FM series fieldbus control stepper motor drive

User manual

Kinco Electric (shenzhen) Ltd.
en.kinco.cn

Content

Revision record

Version	Date	Description
1.0	2013-06-25	1.0 version publish
1.1	2013-09-06	Modify drive and motor parameters Add contents 4.4.2-4.4.4
1.2	2013-10-18	Modify the text
1.3	2014-7-9	Correct a mistake
1.4	2018-12-13	Add FM880 drive
1.5	2020-9-14	Add FM560 drive

Safety Precaution

To avoid personal injury and property damage, please read the following safety information carefully before commissioning and using the drive.

The following safety measures must be strictly observed:

- FM driver in normal operation will have a maximum internal low voltage of about 70VDC, cut off the power of the driver within 30 seconds, the driver still has a certain voltage, please wait until the driver voltage drops to the safe range, then wiring or check, otherwise it may be electric shock. Do not wire the driver and motor while they are working, otherwise you may get an electric shock.
- Do not wire while the driver and motor are working, as there is a risk of electric shock
- To prevent personal injury and property damage, only personnel with relevant professional knowledge are allowed to operate the driver.
- Follow the technical specifications and electrical installation standards during installation. The driver must be well grounded.
- Please do not put any object into the drive, otherwise it may cause damage to the drive.
- If the drive is faulty and needs to be repaired, return the drive to the repair center. Unauthorized opening of the drive or incorrect operation can damage the drive. If you open the drive shell without permission, the warranty will be void.
- When discarding the drive, please treat it according to the standard of industrial waste, so as not to cause environmental pollution.

Warning

- When the driver is applied to mechanical equipment directly involved in personal safety (nuclear power control, medical equipment, trucks, trains, aircraft, such as music and safety protection equipment, etc.), it is necessary to install preventive installation equipment to avoid possible personal injury.
- Electronic devices have a corresponding life cycle. Machinery and equipment must have adequate safety measures to ensure the safety of personnel and equipment itself in the event of driver failure. Customers who install or use the drive must bear the costs of machine failure and misoperation of the drive.

Chapter 1 Product overview

1.1Product confirmation

Thank you for choosing the Kinco Stepper Drive product. After receiving the product, please check the following items carefully:

- Check the nameplate on the packing case and body to make sure the driver model is the same as the one you ordered.
- After opening the product package, please confirm that there are no abnormal conditions such as damage or missing parts.
- Please confirm that all fixed screws on the drive are loose.
- Please check the received products according to the product list, if missing, please contact customer service.

Product List	
Item	**count**
Drive	1
Bus control stepper driver usage guide	1
Service Directory	1
Certificate of Conformity	1
SCSI 20P plug	1
5mm pitch size terminal(6 Pin)	1
Length 10mm diameterΦ 10 insulated terminal	6
Length 15mm diameterΦ 12 insulated terminal	6
Console cable(RS232 serial port to RJ45 crystal head cable)	1

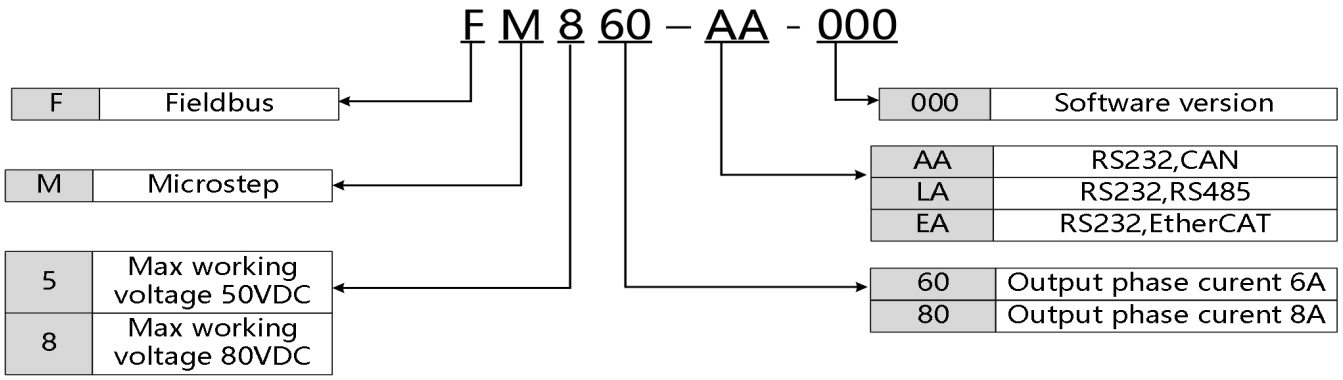
Note: Console cable is an optional item. Please contact customer service if you have any requirements.

1.2 Driver nameplate

Figure 1-1 Driver nameplate

1.3 Product model description

1.3.1 Naming rule



1.3.2 Product feature

Table 1-1 Product specification

Model	FM860-AA/LA-000	FM880-EA-000	FM560-EA-000
Input voltage	24~70VDC	24VDC∽70VDC	24VDC∽50VDC
Over-voltage protection	Above 85VDC		
Under-voltage protection	Below 15VDC		
Output current	0.15～8A （Peak）	0.1～10A （Peak）	0.1～6A （Peak）
Subdivision	Set by "KincoStep" software		
Adaptable motor	42/57/86series 2-phase or57/85series 3-phase hybrid stepping motor	Two-phase86 and three-phase 57,85 hybrid stepping motor	Two-phase type 42, 57 hybrid stepping motor
Control mode	CANopen or Modbus control;I/O control;Analog control, pulse control：PLS+DIR,CW/CCW, A+B	EtherCAT bus protocol control,IOcontrol,analog control pulse control: PLS+DIR ,CW/CCW,A+Bphase	
Cooling method	Natural cooling		
Weight (Suttle weight)	0.36 Kg	0.363Kg	
IO	6 digital inputs , three of which support wide voltageinput(5~24VDC); 3 digital outputs; 1 analog input channel(±10V) speed control ;		
Protection function	Over voltage,under voltage,over heat and over current protection		

Chapter 2 Precautions and installation requirement

2.1 Precaution

- Tightly fasten the screws that fix the motor.
- Make sure to tightly fasten all fixed points when fixing the driver.
- Do not tighten the cables between the driver and the motor.
- Use a coupling shaft or expansion sleeve to ensure that both the motor shaft and equipment shaft are properly centered.
- Do not mix conductive materials (such as screws and metal filings) or combustible materials (such as oil) into the driver.
- Avoid the driver and stepper motor from dropping or striking because they are precision equipment.
- For safety, do not use any damaged driver or any driver with damaged parts.
- The use of the driver in the industrial electromagnetic environment has requirements for EMC, which needs to connect the power filter and add a magnetic ring to the motor cable.

2.2 Environment Condition

Table 2-1 Environment requirement

Environment	Condition
Temperature	Operation temperature：0℃～40℃（ice-free） Storage temperature：-10℃～70℃（ice-free）
Humidity	Operating humidity：Less than 90%RH (no condensation) Storage humidity: Less than 90%RH (no condensation)
Air	Indoor (No direct sunlight), no corrosive gas or combustible gas, no oil vapor or dust
altitude	Below 1000m above the sea level
Vibration	5.9 m/s^2

2.3 Mounting direction and spacing

The driver should be vertically installed on wall,and the upper and lower spacing should be reserved.Take fully into account heat dissipation when using any heating components (such as braking resistors) so that the driver is not affected.

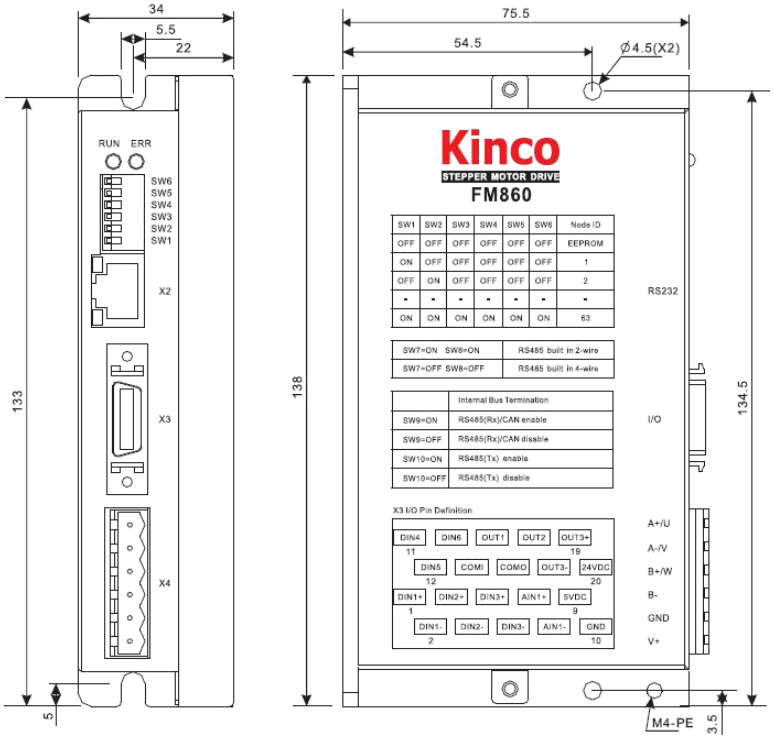


Figure 2-1 FM860 dimension

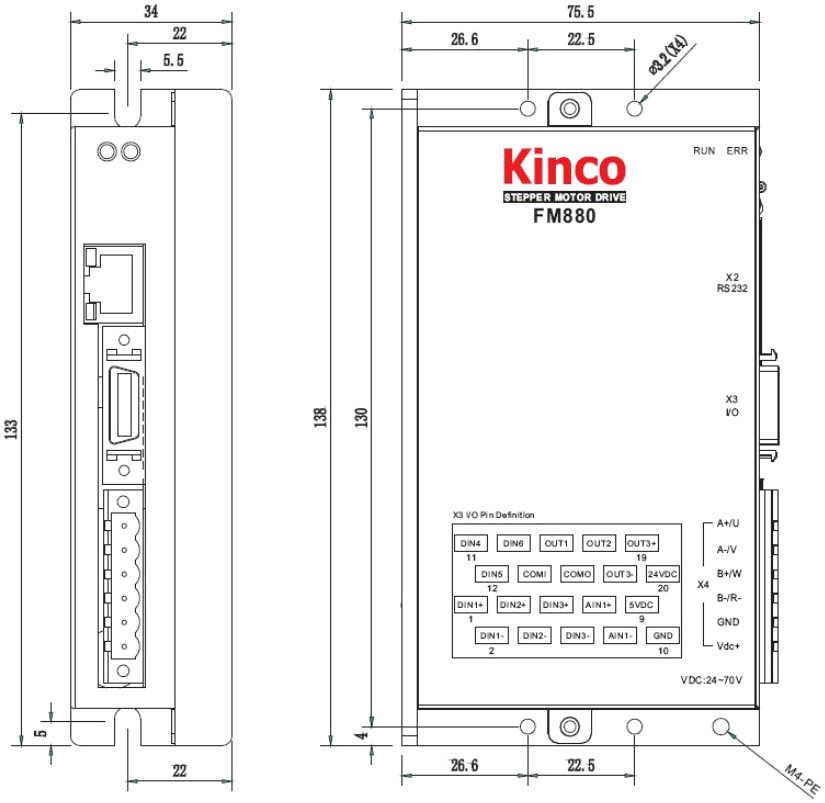


Figure 2-2 FM880 dimension

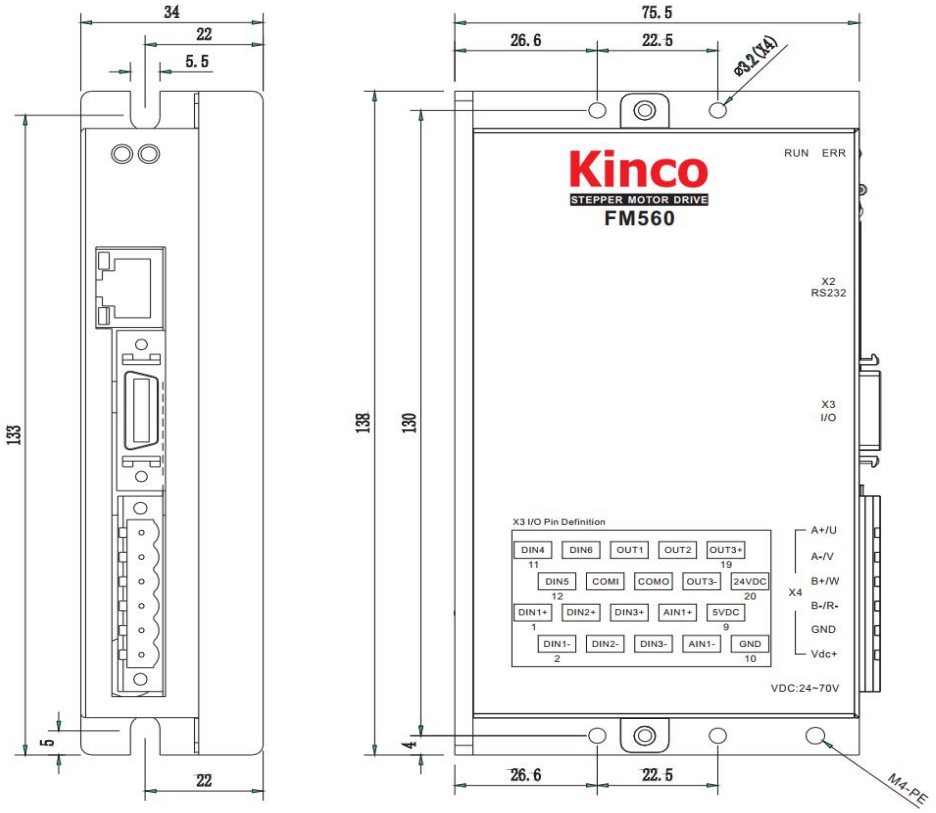


Figure 2-3 FM560 dimension

Chapter 3 Drive introduction

3.1 FM860

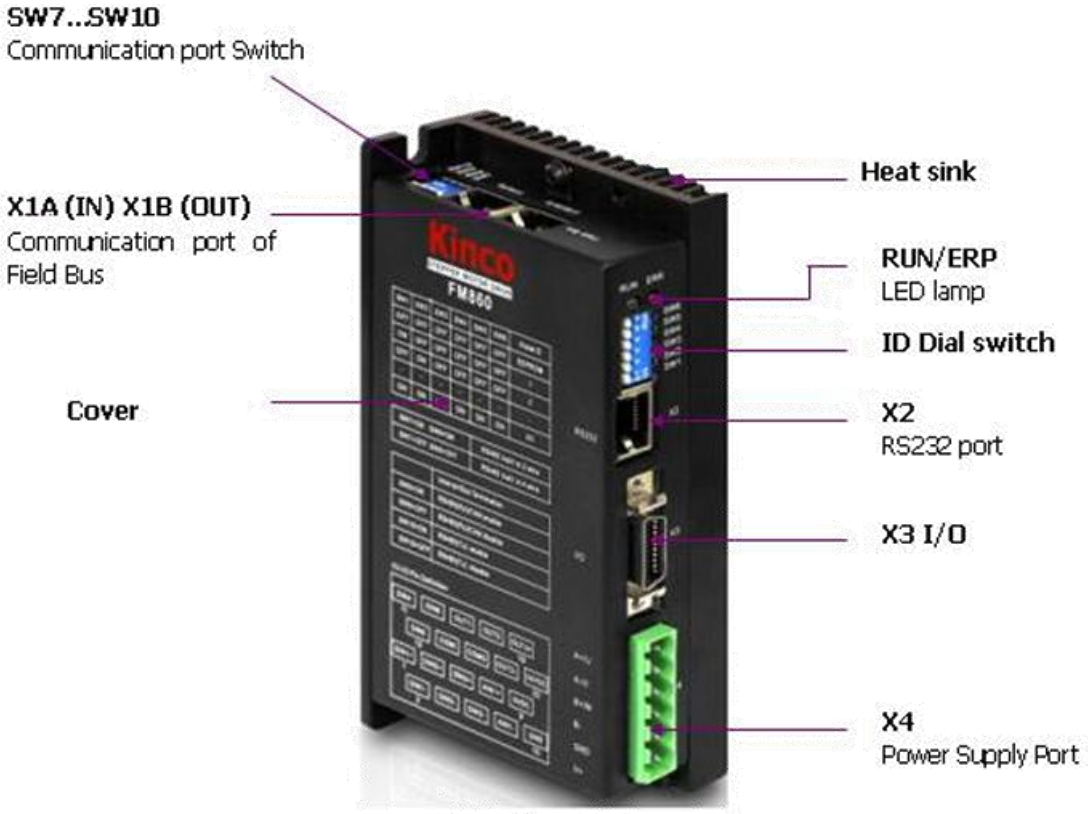


Figure 3-1 FM860 overview

3.1.1 Interface introduction

Table 3-1 Interface introduction

Interface		Symbol	Function
Field Bus		SW7～SW10	Communication port Switch
		X1A(IN) X1B(OUT)	CAN Bus Or RS485 Bus interface
ID Switch		SW1～SW6	ID Switch
RS232		X2	RS232 interface
IO	DIN1+	X3	Digital signal input
	DIN1-		
	DIN2+		
	DIN2-		
	DIN3+		
	DIN3-		
	DIN4		
	DIN5		
	DIN6		
	COMI		

	AIN1+		Analog signal input and logic voltage interface
	AIN1-		
	GND		
	5VDC		
	24VDC		
	OUT1		Digital signal output
	OUT2		
	COMO		
	OUT3+		
	OUT3-		
Power and motor interface	A+/U	X4	2 or 3 phase stepper motor cable interface
	A-/V		
	B+/W		
	B-		
	GND		Power supply input DC24-70V
	Vdc+		

3.1.2 DIP switch

3.1.2.1 ID Setting up

Table 3-2 ID setting up

SW1	SW2	SW3	SW4	SW5	SW6	Node ID
OFF	OFF	OFF	OFF	OFF	OFF	EEPROM
ON	OFF	OFF	OFF	OFF	OFF	1
OFF	ON	OFF	OFF	OFF	OFF	2
ON	ON	OFF	OFF	OFF	OFF	3
-	-	-	-	-	-	-
ON	ON	ON	ON	ON	ON	63

Note: Pease use the PC software to set and save your parameters when Node-ID is bigger than 63, and all of switch have to be set as OFF. And use the value of EEPROM when it is power-on, but not in value of switch.

3.1.2.2 Communication interface switch setting

Table 3-3Communication port

SW7=ON，SW8=ON	RS485 2-wire mode (half-duplex data transmission)
SW7=OFF，SW8=OFF	RS485 4-wire mode (full duplex data transmission)

		120Ω DIP switch of termination resistors	
CAN bus		Used when SW9=ON，SW10=OFF	SW9=OFF，SW10=OFF Disable
RS485 bus	2-wire mode	Used when SW9=ON，SW10=OFF	SW9=OFF，SW10=OFF Disable
	4-wire mode	Used when SW9=ON，SW10=ON	SW9=OFF，SW10=OFF Disable

3.1.3 I/O

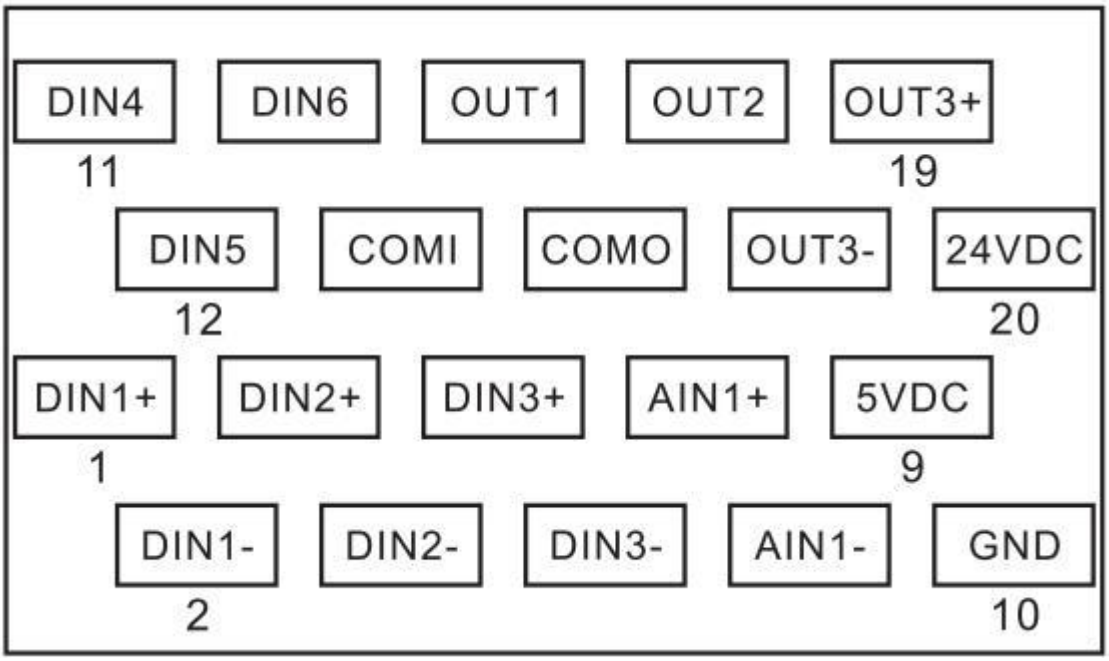


Figure 3-2 Wiring diagram

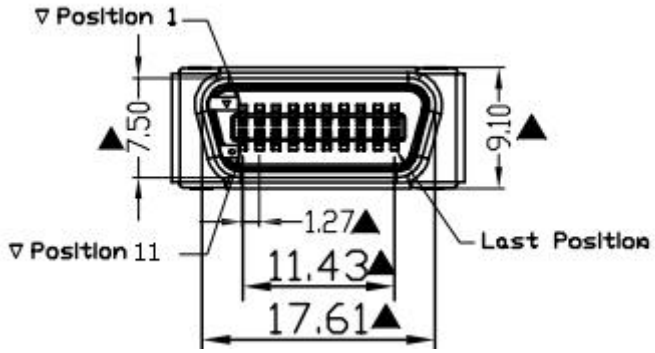


Figure 3-3 I/O pins

Table 3-4 IO function definition

Name	SCSI pin	Signal	Description	Function
X3 I/O	1	DIN1+	DIN1+ input	High speed digital signal input interface Input voltage：5～24VDC Input current：8mA@5VDC,12mA@24VDC Effective input signal: ＞3VDC Invalid input signal: ＜1.5VDC Maximum input frequency:400KHz
	2	DIN1-	DIN1- input	
	3	DIN2+	DIN2+ input	
	4	DIN2-	DIN2- input	
	5	DIN3+	DIN3+ input	
	6	DIN3-	DIN3- input	
	11	DIN4	DIN4 input	Low speed digital signal input interface Input voltage：12～24VDC Input current：4mA@12VDC,8mA@24VDC Effective input signal:＞8VDC Invalid input signal:＜5VDC Maximum input frequency:10kHz
	12	DIN5	DIN5 input	
	13	DIN6	DIN6 input	
	14	COMI	DIN4，DIN5，DIN6 Input common terminal	

	7	AIN1+	AIN1+ Differential input	Analog signal input interface Input impedance:180K Max input frequency:4kHz Max withstand voltage:24VDC
	8	AIN1-	AIN1- Differential input	
	10	GND	Common port both for AIN1and Logic power	
	9	5VDC	5VDC logic power output	Max output current:200mA
	20	24VDC	24VDC logic power input	Auxiliary logic power supply when external power supply does not work
	15	OUT1	OUT1 output	Max output current：100mA Max voltage drop：0.8VDC@100mA Max withstand voltage:30VDC Max output frequency:1kHz
	17	OUT2	OUT2 output	
	16	COMO	OUT1,OUT2 Output common terminal	
	19	OUT3+	OUT3+ output	
	18	OUT3-	OUT3- output	

3.1.4 Computer interface

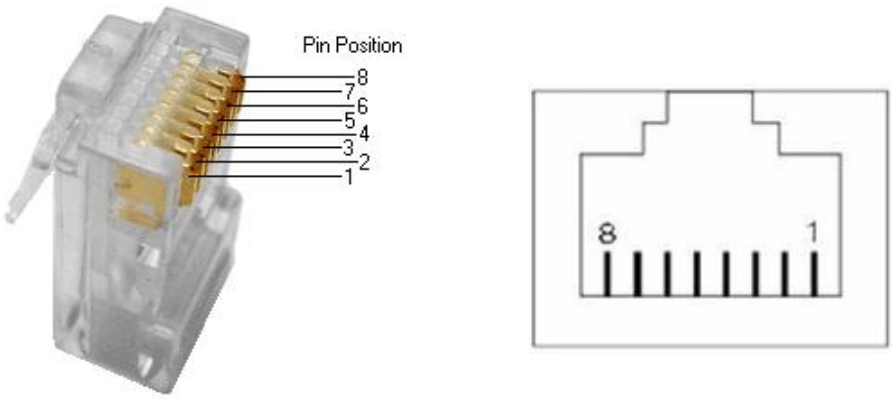


Figure 3-4 RJ45 plug and socket

3.1.4.1 RS232 interface

Table 3-5 RS232 definition

Name	RJ45 pin	Signal	Description	Function
X2 RS232	1	NC	-	RS232 communication interface Default baud rate：38400 Communication data bit：8 Stop bit：1 None Parity bit
	2	NC	-	
	3	TX	Transmitted data	
	4	GND	Signal ground	
	5	NC	-	
	6	RX	Received data	
	7	NC	-	
	8	NC	-	

Note：You can use the console cable，convert DB9 needle interface

3.1.4.2 CAN interface

Table 3-6 CAN definition

Name	RJ45 pin	Signal	Description	Function
X1 CAN	1	CAN_H	CAN differential signal	CAN bus interface Default baud rate: 500K
	2	CAN_L		
	3	GND	Signal Ground	
	4	NC	-	
	5	NC	-	
	6	NC	-	
	7	NC	-	
	8	GND	Signal Ground	

3.1.4.3 RS485 interface

Table 3-7 RS485 definition

Name	RJ45 pin	Signal	Description	Function
X1 RS485	1	RX	Received data	RS485 communication interface Default baud rate：19200 Communication data bit：8 Stop bit：1 None Parity bit
	2	/RX		
	3	GND	Signal Ground	
	4	/TX	Transmitted data	
	5	TX		
	6	NC	-	
	7	NC	-	
	8	GND	Signal Ground	

Note：

1）The CAN interface and RS485 interface cannot coexist on the same driver. X1 of FM860-AA-000 is the CAN interface,and X1 of FM860-LA-000 is the RS485 interface.

2）There are two X1 interfaces, X1A (IN) and X1B (OUT). The pins of the two interfaces are directly connected and have the same functions. This is mainly to facilitate use of standard network cables and connect the driver in parallel to the bus.

3）RS232 can use the console cable，convert DB9 needle interface

3.1.5 X1 and X2 LED lamp definition

Table 3-8 LED definition

Name	Color	Definition
X2	Green	Power instruction
	Orange	RS232 receive instruction
X1	Green	Fieldbus send instruction
	Orange	Fieldbus receive instruction

3.2 FM880&FM560

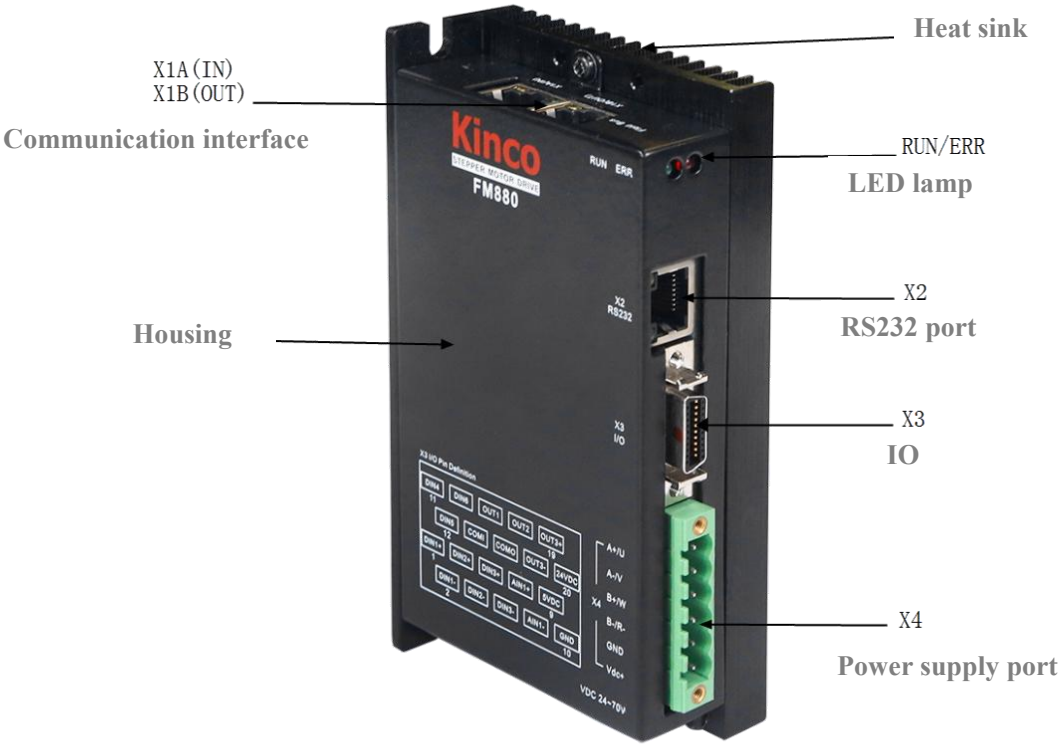


Figure 3-5 FM880 overview

3.2.1Interface introduction

The FM880 and FM560 driver pin definitions are the same, and this section of interface definition introduction applies to both FM560 and FM880 drivers.

Table 3-9 Interface introduction

Interface		Symbol	Function
Fieldbus		X1A(IN) X1B(OUT)	EtherCAT bus interface
RS232		X2	RS232 interface
IO	DIN1+	X3	Digital signal input interface
	DIN1-		
	DIN2+		
	DIN2-		
	DIN3+		
	DIN3-		
	DIN4		
	DIN5		
	DIN6		
	COMI		
	AIN1+		Analog signal input interface Logical voltage interface
	AIN1-		
	GND		

	5VDC		
	24VDC		
	OUT1		Digital signal output interface
	OUT2		
	COMO		
	OUT3+		
	OUT3-		
Power and motor interface	A+/U	X4	2 or 3 phase stepper motor cable interface
	A-/V		
	B+/W		
	B-/R-		
	GND		Power supply input DC24-70V(FM880) Power supply input DC24-50V(FM560)
	Vdc+		

3.2.2 I/O

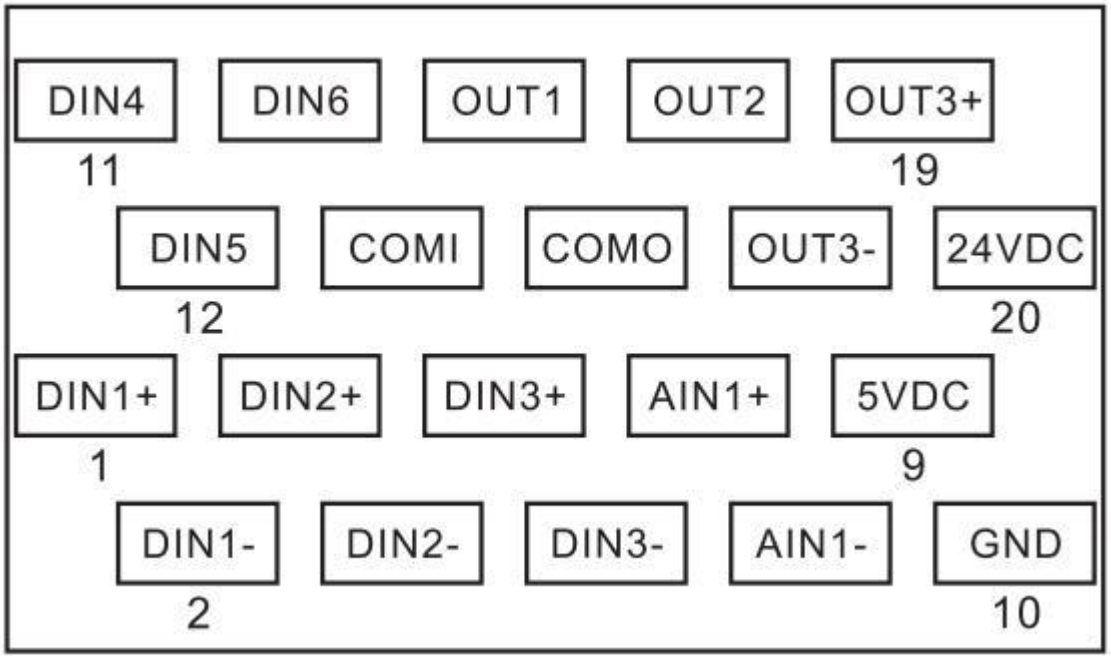


Figure 3-6 Wiring Diagram

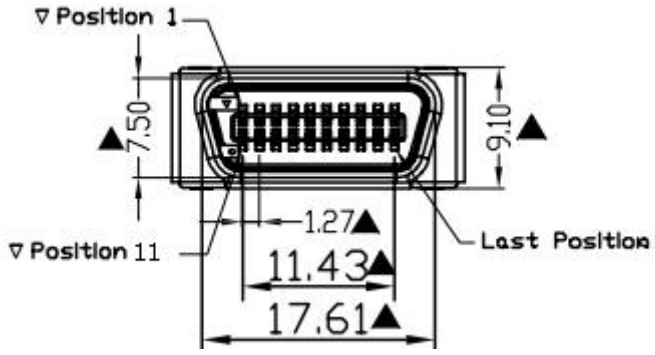


Figure 3-7 I/O pins

Table 3-10 IO function definition

Name	SCSI pin	Signal	Description	Function
X3 I/O	1	DIN1+	DIN1+ input	High speed digital signal input interface Input voltage：5～24VDC Input current：8mA@5VDC,12mA@24VDC Effective input signal: ＞3VDC Invalid input signal: ＜1.5VDC Maximum input frequency:400KHz
	2	DIN1-	DIN1- input	
	3	DIN2+	DIN2+ input	
	4	DIN2-	DIN2- input	
	5	DIN3+	DIN3+ input	
	6	DIN3-	DIN3- input	
	11	DIN4	DIN4 input	Low speed digital signal input interface Input voltage：12～24VDC Input current：4mA@12VDC,8mA@24VDC Effective input signal:＞8VDC Invalid input signal:＜5VDC Maximum input frequency:10kHz
	12	DIN5	DIN5 input	
	13	DIN6	DIN6 input	
	14	COMI	DIN4，DIN5，DIN6 Input common terminal	
	7	AIN1+	AIN1+ Differential input	Analog signal input interface Input impedance:180K Max input frequency:4kHz Max withstand voltage:24VDC
	8	AIN1-	AIN1- Differential input	
	10	GND	Common port both for AIN1and Logic power	
	9	5VDC	5VDC logic power output	Max output current:200mA
	20	24VDC	24VDC logic power input	Auxiliary logic power supply when external power supply does not work
	15	OUT1	OUT1 output	Out1 and OUT2 max output current：100mA OUT3 max output current：400mA Max voltage drop：0.8VDC@100mA Max withstand voltage:30VDC Max output frequency:1kHz
	17	OUT2	OUT2 output	
	16	COMO	OUT1,OUT2 Output common terminal	
	19	OUT3+	OUT3+ output	
	18	OUT3-	OUT3- output	

3.2.3 Computer interface

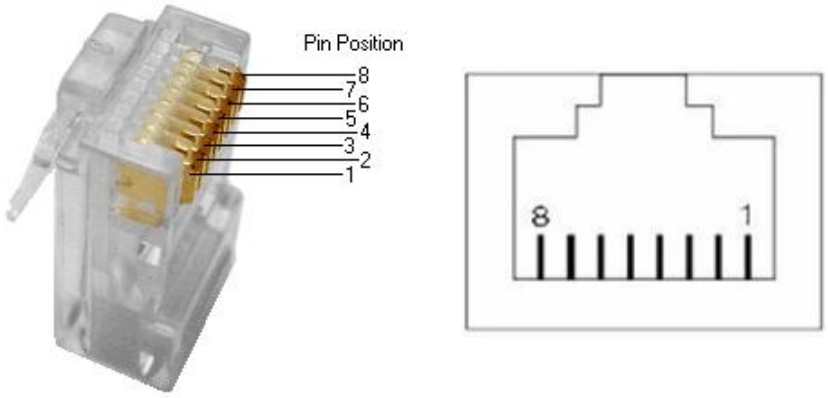


Figure 3-8 RJ45 plug and socket

3.2.3.1 RS232 interface

Table 3-11 RS232 definition

Name	RJ45 pin	Signal	Description	Function
X2 RS232	1	NC	-	RS232 communication interface Default baud rate：38400 Communication data bit：8 Stop bit：1 None Parity bit
	2	NC	-	
	3	TX	Transmitted data	
	4	GND	Signal ground	
	5	NC	-	
	6	RX	Received data	
	7	NC	-	
	8	NC	-	

Note：You can use the console cable，convert DB9 needle interface

3.2.3.2 EtherCAT interface

Table 3-12 EtherCAT definition

Name	RJ45 pin	Signal	Description	Function
X1 EtherCAT	1	TD+	Send signal positive end	EtherCATbus interface X1A signal input X1B signal output 100BASE-TX(IEEE802.3) HP AUTO-MDIX
	2	TD-	Send signal negative end	
	3	RD+	Receive signal positive end	
	4	NC	-	
	5	NC	-	
	6	RD-	Receive signal negative end	
	7	NC	-	
	8	NC	-	

Note：

1） There are two X1 interfaces, X1A (IN) and X1B (OUT). The pins of the two interfaces have the same name, but they are independent physical interfaces and are not directly connected to each other.

2） The EtherCAT port of the driver has the Auto-MDIX detection function, which automatically identifies whether the sending and receiving pairs need to be exchanged according to the status of the pins during operation, so the communication can be carried out by using the directly connected network cable or the crossed network cable.

3） Class 5 twisted pair (or higher standard) is recommended for field applications to reduce the introduction of field interference signals.

3.2.3.3 X1 and X2 LED lamp definition

Table 3-13 LED definition

Name		Definition	Status	Description
X2	Orange	RS232 receive indicator	Off	No data communication
			Fast blink	Have data communication
	Green	RS232 send indicator	Off	No data communication
			Fast blink	Have data communication

X1	Orange	EtherCATlink activity indicator	Off	No network connection
			Normally on	Network connection
			Fast blink	There is data communication on the network
	Green	EtherCAT running status indicator	Off	initialization
			Slow blink	pre-operational
			Single blink	safe-operational
			Normally on	operational
Note： Fast blink：On 50ms, off 50ms(10Hz) ,repeat in this way Slow blink：On 200ms, off 200ms(2.5Hz), repeat in this way Single blink: On 200ms, off 1s,repeat in this way				

3.3 Consolecable

The Console cable is the conversion cable between the driver and the computer. One side is connected to the computer RS232 (DB9 needle interface) and the other side is connected to the driver RS232 (RJ45 interface).

Figure 3-9 RS232 conversion cable (1)

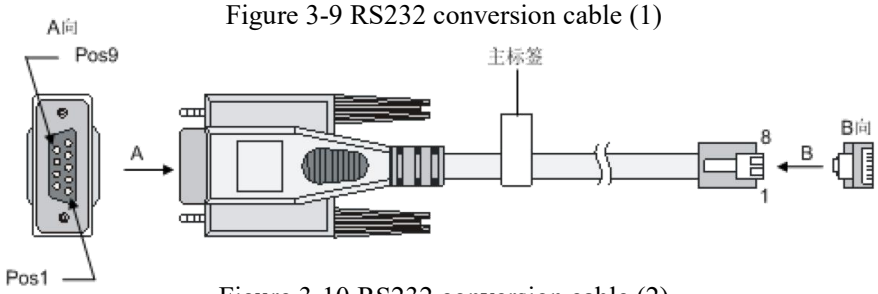


Figure 3-10 RS232 conversion cable (2)

The pin connection sequence of the conversion cable is as follows:

DB9 female		**RJ45（network cable port）**
RXD (2)	----------	TXD (3)
TXD (3)	----------	RXD (6)
DTR (4)	----------	DSR (7)
GND (5)	----------	GND (4)&(5)
DSR (6)	----------	DTR (2)
RTS (7)	----------	CTS (8)
CTS (8)	----------	RTS (1)

3.4 Driver external wiring diagram

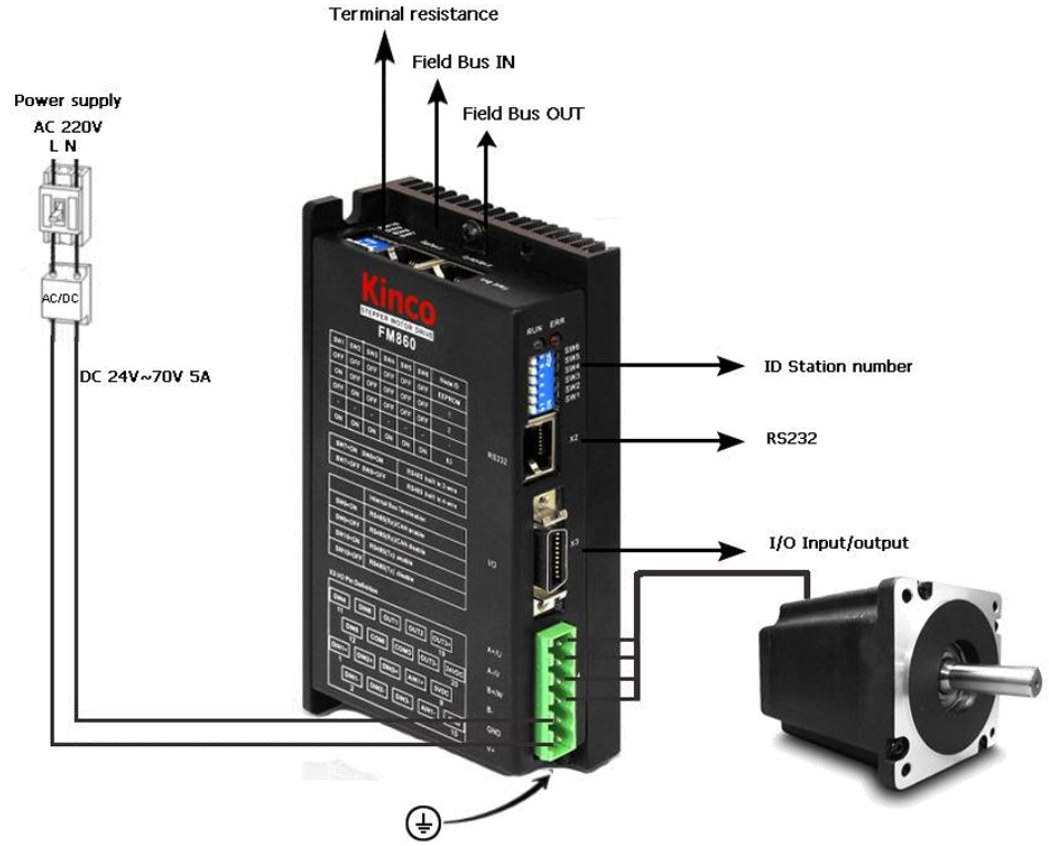


Figure 3-11 FM860 driver external wiring diagram

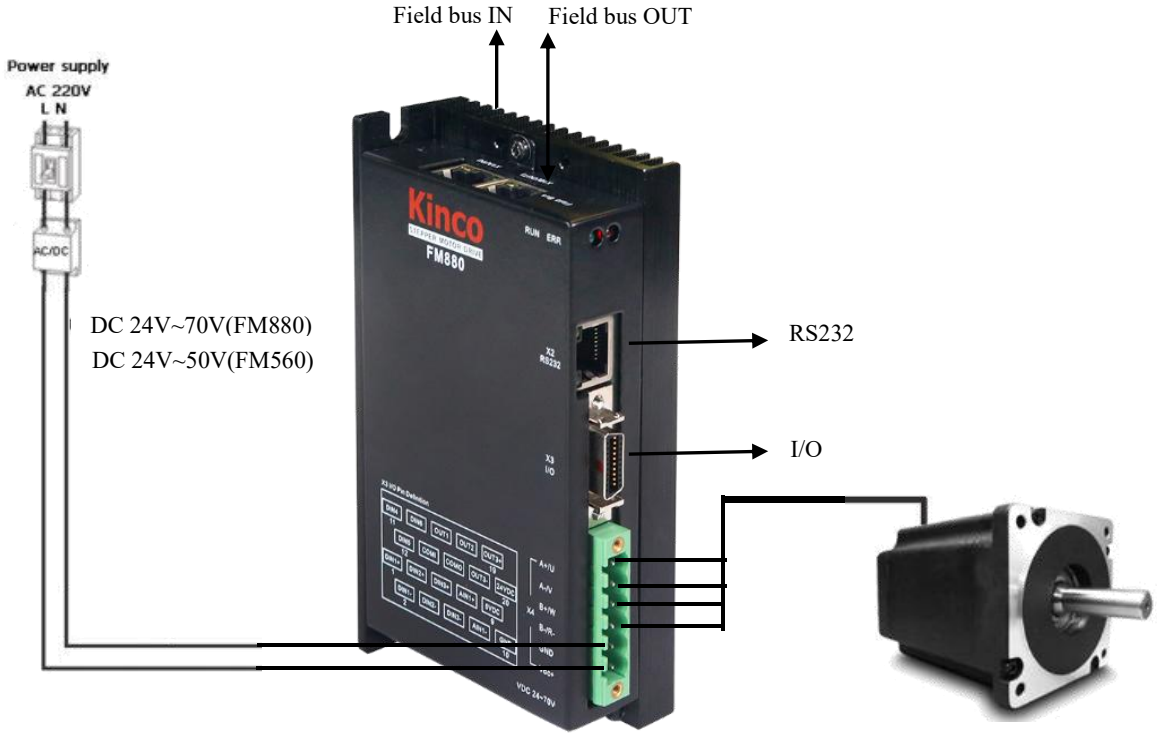


Figure 3-12 FM880/FM560 driver external wiring diagram

Chapter 4 Bus step debugging software instruction

4.1Software installation

Please access www.kinco.cn to download Kinco Stepping PC Software and unzip to use directly.

4.2 Quick start

4.2.1Debug software connection

You have to use Kinco Stepping PC software to set up the parameter by RS232 or CAN connector.

Please refer chapter 3 to connect step drive and motor before using.

Minimum system requirements for programming with RS232 ports:

- FM series stepper drivers, such as FM860
- The control logic voltage provided to the driver is 24VDC
- Console cable (see Appendix III)

PC 9 Pin D		**RS232（RJ45 connector)**
RxD (2)	----------	TXD (3)
TxD (3)	----------	RXD (6)
GND (5)	----------	GND (4)

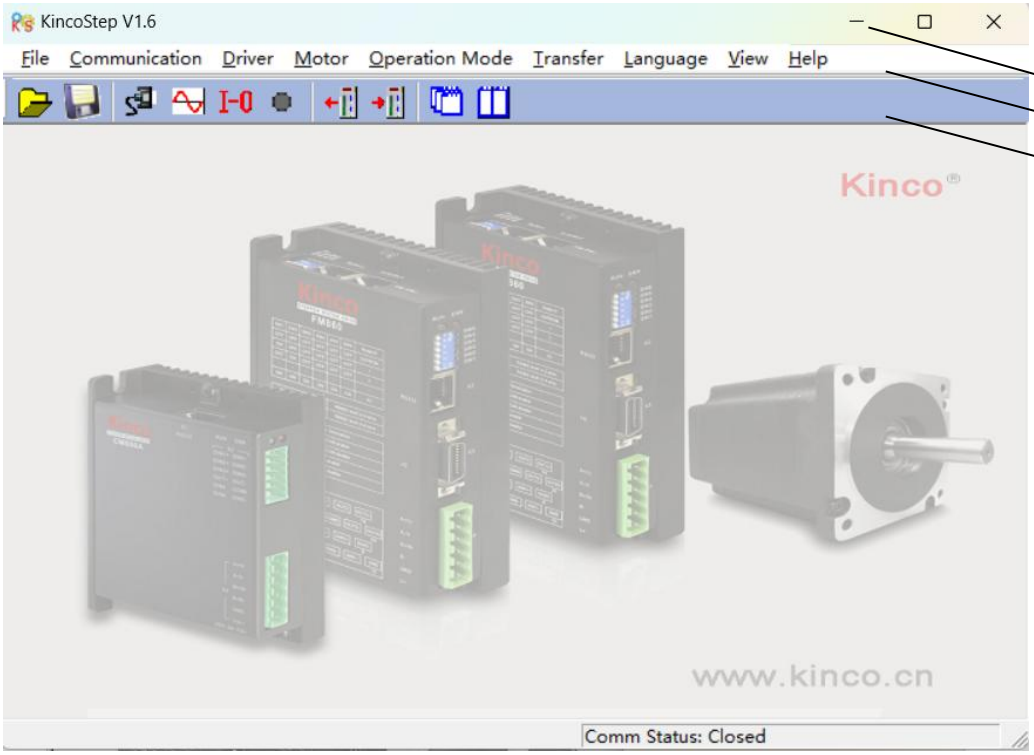
Minimum system requirements for programming with the CAN interface:

- FM860-AA
- The control logic voltage provided to the driver is 24VDC
- PEAK series USB or LPT adapter
- CAN communication cable, needn’t power supply external.

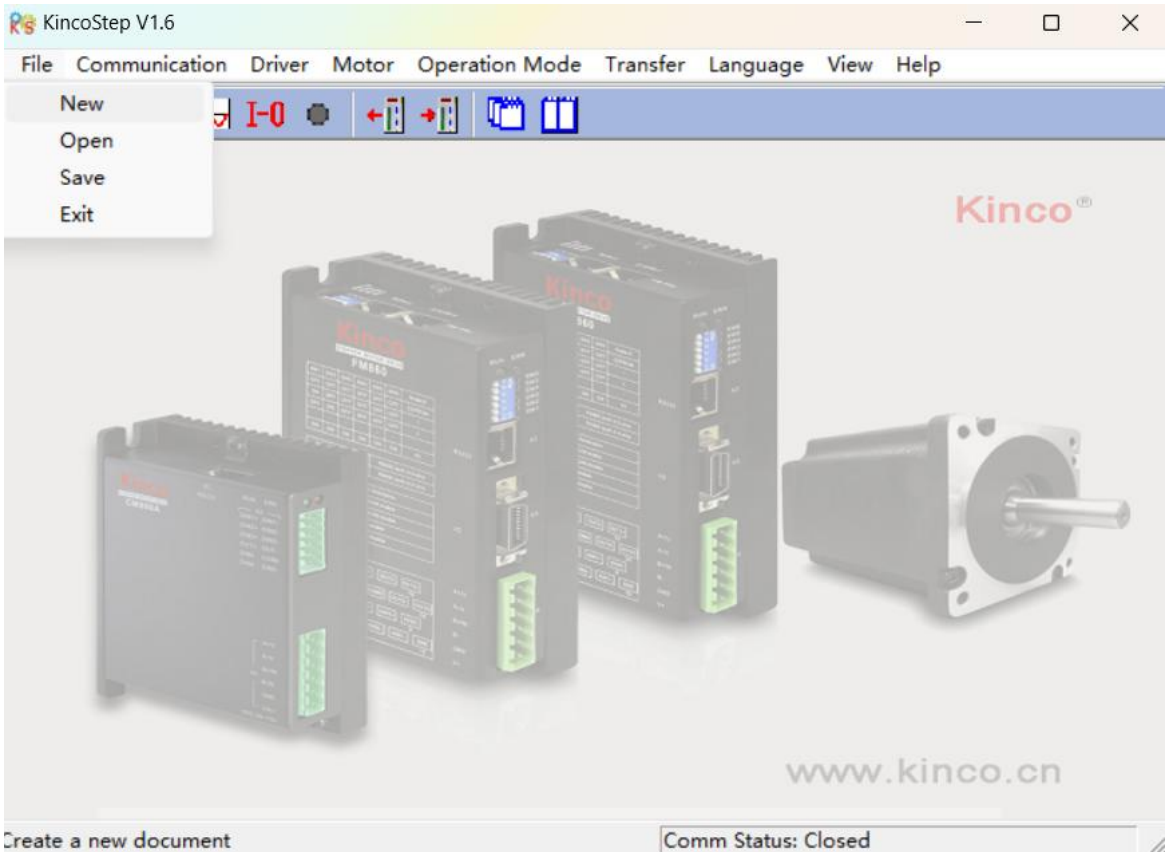
Pecan 9 Pin D		**CAN(RJ45 connector)**
CAN_L (2）	----------	CAN_L (2)
CAN_H (7)	----------	CAN_H (1)
GND (3)	----------	GND (3)

4.2.2 How to use Kinco Stepping PC software

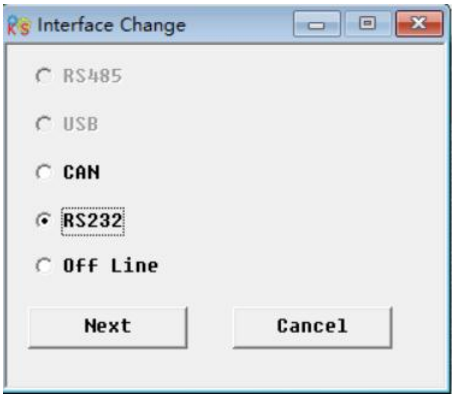
1.. Click the Kinco Stepping PC software folder and find the icon as Kincostep , and double click it to open the software as following picture:



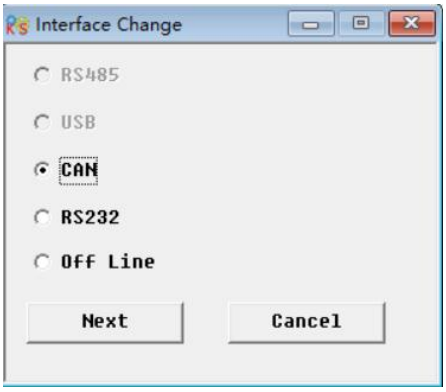
2.New project



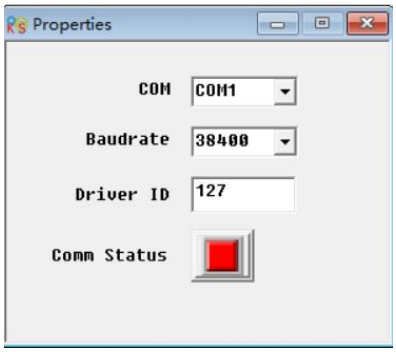
3.It will popup a dialog as following picture. Please click RS232 if the communication type is serial port, then click the Next.



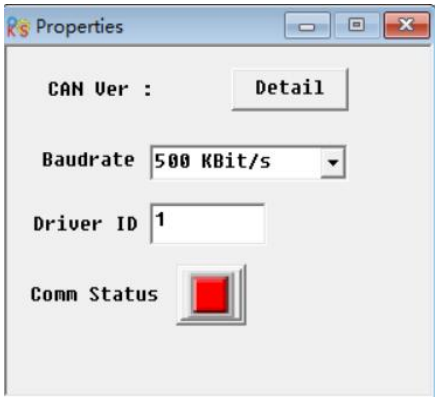
Click CAN if you use PECAN and press Next key.



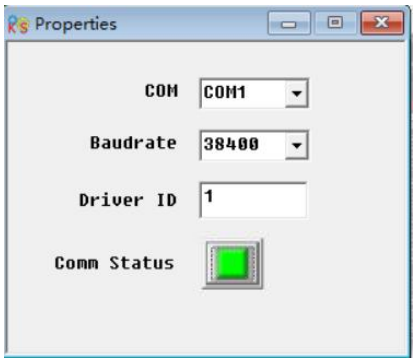
4． Then it will show us a dialog as following to set up COM, baud rate, ID and click the Red communication button  。



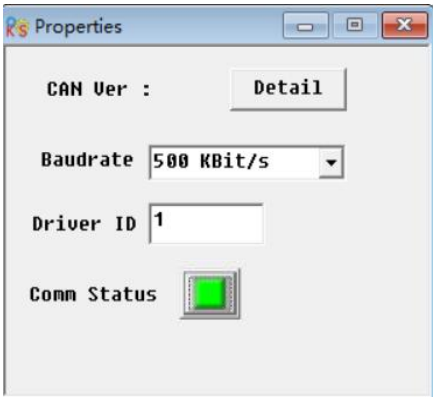
if you use PECAN, set CAN baud rate, drive ID number (follows are default parameters) and click the Communication status button.



5.Check whether there is Comm Status: Open COM1 38400 on the right-bottom and the light of Comm Status has turned on in green. If yes, it is connecting successful.

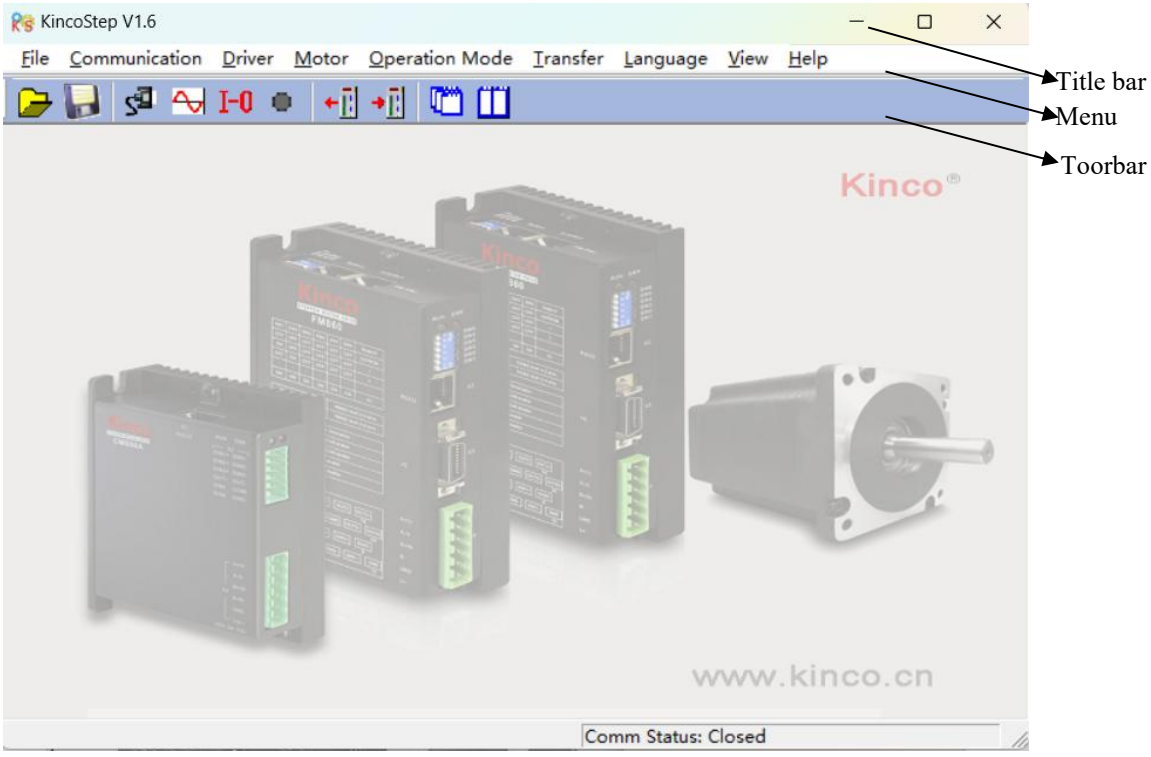


For CAN, it will show us Comm Status: Open 500K Bit/S, and the light is also in green if accessed successful.



4.3Menu

Open Kincostep PC software, the use window is as follow:



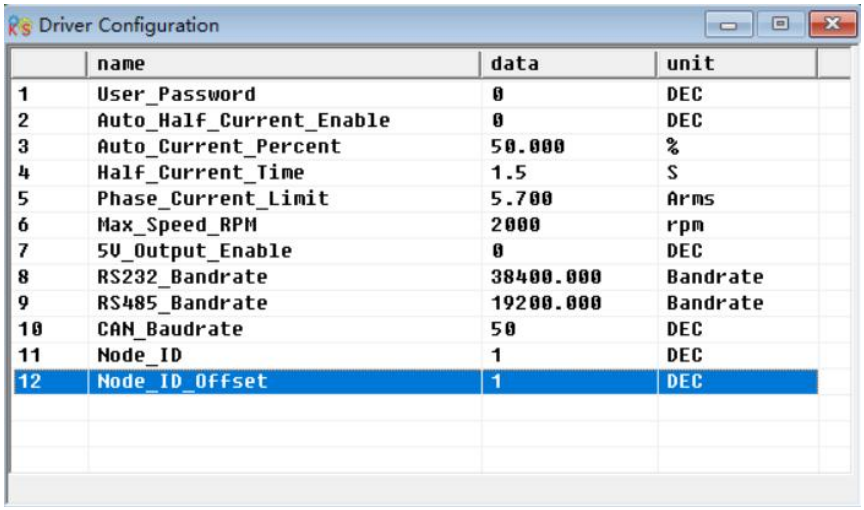
As shown in the figure , the top is the menu bar, mouse click each menu has the corresponding function to select the submenu, the toolbar icon is the shortcut of common functions, The function of each menu are described in the following table.

Table 4-1 Menu bar describtion

Name	Function description
File	New tasks, Open old file, Save
Communication	Setting up the communication way and parameter from driver to PC.
Driver	To control the drive, see 4.4
Motor	To set up the motor parameter.
Operation mode	Operation mode parameter configuration
Transfer	Read/write drive parameters, see the appendix example for details
Language	Set the language

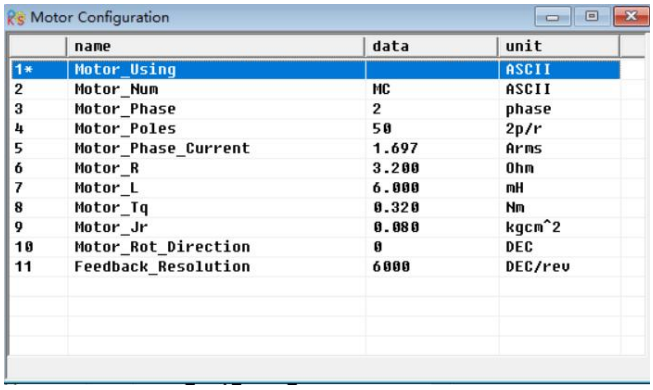
4.4 Driver control

4.4.1 Driver configuration



In this menu, you can perform the basic configuration of the drive. Such as user password, RS232 communication and other common parameters. You can right mouse, see the help prompt.

4.4.2 Motor configuration



motor type select

ASCII....HEX....TYPE

"00".....3030...no motor select
"MC".....434d...self detect motor parameter
"XX".....5858...customer write parameter
"A1".....3141...2S42Q-03848
"A2".....3241...2S42Q-02940
"A3".....3341...2S42Q-0240
"A4".....3441...2S42Q-0348(serial connect)
"A5".....3541...2S42Q-0348(parallel connect)
"B1".....3142...2S56Q-030B5
"B2".....3242...2S56Q-02976
"B3".....3342...2S56Q-02054
"B4".....3442...2S56Q-02741
"B5".....3542...2S57Q-0541(serial connect)
"B6".....3642...2S57Q-0541(parallel connect)
"B7".....3742...2S57Q-0956(serial connect)
"B8".....3842...2S57Q-0956(parallel connect)
"B9".....3942...2S57Q-1376(serial connect)
"BA".....4142...2S57Q-1376(parallel connect)
"BB".....4242...2S57Q-2280(serial connect)
"BC".....4342...2S57Q-2280(parallel connect)
"BD".....4442...2S57Q-25B2(serial connect)
"BE".....4542...2S57Q-25B2(parallel connect)

"C1".....3143...2S86Q-069B8
"C2".....3243...2S86Q-05180
"C3".....3343...2S86Q-03865
"C4".....3443...2S86Q-051F6
"C5".....3543...2S86Q-030B8...2S86Q-85B8
"C6".....3643...2S86Q-03080...2S86Q-4580
"C7".....3743...2S86Q-01865...2S86Q-3465
"D1".....3144...2S110Q-054K1
"D2".....3244...2S110Q-047F0
"D3".....3344...2S110Q-03999
"E1".....3145...2S130Y-063R8
"E2".....3245...2S130Y-039M0
"F1".....3146...3S57Q-04079
"F2".....3246...3S57Q-04056
"F3".....3346...3S57Q-04042
"G1".....3147...3S85Q-04097
"G2".....3247...3S85Q-04067
"G3".....3347...3S85Q-040F7

There are 3 kinds of methods for user to set up motor parameters.

1.Automatic detection of motor parameters（The driver is factory default and the motor model is MC）

Drive factory default setting：

The motor model is MC, the motor phase number is 2, and the motor phase current is the default value.

Driver setting is defaulted as 2 phase, the LED will show error if you use 3 phase motor:the RUN indicator fast blink, and the ERR indicator is steady on. (If the user uses the debugging software, the real-time error menu will display the internal error of the drive and find the motor error.) The user needs to change the number of phases of the motor to 3, store the motor parameters, and the drive can be used normally after restart.

2.Select motor type

User also can select the right motor type if you don’ t like to use automatic detection of motor parameters, then the parameters will be listed into the dialog by automation.

3.User defined(Motor type as XX)

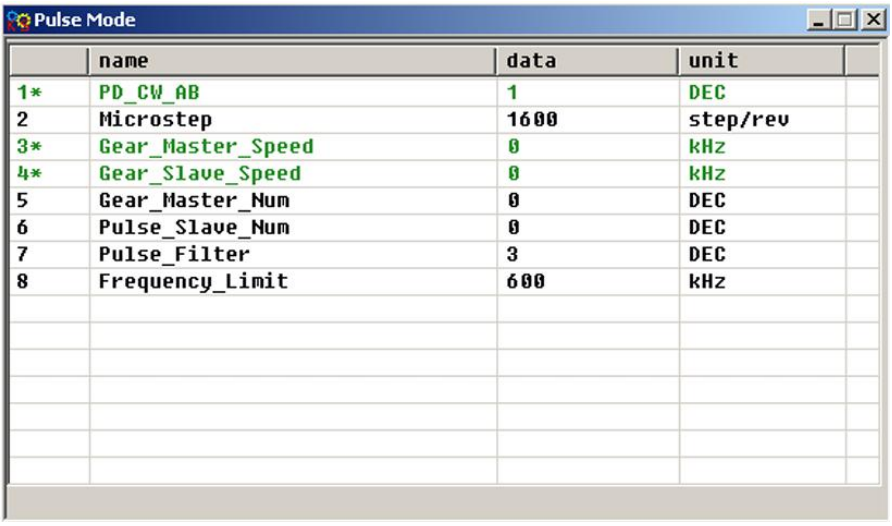
If you selected the motor which are not in the list, please set motor type as XX, then set the motor parameters by manual.

4.4.3 Current setting

For the factory driver, the default motor phase current is 3Arms for FM860 and 1Arms for FM880. Users can set the motor phase current according to the application requirements. After the user changes the current Settings, the motor parameters need to be stored, and the drive can be used normally after restarting.

If the user needs the driver to output more current, please contact the manufacturer.

4.4.4 Microstep setting



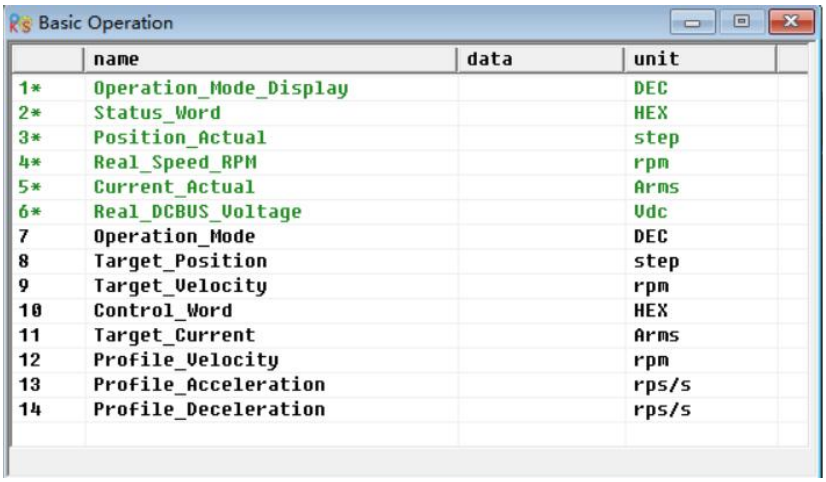
	name	data	unit
1*	PD_CW_AB	1	DEC
2	Microstep	1600	step/rev
3*	Gear_Master_Speed	0	kHz
4*	Gear_Slave_Speed	0	kHz
5	Gear_Master_Num	0	DEC
6	Pulse_Slave_Num	0	DEC
7	Pulse_Filter	3	DEC
8	Frequency_Limit	600	kHz

In pulse mode (-4 mode)，The microstep settings : Microstep = the number of pulses per revolution/（360° / Step angle）

Note:

The number of pulses per revolution must lager than or equal to 200 for 2 phase motor setting. As for 3 phase motor, the number of pulses per revolution must larger than or equal to 300.

4.4.5 Basic operation



	name	data	unit
1*	Operation_Mode_Display		DEC
2*	Status_Word		HEX
3*	Position_Actual		step
4*	Real_Speed_RPM		rpm
5*	Current_Actual		Arms
6*	Real_DCBUS_Voltage		Vdc
7	Operation_Mode		DEC
8	Target_Position		step
9	Target_Velocity		rpm
10	Control_Word		HEX
11	Target_Current		Arms
12	Profile_Velocity		rpm
13	Profile_Acceleration		rps/s
14	Profile_Deceleration		rps/s

In this menu, you can perform the basic control operations of the drive. For details about the operation objects of each mode, see Chapter 6 Mode Operation.In addition, in this menu you can monitor the main running data of the drive.

Example 4-1: Set a user password with Kincostep PC software

1. In the window of Drive Configuration, write the password as 1234 (the password range is from 1 to 65535),press Enter to confirm.

2.Click "Drive" - "Initialization/save" - "save control parameters" to save; If you modify the motor parameters, you need to "save motor parameters" and then click "Reboot drive ".

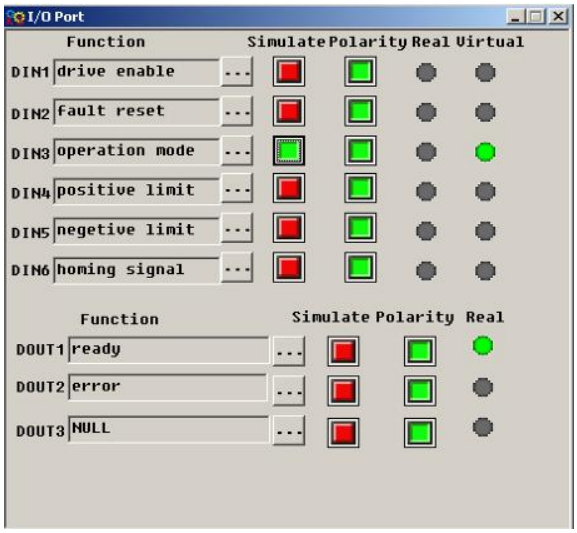
3. The password is effective after Reboot drive. So you can’ t do anything for the drive, have to return to Drive Configuration dialog and write the password first. 4. If you want to cancel the password, need to write the old first then change it to 0 and click Reboot drive,save and restart. So be sure to remember the password you set.

Example 4-2: Speed mode operation of the stepper drive using Kincostep PC software (manual operation)

1.Follow example 4-3 and open I/O port. Set DIN1 as Drive enable, DIN2 as Fault reset, DIN3 as Operation mode.

2.Refer to chapter 6.2 profile velocity mode to set up the parameters,And change the value of speed as negative if you want to run it reverse.

4.4.6 I/O operation

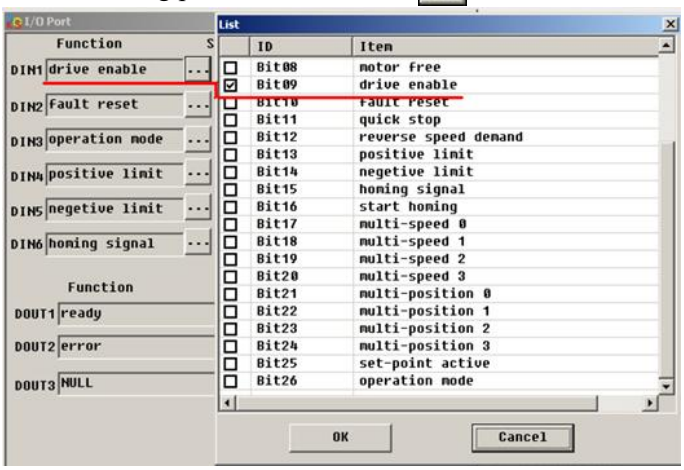


In this menu, you can define the drive I/O function and set the polarity. For details, refer to the example.It can also monitor the actual I/O input and output status, analog simulation and other functions. The figure above shows the default I/O Setting.

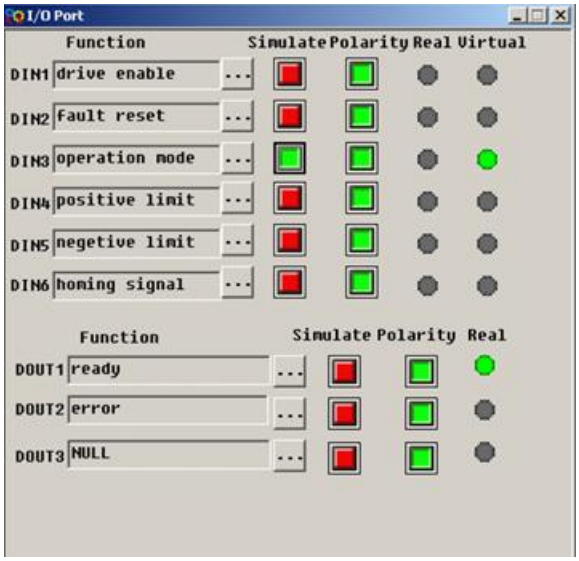
Example 4-3: Setting up step I/O using Kincostep PC software

Set DIN1 as Drive enable, DIN2 as fault reset, DIN3 as operation mode, DIN4 as positive limit, DIN5 as negative limit, DIN6 as homing signal.

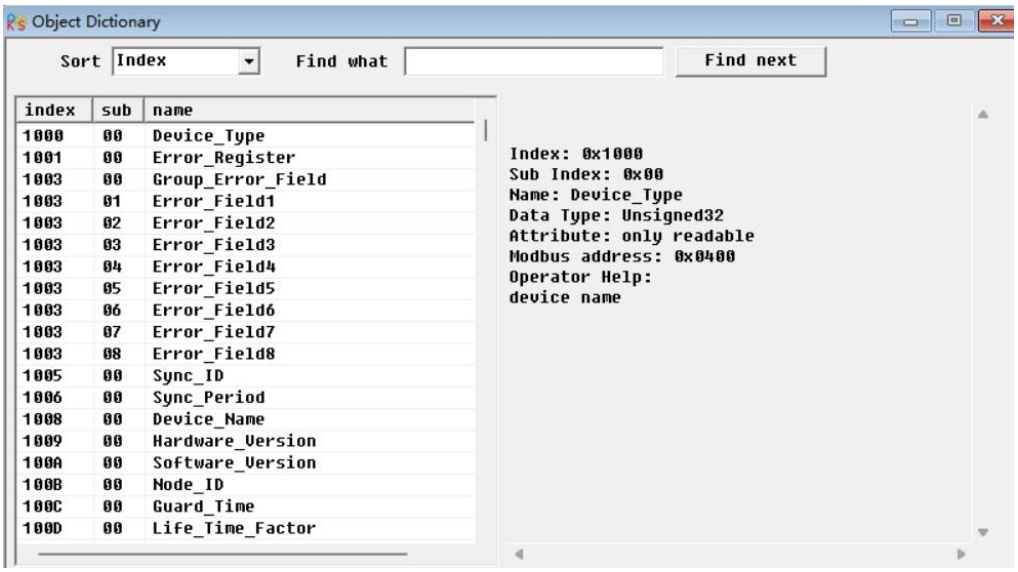
1. As following picture, click button ... to show us a list and select the item we want and click OK to finish it.



2. Follow step1 to operate the I/O port and finish as below.



4.4.7 Object dictionary



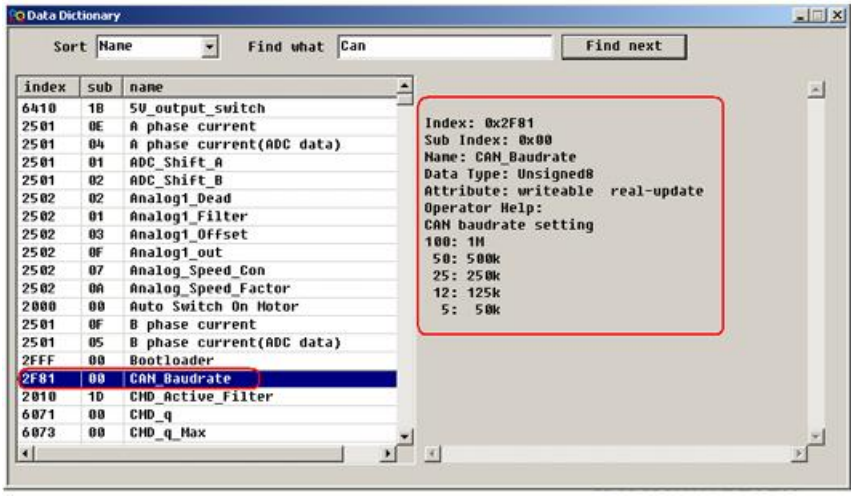
You can get the index and detail information of all FM Series in the object dictionary. As shown in following picture, left side includes information of index, sub-index and naming of CANopen.Right side is the details of the parameters.

Example 4-4: Using Kincostep PC software to add an object for setting

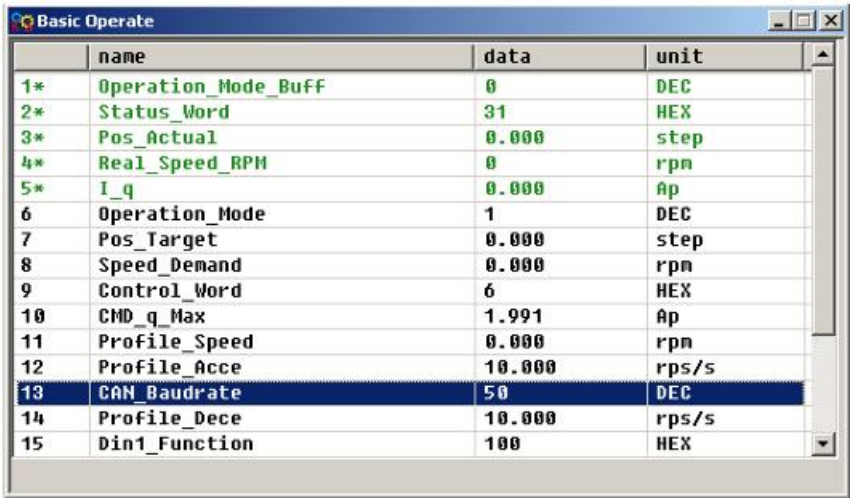
Let’s learn how to add an index into Basic operation list and setting it. Now we add CANopen Baud rate.

1.Right click in the Basic operation list and choose Add, then it will popup a window of Data Dictionary such as the picture.

2. Input CAN and click Find next, it will jump to 2F81, and to know more information on the right side.



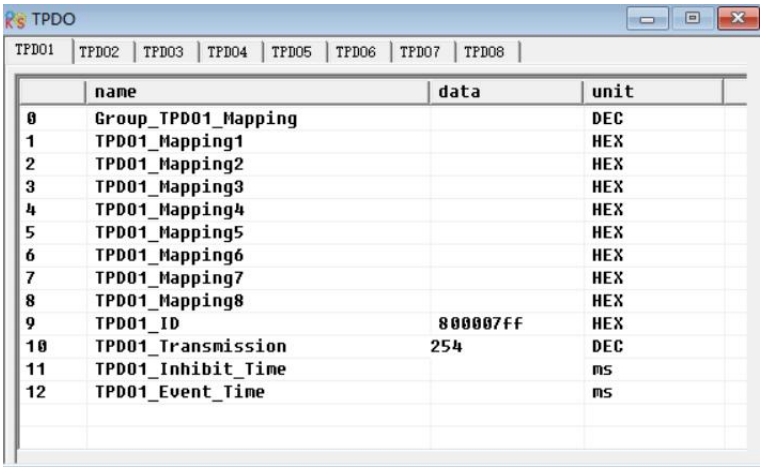
3. Double click the index 2F81, the index will be added into the menu list successful as below.

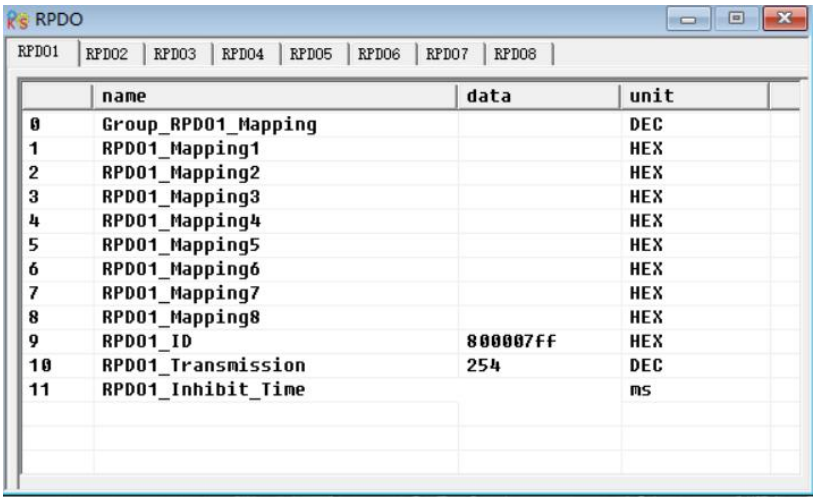


4. How to cancel the object in the menu list? Find it and click your PC mouse on the right then click del to finish it. If want to know more information about a index, find it and click your PC mouse then click help.

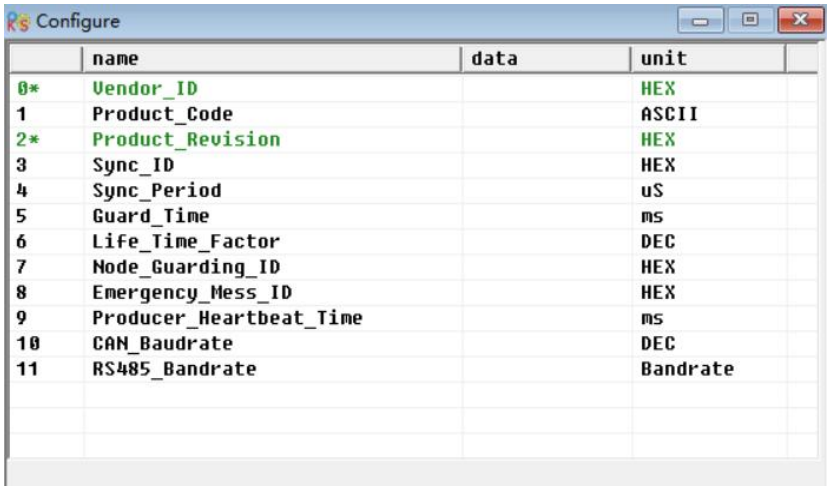
4.4.8 ECAN setting

As for this menu list,it can set the communication parameters of CANopen. please refer to chapter 7.3.





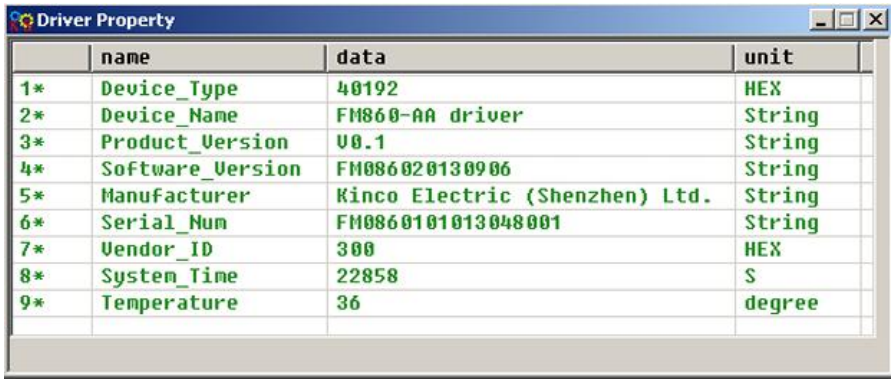
	name	data	unit
0	Group_RPDO1_Mapping		DEC
1	RPDO1_Mapping1		HEX
2	RPDO1_Mapping2		HEX
3	RPDO1_Mapping3		HEX
4	RPDO1_Mapping4		HEX
5	RPDO1_Mapping5		HEX
6	RPDO1_Mapping6		HEX
7	RPDO1_Mapping7		HEX
8	RPDO1_Mapping8		HEX
9	RPDO1_ID	800007ff	HEX
10	RPDO1_Transmission	254	DEC
11	RPDO1_Inhibit_Time		ms



	name	data	unit
0*	Vendor_ID		HEX
1	Product_Code		ASCII
2*	Product_Revision		HEX
3	Sync_ID		HEX
4	Sync_Period		uS
5	Guard_Time		ms
6	Life_Time_Factor		DEC
7	Node_Guarding_ID		HEX
8	Emergency_Mess_ID		HEX
9	Producer_Heartbeat_Time		ms
10	CAN_Baudrate		DEC
11	RS485_Bandrate		Bandrate

4.4.9 Drive property

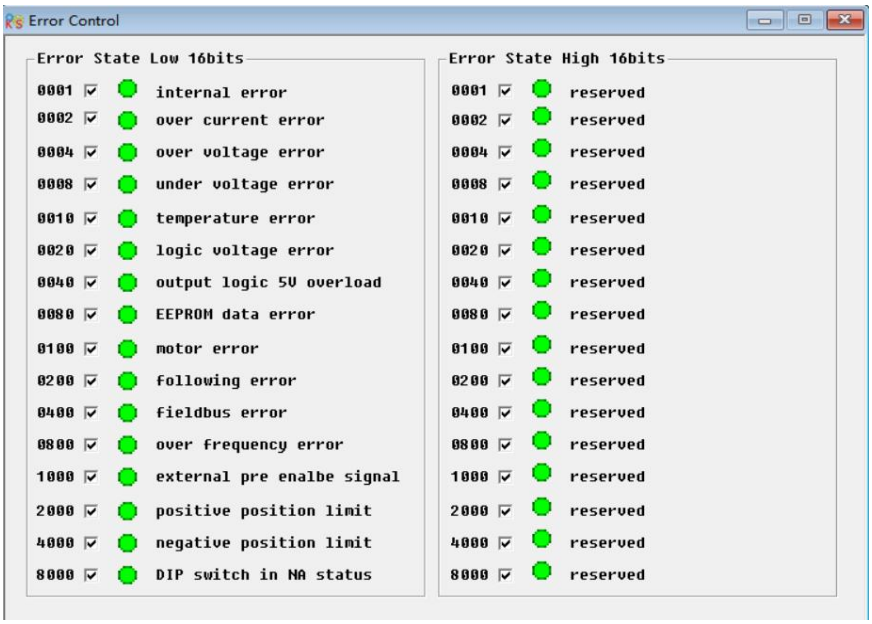
We can know the information of drive such as software version, S/N and so on in the window.



	name	data	unit
1*	Device_Type	40192	HEX
2*	Device_Name	FM860-AA driver	String
3*	Product_Version	V0.1	String
4*	Software_Version	FM086020130906	String
5*	Manufacturer	Kinco Electric (Shenzhen) Ltd.	String
6*	Serial_Num	FM0860101013048001	String
7*	Vendor_ID	300	HEX
8*	System_Time	22858	S
9*	Temperature	36	degree

4.4.10 Error control

This menu will show the error status if there is any error. In the menu, the code of 16 hex is the error code, and the box is used to select whether the alarm is shielded or not. The light will turn red if there is error,more details please refer to chapter 8. Note: Untick the box to shield the error if you need,but some of errors can’t shield.

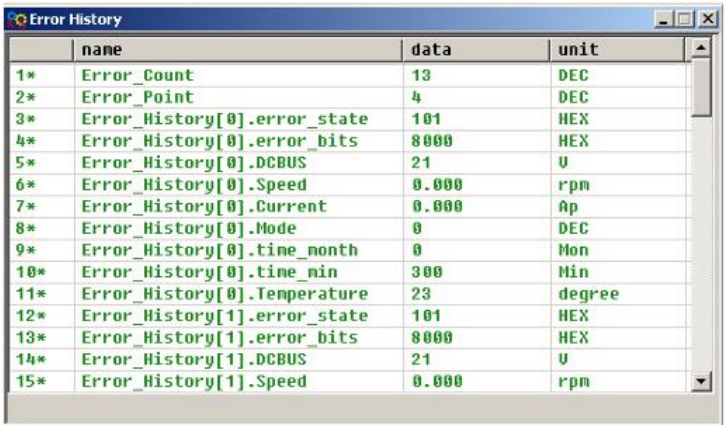


4.4.11 Error History

Stepper driver provides 8 groups of historical alarm information, the user can query the alarm code when the alarm occurs, voltage, current, temperature, speed, working mode, driver accumulated working time and other information, better facilitate user equipment maintenance.

The error history pointer is the most recent error address, the value N indicates that the current alarm is "error N", the last record is N - 1, the next record is N + 1, 8 groups of error records loop overwrite.

For example, the error history pointer =4 indicates that the current alarm is "Error 4". The next alarm location is "Error 5".



4.4.12 Initialize/Save

This menu is used to save drive parameters, initialize, restart the drive, etc. It should be noted that the motor parameters and other parameters are saved separately!

If the button is gray, the storage operation is in progress.

Chapter 5 I/O operation

The stepper driver has 6 channel digital input and 3 channel digital output. Digital input and output can be freely configured according to application need.

5.1 Digital signal input

5.1.1 Digital signal input polarity control

Now, make S1 as open or close to change the polarity of IO via Kincostep PC software. It means the polarity is open if the light is in green, the DIN1 is un-working.

Example 5-1: Digital signal input DIN1 polarity setting

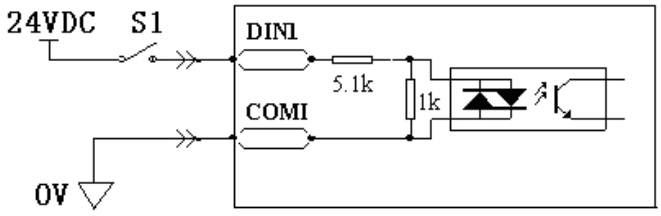


Figure 5-1 Digital signal input DIN1 polarity setting diagram

After the debugging software is connected to the stepper driver, the IO port page is opened. By default, the polarity indicator is green, indicating that the input port is normally open at this time, that is, the status of DIN1 is invalid. Click DIN4 and DIN5 to change the polarity indicator to red, then it means DIN4,DIN5 becomes normally closed, and the state is effective.

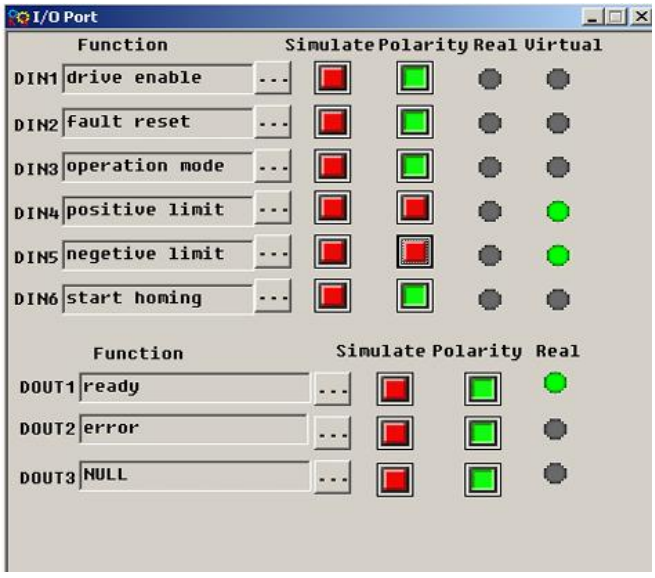


Figure 5-2 Digital I/O View

5.1.2 Digital signal input function

Table 5-1 Digital signal input function and default value

Name	Function	Default
DIN1_Function	Pulse input Direction input Clockwise pulse input Counterclockwise pulse input Quadrature encoder phase A input Quadrature encode phase B input Quadrature encoder index input Motor free Drive enable Fault reset Quick stop Reverse speed demand Positive limit Negative limit Homing signal Start homing Multi-speed 0 Multi-speed 1 Multi-speed 2 Multi-speed 3 Multi-position 0 Multi-position 1 Multi-position 2 Multi-position 3 Set-point active Operation mode Pre enable Multi-gear 0 Multi-gear 1 Multi-gear 2	0x1（Pulse）
DIN2_Function		0x2（Direction）
DIN3_Function		0x8（Motor free）
DIN4_Function		0(NULL)
DIN5_Function		0(NULL)
DIN6_Function		0(NULL)
DIN7_Function		0(NULL)
DIN8_Function		0(NULL)

Table 5-2 Digital signal input function description

Function	Description	
Pulse input	In the mode as pulse + direction, the pulse signal input	Note: Only DIN1 and DIN2 can support these functions.
Direction input	In the mode as pulse + direction, the direction signal input	
Clockwise pulse input	In double-pulse mode, the CW pulse input as clockwise	
Counterclockwise pulse input	In double-pulse mode, the CW pulse input as counterclockwise	
Quadrature encoder phase A input	Incremental encoder mode，A phase signal input	
Quadrature encode phase B input	Incremental encoder mode，B phase signal input	
Quadrature encoder index input	Incremental encoder mode，Z phase signal input	
Motor free	Motor off, and motor axis is loose in avail status	
Drive enable	Drive enable, control word in avail status can be configured via object 202033 to input control word	
Fault reset	Clear error alarm, the rising edge of the signal is detected	
Quick stop	When the signal is valid, the motor shaft releases. After the signal is removed, the drive requires re-enabling.	
Reverse speed demand	To reverse the target speed in the speed mode.	
Positive limit	Indicates the limit of forward running of motors (normally closed by default. the drive regards position positive limits as valid, and polarity can be modified to adjust to normally open switches.	

Negative limit	Indicates the limit of inverted running of motors (normally closed contact by default. the drive regards position negative limits as valid, and polarity can be modified to adjust to normally open switches.)
Homing signal	Homing switch signal
Start homing	Star the command of homing, the rising edge of the signal is detected.
Multi-speed 0	Used to control multi-speed switching, the four bits of signal input 3~0 are combined into a hexadecimal number. For example, if the control 2 is 1 and the others are 0, the current hexadecimal number is 0x4, which corresponds to the speed of the fourth paragraph.
Multi-speed 1	
Multi-speed 2	
Multi-speed 3	
Multi-position 0	Used to control multi - position switching. the four bits of signal input 3~0 are combined into a hexadecimal number.For example, if 2 and 3 are set to 1 and the others are set to 0, the current hexadecimal number is 0xC, which corresponds to the position of the 12th paragraph.
Multi-position 1	
Multi-position 2	
Multi-position 3	
Set-point active	Activate the target position or target position segment, the signal rising edge is effective.
Operation mode	To switch 2 kind of mode. The mode of valid signal and invalid can be defined as flexible. Need to via object 202031 input operation mode select 0 and the object 202032 to select 1 to configure.
Pre enable	It is used for safety control, indicating that the controller is ready and the driver can be enabled. An alarm will also appear if the input signal is invalid, or if the input signal becomes invalid while the motor is running.
Multi-gear 0	Control multi-electronic gear switching (electronic gear molecule and denominator determined by electronic gear group input), signal input 2 ~ 0 of the 3 bits combined into a hexadecimal number, for example: control 2 is 1, the other is 0, then the current hexadecimal number is 0x4, corresponding to the fourth segment position.。
Multi-gear 1	
Multi-gear 2	

Example 5-2: Drive enable setting

In this example, kicnostep software is used to set the I/O function. After the software is connected to the drive, open the I/O , click the button[...] marked in red on the right of DIN1, select the function of "Drive Enable" from the list, and then confirm. Unneeded functions are removed from the front checkmark.

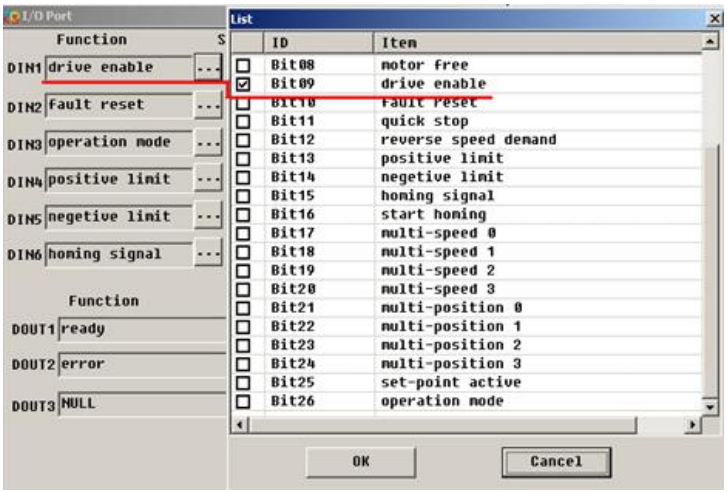


Figure 5-3 I/O function setting

Example 5-3 Pulse input mode setting

As for the pulse input function, it is supported both for DIN1 and DIN2 support only. If DIN1 is pulse input, DIN2 will be as direction input if DIN1 have the signal as pulse input, and DIN2 can not be as NULL. Otherwise it will not work.

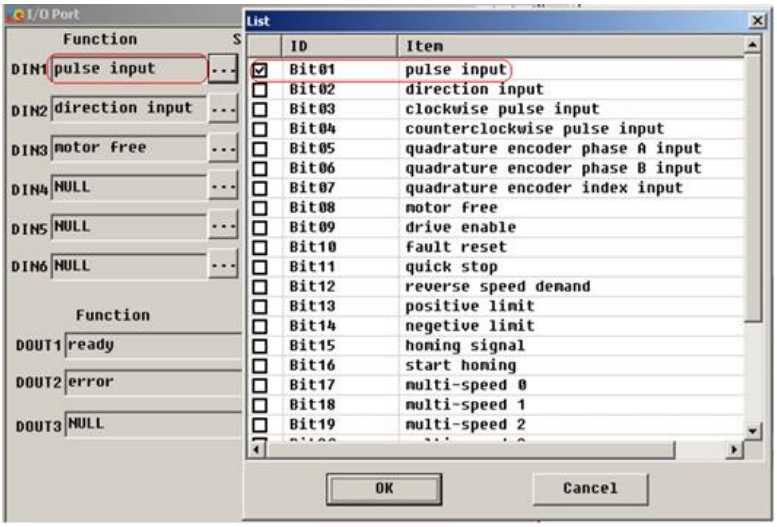


Figure 5-4 Pulse input mode setting

5.1.3 Digital input port wiring

1.NPN wiring diagram（Support the effective controller of low level output）

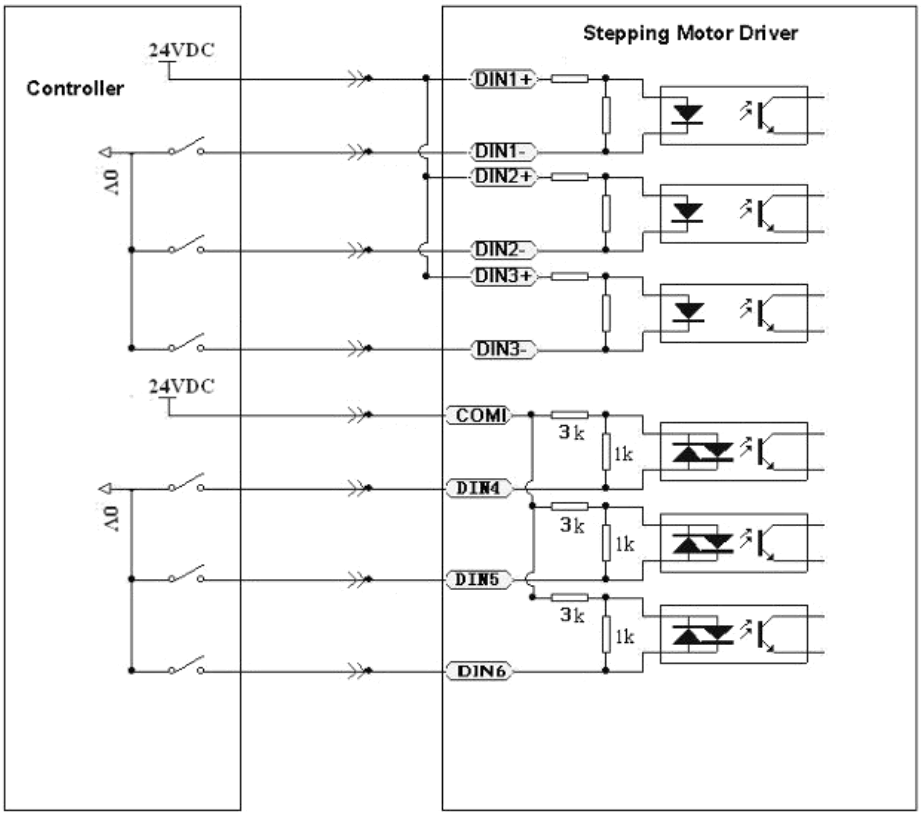


Figure 5-5 NPN wiring

2. PNP wiring diagram（Support the effective controller of high level output）

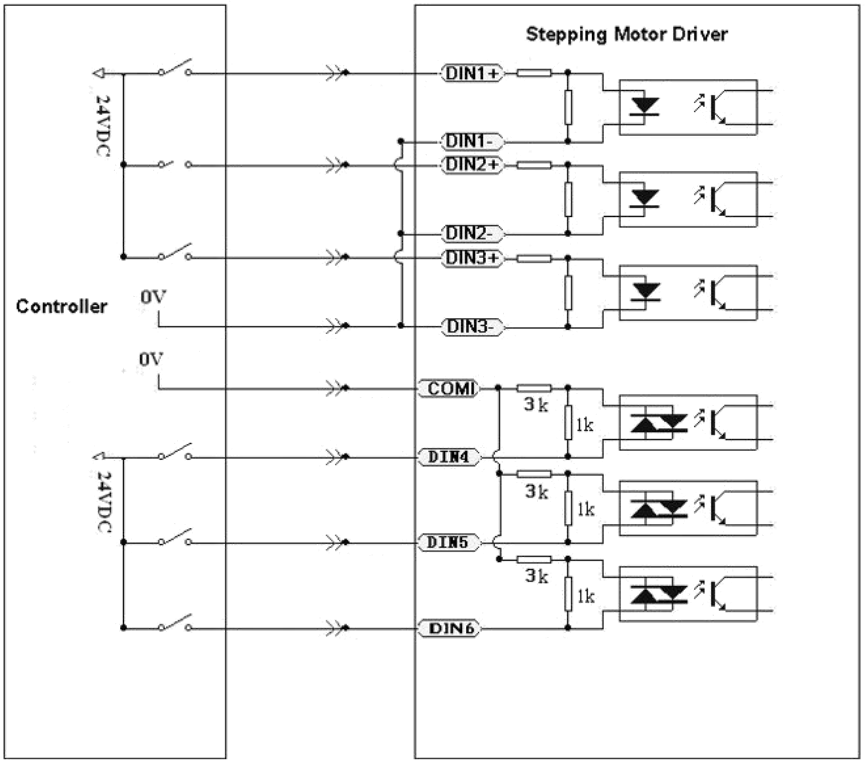


Figure 5-6 PNP wiring

5.2 Digital signal output

5.2.1Digital signal polarity control

The polarity of digital output is defaulted as open. About how to change the polarity please refer to example 5-1.

5.2.2 Digital signal output address and function

Table 5-3 Digital signal output addresses and default function

Name	Function	Default
DOUT1_Function	Ready Error Position reached Zero velocity Motor brake Velocity reached Index signal Max velocity limit PWM ON Position limiting Reference found Index signal multiplier Absorb DC_bus voltage	0x1 (Ready)
DOUT2_Function		0x2 （Error）
DOUT3_Function		0 （NULL）

Table 5-4 Digital signal output function description

Function	Description
Ready	Drive is on operation mode
Error	Alarm status
Position reached	In position mode, target-position data is no change from position to time window. And while the position error in the position to the window.
Zero velocity	Motor velocity is zero after motor enable
Reserved	Reserved
Velocity reached	Motor speed reached the target in the control of speed mode
Reserved	Reserved
Reserved	Reserved
Motor brake	Drive enable motor
Position limiting	Motor is in limit
Reference found	End of homing
Index signal multiplier	Index signal seems to be homing, and set output periodic pulse.
Absorb DC_bus voltage	To absorb the main voltage, used as 3 phase driver and connecting resistor between V+ and B-

Example 5-5: Drive ready setup

In this example, the kicnostep software is used to set the I/O function. After the software is connected to the stepper drive, open the I/O page, click the button [...] marked in red on the right side of DOUT1, select the "Ready" function from the list, and then confirm. Unneeded functions are removed from the front checkmark. The following figure shows the setting method.

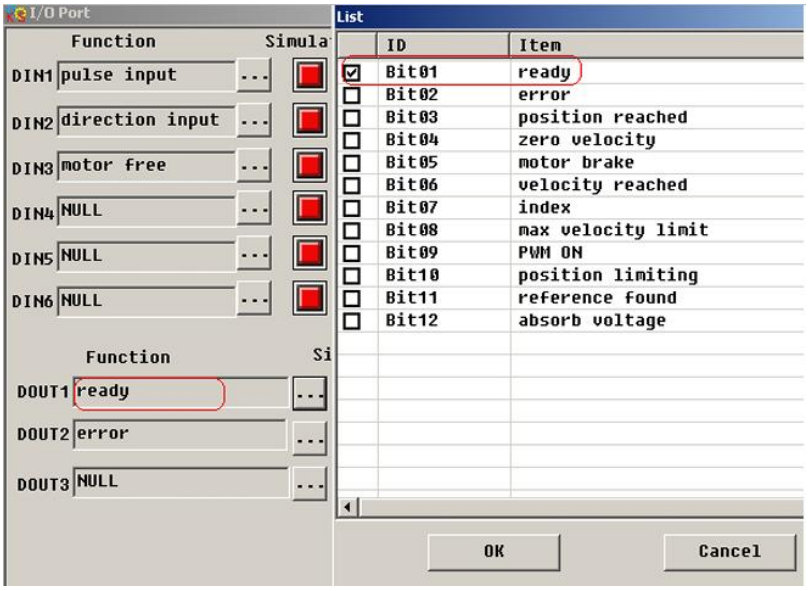


Figure 5-7 Ready setting

5.2.3 Digital output wiring

1.Digital output wiring diagram

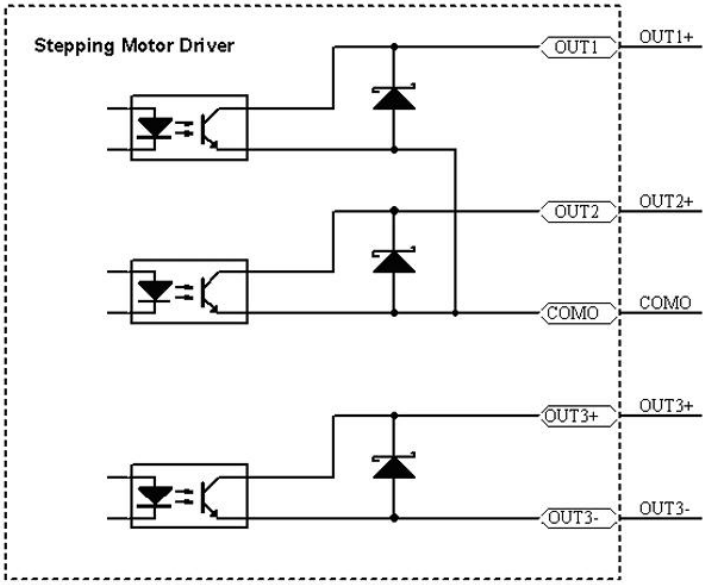


Figure 5-8 Digital output internal circuit diagram

2.NPN wiring diagram（OUT1-OUT3 support NPN and the effective controller of low level input）

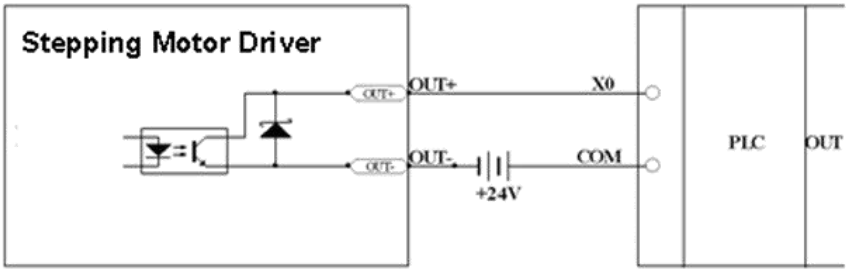


Figure 5-9 NPN wiring

3. PNP wiring diagram（only OUT3 support PNP and the effective controller of high level input）

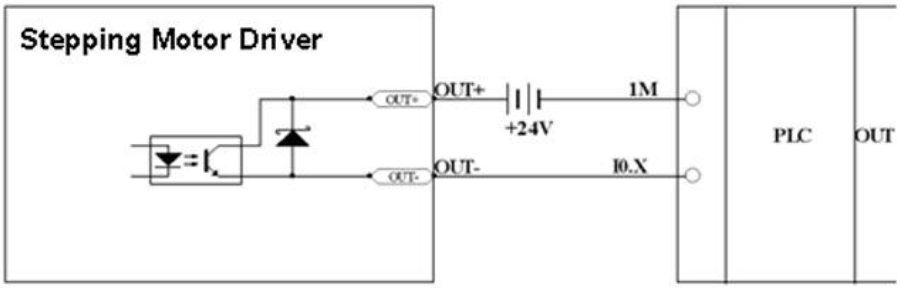


Figure 5-10 PNP wiring

4. Follow the diagram as below to connect relay to the digital output.

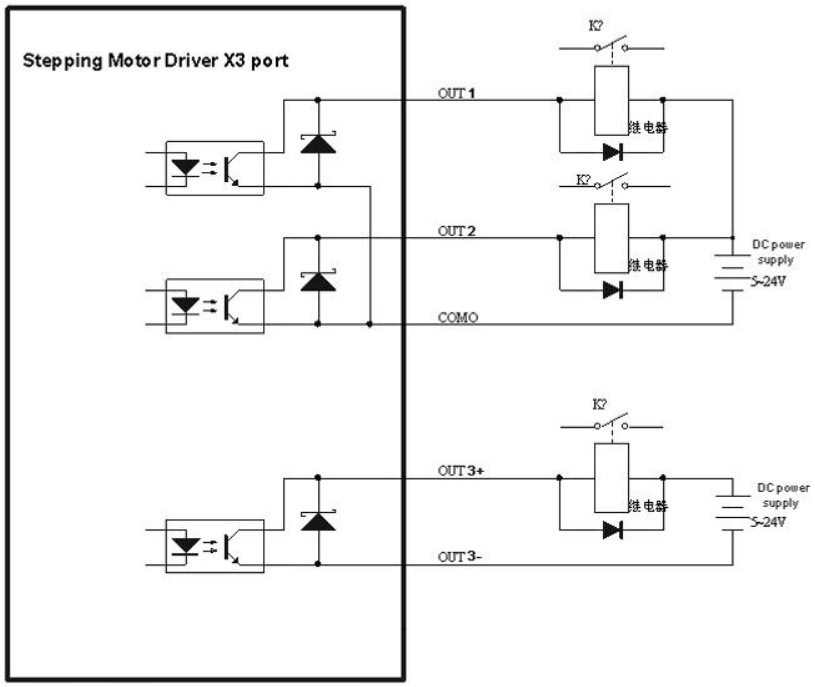


Figure 5-11 Digital output relay diagram (anti-parallel diode)

5.3 Analog signal input

5.3.1AIN1 common mode voltage input wiring diagram

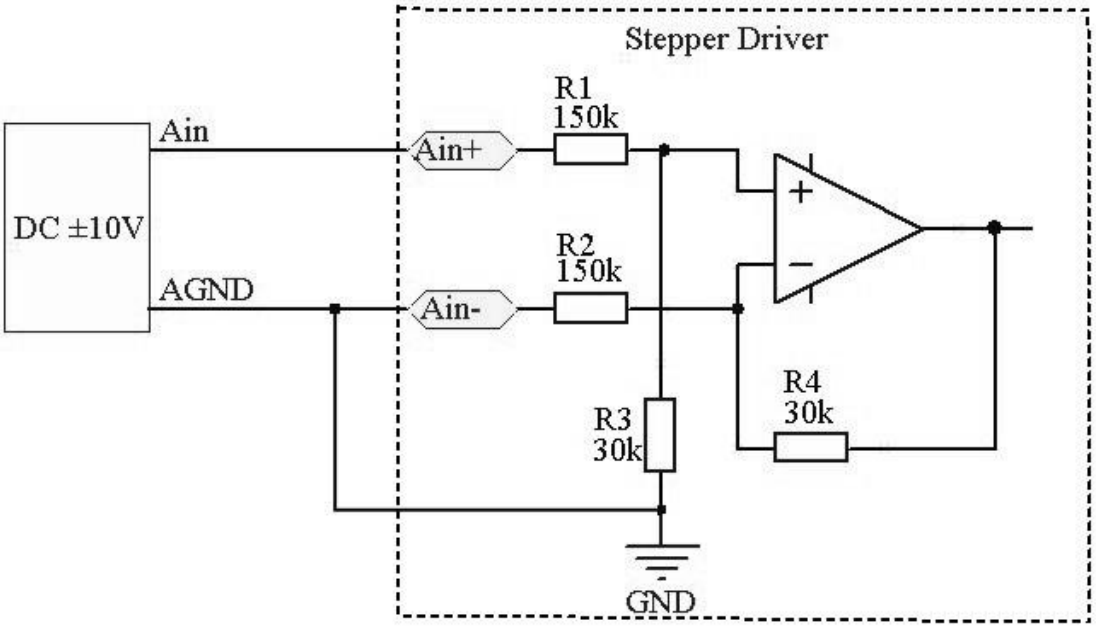


Figure 5-12 Common mode input voltage wiring diagram

5.3.2 AIN1 differential mode voltage input wiring diagram

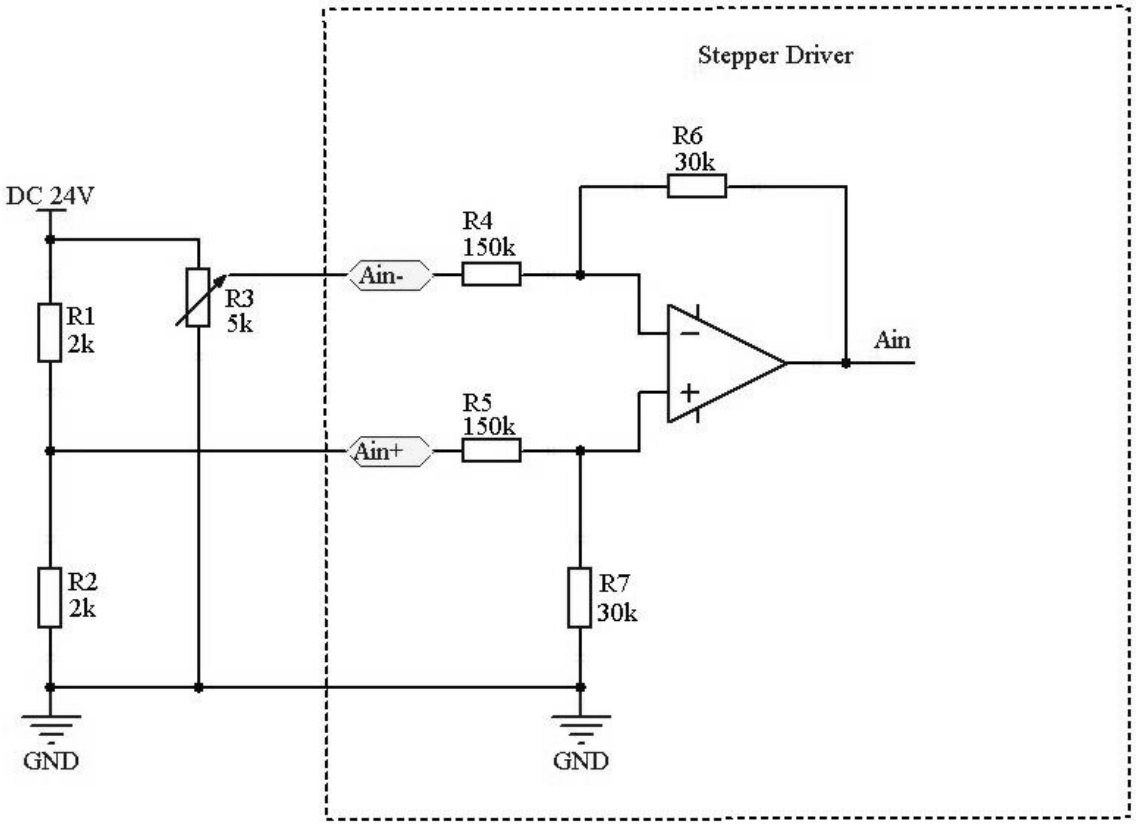


Figure 5-13 Differential mode input voltage wiring diagram

Chapter 6 Operating Mode

6.1 Position mode（1）

Now make an example to explain this mode. Coordinate system as shown below, the red arrow is defined as the current position 440, if we make the absolute position movement, when the target position is set to 700, the motor will run to coordinate position of 700 ; if we make a relative position movement, when the target position is set to 700, the motor will run to the coordinate position of 1140.

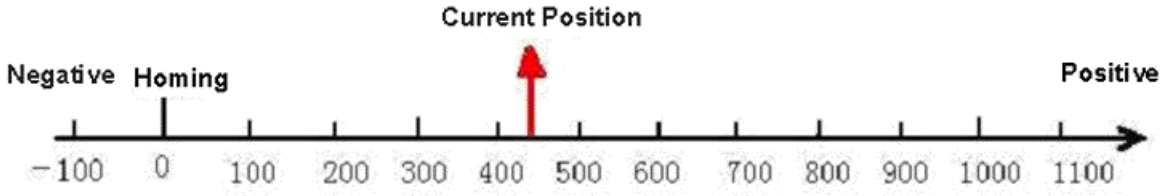


Figure 6-1 Absolute/ Relative position

Table 6-1 Object in position mode 1 definition

Name	CANopen address	Modbus address	Value	Description
Operation_Mode	60600008	0x3500	1	To set up operation mode as absolute position mode and relative position.
Profile_Speed	60810020	0x4A00	User Setting	Trapezoidal curve speed
Profile_Acce	60830020	0x4B00	User Setting	Acceleration
Profile_Dece	60840020	0x4C00	User Setting	Deceleration
Pos_Target	607A0020	0x4000	User Setting	Locate the target position in position mode
Control Word	60400010	0x3100	Write F first then 1F	When the absolute position is moving, the motor starts to move after the last target position is reached
			Write 2F first then 3F	When the absolute position is moved, the motor starts to move immediately
			Write 4F first then 5F	When the relative position is moving, the motor starts to move after the last target position is reached
			Write 6F first then 7F	Relative position is running then Motor begin run.
			103F	According to the change of target position, begin absolute position moving immediately.

Please refer **Appendix VII** for more information.

For position mode control through communication, refer to the communication cases in the appendix

Position mode control can also be achieved via external I/0 Setting

For example:

1）In absolute position mode, run from position 440 to position 700. Operation process: A) "Operation mode" is set to 1, "Trapezoidal speed" is set to 100rpm, "trapezoidal acceleration" and "trapezoidal deceleration" are set to 10rps/s, "target position" is set to 700, "control word" is changed from 6 to 2F, so that the motor has current lock shaft (skip this step if the shaft has been locked), and then write 3F, the motor will immediately move to the target

position. Stop at 700. If you still need to run to the position 1140, you need to set the "target position" to 1140, "control word" first write 2F, then write 3F, the motor immediately to the target position, stop at 1140 position.

2）In relative position mode, run from position 440 to position 700. Operation process: A) "Operation mode" is set to 1, "Trapezoidal speed" is set to 100rpm, "trapezoidal acceleration" and "trapezoidal deceleration" are set to 10rps/s, "target position" is set to 260, "control word" is changed from 6 to 6F, so that the motor has current lock shaft (skip this step if the shaft has been locked), and then write 7F, the motor will immediately move to the target position and stop at 700. If you still need to run to the position 1140, you need to set the "target position" to 440, "control word" first write 6F, then write 7F, the motor immediately to the target position, stop at 1140 position.

6.2 Velocity mode（3）

As for the mode 3, can realize the speed control of the motor.The running curve includes acceleration, uniform speed and deceleration. The time of accelerating can figure out by initial velocity, uniform speed and accelerated speed .

Vt＝Vo＋at　　Vt－Uniform speed

Vo－Initial velocity

a –Accelerated speed or depend reduction

t - Acceleration time

S＝Vot + (1/2) at2　　S－Accelerated segment displacement

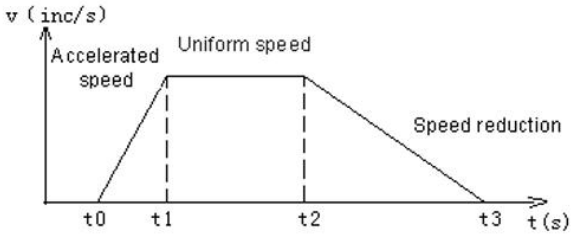


Figure 6-2 Speed and time curve in mode 3

Table 6-2 Object in position mode 3 definition

Name	CANopen address	Modbus address	Value	Description
Operation_Mode	60600008	0x3500	3	Set speed mode
Profile_Speed	60FF0020	0x6F00	User Setting	Target velocity
Profile_Acce	60830020	0x4B00	User Setting	Acceleration
Profile_Dece	60840020	0x4C00	User Setting	Deceleration
Control Word	60400010	0x3100	Write 6 first then F	Lock the motor shaft, write such parameters and run it as the parameter request. And motor will re-run after you write new speed value

Please refer **Appendix VII** for more information.

For velocity mode control through communication, refer to the communication cases in the appendix

Velocity mode control can also be achieved via external I/0 Setting

6.3Periodic synchronization position control mode （8）

In this mode, the motor's motion is directly controlled by the bus data from the driver's X1 port. The objects to be defined in this mode are:

Name	CANopen address	Modbus address	Value	Description
Operation_Mode	60600008	0x3500	8	Set the operation mode to synchronous position control mode
Target position	607A0020	0x4000	User Setting	Target position
IP_Time_Period	60C20108	--	User Setting	The communication period for synchronizing data
IP_Time_Index	60C20208	--	User Setting	
Control Word	60400010	0x3100	F	Lock motor shaft, motor control movement according to synchronous data

6.4 Pulse mode（-4）

The motor running is monitored by X3 port of Drive in the mode -4. And the index is as below.

Table 6-3 Objects in Mode -4

Name	CANopen address	Modbus address	Value	Description
Pulse mode	25080308	0x1930	1	Pulse mode, configured by IO ports DIN1 and DIN2
Operation_Mode	60600008	0x3500	-4	Set up operation mode to be pulse control mode
Microstep	64101810	0x7180	User Setting	The number of pulses per motor revolution
Control Word	60400010	0x3100	F	Lock motor, motor begin to move.

Note: Change the pulse mode requires redefining IO ports DIN1 and DIN2. For details, see Example 5-3

Please refer **Appendix VII** for more information.

6.5 Homing Control Mode（6）

When performing absolute position, you must define the homing.

As following picture shown that have to define the homing as(0,0) before locate (X,Y）=(100mm,200mm).

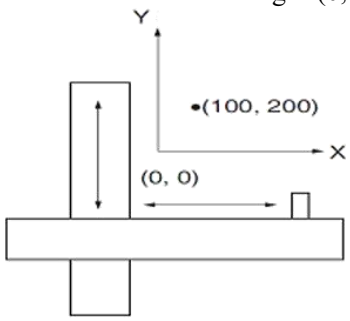


Figure 6-3 Homing control mode

Table 6-4 Objects in Mode 6

Name	CANopen address	Modbus address	Value	Description
Operation_Mode	60600008	0x3500	6	Operation mode of drive is homing mode
Homing mode	60980008	0x4D00	User Setting	Searching homing mode
Home_Offset	607C0020	0x4100	User Setting	Offset after homing
Homing_Speed_Switch	60990120	0x5010	User Setting	Velocity for searching homing switch signal

Homing_Speed_Zero	60990220	0x5020	User Setting	velocity for searching homing signal
Homing_Accelaration	609A0020	0x5200	User Setting	Homing Accelaration
Control word	60400010	0x3100	Write F first the 1F	Lock motor, motor begin to move.

For details about how to find Homing , see Appendix IV.

For Homing mode control through communication, refer to the communication cases in the appendix

Homing mode control can also be achieved via external I/0 Setting

6.6 Analog - Speed mode（3）

6.6.1 Related object

In this mode, the motor movement is directly controlled by the external analog signal from the driver X3 port. The objects to be defined in this mode are:

Table 6-5 Objects in analog - speed mode 3

Name	CANopen address	Modbus address	Value	Description
Operation_Mode	60600008	0x3500	3	Operation mode of drive is target speed control mode.
Analog 1_Filter	25020110	0x1610	User Setting	Filter parameter of analog signal f=4000/（2π*Analog1_Filter） τ= Analog1_Filter/4000（S）
Analog 1_Dead	25020210	0x1620	User Setting	Dead space of analog signal 1
Analog1_Offset	25020310	0x1630	User Setting	Offset of analog signal 1
Analog1_out_polarity	25021410	0x1740	User Setting	polarity of analog output 0: The same as the input polarity 1: In contrast to the input polarity
Analog_speed_rpm at 10V	25021310	0x1730	User Setting	Analog 10V speed, it is without dead and offset.
Analog_Speed_Con	25020708	0x1670	User Setting	Analog signal control speed 0：Not valid 1：Analog Channel 1 valid (AIN1) 0x10 ～ 0x1f： AIN1“control the internal speed【x-10】” Mode 3 and mode 1 are valid
Profile_Acce	60830020	0x4B00	User Setting	Acceleration
Profile_Dece	60840020	0x4C00	User Setting	Deceleration
Control word	60400010	0x3100	F	Lock motor, motor begin to move.

6.6.2 Analog signal process diagram

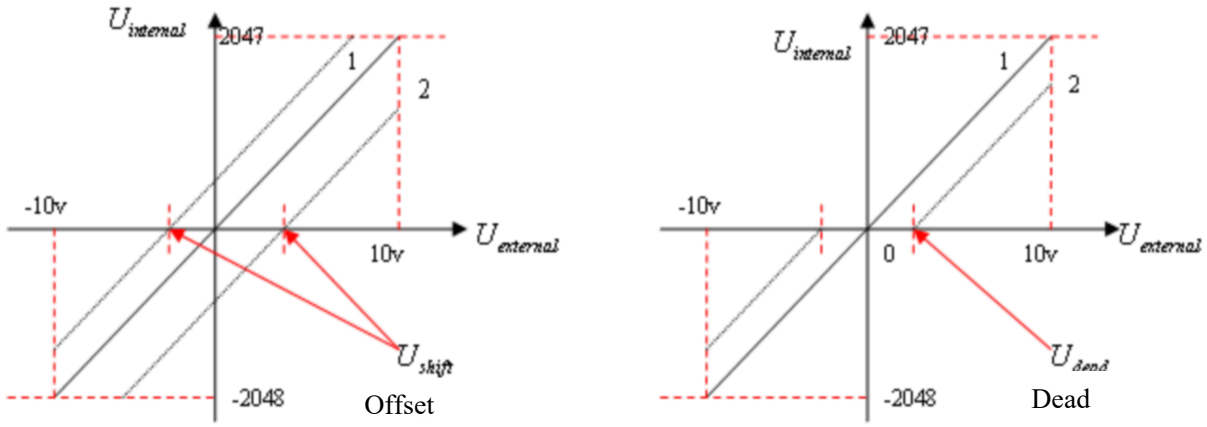


Figure 6-4 Analog signal process

Table 6-6 Analog signal variable describtion

Variable	Description	Scope
Uinternal	External voltage is same to internal data	-10V to 10V If there is no offset or dead zone voltage, the value ranges from -2048 to 2047
Uexternal	External input voltage	-10V～10V
Ushift	Offset voltage	0 to 10 corresponds to 0 to 8191
Udead	Dead zone voltage	0 to 10 corresponds to 0 to 8191

6.6.3 Analog - speed mode（without dead and offset voltage）

Requirement: DIN1 for drive enable, DIN2 for drive error reset, unlimited bit switch. 10V corresponds to the rated speed of 1000rpm, and -10V corresponds to the rated speed of -1000 rpm. Select Analog Channel 1 (AIN1) to control the speed.

Table 6-7 Parameter setting

Name	CANopen address	Modbus address	Value	Description
Operation_Mode	60600008	0x3500	3	Operation mode of drive is target speed control mode.
Analog 1_Filter	25020110	0x1610	5	Filter parameter of analog signal f=4000/（2π*Analog1_Filter） τ= Analog1_Filter/4000（S）
Analog 1_Dead	25020210	0x1620	0	Dead space of analog signal 1
Analog1_Offset	25020310	0x1630	0	Offset of analog signal 1
Analog1_out_polarity	25021410	0x1740	0	polarity of analog output 0: The same as the input polarity 1: In contrast to the input polarity
Analog_speed_rpm at 10V	25021310	0x1730	1000	Analog 10V speed, it is without dead and offset.
Analog_Speed_Con	25020708	0x1670	1	Analog signal control speed 0：Not valid 1：Analog Channel 1 valid (AIN1) 0x10～0x1f： AIN1“control the internal speed【x-10】” Mode 3 and mode 1 are valid
Profile_Acce	60830020	0x4B00	9830	Acceleration
Profile_Dece	60840020	0x4C00	9830	Deceleration
Store_Data	2FF00108	0x2910	1	1：Store the parameters un-involves Motor. 10：initialize the parameters un-involves motor

6.6.4 Analog - speed mode（set dead zone voltage）

Requirement: -0.5V～0.5V are dead zone voltage, means the speed is 0 during the voltage from -0.5V～0.5V.

And 10V is at 1000rpm, -10V is at -1000rpm.

Use analog channel 1 as AIN1 to control speed.

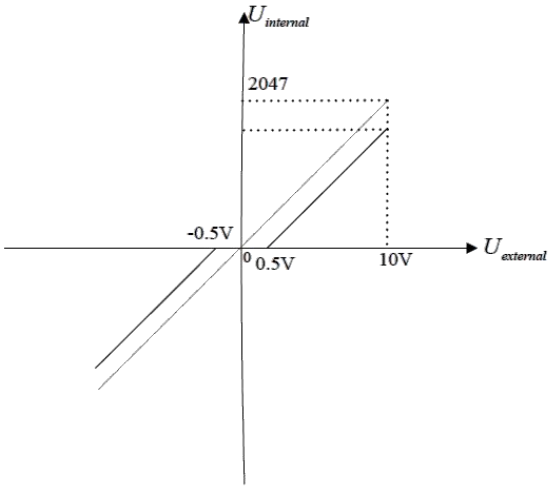


Figure 6-5 Control speed diagram 1

In fact the speed of 10V is 950rpm (=(10V-0.5V)/10V*1000rpm) due to the dead zone voltage.

Base on chapter 6.6.3, the index have to change as below.

Table 6-8 Parameter of Analog 1

Name	CANopen address	Modbus address	Value	Description
Analog 1_Dead	25020210	0x1620	0.5V （=410DEC）	Dead zone voltage of external analog signal 1

6.6.5 Analog - speed mode（set dead zone voltage and offset voltage）

Requirement: offset voltage is 1V, Dead zone is 0.5V-1.5V, 10V 对应 1000rpm. And choose analog channel 1 as AIN1 control speed.

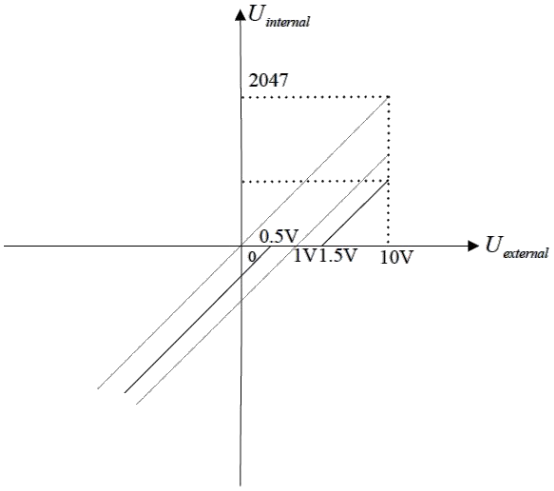


Figure 6-6 Speed mode diagram 2

In fact the speed of 10V is 850rpm (=(10V-1V-0.5V)/10V*1000rpm)

-10V is 1050rpm (=(-10V-1V+0.5V)/10V*1000rpm) due to the dead zone and offset voltage.

Base on chapter 6.6.3, the index have to change as below.

Table 6-9 Parameters of Analog signal 1

Name	CANopen address	Modbus address	Value	Description
Analog 1_Dead	25020210	0x1620	0.5V(=410DEC)	Dead space of analog signal 1
Analog1_Offset	25020310	0x1630	1V(=819DEC)	Offset of analog signal 1

6.7 Multi- position control mode（1）

As for multi-position control mode is active the target position to control motor by input port. And there are 3 items to active it as below.

1.Multi- speed control can run in 1 mode only.

2.Analog - Speed control (25020708) object is 0, the analog - speed channel is invalid.

3.At least, there is 1 input port that defined multi-target position control 0/1/2/3”.

Please input signal 0,1,2,3 which will be constituted the codes of binary system and selected as multi-target speed control from 0--15 channel, and the list as below.

Table 6-10 Binary code

Multi-position 3	Multi-position 2	Multi-position 1	Multi-position 0	Din_Position
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

Note: If the input signals 0,1,2,3 in the multi- position are not selected, the signal defaults 0.

The motor needs to run seven positions, requiring the position of the 0 stage to go to the position of 5000 pulses at a speed of 100 RPM, the one stage to go to the position of 15000 pulses at a speed of 150 RPM, and the second stage to go to the position of 28500 pulses at a speed of 175 RPM. Paragraph 3 goes to the position of -10,500 pulses at 200 RPM, paragraph 4 goes to the position of -20,680 pulses at 300 RPM, paragraph 5 goes to the position of -30,550 pulses at 325 RPM, paragraph 6 goes to the position of 850 pulses at 275 RPM. Paragraph 7 goes to a position of 15,000 pulses at 460 RPM.

Set I/O in the following table:

Table 6-11 IO setting

Name	Value
DIN1	Drive enable
DIN2	Drive enable
DIN3	Set-point active
DIN4	Multi-position 0
DIN5	Multi-position 1
DIN6	Multi-position 2

Table 6-12 Position and speed setting

Name	CANopen address	Modbus address	Value	Description
Din_Pos 0	20200120	0x0C10	5000	Multi-position 0
Din_Speed 0	20201120	0x0D10	100	Multi-speed 0
Din_Pos 1	20200220	0x0C20	15000	Multi-position 1
Din_Speed 1	20201220	0x0D20	150	Multi-speed 1
Din_Pos 2	20200320	0x0C30	28500	Multi-position 2
Din_Speed 2	20201320	0x0D30	175	Multi-speed 2
Din_Pos 3	20200420	0x0C40	-10500	Multi-position 3
Din_Speed 3	20201420	0x0D40	200	Multi-speed 3
Din_Pos 4	20200520	0x0C50	-20680	Multi-position 4
Din_Speed 4	20201520	0x0D50	300	Multi-speed 4
Din_Pos 5	20200620	0x0C60	-30550	Multi-position 5
Din_Speed 5	20201620	0x0D60	325	Multi-speed 5
Din_Pos 6	20200720	0x0C70	850	Multi-position 6
Din_Speed 6	20201720	0x0D70	275	Multi-speed 6
Din_Pos 7	20200820	0x0C80	15000	Multi-position 7
Din_Speed 7	20201820	0x0D80	460	Multi-speed 7
Multi-speed/ Position switching delay	20203810	0x0F80	10	Multi-speed/Effective input delay after changed position
Din_Mode0	20203108	0x0F10	1	Operation mode select by din port when inactive
Din_Mode 1	20203208	0x0F20	3	operation mode select by din port when active
Din_Control_Word	20203310	0x0F30	2F	Input "enable" signal controls the control word setting
Profile_Acce	60830020	0x4B00	50	Acceleration
Profile_Dece	60840020	0x4C00	50	Deceleration
Store_Data	2FF00108	0x2910	1	1：Store the parameters un-involves Motor. 10：initialize the parameters un-involves motor

Note: The unit of multi- position is "step", the unit of multi- speed is "rpm", and the unit of acceleration is "rps/s", which should be converted into internal DEC units during communication.

The steps of operation are as following.

1.Input enable to the drive.

2. Select the line position you want to move. And changed DIN4, DIN5, DIN6 level.

3. Set the active point and run the program.

6.8 Multi-speed control mode

Table 6-13Multi-speed binary code

Multi-speed 3	Multi-speed 2	Multi-speed 1	Multi-speed 0	Input speed control
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10

1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

Note: The signal is defaulted as 0 if do not select input signal 0，1，2，3 for multi-target position.

The parameter of multi-speed control 0-15, can select by IO port, and also will be achieved by analog signal to control it.

The parameters of analog speed will be covered by the value of using analog signal.

Table 6-14 Parameter setting of multi-speed

Name	CANopen address	Modbus address	Value	Description
Analog_Speed_Con	25020708	0x1670	1x	Analog signal control speed 0：Not valid 1：Analog Channel 1 valid (AIN1) 0x10～0x1f：AIN1“control the internal speed【x-10】” Mode 3 and mode 1 are valid

Chapter 7 Communication function

Bus stepper motor driver adopts communication protocols: RS232 serial communication protocol, RS485 Modbus communication protocol, CANOpen bus communication protocol,EtherCAT bus communication protocol.The solution of selected controller as KINCOstep PC software to communicate to control drive in the field as uniaxial and multiaxial system replaced the old operation as pulse direction mode， makes the system anti-interference ability and transparency better, and reduces the motion control module for the system, thus reducing the system cost.

The following points should be noted when using the communication control driver:

1.The I/O port DIN1 is defaulted as pulse input, DIN2 as direction input, DIN3 is motor free. Cancel the I/O settings by software if use communication control.

2.The internal objects of the drive are divided into engineering units and internal units. Writing and reading using communication need to be converted to the internal unit DEC. Using communication to write and read are internal units, need to pay attention to conversion. Refer to the appendix "Common object conversion relationship table" for details.

3. When using the CANopen bus communication protocol, RS232 serial port communication protocol and RS485 Modbus communication protocol for control, it is necessary to deal with many read and write data instruction intervals, ensure that only one driver sends data requests at any time on the communication network, and do a good job of communication error processing, so as to avoid communication entering a dead loop.

4. Some internal objects of the drive although the data length is multiple bits, but the actual engineering use does not need to use the maximum value, so some objects default to the maximum value limit, such as the target speed although the 32-bit data length, theoretically can write a large data, but in fact commonly used motors do not allow such high speed. Therefore, the maximum value of the drive is limited to 24576000 (converted to 1500RPM) to ensure system security. Data exceeding the maximum value will not be accepted by the drive and will automatically default to this maximum value.

7.1 RS232 commnunication

7.1.1 RS232 hardware

When the driver communicates with the upper computer, HMI, and PC,the wiring of a single stepper driver is as follow,you can also directly use the Console cable.

PC 9 Pin D		**Driver RJ45(X2)**
RxD (2)	----------	TXD (3)
TxD (3)	----------	RXD (6)
GND (5)	----------	GND (4)

7.1.2 RS232 parameter

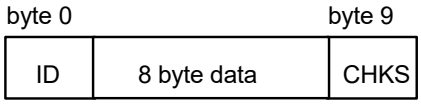
Table 7-1 RS232 communication parameter

Name	CANopen address	Modbus address	Value	Description
Node_ID	0x100B0008	0x0600	1	Driver ID Note：1) The station ID can be set from 1 to 63 by SW6-SW1 or from 1 to 127 by Ox2FE400. 2)Store and reboot the parameters if updated.
Node_ID_Offset	0x2FE40008	0x2800	127	ID node is from 1 to127; Note: the setting is valid when SW6-SW1 is on OFF. And Store and reboot the parameters if updated.
RS232_Baudrate	0x2FE00010	0x2400	259	For RS232 Baud rate setting Value Baud rate 2082 4800 1041 9600 520 19200 259 38400 86 115200 Need to reboot
Store_Control_Data	0x2FF00108	0x2910	1	1：Store the parameter which changed. 10: Initialize the parameters which updated

7.1.3 Free transfer protocol

RS232 follow up the protocol both for master and slave. PC can send any message to stepping drive, and the drive which set address will return a message after calculated these data.

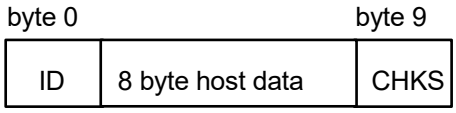
The formatting of RS232 transfer protocol is 10bytes.



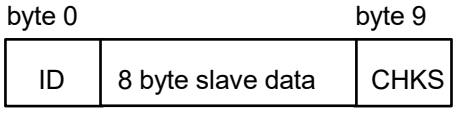
ID:Slave address

CHKS ＝－SUM(byte0,…,byte8)，CHKS is the last two digits of the above calculation result.

PC send：



PC receive（Driver return）：



Note: Each 10bytes has a CHKS.

There isn’t any return from drive if you send an error address which is not in the list. After the master sends the data correctly, the slave looks for the data corresponding to the address number and checks the parity value, if the value does not match the value calculated by the slave , the slave does not respond.

7.1.4 Data Protocol

Data protocol is different with transfer. The content of Data protocol is the 8 bytes middle of the 10bytes of transfer protocol. All value and function is shown by index and sub-index. The formatting are download and upload.

A:Download means master write value to the object of slave, then will be error if download on the address which do not in the list.

Table 7-2 Master send message

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
CMD	INDEX		SUB INDEX	DATA			

CMD Specified the data transfer direction and data size.
0x23 Send 4bytes of data (bytes 4...7 involves 32bits)
0x2b Send 2bytes of data (bytes 4，5 includes 16bits)
0x2f Send 1bytes of data (bytes 4 includes 8bits)
INDEX The address of send object
SUB INDEX The address of send sub object

The order of the 4bytes the data is from low to high. For example, write 7650DEC, 607A0020 inc. and 7650 is 10 hex, 1DE2 is 16 hex. The object is required 4 bytes, the result of value is only 2bytes as 1D E2. so can add 0 on the hight position as 00 00 1D E2.

DATA： byte4=E2 byte5=1D byte6=00 byte7=00

Table 7- 3 Slave return

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
RES	INDEX		SUB INDEX	RESERVED			

RES Show the return of slave
0x60 Message send out successful
0x80 Error，figure out from the byte as 4…7
INDEX 16bytes address same to master
SUBINDEX 8bytes address same to master
RESERVED Backup

For example:
Master send download order to slave:
01 23 7A 60 00 E2 1D 00 00 03 （The order written to the target position of slave as 607A0020）
Slave return:
01 60 7A 60 00 E2 1D 00 00 C6
Meaning:
01－Slave address
60－the data bytes of transfer is 2bytes, saved by byte4... byte5 from the 10 bytes of the response.
byte4=E2，byte5=1D，byte6=00，byte7=00
So DATA= byte7 byte6 byte5 byte4 = 1DE2（hex）=7650 DEC

B:Upload，means master read the object address of slave, and will be error if upload on the target position which do not in the list.

Table 7-4 Master send message

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
CMD	INDEX		SUB INDEX	RESERVED			

CMD	**Specified the data transfer direction**
0x40	
INDEX	16bytes address
SUBINDEX	8bytes address
RESERVED	4…7 bytes is useless

Table 7- 5 Slave return

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
RES	INDEX		SUB INDEX	DATA			

RES	**Show the return of slave**
0x43	Bytes 4...7 involves 32 bits data
0x4B	Bytes 4,5 involves 16 bits data
0x4F	Bytes 4 involves 8 bits data
0x80	Error, bytes 4…7 with error
INDEX	16bytes address same to master
SUBINDEX	8 bytes address same to master

If data is correct, byte4... byte7 contains a total of 4 bytes to store the values read from the slave object, the order is from low to high. And the correct value is byte7,byte6,byte5,byte4； if there is any error. The 4bytes is different.

For example:
Master send upload order to slave:
01 40 7A 60 00 00 00 00 E5 （（The target position is 607A0020）
Slave return:
01 43 7A 60 00 E2 1D 00 00 E3
Meaning：
01－Slave address is 1
43－Received data is 4bytes, saved by byte4... byte5 from the 10 bytes of the response.
byte4=E2，byte5=1D，byte6=00，byte7=00
So DATA= byte7 byte6 byte5 byte4 = 1DE2（hex）=7650 DEC

7.1.5 RS232 communication address

Please refer chapter 6 to know more about each mode operation.
As for object address, visit appendix VII.
All of communication address is on appendix VII
RS232 Communication case, see RS232 Appendix III

7.2 RS485

7.2.1 RS485communication port

X1 interface of FM860-LA support both for RS485 and RS422. Can modify the drive parameter and monitor the status. Please refer the picture as following.

RS485 wiring

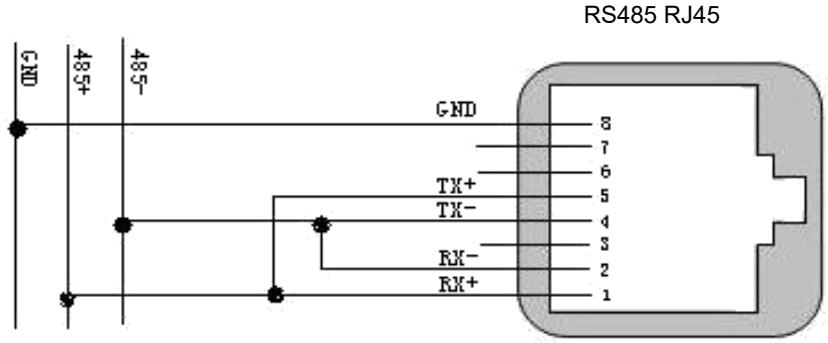


Figure 7-1 RS485 wiring diagram

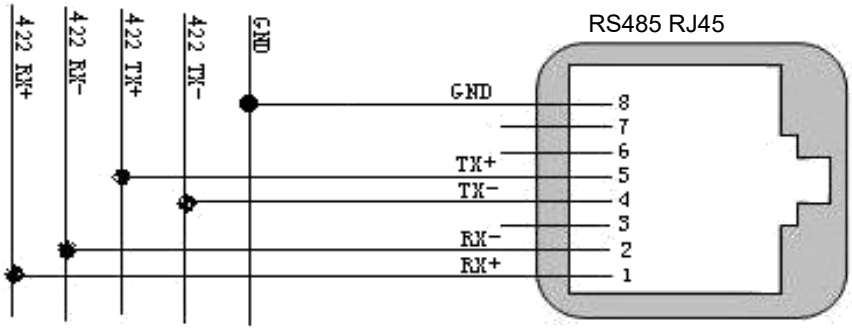


Figure 7-2 RS422 wiring diagram

The pin description of the RS485 conversion cable transfers from the 9-pin D connector to the RJ45 connector

9 Pin D	**FM860 RJ45(X2)**
RX+ (2) ----------	RX+ (1)
TX+ (3) ----------	TX+ (5)
RX- (7) ----------	RX- (2)
TX- (8) ----------	TX- (4)
GND (5) ----------	GND (8)

The slave can be directly connected with network cable

FM860 RJ45(X2)		**FM860 RJ45(X2)**
RX+ (1)	----------White and orange--------	RX+ (1)
RX- (2)	----------Orange----------	RX- (2)
NC(3)	----------White and green----------	NC (3)
TX- (4)	----------Blue----------	TX- (4)
TX+ (5)	----------White and blue----------	TX+ (5)
NC (6)	---------Green----------	NC(6)
NC (7)	-------White and brown----------	NC(7)
GND (8)	----------Brown----------	GND (8)

Note:

1、All the TX, RX of the slave can be directly connected, using a series connection mode, can not use a star connection mode;

2、The master and the last slave need to connect the terminal resistance of 120 Ω; The FM860 can be connected to the terminal resistor with SW9 and SW10 switches.

3、FM Series needn't external 24V power supply to RS485.

4、Use net cable with Shielded to communicate and don't forget GND.

7.2.2 RS485 communication parameter

Table 7-6 Communication parameter

Name	CANopen address	Modbus address	Value	Description
Node_ID	0x100B0008	0x0600	1	Drive ID Note: 1) The ID station number can be set 1 ~ 63 by SW6-SW1 or 1 ~ 127 by 0x2FE400. 2) Store and reboot the parameters if updated.
Node_ID_offset	0x2FE40008	0x2800	127	ID node is from 1 to127; Note: the setting is valid when SW6-SW1 is on OFF. And Store and reboot the parameters if updated.
RS485 Baud rate	0x2FE00010	0x2400	520	For RS485 Baud rate setting Value Baud rate 1041 9600 520 19200 259 38400 86 115200 Need store and reboot
Store_Control_Data	0x2FF00108	0x2910	1	1：Store the parameter which changed. 10: Initialize the parameters which updated
Others			Fixed value	Data bit = 8 stop bit = 1 No parity

7.2.3 MODBUS RTU communication protocol

Driver RS485 supports MODBUS RTU communication protocol, each 8-bit data is composed of two 4-bit hexadecimal data, that is, the general hexadecimal number, the data structure is 11 Bit character format, the data bit is 8 Bit, and the verification method is CRC verification. Its internal object is a discontinuous data register (mapped to 4X when read or written by the upper computer).

Modbus RTU communication protocol format

A MODBUS message is placed by the transmitting device into a frame that has a known beginning and ending point. This allows devices that receive a new frame to begin at the start of the message, and to know when the message is completed. In RTU mode, message frames are separated by a silent interval of at least 3.5 character times.

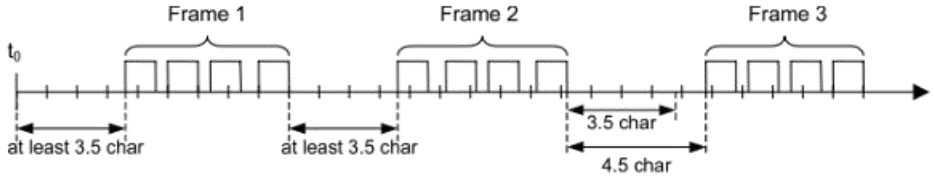


Figure 7-3 Frame formatting 1

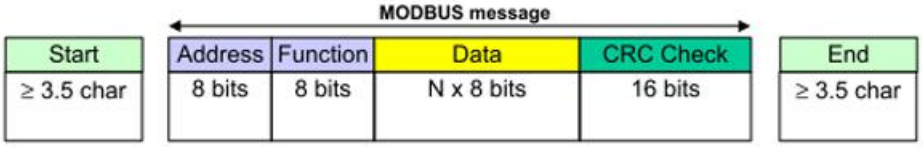


Figure 7-4 MODBUS communication formatting

The entire message frame must be transmitted as a continuous stream of characters. If a silent interval of more than 1.5 character times occurs between two characters, the message frame is declared incomplete and should be discarded by the receiver.

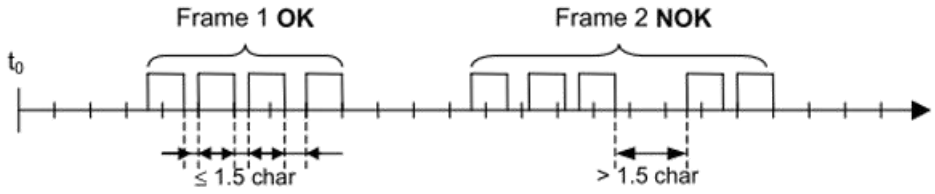


Figure 7-5 Frame formatting 2

7.2.4 Modbusfunction mode

Function code 0x03: Read data registers

Table 7-7 Request format

Target station number	Function code	Start address High byte	Start address Low byte	Read count High byte	Read count low byte	CRC
1 Byte	03	1 Byte	1 Byte	1 Byte	1 Byte	2 Byte

Table 7-8 Correct answer format:

Target station number	Function code	Return data number of bytes	Register 1 High byte	Register 1 Low byte	…	CRC
1 Byte	03	1 Byte	1 Byte	1 Byte	…	2 Byte

For example:

Send massage 01 03 32 00 00 01 8A B2 meaning

01： ID

03： Function code read data register

32 00： Read object status word 60410010 modbus address

00 01： Read word data number

8A B2： Check code

Return message 01 03 02 00 31 79 90 meaning

01： ID

03： Function code read data register

02： Returns the number of byte data

00 31： Returns the status word data of object

79 90： Check code

Function code 0x06：Write a single data register

Table 7-9 Request format

Target station number	Function code	Register address High byte	Register address Low byte	Register value High byte	Register value low byte	CRC
1 Byte	06	1 Byte	1 Byte	1 Byte	1 Byte	2 Byte

Return formatting: The messgae will return if the setup is successful.

For example:

Send massage 01 06 31 00 00 0F C7 32 meaning

01： ID

06： Function code write word

31 00： Write object control word 60400010's modbus address, data length is a WORD.

00 0F： Write data in hexadecimal 000F

C7 32： Check code

Function 0x10：Write multiple register

Table 7-10 Request format

Target station number	Function code	Start address High byte	Start address Low byte	Quantity High byte	Quantity Low byte	Number of bytes	Register 1 High byte	Register 1 Low byte	..	CRC
1 Byte	10	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	..	2 Byte

Table 7-11 Correct answer format

Target station number	Function code	Start address High byte	Start address Low byte	Quantity High byte	Quantity Low byte	CRC
1 Byte	10	1 Byte	1 Byte	1 Byte	1 Byte	2Byte

For example:

Send massage 01 10 6F 00 00 02 04 00 00 00 32 9B 88 **meaning**

01： ID
10： Function code write multiple WORD
6F 00： Write “target speed” 60FF0020’s modbus address, Data length is 2 WORD
00 02: Wirte 2 WORD
04： Data length is 4 BYTE（2 WORD)
00 00 00 32： Write data in hexadecimal as 00320000, Decimalism as 3276800, and converted to 200RPM
9B 88： Check code

Return message 01 10 6F 00 00 02 5C DC **meaning**

01： ID
10： Function code read register
6F 00： Object address
00 02： Write WORD number
5C DC： Check code
It will response error if access the error number. And return a no-normal function code as 0x80+ function code.

Table 7-12 Abnormal answer format

Station number	Abnormal function code	Error Code	CRC
1 Byte	1 Byte	1 Byte	2Byte

Table 7-13 Meaning of Error value

Error code value	Meaning
0x01	Function code do not support
0x02	Register address does not exist
0x03	Data error or register number is wrong
0x04	Write operation failed, including data range or object is read-only

7.2.5 Modbus communication address

Please refer chapter 6 to know more about each mode operation.

As for object operation address, access appendix VII (RS485 do not support all in internal objects.)

And RS485 communication case, please refer appendix II.

7.3 CANopen communication

CANopen is the most famous and successful of the open fieldbus standards, which has been widely recognized and widely used in Europe and the United States. In 1992, the Association of Automation CAN Users and Manufacturers (CiA) was established in Germany and began to develop CANopen, an application layer protocol for automation CAN. Since then, the members of the Association have developed a series of CANopen products, which are widely used in machinery manufacturing, pharmaceuticals, food processing and other fields.

FM series stepping motor drive is standard CAN slave device, and follow CANopen 2.0A/B protocol. Any PC can communicate with FM series Drive if the PC supported the protocol. The drive uses a strictly defined list of objects called the Object dictionary， which based on the standard as CANopen, all objects has clear definition of the functions. The objects is same to the memory address, part of object as speed and position can modify by external controller, and part of object as status and error messages just can modify by the drive. The objects are as following.

For example:

Index	Sub	Bits	Property	Meaning
6040	00	16(=0x10)	RW	Device status control word
6060	00	8(=0x08)	RW	Operate mode
607A	00	32(=0x20)	W	Target position
6041	00	16(=0x10)	MW	Device status word

The property of object is as below :

RW： Object can be read and written;

RO： Read only;

WO： Write only;

M： Can be mapped, similar to indirect addressing;

S： Objects can be stored in Flash-ROM area, and data don’t be lost even power-off;

7.3.1 Hardware introduction

CAN communication protocol mainly describes the mode of information transmission between devices. The definition of CAN layer is consistent with the open system interconnection model (OSI). Each layer communicates with the same layer on another device. The actual communication occurs in the two adjacent layers of each device, and the devices are only interconnected through the physical medium of the physical layer of the model. The specification of CAN defines the bottom two layers of the model, the data link layer and the physical layer. CAN bus physical layer is not strictly stipulated, can use a variety of physical media such as twisted pair optical fiber, etc., the most commonly used is twisted pair signal, the use of differential voltage transmission (commonly used bus transceiver), two signal lines are called CAN_H and CAN_L, static time is about 2.5V, at this time the state is expressed as logic 1, can also be called hidden bit, CAN_H is higher than CAN_L to represent logical 0, which is called display, and the usual voltage value at this time is CAN_H=3.5V and CAN_L=1.5V, and display is preferred in competition.

Standard CAN slave station communication RJ45 wiring diagram:

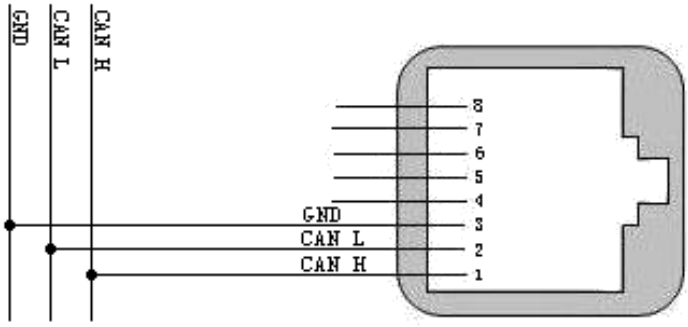


Figure 7-6 Standard CAN interface as RJ45

CAN conversion cable pin description, from 9-pin D connector to RJ45 connector

9 pin D	**FM860 RJ45(X2)**
CAN_H (7) ----------	CAN_H (1)
CAN_L(2) ----------	CAN_L (2)
GND (3) ----------	GND(3)

The slave can be directly connected with network cable

FM860 RJ45(X2)		FM860 RJ45(X2)
CAN_H (1) ----------	White and orange----------	CAN_H (1)
CAN_L(2) ----------	Orange----------	CAN_L(2)
GND(3) ---------	White and green----------	GND (3)
NC(4) ----------	Blue----------	NC(4)
NC(5) ----------	White and blue----------	NC (5)
NC(6) ----------	Green----------	NC(6)
NC 7) ----------	White and brown----------	NC(7)
NC(8) ----------	Brown----------	NC(8)

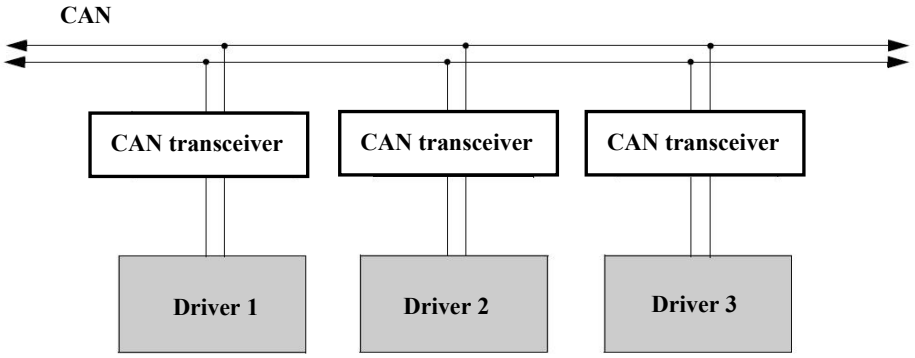
Note：

1.All CAN_L and CAN_H pins of the slave station can be directly connected and connected by series connection.
2.The master and the last slave need to be connected to a 120 Ω terminal resistor, the FM860 driver is built in and can be enabled by the dip switch SW9;
3.The FM series stepper driver does not require an external 24V power supply to power the CAN；
4.Use net cable with Shielded to communicate and don’t forget GND.
5.The maximum communication distances of various baud rates.

Table 7-14 The max communication distance of baud rate

Communication speed	Communication distance
1Mbit/S	25M
500Kbit/s	100M
250Kbit/s	250M
125Kbit/s	500M
50Kbit/s	600M
25Kbit/s	800M

Typical CAN bus topology figure



7.3.2 CAN communication frame structure

Standard data frame messages begin with the frame start bit, followed by a 12-bit arbitration field. The arbitration field contains an 11-bit identifier (COB-ID) and a Remote Send request (RTR) bit. An identifier defines the type of information contained in a message, and each receiving node uses it to determine whether the message is its own content.

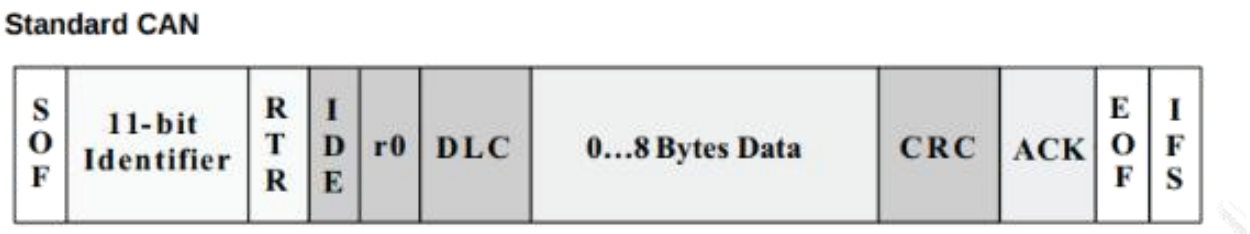
The arbitration field is followed by a 6-bit control field that provides more information about the content of the message. The first digit in the control field is the identifier extension (IDE) bit, which is used to distinguish whether the message is a standard data frame or an extended data frame, which is indicated by the dominant state (logic level 0). The second place in the control field is the reserved (RB0) bit, which is in the dominant state (logic level 0). The last four bits in the control field represent the Data Length encoding (DLC), which specifies the number of bytes of data contained in the message.

The control field is followed by the data field. This field carries message data, such as SDO and PDO messages. The length of this field is variable, ranging from 0 to 8 bytes. The number of bytes can be selected by the user.

The data field is followed by the cyclic redundancy check (CRC) field, which consists of a 15-bit CRC sequence and a delimiter bit. The reply (ACK) field is sent as a hidden bit (logic level 1) and is rewritten as a dominant bit by any receiver that has correctly received the data. The receiver always replies regardless of the result of the receive filter comparison.

The last field is the End of Frame field, which consists of seven hidden bits indicating the end of the message.

Format of a standard frame



7.3.3 CANopen Communication parameter

Table 7-15 Communication parameter

Name	CANopen address	Modbus address	Value	Description
Node_ID	0x100B0008	0x0600	1	Drive ID Note: 1) The ID station number can be set 1 ~ 63 by SW6-SW1 or 1 ~ 127 by 0x2FE400. 2) Store and reboot the parameters if updated.
Node_ID_offset	0x2FE40008	0x2800	127	ID node is from 1 to127; Note: the setting is valid when SW6-SW1 is on OFF. And Store and reboot the parameters if updated.
CAN Baud rate	2F810008	0x2300	50	CAN Baud rate setting Value Baud rate 100： 1M 50： 500k 25： 250k 12： 125k 5： 50k Need store and reboot
Store_Control_Data	0x2FF00108	0x2910	1	1：Store the parameters which changed. 10: Initialize the parameters which updated

7.3.4 EDS

EDS (electronic data form) file is the identification file or similar code of the slave station connected to the PLC, through which to identify the type of slave station (which is similar in 401, 402, 403, or which device belongs to 402). This file contains all the information of the slave station, such as manufacturer, serial number, software version, supported baud rate type, OD that can be mapped and the attributes of each OD and so on, similar to the GSD file of Profibus. Therefore, before hardware configuration, we first need to import the EDS file from the station to the upper configuration software.

7.3.5 SDO

SDO is mainly used to transmit low-priority objects between devices. Typically, it is used to configure and manage slave devices, such as modifying PID parameters and PDO configuration parameters of current ring, speed ring, position ring, etc. This kind of data transmission is the same as MODBUS, that is, after the master station sends out, the slave station needs to return data response. This communication mode is only suitable for parameter setting, and is not suitable for data transmission with high real-time requirements.

SDO is used to access the object dictionary of device in CANopen protocol. Visitor is called client, and abject dictionary is accessed and provided CANopen device as server. The CAN message of client and the response of server always involves 8 bytes. And one request of client must have a response from the server.

SDO has 2 kinds of transmission process：

- Expedited transfer：Up to 4 bytes of data transmission
- Segmented transfer: Data length larger than 4 bytes

Table 7-16 SDO Basic Structure as below: Client→Server/Server→Client

Byte0	Byte1-2	Byte3	Byte4-7
SDO Command specifier	Object Index	Object sub-index	Data

SDO Command word contains the following information ：

- Download/upload
- Request /response
- Segmented / expedited transfer
- CAN frame bytes
- For subsequent each segment alternately reset and set the toggle bit.

SDO achieved five requests and response.

- Initiate Domain Download
- Download Domain Segment
- Initiate Domain Upload
- Upload Domain Segment
- Abort Domain Transfer

Download is for the write operation of object dictionary. Upload is for read operation. The details of SDO command word is as below.

Table 7-17 Initiate Domain Download

Initiate Domain Download Byte0							Byte1-3	Byte4-7
Bit	7-5	4	3	2	1	0	Subject index and sub-index	
Client→	ccs=1	-	n		e	s		Data
←Server	scs=3	-	-	-	-	-		-

Explanation:

- ccs: client command specifier，,1=Initiate download request。
- scs: server command specifier，3=Initiate download response
- n：The number of bytes representing meaningless data in the packet data [from (8-n) bytes to the seventh byte is meaningless] (n is valid when e=1 and s=1, otherwise n is 0).
- e：When e=0, the transmission is normal; when e=1, the transmission is accelerated.
- s：Indicates whether the data length is specified. 0 indicates that the data length is not specified, and 1 indicates that the data length is specified.
- e=0，s=0：Keep by CiA.
- e=0，s=1：The data byte is the byte counter, byte 4 is the data low part (LSB), byte 7 is the data high part (MSB).
- e=1，s=1：Data bytes 4-n indicates the data to be downloaded.
- e=1，s=0：Download an unknown number of bytes of data.

For example:

0x2f means the data to download 1byte（bytes 4 involves 8 bits）
0x2b means the data to download 2bytes（bytes 4,5 included 16bits）
0x23 is the data to download 4bytes（bytes 4,5,6,7 included 32bits）
0x21 is the Start frame to download over 4 bytes data, segmented downloading data
0x60 is download data successful

Table 7-18 Download domain segment

Download Domain Segment Byte0							Byte1-7
Bit	7-5	4	3	2	1	0	
Client→	ccs=0	t	n			c	Data
←Server	scs=1	t	-	-	-	-	-

Explanation:

- n ： Meaningless data bytes. The value will be 0 if do not specified data length. 。
- c ： ： 0 = Subsequent segment requires download
- t ： The trigger bit is alternately cleared and set for each subsequent segment (the first transmission is 0, equivalent to request/response).

For example:

0x00/0x10 is segmented downloading data, involves 8bytes data（bytes 1-7）
0x0bmeans the last piece of data, involves 2 bytes data.（bytes 1, 2）
0x20/0x30 is segmented downloading successful.

Table 7-19 Initiate Domain Upload

Initiate Domain Upload Byte0							Byte1-3	Byte4-7
Bit	7-5	4	3	2	1	0	Subject index and sub-index	
Client→	ccs=2	-	-	-	-	-		-
←Server	scs=2	-	n		e	s		Data

Explanation:

- n，e，s：Same to download of domain starting

For example:

0x40 is the data to ask upload subject
0x4f is the data to upload 1 byte.（bytes 4 involves 8 bits）
0x4b is the data to upload 2bytes.（bytes 4, 5 involves 16bits）
0x43 is the data to upload 4 bytes（bytes 4, 5, 6, 7 includes 32bits）
0x41is the start frame to upload over 4bytes, and segmented uploading data

Table 7-20 Upload Domain Segment

Upload Domain Segment Byte0							Byte1-7
Bit	7-5	4	3	2	1	0	
Client→	ccs=3	t	-	-	-	-	-
←Server	scs=0	t	n			c	Data

Explanation:

- n，c，t ： Same to download domain segment

For example:

0x60/0x70 is ask to upload segmented data
0x00/0x10 is the data to upload segmented data, includes 8bytes data（bytes 1-7）
0x19 is the last piece of data, involves 3 bytes data（bytes 1,2,3）

Table 7-21 Abort domain transfer

Abort Domain Transfer Byte0							Byte1-3	Byte4-7
Bit	7-5	4	3	2	1	0	Subject index and sub-index	Error code
Client→/←Server	cs=1	-	-	-	-	-		

For example:

0x80 is an interrupt transfer

Table7-22 Error code

Error code	Meaning
0x0504 0001	Client/server commands are invalid or unknown
0x0601 0002	Try to write a read-only object

0x0602 0000	Object is not in the object dictionary
0x0604 0041	Objects cannot be mapped to PDO
0x0607 0010	The data type does not match
0x0609 0011	The subindex does not exist
0x0609 0030	The parameter value is invalid and out of range
0x0800 0000	common fault

Example 1: Read control word parameters

Table 7-23 Send SDO message

Identifier	DLC	Daten							
		0	1	2	3	4	5	6	7
0x600+Node_ID	8	40	40	60	00	00			

0x600+Node_ID -- COB-ID
40 -- Request to upload the object's data
60 40 00 -- Control word object address
00 -- useless data

Table 7-24 Return SDO message

Identifier	DLC	Daten							
		0	1	2	3	4	5	6	7
0x580+Node_ID	8	4b	40	60	00	06	00	00	00

0x580+Node_ID -- COB-ID
4b -- Upload two bytes of data (bytes 4,5 contains 16 bits)
60 40 00 -- Control word object address
0006 -- Data

Example 2: Modify control word parameters

Table 7-25 Send SDO message

Identifier	DLC	Daten							
		0	1	2	3	4	5	6	7
0x600+Node_ID	8	2b	40	60	00	2f	00	00	00

0x600+Node_ID -- COB-ID
2b -- Download two bytes of data (bytes 4,5 contains 16 bits)
60 40 00 -- Control word object address
002f -- Data

Table 7-26 Return SDO message

Identifier	DLC	Daten							
		0	1	2	3	4	5	6	7
0x580+Node_ID	8	60	40	60	00	2f	00	00	00

0x580+Node_ID -- COB-ID
60 -- Successful data download
60 40 00 -- Control word object address
002f -- Data

7.3.6 COB-ID

COB-ID is a unique method of CANopen Communication protocol. Its full name is Communication Object Identifier-Communication object-ID. These COB-ids define the corresponding transport levels for PDO. The controller and the servo can define the same transmission level and the transmission content in their respective software configurations, so that after the controller and the servo use the same transmission level and transmission

content, the data transmission is transparent, that is, both sides know the data content to be transmitted. It is not necessary to reply whether the data is transmitted successfully when the data is transmitted.

The default ID allocation table is based on the 11-bit CAN-ID defined by CANopen 2.0A (CANopen 2.0B protocol COB-ID is 27 bits), which contains a 4-bit function code part and a 7-bit Node-ID part.

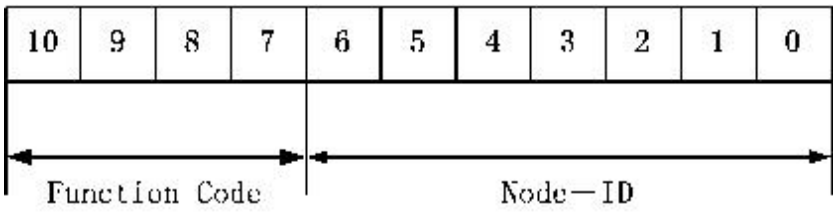


Figure 7-7 CAN-ID distribution format

Node-ID can define by defined by system integrators via the DIP switch of device. The scope of Node-id is from 1 to 127(0 is not allowed to be used)

Function Code is for data transmission, define the transport level of each PDO, SDO and message management, and function code is smaller, the priority is higher.

Table 7-27 Allocation list for CAN identifiers in master/slave connection set predefined by CANopen

Broadcast objects			
Object	Function code （ID-bits 10-7）	COB-ID	The index of communication parameters in OD
NMT Module Control	0000	000H	-
SYNC	0001	080H	1005H，1006H，1007H
TIME SSTAMP	0010	100H	1012H，1013H
CANopen Peer object of the master/slave connection			
Object	Function code （ID-bits 10-7）	COB-ID	The index of communication parameters in OD
EMCY	0001	081H-0FFH	1024H，1015H
TPDO1(send)	0011	181H-1FFH	1800H
RPDO1(receive)	0100	201H-27FH	1400H
TPDO2(send)	0101	281H-2FFH	1801H
RPDO2(receive)	0110	301H-37FH	1401H
TPDO3(send)	0111	381H-3FFH	1802H
RPDO3(receive)	1000	401H-47FH	1402H
TPDO4(send)	1001	481H-4FFH	1803H
RPDO4(receive)	1010	501H-57FH	1403H
SDO(Send/server)	1011	581H-5FFH	1200H
SDO(Receive/client)	1100	601H-67FH	1200H
NMT Error Control	1110	701H-77FH	1016H-1017H

Note:

1. COB-ID is smaller, the priority is higher.
2. The function code of COB-ID is fixed formatting；
3. COB-ID 为 00H、80H、100H、701H-77FH、081H-0FFH are systems management formatting.

7.3.7 PDO

Process data object can send 8 bytes data one time. No other pre-set protocols, use for the high-frequency switching data. PDO transmission mode used a new data exchange, the devices have to define the area both for data sending and receiving. And send related data to each other directly during the data exchange, which reduced more time to improve the efficiency of bus communication. So resulting in a high bus utilization.

1）Send PDO（TPDO）

Send PDO, for Stepping motor drive, is send out data, and receive by PLC. The function code as COB-ID of PDO is

1. 0x180+Station No. of Stepping motor drive
2. 0x280+Station No. of Stepping motor drive
3. 0x380+Station No. of Stepping motor drive
4. 0x480+Station No. of Stepping motor drive

2）Receive PDO（RPDO）

Receive PDO, as for Stepping motor drive, is receive data and which sent by PLC. The function code as COB-ID of sending PDO is

1、 0x200+Station No. of Stepping motor drive
2、 0x300+Station No. of Stepping motor drive
3、 0x400+Station No. of Stepping motor drive
4、 0x500+Station No. of Stepping motor drive

The design of FM series Stepping motor drive is follow up CANopen 2.0A. It support CANopen 2.0B also. And meanwhile, if the 8 PDO are un-enough, user can define a new PDO as 0x43FH for station 1, controller and drive are also follow it.

3）PDO transport Type

PDO have 2 type transport mode:

1.SYNC: Transmission is triggered by the synchronization message (Transmission type:0-240) In this transmission mode, controller must have the ability to send synchronous messages (the message is sent periodically at a maximum frequency of 1KHz), and servo will send after receiving the synchronous message.

- Acyclic: Pre-triggered by remote frame or by specific event of objects speicficed by the equipment sub-protocol. In this mode, Stepping Motor Drive will send out data in PDO after receiving SYNC message.
- Cyclic: Triggered after sending 1 to 240 SYNC message .In this mode, Stepping Motor drive will send out data in PDO after receiving n SYNC messages.

2.ASYNC (Transmission type: 254/255). The slave station sends the packet after the change regardless of whether the master station asks for it. In addition, you can define the interval between sending the same packet twice to prevent the packets with higher priorities from occupying the bus all the time (the lower the value of PDO, the higher the priority).

For FM series stepper drivers, synchronous cycle transmission and asynchronous transmission modes are currently supported.

4）PDO inhibit time

Each PDO can define a inhibit time. That is the minimum interval time between two continue PDO transmission. It is used to avoid the PDO in higher priority always occupying the communication. The inhibit time is 16bits un-signged integer. Its unit is 100us.

5））For example:

Use TPDO to update Target-speed by the object as 0x1A00. And seclected 0x1800 to configure the property of mapping object as 0x1A00.

Table 7-28 Property of object 0x1800

Property of object 0x1800		
Sub-index	Value	Meaning
0	3	The object has 3 sub-index
1	0x250+Node_ID	Node-ID
2	254	Asynchronous transmission
3	50	Disable transmission time interval

Table 7-29 Mapping of 0x1A00 TPDO

Mapping of 0x1A00 TPDO		
Sub-index	Value	Meaning
0	1	1 object are mapped to PDO
1	0x60ff0020	Object 0x60ff, sub-index 0x00, consisted of 32 bits

Table 7-30 The formatting of Send message

Identifier	DLC	Daten							
		0	1	2	3	4	5	6	7
0x250+Node_ID	8	Target-speed							

7.3.8 Boot-upp process

In the network initialization process, CANopen support both for extended boot-up and the process of minimum boot-up. And for extended boot-up is optional, but for minimum boot-up must be supported by each node. And the both of boot-up can exist in the same network at the same time. The node must be support extend boot-up process if selected the DBT of CAL to configure ID

These two initialization processes can be represented by a node state transition diagram, as shown in the following figure. The status diagram for extended boot-up has more states between pre-operation and operational states than it does for minimized boot-up.

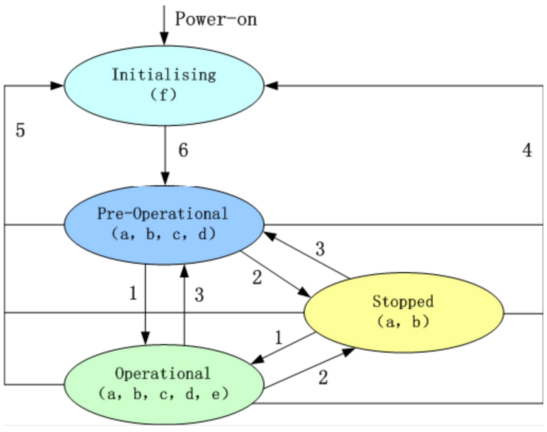


Figure 7-8 Node status conversion

Note：
The letters in the parenthesis means the objects which can used in this status:
a. NMT ， b. Node Guard ， c. SDO ， d. Emergency ， e. PDO ， f. Boot-up
State transition(1－5 are sent by NMT service)
1：Start_Remote_node
2：Stop_Remote_Node
3：Enter_Pre-Operational_State
4：Reset_Node
5：Reset_Communication
6：Initialization finish, enter pre-operational status and send boot-up message.

NMT services can make all or some nodes into different work state at any time. The CAN message of NMT sevice is consisted of 2 bytes as COB-ID=0. the first byte represents the type of service requests as NMT comm. And specifier, the second byte is Node-ID or 0.

7.3.9 NMT Module Control

NMT management message can be used to change the modes. Only NMT-Master node can send NMT module control message, and all slave must support NMT module control service, meanwhile NMT module control message needn’t response. The format of NMT message is as follows.

Table 7-31 NMT-Master → NMT-Slave(s)

COB-ID	Byte 0	Byte 1
0x000	CS	Node-ID

When Node-ID is 0, then all the NMT slave device are addressing, CS is command, its value is as follows.

Table 7-32 CS command value

Command word	NMT service
0x01	Enable remote node
0x02	Close remote node
0x80	Enter the pre-operation state
0x81	Reset node
0x82	Reset communication

For example, if you want a node in the operational status to return to the pre-operational status, then the controller needs to send following message.

Table 7-33 Controller message

COB-ID	Byte 0	Byte 1
0x000	0x80	0x02

7.3.10 Protection mode（Supervision Type）

Supervision type is that master selected what kind of check type during operation. And there are two kind of type as Heartbeat message and Node guarding to check the fault of slave and handle it.

- Heat beat message：The slave periodically sends message to the master during the "monitoring time". If the master does not receive the next heartbeat message from the slave after the "Heartbeat time" expires, the master determines that the slave is faulty.

Heartbeat Producer → Consumer(s)

Table 7-34 Heat beat message

COB-ID	Byte 0
0x700 + Node_ID	Status

Table 7-35 Status value

Value	Meaning
0x00	Boot-up
0x04	Stopped
0x05	Operational
0c7f	Pre-operational

When a Heartbeat node starts, its Boot-up packet is the first Heartbeat packet.

- Node protection: The master periodically sends the packet to the slave in the "monitoring time", and the slave responds immediately after receiving the packet. If the master does not receive the packet from the slave after the "Monitoring time * Life factor" time, the master judges that the slave is faulty!

NMT-MasterNode send remote frame(without data)as below.

NMT-Master → NMT-Slave

COB-ID
0x700+Node_ID

NMT-Slave Node send message as following:

Table 7- 36 NMT-Master → NMT-Slave message

COB-ID	Byte0
0x700 + Node_ID	Bit 7 : toggle bit Bit6-0 : Status

As for Byte 0, involves a toggle bit(bit7), and the toggle bit is turned to configure the 0 or 1 in node protection each time. The first node protection of toggle bit is “0”. And the bit from 0 to 6 means node status.

Table 7-37 Status value definition

Value	Meaning
0	Initializing
1	Disconnnected
2	Connecting
3	Preparing
4	Stopped
5	Operational
127	Pre-operational

Note:
The parts with the icon as *, will be provided to the node with extend boot-up. Please pay attention, the status 0, is never shown for node protection.
A node can not support Node Guarding and Heartbeat protocol at the same time. You can seclec one of them.

7.3.11CANopen communication address of drive parameter

Please refer Chapter 6 to know more about each operation mode.
Please access object list as Appendix VII in Chapter 9 for common object address
Please refer all communication address as Appendix VII
Please refer Chapter 9 Appendix I of CANopen communication case.

7.4 EtherCAT communication

EtherCAT (Ethernet for Control Automation Technology) is a fieldbus system based on Ethernet, used for industrial field level ultra-high speed I/O network, using the standard Ethernet physical layer, The transmission medium is twisted pair or optical fiber (100Base-TX or 100Base-FX). It is an industrial Ethernet technology with high performance, low cost, simple application and flexible topology.

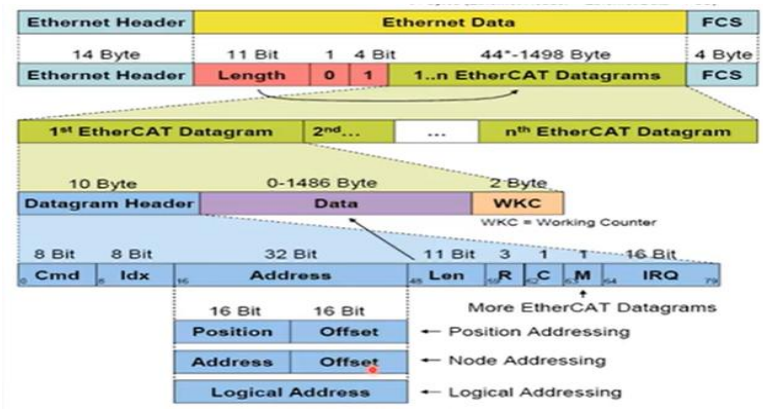
7.4.1 EtherCATcomposition and operation principle

EtherCAT system consists of master and slave . The master uses a standard Ethernet interface card or an embedded industrial control computer with an Ethernet interface, and the slave uses a dedicated slave control chip (ESC, such as ET1100, LAN9252, etc.). The master sends data frame packets to each slave , after receiving the packets, the slave extracts data from the data frame, or inserts data into the data frame, and then transmits the data frame to the next slave . After the last slave processes the data, it returns the fully processed packet to the last slave through the internal loop. It simply forwards the message to the previous slave station, and the first slave sends the response message to the master (see the figure below). ESC adopts the receive and forward mechanism. Ethernet frames can pass in both directions, but only when Ethernet frames enter from the downlink direction, the corresponding packets can be processed. If the network port to be forwarded is disconnected or disconnected, the ESC automatically performs an internal loopback forward and returns the packet to the previous slave station instead of losing the packet.

7.4.2 EtherCAT protocol

EtherCAT uses the standard Ethernet frame structure, in which the frame header is the standard Ethernet frame header, the frame type is fixed to 0x88A4. The packet consists of EtherCAT header and EtherCAT data, and several sub-packets form the EtherCAT data area. The sub-packet is composed of sub-packet header, data field and working counter. The sub-packet header marks the slave to which it is transmitted, and reads or writes to the sub-packet. The data area of the sub-packet can be the data of the slave ESC register, mailbox message or procedure data. In the process of communication between master and slave, the value of the 16-bit working counter (WKC) is the count of the read and write operations of the slave station. When the master station initiates the period control, a WKC value is set in advance. When the data frame traverses the entire device, the returned WKC value can be compared to verify whether the data packet is correctly processed by each slave node.

EtherCAT data frame structure：

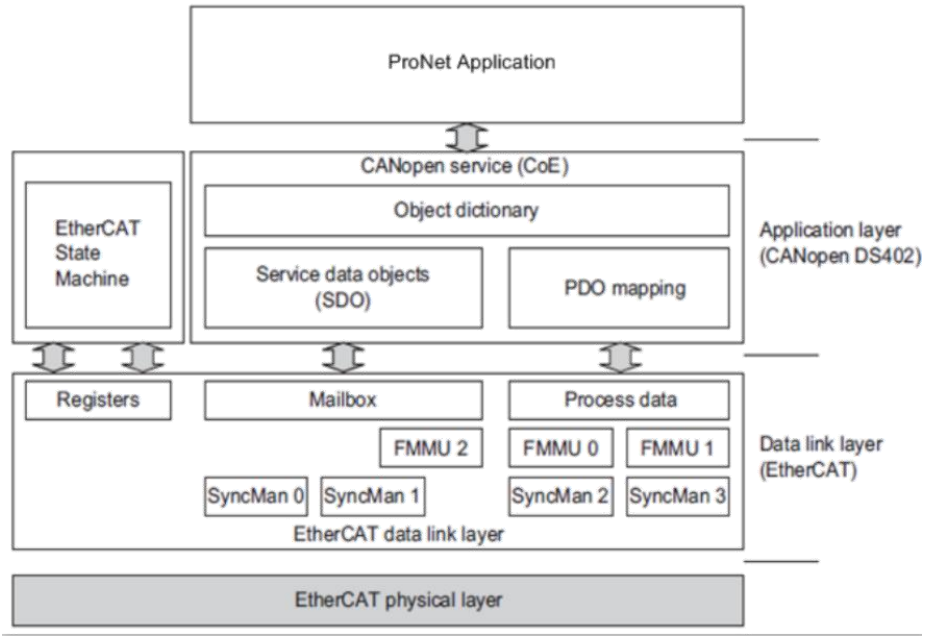


7.4.3 EtherCAT（CoE）network reference model

The EtherCAT (CoE) network reference model consists of two parts: the data link layer and the application layer. The data link layer is mainly responsible for EtherCAT communication protocol, and the application layer is embedded with CANopen communication protocol. The object dictionary in the CoE contains parameters, application data, and PDO mapping information. A process data object (PDO) consists of objects in the object dictionary that can be PDO mapped, and the contents of the PDO data are defined by the PDO mapping. PDO data is read and written periodically and does not need to look up the object dictionary. Mailbox SDO are aperiodic communications ,so look up an object dictionary when reading or writing them.

Note:

In order for SDO and PDO data to be properly parsed on the EtherCAT data link layer, FMMU (Fieldbus memory management Unit) and SyncManager (Synchronization Manager) are required for management and configuration.



Sync manager setting

Synchronization manager	Allocation (fixed)	Byte	Start address (fixed)
Synchronization manager0	The mailbox message from the master was received	128 byte（fixed）	0x1000
Synchronization manager1	The mailbox message sent to the master	128 byte（fixed）	0x1400
Synchronization manager2	The PDO message from the master station was received	0 ~ 128 byte	0x1800
Synchronization manager3	PDO message sent to the master	0 ~ 128 byte	0x1C00

FMMU setting

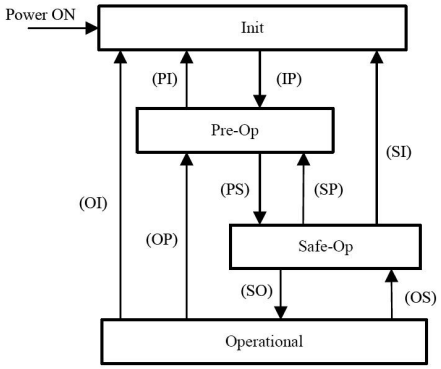
FMMU	Allocation
FMMU 0	FMMU0 is used for RxPDO data transmission, that is, the master periodically sends data to the slave
FMMU 1	FMMU1 is used for TxPDO data transmission, that is, the master periodically reads the slave data.
FMMU 2	FMMU2 is configured to send mailbox data preparation flags from the slave

7.4.4 EtherCAT slave information

EtherCAT slave information file (XML file) contains the slave device description, object dictionary, SyncManager and FMMU Settings, distribution clock, PDO configuration ,etc.

7.4.5 EtherCAT network state machine

The EtherCAT state machine describes the state and state changes of the slave application. The state change request is usually initiated by the master, and the slave responds.



Status description

Init：Initialization state, abbreviated as I;

Pre-OP：Pre-operation state, abbreviated as P;

Safe-OP：Safe operating status, abbreviated as S;

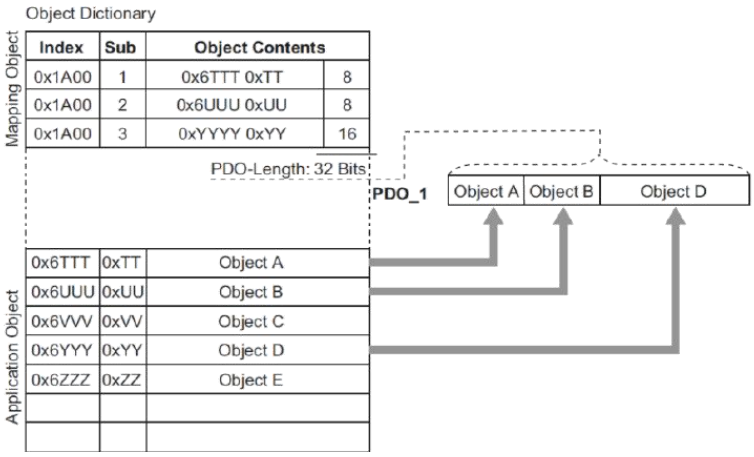
Operational：Operation status, abbreviated as O;

When converting from the initialization state to the operation state, it must be converted step by step in order, but it can be converted from the operation state back to each state. The status conversion operation process is as follows:

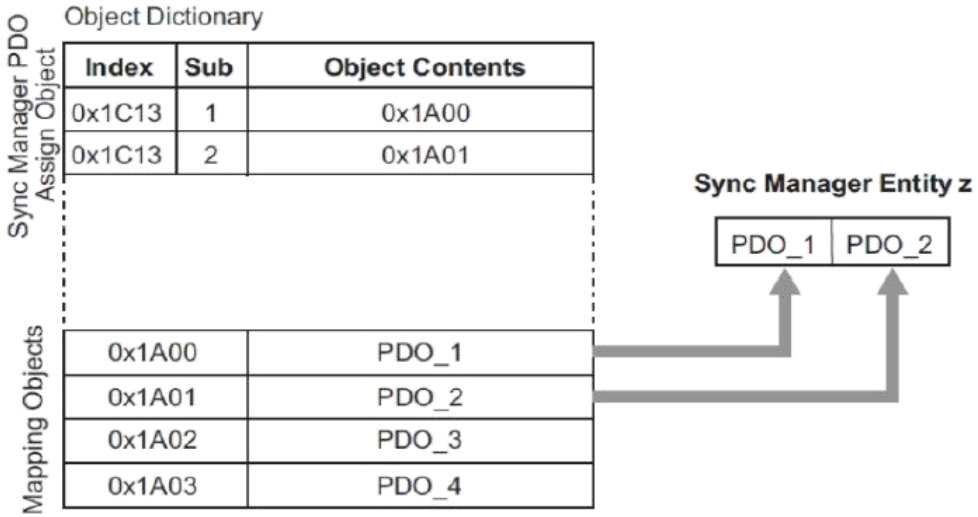
State	Description
PowerON	Power on the ESC parameter and enter the Init state
Init	There is no direct communication between the master station and the slave station on the application layer, and the master station initializes the relevant register on the data link layer to prepare for the communication at the application layer.
Pre-OP	The master communicates with the slave via mailbox (SDO), reading or writing to the slave's object dictionary, but there is no PDO data communication
Safe-OP	Master and slave can communicate by mailbox (SDO), PDO communication is limited to input data (slave update) evaluation, and output data (master update) is in a "safe" state.
Operational	PDO data input and output updates are possible.
“Init”→“Pre-OP”	The master configures the link layer address, starts the mailbox communication, initializes the DC clock synchronization, and requests to enter the Pre-OP state. The slave checks that the mailbox initialization is correct in order to respond to status requests.
“Pre-OP”→“Safe-OP”	The master configures relevant SyncManager, FMMU and mapped object addresses for PDO communication, and requests to enter the "Safe-OP" state. The slave station checks that the configuration is correct.
“Safe-OP”→“Operational”	The master outputs valid data and requests to enter the "Operational" state

7.4.6 PDO process data mapping

The process data of the slave consists of synchronization manager channel objects, each synchronization manager channel object contains multiple process data objects, among the PDO mapping object list 0x1600~0x1607 stores the RPDO mapping object index, 0x1A00~0x1A07 stores the TPDO mapping object index. In the figure below,0x1A00 stores a set of 32-bit TPDO object indexes, namely object A (8 bits), object B (8 bits), and object D (16 bits).



The synchronous manager channels (0x1C12 and 0x1C13) store lists of multiple sets of mapped objects. In the following figure, the 0X1C13 channel stores two sets of lists 0x1A00 and 0x1A01, each of which stores indexes of related PDO objects.



Chapter 8 Alarm and Troubleshooting

8.1 Alarm messages

LED of ERP light flashes red, it indicates that the drive failure alarm. As for details, please refer the code list as below. And the alarm information is hex.

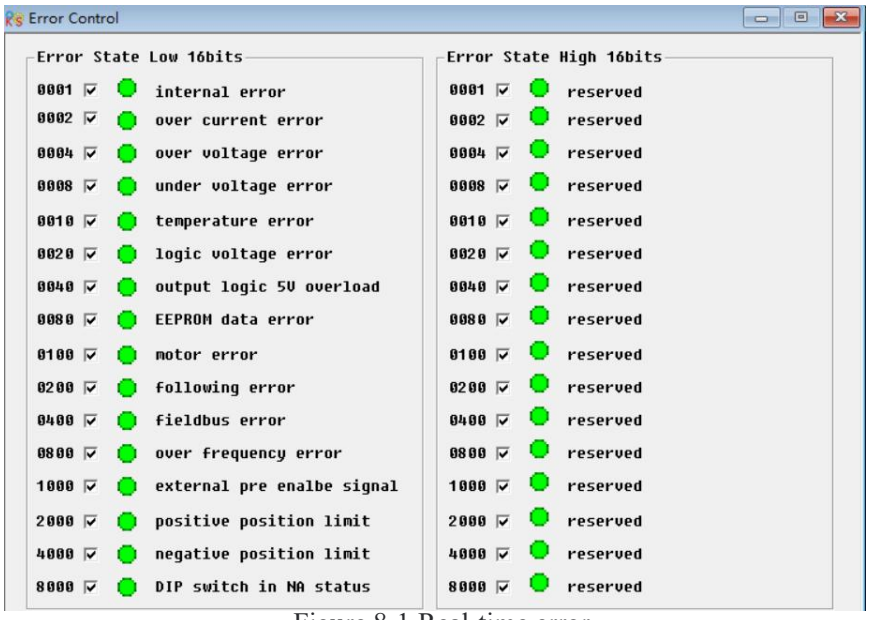


Figure 8-1 Real-time error

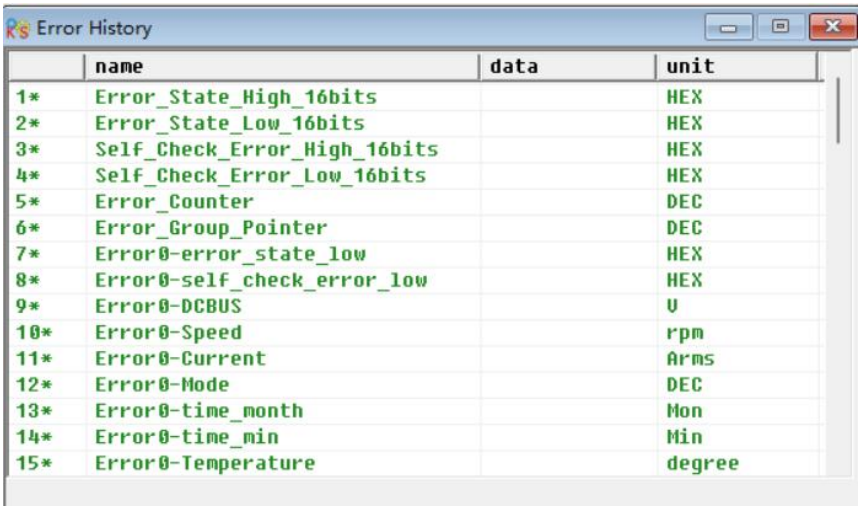
Table 8-1 Error alarm and solution

Alarm	LED		Error code	Alarm reason	Treatment measure
	RUN	ERR			
Internal Error	Slow flash	Fast flash	0x1000	1. The motor configuration is wrong 2. Drive internal problem	1.Please refer FM series field bus control stepping motor driver operating guide 2. Contact manufacturer
driver output short circuit	Extinguished	Fast flash	0x2320	1. The short circuit of Motor phase 2. Driver's problem	1. Check Motor wiring 2. Contact manufacturer
Over voltage	Fast flash	Fast flash	0x3210	1. The voltage of power supply is too high 2. high-speed stop occasion feedback energy is too high	1. Check power supplier 2. Add braking resistor
Under voltage	Extinguished	Open	0x3220	1.The voltage of power supply is too lower 2.Rapid start	1.Check power supply 2. Reduce acceleration
Over temperature	Extinguished	Slow flash	0x4210	The power module of driver is over than 80 ° C	Check the temperature is whether larger than 40° C
EEPROM Error	Fast flash /Slow flash	Open	0x6310	1. Cause for update the driver underlying firmware 2. Driver's problem	Initialize the parameters first, and save and reboot driver
Motor Error	Fast flash	Open	0x7122	1. The motor is not wired or wired incorrectly 2. the motor configuration is wrong	Check the motor wiring

Logic voltage Error	Low flash	Open	0x5111 or 0x5113	The Internal logic voltage both for 15V or 5V is out of the scope	Contact manufacturer
Overload of Output 5V current			0xFF01	The output of 5V current is too large	Check the wiring of 5V load
Following Error			0x8611	Overload or get stuck	Check load or reduce acceleration
Field bus Error			0x8100	Bus communication is closed	Check bus communication parameters
Overfrequency			0x8A80	Input pulse frequency is over the max. value.	Check whether the input pulse frequency is larger than the max. value
External pre enable signal			0x5443	External pre-enabled signals are configured for the I/O port, but no valid signals are input from the external port	Check the external wiring and confirm the input signal
Positive position limit			0x5442	The IO port is configured with a positive limit, and the driver detects a valid signal input.	
Negative position limit			0x5441	The IO port is configured with a negative limit, and the driver detects a valid signal input	

Note: the frequency both for slow flash and fast are 0.5Hz and 5Hz.

8.2 History error window



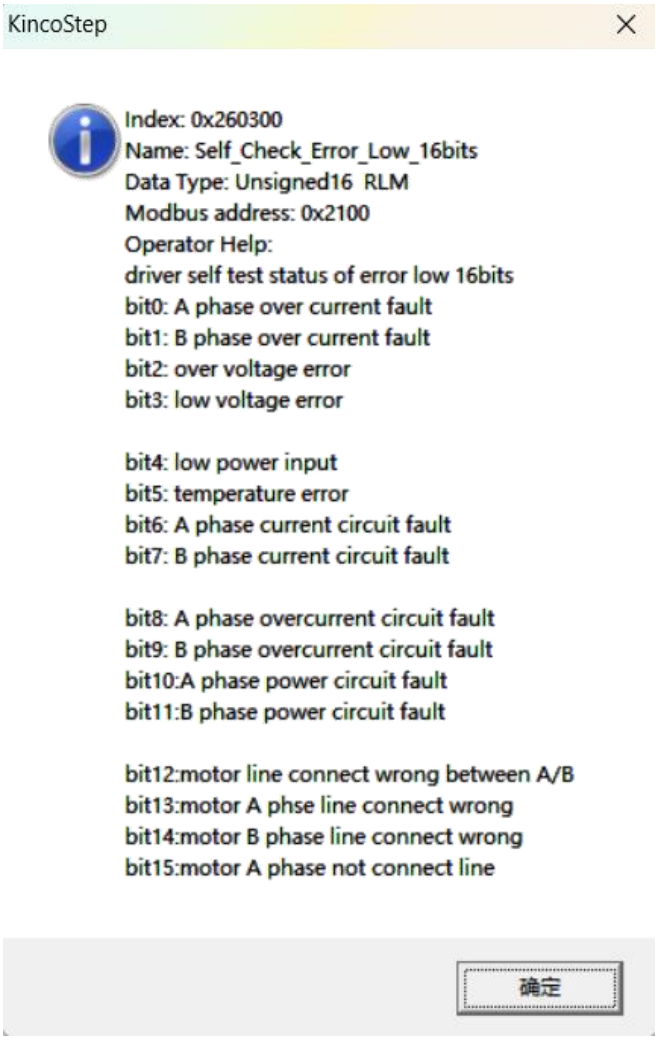
The driver provides 8 groups of historical alarm information, When the alarm occurs, the user can query the alarm code, voltage, current, temperature, speed, working mode, driver cumulative working time and other information,which is better for the user's equipment maintenance.

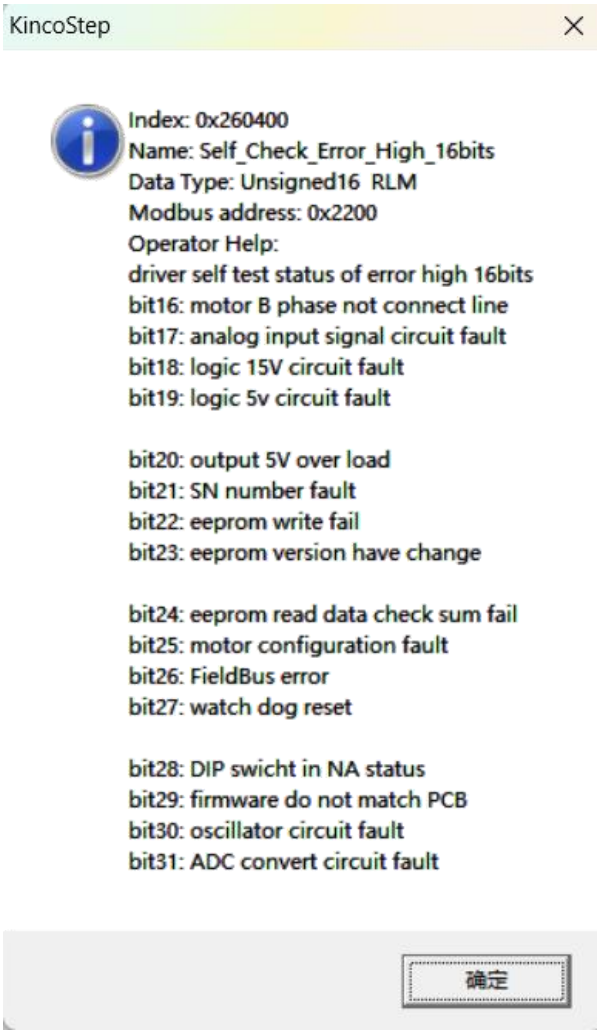
"Error status word low 16 bits/High 16 bits" is an error that occurs during the drive running, and "Self-check error status word low 16 bits/high 16 bits" is an error that occurs in the drive's power-on self-detection hardware.

The value of "Error group pointer" indicates which group the most recent error is saved in, and the value N indicates that the current alarm is in "error N". The last record is N - 1, and the next record is N + 1.8 groups of error records loop overwrite.

For example, if the current position of the error group =4, the current alarm is "Error 4". The next alarm location is "Error 5".

8.3 Self-check error status word definition





Chapter 9 Appendix

Appendix I CANopen bus communication example

CANopen communication between FM860 and Kinco F1 PLC

1．Hardware wiring

F1 PLC CAN port FM860 CAN(RJ45) port
CAN_L(2)----------CAN_L(2)
CAN_H(7)----------CAN_H(1)
GND(3)----------GND(3)

Note：

(1) It must use series connection for multiple slaves.

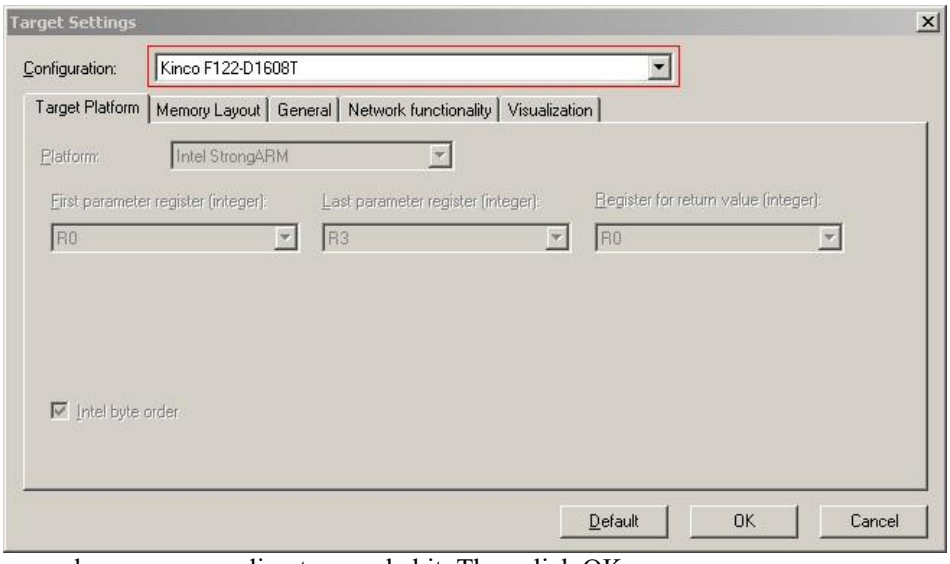
(2) CAN1 and CAN2 of F1 PLC are divided, can be used at the same time.

(3) There are terminal resistors with PLC and FM860, set by DIP switch.

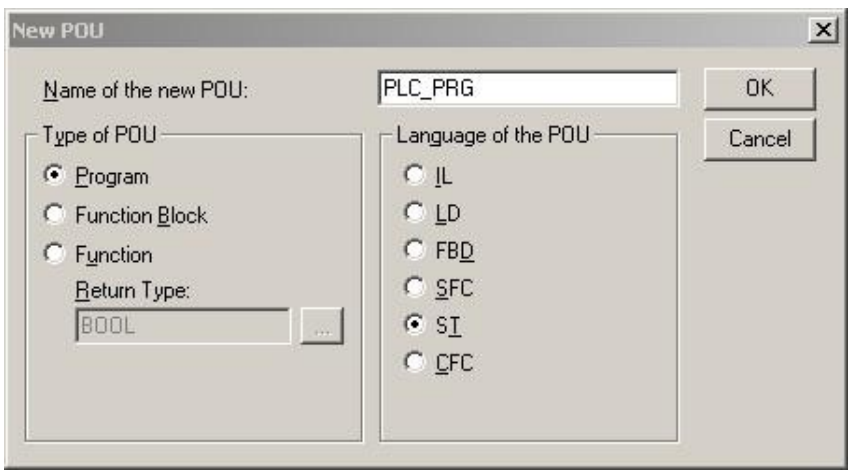
2、Parameter Setting. For FM860 baud rate and station number and so on, please refer to chapter 7.3

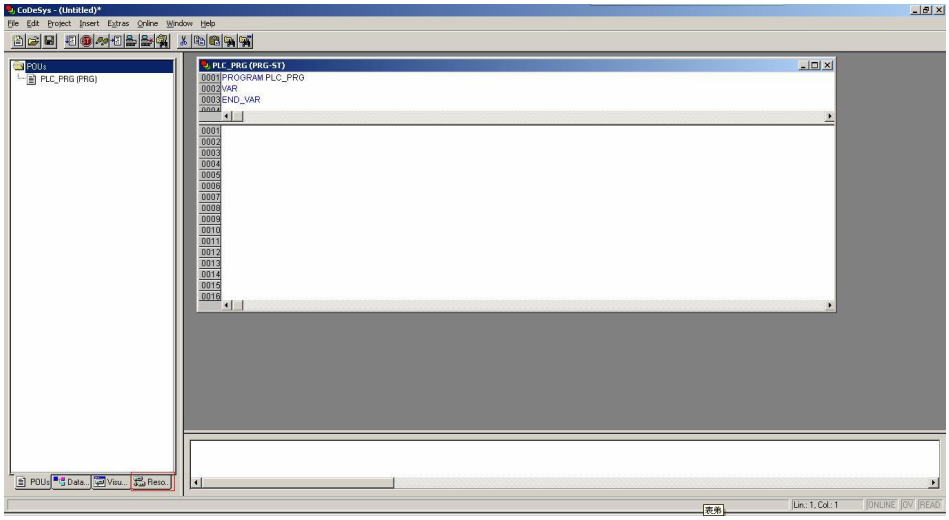
3、Software program

(1）Create a new project, select Kinco F122-D1608T and click OK.

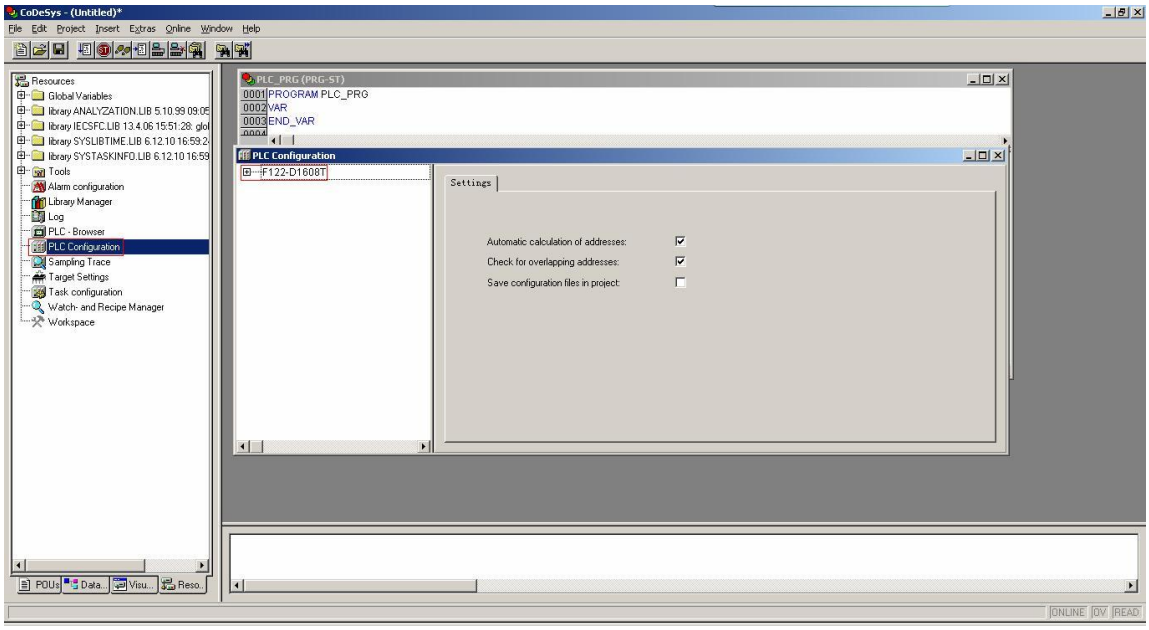


(2）Select program language according to your habit. Then click OK.。



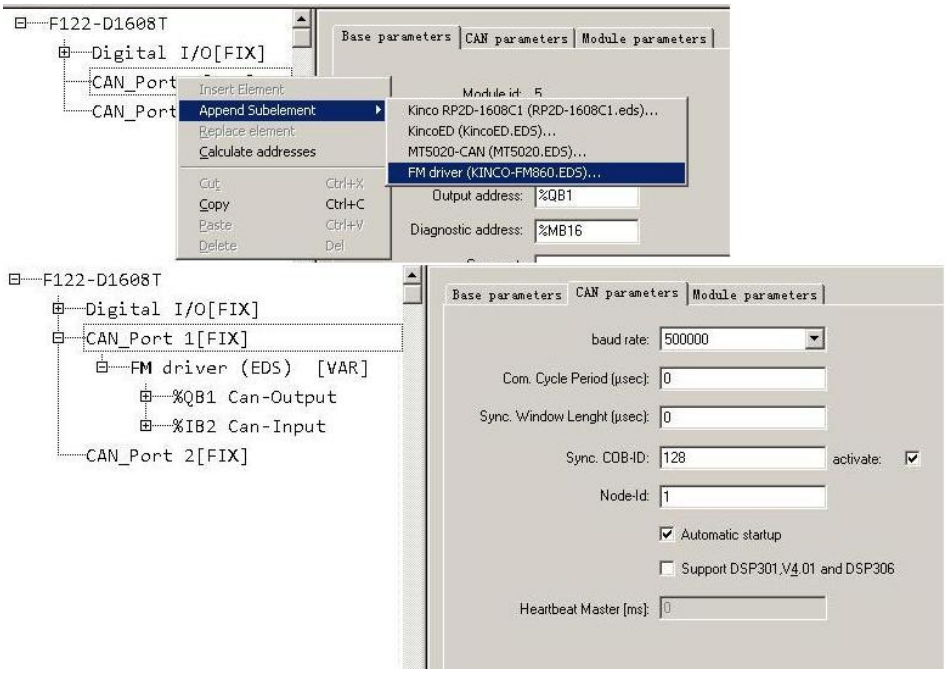


(3）It will show us a window as below after you finished step2. Then select “Resources” option and get into PLC Configuration page

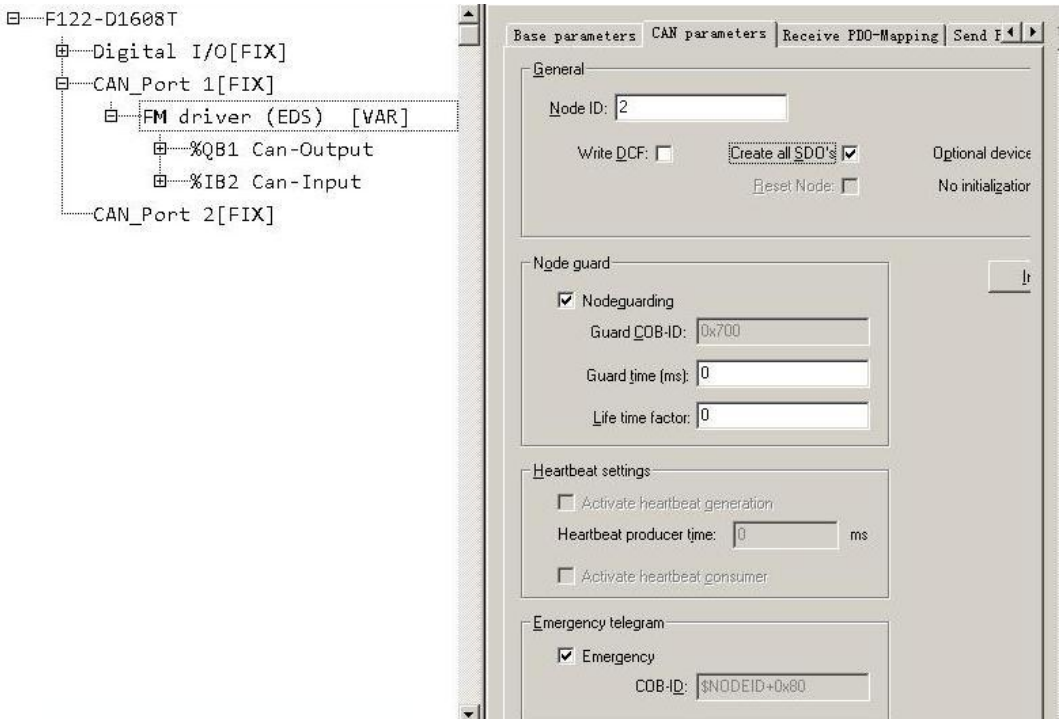


（4）Click “Extras→ add configuration file”, then show us a dialog to add EDS file of FM860.

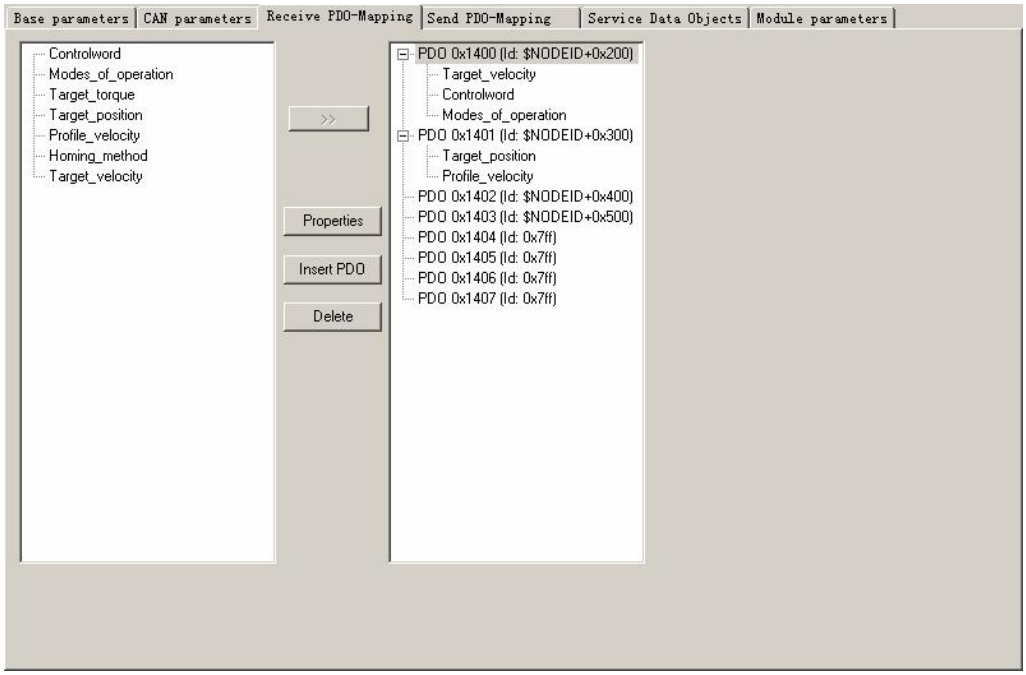
(5）There are 2 CAN ports in F1 PLC, both of them can be used as master. Set baud rate and Node-ID for CAN port. If you need synchronous message, please click “activate”, then set transmission cycle synchronization message and COB-ID in “Com. Cycle period”.

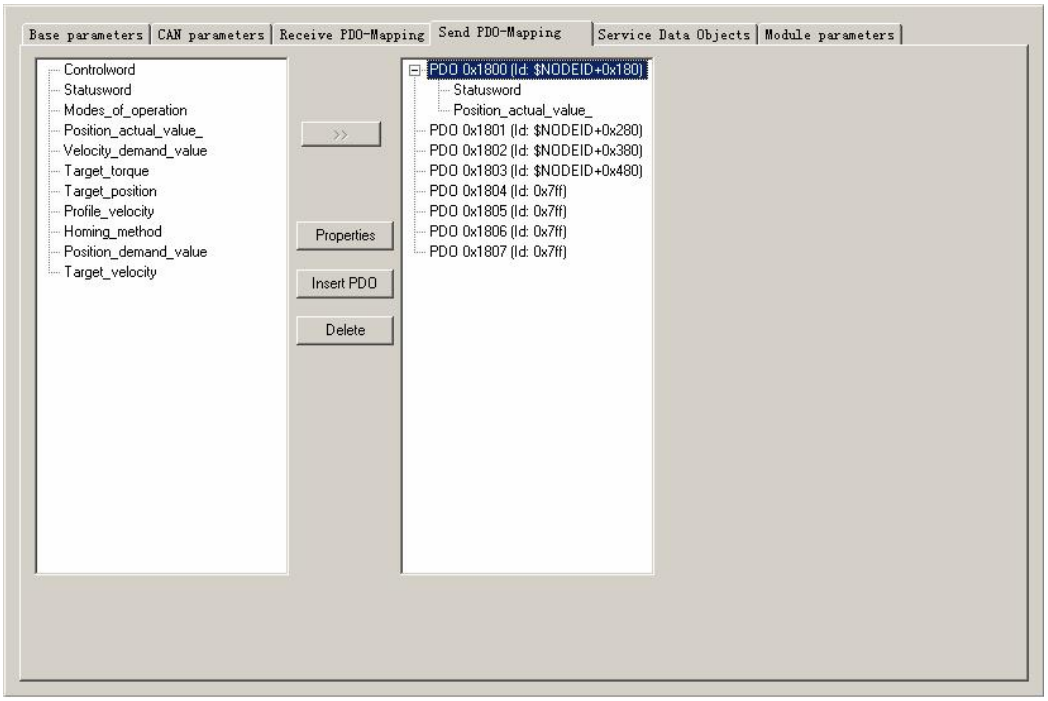


(6）Right click “CANMaster” and select “Append FM Drive” to add slaves. Then set parameters such as Node ID, Node guarding, RX-PDO and TX-PDO.

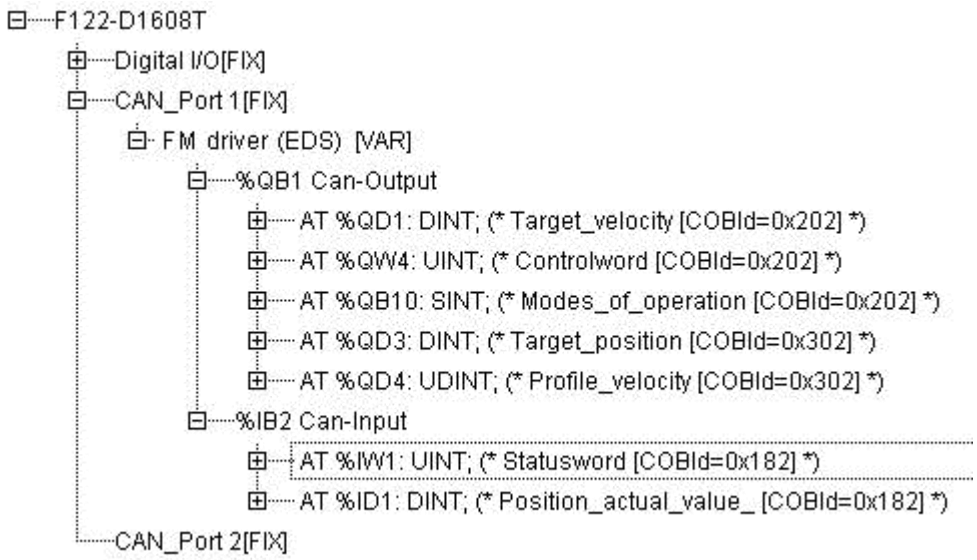


(7）Configure the PDO objects of slave according to the requirement

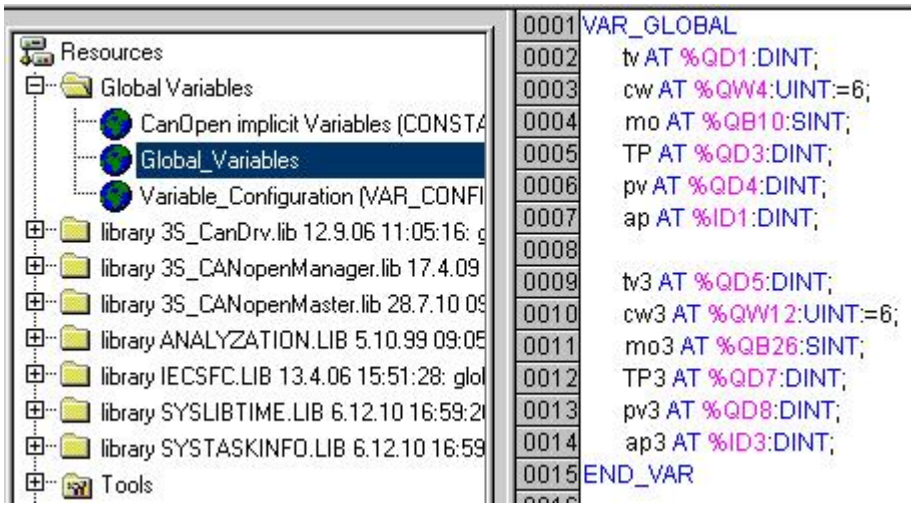




(8) After configure all the parameters, there will be all the registers corresponding to all the OD as shown in following figure. For example, the register for control word is QW4, and the register for status word is IW1.



(9）Refer to such procedure to configure several slaves, then you can start to program to control FM860. As for Variable name, can define it first for program or use the address directly.



(10）The program is as below figure. More details please refer to the chapter 6 Mode operation.(and pay attention to Chapter 7 before you do the control operation) And you have to initialize Stepping drive control word as “f” when created the communication between F1 and CAN, otherwise Stepping drive does not respond to other commands.

```
0001 PROGRAM JD2
0002 VAR
0003     m1: BOOL;
0004     m2: BOOL;
0005     spv: DINT;
0006     m3: BOOL;
0007     m4: BOOL;
0008     m5: BOOL;
0009     actual_pos: DINT;
0010     m6: BOOL;
0011 END_VAR
```

```
0001 actual_pos:=ap;
0002 (* power off*)
0003 IF m1=1 THEN
0004 tv:=27300;
0005 cw:=6;
0006 mo:=3;
0007 m1:=0;
0008 END_IF
0009 (* velocity*)
0010
0011 IF m2=1 THEN
0012 tv:=273000;
0013 cw:=47;
0014 mo:=3;
0015 m2:=0;
0016 END_IF
0017
0018 (* absolute postion*)
0019 IF m3=1 THEN
0020 tp:=0;
0021 pv:=2730000;
0022 cw:=63;
0023 mo:=1;
0024 m3:=0;
0025 END_IF
```

(11）If the objects are not in the EDS file or not commonly use, we can use SDO to read and write these objects as shown in following figure. The format is shown in the following figure. For CAN1 and CAN2 interfaces, only the wDrvNr bus interface numbers are different. CAN1 is 0 and CAN2 is 1.

```
0001 PROGRAM sdo
0002 VAR
0003     sdo1:CanOpenSendSDO;
0004     m9: BOOL;
0005     m10: BOOL;
0006     sdo2: CanOpenSendSDO;
0007     vw: ARRAY [0..7] OF BYTE;
0008     val: DWORD;
0009 END_VAR
```

```
0001 (*SDO写速度*)
0002 sdo1(      Enable:= m9,(*使能*)
0003     wDrvNr:= 0, (*总线接口号，CAN1固定为0*)
0004     ucNodeId:=2 ,(*从站站号*)
0005     wIndex:=16#60FF ,(*OD INDEX*)
0006     bySubIndex:=16#00 ,(*subINDEX*)
0007     ucModus:= 16#23, (*use 16#23 FOR 4-BYTE-write-request
0008                        use 16#27 FOR 3-BYTE
0009                        use 16#2B for 2-byte
0010                        use 16#2F for 1-byte
0011                        use 16#21 FOR downloading more than 4 bytes using the segmented transfer*)
0012     ucByte0 :=16#10,
0013     ucByte1 :=16#A8,
0014     ucByte2 :=16#29,
0015     ucByte3 :=00) ;
0016 (*SDO读实际速度*)
0017 sdo2(Enable:= m10, (*使能*)wDrvNr:= 0, (*总线接口号，CAN1固定为0*)ucNodeId:=2 ,(*从站站号*)
0018     wIndex:=16#606C ,(*OD INDEX*)
0019     bySubIndex:=16#00 ,(*subINDEX*)
0020     ucModus:=16#40);(*SDO-mode, use 16#40 for read-request.*)
0021 IF sdo2.bAnswerRec THEN
0022   val := SHL(BYTE_TO_DWORD(sdo2.ucAnswerBytes[7]),24);
0023   val := val + SHL(BYTE_TO_DWORD(sdo2.ucAnswerBytes[6]),16);
0024   val := val + SHL(BYTE_TO_DWORD(sdo2.ucAnswerBytes[5]),8);
0025   val := val + BYTE_TO_DWORD(sdo2.ucAnswerBytes[4]);
0026 END_IF
```

CANopen communication between FM860 and Peak CAN

Peak's CAN adapter has ISA, PCI, USB-CAN and other products,which provides Windows 98/ME and Windows2000/XP device drivers (*.vxd and *.sys) and dynamic connection library (*.DLL), supported software VB, VC, Delphi and BCB and so on. This example is to use PCAN-USB connected to FM860.

Other software programming can refer to the operation of this example!

(1)Please refer PCAN-USB hardware manual to install.

(2)Wiring connecting

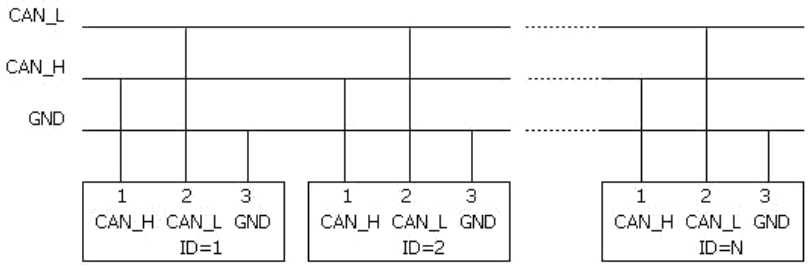


Figure 9-1 Several FM860 communication

It need to add a 120-150ohm resistor between CAN_L and CAN_H, and can set the DIP switch as SW9 to connecting the resistor.

(3)Set the communication parameters such as baud rate, ID address, default as 500K and 1 respectively. Need to save and reboot after updated.

4）After completing the above steps, the FM stepper driver can be controlled by referring to the CANopen communication protocol. Since the PCAN-View software cannot import EDS files, it is difficult to operate PDO. In this example, various modes of FM stepping are controlled according to the data format stipulated by the communication protocol. (Please refer to Chapter 7 before performing controls).

Following figure is the example to send command to set 6040 as 3F. the lower part of the figure is send data, upper part of the figure is receive data.

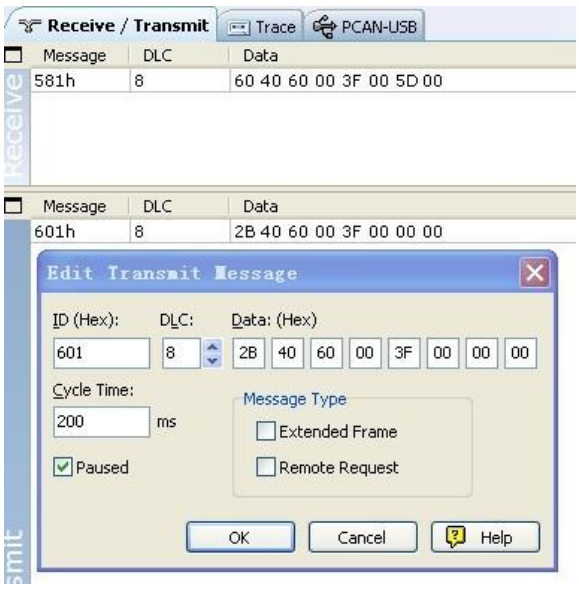


Table 9-1 Sending and receiving data packets in Homing mode

Homing control mode(the control word should change from F to 1F)				
CANopen address	Name	Value	Send or Return message（ID=1）	Note
60600008	Operation mode	0x6	601 2F 60 60 00 06 00 581 60 60 60 00 06 00	Speed RPM have to Change as the unit of DEC. DEC=[(RPM*512*60000)/1875]
60980008	Homing mode	0x14	601 2F 98 60 00 14 00 581 60 98 60 00 14 00	
60990120	Turning signal speed of Homing	200RPM	601 23 99 60 01 00 00 32 00 581 60 99 60 01 00 00 32 00	
60990220	Homing signal speed	150RPM	601 23 99 60 02 00 80 25 00 581 60 99 60 02 00 80 25 00	
60400010	Contorl word	0xF	601 2B 40 60 00 0F 00 581 60 40 60 00 0F 00	
60400010	Contorl word	0x1F	601 2B 40 60 00 1F 00 581 60 40 60 00 1F 00	
601 40 41 60 00 00 00 00 00		Read status word, 9437means found homing		

Table 9-2 Sending and receiving data packets in Position control mode

Position control mode(Absolute positioning of control word is changed from 2F to 3F, relative positioning is changed from 4F to 5F. 103F means activate immediately when position change)				
CANopen address	Name	Value	Message（ID=1）	Note
60400010	Control word	0xF	601 2B 40 60 00 0F 00 581 60 40 60 00 0F 00	Speed RPM need change to DEC DEC=[(RPM*512*60000)/1875] Profile Acce and profile dece is defaulted as DEC. DEC=[(RPS/S*65536*60000)/1000/4000]
60600008	Operation mode	0x1	601 2F 60 60 00 01 00 581 60 60 60 00 01 00	
607A0020	Target-position	50000 DEC	601 23 7A 60 00 50 C3 00 00 581 60 7A 60 00 50 C3 00 00	
60810020	Profile velocity	200RPM	601 23 81 60 00 00 00 32 00 581 60 81 60 00 00 00 32 00	
60830020	Profile_Acce	10rps/s	Default value	
60840020	Profile_Dece	10rps/s	Default value	
60400010	Control word	0x2F	601 2B 40 60 00 2F 00 581 60 40 60 00 2F 00	
		0x3F(Absolute positioning)	601 2B 40 60 00 3F 00 581 60 40 60 00 3F 00	
		0x4F	601 2B 40 60 00 4F 00 581 60 40 60 00 4F 00	
		0x5F(Relative positioning)	601 2B 40 60 00 5F 00 581 60 40 60 00 5F 00	
601 40 41 60 00 00 00 00 0 Read status word, D437 means target position reach.				

Table 9-3 Sending and receiving data packets in Speed control mode

Speed control mode				
CANopen address	Name	Value	Message（ID=1）	Note
60600008	Operation mode	0x3	601 2F 60 60 00 03 00 581 60 60 60 00 03 00	Speed RPM need change toDEC. DEC=[(RPM*512*60000)/1875] Profile Acce and profile dece is defaulted as DEC. DEC=[(RPS/S*65536*60000)/1000/4000]
60FF0020	Target speed	150RPM	601 23 FF 60 00 00 80 25 00 581 60 FF 60 00 00 80 25 00	
60400010	Control word	0xF	601 2B 40 60 00 0F 00 581 60 40 60 00 0F 00	
60830020	Profile Acceleration	10rps/s	Default value	
60840020	Profile Deceleration	10rps/s	Default value	

Note:

The transmission formatting of all data is following Hexadecimal in communication mode.

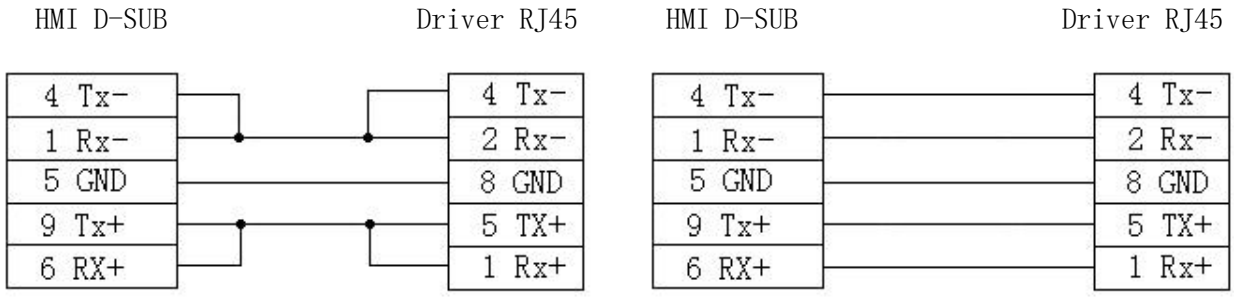
Appendix II RS485 communication example

Modbus Communication between FM860 and KINCO HMI

1.All series of Kinco HMI can be connected with the RS485 serial port of the FM driver. This example only introduces the main product of Kinco HMI MT4000, and MT5000 series is connected with the stepper driver. For other series of products, please refer to the user manual of each product or consult Kinco technical personnel.
2.This example only introduces simple connection, users can go to the Kinco website "Download center - Data download - sample program" download "Kinco and FM stepper driver and 485 communication example" for detailed application.

（1）HMI control single FM Drive

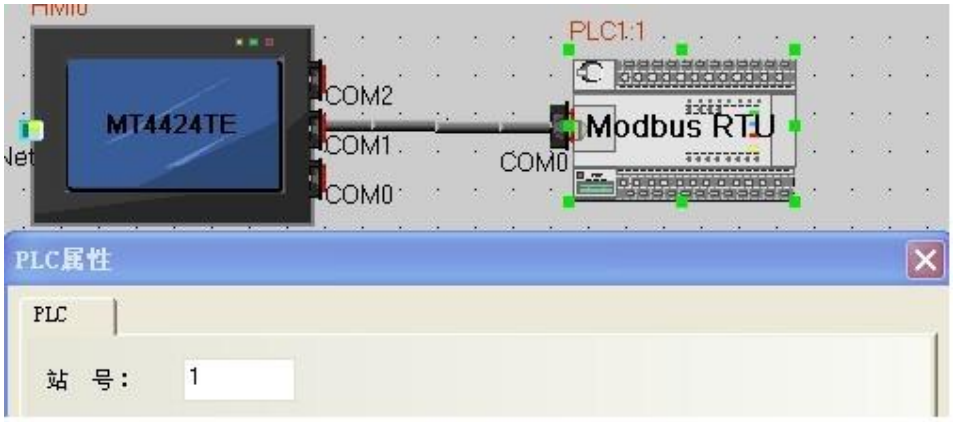
a. Wiring diagram



RS485 half-duplex communication　　RS485 full-duplex communication

b. Communication parameter setting

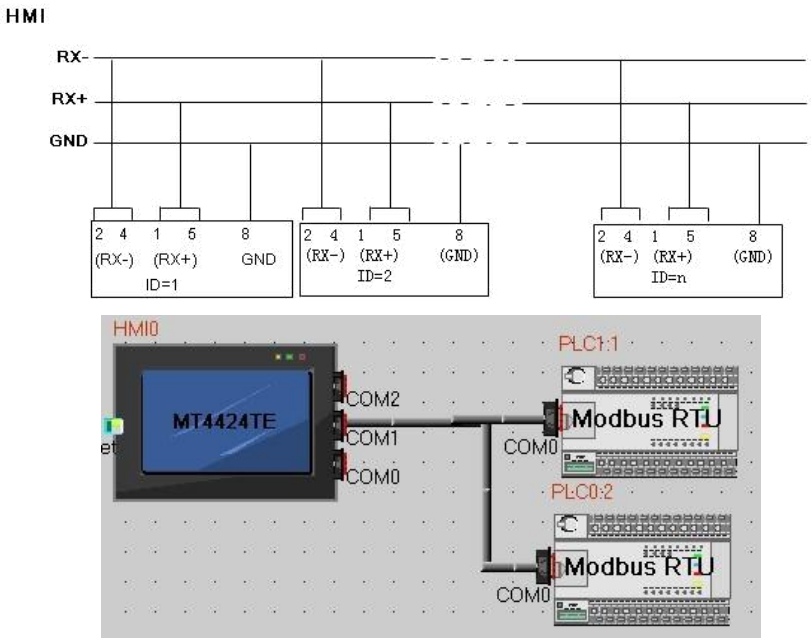
HMI select Modbus RTU driver, HMI communication parameter Settings are shown in figure, it should be noted that the PLC station number is the ID number of FM driver, FM driver factory ID default is 1, so the HMI control a single FM stepper PLC station number set to 1. If the FM ID number is set to N, then the following PLC station number should also be set to N. FM baud rate, station number and other parameters are set according to the 485 communication protocol chapter.



c. Address setting

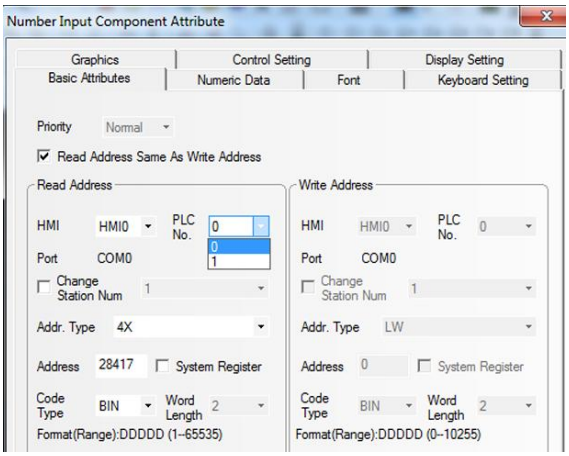
It need use address type 4X in HMI program (all the objects of FM are corresponding to 4X). According to Modbus address of objects in the Common Object List, the Modbus address of the object Target velocity (60FF0020) is 0x6F00, its decimal value is 28416. when we use this address in HMI, we need to add 1, so in HMI the address for Target velocity is 28417 as following figure.

(2) HMI control multiple FM drive



b. Communication parameter setting

The parameters setting in HMI is same as such example. The difference is different station number for different FM Drive. In the attribute of components of HMI. It needs to select the PLC No. for different FM drive. (The PLC No. is not the drive station No. as shown in the figure above. PLC0:2 means the PLC No. as 0, and station No. as 2) .



Communication between FM860 and Modbus

FM stepper RS485 communication port supports Modbus RTU communication protocol, and users can write programs to control FM stepper driver according to modbus protocol through VB, VC and other software. The following uses the common serial Modbus debugging tool software to control the FM stepper driver through the Modbus protocol. Other software programming can refer to the operation of this example!

1．Wiring diagram

PC has to use RS232-RS485 module to connect FM drive.

RS485 module terminal		FM driver RS485（X1）
D-	————————————	2\4(RX-、TX-)
D+	————————————	1\5(RX+、TX+)
GND	————————————	8 (GND)

2. Set the ID address and baud rate by referring to the serial port communication section. The default baud rate is 19200 and the station number is 1. Need to save and reboot if updated.

3．After completing the above steps, you can refer to the Modbus RTU communication protocol to control the FM stepper driver. (Please refer to Chapter 7 before proceeding control)

As below figure read object 0x60400010 control words.

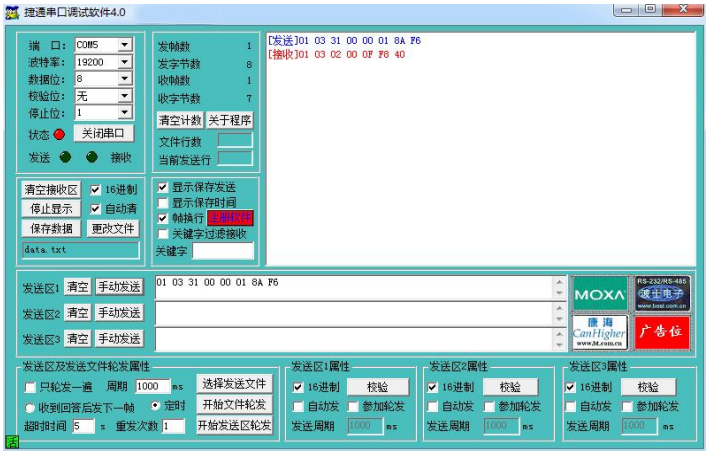


Table 9-4 Sending message in homing and position control mode

Homing control mode(the control word should change from F to 1F)				
CANopen address	Name	Value	Send or Return message（ID=1）	Note
0x3500	Operation mode	0x6	01 06 35 00 00 06 06 04 01 06 35 00 00 06 06 04	Speed RPM have to Change as the unit of DEC. DEC=[(RPM*512*60000)/1875]
0x4D00	Homing mode	0x14	01 06 4D 00 00 14 9E A9 01 06 4D 00 00 14 9E A9	
0x5010	Turning signal speed of Homing	200RPM	01 10 50 10 00 02 04 00 00 00 32 8F 75 01 10 50 10 00 02 51 0D	
0x5020	Homing signal speed	150RPM	01 10 50 20 00 02 04 80 00 00 25 E5 AF 01 10 50 20 00 02 51 02	
0x3100	Contorl word	0xF	01 06 31 00 00 0F C7 32 01 06 31 00 00 0F C7 32	
0x3100	Contorl word	0x1F	01 06 31 00 00 1F C6 FE 01 06 31 00 00 1F C6 FE	
01 03 32 00 00 01 8A B2		Read status word, 9437 means found homing		
Position control mode(Absolute positioning of control word is changed from 2F to 3F, relative positioning is changed from 4F to 5F. 103F means activate immediately when position change)				
CANopen address	Name	Value	Send or Return message（ID=1）	Note
0x3100	Control word	6	01 06 31 00 00 06 07 34 01 06 31 00 00 06 07 34	Speed RPM need change to DEC DEC=[(RPM*512*60000)/1875] Profile Acce and profile dece is defaulted as DEC. DEC=[(RPS/S*65536*60000)/1000/4000]
0x3500	Operation mode	1	01 06 35 00 00 01 47 C6 01 06 35 00 00 01 47 C6	
0x4000	Target position	50000DEC	01 10 40 00 00 02 04 C3 50 00 00 FE 39 01 10 40 00 00 02 54 08	
0x4A00	Profile velocity	200RPM	01 10 4A 00 00 02 04 00 00 00 32 3D 19 01 10 4A 00 00 02 57 D0	
0x4B00	Profile_Acce	10rps/s	Default value	
0x4C00	Profile_Dece	10rps/s	Default value	
0x3100	Control word	2F	01 06 31 00 00 2F C6 EA 01 06 31 00 00 2F C6 EA	
		0x3F(Absolute positioning)	01 06 31 00 00 3F C7 26 01 06 31 00 00 3F C7 26	
		0x4F	01 06 31 00 00 4F C6 C2 01 06 31 00 00 4F C6 C2	
		0x5F(Relative positioning)	01 06 31 00 00 5F C7 0E 01 06 31 00 00 5F C7 0E	

Table 9-5 Sending message in speed mode

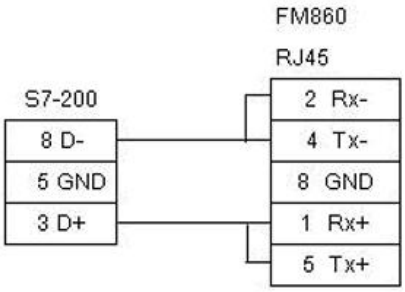
Speed control mode				
CANopen address	Name	Value	Send or Return message（ID=1）	Note
0x3500	Operation mode	3	01 06 35 00 00 03 C6 07 01 06 35 00 00 03 C6 07	Speed RPM need change toDEC. DEC=[(RPM*512*60000)/1875] Profile Acce and profile dece is defaulted as DEC. DEC=[(RPS/S*65536*60000)/1000/4000]
0x6F00	Target speed	150RPM	01 10 6F 00 00 02 04 80 00 00 25 F2 46 01 10 6F 00 00 02 5C DC	
0x4B00	Profile_Acce	10rps/s	Default value	
0x4C00	Profile_Dece	10rps/s	Default value	
0x3100	Control word	F	01 06 31 00 00 0F C7 32 01 06 31 00 00 0F C7 32	

Note: Data is transmitted in hexadecimal format in communication mode.

Modbus Communication Between FM860 and Siemens S7-200

1. Wiring diagram

Signal connecting is as below.



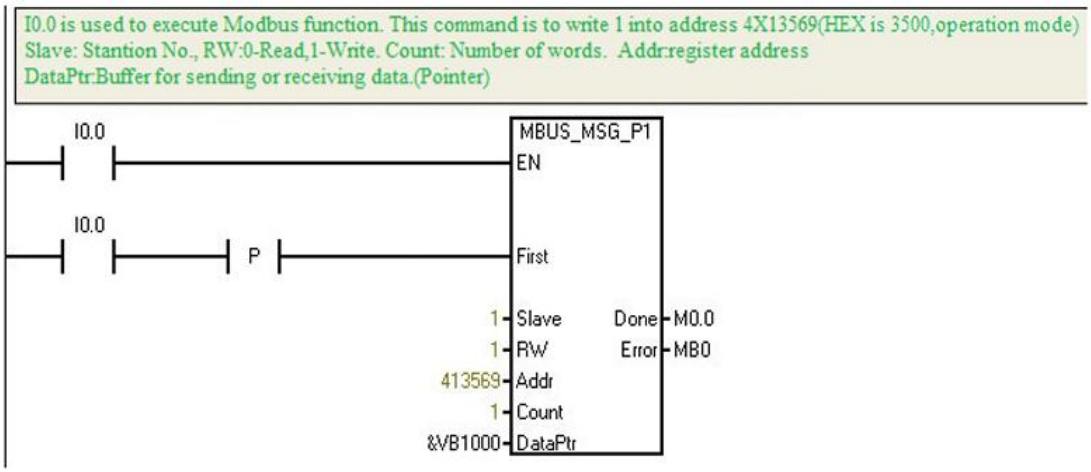
2.Parameter Setting.

FM stepper driver parameters are set by referring to the 485 Communication section. The default parameter is Modbus RTU, baud rate 19200, no check

In the software of S7-200 PLC, there is a library function used to set communication parameters as shown in following figure.

3.Program

It use Modbus function (MODBUS_MSG) to send and receive data. The descriptions of Modbus function are shown in following figure.



4.Example program description

Table 9-6 Case

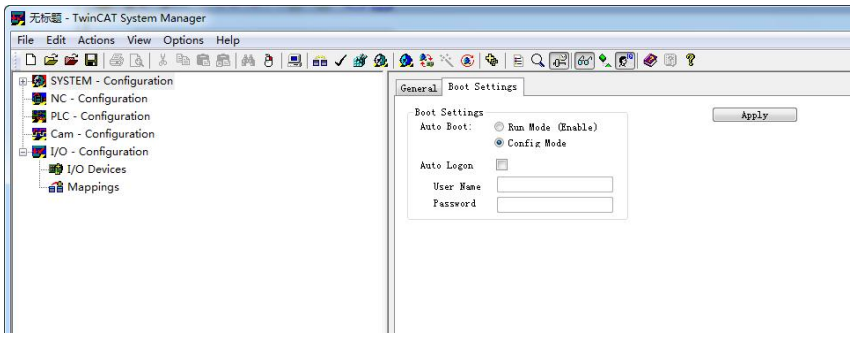
S7200 PLC input port	Function	Explanation
I0.0	Write 60600008=1	Set position mode
I0.1	Write 607A0020=10000	Set the target position
I0.2	Write 60810020=1000rpm	Set the target speed in position mode
I0.3	Write 60400010=0x4F then write 0x5F	Control command = relative motion
I0.4	Read 60630020	Read motor position
I0.5	Read 60410010	Read driver status word

Appendix III EtherCAT bus communication example

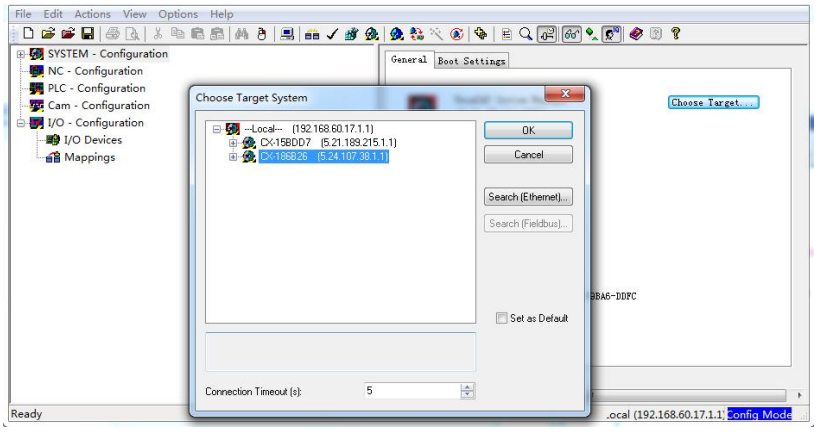
Communication between FM880 and Beckhoff PLC

1. Before opening the software, copy the device description XML file of FM880 to the installation directory of Twincat, the default path is C:\TwinCAT\Io\EtherCAT. The XML file can be downloaded at www.kinco.cn.

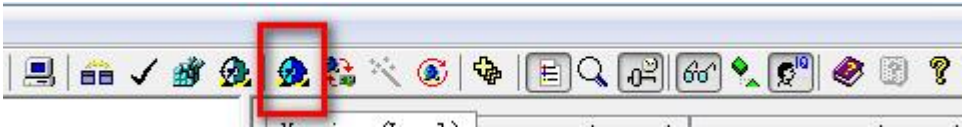
2. Open the Twincat System Manager software.



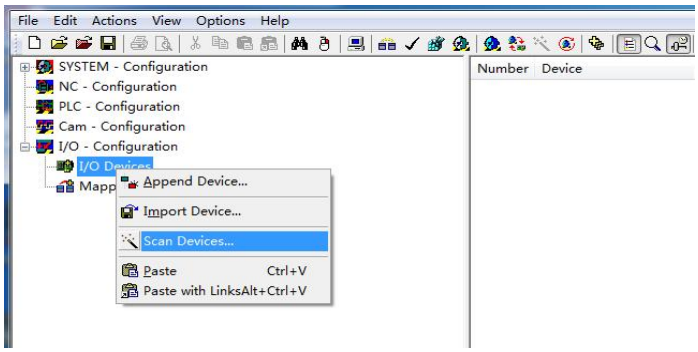
3. Configure the controller hardware equipment, select the current use of PLC model. Here we use the CX5020 controller from Beckhoff, select it and click OK. If the device model is not displayed, run the Search operation to scan for it and add it to the system.



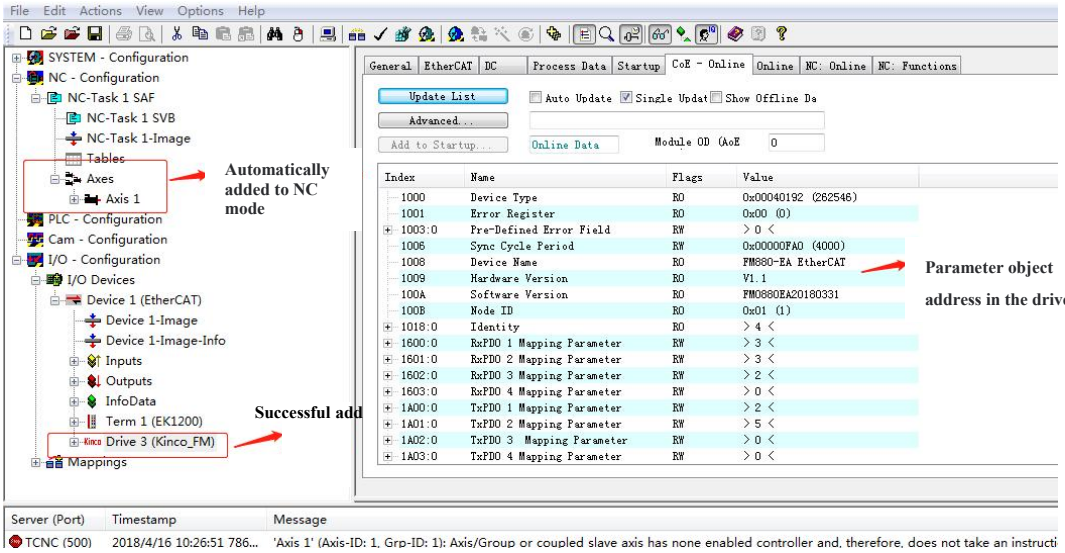
4.After the controller communicates with the Twincat software, the status bar of the software will display "Config Mode" or "RTime". If it is "Timeout", it indicates that the controller and the Twincat software cannot be connected. The controller configuration parameter needs to be in config mode. Click the icon below to switch the controller to config mode.



5. Right-click on I/O Devices, click Scan Devices, Scan for Ethercat's slave station, select the network port connected to the drive, and click Scan boxes. Scan to the end and click automatically add to NC mode. If the controller Ethercat interface is properly connected to the Kinco stepper drive, the box (Kinco_FM) can be seen below the device and has been automatically associated with the controller's NC mode.



Scan result：

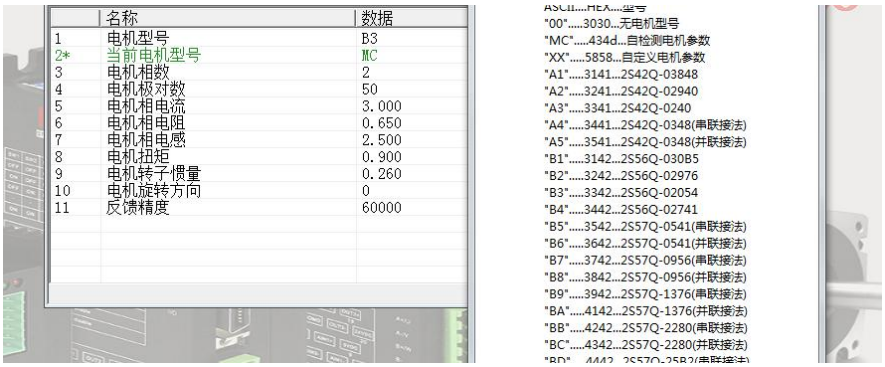


6.Click on the slave drive "Kinco Drive3 (Kinco_FM)", you can see a COE-Online on the right (as shown in the picture above), here you can set the parameters of the stepper drive. The parameters are set in the following steps:

Note: If the operation is correct, but no corresponding parameter table is found in COE-Online, check whether the XML file corresponding to the stepper drive is in the correct folder. If the file exists, the fault occurs because the configuration software is opened before the XML file is placed in the corresponding folder. You need to open the software again and perform the preceding steps.

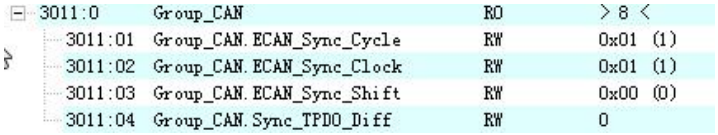
（1）. Configure the motor parameters of the stepper driver. As shown in the following figure, it is only necessary to fill in the hexadecimal corresponding motor code in 6410:01, and the corresponding motor parameters will be matched after filling. The corresponding motor code can be queried in the motor chapter of the stepper driver user manual, or in the upper computer software, motor configuration page, right-click the "motor model" object to view.

6410:0	Group Motor	RO	> 29 <
6410:01	Motor Num	RW	0x3342 (13122)
6410:03	Feedback Resolution	RO	0x0000EA60 (60000)
6410:05	Motor Poles	RW	0x32 (50)
6410:0B	Motor Phase Current	RW	0x001E (30)
6410:0C	Motor L	RW	0x0019 (25)
6410:0D	Motor R	RW	0x0041 (65)
6410:0E	Motor Ke	RW	0x0122 (290)
6410:0F	Motor Tq	RW	0x005A (90)
6410:10	Motor Jr	RW	0x0104 (260)
6410:12	Brake Delay	RW	0x0096 (150)
6410:13	Motor Rot Direction	RW	0x00 (0)
6410:14	Motor NumRD	RW	0x3342 (13122)

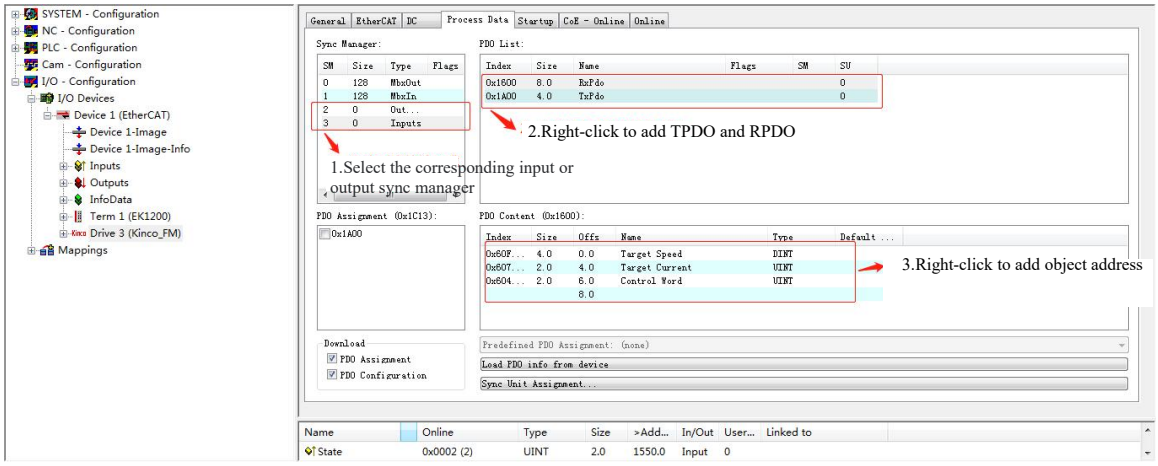


For example, in the figure above, the current motor model is 2S56Q-02054, and the corresponding motor code is B3 (ASIIC code), which is converted to hexadecimal 0x3342. Pay attention to the rules of conversion!

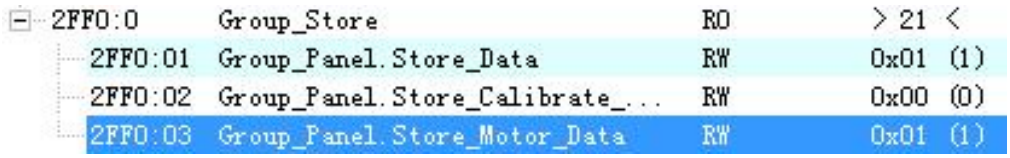
(2).Set the synchronization period of the stepper drive to 301101 and ensure that the synchronization clock mode is enabled (301102 is 2). A synchronization period of 2 milliseconds (with a 301101 value of 1) or 4 milliseconds (with a 301101 value of 2) is recommended



(3).Configure mapping objects for RPDO and TPDO



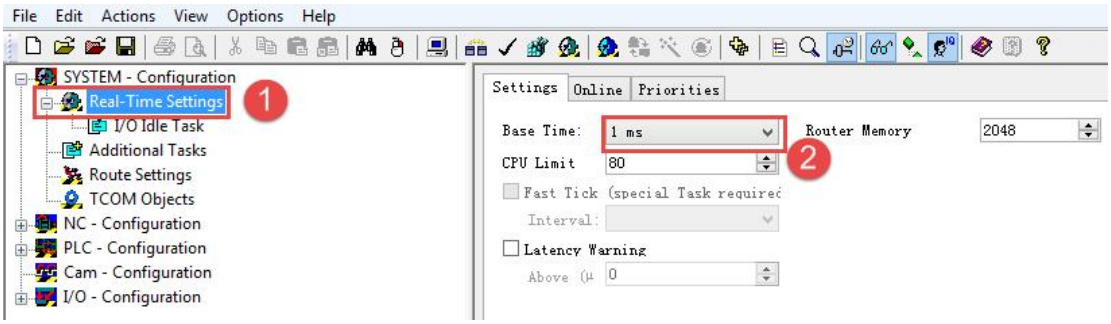
After setting the parameters, if it is the first time to set, you need to save the corresponding control parameters (2FF001 set to 1) and motor parameters (2FF003 set to 1), power off and restart the drive to be effective.



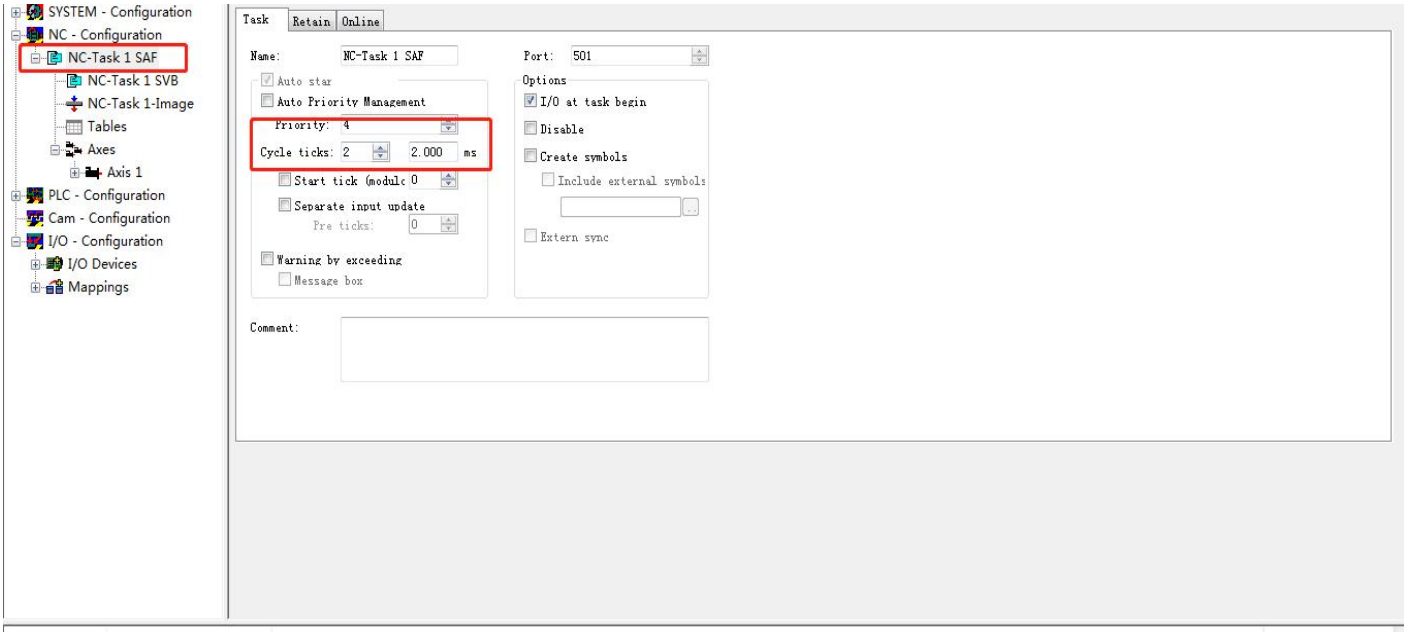
Note: The above two storage parameter addresses will automatically change back to 0 after writing 1 for a few seconds. This is a normal phenomenon, indicating that the storage operation has taken effect, and the initial value is returned.

7. Set the NC mode parameters

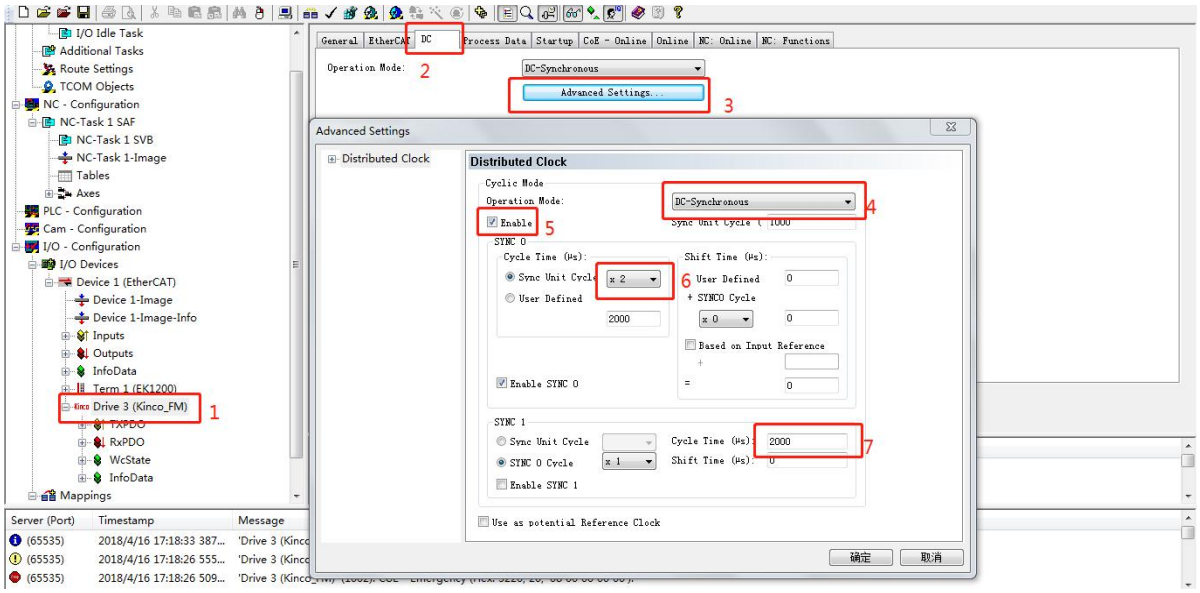
A.Set the basic scan time of the system to 1ms



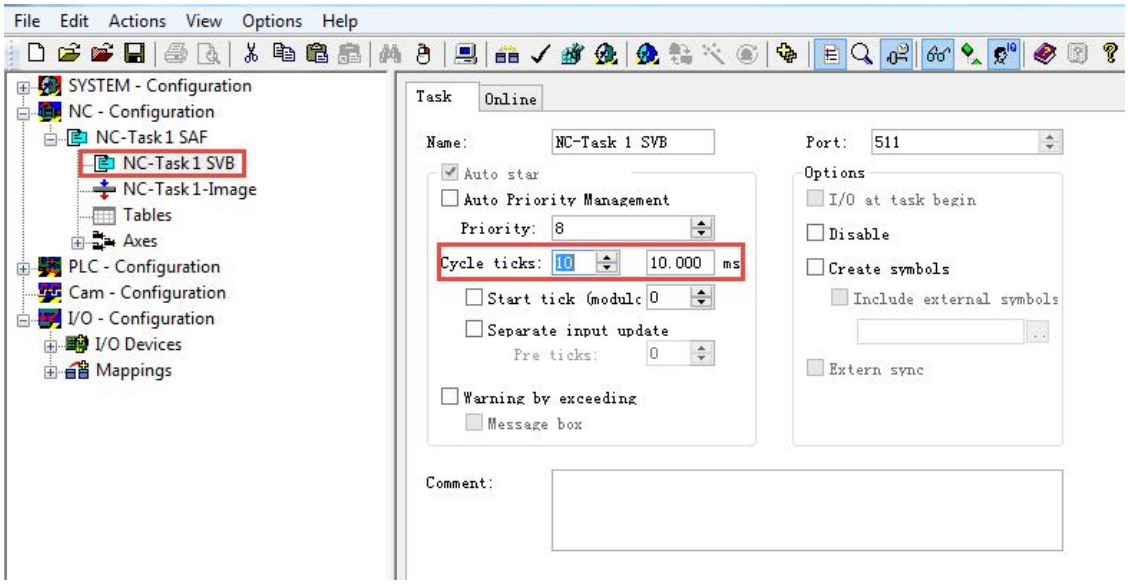
B.Set the control period of the NC PTP on the automatically connected NC Task.The NC period refers to the period of the NC Task SAF, which is set to 2ms. In this task, TwinCAT NC completes the generation and calculation of position, speed, and acceleration setpoints, and determines the direction.



C. Check the clock synchronization enable, pay attention to the Cycle Time setting, to be consistent with the synchronization period in the stepper drive (the value of 301101 mentioned above) set, such as: the synchronization period in the stepper drive is set to 1 (2ms), then the controller NC mode should also be set to 2ms. This cycle inconsistency will cause the motor to run jitter and error.

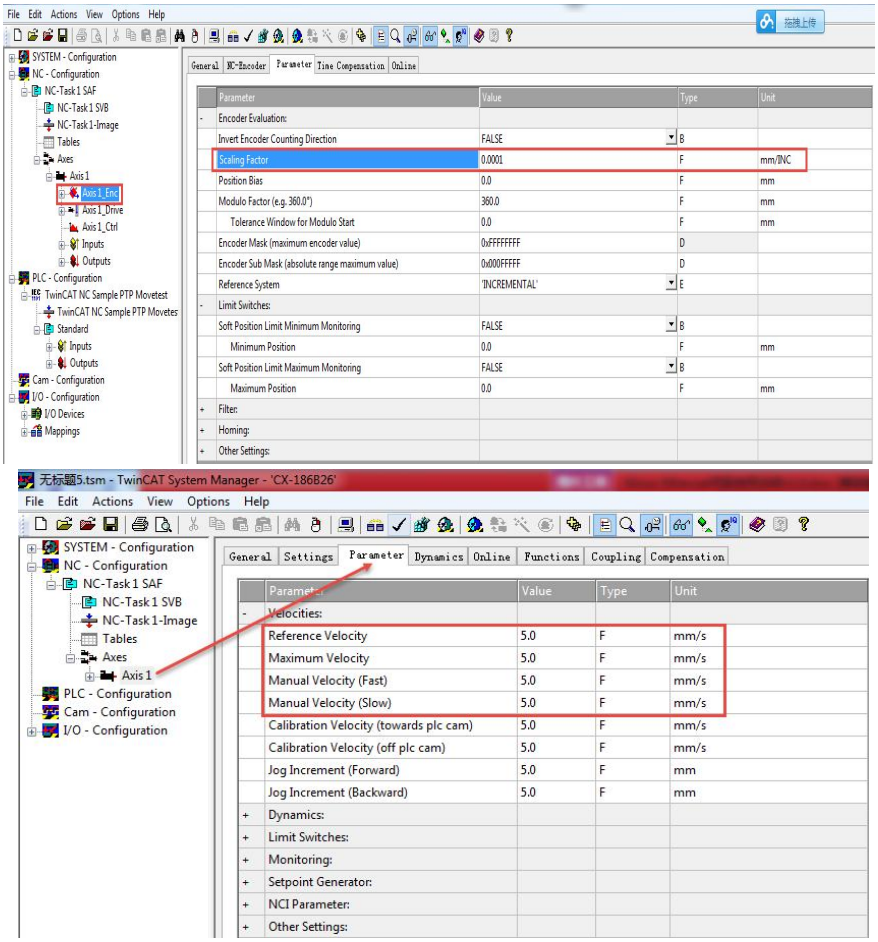


D. The other NC period is the NC Task SVB period, which is typically 10ms. In this task, TwinCAT NC completes global path planning and checks whether the planning conditions are reasonable. TwinCAT NC tasks have higher priority than TwinCAT PLC .



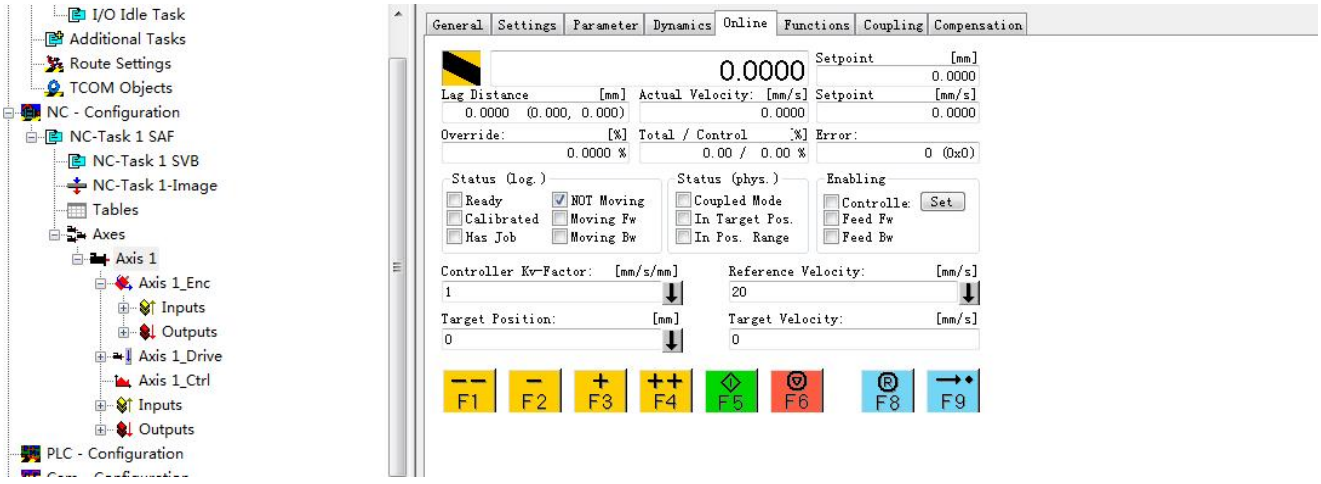
E. Set the unit of NC operation in the NC axis. In Axix 1_ENC, you can set Scaling Factor, that is, the corresponding distance of encoder pulse for each position feedback, for example: If the KINCO stepper motor outputs 60000 pulses per turn, and if the motor rotates 6mm per turn, the Scaling Factor should be 6/60000=0.0001mm/Inc. At this time, if the target position needs to be increased by 12mm, that is, the actual position of the stepper motor should be increased by 120000INC. Control PLC to send the corresponding pulse. In general, it is also necessary to set the speed and other parameters of the corresponding NC control in Axis to ensure safe operation.

Note: The FM880 drive sets the motor feedback accuracy to 60000 by default and cannot be modified.



As shown in the figure above, the speed is set to 5.0mm/S, then: stepper motor speed =5.0/0.0001=50000inc/S=300RPM

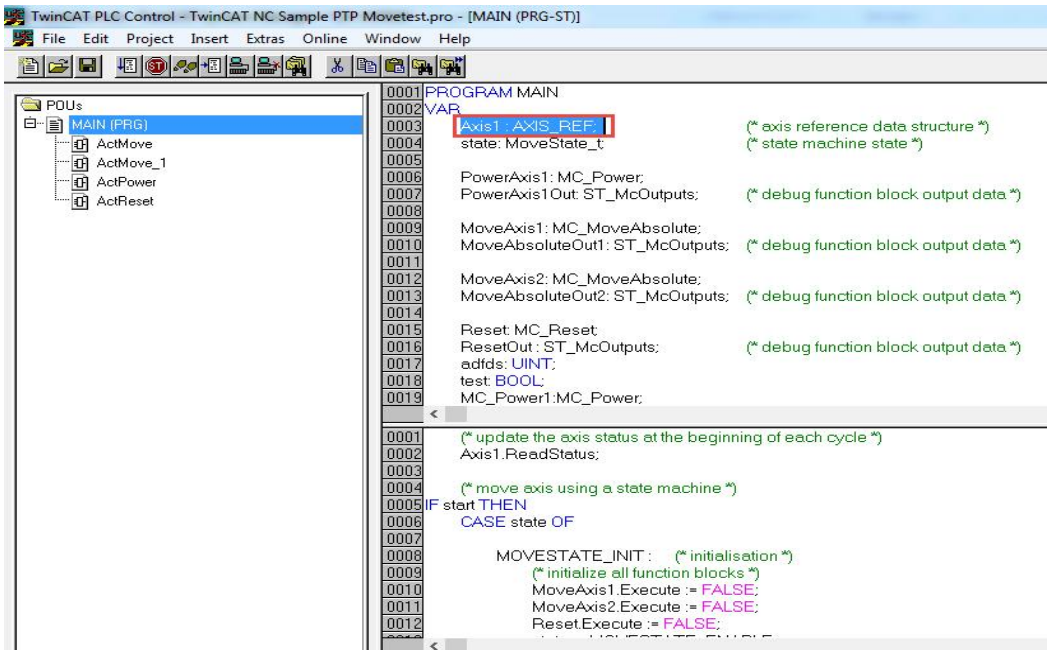
The above is to complete the basic setup of the Kinco Ethercat stepper driver in Beckhoff NC mode, and the rest is to control the motor operation through the controller's NC debugging interface or programming.



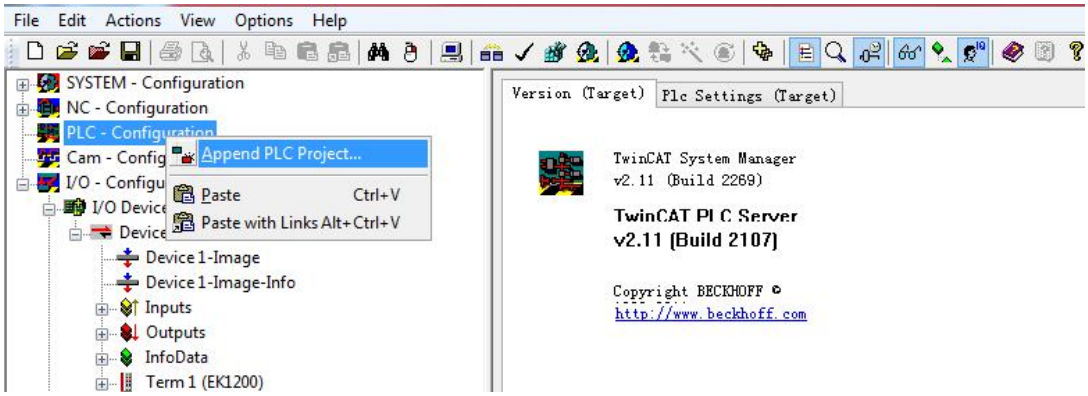
8. Add PLC project:

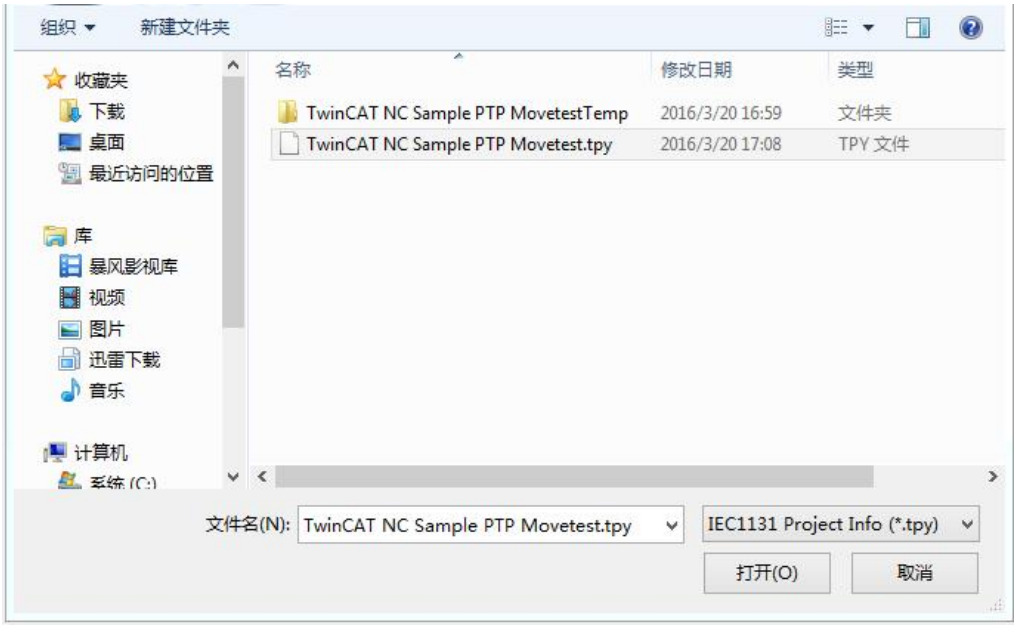
Open TwinCAT PLC Control software, create a new PLC project, and declare Axis 1 as the axis structure variable and other Motion function blocks to be used in the PLC program.

Axis1 : AXIS_REF;

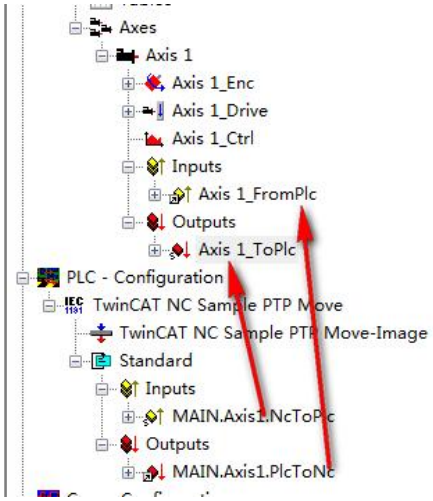


Write a program to save the compilation will generate a .tpy file. Go back to Twincat System Manager and import the file into the system, as shown below:

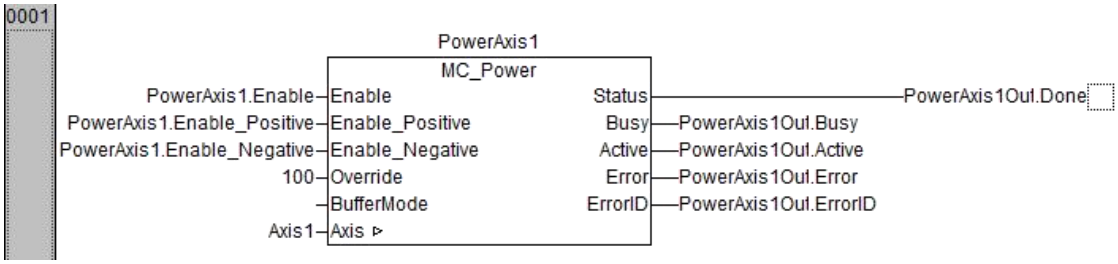




After importing the program, the connection between PLC program variable and NC axis variable is established.



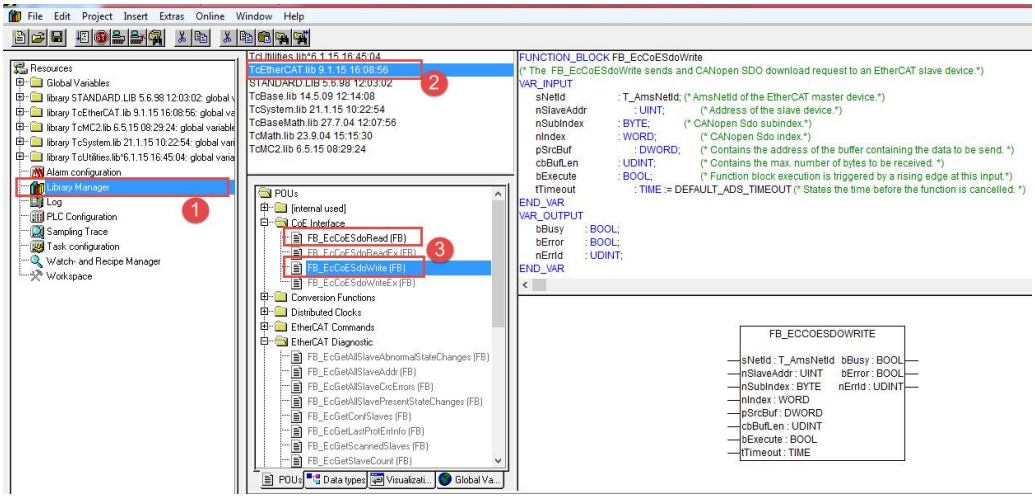
After completing the above steps, we can Control the stepper motor in TwinCAT PLC Control through the standard MC motion control function block, as shown in the following figure.



9.CoeSDO

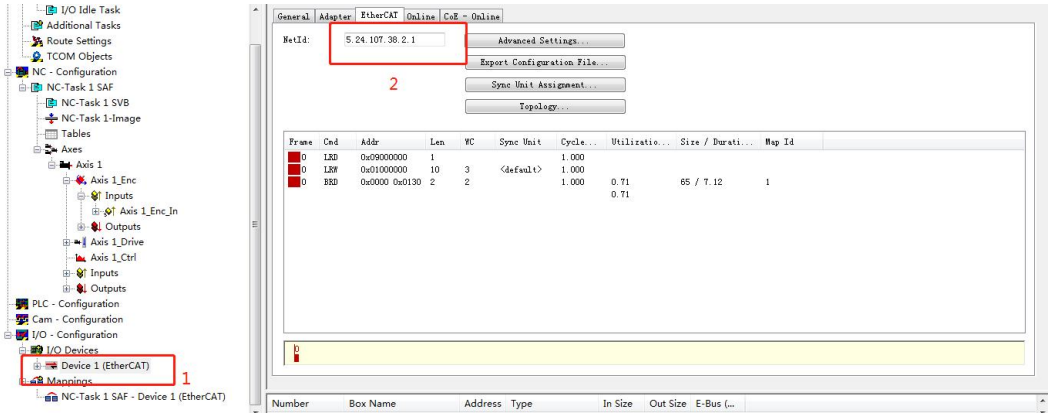
CoeSDO is similar to SDO in CANOPEN, which can be used to read and write some objects that exchange data infrequently or do not support PDO communication.The steps to use are as follows:

A. add "TcEtherCAT.lib" in the TwinCAT PLC Control library manager, the default directory is: C:\Program Files\TwinCAT\Plc\Lib.

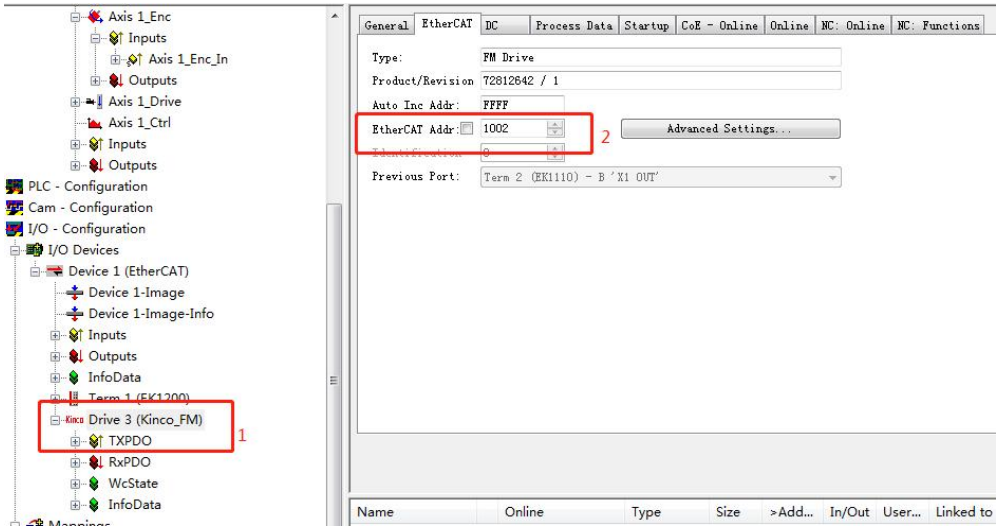


B.After the above steps are completed, the function blocks that CoeSDO reads and writes are declared in the program. We take the read stepper driver status word 60410010 and the write stepper driver target current parameter 60730010 as an example.

C.The first thing you need to do is look up the AmsNetId of EtherCAT's master station and go back to the Twincat System Manager.



Look for the address of the slave



The program is defined as follows:

```
PROGRAM CoeSDO
VAR
    FB_EcCoESdoRead1:FB_EcCoESdoRead;
    FB_EcCoESdoWrite1: FB_EcCoESdoWrite;

    sNetId          : T_AmsNetId:= '5.24.107.38.2.1';(* AmsNetId of the EtherCAT master device.*)
    nSlaveAddr      : UINT:=1002;                    (* Address of the slave device.*)
    bExecute        : BOOL;
    uReceiveData    :UINT;                           (* Contains the address of the buffer for the received data. *)
    uWriteData      :INT:=16#400;                    (* Contains the address of the buffer containing the data to be send. *)
    bReal           : BOOL;                          (* Function block execution is triggered by a rising edge at this input.*)
    bWrite          : BOOL;                          (* Function block execution is triggered by a rising edge at this input.*)

    bRbusy: BOOL;
    bRError: BOOL;
    nRErrId: UDINT;

    bWbusy: BOOL;
    bWError: BOOL;
    nWErrId: UDINT;
END_VAR
```

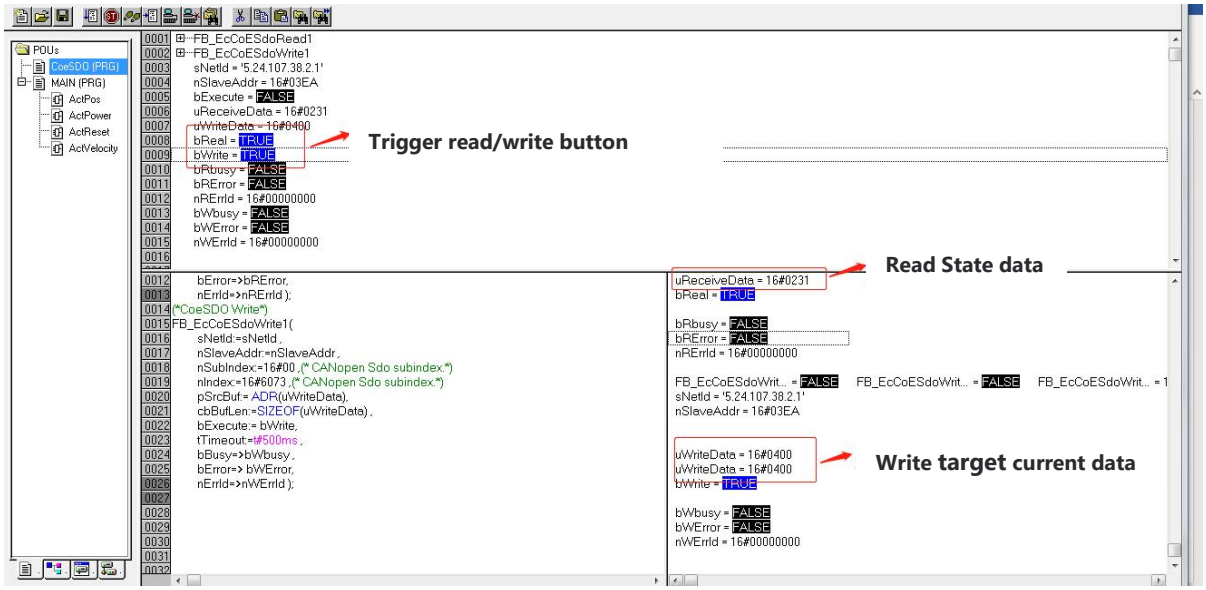
状态字是无符号16位数据
目标电流是整型16位数据

```
(*CoeSDO Read*)
FB_EcCoESdoRead1(
      sNetId:= sNetId,
      nSlaveAddr:=nSlaveAddr,
      nSubIndex:= 16#00,(* CANopen Sdo subindex.*)
      nIndex:=16#6041 ,(* CANopen Sdo subindex.*)
      pDstBuf:=ADR(uReceiveData) ,
      cbBufLen:=SIZEOF(uReceiveData) ,
      bExecute:=bReal ,
      tTimeout:=t#500ms ,
      bBusy=> bRbusy,
      bError=>bRError,
      nErrId=>nRErrId );
(*CoeSDO Write*)
FB_EcCoESdoWrite1(
      sNetId:=sNetId ,
      nSlaveAddr:=nSlaveAddr ,
      nSubIndex:=16#00 ,(* CANopen Sdo subindex.*)
      nIndex:=16#6073 ,(* CANopen Sdo subindex.*)
      pSrcBuf:= ADR(uWriteData),
      cbBufLen:=SIZEOF(uWriteData) ,
      bExecute:= bWrite,
      tTimeout:=t#500ms ,
      bBusy=>bWbusy ,
      bError=> bWError,
      nErrId=>nWErrId );
```

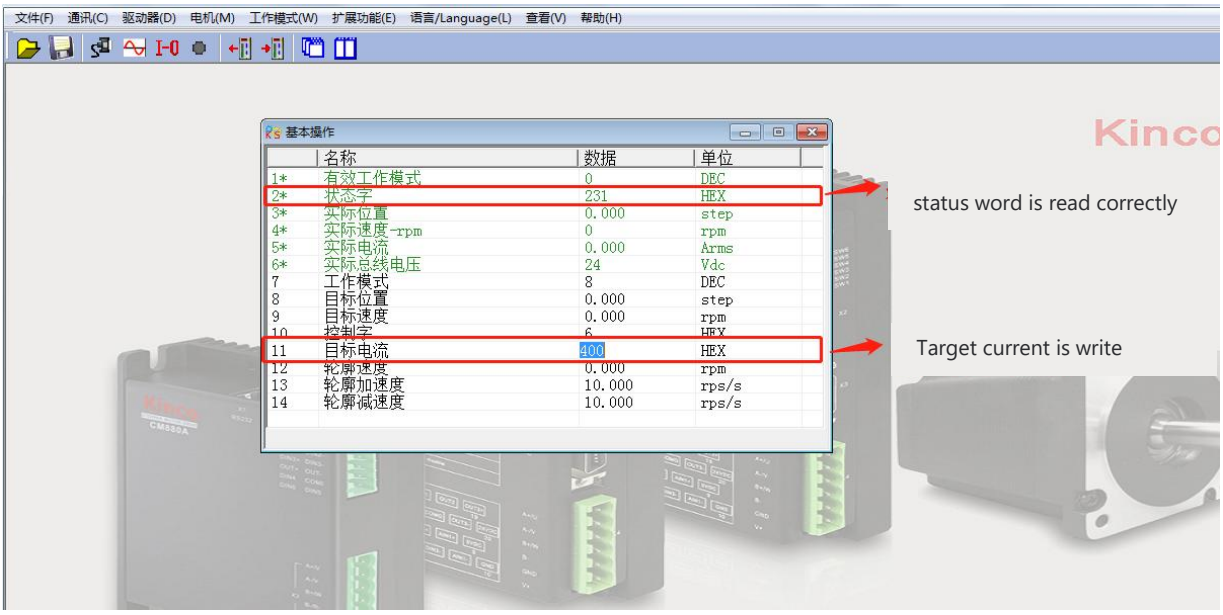
State word address

Torque current address

D. Call the read / write function block in the program, fill in the parameters to download the program, monitor the program and then trigger the read and write button respectively, if there is no error and accept the value in the register, CoeSDO communication is successful. As shown in the picture below:



E. Connect the stepper drive through the serial port and check whether the corresponding object parameters are correct.

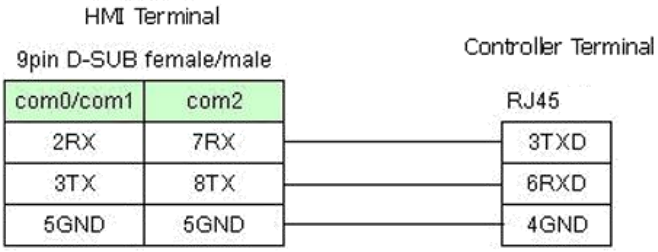

文件(F) 通讯(C) 驱动器(D) 电机(M) 工作模式(W) 扩展功能(E) 语言/Language(L) 查看(V) 帮助(H)
基本操作
名称 数据 单位
1* 有效工作模式 0 DEC
2* 状态字 231 HEX
3* 实际位置 0.000 step
4* 实际速度-rpm 0 rpm
5* 实际电流 0.000 Arms
6* 实际总线电压 24 Vdc
7 工作模式 8 DEC
8 目标位置 0.000 step
9 目标速度 0.000 rpm
10 控制字 6 HEX
11 目标电流 400 HEX
12 轮廓速度 0.000 rpm
13 轮廓加速度 10.000 rps/s
14 轮廓减速度 10.000 rps/s
status word is read correctly
Target current is write

Appendix IV RS232 Communication example

Communication between FM860 and KINCO HMI

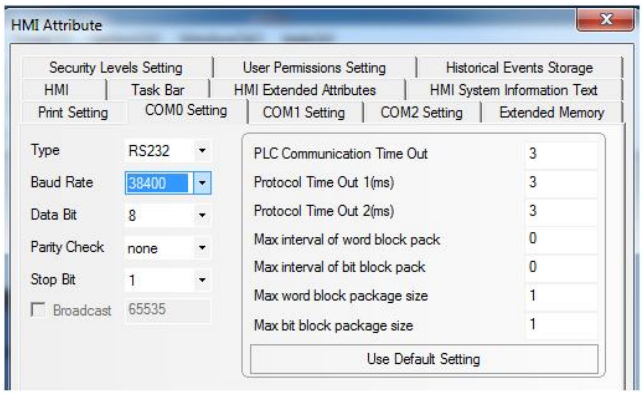
1.All series of Kinco HMI can be connected with the RS232 serial port of the FM driver. This example only introduces the main product of Kinco HMI MT4000, and MT5000 series is connected with the stepper driver. For other series of products, please refer to the user manual of each product or consult Kinco technical personnel.
2.Kinco MT4000,MT5000 series HMI can be connected with the FM stepper driver RS232 serial port, the user can use the HMI to set the FM internal parameters and operating status. The HMI can be connected to either a single FM driver or multiple FM drivers.
(1）HMI control single FM Drive

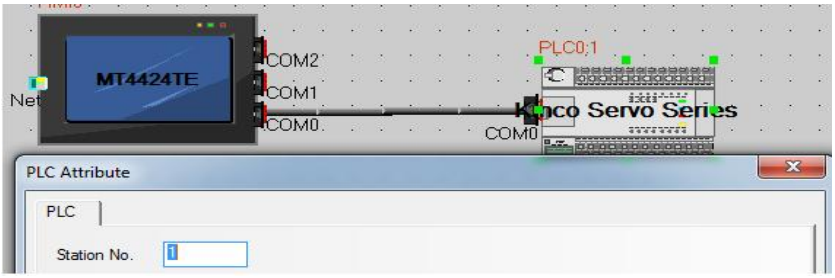
a．Wiring diagram



b. Communication parameter setting

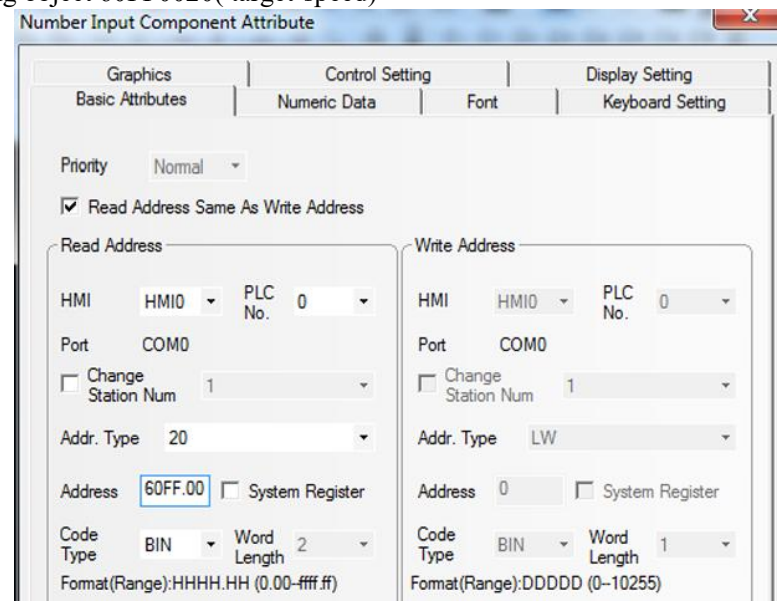
HMI select Kinco stepping driver, HMI communication parameter Settings are shown in figure, it should be noted that the PLC station number is the ID number of FM driver, FM driver factory ID default is 1, so the HMI control a single FM stepper PLC station number set to 1. If the FM ID number is set to N, then the following PLC station number should also be set to N. FM baud rate, station number and other parameters are set according to the 232 communication protocol chapter.





c. Address setting

At the first, we have to set up the data length of object as below picture to write HMI program. The address types are 08(8bits), 10(16bits) and 20(32 bits). The formatting of address is Index, Sub-index, following figure is an example for using object 60FF0020(target-speed)



Free Protocol Communication between FM860 and Adjustment tool of Serial port

RS232 communication protocol of FM drive is customized. User can control the drive by VB and VC. Following is the case for your reference.

1．Wiring diagram

PC RS232		FM driver RS232（RJ45）
RXD(2)	————————————	TXD(3)
TXD(3)	————————————	RXD(6)
GND(5)	————————————	GND(4)

2. Set the ID address and baud rate by referring to the serial port communication section. The default baud rate is 38400 and the station number is 1. Need to save and reboot if updated.

3．After completing the above steps, you can refer to the RS232 protocol to control the FM stepper driver. (Please refer to Chapter 7 before proceeding control)

In the following figure, send 6040 as 3F, the blue box is sending data, and the green box is sending correctly returned data. The red part is the communication parameters.

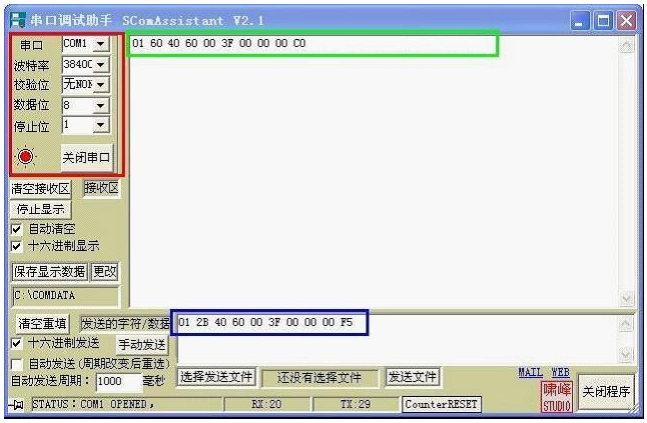


Table 9-7 Sending message in homing and position control mode

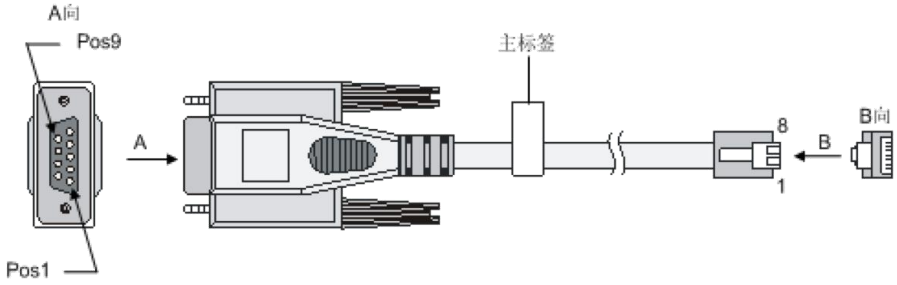
Homing control mode(the control word should change from F to 1F)				
CANopen address	Name	Value	Send or Return message（ID=1）	Note
60400010	Operation mode	0xF	01 2B 40 60 00 0F 00 00 00 25 01 60 40 60 00 0F 00 00 00 F0	Speed RPM have to Change as the unit of DEC. DEC=[(RPM*512*60000) /1875]
60600008	Homing mode	0x6	01 2F 60 60 00 06 00 00 00 0A 01 60 60 60 00 06 00 00 00 D9	
60980008	Turning signal speed of Homing	0x23	01 2F 98 60 00 23 00 00 00 B5 01 60 98 60 00 23 00 00 00 84	
60990120	Homing signal speed	200RPM	01 23 99 60 01 00 00 32 00 30 01 60 99 60 01 00 00 32 00 73	
60990220	Contorl word	150RPM	01 23 99 60 02 00 80 25 00 3C 01 60 99 60 02 00 80 25 00 FF	
60400010	Operation mode	0x1F	01 2B 40 60 00 1F 00 00 00 15 01 60 40 60 00 1F 00 00 00 E0	
01 40 41 60 00 00 00 00 00 1E Read status word, 9437 means found homing				
Position control mode(Absolute positioning of control word is changed from 2F to 3F, relative positioning is changed from 4F to 5F. 103F means activate immediately when position change)				
CANopen address	Name	Value	Send or Return message（ID=1）	Note
60400010	Control word	0xF	01 2B 40 60 00 0F 00 00 00 25 01 60 40 60 00 0F 00 00 00 F0	Speed RPM need change to DEC DEC=[(RPM*512*60000) /1875] Profile Acce and profile dece is defaulted as DEC. DEC=[(RPS/S*65536*60000)/1000/4000]
60600008	Operation mode	0x1	01 2F 60 60 00 01 00 00 00 0F 01 60 60 60 00 01 00 00 00 DE	
607A0020	Target position	50000 DEC	01 23 7A 60 00 50 C3 00 00 EF 01 60 7A 60 00 50 C3 00 00 B2	
60810020	Profile velocity	200RPM	01 23 81 60 00 00 00 32 00 C9 01 60 81 60 00 00 00 32 00 8C	
60830020	Profile_Acce	10rps/s	Default value	
60840020	Profile_Dece	10rps/s	Default value	
60400010	Control word	0x2F	01 2B 40 60 00 2F 00 00 00 05 01 60 40 60 00 2F 00 00 00 D0	
		0x3F(Absolute positioning)	01 2B 40 60 00 3F 00 00 00 F5 01 60 40 60 00 3F 00 00 00 C0	
		0x4F	01 2B 40 60 00 4F 00 00 00 E5 01 60 40 60 00 4F 00 00 00 B0	
		0x5F(Relative positioning)	01 2B 40 60 00 5F 00 00 00 D5 01 60 40 60 00 5F 00 00 00 A0	

Table 9-8 Sending message in speed mode

Speed control mode				
CANopen address	Name	Value	Send or Return message（ID=1）	Note
60600008	Operation mode	0x3	01 2F 60 60 00 03 00 00 00 0D 01 60 60 60 00 03 00 00 00 DC	Speed RPM need change to DEC DEC=[(RPM*512*60000)/1875] Profile Acce and profile dece is defaulted as DEC. DEC=[(RPS/S*65536*60000)/1000/4000]
60FF0020	Target speed	150RPM	01 23 FF 60 00 00 80 25 00 D8 01 60 FF 60 00 00 80 25 00 9B	
60830020	Profile_Acce	10rps/s	Default value	
60840020	Profile_Dece	10rps/s	Default value	
60400010	Control word	0xF	01 2B 40 60 00 0F 00 00 00 25 01 60 40 60 00 0F 00 00 00 F0	

Console cable

Console wire is the patch cord from Drive to PC, the end terminal is RS232 as DB9 port, others end is RJ45 port. The picture is as below.



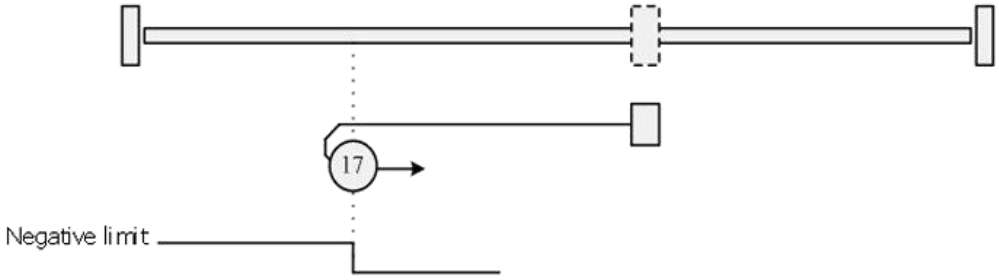
The connecting of the pins of cable as bellows:

DB9 (female)----------RJ45 port
RXD (2） ---------- TXD (3)
TXD (3) ---------- RXD (6)
DTR (4) ---------- DSR (7)
GND (5） ---------- GND (4)&(5)
DSR (6) ---------- DTR (2)
RTS (7) ---------- CTS (8)
CTS (8) ---------- RTS (1)

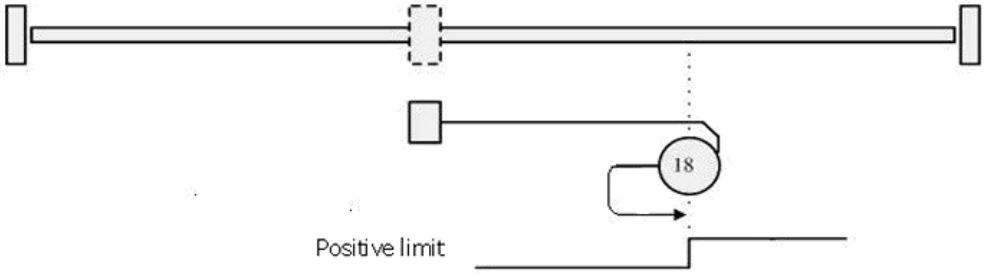
Appendix V Homing Method

The homing of FM drive is following the definition as DSP402 of CANopen. Support the homing mode from 17 to 30 and 35.

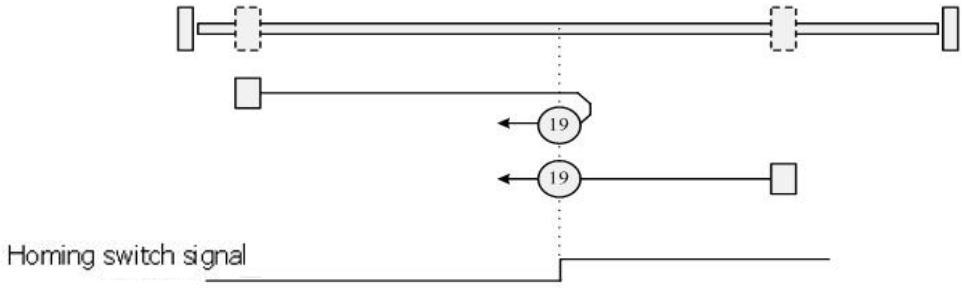
Mode17： Defined the homing signal as Negative limit, the running line is as belows.



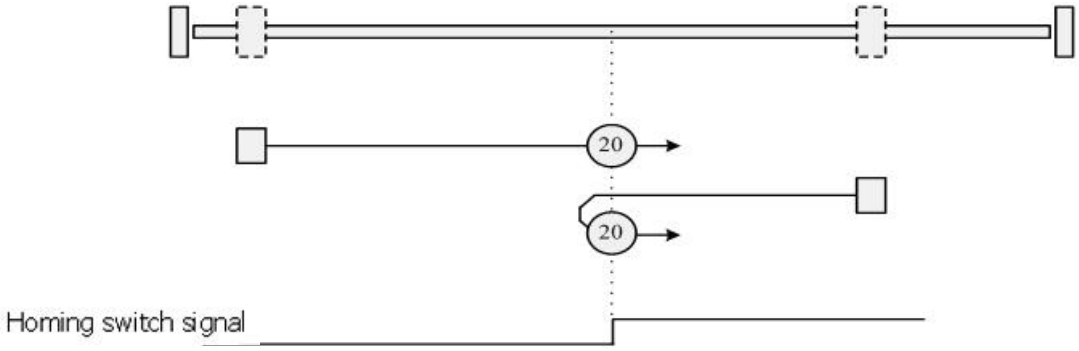
Mode 18： Defined Positive limit to be the homing signal, the running line is as bellows.



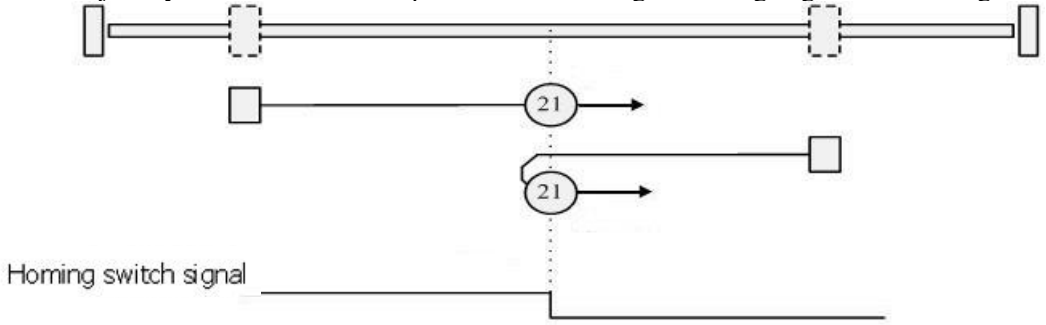
Mode 19： The external homing switch is the homing trigger signal, and the initial movement direction is positive. The movement trajectory is shown below. Stop when encountering the falling edge of the homing switch signal



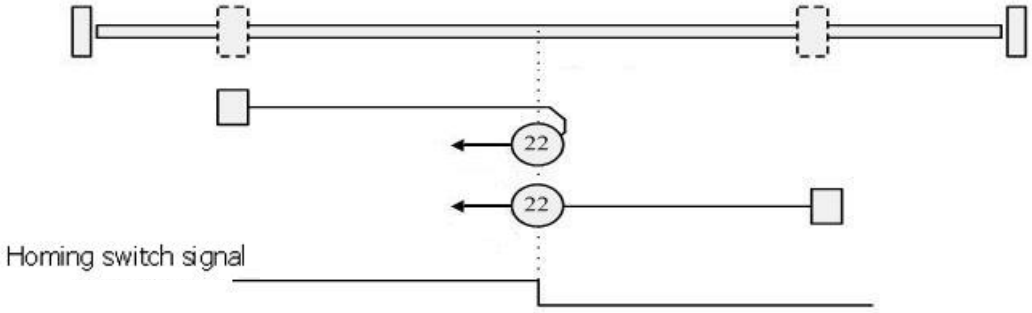
Mode 20： The external homing switch is the homing trigger signal, and the initial movement direction is positive. The movement trajectory is shown below. Stop when encountering the rising edge of the homing switch signal



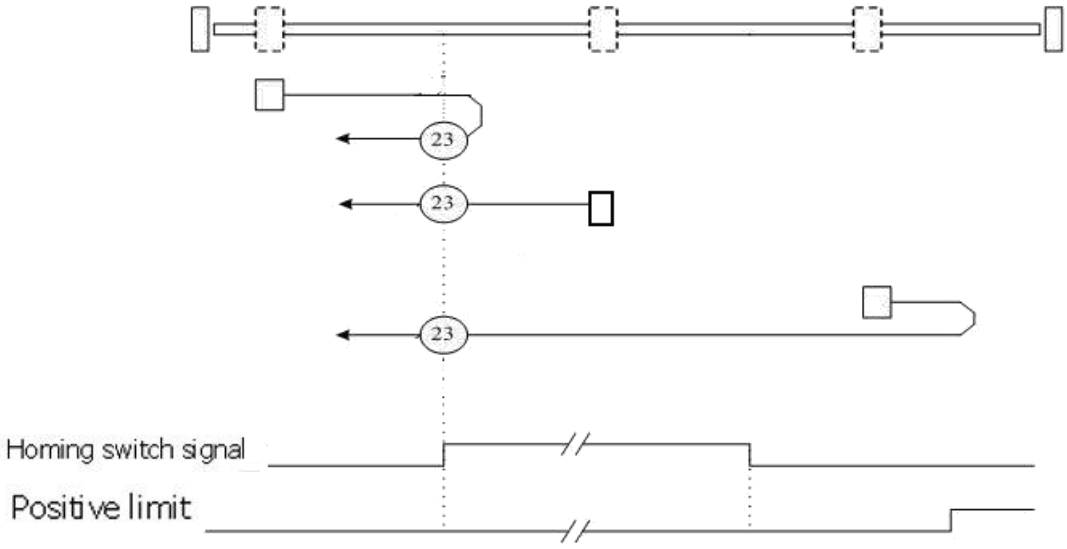
Mode 21: The external homing switch is the homing trigger signal, and the initial movement direction is negative. The movement trajectory is shown below. Stop when encountering the falling edge of the homing switch signal



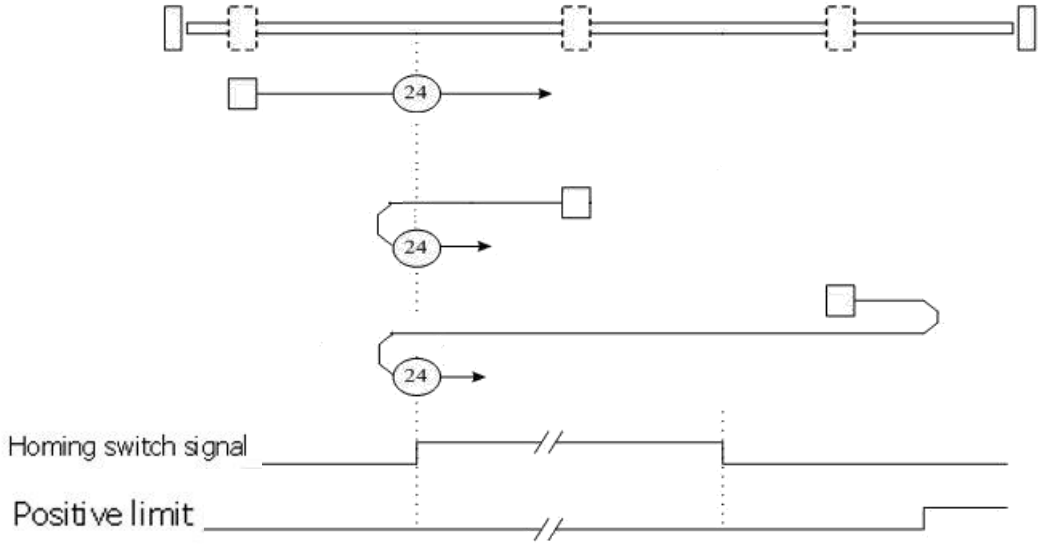
Mode 22: The external homing switch is the homing trigger signal, and the initial movement direction is negative. The movement trajectory is shown below. Stop when encountering the rising edge of the homing switch signal



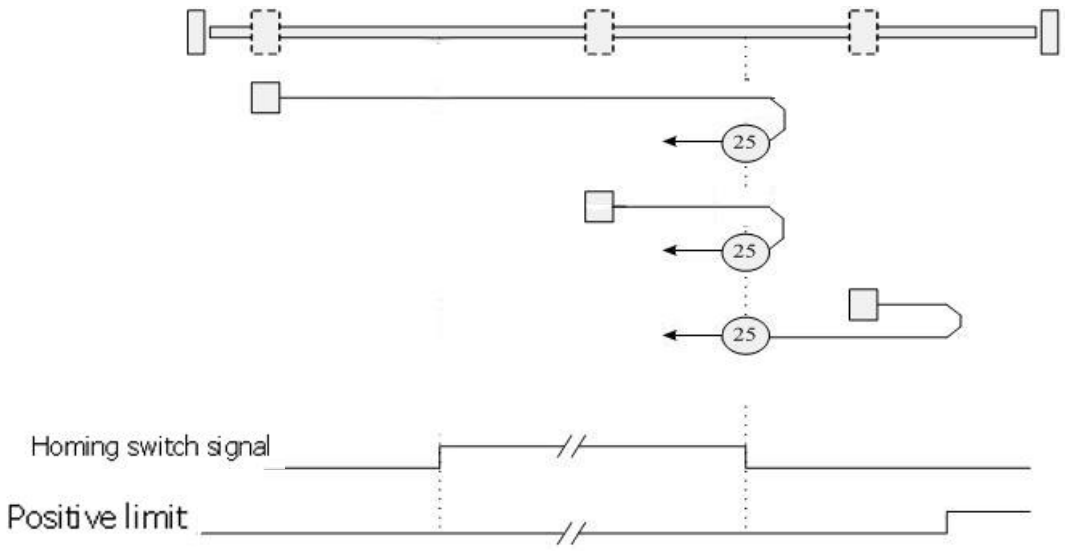
Mode 23: With a positive limit, the external homing switch is the homing trigger signal, the initial movement direction is positive, and the movement trajectory is shown as follows. It reverses when it encounters a positive limit effective signal, and stops when it encounters a negative direction falling edge of the homing switch signal.



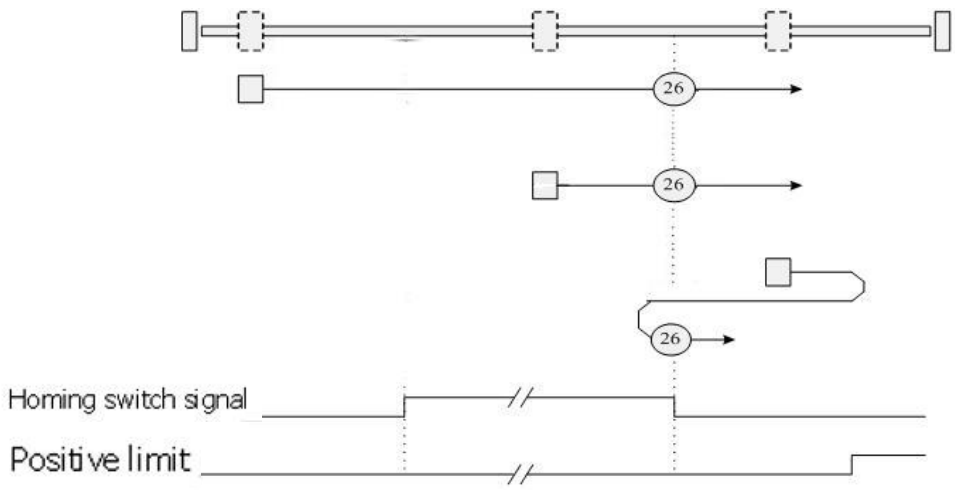
Mode 24： With a positive limit, the external homing switch is the homing trigger signal, the initial movement direction is positive, and the movement trajectory is shown as follows. It reverses when it encounters a positive limit effective signal, and stops when it encounters a positive direction rising edge of the homing switch signal.



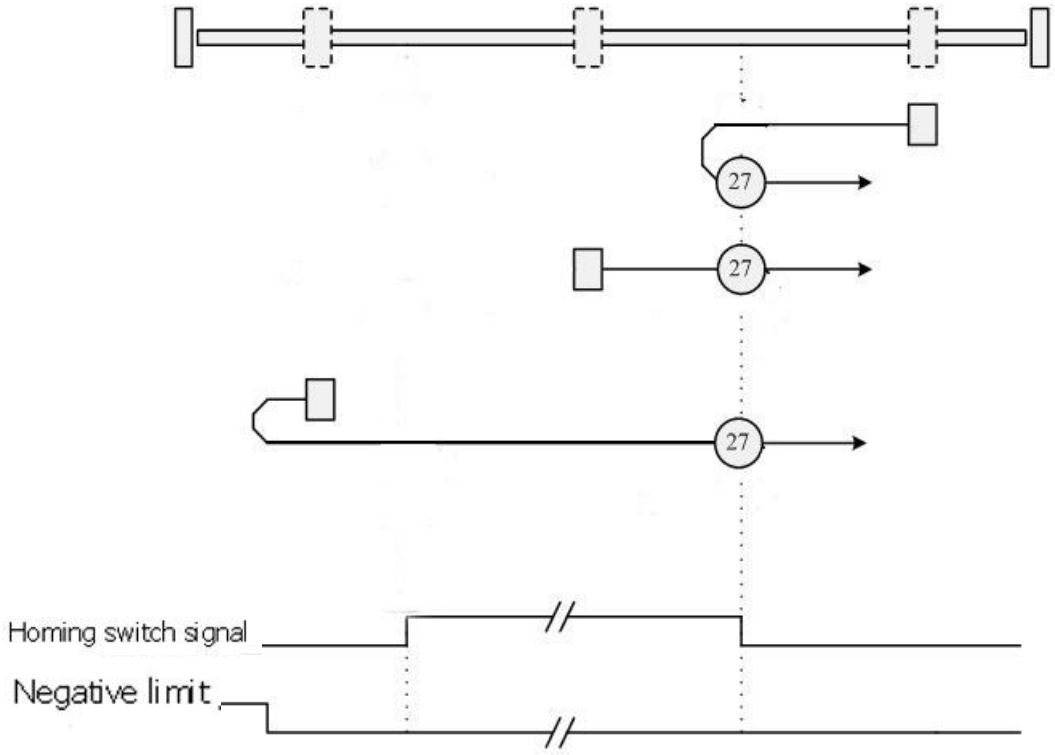
Mode 25： With a positive limit, the external homing switch is the homing trigger signal, the initial movement direction is positive, and the movement trajectory is shown as follows. It reverses when it encounters a positive limit effective signal, and stops when it encounters a negative direction rising edge of the homing switch signal.



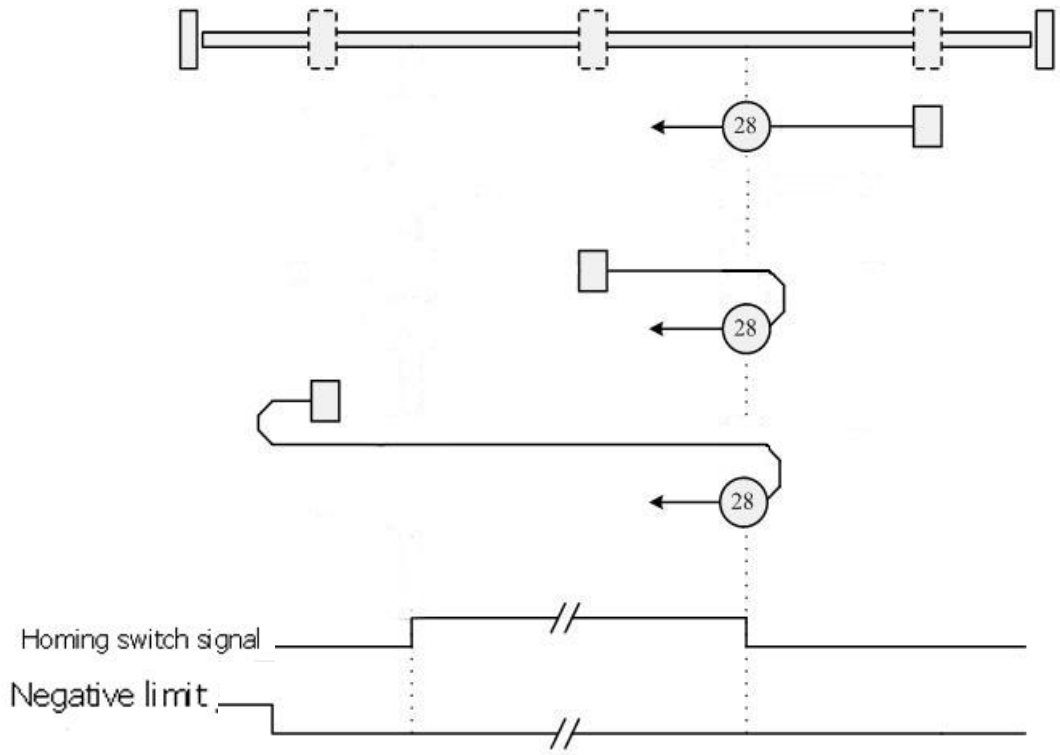
Mode 26： With a positive limit, the external homing switch is the homing trigger signal, the initial movement direction is positive, and the movement trajectory is shown as follows. It reverses when it encounters a positive limit effective signal, and stops when it encounters a positive direction falling edge of the homing switch signal.



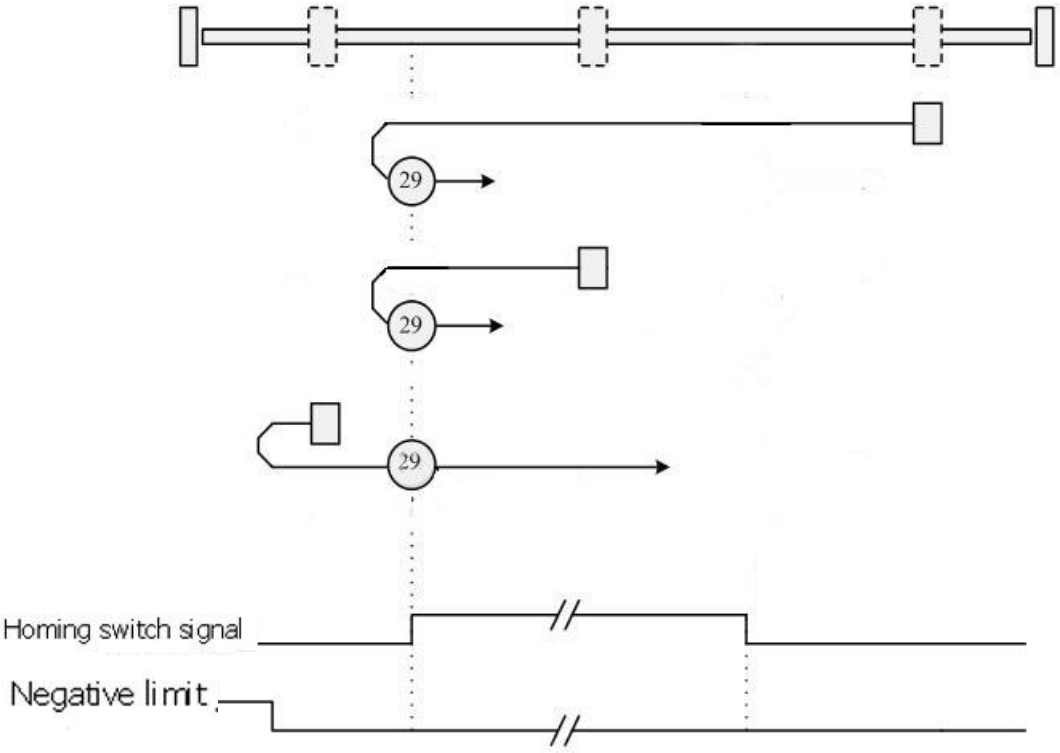
Mode 27： With a negative limit, the external homing switch is the homing trigger signal, the initial movement direction is negative, and the movement trajectory is shown as follows. It reverses when it encounters a negative limit effective signal, and stops when it encounters a positive direction falling edge of the homing switch signal.



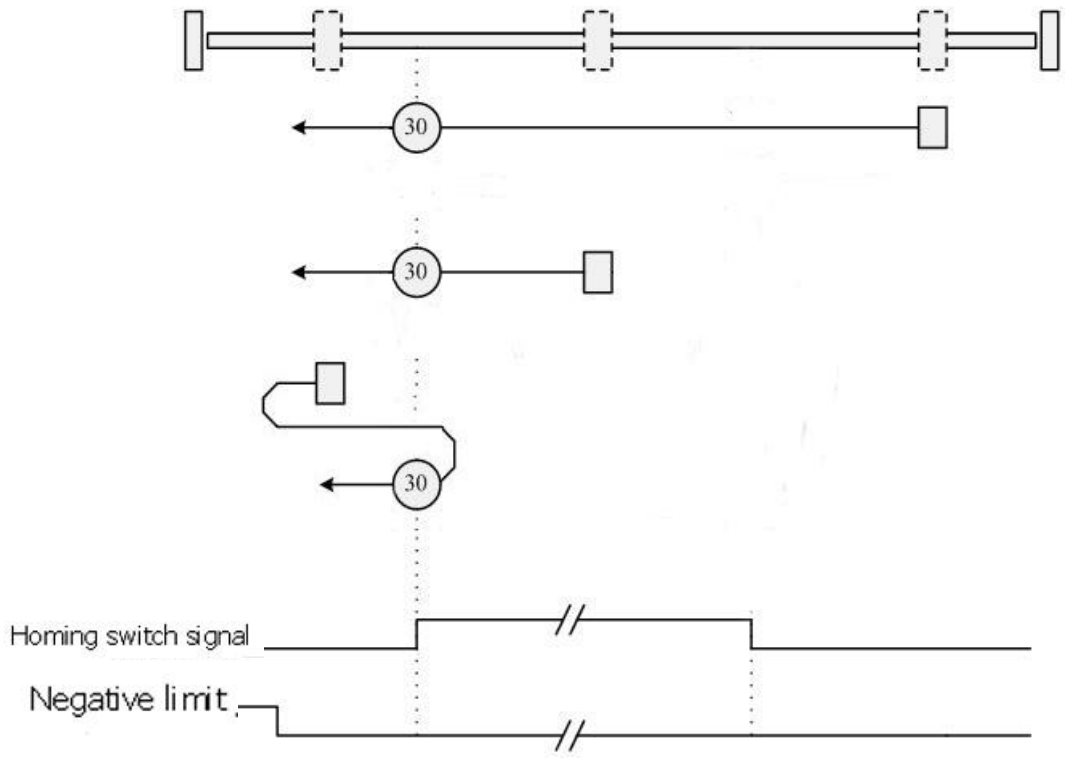
Mode 28： With a negative limit, the external homing switch is the homing trigger signal, the initial movement direction is negative, and the movement trajectory is shown as follows. It reverses when it encounters a negative limit effective signal, and stops when it encounters a negative direction rising edge of the homing switch signal.



Mode 29： With a negative limit, the external homing switch is the homing trigger signal, the initial movement direction is negative, and the movement trajectory is shown as follows. It reverses when it encounters a negative limit effective signal, and stops when it encounters a positive direction rising edge of the homing switch signal.



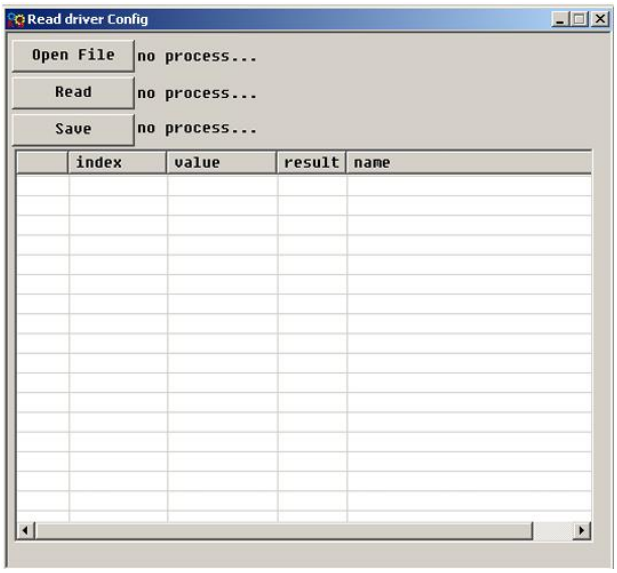
Mode 30： With a negative limit, the external homing switch is the homing trigger signal, the initial movement direction is negative, and the movement trajectory is shown as follows. It reverses when it encounters a negative limit effective signal, and stops when it encounters a negative direction falling edge of the homing switch signal.



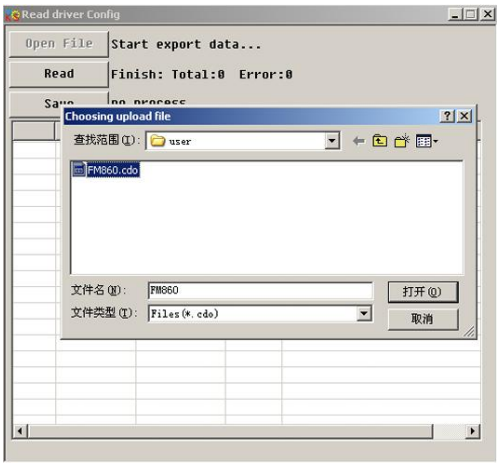
Mode 35: The current position of the motor is used as the reference homing.

Appendix VI Use Kincostep PC software to read / write drive

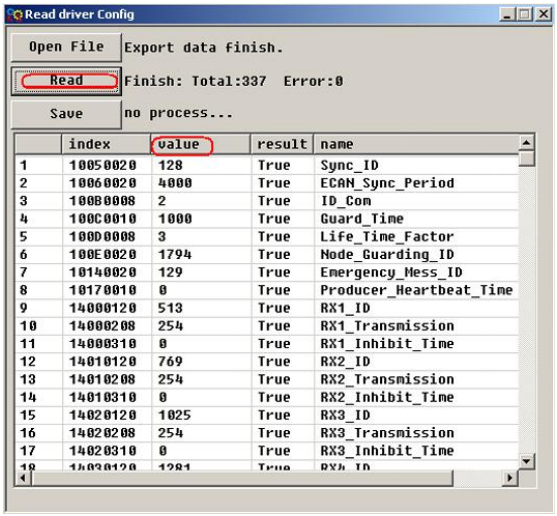
1. Please refer to Chapter 5, make your Drive to connect the software, and click extend function, then select Read Drive configuration.



2. Click Open File button to find the cod file in the default address, and open it.



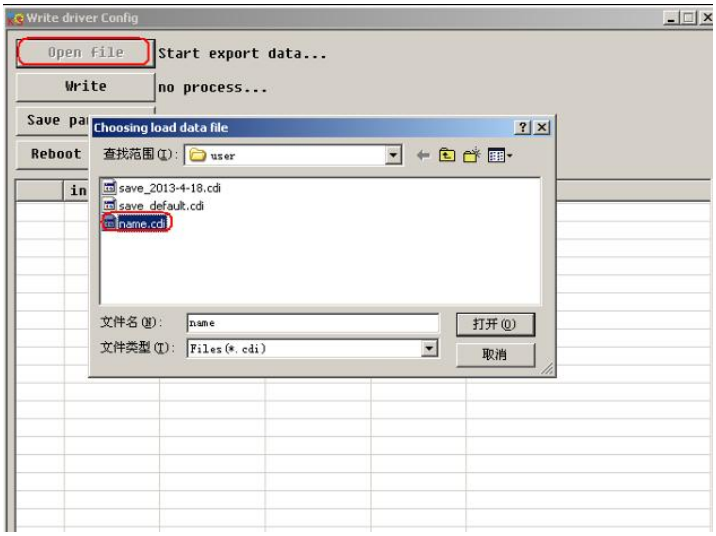
3. Click Read, then column of value will be read and shown as bellows.



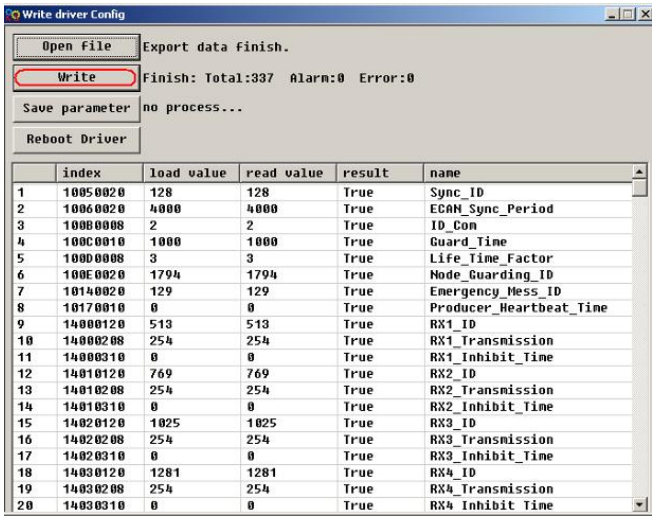
	index	value	result	name
1	10050020	128	True	Sync_ID
2	10060020	4000	True	ECAN_Sync_Period
3	100B0008	2	True	ID_Com
4	100C0010	1000	True	Guard_Time
5	100D0008	3	True	Life_Time_Factor
6	100E0020	1794	True	Node_Guarding_ID
7	10140020	129	True	Emergency_Mess_ID
8	10170010	0	True	Producer_Heartbeat_Time
9	14000120	513	True	RX1_ID
10	14000208	254	True	RX1_Transmission
11	14000310	0	True	RX1_Inhibit_Time
12	14010120	769	True	RX2_ID
13	14010208	254	True	RX2_Transmission
14	14010310	0	True	RX2_Inhibit_Time
15	14020120	1025	True	RX3_ID
16	14020208	254	True	RX3_Transmission
17	14020310	0	True	RX3_Inhibit_Time

4. Click Save button, make the data to be saved on the default position. Now you finished the operation.

5. If you want to use the data which you just saved to a new drive, then you can click Write Drive Configuration in the Extend list and select Open file button to find your data.

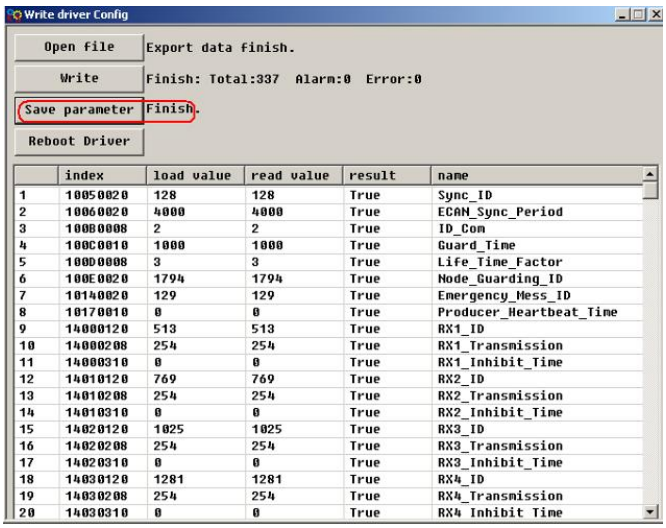


6. Click Write data, the data you saved will be updated to the drive, will be show True on result column. It means updated OK.



	index	load value	read value	result	name
1	10050020	128	128	True	Sync_ID
2	10060020	4000	4000	True	ECAN_Sync_Period
3	100B0008	2	2	True	ID_Con
4	100C0010	1000	1000	True	Guard_Time
5	100D0008	3	3	True	Life_Time_Factor
6	100E0020	1794	1794	True	Node_Guarding_ID
7	10140020	129	129	True	Emergency_Mess_ID
8	10170010	0	0	True	Producer_Heartbeat_Time
9	14000120	513	513	True	RX1_ID
10	14000208	254	254	True	RX1_Transmission
11	14000310	0	0	True	RX1_Inhibit_Time
12	14010120	769	769	True	RX2_ID
13	14010208	254	254	True	RX2_Transmission
14	14010310	0	0	True	RX2_Inhibit_Time
15	14020120	1025	1025	True	RX3_ID
16	14020208	254	254	True	RX3_Transmission
17	14020310	0	0	True	RX3_Inhibit_Time
18	14030120	1281	1281	True	RX4_ID
19	14030208	254	254	True	RX4_Transmission
20	14030310	0	0	True	RX4 Inhibit Time

7. Click Save button, the parameters will be saved to drive. Then will show you Finish.



	index	load value	read value	result	name
1	10050020	128	128	True	Sync_ID
2	10060020	4000	4000	True	ECAN_Sync_Period
3	100B0008	2	2	True	ID_Con
4	100C0010	1000	1000	True	Guard_Time
5	100D0008	3	3	True	Life_Time_Factor
6	100E0020	1794	1794	True	Node_Guarding_ID
7	10140020	129	129	True	Emergency_Mess_ID
8	10170010	0	0	True	Producer_Heartbeat_Time
9	14000120	513	513	True	RX1_ID
10	14000208	254	254	True	RX1_Transmission
11	14000310	0	0	True	RX1_Inhibit_Time
12	14010120	769	769	True	RX2_ID
13	14010208	254	254	True	RX2_Transmission
14	14010310	0	0	True	RX2_Inhibit_Time
15	14020120	1025	1025	True	RX3_ID
16	14020208	254	254	True	RX3_Transmission
17	14020310	0	0	True	RX3_Inhibit_Time
18	14030120	1281	1281	True	RX4_ID
19	14030208	254	254	True	RX4_Transmission
20	14030310	0	0	True	RX4 Inhibit Time

Appendix VII Conversion between Engineering unit and internal unit of Common objects

FM internal object some parameters have engineering units and internal units, in the communication control need to pay attention to conversion, such as speed engineering unit is RPM, internal unit is DEC, the relationship between the two is 1RPM is equal to 16384dec (internal turn 60000DEC)! Assuming the required speed is 10rpm, then the write speed is 163840dec when using the communication control! The following table lists the parameters that are commonly used to convert units!

Table 9-9 Internal unit conversion table

Parameter Name	Engineering unit	Internal unit	Conversion relationships
Speed	RPM	DEC	DEC=[(RPM*512*60000)/1875]
Acceleration	r/s*s	DEC	DEC=[(RPS/S*65536*60000)/4000000]
The number of step/rev pulses per revolution （Mcrostep）	step/rev	DEC	1rev=60000DEC,if 400step=1rev,1step=60000/400=150DEC

Appendix VIII Common Object List

Based on Chapter 7 Communication protocol described, all parameter value are transferred in hexadecimal data. In the later section of this document, we adopt the hexadecimal system and use index (16-bi index) and Sub-index (8-bit sub-index) to represent the register addressing. The digit 08 indicates the register will store data up to 1 byte, and the digit 10 indicates that the register will store data up to 2 bytes, and the digit 20 indicates the register will store data up to 4 bytes. It also covers the storage digits and read/write property of the register, read or write flag(RW), read-only or write-only flag(RO, WO) and mapping flag(M).

Table 9-10 Mode and control

CANopen adress	Bits	Modbus address	Type	Unit	Description
6040+00	10	0x3100	RW	Bit	bit0：Switch on bit1：Disable voltage bit2：Quick stop bit3：Enable operation bit4：New setpoint/Start homing bit5：Change set immediately bit6：Absolute or relative bit7：Fault reset bit8：Stop bit9：Reserved bit10：Reserved bit11：Manufacturer specific 1 bit12：Manufacturer specific 2 bit13：Manufacturer specific 3 bit14：Manufacturer specific 4 bit15：Manufacturer specific 5 Use control word to change status of drive=>machine state 0x06 Motor power-off 0x0F Motor power-on 0x0B Quick stop, load stop-voltage switched off 0x2F-3F Start absolute positioning immediately 0x4F-5F Start relative positioning 0x103F Start absolute positioning mode according to target-position changes 0x0F-1F Start Homing 0x80 Clear internal error
6041+00	10	0x3200	RO	Bit	Status byte shows the status of drive bit0：Ready to switch on bit1：Switch on bit2：Operation enable bit3：Fault bit4：Voltage Disable bit5：Quick Stop bit6：Switch on disable bit7：Warning bit8：Internal reserved bit9：Reserved bit10：Target reach bit11：Internal limit active bit12：Step.Ach./V=0/Hom.att bit13：Foll.Err/Res.Hom.Err. bit14：Commutation found bit15：Homing found

6060+00	08	0x3500	RW	DEC	Operation mode: 1---Position mode 3---Speed mode -4---Pulse mode 6---Return to homing mode 8---Synchronous position mode
6061+00	08	0x3600	RO	DEC	Effective operation mode

Table 9-11 Measurement data

CANopen adress	Bits	Modbus address	Type	Unit	Description
6063+00	20	0x3700	RO	DEC	Real position value
606C+00	10	0x3B00	RO	RPM	Real speed（rpm） Internal sampling time 10mS
6078+00	10	0x3E00	RO	1 Ap=1.414*Arms	Real current value FM860：1 Arms =79dec FM880：1Arms =210.6dec
60FD+00	20	0x6D00	RO	Bit	Input status bit0: Negative limit signal status bit1: Positive limit signal status bit2: Homing signal status
6079+00	10	0x3F00	RO	V	Actual bus voltage

Table 9-12 Target object

CANopen adress	Bits	Modbus address	Type	Unit	Description
607A+00	20	0x4000	RW	DEC	Target position in operation mode 1, shift demand position if control word starts motion.
6081+00	20	0x4A00	RW	DEC=[(RPM*512*60000)/1875]	Max. velocity of trapezium profile in mode1.
6083+00	20	0x4B00	RW	DEC=[(RPS/S*65536*60000)/4000000]	Profile_Acce Default value:10rps/s
6084+00	20	0x4C00	RW		Profile_Dece Default value:10rps/s
6085+00	20	0x3300	RW		Quick stop deceleration Default value：100rps/s
60FF+00	20	0x6F00	RW	DEC=[(RPM*512*60000)/1875]	Target speed in mode 3
6073+00	10	0x3D00	RW	1 Ap=1.414*Arms	Target current FM860：1 Arms =79dec FM880：1Arms =210.6dec
6080+00	10	0x4900	RW	RPM	Max. speed limit
605A+00	08	0x3400	RW	DEC	Quick stop mode
605D+00	08	0x3430	RW	DEC	Pause mode

Table 9-13 Multiple-position, multiple-speed (DEC=[(RPM*512*60000)/1875])

CANopen adress	Bits	Modbus address	Type	Unit	Description
2020+01	20	0x0C10	RW	DEC	Multiple position control 0
2020+02	20	0x0C20	RW	DEC	Multiple position control 1
2020+03	20	0x0C30	RW	DEC	Multiple position control 2
2020+04	20	0x0C40	RW	DEC	Multiple position control 3
2020+05	20	0x0C50	RW	DEC	Multiple position control 4
2020+06	20	0x0C60	RW	DEC	Multiple position control 5
2020+07	20	0x0C70	RW	DEC	Multiple position control 6
2020+08	20	0x0C80	RW	DEC	Multiple position control 7
2020+09	20	0x0C90	RW	DEC	Multiple position control 8
2020+0A	20	0x0CA0	RW	DEC	Multiple position control 9
2020+0B	20	0x0CB0	RW	DEC	Multiple position control 10
2020+0C	20	0x0CC0	RW	DEC	Multiple position control 11
2020+0D	20	0x0CD0	RW	DEC	Multiple position control 12
2020+0E	20	0x0CE0	RW	DEC	Multiple position control 13
2020+0F	20	0x0CF0	RW	DEC	Multiple position control 14
2020+10	20	0x0D00	RW	DEC	Multiple position control 15
2020+11	20	0x0D10	RW	DEC	Multiple speed control 0
2020+12	20	0x0D20	RW	DEC	Multiple speed control 1
2020+13	20	0x0D30	RW	DEC	Multiple speed control 2
2020+14	20	0x0D40	RW	DEC	Multiple speed control 3
2020+15	20	0x0D50	RW	DEC	Multiple speed control 4
2020+16	20	0x0D60	RW	DEC	Multiple speed control 5
2020+17	20	0x0D70	RW	DEC	Multiple speed control 6
2020+18	20	0x0D80	RW	DEC	Multiple speed control 7
2020+19	20	0x0D90	RW	DEC	Multiple speed control 8
2020+1A	20	0x0DA0	RW	DEC	Multiple speed control 9
2020+1B	20	0x0DB0	RW	DEC	Multiple speed control 10
2020+1C	20	0x0DC0	RW	DEC	Multiple speed control 11
2020+1D	20	0x0DD0	RW	DEC	Multiple speed control 12

2020+1E	20	0x0DE0	RW	DEC	Multiple speed control 13
2020+1F	20	0x0DF0	RW	DEC	Multiple speed control 14
2020+20	20	0x0E00	RW	DEC	Multiple speed control 15
2020+36	08	0x0F60	RW	DEC	Multiple position control choose display
2020+37	08	0x0F70	RW	DEC	Multiple speed control choose display
2020+38	10	0x0F80	RW	ms	Multiple speed/position Switching delay

Table 9-14 Performance Object

CANopen adress	Bits	Modbus address	Type	Unit	Description
6065+00	20	0x3800	RW	DEC	Maximum following error Default value 200 DEC
6067+00	20	0x3900	RW	DEC	Position reach window Position range for target reached Default value 10 DEC
607D+01	20	0x4410	RW	DEC	Soft positive limit
607D+02	20	0x4420	RW	DEC	Soft negative limit

Table 9-15 Homing control

CANopen adress	Bits	Modbus address	Type	Unit	Description
6098+00	08	0x4D00	RW	DEC	Homing mode (refer to homing control mode)
6099+01	20	0x5010	RW	DEC=[(RPM*512 *60000)/1875]	Speed of searching limit switch
6099+02	20	0x5020	RW		Speed for searching homing signal
609A+00	20	0x5200	RW	DEC=[(RPS/S*65536 *60000)/4000000]	Acceleration for searching homing
607C+00	20	0x4100	RW	DEC	Homing offset
6099+04	10	0x5040	RW	1 Ap=1.414*Arms	Homing current FM860：1 Arms =79dec FM880：1Arms =210.6dec

Table 9-16 Input and output parameter

CANopen adress	Bits	Modbus address	Type	Unit	Description
2010+03	20	0x0830	RW	Bit	Function definition of Digital input 1
2010+04	20	0x0840	RW	Bit	Function definition of Digital input 2
2010+05	20	0x0850	RW	Bit	Function definition of Digital input 3
2010+06	20	0x0860	RW	Bit	Function definition of Digital input 4
2010+07	20	0x0870	RW	Bit	Function definition of Digital input 5
2010+08	20	0x0880	RW	Bit	Function definition of Digital input 6
2010+10	20	0x0900	RW	Bit	Function definition of Digital output 1
2010+11	20	0x0910	RW	Bit	Function definition of Digital output 1
2010+12	20	0x0920	RW	Bit	Function definition of Digital output 1
2010+0B	10	0x08B0	RO	Bit	Input status bit0：Din1 bit1：Din2 bit2：Din3 bit3：Din4 bit4：Din5 bit5：Din6

2010+18	10	0x0980	RO	Bit	Output status bit0：Dout1 bit1：Dout2 bit2：Dout3
2010+01	10	0x0810	RW	Bit	Input signal polarity definition (0: normally closed; 1: Normally open) bit0：Din1 bit1：Din2 bit2：Din3 bit3：Din4 bit4：Din5 bit5：Din6 Default FF
2010+0E	10	0x08E0	RW	Bit	Output signal polarity definition (0: normally closed; 1: Normally open) bit0：Dout1 bit1：Dout2 bit2：Dout3 Default FF
2010+02	10	0x0820	RW	Bit	Signal Simulation of input bit0：Din1 bit1：Din2 bit2：Din3 bit3：Din4 bit4：Din5 bit5：Din6
2010+0F	10	0x08F0	RW	Bit	Signal Simulation of output bit0：Dout1 bit1：Dout2 bit2：Dout3
2020+31	08	0x0F10	RW	DEC	Input operation mode control 0 Without input mode
2020+32	08	0x0F20	RW	DEC	Input operation mode control 1 With input mode
2020+33	10	0x0F30	RW	DEC	Input control word

Table 9-17 Pulse input parameter

CANopen adress	Bits	Modbus address	Type	Unit	Description
6410+18	10	0x7180	RW	DEC	Numerator of pulses per revolution
2508+03	08	0x1930	RO	DEC	Pulse mode control 0...Double pulse mode 1...Pulse direction mode 2...Incremental encoder mode Note: select pulse mode via IO
2508+04	20	0x1940	RW	DEC	Input pluses before electronic gear
2508+05	20	0x1950	RW	DEC	Input pluses after electronic gear
2508+06	10	0x1960	RW	DEC	Pulse filter parameters f=1000/（2π*Pulse_Filter） τ = Pulse _Filter/1000（S）
2508+0C	10	0x19C0	RW	kHz	Pulse frequency before gear
2508+0D	10	0x19D0	RW	kHz	Pulse frequency after gear
6410+18	10	0x7180	RW	Step/rev	The pulse number of motor per revolution

Table 9-18 Analog input parameters

CANopen adress	Bits	Modbus address	Type	Unit	Description
2502+0F	10	0x16F0	RW	DEC	Output data of analog 1
2502+01	10	0x1610	RW	DEC	Analog 1 filter parameter f=4000/（2π*Analog1_Filter） τ = Analog1_Filter/4000（S）
2502+02	10	0x1620	RW	DEC	Analog 1 Dead voltage DEC=Dead_Voltage/10V*8192
2502+03	10	0x1630	RW	DEC	Analog 1 Offset voltage DEC=Offset_Voltage/10V*8192
2502+14	08	0x1740	RW	DEC	Analog 1 Output data polarity
2502+13	10	0x1730	RW	RPM	Analog 10V corresponds to speed
2502+07	08	0x1670	RW	HEX	Simulate-speed control 0: Invalid analog channel 1:Valid analog channel 1 (AIN1 ） 0x10～0x1f：AIN1“Control inside speed control section【x-10】”

Table 9-19 Motor parameters

CANopen adress	Bits	Modbus address	Type	Unit	Description
6410+01	10	0x7010	RW	HEX	Select Motor type ASCI....HEX....Model "00".....3030....no motor select "MC".....434d....self detect motor parameter "XX".....5858...Customize motor parameters "A1".....3141...2S42Q-03848 "A2".....3241...2S42Q-02940 "B1".....3142...2S56Q-030B5 "B2".....3242...2S56Q-02976 "B3".....3342...2S56Q-02054 "B4".....3442...2S56Q-02741 "C1".....3143...2S86Q-069B8 "C2".....3243...2S86Q-05180 "C3".....3343...2S86Q-03865 "C4".....3443...2S86Q-051F6 "C5".....3543...2S86Q-030B8 "C6".....3643...2S86Q-03080 "C7".....3743...2S86Q-01865 "D1".....3144...2S110Q-054K1 "D2".....3244...2S110Q-047F0 "D3".....3344...2S110Q-03999 "E1".....3145...2S130Y-063R8 "E2".....3245...2S130Y-039M0 "F1".....3146...3S57Q-04079 "F2".....3246...3S57Q-04056 "F3".....3346...3S57Q-04042 "G1".....3147...3S85Q-04097 "G2".....3247...3S85Q-04067 "G3".....3347...3S85Q-040F7
6410+05	08	0x7050	RW	2p/r	Number of motor polarity is defaulted as 50

6410+0B	10	0x70B0	RW	1Arms=10dec	The motor phase current
6510+04	10	0x8040	RW	1Arms=10dec	The motor phase current limit
6410+0C	10	0x70C0	RW	1mH=10dec	The inductance of motor phase
6410+0D	10	0x70D0	RW	1Ω=100dec	The resistor of motor phase
6410+13	08	0x7130	RW	DEC	Direction of motor turn 0：clockwise 1：counter-clockwise
6410+16	10	0x7160	RO	HEX	Motor model currently in use
6410+19	08	0x7190	RW	DEC	Motor parameter with power on and self-test switch 0：Disable 1：Enable
6410+1A	08	0x71A0	RW	DEC	The phase of motor 2：2 phase stepping motor 3：3 phase stepping motor
6410+1B	08	0x71B0	RW	DEC	IO port with 5V voltage output switch 0：Disable 1：Enable

Table 9-20 Parameters for saving

CANopen adress	Bits	Modbus address	Type	Unit	Description
2FF0+01	08	0x2910	RW	DEC	1: Store all configuration parameters set 10: Initialize all configuration parameters Note: Store control loop parameters, excluding motor parameters.
2FF0+03	08	0x2930	RW	DEC	1：Store all motor parameters set

Table 9-21 Error code

CANopen adress	Bits	Modbus address	Type	Unit	Description
2601+00	10	0x1F00	RO	Bit	Real-time alarm error lower to 16 bits bit 0：internal bit 1：over current bit 2：over voltage bit 3：low voltage bit 4：Temperature bit 5：internal logic voltage bit 6：5V input current overload bit 7：internal EEPROM bit 8：Searching Motor error bit 9：Position error bit10：Bus error bit 11：Input pulse frequent bit 12：-- bit 15：
2602+00	10	0x2000	RO	Bit	Real-time alarm error over to 16 bits Bit 0：-- Bit 15:
2603+00	10	0x2100	RO	Bit	Self-test error status lower to 16 bit with power-on bit 0：A phase over current fault bit 1：B phase over current fault bit 2：Over voltage error

					bit 3：Low voltage error bit 4：Low power input bit 5：Temperature error bit 6: A phase current circuit fault bit 7: B phase current circuit fault bit 8：A phase over current circuit fault bit 9：B phase over current circuit fault bit 10：A phase power circuit fault bit 11：B phase power circuit fault bit 12：Motor phase wrong wiring bit 13：Motor A phase wrong wiring bit 14：Motor B phase wrong wiring bit 15：Motor A phase without wiring
2604+00	10	0x2200	RO	Bit	Self-test error status up to 16 bits bit 0：Motor B phase without wiring bit 0：Motor B phase without wiring bit 2：Logic 15V voltage bit 3：Logic 5V voltage bit 4：Output 5V overload bit 5：Memory read failure bit 6：Memory write failure bit 7: Memory read and write failure bit 8：Memory read data checking error bit 9：Reserved bit 10：Reserved bit 11：watch-dog reset bit 12：Reserved bit 13：Program does not match with the PCB bit 14：A crystal oscillator circuit fault bit 15：ADC Conversion circuit fault
2610+00	08		RO	Bit	Drive error group 0
2611+00	08		RO	Bit	Drive error group 1
2612+00	08		RO	Bit	Drive error group 2
2613+00	08		RO	Bit	Drive error group 3
2614+00	08		RO	Bit	Drive error group 4
2615+00	08		RO	Bit	Drive error group 5
2616+00	08		RO	Bit	Drive error group 6
2617+00	08		RO	Bit	Drive error group 7

Table 9-22 Bus specification parameter

CANopen adress	Bits	Modbus address	Type	Unit	Description
1005+00	20	--	RW	HEX	Drive synchronization message ID 0x80
1006+00	20	--	RW	uS	Set the synchronization cycle time of the drive
100B+00	08	0x0600	RW	DEC	Current station number of the device drive. Note: 1) The ID station number can be set 1 ~ 63 by SW6-SW1 or 1 ~ 127 by 0x2FE400. 2) Changing this parameter requires storage and restart.
100C+00	10	--	RW	mS	Set the protection time of the drive node

100D+00	08	--	RW	DEC	Driver node protection time coefficient It is multiplied by the node protection time to obtain the total time of the node protection protocol
1014+00	20	--	RW	HEX	Drive emergency message ID
2FE4+00	08	0x2800	RW	DEC	Drive station No. 1～127; Note: your setting is valid when SW6-SW1 on OFF. Then reboot it when you updated the parameters.
2F81+00	08	0x2300	RW	DEC	CAN baud rate setting default value is 50 Value Baud rate 100： 1M 50： 500k 25： 250k 12： 125k 5： 50k Need to save and reboot
2FE0+00	10	0x2400	RW	DEC	RS232 baud rate setting default value is 259 Value Baud rate 2082 4800 1041 9600 520 19200 259 38400 86 115200 Need to save and reboot
2FE2+00	10	0x2600	RW	DEC	RS485 baud rate setting default value is 520 Value Baud rate 1041 9600 520 19200 259 c38400 86 115200 Need to save and reboot