CyberGear micromotor instruction manual

Table of contents

1. Motor specifications

   1.1 Shape and installation dimensions
   1.2 Standard usage status
   1.3 Electrical characteristics
   1.4 Mechanical properties

2. Drive product information

   2.1 Driver Appearance Introduction & Product Specifications
   2.2 Driver interface definition
   2.2.1 Driver interface diagram
   2.2.2 Recommended brands and models of drive interfaces
   2.2.3 Driver interface pin definition
   2.3 Drive indicator definitions
   2.4 Main components and specifications

3. Debugger usage instructions

   3.1 Hardware Configuration
   3.2 Debugger interface and instructions
   3.3 Motor settings
   3.4 control demo
   3.5 Firmware update

4. Driver communication protocol and usage instructions

   4.1 Communication protocol type description
   4.2 Control mode usage instructions

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 while 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 parts are missing or damaged, please contact technical support in time.
4. Please do not disassemble the motor at will to avoid irrecoverable faults.
5. Make sure there is no short circuit when connecting the motor and the interface is connected correctly as required.

Legal Notices

Before using this product, the user must read this manual carefully and operate the product in accordance with the contents of this manual. If the user uses this product in violation of the contents of this manual, any property damage or personal injury caused
Therefore, our company does not assume any responsibility. Because this product consists of many parts, do not let children come into contact with this product to avoid accidents. In order to extend the service life of the product, please do not use this product in high temperature and high pressure environments. This manual has tried its best to include various function introductions and usage instructions at the time of printing. However, due to the continuous improvement of product functions, design changes, etc., there may still be discrepancies with the products purchased by users.

There may be differences between this manual and the actual product in terms of color, appearance, etc. Please refer to the actual product. This manual is published by Xiaomi or its local subsidiaries. Xiaomi may correct any printing errors in this manual at any time.
We may make necessary improvements and changes to the latest information, errors, inaccuracies, or improvements to programs and/or equipment 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 functional description only, please refer to the actual product.

After-sales policy

The after-sales service of this product is strictly implemented in accordance with the "Consumer Rights and Interests 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） Users who place an order to purchase this product through online channels can enjoy a no-reason return service within seven days from the day of receipt. When returning goods, users must present a valid proof of purchase and return the invoice. Users must ensure that the returned goods maintain their original quality, functionality, appearance, and that the trademarks and logos of the goods themselves and accessories are complete and complete. If there are any gifts, they must be returned together. If the product is damaged artificially, dismantled manually, the packaging box is missing, or the spare parts are missing, returns will not be

processed. The logistics costs incurred when returning goods are borne by the user (See "After-sales Service Charging Standards" for charging standards). If the user fails to settle the logistics fees, the actual amount will be deducted from the refund amount. The paid price will be refunded to the user within seven days from the date of receipt of the returned goods. Refund methods are the same as payment methods. The specific arrival date may be affected by factors such as banks and payment institutions.

（2） If non-human-damaged performance failure occurs within 7 days from the day after the user signs for it, the Xiaomi after-sales service center will handle the return business for the user after inspection and confirmation. When returning the product, the user must present a valid purchase voucher and return the invoice. Any gifts must be returned together.

（3） If non-human damage or performance failure occurs from 7 days to 15 days from the day after the user signs for it, Xiaomi after-sales service center will handle the exchange business for the user and replace the entire set of products after inspection and confirmation. After the exchange, the three-guarantee period of the product itself will be recalculated.

（4） From 15 days to 365 days from the day after the user signs for it, if the Xiaomi after-sales service center detects and confirms that it is a quality failure of the product itself, free maintenance services will be provided. The replaced faulty product belongs to Xiaomi. Non-faulty product will be returned in its original condition. This product leaves the factory after undergoing various strict tests. If there is any quality failure that is not the product itself, we will have the right to refuse the user's return or exchange request.

If the after-sales policy in this manual is inconsistent with the store’s after-sales policy, the store’s after-sales policy shall prevail. 2. Non-warranty regulations: The following situations are not covered by the warranty:

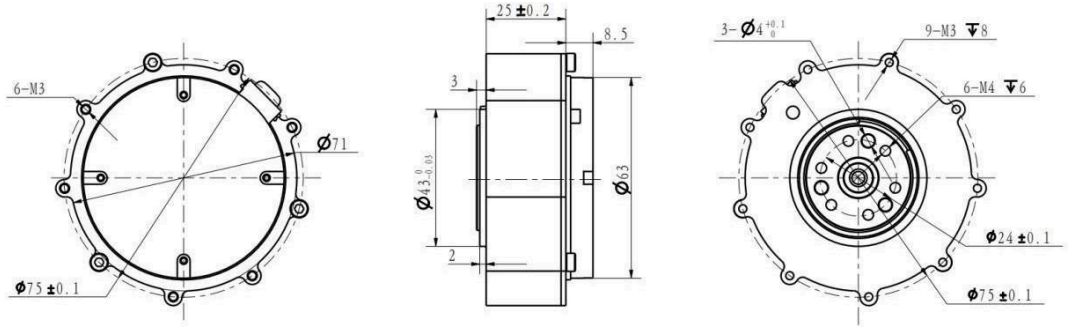
1. Exceeds the warranty period limited by the warranty terms.
2. Product damage caused by incorrect use without following the instruction manual.
3. Damage caused by improper operation, maintenance, installation, modification, testing and other improper use.
4. Conventional mechanical loss and wear caused by non-quality faults.
5. Damage caused under abnormal working conditions, including but not limited to drops, impacts, liquid immersion, violent impacts, etc.
6. Damage caused by natural disasters (such as floods, fires, lightning strikes, earthquakes, etc.) or force majeure.
7. Damage caused by use beyond peak torque.
8. Items that are not genuine Xiaomi products may not be able to provide legal proof of purchase.
9. Other failures or damages not caused by problems such as product design, technology, manufacturing, quality, etc.
10. Use this product for commercial purposes.

If the above situation occurs, users need to pay the fees themselves.

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

1 Motor specifications

1.1 Shape and installation dimensions

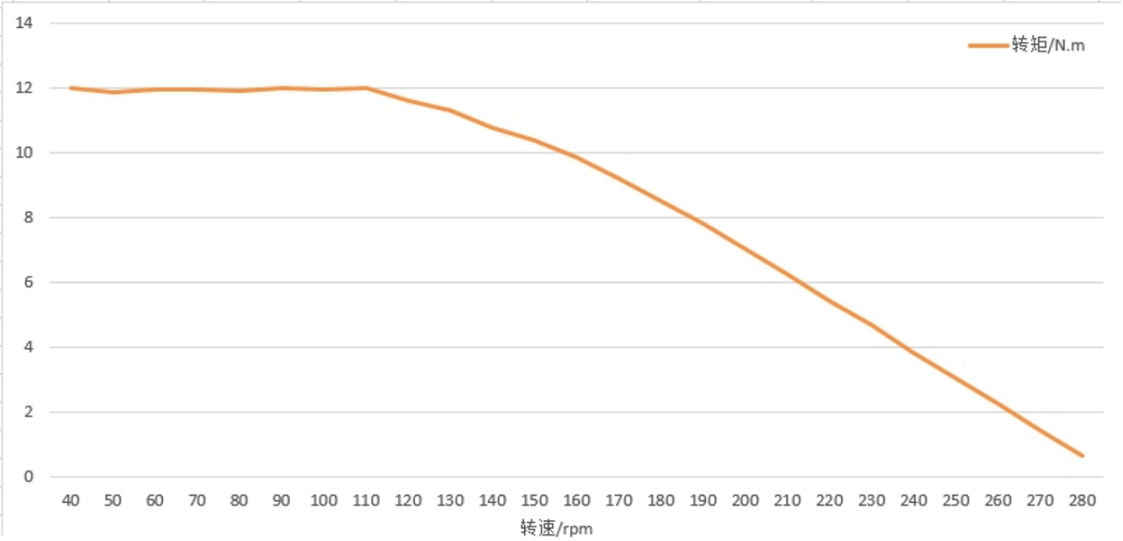


1.2 Standard usage status

1.2.1 Rated voltage: 24 VDC
1.2.2 Operating voltage range: 16V-28 VDC
1.2.3 Rated load (CW): 4 N.m
1.2.4 Running direction: CW/CCW looking from the direction of the shaft
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 level: Class B

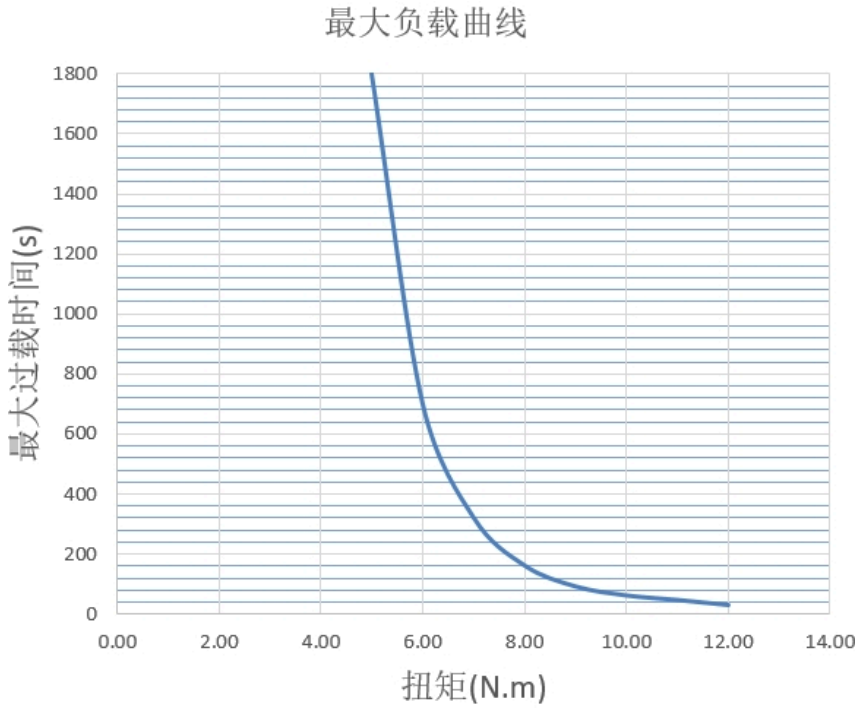
1.3 Electrical characteristics

1.3.1 No-load speed: 296 rpm±10%
1.3.2 No-load current: 0.5 Arms
1.3.3 Rated load: 4 N.m
1.3.4 Rated load speed: 240rpm±10%
1.3.5 Rated load current (peak): 6.5A±10%
1.3.6 Peak load: 12 N.m
1.3.7 Peak current (peak value): 23A±10%
1.3.8 Insulation resistance/stator winding: DC 500VAC, 100M Ohms
1.3.9 High voltage resistance/stator and casing: 600 VAC, 1s, 2mA
1.3.10 Motor back electromotive force: 0.054-0.057Vrms/rpm
1.3.11 Line resistance: 0.45Ω±10%
1.3.12 Torque constant: 0.87N.m/Arms
1.3.13 Motor inductance: 187-339μH
1.3.14 T-N 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

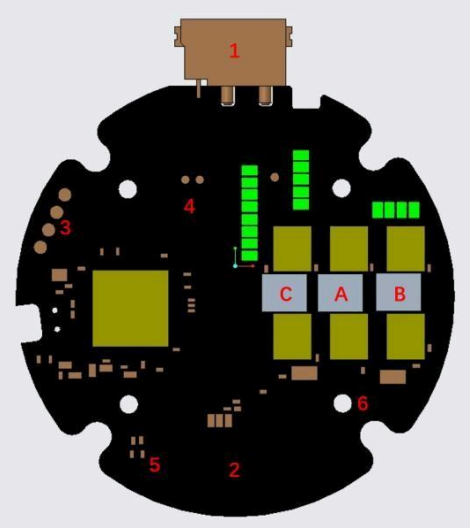
1.4 Mechanical properties

1.4.1 Weight: 317g±3g
1.4.2 Number of poles: 28 poles

1.4.3 Number of phases: 3 phases
1.4.4 Drive mode: FOC
1.4.5 Reduction ratio: 7.75:1

2 Drive product information

2.1 Driver Appearance Introduction & Product Specifications

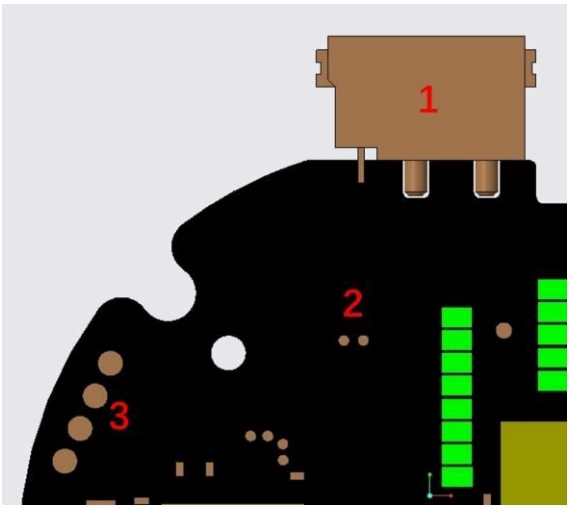


1.24V power supply and CAN communication integrated terminal;
2. Hardware version and laser engraving QR code;
3. MCU download port;
4. CAN communication test point;
5. indicator light;
6. Mounting holes;
7. “C, A, B” are three-phase winding welding points;

product specifications	
Rated working voltage	24VDC
maximum voltage allowed	28VDC
Rated operating current	6.5A
Maximum allowable current	23A
Standby power consumption	≤18mA
CAN bus bit rate	1Mbps
size	Φ58mm
Working temperature	-20℃ to 50℃
The maximum temperature allowed by the control panel	80℃
Encoder resolution	14bit (single lap absolute value)

2.2 Driver interface definition

2.2.1 Driver interface diagram

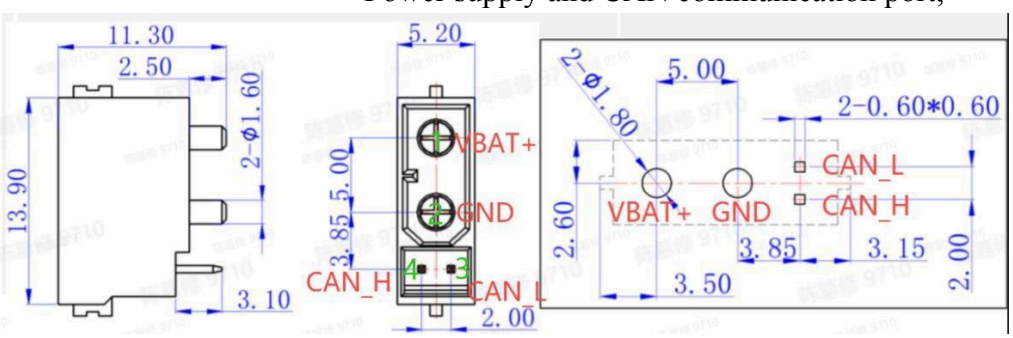


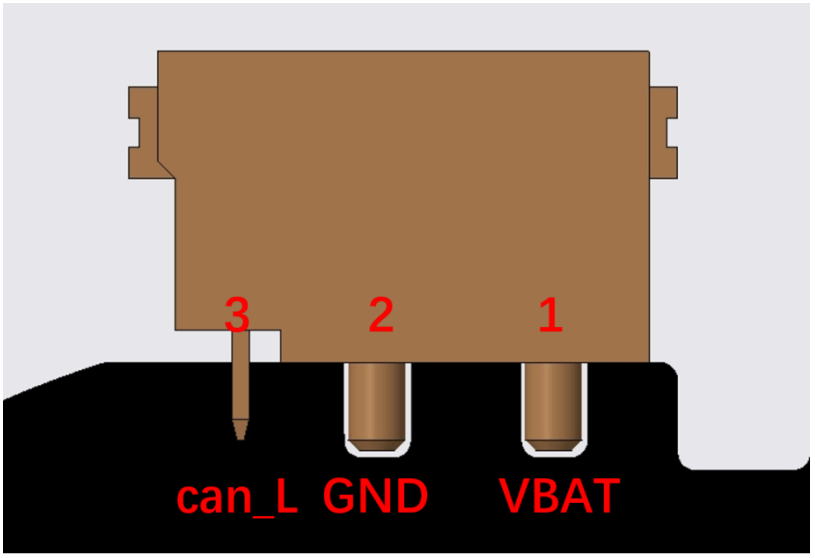
2.2.2 Recommended brands and models of drive interfaces

serial number	Board model	Brand manuf acturer	Line end model	Brand manufactu rer
1	XT30PB(2+ 2)- M.G.B	AMASS	XT30(2+2)- F.G.B	AMASS
2	2.0mm-2P pad	/	2.0mm-2P probe	/
3	2.54mm-4P pad	/	2.54mm-4P probe	/

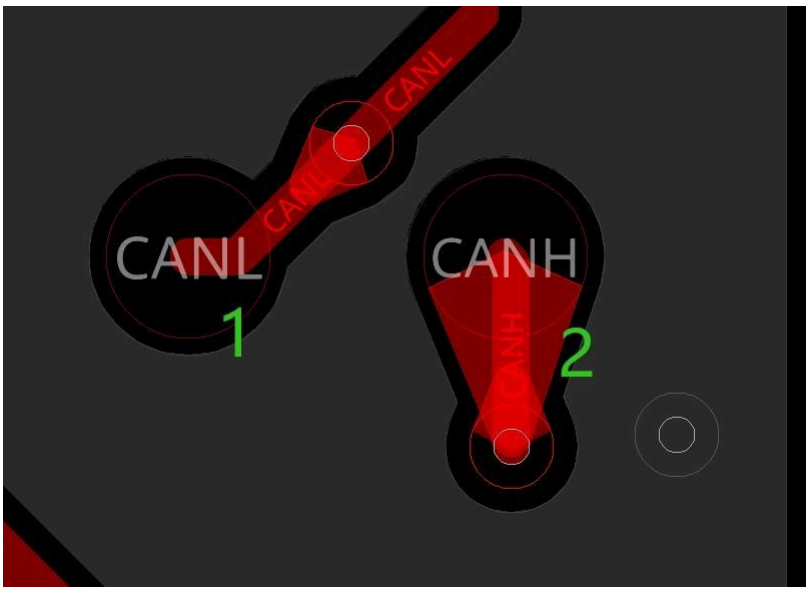
2.2.3 Driver interface pin definition

Power supply and CAN communication port;



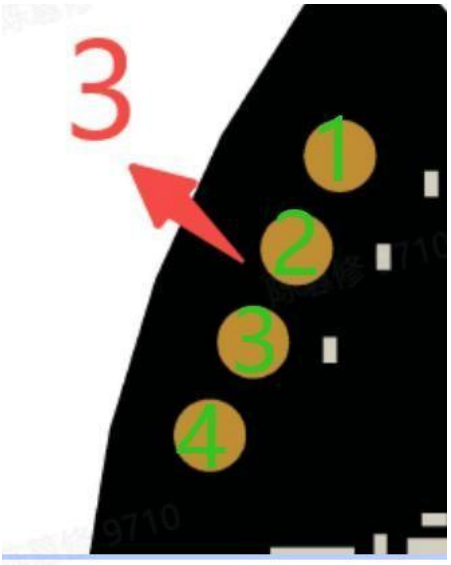


CAN



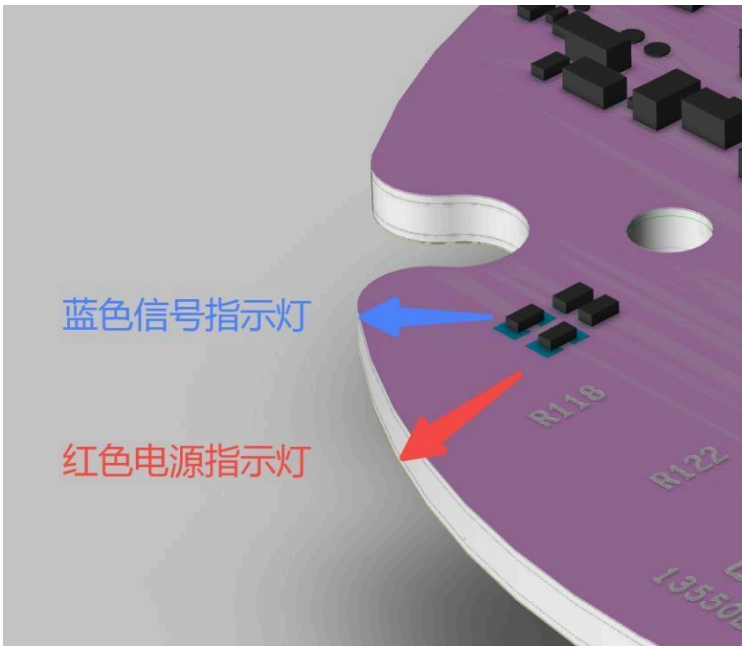
communication test
pad

Download port



serial number	Interface function	pin	illustrate
1	Power and CAN communication	1	Power supply positive (+)
		2	Negative pole of power supply (-)
		3	CAN communication low side CAN_L
		4	CAN communication high side CAN_H
2	CAN communication test point	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 Drive indicator definitions



Indicator light definition	
Power Indicator (red light when on)	Power indicator light, used to indicate MCU 3.3V power supply status, total power input 24V When the indicator light is red, it proves that the power supply of the entire network is normal; if the input power is 24V, the indicator light does not light up and the power supply needs to be cut off immediately;
signal light (Blue light when on)	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 usage instructions (scan the QR code at the end of the paper manual to obtain the debugger)

3.1 Hardware Configuration

The joint motor uses CAN communication. There are two communication lines, which are connected to the debugger through a 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 we need to use our recommended serial port tool for debugging. If you want to port it to other debugger platforms, you can refer to Chapter 3 of the manual. development.

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

3.2 Debugger interface and instructions

mainly include:

A. Module selection

- device module
- Configuration module
- Analysis module
- help module

B. Submodule selection
Equipment modules include

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

Configuration modules include:

- Parameter table, you can view and modify motor parameters
- Upload parameters, you can upload the parameters in the motor 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 the local
- Factory reset, you can restore the data in the parameter table to factory settings.
- Clear warning, you can clear motor errors, such as excessive temperature, etc.

Analysis modules include:

- Oscilloscope to view 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

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 debugging area

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

F. Submodule display area

3.3 Motor settings

3.3.1 Motor connection settings

Connect the can to USB tool (install ch340 driver, work 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 braiding calibration, reinstalling the motor board and motor, or reconnecting the motor wires in order, etc. require re-magnetic braiding calibration.
（3） Set the zero position (lost in case of power failure) and set the current position to 0.
（4） Motor program upgrade. When the motor program is updated, click the upgrade button to select the upgrade file to upgrade.

3.3.3 Parameters Table

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

Note: Please do not change the motor's torque limit, protection temperature, and over-temperature time at will. Our company will not bear any legal responsibility for any injury to the human body or irreversible damage to joints due to illegal operation of this product.

Parameters Table							
function code	Name	Parameter Type	Attributes	maximum value	minimum value	Current value (for reference)	Remark
0X0000	Name	String	read/write			ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ	
0X0001	BarCode	String	read/write			ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ	
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			7b844b0fM	

			only				
0X1005	AppBuildDate	String	read only			Apr 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	10000	1	500	
0X2005	MechOffset	float	set up	7	-7	4.619583	Motor magnetic encoder angle offset
0X2006	MechPos_heat	float	read/write	50	-50	4.52	Reference angle for initial multi-turn
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 0
0X2009	motor_index	uint8	set up	20	0	1	Motor index, marks 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	100000	0	0	can timeout threshold, default 0
0X200d	engineOverTemp	int16	read/write	1500	0	800	Motor protection temperature value, temp (degree) *10
0X200e	overTempTime	uint32	read/write	1000000	1000	20000	Overtemperature time
0X200f	GearRatio	float	read/write	64	1	7.75	Transmission ratio
0X2010	Tq_caliType	uint8	read/write	1	0	1	Torque calibration method setting
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	electric current ki
0X2014	spd_kp	float	read/write	200	0	2	Speed kp
0X2015	spd_ki	float	read/write	200	0	0.021	speed ki

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	Location mode speed limit
0X2019	limit_cur	float	read/write	23	0	23	Position, speed mode current limit

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	engine temp	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 sample value 1
0X300b	adc2Raw	uint16	read only			1212	adc sample value 2
0X300c	VBUS	float	read only			24.195	Bus voltage V
0X300d	cmdId	float	read only			0	id ring command, A
0X300e	cmdIq	float	read only			0	iq ring command, 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 angle command
0X3013	cmdVel	float	read only			0	mit protocol speed command
0X3014	rotation	int16	read only			1	Number of turns

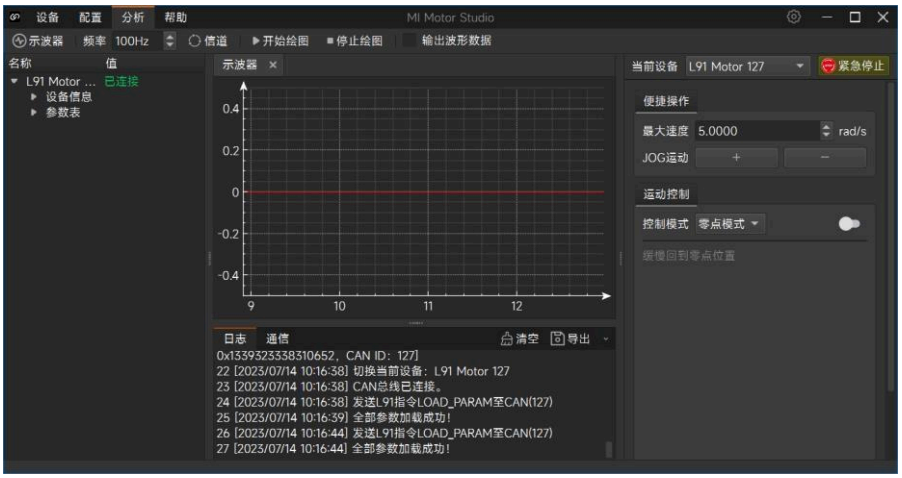
0X3015	modPos	float	read only			4.363409	Mechanical angle of motor without counting laps, rad
0X3016	mechPos	float	read only			0.777679	Mechanical angle of load end lap counting, rad
0X3017	mechVel	float	read only			0.036618	Load end speed, rad/s
0X3018	elecPos	float	read only			4.714761	electrical angle
0X3019	it	float	read only			0	U line current, A
0X301a	a	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	Board temperature, *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, degrees
0X3026	Uq	float	read only			0	q-axis voltage
0X3027	Out	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 combined 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 limits maximum current

3.3.4 Oscilloscope

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

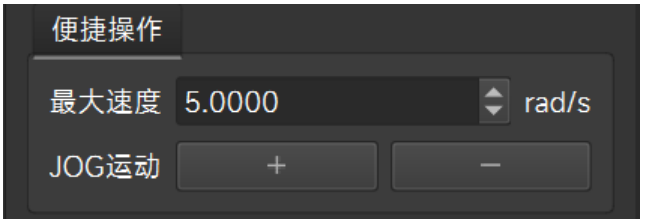
Click the oscilloscope module in the analysis module, select the appropriate parameters in the channel (for parameter meanings, please refer to 3.3.3), set the output frequency and click Start Drawing to observe the data spectrum, stop drawing to stop observing the spectrum.



3.4 control demo

jog runs:

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



directions.

Control mode switching:
The motor control mode can be converted on the motion mode interface.

Zero point mode

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

3.4.1 Operation control mode

Click the switch button on the right, then set five parameter values, click Start or Continuous Send, the motor will return to the feedback frame and run according to the target instructions; 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, and then set the Iq current command value, start or send continuously, the motor will follow 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 sine-based automatic test, and then click the switch button on the right side of the sine-based automatic test. The motor's iq (A) 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 sine automatic test, and then click the switch button on the right side of the sine automatic test. The motor speed (rad/s) will run according to the set amplitude and frequency.

3.4.4 location mode

Manually switch the position mode, click the switch button on the right, and 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 sine automatic test, and then click the switch button on the right side of the sine 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 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 will start to update the firmware. After the progress is completed, the motor update will be completed and it will restart automatically.

4 Driver communication protocol and usage instructions

Motor communication is a 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 control mode: given 5
parameters for motor operation control;
Current mode: given Iq current specified
by the motor; Speed mode: given specified
operating speed of the motor;
Position mode: Given a specified position of 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 the device ID and 64-bit MCU unique identifier

number of fixed areas	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 control mode motor control instructions (communication type 1) are used to send control instructions to the motor

data field	29-digit ID			8Byte data area
size	Bit28 ~bit24	bit23~8	bit7~0	Byte0~Byte7
describe	1	Byte2: Torque (0~65535) Corresponding (-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 (0.0~5.0)

Reply frame: Reply motor feedback frame (see communication type 2)

4.1.3 Motor feedback data (communication type 2) is used to feedback the motor operating status to the host

data field	29-digit ID			8Byte data area
size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
trace describe	2	Bit8~Bit15: current Motor CAN ID bit21~16: fault information Interest (0 no 1 yes) bit21: not calibrated bit20: HALL encoding Fault bit19: magnetic encoding reason obstacle bit18: over temperature bit17: overcurrent bit16: Undervoltage fault bit22~23: mode status state 0: Reset mode [reset] 1: Cali mode [standard 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 field	29-digit ID			8Byte data area
size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
describe	3	bit15~8: used to identify the main CAN_ID	Target motor CAN_ID	

Reply frame: Reply motor feedback frame (see communication type 2)

4.1.5 Motor stopped (communication type 4)

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

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

Reply frame: Reply motor feedback frame (see communication type 2)

4.1.6 Setting the mechanical zero position of the motor (communication type 6) will set the current motor position to the mechanical zero position (lost after power failure)

data field	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

Reply frame: Reply motor feedback frame (see communication type 2)

4.1.7 Set motor CAN_ID (communication type 7) to change the current motor CAN_ID, which will take effect immediately.

data field	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	

Reply frame: Reply motor broadcast frame (see communication type 0)

4.1.8 Single parameter reading (communication type 17)

data field	29-digit ID			8Byte data area
size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
describe	17	bit15~8: used to identify the main CAN_ID	Target motor CAN_ID	Byte0~1: index, for parameter list, see 4.1.11 Byte2~3: 00 Byte4~7: 00

Response frame:

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

4.1.9 Single parameter writing (communication type 18) (lost in case of power failure)

data field	29-digit ID			8Byte data area
size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
descr ibe	18	bit15~8: used to identify the main CAN_ID	Target motor CAN_ID	Byte0~1: index, see 4.1.11 for parameter list Byte2~3: 00 Byte4~7: parameter data

Reply frame: Reply motor feedback frame (see communication type 2)

4.1.10 Fault feedback frame (communication type 21)

data field	29-digit ID			8Byte data area
size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
descr ibe	21	bit15~8: used to identify the main CAN_ID	Motor CAN_ID	Byte0~3: fault value (not 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: under voltage fault bit1: driver chip fault bit0: motor over-temperature fault, default 80 degrees Byte4~7: warning value bit0: Motor over-temperature warning, default 75 degrees

4.1.11 Baud rate modification (communication type 22) (version 1.2.1.5 can be modified, please refer to the document process to modify it carefully. Operation errors may cause problems such as being unable to connect to the motor and being unable to upgrade)

data field	29-digit ID			8Byte data area
size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
describe	22	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

Reply frame: Reply motor broadcast frame (see communication type 0)

4.1.12 Readable and writable single parameter list (7019-7020 is readable by firmware version 1.2.1.5)

Parameter index	parameter name	describe	type	Number of bytes	Unit/Description	R/W read and write permissions
0X7005	run_mode	0: Operation control mode 1: Location mode 2: Speed mode 3: Current mode	uint8	1		W/R
0X7006	iq_ref	Current Mode Iq Command	float	4	-23~23A	W/R
0X700A	spd_ref	Speed mode speed command	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	Current Ki	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 command	float	4	rad	W/R

0X7017	limit_spd	Location mode speed limit	float	4	0~30rad/s	W/R
0X7018	limit_cur	Speed Position Mode Current Limit	float	4	0~23A	W/R
0x7019	mechPos	Load end lap counting 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	IN	R
0x701D	rotation	Number of turns	int16	2	Number of turns	W/R
0x701E	loc_kp	kp of position	float	4	Default value 30	W/R
0x701F	spd_kp	Speed in kp	float	4	Default value 1	W/R
0x7020	spd_ki	Speed of ki	float	4	Default value 0.002	W/R

4.2 Control mode usage instructions

4.2.1 Program sample

The following provides examples of controlling motors in various modes (taking gd32f303 as an example). The following calls libraries, functions 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_challenge = 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 common communication types 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. Motor control instructions in operation control mode (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 running 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 writing command (communication type 18, running mode switching)

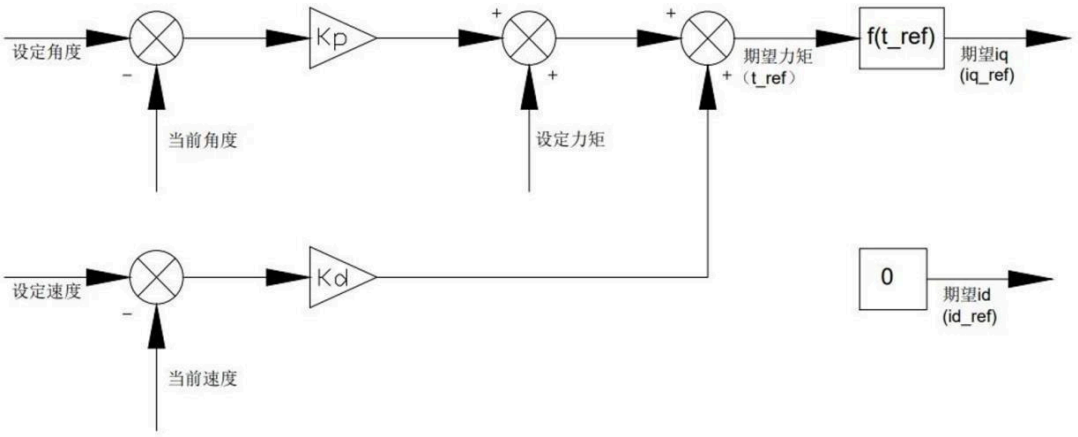
```
  uint8_t runmode;
  uint16_t index;
  void motor_modechange(uint8_t id, uint16_t master_id)
  {
    txCanIdEx.mode = 0x12;
    txCanIdEx.id = id; txCanIdEx.res =
    0; txCanIdEx.data = master_id;
    txMsg.tx_dlen = 8; for(uint8_t
    i=0;i<8;i++)
    {
      txMsg.tx_data[i]=0;
    } memcpy(&txMsg.tx_data[0],&index,2);
    memcpy(&txMsg.tx_data[4],&runmode, 1);
    can_txd();
  }
```

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

```
  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 control mode

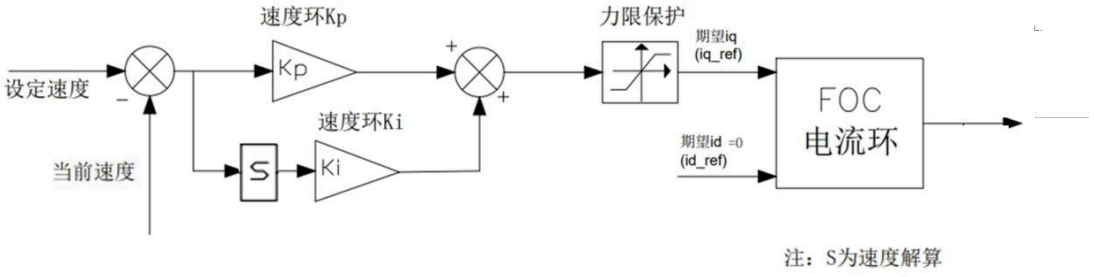


After the motor is powered on, it is in the operation control mode by default;
Send motor enable running frame (communication type 3) --> Send operation control mode motor control command (communication type 1) --> Receive motor feedback frame (communication type 2)

4.2.3 current mode

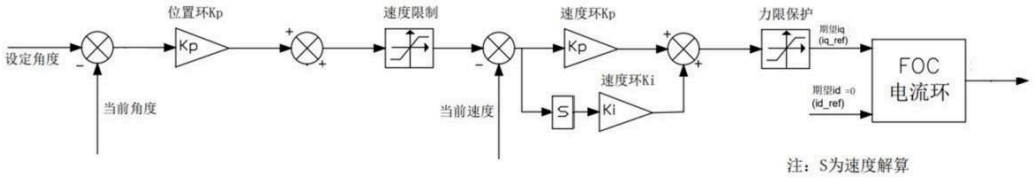
Send the motor mode parameter write command (communication type 18) to set the runmode parameter to 3 > Send
Motor enable running frame (communication type 3) --> Send motor mode parameter write command (communication type 18) to set the iq_ref parameter to the preset current command

4.2.4 speed mode



Send the motor mode parameter write command (communication type 18) to set the runmode parameter to 2 > Send
Motor enable running frame (communication type 3) --> Send motor mode parameter write command (communication type 18) Set the limit_cur parameter to the preset maximum current command --> Send motor mode parameter write command
(Communication type 18) Set the spd_ref parameter to the preset speed command

4.2.5 location mode



Send motor mode parameter write command (communication type 18) set the runmode parameter to 1 > send motor
Machine enable running frame (communication type 3) --> Send motor mode parameter write command (communication type 18) Set the limit_spd parameter to the preset maximum speed command --> Send motor mode parameter write command (communication type 18) settings The loc_ref parameter is the preset position instruction

4.2.6 stop running

Send motor stop frame (communication type 4)