SVD48 Series

Servo Driver
User Manual

Version：V1.00

【Please read this manual carefully before use to avoid damaging the drive】

Content

Manual Version Revision Record

Date	Revision
2024.10.23	V1.0

Chapter 1 Precaution

1.1 Product Introduction

SVD48 series servo drive is a cost-effective servo motor drive independently developed by Changzhou Fulling Motor Co., Ltd .

It adopts a 32-bit microcontroller based on ARM core with 512K bytes of flash memory, with high integration and complete protection measures.

This series of drivers adopts a new control technology, which has the characteristics of small size, good performance, and high stability.

1.2 Characteristics

● Adopting ARM32-bit Cortex ™-M3 Core chip;

● Voltage range 20V-56V, supports wide voltage input;

● Supports incremental differential encoders (5V) and Tamagawa protocol communication encoders;

● Support external braking resistors;

● Supports serial communication, RS485 communication, and CAN communication;

Support input configuration (driver enable, alarm reset, emergency stop, positive limit, negative limit, multi-stage speed control, multi-stage position Control, etc.);

● Support output configuration (driver ready, driver error, motor zero speed, motor brake effective, limit effective, etc.);

● Equipped with overvoltage protection, undervoltage protection, motor overheating (I^2T) protection, short circuit protection and other driver protections;

● Ultra high cost-effectiveness.

1.3 Application Area

●Logistics robots: automatic navigation freight robots, shuttle cars, automatic parking robots, etc;

●Logistics equipment: fully automatic sorting lines, three-dimensional warehouses, etc;

●Medical equipment industry: small-scale systems;

●Other situations that require high response speed and high positioning accuracy.

1.4 Confirmation Items (Confirm Packaging and Accessories)

Check if the packaging is intact without obvious damage, deformation, or moisture.

Confirm that the received goods match the ordered product model, quantity, and specifications.

Check that the appearance of the drive and motor is intact without obvious scratches, cracks, or rust.

Confirm that the goods include the user manual, warranty card, certificate of conformity,

and other related documents for the drive and motor..

Table 1-1 SVD4812 Goods List

Goods List			
Item	Model	describe	Quantity
Driver	SVD4812RC-AA	Driver	1
Plug-in Connector	2EDGK-5.08-2P	2PIN spacing 5.08 Used for power supply and braking resistor interfaces	2
Plug-in Connector	2EDGK-5.08-4P	4PIN spacing 5.08 Used for logical power interface	1
Plug-in Connector	TB13-15K/3.81-02P	2PIN spacing 3.81 Used for logical power interface	1
Plug-in Connector	15EDGKNG-3.5-2x7P	2x7PIN spacing 3.5 Used for IO interface	1
Anti-static Bag	Flat anti-static bag 140*200mm	-	1
Box	Material:K6K(Very hard) Diameter:17*9*4mm	-	1

Table 1-2 SVD4822 Goods List

Goods List			
Item	Model	describe	Quantity
Driver	SVD4822RC-AA	Driver	1
Plug-in Connector	2EDGK-5.08-2P	2PIN spacing 5.08 Used for braking resistor interfaces	1
Plug-in Connector	TB13-15K/3.81-02P	2PIN spacing 3.81 Used for logical power interface	1
Plug-in Connector	15EDGKNG-3.5-2x7P	2x7PIN spacing 3.5 Used for IO interface	1
Anti-static Bag	Flat anti-static bag 140*200mm	-	1
Box	Material:K6K(Very hard) Diameter:17*9*4mm	-	1

Table 1-3 SVD4835 Goods List

Goods List			
Item	Model	describe	Quantity
Driver	SVD4835 RC-AA	Driver	1
Plug-in	TB13-15K/3.81-02P	2PIN spacing 3.81	1

Connector		Used for logical power interface	
Plug-in Connector	15EDGKNG-3.5-2x7P	2x7PIN spacing 3.5 Used for IO interface	1
Anti-static Bag	Flat anti-static bag 140*200mm	-	1
Box	Material:K6K(Very hard) Diameter:17*9*4mm	-	1

1.5 Transportation and Storage Conditions

Before transportation, check if the appearance of the drive and motor is intact, without obvious damage or deformation, and if the cables are firmly connected without fractures or wear.

During transportation, use a dedicated packaging box or pallet to avoid direct contact with the ground or other objects to prevent impact or compression. The packaging box or pallet should have obvious signs such as "Fragile," "Attention to Moisture," "Do Not Invert," etc. Handle with care during transportation to avoid violent vibrations or bumps, and try to maintain a horizontal state without tilting or inverting.

After transportation, store the drive and motor in a dry, ventilated environment free from corrosive gases, avoiding direct sunlight or high temperatures. Keep the packaging box or pallet intact when storing, and do not stack too high or too heavy to prevent deformation or damage.

1.6 Technical Requirements

To use this product correctly and safely, you need to pay attention to the following aspects: The operating methods and precautions of this product, as well as possible risks and preventive measures, are clearly stated and warned in this document. You must read and comply with this information carefully to ensure that the operation process meets safety standards. Ignoring this information may cause personal injury or property loss. Therefore, before operating this product, please study and master the content in this document carefully.

1.7 Operator Requirements

Safety Awareness: Operators must have a good sense of safety, follow operating procedures and safety guidelines to ensure their own and equipment safety.

Technical Knowledge: For complex drive and motor systems, operators need to have certain technical knowledge to understand and solve some common faults and problems.

Follow Operation Guide: Operators should follow the user manual provided by the manufacturer, correctly set and adjust parameters to achieve the expected operating effect.

Avoid Operational Errors: Operators should avoid misoperation or improper operation,

such as overloading, reversing, etc., to avoid damaging equipment or causing safety risks. Emergency Handling: Operators need to know how to deal with emergencies, including stopping equipment, cutting off power, etc., to prevent accidents.

1.8 Environmental requirements

Table 1-2 Environmental requirements

Environment	**Conditions**
Operating temperature	0℃ - 40℃
Operating humidity	5 - 95%RH (non-condensation)
Storage temperature	-10℃ - 70℃ (non-freezing)
Storage humidity	Below 90%RH (non-condensation)
Protection class	IP20
Installation site	Indoor, free from sunlight, corrosive gas, flammable gas, oil gas, and dust
Installation method	Vertical installation or horizontal installation
Barometric pressure	85kpa~105kpa
Altitude	Rated working altitude below 1000 meters, for working altitudes above 1000 meters, reduce by 1.5% for every 100 meters increase, maximum working altitude 4000 meters

Chapter 2 System Configuration and Model Description

SVD	4812	NONE/RC/EC/ER	-	AA	-	XXX
①	②	③		④		⑤

①**Driver Series Name:** SVD Series

② **Rated Voltage/Rated Current:**
4812: 48V 12A

③ **Communication Type**
RC：RS-485 Communication + Can Communication
EC：Ether-Cat Communication + Can Communication
ER：Ether-Cat Communication + RS-485 Communication

④ **Encoder Type**
AA：Differential UVW + Differential ABZ Encoder, Tamagawa Encoder, TTL UVW + Differential AB encoder
HD：Differential UVW + Differential ABZ Encoder
CT：Tamagawa Encoder
HN：TTL UVW Signal
HE：TTL UVW + Differential AB encoder
CA：Custom Communication Magnetic Encoder

⑤ **Serial Number**
This code serves as the default function configuration code for the drive, including motor parameters, control loop parameters, I/O functions, communication function parameters, etc. Many formulas come from customer's field application needs, aiming to facilitate customer use, and to meet customer's field use as much as possible without the need for further configuration of related parameters.

Chapter 3 Installation Dimensions

3.1 Drive Installation

3.1.1 Driver Installation Dimensions

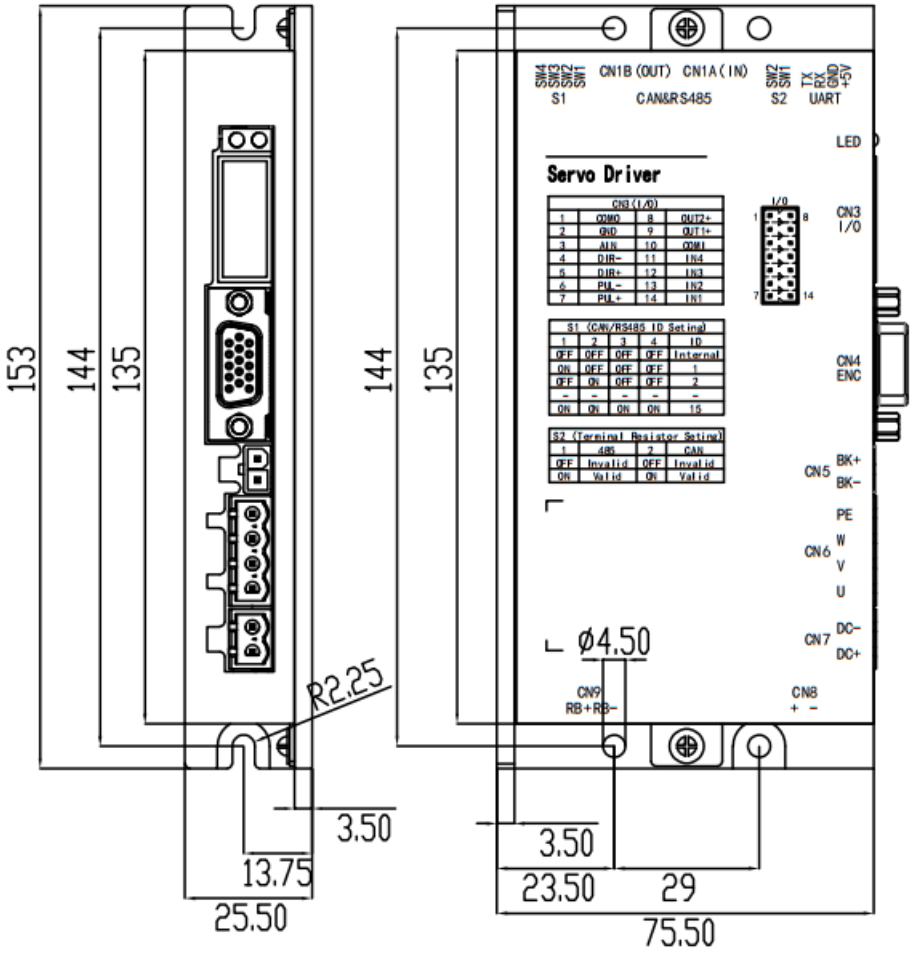


Figure 3-1 SVD4812RC-AA Drive Installation Dimensions

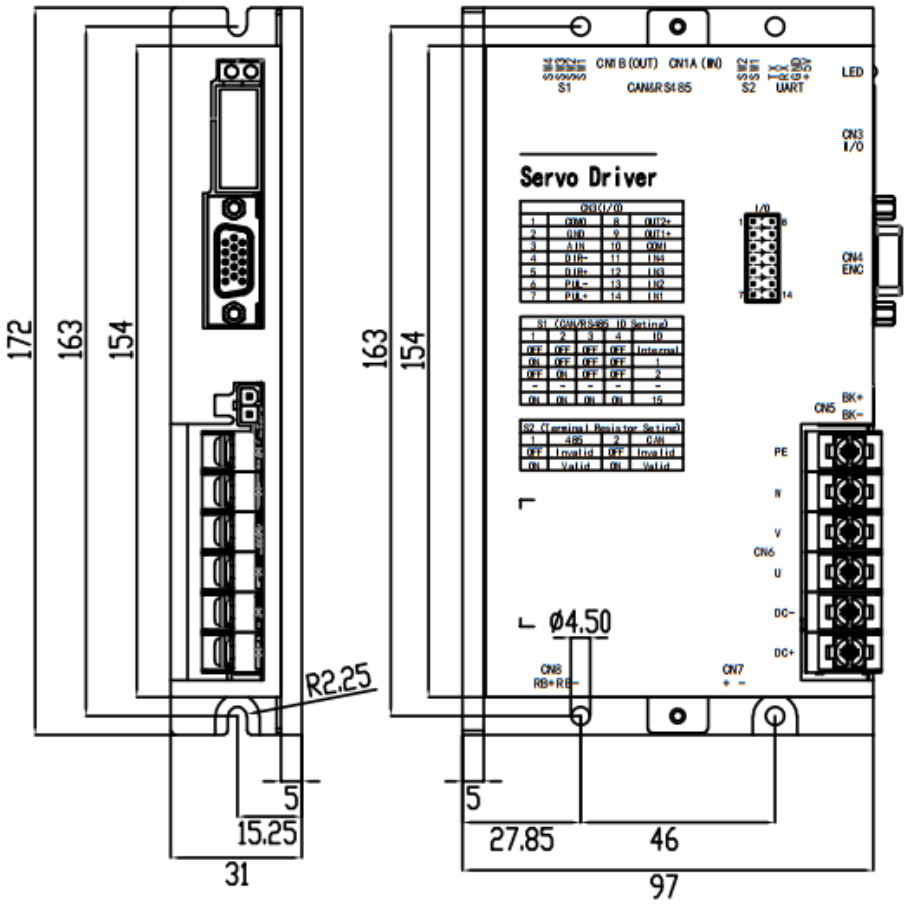


Fig.3-2 SVD4822RC-AA Driver Installation Dimensions

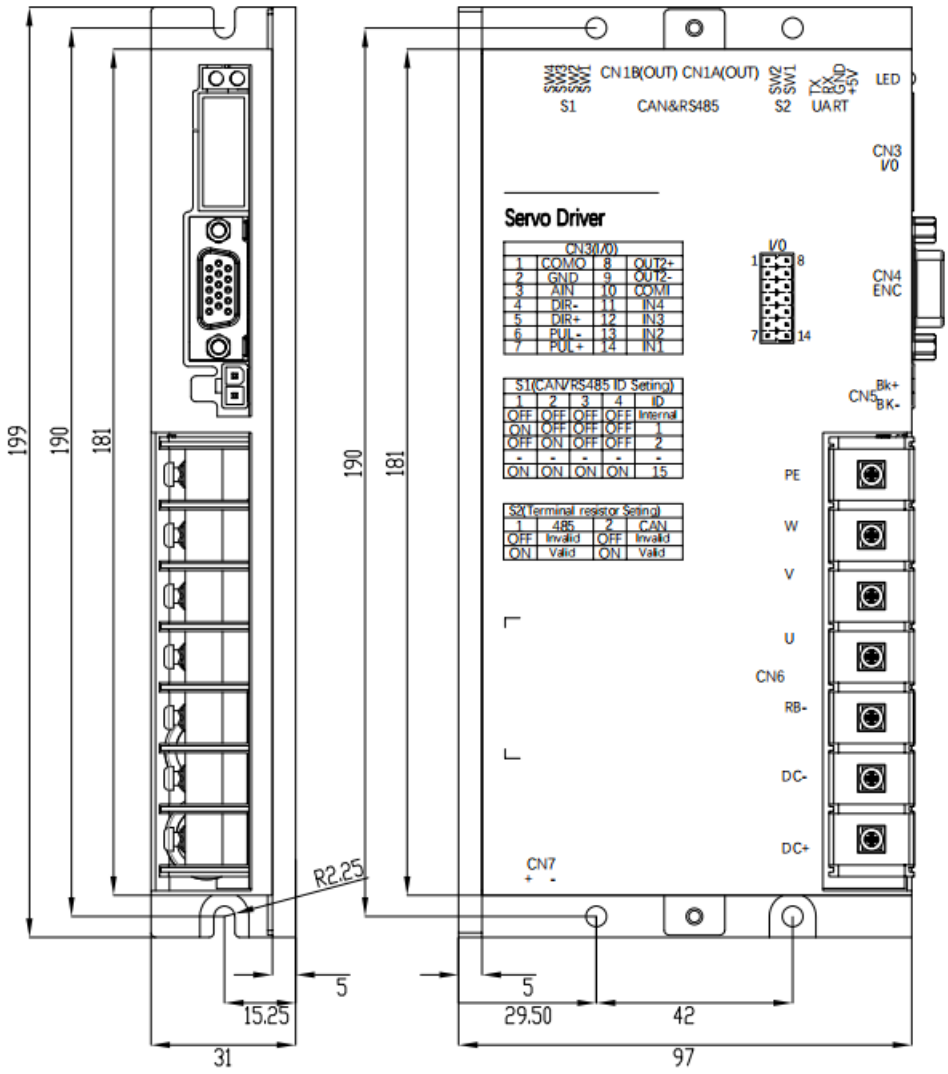


Fig.3-2 SVD4835RC-AA Driver Installation Dimensions

3.1.2 Precautions of Servo Drive Installation

Here are some precautions for installing the drive:

Location and Installation: The drive should be installed in a dry, well-ventilated environment, away from humidity, dust, and corrosive gases. Also, avoid installing in areas with high temperatures, direct sunlight, or frequent vibrations.

Fixing and Connection: Ensure that the drive is firmly installed on a stable bracket or surface. All screws and connectors should be properly tightened to prevent loosening and vibration. When connecting cables, ensure that the cable plugs and sockets are firmly in place to avoid pulling on the cables.

Heat Dissipation: If heat-generating components are used, such as braking resistors, heat

dissipation should be considered. Ensure that there is enough space around the drive for heat dissipation, and avoid piling up other objects that may affect the cooling effect. When installing the drive, the distance shown in Figure 3-2 can be used as a reference.

Electrical Connection: Ensure that the power and motor cables are correctly connected. Follow the correct wiring diagram or instruction manual. Ensure that the power supply meets the requirements and the voltage is stable. Pay attention to grounding and insulation issues to avoid electrical problems.

Protective Measures: When installing the drive, prevent foreign objects from entering the interior of the drive, especially conductive or flammable materials such as metal debris, dust, and oil. The drive and motor are precision equipment, avoid subjecting them to external forces and vibrations.

Inspection and Testing: After installation, carry out the necessary inspections and tests to ensure that the connections of the drive are correct and the functions are normal. Then, you can proceed with the corresponding tests according to the manual.

EMC Requirements: If the drive will be used in an industrial electromagnetic environment, it is necessary to comply with the requirements of electromagnetic compatibility (EMC). It may be necessary to install power filters, shielded cables, etc., to reduce interference and electromagnetic radiation.

Safety and Maintenance: After installation, ensure that appropriate safety measures are in place to prevent unauthorized personnel from approaching the drive. Regularly inspect the drive and connections to ensure there is no loosening or damage.

Please read the relevant drive installation manual and instructions in detail before installation, follow the guidance and recommendations provided by the manufacturer. If there is any uncertainty, it is recommended to consult the company for technical support.

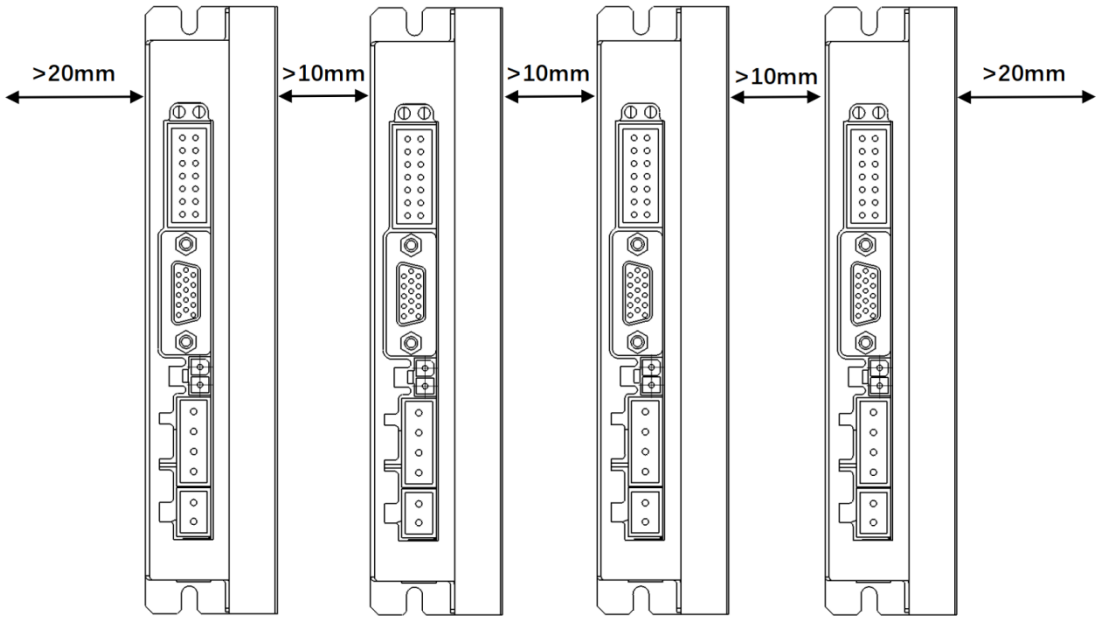


Fig.3-3 Recommended Installation Distance Example Diagram

3.1.3 Servo Drive Related Parameters

Drive Model		SVD4812RC-AA	SVD4822RC-AA	SVD4835RC-AA
Supported Motor Type		Servo motor		
Input Voltage Range		20~56V		
Continuous Current		12Arms（without auxiliary cooling） 15arms（with auxiliary cooling）	22Arms（without auxiliary cooling） 30arms（with auxiliary cooling）	35Arms（without auxiliary cooling） 45arms（with auxiliary cooling）
Peak Current		45Ap	100Ap	160AP
Feedback Signal		Incremental differential encoder(5V)、Tamagawa protocol communication encoder		
Dynamic Braking		Support external braking resistor		
Dynamic Braking Voltage		DC65V(default value, settable)		
Over-voltage Alarm Voltage		DC70V		
Under-voltage Alarm Voltage		DC18V		
Cooling Method		Natural cooling		
Weight		0.34kg	0.6kg	0.8kg
Common Functions	Input Specifications	4 digital inputs, common COMI terminal, high level: 12VDC~30VDC, low level: 0~5VDC, maximum frequency 1kHz, input impedance 5kΩ		
	Input Functions	Configurable functions include driver enable, alarm reset, emergency stop, positive limit, negative limit, multi-speed control, multi-position control, etc.		
	Output	2 digital outputs, common COMO terminal		

	Specifications	
	Output Functions	Configurable functions include driver ready, driver error, motor zero speed, motor brake effective, limit effective, etc.
	Brake Output	Default 24VDC brake output, configurable voltage 0VDC~input voltage, configurable brake duty cycle
	Pulse Control	Pulse + direction, CCW + CW, A phase + B phase (5V-24V)
	TTL232	Default baud rate 115200, maximum support 115200, can be connected with FULLING master station, can also use custom protocol to communicate with the controller
	Protection Functions	Overvoltage protection, undervoltage protection, motor overheating (I^2T) protection, short circuit protection, driver overheating protection, etc.
UART Baud Rate		115200bps (default value, can be modified)
RS485 Baud Rate		115200bps (default value, can be modified)
CAN Baud Rate		500Kbps (default value, can be modified)

3.2 Installation of Servo Motor

During the installation of the servo motor, the following points should be noted:

Do not directly connect the servo motor to the industrial power supply, as it will damage the servo motor.

Storage Temperature and Humidity: The servo motor should be stored within a temperature range of -10℃ to +70℃, with a relative humidity not exceeding 90%RH, without dew or condensation.

Servo Motor Oil and Water Protection: The servo motor can be used in places that may be affected by water or oil droplets, but it is not fully waterproof or oilproof. Therefore, the servo motor should not be placed or used in water or oil-soaked environments. If the servo motor is connected to a reduction gear, it should be sealed with oil when used to prevent oil from the reduction gear from entering the servo motor. The cables of the servo motor should not be immersed in oil or water; if necessary, use oil-resistant cables.

Servo Motor Oil and Water Protection: The servo motor can be used in places that may be affected by water or oil droplets, but it is not fully waterproof or oilproof. Therefore, the servo motor should not be placed or used in water or oil-soaked environments. If the servo motor is connected to a reduction gear, it should be sealed with oil when used to prevent oil from the reduction gear from entering the servo motor. The cables of the servo motor should not be immersed in oil or water; if necessary, use oil-resistant cables.

Servo Motor Allowable Shaft Load: Ensure that the radial and axial loads applied to the shaft of the servo motor during installation and operation are controlled within the specified values for each model. Be extra careful when installing a rigid coupling, as excessive bending load may cause damage or wear to the shaft end and bearings. It is best to use a flexible coupling to keep the radial load below the allowable value, which is designed for high mechanical strength servo motors.

Chapter 4: System Interface and Wiring

4.1 Names and Functions of Drive Parts

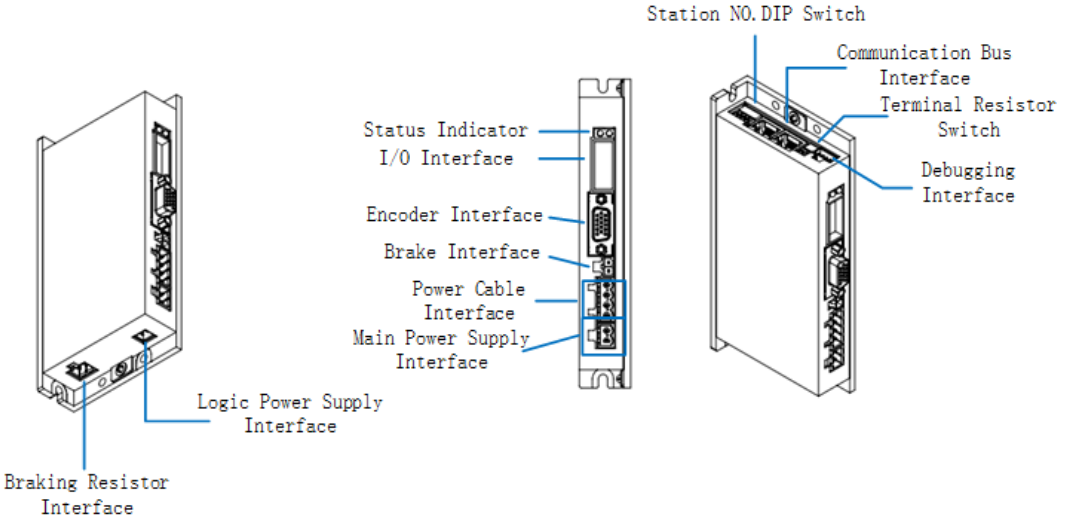


Figure 4-1 SVD4812RC-AA Names and Functions

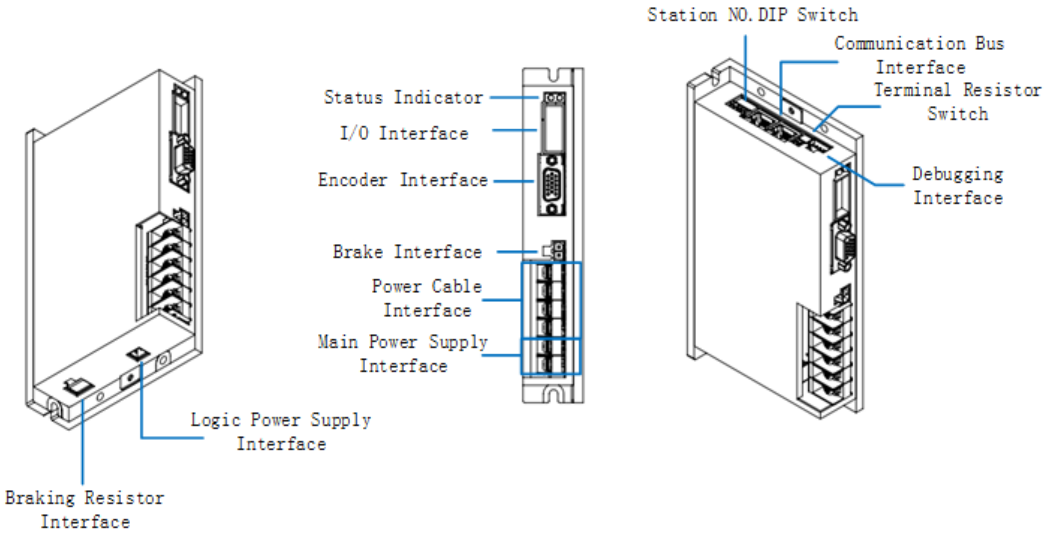


Figure 4-2 SVD4822RC-AA Names and Functions

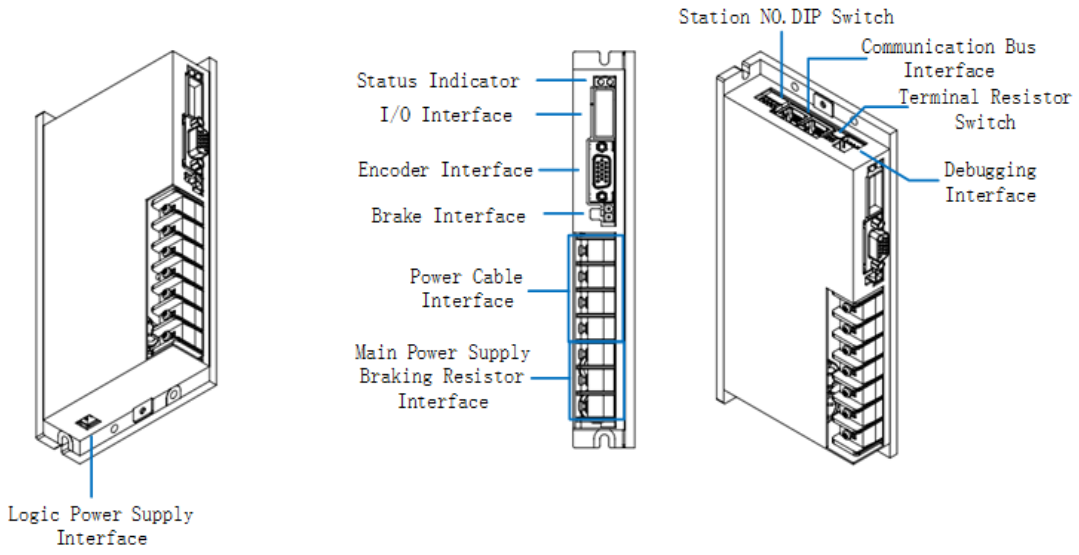
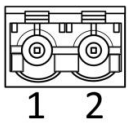
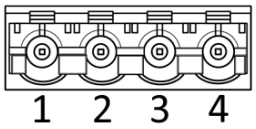
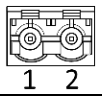
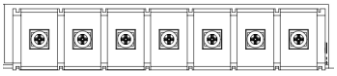


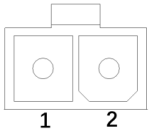
Figure 4-3 SVD4835RC-AA Names and Functions

4.1.1 Power Supply, Motor Cable, Braking Resistor Interface (CN7)

	Figure	Pin No.& Name	Description
SVD4812	CN7  1 2	PIN1:DC+	Power supply positive
		PIN2:DC-	Power supply negative
	CN6  1 2 3 4	PIN1:U	Motor power U phase
		PIN2:V	Motor power U phase
		PIN3:W	Motor power U phase
		PIN4:PE	Motor PE
	CN9  1 2	PIN1:RB+	Braking Resistor
		PIN2:RB-	
SVD4822	CN6  1 2 3 4 5 6	PIN1:DC+	Power supply positive
		PIN2:DC-	Power supply negative
		PN3:U	Motor power U phase
		PIN4:V	Motor power V phase
		PIN5:W	Motor power W phase

		PIN6:PE	Motor PE
	CN8  	PIN1:RB+	Braking Resistor
		PIN2:RB-	
SVD4835	CN6  	PIN1:DC+/RB+	Power supply positive Braking Resistor positive
		PIN2:DC-	Power supply negative
		PN3:RB-	Braking Resistor negative
		PIN4:U	Motor power U phase
		PIN5:V	Motor power V phase
		PIN6:W	Motor power W phase
		PIN7:PE	Motor PE

4.1.2 Brake Interface（CN5）

Figure	Pin No. & Name	Description
CN5  	PIN1:BK-	Brake output negative
	PIN2:BK+	Brake output positive

4.1.3 Encoder Interface (CN4)

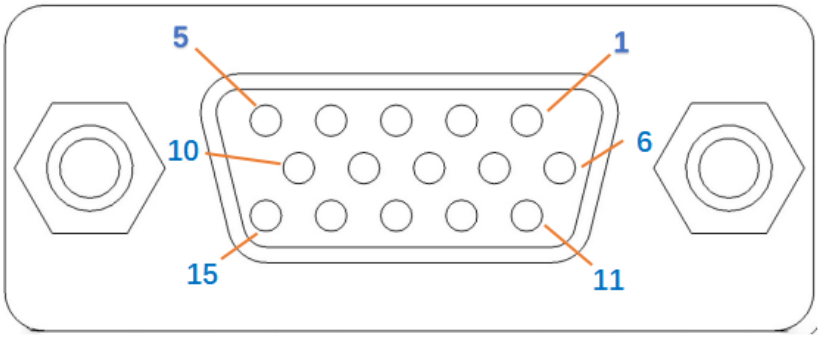


Figure 4-3 Encoder Interface Pin Definition

Incremental Encoder Interface Pin Definition

Interface	Pin. No. & Name	Description
CN4 ENCODER	PIN13: A -	Encoder input A-
	PIN12: B -	Encoder input B -
	PIN11: Z -	Encoder input Z -
	PIN5: U -	Encoder input U -
	PIN15: V -	Encoder input V -
	PIN14: W -	Encoder input W -
	PIN8: A +	Encoder input A +
	PIN7: B +	Encoder input B +
	PIN6: Z +	Encoder input Z +
	PIN4: U +	Encoder input U +
	PIN10: V +	Encoder input V +
	PIN9: W +	Encoder input W +
	PIN1: 5V+	Encoder 5V output
	PIN2: GND	Encoder signal ground

Single-turn/Multi-turn Communication Encoder Interface Pin Definition

Interface	Pin. No. & Name	Description
CN4 ENCODER	PIN1:ENC5V	Encoder 5V output
	PIN2:GND	Encoder signal ground
	PIN9:SD+	Data signal positive
	PIN14:SD-	Data signal negative

4.1.4 I/O Interface (CN3)

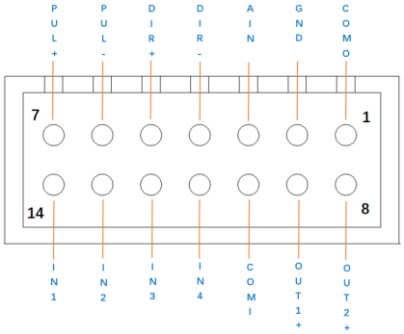


Figure 4-4 IO Interface Schematic

Interface	Pin. No. & Name	Description
CN3	PIN1:COMO	Output common terminal
	PIN2:GND	Ground terminal
	PIN3:AIN	Analog input termina
	PIN4:DIR-	Pulse input terminal Input voltage: 3.3V～24V Maximum input frequency: 500KHz
	PIN5:DIR+	
	PIN6:PUL-	

	PIN7:PUL+	
	PIN8:OUT2+	Output signal 2 positive, maximum output current: 50mA
	PIN9:OUT1+	Output signal 1 positive, maximum output current: 50mA
	PIN10:COMI	Input common terminal
	PIN11:IN4	Digital input (DIN), in DIN mode, through the digital logic input of the 4 pins, you can choose the corresponding different states. (For specific details, please refer to the corresponding chapter content) High level: Input voltage 12VDC~30VDC, input current 4-20mA Low level: Input voltage 0VDC~5VDC, input frequency: <1KHz
	PIN12:IN3	
	PIN13:IN2	
	PIN14:IN1	

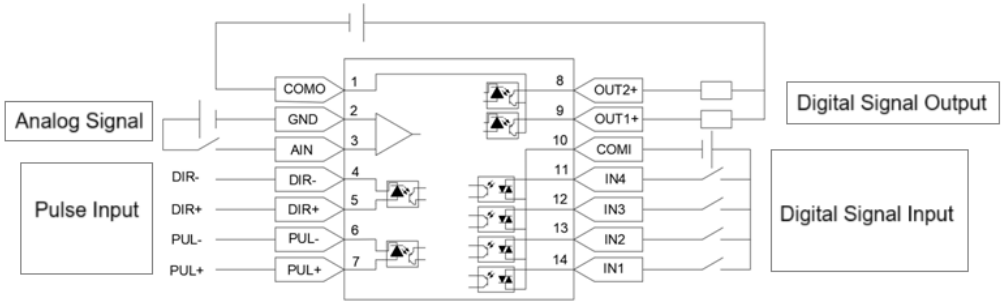
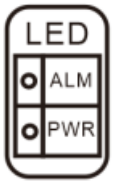


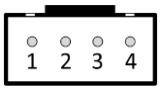
Figure 4-5 I/O Interface Wiring Diagram

4.1.5 Status Indicator Light (LED)

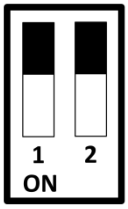
Figure	Pin No.&Name	Description
LED 	Motor fault and operation indicator light	Flashing red LED indicates an alarm on this machine, only green LED on indicates the system is running normally. Please refer to "Chapter 8 Alarms and Troubleshooting" for details.

4.1.6 Debugging Interface (CN2-UART)

Mainly used for UART communication, when using the master station, connect with the driver through this interface.

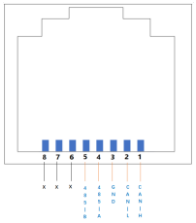
Figure	Pin No.&Name	Description
CN2 	PIN1:+5V	UART +5V output terminal
	PIN2:GND	UART analog ground
	PIN3:RX	UART data receiving terminal
	PIN4:TX	UART data transmission terminal

4.1.7 Terminal Resistor Switch (S2)

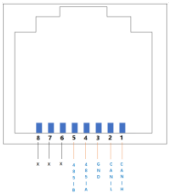
Figure	Pin No.&Name	Description
S2 	SW1	Terminal resistor matching selection (485) OFF: Disconnect 120Ω terminal resistor ON: Connect 120Ω terminal resistor
	SW2	Terminal resistor matching selection (CAN) OFF: Disconnect 120Ω terminal resistor ON: Connect 120Ω terminal resistor

4.1.8 Communication Bus Interface (CN1A)

Mainly used for CAN communication and RS485 communication.

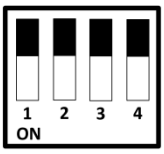
Figure	Pin No.&Name	Description
CN1A(IN) 	PIN1:CAN_H	CAN_H
	PIN2:CAN_L	CAN_L
	PIN3:CAN/485 GND	RS485/CAN Bus Common Ground
	PIN4:485-A	RS485_A
	PIN5:485-B	RS485_B
	PIN6-PIN8: Reserved	

4.1.9 Communication Bus Interface (CN1B)

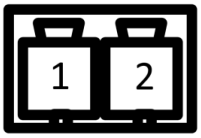
Figure	Pin No.&Name	Description
CN1B(OUT) 	PIN1:CAN_H	CAN_H
	PIN2:CAN_L	CAN_L
	PIN3:CAN/485 GND	RS485/CAN Bus Common Ground

	PIN4:485-A	RS485_A
	PIN5:485-B	RS485_B
	PIN6-PIN8: Reserved	

4.1.10 Station No. DIP Switch (S1)

<table>
<tr><th>Figure</th><th colspan="2">Pin No.&Name</th><th>Description</th></tr>
<tr><td rowspan="4">S1
 
</td><td>SW1</td><td rowspan="4">Note:
1.When the station No. is greater than 15, use internal storage to set the station No..
2.The state of this dip switch is only recognized once at power-on.</td><td rowspan="4">The station No. set by the dip switch applies to all communication methods.
<table><tr><th>SW4</th><th>SW3</th><th>SW2</th><th>SW1</th><th>Station No.</th></tr><tr><td>OFF</td><td>OFF</td><td>OFF</td><td>OFF</td><td>Internal</td></tr><tr><td>OFF</td><td>OFF</td><td>OFF</td><td>ON</td><td>1</td></tr><tr><td>OFF</td><td>OFF</td><td>ON</td><td>OFF</td><td>2</td></tr><tr><td>OFF</td><td>OFF</td><td>ON</td><td>ON</td><td>3</td></tr><tr><td>OFF</td><td>ON</td><td>OFF</td><td>OFF</td><td>4</td></tr><tr><td>OFF</td><td>ON</td><td>OFF</td><td>ON</td><td>5</td></tr><tr><td>OFF</td><td>ON</td><td>ON</td><td>OFF</td><td>6</td></tr><tr><td>OFF</td><td>ON</td><td>ON</td><td>ON</td><td>7</td></tr><tr><td>ON</td><td>OFF</td><td>OFF</td><td>OFF</td><td>8</td></tr><tr><td>ON</td><td>OFF</td><td>OFF</td><td>ON</td><td>9</td></tr><tr><td>ON</td><td>OFF</td><td>ON</td><td>OFF</td><td>10</td></tr><tr><td>ON</td><td>OFF</td><td>ON</td><td>ON</td><td>11</td></tr><tr><td>ON</td><td>ON</td><td>OFF</td><td>OFF</td><td>12</td></tr><tr><td>ON</td><td>ON</td><td>OFF</td><td>ON</td><td>13</td></tr><tr><td>ON</td><td>ON</td><td>ON</td><td>OFF</td><td>14</td></tr><tr><td>ON</td><td>ON</td><td>ON</td><td>ON</td><td>15</td></tr></table></td></tr>
<tr><td>SW2</td></tr>
<tr><td>SW3</td></tr>
<tr><td>SW4</td></tr>
</table>

4.1.11 Logic Power Interface (CN8)

Figure	Pin No.&Name	Description
CN8  	PIN1:+	Logic power input terminal Input voltage: 24V Maximum input current: 1A
	PIN2:-	

4.2 External Wiring Diagram

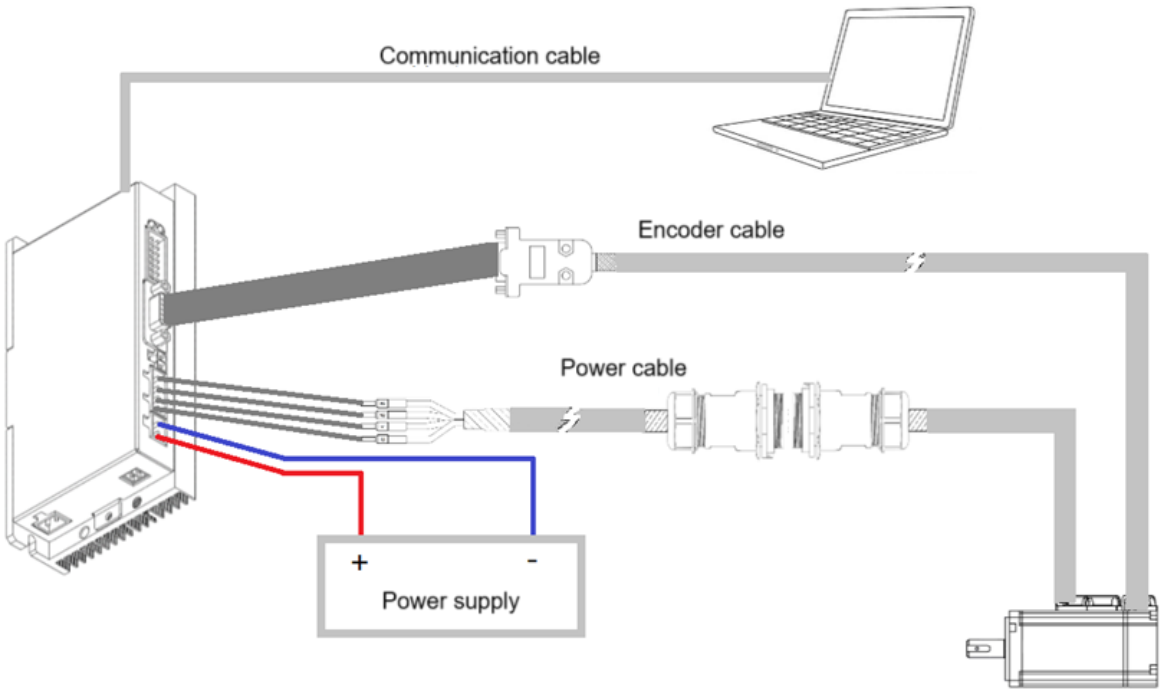


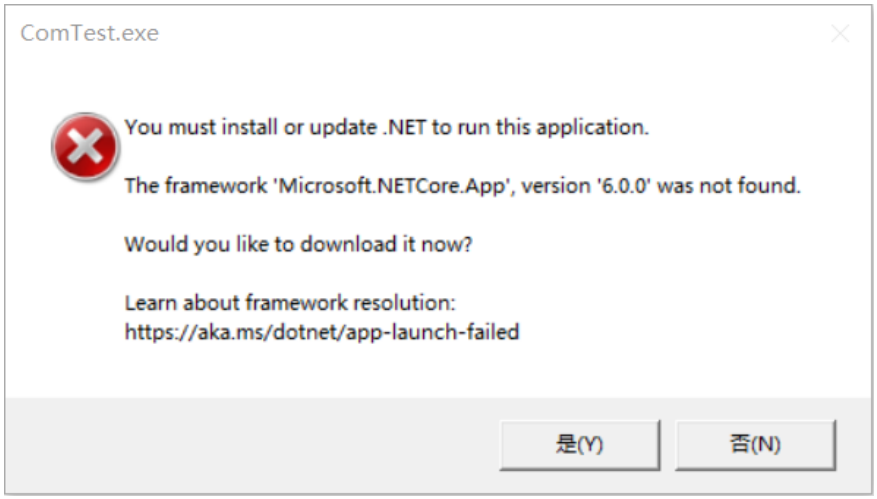
Figure 4-6 External Wiring Diagram

Chapter 5: FULLING-tech Software User Guide

Read before use (The software connects to PC through USB-UART convert cable)

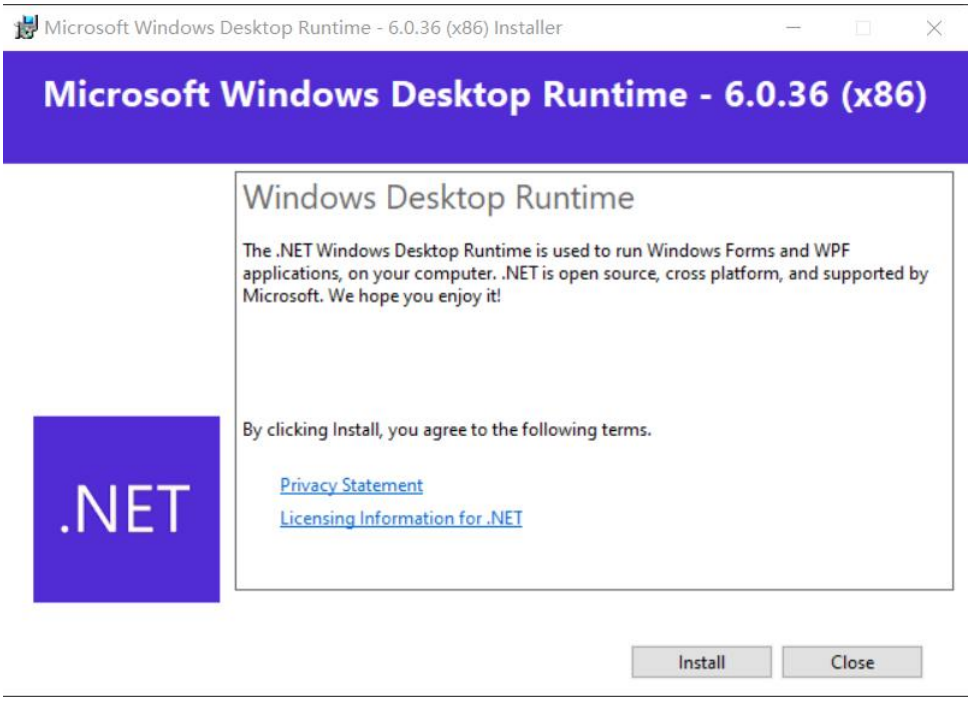
1. Plugin Issues:

When some computers open the FULLINGTech software, the following window may appear.



This is because the computer is missing the plugin required for the software to run. Here, we click "Yes", and the page will automatically jump to download the plugin (requires an internet connection).

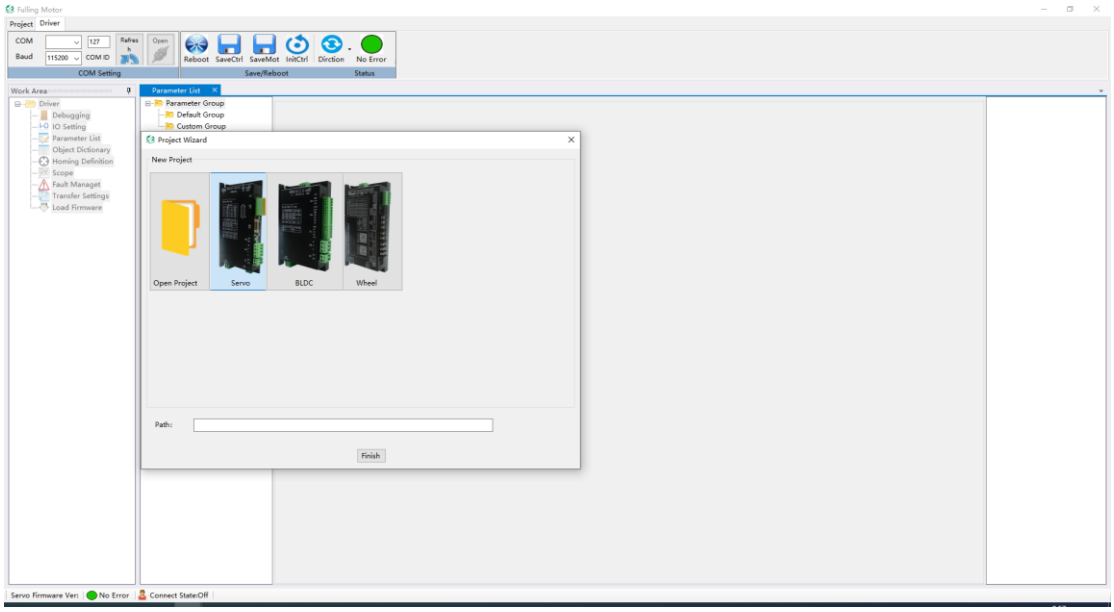
After downloading is completed, open the installation package, click "Install", and after the installation is complete, click "Close".



Then you can normally open the FULLINGTech software.

2. Basic Page Introduction:

After opening the software, the following page will pop up



According to the driver used, select the corresponding series to enter the software. Here we choose the servo series.

After entering, the interface is divided into five areas:

1.Serial port settings: It is used for the connection between the computer and the driver.

2.Motor control: It is used for the management of motor-related parameters.

3.Status: It is used to display the status of the driver and the motor.

4.Work area: It is used for users select different objects in the work area for corresponding operations.

5.Popup window: After clicking on an object in the work area, the corresponding window will pop up. After logging into the software, the parameter list in the work area is shown by default.

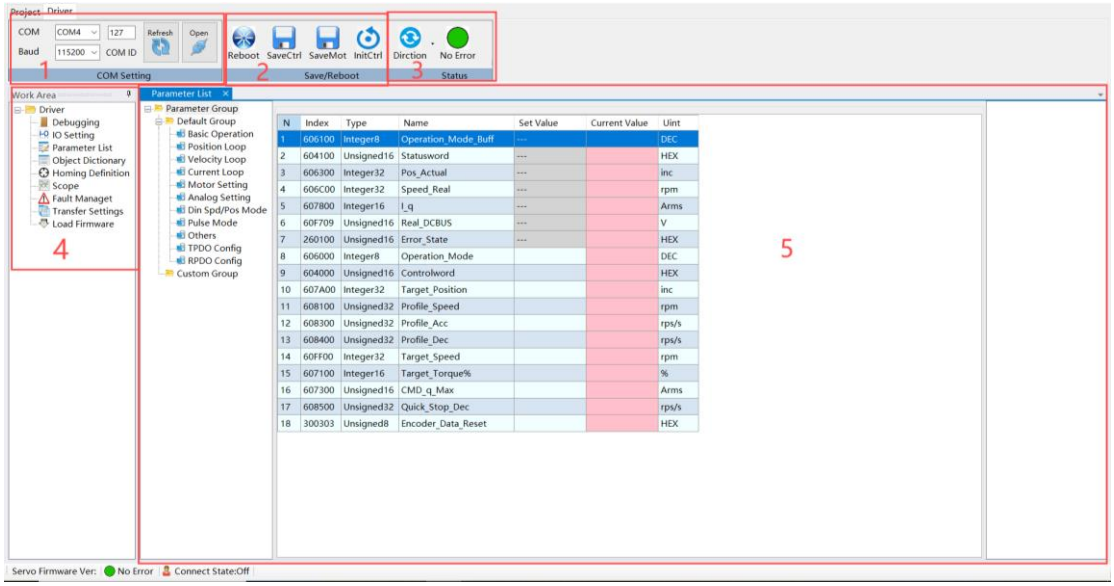


Figure 5-1 Software Basic Page

Regarding the addition of new lines in the popup window:

All popup window have default parameters. If you need to add some parameters, you can right click in the blank area of the popup window, then click "Add", and the following page will appear.

In the search box, enter the parameter name -> click search -> click >> -> click "OK".

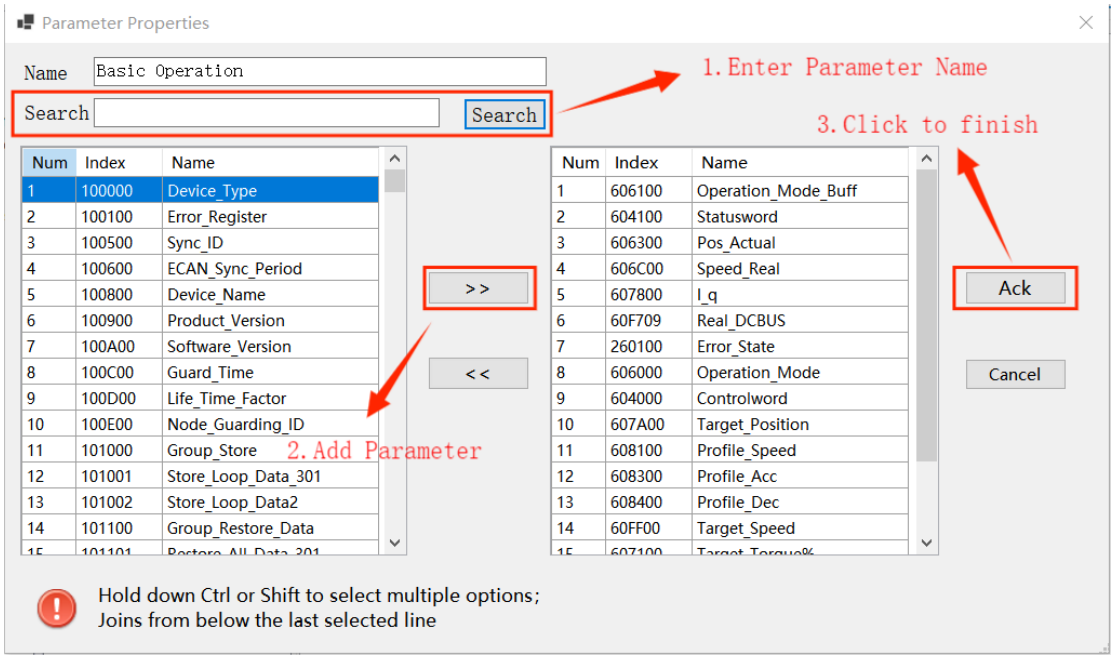


Figure 5-2 Add Parameter

3. Use of the Object Dictionary

In the work area, you can find the "Object Dictionary". Click on "Object Dictionary" and the following popup window will appear.

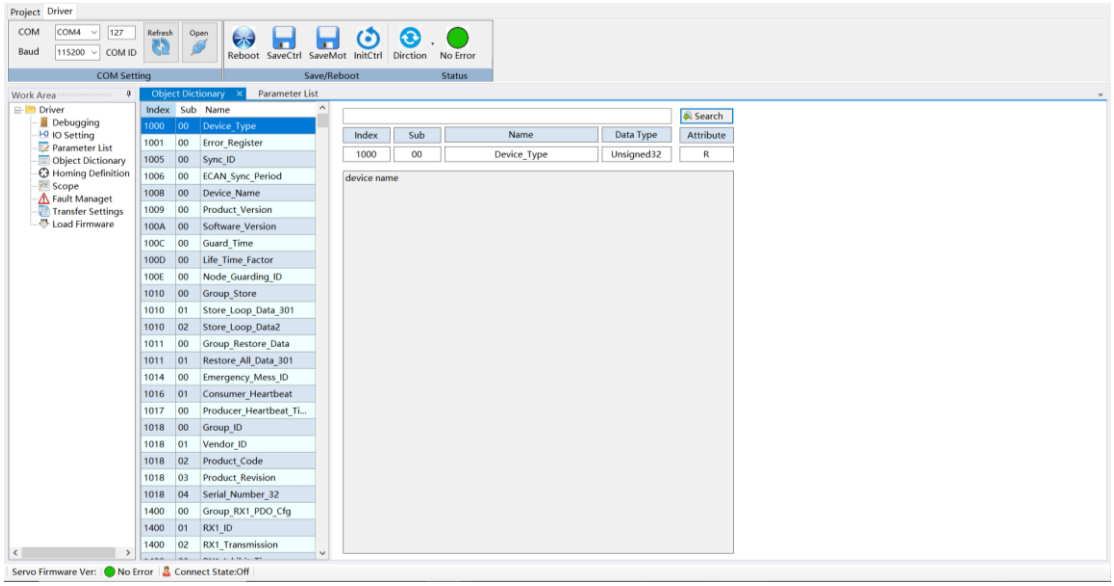


Figure 5-3 Object Dictionary Page

Through the search bar search, we can quickly find information related to parameters, such as index, data type, etc.

5.1 Connection between Driver and Software

5.1.1 Engineering File Management

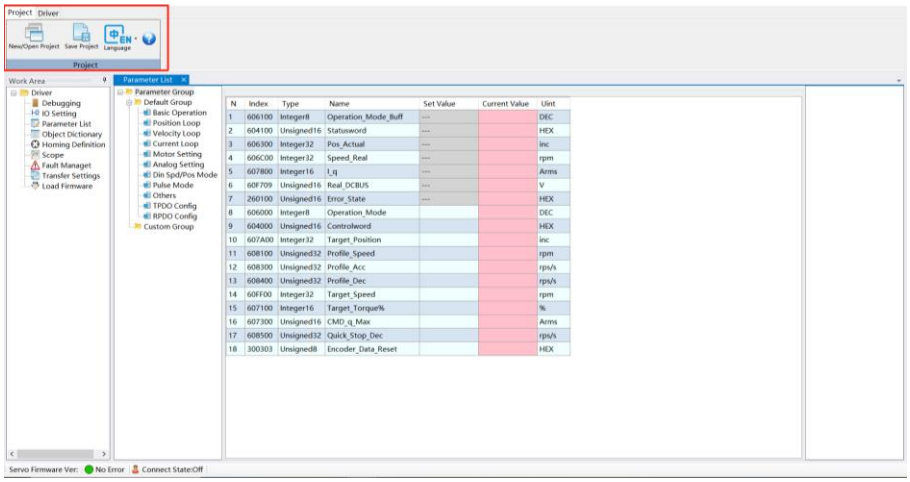
Engineering files are used to save the configuration and parameter settings. By saving an engineering file, users can quickly load these settings in subsequent uses, avoiding repetitive settings, saving time and effort, and easily switching to different scenarios for experiments and tests in different working environments.

Specific usage method is as follows:

Open the master station and click on the "Engineering" in the upper left corner, and the following menu bar will appear.

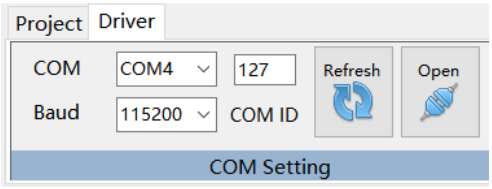
"New/Open Engineering" is used to create a new engineering or open an existing engineering file .

"Save Engineering" is used to save the current engineering as a .kpjt format file.



5.1.2 Driver Connection

Click on the "Driver" in the upper left corner, and the following menu bar will appear



Select the corresponding port number, baud rate, and device ID, and click "Open".

The device ID of any communication method is determined by the DIP switch, for details, please refer to “Chapter 4 System Interface and Wiring”. (The driver's universal ID is 127, that is, any communication method can set the device ID to 127 to connect to the driver)

Click on the parameter list -> basic parameters, when the actual current, status word, etc. appear values,then it indicates successful connection.

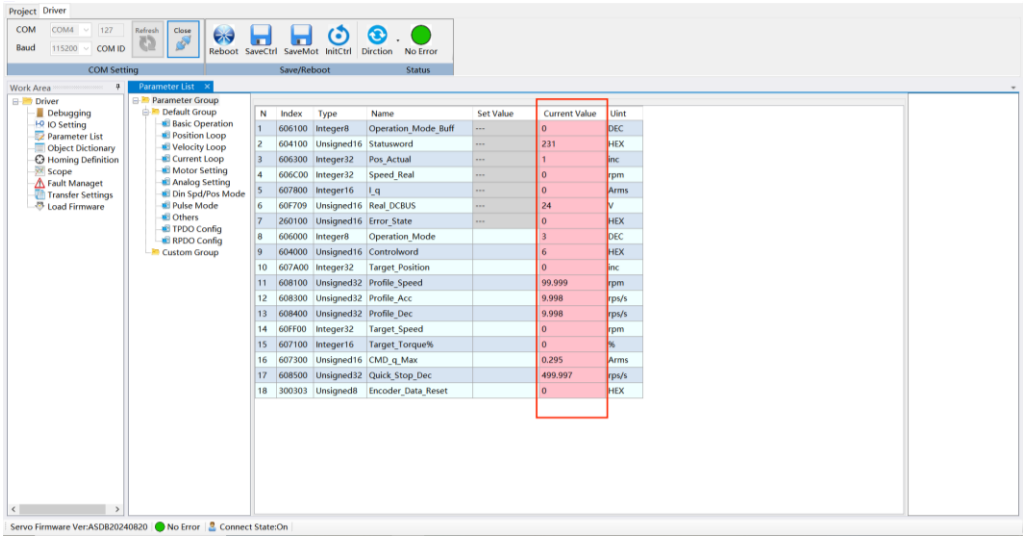


Figure 5-4 Basic Parameters

Note:

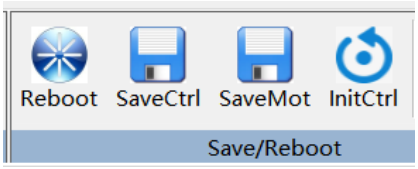
After setting the device ID and baudrate, you need to store the control parameters and restart the driver for them to take effect.

5.2 Motor Parameter and Control Parameter Management

5.2.1 Basic Motor Configuration

Motor configuration: through "Parameter List" -> "Motor Configuration"

After the driver is connected, you can manage motor parameters and control parameters through the following four buttons.



After completing the motor related parameter settings, click in sequence:

SaveMot -> Reboot -> InitCtrl -> SaveCrtl

Note:

Motor initialization only needs to be done once, and unless there are special reasons such as changing the motor or driver, there is no need to configure it again.

5.2.2 Specific Operations for Driver Matching Motor：

(1) Communication type

For FULLING' communication encoder motor, you can enter "????" in the motor model, then click "SaveMot", and “Reboot” the driver, it will automatically read the motor type and parameters in the encoder.

(2) Incremental type

For incremental encoder motors or third-party motor. Please refer to “Appendix II: Configuring Third-Party Motors”.

(3) **Feedback Type**

In the "Motor Configuration" section, there is a parameter called "Feedback Type." This should be filled out based on the type of encoder feedback signal. Different signal types can be referred to in the table below to enter the corresponding values.

Feedback Type / Bit / Encoder Signal	bit3：Communication Encoder	bit2：UVW TTL output	bit1：UVW wiring detection（Differential signal output by default）	bit0：ABZ wiring detection（Differential signal output by default）	Input value（HEX）
Differential signal ABZ Without UVW	0	0	0	1	1
Differential singal ABZ Differential singal UVW	0	0	1	1	3
Differential singal ABZ TTL signal UVW	0	1	0	1	5
Communication encoder	1	0	0	0	8

5.2.3 Absolute Encoder Related Parameters

(1) Absolute encoder reset command

18	300303	Unsigned8	Encoder_Data_Reset		0	HEX

0xBA: Reset fault
0x62: Clear multi-turn data
0xC2: Clear single-turn data

(2) ALMC（Encoder fault information）

27	300302	Unsigned8	ALMC	---	0	HEX

- bit0: Speed exception
- bit2: Single-turn information calculation fault
- bit5: Multi-turn data loss
- bit6: Battery low-voltage fault
- bit7: Battery low-voltage warning

(3) Encoder data

23	300304	Integer32	AbsE_Counter	---	0	DEC
24	300305	Integer32	AbsE_Multi	---	0	DEC
25	300306	Integer32	AbsE_Position	---	0	DEC

Absolute encoder position = (multi-turn data × feedback precision) + single-turn data

5.3 Simple Debugging

Simple debugging refers to the simple and easy-to-operate debugging functions provided in the master station software to help users quickly check the working status and performance of the equipment or system. This kind of debugging function usually does not require complex parameter settings or professional knowledge and is suitable for quick problem diagnosis and basic performance testing.

Simple debugging procedure is as follows:

Click on "Simple Debugging" in the work area, and the following page will pop up.

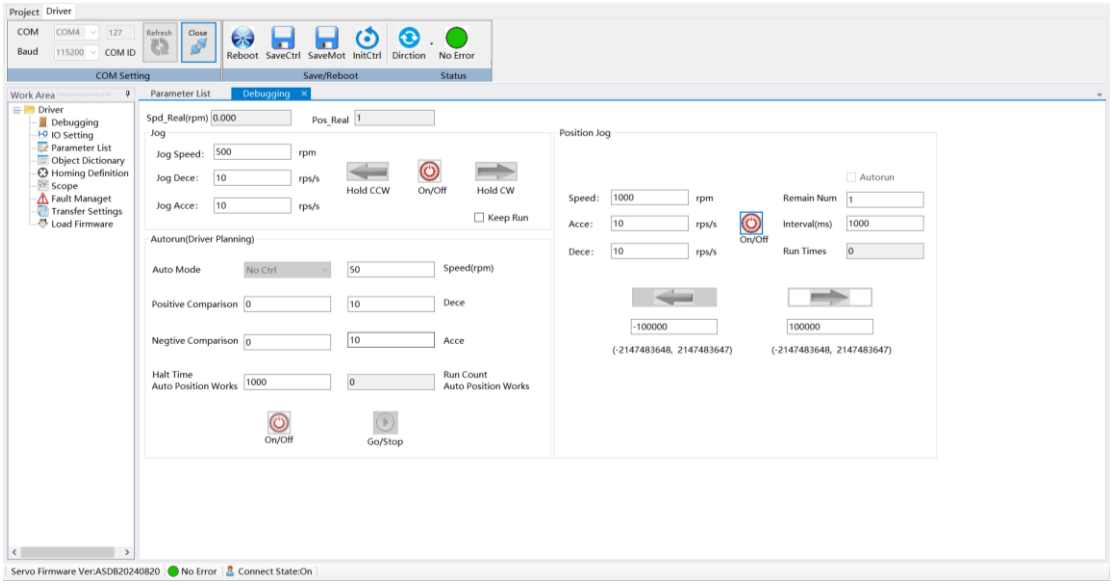


Figure 5-6 Simple Debugging Page

This page is divided into three parts:

(1) **Jog**: In the speed jogging mode, users can set the target speed and acceleration of the device, which will smoothly accelerate and decelerate according to these settings, and continue to move after reaching the target speed until the stop command is sent or the device stops.

After setting the corresponding jogging speed, jogging acceleration, and jogging deceleration, click "on". When the "on" icon changes to , it indicates that the driver is enabled and the motor is locked. After the motor is locked, long press "Long press positive rotation" or "Long press reverse rotation" to make the motor rotate. We can also click "Full speed operation", check this option, and then click "Long press positive rotation" or "Long press reverse rotation" to make the motor rotate.

(2) **Position jog**:Users can enter the target position coordinate information, and the device will move smoothly to the target position according to these coordinate information, set the speed and acceleration.

After setting the corresponding speed, acceleration, and deceleration, and then setting the corresponding position that the motor needs to rotate to, click "on". When the "on" icon changes to , it indicates that the driver is enabled and the motor is locked. Click to make the motor rotate; set the remaining number of runs and interval time after checking the automatic operation, and the motor will automatically move between the two positions.

(3) **Autorun(Drivier Planning)**: The automatic operation mode can choose not to control, position control, speed control, time control, and automatic position control. The concepts of speed control and position control are the same as the above speed jogging and position jogging, which will not be explained here. The following are explanations for time control and automatic position control.

Time control: You can set the positive and negative rotation times, and the motor will keep rotating positively and negatively.

As shown in the figure, through time control, set the positive rotation time to 2000ms, the negative rotation time to 2000ms, the running speed to 1000rpm, and the acceleration and deceleration to 100rps/s.

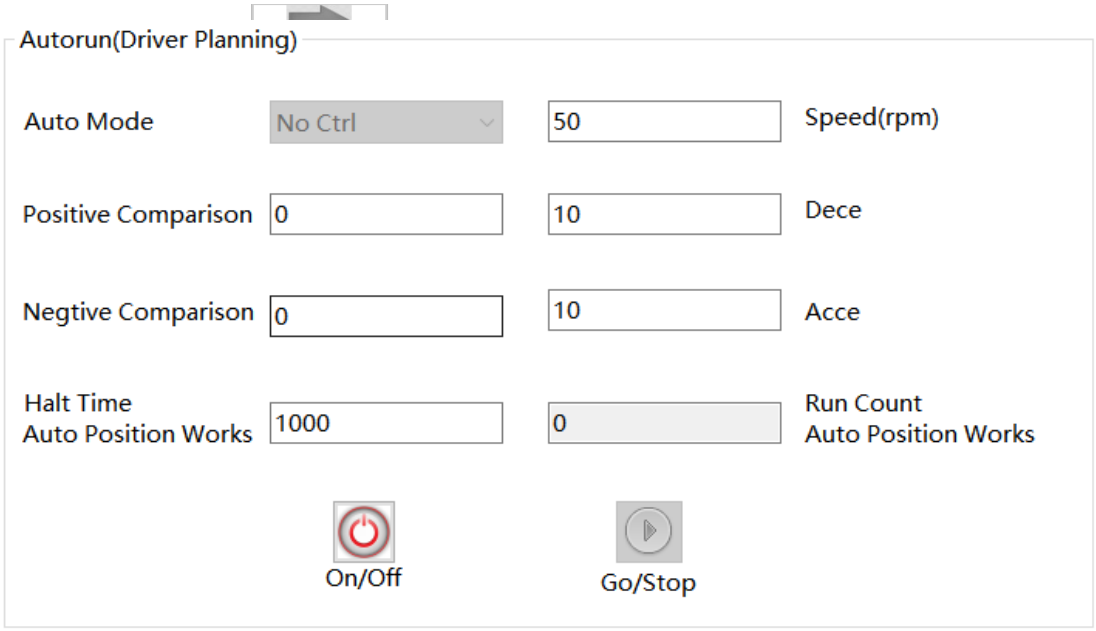


Figure 5-7 Automatic Operation

Automatic position control: By setting the automatic operation positive comparison point and the automatic operation negative comparison point, the motor can run back and forth between these two positions, and set the pause time to set the time to stay at a certain position. (This mode is similar to checking the automatic operation in the point position mode)

Note:

If you need to switch modes or modify parameters, you need to click pause first, modify the parameters and mode after that, and then click continue.

5.4 IO Port Settings

Click “IO settings” in the work area to enter the digital input and digital output modules.

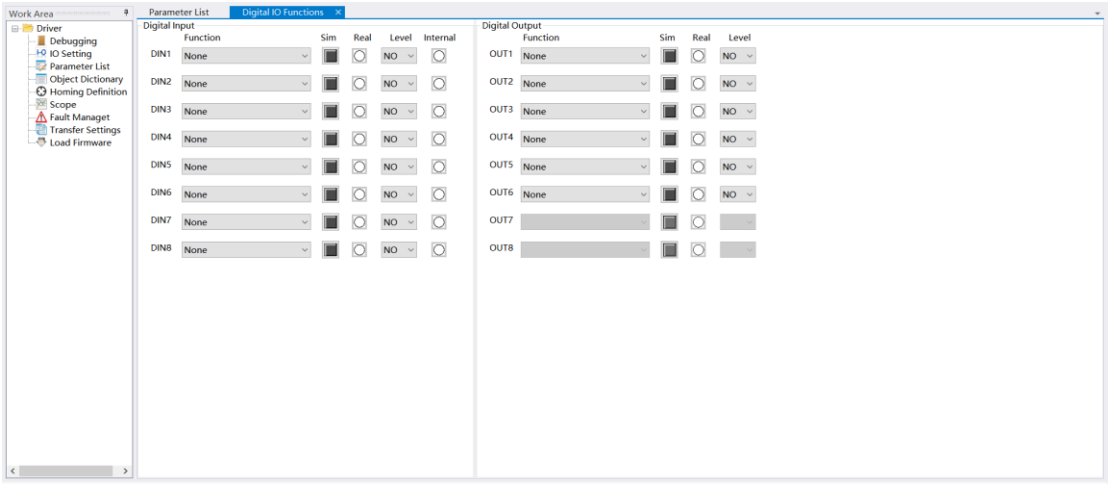


Figure 5-8 IO Port Settings

5.4.1 Input Mode

Function: Click None to select the function.

Simulation: Indicates the simulation power situation. Power-on is , power-off is .

Actual input: Indicates the actual digital input situation.

Electrical Level:

(1) Normally open mode - High level is conductive, low level is not conductive.

(2) Normally closed mode - High level is not conductive, low level is conductive.

Effective input: The digital logic 0 is , the digital logic 1 is , and it is jointly determined by the simulation (or actual input) and the level property.

Input example:

At this time, it indicates that DIN1 is set as the driver enable function, and it also indicates that the driver is enabled.

Table 5-1 Input Function Module List and Description

Function name	Description
Emergency stop	Used to stop the system in emergency situations.
Driver enable	Used to enable the driver. If IO port is used for enabling, then the status word cannot be written.

	1: Write 0x2F to 604000 0: Write 0x06 to 604000
Alarm reset	Used to clear alarm
Pre-enable	Used to determine if the entire system is ready. 1: The driver can be enabled. 0: The driver cannot be enabled.
Kvi close	Close the integral gain in speed loop.
Positive limit	Used for position limit
Negative limit	
Negative limit	Home switch signal, can only be used to find the home position
Speed command reverse	In speed and torque modes, the speed command can be reversed
Multi-speed control 0	Used for Din speed mode under Din speed index
Multi-speed control 1	
Multi-speed control 2	
External input fault	External input fault, such as: external temperature control switch, when the temperature exceeds the limit, the input IO signal causes the driver to stop
Homing	Used for searching the home signal
Multi-position control 0	In the DIN position mode, used to select the multi-position through BCD code combination, such as multi-position control 2, multi-position control 1, and multi-position control 0 are 011, which means selecting the multi-position 3. (All multi-positions represent absolute positions)
Multi-position control 1	
Multi-position control 2	
Electronic gear selection 0	In the DIN position mode, used to select the electronic gear through BCD code combination, such as electronic gear selection 2, electronic gear selection 1, and electronic gear selection 0 are 001, which means selecting the electronic gear 1
Electronic gear selection 1	
Electronic gear selection 2	

Note: When using multi-position, the multi-speed should be given a value. Multi-speed 0 indicates the speed from the current position to the multi-position control 0.

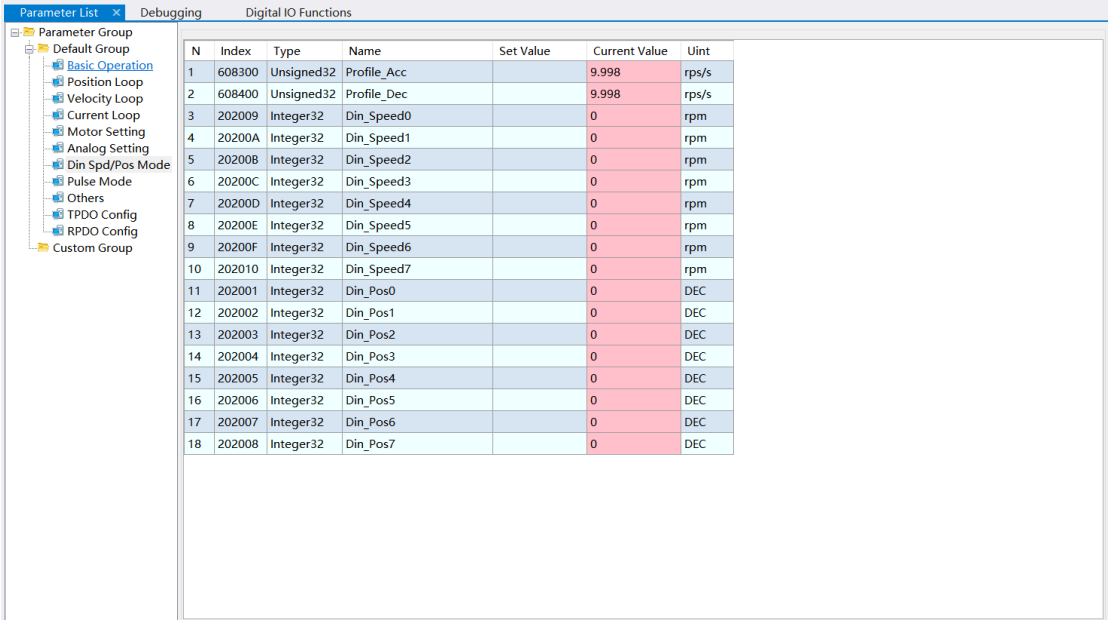


Figure 5-9 Multi-speed control

5.4.2 Output Mode

Function：Click 

to select the function.

Simulation：  Indicates the simulation power-on situation. Power-on is  , power-off is  .

Actual output: Indicates the actual digital output situation.

Electrical Level: (1) Normally open mode - High level is conductive, low level is disconnected.

(2) Normally closed mode - High level is disconnected, low level is conductive.

Effective output: Jointly determined by the simulation (or actual output) and electrical level property. Indicates the digital logic 0 is  , the digital logic 1 is  .

Table 5-2 Output Function Module List and Description

Function name	Description
Driver ready	The driver is ready and can be enabled
Driver alarm	The driver has an error
Motor position reached	The position is less than the position reach window
Motor zero speed	The speed is less than the speed reach window
Motor brake effective	Indicates that the brake is effective
Motor speed reached	The motor has reached the speed
Index signal appears	The index signal appears
Speed limit reach	In the torque mode, the actual speed reaches the maximum speed limit

Motor locked axis	The driver is enabled, and the motor is locked
Position limit	Position limit switch has been triggered
Home found	In the home mode, the home position has been found
Target torque reached	The target torque has been reached

5.5 Trigger-based Oscilloscope

A trigger-based oscilloscope is a special type of oscilloscope used to capture and display specific events or signals in a waveform diagram. It controls the timing of the oscilloscope's data acquisition through a trigger signal to ensure that the waveform is only displayed under specific conditions, making it easier to observe and analyze.

Click on "Oscilloscope" in the work area to pop up the following page.

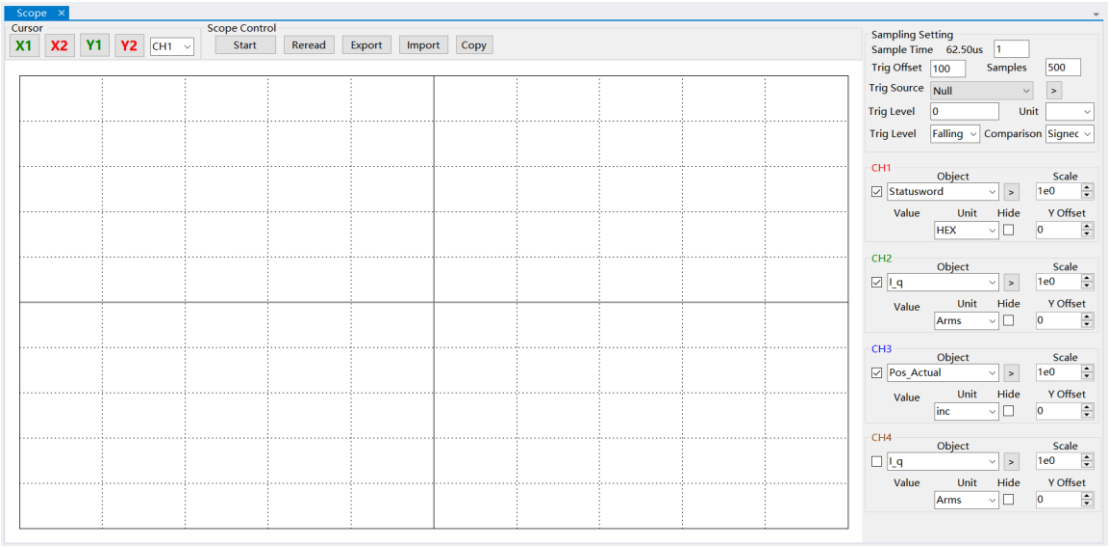


Figure 5-10 Oscilloscope Page

Below are some explanations of the oscilloscope-related parameters:

Sampling period: The period for collecting data, set to 1 indicates that data is collected every 62.5us.

Number of acquisitions: Indicates the total number of data collected in this sampling, set to 500 indicates that 500 pieces of data are collected.

Pre-trigger data points: The number of samples taken before the trigger source is triggered, set to 100 indicates that 100 samples are taken before the trigger.

Trigger source and trigger level: You can set when to start sampling through this setting, and the condition is set by yourself.

Trigger edge: Click to change to rising edge trigger, falling edge trigger, or edge trigger.

Object: The sum of the data lengths of the four objects being sampled at the same time is a maximum of 64 bits, for example, 2 32-bit objects, or 4 16-bit objects.

Cursor: By clicking the button, you can select the corresponding cursor, which will be displayed on the oscilloscope and select the channel you need to observe on the right side of the cursor.

Copy: Copy the sampled data to the clipboard.

Export: Export the sampled data to a .scope file.

Import: Import the .scope file and display the waveform.

Re-read data: Read out and display the waveform of the recently collected data from the driver.

If the required parameters are not in the default object list of the channel, you can click > to jump to the "Object Dictionary", enter the parameter name in the search box, and then double-click to add the parameter.

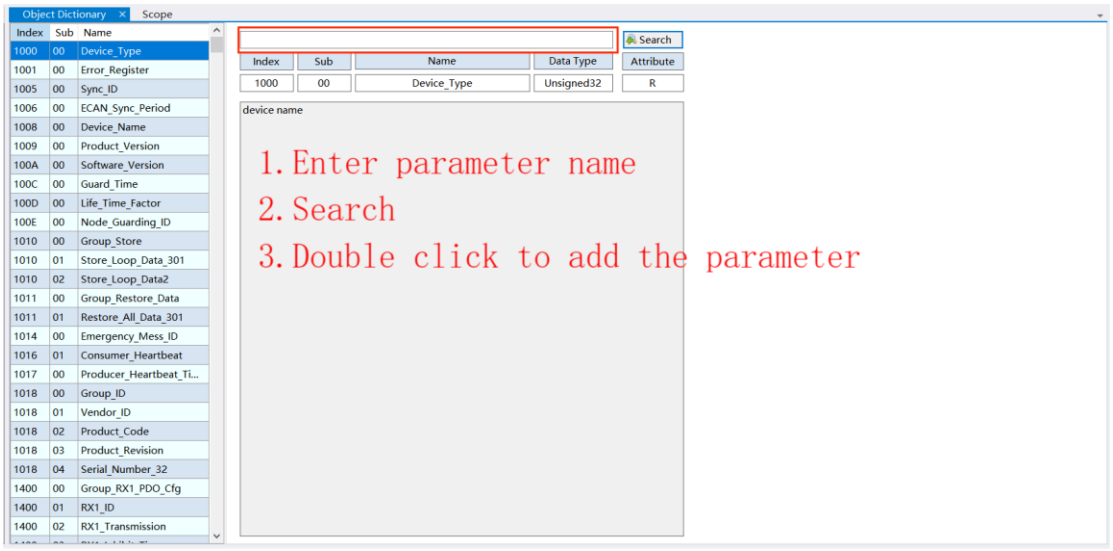


Figure 5-11 Adding Parameters

Below is an example of using the oscilloscope to read the actual position and actual current:

(1) First, set the working mode to "3" mode, and then set the contour acceleration and contour deceleration to 10, and set the target speed to 500. (This step is to have obvious parameters that can be read)

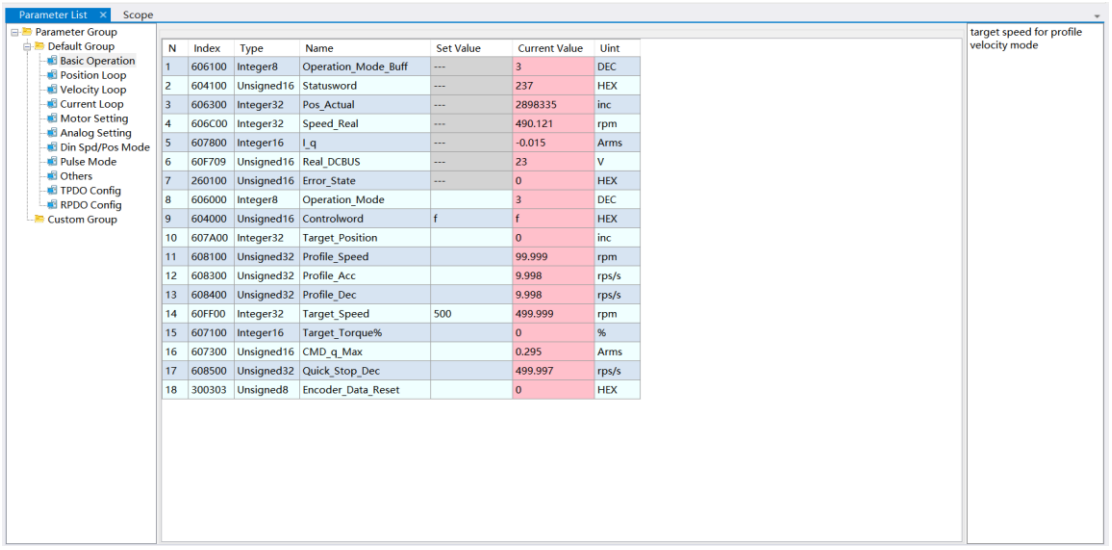


Figure 5-12 Modify Parameters

(2)Open the oscilloscope and follow these steps:

First step: On the right CH1, set the object to the actual position.

Second step: Set the trigger source to NULL, that is, click "Start Acquisition" to start collecting data immediately.

Third step: Click "Start Acquisition" and wait for the progress bar to complete, which means the collection is successful.

Fourth step: Switch the "actual position" in the first step to "actual current" and click "Start Acquisition" again to switch the image from the actual position image to the actual current image.

Note：

If you want to set the sampling trigger conditions, such as setting the trigger condition to when the effective target speed reaches 500rpm, you need to confirm with "Enter" on the keyboard just like modifying the control word.

Every time you switch an object or channel, you need to click "Start Acquisition" once.

We can zoom in on the image through the "scale" on the right side;

In addition, you can also zoom in on the waveform by pressing the middle mouse button for a long time.

Move the image up and down through "Y-axis offset";

Click "X1" and "Y1" near the cursor, and two vertical lines perpendicular to the X-axis and Y-axis will appear on the screen, and the coordinates of the intersection point of the two lines will be displayed in the upper right corner of the image. X2 and Y2 are used in the same way. dX and dY represent the difference between X1 and X2, Y1 and Y2.

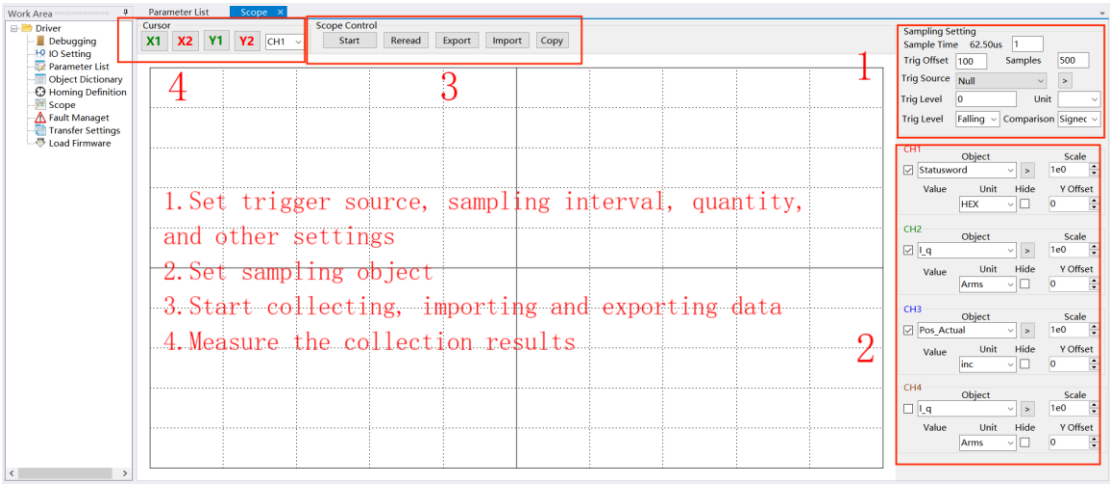


Figure 5-13 Steps to Read the Actual Position

The above lists the steps to read the actual position, and the method to read parameters such as actual speed is basically the same.

The above lists the steps to read the actual position, and the method to read parameters such as actual speed is basically the same.

Click "Export" to generate a .wave file 

at the specified path (customized),

Click "Import" again to display this waveform on the oscilloscope.

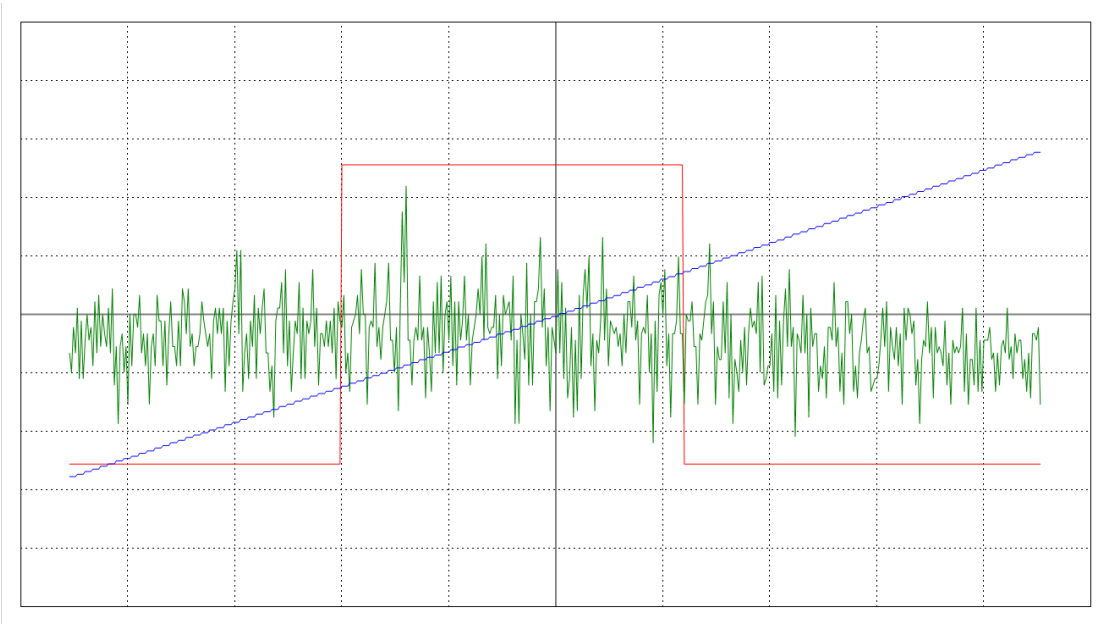


Figure 5-14 The waveform after importing the test.wave file

In addition, we can also click "Copy" to copy the data to the Excle table.

5.6 Historical Errors and Alarm

When the icon 

of the master station changes to 

, it indicates that a fault has occurred.

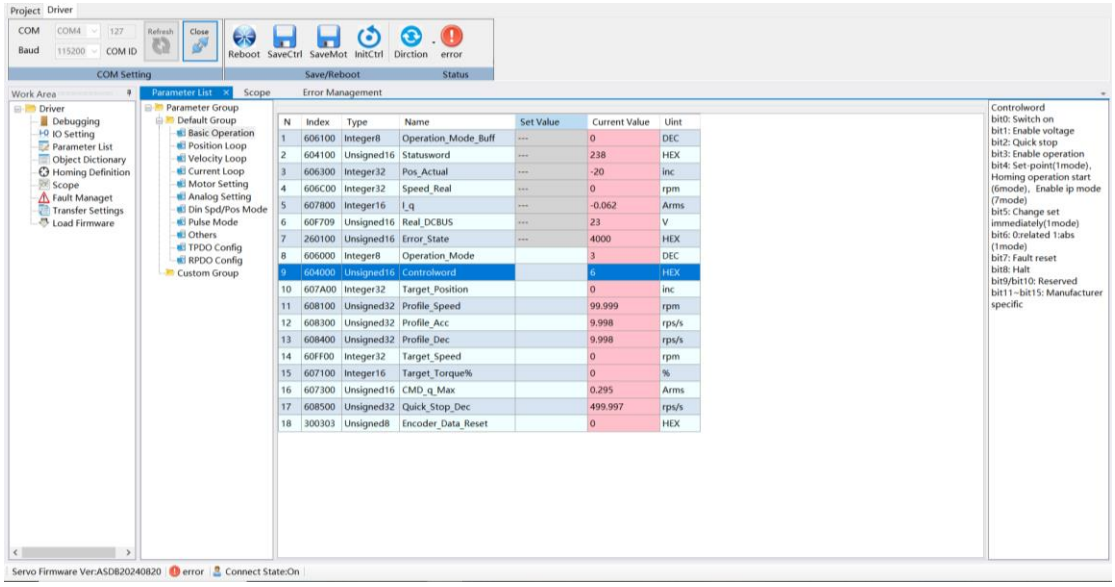


Figure 5-15 Fault Example

Click 

directly to jump to fault management, or click on fault management in the work area.

The fault management page is as follows:

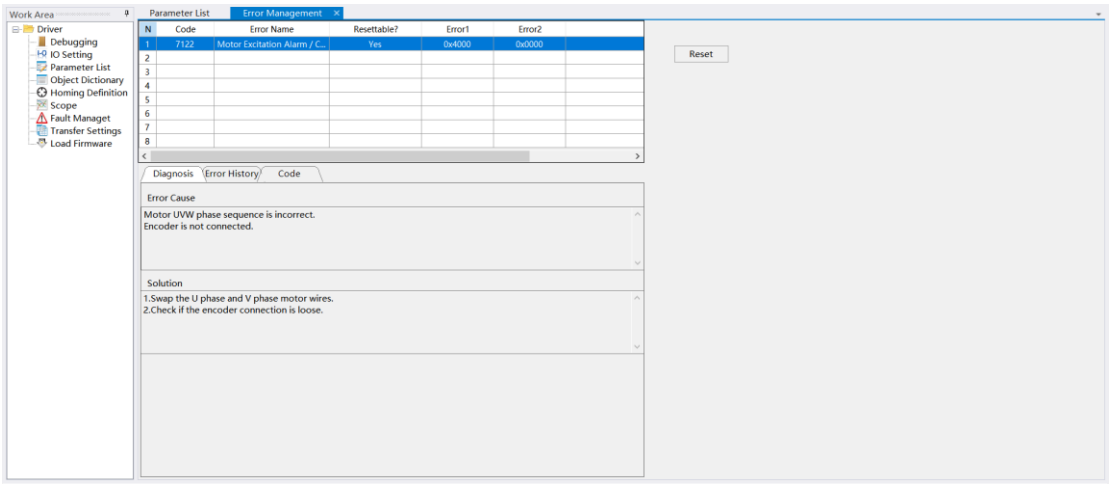


Figure 5-16 Fault Management

Through the master station page, we can directly obtain the cause of the fault and the solution. In addition, by clicking on the historical fault, you can query the last 8 error messages, including fault code, bus voltage, speed, current, temperature, working mode, time, and PWM status. The first line is the latest error information, and so on. The following is a common error code table. If the problem cannot be solved, please refer to Chapter 8 Alarm Elimination for solutions.

Table 5-3 Error Code Table

Error code	Error name	Error description
0x1001	MCU internal error	Detected MCU model error
0x2214	Software overcurrent	Software overcurrent
0x2320	Overcurrent alarm	The driver power tube or motor is short-circuited
0x3130	Motor phase missing	One phase of the motor power lines UVW is not connected
0x3210	Over voltage alarm	The bus voltage is too high
0x3220	Low voltage alarm	The bus voltage is too low
0x4210	Temperature alarm	The radiator temperature is too high
0x4310	Motor temperature too high	Motor temperature sensor alarm
0x5112	Logic low voltage alarm	Logic power supply voltage is too low
0x5210	Current sensor	Current sensor signal offset or excessive ripple
0x5441	Negative limit error	Negative limit error only occurs when the limit function is defined as 1
0x5442	Positive limit error	Positive limit error only occurs when the limit function is defined as 1
0x5443	Pre-enable error	When the pre-enable input is defined, there is no valid input on the pre-enable input port when the driver is enabled or about to be enabled
0x5530	EEProm alarm,	Eeprom data verification error
0x6320	Motor configuration	There is no motor configuration information in the

	error	EEPROM/motor has never been configured
0x7110	Absorption resistor alarm	Braking resistor overload
0x7122	Motor excitation alarm	The motor UVW phase sequence is incorrect or the encoder is not connected
0x7305	Encoder count alarm	The encoder is disturbed
0x7306	Main encoder count error	Main encoder count error
0x7310	Speed deviation alarm	The set speed deviates too much from the actual speed
0x7382	Main encoder connection error	Main encoder connection error
0x7500	Bus offline erro	Abnormal bus communication
0x8611	Position error alarm	The actual following error exceeds the set maximum following error
0x8613	Home finding error	Problems or failures when performing home search or zeroing operations
0x8A81	Fully closed-loop encoder counting direction error	In the all closed-loop working state, the main encoder count direction is opposite to the motor encoder count direction
0xFF10	User Iit fault	Motor or driver power tube IIt fault
0xFF21	Reserved	
0xFF22	Pulse frequency too high	Pulse input frequency is too high
0xFF30	Encoder ABZ connection alarm	Encoder ABZ connection error or not connected
0xFF40	Encoder UVW connection alarm	Encoder UVW connection error or not connected
0xFF41	Reserved	
0xFF42	Reserved	

5.7 Driver Parameter Reading and Writing

Through driver parameter reading and writing, you can save the driver configuration in the master station or controller, which is convenient for subsequent device restoration and configuration. This is especially useful when replacing or maintaining equipment, as it allows you to quickly apply previous configurations to new equipment, saving debugging time and cost.

5.7.1 Read Setting

Click on "Transfer Setting" in the work area -> "Read Setting" -> "Open List" (select a file with a .oparam suffix) -> "Read" -> "Save to File".

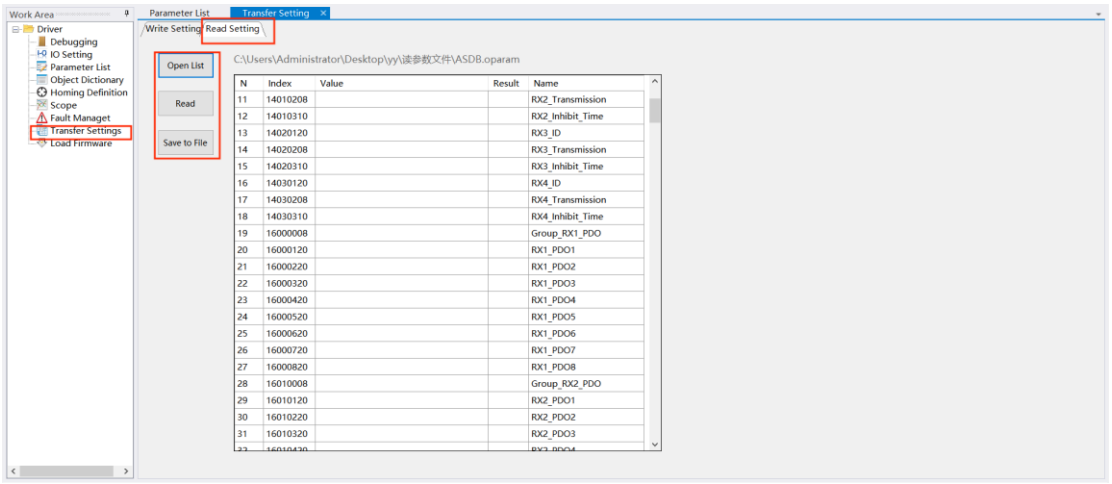


Figure 5-17 Read Driver Configuration

5.7.2 Write Driver Configuration

Click on "Transfer Setting" in the work area -> "Write Setting" -> "Open File" (select a file with a .iparam suffix) -> "Write" -> "Save in EEPROM"

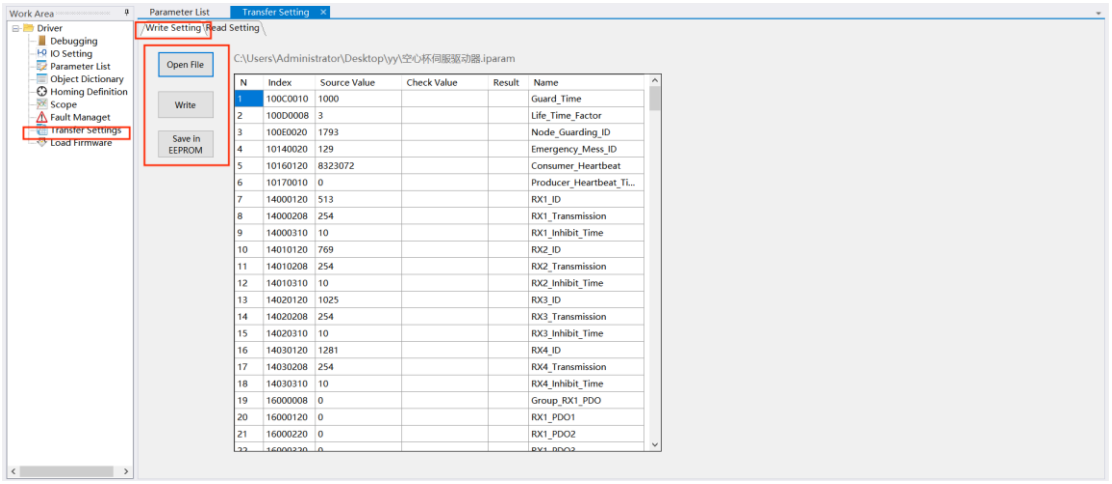


Figure 5-18 Write Driver Configuration

5.8 Load Firmware

If for some special reasons, you need to re-flash the firmware, you can follow these steps: Click on "Load Firmware" → select the file → click on "Download"

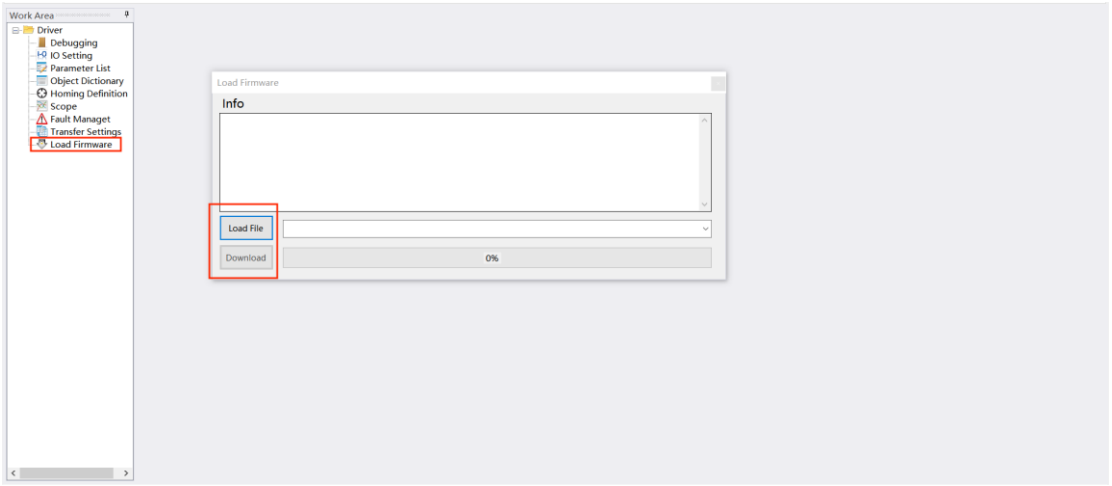


Figure 5-19 Firmware Download

Chapter 6: Operation Mode Introduction

6.1 Speed Mode Introduction

The speed mode is include immediate speed mode (-3) and speed mode with acceleration (3). The speed in the speed mode can be set by internal commands, external IO ports (DIN speed mode), and three types of analog inputs.

6.1.1 Internal Commands Setting Speed (3)

Table 6-1 Speed Mode Related Parameters

Index	Data Type	Name	Description	Setting Value
606000	Integer8	Operation_mode	Used to set the motor operation mode 3 :Speed mode with acceleration	3
604000	Unsigned16	Controlword	Used to set the motor state F (Motor shaft lock) 6 (Motor shaft release)	F or 6
60FF00	Integer32	Target_speed	The preset speed value set by the user	Set by the user
608300	Unsigned32	Profile_acc	The acceleration set in the speed mode with acceleration (3)	Set by the user
608400	Unsigned32	Profile_dec	The deceleration set in the speed mode with acceleration (3)	Set by the user

Note:

After entering the data, press the "Enter" key, and then check whether the current value has been modified.

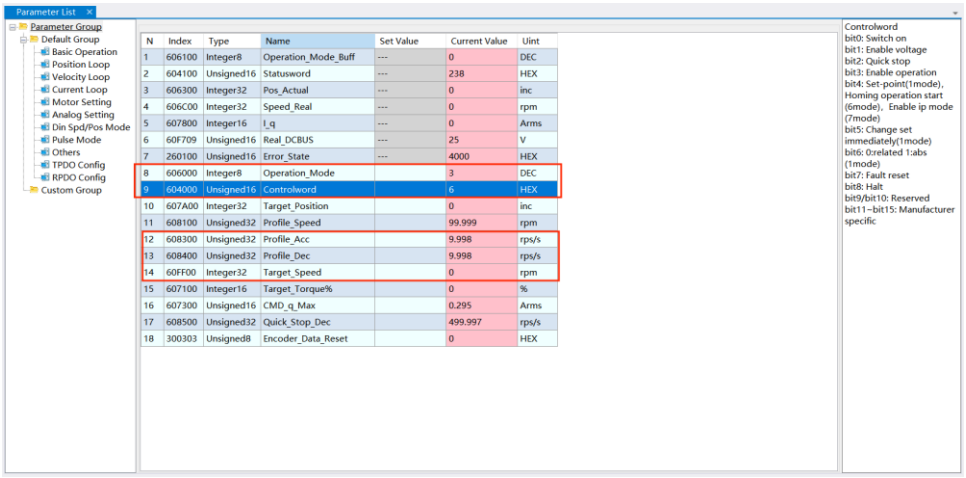


Figure 6-1 FULLING software Modify Speed

6.1.2 Internal Commands Setting Speed (-3)

Table 6-2 Immediate Speed Mode Related Parameters

Index	Data Type	Name	Description	Setting Value
606000	Integer8	Operation_mode	Used to set the motor operation mode -3: Immediate speed mode	3
604000	Unsigned16	Controlword	Used to set the motor state F (Motor shaft lock) 6 (Motor shaft release)	F or 6
60FF00	Integer32	Target_speed	The preset speed value set by the user	Set by the user

6.1.3 Setting Speed with External IO Ports (DIN Speed Mode)

Table 6-3 DIN Speed Mode Setting

Index	Data Type	Name	Description	Setting Value	Unit
202009	Integer32	Din_speed0	Controlled by the Din Vel index 2, Din Vel index 1, and Din Vel index 0 set by the DIN port. If the digital logic input of multi-speed control 2, 1, and 0 is 010, it means to use multi-speed control 2; if the input is 110, it means to use multi-speed control 6.	Set by the user	rpm
20200A	Integer32	Din_speed1			
20200B	Integer32	Din_speed2			
20200C	Integer32	Din_speed3			
20200D	Integer32	Din_speed4			
20200E	Integer32	Din_speed5			
20200F	Integer32	Din_speed6			
202010	Integer32	Din_speed7			
608300	Unsigned32	Profile_Acc	The acceleration set in the speed mode with acceleration (3)	Set by the user	rps/s
608400	Unsigned32	Profile_Dec	The deceleration set in the speed mode with acceleration (3)		
606000	Integer8	Operation_mode	Used to set the	3	DEC

			motor operation mode 3 (Speed mode with acceleration)		
604000	Unsigned16	Controlword	Used to set the motor state 6->2F Enable	6->2F	HEX

If you use FULLING software for debugging, you can follow these steps:

(1) Work area "Parameter List" → "Din Spd/Pos Mode" → Set the corresponding values

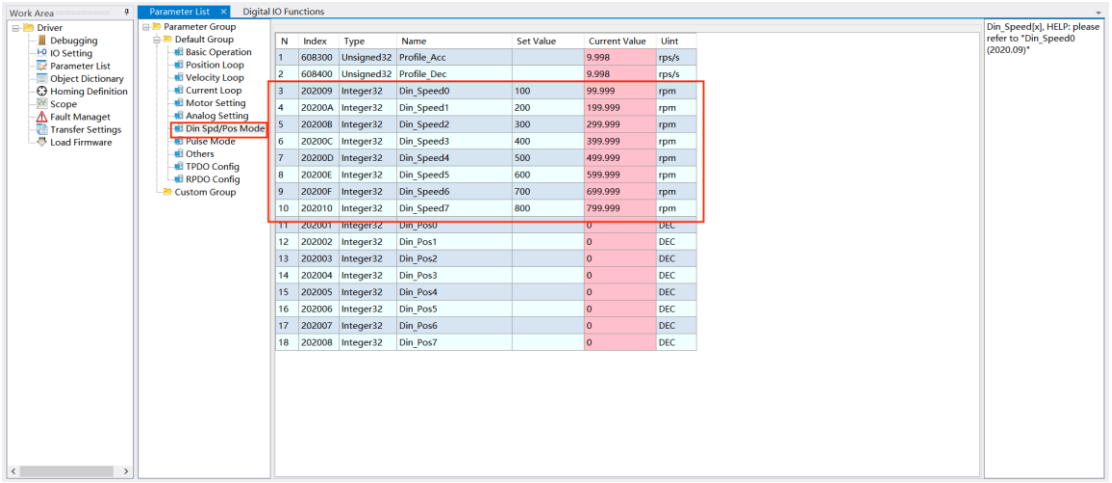


Figure 6-2 Multi-speed parameter settings

(2) Click on "IO Settings" -> Select the DIN port and set it to Din Vel

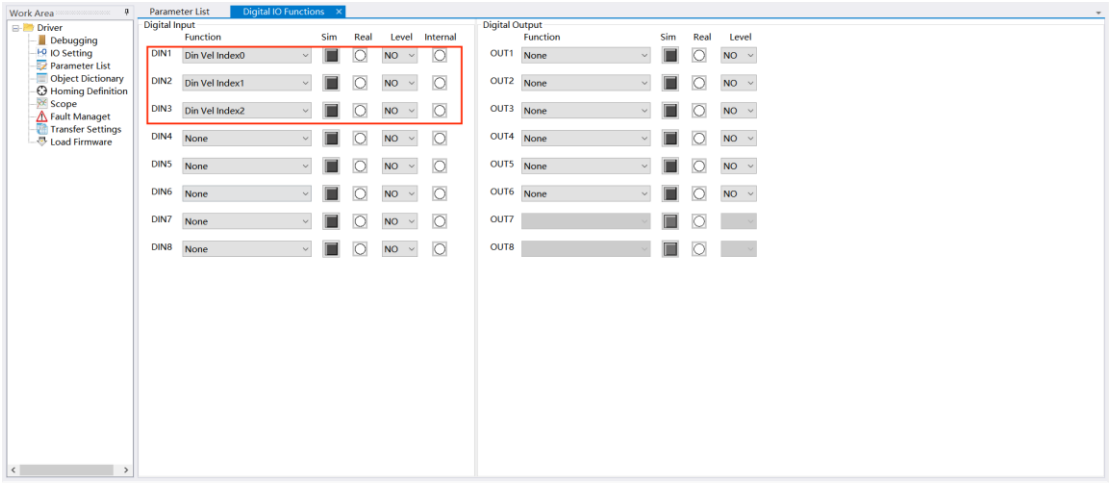


Figure 6-3 IO port settings

(3)Enter the "Basic Parameters" and set the operation mode to 3 and the control word to F, so we can control different speeds through the input of the IO port.

For example, set DIN1 to multi-speed control 0, DIN2 to multi-speed control 1, and DIN3 to multi-speed control 2. Assuming the input for DIN3-DIN2-DIN1 is 110, the motor will rotate at a speed of 700rpm (multi-speed control 6).

Note:

If you use the IO port to set the speed, then the input of the speed through other modes will be invalid.

6.1.4 Setting Speed with Analog Input

Table 6-4 Analog input setting speed related parameters

Index	Data Type	Name	Description	Setting Value	Unit
250106	Unsigned16	ADC_AI	Analog signal 1 original ADC data	Read Only	DEC
250206	Integer16	Ain_out	Analog Input signal1 (AIN1) input voltage after filter, deadband and offset	Read Only	V
250201	Integer16	Ain_Correction_ Gain	Analog input correction gain	User-de fined	DEC
250202	Integer16	Ain_Correction_ Offset	Analog input correction offset	User-de fined	DEC
250203	Unsigned8	Ain_Filter	Analog input signal 1 (AIN1) filter	User-de fined	DEC
250204	Integer16	Ain_Offset	Analog input signal 1 (AIN1) offset		V
250205	Integer16	Ain_Dead	Analog input signal 1 (AIN1) deadband		V
250207	Unsigned8	Ain_Speed_con	analog signal control speed, valid at operation mode 3 or -3 0: Invalid 1: Ain Control Speed 2: Ain Control Position	1	DEC
25020A	Integer32	Analog_speed_ factor	Proportion factor for converting the analog input signal to motor speed	User-de fined	rpm/v
606000	Integer8	Operation_mode	Used to set the motor operation mode 3 (Speed mode with acceleration) -3 (Immediate speed mode)	3 or -3	DEC
604000	Unsigned16	Controlword	Used to set the motor state F (Motor shaft lock) 6 (Motor shaft release)	F or 6	HEX

Assuming the input analog signal is Vin, after data processing, it is obtained as Vin_r, then

we have:
Vin_r= Vin - Analog input offset voltage
If the absolute value of Vin_r is greater than the dead zone (i.e., |Vin_r| > dead zone), then the analog input effective data is: Analog input effective data = |Vin_r| - dead zone
Otherwise, the analog input effective data is 0.

Current speed = Analog input effective data * Analog-speed factor

If you use FULLING software for operation, you can follow these steps:
Connect an analog voltage between GND and AIN on CN3

After AMPS software is connected, click on "Parameter List" -> "Analog Quantity Parameters" to enter the following page.
Set the values, modify the working mode (3 or -3), and then enable (F).

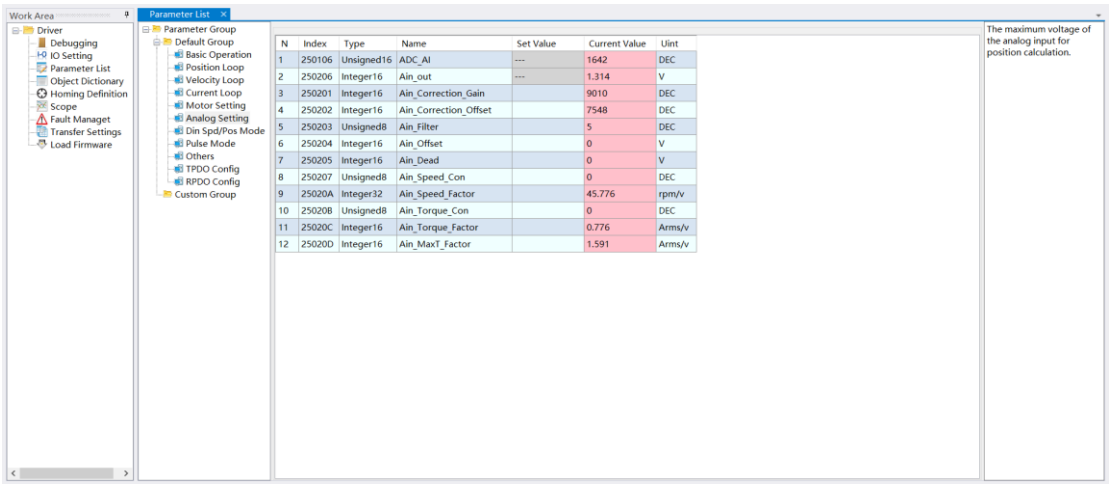


Figure 6-5 Analog input related parameters

6.2 Position Mode (1)

The position mode is include relative position mode, absolute position mode, and DIN position mode.
Regarding absolute position and relative position:
Absolute position: In the absolute position, each position has a unique encoding value corresponding to it. By reading the encoding value, the absolute position of the motor can be accurately determined.
Relative position: The relative position refers to the displacement of the motor or motion device relative to the starting position. The relative position usually requires an initial position reference point to accurately accumulate the displacement during the motion process.

We can further understand the absolute position and relative position through the following table:

Actual position	Execute operation	Actual position after execution
500	Absolute position 600	600
500	Relative position 600	1100

Table 6-5 Common Object List for Position Mode

Index	Data Type	Name	Description	Setting Value	Unit
606000	Integer8	Operation _ mode	User to set the motor operation mode 1: Position mode	1	DEC
604000	Unsigned16	Control word	Used to set the motor state Absolute position mode： 0x2F->0x3F(Single time) 0x103F(Multiple times) Relative position mode： 0x4F->0x5F DIN position mode：0x2F	According to application	HEX
607A00	Integer32	Target_ position	The position specified by the user	User-defined	inc
608100	Unsigned32	Profile_ speed	The speed at which the motor moves towards the target position in position mode		rpm

Note: 0x3F can only run to the first recognized target position. If you want to move to another position after reaching the target position, you need to write the control word 6 first, and then write 3F, and the motor will move to the next position.
0x103F does not need to change the control word, and the motor will move to the current value of the target position.

6.2.1 Internal Command Setting Position

The operation of setting the position mode with internal commands is basically the same as setting the speed mode in section 6.1.

6.2.2 External IO Port Setting Position (DIN Position Mode)

Table 6-6 Common Object List for DIN Position Mode

Index	Data Type	Name	Description	Setting Value	Unit
608100	Unsigned32	Profile speed	The speed at which the user will run to the specified position	User-defined	rpm
202001	Integer32	Din_Pos0	The position the user		DEC

202002	Integer32	Din_Pos1	needs the motor to reach, up to 8 different positions can be set, and the motor will select according to the BCD code input of the multi-position control through the IO port		DEC
202003	Integer32	Din_Pos2			DEC
202004	Integer32	Din_Pos3			DEC
202005	Integer32	Din_Pos4			DEC
202006	Integer32	Din_Pos5			DEC
202007	Integer32	Din_Pos6			DEC
202008	Integer32	Din_Pos7			DEC

Assuming the IO port is set as shown below:

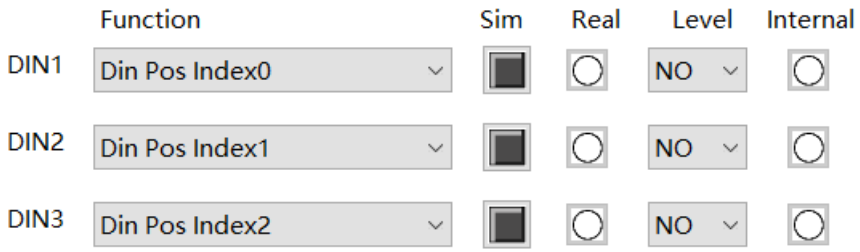


Figure 6-6 Multi-position - IO port settings

Then when the digital logic input of DIN3-DIN2-DIN1 is 010, it means to use multi-position control 2.

When the digital logic input of DIN3-DIN2-DIN1 is 011, it means to use multi-position control 3, and the motor will run to the specified position according to the set value.

6.2.3 Analog Input Setting Position

To use this mode, you need to first add "Analog input maximum value," "Analog input minimum value," "Analog position maximum value," and "Analog position minimum value" in the analog quantity parameter interface.

13	25020E	Integer16	Ain_MaxVol			V
14	25020F	Integer16	Ain_MinVol			V
15	250210	Integer32	Ain_MaxPosition			DEC
16	250211	Integer32	Ain_MinPosition			DEC

Table 6-7 Analog input setting speed related parameters

Index	Data Type	Name	Description	Setting Value	Unit
250106	Unsigned16	ADC_AI	Analog signal 1 original ADC data	Read Only	DEC
250206	Integer16	Ain_out	Analog Input signal1 (AIN1) input voltage after filter, deadband	Read Only	V

			and offset		
250201	Integer16	Ain_Correction_Gain	Analog input correction gain	User-defined	DEC
250202	Integer16	Ain_Correction_Offset	Analog input correction offset	User-defined	DEC
250203	Unsigned8	Ain_Filter	Analog input signal 1 (AIN1) filter	User-defined	DEC
250204	Integer16	Ain_Offset	Analog input signal 1 (AIN1) offset		V
250205	Integer16	Ain_Dead	Analog input signal 1 (AIN1) deadband		V
250207	Unsigned8	Ain_Speed_con	analog signal control speed, valid at operation mode 3 or -3 0: Invalid 1: Ain Control Speed 2: Ain Control Position	2	DEC
606000	Integer8	Operation mode	User to set the motor operation mode 1: Position mode	1	DEC
604000	Unsigned16	Control word	Used to set motor status 103F:Enable analog-position mode 6: Motor shaft release	103F or 6	HEX
25020E	Integer16	Analog_Input_MAX	Maximum value of analog input used for position calculation	User-defined	DEC
25020F	Integer16	Analog_Input_MIN	Minimum value of analog input used for position calculation	User-defined	DEC
250210	Integer32	Analog_Position_MAX	Maximum position value in analog position mode	User-defined	inc
250211	Integer32	Analog_Position_MIN	Minimum position value in analog position mode	User-defined	inc

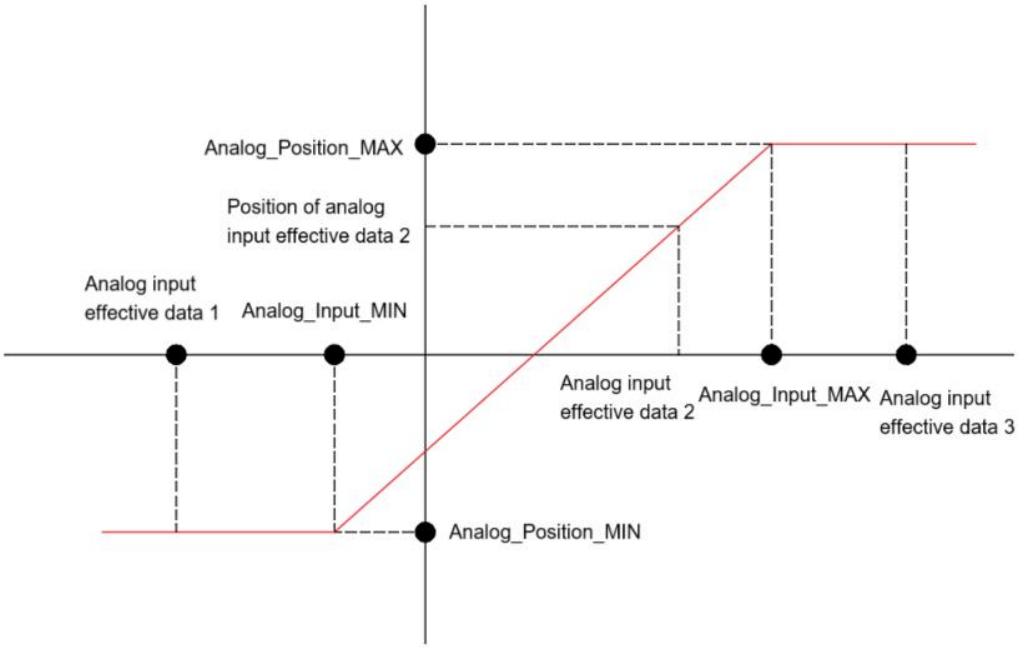


Figure 6-7 Simulated Input and Position Relationship

As shown in the figure above, in this mode:

When “analog input effective value” is less than Analog_Input_MIN, the motor is at Analog_Position_MIN.

When “analog input effective value” is greater than Analog_Input_MIN and less than Analog_Input_MAX, the motor position is on the straight line of the analog position function.

When “analog input effective value” is greater than Analog_Input_MAX, the motor is at "maximum simulated position".

6.3 Torque Mode (4)

Torque mode refers to the motor's output torque being set as the target value, and the control system is used to adjust the motor's output to make it as close as possible or maintain it at the setting target torque.

6.3.1 Internal Command Setting Torque

Table 6-8 Common Object List for Torque Mode

Index	Data Type	Name	Description	Setting Value	Unit
606000	Integer8	Operation mode	Used to set motor operation mode 4：Torque mode	4	DEC
604000	Unsigned16	Control word	Used to set motor status 0X0F: Enable driver	0x0F	HEX

607100	Integer16	Target torque%	Target torque/Rated torque*100%	User-defined	%
608000	Unsigned32	Max_Speed_Limit_rpm	Maximum speed limit in torque mode	User-defined	rpm

If using AMPS software for operation, follow these steps:

"Parameter List" -> "Basic Parameters" -> Set parameters such as Control Word, Operation Mode, etc.

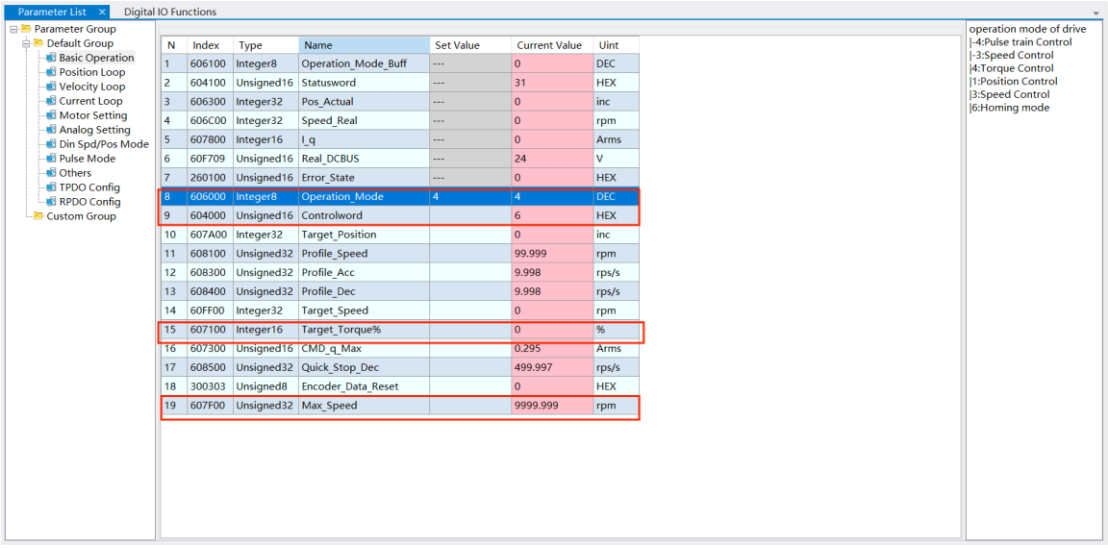


Figure 6-7: Parameters related to the torque mode

6.3.2 Analog Input Setting Torque

Table 6-9 Related Parameters of Analog Input Settings for Torque

Index	Data Type	Name	Description	Setting Value	Unit
250106	Unsigned16	ADC_AI	Analog signal 1 original ADC data	Read Only	DEC
250206	Integer16	Ain_out	Analog Input signal1 (AIN1) input voltage after filter, deadband and offset	Read Only	V
250201	Integer16	Ain_Correction_Gain	Analog input correction gain	User-defined	DEC
250202	Integer16	Ain_Correction_Offset	Analog input correction offset		DEC
250203	Unsigned8	Ain_Filter	Analog input signal 1 (AIN1) filter		DEC
250204	Integer16	Ain_Offset	Analog input signal 1 (AIN1) offset		V

250205	Integer16	Ain_Dead	Analog input signal 1 (AIN1) deadband		V
25020B	Unsigned8	Ain_torque control	analog signal control torque, valid at operation mode 4 0:Invalid 1:Ain Control Torque 2:Ain Control Max.Torque		DEC
25020C	Integer16	Ain_Torque_Factor	Proportion factor for converting analog input voltage to torque		Arms/V
25020D	Integer16	Ain_MaxT_Factor	Proportion factor for converting max analog input voltage to maximum torque		Arms/V

Assuming the input analog signal is Vin, after data processing, it is obtained as Vin_r, then we have:
Vin_r = Vin - Analog input offset voltage
If the absolute value of Vin_r is greater than the dead zone (i.e., |Vin_r| > dead zone), then the analog input effective data is:
Analog input effective data = |Vin_r| - dead zone
Otherwise, the analog input effective data is 0.
Driver's target torque = Analog input effective data * Analog-torque factor
For analog input setting torque can refer to section 6.1.4, which describes analog input setting speed

6.4 Pulse Mode (-4)

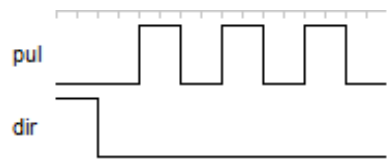
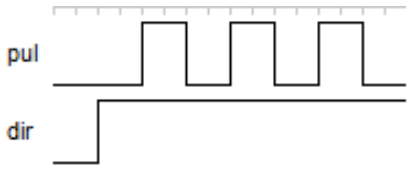
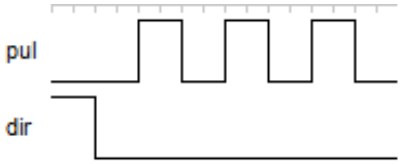
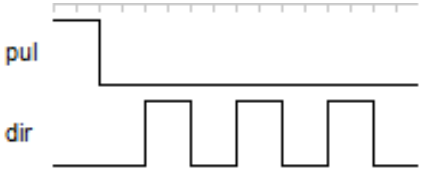
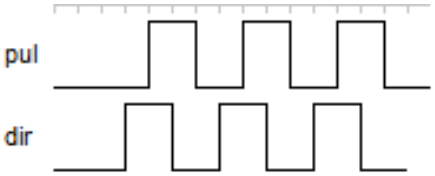
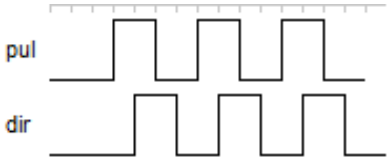
Table 6-10 Pulse Mode Related Parameters

Index	Data Type	Name	Description	Setting Value	Unit
606000	Integer8	Operation mode	Used to set motor operation mode -4：Pulse mode	-4	DEC
604000	Unsigned16	Control word	Used to set motor status 0x2F : Enable driver	0x2F	HEX
250806	Integer16	Pulse frequency before gear conversion	Master input pulse speed without gear ratio, unit:inc/ms	Read Only	kHz
250807	Integer16	Pulse frequency after gear	Master input pulse speed with gear ratio,		kHz

		conversionn	unit:inc/ms		
250804	Integer32	Gear_Master	Master_encoder pulse input counting without gear ratio	User-de fined	DEC
250805	Integer32	Gear_Slave	Master_encoder pulse input counting with gear ratio		DEC
250801	Integer16	Gear_Factor[0]	Gear ratio = Gear_Factor[0]/ Gear_Divider[0]		DEC
250802	Unsigned16	Gear_Divider[0]			DEC
250803	Unsigned8	PD_CW	Pulse control mode 0:CW/CCW 1:Pulse/Dirction 2:A/B(incremental encoder) Mode	0、1、2	DEC
250808	Unsigned16	PD_Filter	Master_encoder pulse input filter	User-de fined	DEC

Note: It needs to store control parameters and reboot driver after setting pulse mode.

Table 6-11 Pulse Mode Illustration

Mode Name	Motor rotary forward	Motor rotary reverse
CW/CCW		
Pulse/Dirction		
A/B(incremental encoder) Mode		

If using AMPS software for operation, follow these steps:

(1) Connect the DIR+, DIR-, PUL+, PUL- on the driver to the pulse source.

(2) Connect the driver to AMPS software.

(3) Click on "Parameter List" -> "Basic Operation" -> Set the corresponding parameters.

(4) Change the working mode and enable the driver.

Then it will allow control of the motor via external pulses.

6.5 Homing Mode (6)

In motor control systems, homing mode is a motion control mode used to determine the initial position of the motor, also known as the reference point or zero point.
The workflow for the home position mode is as follows:
Enter homing mode-> Search for the home position-> Determine the home position-> Return to the home position

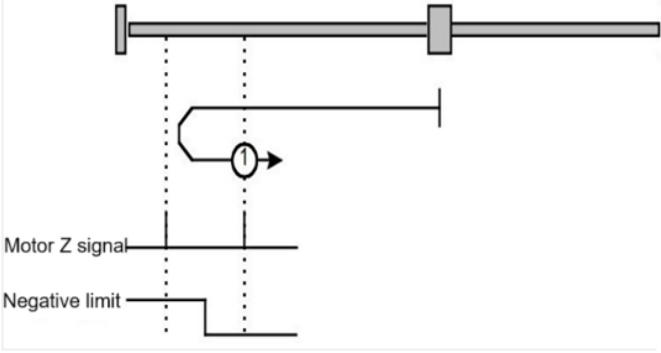
Table 6-12 Homing Mode Related Parameters

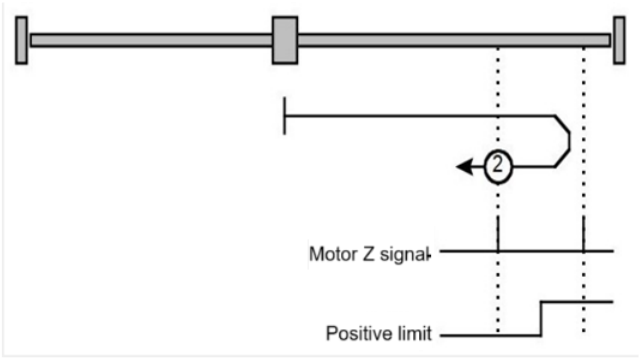
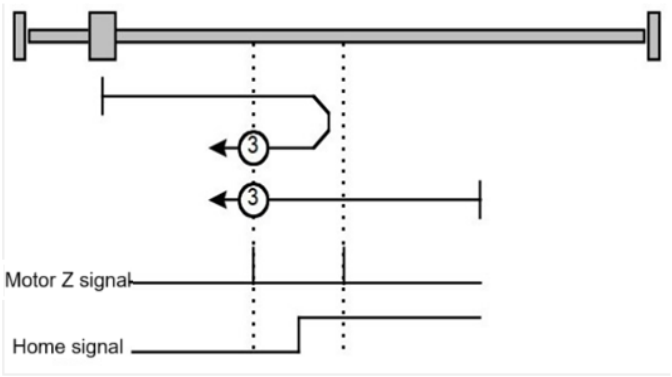
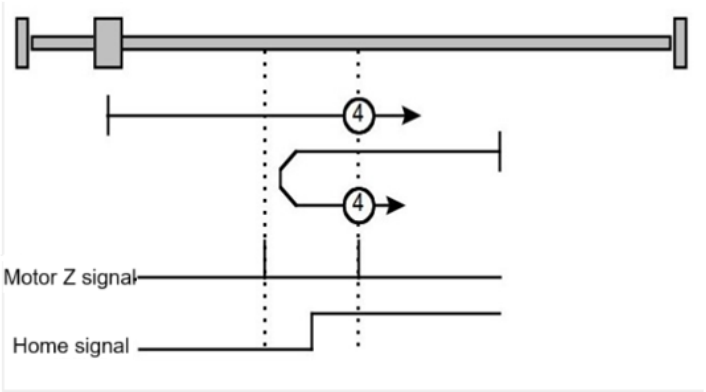
Index	Data Type	Name	Description	Setting Value	Unit
606000	Integer8	Operation mode	Used to set motor operation mode 6：Homing mode	6	DEC
604000	Unsigned16	Control word	Used to set motor status 0x0F->0x1F : Enable driver	F->1F	HEX
609800	Integer8	Homing method	method for searching homing	User-defined	DEC
609900	Unsigned8	Group_Homing_Speed			
609901	Unsigned32	Homing_Speed_Switch	velocity for searching position_limit switch/home_switch signal	User-defined	rpm
609902	Unsigned32	Homing_Speed_Zero	velocity for searching home signal and Zero position	User-defined	rpm
609903	Unsigned8	Homing_Power_On	bit0: 1--start searching homing when power on and the first enable driver bit1:1--auto save homing information	0、1	DEC
609904	Integer16	Homing_Current	Max current for finding homing	User-defined	Arms
609905	Unsigned8	Home_Offset_Mode	homing offset mode control 0: run to the homing offset point. The actual position will be 0. 1: run to the home event happen point.	0、1	DEC

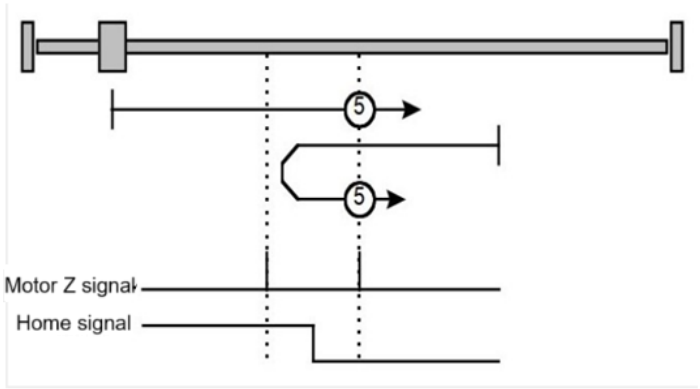
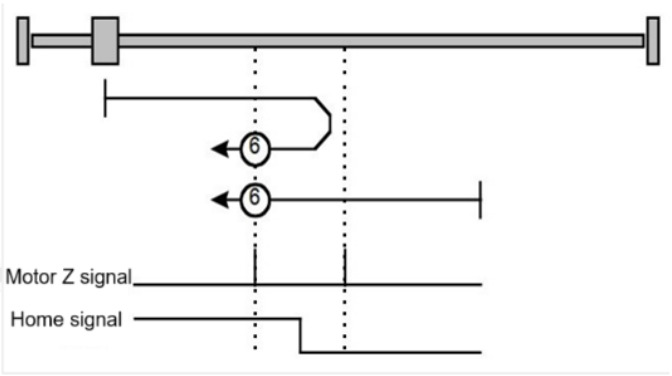
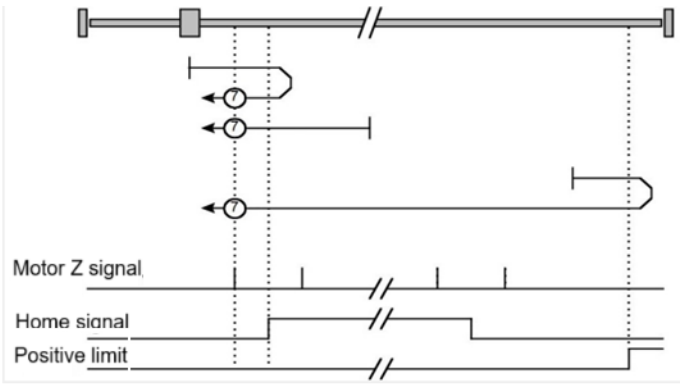
			The actual position will be "-homing offset"		
609906	Unsigned8	Home_N_Blind	Home_N_Blind	0、1	DEC
609A00	Unsigned32	Homing_Acceleration	Acceleration for searching home position		rps/s
607C00	Integer32	Home_Offset	The offset position between final stop position and zero position	User-defined	DEC

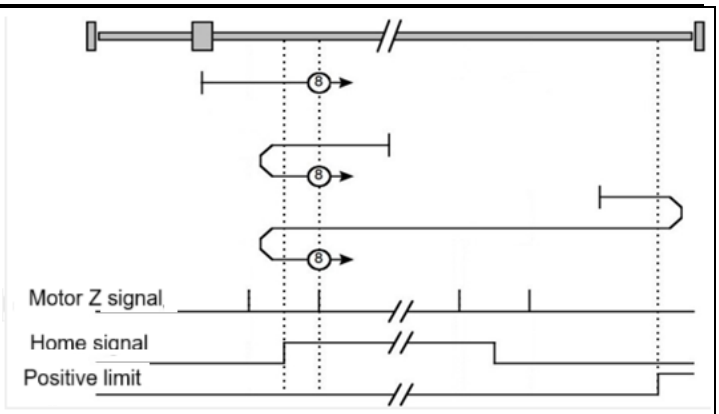
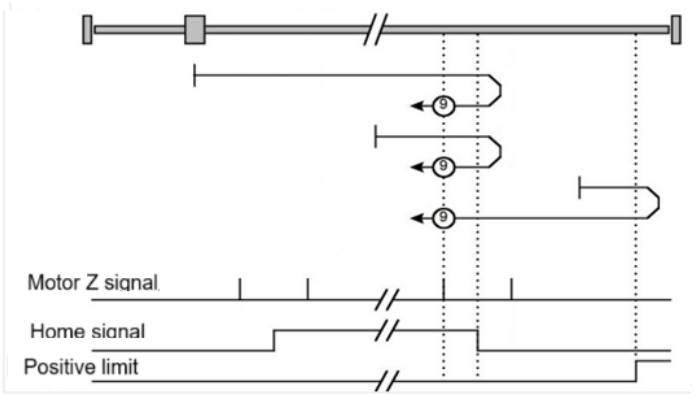
Regarding the Homing_Index_Blind:
Homing_Index_Blind is set to avoid different results when homing in the same machine. It is used in situations where the homing signal and the index signal are very close. By setting this parameter to 1 before homing, the driver will automatically find a suitable blind window. In this way, during the homing process, when the homing signal is found, the index signal within the blind window will be ignored. By default, the Homing_Index_Blind is 0, but setting it to 1 will automatically change to 0 or 2 according to the index signal position. If there are changes in the mechanical design later, just reset this parameter to 1. This ensures that the result of each homing is the same, avoiding inconsistency problems caused by close signals.

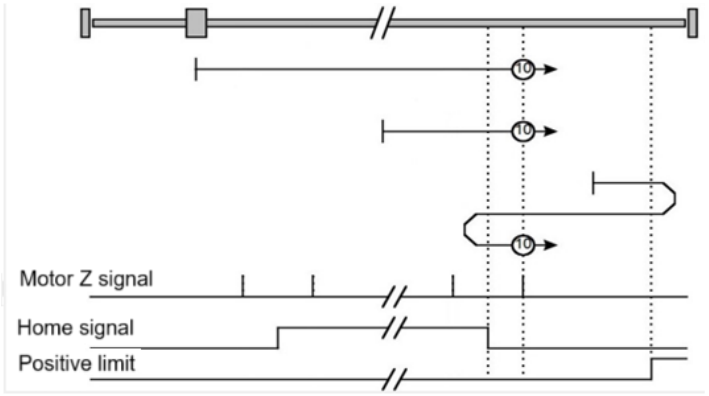
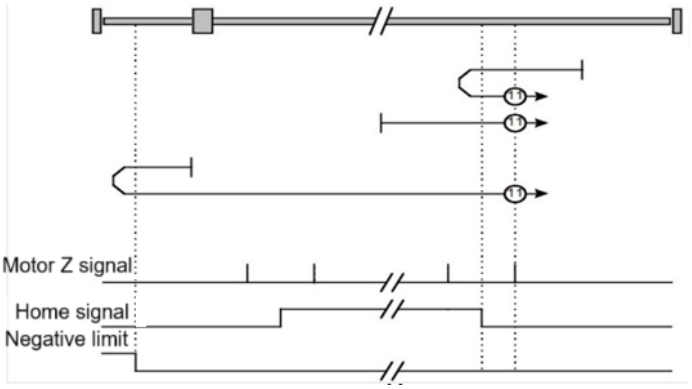
In the process of searching for the home position, there are different searching methods, as described in the homing mode mentioned above, and the following table describes some of the homing modes with illustrations.
Note:
（1）The high levels in the list below are effective levels, and the low levels are ineffective levels.
（2）When configuring the mode, attention should be paid to the timing of all parameters, not just one parameter line.

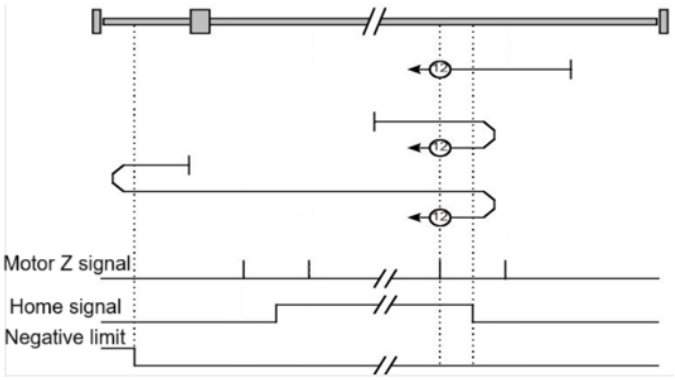
Homing Method	Mode Description	Illustration
1	When the negative limit is high, it reverses upon encountering the index signal; when the negative limit is low, it is the home position when encountering the index signal.	

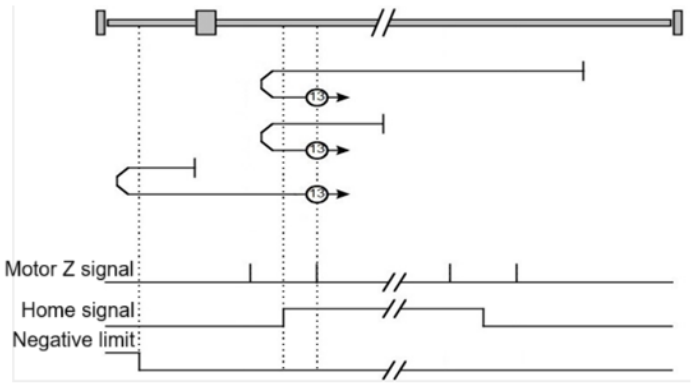
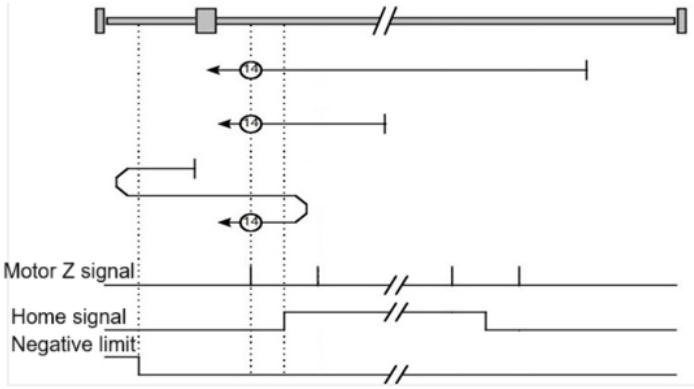
2	When the positive limit is high, it reverses upon encountering the index signal; when the positive limit is low, it is the home position when encountering the index signal.	
3	The initial direction is positive. If the home signal is low, the motor reverses when the home signal goes high and encounters the index signal. If the home signal is high, it marks the home position when it goes low and encounters the index signal.	
4	The initial direction is positive. If the home signal is high, the motor reverses when the home signal goes low and encounters the index signal. If the home signal is low, it marks the home position when it goes high and encounters the index signal.	

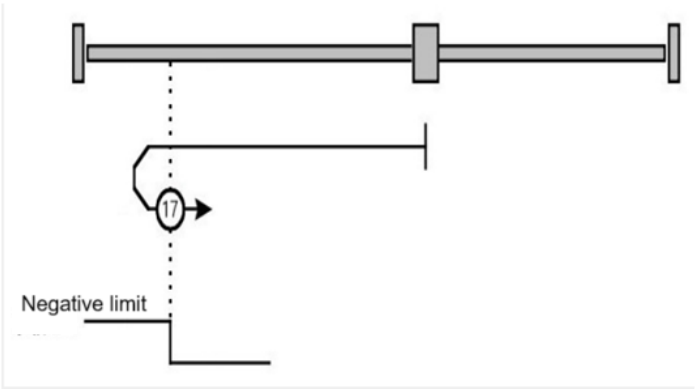
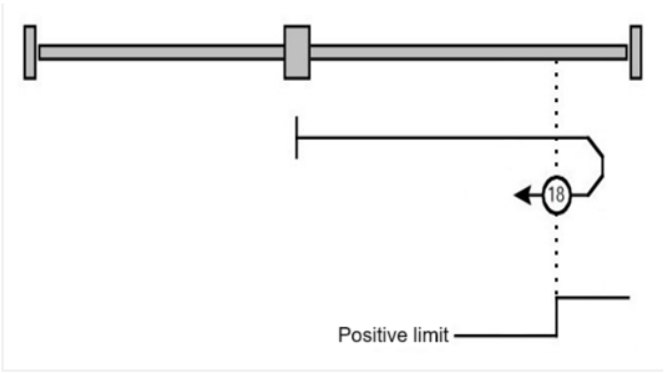
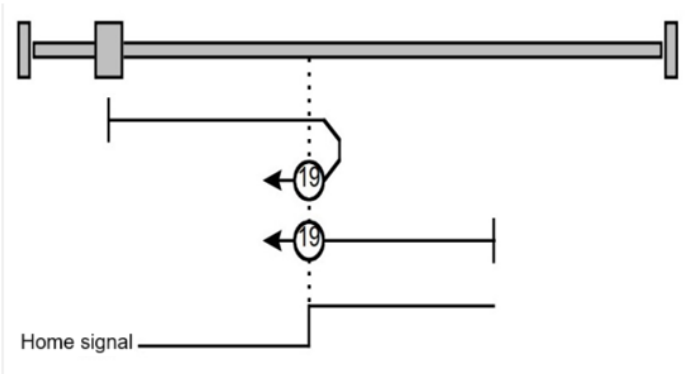
5	The initial direction is negative. If the home signal is low, the motor reverses when the home signal goes high and encounters the index signal. If the home signal is high, it marks the home position when it goes low and encounters the index signal.	
6	The initial direction is negative. If the home signal is high, the motor reverses when the home signal goes low and encounters the index signal. If the home signal is low, it marks the home position when it goes high and encounters the index signal.	
7	The motor Z signal and the home signal timing refer to mode 3; if the positive limit generates an rising edge, the motor reverses, and when the home signal generates a pulse (010) after encountering the index signal, it is marked as the home position.	

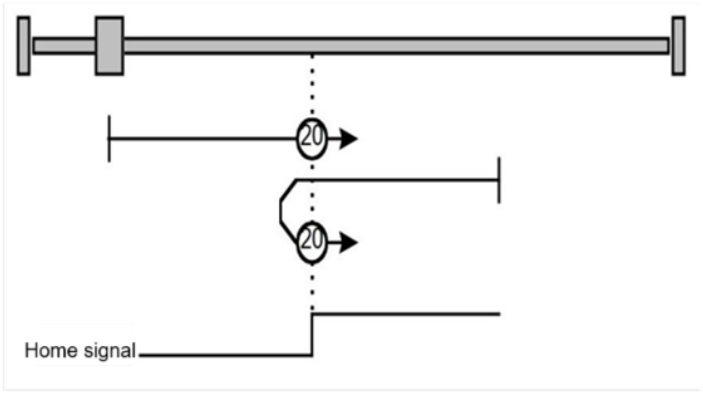
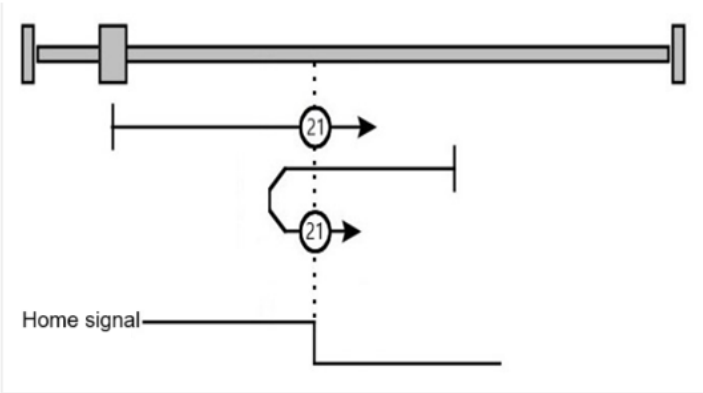
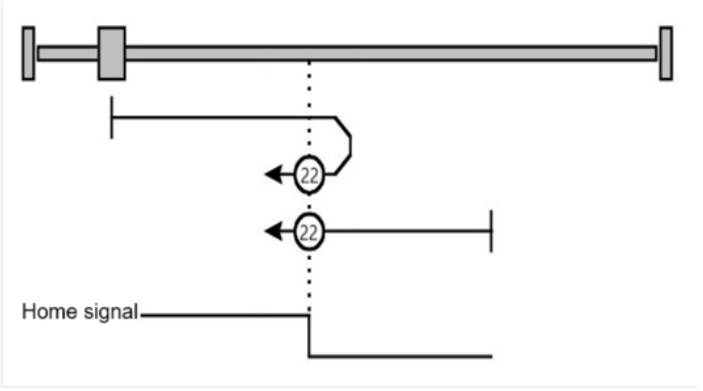
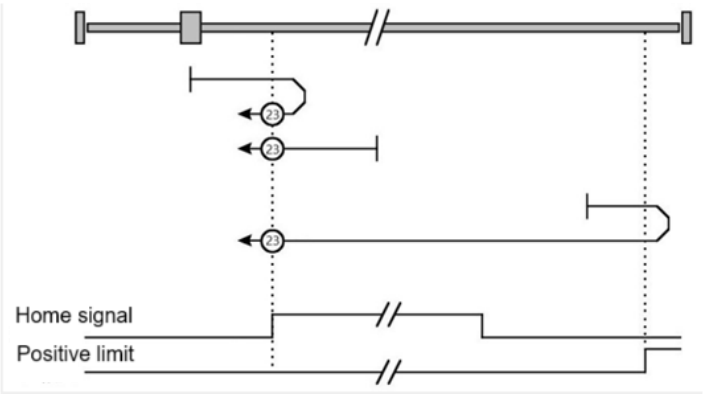
8	The motor Z signal and the home signal timing refer to mode 4; if the positive limit generates an rising edge, the motor reverses, and when the home signal generates an rising edge and then a downward edge, the motor reverses, and after reversing and encountering the index, it is marked as the home position.	
9	The initial direction is positive. ① When the positive limit encounters an rising edge, the motor reverses, and after the home signal generates an rising edge, it encounters the index signal and is marked as the home position. ② When the home signal generates a downward edge, the motor reverses, and after the home signal generates an rising edge again, it encounters the index signal and is marked as the home position.	

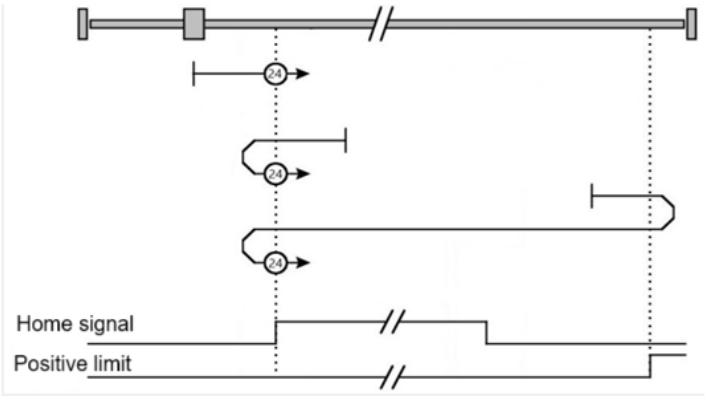
10	The initial direction is positive. ① When the positive limit generates an rising edge, the motor reverses, and after the home signal generates an rising edge, the motor reverses again, and after the home signal generates a downward edge, it encounters the index signal and is marked as the home position. ② When the home signal generates a downward edge, it encounters the index signal and is marked as the home position.	
11	The initial direction is negative. ① When the negative limit generates a downward edge, the motor reverses, and after the home signal generates a pulse (010) and encounters the index, it is marked as the home position. ② If the home signal changes from low to high, the motor reverses, and if the home signal	

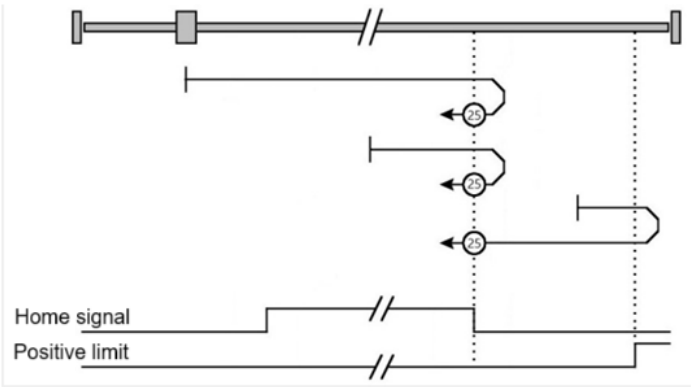
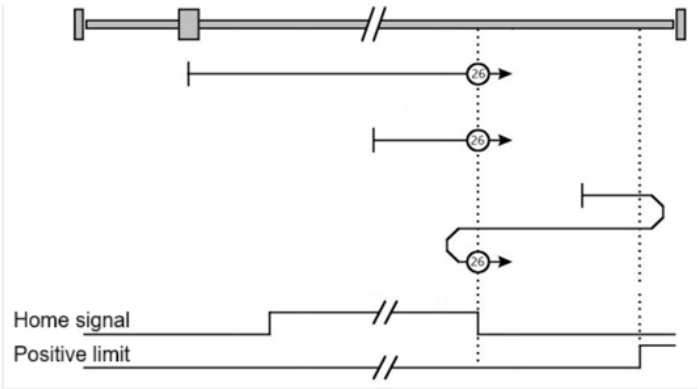
	changes from high to low again, it encounters the index and is marked as the home position.	
12	The initial direction is negative. ① When the negative limit generates a downward edge, the motor reverses, and after the home signal generates a pulse (010), the motor reverses again, and if the home signal changes from low to high, it encounters the index and is marked as the home position. ② If the home signal changes from high to low, the motor reverses, and if the home signal changes from low to high again, it encounters the index and is marked as the home position.	

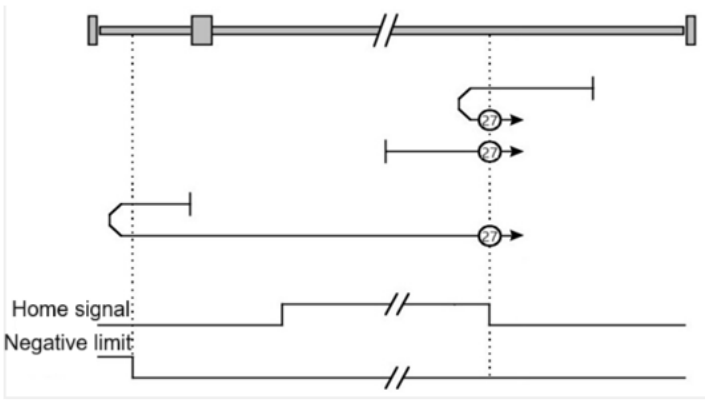
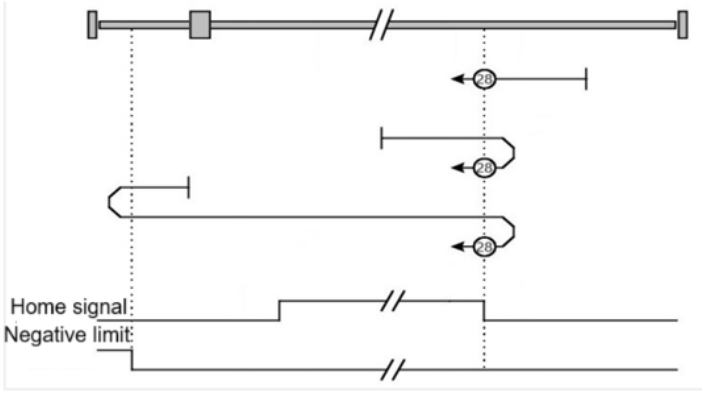
13	The initial direction is negative. ① When the negative limit generates a downward edge, the motor reverses, and if the home signal changes from low to high, it encounters the index and is marked as the home position. ② If the home signal changes from high to low, the motor reverses, and if the home signal changes from high to low again, it encounters the index and is marked as the home position.	
14	The initial direction is negative. ① When the negative limit generates a downward edge, the motor reverses, and if the home signal changes from low to high, the motor reverses again, and when the home signal changes from high to low, it encounters the index and is marked as the home position. ② If the home signal changes from	

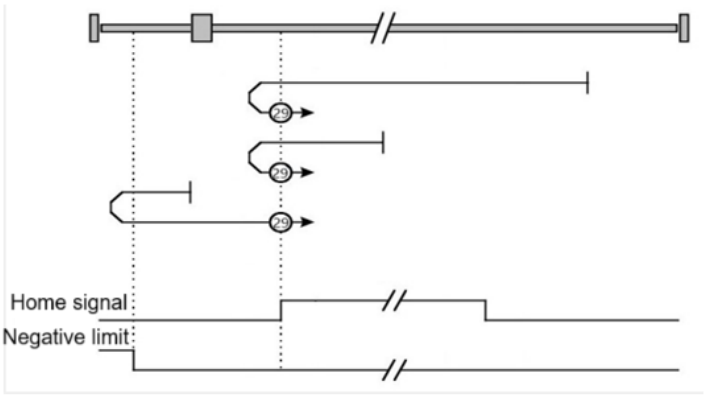
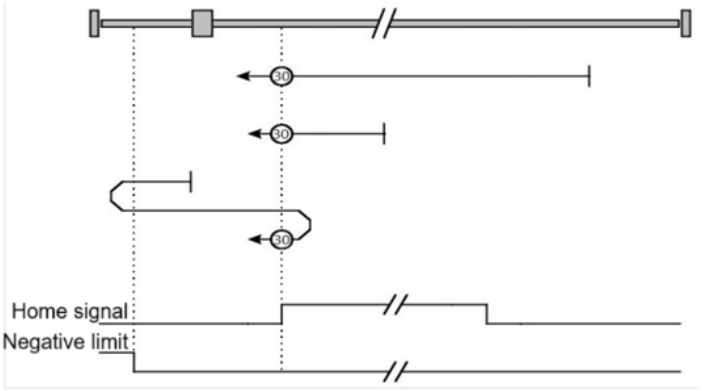
	high to low, the motor reverses, and if the home signal changes from high to low again, it encounters the index and is marked as the home position.	
17	When the negative limit generates an rising edge, the motor reverses; when the negative limit generates a downward edge, it is the home position.	
18	When the positive limit generates an rising edge, the motor reverses; when the positive limit generates a downward edge, it is the home position.	
19	The initial direction is the positive direction, when the home signal generates an rising edge, the motor reverses; when the home signal generates a downward edge, it is the home position.	

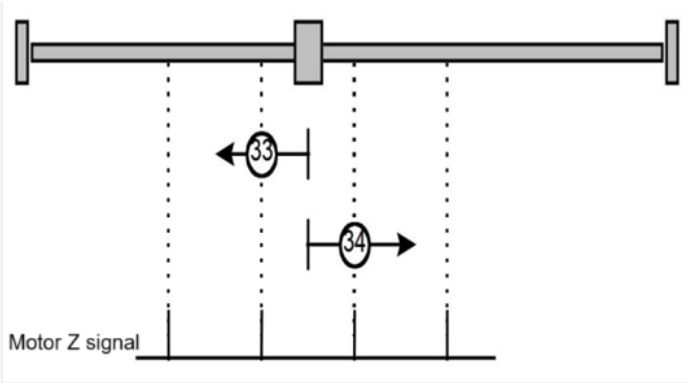
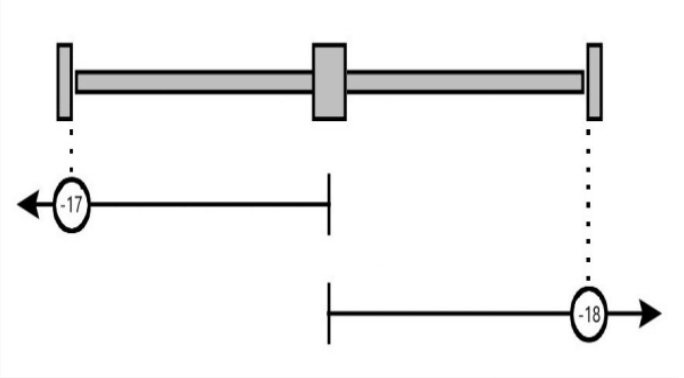
20	The initial direction is the positive direction, when the home signal generates a downward edge, the motor reverses; when the home signal generates an rising edge, it is the home position.	
21	The initial direction is the negative direction, when the home signal generates an rising edge, the motor reverses; when the home signal generates a downward edge, it is the home position.	
22	The initial direction is the negative direction, when the home signal generates a downward edge, the motor reverses; when the home signal generates an rising edge, it is the home position.	
23	The initial direction is positive. ① When the positive limit generates an rising edge, the motor reverses, and when the home signal generates a pulse (010), the downward edge is	

	marked as the home position. ② When the home signal generates an rising edge, the motor reverses, and when the home signal generates a downward edge again, it is marked as the home position.	
24	The initial direction is positive. ① When the positive limit generates an rising edge, the motor reverses, and when the home signal generates a pulse (010), the motor reverses again, and when the home signal changes from low to high, it is marked as the home position. ② When the home signal generates a downward edge, the motor reverses, and when it generates an rising edge again, it is marked as the home position.	

25	The initial direction is positive. ① When the positive limit generates an rising edge, the motor reverses, and when the home signal changes from low to high, it is marked as the home position (at the rising edge). ② When the home signal generates a downward edge, the motor reverses, and when the home signal generates an rising edge again, it is marked as the home position.	
26	The initial direction is positive. ① When the positive limit generates an rising edge, the motor reverses, and when the home signal changes from low to high, the motor reverses again, and if the home signal changes from high to low again, the downward edge is marked as the home position. ② When the home signal generates a downward edge, it is marked as the home position.	

27	The initial direction is negative. ① When the negative limit generates an rising edge, the motor reverses, and when the home signal generates a pulse (010), the downward edge is marked as the home position. ② When the home signal changes from low to high, the motor reverses, and when the home signal changes from high to low again, it is marked as the home position.	
28	The initial direction is negative. ① When the negative limit generates an rising edge, the motor reverses, and when the home signal generates a pulse and then reverses again, and when the home signal changes from low to high, it is marked as the home position. ② When the home signal changes from high to low, the motor reverses, and when the home signal changes from low to high again, it	

	is marked as the home position.	
29	The initial direction is negative. ① When the negative limit generates an rising edge, the motor reverses, and when the home signal generates an rising edge, it is the home position. ② When the home signal generates a downward edge, the motor reverses, and when the home signal generates an rising edge again, it is marked as the home position.	
30	The initial direction is negative. ① When the negative limit generates an rising edge, the motor reverses, and when the home signal generates an rising edge, the motor reverses again, and when the home signal generates a downward edge, it is marked as the home position. ② When the home signal generates a downward edge, it is marked as the home position.	

33、34	The next pulse of the motor Z signal	
35	Use the current position of the motor as the reference home position	
-17、-18	-17: The home position is the negative end of the mechanical limit -18: The home position is the positive end of the mechanical limit	

If using AMPS software for debugging, there are two methods:
The first method: Configure the homing curve by selecting:
Click "Homing Definition" in the work area

Assuming we need follows:
home signal - home switch signal
Use limit switches
Motor stop position - negative direction home signal falling edge
Set the homing parameters to default
Then we can follow these steps:
First step: Set homing parameters
Second step: Configure the homing curve according to the requirements
Third step: Click "Write"

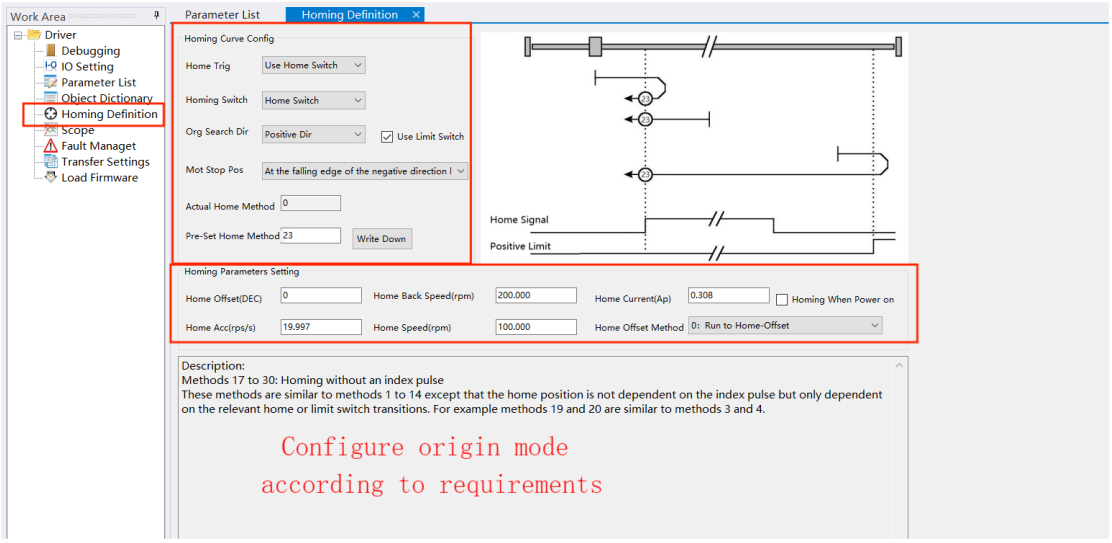


Figure 6-8 Homing Mode Configuration

（2）Click on "IO Settings" -> Set DIN1, DIN2 as home signal and positive limit (in the example, the electric level property is set to normally open, remember to power on first)

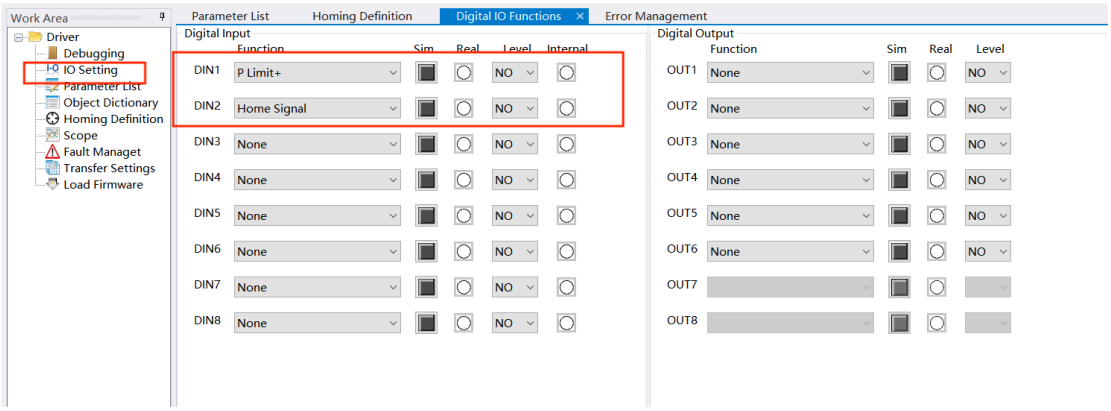
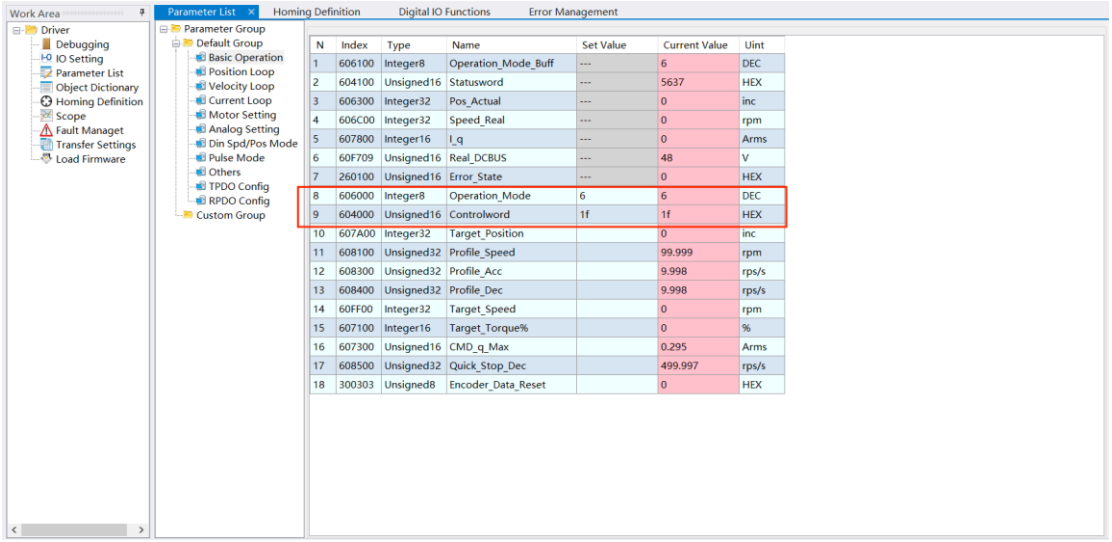


Figure 6-9 IO Port Settings

（3）In the basic parameters, set the working mode to 6, and change the control word from 0x0F -> 0x1F



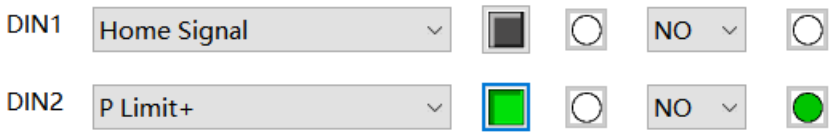
N	Index	Type	Name	Set Value	Current Value	Uint
1	606100	Integer8	Operation_Mode_Buff	---	6	DEC
2	604100	Unsigned16	Statusword	---	5637	HEX
3	606300	Integer32	Pos_Actual	---	0	inc
4	606C00	Integer32	Speed_Real	---	0	rpm
5	607800	Integer16	I_q	---	0	Arms
6	60F709	Unsigned16	Real_DCBUS	---	48	V
7	260100	Unsigned16	Error_State	---	0	HEX
8	606000	Integer8	Operation_Mode	6	6	DEC
9	604000	Unsigned16	Controlword	1f	1f	HEX
10	607A00	Integer32	Target_Position		0	inc
11	608100	Unsigned32	Profile_Speed		99.999	rpm
12	608300	Unsigned32	Profile_Acc		9.998	rps/s
13	608400	Unsigned32	Profile_Dec		9.998	rps/s
14	60FF00	Integer32	Target_Speed		0	rpm
15	607100	Integer16	Target_Torque%		0	%
16	607300	Unsigned16	CMD_q_Max		0.295	Arms
17	608500	Unsigned32	Quick_Stop_Dec		499.997	rps/s
18	300303	Unsigned8	Encoder_Data_Reset		0	HEX

Figure 6-10 Basic Parameter Settings

（4）By modifying the input value of the IO port, you can find the home position.

Note:

If using simulation, the limit signal needs to be inverted (the home signal does not need to be inverted). For example, in method 24, when the positive limit generates an rising edge, it can cause the motor to reverse. We need to change the IO port from the following state



to

to create an rising edge that will reverse the motor, and so on for other methods.

The second method: Directly write and select the mode

The second method is not much different from the first method. The second method is suitable for people who are familiar with the homing method. Direct writing can save some time.

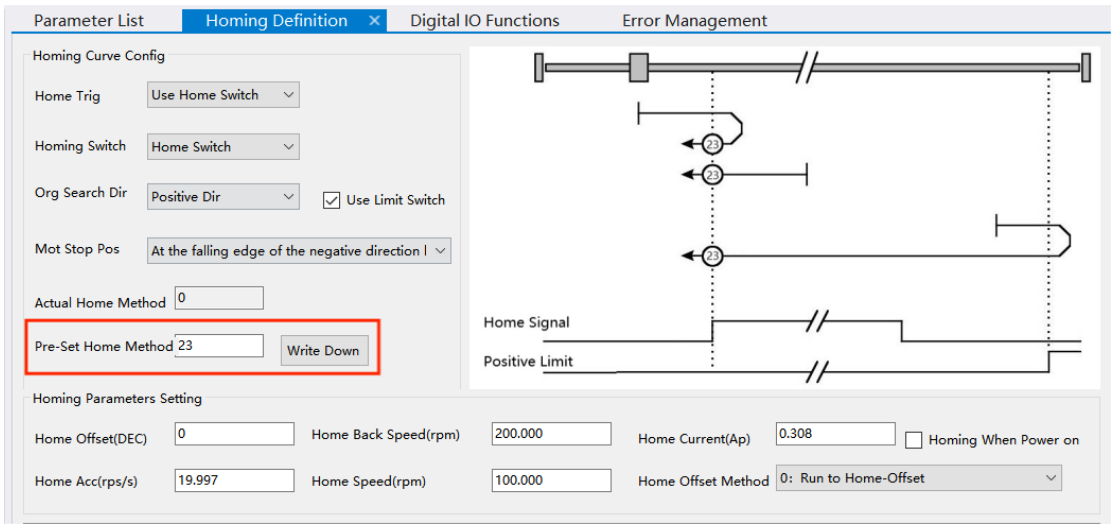


Figure 6-11 Directly Write Homing Method

Chapter 7: Performance Tuning

7.1 Speed Loop Tuning

Table 7-1 Speed Loop Tuning Related Parameters

Index	Data Type	Name	Description	Unit
60F901	Unsigned16	Kvp[0]	Speed loop proportional gain ,used to set the speed loop to follow the speed	DEC
60F902	Integer32	Kvi[0]	Speed loop integral gain ,used to eliminate the static error in the speed control	DEC
60F907	Integer32	Kvb_threshold		Ap
60F908	Integer32	Kvi_Sum_Limit	limiter of velocity control PI loop's integral part	Ap
60F915	Unsigned8	Speed_Demand_Filter	filtered Speed_Demand	rpm
60F903	Unsigned8	Notch_N	Frequency of notch filter BW=Notch_N*10+100[Hz]	Hz
60F904	Unsigned8	Notch_On	Notch filter enable	DEC
60F905	Unsigned8	Speed_Fb_N	Bandwidth of speed feedback filter BW=Speed_Fb_N*20+100[Hz]	Hz
60F906	Unsigned8	Speed_Mode	0:2nd Order FB LPF 1:No FB LPF 2:Observer FB 4:1st Order FB LPF 10:2nd LPF+SPD_CMD FT	HEX
60F90A	Integer32	Target_Speed_Window		inc/16.38 DEC rpm
60F91C	Unsigned16	Speed-loss judgment time	Loss of speed judgment time	ms
201018	Unsigned16	Dout_Real	bit0：Dout1 bit1：Dout2 bit2：Dout3 …	HEX
60F914	Integer16	CMD_q_PID	input value of notch filter	DEC

In the process of speed loop tuning, we mainly use two parameters - speed loop proportional gain and speed loop integral gain. They work together to affect the system's

response characteristics and stability. How they are used in conjunction depends on the specific requirements and characteristics of the system.

Speed loop proportional gain: The proportional gain controls the intensity of the system's response to speed errors, that is, the dynamic characteristics of the speed loop. Increasing the proportional gain can increase the system's response speed, but it may also lead to an increase in overshoot and system instability.

Speed loop integral gain: The integral gain is mainly used to eliminate static errors in the speed loop, ensuring that the system reaches accurate speed tracking in a steady state. Increasing the integral gain can reduce static errors, but it may also lead to system overshoot and oscillation.

When using these two parameters together, you can follow these principles:
Firstly adjust the speed loop proportional gain: Start with a relatively small speed loop proportional gain and gradually increase it to observe the system's response. Increasing the proportional gaincan improve the system's response speed, but be careful that too high a gain may cause overshoot and instability.

Then adjust the speed loop integral gain: Once a satisfactory speed response is achieved, you can start to gradually increase the speed loop's integral gain to reduce static errors. But be careful that too high an integral gain may cause system oscillation.

Observe overshoot and stability: During the adjustment process, observe the system's overshoot and stability. If there is too much overshoot or the system oscillates, it may be necessary to appropriately reduce the speed loop proportional gain and integral gain.

Balance response and stability: Adjust to find a balance between response speed and stability. If you need a faster response speed, you can appropriately increase the proportional gain; if you need lower static errors, you can appropriately increase the integral gain. But avoid too high a parameter setting that leads to system instability.

In summary, the speed loop proportional gain and speed loop integral gain need to be adjusted gradually in practical applications, and observe the system's response and stability. According to the system's requirements, balance the response speed and stability to find the best parameter settings.

The following are the effects of different KP and KI on speed in speed mode when the motor accelerates from 0rpm to 100rpm with an acceleration of 50rps/s.

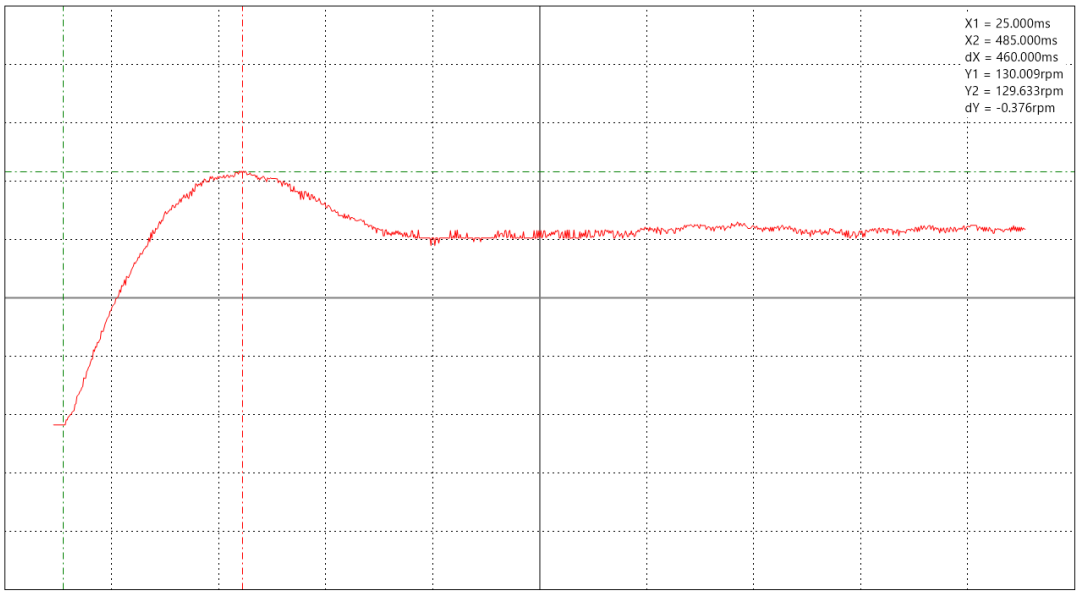


Figure 7-1 KP=100 KI=0

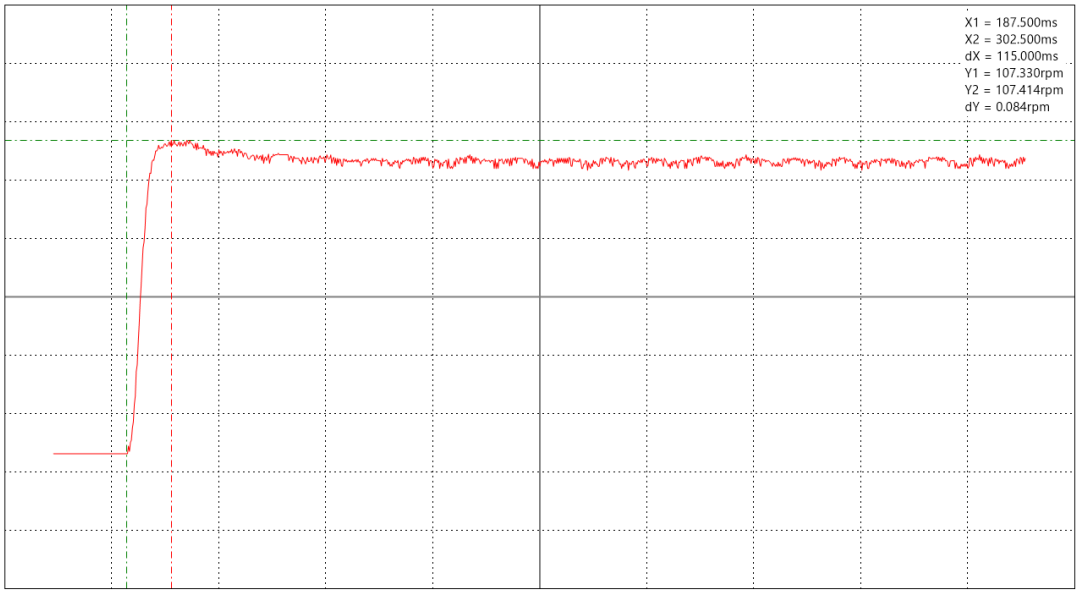


Figure 7-2 KP=1000 KI=0

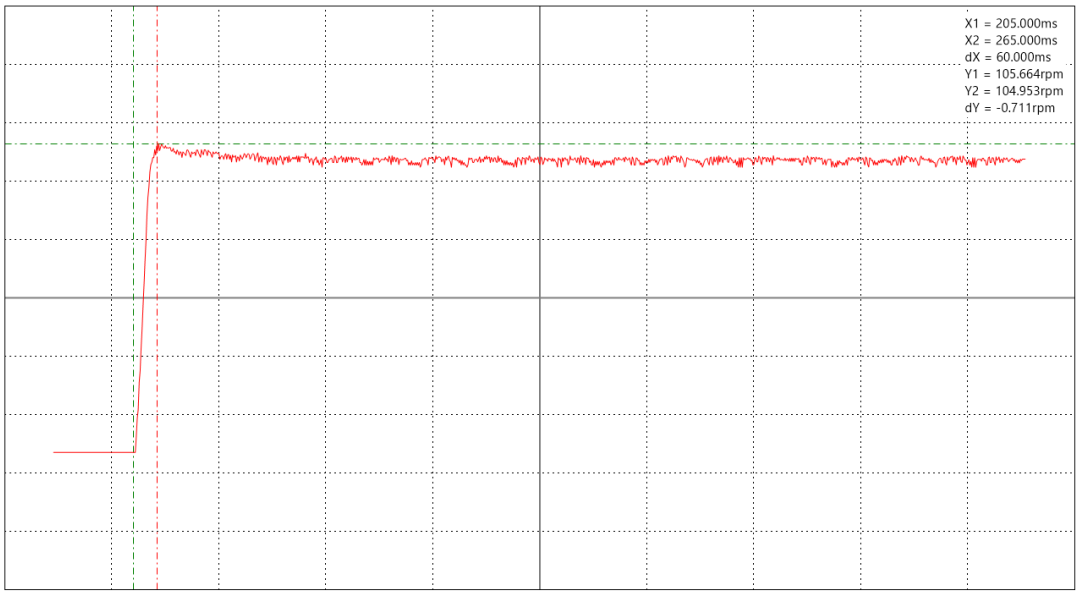


Figure 7-3 KP=2000 KI=0

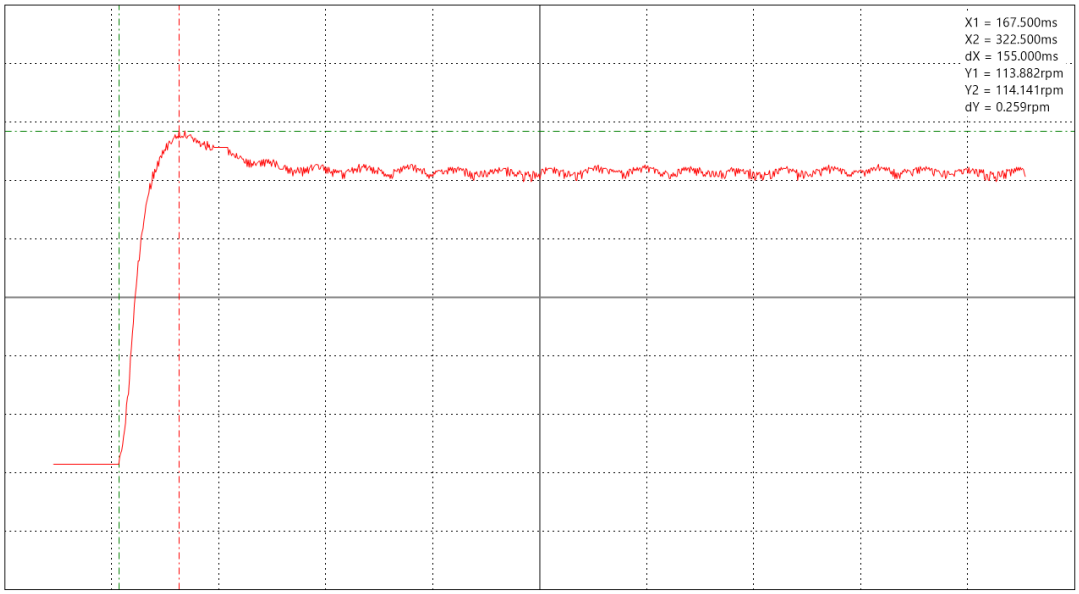


Figure 7-4 KP=500 KI=0

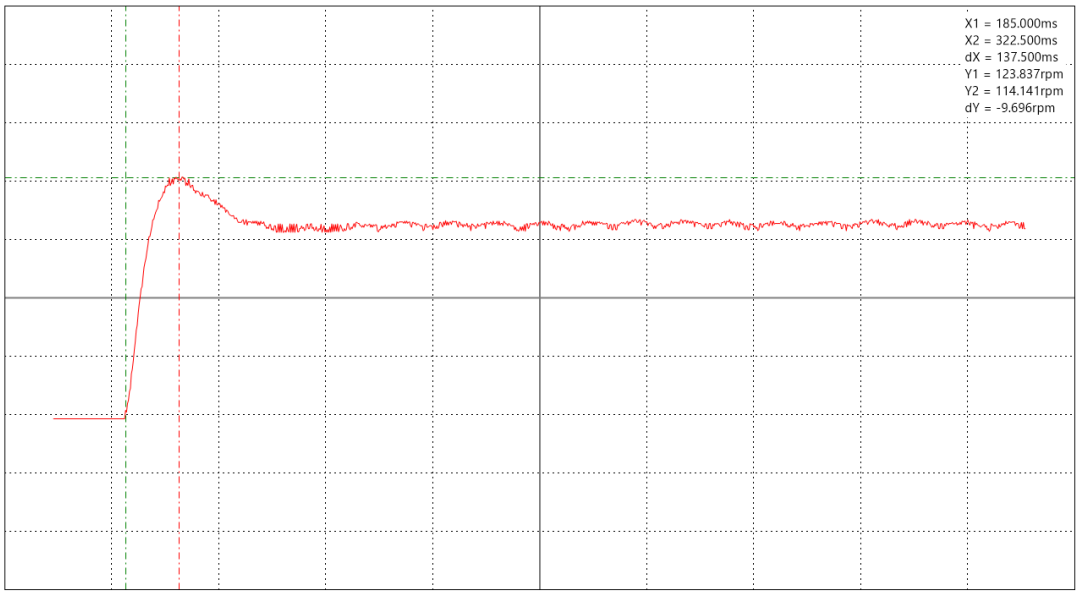


Figure 7-5 KP=500 KI=20

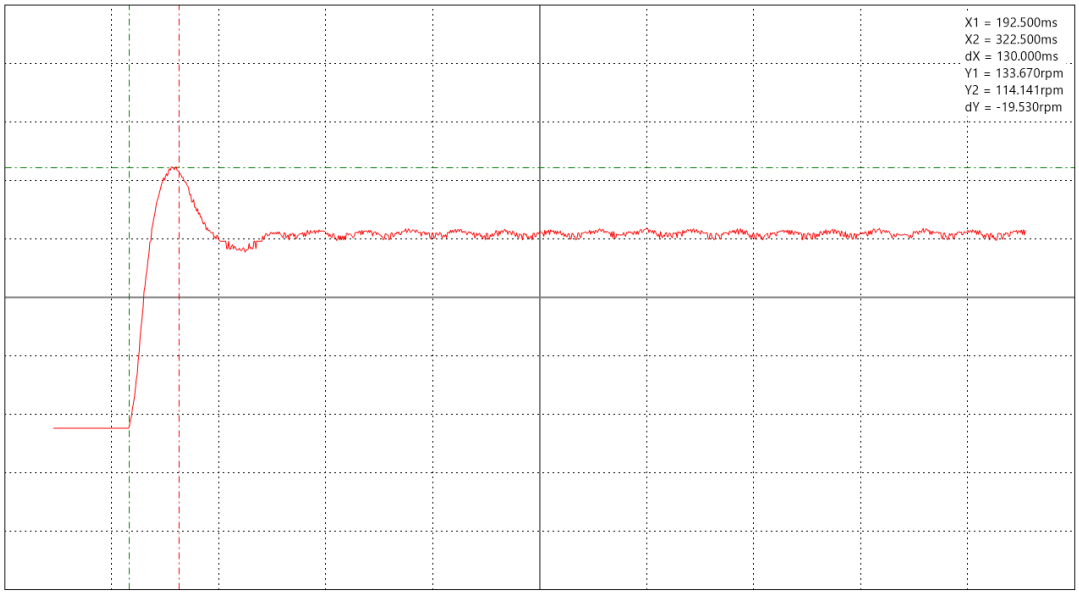


Figure 7-6 KP=500 KI=60

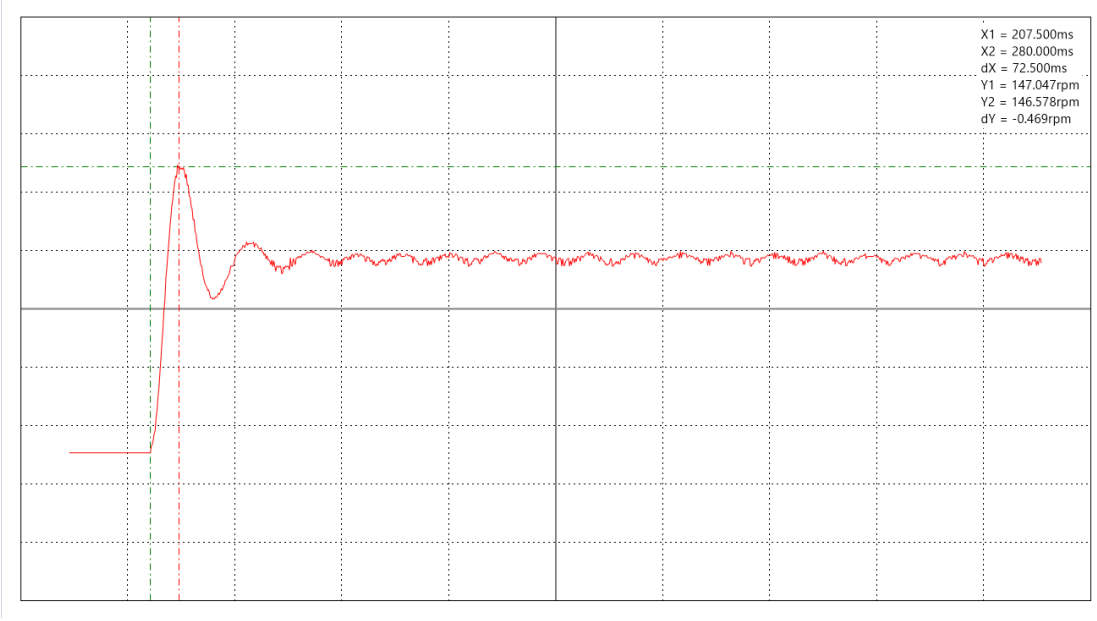


Figure 7-7 KP=500 KI=200

7.2 Position Loop Tuning

Table 7-2 Position Loop Tuning Related Parameters

Index	Data Type	Name	Description	Unit
60FB01	Integer16	Kpp[0]	Kp of position loop	Hz
60FB02	Integer16	K_Velocity_FF	velocity feedforward of position loop	%
60FB03	Integer16	K_Acc_FF	Acceleration feedforward of position loop Note:only CD3 accept the unit ""%"",set the value by the ""%"" must be base on the right auto-tuning result.Otherwise the value show by ""%"" could be not right.	DEC
60FB05	Unsigned8	Pos_Filter_N	Average filter parameter	DEC
606500	Unsigned32	Max_Following_Error	Applied to set the value of maximal following error	inc
606600	Unsigned16	Time_Following_Error	Following Error Timeout	ms
250809	Integer16	Reserved		DEC
606700	Unsigned32	Target_Pos_Window	target position window; In positioning mode, if position difference between Pos_Actual and Pos_Target is smaller than	inc

			"Target_Pos_Window"(6067.00) and lasting time >= Position_Window_time(6068.00) then Status_Word.bits.Target_reached is set to 1	
60F400	Integer32	Pos_Error	Following error of position	inc

The specific steps for position loop tuning are as follows:
Similar to speed loop tuning, we often use two parameters, position loop speed feedforward and position loop proportional gain, to work together to improve the system's positioning performance. The general steps are as follows:
Initial settings: First, set both the position loop proportional gain and the position loop velocity feedforward to smaller initial values to ensure that the system does not produce excessive overshoot or instability during the adjustment process.

Increase the position loop proportional gain: Gradually increase the position loop proportional gain and observe whether the system's positioning performance has improved. A larger proportional gain can make the system respond faster to position errors and reduce positioning errors.

Observe overshoot and stability: As the position loop proportional gain increases, observe whether the system's positioning response has become faster, and pay attention to whether there is an overshoot phenomenon and whether stability is affected.

Increase the position loop velocity feedforward: Under the appropriate proportional gain, gradually increase the position loop velocity feedforward. Velocity feedforward can predict the velocity command in advance to further reduce positioning errors, especially in situations such as variable speed or sudden stops.

Observe performance improvement: After increasing the position loop velocity feedforward, observe whether the system's positioning performance has improved. Pay special attention to whether the position error has been reduced and whether good stability is maintained under rapidly changing commands.

Parameter coordination and fine-tuning: The position loop velocity feedforward and the position loop proportional gain are interrelated, and their adjustments will affect each other. During the fine-tuning process, based on the actual situation, it may be necessary to adjust the two parameters multiple times to achieve the best positioning performance and stability.

7.3 Comprehensive Adjustment

Position loop tuning and speed loop tuning are two important links in the control system. They affect each other and need to be used reasonably in conjunction to achieve excellent control performance. The following are the general steps for their coordinated use:

Determine the tuning order: Generally, speed loop tuning should be performed first, followed by position loop tuning. This is because the position loop is controlled on the basis of the speed loop, so it is crucial to first ensure the stability and performance of the speed loop.

Speed loop tuning: Adjust the speed loop proportional gain and integral gain to achieve rapid speed response and stable speed tracking. Observe the overshoot, stability, and response time of the speed loop, and make fine adjustments according to the requirements.

Position loop tuning: Based on the stable speed loop, adjust the proportional gain of the position loop. Increasing the proportional gain can improve the positioning response speed, but be careful about the phenomenon of overshoot. Set the velocity feedforward parameters of the position loop to reduce position errors, especially in situations such as variable speed and sudden stops. According to the actual application requirements, adjust the integral gain of the position loop to eliminate static errors.

The tuning process may require multiple iterations of adjustment to achieve the best control performance. According to the actual test results, fine-tune the parameters of the speed loop and position loop as needed until satisfactory results are achieved.

Chapter 8: Alarm Troubleshooting

8.1 Troubleshooting with LED Alarm

When the drive operates normally, the green LED lights up, if the red LED lights up, you can further determine the cause of the fault by connecting to AMPS software. You can view the error code through Chapter 5.6 "Historical Errors and Alarms" and then go to 8.2 " Troubleshooting with Alarm Code" to resolve the issue further.

8.2 Troubleshooting with Alarm Code（603F00）

Alarm Code	Error Name	Error Causes	Correction Measures
FF40	Encoder ABZ Connection Alarm / Communication Encoder Break Error	1. Encoder ABZ wiring error, encoder connector loose, ABZ is damaged. 2. Communication encoder connection is loose, wiring sequence error, 3. encoder is damaged, driver encoder 5V output is damaged.	1.Check if the encoder cable is correctly connected. 2.Check if the encoder connector is loose. 3.Check if the encoder is damaged. 4.Replace the motor or encoder. 5.Check if the encoder power supply is intact.
FF41	Encoder UVW Connection Alarm / Communication Encoder Multi-turn Error	1. The Hall UVW wiring isincorrect, the Hall connector is loose, and the Hall is damaged. 2. The encoder has been disconnected from all power sources (including the encoder battery).	1.Check if the Hall sensor cable is correctly connected. 2.Check if the Hall sensor installation is loose. 3.Check if Hall sensor is damaged. 4.Replace Hall sensor or encoder. 5.Check the encoder battery voltage and reset the multi-turn error.
7305	Encoder Count Alarm / Communication Encoder CRC Error	Encoder interference or incorrect feedback cycle settings.	1.Check if the driver's ground wire is connected properly. 2.Check if the equipment's ground wire is good. 3.Power the driver with a separate power supply.

4210	Temperature Alarm	The temperature of the drive's power module has reached the alarm value.	1.Increase fans to improve the cooling environment of the electrical cabinet. 2.Appropriately increase the installation distance of the driver. 3.Check if the motor and driver selection is correct.
3210	High Voltage Alarm	Power supply voltage exceeds the allowed input range. Braking resistor not connected. Braking resistor mismatch.	1.Check if the power supply voltage is higher than the input range allowed by the driver. 2.Check if the power supply voltage is stable. 3.Confirm if the error occurs during deceleration; if so, consider increasing the braking resistor. 4.Confirm the load inertia and re-evaluate the selection of the braking resistor.
3220	Low Voltage Alarm	Power supply voltage is below the allowed input range	1.Check if the power supply meets the required specifications. 2.Replace with a power supply of higher wattage.
2320	Drive output short circuit	1.There is a short circuit problem at the drive UVW and PE output terminals.	1.Check if the motor power cable connection is correct. 2.The drive may be damaged, please consider replacing the drive.
7110	Absorption Resistor Alarm	Braking resistor parameters are not set correctly	
8611	Position Error Excessive Alarm	Control loop stiffness is too low. Motor phase sequence is incorrect.	1.Appropriately increase the ""Speed Loop Proportional Gain"" and ""Position Loop

		Drive or motor power is too low. Maximum following error value is too small.	Proportional Gain"". 2.Replace the motor UV phase wiring for testing. 3.Change to a higher power motor and driver. 4.Appropriately increase the ""Maximum Following Error"".
5112	Logic Low Voltage Alarm	Logic voltage is below 18V, power supply voltage is being pulled down.	1.Check if the power supply output power meets the requirements. 2.Replace with a power supply of greater output power.
2350	Motor or Driver IIT Alarm	1.Mechanical device is jammed or has excessive friction. 2.Motor phase sequence is incorrect. 3.Motor or driver power is too low.	1.Check if the motor is equipped with a brake and confirm whether the brake is normally released. 2.Power off the driver or disconnect the motor shaft from the load to check 3.if the motor and load move smoothly. Replace with a higher power motor and driver.
7310	Speed Deviation	Control loop stiffness is too low. Speed following error threshold is too small. Motor wiring sequence is incorrect. Encoder signal is faulty.	1.Appropriately increase the "Speed Loop Proportional Gain" and "Speed Loop Integral Gain". 2.Replace the motor's UV phase wiring for testing. 3.Appropriately increase the ""Speed Following Error Threshold"". 4.Check or replace the encoder.
4310	Speed Deviation	Control loop stiffness is too low. Speed following error threshold is too small. Motor wiring sequence is	1.Appropriately increase the "Speed Loop Proportional Gain" and "Speed Loop Integral Gain".

		incorrect. Encoder signal is faulty.	2.Replace the motor's UV phase wiring for testing. 3.Appropriately increase the "Speed Following Error Threshold". 4.Check or replace the encoder.
7122	Motor Excitation Alarm / Communication Encoder Other Errors (Refer to 30030208)	Motor UVW phase sequence is incorrect. Encoder is not connected.	1.Swap the U phase and V phase motor wires. 2.Check if the encoder connection is loose.
5530	Eeprom error	The drive fails to read the eeprom data correctly when powered on, the drive is disturbed, or the firmware is updated	1.Initialize the parameters first -> store the control loop parameters -> store the motor parameters -> restart. 2.Re-set the parameters or import the parameter file. 3.If the above two steps do not work, please contact the supplier.
5210	Current sensor exception	The current sensor signal is offset or has too much ripple.	The current sensor is damaged, please contact the supplier.
2214	Software overcurrent	The current ADC sampling is close to the maximum value.	1.Appropriately reduce the "target current limit." 2.Or contact the supplier.
8613	Homing error		
3130	Motor lack of phase	The motor UWV wiring is loose or the motor wire is not connected	Check if the motor UWV phase cable is loose or not connected.
6320	Motor data error	The motor model is not set, or the motor model does not exist.	Correctly set the motor model, save the motor parameters, and restart.
ff10	User Iit fault	Please refer to the reasons for the motor or drive Iit fault.	Please refer to the reasons for the motor or drive Iit fault.
5443	Pre-enable error	The IO input is set to "pre-enable", and the	1.If the "pre-enable" function is not needed,

		corresponding input does not receive a signal before the drive is enabled.	please cancel the corresponding IO function. 2.If the "pre-enable" function is needed, please first give a hardware enable signal at the corresponding IO port, and then send an enable command to the drive.
5442	Reached positive limit	The positive limit signal is triggered.	Check and eliminate the cause of triggering the limit
5441	Reached negative limit	The negative limit signal is triggered	Check and eliminate the cause of triggering the limit.
FF30	Pulse frequency too high	The input pulse frequency exceeds the allowed value.	Check the input pulse frequency
7500	Communication bus offline	The communication bus is loose, or the controller does not send a heartbeat signal to the drive on time.	1.Correctly set the bus offline time according to the controller's heartbeat signal. 2.Check if the controller's heartbeat signal is sent on time.
FF42	MCU watchdog error		Please contact the supplier

8.3 Troubleshooting with Error States

8.3.1 Error State 1（260100）

Error Bit	Error Name	Error Cause	Correction Measures
bit0	Refer to 26020010		
bit1	Encoder ABZ connection alarm/communication encoder disconnection error	1. The encoder ABZ wiring is incorrect, the encoder connector is loose, and the ABZ is damaged. 2.The communication	1.Check if the encoder cable is correctly connected. 2.Check if the encoder connector is loose. 3.Check if the encoder

		encoder is loosely connected, wrong wiring sequence, the encoder is damaged, the 5V output of the drive encoder is damaged.	is damaged. 4.Replace the motor or encoder. 5.Check if the encoder power supply is intact.
bit2	Encoder UVW connection alarm/communication encoder multi-turn error	1.The Hall UVW wiring isincorrect, the Hall connector is loose, and the Hall is damaged. 2.The encoder has been disconnected from all power sources (including the encoder battery).	1.Check if the Hall sensor cable is correctly connected. 2.Check if the Hall sensor installation is loose. 3.Check if Hall sensor is damaged. 4.Replace Hall sensor or encoder. 5.Check the encoder battery voltage and reset the multi-turn error.
bit3	Encoder counting error/communication encoder CRC error	The encoder is interfered with or the feedback cycle is set incorrectly.	1.Check if the drive ground wire is well connected. 2.Check if the equipment ground wire is well connected. 3.Use an independent power supply for the drive.
bit4	Drive temperature too high	The temperature of the drive power module reaches the alarm value.	1.Add the fan to improve the ventilation environment of the electrical cabinet. 2.Appropriately increase the installation distance of the drive. 3.Check if the motor and drive selection are correct.
bit5	Bus voltage too high	1.The power supply voltage exceeds the allowed input range. 2.No braking resistor is connected.	1.Check if the power supply voltage is higher than the allowed input range of the drive. 2.Check if the power

		3.The braking resistor does not match.	supply voltage is stable. 3.Confirm whether the error occurs during deceleration, if so, consider increasing the braking resistor. 4.Confirm the load inertia and re-evaluate the selection of the braking resistor.
bit6	Bus voltage too low	1.The power supply voltage is below the allowed input range. 2.The power supply power is too low.	1.Check if the power supply power meets the requirements. 2.Replace with a higher power power supply.
bit7	Drive output short circuit	1.There is a short circuit problem at the drive UVW and PE output terminals.	1.Check if the motor power cable connection is correct. 2.The drive may be damaged, please consider replacing the drive.
bit8	Braking resistor exception	The brake resistor parameters are not set correctly	
bit9	Following error too large	1.The control loop rigidity is too low. 2.The motor phase sequence is incorrect. 3.The power of the drive or motor is too small. 4.The maximum following error value is too small.	1.Appropriately increase the "Kvp" and "Kpp". 2.Replace the motor UV phase wiring for testing. 3.Change to a higher power motor and drive. 4.Appropriately increase the "maximum following error."
bit10	Logic power low voltage	The logic voltage is lower than 18V, and the power supply voltage is pulled down.	1.Check if the power supply output power meets the requirements. 2.Replace with a power supply with greater output power.
bit11	Motor or drive iit error	1.The mechanical device is stuck or the friction is	1.Check if the motor has a brake and confirm

		too large. 2.The motor phase sequence is incorrect. 3.The power of the motor or drive is too small.	whether the brake is normally released. 2.Power off the drive or disconnect the motor shaft from the load to check if the motor and load move smoothly. 3.Replace with a higher power motor and drive.
bit12	Speed follow error	1.The control loop rigidity is too low. 2.The speed follow error threshold is too small. 3.The motor phase sequence is incorrect. 4.There is a problem with the encoder signal.	1.Appropriately increase the "speed loop proportional gain" and "speed loop integral gain." 2.Replace the motor UV phase wiring for testing. 3.Appropriately increase the "speed follow-up error threshold." 4.Check or replace the encoder.
bit13	Motor temperature too high	The motor temperature exceeds the alarm value	1.Reduce the ambient temperature and improve cooling conditions. 2. Reduce the motor acceleration and deceleration. 3. Reduce the load.
bit14	Motor excitation error/other errors of the communication encoder (please refer to 30030208)	1.The motor UVW phase sequence is incorrect. 2.The encoder is not connected.	1.Swap the U and V motor wires. 2.Check if the encoder connection is loose.
bit15	Eeprom error	The drive fails to read the eeprom data correctly when powered on, the drive is disturbed, or the firmware is updated	1.Initialize the parameters first -> store the control loop parameters -> store the motor parameters -> restart. 2.Re-set the parameters or import

			the parameter file. 3.If the above two steps do not work, please contact the supplier.

8.3.2 Error State 2（260200）

Error Bit	Error Name	Error Cause	Correction Measures
bit0	Current sensor exception	The current sensor signal is offset or has too much ripple.	The current sensor is damaged, please contact the supplier.
bit1	Software overcurrent	The current ADC sampling is close to the maximum value.	1.Appropriately reduce the "target current limit." 2.Or contact the supplier.
bit2	Homing error		
bit3	Motor lack of phase	The motor UWV wiring is loose or the motor wire is not connected	Check if the motor UWV phase cable is loose or not connected.
bit4	Motor data error	The motor model is not set, or the motor model does not exist.	Correctly set the motor model, save the motor parameters, and restart.
bit5	User Iit fault	Please refer to the reasons for the motor or drive Iit fault.	Please refer to the reasons for the motor or drive Iit fault.
bit6	Reserved		
bit7	Reserved		
bit8	Pre-enable error	The IO input is set to "pre-enable", and the corresponding input does not receive a signal before the drive is enabled.	1. If the "pre-enable" function is not needed, please cancel the corresponding IO function. 2. If the "pre-enable" function is needed, please first give a hardware enable signal at the corresponding IO port, and then send an enable command to the drive.
bit9	Reached positive limit	The positive limit signal is triggered.	Check and eliminate the cause of triggering

			the limit
bit10	Reached negative limit	The negative limit signal is triggered	Check and eliminate the cause of triggering the limit.
bit11	Pulse frequency too high	The input pulse frequency exceeds the allowed value.	Check the input pulse frequency
bit12	Communication bus offline	The communication bus is loose, or the controller does not send a heartbeat signal to the drive on time.	1.Correctly set the bus offline time according to the controller's heartbeat signal. 2.Check if the controller's heartbeat signal is sent on time.
bit13	Full closed-loop encoder counting direction error	1.Phase A and phase B signals are connected incorrectly. 2.Incorrect setting of the encoder type. 3.There is a phase difference between the A and B signals of the encoder.	1.Phase A and phase B signals are connected incorrectly. 2.Incorrect setting of the encoder type. 3.There is a phase difference between the A and B signals of the encoder.
bit14	Main encoder connection error	The main encoder connection is incorrect.	Check if the encoder interface is normally connected.
bit15	Main encoder counting error	The encoder is interfered with or the feedback cycle is set incorrectly.	1.Check if the drive ground wire is well connected. 2.Check if the equipment ground wire is well connected. 3.Use an independent power supply for the drive.

Chapter 9 Common Object List

Object List Description
The object properties include data types, operation permissions, and parameter units.
Operation permissions include: R — Readable W — Writable

If there is no parameter unit in the object properties column, it means the parameter unit defaults to internal units (DEC).

Note:
When performing write operations on parameters, attention should be paid to unit conversion, i.e., converting the required data into internal units (DEC). For specific parameter-related conversions, please refer to the last section of this chapter "Unit Conversion."

9.1 Common Object List

9.1.1 Control Parameters

Name	UART Address	CANopen Address	Modbus Address	Properties	Description
Control word	604000	60400010	0x7400	Unsigned16 RW	bit0: Switch on bit1: Enable voltage bit2: Quick stop bit3: Enable operation bit4: Set-point(Mode 1)，Homing operation start(Mode 6)，Enable ip mode(Mode 7) bit5: Change set immediately(Mode 1) bit6: 0:related 1:absolute (Mode 1) bit7: Fault reset bit8: Halt bit9/bit10: Reserved bit11~bit15:Manufacturer specific
Status word	604100	60410010	0x7410	Unsigned16 R	bit0: Ready to switch on bit1: Switch on bit2: Operation

					enabled bit3: Fault bit4: Voltage enabled bit5: Quick stop bit6: Switch on disabled bit7: Warning bit8: Manufacturer specific bit9: Remote bit10: Target reached bit11: Internal limit active bit12: Set-point ack(Mode 1)，speed=0(Mode 3)，Homing attained(Mode 6)，Ip-Mode active(Mode 7) bit13: Following error(Mode 1)，Homing error(Mode 6) bit14: speed=0 bit15: Manufacturer specific
Operation mode	606000	60600008	0x7600	Integer8 R	Operation mode 式 -4: Pulse mode -3: Immediate speed mode 1: Position mode 3: Speed mode with Acceleration 4: Torque mode 6: Homing mode 7: Interpolation mode

9.1.2 DIN Mode

Name	UART Address	CANopen Address	Modbus Address	Properties	Description
Multi-position control 0	202001	20200120	0x4601	Integer32 RW	Select the absolute position of the motor

Multi-position control 1	202002	20200220	0x4602		through the IO port. For example, if the multi-position control 2 - multi-position control 1 - multi-position control 0 are set to 010 respectively, it indicates that the motor will move to the position controlled by multi-position control 2.
Multi-position control 2	202003	20200320	0x4603		
Multi-position control 3	202004	20200420	0x4604		
Multi-position control 4	202005	20200520	0x4605		
ulti-position control 5	202006	20200620	0x4606		
Multi-position control 6	202007	20200720	0x4607		
Multi-position control 7	202008	20200820	0x4608		

Name	UART Address	CANopen Address	Modbus Address	Properties	Description
Multi-speed control 0	202009	20200920	0x4609	Integer32 RW rpm	Select the motor running speed through the input of the IO port. For example, if the multi-speed control 2,Multi-speed Control 1 and Multi-speed Control 0 as set to 010, it indicates that the motor will run at the speed of Multi-speed Control 2.
Multi-speed control 1	20200A	20200A20	0x460A		
Multi-speed control 2	20200B	20200B20	0x460B		
Multi-speed control 3	20200C	20200C20	0x460C		
Multi-speed control 4	20200D	20200D20	0x460D		
Multi-speed control 5	20200E	20200E20	0x460E		
Multi-speed control 6	20200F	20200F20	0x460F		
Multi-speed control 7	202010	20201020	0x4610		

9.1.3 Digital I/O Parameters

Name	UART Address	CANopen Address	Modbus Address	Properties	Description
DIN polarity	201001	20100110	0x4201	Unsigned16 RW	Polarity definition of digital input signal bit0：Din1 bit1：Din2

					bit2：Din3 …
DIN simulate	201002	20100210	0x4202	Unsigned16 RW	Digital input simulate bit0：Din1 bit1：Din2 bit2：Din3 …
DIN1 function	201003	20100310	0x4203	Unsigned16 RW	DIN1 function definition（HEX） 0001：Drive enable 0002：Fault reset 0004：Pre-enable 0008：Kvi off 0010：Positive limit 0020：Negative limit 0040：Homing signal 0080 ： Speed command reverse 0100 ： Multi-speed control 0 0200 ： Multi-speed control 1 0400 ： Multi-speed control 2 0800：External input failure 1000 ： Emergency stop 2000: Start homing 4000: Reserved 8001: Multi-position control 0 8002: Multi-position control 1 8004: Multi-position control 2 8008: Electronic gear 0 8010: Electronic gear 1 8020: Electronic gear 2
DIN2 function	201004	20100410	0x4204	Unsigned16 RW	DIN2 function definition(Details

					refer to 201003）
DIN3 function	201005	20100510	0x4205	Unsigned16 RW	DIN3 function definition(Details refer to 201003）
DIN4 function	201006	20100610	0x4206	Unsigned16 RW	DIN4 function definition(Details refer to 201003）
DIN5 function	201007	20100710	0x4207	Unsigned16 RW	DIN5 function definition(Details refer to 201003）
DIN6 function	201008	20100810	0x4208	Unsigned16 RW	DIN6 function definition(Details refer to 201003）
DIN7 function	201009	20100910	0x4209	Unsigned16 RW	DIN7 function definition(Details refer to 201003）
DIN8 function	20100A	20100A10	0x420A	Unsigned16 RW	DIN8 function definition(Details refer to 201003）
DIN status	20100B	20100B10	0x420B	Unsigned16 R	bit0: Din1 bit1: Din2 bit2: Din3 …
DOUT polarity	20100E	20100E10	0x420E	Unsigned16 RW	Polarity definition of digital output
DOUT simulate	20100F	20100F10	0x420F	Unsigned16 RW	Digital output simulate
DOUT1 function	201010	20101010	0x4210	Unsigned16 RW	DOUT1 definition 0001：Drive ready 0002：Drive error 0004：Position reach 0008：Zero speed 0010：Motor brake 0020：Speed reach 0040：Index signal 0080：Speed limit 0100：Motor enable 0200: Position limit 0400: Home found 0800: Torque reach
DOUT2 function	201011	20101110	0x4211	Unsigned16 RW	DOUT2 definition(Details refer to 20100F)
DOUT3	201012	20101210	0x4212	Unsigned16	DOUT3

function				RW	definition(Details refer to 20100F)
DOUT4 function	201013	20101310	0x4213	Unsigned16 RW	DOUT4 definition(Details refer to 20100F)
DOUT5 function	201014	20101410	0x4214	Unsigned16 RW	DOUT5 definition(Details refer to 20100F)
DOUT6 function	201015	20101510	0x4215	Unsigned16 RW	DOUT6 definition(Details refer to 20100F)
DOUT7 function	201016	20101610	0x4216	Unsigned16 RW	DOUT7 definition(Details refer to 20100F)
DOUT8 function	201017	20101710	0x4217	Unsigned16 RW	DOUT8 definition(Details refer to 20100F)
DOUT status	201018	20101810	0x4218	Unsigned16 RW	Digital output status bit0：Dout1 bit1：Dout2 bit2：Dout3 …

9.1.4 Analog Input Mode

There is no Modbus address for following parameters,usually use AMPS software to modify parameters

Name	UART Address	CANopen Address	Properties	Description
Analog raw value	250201	25020110	Unsigned 16R	Raw value for analog input
Analog Input calibration gain	250202	25020210	Integer16 RW	Calibration gain for external analog input signal(related to hardware parameters)
Analog Input Calibration Offse	250203	25020308	Integer16 RW	Calibration offset for external analog input signal(related to hardware parameters)
Analog_Filter	250204	25020410	Unsigned8 RW	Filter coefficient for external analog input signal
Analog_Offset	250205	25020510	Integer16 RW	Offset for external analog input signal

Analog_Dead	250206	25020610	Integer16 RW	Dead zone setting for external analog input signal
Analog Input Effective Data	250207	25020708	Integer16 RW	Analog input signal after filter
Analog_Speed Control	250208	25020810	Unsigned8 RW	Analong input signal control speed,valid in mode 3 and -3 0：Disable 1：Ain control speed
Analog_Dead_H	250209	25020910	Integer16 RW	For analog control,if actual analog input value greater than this data,it will output 0 Default: 0, indicates invalid
Analog_Dead_L	25020A	25020A20	Integer16 RW	For analog control,if actual analog input value smaller than this data,it will output 0 Default: 0, indicates invalid
Analog_Speed_ Factor	25020B	25020B08	Integer32 RW	Convertion coefficient between analog and speed
Analog_Torque Control	25020C	25020C10	Unsigned8 RW	Analong input signal control torque,valid in mode 4 0：Disable 1：Ain control output torque 2：Ain control maximum torque
Analog_Torque_ Factor	25020D	25020D10	Integer16 RW	Convertion coefficient between analog and torque
Analog_MaxTorq ue_Factor	250201	25020110	Integer16 RW	Convertion coefficient between analog and maximum torque

9.1.5 Pulse Mode

Name	UART Address	CANopen Address	Modbus Address	Properties	Description
Electronic gear numerator 0	250801	25080110	0x5601	Integer16 RW	Electronic gear numerator
Electronic gear denominator 0	250802	25080210	0x5602	Unsigned16 RW	Electronic gear denominator
Pulse mode	250803	25080308	0x5603	Unsigned8 RW	Pulse mode control 0:Pulse direction mode 1:Dual pulse mode 2:Incremental encoder mode
Pulse raw data	250804	25080420	0x5604	Integer32 RW	Pulse count amount from IO port
Pulse data After electronic gear conversion	250805	25080520	0x5605	Integer32 RWL	Pulse data After electronic gear conversion
Pulse raw frequency	250806	25080610	0x5606	Integer16 R kHz	Pulse frequency from IO port
Pulse frequency After electronic gear conversion	250807	25080710	0x5607	Integer16 R kHz	Pulse frequency after electronic gear conversion = Pulse raw frequency*electronic gear ratio
Pulse filter coefficient	250808	25080810	0x5608	Unsigned16 RW	Filter parameter for pulse input
Electronic gear numerator 0	250801	25080110	0x5601	Integer16 RW	Electronic gear ration=electronic gear numerator/electronic gear denominator Similar to DIN mode, select the electronic gear through the input of the IO port. For example, if the
Electronic gear denominator 0	250802	25080210	0x5602	Integer16 RW	
Electronic	250901	25090110	0x5701	Integer16	

gear numerator 1				RW	Electronic gear 2,Electronic gear 1 and Electronic gear 0 as set to 010, it indicates that the motor will use Electronic gear 2.
Electronic gear denominator 1	250902	25090210	0x5702	Unsigned16 RW	
Electronic gear numerator 2	250903	25090310	0x5703	Integer16 RW	
Electronic gear denominator 2	250904	25090410	0x5704	Unsigned16 RW	
Electronic gear numerator 3	250905	25090510	0x5705	Integer16 RW	
Electronic gear denominator 3	250906	25090610	0x5706	Unsigned16 RW	

9.1.6 Device ID and Baudrate

Name	UART Address	CANopen Address	Modbus Address	Properties	Description
Device ID	2F8001	2F800108	0x6001	Unsigned8 RW	Drive station No.
CAN Baudrate	2F8002	2F800208	0x6002	Unsigned8 RW bps	CAN baudrate setting 100：1M 50：500k 25：250k 12：125k 10: 100k 5： 50k 2: 20k It needs to reboot drive for valid
UARTBaudrate	2F8003	2F800308	0x6003	Unsigned8 RW bps	UART baudrate setting 0: 4800 1: 9600 2: 14400

					3: 19200 4: 38400 5: 56000 6: 57600 7: 115200 It needs to reboot drive for valid
RS485Baudrate	2F8004	2F800410	0x6004	Unsigned16 RW bps	RS485 baudrate setting 48: 4800 96: 9600 192: 19200 384: 38400 576: 57600 1152: 115200 It needs to reboot drive for valid
Software_Version	100A00	100A00		Visible String	Drive Software Version

9.1.7 Motor Status

Name	UART Address	CANopen Address	Modbus Address	Properties	Description
Actual Position	606300	60630020	0x7630	Integer32R	Motor current position
Actual Speed	606C00	606C0020	0x76C0	Integer32R	Actual speed after filter
Actual Current_q	607800	60780010	0x7780	Integer16R	Actual current

9.1.8 Speed Mode

Name	UART Addres	CANopen Address	Modbus Address	Properties	Description
Target Speed	60FF00	60FF0020	0x8C00	Integer32 RW rpm	Target speed for speed mode
Profile Acceleration	608300	60830020	0x7930	Unsigned32 RW rps/s	Acceleration for accelerating from current speed to specific speed
Profile Deceleration	608400	60840020	0x7940	Unsigned32 RW	Deceleration for decelerating from

				rps/s	current speed to specific speed

9.1.9 Position Mode

Name	UART Address	CANopen Address	Modbus Address	Properties	Description
Target Position	607A00	607A0020	0x77A0	Integer32 RW	Target position for position mode
Profile Speed	608100	60810020	0x7910	Unsigned32 RW rpm	Speed for position mode

9.1.10 Torque Mode

Name	UART Address	CANopen Address	Modbus Address	Properties	Description
TargetTorque%	607100	60710010	0x7710	Integer16 RW %	Target torque/Rated torque*100%
MAX_Speed_Limit_rpm	608000	60800010	0x7900	Unsigned16 RW rpm	Maximum speed limit for torque mode

9.1.11 Speed Loop Parameter

Name	UART Address	CANopen Address	Modbus Address	Properties	Description
Kvp 0	60F901	60F90110	0x8601	Unsigned16 RW	Speed loop kp
Kvi 0	60F902	60F90210	0x8602	Unsigned16 RW	Speed loop ki
Notch_N	60F903	60F90308	0x8603	Unsigned8 RW	Notch filter frequency setting for spped loop BW=Notch_N*10+100[Hz]
Notch_Control	60F904	60F90408	0x8604	Unsigned8 RW	Notch filter control for speed loop
Speed_Fb_N	60F905	60F90508	0x8605	Unsigned8 RW	Speed feedback filter for speed loop BW=Speed_Fb_N*20+100[Hz]
Speed_Fb_Mode	60F906	60F90608	0x8606	Unsigned8 RW	Speed feedback mode selection

					Bit 0: Direct feedback Bit 1: Observer feedback Bit 2: Low-pass feedback order, 0 = 2nd order, 1 = 1st order Bit 4: Speed command filter, 0 = Disabled, 1 = Enabled Default: 0, which means 2nd order low-pass filte
Speed loop Kvb shreshold	60F907	60F90720	0x8607	Integer32 RW	Speed loop Kvb shreshold
Speed Loop Kvi limit	60F908	60F90820	0x8608	Integer32 RW	Speed loop integral gain limit
Speed loop Kvb	60F909	60F90910	0x8609	Unsigned16 RW	Speed loop Kvb
Speed_reach_window	60F90A	60F90A20	0x860A	Integer32 RW	Speed error window
Speed_loss judgement time	60F91C	60F91C10	0x861C	Unsigned16 RW ms	Time for speed-loss judgement
Notch_filter_input	60F914	60F91410	0x8614	Integer16 R	Notch filter input

9.1.12 Position Loop Parameter

Name	UART Address	CANopen Address	Modbus Address	Properties	Description
Kvp 0	60FB01	60FB0110	0x8801	Integer16 RW	Position loop kp
K_Speed_FF	60FB02	60FB0210	0x8802	Integer16 RW	Position loop feedforward
K_Acceleration_FF	60FB03	60FB0310	0x8803	Integer16 RW	Position loop acceleration feedforward
Smoothing filter	60FB05	60FB0508	0x8805	Unsigned8 RW	Smoothing filter parameter adjustment

Max_Following_Error	606500	60650020	0x7650	Unsigned32 RW	If following error is greater than 6065h,and the time is greater than 6066h，drive will appear following error alarm
Position_Reach_Window	606700	60670020	0x7670	Unsigned32 RW	The difference between the position command and the actual position is less than Position_Reach_Window, and the time reaches the Position_Reach_Window_Time, it is considered that the position has arrived.
Position following error	60F400	60F40020	0x8100	Integer32 R	Position following error

9.1.13 Current Loop Parameter

Name	UART Address	CANopen Address	Modbus Address	Properties	Description
Target_Current_Limit	607300	60730010	0x7730	Unsigned16 RW Ap	Maximum value Of current command
Kcp	60F601	60F60110	0x8301	Unsigned16 RW	Current loop kp
Kci	60F602	60F60210	0x8302	Unsigned16 RW	Current loop ki
Current_Compensation_Factor	60F603	60F60310	0x8303	Unsigned16 RW	Current compensation factor
Voltage_Feedback_Factor	60F604	60F60410	0x8304	Integer16 RW	Voltage feedback factor
Internal_MaxTorque_Limit	60F60C	60F60C10	0x830C	Integer16R	Drive internal maximum torque limit
Drive_Actu	60F60D	60F60D10	0x830D	Unsigned16R	Actual data of iit

al_iit				%	Protection for drive
Drive_iit_Max	60F60E	60F60E10	0x830E	Unsigned16RAp	Maximum data of iit Protection for drive
Motor_Actual_iit	60F60F	60F60F10	0x830F	Unsigned16R%	Actual data of iit Protection for motor
Motor_iit_Max	60F610	60F61010	0x8310	Unsigned16RAp	Maximum data of iit Protection for motor

9.1.14 Homing Mode

Name	UART Address	CANopen Address	Modbus Address	Properties	Description
Homing Method	609800	60980008	0x7C00	Integer8 RW	Method for homing,refer to chapter 6 for details
Homing_Speed_Switch	609901	60990120	0x7D01	Unsigned32 RW rpm	Speed for searching home switch or position limit switch
Homing_Speed_Zero	609902	60990220	0x7D02	Unsigned32 RW rpm	Speed for searching home position or zero position
Homing_Power_On	609903	60990308	0x7D03	Unsigned8 RW	Everytime execute homing mode once when driver power on
Homing_Current_Max	609904	60990410	0x7D04	Integer16 RW Ap	Maximum current during homing
Home_Offset_Mode	609905	60990508	0x7D05	Unsigned8 RW	Home offset mode control 0：Run to home offset position 1：Run to home trigger position.The actual position will be –Home_Offset
Homing_Index_Blind	609906	60990608	0x7D06	Unsigned8 RW	When the homing mode is used simultaneously with the limit/home switch and index signal,

					ignore the detected index signal during homing process within the blind area after encountering the switch signal. 0: 0 revolutions 1: 0.25 revolutions 2: 0.5 revolutions Default: 0. When this value is set to 1, the homing process will offset this value to 0 or 2 based on the position of the index signal relative to the switch signal. After mechanical fixation, this parameter needs to be saved, and after mechanical changes, it can be reset to 1.
Homing_Acceleration	609A00	609A0020	0x7E00	Unsigned32 RW rps/s	Acceleration for homing

9.1.15 Error States

Name	UART Address	CANopen Address	Modbus Address	Properties	Description
Error_States	260100	26010010	0x5A00	Unsigned16 R	Error states bit0：Refer to 26020010 bit1 ： Encoder ABZ connection alarm/communication encoder disconnection error bit2 ： Encoder UVW connection alarm/communication encoder multi-turn error bit3 ： Encoder count error/communication encoder CRC error

					bit4：Drive temperature too high bit5：Bus voltage too high bit6：Bus voltage too low bit7：Drive output short circuit bit8：Brake resistor exception bit9：Following error too large bit10：Logic power low voltage bit11：Motor or drive iit fault bit12：Speed follow error bit13：Motor temperature too high bit14：Motor excitation error/other errors of the communication encoder (please refer to 30030208) bit15：Eeprom alarm
Error_States_2	260200	26020010	0x5B00	Unsigned16 R	Error states 2 bit0：Current sensor exception bit1：Drive output short circuit bit2：Homing error bit3：Motor lack of phase bit4：Motor configuration error bit5：User Iit fault bit6：Reserved bit7：Reserved bit8：Pre-enable error bit9：Positive limit error bit10：Negative limit error bit11: Pulse frequency too high

					bit12：Bus offline error bit13：Full closed-loop encoder counting direction error bit14：Main encoder connection error bit15：Main encoder counting error

9.1.16 Stop Mode

Name	UART Address	CANopen Address	Modbus Address	Properties	Description
Parameter name: Fast stop mode	605A00	605A0010	0x7510	Integer16 RW	Fast stop mode 0: Uncontrolled stop 1: Stop with deceleration(0x6084),motor will unlock the shaft at zero speed 2: Stop with quick deceleration(0x6085), motor will unlock the shaft at zero speed 5：Stop with deceleration(0x6084),motor will lock the shaft at zero speed 6: Stop with quick deceleration(0x6085), motor will lock the shaft at zero speed
Shutdown stop mode	605B00	605B0010	0x7520	Integer16 RW	Shutdown stop mode（Control word changes from 0x0F to 0x06） 0: Uncontrolled stop 1: Stop with deceleration(0x6084),motor will unlock the shaft at zero speed 2: Stop with quick deceleration(0x6085), motor will unlock the shaft at zero speed
Inhibit	605C00	605C0010	0x7530	Integer16	Inhibit stop mode (Control

stop mode				RW	word changes from 0x0F to 0x07) 0: Uncontrolled stop 1: Stop with deceleration(0x6084),motor will unlock the shaft at zero speed 2: Stop with quick deceleration(0x6085), motor will unlock the shaft at zero speed
Pause mode	605D00	605D0010	0x7540	Integer16 RW	Pause mode 1: Stop with deceleration(0x6084),motor will lock the shaft at zero speed 2: Stop with quick deceleration(0x6085), motor will lock the shaft at zero speed
Alarm stop mode	605E00	605E0010	0x7550	Integer16 RW	Alarm emergency stop mode 0: Stop immediately 1: Stop with deceleration(0x6084),motor will unlock the shaft at zero speed 2: Stop with quick deceleration(0x6085), motor will unlock the shaft at zero speed

9.2 Unit Conversion

Note:

The 'N' in the unit conversion formulas represents the value before conversion.

When writing parameters, firstly you need to convert the parameter value to the internal unit (DEC), and then convert it to hexadecimal for writing.

The position unit does not require conversion.

For example:

Assuming the target speed is 500 rpm and the feedback resolution is 131072, then 500 rpm is converted to DEC as follows:

rpm -> DEC = N / 1875 * 512 * feedback resolution

DEC = 500 / 1875 * 512 * 131072 * 1000 / 1000 = 17895697

Converted to hexadecimal it is 0111 1111, then write the hexadecimal number into the

corresponding address of the target speed.

9.2.1 Current Conversion Formula

For SVD4812 driver, the maximum current (DEC) = 450
For SVD4822 driver, the maximum current (DEC) = 100
For SVD4835 driver, the maximim current (DEC) =160

DEC -> Ap = N / 2048 * [maximum current (DEC)] / 10
Ap -> DEC = N * 2048 / [maximum current (DEC)] * 10
DEC -> Arms = N / 2048 * [maximum current (DEC)] / 10 / 1.414
Arms -> DEC = N * 2048 / [maximum current (DEC)] * 10 * 1.414

9.2.2 Speed Conversion Formula

DEC -> rpm = N * 1875 / 512 / feedback resolution
rpm -> DEC = N / 1875 * 512 * feedback resolution

9.2.3 Acceleration (Deceleration) Conversion Formula

DEC -> rps/s = N * 1000 * 4000 / 65536 / feedback resolution
rps/s -> DEC = N / 1000 / 4000 * 65536 * feedback resolution

9.2.4 Torque Conversion Formula

DEC->%=N/10
%->DEC=N*10

Chapter 10 UART Communication

This model of the driver only supports one-to-one mode for UART communication through the CN2 port. For details, please refer to Chapter 4 System Interface and Wiring.

10.1 UART Communication Format

The default baud rate is 115200bps, which can be modified through the index address 2F8003.

The UART communication parameters are: 115200, 8, N, 1

That is, the baud rate is fixed at 115200bps, the data bits are 8 bits, no parity check, and 1 stop bit.

UART Communication Protocol

This drive's UART communication follows a strict master-slave station protocol. The master station/supervisory controller sends one frame or multiple frames of data to drive, and drive will respond with one frame or multiple frames of corresponding data after receiving the correct data.

The UART communication protocol adopts a fixed ten-byte format:

Device ID	Command	Object address			Data area				Check
Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	Byte10
Device ID	Command	Index		Sub-index	Low——>High Such as 0x01 means 01 00 00 00				Check

Device ID: Set by the DIP switch, for details, refer to Chapter 4 (universal station number is 127).

Command: Take different values according to the user's purpose, which can refer to the list below.

Object address: Known through the object list, where the index (Byte3-Byte4) is also from low to high.

Data area: Used to store the sent data or the received data.

If the four bytes of the data area are 0F 65 77 83, then the correct HEX format is 83 77 65 0F.

Check: The check bit, the value is the sum of the previous nine bytes taken inversely, and then get the lower byte of the result.

The specific definition of the command is as follows:

(1) Read command, the host reads the data of the relevant object address of drive

Host sends	Drive replies	Description
0x40	0x4f	The effective data of the object is Byte6
0x40	0x4b	The effective data of the object is Byte6-Byte7
0x40	0x43	The effective data of the object is Byte6-Byte9

0x40	0x80	Check error, the error word is included in Byte6-Byte9

(2) Write command, the host writes data to the relevant object address of drive.

Host sends	Drive replies	Description
0x23	0x60	The effective data of the object is Byte6-Byte9
	0x80	Error, Byte6-Byte9 contains error code
0x2b	0x60	The effective data of the object is Byte6-Byte7
	0x80	Error, Byte6-Byte9 contains error code
0x2f	0x60	The effective data of the object is Byte6
	0x80	Error, Byte6-Byte9 contains error code

10.2 UART Communication Examples

The following examples set Device ID 01 and Baud rate 115200 as default.

10.2.1 Speed Mode

Speed conversion formula
The value of the speed = N / 1875 * 512 * motor encoder resolution (feedback resolution), where N is the rotation speed (rpm)
Here, taking a 17-bit motor as an example, the motor encoder resolution = 131072
Target speed 10 rpm is converted to decimal as 357913, hexadecimal as 57619
Target speed -10 rpm is converted to decimal as -357913, hexadecimal as FFFA 89E7

Profile acceleration conversion formula
The value of the acceleration=N/1000/4000*65536*motor encoder resolution（feedback resolution）,where N is acceleration(rps/s).
Motor encoder resolution is as above.
Profile acceleration 50rps/s is converted to decimal as 107374,hexadecimal as 1A36E.

Object address	Name	Data Type	Setting Value	Message	Note
606000	Operation mode	Integer8	3	01 2F 60 60 00 03 00 00 00 0D	Operation mode is set as 3
60FF00	Target speed	Integer32	10rpm	01 23 FF 60 00 19 76 05 00 E9	Target speed is set as 10rpm
			-10rpm	01 23 FF 60 00 E7 89 FA FF 14	Target speed is set as -10rpm
608300	Profile	Unsigned	50rps/s	01 23 83 60 00 6E A3 01	Profile

	acceleration	32		00 E7	acceleration is set as 50rps/s
608400	Profile deceleration	Unsigned 32	50rps/s	01 23 84 60 00 6E A3 01 00 E6	Profile deceleration is set as 50rps/s
604000	Control word	Unsigned 16	F	01 2B 40 60 00 0F 00 00 00 25	Controlword is set as F to lock the motor axis

According to the above operations, the motor will accelerate to 10 rpm or -10 rpm with profile acceleration of 50 rps/s. If the target speed is written to 0 again, the motor will decelerate to 0 with profile deceleration of 50 rps/s.

10.2.2 Absolute Position Mode

The profile speed is converted in the same way as the speed mode.
50 rpm can be converted to decimal as 1789569, hexadecimal as 1B4E81.
Target position 10000 inc is converted to hexadecimal as 2710.
If it is -10000 inc, it is converted to hexadecimal as FFFFD8F0.

Object address	Name	Data Type	Setting Value	Message	Note
606000	Operation mode	Integer8	1	01 2F 60 60 00 01 00 00 00 0F	Operation mode is set as 1
608100	Profile speed	Unsigned32	50 rpm	01 23 81 60 00 81 4E 1B 00 11	Profile speed is set as 50 rpm
607A00	Target position	Integer32	10000 inc	01 23 7A 60 00 10 27 00 00 CB	Target position is set as 10000 inc
			-10000 inc	01 23 7A 60 00 F0 D8 FF FF 3C	Target position is set as -10000 inc
604000	Control word	Unsigned16	0x2F->0x3F	01 2B 40 60 00 2F 00 00 00 05 01 2B 40 60 00 3F 00 00	Control word is set ans 2F and then

				00 F5	changed to 3F

According to the above operations, the motor will run to the position of 10000 inc or -10000 inc at a speed of 50 rpm.

10.2.3 Relative Position Mode

The profile speed is converted in the same way as the speed mode.
50 rpm can be converted to decimal as 1789569, hexadecimal as 1B4E81.
Target position 10000 inc is converted to hexadecimal as 2710.
If it is -10000 inc, it is converted to hexadecimal as FFFFD8F0.

Object address	Name	Data Type	Setting Value	Message	Note
606000	Operation mode	Integer8	1	012F60 600001 00 00 000F	Operation mode is set as 1
608100	Profile speed	Unsigned32	50rpm	012381 600081 4E 1B 0011	Profile speed is set as 50 rpm
607A00	Target position	Integer32	10000 inc	01237A 600010 27 00 00CB	Target position is set as 10000 inc
			-10000inc	01 23 7A 60 00 F0 D8 FF FF 3C	Target position is set as -10000 inc
604000	Control word	Unsigned16	0x4F->0x5F	01 2B 40 60 00 4F 00 00 00 E5 01 2B 40 60 00 5F 00 00 00 D5	Control word is set ans 4F and then changed to 5F

According to the above operations, the motor will move 10000 inc on the original position at a speed of 50 rpm in the positive direction or the negative direction of 10000 inc.
The relative position mode and the absolute position mode have similar operation methods. The relative position mode moves 10000 inc on the original position, while the absolute position mode moves the motor to the 10000 inc position.

10.2.4 Torque Mode

Torque conversion formula
The value filled in (decimal) = N * 10
If set to 10%, after conversion it becomes 100, hexadecimal is 64
The MAX_Speed_Limit is converted in the same way as the speed mode.
5000 rpm can be converted to decimal as 178,956,970, hexadecimal as AAAAAAA

Object address	Name	Data Type	Setting Value	Message	Note
606000	Operation mode	Integer8	4	01 2F 60 60 00 04 00 00 00 0C	Operation mode is set as 4
607100	Target_Torque%	Unsigned32	10	01 2B 71 60 00 64 00 00 00 9F	Target_Torque% is set as 10%（10% of rated torque）
607F00	MAX_Speed_Limit	Unsigned32	5000 rpm	01 23 7F 60 00 AA AA AA 0A 58	MAX_Speed_Limit is set as 5000rpm
604000	Control word	Unsigned16	F	01 2B 40 60 00 0F 00 00 00 25	Control word is set as F

According to the above operations, the motor will operate at 10% of the rated torque.

10.2.5 Homing Mode

Homing_Speed_Switch and Homing_Speed_Zero is converted in the same way as the speed mode. 50 rpm can be converted to decimal as 1,789,569, hexadecimal as 1B4E81

The Homing Method 17 is converted to hexadecimal as 11

Object address	Name	Data Type	Setting Value	Message	Note
606000	Opeation mode	Integer8	6	01 2F 60 60 00 06 00 00 00 0A	Operation mode is set as 6
609901	Homing_Speed_Switch	Unsigned32	50rpm	01 23 99 60 01 81 4E 1B 00 F8	Homing_Speed_Switch is set as 50rpm
609902	Homing	Unsigned32	50rpm	01 23 99 60 02 81 4E 1B	Homing_Sp

	_Speed _Zero			00 F7	eed_Zero is set as 50rpm
609800	Homing Method	Integer8	17	01 2F 98 60 00 11 00 00 00 C7	Homing Method is set as 17
604000	Control word	Unsigned16	F> 1F	01 2B 40 60 00 0F 00 00 00 25 01 2B 40 60 00 1F 00 00 00 15	Control word is set ans F and then changed to 1F

According to the above operations, the motor will search for the home position according to the homing method 17, with homing_speed_switch of 50 rpm and homing_speed_zero of 50 rpm.

Note:

Pay attention to the unit conversion, and detailed can refer to unit conversion behind Chapter 6 Common Object List.

Communication error code table

Serial number	Error code	Description
1	0x05040001	Invalid command
2	0x06010001	Write-only parameter
3	0x06010002	Read-only parameter
4	0x06020000	Invalid index
5	0x06040041	Unable mapping
6	0x06060000	Device hardware failure
7	0x06070010	Data length error
8	0x06070013	Object data is too long
9	0x06070013	Object data is too short
10	0x06090011	Invalid sub-index
11	0x06090030	Invalid value
12	0x06090031	Value is too high
13	0x06090032	Value is too low
14	0x08000000	General error
15	0x08000021	Incorrect control word
16	0x08000022	Incorrect status word
17	0x08000023	No object dictionary

Chapter 11 RS485 Communication

11.1 RS485 Communication Hardware Introduction

This model of the driver uses the CNA1(IN) port and CNB1(OUT) for RS485 communication. For specific details, please refer to Chapter 4 System Interface and Wiring.

11.2 RS485 Communication Format

The RS485 baud rate of this machine can be set through the AMPS software, or modified through the command.

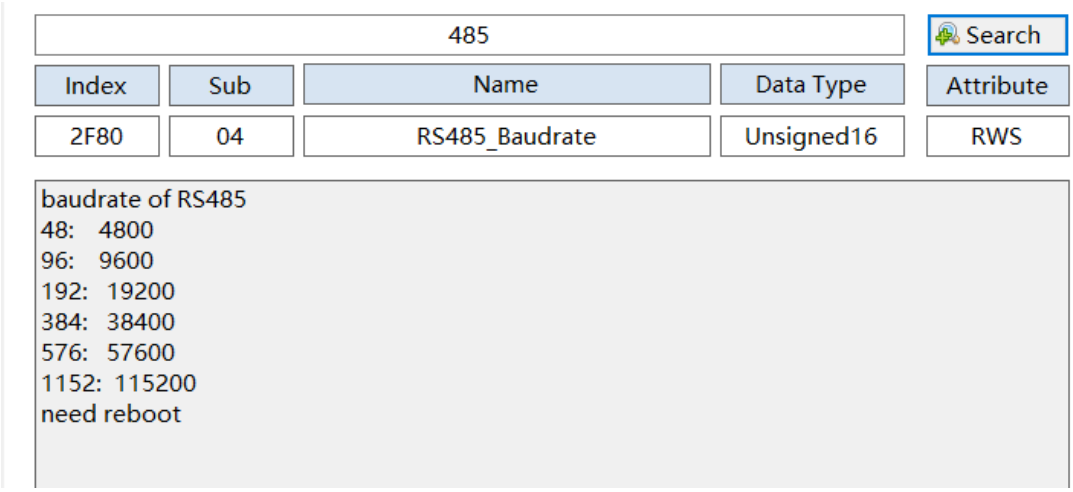


Figure 11-1 RS485 Baud Rate Related Settings

11.2.1 Communication Protocol

This model supports the Modbus RTU communication protocol, and its internal objects are discontinuous 16-bit data registers. The message format is as follows:

Station No.	Function code	Data area	Check bit	
Byte1	Byte2	ByteN	Byte(N+1)	Byte(N+2)

Station No.: The RS485 address of drive, determined by the SW1-SW4 of the drive's DIP switch S1, for details, refer to Chapter 4 (universal station No. is 127).
Function code: Take different values according to the user's purpose, which can refer to the list below.
Data area: Stores the index number and operation-related data.
Check bit: Used for verification during communication, this model of the driver uses Modbus CRC16 check by default.

11.2.2 Common Function Introduction

Read Data Register (0x03)

Send message:

Device ID	Function code	Modbus address		Number of registers to read		CRC
		High byte	Low byte	High byte	Low byte	
1 byte	03	1 byte	1 byte	1 byte	1 byte	2 byte

Reply message:

Device ID	Function code	Number of bytes returned	Register data		CRC
			High byte	Low byte	
1 byte	03	1 byte	1 byte	1 byte	2 byte

Note:

If there is a response error such as an address does not exist, the function code returned is 0x81.

Write Single Data Register (0x06)

Device ID	Function code	Modbus address		Number of registers to read		CRC
		High byte	Low byte	High byte	Low byte	
1 byte	06	1 byte	1 byte	1 byte	1 byte	2 byte

Note:

If the object is written successfully, the original message is returned.

Write Multiple Holding Registers (0x10)

Device ID	Function code	Modbus address	Data length (word)		Number of bytes to write	Low data		High data		CRC
			High byte	Low byte		High byte	Low byte	High byte	Low byte	
1 byte	10	2 byte	1byte	1 byte	1 byte	1 byte	1 byte	1 byte	1 byte	2 byte

Reply message:

Device ID	Function code	Modbus address	Data length (word)		CRC
			High byte	Low byte	
1 byte	10	2 byte	1byte	1 byte	2 byte

Note:

If there is an illegal operation such as writing to a non-existent address or writing to read-only data, the function code returned is 0x90.

11.3 RS485 Communication Examples

The following examples take station number 1 as an example.

For details on the conversion of parameters such as target speed, please refer to the UART communication example or Chapter 6.

11.3.1 Read Actual Position and Response Message

The Modbus address for the actual position is 0x7630
Then, the message sent to read the actual position is:
01 03 76 30 00 02 DE 4C
The response message is:
01 03 04 2B 60 71 26 56 43

01 03 76 30 00 02 DE 4C
01: Station No.
03：Function code
76 30：Modbus address of actual position
00 02：Number of registers to read，2word=32bit
DE 4C：Modbus CRC16 check bit

01 03 04 2B 60 71 26 56 43
01：Station No.
03：Function code
04：Number of bytes returned
2B 60 71 26: The returned data, converted to hexadecimal is 7126 2B60, converted to decimal is 1,898,326,880, that is, the actual position is 1,898,326,880 inc
56 43：Modbus CRC16 check bit

11.3.2 Speed Mode

The value of the speed (decimal number)=N/1875*512* motor encoder resolution (feedback resolution), where N is the rotation speed (rpm)
Here, taking a 17-bit motor as an example, the motor encoder resolution is 131072
Target speed is 100 rpm, after conversion, the decimal is 3,579,139, and the hexadecimal is 36 9D03

Modbus address	Name	Data type	Setting value	Message	Description
0x7600	Operation mode	Integer8	3	01 06 76 00 00 03 D3 83	Operation mode is set as 3
0x8C00	Target speed	Integer32	100rpm	01 10 8C 00 00 02 04 9D 03 00 36 98 D3	Target speed is set as 100rpm
0x7930	Profile acceleration	Unsigned32	50rps/s	01 10 79 30 00 02 04 A3 6E 00 01 7D 70	Profile acceleration is set as 50rps/s

0x7940	Profile deceleration	Unsigned32	50rps/s	01 10 79 40 00 02 04 A3 6E 00 01 7A 54	Profile deceleration is set as 50rps/s
0x7400	Control word	Unsigned16	F	01 06 74 00 00 0F D2 3E	Controlword is set as F to lock the motor axis

01 06 76 00 00 03 D3 83
01: Station No.
06：Write signal data regsiter
76 00：RS485 address for operation mode
00 03：Set operation mode as 3
D3 83:Modbus CRC16 check bit

01 10 8C 00 00 02 04 9D 03 00 36 98 D3
01: Station No.
10：Write multiple holding regsiter
8C 00:RS485 address for target speed
02：Write data length of 2 WORD(4 Bytes)
04：Writie 4 bytes of data
9D 03 00 36：The data to write into register,converted to hexadecimal is 369D03,converted to decimal is 3579139,corresponding to speed 100rpm.More details of conversion formula refer to UART communication example or object list.

Profile speed,profile deceleration and controlword are similar to example above.

11.3.3 Absolute Position Mode

50 rpm can be converted to decimal as 1,789,569, and hexadecimal as 1B 4FAD.

Modbus address	Name	Data type	Setting value	Message	Description
0x7600	Operation mode	Integer8	1	01 06 76 00 00 01 52 42	Operation mode is set as 1
0x7910	Profile speed	Unsigned32	50rpm	01 10 79 10 00 02 04 4F AD 00 1B 3A 0F	Profile speed is set as 50 rpm

0x77A0	Target position	Integer32	10000 inc	01 10 77 A0 00 02 04 27 10 00 00 B1 54	Target position is set as 10000 inc
0x7400	Control word	Unsigned16	0x2F-> 0x3F	01 06 74 00 00 2F D3 E6 01 06 74 00 00 3F D2 2A	Control word is set ans 2F and then changed to 3F

The absolute position mode is similar to the relative position mode in operation, and is not introduced here.

The proile speed is calculated in the same way as the above speed mode, and here we introduce the calculation of the target position:

If the target position is 10000 inc, it can be directly converted to hexadecimal as 2710, without other conversions.

11.3.4 Torque Mode

Torque conversion formula

The value filled in (decimal) = N * 10

If set to 10%, after conversion it becomes 100, hexadecimal is 64

The MAX_Speed_Limit is converted in the same way as the speed mode.

5000 rpm can be converted to decimal as 178,956,970, hexadecimal as AAAAAAA

Modbus address	Name	Data Type	Setting Value	Message	Description
0x7600	Operationmode	Integer8	4	01 06 76 00 00 04 92 41	Operation mode is set as 4
0x7710	Target_Torque%	Unsigned32	10	01 06 77 10 00 64 92 50	Target_Torque% is set as 10%（10% of rated torque）
0x77F0	MAX_Speed_Limit	Unsigned32	5000 rpm	01 10 77 F0 00 02 04 AA AA 0A AA 38 3E	MAX_Speed_Limit is set as 5000rpm
0x7400	Control word	Unsigned16	F	01 06 74 00 00 0F D2 3E	Control word is set as F

According to the above operations, the motor will operate at 10% of the rated torque.

11.3.5 Homing Mode

50 rpm can be converted to decimal as 1,789,569, hexadecimal as 1B4FAD

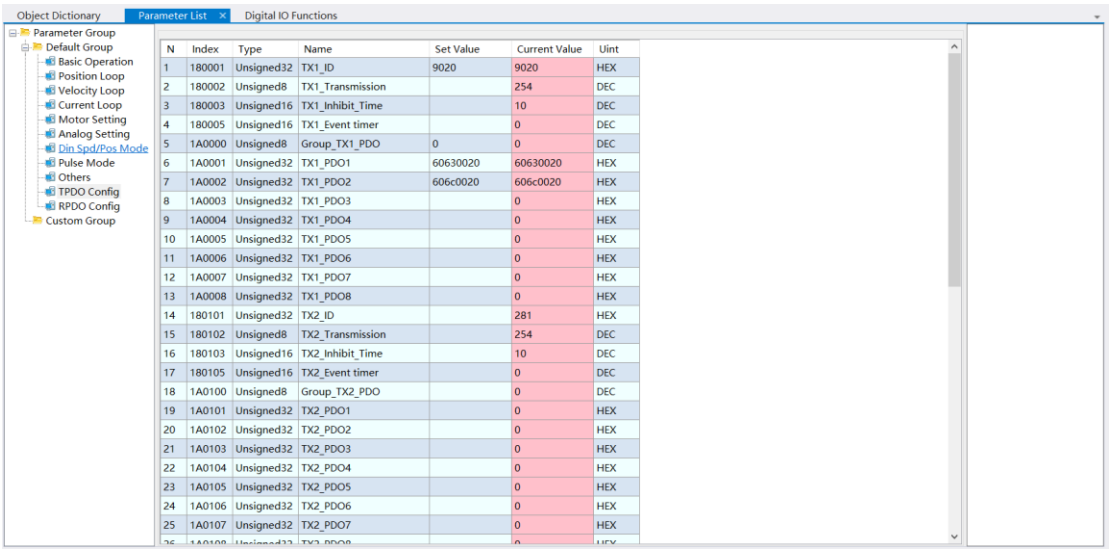
Modbus address	Name	Data Type	Setting Value	Message	Description
0x7600	Opeation mode	Integer8	6	01 06 76 00 00 06 13 80	Operation mode is set as 6
0x7D01	Homing_Speed_Switch	Unsigned32	50rpm	01 10 7D 01 00 02 04 4F AD 00 1B C8 0F	Homing_Speed_Switch is set as 50rpm
0x7D02	Homing_Speed_Zero	Unsigned32	50rpm	01 10 7D 02 00 02 04 4F AD 00 1B 88 1A	Homing_Speed_Zero is set as 50rpm
0x7C00	Homing Method	Integer8	17	01 06 7C 00 00 11 50 56	Homing Method is set as 17
0x7400	Control word	Unsigned16	F->1F	01 06 74 00 00 1F D3 F2	Control word is set ans F and then changed to 1F

According to the above operations, the motor will search for the home position according to the homing method 17, with homing_speed_switch of 50 rpm and homing_speed_zero of 50 rpm.

11.3.6 RS485-PDO Mode(Read Multiple Parameters)

This mode is applicable to this model, for other models, refer to the corresponding manual for details. The specific steps are as follows:

Step 1: Set the TPDO1 station number to a specified address (which should not conflict with the Modbus address already used), and at the same time, set the TPDO1 mapping to the index address that needs to be returned (the same address as the CAN communication).
Following figure is the example of mapping to actual position(60630020) and actual speed(606C0020).



N	Index	Type	Name	Set Value	Current Value	Uint
1	180001	Unsigned32	TX1_ID	9020	9020	HEX
2	180002	Unsigned8	TX1_Transmission		254	DEC
3	180003	Unsigned16	TX1_Inhibit_Time		10	DEC
4	180005	Unsigned16	TX1_Event timer		0	DEC
5	1A0000	Unsigned8	Group_TX1_PDO	0	0	DEC
6	1A0001	Unsigned32	TX1_PDO1	60630020	60630020	HEX
7	1A0002	Unsigned32	TX1_PDO2	606c0020	606c0020	HEX
8	1A0003	Unsigned32	TX1_PDO3		0	HEX
9	1A0004	Unsigned32	TX1_PDO4		0	HEX
10	1A0005	Unsigned32	TX1_PDO5		0	HEX
11	1A0006	Unsigned32	TX1_PDO6		0	HEX
12	1A0007	Unsigned32	TX1_PDO7		0	HEX
13	1A0008	Unsigned32	TX1_PDO8		0	HEX
14	180101	Unsigned32	TX2_ID		281	HEX
15	180102	Unsigned8	TX2_Transmission		254	DEC
16	180103	Unsigned16	TX2_Inhibit_Time		10	DEC
17	180105	Unsigned16	TX2_Event timer		0	DEC
18	1A0100	Unsigned8	Group_TX2_PDO		0	DEC
19	1A0101	Unsigned32	TX2_PDO1		0	HEX
20	1A0102	Unsigned32	TX2_PDO2		0	HEX
21	1A0103	Unsigned32	TX2_PDO3		0	HEX
22	1A0104	Unsigned32	TX2_PDO4		0	HEX
23	1A0105	Unsigned32	TX2_PDO5		0	HEX
24	1A0106	Unsigned32	TX2_PDO6		0	HEX
25	1A0107	Unsigned32	TX2_PDO7		0	HEX

Figure 11-2 Reading Multiple Parameter

Step 2: Send the message, the format is as follows:

Device ID	Function code	Object Modbus address		Number of bytes to read		CRC
		High byte	Low byte	High byte	Low byte	
1 byte	03	1 byte	1 byte	1 byte	1 byte	2 byte

Here we send the message: 01 03 90 20 00 04 68 C3
Response message: 01 03 08 29 DC 04 12 78 80 00 1D 2A AE

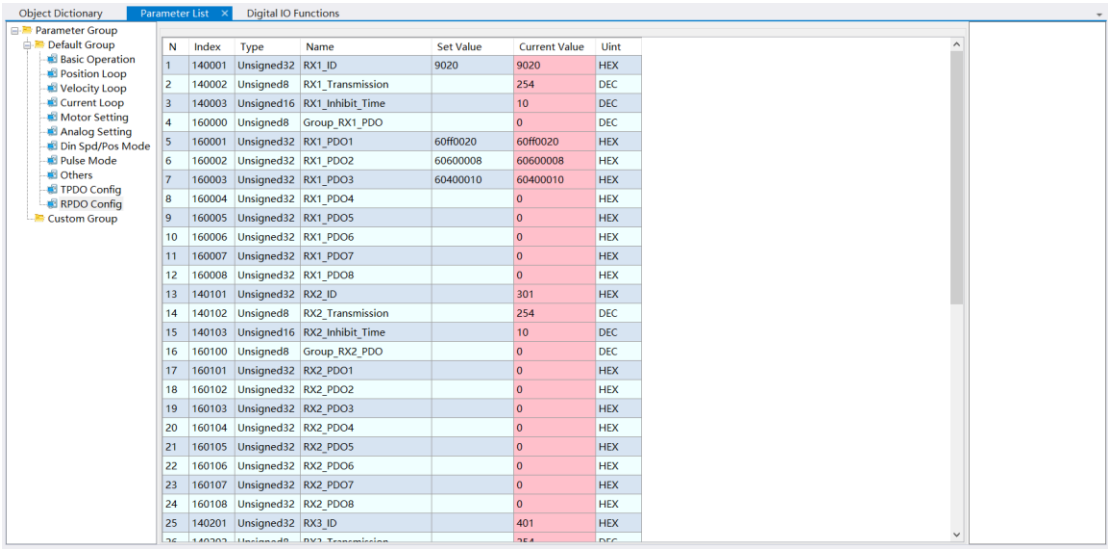
01 03 90 20 00 04 68 C3
01：Station No.
03：Function code- read
90 20：Address in RS485-PDO mode
00 04：Number of registers to read
68 C3：Check bit

01 03 08 29 DC 04 12 78 80 00 1D 2A AE
01：Station No.
03：Function code- read
08：Number of bytes to read
29 DC 04 12：Indicates the actual position (TPDO mapping 1), the hexadecimal number is 412 29DC, and the decimal number is 68 299 228, that is, the current position is 68 299 228
78 80 00 1D：Indicates the actual speed (TPDO mapping 2), the hexadecimal number is 1D 7880, and the decimal number is 1 931 392

11.3.7 RS485-PDO Mode (Write Multiple Parameters)

The specific operation steps are the same as above. Here we take the example of writing the speed mode to the driver.

As shown in the figure below, it is mapping to target speed (60FF0020), working mode (60600008), and control word (60400010).



N	Index	Type	Name	Set Value	Current Value	Uint
1	140001	Unsigned32	RX1_ID	9020	9020	HEX
2	140002	Unsigned8	RX1_Transmission		254	DEC
3	140003	Unsigned16	RX1_Inhibit_Time		10	DEC
4	160000	Unsigned8	Group_RX1_PDO		0	DEC
5	160001	Unsigned32	RX1_PDO1	60ff0020	60ff0020	HEX
6	160002	Unsigned32	RX1_PDO2	60600008	60600008	HEX
7	160003	Unsigned32	RX1_PDO3	60400010	60400010	HEX
8	160004	Unsigned32	RX1_PDO4		0	HEX
9	160005	Unsigned32	RX1_PDO5		0	HEX
10	160006	Unsigned32	RX1_PDO6		0	HEX
11	160007	Unsigned32	RX1_PDO7		0	HEX
12	160008	Unsigned32	RX1_PDO8		0	HEX
13	140101	Unsigned32	RX2_ID		301	HEX
14	140102	Unsigned8	RX2_Transmission		254	DEC
15	140103	Unsigned16	RX2_Inhibit_Time		10	DEC
16	160100	Unsigned8	Group_RX2_PDO		0	DEC
17	160101	Unsigned32	RX2_PDO1		0	HEX
18	160102	Unsigned32	RX2_PDO2		0	HEX
19	160103	Unsigned32	RX2_PDO3		0	HEX
20	160104	Unsigned32	RX2_PDO4		0	HEX
21	160105	Unsigned32	RX2_PDO5		0	HEX
22	160106	Unsigned32	RX2_PDO6		0	HEX
23	160107	Unsigned32	RX2_PDO7		0	HEX
24	160108	Unsigned32	RX2_PDO8		0	HEX
25	140201	Unsigned32	RX3_ID		401	HEX

Figure 11-3 Write Multiple Parameters

After setting,

Send message: 01 10 90 20 00 04 08 9D 03 00 36 00 03 00 0F 75 61

01：Station No.

10：Write multiple holding registers

90 20：Address in RS485-PDO mode

00 04：The modified content is 4 words, the target speed is 4 bytes, the operation mode is 2 bytes (only one byte originally, here we fill the high byte with zero), and the control word is 2 bytes

08：The modified content is 8 bytes (the reason why it is not 7 bytes is described above)

9D 03 00 36: The content of the target speed modification, hexadecimal is 0036 9D03, converted to decimal is 100 rpm.

00 03：The content of the working mode modification, the operation mode is set to 3

00 0F：The content of the control word modification, write F to the control word

75 61：Check bit

Note:

If you want to read multiple parameters, you should switch to TPDO (Transmit Process Data Object);

If you want to write multiple parameters, you should switch to RPDO (Receive Process Data Object).

Chapter 12 CANopen Communication

CANopen (Controller Area Network Open) is a communication protocol based on the CAN bus, which is used to achieve data exchange and communication between different devices in industrial automation and control systems. It provides a set of standardized communication and device description methods, enabling various devices to work collaboratively on the same network.

The following is a brief introduction to CANopen communication:

Communication medium: CANopen uses the CAN bus as the communication medium. The CAN bus is a highly reliable serial communication protocol commonly used in industrial environments, characterized by strong anti-interference capabilities and good real-time performance.

Device types: CANopen communication can be used to connect various types of devices, such as motor drivers, sensors, controllers, HMI (Human-Machine Interface), etc. These devices can be produced by different manufacturers as long as they comply with the communication methods and data structures specified by the CANopen communication protocol.

Communication protocol: CANopen defines a set of communication protocols, including data frame formats, communication objects, object dictionaries, PDO (Process Data Object), SDO (Service Data Object), etc. These protocols specify the data exchange methods and communication processes between devices.

Object dictionary: CANopen uses an object dictionary to describe the parameters, status, and functions of the device. The object dictionary contains a series of indexes and sub-indexes, each corresponding to a specific parameter or function. Devices can exchange data and configure by reading and writing the object dictionary.

PDO and SDO: PDO is used for real-time data transmission, while SDO is used for non-real-time data transmission and configuration. PDO can efficiently transmit real-time data between devices, while SDO is used for configuring and managing device parameters.

Node: Each device in the CANopen network is called a node. Each node has a unique node ID, which is used to identify and address in the network.

The following table lists the CANopen object dictionary with explanations:

Parameter name	CANopne address			Attribute
	Index	Sub-index	Data length	
Operation mode	6060	00	08	RW
Control word	6040	00	10	RW
Target speed	60FF	00	20	RW

Here, only the data length and attribute meanings are explained.

Data length: 08 - Data length is 1 byte

10 - Data length is 2 bytes

20 - Data length is 4 bytes

Attribute: R - Readable;

W - Writable;
M - Mappable
S - Storable (non-volatile)

12.1 Hardware Description

This model of the driver supports one-to-multiple mode. If the CAN communication method is used, the default is to use CAN1A(IN) for input and CAN1B(OUT) for output. For details, please refer to Chapter 4 System Interface and Wiring.

12.2 CAN Communication

12.2.1 EDS

EDS stands for "Electronic Data Sheet." It is an electronic data sheet used to describe the communication parameters, object dictionary, mapping information, and functions of CANopen devices in detail. EDS files are usually written in XML format, used to configure and identify CANopen devices, and ensure correct communication between devices.
In the CANopen network, each device has an object dictionary that stores the device's parameters, status, and function information. The EDS file describes the structure and content of this object dictionary, as well as the device's communication parameters, PDO mapping, and other information. By reading the device's EDS file, users can understand the device's functions and communication parameters, thereby correctly configuring and integrating the device into the CANopen network.
The content of the EDS file usually includes:
Device description information, such as device name, manufacturer information, etc.
Communication parameters, such as node ID, baud rate, etc.
The structure and content of the object dictionary, including object index, sub-index, data type, and access permissions.
PDO mapping information, describing the mapping and transmission method of data in PDO communication.
By using the EDS file, users can conveniently configure and integrate various different CANopen devices, ensuring that devices can communicate and exchange data correctly.

12.2.2 SDO

(1) SDO Introduction

In the CANopen communication protocol, SDO stands for "Service Data Object." It is a communication mechanism used for parameter configuration, status query, and data exchange in the CANopen network. The SDO mechanism allows point-to-point communication between the master (Master) device and the slave (Slave) device to read or write data in the object dictionary of the slave device. The object dictionary is an important concept in the CANopen protocol, which is a data structure used to store the device's parameters, status, and function information. Through SDO communication, the

master can send requests to the slave to read or write data in specific object dictionaries.
The basic process of SDO communication is as follows:
The master sends an SDO request frame, which includes information such as the node ID of the slave device to be accessed, the object dictionary index and sub-index to be read or written, etc.
The slave receives the SDO request and performs the corresponding operation, such as reading or writing data in the object dictionary.
The slave encapsulates the operation result in the SDO response frame and sends it back to the master.
The master receives the SDO response and parses the data to complete the read or write operation.

The local CAN interface supports the CANopen SDO data transfer protocol. SDO is primarily used for transferring low-priority objects between devices. This type of data transfer is similar to the MODBUS method, where after the master station sends out a request, the slave station needs to respond with data: Client→Server/Server→Client. The basic structure of SDO is shown in the table below:

SDO Command specifier	Object index	Object sub-index	Up to 4 bytes of data
Byte0	Byte1-Byte2	Byte3	Byte4-Byte7

The 2 bytes of the object index and the 4 bytes of the data object are both arranged in little-endian format, meaning the least significant byte comes first and the most significant byte comes last. For example, if the object index is 0x606C, then Byte1 = 6C and Byte2 = 60.

(2) SDO Read Parameter

When reading parameters, the SDO message format is as follows:

Identifier	DLC	Data							
		0	1	2	3	4	5	6	7
0x600+Node_ID	8	Send Command	Object index		Object sub-index	00			

When receiving the SDO message, the format is:

Identifier	DLC	Data							
		0	1	2	3	4	5	6	7
0x580+Node_ID	8	Receive Command	Object index		Object sub-index	Up to 4 bytes of data			

The following are the send and receive command for reading parameters:

Send Command	Receive Command	Description
0x40	0x43	The received data is 4 bytes

	0x4B	The received data is 2 bytes
	0x4F	The received data is 1 bytes
	0x80	There is an error in the received data

(3) SDO Modify Parameter

When modifying parameters, the SDO message format is as follows:

Identifier	DLC	Data							
		0	1	2	3	4	5	6	7
0x600+Node_ID	8	Send Command	Object index		Object sub-index	Up to 4 bytes of data			

When receiving the SDO message for modifying parameters, the format is:

Identifier	DLC	Data							
		0	1	2	3	4	5	6	7
0x580+Node_ID	8	Receive Command	Object index		Object sub-index	Up to 4 bytes of data			

The following are the send and receive command words for modifying parameters:

Send Command	Receive Command	Description
0x23	0x60	Successfully sent a message with 4 bytes of data
0x23	0x80	SDO message transmission failed
0x2B	0x60	Successfully sent a message with 2 bytes of data
0x2B	0x80	SDO message transmission failed
0x2F	0x60	Successfully sent a message with 1 bytes of data
0x2F	0x80	SDO message transmission failed

(4) SDO Communication Error Code Table

Serial number	Error code	Description
1	0x05040001	Invalid command
2	0x06010001	Write-only parameter
3	0x06010002	Read-only parameter
4	0x06020000	Invalid index
5	0x06040041	Unable mapping
6	0x06060000	Device hardware failure
7	0x06070010	Data length error
8	0x06070013	Object data is too long
9	0x06070013	Object data is too short
10	0x06090011	Invalid sub-index

11	0x06090030	Invalid value
12	0x06090031	Value is too high
13	0x06090032	Value is too low
14	0x08000000	General error
15	0x08000021	Incorrect control word
16	0x08000022	Incorrect status word
17	0x08000023	No object dictionary

12.2.3 PDO

(1) PDO Introduction

In the CANopen communication protocol, PDO stands for "Process Data Object." It is a communication mechanism used for real-time data exchange, allowing for the periodic transmission of data between CANopen devices. PDO communication allows devices in the CANopen network to exchange data periodically, typically used for transmitting real-time control, monitoring, and feedback data. Unlike SDO (Service Data Object), PDO communication is point-to-multipoint, allowing one master station to broadcast data to multiple slave stations.

The basic process of PDO communication is as follows:
The master station configures the PDO communication parameters, including the period, mapping of the object dictionary, etc.
The slave station periodically sends PDO data frames to the network according to the parameters set by the master station, which contain real-time data.
Other devices receive the PDO data frames and can parse the data and respond accordingly.

The features of PDO communication include:
Real-time: PDO communication is a real-time communication mechanism based on the CAN bus, suitable for applications that require rapid transmission of real-time data.
Efficiency: Since PDO communication is periodic broadcasting, it can quickly convey data across the network, suitable for real-time control and monitoring.
Predefined mapping: PDO communication requires pre-configured mapping of the object dictionary, which clarifies the data transmission format and period.

(2) PDO COB-ID

In the CANopen communication protocol, PDO COB-ID stands for "Process Data Object Communication Object Identifier." It is a unique identifier used to identify and distinguish different PDO communication objects. PDO communication is used for real-time data exchange, allowing CANopen devices to transmit data periodically. Each PDO communication object has a unique PDO COB-ID to identify the data frame on the CAN bus.

The composition of PDO COB-ID includes the following parts:

11-bit or 29-bit identifier: Identifies the communication object on the CAN bus, 11-bit identifier for standard CANopen networks, and 29-bit identifier for extended CANopen networks. (This model only uses the 11-bit identifier, i.e., only standard frames are used)
Node ID (Node ID): Specifies the sending or receiving node of the PDO. This is a unique identifier in the CANopen network, used to determine the source and destination of the data flow.
COB type: Indicates the type of the data frame, such as TPDO (transmission) or RPDO (reception), detailed content is introduced below.

10	9	8	7	6	5	4	3	2	1	0
Function Code				Node-ID						

Function Code：he data transmission function code defines the transmission level of various messages, the smaller the function code, the higher the priority.
Node-ID：The device station number, the range is 1 to 127

Predefined master/slave connection set

Object	COB-ID
NMT Module Control	000H
SYNC 080H	080H
TIME SSTAMP	100H
Object	COB-ID
Emergency	081H-0FFH
TPDO1（transmit）	181H-1FFH
RPDO1（receive）	201H-27FH
TPDO2（transmit）	281H-2FFH
RPDO2（receive）	301H-37FH
TPDO3（transmit）	381H-3FFH
RPDO3（receive）	401H-47FH
TPDO4（transmit）	481H-4FFH
RPDO4（receive）	501H-57FH
SDO（transmit /server）	581H-5FFH
SDO（receive/client）	601H-67FH
NMT Error Control	701H-77FH

Note:
The smaller the COB-ID, the higher the priority.
The function code in front of each level of COB-ID is a fixed format.
COB-ID 00H, 80H, 100H, 701H-77FH, 081H-0FFH are all system management formats.

Through PDO COB-ID, CANopen devices can identify and transmit real-time data. The master station can configure the COB-ID of each PDO communication object to achieve real-time data exchange between different devices.

(3) Sending PDO(TPDO)

Through PDO, the master station (controller) can send real-time data to the slave station (device). The function code (COB-ID) for sending PDO is:
0x180 + Servo station number
0x280 + Servo station number
0x380 + Servo station number
0x480 + Servo station number

(4) Receiving PDO(RPDO)

Through PDO, the slave station (device) can send real-time data to the master station (controller). The function code (COB-ID) for sending PDO is:
0x200 + Servo station number
0x300 + Servo station number
0x400 + Servo station number
0x500 + Servo station number

(5)PDO Transmission Mode

In the CANopen communication protocol, PDO (Process Data Object) has two main types of transmission: synchronous transmission and asynchronous transmission.

Synchronous transmission: Synchronous transmission is triggered by a synchronous message. The transmission type range is 0 to 240. In this mode, the master station must have the ability to send synchronous messages at a fixed frequency (up to 1 kHz). Once the driver receives the synchronous message, it will send the PDO data immediately after receiving. Synchronous transmission can be divided into two ways:
Non-periodic synchronous transmission: In this mode, PDO data can be pre-triggered by a remote frame or a specific event defined in the device sub-protocol. The driver will send PDO data once immediately after receiving the synchronous message.
Periodic synchronous transmission: In this mode, the transmission of PDO data is triggered after receiving 1 to 240 SYNC messages. The driver will send PDO data once after receiving a certain number of synchronous messages.
Synchronous message format

COB-ID	DLC
0x80	0

Asynchronous transmission: The transmission type for asynchronous transmission is 254 or 255. In asynchronous transmission, when the data of the slave station message changes, the slave station will immediately send PDO data regardless of whether the master station requests it. In addition, the time interval between two transmissions of the same message can be defined to avoid the situation where some high-priority messages always occupy the bus. In asynchronous transmission, the lower the value of PDO, the higher the priority. Asynchronous transmission can be summarized as follows:

Event-driven immediate transmission function: The slave station sends a message immediately after the data changes, regardlessof whether the master station inquires, and the time interval between two transmissions of the same message can be defined to avoid high-priority messages always occupying the bus (the lower the value of PDO, the higher the priority).
Event time periodic reporting function: Set the event time, and the driver will periodically upload data to the controller.

(6) PDO Inhibit Time

In the CANopen communication protocol, PDO Inhibit Time is an important parameter used to control the transmission interval of PDO data frames. The inhibit time refers to the period during which the transmission of PDO is prohibited after a PDO data frame has been sent to prevent too frequent data transmission.

Specifically, after a PDO data frame is sent, the PDO inhibit time starts, preventing the transmission of the same PDO data frame again within a certain period. This time interval can be set to a fixed value to ensure that data transmission is not too dense. The setting of the inhibit time can avoid network congestion and data conflicts, thereby improving the reliability and efficiency of communication.

In the CANopen network, each PDO communication object can configure its own inhibit time to adapt to different application scenarios and needs. By reasonably setting the inhibit time, stable transmission of PDO data can be ensured, avoiding data loss and conflicts, thus achieving reliable real-time data exchange and communication.

(7) Heartbeat Messages and Node Guarding

Heartbeat messages and node guarding are designed to improve the reliability and stability of the CANopen network. Heartbeat messages are used to monitor the status of nodes in real-time, while node guarding takes preventive measures when exceptions occur to protect the safety and normal operation of nodes and the entire communication network.

Heartbeat messages: Slaves periodically send heartbeat messages to the master. If the master does not receive the next heartbeat message within a set period, it will be considered that the slave may have a fault.

Heartbeat message format - (0x700 + node number) + status
Status - 0: Start, 4: Stop, 5: Run, 127: Pre-operation

Node guarding: The master periodically sends messages to the slave. If the slave does not receive the master's message within a set period, an alarm will be triggered. The alarm time is "supervision time × life factor."

Master request message format - (0x700 + node number) (remote frame)
Slave response message format - (0x700 + node number) + status:
Status - The data part includes a trigger bit (bit7), which must alternate between "0" or "1" in each node protection response.

The trigger bit is set to "0" during the first node protection request. Bits 0 to 6 (bit0～bit6) represent the node status;
0: Initialization, 1: Disconnected, 2: Connected, 3: Operation, 4: Stop, 5: Run, 127: Pre-operation.

Consumer heartbeat time: The consumer heartbeat time includes three parts: node protection time, life factor, and node protection ID. Among them, bit0~bit15 is the heartbeat protection time, and bit16~bit23 is the heartbeat protection ID. This is derived for user convenience.

SVD series support both heartbeat messages and node protection modes.

(8)NMT Management

During the network initialization process, CANopen supports extended boot-up and also supports a minimized boot-up process. This initialization process can be represented by a node status transition diagram.

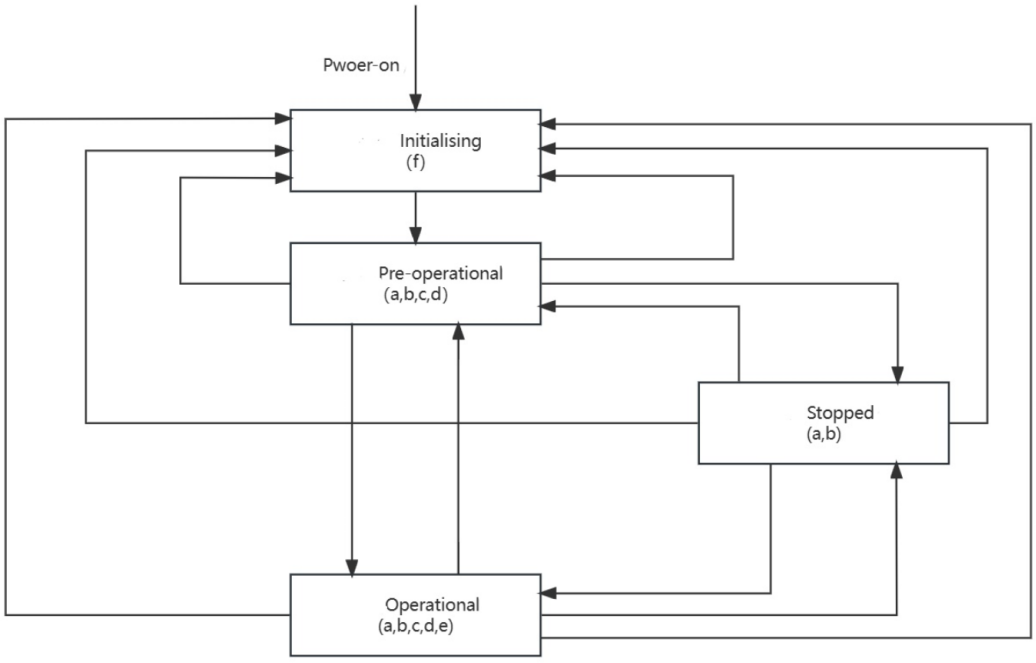


Figure 12-1 Status Transition Diagram

a：MNT　　b：Node Guard
c：SDO　　d：Emergency
e：PDO　　f：Boot-up

Initialization (Initialising): The first state after the node is powered on or reset, the node performs basic hardware and software initialization and cannot communicate.

Pre-operational (Pre-operational): The state automatically entered after the node initialization is completed, the node can perform SDO communication but cannot perform PDO communication. This state is usually used for configuring node parameters and PDO mapping.

Operational (Operational): The node can perform all types of communication, including SDO and PDO. This state is the normal working state of the node. (Generally, enter the operational state through NMT management: such as 000 01 01, detailed introduction below.)

Stopped (Stopped): The node stops all communication, except for NMT and heartbeat (if enabled). This state can be used to implement specific application behaviors.

NMT Management Message Format

COB-ID	DLC	Byte0	Byte1
0x000	02	Command	Station No.

Command word related content is as follows:

Command	NMT service
0x01	Start the node and start PDO transmission
0x02	Close the node and stop PDO transmission
0x80	Enter the pre-operational state
0x81	Reset the node
0x82	Reset communication

When Node-ID=0, all NMT slave devices are addressed.

Note:

You can use NMT management messages to switch between various modes. Only the NMT-Master node can transmit NMT Module Control messages, and all slave devices must support the NMT module control service. In addition, NMT Module Control messages do not require a response. PDO can only be transmitted in the operational state (generally through PDO).

12.3 CAN Communication Examples

12.3.1 SDO Communication Example

The default station number is 01.

（1）Write speed mode

For details on the conversion of target speed and other parameters, please refer to the UART communication example or the parameter conversion formula in Chapter 6.

10rpm is converted to hexadecimal as 57619

50rps/s is converted to hexadecimal as 1A36E

CAN address	Name	Setting value	Message	Description
60600008	Operation	3	601 2F 60 60 00 03 00 00 00	Operation

	mode			mode is set to 3
60FF0020	Target speed	10rpm	601 23 FF 60 00 19 76 05 00	Target speed is set to 10rpm
60830020	Profile acceleration	50rps/s	601 23 83 60 00 6E A3 01 00	Profile acceleration is set to 50rps/s
60840020	Profile deceleration	50rps/s	601 23 84 60 00 6E A3 01 00	Profile deceleration is set to 50rps/s
60400010	Control word	F	601 2B 40 60 00 0F 00 00 00	The control word is set to F, locking the motor axis

According to the above operations, the motor will accelerate to 10 rpm with profile acceleration of 50 rps/s. If the target speed is written to 0 again, the motor will decelerate to 0 with aprofile deceleration of 100 rps/s.

（2）Write relative position mode

For details on the conversion of target speed and other parameters, please refer to the UART communication example or the parameter conversion formula in Chapter 6.
50 rpm is converted to decimal as 1,789,569, and hexadecimal as 1B 4E81.
10000 is converted to hexadecimal as 2710.

CAN address	Name	Setting value	Message	Description
60600008	Operation mode	1	601 2F 60 60 00 01 00 00 00	Operation mode is set as 1
60810020	Profile speed	50rpm	601 23 81 60 00 81 4E 1B 00	Profile speed is set to 50 rpm
607A0020	Target position	10000 inc	601 23 7A 60 00 10 27 00 00	Target position is set to 10000 inc
60400010	Control word	0x4F->0x5F	601 2B 40 60 00 4F 00 00 00 601 2B 40 60 00 5F 00 00 00	The control word is changed from 4F to 5F

According to the above operations, the motor will run 10000 inc distance at a speed of 50 rpm on the original position.

（3）SDO Read Status Word, Actual Speed

By querying the object list, we can know:

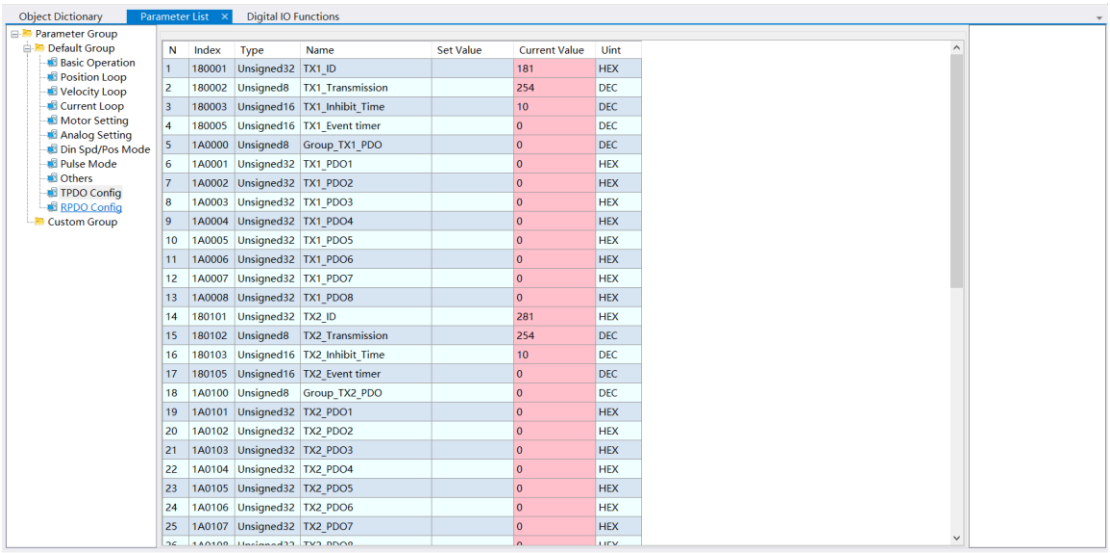
Status word CAN address: 6041002B
Actual position CAN address: 60630023
In the SDO section above, it has been introduced that the command word for executing the read instruction is 0x40.

Operation	Send message	Receive message
①Read status word	601 40 41 60 00 00 00 00 00	5814B41 600031 0200 00
②Read actual position	601 40 63 60 00 00 00 00 00	5814363 60000D 0D00 00
③Read actual speed	601 40 6C 60 00 00 00 00 00	581436C 6000 00 DB FF FF

For the above responsemessages:
① Read status word: 581 4B 41 60 00 31 02 00 00
4B indicates that the returned data is a 16-bit data. The returned data (3102) is converted to hexadecimal as 0231, indicating the status word is 231.
② Read actual position: 581 43 63 60 00 0D 0D 00 00
43 indicates that the returned data is a 32-bit data. The returned data (0D0D 0000) is converted to hexadecimal as 0D0D, and then converted to decimal as 3341.
③ Read actual speed: 581 43 6C 60 00 00 DB FF FF
43 indicates that the returned data is a 32-bit data. The returned data (00DB FFFF) is converted to hexadecimal as FFFF 00DB. Since the highest bit is F, the actual value is a negative number -65317, converted to rpm as -2.
Note:
The most significant bit (MSB) of the negative values of speed and position is 1, and they are transmitted in two's complement form. If the actual position is -5000, it will be represented in the computer as FFFF EC78.

12.3.2 PDO Configuration

If configured through the AMPS software, you can follow these steps:
In the work area, Click on "Parameter List" -> Click on "TPDO Configuration"



N	Index	Type	Name	Set Value	Current Value	Uint
1	180001	Unsigned32	TX1_ID		181	HEX
2	180002	Unsigned8	TX1_Transmission		254	DEC
3	180003	Unsigned16	TX1_Inhibit_Time		10	DEC
4	180005	Unsigned16	TX1_Event timer		0	DEC
5	1A0000	Unsigned8	Group_TX1_PDO		0	DEC
6	1A0001	Unsigned32	TX1_PDO1		0	HEX
7	1A0002	Unsigned32	TX1_PDO2		0	HEX
8	1A0003	Unsigned32	TX1_PDO3		0	HEX
9	1A0004	Unsigned32	TX1_PDO4		0	HEX
10	1A0005	Unsigned32	TX1_PDO5		0	HEX
11	1A0006	Unsigned32	TX1_PDO6		0	HEX
12	1A0007	Unsigned32	TX1_PDO7		0	HEX
13	1A0008	Unsigned32	TX1_PDO8		0	HEX
14	180101	Unsigned32	TX2_ID		281	HEX
15	180102	Unsigned8	TX2_Transmission		254	DEC
16	180103	Unsigned16	TX2_Inhibit_Time		10	DEC
17	180105	Unsigned16	TX2_Event timer		0	DEC
18	1A0100	Unsigned8	Group_TX2_PDO		0	DEC
19	1A0101	Unsigned32	TX2_PDO1		0	HEX
20	1A0102	Unsigned32	TX2_PDO2		0	HEX
21	1A0103	Unsigned32	TX2_PDO3		0	HEX
22	1A0104	Unsigned32	TX2_PDO4		0	HEX
23	1A0105	Unsigned32	TX2_PDO5		0	HEX
24	1A0106	Unsigned32	TX2_PDO6		0	HEX
25	1A0107	Unsigned32	TX2_PDO7		0	HEX

Figure 12-2 TPDO Configuration
TPDO Station Number: Determined by the driver's DIP switch, for details, see Chapter 4

TPDO Transmission Type:

0: Non-periodic synchronous mode, that is, data is sent when the synchronous message is received and the data changes

1~240: Periodic synchronous, that is, data is sent when x synchronous messages are received, x is the set value

241~253: Reserved

254/255: Non-periodic non-synchronous, at this time, "Event Time" is effective. If the event time is non-zero, it is sent immediately after the event time; if it is zero, the data is sent when the change occurs and the time since the last send is greater than the inhibition time.

TPDO Inhibition Time: Send PDO data frames after a period of time to avoid network congestion and data conflicts, unit is ms

TPDO Event Time: The period of time for the driver to send PDO to the controller, unit is ms

TPDO Valid Mapping Object Count: The set number of mappings.

TPDO Mapping 1-8: Configure CANopen control objects

Note:

When using the asynchronous transmission event-driven immediate transmission function, the corresponding inhibition time should be set, and the event time should be set to 0;

When using the time-based periodic reporting function: the corresponding event time should be set, and the inhibition time should be set to 0.

The total length of the objects mapped in each PDO should not exceed 8 bytes.

SDO Communication Common Object List

CANopen address	Name	Description	Default
10050020	Sync ID	Used when the transmission type is 1-240 synchronous mode, not needed in asynchronous mode.	80
100C0010	Node guarding time	The master station periodically sends remote frames to inquire about the status of the slave node. The slave node must respond within a certain period of time, otherwise, the master station will consider the slave node to be offline and the driver will alarm. Guarding time * Life factor = Life time of node guarding	1000
100D0008	Life factor		3
100E0020	Node guarding ID	700+ Driver ID	
10140020	Emergency message station number	80 + Driver station number	

10160120	Consumer Heartbeat time	"bit0~bit15: Heartbeat guarding time bit16~bit23: Heartbeat guarding ID"	
2F800208	CAN baud rate	CAN baud rate setting 100：1M 50：500k 25：250k 12：125k 10: 100k 5： 50k 2: 20k	50
30110410	ECAN sync loss count	Monitor communication status in synchronous mode, no change in the value indicates good communication status, if the value keeps changing, it indicates interference or incorrect sync period setting.	
60070010	Communication interruption mode	CAN communication interruption mode, after the set time has passed without receiving the node guarding message, the action logic. 0: Do not process 1/2: Error, and release axis 3: Warning, and emergency stop	0

12.3.3 NMT Management Example

According to Section 12.2.3, the NMT management message format is as follows:

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

For example, with Node ID 1:
The command to start the node is 0x01, then the start PDO node message is: 000 01 01;
The command to close the node is 0x02, then the close PDO node message is: 000 02 01;
The command to reset the node is 0x81, then the close PDO node message is: 000 81 01;

12.3.4 PDO Communication Example

（1）TPDO Synchronous Communication Example

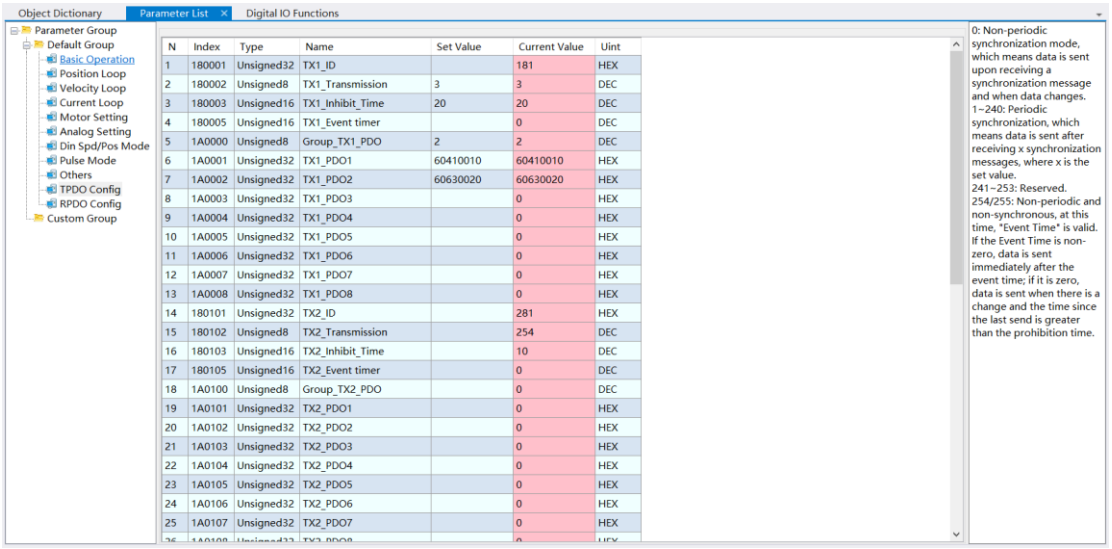


Figure 12-3 Parameter Mapping

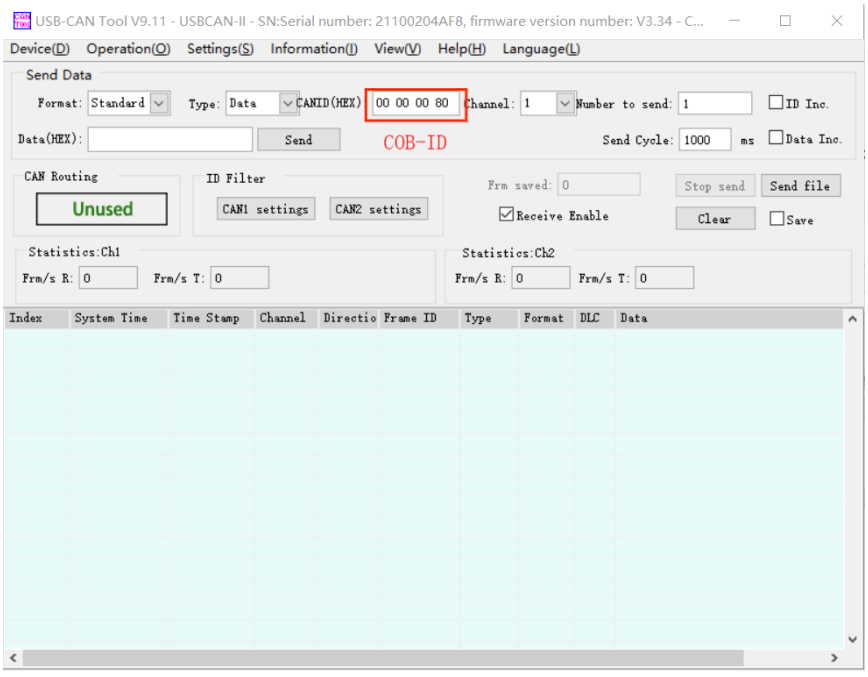
We set the TPDO1 station number to 181, the transmission type to 3 (send data once after receiving 3 synchronous messages), the inhibition time to 20, TPDO1 Mapping 1 to 60410010, and TPDO1 Mapping 2 to 60630020.

Send message: 000 01 01 to start the message (no response message)

000: COB-ID, 01: Start, if 02 is entered, it is to close, 01: Node ID

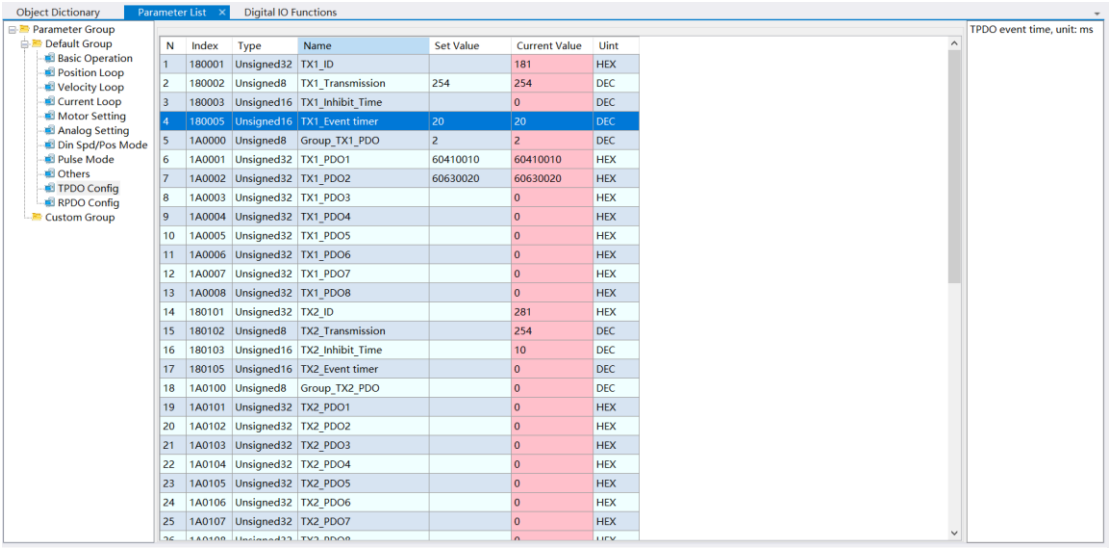
Below is a demonstration through "Innovative Chip Technology" CAN interface software.

After sending three times 0x80 (0x80 refers to COB-ID), you can receive the response message from the driver.



The received message is as follows: B1 02 17 F0 06 00
It indicates that the data in the status word (60410010) is 02B1
The actual position (60630020) data is 0006F017 (454679)

（2）TPDO Asynchronous Communication Example

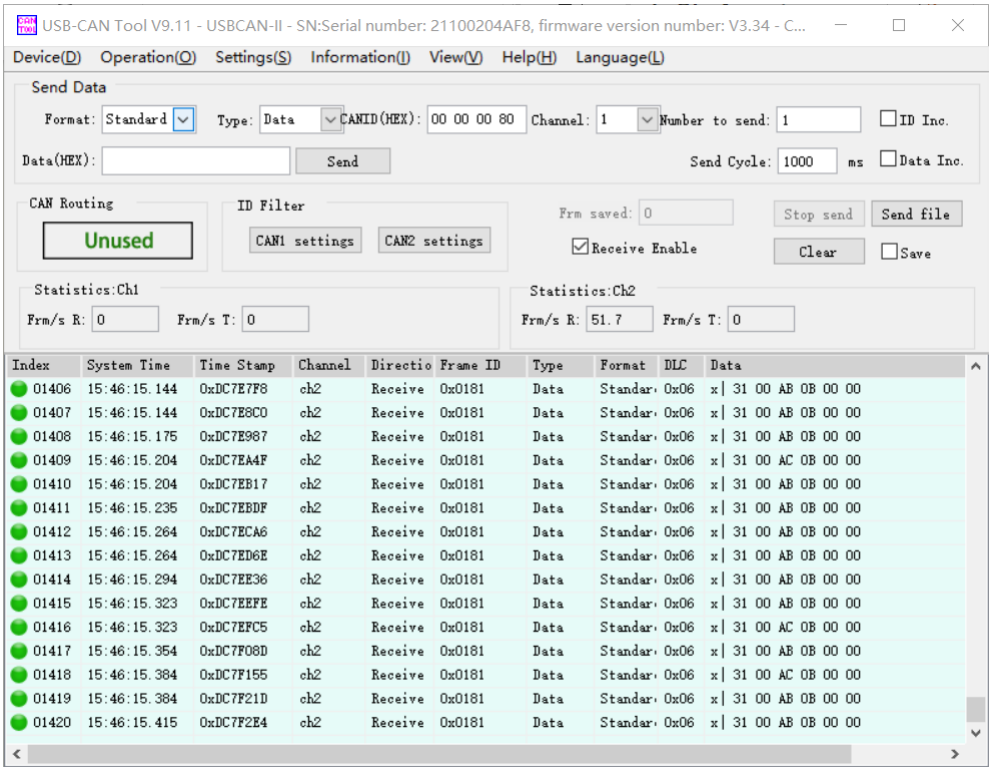


N	Index	Type	Name	Set Value	Current Value	Uint
1	180001	Unsigned32	TX1_ID		181	HEX
2	180002	Unsigned8	TX1_Transmission	254	254	DEC
3	180003	Unsigned16	TX1_Inhibit_Time		0	DEC
4	180005	Unsigned16	TX1_Event timer	20	20	DEC
5	1A0000	Unsigned8	Group_TX1_PDO	2	2	DEC
6	1A0001	Unsigned32	TX1_PDO1	60410010	60410010	HEX
7	1A0002	Unsigned32	TX1_PDO2	60630020	60630020	HEX
8	1A0003	Unsigned32	TX1_PDO3		0	HEX
9	1A0004	Unsigned32	TX1_PDO4		0	HEX
10	1A0005	Unsigned32	TX1_PDO5		0	HEX
11	1A0006	Unsigned32	TX1_PDO6		0	HEX
12	1A0007	Unsigned32	TX1_PDO7		0	HEX
13	1A0008	Unsigned32	TX1_PDO8		0	HEX
14	180101	Unsigned32	TX2_ID		281	HEX
15	180102	Unsigned8	TX2_Transmission		254	DEC
16	180103	Unsigned16	TX2_Inhibit_Time		10	DEC
17	180105	Unsigned16	TX2_Event timer		0	DEC
18	1A0100	Unsigned8	Group_TX2_PDO		0	DEC
19	1A0101	Unsigned32	TX2_PDO1		0	HEX
20	1A0102	Unsigned32	TX2_PDO2		0	HEX
21	1A0103	Unsigned32	TX2_PDO3		0	HEX
22	1A0104	Unsigned32	TX2_PDO4		0	HEX
23	1A0105	Unsigned32	TX2_PDO5		0	HEX
24	1A0106	Unsigned32	TX2_PDO6		0	HEX
25	1A0107	Unsigned32	TX2_PDO7		0	HEX

Figure 12-4 Parameter Mapping

We set the TPDO1 station number to 181, the transmission type to 254, the inhibition time to 20, TPDO1 Mapping 1 to 60410010, and TPDO1 Mapping 2 to 60630020.

Then send the message: 000 01 01 to start the message to receive the response message from the driver. When the data changes, it returns the data every 20ms (TPDO inhibition time).

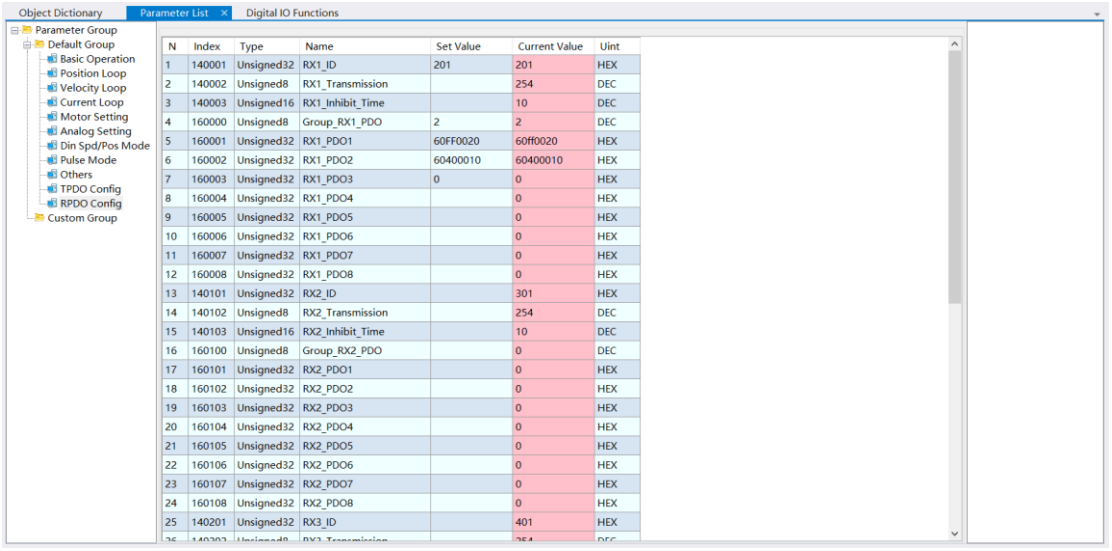


Index	System Time	Time Stamp	Channel	Directio	Frame ID	Type	Format	DLC	Data
01406	15:46:15.144	0xDC7E7F8	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AB 0B 00 00
01407	15:46:15.144	0xDC7E8C0	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AB 0B 00 00
01408	15:46:15.175	0xDC7E987	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AB 0B 00 00
01409	15:46:15.204	0xDC7EA4F	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AC 0B 00 00
01410	15:46:15.204	0xDC7EB17	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AB 0B 00 00
01411	15:46:15.235	0xDC7EBDF	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AB 0B 00 00
01412	15:46:15.264	0xDC7ECA6	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AB 0B 00 00
01413	15:46:15.264	0xDC7ED6E	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AB 0B 00 00
01414	15:46:15.294	0xDC7EE36	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AB 0B 00 00
01415	15:46:15.323	0xDC7EEFE	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AB 0B 00 00
01416	15:46:15.323	0xDC7EFC5	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AC 0B 00 00
01417	15:46:15.354	0xDC7F08D	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AB 0B 00 00
01418	15:46:15.384	0xDC7F155	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AC 0B 00 00
01419	15:46:15.384	0xDC7F21D	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AB 0B 00 00
01420	15:46:15.415	0xDC7F2E4	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AB 0B 00 00

The received message is as follows: B1 02 17 F0 06 00
It indicates that the data in the status word (60410010) is 02B1
The actual position (60630020) data is 0006F017 (454679)

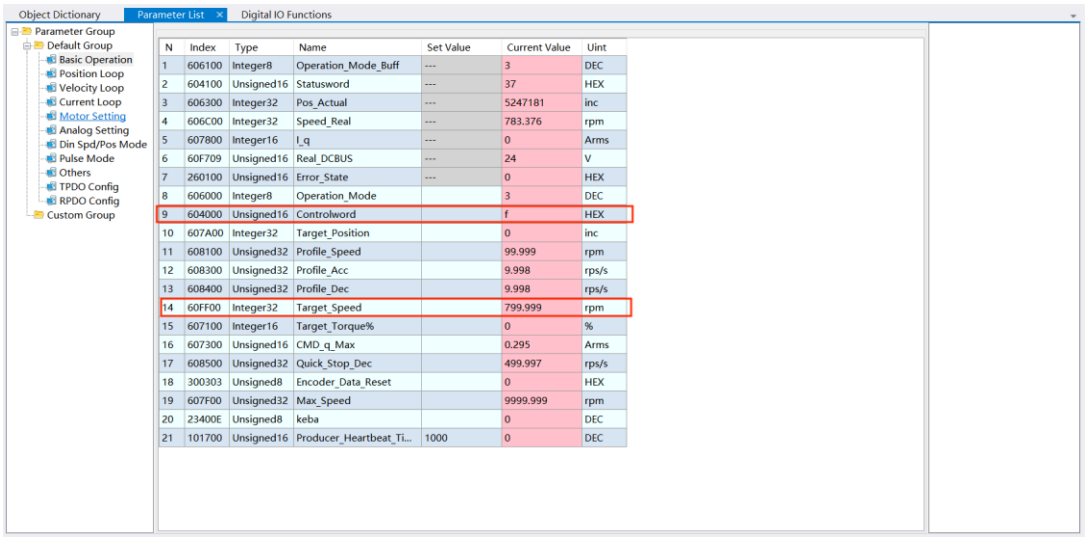
If you need to report periodically, just set the TPDO inhibition time to 0, and set the TPDO event time to the corresponding time (for example, 50), the driver will periodically send TPDO data (regardless of whether the data has changed).

（3）RPDO Communication Example



N	Index	Type	Name	Set Value	Current Value	Uint
1	140001	Unsigned32	RX1_ID	201	201	HEX
2	140002	Unsigned8	RX1_Transmission		254	DEC
3	140003	Unsigned16	RX1_Inhibit_Time		10	DEC
4	160000	Unsigned8	Group_RX1_PDO	2	2	DEC
5	160001	Unsigned32	RX1_PDO1	60FF0020	60ff0020	HEX
6	160002	Unsigned32	RX1_PDO2	60400010	60400010	HEX
7	160003	Unsigned32	RX1_PDO3	0	0	HEX
8	160004	Unsigned32	RX1_PDO4		0	HEX
9	160005	Unsigned32	RX1_PDO5		0	HEX
10	160006	Unsigned32	RX1_PDO6		0	HEX
11	160007	Unsigned32	RX1_PDO7		0	HEX
12	160008	Unsigned32	RX1_PDO8		0	HEX
13	140101	Unsigned32	RX2_ID		301	HEX
14	140102	Unsigned8	RX2_Transmission		254	DEC
15	140103	Unsigned16	RX2_Inhibit_Time		10	DEC
16	160100	Unsigned8	Group_RX2_PDO		0	DEC
17	160101	Unsigned32	RX2_PDO1		0	HEX
18	160102	Unsigned32	RX2_PDO2		0	HEX
19	160103	Unsigned32	RX2_PDO3		0	HEX
20	160104	Unsigned32	RX2_PDO4		0	HEX
21	160105	Unsigned32	RX2_PDO5		0	HEX
22	160106	Unsigned32	RX2_PDO6		0	HEX
23	160107	Unsigned32	RX2_PDO7		0	HEX
24	160108	Unsigned32	RX2_PDO8		0	HEX
25	140201	Unsigned32	RX3_ID		401	HEX

We set the RPDO1 station number to 201, the transmission type to 254, the inhibition time to 10, RPDO1 Mapping 1 to 60FF0020 (target speed), and RPDO1 Mapping 2 to 60400010 (control word).
Send the message: 000 01 01 to start the node (no return message)
Then send 201 03 9D 36 00 0F 00 to write multiple parameters to the driver at the same time (no return message)

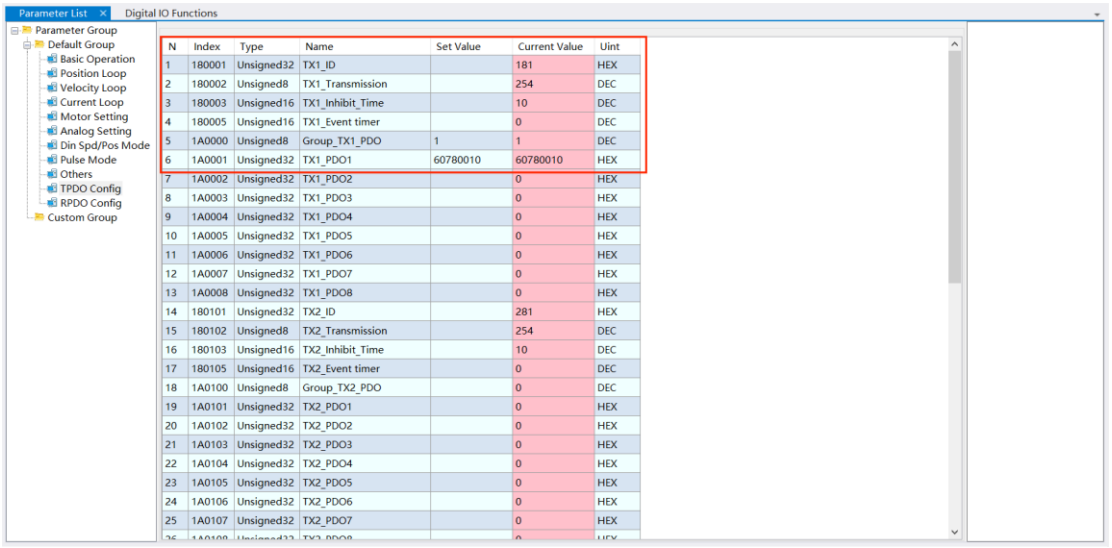


N	Index	Type	Name	Set Value	Current Value	Uint
1	606100	Integer8	Operation_Mode_Buff	---	3	DEC
2	604100	Unsigned16	Statusword	---	37	HEX
3	606300	Integer32	Pos_Actual	---	5247181	inc
4	606C00	Integer32	Speed_Real	---	783.376	rpm
5	607800	Integer16	I_q	---	0	Arms
6	60F709	Unsigned16	Real_DCBUS	---	24	V
7	260100	Unsigned16	Error_State	---	0	HEX
8	606000	Integer8	Operation_Mode		3	DEC
9	604000	Unsigned16	Controlword		f	HEX
10	607A00	Integer32	Target_Position		0	inc
11	608100	Unsigned32	Profile_Speed		99.999	rpm
12	608300	Unsigned32	Profile_Acc		9.998	rps/s
13	608400	Unsigned32	Profile_Dec		9.998	rps/s
14	60FF00	Integer32	Target_Speed		799.999	rpm
15	607100	Integer16	Target_Torque%		0	%
16	607300	Unsigned16	CMD_q_Max		0.295	Arms
17	608500	Unsigned32	Quick_Stop_Dec		499.997	rps/s
18	300303	Unsigned8	Encoder_Data_Reset		0	HEX
19	607F00	Unsigned32	Max_Speed		9999.999	rpm
20	23400E	Unsigned8	keba		0	DEC
21	101700	Unsigned16	Producer_Heartbeat_Ti...	1000	0	DEC

（4）keba

This driver can not only enable PDOs through NMT management but also by modifying the value of 0x23400E — the KEBA value to enable PDOs. An example is as follows:

1. First, set the PDO-related parameters; here, we set the actual current value to be read through TPDO.

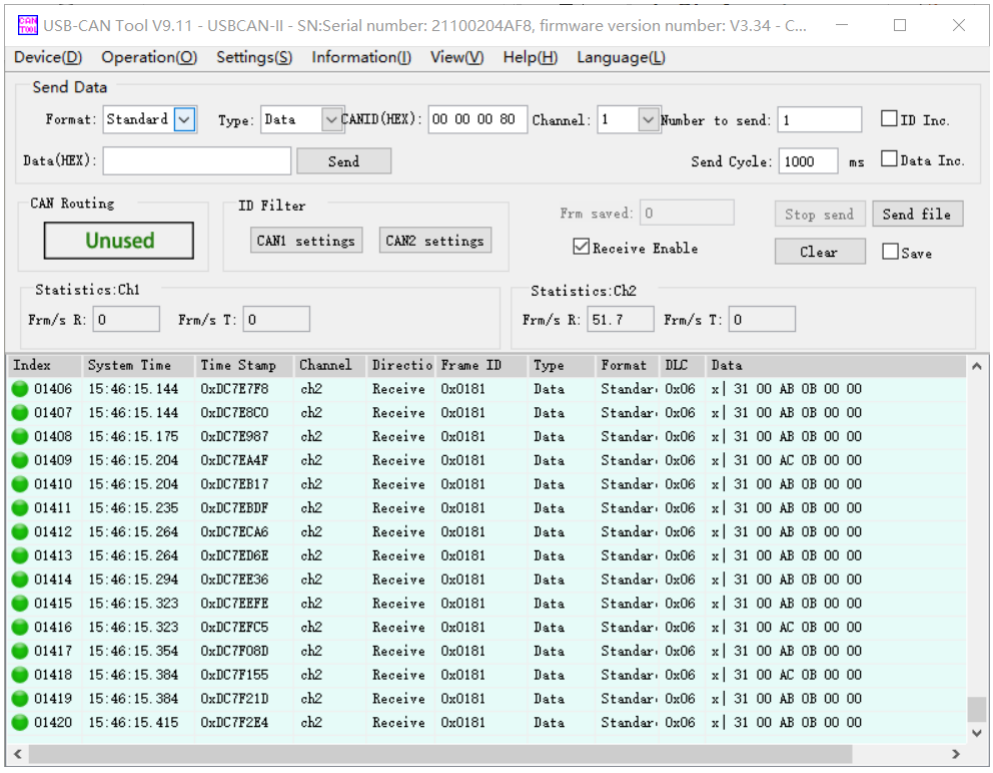


N	Index	Type	Name	Set Value	Current Value	Uint
1	180001	Unsigned32	TX1_ID		181	HEX
2	180002	Unsigned8	TX1_Transmission		254	DEC
3	180003	Unsigned16	TX1_Inhibit_Time		10	DEC
4	180005	Unsigned16	TX1_Event timer		0	DEC
5	1A0000	Unsigned8	Group_TX1_PDO	1	1	DEC
6	1A0001	Unsigned32	TX1_PDO1	60780010	60780010	HEX
7	1A0002	Unsigned32	TX1_PDO2		0	HEX
8	1A0003	Unsigned32	TX1_PDO3		0	HEX
9	1A0004	Unsigned32	TX1_PDO4		0	HEX
10	1A0005	Unsigned32	TX1_PDO5		0	HEX
11	1A0006	Unsigned32	TX1_PDO6		0	HEX
12	1A0007	Unsigned32	TX1_PDO7		0	HEX
13	1A0008	Unsigned32	TX1_PDO8		0	HEX
14	180101	Unsigned32	TX2_ID		281	HEX
15	180102	Unsigned8	TX2_Transmission		254	DEC
16	180103	Unsigned16	TX2_Inhibit_Time		10	DEC
17	180105	Unsigned16	TX2_Event timer		0	DEC
18	1A0100	Unsigned8	Group_TX2_PDO		0	DEC
19	1A0101	Unsigned32	TX2_PDO1		0	HEX
20	1A0102	Unsigned32	TX2_PDO2		0	HEX
21	1A0103	Unsigned32	TX2_PDO3		0	HEX
22	1A0104	Unsigned32	TX2_PDO4		0	HEX
23	1A0105	Unsigned32	TX2_PDO5		0	HEX
24	1A0106	Unsigned32	TX2_PDO6		0	HEX
25	1A0107	Unsigned32	TX2_PDO7		0	HEX

2. Then modify keba to 1, and enable PDO to automatically upload the actual current value.

20	23400E	Unsigned8	keba	1	1	DEC

3. To close PDO, it is necessary to change it back to 0, then store the control parameters and restart.



Index	System Time	Time Stamp	Channel	Directio	Frame ID	Type	Format	DLC	Data
01406	15:46:15.144	0xDC7E7F8	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AB 0B 00 00
01407	15:46:15.144	0xDC7E8C0	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AB 0B 00 00
01408	15:46:15.175	0xDC7E987	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AB 0B 00 00
01409	15:46:15.204	0xDC7EA4F	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AC 0B 00 00
01410	15:46:15.204	0xDC7EB17	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AB 0B 00 00
01411	15:46:15.235	0xDC7EBDF	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AB 0B 00 00
01412	15:46:15.264	0xDC7ECA6	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AB 0B 00 00
01413	15:46:15.264	0xDC7ED6E	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AB 0B 00 00
01414	15:46:15.294	0xDC7EE36	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AB 0B 00 00
01415	15:46:15.323	0xDC7EEFE	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AB 0B 00 00
01416	15:46:15.323	0xDC7EFC5	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AC 0B 00 00
01417	15:46:15.354	0xDC7F08D	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AB 0B 00 00
01418	15:46:15.384	0xDC7F155	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AC 0B 00 00
01419	15:46:15.384	0xDC7F21D	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AB 0B 00 00
01420	15:46:15.415	0xDC7F2E4	ch2	Receive	0x0181	Data	Standar	0x06	x| 31 00 AB 0B 00 00

12.3.5 Node Guarding Example

the message has no data, and it should be set as a remote frame
Host send message: 0x700 + node number
Slave (driver) response message: 0x700 + Node ID+ status

For example, with node number 1
Then the host sends the message: 0x701 (continuously sending state)
Slave response message: 0x701 7F
0x701 FF
In the continuous sending state, the slave alternates between responding to two messages, that is, the trigger bit is set to "0" or "1" in each node protection response (as mentioned above in the node protection).

12.3.6 Heartbeat Message Example

CAN address	Name	Setting value	Send and reply message
10170010	Producer heartbeat time	1000	6012B17 1000E8 0300 00
			5816017 1000E8 0300 00

According to the above settings and sending messages, the slave (driver) will report the message every 1 second.
We will clear the producer heartbeat time again to cut off the reply message.

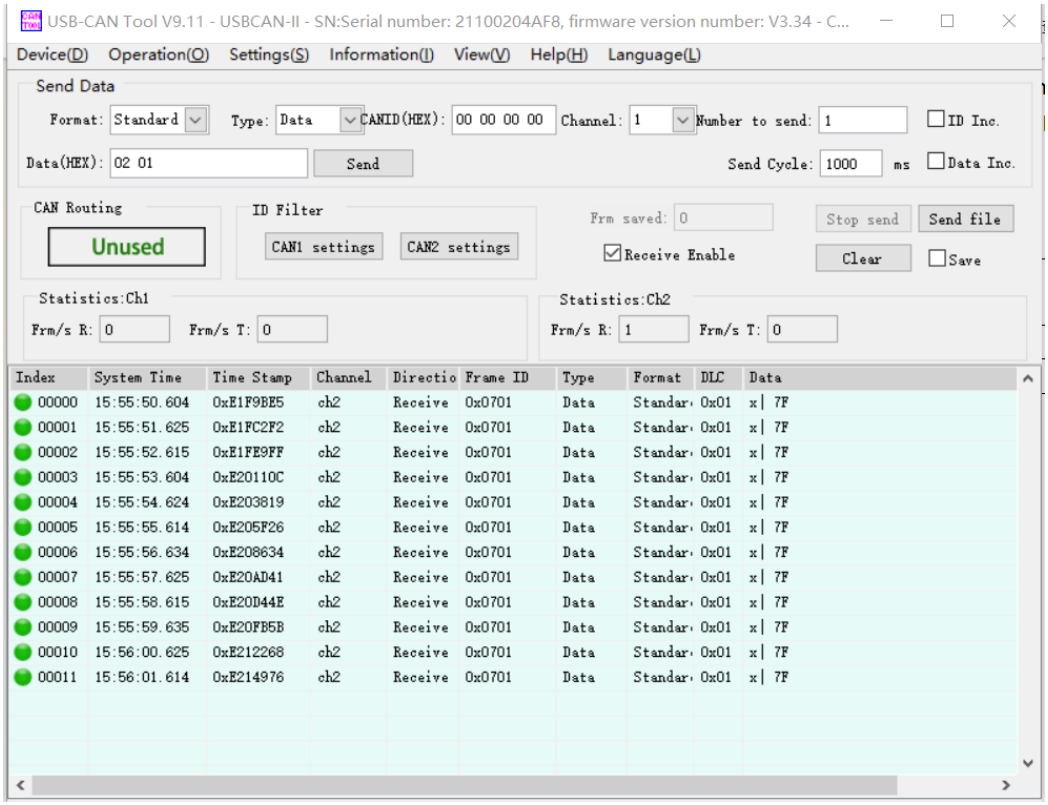


Figure 12-5 Heartbeat message return

Appendix I: Configuring Third-Party Motors

If you wish to configure a third-party motor, please pay attention to the following points:

You need to select a third-party motor that is compatible with the Anpush Technology servo driver, such as a motor with the same or similar rated voltage, rated current, rated torque, and other parameters.
You need to choose the appropriate motor model and specifications based on your application scenario and requirements, such as the motor's maximum speed, rated torque, and rated current.
You need to connect the communication and power lines between the motor and the driver correctly according to the manual provided by Anpush Technology, and set the relevant parameters, such as the communication protocol and control mode.
You need to test the motor's operating performance, such as speed, torque, and temperature, and adjust or optimize according to the actual situation.

For configuring third-party motors, refer to the third-party motor drawings to confirm whether the encoder type is incremental or communication-based, and then find the relevant parameters in the table below.

Incremental Type

Parameter Name	Setting Value
Motor Model	XDDA
Feedback Type	Bit0: ABZ wiring check bit1: UVW wiring check bit2: UVW as OC output Example: If both encoder ABZ and UVW are differential inputs, input 3. Iff encoder ABZ is differential and UVW is OC input, then input 5
Feedback Resolution	Change unit to DEC encoder lines × 4 Example: If the encoder has 2500 lines, input 10000
Feedback Period	Same as feedback resolution
Motor Pole Pairs	Refer to motor drawings
Excitation Mode	20
Excitation Current	Default to 3.295 Ap
Excitation Time	Default to 1000ms
Motor iit Current	Change unit to Arms, then input the motor's rated

	current
Motor iit Time	Default to 60 seconds
Motor Maximum Current	Change unit to Arms, then input the motor's maximum current
Line-line Inductance	Refer to motor drawings
Line-line Resistance	Refer to motor drawings
Reverse Electromotive Force	Refer to motor drawings
Torque Constant	Refer to motor drawings
Rotor Inertia	Refer to motor drawings
Brake Duty Cycle	Default to 70%
Brake Delay	Default to 150ms
Motor Current Loop Bandwidth	Default to 2000Hz

Communication Type:

Parameter Name	Setting Value
Motor Model	Single-turn absolute：XTDA Multi-turn absolute：XMDA
Feedback Type	8
Feedback Resolution	If the encoder has x bits of single-turn, then the input value = 2^x Example: If the single-turn encoder has 17 bits, then input 131072
Feedback Period	Change unit to HEX, higher 8 bits for multi-turn bits, lower 8 bits for single-turn bits Example: If the single-turn absolute encoder has 17 bits, then input 17. If the multi-turn absolute encoder has 17 bits, then input 1617
Motor Pole Pairs	Refer to motor drawings
Excitation Mode	20
Excitation Current	Default to 3.295 Ap
Excitation Time	Default to 1000ms
Motor iit Current	Change unit to Arms, then input the motor's rated current
Motor iit Time	Default to 60 seconds
Motor Maximum Current	Change unit to Arms, then input the motor's maximum current
Line-line Inductance	Refer to motor drawings
Line-line Resistance	Refer to motor drawings
Reverse Electromotive Force	Refer to motor drawings
Torque Constant	Refer to motor drawings
Rotor Inertia	Refer to motor drawings

Brake Duty Cycle	Default to 70%
Brake Delay	Default to 150ms
Motor Current Loop Bandwidth	Default to 2000Hz

1. In the motor configuration interface of the AMPS software, correctly input the motor parameters according to the table above.

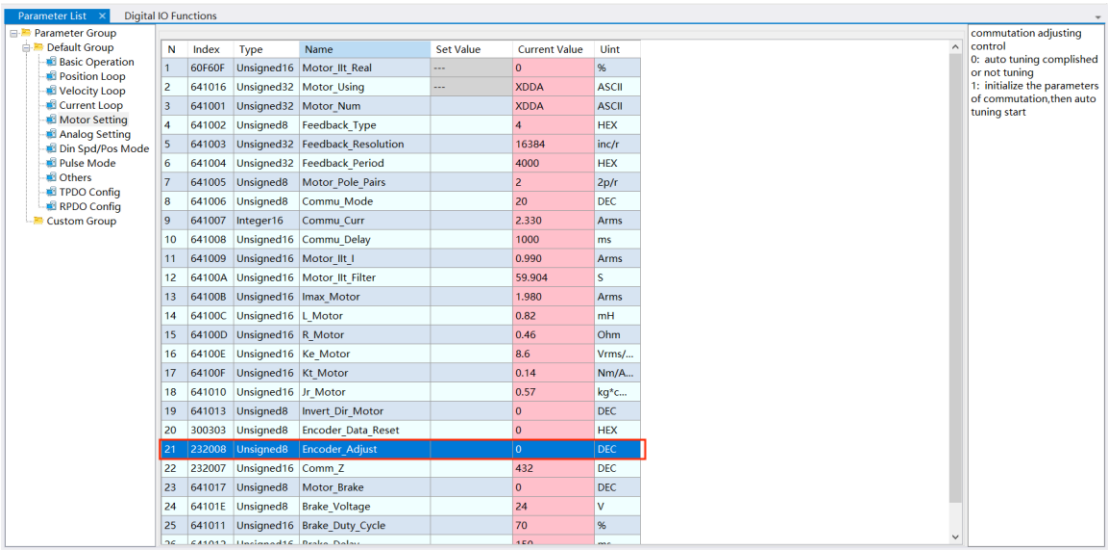
2. After storing the motor parameters, restart the driver.

3. Initialize the motor.

4. Store the control parameters.

6. Write 1 to "Encoder_Adjust" to start self-tuning the Hall angle

7. Wait until "Encoder_Adjust" becomes 0, that is, the self-tuning is complete.

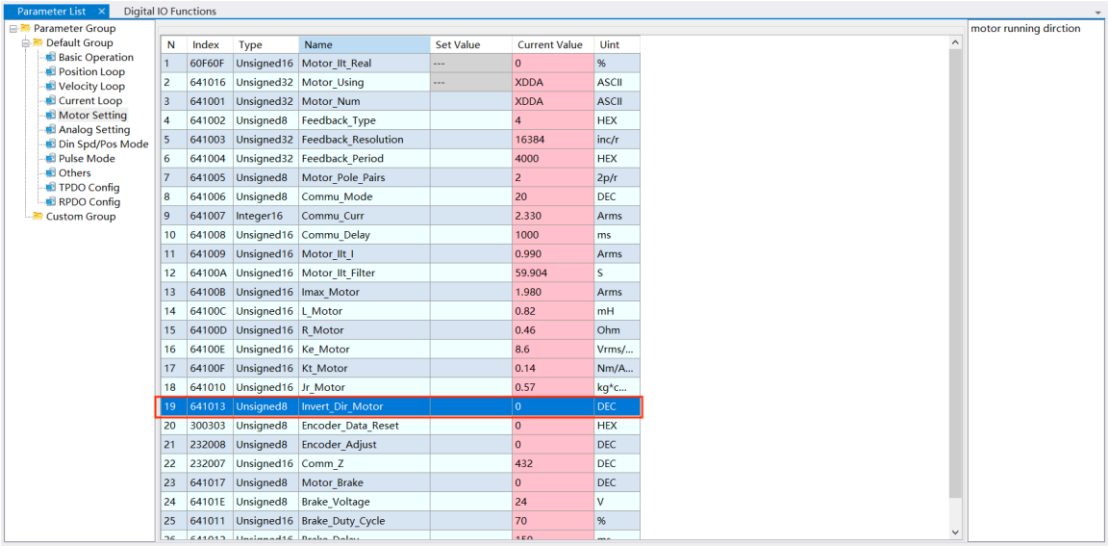


8. After storing the motor parameters, restart the driver.

9. Set the target current limit to 5Ap, then enable the motor and give speed to see if the motor can run.

10. If the motor cannot run, you can repeat step 6.

11. If it still cannot run after multiple attempts, try to modify parameter “Motor rotary direction” as 1



12. Store motor parameters and restart driver,then repeat step 6.

Appendix II: Use of Braking Resistor

The braking resistor inside the driver is a component used to consume the regenerative energy generated when the motor brakes. It can protect the driver from overvoltage damage and achieve rapid stopping or deceleration effects. The principle of the braking resistor is to convert the motor's kinetic and magnetic energy into thermal energy, which is then dissipated by the resistance heating, thereby reducing the voltage on the DC bus.

This model of the driver connects the braking resistor through CN9's RB+ and RB-, and sets the correct braking resistor resistance value, braking resistor power, etc., through software. The selection of the braking resistor is related to the motor model, and the specific selection can refer to the following table:

Motor Power	Braking Resistor Resistance [Ω]	Braking Resistor Power [W]	Braking Resistor Voltage [VDC]
50W	27	100	500
100W	10	100	500
200W	5	100	500
400W	3.5	200	500
750W	0.8	700	500

Parameter Name	UART Address	Object Property	Unit	Description
Braking Resistor Resistance	60F701	Unsigned16 RW	Ω	Braking resistor resistance
Braking Resistor Power	60F702	Unsigned16 RW	W	Braking resistor rated power
Braking Resistor Time Constant	60F703	Unsigned16 RW	S	Braking resistor time constant Time is N*256/1000
Chopping Voltage	651006	Unsigned16 RW	V	Driver chopping voltage

Appendix III: Selection of Fuse Specifications

Motor Power (W)	Fuse Reference Specification
50	3A/58VDC
100	5A/58VDC
200	10A/58VDC
400	20A/58VDC
750	40A/58VDC