Shanghai Siheng Motor Co., Ltd.

AIMOTOR
EtherCAT bus type
Driver Instructions
Simple version

V20250609

Address: No. 1, Lane 399, Songchun Road, Qingpu District, Shanghai
www.sihengmotor.com

First. EtherCAT Product Brief

1.1 Overview

The AIMOTOR drive uses the Ethercat bus communication interface, integrating Ethercat slave technology, vector control technology, built-in ESC slave processor, adaptive filtering technology, and closed-loop control technology, realizing real-time control and real-time data transmission of the system, and optimizing the performance of the motor: it has excellent stability and ultra-low noise; smooth and precise pure sinusoidal current vector control technology effectively reduces motor heating.

Ethercat, the full name of Ethernet for Control Automation Technology, was developed by Beck hoff Atuomation GmbH. It is a real-time Ethernet used for open network communication between master and slave stations. As a mature industrial Ethernet technology, Ethercat has the characteristics of high performance, low cost, and ease of use. The Ethercat network is a master-slave structure network. The network segment can consist of a master station and one or more slave stations. The master station is the control center of the network and the initiator of communication. An Ethercat network segment can be simplified as an independent Ethernet device. The slave station can directly process the received message and extract or insert relevant data from the message. The message is then transmitted to the next Ethercat slave station in sequence. The last Ethercat slave station returns the fully processed message, which is passed back to the first slave station in reverse order and finally sent to the control

unit.

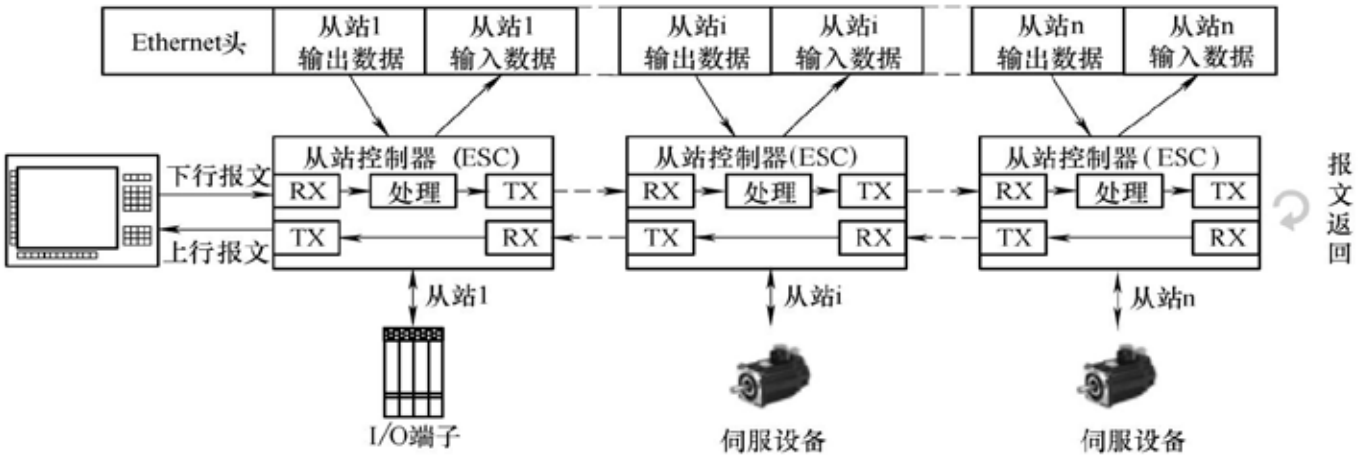


Figure Ethercat operating system

1.2 Ethercat Communication Specifications

EtherCAT Communication function	Physical layer	100BASE-TX
	Communication interface	RJ45(IN input, OUT output)
	Network architecture serial connection	Concatenation (linear link)
	Transmission medium	shielded Category 5e or above network cable
	Transmission rate	2*100Mbps(full duplex)
	Transmission distance	Between two nodes is <100M (good environment and cables), subject to on-site working conditions.
	Number of slave stations	The protocol supports 65535, which is actually limited by the master station drive and on-site working conditions.
	Synchronization Manager	SM0: Mailbox Output SM1: Mailbox Input SM2: Periodic data output SM3: Periodic data output
	Application layer protocol	CoE： CANopen over Ethercat
	Synchronous mode	DC synchronization (Sync0) SM event synchronization FreeRun mode
	The minimum synchronization period	>= 1MS (the actual synchronization period is determined by the number of slave stations and the size of the data stream)
	Communication Object	SDO: acyclic data Object t PDO: Periodic Data Object t
	RJ45 network port LED indication	EtherCAT ERR(ER)x1 EtherCAT Link/Activity(L/A)x2 EtherCAT RUN(RN)x1
	Application layer specifications	IEC61800-7 CiA402 Drive Profile

Supported CIA402 Control Mode	0 x01 PP (contour position mode) 0 x03 PV (contour speed mode) 0 x04 PT (contour torque mode) 0 x06 HM (origin back to zero mode) 0 x08 CSP (position synchronous mode) 0 x09 CSV (synchronous speed mode) 0x0A CST(Synchronous Torque Mode)

1.3 Ethercat communication structure

There are many application layer protocols that can be used for Ethercat communication. However, in AIMortor servo driver, IEC 61800 7 (CiA402) CANopen motion control sub-protocol is used. The following figure shows the Ethercat communication structure based on the CANopen application layer.

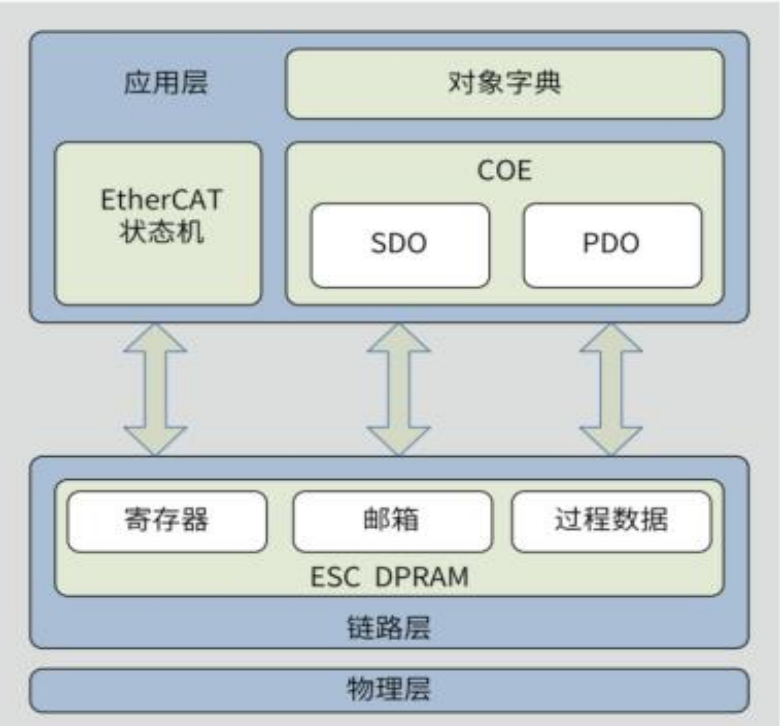


Figure Ethercat communication structure based on CANOpen CIA402 Application layer

In the structure diagram, the application layer object dictionary contains: communication parameters, application data and PDO mapping data; The PDO process data object contains the real-time data during the operation of the servo drive, and it is accessed periodically. For SDO mailbox communication, some communication parameter objects and PDO process data objects are accessed and modified aperiodically.

1.4 Ethercat communication state machine

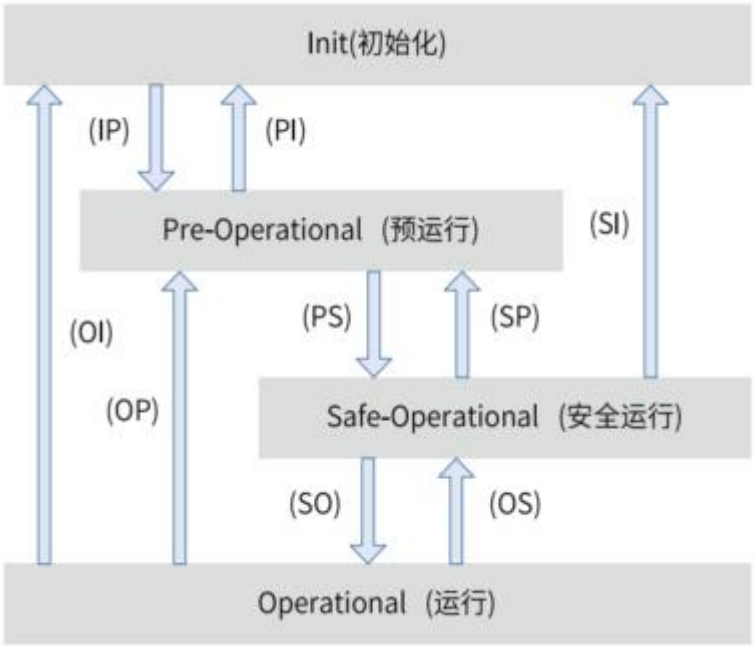


Figure Ethercat Network State Machine

EtherCAT device must support four states and is responsible for coordinating the state of the master and slave applications during initialization and runtime.

Init: initialization, abbreviated as I;

Pre Operational: Pre-run, abbreviated as P;

Safe Operational: Safe operation, abbreviated as S;

Operational: Run, abbreviated as O.

When converting from the initialization state to the running state, the conversion must be performed in the order of "initialization pre-operation safe operation" and cannot be overstepped. Can leapfrog conversion when returning from running state. The conversion operation and initialization process of the state are as follows:

Status	SDO	RPDO	TPDO	Description
Init：Initialization	No	No	No	Communication Initialization; There is no communication in the application layer, and the master station can only read and write the ESC register.。
IP	No	No	No	The master station configures the slave station site address; Configure the mailbox channel; Configure DC distributed clock; Request Pre-Run status.
Pre-Operational：Pre-run	Yes	No	No	Application layer mailbox data communication (SDO)
PS	Yes	No	No	The master station uses SDO to initialize the process data mapping; The master station configures the SM channel used for process data communication; Request Security Status ".
Safe- Operational：Safe operation	Yes	No	Yes	SDO and TPDO can be used, and distributed clock mode can be used.
SO	Yes	No	Yes	The master station sends valid output data; To request the Running Status ".
Operational：run	Yes	Yes	Yes	Normal operation State; Input and output are all valid; You can still use mailbox communications.

1.5 supported synchronization modes

1.5.1 SM2/3 event synchronization

SM Event: the mechanism of Ethercat bus communication is that after the Frame data Frame arrives at the slave station, the slave station ESC will trigger the SM Event signal, and the application layer processor receives the SM Event signal to interrupt for input and output data processing.

Generally speaking, if the synchronization mode during EtherCAT bus communication is not DC mode, then it is SM synchronization mode. SM(Sync Manager) refers to the synchronization of the Sync Manager. Its triggering method is through SM Event. That is, when the data frame of the master station reaches the corresponding slave station, a signal called the Sync Manager Event (i.e., SM Event) will be triggered. When the slave station receives this signal, It will enter the corresponding interrupt service routine to handle the corresponding data (for example, making the output data valid and then placing the input data on the channel of the synchronization manager for the master station to retrieve).

1.5.2 DC distributed clock

The distributed clock enables all Ethercat devices to use the same system time, thereby controlling the synchronous execution of tasks of each device, and the Slave station ESC device can generate a synchronization signal according to the synchronized system time. Support DC synchronization mode, the synchronization cycle is controlled by SYNC0, the application layer processor receives the SYNC0 event signal interrupt for input, output data processing.

The Sync0 Event and SM Event event signals are similar. They are both interrupt event signals. For a slave station, if the Sync0 interrupt signal is triggered and the interrupt masking register does not mask the interrupt signal, then the slave station will enter the interrupt service routine to handle the corresponding data (such as rendering the output data valid). Then, the input data is placed on the channel of the synchronization manager for the master station to retrieve. The difference from the SM Event is that the Sync0 Event is triggered based on the synchronization clock delay time calibrated by ESC, rather than when the frame arrives.

1.5.3 FreeRun free mode

In free mode, the application layer processor cycles to the slave station ESC for input and output data processing, regardless of the synchronization cycle of the master station.

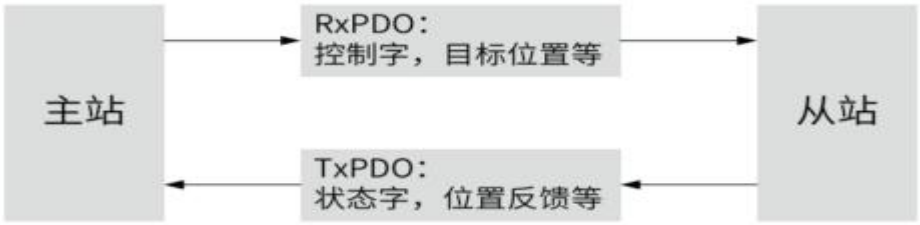
To each time period, each from the station to carry out your own programs, < check channels such as for appropriate new input data, so makes the Output effective (the Output valid) or to check there is no need to input data, Put it on the corresponding synchronization manager channel (input prepare) and let the master station retrieve it >

Note: It is recommended that the synchronization cycle time should not be less than 1ms when configuring the master station. The synchronization cycle time should be appropriately adjusted and increased with the number of network nodes and the size of transmission data stream.

1.6 PDO Communication

1.6.1 PDO Transmission and mapping mechanism

EtherCAT real-time data transmission through the Process data (Process data Object t) to achieve. According to the data transmission direction, PDO can be divided into RxPDO(Reception PDO) and TxPDO(Trasmission PDO). RxPDO transmits data from the master station to the slave station, and TxPDO feeds back data from the slave station to the master station.



4groups of RxPDO are supported, and the mapping Object t parameter sets are 0x 1600,0x 1601,0x 1602, and 0x 1603. Among them, 0x 1600 RxPDO1 supports variable mapping, and the rest are factory-fixed mapping.
Object t 0x1C12-01 decides to select 0x 1600, 0x 1601, 0x 1602, 0x 1603, one of the set of mapping Object ts as the actual RxPDO to use.
4TxPDO groups are supported, and the mapping Object t parameter sets are 0x1A00, 0x1a01, 0x1a02, and 0x1A03. Among them, 0x1A00 TxPDO1 supports variable mapping, and the rest are factory-fixed mapping.
The Object t 0x1C13-01 decides to select 0x1a00, 0x1a01, 0x1a02, 0x1a03, one of the mapped Object t parameter groups as the actual TxPDO used.
The maximum number of (Tx/RxPDO) Object ts in each variable mapping is not more than 10. The length of each PDO data is not more than 40 bytes,

1.6.2 PDO Configuration

The PDO mapping parameter contains a pointer to the process data corresponding to the PDO that needs to be sent or received by the PDO, including an index, a sub-index, and a mapping Object t length. The sub-index 0 records the number N of Object ts to which the PDO is specifically mapped, and one or more Object ts can be mapped at the same time. The sub-indexes 1 to N are mapping contents. The mapping parameters are defined as follows

Bit	31 --- 16	15 --- 8	7 --- 0
Meaning	Object Index	Object sub-index	Data length (8 means 8 bits, 10=16 bits, 20=32 bits)

For example, the control word 0x 6040-00 is mapped to the 1600 Object t 01, and the sub-index representation is 0x60400010;
Target position 0x607A-00, mapped to 1600 Object t 02 sub-index representation is 0x607a0020;
The mapping configuration of PDO follows a specific process, as follows:
Note: The PDO configuration can only be done when the Ethercat communication state machine is in a Pro Operation, otherwise the configuration is unsuccessful.
1.Configure mapping parameter set collection of objects (with 0 x1600 RxPDO1, 0 x1a00 TxPDO1, for example)
A: To clear the original map object (0) x1600-00 0 = 0 x00; 0x1A00-00=0X00;
B: Again write new mapping object, 0 x1600-01 = 0 x60400010 (control word), 0 = 0 x60600008 x1600-02 (control model), 0

x1600-03 = 0 x607a0020 (target); 0 x1a00-01 = 0 x60410010 (status word), 0 = 0 x1a00-02 x60610008 feedback (mode), 0 x1a00-03 = 0 x60640020 (actual position);

C:Finally writes the PDO mapping object number 0 x1600-00 = 0 x03; 0x1A00-00=0x03;

2. Select the actual mapping group

A:To clear the original choice of mapping object parameter set (0) x1c12-00 0 = 0 x00; 0x1C13-00=0x00;

B:Write again choose mapping object parameter set 0 x1c12-01 = 0 x1600; (select RxPDO1 mapping group as communication) RxPDO process 0 x1c13-01 = 0 x1a00; (Select the TxPDO1 mapping group as the TxPDO process communication)

C:Finally, write the number of mapping parameter groups (fixed to 1)

0x1C12-00=0x01; 0x1C13-00=0x01;

1.7 SDO Communications

EtherCAT mailbox data SDO is used to transmit non-periodic data. Generally, the master station uses SDO to carry out a series of basic configurations before controlling the slave station to enter the Operation State, such as the configuration of communication parameters, mapping configuration, servo drive operation parameter configuration, etc;

1.8 Port and Networking Mode

1.8.1 Linear topology networking connection mode

Communication Cable Specifications: at least over 5 twisted pair shielded Ethernet cables; The transmission distance is long, and the working conditions of electromagnetic interference are harsh. It is necessary to appropriately increase the Ethernet cable grade such as over 6 and above.

1.8.2 RJ45 Description

EtherCAT-IN		EtherCAT-OUT		Function
Description	Pin	Description	Pin	
TD+	1	TD+	1	Send Data +
TD-	2	TD-	2	Send Data -
RD+	3	RD+	3	Data Receive +
-	4	-	4	-
-	5	-	5	-
RD-	6	RD-	6	Data Receive -
-	7	-	7	-
-	8	-	8	-
FG	Connector housing	FG	Connector housing	Protective shield ground

1.8.3 RJ45 Indicator Description

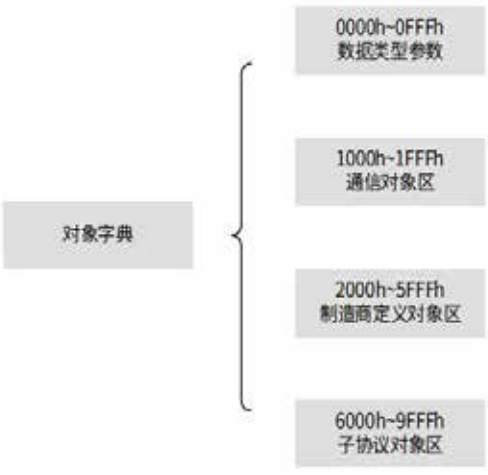
Indicator light	Status	Status description
RUN indicator	OFF	INIT Status
	Blinking（on 200 ms/ off 200ms)	Pre-Operational Status
	Single flash（on 200 ms / off 1000 ms)	Safe-Operational Status
	ON	Operational Status
ERR indicator	OFF	No network failure
	Blinking（on 200 ms/ off 200ms)	Communication setting is abnormal
	Single flash（on 200 ms / off 1000 ms)	Sync error/SM error
	Double flash（on 200 ms / off 1000 ms；on 200ms， off 1000ms)	Application Watchdog Timeout
L/A0 indicator L/A1 indicato r	OFF	Communication not connected
	Flickering(on 50 ms/ off 50ms)	Communication connected and activated
	ON	Communication connected and activated

Second. Parameter description and setting of Object dictionary

2.1 Object dictionary

The Object dictionary is the most important part of the device specification. It is an ordered set of parameters and variables, containing all the parameters of the device description and device network status. A set of Object ts that can be accessed in an ordered, predefined manner through the network.

This Ethercat device is based on the COE service (CAN Application protocol based on Ethercat) and uses an Object dictionary with 16-bit index and 8-bit sub-index. The structure of the Object dictionary is shown in the figure on the right.



0000h-0FFFh user has no access meaning;

1000h-1FFFh defines communication Object t parameters and device types according to the CIA301 protocol;

2000h-5FFFh manufacturer defines the Object t area, and our company defines all parameter function code Object ts of the driver in this area;

6000h-9FFFh Some common CIA402 control Object ts defined according to the CIA402 protocol;

The correspondence between the drive parameter function code and the Object dictionary is as follows:

Object dictionary index = 0x 2000 + function code group number; Object dictionary sub-index = biased hexadecimal within function code Group +1

Example of mapping between driver parameter function code and Object dictionary:

Parameter function code			Corresponding Object dictionary address	
Parameter function code name	Function code group number (hexadecimal)	Intra-group offset (base 10)	Index (hex)	Sub-index (hex)
H02-00	0x02	00	0x2002	0x01
H03-09	0x03	09	0x2003	0x0A
H0B-26	0x0B	26	0x200B	0x1B
H12-20	0x12	20	0x2012	0x15

2.1.1 CIA301 Protocol communication parameters

Index	Sub-index	Name	Description	Type	Property	Default Value	Scope
1000	0	Device Type	Consistent with CIA rules	Uint32	RO	0x20192	-
1001	0	Error register	The error type information is represented by bits;	Uint8	RO	-	
1008	0	Manufacturer Equipment Name	-	str	RO	sh_aimotor	-
1009	0	Manufacturer Hardware Version	-	str	RW	n.a.	-
100A	0	Manufacturer Software Version	-	str	RO	5.11	-
1018	0	Number of sub-indexes	Product information related	Uint8	RO	0x04	-
	1	Supplier ID		Uint32	RO	0x01	-
	2	Product Code		Uint32	RO	0x26483052	-
	3	Revision Number		Uint32	RO	0x01	-
	4	Product Serial Number		Uint32	RO	0x01	-
1600	0	RxPDO1 Number of mappings	Number of variable mapping Object ts （Variable mapping）	Uint8	RW	0x03	0x00-0x0A
	1	RxPDO1Mapping Object t 1	Default mapping 0x 60400010 (control word)	Uint32	RW	0x60400010	0x00~0xFFFFFFFF
	2	RxPDO1 Mapping Object t 2	Default mapping 0x 60600008 (control mode)	Uint32	RW	0x60600008	0x00~0xFFFFFFFF
	3	RxPDO1 Mapping Object t 3	Default mapping 0x607a0020 (target location))	Uint32	RW	0x607A0020	0x00~0xFFFFFFFF
	4-10	RxPDO1 Mapping	Not mapped, user can freely	Uint32	RW	0x00000000	0x00~0xFFFFFFFF

		Object t 4-10	assign				
Index	Sub-index	Name	Description	Type	Property	Default Value	Scope
1601	0	RxPDO2 Number of mappings	Fixed number of mapping Objects (PP mode is applicable)	Uint8	RW	0x06	0x00-0x06
	1	RxPDO2 Mapping Object 1	Fixed mapping 0x 60400010 (control word)	Uint32	RW	0x60400010	0x00~0xFFFFFFFF
	2	RxPDO2 Mapping Object 2	Fixed mapping 0x 60600008 (control mode)	Uint32	RW	0x60600008	0x00~0xFFFFFFFF
	3	RxPDO2 Mapping Object 3	Fixed mapping 0x607a0020 (target position)	Uint32	RW	0x607A0020	0x00~0xFFFFFFFF
	4	RxPDO2Mapping Object 4	Fixed mapping 0x 60810020 (contour speed)	Uint32	RW	0x60810020	0x00~0xFFFFFFFF
	5	RxPDO2Mapping Object 5	Fixed mapping 0x 60830020 (acceleration)	Uint32	RW	0x60830020	0x00~0xFFFFFFFF
	6	RxPDO2Mapping Object 6	Fixed mapping 0x 60840020 (deceleration)	Uint32	RW	0x00000000	0x00~0xFFFFFFFF
1602	0	RxPDO3 Number of mappings	Fixed number of mapping Objects (applicable to PV mode)	Uint8	RW	0x05	0x00-0x05
	1	RxPDO3Mapping Object 1	Fixed mapping 0x 60400010 (control word)	Uint32	RW	0x60400010	0x00~0xFFFFFFFF
	2	RxPDO3Mapping Object 2	Fixed mapping 0x 60600008 (control mode)	Uint32	RW	0x60600008	0x00~0xFFFFFFFF
	3	RxPDO3Mapping Object 3	Fixed mapping 0x60ff0020 (target speed)	Uint32	RW	0x60FF0020	0x00~0xFFFFFFFF
	4	RxPDO3Mapping Object 4	Fixed mapping 0x 60830020 (acceleration)	Uint32	RW	0x60830020	0x00~0xFFFFFFFF
	5	RxPDO3Mapping Object 5	Fixed mapping 0x 60840020 (deceleration)	Uint32	RW	0x60840020	0x00~0xFFFFFFFF
1603	0	RxPDO4 Number of mappings	Fixed number of mapping Objects (for HM mode)	Uint8	RW	0x05	0x00-0x0A
	1	RxPDO4 Mapping Object 1	Fixed mapping 0x 60400010 (control word)	Uint32	RW	0x60400010	0x00~0xFFFFFFFF
	2	RxPDO4 Mapping Object 2	Fixed mapping 0x 60600008 (control mode)	Uint32	RW	0x60600008	0x00~0xFFFFFFFF
	3	RxPDO4 Mapping Object 3	Fixed mapping 0x 60980008 (back to original mode)	Uint32	RW	0x60980008	0x00~0xFFFFFFFF
	4	RxPDO4 Mapping Object 4	Fixed mapping 0x 60990120 (back to the original fast)	Uint32	RW	0x60990120	0x00~0xFFFFFFFF
	5	RxPDO4 Mapping Object 5	Fixed mapping 0x 60990220 (slow return)	Uint32	RW	0x60990220	0x00~0xFFFFFFFF
1A00	0	TxPDO1 Number of	Number of variable mapping	Uint8	RW	0x05	0x00-0x0A

		mappings	Object ts (variable mapping)				
	1	TxPDO1Mapping Object 1	Default mapping 0x 60410010 (status word)	Uint32	RW	0x60410010	0x00~0xFFFFFFFF
	2	TxPDO1Mapping Object 2	Default mapping 0x 60610008 (mode feedback)	Uint32	RW	0x60610008	0x00~0xFFFFFFFF
	3	TxPDO1Mapping Object 3	Default mapping 0x 60640020 (actual location)	Uint32	RW	0x60640020	0x00~0xFFFFFFFF
	4	TxPDO1Mapping Object 4	Default mapping 0x606c0020 (actual speed)	Uint32	RW	0x606C0020	0x00~0xFFFFFFFF
	5	TxPDO1Mapping Object 5	Default mapping 0x 60770010 (actual torque)	Uint32	RW	0x60770010	0x00~0xFFFFFFFF
	6-10	TxPDO1Mapping Object 6-10	Not mapped, user can freely assign	Uint32	RW	0x00000000	0x00~0xFFFFFFFF

Index	Sub-index	Name	Description	Type	Property	Default Value	Scope
1A01	0	TxPDO2 Number of mappings	Fixed number of mapping Objects (PP mode is applicable)	Uint8	RW	0x03	0x00-0x06
	1	TxPDO2 Mapping Object 1	Fixed mapping 0x 60410010 (status word)	Uint32	RW	0x60410010	0x00~0xFFFFFFFF
	2	TxPDO2Mapping Object 2	Fixed mapping 0x 60610008 (mode feedback)	Uint32	RW	0x60610008	0x00~0xFFFFFFFF
	3	TxPDO2Mapping Object 3	Fixed mapping 0x 60640020 (actual position)	Uint32	RW	0x60640020	0x00~0xFFFFFFFF
1A02	0	TxPDO3 Number of mappings	Fixed number of mapping Objects (applicable to PV mode)	Uint8	RW	0x05	0x00-0x05
	1	TxPDO3Mapping Object 1	Fixed mapping 0x 60410010 (status word)	Uint32	RW	0x60410010	0x00~0xFFFFFFFF
	2	TxPDO3Mapping Object 2	Fixed mapping 0x 60610008 (mode feedback)	Uint32	RW	0x60610008	0x00~0xFFFFFFFF
	3	TxPDO3Mapping Object 3	Fixed mapping 0x606c0020 (actual speed)	Uint32	RW	0x606C0020	0x00~0xFFFFFFFF
1A03	0	TxPDO4 Number of mappings	Fixed number of mapping Objects (applicable to PT mode)	Uint8	RW	0x05	0x00-0x05
	1	TxPDO4Mapping Object 1	Fixed mapping 0x 60410010 (status word)	Uint32	RW	0x60410010	0x00~0xFFFFFFFF
	2	TxPDO4Mapping Object 2	Fixed mapping 0x 60610008 (mode feedback)	Uint32	RW	0x60610008	0x00~0xFFFFFFFF
	3	TxPDO4Mapping Object 3	Default mapping 0x 60770010 (actual torque)	Uint32	RW	0x60770010	0x00~0xFFFFFFFF
1C00	0	Number of sub-indexes	4 SM channel configuration SM0: mailbox receive SM1 mailbox send SM2: Process data output RxPDO SM3: Process data input TxPDO	Uint8	RO	0x04	-
	1	SM0 Communication Type		Uint8	RO	0x01	-
	2	SM1 Communication Type		Uint8	RO	0x02	-
	3	SM2 Communication Type		Uint8	RO	0x03	-
	4	SM3 Communication Type		Uint8	RO	0x04	-
1C12	0	Number of sub-indexes	Select one of 0x 1600,0x 1601,0x 1602,0x 1603 as the RxPDO transport index	Uint8	RW	0x01	0x00~0x01
	1	Set the allocated Object t index of RxPDOd (SM2)		Uint16	RW	0x1600	0x1600~0x1603
1C13	0	Number of	Select	Uint8	RW	0x01	0x00~0x01

		sub-indexes	0x1a00, 0x1a01, 0x1a02, 0x1A03 One of them as TxPDO transport index				
	1	Set the allocated Object t index of TxPDOd (SM3))		Uint16	RW	0x1A00	0x1A00~0x1A03

Index	Sub-index	Name	Description	Type	Property	Default Value	Scope
1C32	0	Number of sub-indexes	SM2 Synchronization Manager Output Parameters	Uint8	RO	0x20	-
	1	Synchronization type	Configure the SM2 synchronization type	Uint16	RW	0x0000	0~0xFFFF
	2	Cycle time	SM2 cycle time	Uint32	RO	0x0000	-
	3	Shift time	Retention	Uint32	-	-	-
	4	Supported Synchronization Modes	Indicates the sync modes supported by this device	Uint16	RO	0x401F	0
	5	Support minimum cycle time	Minimum> 1000000ns	Uint32	RO	0x000F4240	0~0xFFFFFFFF
	6	Calculation and replication time	Retention	Uint32	-	-	-
	8	Get cycle time	For measuring cycle time	Uint16	RW	0x0000	0~0xFFFF
	9	Delay time	Retention	Uint32	-	-	-
	10	SYNC0 Cycle Time	Cycle time for SYNC0	Uint32	RW	0x00000000	0~0xFFFFFFFF
	11	SM event loss count	Record the number of times SM events are lost	Uint16	RO	0x0000	0~0xFFFF
	12	Cycle Overflow Count	Record the number of times the loop overflows	Uint16	RO	0x0000	0~0xFFFF
	32	SYNC error	SYNC error indication	Uint8	RO	0x00	0~0x01
1C33	0	Number of sub-indexes	SM3 Synchronization Manager Output Parameters	Uint8	RO	0x20	-
	1	Synchronization type	Configuring the SM3 Synchronization Type	Uint16	RW	0x0000	0~0xFFFF
	2	Cycle time	SM3 cycle time	Uint32	RO	0x0000	-
	3	Shift time	Retention	Uint32	-	-	-
	4	Supported Synchronization Modes	Indicates the sync modes supported by this device	Uint16	RO	0x401F	0
	5	Support minimum cycle time	Minimum> 1000000ns	Uint32	RO	0x000F4240	0~0xFFFFFFFF
	6	Calculation and replication time	Retention	Uint32	-	-	-
	8	Get cycle time	For measuring cycle time	Uint16	RW	0x0000	0~0xFFFF
	9	Delay time	Retention	Uint32	-	-	-

	10	SYNC0 Cycle Time	Cycle time for SYNC0	Uint32	RW	0x00000000	0~0xFFFFFFFF
	11	SM event loss count	Record the number of times SM events are lost	Uint16	RO	0x0000	0~0xFFFF
	12	Cycle Overflow Count	Number of times the loop overflows are recorded	Uint16	RO	0x0000	0~0xFFFF
	32	SYNC error	SYNC error indication	Uint8	RO	0x00	0~0x01

2.1.2 Manufacturer customized parameters

2000h-5FFFh Manufacturer defined Object t area, our company defines all parameter function code Object ts of AIMOTOR drive in this area;

Each parameter has a corresponding dictionary Object t index and sub-index. The correspondence between the AIMOTOR driver parameter function code and the Object dictionary is as follows:

Object dictionary index = 0x2000 + function code group number; Object dictionary sub-index = hexadecimal offset within the function code group + 1;

Example: Mapping relationship between driver parameter function code and Object dictionary

Parameter function code			Corresponding Object dictionary address	
Parameter function code name	Function code group number (hexadecimal)	Intra-group offset (base 10)	Index (hex)	Sub-index (hex)
H02-00	0x02	00	0x2002	0x01
H03-09	0x03	09	0x2003	0x0A
H0B-26	0x0B	26	0x200B	0x1B
H12-20	0x12	20	0x2012	0x15

Due to too many parameter function codes, they are not listed in the Ethercat specification. Please refer to the parameter description in the specification of the same series of standard pulse products.

When using Ethercat communication to read and write factory-defined parameter Object ts, please pay attention to the properties of the parameters;

1. If the modification method attribute is parameter, you need to disconnect the motor enable first and then write the parameter Object t value;

2. If the factory value attribute is the parameter, the user can only read but not write;

H00 group, H01 group, H0B group parameters, the user can only read but not write;

Note: Regarding the setting of the number of pulses in one revolution of the motor (feedback resolution)

Motor encoder resolution	Electronic Gear Molecule (H05-07)	Electronic Gear Molecule (H05-09)	Input and feedback one round actual pulse equivalent
Take 17-bit resolution encoder as an example, the resolution is 131072	0（Indicates the corresponding encoder resolution）	1000	=1000
	4	3	131072/4*3=98304
	10000	2500	131072/10000*2500=32768
	131072	2500	131072/131072*2500=2500

Factory default electronic gear numerator (H05-07)= 0, electronic gear ratio mother (H05_09)= 1000, I .e. input and feedback pulse equivalent = 1000;

2.1.3 CIA402 Axis control parameters

<table>
<tr><th>Index</th><th>Sub-index</th><th>Name</th><th>Description</th><th>Type</th><th>Property</th><th>Default Value</th><th>Scope</th></tr>
<tr><td>603F</td><td>0</td><td>Fault Code</td><td>The fault code currently generated by the driver is described in chapter '4.1 Fault Code Information Table;</td><td>Uint16</td><td>RO</td><td>0</td><td>-</td></tr>
<tr><td>6040</td><td>0</td><td>Control Word</td><td>See the description of "2.4 6040 Control Word" for details;</td><td>Uint16</td><td>RW</td><td>0</td><td>0~65535</td></tr>
<tr><td>6041</td><td>0</td><td>Status word</td><td>See Chapter 2.5 6041 Status Word Description for details.;</td><td>Uint16</td><td>RO</td><td>-</td><td>-</td></tr>
<tr><td>605A</td><td>0</td><td>Quick Stop Code</td><td><table><tr><th>Value</th><th>Source of deceleration</th><th>After reducing to 0</th></tr><tr><td>0</td><td>Inertia free stop</td><td>Disable enable</td></tr><tr><td>1</td><td>According 0x6084</td><td>Disable enable</td></tr><tr><td>2</td><td>According 0x6085</td><td>Disable enable</td></tr><tr><td>3</td><td>According 0x6084</td><td>Hold lock shaft</td></tr><tr><td>4</td><td>According 0x6085</td><td>Hold lock shaft</td></tr></table></td><td>Uint16</td><td>RW</td><td>1</td><td>0~65535</td></tr>
<tr><td>6060</td><td>0</td><td>Run Mode Settings</td><td>0: No effect
1:PP (contour position mode)
3:PV (contour speed mode)
4:PV (contour torque mode)
6:HM (back to Origin mode)
8:CSP (cyclic synchronous position mode)</td><td>Int8</td><td>RW</td><td>1</td><td>0~8</td></tr>
<tr><td>6061</td><td>0</td><td>Running mode status</td><td>Query the status of 6060H;
0: No effect
1:PP (position mode)
3:PV (speed mode)
4:PV (contour torque mode)
6:HM (back to Origin mode)
8:CSP (cyclic synchronous position mode)</td><td>Int8</td><td>RO</td><td>-</td><td>-</td></tr>
<tr><td>60FC</td><td>0</td><td>Internal Command Location</td><td>Position commands generated inside the drive;
Unit: Enc</td><td>Int32</td><td>RO</td><td>-</td><td>-</td></tr>
<tr><td>6062</td><td>0</td><td>Internal Command</td><td>Position commands generated inside the drive;</td><td>Int32</td><td>RO</td><td>-</td><td>-</td></tr>
</table>

		Location	Unit: Pul				
6063	0	Actual Location	Actual position of motor, Unit: Enc	Int32	RO	-	-
6064	0	Actual Location	Actual position of motor, Unit: Pul	Int32	RO	-	-
606B	0	Internal Command Speed	Speed command generated inside the drive; Unit: Pul/s	Int32	RO	-	-
606C	0	Actual speed	Actual motor speed, Unit: Pul/s	Int32	RO	-	-
6071	0	Target torque	Set PT (contour torque mode) operating torque;(1000 indicates rated torque) Unit: 0.1 percent	Int16	RW	0	-3000~3000
Index	Sub-index	Name	Description	Type	Property	Default Value	Scope
6074	0	Internal commanded torque	Torque command generated inside the drive; Unit: 0.1 percent	Int16	RO	-	-
6077	0	Actual torque	Actual motor torque, Unit: 0.1 percent	Int16	RO	-	-
6078	0	Actual current	Actual motor current, Unit: 0.01A	Int16	RO	-	-
607A	0	Target Location	Set PP (contour position mode) running position; Unit: Pul	Int32	RW	0	-2147483647~2147483647
607C	0	Origin Offset	Set HM (return to Origin mode) origin offset; Unit: Pul	Int32	RW	0	-2147483647~2147483647
607D	0	number of sub-indexes	-	Uint8	RO	2	0~255
	1	Negative limit of the software	Software Settings take effect CANOPEN pattern software limit, limit function also need to set H0A - 40 parameters	Int32	RW	-2147483648	-2147483648~2147483647
	2	Software forward limit		Int32	RW	2147483647	-2147483648~2147483647
607E	0	Command polarity	Set commands in PP/PV/PT profile mode Directional logic;	Uint8	RW	0	0~255
607F	0	Maximum profile speed	Set the maximum speed limit for PT (contour torque mode) operation; Unit: Pul/s	Uin32	RW	0	-2147483647~2147483647
6081	0	Contour speed	Set the maximum speed of PP contour position mode;	Int32	RW	0	-2147483647~2147483647

			Unit: Pul/s				
6083	0	Contour acceleration	Setting PP and PV contour mode acceleration; Unit: Pul/s ^ 2	Int32	RW	100000	-2147483647~2147483647
6084	0	Contour deceleration	Set the deceleration of PP and PV contour modes; Unit: Pul/s ^ 2	Int32	RW	100000	-2147483647~2147483647
6085	0	Quick stop deceleration	Set the fast stop deceleration in PP/PV/PT contour mode; Unit: Pul/s ^ 2	Int32	RW	500000	-2147483647~2147483647
6098	0	Return to Origin mode	At present, the value that can be set to return to the origin is: 01/02/04/06/17/18/24/28/33/34/35/37/38; See the description in chapter '2.8 Return to Origin Mode Method;	Int8	RW	35	0~38
6099	0	Number of sub-indexes	-	Uint8	RO	2	-
	1	Origin Seeker Speed 1	HM (return to Origin mode) speed approaching the origin: high speed to find the origin; Unit: Pul/s	Int32	RW	10000	-2147483647~2147483647
	2	Origin Seeking Speed 2	HM (return to Origin mode) determine the origin speed: low speed tao find the origin; Unit: Pul/s	Int32	RW	1000	-2147483647~2147483647
609A	0	Return to Origin plus or minus speed	Return to the origin of HM (return to Origin mode) Acceleration and deceleration; Unit: Pul/s ^ 2	Int32	RW	10000	-2147483647~2147483647
60FD	0	Digital signal input status	DI1-DI7 input port status feedback Bary 2 bit15-bit21 Bit Representation	Uint32	RO	-	-

Index	Sub-index	Name	Description	Type	Property	Default Value	Scope
60FE	0	Number of sub-indexes	-	Uint8	RO	2	-
	1	Digital signal output	See the description in chapter '2.11 60FEh'	Uint32	RW	-	-2147483647~2147483647
	2	Digital Signal Force Mask		Uint32	RW	0	-2147483647~2147483647
60FF	0	Target Speed	Set PV (profile speed mode) Operating speed; Unit: Pul/s	Int32	RW	0	-2147483647~2147483647

6502	0	Operating modes supported by this driver	The CIA402 motion control mode for querying product support	Uint32	RO	0x00AD	-

2.2 Each control mode and associated Object Dictionary

Index	Sub-index	Name	Description	Type	Property	Default Value
Synchronous position mode (CSP)	6060	Control Mode Setting = 8	Int8	RW	-	√
	6040	Control Word	Uint16	RW	-	√
	607A	Target Location	Int32	RW	Pul	√
	6061	Control Mode Feedback	Int8	RO	-	
	6041	Status word	Uint16	RO	-	√
	6064	Actual Location	Int32	RO	Pul	√
	606C	Actual speed	Int32	RO	Pul/s	
	6077	Actual torque	Int16	RO	0.1%	
Profile Position Mode (PP)	6060	Control Mode Setting = 1	Int8	RW	-	√
	6040	Control Word	Uint16	RW	-	√
	607A	Target Location	Int32	RW	Pul	√
	6081	Maximum speed	Int32	RW	Pul/s	√
	6083	Acceleration	DINT	RW	Pul/s^2	√
	6084	Decrate	DINT	RW	Pul/s^2	√
	6061	Control Mode Feedback	Int8	RO	-	
	6041	Status word	Uint16	RO	-	√
	6064	Actual Location	Int32	RO	Pul	
	606C	Actual speed	Int32	RO	Pul/s	
	6077	Actual torque	Int16	RO	0.1%	
Index	Sub-index	Name	Description	Type	Property	Default Value
Profile speed mode (PV)	6060	Control Mode Setting = 3	Int8	RW	-	√
	6040	Control Word	Uint16	RW	-	√
	60FF	Target Speed	DINT	RW	Pul/s	√
	6083	Acceleration	DINT	RW	Pul/s^2	√

	6084	Decrate	DINT	RW	Pul/s^2	
	6061	Control Mode Feedback	Int8	RO	-	
	6041	Status word	Uint16	RO	-	
	6064	Actual Location	Int32	RO	Pul	
	606C	Actual speed	Int32	RO	Pul/s	
	6077	Actual torque	Int16	RO	0.1%	
Contour torque mode (PT)	6060	Control Mode Setting = 4	Int8	RW	-	√
	6040	Control Word	Uint16	RW	-	√
	6071	Target torque	Int16	RW	0.1%	√
	607F	Maximum speed limit	Uint32	RW	Pul/s	√
	6061	Control Mode Feedback	Int8	RO	-	
	6041	Status word	Uint16	RO	-	
	6064	Actual Location	Int32	RO	Pul	
	606C	Actual speed	Int32	RO	Pul/s	
	6077	Actual torque	Int16	RO	0.1%	
Back to Origin Mode (HM)	6060	Control Mode Setting = 6	Int8	RW	-	√
	6040	Control Word	Uint16	RW	-	√
	6098	Back to zero method	Int8	RW	-	√
	6099-01	Origin Seeker High Speed 1	Int32	RW	Pul/s	√
	6099-02	Seek Origin Low Speed 2	Int32	RW	Pul/s	√
	609A	Origin acceleration	Int32	RW	Pul/s^2	√
	607C	Origin Offset	Int32	RW	Pul	
	6061	Control Mode Feedback	Int8	RO	-	
	6041	Status word	Uint16	RO	-	
	6064	Actual Location	Int32	RO	Pul	
	606C	Actual speed	Int32	RO	Pul/s	
	6077	Actual torque	Int16	RO	0.1%	
	605A	Quick Stop Code	Uint16	RW	-	

Other Auxiliary Association parameters	6085	Emergency stop deceleration	Uint32	RW	Pul/s^2	
	607E	Command polarity	Uint8	RW	-	
	607D-01	Negative limit of the software	Int32	RW	-	
	607D-02	Software forward limit	Int32	RW	-	
	60FD	Command polarity	Uint32	RO	-	
	60FE	Digital output status	Uint32	RO	-	

No matter which control mode is used to control the slave station, it is inseparable from the reading and writing operations of the two Object t dictionaries of 6040H (control word) and 6041H (status word). The master slave station uses these two Object t dictionaries as transmission media to realize instruction issuance and status monitoring. Section 2.4-2.5 focuses on the definitions and meanings of the bits in these two Object t dictionaries.

2.3 CIA402 Object dictionary position, speed, acceleration and deceleration unit calculation

The speed value in CIA402 protocol is not the unit of rotation per minute r/min as we understand it, but the unit of instruction pulses per second Pul/s; The unit of acceleration and deceleration is Pul/s ^ 2;If I want to set the speed per minute, I should first convert the minutes into seconds, and then multiply it by the number of pulses in 1 turn of the motor;

AIMotor series drives the factory default 1000 pulse motor to rotate for one turn. If you change the number of pulses for one turn, please use debugging software or SDO communication method to set the denominator of H05-07 electronic gear and H05-09 electronic gear. If you set the 2000 pulse motor to rotate for one turn, you can set the H05-09 parameter value to 2000;

Schematic Table of Unit Conversion:

1Number of lap pulses	Unit of user understanding				Convert to units that follow the CIA402 protocol			
H05-07 molecule H05-09 denominator	Target position	Target speed r/min	Acceleration time to target	Deceleration time to slow down to 0	Target location Pul	Target Speed Pul/s	Acceleration Pul/s ^ 2	Deceleration Pul/s ^ 2
Factory default 1000	Run 100.5 turn	2000 revolutions per minute	Accelerate to 2000 revolutions per minute in 2 seconds	3 seconds to slow to a full stop	100.5*1000 =100500	2000/60*1000 =33333	33333/2 =16666	33333/3 =11111

For example: assuming that the control mode is PP contour position mode, as shown in the above table, according to the number of pulses in one turn and the target position, target speed and acceleration and deceleration time expected by the user, calculate the position value, speed value and acceleration and deceleration value in CIA402, and then use Ethercat communication to control the specific steps of motor absolute positioning operation;

Step1：Use the upper computer to set the driver control mode parameter H02-00 to 9 (EtherCAT control mode);

Step2: Communication writes the 6060H motion mode in the CIA402 Object dictionary into 1 (contour position mode);

Step3: Communication writes the target position 607Ah, the related Object t of the contour position mode in the CIA402 Object

dictionary, into 100500; The target speed 6081H is written into 33333;

Acceleration 6083H write 16666; Deceleration 6084H write 11111;

Step4: Communication sequentially writes 6040H control words in CIA402 Object Dictionary into 00h 06h 07h 0Fh 1F motor to start running, and the motor stops automatically when running to 100500 position.

2.4 6040H control word bit definition

Bit	0	1	2	3	4-6	7	8	9-15
Function	Enter servo ready	Enter the main circuit.	Perform Quick Stop	Execute enable run	Execute enable run	Fault Reset	Pause	NA

bit0 to bit3 have the same meaning in each servo mode. The control word 6040h sends commands in sequence according to the state transition requirements, guiding the servo to jump to the corresponding state to run related functions.

bit0~bit3 value	It indicates the state to be entered
0x0	no fault when entering the servo
0x6	Enter the servo and get ready
0x7	Enter the waiting enable
0xF	Enter the servo enable
0x2	Perform rapid shutdown

bit7 fault reset, bit8 operation pause, the trigger logic is that the rising edge is valid. The different operation modes of bit4 to bit6 have different functional meanings (please refer to the control command steps in Section 2.7).

It is meaningless to assign a value to each bit of a control word separately; it must be combined with other bits to form a certain control instruction (refer to 2.6 Control Instructions and State Transitions).

2.5 6041h status bit definition

Bit	0	1	2	3	4	5	6	7/8	9	10	11	12-13	14	15
Function	Ready	Wait enable	Run enabled	Fault occurred	Powered	Quick Stop Activation	Not Runable	NA	Remote Control	Target arrival	NA	Operating mode related	NA	Operating mode related

bit0 to bit9 have the same meaning in each servo mode. The servo feedback represents the current unique and definite operating state. The possible states are shown in the following table.

bit0~bit9 Value	It indicates the state to be entered
0x000	being initialized
0x250	no faults at present
0x231	servo is ready
0x233	Wait for the enable to be activated
0x237	servo has been enabled
0x217	Rapid shutdown in progress
0x218	servo is in a faulty state
0x21F	malfunctioning and shutting down

◆bit10 to bit15 Among them, bit10/bit12/bit13 have different meanings in each servo mode, bit11/bit15 have the same meaning in each servo mode, and bit14 is retained by the manufacturer as meaningless. As follows table

Bit	PP	PV	PT	HM	CSP	CSV	CST
bit10	Position reached	Velocity Matching	Torque reached	Position reached	Meaningless fixation 1	Meaningless fixation 1	Meaningless fixation 1
bit11	When the software limit or hardware limit is detected internally and triggered, this bit is 1. When the limit is lifted, this bit is 0						
bit12	Postition update allowed	Zero-speed standby	Meaningless fixation 1	reset to zero has been completed	Meaningless fixation 1	Meaningless fixation 1	Meaningless fixation 1
bit13	position follows the deviation	Meaningless fixation 0	Meaningless fixation 0	回零有错误	position follows the deviation	Meaningless fixation 0	Meaningless fixation 0
bit15	When the origin return is successfully executed, this bit is 1. If an alarm is generated or the disconnection is enabled, this bit is 0						

2.6 CIA402 Control Instructions and Status Transitions

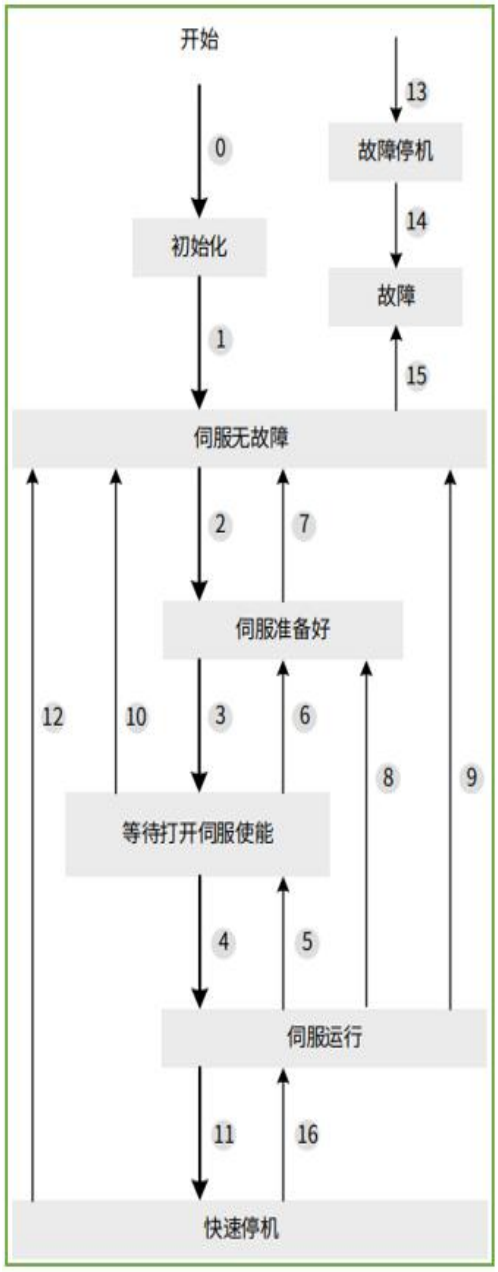
CIA402 State Transition Diagram/Table

CIA402 state switching		Control Word 6040	Status word6041
Label	**Process Migration**	**Control command**	**Feedbackbit0 -bit9**
0	Power-on Initialization	Internal automatic transition, no command required	0x0000
1	Initialize servo without fault	1. If a fault is detected at the end of initialization, enter directly 13, 2. If there is no fault at the end of initialization, automatic transition without command	0x0250

2	Servo No Fault Servo Ready	0x06	0x0231
3	Servo ready waiting to turn on servo enable	0x07	0x0233
4	Wait for open servo to enable servo to run	0x0F	0x0237
5	Servo Run Wait On Servo enable	0x07	0x0233
6	Wait for the open servo enable servo to be ready	0x06	0x0231
7	Servo Ready Servo No Fault	0x00	0x0250
8	Servo Run Servo Ready	0x06	0x0231
9	Servo operation No servo fault	0x00	0x0250
10	Wait open servo enable servo no fault	0x00	0x0250
11	Servo operation fast stop	0x02	0x0217
12	Fast stop servo without fault	Automatic transition after quick shutdown completes without command	0x0250
13	Fault shutdown	In addition to the fault state, in other states, once the driver detects the alarm, it will automatically switch to the fault stop state without command.	0x021F
14	Fault Shutdown Fault	Automatic transition after completion of fault shutdown without command	0x0218
15	Fault Servo No fault	0x80	0x0250
16	Fast stop servo operation	Send command 0x0F after Quick Stop is complete	0x0237

2.6 CIA402 Control Instruction and State Transition

CIA402 State Transition Chart/Table



Users must control the CANOPEN motor in accordance with the state transition table. Control word 6040 h every single assignment a bit pointless, must constitute a control instruction together with the other bits. Status word bit0 ~ 6041 h of combination and bit11

bit15 in servo any operation mode the same meaning. The status word 6041 h bit10, bit12, bit13 in servo different operation mode is not the same meaning.

2.7 6040H Control Command steps

Note: To use Ethercat control, we need to use our company's debugging AIMOTOR software to set the H02-00 control mode parameter to 9(EtherCAT control); Once it is set to Ethercat control, it must follow the CIA402 control requirements, its corresponding Object dictionary takes effect, and other control modes are invalid;

Sequence Control Steps		Step 0	Step1	Step 2	Step3	Step4	Step5	Step6	Step7
Mode	Control Word	Preparatory work	Initial	Get ready.	Etc enable	Enable	Start running	Displacement/speed	Normal Stop
CSP cyclic synchronization position mode	6040	Establish communication Assign PDO mapping Activate Master Cycle Synchronization	00h	06h	07h	0Fh	Master station issues absolute position command	Master station control	The master station stops issuing absolute position commands
PP contour Position mode	6040	Establish Communication Status Set motion parameters related to PP mode	00h	06h	07h	0Fh	Absolute Positioning Non-Immediate Update 0Fh→1Fh		Automatic stop without control command when position is reached
						2Fh	Absolute Positioning Update Now 2Fh→3Fh		
						4Fh	Relative positioning non-immediate update4Fh→5Fh		
						6Fh	Relative positioning non-immediate update6Fh→7Fh		
PV/CSV speed mode	6040	Establish Communication Status Set motion parameters related to PV mode	00h	06h	07h	0Fh	After enabling Start running	You can change the speed	Target Speed Object t Index 60FFh write 0
PT/CST torque mode	6040	Establish Communication Status Set the motion	00h	06h	07h	0Fh	After enabling Start running	Change the torque	Highest Speed Object t Index 607Fh write 0

		parameters related to PT mode							
HM back to Origin mode	6040	Establish Communication Status Set the motion parameter state related to HM mode.	00h	06h	07h	0Fh	0Fh 1Fh Start Origin Search	-	1: Origin to find automatic stop 2:1Fh 0Fh stop

CSP cyclic synchronous position mode, the main control unit sends the position according to the synchronous period, the motor performs displacement according to the absolute position point and synchronous period, and the contour curve of the motor operation (such as position, speed, etc.) is planned and calculated by the main control unit in advance;

In PP/PV/PT contour mode, the main control unit can first set the necessary parameters of the mode, and then control the motor operation according to the mode control command step. The contour curve of the motor operation is planned and calculated by the motor according to the set parameter values (such as acceleration and deceleration, operating speed, etc.).

In CSV synchronous speed mode, the main control unit independently plots the speed curve, and there is no acceleration or deceleration hysteresis at the motor end.

Note: abnormal stop mode during operation

Switching mode: Write the control character 6040h to 0h, 6h or 7h. The control state of CIA402 will no longer be in operation, and the motor will be de-enabled and stop freely according to inertia.

Quick stop way: the control word write 2 h, 6040 h motor will be in accordance with the 605 ah object index after choose the way of deceleration and stop state;

Deceleration suspend mode: write control word 6040 h 10 fh, motor will be in accordance with the 605 ah object index after choose the way of deceleration and stop state;

Quick outage and suspended shall not apply and the CSP, HM model, CSP model plan trajectory is controlled by the host. In HM mode, the control character 6040h writes from 1F to 0F and stops.

2.8 6098h Origin Search Method

Set point	Initial direction	Deceleration point	Origin	Process steps
24	Forward	Origin Switch	Origin Switch	The motor first searches the origin switch at high speed in the set direction, and begins to slow down when the origin switch signal rises along the edge Run disengaged, disengaged to the origin switch signal drops along the motor commutation and continues to search for the origin switch signal at low speed When the origin switch signal is encountered, the rise edge will stop immediately, and the return to zero will be successful.
28	Reverse	Origin Switch	Origin Switch	
34	Forward	Z signal	Motor Z signal	The Motor first searches for phase Z at high speed in the set direction, starts low-speed commutation operation when encountering the rising edge of phase Z signal of the motor, stops immediately when encountering the rising edge signal on the other side of phase Z, and returns to zero successfully.
33	Reverse	Z signal	Motor Z signal	
4	Forward	Origin Switch	Motor Z signal	The Motor first searches the Origin switch at high speed in the set direction. When it encounters the rising edge of the Origin switch signal, it starts to switch off at low speed. When it disengages from the falling edge of the Origin switch signal, it switches again to the rising edge of the low speed search origin switch signal. When it encounters the rising edge of the Origin switch signal, it runs until the Z phase signal stops immediately and returns to zero successfully.
6	Reverse	Origin Switch	Motor Z signal	
18	Forward	Forward overpass	Forward overpass	The Motor first searches the Origin switch at high speed in the set direction. When it encounters the rising edge of the forward overpass switch signal, it starts to reverse and disengage at low speed. When it disengages from the falling edge of the forward overpass switch signal, the motor reverses again to search for the rising edge of the forward overpass switch signal at low speed. When it encounters the rising edge of the forward overpass switch signal, it stops immediately and returns to zero successfully.
17	Reverse	Reverse overpass	Reverse overpass	
2	Forward	Forward overpass	Z signal	The motor first searches for the forward over range switch at a high speed in the set direction and encounters the rising edge of the forward over range switch signal Start to slow down the reversing operation to disengage, disengage to the forward over range switch signal drop edge continue to run to the motor Z phase Signal stops immediately. Return to zero.
1	Reverse	Reverse overpass	Z signal	
37	Forward	Mechanical limit position	Mechanical limit position	First, the motor searches for the mechanical limit position at a low speed with the set direction and torque (torque set by H05_58). When the mechanical limit position is against the locked rotor and the torque reaches the upper limit of the zero torque limit and keeps the default time, the motor stops immediately and returns to zero successfully.
38	Reverse	Mechanical limit position	Mechanical limit position	
35	-	-	-	Set the current position as the origin

NOTE:

If there is an external switch signal used in the Origin search mode, please associate it with the corresponding DI function option selected, otherwise the alarm ER.601 will fail to return to zero;

To change the search method of origin, please modify the Object parameters of 6098H in the mode of origin when the motor is deactivated;

In the selected mode, if the deceleration point is the origin switch and is associated with the positive and negative overrun switch DI, the motor will automatically change direction to continue searching when the motor encounters the overrun switch in the search path;

In the process step, the high-speed search speed is set 6099 the sub-index 1 Object t of the index, and the low-speed search speed is set by the sub-index 2 of the 6099 index;

If the origin is not found within the H05_35 parameter time, an alarm ER.601 Return to Zero timeout will occur;

After the return of the origin is successful, the DO function (OutFun16-HomeAttain) returns the zero origin to complete the output is valid, and the zero origin to complete the output is invalid after the OFF is enabled;

2.9 The polarity of the 607Eh command

Bit	0	1	2	3	4	5	6	7
Function	NA	NA	NA	NA	NA	Torque command Polarity	Speed command Polarity	Position Command Polarity

Bit7 = 1, indicating that in PP contour position mode, the position command 607Ah ×(-1) and the motor direction are reversed.

Bit6 = 1, indicating that in PV contour speed mode, the speed command is 60FFh ×(-1) and the motor is turned in reverse.

Bit5 = 1, indicating that in PT contour torque mode, the torque command 6071H ×(-1) and the motor steering are reversed.

NA: means no definition

2.10 60FDh Digital Input Status Monitoring

BIT	0-15	16	17	18	19	20	21	22	23-31
Physical input terminal monitoring	Retention	DI1	DI2	DI3	DI4	DI5	DI6	DI7	Retention

2.11 60FEh Digital Output Status Monitoring and Forcing

BIT	0	1-15	16	17	18	19	20	21-31
Physical output terminal	Brake holding brake	Retention	DO1	DO2	DO3	DO4	DO5	Retention
60FE-01h Status Monitoring & Bit Force Control	1. When the bit corresponding to 60FE-02h is set to 0, the bit state of 60FE-01h is determined by the internal state, and the user can monitor the current state of the bit; 2. When the bit corresponding to 60FE-02h is set to 1, the current output state of the bit of 60FE-01h is no longer determined by the internal state. Moreover, the user can force write the bit of 60FE-01h, write 0 to force the output of the physical port to be OFF; 3. bit0 bit is a special control bit for the brake brake motor. When the motor is enabled to run, this bit is invalid for the user to force the shutdown. Only after the enable operation is released, the forced write 1 can be turned on or write 0 can be turned off;							
60FE-02h force control bit Select Enable								

Third. Communication Configuration Example

3.1 Adapt to the operation case of Beifu controller

The following takes TwinCAT3 of Beifu company as the main station as an example to demonstrate the simple configuration and use process of our AIMortor servo driver.

3.1.1 Foundation construction

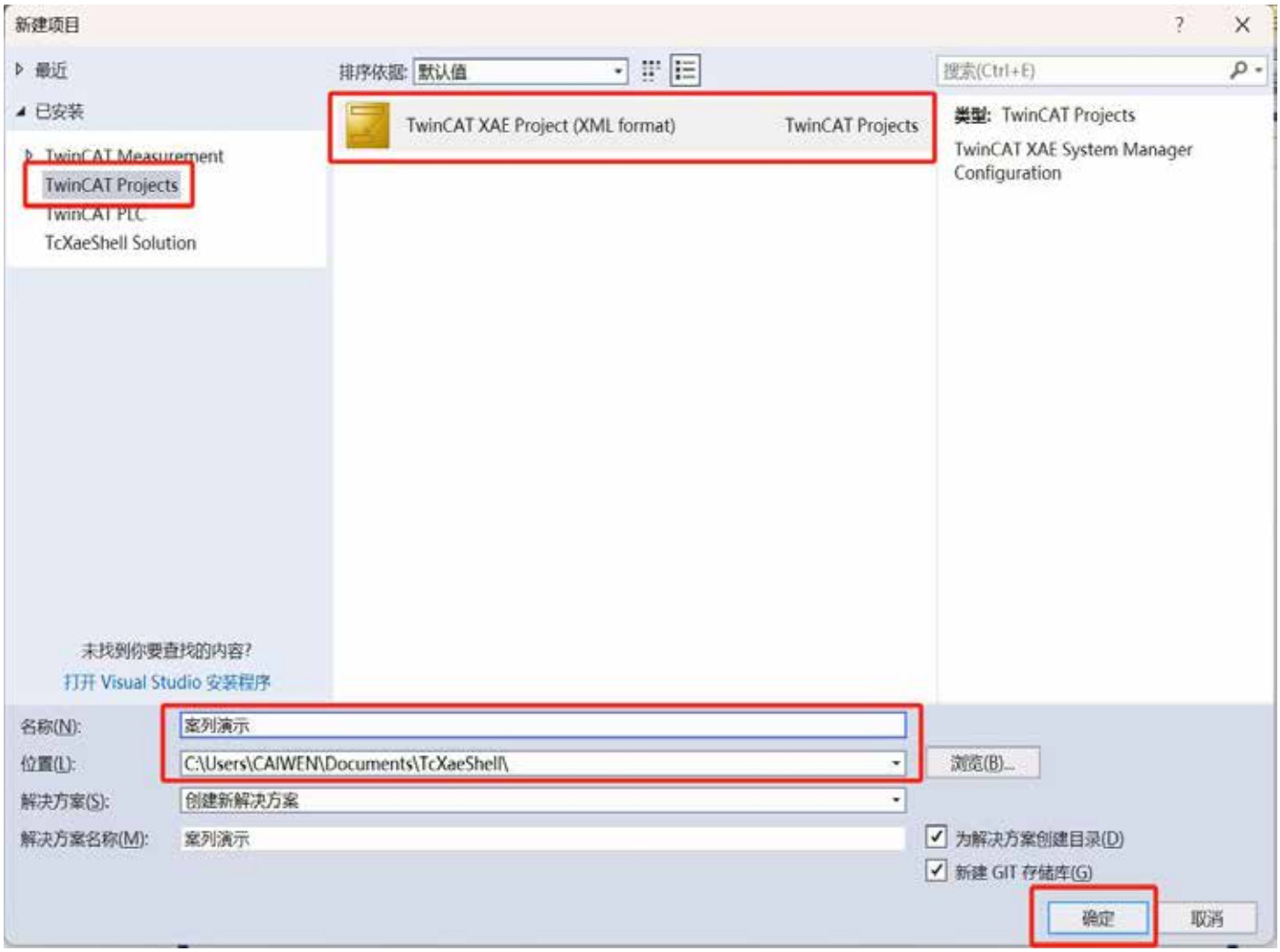
1) First of all, users need to go to Beifu official website or other channels to obtain Beifu TWINCAT3 software and install it correctly on PC computer. The PC network card must choose a 100-megabit Ethernet card with intel chip. There is a risk that the network cards of other brands do not support the operation of Ethercat.

2) Copy The Ethercat configuration file (SH_AIMotor_ECAT.xml) of our motor products to the TwinCAT installation directory: TwinCAT\3.1\Config\Io\EtherCAT.

3) our Ethercat product is powered on correctly, and the RJ45 of the PC network card of Ethercat is inserted into the network cable and connected to the RJ45 network port of the Ethercat-IN of the product (it is recommended to use at least more than 5 types of network cables)

3.1.2 Software Configuration

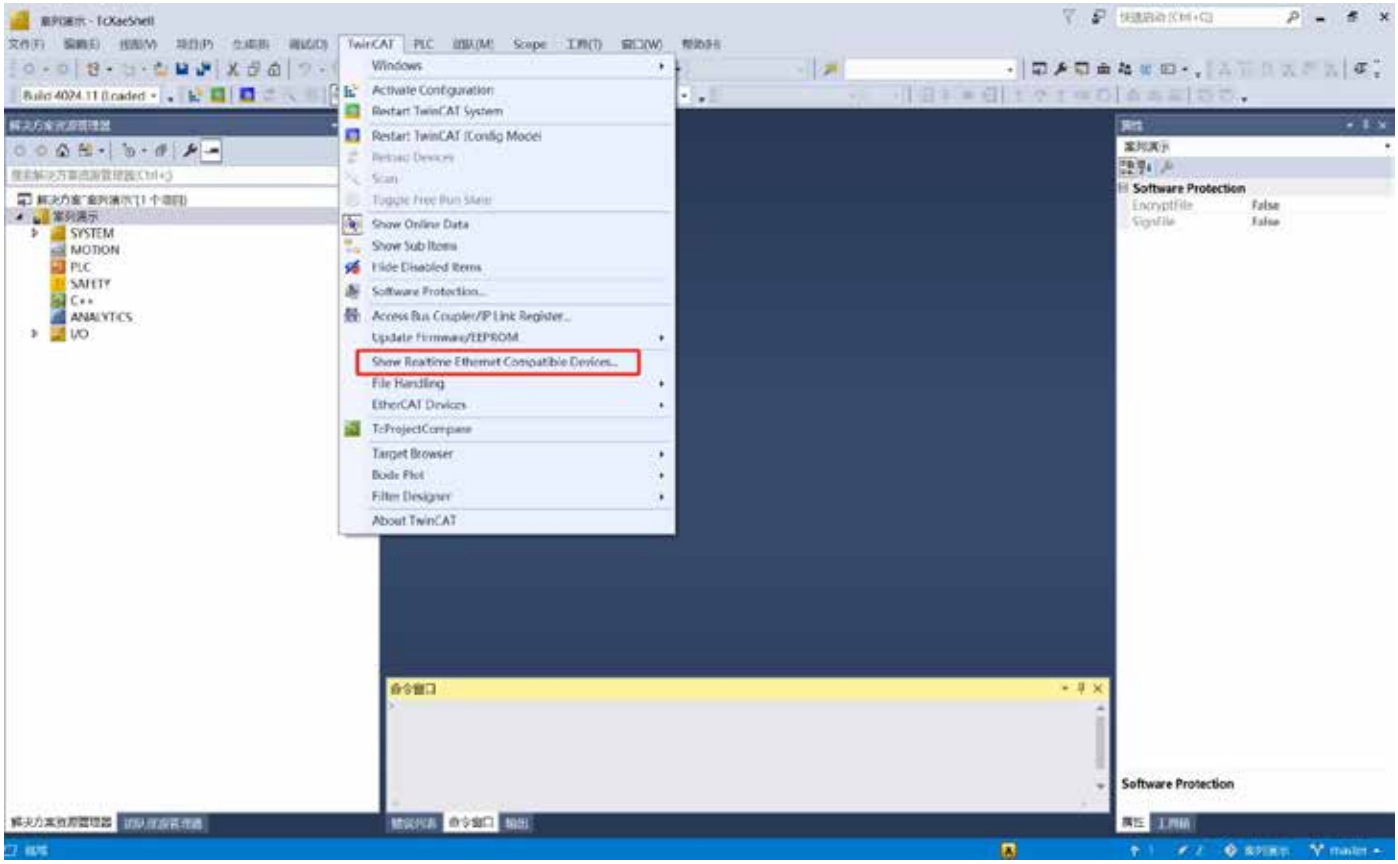
1) Open the software-new project, select the save path and project name as needed

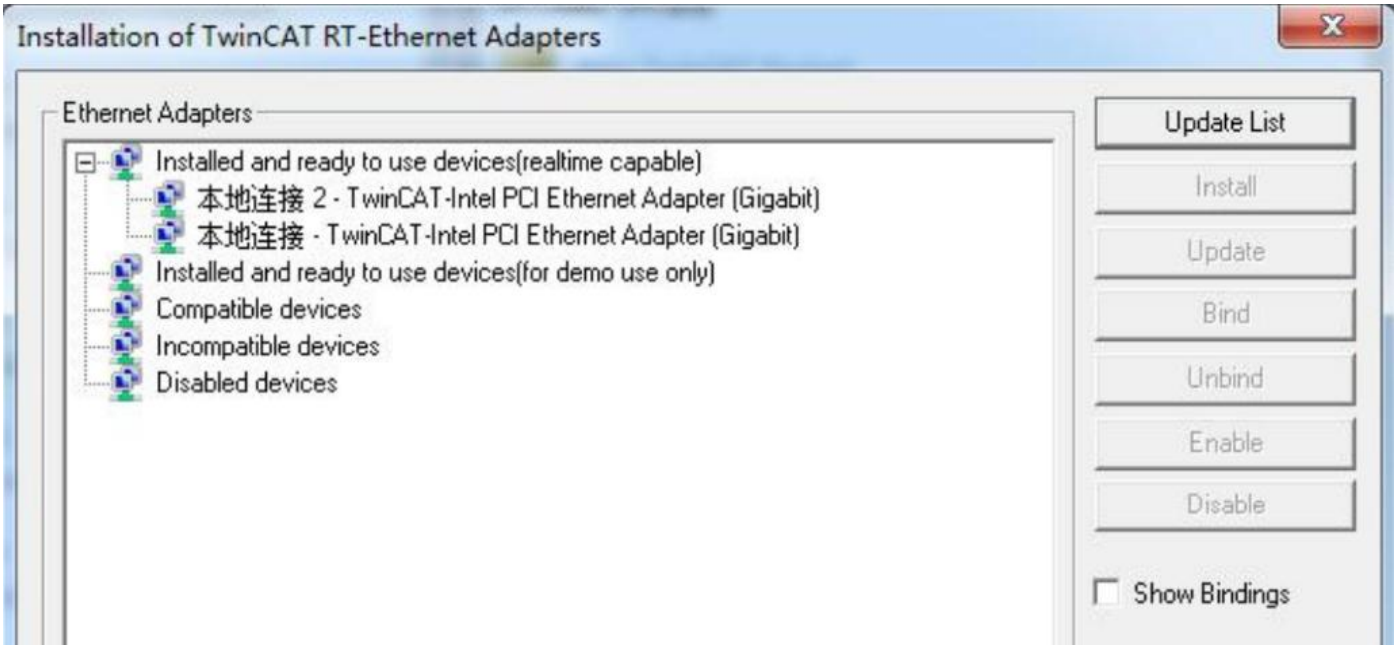


2) Install the network card (this step can be omitted if it is already installed) and click twincat->Show Realtime Ethernet Compatible Devices.

In the dialog box that appears, find the computer's intel network card under Compatible Devices, click the network card, and then click install on the right to install the real-time driver.

(The specific method can refer to Baidu or query the official documents of Beifu). The situation after installing the real-time driver is shown in the following figure

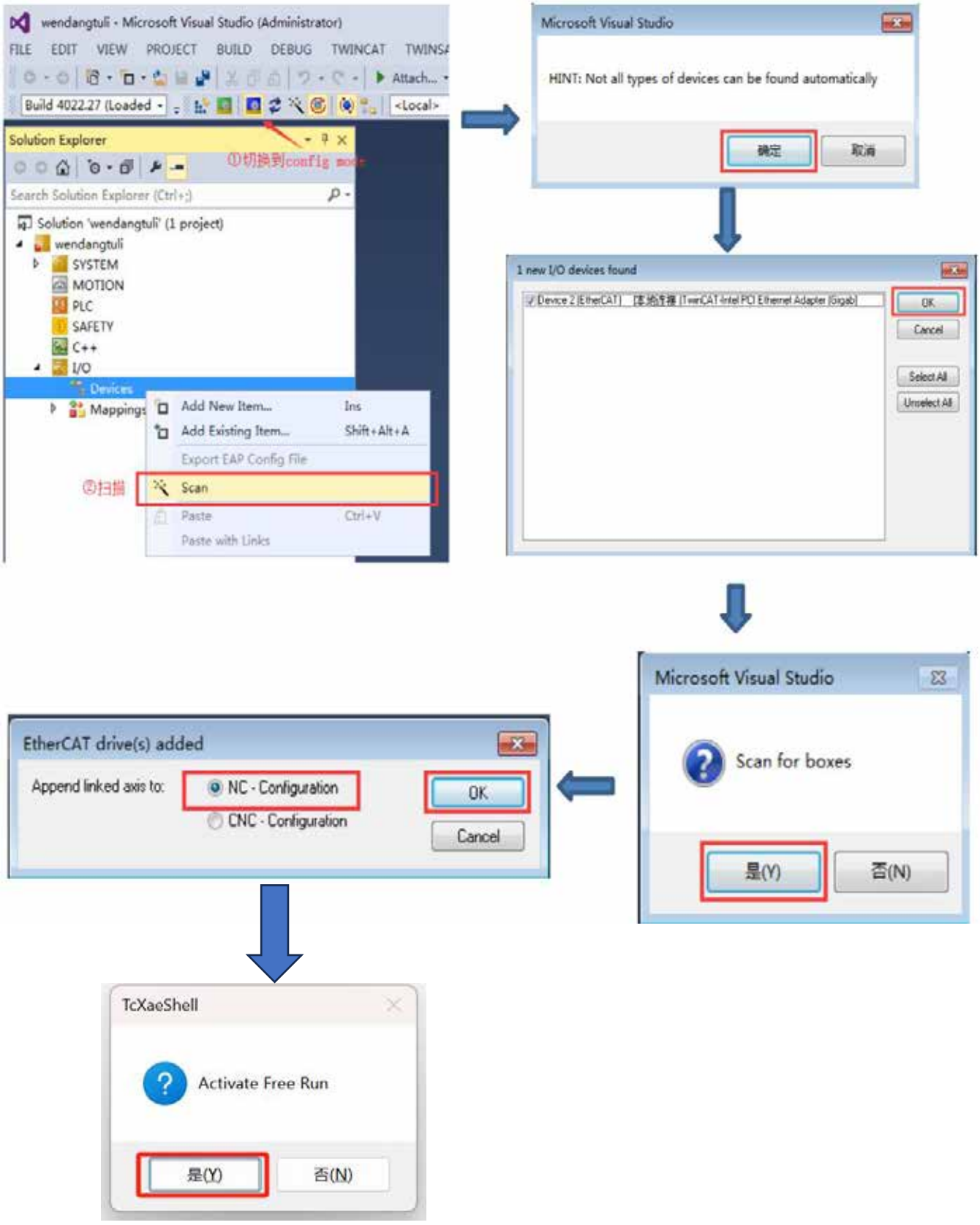




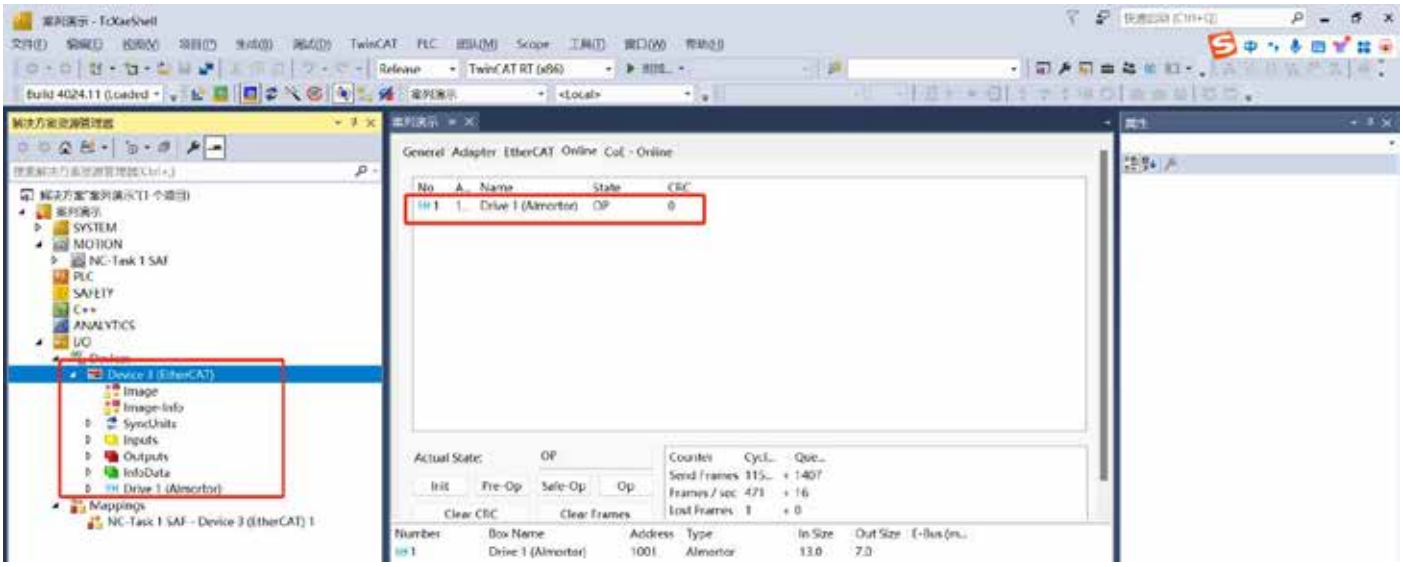
3) Select the hardware scan, 

right-click, 

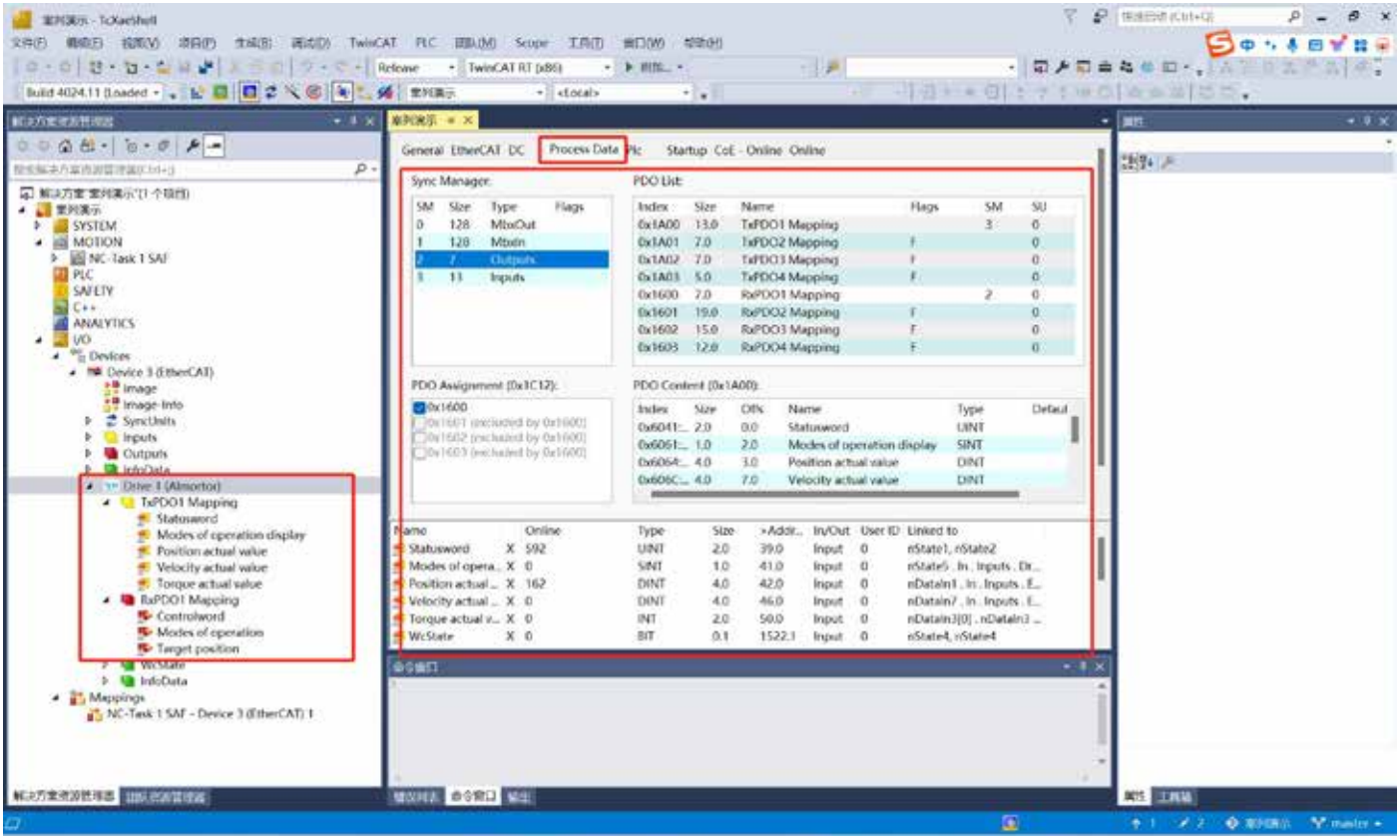
and then click Start Scan. Follow the process shown in the following picture and wait for the scan to be successful to activate free operation. If the hardware device is not scanned, please check whether the network card driver or network cable connection is correct.



4) After the scanning is completed, an NC axis Axis1 can be seen in the MOTION, corresponding to the servo-connected motor, and the scanned Drive 1 (AImotor) can be seen in the Device.

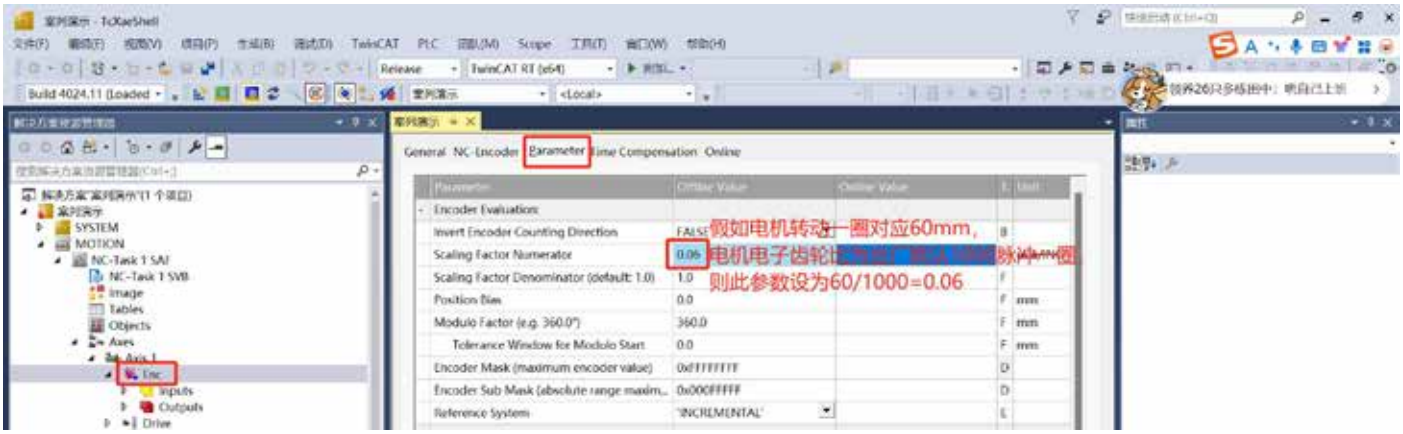


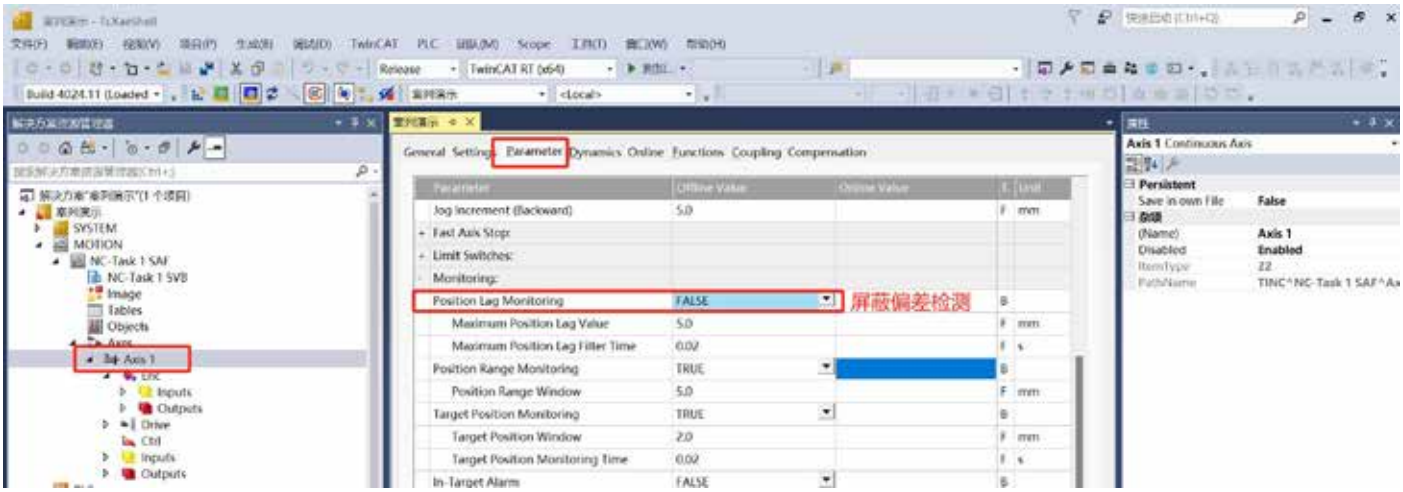
5) configure PDO (on-demand configuration, this case uses the default, do not change)



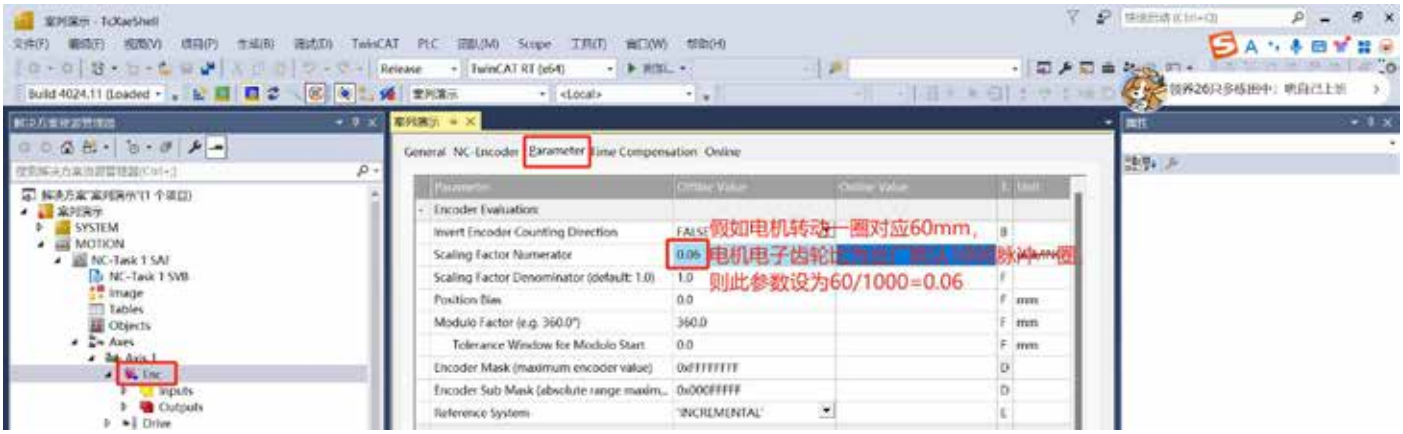
6) Link NC Axis and physical axis. You can select the physical Axis associated with NC axis by Axis1-Settings-Link to. This link will be added automatically when scanning hardware. You can also manually right-click Axis and click Append axis to add Axis to manually link NC axis to physical Axis. This window can see the corresponding relationship between NC axis and physical Axis.

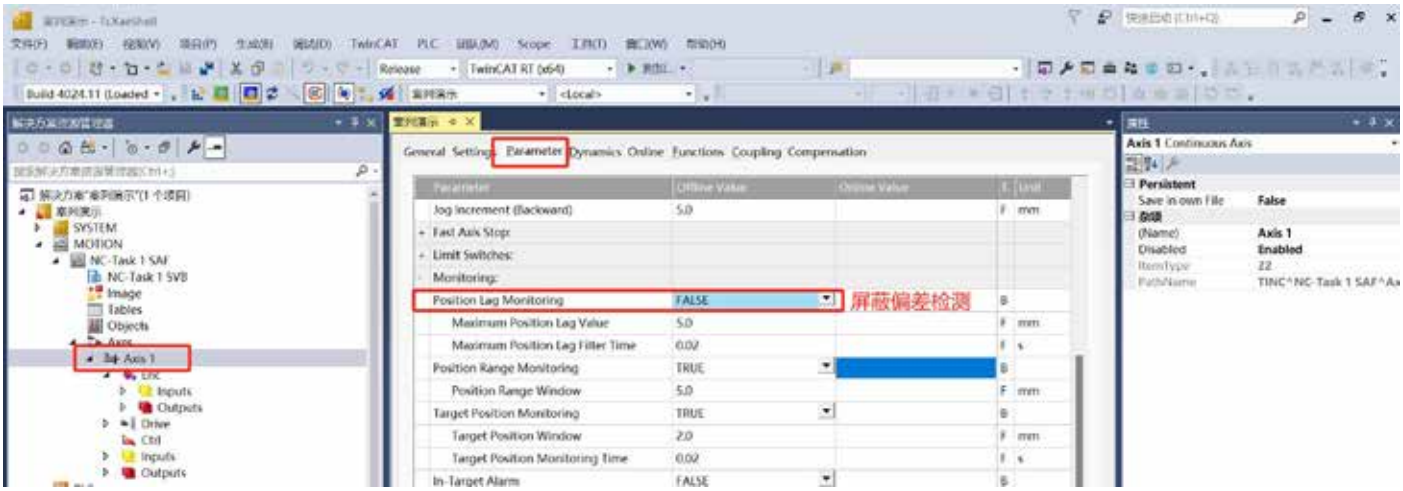
7) Set the necessary basic parameters





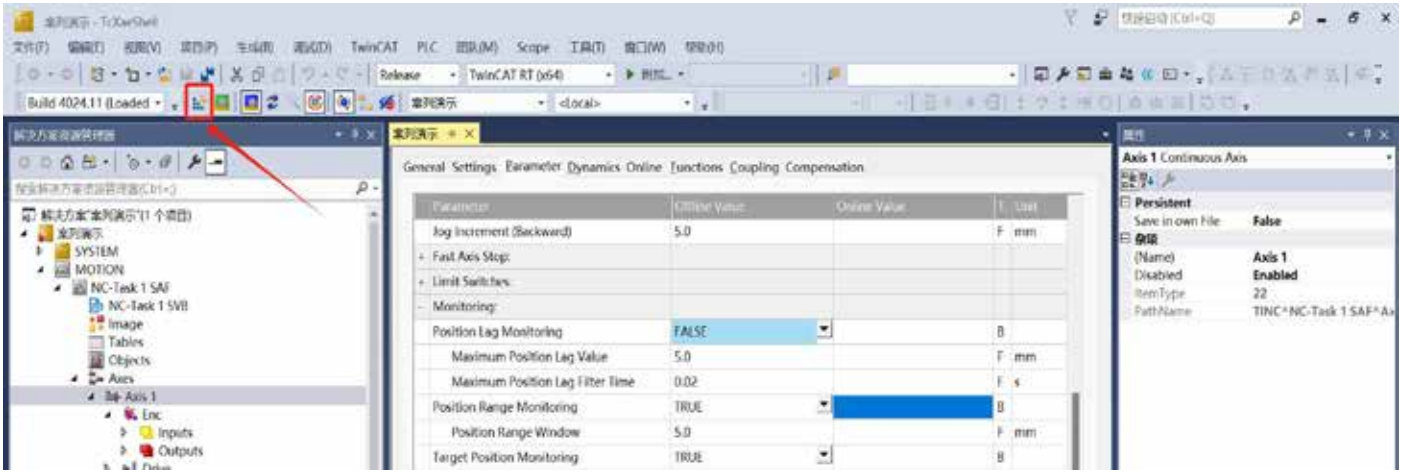
7) Set the necessary basic parameters





3.1.3Activation Configuration

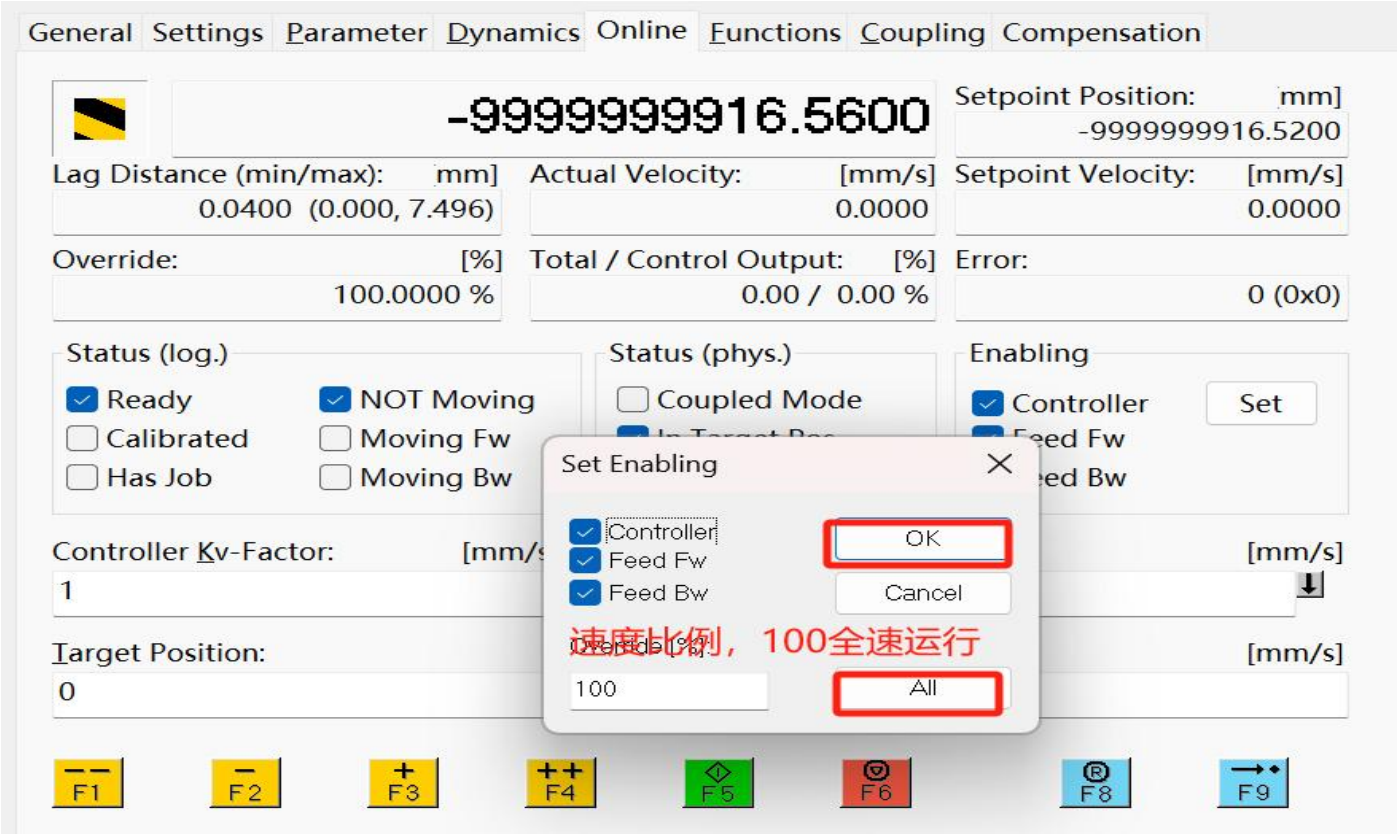
1) Activate the above configuration information



2) Enter the test page

3) Motor enable, configuration start

In the pop-up setting window, click the "All" check box, set the speed scaling ratio as required, and finally click "OK" to enable the motor,



3.1.4 Test Run

Then carry out the F1-F8 function test according to the following figure. If F9 is operated by mistake, the position of the motor and system will jump to -99999999 to move slowly. You can click F6 to stop the movement. If the speed is abnormal, please check the relevant reference of the above software configuration, or reduce the speed ratio.

If there are any omissions or more debugging functions, please refer to relevant materials such as "TC3 training materials V1.1.0"

3.2 Adapted to Huichuan H5U,Easy Series Controller Operation Case

The following uses Huichuan H5u,Easy series PLC as Ethercat master station and "AutoShop" programming software to demonstrate the simple configuration and use process of our company's AIMortor servo driver.

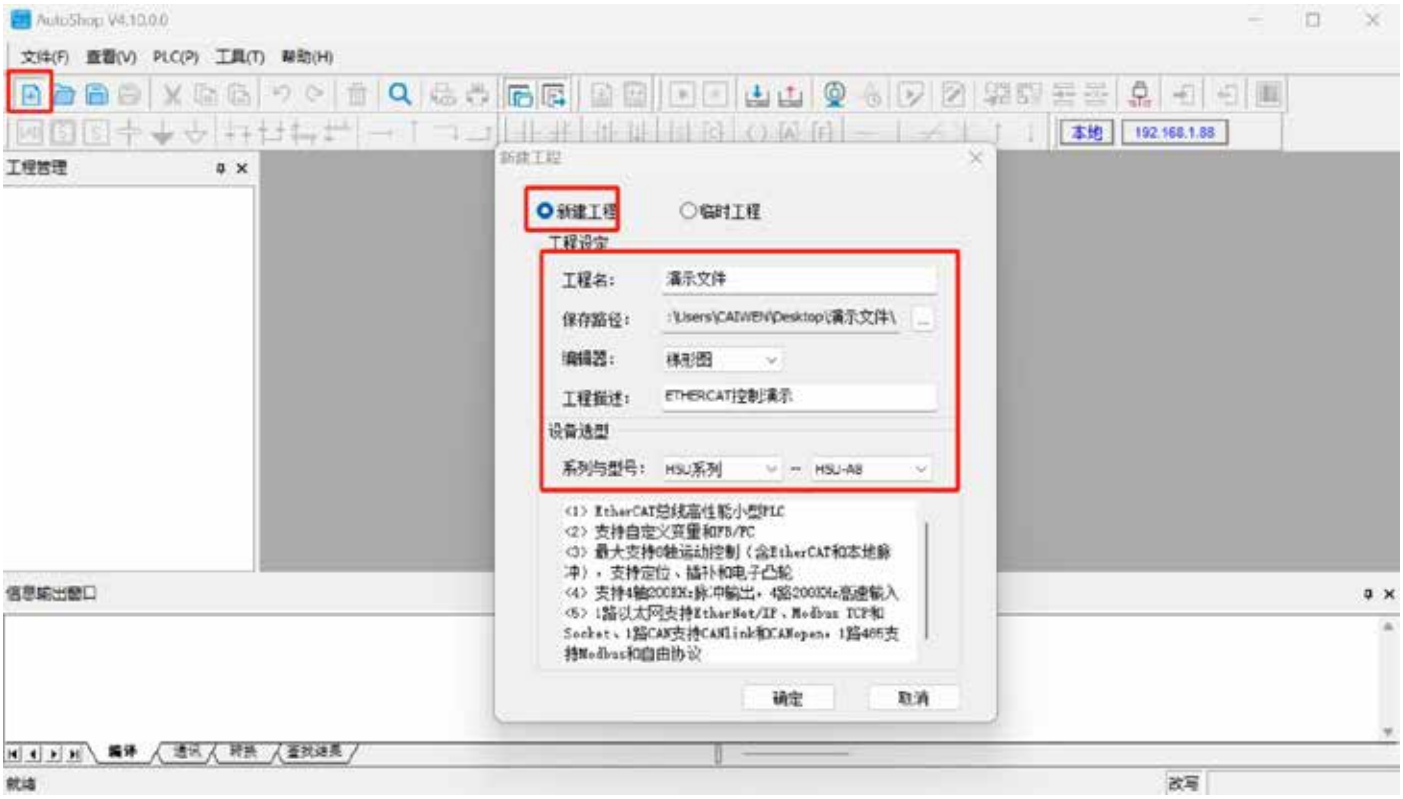
3.2.1 Foundation construction

1) First of all, users need to have Huichuan H5u,Easy series PLC hardware and install "AutoShop" programming software.

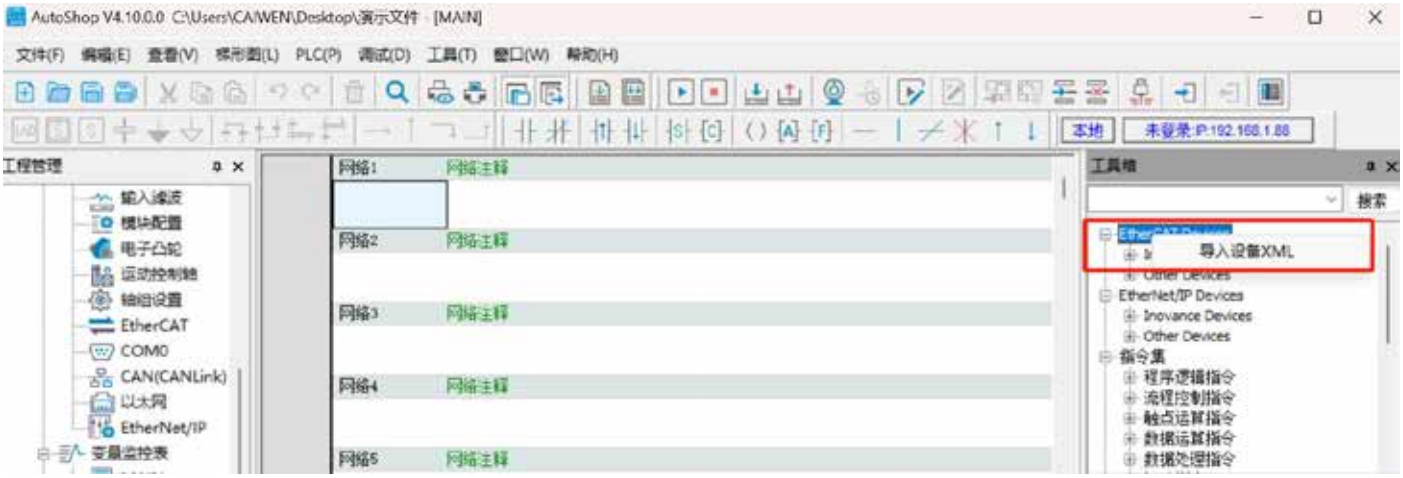
2) our Ethercat products and PLC products are powered on correctly, and the RJ45 network port of Ethercat used for PLC is inserted into the network cable and connected to the RJ45 network port of the Ethercat-IN of the product (it is recommended to use at least more than 5 types of network cables).

3.2.2 Software Configuration

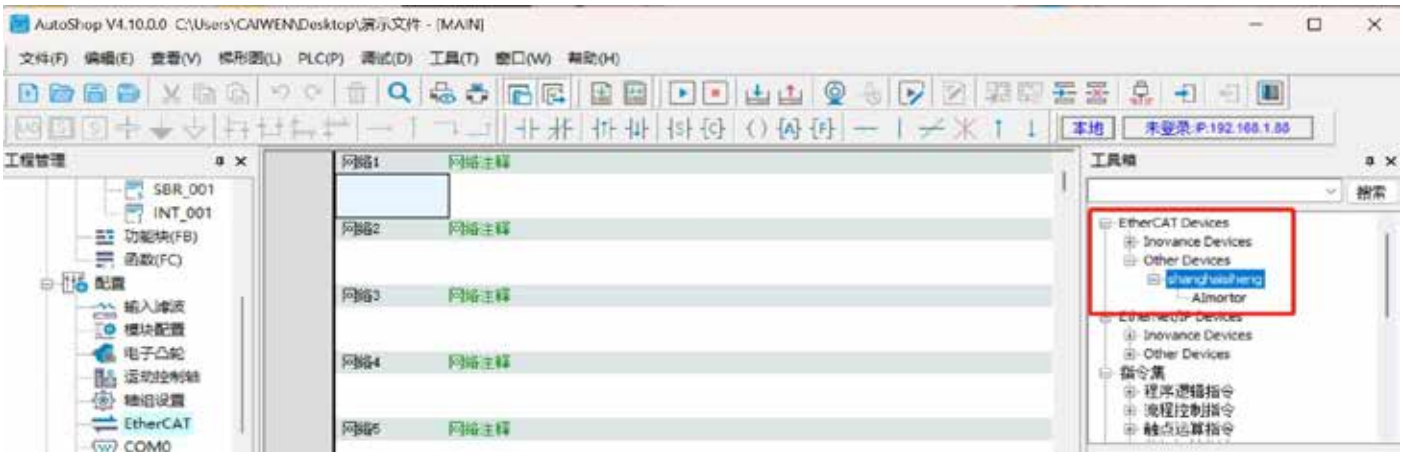
1) New construction



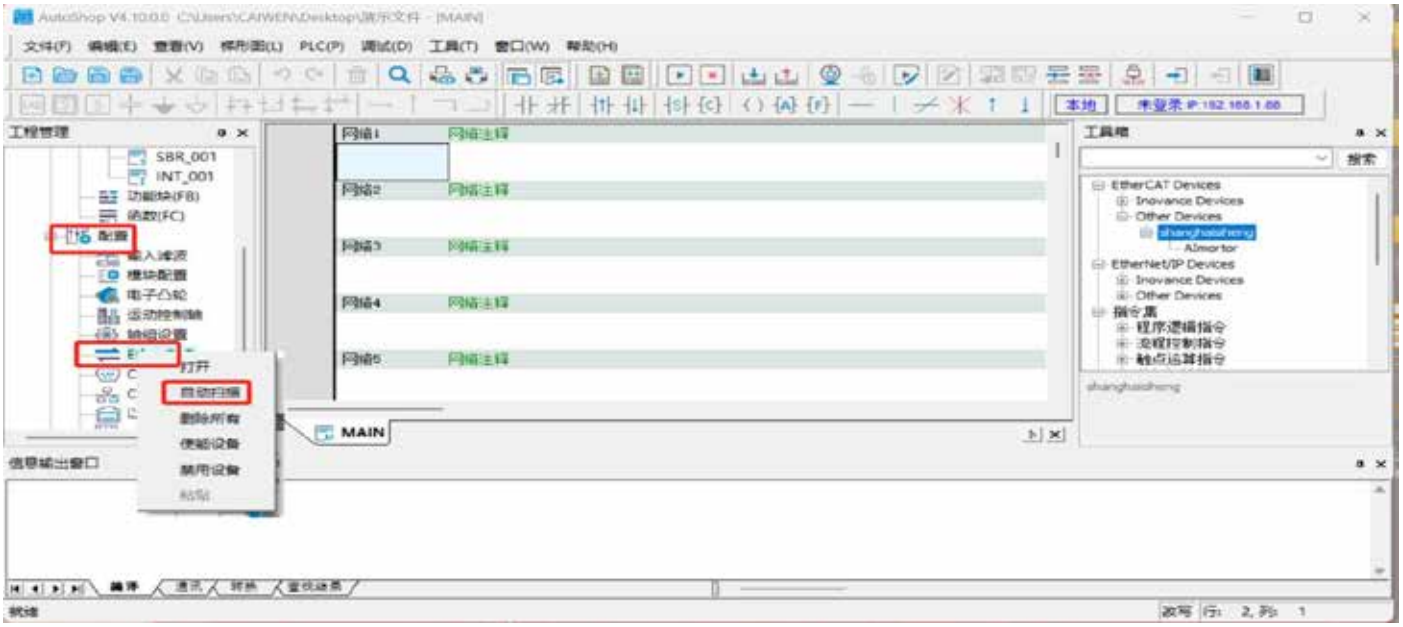
2) Import our product XML file Ethercat configuration file of our motor product (SH_AIMotor_ECAT.xml)

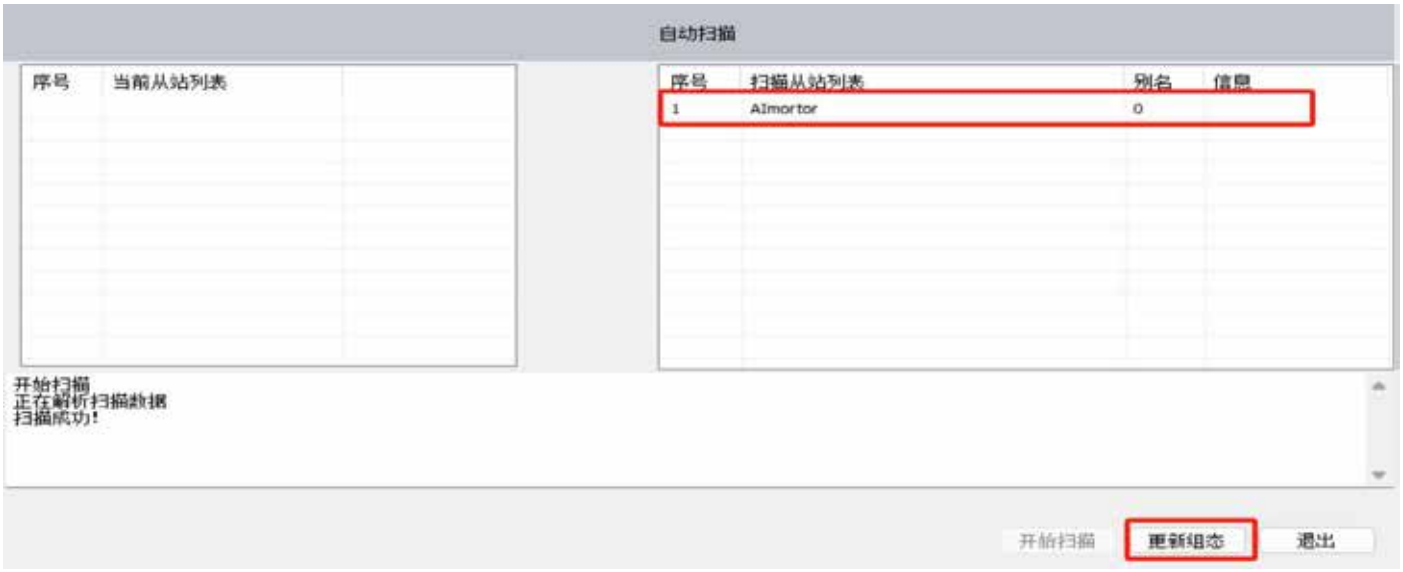


3) After the successful import, the software will restart and display the AImtor driver of our Ethercat product.



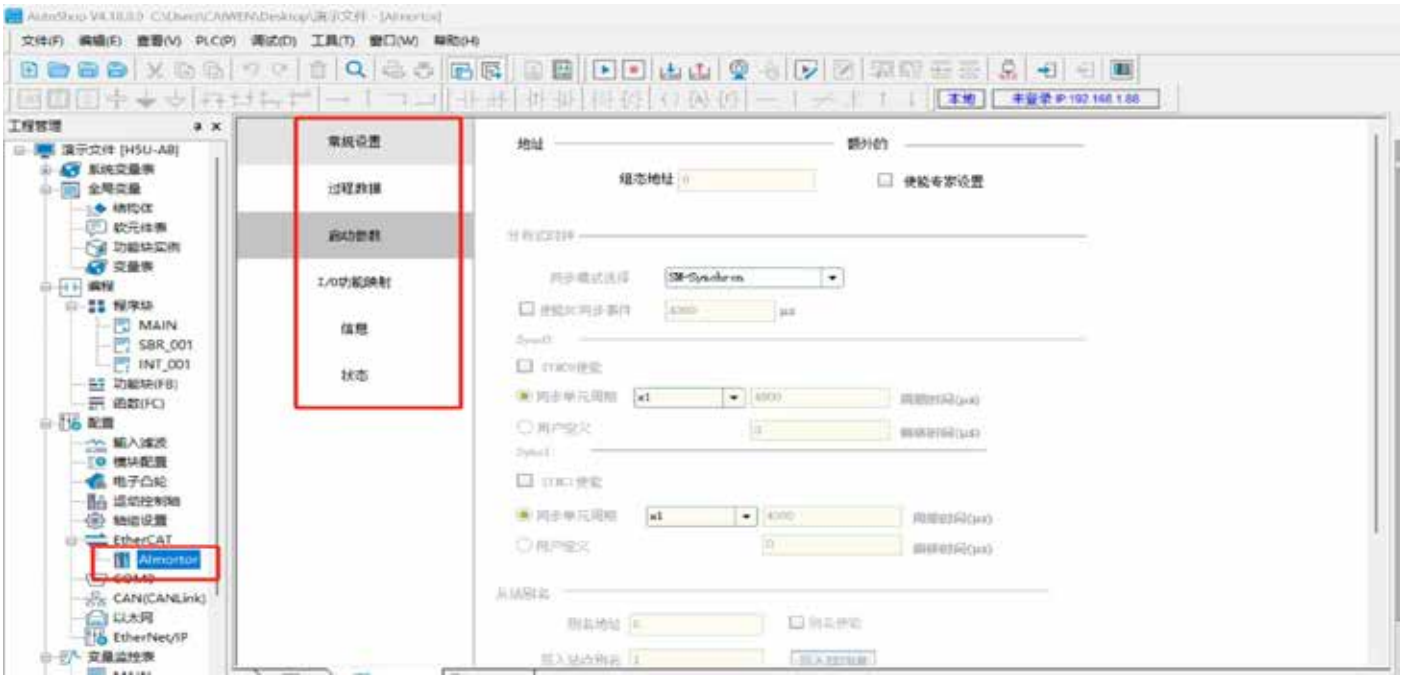
4) the programming software and PLC has been successfully connected to the case of automatic scanning, identification of Ethercat on the network from the device. If the power is correctly connected to the network cable, the existing equipment will be identified and the configuration will be updated, otherwise the wiring and hardware will be checked.



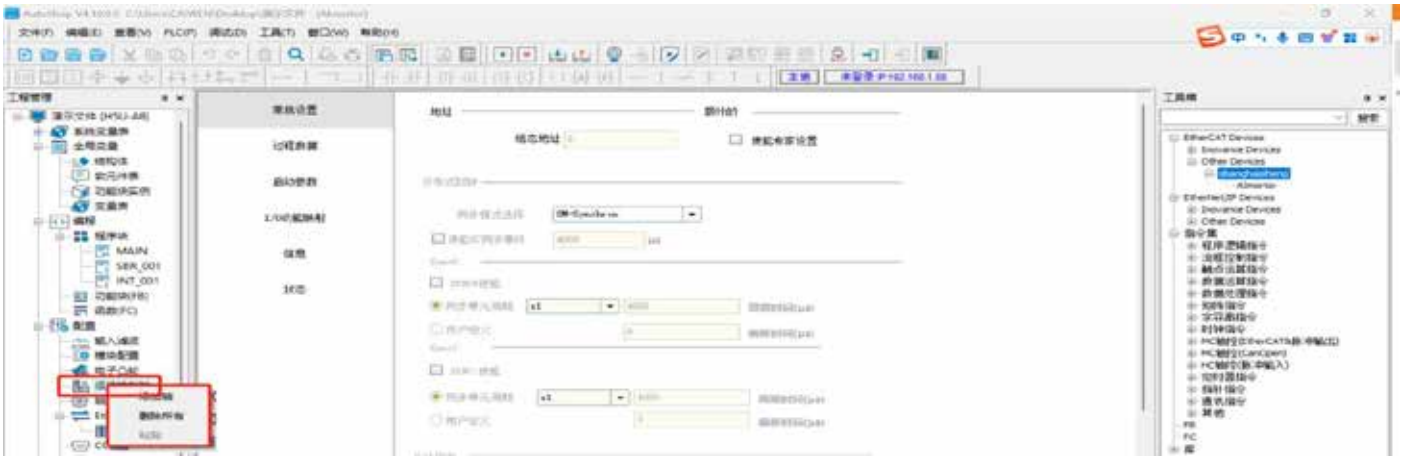


5) Configure the communication cycle time of the master station. The cycle time setting must be greater than 1000us. When there are many slave stations, it is recommended to enlarge by integer times (the default is used in this case)

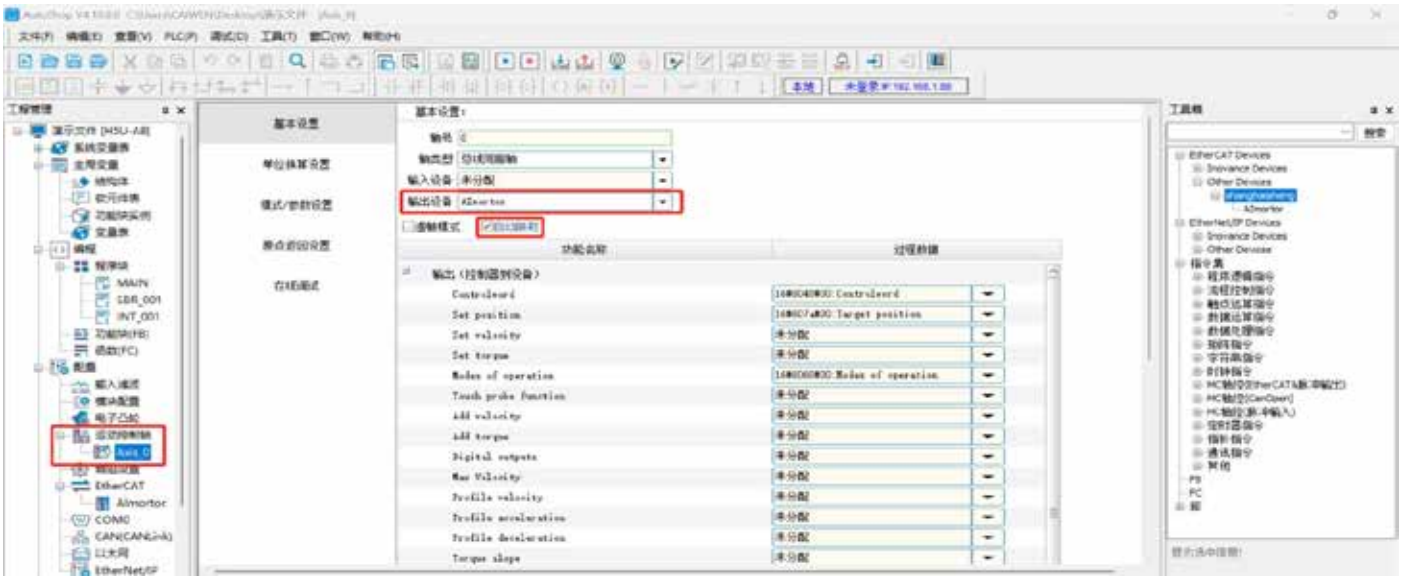
6) Configure slave-related (on-demand configuration, this case uses the default)



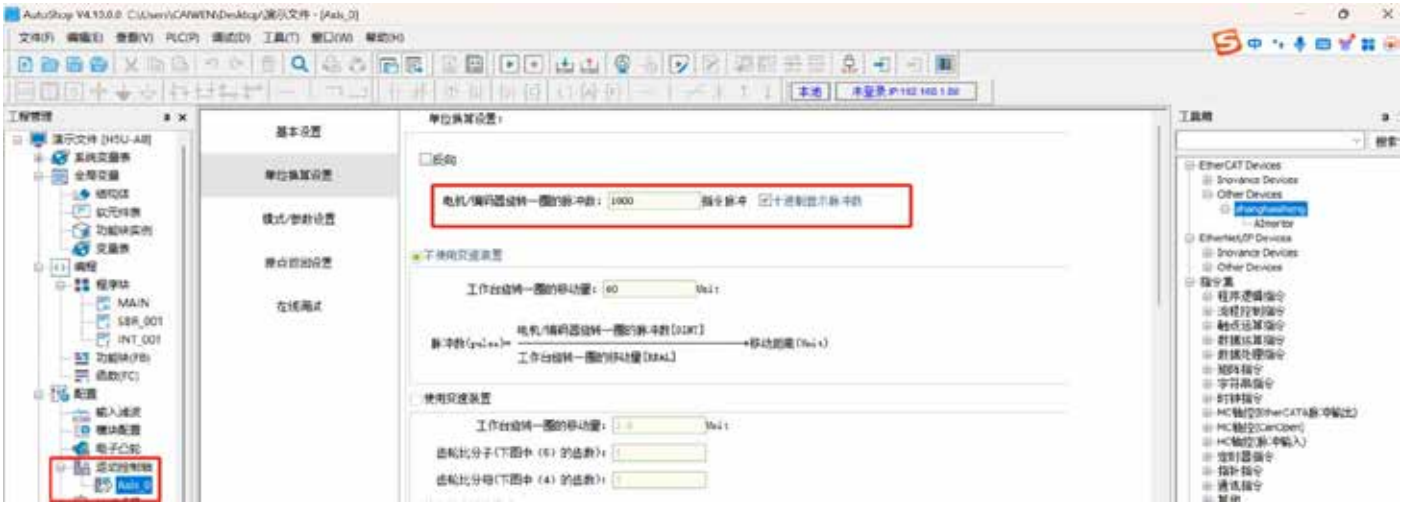
7) Add motion control axis



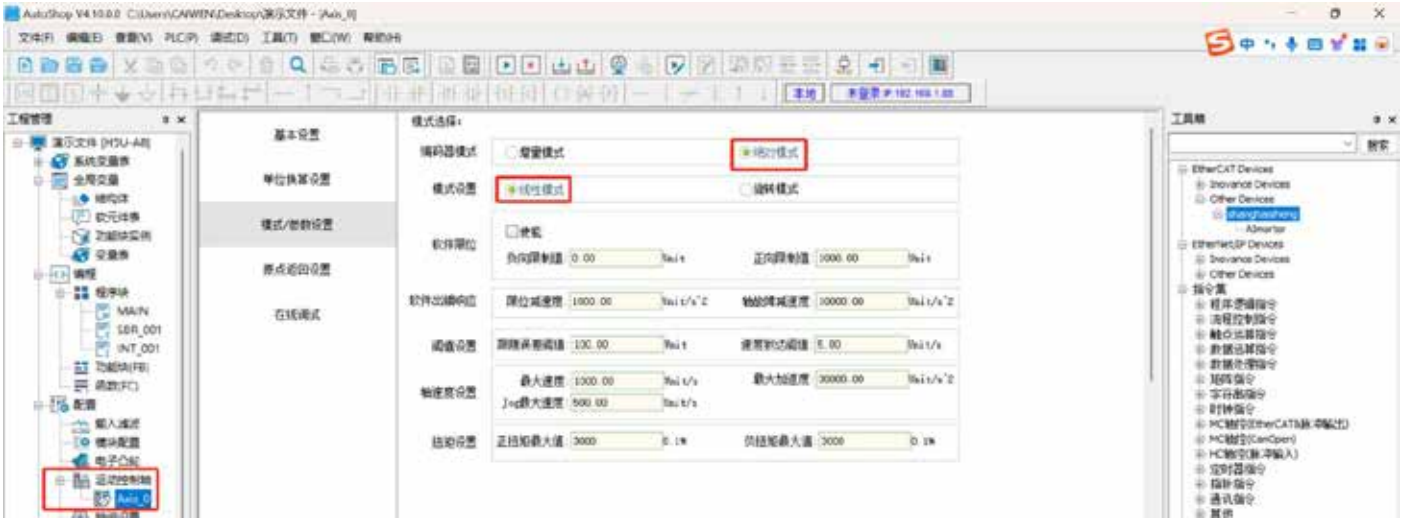
8) The motion control axis is associated with the slave station: the output device selects the scanned slave station.



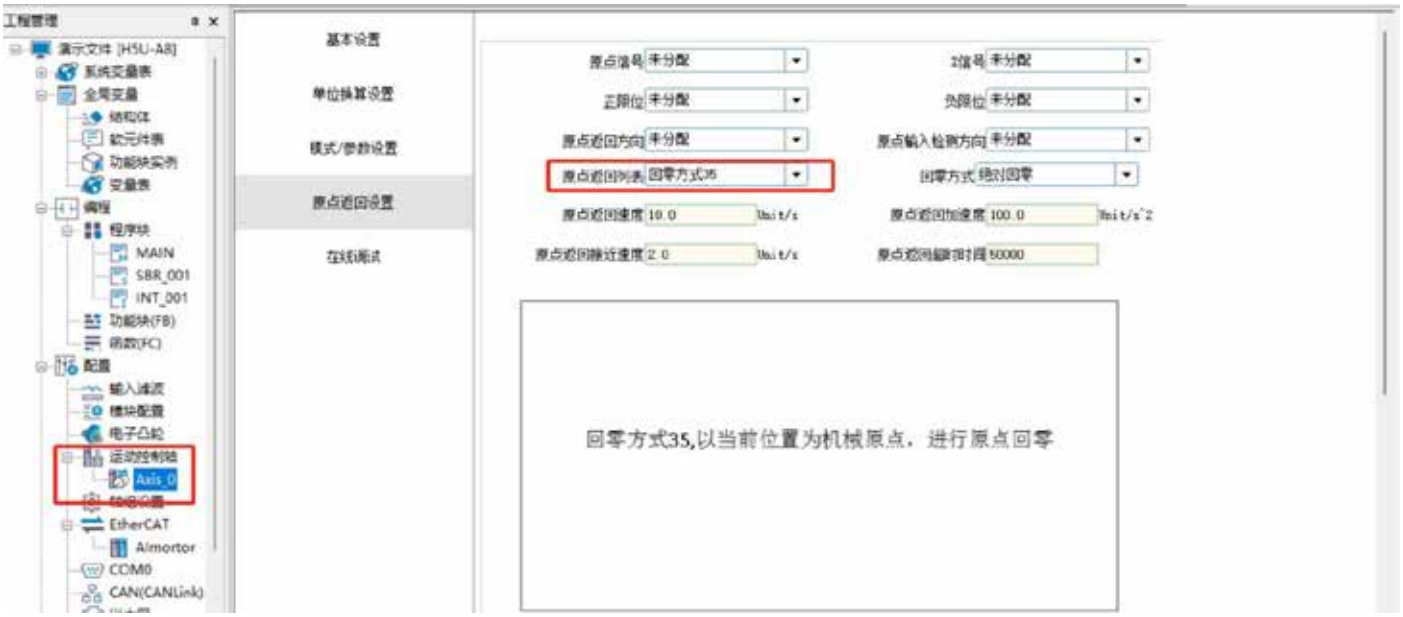
9) Set movement unit conversion: Our products leave the factory with a default 1000 pulse of one turn, and here set a movement command of 60 units corresponding to 1000 pulse (1 turn) without a deceleration mechanism.



10) Encoder selection: our encoder defaults to absolute value encoder (other on-demand configuration, this case uses the default).



11) Set back to zero mode: This case uses zero preset mode, the user needs to configure



3.2.3 Online debugging

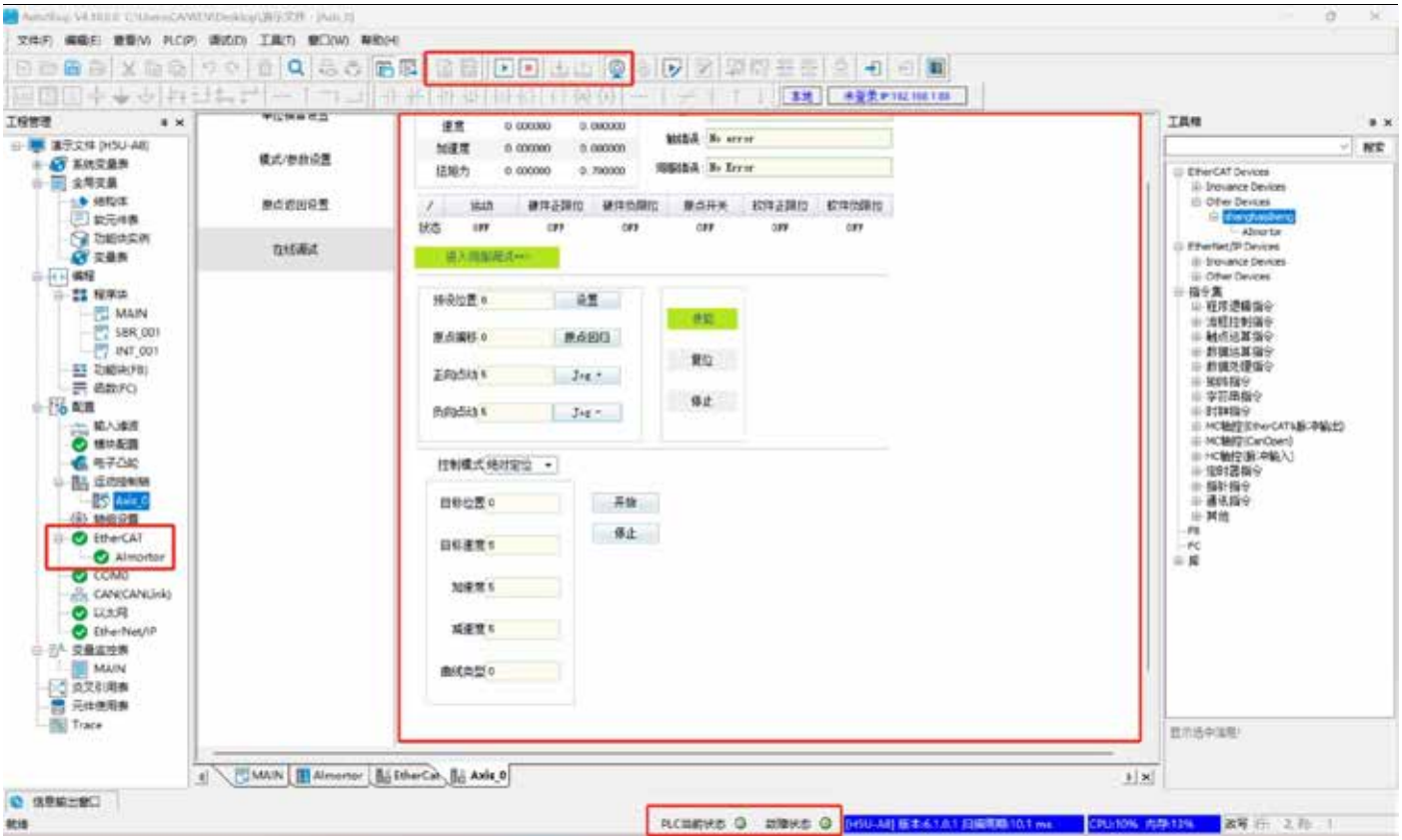
1) The above configuration will be compiled and downloaded into the PLC, and then enter the online monitoring mode.

2) Observe whether the Ethercat master and slave stations on the left side of the software page are started correctly, and whether the current status and fault status of the PLC at the bottom of the software are normal.

If the current state and fault state of the PLC are abnormal, the PLC will be powered off and restarted or switched to the stop state and then switched back to the running state to reinitialize the PLC.

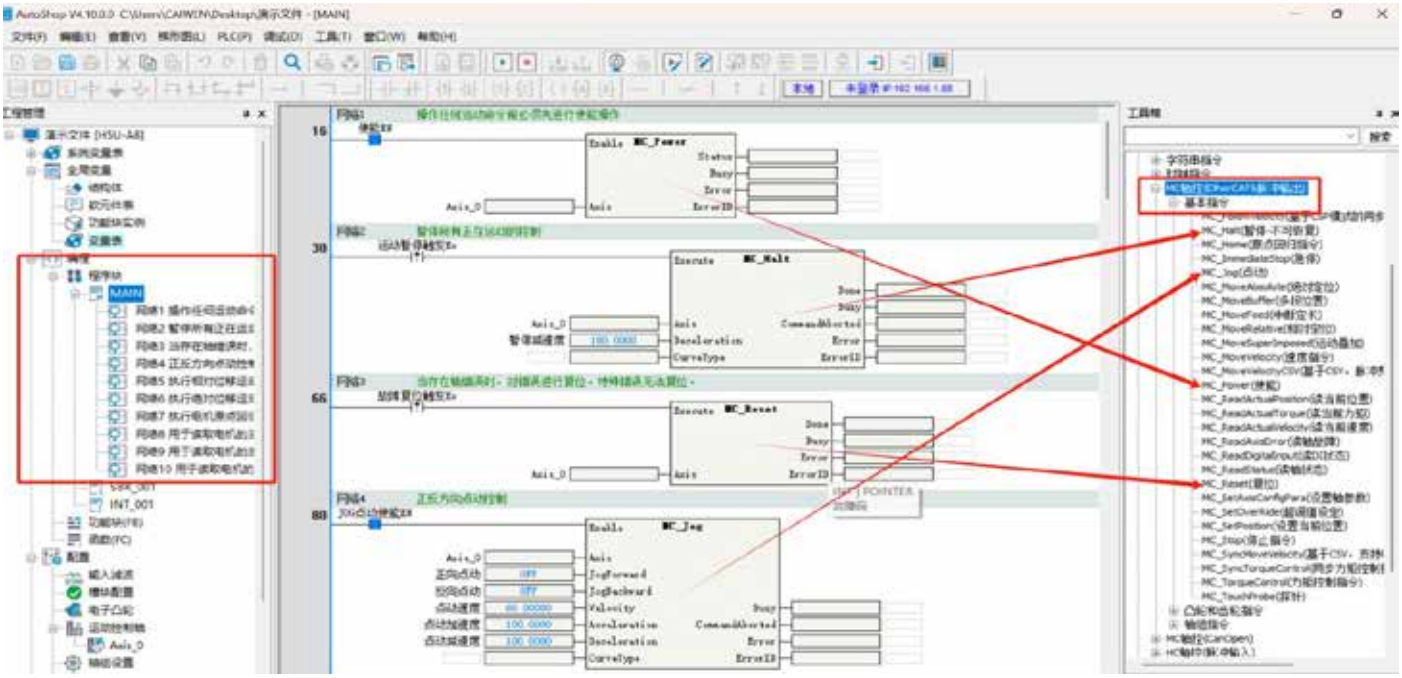
If The Ethercat master and slave stations do not start correctly, power off the slave device and restart it or check the network wiring.

Use the software quick debugging window to debug and test (refer to the relevant PLC manuals of Huichuan Company for detailed usage methods)



3.2.4 Programming Control

Program reference, in order to commonly used several motion command programming demonstration, the actual use according to the process requirements of the application.After writing the program offline, compile through, download. The status at the bottom of the observation window shows that the communication between PLC and Ethercat is normal. A control test may be performed.



3.3 Operation Case of Controller Adapted to CodeSYS Platform

The following uses the PLC or controller of CodeSYS kernel as The Ethercat master station, such as Huichuan AM400,AM600 series PLC. Xinjie's XS3 series PLC and Hexin's C36 series PLC demonstrate the simple configuration and use process of our AIMortor servo driver.

There may be differences between each CodeSYS software version, but the use is similar. The case will use CodeSYS V3.5 SP4 version as the demonstration development software.

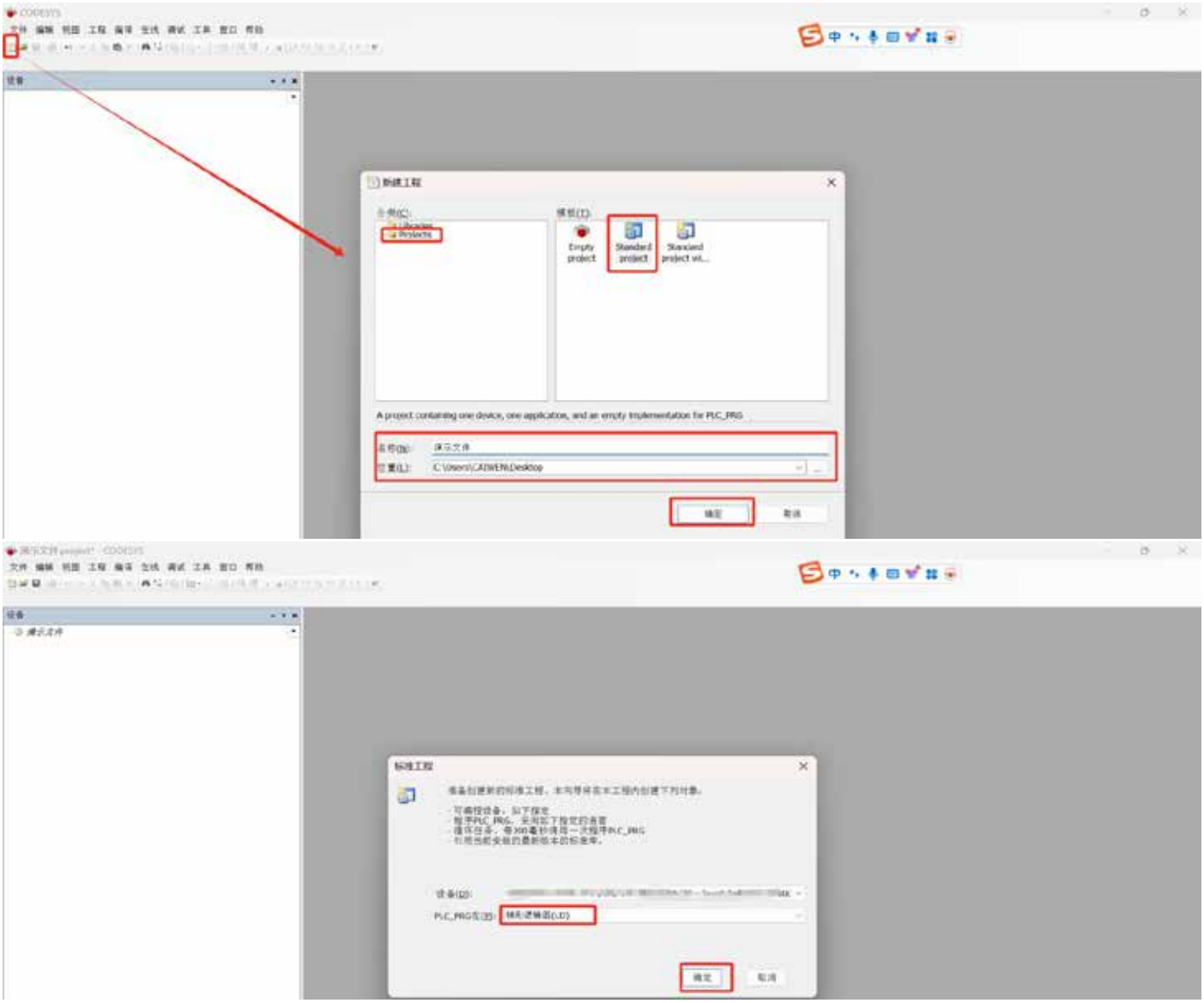
3.3.1 Foundation construction

1)First of all, users need to have CodeSYS kernel PLC and authorized "CodeSYS Vx.x" corresponding version of programming software.

2) Our Ethercat products and PLC products are powered on correctly, and the RJ45 network port of the Ethercat of PLC is inserted into the RJ45 network port of the Ethercat-IN of the product (it is recommended to use at least more than 5 types of network cables)。

3.3.2 Software Configuration

1) New construction

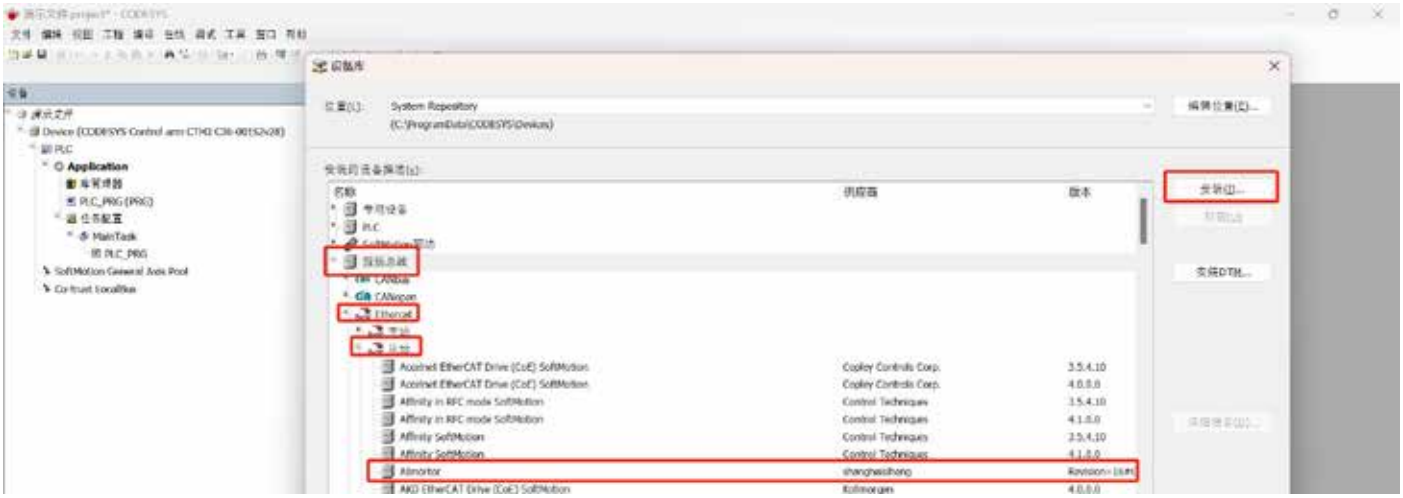


2) Add (install slave device Description XML file)

A, right-click "tools" in the navigation bar, then display the drop-down list, select "equipment library" and click;

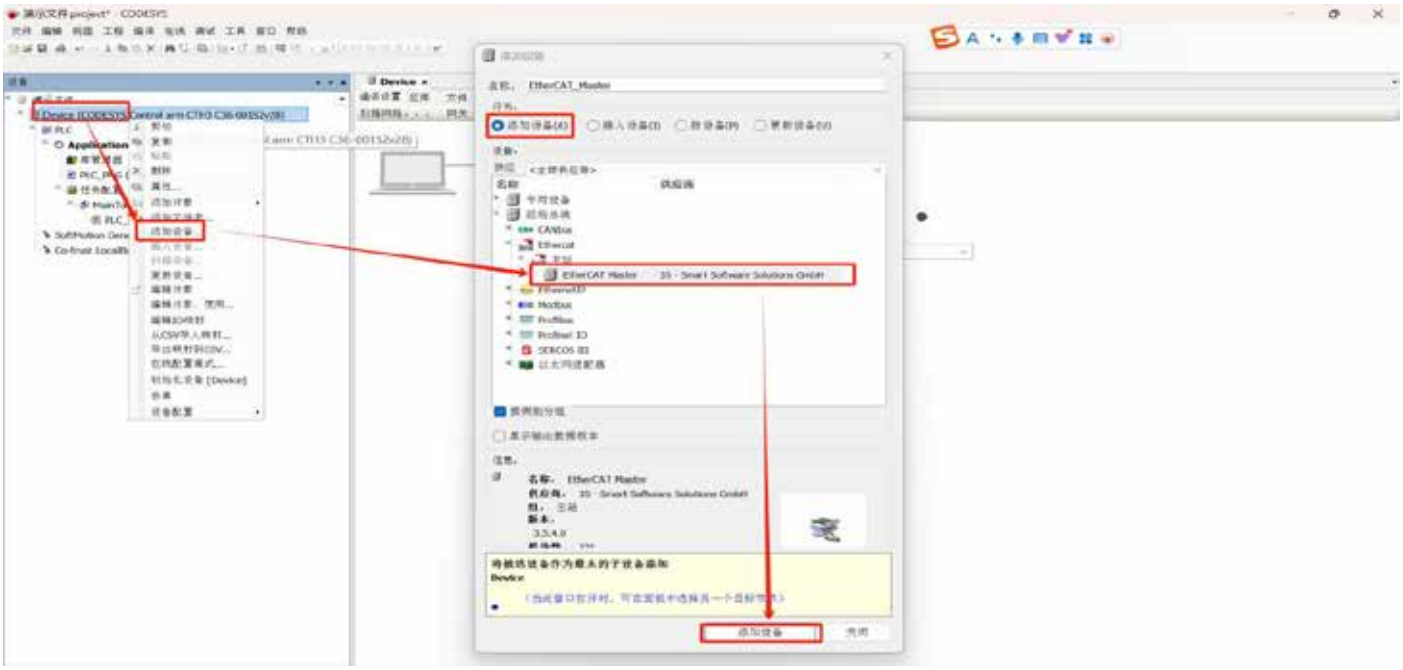
B, click install, path to find the XML file provided by our company to save to the location

After successful installation, the name of our equipment will be displayed in the equipment list in the field bus Ethercat slave station. If not displayed, the installation failed. You can try to reinstall or consult the PLC equipment manufacturer.



3) Add master station equipment

Right-click the PLC device on the left side of the window to display the drop-down list. Select "Add Device" to pop up the "Add Device Window" and select the "EtherCAT Master" Master device. Finally, click on "Add Device".



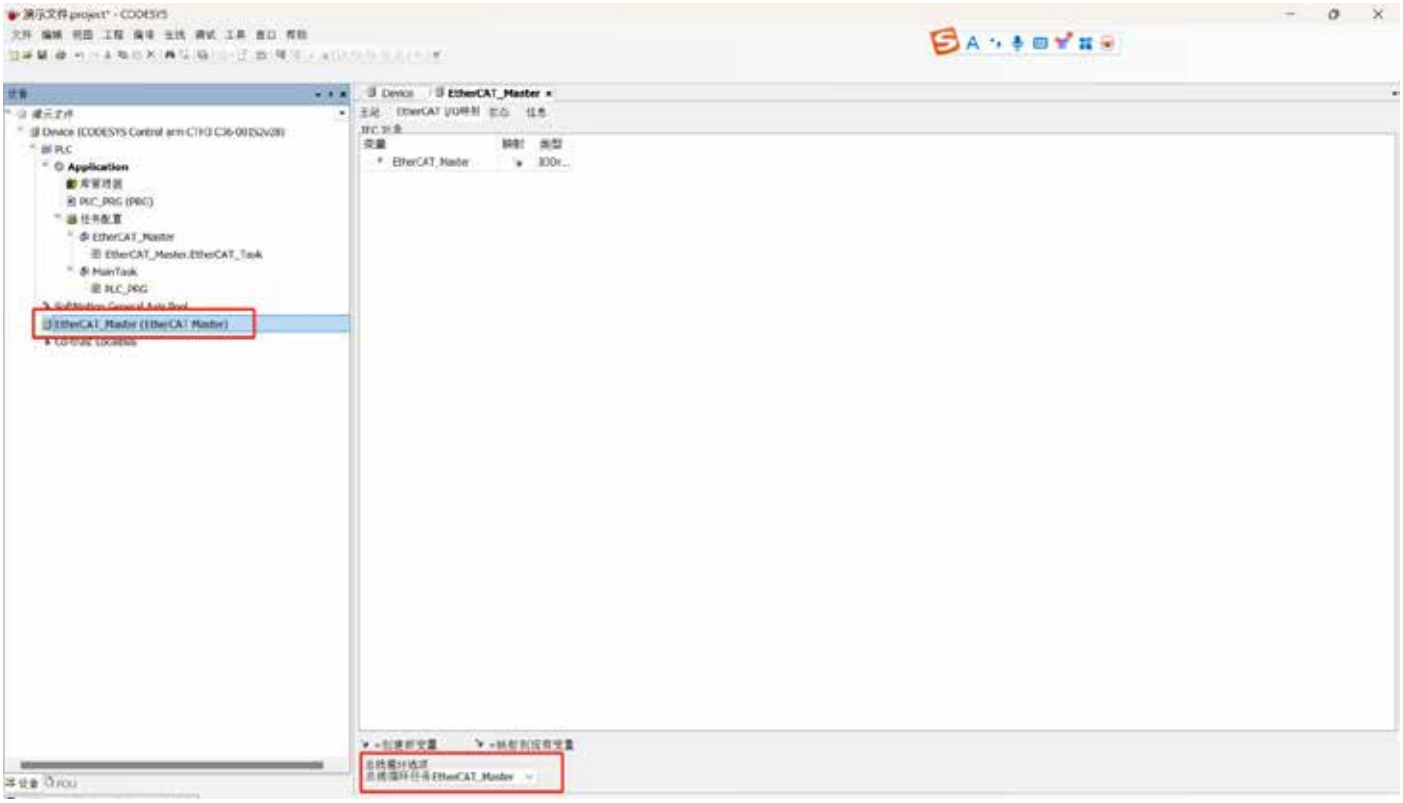
4) Configure the master device

After adding the "EtherCAT Master" Master device in the previous step, the Ethercat Master device will be displayed in the tree directory on the left. Double-click to enter the configuration window.

It is important to note that:

1. Cycle time setting. In this case, the default 4000us is used. The user configures the appropriate time according to the number of slave stations and requirements, and the minimum time is not less than 1000us.

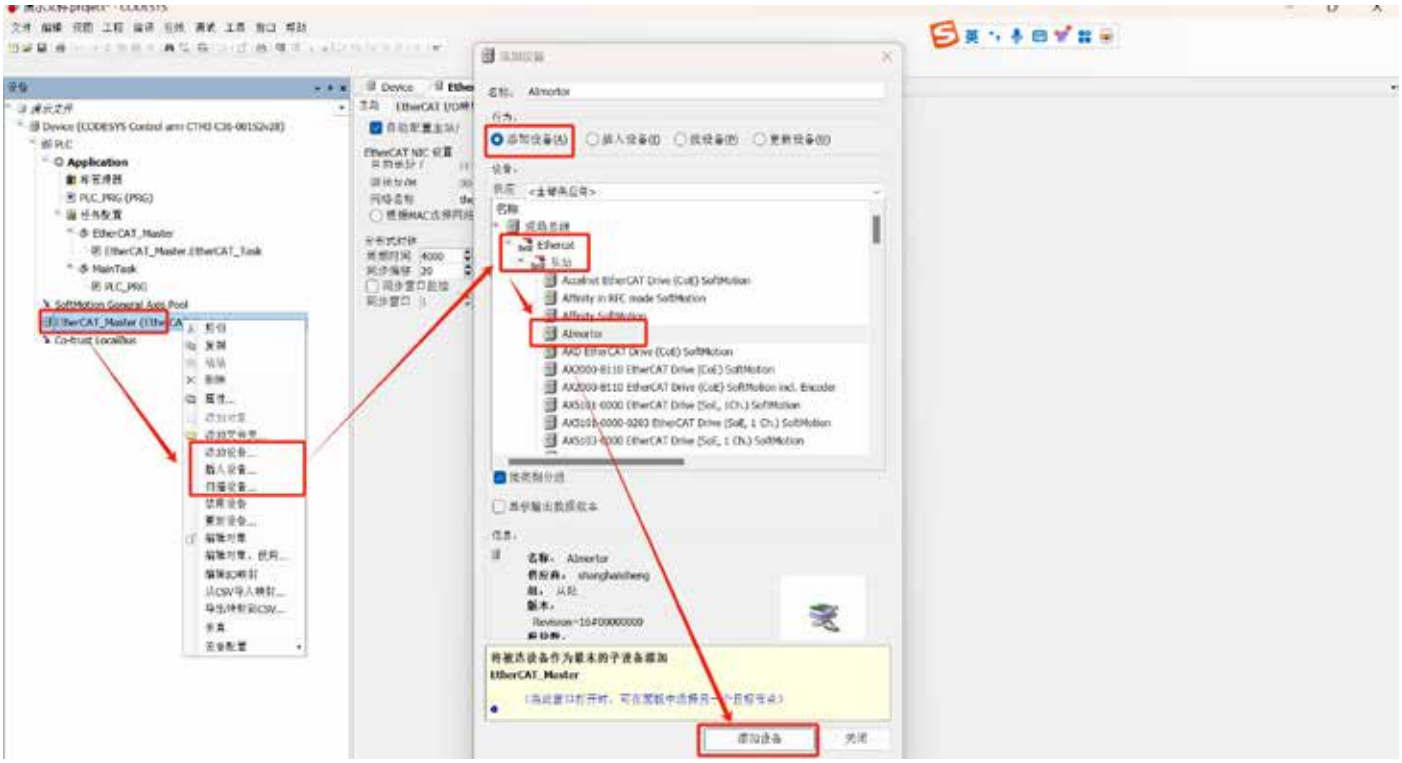
2.Ethercat I/O mapping needs to be configured under the task of Ethercat_Master master;



5) Add slave device

Right-click on the Ethercat Master device, display the drop-down list, select Add device, will pop up the device window, select our company device, and finally click "add device". Scanning equipment can also be added online

.



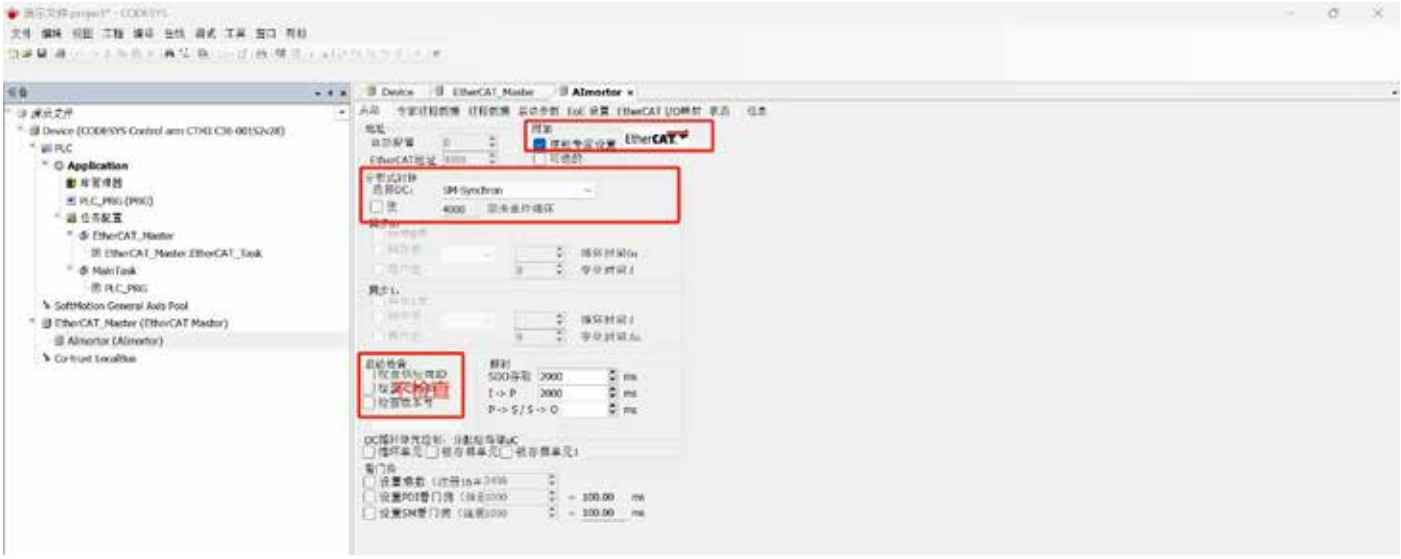
6) Configure the slave device

After the slave device is added in the previous step, the slave device of our company will be displayed under the Ethercat Master device in the tree directory on the left. Double-click to enter the configuration window.

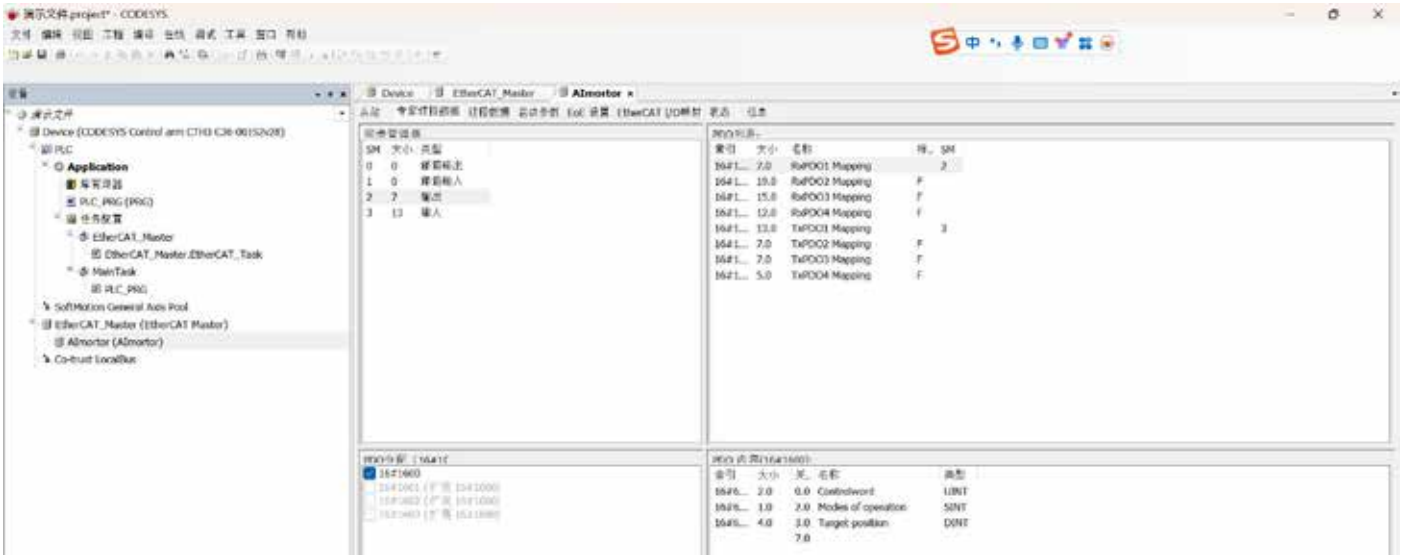
A. Configure the slave device synchronization method. In this case, the default is used for all devices, and SM event synchronization is used.

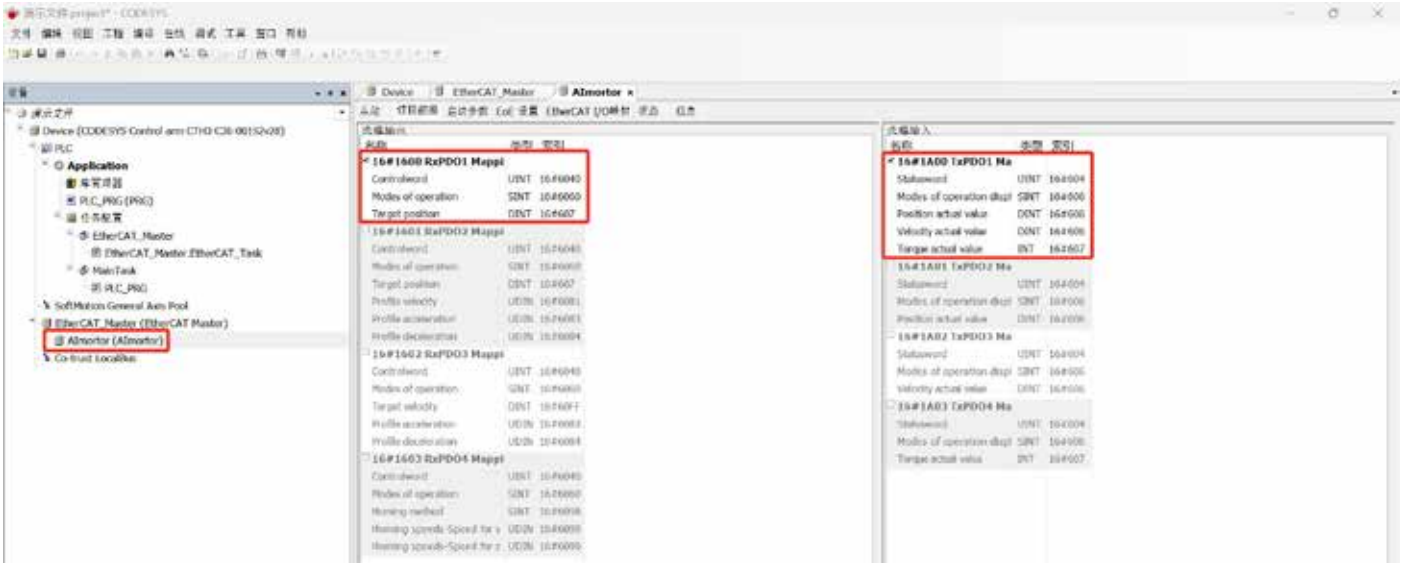
Startup parameters, PDO channel and PDO mapping used by process data and I/O variable mapping, etc

In this case, the default is used. SM event synchronization and the default PDO are used. For the specific configuration method, please refer the relevant information of the PLC manufacturer.

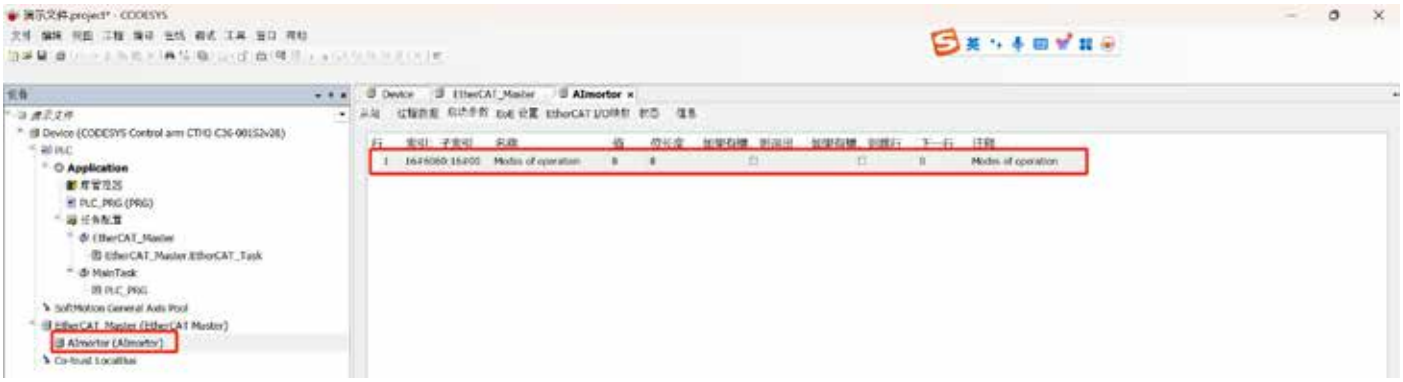


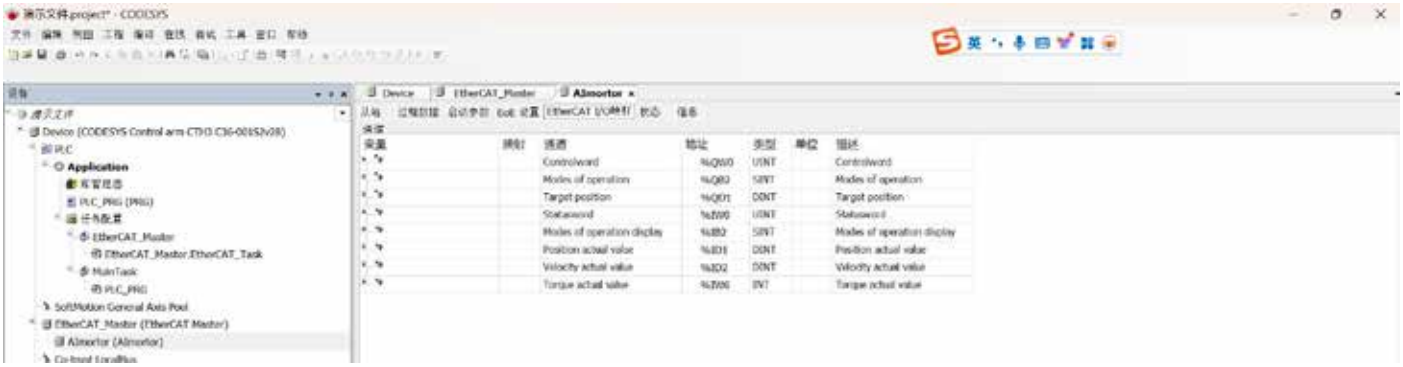
B. Configure expert process data and process data (PDO configuration). In this case, the default configuration is used, and the user can configure it





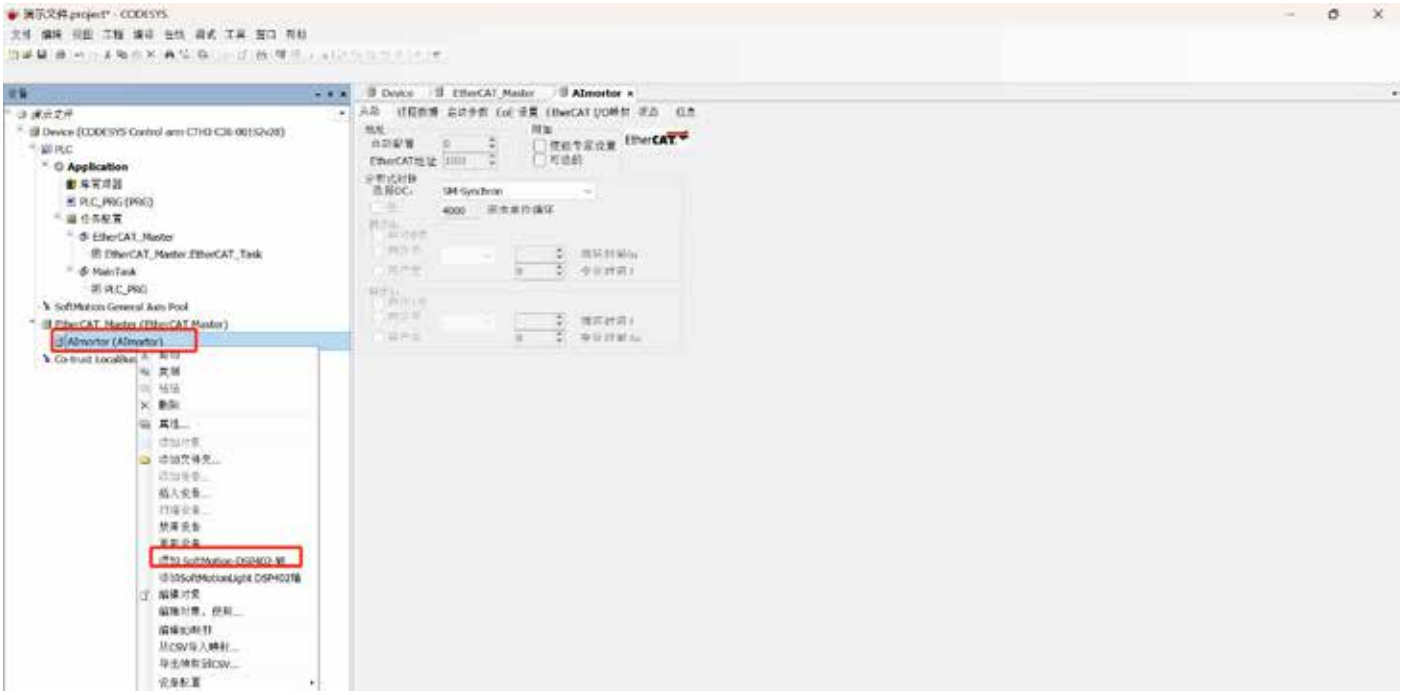
C. Configure startup parameters (default)





7) Add DSP402 axis

In the tree directory on the left, our company right-click the pop-up drop-down box from the device and select "Add SoftMotion-DSP402-Axis"



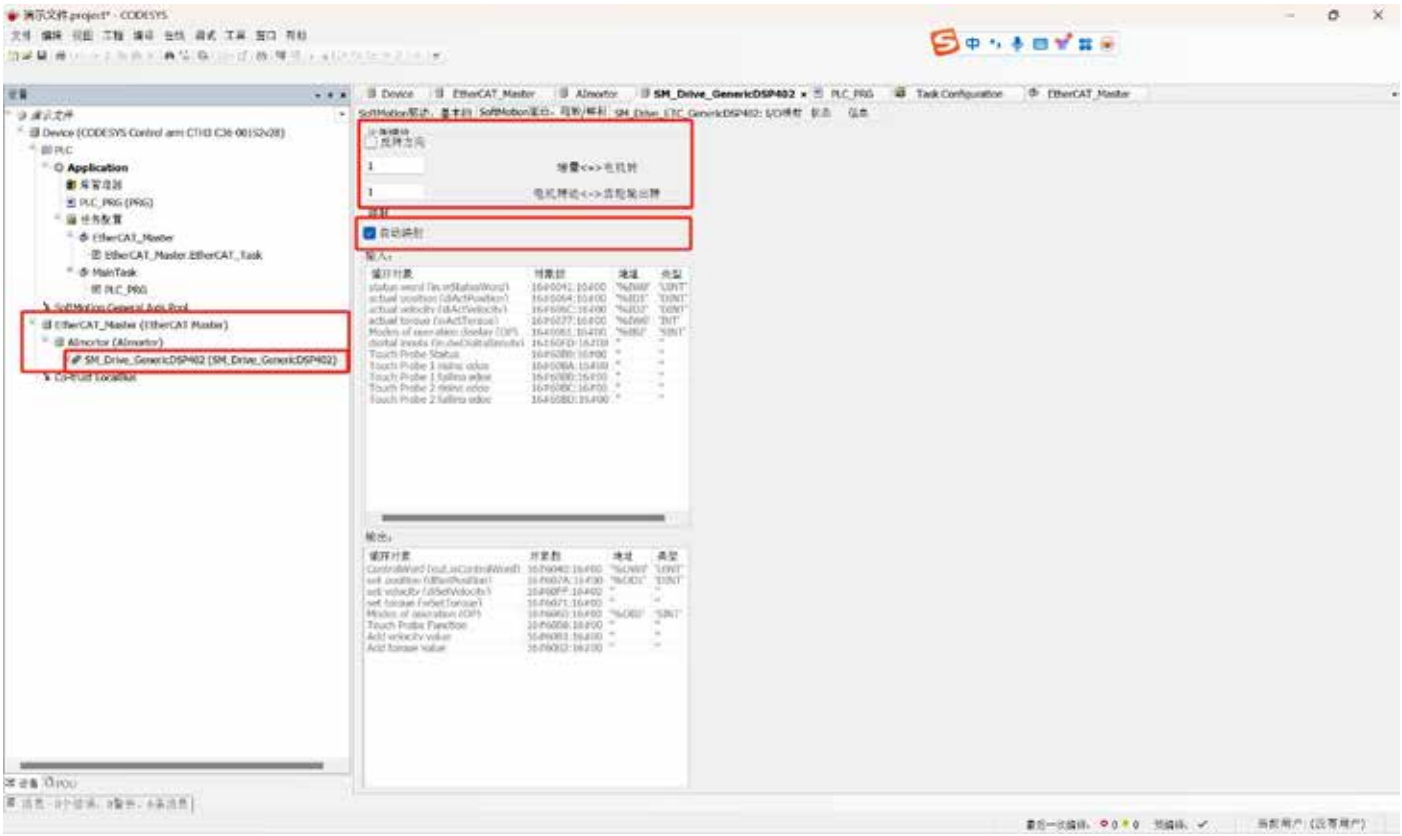
8) configure DSP402 axis

After adding the 402 axis in the previous step, the "SM_Drive_GenericDSP402" axis will appear under the slave device. Double-click to pop up the configuration window.

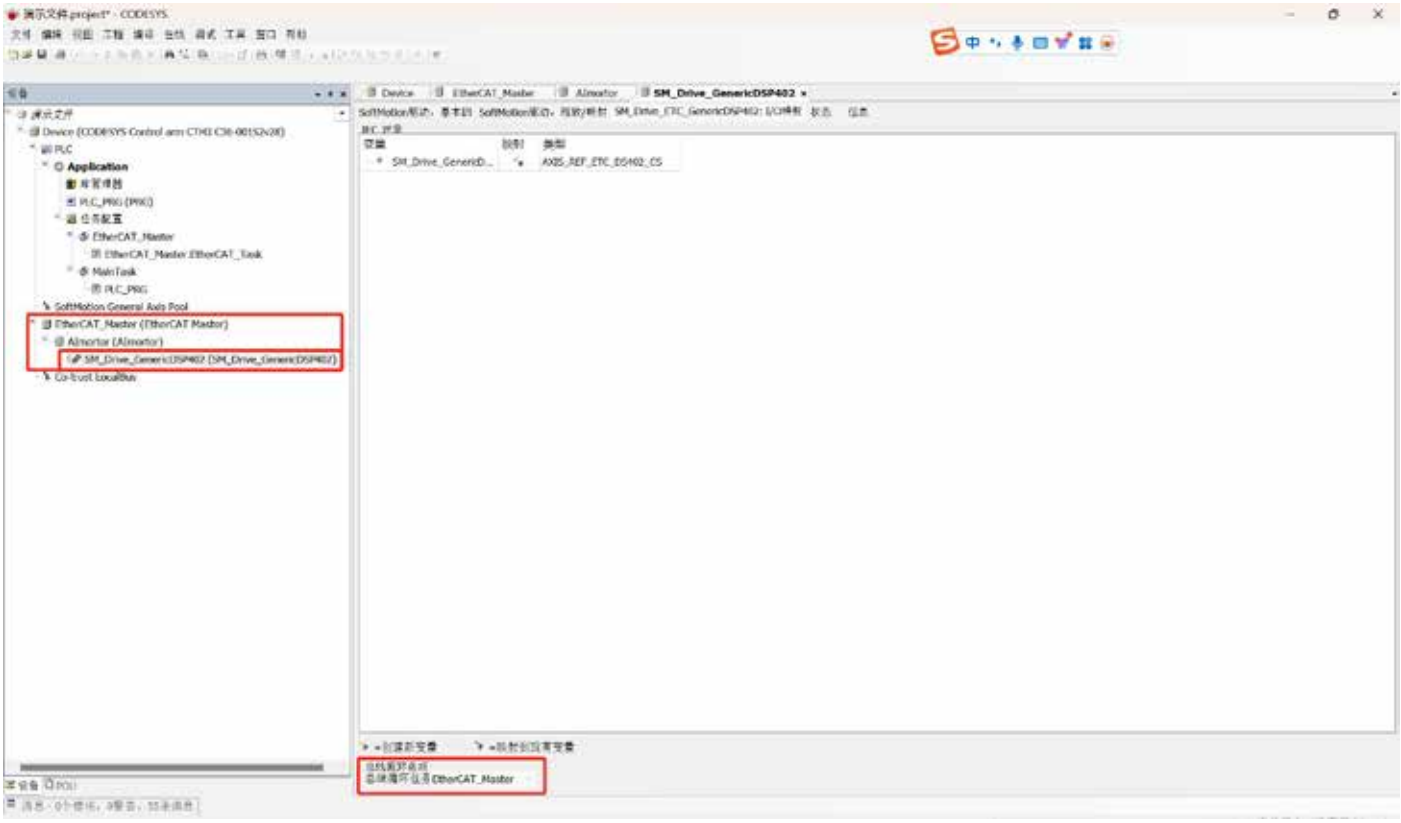
A, configured as A defined axis (that is, the physical axis of linear motion)

B, scaling: our products factory default 1000 pulse one turn, here set the motion command 1000 unit corresponding to 1000 pulse (1 turn) without deceleration mechanism.

Check automatic mapping: the mapping address is automatically assigned by the PLC.



C. DSP402:I/O mapping needs to be configured under the Ethercat_Master master master station task

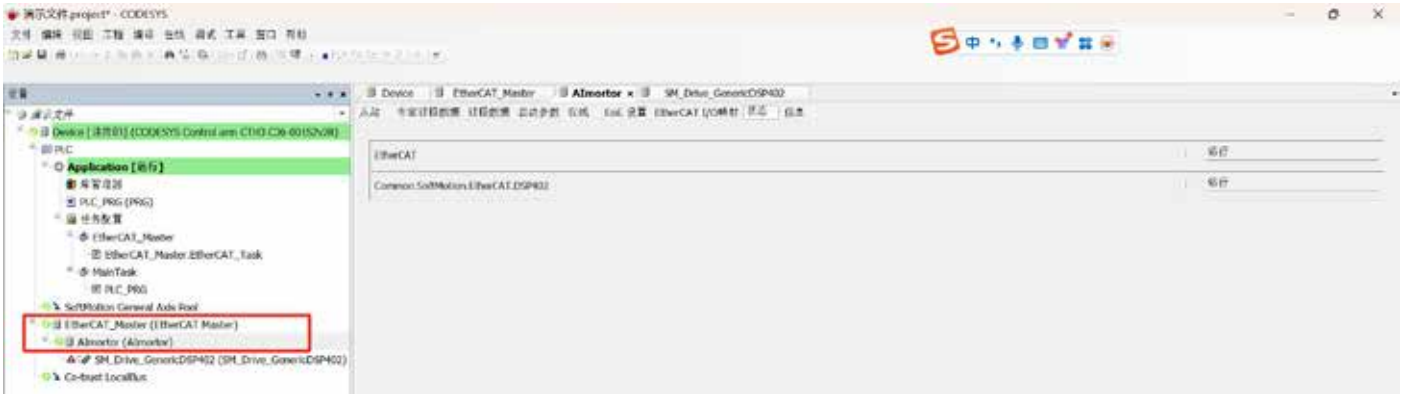


3.3.3 Activation Configuration

Compile first, then configure the PLC to communicate with the computer, download and configure the PLC in online mode, enter the operation monitoring, and observe the bus operation.

1: If the Ethercat_Master master master station is not running, try to perform "hot and cold reset" operation, or contact PLC manufacturer technical support;

2: If Almortor is not running, check whether the network cable is connected correctly and whether the slave device is in a faulty state (you can restart the slave device to reset).



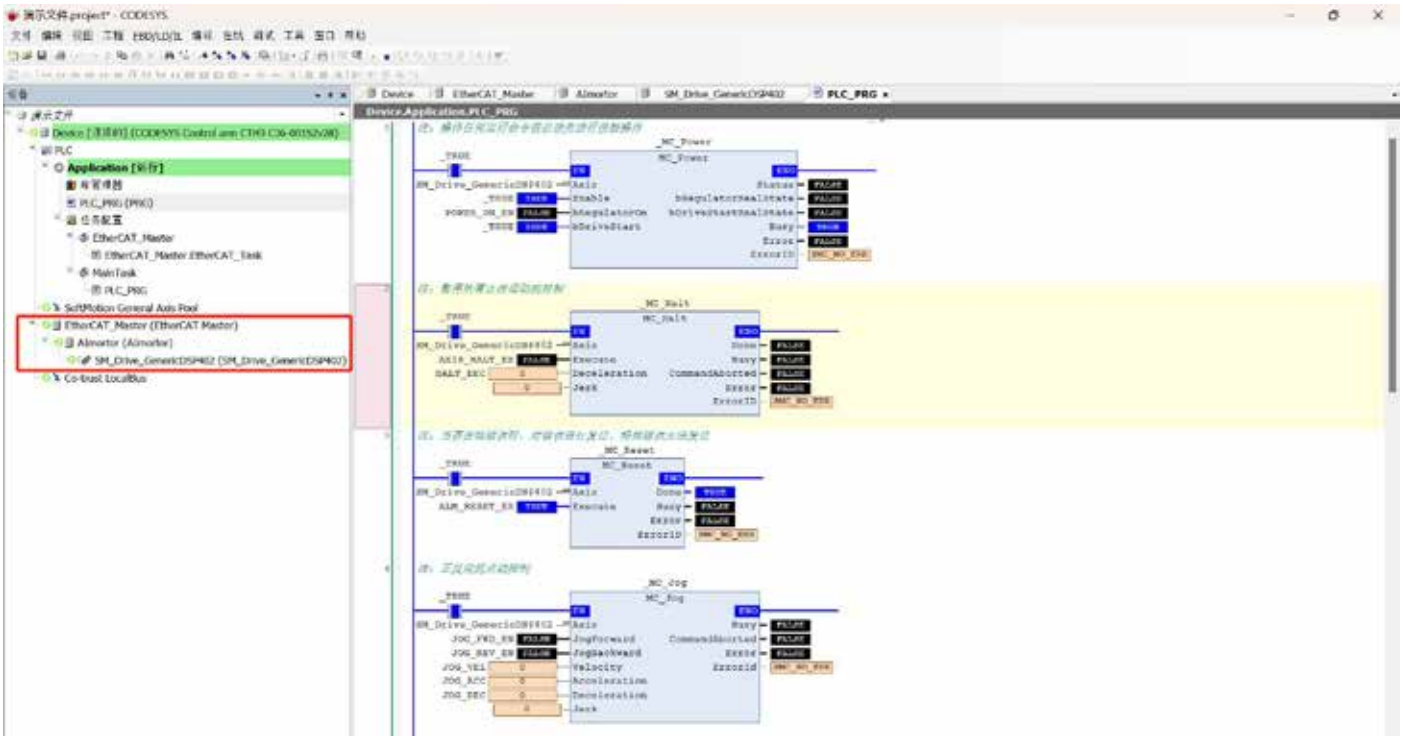
3.3.4 Programming Control

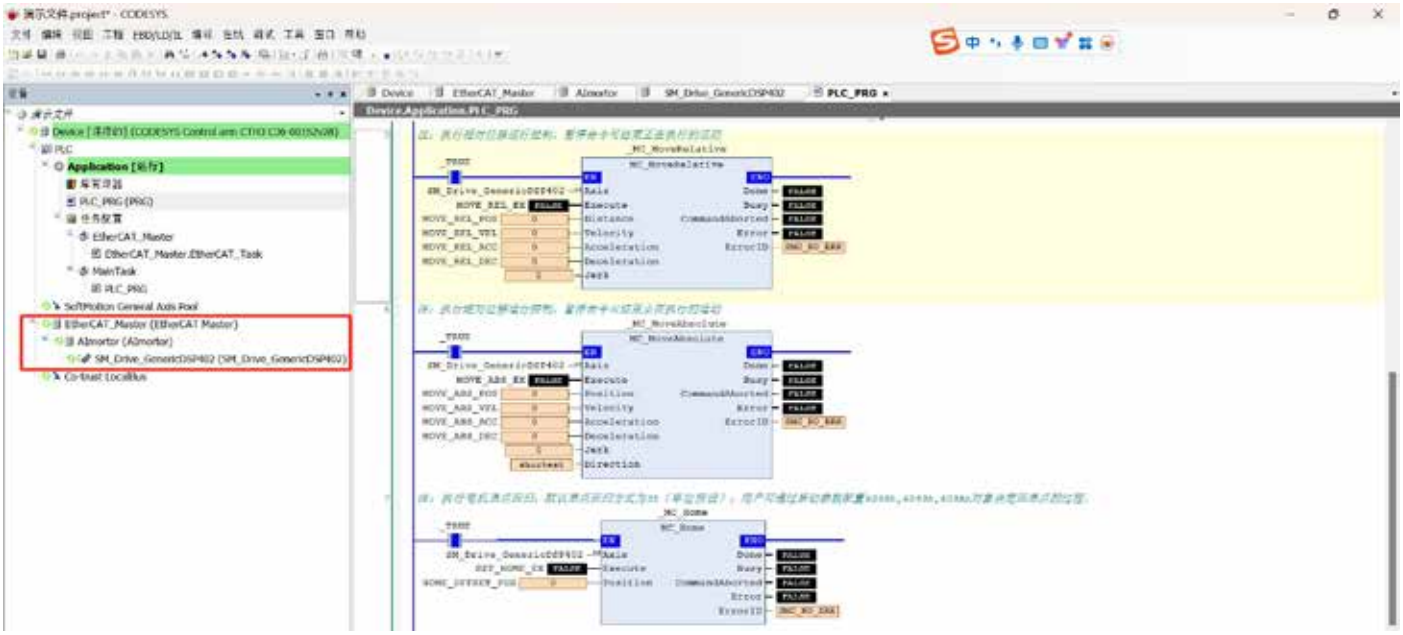
Program reference, in order to commonly used several motion command programming demonstration, the actual use according to the process requirements of the application.

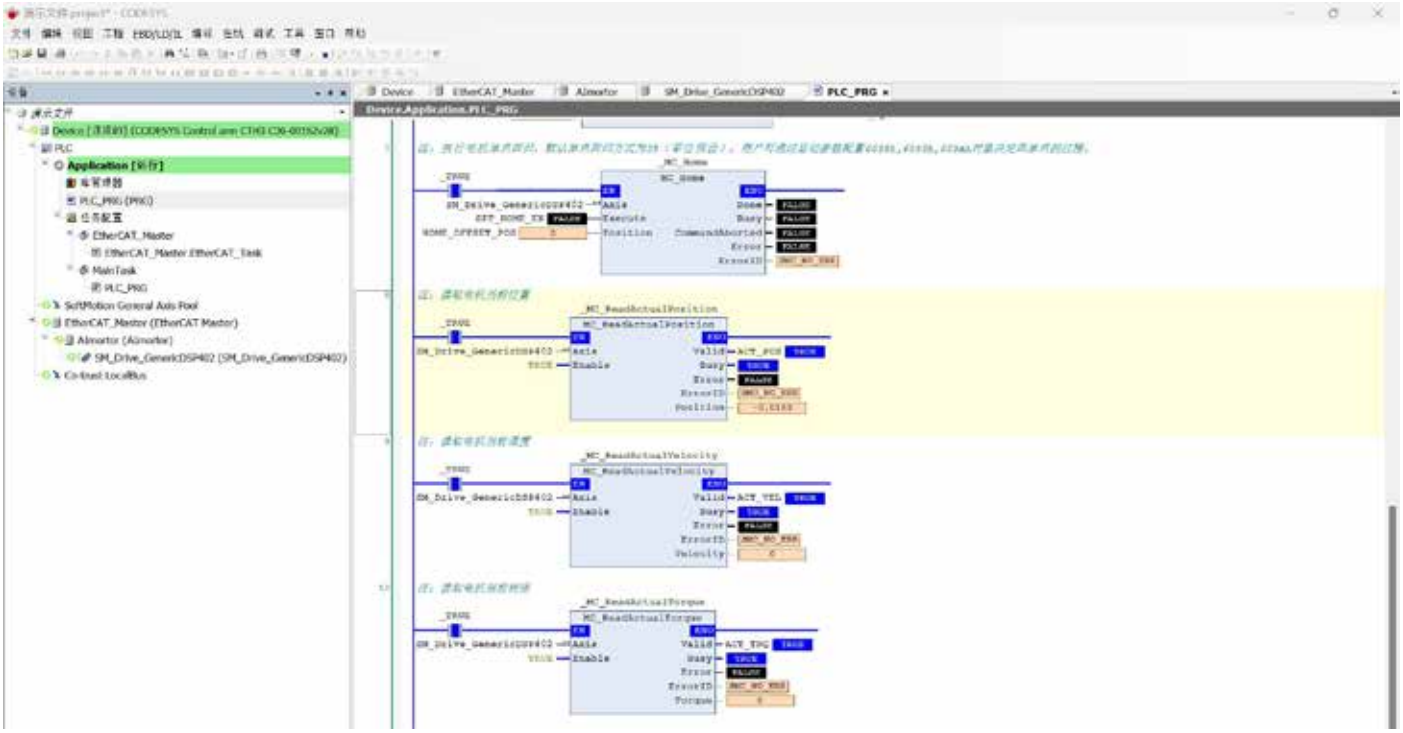
After writing the program offline, compile through, download. Observe the running status of the PLC at the bottom of the window. If the PLC reports an error and runs overtime, check the PLC task configuration or contact the technical support of the PLC manufacturer.

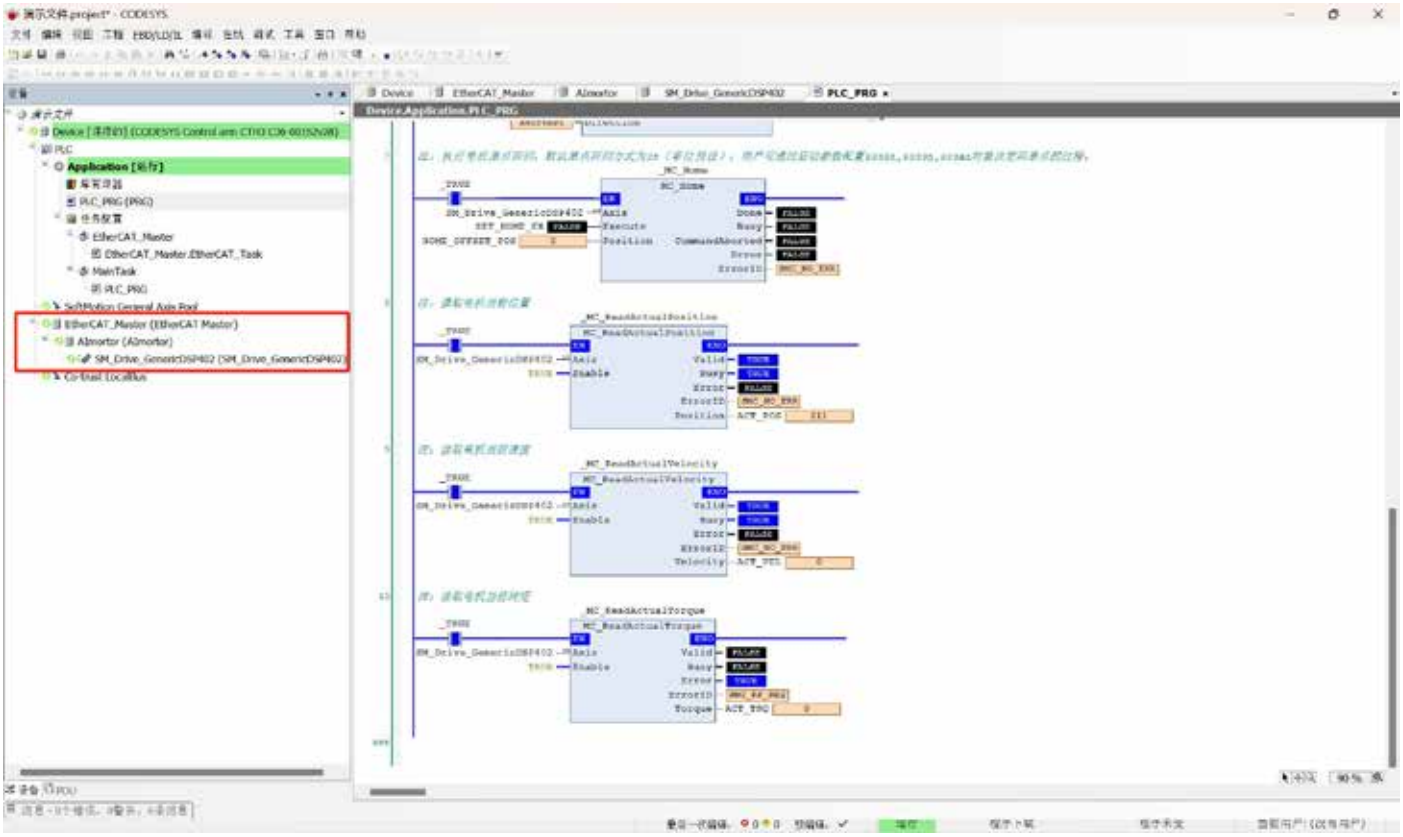
If the left "SM_Drive_GenericDSP402" axis is not running, it means that there is an error in the Axis command, and MC_Reset can be

performed to reset the axis fault;

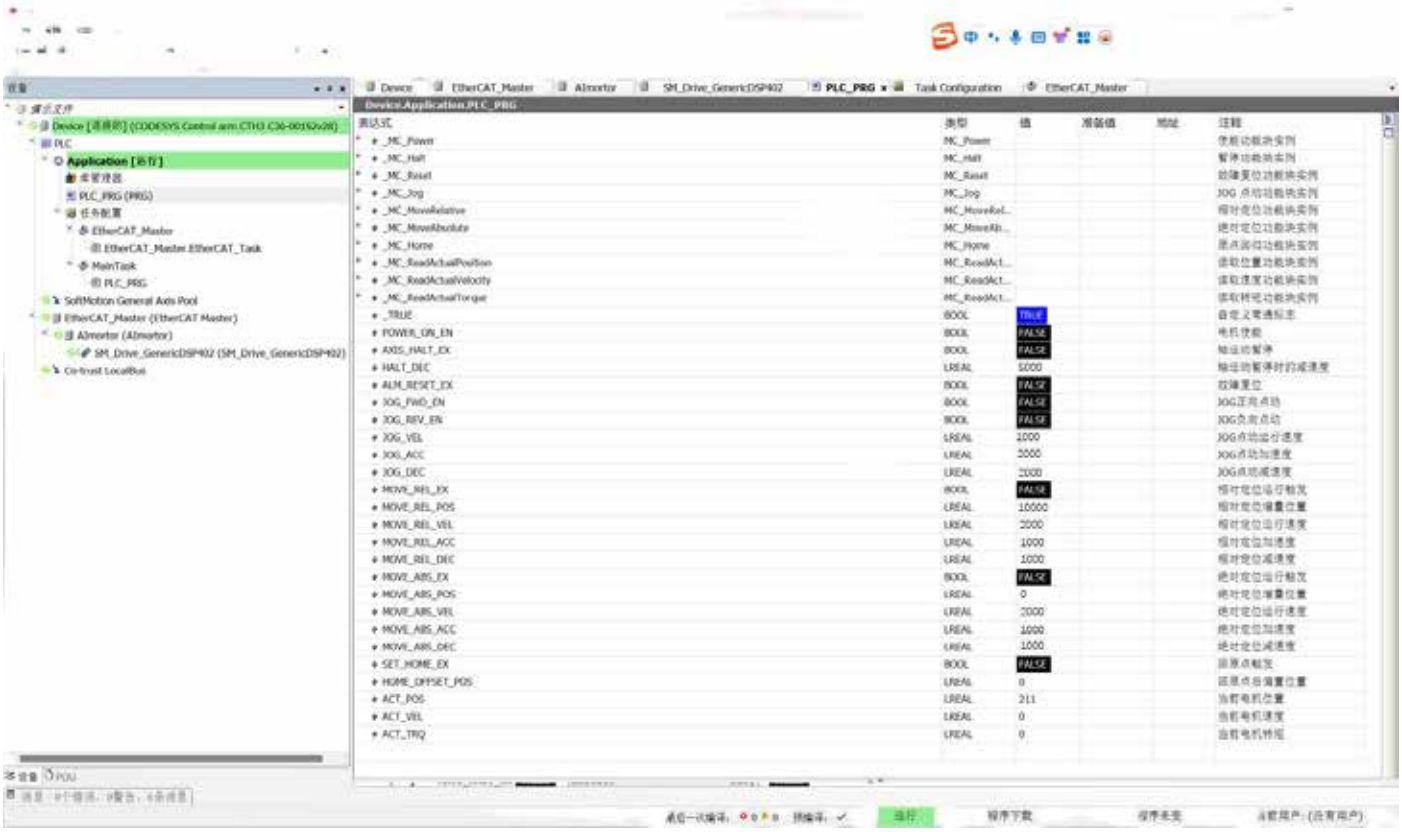








Demonstrate the use of the variable table:



3.4 Adapted to Omron NX/NJ series controller operation case

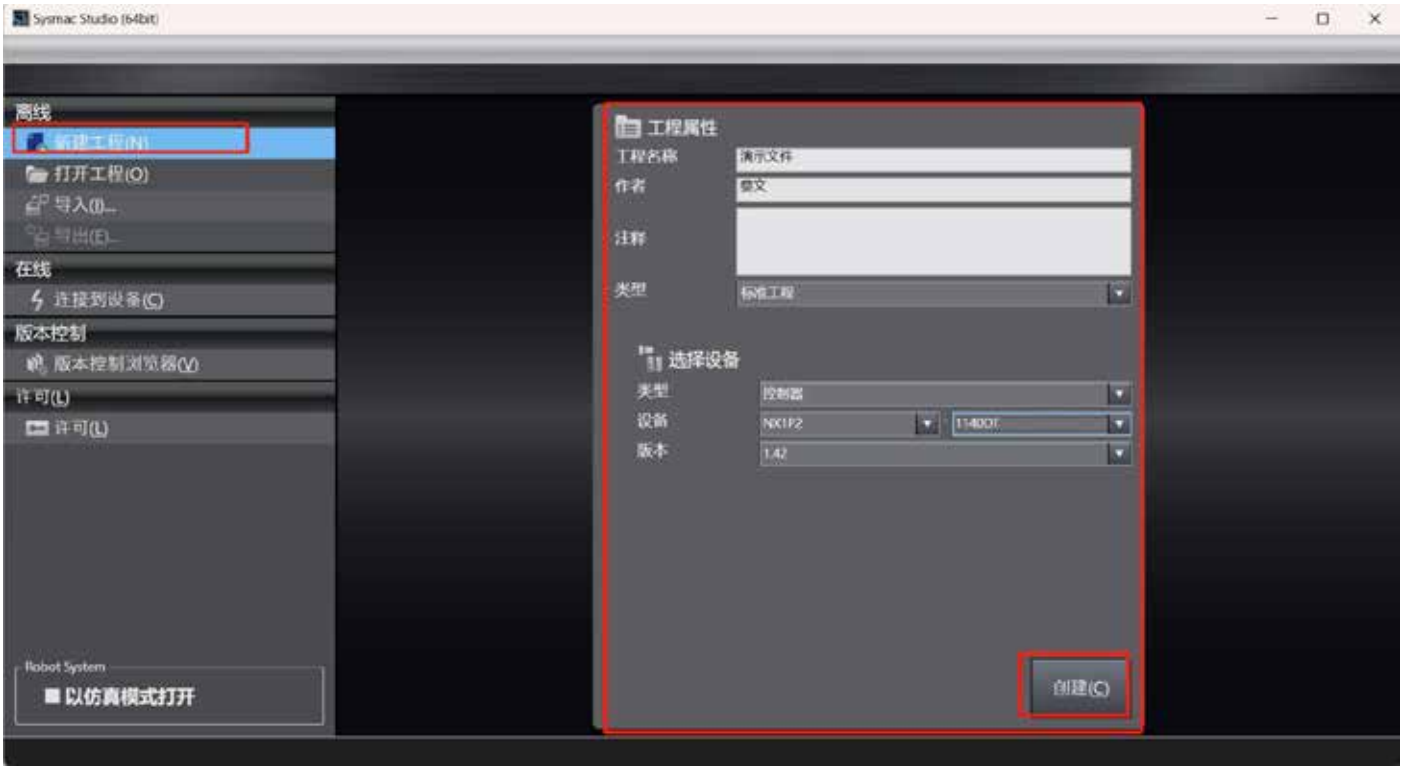
The following uses Omron NX/NJ controller as the Ethercat master station to demonstrate the simple configuration and use process of our AIMortor servo driver.

3.4.1 Foundation construction

1) First of all, users need to have Omron NX/NJ series PLC hardware and install authorized "Sysmac Studio" programming software.
2) our Ethercat products and PLC products are powered on correctly, and the RJ45 network port of Ethercat of PLC is inserted into the network cable and connected to the RJ45 network port of the Ethercat-IN of the products (it is recommended to use at least more than 5 types of network cables).

3.4.2 Software Configuration

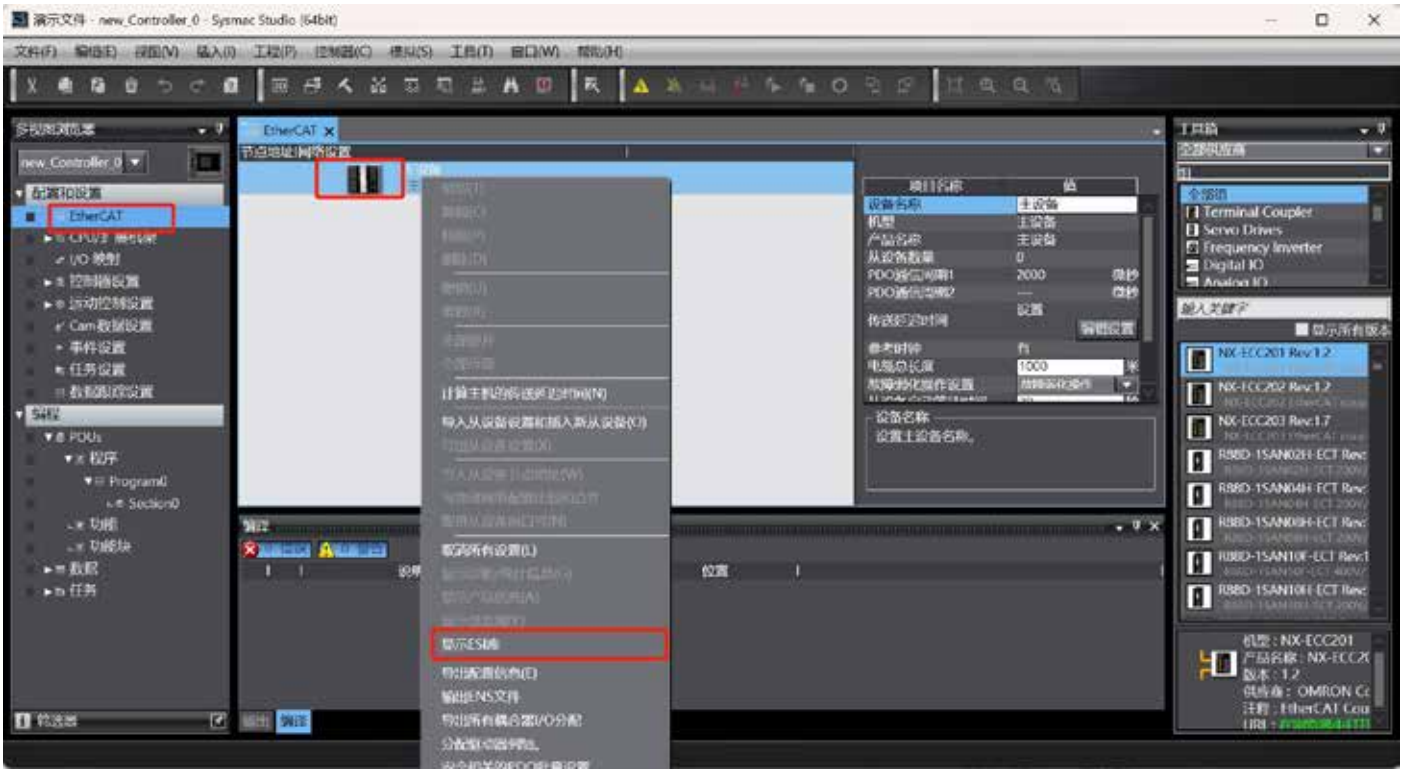
1) New construction



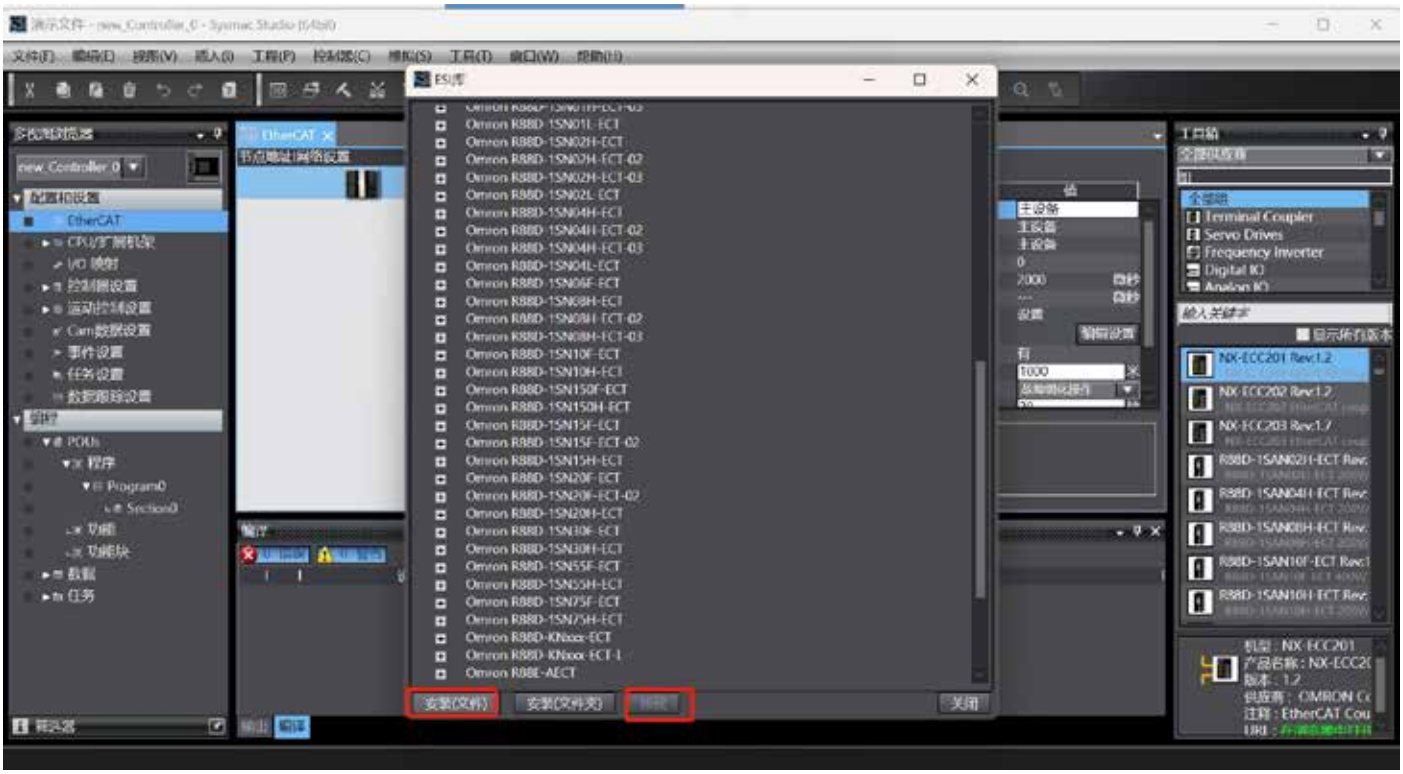
2) Import our product XML file Ethercat configuration file of our motor product (SH_AIMotor_ECAT.xml)

A, first enter the "EtherCAT" window, right-click on the "master device" to display the drop-down list, and select "display ESL Library" on the drop-down list ";

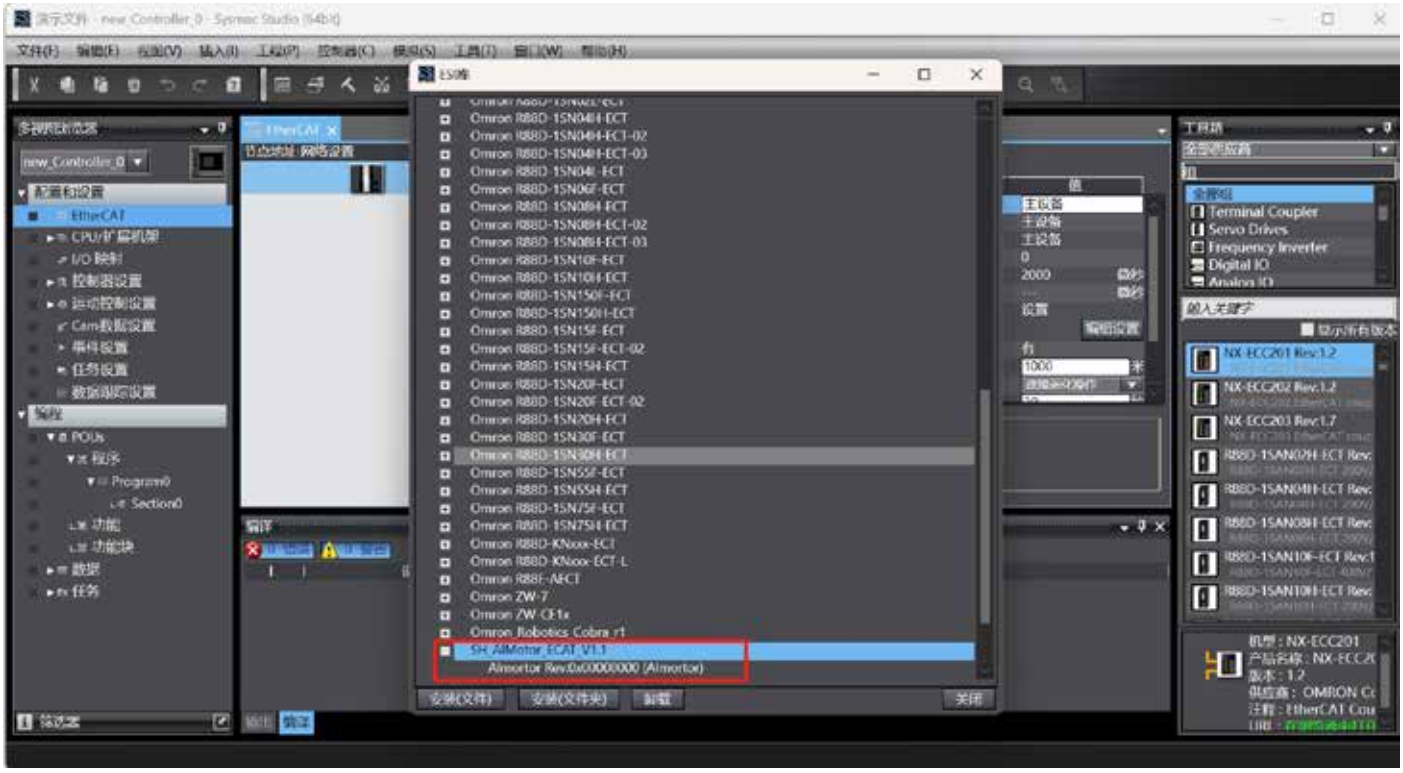
3)



B in the "ESL Library" window, click "install" to find the XML description file of our company's equipment and confirm the installation. If the XML file needs to be upgraded and replaced, uninstall it before installation., Generally, after installing the XML file, the software will remind you to shut down and restart, or update the page again.

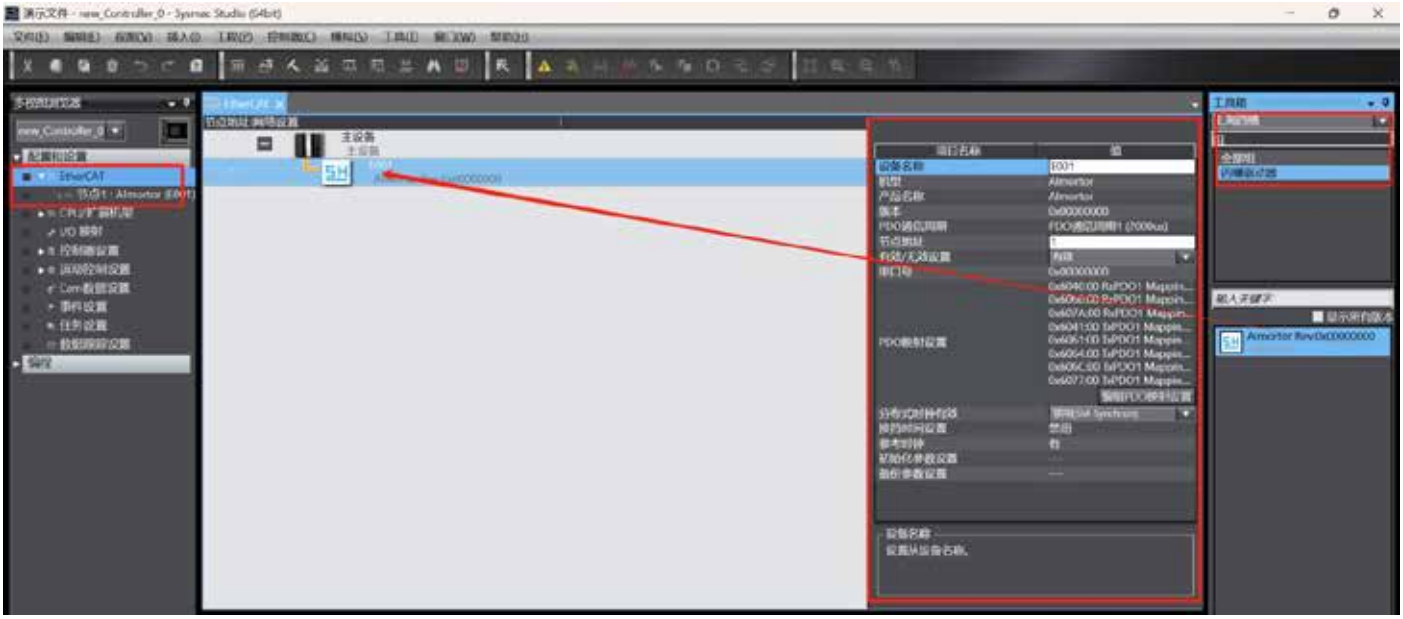


C, after the XML file is installed correctly, the "ESL Library" will display our equipment information;

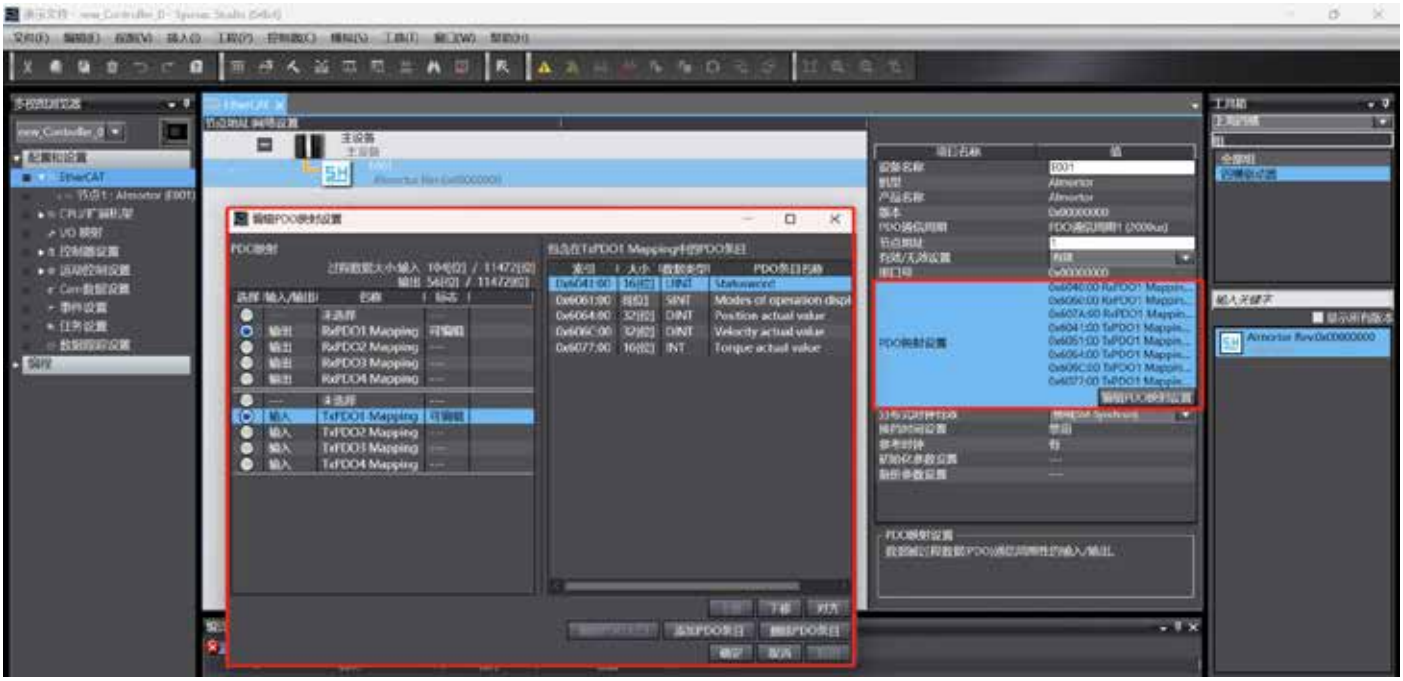


2) Set up Ethercat network equipment

A, in the right "tool box. Find our device and double-click the device icon to add the device to the Ethercat network. Click the device added in the Ethercat network, the device information will be displayed, such as device name, product name, node address, PDO mapping settings, etc.

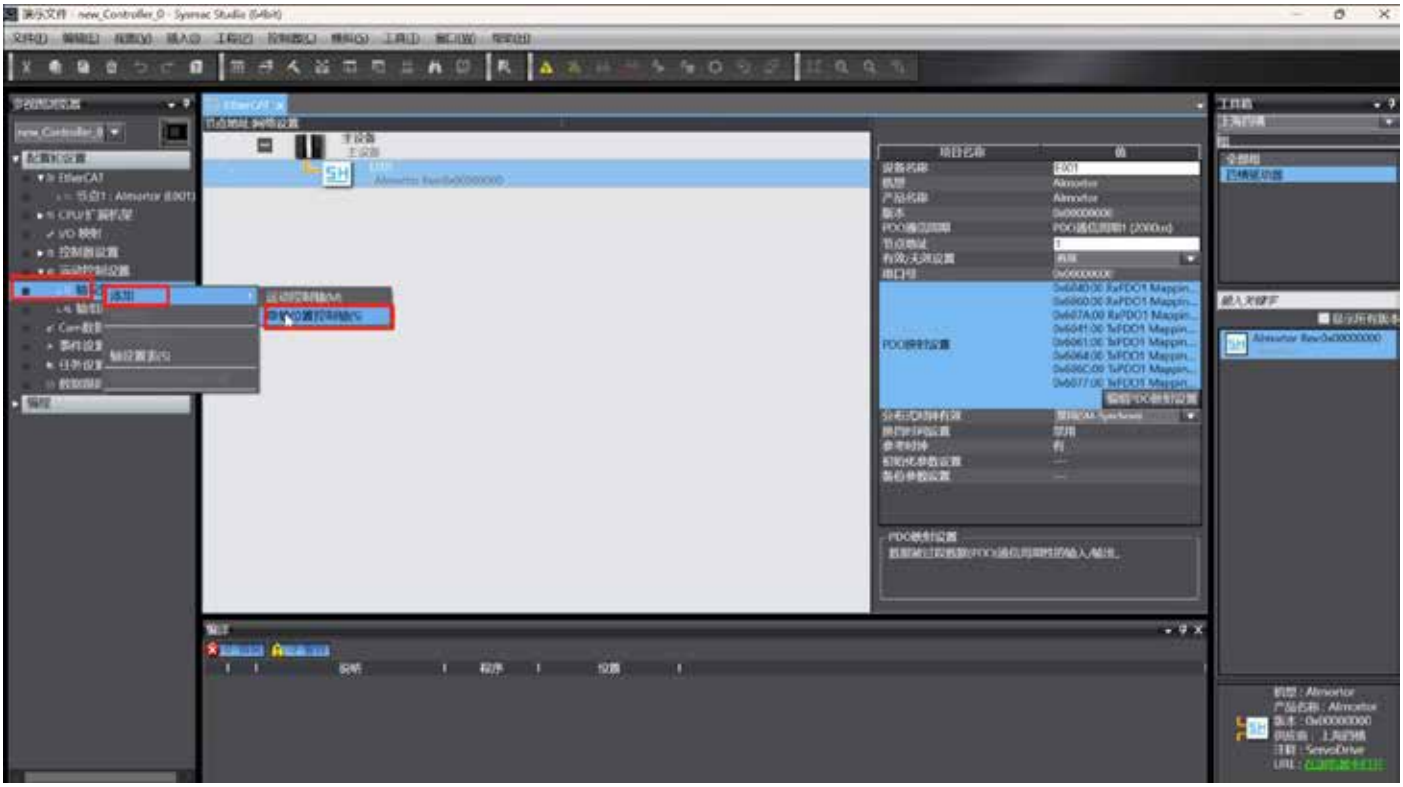


B, PDO configuration, configure the PDO Object t to be mapped according to user requirements, and use the default configuration in this case.

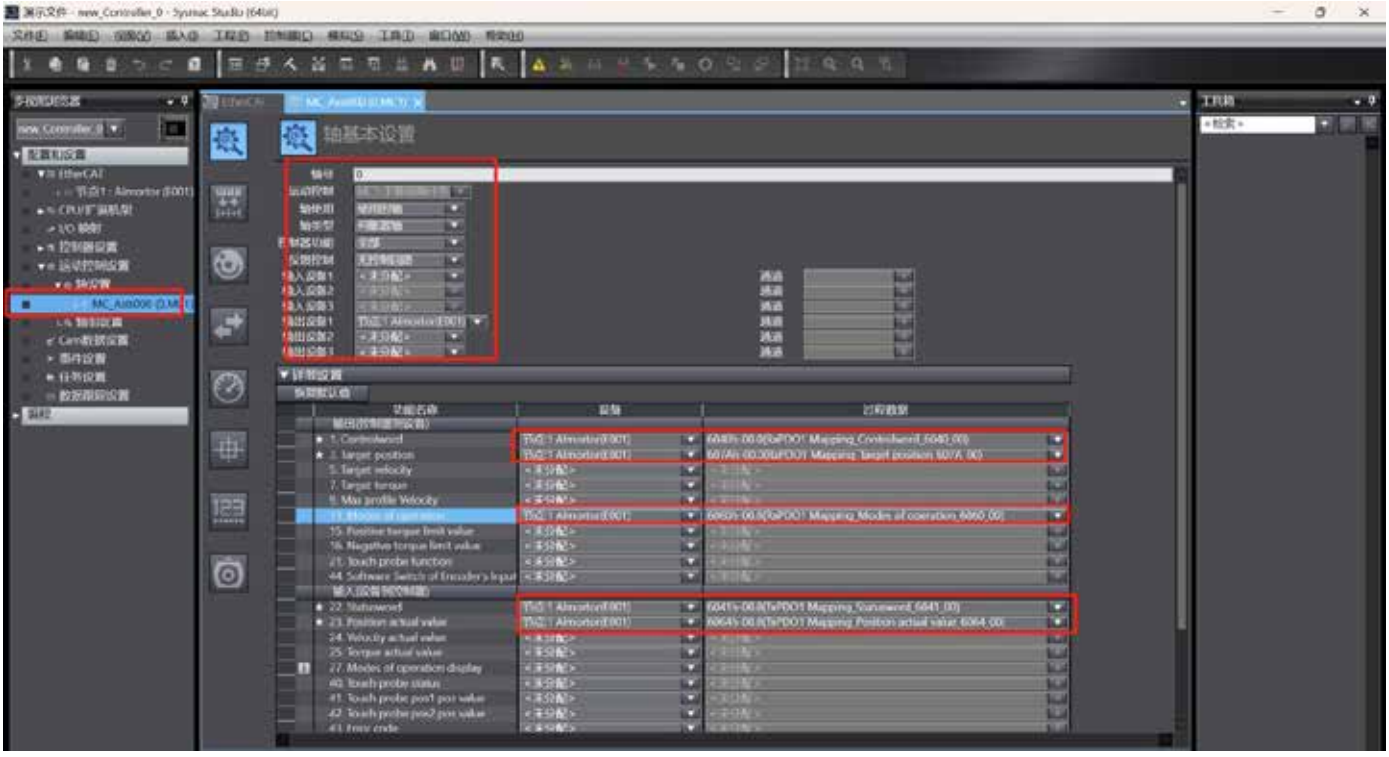


3) Configure 402 axis

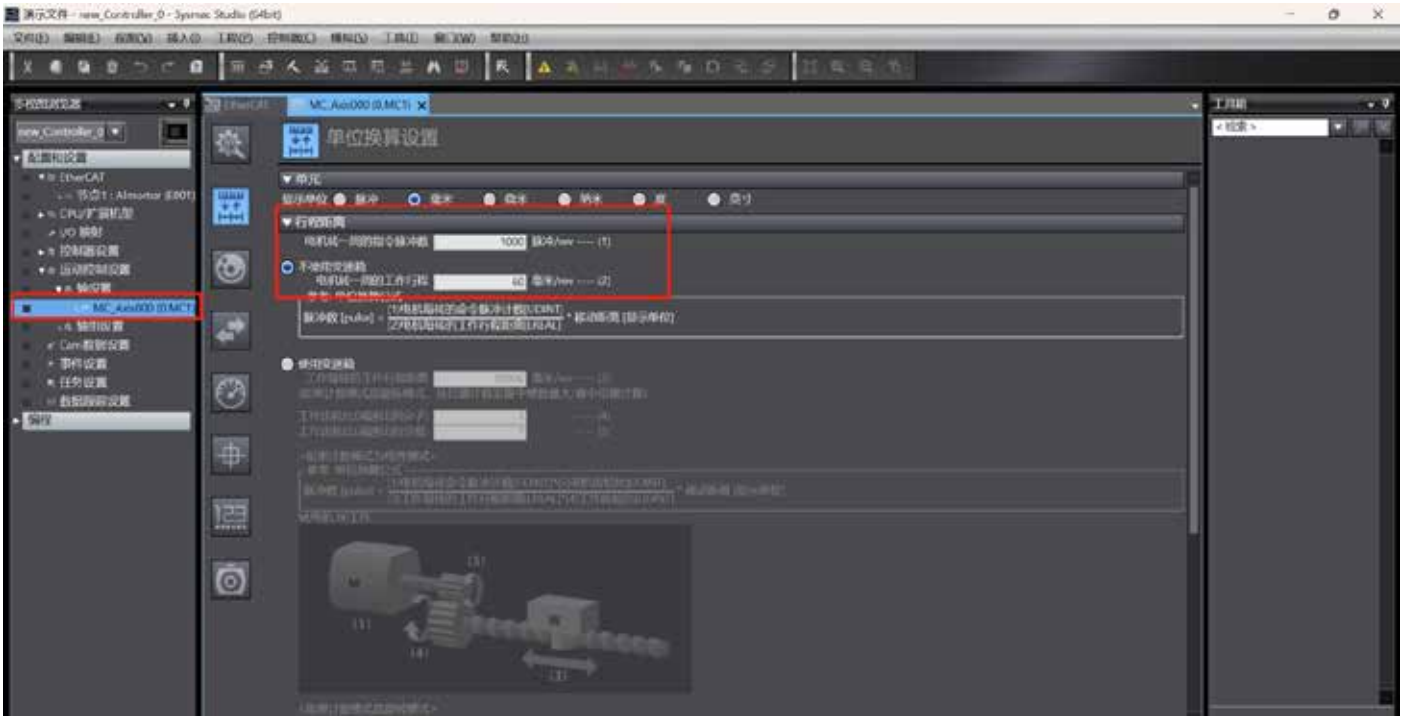
A, on the left, right-click at "axis settings", drop-down box "add", drop-down box click "single-axis position control axis", if you need to form an axis group, click "motion control axis", this case single-axis demonstration. A CIA402 axis "MC_Axis000" will be displayed after configuration ".



B, configure the basic settings of the axis, the third-party servo needs to manually configure the "output" at the bottom of the page, "input" mapping Association. At least five Object ts "6040H, 607A,6060H, 6041H, 6064H" need to be associated, otherwise the motion instruction cannot be used.

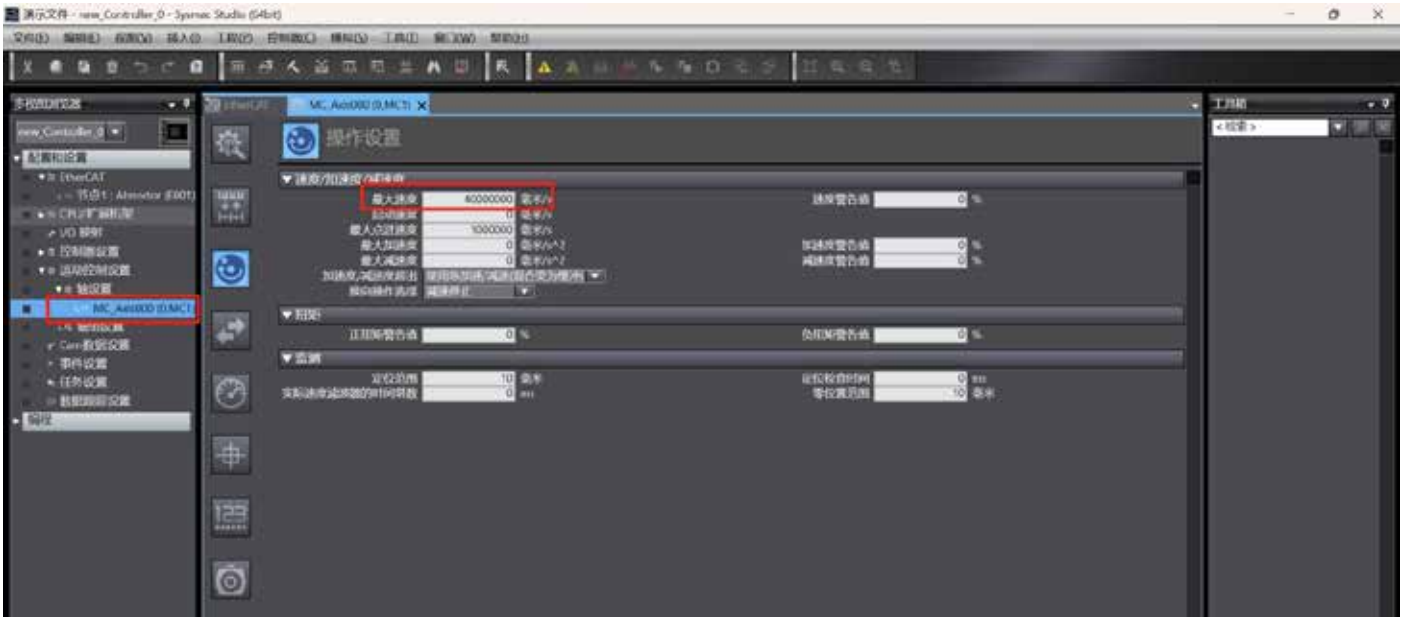


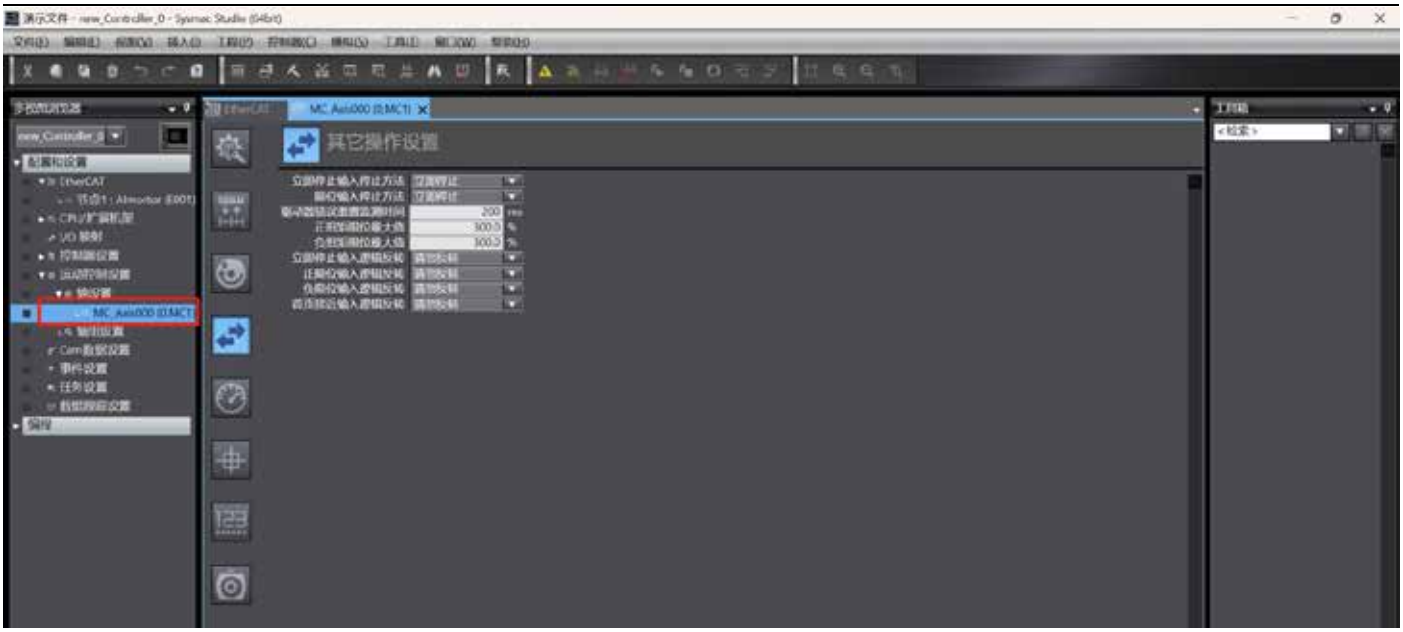
C, configuration unit conversion, our products factory default 1000 pulse one turn, here set the motion command 60 units corresponding to 1000 pulse (1 turn) without deceleration mechanism.

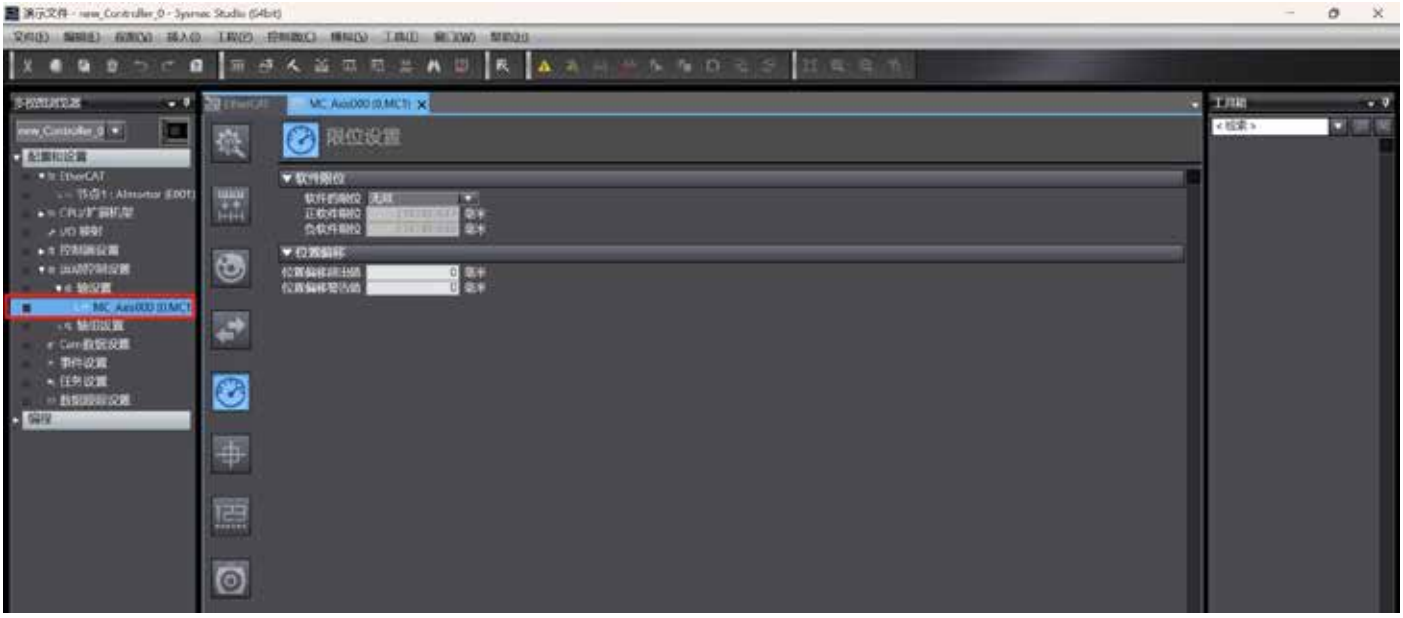


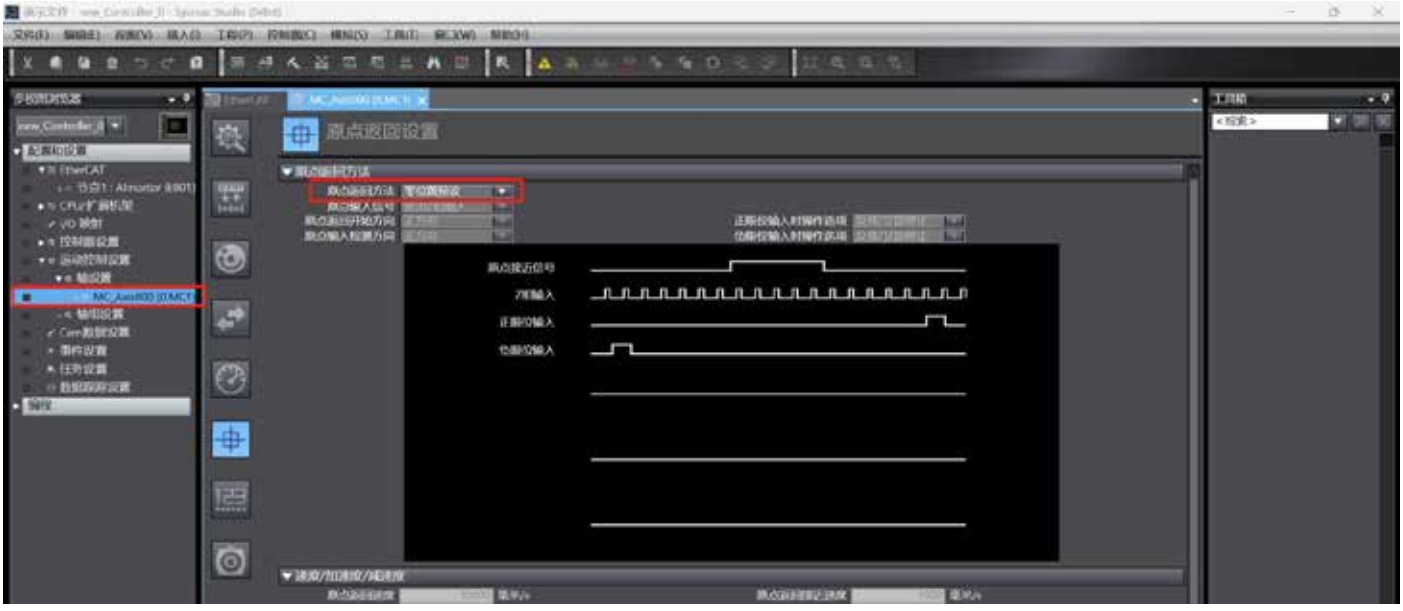
D, other configurations basically use the default, specific reference to the following figure, not listed one by one, specific reference to Omron official documents.

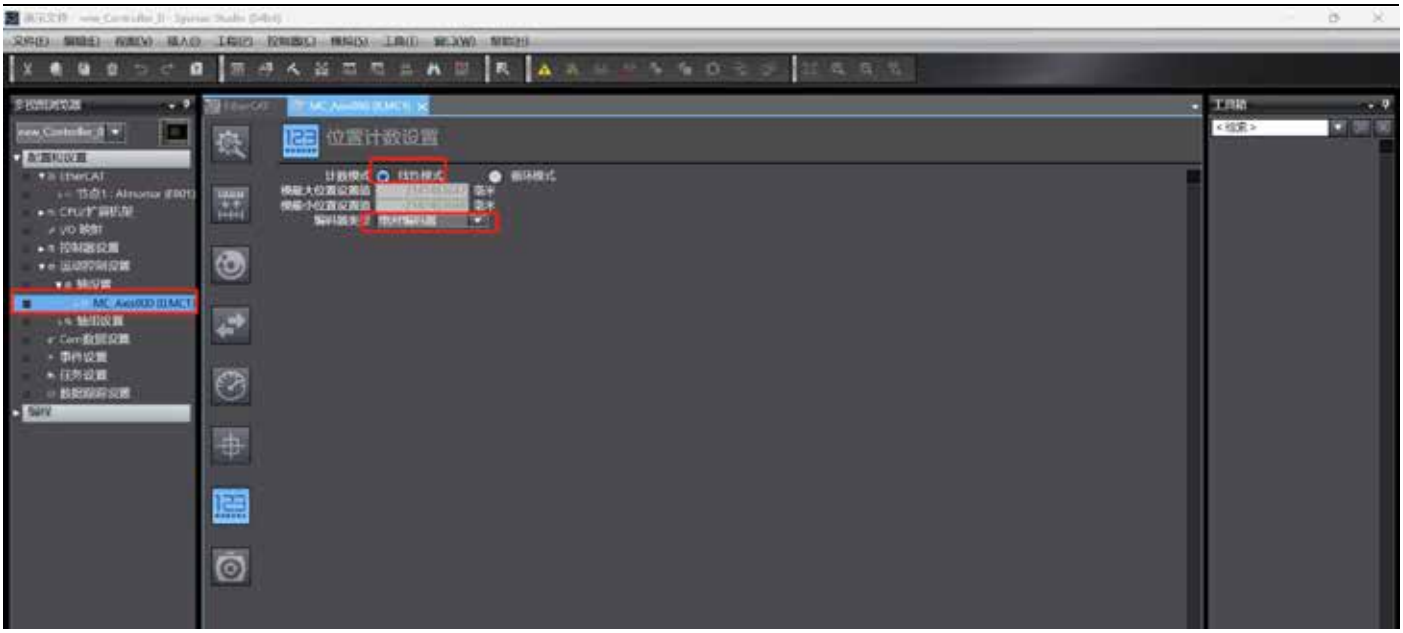
The changed configurations are: the return to Origin method is configured as the zero preset method; The position count mode is linear mode; And the encoder type is absolute encoder.

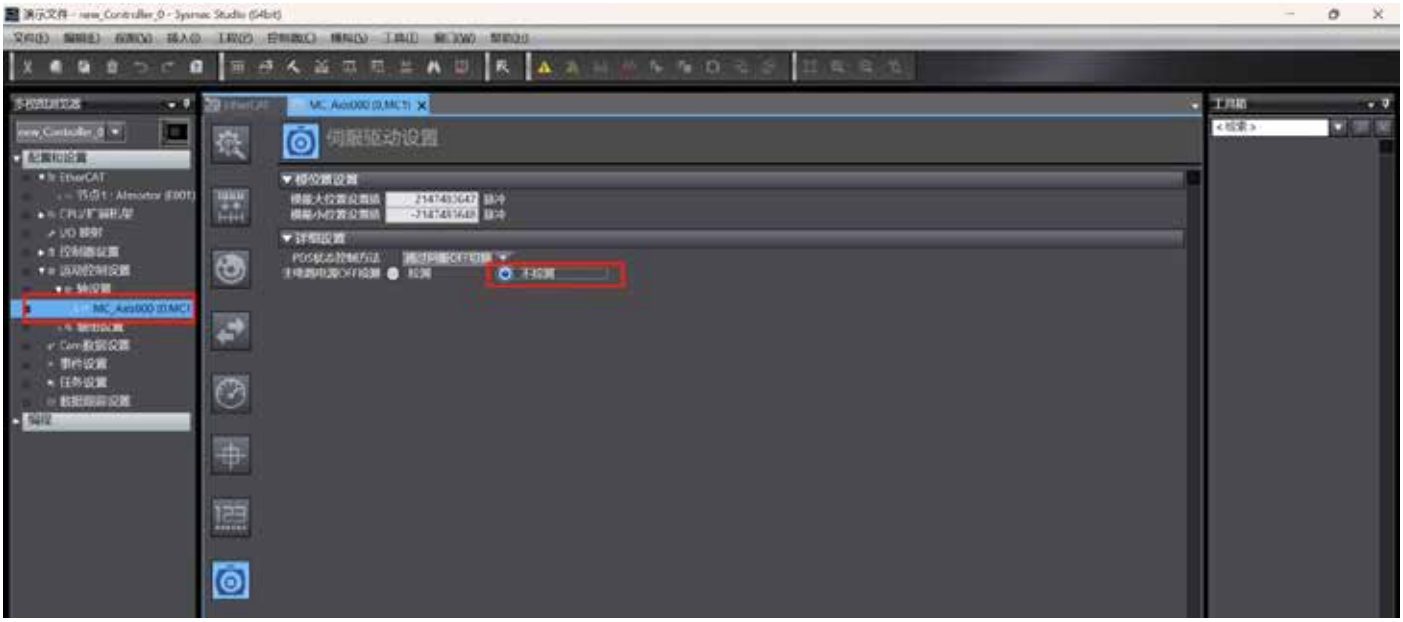



其它操作设置


限位设置


原点返回设置

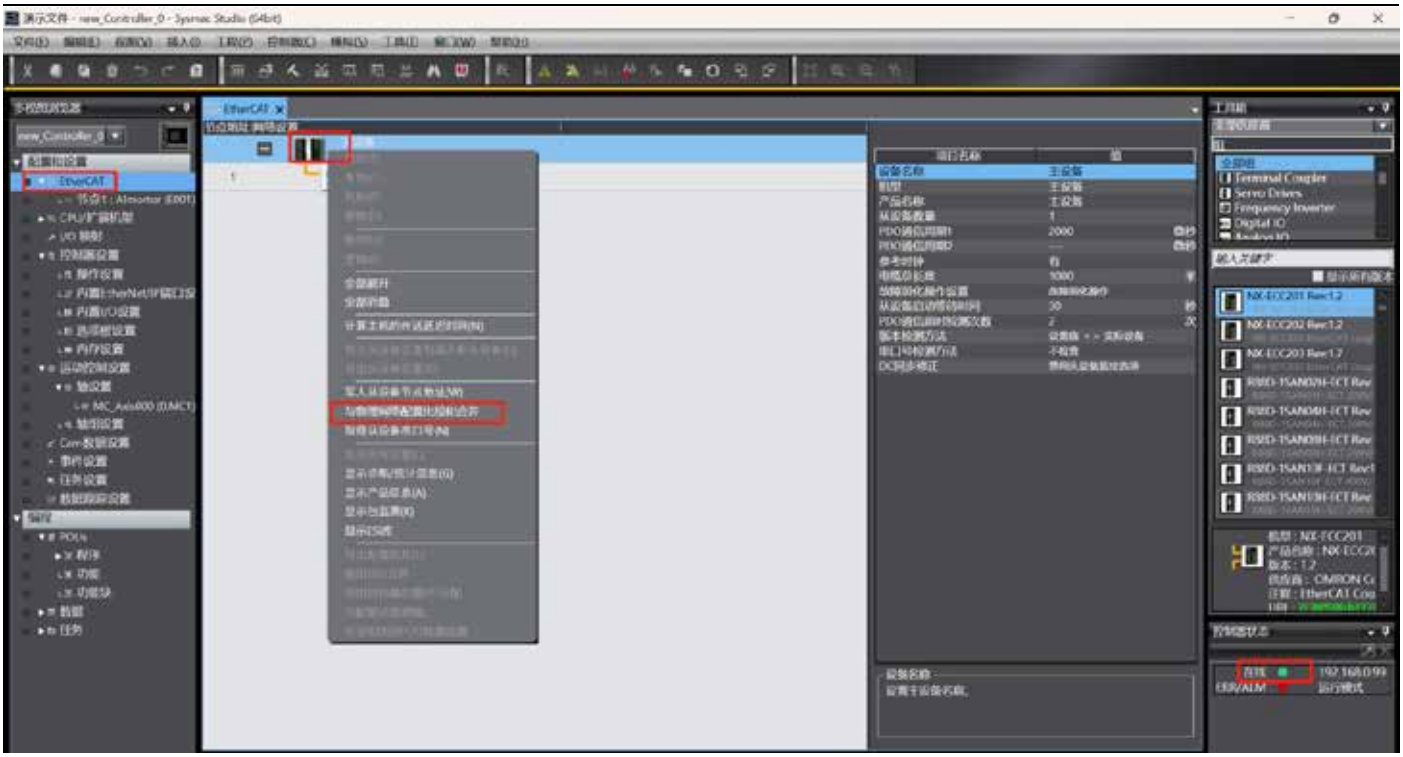




3.4.3 Online node assignment

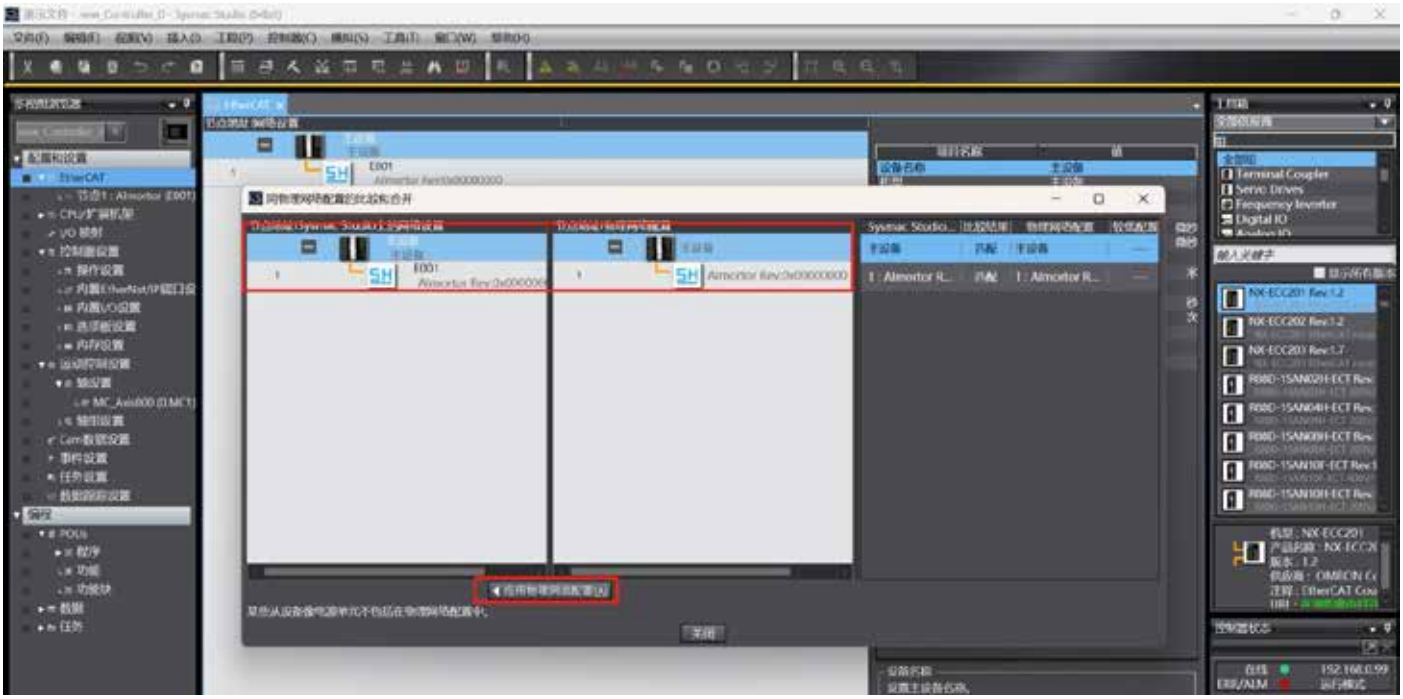
1) First, connect the PLC to the correct communication and be in the "online" state.

Then right-click on the "EtherCAT" master device and click the drop-down "compare and merge with physical network configuration". PLC will scan the actual slave stations on the network.



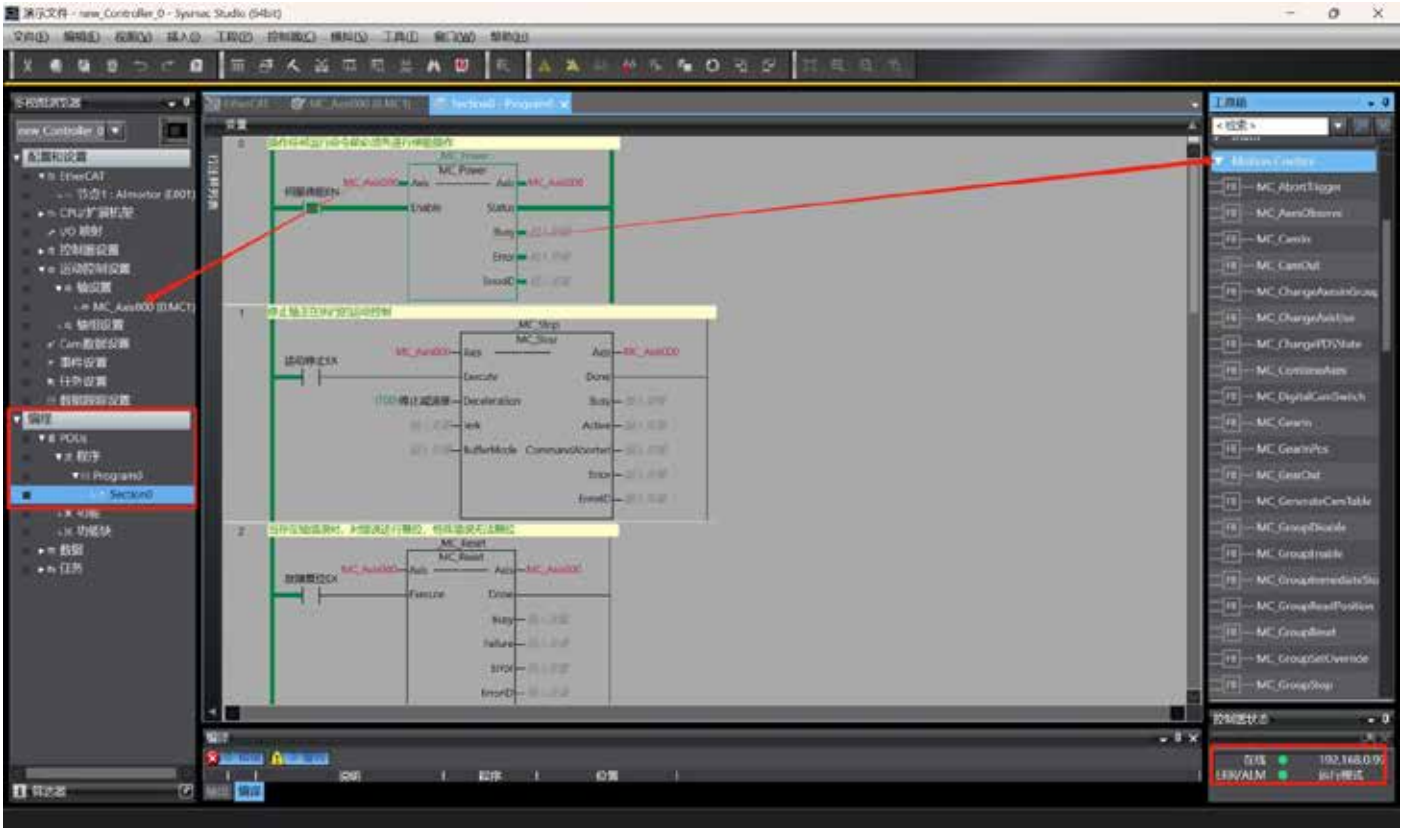
2) If the Slave node address on the actual network is inconsistent with the address configured by the software, click Apply Physical Network Configuration and confirm. The software will then power down and restart the slave device to take effect on the configured address. The aim is to be practical. This may be related to the order in which multiple slave stations are connected on the network.

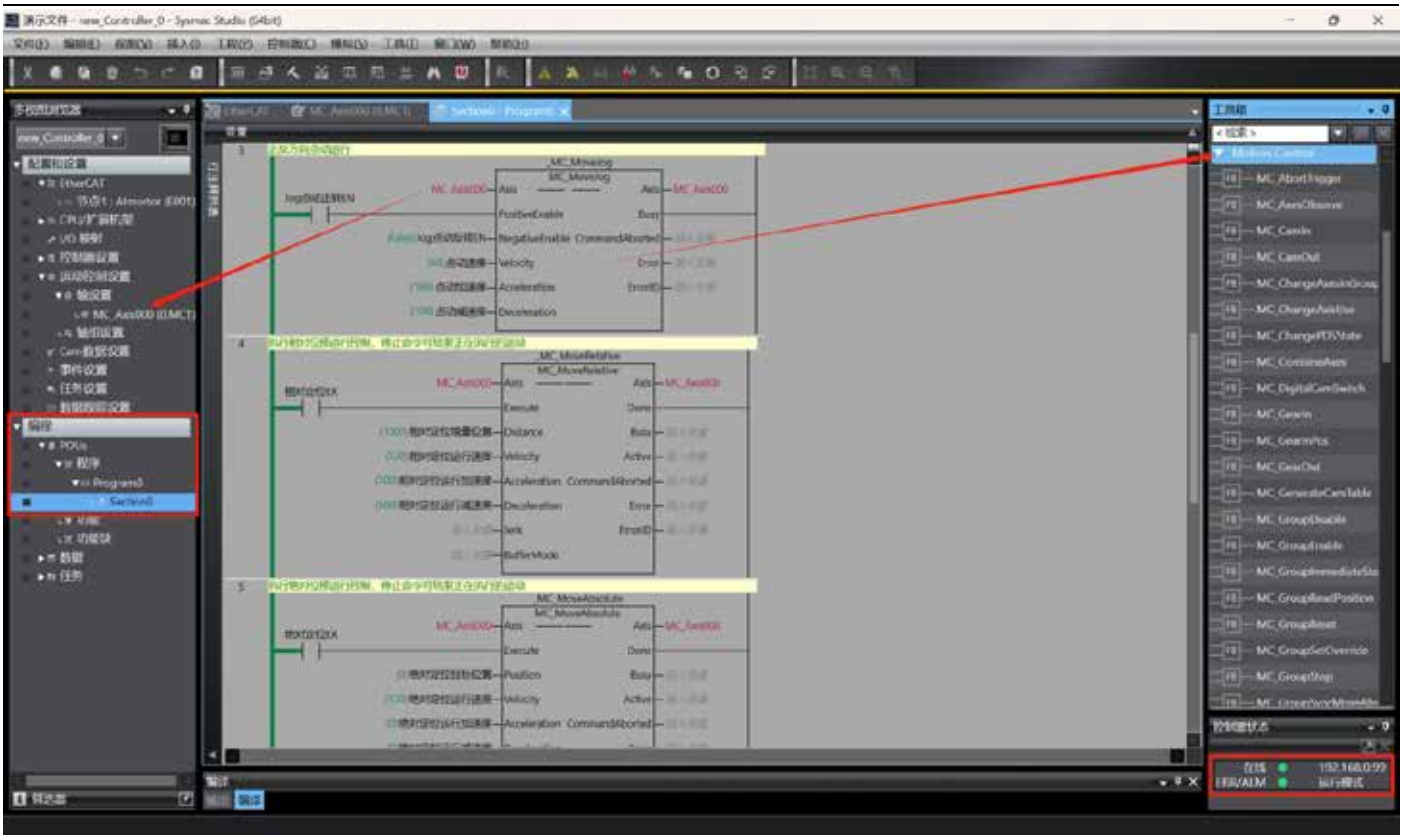
If the Slave node of the physical network is not displayed, check whether the slave is powered on and whether the network cable is connected correctly.

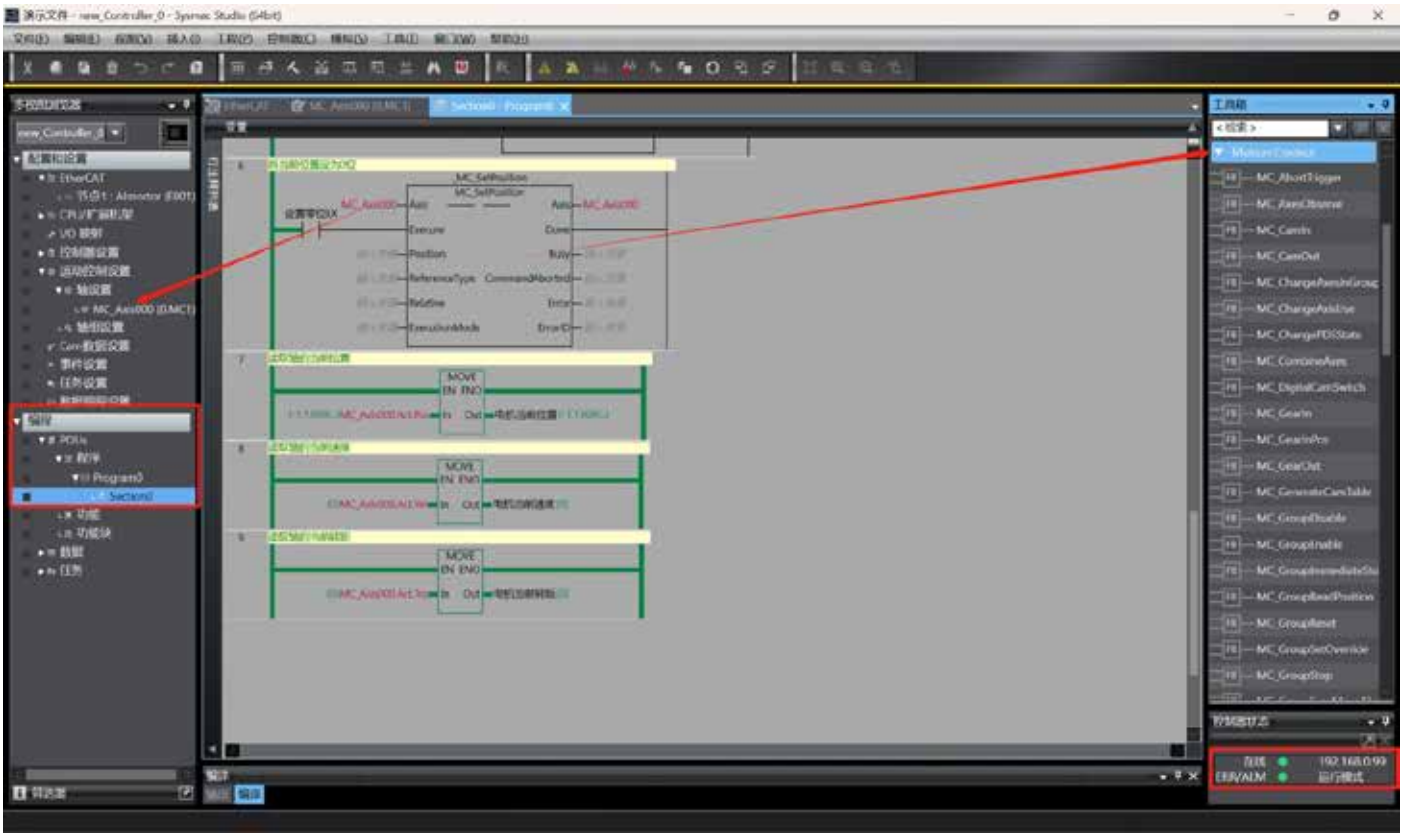


3.4.4 Programming Control

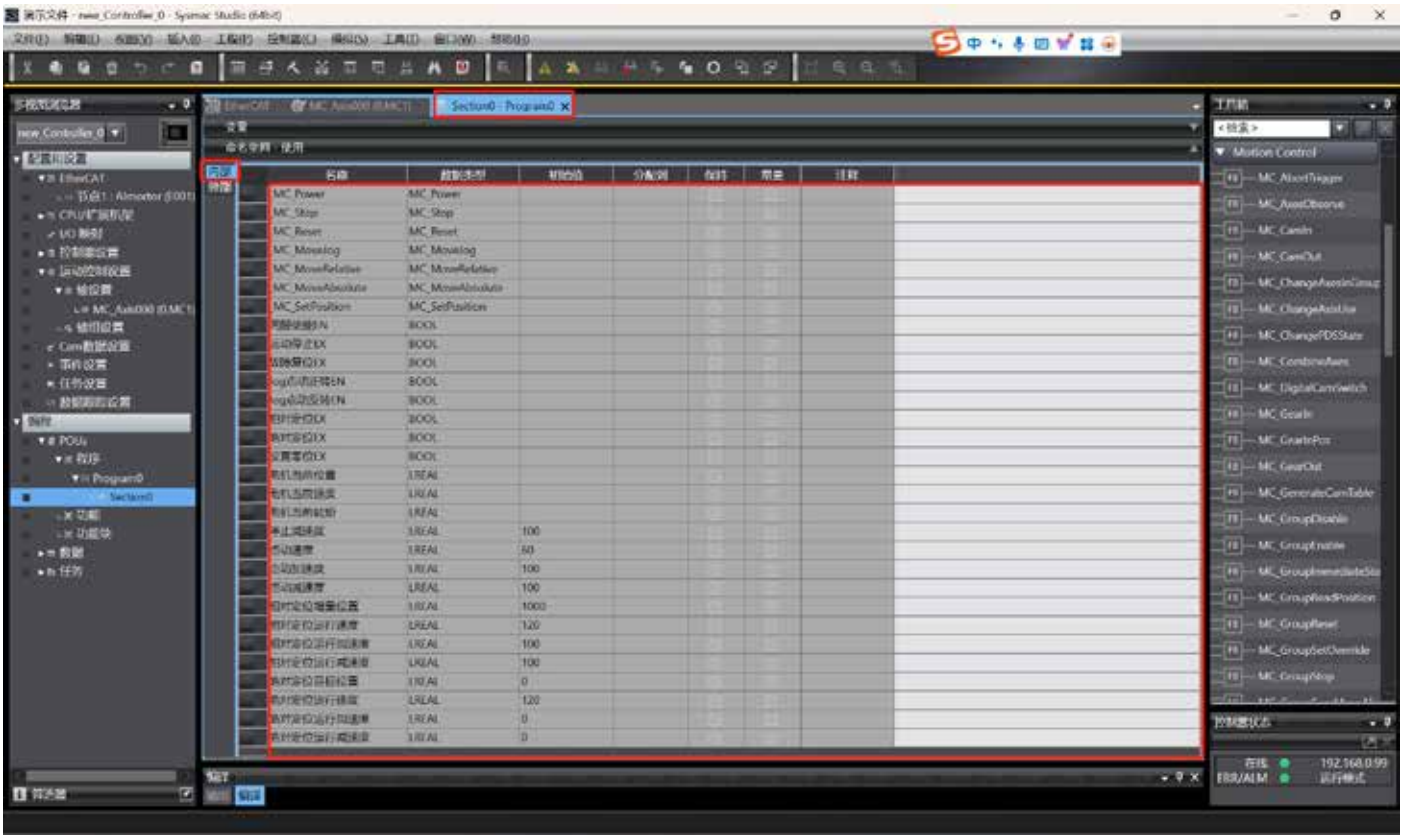
The program reference is demonstrated by programming with several commonly used motion commands. The actual application is based on the process requirements. After offline write good program, compiled through, to download. At the lower right corner of the observation window, the communication between the PLC and EtherCAT is normal. Control tests can be conducted.







Demonstrate the use of the variable table



4、Fault query

4.1 Fault code table

Description	Manufacturer fault code H0B_34 parameter	CIA402 fault code 603Fh Object t	Alarm lamp flashing times	Alarm Type
No fault	0x0000	0x0000	green	-
Parameter id is out of range	0x0101	0x6320	10red1green	NO.1
Parameter CRC error	0x0102	0x6320	12red1green	NO.1
Registration parameter CRC error	0x0104	0x6320	12red1green	NO.1
Internal Program Exception Triggers Watchdog	0x0105	0x6320	11red1green	NO.1
DI function repeat assignment	0x0130	0x6320	12red1green	NO.1
Hardware overcurrent	0x0201	0x2312	4red1green	NO.1
ia/ic quiescent current bias calibration failed	0x0208	0xFF00	12red1green	NO.1
Software overcurrent	0x0207	0x2311	4red1green	NO.1
Power-on flying car	0x0234	0xFF00	12red1green	NO.1

Encoder data exception	0x0A33	0x7306	9red1green	NO.1
Main circuit electrical overvoltage	0x0400	0x3210	3red1green	NO.1
Undervoltage of main circuit	0x0410	0x3220	3red1green	NO.1
Motor overload	0x0620	0x3230	4red1green	NO.1
Radiator overheating	0x0650	0x4210	6red1green	NO.1
Excessive positional deviation	0x0B00	0x8611	2red1green	NO.1
Zero return mode does not match	0x0668	0xFF00	8red1green	NO.2
Return to Origin timeout	0x0601	0x8610	8red1green	NO.2
Emergency stop	0x0900	0x5442	12red1green	NO.3
Forward overpass Warning	0x0950	0x5443	12red1green	NO.3
Negative overpass warning	0x0952	0x5444	12red1green	NO.3
Encoder battery failure	0x0731	0x7306	12red1green	NO.2
Encoder multi-turn count error	0x0733	0x7306	12red1green	NO.2
Encoder multi-turn count overflow	0x0735	0x7306	12red1green	NO.2
Encoder battery warning	0x0730	0x7307	12red1green	NO.3
CAN communication connection interrupted	0x0D03	0x8130	12red1green	NO.2
Changed parameters need to be powered on again to take effect.	0x0941	0xFF00	7red1green	NO.3
EtherCAT network exception	0x0E08	0x0E08	12red1green	NO.2

Manufacturer's fault code, one fault information corresponds to one fault code, which is unique;

The fault code of CIA402 refers to the provisions of cia402. When the fault information beyond the provisions of CIA402 is indicated by 0xFF00 fault code, it is not unique;

Because the integrated motor has no display panel, the alarm information can be judged by referring to the flashing change of the indicator light without connecting the debugging software, which is not unique;

4.2 Fault query method

The following three fault query methods need to configure the corresponding Object ts in the master station system. Read the status or value of the corresponding Object t through the master station to query the fault.

1、Query by CIA402 status word

CIA402 status word 6041h bit3 bit is a fault indication bit, = 1 indicates that there is a current fault; = 0 indicates that there is no current fault.

Note: This method can only determine whether the fault, can not reflect the specific fault information.

2、Query by CIA402 fault code

CIA402 fault code 603Fh, when there is a fault, the value of 603Fh is the corresponding Cia402 standard fault code.

Note: CIA402 fault code is not unique and can only represent the approximate type of fault.

3、Query by manufacturer parameter (recommend)

Parameter No.: H0B-33, select the first N fault records to query; Object dictionary index: 200Bh, sub-index 22h, read/write;

Parameter number: H0B-34, display the fault codes of the first N times selected; Object dictionary index: 200Bh, sub-index 23h, read only;

When the H0B-33(200B-22h) is set to 0, once a fault occurs, the H0B-34(200B-23h) will display the current manufacturer-defined fault code;

When the H0B-33(200B-22h) is set to 1, the system will query the manufacturer-defined fault code generated in the previous history and display it in the H0B-34(200B-23h);

When the H0B-33(200B-22h) is set to 2, the system will query the manufacturer-defined fault codes generated in the previous 2 times in history and display them in the H0B-34(200B-23h);

When the H0B-33(200B-22h) is set to 9, the system will query the manufacturer-defined fault codes generated in the first 9 times of history and display them on the H0B-34(200B-23h);

Note: The fault record is stored up to 9 times. The sampling first-in first-out method is used. Each fault corresponds to a unique fault code. If you need to query detailed fault information, recommend this method.