User Manual

iWMC Series

Integrated Servo Wheel

Version modification instruction

Date	Modification content
2022.09	V1.0
2022.10	V1.1
2022.12	V1.2
2023.02	V1.3

Preface

iWMC integrated servo wheel is a fully integrated design power module released by Kinco. The integration of the wheel, reduction drive, servo motor and driver modules into a single package optimises the wheel structure, simplify the installation process and shortens the whole wheel installation time.

This product is available in two power options and are Ideally suited for the travel axis of mobile robots with loads of 600 kg and 800kgs.

Please read the manual carefully and follow the operating requirements for setting up the drive correctly for optimal performance.

Item for Acceptance	Description
Whether it match the model you ordered	Please confirm whether the motor model, drive model etc. are consistent with the model you ordered according to the motor and drive nameplate information.
Whether the motor wiring is correct	Please check whether the motor wiring model is consistent with the order.
Whether any damage to the appearance of the product happen?	Please confirm whether the product has been damaged during transportation.
Whether the product accessories are complete?	Please confirm whether the various terminals of the driver、the motor oil seal and keys are complete.

List of drive accessories

Product model	Accessories and quantity
iWMC10409-02222-A165-MBDT **iWMC10409-02222-A165-MADT** **iWMC10415-05417-A180-MBDT** **iWMC10415-05417-A180-MADT**	Certificate*1 Service Guide*1

If there is any problem with any of the above, please contact us.

Cautions

Please read and follow the requirements in this manual carefully as it will help you set up and operate the drive correctly and to optimize its performance. Please be aware of the contents of the warning and strictly follow the requirements, otherwise dangerous situations may occur.

⚠ Warnings

- Do not install if parts are missing when unboxing or the appearance of the wheel is damaged.
- Please install in a place which is well-ventilated, dry and dust-free, no abrasive fluid, no oil mist, no iron powder, no chips, etc., and the surrounding items should be non-flammable.
- When installing/removing the servo wheel, stress must not be applied to the motor body and it must be ensured that each fixing bolt is locked in place.
- Avoid any foreign objects entering the servo wheel. Conductive foreign objects such as screws, metal shavings or flammable foreign objects entering the servo wheel may cause fire and electric shock.
- Do not use gasoline, thinner, alcohol, acidic and alkaline detergents to avoid discoloration or damage to the body.
- Before wiring, please make sure that the wheel is isolated for the power source.
- Do not plug and unplug the terminal directly when the power is on.
- Please use the original packaging for storage and transportation, which provides sufficient protection to avoid damage during transport.
- Please ensure that this document is available to design engineers, installers, and those responsible for commissioning machines or systems using this product. Please consider the legal provisions applicable to the destination, and:

  —Regulations and standards

  —Provisions for testing organizations and insurance companies —National specifications
- Please ensure that the product is not burdened more than permitted during transportation and storage, including:

  —Mechanical loads

  —Impermissible temperature

  —Moisture

  —Corrosive gas
- Please use the product by following the instructions and warnings in this document strictly！

Table of Contents

Chapter1 Model Description & Installation

1.1 Product Introduction and Nameplate Description

The iWMC integrated servo wheel is a cutting-edge power module engineered by Kinco. This innovative design seamlessly integrates the wheel, reduction gear, servo motor, and servo drive into one cohesive package. This integration optimises the wheel structure, streamlines installation procedures, and significantly reduces the time required for the entire wheel installation process. With a dual power supply design, the system ensures enhanced safety and reliability. The iWMC servo wheel is ideally suited for mobile robots' travel axis, accommodating loads of up to 600 kgs and 800KG.

1.1.1 iWMC Integrated Servo Wheel Models

Model list	Specifications
iWMC10409-02222-A165-MBDT	600KG Load Rating with reduction drive, with brake, with 165mm diameter rubber covered wheel, standard extension cable connector
iWMC10409-02222-A165-MADT	600KG Load Rating with reduction drive, without brake, with 165mm diameter rubber covered wheel, standard extension cable connector
iWMC10409-02222-0000-MBDT	600KG Load Rating with reduction drive, with brake, without 165mm diameter rubber covered wheel, standard extension cable connector
iWMC10409-02222-0000-MADT	600KG Load Rating with reduction drive, without brake, without 165mm diameter rubber covered wheel, standard extension cable connector
iWMC10415-05417-A180-MADT	800KG Load Rating With reduction drive, without brake, with 180mm diameter rubber covered wheel, standard extension cable connector
iWMC10415-05417-A180-MBDT	800KG Load Rating With reduction drive, with brake, with 180mm diameter rubber covered wheel, standard extension cable connector
iWMC10415-05417-0000-MADT	800KG Load Rating With reduction drive, without brake, without 180mm diameter rubber covered wheel, standard extension cable connector
iWMC10415-05417-0000-MBDT	800KG Load Rating With reduction drive, with brake, without 180mm diameter rubber covered wheel, standard extension cable connector

iWMC - 104 09 - 022 22 - A 165 – M B D T - XX

CODE	MODEL
iWM	iW: 3-1 SERVO WHEEL (NO DRIVE)
	iWM: 4-1 SERVO WHEEL
C	C: GENERATOR

CODE	MOTOR STATOR DIAMETER
104	104mm

CODE	GEARBOX RATIO
09	9
15	15
00	NO GEARBOX

CODE	TORQUE
022	22Nm
054	54Nm

CODE	WHEELL SPEED
22	22*10rpm
17	17*10rpm
00	00*10rpm

CODE	WHEEL METERIAL
A	A: POLYURETHANE RUBBER WHEEL
0	0: NO WHEEL FITTED

CODE	CUSTOM DESIGN
XX	00: STANDARD

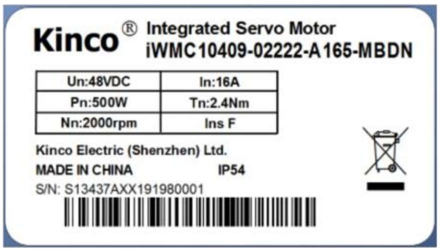
CODE	CONNECTOR
N	X: DIRECTLY CONNECTED TO DRIVE OR PLC
	T: STANDARD CONNECTOR

CODE	VOLTAGE
D	D: DC48V

CODE	BRAKE
B	B: BRAKE FITTED
	A: NO BRAKE FITTED

CODE	ENCODER
M	M: SINGLE TURN ABSOLUTE ENCODER
	W: 2500 PPR INCREMENTAL ENCODER

CODE	WHEEL METERIAL
165	165mm
180	180mm
000	NO WHEEL



Kinco® Integrated Servo Motor
iWMC10409-02222-A165-MBDN

Un:48VDC	In:16A
Pn:500W	Tn:2.4Nm
Nn:2000rpm	Ins F

Kinco Electric (Shenzhen) Ltd.
MADE IN CHINA IP54
S/N: S13437AXX191980001

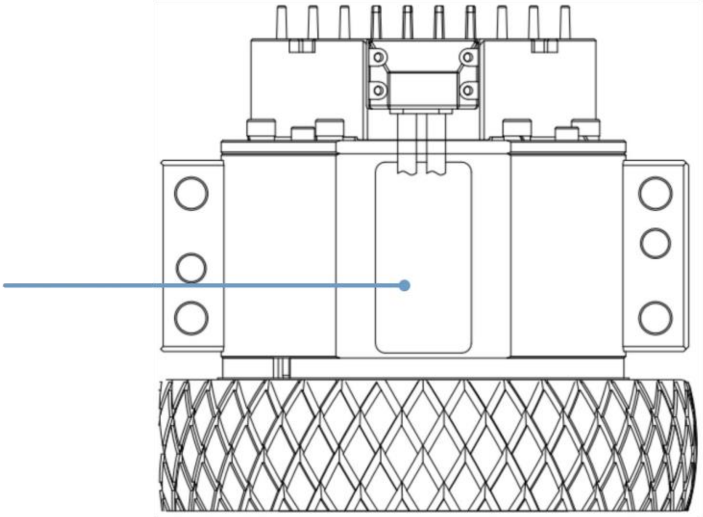


Diagram 1-1 iWMC Integrated servo wheel motor model and nameplate information

1.2 iWMC Integrated Servo Wheel Installation & Precautions

Drive parameters	Minimum	Normal value	Maximum
Input voltage	24V	48V	60V
Brake Voltage	—	24V	—

1.2.1 Installation Dimension

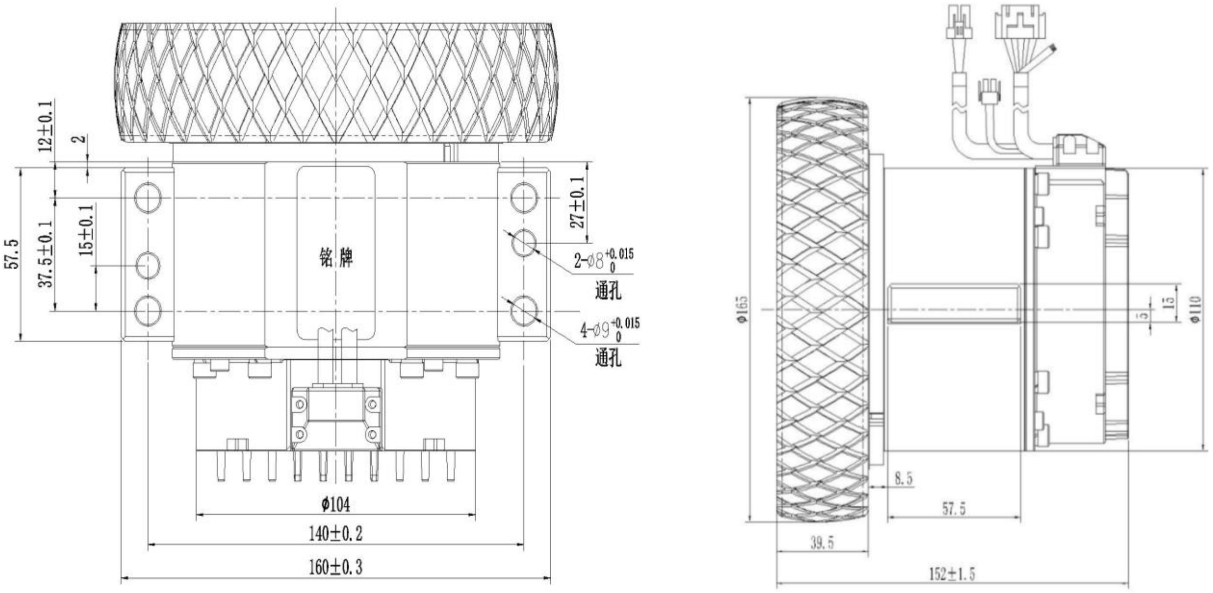


Diagram 1-2 iWMC10409-02222 Integrated servo wheel dimension

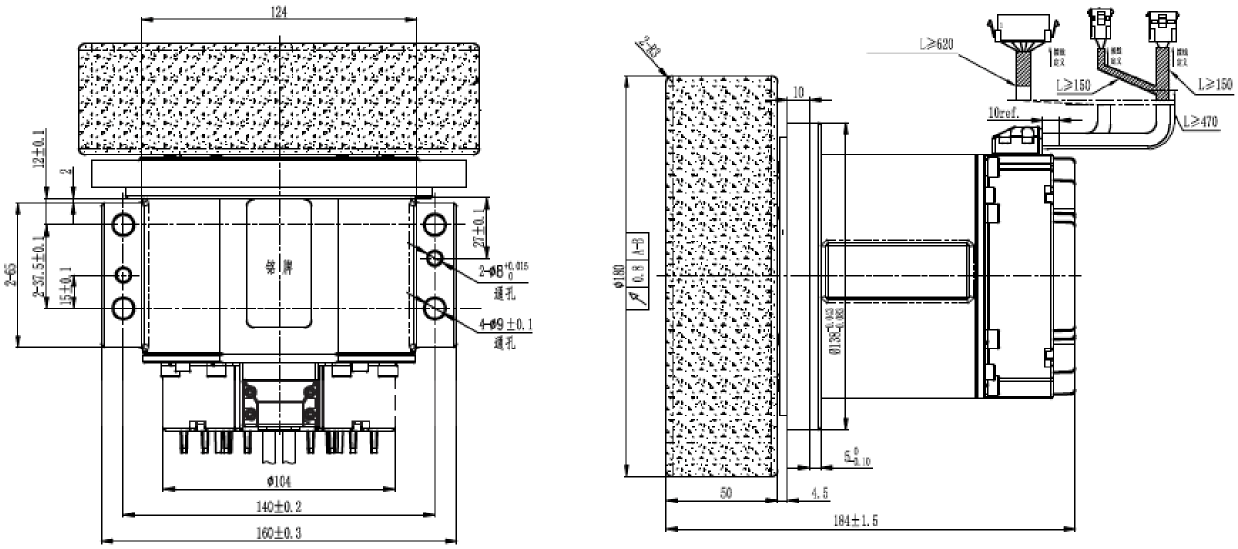


Diagram 1-3 iWMC10415-05417 Integrated servo wheel dimension

1.2.2 Operator Requirements

This product should only be operated by electrical engineers who are familiar with the following regulations:

—Installation and operation of electrical control systems

—Applicable regulations for operating safety engineering systems

—Applicable provisions for accident protection and occupational safety

—Document for the product

1.2.3 Electrical Requirements

Overvoltage alarm	—	68V	—
Undervoltage alarm	—	18V	—

1.2.4 Environment requirements

Environment	**Condition**
Operating temperature	0℃ – 40℃
Operating humidity	5 – 95%RH (No condensation)
Storage temperature	From -10℃ to 70℃ (No freezing)
Storage humidity	Below 90% RH (No condensation)
Protection grade	IP54
Installation site	No sunlight, no corrosive gases, no flammable gases, no oil and gas, no dust, dry and lockable (such as electrical cabinets)
Installation method	Vertical or horizontal installation
Atmospheric pressure	86kpa - 106kpa
Height	The rated working altitude is below 1000 meters, and when the working altitude is above 1000 meters, every 100 meters of rise needs to be reduced by 1.5%. The maximum working altitude is 4000 meters.

Chapter2 System Interfaces And Wiring

2.1 iWMC Servo Wheel Wiring Diagram

2.1.1 iWMC Wiring Diagram

The servo wheel operates with two separate power sources: a 24V logic power supply (connected to pins 1 and 2 of the 12-pin terminal) and a 48V power supply (connected to the 2-pin power terminals) . To ensure proper operation, the servo wheel must be connected to both of these power supplies.

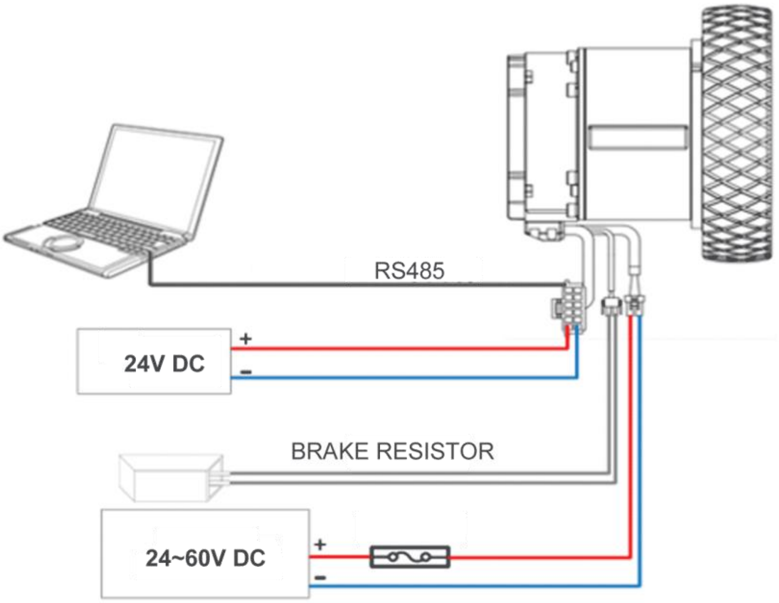


Diagram 2-1 External wiring diagram of servo wheel motor

2.1.2 Brake Resistor And Fuse Specifications

Table 2-1 Brake resistance reference specifications

Brake resistor model	Brake resistance value [Ω]	Braking resistance power [W]	Brake resistance voltage [VDC]
T-10R-100	10	100	500

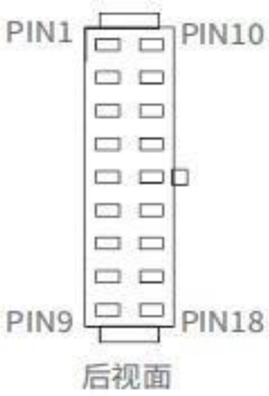
Table 2-2 Fuse reference specifications

Model	Drive power (W)	Fuse specifications
iWMC10409 integrated servo wheel	500	20A/58VDC
iWMC10415 integrated servo wheel	1050	90A/58VDC

2.2 Interface Definition

2.2.1 iWMC Integrated servo wheel with Integrated Terminals

Table 2-3 Definition of servo wheel integrated ports

	Pin No.	**Name**	**Colour**	**Pin Function**
	1	24V	Red	Positive logic power supply input must be plugged in. Input voltage: 24V Maximum input current: 1A Negative logic supply input
	10	GND	Black	
	11	LOCK-	Blue	**The forced release brake input is only used in emergency situations such as the AGV battery is dead. It should be noted that the servo wheel cannot be connected by 48V power supply when using.**
	2	LOCK+	Brown	Input voltage: 24V Maximum input circuit: 0.7A
	3	CANH	Light green	CAN IN
	12	CANL	Light blue	
	4	CANH	Pink	CAN OUT
	13	CANL	White / Black	
	5	485A	Grey	485 IN
	14	485B	White	
	6	485A	Yellow	485 OUT
	15	485B	Green	
	7	OUT+	Purple	Digital signal output Maximum output current: 100mA
	16	COMO	Orange	Output common
	17	DI1	White/ Red	Digital signal input High level: Input voltage 12.5VDC - 30VDC Input current 4-20mA Low level: 0VDC - VDC Input frequency: <1KHz
	18	DI2	White / Orange	
	8	COMI	White / Brown	Input common

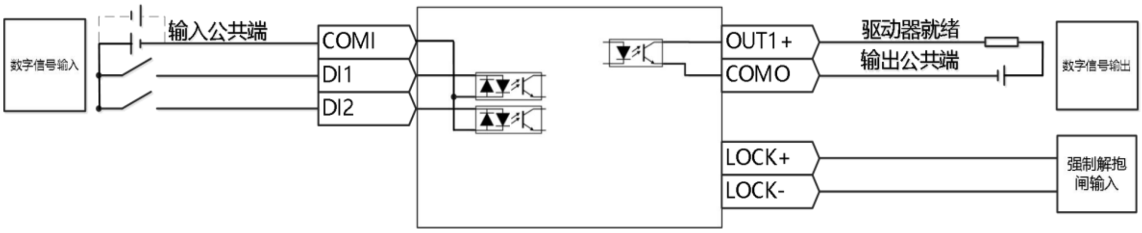


Diagram 2-2 iWMC Integrated servo wheel control wiring diagram

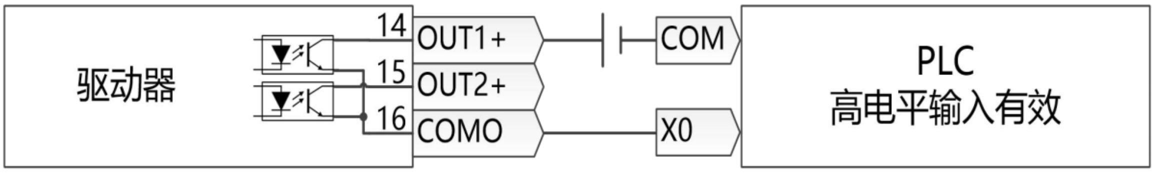


Diagram 2-3 Digital output PNP control wiring diagram

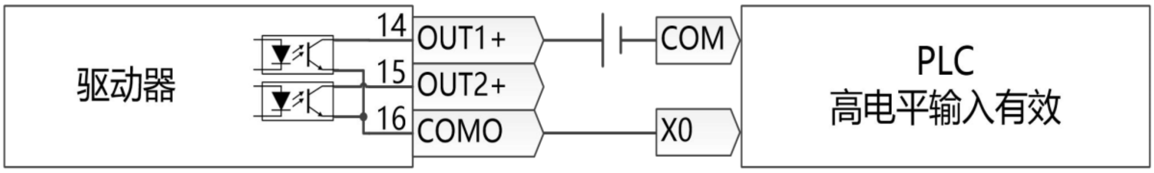


Diagram 2-4 Digital output NPN control wiring diagram

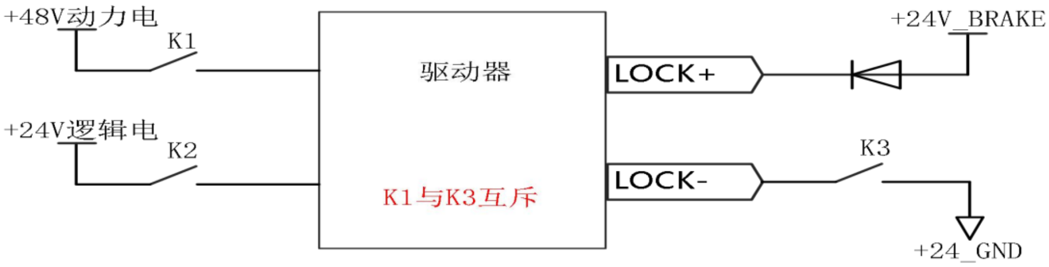
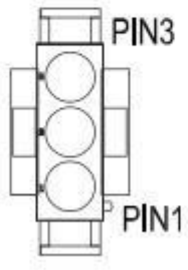
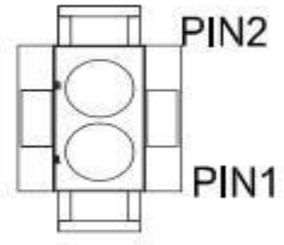


Diagram 2-5 Recommended circuit wiring diagram for forced releasing the brake

2.2.2 Power Supply Port

	Pin No.	Name	Pin Function
 PIN3 PIN1	3	DC-	Drive power supply input must be connected.
	1	DC+	Input voltage: 24~60VDC

2.2.3 Brake Resistor Port

	Pin No.	Name	Pin Function
	1	RB+	External brake resistor input
	2	RB-	

2.2.4 Terminal Specifications

Table 2-4 Terminal Specification Table

	Servo Wheel End	Extension Cord End
Power cord	Rubber shell: SHANGYI C6350HM-3P-V0 (milky white) Pins: SHANGYI C6350M-TBe (Male).	Rubber shell: SHANGYI C6350HF-3P-V0 (milky white) Pins: SHANGYI C6350F-TBe (Female)
	Rubber shell: MOLEX 430201800 Pins: MOLEX 430310004	Rubber shell: MOLEX 430251800 Pins: MOLEX 430300004
Brake resistance	Rubber shell: SHANGYI C6350HM-2P-V0 (milky white) Pins: SHANGYI C6350M-TBe (Male)	Rubber shell: SHANGYI C6350HF-2P-V0 (milky white) Pins: SHANGYI C6350F-TBe (Female)

Cable Specifications:

	Wiring Specifications
Power cord	16AWG
Communication, I/O, etc	28AWG
Brake resistance	16AWG

Chapter 3 KincoServo+ Software Introduction

This chapter will introduce how to debug and configure servo driver by using KincoServo software.

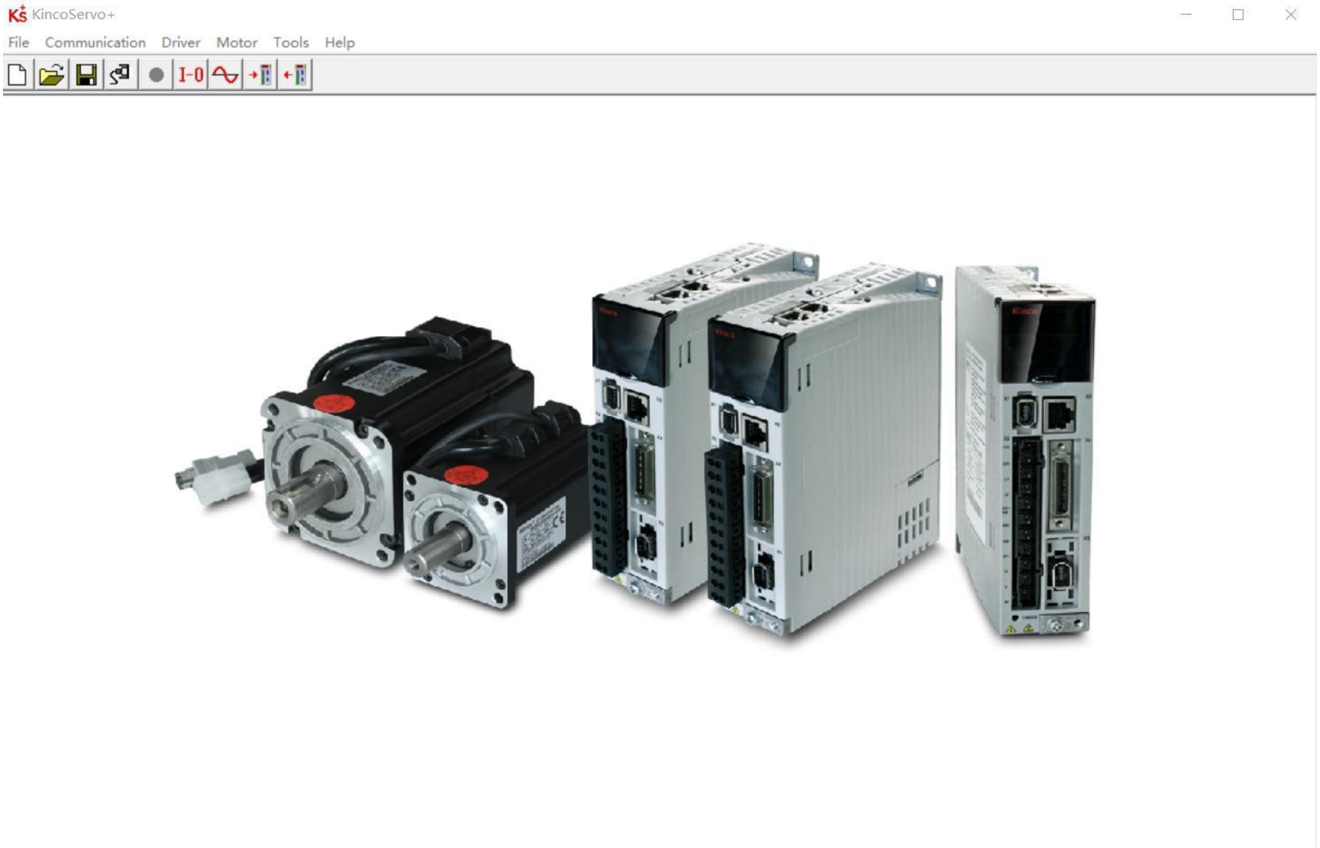


Figure 3–1Software main window

3.1 Fast Start

3.1.1 Language Configuration

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

3.1.2 Open And Save Project Files

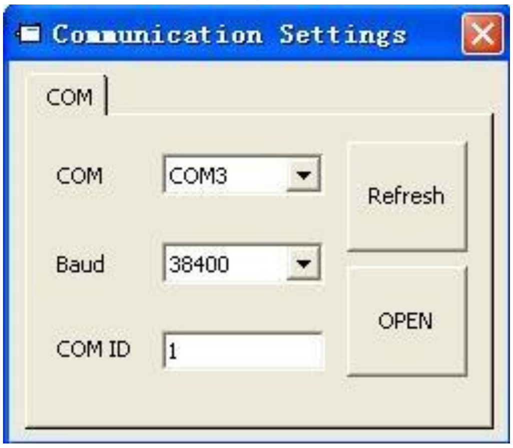
Create a new project file via menu item File->New, or by clicking the button.
Open an existing project via menu item File->Open, or by clicking the button and selecting a .kpjt file.
Save a project via menu item File->Save, or by clicking the button and saving as a .kpjt file.

→ **Note**

Saving the project only saves the window in the computer software, and cannot save the parameters in the drive.

3.1.3 Start Communication

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



COM: COM3

Baud: 38400

COM ID: 1

Select the right COM port (if it's not shown click the "Refresh" button), baud rate (Default 38400) and COM ID (Node ID), and then click the "OPEN" button.

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

3.1.4 Node ID And Baud Rate

The drive's Node ID can be set by the DIP switch on the drive, please refer to the silkscreen on the product for setting.

The drive's Node ID can be set via menu item **Driver**->**Driver Property**.

Table 3-2 Instructions for Node ID and baud rate

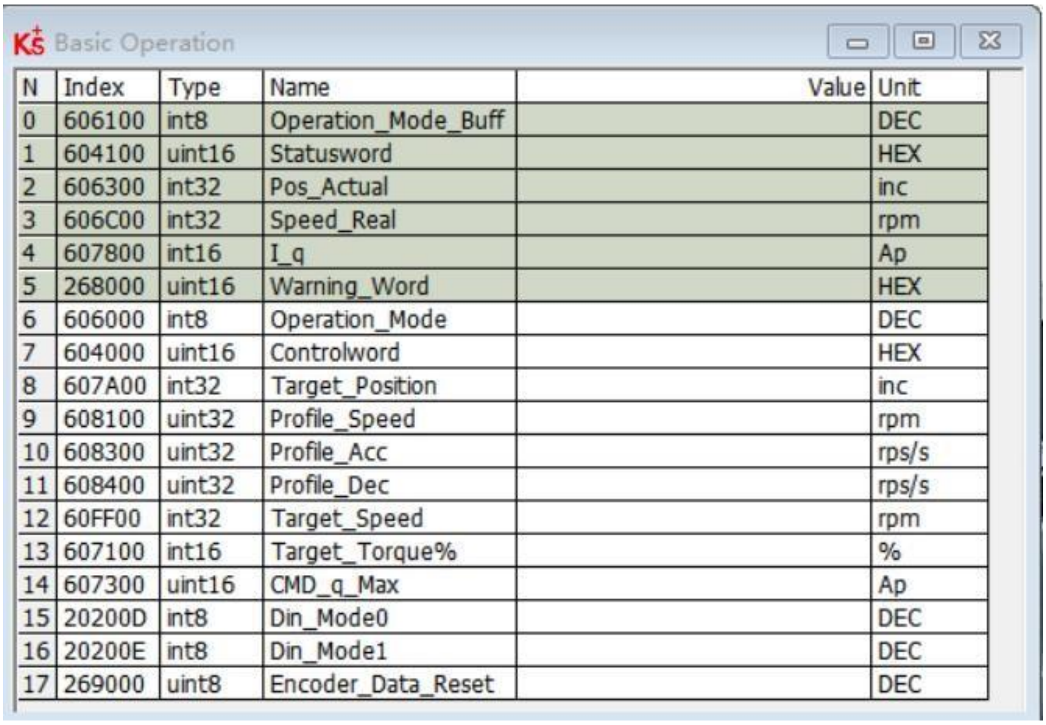
Index	Type	Name	Value	Unit
100B0008	Usigned8	Node ID		DEC
2FE20010	Usigned 16	RS485 baud rate		Baud
65100C08	Usigned8	RS485 protocol		DEC

Note

- iWMC integrated servo wheel does not have RS232 debugging serial port, so you will need to use the RS485 communication port (RS485 protocol defaults to RS232 communication protocol) to connect with the computer. RS485 communication terminal definition refer 2.2.1 for details.
- Node ID and baud rate setting are not activated until after saving and rebooting.

3.1.5 Object (Add, Delete, Help)

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



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

Figure 3–3 Basic Operation Window

Click **Add** and double click the required object from the **Object Dictionary**. The selected object is then added to the list.

Click **Delete**. The selected object is removed from the list.

Click **Help** to read a description of the selected object in the **Object Dictionary**.

3.2 Initialise, Save And Reboot

Click **Driver**->**Initialize/Save**. The following window appears:

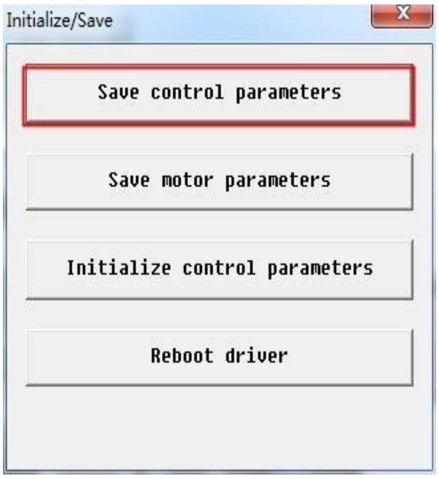


Figure 3–4 Initialise, save, reboot

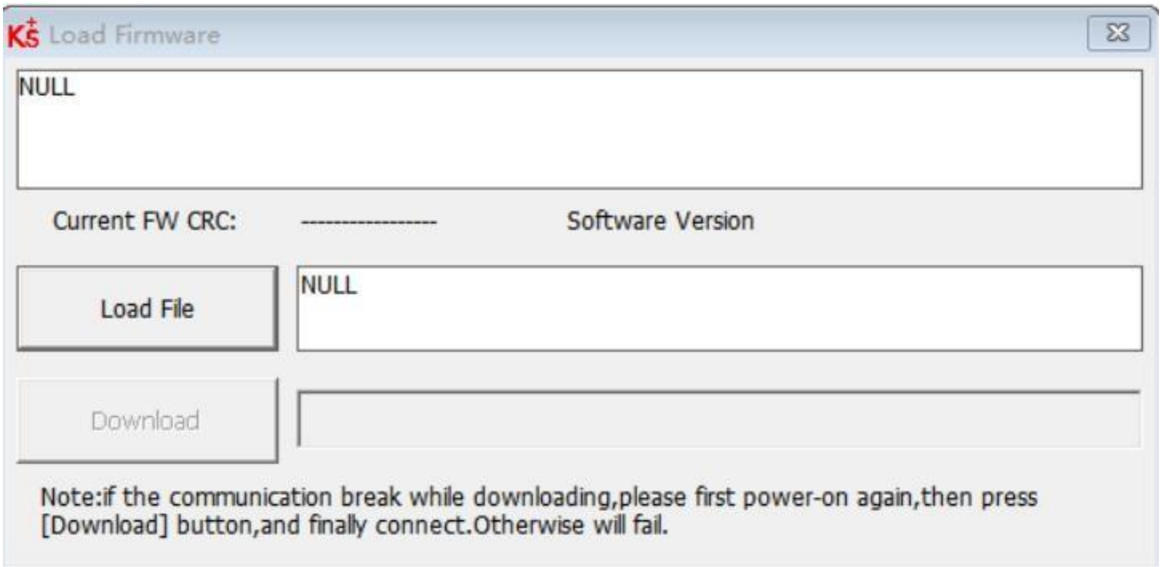
Click the corresponding item to finish the necessary operation.

Note

After completing the **Initialise Control Parameters**, it is recommended to store the control parameters to save the default parameters in the drive.

3.3 Load Firmware

In general, the firmware of the driver is always up to date, but if for some reason you need to update the driver firmware, please click menu bar “**Driver**”>“**Load Firmware**”



NULL

Current FW CRC: ----------------- Software Version

NULL

Note:if the communication break while downloading,please first power-on again,then press [Download] button,and finally connect.Otherwise will fail.

Figure 3-5 Load Firmware

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

Note

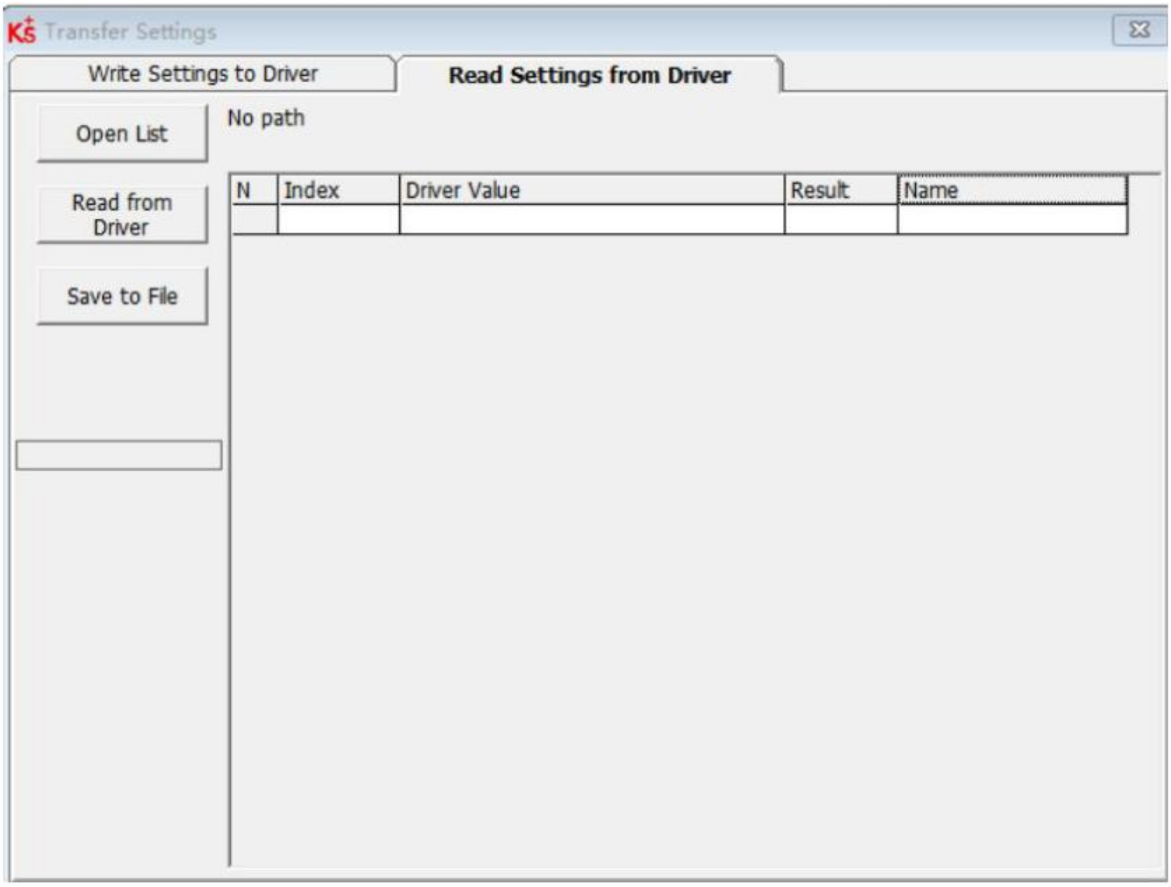
If the download is aborted for some reason, please first turn power off. Then power up the drive, select the firmware version and click Start Download, and finally turn on communication and connect to the computer.

3.4 Read/Write Drive Configuration

This function can be used to read / write multiple parameters simultaneously for large production lots, to avoid setting the drive parameters one by one.

3.4.1 Read Driver Configuration

Click on the menu bar "**Tools**" -> "**R/W Diver Configuration**" - > "**Read Settings from Driver**", or click the button , the following window appears:



No path

N	Index	Driver Value	Result	Name

Figure 3-6 Read Drive Configuration

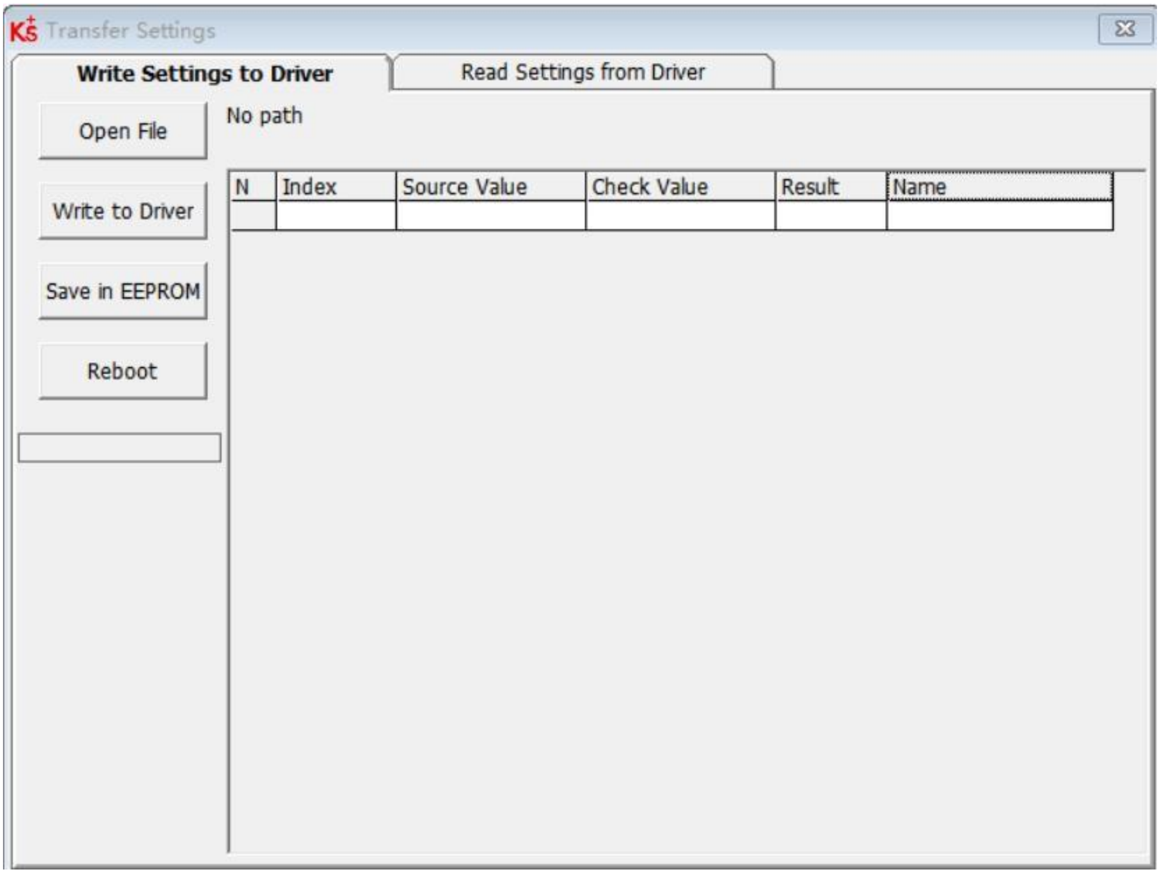
Click **Open File** to select a parameter list file (Kinco_Settings_Without Postable.cdo), The list of parameters is displayed in the window on the right.

Click **Read Settings from Driver** to get the **Drive Value** and **Result**, and then click **Save to File** to save the settings as a .cdi file.

To export the driver's historical failure records, click to open the list and select the errlist.cdo file. It should be noted that the errlist file can only read historical fault records, not driver configuration parameters.

3.4.2 Write Settings to The Drive

Click the menu bar"**Tools**"->"**R/W Diver Configuration**"->"**Write Settings to Driver**", or click button, the following window appears:



N	Index	Source Value	Check Value	Result	Name

Figure 3-7 Write Driver Configuration

Click "**Open File"** to select a parameter file (.cdi) , and the parameters will be displayed in the window on the right.

Click "**Write to Driver**" to get "Check Value" and "Result", "Result" is "False" to indicate that the parameter was not written successfully, most likely the parameter does not exist in the current driver.

Click "**Save in EEPROM**" and then "**Reboot**" for all parameters to take effect.

Note

- When reading the driver configuration, if the object does not exist in the drive, the result will be "False" and marked in red. Only the parameters with the read result of "Ture" will be saved in the .cdi file.
- Disconnect the 485/CAN/EtherCAT bus and disable the drive before writing settings to the drive, as this may prevent some objects from being written successfully.

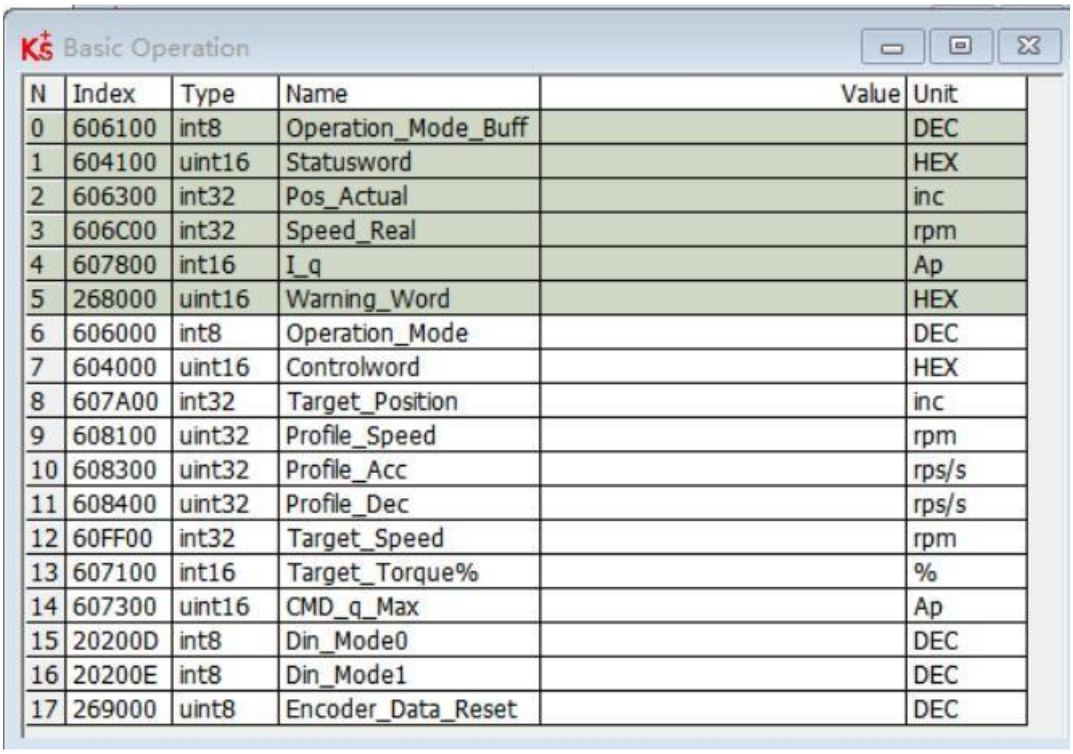
3.5 Velocity Mode Introduction

There are two modes, speed mode 3 and 3, and the control of speed mode can be written by external I/0 and internal instructions.

Table 3–2 Description of the parameters related to the velocity mode

Internal address	No.	Parameter name	Description	Value
60600008	Integer8	Operation_Mode	-3: For immediate speed mode, the actual speed will immediately reach the target speed. 3: For the speed mode with acceleration and deceleration, the actual speed will be accelerated to the target speed according to the acceleration	-3 and 3
60400010	Unsigned16	Controlword	0x0F motor lock shaft; 0x06 motor loose shaft	0x0F
60FF0020	Integer32	Target_Speed	The target speed cannot exceed the rated speed of the motor	Based on user needs
60830020	Unsigned32	Profile_Acc	Mode 3 takes effect	Default 100rps/s
60840020	Unsigned32	Profile_Dec	Mode 3 takes effect	Default 100rps/s

In the "Basic Operation" window of the computer software, we can find these parameters and set them respectively, in sections 6, 7, 10, 11



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

3.6 Digital I/O Functions

Click menu item **Driver**->**Digital IO Functions** or click the I-0 button. The following window appears.

Function and polarity are shown as defaults here.

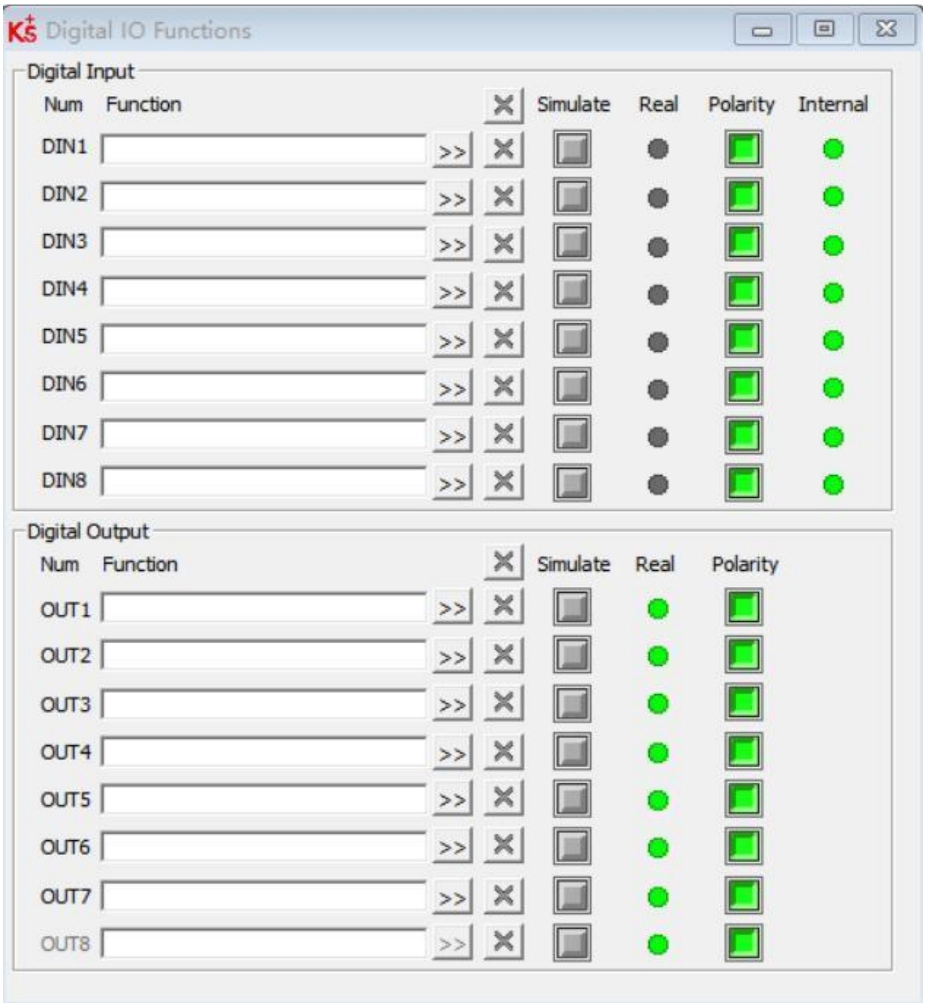


Figure 3-8 Digital input/output

3.6.1 Digital Input

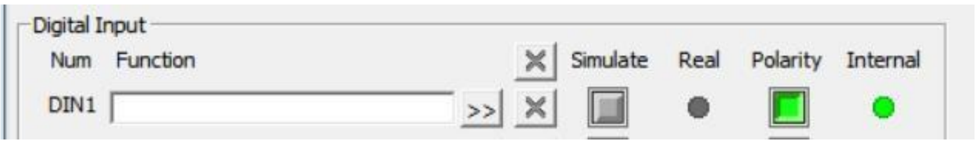


Figure 3-9 Digital input

Function: Click >> to select DIN function setting, click ✕ to delete the DIN function setting.

Simulate: Simulates the digital input active hardware signal.

Real: Shows the real digital input hardware status.

Polarity: 🟩 means Internal is set to 1 by “active” signal. ⬜ means Internal is set to 1 by “inactive” signal.

Internal: This is the result of Simulate, Real and Polarity via the logic formula； mean🟢active”, logic status of the selected function is 1； ● means “inactive”, logic status of the selected function is 0.

DIN function	Description
Enable	Driver enabling 1: Controlword=Din_Controlword (2020.0F) 0: Controlword = 0x06
Reset errors	The bit (bit7) of the reset fault in the control word = 1
Operating mode control	Operation mode selection 1: Operating Mode = Operating Mode Selection 1 (2020.0E) 0: Operating Mode = Operating Mode Selection 0 (2020.0D)
Limit+	Positive/negative limit switch, normally OFF, Din effective input = 0 indicates that the motor has reached the limit position
Limit-	
Invert Direction	Inverts command direction in the velocity and torque mode
Quick stop	Sets the controlword to start quick stop. After quick stop, the controlword needs to be set to 0x06 before 0x0F for enabling (if the enable function is configured in Din, just re-enable it)
Activate command	Activates the position command, for example the control word changes from 0x2F to 0x3F
Pre enable	For safety reasons, Pre_Enable can serve as a signal for indicating whether or not the entire system is ready. 1: driver can be enabled 0: driver cannot be enabled

3.6.2 Digital Output

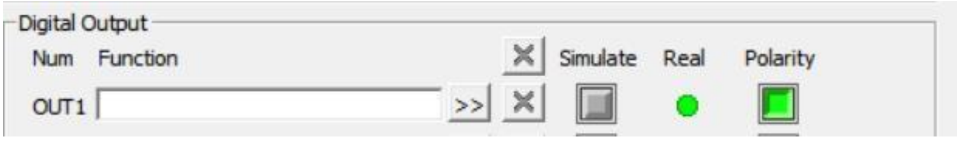


Figure 3-10 digital output

Function: Click >> to select the OUT function setting. Click ✕ to delete the OUT function setting **Simulate**: Simulates the digital output function logic status 1.

Real: Shows the real digital input hardware status. This is the result of Simulate, Polarity and Logic State, 🟢 means that digital input is ON, ● means that digital input is OFF.

Polarity: 🟩means when the logical state is 1, the actual output is ON;▢means when the logic state is 0, the actual output is ON.

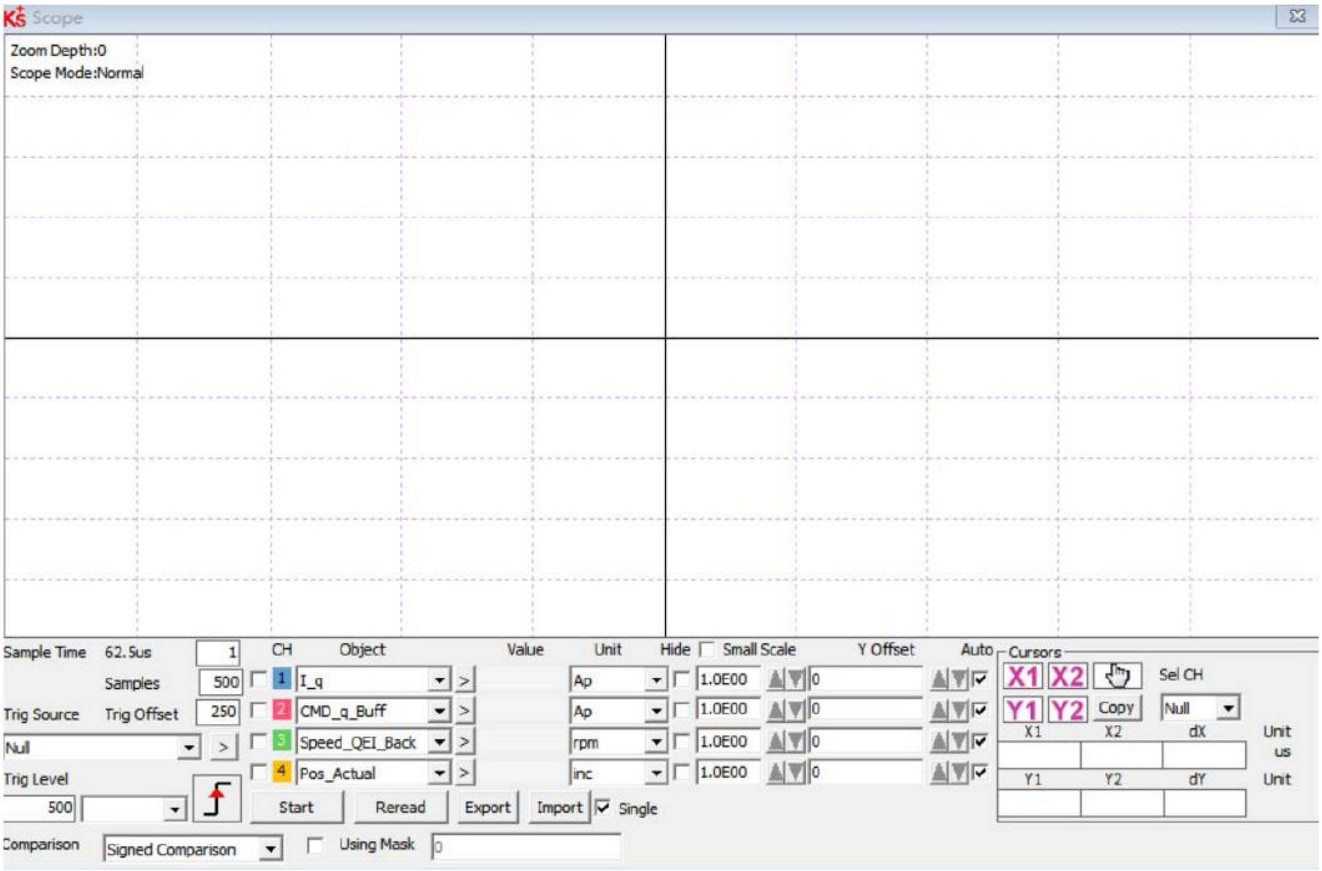
Real: This is the result of Simulate, Polarity and real input. 🟢 means activation logic state of corresponding function is 1. ● means that it is not activated, logic state of corresponding function is 0.

OUT function	Description
Ready	Driver is ready to be enabled
Error	Driver alarms
Zero Speed	Speed_1ms (60F9.1A) |<=Zero_Speed_Window (2010.18) and duration >=Zero_Speed_Time (60F9.14)
Motor brake	The motor brake controls the output signal, if a holding brake motor is used, the function must be set, otherwise the motor will be damaged.
Enc Index	Index signals appear
Speed Limit	In torque mode actual speed reached Max_Speed (607F.00)
Driver Enable	The drive is enabled and the motor is energised to lock the shaft
Position Limit	Position limit function is activated
Torque Reach Limit	When the actual torque (60F5.08) reaches the reference (60F5.06) and the duration exceeds the filtering time (60F5.07) , the output torque reaches the limit, and the torque reaches the reference (60F5.06) set to 0 means that the torque does not open to reach the limit detection.

3.7 Scope

During operation, if the equipment does not perform as expected, or other unexpected occurs, an oscilloscope can be used to analyse the problem.

Click to open the scope window.



KS Scope

Zoom Depth:0
Scope Mode:Normal

Sample Time 62.5us 1
Samples 500
Trig Source Trig Offset 250
Nul
Trig Level
500

CH		Object	Value	Unit	Hide	□ Small Scale	Y Offset	Auto
□	1	I_q		Ap	□	1.0E00	0	☑
□	2	CMD_q_Buff		Ap	□	1.0E00	0	☑
□	3	Speed_QEI_Back		rpm	□	1.0E00	0	☑
□	4	Pos_Actual		inc	□	1.0E00	0	☑

Comparison Signed Comparison □ Using Mask 0

Figure 3-11 Scope Window

Sample Time: The period of data collection, set to 1 represents one data collection every 62.5us.

Samples: Indicates how many data are collected in this sampling, and setting it to 500 indicates that 500 data are collected.

Trig Offset: Number of samples before the trigger event occurs.

Trig Source and Trig Level: The trigger condition is set in Figure 3-11 to start collecting data when the effective target current q rises to 100DEC, DEC is an internal unit, which can be switched to a current un

Trigger edges: Click to change to a rising edge trigger 、 Falling edge trigger or upper and lower edge triggers .

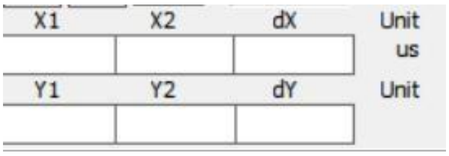
Object: Maximum 64-bit length data can be taken in one sample, e.g.: 2 Int32 objects bit or 4 Int16 objects.

Single: ☑ Single means sample for one trigger event only. □ Single means sample continuously.

Zoom in / zoom out the oscillogram: Hold down the right mouse button and drag the mouse to the lower right to enlarge the scope or drag the mouse to the upper left to zoom out.

Cursors: By clicking on the button Y1 Y2 can select the corresponding cursor, The cursor will appear on the oscilloscope and select the channel to be observed in the Channel Selection drop-down menu.

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



X1	X2	dX	Unit
			us
Y1	Y2	dY	Unit

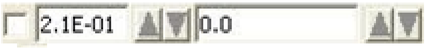
Copy: Copy the sampled data to the pasteboard, you can open Excel to paste the data directly

Moving waveform: When the icon turns yellow, the movement is active, and you can drag the waveform by holding down the left mouse button in the oscilloscope. **Export**: Export the sampled data to a .scope file

Import: Import the .scope file and display the oscilloscope

Reread: Reads the most recently acquired data from the drive and displays an oscilloscope

Auto: If the option box under Auto is checked, the oscilloscope will automatically select the appropriate scale and axis offset for display. If the option box under Auto is not checked, the oscilloscope will be displayed by the scale and offset of the following areas



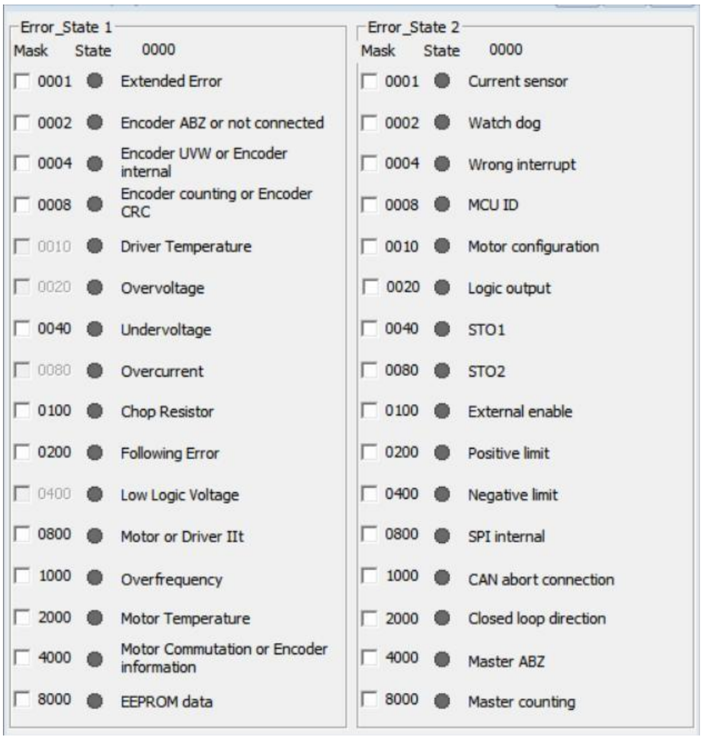
☐ 2.1E-01 ▲▼ 0.0 ▲▼

The values of the scale and offset can via ▲ and ▼ button to increase or decrease. If the small scale option box is checked, the scale increase/decrease corresponding to each button will become 10%

-Import: The oscilloscope is imported by a .scope file, in this mode the Start and Reread Data buttons are disabled, and you can exit the import mode as prompted by the software.

3.8 Error Display And Error History

Error: Click **Driver->Error Display** or click the ● button (which turns red ●), if an error occurs). The Error Display window appears. It shows the last errors. Troubleshooting can be performed according to **Chapter 7** alarm troubleshooting plan.



Error_State 1

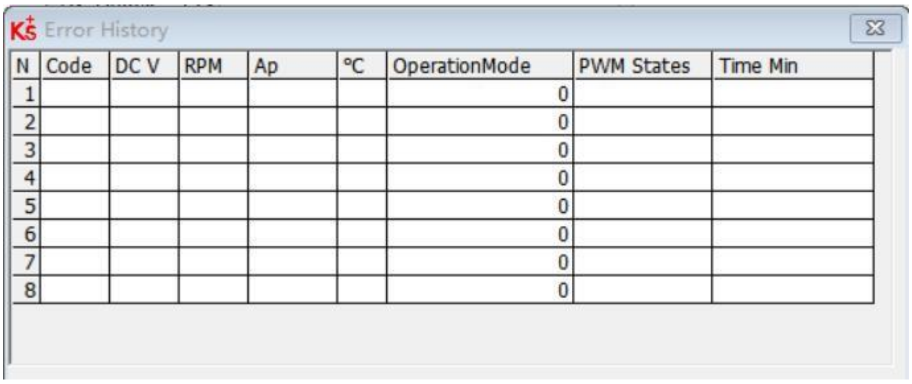
Mask	State	0000
☐ 0001	●	Extended Error
☐ 0002	●	Encoder ABZ or not connected
☐ 0004	●	Encoder UVW or Encoder internal
☐ 0008	●	Encoder counting or Encoder CRC
☐ 0010	●	Driver Temperature
☐ 0020	●	Overvoltage
☐ 0040	●	Undervoltage
☐ 0080	●	Overcurrent
☐ 0100	●	Chop Resistor
☐ 0200	●	Following Error
☐ 0400	●	Low Logic Voltage
☐ 0800	●	Motor or Driver IIt
☐ 1000	●	Overfrequency
☐ 2000	●	Motor Temperature
☐ 4000	●	Motor Commutation or Encoder information
☐ 8000	●	EEPROM data

Error_State 2

Mask	State	0000
☐ 0001	●	Current sensor
☐ 0002	●	Watch dog
☐ 0004	●	Wrong interrupt
☐ 0008	●	MCU ID
☐ 0010	●	Motor configuration
☐ 0020	●	Logic output
☐ 0040	●	STO1
☐ 0080	●	STO2
☐ 0100	●	External enable
☐ 0200	●	Positive limit
☐ 0400	●	Negative limit
☐ 0800	●	SPI internal
☐ 1000	●	CAN abort connection
☐ 2000	●	Closed loop direction
☐ 4000	●	Master ABZ
☐ 8000	●	Master counting

Figure 3-12 Error State Window

Error History: Click menu item“**Driver**”->“**Error History**”, the History Error window will pop up and display the last 8 error messages, including error word, bus voltage, speed, current, temperature, operating mode, power tube status. The most recent historical faults are displayed in the first row.



N	Code	DC V	RPM	Ap	°C	OperationMode	PWM States	Time Min
1						0		
2						0		
3						0		
4						0		
5						0		
6						0		
7						0		
8						0		

Figure 3-13 Error History Window

Table3-3 Error Status (2601.00) Information

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

Table 3-4 Error_state2 (2602.00) information

Bit	Error name	Error code	Description
0	Current sensor	0x5210	Current sensor signal offset or ripple too large
1	Watchdog	0x6010	Software watchdog exception
2	Wrong interrupt	0x6011	Invalid interrupt exception
3	MCU ID	0x7400	Wrong MCU type detected
4	Motor configuration	0x6320	No motor data in EEPROM / motor never configured
5	Reserved		
6	Reserved		
7	Reserved		
8	External enable	0x5443	DIN "pre_enable" function is configured, but the DIN is inactive when the controller is enabled / going to be enabled
9	Positive limit	0x5442	Positive position limit (after homing) – position limit only causes error when Limit_Function (2010.19) is set to 0.
10	Negative limit	0x5441	Negative position limit (after homing) position limit only causes error when Limit_Function (2010.19) is set to 0.
11	SPI internal	0x6012	Internal firmware error in SPI handling
12	CAN bus interrupt	0x8100	A fault alarm is generated only when the communication interruption mode (6007.00) is set to 1

Table 3-5 Error extension (2605.07) information

Bit	Error name	Error code	Description
0	Log the error	0x5210	Current sensor signal offset or excessive ripple
1	The internal brake resistor is over temperature	0x7111	The internal braking resistor is actually too powerful
2	Internal brake resistor short circut	0x7112	The internal brake unit is damaged, the brake circuit is short-circuited
3	The motor is out of phase	0x6321	A phase in the motor power line UVW is not connected
4	The ADC samples saturated	0x2321	The current sampling ADC reaches its limit and the current runs out of control
12	Service timeout	0x81FF	Communication bus error extension

Chapter4 Performance Tuning

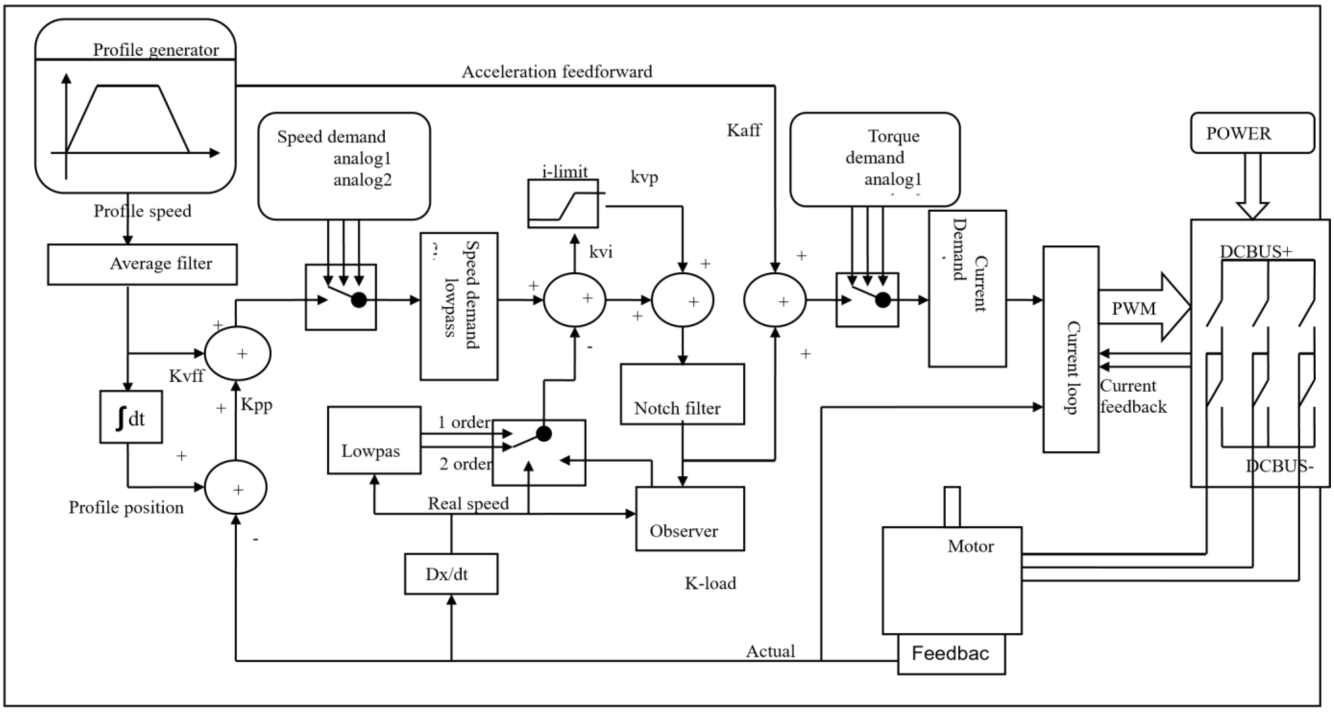


Figure 4－1Servo system control block diagram

Figure 4.1 is the block diagram of the control structure of the servo system, from which the servo system generally includes three control loops: current loop, velocity loop and position loop. For servo systems, good control loop parameters can improve the performance of servos and better meet the process requirements of the site. Therefore, it is very necessary to adjust the parameters of the control loop.

During the debugging process, it is mainly necessary to adjust the velocity loop and position loop parameters. The velocity loop parameter is related to the load inertia converted to the motor shaft by the entire mechanical system. The position loop is the outermost control loop of the servo system, which is related to the motor action mode, that is, the field application. The current loop is the innermost control loop in the servo system, and the current loop parameters are related to the motor parameters. After the motor is configured correctly, the system will default the current loop parameters to the optimal parameters of the equipped motor, so there is no need to adjust them again.

4.1 Tuning Velocity Loop

Table 4–2 List of velocity loop parameters

Internal address	Name	Description	Default	Range
60F90110	Kvp[0]	Used to set the response speed of the velocity loop	/	1~32767
60F90210	Kvi[0]	The time used to adjust the speed control to compensate for small errors, increasing the integral gain will result in greater overshoot.	/	0-1023
60F90710	Kvi/32	This data is 1/32 of KVI and is mainly used for setting up high-resolution encoders	/	0-32767
60F90508	Speed_Fb_N	Speed feedback filtering for velocity loops	7	0~45

		BW=Speed_Fb_N*20+100[Hz]		
60F90608	Speed_Mode	0: 2nd order FB LPF 1: Directly feedback the original velocity 2: Velocity feedback after velocity observer 4: Velocity feedback after 1st order LPF 10: Velocity feedback after 2nd order LPF and the velocity command is filtered by a 1st order LPF. Both filters have the same bandwidth. 11: The velocity command is filtered by a 1st order LPF 12: Velocity feedback after velocity observer, the velocity command is filtered by a 1st order LPF 14: Velocity feedback after 1st order LPF and the velocity command is filtered by a 1st order LPF. Both filters have the same bandwidth	1	/
60F91508	Output_Filter_N	A 1st order lowpass filter in the forward path of the velocity loop	1	1-127
60F90820	Kvi_Sum_Limit	Integral output limit of the velocity loop	/	0-2^15

Step of velocity loop tuning is shown below:

Step 1: Confirm the limit of the velocity loop bandwidth

Velocity loop bandwidth limits position loop bandwidth, so it is especially important to adjust speed loop bandwidth.

The limit of the velocity ring bandwidth can be determined by several aspects:

- Feel the motor vibration and noise through your fingers and ears. It is actually an empirical story, but it's very valid. Users can choose to increase or decrease the velocity loop bandwidth by listening and touching the machine.
- Another way is to observe the oscilloscope, where the user generates a step curve for velocity control and samples the actual velocity and current. Comparing the sampling patterns at different velocity loop bandwidths, we can find the optimal curve – the velocity curve follows the command quickly and without oscillations.

Step 2: Velocity feedback filter adjustment

The velocity feedback filter can reduce noise that comes from the feedback path, e.g. reduce encoder resolution noise.

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

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

The 2nd order filter reduces noise to a greater extent, but it also results in more phase shifting so that velocity loop gain can be limited.

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

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

Step 3: Output filter adjustment

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

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

The filter bandwidth can be calculated using the following formula:

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

Step 4: Velocity loop bandwidth calculation

Use the following formula to calculate velocity loop bandwidth:

$$kvp = \frac{1.853358080\ 10^{5}\ J\pi^{2}\ Fbw}{I_{Max}\ kt\ encoder}$$

Kt——motor torque constant, unit: N.m/Arms*100

J——inertia, unit: kg*m^2*10^6

Fbw ——Velocity loop bandwidth, unit: Hz

Imax——The value of the object 0x651003, unit: DEC Encoder——resolution of the encoder

Step 5: Integral gain adjustment

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

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

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

Step 6: Adjust Kvi_sum_limit

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

4.2 Tuning Position Loop

Table 4–2List of position loop parameters

Internal address	Name	Description	Default	Range
60FB0110	Kpp[0]	Set the position loop response bandwidth, unit: 0.01Hz	10	0～327
60FB0210	K_Velocity_FF	0 means no feedforward, 1000 means 100% feedforward.	100	0～100
60FB0310	K_Acc_FF	Under the premise that the inertia ratio is set correctly, this parameter can be set, if you do not know the inertia ratio, please directly set the position loop acceleration feed forward (0x60FB03)	/	0-32767
60FB0510	Pos_Filter_N	The smooth acceleration and deceleration process needs to be set in the motor loose shaft state	1	1~255
60650020	Max_Following_Error_16	The maximum allowable error, exceeding the changed value will alarm 020.0	10000	/

Step of Position loop tuning is shown below:

Step1: Position loop proportional gain adjustment

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

Step2: Position loop velocity feedforward adjustment

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

The velocity feedforward function can be treated as the upper controller (e.g. PLC) have a chance to directly control the velocity in a position operation mode. In fact this function will expend part of the velocity loop response ability, so if the setting can't match the position loop proportional gain and the velocity loop bandwidth, the overshot will happen.

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

Step3: Position loop acceleration feedforward

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

The acceleration feedforward function can be treated as the upper controller (e.g. PLC) have a chance to directly control the torque in a position operation mode. in fact this function will expend part of the current loop response ability, so if the setting can't match the position loop proportional gain and the velocity loop bandwidth, the overshot will happen.

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

Acceleration feedforward can be calculated with the following formula:

ACC_%=6746518/ K_Acc_FF/EASY_KLOAD*100

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

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

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

→

Note

The smaller the K_Acc_FF, the stronger the acceleration feedforward.

Step4: Smoothing filter

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

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

Step5: Notch filter

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

Antiresonant frequency=Notch_N*10+100

Setting Notch_On to 1 turns on the notch filter. If the resonant frequency is unknown, the user can set the maximum value of the d2.14 current command small, so that the amplitude of system oscillation lies within an acceptable range, and then try to adjust Notch_N and observe whether the resonance disappears.

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

Table 4–3 Notch filter list

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

4.3 Other Factors

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

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

The machine design:

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

Control performance affects the machine's final performance, as well as precision, responsiveness and stability.

Chapter 5 Alarm Exclusion

When drive generates an alarm, red light, ERR will shine.

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

Table 5–1 Alarm codes of Error_State 1

Alarm	Code	Name	Reason	Troubleshooting
000.1		Extended Error	Errors occurs in Error_State2	Open the menu bar of the computer software "Drive" > "Error Display" to view the alarm information of error state2 and refer to **Table 7-2** for alarm content and solution.
000.2	7380	Encoder ABZ signal incorrect (suitable for incremental encoder motor)	Encoder ABZ wiring is wrong or disconnected	1.Check encoder cable is correctly connected 2.Check if corresponding pins of encoder cable is on (refer servo product menu)
	7331	Encoder communication incorrect (suitable for magnetoelectric encoder motor)	The encoder wiring is incorrect or disconnected.	
000.4	7381	Encoder UVW signal incorrect (suitable for incremental encoder motor)	Encoder UVW wiring is wrong or disconnected	1.Check encoder cable is correctly connected 2.Check if corresponding pins of encoder cable is on (refer servo product menu) 3.Change motor
	7320	Encoder internal (suitable for magnetoelectric encoder motor)	Encoder internal is incorrect or encoder is broken	
000.8	7305	Encoder count wrong (suitable for incremental encoder motor)	Encoder is interfered	1.Check encoder cable is correctly connected (different from motor PE cable) 2.Make sure the equipment is well grounded 3.Use isolated power supply to provide power
	7330	Encoder CRC (suitable for magnetoelectric encoder motor)		
001.0	4210	The drive temperature is too high	The temperature of controller's power module has reached the alarm value	1.Add fan, improve the cooling environment of the controller. 2.Add drive installment distance 3.Vertically install drive

Table 5–1 Alarm codes of Error_State 1 *(continued…)*

Alarm	Code	Name	Reason	Troubleshooting
002.0	3210	Overvoltage	Supply power voltage exceeds the allowable input voltage range	1. Check if supply power is higher than standard output voltage 2. Chec if supply power voltage is unstable
			No brake resistor or external brake device is connected	1. Connect suitable braking resistor 2. Open software "Driver"-> “Panel menu”-> “(F005) controller setting” 3. Correctly set "brake resistor value" an "brake resistor power"
			Brake resistor is not configured	1. Change Connect suitable braking resistor 2. Open software "Driver"-> “Panel menu”-> “(F005) controller setting” 3. Correctly set "brake resistor value" an "brake resistor power"
004.0	3220	Undervoltage	The power voltage input is lower than the low voltage protection alarm value.	1. Check if power supply output power can meet with the requirement 2. Change power supply of bigger power
008.0	2320	Short circuit of driver output	Short circuit of driver UVW and PE output	1. Check if motor power cable connection is correct 2. Driver is broken, change driver
010.0	7110	Driver brake resistor is abnormal	The brake resistance parameters are not set correctly	1. Open software "Driver"-> “Panel menu”-> “(F005) controller setting” 2. Correctly set "brake resistor value" an "brake resistor power"
020.0	8611	The error exceeds the allowable error	Stiffness of control loop is too small	1. Open software "Driver"-> “Panel menu”-> “(F005) controller setting” 2. Correctly set "brake resistor value" a "brake resistor power"
			Motor UVW phase sequence is incorrect	1. Open software "Driver" "control loop" "velocity loop" "position loop" 2. Increase "kpp[0]" "kvp[0]"
			The controller and motor together cannot match the requirement of the application	Exchanging wire of U and V
			Max_Following_Error is too small	1. Open software "Driver" "control loop" "velocity loop" "position loop" 2. Increase "max_following_error" (Ensure control loop parameters is fine, user can change this parameter)
040.0	5122	Low logic voltage	Logic voltage is less than 18V, power supply voltage is pulled down	1. Check if power supply output power can meet with requirements 2. Change power supply with bigger power

Table 5–1 Alarm codes of Error_State 1 *(continued…)*

Alarm	Code	Name	Reason	Troubleshooting
080.0	2350	Motor or controller IIt	The brake is not released when the motor shaft is rotating (only for brake motor)	1. Check if brake cable wiring is correct 2. Check brake power can meet with the requirements (output voltage is DC24V, input current is 1A, output power is bigger than 24W)
			Machine equipment stuck or excessive friction	1. Cancel motor enable, or power off driver 2. Please drag load to make it move back and forth in motor's running route. Ensure that there is no machine equipment stuck or excessive friction Add lubricate
			Motor UVW phase sequence is incorrect	Exchange motor wiring of phase U and phase V
100.0	8A80	The input pulse frequency is too high	External input pulse frequency is too high	1. Reduce external pulse input frequency 2. When ensure safely use motor, increase "Frequency_Check" (Open“Driver”-> “Control modes”->“Pulse mode”-> “Frequency_Check”), max 600
200.0	4310	The motor temperature is too high	The motor temperature exceeds the specified value	1. Reduce ambient temperature of the motor and improve cooling conditions 2. Reduce acceleration and deceleration 3. Reduce load
400.0	7122	Motor excitation error (for incremental encoder motors)	Motor UVWphase sequence is wrong	Exchange motor wiring of phase U and phase V
			Encoder is not connected	Check encoder cable
		Encoder information error (for magnetoelectric encoder motors)	Communication is incorrect when the encoder is initialized	Check encoder wiring, restart driver
			The encoder type is wrong, e.g., an unknown encoder is connected	
			The data stored in the encoder is wrong	
			The controller cannot support the current encoder type	
800.0	6310	EEPROM Data errors	Data is damaged when the power is turned on and data is read from the EEPROM	1. Open software “Driver”->“Init Save Reboot” 2. Click “Init Control Parameters”-> “Save Control Parameters”-> “Save 3. Motor Parameters”->“Reboot” Import cdi file by software

Table 5–2 Alarm codes of Error_State2 (extended)

Alarm Code	DS402 Code	Information	Reason	Trouble shooting
000.1	0x5210	Current sensor	Current sensor signal offset or ripple too big	Circuit of current sensor is damaged, please contact the supplier
000.2	0x6010	Watchdog	Software watchdog exception	Please contact the supplier and try to update the firmware
000.4	0x6011	Wrong interrupt	Invalid interrupt exception	Please contact the supplier and try to update the firmware
000.8	0x7400	MCU fault	Check for MCU model errors	Please contact the supplier
001.0	0x6320	The motor is misconfigured	Motor type is not auto-recognised, no motor data in EEPROM / motor never configured	Install a correct motor type to the controller and reboot
010.0	0x5443	Pre-enable alarm	DIN function “pre_enable” is configured, but the input is inactive when the controller is enabled or should become enabled	Solve according to the reason
020.0	0x5442	Positive limit error	Positive position limit (after homing), position limit only causes error when Limit_Function (2010.19) is set to 0	Exclude the condition which causes the limit signal
040.0	0x5441	Negative limit error	Positive position limit (after homing), position limit only causes error when Limit_Function (2010.19) is set to 0	Exclude the condition which causes the limit signal
080.0	0x6012	SPI fault	Internal firmware error in SPI handling	Please contact the supplier
200.0	0x8A81	Fully closed-loop faults	Different direction between motor and position encoder	Change the encoder counting direction
800.0	0x7306	The primary encoder count is incorrect	Master encoder counting error	Ensure that the ground connection and the encoder shield work well.

Chapter 6 RS485 Communication

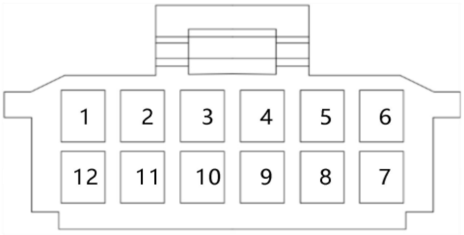
6.1 RS485 Wiring

iWMC integrated servo wheel RS485 port defaults to servo debugging port, and the protocol is RS232 format.

If you need to use the MODBUS communication function to communicate with the PLC, please modify the address of 65100C08 to 0.

When using RS485 for communication control with PLC, please use a CAN card to connect to the Kinco Servo+ software debugging servo.

Table 6-1 RS485 Terminal description

	Pin	Name	Pin function
	7	485A	485IN
	8	485B	
	9	485A	485OUT
	10	485B	

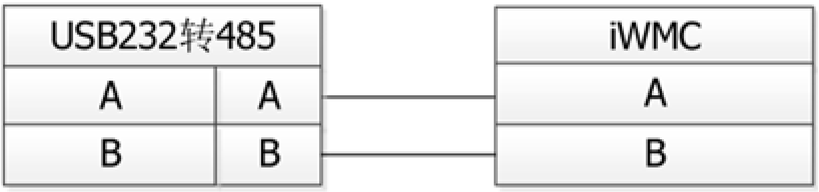


Diagram 6–1 RS485 diagram

6.2 RS485 Communication Parameters

Internal address	Parameter name	Meaning	Default value
100B0010	Device station number	Drive station number	1
2FE20010	RS485 baud rate	Used to set the baud rate of RS485 Setting value Baud rate 1080————9600 540————19200 270————38400 90————115200 Note: You need to save and restart.	540
65100C08	RS485 Communication protocol selection	0: Use the Modbus protocol 1: Use RS232 protocol Note: Itneeds to be set to 0, save and restart.	1
65100E10	RS485 mode	Data = 8, stop = 1, no parity	Fixed value

6.3 MODBUS RTU Communication

The iWMC integrated servo wheel supports the MODBUS RTU communication protocol, and its internal object is a discontinuous 16-bit data register (mapped to 4X when read and written by the host computer). Modbus RTU protocol format:

Station No.	Function code	Data	CRC
1 byte	1 byte	N bytes	2 bytes

6.4 Function Code of Modbus

- Function code 0x03: read data registers Request format:

Station No.	Function code	Modbus address		Read the byte number		CRC
		High byte	Low byte	High byte	Low byte	
1 byte	03	1byte	1byte	1byte	1byte	2byte

Response format:

Station No.	Function code	Return data length	Register data		……	CRC
			High byte	Low byte		
1 byte	03	1 byte	1 byte	1 byte	……	2 bytes

Note

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

- Function code 0x06: write single data register Request format:

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

Response format: If writing successful, then return the same message.

Note

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

- Function code 0x10: Write multiple registers Request format:

Station No.	Function code	Modbus address	Data length (word)		Data length of data-in	Low byte of data		High byte of data		CR C
			High byte	Low byte		High byte	Low byte	High byte	Low byte	
1 byte	10	2 bytes	1 byte	1 byte	1 byte	1 byte	1 byte	1 byte	1 byte	2 bytes

Response registers:

Station No.	Function code	Modbus address	Data length (word)		CRC
			High byte	Low byte	
1 byte	10	2 bytes	1 byte	1 byte	2 bytes

➔

Note

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

For example: send message 01 10 6F 00 00 02 04 55 55 00 08 1A 47

Meaning: 01——ID No.;

10——function code, write multiple WORD;

6F 00——WORD Modbus address for writing data. This is the address corresponding to parameter "Target Velocity" (60FF0020) , data length 2;

00 02——write 2 WORD;

04——data length 4 bytes (2*WORD);

55 55 00 08——write data 00085555 (HEX) , 546133 (OCT) , convert to 200RPM;

1A 47——CRC check.

6.5 Modbus Message Example

Under different modes, message is sent when station No. is 1:

Table 6-2 485 message

Internal address	Name	Meaning	Message (ID=1)
3500	Operation_mode	Operatemode is 3	010635 0000 03C6 07
6F00	Targte_Speed	Speed 150RPM	01106F 0000 020455 5500 081A 47
3100	Controlword	Enable F	010631 0000 0FC7 32
3200	Status word	Read register status	01 03 32 00 00 02 CA B3

Home (Controlword F to 1F)

Internal address	Name	Value	Message (ID=1)	Meaning
60400010	Controlword	F	01 06 31 00 00 0F C7 32	
60600008	Operation_Mo de	6	01 06 35 00 00 06 06 04	
60980008	Home_Method	33	01 06 4D 00 00 21 5E BE	
60990120	Home_Speed_Switch	200RPM	01 10 50 10 00 02 04 55 55 00 08 0E BA	
60990220	Home_Speed Zero	150RPM	01 10 50 20 00 02 04 40 00 00 06 98 76	
60400010	Controlword	1F	01 06 31 00 00 1F C6 FE	
01 03 32 00 00 02 CA B3 read status word, C037 means home found				

Speed

Internal address	Name	Value	Message (ID=1)	Meaning
60600008	Operation_Mode	3	010635 0000 03C6 07	
60FF0020	Target_Speed	150RPM	01106F 0000 020455 5500 081A 47	
60400010	Controlword	F	010631 0000 0FC7 32	
60830020	Profile_Acc	610.352rps/s	Default	
60840020	Profile_Dec	610.352rps/s	Default	

Chapter7 CANOpen Communication

7.1 CANOpen bus communication

CANOpen is one of the most famous and successful open fieldbus standards. It has been widely recognised and applied a lot in Europe and USA. In 1992, CiA (CANinAutomation) was set up in Germany, and began to develop application layer protocol CANOpen for CAN in automation. Since then, members of CiA developed a series of CANOpen products, and applied in many applications in the field of machinery manufacturing such as railway, vehicles, ships, pharmaceutical and food processing etc.

The servo wheel is a standard CAN slave device that strictly follows the CANOpen2.0A/B protocol, and any host computer that supports this protocol can communicate with it. The servo uses a strictly defined object list, which we call the object dictionary, which is designed in a way that is based on the CANOpen international standard, and all objects have clear functional definitions. Some objects such as velocity and position can be modified by external controllers, and some objects can only be modified by the drive itself, such as status and error messages. Examples of these objects are shown in **Table 10-5**.

Table 7-1 Object dictionary example list

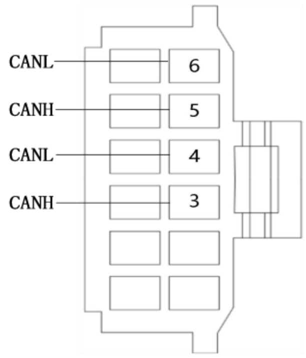
Index	Sub	Bits	Attribute	Meaning
6040	00	16 (=0x10)	RW	Control word
6060	00	8 (=0x08)	RW	Operation mode
607A	00	32 (=0x20)	W	Target position
6041	00	16 (=0x10)	MW	Status word

The attributes of objects are as follows:

1. RW (read & write): The object can be both read and written;
2. RO (read only): The object can be read only;
3. WO (write only): The object can be written only;
4. M (map): The object can be mapping, similar to indirect addressing;
5. S (save): The object can be stored in Flash-ROM without lost after power failure.

7.2 CANOpen Bus Communication Hardware

Table 7-2 Pin name and function description table

Terminals	Pin number	Signal identification	Signal name
	3	CAN_H	CAN in
	4	CAN_L	
	5	CAN_H	CAN out
	6	CAN_L	

CAN communication protocol describes a way of transmitting information between devices. The definition of CAN layer is the same as the open systems interconnection model OSI, each layer communicates with the same layer in another device, the actual communication takes place adjacent layers in each device, but the devices only interconnect by the physical media of the physical layer in the model. CAN standard defines data link layer and physical layer in the mode. The physical layer of CAN bus is not strictly required, it can use a variety of physical media such as twisted pair Fibre. The most commonly used is twisted pair signal, sent by differential voltage transmission (commonly used bus transceiver). The two signal lines are called CAN_H and CAN_L. The static voltage is approximately 2.5V, then the state is expressed as a logical 1, also called hidden bit. It represents a logic 0 when CAN_H is higher than the CAN_L, we called it apparent bit, then the voltage is that CAN_H = 3.5V and CAN_L= 1.5V, apparent bit is in high priority. The names and functions of the CAN communication interface pins are shown in Table 7-4.

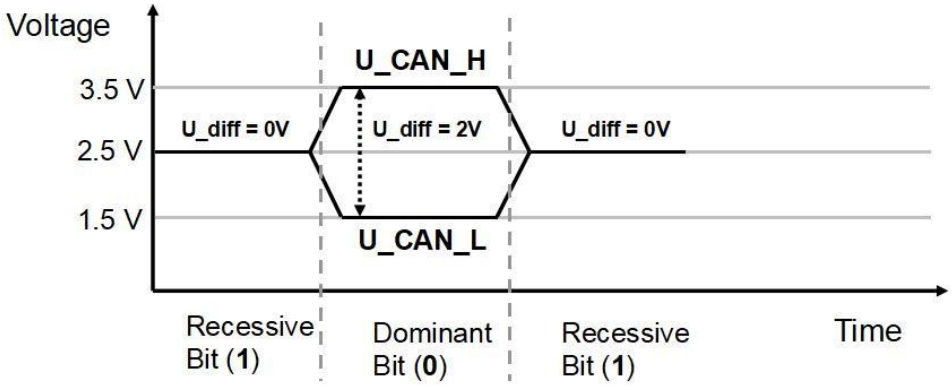


Table 7-3 CAN Signal identification

Note:

1. The CAN_L and CAN_H feet of all slaves can be directly connected, and the wiring is carried out in series.
2. Please use shielded twisted pair as far as possible for communication cables.
3. The longest distance that can theoretically communicate with various baud rates is shown in Table 10-7.
4. The servo wheel does not need to be connected to an external 24V power supply to supply power to CAN.

Table 7-4 The max. distance at different baud rate are shown in following table (Theory)

Communication speed (bit/s)	Communication distance (M)
1M	25
800K	50
500K	100
250K	250
125K	500
50K	600
25K	800
10K	1000

7.2.1 CANOpen bus communication software

EDS Introduction

EDS (Electronic Data Sheet) file is an identification documents or similar code of slave device, to identify what kind of slave device is (Like 401, 402 and 403, or which device type of 402) .This file includes all information of slaves, such as manufacturer, sequence No., software version, supportable baudrate, mappable OD and attributes of each OD and so on, similar to the GSD file for Profibus. Therefore, we need to import the EDS file of slave into the software of master before we configure the hardware.

SDO Introduction

SDO is mainly used in the transmit the low priority object between the devices, typically used to configure and manage the device, such as modifying PID parameters in current loop, velocity loop and position loop, and PDO configuration parameters and so on. This data transmission mode is the same as Modbus, that is it needs response from slave when master sends data to slave. This communication mode is suitable for parameters setting, but not for data transmission frequently.

SDO includes upload and download. The host can use special SDO instructions to read and write the OD of servo. In CANOpen protocol, SDO (Service Data Object) can be used to modify object dictionary. SDO structure and guidelines are shown below:

SDO basic structure is: Client→ Server/Server→ Client

Byte0CAN frame data byte	Byte1-2	Byte3	Byte4-7
length, trigger bit for alternate zeroing and assertion of each subsequent segment (toggle bit)			
SDO Command specifier	Object index	Object subindex	Max 4 bytes data

The SDO command word contains the following information:

- Download/upload
- Request/response
- Segmented/expedited transfer
- CAN frame data byte length, trigger bit for alternating zeroing and asserting of each subsequent segment (toggle bit)

Five request/reply protocols are implemented in SDO:

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

Among them, Download refers to the write operation of the object dictionary, and upload refers to the reading operation of the object dictionary; When reading parameters, use the Initiate Domain Upload protocol; When setting parameters, use the Initiate Domain Download protocol; The SDO command word (first byte of the SDO CAN message) syntax of the protocol is described in Tables 10-8 and 10-9, where "-" indicates irrelevance and should be 0) .

Table7-5 Startup download

Bit	7	6	5	4	3	2	1	0
Client→	0	0	1	-	n		e	s
←Server	0	0	1	-	-	-	-	-

Table 7-6 Startup upload

Bit	7	6	5	4	3	2	1	0
Client→	0	0	1	-	-	-	-	-
←Server	0	0	1	-	n		e	s

Description:

n—— Indicates the number of bytes of meaningless data in the message data [from (8-n) bytes to 7th byte of meaningless data] (n is valid when e=1 and s=1, otherwise n is 0.);

e—— When e=0, the transmission is normal, and when e=1, the transmission is accelerated;

s —— Indicates whether the data length is specified, 0 indicates the data length is not specified, and 1 indicates the data length.

e=0, s=0——retained by the CiA;

e=0, s=1——The data byte is the byte counter, byte4 is the data low-bit part (LSB), and byte7 is the high-bit part of the data (MSB) ;

e=1——The data bytes are the data that will be downloaded.

The format of sending and receiving SDO packets when reading parameters is shown in Tables 7-6 and 7-7.

Table 7-7 Sent SDO message when read parameters

Identifier	DLC	Daten							
		0	1	2	3	4	5	6	7
0x600+Node_ID	8	Send command word	Object index		Object subindex	00			

Table 7-8 Receive SDO message when read parameters

Identifier	DLC	Daten							
		0	1	2	3	4	5	6	7
0x580+Node_ID	8	Receive command	Object index		Object subindex	Max 4 bytes data			

Note:

When SDO message are sent, commands are 0x40;

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

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

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

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

When modifying parameters, the format of sending and receiving SDO packets is shown in Table 7-8 and 7-9.

Table7-9 Send SDO message (Modify parameters)

Identifier	DLC	Daten							
		0	1	2	3	4	5	6	7
0x600+Node_ID	8	Send command	Object index		Object subindex	Max 4 bytes data			

Table 7-10 Receive SDO message (Modify parameters)

Identifier	DLC	Daten							
		0	1	2	3	4	5	6	7
0x580+Node_ID	8	Receive command	Object index		Object subindex	Max 4 bytes data			

Note:

The SDO packet is sent successfully, and the receiving command word is 0x60; The SDO packet failed to be sent, and the command word received is 0x80.

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

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

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

If the SDO packet fails to be sent, you can troubleshoot the problem based on the error code of the reply.

Table7-11 SDO Message error code

Error code	Code feature description
0x05040001	Invalid command, unknown or illegal Client/Server command word
0x06010001	An attempt was made to read a write-only object parameter
0x06010002	An attempt was made to write a read-only object parameter
0x06020000	Invalid index, the object does not exist in the object dictionary
0x06040041	Cannot be mapped, object parameters do not support mapping to PDO
0x06060000	The drive is in an error-fault state, causing object parameter access to fail
0x06070010	The data types do not match, and the service parameter lengths do not match
0x06070012	The data types do not match, and the service parameter length is too large

0x06070013	The data types do not match, and the service parameter length is too short
0x06090011	Invalid subindex
0x06090030	Invalid data, outside the scope of object parameter settings
0x06090031	The number of written data is too large
0x06090032	The number of written data is too small
0x08000022	Data cannot be transferred or saved to the app due to the current device state

Table 7-12 Set the velocity mode via SDO messages

Parameter address	Name	value	message (ID=1)
60600008	Operation_Mode	3	send→601 2F 60 60 00 03 00 00 00 receive←581 60 60 60 00 03 00 00 00
60FF0020	Target_Speed	-100RPM	send→601 23 FF 60 00 7E B1 E4 FF receive←581 60 FF 60 00 7E B1 E4 FF
60400010	Controlword	2F	send→601 2B 40 60 00 2F 00 00 00 receive←581 60 40 60 00 2F 00 00 00
60830020	Profile_Acc	100rps/s	send→601 23 83 60 00 6E A3 01 00 receive←581 60 83 60 00 6E A3 01 00
60840020	Profile_Dec	100rps/s	send→601 23 84 60 00 6E A3 01 00 receive←581 60 84 60 00 6E A3 01 00

Note: The message is represented in decimal 16, and the motor resolution used in this case is 65536

PDO Introduction

PDO can transport 8 bytes of data at one time, and no other protocol preset (Mean the content of the data are preset) , it is mainly used to transmit data in high frequency. PDO uses brand new mode for data exchange, it needs to define the data receiving and sending area before the transmission between two devices, then the data will transmit to the receiving area of devices directly when exchanging data. It greatly increase the efficiency and utilization of the bus communication.

PDO COB-ID introduction

COB-ID is a unique way of CANOpen communication protocol, it is the short name of Communication Object Identifier. These COB-ID defines the respective transmission levels for PDO. These transport level, the controller and servo will be able to be configured the same transmission level and the transmission content in the respective software. Then both sides know the contents of data to be transferred, there is no need to wait for the reply to check whether the data transmission is successful or not when transferring data.

The default ID allocation table is based on the CAN-ID (11 bits) defined in CANOpen 2.0A (The COB-ID of CANOpen 2.0B protocol is 27 bits), include function code (4 bits) and Node-ID (7 bits) as shown in **Diagram7-12.**

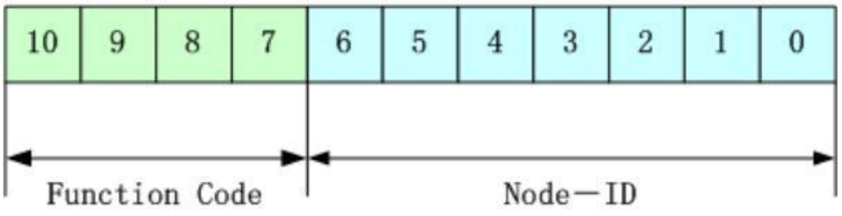


Figure 7-12 Default ID allocation table

Node-ID ——Defined by the driver, set by the device station number, the number Node-ID range is 1~127 (0 is not allowed to be used);

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

Table7-13 The allocation table for CAN identifiers in master/slave connection set predefined by CANOpen

CANOpen predefines the broadcast object for the master/slave connection set			
Object	Function code (ID-bits 9-7)	COB-ID	The index of the communication parameter in OD
NMT Module Control	0000	000H	-
SYNC	0001	080H	1005H, 1006H, 1007H
TIME SSTAMP	0010	100H	1012H, 1013H
CANOpen a peer object for the master/slave connection set			
Object	Function code (ID-bits 9-7)	COB-ID	The index of the communication parameter in OD
Emergency	0001	081H-0FFH	1024H, 1015H
PDO1 (Send)	0011	181H-1FFH	1800H
PDO1 (Receive)	0100	201H-27FH	1400H
PDO2 (Send)	0101	281H-2FFH	1801H
PDO2 (Receive)	0110	301H-37FH	1401H
PDO3 (Send)	0111	381H-3FFH	1802H
PDO3 (Receive)	1000	401H-47FH	1402H
PDO4 (Send)	1001	481H-4FFH	1803H
PDO4 (Receive)	1010	501H-57FH	1403H
SDO (Send/Server)	1011	581H-5FFH	1200H
SDO (Receive/Client)	1100	601H-67FH	1200H
NMT Error Control	1110	701H-77FH	1016H-1017H

Note:

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

2. The function codes of COB-ID in every level are fixed;
3. COB-ID of 00H, 80H, 100H, 701H-77FH, 081H-0FFH are system management format;
4. COB-ID supported by the servo

Send PDO (TXPDO)

Send PDO of servo means servo sends out data, and these data are received by PLC. The function codes of send PDO (COB-ID) are as follow:

- 0x180+Station No. of Servo
- 0x280+Station No. of Servo
- 0x380+Station No. of Servo
- 0x480+Station No. of Servo

Receive PDO (RXPDO)

Receive PDO of servo means servo receive data, and these data are sent by PLC. The function codes of receive PDO (COB-ID) are as follows:

1. 0x200+Station No. of Servo
2. 0x300+Station No. of Servo
3. 0x400+Station No. of Servo
4. 0x500+Station No. of Servo

As long as the controller and servo are defined according to this.

PDO transmission types

PDO support two transmission mode:

SYNC——Transmission is triggered by the synchronization message (Transmission type: 0-240)

In this transmission mode, controller must have the ability to send synchronous messages (The message is sent periodically at a maximum frequency of 1KHz) , and servo will send after receiving the synchronous message.

Acyclic: Pre-triggered by remote frame, or by specific event of objects specified by the equipment sub-protocol.

In this mode, servo will send out data as soon as receiving the data of synchronous message PDO.

Cyclic: Triggered after sending 1 to 240 SYNC messages.

In this mode, servo will send out data in PDO after receiving n SYNC messages.

ASYNC (Transmission type: 254/255)

Slave sends out message automatically as soon as the data change, and it can define an interval time between two messages which can avoid the one in high priority always sending message. (The smaller number of PDO, the higher its priority)

Sending PDO (TPDO) supports synchronous and asynchronous transmission modes, and the corresponding transmission type can be selected according to the transmission mode. For receiving PDO (RPDO), when the driver node is turned on in non-interpolated mode, as long as the RPDO packet sent by the bus is detected, the object data will be received in real time, which has nothing to do with the transmission type setting. In interpolation mode, the driver receives data first after detecting an RPDO signal, but only updates the object data at a specific point in time.

PDO Forbidden time

A PDO can specify a suppression time, that is, define the minimum interval between two consecutive PDO transmissions, to avoid the problem that because the amount of data with high-priority information is too large, it always occupies the bus, and other lower-priority data cannot compete for the bus. The prohibition time is defined by a 16-bit unsigned integer in 100ms units.

PDO Event time

In asynchronous transmission mode, the drive sends PDO messages to the controller in ms. Note When using event time, the suppression time should be set to 0.

Protection mode (Supervision)

Supervision type is to choose which way master uses to check slave during operation, and check whether slave is error or not and handle the error!

1. Master station heartbeat message

The slave station periodically uploads messages to the master station with "supervision time", if the master station has not received the next heartbeat message from the slave after the "heartbeat consumer time", then the master station judges that the communication is wrong, and the master station generates an alarm!

Table 7-14 Heartbeat packet format uploaded from slave station

COB-ID	Byte 0
0x700+Node_ID	state
Case messages (Slave station ID=1): 701 05	

2. Slave station heartbeat message

The master sends messages to the slave periodically with the "supervision time", if the slave has not received the next heartbeat message from the master station after the "heartbeat producer time", then the slave determines that the communication is wrong! When the communication interruption mode (0x600700 setting) is 1, the drive alarm stops when the CAN communication error occurs.

Table 7-15 Heartbeat packet format uploaded from master station

COB-ID	Byte 0
0x700+Master station ID	Master station state
Case messages (Master station ID=127): 77F 05	

Table7-16 State value meaning

State value	Meaning
0x00	Boot-up
0x04	Stopped
0x05	Operational

0x7f	Pre-operational

When a Heartbeat node starts, its boot-up message is its first Heartbeat message.

➔

Note

The heartbeat message generation time and the slave heartbeat message are configured by the master station power-on, and the default power failure is not saved.

3. Node protection

The master sends remote request packets to the slave periodically with "supervision time", and the slave responds after receiving it, if the master station has not received the slave response message after exceeding the "supervision time * life factor" time, the master determines that the slave is wrong. At the same time, the slave can also monitor the remote request status of the master, start the communication protection from the first remote frame received, if the "node protection time * node protection factor" time is exceeded and the master remote frame is not received, the slave will also judge the communication error. The communication interruption mode (0x600700) needs to be set to 1, and the drive will alarm and stop when the CAN communication error occurs.

Master request packet format - (0x700 + node number) (no data in this message)

The slave response packet format - (0x700 + node number) + status

Table7-17 Slave replied message

COB-ID	Byte 0	
0x700+Node_ID	Bit7: Trigger bits	Bit6-Bit0: status

Table7-18 The meaning of the slave reply packet status value

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

Status – The data section includes a trigger bit (bit7), which must alternate "0" or "1" in each node protection response. The trigger bit is set to "0" on the first node protection request. Bits 0 to 6 (bit0~6) are used to indicate the node status, and the numerical meaning is shown in Table 7-18.

Standard CAN slaves generally support only one node protection method, FD5 series servo drives support both protection methods. However, a node cannot support both node protection and heartbeat packets, so only one of them can be selected as protection.

Boot-up process

During the process of internet initialisation, CANOpen support extending boot-up and support min boot-up process. The boot-up process is shown in following figure 7-18.

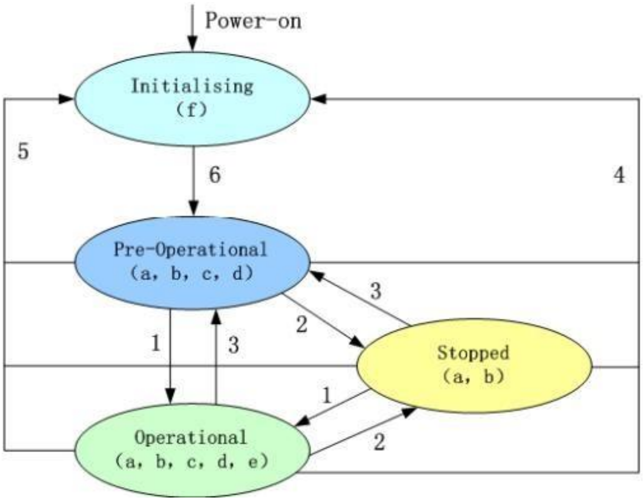


Figure 7-18 Boot-up process

Note: The letters in parentheses in the figure indicate the communication objects that can be used when in different states.

a——NMT	d——Emergency	b——Node Guard
e——PDO	c——SDO	f——Boot-up

Only the NMT-Master node can transmit NMT Module Control messages, all slave devices must support NMT Module Control services, and NMT Module Control messages do not need to be answered. After initialisation, the device automatically enters the Pre_Operational state and sends a boot-up message. The NMT message format is as follows: NMT-Master→ NMT Slave (s)

Table 7-19 NMT Management message format

COB-ID	Byte0	Byte1
0x000	CS	Node-ID

When Node-ID is 0, all NMT slave devices are addressed. CS is command, value table is shown in table 7-20.

Table 7-20 Value table

Command	NMT service
0x01	Open node
0x02	Close node
0x80	Come to pre-operation
	status
0x81	Reset node
0x82	Reset communication

Description of the emergency message

When a fatal error occurs inside the device, an emergency message is triggered and sent by the application device to other devices with the highest priority. An emergency message consists of 8 bytes.

Table 7-21 Emergency message format

COB-ID	Byte 0-1	Byte2	Byte4-5	Byte6-7
Emergency message station number 0x101400	Contingency error code 0x603F00	Error register (0x100100)	Error status 0x260100	Error status 0x260200

Table 7-22 Emergency error code0x603F00

Alarm content	Emergency error codes (Hex)	Alarm content	Emergency error codes (Hex)
The communication encoder is not connected	0x7331	Current sensor failure	0x5210
Communication encoder multiturn error	0x7320	Software watchdog reset	0x6010
Communication encoder verification error	0x7330	Abnormal interruption	0x6011
The drive temperature is too high	0x4210	MCU failure	0x7400
The driver bus voltage is too high	0x3210	The motor model is misconfigured	0x6320
The driver bus voltage is too low	0x3220	The motor power line is out of phase	0x6321
The drive power section is short-circuited, or the motor is short-circuited	0x2320	Pre-enable alarm	0x5443
Current sampling saturation	0x2321	The positive limit bit is reported as an error	0x5442
The drive brake resistance is abnormal	0x7110	Negative limit bit error reported	0x5441
The actual following error exceeds the allowable error	0x8611	SPI failure	0x6012
Low logic voltage	0x5112	Bus communication error	0x8100
The motor or drive is overloaded	0x2350	Bus communication timed out	0x81FF
The input pulse frequency is too high	0x8A80	Full closed-loop check for errors	0x8A81
The motor temperature is too high	0x4310	Main encoder ABZ failure	0x7382
The communication encoder did not respond	0x7331	The primary encoder count is incorrect	0x7306
EEPROM data error	0x6310		

Table7-23 Error register

Bit	Error type
0	General error
1	Current
2	Voltage
3	Temperature
4	Communication error
5	Device profile specific
6	Encoder
7	Retain

7.2.2 CANOpen bus communication settings

This section will introduce the setting of CAN bus communication parameters, click Driver->ECAN Configuration-> other in the host computer software interface to enter the parameter setting interface. The master station with network management function will initialize the parameters of the slave by sending SDO, and under normal circumstances, parameters such as synchronization ID, node protection time, node protection time coefficient, node protection station number, emergency message station number, and heartbeat message generation time do not need to be set by the user.

Table7-24 CANOpen communication parameters

Internal address	Parameter name	meaning	Default value
2FF00108	Control loop parameters storage	1: Store all set parameters except motor 10: Initialize all saveable parameters except the motor	0
100B0008	Device station number	Drive station number Note: Changing this parameter requires saving and rebooting with d5.00.	1
2F810008	CAN baud rate	CAN baud rate setting Setting value baud rate 100 1M 50 500k 25 250k 12 125k 5 50k 1 10k Note: You need to save and then restart.	50
60070010	Communication interruption mode	CAN communication interrupt mode 0: Not processed 1: Error	0

Internal address	Parameter name	meaning	Default value
10050020	Synchronization ID	The synchronous packet COB-ID is valid when the transmission type is 1-240 synchronous mode, and the asynchronous mode does not need to be set	80
100C0010	Node protection time	Through node protection, the master can monitor the current status of each node, the master sends	1000
100D0008	Node protection time factor	remote frames to ask the node status in the node protection time cycle, the node needs to respond within the node protection time * node protection time coefficient, otherwise the master determines that the slave is disconnected, when the communication interruption mode is 1, the drive alarms.	3
100E0020	Node protection ID	700+Device station number (0x100B00)	701
10140020	Emergency message station number	80+Device station number (0x100B00)	81
10170010	The time of heartbeat generated	The slave station periodically sends a message to the master station with the "heartbeat message generation time", and the master station does not receive the message for more than a certain period of time, and the slave station is disconnected, and the master station alarms. The heartbeat packet generation time data is not saved, and the unit is ms, and the data format should be noted as DEC	0
10160120	Heartbeat from the station	Bit24~31: Invalid data Bit16~23: Used to set the master ID Bit0~15: Used to set the heartbeat message detection interval in ms For example, the value of 7F03E8 means that the ID of the master is 127, and the interval between heartbeat packets sent by the detection master is 1000ms The heartbeat packet data of the slave station is not saved when power down, and is configured by the master station on power-on, and it should be noted that the data format is HEX	7F0000