CyberGear Micromotor Instruction Manual

directory (on computer hard drive)

4.2 Control Mode Instructions for Use

caveat

1. Please use it according to the working parameters specified herein, otherwise it will cause serious damage to this product!
2、During the operation of the joints, the control mode cannot be switched. If it is necessary to switch, it is necessary to send a command to stop the operation before switching.
3、Please check whether the parts are intact before use, if the parts are missing or damaged, please contact technical support in time.
4、Do not disassemble the motor at will to avoid unrecoverable failure.
5. Ensure that the motor is connected without a short circuit and that the interface is properly connected as required.

Legal Notices

Before using this product, please be sure to read this manual carefully and operate the product in accordance with the contents of this manual. If the user violates the contents of this manual to use this product, resulting in any property damage, personal injury, etc., the user shall not be liable for any damages.
We will not be responsible for any damage caused by this product. As this product consists of numerous parts, do not allow children to touch this product to avoid accidents. In order to prolong the service life of the product, please do not use the product in high temperature and high pressure environment. This manual has been printed to include as much as possible the introduction of each function and instructions for use. However, due to the continuous improvement of product functions, design changes, etc., there may still be inconsistencies with the product purchased by the user.

This manual may deviate from the actual product in terms of colour, appearance, etc. Please refer to t h e actual product. This manual is published by Xiaomi or its local subsidiaries, and Xiaomi may make typographical errors in this manual at any time.
The latest information that is incorrect or inaccurate to make necessary improvements and changes, or to make improvements to the programme and/or equipment without prior notice. Such changes will be uploaded to a new version of this manual, please scan the QR code of this manual for access. All pictures are for function description only, please refer to the actual product.

After-sales policy

After-sales service of this product is strictly based on the "Law of the People's Republic of China on the Protection of Consumer Rights and Interests" and "Law of the People's Republic of China on Product Quality" to implement after-sales service, the service content is as follows:
1. Warranty period and content

(1) Users who place an order for this product through online channels can enjoy a no-excuses return service within seven days from the next day after signing for the product. When returning the goods, the user must present a valid proof of purchase and return the invoice. Users must ensure that the returned goods maintain the original quality and functionality, appearance intact, the goods themselves and accessories, trademarks and various signs complete, if any gifts need to be returned together. If the goods are man-made damage, man-made dismantling, missing boxes, missing spare parts, will not be returned. Return logistics costs incurred by the user to bear
(See "After-sales Service Charges" for charges). If the user has not settled the logistics costs, the actual amount incurred will be deducted from the refund amount. The amount paid will be refunded to the user within seven days from the date of receipt of the returned goods. Refunds are made in the same manner as payments. The exact date of arrival may be affected by banks, payment institutions and other factors.
(2) Since the user signed within 7 days from the next day, the occurrence of non-man-made damage performance failure, confirmed by the Xiaomi after-sales service centre testing, for the user for the return of business, the return of the user must present a valid proof of purchase, and return the invoice. If there are gifts need to be returned together.

（3） From 7 days after the next day after the user signed to 15 days, the occurrence of non-man-made damage performance failure, confirmed by the Xiaomi after-sales service centre testing, for the user to handle the exchange business, replacement of the entire set of goods. After the replacement, the commodity itself recalculated the period of three packages.

（4） After 15 days to 365 days from the next day after the user's signing, after the Xiaomi after-sales service centre test confirmation, belongs to the product itself quality failure, can provide free repair services. Replacement of faulty products belongs to Xiaomi. No faulty products, will be returned in the original condition. This product after all strict testing after the factory, if not the product itself quality failure, we will have the right to refuse the user's return demand.

If the after-sales policy of this manual is inconsistent with the after-sales policy of the shop, the after-sales policy of the shop shall prevail. 2.Non-warranty regulationsThe following situations are not covered by warranty:

1. Exceeds the warranty period limited by the terms of the warranty.
2. Damage to the product caused by incorrect use without following the instructions.
3. Damage or destruction caused by improper operation, maintenance, installation, modification, testing or other improper use.
4. Routine mechanical wear and tear not caused by quality failures.
5. Damage caused by abnormal working conditions, including, but not limited to, drops, impacts, liquid immersion, violent impacts, etc.
6. Damage caused by acts of God (e.g. floods, fires, lightning strikes, earthquakes, etc.) or force majeure.
7. Damage caused by use beyond peak torque.
8. Not original genuine Xiaomi products or unable to provide legal proof of purchase.
9. Other malfunctions or damages that are not caused by the design, technology, manufacturing or quality of the product.
10. Use this product for commercial purposes.

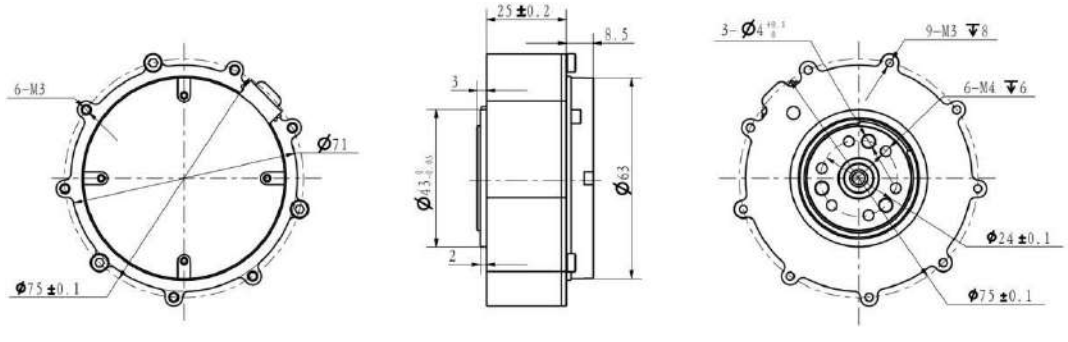
In the event of any of the above situations, the user will be responsible for paying the charges at his/her own expense.

The Group's after-sales policy is detailed at:

https://www.mi.com/service/serviceAgreement?id=17

1 Motor specification parameters

1.1 Shape and mounting dimensions



1.2 Standard state of use

1.2.1 Rated voltage: 24 VDC

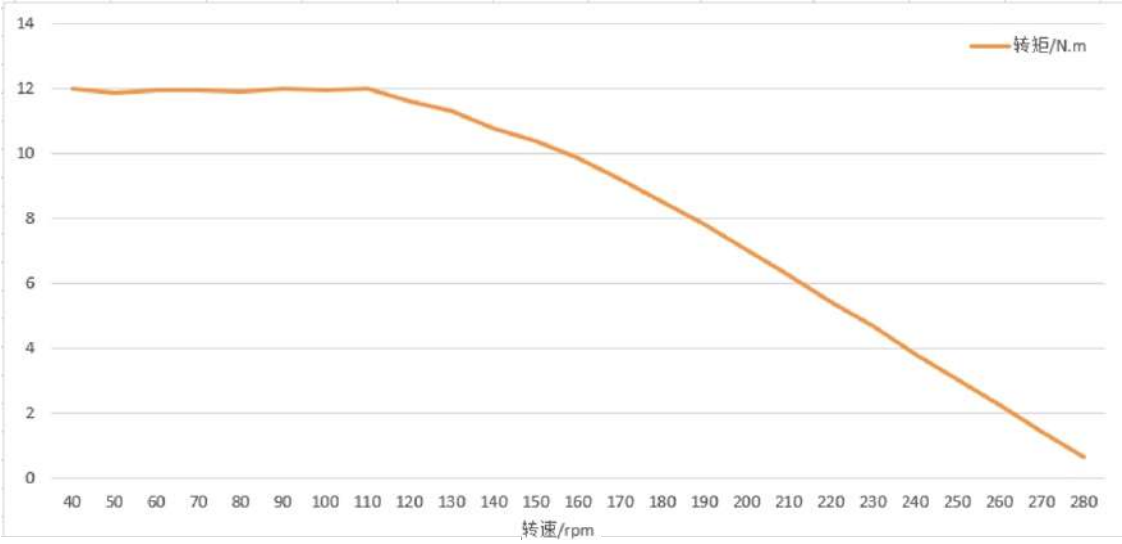
1.2.2 Operating voltage range: 16V-28 VDC

1.2.3 Rated load (CW): 4 N.m.

1.2.4 Direction of operation: CW/CCW from the direction of the outlet shaft
1.2.5 Usage position: horizontal or vertical shaft direction
1.2.6 Standard use temperature: 25±5℃
1.2.7 Use temperature range: -20～50℃
1.2.8 Standard operating humidity: 65 per cent
1.2.9 Operating humidity range: 5 to 85%, no condensation
1.2.10 Storage temperature range: -30～70℃
1.2.11 Insulation class: Class B

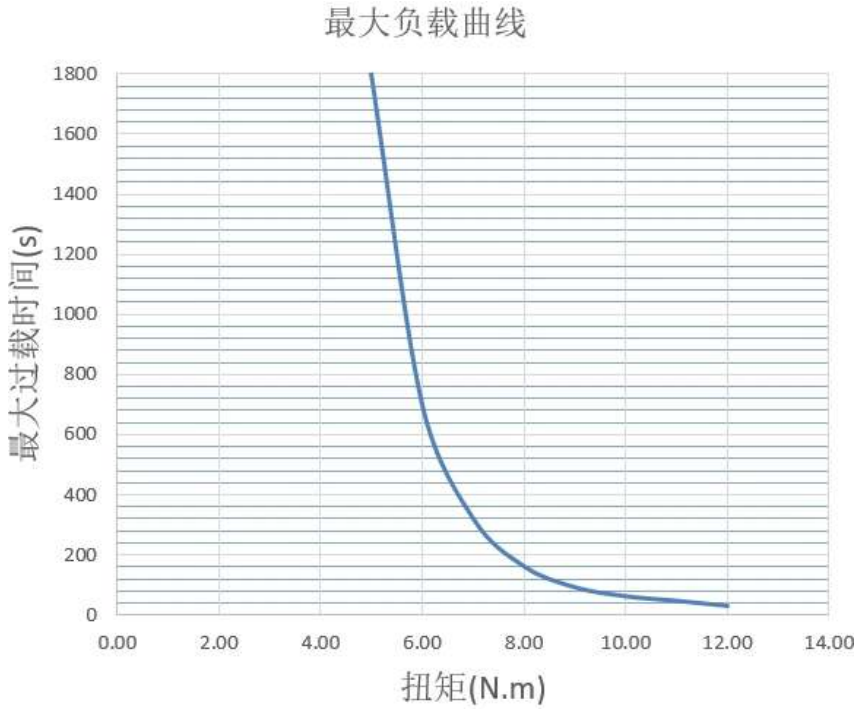
1.3 Electrical Characteristics

1.3.1 No-load speed: 296 rpm±10 per cent
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): 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 reverse potential: 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°C
Winding limit temperature: 120°C

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 property

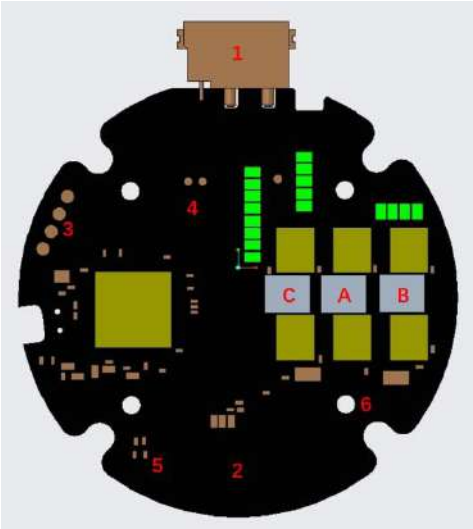
1.4.1 Weight: 317g±3g

1.4.2 Number of poles: 28

1.4.3 Number of phases: 3
1.4.4 Driving method: FOC
1.4.5 Reduction ratio: 7.75:1

2 Driver Product Information

2.1 Driver Appearance Introduction & Product Specification

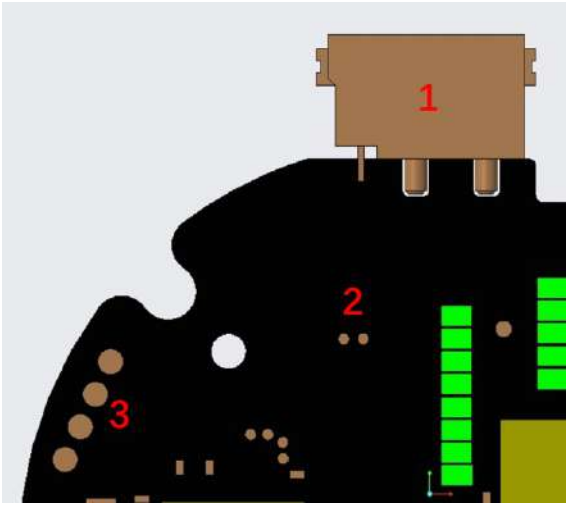


1.24V power supply and CAN communication integrated terminals;
2. Hardware version and laser engraved QR code;
3. MCU download port;
4. CAN communication test point;
5. Indicator light;
6. Mounting holes;
7. "C, A, B" are the three-phase winding welding points;

Product Specification	
Rated Working Voltage	24VDC
Maximum permissible voltage	28VDC
Rated operating current	6.5A
Maximum permissible current	23A
Standby power consumption	≤18mA
CAN Bus Bit Rate	1Mbps
sizes	Φ58mm

2.2 Drive Interface Definition

2.2.1 Drive Interface Diagram

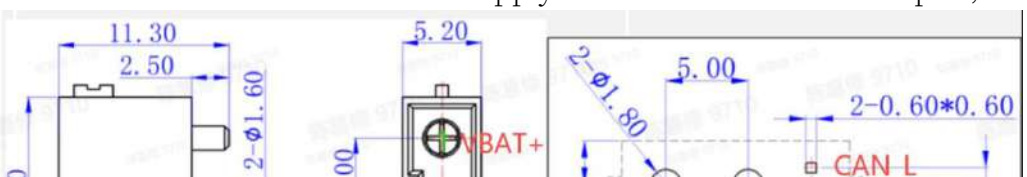


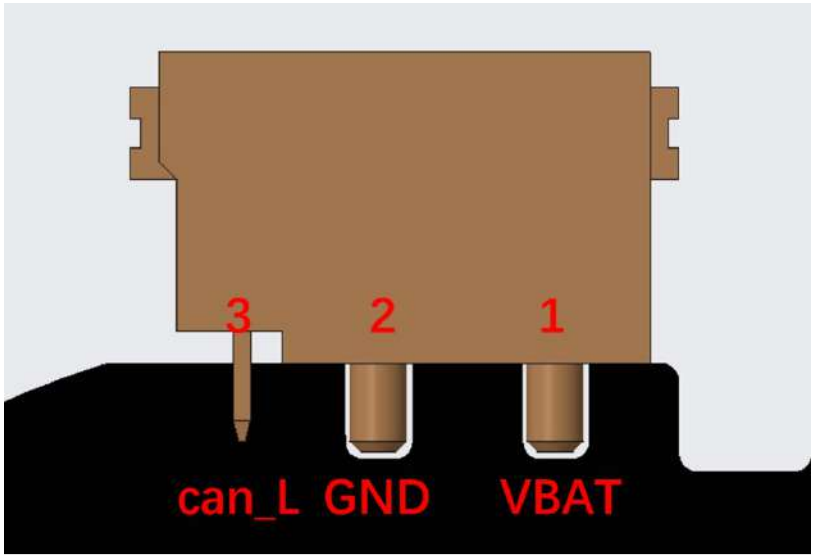
2.2.2 Recommended Brands and Models for Drive Interfaces

preface horn (wind instrument)	Board End Model	brand manufacturer	Wire End Model	brand manufacturer
1	XT30PB(2+2)-M.G.B	AMASS (S)	XT30(2+2)-F.G.B	AMASS (S)
2	2.0mm-2P Pad	/	2.0mm-2P Probe	/
3	2.54mm-4P Pad	/	2.54mm-4P Probe	/

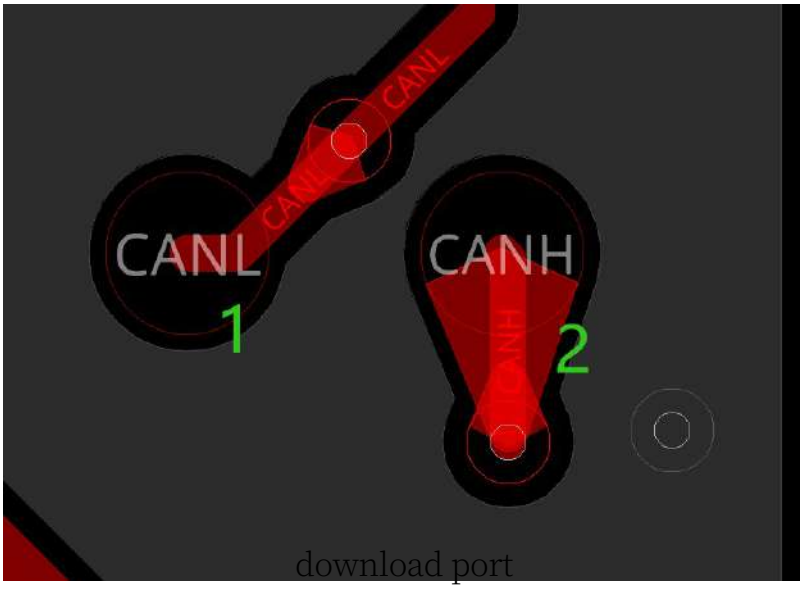
2.2.3 Driver Interface Pin Definitions

Power supply and CAN communication port;

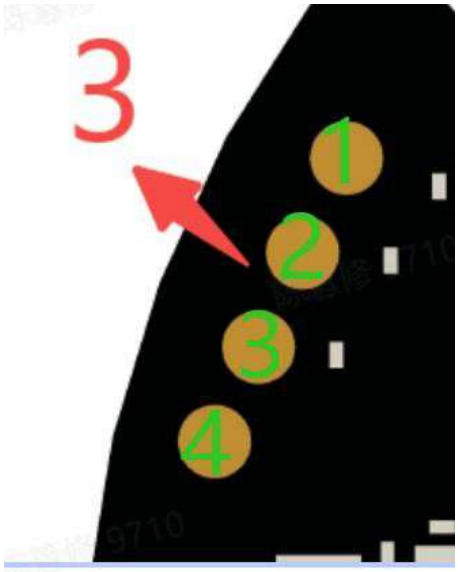




CAN Communication Test Pad

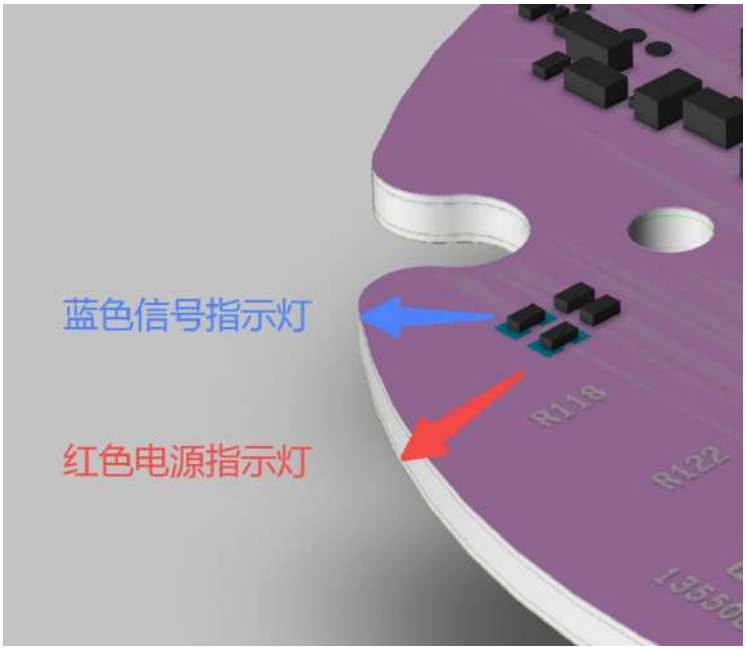


download port



serial number	interface function	pin out	clarification
1	Power supply and CAN communication	1	Power Supply Positive (+)
		2	Power supply negative (-)
		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 Drive Indicator Definition



Indicator Definition	
Power indicator （red when illuminated (Lamp)	Power indicator for MCU 3.3V power supply, total power input 24V If the lamp lights red, it proves that the power supply of the whole network is normal; if the 24V input power supply is not normal, the lamp lights red. When this indicator does not light up you need to disconnect the power supply immediately;
signal indicator （blue (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	sports event	norm	quantities
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 Instructions for using the debugger (scan the QR code at the end of the paper manual to get the debugger)

3.1 Hardware configuration

The joint motor adopts CAN communication, there are two communication lines, 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.

Note that we developed the debugger based on a specific can to USB tool, so you need to use our recommended serial port tool to debug the debugger, if you want to migrate to other debugger platforms you can refer to Chapter 3 of the manual for development.

The can-to-USB tool is recommended to use YourCee's USB-CAN module, which corresponds to the serial protocol with frame header of 41 54 and frame tail of 0D 0A.

3.2 Debugger Interface and Description

Key Inclusions:

A. Module Selection

- Device Modules
- Configuration Modules
- Analysis Module
- help module

B. Submodule Selection Device modules include

- Connecting or disconnecting motor equipment
- Motor Equipment Information
- Motor encoder calibration
- Modify motor CAN ID
- Setting the mechanical zero position of the motor
- Motor programme upgrade

Configuration modules include:

- Parameter table to view and modify motor parameters
- Parameter upload, which allows you to upload the parameters in the motor to the parameter table
- Download parameters, which allows you to download data from the parameter table to the motor
- Exporting parameters allows you to download the data in the parameter table to the local area.
- Restore factory, you can restore the data in the parameter table to the factory settings
- Clearing warnings clears motor errors such as over temperature etc.

The analysis module includes:

- Oscilloscope to view parameter curves over time
- Frequency, which allows you to adjust the frequency of viewing data
- channel, you can configure the data to be viewed
- Starting and stopping the drawing
- Output waveform data to local

Help modules include:

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

C. Motor Information Inquiry

- Equipment Information
- Parameter Sheet Information

D. data field

- Log messages
- communications information

E. debugging area

- Select Equipment
- Convenient operating area for quick control of motor forward and reverse rotation
- Motion control area, which can control the motor in various modes of operation

F. Submodule display area

3.3 Motor Settings

3.3.1 Motor connection settings

Connect the can to USB tool (install ch340 driver, default work in AT mode), select the device module, click Connect Submodule, select the corresponding serial port to connect.

3.3.2 Basic settings

（1） Modify the motor id number.
（2） Motor magnetic coding calibration, motor plate and motor re-installation, or motor wires re-ordered connection, etc., need to be re-calibrated for magnetic coding.
（3） Sets the zero position (lost at power-down) and sets the current position to 0.
（4） Motor programme upgrade, when there is an update to the motor programme, click the upgrade button to select the upgrade file to be upgraded.

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 the parameters have been loaded successfully, indicating that the relevant parameters of the motor have been successfully read (Note: the parameter table needs to be configured when the motor is in standby mode, and it is impossible to refresh the parameter table if the motor is in the running state). The interface will show the relevant parameters of the motor, and the parameters in blue colour are the stored parameters inside the motor, which can be modified in the current value column behind the corresponding parameter. The blue parameters are the internal storage parameters of the motor, which can be modified in the current value column behind the corresponding parameters. Clicking Download Parameters can download the parameters in the debugger to the motor, and clicking Upload Parameters can upload the parameters in the motor to the debugger, and the green parameters of the motor are the observation parameters, which are the parameters obtained from the collection and can be observed in real time.

Note: Please do not change the torque limit, protection temperature and over-temperature time of the motor. We will not assume any legal responsibility for any injury to the human body or irreversible damage to the joints caused by illegal operation of this product.

parameter table							
function code	name (of a thing)	Parameter type	causality	maximum values	minimum value	Current value (for reference)	note
0X0000	Name	String	Read/			yyyyyyyyyyyyyyyy	

			Write			ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ.	
0X0001	BarCode	String	Read/ Write			ÿÿÿ.	
0X1000	BootCodeVersion	String	read-only (computing)			0.1.5	
0X1001	BootBuildDate	String	read-only (computing)			Mar 16 2022	
0X1002	BootBuildTime	String	read-only (computing)			20:22:09	
0X1003	AppCodeVersion	String	read-only (computing)			0.1.5	Motor programme version number
0X1004	AppGitVersion	String	read-only (computing)			7b844b0fM	
0X1005	AppBuildDate	String	read-only (computing)			Apr 14 2022	
0X1006	AppBuildTime	String	read-only			20:30:22	

			(computing)				
0X1007	AppCodeName	String	read-only (computing)			dog_motor	
0X2000	echoPara1	uint16	configure	74	5	5	
0X2001	echoPara2	uint16	configure	74	5	5	
0X2002	echoPara3	uint16	configure	74	5	5	
0X2003	echoPara4	uint16	configure	74	5	5	
0X2004	echoFreHz	uint32	Read/Write	10000	1	500	
0X2005	MechOffset	float	references	7	-7	4.619583	Motor magnetic encoder angle degree bias
0X2006	MechPos_init	float	Read/Write	50	-50	4.52	Reference angle at initial multiturn
0X2007	limit_torque	float	Read/Write	12	0	12	Torque Limit
0X2008	I_FW_MAX	float	Read/Write	33	0	0	Weak current value, default 0
0X2009	motor_index	uint8	references	20	0	1	Motor index, marking the motor joint position
0X200a	CAN_ID	uint8	references	127	0	1	This node id
0X200b	CAN_MASTER	uint8	prefe	127	0	0	can host id

			rences				
0X200c	CAN_TIMEOUT	uint32	Read/Write	100000	0	0	can timeout threshold. Default 0
0X200d	motorOverTemp	int16	Read/Write	1500	0	800	Motor protection temperature value, temp (degrees) *10
0X200e	overTempTime	uint32	Read/Write	1000000	1000	20000	overtemperature time
0X200f	GearRatio	float	Read/Write	64	1	7.75	gear 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 Filtering 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 ki
0X2016	loc_kp	float	Read/Write	200	0	30	Position kp
0X2017	spd_filt_gain	float	Read/Write	1	0	0.1	Velocity filter parameters
0X2018	limit_spd	float	Read/Write	200	0	2	Position Mode Speed Limit regulate
0X2019	limit_cur	float	Read/	23	0	23	Position, speed mode

			Write				Current Limit
0X3000	timeUse0	uint16	read-only (computing)			5	
0X3001	timeUse1	uint16	read-only (computing)			0	
0X3002	timeUse2	uint16	read-only (computing)			10	
0X3003	timeUse3	uint16	read-only (computing)			0	
0X3004	encoderRaw	int16	read-only (computing)			11396	Magnetic Encoder Sample Value
0X3005	mcuTemp	int16	read-only (computing)			337	mcu internal temperature. *10
0X3006	motorTemp	int16	read-only (com			333	motor ntc temperature. *10

			putin g)				
0X3007	vBus(mv)	uint16	read-only (com putin g)			24195	busbar voltage
0X3008	adc1Offset	int32	read-only (com putin g)			2084	adc Sampling Channel 1 Zero current bias
0X3009	adc2Offset	int32	read-only (com putin g)			2084	adc sample channel 2 zero current bias
0X300a	adc1Raw	uint16	read-only (com putin g)			1232	adc Sampling value 1
0X300b	adc2Raw	uint16	read-only (com putin g)			1212	adc Sampling value 2
0X300c	VBUS	float	read-only (com putin g)			24.195	Busbar voltage V
0X300d	cmdId	float	read-only			0	id ring command, A

			(computing)				
0X300e	cmdIq	float	read-only (computing)			0	iq ring command, A
0X300f	cmdlocref	float	read-only (computing)			0	Position loop command, rad
0X3010	cmdspdref	float	read-only (computing)			0	Velocity loop command, rad/s
0X3011	cmdTorque	float	read-only (computing)			0	Torque command, nm
0X3012	cmdPos	float	read-only (computing)			0	mit Protocol Angle Command
0X3013	cmdVel	float	read-only (computing)			0	mit protocol speed indicator virtuous
0X3014	rotation	int16	read-			1	number of laps

			only (computing)				
0X3015	modPos	float	read-only (computing)			4.363409	Motor uncalculated mechanical angle, rad
0X3016	mechPos	float	read-only (computing)			0.777679	Load-side Loop Counting Machinery Angle, rad
0X3017	mechVel	float	read-only (computing)			0.036618	load-side steering Speed, rad/s
0X3018	elecPos	float	read-only (computing)			4.714761	Electrical angle
0X3019	ia	float	read-only (computing)			0	U line current, A
0X301a	ib	float	read-only (computin			0	V line current, A

			g)				
0X301b	ic	float	read-only (computing)			0	W line current, A
0X301c	tick	uint32	read-only (computing)			31600	
0X301d	phaseOrder	uint8	read-only (computing)			0	Calibration direction markers
0X301e	iqf	float	read-only (computing)			0	iq Filter value, A
0X301f	boardTemp	int16	read-only (computing)			359	Temperature on board, *10
0X3020	iq	float	read-only (computing)			0	iq original value, A
0X3021	id	float	read-only (computing)			0	id Original value, A
0X3022	faultSta	uint32	read-only (computin			0	Fault status value

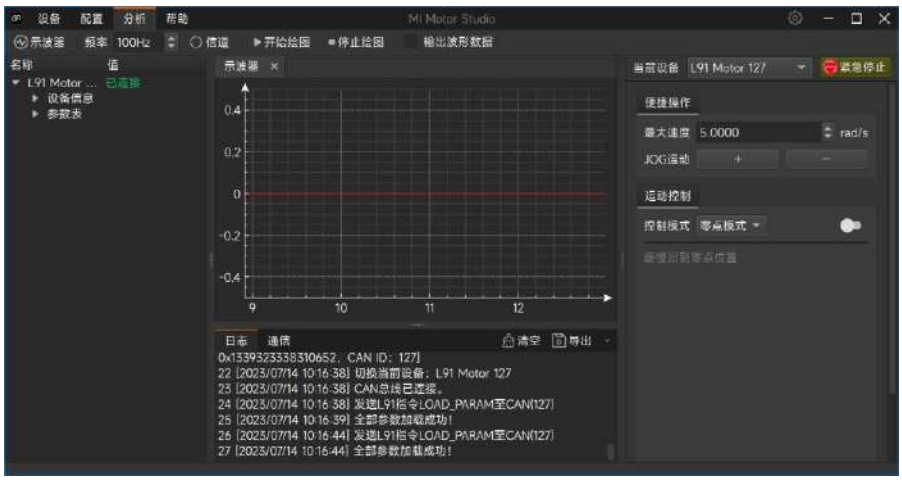
			g)				
0X3023	warnSta	uint32	read-only (computing)			0	Warning status value
0X3024	drv_fault	uint16	read-only (computing)			0	Driver Chip Fault Value
0X3025	drv_temp	int16	read-only (computing)			48	Driver Chip Temperature value, degree
0X3026	Uq	float	read-only (computing)			0	q Axis voltage
0X3027	Ud	float	read-only (computing)			0	d Axis voltage
0X3028	dtc_u	float	read-only (computing)			0	U-phase output duty cycle
0X3029	dtc_v	float	read-only (computing)			0	V Phase Output Duty Cycle
0X302a	dtc_w	float	read-only			0	W Phase Output Duty Cycle

			(computing)				
0X302b	v_bus	float	read-only (computing)			24.195	vbus in closed loop
0X302c	v_ref	float	read-only (computing)			0	Closed-loop vq,vd synthesis input voltage
0X302d	torque_fdb	float	read-only (computing)			0	Torque feedback value, nm
0X302e	rated_i	float	read-only (computing)			8	Motor rated current
0X302f	limit_i	float	read-only (computing)			27	Motor Limit Maximum Electricity stream of water or sth. resembling one

3.3.4 oscillograph

The interface supports the viewing of plots generated by observing real-time data such as motor Id/Iq current, temperature, real-time speed at the outputs, rotor (encoder) position, position at the outputs, etc.

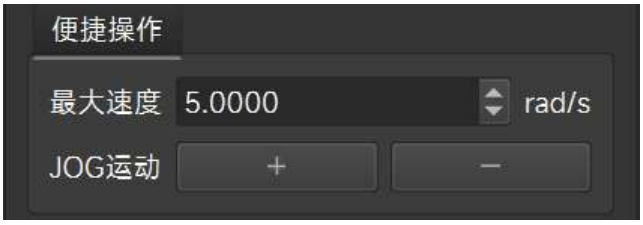
Click on the oscilloscope module in the analysis module, select the appropriate parameters in the channel (refer to 3.3.3 for the meaning of the parameters), set the output frequency and click on Start Plotting to observe the data pattern, and Stop Plotting to stop observing the pattern.



3.4Control Demo

jog runs:

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



Control mode switching:

Switching between motor control modes is possible in the Motion Mode screen.

3.4.1 zero-point mode

The motor will slowly return to the mechanical zero position by clicking on the right switch button.

3.4.2 operational control model

Click the right switch button, then set the five parameter values, click start or send continuously, the motor will return the feedback frame and run according to the target instruction; click the right switch button again, the motor will stop.

3.4.2 current mode

To switch current mode manually, click the right switch button, then set the Iq current command value to start or send continuously, the motor will run following the current command, click the right switch button again, the motor will stop.

Click the control mode right switch button, input the amplitude and frequency of sinusoidal auto test, then click the sinusoidal auto test right switch button, the iq(A) of the motor will run according to the set amplitude and frequency.

3.4.3 Speed Mode

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

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

3.4.4 Position Mode

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

Click the control mode right switch button, input the amplitude and frequency of sinusoidal auto test, then click the sinusoidal auto test right switch button, the motor position (rad) will run according to the set amplitude and frequency.

3.5 Firmware Updates

The first step is to click upgrade of 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 reboot automatically.

4 Driver communication protocols and instructions for use

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

data domain	29-bit ID			8Byte Data Area
magnitude	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7

The control modes supported by the motor include:
Operation and control mode: 5 parameters of operation and control for a given motor; current mode: specified Iq current for a given motor; speed mode: specified running speed for a given motor;
Position Mode: Given a specified position of the motor, the motor will run to that specified position;

4.1 Description of communication protocol types

4.1.1 Get device ID (communication type 0); get device ID and 64-bit MCU unique identifier

Data domain (taxonomy)	29-bit ID			8Byte Data Area
magnitude	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
descriptions	0	bit15~8: Used to identify the host machine CAN_ID	Programme objective Electricity Machine CAN_ID	0

Answer frame:

data domain	29-bit ID			8Byte Data Area
magnitude	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
descriptions	0	Target Motor CAN_ID	0XFE	64-bit MCU Unique Identifier

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

number domain (taxonomy)	29-bit ID	8Byte Data Area

oldest few	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
description	1	Byte2:Torque (0~65535) Correspondence (-12Nm~12Nm)	Target motor CAN _ID	Byte0~1: Target angle [0~65535] corresponds to (-4π~4π) Byte2~3: Target angular velocity [0~65535] corresponding to (-) 30rad/s~30rad/s) Byte4~5: Kp [0~65535] corresponds to (0.0~500.0) Byte6~7: Kd [0~65535] correspondence (0.0~5.0)

Answer frame: Answers the motor feedback frame (see communication type 2).

4.1.3 Motor feedback data (communication type 2) Used to provide feedback to the host computer on the motor's operating status.

numbers domain (taxonomy)	29-bit ID			8Byte Data Area
oldestfew	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
descriptions	2	Bit8~Bit15: Current motor CAN ID bit21~16: Fault message (0 no 1 yes) bit21: Uncalibrated bit20: HALL coded faults bit19: Magnetic code failure bit18: Over-temperature bit17: Over-current bit16: Under-voltage fault bit22~23: Mode Status 0： Reset Mode [Reset] 1：Cali mode [calibration] 2：Motor mode [Running]	Host CAN_ID	Byte0~1: Current angle [0~65535] corresponding to (-4π~4π) Byte2~3: Current angular velocity [0~65535] corresponds to (-30rad/s~30rad/s) Byte4~5:Current torque [0~65535] Corresponding to (-12Nm~12Nm) Byte6~7:Current temperature: Temp(degree Celsius)*10

4.1.4 Motor enable operation (communication type 3)

numbers domain (taxonomy)	29-bit ID			8Byte data quarter
magnitude	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
descriptions	3	bit15~8: used to identify the main CAN_ID	Target motor CAN_ID	

Answer frame: Answers the motor feedback frame (see communication type 2).

4.1.5 Motor stop (communication type 4)

numbers domain (taxonomy)	29-bit ID			8Byte Data Area
magnitude	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
descriptions	4	bit15~8: Used to identify the main CAN_ID.	Target Motor CAN_ID	During normal operation, the data area should be cleared; When Byte[0]=1: Clear fault;

Answer frame: Answers the motor feedback frame (see communication type 2).

4.1.6 Setting the motor mechanical zero position (communication type 6) sets the current motor position to the mechanical zero position (lost at power down)

numbers domain (taxonomy)	29-bit ID			8Byte Data Area
magnitude	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
descriptions	6	bit15~8: Used to identify the main CAN_ID.	Target Motor CAN_ID	Byte[0]=1

Answer frame: Answers the motor feedback frame (see communication type 2).

4.1.7 Setting the motor CAN_ID (communication type 7) Changes the current motor CAN_ID with immediate effect.

numbers domain (taxonomy)	29-bit ID			8Byte Data Area
magnitude	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
descriptions	7	bit15~8: Used to identify the main CAN_ID. Bit16~23: Pre-set CAN_ID	Target Motor CAN_ID	

Answer frame: Answers the motor broadcast frame (see communication type 0).

4.1.8 Single parameter reading (communication type 17)

numbers domain (taxonomy)	29-bit ID			8Byte Data Area
magnitu	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7

de				
description	17	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: 00

Answer frame:

data domain	29-bit ID			8Byte Data Area
magnitude	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
descriptions	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 Individual parameter write (communication type 18) (power-down loss)

numbers domain (taxonomy)	29-bit ID			8Byte Data Area
magnitude	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
descriptions	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

Answer frame: Answers the motor feedback frame (see communication type 2).

4.1.10 Fault feedback frame (communication type 21)

numbers domai	29-bit ID	8Byte Data Area

n (taxonomy)				
magnitude	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
descriptions	21	bit15~8: Used to identify the main CAN_ID.	Motor CAN_ID	Byte0~3: fault value (not 0: faulty, 0: normal) bit16:A phase current sampling overcurrent bit15~bit8:Overload fault bit7:Encoder not calibrated bit5:C phase current sampling overcurrent bit4:B phase current sampling overcurrent bit3:Overvoltage fault bit2:undervoltage fault bit1:driver chip fault Byte4~7: warning values bit0: motor over-temperature warning, default is 75 degrees.

4.1.11 Baud rate modification (communication type 22) (available in version 1.2.1.5, please refer to the document)

(The file process should be modified carefully, operation errors will not be able to connect the motor, can not be upgraded, etc.)

data domain	29-bit ID			8Byte Data Area
magnitude	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
descriptions	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

Answer frame: Answers the motor broadcast frame (see communication type 0).

4.1.12 Individual parameter lists can be read/written (7019-7020 are firmware version 1.2.1.5 readable).

Parameter index	**Parameter name**	**descriptions**	**typology**	**byte count**	**Unit/Description**	**R/W Read/Write Rights set a limit (on)**
0X7005	run_mode	0: Operational control mode	uint8	1		W/R

		1: Position Mode 2: Speed Mode 3: Current Mode				
0X7006	iq_ref	Current Mode Iq Command	float	4	-23~23A	W/R
0X700A	spd_ref	RPM Mode RPM directives	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 the 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 directives	float	4	rad	W/R
0X7017	limit_spd	Position Mode Speed constraint	float	4	0~30rad/s	W/R
0X7018	limit_cur	Velocity Position Mode Current Limit	float	4	0~23A	W/R
0x7019	mechPos	Load-side Loop Counting Machine mechanical angle	float	4	rad	R
0x701A	iqf	iq Filter value	float	4	-23~23A	R
0x701B	mechVel	Load-side	float	4	-30~30rad/s	R

		speed				
0x701C	VBUS	busbar voltage	float	4	V	R
0x701D	rotation	number of laps	int16	2	number of laps	W/R
0x701E	loc_kp	kp of position	float	4	Default value 30	W/R
0x701F	spd_kp	kp of speed	float	4	Default value 1	W/R
0x7020	spd_ki	ki of the velocity	float	4	Default value 0.002	W/R

4.2 Control Mode Instructions for Use

4.2.1 sample program

The following are examples of various modes of motor control (gd32f303 as an example) The following are examples of libraries, functions and macro definitions.

```
#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)
```

Below is a list of common communication types sent:

1. Motor enable run 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 control mode motor control commands (communication type 1)

```
    void motor_controlmode(uint8_t id, float torque, float
    MechPosition, float speed, float kp, float k d )
    {
      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, operation mode switching)

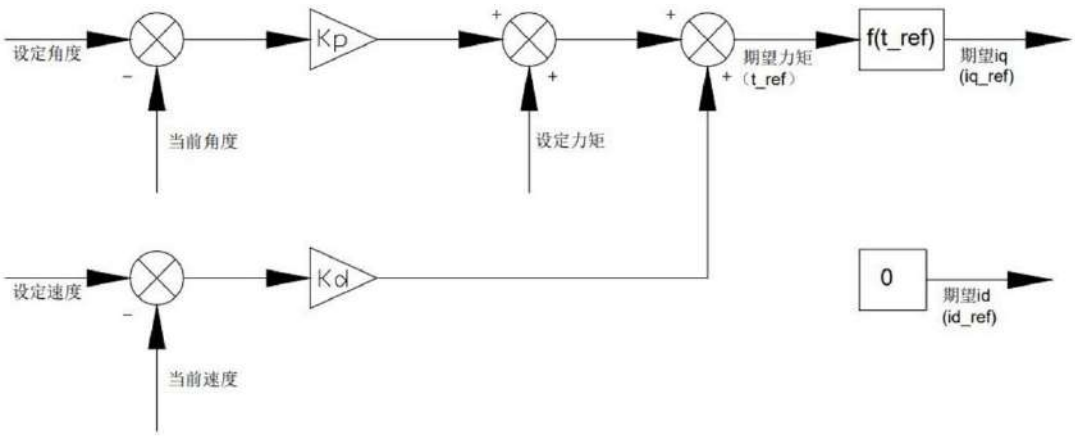
```
  uint8_t runmode;
  uint16_t index;
  void motor_modechange(uint8_t id, uint16_t master_id)
  {
    txCanIdEx.mode = 0x12;
    txCanIdEx.id = id;
    txCanIdEx.res = 0;
    txCanIdEx.data = master_id;
    txMsg.tx_dlen = 8;
    for(uint8_t i=0;i<8;i++)
    {
      txMsg.tx_data[i]=0;
    }
    memcpy(&txMsg.tx_data[0],&index,2);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

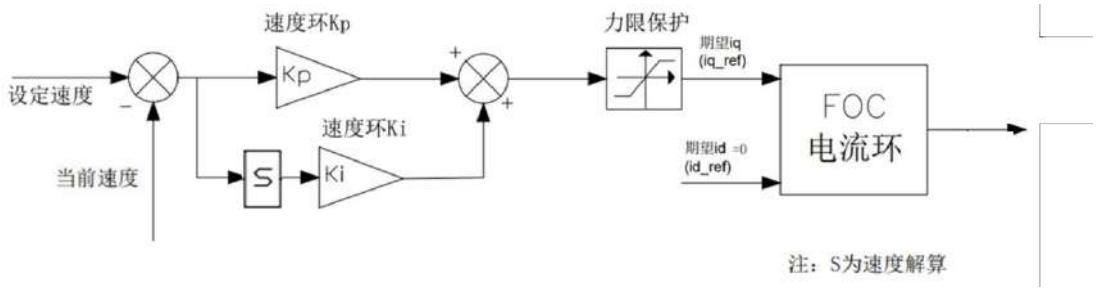


The motor is in Run Control mode by default after power up;
Send motor enable run frame (communication type 3) --> Send operation control mode motor control command (communication type 1) --> Receive motor feedback frame (communication type 2)

4.2.3 current mode

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

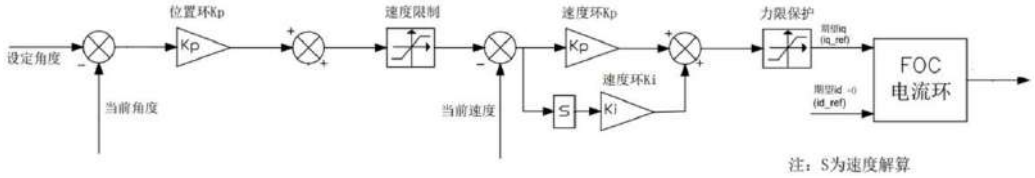
4.2.4 Speed Mode



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** parameter to preset max. current command ---> Send motor mode parameter write command

(Communication type 18) Setting the **spd_ref** parameter to 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** parameter to preset maximum speed command --> Send motor mode parameter write command (communication type 18) Set the **loc_ref** parameter to the preset position command.

4.2.6 stop running

Send motor stop frame (communication type 4)