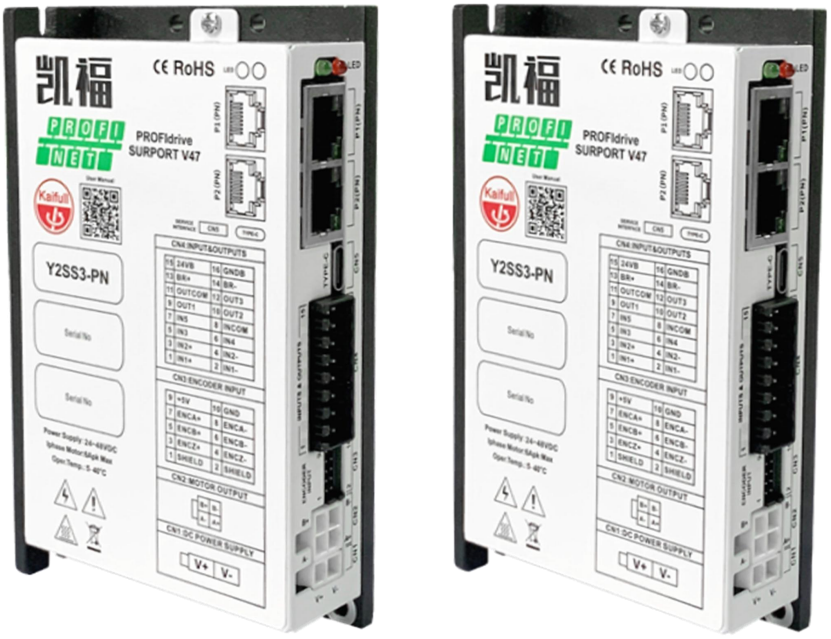
Y2SS3-PN

PROFINET Bus Stepper Driver

User Manual



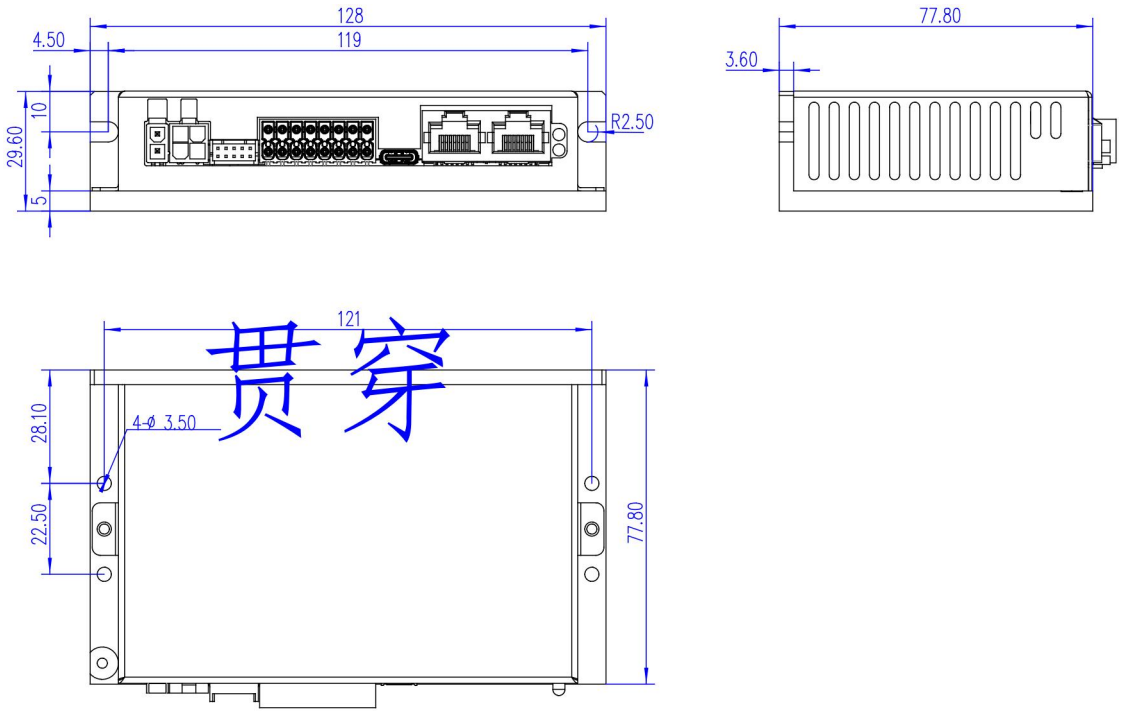
Guangdong Kaifull Electronics Technology Co., Ltd.

Contents

1 Foreword

- Thank you for choosing Kaifull's product.
- This manual describes the use methods and safety precautions of the product.
- Please read this user manual carefully and use this product correctly and safely.
- After reading, please save it at a suitable place for easy access at any time.
- For technical support, please dial 400-960-1069 or +86-769-23033384.

2 Installation Dimensions

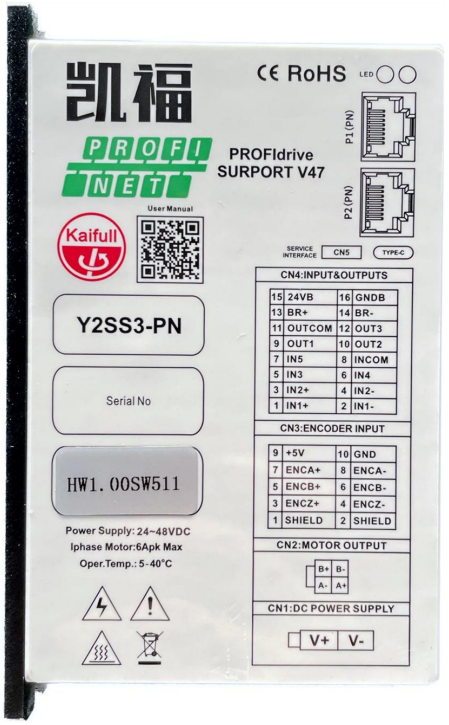


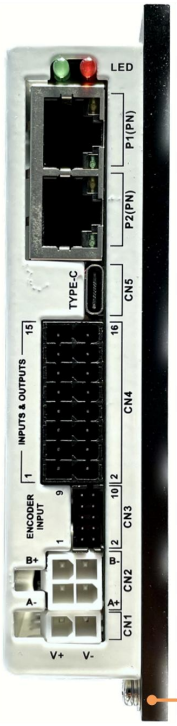
3 Technical Specifications

Technical Specifications		
Installation Dimensions	128 × 77.8 ×29.6 mm	
Input power	24~60V DC（±15%）	
Current output	0.1-7A (peak)	
Adaptive motor	Two-phase stepper motors of size 86 and below	
Open loop/closed-loop control	Compatible with open-loop or closed-loop control	
Encoder interface	Supports up to 5000-line incremental encoders (20000 pulses/revolution)	
Control mode	PROFINET bus communication control	
Communication interface	USB-C	Connect to PC for parameter settings, status monitoring, etc.
	RJ45 network port	PROFINET communication interface
Control signal	Digital input signal	Differential: 3 pcs, single end: 2 pcs; opto-isolator; the common port supports 5～24VDC
	Digital output signal	3 open collector outputs; opto-isolator; maximum output 100mA@30V;
		1 brake output, maximum output 100mA@30V
Recommended service environment	Temperature	0 ~ +55 ℃
	Humidity	0~ 90%RH below
	Altitude	1000 m below
	environment	No corrosive gases or dust.
		The product shall not come in contact with water and oil.
Dielectric strength	AC1.5KV between ground wires, capable of withstanding voltage for 1 minute	
Protection grade	IP20	

4 Wiring

4.1 Real product photo





驱动器状态指示灯

P1、P2：PN通讯网口

CN5：PC软件连接口

CN4：数字输入/输出

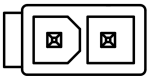
CN3：编码器输入

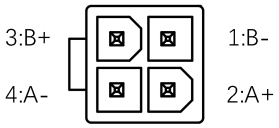
CN2：电机动力线接口

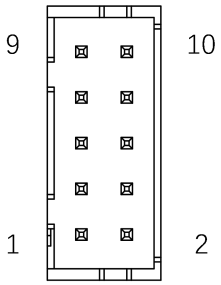
CN1：电源输入

接地螺丝

4.2 Interface Definitions

CN1: Power supply (24~60VDC)		
CN1.1	V-	 2:V+ 1:V-
CN1.2	V+	

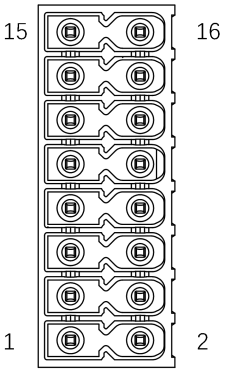
CN2: Motor connection				
CN2.1	B-	CN2.2	A+	 3:B+ 1:B- 4:A- 2:A+
CN2.3	B+	CN2.4	A-	

CN3: Encoder input				
CN3.1	Cable shielded wire	CN3.2	Cable shielded wire	 9 10 1 2
CN3.3	ENCZ+	CN3.4	ENCZ-	
CN3.5	ENCB+	CN3.6	ENCB-	
CN3.7	ENCA+	CN3.8	ENCA-	
CN3.9	+5V output	CN3.10	GND	

CN4: Digital I/O input

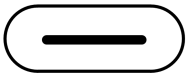
Note: "[ ]" is an example of commonly used NPN type wiring method

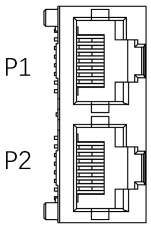
CN4.1	IN1+	Digital input 1 positive pole
CN4.2	IN1-	Digital input 1 negative pole
CN4.3	IN2+	Digital input 2 positive pole [connected to+24V]
CN4.4	IN2-	Digital input 2 negative pole [home position signal]
CN4.5	IN3	Digital input 3 [positive limit]
CN4.6	IN4	Digital input 4 [negative limit]
CN4.7	IN5	Digital input 5
CN4.8	INCOM	Digital input 3/4/5 common terminal [connected to+24V]
CN4.9	OUT1	Digital output 1
CN4.10	OUT2	Digital output 2
CN4.11	OUTCOM	Digital output 1/2/3 common terminal
CN4.12	OUT3	Digital output 3
CN4.13	BR+	Power-off brake output 24V+[brake line+]
CN4.14	BR-	Power-off brake output 0V [brake line -]
CN4.15	24VB	Power-off brake supply input 24V+ [Connected to+24V]
CN4.16	GNDB	Power-off brake supply input 0V [Connected to 0V]



CN5: PC software debugging interface

Use USB TYPE-C connecting cable to connect to computer debugging software

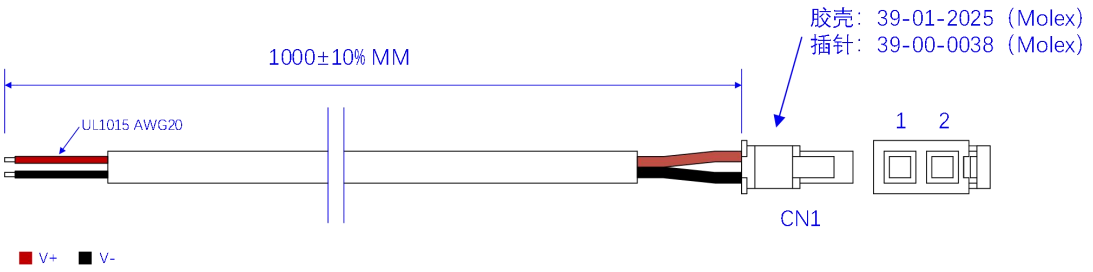


PROFINET communication network connection			Remark
P1	RJ45 Network port		P1 serves as an IN port to connect to the upper computer/previous slave station device, and P2 serves as an OUT port to connect to the next slave station device.
P2	RJ45 Network port		

4.3 Wiring method

4.3.1 Power Connection

The Y2SS3-PN driver product comes with a 1m-long power cord. When you connect the power supply, connect the red wire of this power cord to the switch power supply V+and the black wire to V -.



➢ **Selecting the appropriate power supply:**

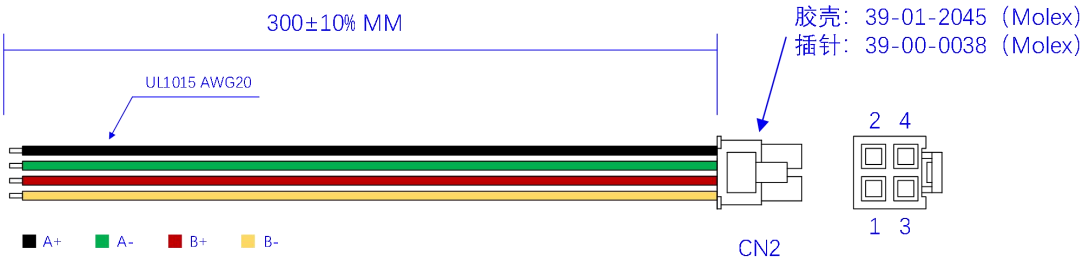
The following are recommendations for selecting the power supply when using different motors:

Motor flange (MM)	Supply voltage	Supply current
20/35	24V	≥1.0A
42	24V	≥2.0A
57/60	24-36V	≥4.5A
86	36-48V	≥6A

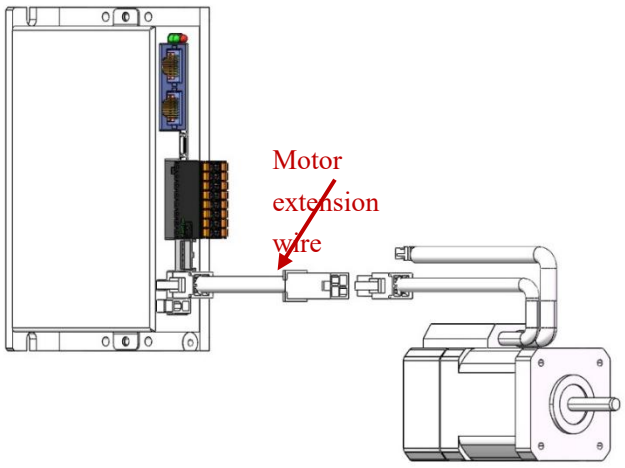
- Be careful not to connect the power supply reversely, as it may cause damage to the drive and result in no warranty coverage
- When the motor 57 and above is used and the motor is operating at a high speed, it will generate a large reverse electromotive force. At this time, use a higher-voltage power supply to improve the high-speed performance of the motor.

4.3.2 Motor Connection

The Y2SS3-PN driver product comes with a 4-core motor connection line with a length of 30cm, which are black, green, red, and yellow, corresponding to A+, A -, B+, and B - of the driver.

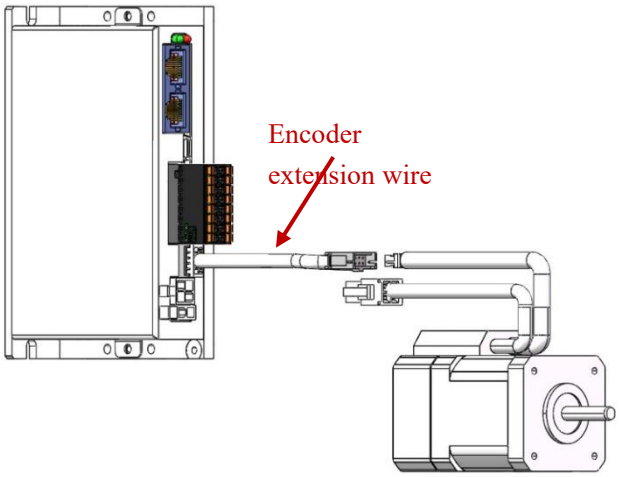


- If you use an open-loop motor, follow the wiring instructions in the motor specification to connect the motor lead to the connection line shown in the above figure.
- If you use a Kaifull closed-loop motor, the motor power line can be connected to the driver through the motor extension line (optional) (as shown in the figure below).



4.3.3 Encoder Connection

When a Kaifull closed-loop motor is used, the motor encoder cable can be connected to the driver through the encoder extension line (optional).



➢ **Accessory Information**

Type	Model	Length
Motor extension wire	2103-100	1 m
	2103-300	3 m
	2103-500	5 m
Encoder extension wire	E208-100	1 m
	E208-300	3 m
	E208-500	5 m

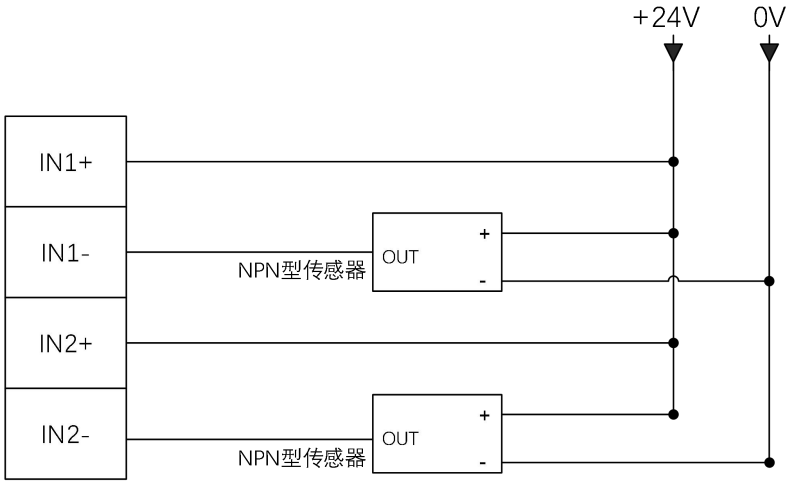
4.3.4 Digital input connection

The default definition of each input point is as follows

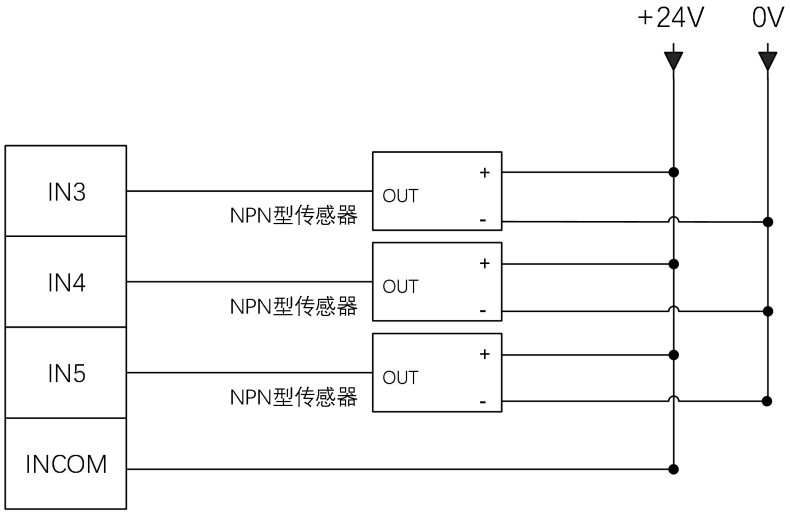
Input point	Definition
IN1	**Invalid**
IN2	**Home position**
IN3	**Positive limit**
IN4	**Negative limit**
IN5	**Invalid**

- **Connecting NPN type sensor**

➢ IN1、IN2：

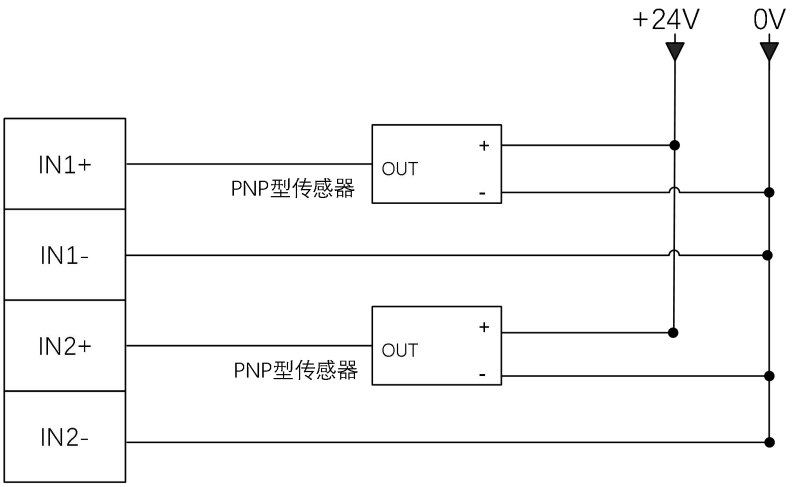


➢ IN3、IN4、IN5：

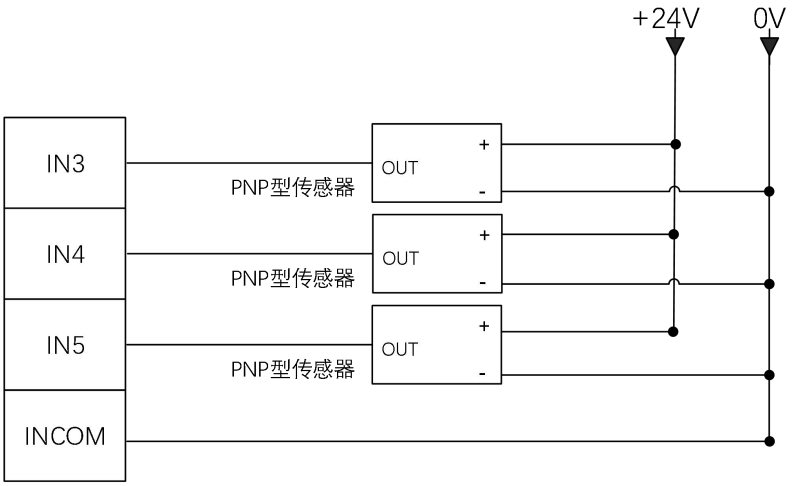


- **Connecting PNP type sensor**

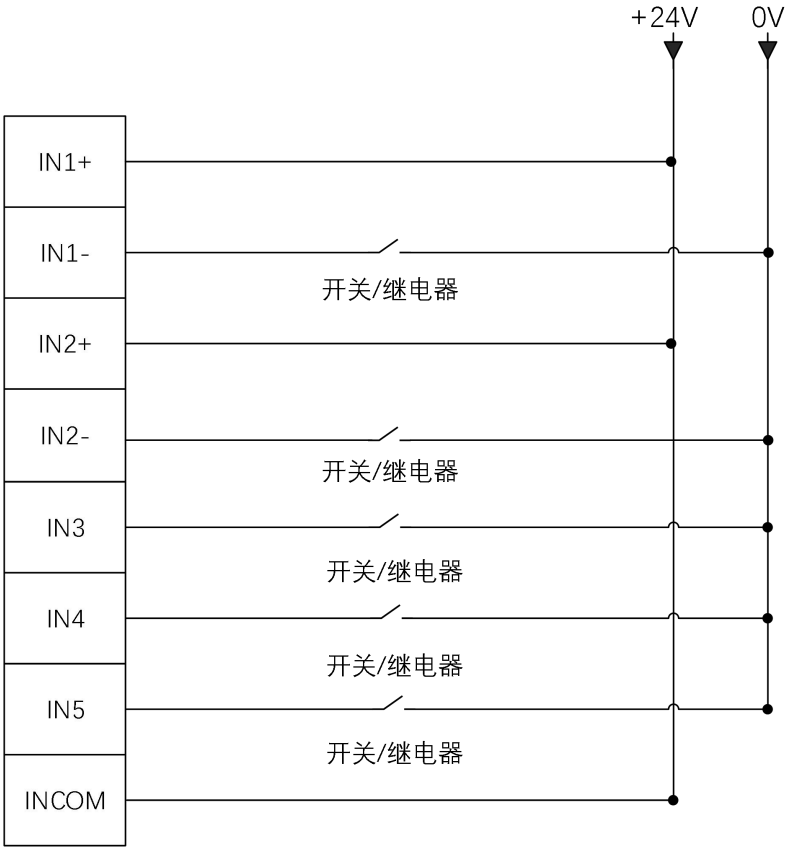
➢ IN1、IN2:



➢ IN3、IN4、IN5:



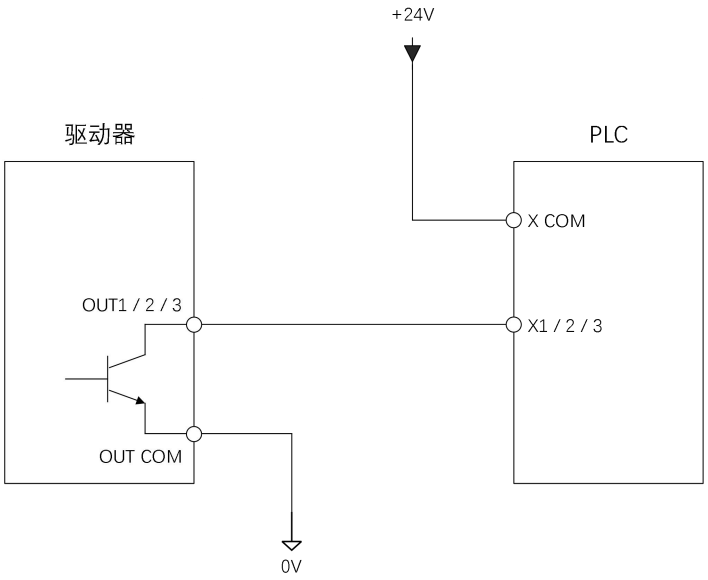
- **Connection to switches/relays**



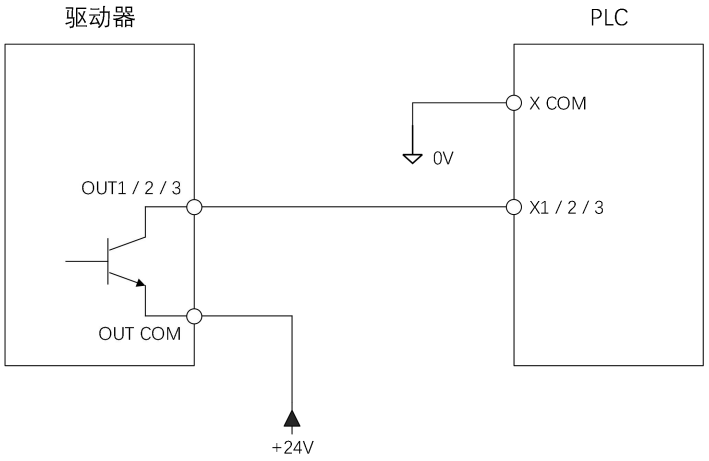
4.3.5 Digital Output Connection

The output interface of Y2SS3-PN is an open-drain output, and the output level depends on the connection of the output common terminal OUTCOM.

- **Low level output connection method:**



- **High level output connection method:**



4.3.6 Brake output wiring

A power brake is an electromagnetic brake device installed at the tail end of a stepper motor. A stepper motor with a brake is commonly used on mechanisms that move vertically. It can provide holding force in the event of sudden loss of equipment power to prevent the vertical mechanism from falling due to its own gravity.

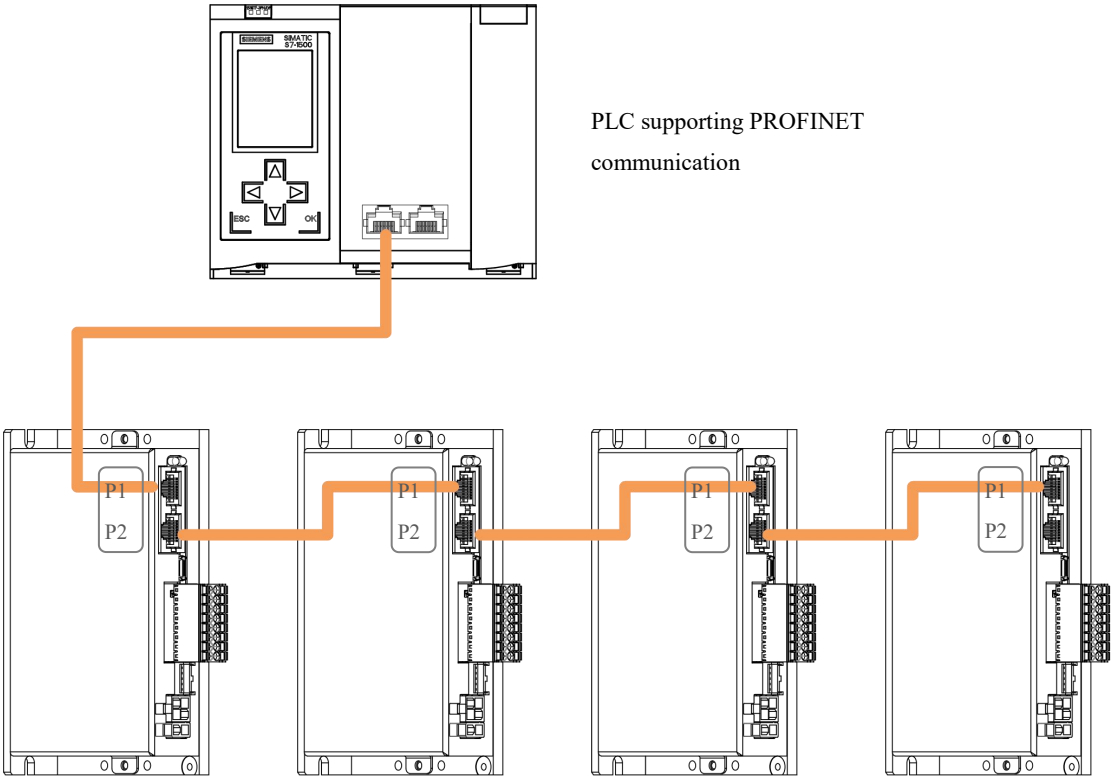
An automatic control brake function is built in the Y2SS3-PN driver, and it can help turn on the brake at the moment when the motor is enabled, and turn off the brake at the moment when the driver gives an error alarm and the motor is disabled, and users do not need to manually control it.

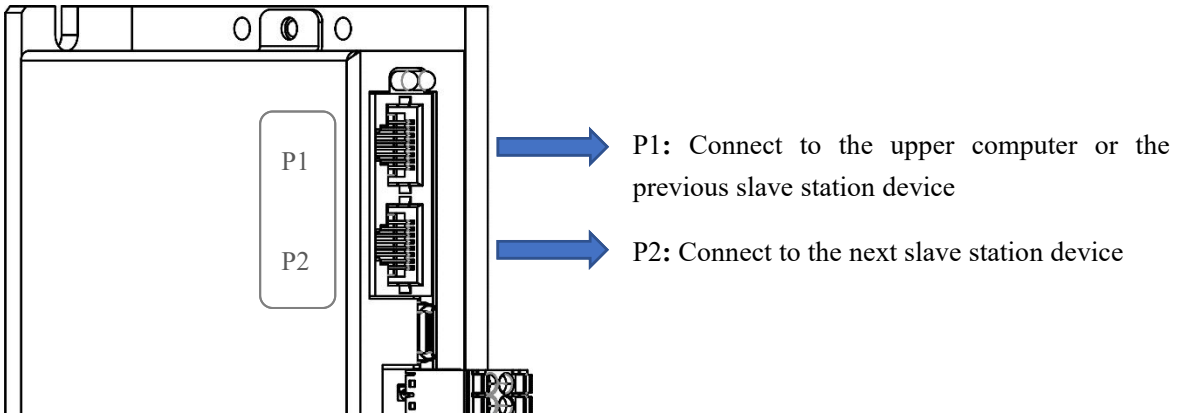
- **Brake wiring method:**

- Users need to prepare a 24V power supply to power "24VB" and "GNDB"
- The maximum output of the brake output port is 500mA@30V , and it can drive the brake device without being connected to a relay
- It is recommended to use cables of 0.3mm ² and above

4.3.7 Communication network connection

- Network topology:





- Category-5 (CAT.5E) or higher-specification network cables are recommended

5 PROFINET Communication

5.1 PROFINET Overview

PROFINET is an industrial automation fieldbus protocol based on Ethernet, proposed and managed by Profibus International (PI).

PROFINET is divided into two types of real-time periodic communication with different performances. One type is called Real-Time Communication (RT), which does not require time synchronization and generally requires a response time of 5-10 ms, mainly used for factory automation; the other type is Isochronous Real-Time Communication (IRT), mainly used in the situations with strict time synchronization requirements, such as electronic gears, motion control, etc. In IRT communication mode, the response time of data can be less than 1ms.

The Y2SS3-PN driver supports both RT and IRT communication.

5.2 Supported Messages

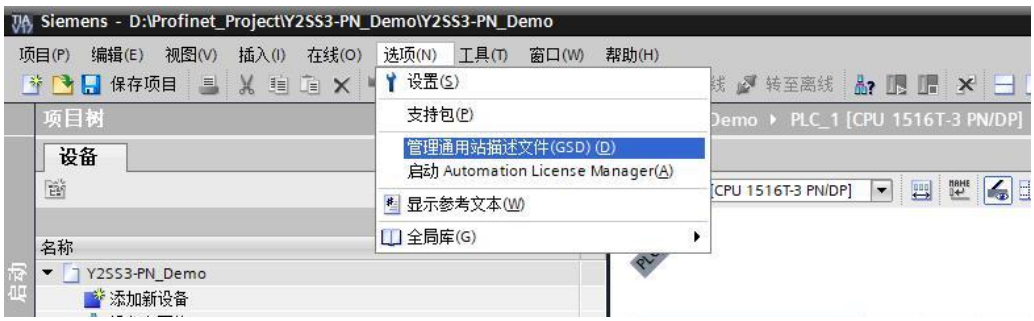
Y2SS3-PN supports the following standard messages

- Standard message 1: Speed control
- Standard message 3: Speed/position control (process axis, IRT communication)
- Standard message 111: Speed/position control

5.3 Configuration Application

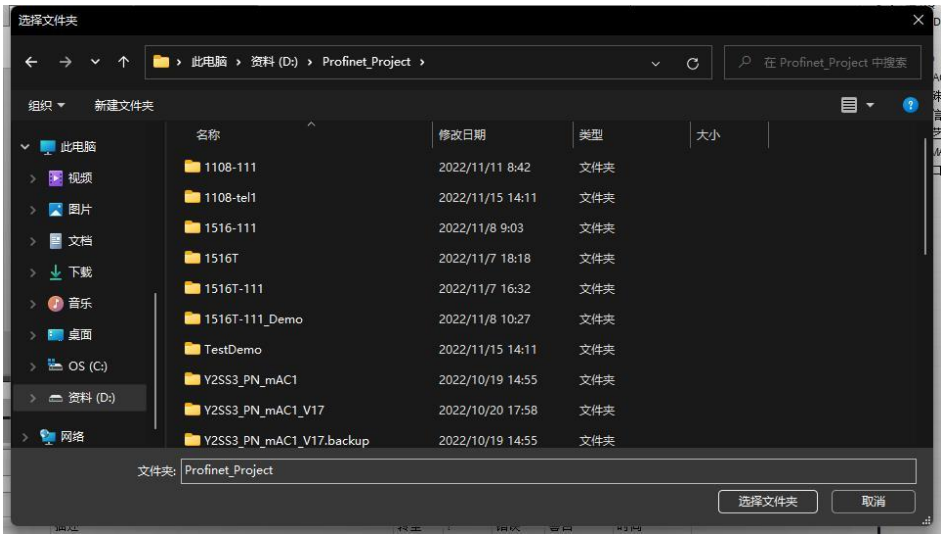
① Install GSD files

Click the menu bar "Option" - "Manage General Station Description Files (GSD)"

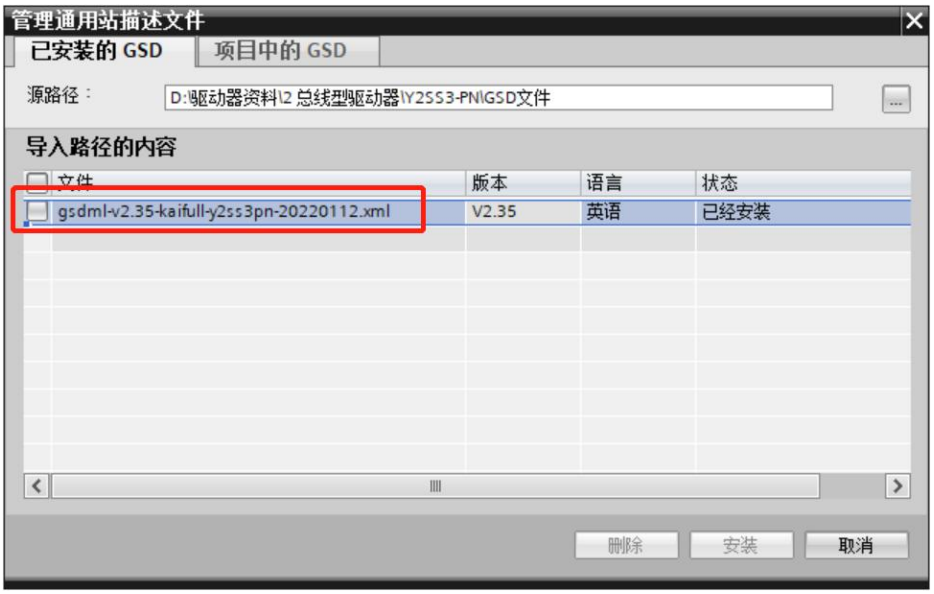


Click the button in the upper right corner to open the folder where the GSD file is located

Select a folder. Note: Select a folder, not a file

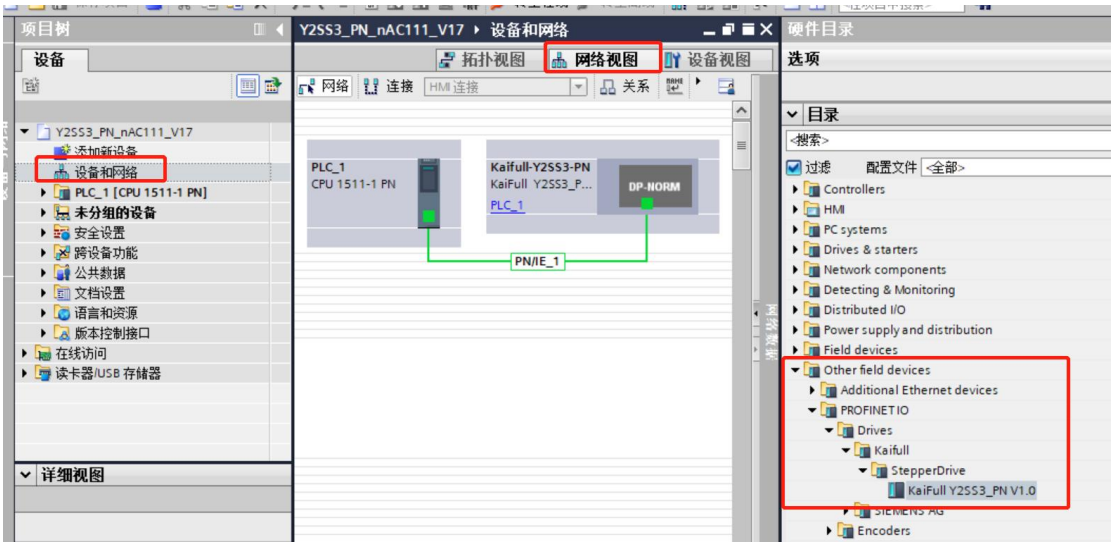


Check GSD File - Install. Note: According to the regulations of Botu Software, the GSD file names have a fixed naming rule and cannot be named arbitrarily; otherwise, an error will be reported when they are imported.
Example of wrong file name: gsdml-v2.35-kaifull-y2ss3pn-20220112 (1).xml



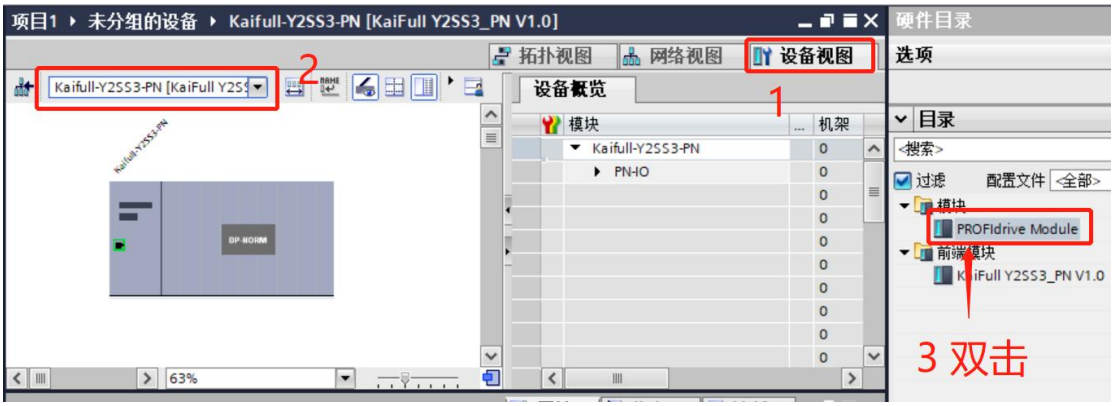
② Add a slave station device

In the Device and Network - Network view, find the Kaifull Driver Y2SS3-PN in the lower right corner, double-click to add it and connect to the PLC.

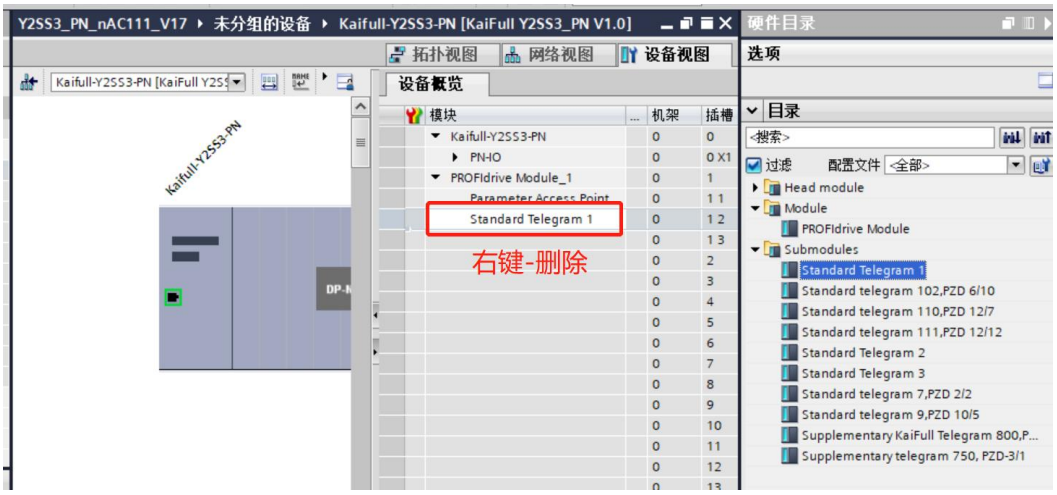


③ Add a message module

Find "PROFIdrive Module" in the lower right corner of the Device view and double-click to add it



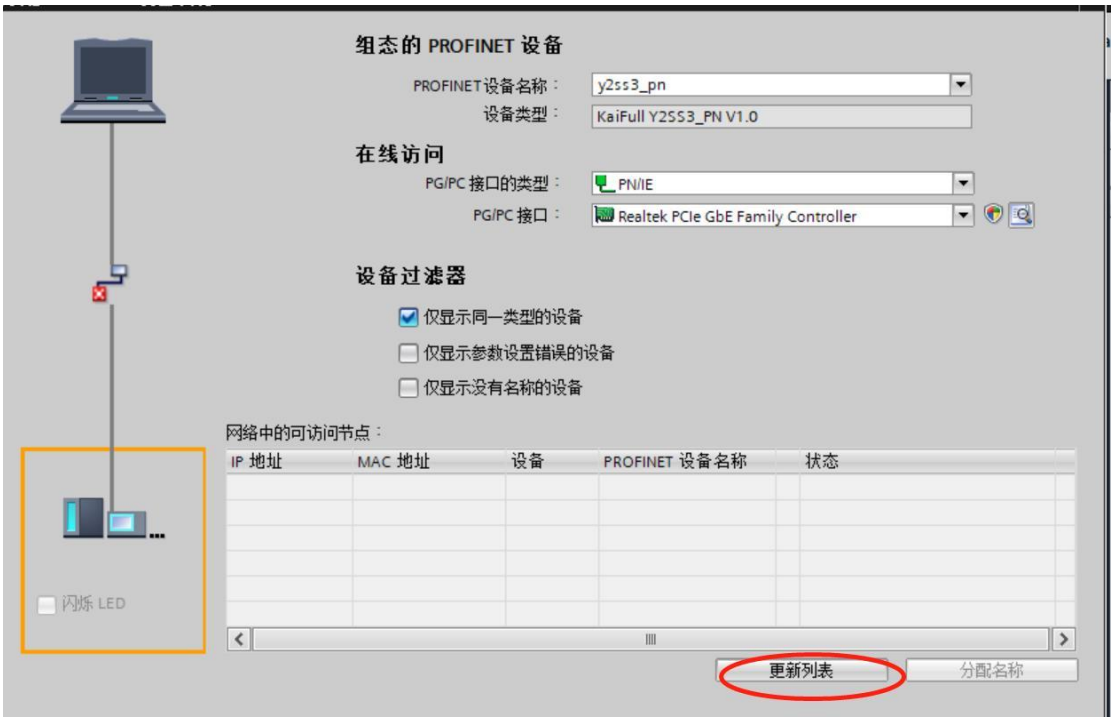
After adding successfully, the default sub-module is Standard Telegram 1, which is message 1. If needing to use other messages, you can right-click to delete message 1 and double-click the messages on the right to add them.



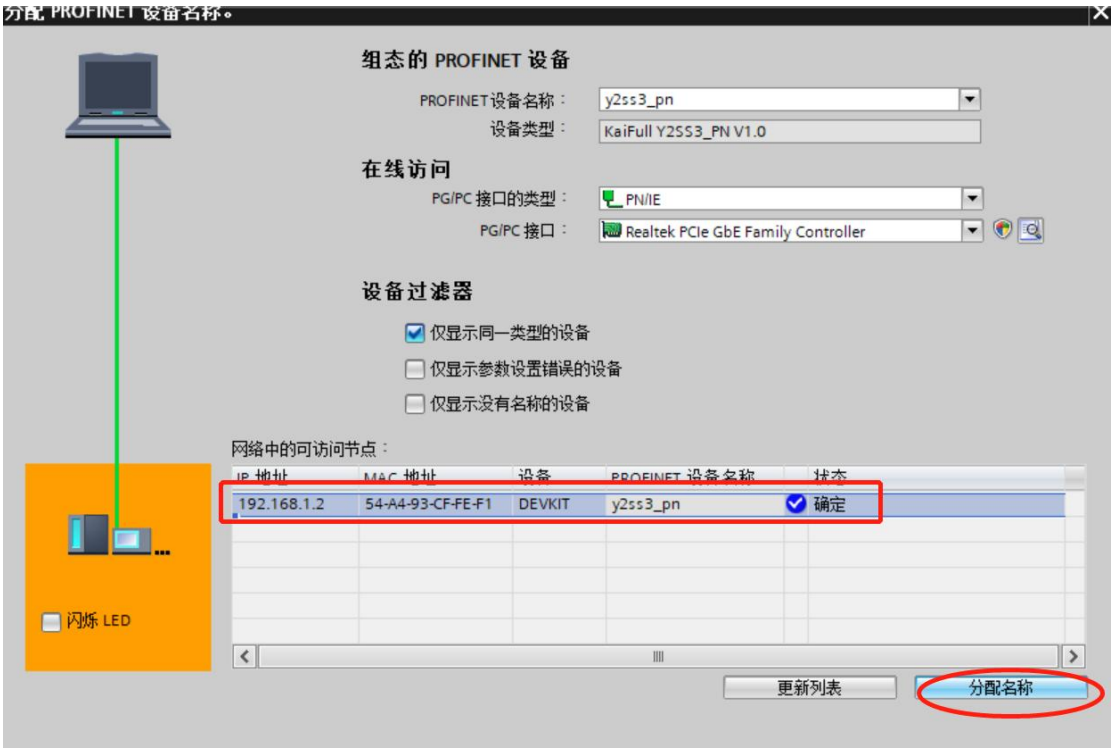
④ Assign device name

In this step, the PC, PLC, and driver need to be connected properly

In the pop-up window, select the correct network card and click "Update List" to start to search for drivers



Find the corresponding driver, click "Assign Name", and after signing successfully, the display status will be "✓ OK"



6 Message Application Examples

6.1 Driver Message Settings

Supported messages of drivers:

Message	Supported or not	Communication mode
1	√	RT
3	√	IRT
111	√	RT
Auxiliary message 800	√	RT

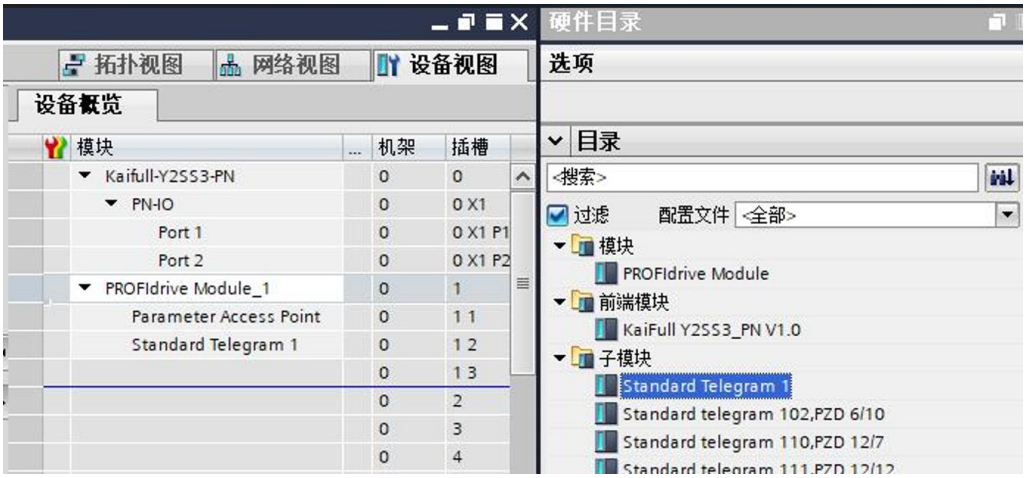
The message selected using Botu configuration needs to be consistent with the message set by the driver; otherwise, the communication will be abnormal

6.2 Message 1

Message 1 uses Profinet RT communication, and the motor speed control can be achieved using the function block SinaSpeed (FB285) in the driver library provided by Botu software.

6.2.1 Add message 1

Delete the original message in the device view, select message 1 in the lower right corner, and double-click to add it

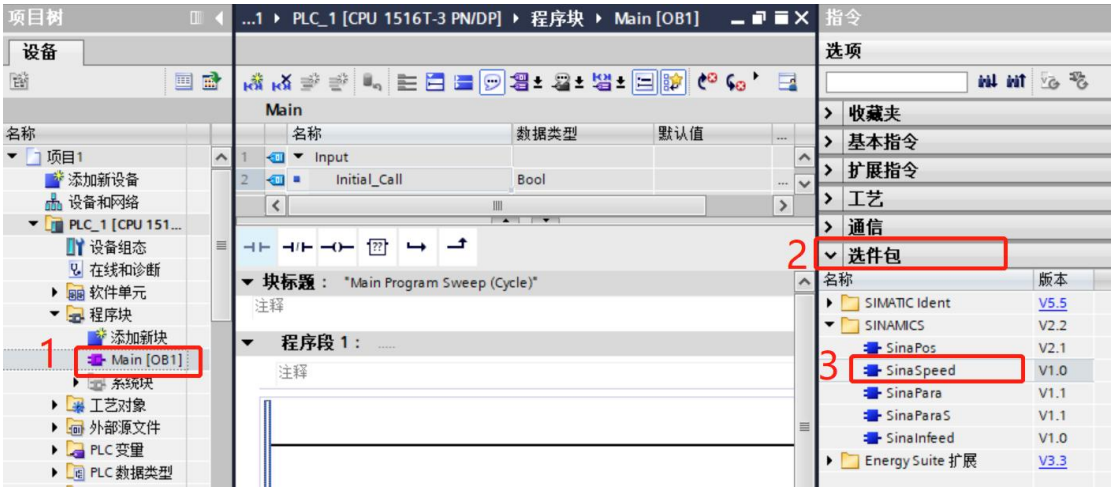


6.2.2 View hardware identifiers

The hardware identifiers are used in the following programming

6.2.3 Edit programs

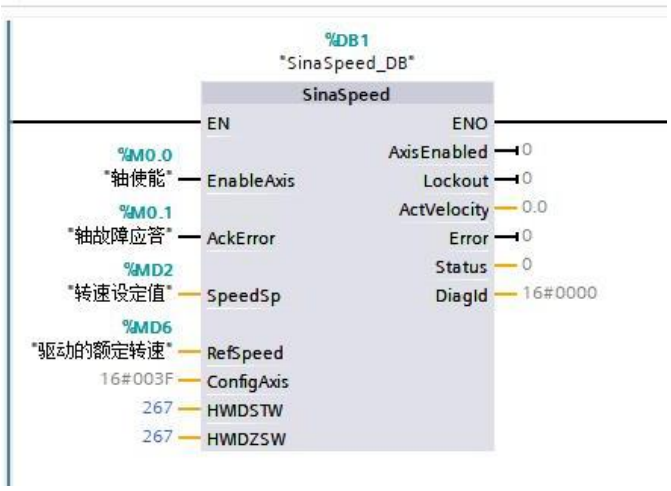
Find and add the SinaSpeed instruction (FB285) on the programming page



In the menu bar - "Help" - "Display Help", search for relevant instruction description and fill in parameters according to the instruction description.

SINA_SPEED 的输入接口

输入信号	类型	默认值	含义
EnableAxis	BOOL	0	"EnableAxis"=1 → 打开驱动
AckError	BOOL	0	轴故障应答 → AckFlt=1
SpeedSp	REAL	0.0[rpm]	转速设定值
RefSpeed	REAL	0.0[rpm]	驱动的额定转速 → p2000
ConfigAxis	WORD	3	更多信息参见预分配的ConfigAxis输入。
HWIDSTW	HW_IO	0	设定值槽中SIMATIC S7-1200/1500上的符号名称或硬件ID
HWIDZSW	HW_IO	0	实际值槽中SIMATIC S7-1200/1500上的符号名称或硬件ID



The parameter ConfigAxis does not need to be changed.

Fill in the above hardware identifiers for the last two variables "HWIDSTW" and "HWIDZTW"

6.2.4 Example demonstration

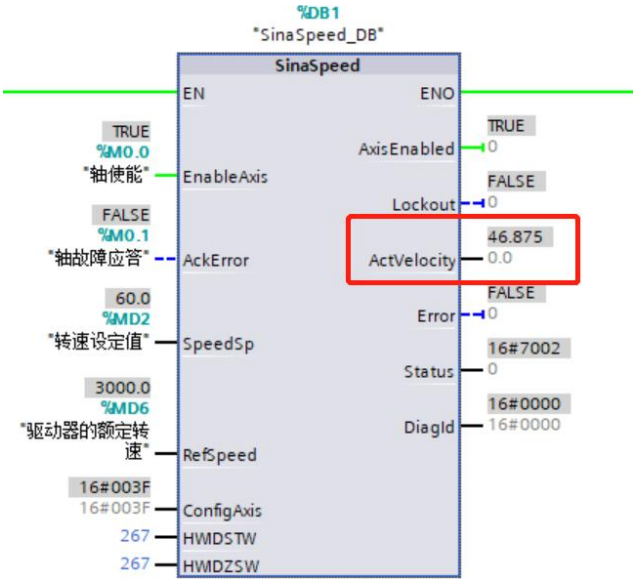
① EnableAxis=True, enable on motor

② The internal speed of the driver is based on 3000rpm, so RefSpeed is filled with a constant of 3000

③ After the motor shaft is enabled, modify the value of the speed parameter SpeedSp, and the motor can rotate. SpeedSp is in RPM (revolutions per minute).

SpeedSp value	Motor status
=0	Stop
>0	Rotate clockwise
<0	Rotate counterclockwise

Actual speed (RPM)=SpeedSp*(3000 / RefSpeed). For example, when SpeedSp=60 and RefSpeed=3000, speed=60 RPM.



The actual speed value ActVelocity indicates that the motor is rotating clockwise

6.3 Message 3

Message 3 adopts Profinet IRT communication, and during configuration, an axis process object needs to be added to control the motor (such as MC_Power, MC_MoveAbsolute, etc.) through Siemens PLC standard instructions.

6.3.1 Add message 3

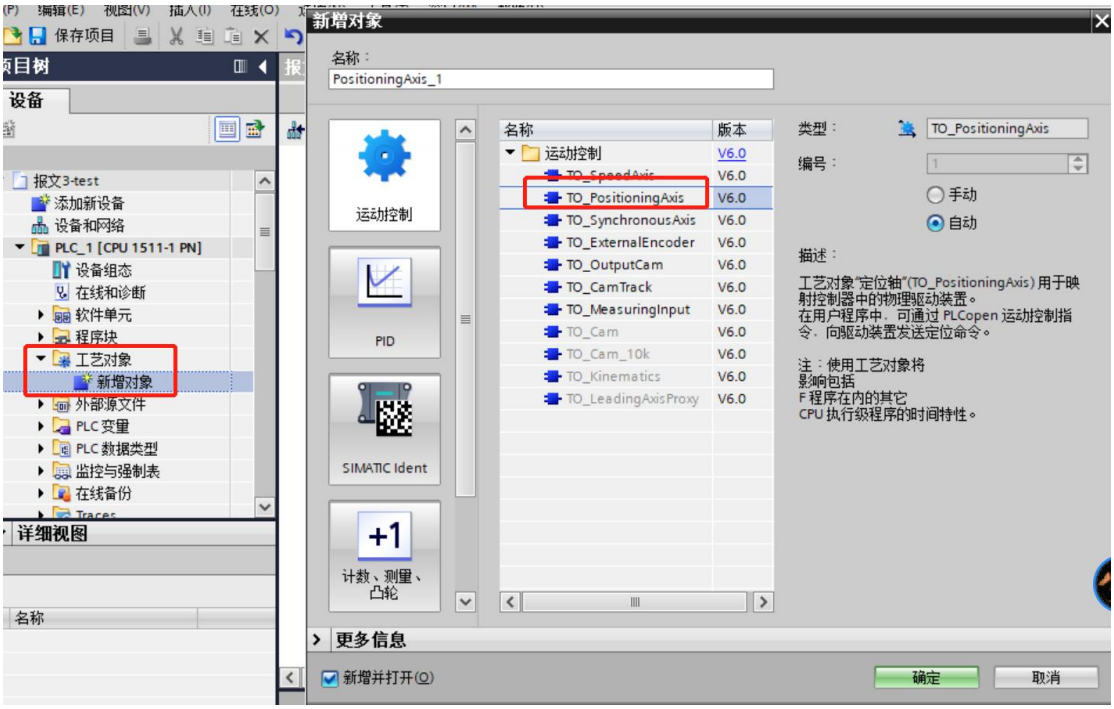
Delete the original message in the device view, select message 3 in the lower right corner, and double-click to add it

6.3.2 Topology view connection

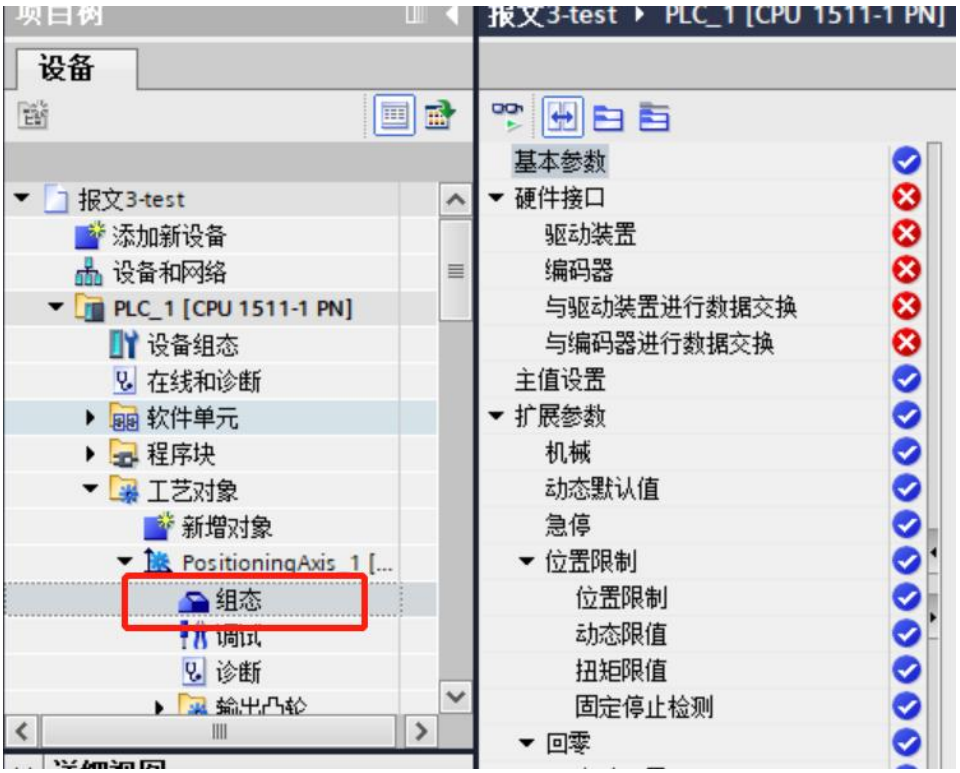
Unlike other messages, when using the message 3, it is necessary to connect the PLC to the driver in the "Topology View" **according to the actual physical connection**.

6.3.3 Process object configuration

1. Process Object - New Object - Select TO_PositioningAxis and add it

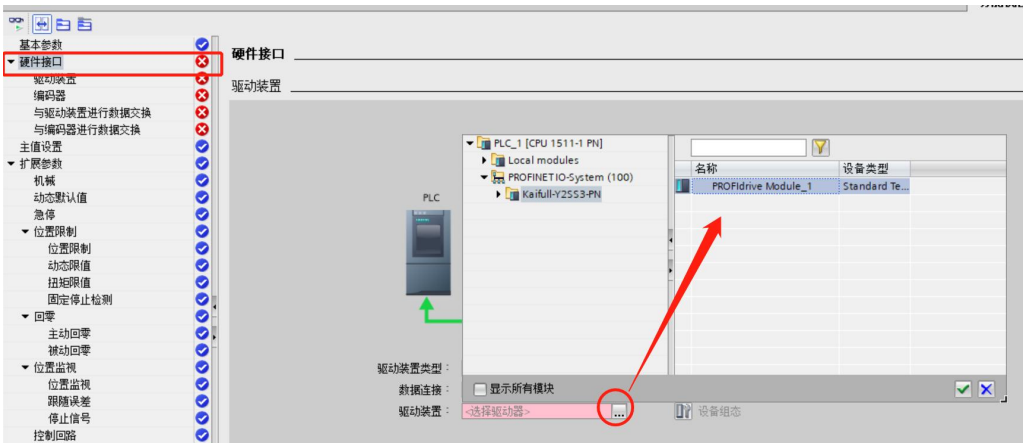


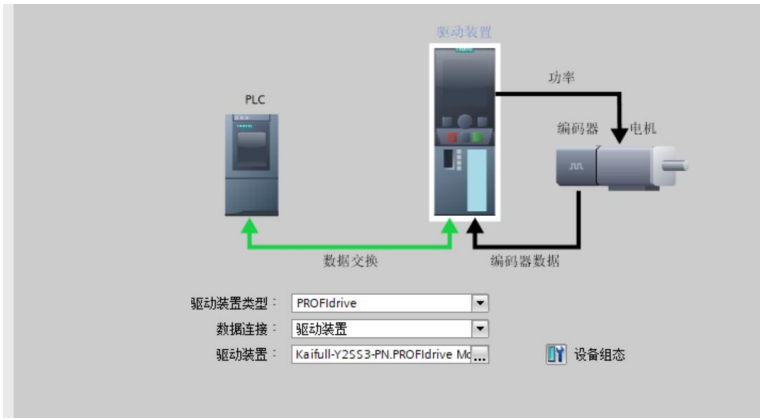
2. Double-click "Configuration"



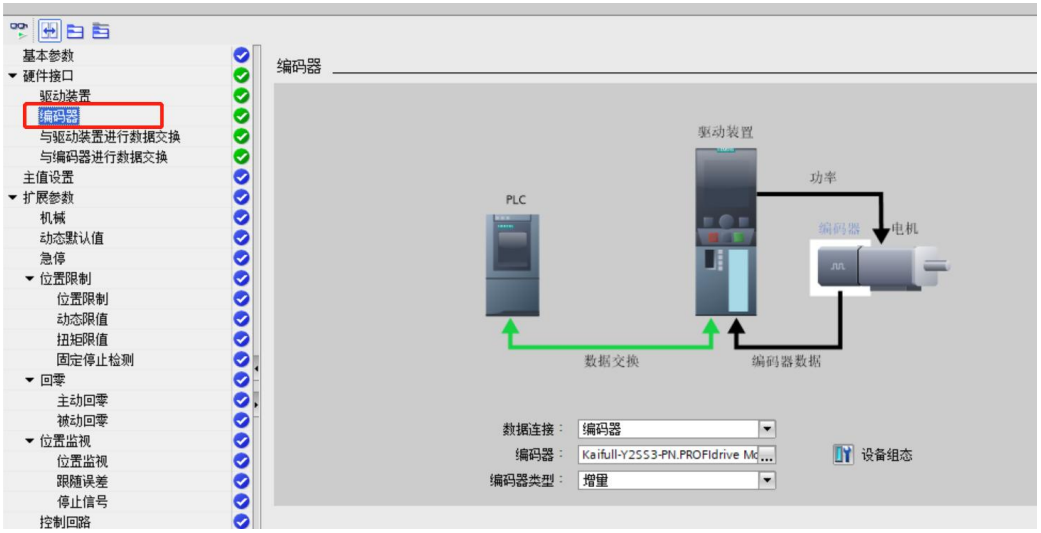
3. **Basic parameters:** Use the default parameters

4. **Hardware interface:** Select Y2SS3-PN as the driver

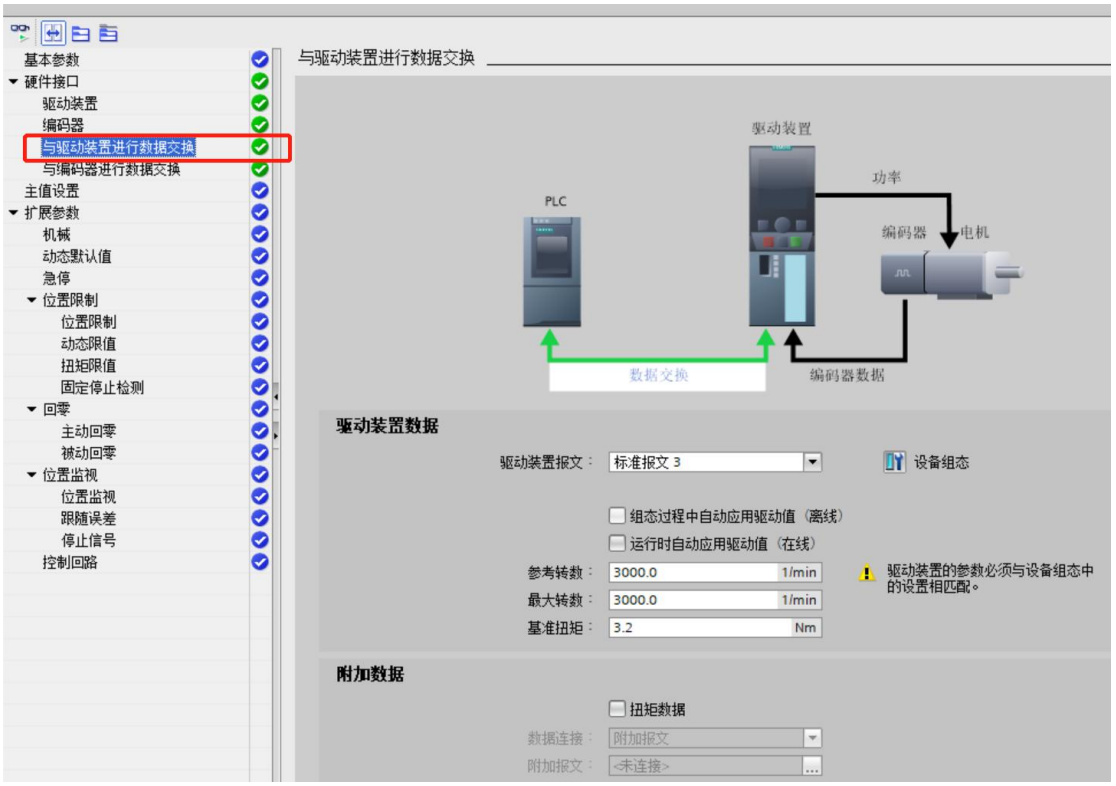




5. **Encoder:** Use the default parameters

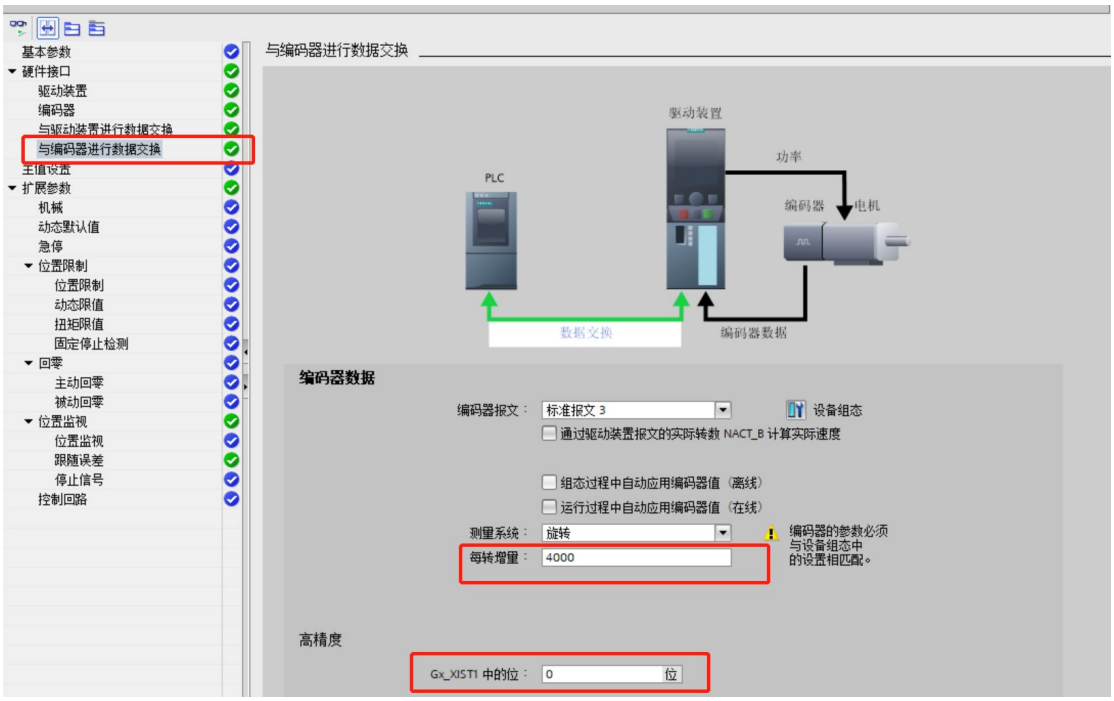


6. **Data exchange with the driver:** Use the default parameters



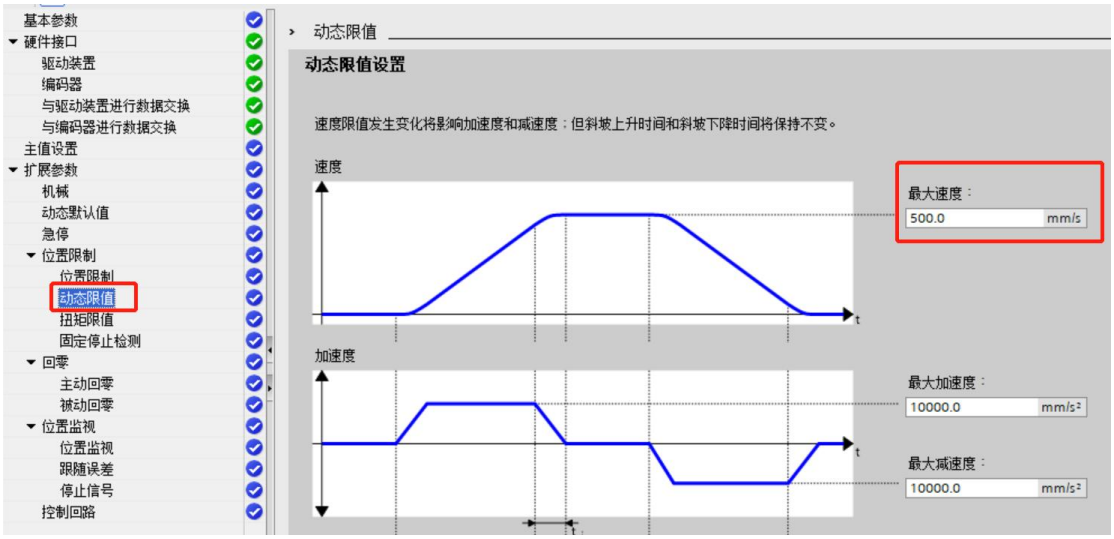
7. **Data exchange with the encoder:** This parameter means the number of pulses required for the motor to rotate one revolution. The value filled here needs to be consistent with the value of "subdivided electronic gear ratio" of the driver, and the factory default for the driver is 4000 pulses per revolution.

Fill in 0 for the "Bit in the high-accuracy-Gx-XIST1" below; otherwise, the positioning will be inaccurate.

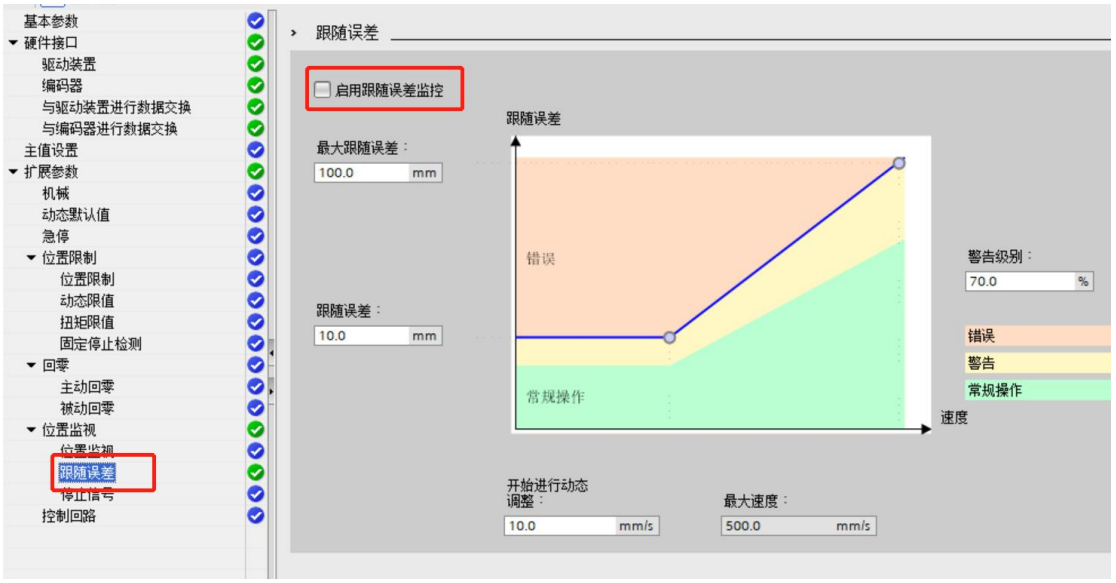


8. **Machinery:** This parameter means the movement distance of the load when the motor rotates one revolution on the actual mechanism, which can be set according to the actual situation.

9. **Dynamic limit:** Please note that the maximum speed cannot be set too low; otherwise, the speed may not increase during use



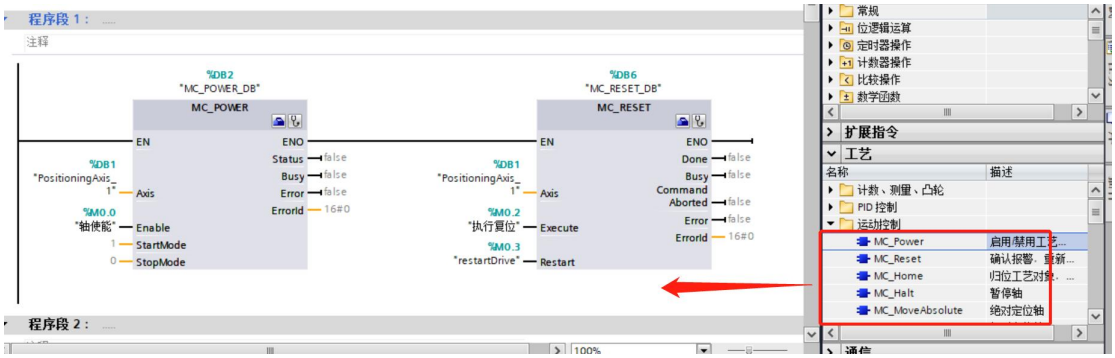
10. **Following error:** It is recommended that "Enable following error monitoring" should be unchecked; otherwise, errors may occur during running



So far, configuration of message 3 has been completed, and others can use the default parameters.

6.3.4 Edit programs

Find the required movement instruction in the instruction library



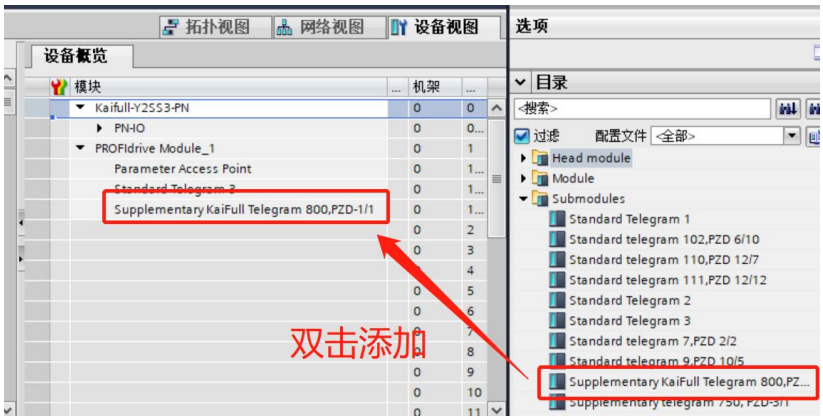
6.3.5 Instructions for message 3 homing

When using message 3, the motor needs to return to the home position first after enabled; otherwise, the motion control instruction cannot be executed.

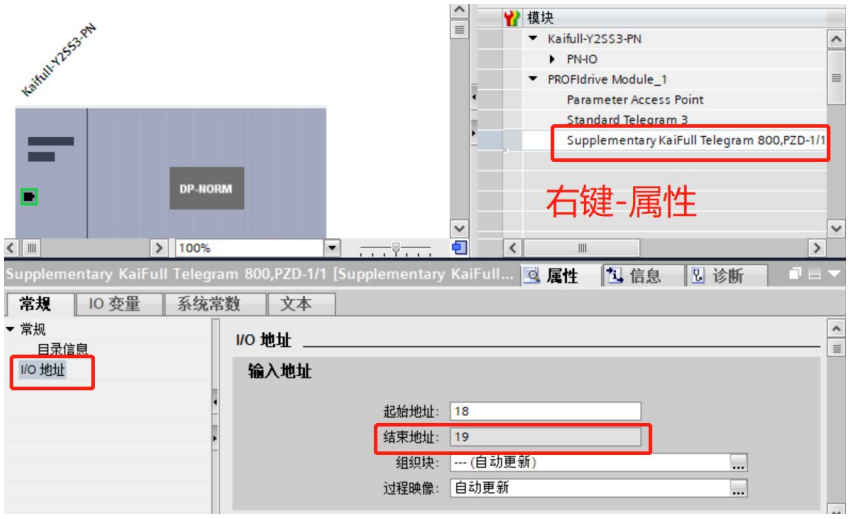
The homing action of message 3 is planned by the PLC according to the user configuration. The sensors related to homing (origin/limit) can be connected to the PLC or connected to the driver according to the above Interface Description.

- **Obtain the driver IO status**

When the sensor is connected to the driver, it is necessary to add the auxiliary message 800 to obtain the IO point status.



The input point status of the driver is obtained through the "end address" value of the message 800, and the address of each axis is different.

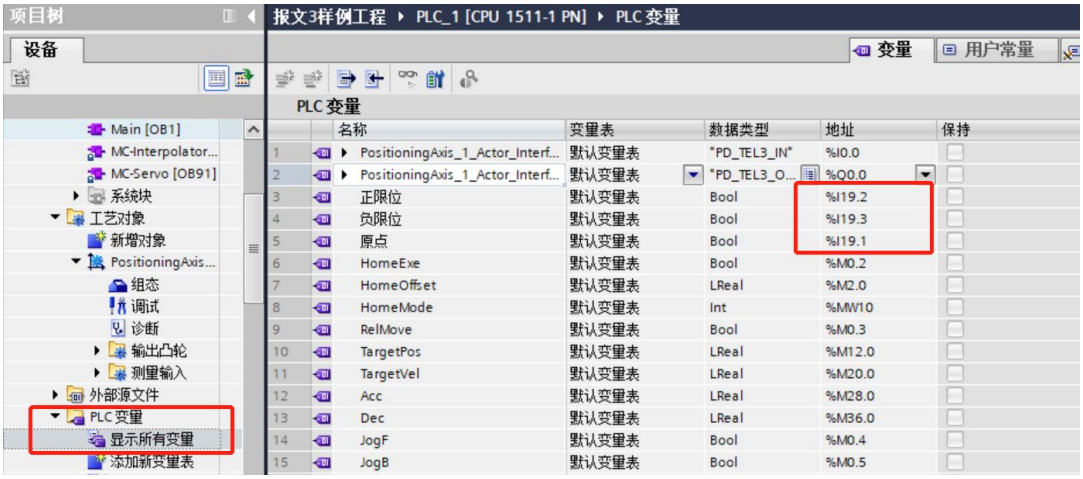


Assuming the "end address" of the message 800 is 19, Bit0-4 of %IB19 correspond to the input point status of the driver IN1-IN5 respectively

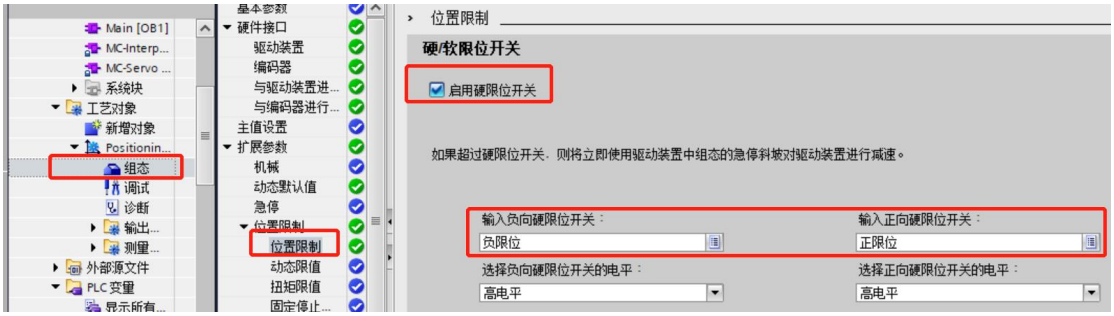
%IB19	Driver input point
Bit 0	IN1
Bit 1	IN2 (home position)
Bit 2	IN3 (positive limit)
Bit 3	IN4 (negative limit)
Bit 4	IN5

- **Homing configuration**

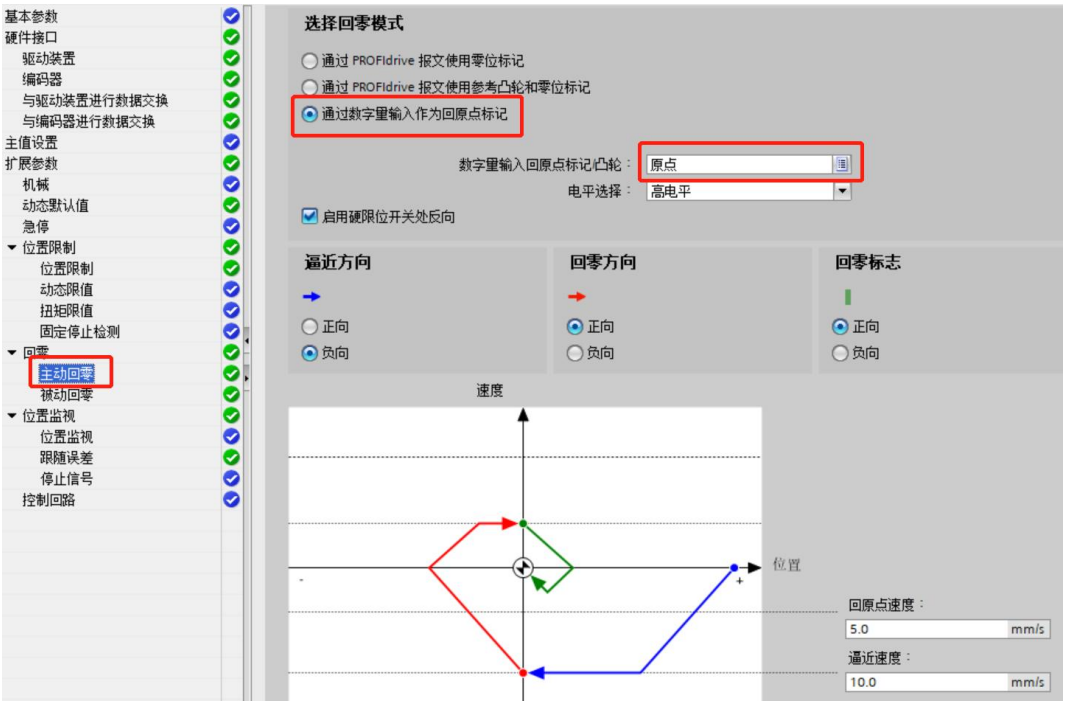
Create a new home position with positive and negative limit variables, and the address for this example is 19. If there is no limit in actual application, the limit variable is not required.



If you need to use the positive and negative limits, check "Enable hard limit" and then select the positive and negative limit variables below the switches.



Configuration of active return to zero is applicable to the homing modes 3 and 5. The parameters such as homing direction and speed are configured according to the actual application requirements



- **Introduction to homing modes**

The MC_Home function block supports multiple homing modes, which can be selected by modifying the parameter "Mode". (The common mode is 3)

Homing modes	Description
0	Absolute direct return to home position After execution, the axis does not move, and the axis position is updated to the value of the parameter "Position", which means using the current physical position as the home position.
1	Relative direct return to home position Similar to Mode 0, after execution, the axis does not move and the axis position is updated to the current position of the axis and the value of the parameter "Position". Example: the current position of the axis is 10.0mm, and the parameter "Position"=20.0mm, and after executing Mode 1 homing, the axis position is updated to 10+20=30.0mm.
2	Passive return to home position (no reset) This mode requires configuring the "Passive return to zero" in the process object first. Only execute the homing instruction, the axis does not move, and it needs to be used with other motion control instructions (e.g. relative/absolute movement) to complete returning to the home position. After completing returning to the home position, the axis position="Position" parameter. In the process of returning to home position in this mode, <TO>.StatusWord.X5 (HomingDone) will not be set to 0.
3	Active return to home position The axis returns to the home position according to the "Active return to zero" configuration in the process object. When executing returning to home position, the axis begins to move to search for the home position. After returning to the home position, the axis position=the value of the parameter "Position".
5	Active return to home position It is similar to Mode 3, except for the difference that in Mode 5, after completing returning to the home position, the axis position is the value of Process Object>Configuration>Extended Parameters>Return to Zero>"Active Return to Home Position">"Home Position" (<TO>.Homing.HomePosition)
6，7	A motor with absolute value encoder is required, and currently Y2SS3-PN does not support it
8	Passive return to home position It is basically consistent with Mode 2, except for the difference that in Mode 8, in the process of returning to home position, <TO>.StatusWord.X5 (HomingDone) will be set to 0 first, and then set to 1.

9	Suspend passive return to home position The active operation of passive return to the home position will be suspended.
10	Passive return to home position (the "Position" parameter has no effect) It is similar to Modes 2 and 8, except for the difference that after detecting the mark of returning to home position, the actual value will be set to the "Home Position" (<TO>.Homing.HomePosition) set under "Process Object>Configuration>Extended Parameters>Return to Zero>Passive Return to Zero", while in Modes 2 and 8, the home position is the "Position" parameter.

6.3.6 Example demonstration

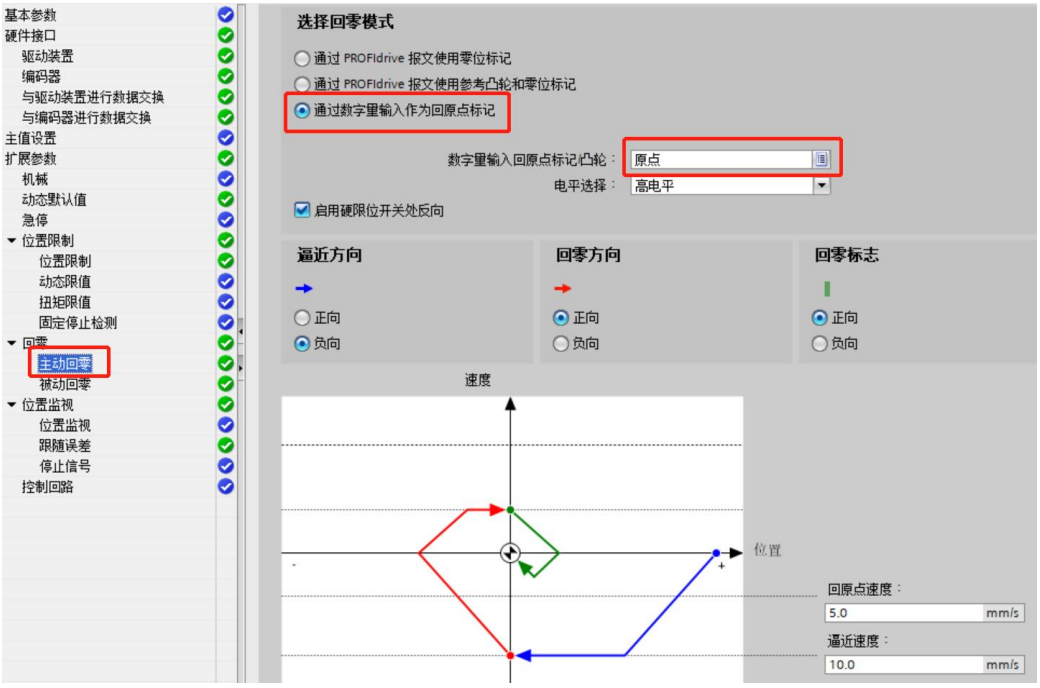
1) Example of homing mode 3

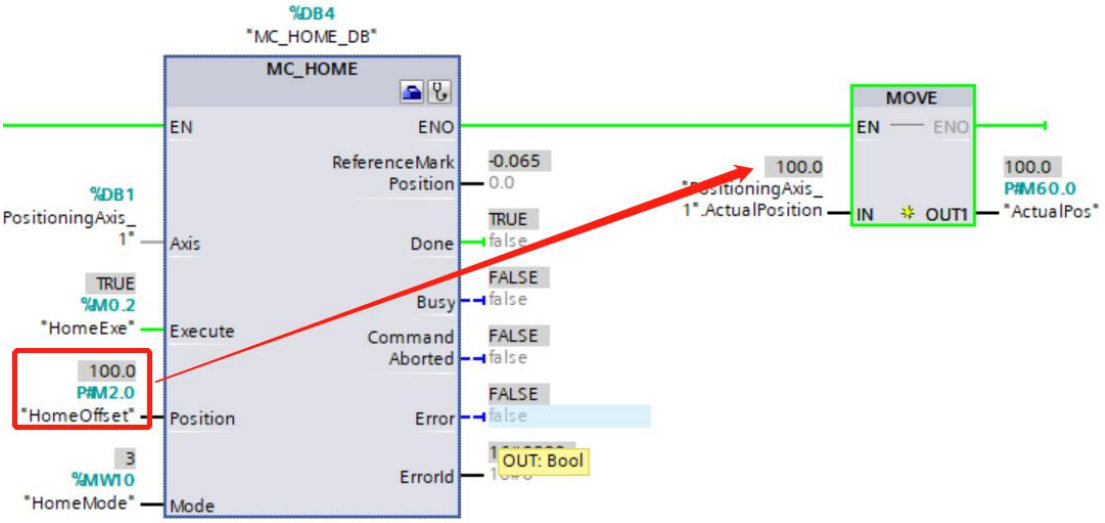
① Configure the "Active return to zero" parameter in the process object configuration

② Enable MC_POWER instruction

③ Fill in 3 for the MC_Home parameter "Mode" and execute it

④ After execution, the motor starts to move to look for the home position, and after that, the axis position is set to the value of the parameter "Position".





2) Axis jog

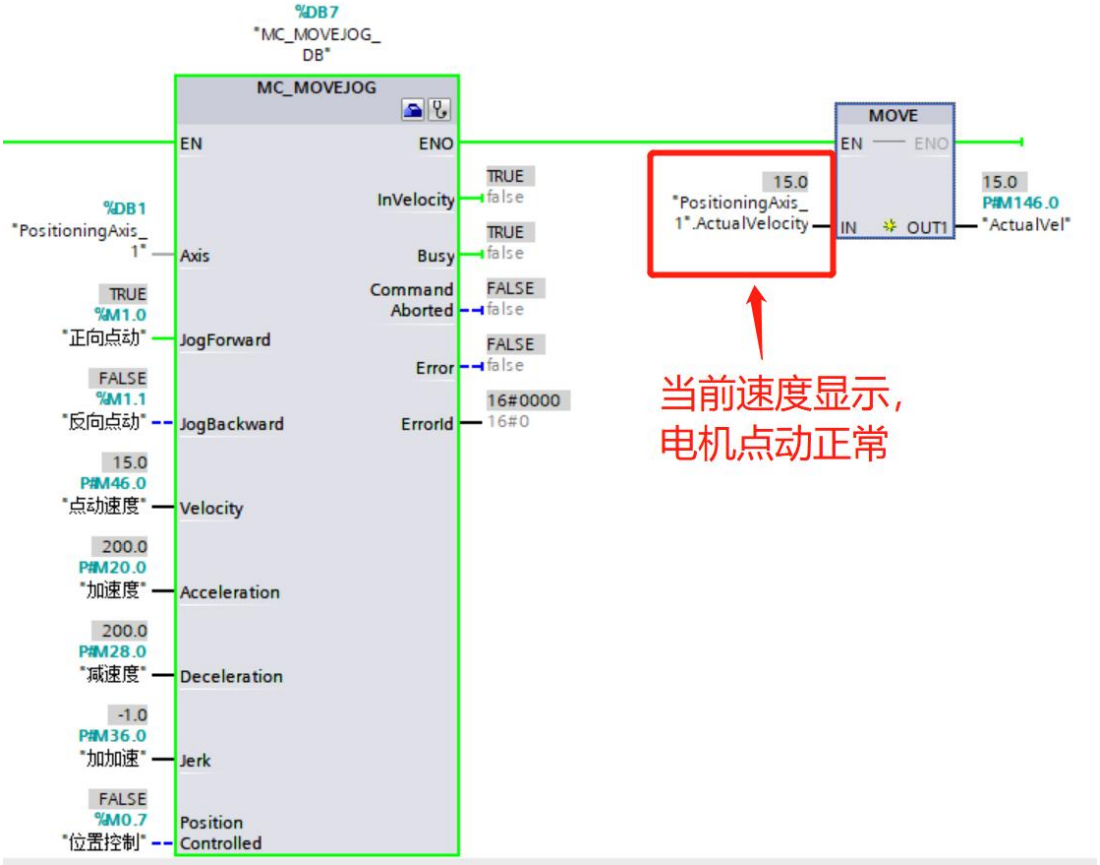
① Insert the MC_POWER, MC_HOME and MC_MOVEJOG instructions

② Execute the MC_POWER and MC_HOME separately. (Before controlling the motor movement, it is necessary to perform returning to the home position once; otherwise, the motor executing other motion control instructions will not move)

③ Fill in the parameters such as speed, acceleration, and deceleration

④ At this point, set JogForward=True and the motor will rotate clockwise; set JogBackward=True and the motor will rotate clockwise; when both JogForward and JogBackward are false, the motor will stop.

⑤ Support speed modification during motor movement



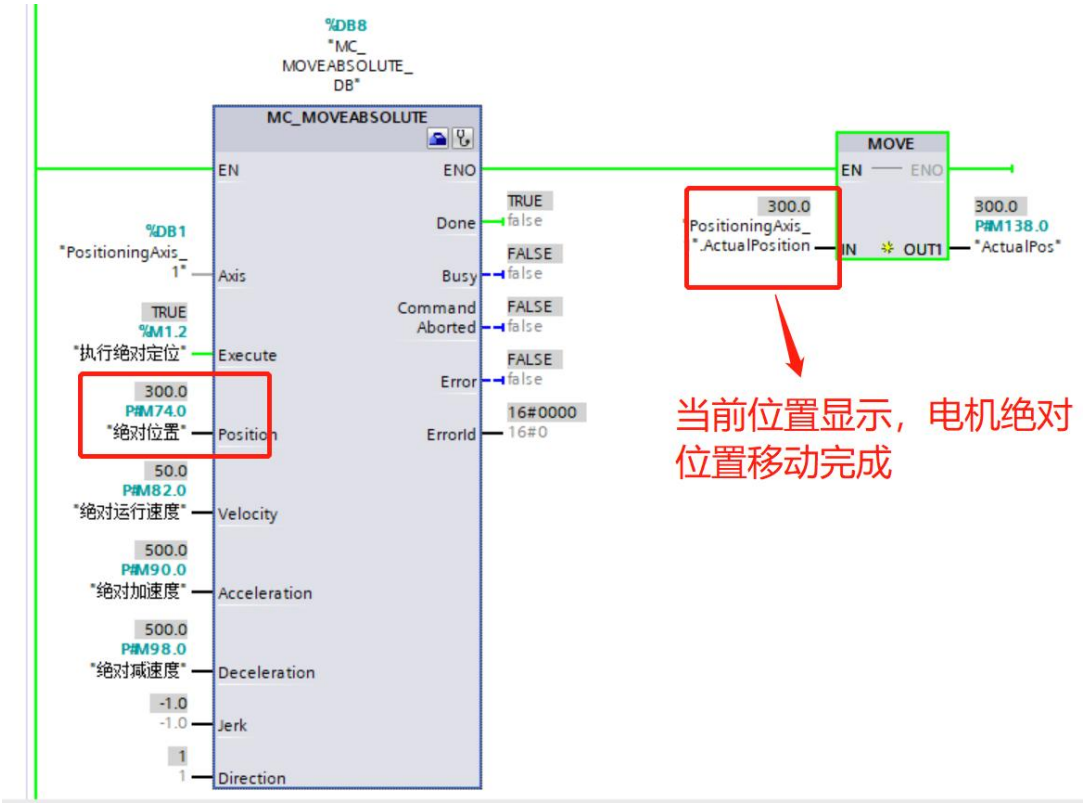
3) Absolute position control

① Insert the MC_POWER, MC_HOME and MC_MOVEABSOLUTE instructions

② Execute the MC_POWER and MC_HOME separately. (Before controlling the motor movement, it is necessary to perform returning to the home position once; otherwise, the motor executing other motion control instructions will not move)

③ Fill in the parameters such as target position, speed, acceleration, and deceleration respectively

④ Set Execute=True and the motor will begin to move.

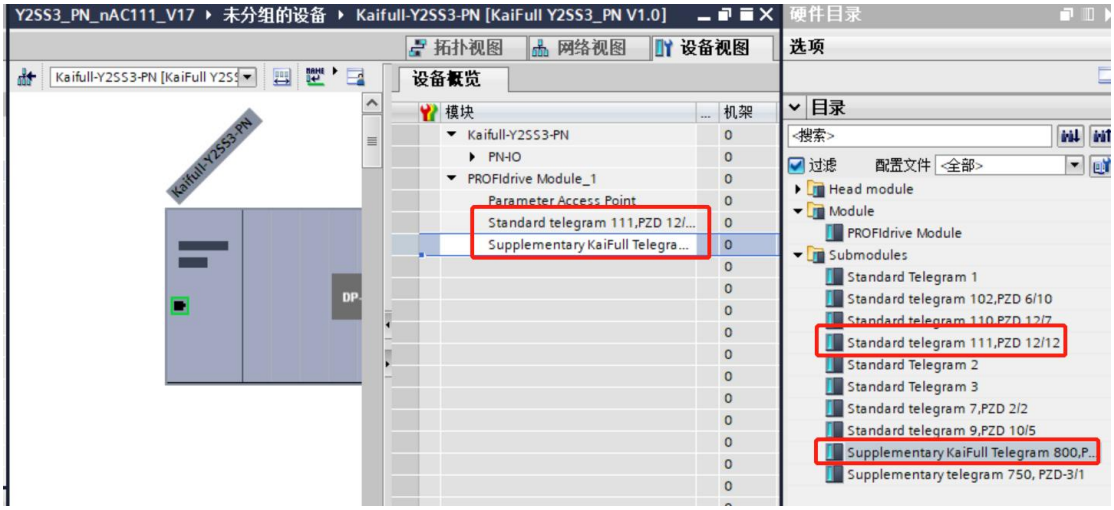


6.4 Message 111

Message 111 uses Profinet RT communication, and the motor speed and position control can be achieved using the function block SinaPos (FB284) in the driver library provided by Botu software.

6.4.1 Add message 111 and auxiliary message 800

Delete the original message in the device view, select message 111 and auxiliary message 800 in the lower right corner, and double-click to add it

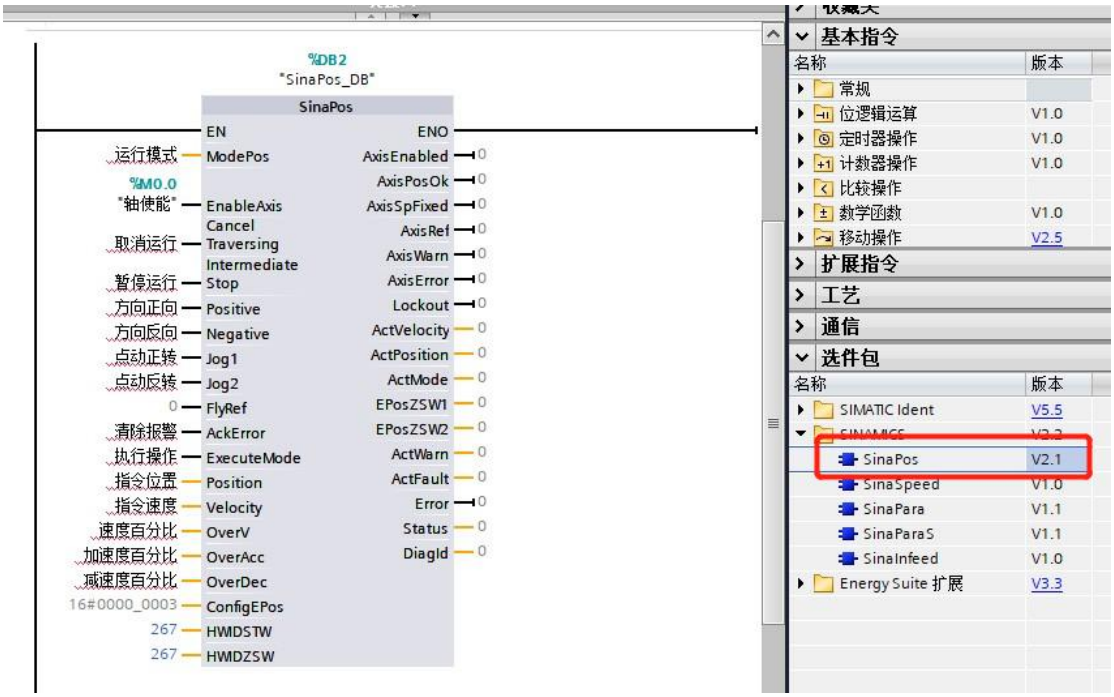


6.4.2 View hardware identifiers

The hardware identifiers are used in the following programming

6.4.3 Edit programs

Find and add the SinaPos instruction (FB284) on the programming page



- Description of SinaPos instruction

Input signal	Type	Default value	Meaning
ModePos	INT	0	Operation mode: 1=Relative operation 2=Absolute operation 3=Perform positioning based on the location 4=Process of returning to reference point 5=Set the position of returning to reference point 6=Run the program segments 0-15/63 (G120/S120) 7=Jog 8=Jog increment
EnableAxis	BOOL	0	Switch instruction: 0=OFF, 1=ON
CancelTraversing	BOOL	1	0=Reject the running jobs in activated state, 1=Not reject
IntermediateStop	BOOL	1	0=Interrupt the running instruction in activated state, 1=No intermediate stop
Positive	BOOL	0	Positive direction
Negative	BOOL	0	Negative direction
Jog1	BOOL	0	Jog signal source 1
Jog2	BOOL	0	Jog signal source 2
FlyRef	BOOL	0	0=Cancel active return to reference point, 1=Select active return to reference point
AckError	BOOL	0	Fault response
ExecuteMode	BOOL	0	Activate running job/receive set value/activate the function of returning to reference point
Position	DINT	0[LU]	Applicable to the position set values (in LU) of "direct set value designation/MDI" of the operating mode or applicable to the running program segment number of "running program segment" of the operating mode
Velocity	DINT	0[LU/min]	Speed (in LU/min) applicable to MDI operating mode
OverV	INT	100[%]	Effective speed multiplier for all operating modes: 0-199%
OverAcc	INT	100[%]	Effective acceleration multiplier: 0-100%
OverDec	INT	100[%]	Effective deceleration rate 0-100%
ConfigEPos	DWORD	3h	For more detailed information, please refer to Relative Positioning
HWIDSTW	HW_IO	0	Symbol name or hardware ID on SIMATIC S7-1200/1500 in set value slot
HWIDZSW	HW_IO	0	Symbol name or hardware ID on SIMATIC S7-1200/1500 in actual value slot

FB284 modes supported by Y2SS3-PN:

FB284 mode	Description	Supported or not
1	Relative operation	√
2	Absolute operation	√
3	Perform positioning based on the location	√
4	Process of returning to reference point	√
5	Set the position of returning to reference point	√
6	Running program segment	×
7	Jog	√
8	Jog increment	√

6.4.4 IO setting

The default IO definition for driver CN3 is as follows:

Input port	Definition	Polarity
IN2	Home position	Normally on
IN3	Positive limit	Normally on
IN4	Negative limit	Normally on

1) Enable limit

The default limit of the driver does not take effect. If the user needs to use the limit function, he needs to set "Limit control"="enable" through the software

2) Polarity setting

The default polarity of the driver IO is normally on. If using a normally closed sensor, the corresponding IO needs to be set to "Normally closed"

6.4.5 Example demonstration

1) Relative/absolute position control

① ModePos=1 represents relative movement, while ModePos=2 represents absolute movement

② EnableAxis=True, enable on motor

③ CancelTraversing and IntermediateStop represent canceling and suspending the running, respectively. When these two parameters are false, the motor will stop. When the motor is running, these two parameters must be True

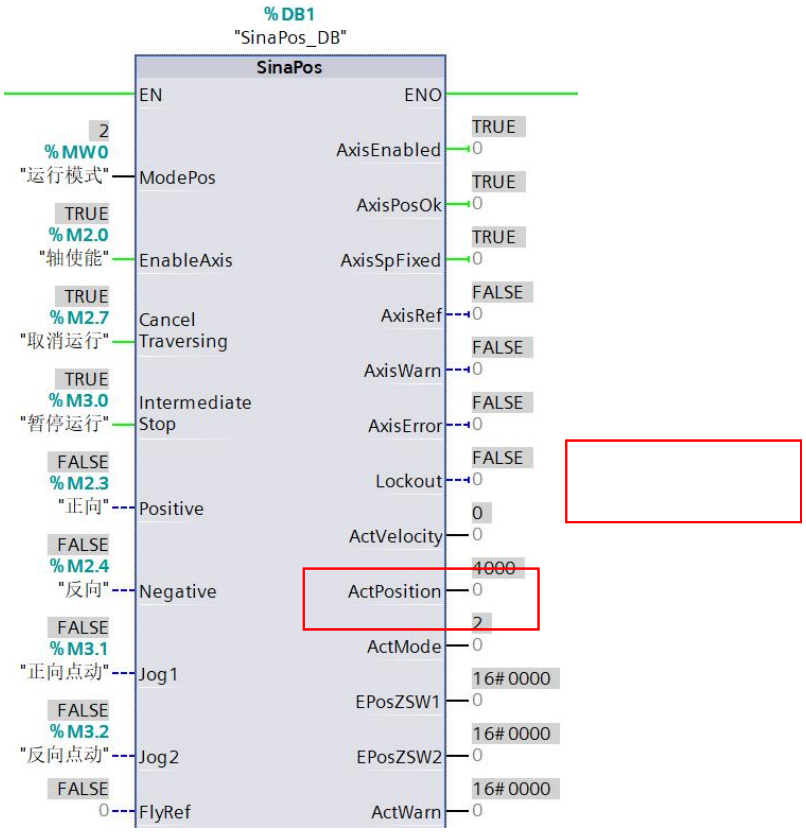
④ Position: Fill in the target position, in pulse

⑤ Velocity: Fill in the axis velocity value, in 1000 pulses/minute. Example: Fill in 240, and it means the velocity value is 240000 pulses/minute=4000 pulses/second

⑥ OverV, OverAcc, and OverDec represent the percentage of velocity, percentage of acceleration, and percentage of deceleration, respectively. The default value 100% is used here

⑦ HWIDSTW and HWIDZSW: Fill in the above hardware identifier

⑧ Finally, set ExecuteMode=Tyre, and the motor will start to rotate



After writing ExecuteMode=True, the motor will rotate. When stopping, the actual position ActPosition is equal to the instruction position Position

2) Axis jog control

There are two methods for implementing jog control in message 111, namely Mode 3 and Mode 7

- **Method 1:**

① ModePos=3

② EnableAxis=True, enable on motor

③ CancelTraversing and IntermediateStop=True

④ OverV, OverAcc, and OverDec represent the percentage of velocity, percentage of acceleration, and percentage of deceleration, respectively. The default value 100% is used here

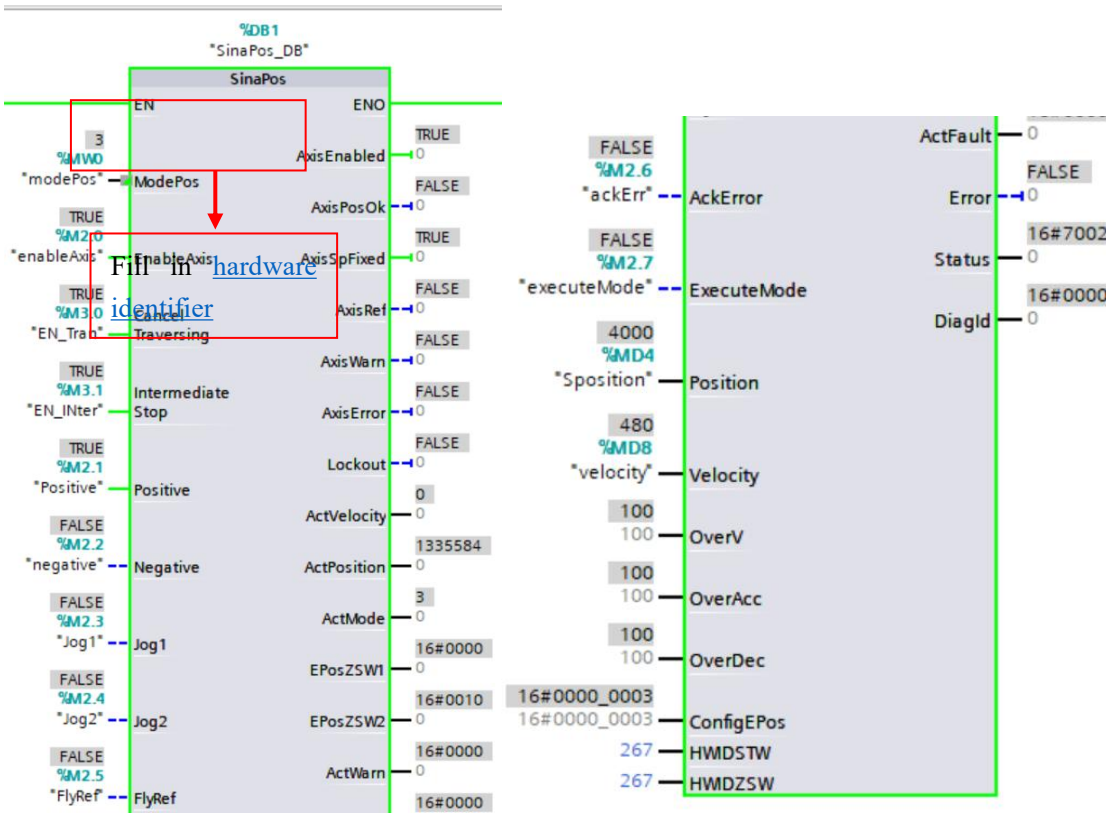
⑤ HWIDSTW and HWIDZSW: Fill in the above hardware identifier

⑥ Velocity: Fill in the axis velocity value, in 1000 pulses/minute. Example: Fill in 240, and it means the velocity value is 240*1000=240000 pulses/minute=4000 pulses/second

⑦ Write Execute=True, the motor will not move, and this step is equivalent to writing the velocity value

⑧ Finally, write Positive=True, and the motor will rotate in the forward direction; Negative=True, the motor will jog in the reverse direction. When both Positive and Negative are true or false, the motor will stop.

⑨ If you need to change the velocity, repeat steps ⑥ and ⑦



Fill in hardware identifier

- **Method 2:**

① **ModePos=7**

② EnableAxis=True, enable on motor

③ CancelTraversing and IntermediateStop can be true or false

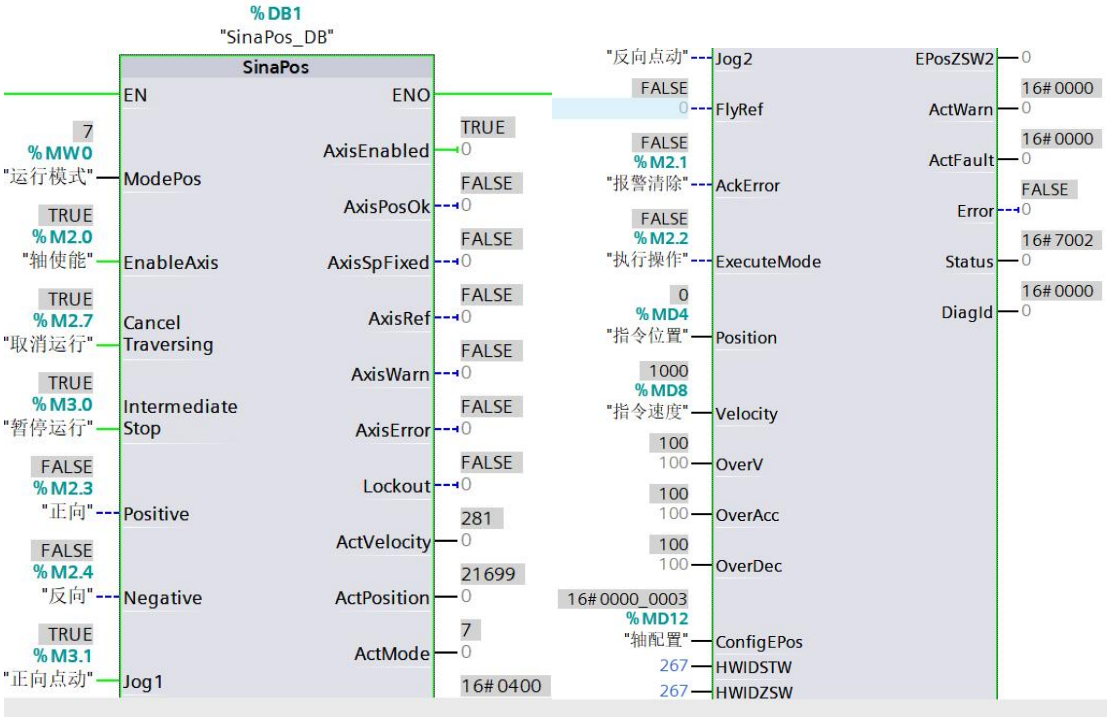
④ OverV, OverAcc, and OverDec represent the percentage of velocity, percentage of acceleration, and percentage of deceleration, respectively. The default value 100% is used here

⑤ HWIDSTW and HWIDZSW: Fill in the above hardware identifier

⑥ The jog speed in Mode 7 is determined by internal parameters of the driver and can be modified through the non-periodic data read and write function block SinaPara or SinaParaS

Parameter address	Data type	Default value	Unit	Description
847	16 bits	200	0.01 rps	Reverse jog speed of Message 111 FB284 Mode 7, Example: 847=200, which means the motor speed is 2 rps
848	16 bits	1000	0.01 rps	Forward jog speed of Message 111 FB284 Mode 7, Example: 848=200, which means the motor speed is 2 rps

⑦ Finally, write Jog1=True, and the motor will rotate in the forward direction; when Jog2=True, the motor will rotate in the reverse direction; when both Jog1 and Jog2 are True or False, the motor will stop.



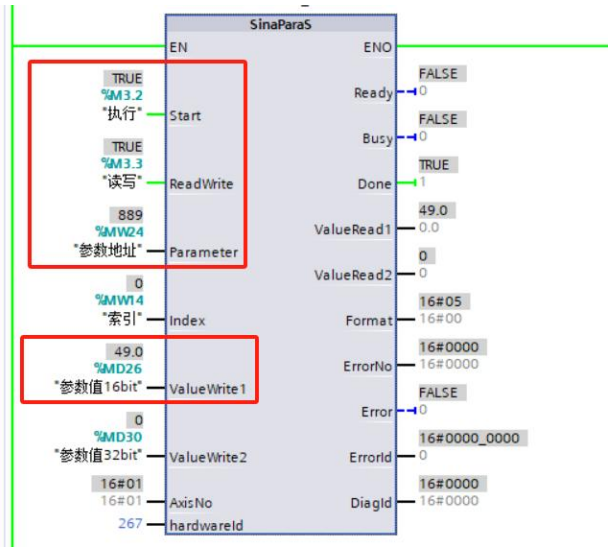
3) Axis homing control

The homing process in Message 111 is an internal planning of the driver, and the user needs to connect the sensor or encoder Z-phase signal to the driver IO interface according to the above Interface Description.

Y2SS3-PN provides multiple homing methods, and the appropriate homing method can be selected by modifying the internal parameter PNU889 of the driver through the function block SinaParaS.

Homing example:

① Modify the parameter PNU889 and select the homing method (default value PNU889=17)



② ModePos=4

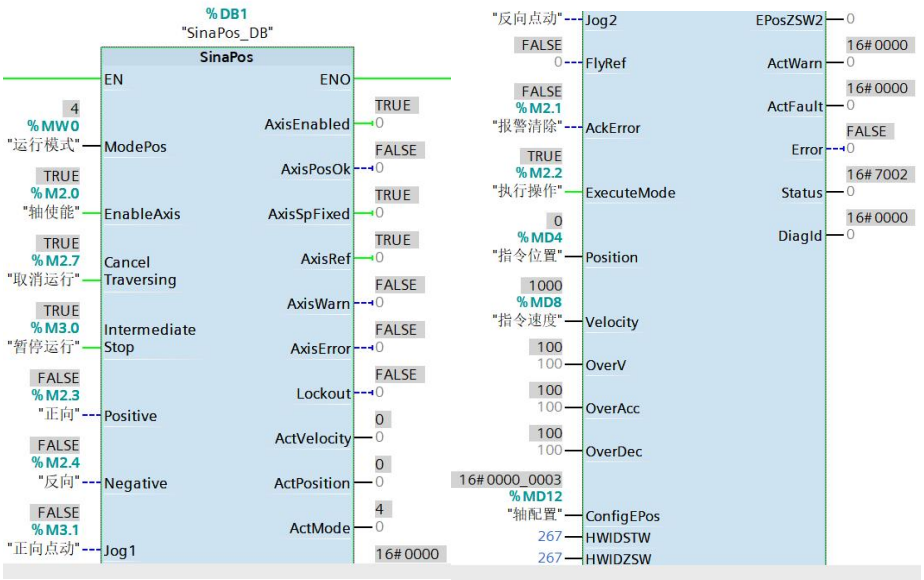
③ EnableAxis=True, enable on motor

④ CancelTraversing and IntermediateStop=Ture

⑤ HWIDSTW and HWIDZSW: Fill in the above hardware identifier

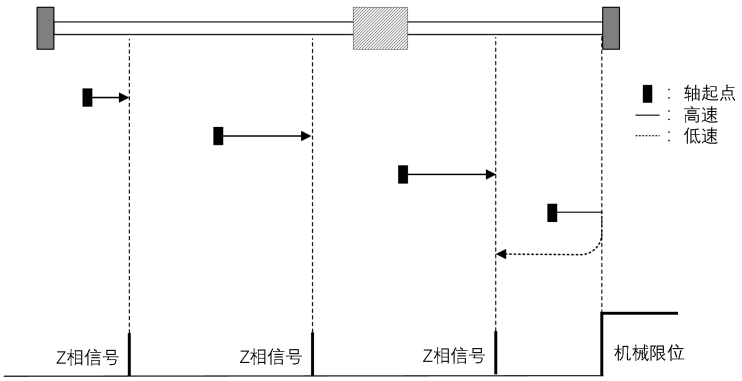
⑥ After writing ExecuteMode=True, the motor will start to return to the home position

⑦ After homing successfully, AxisRef=True, ActPosition=0



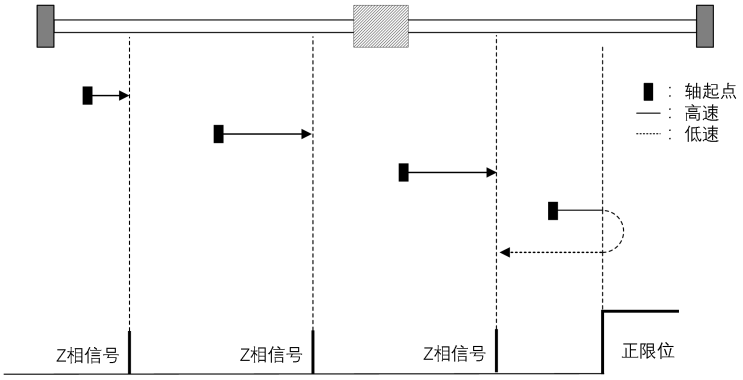
- **Homing method diagram**

① PNU889=0, home position: rising edge of encoder Z-phase signal



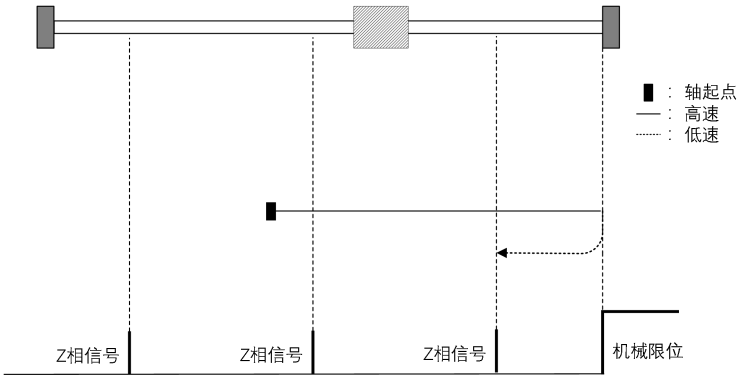
回原方式0

② PNU889=1, home position: rising edge of encoder Z-phase signal



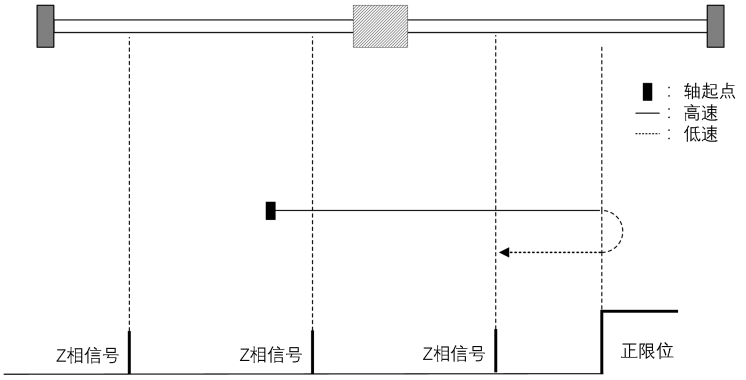
回原方式1

③ PNU889=2, home position: rising edge of Z-phase signal



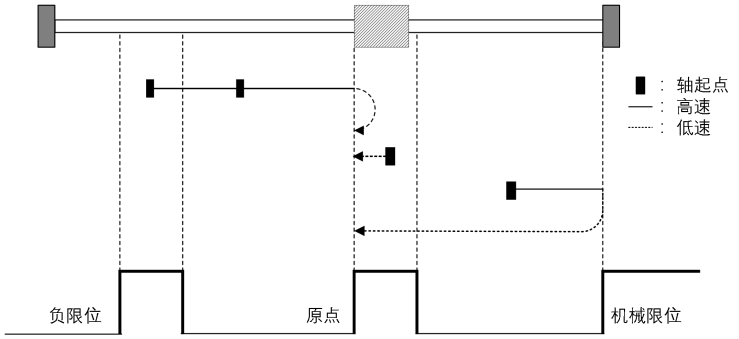
回原方式2

④ PNU889=3, home position: rising edge of Z-phase signal



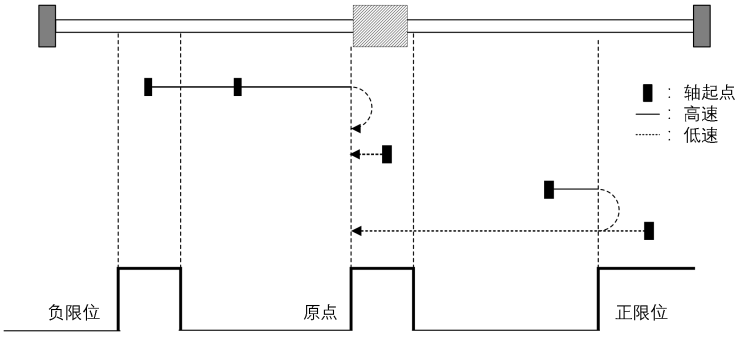
回原方式3

⑤ PNU889=16, home position: falling edge of home sensor



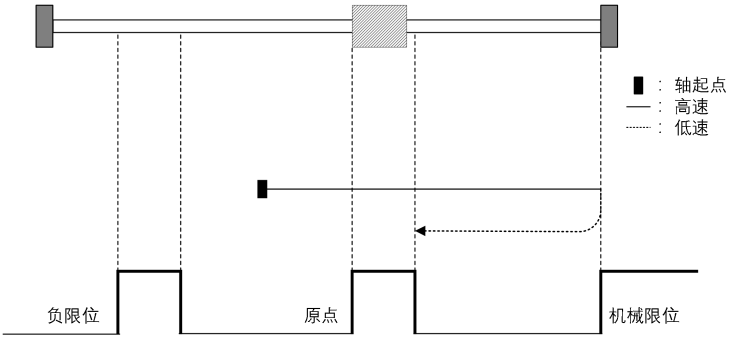
回原方式16

⑥ PNU889=17, home position: falling edge of home sensor



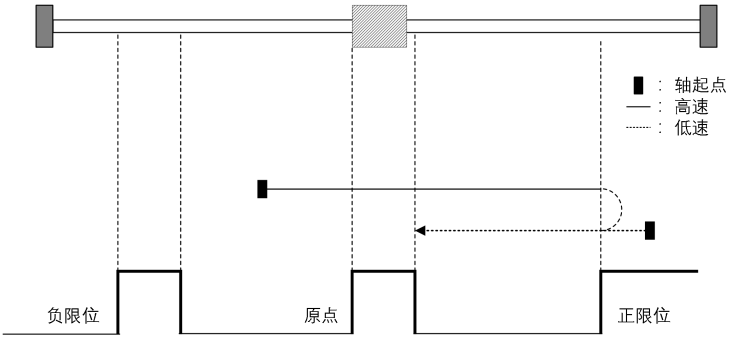
回原方式17

⑦ PNU889=18, home position: rising edge of home sensor



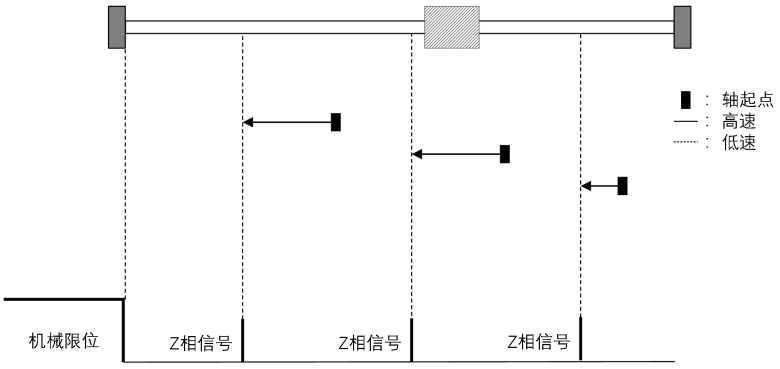
回原方式18

⑧ PNU889=19, home position: rising edge of home sensor



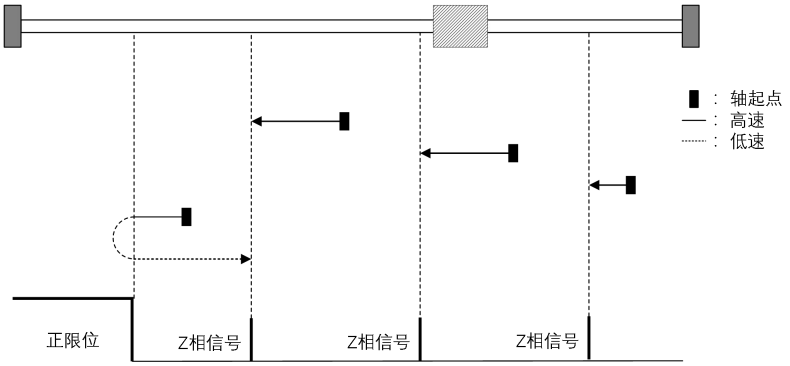
回原方式19

⑨ PNU889=32, home position: rising edge of encoder Z-phase signal



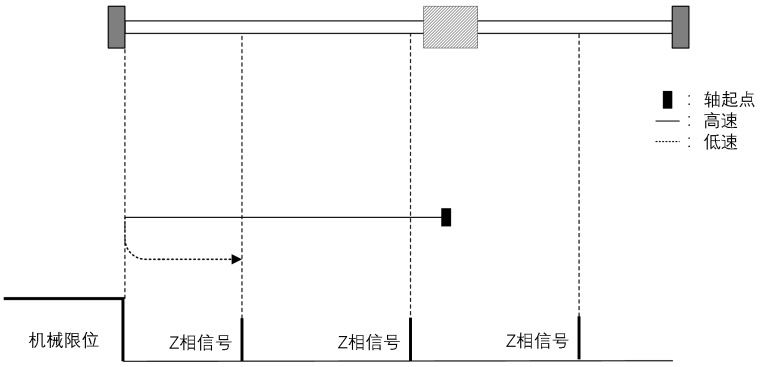
回原方式32

⑩ PNU889=33, home position: rising edge of encoder Z-phase signal



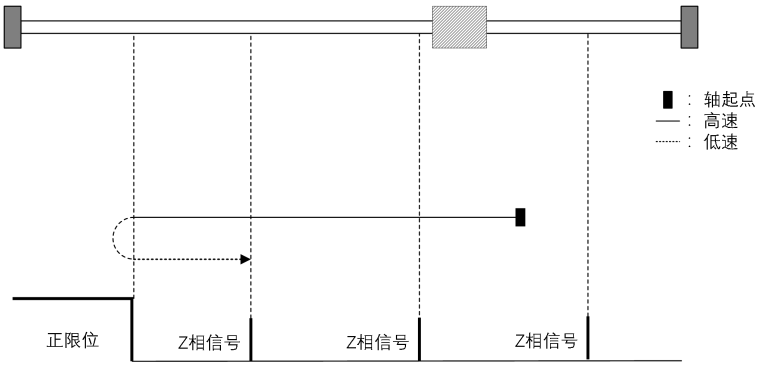
回原方式33

⑪ PNU889=34, home position: rising edge of encoder Z-phase signal



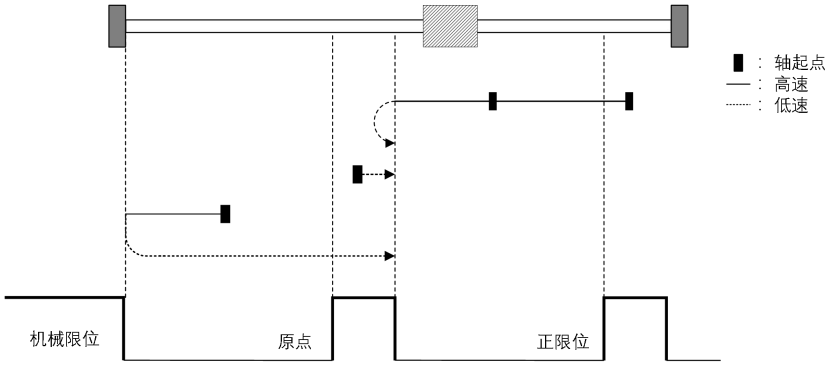
回原方式34

⑫ PNU889=35, home position: rising edge of encoder Z-phase signal



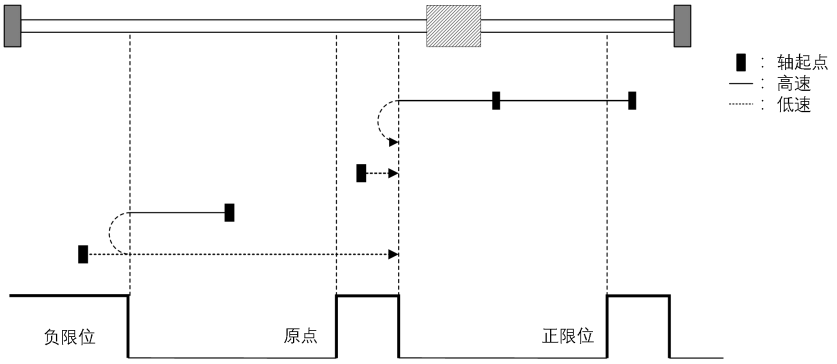
回原方式35

⑬ PNU889=48, home position: falling edge of home sensor



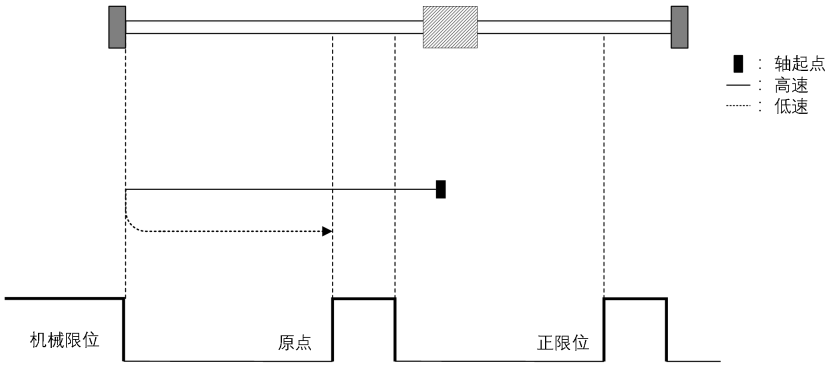
回原方式48

⑭ PNU889=49, home position: falling edge of home sensor



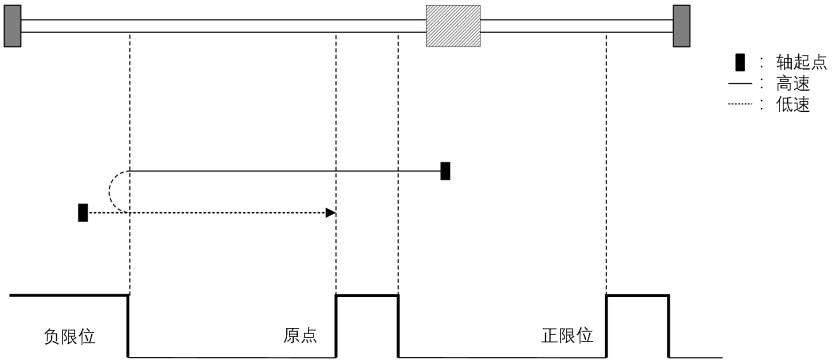
回原方式49

⑮ PNU889=50, home position: rising edge of home sensor



回原方式50

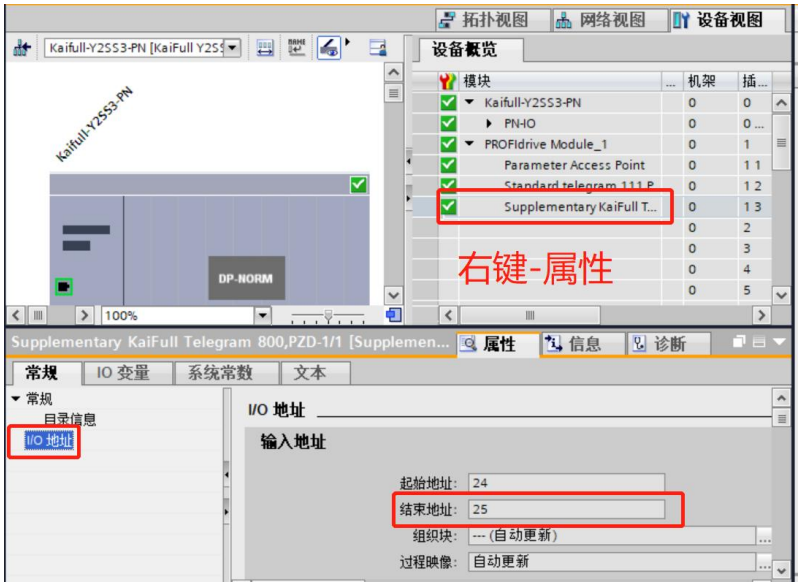
⑯ PNU889=51, home position: rising edge of home sensor



回原方式51

6.4.6 Obtain the driver IO status

1. View the IO "end address" of the driver message 800



2. Assuming the "end address" is 25, bit0-bit4 of %IB25 correspond to the status of the driver IN1-IN5 respectively.

%IB25	Driver input point
Bit 0	IN1
Bit 1	IN2 (home position)
Bit 2	IN3 (positive limit)
Bit 3	IN4 (negative limit)
Bit 4	IN5

Example: %I25.2=True means that there is an input signal for the driver IN3 (positive limit)

7 Non-periodic data read and write

The non-periodic data function block SinaPara and SinaParaS provided by Siemens is a function block that can interact with the driver for data interaction, and users can read and write internal parameters of the driver through this function block.

The readable and writable parameter addresses are defined by the driver, including the driver output current, subdivision, and message 111 homing method.

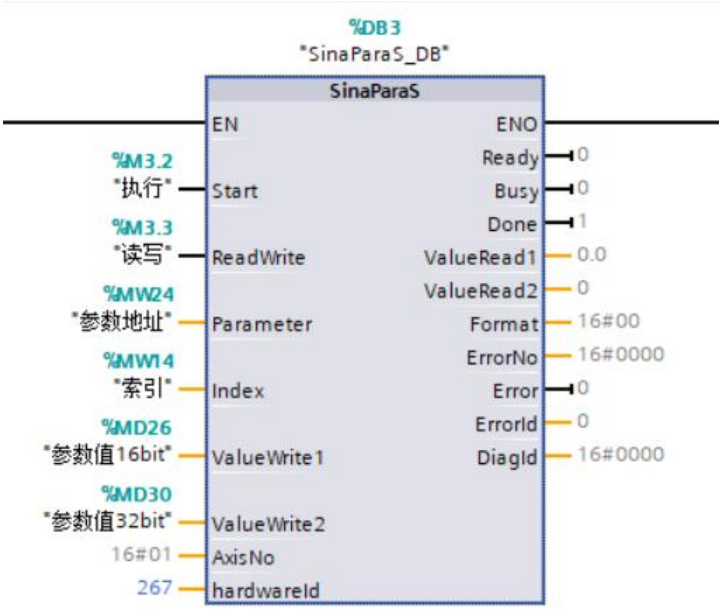
7.1 Driver parameter address

Address	Definition	Data type	Default value	Unit	Description
803	Encoder resolution	UINT32	4000	Pulse/turn	Example: The resolution of the encoder using a 1000 line is 4000
804	Driver subdivision number	UINT32	4000	Pulse/turn	Number of pulses per revolution of the motor
813	Open loop/closed-loop selection	UINT8	0	-	0=Closed-loop, 1=Open-loop
816	Default parameters setting of motor	UINT8	0	-	0=Enable, 1=Disable
847	Reverse jog speed	UINT16	200	0.01 rps	Reverse jog speed of Message 111 FB284 Mode 7, e.g. 200=2 rps
848	Forward jog speed	UINT16	1000	0.01 rps	Forward jog speed of Message 111 FB284 Mode 7, e.g. 200=2 rps
849	Motor acceleration	UINT32	2000	0.01 rps²	Motor acceleration
850	Motor deceleration	UINT32	2000	0.01 rps²	Motor deceleration
889	Homing setting	UINT8	17	-	Homing method selection in Message 111 FB284 Mode 4
891	Motor homing low speed	UINT16	50	0.01 rps	Motor homing low speed in Message 111 FB284 Mode 4
892	Motor homing high speed	UINT16	100	0.01 rps	Motor homing high speed in Message 111 FB284 Mode 4
893	Acceleration of returning to home position	UINT32	3000	0.01 rps²	Acceleration of returning to home position in Message 111 FB284 Mode 4
894	Emergency stop acceleration and deceleration	UINT32	10000	0.01 rps²	Emergency stop acceleration and deceleration

895	Limit switch	UINT8	0	-	0: Disable, 1: Enable
896	Homing offset	INT32	0	Pulse	Distance moved after homing in place; the current position will be cleared to zero after being in place
1010	Power-off saving of parameters	UINT32	0	-	Write 1010=1702257011, and all parameters will be saved after power off;

7.2 Description of SinaParaS instruction

SinaParaS is a function block that can read and write a single driver parameter at a time.



- **Description of SinaParaS pin**

Pin	Data type	Description
Start	Bool	False=No operation; True=Perform read/write operations
ReadWrite	Bool	False=Read; True=Write
Parameter	Int	Parameter address
Index	Int	Parameter index
ValueWrite1	Real	16-bit parameter value
ValueWrite1	DInt	32-bit parameter value
AxisNo	Int	Axis number; fill in 1 regardless of how many axes there are
hardwareId	HW IO	Hardware ID, which can be viewed by right-clicking the Message - Properties
Ready	BOOL	Feedback signal connected in the LAcycCom environment; 1=End of job or cancellation of job (one cycle)
Busy	BOOL	Ongoing job (if "Busy"=1)
Done	BOOL	If the job is ended correctly, it indicates that the edge changes from 0 to 1
ValueRead1	REAL	Read the value of the parameter (REAL format, 8-16 bits)
ValueRead2	DINT	Read the value of the parameter (DINT format, 32bit)
Format	INT	Format of the read parameter
ErrorNo	INT	Error number that complies with the PROFIdrive protocol
Error	BOOL	Activate the group fault ->"Error"=1
Status	DWORD	The first word -> indicates in binary encoding form which parameter access fault occurred The second word: Fault type
DiagId	WORD	Extended communication error ->SFB call error

7.3 Use Example of SinaParaS

The following demonstrates changing the homing method through the SinaParaS function block, and it can be obtained that the address of relevant parameters is 889 by querying the above table

1. Read the current homing method

① Write ReadWrite=False (False=Read, True=Write)

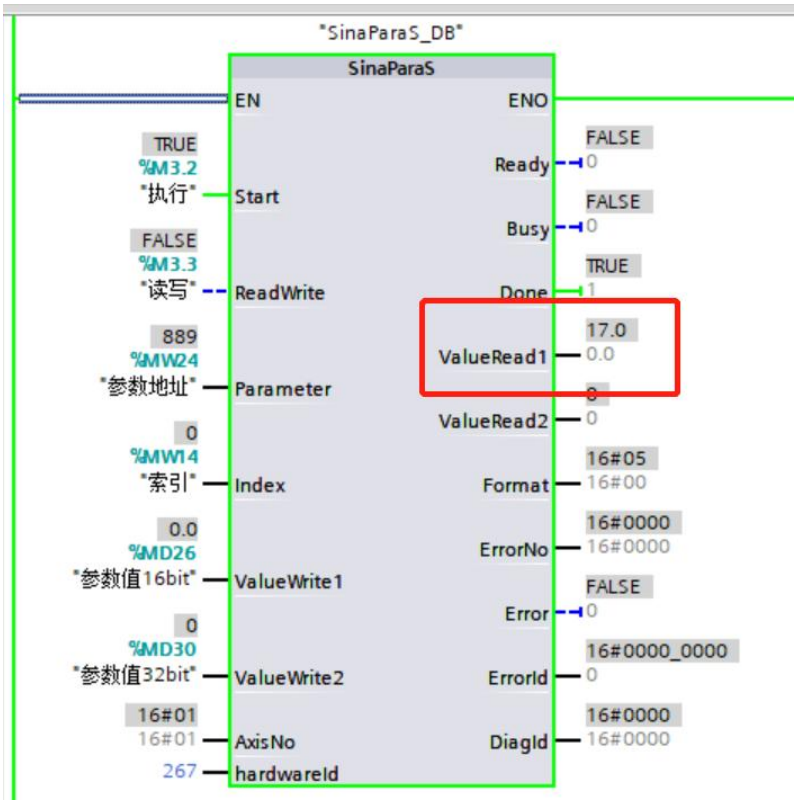
② Parameter address Parameter=889

③ AxisNo=1, fill in 1 regardless of how many axes are connected

④ hardwareId=hardware identifier

⑤ Start=True, perform read operation.

⑥ After reading successfully, the default value of PNU889 is 17. Because this parameter is an 8-bit data, only view the value of Value Read1

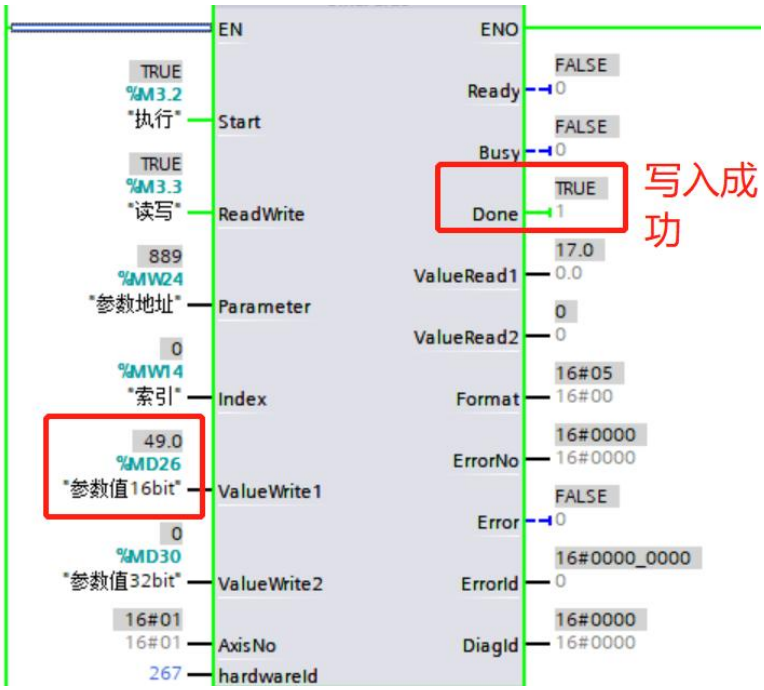


2. Modify the homing method PNU889=49

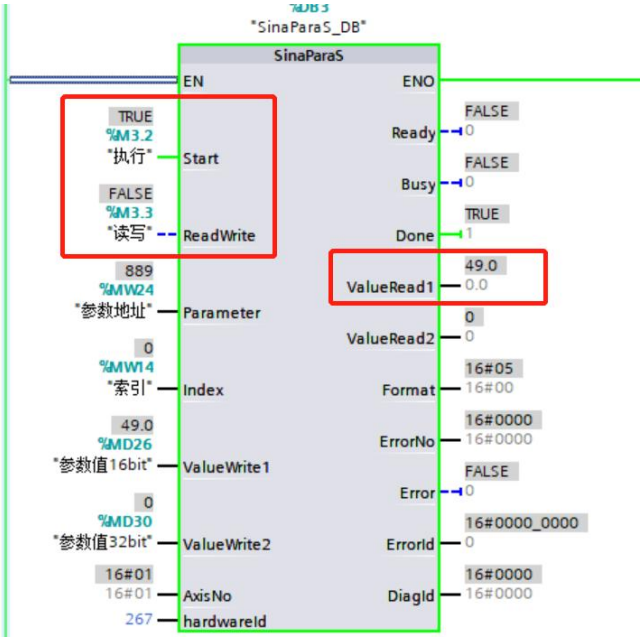
① Input ValueWrite1=49.0 (for 8-16 bit parameters, write ValueWrite1; for 32-bit parameters, write ValueWrite2)

② ReadWrite=True

③ Start=True, write the parameter

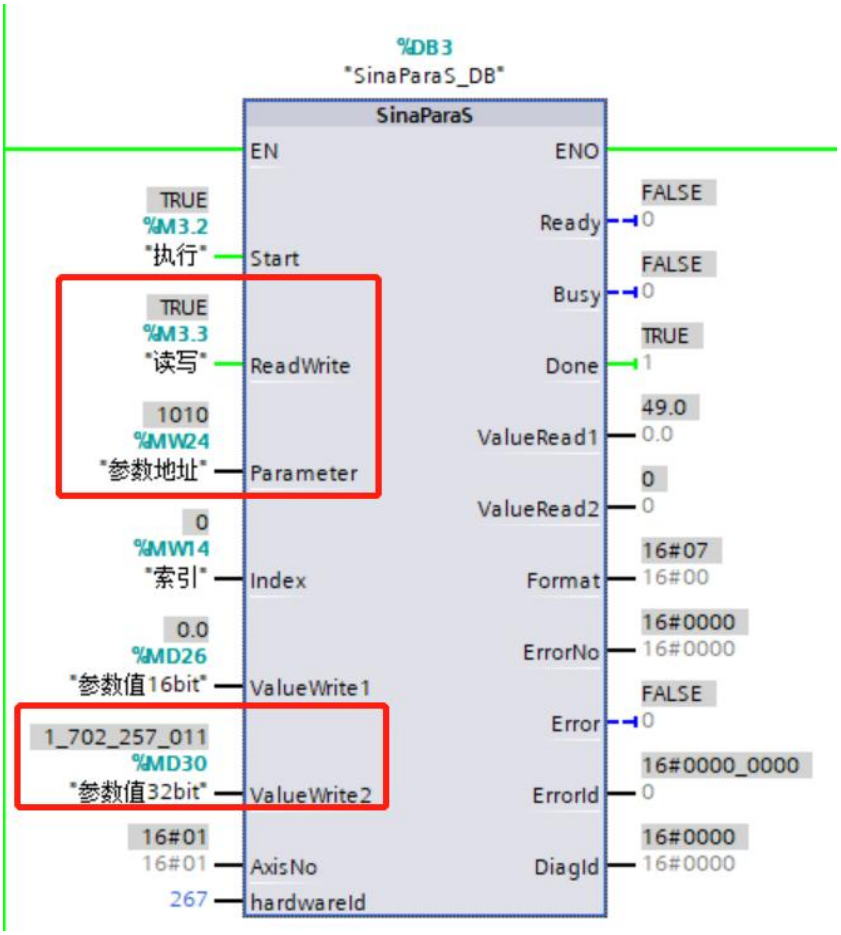


④ After writing successfully, read the current value again and confirm the parameter is written successfully



3. Write the power-off saving parameters

After all the parameters that need to be modified are written, write the "Power-off save" parameter 1010=1702257011 to save all parameters; otherwise, all unsaved parameters will be restored to the original values after power off.

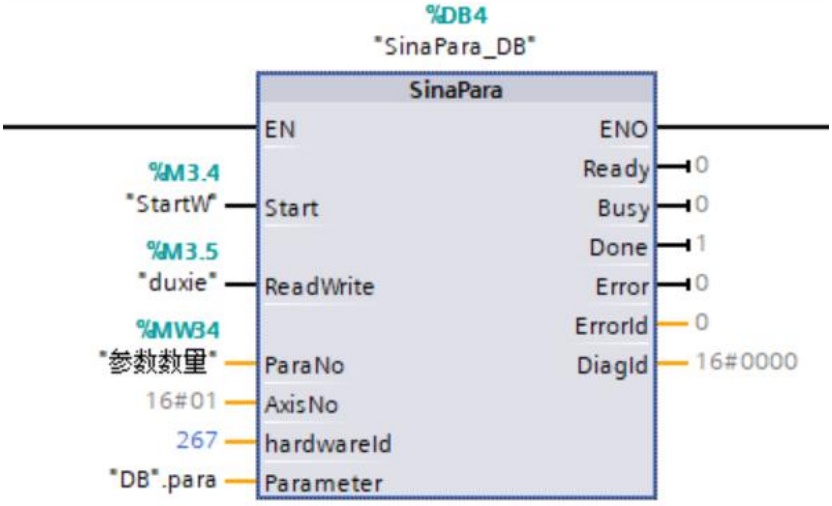


- SinaParaS can only write a single parameter address
- After writing all parameters, write the "power-off save" parameter once to save the parameters

7.4 Description of SinaPara instruction

SinaPara is a function block that can read and write multiple driver parameters at a time, and the array is required during use.

At present, the driver can read and write up to 5 parameters simultaneously.



- **Description of SinaPara pin**

Pin	Data type	Description
Start	Bool	False=No operation; True=Perform read/write operations
ReadWrite	Bool	False=Read; True=Write
ParaNo	Int	The number of parameters that need to be read and written
AxisNo	Int	Axis number; fill in 1 regardless of how many axes there are
hardwareId	HW IO	Hardware ID, which can be viewed by right-clicking the Message 111 - Properties
Ready	BOOL	Feedback signal connected in the LAcycCom environment; 1=End of job or cancellation of job (one cycle)
Busy	BOOL	Ongoing job (if "Busy"=1)
Done	BOOL	If the job is ended correctly, it indicates that the edge changes from 0 to 1
Error	BOOL	Activate the group fault ->"Error"=1
Status	DWORD	The first word -> indicates in binary encoding form which parameter access fault occurred The second word: Fault type
DiagId	WORD	Extended communication error ->SFB call error

7.5 Use Example of SinaPara

The following demonstrates changing the following 5 parameters through the SinaPara function block

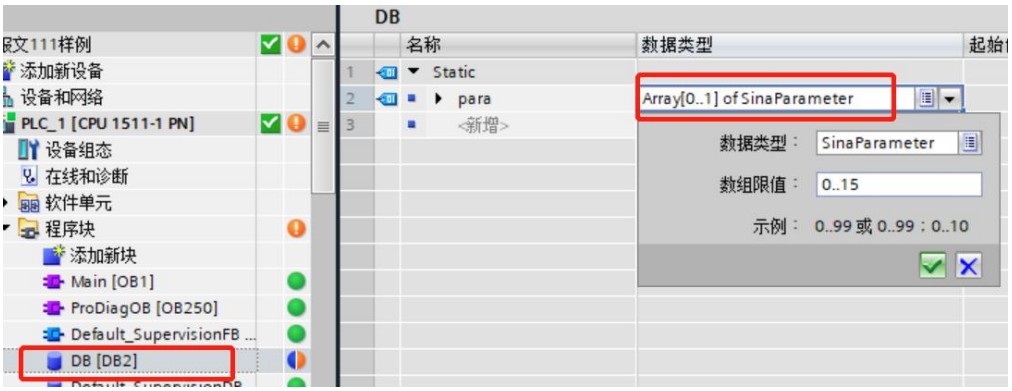
Address	Definition	Data type	Default value	Unit	Description
804	Driver subdivision number	32 bits	4000	Pulse/turn	Number of pulses per revolution of the motor
848	Forward jog speed	16 bits	1000	0.01 rps	Forward jog speed of Message 111 FB284 Mode 7, e.g. 200=2 rps
891	Motor homing low speed	16 bits	50	0.01 rps	Motor homing low speed in Message 111 FB284 Mode 4
892	Motor homing high speed	16 bits	100	0.01 rps	Motor homing high speed in Message 111 FB284 Mode 4
1010	Power-off saving of parameters	32 bits	0	-	Write 1010=1702257011, and all parameters will be saved after power off;

1. Create a new parameter array

① Create a new DB data block

② Add data with the data type of "Array of SinaParameter"

③ Modify the array limit within a range of 0-15.



④ Expand the structure array, mainly focusing on these three structure members

Structure members	Definition
siParaNo	Parameter address
srValue	16-bit parameter value
sdValue	32-bit parameter value

⑤ Fill the parameter address that needs to be modified into the "siParaNo" in the structure

Static		
para	Array[0..15] of SinaParameter	
para[0]	SinaParameter	
siParaNo	UInt	804
siIndex	UInt	0
srValue	Real	0.0
sdValue	DInt	0
syFormat	Byte	BYTE#16#00
swErrorNo	Word	WORD#16#0000
para[1]	SinaParameter	
siParaNo	UInt	848
siIndex	UInt	0
srValue	Real	0.0
sdValue	DInt	0
syFormat	Byte	BYTE#16#00
swErrorNo	Word	WORD#16#0000
para[2]	SinaParameter	
siParaNo	UInt	891

⑥ Fill the "Power-off save" parameter address 1010 into the last structure (this example is para[4], which is the fifth), so that after writing successfully, the parameters can be saved after power off.

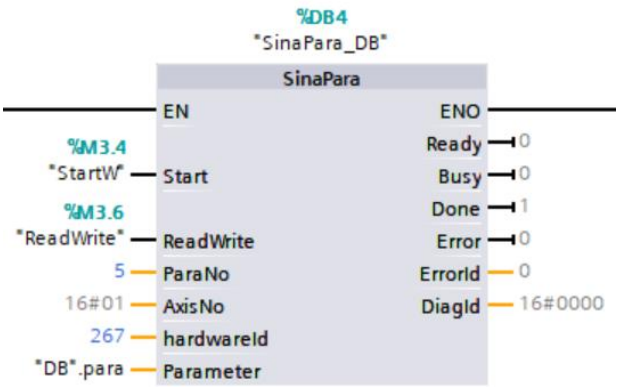
31	para[4]	SinaParameter	
32	siParaNo	UInt	1010
33	siIndex	UInt	0
34	srValue	Real	0.0
35	sdValue	DInt	0
36	syFormat	Byte	BYTE#16#00
37	swErrorNo	Word	WORD#16#0000

2. Add the function block SinaPara

① "ParaNo": Fill in the number of parameters needing to be read and written

② "hardwareId": Fill in the hardware identifier

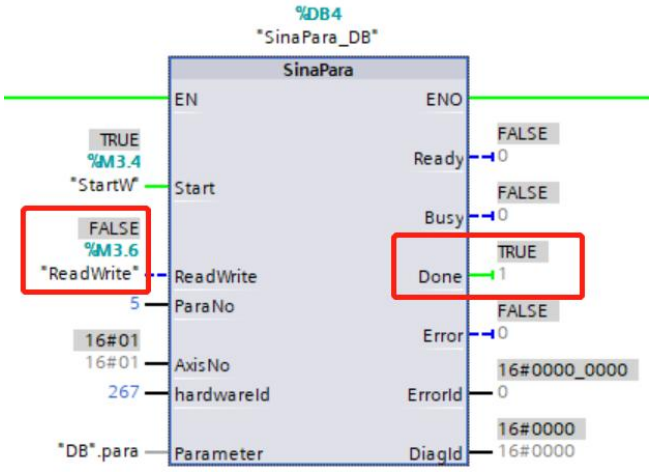
③ "Parameter": Fill in the newly added array



3. Use SinaPara to read parameters

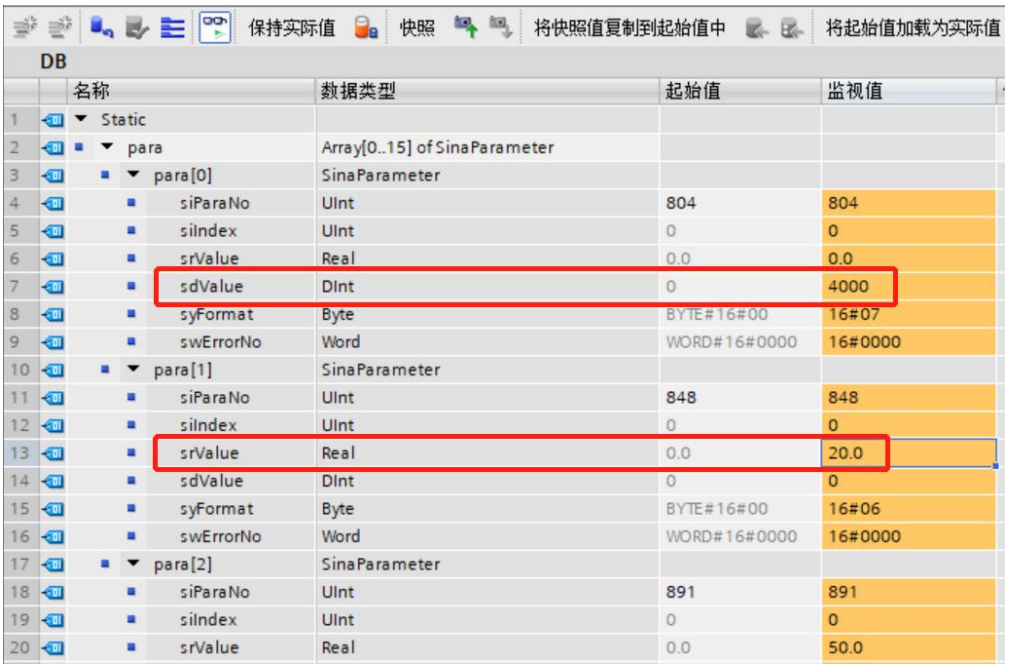
"ReadWrite"=false means reading parameters, and=true means writing parameters

If "Done" is true, it means successful reading



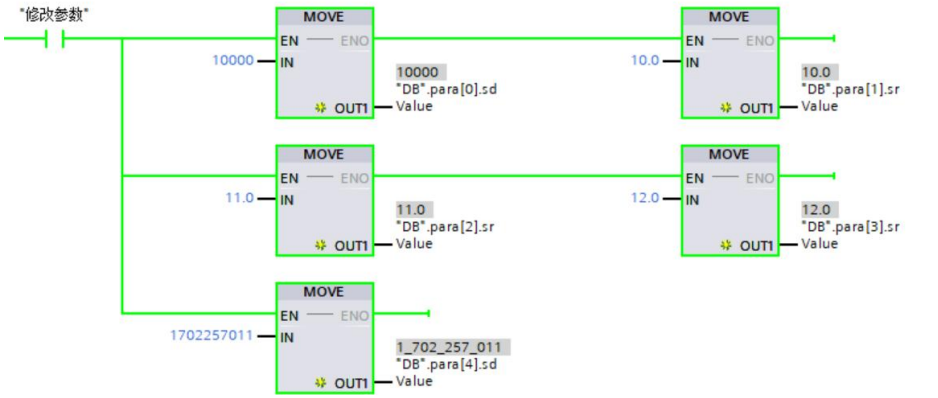
Online monitoring array can display the read parameters.

The 32-bit parameter value will be displayed in "sdValue", and the 16-bit parameter value will be displayed in "srValue".

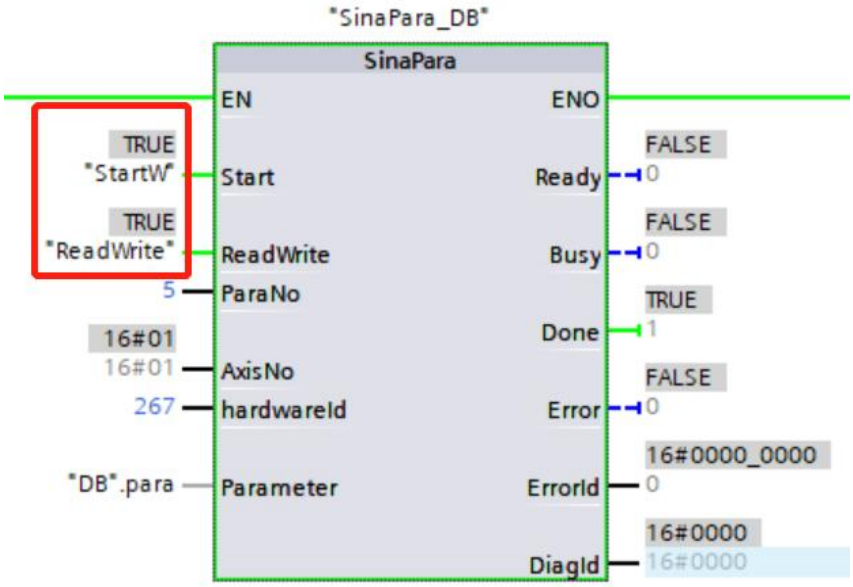


4. Use SinaPara to write the parameter

① Modify the parameter



② Write the parameter



③ Verify if the writing is successful

As the power-off saved parameters have been written to the above examples, we can restart the driver after power-off, and use SinaPara or Kaifull software to read the parameters and verify if it is successful.

DB

名称	数据类型	起始值	监视值
siParaNo	UInt	804	804
siIndex	UInt	0	0
srValue	Real	0.0	0.0
sdValue	DInt	0	10000
syFormat	Byte	BYTE#16#00	16#07
swErrorNo	Word	WORD#16#0000	16#0000
para[1]	SinaParameter		
siParaNo	UInt	848	848
siIndex	UInt	0	0
srValue	Real	0.0	10.0
sdValue	DInt	0	0
syFormat	Byte	BYTE#16#00	16#06
swErrorNo	Word	WORD#16#0000	16#0000
para[2]	SinaParameter		
siParaNo	UInt	891	891
siIndex	UInt	0	0
srValue	Real	0.0	11.0
sdValue	DInt	0	0
syFormat	Byte	BYTE#16#00	16#06
swErrorNo	Word	WORD#16#0000	16#0000

Monitor and display the read parameters to ensure they are consistent with the written parameters.

8 Alarm Code

The Y2SS3-PN driver displays statuses through a combination of flashing green and red LED indicator lights, with the specific meaning as follows:

LED indicator light	Meaning	Resolution
Green light flashing with an interval of 1 second	Motor not enabled	-
Green light flashing with an interval of 0.5 seconds	Motor enabled normally	-
4 red lights and 1 green light	Excessively high bus voltage	1. Check whether the supply voltage of the driver is too high; 2. In case of overvoltage during movement, the motor deceleration time can be increased.
4 red LED lights and 2 green LED lights	Excessively low bus voltage	Check whether the supply voltage of the driver is too low;
5 red lights +1 green light	Motor overcurrent	1. Check whether the motor has been damaged; 2. Check whether the set current of the driver is too high;
6 red lights +1 green light	Motor open circuit	1. Check whether the motor wiring is correct; 2. Check whether the motor has been damaged
5 red LED lights and 2 green LED lights	Position deviation	1. Check whether the encoder wire is connected correctly and securely 2. Check whether the encoder resolution is set correctly 3. Check whether the motor is blocked 4. Increase the acceleration and deceleration time appropriately

9 Contact Kaifull

更多更新的凯福咨询，请扫码关注

微信公众号　　官网

广东凯福电子科技有限公司
GUANGDONG KAIFULL ELECTRONICS TECHNOLOGY CO., LTD.

■ 苏州分公司　■ 宁波分公司　■ 武汉分公司　■ 上海分公司

电话：0769-23033384
传真：0769-22493047
网站：www.kaifull.net
详细地址：广东省东莞市高埗镇高龙东路5号 凯福科技园