iSVD78C-48V-230-A14-002

User Manual

Versions：V1.00

[Please read this manual carefully before use to avoid damaging the driver]

iSVD78C-48V-230-A14-002-EN

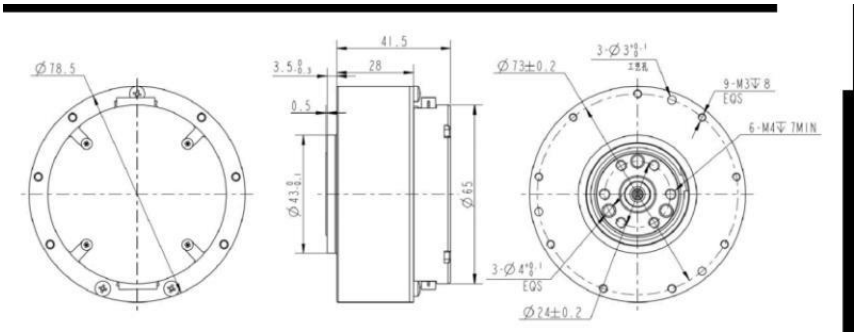
iSVD78C-48V-230-A14-002 instruction manual

Precautions

1. Please use according to the working parameters specified in this article, otherwise it may cause serious damage to the product!
2. Do not switch the control mode when the joint is running. If you need to switch, send the command to stop the operation before switching.
3. Check whether the parts are in good condition before use. If the parts are missing or damaged, contact technical support in time.
4. Do not disassemble the motor at will, so as to avoid unrecoverable failure.
5. Ensure that there is no short circuit when the motor is connected, and the interface is correctly connected as required.

Motor specification

Outline and mounting dimensions



When fixing, the screw depth should not exceed the depth of the casing thread

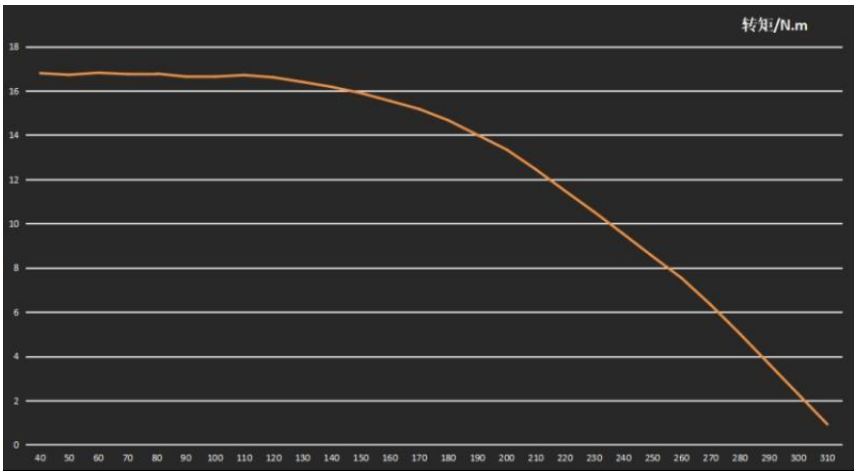
Standard usage status

1. Rated voltage: 48 VDC
2. Operating voltage range: 24V-60 VDC
3. Rated load (CW) : 6 N.m
4. Operation direction: CW/CCW from the direction of the exit shaft
5. Use posture: the direction of the exit axis is horizontal or vertical
6. Standard operating temperature: 25±5℃
7. Operating temperature range: -20 ~ 50℃
8. Standard operating humidity: 65%
9. Humidity range: 5 ~ 85%, no condensation
10. Storage temperature range: -30 ~ 70℃
11. Insulation Class: Class B

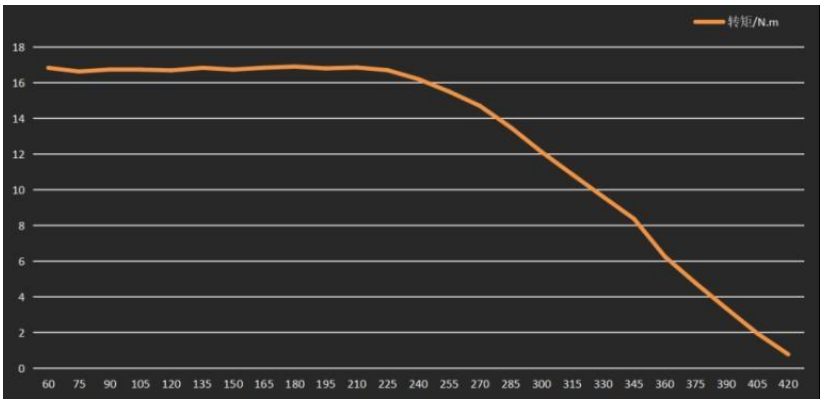
Electrical characteristics

1. No load speed: 410 rpm±10%
2. No-load current: 0.5 Arms
3. Rated load: 6 N.m
4. Rated load speed: 360rpm±10%
5. Rated load phase current (peak) : 7Apk±10%
6. Peak load: 17 N.m
7. Maximum load phase current (peak) : 23Apk±10%

8. Insulation resistance/stator winding: DC 500VAC, 100M Ohms
9. High voltage/stator and housing: 600 VAC, 1s, 2mA
10. Motor back potential: 0.096Vrms/rpm ±10%
11. Torque constant: 1.22N.m/Arms
12. T-N curve (36V)



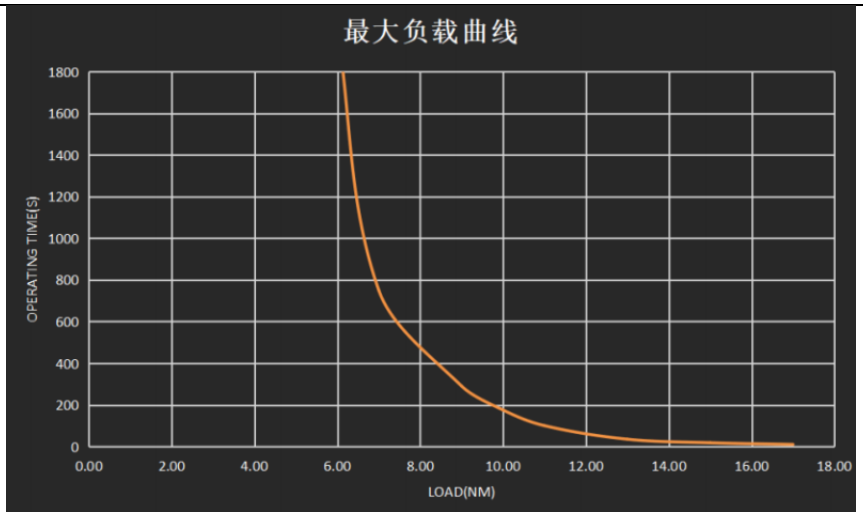
13. T-N curve （48V）



14. Maximum overload curve

Test conditions: Ambient temperature: 25℃

Winding limit temperature: 135℃ (this is the constraint temperature, the actual is 180 degrees) Speed: 24rpm



Test data

Load	Operating time(s)
17.00	10
15.00	18
13.00	35
11.00	100
9.00	370
7.00	1000
6.50	3000
6.00	rated

Mechanical properties

1. Weight: 380g±3g
2. Number of poles: 28
3. Phase number: 3 phases
4. Drive mode: FOC
5. Deceleration ratio: 7.75:1

Drive product information

Drive product specifications

project	data
The rated working voltage	48VDC
The maximum allowable voltage	60VDC
Rated working phase current	7Apk
Maximum allowable phase current	23Apk
Standby power	≤18mA
CAN bus bit rate	1Mbps
Dimensions	Φ58mm
Working environment temperature	-20℃ to 50℃
The maximum allowable temperature of the control board	105℃
encoder resolution	14bit (absolute turn)

Driver interface definition

Driver interface recommended brand and model

board end model	brand manufacturer	line end model	brand manufacturer
XT30PB(2+2)-M.G.B	AMASS (Ams)	XT30(2+2)-F.G.B	AMASS (Ams)

Driver function pin and device description

1. Power supply and CAN communication

Pin	description
1	The positive electrode of the power supply (+)
2	Negative electrode of the power supply (-)
3	CAN CAN_L
4	CAN the high side of the communication CAN_H

1. Download port

Pin	description
1	SWDIO (data)
2	SWCLK (clock)
3	3V3 (positive 3.3V)
4	GND

1. Indicator light

Pin	description
1	If the blue indicator blinks, the program is running normally
2	Power indicator. If the indicator is red, the power supply to the entire network is normal

Main devices and specifications

No.	Item	Specifications	quantity
1	MCU chip	GD32F303RGT6	1 PCS
2	Driver chip	DRV8353-SRTAR	1 PCS
3	magnetic encoder chip	AS5047P	2 PCS
4	thermistor	LTS00-104J395t19e010 / NCP18XH103F03RB	2 PCS
5	Power MOS	JMGG031V06A	6 PCS

Upper computer instructions

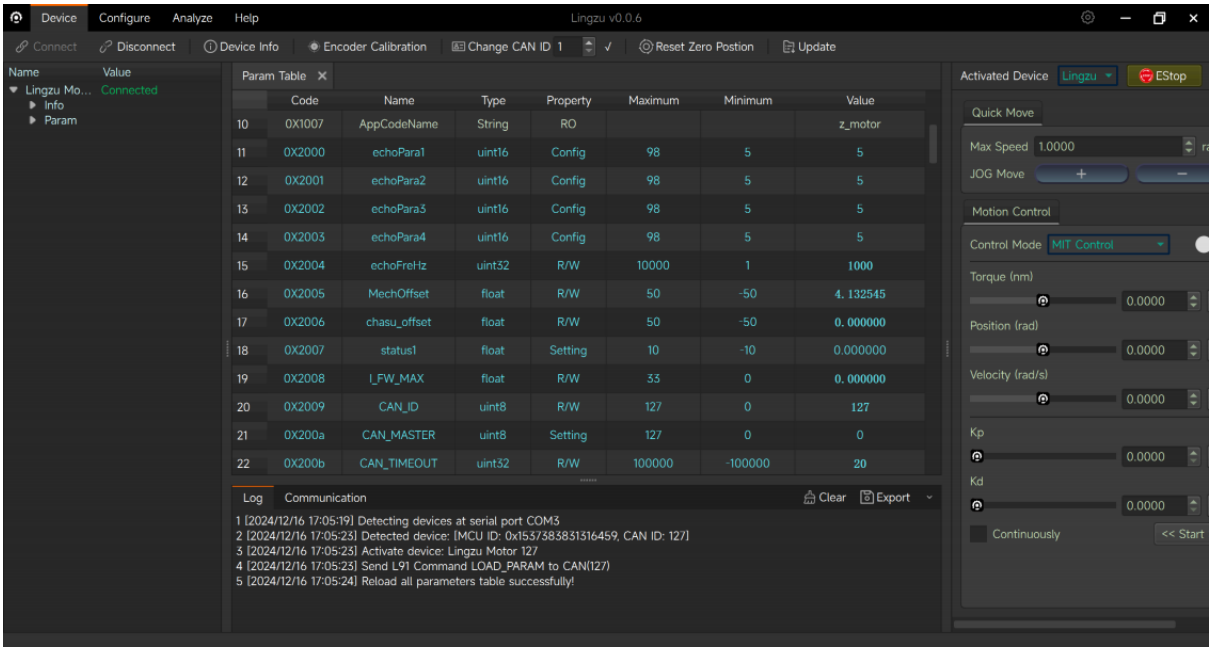
Hardware disposition

The articulated motor uses the CAN communication mode and has two communication cables. It is connected to the debugger through the can to USB tool. The debugger needs to be installed with the ch340 driver in advance and works in AT mode by default.

It should be noted that we are based on the specific can to USB tool development of the debugger, so we need to use our recommended serial port tool to debug the debugger, if you want to transplant to other debugger platform can refer to the third chapter of the instructions for development.

The frame header of the corresponding serial port protocol is 41 54, and the frame tail is 0D 0A.

Upper computer interface and description



It mainly includes:

A. Select a module •

Device module

· disposition module ·

Analysis module

· Help Module

B. Select a submodule

· Connect or disconnect motor equipment ·

Motor equipment information

· Motor encoder calibration ·

Modify the motor CAN ID

· Set the mechanical zero position of the motor ·

Motor program upgrade

Parameter table, you can view and modify the motor parameters

· Upload parameters. The parameters in the motor can be uploaded to the parameter table

· Download parameters, you can download the data in the parameter table to the motor ·

Export parameters. You can download data in the parameter table to a local computer · Restore the data in the parameter table to factory defaults

· Clear warning, can clear motor errors, such as high temperature

Analysis modules include:

· Oscilloscope, you can view the curve of parameter change with time ·

Frequency: You can adjust the frequency of viewing data

· The channel can be disposition to view the data ·

Start and stop drawing

· Output waveform data locally

Help modules include:

· Instructions, you can open the instruction manual ·

Yes, you can check the software information

C. Motor information query ·

Device information

· Parameter table information D.

Data field

· Log information

· Communication information E.

Run the debugging area

· Select equipment

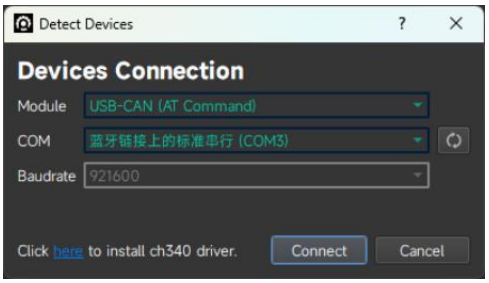
· Convenient operation area, can quickly control the positive and negative rotation of the motor ·

Motion control area, which can control the motor operation according to various modes

F. Submodule display area

Motor setup

Motor connection setup



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

Basic setup

1. Change the motor id.
2. Motor magnetic coding calibration, motor board and motor re-installation, or motor three-phase line re-sequential connection, need to be re-calibrated magnetic coding.
3. Set the zero position to 0.
4. Motor program upgrade, when the motor program is updated, click the upgrade button to select the upgrade file to upgrade.

Parameter list

After the motor is successfully connected, click the parameter table module in disposition module. The log will show that all parameters are loaded successfully, indicating that the relevant parameters of the motor are successfully read (Note: The parameter table is required for disposition under the standby state of the motor. If the motor is in the running state, the parameter table cannot be refreshed), the interface will display the relevant parameters of the motor. The parameters in blue are the stored parameters in the motor, which can be modified in the current value bar after the corresponding parameters. Click to download parameters to download the parameters in the debugger to the motor, click to upload parameters to upload the parameters in the motor to the debugger, and the green parameters of the motor are observed parameters, which are collected parameters and can be observed in real time.

Note: Please do not change the torque limit, protection temperature and overtemperature time of the motor. Our company will not bear any legal responsibility for any damage to human body or irreversible damage to joints caused by illegal operation of this product.

function code	name	parameter type	attribute	Maximum value	Minimum value	Current value (for reference)	备注
0X0000	Name	String	Read/Write			[redacted]	
0X0001	BarCode	String	Read/Write			[redacted]	
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			0.0.0.1	Motor program version numbe
0X1004	AppGitVersion	String	Read only			7b844b0fM	
0X1005	AppBuildDate	String	Read only			Apr 14 2022	
0X1006	AppBuildTime	String	Read only			20:30:22	
0X1007	AppCodeName	String	Read only			motor	
0X2000	echoPara1	uint16	disposition	74	5	5	
0X2001	echoPara2	uint16	disposition	74	5	5	
0X2002	echoPara3	uint16	disposition	74	5	5	
0X2003	echoPara4	uint16	disposition	74	5	5	
0X2004	echoFreHz	uint32	Read/Write	10000	1	500	
0X2005	MechOffset	float	Settings	7	-7	4.619583	Motor magneti encoder Angle offset
0X2006	status2	float	Read/Write	50	-50	4.52	Reserved parameter
0X2007	limit_torque	float	Read/Write	17	0	17	Torque limitati
0X2008	I_FW_MAX	float	Read/Write	33	0	0	Weak magneti current value, default 0
0X2009	motor_baud	uint8	Settings	20	0	1	Baud rate flag bit

0X200a	CAN_ID	uint8	Settings	127	0	1	id of this objec
0X200b	CAN_MASTER	uint8	Settings	127	0	0	can host id
0X200c	CAN_TIMEOUT	uint32	Read/Write	100000	0	0	can timeout threshold. The default value i 0
0X200d	status2	int16	Read/Write	1500	0	800	Reserved parameter
0X200e	status3	uint32	Read/Write	1000000	1000	20000	Reserved parameter
0X200f	status1	float	Read/Write	64	1	7.75	Reserved parameter
0X2010	Status6	uint8	Read/Write	1	0	1	Reserved parameter
0X2011	cur_filt_gain	float	Read/Write	1	0	0.9	Current filterin parameter
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	Velocity 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 parameter
0X2018	limit_spd	float	Read/Write	200	0	2	Location mode speed limit
0X2019	limit_cur	float	Read/Write	23	0	23	Position, Velocity mode current limit
0X201a	loc_ref_filt_gain	float	Read/Write	100	0	0	Reserved parameter
0X201b	limit_loc	float	Read/Write	100	0	0	Reserved parameter
0X201c	position_offset	float	Read/Write	27	0	0	High speed segment offse
0X201d	chasu_angle_offset	float	Read/Write	27	0	0	The low end is offset
0X201e	zero_sta	float	Read/Write	150	0		Zero marker
0x201f	protocol_1	float	Read/Write	20	0		Protocol flag
0X3000	timeUse0	float	Read/Write	1000	0		
0X3001	timeUse1	float	Read/Write	100	0	0	
0X3002	timeUse2	uint16	Read only			5	
0X3003	timeUse3	uint16	Read only			0	
0X3004	encoderRaw	uint16	Read only			10	Magnetic encoder sampling value
0X3005	mcuTemp	uint16	Read only			0	mcu internal temperature, *
0X3006	motorTemp	int16	Read only			11396	Motor ntc temperature, *
0X3007	vBus(mv)	int16	Read only			337	Bus voltage
0X3008	adc1Offset	int16	Read only			333	adc sampling channel 1 Zero current bias
0X3009	adc2Offset	uint16	Read only			24195	adc sampling channel 2 Zero current bias
0X300a	adc1Raw	int32	Read only			2084	adc sampling value 1
0X300b	adc2Raw	int32	Read only			2084	adc sampling value 2

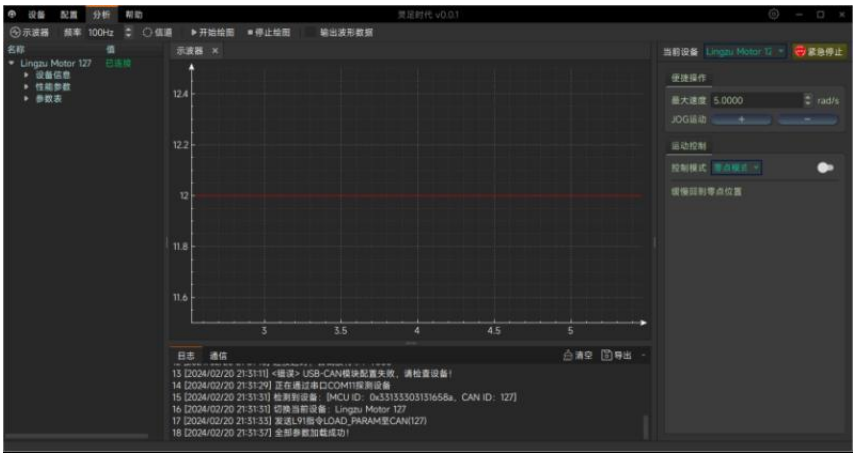
0X300c	VBUS	uint16	Read only			1232	Bus voltage V
0X300d	cmdId	uint16	Read only			1212	id ring instruction, A
0X300e	cmdIq	float	Read only			36	iq ring command, A
0X300f	cmdlocref	float	Read only			0	Position loop command, rad
0X3010	cmdspdref	float	Read only			0	Speed loop command, rad
0X3011	cmdTorque	float	Read only			0	Torque instruction, nm
0X3012	cmdPos	float	Read only			0	mit Protocol Angle instructi
0X3013	cmdVel	float	Read only			0	mit Protocol Speed instruction
0X3014	rotation	float	Read only			0	Number of turn
0X3015	modPos	float	Read only			0	Motor uncounted coi mechanical Angle, rad
0X3016	mechPos	int16	Read only			1	Load end loop mechanical Angle, rad
0X3017	mechVel	float	Read only			4.363409	Load speed: rad/s
0X3018	elecPos	float	Read only			0.777679	Electrical Angl
0X3019	ia	float	Read only			0.036618	U-wire current A
0X301a	ib	float	Read only			4.714761	V-wire current A
0X301b	ic	float	Read only			0	W-wire curren A
0X301c	timeout	float	Read only			0	Timeout count value
0X301d	phaseOrder	float	Read only			0	Directional marking
0X301e	iqf	uint32	Read only			31600	iq filter value,
0X301f	boardTemp	uint8	Read only			0	Plate temperature, *10
0X3020	iq	float	Read only			0	iq Original value, A
0X3021	id	int16	Read only			359	id Original value, A
0X3022	faultSta	float	Read only			0	Fault status value
0X3023	warnSta	float	Read only			0	Warning status value
0X3024	drv_fault	uint32	Read only			0	The driver chip fault value is 1
0X3025	drv_temp	uint32	Read only			0	The driver chip fault value is 2
0X3026	Uq	uint16	Read only			0	Q-axis voltage
0X3027	Ud	int16	Read only			48	D-axis voltage
0X3028	dtc_u	float	Read only			0	The duty cycle of the U-phase output
0X3029	dtc_v	float	Read only			0	The duty cycle of the V-phase output

0X302a	dtc_w	float	Read only			0	The duty cycle of the W-phas output
0X302b	v_bus	float	Read only			0	Vbus in the closed loop
0X302c	torque_fdb	float	Read only			0	Torque feedba value , nm
0X302d	rated_i	float	Read only			24.195	Rated current motor
0X302e	limit_i	float	Read only			0	The motor limi the maximum current
0X302f	spd_ref	float	Read only			8	Motor speed expectation
0X3030	motor_mech_angle	float	Read only			27	Motor speed expectation 2
0X3031	position	float	Read only			0	Motor position determination parameters
0X3032	chasu_angle_init	float	Read only			0	Motor position determination parameters
0X3033	chasu_angle_out	float	Read only			0	Motor position determination parameters
0X3034	motormechinit1	float	Read only			0	Motor position determination parameters
0X3035	mech_angle_init2	float	Read only			0	Motor position determination parameters
0X3036	mech_angle_ rotations	float	Read only			0	Motor position determination parameters
0X3037	cmdlocref_1	float	Read only			0	Motor position determination parameters
0X3038	status_1	float	Read only			0	Retention parameter
0X3039	ElecOffset	uint32	Read only			0	electrical Angl offset
0X303a	mcOverTemp	uint32	Read only			0	Overtemperatu threshold
0X303b	Kt_Nm/Amp	uint32	Read only			0	Moment coefficient
0X303c	Tqcali_Type	uint32	Read only			0	Motor type
0X303d	fault1	uint32	Read only			0	Log failure
0X303e	fault2	uint32	Read only			0	Log failure
0X303f	fault3	uint32	Read only			0	Log failure
0X3040	fault4	uint32	Read only			0	Log failure
0X3041	fault5	float	Read only			0	Log failure
0X3042	fault6	int16	Read only			0	Log failure
0X3043	fault7	float	Read only			0	Log failure
0X3044	fault8	uint8	Read only			0	Log failure
0X3045	theta_mech_1	float	Read only			0	Type 2 Low speed Angle

Oscilloscope

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

Click on the oscilloscope module in the analysis module, select the appropriate parameters in the channel (parameter meaning can be referred to the parameter table), set the output frequency, click on the start plot to observe the data graph, stop the plot to stop the observation graph.



Communication box instruction example:

41 54 90 07 e8 0c 08 05 70 00 00 01 00 00 00 0d 0a The

meaning is as follows

41 54	**90 07 e8 0c**	**08**	**05 70 00 00 01 00 00 00**	**0d 0a**
frame header	Number of data bits	extended frame	data frame	frame tail

The translation of extended frame canid into real canid requires the following transformations:

90 07 e8 0c converts to binary as 1001 0000 0000 0111 1110 1000 0000 1100, remove the 100 on the right and it becomes 1 0010 0000 0000 1111 1101 0000 0001, convert it to hexadecimal, It is 12 00 FD 01. According to the communication protocol, the meaning is as follows:

12 in hexadecimal	**00**	**FD**	**01**
Communication type 18 (in decimal base)	No meaning	host id	motor canid

can communication failure protection

When the value of CAN_TIMEOUT is 0, this function is disabled

When the CAN_TIMEOUT value is non-0, when the motor does not receive the can command within a certain period of time, the motor enters the reset mode, and 20000 is 1s

Motor fault instructions

Function code 0x3022 indicates the fault code, where

bit14:i square t overload fault: motor blocking overload algorithm protection bit7:

Encoder uncalibrated: Motor uncalibrated encoder

bit3: Overvoltage fault: the motor voltage exceeds the protection voltage by 60V

bit2: Undervoltage fault: the motor voltage is lower than the protection voltage of 12V bit1:

Driver chip failure: Motor driver chip failure reported

bit0: Motor overtemperature fault: motor thermistor temperature exceeds 145 degrees

Function code 0x3024 is driver chip fault code 1. The specific faults are as follows

Bit	Field	Type	Default	Description
10	FAULT	R	0b	Logic OR of FAULT status registers. Mirrors nFAULT pin.
9	VDS_OCP	R	0b	Indicates VDS monitor overcurrent fault condition
8	GDF	R	0b	Indicates gate drive fault condition
7	UVLO	R	0b	Indicates undervoltage lockout fault condition
6	OTSD	R	0b	Indicates overtemperature shutdown
5	VDS_HA	R	0b	Indicates VDS overcurrent fault on the A high-side MOSFET
4	VDS_LA	R	0b	Indicates VDS overcurrent fault on the A low-side MOSFET
3	VDS_HB	R	0b	Indicates VDS overcurrent fault on the B high-side MOSFET
2	VDS_LB	R	0b	Indicates VDS overcurrent fault on the B low-side MOSFET
1	VDS_HC	R	0b	Indicates VDS overcurrent fault on the C high-side MOSFET
0	VDS_LC	R	0b	Indicates VDS overcurrent fault on the C low-side MOSFET

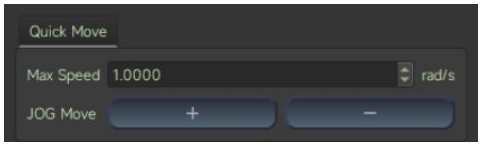
Function code 0x3025 is driver chip fault code 2. The specific faults are as follows

Bit	Field	Type	Default	Description
10	SA_OC	R	0b	Indicates overcurrent on phase A sense amplifier (DRV8353xS)
9	SB_OC	R	0b	Indicates overcurrent on phase B sense amplifier (DRV8353xS)
8	SC_OC	R	0b	Indicates overcurrent on phase C sense amplifier (DRV8353xS)
7	OTW	R	0b	Indicates overtemperature warning
6	GDUV	R	0b	Indicates VCP charge pump and/or VGLS undervoltage fault condition
5	VGS_HA	R	0b	Indicates gate drive fault on the A high-side MOSFET
4	VGS_LA	R	0b	Indicates gate drive fault on the A low-side MOSFET
3	VGS_HB	R	0b	Indicates gate drive fault on the B high-side MOSFET
2	VGS_LB	R	0b	Indicates gate drive fault on the B low-side MOSFET
1	VGS_HC	R	0b	Indicates gate drive fault on the C high-side MOSFET
0	VGS_LC	R	0b	Indicates gate drive fault on the C low-side MOSFET

Control demo

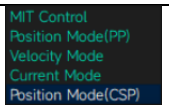
jog running

Set the maximum speed, click Run, click JOG run to make the motor run forward and backward

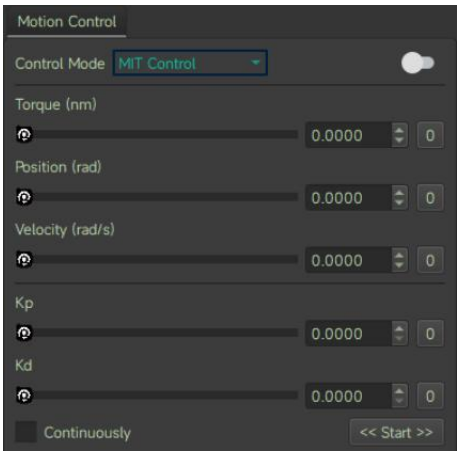


Control mode switching

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

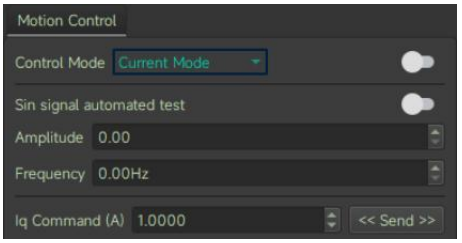


Operation control mode



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

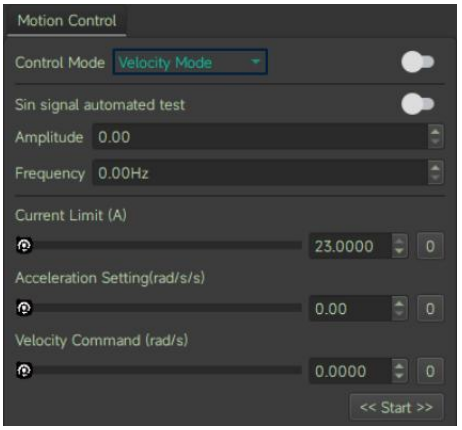
Current mode



Manually switch the current mode, click the switch button on the right side, then set the Iq current command value, start or continue to send, the motor will follow the current command, click the switch button on the right side 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, then click the switch button on the right side of the sinusoidal automatic test, and the iq (A) of the motor will run according to the amplitude and frequency of the Settings.

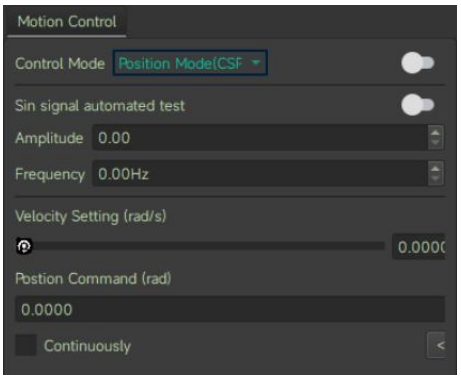
Velocity mode



Manually cut the Velocity mode, click the right switch button, then set the speed command value, start or continue to send, the motor will follow the speed command, click the right switch button 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, then click the switch button on the right side of the sinusoidal automatic test, and the motor speed (rad/s) will run according to the amplitude and frequency of the Settings.

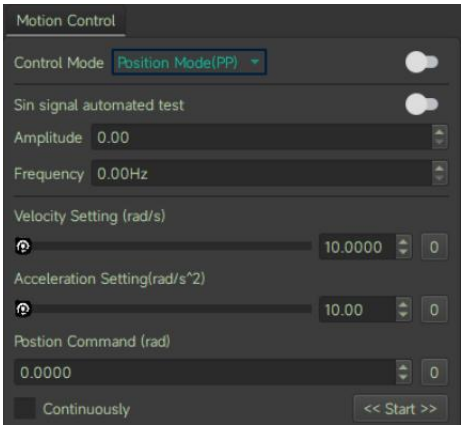
Location Mode (CSP)



Manually switch the position mode (CSP), click the right switch button, then set the position instruction value (rad), start or continuous transmission, the motor will follow the target position instruction, click the right switch button again, the motor will stop. You can set the speed to change the maximum speed for following the position.

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

Location Mode (PP)



Manually switch the position mode (PP), click the switch button on the right side, and then set the position instruction value (rad), speed setting instruction value (rad/s), acceleration setting (rad/s^2) to start or continue to send, the motor will follow the target position instruction to run, click the switch button on the right side again, the motor will stop. You can modify the maximum speed and acceleration followed by the position by setting the speed.

Firmware update

First, click Upgrade of device module and select bin file to burn; 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.

Driver protocol and instructions

The motor communication is the CAN 2.0 communication interface, the baud rate is 1Mbps, and the extended frame format is adopted as follows:

data field	**29-bit ID**			**8Byte data field**
Size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7

Description	Communication type	data area 2	Destination address	data area 1

The control modes supported by the motor include:

· Operation control mode: set 5 parameters of motor operation control; ·

Current mode: the specified Iq current of the given motor;

· Velocity mode: the specified running speed of the given motor;

· Position mode: Given the specified position of the motor, the motor will run to the specified position;

escription of the communication protocol type

Communication type 0: Get device ID

Gets the device's ID and 64-bit MCU unique identifier

data field	**29-bit ID**			**8Byte data field**
Size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
Description	0x0	bit15~8: identifies host CAN_ID	target motor CAN_ID	0

Reply frame:

data field	**29-bit ID**			**8Byte data field**
Size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
Description	0x0	target motor CAN_ID	0XFE	64-bit MCU unique identifier

Communication Type 1: operation control mode motor control instruction

Data field	**29 bit ID**			**8Byte data field**
Size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
description	0x1	Byte2: Torque (0~65535) corresponds to (-17Nm~17Nm)	target motor CAN _ID	Byte0~1: target Angle [0~65535] corresponds to (-4π~4π) Byte2~3: Target angular velocity [0~65535] corresponds to (-44rad/s~44rad/s) Byte4~5: Kp [0~65535] corresponds to (0.0~500.0) Byte6~7: Kd [0 to 65535] corresponds to the above data (0.0 to 5.0). After the conversion, the high byte is in front and the low byte is in

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

Communication Type 2: motor feedback data

Data field	**29 bit ID**			**8Byte data field**
Size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
Description	0x2	Bit8~Bit15: CAN ID of the current motor bit21~16: fault information (0 none 1 has) bit21: uncalibrated bit20: Uncalibrated bit20: Gridlock overload fault bit19: magnetic coding fault bit18: overtemperature bit17: overcurrent bit16: undervoltage fault bit22~23: Mode status 0: Reset mode [reset] 1: Cali mode [calibration] 2: Motor mode [Run]	host CAN _ID	Byte0~1: The current Angle [0~65535] Corresponding to (-4π~4π) Byte2~3: Current angular velocity [0~65535] corresponds to (-44rad/s~44rad/s) Byte4~5: Current torque [0~65535] corresponds to (-17Nm~17Nm) Byte6~7: Current temperature: Temp(Celsius) *10 If the value is higher than 10, the high byte is first and the low byte is last

Communication Type 3: Motor enabled to run

data field	**29-bit ID**			**8Byte data field**
Size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
Description	0x3	bit15~8: identifies the main CAN_ID	and target motor CAN_ID	

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

Communication Type 4: Motor stops running

data field	29-bit ID			8Byte data field
Size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
description	0x4	bit15~8: used to identify the main CAN_ID	target motor CAN_ID	When the motor is running normally, 0 must be cleared in the data field. Byte[0]=1: The fault is cleared.

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

Communication type 6: Set motor mechanical zero

data field	29-bit ID			8Byte data field
Size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
description	0x6	bit15~8: Identifies the main CAN_ID	and target motor CAN_ID	Byte[0]=1

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

Communication type 7: Set motor CAN_ID

Change the current motor CAN_ID, effective immediately.

data field	29-bit ID			8Byte data field
Size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
description	0x7	bit15~8: used to identify main CAN_ID Bit16~23: preset CAN_ID	Target motor CAN_ID	

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

Communication type 17: Single parameter read

Data field	29 bit ID			8Byte data field
Size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
description	0x11	bit15~8: Used to identify the main CAN_ID	target motor CAN_ID	Byte0~1: index. For details, see the readability parameter table below Byte2~3: 00 Byte4~7: In data above 00, the low byte is first and the high byte is second (

Reply frame:

Data field	29 bit ID			8Byte data field
Size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
description	0x11	bit15~8: indicates that the master CAN_ID Bit23~16:00 indicates that the master CAN_ID is successfully read. 01 indicates that the master can_ID	Byte0~1: Byte2~3: 00 Byte4~7: Parameter data. 1 byte of data above Byte4 is preceded by low bytes and followed by high bytes at	

Communication type 18: Single parameter write (lost in power failure)

With type 22, the parameter starting with function code 0x20 of the parameter table in the upper computer module can be saved

Data field	29 bit ID			8Byte data field
Size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
description	0x12	bit15~8: Used to identify the main CAN_ID	target motor CAN_ID	Byte0~1: index. For details, see the readability parameter table below Byte2~3: 00 Byte4~7: Parameter data In the preceding data, the low byte is in the front and the high byte is in the rear

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

Communication type 21: Fault feedback frame

data field	29-bit ID			8Byte data field
Size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
description	0x15	bit15~8: motor CAN_ID	identifies the main CAN_ID	Byte0~3: fault value (non-0: faulty; 0: faulty). Normal) bit14: gridlock i square t overload fault bit7: encoder not calibrated

				bit3: overvoltage fault bit2: undervoltage fault bit1: driver chip fault bit0: motor overtemperature fault, Default 145 °C Byte4~7: warning Value bit0: motor overtemperature warning, the default is 135 °c

Communication type 22: Motor data save frame

data field	**29-bit ID**			**8Byte data field**
Size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
Description	0x16	bit15~8: identifies the main CAN_ID	and target motor CAN_ID	01 02 03 04 05 06 07 08

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

Communication type 23: Motor baud rate modification frame (re-power-on effect)

Data field	**29 bit ID**			**8Byte data field**
Size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
description	0x17	bit15~8: used to identify the main CAN_ID	target motor CAN_ID	01 02 03 04 05 06 **F_CMD** Among them, the F_CMD byte is the motor baud rate Among them, 01 is 1M 02 is 500K 03 is 250K 04 is 125K

Response frame: Response motor feedback frame (see communication type 0)

Communication type 24: The motor actively reports frames

data field	**29-bit ID**	**8Byte data field**		
Size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
description	0x18	bit15~8: identifies the main CAN_ID	target motor CAN_ID	01 02 03 04 05 06 **F_CMD** Among them, the F_CMD byte is the motor reporting switch 00 is to disable active reporting (default) 01 To enable active reporting, the default reporting interval is 10ms

Response frame:

数据域	**29位ID**			**8Byte数据区**
大小	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
描述	0x18	Bit8~Bit15: CAN ID of the current motor bit21~16: fault information (0 none 1 has) bit21: uncalibrated bit20: Uncalibrated bit20: Gridlock overload fault bit19: magnetic coding fault bit18: overtemperature bit17: overcurrent bit16: undervoltage fault bit22~23: Mode status 0: Reset mode [reset] 1: Cali mode [calibration] 2: Motor mode [Run]	target motor CAN_ID	Byte0~1: The current Angle [0~65535] Corresponding to (-4π~4π) Byte2~3: Current angular velocity [0~65535] corresponds to (-44rad/s~44rad/s) Byte4~5: Current torque [0~65535] corresponds to (-17Nm~17Nm) Byte6~7: Current temperature: Temp(Celsius) *10 If the value is higher than 10, the high byte is first and the low byte is last

Communication type 25: Motor protocol modification frame (re-power-on effect)

Data field	29 bit ID			8Byte data field
Size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
description	0x19	bit15~8: used to identify the main CAN_ID	target motor CAN_ID	01 02 03 04 05 06 **F_CMD** Among them, the F_CMD byte is the motor protocol type Among them, 0 is a private protocol (default) 1 is the Canopen protocol 2 is the MIT protocol

Response frame: Response motor feedback frame (see communication type 0)

Read and write a single parameter list

index		Description	Type	Number of bytes		R/W Read and write permission
0X7005	run_mode	0: operation mode 1: position mode (PP) 2: Velocity mode 3: Operation mode Current mode 5: Position mode (CSP)	uint8	1		W/R
0X7006	iq_ref	Current mode Iq command	float	4	-23 to 23A	W/R
0X700A	spd_ref	Rotational Velocity mode Rotational speed command	float	4	-44to 44rad/s	W/R
0X700B	limit_torque	torque limit	float	4	0 to 17Nm	W/R
0X7010	cur_kp	Kp	float	4	The default value is 0.17	W/R
0X7011	cur_ki	Ki	float	4	The default value is 0.012	W/R
0X7014	cur_filt_gain	filt_gain	float	4	0 to 1.0, The default value is 0.1	W/R
0X7016	loc_ref	Position Mode Angle instruction	float	4	rad	W/R
0X7017	limit_spd	Location mode (CSP) speed limit	float	4	0 to 44rad/s	W/R
0X7018	limit_cur	Velocity position mode Current limitation	float	4	0 to 23A	W/R
0x7019	mechPos	Mechanical Angle of the loading coil	float	4	rad	R
0x701A	iqf	iq Filter	float	4	-16 to 16A	R
0x701B	mechVel	Speed of the load	float	4	-44 to 44rad/s	R
0x701C	VBUS	Bus voltage	float	4	V	R
0x701E	loc_kp kp	at	float	4	The default value is 40	W/R
0x701F	spd_kp	Indicates the speed kp	float	4	The default value is 6	W/R
0x7020	spd_ki	ki	float	4	The default value is 0.02	W/R
0x7021	spd_filt_gain	Speed filter value	float	4	The default value is 0.1	W/R
0x7022	acc_rad	velocity mode acceleration	float	4	The default value is 20rad/s^2	W/R
0x7024	vel_max	Location mode (PP) speed	float	4	The default value is 10rad/s	W/R
0x7025	acc_set	Location mode (PP) acceleration	float	4	The default value is 10rad/s^2	W/R
0x7026	EPScan_time	Indicates the report time. 1 indicates 10ms. Plus 1 increments by 5ms	uint16	2	The default value is 1	W/R
0x7028	canTimeout	can The timeout threshold, 20000 is 1s	uint32	4	The default value is 0	W/R

0x7029	zero_sta	Indicates the zero flag bit, 0 means 0-2π and 1 means -π-π	uint8	1	The default is 0	W/R

Read example:

Take reading loc_kp as an example:

Read instruction is

Size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
	0x11	0x00FD	0x7F	1E 70 00 00 00 00 00 00
Description	Type 17	Host id 0xFD	Target motor CAN_ID 7F	Byte0~1: index, corresponding to loc_kp

The feedback instruction is

Size	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
	0x11	0x007F	0xFD	1E 70 00 00 00 00 F0 41
Description	Type 17	bit15~8: Target motor CAN_ID 7F	Host id 0xFD	Byte0~1: index, corresponding to loc_kp Byte4~7:loc_kp value 30, high right byte, (32-bit single precision) hexadecimal IEEE-754 standard floating point number

Motor Function Description

(If the following features are unavailable, please upgrade to the latest version via the oficial Git repository.)

---

1. Active Reporting

Disabled by default. Enable via **Type 24**.

Report type: **Type 2** (default interval: **10ms**). Adjust interval by modifying `EPScan_time` via **Type 18**.

---

2. Zero-Point Flag (`zero_sta`)

Modify via:

- **Host computer**
- **Type 18** (*requires saving via **Type 22** for communication*)

Default flag: `0` → Power-on position range: **0–2π**.

If set to `1`: Power-on position range: **π–π**.

---

3. Type 2 Update

Updated to **periodic looping within -4π–4π** (enables cycle counting). **Note**:

Position interface parameters must be adjusted:

- `P_MIN`: **12.57f**
- `P_MAX`: **12.57f**

---

4. Protocol Switching *(Requires CAN adapter)*

Methods:

- Modify `protocol_1` via host computer.

Send **Type 25** command.

Reboot required after switching.

Post-switch CAN commands:

- **CANopen**: Send extended frame (protocol switch frame).

MIT Protocol: Send standard frame (**Command 8**).

---

5. Post-Power-Off Anti-Backdrive Protection

Default: Motor imposes damping if rotated rapidly while powered off (prevents surge).

Disable: Set `damper = 1`.

---

6. Zero Calibration Rules

- **Supported modes**: **CSP** and **Motion Control**.

- **PP Mode**: Zero calibration is **blocked**. •

Old vs. New Versions:

- **Old**: Zero calibration causes large deviation → motor immediately moves to target.
- **New** (CSP/Motion Control): Target updates to `0` instantly → motor remains stationary.

---

7. Position Offset (`add_offset`)

Example: If offset = `1` , the current zero shifts to **(current position + 1 rad)**.

Use case: Bypass mechanical limits (e.g., set zero at `1 rad` → power-on treats `1 rad` as new zero).

---

8. CANopen ID

Old version: Fixed to `1` .

New version: Matches the **private protocol CAN ID**.

Notes for Implementation

- Always **save settings** (e.g., Type 22 for `zero_sta`).
- Verify **CAN adapter compatibility** for protocol switching.
- For zero offsets, ensure mechanical safety limits are respected.

Control mode instructions

Sample program

Examples of various mode control motors are provided below (take gd32f303 as an example) The following are library, function, and macro definitions for the various instances

```
#define P_MIN -12.57f

#define P_MAX 12.57f

#define V_MIN -44.0f

#define V_MAX 44.0f

#define KP_MIN 0.0f

#define KP_MAX 500.0f

#define KD_MIN 0.0f

#define KD_MAX 5.0f

#define T_MIN -17.0f

#define T_MAX 17.0f

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))) //Parses 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 lists the common types of communication sent:

Motor Enabled 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();
}
```

Operation control mode Motor control instruction (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();
}
```

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();
}
```

Motor mode parameter write command (communication type 18, running mode switch)

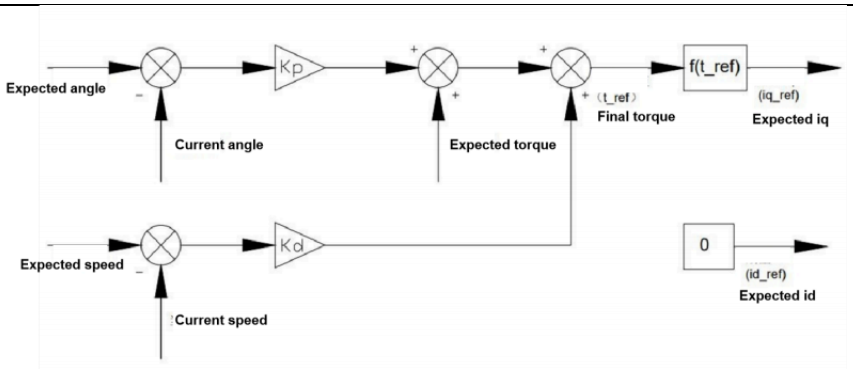
```
uint8_t runmode;
uint16_t index;
void motor_modechange(uint8_t id, uint16_t master_id)
{
    txCanIdEx.mode = 0x12;
    txCanIdEx.id = id;
```

```
    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();
}
```

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();
}
```

Operation control mode



The motor is in operation control mode by default after power-on.

Send motor Enable Run frame (communication type 3) → Send operation mode motor control command (communication type 1) → Receive motor feedback frame (communication type 2)

Operation control mode description:

The control logic of the operation and control mode is t_ref=Kd * (v_vset-v_actual)+Kp * (p_set-p_actual)+t_ff. Tref is converted to the expected iq current through an internal formula and output through the current loop Simple control demonstration:

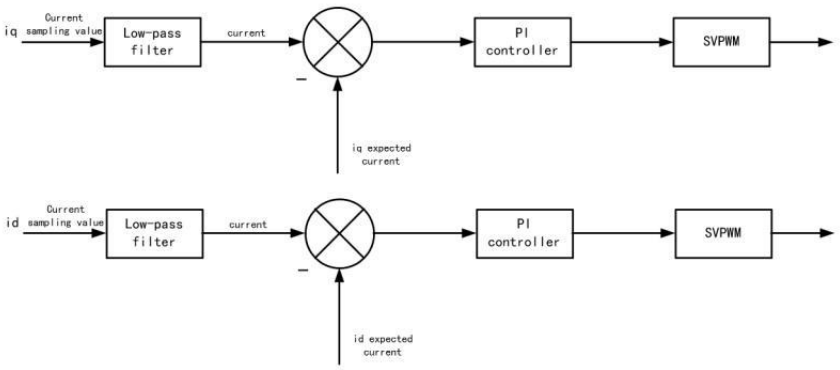
Set t_ff to 0, v_vset to 1, Kd to 1, p_set to 0, Kp to 0. If there is no external load on the motor, it will run at a speed of 1rad/s. If there is an external load, kd needs to be increased to resist the external load

Set t_ff to 0, v_vset to 0, Kd to 1, p_set to 0, Kp to 0, the motor is in damping mode. When the motor is externally rotated, a damping is applied, which increases with the increase of kd. It should be noted that the motor generates electricity under this condition and requires power supply to prevent overvoltage

Set t_ff to 0, v_vset to 0, Kd to 1, p_set to 5, Kp to 1. If there is no external load on the motor, it will run to the target position of 5.

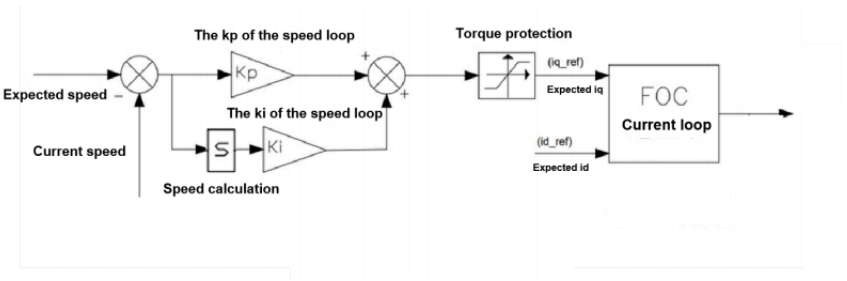
Increasing kp will increase the force required to maintain the target position, and kd is damping. Without kd, the motor will sway to the target position

Current mode



Send motor mode parameter write command (communication type 18) Set the runmode parameter to 3 → Send motor Enable run frame (communication type 3) → Send motor mode parameter write command (communication type 18) set the iq_ref parameter to the default current instruction

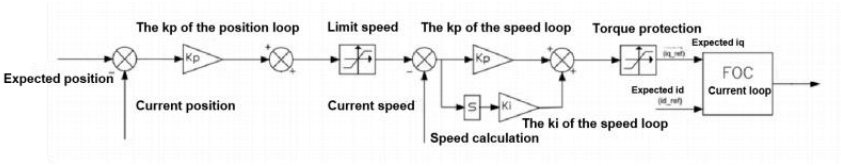
Velocity mode



Send motor mode parameter write command (communication type 18) Set the 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 as

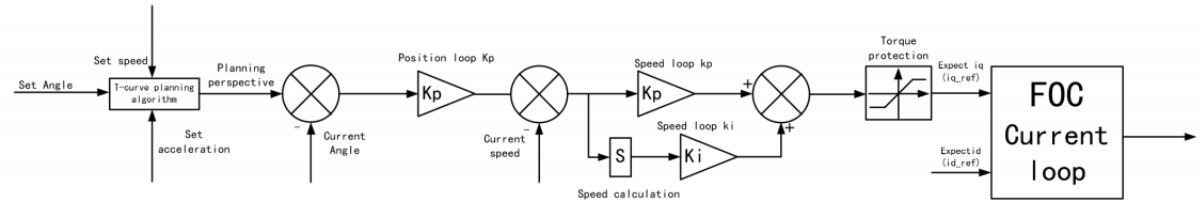
default maximum current instruction → Send motor mode parameter write command (communication type 18) Set acc_rad parameter as default acceleration instruction → Send motor mode parameter write command (communication type 18) Set spd_ref parameter as default speed instruction

Location Mode (CSP)



Send motor mode parameter write command (communication type 18) Set the runmode parameter to 5 → Send motor Enable run frame (communication type 3) → Send motor mode parameter write command (communication type 18) set limit_spd parameter as default maximum speed instruction → Send motor mode parameter write command (communication type 18) Sets loc_ref parameter as default position instruction

Location Mode (PP)



Send motor mode parameter write command (communication type 18) Set the runmode parameter to 1 → Send motor Enable run frame (communication type 3) → Send motor mode parameter write command (communication type 18) set The vel_max parameter is the default maximum speed instruction → Send motor mode parameter write command (communication type 18) Set the acc_set parameter to the default acceleration instruction → Send motor mode parameter write command (communication type 18) Set the loc_ref parameter to the default position instruction

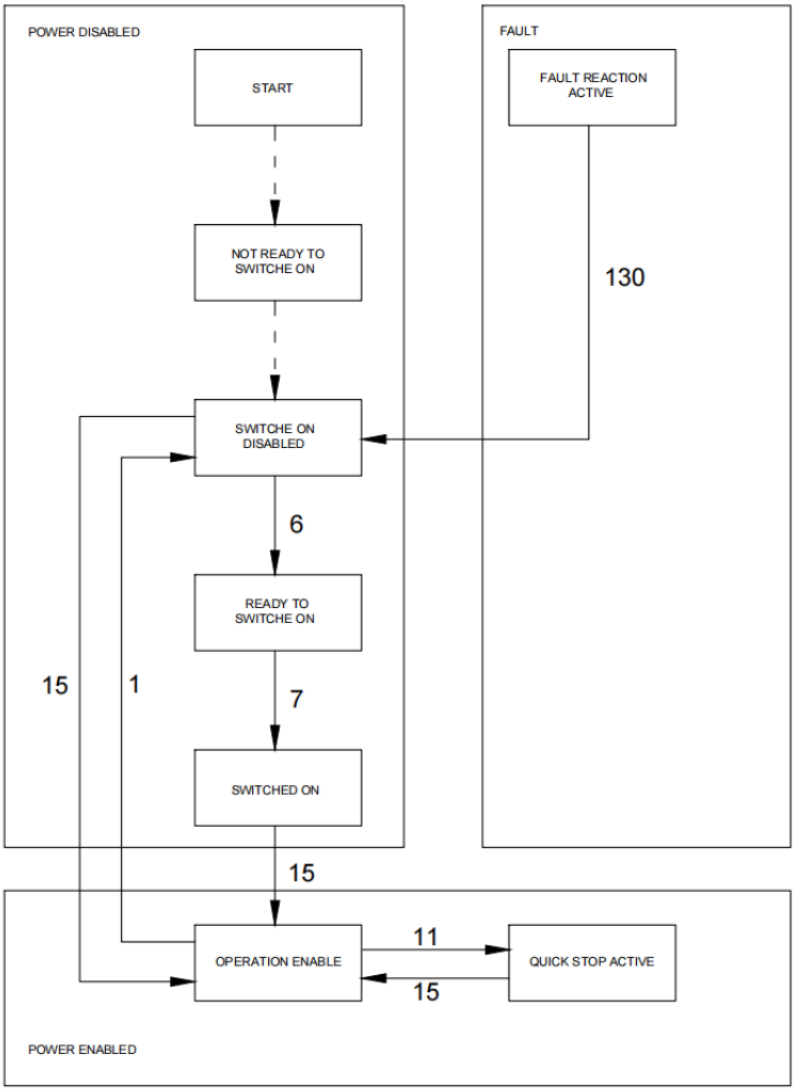
Note: This mode does not support changing the speed and acceleration during operation. If you want to make an emergency stop, you can change vel_max to 0 during the process, and it will stop at the current speed and acceleration plan

Stop running

Sending motor stop frame (communication type 4)

Explanation of Canopen Communication Protocol Types

State Machine Description



Motor Enable:

When initially powered on, the motor defaults to the **SWITCH_ON_DISABLED** state. To transition to **OPERATION_ENABLE**, modify the **Controlword (6040H)** to **6, 7, or 15** (step-by-step transition), or directly set it to **15** for immediate enablement.

Stopping the Motor:

If the motor is in **OPERATION_ENABLE** state and needs to stop normally, modify the **Controlword (6040H)** to **1**. The motor will return to the disabled state (**SWITCH_ON_DISABLED**).

Emergency Stop (Use with Caution—Risk of Voltage Surge):

During operation, an emergency stop can be triggered by setting the **Controlword (6040H)** to **11**. **Fault**

Clearance:

If the motor enters a **FAULT** state due to protection mechanisms, modifying the **Controlword (6040H)** can clear standard errors. **Important**

Note:

Mode changes for this motor must be performed in the **disabled state (SWITCH_ON_DISABLED)**. Ensure the desired mode is configured **before** enabling **OPERATION_ENABLE** to avoid unexpected behavior.

Status Feedback Parameters

Index	Name	Attribute	Type	Unit
603F	**Error_code**	Read-only	UINTEGER16	/
6041	**Statusword**	Read-only	UINTEGER16	/
6061	**Modes_of_operation_display**	Read-only	INTEGER8	/
6062	**Position_demand_value**	Read-only	INTEGER32	Pulses (1 rev = 16,384 pulses)

Index	Name	Attribute	Type	Unit
6064	Position_actual_value	Read-only	INTEGER32	Pulses (1 rev = 16,384 pulses)
606B	Velocity_demand_value	Read-only	INTEGER32	0.1 rpm
606C	Velocity_actual_value	Read-only	INTEGER32	0.1 rpm
6077	Torque_actual_value	Read-only	INTEGER16	0.1% load ratio (1000 = 6 N·m)
6078	Current_actual_value	Read-only	INTEGER16	mA
6079	DC_link_circuit_voltage	Read-only	INTEGER32	mV

Homing Mode (Zero Position Setting)

Index	Name	Attribute	Type	Unit
6040	Controlword	Read-write	UINTEGER16	/
6060	Modes of operation	Read-write	INTEGER8	/

Homing method:

- Set **Modes of operation** to **6** while the motor is in the **disabled state (SWITCH_ON_DISABLED)**. The motor will then define the current position as the zero point.
- To **hold the zero position**, modify the **Controlword** to **15**, and the motor will maintain its position at the home location.

Position Mode (PP - Profile Position)

Index	Name	Attribute	Type	Unit
6040	Controlword	Read-write	UINTEGER16	/
6060	Modes of operation	Read-write	INTEGER8	/
6067	Position_window	Read-write	UINTEGER32	Pulses (1 rev = 16,384 pulses)
6068	Position_window_time	Read-write	UINTEGER16	ms
6071	Target_torque	Read-write	INTEGER16	0.1% load ratio (1000 = 6 N·m)
607A	Target_position	Read-write	INTEGER32	Pulses (1 rev = 16,384 pulses)
6081	Profile_velocity	Read-write	UINTEGER32	0.1 rpm
6083	Profile_acceleration	Read-write	UINTEGER32	0.1 rpm/s

Steps to Configure Position Mode (PP):

1. While the motor is in the **disabled state (SWITCH_ON_DISABLED)**, set **Modes of operation** to **1**.

   Mandatory parameters:

   - **Target_torque** (absolute max torque in position mode)
   - **Profile_velocity** (absolute speed in position mode)
   - **Profile_acceleration** (absolute acceleration in position mode)

   Optional parameters:

   - **Position_window** (if not set, window check is disabled)
   - **Position_window_time** (if not set, window check is disabled)

2. Set **Controlword (6040)** to **15** to enable operation.
3. **Set Target_position** (absolute position) to move the motor to the desired position.

Position Mode (CSP - Cyclic Synchronous Position)

Index	Name	Attribute	Type	Unit
6040	Controlword	Read-write	UINTEGER16	/
6060	Modes of operation	Read-write	INTEGER8	/
6067	Position_window	Read-write	UINTEGER32	Pulses (1 rev = 16,384 pulses)
6068	Position_window_time	Read-write	UINTEGER16	ms
6071	Target_torque	Read-write	INTEGER16	0.1% load ratio (1000 = 6 N·m)
607A	Target_position	Read-write	INTEGER32	Pulses (1 rev = 16,384 pulses)
6081	Profile_velocity	Read-write	UINTEGER32	0.1 rpm

Steps to Configure Position Mode (CSP):

1. While the motor is in the **disabled state (SWITCH_ON_DISABLED)**, set **Modes of operation** to **5**.

- **Mandatory parameters**:
  - **Target_torque** (absolute max torque in position mode)
  - **Profile_velocity** (absolute speed in position mode)

Optional parameters:

- **Position_window** (0 = disabled)
- **Position_window_time** (0 = disabled)

2. Set **Controlword (6040)** to **15** to enable operation.
3. **Set Target_position** (absolute position) to move the motor to the desired position.

Velocity Mode

Index	Name	Attribute	Type	Unit
6040	**Controlword**	Read-write	UINTEGER16	/
6060	**Modes of operation**	Read-write	INTEGER8	/
6071	**Target_torque**	Read-write	INTEGER16	0.1% load ratio (1000 = 6 N·m)
60FF	**Target_velocity**	Read-write	INTEGER32	0.1 rpm

Steps to Configure Velocity Mode:

1. While the motor is in the **disabled state (SWITCH_ON_DISABLED)**, set **Modes of operation** to **3**.

Mandatory parameter:

- **Target_torque** (absolute max torque in velocity mode)

2. Set **Controlword (6040)** to **15** to enable operation.
3. **Set Target_velocity** to reach the desired speed.

Torque Mode

Index	Name	Attribute	Type	Unit
6040	**Controlword**	Read-write	UINTEGER16	/
6060	**Modes of operation**	Read-write	INTEGER8	/
6071	**Target_torque**	Read-write	INTEGER16	0.1% load ratio (1000 = 6 N·m)

Steps to Configure Torque Mode:

1. While the motor is in the **disabled state (SWITCH_ON_DISABLED)**, set **Modes of operation** to **4**.
2. Set **Controlword (6040)** to **15** to enable operation.
3. **Set Target_torque** to output the desired torque.

Protocol Switching (Extended Frame): Switch Motor Protocol (Takes Effect After Power Cycle)

Data Field	29-bit ID	8-Byte Data Area
Size	Bit 28~0	Byte 0~6
Description	**0xFFF**	01 02 03 04 05 06 F_CMD

- **F_CMD** (Byte 6) defines the motor protocol:
  - **0**: Private protocol (default)
  - **1**: CANopen protocol
  - **2**: MIT protocol

Response Frame:

Data Field	11-bit ID	8-Byte Data Area
Size	Bit 10~0	Byte 0~7
Description	Motor ID	64-bit MCU unique identifier

MIT Communication Protocol Description

The motor communication adopts the CAN 2.0 interface with a default baud rate of 1 Mbps. The baud rate can be modified by switching to the private protocol. The standard frame format is as follows:

Data Field	11-bit ID		8-byte Data Area
Size	Bit 10~8	Bit 7~0	Byte 0~7
Description	Mode type	ID	

Supported Control Modes:

- **MIT Mode**: Provides five motion control parameters to the motor. .

Velocity Mode: Specifies the target speed for the motor.

- **Position Mode**: Specifies the target position and speed, allowing the motor to run to the designated position at the configured speed.

Response Command 1: Data Feedback (Motor Status)

Data Field	11-bit ID	8-byte Data Area
Size	Bit 10~0	Byte 0~7
Description	Host ID	**Byte 0**: Motor CAN ID **Byte 1~2**: Target angle [0~65535], corresponds to (-12.57 rad ~ 12.57 rad) **Byte 3 (high 8 bits), Byte 4[7-4] (low 4 bits)**: Target speed [0~4096], corresponds to (-44 rad/s ~ 44 rad/s) **Byte 4[3-0] (high 4 bits), Byte 5 (low 8 bits)**: Target torque [0~4096], corresponds to (-17 N·m ~ 17 N·m) **Byte 6~7**: Winding temperature (in degrees)

Response Command 2: MCU Identification

Data Field	11-bit ID	8-byte Data Area
Size	Bit 10~0	Byte 0~7
Description	Motor ID	64-bit MCU unique identifier

Command 1: Enable Motor Operation

Data Field	11-bit ID	8-byte Data Area
Size	Bit 10~0	Byte 0~7
Description	Target motor CAN ID	FF FF FF FF FF FF FF FC

Response: Response Command 1

Command 2: Stop Motor Operation

Data Field	11-bit ID	8-byte Data Area
Size	Bit 10~0	Byte 0~7
Description	Target motor CAN ID	FF FF FF FF FF FF FF FD

Response: Response Command 1

Command 3: MIT Dynamic Parameters

Data Field	11-bit ID	8-byte Data Area
Size	Bit 10~0	**Byte 0~1**: Target angle [0~65535], (-12.57 rad ~ 12.57 rad)**Byte 2 (high 8 bits), Byte 3[7-4] (low 4 bits)**: Target speed [0~4096], (-44 rad/s ~ 44 rad/s)**Byte 3[3-0] (high 4 bits), Byte 4 (low 8 bits)**: Kp [0~4096], (0~500)**Byte 5 (high 8 bits), Byte 6[7-4] (low 4 bits)**: Kd [0~4096], (0~5)**Byte 6[3-0] (high 4 bits), Byte 7 (low 8 bits)**: Target torque [0~4096], (-17 N·m ~ 17 N·m)

Response: Response Command 1

Command 4: Set Zero Position (Non-Position Mode)

Data Field	11-bit ID	8-byte Data Area
Size	Bit 10~0	Byte 0~7
Description	Target motor CAN ID	FF FF FF FF FF FF FF FE

Response: Response Command 1

Command 5: Clear Errors & Read Fault Status

Data Field	11-bit ID	8-byte Data Area
Size	Bit 10~0	FF FF FF FF FF FF F_CMD FB**F_CMD**:- 0xFF → Clear current fault- Any other value → Returns fault value in **Byte 1** of the response

Response (Fault Clear): Response Command 1

Fault Status Response:

Data Field	11-bit ID	8-byte Data Area
Size	Bit 10~0	**Byte 0**: Motor CAN ID**Byte 1~4**: Fault value (Non-zero: Fault present; 0: Normal)**Bit 14**: Stall/I ŧ overload fault<**Bit 7**: Encoder not calibrated **Bit 3**: Overvoltage fault**Bit 2**: Undervoltage fault**Bit 1**: Driver IC fault**Bit 0**: Motor overtemperature fault (Default threshold: 103 ℃)

Command 6: Set Operation Mode

Data Field	11-bit ID	8-byte Data Area
Size	Bit 10~0	FF FF FF FF FF FF F_CMD FC**F_CMD**: Mode type- 0: MIT mode (default)- 1: Position mode- 2: Velocity mode

Response: Response Command 1

Command 7: Modify Motor CAN ID

Data Field	11-bit ID	8-byte Data Area
Size	Bit 10~0	FF FF FF FF FF FF F_CMD FA**F_CMD**: Target motor CAN ID

Response: Response Command 2

Command 8: Change Communication Protocol (Takes Effect After Power Cycle)

Data Field	11-bit ID	8-byte Data Area
Size	Bit 10~0	FF FF FF FF FF FF F_CMD FD**F_CMD**: Protocol type- 0: Private protocol (default)- 1: CANopen - 2: MIT protocol

Response: Response Command 2

Command 9: Modify Host CAN ID

Data Field	11-bit ID	8-byte Data Area
Size	Bit 10~0	FF FF FF FF FF FF F_CMD 01**F_CMD**: Host CAN ID

Response: Response Command 2

Command 10: Position Mode Control Command

Data Field	11-bit ID		8-byte Data Area
Size	Bit 10~8	Bit 7~0	**Byte 0~3**: Target position (rad, 32-bit float)**Byte 4~7**: Target speed (rad/s, 32-bit float)
Description	1	Target motor CAN ID	

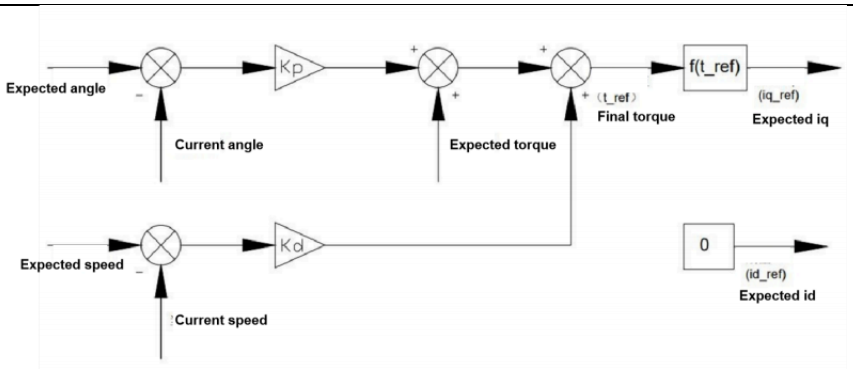
Response: Response Command 1

Command 11: Velocity Mode Control Command

Data Field	11-bit ID		8-byte Data Area
Size	Bit 10~8	Bit 7~0	**Byte 0~3**: Target speed (rad/s, 32-bit float)**Byte 4~7**: Current limit in speed/position mode (A, 32-bit float)
Description	2	Target motor CAN ID	

Response: Response Command 1

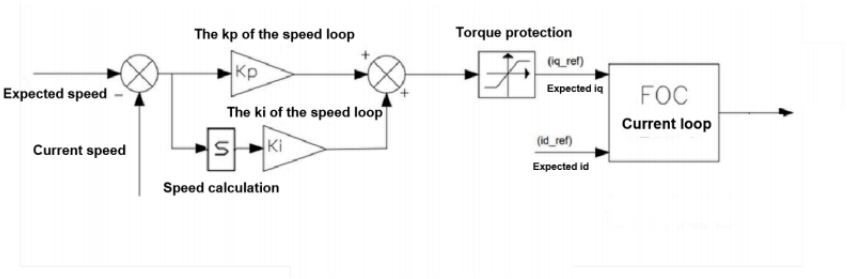
Motion Control Mode



The motor defaults to Motion Control Mode upon power-up.

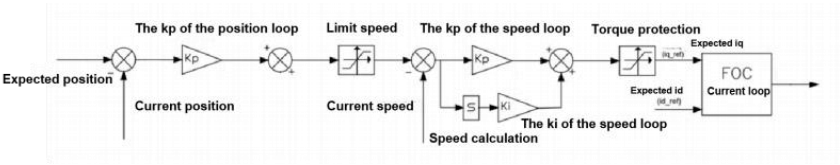
1. Send the *Motor Enable Command* (Command 1).
2. Send the *Motion Control Command* (Command 3) to activate dynamic parameter control.
3. Send the *Motor Stop Command* (Command 2) to halt operation when needed.

Velocity Mode



1. Configure the motor's operation mode by sending *Set Operation Mode Command* (Command 6) with **Mode = 2 (Velocity Mode)**.
2. Send the *Motor Enable Command* (Command 1) to activate the motor.
3. Send the *Velocity Mode Control Command* (Command 11) to set the **maximum current (absolute value)** and **target speed**.
4. To stop, send the *Motor Stop Command* (Command 2).

Position Mode (CSP - Cyclic Synchronous Position)



1. Configure the motor's operation mode by sending *Set Operation Mode Command* (Command 6) with **Mode = 1 (Position Mode)**.
2. Send the *Motor Enable Command* (Command 1) to activate the motor.
3. Send the *Position Mode Control Command* (Command 10) to set the **maximum speed (absolute value)** and **target position**.
4. To stop, send the *Motor Stop Command* (Command 2).

Product warranty clause of FULLING MOTOR

1) One year warranty

FULLING provides a one-year warranty against defects in the raw materials and workmanship of its products from the date of shipment. During the warranty period, FULLING provides free repair service for defective products.

2) Not covered by warranty

A、Inappropriate wiring, such as reversed polarity of the power supply and live plugging and unplugging

B、Unauthorized modification of internal components

C、Use beyond electrical and environmental requirements

D、Poor environmental heat dissipation

3) Repair process

If the product needs to be repaired, the following process will be followed:

(1) Before shipping, please call the customer service personnel of FULLING to obtain a repair permit number;

(2) Please send a written explanation along with the goods, explaining the phenomenon of the faulty drive being repaired; The voltage, current, and usage environment at the time of the malfunction; The name, phone number, and mailing address of the contact person.

(3) Prepaid postage to JIANGSU FULLING MOTOR TECHNOLOGY CO., LTD., No. 217 Huanghe West Road, Xinbei District, Changzhou City, Jiangsu Province

Postal code: 213032.

4) Warranty Limitations

A、The warranty scope of FULLING's products is limited to the components and processes of the products (i.e. consistency)

B、FULLING does not guarantee that its products will be suitable for the specific use of customers, as the suitability i s also related to the technical specifications, usage conditions, and environment of the use.

5) Maintenance requirements

When repairing, please truthfully fill out the "Repair Report" (this form can be downloaded from www.fullingmotor.com) for maintenance analysis. Mailing address: JIANGSU FULLING MOTOR TECHNOLOGY CO., LTD., No. 217 Huanghe West Road, Xinbei District, Changzhou City, Jiangsu Province. Postal code: 213032.

JIANGSU FULLING MOTOR TECHNOLOGY CO., LTD

Address: No. 217 Huanghe West Road, Xinbei District, Changzhou City, Jiangsu Province

Postal Code: 213032

Phone:+86-519-85132957

Fax:+86-519-85132956

Email: info@fullingmotor.com