CyberGear Micromotor User Manual

Table of contents

Precautions

1. Please use it according to the working parameters specified in this article, otherwise it will cause serious damage to the product! 2. The control mode cannot be switched when the joint is running. If you need to switch, you need to send a stop command before switching.

3. Please check whether all parts are intact before use. If any parts are missing or damaged, please contact technical support in time.

4. Do not disassemble the motor at will to avoid irreversible failure. 5. Ensure that there is no short circuit when connecting the motor and the interface is correctly connected as required.

Legal Notice

Before using this product, please read this manual carefully and operate this product in accordance with the contents of this manual. If the user violates the contents of this manual and uses this product, the company will not assume any responsibility for any property loss or personal injury caused. Since this product is composed of many parts, children should not be allowed to touch this product to avoid accidents. To extend the service life of the product, do not use this product in a high temperature or high pressure environment. This manual has been printed to include the introduction of various functions and instructions for use as much as possible. However, due to the continuous improvement of product functions and design changes, there may still be discrepancies with the products purchased by users.

This manual may differ from the actual product in terms of color, appearance, etc., so please refer to the actual product. This manual is published by Xiaomi or its local subsidiaries. Xiaomi may make necessary improvements and changes to the typographical errors, inaccurate latest information in this manual, or improve the program and/or equipment at any time without prior notice. Such changes will be uploaded to the new version of this manual. Please scan the QR code of this manual to obtain it. All pictures are for reference only for functional descriptions. Please refer to the actual product.

After-sales policy

The after-sales service of this product is strictly implemented in accordance with the "Consumer Protection Law of the People's Republic of China" and the "Product Quality Law of the People's Republic of China". The service content is as follows:

1. Warranty period and content

(1) Any user who places an order for this product through online channels can enjoy the unconditional return service within seven days from the day after the receipt. When returning the goods, the user must present a valid purchase receipt and return the invoice. The user must ensure that the returned goods maintain their original quality and function, the appearance is intact, and the trademarks and various logos of the goods and accessories are intact. If there are any gifts, they must be returned together. If the goods are damaged by human factors, disassembled by human factors, the packaging box is missing, or the spare parts are missing, they will not be returned. The logistics costs incurred during the return shall be borne by the user (see the "After-sales Service Fee Standard" for the charging standard). If the user has not settled the logistics costs, the actual amount incurred will be deducted from the refund amount. The paid amount will be returned to the user within seven days from the date of receipt of the returned goods. The refund method is the same as the payment method. The specific arrival date may be affected by factors such as banks and payment institutions.

(2) Within 7 days from the day after the user signs for the product, if there is any performance failure caused by non-human damage, the Xiaomi After-Sales Service Center will check and confirm the product for you. When returning the product, the user must present a valid purchase receipt and return the invoice. Any gifts must be returned together with the product.

(3) If a non-human-caused damage or performance failure occurs within 7 to 15 days after the user signs for the product, the Xiaomi After-Sales Service Center will provide a replacement service for the user and replace the entire product after inspection and confirmation. After the replacement, the three-year warranty period of the product will be recalculated.

(4) From 15 days to 365 days after the user signs for the product, if the product is found to be defective after inspection and confirmation by the Xiaomi After-Sales Service Center, free repair service will be provided. The replaced defective product will become the property of Xiaomi. Non-defective products will be returned in their original condition. This product is shipped after undergoing various strict inspections. If there is a quality failure not caused by the product itself, we have the right to refuse the user's return or exchange request.

If the after-sales policy in this manual is inconsistent with the after-sales policy of the store, the after-sales policy of the store shall prevail. 2.

Non-warranty regulations The following situations are not covered by the warranty:

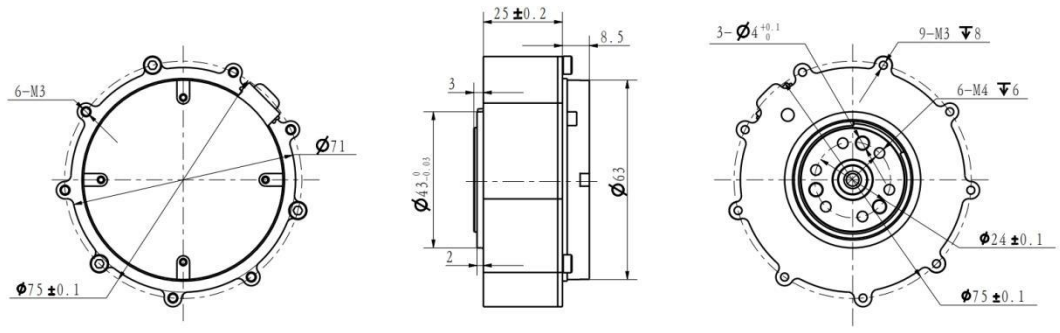
1. Exceeding the warranty period specified in the warranty terms.
2. Product damage caused by failure to comply with the instructions or improper use.
3. Damage caused by improper operation, maintenance, installation, modification, testing or other improper use.
4. Routine mechanical loss and wear not caused by quality failure.
5. Damage caused by abnormal operating conditions, including but not limited to falling, impact, liquid immersion, severe impact, etc.
6. Damage caused by natural disasters (such as floods, fires, lightning strikes, earthquakes, etc.) or force majeure.
7. Damage caused by use exceeding the peak torque.
8. The product is not a genuine Xiaomi product or the legal proof of purchase cannot be provided.
9. Failure or damage caused by other issues not related to product design, technology, manufacturing, quality, etc.
10. Use this product for commercial purposes.

If the above situation occurs, the user will be responsible for paying the fees.

For details of the group's after-sales policy, please visit: https://www.mi.com/service/serviceAgreement?id=17

1 Motor Specifications

1.1 Appearance and installation dimensions



1.2 Standard usage status

1.2.1 Rated voltage: 24 VDC

1.2.2 Operating voltage range: 16V—28 VDC

1.2.3 Rated load (CW): 4 Nm

1.2.4 Running direction: CW/CCW Viewed from the shaft exit direction

1.2.5 Usage posture: the axis direction is horizontal or vertical

1.2.6 Standard operating temperature: 25±5℃

1.2.7 Operating temperature range: -20～50℃

1.2.8 Standard operating humidity: 65%

1.2.9 Operating humidity range: 5~85%, no condensation

1.2.10 Storage temperature range: -30～70℃

1.2.11 Insulation Class: Class B

1.3 Electrical characteristics

1.3.1 No-load speed: 296 rpm ± 10%

1.3.2 No-load current: 0.5 Arms

1.3.3 Rated load: 4 Nm

1.3.4 Rated load speed: 240rpm±10%

1.3.5 Rated load current (peak): 6.5A±10%

1.3.6 Peak load: 12 Nm

1.3.7 Peak current (peak): 23A±10%

1.3.8 Insulation resistance/stator winding: DC 500VAC, 100M Ohms

1.3.9 High voltage resistance/stator and housing: 600 VAC, 1s, 2mA

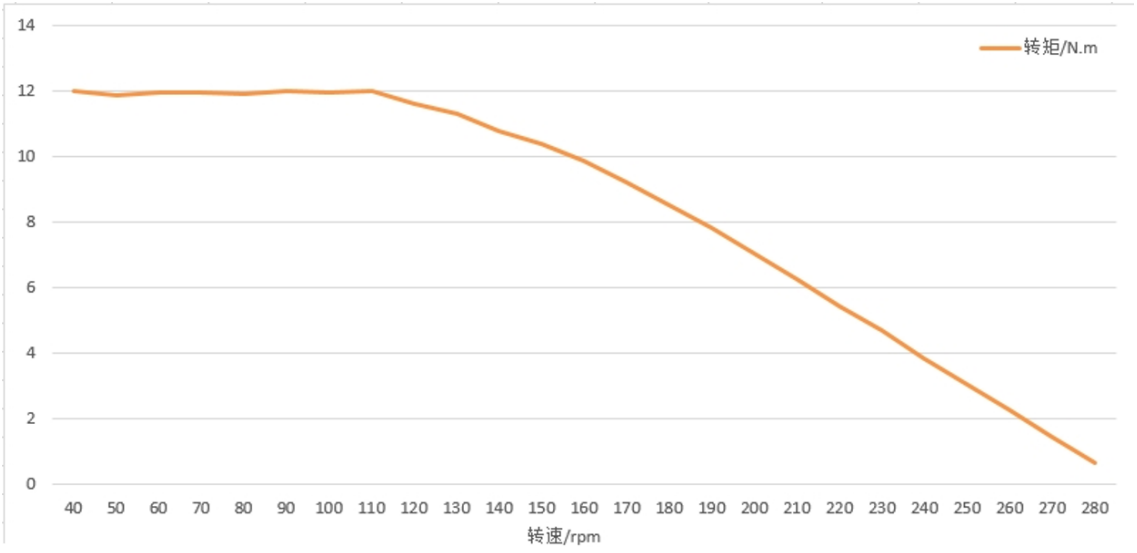
1.3.10 Motor back EMF: 0.054-0.057Vrms/rpm

1.3.11 Line resistance: 0.45Ω±10%

1.3.12 Torque constant: 0.87Nm/Arms

1.3.13 Motor inductance: 187-339μH

1.3.14 TN Curve



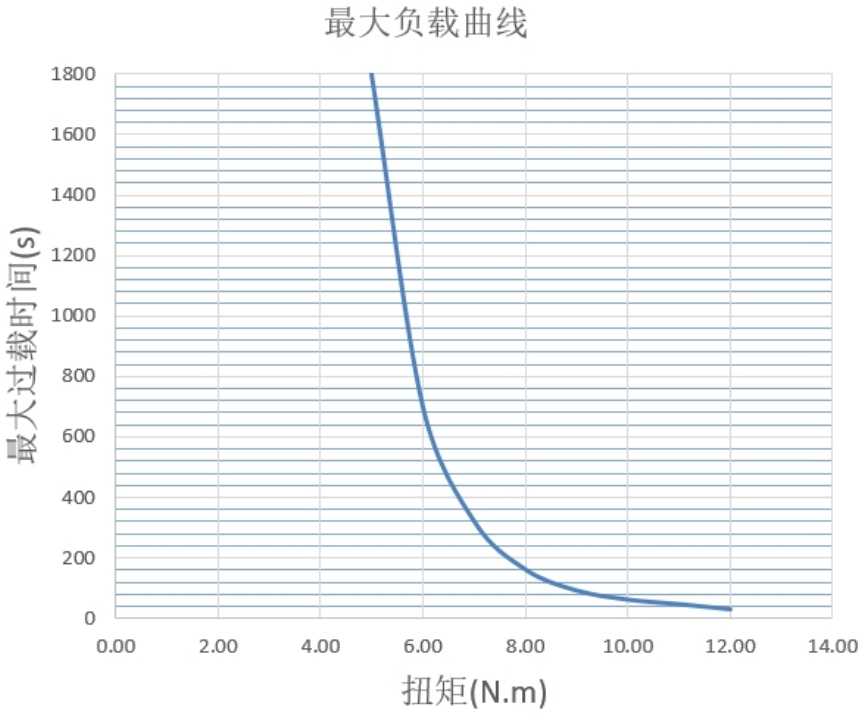
1.3.15 Maximum overload curve

Test conditions:

Ambient temperature: 25℃

Winding limit temperature: 120℃

Speed: 24rpm



Load	Operating time(s)
12.00	28
11.00	45
10.00	60
9.00	90
8.00	160
7.00	320
6.00	700
5.00	1800
4.50	2500
4.00	rated

Test Data

1.4 Mechanical properties

1.4.1 Weight: 317g ± 3g

1.4.2 Number of poles: 28 poles

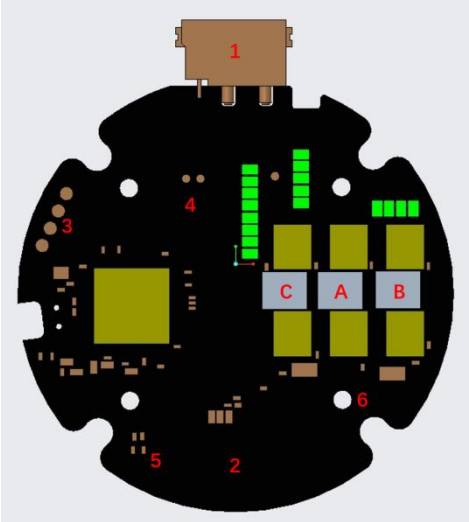
1.4.3 Phases: 3

1.4.4 Drive mode: FOC

1.4.5 Reduction ratio: 7.75:1

2 Driver Product Information

2.1 Driver Appearance and Product Specifications



1.24V power supply and CAN communication integrated terminals;

2. Hardware version and laser engraved QR code;

3.MCU download port;

4.CAN communication test point;

5. Indicator light;

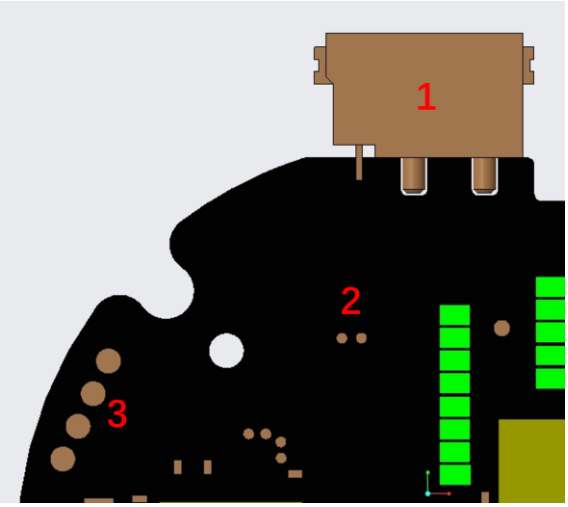
6. Mounting holes;

7. "C, A, B" are the welding points of the three-phase winding;

Product Specifications	
Rated operating voltage	24VDC
Maximum allowable voltage	28VDC
Rated operating current	6.5A
Maximum allowable current	23A
Standby power consumption	⩽18mA
CAN bus bit rate	1Mbps
size	Φ58mm
Operating temperature	- 20℃ to 50℃
Maximum allowable temperature of control board	80℃
Encoder resolution	14bit (single-turn absolute value)

2.2 Driver interface definition

2.2.1 Driver interface diagram

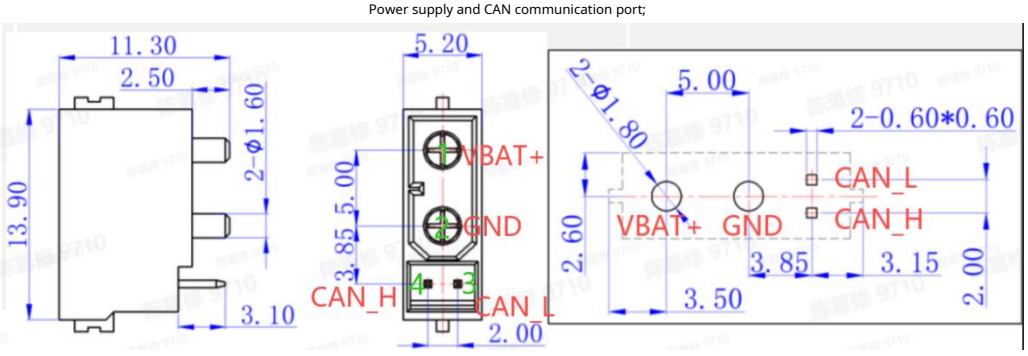


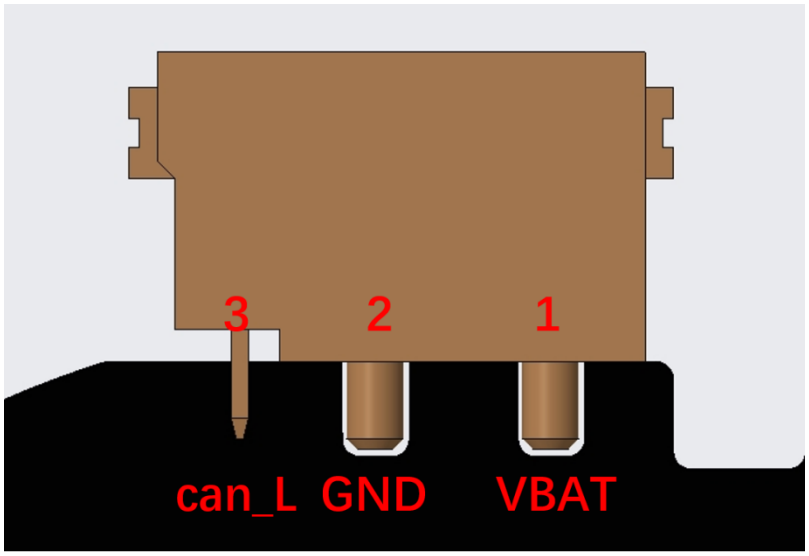
2.2.2 Recommended brands and models of driver interfaces

sequence Number	Board end model	Brand manufacturers	Line end model	Brand manufacturers
1	XT30PB(2+2)-MGB	AMASS Sri Lanka	XT30(2+2)-FGB	AMASS Sri Lanka
2	2.0mm-2P pad	/	2.0mm-2P Probe	/
3	2.54mm-4P solder pad	/	2.54mm-4P Probe	/

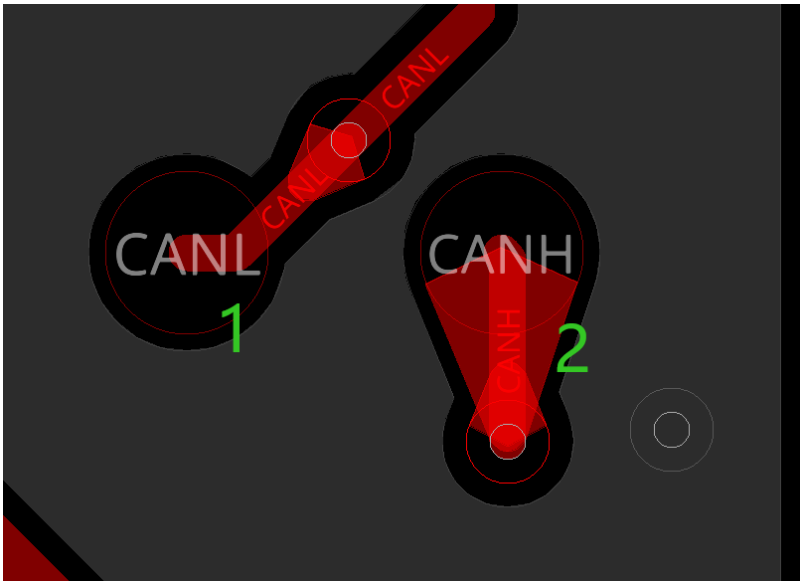
2.2.3 Driver interface pin definition

Power supply and CAN communication port;

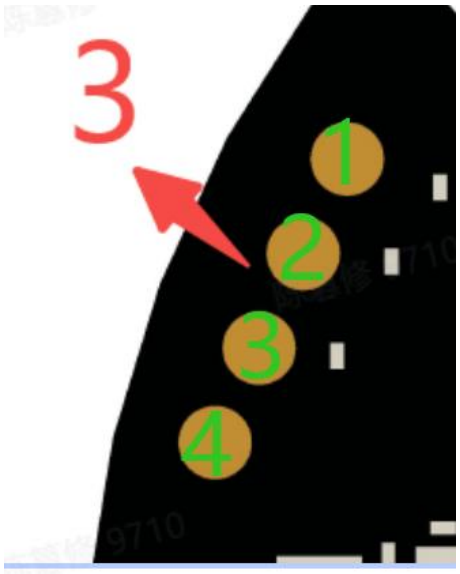




CAN communication test pads

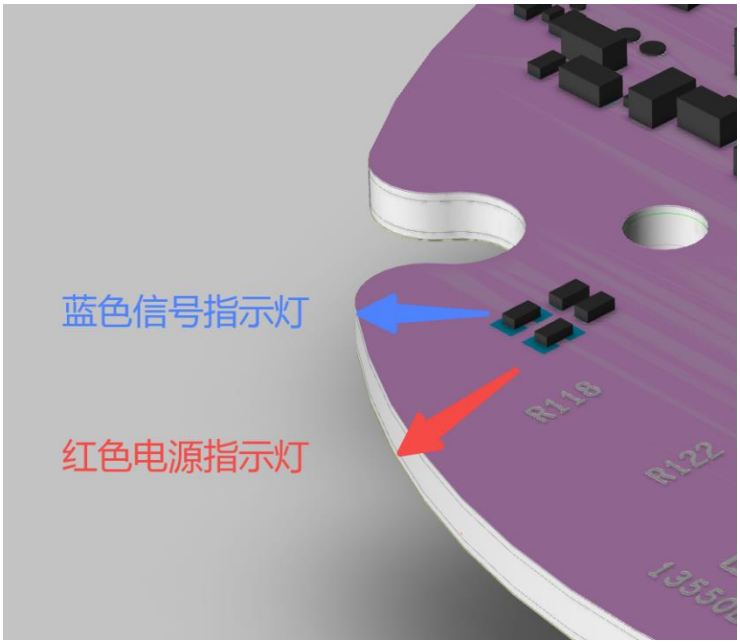


Download Port



Serial number	Interface Function	Pinout	illustrate
1	Power supply and CAN communication	1	Positive power supply (+)
		2	Negative pole of power supply (-)
		3	CAN communication low side CAN_L
		4	CAN communication high side CAN_H
2	CAN communication test points	1	CAN communication low side CAN_L
		2	CAN communication high side CAN_H
3	Download Port	1	SWDIO (data)
		2	SWCLK (clock)
		3	3V3 (positive 3.3V)
		4	GND (negative ground)

2.3 Definition of drive indicator lights



Indicator lamp definition	
Power indicator (Red when bright lamp)	The power indicator light is used to indicate the MCU 3.3V power supply status. When the total power input is 24V, the light is red, which proves that the power supply of the entire network is normal; if the indicator light is not on when the 24V power input is on, the power supply needs to be disconnected immediately;
Signal indicator (Blue when bright lamp)	When the signal light flashes, it proves that the MCU is running normally; and the driver chip is running normally;

2.4 Main components and specifications

Serial number	project	Specification	quantity
1	MCU Chip	GD32F303RET6	1 PCS
2	Driver chip	6EDL7141	1 PCS
3	Magnetic encoder chip	AS5047P	1 PCS
4	Thermistor	NXFT15XH103FEAB021/NCP18XH103F03RB	2 PCS
5	Power MOS	JMGG031V06A	6 PCS

3 Debugger Instructions (scan the end of the paper manual The last QR code gets the debugger)

3.1 Hardware Configuration

The joint motor uses CAN communication. There are two communication lines, which are connected to the debugger through the CAN to USB tool. The debugger needs to install the ch340 driver in advance and works in AT mode by default.

It should be noted that we developed the debugger based on a specific CAN to USB tool, so you need to use the serial port tool we recommend to debug the debugger. If you want to port it to other debugger platforms, you can refer to Chapter 3 of the manual for development.

The can to USB tool recommends using YourCee's USB-CAN module, and the corresponding serial port protocol frame header is 41 54, and the frame tail is 0D 0A.

3.2 Debugger interface and description

Mainly include:

A. Module Selection

- Equipment Module
- Configuration Module
- Analysis Module
- Help module

B. Submodule selection

Equipment modules include

- Connecting or disconnecting electrical equipment
- Motor equipment information
- Motor encoder calibration
- Modify the motorCAN ID
- Set the mechanical zero position of the motor
- Motor program upgrade

The configuration modules include:

- Parameter table, you can view and modify the motor parameters
- Upload parameters, you can upload the motor parameters to the parameter table
- Download parameters, you can download the data in the parameter table to the motor
- Export parameters, you can download the data in the parameter table to your local computer
- Restore to factory, you can restore the data in the parameter table to factory settings
- Clear warnings, you can clear motor errors, such as overtemperature, etc.

Analysis modules include:

- Oscilloscope, you can view the curve of parameter changes over time
- Frequency, you can adjust the frequency of viewing data
- Channel, you can configure the data to be viewed
- Start and stop drawing
- Output waveform data to local

The help modules include:

- Instructions for use, you can open the instruction manual
- About, you can view software information

C. Motor information query

- Device Information
- Parameter table information

D. Data Column

- Log information
- Communication Information

E. Run debug area

- Select device
- Convenient operation area, can quickly control the motor forward and reverse
- Motion control area, which can control the motor to run in various modes

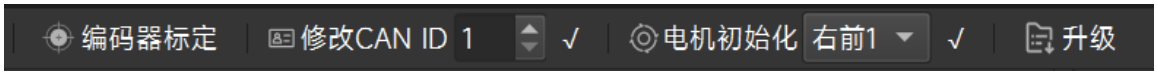
F. Submodule display area

3.3 Motor Settings

3.3.1 Motor connection settings

Connect the CAN to USB tool (install the ch340 driver, which works in AT mode by default), select the device module, click the connection submodule, and select the corresponding serial port connection.

3.3.2 Basic Settings



(1) Modify the motor ID number.

(2) Motor magnetic code calibration: reinstalling the motor board and motor, or reconnecting the motor wires in a different order, etc. requires recalibrating the magnetic code.

(3) Set zero position (lost in case of power failure), set the current position to 0. (4) Motor program upgrade, when the motor program is updated, click the upgrade button and select the upgrade file to upgrade.

3.3.3 Parameter table

After successfully connecting the motor, click the parameter table module in the configuration module. The log will show that all parameters have been loaded successfully, indicating that the motor-related parameters have been read successfully (Note: the parameter table needs to be configured when the motor is in standby state. If the motor is in running state, the parameter table cannot be refreshed). The interface will display the relevant parameters of the motor. The blue parameters are the storage parameters inside the motor, which can be modified in the current value column behind the corresponding parameters. Click Download Parameters to download the parameters in the debugger to the motor. Click Upload Parameters to upload the parameters in the motor to the debugger. The green parameters restored by the motor are the observed parameters, which are the collected parameters and can be observed in real time.

Note: Do not change the motor's torque limit, protection temperature, and over-temperature time at will. Our company will not assume any legal responsibility for any harm to the human body or irreversible damage to the joints caused by illegal operation of this product.

Parameter Table							
Function code	name	Parameter Class type	property	most big value	Minimum value	Current value (for reference)	Remark
0X0000	Name	String	Read/Write			yyyyyyyyyyyyyyy	
0X0001	BarCode	String	Read/Write			yyyyyyyyyyyyyyy	
0X1000	BootCodeVersion	String	Read-only			0.1.5	
0X1001	BootBuildDate	String	Read-only			Mar 16 2022	
0X1002	BootBuildTime	String	Read-only			20:22:09	
0X1003	AppCodeVersion	String	Read-only			0.1.5	Motor program version number
0X1004	AppGitVersion	String	Read-only			7b844b0fM	
0X1005	AppBuildDate	String	Read-only			April 14, 2022	
0X1006	AppBuildTime	String	Read-only			20:30:22	
0X1007	AppCodeName	String	Read-only			dog_motor	
0X2000	echoPara1	uint16	Configuration	74	5	5	
0X2001	echoPara2	uint16	Configuration	74	5	5	

0X2002	echoPara3	uint16	Configuration	74	5	5	
0X2003	echoPara4	uint16	Configuration	74	5	5	
0X2004	echoFreHz	uint32	Read/Write	100 00	1	500	
0X2005	MechOffset	float	set up	7	- 7	4.619583	Motor magnetic encoder angle Degree Bias
0X2006	MechPos_init	float	Read/Write	50	- 50	4.52	Initial multi-turn parameters Test angle
0X2007	limit_torque	float	Read/Write	12	0	12	Torque limit
0X2008	I_FW_MAX	float	Read/Write	33	0	0	Field weakening current value, default Recognize 0
0X2009	motor_index	uint8	set up	20	0	1	Motor index, standard Note the motor joint position
0X200a	CAN_ID	uint8	set up	127	0	1	This node id
0X200b	CAN_MASTER	uint8	set up	127	0	0	can host id
0X200c	CAN_TIMEOUT	uint32	Read/Write	100 000	0	0	can timeout threshold, Default 0
0X200d	motorOverTemp	int16	Read/Write	150 0	0	800	Motor protection temperature Value, temp (degrees) * 10
0X200e	overTempTime	uint32	Read/Write	100 000 0	1000	20000	Over temperature time
0X200f	GearRatio	float	Read/Write	64	1	7.75	Gear Ratio
0X2010	Tq_caliType	uint8	Read/Write	1	0	1	Torque calibration method Certainly
0X2011	cur_filt_gain	float	Read/Write	1	0	0.9	Current filter parameters
0X2012	cur_kp	float	Read/Write	200	0	0.025	Current kp
0X2013	cur_ki	float	Read/Write	200	0	0.0258	Current ki
0X2014	spd_kp	float	Read/Write	200	0	2	Speed kp
0X2015	spd_ki	float	Read/Write	200	0	0.021	Speed
0X2016	loc_kp	float	Read/Write	200	0	30	Location kp
0X2017	spd_filt_gain	float	Read/Write	1	0	0.1	Speed filter parameters
0X2018	limit_spd	float	Read/Write	200	0	2	Position mode speed limit system
0X2019	limit_cur	float	Read/Write	twenty three	0	twenty three	Position, speed mode Current Limitation

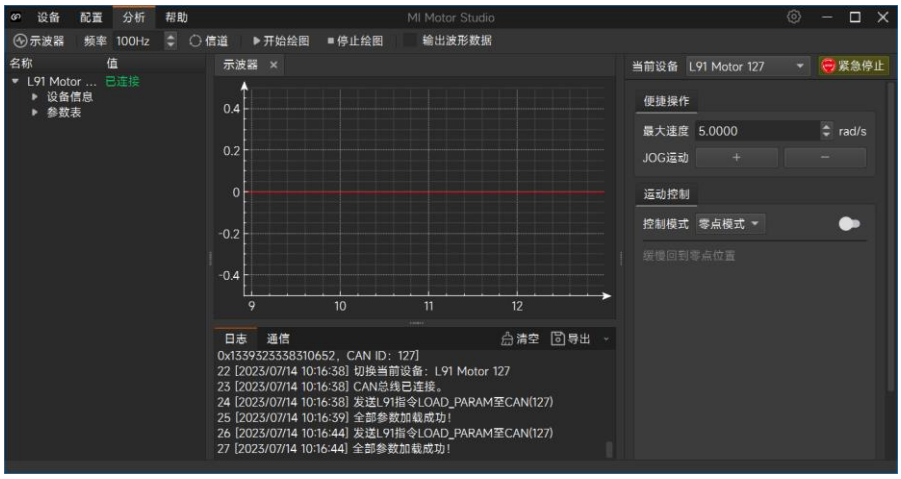
0X3000	timeUse0	uint16	Read-only			5	
0X3001	timeUse1	uint16	Read-only			0	
0X3002	timeUse2	uint16	Read-only			10	
0X3003	timeUse3	uint16	Read-only			0	
0X3004	encoderRaw	int16	Read-only			11396	Magnetic encoder sampling value
0X3005	mcuTemp	int16	Read-only			337	MCU internal temperature, * 10
0X3006	motorTemp	int16	Read-only			333	Motor ntc temperature, * 10
0X3007	vBus(mv)	uint16	Read-only			24195	Bus voltage
0X3008	adc1Offset	int32	Read-only			2084	adc sampling channel 1 Zero current bias
0X3009	adc2Offset	int32	Read-only			2084	adc sampling channel 2 Zero current bias
0X300a	adc1Raw	uint16	Read-only			1232	adc sampling value 1
0X300b	adc2Raw	uint16	Read-only			1212	adc sampling value 2
0X300c	VBUS	float	Read-only			24.195	Bus voltage V
0X300d	cmdId	float	Read-only			0	id ring instruction, A
0X300e	ikB	float	Read-only			0	iq ring instruction, A
0X300f	cmdlocref	float	Read-only			0	Position loop command, rad
0X3010	cmdspdref	float	Read-only			0	Speed loop command, rad/s
0X3011	cmdTorque	float	Read-only			0	Torque command, nm
0X3012	cmdPos	float	Read-only			0	MIT protocol perspective make
0X3013	cmdVel	float	Read-only			0	MIT protocol speed index make
0X3014	rotation	int16	Read-only			1	Number of laps
0X3015	modPos	float	Read-only			4.363409	Motor uncounted revolutions Angle, rad
0X3016	mechPos	float	Read-only			0.777679	Load end circle counter Angle, rad
0X3017	mechVel	float	Read-only			0.036618	Load end transfer Speed, rad/s
0X3018	elecPos	float	Read-only			4.714761	Electrical angle
0X3019	ia	float	Read-only			0	U Line current, A

0X301a	ib	float	Read-only			0	V Line current, A
0X301b	ic	float	Read-only			0	W Line current, A
0X301c	tick	uint32	Read-only			31600	
0X301d	phaseOrder	uint8	Read-only			0	Calibration direction mark
0X301e	iqf	float	Read-only			0	iq filter value, A
0X301f	boardTemp	int16	Read-only			359	Temperature on board, *10
0X3020	iq	float	Read-only			0	iq original value, A
0X3021	id	float	Read-only			0	id original value, A
0X3022	faultSta	uint32	Read-only			0	Fault Status Value
0X3023	warnSta	uint32	Read-only			0	Warning status value
0X3024	drv_fault	uint16	Read-only			0	Driver chip fault value
0X3025	drv_temp	int16	Read-only			48	Driver chip temperature Value, degree
0X3026	Uq	float	Read-only			0	q-axis voltage
0X3027	Ud	float	Read-only			0	d-axis voltage
0X3028	dtc_u	float	Read-only			0	U phase output duty cycle
0X3029	dtc_v	float	Read-only			0	V phase output duty cycle
0X302a	dtc_w	float	Read-only			0	W phase output duty cycle
0X302b	v_bus	float	Read-only			24.195	Vbus in closed loop
0X302c	v_ref	float	Read-only			0	Closed loop vq,vd synthesis Voltage
0X302d	torque_fdb	float	Read-only			0	Torque feedback value, nm
0X302e	rated_i	float	Read-only			8	Motor rated current
0X302f	limit_i	float	Read-only			27	Motor limit maximum power flow

3.3.4 Oscilloscope

This interface supports viewing graphs generated by observing real-time data. The observable data includes motor Id/Iq current, temperature, real-time speed of the output end, rotor (encoder) position, output end position, etc.

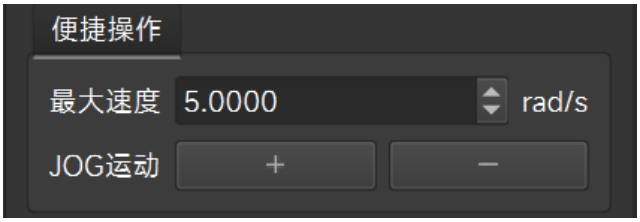
Click the oscilloscope module in the analysis module and select the appropriate parameters in the channel (the meaning of the parameters can be found in 3.3.3), after setting the output frequency, click Start Drawing to observe the data spectrum, and Stop Drawing to stop observing the spectrum.



3.4 Control Demonstration

jog operation:

Set the maximum speed, click Run, and then click JOG to make the motor run forward and reverse.



Control mode switching:

The motor control mode can be switched in the motion mode interface

3.4.1 Zero Point Mode

Click the switch button on the right and the motor will slowly return to the mechanical zero position.

3.4.2 Operation and Control Mode

Click the switch button on the right, set five parameter values, click Start or Send Continuously, the motor will return the feedback frame and run according to the target command; click the switch button on the right again, the motor will stop.

3.4.2 Current Mode

Manually switch the current mode, click the switch button on the right, then set the Iq current command value, start or send it continuously, the motor will run according to the current command, click the switch button on the right again, the motor will stop.

Click the switch button on the right side of the control mode, enter the amplitude and frequency of the sinusoidal automatic test, and then click the switch button on the right side of the sinusoidal automatic test. The iq (A) of the motor will run according to the set amplitude and frequency.

3.4.3 Speed Mode

Manually switch to speed mode, click the switch button on the right, then set the speed command value (-30~30rad/s), start or send continuously, the motor will follow the speed command, click the switch button on the right again, the motor will stop.

Click the switch button on the right side of the control mode, enter the amplitude and frequency of the sinusoidal automatic test, and then click the switch button on the right side of the sinusoidal automatic test. The motor speed (rad/s) will run according to the set amplitude and frequency.

3.4.4 Position Mode

To manually switch the position mode, click the switch button on the right, then set the position command value (rad), start or send continuously, the motor will follow the target position command, click the switch button on the right again, the motor will stop. You can modify the maximum speed of position following by setting the speed.

Click the switch button on the right side of the control mode, enter the amplitude and frequency of the sinusoidal automatic test, and then click the switch button on the right side of the sinusoidal automatic test. The motor position (rad) will run according to the set amplitude and frequency.

3.5 Firmware Update

The first step is to click on the upgrade of the device module and select the bin file to be burned; the second step is to confirm the upgrade, and the motor starts to update the firmware. After the progress is completed, the motor is updated and automatically restarts.

4 Driver communication protocol and instructions

The motor communication is CAN 2.0 communication interface, with a baud rate of 1Mbps and an extended frame format, as shown below:

Data Field	29-digit ID			8Byte data area
size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
describe	Communication Type	Data Area 2	Target Address	Data Area 1

The control modes supported by the motor include:

Operation and control mode: give 5 motor operation and control parameters;

Current mode: given a specified Iq current for the motor;

Speed mode: set the motor to a specified running speed;

Position mode: given a specified position for the motor, the motor will run to the specified position;

4.1 Communication protocol type description

4.1.1 Get device ID (communication type 0); Get device ID and 64-bit MCU only

An identifier

number domain	according to 29-digit ID			8Byte data area
size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
describe	0	bit15~8: used to identify the host CAN_ID	Target Motor CAN_ID	0

Response frame:

Data Field	29-digit ID			8Byte data area
size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
describe	0	Target motor CAN_ID	0XFE	64-bit MCU unique identifier

4.1.2 Operation and Control Mode Motor Control Command (communication type 1) is used to send control commands to the motor.

make

number according to domain	29-digit ID			8Byte data area
big Small	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
Description Description	1	Byte2: torque (0~65535) correspond(-12Nm~12Nm）	Target Motor CAN_ID	Byte0~1: target angle [0~65535] Corresponding to (-4π~4π) Byte2~3: Target angular velocity [0~65535] corresponds to (-30rad/s~30rad/s) Byte4~5: Kp [0~65535] corresponds to (0.0~500.0) Byte6~7: Kd [0~65535] corresponds to (0.0~5.0)

Response frame: Response to motor feedback frame (see communication type 2)

4.1.3 Motor feedback data (communication type 2) Used to feedback the motor running status to the host

number according to domain	29-digit ID			8Byte data area
big Small	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
Description Description	2	Bit8~Bit15: Current Motor CAN ID bit21~16: fault signal Information (0 no 1 yes) bit21: uncalibrated bit20: HALL encoding Fault bit19: Magnetic encoding error barrier bit18: Over temperature bit17: overcurrent bit16: undervoltage fault bit22~23: mode status state 0 : Reset mode [Reset] 1: Cali mode [label Certainly] 2: Motor mode [run]	Host CAN _ID	Byte0~1: Current angle [0~65535] corresponds to (-4π~4π) Byte2~3: Current angular velocity [0~65535] corresponds to (-30rad/s~30rad/s) Byte4~5: current torque [0~65535] corresponds to (-12Nm~12Nm) Byte6~7: Current temperature: Temp (degrees Celsius)*10

4.1.4 Motor Enable Operation (Communication Type 3)

data domain	29-digit ID			8Byte data district
size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
describe	3	bit15~8: used to identify the master CAN_ID	Target Motor CAN_ID	

Response frame: Response to motor feedback frame (see communication type 2)

4.1.5 Motor stop (communication type 4)

data domain	29-digit ID		8Byte data area
size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7

describe	4	bit15~8: used to identify the master CAN_ID	Target motor CAN_ID	During normal operation, the data area needs to be cleared 0; When Byte[0]=1: clear fault;

Response frame: Response to motor feedback frame (see communication type 2)

4.1.6 Setting the motor mechanical zero position (communication type 6) will set the current motor position to mechanical zero.

Bit (lost on power failure)

data domain	29-digit ID			8Byte data area
size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
describe	6	bit15~8: used to identify the main CAN_ID	Target motor CAN_ID	Byte[0]=1

Response frame: Response to motor feedback frame (see communication type 2)

4.1.7 Set the motor CAN_ID (communication type 7) Change the current motor CAN_ID, immediately

effect.

data domain	29-digit ID			8Byte data area
size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
describe	7	bit15~8: used to identify the main CAN_ID Bit16~23: Preset CAN_ID	Target motor CAN_ID	

Response frame: Response to motor broadcast frame (see communication type 0)

4.1.8 Single parameter reading (communication type 17)

data domain	29-digit ID			8Byte data area
size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
describe	17	bit15~8: used to identify the master CAN_ID	Target Motor CAN_ID	Byte0~1: index, parameter list See 4.1.11 for details of the table Byte2~3: 00 Byte4~7: 00

Response frame:

data domain	29-digit ID			8Byte data area
size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
describe	17	bit15~8: target motor CAN_ID	Host CAN_ID	Byte0~1: index, see 4.1.11 for parameter list Byte2~3: 00 Byte4~7: parameter data, 1 byte data in Byte4

4.1.9 Single parameter writing (communication type 18) (power failure loss)

data domain	29-digit ID			8Byte data area
size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
describe	18	bit15~8: used to identify the master CAN_ID	Target Motor CAN_ID	Byte0~1: index, parameter list details See 4.1.11 Byte2~3: 00 Byte 4~7: parameter data

Response frame: Response to motor feedback frame (see communication type 2)

4.1.10 Fault feedback frame (communication type 21)

data domain	29-digit ID			8Byte data area
size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
describe	twenty one	bit15~8: used to identify Primary CAN_ID	Motor CAN_ID	Byte0~3: fault value (non-0: fault, 0: normal) bit16: A phase current sampling overcurrent bit15~bit8: overload fault bit7: encoder is not calibrated bit5:C phase current sampling overcurrent bit4: B phase current sampling overcurrent bit3: overvoltage fault bit2: undervoltage fault bit1: driver chip failure bit0: Motor over-temperature fault, default is 80 degrees Byte4~7: warning value bit0: Motor over-temperature warning, default is 75 degrees

4.1.11 Baud rate modification (communication type 22) (version 1.2.1.5 can be modified, please refer to the document

Be careful when modifying the file process. If you make an operation error, you may be unable to connect to the motor or upgrade.)

Data Field	29-digit ID			8Byte data area
size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
describe	twenty two	bit15~8: used to identify the main CAN_ID	Target motor CAN_ID	Byte0: Motor baud rate 1: 1Mbps 2: 500kbps 3: 250kbps 4: 125kbps

Response frame: Response to motor broadcast frame (see communication type 0)

4.1.12 Can read and write single parameter list (7019-7020 is readable by firmware version 1.2.1.5)

parameter index	**Parameter name**	**describe**	**type**	**Character Festival number**	**Unit/Description**	**R/W read Write right limit**
0X7005	run_mode	0: Operation and control mode 1: Position mode 2: Speed Mode 3: Current mode	uint8	1		W/R
0X7006	iq_ref	Current mode Iq refers to make	float	4	- 23~23A	W/R
0X700A	spd_ref	Speed mode speed instruction	float	4	- 30~30rad/s	W/R
0X700B	imit_torque	Torque limit	float	4	0~12Nm	W/R
0X7010	cur_kp	Kp of current	float	4	Default value 0.125	W/R
0X7011	cur_ki	Ki of current	float	4	Default value 0.0158	W/R
0X7014	cur_filt_gain	Current filter coefficient filt_gain	float	4	0~1.0, default value 0.1	W/R
0X7016	loc_ref	Position mode angle instruction	float	4	rad	W/R

0X7017	limit_spd	Position mode speed limit	float	4	0~30rad/s	W/R
0X7018	limit_cur	Speed Position Mode Current Limitation	float	4	0~23A	W/R
0x7019	mechPos	Load end circle counter Mechanical Angle	float	4	rad	R
0x701A	iqf	iq filter value	float	4	- 23~23A	R
0x701B	mechVel	Load end speed	float	4	- 30~30rad/s	R
0x701C	VBUS	Bus voltage	float	4	V	R
0x701D	rotation	Number of laps	int16	2	Number of laps	W/R
0x701E	loc_kp	kp of location	float	4	Default value: 30	W/R
0x701F	spd_kp	Speed kp	float	4	Default value 1	W/R
0x7020	spd_ki	Speed ki	float	4	Default value 0.002	W/R

4.2 Control Mode Instructions

4.2.1 Program Example

The following are examples of controlling motors in various modes (taking gd32f303 as an example). The following are library calls, function and macro definitions for various examples.

```
#define P_MIN -12.5f
#define P_MAX 12.5f
#define V_MIN -30.0f
#define V_MAX 30.0f
#define KP_MIN 0.0f
#define KP_MAX 500.0f
#define KD_MIN 0.0f
#define KD_MAX 5.0f
#define T_MIN -12.0f
# define T_MAX 12.0f
struct exCanIdInfo{
uint32_t id:8;
uint32_t data:16;
```

```
uint32_t mode:5;
uint32_t res:3;
};
can_receive_message_struct rxMsg;
can_trasnmit_message_struct txMsg={
  . tx_sfid = 0,
  . tx_efid = 0xff,
  . tx_ft = CAN_FT_DATA,
  . tx_ff = CAN_FF_EXTENDED,
  . tx_dlen = 8,
};
# define txCanIdEx (((struct exCanIdInfo)&(txMsg.tx_efid)))
# define rxCanIdEx (((struct exCanIdInfo)&(rxMsg.rx_efid))) //Parse the extended frame id into a custom data structure
int float_to_uint(float x, float x_min, float x_max, int bits){
  float span = x_max - x_min;
  float offset = x_min;
  if(x > x_max) x=x_max; else if(x
  < x_min) x= x_min;
  return (int) ((x-offset)*((float)((1<<bits)-1))/span);
}
# define can_txd() can_message_transmit(CAN0, &txMsg)
# define can_rxd() can_message_receive(CAN0, CAN_FIFO1, &rxMsg)
```

The following are some common types of communications sent:

1. Motor enable running frame (communication type 3)

```
    void motor_enable(uint8_t id, uint16_t master_id) {

      txCanIdEx.mode = 3;
      txCanIdEx.id = id;
      txCanIdEx.res = 0;
      txCanIdEx.data = master_id;
      txMsg.tx_dlen = 8;
      txCanIdEx.data = 0;
      can_txd();
    }
```

2. Operation mode motor control instructions (communication type 1)

```
    void motor_controlmode(uint8_t id, float torque, float
    MechPosition, float speed, float kp, float kd)
    {
      txCanIdEx.mode = 1;
      txCanIdEx.id = id;
      txCanIdEx.res = 0;
      txCanIdEx.data = float_to_uint(torque,T_MIN,T_MAX,16);
      txMsg.tx_dlen = 8;
```

```
    txMsg.tx_data[0]=float_to_uint(MechPosition,P_MIN,P_MAX,16)>>8;
    txMsg.tx_data[1]=float_to_uint(MechPosition,P_MIN,P_MAX,16);
    txMsg.tx_data[2]=float_to_uint(speed,V_MIN,V_MAX,16)>>8;
    txMsg.tx_data[3]=float_to_uint(speed,V_MIN,V_MAX,16);
    txMsg.tx_data[4]=float_to_uint(kp,KP_MIN,KP_MAX,16)>>8;
    txMsg.tx_data[5]=float_to_uint(kp,KP_MIN,KP_MAX,16);
    txMsg.tx_data[6]=float_to_uint(kd,KD_MIN,KD_MAX,16)>>8;
    txMsg.tx_data[7]=float_to_uint(kd,KD_MIN,KD_MAX,16); can_txd();

  }
```

3. Motor stop frame (communication type 4)

```
  void motor_reset(uint8_t id, uint16_t master_id) {

    txCanIdEx.mode = 4;
    txCanIdEx.id = id;
    txCanIdEx.res = 0;
    txCanIdEx.data = master_id;
    txMsg.tx_dlen = 8;
    for(uint8_t i=0;i<8;i++) {

      txMsg.tx_data[i]=0;
    }
    can_txd();
  }
```

4. Motor mode parameter write command (communication type 18, operation mode switching)

```
  uint8_t runmode;
  uint16_t index;
  void motor_modechange(uint8_t id, uint16_t master_id) {

    txCanIdEx.mode = 0x12;
    txCanIdEx.id = id;
    txCanIdEx.res = 0;
    txCanIdEx.data = master_id;
    txMsg.tx_dlen = 8;
    for(uint8_t i=0;i<8;i++) {

      txMsg.tx_data[i]=0;
    }
    memcpy(&txMsg.tx_data[0],&index,2);
    memcpy(&txMsg.tx_data[4],&runmode, 1);
    can_txd();
  }
```

5. Motor mode parameter write command (communication type 18, control parameter write)

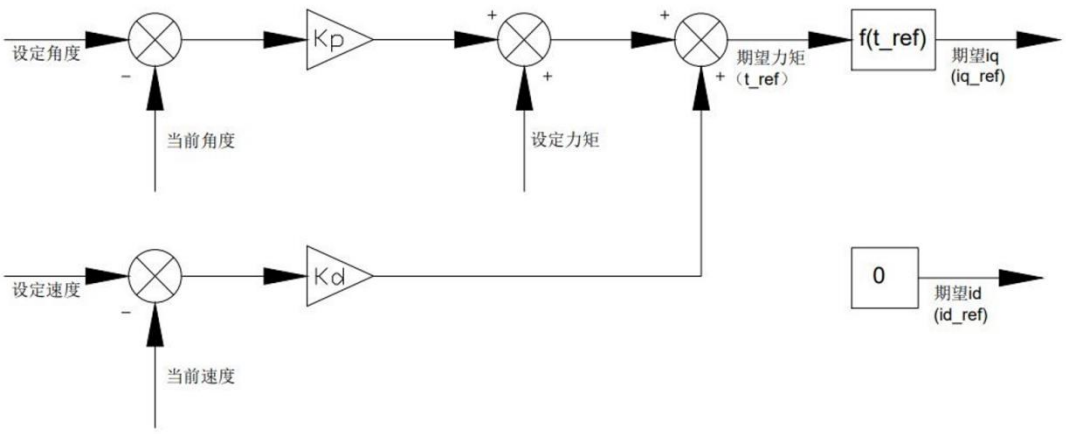
```
  uint16_t index;
```

```
float ref;
void motor_write(uint8_t id, uint16_t master_id) {

   txCanIdEx.mode = 0x12;
   txCanIdEx.id = id;
   txCanIdEx.res = 0;
   txCanIdEx.data = master_id;
   txMsg.tx_dlen = 8;
   for(uint8_t i=0;i<8;i++) {

      txMsg.tx_data[i]=0;
   }
   memcpy(&txMsg.tx_data[0],&index,2);
   memcpy(&txMsg.tx_data[4],&ref,4);
   can_txd();
}
```

4.2.2 Operation and control mode



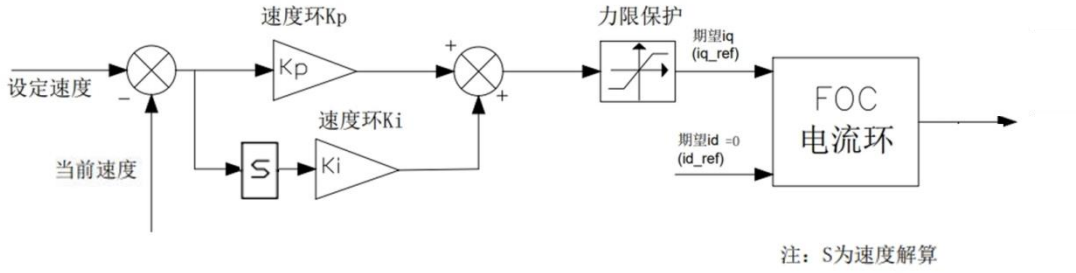
After the motor is powered on, it is in operation control mode by default;

Send motor enable frame (communication type 3) --> send motor control command in operation control mode (communication type 1) --> receive motor feedback frame (communication type 2)

4.2.3 Current Mode

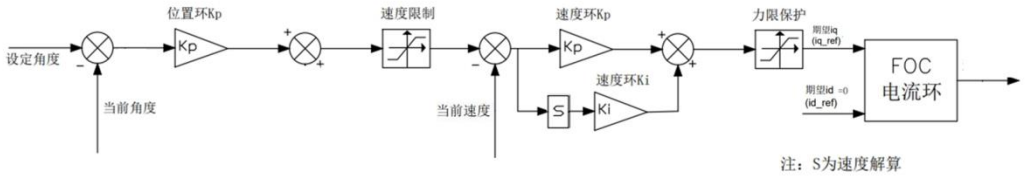
Send motor mode parameter write command (communication type 18) to set the runmode parameter to 3 ---> Send motor enable run frame (communication type 3) --> Send motor mode parameter write command (communication type 18) to set **iq_ref**The parameter is the preset current command

4.2.4 Speed Mode



Send motor mode parameter write command (communication type 18) Set runmode parameter to 2 ---> Send motor enable run frame (communication type 3) --> Send motor mode parameter write command (communication type 18) Set**limit_cur** The parameter is the preset maximum current instruction --> send the motor mode parameter write command (communication type 18) to set**spd_ref**The parameter is the preset speed command

4.2.5 Position Mode



Send motor mode parameter write command (communication type 18) Set runmode parameter to 1 --> Send motor enable run frame (communication type 3) --> Send motor mode parameter write command (communication type 18) Set**limit_spd**The parameter is the preset maximum speed command --> send the motor mode parameter write command (communication type 18) to set**loc_ref**Parameters are preset position instructions

4.2.6Stop running

Send motor stop frame (communication type 4)