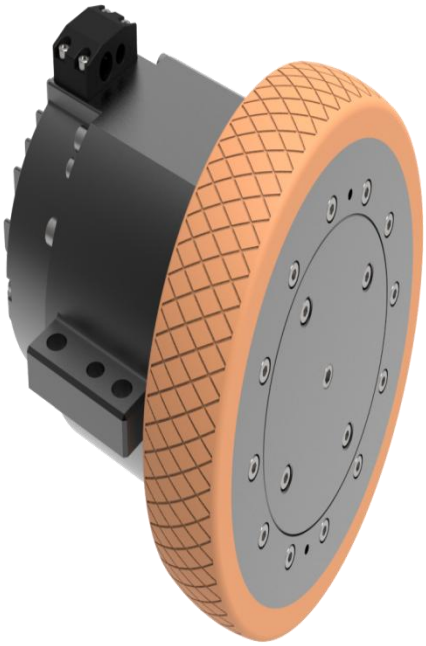
※ AMD Servo Integrated Wheel Set ※

KY165ACS0405-02B

KY180ACS0410-02B

User's Manual (V1.4.1)



KY165ACS0405-02B

KY180ACS0410-02B

Jinan Keya Electron Science And Technology Co., Ltd.

Catalog

Use warning:

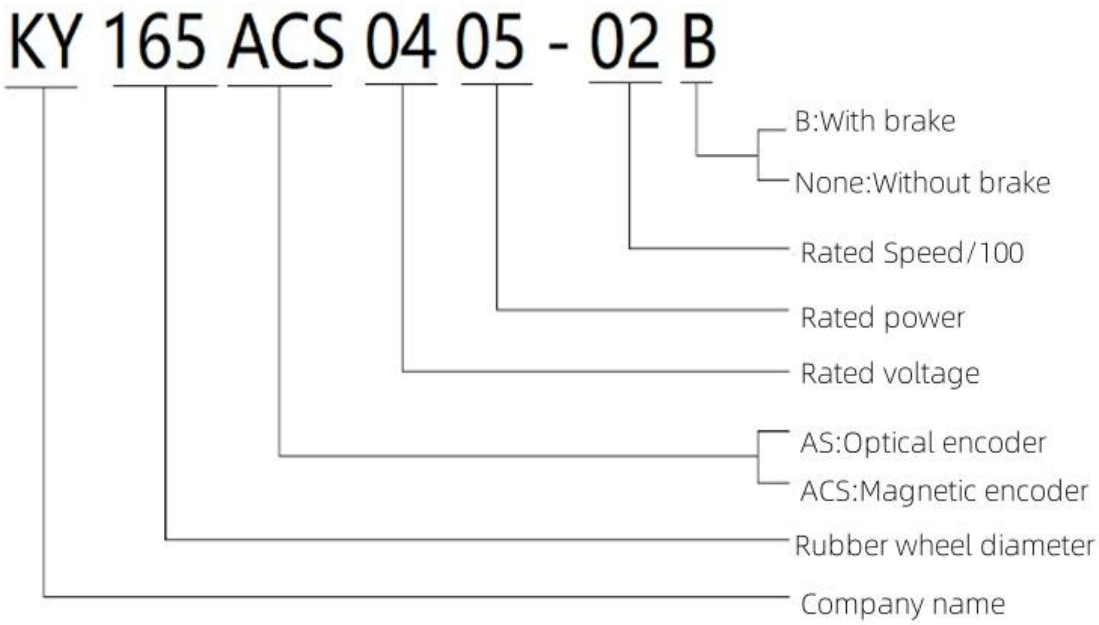
DANGER

1.No anti-reverse connection function inside the motor, the first time you use it, you should confirm the power supply first, and only after confirming the voltage range and the wiring are correct, you can carry out the power-on operation.

2.If there is a fault alarm in the trial operation, please check the reason according to the fault table first, and try the operation again after the fault is clear.

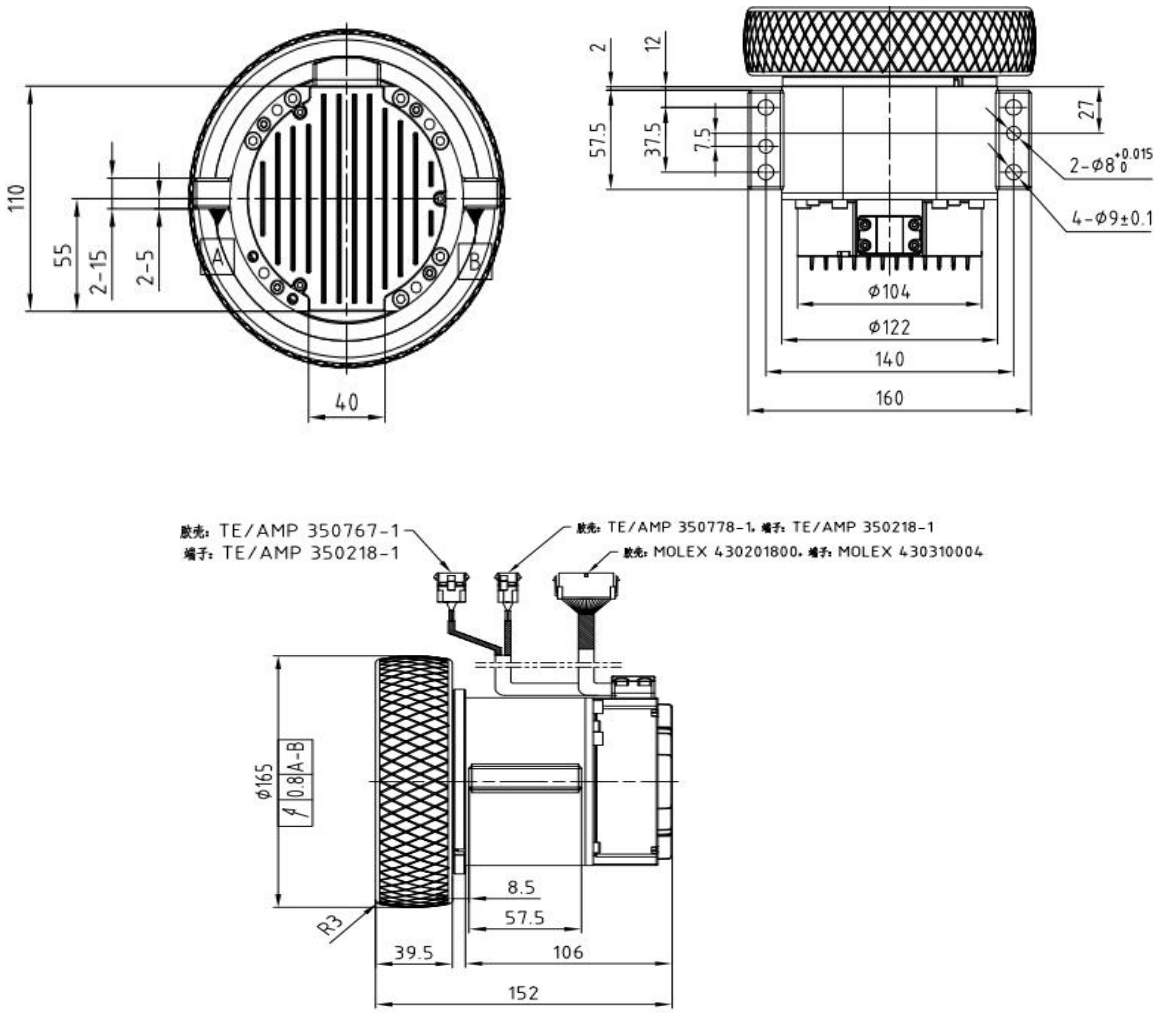
Ⅰ Overview

1.Model description



2.Installation dimension drawing

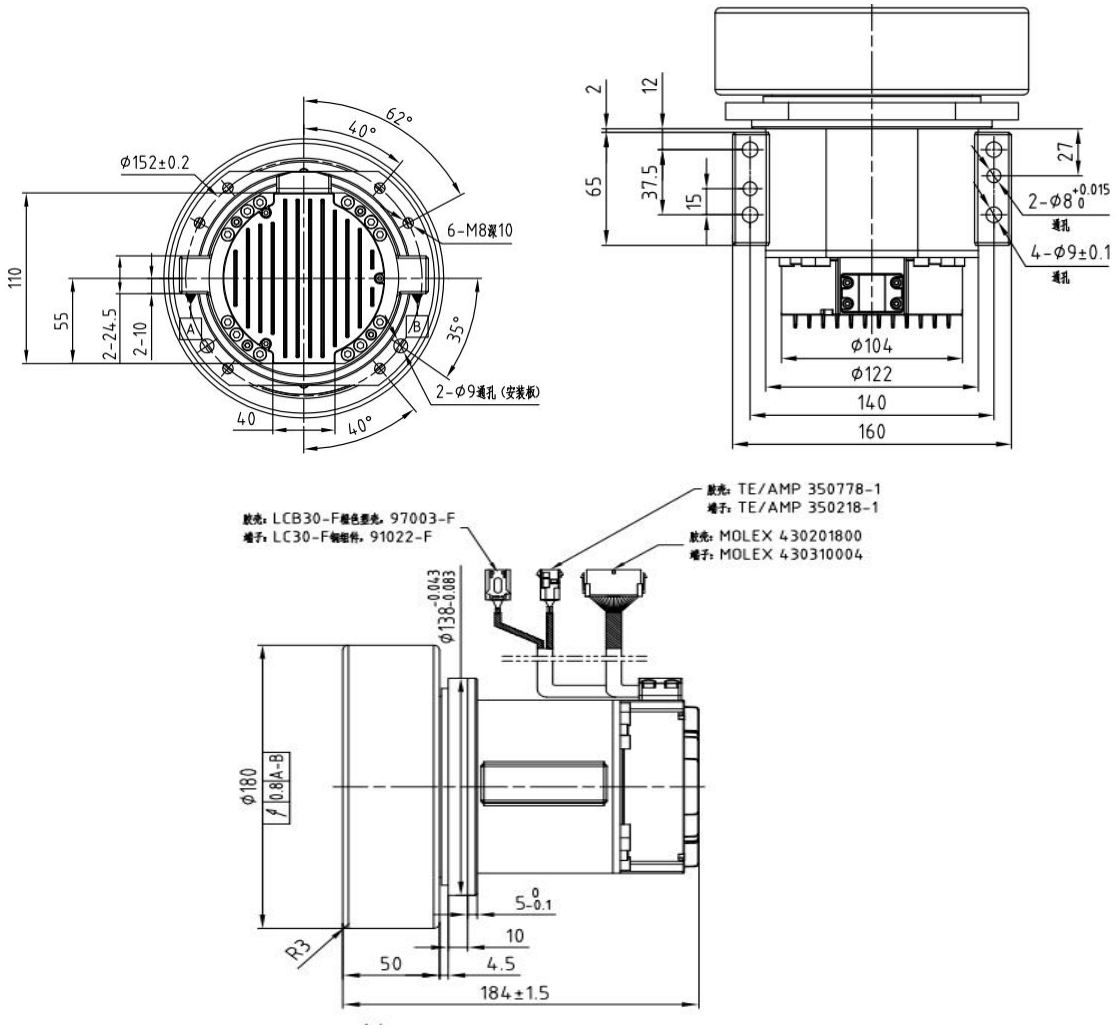
- KY165ACS0405-02B



Remarks:

1. Rubber wheel material: polyurethane; anti-static: resistance 10^5 Ω≤ R≤10^7 Ω; hardness: 85A±2; material tensile strength: ≥35MPa; elongation at break: ≥450%; tear strength: ≥73kN/m; operating temperature: -50-80℃ (rubber wheel only);
2. Motor noise: 65dB(A)(@1m,2000rpm), and there shall be no obvious noise, foreign sound;
3. Maximum load of single motor: 300Kg;
4. Working system: S2-60min.
5. Rated line speed: 1.9m/s.

- KY180ACS0410-02B



Remarks:

1.Rubber wheel material: polyurethane (anti-static); hardness: 93A ± 2; material tensile strength: ≥40MPa; elongation at break: ≥440%; tear strength: ≥105kN/m; operating temperature: -50-80 ℃ (only rubber wheel);

2. Motor noise: 65dB(A)(@1m,2500rpm), and there shall be no obvious murmur, strange sound;

3. Maximum load of single motor: 400Kg for side lug mounting; 700Kg for flange mounting;

4. Working system: S3,40%@10min.

5. Rated line speed: 1.57m/s

3.Operator requirements

This product can only be operated by an electrical engineer who is familiar with the following regulations;

- Operation and installation of the electrical system;
- Applicable provisions for operating safety engineering systems;
- Applicable provisions for accident protection and occupational safety;
- Product documentation.

4.Electrical requirements

- Input voltage: 24V-60V;
- Holding brake control voltage: 24V (conventional drive self-control);
- Over-voltage alarm point: 68V (upper computer can be set);
- Under-voltage alarm point: 18V (the upper computer can be set).

5.Environmental requirements

- Use temperature: -25～55℃ (subject to ambient temperature), storage temperature: -35～65℃ (subject to ambient temperature);
- Humidity: 5%--90%RH, with condensation (25℃);
- Protection grade: IP54;
- Insulation performance: input to the chassis DC600V, leakage current 0.07mA. insulation resistance of 20MΩ or more;
- Three-proof requirements: to meet the conventional three-proof requirements (dust, moisture, salt spray);
- Vibration requirements: frequency 5HZ ~ 25HZ, amplitude 3mm, 0.09g. 25HZ ~ 200HZ, amplitude 1.47mm, 116g. Horizontal, vertical, longitudinal each direction for 30min;
- Cooling method: natural cooling.

Ⅱ Functions and Technical Indexes

1.Main function

- ●Operating modes: speed mode, torque mode, position mode;
- ●Feedback element: magnetic encoder;
- ●Control ports: CAN2.0, CANopen, RS232, RS485;
- ●Can be networked for control and monitoring via CAN bus;
- ●Motor speed control and data reading through RS232;
- ●Internal drive temperature monitoring and over-temperature protection;
- ●Over-current protection;
- ●Over-voltage and under-voltage protection;
- ●Blocking rotation, flying rotation protection;
- ●Motor short circuit protection .

2.Technical parameters

- ● Parameters of the whole machine

Parameter	KY165ACS0405-02B	KY180ACS0410-02B	Unit
Rated voltage	48	48	VDC
Logic supply voltage (not connected by default)	24	24	VDC
Power rating	0.5	1.05	kW
Drain voltage	63	63	VDC
Rated torque of wheel set	21	54	N.m
Rated linear speed of wheel set	1.9m/s	1.57m/s	m/s
Rated rotation speed of wheel set	222	167	r/min
Rated motor torque	2.4	4	N.m
Rated motor speed	2000	2500	r/min
Maximum continuous current	13	26	A
Maximum peak current	40 (1s over current protection)	55 (1s over current protection)	A

Communication port	RS232 ---- 115200	RS232 ---- 115200	bps
	CAN ----125/250/500	CAN ----125/250/500	Kbps
	RS485 ---- 115200	RS485 ---- 115200	bps
PWM switching frequency	10	10	kHz
Under voltage protection	18 kHz	18	V
Over voltage protection	68	68	V

- Gearbox parameters

Parameter	KY165ACS0405-02B	KY180ACS0410-02B	Unit
Speed ratio	9	15	
Rated torque	30	110	N.m
Peak torque	60	275	N.m
Allowable radial load Fr	3000	7000	N
Allowable axial load	1000	1500	N

- Brake parameters

Parameter	KY165ACS0405-02B	KY180ACS0410-02B	Unit
Voltage	24	24	VDC
Wattage	16.5	16.5	W
Braking torque	4	4	N.m

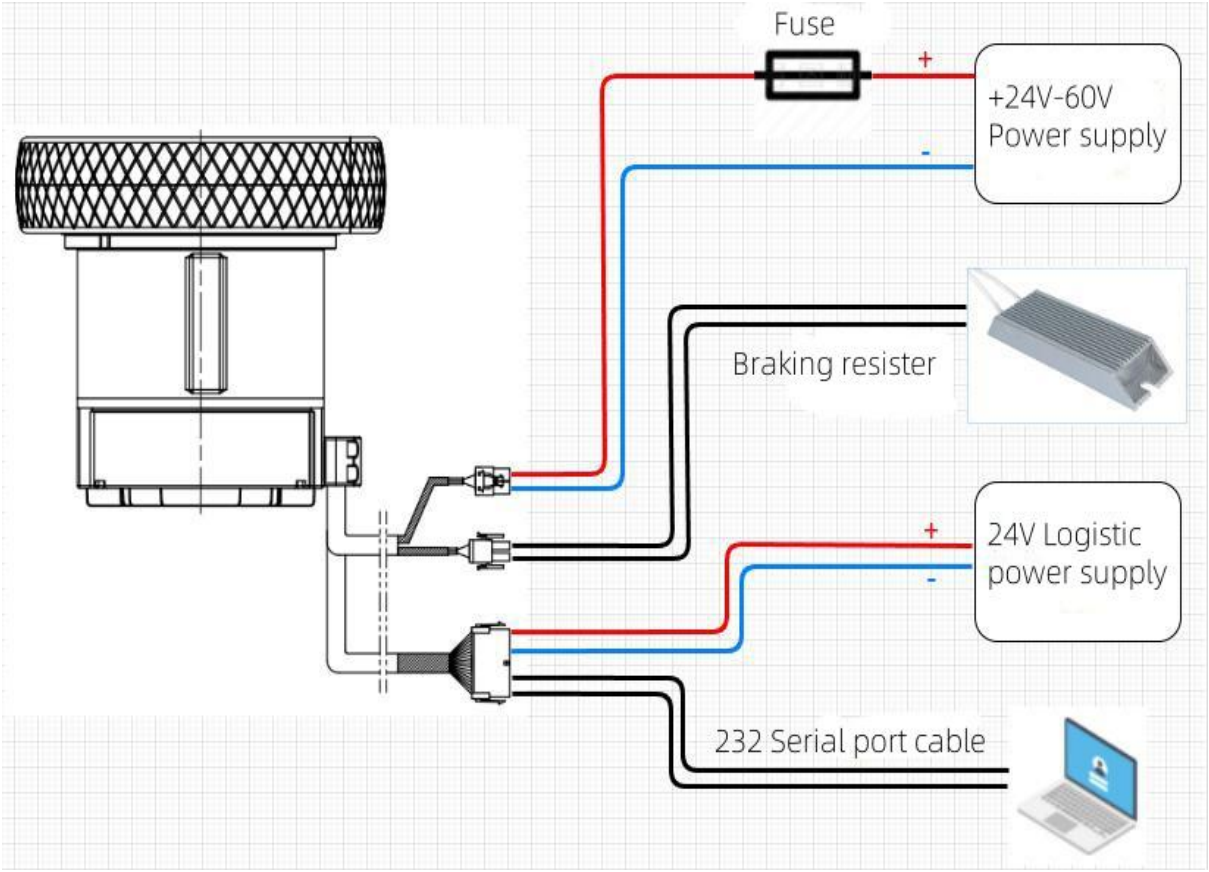
WARNING

Brake type: Power failure brake type

The brake is a holding brake to hold the motor shaft when the motor is stationary and should not be used as a working brake when the motor is rotating.

III Port Description

1.External wiring diagram



2.Braking resistor reference specifications

Model No.: T-10R-100

Resistance Value (Ω): 10

Power (W): 100

Withstanding voltage (VDC): 500

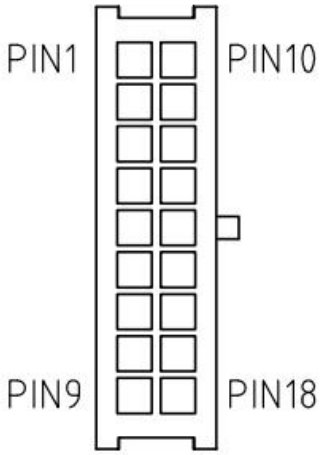
3.Fuse reference specification

Fuse reference specification is 20A/58VDC when the driver power is 500W.

Fuse reference specification is 40A/58VDC when the driver power is 1000W

4.Interface definition

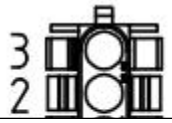
Control port definition (same for both models): Interface Definition



Rear View

No.	Name	Outgoing Wire Color	Pin Function
1	DC24V	Red	Logic power input positive Input voltage: DC24V Maximum input current: 1A
10	GND	Orange	Logic power input negative terminal
2	LOCK+	Light blue	Forced unlock input, only when the AGV body battery is dead and other emergencies, the use of the integrated wheel should be noted that there can not be 24V logic power and 48V power supply access. Input voltage: DC24V Maximum input current: 0.7A
11	LOCK-	Light green	
3	CANH	Green	CAN IN
12	CANL	Yellow	
4	CAN-H	White Brown	CAN OUT
13	CAN-L	White Red	
5	485A	Blue	RS485 Communication
14	485B	Violet	
6	TX	Gray	232 Communication
15	RX	White	
16	GND	Black	
7	OUT	White Yellow	Digital signal output Maximum output current: 20mA
8	I5GND	Pink	GND for CAN and 485 (isolated)
17	DIR	White Green	Direction signal access terminal (valid for 16-GND)
9	PLASH	White-orange	Program writing control terminal (effective for short connection to GND)
18	EN	White-blue	Enable signal access terminal (valid for 16-GND)

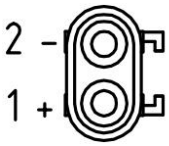
● KY165ACS0405-02B Power Port Definition:



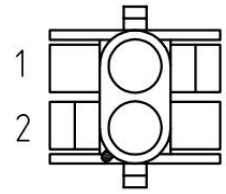
No.	Name	Outlet Color	Pin Function
1	DC+	Red	Positive power input, wire diameter 1.5 mm²

	2	NC		
	3	DC-	Black	Power supply input negative, wire diameter 1.5 mm²

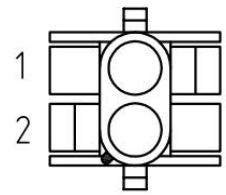
● KY180ACS0410-02B Power Port Definition

	No.	Name	Outlet Color	Pin Function
	1	DC+	Red	Positive power input, wire diameter 2.5mm²
	3	DC-	Black	Power supply input negative, wire diameter 2.5mm²

● KY165ACS0405-02B Braking Resistor Port Definition:

	No.	Name	Outlet Wire Color	Pin Function
	1	RB+	Yellow	Braking resistor input, wire diameter 1.5 mm²
	2	RB-	Blue	

● KY180ACS0410-02B Braking Resistor Port Definition:

	No.	Name	Outgoing Wire Color	Pin Function
	1	RB+	Yellow	Braking resistor input, wire diameter 1.5 mm²
	2	RB-	Blue	

5. Terminal specification

● KY165ACS0405-02B Terminal Specification Table

	Motor terminal	Extension cord terminal
Power supply cable	Rubber housing: TE/AMP 350767-1 Pin: TE/AMP 350218-1	Housings: TE/AMP 350766-1 Pin: TE/AMP 350536-I
Communication Cables	Housings: MOLEX 430201800 Pin: MOLEX 430310004	Housings: MOLEX 430251800 Pin: MOLEX 430300004

Brake Resistor	Housings: TE/AMP 350778-I Pin: TE/AMP 350218-I	Housings: TE/AMP 350777-4 Pin: TE/AMP 350536-I

● KY180ACS0410-02B Terminal Specification Sheet

	Motor terminal	Extension cord terminal
Power Supply Cord	Assembly Model: 97003-M Orange Plastic Case: LCB30-M Assembly Model: 99030-M Copper Assembly: LC30-M	Assembly Model: 97003-F Orange Plastic Case: LCB30-F Assembly Model: 91022-F Copper Assembly: LC30-F
Communication Cable	Rubber Shell: MOLEX 430201800 Pin: MOLEX 430310004	Rubber Shell: MOLEX 430251800 Pin: MOLEX 430300004
Brake Resistor	Housings: TE/AMP 350778-I Pin: TE/AMP 350218-I	Housings: TE/AMP 350777-4 Pin: TE/AMP 350536-I

● KY165ACS0405-02B Cable Specification Sheet

	Wiring Specifications	Cable Length
Power Cord	16AWG	Motor end 0.7m + extension cord 0.7m
Communication cable	28AWG	Motor end 0.7m+extension cable 0.7m
Brake Resistor	16AWG	Motor end 0.7m+extension wire 0.7m

●KY180ACS0410-02B Cable Specification Sheet

	Wiring specification	Cable length
Power cable	13AWG	Motor end 0.7m + extension cord 0.7m
Communication cable	28AWG	Motor end 0.7m+extension cable 0.7m
Brake Resistor	16AWG	Motor end 0.7m+extension wire 0.7m

Ⅳ The Operation Of the Instructions

1.The upper computer software description

1.1. Configuration instructions

（1）Servo controller parameters can be set through the host computer software

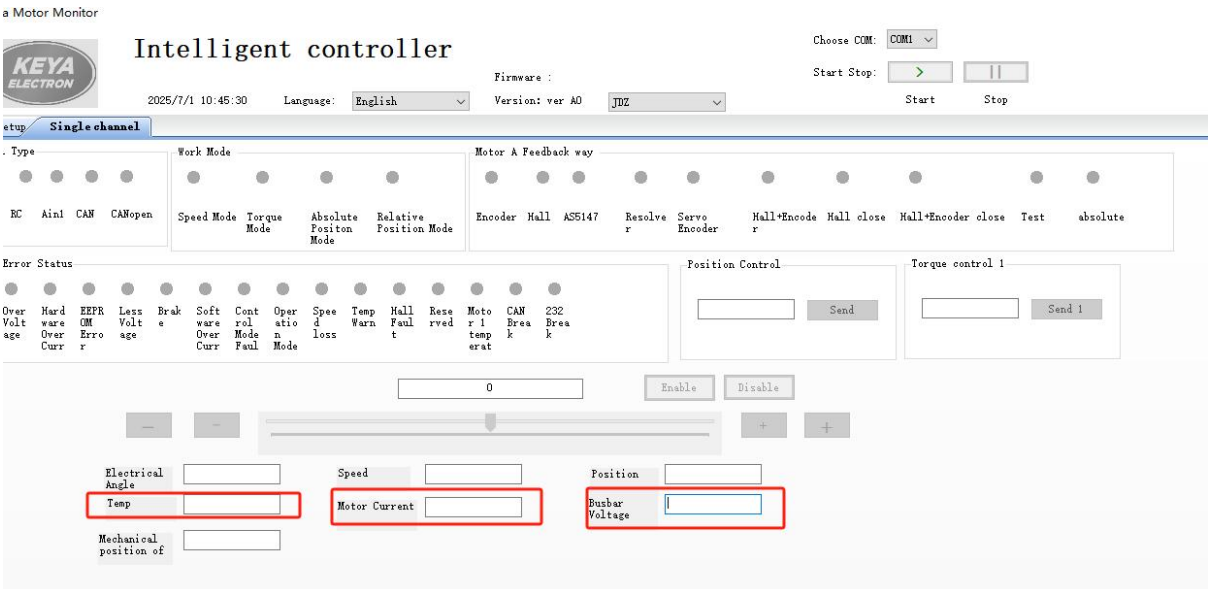
（2）Upper computer software through the RS232 and control control communication, baud rate 115200bit

（3）The upper computer software utilizes .NET environment development, XP system needs to install .NET4.0.

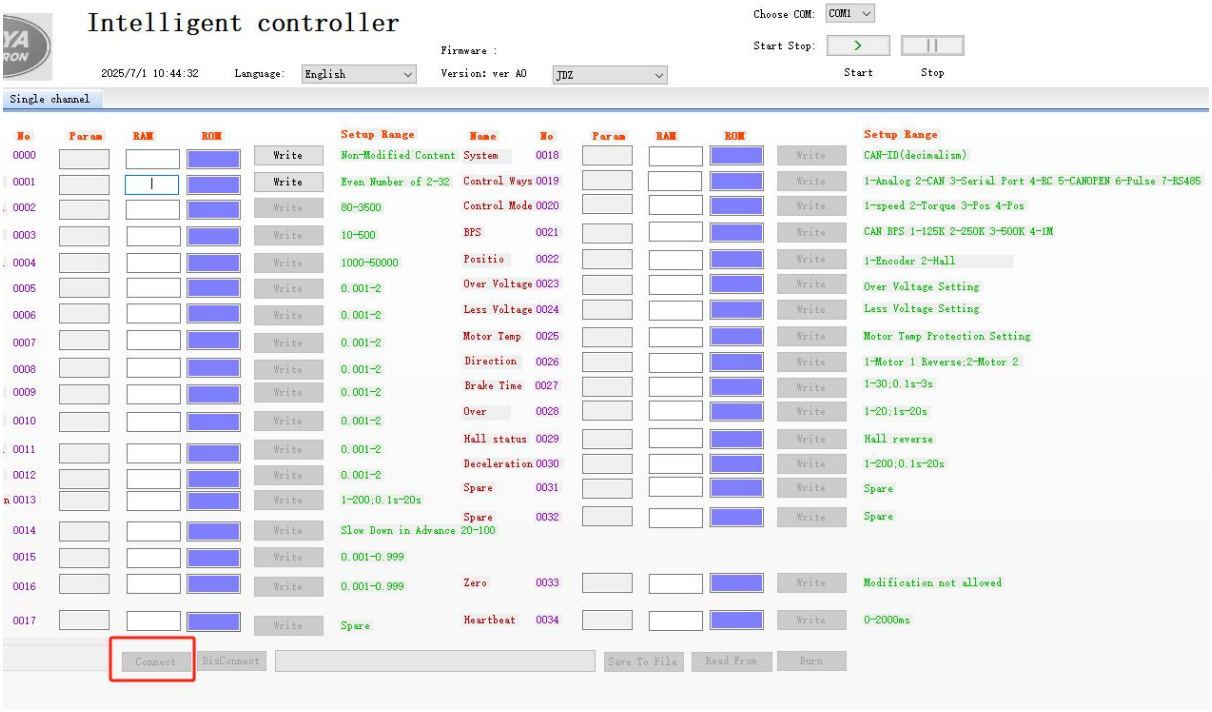
1.2. Software Description

(1) Double-click the icon

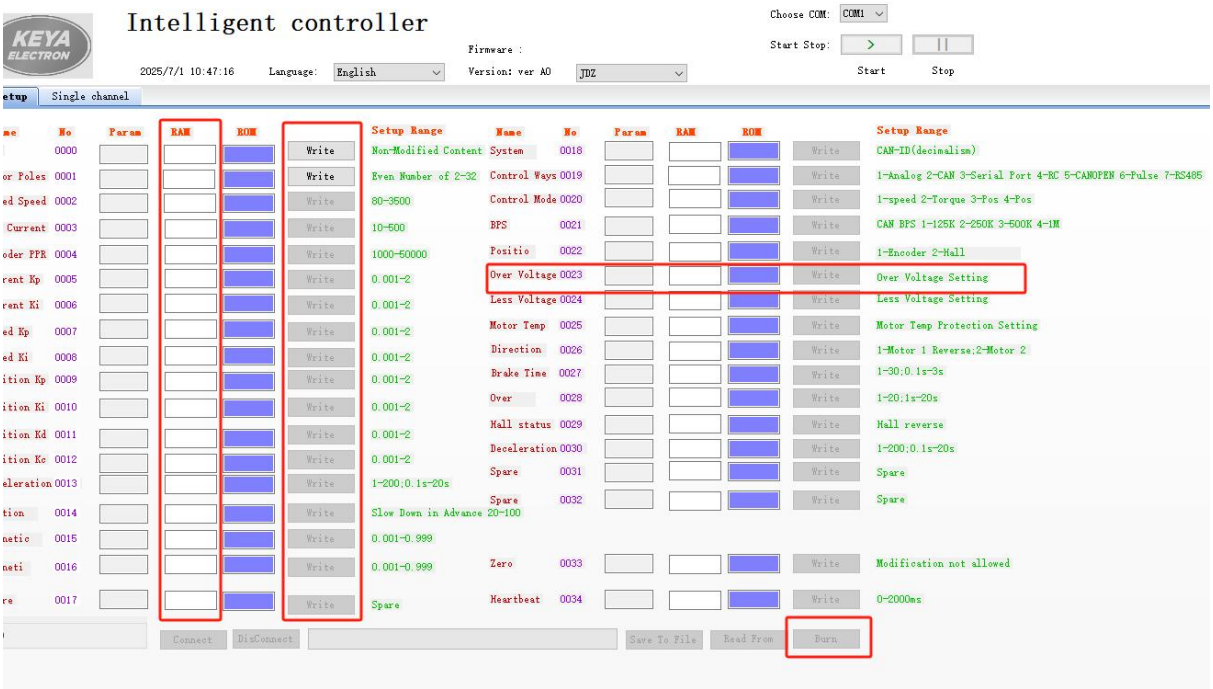
(2) Click the "Start" button, if the communication is normal, the interface will read the feedback parameters, and at the same time, the upper left LOGO light will turn green, indicating that the communication is normal.



(3) Open the configuration interface, click on the lower left corner of the connection button to establish the connection between the software and the controller.



（4） **The RAM** in the red box in the figure is the inputable data, the left side of it is the controller parameters, and the right side is reading the data in the E²ROM, and in the correct case, the three data are consistent (equal). As the software data is constantly scanning, modify the data, quickly modify, and click the corresponding write number button.



(5) For example, if you need to change the overvoltage protection value, and the value stored in E^2ROM is 60, and you need to change it to 58, you should change the parameter of the RAM corresponding to serial number 0023 to 58, and then click the corresponding Write button to write the data into the RAM. Confirm that 58 is no longer changed. The same procedure is followed for other parameters, and multiple parameters can be modified at the same time.

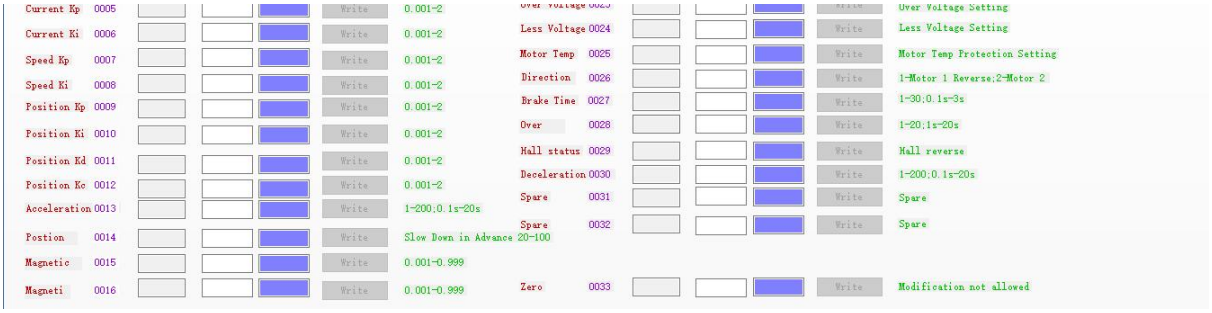
(6) Write the data from RAM to E^2ROM and click the "Write" button at the bottom right. Note: The process of writing is long, about 3 seconds.

(7) the lower "write" button turns red, that is, the data is being written, please wait for the prompt "write successfully", observe the data you need to modify, the three columns of data is consistent, that is, the data of the ROM re-read to the control.

(8) At this point, the end of the control parameter modification, click the "Disconnect" button, click the "Exit" button.

(9) re-power on the controller can be **(Note: read the configuration whether there is no modification, have to power-off reset to normal start)**

(10) When you need to burn and write the configuration of multiple drives, you can "save to file" the modified configuration of one drive, and then "read from file" to download it to another drive.



1.3 Parameter Function Description

0000 Parameter: Identifier to recognize whether the system is connected to the host computer for communication or serial control.

(No need to modify)

0001 parameter: number of motor poles

Parameter 0002: motor rated speed (set according to the motor parameters, the default is 2000)

0003 parameter: maximum motor protection current (default 40A)

0004 parameter: number of encoder lines, set according to the encoder

0005 parameter: Kp parameter of controller current loop PI control

0006 parameter: Ki parameter of controller current loop PI control

0007 parameter: Kp parameter of controller speed ring PI control

Parameter 0008: Ki parameter for controller speed loop PI control

Parameters 0009-0012: Position loop PID control parameters

0013 parameter: acceleration time, "50" means: acceleration time from 0rpm to rated speed is 5 seconds.

0015 parameter: magnetic encoder zero position compensation

Parameter 0016: Rotary transformer zero-point position compensation

0017 parameter: operating mode (dual drive only)

Parameter 0018: Controller system address, or control node number (decimal number)

This parameter is used by slaves in the CAN and CANOpen buses.

Example: ID in CAN bus: 0x0600000 + controller setup address

Parameter 0019: Control mode selection

1 - analog; 2 - CAN control; 3 - RS232 control;

5 - CANOPEN; 7 - RS485 control;

0020 Parameter: control mode selection

1--Speed control

2--Torque control

3--Absolute position control

4--Relative position control

0021 parameter: CAN bus baud rate selection (default CAN:250k in the system)

Baud rate selection: upper 21 parameters - (hexadecimal)

A: 1--125K 2--250k 3--500K

Parameter 0022: Position sensor selection (default incremental encoder)

3--Magnetic encoder

0023 Parameter: Over-voltage protection value setting

0024 Parameter: under-voltage protection value setting

0025 Parameter: motor temperature protection value setting

0026 parameter: motor default direction (speed, torque)

0 - (forward); 1 - (reverse); 256 - (open STO, forward); 257 - (open STO, forward); 258 - (open STO, forward); 259 - (open STO, forward); 259 - (open STO, forward) -(turn on STO,reverse rotation)

0027 Parameter: Braking time of holding brake delay

"10" means: 1 second after receiving the deactivation signal, the motor loses power to hold the brake.

0028 Parameter: Overload delay protection time

" 1" means that the motor is protected immediately after the protection current has been reached for 1 second;

"20" means 20S protection delay after reaching the protection current;

0030 Parameter: deceleration time, "50" means: deceleration time from rated speed to 0rpm is 5 seconds.

Other parameters: standby

2.Indicator description

(1) Status indicator (blue light): Observe the controller status according to the indicator blinking frequency.

Blinking frequency	Definition	Fault cause
0 (always on)	Enable state	No fault
1	Failure	No fault, just enable
2	Over voltage	Supply voltage higher than 68V (settable)
3	Hardware over current protection 110 A	Over current protection caused by instantaneous high motor current, short circuit, field tube damage
4	EEPROM error	Enter Flash configuration program, power off and restart is required after configuration is completed
5	Under voltage	Supply voltage lower than 18V (settable)
6	Reserved	Reserved
7	Software over current protection	Phase current reaches the protection value set by the

	(protection value set by software)	software to stop the output
8	Control mode fault	Control mode selection error
9	Operating mode fault	Operating mode not selected or error
10	Reservation	Not enabled
11	Temperature alarm	Temperature over 85℃ Stop
12	Hall Failure	Motor hall detached or faulty
13	Reservation	Not enabled
14	Motor temperature alarm	Temperature over 120℃ Stop
15	CAN disconnection	CAN mode, no CAN signal input
16	232 disconnected	232 mode, no 232 signal input

(2) Fault indicator (red)

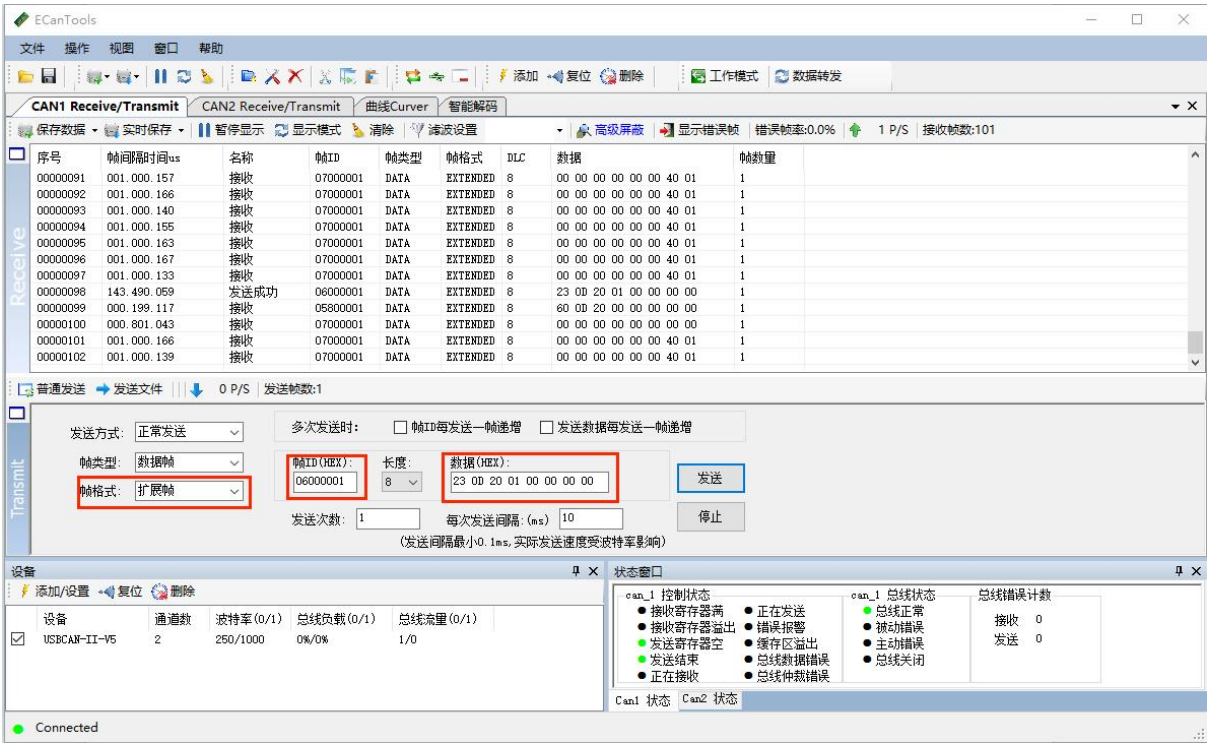
The red indicator light is always on after a fault in any control mode.

Note: The one-piece wheel is a closed structure, the indicator is not exposed, this description is provided for maintenance personnel, non-maintenance personnel ignore the above two!

3.CAN Instruction Description

3.1 General Configuration

- Baud rate: default 250Kbps
- Frame format: extended frame hexadecimal
- Dropout detection period 1000ms (command sending interval shall not exceed 1000ms)
- Data adopts query mode
- There is a fixed heartbeat to send relevant data (frequency is 1Hz)
- Data command is 8 bytes
- Send ID: 0x0600000 + controller address (ID can be configured by upper computer software, factory default is 1)
- Feedback ID: 0x0580000 + controller address
- Heartbeat ID: 0x0700000+controller address
- Query data return are hexadecimal numbers, need to be converted to decimal numbers in sequence



3.2 Description of instruction

Enable = 0x23 0D 20 01 00 00 00 00

Disable=0x23 0C 20 01 00 00 00 00

Speed = 0x23 00 20 01 00 00 00 00

Torque = 0x23 01 20 01 00 00 00 00

Position = 0x23 02 20 01 00 00 00 00

Speed: -10000 ---- +10000 Corresponds to: Negative Rated Speed ----- Rated Speed

Torque: -10000 ---- +10000 Corresponds to: Negative rated torque ----- Rated torque

Position: -2147483648 ----- +2147483647 (10000/turn)

- Enable: 23 0D 20 01 00 00 00 00

Return address: 0x05800000 + controller setting address

Return data: 60 0D 20 00 00 00 00 00

- Disable: 23 0C 20 01 00 00 00 00

Return address: 0x05800000 + controller setting address

Return data: 60 0C 20 00 00 00 00 00 00

- Speed control: 23 00 20 01 DATA_H (h) DATA_H (l) DATA_L (h) DATA_L (l)

  Return address: 0x05800000 + controller setting address

  Return data: 60 00 20 00 00 00 00 00

- Torque control: 23 01 20 01 DATA_H (h) DATA_H (l) DATA_L (h) DATA_L (l)

  Return address: 0x05800000 + controller setting address

  Return data: 60 01 20 00 00 00 00 00

- Position control: 23 02 20 01 DATA_H (h) DATA_H (l) DATA_L (h) DATA_L (l)

  Return address: 0x05800000 + controller setting address

  Return data: 60 02 20 00 00 00 00 00

- Motor current query (A): 40 00 21 01 00 00 00 00

  Return address: 0x05800000 + controller setting address

  Return data: 60 00 21 01 DATA_H DATA_L 00 00

- Fault query: 40 12 21 01 00 00 00 00

  Return address: 0x05800000 + controller setting address

Return data: 60 12 21 01 DATA-H DATA-L 00 00

- Motor speed query (RPM): 40 03 21 01 00 00 00 00

  Return address: 0x05800000 + controller setting address

  Return data: 60 03 21 01 DATA_H DATA_L 00 00

- Rotor mechanical position query (0-9999): 40 04 21 01 00 00 00 00

Return address: 0x05800000 + controller setting address

Return data: 60 04 21 01 00 00 DATA-H DATA-L

- Encoder totalized count value query (four bytes): 40 04 21 02 00 00 00 00

Return address: 0x05800000 + controller setting address

Return data: 60 04 21 02 DATA_H (h)　DATA_H (l)　DATA_L (h)　DATA_L (l)

- Power supply voltage query (V): 40 0D 21 02 00 00 00 00

Return address: 0x05800000 + controller setting address

Return data: 60 0D 21 02 00 00 00 DATA

- Temperature query (℃): 40 0F 21 01 00 00 00 00

Return address: 0x05800000 + controller setting address

Return data: 60 0F 21 01 00 00 DATA 00

- Heartbeat return command:

Return address: 0x07000000 + controller address

Return data:

ELE-H ELE-L speed_H speed_L DATA_H DATA_L Err_H Err_L

Description: Ele-H ELE-L : Motor angle (360° /turn)

Speed_H speed_L : Motor speed (positive or negative)

DATA_H DATA_L : Omitted

Err_H Err_L ： Fault code, corresponding to the number of times the fault indicator flashes.

3.3 CAN bus control example

(1) Speed control:

(speed command value‰) * (set maximum speed) = actual speed.

The upper computer sets the system address to 1 (0018 is set to 1)

The upper computer sets the control mode to CAN control (0019 set to 2)

The upper computer sets the control mode setting to speed control (0020 set to 1)

Control command ID: 0x06000001 (extended ID)

Speed given value -10000 - +10000 represents negative rated speed - positive rated speed

(0xD8F0) (0x2710)

Example: Give speed 1000 rpm (set rated speed 2000 rpm)

Enable command: 23 0D 20 01 00 00 00 00

Speed command: 23 00 20 01 00 00 13 88 (0x1388 = 5000)

Example: For a given speed of -2000 (set rated speed 2000)

Enable: 23 0D 20 01 00 00 00 00 00

Speed command: 23 00 20 01 FF FF D8 F0 (0xD8F0 = -10000)

(2) Torque control:

Limit current value = (given command value ‰) * (set rated current) *80%

The upper computer sets the system address to 1 (0018 is set to 1)

The upper computer sets the control mode to CAN control (0019 set to 2)

Upper computer setting control mode set to speed control (0020 set to 2)

Control command ID: 0x06000001 (extended ID)

Command given value -10000 - +10000 represents negative rated torque - positive rated torque

(0xD8F0) (0x2710)

Example 1: Output 100% torque

Enable: 23 0D 20 01 00 00 00 00

Command: 23 01 20 01 00 00 27 10 (0x2710 = 10000)

When running in this state, the motor is protected after 5 seconds of limiting when the phase current reaches 80% of the rated current value.

Example 2: Output -80% torque

Enable: 23 0D 20 01 00 00 00 00 00

Command: 23 01 20 01 FF FF E0 C0 (0xFFFFE0C0 = -8000)

When running in this state, the motor is protected after 5 seconds of limiting when the phase current reaches 80%*80% of the rated current value.

Note: When sending speed control command and torque control command, it is necessary to send them consecutively, and the time interval must not exceed 1000ms, otherwise it will be judged as CAN dropout error report, and it is necessary to re-send the enable command when starting again;

(3) Position control: (10000/turn)

* Position given value -50000 - +50000 represents clockwise mechanical five revolutions - counterclockwise mechanical five revolutions

(0xFFFF 3CB0) (0xC350)

Control mode set to CAN control by the host computer (0019 set to 2)

Control mode set to absolute position control by the host computer (0020 set to 3)

Or the upper computer sets the control mode to relative position control (0020 set to 4)

The upper computer sets the system address to 1 (0018 set to 1)

Control command ID: 0x0600 0001 (extended ID)

Data sending sequence:

（a）Enable: 23 0D 20 01 00 00 00 00

（b）Running speed: 23 03 20 01 00 00 DATA_H DATA_L (motor speed rpm/min)

（c）Position control: 23 02 20 01 DATA_H (H) DATA_H (L) DATA_L (H) DATA_L (L)

Example 1: Commanding the motor to rotate clockwise 1.8 revolutions at a speed of 500 rpm

（a）Enable: 23 0D 20 01 00 00 00 00

（b）Running speed: 23 03 20 01 00 00 01 F4

（c）Position control: 23 02 20 01 FF FF B9 B0

Example 2: Command the motor to rotate the mechanical angle 72 degrees counterclockwise at 500 rpm (72 * (10000/360) = 2000 = 0x7D0)

（a）Enable: 23 0D 20 01 00 00 00 00

（b）Operating speed: 23 03 20 01 00 00 01 F4

（c）Position control: 23 02 20 01 00 00 07 D0

4.RS232 serial port command description

4.1.General Configuration

The controller serial communication port is set as follows:

- 115200bits/s
- 8 bits of data
- 1 start bit
- 1 stop bit
- No parity
- HEX send/receive
- Dropped line detection period 1000ms (command sending interval should not exceed 1000ms)

4.2 Control format

E0	data0	00	00	00	00	00	00
				H		L	
				Hexadecimal (millionths ratio)			

E0 : Indicates a control instruction

data0 : 01 ------ enable

00------ disable

- Speed mode: -10000 ---- +10000 corresponds to: Negative rated speed ----- Positive rated speed

Enable: E0 01 00 00 00 00 00 00

Send forward speed control 10%: E0 01 00 00 00 00 03 E8

Send reverse speed control 10%: E0 01 00 00 FF FF FC 18

Deactivation: E0 00 00 00 00 00 00 00

- Torque mode: -10000 ---- +10000 corresponds to: negative rated torque ----- positive rated torque

Enable: E0 01 00 00 00 00 00 00

Send torque 80%: E0 01 00 00 00 1F 40 (0x1f40 = 8000)

Send torque -60%: E0 01 00 FF FF E8 90 (0xFFFE890 = -6000)

Deactivation: E0 00 00 00 00 00 00 00

- Position mode: -2147483648 ----- +2147483647 (10000/turn)

Counterclockwise position 5 turns

Enable: E0 01 00 00 00 00 00 00

Position command: E0 01 00 00 00 00 c3 50 (0x0000C350 = 50000)

- Clockwise position 3 turns

Enable: E0 01 00 00 00 00 00 00

Position instruction: E0 01 00 00 FF FF 8A D0 (0xFFFF8AD0 = -30000)

4.3 Query Format

ED	Data1	00	00	00	00	00	00

ED: Indicates a query instruction

data1: 00 ------- control status

01-------- electrical angle (internal parameter)

02-------- Speed (RPM)

03-------- Current (A)

04-------- rotor mechanical position (0-9999)

05--------Voltage (V)

06-------- temperature (℃)

07-------- fault code

08-------- position (10000/turn)

09-------- program version number

- Query current motor speed (rpm)

Upper computer sends: ED 02 00 00 00 00 00 00

Controller feedback: ED 02 05 DC 00 00 00 00 00 00 00 00

That is, the current speed is 1500rpm

- Query the current running current (A)

  Upper computer sends: ED 03 00 00 00 00 00 00 00 00

  Controller feedback: ED 03 00 08 00 00 00 00 00 00 00 00

  That is, the current running motor phase current is 8A

- Query current rotor mechanical position (0-9999)

  Upper computer sends: ED 04 00 00 00 00 00 00

  Controller feedback: ED 04 1B 1F 00 00 00 00 00 00 00 00

  That is, the current rotor mechanical position is 6943 (10000/turn)

- Query controller voltage (V)

  Upper computer sends: ED 05 00 00 00 00 00 00

  Controller feedback: ED 05 30 00 00 00 00 00 00 00 00 00

  The current supply voltage is 48V

- Query controller temperature (℃)

  Host computer sends: ED 06 00 00 00 00 00 00

  Controller feedback: ED 06 1A 00 00 00 00 00 00 00 00 00

  The current controller temperature is 26℃

- Query accumulated position value (10000/turn) (with positive and negative)

  Upper computer sends: ED 08 00 00 00 00 00 00 00 00

  Controller feedback: ED 08 00 01 86 A2 00 00 00 00 00 00

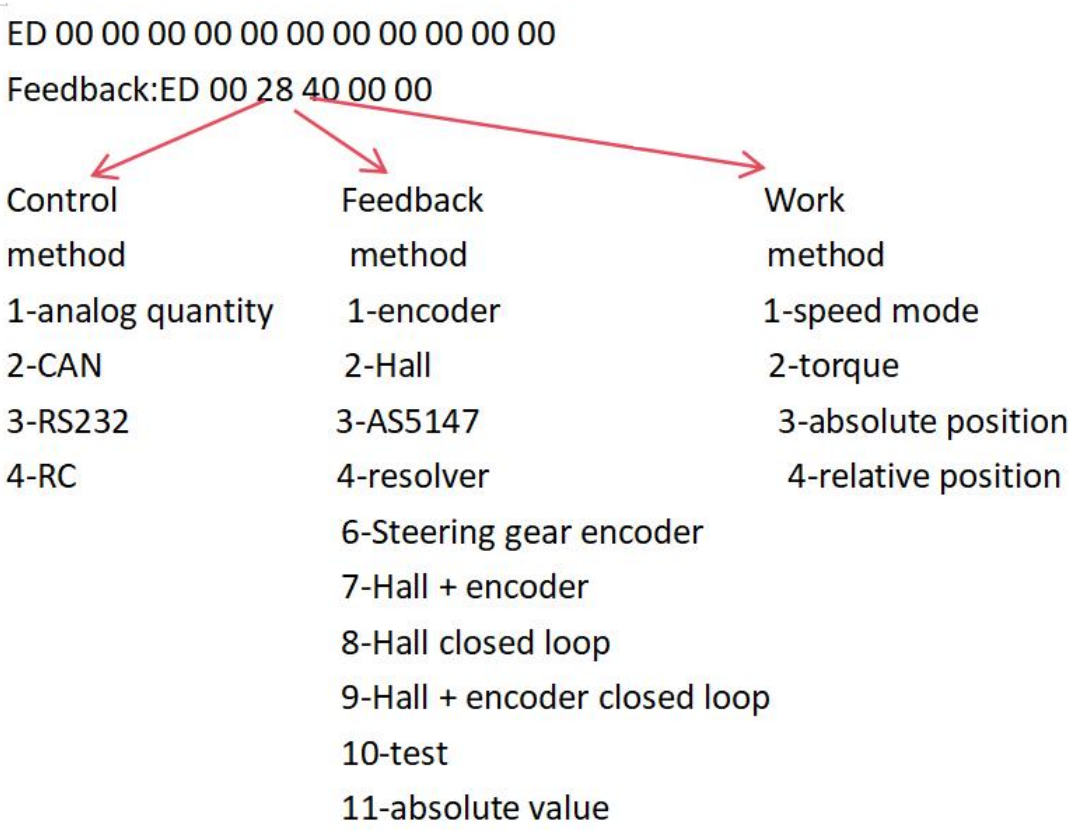
  That is, the current position is 100002

- Query current control status

  Upper computer sends: ED 00 00 00 00 00 00 00

Controller feedback: ED 00 31 20 00 00 00 00 00 00 00 00

That is, the current control status is 3: RS232; 1: Encoder; 2: Torque mode;



- Query fault code

  Upper computer sends:ED 07 00 00 00 00 00 00

  Controller feedback: ED 07 00 10 00 00 00 00 00 00 00 00

  That is, the current fault code is 5 for under voltage.

◆ Fault Code Failure Analysis: hexadecimal first converted to binary, and then from the right to the left number of 1 are in the first position, the **corresponding state indicator flashes the number of times the corresponding faults**

Example: Feedback data is 03 01

0 3 0 1

0000 0011 0000 0001; then corresponds to 1, 9, 10 three faults.

For details, please refer to [Ⅳ2, (1) Indicator Description].

4.4 Serial port heartbeat data

After successful communication, the driver status will be uploaded automatically, 0xEE start bit, the data is defined as follows: (hexadecimal)

Dat0	Dat1	Dat2	Dat3	Dat4	Dat5	Dat6	Dat7	Dat8	Dat9	Dat10	Dat11	Dat12
EE	Electricity Angle		Fault Code		Temperature	Voltage	RPM		Position			

Note:feedback data are hexadecimal, should be converted to decimal reading.

5. RS485 control instructions

5.1 General settings of serial port

1.the default baud rate of 115200bits/s
2.8 bit data
3.1 start bit
4.1 stop bit
5. send data for and parity
6.HEX send and receive
7.Dropped detection period 1000ms (command sending interval shall not exceed 1000ms)
8.Each command, the driver has a return value, return data as follows:

Data_0	Data_1	Data_2	Data_3	Data_4	Data_5	Data_6	Data_7	Data_8	Data_9	Data_10	Data_11	Data_12
Start bit E0+ID	Fault code (H)	Fault code (L)	Bus voltage (V))	Drive temperature (℃)	Speed H (signed)	Speed L (signed)	Current (H) (Symbolized)	Current (L) (Symbolized)	Position 1	Position 2	Position 3	Position 4

5.2 Control format

Data_0	Data_1	Data_2	Data_3	Data_4	Data_5	Data_6	Data_7
Identifier Bit	ID	Function Bit	Function Bit	Function Bit	Function Bit	Data	Check digit

Data0 Identifier: command 0xE0

Date1 Control ID number

Date2 Enable status 01 - enable; 00 - disable

Date3-Date6 First circuit speed value (high bit before, low bit after)

Date7 Checksum (DataO+Data1+..... .Data6=Data7 low)

(1) Speed mode: -10000 ---- +10000 Corresponds to: negative rated speed ----- positive rated speed: controller ID=1

Example: Enable: E0 01 01 00 00 00 00 E2

Example: Send forward speed control 10%: E0 01 01 00 00 03 E8 CD

Send reverse speed control 10%: E0 01 01 FF FF FC 18 F4

Send forward speed control 100%: E0 01 01 00 00 27 10 19

Send reverse speed control 100%: E0 01 01 FF FF D8 F0 A8

Deactivation: E0 01 00 00 00 00 00 E1

(2) Torque mode: -10000 ---- +10000 Corresponds to: Negative rated torque ----- Positive rated torque

Example: Enable: E0 01 01 00 00 00 00 E2

Send torque 80%: E0 01 01 00 00 1F 40 41 (0x1f40 = 8000)

Send torque -60%: E0 01 01 FF FF E8 90 58 (0xFFFE890 = -6000)

Deactivation: E0 01 00 00 00 00 00 E1

(3) Position mode: -2147483648----- +2147483647 (10000/turn)

Counterclockwise position 5 turns

Example: Enable: E0 01 01 00 00 00 00 E2

Position command: E0 01 01 00 00 C3 50 F5 (0x0000C350 = 50000)

Clockwise position 3 turns

Example: Enable: E0 01 01 00 00 00 00 E2

Position instruction: E0 01 01 FF FF 8A D0 3A (0xFFFF8AD0 = -30000)

5.3 Query Format

ED	**Date1**	**Date2**	00	00	00	00	00	00	00	00	00
Starting Bit	ID number	Query command	Fill	Fill	Fill	Fill	Fill	Fill	Fill	Fill	Fill

Date2 00 Indicates control status

01-------- electrical angle

02-------- speed

03-------- current

04-------- rotor mechanical position

05--------voltage

06--------temperature

07-------- fault

08--------encoder count value

09-------- program version number

1. Query control status: current status: 1 - analog 1 - encoder 1 - speed mode

Data sending: ED 01 00 00 00 00 00 00

Drive Feedback: ED 01 00 71 10 00 00 00 00 00 00 00 00

Control mode	Feedback mode	Operating mode
1 - Analog	1 - Encoder	1 - Speed mode
2--CAN	9--Hall+Encoder	2--Torque mode
3--RS232	11--Hall+Encoder	3--Absolute Position Mode
4--RC		3--Relative position mode
5--CANOPEN		
7--RS485		

2.Query motor speed (RPM)
Data sending: ED 01 02 00 00 00 00 00
Drive feedback: ED 01 02 01 F5 00 00 00 00 00 00 00
Motor speed 501rpm (01 F5)

3. Query current (A)
Data transmission:ED 01 03 00 00 00 00 00

Drive feedback: ED 01 03 00 01 00 00 00 00 00 00 00
Motor current 1A

4.Query bus voltage (V)
Data send: ED 01 05 00 00 00 00 00
Drive feedback: ED 01 05 31 00 00 00 00 00 00 00 00
Current bus voltage 49V
5.Query the current temperature (℃)
Data transmission: ED 01 06 00 00 00 00 00
Drive feedback: ED 01 06 1D 00 00 00 00 00 00 00 00
Current current temperature 29℃

6.Query the current fault
Data transmission: ED 01 07 00 00 00 00 00
Drive feedback: ED 01 07 00 11 00 00 00 00 00 00 00
Current drive fault: under voltage

Feedback data in addition to the error code are hexadecimal, should be converted to binary read.

Converted into binary, and then from right to left count 1 are in the first few bits, then the

corresponding state indicator flashes the number of times the corresponding fault.

7, query encoder count value

Data send: ED 01 08 00 00 00 00 00

Drive feedback: ED 01 08 00 00 27 10 00 00 00 00 00

Motor: 00 00 27 10 that is, the current position of 10000 [10000 (encoder resolution * 4 times the frequency) = 1 turn].

8、Query driver program version
Data sending: ED 01 09 00 00 00 00 00
Drive feedback: ED 01 09 01 34 D8 E7 00 00 00 00 00
Current drive program version: 20240615

V Failure Protection and Reset

1. fault protection basis

(1) Temperature alarm
When the temperature of the drive exceeds 85℃ , the temperature alarm is generated, and the alarm flag is automatically cleared when the temperature is restored to 75℃;
(2) Over-current protection
When the phase current reaches 40A for 1 second to stop, disable and speed to 0 reset.
(3) Over-voltage and under-voltage protection
When the power supply voltage is lower than 18V the system will generate under voltage protection;
When the power supply voltage is higher than 68V system will generate over-voltage protection;

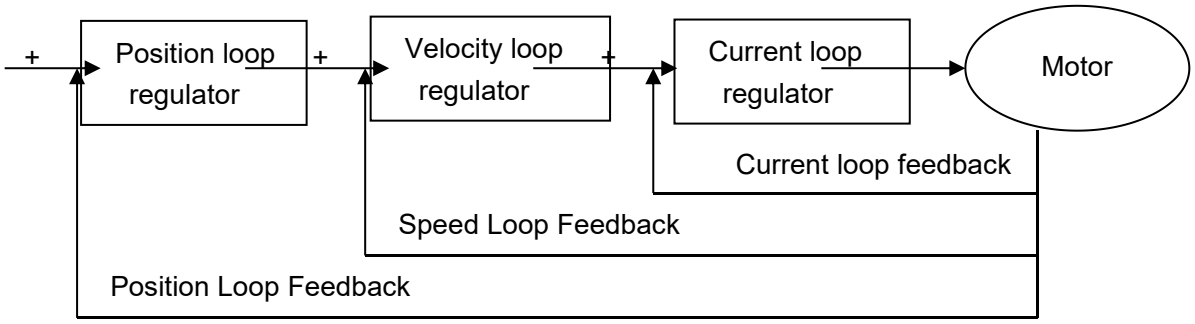
2.Fault information list

Protection category	Safety Level	Shutdown PWM output	FAULT output
Temperature protection	Status latch	yes	yes
Overcurrent protection	Status latch	yes	yes
Undervoltage protection	Status latch	yes	yes
Overvoltage protection	Status latch	yes	yes
EEPROMerror protection	Status latch	yes	yes

Note: After the fault status is locked, the driver will stop the power output; after re-enabling or re-powering up, all the fault flags can be cleared.

VI PID Debugging

In order to make the system get ideal control effect, users need to debug the PID parameters according to their actual applications, so as to improve the dynamic characteristics of the system. (This product is more compatible with the scene, generally do not need to adjust the PID parameters)



In case of multi-loop debugging, the inner loop should be debugged first, and then the outer loop. Examples of parameter adjustment are as follows:

1.Speed ring PID debugging

(1) Through the "host computer software" to set the relevant parameters

(2) Adjust the PID

A. Phenomenon: long starting time, load fluctuation, stopping time.

Adjustment: the parameter is too small, at this time you can increase PI at the same time, D keep 0 unchanged.

B. Phenomenon: fast start, fast adjustment after adding load, fast stop.

Adjustment: better rigidity of the PID, without further adjustment.

C. Phenomenon: motor vibration, speed instability, after the signal is 0, the motor vibration, can not stop.

Adjustment: PID is too large, at this time should be reduced at the same time PI, when the PID is too large, the motor will vibrate.

2.Torque ring PID debugging

(1) Through the "host computer software" to set the relevant parameters

(2) Adjust PID

According to the motor state to judge the PID parameter is too big or too small, and adjust the PID parameter. (Principle as above)

Note: When the rated speed changes, the PID must be readjusted.

VII CAN open Instruction Manual

1.Communication protocol

CANopen is one of the most famous and successful open fieldbus standards, which has been widely recognized and applied in Europe and the U.S.A. In 1992, the CAN in Automation Users' and Manufacturers' Association (CiA, CANinAutomation) was established in Germany, and started to formulate the application layer protocol CANopen for automation. Since then, CiA members have developed a series of CANopen products, which are widely used in the fields of machinery manufacturing, pharmaceuticals, food processing, etc. The DAS series servo is a standard CANopen product.

DAS series servo is a standard CAN slave device, which strictly follows the CANopen2.0A/B protocol, and any host computer that supports this protocol can communicate with it. DAS series servo uses a strictly defined object list, which we call object dictionary, and this object dictionary is designed based on the international standard of CANopen, and all the objects have a clear function definition. All objects have clear functional definitions. Objects are similar to memory addresses. Some objects, such as speed and position, can be modified by an external controller, while others can only be modified by the drive itself, such as status, error messages, etc. These objects are hexadecimal numbers. These objects are hexadecimal numbers, such as the CANopen address of 0x60400010 for the operating mode, for example, as shown in Table 1-1.

Table 1-1 List of Object Dictionary Examples

Composition of the complete CANopen address			Attribute	Meaning
Index	Subindex	Bits (data length)		
0x6040	00	0x10	RW	Device status control word
0x6060	00	0x08	RW	Operating mode
0x6041	00	0x10	MW	Device status word

There are the following attributes of the object:

(1) RW (read-write): the object can be read or written to;

(2) RO (read only): the object can only be read;

(3) WO (write-only): the object can only be written to;

(4) M (mappable): objects can be mapped, similar to indirect addressing;

(5) S (storable): the object can be stored in the Flash-ROM area, and will not be lost if power is lost.

2.Hardware description

CAN communication protocol mainly describes the information transfer between devices, CAN layer definition and open system interconnection model OSI consistent, each layer and another device on the same layer of communication, the actual communication occurs in each device on the two adjacent layers and devices only through the model of the physical layer of the physical interconnection of the physical medium, CAN specification defines the model of the bottom two layers of the data link layer and the physical layer.CAN The physical layer of the bus is not strictly defined, and can use a variety of physical media such as twisted-pair fiber optic cable, etc. The most commonly used is the twisted-pair signal, using differential voltage transmission (commonly used bus transceivers), the two signal lines are known as the CAN_H and CAN_L, static are about 2.5V, the state of the state is represented by the logic of the 1, which can be called a hidden bit, with the CAN_H higher than the CAN_L that is represented by the logic of the 0, called a dominant bit, the voltage is usually at this time, the logic of the 0, which is called a dominant bit. CAN_H is higher than CAN_L to indicate a logic 0, which is called the dominant bit. In this case, the usual voltage values are CAN_H=3.5V and CAN_L=1.5V, and the dominant bit takes precedence in the competition.

Note

1. The CAN_L and CAN_H pins of all slaves are directly connected to each other, and are wired in series, not in star connection;

2.The master and the last slave need to be connected to the 120 ohm termination resistor, the driver is external, not internal.

3.Various baud rates can theoretically communicate the longest distance as shown in Table 1-2.

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

Table 1-2 Theoretical longest communication distance of each baud rate

3.Software description

3.1 EDS description

The EDS (Electronic Data Sheet) file is the identification file or similar code of the slave connected to the PLC, which is used to recognize the type of slave (what kind of similarity among 401, 402, 403, or what kind of device among 402). This file contains all the information of the slave, such as manufacturer, serial number, software version, supported baud rate type, mappable ODs and attributes of each OD, etc. It is similar to the GSD file of Profibus. Therefore, before configuring the hardware, we need to import the EDS file of the slave into the upper configuration software firstly (please contact the salesperson at the time of purchase for the EDS file).

3.2 SDO description

SDO is mainly used to transmit low-priority objects between devices, typically used to configure and manage the slave devices, such as to modify the PID parameters of current loop, speed loop, position loop, PDO configuration parameters, etc. This kind of data transmission is the same as the MODBUS method, i.e., after the master sends out the data, the slave station needs to return the data response. This type of communication is only suitable for parameter setting, not suitable for data transmission with high real-time requirement. The communication method of SDO is divided into uploading and downloading, and the host computer can read and write the OD of the servo according to the special SDO reading and writing commands.

In CANopen protocol, the modification of the contents of the object dictionary can be accomplished by SDO (Service Data Object), and the structure of SDO commands and the guidelines to be followed are described below.

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

Data Sending Format

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

Data Return Format

Identifier	DLC	Data							
		DLC Data	1	2	3	4	5	6	7
0x580+Node_ID	8	Receive Command Word	Object Index		Object subindex				

Note:

The command words are all 0x40 when the SDO message reads node parameters;

If the received data is 1 byte, the receive command word is 0x4F; if the received data is 2 bytes, the receive command word is 0x4B; if the received data is 4 bytes, the receive command word is 0x43;

If there is an error in the receive data, the receive command word is 0x80.

Read node commands Data_0	Data_1,2	Data_3	Data_4	Data_5	Data_6	Data_7
0x40	Index	Sub index	00	00	00	00

SDO message when modifying parameters

Identifier	DLC	Data							
		0	1	2	3	4	5	6	7
0x600+Node_ID	8	Send Command Word	Object Index		Object sub index	Maximum 4 bytes			

Receive SDO when changing parameters:

Identifier	DLC	Data							
		0	1	2	3	4	5	6	7
0x580+Node_ID	8	Receive Command Word	Object Index		Object Sub-Index	0			

Note:

Write Node Parameters If the data to be sent is 1 byte, the send command word is 0x2F; if the data to be sent is 2 bytes, the send command word is 0x2B; if the data to be sent is 4 bytes, the send command word is 0x23; if the SDO message is sent successfully, the receive command word is 0x60; if the SDO message is sent unsuccessfully, the receive command word is 0x80.

Multi-byte indexes, data are preceded by the lower byte

Write node command Data_0	Data_1, 2	Data_3	Data_4	Data_5	Data_6	Data_7
0x2F	Index	Sub-Index	XX	XX	00	00
0x2B	Index	Sub-Index	XX	XX	00	00
0x27	Index	Sub-Index	XX	XX	XX	00
0x23	Index	Sub-Index	XX	XX	XX	XX

3.3 PDO description

PDO can transmit 8 bytes of data at one time, there is no other protocol preset (meaning that the data content has been predefined), mainly used to transmit the data that needs high frequency exchange.PDO transmission method breaks the existing data question and answer transmission concept, and adopts a brand new mode of data exchange, the two sides of the equipment in the transmission of the first in each device to define a good data receiving and sending area, and directly send the relevant data to the other side's data receiving area can be sent in the data exchange. The two devices define the data receiving and transmitting area in each device before transmission, and then directly send the relevant data to each other's data receiving area during data exchange, which reduces the question-and-answer inquiry time, thus greatly improves the efficiency of bus communication, and thus obtains a very high utilization rate of the bus.

3.3.1 PDO COB-ID description

COB-ID is a unique method of CANopen communication protocol, which is called Communication Object Identifier. These COB-IDs define the corresponding transmission levels for PDO, with which the controller and the servo can define the same transmission levels and the contents of the transmission in their respective software configurations, so that the controller and the servo are able to define the same transmission levels and the contents of the transmission. With these transmission levels, the controller and the servo can define the same transmission level and contents in their respective software configurations, so that if the controller and the servo use the same transmission level and contents, the transmission of data becomes transparent, i.e., both parties know the contents of the data to be transmitted, and there is no need to ask the other party to review whether the data transmission was successful or not when transmitting the data.

The default ID assignment table is based on the 11-bit CAN-ID defined in CANopen 2.0A (CANopen 2.0B protocol COB-ID is 29 bits), and consists of a 4-bit function code part and a 7-bit Node-ID part.

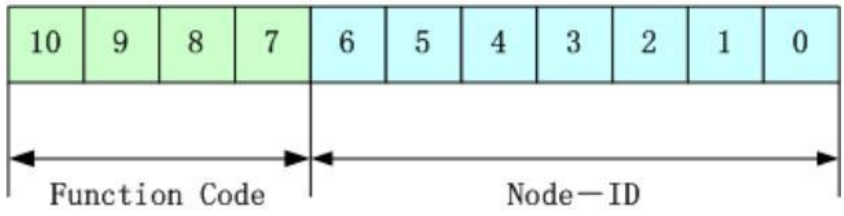


Figure 11-2 Default ID Illustration

Note

Node-ID - the station number of the servo, the Node-ID range is 1～ 127;

Function Code - the function code for data transmission, defines the transmission level of various PDO, SDO and management messages, the smaller the function code, the higher the priority.

Table 11-4 CANopen Predefined Master/Slave Connection Sets CAN Identifier Assignment Table

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

3.3.2 COB-ID

for Transmit PDO refers to the data sent out by the servo and received by the PLC. The function code sending PDO (COB-ID) is (COB-ID) is:

1.0x180+servo station number

2.0x280+servo station number

3.0x380+servo station number

4.0x480+servo station number

Receive PDO relative to the servo is the data received by the servo, these data are sent by the PLC, send PDO function code (COB-ID).

（The function code of sending PDO (COB-ID) is:

1.0x200+servo station number

2.0x300+servo station number

3.0x400+servo station number

4.0x500+servo station number

3.3.3 PDO transmission types

There are two types of PDO transmission:

Synchronous (SYNC) - transmission triggered by a synchronization telegram (transmission type: 0-240)

In this transmission mode, the controller must have the ability to send synchronization telegrams (telegrams sent periodically with a frequency of up to 1KHZ), which the servo sends after receiving the synchronization telegram.

Off-cycle - transmission is pre-triggered by a remote frame, or by an object-specific event specified in the device sub-protocol. In this mode the servo drive sends the data in the PDO every time it receives a synchronization message.

Cyclic - Transmission is triggered after every 1 to 240 SYNC messages. In this mode, the data in the PDO is sent once for every n synchronization messages received by the servo drive.

Asynchronous (transmission type: 254/255)

Slave messages are sent as soon as the data is changed, regardless of whether the master asks for it or not, and the time interval between two sends of the same message can be defined to avoid high-priority messages from occupying the bus all the time (the lower the value of PDO, the higher the priority).

For DAS series servo drives, it supports asynchronous transmission

Note

A PDO can specify an inhibit time, i.e., define the minimum interval between two consecutive PDO transmissions, to avoid the problem that the data volume of high-priority messages is

3.3.4 Protection mode/supervision type description

Supervision type refers to what kind of checking method the master chooses to check the slave during operation, through which it determines whether the slave is faulty or not, and makes corresponding treatment according to these faults!

1、Heartbeat message

The slave station periodically sends telegrams to the master station according to the "heartbeat telegram generation time", if the master station has not received the next heartbeat telegram from the slave station after a certain period of time (set in the master station), then the master station determines that the slave station is in error!

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

Status - 0: start, 4: stop, 5: run, 127: pre-operation

2. Node protection

The master sends telegrams to the slave periodically with "supervision time", if the slave has not received the node telegram from the master after the time of "supervision time*life factor", then the slave alarms!

Master request message format - (0x700 + node number) (no
data in this message) Slave response message format -
(0x700 + node number)+ Status:

The Status - Data section consists of a trigger bit (bit 7) that must be alternately set to "0" or "1" for each node protection response. The trigger bit is set to "0" at the first node protection request. Bits 0 to bit 6 (bit0～ 6) indicate the node status; 0: initialization, 1: not connected, 2: connected, 3: operation, 4: stop, 5: running, 127: pre-operation.

Standard CAN slaves generally support only one type of node protection, and DAS series servo drives use heartbeat message detection.

3.3.5 Description of Startup Process

During network initialization, CANopen supports extended boot-up as well as minimized boot-up process. This initialization process can be represented by a node state transition diagram, as shown in Figure 11-3

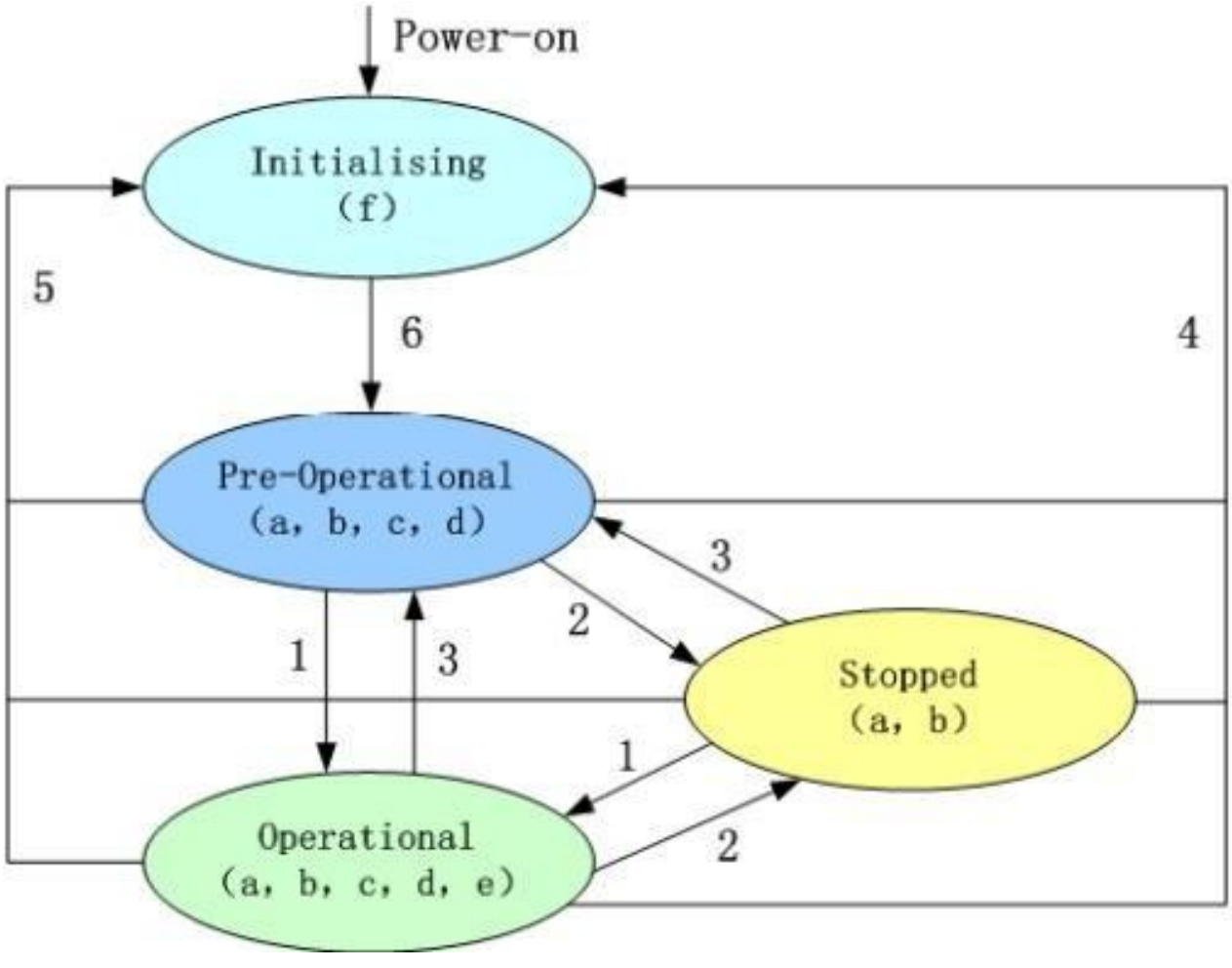


Figure 11-3 Node State Transition Diagram

Table 11-5 CANopen Network Status

Code	Meaning
a	NMT
b	Node Guard
c	SDO

d	Emergency
e	PDO
F	Boot-up

Management Message Format

COB-ID	DLC	Byte0	Byte1
0x000	02	CS	Station Number

All NMT slave devices are addressed when Node-ID=0. CS is a command word and its value is shown in Table 1-11.

Table 1-11 CS Value Table

Command word	NMT Service
0x01	Turn on node, start PDO transmission
0x02	Turn off node, turn off PDO transmission
0x80	Enter pre-operational state
0x81	Reset node
0x82	Reset communication

4. Object Dictionary Description

The CANopen dictionaries shown in this section are subject to change. The CANopen EDS file can be contacted to business personnel.

The object dictionary given in the following table is the EDS file mapping that comes standard with the DAS series dual drives, and also contains the runtime query and runtime related command introduction.

Note:

The motor B parameter is meaningless when using a single driver!

COB-ID	Index	Sub-index	Parameter Name	Access Attributes	Data type	Variable Name	Remarks
	0x1000	0	Device type	RO	Uint32	Func.obj1000	
	0x1001	0	Error Register	RO	Uint8	Func.obj1001	
	0x1003	0	Predefined Error Region	RW	Uint8	Func.highestSubIndex_obj1003	
		1		RO	Uint32	Func.obj1003[0]	
		2		RO	Uint32	Func.obj1003[1]	
		3		RO	Uint32	Func.obj1003[2]	
		4		RO	Uint32	Func.obj1003[3]	
	0x1005	0	Synchronization frame ID	RW	Uint32	SYNC COB-ID	
	0x1009	0	Hardware Version	RO	Uint16	Func.obj1009	
	0x100A	0	Firmware Version	RO	Uint16	Func.obj100A	
	0x1014	0	Emergency COB-ID	RW	Uint32	Emergency COB ID	
	0x1017	0	Heartbeat Time	RWS	Uint32	Func.obj1014	
	0x1200	0	Slave SDO Parameters	RO	Uint8	Func.highestSubIndex_obj1200	
0x600 +Node ID			Slave Receive ID	RO	Uint64	COB_ID_Client_to_Server_Receive_SDO	

0x580+_Node ID			Slave Transmit ID	RO	Uint64	COB_ID_Server_to_Client_Transmit_SDO	

4.1 RPDO parameter table

COB-ID	Index	Sub-index	Name	Access Attributes	Data Type	Remarks
0x600+id	0x1400	0	RPDO1 Parameter	RO	Uint8	Number of entries
		1	COB-ID	RW	Uint32	0x200 + id
		2	Transmission Type	RWS	Uint8	255
		3	Disable Time	RWS	Uint16	0
		4	Compatibility (Reserved)	RW	Uint8	
		5	Processing Time	RWS	Uint16	1000
	0x1401		RPDO2 Parameter	RO	Uint8	Number of entries
		1	COB-ID	RW	Uint32	0x300 + id
		2	Transmission Type	RWS	Uint8	255
		3	Suppression Time	RWS	Uint16	0
		4	Compatibility	RW	Uint8	
		5	Processing Time	RWS	Uint16	1000
	0x1402	0	RPDO3 Parameters	RO	Uint8	Number of entries
		1	COB-ID	RW	Uint32	0x400 + id
		2	Transmission Type	RWS	Uint8	255

		3	Disable Time	RWS	Uint16	0
		4	Compatibility (Reserved)	RW	Uint8	
		5	Processing Time	RWS	Uint16	1000
	0x1403		RPDO4 Parameters	RO	Uint8	Number of entries
		1	COB-ID	RW	Uint32	0x500 + id
		2	Transmission Type	RWS	Uint8	255
		3	Disable Time	RWS	Uint16	0
		4	Compatibility (Reserved)	RW	Uint8	
		5	Processing Time	RWS	Uint16	1000

4.2 RPDO Mapping

COB-ID	Index	Sub-Index	Name	Access Attributes	Data Type	Remarks
0x200 + id	0x1600	0x1600	RPDO1 Number of mapped objects	RW	Uint8	
		1	Maps the first object	RWS	Uint32	607A (A enable)
		2	Maps the 2nd object	RWS	Uint32	6081 (A Data)
		3	Maps the 3rd object	RWS	Uint32	
		4	Maps the 4th object	RWS	Uint32	
0x300 + id	0x1601	0	RPDO2 Number of mapped objects	RW	Uint8	
		1	Maps the first object	RWS	Uint32	6083 (B enabled)
		2	Maps the 2nd object	RWS	Uint32	6084 (B Data)
		3	Maps the 3rd object	RWS	Uint32	
		4	Maps the 4th object	RWS	Uint32	
0x400 + id	0x1602	0	RPDO3 Mapping	RW	Uint8	
		1	Mapping 1st object	RWS	Uint32	
		2	Mapping the 2nd object	RWS	Uint32	
		3	Maps the 3rd object	RWS	Uint32	
		4	Maps the 4th object	RWS	Uint32	

0x500 + id	0x1603	0	RPDO4 Number of mapped objects	RW	Uint8	
		1	Maps the first object	RWS	Uint32	
		2	Maps the 2nd object	RWS	Uint32	
		3	Maps the 3rd object	RWS	Uint32	
		4	Maps the 4th object	RWS	Uint32	

4.3 TRDO parameter table

COB-ID	Index	Sub-Index	Name	Data/Read/Write	Data Type	Remarks
0x180+id	0x1800	0	TPDO1 Parameter	RO	Uint8	Number of entries
		1	COB-ID	RW	Uint32	180 + Nodes
		2	Transmission Type	RWS	Uint8	255
		3	Disable Time	RWS	Uint16	0
		4	Compatibility (Reserved)	RW	Uint8	
		5	Processing Time	RWS	Uint16	1000
0x280+id	0x1801		TPDO2 Parameters	RO	Uint8	Number of entries
		1	COB-ID	RW	Uint32	280 + Nodes
		2	Transmission Type	RWS	Uint8	255
		3	Disable Time	RWS	Uint16	0
		4	Compatibility (Reserved)	RW	Uint8	
		5	Processing Time	RWS	Uint16	1011
0x380+id	0x1802	0	TPDO3 Parameters	RO	Uint8	Number of entries
		1	COB-ID	RW	Uint32	380 + Nodes

		2	Transmission Type	RWS	Uint8	255
		3	Disable Time	RWS	Uint16	0
		4	Compatibility (Reserved)	RW	Uint8	
		5	Processing Time	RWS	Uint16	1022
0x480+id	0x1803		TPDO4 Parameters	RO	Uint8	Number of entries
		1	COB-ID	RW	Uint32	
		2	Transmission Type	RWS	Uint8	
		3	Disable Time	RWS	Uint16	
		4	Compatibility (Reserved)	RW	Uint8	
		5	Processing Time	RWS	Uint16	

4.4 TPDO Mapping

COB-ID	Index	Sub-Index	Name	Data/Read/Write	Data Type	Remarks
	0x1A00	0	TPDO1 Mapping	RW	Uint8	
		1	Mapping 1st object	RWS	Int16	2012 (A Speed)
		2	Mapping 2nd object	RWS	Int16	2010 (A current)
		3	Mapping 3rd object	RWS	Int32	2016 (A position)
		4	Mapping 4th object	RWS		
	0x1A01	0	TPDO2 Mapping	RW	Uint8	
		1	Mapping 1st object	RWS	Int32	2022 (B Speed)
		2	Maps the 2nd object	RWS	Int8	2020 (B current)
		3	Mapping 3rd object	RWS	Int32	2026 (B position)
		4	Mapping 4th object	RWS	Uint32	
	0x1A02	0	TPDO3 Mapping	RW	Uint8	

		Uint8	Mapping 1st object	RWS	Uint16	2011 (A Fault)
		2	Mapping 2nd object	RWS	Uint16	2021 (B fault)
		3	Mapping 3rd object	RWS	Uint16	2013 (Voltage)
		4	Mapping 4th object	RWS	Uint16	203C (temperature)
	0x1A03	0	TPDO4 Mapping	RW	Uint8	
		1	Mapping 1st object	RWS	Uint32	2003 (control state)
		2	Mapping 2nd object	RWS	Uint32	2004 (control state)
		3	Mapping 3rd object	RWS	Uint32	
		4	Mapping 4th object	RWS	Uint32	

4.5 Function Code

	Index	Subindex	Name	Access Attributes	Data type	Variable Name	Remarks
Monitoring Parameter							
	0x2000	0	Slave node number	RO	Uint16	Func.Monitor.Slavenodes	
Motor A							
	0x2003	0	Motor A-control state	RO	Uint16	InStateGroup1	
	0x2010	0	Motor A-current RMS value	RO	Uint16	MotorA_Current	A
	0x2011	0	MotorA-fault status bit	RO	Uint16	MotorA_Close	
	0x2012	0	Motor A-RPM	RO	int16	MotorA_Speed	RPM
	0x2013	0	Motor A DC bus voltage	RO	Uint16	MotorA_Voltage	V
omitted	0x2014	0	MotorA_Temperature	RO	Uint16	MotorA_Temp	℃
	0x2015	0	MotorA - Mechanical Angle	RO	Uint16	MotorA_QepRewTeta	0-9999
	0x2016	0	MotorA - position feedback	RO	int32	MotorA_Position	RPM/10000
	0x2017	0	MotorA - Electrical Angle	RO	Uint16	MotorA_ElecTheta	Internal parameters
Motor B							

	0x2004	0	Motor B-control state	RO	Uint16	OutStateGroup1	
	0x2020	0	Motor B - Current RMS	RO	Uint16	MotorB_Current	Unit A
	0x2021	0	Motor B - Fault status bit	RO	Uint16	MotorB_Close	Bit
	0x2022	0	MotorB - speed	RO	int16	MotorB_Speed	RPM
	0x2023	0	B drive DC bus voltage	RO	Uint16	MotorB_Voltage	V
omitted	0x2024	0	Motor B - Temperature	RO	Uint16	MotorB_Temp	℃
	0x2025	0	Motor B - Mechanical Angle	RO	Uint16	MotorB_QepRewTeta	0-9999
	0x2026	0	MotorB-position feedback	RO	int32	MotorB_Position	RPM/10000
	0x2027	0	MotorB-electrical angle	RO	Uint16	MotorB_ElecTheta	Internal Parameters

4.6 System Code

Control Data	Index	Sub-index	Name	Access Attributes	Data Type	Variable Name	Remarks
	0x607A		Motor A enable Hold control mode	RW	Uint32	Enable motor-A	0x030D2001 (enable) 0x030C2001 (disable)
	0x607A		Motor-A speed mode enable	RW	RW	Enable motor-A	0x030D2011 (enable) 0x030C2001 (disable)
	0x607A		Motor-A absolute position mode enable	RW	RW	Enable motor-A	0x030D2031 (enable) 0x030C2001 (disable)
	0x607A		Motor-A relative position mode enable	RW	RW	Enable motor-A	0x030D2041 (enable) 0x030C2001 (disable)
	0x607A		Motor-A fault reset	RW	RW	Enable motor-A	0x030C20F1
	0x607A		Motor-A forced zero position	Uint32 Enable motor-A	RW	Enable motor-A	0x030D0000
	0x6081		Motor-A action data	RW	RW	The data of motor-A	-10000~10000
	0x6083		Motor B Enable Hold control mode	RW	RW	Enable motor-B	0x030D2002 (enable) 0x030C2002 (disable)
	0x6083		Motor B speed mode enable	RW	RW	Enable motor-B	0x030D2012 (enable) 0x030C2002 (disable)
	0x6083		Motor B absolute position mode enable	RW	RW	Enable motor-B	0x030D2032 (enable) 0x030C2002 (disable)
	0x6083		Enable motor-B relative position mode	RW	RW	Enable motor-B	0x030D2042 (enable) 0x030C2002 (disable)
	0x6083		Motor B fault reset	RW	RW	Enable motor-B	0x030C20F2
	0x6083		Motor-B forced position zero	Uint32 Enable motor-B	RW	Enable motor-B	0x030D0000
	0x6084		Motor B action data	Uint32	RW	The data of motor-B	-10000~10000

	Index	Sub-Index	name	Access Attributes	Data Type	Variable Name	Remarks
Control Data							
	0x6099	01	Motor A speed limit (Position mode)	RW	Uint32	Homing speed	0~10000
		02	Motor B speed limit (Position mode)	RW	Uint32	Homing search speed	0~10000
System parameter							
	0x203A		Control status	RO	Uint16	MotorState	
	0x203B		System Failure Status	RO	Uint32	MotorClose	Two way fault merge
	0x203C		Drive Temperature	RO	Uint16	Driver_Temp	
	0x203D		Driver Software Version Number	RO	int32	MotorVersion	

5.SDO usage example

A Enable 0x030D2001 51191809 (decimal)
A Disable 0x030C2001 51126273 (decimal)
B Enable 0x030D2002 51191810 (decimal)
B Disable 0x030C2002 51126274 (decimal)

SDO Test Drive ID : 2

ID2 device CANopen startup : ID 0000 01 02
Bus all devices CANopen startup : ID 0000 01 00
Motor A enable:
Send: 0x602 23 7A 60 00 01 20 0D 03 (0x030D2001, low before, high after)

Feedback: 0x582 60 7A 60 00 00 00 00 00

Motor A deactivated:

Send: 0x602 23 7A 60 00 01 20 0C 03 (0x030C2001, low in front, high in back)

Feedback: 0x582 60 7A 60 00 00 00 00 00

Motor A speed 50% :

Send: 0x602 23 81 60 00 88 13 00 00 (5000=0x1388, low in front, high in back)

Feedback: 0x582 60 81 60 60 00 00 00 00 00

Motor A forced position zero :

Send: 0x602 23 7A 60 00 00 00 0D 03 (0x030D0000, low before, high after)

Feedback: 0x582 60 7A 60 00 00 00 00 00

Motor operation status related index:

Drive Temp: 0x203C Drive Bus Voltage: 0x2013 (low before, high after)

(A circuit: speed 0x2012 phase current 0x2010 encoder count (position) 0x2016 fault 0x2011)

(B circuit: speed 0x2022 phase current 0x2020 encoder count (position) 0x2026 fault 0x2021)

Read motor speed:

Transmit: 0x602 40 12 20 00 00 00 00 00

Feedback: 0x582 4B 12 20 00 96 00 00 00 (0x96 motor speed 150RPM)

Read controller temperature:

Send: 0x602 40 3c 20 00 00 00 00 00

Feedback: 0x582 4B 3C 20 00 21 00 00 00 (0x21 controller temperature 33℃)

Read controller bus voltage:

Send: 0x602 40 13 20 00 00 00 00 00

Feedback: 0x582 4B 13 20 00 30 00 00 00 (0x30 controller input voltage 48V)

Read encoder count value (position):

Send: 0x602 40 16 20 00 00 00 00 00

Feedback: 0x582 43 16 20 00 F1 D2 48 00 (0x0048D2F1 encoder count value 4772593)

Ans: 0x0048D2F1 Encoder count value 4772593 (Int32, count from zero again after out of range)

Note:The return value is in hexadecimal, after conversion to decimal, it is the pulse count value of the actual encoder after 4 times frequency.

(10000 (encoder resolution * 4x frequency) = 1 lap)

Fault query:

Send: 0x602 40 11 20 00 00 00 00 00

Feedback: 0x582 4B 11 20 00 01 08 00 00

Note; The data fed back are hexadecimal except for the error code, which should be converted to binary for reading.

Error Code Failure Analysis: See [IV.3 Failure Analysis] for details.

Example: The feedback data is: 4B 11 20 00 01 08 00 00

A road fault: 08 01 Convert to binary:100000000001

A road then the fault is: 1 12 (deactivation, Hall fault)

Note: When using the CAN transceiver single frame data transmission, after sending the enable command must be sent within 1000ms speed command, otherwise the driver judgment CAN communication drop automatically protect, after protection need to send the enable command again to start.

The following figure shows the test data format using CAN transceiver.

6.PDO use example

PDO test Driver ID: 1

CANopen start: ID 0000 01 00

CANopen shutdown: ID 0000 02 00

RPDO1: Motor control instruction

- Motor A enable:
  ID 0x201 01 20 0D 03 00 00 00 00 (0x030D2001, low before, high after)

- Motor A disable:
  ID 0x201 01 20 0C 03 00 00 00 00 (0x030C2001, low before, high after)

- Motor A action data 10%:
  ID 0x201 00 00 00 00 E8 03 00 00 (1000=0x03E8)

Speed value: 0x03E8=1000;

Target speed: 10% of the drive's set rated speed, in parts per million ratio (1000/10000)

- Motor A enable + speed 10%:

  ID 0x201 01 20 0D 03 E8 03 00 00 (0x030D2001 enable + 0x03E8 speed 10%)

- Motor A forced position zeroing:

  ID 0x201 00 00 0D 03 00 00 00 00 (0x030D0000)

1. Set the mapping of TPDO and RPDO with SDO.

Set before CANopen is turned on, ID is 600+ID

1. T/RPDO mapping needs DLC after sub index (0x0010 two bytes or 0x0020 four bytes);

2.T/RPDO No need;

3.All of them are preceded by the low bit and followed by the high bit;

Example:

- Set the 1st object mapped by TPDO1 as 2012 (A speed)

  Send: 0x601 23 00 1A 01 10 00 12 20

  Feedback: 0x581 60 00 1A 01 00 00 00 00

  Index 0x1A00, subindex 0x01.

  Mapping object:RPM 0x2012, 0x0010, data length 16 bits;

- Set the 2nd object mapped by TPDO1 to 2010 (A current)

  Send: 0x601 23 00 1A 02 10 00 10 20

  Feedback: 0x581 60 00 1A 02 00 00 00 00

  Index 0x1A00, subindex 0x02.

  Mapping object:current 0x2010, 0x0010, data length 16 bits;

- Set the 3rd object mapped by TPDO1 to 2016 (position A)

  Send: 0x601 23 00 1A 03 20 00 16 20

  Feedback: 0x581 60 00 1A 03 00 00 00 00

  Index 0x1A00, sub-index 0x03,

  Mapping object:position 0x2016, 0x0020, data length 32 bits;

- Set the 3rd object mapped by TPDO3 to drive temperature 0x2013 (voltage)

  Send: 0x601 23 02 1A 03 10 00 13 20

  Feedback: 0x581 60 02 1A 03 00 00 00 00

  Index 0x1A02, subindex 0x03.

  Mapped objects:Temperature 0x2013, 0x0010, data length 16 bits;

- Set the 4th object mapped by TPDO3 to drive temperature 0x203C (temperature)

  Send: 0x601 23 02 1A 04 10 00 3C 20

  Feedback: 0x581 60 02 1A 04 00 00 00 00

Index 0x1A02, subindex 0x04.
Mapping object:temperature 0x203C, 0x0010, data length 16 bits;

- Set the feedback time of TPDO: (TPDO1:0x1800, TPDO2:0x1801, TPDO3:0x1802, TPDO4:0x1803)

Example: TPDO1
Send: 0x601 2B 00 18 05 E8 03 00 00
Feedback: 0x581 60 00 18 05 00 00 00 00
Index 0x1800, subindex 0x05.
Feedback time: 1000ms, 0x03E8 (low before, high after)

RPDO mapping object; (RPDO1: 0x1600, RPDO2: 0x1601)
Example:

- Set the 1st object of RPDO1 mapping to (A enable)

Send: 0x601 23 00 16 01 20 00 7A 60
Feedback: 0x581 23 00 16 01 20 00 00 00
Index 0x1A02, subindex 0x04.
Mapping object:enable 0x607A, 0x0020, data length 32 bits;

- Set the 2nd object mapped by RPDO1 to (A action data)

Send: 0x601 23 00 16 02 20 00 81 60
Feedback: 0x581 23 00 16 02 20 00 00 00
Index 0x1A02, subindex 0x04.
Mapping object:Temperature 0x6081, 0x0020, data length 32 bits;

The drive maps TPDO uploaded data by default (1000ms):
(0x281 is B motor data, same format as 0x181 A motor data)

- TPDO1 uploaded data: 0x181 DC 05 01 00 92 99 2A 01

8-bit Hexadecimal
0x05DC: Current motor speed 1500RPM
0x01: Current motor current 1A
0x012A9992: Current running position 19569042

- TPDO2 upload data: 0x281 is B motor data, same format as 0x181 A motor data.

- TPDO3 upload data: 0x381 01 08 01 00 30 00 1E 00

8-bit Hexadecimal
0x0801: Current A fault (deactivation + Hall fault) 0x0001: Current B fault (deactivation)
0x0030: Current bus voltage 48V 0x001E: Current drive temperature 30℃

- TPDO4 upload data: 0x481 14 51 14 51

  4-bit hexadecimal

  0x5114: Current control status feedback A-way enable; (see 7, Control Status Feedback for details)

7. Control status feedback:

A - 2003; B - 2004; Add TPDO4: Feedback two data A road status and B road status.

For example, if the uploaded data is 0x514C, the current status of the drive is the part marked red in the illustration.

0X514C

	5	1	4
State Catego	Control M	Feedback Mode	Operating Mode Enable State
1	Analog	Encoder	Speed
2	CAN	Hall Open - Loop	Torque
3	RS232	AS5147	Absolute Position
4	RC	Rotary Transformer	Relative Position
5	CANopen		
6	Pulse	Servo Encoder	
7	RS485		
8		Hall Closed - Loop	
9		Hall + Encoder	
A			
B		Absolute Encoder	
		Linear Encoder	

	C			
	Enable State			
	Bit3	Bit2	Bit1	Bit0
0	B - channel Disabled	A - channel Disabled		
1	B - channel Enabled	A - channel Enabled		

Note:
In the single - channel driver, the B - channel information is invalid. Only pay attention to the A - channel.
In the diagram, the highlighted (marked in red) part is the status of the current driver.

Version	Revision	Revision Date	Revised by
V1.0	Initial version	2024.8.23	Liu Li
V1.1	1. Addition of drain threshold; 2. RS485 command description revision; 3. Product default configuration image replacement;	2024.10.29	Li Zhongjian
V1.2	1. Revision of data return position for motor current query, fault query and motor speed query of CAN instruction; 2. Modify the description of heartbeat return instruction; 3. Completing the number of data bits returned by the controller for the 232 query instruction;	2024.12.27	Liu Li
V1.3	Modify the interface definition and power line terminal model;	2025.2.3	Liu Li
V1.4	Add 185 one-piece wheel description and combine it with 165 one-piece wheel description into one;	2025.6.5	Cao Fulu
V1.41	Add rated line speeds for 165/180 wheel sets;	2025.6.13	Cao Fulu