Kinco
UK & Ireland Official Distributor
MOTION CONTROL PRODUCTS

MD Series
Integrated DC Servo Motors
User Manual

Kinco Electric (shenzhen) Ltd.
UK & Ireland Official Distributor: Motion Control Products Ltd.
11-15 Francis Avenue, Bournemouth, Dorset, UK, BH11 8NX
Tel.: +44 (0)1202 599922
enquiries@motioncontrolproducts.com
www.motioncontrolproducts.com

Preface and product confirmation

Thank you for using Kinco servo products!

The accessories of different models of Kinco drives are different. We recommend that you confirm the product carefully before use.

Confirmation

Confirmation item	Description
Does it match the model you ordered?	Please refer to the product nameplate information to confirm whether it matches the model you ordered.
Is there any damage to the product appearance?	Please confirm whether the product was damaged during transportation.
Are the product accessories complete?	Please confirm whether the integrated servo product includes accessories.

If you have any problem with any of the above, please contact us or your supplier.

Manual version change records

Date	Change description
2020.5	New product manual released
2020.8	Update Section 1.1, Section 2.2 , Section 3.1 and Section 3.2 , amend Section 9.1, Appendix I
2020.12	Chapter 2 and Chapter 3 add MD-EA product description
2021.4	Added 2.1.7
2024.8	Modified 1.1 Power Supply Range Update product specifications, dimensions and weight parameters
2024.9	Update section 3.3.3

Catalog

Chapter 1 System configuration and model description

1.1 Product specifications

Model		MD60-020-D□■K-★A-000	MD60-040-D□■K-★A-000	MD80-075-D□■K-★A-000
Power supply		24VDC ~ 60VDC		
Current	Rated current（rms）	5Arms	10Arms	20Arms
	Peak current(PEAK)	21Ap	36Ap	80Ap
Rated power(W)		200	400	750
Rated speednN(rpm)		3000	3000	3000
Rated torqueTs(Nm)		0.64	1.27	2.39
Maximum torque(Nm)		1.92	3.81	7.17
Rotor moment of inertiaJm(Kg·cm²)		0.214	0.405	1.087
		0.218(With brake)	0.409(With brake)	1.099(With brake)
Brake chopper		Via wiring an external braking resistor（mainly in quick start and stop application）		
Brake chopper threshold		The default value is 73V, which can be set by software		
Over-voltage alarming threshould		The default is 83V, which can be set by softwareThe default is 83V, which can be set by software		
Under-voltage alarming threshould		The default is 18V, which can be set by software		
Cooling method		Natural air cooling		
General function	Input specification	4 channel digital input,with COMI terminal, high level:12.5~30VDC, low level:0~5VDC, max frequency:1KHZ, input impedance:5KΩ		
	Output specification	2-channel digital output common COMO terminal, Maximum output current: 100mA		
	Pulse direction control	Pulse + direction , CCW+CW, phase A + phase B (5-24VDC), Input voltage 3.3V-24VDC, Maximum frequency 500KHz		
	Brake	Built-in brake power supply 24V maximum current 0.5A		Built-in brake power supply 24V maximum current 1 A
RS232		Default baudrate setting is 38400， the max. baudrate is 115.2KHz, use Kinco software to communicate with PC, or via free protocol to communicate with controller.		
RS485		The max. baudrate is 115.2KHz, use Modbus RTU protocol to communicate with controller.		
CAN BUS		Support maximum 1MHz baudrate. Communicate with controller via CANopen protocol		
EtherCAT		Support CoE(CiA402 protocol)and CSP/CSV/PP/PV/PT/HM mode， communication speed 100M		
Profinet		Support No. 1 message, No. 3 message, No. 111 message, process object, aperiodic data read and write, etc.		

Note1：□=M：：Single-turn 16-bit communication magneto encoder

Note2：■=A：without brake

=B：brake

Note3：★=L：Communication portRS232, RS485, Pulse

=C：Communication portRS232、CANopen, Pulse

=E：Communication portRS232、EtherCAT

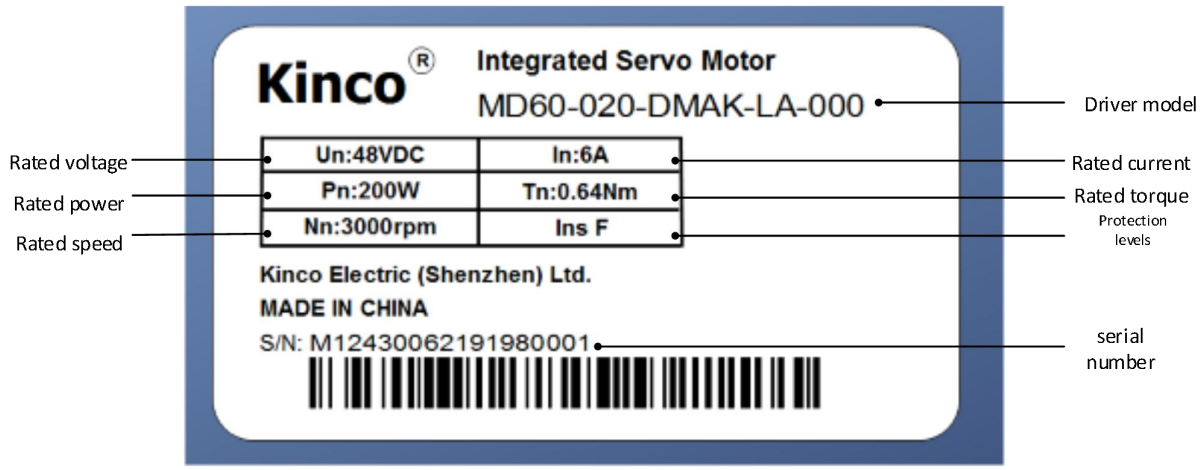
=P：Communication portRS232、Profinet

1.2 Product description

1.1.1 MD naming rule

MD 60- 040 - D M A K - CA - 000 ① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨	
①-Series name MD: Integrated servo motor	⑤-Encoder type M:Magnetoelectric encoder
②-Flange 60:60x60(mm) 80:80x80(mm)	⑥-Brake A: Without brake B: With brake
③-Rated power 020:20*10(W) 040:40*10(W) 075:75*10(W)	⑦-Outgoing shaft style K: With key
④-Supply voltage D:DC48V	⑧-Control mode LA: RS232, RS485, Pulse CA: RS232, CANopen, Pulse EA: RS232, EtherCAT PA: RS232, Profinet
	⑨-Software version 000: Software version number

1.1.2 Nameplate description



Chapter 2 System installment requirements and precautions

2.1 Installation of integrated servo motor

- Ensure that this document is available to design engineers, installers, and personnel responsible for commissioning machines or systems that use the product.
- Please ensure that you always follow the requirements of this document and also consider the documentation for other components and modules.
- Please consider the legal requirements applicable to your destination and:
  - Regulations and standards
  - Test organization and insurance company regulations
  - National specifications

2.1.1 Transportation and saving conditions

- Please ensure that the product is not subjected to more than permitted burdens during transportation and storage, including:
  - Mechanical load
  - The temperature is not allowed
  - Moisture
  - Corrosive gas
- Please store and transport in original packaging, which provides adequate protection against routine problems.

2.1.2 Technology requirements

The general conditions for the correct and safe use of the product must always be observed:

- The connection and environmental conditions specified in the product technical data and the technical requirements of all other connected components. Products are only allowed to operatein accordance with relevant safety procedures if they meet product specifications.
- Follow the instructions and warnings in this document.

2.1.3 Operator’s requirements

This product should only be operated by an electrical engineer who is familiar with the following provisions:

- Installation and operation of electrical control systems
- Applicable regulations for the operation of safe engineering systems
- Applicable provisions for accident protection and occupational safety
- Familiarize yourself with the documentation of the product

2.1.4 Environment requirements

Environment	Conditions
Working Temperature	0℃～40℃（no ice）
Working humidity	Less than 90%RH（no condensation）
Storage temperature	-10℃～70℃（no ice）
Storage humidity	5～95%RH（no condensation）
Installation requirement	Dust-free, dry and lockable (such as electrical cabinets)
Altitude	The rated working altitude is less than 1000 meters above sea level. When the working altitude is higher than 1000 meters, it is necessary to reduce the rated value by 1.5% for every 100 meters of elevation. The maximum working altitude is 4000 meters above sea level.
Vibration	5.9m/s2 below 10～60Hz (not available at resonance point) Running: less than 49m/s2(5G)、Stopping: less than 24.5m/s2(2.5G)
Protection level	IP20
Installation method	Install vertically or horizontally

2.1.5 Precautions

Item	Description
Stain proofing	Please wipe anti-rust agent on the motor's shaft and then make some anti-rust treatments.
Installation method	Improper installation method will cause damage of motor's encoder. Please note the following during the installation process: ● When operators installation pulleys on the servo motor shaft with key, it is necessary to use screw hole. In order to install pulleys, operators need to insert double-headed nail into screw holes and use washers on the surface of coupled end. Then use nuts to fix into pulleys gradually. ● For servo motor shaft with keys, Operator need to use screw hole on the shaft to install. For motors shaft with no key, operators need to use friction coupling or other analogous methods. ● When operators need to disassemble pulleys, operators need to use pulley remover so as to make shaft avoid strong impact of load. ● In order to make it more safe, it is necessary to install protection cover or some analogous equipment in rotation area. For example, pulleys installed on the shaft.
Centering	●When it is connected with machine, please use coupling and make shaft center of servo motor and machine stay in a line. When operators install servo motors, please achieve requirements of centering accuracy. If centering is not accurate, there will be shock and sometimes it will make bearings and encoders.
Installation direction	● Servo motors can be installed in vertical or horizontal direction.

Oil & water solution	When it is used in the occasion with drops, please use after make sure protection level of servo. When oil will drop into shaft penetrating part (beside shaft penetrating part, please choose servo motors with oil seal. The using condition of servo motors with oil seal: ● Make sure the oil level is lower than month of oil seal. ● Please use when oil seal make sure that oil splash degree is good. ● When servo motors are installed in vertical upward direction, please avoid oil accumulating in the month of oil seal.

2.1.6 Oil seal installation instruction

Because the assembly of the oil seal will increase the loss of the motor, it will cause the motor temperature rise to increase by about 5K.The bearing of the motor has a double-sided dust proof effect, if it is not necessary to install the oil seal, it is not recommended to install the oil seal. The fit friction interference between the motor shaft and the oil seal is about 0.25-0.3mm, and the surface roughness of the shaft is 0.8.Before assembling the oil seal, please ensure that the installation hole and the oil seal are free of debris, oil, dust, etc.When assembling, please fill the lip of the oil seal with high temperature grease (recommended HR12 of the Great Wall, grease with temperature resistance of 150 degrees) to strengthen the performance of lubrication and temperature resistance, and increase the sealing and waterproof effect of the oil seal. When paying attention to water and oil proofing, the oil seal has a self-tightening spring side (that is, a side with a groove) facing outwards. Please refer to the following steps to install the oil seal correctly.

1.Apply high temperature lubricating oil evenly to the sealing ring of the oil seal lip.

2.Turn the fluted side of the oil seal outward to ensure that the oil seal is perpendicular to the machine shaft and push the oil seal into the chamber using an even application of force.

3.After successful installation, check whether the oil seal is tilted. The oil seal should be fitted with the motor bearing cover, and the lip of the oil seal should be completely closed to ensure the tightness of the oil seal.

Warning

- Please strictly follow the instructions in this manual to install the servo system correctly. It can help you set up and operate the drive correctly and achieve the best performance of the drive

2.1.7 Scope of application and certification

The certificate and declaration of conformity of this product can be found at the following website:

The current version of the product has passed CE EU certification. Marked as follows:

This product meets the basic requirements of the EU's "New Methods for Technical Coordination and Standardization"directive

test standards：EN 61000-6-4：2007+A1:2001&EN 61000-6-2:2005

2.2 Mounting dimension

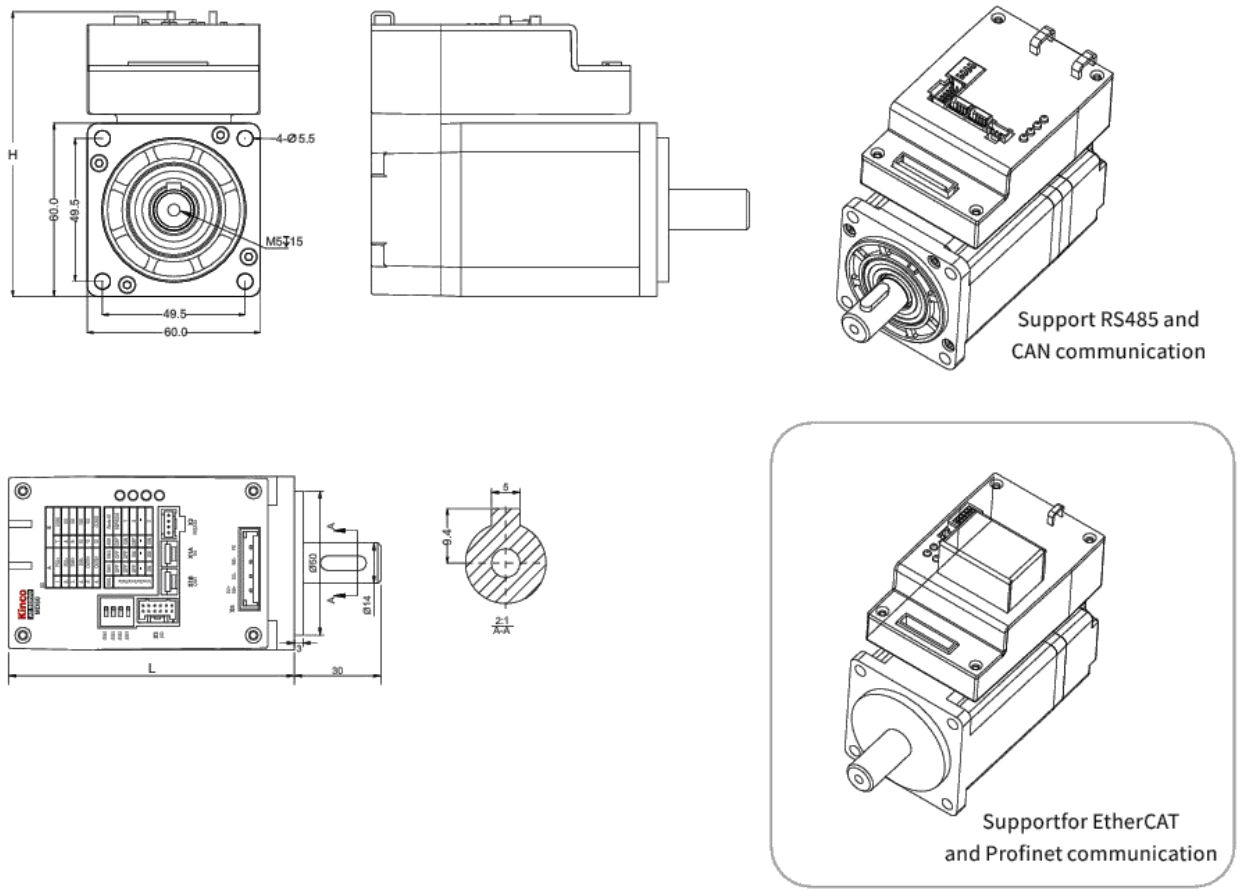


Figure 2-1 Installation dimensions of a 60-flange motor

Model	With brake	Weight (KG)	H (mm)	L (mm)
MD60-020-DMAK-LA-000		1.2	98.6	99.2±1.5
MD60-020-DMAK-CA-000				
MD60-020-DMAK-EA-000		1.25	113.1	
MD60-020-DMAK-PA-000				
MD60-020-DMBK-LA-000	√	1.6	98.6	129.2±1.5
MD60-020-DMBK-CA-000				
MD60-020-DMBK-EA-000		1.65	113.1	
MD60-020-DMBK-PA-000				
MD60-040-DMAK-LA-000		1.6	98.6	125.2±1.5
MD60-040-DMAK-CA-000				
MD60-040-DMAK-EA-000		1.65	113.1	
MD60-040-DMAK-PA-000				
MD60-040-DMBK-LA-000	√	2	98.6	155.2±1.5
MD60-040-DMBK-CA-000				
MD60-040-DMBK-EA-000		2.05	113.1	
MD60-040-DMBK-PA-000				

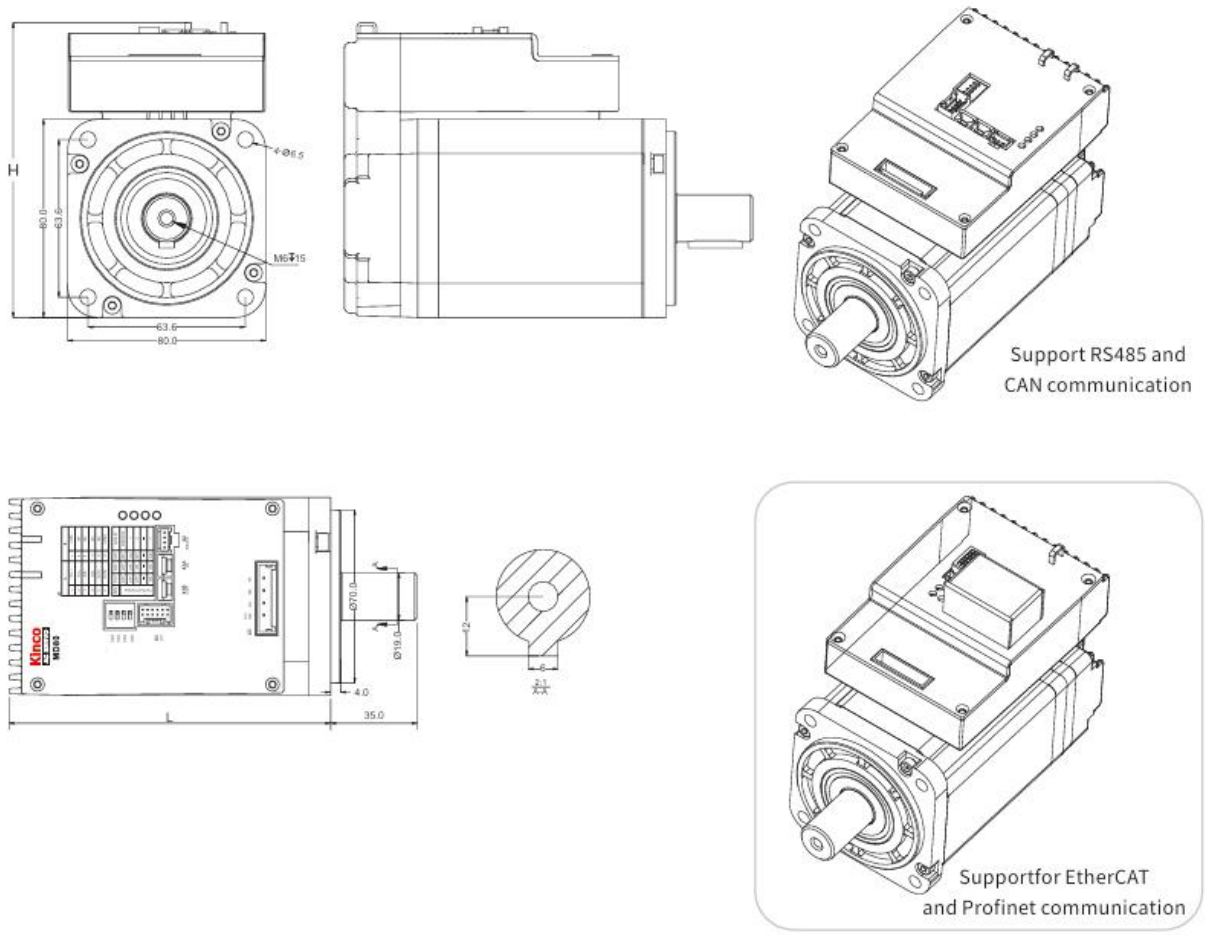


Figure 2-2 Installation dimensions of a 80-flange motor

Model	With brake	Weight（KG）	H（mm）	L（mm）
MD80-075-DMAK-LA-000		2.9	119.1	130±1.5
MD80-075-DMAK-CA-000				
MD80-075-DMAK-EA-000		2.95	133.6	
MD80-075-DMAK-PA-000				
MD80-075-DMBK-LA-000	√	3.5	119.1	164.2±1.5
MD80-075-DMBK-CA-000				
MD80-075-DMBK-EA-000		3.55	133.6	
MD80-075-DMBK-PA-000				

2.3 Servo motor torque-speed curve

2.3.1 200W servo motor torque-speed curve

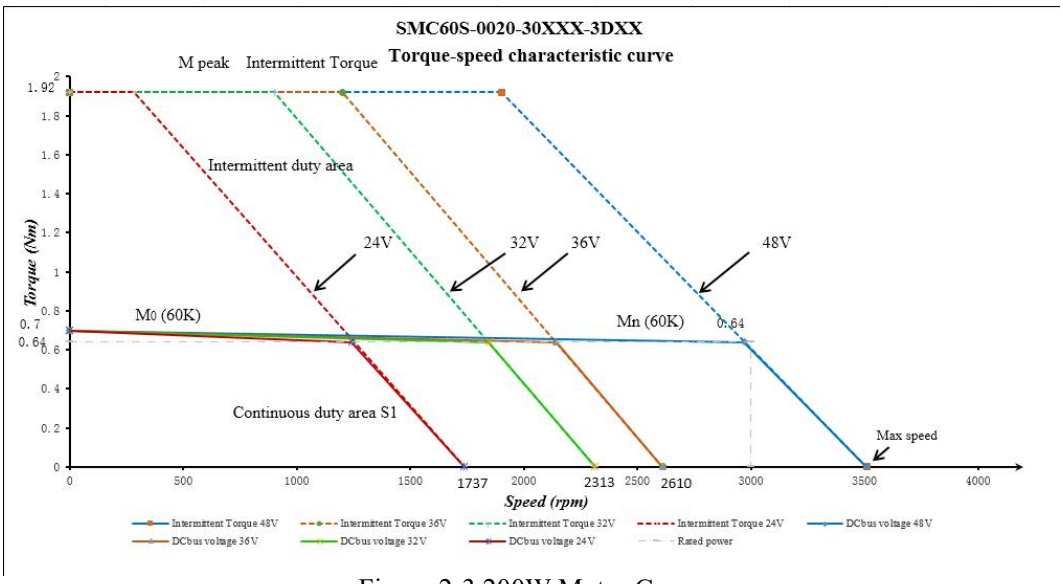


Figure 2-3 200W Motor Curve

2.3.2 400W servo motor torque-speed curve

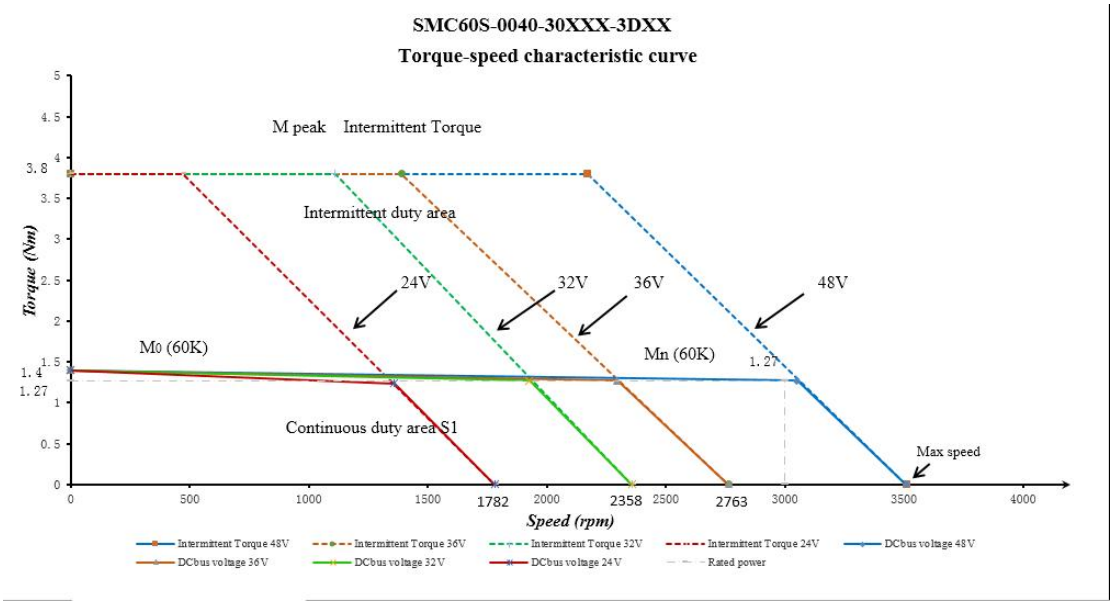


Figure 2-4 400W Motor Curve

2.3.3 750W servo motor torque curve

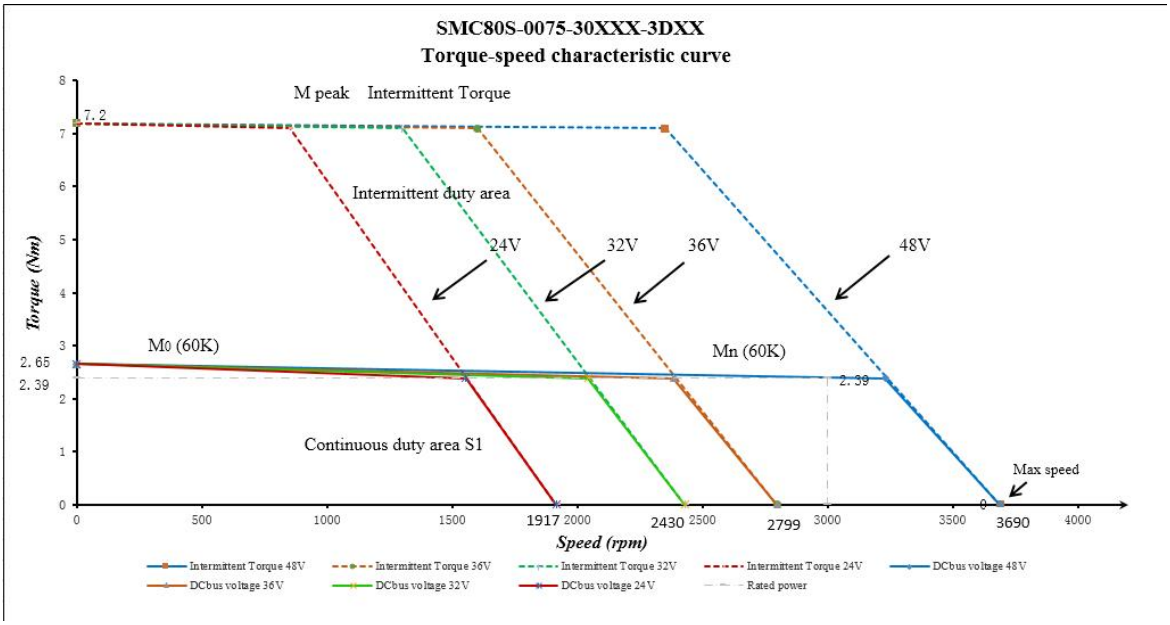


Figure 2-5 750W Motor Curve

Chapter 3 Interface and wiring

3.1 integrated servo motor components name

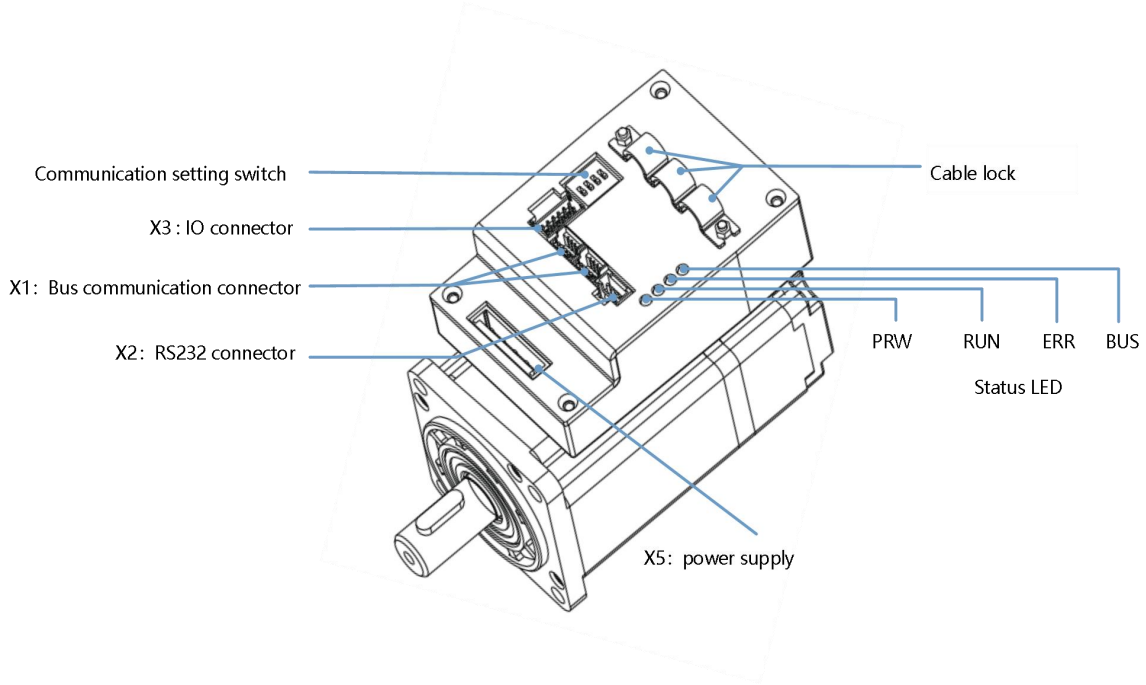


Figure 3−1 MD Series Interface Definition

3.2 External wiring

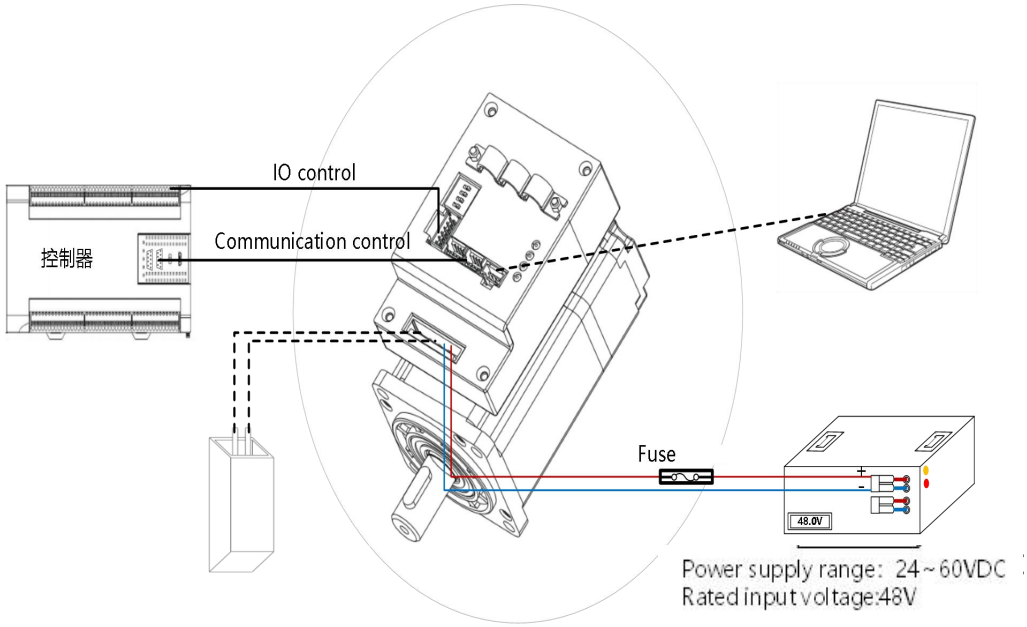


Figure 3−2 MD series external wiring diagram

Note:

- The interface definition for different power products in MD series is the same, with MD60(200W) as an example in figure. 3-1 and figure 3-2
- Fuses should be selected according to table 3-1. please refer to the appendix for brake resistance specification recommendations.

Table 3-1 Fuse specification recommendation

Integrated servo motor models	power(W)	Fuse reference specification
MD60-020-DMAK-□A-000 MD60-020-DMBK-□A-000	200	20A/58VDC
MD60-040-DMAK-□A-000 MD60-040-DMBK-□A-000	400	20A/58VDC
MD80-075-DMAK-□A-000 MD80-075-DMBK-□A-000	750	40A/58VDC

3.3 Interface description

3.3.1 Bus communication interface （X1）

Table 3-2 MD-LA/CA X1Interface definition

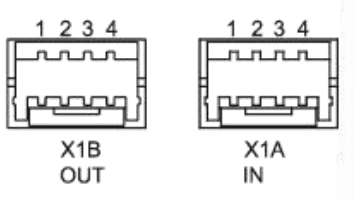
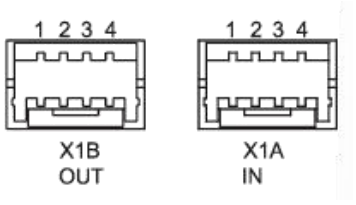
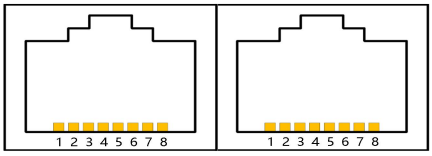
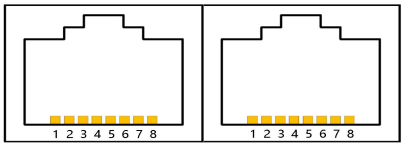
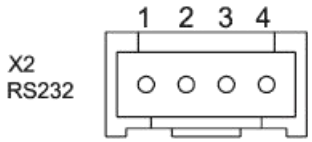
<table>
<tr><td>Bus Type</td><td colspan="2">CANopen</td><td colspan="2">RS485</td></tr>
<tr><td>Applicable Products</td><td colspan="2">MD60-020-DMAK-CA-000
MD60-040-DMAK-CA-000
MD80-075-DMAK-CA-000
MD60-020-DMBK-CA-000
MD60-040-DMBK-CA-000
MD80-075-DMBK-CA-000</td><td colspan="2">MD60-020-DMAK-LA-000
MD60-040-DMAK-LA-000
MD80-075-DMAK-LA-000
MD60-020-DMBK-LA-000
MD60-040-DMBK-LA-000
MD80-075-DMBK-LA-000</td></tr>
<tr><td rowspan="6">Pin Definition</td><td colspan="2"> 
</td><td colspan="2"> 
</td></tr>
<tr><td>Pin No.</td><td>Pin name</td><td>Pin No.</td><td>Pin name</td></tr>
<tr><td>1</td><td>GND</td><td>1</td><td>GND</td></tr>
<tr><td>2</td><td>GND</td><td>2</td><td>GND</td></tr>
<tr><td>3</td><td>CAN_L</td><td>3</td><td>485+</td></tr>
<tr><td>4</td><td>CAN_H</td><td>4</td><td>485-</td></tr>
</table>

Table 3-3 MD-EA/PA X1Interface definition

<table>
<tr><td>Bus Type</td><td colspan="2">ETherCAT</td><td colspan="2">Profinet</td></tr>
<tr><td>Applicable Products</td><td colspan="2">MD60-020-DMAK-EA-000
MD60-040-DMAK-EA-000
MD80-075-DMAK-EA-000
MD60-020-DMBK-EA-000
MD60-040-DMBK-EA-000
MD80-075-DMBK-EA-000</td><td colspan="2">MD60-020-DMAK-PA-000
MD60-040-DMAK-PA-000
MD80-075-DMAK-PA-000
MD60-020-DMBK-PA-000
MD60-040-DMBK-PA-000
MD80-075-DMBK-PA-000</td></tr>
<tr><td rowspan="5">Pin Definition</td><td colspan="2">  </td><td colspan="2">  </td></tr>
<tr><td>Pin No.</td><td>Pin name</td><td>Pin No.</td><td>Pin name</td></tr>
<tr><td>1</td><td>TD+</td><td>1</td><td>TD+</td></tr>
<tr><td>2</td><td>TD-</td><td>2</td><td>TD-</td></tr>
<tr><td>3</td><td>RD+</td><td>3</td><td>RD+</td></tr>
<tr><td>6</td><td>RD-</td><td>6</td><td>RD-</td></tr>
</table>

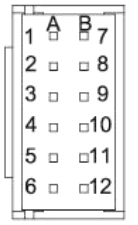
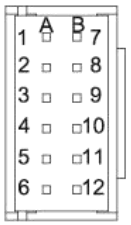
3.3.2 RS232 port（X2）

Table 3-4 X2Interface definition

<table>
<tr><td rowspan="5">  </td><td>Pin No.</td><td>Pin name</td><td>Pin function</td></tr>
<tr><td>1</td><td>GND</td><td>Signal Ground</td></tr>
<tr><td>2</td><td>GND</td><td>Signal Ground</td></tr>
<tr><td>3</td><td>TX</td><td>Drive send data</td></tr>
<tr><td>4</td><td>RX</td><td>Driver received data</td></tr>
</table>

3.3.3 External input&output（X3）

Table 3-5 X3Interface definition

<table>
<tr><td rowspan="13"> 
Suitable for LA and CA
 
Suitable for EA and PA</td><td>Pin No.</td><td>Pin name</td><td>Pin function</td></tr>
<tr><td>1</td><td>PUL+</td><td rowspan="4">Pulse input function
Input voltage：3.3V～24V
Maximum frequency：500KHz</td></tr>
<tr><td>2</td><td>PUL-</td></tr>
<tr><td>3</td><td>DIR+</td></tr>
<tr><td>4</td><td>DIR-</td></tr>
<tr><td>5</td><td>OUT1+</td><td rowspan="2">Digital signal output
Maximum output current：100mA</td></tr>
<tr><td>6</td><td>OUT2+</td></tr>
<tr><td>7</td><td>COMI</td><td>Input common terminal</td></tr>
<tr><td>8</td><td>IN1</td><td rowspan="4">Digital signal input
high level：12.5VDC~30VDC
Low level：0VDC~5VDC
Input frequency：<1KHz</td></tr>
<tr><td>9</td><td>IN2</td></tr>
<tr><td>10</td><td>IN3</td></tr>
<tr><td>11</td><td>IN4</td></tr>
<tr><td>12</td><td>COMO</td><td>Output common</td></tr>
</table>

Note

- Please refer to appendix 4 for wire gauges and wire making methods of X1, X2 and X3 terminals.
- The above figure shows the definition of the interface on the driver side, not the communication cable. Please be careful to avoid the wrong welding line.
- MD-EA/MD-PA drive has no pulse input function， pin1/2/3/4 is NC.and the brake model 1 ,2 , 3,4,6 are empty

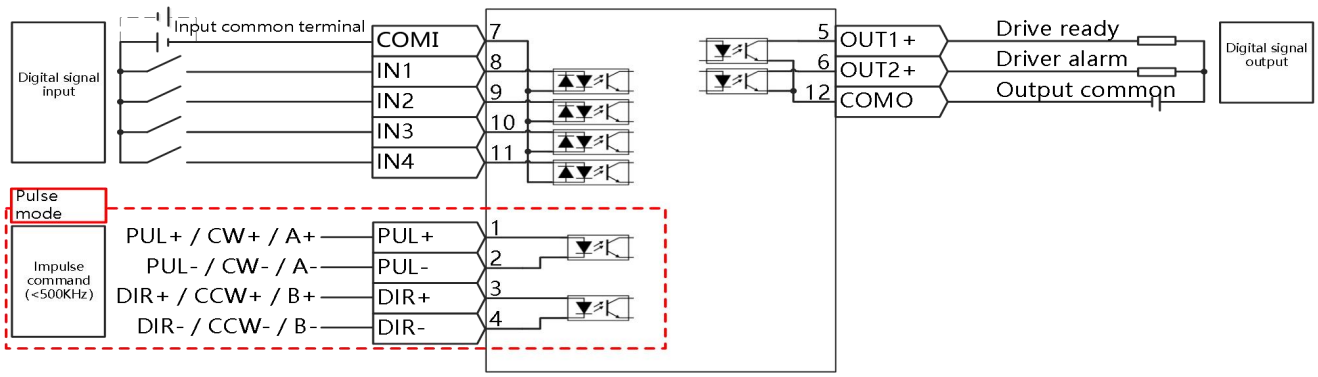


Figure3−3 MD seriescontrol wiring diagram

Note

- Figure 3-3 shows the wiring with default IO function. More IO functions can be defined by the Kinco servo software. For more details on IO functions, please refer to the relevant sections.
- For digital output, Figure 3-3 only shows NPN connection, and Figure 3-4 shows PNP connection

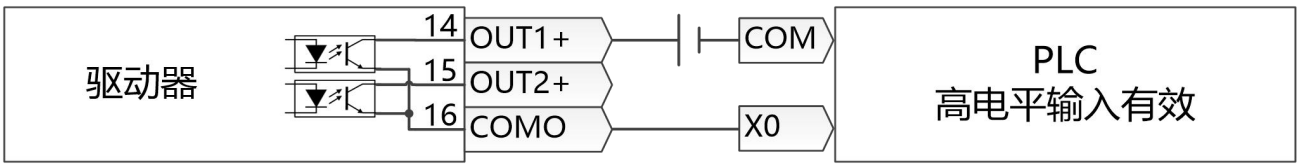
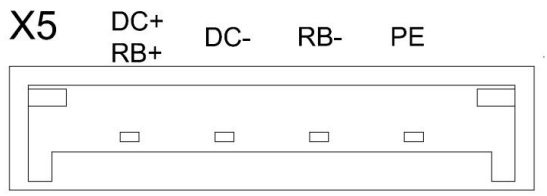


Figure3–4 PNP input wiring

3.3.4 Power interface（X5）

Table 3-6 X5 Interface Definition



Pin name	Pin function
DC-	DC power input（24-60V）
DC+	
RB-	External brake resistor
RB+	
PE	ground

Table 3-7 Specification for power cable

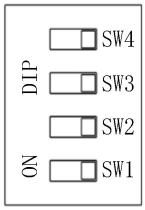
Model	X5 Power interface wiring specifications
MD60（200W）	Crimp terminal wiring specification range：0.21~1.31mm² （24~16AWG） Recommended conductor cross-sectional area：0.75~1.31mm² （18~16AWG） Recommended stripping cable length：8~9mm
MD60（400W）	Crimp terminal wiring specification range：0.21~1.31mm² （24~16AWG） Recommended conductor cross-sectional area：1.3~1.5mm² （16~15AWG） Recommended stripping cable length：8~9mm
MD80（750W）	Crimp terminal wiring specification range：0.21~3.3mm² （24~12AWG） Recommended conductor cross-sectional area：2.5~3.3mm² （13~12AWG） Recommended stripping cable length：12~13mm

Note

- Please use the power cable with shielding layer, twist and fold the wire core into bundles and insert it into the crimping terminal. After the cable is connected, pull the cable to confirm that the wire is firmly connected with the terminal, and there is no flying wire or touching wire on the adjacent cable.
- The bending radius of the cable shall be more than 10 times of the outer diameter of the cable itself, and frequent bending of the cable shall be avoided.

3.3.5 Dip switch and Indicators

Table 3-8 Dip switch

	Pin name	Pin function
	SW1	The equipment station number is determined by the BCD code composed of SW1-SW3. Restart of the driver takes effect after dip switch changes. When SW1-SW3 are all OFF, the driver reads the equipment station number in EEPROM.
	SW2	
	SW3	
	SW4	When SW4 is ON, turn on the terminal resistance

note

- Integrated servo system factory default SW1 is ON, other dial codes are OFF.
- EA/PA driver has no dip switch.

Table 3-9 MD-LA/CA Indicator light

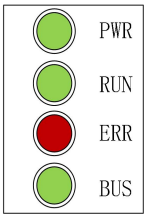
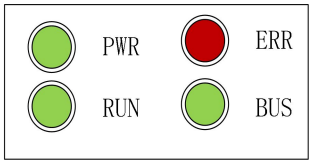
	Pin name	Pin function
	PWR	The driver has been POWERed on, and the power lamp is always on.
	RUN	When the drive is ready, it is always on and associated with out3.
	ERR	When the driver reports an error, it is in a normally bright state and is associated with out4.
	BUS	When there is message transmission on CANopen bus, it will flash, and the flashing frequency is related to the message transmission speed.

Table 3-10 MD-EA/PA Indicator light

	Pin name	Pin function
	PWR	The driver has been POWERed on, and the power lamp is always on.
	RUN	When the drive is ready, it is always on and associated with out3.
	ERR	When the driver reports an error, it is in a normally bright state and is associated with out4.
	BUS	When there is message transmission on CANopen bus, it will flash, and the flashing frequency is related to the message transmission speed.

Note

- In the software, out3 defines drive readiness by default and out4 defines drive failure by default. When the RUN and ERR lights do not illuminate, please check whether the default definition has been modified.

Chapter 4 KincoServo software introduction

This chapter will introduce how to use KincoServo software adjust and configure servo driver.

Figure 4−1 Software main window

4.1 Fast start

4.1.1 Language configuration

Language can be switched between English and Chinese via menu item **Tools**->**Language**.

4.1.2 Opening and saving project files

Create a new project file via menu item File->New, or by clicking the button.

Open an existing project via menu item File->Open, or by clicking the button and selecting a .kpjt file

Save a project via menu item File->Save, or by clicking the button and saving as a .kpjt file

Note

Only the windows (object list, scope etc.) are saved-parameters in the controller can’ t be saved in this way.

4.1.3 Start communication

Click menu item **Communication->Communication settings**. The following window appears:

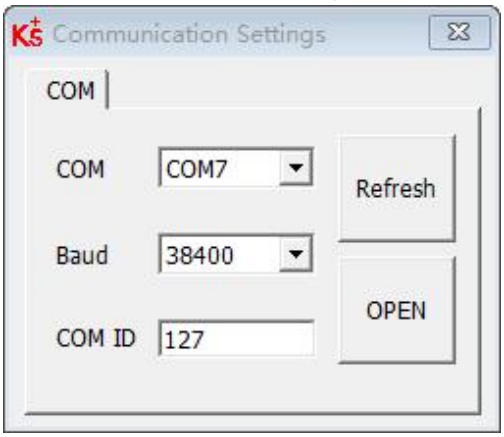


Figure 4–2 Communication setting

Select the right COM port (if it′ s not shown click the ″Refresh″ button), baud rate and COM ID (Node ID), and then click the "OPEN″ buton.

Once communication has been established with the controller, communication can be opened or closed by clicking the button  .

4.1.4 Node ID and baud rate

The communication ID (station ID) can be set using the DIP switch on the product. For the setting method, see the silk-screen printing description.

The drive station number can be set through the menu bar "Drive" -> "Drive Property".

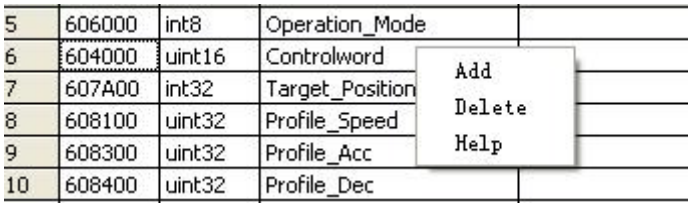
Parameter adress	Type	Name	Value	Unit
100B0008	Usigned8	Node_ID		DEC
2FE00010	Usigned 16	RS232_baudrate		Baud

Note

- The device station number and baud rate take effect only after the control parameters are stored and the drive is restarted.
- When KincoServo+ is connected through the RS232 baud rate of 115200, due to the high frequency of signal transmission, there are certain requirements on the communication environment. If there is a communication drop or frame loss, it is recommended to reduce the communication baud rate and reconnect the software
- If the serial port is not displayed after click Refresh, check whether the driver is not installed.

4.1.5 Object (add，delete，help)

Open any window with an object list, move the mouse pointer to the object item and right click. Thefollowing selection window appears:



Click **Add** and double click the required object from the **Object Dictionary**. The selected object is then added to the list.
Click **Delete**. The selected object is removed from the list.
Click **Help** to read a description of the selected object in the **Object Dictionary**.

4.2 Init, save and reboot

Click **Controller->Init Save Reboot**. The following window appears:

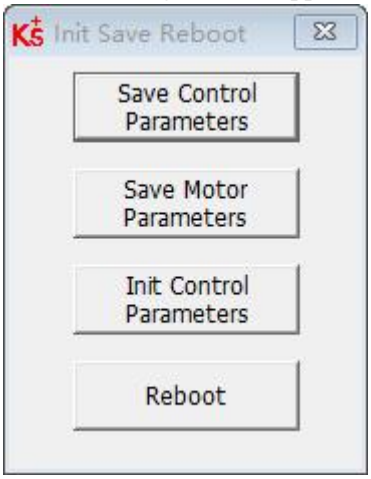


Figure 4–3 Initialize, save, reboot

Click the corresponding item to finish the necessary operation.

Note

After completing the **Init Control Parameters**, the Save Control Parameters and Reboot buttons must be clicked to load the default control parameters to the controller.

4.3 Firmware update

In general, the firmware of the drive is always the latest version, but if the drive firmware needs to be updated for some reason, please go to the menu bar "**Controlle**r" -> "**Load Firmware** ".

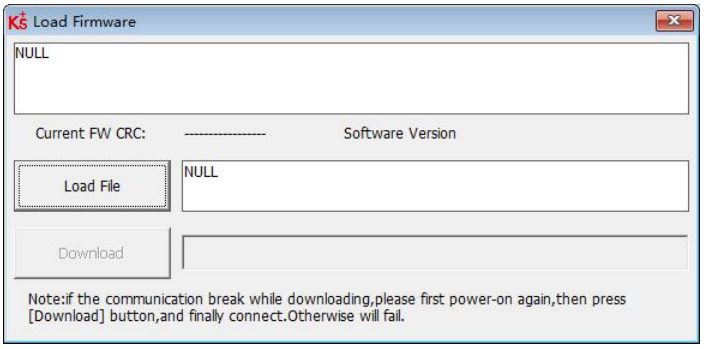


Figure 4-4 Load Firmware

Click "Load File" to select the firmware file (.kinco), then click "Download" to start updating the drive firmware.

Note

If the download is stopped for some reason, first power off, then power on the drive, select the firmware version and start downloading, and finally turn on the communication and connect to the host computer.

4.4 Read/write controller configuration

For a large number of the same applications, in order to avoid setting drive parameters one by one, you can use this feature to configure drives.

4.4.1 Read setting from controller

Click **Tools->R/W Controller Configuration->Read Settings** from Controller or click the button. The following window appears.

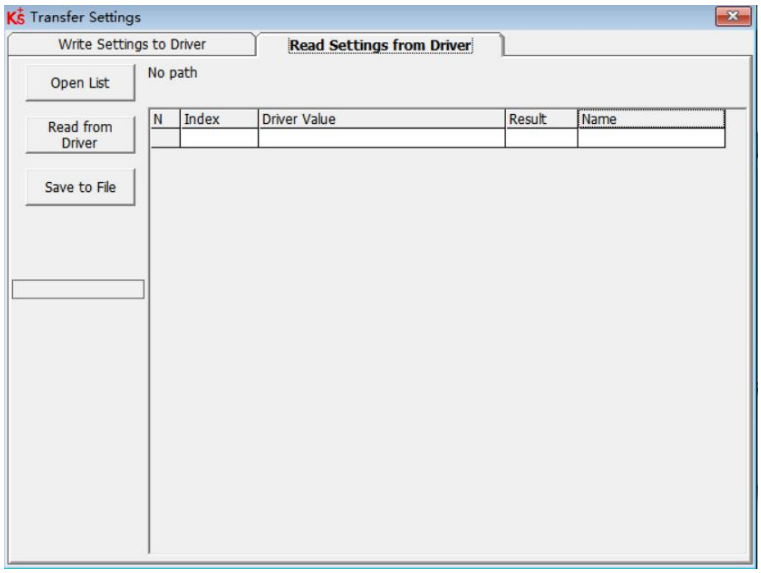


Figure 4-5 Read driver configuration

- Click Open List to select a parameter list file (.cdo). The parameter appears in the window.
- Click Read Settings from Driver to get the Drive Value and Result, and then click Save to File to save the setting as a .cdi file

Note

If the object does not exist in the drive, the result will be "False" and will be highlighted in red, and only read parameters with a result of "true" will be saved in the.cdi file.

4.4.2 Write settings to controller

Click Tools->R/W Controller Configuration->Write Settings to Controller or click the button.

The following window appear  button, the following window appears:

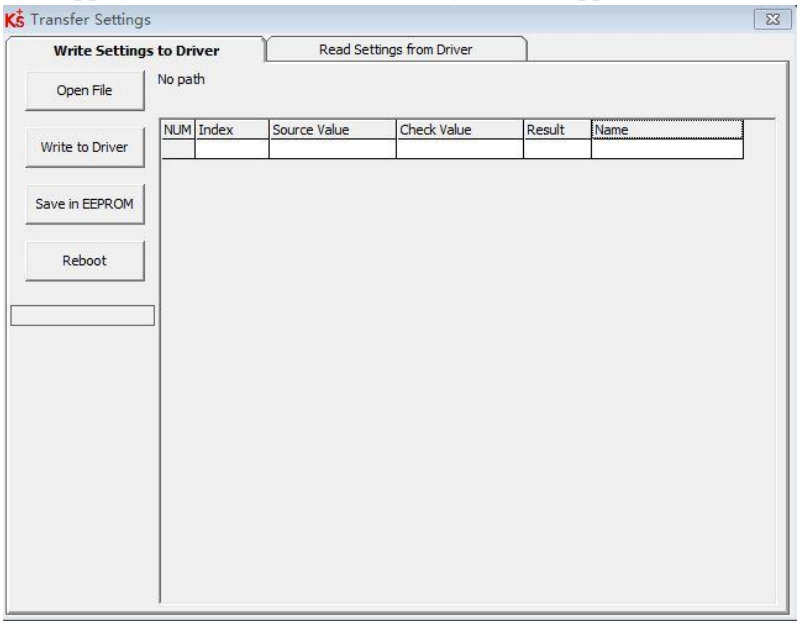


Figure 4–6 Write driver configuration

Click **Open File** to select a parameter settings file (.cdi). The parameter settings appear in the window.

Click **Write to Controller** to get the **Check Value** and **Result**. The “False” **Result** means the value has not been written successfully, probably because the object doesn’t exist in the controller.

Click **Save in EEPROM** and **Reboot** to activate all parameters.

Note

Before write setting to driver, please cancel driver enable. If driver is enabled, some object cannot be written.

4.5 Digital IO functions

Click menu item Controller->Digital IO Functions or click the button  . The following windowappears. Function and polarity are shown. Shown as default function and polarity.

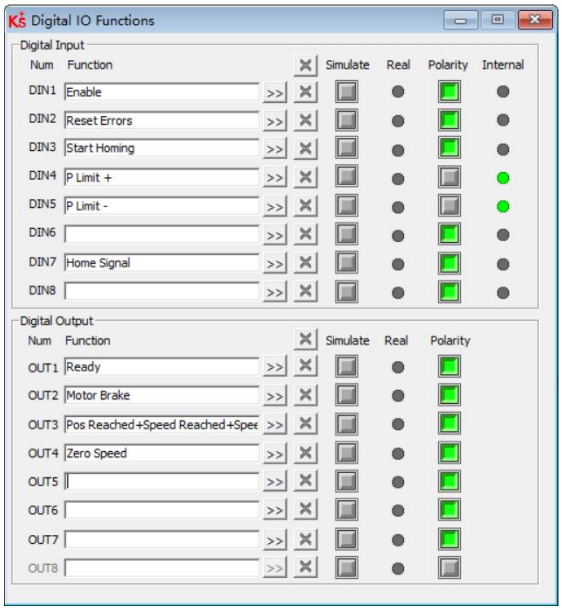


Figure 4–7 Digital IO

4.5.1 Digital input

Figure 4–8 Digital input

Function：Click  to select Din function setting click  to delete the DIN function

Real：Shows the real digital input hardware status.

Actual input: Displays the actual digit input status

Polarity：  indicates a high level input, and the effective input is 1；  indicates a low level input, and the effective input is 1.

Real：Shows the real digital input hardware status.

 means "active", logic status of the digital input is means "not inactive", logic status of the digital input is 0.

DIN function	Description
Enable	Controller enabling 1: Enable controller = Din control word selection(2020.0F) 0: Disable controller = 0x06
Reset errors	Set the Controlword to reset errors, active edge（bit7） = 1
Operationg mode	Operation_mode selection 1:Operation_mode = EL.Din_Mode1 (2020.0E) 0:Operation_mode = EL.Din_Mode 0 (2020.0D)
Kvi off	Velocity control loop integrating gain off
P limit+	Positive/negative limit switch, normally OFF, Din effective input = 0 indicates that the motor has reached the limit position
P limit-	
Home signal	Home switch signal for homing
Invert Direction	Inverts command direction in the velocity and torque mode
Din Speed index 0	Din_Speed Index in the DIN speed mode
Din Speed index1	
Din Speed index2	
Din position index0	Din position index in Din position mode
Din position index1	
Din position index2	
Activate Command	Actives the position command.Controls bit 4 of the Controlword, e.g. Controlword=0x2F->0x3F
Pre enable	For safety reasons, pre-enable can serve as a signal for indicating whether or not the entire system is ready， 1：Drive can be enabled；0：Drive can not be enabled
Cleaning pulse	Clear the number of pulses the drive has received but not completed
Pause	1：Pause motor，code: bit8 = 1 0：Motor continues to execute unfinished instructionsNote: Pause is not supported in torque mode

Note

Relative/Absolute position control select (2020.0F) default setting is 0x2F. For Control word definition, please refer to Chapter 6.1.

4.5.2 Digital output

Figure 4–9 Digital output

Function：Click >> to select Din function setting click ✕ to delete the DIN output function

Real：Shows the real digital output hardware status.

Actual output: Displays the actual digital output state, which is the result of the comprehensive action of simulation, polarity and logic state, ● indicates the digital output is ON，● indicates the digital output is OFF

Polarity：■ indicates that if the logical status is 1, the actual output is ON；■ indicates that if the logical status is 0, the actual output isON。

Effective output: the result of simulation, actual output and polarity action; ● indicates activation,the logical status of the corresponding function is 1. ● indicates not activation,the logical status of the corresponding function is 0.

OUT function	Description
Ready	Controller is ready to be enabled
Error	Controller error
Pos reached	In position mode, when the difference between the actual position and the target position is less than the target position window (6067.00), and the duration is greater than or equal to the position window time (6068.00), the output position to the function is displayed
Zero speed	When the absolute value of the actual velocity -ms (60F9.1A) is less than or equal to the zero velocity window (2010.18), and the duration is greater than or equal to the zero velocity time (60F9.14), the zero velocity function is output
Motor Brake	The motor brake control output signal can be used to connect an external relay that controls the motor lock. If the brake motor is used, this function must be set, otherwise it will damage the motor. An effective output of green indicates that the lock is opened, and an effective output of gray indicates that the lock is closed.
Speed reached	Output speed to function when the speed error (60F9.1C) is less than the speed to window (60F9.0A)
Index signal occurrence	Motor index signal appears
Speed limit	In torque mode actual speed reached Max_Speed(607F.00)
Motor lock shaft	Drive is enabled. Motor locks shaft
Position limit	Position limit function is active
Home Found	Home found
Multifunction signal 0	Din multi-segment position control position to output function
Multifunction signal 1	
Multifunction signal 2	
Torque reaches limit	When the actual torque (60F5.08) reaches the reference (60F5.06) and the duration exceeds the filter time (60F5.07), the output torque reaches the limit, and the torque reaches the reference (60F5.06) is set to 0, indicating that the torque limit detection is not enabled .

4.6 Scope

In the operation process, if the operation effect of the equipment can not meet the requirements, or other accidents occur, you can use the oscilloscope to analyze the problem.

Click the  button in the software to turn on the oscilloscope.

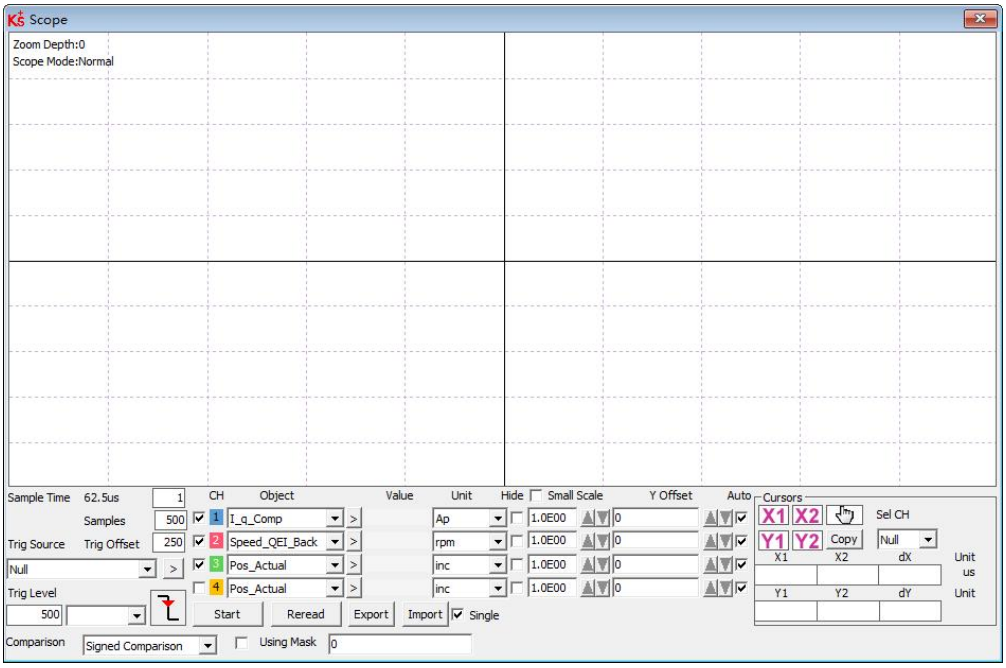


Figure 4-10 scope display

Sample time： Data collection period. If the value is set to 1, data is collected every 62.5us.

Samples： This parameter indicates how many data are collected during the sampling. If the value is set to 500, 500 data are collected.

Trig offset： The number of samples before the trigger source is triggered.

Trig source and Trig level： The trigger condition is set in Figure 5-13 to start collecting data when the effective target current q rises to 100DEC. DEC is the internal unit and can be switched to the current unit.

Trigger edge： clicking could change it to rising edge trigger 、 falling edge trigger or rising and falling edge trigger .

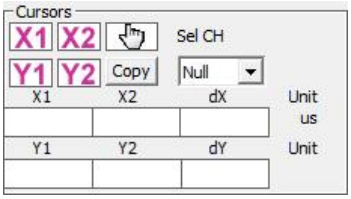
Object**: Maximum 64-bit length data can be taken in one sample, e.g.: 2 Int32 objects bit or 4 Int16 objects.**

Single： Single means sample for one trigger event only； Single means sample continuously.

Zoom in/ Zoom out the oscillogram： Hold down the right mouse button and drag the mouse down to the right to zoom in, and to the upper left to zoom out.

Cursors： Up to 4 scope cursors can be selected by clicking X1 X2 Y1 Y2 button. The scope cursors appear in the oscillogram ,and select the channel you want to observe from the Channel Selection drop-down menu.

Moving Cursors： Hold down the left mouse button, drag the cursor to move, the sampled data, the difference between X1X2 and Y1Y2 will be displayed in the following area：



Copy： Copy the sampled data to the clipboard, you can open excel and paste the data directly.

Move ： The button turns to yellow means the moving works. You can drag the wave by holding down the left mouse button in the oscilloscope .

Export: Exports the sampled data as a .scope file.

Import: Imports a .scope file and shows the oscillogram in the scope window.

Reread： Rereads the last scope data out of the controller and shows the oscillogram in the scope window.

Auto： If the checkbox Auto is checked, the oscillogram is auto-scaled.If Auto is not checked, the oscillogram is scaled by scale and offset value in following

Scale and offset value can be increased by pressing the △ button, and can be reduced by pressing the ▽ button. If Small scale checkbox is checked, scale value changing step is changed to 10% as before.

Oscilloscope mode： On the upper left side of the oscillogram the Scope Mode "normal " or "import" is shown.
-Normal: All buttons on the oscilloscope are available
-Import: The oscillogram is imported from the.scope file. In this mode, the start and reread data buttons are disabled, and you can exit the import mode as prompted by the software.

4.7 Error display and error history

Error： Click "Controller" -> "Error display" , or click  button (which turns red  if an error occurs) , The Error Display window appears

Error history： Click "Controller" -> "Error history" , The error history window will pop up and display the last 8 error messages, including error word, bus voltage, speed, current, temperature, working mode, and power tube status.

Error_state information：

Bit	Error name	Error code	Description
0	Extended Error		Refer to object "Error_State 2"(2602.00)
1	Encoder not connected	0x7331	No communication encoder connected
2	Encoder internal	0x7320	Internal encoder error
3	Encoder CRC	0x7330	Communication with encoder disturbed
4	Controller Temperature	0x4210	Heatsink temperature too high
5	Overvoltage	0x3210	DC bus overvoltage
6	Undervoltage	0x3220	DC bus undervoltage
7	Overcurrent	0x2320	Power stage or motor short circuit
8	Chop Resistor	0x7110	Overload, brake chopper resistor
9	Following Error	0x8611	Max. following error exceeded
10	Low Logic Voltage	0x5112	Logic supply voltage too low
11	Motor or controller IIt	0x2350	Motor or power stage IIt error
12	Overfrequency	0x8A80	Pulse input frequency too high
13	Motor Temperature	0x4310	Motor temperature sensor alarm
14	Encoder information	0x7331	No encoder connected or no encoder communication reply
15	EEPROM data	0x6310	EEPROM checksum fault

Error_state 2 information：

Bit	Error name	Error code	Description
0	Current sensor	0x5210	Current sensor signal offset or ripple too large
1	Watchdog	0x6010	Software watchdog exception
2	Wrong interrupt	0x6011	Invalid interrupt exception
3	MCU ID	0x7400	Wrong MCU type detected
4	Motor configuration	0x6320	No motor data in EEPROM / motor never configured
5	Reserved		
6	Reserved		
7	Reserved		
8	External enable	0x5443	DIN "pre_enable" function is configured, but the DIN is inactive when the controller is enabled / going to be enabled
9	Positive limit	0x5442	Positive position limit (after homing) – position limit only causes error when Limit_Function (2010.19) is set to 0.
10	Negative limit	0x5441	Negative position limit (after homing) position limit only causes error when Limit_Function(2010.19) is set to 0.
11	SPI internal	0x6012	Internal firmware error in SPI handling
12	Reserved		
13	Closed loop direction	0x8A81	Different direction between motor and position encoder in closed loop operation by a second encoder.
14	Reserved		
15	Master counting	0x7306	The main encoder index signal is abnormal

Note

- Click on the menu bar in the software interface “Help” -> “error code”, Error code description can be opened.

Chapter 5 Operation mode

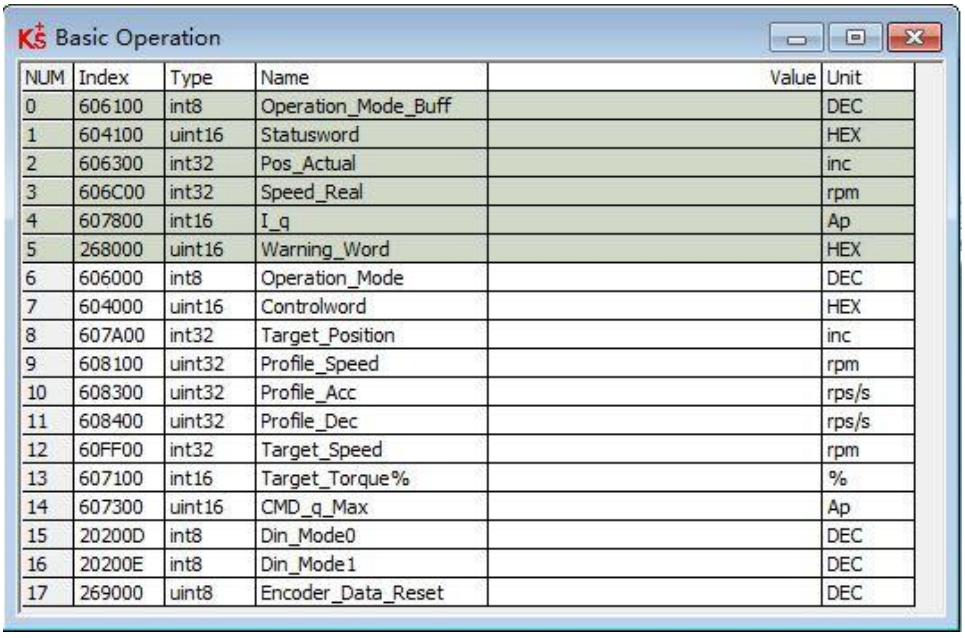
5.1 Velocity mode (-3, 3)

The speed mode has two modes: 3 and -3. The speed mode can be controlled by external I/0 control and internal instruction writing.

Table 5–1 Velocity mode

Internal address	Bit	Name	Description	Value
60600020	Integer8	Operation_Mode	-3: Immediate speed mode, the actual speed will immediately reach the target speed; 3: Speed mode with acceleration and deceleration, the actual speed will be accelerated to the target speed;	-3 and 3
60400010	Unsigned16	Operation_Mode	0x0F Enables the drive 0x06 Loosen the shaft and disable the drive	0x0F
60FF0020	Integer32	Target_Speed	Target velocity	User defined
60830020	Unsigned32	Profile_Acc	It takes effect in mode 1 and mode 3	Default 100rps/s
60840020	Unsigned32	Profile_Dec	It takes effect in mode 1 and mode 3	Default 100rps/s

In software "**Basic operation**" window, we can find these parameters and set, on the 6th, 7th, 10th, 11th, 12th, respectively.



NUM	Index	Type	Name	Value	Unit
0	606100	int8	Operation_Mode_Buff		DEC
1	604100	uint16	Statusword		HEX
2	606300	int32	Pos_Actual		inc
3	606C00	int32	Speed_Real		rpm
4	607800	int16	I_q		Ap
5	268000	uint16	Warning_Word		HEX
6	606000	int8	Operation_Mode		DEC
7	604000	uint16	Controlword		HEX
8	607A00	int32	Target_Position		inc
9	608100	uint32	Profile_Speed		rpm
10	608300	uint32	Profile_Acc		rps/s
11	608400	uint32	Profile_Dec		rps/s
12	60FF00	int32	Target_Speed		rpm
13	607100	int16	Target_Torque%		%
14	607300	uint16	CMD_q_Max		Ap
15	20200D	int8	Din_Mode0		DEC
16	20200E	int8	Din_Mode1		DEC
17	269000	uint8	Encoder_Data_Reset		DEC

Figure 5-1 “Basic operation” window

5.1.1 DIN speed mode introduction

First, when using Din speed mode, at least one of Din Vel index 0, Din Vel index 1 and DIN Vel index 2 must be defined in the I/O configuration as the switching signal of the speed segment.

The DIN speed object window in the PC software can be accessed via menu item **Controller->Control Modes->DIN Speed Mode.**

Table 5–2 DIN speed mode

Internal address	Bit	Name	Description	Value
20200520	Integer32	Din_Speed[0]	The velocity command is specified via Din_Speed[x]. x is the BCD code of Bit 0: Din Vel Index0 Bit 1: Din Vel Index1 Bit 2: Din Vel Index2	User defined
20200620	Integer32	Din_Speed[1]		
20200720	Integer32	Din_Speed[2]		
20200820	Integer32	Din_Speed[3]		
20201420	Integer32	Din_Speed[4]		
20201520	Integer32	Din_Speed[5]		
20201620	Integer32	Din_Speed[6]		
20201720	Integer32	Din_Speed[7]		

For example

The configuration interface of I/O is shown in the following figure:

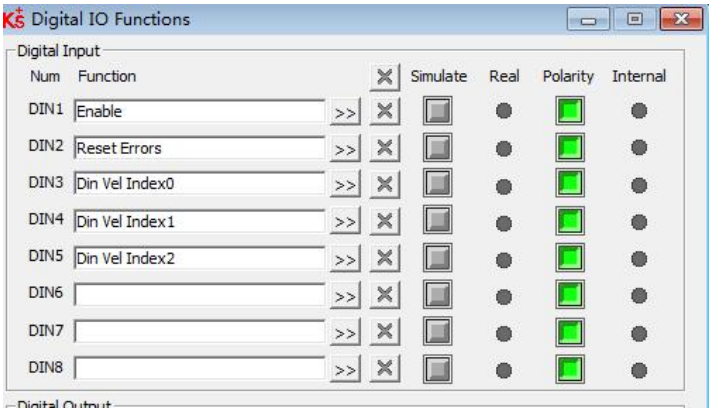
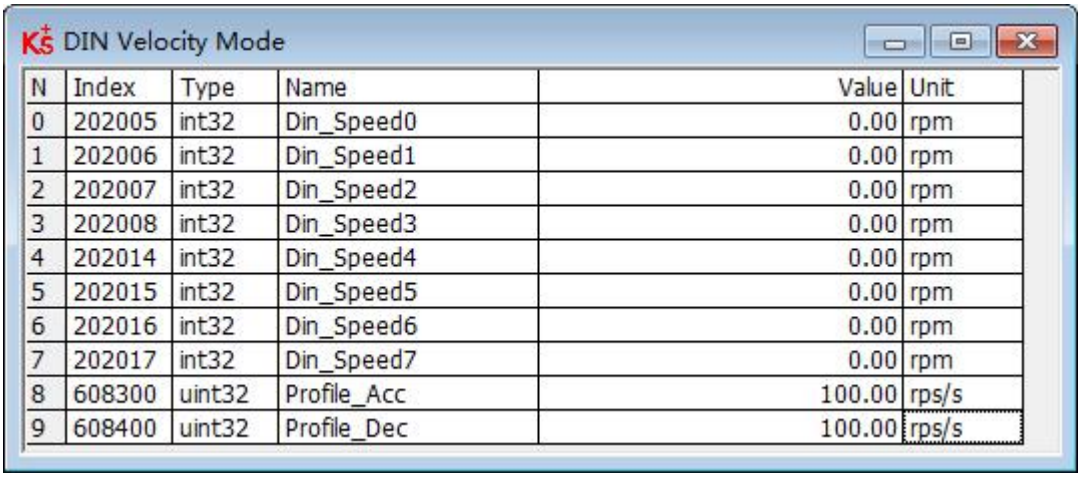


Figure 5-2 DIN Configuration window



N	Index	Type	Name	Value	Unit
0	202005	int32	Din_Speed0	0.00	rpm
1	202006	int32	Din_Speed1	0.00	rpm
2	202007	int32	Din_Speed2	0.00	rpm
3	202008	int32	Din_Speed3	0.00	rpm
4	202014	int32	Din_Speed4	0.00	rpm
5	202015	int32	Din_Speed5	0.00	rpm
6	202016	int32	Din_Speed6	0.00	rpm
7	202017	int32	Din_Speed7	0.00	rpm
8	608300	uint32	Profile_Acc	100.00	rps/s
9	608400	uint32	Profile_Dec	100.00	rps/s

Figure 5-3 DIN Speed Mode window

Table 5–3 DIN Speed Mode related settings

Internal address	Bit	Name	Description	Value
20200E08	Integer8	Din_Mode1	-3	
20200732	Integer32	Din_Speed[2]	500	rpm

When Din speed index 0 = 0, Din speed index 1 = 1, Din speed index 2 = 0, and the DIN1 input signal is valid, the drive will operate in -3 mode at 500rpm.

5.2 Position mode (1)

In the position mode, the driver control motor can carry out absolute position positioning and relative position positioning. The speed and position instructions are controlled by the target position, ladder speed and position table inside the driver.

Table 5-4 Position mode parameter description

Internal address	Bit	Name	Description	Value
60600008	Integer8	Operation_Mode	Set the working mode to absolute/relative position mode	1
607A0020	Integer32	Target_Position	Target absolute / relative position	User defined
60810020	Unsigned32	Profile_Speed	Profile speed for positioning	User defined
60400010	Unsigned16	Control word	0x2F->0x3F: Activate the absolute position command 0x103F: Immediately execute absolute positioning instructions according to target position changes, used when the operation mode is 1. 0x4F->0x5F: Activate relative position command	0x2F->0x3F or 103For 0x4F->0x5F

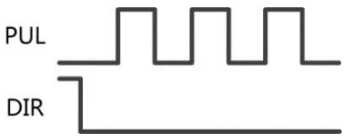
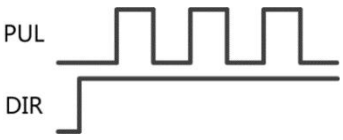
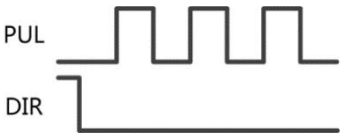
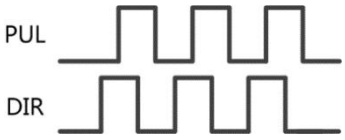
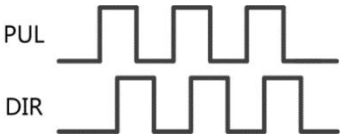
5.3 Pulse mode (-4)

In pulse mode, the target speed command is determined by the external pulse frequency and the electronic gear ratio.

Table 5-5 Pulse mode parameter description

Internal address	Bit	Name	Description	Value
60600008	Integer8	Operation_Mode	Setting operation mode	-4
25080110	Integer16	Gear_Factor[0]	Gear_ratio=Gear_Factor/Gear_Divider	User define
25080210	Unsigned16	Gear_Divider[0]		
60400010	Unsigned16	Controlword	Enable drive	0x2F:
25080308	Unsigned 8	PD_CW	Pulse train mode 0: CW / CCW 1: Pulse / direction 2: A / B (incremental encoder)	0, 1, 2
25080610	Unsigned16	PD_Filter	Master_encoder pulse input filter	User define
25080810	Unsigned16	Frequency_Check	Main encoder port pulse input pulse frequency alarm point set	

Table 5-6 Pulse input supported by the drive

Pulse mode	Forward	Reverse
P/D		
CW/CCW		
A/B		

Note

Forward means positive position counting’ s defaulted to the CCW direction. You can set Invert_Dir(607E.00) to 1 in order to invert the direction of motor shaft rotation.

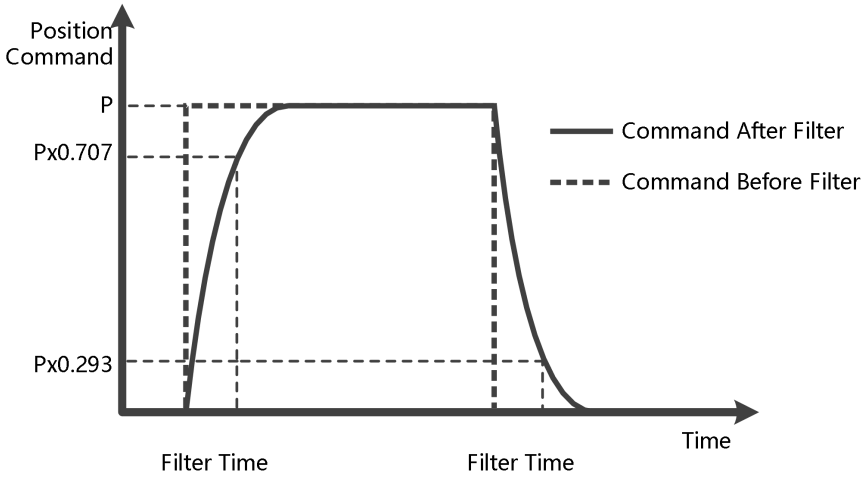


Figure 5-4 Pulse filter

5.4 Homing mode (6)

In some applications, the system requires every movement of the mechanical load to start from the same position, so the user can meet the demand by using the origin mode. In origin mode, the user can define an origin or zero point to ensure that the mechanical load runs from the same starting point every time. The origin mode operation interface can be opened by menu bar -> Drive -> Control Mode -> Origin definition, and the operation interface after opening is shown as follows:

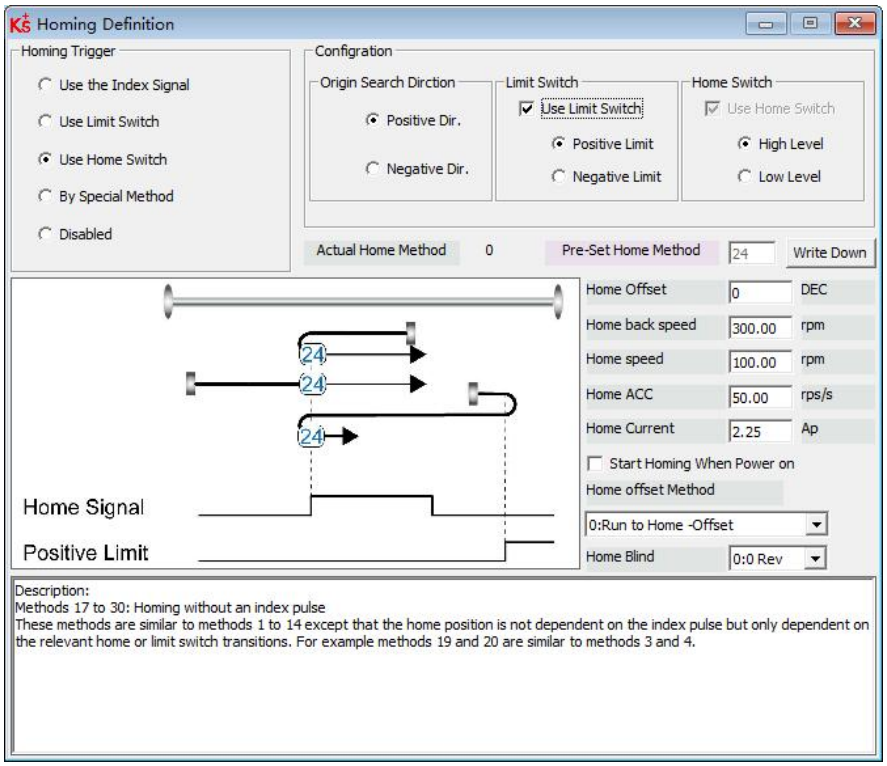


Figure 5-5 Homing setting

Select a home trigger under **Homing Trigger**. The related items appear in the **configuration** area. Select a suitable item according to mechanical design and wiring. The Appropriate homing_method then appears in the **Pre-Set Home Method** box. If **Disabled** is selected under homing trigger, you enter a number directly to the **Pre-Set Home Method** field. Click Write Down to set it to the controller.

The corresponding diagram of the Pre-Set Home method appears in the middle area.

Table 5-7 Homing mode

Internal address	Name	Bit	Value	Description
607C0020	Home_Offset	Integer32	User defined	Zero position offset to the home position
60980008	Homing_Method	Integer 8	User defined	Chose the homing method
60990220	Homing_Speed_Zero	Unsigned32	User defined	Velocity for finding home position and zero position
60990308	Homing_Power_On	Unsigned 8	0，1	0：Default, turn off the Homing_Power_On 1： Start homing after power on or reboot and first controller enable 2：Automatically save the origin position of the multi-turn absolute encoder motor
609A0020	Homing_Accelaration	Unsigned32	User defined	Profile deceleration and acceleration during homing
60990120	Homing_Speed_Switch	Unsigned32	User defined	Velocity for searching position limit switch / home switch signal
60990410	Homing_Current	Integer8	User defined	Max. current during homing
60990508	Home_Offset_Mode	Unsigned 8	0，1	0: Go to the homing offset point. The actual position will be 0. 1: Go to the home trigger point. The actual position will be -homing offset
60990608	Home_N_Blind	Unsigned 8	0，1	Home blind window
60600008	Operation_Mode	Integer8	6	Operation mode of drive
60400010	Controlword	Unsigned16	0x0F->0x1F	Enable drive

Note

Homing_Power_On=1 causes the motor to start rotating as soon as the controller is enabled after power on or reboot. Consider all safety issues before using.

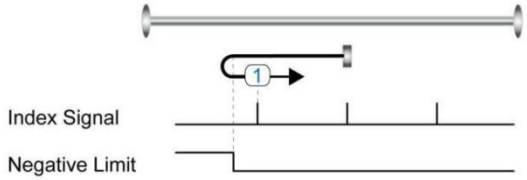
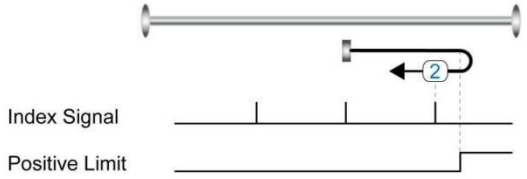
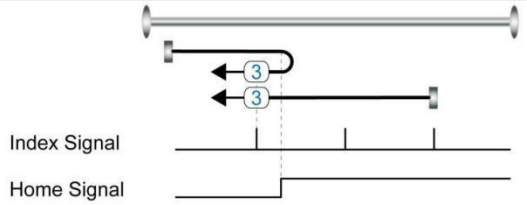
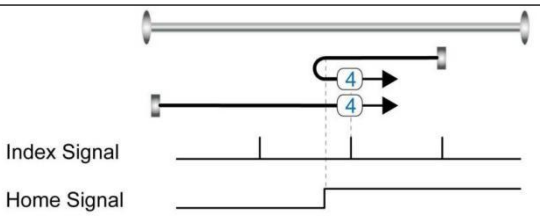
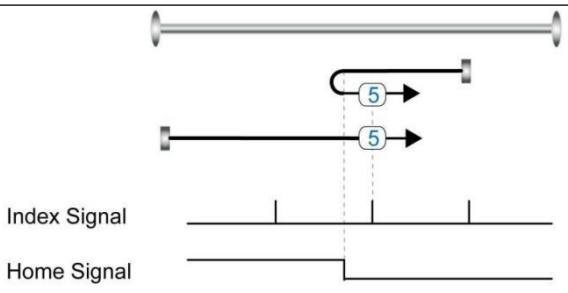
Home_N_Blind:

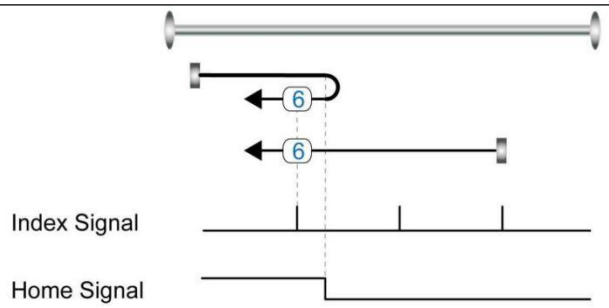
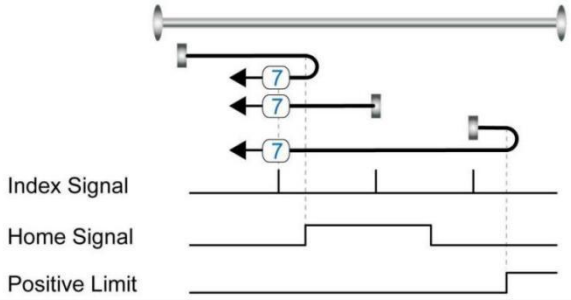
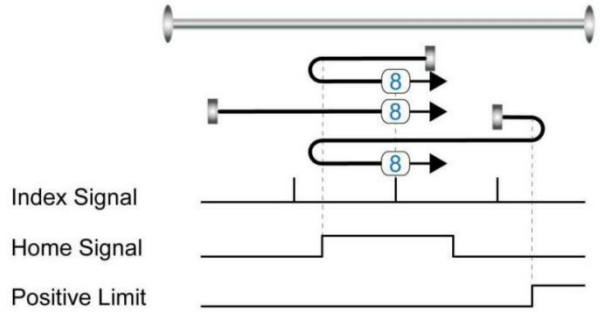
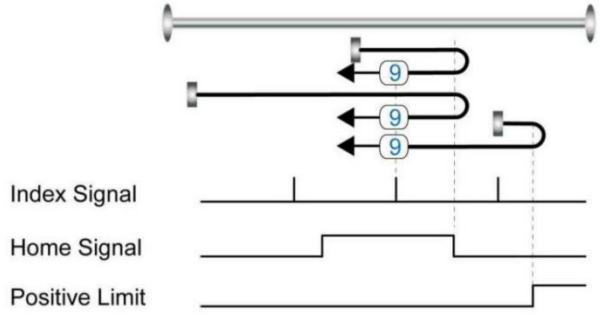
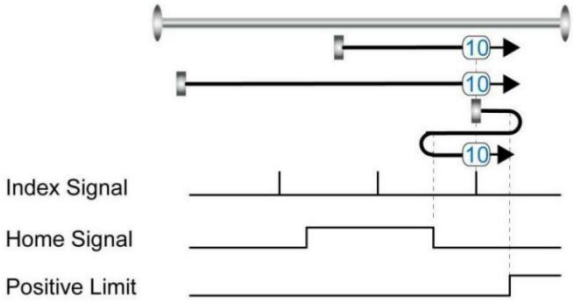
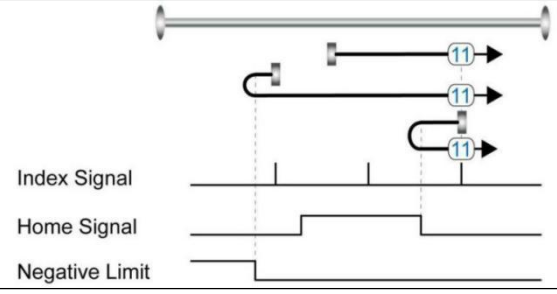
If the homing_method needs home signal (position limit / home switch) and index signal, Home_N_Blind function can avoid the homing result being different with the same mechanics, when the Index signal is very close to the home signal. By setting to 1 before homing, the controller detects a suitable blind window for homing automatically. It can be used to assure that homing results are always the same.

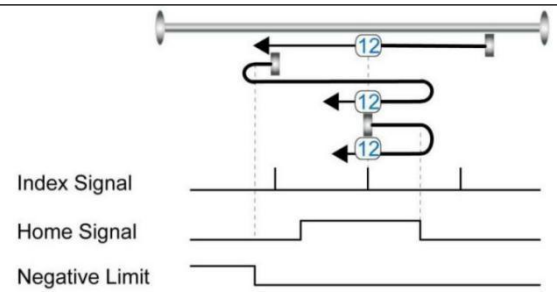
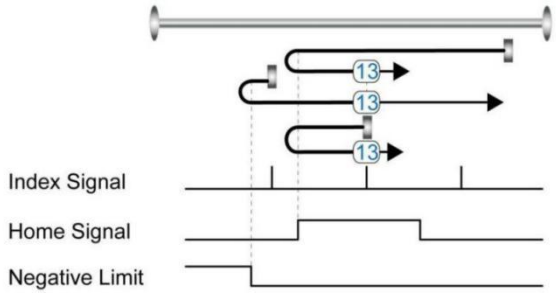
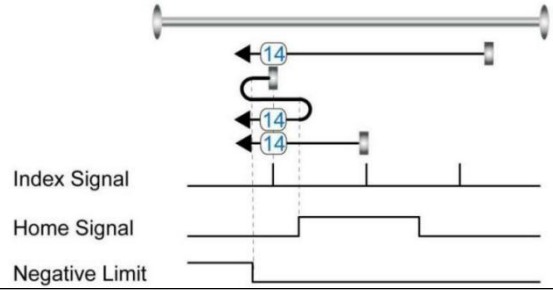
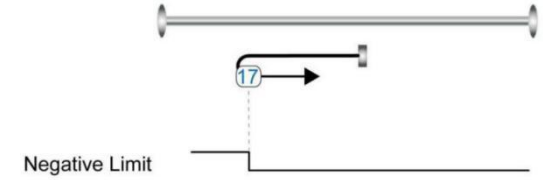
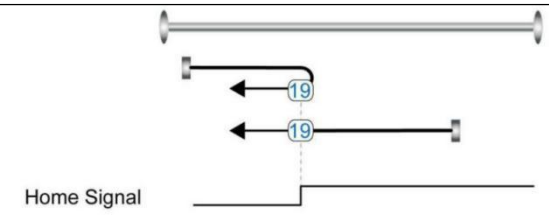
During homing, the index signal inside this blind window is ignored after the home signal is found. Home_N_Blind (0:0rev;1:0.25rev;2:0.5rev) is defaulted to 0. If it's set to 1, it’ s changed to 0

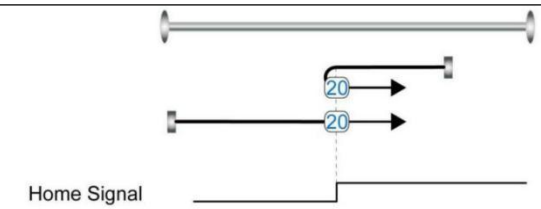
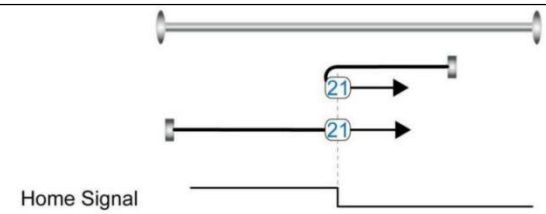
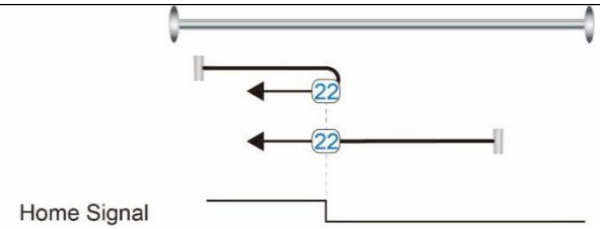
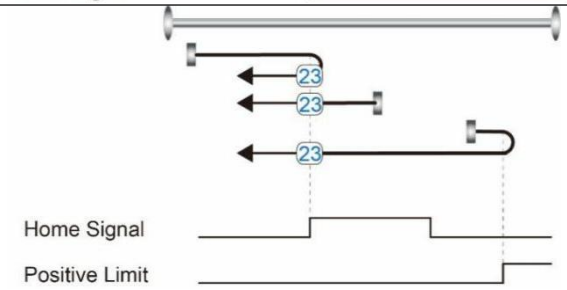
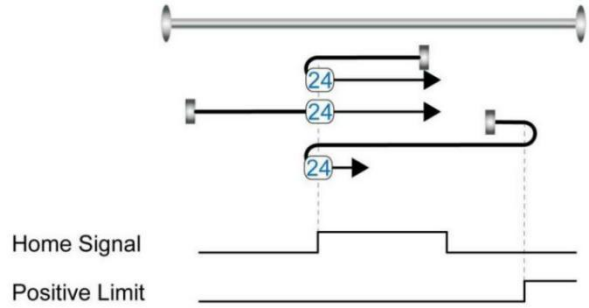
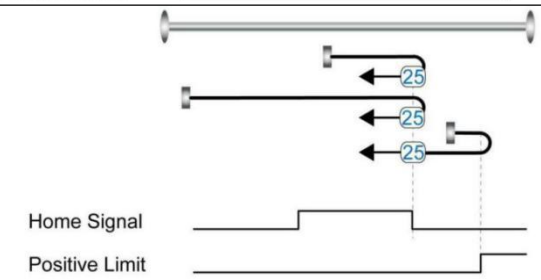
or 2 after homing depending on the index signal position relative to the homing signal.This parameter needs to be saved. If the mechanical assembly is changed or the motor has been replaced, just set it to 1 again for initial homing.

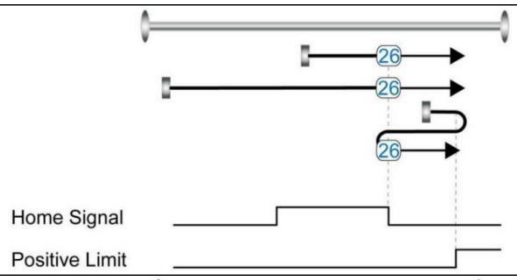
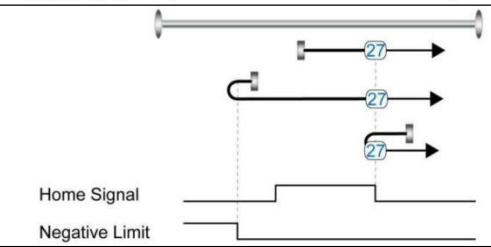
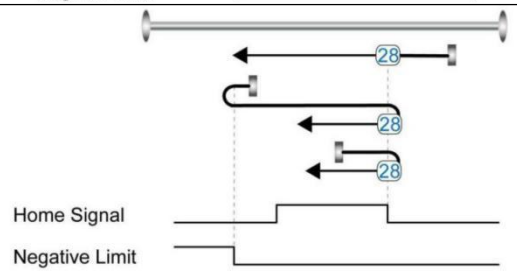
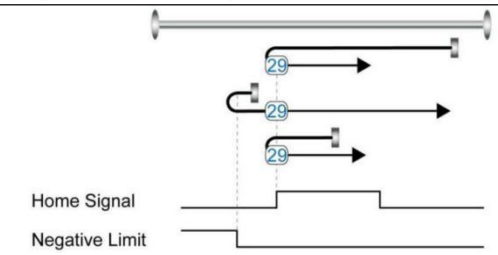
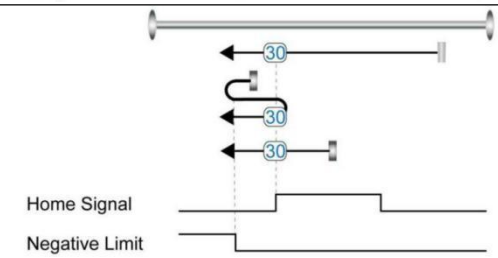
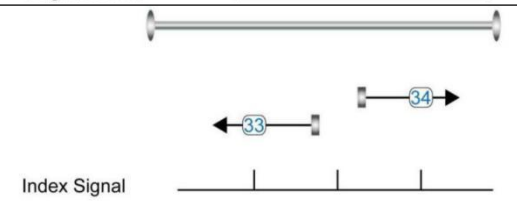
Table 5-8 Homing mode introduction

Homing_ method	Description	Schematic
1	Homing with negative position limit switch and index pulse	
2	Homing with positive position limit switch and index pulse	
3	Homing with home switch and index pulse	
4	Homing with home switch and index pulse	
5	Homing with home switch and index pulse	

6	Homing with home switch and index pulse	
7	Homing with positive position limit switch, home switch and index pulse	
8	Homing with positive position limit switch, home switch and index pulse	
9	Homing with positive position limit switch, home switch and index pulse	
10	Homing with positive position limit switch, home switch and index pulse	
11	Homing with negative position limit switch, home switch and index pulse	

12	Homing with negative position limit switch, home switch and index pulse	
13	Homing with negative position limit switch, home switch and index pulse	
14	Homing with negative position limit switch, home switch and index pulse	
17	Homing with negative position limit switch	
18	Homing with positive position limit switch	
19	Homing with home switch	

20	Homing with home switch	
21	Homing with home switch	
22	Homing with home switch	
23	Homing with positive position limit switch and home switch	
24	Homing with positive position limit switch and home switch	
25	Homing with positive position limit switch and home switch	

26	Homing with positive position limit switch and home switch	
27	Homing with negative position limit switch and home switch	
28	Homing with negative position limit switch and home switch	
29	Homing with negative position limit switch and home switch	
30	Homing with negative position limit switch and home switch	
33, 34	Homing with index pulse	
35	Homing to actual position	
-17, -18	Homing via mechanical limit	

Chapter 6 Performance regulation

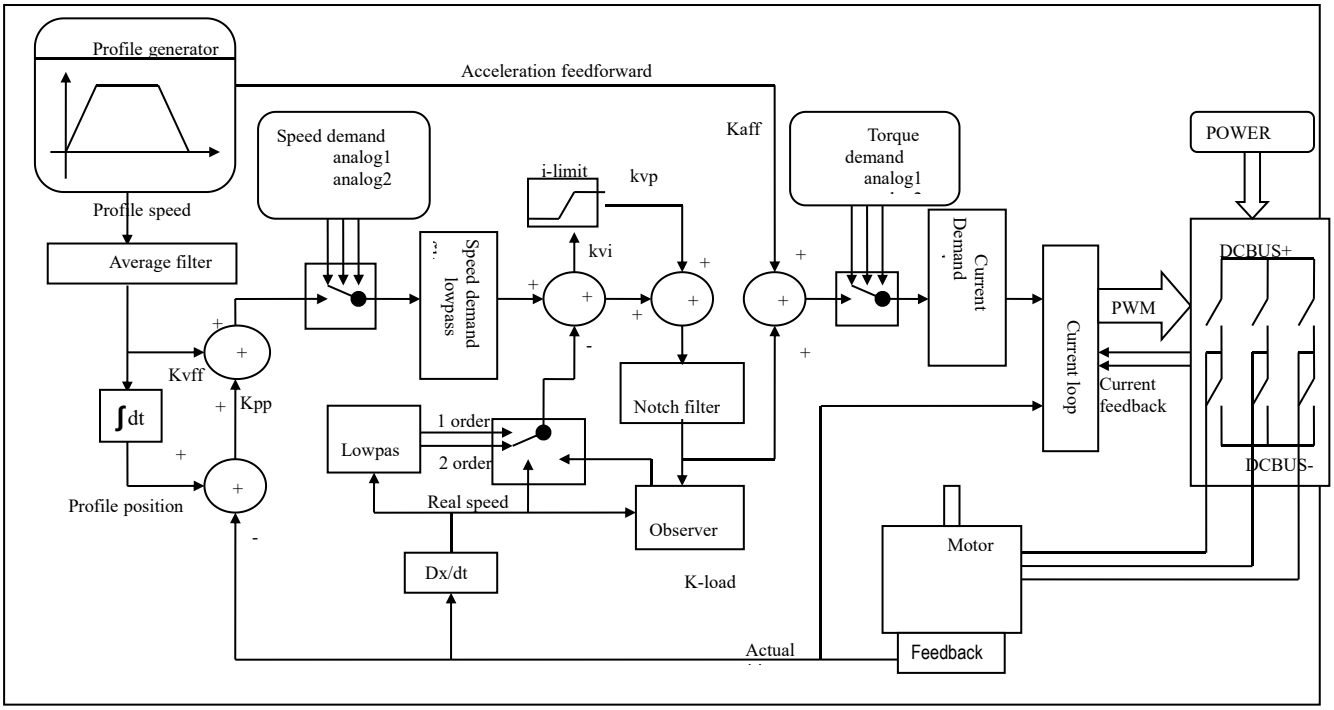


Figure 6–1 Servo system control block diagram

Figure 6-1 shows the servo system control block diagram. It can be seen from the figure that the servo system generally includes three control loops: current loop, velocity loop and position loop.

For the servo system, good control loop parameters can improve the performance of the servo and better meet the field process requirements. Therefore, it is necessary to adjust good control loop parameters.

The parameters of velocity loop and position loop should be adjusted during debugging. The velocity loop parameter is related to the load inertia of the whole mechanical system converted to the motor shaft. The position loop is the outermost control loop of the servo system and is related to the motor action mode, that is, the field application. The current loop is the innermost control loop in the servo system, and the current loop parameters are related to the motor parameters. After the motor is correctly configured, the system defaults the current loop parameters to the best parameters of the configured motor, so there is no need to adjust again.

- kaff：Position loop acceleration feedforward
- kvff：Position loop velocity feedforward
- kvp：Velocity loop proportional gain
- kvi：Velocity loop integration gain
- kpp：Position loop proportional gain

6.1 Tuning of velocity loop

Table 6-1 Velocity loop parameter

Internal address	Name	Description	Default	Range
60F90110	Kvp[0]	Proportional velocity loop gain Can be displayed in Hz in the PC tool can if the inertia ratio is right.	/	1~32767
60F90210	Kvi[0]	Integral velocity loop gain	/	0-1023
60F90710	Kvi/32	Integral velocity loop gain of in a smaller unit of measure	/	0-32767
60F90508	Speed_Fb_N	Used to set Velocity feedback filter bandwidth Filter bandwidth=100+Speed_Fb_N*20	7	0~45
60F90608	Speed_Mode	Used to set the velocity feedback mode 0: 2nd order FB LPF 1: Directly feedback the original velocity 2: Velocity feedback after velocity observer 4: Velocity feedback after 1st order LPF 10:Velocity feedback after 2nd order LPF and the velocity command is filtered by a 1st order LPF. Both filters have the same bandwidth. 11: The velocity command is filtered by a 1st order LPF 12: Velocity feedback after velocity observer, the velocity command is filtered by a 1st order LPF 14: Velocity feedback after 1st order LPF and the velocity command is filtered by a 1st order LPF. Both filters have the same bandwidth	1	/
60F91508	Output_Filter_N	A 1st order lowpass filter in the forward path of the velocity loop	1	1-127
60F90820	Kvi_Sum_Limit	Integral output limit of the velocity loop	/	0-2^15

Speed loop setting steps are as follows:

Step 1: Determine the upper limit of the speed loop bandwidth

The bandwidth of the speed loop limits the bandwidth of the position loop, so it is important to adjust the bandwidth of the speed loop.

The upper limit of the bandwidth of the speed loop can be determined by several aspects:

- Feel motor oscillations and noises through your fingers and ears. It's actually a rule of thumb, but it's very effective. Users can choose to increase or decrease the speed loop bandwidth by listening and touching the machine.
- Another way is to observe the oscilloscope, the user generates a step curve of the speed control, and samples the actual speed and electrical flow of the line. By comparing the sample graphs under different speed loop bandwidths, we can find the optimal curve - the speed curve follows the instruction quickly and does not oscillate.

Step 2: Speed feedback filter adjustment

Feedback filters can reduce noise from the feedback path, for example, reducing encoder resolution noise.

The velocity feedback filter can be configured as 1st and 2nd order via the Speed_Mode for different applications.

The 1st order filter reduces noise to a lesser extent, but its also results in less phase shifting so that velocity loop gain can be set higher. The 2nd order filter reduces noise to a greater extent, but its also results in more phase shifting so that velocity loop gain can be limited.

Normally, if the machine is stiff and light, we can use the 1st feedback filter or disable the feedback filter. If the machine is soft and heavy, we can use the 2nd order filter.

If there′ s too much motor noise when velocity loop gain is adjusted, velocity loop feedback filter parameter Speed_Fb_N can be reduced accordingly. However, velocity loop feedback filter bandwidth F must be more than twice as large as the velocity loop bandwidth. Otherwise, it may cause oscillation. Velocity loop feedback filter bandwidth F=Speed_Fb_N*20+100 [Hz].

Step 3：Output filter adjustment

The output filter is a 1st order torque filter. It can reduce the velocity control loop to output high frequency torque, which may stimulate overall system resonance.

The user can try to adjust Output_Filter_N from small to large in order to reduce noise.

The filter bandwidth can be calculated using the following formula.

$$\frac{1}{2}\ \frac{\ln\left(1-\frac{1}{Output\_Filter\_N}\right)}{Ts\,\pi},\ Ts=62.5\,us$$

Velocity loop bandwidth calculation

Use the following formula to calculate velocity loop bandwidth:

$$kvp=\frac{1.853358080\,10^{5}\,J\pi^{2}\,Fbw}{\mathrm{I}_{Max}\,kt\,encoder}$$

kt motor torque constant, unit: Nm/Arms*100

J inertia, unit: kg*m^2*10^6

Fbw Velocity loop bandwidth, unit: Hz

Imax max motor current I_max(6510.03) as DEC value

encoder resolution of the encoder

Step 4: Speed loop integral gain regulation

Integral gain is used to eliminate static error. It can boost velocity loop low frequency gain, and increased integral gain can reduce low frequency disturbance response.

Normally, if the machine has considerable friction, integral gain (kvi) should be set to a higher value.

If the entire system needs to respond quickly, integral should be set to a small value or even 0, and the gain switch should be used.

Step 5 : Adjust Kvi_sum_limit

Normally the default value is fine. This parameter should be added if the application system has a big extend force, or should be reduced if the output current is easily saturation and the saturation output current will cause some low frequency oscillation.

6.2 Tuning of position loop

Table 6-2 Position loop parameters list

Internal address	Name	Description	Default	Range
60FB0110	Kpp[0]	Set the response bandwidth of the position loop, unit: 0.01Hz	10	0 ~ 327
60FB0210	K_Velocity_FF	0 indicates no feedforward and 100 indicates 100% feedforward	100	0 ~ 100
60FB0310	K_Acc_FF	This parameter can only be set if the inertia ratio is set correctly. If you do not know the inertia ratio, please directly set the position loop acceleration feedforward (0x60FB03).	/	0-32767
60FB0510	Pos_Filter_N	Smooth acceleration and deceleration processes need to be set in the loose shaft state of the motor	1	1~255
60650020	Max_Following_ Error_16	The maximum allowable error, over the change value will alarm 020.0	10000	/

Step of Position loop tuning is shown below:

Step 1:Position loop proportional gain adjustment

Increasing position loop proportional gain can improve position loop bandwidth, thus reducing positioning time and following error, but setting it too high will cause noise or even oscillation. It must be set according to load conditions. Kpp = 103 * Pc_Loop_BW, Pc_Loop_BW is position loop bandwidth. Position loop bandwidth cannot exceed velocity loop bandwidth. Recommended velocity loop bandwidth: Pc_Loop_BW<Vc_Loop_BW / 4, Vc_Loop_BW.

Step 2: Position loop velocity feedforward adjustment

Increasing the position loop velocity feedforward can reduce position following error, but can result in increased overshooting. If the position command signal is not smooth, reducing position loop velocity feedforward can reduce motor oscillation.

The velocity feedforward function can be treated as the upper controller (e.g. PLC) have a chance to directly control the velocity in a position operation mode. In fact this function will expend part of the velocity loop response ability, so if the setting can′ t match the position loop proportional gain and the velocity loop bandwidth, the overshot

will happen.

Besides, the velocity which feedforward to the velocity loop may be not smooth, and with some noise signal inside, so big velocity feedforward value will also amplified the noise.

Step 3 : Position loop acceleration feedforward

It is not recommended that the user adjust this parameter. If very high position loop gain is required, acceleration feedforward K_Acc_FF can be adjusted appropriately to improve performance.

The acceleration feedforward function can be treat as the upper controller (e.g. PLC) have a chance to directly control the torque in a position operation mode. in fact this function will expend part of the current loop response ability, so if the setting can′ t match the position loop proportional gain and the velocity loop bandwidth, the overshot will happen.

Besides, the acceleration which feedforward to the current loop can be not smooth, and with some noise signal inside, so big acceleration feedforward value will also amplified the noise.

Acceleration feedforward can be calculated with the following formula:

ACC_%=6746518/ K_Acc_FF/EASY_KLOAD*100

ACC_%: the percentage which will be used for acceleration feedforward.

K_Acc_FF(60FB.03): the final internal factor for calculating feedforward.

EASY_KLOAD(3040.07): the load factor which is calculated from auto-tuning or the right inertia ratio input.

Note

The smaller the K_Acc_FF, the stronger the acceleration feedforward.

Step 4 : Smoothing filter

The smoothing filter is a moving average filter. It filters the velocity command coming from the velocity generator and makes the velocity and position commands more smooth. As a consequence, the velocity command will be delayed in the controller. So for some applications likeCNC, it′ s better not to use this filter and to accomplish smoothing with the CNC controller.

The smoothing filter can reduce machine impact by smoothing the command. The Pos_Filter_N parameter define the time constant of this filter in ms. Normally, if the machine system oscillates when it starts and stops, a larger Pos_Filter_N is suggested.

Step5 : Notch filter

The notch filter can suppress resonance by reducing gain around the resonant frequency.

Antiresonant frequency=Notch_N*10+100

Setting Notch_On to 1 turns on the notch filter. If the resonant frequency is unknown, the user can set the maximum value of the d2.14 current command small, so that the amplitude of system oscillation lies within an

acceptable range, and then try to adjust Notch_N and observe whether the resonance disappears.

Resonant frequency can be measured roughly according to the Iq curve when resonance occurs on the software oscilloscope.

Table 6-3 Notch filter parameters

Internal address	Name	Description	Default	Range
60F90308	Notch_N	Used to set the frequency of the internal notch filter to eliminate mechanical resonance generated when the motor drives the machine. The formula is F=Notch_N*10+100. For example, if mechanical resonance frequency F=500 Hz, the parameter setting should be 40.	45	0~90
60F90408	Notch_On	Used to turn on or turn off the notch filter. 0：Turn on the notch filter 1：Turn off the notch filter	0	0~1

6.3 Factors which influence tuning results

The control command is created by the upper controller (e.g. PLC):

- The control command should be smooth as much as possible, and must be correct. For example, the control command should not create the acceleration commands (inside the position commands) that the motor cannot provide.
- The control command should follow the bandwidth limit of the control loop.

The machine design:

- In the actual application, performance is normally limited by the machine. Gaps in the gears, soft connection in the belts, friction in the rail, resonance in the system － all of these can influence final control performance.

  Control performance affects the machine′ s final performance, as well as precision, responsiveness and stability. However, final machine performance is not only determined by control performance.

Chapter 7 Alarms and troubleshooting

When driver generate an alarm, red light, ERR, will shine.

If you need more detailed information about errors and error history, please connect the controller to the PC via RS232.

Table 7-1 Error status word 1 alarm code

Alarm	Code	Name	Reason	Troubleshooting
000.1		Extended Error	Errors occurs in Error_State2	Open the menu bar of the upper computer software "Drive" -> "Fault Display" to view the alarm information of the error status word 2. For details about the alarm content and solution, see Table 7-2
000.2	7380	Encoder ABZ signal incorrect（suitable for incremental encoder motor）	Encoder ABZ wiring is wrong or disconnected	1. Check whether the motor model (6410.01) is set correctly. 2. The integrated servo internal encoder is disconnected or connected incorrectly.
		Encoder communication incorrect (suitable for magnetoelectric encoder motor)	The encoder wiring is incorrect or disconnected.	
000.4	7381	Encoder UVW signal incorrect（suitable for incremental encoder motor）	Encoder UVW wiring is wrong or disconnected	1. Check that the motor model (6410.01) in the drive is set correctly. The integrated servo internal encoder is disconnected or incorrectly connected.
		Encoder internal error (suitable for communication encoder motors)	Encoder internal is incorrect or encoder is broken	
000.8	7306	Encoder count wrong （suitable for incremental encoder motor）	Encoder is interfered	1. Check whether the PE end of the X5 is properly connected to the motor shell 2. Check whether the integrated servo PE is properly grounded 3.Use a separate power supply to power the drive
		Encoder CRC (suitable for magnetoelectric encoder motor)		
001.0	4210	Controller temperature	The temperature of controller’s power module has reached the alarm value	1. Add fans to improve the heat dissipation environment of the electric cabinet Appropriately increase the installation distance of the product
002.0	3210	Overvoltage	Supply power voltage exceeds the allowable input voltage range	Check if supply power is higher than standard output voltage Check to see if supply power voltage is unstable
			In case of emergency stop,	Connect suitable braking resistor

			there is no external braking resistor or braking.	Open software "Driver"->“Panel menu”->“（F005）controller setting” Correctly set "brake resistor value" an "brake resistor power"
			Brake resistor is not configured	Change Connect suitable braking resistor Open software "Driver"->“Panel menu”->“（F005）controller setting” Correctly set "brake resistor value" an "brake resistor power"
004.0	3220	Undervoltage	The power voltage input is lower than thelow voltage protection alarm value.	. Check whether the power output meets requirements If the impedance of the cable is high, the line loss is high. For details, see Chapter 3.5 Power Cable Specifications Replace the power supply with a higher power output
008.0	2320	Driver output short circuit	The UVW and PE outputs of the driver have short circuits	Check whether the power cable of the motor is correctly connected 2.The drive is damaged. Replace the drive
010.0	7110	Driver brake resistor is abnormal	Not configure correct brake resistor parameters	Open software "Driver"->“Panel menu”->“（F005）controller setting” Correctly set "brake resistor value" an "brake resistor power"
020.0	8611	Following error	Stiffness of control loop is too small	1.Open software "Driver""->control loop""->velocity loop"and"position loop" 2.Increase "kpp[0]""kvp[0]"
			The controller and motor together can’t match the requirement of the application	Change motor and driver with bigger power
			The maximum motor speed limit is too small	Open the menu bar of the upper computer software "Drive" -> "Control Panel" -> "Control Ring Settings" to check the setting value of "Maximum Speed Limit rpm"
			Max_Following_Error is too small	1.Open software "Driver""control loop""velocity loop""position loop" 2.Increase "max_following_error" (Ensure control loop parameters is fine, user can change this parameter)
			The target current limit is too small	Open the menu bar of the upper computer software "Drive" -> "Basic Operation" to check the setting value of "Target Current Limit"
040.0	5122	Low logic voltage	Logic voltage is less than 18V，power supply voltage is pulled down	1.Check if power supply output power can meet with requirements 2.Change power supply with bigger power
080.0	2350	Motor or controller IIt	The brake is not released when the motor shaft is rotating (only for brake motor)	Integrated servo internal cable break or drive brake circuit fault
			Machine equipment stuck or excessive friction	1. Disable or power off the drive 2. Drag the load by hand to move back and forth on the running stroke of the

				motor to ensure that there is no stuck or excessive friction on the mechanical structure 3. Apply lubricant
			Motor UVW phase sequence is incorrect	Switch motor U phase and V phase connection
100.0	8A80	Over input frequency	External input pulse frequency is too high	1.Reduce external pulse input frequency 2.When ensure safely use motor, increase "Frequency_Check" （Open“Driver”->“Control modes”->“Pulse mode”->“Frequency_Check”），max 500
200.0	4310	Motor temperature	The motor temperature exceeds the specified value	1. Reduce the ambient temperature and improve the cooling condition Reduce motor acceleration and deceleration Load reduction
400.0	7122	Motor excitation（suitable for incremental encoder）	Motor UVW phase sequence is wrong	Exchange motor wiring of phase U and phase V
			Encoder is not connected	The integrated servo internal encoder connection is disconnected
		Encoder information（suitable for magnetoelectric encoder）	Communication is incorrect when the encoder is initialized	Check the encoder wiring and restart the drive
			The encoder type is wrong, e.g. an unknown encoder is connected	
			The data stored in the encoder is wrong	
			The controller can’t support the current encoder type	

Table 7-2 error status word 2 alarm code

Alarm	Code	Name	Reason	Trouble shooting
000.1	0x5210	Current sensor	Current sensor signal offset or ripple too big	1, the integrated servo PE end is well grounded 2, eliminate the surrounding magnetic field interference 3, the current sensor circuit is damaged, please contact the supplier
000.2	0x6010	Watchdog	Software watchdog exception	Initialize control parameters - Store control parameters 2, please contact the vendor, or try to update the firmware
000.4	0x6011	Wrong interrupt	Invalid interrupt exception	1, Initialize control parameters - Store control parameters 2, please contact the vendor, or try to update the firmware
000.8	0x7400	MCU ID	Wrong MCU type detected	Please contact the supplier
001.0	0x6320	Motor configuration	Motor type is not auto-recognized, no motor	Reset motor parameters and save, restart the drive

			data in EEPROM / motor never configured	
			he driver supply voltage is too low.	Select the power supply based on the driver specifications
010.0	0x5443	External enable	DIN function “pre_enable” is configured, but the input is inactive when the controller is enabled or should become enabled	Solve according to the reason
020.0	0x5442	Positive limit	Positive position limit (after homing), position limit only causes error when Limit_Function (2010.19) is set to 0	Exclude the condition which causes the limit signal
040.0	0x5441	Negative limit	Positive position limit (after homing), position limit only causes error when Limit_Function (2010.19) is set to 0	
080.0	0x6012	SPI internal	Internal firmware error in SPI handling	Please contact the supplier
200.0	0x8A81	Close loop direction	Different direction between motor and position encoder	Change the encoder counting direction

Chapter 8 List of motor controller parameters

Note

CANopen address is same as 232 communication address:

- Use Index (16 bits address), Subindex (8 bit subaddress) to show register addressing,
- 0x08 means that data length of register store is 1 Byte, 0x10 means that data length of register store is 2 Byte, 0x20 means that data length of register store is 4 Byte,
- R: Read, W: Write, S:Save, M: Map,
- A complete CANopen address format is : 60400010(controlword), Modbus address is 4 bits Hexadecimal number
- A complete Modbus address format is : 3100(controlword)

8.1 Mode and Control (0x6040)

Name	CANopen	Modbus	Command Properties	Data type	Explanation
Control word	60400010	3100	RWM	Unsigned16	0x06: Loose shaft 0x0F: Lock shaft 0x0B: Quick stop, load stop - voltage disconnect 0x2F→3F : Enter the absolute positioning mode 0x4F→5F: Enter relative positioning mode 0x103F:Immediate absolute positioning as target position changes 0x0F-1F: homing 0X86: Error reset

Status word	60410010	3200	RM	Unsigned16	The status byte shows the status of the drive bit0：Ready_on bit1：Switched_on bit2：Operation_enable bit3： Fault bit4：Voltage_enable bit5： Quick_stop bit6：Switchon_disabled bit7：Warning bit8：Internal retention bit9：Remote bit10：Target_reached bit11：Internal limit activation bit12：Pulse response bit13：Fllowing_Error bit14：Commutation_Found bit15：Reference_Found
Operating mode	60600008	3500	RWM	Integer8	1：Positioning mode with position ring 3：Speed mode with position loop 4：Torque mode -3： Speed loop(Immediate velocity mode) -4： Pulse mode 6：Homing mode 7： Motion interpolation based on CANopen
Absolute/ relative position control selection	20200F	0CF0	RWS	Unsigned16	When the "Drive Enable" function is configured to Din and the corresponding Din valid input is 1, the "Control word" (6040.00) is set to this value; 0x2F：Absolute position control 0x4F：Relative position control

8.2 Data measuring

Name	CANopen	Modbus	RWS	Data type	Description
Pos_Actual	60630020	3700	RM	Integer32	
Real current	60780010	3E00	RM	Integer16	
Status of input port	60FD0020	6D00	RM	Unsigned32	bit0: negative limit switch bit1: positive limit switch bit2: home switch bit3: interlock
Real speed	606C0020	3B00	RM	Integer32	rpm

Note

0x606C0020，conversion between engineering unit and internal unit of common objects DEC=[(RPM*512*Encoder_Resolution)/1875]

8.3 Target object（0x607A）

Name	CANopen	Modbus	Command Properties	Data type	Explanation
Velocity position direction control	607E0008	4700	RWS	Unsigned8	Run polarity reversal 0: Counterclockwise is the positive direction 1: Clockwise is the positive direction
Target position	607A0020	4000	RWM	Integer32	The target position in position mode 1, if the control word is set to start motion, transitions to the valid command position inc
Trapezoidal velocity	60810020	4A00	RWM	Unsigned32	Trapezoidal velocity (rpm) in operating mode 1
Target speed	60FF0020	6F00	RWM	Integer32	Target speed in mode 3 and -3,
Max velocity limit	60800010	4900	RW	Unsigned16	Default:5000rpm
Trapezoidal acceleration	60830020	4B00	RWSM	Unsigned32	Default：100rps/s
Trapezoidal deceleration	60840020	4C00	RWSM	Unsigned32	Default：100rps/s
Target torque	60710010	3C00	RW	Integer16	Torque command in torque mode, the percentage of target torque to rated torque
Target current	60F60810	5880	RWM	Integer16	Current command in torque mode
Target current limit	60730010	3D00	RWSM	Unsigned16	Current instruction maximum

Note

Speed address：0x60810020，0x60800020，0x60FF0020

Conversion between engineering unit and internal unit of common objects DEC=[(rpm*512*Encoder_Resolution)/1875]

Acc & Dec address：60830020，60840020，

Conversion between engineering unit and internal unit of common objects DEC=[(rps/s*65536*Encoder_Resolution)/4000000]

Current address：60710010，60730010

Conversion between engineering unit and internal unit of common objects 1Arms=(2048/Ipeak/1.414)DEC Ipeak is driver max current

8.4 Din speed/position（0x2020）

Name	CANopen	Modbus	Command Properties	Data type	Explanation
Multi-stage position control 0	20200120	0C10	RWS	Integer32	
Multi-stage position control 1	20200220	0C20	RWS	Integer32	
Multi-stage position control 2	20200320	0C30	RWS	Integer32	
Multi-stage position control 3	20200420	0C40	RWS	Integer32	
Multi-stage Position Control 4	20201020	0D00	RWS	Integer32	
Multi-stage Position Control 5	20201120	0D10	RWS	Integer32	
Multi-stage position control 6	20201220	0D20	RWS	Integer32	
Multi-stage position control 7	20201320	0D30	RWS	Integer32	
Multi-stage speed control 0	20200520	0C50	RWS	Integer32	
Multi-stage speed control 1	20200620	0C60	RWS	Integer32	
Multi-stage speed control 2	20200720	0C70	RWS	Integer32	
Multi-stage speed control 3	20200820	0C80	RWS	Integer32	
Multi-stage speed control 4	20201420	0D40	RWS	Integer32	
Multi-stage speed control 5	20201520	0D50	RWS	Integer32	
Multi-stage speed control 6	20201620	0D60	RWS	Integer32	
Multi-stage speed control 7	20201720	0D70	RWS	Integer32	

8.5 Performance objects（0x6065）

Name	CANopen	Modbus	Command Properties	Data type	Explanation
Max following error	60650020	3800	RWSM	Unsigned32	Following error value alarm value Default 524288inc
target location window	60670020	3900	RWS	Unsigned32	“Target location reached”target range，Default 327inc
reach position time window	25080916	1990	RW	Unsigned16	Target（location.velocity）reach time window，Determine the reach position signal together with 0x60670020
Reach speed window	60F90A20	63A0	RWS	Integer32	The error window when the actual speed reaches the target speed or trapezoidal speed, which together with 25080916 determines the speed to signal
Zero speed output speed window	20101810	0980	RWS	Unsigned16	Error window when actual speed is 0
Zero speed output time	60F91410	6440	RWS	Unsigned16	Zero-speed output speed window 0x20101810 After reaching the set range, it takes a period of time to output the zero-speed signal, and the time is determined by the zero-speed output time
Positive setting of soft limit	607D0120	4410	RWS	Integer32	Soft limit positive setting
Soft limit negative setting	607D0220	4420	RWS	Integer20	Soft limit negative setting
Limit function definition	20101908	0990	RWS	Unsigned8	Used to set the action after the limit is reached 0: If there is a limit after finding the origin, it will alarm 1: do nothing

8.6 Home control（0x6098）

Name	CANopen	Modbus	Command Properties	Data type	Explanation
Homing mode	60980008	4D00	RWSM	Integer8	Homing function Refer origin control chapter
Origin turning signal speed	60990120	5010	RWSM	Unsigned32	After touching the trigger event, the speed of homing process (rpm)
Origin signal speed	60990220	5020	RWSM		Speed when starting to find the origin
Homing acceleration	609A0020	5200	RWS	Unsigned32	Acceleration when homing process unit: rps/s
Origin offset	607C0020	4100	RWSM	Integer32	Offset value after homing unit: inc
Origin Offset Mode	60990508	5050	RWS	Unsigned8	Origin Offset Mode Control 0: run to the origin offset position, the actual position is displayed as 0 1: Run to the event trigger point, after the end the actual position will become: - Origin offset

8.7 Velocity loop（0x60F9）

Name	CANopen	Modbus	RWS	Data type	Description
Name	CANopen	Modbus	Command Properties	Data type	Explanation
Kvp	60F90110	6310	RW	Unsigned16	The larger the value, the stronger the gain, but it may cause the motor howling
Kvi	60F90210	6320	RW	Unsigned16	The larger the value, the stronger the gain, but it may cause the motor howling
Kvi/32	60F90710	6370	RWSL	Unsigned16	Kvi's 1/32

8.8 Position loop（0x60FB）

Name	CANopen	modbus	RWS	Data type	Description
kpp 0	60FB 0110	6810	RWS	Unsigned16	Scale value of the position loop
kvff	60FB0210	6820	RWS	Unsigned16	Position loop speed feedforward
kaff	60FB0310	6830	RWS	Unsigned16	Acceleration feedforward for position loop
Smooth filtering	60FB0510	6850	RWS	Unsigned16	The value is modified when the function is disabled

8.9 Input & Output（0x2010）

Name	CANopen	Modbus	Command Properties	Data type	Explanation
Digital input 1	20100310	0830	RWS	Unsigned16	Refer to the function definition below
Digital input 2	20100410	0840	RWS	Unsigned16	
Digital input 3	20100510	0850	RWS	Unsigned16	
Digital input 4	20100610	0860	RWS	Unsigned16	
Digital output 1	20100F10	08F0	RWS	Unsigned16	
Digital output 2	20101010	0900	RWS	Unsigned16	
Input port status	20100A10	08A0	RM	Unsigned16	bit0：Din1 bit1：Din2 bit2：Din3 bit3：Din4
Output port status	20101410	0940	RM	Unsigned16	bit0：Dout1 bit1：Dout2
Change the input signal polarity definition	20100110	0810	RWS	Unsigned16	0：Turn off; 1：Turn on bit0：Din1 bit1：Din2 bit2：Din3 bit3：Din4 bit4：Din5 bit5：Din6 bit6：Din7 bit7：Din8 Default 0xFF
Polarity definition of output port	20100D10	08D0	RWSM	Unsigned16	Polarity definition of output port
Input port signal simulation	20100210	0820	RW	Unsigned16	bit0：Din1 bit1：Din2 bit2：Din3 bit3：Din4 bit4：Din5 bit5：Din6 bit6：Din7 bit7：Din8
Output port signal simulation	20100E10	08E0	RWM	Unsigned16	bit0：Dout1 bit1：Dout2 bit2：Dout3 bit3：Dout4 bit4：Dout5

Note

Definition of digital input function (hexadecimal)	Output port definition (hexadecimal)
0001：Drive enable 0002：Drive error reset 0004：Drive working mode control 0008：Speed loop kvp control 0010：Positive limit 0020： Negative limit 0040：Origin signal 0080：Speed command reverse 0100：Din speed index 0 0200：Din speed index 1 0400：Din position index 0 0800： Din position index 1 1000：Emergency stop 2000: Start to find the origin 4000： Command activated 8001: Din speed index 2 8002： Din position index 2 8004: Multi-function input signal 0 (for setting multi-stage electronic gear ratio) 8008： Multi-function input signal 1 8010: Multi-function input signal 2 8020:Gain switching input signa 0 8040: Gain switching input signa 1 8080: Maximum current toggle input switch 8100: Motor failure 8200: Pre-enable (The IO port must have an enable signal, otherwise it will alarm, which is used in some occasions where the machine needs to be determined before running the machine) 8400: Quick Capture 1 8800: Quick Capture 2 A001: Clear pulse A002: Pulse	0001：Drive ready 0002：Drive error 0004：Pos reached 0008：Motor zero speed 0010：Motor holding brake 0020：Motor speed reached 0040：Index signal appears 0080：Speed limit 0100：Driver Enabled 0200: Position limit 0400: Origin found 0800: Torque reached limit 1000: Multi-function signal 0 2000: Multifunction Signal 1 4000: Multifunction Signal 2

8.10 Pulse input（0x2508）

Name	CANopen	Modbus	Command Properties	Data type	Explanation
Electronic gear molecule 0	25080110	0x1910	RWSM	Integer16	Electronic gear molecule 0
Electronic gear denominator 0	25080210	0x1920	RWSM	Unsigned16	Electronic gear denominator 0
Pulse mode control	25080310	0x1930	RWSB	Integer16	0: Double pulse mode 1: Pulse direction mode 2: Incremental encoder mode

Number of input pulses before electronic gear	25080410	0x1940	RWM	Integer16	Number of input pulses before electronic gear
Number of input pulses after electronic gear	25080510	0x1950	RW	Integer16	Number of input pulses after electronic gear
Pulse filter parameters	25080610	0x1960	RWS	Unsigned16	Pulse filter parameters
Pulse frequency before gear	25080C10	0x19C0	RM	Integer16	Pulse frequency before gear (pulse/mS)
Pulse frequency after gear	25080D10	0x19D0	RW	Integer16	Pulse frequency after gear (pulse/mS)

8.11 Save（0x2FF0）

Name	Subindex	modbus	RWS	Data type	Description
Storage control parameters	2FF00108	2910	RW	Unsigned8	1: Store all configuration parameters set 10: Initialize all configuration parameters Note: The control loop parameters are stored, excluding motor parameters.
Store motor parameters	2FF00308	2930	RW	Unsigned8	1: Store all set motor parameters

8.12 Error code（0x2601）

Name	CANopen	Modbus	Command Properties	Data type	Explanation
Error state	26010010	1F00	RM	Unsigned16	Real-time alarm error status bit 0: Extended Error bit 1：Encoder ABZ/not connected bit 2：Encoder UVW/ Encoder internal bit 3：Encoder Counting/Encoder CRC bit 4: Driver temperature bit 5：Over voltage bit 6：Under voltage bit 7：Over current bit 8：Chop Resistor bit 9：Position Following bit 10：Low logic voltage bit 11：Motor or Driver IIt bit 12：Over frequency bit 13：Motor temperature bit 14：Motor commutation bit 15：EEPROM data

8.13 Stop

Name	CANopen	Modbus	Command Properties	Data type	Explanation
Quick stop mode	605A0010	3400	RWS	Integer16	Encountered limit switch, emergency stop switch, or control word is 0x000B 0: stop without control 1: stop by using ramp, then switch off 2: stop by using quick stop deceleration, then switch off 5: stop with profile deceleration, stay in quick stop active 6: stop with quick stop deceleration, stay in quick stop active
Shutdown mode	605B0010	3410	RWS	Integer16	Shutdown stop mode (Drive enable cancel) 0: stop without control 1: stop by using ramp, then switch off 2: stop by using quick stop deceleration, then switch off
Disable stop mode	605C0010	3420	RWS	Integer16	0: stop without control 1: stop by using ramp, then switch off 2: stop by using quick stop deceleration, then switch off
Pause mode	605D0010	3430	RWS	Integer16	Control word bit8 is set to 1 The motor is paused and enabled 1: stop by current ramp 2: stop by quick stop deceleration
Error stop mode	605E0010	3440	RWS	Integer16	0: stop without control 1: stop by using ramp, then switch off 2: stop by using quick stop deceleration, then switch off
Trapezoidal deceleration	60840020	4C00	RWSM	Unsigned32	Working Mode Deceleration in modes 1 and 3
Quick stop deceleration	60850020	3300	RWS	Unsigned32	Deceleration for quick stop

Chapter 9 RS232 Communication

MD servo can be connected to the host computer through RS232 interface (X2) for configure parameters and debugging. The specific interface definition and communication protocol are described as follows:

9.1 RS232 wiring definition

Users can buy Kinco OD124RS232-0.5M cable to connect the computer serial port to debug MD series integrated servo

Figure 9-1 OD124RS232-0.5M communication cable

Communication cable can also be made using RS232 terminals and pins shipped with the product. Figure 9-2 shows the pin mapping.

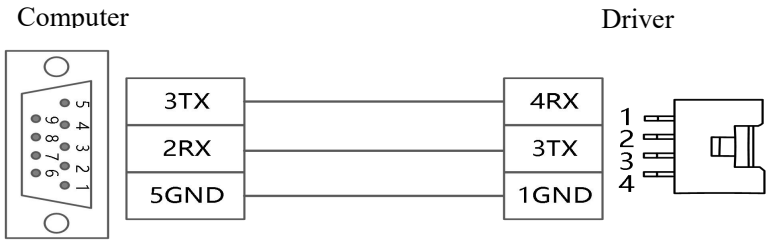


Figure 9-2 Pin mapping between RS232 to DB9 cable

9.2 Transport protocol

RS232 communication follows the master-slave protocol. The host sends data to the drive, and the drive checks whether the data verification code matches the ID number. If so, the drive processes the data and replies. The default communication parameters of the drive are set as follows:

Baud rate: 38400bps
Data bit: 8
Stop bit: 1
no-check

You can modify the communication baud rate of the driver using object dictionary 2FE0.00. After the change is complete, you need to save and restart the initiator for it to take effect.

The drive ID number can be set by dip switch or object device station number (2F80.01). After the change is complete, you need to save and restart the drive for it to take effect. The DIP switch priority is higher than the station number. If the station number is set by using the DIP switch, the station number cannot be changed on the host. To change the station number on the host computer, set all DIP switches of the ID to OFF first.

The transport protocol used by RS232 uses a fixed ten-byte packet format:

Byte 0	Byte 1 …Byte 8	Byte 9
ID	Data	CHKS

CHKS = -SUM(byte 0 …. byte 8)

9.2.1 Point-to-point protocol

One master station and one servo communication, 6510.0B is set to 0.

Master send：

Byte 0	Byte 1 …Byte 8	Byte 9
ID	Data	CHKS

Slave send/Master receive

Byte 0	Byte 1 …Byte 8	Byte 9
ID	Slave send data	CHKS

The driver checks the CHKS of the data frame that matches the driver ID. If the CHKS of the data frame does not match, the driver does not reply to the frame and discards the frame.

9.3 Data protocol

Data protocol refers to the protocol that transmits data from byte 1 to byte 8 in the data frame, with a total of 8 bytes. The MD Series integrated servo RS232 data protocol complies with the CANopen SDO protocol standard and the internal data object structure complies with the CANopen standard. All data objects are specified by a 24-bit data address consisting of a 16-bit index address and an 8-bit subindex.

9.3.1 Download (from host to slave)

Write means that the master sends a command to write a value to the slave. If it is written to a non-existent object, the slave will generate an error.

Master send data format:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Function code	Indexes		Subindexes	Data			

Note

Function code: Specifies the direction and size of data transfer.

23（hex） send 4 bytes of data (bytes 4... 7 contains 32 bits)

2b （hex）send 2 bytes of data (bytes 4... 5 contains 16 bits)

2f（hex） send 1 byte of data (bytes 4 contains 8 bits)

Index: Address of the sent data object, 16 bits

Subindex: The subaddress of the sent data object, 8 bits

Data: The data to be sent, 8, 16, or 32 bits

Slave receive data format:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Function code	Index		Subindex	Reserve			

Note

Function code: Indicates the response of the slave

60（hex）Data sent successfully

80（hex）Error, by byte 4... 7 generation

Index： The address of the sending object, 16 bits,Same as the main station

Subindex： Subaddress of the sending object, 8 bits,Same as the main station

Reserve： Reserved

9.3.2 Upload (from slave to host)

Upload means the master sends a command to read the object value from the slave. The slave generates an error if a non-existent object is requested.

The master sends:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Function code	Index		Subindex	Reserve			

Note

Function code： Define the direction of the data transfer

40（hex）Read data, any length

Index： The address of the sending object, 16 bits

Subindex： Subaddress of the sending object, 8 bits

Reserve： Byte 4... 7 no useRESERVED: Bytes 4...7 not used

Slave receive data format:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Function code	Index		Subindex	Data			

Note

Function code： Display slave response

43（hex） Bytes 4... 7 contains 32 bits

4b（hex） Bytes 4... 5 contains16 bits

4f（hex） Bytes 4contains 8 bits

80（hex） Error, by byte 4... 7 generation

Index： The address of the sending object, 16 bits,Same as the main station

Subindex： Subaddress of the sending object, 8 bits,Same as the main station

Data： Data returned from the slave station

9.4 RS232 telegram example

RS232 telegram example：

Drive ID	Read/write command	Index	Subindex	Data	Check	Note
01	2B	40 60	00	2F 00 00 00	05	Set the control word to 0x2F to enable the drive
01	2F	60 60	00	06 00 00 00	0A	Set the operation mode to 0x06
01	23	7A 60	00	50 C3 00 00	EF	Set the target position to 50000
01	40	41 60	00	00 00 00 00	1E	Read the drive status word

The following are various modes of sending messages, all using station number 1 as an example.

Homing mode（Controlword F to 1F)				
Parameter address	Name	Value	Message(ID=1)	Note
60400010	Control word	F	01 2B 40 60 00 0F 00 00 00 25	Homing_Speed_Switch and Homing_Speed_Zero fault unit is DEC , DEC=[(RPM*512*Encoder resolution)/1875]
60600008	Operating mode	6	01 2F 60 60 00 06 00 00 00 0A	
60980008	Homing mode	33	01 2F 98 60 00 21 00 00 00 B7	
60990120	Home turning signal speed	200RPM	01 23 99 60 01 55 55 08 00 30	
60990220	Homing signal velocity	150RPM	01 23 99 60 02 00 40 06 00 9B	
60400010	Controlword	1F	01 2B 40 60 00 1F 00 00 00 15	
01 40 41 60 00 00 00 00 00 1E Read the status word, C037 indicates origin found				

Position mode（The absolute positioning of the control word is first 2F, then 3F, and the relative positioning is first 4F, then 5F. 103F is updated immediately)				
Parameter address	Name	Value	Message(ID=1)	Note
60400010	Control word	F	01 2B 40 60 00 0F 00 00 00 25	DEC=[(RPM*512* encoder resolution)/1875]
60600008	Operating mode	1	01 2F 60 60 00 01 00 00 00 0F	
607A0020	Target position	50000inc	01 23 7A 60 00 50 C3 00 00 EF	
60810020	Trapezoidal speed	200RPM	01 23 81 60 00 55 55 08 00 49	
60830020	Trapezoidal acceleration	100.00rps/s	Use default value	DEC=[(RPS/S* 65536*Encoder resolution)/1000/4000]
60840020	Trapezoidal deceleration	100.00rps/s	Use default value	
60400010	Controlword	2F	01 2B 40 60 00 2F 00 00 00 05	
		3F (absolute positioning)	01 2B 40 60 00 3F 00 00 00 F5	
		4F	01 2B 40 60 00 4F 00 00 00 E5	
		5F (relative positioning)	01 2B 40 60 00 5F 00 00 00 D5	
01 40 41 60 00 00 00 00 00 1E		Read status word,D437 indicates position reach		

Speed mode				
Parameter address	Name	Value	Message(ID=1)	Note
60600008	Operation mode	3	01 2F 60 60 00 03 00 00 00 0D	Default unit of Target speed DEC，DEC=[(RPM*512*Encoder Resolution)/1875] Default unit of Profile acc/dec DEC，DEC=[(RPS/S*65536* Encoder Resolution)/1000/ 4000]
60FF0020	Target speed	150RPM	01 23 FF 60 00 00 40 06 00 37	
60400010	Control word	F	01 2B 40 60 00 0F 00 00 00 25	
60830020	Trapezoidal acceleration	100.00 rps/s	Use default value	
60840020	Trapezoidal deceleration	100.00 rps/s	Use default value	

Note

Under communication mode, data are transmitted in HEX.

Chapter 10 RS485 communication

10.1 RS485 wiring

The RS485 port of the MD servo system supports RS485-2, but does not support RS422 communication function. The communication port can be used to modify the internal parameters of the servo and monitor the servo state.

Figure 10-1 485 Communication connection

10.2 RS485 communication parameters

Internal address	Name	Meaning	Default
100B0010	Device station number	Driver station number	1
2FE20010	RS485 baud rate	Set the baud rate of the RS485 port (Modbus address: 0X2600) Set value Baud rate 1080————9600 540————19200 270————38400 90————115200 Note: Need to save and restart.	540
65100C08	RS485 Communication protocol selection	0: Using MODBUS protocol 1: Using RS232 communication protocol Note: It needs to be set to 0, save and restart.	1
65100E10	RS485 mode	Data bit =8, stop bit =1, no parity check	Fixed value

10.3 MODBUS RTU

MD servo supports the MODBUS RTU communication protocol, and its internal object is a discontinuous 16-bit data register (mapped to 4X when it is read and written by the upper computer). The format of the message is as follows:

Target station	Function code	Data	CRC check code
1 Byte	1 Byte	N Byte	2 Byte

10.4 Function code of Modbus

- Function code 0x03: Read data register

Request format:

Target station	Function code	Modbus address		Read byte		CRC
		High Byte	Low byte	High Byte	Low byte	
1 Byte	03	1 Byte	1 Byte	1 Byte	1 Byte	2 Byte

Correct response：

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

Note

If there is error such as non-exist address,then it will return function code 0x81.

- Function code 0x06：write single data register

Request format:

Station No.	Function code	Modbus address		Writing value		CRC
		High byte	Low byte	High byte	Low byte	
1 byte	06	1 byte	1 byte	1 byte	1 byte	2 bytes

Response format: If the setting is successful, the original text will be returned.

Note

If there is error such as address over range,non-exist address and the address is read only,then it will return function code 0x86.

- Function code 0x10: Write multi-holding register

Request format:

Target station	Function code	Modbus address	Data length（word）		Number of bytes of data written（byte）	Low level data		High level data		CRC
			High Byte	Low byte		High Byte	Low byte	High Byte	Low byte	
1 Byte	10	2Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	2Byte

Correct response：

Target station	Function code	Modbus address	Data length（word）		CRC
			High Byte	Low byte	
1 Byte	10	2Byte	1Byte	1Byte	2Byte

Note

If there is error such as address over range,non-exist address and the address is read only,then it will return function code 0x90.

Example: Send packet 01 10 6F 00 00 02 04 55 55 00 08 1A 47
Message meaning:
01——ID number
10——Function code, write multiple WORD
6F 00——The modbus address of the servo writable object "target speed" 60FF0020, the data length is 2 WORD;
00 02——write 2 WORD
04——Data length is 4 Byte（2 WORD)；
55 55 00 08——Write data in hexadecimal 00085555, decimal 546133, converted to 30RPM;
1A 47——Check code

10.5 Modbus message example

The following is an example of sending packets in different modes. The station number is 1.

485 Message format

Modbus address	Name	Note	Message（ID=1）
3500	Operate mode	Operate mode is 3	01 06 35 00 00 03 C6 07
6F00	Target speed	Speed 150RPM	01 10 6F 00 00 02 04 55 55 00 08 1A 47
3100	Control word	Enable to write F	01 06 31 00 00 0F C7 32
3200	Statu word	Read drive status	01 03 32 00 00 02 CA B3

Homing control mode (control word first F then 1F)				
Internal address	Name	Value	Message（ID=1）	Note
60400010	Control word	F	01 06 31 00 00 0F C7 32	
60600008	Operate mode	6	01 06 35 00 00 06 06 04	
60980008	Homing mode	33	01 06 4D 00 00 21 5E BE	
60990120	Homing transition signal speed	200RPM	01 10 50 10 00 02 04 55 55 00 08 0E BA	
60990220	Homing signal speed	150RPM	01 10 50 20 00 02 04 40 00 00 06 98 76	
60400010	Control word	1F	01 06 31 00 00 1F C6 FE	
01 03 32 00 00 02 CA B3 Read the status word. C037 indicates that the origin is found				

Position （Controlword Absolute positioning 2F to 3F Relative positioning 4F to 5F， 103F Start absolute positioning)				
Internal address	Name	Value	Message（ID=1）	Note
60400010	Control word	F	01 06 31 00 00 0F C7 32	
60600008	Operate mode	1	01 06 35 00 00 01 47 C6	
607A0020	Target position	50000inc	01 10 40 00 00 02 04 C3 50 00 00 FE 39	
60810020	Trapezoid speed	200RPM	01 10 4A 00 00 02 04 55 55 00 08 BC D6	

60830020	Trapezoidal acceleration	100.00rps/s	Use default value	
60840020	Trapezoidal deceleration	100.00rps/s	Use default value	
60400010	Control word	2F	01 06 31 00 00 2F C6 EA	
		3F(Absolute positioning	01 06 31 00 00 3F C7 26	
		4F	01 06 31 00 00 4F C6 C2	
		5F(Relative positioning)	01 06 31 00 00 5F C7 0E	
01 03 32 00 00 02 CA B3 Reads the status word. D437 indicates the location recah				
Speed control mode				
Internal address	Name	Value	Message（ID=1）	Note
60600008	Operate mode	3	01 06 35 00 00 03 C6 07	
60FF0020	Target position	150RPM	01 10 6F 00 00 02 04 55 55 00 08 1A 47	
60400010	Control word	F	01 06 31 00 00 0F C7 32	
60830020	Trapezoidal acceleration	100.00rps/s	Use default value	
60840020	Trapezoidal deceleration	100.00rps/s	Use default value	

Note

Under communication mode, data are transmitted in HEX.

Chapter 11 CANopen Communication

11.1 CANopen communication protocol

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.

MD servo is a standard CAN slave device, which strictly follows the CANOpen 2.0A/B protocol, and any host computer that supports the protocol can communicate with it. The servo uses a strictly defined list of objects, we call it the object dictionary, this object dictionary is designed based on the CANopen international standard, all objects have a clear function definition. The Objects mentioned here are similar to the memory address we often say, some objects such as speed and position can be modified by the external controller, and some objects can only be modified by the drive itself, such as status and error messages. Table 11-1 lists these objects.

Table 11-1 the example object dictionaries lists

Complete CANOpen address			Attribute	Meaning
Index	Subindex	Bits(data length)		
0x6040	00	0x10	RW	Control word
0x6060	00	0x08	RW	Operation mode
0x607A	00	0x20	W	Target position
0x6041	00	0x10	MW	Status word

The properties of an object are as follows:

1. RW(Read and write) : Objects can be read or written;
2. RO(read-only) : The object can only be read;
3. WO(Write only) : Write only;
4. M(Mappable) : Objects can be mapped, similar to indirect addressing;
5. S(Storable) : Objects can be stored in the Flash-ROM area and are not lost when powered off.

11.2 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.

Note

This is pin definition of driver, not plugs.

Table 11-2 Pin name and function description

PIN	Name	Description
1	GND	CAN ground
2	GND	CAN ground
3	CAN_L	CAN_L bus(low dominant)
4	CAN_H	CAN_H bus （high dominant)

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 driver is built in and can be enabled by the dip switch SW8 or SW9;

3.Does not require external 24V power supply;；

4.Please use shielded twisted pair cables for communication cables, and do grounding treatment （The ground wire can be disconnected during short-distance communication, but it is recommended to connect the ground wire during long-distance and high baud rate communication）；

5.Table 11-3 lists the theoretical maximum communication distances of various baud rates.

Table 11-3 The longest distance table that can theoretically communicate with each baud rate

Communication speed（bit/s）	Communication distance（M）
1M	25
500K	100
250K	250
125K	500
50K	600

11.3 Software introduction

11.3.1 EDS introduction

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.

11.3.2 SDO introduction

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.

The communication mode of SDO is divided into upload and download, and the host computer can read and write the OD inside the servo according to the special SDO read and write instructions. In the CANopen protocol, the content of the Object dictionary can be modified through the Service Data Object (SDO). The following describes the structure of the SDO command and the guidelines to follow.

The basic structure of SDO is as follows：Client→Server/Server→Client

Identifier	DLC	Data							
		0	1	2	3	4	5	6	7
0x600+Node_ID	8	Send command word	object index		object subindex				

SDO message received when parameter read

Identifier	DLC	Data							
		0	1	2	3	4	5	6	7
0x580+Node_ID	8	receive command word	object index		object subindex	Maximum 4 Bit data			

Note

When SDO message are sent, commands are 0x40;

When received data is 1 byte, received command is 0x4F;

When received data is 2 bytes, received command is 0x4B;

When received data is 4 bytes, received command is 0x43;

If received data have errors, received command is 0x80.

Send SDO message when edit parameters

Identifier	DLC	Data							
		0	1	2	3	4	5	6	7
0x600+Node_ID	8	Send command word	Object Index		Object subindex	Maximum 4 bytes of data			

Receive SDO message (Modify parameters)

Identifier	DLC	Data							
		0	1	2	3	4	5	6	7
0x580+Node_ID	8	Receive command word	object index		object subindex	Maximum 4 Bit data			

Note

If sent data ready is 1 byte, command is 0x2F;

If sent data ready is 2 bytes, command is 0x2B;

If sent data ready is 4 bytes, command is 0x23;

If SDO message is sent successfully, receive command is 0x60;

If SDO message is not sent successfully, receive command is 0x80.

11.3.3 PDO introduction

PDO can transmit 8 bytes of data at a time, with no other protocol presetting (meaning that the data content is predefined), and is mainly used to transmit data requiring high frequency exchange.

The transmission mode of PDO breaks the existing data question-and-answer transmission concept and adopts a new data exchange mode. The two sides of the device define the data receiving and sending area in each device before transmission, and directly send the relevant data to the data receiving area of the other side during data exchange, which reduces the question-and-answer inquiry time and greatly improves the efficiency of bus communication. As a result, high bus utilization is achieved.

11.3.3.1 PDO COB-ID introduction

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 29 bits), which contains a 4-bit function code part and a 7-bit Node-ID part, as shown in Figure 11-1.

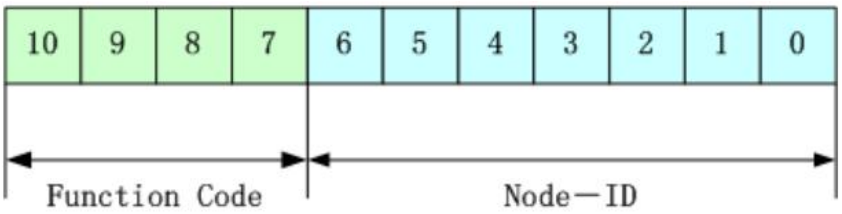


Figure 11–1 Default ID description diagram

Note

Node-ID —— Servo station No., Node-ID range is 1～127;

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

Table 11-4 CANopen predefined master/slave connection set CAN identifier assignment table

Object	COB-ID
NMT Module Control	000H
SYNC	080H
TIME SSTAMP	100H
Object	COB-ID
Emergency	081H-0FFH
PDO1（Send）	181H-1FFH
PDO1（Receive）	201H-27FH
PDO2（Send）	281H-2FFH
PDO2（Receive）	301H-37FH
PDO3（Send）	381H-3FFH
PDO3（Receive）	401H-47FH
PDO4（Send）	481H-4FFH
PDO4（Receive）	501H-57FH
SDO（Send/Server）	581H-5FFH
SDO（Receive/Client）	601H-67FH
NMT Error Control	701H-77FH

Note

1、The smaller the COB-ID,the higher the priority;

2、The function codes of COB-ID in every level are fixed;

3、COB-ID of 00H, 80H, 100H, 701H-77FH, 081H-0FFH are system management format;

11.3.3.2 COB-ID

- Send PDO relative to the servo refers to the data sent by the servo, which is received by the PLC. The function code (COB-ID) for sending the PDO is:
  1.0x180+Servo station NO.
  2.0x280+ Servo station NO.
  3.0x380+ Servo station NO.
  4.0x480+ Servo station NO.
- Receiving PDO relative to the servo refers to the data received by the servo, which is sent by the PLC, and the function code (COB-ID) for sending PDO is:
  1.0x200+Servo station NO
  2.0x300+Servo station NO
  3.0x400+Servo station NO

Note

Since MD series servo products are designed according to the standard CANopen 2.0A protocol, but also support CANopen 2.0B protocol, that is, if the above 8 PDO is not enough, you can also define a new PDO, such as 0x43FH as the communication PDO of station 1. As long as both the controller and the servo are defined according to this.

11.3.3.3PDO transmission types

PDO has two different transmission type：

SYNC - Transmission triggered by a synchronization message (transmission type: 0-240)

In this transmission mode, the controller must have the ability to send synchronous messages (periodic messages with a frequency of up to 1KHZ), which the servo sends after receiving the synchronous message.

Aperiodic - pre-triggered transfer by a remote frame, or by an object-specific event specified in the device subprotocol. In this mode, the data in the PDO is sent once every time the servo drive receives a synchronization message.

Periodic - Delivery is triggered after every 1 to 240 SYNC messages. In this mode, the data in the PDO is sent once every time the servo drive receives n synchronization packets.

Asynchronous (Transfer 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 the integrated servo, it supports all 256 transmission modes, and the user only needs to select the transmission mode of the drive according to the transmission mode supported by the controller.

Note

A PDO can specify a forbidden time, that is, define the minimum interval between two consecutive PDO transmissions, to avoid the problem that the data of high priority information always occupies the bus because the amount of data is too large, and other low priority data cannot compete with the bus. The forbidden time is defined by a 16-bit unsigned integer, in 1ms.

11.3.3.4 Protection mode (Supervision)

Monitoring type refers to the check method selected by the master station to check the slave station during operation. Through these two ways, the slave station can be judged whether there is a fault, and the corresponding treatment is made according to these faults!

1. Master heartbeat message

The slave station periodically uploads the message to the master station at the "monitoring time". If the master station does not receive the next heartbeat message from the slave station after the "heartbeat consumer time", the master station determines that the communication is wrong and the master station generates an alarm!

Message format -- (0x700+ node number) + status

Status - 0: Start, 4: Stop, 5: Run, 127: preoperation

2、Node protection

The master station periodically sends the message to the slave station with the "supervision time". If the slave station has not received the node message sent by the master station after the "supervision time * life factor" time, then the slave station alarms!

Master request message format——（0x700+node number） （The message has no data）

Slave response message format——（0x700+node number）+state

Status - The data part includes a trigger bit (bit7), which must be alternately set to '0' or '1' in each node guard response. The trigger bit is set to '0' on the first node guard request. Bits 0 to 6 (bit0 to 6) are used to indicate the node status;0: initialization, 1: disconnected, 2: connected, 3: operation, 4: stop, 5: Run, 127: preoperation.

The standard CAN slave station generally supports only one node protection mode, and the MD series servo supports both protection modes.

11.3.3.5 Boot-up process

CANopen supports both extended boot-up and minimal boot-up procedures during network initialization. The initialization process can be represented by a node state transition diagram, as shown in Figure 11-2

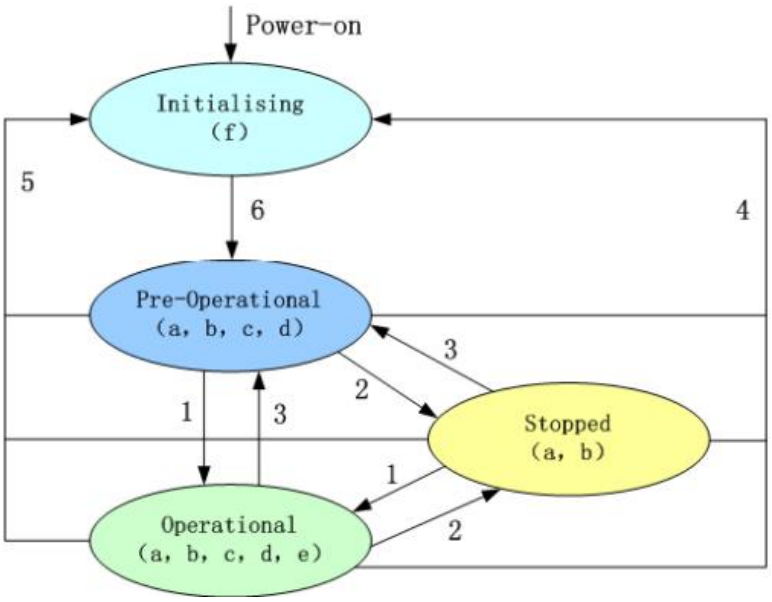


Figure 11-2 Node state transition diagram

Table 11-5 CANopen network status

Code	Meaning
a	NMT
b	Node Guard
c	SDO
d	Emergency
e	PDO
F	Boot-up

Table 11-6 CANopen network status change

Code	Meaning
1	Start_Remote_node
2	Stop_Remote_Node
3	Enter_Pre-Operational_State
4	Reset_Node
5	Reset_Communication
6	After the device initialization is complete, the device automatically enters the Pre-operation state and sends a Boot-up message

Management message format

COB-ID	DLC	Byte0	Byte1
0x000	02	CS	Node-ID

When Node-ID=0, all NMT slaves are addressed.CS is the command word, and its value is shown in Table 11-7.

Table 11-7 CS value

Command	NMT service
0x01	Start the remote node

0x02	Shut down the remote node
0x80	Enter pre-operational state
0x81	Enter pre-operational state
0x82	Reset communication

Switching between modes can be achieved through NMT management messages, only the NMT-Master node can send NMT Module Control messages, all slave devices must support the NMT Module Control service, and NMT Module control messages do not need to be answered.

NMT message format is as follows: Nmt-master →NMT Slave(s)

Note:

Only in operation status 0x5, PDO can be transmitted. If users want to open node which operation status is 6, then controllers can send messages below:

COB-ID	DLC	Byte0	Byte1
0x000	02	01	06

11.4 CANopen communication example

11.4.1 Connect to KincoServo+

Configure CANopen communication parameters using the KincoServo+ software, which needs to be installed.

Installation package download address：https://www.kinco.cn/download/software/servo

After the software is installed, the software page is displayed, as shown in Figure 11-3.

Figure 11–3 Software surface

Click the menu bar "Communication" -> "Communication Settings", enter the communication Settings interface, set COM port, drive ID. The default baud rate is 38400, and the default drive ID is 1. If you do not know the drive ID, you can set it to broadcast address 127,click open after the Settings are complete.

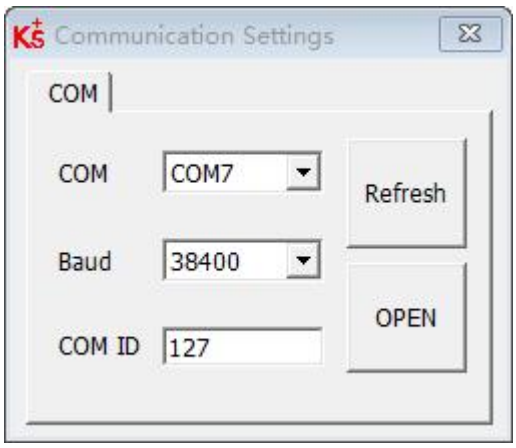


Figure 11-4 Communication Settings window

Note

- Following the logic that IO port control takes precedence over communication control, it is necessary to disable the IO port function when using CANopen communication. If there is a drive enabled, working mode switch function, then the communication will not be able to control the servo control word, working mode and other objects.
- Set the drive ID using the dip switch on the drive body. For the setting method, see the silk-screen description on the servo

11.4.2 Configure CANopen parameters

Note

- For CANopen parameters, please check **driver→ECAN configuration→Others**

When the master station with the network management function is powered on, the parameters of the slave station are initialized by sending SDO. Generally, parameters such as synchronization ID, node protection time, node protection time coefficient, node protection station number, emergency message station number, and heartbeat message generation time do not need to be set by the user.

	Index	Name	Value	Unit
1	101801	Vendor_ID		HEX
2	301107	ECAN_Sync		HEX
3	100500	Sync_ID		HEX
4	100C00	Guard_Time		DEC
5	100D00	Life_Time_Factor		DEC
6	100E00	Node_Guarding_ID		HEX
7	101400	Emergency_Mess_ID		HEX
8	101700	Producer_Heartbeat_T...		DEC
9	2F8100	CAN_Baudrate		DEC
10	301101	ECAN_Sync_Cycle		DEC
11	301102	ECAN_Sync_Clock		DEC
12	301103	ECAN_Sync_Shift		DEC
13	301104	Sync_TPDO_Diff		DEC
14	600700	Abort_Connection_M...		DEC

Figure 11–5 CANopen parameter setting window in KincoServo+

Table 11–8 CANopen related parameters

CANopen address	Name	Meaning	Default
10050020	Synchronization ID	The transmission type ranges from 1 to 240. This parameter is available in synchronous mode but does not need to be set in asynchronous mode.	80
100C0010	Node protection time	Through node protection, the master station can monitor the current status of each node. The master station sends a remote frame (the default COBID is 0x700+ station number and contains no message) to query the status of the slave node according to the node protection period. The slave node needs to respond within a certain time range, otherwise the master node considers the slave node offline and the driver enters the alarm state.	1000
100D0008	Node protection time coefficient		3
100E0020	Node protection ID	700+ drive ID	
10140020	Emergency message station number	80+ drive station number	
10170010	Timestamp of heartbeat message generation	The slave node periodically sends message to the master node. If the master node does not receive the message within a certain period of time, the slave node is considered to be disconnected	0
2F810008	CAN baud rate	CAN baud rate Setting 100：1M 50：500k 25：250k 12：125k 5： 50k 1： 10k	50
30110108	ECAN synchronization period	In interpolation mode, this parameter is set based on the synchronization packet interval of the master station. In asynchronous mode, this parameter is not required. 0:1ms 1:2ms 2:4ms 3:8ms	2
30110208	ECAN Synchronous clock mode	Set to 1 in interpolation mode to enable the synchronization clock, and set 0 to turn off the synchronization clock in non-interpolation mode	0
30110410	ECAN synchronization Lost count	In synchronization mode, the communication status is monitored. If the value does not change, the communication status is good. If the value keeps changing, interference occurs or the synchronization period is incorrectly set.	
60070010	Communication interrupt mode	CAN communication interruption mode: determines the action logic that the driver still does not receive the node protection packet after the node protection time x node protection coefficient is exceeded 0: No processing 1: An error is reported	0

Configure PDO parameters through PLC initialization

For the CANopen master station that can import EDS files, PDO Settings in the server are not required, and

PDO information can be directly configured in the master station. After power-on, PLC initialization will send SDO messages to configure the PDO of the server. After the configuration is completed, the master station will send startup messages to start the slave station, and then PDO communication can be carried out. Most PLCS can be used in this way, such as: Schneider PLC, Siemens S7-1200+CM CANOPEN module, Kinco F1 and so on.

➔ **Note**

EDS file download address:http://download.kinco.cn/D_Software/Servo/EDS.zip

Configure PDO parameters through KincoServo

There are also some PLCS that need to manually configure PDO parameters in the servo, for example:

Table 11-9 Common control objects

Name	CANopen	Address	RWS
Controlword	0x60400010	2 bytes	RW
Operation_Mode	0x60600008	1 byte	RW
Target_Position	0x607A0020	4 bytes	W
Target_Speed	0x60FF0020	4 bytes	W
Profile_Speed	0x60810020	4 bytes	W
Statusword	0x60410010	2 bytes	R
Pos_Actual	0x60630020	4 bytes	R

TPDO in servo is: (Servo send to PLC)

TPDO1: Actual position + status word

RPDO in servo is: (PLC send to servo)

RPDO1: Target position+Operation mode+Control word;

RPDO2: Target speed+profile speed;

➔ Overall object length in each PDO is not more than 8 bytes.

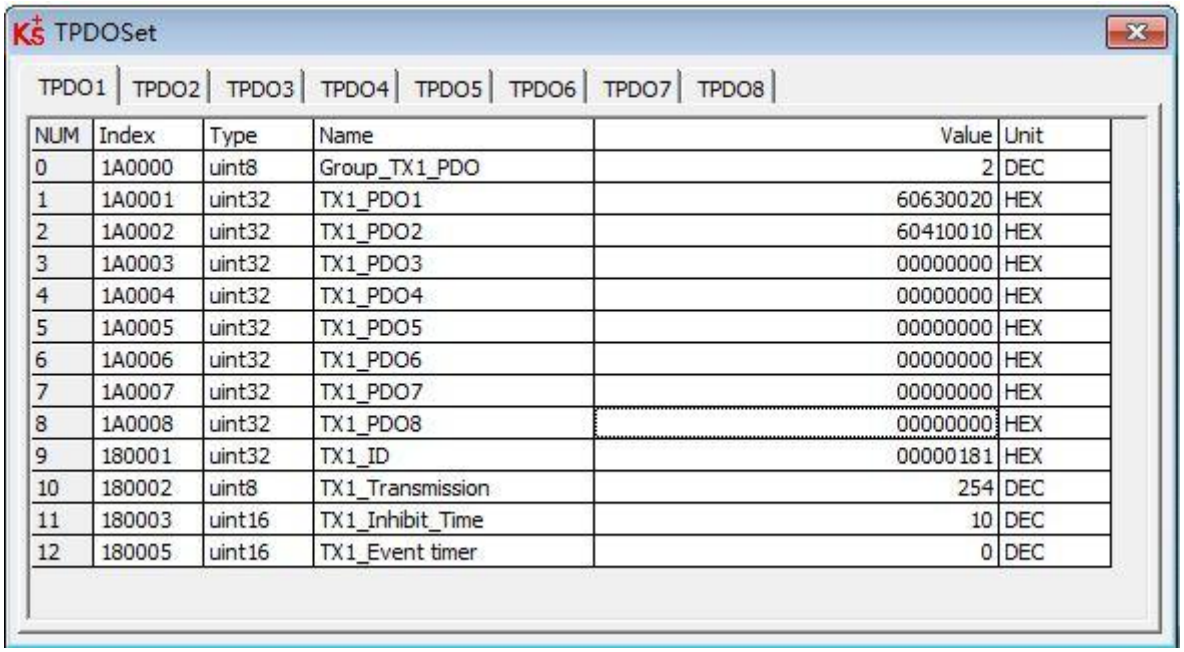
➔ Before using PDO to transmit data, it is necessary to send message open node of management message. Take No. 2 station for example

COBID	DLC	Message
000	02	01 02

11.4.3 PDO transmission mode configuration

Async transmission mode

In Async transmission mode, mapping data in PDO will transmit once they change.



NUM	Index	Type	Name	Value	Unit
0	1A0000	uint8	Group_TX1_PDO	2	DEC
1	1A0001	uint32	TX1_PDO1	60630020	HEX
2	1A0002	uint32	TX1_PDO2	60410010	HEX
3	1A0003	uint32	TX1_PDO3	00000000	HEX
4	1A0004	uint32	TX1_PDO4	00000000	HEX
5	1A0005	uint32	TX1_PDO5	00000000	HEX
6	1A0006	uint32	TX1_PDO6	00000000	HEX
7	1A0007	uint32	TX1_PDO7	00000000	HEX
8	1A0008	uint32	TX1_PDO8	00000000	HEX
9	180001	uint32	TX1_ID	00000181	HEX
10	180002	uint8	TX1_Transmission	254	DEC
11	180003	uint16	TX1_Inhibit_Time	10	DEC
12	180005	uint16	TX1_Event timer	0	DEC

Figure 11–6 Event time is used to schedule the upload in asynchronous mode

Table 11−10 TPDO configuration in asynchronous transport mode

Name	Meaning
TPDO1 Mapping group	Indicates the number of objects configured in the PDO. TPDO1 is configured with two objects: actual location and status word
Mapping 1-8	Configure the servo-CANOPEN control object
TPDO1 station number	180+ Drive ID (TPDO2 station number should be set to: 280+ drive ID)
TPDO1 Transport type	254 or 255, asynchronous transmission modev
TPDO1 Disable time	The unit is ms. This parameter prevents the network from being blocked by frequent packets sent by the server. In multi-axis asynchronous transmission mode, set this parameter based on site requirements

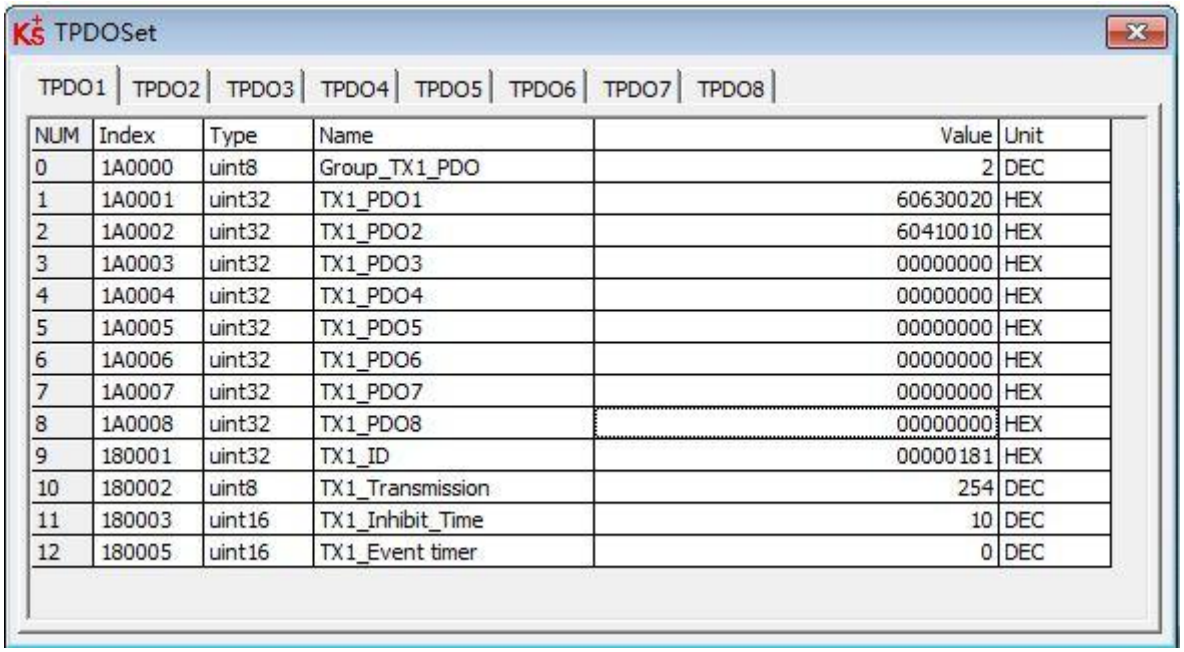
Note

- The sum of the actual position of the object and the length of the status word in TPDO1 is 4+2=6 bytes.
- The default RPDO transfer mode is 254. This parameter takes effect immediately after data is received

Event time timing reported function

In asynchronous transfer mode, in addition to the instantaneous, the event time can be set if the driver needs to periodically upload data to the controller.

The event time priority is higher than the disable time.



NUM	Index	Type	Name	Value	Unit
0	1A0000	uint8	Group_TX1_PDO	2	DEC
1	1A0001	uint32	TX1_PDO1	60630020	HEX
2	1A0002	uint32	TX1_PDO2	60410010	HEX
3	1A0003	uint32	TX1_PDO3	00000000	HEX
4	1A0004	uint32	TX1_PDO4	00000000	HEX
5	1A0005	uint32	TX1_PDO5	00000000	HEX
6	1A0006	uint32	TX1_PDO6	00000000	HEX
7	1A0007	uint32	TX1_PDO7	00000000	HEX
8	1A0008	uint32	TX1_PDO8	00000000	HEX
9	180001	uint32	TX1_ID	00000181	HEX
10	180002	uint8	TX1_Transmission	254	DEC
11	180003	uint16	TX1_Inhibit_Time	10	DEC
12	180005	uint16	TX1_Event timer	0	DEC

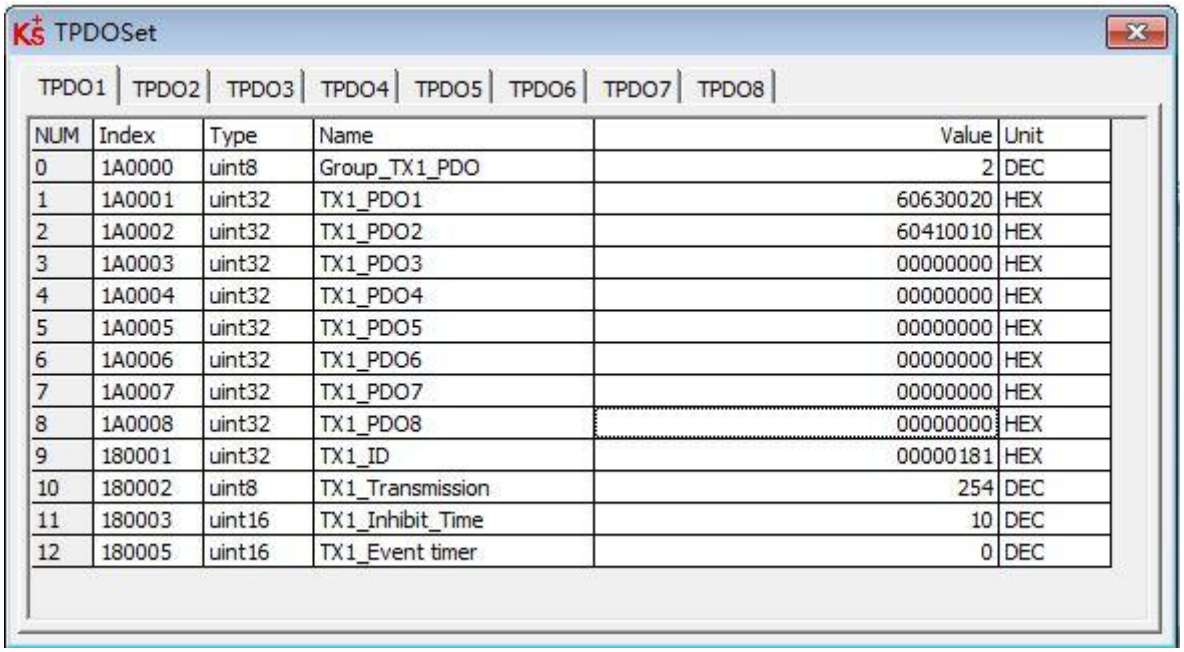
Figure 11–7 Event time is used to schedule the upload in asynchronous mode

Table 11−11 Event time is used to schedule the upload in asynchronous mode

Name	Meaning
TPDO1 Mapping group	Represents the number of objects configured in the PDO. TPDO1 is configured with two objects: actual location and status word
Mapping 1-8	Configure the servo-CANOPEN control object
TPDO1 station number	180+ Drive ID (TPDO2 station number should be set to: 280+ drive ID)
TPDO1 Transport type	254 or 255, asynchronous transmission mode
TPDO1 Disable time	When uploading using event time, this object is set to 0
TPDO1 Event time	Cycle time (in ms) for the driver to send PDO to the controller
Note: The sum of the actual position and status word length of the assigned object in TPDO1 is 4+2=6 bytes	

Sync transmission mode

When CANopen communication is configured in synchronous transmission mode, the driver uploads the mapped data in TPDO only after receiving the synchronization message.



NUM	Index	Type	Name	Value	Unit
0	1A0000	uint8	Group_TX1_PDO	2	DEC
1	1A0001	uint32	TX1_PDO1	60630020	HEX
2	1A0002	uint32	TX1_PDO2	60410010	HEX
3	1A0003	uint32	TX1_PDO3	00000000	HEX
4	1A0004	uint32	TX1_PDO4	00000000	HEX
5	1A0005	uint32	TX1_PDO5	00000000	HEX
6	1A0006	uint32	TX1_PDO6	00000000	HEX
7	1A0007	uint32	TX1_PDO7	00000000	HEX
8	1A0008	uint32	TX1_PDO8	00000000	HEX
9	180001	uint32	TX1_ID	00000181	HEX
10	180002	uint8	TX1_Transmission	254	DEC
11	180003	uint16	TX1_Inhibit_Time	10	DEC
12	180005	uint16	TX1_Event timer	0	DEC

Figure 11–8 TPDO configuration in synchronous mode

Table 11−12 TPDO configuration in synchronous mode

Name	Meaning
TPDO1 Mapping group	2, the number of objects configured in this PDO, TPDO1 is configured with two objects: actual location and status word
Mapping 1-8	Configure the servo-CANOPEN control object
TPDO1 station number	180+ Drive ID (TPDO2 station number should be set to: 280+ drive ID)
TPDO1 Transport type	Synchronous transmission mode: The driver sends TPDO to the controller after receiving synchronization packets
TPDO1 Disable time	It must be set to 0
Note: The sum of the actual position and status word length of the assigned object in TPDO1 is 4+2=6 bytes	

Note

- The sum of the actual position of the object and the length of the status word in TPDO1 is 4+2=6 bytes.
- The default RPDO transmission mode is 254. You do not need to set the transmission mode. The transmission mode takes effect immediately after data is received.
- The default value of synchronizing packets is

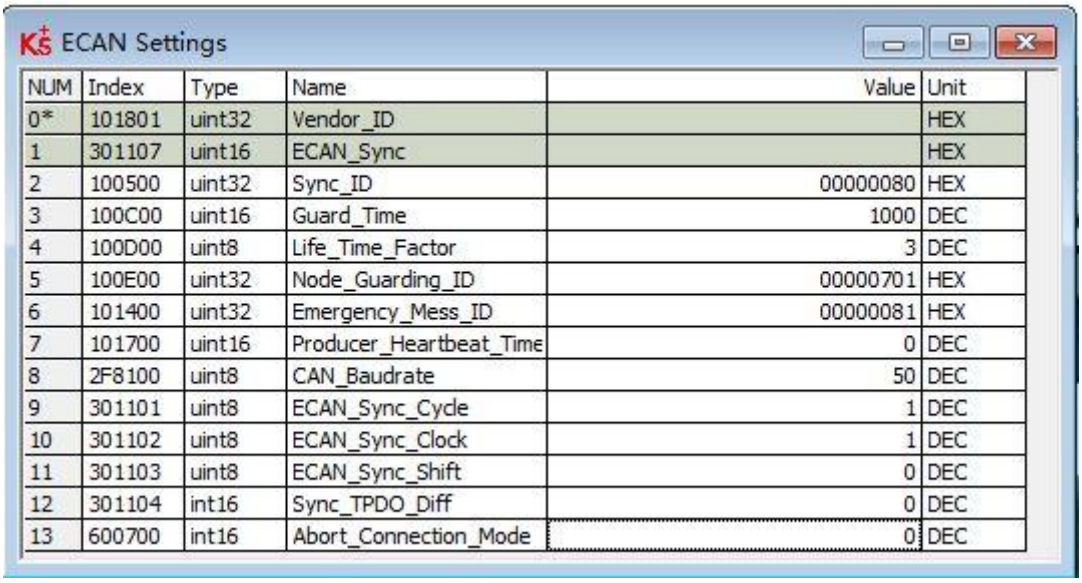
COB-ID	DLC
0x80	0

Differential complement mode based on CANopen

In differential complement mode based on CANopen, operation mode (0x60600008) is 7.

Table 11−13 The interpolation working mode is 7

Name	CANopen address	Length	property	Value
Operation mode	60600008	1 byte	RW	7



NUM	Index	Type	Name	Value	Unit
0*	101801	uint32	Vendor_ID		HEX
1	301107	uint16	ECAN_Sync		HEX
2	100500	uint32	Sync_ID	00000080	HEX
3	100C00	uint16	Guard_Time	1000	DEC
4	100D00	uint8	Life_Time_Factor	3	DEC
5	100E00	uint32	Node_Guarding_ID	00000701	HEX
6	101400	uint32	Emergency_Mess_ID	00000081	HEX
7	101700	uint16	Producer_Heartbeat_Time	0	DEC
8	2F8100	uint8	CAN_Baudrate	50	DEC
9	301101	uint8	ECAN_Sync_Cycle	1	DEC
10	301102	uint8	ECAN_Sync_Clock	1	DEC
11	301103	uint8	ECAN_Sync_Shift	0	DEC
12	301104	int16	Sync_TPDO_Diff	0	DEC
13	600700	int16	Abort_Connection_Mode	0	DEC

Figure 11-9 Setting parameters of the interpolation mode

Table 11−14 Interpolation control relate parameter

CANopen address	Name	Description	Value
30110108	ECAN sync period	In interpolation mode, it is set according to the synchronization message cycle of the master station 0: 1ms 1: 2ms 2: 4ms 4: 8ms	2
30110208	ECAN Synchronous Clock Mode	Set to 1 in interpolation mode to enable clock synchronization, and set to 0 in non-interpolation mode to disable clock synchronization.	0

CAN communication interruption alarm function

For the communication interruption alarm function, the following parameters need to be set

Table 11−15 Communication interrupt alarm function setting

CANopen address	Name	Meaning	Default value
100C0010	Node protection time	Through node protection, the master station can monitor the current status of each node. The master station sends a remote frame (the default COBID is	1000

100D0008	Node protection time coefficient	0x700+ station number and contains no message) to query the status of the slave node according to the node protection period. The slave node needs to respond within a certain time range, otherwise the master node considers the slave node offline and the driver enters the alarm state.	3
100E0020	Node protection ID	700+ drive ID	
10140020	Emergency message station number	80+ drive station number	
60070010	Communication interrupt mode	CAN communication interruption mode: determines the action logic that the driver still does not receive the node protection packet after the node protection time x node protection coefficient is exceeded 0: No processing 1: An error is reported	0

11.4.4 CANopen send message example

Node protection message and heartbeat message

CANopen node sends heartbeat packets at a fixed frequency. The packet is used to tell the controller that the communication is normal. The format of the packet is simple. COB-ID is 0x700+Node_ID, and the data is one-byte status data.

No	Time	CAN channel	Transmission direction	ID	Frame type	Frame format	Data length	Data
0	16:50:14:031	0	Receive	0706	Data frame	Standard frame	1	7F
1	16:50:15:093	0	Receive	0706	Data frame	Standard frame	1	7F
2	16:50:16:171	0	Receive	0706	Data frame	Standard frame	1	7F
3	16:50:17:234	0	Receive	0706	Data frame	Standard frame	1	7F
4	16:50:18:296	0	Receive	0706	Data frame	Standard frame	1	7F
5	16:50:19:375	0	Receive	0706	Data frame	Standard frame	1	7F
6	16:50:20:437	0	Receive	0706	Data frame	Standard frame	1	7F
7	16:50:21:500	0	Receive	0706	Data frame	Standard frame	1	7F
8	16:50:22:578	0	Receive	0706	Data frame	Standard frame	1	7F
9	16:50:23:640	0	Receive	0706	Data frame	Standard frame	1	7F
10	16:50:24:718	0	Receive	0706	Data frame	Standard frame	1	7F
11	16:50:25:781	0	Receive	0706	Data frame	Standard frame	1	7F
12	16:50:26:859	0	Receive	0706	Data frame	Standard frame	1	7F
13	16:50:27:921	0	Receive	0706	Data frame	Standard frame	1	7F
14	16:50:29:000	0	Receive	0706	Data frame	Standard frame	1	7F
15	16:50:30:062	0	Receive	0706	Data frame	Standard frame	1	7F

Figure 11-10 Node packets and heartbeat packets

In the figure, 706 indicates the heartbeat of node 06, and the status is 0x7F, indicating the Pre-Operational state (the node enters the Pre-Operational state after initialization). Check the time. The interval of each heartbeat packet is about 1 second.

NMT manages packets

NMT is a management message used to implement some management operations, such as restarting the node and entering the Operational state. The format of the NMT message is very simple, the ID is 000, and the data is one-byte command + one-byte node number (0 indicates broadcast).

No	Time	CAN channel	Transmission direction	ID	Frame type	Frame format	Data length	Data
0	16:51:35:296	0	Receive	0706	Data frame	Standard frame	1	7F
1	16:51:36:375	0	Receive	0706	Data frame	Standard frame	1	7F
2	16:51:37:796	0	Send	0000	Data frame	Standard frame	2	01 06
3	16:51:38:437	0	Receive	0706	Data frame	Standard frame	1	7F
4	16:51:39:500	0	Receive	0706	Data frame	Standard frame	1	05
5	16:51:40:562	0	Receive	0706	Data frame	Standard frame	1	05
6	16:51:41:625	0	Receive	0706	Data frame	Standard frame	1	05
7	16:51:42:687	0	Receive	0706	Data frame	Standard frame	1	05

Figure 11-11 Open node

The second packet enables node 06 to enter the Operational state. After the operation is complete, the node state in the node heartbeat packet also changes to the Operational state. PDO starts transmission only when the node is in this state.

No	Time	CAN channel	Transmission direction	ID	Frame type	Frame format	Data length	Data
0	16:51:47:843	0	Receive	0706	Data frame	Standard frame	1	05
1	16:51:48:906	0	Receive	0706	Data frame	Standard frame	1	05
2	16:51:49:968	0	Receive	0000	Data frame	Standard frame	1	05
3	16:51:50:031	0	Receive	0706	Data frame	Standard frame	1	05
4	16:51:51:578	0	Send	0000	Data frame	Standard frame	2	02 06
5	16:51:52:109	0	Receive	0706	Data frame	Standard frame	1	05
6	16:51:53:156	0	Receive	0706	Data frame	Standard frame	1	04
7	16:51:54:218	0	Receive	0706	Data frame	Standard frame	1	04
8	16:52:55:281	0	Receive	0706	Data frame	Standard frame	1	04

Figure 11-12 Close node

Send the stop remote node command to enter the Stopped state. Of course, the heartbeat is still there, but the node is not working

No	Time	CAN channel	Transmission direction	ID	Frame type	Frame format	Data length	Data
0	16:53:58:890	0	Receive	0706	Data frame	Standard frame	1	04
1	16:53:59:953	0	Receive	0706	Data frame	Standard frame	1	04
2	16:54:00:375	0	Send	0000	Data frame	Standard frame	2	81 06
3	16:54:01:015	0	Receive	0706	Data frame	Standard frame	1	04
4	16:54:02:093	0	Receive	0706	Data frame	Standard frame	1	00
5	16:54:03:156	0	Receive	0706	Data frame	Standard frame	1	7F
6	16:54:04:218	0	Receive	0706	Data frame	Standard frame	1	7F

Figure 11-13 Reset node

This is the reset node command, which is used to reset the node. After the reset, the system enters the initializing state (0x00 in heartbeat messages) and the Pre-Operational state (0x7F in heartbeat messages) after initialization.

Send and receive SDO

Sdo is primarily used to access the object dictionary (OD) of a node, and CANopen nodes need to support at least SDO_Server. The object dictionary is the data organization form of the CANopen node, and contains various parameters and data of the CANopen node, such as the sending frequency of heartbeat packets, the number of system startup times, and the communication parameters of the node. In other words, SDO is used to set the various running parameters of the CANopen node.

No	Time	CAN channel	Transmission direction	ID	Frame type	Frame format	Data length	Data
0	16: 55: 24: 421	0	Receive	0706	Data frame	Standard frame	1	05
1	16: 55: 25: 281	0	Send	0606	Data frame	Standard frame	8	40 17 10 00 00 00 00 00
2	16: 55: 25: 500	0	Receive	0706	Data frame	Standard frame	1	05
3	16: 55: 25: 500	0	Receive	0706	Data frame	Standard frame	8	4B 17 10 00 E8 03 00 00
4	16: 55: 26: 562	0	Receive	0706	Data frame	Standard frame	1	05

Figure 11-14 Send an SDO message to read the data

Message number 1 0606:40 17 10 00 00 00 00 00 00 is a SDO_Read message that tells the node the OD index and sub-index to be read, including the data length. The node then sends the corresponding data (the 8-byte message at the end). The first byte is the command word, the second and third bytes are the primary address of OD, the fourth byte is the sub-address of OD, and the last four bytes are the data. In the figure above, the host sends a command to read data in the OD 1017:00 location, which stores the heartbeat frequency, and the result is 0x03EB (1000ms).

No	Time	CAN channel	Transmission direction	ID	Frame type	Frame format	Data length	Data
0	17:09:35:828	0	Receive	0706	Data length	Standard frame	1	05
1	17:09:36:921	0	Receive	0706	Data length	Standard frame	1	05
2	17:09:38:015	0	Receive	0706	Data length	Standard frame	1	05
3	17:09:39:109	0	Receive	0706	Data length	Standard frame	1	05
4	17:09:40:187	0	Receive	0706	Data length	Standard frame	1	05
5	17:09:41:281	0	Receive	0706	Data length	Standard frame	1	05
6	17:09:42:375	0	Receive	0706	Data length	Standard frame	1	05
7	17:09:43:453	0	Receive	0706	Data length	Standard frame	1	05
8	17:09:44:546	0	Receive	0706	Data length	Standard frame	1	05
9	17:09:45:437	0	Send	0606	Data length	Standard frame	8	2B 17 10 00 FF 01 00 00
10	17:09:45:640	0	Receive	0706	Data length	Standard frame	1	05
11	17:09:45:640	0	Receive	0566	Data length	Standard frame	8	60 17 10 00 00 00 00 00
12	17:09:46:187	0	Receive	0706	Data length	Standard frame	1	05
13	17:09:46:734	0	Receive	0706	Data length	Standard frame	1	05
14	17:09:47:265	0	Receive	0706	Data length	Standard frame	1	05
15	17:09:47:812	0	Receive	0706	Data length	Standard frame	1	05
16	17:09:48:359	0	Receive	0706	Data length	Standard frame	1	05
17	17:09:48:906	0	Receive	0706	Data length	Standard frame	1	05
18	17:09:49:453	0	Receive	0706	Data length	Standard frame	1	05
19	17:09:50:000	0	Receive	0706	Data length	Standard frame	1	05

Figure 11-15 Send SDO messages to modify data

Packet No. 9 is SDO_Write, which writes data at 1017:00 in OD, that is, changes the heartbeat frequency. After receiving a complete response, the heart rate changes.

Various modes are used to send and receive data packets. For example, the station number is 1.

Homing（Controlword F to 1F)				
CANopen	Name	Value	Send and reply message（ID=1）	Meaning
60400010	Controlword	F	6012B40 60000F 00 5816040 60000F 00	
60600008	Operation_Mode	6	6012F60 600006 00 5816060 600006 00	
60980008	Homing_Method	33	6012F98 600021 00 5816098 600021 00	

60990120	Homing_Speed_Switch	200RPM	6012399 600155 55 08 00 5816099 600155 55 08 00	
60990220	Homing_Speed_Zero	150RPM	6012399 600200 40 06 00 5816099 600200 40 06 00	
60400010	Controlword	1F	6012B40 60001F 00 5816040 60001F 00	
6014041 600000 00 00 00read status word，C037means home found				

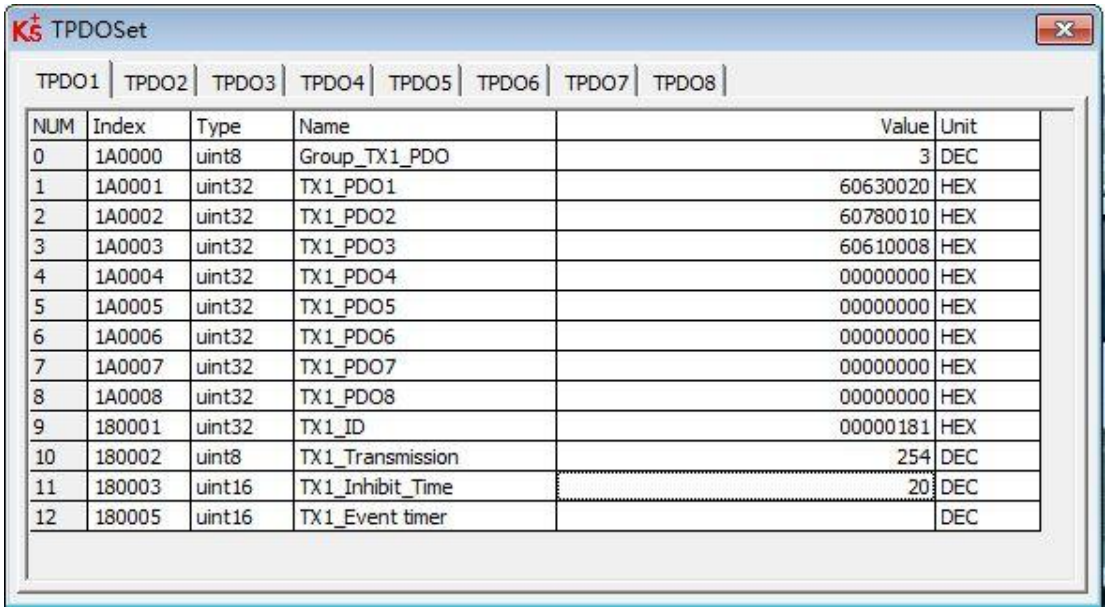
Position （Controlword Absolute positioning 2F to 3F Relative positioning 4F to 5F， 103F Start absolute positioning)				
CANopen	Name	Value	Message（ID=1）	Meaning
60400010	Controlword	F	6012B40 60000F 00 5816040 60000F 00	
60600008	Operation_Mode	1	6012F60 600001 00 5816060 600001 00	
607A0020	Target_Position	50000inc	601237A 600050 C3 00 00 581607A 600050 C3 00 00	
60810020	Profile_Speed	200RPM	6012381 600055 55 08 00 5816081 600055 55 08 00	
60830020	Profile_Acc	610.352rps/s	Default	
60840020	Profile_Dcc	610.352rps/s	Default	
60400010	Controlword	2F	6012B40 60002F 00 5816040 60002F 00	
		3F(absolute positioning)	6012B40 60003F 00 5816040 60003F 00	
		4F	6012B40 60004F 00 5816040 60004F 00	
		5F(relative positioning)	6012B40 60005F 00 5816040 60005F 00	
6014041 600000 00 00 00read status word，D437means position reach				

Speed				
CANopen	Name	Value	Message（ID=1）	Meaning
60600008	Operation_Mode	3	6012F60 600003 00 5816060 600003 00	
60FF0020	Target_Speed	150RPM	60123FF 600000 40 06 00 58160FF 600000 40 06 00	
60400010	Controlword	F	6012B40 60000F 00 5816040 60000F 00	
60830020	Profile_Acc	Default 610.352rps/s	Default	
60840020	Profile_Dcc	Default 610.352rps/s	Default	

Note：PDO Under communication mode, data are transmitted, sent and received in HEX.

SDO Send and received SDO

PDO is used to send (TPDO) or receive (RPDO) data, and there are several different ways to trigger it, such as asynchronous transmission, synchronous transmission, and so on. The data content of PDO is defined in MD by mapping. A node can have multiple PDO channels. The communication parameters of PDO can also be modified by the way SDO accesses OD.



NUM	Index	Type	Name	Value	Unit
0	1A0000	uint8	Group_TX1_PDO	3	DEC
1	1A0001	uint32	TX1_PDO1	60630020	HEX
2	1A0002	uint32	TX1_PDO2	60780010	HEX
3	1A0003	uint32	TX1_PDO3	60610008	HEX
4	1A0004	uint32	TX1_PDO4	00000000	HEX
5	1A0005	uint32	TX1_PDO5	00000000	HEX
6	1A0006	uint32	TX1_PDO6	00000000	HEX
7	1A0007	uint32	TX1_PDO7	00000000	HEX
8	1A0008	uint32	TX1_PDO8	00000000	HEX
9	180001	uint32	TX1_ID	00000181	HEX
10	180002	uint8	TX1_Transmission	254	DEC
11	180003	uint16	TX1_Inhibit_Time	20	DEC
12	180005	uint16	TX1_Event timer		DEC

Figure 11-16 Contents of the TPDO1 mapping group

TPDO1 maps three objects, which in turn are the actual position, the actual current, and the effective working mode.

From the PDO message captured in the following figure, it can be seen that the actual location of the drive is 0x5B944270.

```
835)      6560.6  Rx        0186  7  70 42 E8 FA 00 00 00
836)      6560.7  Rx        0181  7  70 42 94 5B 00 00 00
837)      6560.8  Rx        0182  7  70 42 23 C8 00 00 00
838)      6560.9  Rx        0183  7  70 42 25 11 00 00 00
839)      6561.0  Rx        0184  7  70 42 D5 E7 00 00 00
840)      6561.1  Rx        0185  7  70 42 3A 41 00 00 00
841)      6561.2  Rx        0287  8  20 9C FF FF 00 00 00 00
842)      6561.4  Rx        0187  7  70 C2 20 9C FF FF 00
843)      6562.4  Rx        0201  6  00 00 94 5B 00 00
844)      6562.5  Rx        0202  6  00 00 23 C8 00 00
845)      6562.6  Rx        0203  6  00 00 25 11 00 00
846)      6562.7  Rx        0204  6  00 00 D5 E7 00 00
847)      6562.8  Rx        0205  6  00 00 3A 41 00 00
848)      6562.9  Rx        0206  6  00 00 E8 FA 00 00
849)      6563.0  Rx        0207  6  00 00 20 9C FF FF
```

Figure 11-17 Intercepted PDO packet content

Appendix I Commonly used formulas

The selection method of the trolley motor is suitable for the mechanism of the motor + reducer + whee

Formula：T*n=μ*m*g*d/2	
Diameter of wheel d	m
The reduction ratio of the reducer n	1：n
Torque of motor T	Nm, kgm²/s²
Full load capacity m	kg
Friction coefficient μ	unitless
Gravitational acceleration g	m/s²

The relationship between the number of pulses and mechanical displacement in pulse mode

Formula：N*A/B=s*n*r/P	
Gear ratio molecule A	unitless
Gear score B	unitless
Ball screw lead P	mm
The number of pulses per turn of the motor r	unitless
Reduction ratio 1：n	unitless
Mechanical displacement s	mm
Formula：N*A/B=s*n*r/P	

The relationship between speed and linear velocity

Formula：n = v÷r÷π	
Speed n	rpm
linear velocity v	mm/s
Radius r	mm

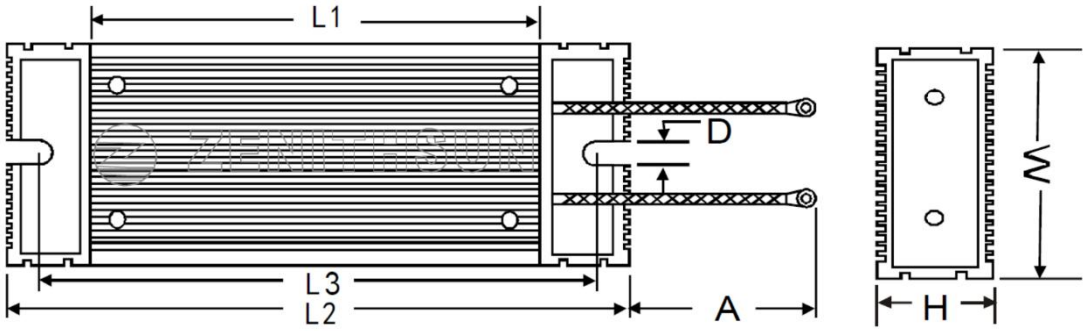
Parameter Name	Engineering unit	Internal unit	Conversion relation
Speed	rpm	DEC	DEC=[(RPM*512* encoder resolution)/1875]
Acceleration		DEC	DEC=[(RPS/S*65536* encoder resolution)/4000000]
Current	A	DEC	1Arms=[2048/(drive peak current Ipeak/1.414)]dec

For example, the speed engineering unit is rpm, the internal unit is dec, and the relationship between the two is that 1RPM is approximately equal to 2730dec (encoder resolution 10000)! Assuming that the required speed is 10rpm, the writing speed for communication control is 27300dec, and the hexadecimal number is 6AA4. similarly, when the motor encoder resolution is 65536, the relationship between the two is 1RPM approximately equal to 17896DEC. The current engineering unit is Arms and the internal unit is dec. Assuming that the driver used is MD60 (the peak driver current Ipeak is 50A), then 1Arms is approximately equal to 29dec. If the target current limit needs to be set to 10Arms, the write current needs to be 290dec when using communication control.

Model	MD60-020	MD60-040	MD80-075
Peak current（Ipeak)	50Ap	50Ap	80Ap

Appendix II Brake resistor

The energy generated by the servo motor in the braking state will be fed back to the DC bus of the driver. When the voltage value of the DC bus exceeds the protection range, the driver will report that the bus voltage is too high and the excess energy needs to be consumed by an external brake resistor. The resistance value of the optional brake resistor cannot be lower than the recommended resistance value. Connect the brake resistance through RB+ and RB- at the power end, and correctly set the brake resistance value and brake resistance power.



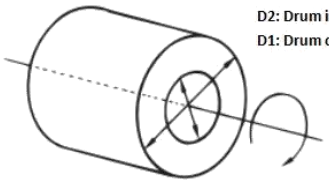
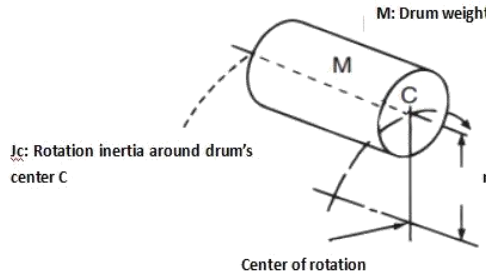
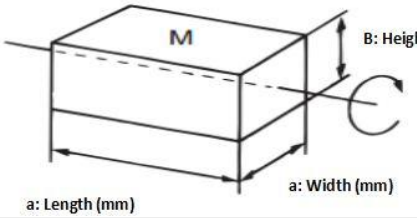
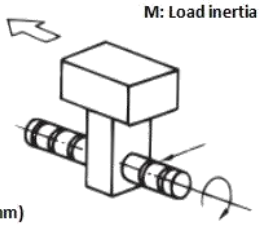
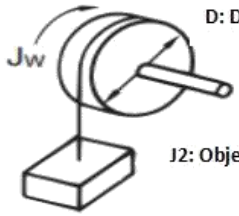
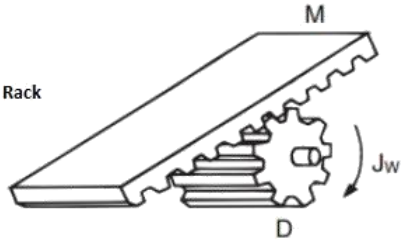
Power	Size						
	W±1	H±1	L1±2	L2±2	L3±2	D±0.5	A±10
100W	40	20	110	140	125	5.2	300

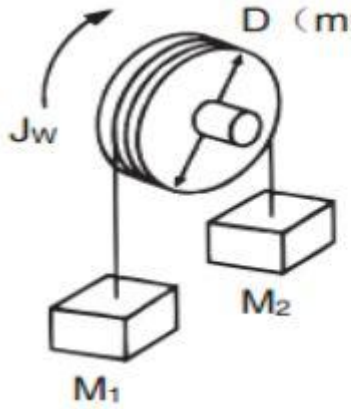
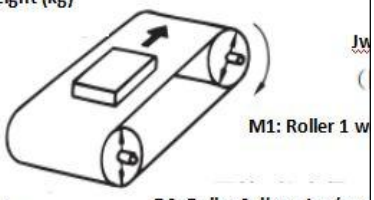
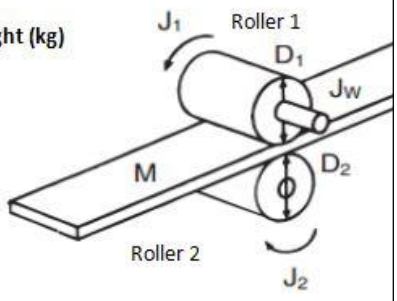
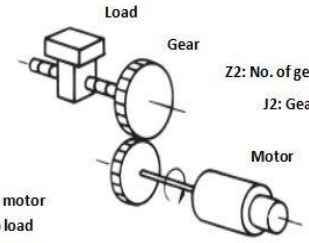
Driver type	Brake resistor type	Brake resistor resistance[Ω]	Brake resistor power[W]	Brake resistor voltage resistant [VDC] (Minimum)
MD60(200W)	T-27R-100	27	100	500
MD60(400W)	T-10R-100	10	100	500
MD80(750W)	T-5R-100	5	100	500

Brake resistance parameter setting

Address	Data type	Modbus address	RWS	Unit	Meaning
65100810	Unsigned16	0x6810	RW	V	Chopping voltage point, default 70V
60F70110	Unsigned16	0x6010	RW	Ω	Brake resistance
60F70210	Unsigned16	0x6020	RW	W	Brake resistance power

Appendix III General load inertia calculation

Drum inertia	 D2: Drum inside diameter (mm) D1: Drum outside diameter (mm) M: Drum weight (kg) Jw: Drum inertia （kg・m²）	$J_W=\frac{M(D_1^2+D_2^2)}{8}\times10^{-6}(kg\cdot m^2)$
Eccentric circular plate inertia/Drum inertia (Rotation center is shifted)	 M: Drum weight (kg) M C Jw: Drum inertia （kg・m²） Jc: Rotation inertia around drum's center C re: Rotation radius (mm) Center of rotation	$J_W=J_C+M\cdot re^2\times10^{-6}(kg\cdot m^2)$
Rotating prism inertia	 M: Prism weight (kg) M B: Height (mm) Jw: Inertia （kg・m²） a: Width (mm) a: Length (mm)	$J_W=\frac{M(a^2+b^2)}{12}\times10^{-6}(kg\cdot m^2)$
Rectilinear motion object inertia	 M: Load inertia (mm) JB: Slip-ball screw inertia （kg・m² P: Slip-ball screw pitch (mm) Jw: Inertia （kg・m²）	$J_W=M\left(\frac{P}{2\pi}\right)^2\times10^{-6}+J_B(kg\cdot m^2)$
Object inertia when it is lifted by pulley	 D: Diameter (mm) Jw M1: Drum weight (kg) J1: Drum inertia (kg・m²) J2: Object inertia (kg・m²) M2: Object weight (kg) Jw: Inertia (kg・m²)	$J_W=J_1+J_2$ $=\left(\frac{M_1\cdot D^2}{8}+\frac{M_2\cdot D^2}{4}\right)\times10^{-6}(kg\cdot m$
Object inertia when it is transmitted by rack or gear	 M Rack Jw D Jw: Inertia (kg・m²) M: Weight (kg) D: Gear diameter (mm)	$J_W=\frac{M\cdot D^2}{4}\times10^{-6}$ （kg・m²）

Inertia with counterweight	 Jw: Inertia (kg·m²) M1: Weight (kg) M2: Weight (kg)	$J_W=\frac{D^2(M_1+M_2)}{4}\times10^{-6}$（kg・m²）
Inertia when object is transmitted by conveyor belt	M3: Object weight (kg) M4: Conveyor belt weight (kg) D1: Roller 1 diame  Jw: Inertia (kg·m²) J1: Roller 1 inertia (kg·m²) M1: Roller 1 w D2: Roller 2 diameter (mm M2: Roller 2 weight (kg) J2: Inertia produced by roller 2 (kg·m²) J3: Inertia produced by object (kg·m²) J4: Inertia produced by conveyor belt (kg·m²)	$J_W=J_1+J_2+J_3+J_4$ $=\left(\frac{M_1\cdot D_1^2}{8}+\frac{M_2\cdot D_2^2}{8}\cdot\frac{D_1^2}{D_2^2}+\frac{M_3\cdot D_1^2}{4}+\frac{M_4\cdot D_1^2}{4}\right)\times10^{-6}$ （kg・m²）
Inertia when workpiece is nipped by roller	Jw: System inertia (kg·m²) J1: Roller 1 inertia (kg·m²) J2: Roller 2 inertia (kg·m²) D1: Roller 1 diameter (mm) D2: Roller 2 diameter (mm) M: Workpiece equivalent weight (kg) 	$J_W=J_1+\left(\frac{D_1}{D_2}\right)^2J_2+\frac{M\cdot D_1^2}{4}\times10^{-6}$ （kg・m²）
Load inertia when convert to motor shaft	 Z2: No. of gear near to load J2: Gear inertia near to load （kg・m²） Jw: Load inertia （kg・m²） Z1: No. Of gear near to motor J1: Gear inertia near to load （kg・m²） JL: Load inertia when convert to motor shaft （kg・m²） Variable transmission ratio G=Z1/Z2	$J_L=J_1+G^2(J_2+J_W)$(kg・m²)

Appendix IV Control Terminal Wiring Instructions

The MD series distributes the plug terminals and pins of X1 and X2 communication ports and X3 external output ports with the products. wires in the specification range of 30~22AWG shall be used together, and DuPont terminal crimping pliers shall be used to make cables.

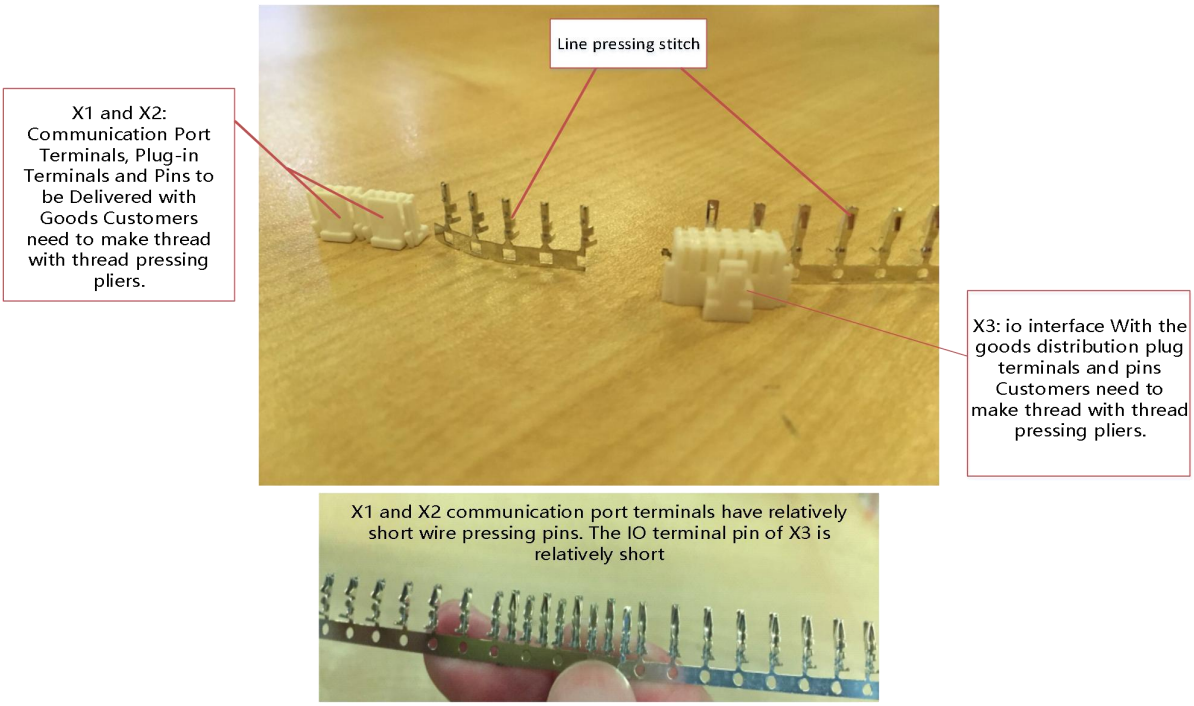
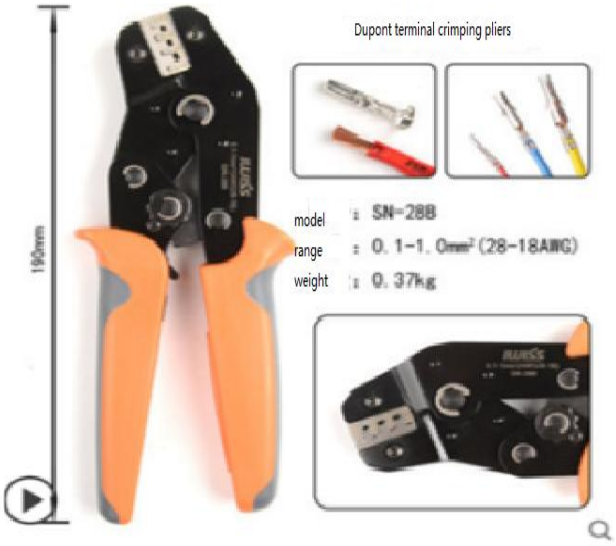


Figure. 1 description of crimping terminals and crimping pins



Instructions for use and purchase link of crimping tool (for reference only):
https://detail.tmall.com/item.htm?id=435619 04131&spm=a1z09.2.0.0.6f0e2e8dckA6sS&_u=d 2393kj15fc&skuId=3503574049108

Figure 2 Crimping pliers description

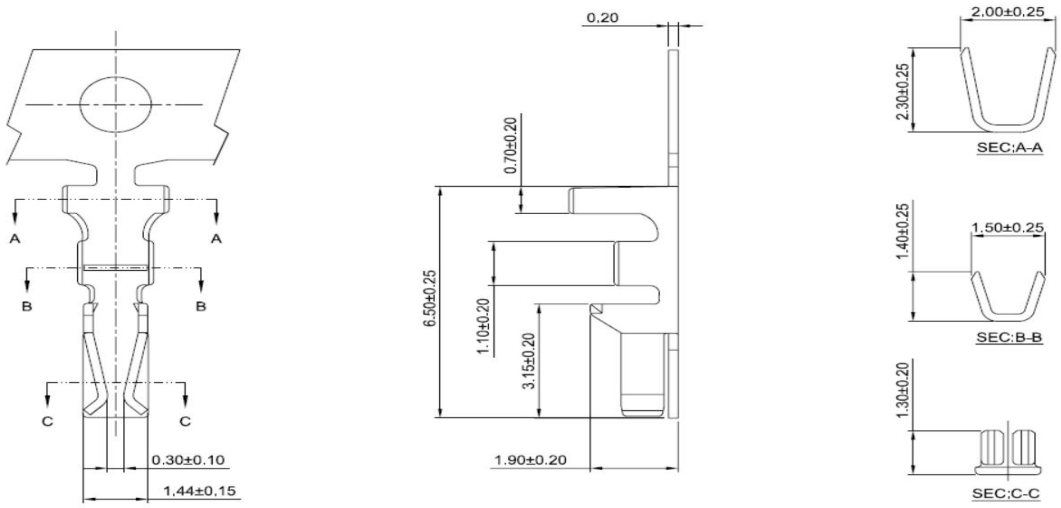


Figure 3 Specifications of Pins for X1 and X2 Communication Ports

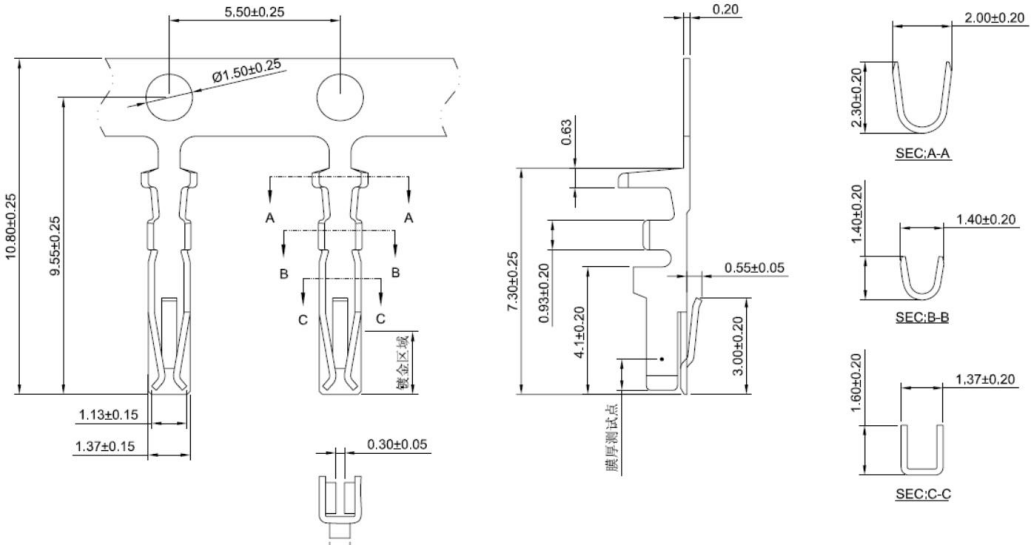


Figure 4 X3 IO Port Press Pin Specification

MD series integrated servo motor Parts list

Port		Docking terminal		
		Name	Specification model	Quantity
X1 (CANor RS485)		Plug	JST ZER-04V-S	2
		Pin	JST SZE-002T-P0.3	8
X2 RS232		2.0mm 4P Plug	CJT A2008H-04P	1
		Single row of metal pins	CJT A2008-TP	4
X3 IO		Terminal (Head)	JST PUDP-12V-S	1
		Crimp pin	JST SPUD-002T-P0.5	12
X5	MD60Power input	Terminal (Head)	DINKLE 0226-0704	1
	MD80Power input	Terminal (Head)	DINKLE 0227-0704	1