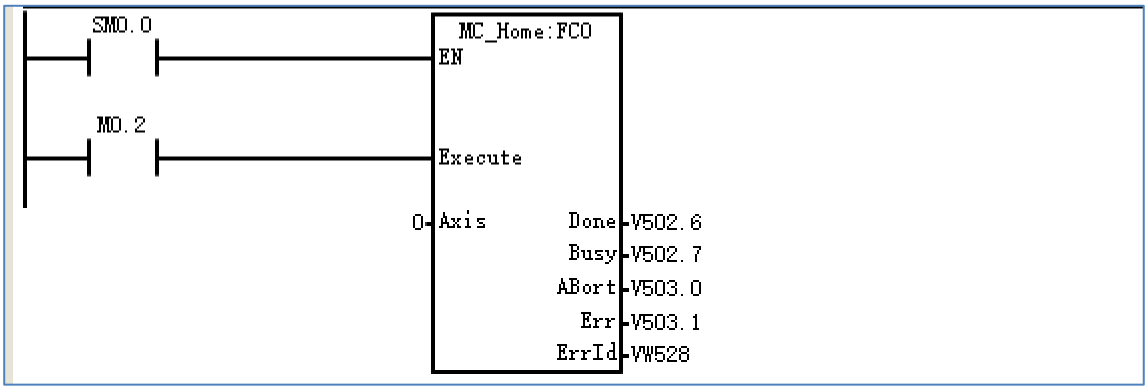
符号	地址	注释
MC_MoveJog	FC2	
正转	M0.2	设为TRUE，开始正方向移动
反转	M0.3	设为 TRUE，开始负方向移动
目标速度	VD50	单位：指令单位/秒
加速度	VD54	单位：单元/秒/秒
忙位	V100.4	指令正在执行为TRUE，执行完成为FLASE。
中止位	V100.5	当指令执行过程中被终止时，Abort位被置位；当…
错误位_Jog	V100.6	如果检测到有错误，Error位被置位；
减速度	VD58	单位：单元/秒/秒
错误代码_Jog	VW2	

Network 3: Call MC_SeadStatus to read axis status for easy monitoring during operation. The front axle should be in a stationary state when returning to its original position. If it is not stationary, use the stop command MC_Stop to stop the axle first.

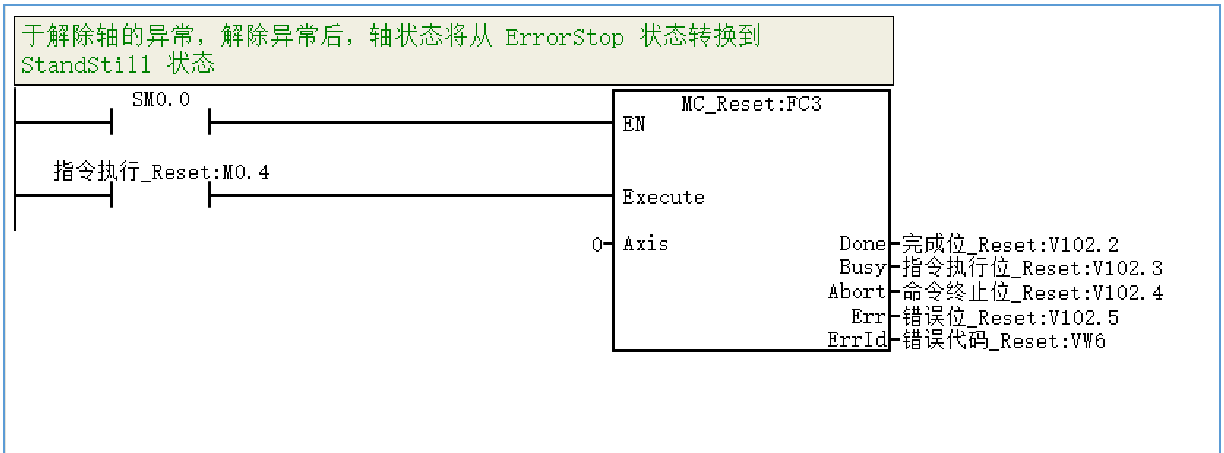


符号	地址	注释
MC_ReadStatus	FC1	
指令执行	M0.1	
状态位_Read	V100.7	
错误位_Read	V101.0	如果检测到有错误，Error位被置位；
错误码_Read	VW4	
状态_ErrStop	V101.1	如果轴位于状态ErrStop，TRUE
状态_Stopping	V101.2	如果轴位于状态Stopping，TRUE
状态_Disabled	V101.3	如果轴位于状态Disabled，TRUE
状态_StandStill	V101.4	如果轴位于状态StandStill，TRUE
状态_DisCreteMotion	V101.5	如果轴位于状态DisCreteMotion，TRUE
状态_ContinousMotion	V101.6	如果轴位于状态ContinousMotion，TRUE
状态_SyncMotion	V101.7	如果轴位于状态SyncMotion，TRUE
状态_Homing	V102.0	如果轴位于状态Homing，TRUE
状态_Coordinate	V102.1	如果轴位于状态Coordinate，TRUE
当前坐标	VD62	单位： 单元
当前速度	VD66	单位：单元/秒

Network 4: Call the MC_Some command to set the Homing mode (Homing on the 19th). The axis runs in the positive direction at a search speed of 1000. When encountering the Homing signal of I0.2 rising edge, it runs in the negative direction at a crawling speed of 100. When encountering the I0.2 falling edge, the axis stops and the position is reset, and the Homing is successful.



Network 4: When the error code displays an error, call MC-Reset to clear the alarm. For specific details on the error code, please refer to section 6.4.4 error code..



符号	地址	注释
MC_Reset	FC3	
指令执行_Reset	M0.4	
完成位_Reset	V102.2	当轴状态被复位至准备执行状态后，Done 位被置…
指令执行位_Reset	V102.3	当指令执行时Busy位被置位。当指令完成Busy复…
命令终止位_Reset	V102.4	当该指令执行过程中被终止时，Abort位被置位；…
错误位_Reset	V102.5	
错误代码_Reset	VW6	

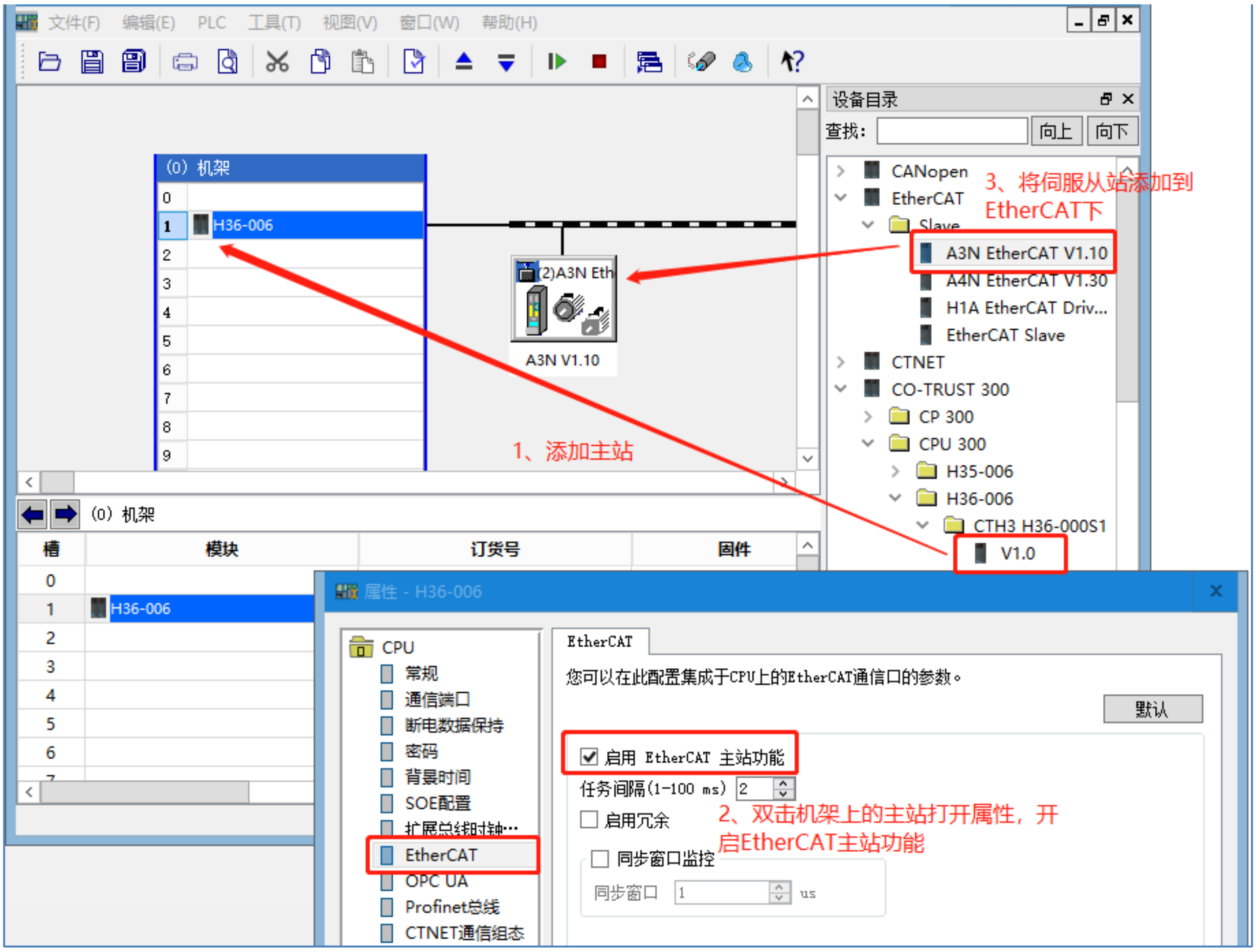
6.7.4 Example 2: Communication Axis 19 Homing Mode Example

Select "Hardware Configuration" on the project manager interface of H36-006 CPU site to perform hardware configuration.

Step 1: Add EtherCAT master and configure slave servo

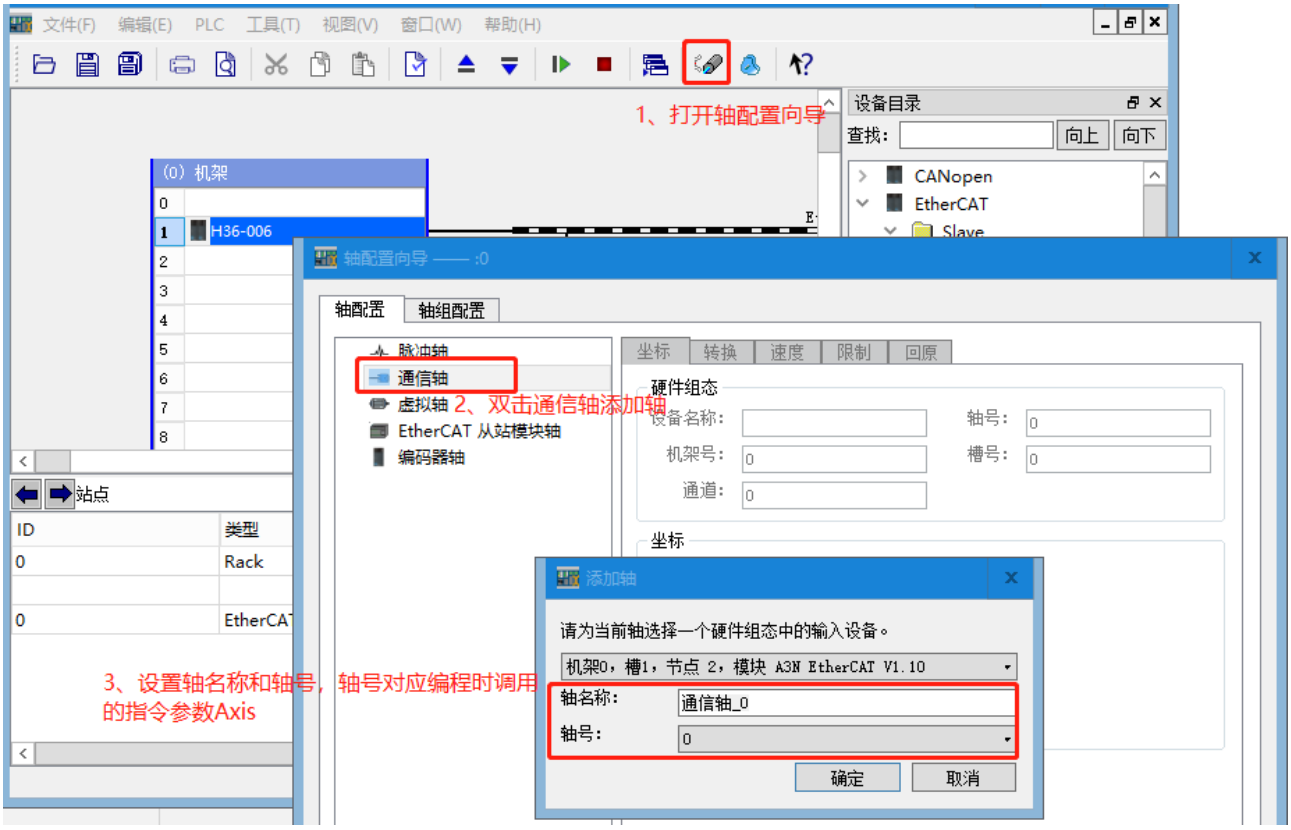
1. In the hardware configuration interface, sequentially expand the "CPU 300" folder under the "CO-TTRUST 300" project tree and add the CPU to slot 1 of the rack.

2. Before configuring the communication axis, it is necessary to add the corresponding EtherCAT slave station in the hardware configuration interface. The operation is as follows: right-click on H36 on the rack, select "Properties" - "EtherCAT", and click "Enable EtherCAT Master Station Function". After confirmation, you can see the "EtherCAT BUS" in the hardware configuration interface. Click on "EtherCAT" "slave" under the project tree on the right, select the slave and drag it to "EtherCAT BUS".

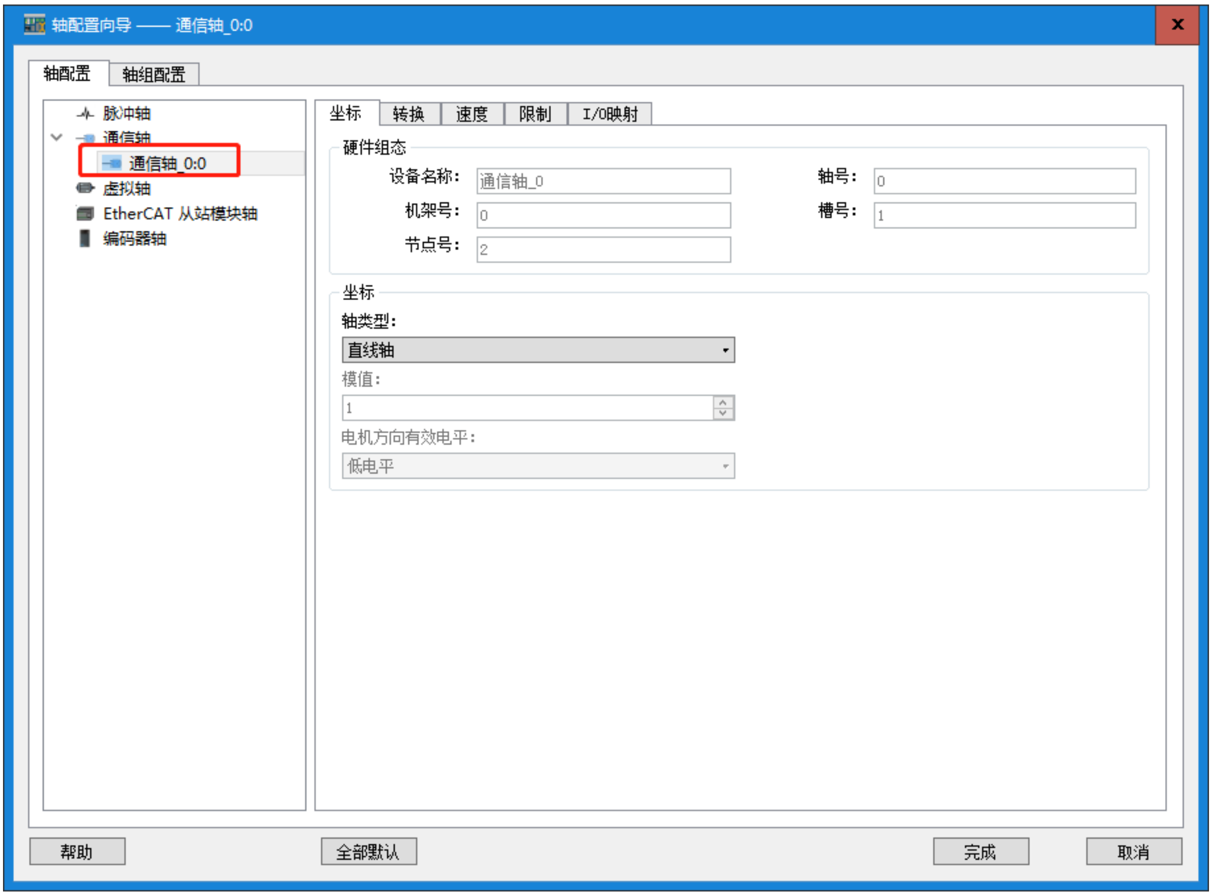


Step 2: Configure the axis

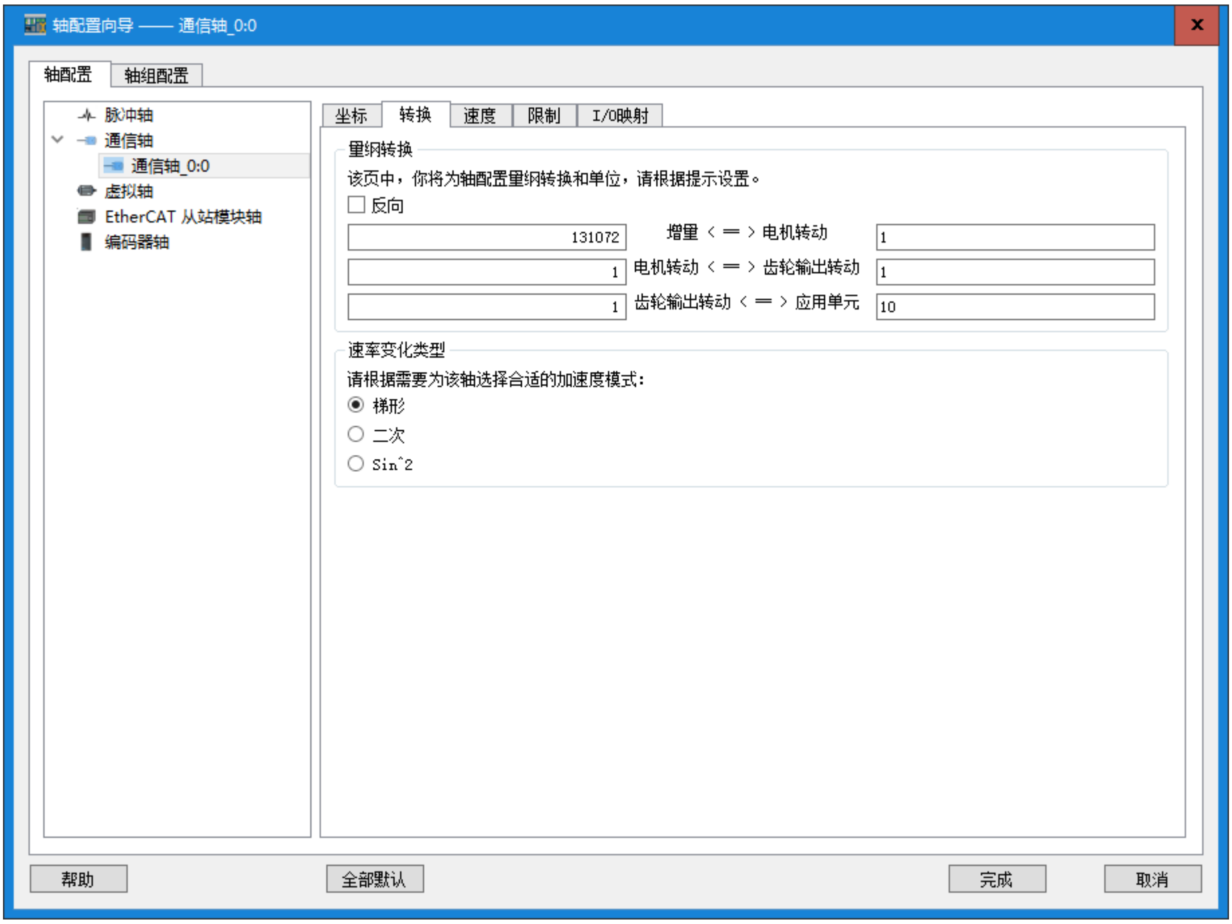
1. Follow the steps below to add a communication axis, note that only one communication axis can be added for each slave station.



2. The added communication axis interface is shown below:

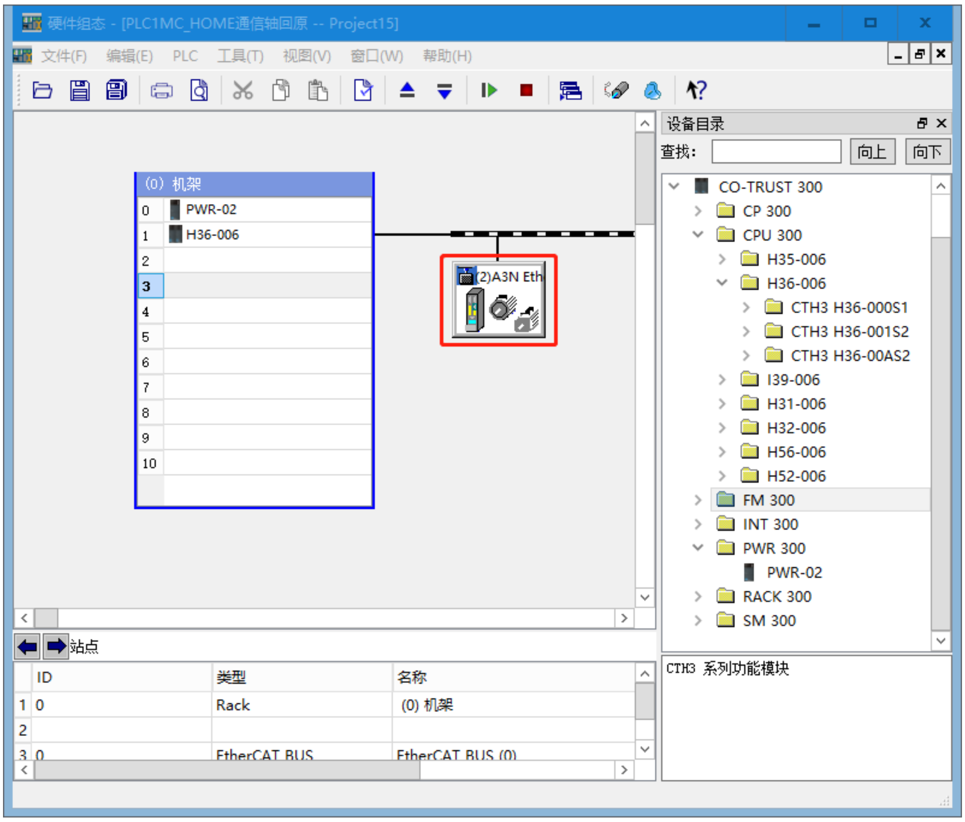


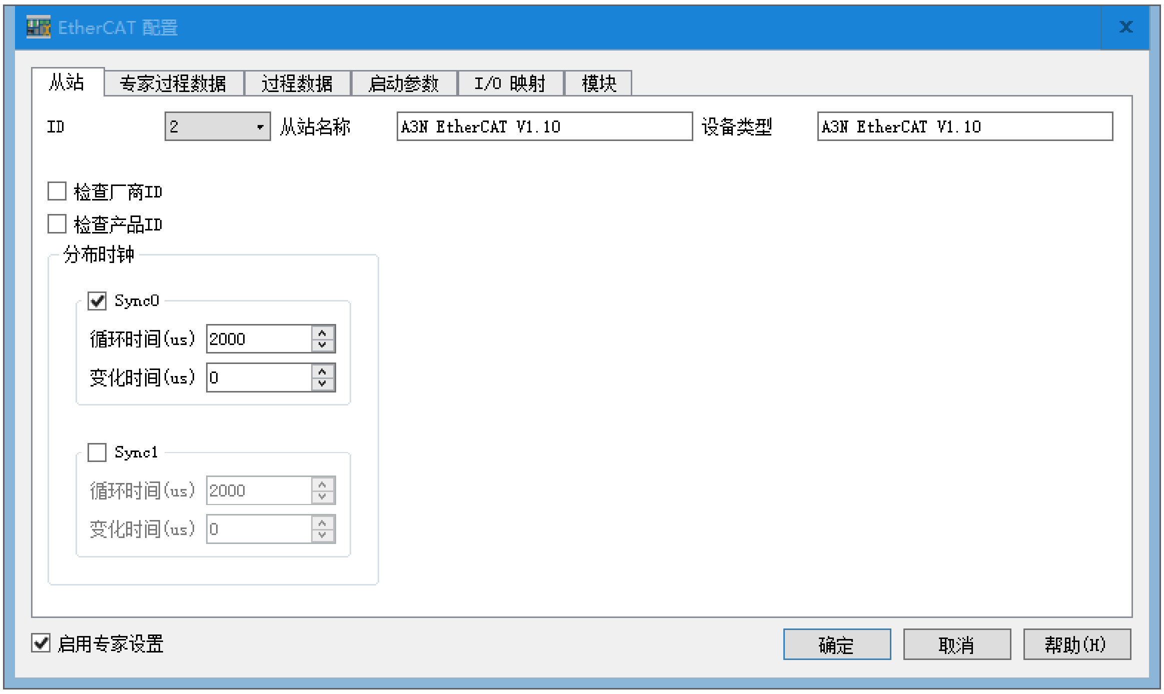
3. In the "Conversion" interface, you can set the type of program conversion and rate change according to the prompts. Set the "application unit" in the figure to 10. When the program instruction requires the servo to run 1 unit, the gear end will select 1/10 turn, the servo motor end will rotate 1 * 1/10 turn, and so on. After setting the corresponding parameters (i.e. electronic gear ratio) according to the actual mechanical structure, distance commands can be input based on the physical unit of motion distance of the application system, making parameter control intuitive and easy to understand.



4. When configuring the communication axis return function using the MC_Some command, it is necessary to configure 16 # 6098:0 (Homing mode), 16 # 6099:1 (Homing search speed), and 16 # 6099:2 (Homing crawling speed). The operation is as follows:

Firstly, double-click the slave station attached to "EtherCAT BUS" to display the EtherCAT configuration interface.

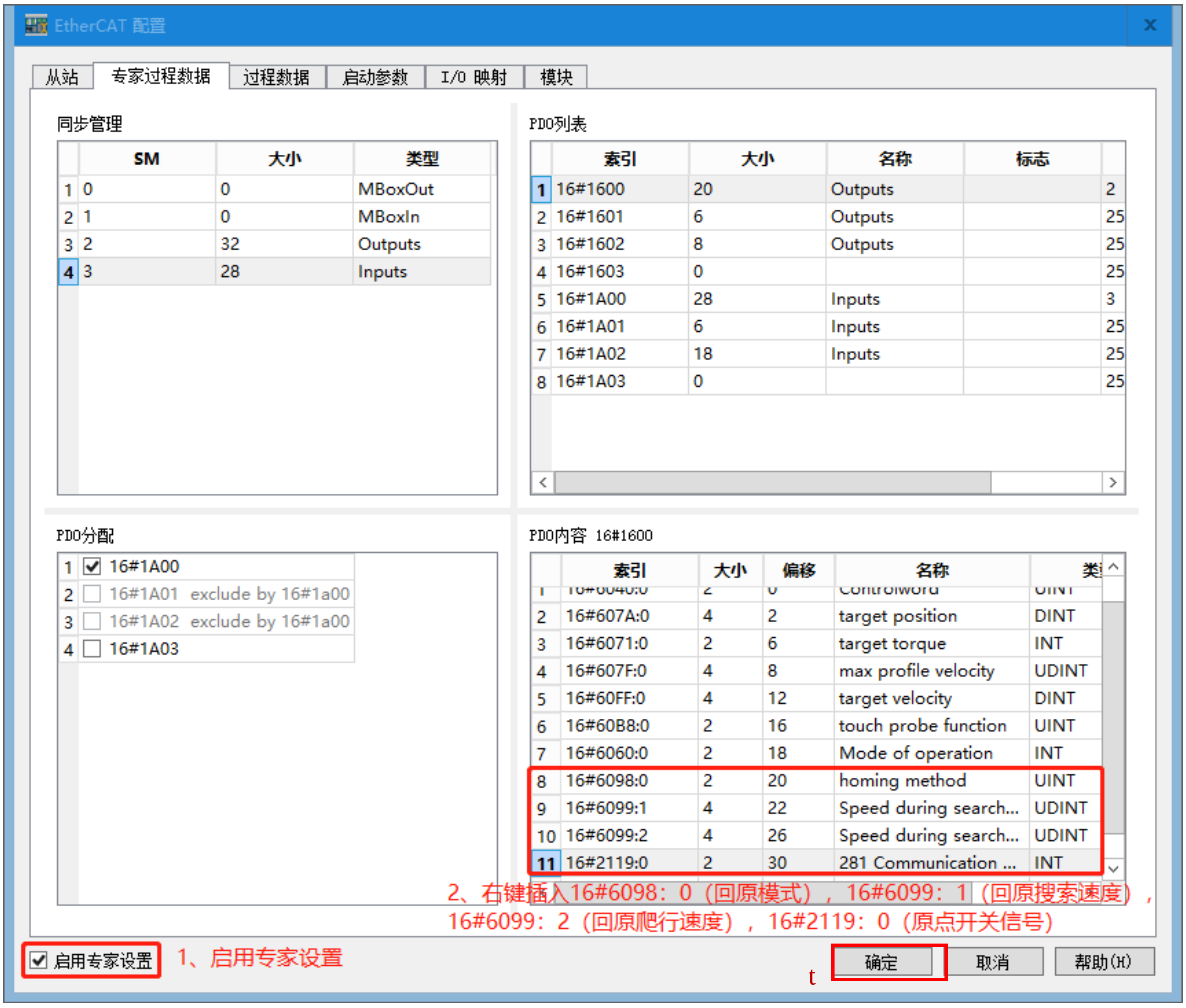




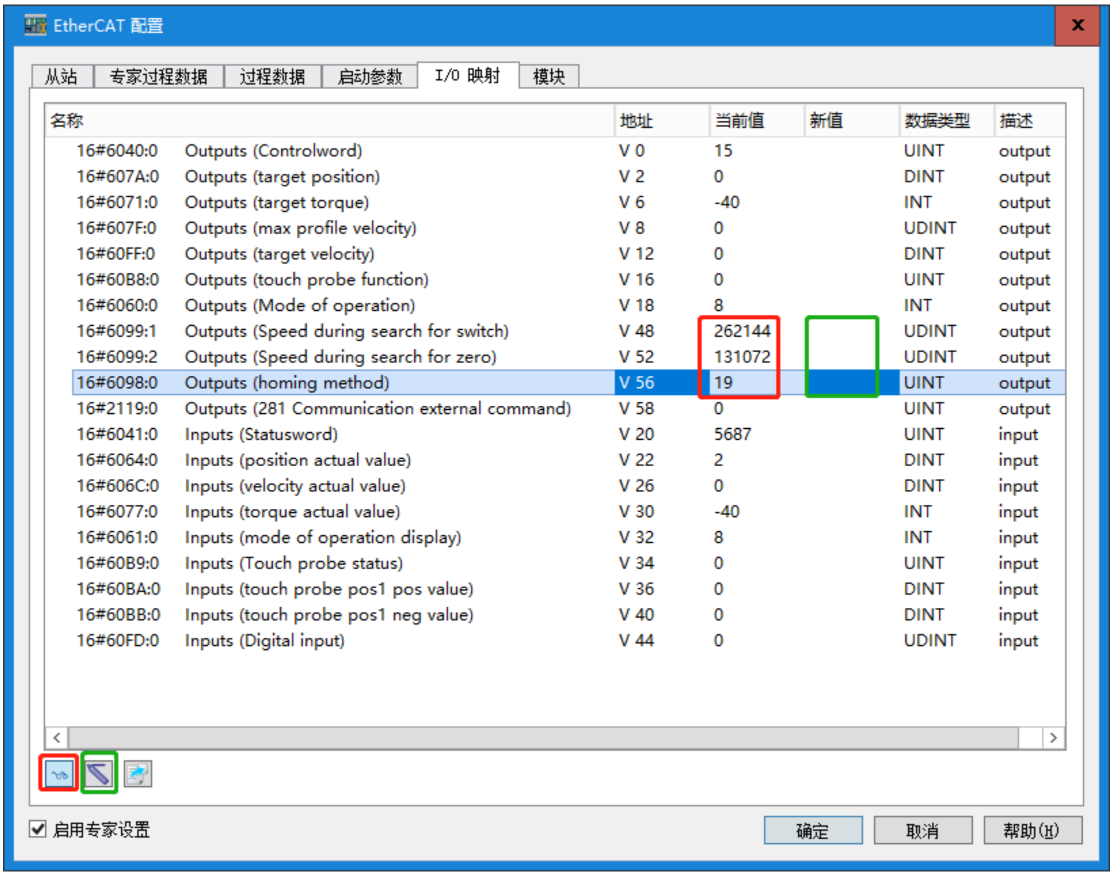
5. Click on 'Enable Expert Settings' and right-click on the PDO content to insert three parameters: 16 # 6098:0 (Return to Origin Mode), 16 # 6099:1 (Return to Origin Search Speed), and 16 # 6099:2 (Return to Origin Crawling Speed).

To use the 19th Homing mode, an origin switch signal is also required, which requires adding 16 # 2119:0 and enabling the origin switch signal by changing its bit11 value. The method of adding 16 # 2119:0 is consistent with the method of adding back to the original mode and returning to the original speed mentioned above.

There are two types of command units for search speed and crawling speed, which are PPS (pulse count/second) and rpm (revolutions per minute). Please refer to the driver manufacturer's manual for details.



6. After completing the parameter addition, click on the I/O mapping page and click on the glasses icon in the lower left corner to view the current value. Here, we need to set the Homing mode to 19, the Homing search speed to 262144 (i.e. speed 20), and the Homing crawl speed to 131072 (i.e. speed 10). Write the desired value in the new value column in the green box in the figure below, and then click the pencil button in the lower left corner to perform the writing. It can also be written in the state table to compare the address mapping of each parameter. For example, in the following figure, the 16 # 2119:0 origin switch signal corresponds to VW58, the return mode corresponds to VW56, the search speed VD48, and the crawling speed VD52.

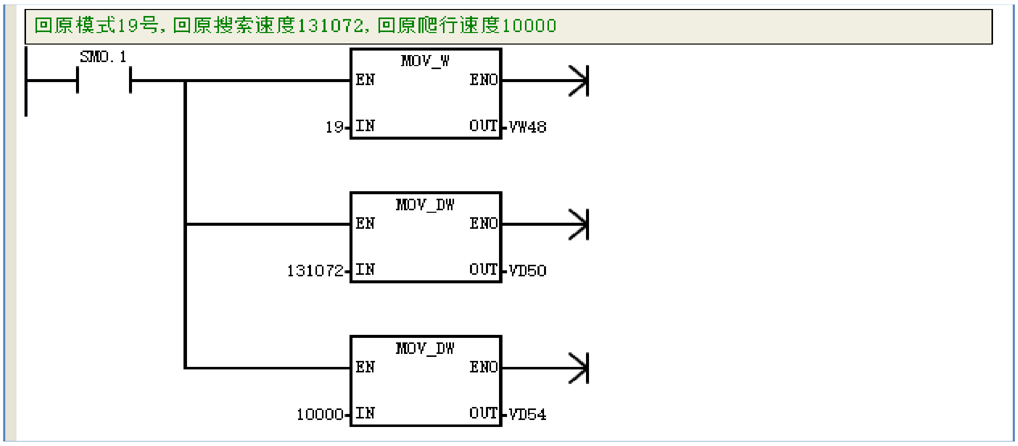


名称		地址	当前值	新值	数据类型	描述
16#6040:0	Outputs (Controlword)	V 0	15		UINT	output
16#607A:0	Outputs (target position)	V 2	0		DINT	output
16#6071:0	Outputs (target torque)	V 6	-40		INT	output
16#607F:0	Outputs (max profile velocity)	V 8	0		UDINT	output
16#60FF:0	Outputs (target velocity)	V 12	0		DINT	output
16#60B8:0	Outputs (touch probe function)	V 16	0		UINT	output
16#6060:0	Outputs (Mode of operation)	V 18	8		INT	output
16#6099:1	Outputs (Speed during search for switch)	V 48	262144		UDINT	output
16#6099:2	Outputs (Speed during search for zero)	V 52	131072		UDINT	output
16#6098:0	Outputs (homing method)	V 56	19		UINT	output
16#2119:0	Outputs (281 Communication external command)	V 58	0		UINT	output
16#6041:0	Inputs (Statusword)	V 20	5687		UINT	input
16#6064:0	Inputs (position actual value)	V 22	2		DINT	input
16#606C:0	Inputs (velocity actual value)	V 26	0		DINT	input
16#6077:0	Inputs (torque actual value)	V 30	-40		INT	input
16#6061:0	Inputs (mode of operation display)	V 32	8		INT	input
16#60B9:0	Inputs (Touch probe status)	V 34	0		UINT	input
16#60BA:0	Inputs (touch probe pos1 pos value)	V 36	0		DINT	input
16#60BB:0	Inputs (touch probe pos1 neg value)	V 40	0		DINT	input
16#60FD:0	Inputs (Digital input)	V 44	0		UDINT	input

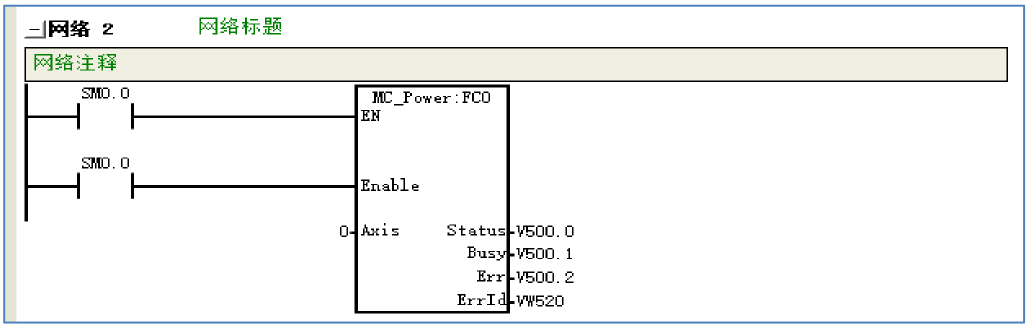
Step 3: Call MC_Home instruction (origin regression instruction) in the main program to program.

Open the OB1 program block, expand ct_plcopentlib, and call the instructions in the library to complete a simple single axis usage control.

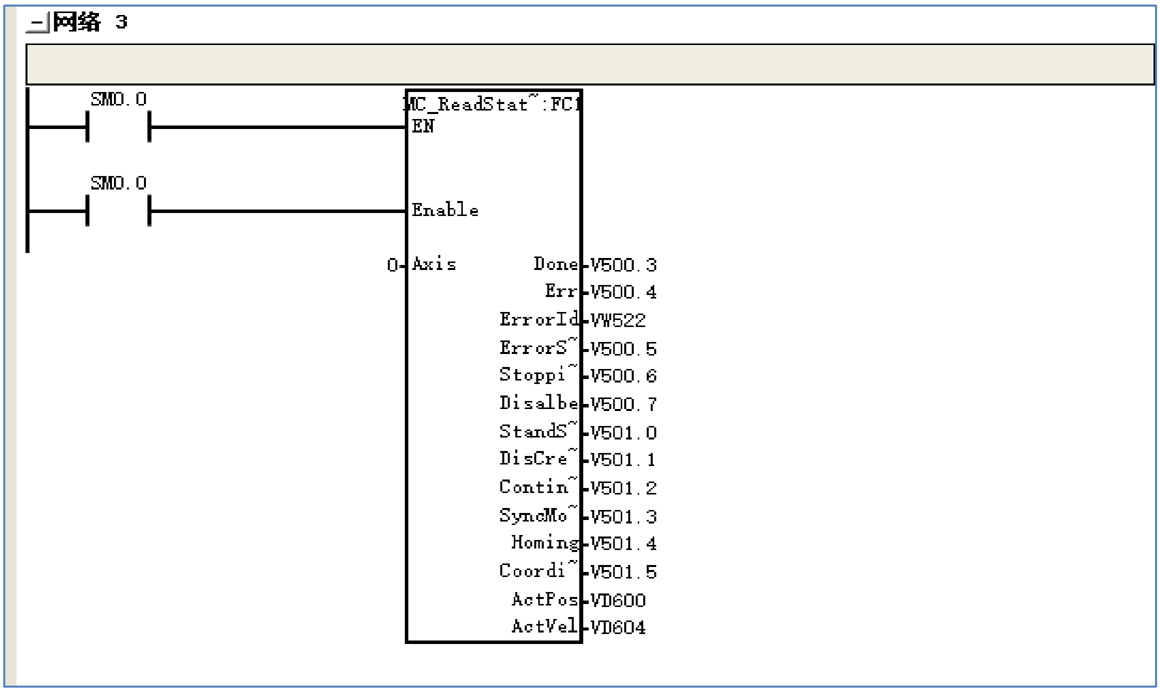
Network 1: Return to the original mode on the 19th (refer to the forward origin signal switch).



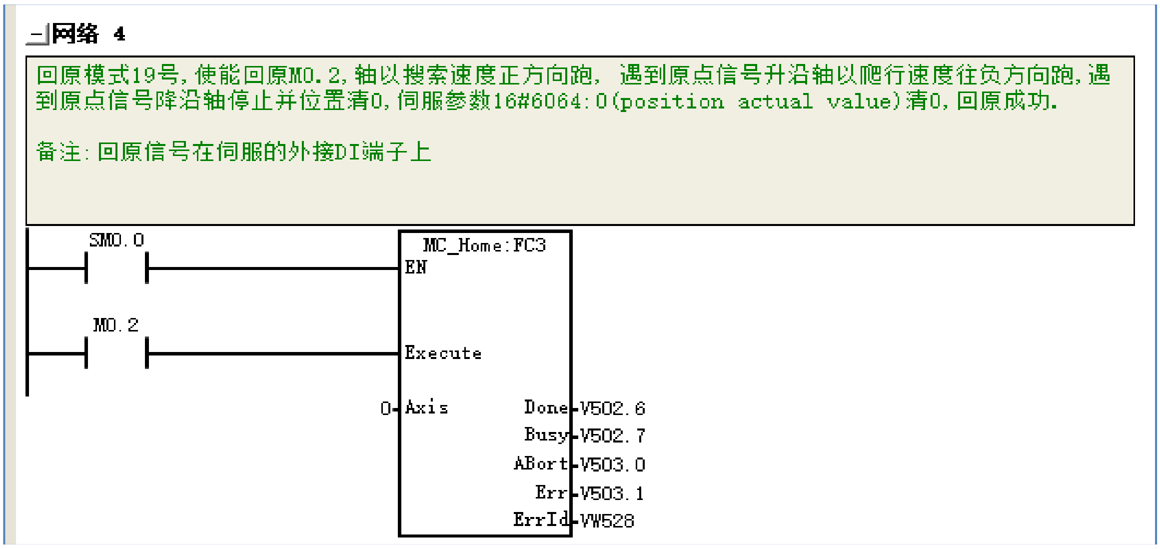
Network 2: Axis Enable Operation



Network 3: Read axis status



Homing mode 19, Homing signal I0.2, the axis runs in the positive direction of the search speed. When encountering the rising edge of the origin signal, the axis runs in the negative direction at crawling speed. When encountering the falling edge of the origin signal, the axis stops moving and its position is cleared to 0. Servo parameter 16 # 6064:0 (position actual value) is cleared to 0.

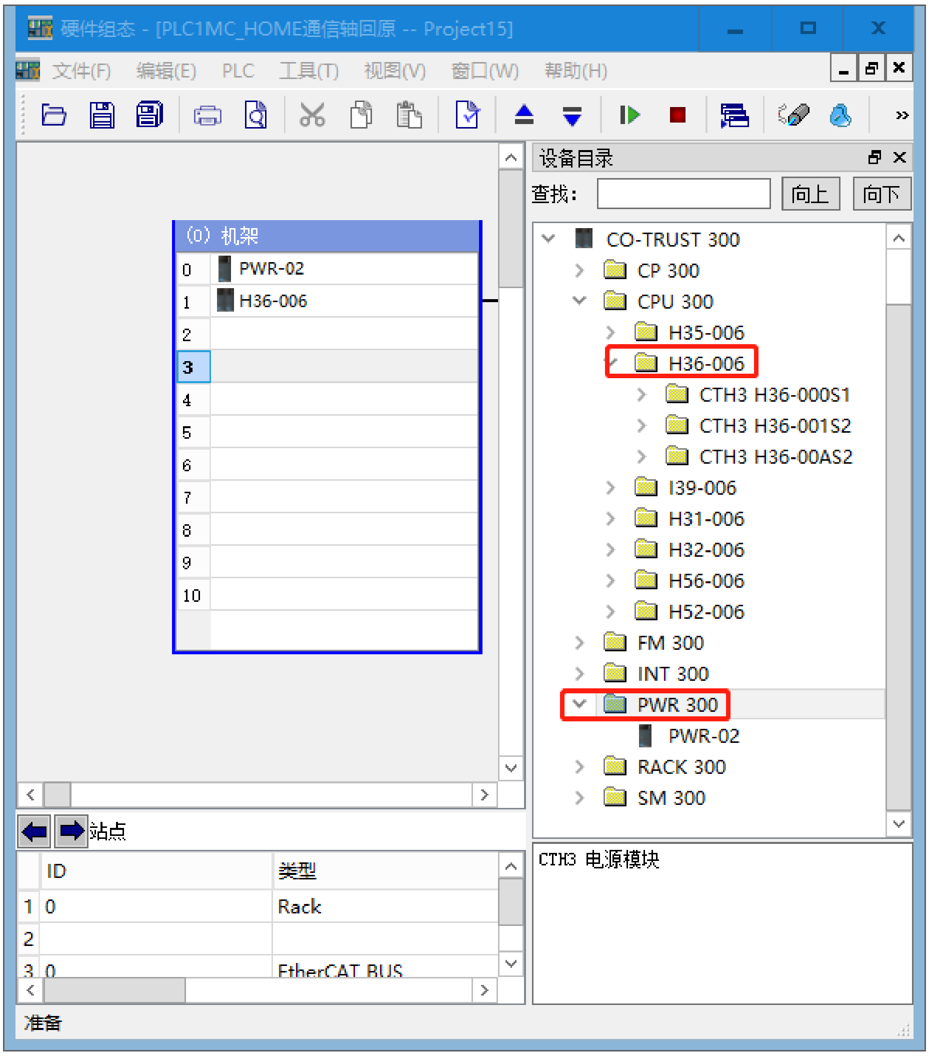


6.7.5 Example 3: Communication axis special origin regression example

Select "Hardware Configuration" on the project manager interface of H36-006 CPU site to perform hardware configuration.

Step 1: Add power module and CPU

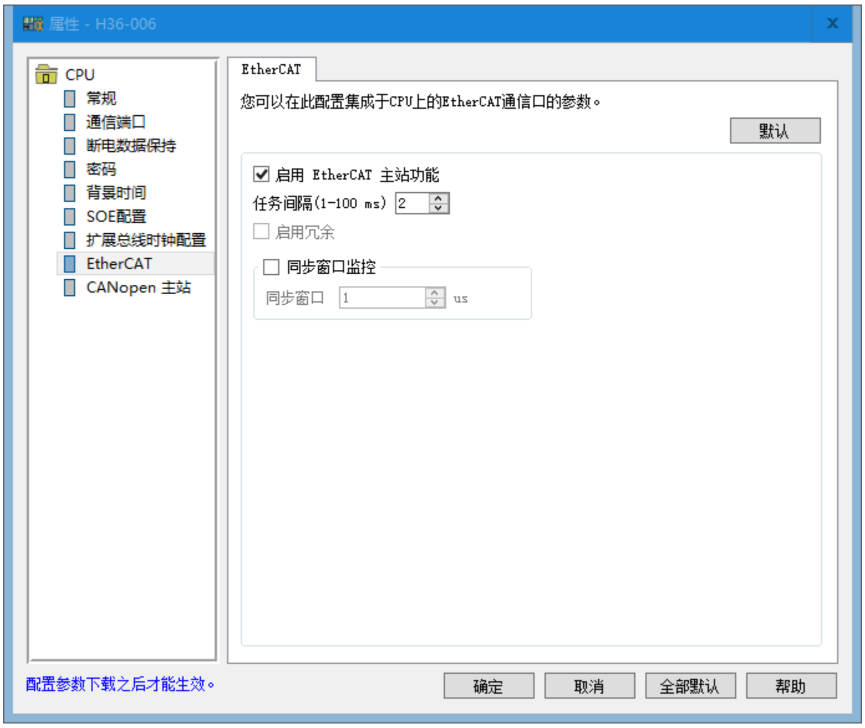
In the hardware configuration interface, sequentially expand the "PWR 300" and "CPU 300" folders under the "CO-TTRUST 300" project tree, add power modules and CPUs. Power modules can only be placed in slot 0 of the rack, and CPUs can only be placed in slot 1 of the rack, as shown in the following figure.



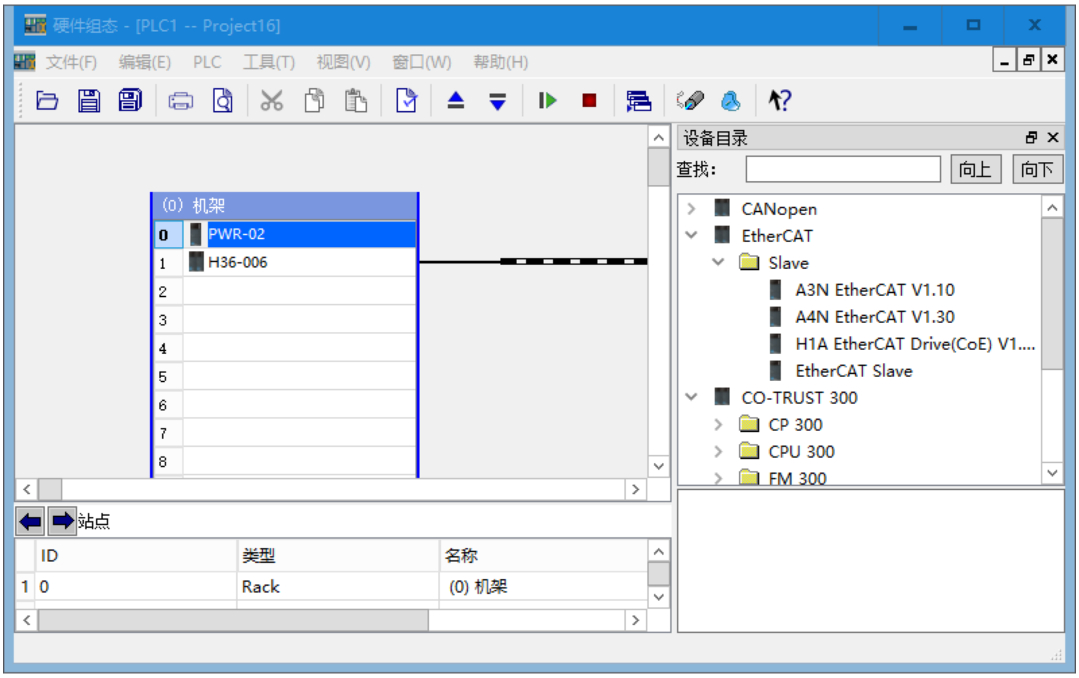
Step 2: Configure the axis

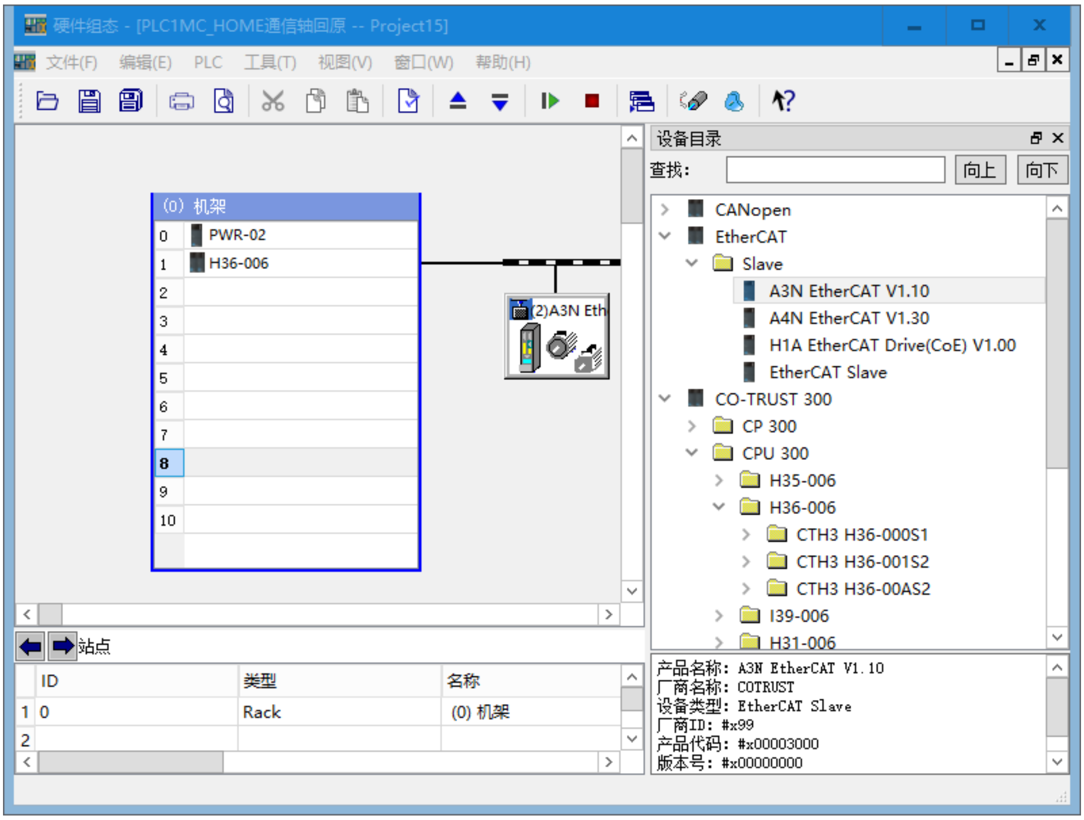
Before adding a communication axis, it is necessary to add the corresponding EtherCAT slave in the hardware configuration interface, otherwise the communication axis cannot be added. The specific operation is as follows:

1) Right click on H36 on the rack, select "Properties" "EtherCAT", and click on "Enable EtherCAT Master Station Function" on the page.

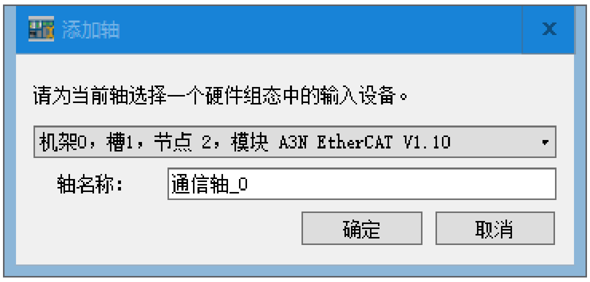


2) After confirmation, you can see the "EtherCAT BUS" on the hardware configuration interface. Click on "EtherCAT" on the right, select the slave station, and drag it to "EtherCAT BUS".



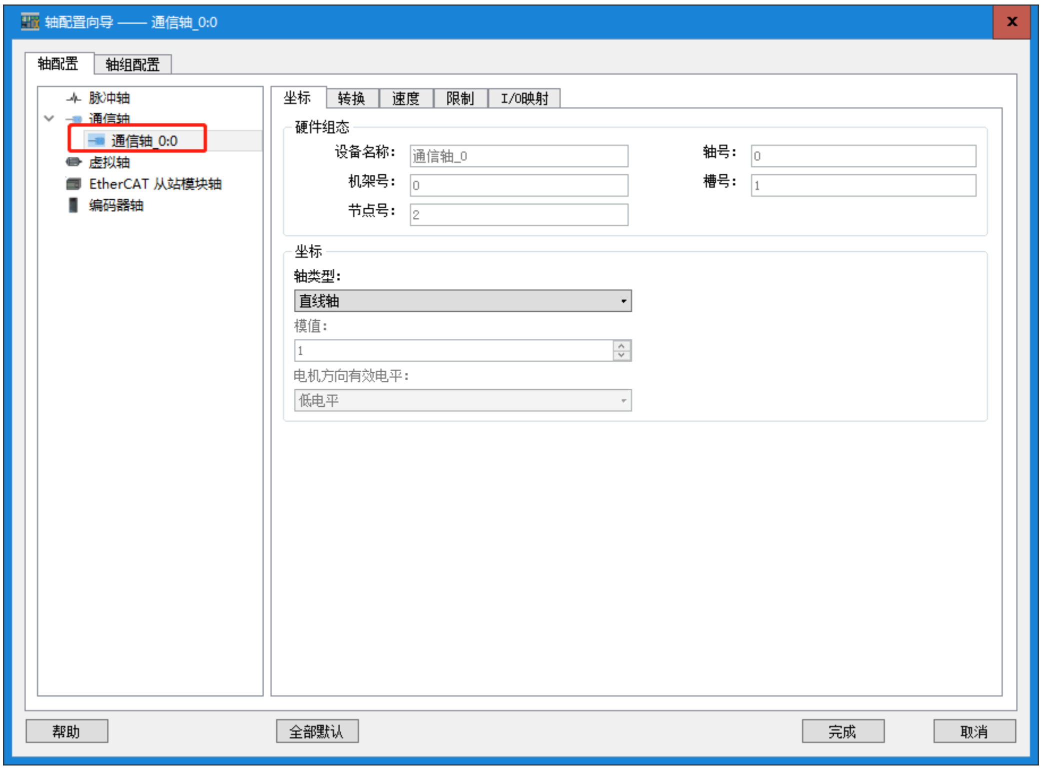


3) After completing the above configuration, select "Axis Wizard Configuration" in the "Tools" option, double-click "Communication Axis" in the Axis Wizard Configuration interface to add a communication axis. The "Axis Name" can be modified, and click "OK" to complete the addition.



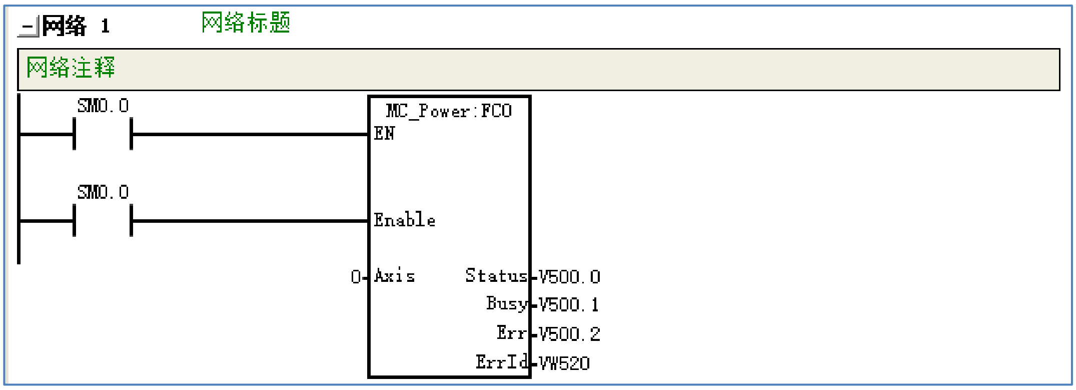
Note: Only one communication axis can be added to a slave station.

4) The added communication axis interface is shown below:

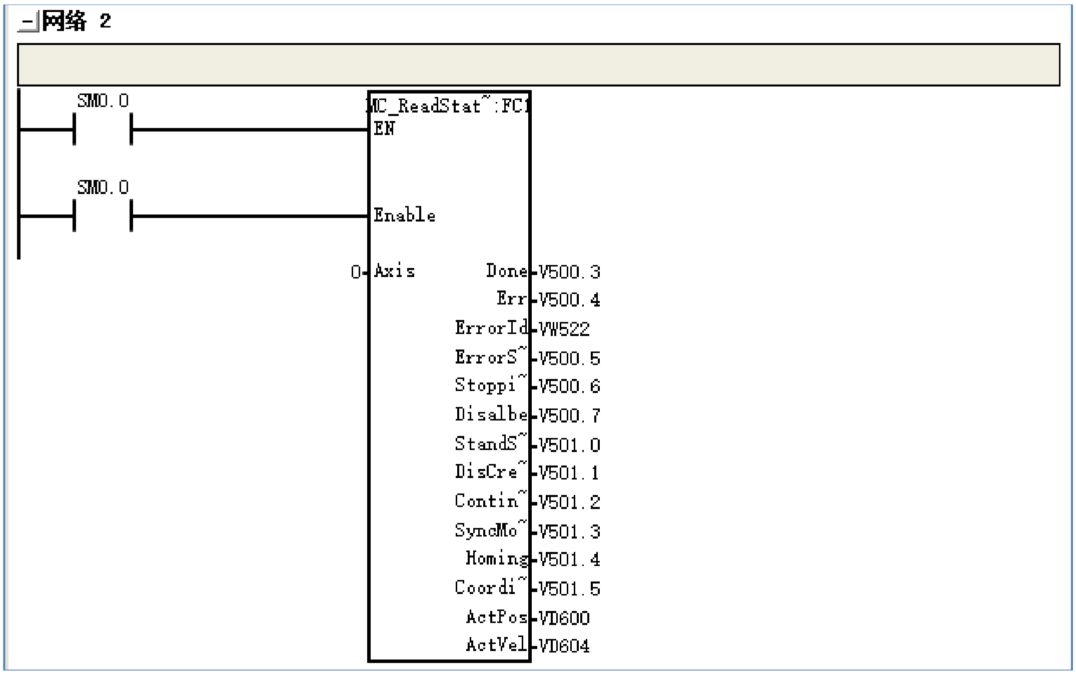


Step 3: Call SMC_Home instruction (special origin regression instruction) in the main program for programming.

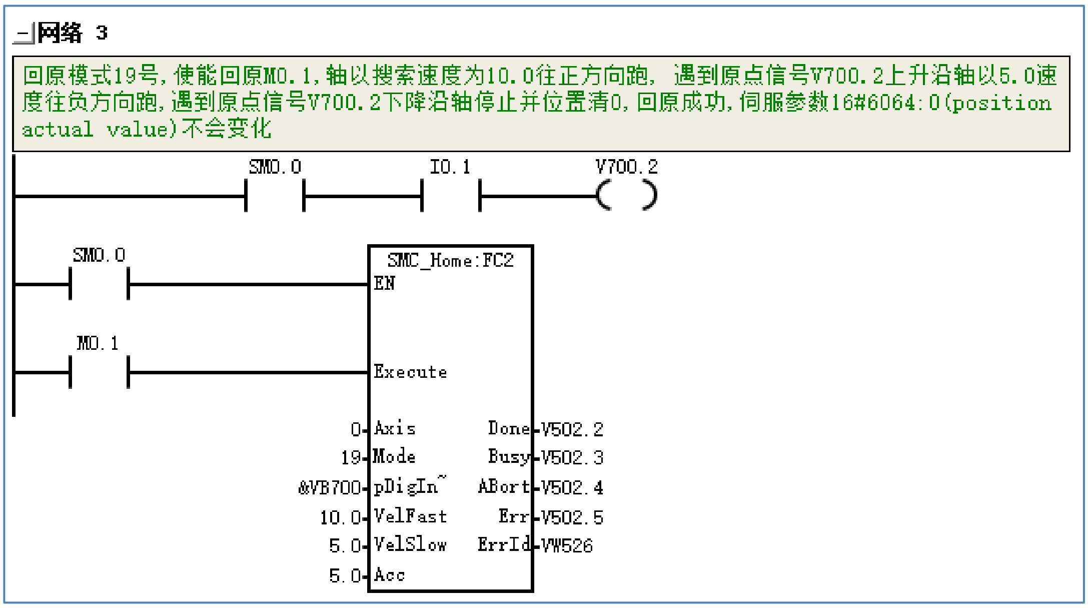
Network 1: Enable the axis with Axis ID set to 0.



Network 2: Read axis status.



Network 3: Call the SMC_Home special origin regression command, select the return to origin mode 19, with a search speed of 10.0 for the axis and an acceleration of 5.



Appendix

7

7.1 Ordering Information

Table7-1 List of Product Ordering Information

Order number	Specification Description
Main control module	
CTMC 216-2AE35-0X40	Chasing Flying Clipper CPU, abbreviated as M226ES. Supports 16 EtherCAT bus axes, 10 * 200kHz operation control outputs, and 24DI/16DO transistor drain type outputs. 200KB+48KB+48KB program space, 64KB data space, 24V DC power supply. Two PPI and free communication ports can be switched, one EtherNET communication port and one EtherCAT bus communication port. Supports CAN, RS485, and analog expansion boards. Support online firmware upgrade for network ports. Support C programming language. Support Trace tracing function. By using the PLCopen standard instruction library, single axis operation control (such as positioning, speed, and return to the original) functions can be achieved, as well as multi axis multi-channel linear/arc interpolation, continuous interpolation, electronic gears, electronic camshafts, and chase shear functions.
Digital Expansion Module	
CTH2 221-1BF32	SM221 digital input module, 8 channel inputs, 24V DC
CTH2 221-1BH32	SM221 digital input module, 16 channel inputs, 24V DC
CTH2 221-1BL32	SM221 digital input module, 32 channel inputs, 24V DC
CTH2 222-1BF32	SM222 digital output module, 8 channel transistor outputs, 24V DC, 0.5A (output protection)
CTH2 222-1DF32	SM222 digital output module, 8 channel t transistor NPN outputs, 24VDC, 0.5A
CTH2 222-1BH32	SM222 digital output module, 16 channel transistor outputs, 24V DC, 0.5A (output protection)
CTH2 222-1BL32	SM222 digital output module, 32 channel transistor outputs, 24V DC, 0.5A (output protection)
CTH2 222-1HF32	SM222 digital output module, 8-channel relay outputs, 2A
CTH2 222-1HH32	SM222 digital output module, 16 channel relay outputs, 2A
CTH2 223-1BF32	SM223 digital input/output module, 4 channel DC 24V input, 4 channel transistor outputs, 24V DC, 0.5A (output protection)
CTH2 223-1BH32	SM223 digital input/output module, 8 channel DC 24V input, 8 channel transistor outputs, 24V DC, 0.5A (output protection)
CTH2 223-1BL32	SM223 digital input/output module, 16 channel DC 24V inputs, 16 channel transistor output, 24V DC, 0.5A (output protection)
CTH2 223-1HF32	SM223 digital input/output module, 4 channel DC 24V inputs, 4 channel relay outputs, 2A
CTH2 223-1PH32	SM223 digital input/output module, 8 channel t DC 24V inputs, 8channel relay outputs, 2A
CTH2 223-1PL32	SM223 digital input/output module, 16 channel DC 24V inputs, 16 channel relay outputs, 2A
CTH2 231-0HC32	SM231 analog input module, 4 channel t, 0-20 mA current or ± 5V, ± 2.5V, 0-10V, 0-5V voltage inputs, isolated 12 bit precision
CTH2 231-0HF32	SM231 high-precision analog input module, 8 channel t inputs, voltage input, optically isolated 16 bit precision

CTH2 231-1HF32	SM231 high-precision analog input module, 8 channel inputs, current input, optically isolated 16 bit precision
CTH2 231-5HF32	SM231 analog voltage input module, 8 channel, ± 2.5V, 0-10V, 0-5V voltage input, with two channels of optional 0-20mA current input, isolated 12 bit precision
CTH2 231-7PB32	SM231 thermistor temperature input module, 2 channel RTD, isolated 16 bit precision
CTH2 231-7PC32	SM231 thermistor temperature input module, 4 channel RTD, isolated 16 bit precision
CTH2 231-7PD32	SM231 thermocouple temperature input module, 4 channel TC, J/K/R/S/T/E/N, isolated 16 bit precision
CTH2 231-7PF32	SM231 thermocouple temperature input module, 8 channel TC, J/K/R/S/T/E/N, isolated 16 bit accuracy
CTH2 231-7TD32	SM231 thermocouple PID module, 4 channel J/K type, with intelligent PID, isolated 16 bit accuracy
CTH2 231-7TF32	SM231 thermocouple PID module, 8 channel J/K type, with intelligent PID, isolated 16 bit accuracy
CTH2 231-7HF32	SM231 8-channel current input, 0-20mA/4-20mA with intelligent PID, isolated 16 bit precision
CTH2 231-7ND32	SM231 hybrid temperature input module, 2-channel NTC or PT100, 2-channel 0-20mA current or ± 5V, ± 10V, 0-10V, 0-5V voltage input, isolated 16 bit precision
CTH2 231-7NF32	SM231 thermistor temperature input module, 8NTC/PT100, isolated 16 bit precision
CTH2 231-7WA32	SM231 weighing module, 1-channel sensor signal input, sampling frequency 50Hz, 0.01% module accuracy, 6V DC 150MA excitation power output per channel, isolated 16 bit accuracy
CTH2 232-0HB32	SM232 analog output module, 2 channel, ± 10V voltage or 0-20mA current output, isolated voltage with 12 bit accuracy or current with 11 bit accuracy
CTH2 232-0HD32	SM232 analog output module, 4 channel, ± 10V voltage or 0-20mA current output, isolated voltage with 12 bit accuracy or current with 11 bit accuracy
CTH2 235-0KD32	SM235 analog input/output module, 4 channel voltage or current input/1-channel voltage or current output, isolated voltage with 12 bit accuracy or current with 11 bit accuracy
CTH2 253-1BH32	SM253 motion control module, 2 channel single-phase or AB phase HSC input, 200kHz, 2-axis PTO/PWM output, 200kHz, COTRUST motion control library
CTH2 277-0AA32	SM277A Profibus DP Slave Interface Module, 12M Communication Rate, Optical Isolation
CTH2 277-0AB32	SM277B Profibus DP slave module, 1.5M communication rate, optoelectronic isolation
CTH2 277-0AC32	SM277C CAN slave module, comes with 8DI/6DO, optoelectronic isolation, and can expand up to 7 modules
CTH2 277-0PN32	SM277 Profinet slave module, 100M communication rate, expandable to 8 modules
Expansion board	
CTH2 AMS-03S1-EB	EBH AMS-03 Analog Input/Output Expansion Board, 2 channel * 12 bit Precision Voltage Input, 1 channel * 12 bit Precision Voltage/Current Output
CTH2 AMS-06S1-EB	EBH-AMS-06 Analog Input/Output Expansion Board, 4 channel * 12 bit Precision Voltage Input, 2 channel * 12 bit Precision Voltage Output
CTH2 AMS-06S2-EB	EBH-AMS-06 Analog Input/Output Expansion Board, 4 channel * 12 bit Precision Voltage Input, 2 channel * 12 bit Precision Current Output
CTH2-CAN-01S2-EB	CTH2 CAN main station communication expansion board, 1Mbps,

	optoelectronic isolation, configuration mode supports CAN hardware configuration block and EDS file
CTH2 RS485-01S1-EB	One channel RS485 communication expansion board isolation type

7.2 Fault diagnosis

When the system malfunctions, the CPU performs the following operations:

1) Enter STOP mode
2) Turn on the SF (red) LED indicator light and STOP indicator light
3) Disconnect output

Please first check if the following conditions are met:

1) Is the main control module and expansion module powered normally.
2) Are the screws and connectors of the I/O terminals of the main control module and expansion module loose.
3) Check the connection of communication cables to ensure accuracy.
4) Unable to search for PLC, please check communication settings, such as changing baud rate, connecting serial port or IP, etc., and search again.

In addition to the above methods, diagnostic information can also be read through MagicWorks PLC, or the status of the PLC's LED indicator lights can be checked for any abnormalities in the PLC itself and externally.

7.2.1 Diagnosis through MagicWorks PLC

The fault state will continue until the error is cleared. The diagnostic information reading method is: open MagicWorks PLC software → double-click to open the project interface → select the menu item "PLC" → "Information" to view the error information.

Table7-2 Diagnostic function

Supported event types	**Specific event code and description**	
	code	**Event description**
CPU non fatal error	0x00	No errors
	0x01	Enable HSC (reserved) before executing the HDEF box
	0x02	Input interrupt allocation conflict, allocated to points already assigned to HSC (reserved)
	0x03	Input allocation conflict, allocate to HSCs that have already been assigned to input interrupts or other HSCs (reserved)
	0x04	Attempt to execute ENI, DISI, or HDEF instructions in interrupt routines
	0x05	Attempt to execute a second HSC/PLS with the same number before completing the first HSC/PLS, causing an interrupt routine conflict with the HSC/PLS in the main program (reserved)
	0x06	Indirect addressing error
	0x07	TODW (Daily Write Time) or TODR (Daily Read Time) data error

Supported event types	Specific event code and description	
	code	Event description
	0x08	Exceeding the maximum number of nested user subroutines
	0x09	Simultaneously execute XMT/RCV instructions in port 0
	0x0A	Attempt to redefine HSC by executing another HDEF instruction for the same HSC (reserved)
	0x0B	Simultaneously execute XMT/RCV instructions in port 1
	0x0C~ 0x0E	retain
CPU non fatal error	0x0F	Encountering illegal numerical values in comparing touch point commands
	0x10~ 0x13	retain
	0x13	Illegal PID circuit table
	0x80	The program is too large for the CPU to generate executable code. Please reduce the program size
	0x81	retain
	0x82	Illegal instructions. Check instruction mnemonic 0
	0x83	Missing MEND or instruction not allowed in the main program. Add MEND instruction or remove incorrect instruction
	0x85	Lack of FOR. Add FOR instruction or remove NEXT instruction
	0x86	Lack of NEXT. Add NEXT instruction or remove FOR instruction
	0x87	Missing tags (LBL, INT, subroutine). Add appropriate labels
	0x88	Missing RET or instruction not allowed in subroutine. Add RET or remove incorrect instructions at the end of the subroutine
	0x89	Lack of RETI or instruction not allowed in interrupt routine. Add RETI or remove incorrect instructions at the end of the interrupt routine
	0x8B	Illegal JMP from SCR section or illegal JMP from SCR section
	0x8C	Duplicate tags (LBL, INT, SBR). Rename one of the tags
	0x8D	Illegal tags (LBL, INT, SBR). Verify that the allowed number of tags has not been exceeded
	0x90	Illegal parameters. Verify the parameters allowed by the instruction
	0x91	Range error (including address information). Check the range of operands
	0x92	Instruction count field error (including count information). Verify maximum count
	0x93	Exceeding FOR/NEXT nesting levels
	0x94	Write range error to non-volatile memory using address information
	0x95	Missing LSCR instruction (load SCR)
	0x96	Missing SCRE instruction (SCR ends) or an instruction that is not allowed to be used before SCRE
	0x97	The user program includes EU/ED directives without and with numbers
	0x98	Attempt to perform runtime editing in programs equipped with EU/ED instructions without numbers
	0x99	Too many hidden program segments

Supported event types	Specific event code and description	
	code	Event description
	0x9A	Attempt to switch to free port mode during user interruption
	0x9B	Illegal index (string operation, with a starting position value of 0 specified in the operation)
CPU fatal error	0x00	There are no serious errors present
	0x01	retain
	0x02	retain
	0x03	Scan watchdog timeout error
	0x04~ 0x0F	retain
	0x10	Internal software error
	0x11	Comparing indirect addressing errors of contact points
	0x12	Illegal floating point value error in comparing nodes
	0x13	retain
	0x14	Comparison node range error
	0x15	Power fluctuation alarm
Monitor and record once per scanning cycle		
Module diagnostic events Refer to Table 8-3	0x00	Module without malfunction
	0x01	Module busy
	0x02	Module timeout not responding
	0x03	Module type mismatch
	0x04	Module version mismatch
	0x05	Software error
	0x06	Module waiting flag timeout
	0x07	Bus response error
	0x08	Bus CRC check error
	0x10	Memory offset exceeds the range
	0x11	The module is not ready
	0x12	Module configuration error
	0x13	The module does not support this instruction
	0x15	Module internal diagnosis
	0x16	Module without power supply

Table7-3 Special storage area diagnostic function

SMB	Error description
SMB8	Module 1 Identification Register
SMB9	Module 1 Error Register
SMB10	Module 2 Identification Register
SMB11	Module 2 Error Register
SMB12	Module 3 Identification Register
SMB13	Module 3 Error Register
SMB14	Module 4 Identification Register
SMB15	Module 4 Error Register
SMB16	Module 5 Identification Register
SMB17	Module 5 Error Register
SMB18	Module 6 Identification Register
SMB19	Module 6 Error Register
SMB20	Module 7 Identification Register
SMB21	Module 7 Error Register
SMB200~SMB549	Intelligent module status

7.2.2 Diagnosis through the main control module

Table7-4 LED status indicator status description

indicator light	colour	describe
LINK1 indicator light	Green	ON: EtherNet network port hardware is connected Flash: There is data exchange OFF: EtherNet network port hardware has been disconnected
LINK2 indicator light	Green	ON: EtherCAT network port hardware is connected Flash: There is data exchange OFF: EtherCAT network port hardware has been disconnected
RUN indicator light	Green	ON: System running, OFF: System stopped
STOP indicator light	Orange	ON: System stopped, OFF: System running
SF/DIAG indicator light	SF Red DIAG Orange	SF indicator light ON: system error. OFF: Normal DIAG indicator light ON: program control. OFF: Normal
RMC indicator light	Green	ON: successfully connected to remote server OFF: Remote not connected or remote not enabled

<Note>: During the power fluctuation phase from the beginning of device power failure to complete power failure, the STOP indicator light and SF indicator light will light up simultaneously, and the system will record an event.

7.3 Power budget

The M226ES unit has an internal power supply that can provide power to both the unit and the expansion unit. At the same time, M226ES provides a 5V DC logic power supply, which can be used to power any expansion unit in the system. Pay close attention to the system configuration to ensure that the CPU can provide the 5V power required by the selected expansion module. If the configuration requires more power than the power provided by the CPU, certain expansion modules must be removed or a CPU that can provide greater power must be selected.

M226ES also provides 24V DC sensor power supply, which can provide 24V DC for input points, relay coils on expansion modules, or other power sources. If the required power exceeds the sensor power budget, an external 24V DC power supply must be added to the system. If an external 24V DC power supply is required, please ensure that the power supply is not connected in parallel to the sensor power supply of M226ES. To improve electrical noise protection, it is recommended to connect to the common terminal (M) of different power sources.

7.3.1 Power requirements

The CPU unit provides 5VDC and 24V DC power supply:

Most CPU body units have an external power supply of 24V DC, which can provide 24V DC for the input points of the body and the relay coils of the expansion module.

When there is an expansion unit connected, the CPU body unit provides 5V DC power to the expansion unit. If the 5V power requirement of the expansion unit exceeds the rated output of the CPU module, the expansion unit must be removed until the requirement is within the predetermined power supply value.

The technical specifications have stipulated the provision of information regarding CPU power budget and expansion module power requirements.

Tips
If the 5V DC power supply exceeds the CPU power budget, it may not be possible to connect the maximum number of modules allowed by the CPU.

be careful
The external 24V DC power supply connected in parallel with the M226ES sensor's power supply can cause conflicts, as each power supply seeks to establish its own potential output. This conflict can shorten the lifespan or cause immediate failure of one or two power sources, followed by uncertain operations on the PLC system. This unpredictable operation can lead to personal injury and equipment damage.

The power supply for the 24V DC sensor of M226ES and any external power supply should be separately supplied to different points, while separate connection of the common power supply is allowed.

7.3.2 5V DC power supply

Table7-5 5V DC power consumption meter

CPU and DP modules	5V DC current (mA) for expanding IO	Expansion module	5V DC current consumption (mA)
CPU M226ES SM277B SM277C SM277PN	one thousand six hundred and sixty six hundred and sixty one thousand and one hundred	SM 221-1BF	57
		SM 221-1BH	79
		SM 221-1BL	179
		SM 222-1BF	57
		SM 222-1BH	79
		SM 222-1BL	174
		SM 222-1HF	68
		SM 222-1HH	115
		SM 223-1BF	57
		SM 223-1BH	73
		SM 223-1BL	115
		SM 223-1HF	58
		SM 223-1PH	89
		SM 223-1PL	150
		SM 231-0HC	87
		SM 231-0HF	87
		SM 231-1HF	87
		SM 231-5HF	87
		SM 232-0HB	87
		SM 232-0HD	87
		SM 235-0KD	87
		SM 231-7PB	87
		SM 231-7PC	87
		SM 231-7PD	87
		SM 231-7PF	87
		SM 231-7TD	87
		SM 231-7TF	87
		SM 231-7HF	87
		SM 231-7ND	87

		SM 231-7NF	87
		SM 231-7WA	140
		SM 277A	70
		SM 253 operation control module	90

Calculate the maximum input-output configuration of the CTMC M226ES system based on the power consumption in the table above.

7.3.3 24V DC power supply

Table7-6 24V DC power consumption meter

24VDC current supply (mA)		Expansion module 24V DC current consumption (mA)	
CPU M226ES SM277B SM277C SM277PN	three hundred four hundred three hundred eight hundred	SM222-1BL	54
		SM222-1HF	80
		SM222-1BF	14
		SM222-1BH	22
		SM222-1HH	159
		SM223-1BF	10
		SM223-1BH	11
		SM223-1BL	22
		SM223-1HF	40
		SM223-1PH	80
		SM223-1PL	159
		SM231-0HC	17
		SM231-0HF	31
		SM231-1HF	30
		SM231-5HF	31
		SM232-0HD	112
		SM232-0HB	61
		SM235-0KD	48
		SM231-7PB	34
		SM231-7PC	37
		SM231-7PD	130
		SM231-7PF	30
		SM231-7ND	60

		SM231-7NF	33
		SM231-7HF	37
		SM231-7TD	34
		SM231-7TF	39
		SM 231-7WA	100
		SM277A	70
		SM253 operation control module	80

7.3.4 Example of power budget calculation

Use the following design method to determine how many units of power configuration the main unit power supply can provide.

Table 8-5 shows the power budget calculation for an M226ES system, which includes the following contents:

- CPU M226ES
- SM223 expansion module, where SM223-1PH32 has 8 DC inputs/8 relay outputs
- SM221 expansion module, where SM221-1BF32 has 8 DC inputs
- The device has a total of 62 inputs and 42 outputs.

In this example, the CPU provides sufficient 5V DC current for the expansion module, but does not provide sufficient 24V DC current for all inputs and expansion relay coil sensors. I/O requires 536 milliamps, while the CPU only provides 300 milliamps. This device requires a minimum of 236mAh 24V DC additional power supply to operate all included 24V DC inputs and outputs.

Table7-7 Power Budget Table

CPU body power budget	**5V DC**	**24V DC**
CPU M226ES body power supply can provide current	1000mA	300mA
reduce		
System requirements	5V DC	24V DC
CPU H224X, 14 inputs	--	14* 4 mA = 56mA
4 SM 223-1PH32, requiring 5V power supply current	4*89 mA = 356mA	--
2 SM 221-1BF32, requiring 5V power supply current	2*57 mA = 114mA	--
4 SM 223-1PH32, each with 8 inputs	--	4*8*4 mA = 128mA
4 SM 223-1PH32, each with 8 relay coils	--	4*8*9 mA 288mA
2 SM 221-1BF32, each with 8 inputs	--	2*8*4 mA = 64mA
Total demand for power consumption	470mA	536mA
be equal to		
Current balance budget	5V DC	24V DC
Total current balance	530mA	-236mA

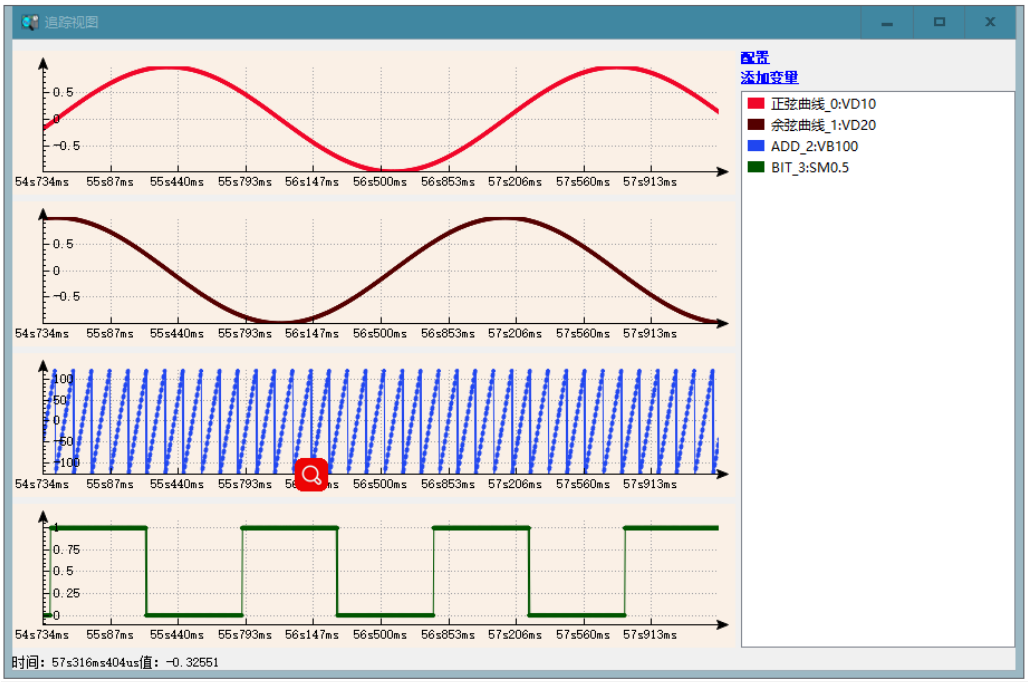
7.4 Trace tracing function

The Trace function is a newly added feature in the state table control of MagicWorks PLC software, used to configure and display application specific trace data in one or more charts.

When the application is running, the value curve records of the tracking variables can be viewed in the "Tracking View" of MagicWorks PLC. The requirement is to set up tracking configuration, transmit the tracking configuration to the PLC, and start tracking recording. The recorded data will be transmitted to the upper computer and displayed in chart form according to the configuration. You can browse data while tracking.

Activate the "Tracking" function, users can start the tracking view by clicking on the icon in the Status Table ->Menu ->View ->Tracking or in the Status Table toolbar.

As shown in the following figure: the left side is the data display chart, and the right side is the variable list. Currently, up to 8 variables and 8 charts can be added.



Configure dialog box

The dialog box includes tracking configuration for data records of the tracking project. Clicking on "Configuration" in the upper right corner of the "Tracking View" or the right mouse button will bring up the "Configuration" menu to enter the configuration dialog box.

Add variable: Add a variable to be monitored, current specification: Up to 8 variables can be added.

You can click on 'Add Variables' in the upper right corner of the' Tracking View 'to add variables

You can add it in the right-click menu of the "Tracking Records" view

Select the "Display Variables" node of one of the charts in the chart view and add it to the right-click pop-up menu

Delete variable: Right click on a variable in the tracking view to delete it, and a confirmation dialog box will pop up. After confirmation, delete the variable. Note that deleting a variable cannot be undone

Assign variables: Select a variable in the tracking view and choose "Assign to Chart" from the right-click pop-up menu to assign the variable to the specified chart. When adding variables, they will be assigned to the first chart by default. Currently, up to 8 charts can be added.

Add Chart: In the chart view, you can insert a new chart into the view by right clicking on the menu and selecting "Add Chart".

Delete Chart: In the chart view, you can select the chart you want to delete by right clicking on the menu and selecting "Delete Chart". After confirming in the dialog box, you can delete the chart. Note: After deleting a chart, it cannot be undone.

Record Settings: Click on "Tracking" in the tracking record to perform "Record Settings" on the right side

Enable Trigger: By enabling a variable to record data over a period of time, click the checkbox on the right to enable/disable it. When the trigger is enabled, you also need to configure the trigger variable, trigger edge, and post trigger sampling time.

Trigger variable: When the trigger is enabled, it is effective and used as a signal for the trigger. The effective trigger signal is a variable, a bit address in the PLC. For example, M0.0.

Trigger edge: Effective when the trigger is enabled, define trigger edge detection:

Rising edge: For BOOL trigger variables, the trigger occurs when the value changes from False to True.

Falling edge: For BOOL trigger variables, the trigger occurs when the value changes from TRUE to False.

Rising and falling edges: For BOOL trigger variables, triggering occurs when the value changes.

Sampling time after triggering: The trigger continues sampling for a specified time after triggering and stops sampling.

Sampling period: The defined time period for PLC sampling, which can be selected from the drop-down menu as PLC period, EtherCAT period, pulse period, and 1ms interrupt.

Variable setting: Configure added or existing variables

Variable address: Enter a valid address in the PLC to be monitored. When the address is invalid, an error will be reported and must be configured.

Variable Name: The alias of the current variable, which allows users to better identify the variable and can be left unconfigured.

Display format: There are 4 display formats for users to choose from: signed, unsigned, floating point, and bit.

Curve color: When adding variables, a random color will be generated, and users can also modify it themselves.

Line type: The types of lines connecting two points can be selected as straight lines, stairs, or no lines.

Point type: There are three styles to choose from: dot, cross, and none.

Timeline configuration: Click on the timeline node in the chart view to configure the timeline settings.

Display mode: The way the timeline (X-axis) changes during the process of monitoring data:

Auto: Timeline auto scaling

Fixed length: Display a constant length.

Fixed: Display a fixed range of values from minimum to maximum.

Minimum value: The initial value of the segment. Valid when the display mode is "fixed".

Maximum value: The end value of the segment. Valid when the display mode is "fixed".

Length: Constant segment length.

Grid: Grid lines in the timeline direction diagram.

Y-axis configuration

Click on the Y-axis node in the chart view to configure settings for the Y-axis.

Display mode: The way the Y-axis changes during the process of monitoring data:

Auto: The timeline automatically scales.

Fixed: Fixed display of a range of values from minimum to maximum

Minimum value: The initial value of the segment. Valid when the display mode is "fixed".

Maximum value: The end value of the segment. Valid when the display mode is "fixed".

Grid: Grid lines in the Y-axis direction of the graph.

Description: The Y-axis is marked as "Description".

Preview of tracking chart: You can preview the pre configured chart and adjust the color of the chart.

Tracking View Right Click Menu

Configuration: The configuration checkbox can be opened, and if it is currently in monitoring mode, it will automatically pause

Download configuration: Download the configured variables to the PLC. If there are errors in the data, an error will be reported. Otherwise, the data will be automatically read. If a variable has been modified, the data must be downloaded again before it can be read, such as adding or deleting variables, modifying variable addresses, display formats, etc.
Attention: Before downloading, make sure that the software is connected to the PLC and the "CTH300/200 Local (TCP/IP)" communication interface is selected, otherwise it cannot connect to the PLC.

Start/Pause Reading: After downloading data, you can start/pause reading.

Cursor: The cursor is a tracking cursor, parallel to the vertical black line in the y-axis. It can be used to identify the time and corresponding y value of the cursor position. If the time point where the cursor is located does not have a corresponding Y value, the value of the closest point to the cursor will be displayed.

Add a cursor to the tracking map when no tracking cursor is available.

When a tracking cursor is available, add another cursor to the tracking map.

Delete the displayed tracking cursor when two are available.

No trace cursor tracking map: In this mode, you can run the trajectory map through the mouse pointer. The X value focused on the cursor will be displayed in the status bar in a regular style (e.g. time: 1m23s456ms Value: 1).

A tracking map of a cursor: In the status bar and y value, the output result is mainly marked by the time of tracking the cursor. Example: Time: 1m23s456ms.

Tracking diagram of two tracking cursor: In the status bar, output twice by the two tracking cursor (e.g. marked time interval, Time: 1m23s456ms - Time: 1m24s456ms (Δ 1s)).

User input for tracking chart: If one or two tracking cursor are available, you can move along the X-axis.

Mouse: Drag the triangle that tracks the cursor to another position. Synchronize the update of y value in the status bar.

Keyboard: The arrow keys can move the black tracking cursor left and right.

Show/Hide Variables: If you do not want to view a variable, you can select it in the variable list on the right and right-click to pop up the menu "Hide Variables". The variable will not be displayed in the chart. If a variable has been hidden and you want to view its curve again, you can right-click on the variable in the variable list on the right and select "Show Variable" from the pop-up menu, and the variable will be displayed again on the chart.

Reset View: Based on the display mode configured for the timeline and Y-axis, restore the view to its default refresh state.

Multi channel: Configure a chart for each variable in the tracking view using the first chart as a template and reinitialize the chart.

Note: The old chart configuration data will be deleted after using the "multi-channel" function.

Single channel: Track the view to place all variables in the first chart and reinitialize the chart.

Note: The old chart configuration data will be deleted after using the "single channel" function.

Save trajectory: Save the read data to a *. ctrtrace file.

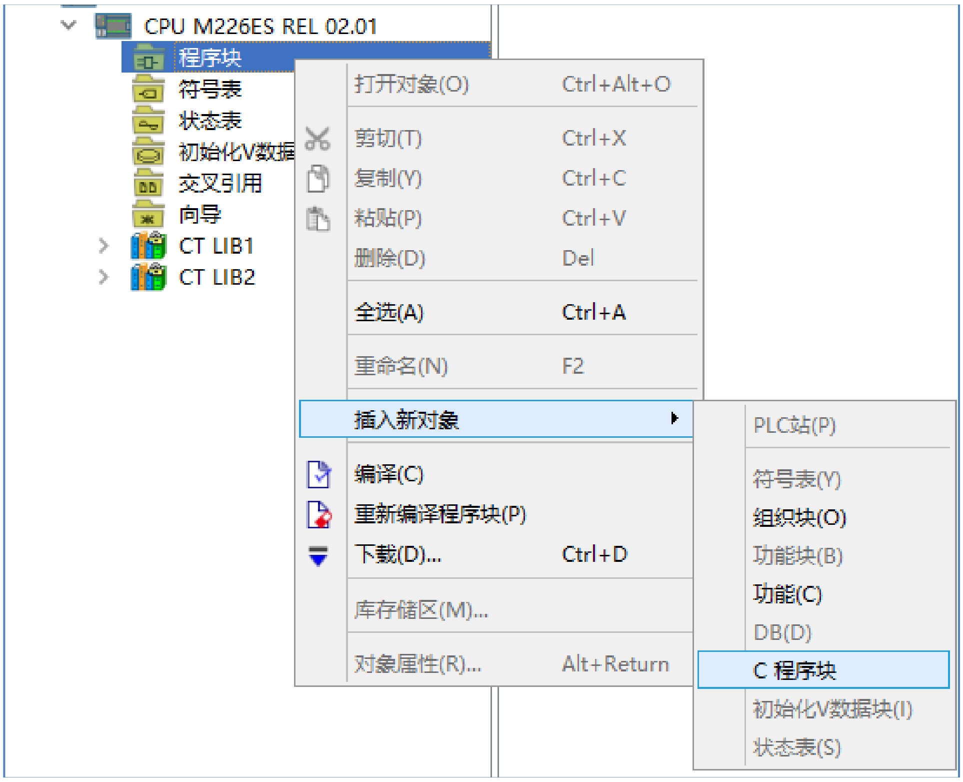
Loading trajectory: Read the loaded and saved data from the *. ctrtrace file.

7.5 C language functionality

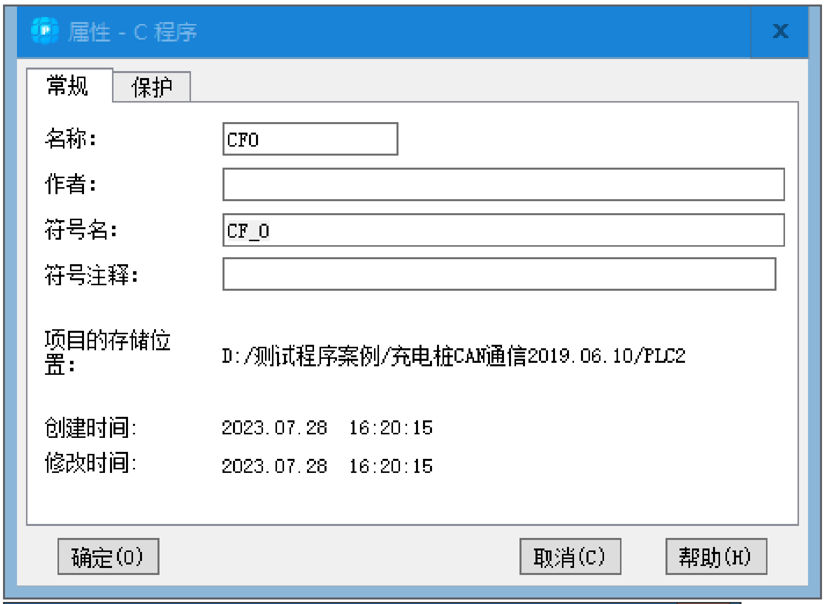
The C programming language is an upgraded and strengthened feature of MagicWorks PLC software. On the basis of the original ladder diagram and STL programming languages, C function programming is allowed to work together with programming languages such as ladder diagrams. Using C language to write functional blocks enhances the readability and maintainability of programs. At the same time, the rich computational functions in C language can realize various calling functions, saving internal space and improving programming efficiency.

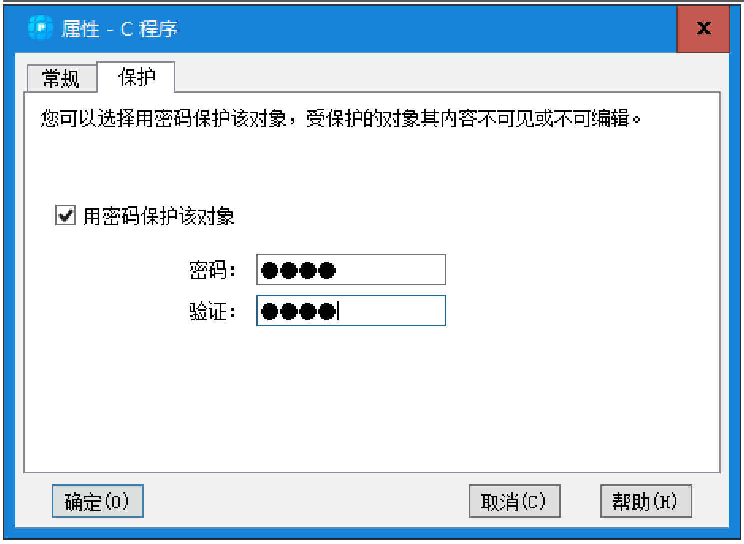
7.5.1 C programming steps in Magicworks PLC

1. Create a new project in MagicWorks PLC, select "Program Block", right-click on the blank space, and choose "Insert New Object" "C Program Block".

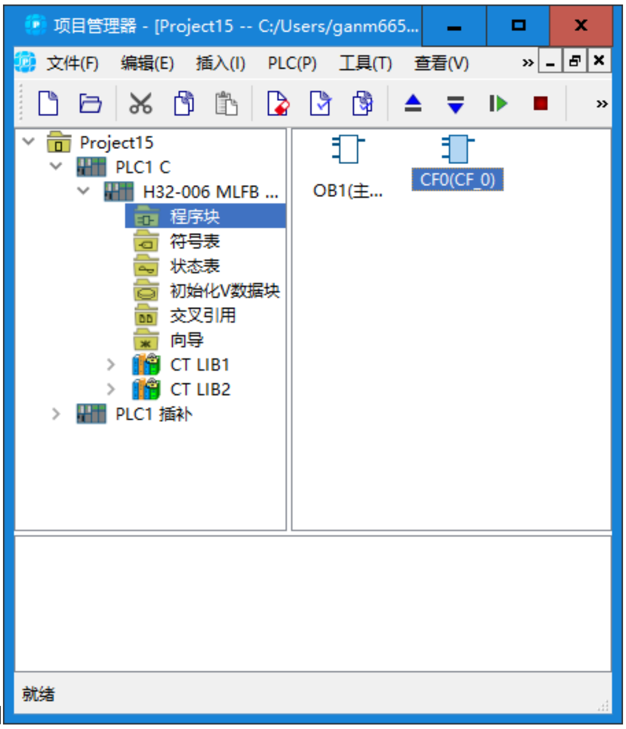


2. Edit the properties of the C program, and the symbol names defined here will automatically be used as C function names. It is recommended not to define them in Chinese. When copying code, the symbol names of the CF block must be the same as the C language function names, otherwise they will not be executed. If engineers wish to implement code protection, they can also refer to the following steps to add password protection.





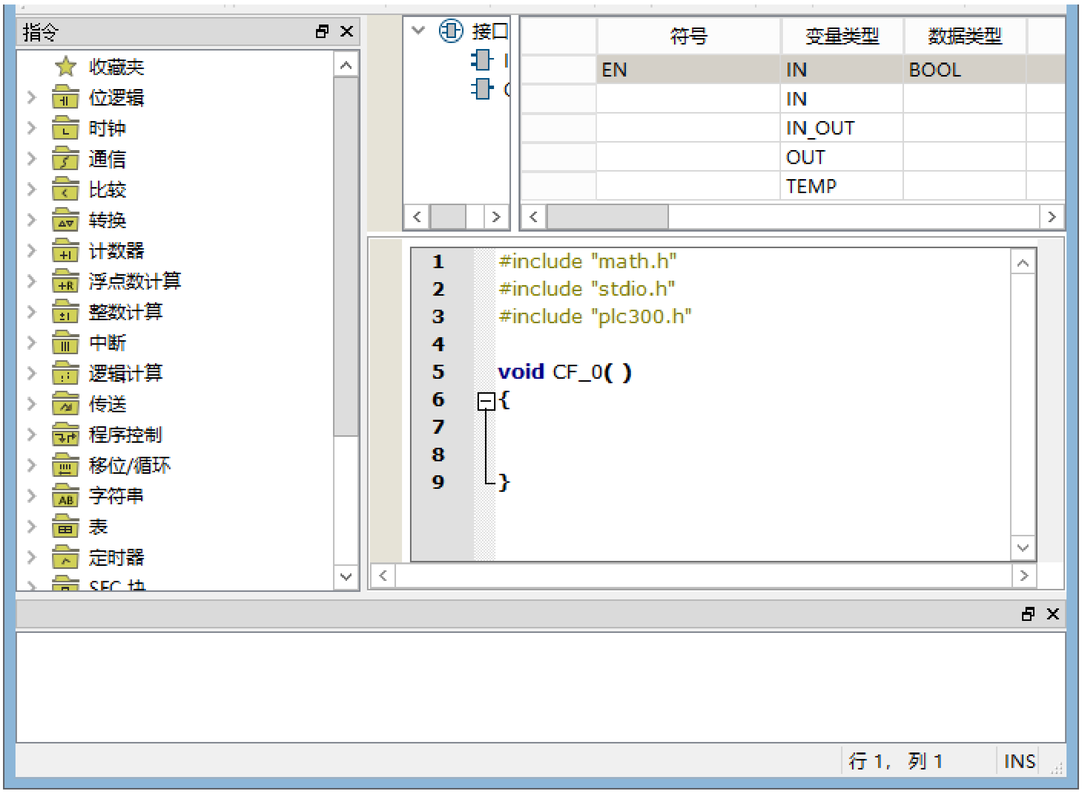
3. After the function is successfully created, CF0 can be seen in the program block.



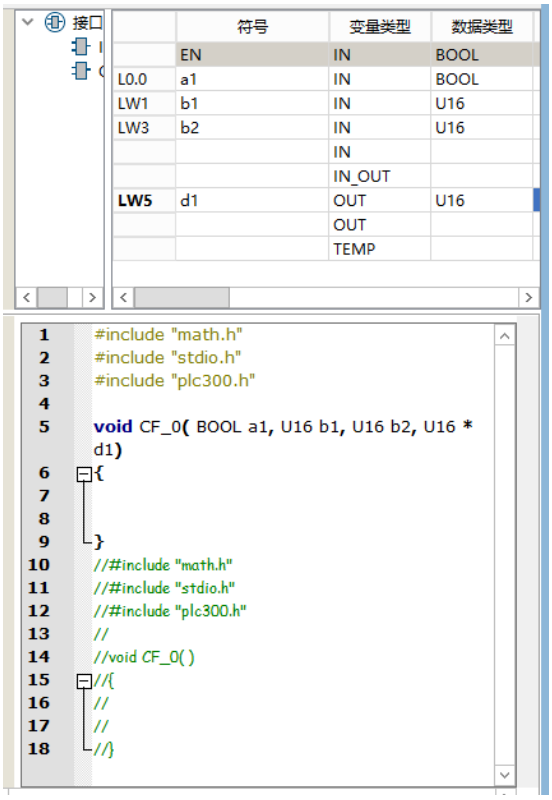
png" \ * MERGEFORM

4. Double click CF0 to enter the C function program programming window. After creating a CF0 in

MagicWorks PLC, the following three header files will be added by default.



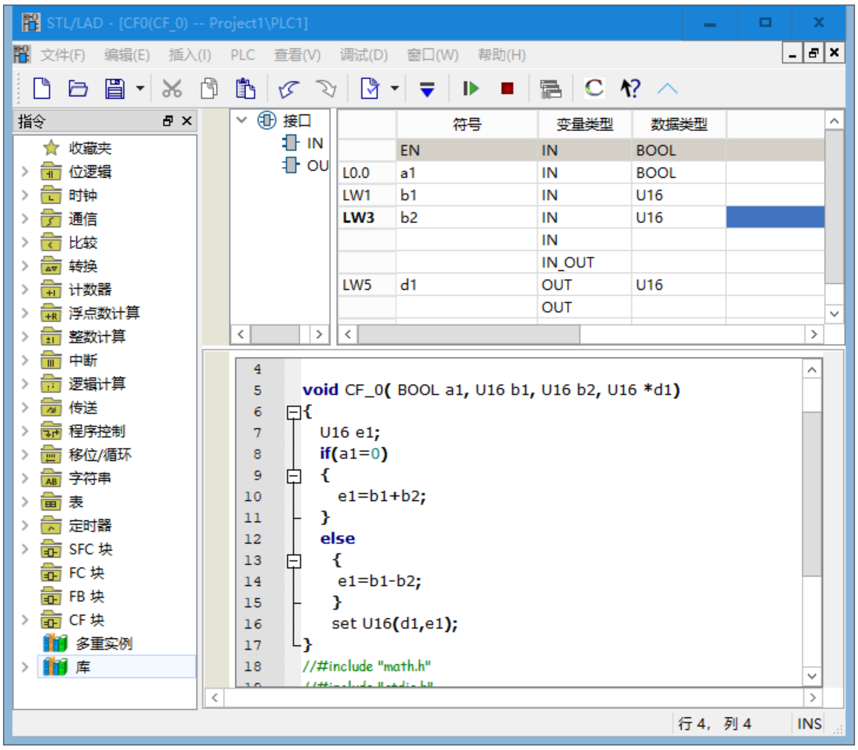
5. Define the parameter values that require external input or output in the local variable window, click "" to initialize the C program block after all definitions are complete, and regenerate the function template. Do not change the name and parameters of the C function casually.



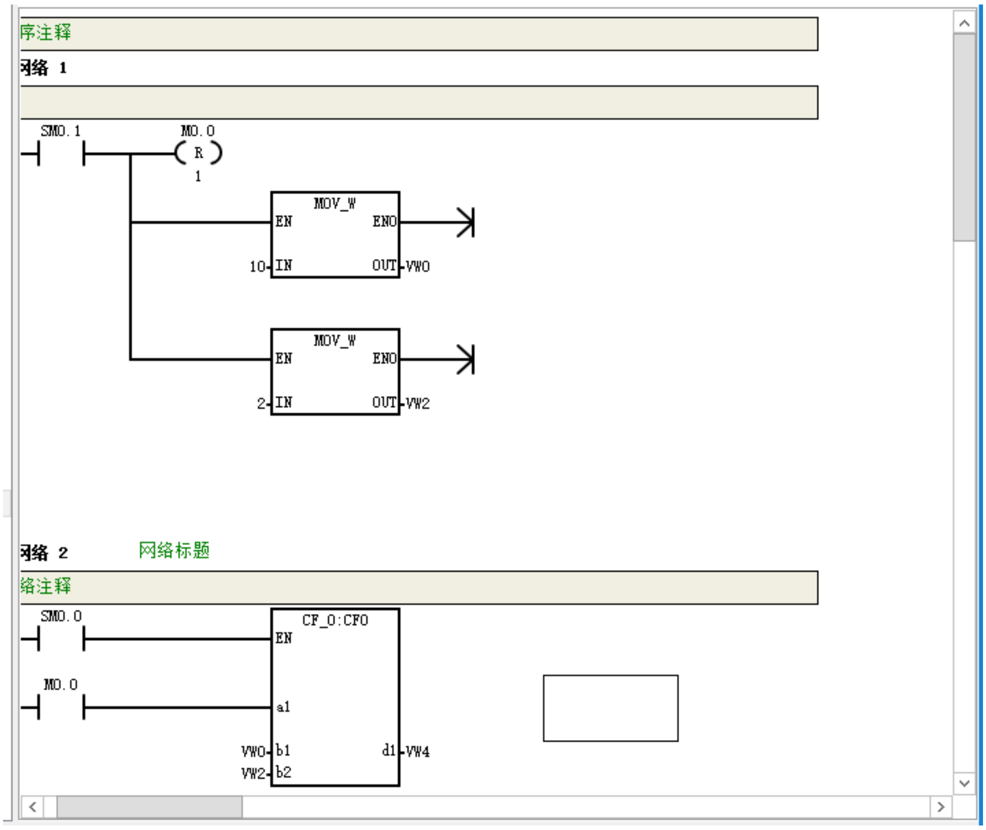
<Note>The local variable window can only define a maximum of 16 variables.

6. Write a program in the generated template, and after programming is completed, click to execute the programming operation. If there is no error prompt in the output window, it can be called by OB1 or other FCs. The parameters defined from the parameter symbol table will transform into input parameters of the corresponding type in C language, which can be directly used. And the input/output and output

parameters will become pointers of corresponding types, and the results will be written back to the STL system through pointers in the C function, that is, the C function has no return value, and the return value can only be returned through pointers.



7. The completed C function subprogram needs to be called in the OB1 main program or other executed FC. When the program block is downloaded, the C subroutine will also be downloaded to the PLC. (Note: The C function cannot be monitored online internally, but it can monitor input and output in ladder diagram tuning.). The function executed by the following program is: if M0.0=0, then vw4=vw0+vw2. If M0.0=1, then vw4=vw0-vw2.



7.5.2 Fundamentals of C Language

Head file

The compiler provides many standard functions declared in header files. Header files, also known as containing files, are carrier files that contain functional functions and data interface declarations. They are mainly used to store program declarations.

<stdio.h>	Input output function.
<stdarg.h>	Macros that support functions with variable number of variables.
<math.h>	Mathematical floating-point functions.
<stdlib.h>	Memory allocation function.
<string.h>	String processing function.
<stdbool.h>	Boolean type and boolean values true and false.
<complex.h>	Support for plural numbers.
<ctype.h>	Character classification function.
<wctype.h>	Wide character conversion function

The header file contains declarations for various functions and variables. MagicWorks PLC will add the following three header files by default when creating a new C program block:

```
#include "math.h"

#include "stdio.h"

#include "plc300.h"
```

Among them, the header file<stdio. h>contains a large number of declarations for advanced input/output (I/O) functions and is a standard input/output header file in C language. STDio stands for stand input/output, and if there are input/output operations in the program, this header file must be used.

The header file<math. h>declares some commonly used mathematical operations, such as power, square root, trigonometric functions, logarithms, etc., which can be directly called.

Plc300. h is the header file of the library libplc300. a developed by COTRUST. The library libplc300. a mainly enables users to call underlying functions and interact with STL PLC series. The functions that users can use can be viewed in the header file plc300. h, which is located in the programming software installation directory \ ccomplile \ include \ plc300. h.

variable

1. Define variables

The general form of variable definition: data type variable name

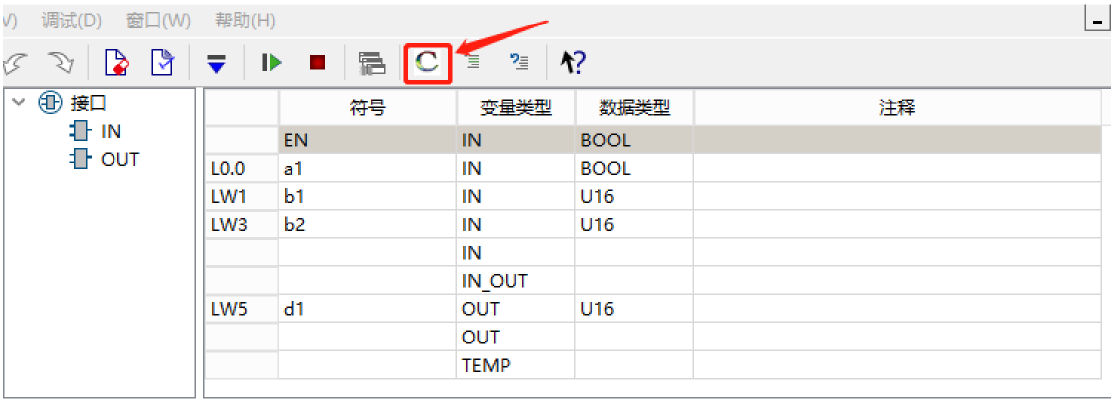
For example:

```
FP32   real1.                  //Define 32-bit floating-point numbers

S16    int1.                   //Define a 16 bit signed integer

const FP32 PI=3.1415926.       //Define 32-bit floating-point constant PI
```

The parameter values defined by the local variable window will be automatically defined into the subroutine by executing the "Initialize C Program Block".



	符号	变量类型	数据类型	注释
	EN	IN	BOOL	
L0.0	a1	IN	BOOL	
LW1	b1	IN	U16	
LW3	b2	IN	U16	
		IN		
		IN_OUT		
LW5	d1	OUT	U16	
		OUT		
		TEMP		

```
#include "math.h"
#include "stdio.h"
#include "plc300.h"

void CF_0( BOOL a1, U16 b1, U16 b2, U16 *d1)
```

be careful:

- Variable names must start with a letter or underscore, and can only consist of letters, numbers, and the underscore '_' in the middle of the name. The last character can be a type descriptor.
- Variables must be defined before use.
- A variable name can only be defined once, and it is recommended to define variables at the beginning of the program. Chinese variables are not allowed in variable definitions, and case sensitivity is strictly enforced.
- Some simpler character variable names are easily confused with some existing counters, such as c representing a counter and c1 not allowed to be used.

2. Static local variables and regular variables

The difference between local variables and ordinary variables

1) Memory allocation and release

- Ordinary local variables allocate space only when executing statements defined by the variable.
- Static local variables are in the compilation stage (before the function is executed), and the space of the variable has already been allocated.
- If a regular local variable leaves its scope {} and automatically releases its space, it cannot be used.
- Static local variables are automatically released only at the end of the entire program.

2) Initialization

- Ordinary local variables are not initialized and are random values.
- Static local variable is not initialized and is set to 0.
- The static local variable initialization statement is only valid on the first execution, but can be assigned multiple times.
- Static local variables can only be initialized with their constants.

Allocate memory space for variables in the global data area. The compiler automatically initializes its scope to a local scope, and when the function defining it ends, its scope also ends.

An example of C programming is as follows: as long as var1 is not equal to 0, var2 is incremented by one.

```
void CF_3( S16 var1, S16 *var2 )
{
 static S16 temp.
 if(var1!=0)
 {
     temp=temp+1.
  }
 setS16(var2, temp).
}//Use static modifier for temp variable
```

The use of annotations

Using comments when writing C language helps to understand the code. Comments are not compiled and will not affect the execution of the program. There are two annotation methods in C language:

- Block comments starting with/* and ending with */
- A single line comment starting with//and ending with a line break

Magicworks PLC can be manually annotated or annotated using the annotation key. Click the comment button in the menu bar to comment, and click the cancel comment button in the menu bar to cancel the comment.

```
void CF_3( S16 i, S16 *j )//令i=3
{
    S16 temp;
    temp=++i;        //i,temp的值为4
    temp=i++;        //temp的值为4，然后i的值变为5
    temp=-(i++);     //temp的值为-5，然后i的值变为6
    setS16(j,temp);

/* 这段代码为测试代码
    演示注释的方法
         —2020-02-16*/
}
```

Mixed operations between various numerical data types

When performing operations, different types of data need to be converted to the same type of data before the operation can be performed.

Calculation process: First automatically convert to data of the same type, and then perform calculations.

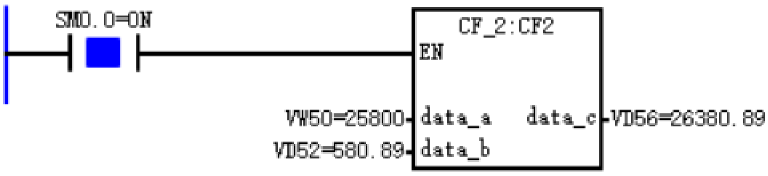
Conversion rule: Convert low byte types to high byte types.

C programming routine: c=S16 a+FP32 b, a will automatically convert to the data type of FP32, and then add it to b to obtain an FP32 type c.

The programming in C language is as follows:

```
#include "math.h"
#include "stdio.h"
#include "plc300.h"
void CF_2( S16 data_a, FP32 data_b, FP32 *data_c )
{
 FP32 temp.
 temp=data_a+data_b.
 setFP32(data_c,temp).
}
```

The CF block obtained after compilation can be called through OB1 or other FC calls:



Force type conversion

To convert the value of a specified expression to a specified type, the programmer needs to specify the type to be converted.

Form: (Type Name) (Expression)

For example: (S16) (a+b)

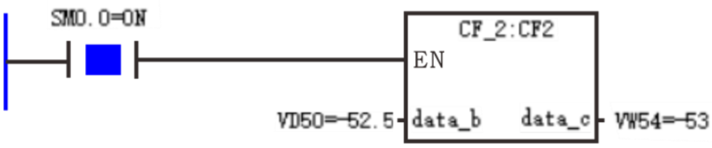
Explanation: 1) Type names and expressions need to be enclosed in parentheses

2) During forced type conversion, an intermediate variable of the desired type is obtained, and the original variable type remains unchanged.

C programming routine: Round a 32-bit floating-point number to the nearest integer and convert it to a 16 bit integer.

```
void CF_2(FP32 data_b, S16 *data_c )
{
  S16 temp.
  If (data_b>0)
  {
        temp=(S16)(data_b+0.5).
  }
  else
  {
    temp=(S16)(data_b-0.5).
  }
  setS16(data_c,temp).
}
```

The CF block obtained after compilation can be called through OB1 or other FC calls:



array

An array is a collection of ordered data. If a set of finite variables of the same type is named, the name is an array name. The data type of each element in the array is the same, and users determine the elements in the array by the array name and index.

Definition of one-dimensional array: type descriptor array name [integer expression]

For example: FP32 real0 [10].

Explanation: An array named real0 is defined with 10 elements, each of which is of type FP32. These 10 elements are real [0], real [1], real [2],... real [8], real [9].

Initialize array: The array can be initialized in the declaration. When initializing the array, the values of each index variable should be enclosed in {} and separated by commas, for example: S16 aa [3]={2,6,1}. Equivalent to: S16 aa [3]. aa[0]=2. aa[1]=6. aa[2]=1.

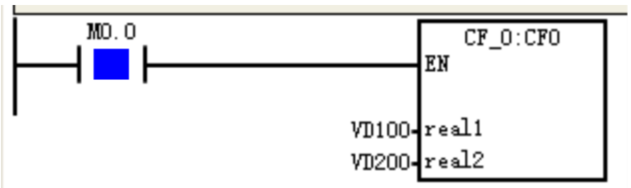
be careful:

- The index starts from 0. For example, the first element of the array is real1 [0], and the eighth element is real1 [7].
- C language generally does not allow dynamic definition of array size, such as FP32 real1 [n], which is not allowed.
- The language does not check for array index out of bounds, so do not use them when using them. In this example, the number of array elements is 10, and the index range is 0-9, which cannot exceed 9.
- The storage of elements in an array has a sequential order, and they are stored together in memory in this order.

C programming example: Start VD100 with 10 consecutive values and copy them to VD200.

```
void CF_0( void* real1, void* real2 )
{
  S16 i, j.
  FP32 temp[10] ={0,0,0,0,0,0,0,0,0,0}.   //  Define array
  for (i=0.i<10.  i++)
   {
     temp[i]=getFP32(real1+4*i).       //  Array address
   }
  for (j=0.j<10.  j++)
   {
     set FP32((real2+4*j),temp[j]).
   }
}
```

Call the generated CF0



The results of the status table are as follows:

	地址	格式	当前值
1	VD100	浮点数	0.1111
2	VD104	浮点数	0.2222
3	VD108	浮点数	0.3333
4	VD112	浮点数	0.4444
5	VD116	浮点数	0.5555
6	VD120	浮点数	0.6666
7	VD124	浮点数	0.7777
8	VD128	浮点数	0.8888
9	VD132	浮点数	0.9999
10	VD136	浮点数	1.0
11		有符号	
12	VD200	浮点数	0.1111
13	VD204	浮点数	0.2222
14	VD208	浮点数	0.3333
15	VD212	浮点数	0.4444
16	VD216	浮点数	0.5555
17	VD220	浮点数	0.6666
18	VD224	浮点数	0.7777
19	VD228	浮点数	0.8888
20	VD232	浮点数	0.9999
21	VD236	浮点数	1.0

7.5.3 Operators and Expressions

Assignment operators and assignment expressions

1. Assignment operator=

Assignment expression: Assign a value to a variable or assign the value of a variable to another variable.

Assignment form: Variable name=constant or expression.

Assignment function: Assign the value of a constant or expression on the right to a variable on the left. For example: a=3, a=b * c

be careful:

- All statements, including assignment statements, must end with a semicolon '.'.
- After assignment, the original value on the left is replaced by the value of the expression on the right
- If the value type of the expression on the right is inconsistent with the type of the variable on the left, the type of the variable on the left shall prevail, and the value type of the expression on the right shall be automatically converted to the type of the variable on the left.

2. Multiple assignment operator: Multiple assignment is to assign the value of an expression to several variables simultaneously. In this case, the names of the assigned variables should be written to the left of the assignment number in order.

Assignment form: Variable 1=Variable 2=...=Variable n=Expression

Equivalent to: Variable 1=(Variable 2=(...=(variable n=expression)))

For example: a=b=c=7.

3. Composite assignment operator: Add other operators before the assignment symbol '='. Common ones include:+=, -=, *=,/=,%=

The function of composite assignment operation: perform composite operation with the left variable and then assign it to the left variable.

For example:

a+=3.	Equivalent to a=a+3
x*=y+8.	Equivalent to x=x * (y+8)
x%=3.	Equivalent to x=x% 3
int a=1,b=2,c=2. c*=a-b.	Equivalent to c=c * (a-b)=2 * (-1)=-2

Arithmetic operator

Arithmetic operators in C language:

+ Add or take positive value operator
- Subtract or take negative value operator
* multiply operator
/ Dividing operator
% Remainder operator

Among them, addition, subtraction, multiplication, and division are all remainder operators that require two operands.

Self increasing and decreasing operator

Operator description:++represents auto increment, and the variable value is incremented by 1,

--Represents self reduction, where the variable value is reduced by 1.

Usage 1: Used as a standalone expression. For example, i++or++i is equivalent to i=i+1

Usage 2: Appears in other expressions to participate in operations,++i/-- i represents i increasing/decreasing by 1 before participating in other operations. And i+/i -- means that after i participates in the operation, the value of i increases/decreases by 1.

For example: S16 j, i=3.

```
j=++i.    // i. The value of j is 4
j=i++.    // The value of j is 4, and then the value of i becomes 5
j=-(i++). // The value of j is -5, and then the value of i becomes 6
```

Relational operators and expressions

Relational operation: Comparing two operands to determine if they satisfy a given relationship, used for simple conditional judgments. In C language, there are the following relational operators:

(1)<less than

(2)<=less than or equal to

(3

)>greater than

(4)>=greater than or equal to

(5)==equal to (6)= The

result of the non equal relationship operator can only be 0 or 1. When the value is true, the result is 1, and when the value is false, the result is 0.

For example: 3.24<=2.98// The value is false, and the result of the operation is 0

0.5! = 3+1. // The value is true, and the result of the operation is 1

Relational expression: An expression that connects two expressions (arithmetic expression, relational expression, logical expression, assignment expression, character expression) using relational operators is called a relational expression

The general form of relational expression is: expression, relational operator, expression

The priority of relational operators is lower than arithmetic operators and higher than assignment operators.

For example: a=1+2>3-1// Equivalent to a=((1+2)>(3-1)), the result is a=1

Logical Operators and Expressions

Table7-8 Logical operations, also known as Boolean operations, are used for complex conditional judgments.

operator	meaning	priority	ST description
!	NOT	1	NOT
&&	AND	2	AND
ll	OR	3	OR

The general form of logical expression: expression, logical operator, expression

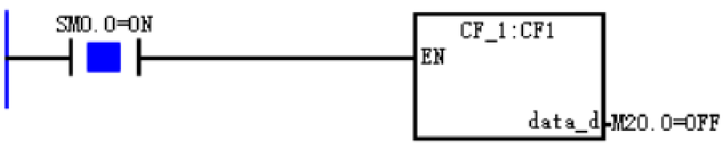
Table7-9 Logical operation truth table

A value	B value	! A	A&&B	AllB
Not 0	Not 0	0	1	1
Not 0	0	0	0	1
0	Not 0	1	0	1
0	0	0	0	0

C programming routine: When the result of (a>b)&&(b>c) is true, d=1. when the result of (a>b)&&(b>c) is false, d=0.

```
void CF_1( BOOL *data_d )
{
 S16 data_a=3,data_b=2,data_c=1.
 if ((data_a>data_b)&&(data_b>data_c))
 {
  *data_d=1.
 }
 else
 {
  *data_d=0.
 }
}
```

The CF block obtained after compilation can be called through OB1 or other FC calls:



Conditional operators and conditional expressions

The conditional operator and conditional expression have two symbols: question mark and semicolon, which form a ternary operation with three operation arrays.

General form:<Expression 1>< Expression 2>:<Expression 3>

Priority: Logic>Condition>Assignment

For example: S16 max, a=5. b=3.

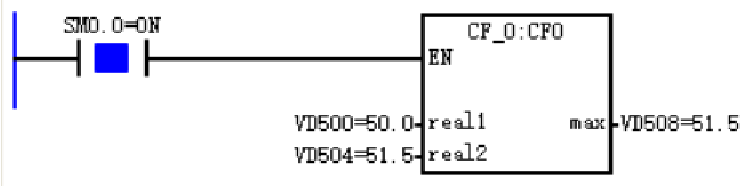
```
max=a>b? a:b
```

Result: max=5

C programming routine: Compare the sizes of two values and output the maximum value.

```
void CF_0( FP32 real1, FP32 real2, FP32 *max )
{
 FP32 max_t.
 max_t=real1>real2? real1:real2.
 set FP32(max,max_t).
}
```

The CF block obtained after compilation can be called through OB1 or other FC calls:



For Loop Statement

In an application, certain steps need to be executed multiple times, and this process is called a loop. For loop is a type of loop statement in programming languages, and the design of a loop program requires the establishment of a program that can loop back and execute its own program.

The general form of a for loop statement:

```
For (single expression. conditional expression. end loop body)
{
Intermediate circulation system.
}
```

For example: for (i=1. i<=100. i++)

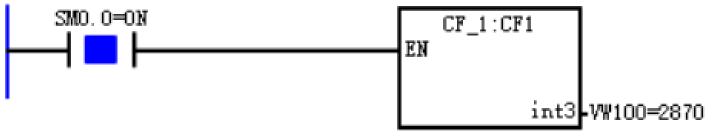
```
{
sum=i+sum.
}
```

The statements in the for loop are repeatedly executed, and after each loop, the variable i is automatically incremented by 1. I=1 and i<=100 are the starting and ending values of the loop control variable. When the control variable reaches its termination value, the program executes the next instruction after the for statement.

For example, find the sum of squares between 1-20!

```
void CF_1( S16 *int3 )
{
  S16 int3_t=0.
  S16 i.
  for (i=1.i<=20.i++)
  {
    int3_t=int3_t+i*i.
  }
  setS16(int3,int3_t).
}
```

The CF block obtained after compilation can be called through OB1 or other FC calls:



7.5.4 Select statement

If Select Structure Statement

If selection structure: a selection structure that judges the if condition and executes subsequent operations based on the judgment result.

1) The general format for selecting a single branch if structure is:

If (expression)//Expression: Refers to the judgment condition, true is 1, false is 0

{

Execute statements// Execute the statement here only when the expression is true

}

For example, input a number, determine whether it can be divided by 4 or 7, and output 1 if the condition is met.

```
void CF_0( S16 int1, S16 *out1 )
{
  S16 out_t=0.
  if (int1%4==0||int1%7==0)
   {
    out_t=1.
  }
  setS16(out1,out_t).
}
```

The CF block obtained after compilation can be called through OB1 or other FC calls:

2) The general format for selecting a double branch if structure is:

If (expression)//Expression: Refers to the judgment condition, true is 1, false is 0

{

Execute statements// Execute the statement here only when the expression is true

}

else

{

Execute statements// Condition is 0, execute the statement after else

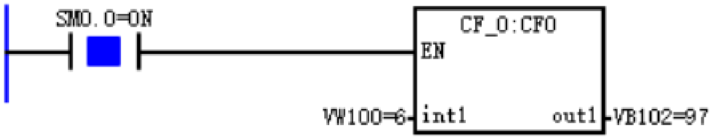
}

For example: Input a number, even output a, odd output b.

```
void CF_0(S16 int1, U8 *out1 )
{
  char out.
  if (int1%2==0)
```

```
  {
    out=a.
  }
  else
  {
    out=b.
  }
  setU8(out1,out).
}
```

The CF block obtained after compilation can be called through OB1 or other FC calls:

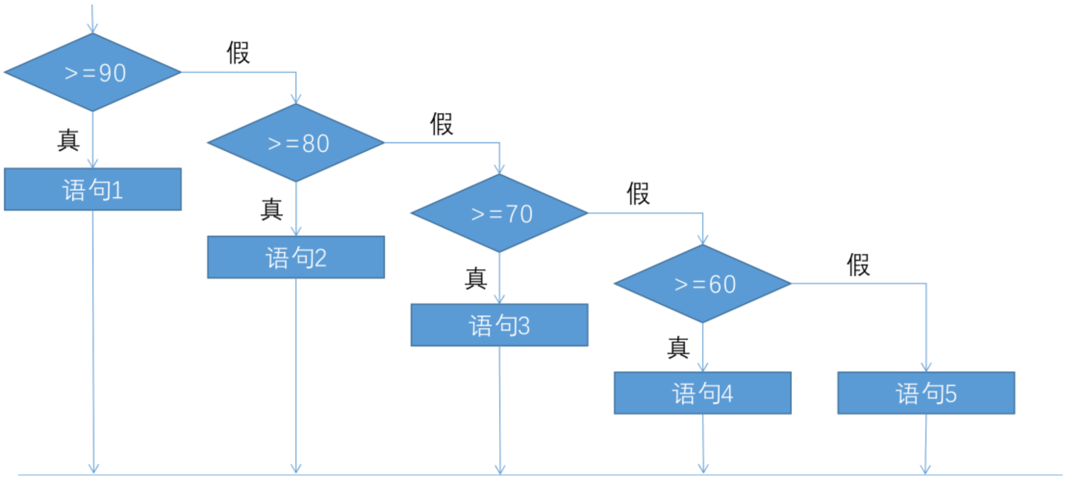


3) The general format for multiple if selection structures:

```
If (expression)
{
Execute statements.
}
  Else if (expression 1)
  {
Execute statements.
  }
    Else if (expression 2)
    {
Execute statements.
    }
……
else
{
Execute statements.
}//The last else {} can be omitted, if it exists, it must be placed after the else if
```

For example, when entering grades, A is greater than or equal to 90, B is between 80-89, C is between 70-79, D is between 60 and 69, and E is below 60.

The flowchart of multiple if execution is as follows:



The programming in C language is as follows:

```
void CF_1( S16 score, U8 *level )
{
  char level_t.
  if (score>=90)
  {
    level_t=A.
  }
  else if (score>=80)
  {
    level_t=B.
  }
  else if(score>=70)
  {
    level_t=C.
  }
  else if(score>=60)
  {
    level_t=D.
  }
  else
  {
    level_t=E.
  }
  setU8(level,level_t).
}
```

Switch multi branch selection statement

The switch statement is a multi branch conditional statement that causes the program to select a branch from multiple branches for execution based on the value of the expression.

The general format of switch:

```
Switch (expression)
{
Case constant expression 1: statement 1.
        break.
Case constant expression 2: statement 2.
        break.
  ……
Case constant expression n: statement n.
        break.
Default: statement n+1.
}
```

be careful:

- After executing the case condition statement, a break must be added to exit the switch structure, otherwise branching is not possible.
- The values of constant expressions must not be equal to each other, otherwise there is duality.

- The order of cases is irrelevant, default is optional.

- The switch statement is not equivalent to the if statement. The switch statement can only perform value equality checks, not logical judgments. The if statement can not only determine the equality of values, but also calculate relational or logical expressions to make logical judgments on their authenticity.

For example:

```
void CF_0( S16 Level,  BOOL *heat1, BOOL *heat2, BOOL *heat3, BOOL *fan )
{
 *heat1=*heat2=*heat3=*fan=0.
 switch (Level)
 {
  case 1:*fan=1.
       break.
  case 2:*fan=0.
       break.
  case 3:*heat1=1.
       break.
  case 4:*heat1=*heat2=1.
       break.
  case 5:*heat1=*heat2=*heat3=1.
       break.
  default:*heat1=*heat2=*heat3=1.
 }
}
```

7.5.5 Introduction to libplc300. a Library

The libplc300. a library is a built-in library of Magicworks PLC, which contains functions for taking values from STL and writing values back to STL, as well as data conversion operation functions between STL system and C language system. Users can directly call these functions. The functions that users can use can be viewed in the header file plc300. h, which is located in the programming software installation directory \ precompile \ include \ plc300. h.

The following will introduce the data types and various operation functions of C program blocks.

The selectable data types in the CF block parameter symbol table

type	explain
typedef unsigned char **BOOL**.	8-bit unsigned
typedef unsigned char **U8**.	8-bit unsigned
typedef signed char **S8**.	8-bit signed
typedef unsigned short **U16**.	16 bit unsigned
typedef signed short **S16**.	16 bit signed
typedef unsigned int **U32**.	32-bit unsigned
typedef signed int **S32**.	32-bit signed

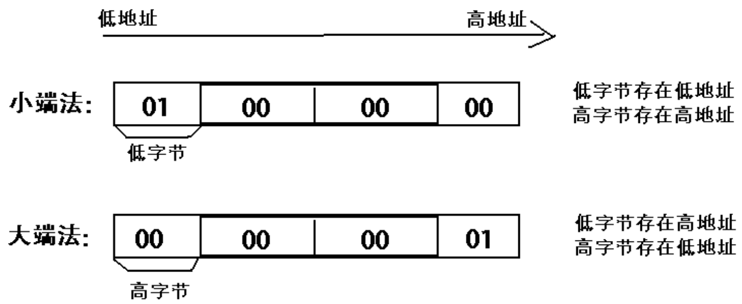
typedef float **FP32**.	Single precision floating point
typedef double **FP64**.	Double precision floating point
typedef long long **S64**.	64 bit unsigned
typedef unsigned long long **U64**.	64 bit signed
typedef unsigned char **BYTE**.	8-bit unsigned
typedef unsigned short **WORD**.	16 bit unsigned
typedef signed short **INT**.	16 bit signed
typedef unsigned int **DWORD**.	32-bit unsigned
typedef signed int **DINT**.	32-bit unsigned
typedef float **REAL**.	Single precision floating point

Size conversion

Due to the fact that the data in the STL system uses the big end mode, while C language uses the small end mode, directly operating pointer assignment may cause byte order confusion. Therefore, we provide a set of functions that take values from STL through pointers and write values back to STL. The specific form of these functions can be found in the MagicWorks PLC installation directory \ precompile \ include \ plc300. h.

Big end mode: The high bytes of data are stored in the low addresses of memory, while the low bytes of data are stored in the high addresses of memory.

Small end mode: refers to the high byte of data being stored in the high address of memory, while the low byte of data is stored in the low address of memory.



When parameters are passed through pointers in the C language system and STL system, because the C language system uses the small end mode, while the STL system uses the large end mode, it is necessary to call the size conversion interface.

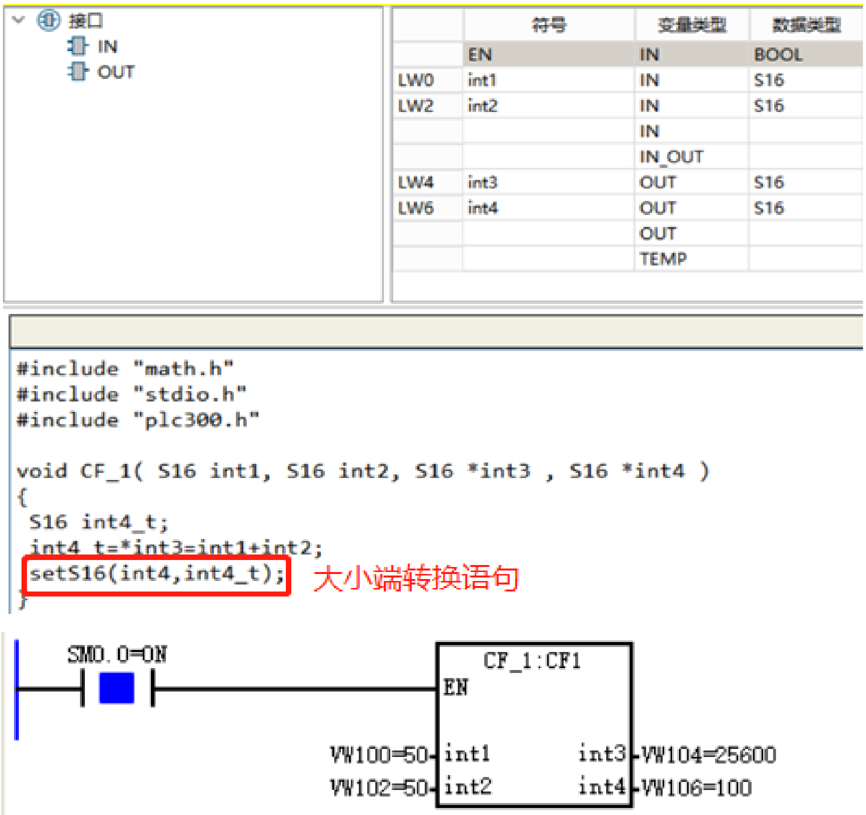
The data conversion operation functions between STL system and C language system

1. Writing values back to STL environment through pointers: Due to the fact that input/output and output parameters will become pointer types of corresponding types, and the issue of size conversion, all input/output or output parameters must be written back to STL environment from C language through the following instructions.

function	describe	input	return
void setBOOL(BOOL *p, BOOL d)	Write a BOOL data from C language back to STL system	p: Pointer transmitted from STL system	--

		d: BOOL data used in C language	
void setU8(U8 *p, U8 d)	Write U8 data from C language back to STL system	p: Pointer transmitted from STL system d: U8 data used in C language	--
void setS8(S8 *p, S8 d)	Write an S8 data from C language back to STL system	p: Pointer transmitted from STL system d: S8 data used in C language	--
void setU16(U16 *p, U16 d)	Write U16 data from C language back to STL system	p: Pointer transmitted from STL system d: U16 data used in C language	--
void setS16(S16 *p, S16 d)	Write a S16 data from C language back to STL system	p: Pointer transmitted from STL system d: S16 data used in C language	--
void setU32(U32 *p, U32 d)	Write U32 data from C language back to STL system	p: Pointer transmitted from STL system d: U32 data used in C language	--
void setS32(S32 *p, S32 d)	Write a S32 data from C language back to STL system	p: Pointer transmitted from STL system d: S32 data used in C language	--
void setFP32(FP32 *p, FP32 d)	Write FP32 data from C language back to STL system	p: Pointer transmitted from STL system d: FP32 data used in C language	--

The example is as follows: Input int1 and int2 are added, output int4 is size converted, and the output result is correct. The output int3 did not undergo size conversion, resulting in incorrect output results.



2. Taking values from STL through pointers: If the PLC's V, M, and other registers are directly called in the input parameters of a C function, the pointer of the PLC register can be input to the C function, and the value of the PLC memory area can be obtained to C language by taking the pointer value, but it must be converted through the following functions.

function	describe	input	return
U8 getU8(U8 *p)	Retrieve U8 data available for the C language system from the p address	p: Pointer transmitted from STL system	U8 data available for C language system
S8 getS8(S8 *p)	Retrieve S8 data available for the C language system from the p address	p: Pointer transmitted from STL system	S8 data available for C language system
U16 getU16(U16 *p)	Retrieve U16 data available for the C language system from the p address	p: Pointer transmitted from STL system	U16 data available for C language system
S16 getS16(S16 *p)	Retrieve S16 data available for the C language system from the p address	p: Pointer transmitted from STL system	S16 data available for C language system
U32 getU32(U32* p)	Retrieve U32 data available for the C language system from the p address	p: Pointer transmitted from STL system	U32 data available for C language system
S32 getS32(S32* p)	Retrieve S32 data available for the C language system from the p address	p: Pointer transmitted from STL system	S32 data available for C language system
FP32 getFP32(FP32* p)	Retrieve FP32 data available for the C language system from the p address	p: Pointer transmitted from STL system	FP32 data available for C language system

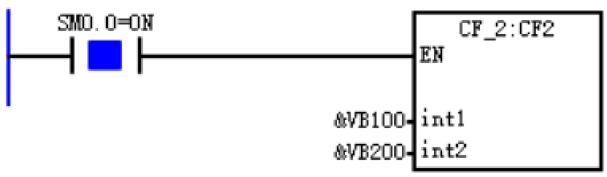
<Note>Do not use pointers to retrieve values from STL: Input variables do not use pointers, but directly obtain the values of PLC registers, so there is no need to use the get function.

The programming in C language is as follows:

1. Assign 20 consecutive VW values starting from VW100 to the VW200 address segment.

```
void CF_2( void* int1, void* int2 )
{
 S16 r,i.
 for(i=0.i<20.i++)
 {
  r=getS16(int1+2*i).
  setS16((int2+2*i),r).
 }
}
```

Call the generated CF block



The results of the status table are as follows:

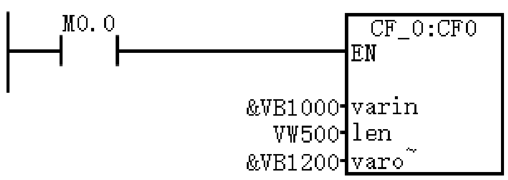
3	VW100	有符号	+1
4	VW102	有符号	+2
5	VW104	有符号	+3
6	VW106	有符号	+4
7	VW108	有符号	+5
8	VW110	有符号	+6
9	VW112	有符号	+7
10	VW114	有符号	+8
11	VW116	有符号	+9
12	VW118	有符号	+10
13	VW120	有符号	+11
14	VW122	有符号	+12
15	VW124	有符号	+13
16	VW126	有符号	+14
17	VW128	有符号	+15
18	VW130	有符号	+16
19	VW132	有符号	+17
20	VW134	有符号	+18
21	VW136	有符号	+19

1	VW200	有符号	+1
2	VW202	有符号	+2
3	VW204	有符号	+3
4	VW206	有符号	+4
5	VW208	有符号	+5
6	VW210	有符号	+6
7	VW212	有符号	+7
8	VW214	有符号	+8
9	VW216	有符号	+9
10	VW218	有符号	+10
11	VW220	有符号	+11
12	VW222	有符号	+12
13	VW224	有符号	+13
14	VW226	有符号	+14
15	VW228	有符号	+15
16	VW230	有符号	+16
17	VW232	有符号	+17
18	VW234	有符号	+18
19	VW236	有符号	+19

2. Sort the LEN INT values starting from Zone V by size and display them in another Zone V.

```
void CF_0( void* varin, S16 len, void* varout )
{
  S16 i,j,n,k,l.
  S16 temp[len].
For (l=0. l<len. l++)//Store the value of V in the temp array
  {
    temp[l]=getS16(varin+2*l).
  }
  for (i=len.i>1.  I -)//Bubbling method sorts the size values, with the maximum value "floating" on top of the array like a bubble each time
            {
    for(j=0.j<i-1.  J++)//Compare adjacent elements one by one, moving the larger element backwards
if (temp[j]<temp[j+1])
{
n=temp[j].
temp[j]=temp[j+1].
temp[j+1]=n.
}
}
For (k=0. k<len. k++)//Transfer the temp array values to Zone V
  {
    setS16((varout+2*k),temp[k]).
  }
}
```

The CF block obtained after compilation can be called through OB1 or other FC calls:



The results of the status table are as follows:

地址	格式	当前值
VW1000	有符号	+24
VW1002	有符号	+231
VW1004	有符号	+2
VW1006	有符号	+141
VW1008	有符号	+4
VW1010	有符号	+23
VW1012	有符号	+1
VW1014	有符号	+7
VW1016	有符号	+6
VW1018	有符号	+4
VW1020	有符号	+0
VW1022	有符号	+578
VW1024	有符号	+98
VW1026	有符号	+74
VW1028	有符号	+15
VW1030	有符号	+97
VW1032	有符号	+88
VW1034	有符号	+17136
VW1036	有符号	+11
VW1038	有符号	+17136

VW1200	有符号	+17136
VW1202	有符号	+17136
VW1204	有符号	+578
VW1206	有符号	+231
VW1208	有符号	+141
VW1210	有符号	+98
VW1212	有符号	+97
VW1214	有符号	+88
VW1216	有符号	+74
VW1218	有符号	+24
VW1220	有符号	+23
VW1222	有符号	+15
VW1224	有符号	+11
VW1226	有符号	+7
VW1228	有符号	+6
VW1230	有符号	+4
VW1232	有符号	+4
VW1234	有符号	+2
VW1236	有符号	+1

The operation function for obtaining the memory base address

The operation function for obtaining the memory base address

function	describe	input	return
*getLBase(void)	Get the memory address of L area	nothing	L area memory size
*getIBase(void)	Obtain the memory address of Zone I	nothing	Memory size in Zone I
*getQBase(void)	Get Q-zone memory address	nothing	Q-zone memory size
*getAQBase(void)	Obtain the memory address of AQ area	nothing	AQ area memory size
*getAIBase(void)	Obtain the memory address of the AI area	nothing	AI area memory size
*getSMBase(void)	Obtain the memory address of SM area	nothing	SM area memory size
*getTVBase(void)	Obtain the memory address of the TV area	nothing	TV area memory size
*getTBBase(void)	Obtain the memory address of TB area	nothing	TB area memory size
*getSBase(void)	Obtain the memory address of the S area	nothing	Memory size of S area
*getCVBase(void)	Obtain the memory address of the CV area	nothing	Memory size of CV area
*getCBBase(void)	Obtain the memory address of CB area	nothing	CB area memory size
*getACBase(void)	Obtain the memory address of the AC area	nothing	AC area memory size

*getVBase(void)	Obtain the memory address of Zone V	nothing	Memory size in Zone V
*getMBase(void)	Obtain the memory address of zone M	nothing	Memory size of M area

IO related functions

function	describe	input	return
U8 ReadExtDI(U16 oft, U8 *Err)	Read one byte of the extended IO without updating the memory mapping area.	Softth byte of IB memory Err: Store the returned error code 0- No error. Other - Access error	If there are no errors, it is the actual input of the bus extension IO.
U16 ReadExtAI(U16 oft, U8 *Err)	Read an analog input of the extended IO and update the memory mapping area	Oft: The softth channel 0-AIW0 of AIW memory. 2-- AIW2 odd number is invalid Err: Store the returned error code 0- No error Other - Access error	If there are no errors, it is the simulated actual input.
Void WriteExtAQ(U16 oft, U16 Data, U8 *Err)	Write an analog output for extended IO, The new content will be updated to the memory mapped area.	Oft: The softth channel of AWW memory 0--AQW0 2-AWW2 odd number is invalid Data: Value to be updated Err: Store the returned error code 0- No error Other - Access error	--
Void WriteExtDQ(U16 oft, U16 Data, U8 *Err)	Write one byte of extended IO, The new content will be updated to the memory mapped area.	Oft: The softth byte of QB memory Data: Value to be updated Err: Store the returned error code 0- No error Other - Access error	--
U8 STL_SI(U32 byteOffset, U8 bitOffset, U8 pointNum)	SI (immediate output 1) instruction in STL	ByteOffset: Byte offset BitOffset: Bit offset PointNum: Number of bits	Return error code 0: No errors Other: There are errors.
U8 STL_RI(U32 byteOffset, U8 bitOffset, U8 pointNum)	RI (immediate output 1) instruction in STL	ByteOffset: Byte offset BitOffset: Bit offset PointNum: Number of bits	Return error code 0: No errors Other: There are errors.
mc_readSetPos(U8 AxisId, FP64 *ppos)	Read the current setting position of the axis	AxisId: Axis number	Err error pointer 0- No error, ELS - Error
mc_readActPos(U8 AxisId, U16 *pErr)	Read the current actual position of the axis	AxisId: Axis number	PErr: Error pointer 0- No error Else - Error

7.6 Expansion module specifications

M226ES supports all expansion modules of the CTH200 series, including digital modules, analog modules, temperature modules, communication modules, weighing modules, operation control modules, etc. The following chapters will introduce the technical specifications of these modules.

7.6.1 Specification for Digital Expansion Module

Table7-10 CTH200 series PLC digital expansion module ordering data

Specification parameters	Order number
SM221 digital input module, 8-point input, 24V DC	CTH2 221-1BF32
SM221 digital input module, 16 point input, 24V DC	CTH2 221-1BH32
SM221 digital input module, 32 point input, 24V DC	CTH2 221-1BL32
SM222 digital output module, 8-point transistor output, 24V DC, 0.5A (output protection)	CTH2 222-1BF32
SM222 digital output module, 8-point transistor NPN output, 24V DC, 0.5A	CTH2 222-1DF32
SM222 digital output module, 16 point transistor output, 24V DC, 0.5A (output protection)	CTH2 222-1BH32
SM222 digital output module, 32 point transistor output, 24V DC, 0.5A (output protection)	CTH2 222-1BL32
SM222 digital output module, 8-point relay output, 2A	CTH2 222-1HF32
SM222 digital output module, 16 point relay output, 2A	CTH2 222-1HH32
SM223 digital input/output module, 4-point 24V DC input, 4-point transistor output, 24V DC, 0.5A (output protection)	CTH2 223-1BF32
SM223 digital input/output module, 8-point 24V DC input, 8-point transistor output, 24V DC, 0.5A (output protection)	CTH2 223-1BH32
SM223 digital input/output module, 16 point 24V DC input, 16 point transistor output, 24V DC, 0.5A (output protection)	CTH2 223-1BL32
SM223 digital input/output module, 4-point 24V DC input, 4-point relay output, 2A	CTH2 223-1HF32
SM223 digital input/output module, 8-point 24V DC input, 8-point relay output, 2A	CTH2 223-1PH32
SM223 digital input/output module, 16 point 24V DC input, 16 point relay output, 2A	CTH2 223-1PL32

Specification for Digital Input Module

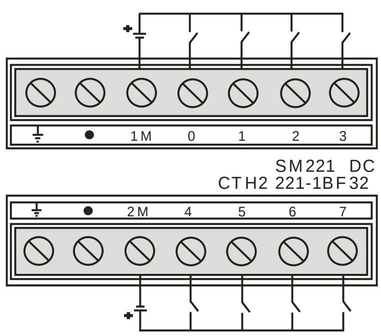
Table7-11 Specification for Digital Input Module

standard	Input at 8 o'clock, 24V DC	16 point input, 24V DC	32 point input, 24V DC
Order number	CTH2 221-1BF32	CTH2 221-1BH32	CTH2 221-1BL32
physical characteristics			
Dimensions (width x height x depth)	46 × 96 × 62mm	71.3 × 96 × 62mm	138 × 96 × 62mm
Power loss			
power loss	2W	3W	
+5V DC consumption current	57mA	79mA	179mA
Input characteristics			
Input points	eight	sixteen	thirty-two

Input type	Leakage type/source type		
input voltage			
Rated value	24V DC		
Maximum allowable voltage	30V DC		
Surge voltage	35V DC，0.5s		
Signal "1"	15～30V		
Signal '0'	0～5V		
quarantine			
Photoelectric isolation	500V AC, duration within 1 minute		
Number of isolation points per group	four o'clock		eight o'clock
Maximum input latency	4.5ms		
Maximum leakage current	1mA AC		
Cable length (maximum)			
Unshielded	300m		
shield	500m		
The number of input points that can be connected simultaneously			
40℃	eight	sixteen	thirty-two
50℃	eight	sixteen	thirty-two

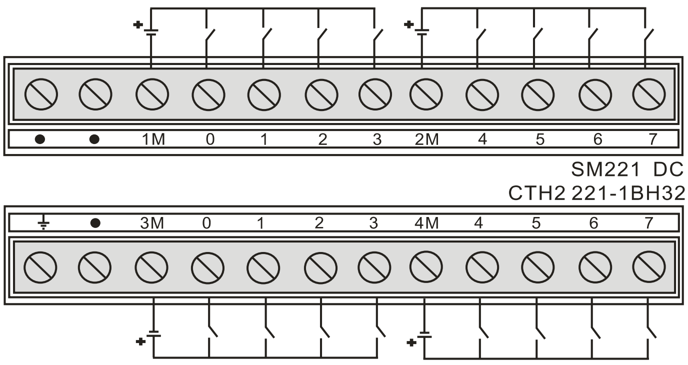
Wiring diagram

SM221 Digital Input Module (CTH2 221-1BF32)



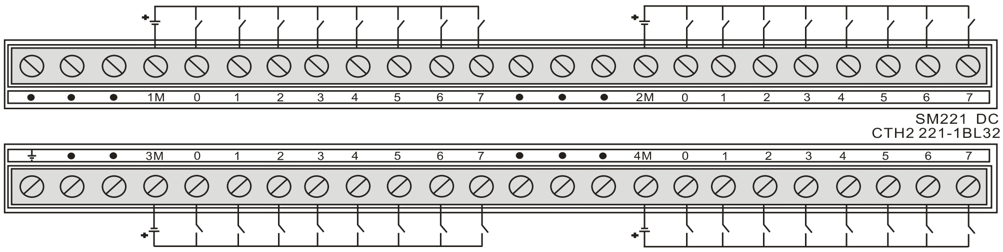
CTH2 221-1BF接线图

SM221 Digital Input Module (CTH2 221-1BH32)



CTH2 221-1BH接线图

SM221 Digital Input Module (CTH2 221-1BL32)



CTH2 221-1BL接线图

Specification for Digital Output Module

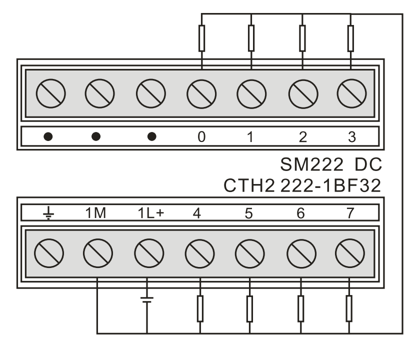
Table7-12 Specification for Digital Output (Transistor Output) Module

characteristic	8-point output, 24V DC	16 point output, 24V DC	32 point output, 24V DC
Order number	CTH2 222-1BF32 CTH2 222-1DF32	CTH2 222-1BH32	CTH2 222-1BL32
physical characteristics			
Dimensions (width x height x depth)	46 × 96 × 62mm	71.3 × 96 × 62mm	138 × 96 × 62mm
Power loss			
power loss	CTH2 222-1BF32：2W CTH2 222-1DF32：1W	3W	
+5V DC consumption current	CTH2 222-1BF32：57mA CTH2 222-1DF32：110mA	79mA	174mA
output characteristic			
Output points	eight	sixteen	thirty-two
type of output	CTH2 222-1BF32: Solid state MOSFET (PNP source type) CTH2 222-1DF32: Solid state MOSFET (NPN drain type)	Solid state MOSFET (PNP source type)	
output voltage			
Rated load voltage	24V DC		
Allowable voltage range	20.4～28.8V DC		

Signal "1"	Minimum 20V DC		
Signal '0'	Maximum 0.1V DC (10k Ω load)		
Output current			
Signal "1"	0.5A		
Output the number of groups	1	2	4
Output for each group	eight		
Simultaneously connect the output points	8	16	32
Maximum current per group	4A		
Lamp load	5W		
contact resistance	0.3Ω		
Maximum leakage current per point	10μA		
Surge current (maximum)	8A，100ms		
quarantine			
quarantine	Optocoupler isolation, 500V AC, duration within 1 minute		
Number of isolation points per group	eight o'clock		
Output delay (RL=50 Ω)			
Close Open	Maximum 50 μ s		
Open close	Maximum 200 μ s		
Cable length (maximum)			
Unshielded	150m		
shield	500m		

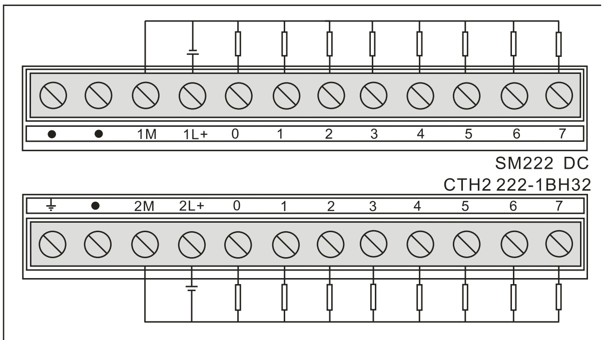
Wiring diagram

SM222 transistor output module (CTH2 222-1BF32)



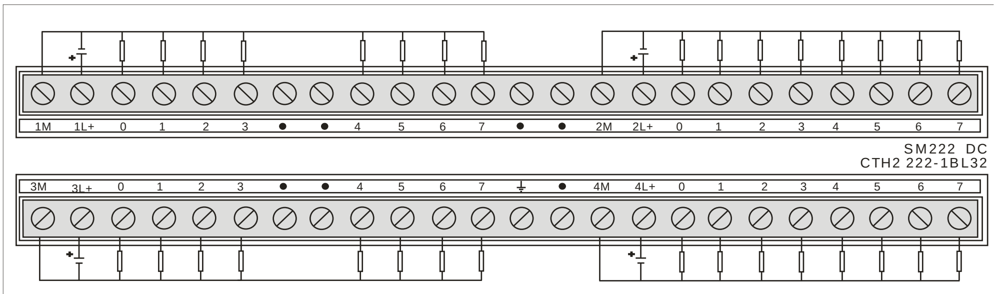
CTH2 222-1BF接线图

SM222 transistor output module (CTH2 222-1BH32)



CTH2 222-1BH接线图

SM222 transistor output module (CTH2 222-1BL32)



CTH2 222-1BL接线图

SM222 transistor NPN output module (CTH2 222-1DF32)

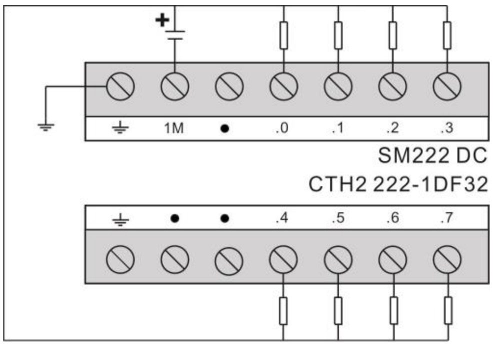


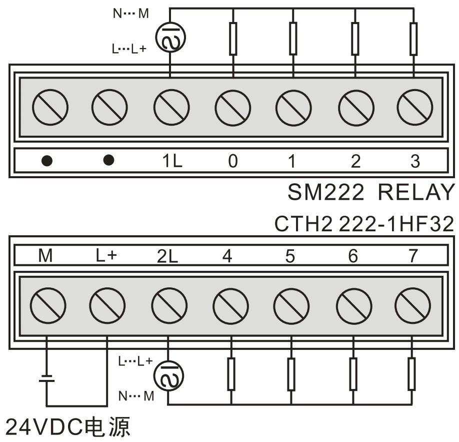
Table7-13 Specification for Digital Output (Relay Output) Module

characteristic	8 o'clock output, relay	16 point output, relay
Order number	CTH2 222-1HF32	CTH2 222-1HH32
physical characteristics		
Dimensions (width x height x depth)	46 × 96 × 62mm	71.3 × 96 × 62mm
Power loss		
power loss	2W	3W
From the backplane bus (+5V DC)	68mA	115mA
output characteristic		
Output points	8	16
type of output	Relay - Dry Contact	
Output voltage range	DC：5～30V，AC：5～250V	

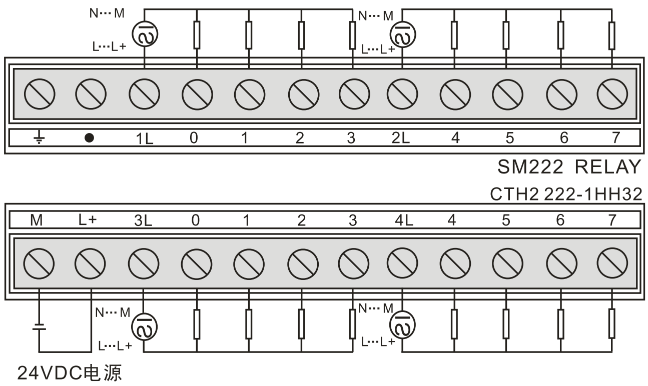
Output current		
Signal "1"	2A	
Output the number of groups	two	
Output points for each group	four	
Simultaneously output the number of ON points	8	16
Maximum current per group	8A	
Lamp load	5W	
contact resistance	0.2Ω	
Surge current (maximum)	7A, when the contact is closed	
Short circuit protection	Externally provided	
quarantine		
Number of isolation points per group	four o'clock	
Between the coil and the logic power supply	nothing	
Between the coil and the contact point	1500V AC, lasting for 1 minute	
Resistance (between coil and contact)	Minimum 100M Ω	
Relay characteristics		
Switch delay	15ms	
Switching frequency (maximum)	1Hz	
Mechanical switch frequency (no load)	30,000,000	
Contact life (rated load)	300,000	
Cable length (maximum)		
Unshielded	150m	
shield	500m	

Wiring diagram

SM222 Relay Output Module (CTH2 222-1HF32)



SM222 Relay Output Module (CTH2 222-1HH32)



Specification for Digital Input/Output Modules

Table7-14 Specification for Digital Input/Output (Transistor Output) Modules

characteristic	4-point I/O 24V DC	8-point I/O 24V DC	16 point I/O 24V DC
Order number	CTH2 223-1BF32	CTH2 223-1BH32	CTH2 223-1BL32
physical characteristics			
Dimensions (width x height x depth)	46 × 96 × 62mm	71.3 × 96 × 62mm	138 × 96 × 62mm
Power loss			
power loss	2W	3W	
+5V DC consumption current	57mA	73mA	115mA
Input characteristics			
Input points	4	8	16
Input type	Leakage type/source type		
input voltage			
Rated value	24V DC		
Maximum allowable voltage	30V DC		

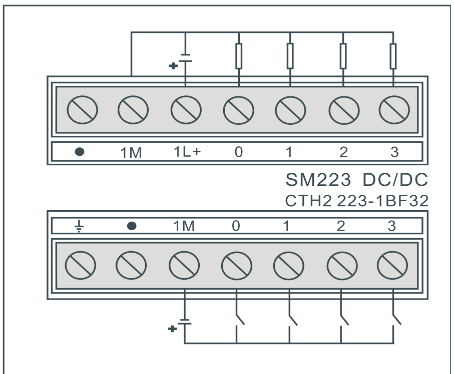
Surge voltage	35V DC，0.5s		
Signal "1"	15～30V		
Signal '0'	0～5V		
quarantine			
Optocoupler isolation	500V AC, duration within 1 minute		
Number of isolation points per group	four o'clock	eight o'clock	
Maximum input latency	4.5ms		
Maximum leakage current	1mA AC		
Cable length (maximum)			
Unshielded	300m		
shield	500m		
The number of input points that can be connected simultaneously			
40℃/50℃	4	8	16
output characteristic			
Output points	4	8	16
type of output	Solid state MOSFET		
output voltage			
Rated load voltage	24V DC		
Allowable voltage range	20.4～28.8V DC		
Signal "1"	Minimum 20V DC		
Signal '0'	Maximum 0.1V DC (10k Ω load)		
Output current			
Signal "1"	0.5A		
Output the number of groups	one		two
Output points for each group	four	eight	
Number of output points connected simultaneously	four	eight	sixteen
Maximum current per group	2A	4A	
Lamp load	5W		
contact resistance	0.3Ω		
Peak current at each point	10mA		

Surge current (maximum)	8A，100ms	
quarantine	Optocoupler isolation, 500V AC, duration within 1 minute	
Number of isolation points per group	four o'clock	4/4/8 o'clock
Output delay (RL=50 Ω)		
Close Open	Maximum 50ms	
Open close	Maximum 200ms	
Cable length (maximum)		
Unshielded	150m	
shield	500m	

> ⚠ **be careful**
> The transistor output type digital quantity expansion module has output short-circuit, overcurrent, and overvoltage protection functions.

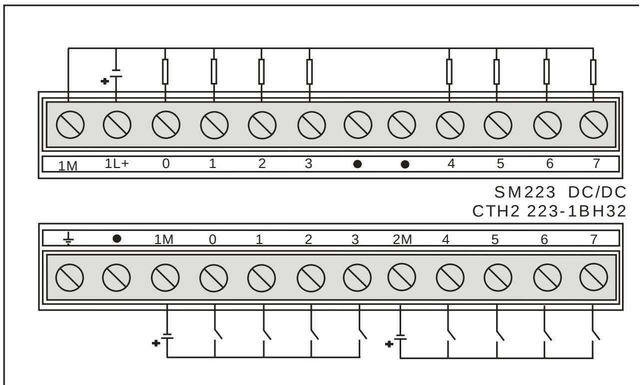
Wiring diagram

SM223 Digital I/O Module (CTH2 223-1BF32)



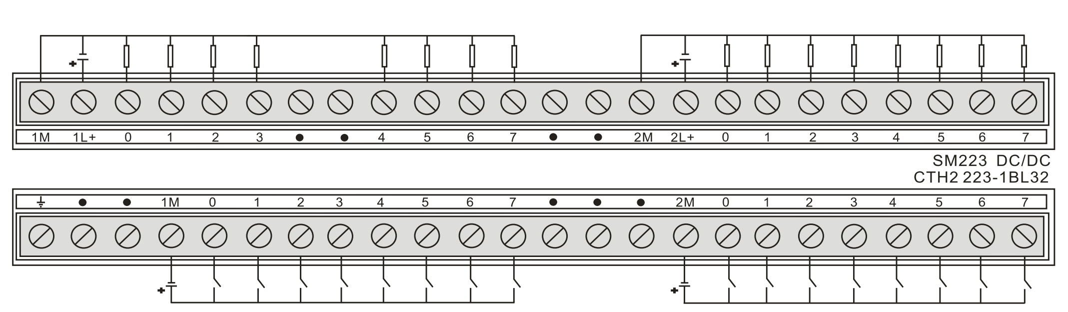
CTH2 223-1BF接线图

SM223 Digital I/O Module (CTH2 223-1BH32)



CTH2 223-1BH接线图

SM223 Digital I/O Module (CTH2 223-1BL32)



CTH2 223-1BL接线图

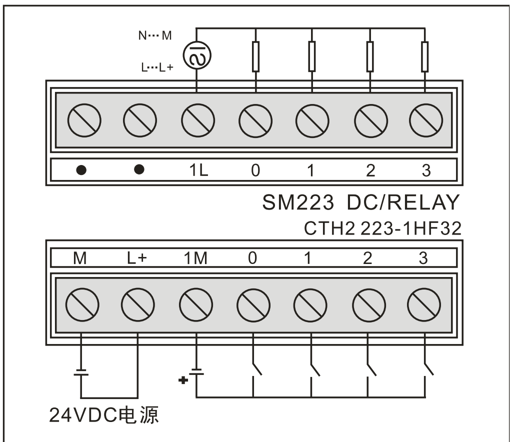
Table7-15 Specification for Digital Input/Output (Relay Output) Module

characteristic	4DI×24V DC 4DO × Relay	8DI×24V DC 8DO x relay	16DI×24V DC 16DO x relay
Order number	CTH2 223-1HF32	CTH2 223-1PH32	CTH2 223-1PL32
physical characteristics			
Dimensions (width x height x depth)	46 × 96 × 62mm	71.3 × 96 × 62mm	138 × 96 × 62mm
Power loss			
power loss	2W	3W	6W
+5V DC consumption current	58mA	89mA	150mA
L+	When connected, each output is 9mA		
L+coil voltage range	20.4 - 28.8V DC		
Input characteristics			
Input points	four	eight	sixteen
Input type	Leakage type/source type		
input voltage			
Rated value	24V DC		
Maximum allowable voltage	30V DC		
Surge voltage	35V DC，0.5s		
Signal "1"	15～30V		
Signal '0'	0～5V		
quarantine			
Optocoupler isolation	500V AC, duration within 1 minute		
Number of isolation points per group	four o'clock		eight o'clock
Maximum input latency	4.5ms		
Maximum leakage current	1mA AC		
Cable length (maximum)			
Unshielded	300m		

shield	500m		
The number of input points that can be connected simultaneously			
40℃/50℃	four	eight	sixteen
output characteristic			
Output points	four	eight	sixteen
type of output	Relay - Dry Contact		
Output voltage range	DC：5~30V，AC：5~250V		
Output current			
Signal "1"	2A		
Output the number of groups	one	two	four
Output points for each group	four		
Simultaneously connect the output points	four	eight	sixteen
Maximum current per group	8A		
Lamp load	DC：30W，AC：200W		
contact resistance	0.2Ω		
Surge current (maximum)	7A, contact closed		
Short circuit protection	Externally provided		
quarantine			
Number of isolation points per group	four o'clock		
Coil logic power supply room	nothing		
Coil - Contact	1500V AC, lasting for 1 minute		
Resistance (between coil and contact)	Minimum 100M Ω		
Relay characteristics			
Switch delay	15ms		
Switching frequency (maximum)	1Hz		
Mechanical switch frequency (no load)	30,000,000		
Contact life (rated load)	300,000		
Cable length (maximum)			
Unshielded	150m		
shield	500m		

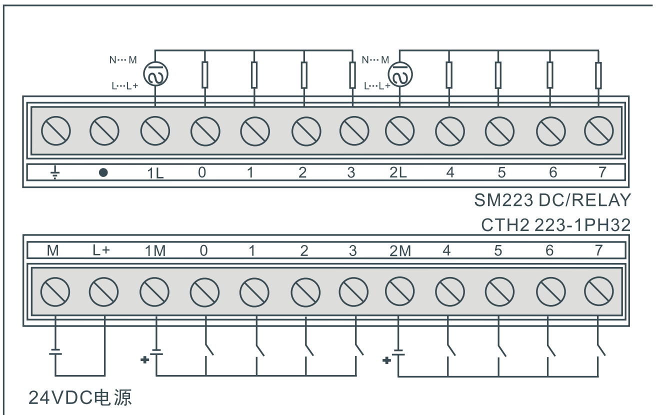
Wiring diagram

SM223 Digital I/O Module (CTH2 223-1HF32)



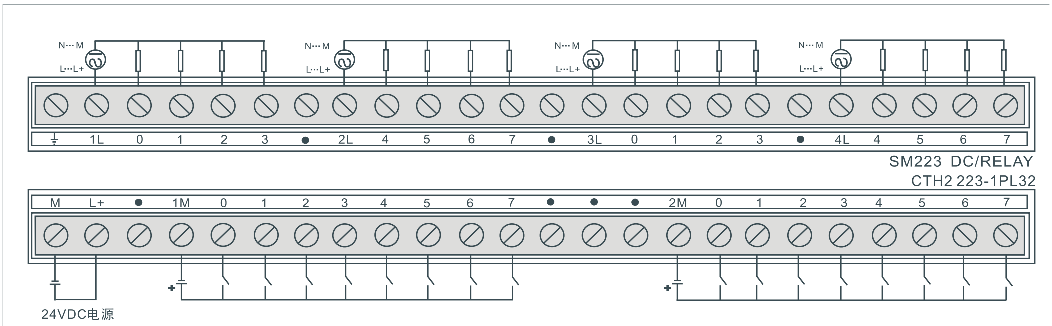
CTH2 223-1HF接线图

SM223 Digital I/O Module (CTH2 223-1PH32)



CTH2 223-1PH接线图

SM223 Digital I/O Module (CTH2 223-1PL32)



CTH2 223-1PL接线图

7.6.2 Specification for Analog Expansion Module

M226ES supports the analog expansion module of CTH200 series. The following table shows the ordering data of the analog expansion module. The subsequent chapters classify and introduce each module one by one.

Table7-16 Order data for analog expansion module

Specification parameters	Order number
SM231 analog input module, 4-point, 0-20mA current or ± 5V, ± 2.5V, 0-10V, 0-5V voltage input, isolated 12 bit precision	CTH2 231-0HC32
SM231 high-precision analog input module, 8-point input, voltage input, optically isolated 16 bit precision	CTH2 231-0HF32
SM231 high-precision analog input module, 8-point input, current input, optically isolated 16 bit precision	CTH2 231-1HF32

SM231 analog voltage input module, 8-point, ± 2.5V, 0-10V, 0-5V voltage input, two channel optional 0-20mA current input, isolated 12 bit precision	CTH2 231-5HF32
SM232 analog output module, 2-point, ± 10V voltage or 0-20mA current output, isolated voltage with 12 bit accuracy or current with 11 bit accuracy	CTH2 232-0HB32
SM232 analog output module, 4-point, ± 10V voltage or 0-20mA current output, isolated voltage with 12 bit accuracy or current with 11 bit accuracy	CTH2 232-0HD32
SM235 analog input/output module, 4-point voltage or current input/1-point voltage or current output, isolated voltage with 12 bit accuracy or current with 11 bit accuracy	CTH2 235-0KD32

Specification for Analog Input Module

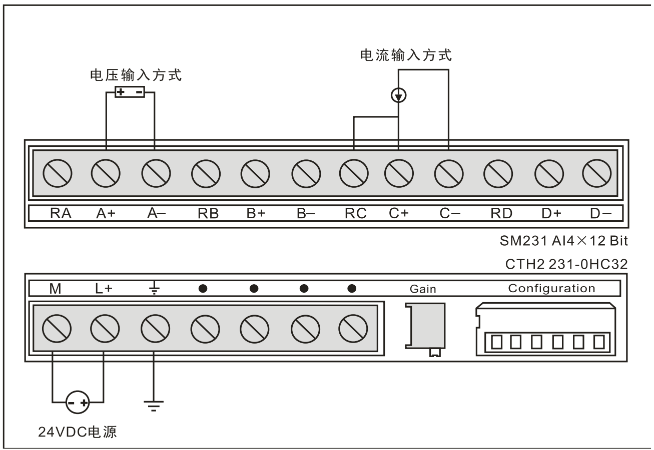
Table7-17 SM231-0HC specification

physical characteristics	
Dimensions (width x height x depth)	71.3×96×62mm
LED indicator light	24V power indicator light, on: indicates normal power supply, off: indicates power failure
Power loss	
+5V DC consumption current	87mA
L+	17mA
L+coil voltage range	20.4～28.8V DC
power loss	2W
Analog input characteristics	
Input points	four
Isolation (on-site and logical)	Optocoupler isolation: 500V AC, 1 minute
Input type	Differential input
Range of measurement	
Voltage input (unipolar)	0～10V，0～5V
Voltage input (bipolar)	±5V，±2.5V
Current input	0～20mA
Data Word Format	
Unipolar, full range	0~32000
Bipolar, full range	-32000～32000
Input resolution	
Voltage input (unipolar)	2.5mV (0-10V range). 1.25mV (0-5V range)
Voltage input (bipolar)	2.5mV (± 5V range). 1.25mV (± 2.5V range)
Current input	5 μ A (0-20mA range)
Analog to digital conversion time	Less than 300 μ s
Analog input response	1.5ms

time	
Common mode rejection	40dB，DC 60Hz
Common mode voltage	Signal voltage+common mode voltage (must be less than or equal to 12V)
Input impedance	Not less than 10M Ω
Maximum input voltage	30V
Maximum input current	30mA

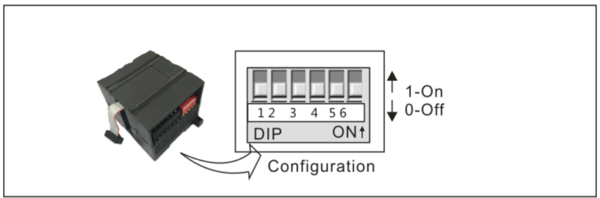
Wiring diagram

SM231 Analog Input Module (CTH2 231-0HC32)



Calibration and Range Selection

Calibration and Range Selection Switch Position



[Input Calibration]

Due to the impact of calibration adjustments on the operational amplifier after simulating multiple switches, it will affect all user input channels. In addition, due to the possible differences in the component parameters of each input channel before the multi way switch, even after calibration, there may be slight differences in the readings of the same input signal on different channels.

The input has been filtered internally in the module, and the measurement results are relatively stable. If better performance parameters are required, the analog input filter for all inputs of the module can be activated. When calculating the average value, select a sampling frequency of 64 or more.

The calibration steps are as follows:

1) Cut off the external power supply of the module and configure DIP switches to select the desired input range.

2) Connect the CPU and module power of the PLC system and wait for more than 15 minutes

3) Using a transmitter, a voltage source, or a current source, apply the full-scale value signal to an input channel

4) Read the measurement value of the input channel in the CPU

5) Adjust the GAIN potentiometer until the reading is 32000.

Range selection

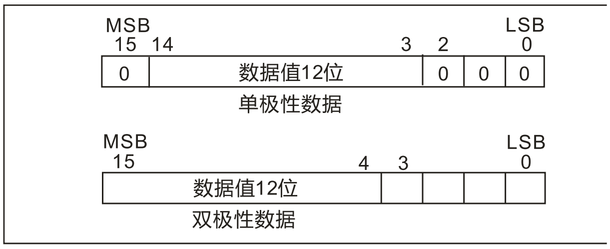
The table below shows how to set the range of the SM231 analog input module using DIP switches.

Switches 1, 2, and 3 can select the analog input range. All inputs are set to the same analog input range. In the table below, ON is ON and OFF is OFF. It should be noted that unused DIP switches SW4 to SW6 must be set to the OFF position.

Table7-18 DIP switch configuration for SM231-0HC

Unipolar			**Full range input**	**resolving power**
SW1	**SW2**	**SW3**		
ON	OFF	ON	0 - 10V	2.5mV
ON	ON	OFF	0 - 5V	1.25mV
ON	ON	OFF	0 - 20mA	5μA
Bipolar			**Full range input**	**resolving power**
SW1	**SW2**	**SW3**		
OFF	OFF	ON	±5V	2.5mV
OFF	ON	OFF	±2.5V	1.25mV

Input data format



Tips

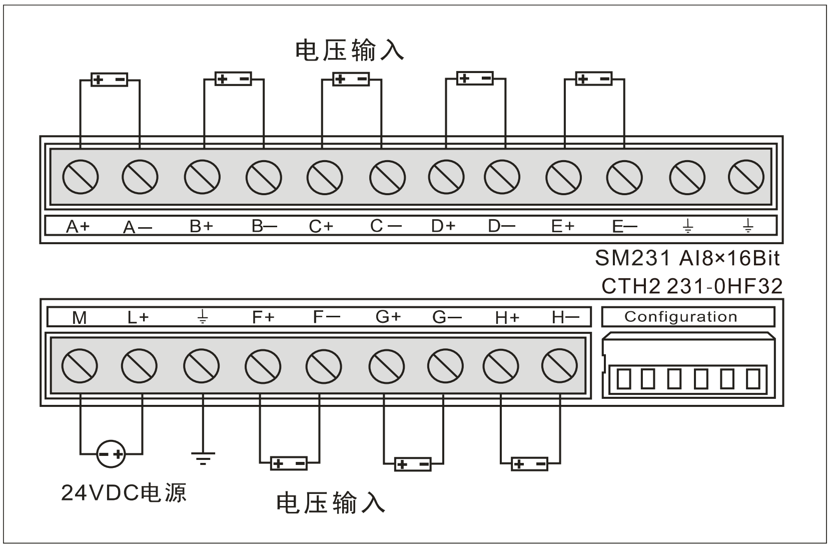
The 12 bit reading of an analog-to-digital converter (ADC) has a left end aligned data format. The most significant bit is the sign bit (0 represents a positive number). For unipolar format, three consecutive 0s result in a data word change of 8 units for every 1 unit change in ADC count value. For bipolar format, four consecutive 0s result in a data word change of 16 units for every 1 unit change in ADC count value

Table7-19 SM231-0HF Analog Input Module Specification

physical characteristics	
Dimensions (width x height x depth)	71×80×62mm
LED indicator light	24V power indicator light, on: indicates normal power supply, off: indicates power failure
Power loss	
+5V DC consumption current	87 mA
L+	31mA
L+coil voltage range	20.4 ~ 28.8V DC
power loss	2.5 W
Analog input characteristics	
Input points	eight

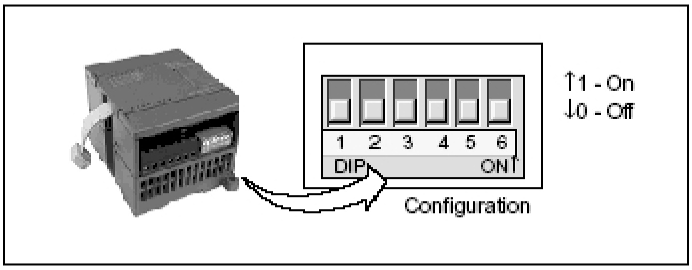
Isolation (on-site and logical)	Optocoupler isolation
Input type	Differential input
Range of measurement	
Voltage input (unipolar)	0～10V，0～5V
Voltage input (bipolar)	±5V，±2.5V
Data Word Format	
Unipolar, full range	0~32000
Bipolar, full range	-32000～32000
Input resolution	
Voltage input (unipolar)	300 μ V (0-10V range). 150 μ V (0-5V range)
Voltage input (bipolar)	150 μ V (± 5V range). 75 μ V (± 2.5V range)
Analog to digital conversion time	About 10ms
Eight channel conversion time	100ms
Common mode rejection	40dB，DC - 60Hz
Common mode voltage	Signal voltage+common mode voltage (must be less than or equal to 12V)
Input impedance	Not less than 10M Ω
Maximum input voltage	30V
Maximum input current	30mA
AD converter resolution	16 bits

Wiring diagram



Range selection and software configuration

Range selection switch position



Range selection

The following table shows how to set the range of SM231 8AI module using DIP switch. Switches 1, 2, and 3 can select the analog input range. All inputs are set to the same analog input range. In the table below, ON is ON and OFF is OFF. It should be noted that unused DIP switches SW4 to SW6 must be set to the OFF position.

Table7-19 DIP switch configuration for SM231-0HF

Unipolar			**Full range input**	**resolving power**
SW1	**SW2**	**SW3**		
ON	OFF	ON	0-10V	300μV
ON	ON	OFF	0 - 5V	150μV
Bipolar			**Full range input**	**resolving power**
SW1	**SW2**	**SW3**		
OFF	OFF	ON	± 5V	150μV
OFF	ON	OFF	± 2.5 V	75μV

[Software Configuration]

For the SM231 8AI × 16 bit analog input module, the reading is in VW instead of AIW, and the corresponding address varies depending on the relative position of the module. The address calculation formula is as follows:

<u>X (VWx)=slot number x 64+input channel number x 2</u>

The slot number corresponds to the installation position of the module. The slot number of the first expansion module adjacent to the CPU is 0, the slot number of the second expansion module is 1, and so on. There are a total of 8 input channels, from A to G, corresponding to numbers 0 to 7. The following table is the address lookup table calculated based on the above formula.

Table7-20 Address Quick Search Table

VWxx	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7
Slot 0	VW0	VW2	VW4	VW6	VW8	VW10	VW12	VW14
Slot 1	VW64	VW66	VW68	VW70	VW72	VW74	VW76	VW78
Slot 2	VW128	VW130	VW132	VW134	VW136	VW138	VW140	VW142
Slot 3	VW192	VW194	VW196	VW198	VW200	VW202	VW204	VW206
Slot 4	VW256	VW258	VW260	VW262	VW264	VW266	VW268	VW270
Slot 5	VW320	VW322	VW324	VW326	VW328	VW330	VW332	VW334

Slot 6	VW384	VW386	VW388	VW390	VW392	VW394	VW396	VW398

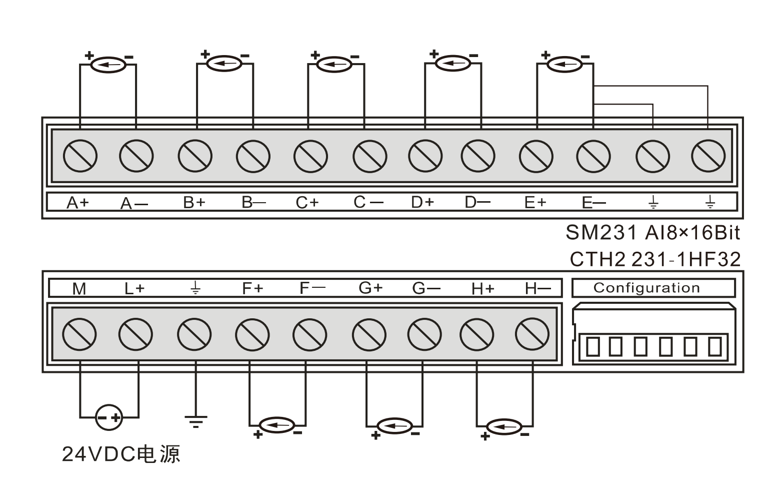
Tips
Due to the fixed address allocation of VW0 for COTRUST's TD2X text display and SIEMENS' TD200 text display panel, if you need to use both TD2X or TD200 and SM231 8AI modules in your system, the SM231 8AI module cannot be installed in the first position (Slot 0), otherwise it will not work properly.

表7-1 SM231-1HF Analog Input Module Specification

physical characteristics	
Dimensions (W × H × D) mm	120.5×80×62
weight	210 g
Plug in I/O terminal	no
Power consumption	
Bus (5V DC)	87mA
L+	30MA
L+voltage range	20.4～28.8V DC
power loss	1W
Analog input characteristics	
Input points	eight
Input type	electric current
Range of measurement	0～20mA； 4～20mA
Data Word Format	0~32000
resolving power	0.000625 mA （0～20mA）； 0.0005mA （4～20mA）
measurement accuracy	0.1%
Input impedance	250Ω
Update time	100ms (8 channels)
Common mode voltage	120V AC
Common mode rejection	90dB，60Hz @ DC
Input filtering attenuation	-3db @ 325 Hz
diagnosis	
Diagnostic program	+24V DC power indicator light, on: module power supply is normal, Extinguish: The module has no power supply, and all channel data becomes 32766 The dip switch is set incorrectly, and all channel data becomes 32767
Isolation characteristics	
On site side - Logic	500V AC

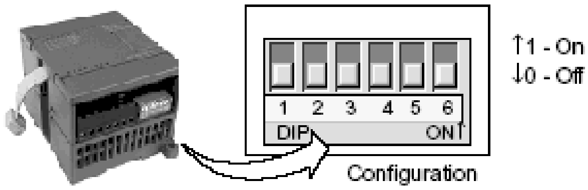
On site side - DC 24V	500V AC
DC 24V - Logic	500V AC

Wiring diagram



Range selection and software configuration

Range selection switch position



Range selection

The following table shows how to set the range of SM231 8AI module using DIP switch. Switches 1, 2, and 3 can select the analog input range. All inputs are set to the same analog input range.

In the table below, ON is ON and OFF is OFF.

It should be noted that unused DIP switches SW4 to SW6 must be set to the OFF position.

DIP switch configuration for SM231-1HF

Unipolar			**Full range input**	**resolving power**
SW1	**SW2**	**SW3**		
OFF	**OFF**	**OFF**	0-20mA	0.000625mA
	OFF	**ON**	4-20mA	0.0005mA

[Software Configuration]

For the SM231 8AI × 16 bit analog input module, the reading is in VW instead of AIW, and the corresponding address varies depending on the relative position of the module. The address calculation formula is as follows:

<u>X (VWx)=slot number x 64+input channel number x 2</u>

The slot number corresponds to the installation position of the module. The slot number of the first expansion module adjacent to the CPU is 0, the slot number of the second expansion module is 1, and so on. There are a total of 8 input channels, from A to G, corresponding to numbers 0 to 7. The following table is the address lookup table calculated based on the above formula.

Address Quick Search Table

VWxx	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7
Slot 0	VW0	VW2	VW4	VW6	VW8	VW10	VW12	VW14
Slot 1	VW64	VW66	VW68	VW70	VW72	VW74	VW76	VW78
Slot 2	VW128	VW130	VW132	VW134	VW136	VW138	VW140	VW142
Slot 3	VW192	VW194	VW196	VW198	VW200	VW202	VW204	VW206
Slot 4	VW256	VW258	VW260	VW262	VW264	VW266	VW268	VW270
Slot 5	VW320	VW322	VW324	VW326	VW328	VW330	VW332	VW334
Slot 6	VW384	VW386	VW388	VW390	VW392	VW394	VW396	VW398

Tips

Due to the fixed address allocation of VW0 for COTRUST's TD2X text display and SIEMENS' TD200 text display panel, if you need to use both TD2X or TD200 and SM231 8AI modules in your system, the SM231 8AI module cannot be installed in the first position (Slot 0), otherwise it will not work properly.

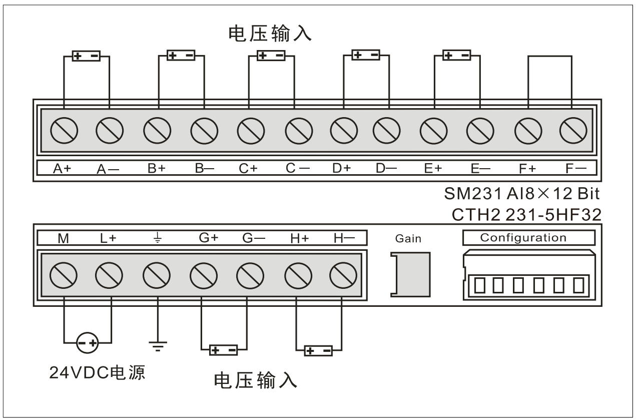
SM231-5HF specification

physical characteristics	
Dimensions (width x height x depth) mm	71.3 × 96 × 62
LED indicator light	24V power indicator light, on: indicates normal power supply, off: indicates power failure
Power loss	
+5V DC consumption current	87mA
L+	50mA
L+coil voltage range	20.4～28.8V DC
power loss	2.5W
Analog input characteristics	
Input points	eight
Isolation (between site and logic circuit)	Optocoupler isolation
Input type	Differential input, can select two channels for current
Range of measurement	
Voltage input (unipolar)	0～10V，0～5V
Voltage input (bipolar)	± 2.5V (channels 0-7)
electric current	0~20mA (channels 6-7)
Data Word Format	

Unipolar, full range	0~32000
Bipolar, full range	-32000 ~ 32000
Input resolution	
Voltage input (unipolar)	2.5mV (0-10V range). 1.25mV (0-5V range)
Voltage input (bipolar)	2.5mV (± 5V range). 1.25mV (± 2.5V range)
Analog to digital conversion time	About 10ms
Eight channel conversion time	100ms
Common mode rejection	40dB，DC - 60Hz
Common mode voltage	-12V ≤ signal voltage+common mode voltage ≤+12V
Input impedance	Voltage: ≥ 2M Ω Current: 250 Ω
Maximum input voltage	30V DC
Maximum input current	32mA
Resolution of A/D converter	Monopolar: 12 bits. Bipolar: 11 bits+sign bit
measuring principle	successive approximation
measurement error	0.5% (maximum)
Isolation (from site to logic)	500V AC
Analog input addressing	AIW addressing

Wiring diagram

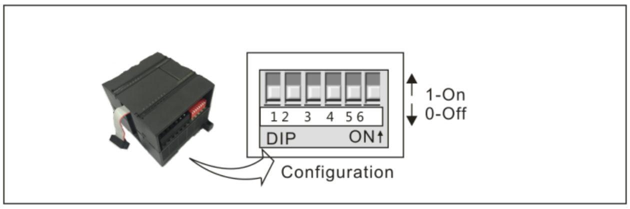
SM231 high-precision analog input module (CTH2 231-5HF32)



In the above figure, unused inputs such as F+and F - are short circuited.

Range selection and software configuration

Range selection switch position



Range selection

The following table shows how to set the range of SM231 8AI module using DIP switch. Switches 3, 4, and 5 select the analog input range, use switches 1 and 2 to select the current mode input, and when switch 1 is ON, select the current input mode of channel 6. Select voltage mode when OFF. When switch 2 is ON, select the current input mode of channel 7. Select voltage mode when OFF.

DIP switch configuration for SM231-5HF

Unipolar			Full range input	resolving power
SW3	**SW4**	**SW5**		
ON	OFF	ON	0-10V	2.5mV
ON	ON	OFF	0-5V	1.25mV
			0-20mA	5μA
Bipolar			**Full range input**	**resolving power**
SW3	**SW4**	**SW5**		
OFF	OFF	ON	±5V	2.5mV
OFF	ON	OFF	±2.5V	1.25mV

Specification for Analog Output Module

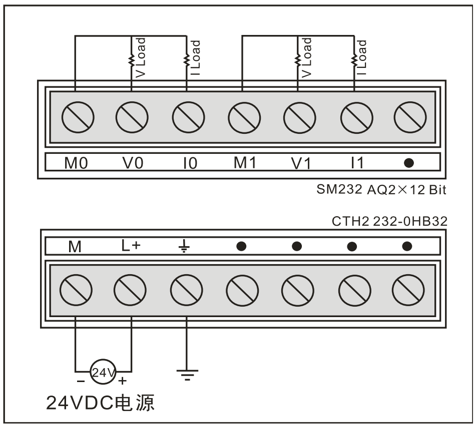
Specification for Analog Output Module

characteristic	SM232:2 AQ x 12 bits	SM232:4 AQ x 12 bits
Order number	CTH2 232-0HB32	CTH2 232-0HD32
physical characteristics		
Dimensions (width x height x depth)	46 × 96 × 62mm	71.3 × 96 × 62mm
LED indicator light	24V power indicator light, on: normal power supply, off: power failure	
Power loss		
+5V DC consumption current	87mA	
L+	61mA	112mA
L+coil voltage range	20.4～28.8V DC	
power loss	2W	
Analog output characteristics		
Output points	two	four
Isolation (analog to digital)	Optocoupler isolation: 500V AC, lasting for 1 minute	

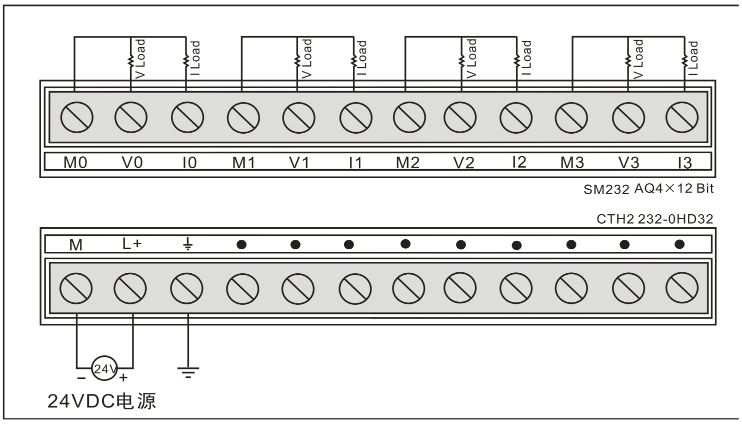
Output range	
Voltage output	±10V
Current output	0 ~ 20mA
Output resolution	
Voltage output	12 digits
Current output	11 people
Data Word Format	
Voltage output	-32000 ~ +32000
Current output	0~32000
measurement error	Typical scenario: Full range ± 0.5%. Worst case scenario: ± 2% of full scale
Stable time	
Voltage output	100μs
Current output	2ms
Maximum Drive @ 24V User Power Supply	
Voltage output	Minimum 5000 Ω
Current output	Maximum 500 Ω

Wiring diagram

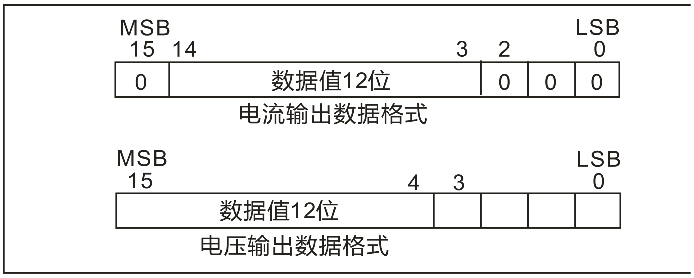
SM232 Analog Output Module (CTH2 232-0HB32)



SM232 Analog Output Module (CTH2 232-0HD32)



Output data format



Tips

The 12 bit reading of a digital to analog converter (DAC) has a left end aligned output data format, with the most significant bit being the sign bit (0 represents a positive number). Before loading the data into the DAC register, four consecutive 0s are truncated and do not affect the output signal value.

Specification for Analog Input/Output Modules

Specification for Analog Input/Output Modules

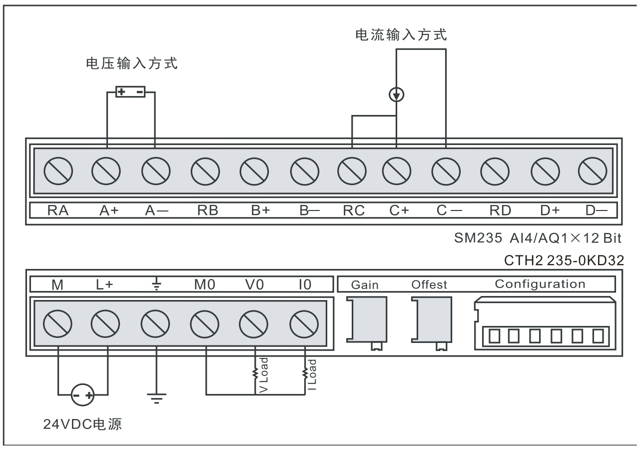
physical characteristics	
Dimensions (width x height x depth) mm	71.3 × 96 × 62
LED indicator light	24V power indicator light, on: indicates normal power supply, off: indicates power failure
Power loss	
+5V DC consumption current	87mA
L+	48mA
L+coil voltage range	20.4 ~ 28.8V DC
power loss	2W
Analog input characteristics	
Input points	four
Isolation (between site and logic circuit)	Optocoupler isolation: 500V AC, lasting for 1 minute
Input type	Differential input
Range of measurement	

Voltage input (unipolar)	0-10V，0-5V，0-1V，0-500mV，0-100mV，0-50mV
Voltage input (bipolar)	±10V，±5V，±2.5V，±1V，±500mV，±250mV，±100mV，±50mV，±25mV
Current input	0～20mA
Data Word Format	
Unipolar, full range	0~32000
Bipolar, full range	-32000~32000
Input resolution	
Voltage input (unipolar)	2.5mV (0-10V range). 1.25mV (0-5V range)
Voltage input (bipolar)	2.5mV (± 5V range). 1.25mV (± 2.5V range)
Current input	5 μ A (0-20mA range)
Analog to digital conversion time	Less than 300 μ s
Analog input response time	1.5ms
Common mode rejection	40dB，DC~60Hz
Common mode voltage	Signal voltage+common mode voltage (must be less than or equal to 12V)
Input impedance	Not less than 10M Ω
Maximum input voltage	30V
Maximum input current	30mA
Resolution of A/D converter	12 digits
Analog output characteristics	
Output points	one
Voltage output	±10V
Current output	0～20mA
Output resolution	
Voltage output	12 digits
Current output	11 people
Data Word Format	
Voltage output	-32000～+32000
Current output	0~32000
accuracy	
Voltage output	Typical scenario: Full range ± 0.5%, worst-case scenario: Full range ± 2%
Current output	
Stable time	

Voltage output	100μs
Current output	2ms
Maximum Drive @ 24V User Power Supply	
Voltage output	Minimum 5000 Ω
Current output	Maximum 500 Ω

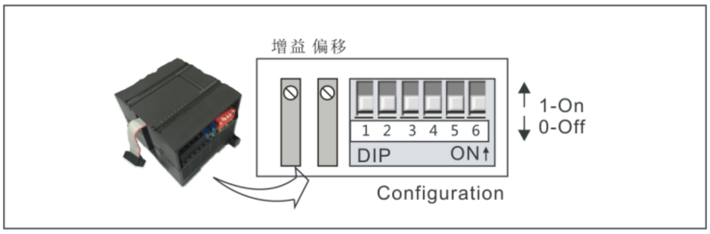
Wiring diagram

SM235 Analog Mixed Input/Output Module (CTH2 235-0KD32)



Calibration, Range and Gain Selection

Calibration and Range Selection Switch Position



[Input Calibration]

Due to the impact of calibration adjustments on the operational amplifier after simulating multiple switches, it will affect all user input channels. In addition, due to the possible differences in the component parameters of each input channel before the multi way switch, even after calibration, there may be slight differences in the readings of the same input signal on different channels.

The input has been filtered internally in the module, and the measurement results are relatively stable. If better performance parameters are required, the analog input filter for all inputs of the module can be activated. When calculating the average value, select a sampling frequency of 64 or more.

The calibration steps are as follows:
1) Cut off the external power supply of the module and configure DIP switches to select the desired input range.
2) Connect the CPU and module power of the PLC system and wait for more than 15 minutes.
3) Apply a zero value signal to one input of the module using a transmitter, a voltage input source, or a current input source.
4) Read the measurement value of the input terminal in the CPU.
5) Adjust the OFFSET potentiometer until the reading is zero, completing the zero calibration.

6) Next, input a full-scale value signal at the input terminal and read the measurement value reading from the CPU.
7) Adjust the GAIN potentiometer until the reading is 32000.
8) If necessary, repeat steps 3-7.

Range and gain selection

Tables 3-20 and 3-21 show how to set the SM235 module using DIP switches. Switches SW1 to SW6 allow for the selection of analog input range and resolution. All inputs are set to the same analog input range and format. Table 3-22 shows how to select single/bipolar (SW6), gain (switches SW4 and SW5), and attenuation (switches SW1, SW2, and SW3). In the following table, ON is ON and OFF is OFF.

DIP switch configuration for SM235-0KD (unipolar)

Unipolar						Full range input	resolving power
SW1	SW2	SW3	SW4	SW5	SW6		
ON	OFF	OFF	ON	OFF	ON	0~50mV	12.5μV
OFF	ON	OFF	ON	OFF	ON	0~100mV	25μV
ON	OFF	OFF	OFF	ON	ON	0 ~500mV	125μV
OFF	ON	OFF	OFF	ON	ON	0~1V	250μV
ON	OFF	OFF	OFF	OFF	ON	0~5V	1.25μV
ON	OFF	OFF	OFF	OFF	ON	0~20mA	5μA
OFF	ON	OFF	OFF	OFF	ON	0~10V	2.5mV

DIP switch configuration for SM235-0KD (bipolar)

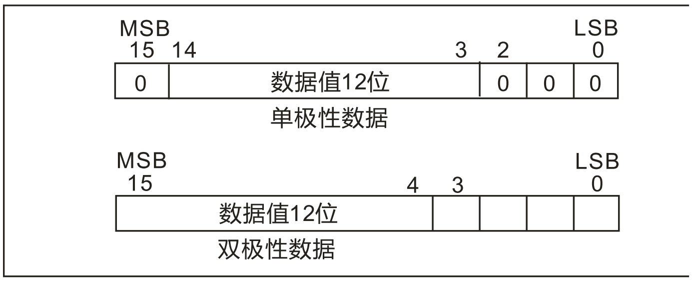
Bipolar						Full range input	resolving power
SW1	SW2	SW3	SW4	SW5	SW6		
ON	OFF	OFF	ON	OFF	OFF	±25mV	12.5μV
OFF	ON	OFF	ON	OFF	OFF	±50mV	25μV
OFF	OFF	ON	ON	OFF	OFF	±100mV	50μV
ON	OFF	OFF	OFF	ON	OFF	±250mV	125μV
OFF	ON	OFF	OFF	ON	OFF	±500mV	250μV
OFF	OFF	ON	OFF	ON	OFF	±1V	500μV
ON	OFF	OFF	OFF	OFF	OFF	±2.5V	1.25mV
OFF	ON	OFF	OFF	OFF	OFF	±5V	2.5mV
OFF	OFF	ON	OFF	OFF	OFF	±10V	5mV

SM235-0KD Single/Bipolar, Gain and Attenuation Switch Configuration

SM235 switch						Single/Bipolar Selection	Gain selection	Attenuation selection
SW1	SW2	SW3	SW4	SW5	SW6			
---	---	---	---	---	ON	Unipolar	---	---

---	---	---	---	---	OFF	Bipolar	---	---
---	---	---	OFF	OFF	---	---	x1	---
---	---	---	OFF	ON	---	---	x10	---
---	---	---	ON	OFF	---	---	x100	---
---	---	---	ON	ON	---	---	nothing	---
ON	OFF	OFF	---	---	---	---	---	zero point eight
OFF	ON	OFF	---	---	---	---	---	zero point four
OFF	OFF	ON	---	---	---	---	---	zero point two

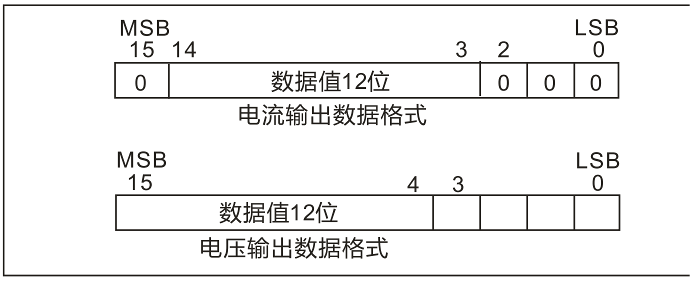
Input data format



Tips

The 12 bit reading of an analog-to-digital converter (ADC) has a left end aligned data format. The most significant bit is the sign bit (0 represents a positive number). For unipolar format, three consecutive 0s result in a data word change of 8 units for every 1 unit change in ADC count value. For bipolar format, four consecutive 0s result in a data word change of 16 units for every 1 unit change in ADC count value.

Output data format



Tips

The 12 bit reading of a digital to analog converter (DAC) has a left end aligned output data format, with the most significant bit being the sign bit (0 represents a positive number). Before loading the data into the DAC register, four consecutive 0s are truncated and do not affect the output signal value.

7.6.3 Specification for Temperature Measurement Module

M226ES supports a mixed temperature input module, with specific information as follows:

Ordering data for mixed temperature input module

Specification parameters	Order number
SM231 thermistor temperature input module, 2-point RTD, isolated 16 bit precision	CTH2 231-7PB32

SM231 thermistor temperature input module, 4-point RTD, isolated 16 bit precision	CTH2 231-7PC32
SM231 thermocouple temperature input module, 4-point TC, J/K/R/S/T/E/N, isolated 16 bit precision	CTH2 231-7PD32
SM231 thermocouple temperature input module, 8-point TC, J/K/R/S/T/E/N, isolated 16 bit accuracy	CTH2 231-7PF32
SM231 hybrid temperature input module, 2NTC/PT100, 2-point 0-20mA current or ± 5V, ± 10V, 0-10V, 0-5V voltage input, isolated 16 bit precision	CTH2 231-7ND32
SM231 thermistor temperature input module, 8NTC/PT100, isolated 16 bit precision	CTH2 231-7NF32

Specification for Thermal Resistance Measurement Module

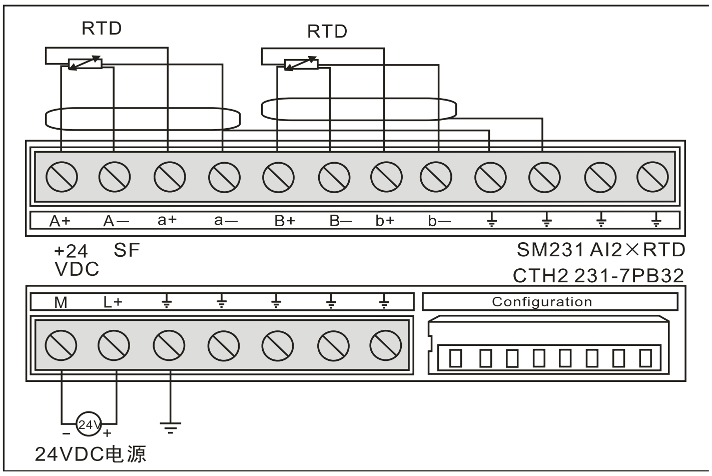
RTD Module Technical Specification

characteristic	SM231：2AI×RTD	SM231：4AI×RTD
Order number	CTH2 231-7PB32	CTH2 231-7PC32
physical characteristics		
Dimensions (width x height x depth)	71.3 × 96 × 62mm	
LED indicator light	24V DC power indicator light, ON=no error, OFF=no 24V DC power supply. SF indicator light, ON=module failure. Flashing=Over range or disconnected, OFF=No error	
Power loss		
+5V DC consumption current	87mA	
L+	34mA	
L+coil voltage range	20.4～28.8V DC	
power loss	1.7W	
Input characteristics		
Input type	Module reference grounding thermistor	
Input points	two	four
Input Range	Thermal resistance type (choose one): Pt-100 Ω, 200 Ω, 500 Ω, 1000 Ω (α=3850ppm, 3920ppm, 3850.55ppm, 3916ppm, 3902ppm) Pt-10000Ω（α=3850ppm） Cu-9.035Ω（α=4720ppm） Ni-100Ω，120Ω，1000Ω（α=6720ppm，6178ppm） R-150Ω，300Ω，600Ω	
Temperature measurement range	Pt-100Ω，200Ω，500Ω，1000Ω：-200℃～850℃ Pt-10000Ω：-200℃～600℃ NI-0.00672：-80℃～260℃ NI-0.006178：-60℃～300℃ Cu-0.004270：-200℃~260℃ If the measurement result exceeds the range, an error will be reported. Please refer to Table 3-25 for diagnostic details.	
quarantine		
From site to logic	500V AC	
On site to 24V DC	500V AC	

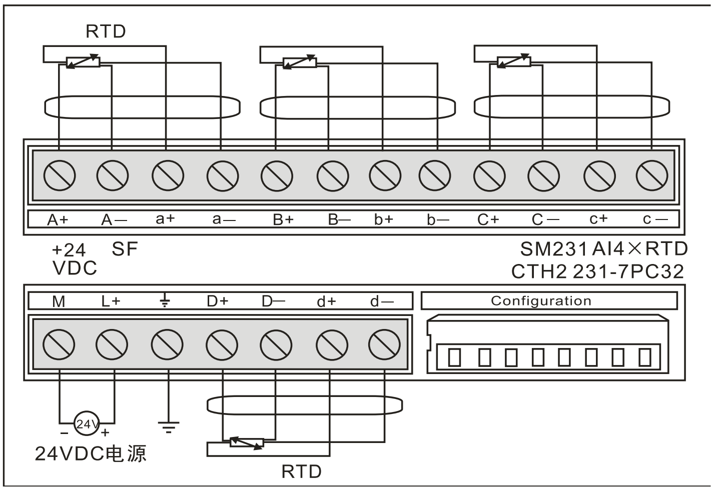
24V to Logic	500V AC	
Common mode input range (input channel to input channel)	0	
Common mode rejection	> 120dB@120V AC	
Sampling characteristics		
Temperature resolution	0.1℃/0.1℉	
Voltage resolution	15+symbol bit	
measuring principle	Σ-Δ	
Module update time (all channels)	425ms	825ms
The length of the wire to the sensor	Maximum 100 meters	
Wire circuit resistance	20 Ω, copper wire type 2.7 Ω	
Noise suppression	85dB@ 50Hz/60Hz/400Hz	
Data Word Format		
Input impedance	>10MΩ	
Maximum input voltage	30V DC (detection), 5V DC (source)	
resolving power	15 bits+symbol bit	
Input filtering attenuation	-3dB@21kHz	
Basic error	0.1% FS (resistance)	
Repeatability	0.05% FS	
Analog input addressing method	AIW addressing	

Wiring diagram

SM231 thermistor input module 2AI x RTD (CTH2 231-7PB32)



SM231 thermistor input module 4AI x RTD (CTH2 231-7PC32)



Diagnostic information of thermistor module

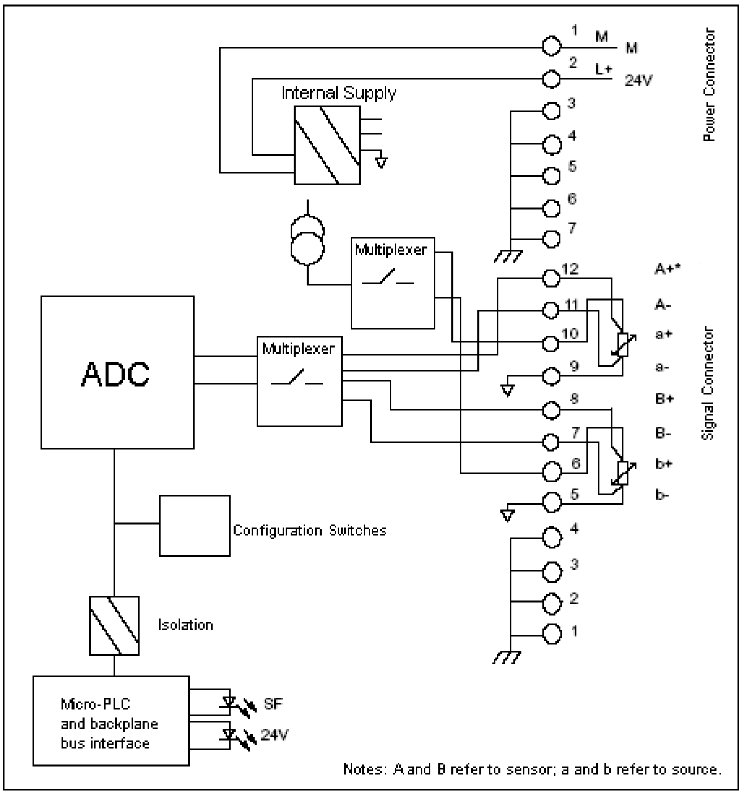
Error Type	Channel data	SF indicator light	24V indicator light	Range status bit	24V power supply failure
No module power supply	thirty-two thousand seven hundred and sixty-six	Extinguish	Extinguish	0	one
Broken wire	32767 (positive calibration) -32768 (negative calibration)	twinkle	light up	one	0
Exceeding the temperature range	32767 (positive calibration) -32768 (negative calibration)	twinkle	light up	one	0

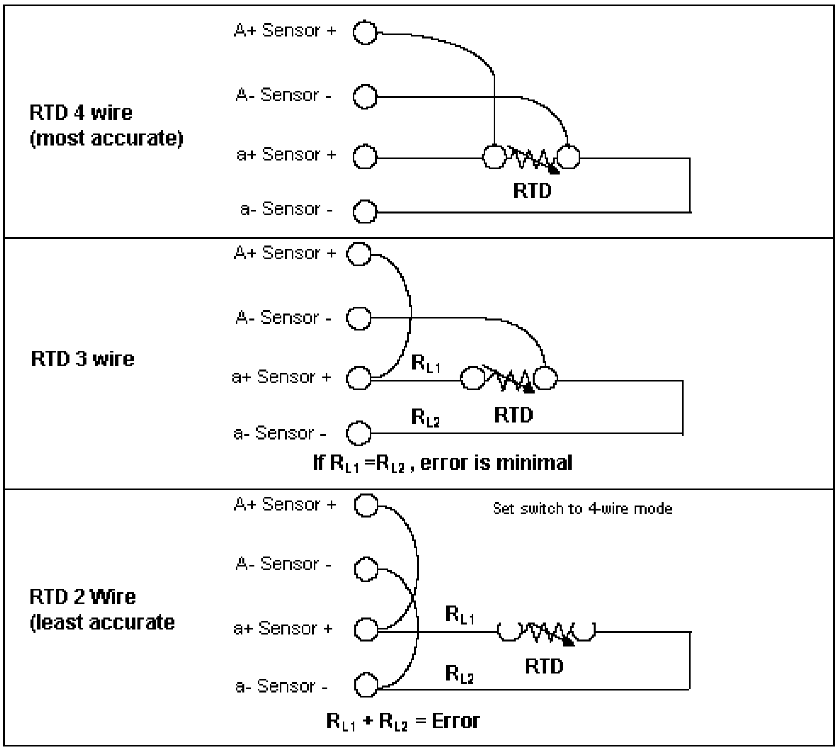
<Note>The range status bit is a bit in the module error register byte. You can check SMB8-21 according to the module order to query the corresponding module error code.

SMB8-21 diagnostic information

SMB8	Module 1 Identification Register
SMB9	Module 1 Error Register
SMB10	Module 2 Identification Register
SMB11	Module 2 Error Register
SMB12	Module 3 Identification Register
SMB13	Module 3 Error Register
SMB14	Module 4 Identification Register
SMB15	Module 4 Error Register
SMB16	Module 5 Identification Register
SMB17	Module 5 Error Register
SMB18	Module 6 Identification Register
SMB19	Module 6 Error Register
SMB20	Module 7 Identification Register
SMB21	Module 7 Error Register

Wiring between RTD and sensor





Notes: A refers to sensor; a refers to source

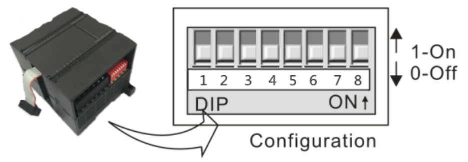
Users can directly connect the thermistor sensor to the SM231 thermistor module or use an extension wiring method. The best noise resistance can be achieved by using shielded wires. If users use shielded wires, they should be connected to the 1 to 4-pin grounding points of the signal connector. The grounding point is shared with pins 3 to 7 of the power connector. If some thermal resistance input channels are not in use, users should connect a resistor to the unused input channel to prevent errors caused by floating input signals from affecting the erroneous display of effective channels.

The user needs to connect the power supply to pins 1 and 2 of the power connector. The user must connect pin 3 of the power connector to the nearby chassis ground. Users can choose one of three wiring methods to connect the thermistor module to the sensor. The highest accuracy is the 4-wire connection method, and the lowest accuracy is the 2-wire connection method. It is recommended to use the 2-wire connection method only when users do not care about wire errors in their applications.

【 Select type of thermistor 】

The SM231 thermistor module enables PLC to easily connect with various thermistor sensors. Users can select the type of thermal resistor, wiring method, measurement unit, and direction of open circuit fault through DIP switches. All thermal resistors connected to the same module must be of the same type.

The DIP selection switch is located at the bottom right of the module, as shown in the following figure. To activate the DIP switch settings, the user needs to power off the PLC or 24V power supply before powering it on again.



SM231 Thermistor Module DIP Switch Configuration

Thermal resistance type	SW1	SW2	SW3	SW4	SW5
100 Pt 0.003850 (Default)	0	0	0	0	0
200Ω Pt 0.003850	0	0	0	0	one
500Ω Pt 0.003850	0	0	0	one	0
1000Ω Pt 0.003850	0	0	0	one	one
100Ω Pt 0.003920	0	0	one	0	0
200Ω Pt 0.003920	0	0	one	0	one
500Ω Pt 0.003920	0	0	one	one	0
1000Ω Pt 0.003920	0	0	one	one	one
100Ω Pt 0.00385055	0	one	0	0	0
200Ω Pt 0.00385055	0	one	0	0	one
500Ω Pt 0.00385055	0	one	0	one	0
1000Ω Pt 0.00385055	0	one	0	one	one
100Ω Pt 0.003916	0	one	one	0	0
200Ω Pt 0.003916	0	one	one	0	one
500Ω Pt 0.003916	0	one	one	one	0
1000Ω Pt 0.003916	0	one	one	one	one
100Ω Pt 0.00302	one	0	0	0	0
200Ω Pt 0.003902	one	0	0	0	one

500Ω Pt 0.003902	one	0	0	one	0
1000Ω Pt 0.003902	one	0	0	one	one
retain	one	0	one	0	0
100Ω Ni 0.00672	one	0	one	0	one
120Ω Ni 0.00672	one	0	one	one	0
1000Ω Ni 0.00672	one	0	one	one	one
100Ω Ni 0.006178	one	one	0	0	0
120Ω Ni 0.006178	one	one	0	0	one
1000Ω Ni 0.006178	one	one	0	one	0
10000Ω Pt 0.003850	one	one	0	one	one
10Ω Cu 0.004270	one	one	one	0	0
150Ω FS Resistance	one	one	one	0	one
300Ω FS Resistance	one	one	one	one	0
600Ω FS Resistance	one	one	one	one	

Wire breakage detection calibration direction, measurement unit, and thermal resistance wiring configuration

SW6	Calibration direction	SW7	Measurement unit	SW8	Wiring method
0	Positive calibration (+3276.7 degrees)	0	Celsius (℃)	0	3 lines
one	Negative calibration (-3276.8 degrees)	one	Fahrenheit (℉)	one	2-wire or 4-wire

Specification for Thermocouple Measurement Module

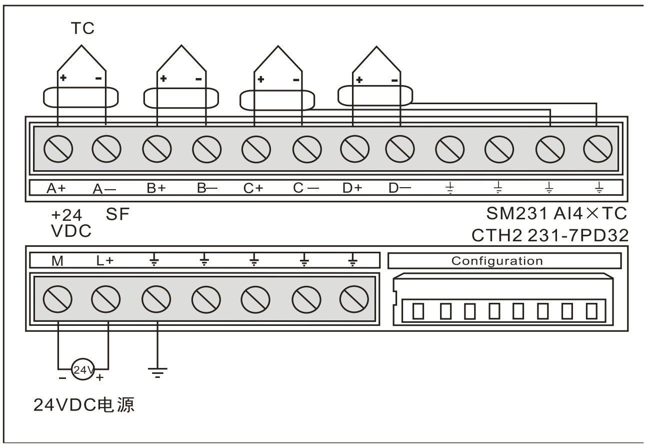
SM231 Thermocouple Input Module Specification

Specification project	SM231：4AI×TC	SM231：8AI×TC
physical characteristics		
Dimensions (width x height x depth) mm	71.3 × 96 × 62	
LED indicator light	24V DC power indicator light, ON=no error, OFF=no 24V DC power supply. SF indicator light, ON=module failure. Flashing=Over range or disconnected, OFF=No error	
Power loss		
+5V DC consumption current	87mA	
L+	30mA	
L+coil voltage range	20.4～28.8V DC	
power loss	1.7W	
Input characteristics		
Input type	Suspension type thermocouple	

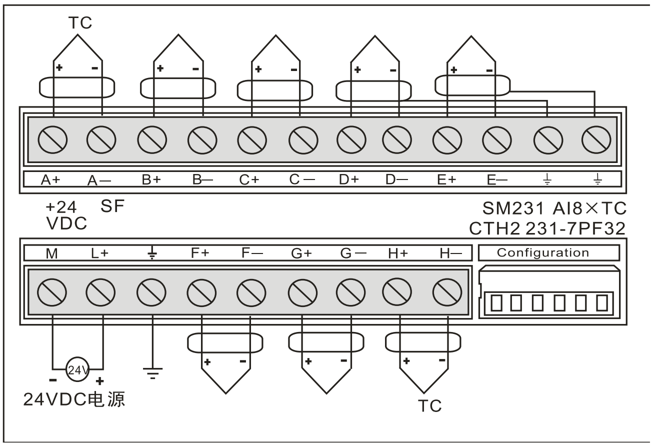
Input points	four	eight
Input Range	Thermocouple type (choose one): S, T, R, E, N, K, J Voltage range:+/-80mV **If** the measurement result exceeds the range, an error will be reported. **Please** refer to Table 3-25 for diagnostic details.	
Isolation characteristics		
From site to logic	500V AC	
On site to 24V DC	500V AC	
24V to Logic	500V AC	
Common mode input range (input channel to input channel)	120V AC	
Common mode rejection	> 120dB@120V AC	
Sampling characteristics		
Temperature resolution	0.1℃/0.1℉	
Voltage resolution	15+symbol bit	
Conversion principle	Σ-Δ	
Module update time (all channels)	425ms	825ms
The length of the wire to the sensor	Maximum 100m	
Wire circuit resistance	Maximum 100 Ω	
Noise suppression	85dB@ 50Hz/60Hz/400Hz	
Data Word Format	Voltage: -27648~+27648	
Input impedance	>1MΩ	
Maximum input voltage	30V DC	
resolving power	15+symbol bit	
Input filtering attenuation	-3dB@ 21kHz	
Basic error	0.1% FS (voltage)	
Repeatability	0.05% FS	
Cold junction error	±1.5℃	
Analog input addressing method	AIW addressing	VW addressing

Wiring diagram

SM231 thermocouple input module (CTH2 231-7PD32) x 4TC



SM231 thermocouple input module (CTH2 231-7PF32) x 8TC



Diagnostic information of thermocouple module

Error Type	Channel data	SF indicator light	24V indicator light	Range status bit	24V power supply failure
No module power supply	thirty-two thousand seven hundred and sixty-six	Extinguish	Extinguish	0	one
Broken wire	32767 (positive calibration) -32768 (negative calibration)	twinkle	light up	one	0
Exceeding the temperature range	32767 (positive calibration) -32768 (negative calibration)	twinkle	light up	one	0

<Note>The range status bit is a bit in the module error register byte. You can check SMB8-21 according to the module order to query the corresponding module error code.

SMB8-21 diagnostic information

Module 1 Identification Register	SMB8/SMB9
Module 2 Identification Register	SMB10/SMB11
Module 3 Identification Register	SMB12/SMB13
Module 4 Identification Register	SMB14/SMB15

Module 5 Identification Register	SMB16/SMB817
Module 6 Identification Register	SMB18/SMB819
Module 7 Identification Register	SMB20/SMB21

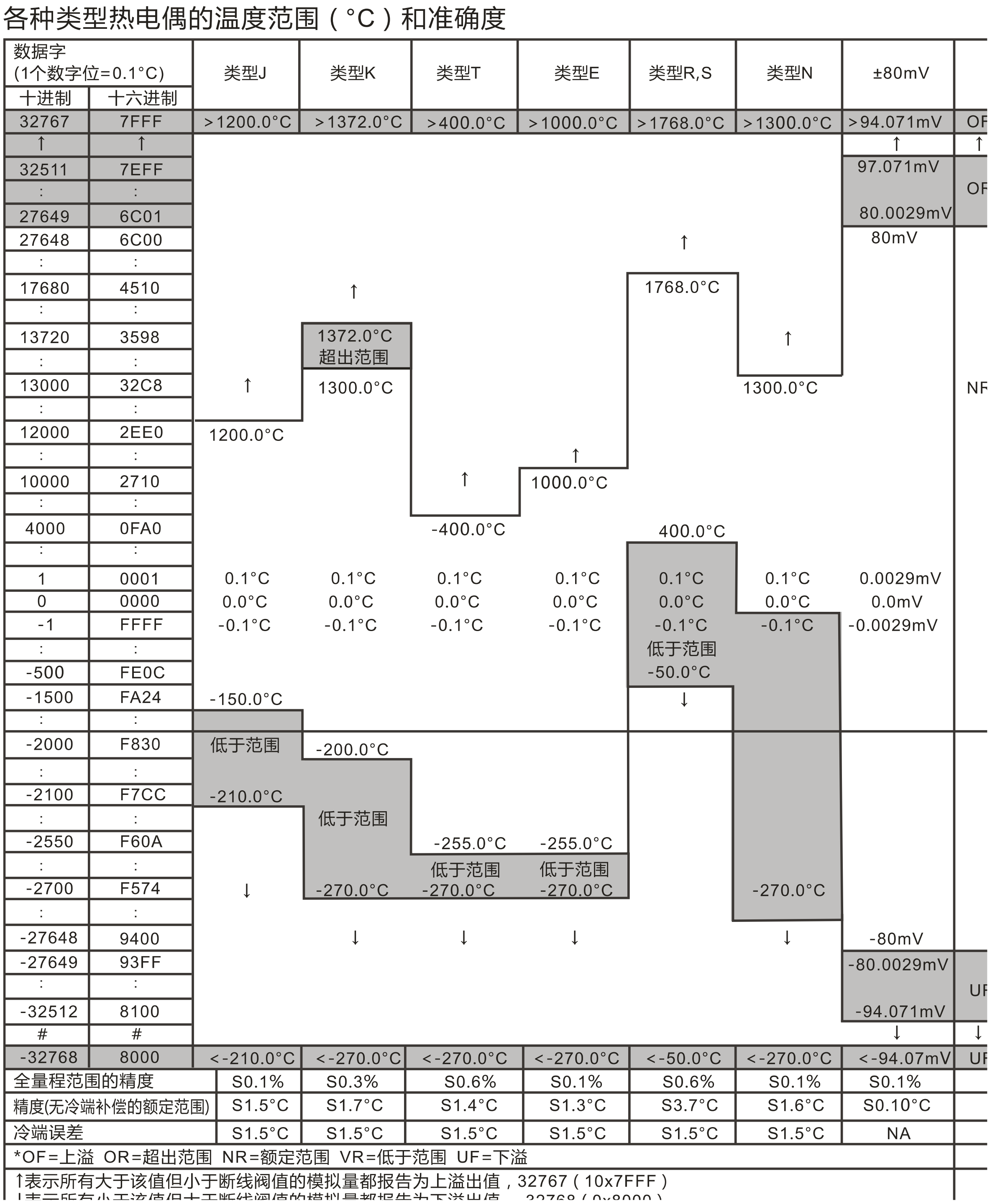


Figure 4-16 Measurement Range of Thermocouple

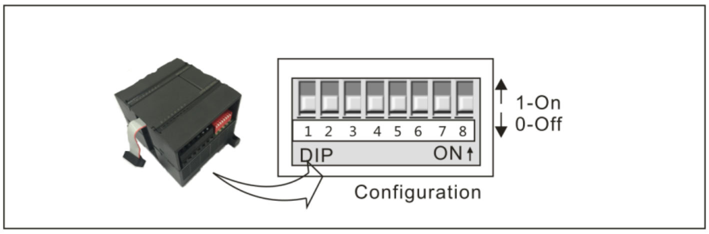
Switch configuration and software configuration

● DIP switch configuration

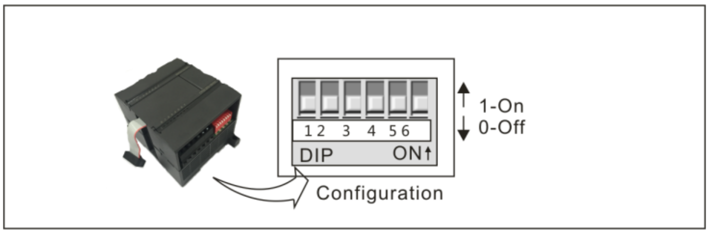
The SM231 thermocouple module provides a convenient, isolated interface for seven types of thermocouples: J, K, E, N, S, T, and R. It also allows for the connection of small analog signals (± 80mV range), and all thermocouples connected to the module must be of the same type, preferably using isolated thermocouple sensors.

The thermocouple module requires users to select the type of thermocouple, wire breakage inspection, measurement unit, cold junction compensation, and open circuit fault direction through DIP switches. Users can easily make the above selections through the configuration DIP switch located at the bottom of the module (as shown in the figure below).

For the SM231 4TC module, SW1-SW3 are used to select the thermocouple type, SW4 is not used (required to be set to the OFF position), SW5 is used to select the wire breakage detection direction, SW6 is used to select whether to perform wire breakage detection, SW7 is used to select the measurement unit, and SW8 is used to select whether to perform cold junction compensation.



For the SM231 8TC module, SW1-SW3 are used to select the thermocouple type, SW4 is used to select the wire breakage detection direction, SW5 is used to select the measurement unit, and SW6 is used to select whether to perform cold junction compensation. And whether to perform wire breakage detection is fixed to yes, without the need for user settings.



In order for the DIP switch settings to take effect, the user needs to power off and then power back on the PLC and/or user's power supply.

Thermocouple module DIP configuration:

Thermocouple type	SW1	SW2	SW3
J (default)	0	0	0
K	0	0	one
T	0	one	0
E	0	one	one
R	one	0	0
S	one	0	one
N	one	one	0
+/- 80mV	one	one	one

Other characteristics of SM231 thermocouple module

project	SM231 4TC		SM231 8TC	
Select project	Switch position	set up	Switch position	set up
Wire breakage detection direction	SW5	0: Positive calibration (+3276.7 degrees)	SW4	0: Positive calibration (+3276.7 degrees)
		1: Negative calibration (-3276.8 degrees)		1: Negative calibration (-3276.8 degrees)
Whether to perform wire breakage detection	SW6	0: Yes, 1: No	Fixed for wire breakage detection	
Selection of measurement units	SW7	0: Celsius, 1: Fahrenheit	SW5	0: Celsius, 1: Fahrenheit
Whether to perform cold end compensation	SW8	0: Yes, 1: No	SW6	0: Yes, 1: No

● **Software configuration**

For the SM231 8AI × TC thermocouple input module, the reading is in VW instead of AIW, and the corresponding address varies depending on the relative position of the module. The address calculation formula is as follows:

X (VWx)=slot number x 64+input channel number x 2

The slot number corresponds to the installation position of the module. The slot number of the first expansion module adjacent to the CPU is 0, the slot number of the second expansion module is 1, and so on. There are a total of 8 input channels, from A to H, corresponding to numbers 0 to 7. The following table is the address lookup table calculated based on the above formula.

Address Quick Search Table

VWxx	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7
Slot 0	VW0	VW2	VW4	VW6	VW8	VW10	VW12	VW14
Slot 1	VW64	VW66	VW68	VW70	VW72	VW74	VW76	VW78
Slot 2	VW128	VW130	VW132	VW134	VW136	VW138	VW140	VW142
Slot 3	VW192	VW194	VW196	VW198	VW200	VW202	VW204	VW206
Slot 4	VW256	VW258	VW260	VW262	VW264	VW266	VW268	VW270
Slot 5	VW320	VW322	VW324	VW326	VW328	VW330	VW332	VW334
Slot 6	VW384	VW386	VW388	VW390	VW392	VW394	VW396	VW398

Tips

Due to the fixed address allocation of VW0 for COTRUST's TD2X text display and SIEMENS' TD200 text display panel, if you need to use both TD2X or TD200 and SM231 8AI modules in your system, the SM231 8AI module cannot be installed in the first position (Slot 0), otherwise it will not work properly.

Specification for Hybrid Temperature Input Module

CTH2 231-7ND32 module specification

The SM231 NTC mixed analog input expansion module is an analog expansion module for PLC, providing 4-channel analog acquisition. Two channels are used to connect thermistor NTC temperature sensors or thermistor PT100 temperature sensors, and the other two channels are used to collect voltage/current signal inputs. The input accuracy (including sign bits) of all channels is 16BIT. Mainly used in sterilization equipment or central air conditioning equipment where both temperature measurement and pressure signal measurement are required.

SM231-7ND Module Technical Specification

Order number	CTH2 231-7ND32
physical characteristics	
Dimensions (width x height x depth) mm	71.3 × 96 × 62
LED indicator light	24V DC power indicator light, ON=no error, OFF=no 24V DC power supply. SF indicator light, ON=module failure. Flashing=Over range or disconnected, OFF=No error
Power loss	
+5V DC consumption current	87mA
L+	60mA
L+coil voltage range	20.4～28.8V DC
power loss	1.7W
Input signal	
Thermistor/Thermistor Input Range	Thermal resistance type (choose one): Pt-100（3850ppm，3920ppm，3850.55ppm，3916ppm，3902ppm） NTC（R25=10kΩ/B=3950，R25=10kΩ/B=3435）
Temperature measurement range	Pt-100：-50℃～850℃ NTC（R25=10k，B=3950）：-40℃～120℃ NTC（R25=10k，B=3435）：-40℃～150℃ If it exceeds the measurement range, an error will be reported**.** For diagnostic details, please refer to Table 3-25
Voltage input	0V～5V，0V～10V，±5V，±10V
Current input	0～20mA
Input points	2PT100/2NTC and 2AI
Isolation characteristics	
From site to logic	500V AC
On site to 24V DC	500V AC
24V to Logic	500V AC
Common mode rejection	＞ 120dB@120V AC
Sampling characteristics	
Temperature resolution	0.1℃/0.1℉
Voltage resolution	15 bits+symbol bit
Conversion principle	Σ-Δ
Module update time (all channels)	425ms
The length of the wire to the sensor	Maximum 100m
Wire circuit resistance	Maximum 20 Ω

Noise suppression	85dB@50Hz /60Hz/400Hz
Data Word Format	Temperature (NTC: R25=10k Ω, B=3950k): -400~1200 (limited to channels 1 and 2 only) Temperature (NTC: R25=10k Ω, B=3435k): -400~1500 (limited to channels 1 and 2 only) Temperature (PT100): -500~2000 (limited to channels 1 and 2 only) Voltage/Current: Unipolar 0-32000, Bipolar -32000-+32000 (limited to channels 3 and 4 only)
Input impedance	Voltage input>10M Ω. Current input=250 Ω. NTC input>10M Ω
Maximum input voltage	30V DC (detection), 5V DC (source)
resolving power	0.1℃/0.1°F
Input filtering attenuation	-3dB@21kHz
Basic error	0.1% FS (resistance)
Repeatability	0.05%FS
Analog input addressing method	AIW addressing

application environment

Working temperature: 0-55 ℃ for horizontal installation, 0-45 ℃ for vertical installation

■ Working humidity: 95% non condensing humidity

usage method

The SM231 NTC module is used to expand the analog measurement capability of PLC and is connected to the CPU module through a bus interface.

SM231-7ND module configuration settings

Input type	SW1	SW2	SW3	SW4	SW5
	Channels 1 and 2 are valid			Channels 3 and 4 are effective	
100 Ω PT0.003850 (factory setting)	0	0	0	-	-
100ΩPT0.003920	0	0	one	-	-
100ΩPT0.00385055	0	one	0	-	-
100ΩPT0.003916	0	one	one	-	-
100ΩPT0.003902	one	0	0	-	-
NTC：R25=10kΩ，B=3950k	one	0	one	-	-
NTC：R25=10kΩ，B=3435k	one	one	0	-	-
Disable	one	one	one	-	-
0 - 5V	-	-	-	0	0
0 - 20mA	-	-	-	0	0
0 - 10V	-	-	-	0	one

-10V - 10V	-	-	-	one	0
-5V - 5V	-	-	-	one	one

SW6	Calibration direction (valid for all channels)	SW7	Measurement unit	SW8	Wiring method
0	Positive calibration (+3276.7 degrees)	0	Celsius (℃)	0	3 lines
one	Negative calibration (-3276.8 degrees)	one	Fahrenheit (℉)	one	2-wire/4-wire

【 Startup steps 】

1) Connect a 24V DC power supply to the L+and M terminals as shown in the wiring terminal diagram.
2) Connect to the CPU module through the extended I/O bus.
3) Connect the CPU power supply and communication port.
4) Connect the input signal according to the wiring terminal diagram. The terminals of the idle channel should be suspended, and the grounding terminal (EARTH) should be connected to the earth according to the grounding specifications (single point grounding).
5) Configure the sensor type, calibration direction, etc. according to the configuration settings requirements.
6) Turn on the power supply for the CPU and modules.

【 Get input value 】

■ Users can read 2-byte input data for each channel in user programs or programming monitoring software (such as MagicWorks PLC software from our company or MicroWIN configuration software from Siemens):

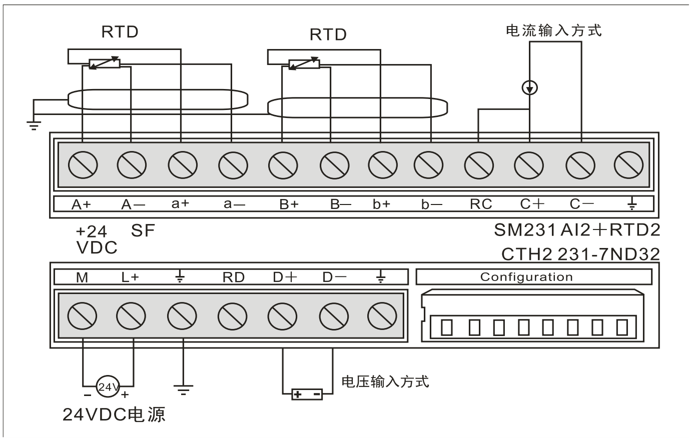
■ The NTC full-scale reading is -400~1200 (R25=10k B=3950), corresponding to a temperature range of -40.0~120.0 ℃. Or -400~1500 (R25=10k B=3435), corresponding temperature range: -40.0~150.0 ℃. The full range reading of PT100 is -500~2000, corresponding to a temperature range of -50.0~200.0 ℃. The return reading in case of over range or disconnection is determined by the calibration direction configuration switch (-32768 or 32767).

■ The full-scale reading of voltage/current is -32000~32000, and it remains valid even beyond the range of -32767/32765.

■ All channel readings without user power supply are 32766.

Wiring diagram

SM231 Hybrid Input Module (CTH2 231-7ND32)



CTH2 231-7NF32 module specification

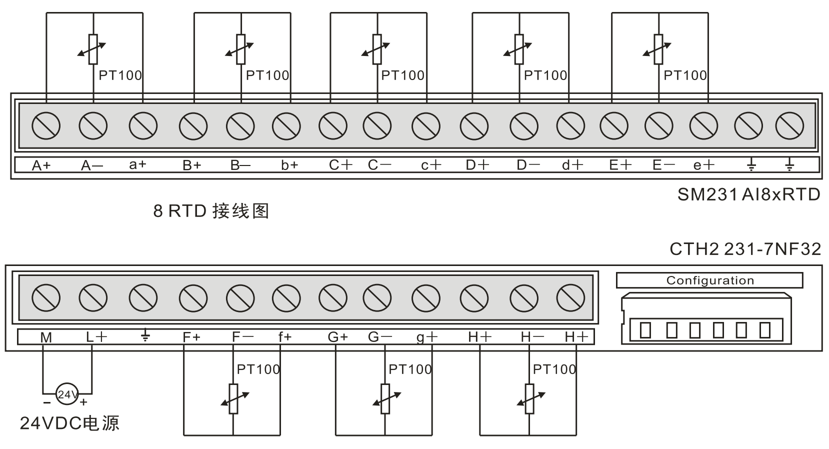
SM231-7NF module specification

project	CTH2 231-7NF32
Dimensions (W x H x D)	71.3 x 96 x 62mm
LED indicator light	24V DC power indicator light, ON=24V DC power supply is normal, OFF=no 24V DC power supply. SF indicator light, ON=module fault, flashing=out of range or disconnected, OFF=no error.
Power loss	
+5V DC consumption current	87mA
L+	32.5mA
L+voltage range	20.4～28.8V DC
power loss	1.8W
Input signal	
Input type	Pt100：α=3850PPm/3920PPM/3850.55PPM/3916PPM/3902PM NTC: R25=10k/B=3950 or R25=10k/B=3435
Temperature measurement range	Pt100：-50℃～800℃ NTC（R25=10k/B=3950）：-40℃～120℃ NTC（R25=10k/B=3435）：-40℃～150℃ If it exceeds the measurement range, an error will be reported. **Please** refer to Table 3-25 for diagnostic details.
Input points	8PT100/8NTC
Plug in I/O terminal	yes
Is the power supply isolated	yes
On site side - Logic	500V AC
On site side - DC 24V	500V AC
DC 24V Logic	500V AC
Common mode rejection	120dB@120V AC

Sampling characteristics	
Temperature resolution	0.1℃/0.1℉
measuring principle	Σ-Δ
Module update time (all channels)	825ms
Maximum cable length	100m
Maximum cable loop resistance	20Ω
Noise suppression	85dB@50 /400Hz
Input impedance	>1kΩ
Measurement accuracy	±0.3℃
Wire breakage detection	Pt100 detects 3 wires (NTC detects 2 wires), with a maximum detection time of 3 minutes

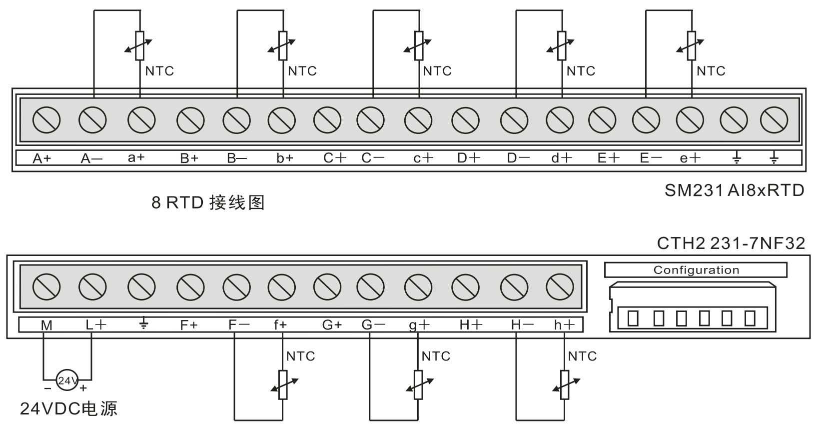
Wiring diagram

SM231 temperature input module Pt100 (CTH2 231-7NF32)



CTH2 231-7NF32 PT100接线图

SM231 Temperature Input Module NTC (CTH2 231-7NF32)



CTH2 231-7NF32 NTC接线图

SM231-7NF DIP switch configuration

Input type	SW1	SW2	SW3
100 Ω Pt 0.003850 (default)	**0**	**0**	**0**
100Ω Pt 0.003920	**0**	**0**	**one**
100Ω Pt 0.00385055	**0**	**one**	**0**
100Ω Pt 0.003916	**0**	**one**	**one**
100Ω Pt 0.00302	**one**	**0**	**0**
NTC R25=10k/B=3950	**one**	**0**	**one**
NTC R25=10k/B=3435	**one**	**one**	**0**
Disable	**one**	**one**	**one**

SW4	Calibration direction	SW5	Measurement unit	SW6	spare
0	Positive calibration (+3276.7 degrees)	0	Celsius (℃)	0	No impact
one	Negative calibration (-3276.8 degrees)	one	Fahrenheit (℉)	one	No impact

software configuration

For the SM231 8AI × 16 bit analog input module, the reading is in VW instead of AIW, and the corresponding address varies depending on the relative position of the module. The address calculation formula is as follows:

X (VWx)=slot number x 64+input channel number x 2

The slot number corresponds to the installation position of the module. The slot number of the first expansion module adjacent to the CPU is 0, the slot number of the second expansion module is 1, and so on. There are a total of 8 input channels, from A to G, corresponding to numbers 0 to 7. The following table is the address lookup table calculated based on the above formula.

Address Quick Search Table

VWxx	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7
Slot 0	VW0	VW2	VW4	VW6	VW8	VW10	VW12	VW14
Slot 1	VW64	VW66	VW68	VW70	VW72	VW74	VW76	VW78
Slot 2	VW128	VW130	VW132	VW134	VW136	VW138	VW140	VW142
Slot 3	VW192	VW194	VW196	VW198	VW200	VW202	VW204	VW206
Slot 4	VW256	VW258	VW260	VW262	VW264	VW266	VW268	VW270
Slot 5	VW320	VW322	VW324	VW326	VW328	VW330	VW332	VW334
Slot 6	VW384	VW386	VW388	VW390	VW392	VW394	VW396	VW398

Tips

Due to the fixed address allocation of VW0 for COTRUST's TD2X text display and SIEMENS' TD200 text display panel, if you need to use both TD2X or TD200 and SM231 8AI modules in your

	system, the SM231 8AI module cannot be installed in the first position (Slot 0), otherwise it will not work properly.

7.6.4 Specification for PID Control Module

M226ES supports PID control module, with specific information as follows:

Ordering data for PID control module

Specification parameters	Order number
SM231 thermocouple PID module, 4-point J/K type, with intelligent PID, isolated 16 bit accuracy	CTH2 231-7TD32
SM231 thermocouple PID module, 8-point J/K type, with intelligent PID, isolated 16 bit accuracy	CTH2 231-7TF32
SM231, 8-channel current input, 0-20mA/4-20mA, with intelligent PID, isolated 16 bit precision	CTH2 231-7HF32

【Main Features】

■ Bus, power supply, and channel are fully isolated, with high reliability and strong anti-interference ability.

■ 16 bit sampling accuracy, using hardware filtering technology, resulting in more accurate and stable measurement values.

■ The power supply has reverse protection and surge absorption functions, suitable for harsh industrial environments.

■ Integrating advanced fuzzy logic control algorithms, it does not occupy CPU resources and can achieve accurate temperature control without programming, with good dynamic performance.

■ PID control output can be PWM or analog, bipolar output, which can control heating and cooling.

[Usage Specification]

■ Insulated thermocouples should be used to achieve ideal anti-interference ability and improve reliability.

■ The signal line should use shielded wire, which needs to be grounded at one end.

■ The grounding terminal of the module should be connected to the ground wire when the system is well grounded, otherwise it will not be grounded.

■ Please short-circuit unused channels to eliminate the disconnection fault alarm.

Technical Specification for SM231-7TD/7TF Thermocouple PID Module

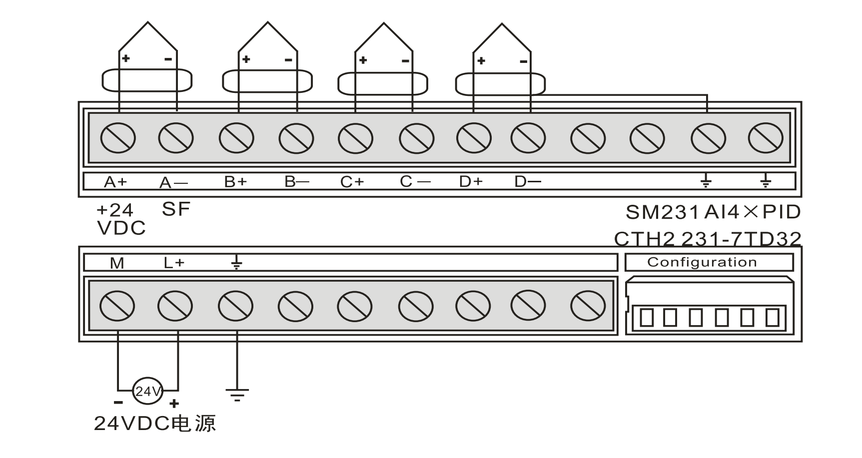
characteristic	SM231 4AI×TC PID	SM231 8AI×TC PID
Order number	CTH2 231-7TD32	CTH2 231-7TF32
physical characteristics		
Dimensions (width x height x depth)	71.3 × 96 × 62mm	71.3 × 96 × 62mm
LED light indication	24V DC power indicator light, ON=no error, OFF=no 24V DC power supply. SF indicator light, ON=module fault, flashing=out of range or disconnected, OFF=no error	
Power loss		
+5V DC consumption current	87mA	87mA
L+current	34mA	39mA
L+coil voltage range	20.4～28.8V DC	

power loss	1.8W	1.8W
Input characteristics		
Input type	Suspension type thermocouple	
Input Range	K-type thermocouple	
Input points	four	eight
Isolation characteristics		
From site to logic	500V AC	
On site to 24V DC	500V AC	
24V to Logic	500V AC	
Common mode input range (input channel to input channel)	120V AC	
Common mode rejection	> 120dB@120V AC	
Sampling performance		
Temperature resolution	0.1℃/0.1℉	
Voltage resolution	15+symbol bit	
measuring principle	Σ-Δ	
Module update time (all channels)	405ms	825ms
The length of the wire to the sensor	Maximum 100m	
Wire circuit resistance	Maximum 100 Ω	
Noise suppression	85dB@ 50Hz/60Hz/400Hz	
Data Word Format	Voltage: -27648~+27648	
Input impedance	>1MΩ	
Maximum input voltage	30V DC	
resolving power	15 bits+symbol bit	
Input filtering attenuation	-3dB@ 21kHz	
Basic error	0.1% FS (voltage)	
Repeatability	0.05% FS	
Cold junction error	±1.5℃	
Diagnostic program	LED：EXTF，SF	
PID characteristics		
PID algorithm	PID+FUZZY parameter self-adjusting	
Sampling time	1s	

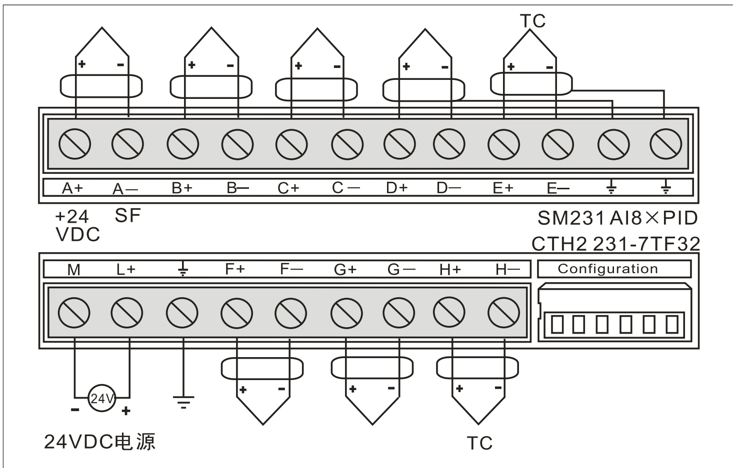
Output minimum pulse width	10ms
PID type	P. PI, PD, PID type
PID output type	Analog or PWM pulse width control
PID output polarity	Bipolar or unipolar

Wiring diagram

SM231 Thermocouple PID Module (CTH2 231-7TD32)



SM231 Thermocouple PID Module (CTH2 231-7TF32)



PID address and parameter configuration

PID address calculation formula

Address Name	Calculation formula	remarks
PID parameter address	A= (2048+S*256) +16*C	S is the slot number (range: 0-6) where the module is located, C is the channel number, SM231-7TF is 0-7, SM231-7TD is 0-3
PID forward pulse output address	X= (2048+S*256) +12	
PID negative pulse output address	Y= (2048+S*256) +13	

PID parameter output section (module to CPU)

content	address	Range of numerical settings	Actual corresponding numerical value
Actual temperature	VW A	-2000～13000	-200-1300 degrees

Status word	VW A+2		-
PID analog output	VW A+4	-32000～32000	-

PID parameter input section (CPU to module)

content	address	Range of numerical settings	Actual corresponding numerical value
setting temperature	VW A+128	-2000～13000	-200-1300 degrees
Control bytes		When VB A+130 bits are equal to 0	When VB A+130 bits equals 1
	V（A+130）.0	PID not running, no output	PID operation
	V（A+130）.1	The integral always works, and the proportional coefficient Kp does not automatically adjust	Integral separation and automatic adjustment of proportional coefficients
	V（A+130）.2	PID unipolar output, 0-32000	PID bipolar output, -32000~32000, with heating and cooling functions
	V（A+130）.3	not used	
	V（A+130）.4	Points work	Points do not work
	V（A+130）.5	Differential action	Differentiation does not work
	V（A+130）.6	Actual temperature value filtering, stronger anti-interference ability	The actual temperature value is not filtered
PID pulse output cycle setting	VW A+132	1~255	1～255s
Kp (proportionality coefficient)	VW A+134	0~9999	0～999.9
Ti (integration time)	VW A+136	0~3600	0～3600s
Td (differential time)	VW A+138	0~3600	0～3600s
Range	VW A+140	0~13000	0~1300 degrees

Forward pulse output address

Pulse output channel	Pulse output address	
0	V X.0	V Y.0
one	V X.1	V Y.1
two	V X.2	V Y.2
three	V X.3	V Y.3
four	V X.4	V Y.4
five	V X.5	V Y.5
six	V X.6	V Y.6
seven	V X.7	V Y.7

Negative pulse output address

Pulse output channel	Pulse output address
0	V Y.0
one	V Y.1

two	V Y.2
three	V Y.3
four	V Y.4
five	V Y.5
six	V Y.6
seven	V Y.7

Application example:

Calculate the address of the last PID loop of SM231-7TF on the second expansion module.

Firstly, calculate the address: S=1, C=7

Address A, A=2048+1 * 256+16 * 7=2416

X address, X=2048+1 * 256+12=2316

Y address, Y=2048+1 * 256+13=2317

Then set or read the parameter content according to the following parameter address instructions:
VW2544//Set temperature
VB2546//Control word (parameter self adjustment, bipolar output)
VW2548//Pulse output period
VW2550//Kp proportional coefficient
VW2552//Ti integration time (seconds)
VW2554//Td differential time (seconds)
VW2416//Actual temperature
VW2418//Status word
VW2420//PID analog output
V2316.7//Positive pulse output
V2317.7//Negative pulse output

To ensure the normal use of the PID module, please do not use the V storage area occupied by the PID module you are using when writing other program blocks. The SM231 PID specific parameter configuration library can be called.

The addresses occupied by the module in slot 0 are VW2048 to VW2298

The addresses occupied by the module in slot 1 are VW2304 to VW2554

The addresses occupied by the module in slot 2 are VW2560 to VW2810

The addresses occupied by the module in slot 3 are VW2816 to VW3066

The addresses occupied by the module in slot 4 are VW3072 to VW3322

The addresses occupied by the module in slot 5 are VW3328 to VW3578

The addresses occupied by the module in slot 6 are VW3584 to VW3834

The ladder diagram for parameter configuration is as follows:

PID模块的参数地址占用CPU的V存储区。

为了保证PID模块能正常使用，编写其他程序块时请一定不要使用您所用的PID模块占用的V存储区。

模块在第0号插槽所占用的地址为：VW2048到VW2298

模块在第1号插槽所占用的地址为：VW2304到VW2554

模块在第2号插槽所占用的地址为：VW2560到VW2810

模块在第3号插槽所占用的地址为：VW2816到VW3066

模块在第4号插槽所占用的地址为：VW3072到VW3322

模块在第5号插槽所占用的地址为：VW3328到VW3578

模块在第6号插槽所占用的地址为：VW3584到VW3834

网络 1

设置第1个扩展模块（插槽号为0）上的EM231 PID模块第1个PID回路（通道号为0）的参数：

假设该回路调试完成后需要再改变PID参数，我们调用PIDSetting来设定该回路的参数。

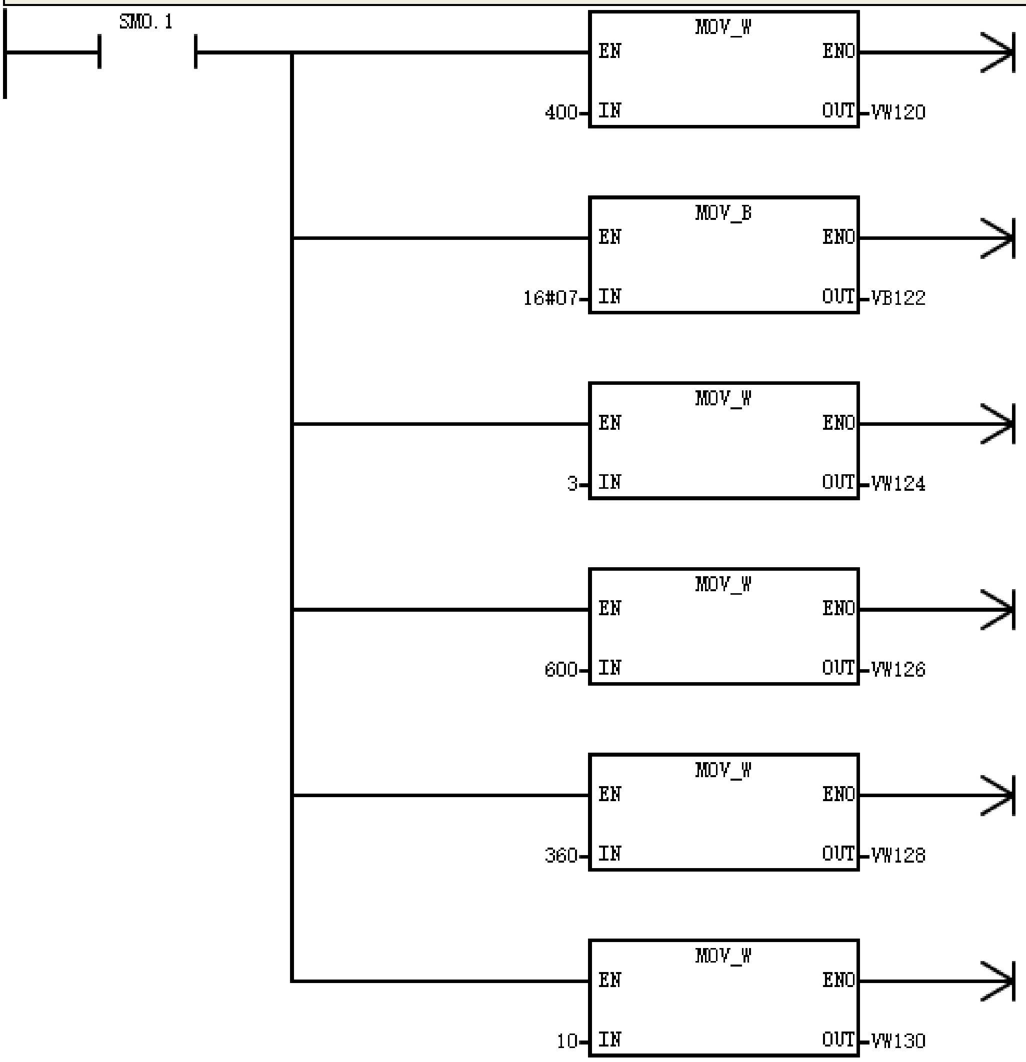
无需计算PID参数地址，只需输入回路所在的插槽号和通道号。

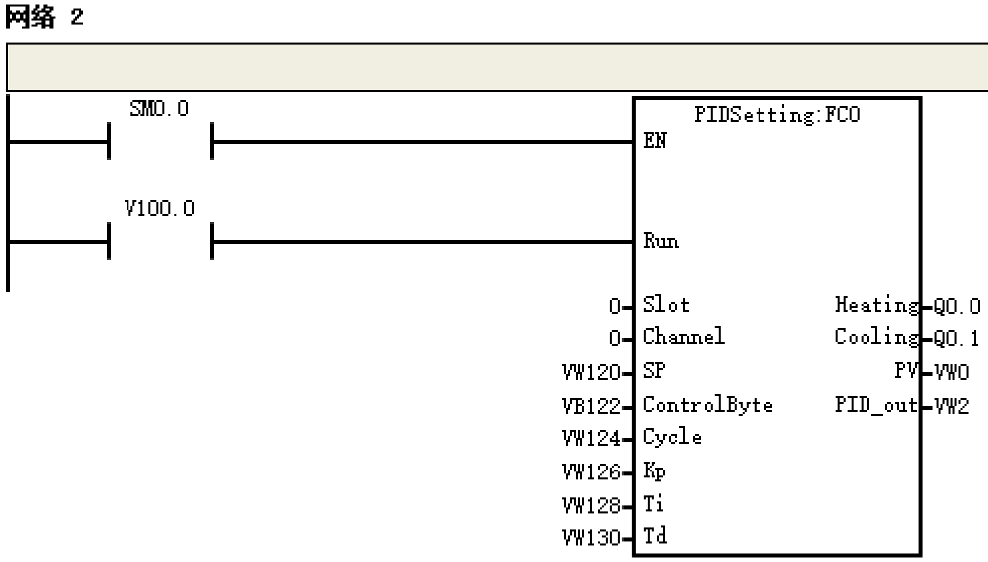
使能Run来运行该回路。

本例包含网络1和网络2。

Q0.0为正向脉冲输出，Q0.1为负向脉冲输出。 VW0为实际温度 ，VW2为PID模拟量输出。

修改PID设定参数使用其他地址，本例设定温度对应地址使用VW120，控制字：VB122，脉冲输出周期：VW124，比例系数：VW126， 积分时间：VW128，微分时间：VW130

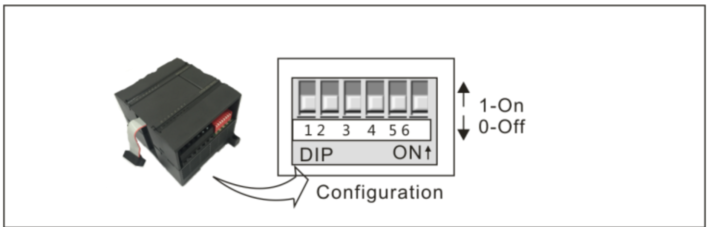




DIP switch configuration

The SM231 thermocouple temperature control module supports J/K type thermocouples and provides a 6-digit DIP switch for selecting measurement units, cold end compensation, and open circuit fault direction. Like the SM231 8TC, the SM231 thermocouple temperature control module enforces wire breakage detection for wire breakage detection.

The 6-digit DIP switch is located below the module, as shown in the following figure. Among them, SW1-SW2 are not used, SW3 is used to select the thermocouple type, SW4 is used to select the wire breakage detection direction, SW5 is used to select the measurement unit, and SW6 is used to select whether to perform cold junction compensation. In order for the DIP switch settings to work, the user needs to power off and then power on the PLC power supply.



SM231 switch configuration

Set up project	Switch position	set up
Thermocouple type selection	SW3	0: J type. 1: K-type
Wire breakage detection direction	SW4	0: Positive calibration (+3276.7 degrees). 1: Negative calibration (-3276.8 degrees)
Selection of measurement units	SW5	0: Celsius degrees. 1: Fahrenheit degrees
Whether to perform cold end compensation	SW6	0: Yes. 1: No, no

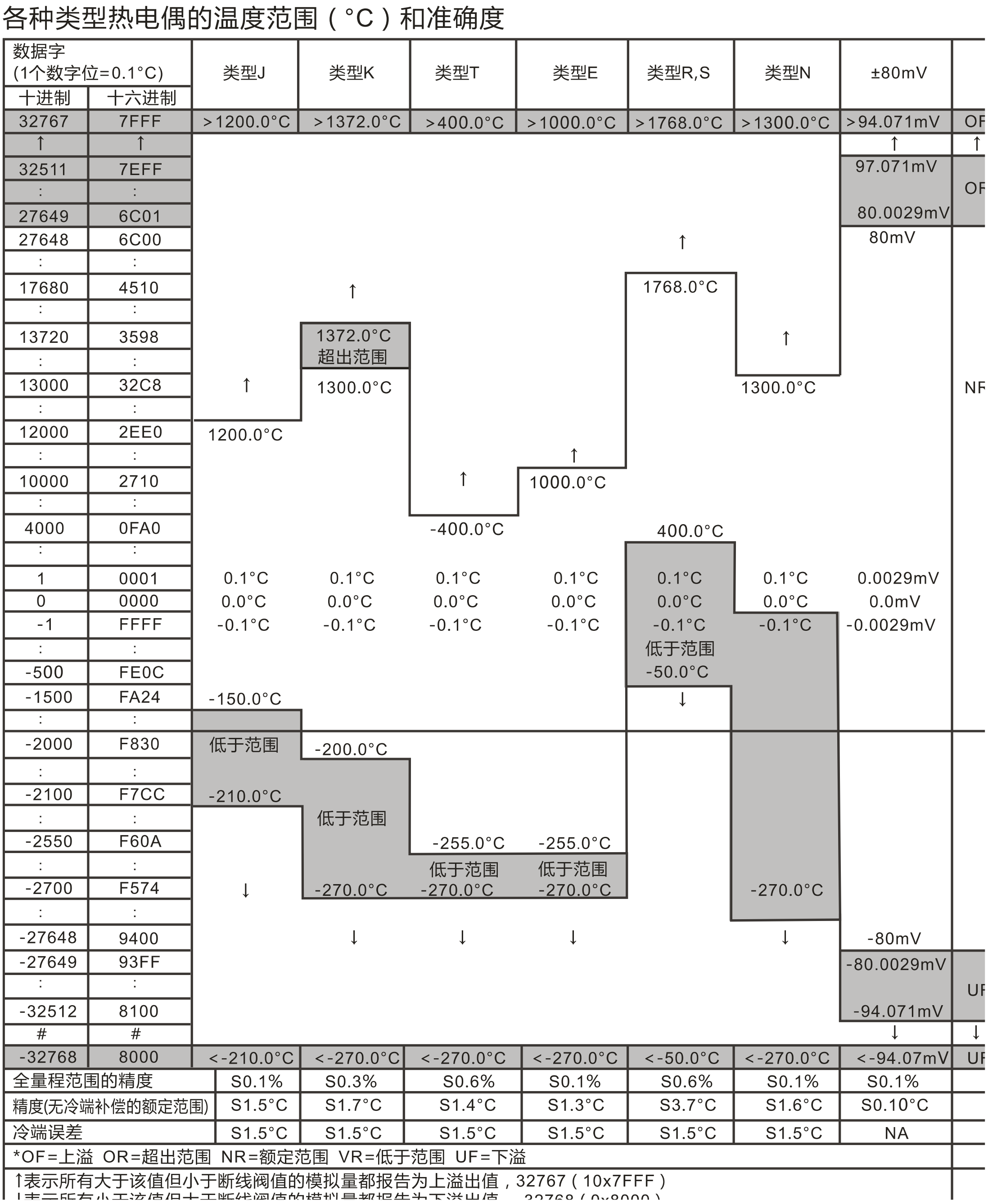


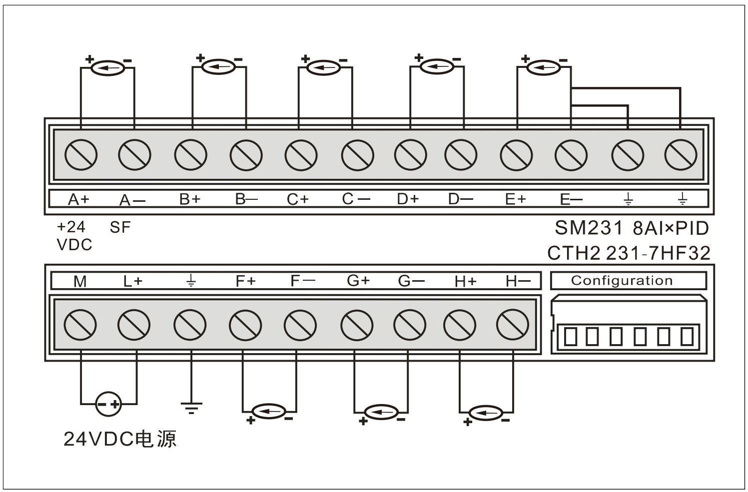
Figure 4-17 Measurement Range of Thermocouple

Specification for 231-7HF32 Current Type PID Module

physical characteristics	
Dimensions (W x H x D), mm	120.5 x 80 x 62
weight	210g
Plug in I/O terminal	no
Power consumption	
Bus (5V DC)	87mA
L+	37mA
power loss	1.8W

Input characteristics	
Enter the number	8. Analog quantity
Input range/input impedance	Current type input
Maximum allowable input voltage at the voltage input terminal	30V DC
Electrical isolation	yes
On site side - Logic	500V AC
On site side - DC 24V	500V AC
DC 24V logic	500V AC
Update time	825ms (all channels)
measuring principle	SIGMA-DELTA
resolving power	15 bit+symbol
temperature	0.1℃/ 0.1℉
Noise suppression on noise frequency	85 dB
Regarding noise frequency	5 /60/400Hz
Common mode voltage	120V AC
Common mode rejection, minimum	120 dB @120V AC
Basic error	0.1% FS (current)
Reproducibility	0.05% FS
Diagnostic program	LED：EXTF，SF
Cable length, maximum	100m - Sensor
Cable loop resistance, maximum	10Ω
PID characteristics	
PID algorithm	PID+FUZZY parameter self-adjusting
Sampling time	1 second
Output minimum pulse width	10ms
PID type	P. PI, PD, PID type
PID output type	Analog or PWM pulse width control
PID output polarity	Bipolar or unipolar

SM231 Current Type PID Module (CTH2 231-7HF32)



DIP switch configuration

Sw1	Sw2	Sw3	Sw4	Sw5	Sw6	range	resolving power
ON	ON	OFF	OFF	OFF	OFF	0~20mA	0.1℃/F
ON	ON	ON	OFF	OFF	OFF	4~20mA (default)	0.1℃/F

PID address and parameter configuration

PID address calculation formula

Address Name	Calculation formula	remarks
PID parameter address	A = （2048+S*256）+16*C	S is the slot number where the module is located (range 0-6) C is channel number 231-7HF, ranging from 0 to 7
PID forward pulse output address	X = （2048+S*256）+12	
PID negative pulse output address	Y = （2048+S*256）+13	

PID parameter output section (module to PLC)

content	address	Numerical Range	actual value
Actual temperature	VW A	-2000~13000	-200~1300 degrees (depending on the Range parameter set by the user, default is 1300)
Status word	VW A+2	-	-
PID analog output	VW A+4	Bipolar: -32000 to+32000 Monopolar: 0 to+32000	-

PID parameter input section (PLC to module)

content	address	Numerical Range	actual value
setting temperature	VW A+128	-2000~13000	-200~1300 degrees
Control bytes	VB A+130	When the position is' 0 '	When the position is "1"
	V (A+130).0	PID not running, no output	PID operation
	V (A+130).1	The integral always works, and the proportional coefficient Kp does not automatically adjust	Integral separation and automatic adjustment of proportional coefficients
	V (A+130).2	PID unipolar output	PID bipolar output with heating and cooling functions
	V (A+130).3	No effect	No effect

	V (A+130).4	Points work	Points do not work
	V (A+130).5	Differential action	Differentiation does not work
	V (A+130).6	Actual temperature value filtering, stronger anti-interference ability	The actual temperature value is not filtered
PID pulse output cycle setting	VW A+132	1-255	1-255 seconds
Kp (proportionality coefficient)	VW A+134	0~9999	0~999.9
Ti (integration time)	VW A+136	0~3600	0~3600 seconds
Td (differential time)	VW A+138	0~3600	0~3600 seconds
Range	VW A+140	0~13000	0~1300 degrees

PID pulse output address calculation

Pulse output channel	Pulse output address	
	Positive pulse output	**Negative pulse output**
0	V X.0	V Y.0
one	V X.1	V Y.1
two	V X.2	V Y.2
three	V X.3	V Y.3
four	V X.4	V Y.4
five	V X.5	V Y.5
six	V X.6	V Y.6
seven	V X.7	V Y.7

Calculation of forward pulse output address: X=(2048+S * 256)+12.

Calculation of negative pulse output address: Y=(2048+S * 256)+13

Application example:

Calculate the address of the last PID loop of 231-7HF on the second expansion module, Kp=1200, Ti=360, Td=50. The pulse output cycle is set to 2s, and the last channel of the second module is set to S=1, C=7	
Address A	A=2048+1*256+16*7=2416
X address	X=2048+1*256+12=2316
Y address	y=2048+1*256+13=2317
The parameter blocks of DB BLOCK are as follows://S=1, C=7 VW2544 500//Set temperature VB2546 16 # 07//Control word (PID enabled, parameter self-adjusting, bipolar output) VW2548 2//Pulse output period VW2550 1200//Kp proportional coefficient VW2552 360//Ti integration time (seconds) VW2554 50//Td differential time (seconds) VW2556 13000//Range of 1300 degrees (default)	
VW2416 0//Actual temperature VW2418 0//Status word VW2420 0 0//PID analog output	

PID模块的参数地址占用CPU的V存储区。

为了保证PID模块能正常使用，编写其他程序块时请一定不要使用您所用的PID模块占用的V存储区。

模块在第0号插槽所占用的地址为：VW2048到VW2298

模块在第1号插槽所占用的地址为：VW2304到VW2554

模块在第2号插槽所占用的地址为：VW2560到VW2810

模块在第3号插槽所占用的地址为：VW2816到VW3066

模块在第4号插槽所占用的地址为：VW3072到VW3322

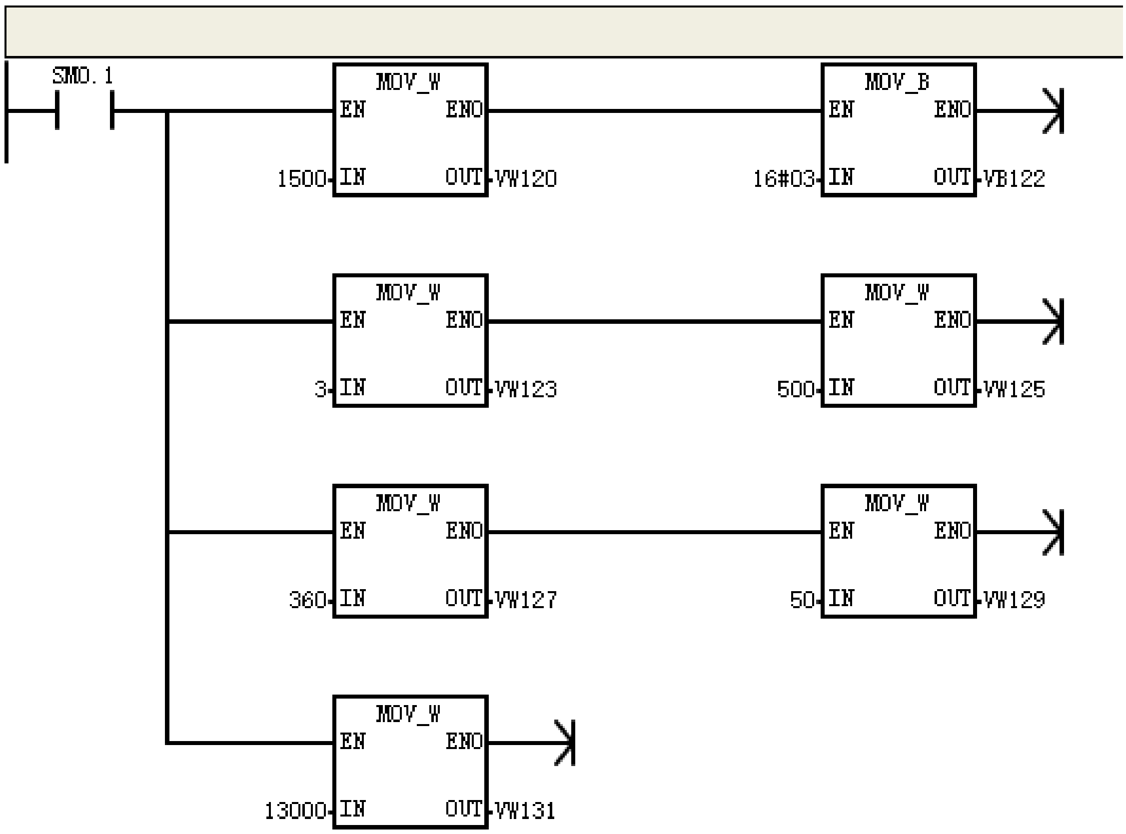
模块在第5号插槽所占用的地址为：VW3328到VW3578

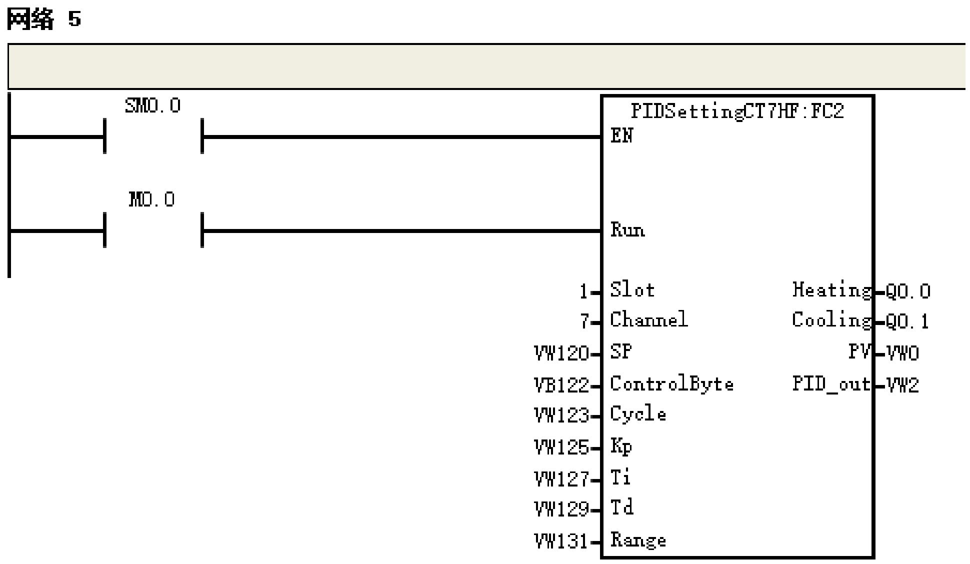
模块在第6号插槽所占用的地址为：VW3584到VW3834

网络 4

设置第一个扩展模块（插槽号为0）的SM231-7HF PID模块第一个PID回路（通道号为0）的参数：
假设该回路调试完成后需要再改变PID参数，调用PID settingCT7HF来设定该回路的参数（需先添加em231-7hf_pid库）
无需计算PID参数地址，只需输入回路所在的插槽号和通道号，使能Run来运行该回路。
本例包含网络1和网络2，网络1初始化PID参数,网络2运行PID
Q0.0为正向脉冲输出，Q0.1为负向脉冲输出，VW0为实际温度，VW2为PID模拟量输出
也可修改PID设定参数使用其他地址，本例对应地址为：
设定温度--VW120；控制字--VB122；脉冲输出周期--VW123；比例系数--VW125
积分时间--VW127；微分时间--VW129；温度传感器量程--VW131

网络 4





7.6.5 Communication module specification

Ordering data

Specification parameters	Order number
SM277A DP slave station, 12M communication rate, optoelectronic isolation	CTH2 277-0AA32
SM277B Profibus DP slave module, 1.5M communication rate, optoelectronic isolation	CTH2 277-0AB32
SM277C CAN slave module, comes with 8DI/6DO, optoelectronic isolation, and can expand up to 7 modules	CTH2 277-0AC32
SM277 PN slave module, 100Mbps communication rate, allows expansion of 8 I/O modules	CTH2 277-0PN32

In the communication module, both SM277A and SM277B use optoelectronic isolation technology, which has high reliability and strong anti-interference ability. Both modules come with terminal resistors, and the bus is wired without the need for dedicated network connectors. Its power supply has reverse connection protection and surge absorption functions, which can be applied to harsh working environments.

However, SM277A needs to be used in conjunction with a CPU, while SM277B can be used as an independent slave with 6 expansion modules, and the slave's I/O scale can reach 128DI/128DO, 48AI/24AQ.

SM277A DP Slave Interface Module Specification

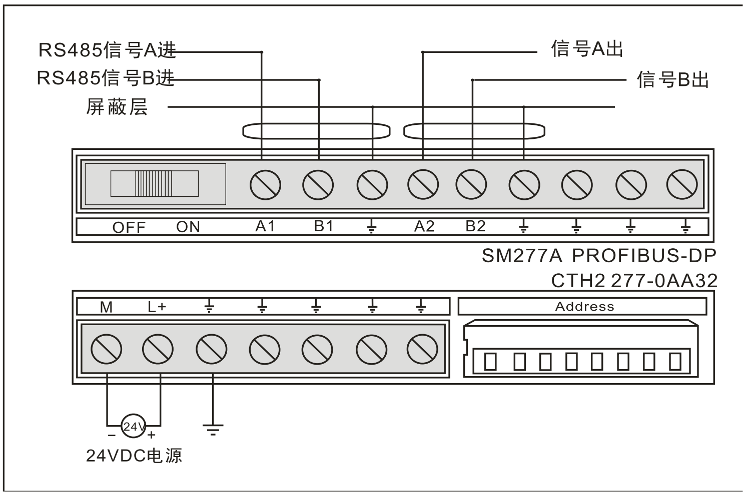
SM277A module parameters

routine	
Order number	CTH2 277-0AA32
Dimensions (width x height x depth)	71.3 × 96 × 62mm
Number of ports	one
Electrical interface	Isolate RS485
Local I/O	0
Maximum number of expansion modules	6 of them

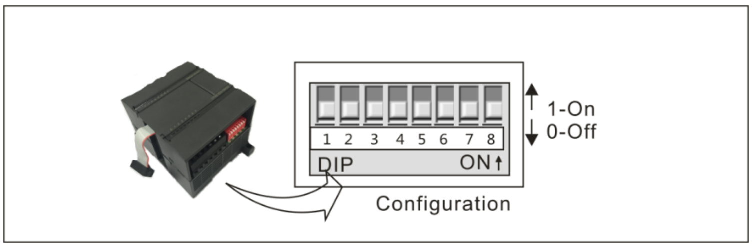
PROFIBUS-DP/MPI baud rate (automatically set)	9.6, 19.2, 45.45, 93.75, 187.5, and 500kbps. 1. 1.5, 3.6, and 12Mbps			
agreement	PROFIBUS-DP slave device and MPI slave device			
LED	**OFF**	**red**	**Red light flashing**	**green**
DX MODE	Not in data exchange mode	--	--	In data exchange mode
DP ERROR	freedom from error	Break away from data exchange mode	Parameterization/configuration error	--
POWER	Without 24V user power supply	--	--	24VDC user power supply is good
CPU FAULT	Good module	Internal module malfunction	--	--
Cable length (maximum)				
Below 93.75kbps baud rate	1200m			
187.5kbps baud rate	1000m			
500kbps baud rate	400m			
1~1.5Mbps baud rate	200m			
3~12Mbps baud rate	100m			
network performance				
Station address setting	0 to 126 (set by dip switch)			
Maximum number of stations per segment	thirty-two			
Maximum number of stations per network	126, up to 125 SM277 sites			
MPI connection	There are a total of 6, 2 reserved (1 for PG and 1 for OP)			
Requirements for 24V DC input power supply				
Voltage range	20.4~28.8V DC (Class 2 or PLC sensor power supply)			
power waste	2.3W			
Maximum current (activation of module communication port)	70mA			
Ripple noise (<10MHz)	<1V peak to peak value (maximum)			

Wiring diagram

SM277A DP Slave Interface Module (CTH2 277-0AA32)



Address switch setting



DIP switch SW1-8, represented by binary numbers, where SW1 is the lowest bit, SW8 is the highest bit (must be turned to "OFF"), SW1-SW7 is turned to "ON" to represent "1", and turned to "OFF" to represent "0". The specific calculation is as follows:

Address=SW1 × 2^0+SW2 × 2^1+SW3 × 2^2+SW4 × 2^3+SW5 × ^{24}+SW6 × ^{25}+SW7 × ^{26}

Quick lookup table for address switch configuration

address	SW1~SW8	address	SW1~SW8	address	SW1~SW8
0	0000 0000	**43**	1101 0100	**86**	0110 1010
1	1000 0000	**44**	0011 0100	**87**	1110 1010
2	0100 0000	**45**	1011 0100	**88**	0001 1010
3	1100 0000	**46**	0111 0100	**89**	1001 1010
4	0010 0000	**47**	1111 0100	**90**	0101 1010
5	1010 0000	**48**	0000 1100	**91**	1101 1010
6	0110 0000	**49**	1000 1100	**92**	0011 1010
7	1110 0000	**50**	0100 1100	**93**	1011 1010
8	0001 0000	**51**	1100 1100	**94**	0111 1010
9	1001 0000	**52**	0010 1100	**95**	1111 1010
10	0101 0000	**53**	1010 1100	**96**	0000 0110
11	1101 0000	**54**	0110 1100	**97**	1000 0110
12	0011 0000	**55**	1110 1100	**98**	0100 0110
13	1011 0000	**56**	0001 1100	**99**	1100 0110

14	0111 0000	**57**	1001 1100	**100**	0010 0110
15	1111 0000	**58**	0101 1100	**101**	1010 0110
16	0000 1000	**59**	1101 1100	**102**	0110 0110
17	1000 1000	**60**	0011 1100	**103**	1110 0110
18	0100 1000	**61**	1011 1100	**104**	0001 0110
19	1100 1000	**62**	0111 1100	**105**	1001 0110
20	0010 1000	**63**	1111 1100	**106**	0101 0110
21	1010 1000	**64**	0000 0010	**107**	1101 0110
22	0110 1000	**65**	1000 0010	**108**	0011 0110
23	1110 1000	**66**	0100 0010	**109**	1011 0110
24	0001 1000	**67**	1100 0010	**110**	0111 0110
25	1001 1000	**68**	0010 0010	**111**	1111 0110
26	0101 1000	**69**	1010 0010	**112**	0000 1110
27	1101 1000	**70**	0110 0010	**113**	1000 1110
28	0011 1000	**71**	1110 0010	**114**	0100 1110
29	1011 1000	**72**	0001 0010	**115**	1100 1110
30	0111 1000	**73**	1001 0010	**116**	0010 1110
31	1111 1000	**74**	0101 0010	**117**	1010 1110
32	0000 0100	**75**	1101 0010	**118**	0110 1110
33	1000 0100	**76**	0011 0010	**119**	1110 1110
34	0100 0100	**77**	1011 0010	**120**	0001 1110
35	1100 0100	**78**	0111 0010	**121**	1001 1110
36	0010 0100	**79**	1111 0010	**122**	0101 1110
37	1010 0100	**80**	0000 1010	**123**	1101 1110
38	0110 0100	**81**	1000 1010	**124**	0011 1110
39	1110 0100	**82**	0100 1010	**125**	1011 1110
40	0001 0100	**83**	1100 1010	**126**	0111 1110
41	1001 0100	**84**	0010 1010		
42	0101 0100	**85**	1010 1010		

Terminal resistance setting

Terminal resistance	**meaning**

ON	Indicates being at the last node of the network
OFF	Indicates that it is not the last node in the network

SM277B DP Slave Module Specification

SM277B Electrical and Environmental Code

Operating environment temperature	0～55℃
Storage environment temperature	-25～70℃
Air mobility	There is flowing air within 50.8mm above and below the device
humidity	5-95% (no condensation)
Anti electrical interference	Pulse width of 50ns, repetition rate of 5kHz, peak voltage of 2000V
Vibration resistance	Frequency 10-57Hz, amplitude 0.1mm, frequency 57-150Hz, acceleration 1.0g, 10 times in each 3D direction
Impact resistance	15g, continuous for 11ms, 3 impacts in 3D direction
High voltage insulation resistance	Apply a voltage of DC 2200V from end to ground, and apply AC1500V from the I/O interface to other ends for 1 minute
Grounding method	3 types of grounding (multi branch single point grounding)
Operating environment	Dust proof and non corrosive environment
fall	100mm, 4 drops, unpackaged
Free fall	1 meter, 5 times, transportation packaging

3 types of grounding methods: Connect the grounding terminal of each device separately to the grounding device

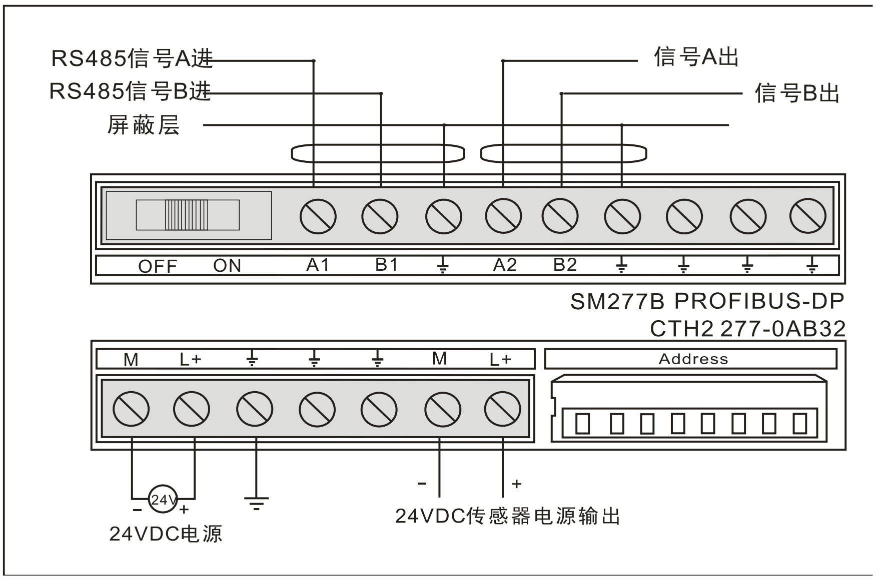
SM277B module specification

routine	
Order number	CTH2 277-0AB32
Dimensions (width x height x depth)	71.3 × 96 × 62mm
Number of ports	one
Electrical interface	RS485 isolation (500V AC)
PROFIBUS-DP baud rate (automatically set)	9.6, 19.2, 45.45, 93.75, 187.5, and 500kbps. 1. 1.5Mbps
agreement	PROFIBUS-DP V0 slave device
Local IO	nothing
Expand IO	
Maximum allowed number of expansion modules	six
Maximum allowed number of intelligent modules	nothing
Maximum I/O (mapping)	256（128 In/128 Out)
LED indicator light	

ON (green)	Lights up after being powered on
SF (red)	If SM277B detects errors during self check (including address setting exceeding range), or if there is a fault in the extended IO module, this light will turn on.
BF (red)	When there is no DP exchange with the main station, the light flashes.
Cable length (maximum)	
Not exceeding 93.75kbps baud rate	1200m
187.5kbps baud rate	1000m
500kbps baud rate	400m
1 to 1.5Mbps	200m
network performance	
Station address setting	0~126 (set by dip switch)
Maximum number of stations per segment	thirty-two
Maximum number of stations per network	126, up to 125 SM277B stations
Requirements for 24V DC input power supply	
Voltage range	20.4~28.8V DC (sensor power supply for Class 2 or PLC)
quarantine	nothing
power waste	8W
Output current provided for the sensor (24V DC)	
Voltage range	20.4~28.8V DC (Class 2 or PLC sensor power supply)
24V output maximum current	400mA
Output current provided for the expansion module (5V DC)	
5V output maximum current	660mA

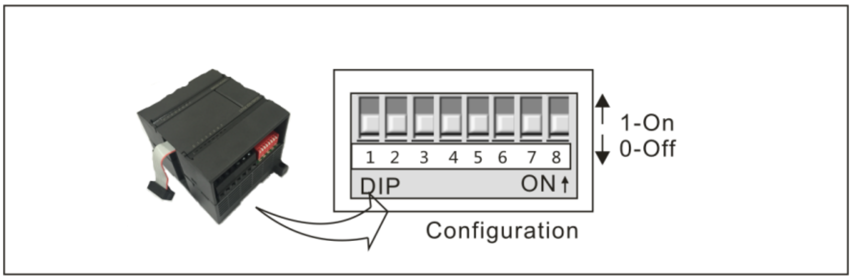
Wiring diagram

SM277B PROFIBUS DP Slave Module (CTH2 277-0AB32)



Set PROFIBUS address

Set the PROFIBUS address through the DIP switch. After changing the PROFIBUS address, the SM277B needs to be powered off and reconnected.



DIP switch SW1-8, represented by binary numbers, where SW1 is the lowest bit and SW8 is the highest bit (must be turned to "OFF"),

Turning SW1-SW7 to "ON" indicates "1", and turning it to "OFF" indicates "0". The specific calculation is as follows:

Address=SW1 × 2^0+SW2 × 2^1+SW3 × 2^2+SW4 × 2^3+SW5 × 2^4+SW6 × 2^5+SW7 × 2^6

Quick lookup table for address switch configuration

address	SW1~SW8	address	SW1~SW8	address	SW1~SW8
0	0000 0000	**43**	1101 0100	**86**	0110 1010
1	1000 0000	**44**	0011 0100	**87**	1110 1010
2	0100 0000	**45**	1011 0100	**88**	0001 1010
3	1100 0000	**46**	0111 0100	**89**	1001 1010
4	0010 0000	**47**	1111 0100	**90**	0101 1010
5	1010 0000	**48**	0000 1100	**91**	1101 1010
6	0110 0000	**49**	1000 1100	**92**	0011 1010
7	1110 0000	**50**	0100 1100	**93**	1011 1010
8	0001 0000	**51**	1100 1100	**94**	0111 1010

9	1001 0000	**52**	0010 1100	**95**	1111 1010
10	0101 0000	**53**	1010 1100	**96**	0000 0110
11	1101 0000	**54**	0110 1100	**97**	1000 0110
12	0011 0000	**55**	1110 1100	**98**	0100 0110
13	1011 0000	**56**	0001 1100	**99**	1100 0110
14	0111 0000	**57**	1001 1100	**100**	0010 0110
15	1111 0000	**58**	0101 1100	**101**	1010 0110
16	0000 1000	**59**	1101 1100	**102**	0110 0110
17	1000 1000	**60**	0011 1100	**103**	1110 0110
18	0100 1000	**61**	1011 1100	**104**	0001 0110
19	1100 1000	**62**	0111 1100	**105**	1001 0110
20	0010 1000	**63**	1111 1100	**106**	0101 0110
21	1010 1000	**64**	0000 0010	**107**	1101 0110
22	0110 1000	**65**	1000 0010	**108**	0011 0110
23	1110 1000	**66**	0100 0010	**109**	1011 0110
24	0001 1000	**67**	1100 0010	**110**	0111 0110
25	1001 1000	**68**	0010 0010	**111**	1111 0110
26	0101 1000	**69**	1010 0010	**112**	0000 1110
27	1101 1000	**70**	0110 0010	**113**	1000 1110
28	0011 1000	**71**	1110 0010	**114**	0100 1110
29	1011 1000	**72**	0001 0010	**115**	1100 1110
30	0111 1000	**73**	1001 0010	**116**	0010 1110
31	1111 1000	**74**	0101 0010	**117**	1010 1110
32	0000 0100	**75**	1101 0010	**118**	0110 1110
33	1000 0100	**76**	0011 0010	**119**	1110 1110
34	0100 0100	**77**	1011 0010	**120**	0001 1110
35	1100 0100	**78**	0111 0010	**121**	1001 1110
36	0010 0100	**79**	1111 0010	**122**	0101 1110
37	1010 0100	**80**	0000 1010	**123**	1101 1110
38	0110 0100	**81**	1000 1010	**124**	0011 1110
39	1110 0100	**82**	0100 1010	**125**	1011 1110
40	0001 0100	**83**	1100 1010	**126**	0111 1110

41	1001 0100	**84**	0010 1010
42	0101 0100	**85**	1010 1010

Terminal resistance

Terminal resistance	meaning
ON	Indicates being at the last node of the network
OFF	Indicates that it is not the last node in the network

SM277C CANopen Slave Module Specification

The CANopen slave interface module adopts optoelectronic isolation, which has strong reliability and anti-interference ability. It comes with two RJ45 CANopen communication ports, which form a distributed control system with the CAN master station system. The module can be extended to 7 expansion modules (does not support PID intelligent modules), and comes with 8DI/6DO transistor input and output and a 12 pin dip switch, making it easy to select station address, baud rate, and terminal resistance.

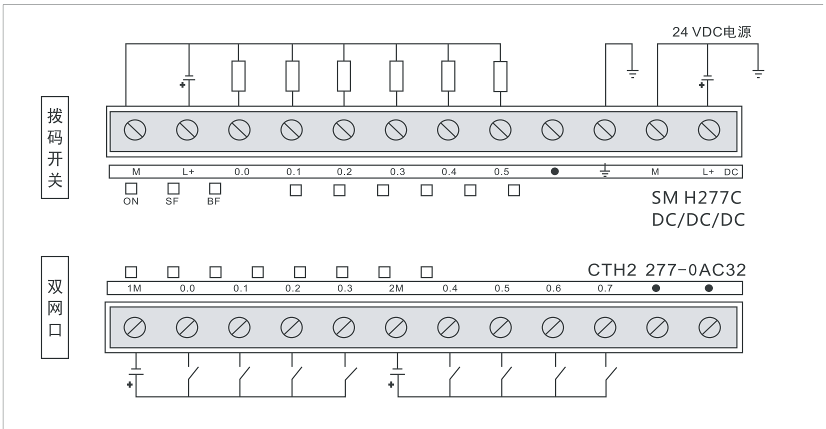
SM277C module specification

<table>
<tr><th colspan="3">routine</th></tr>
<tr><td colspan="2">Order number</td><td>CTH2 277-0AC32</td></tr>
<tr><td colspan="2">Dimensions (width x height x depth)</td><td>71.3 × 96 × 62 mm</td></tr>
<tr><th colspan="3">Digital input</th></tr>
<tr><td colspan="2">Number of ports</td><td>2 RJ45 CANopen communication interfaces</td></tr>
<tr><td colspan="2">Rated input voltage</td><td>24V DC</td></tr>
<tr><td colspan="2">type</td><td>Source/leakage type (IEC Class 1 source point)</td></tr>
<tr><td colspan="2">input voltage</td><td>Typical value 24V DC</td></tr>
<tr><td colspan="2">Maximum allowable continuous voltage</td><td>30V DC</td></tr>
<tr><td colspan="2">Surge voltage</td><td>35V DC, lasting for 0.5s</td></tr>
<tr><td colspan="2">Input delay</td><td>6.4ms (minimum 6.3ms)</td></tr>
<tr><td colspan="2">Isolation (from site to logic circuit)</td><td>yes</td></tr>
<tr><td colspan="2">Simultaneously connect the input points</td><td>8 (all at 55 ℃)</td></tr>
<tr><td rowspan="2">Maximum length of cable</td><td>shield</td><td>500m standard input</td></tr>
<tr><td>Unshielded</td><td>300m standard input</td></tr>
<tr><th colspan="3">Digital output</th></tr>
<tr><td colspan="2">type</td><td>Solid state MOSFET (source type)</td></tr>
<tr><td colspan="2">Rated voltage</td><td>24V DC</td></tr>
<tr><td colspan="2">Output voltage range</td><td>20.4~28.8V DC</td></tr>
<tr><td colspan="2">Output current (signal 1, maximum value)</td><td>0.5A</td></tr>
</table>

Output the number of groups		one
Output points for each group		six
Simultaneously connect the output points		six
Maximum current per group		4.5A
Surge current		8A, lasting for 100ms
quarantine		Optocoupler isolation, 500V AC, duration within 1 minute
Output delay	Close Open	Maximum 15 μ s
	Open close	Maximum 130 μ s
Maximum cable length	shield	500m standard output
	Unshielded	150m standard output

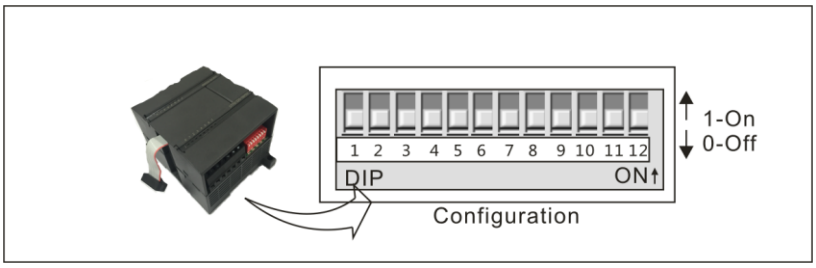
Wiring diagram

SM277C CANopen Slave Module (CTH2 277-0AC32)



CTH2 277-0AC32接线图

Address switch setting



DIP switch SW1-12 configuration

SM277-0AC32 dip switch		
Dial number		**usage method**
SW1	Contact address	Binary number representation: 1 is low, 7 is high. Note: 0 is the global address, and it is prohibited to set it to 0 address when using it.
SW2		
SW3		
SW4		

SW5								
SW6								
SW7								
SW8	Baud rate	Binary number representation: 8 is low, 10 is high						
SW9		Dial code	one hundred and eleven	one hundred and ten	one hundred and one	one hundred	011	010	001
SW10		Baud rate kbps	one thousand	eight hundred	five hundred	two hundred and fifty	one hundred and twenty-five	fifty	twenty
		Maximum length (m)	twenty-five	fifty	one hundred	two hundred and fifty	five hundred	one thousand	two thousand and five hundred
SW11	Terminal resistance	The device needs to be dialed to the "ON" position at the network port, and to the "OFF" position for other devices						
SW12	retain	—						

Hardware diagnosis

The CANopen master-slave network can be diagnosed by the communication indicator status of SM 277C.

LED light status of SM277C

After connecting the power to SM 277C, the POWER LED indicator marked as "ON" (green) will light up. If the BF and SF LEDs remain off, it indicates that the SM 277C is operating normally. If the BF or SF LED lights up or flashes, it indicates that there is an error in the hardware configuration or wiring. If the main system runs without any faults, the reasons for the errors can refer to the following table:

SM 277C Status LED Light Function Description

LED lights	bright	Extinguish	twinkle	remarks
ON	There is power supply	No power supply	---	Power indicator
SF	There is a malfunction in the extended IO module	No faults found in the extended IO module	---	system failure
BF	CAN network not detected	CAN network detected	Inconsistent configuration	Bus malfunction

SM277 PN Slave Module Specification

Tips

SM227 PN can be used with third-party CPU master stations. For specific usage methods and examples, please refer to the "CTH200 Series Profinet Slave Module User Manual". The manual can be downloaded from: https://www.co-trust.com

The environmental standards and specifications followed by the SM277-PN module

Environmental conditions - transportation and storage	
temperature	-40℃~+70℃

relative humidity	10%~95%, non condensing
Environmental conditions - work	
temperature Horizontal installation position Vertical installation position	0℃~60℃ 0℃~40℃
relative humidity	10%~95%, non condensing
Electromagnetic Compatibility - Immunity	
electrostatic discharge IEC61000-4-2	Contact discharge: ± 4kV Air discharge: ± 8kV
Electric fast transient pulse train IEC61000-4-4	Power cord: 2kV, 5kHz Signal line: 2kV, 5kHz (I/O coupling clip) 1kV, 5kHz (communication coupling clip)
Surge IEC61000-4-5	Power cord: 2kV (asymmetrical), 1kV (symmetrical)
Radio frequency electromagnetic field radiation IEC61000-4-3	80MHz~1GHz，10V/m，80%AM（1kHz）
Radio frequency field induced conducted interference IEC61000-4-6	0.15MHz~80MHz，10V/m，80%AM（1kHz）
Short term interruption and voltage variation of DC power input port IEC61000-4-29	Short interruption: 10ms
High temperature operation IEC60068-2 Low temperature operation IEC60068-2	65 ℃, residence time: 24h, temperature change rate: ≤ 1 ℃/min, room temperature recovery time: ≥ 1h -20 ℃, residence time: 24h, temperature change rate: ≤ 1 ℃/min, room temperature recovery time: ≥ 1h
High temperature start-up IEC60068-2 Low temperature start-up IEC60068-2	65 ℃, 2 hours, after the temperature stabilizes, do not turn on continuously for 3 times -20 ℃, 2 hours, after the temperature stabilizes, do not turn on continuously for 3 times
High and low temperature cycle operation IEC60068-2	-10 ℃~60 ℃, residence time 3h, temperature rate 1 ℃/min, 5 cycles, room temperature recovery time 1h
High temperature storage IEC60068-2 Low temperature storage IEC60068-2	75 ℃, storage time: 72h, temperature change rate: ≤ 1 ℃/min, room temperature recovery time: ≥ 1h -40 ℃, storage time: 72h, temperature change rate: ≤ 1 ℃/min, room temperature recovery time: ≥ 1h
Thermal shock IEC60068-2	-40 ℃~75 ℃, residence time 3h, temperature change time 2min, 3 cycles
Constant high humidity IEC60068-2	Storage temperature: 40 ℃, storage time: 48h, storage humidity: 93%
Alternating Damp Heat IEC60068-2	25 ℃~55 ℃, humidity: 95%, single cycle: 24h, 2 cycles
Sinusoidal vibration (bare metal) IEC60068-2	5~150Hz，0.^{05G2}/Hz 150Hz~500Hz, -2dB/oct, 1h/axis, X, Y, Z total 3 axes
Mechanical impact (2 machines) IEC60068-2	15G, 11ms pulse, 3 times/direction
drop	1m, 10 times, transportation packaging

General characteristics of SM277 PN module

physical characteristics	
Dimensions (W × H × D)	71×80×62 mm

Power loss	
Rated input voltage	24V DC
Input voltage range	20.4V~28.8V DC
Input current	0.8A
Bus power supply voltage	+5V DC
Bus power supply current	1.1A
quarantine	Isolation between external power supply and system power supply
Power protection	The power supply end provides reverse connection protection and surge absorption function
Characteristics of LED indicator lights	
24V power indicator light (green)	ON=24V DC power supply is normal
	OFF=No 24V DC power supply
SF indicator light (red)	ON=Expansion I/O bus failure or PROFINET module failure
	OFF=No error
BF indicator light (red)	ON=PROFINET bus communication failure (not connected to switch, network not detected)
	Blinking=inconsistent configuration
	OFF=No error
MT indicator light (yellow) (Maintenance indicator light)	ON=Lost synchronization
	OFF=No error
P1R/P2R port indicator light (green)	ON=There is a connection to the switch/PN master station
	OFF=No connection to switch/PN master station
P1R/P2R port indicator light (yellow)	ON=Data transmission and reception to switch/PN master station
	OFF=No data transmission or reception to the switch/PN master station

Functional characteristics of SMS277 PN module

Expand I/O capability	
Maximum number of modules supported per slave station	8 (digital module, analog module, temperature module, and PID control module, excluding other types of modules)
Protocol type	PLC custom 4MHz bus protocol
Maximum IO configuration per slave station	The maximum configuration of analog IO can reach 64AI/32AQ The maximum configuration of digital IO can reach 256DI/256DQ
topological structure	Supports star, tree, line, and ring shapes
PROFINET communication port	
communication interface	1 dual RJ45 port
Data transmission rate	The Ethernet transmission rate is 10Mbps

	PROFINET transmission rate is 100Mbps, full duplex
Supported Ethernet services	Ping, ARP, Network Diagnostics (SNMP)/MIB-2, LLDP
Sending cycle	250us~4ms
Third party PROFINET main station	Supports S7-300/400, SMART CPU, S7-1200/1500
The longest communication distance between stations	100m（100BASE-TX）
topological structure	Support star, tree, line, and ring topology structures
quarantine	Communication port isolation
Hardware configuration function	
Files to be imported	PROFINET GSD file in XML format
PN slave station	Add CTH2 277-0PN32 module to support expansion of 8 slots
	Expansion modules can add digital, analog, temperature, and PID control modules, while other modules cannot be added

7.6.6 Weighing module specification

M226ES can be used with the CTH200 series weighing module. The following is the specification data of the CTH200 weighing module. For specific usage methods and examples, please refer to the "CTH200 Series Weighing Module User Manual". Download link: https://www.co-trust.com

Weighing module ordering data

Specification parameters	Order number
SM231 weighing module, 1-channel sensor signal input, sampling frequency 50Hz, 0.01% module accuracy, 6V DC 150MA excitation power output per channel, isolated 16 bit accuracy	CTH2 231-7WA32

Performance parameters of SM231 weighing module

Order number	CTH2 231-7WA32
LED indicator light	Power indicator light, ON=24V DC power supply normal, OFF=no 24V DC power supply SF indicator light, flashing=out of range or disconnected, OFF=no error
Power loss	
+5V DC consumption current	＜140mA
L+current consumption	＜100mA
L+coil voltage range	20.4~28.8V DC
power loss	5W
Input characteristics	
Input type	Strain gauges, using 4-wire or 6-wire wiring technology
Input Range	0~1mV/V 0~2mV/V 0~4mV/V
Input points	A weighing sensor along the way

Resistance of weighing sensor	When there is no explosion-proof interface: 40 Ω<R<4010 Ω When there is an explosion-proof interface: 87 Ω<R<4010 Ω
Isolation characteristics	
From site to logic	500V AC
On site to 24V DC	500V AC
24V to Logic	500V AC
Common mode rejection	> 120dB@120V AC
Input resolution	
zero temperature	≤±0.1μV/K
measuring principle	Σ-Δ
The length of the wire to the sensor	Maximum 500 meters
Noise suppression	85db@50Hz /60Hz
Data Word Format	Voltage: 0~65535
resolving power	16 bits
Basic error	0.05%Fs
Linear error	0.01%Fs

Terminal connection

The weighing module is a high-precision measurement module that must reliably measure low signal levels (approximately 1.5 μ V). Therefore, proper assembly and cable connections are crucial for achieving fault free operation.

When connecting the weighing sensor, the following rules should be followed:

1) If more than one sensor is to be connected (weighing sensors must be connected in parallel), a junction box must be used. If the distance from the weighing sensor to the module or to the junction box is greater than the length of the available cable for the weighing sensor, a dedicated extension box should be used.

2) The shielding of the cable should always be connected to the cable gland of the junction box or extension box.

3) Twisted twisted pair cables should be used on the designated lines, and they should be shielded:

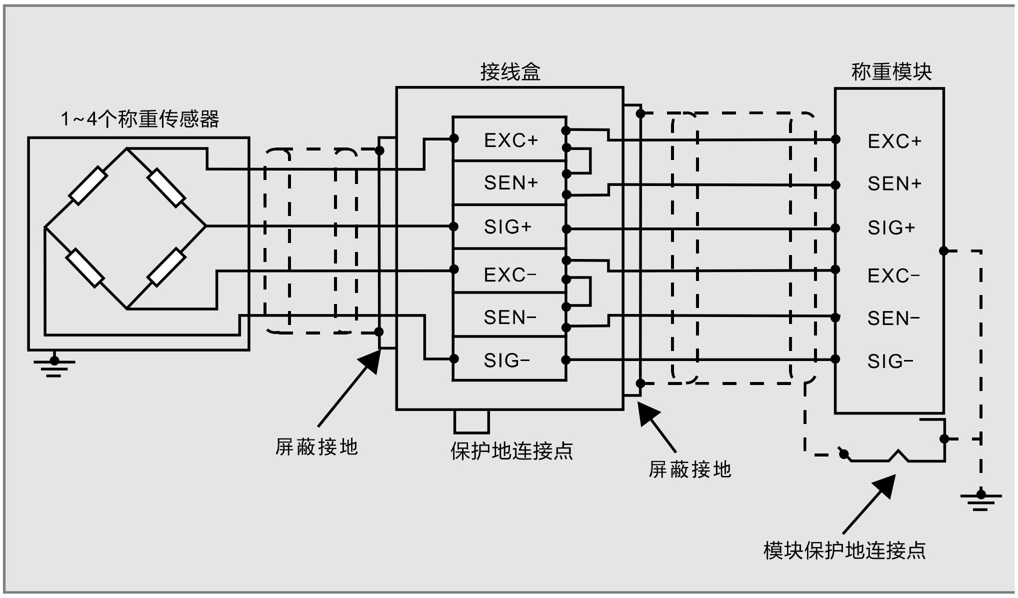
Sensor line SEN+/SEN -.

Measure the voltage line SIG+/SIG -.

Power supply voltage line EXC+/EXC -.

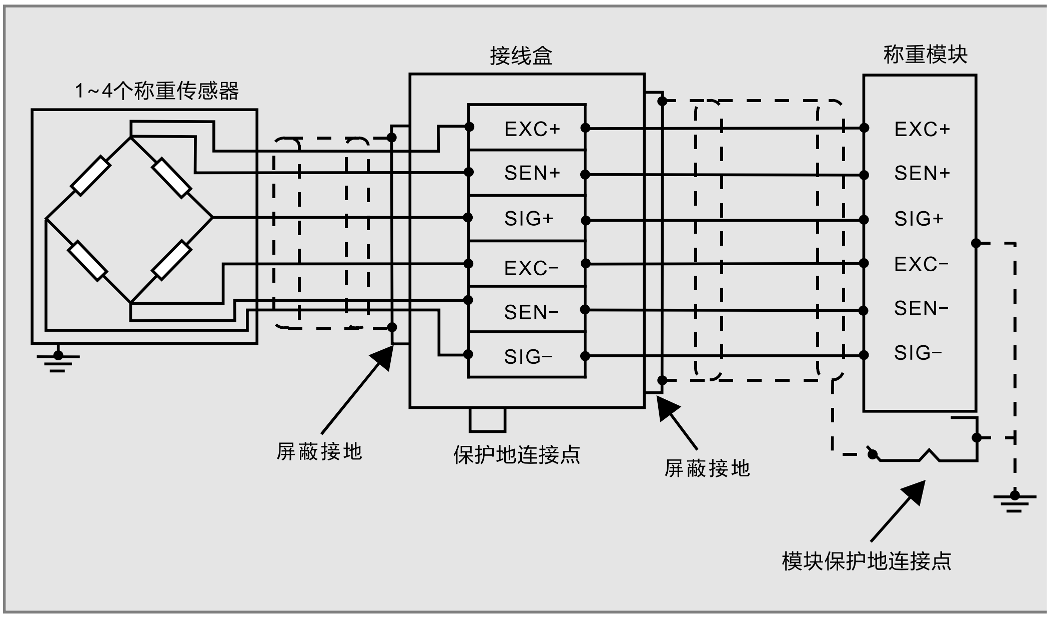
4) The shielded wire must be connected to the shielded joint clamp.

1. Four wire connection method for weighing sensors



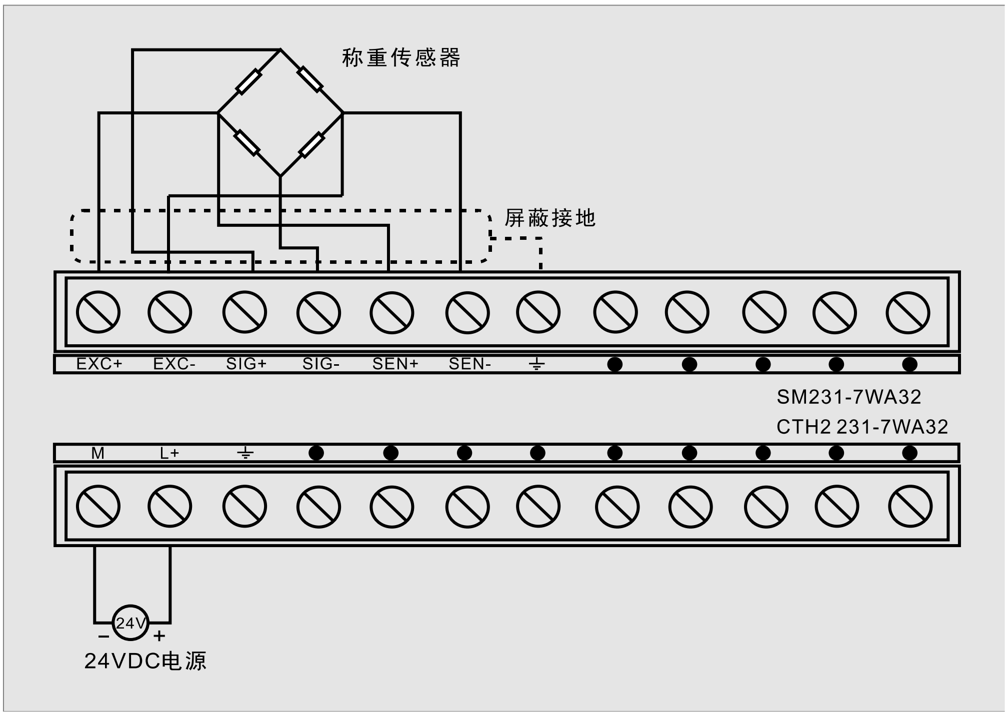
When connecting a four wire sensor, be sure to short-circuit EXC+to SENS+and EXC - to SENS - in the junction box, otherwise the weighing module may not function properly.

2. Six wire connection method for weighing sensors



Wiring diagram

SM231 Weighing Module (CTH2 231-7WA32)



7.6.7 Specification for motion control module

M226ES supports the use of CTH200 series operation control modules. The following are the specifications of CTH200 series operation control modules. For specific usage methods and examples, please refer to the "CTH200 Series Motion Control Module User Manual". Download link: https://www.co-trust.com

Ordering data for motion control module

Specification parameters	Order number
SM253 motion control module, 2-channel single-phase or AB phase HSC input, 200kHz, 2-axis PTO/PWM output, 200kHz, COTRUST motion control library	CTH2 253-1BH32

Digital input characteristics of motion control module

routine	24V DC input
Local integrated digital input points	Input at 8 o'clock
type	Leakage type/source type (IEC type 1/leakage type)
Rated voltage	At a typical value of 5mA, it is 24V DC
Maximum allowable continuous voltage	30V DC
Surge voltage	35V DC for 0.5s
Logic 1 (minimum)	15.6V DC at 2.72mA (I0.0, I0.1, I0.2, I0.4, I0.5, I0.6) 12.8V DC (I0.3, I0.7) at 2.55mA
Logic 0 (maximum)	At 2.69mA, it is 15.4V DC (I0.0, I0.1, I0.2, I0.4, I0.5, I0.6) At 2.51mA, it is 12.6V DC (I0.3, I0.7)
Input delay	<1.1us (I0.0, I0.1, I0.2, I0.4, I0.5, I0.6) <1ms (I0.3, I0.7)
Maximum allowable leakage current for the connection of two wires (Bero) adjacent to	1mA

the sensor		
Isolation (from site to logic circuit) Optical (current) isolation group		yes 500V AC, 1 minute, refer to wiring diagram
High speed input rate High speed counter logic 1=16-26V DC		200kHz (single/dual phase) (I0.0, I0.4) 200kHz (A/B phase) (I0.0 and I0.1, I0.4 and I0.5) (Input waveform duty cycle 40%~60%)
Simultaneously connect the input		All of them
Maximum cable length	shield	500m standard input, 50m HSC input (for HSC input, please use shielded twisted pair and ground the shielding layer)
	Unshielded	300 meter standard input
High speed input, maximum directional frequency supported by the counter		50kHz

Digital output characteristics of motion control module

routine	Transistor output
Local integrated digital output points	8 outputs
type	Solid state MOSFET (drain type)
Rated voltage	24V DC
Output voltage range	5~28.8V DC
Surge current (maximum)	8A lasts for 100ms
Logic 1 (maximum)	0.5V
Logic 0 (minimum)	VCC-0.5V
Rated current per point (maximum)	0.5A
Rated current of each common terminal (maximum)	2.0A
Leakage current (maximum)	10μA
Lighting load (maximum)	3.5W
Inductive clamp voltage	L+minus 48V DC, 1W power consumption
Connect the resistor (contact)	Typical value of 0.3 Ω (maximum value of 0.6 Ω)
quarantine Optics (Electrification, Field to Logic Circuit) Logic circuit to contact Resistance (from logic circuit to contact) Isolation group	500V AC, 1 minute --- --- Refer to the wiring diagram
Delay (maximum) Disconnect and connect Connect to disconnect	0.2 μ s (Q0.0, Q0.1, Q0.2, Q0.3), 50 μ s (Q0.4, Q0.5, Q0.6, Q0.7) 0.2 μ s (Q0.0, Q0.1, Q0.2, Q0.3), 130 μ s (Q0.4, Q0.5, Q0.6, Q0.7)
Pulse frequency (maximum)	200kHz (Q0.0, Q0.2)
Mechanical life cycle	/

Contact life		/
Simultaneously open output		At 55 ° C, output all
Two outputs in parallel		Yes, only output when in the same group
Maximum cable length	shield	500 meters
	Unshielded	150 meters

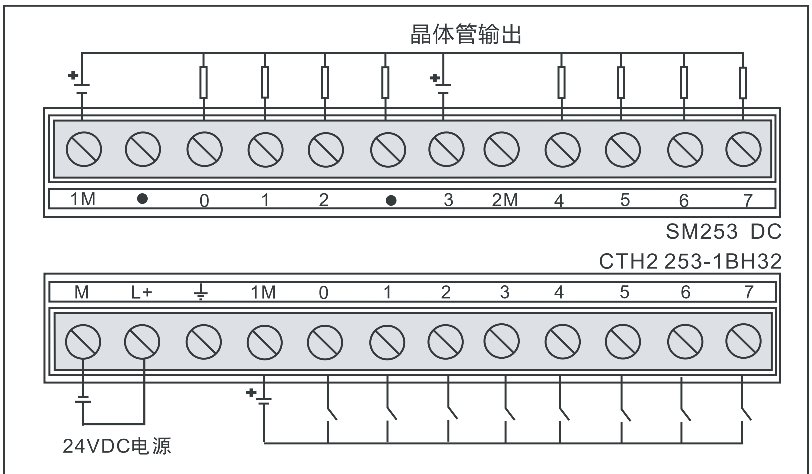
be careful

Due to communication transmission, there is a certain delay in the execution of DO output instructions: if a single SM253 motion control module is hung behind the CPU, the delay from enabling to executing the output instructions is about 780us due to transmission delay. If up to 5 SM253 motion control modules are mounted behind the CPU, the delay from enabling to executing output instructions is approximately 930 microseconds.

The SM253 module is designed for motion control applications and comes with two independent high-speed counters, MC253_SC0 and MC253_SC1 (both single and dual phase are 200kHz). Two independent 200kHz high-speed pulse outputs, supporting commands such as MC253-PTP/SPEED-CTL/PWM from the motion control library motion_ctrl_module_lib.

Wiring diagram

SM253 Motion Control Module (CTH2 253-1BH32)



CTH2 253-1BH接线图

I/O Function Definition

I0.0	I0.1	I0.2	I0.3	I0.4	I0.5	I0.6	I0.7
MC253_SSC0 pulse input	MC253_SSC0 external direction signal	MC253_SSC0 reset signal	Motion axis 0 emergency stop signal	MC253_SC1 pulse input	MC253_SC1 external direction signal	MC253_SC1 reset signal	Motion axis 1 emergency stop signal
Q0.0	**Q0.1**	**Q0.2**	**Q0.3**	**Q0.4**	**Q0.5**	**Q0.6**	**Q0.7**
Axis 0 pulse output	Axis 0 direction signal	Axis 1 pulse output	Axis 1 direction signal	Ordinary output port			

7.6.8 BD expansion board specifications and installation methods

M226ES can be used with CTH200 series BD expansion board. The ordering information for BD expansion board is as follows:

BD expansion board ordering data

Specification parameters	Order number
EBH-AMS-03 Analog Input/Output Expansion Board, 2-point * 12 bit Precision Voltage Input, 1-point * 12 bit Precision Voltage/Current Output	CTH2 AMS-03S1-EB
EBH-AMS-06 Analog Input/Output Expansion Board, 4-point * 12 bit Precision Voltage Input, 2-point * 12 bit Precision Voltage Output	CTH2 AMS-06S1-EB
EBH-AMS-06 Analog Input/Output Expansion Board, 4-point * 12 bit Precision Voltage Input, 2-point * 12 bit Precision Current Output	CTH2 AMS-06S2-EB
CTH2 CAN main station communication expansion board, 1Mbps, optoelectronic isolation, configuration mode supports CAN hardware configuration block and EDS file	CTH2 CAN-01S2-EB
One way RS485 communication expansion board isolation type	CTH2 RS485-01S1-EB

Specification for Analog Expansion Board

Environmental condition specifications

project	parameter
Operating environment temperature	0～55℃
Storage environment temperature	-25～70℃
Air mobility	There is flowing air within 50.8mm above and below the device
humidity	5-95% (no condensation)
Anti electrical interference	Please refer to GB17626.2, GB17626.3, GB17626.4, GB17626.5, GB17626.6 for specific information
Vibration resistance	Frequency 10-57Hz, amplitude 0.1mm, frequency 57-150Hz, acceleration 1.0g, 10 times in each 3D direction
Impact resistance	15g, continuous for 11ms, 3 impacts in 3D direction
High voltage insulation resistance	Apply a DC 2000V voltage from end to ground and an AC 1500V voltage from the I/O interface to other ends for 1 minute
Grounding method	3 types of grounding
Operating environment	Dust proof and non corrosive environment
fall	100mm, 4 drops, unpackaged
Free fall	1 meter, 5 times, transportation packaging

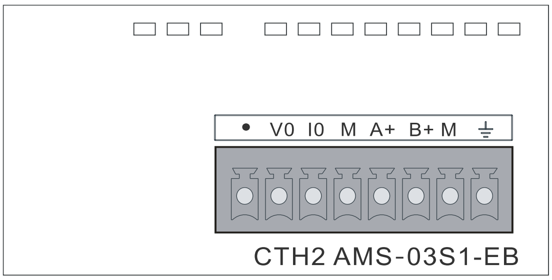
Characteristics of analog expansion board

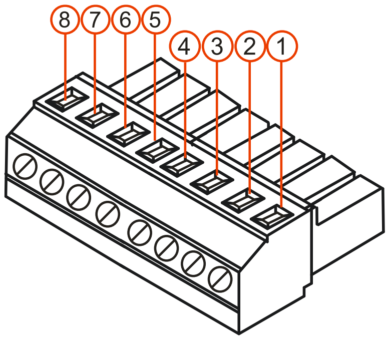
project		standard		
		EBH-AMS-03S1	EBH-AMS-06S1	EBH-AMS-06S2
Basic characteristics	Physical dimensions (W × H × D)	24×14.4×68.2mm		
	Bus power supply voltage	+5V DC		
	Bus power supply current	60mA	60mA	60~250mA
LED indicator	Power indicator light	The power indicator is good. ON=5V DC power supply is normal, OFF=no 5V DC power supply.		
	SF indicator light	Calibration status indicator light, flashing=calibration failed		Calibration status indicator light

	Input points	two	four	
	Input type	Single ended voltage input		
	Input Range	±10V		
	Full range data word format	-32000~+32000		
	DC input impedance	>100kΩ		
	Maximum input voltage	30V DC		
	resolving power	11 bits plus 1 sign bit		
	quarantine	nothing		
	accuracy			
	Worst case scenario (0-55 °)	± 2.5% full scale		
	Typical situation (25 ˚)	± 1.0% full scale		
	Repeatability	± 0.05% full scale		
	Analog to digital conversion time	125ms		
	Conversion principle	Σ-Δ		
	Step response	Maximum 250ms		
	noise suppression	Typical values: -20dB@50Hz		
Analog output characteristics	Output points	one	two	
	Signal range	Voltage: 0~10V	Voltage: 0~10V	--
		Current: 0~20mA	--	Current: 0~20mA
	Full range data word format	0~+32000		
	resolving power	12 digits		
	LSB value	Voltage: 2.50mV	Voltage: 2.50mV	--
		Current: 5.00 μ A	--	Current: 5.00 μ A
	quarantine	nothing		
	accuracy			
	Worst case scenario (0-55 °)	Voltage: ± 2% of full scale	Voltage: ± 2% of full scale	--
		Current: ± 3% of full scale	--	Current: ± 3% of full scale
	Typical situation (25 ˚)	Voltage: ± 1% of full scale	Voltage: ± 1% of full scale	--
		Current: ± 1% of full scale	--	Current: ± 1% of full scale
	Stable time • With resistive load • With capacitive load • With inductive load	0.1ms		0.2 ms 3.3 ms 0.5 ms (1mH)
	Maximum output driver	≥5000Ω		≤500Ω

Terminal definition

1. The structure diagram and terminal diagram of the AMS-03 analog input/output expansion board are shown below. The definitions of each terminal position are listed in the subsequent table. When installing and connecting the expansion board, please strictly follow the terminal position definitions for connection.

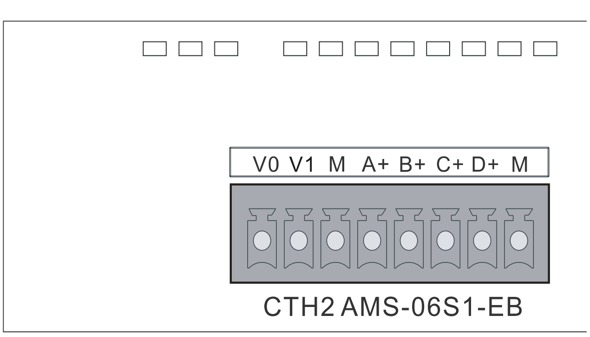


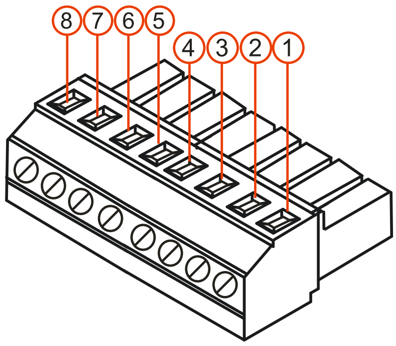


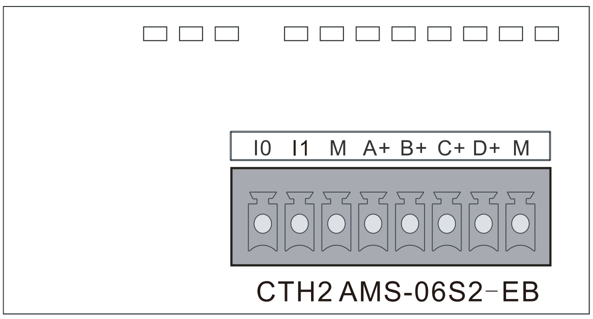
AMS-03 Terminal Position Definition

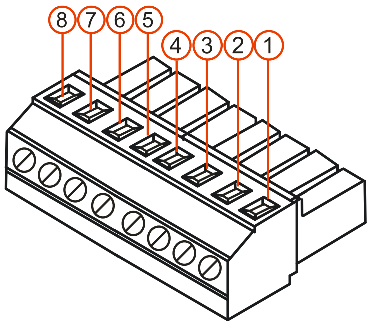
Terminal position	Definition of Signal	Terminal position	Definition of Signal
one	Hanging in the air	five	Analog input A+
two	Voltage output VO	six	Analog input B+
three	Current output IO	seven	Ground GND-M
four	Ground GND-M	eight	earth

2. The structure diagram and terminal diagram of the AMS-06 analog input/output expansion board are shown below. The definitions of each terminal position are listed in the subsequent table. When installing and connecting the expansion board, please strictly follow the terminal position definitions for connection.









AMS-06 Terminal Position Definition

Terminal position	Definition of Signal	Terminal position	Definition of Signal
one	Output V0/I0	five	Analog input B+
two	Output V1/I1	six	Simulate input C+

three	Ground GND-M	seven	Analog input D+
four	Analog input A+	eight	Ground GND-M

<**Note**>A+, B+, C+, D+are voltage input terminals that can input positive or negative voltage. For external wiring, terminal V1 should be connected to the 3rd grounding terminal. Connect the analog input D+to the 8th grounding terminal. M is the common terminal, and it is prohibited to directly connect the output terminal to M to cause an output short circuit.

CAN expansion board specification

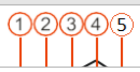
M226ES supports the use of CAN expansion board CTH2-CAN-01S2-EB, supports CANfree and CAN hardware configuration, and imports third-party EDS files.

CAN expansion board specification

physical characteristics							
Dimensions (width x height x depth)	137 x 96 x 62mm						
power waste	9W						
CANopen communication function							
communication interface	1 (8-Pin terminal)						
Transmission rate (kbps)	one thousand	eight hundred	five hundred	two hundred and fifty	one hundred and twenty-five	fifty	twenty
Maximum length (m)	twenty-five	fifty	one hundred	two hundred and fifty	five hundred	one thousand	two thousand and five hundred
Maximum station address	one hundred and twenty-seven						
Site address range	1-127						
Maximum number of master and slave stations	thirty-two						
Maximum Number of Visits				Byte count		Memory starting address	
	input			sixty-four		IB16	
	output			sixty-four		QB16	
Maximum analog access				Number of channels		Memory starting address	
	input			one hundred and sixty-two		AIW64	
	output			one hundred and sixty-two		AQW64	

The wiring diagram and terminal diagram of the CAN expansion board are shown below. The definitions of each terminal position are listed in the subsequent table. When installing and connecting the expansion board, please strictly follow the terminal position definitions for connection.

Terminal position definition

Module interface and connector	**No**	**Definition of Signal**
	1	CAN_H

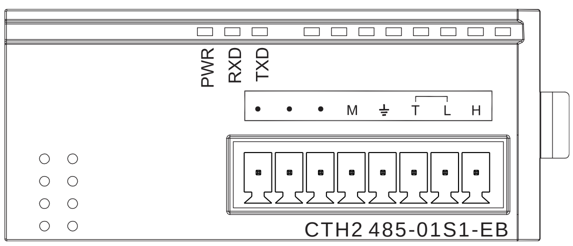
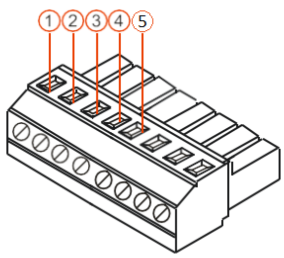
	2	CAN_L
	3	When matching terminal resistors, please short-circuit 2 and 3
	4	land
	5	CAN signal GND
	6	NC
	7	NC
	8	NC

RS485 expansion board specification

Terminal definition

The structure diagram and terminal diagram of the RS485 expansion board are shown below. The definitions of each terminal position are listed in the subsequent table. When installing and connecting the expansion board, please strictly follow the terminal position definitions for connection.

RS485 terminal position definition

Module interface and connector	Terminal position	Definition of Signal
 	1	RS485 signal B/+
	2	RS485 signal A/-
	3	Terminal resistance
	4	Chassis Ground
	5	RS485 signal GND
	6	--
	7	--
	8	--
Note: When terminal resistors are required, short-circuit pin 2 and pin 3.		

Installation of BD expansion board

The specific installation steps for the expansion board are as follows:

1、Ensure that both the CPU and connected devices are powered off, and lay the CPU main control module flat.

2、Remove the cover plate along the left side of the expansion board.

3、Insert the expansion board module into the socket in the board slot, ensuring that the pins are tightly attached to the sockets.

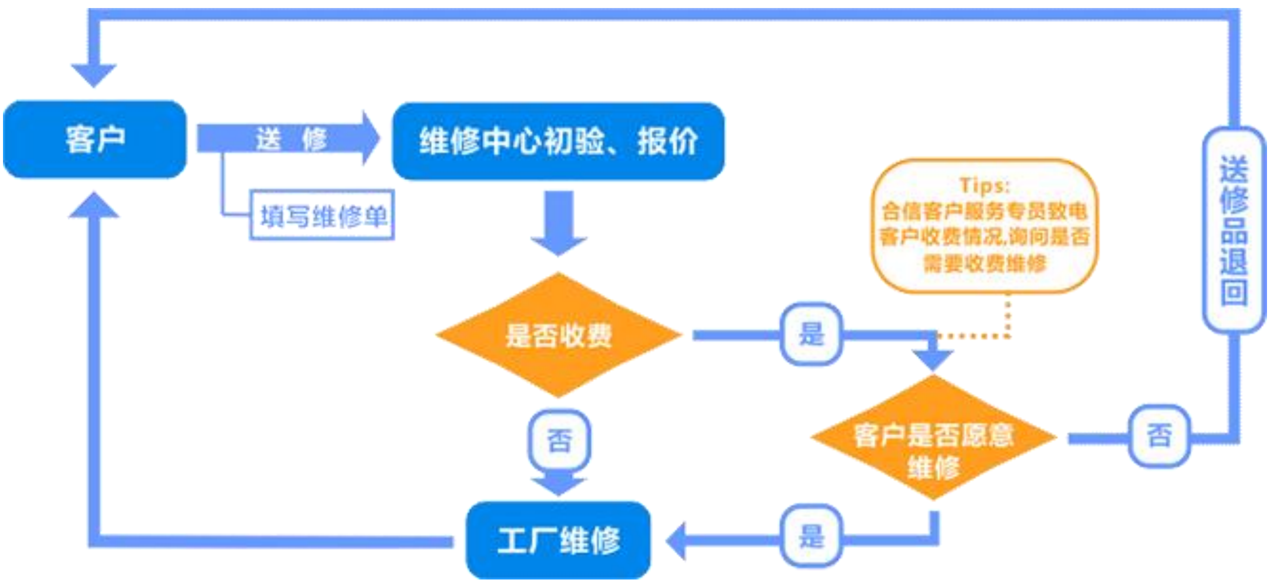
4、Tilt the right flange of the expansion board cover and insert it into the side socket of the expansion board slot on the main control module.

5、Press down diagonally on the left side of the expansion board cover and reinstall it onto the main control module.

⚠	be careful Do not plug or unplug the expansion board with power on.

7.7 After sales maintenance

After sales maintenance process



Maintenance precautions:

1) Please make sure to include a repair checklist in the product being sent for repair.

2) Before sending the product, please backup and save the product program parameters yourself. Repair will erase all product data.

3) Please make sure to send the repaired products and repair orders directly to our after-sales repair center, and the customer will bear the repair shipping cost.

4) To ensure the integrity of the product, please make sure to package it properly when sending it for

repair, which can resist the adverse effects caused by long-distance transportation.

5) Our maintenance center will provide initial inspection feedback within 3 working days of receiving the repaired parts. If you do not receive initial inspection feedback within a week after the repaired parts are shipped, please call to inquire, in order to avoid delays in your work due to product mailing, fax confirmation, and other negligence.

Warranty Responsibility Statement

COTRUST is dedicated to providing comprehensive after-sales service guarantee for all customers, including:

1) Provide free warranty service for PLC products for 24 months from the date of manufacture.

2) Provide free warranty service for building control products for 24 months from the date of manufacture.

3) Provide 18 months of free warranty service for servo drive products from the date of manufacture.

4) For servo motor products, a free warranty service of 18 months is provided from the date of manufacture.

5) Provide 18 months of free warranty service for HMI products from the date of manufacture.

6) Provide paid repair services for human damage and products outside of warranty.

For detailed after-sales maintenance services, please call 075-86226822-6601 (Miss Liu) or call the service hotline 400-700-4858.

Repair Checklist

Repair company:

Delivery address for repair:

Repaired by: Phone: Fax:

No	model	serial number	Fault phenomenon	remarks
1				☑ General repair, ☐ detection, ☐ upgrade
2				☑ General repair, ☐ detection, ☐ upgrade
3				☑ General repair, ☐ detection, ☐ upgrade
4				☑ General repair, ☐ detection, ☐ upgrade
5				☑ General repair, ☐ detection, ☐ upgrade
6				☑ General repair, ☐ detection, ☐ upgrade
7				☑ General repair, ☐ detection, ☐ upgrade
8				☑ General repair, ☐ detection, ☐ upgrade
9				☑ General repair, ☐ detection, ☐ upgrade
10				☑ General repair, ☐ detection, ☐ upgrade

Agent seal:

(Valid with seal)

邮寄地址

收件地址：广东省东莞市大朗镇杨涌村杨新路240号
收件人：售后维修部
联系电话：0769-82220668-7015
邮编：523770

COTRUST TECHNOLOGIES CO., LTD.

9th Floor, Block A, Building 6, Shenzhen International Innovation Valley, Xili Street, Nanshan District, Shenzhen
0755-86226822
market@co-trust.com
www.co-trust.com

Release date: 08/2023
Data Number: CTM1-C2023