Robot Joint J60 Series

J60-6、J60-10

Product Manual V1.0.7-0

2025/06/30

DEEP Robotics

Statement

- This manual is the information asset owned by Hangzhou Yunshenchu Technology Co.,Ltd. (hereafter referred to as “DEEP Robotics”) and any reprinting of part or all of this manual is strictly prohibited without the permission of DEEP Robotics.
- This manual provides corresponding descriptions of the specifications, parameters, interface cables, debugging tools, operation and use, and troubleshooting of J60 series robot joint.
- Please operate and use the joint based on careful reading and full understanding of this manual.
- Basic safety instructions are described in detail in "Reading Tips". Please be sure to read before reading this manual to ensure correct use.
- The diagrams and photographs in this manual are representative examples and may have minor differences from the purchased product.
- This manual may be modified appropriately due to product improvements, specification changes, and other reasons.
- The content recorded in this manual may contain errors or omissions. If this manual is damaged, lost, or if there are any questions about the content of this manual, please contact our company in time.
- Malfunctions caused by customers disassembling or modifying the product without authorization are not covered by warranty, see "Warranty Policy" for details.

Reading Tips

Description of Symbols

Before using (installation, transportation, maintenance, repair), please be sure to read and master this manual, and be familiar with the equipment knowledge and safety precautions before starting to use it. The safety precautions in this manual are divided into three major items: "Caution", "Mandatory", and "Prohibited". Even the content recorded under "Caution" can have serious consequences depending on the situation, so every precaution is extremely important, and must be strictly followed.

Caution Usage tips or operational recommendations. Improper using or operating the robot may cause damage to it.

Mandatory Matters that must be observed.

Prohibition Matters prohibited. Misoperation is dangerous and may cause injury to operators or damage to the robot.

Help Acquiring

To obtain more help resources to assist you in skillfully using the J60 series robot joint, you can also visit DEEP Robotics’ corporate website: http://www.deeprobotics.cn/.

Resources For Development

Deep Robotics provides development kits on Github, including motor control SDK based on CAN bus and debugger software: https://github.com/DeepRoboticsLab/Deep_Motor_SDK.

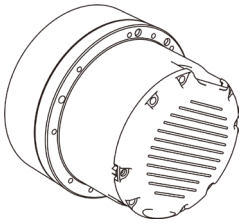
Contents

1 Product Introduction

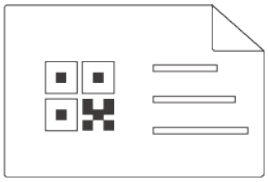
1.1 Product Overview

J60 series robot joint is a lightweight high-performance robot joint module that integrates a reducer, frameless torque motor, servo driver, and absolute encoder. Users can obtain status or send parameters through debugger software or CAN bus for development.

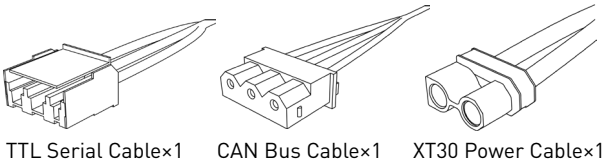
1.2 Product List



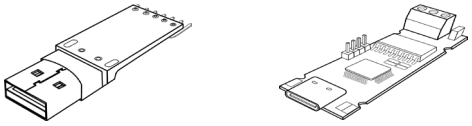
Joint ×1



Product Card×1



TTL Serial Cable×1　CAN Bus Cable×1　XT30 Power Cable×1



USB to TTL Adapter×1　USB to CAN Adapter×1
(including USB to Type-C Cable)

Accessories Kit (optional)

※ Accessories Kit is not included as standard and needs to be purchased separately. Users can also select and purchase suitable accessories according to the interface model.

1.3 Specification Parameters

Standard Operating Conditions	
Standard Operating Voltage	DC 24V
Operating Voltage Range	DC 18V-36V
Rotation Direction	CW/CCW As viewed from the output shaft
Standard Operating Temperature	25°C

Overall Specifications	**J60-6**	**J60-10**
Weight	480g	540g
Dimensions (Diameter × Length)	76.5mm×63mm	76.5mm×72.5mm
Rated Torque	6.09N·m	9.58N·m
Peak Torque	19.94N·m	30.50N·m
Rated Rotate Speed (24V)	15.3rad/s	10rad/s
Peak Rotate Speed (24V)	24.18rad/s	15.49rad/s

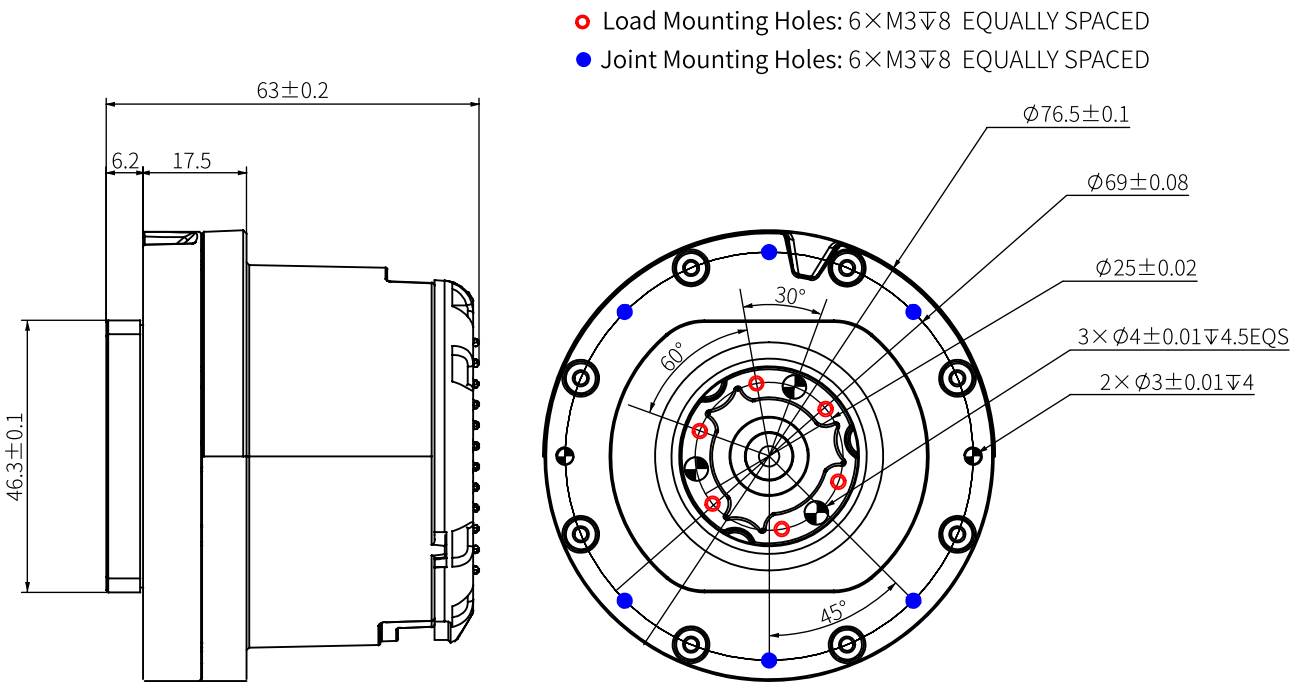
Electrical Parameters	J60-6	J60-10
Standard Operating Voltage	DC 24V	DC 24V
Operating Voltage Range	DC 18V~36V	DC 18V~36V
Motor Rated Current (rms)	5.6A	5.6A
Motor Maximum Current (rms)	18A	18A
Motor Maximum Phase Current (peak)	30A	30A
Current Sampling Range	±40A	±40A
Joint Torque Constant	0.8982N·m/A	1.3473N·m/A
Motor Back EMF Constant	6.41Vrms/krpm	6.41Vrms/krpm
Motor Line Resistance	0.54(1±5%)Ω	0.54(1±5%)Ω
Motor Line Inductance	0.34(1±10%)mH	0.34(1±10%)mH
Communication Mode	CAN	CAN
Communication Baud Rate	1Mbps	1Mbps
Communication Control Frequency	1kHz	1kHz
Motor Overtemperature Threshold	125°C	125°C
Driver Overtemperature Threshold	120°C	120°C
Overcurrent Threshold	28A	28A
Overvoltage Threshold	36V	36V
Undervoltage Threshold	18V	18V
Absolute Encoder Resolution	14bit	14bit
Joint Control Commands	Position, Velocity, Torque, Kp, Kd	
Joint Feedback Commands	Position, Velocity, Torque, Driver Temperature, Motor Temperature	

Mechanical Parameters	J60-6	J60-10
Gear Reducer Type	Planetary Reducer	Planetary Reducer

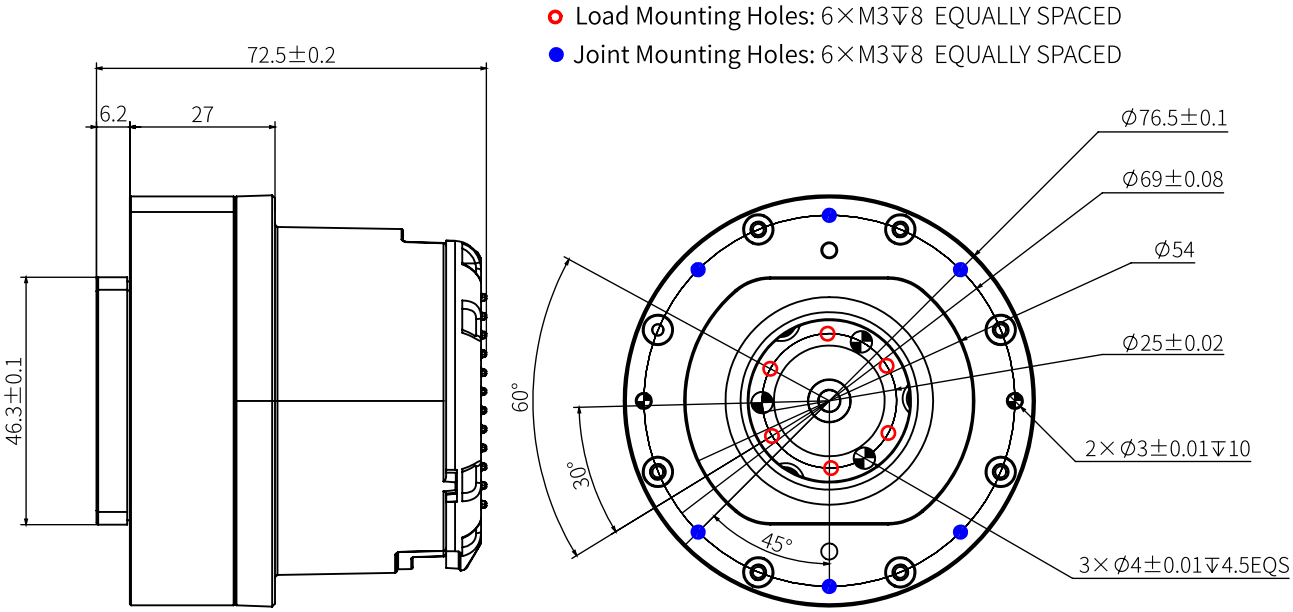
※The above data is measured in an ideal environment, and actual results may vary.

1.4 Installation Dimensions Diagram

1.4.1 J60-6



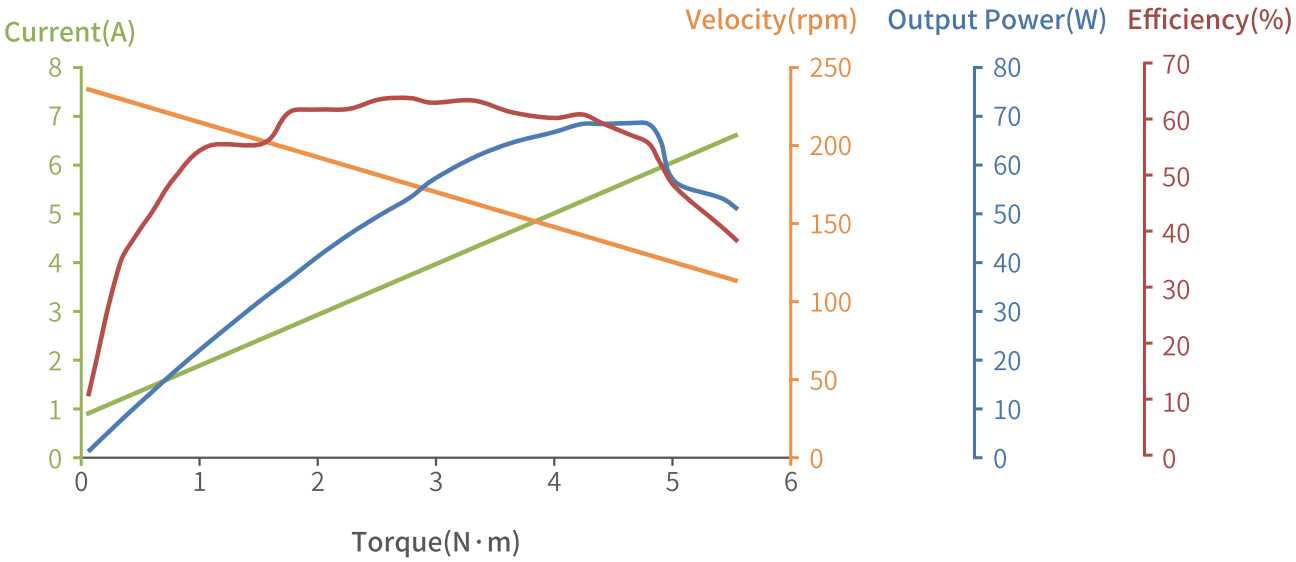
1.4.2 J60-10



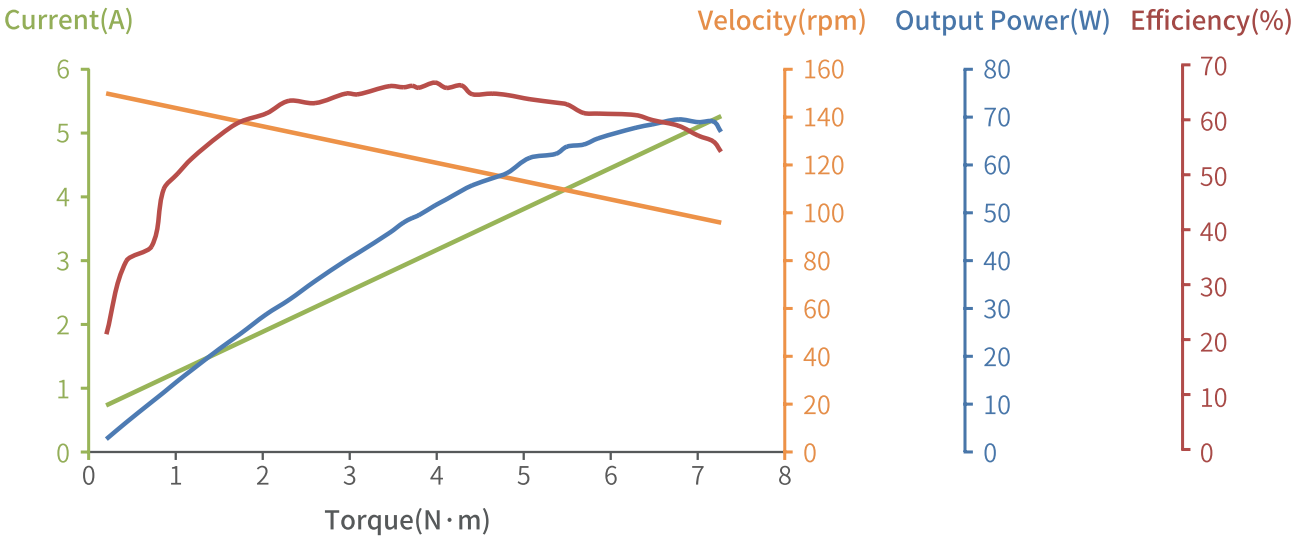
⚠	When installing loads or securing joints, please select screws of appropriate strength based on the specific operating conditions.

1.5 Performance Curve

Performance Curve of J60-6 at 24V

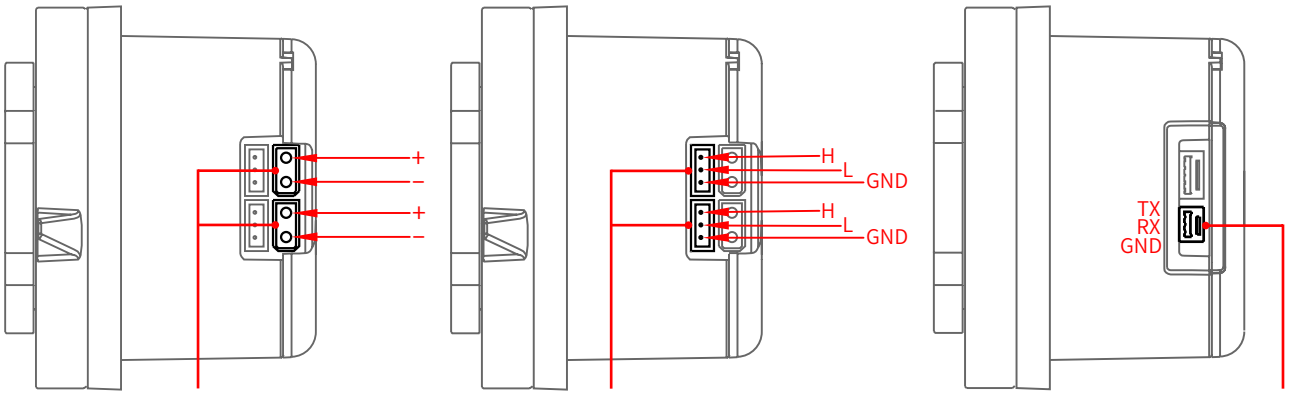


Performance Curve of J60-10 at 24V



2 Interfaces and Cables

2.1 Driver Pins and Interface



Board End Interface	Cable End Interface Model	Brand Manufacturer
① Power Supply	XT30UD-F	Amass
② CAN	50375033	MOLEX
③ TTL Serial	GHR-03V-S	JST

2.2 Accessories Kit

2.2.1 XT30 Power Cable

Order	Color	Pin on Driver	Electrode of Power Supply
1	Red	+	+
2	Black	-	-

2.2.2 CAN Bus Cable

Order	Color	Pin on Driver	Pin on USB to CAN Adapter
1	Yellow	CAN_H	CAN_H
2	Blue	CAN_L	CAN_L
3	Black	GND	GND

2.2.3 TTL Serial Cable

Order	Color	Pin on Driver	Pin on USB to TTL Adapter
1	White	TX	RX
2	Blue	RX	TX
3	Black	GND	GND

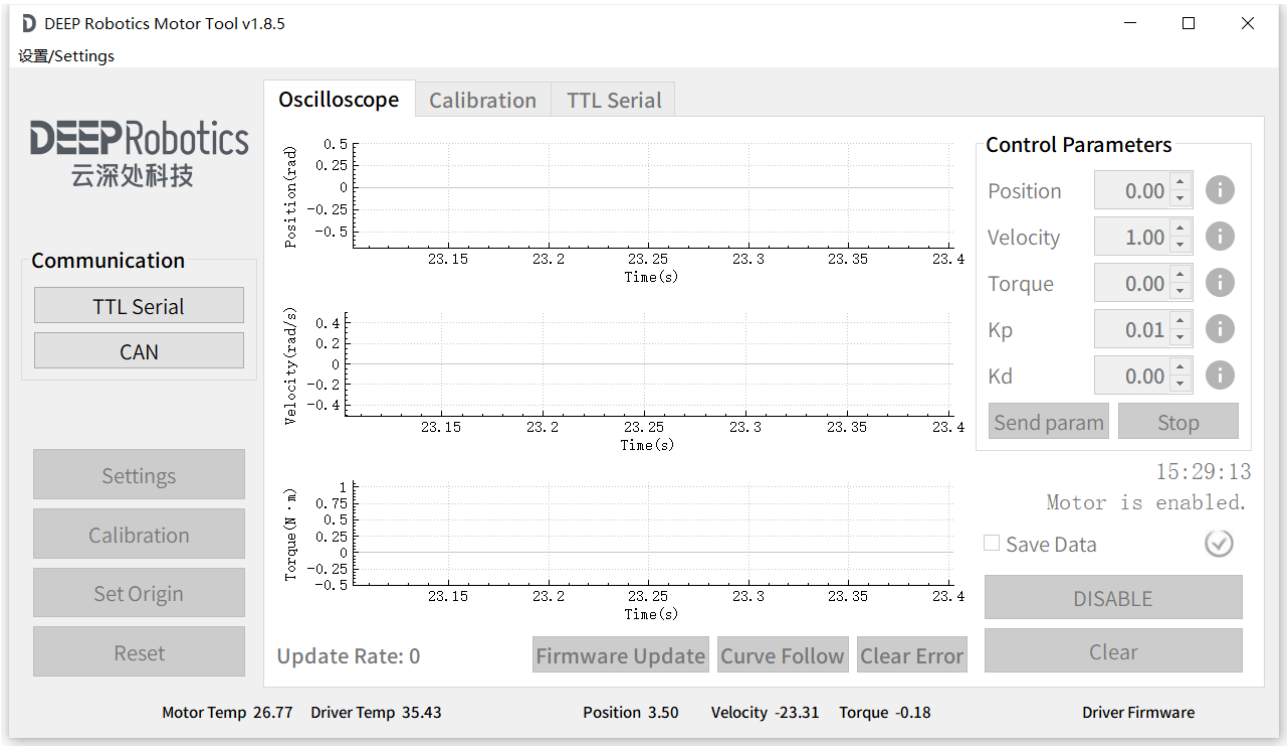
When using USB to TTL Adapter provided by DEEP Robotics to communicate with the joint, please connect as described in the table above. If you choose to use other USB to TTL Adapters, please test and confirm how to connect to the USB to TTL Adapter.

2.2.4 Adapters

Adapter	Model
USB to CAN Adapter	Makerbase CANable Pro
USB to TTL Adapter	CP2102

3 Debugger Software

3.1 Main Interface

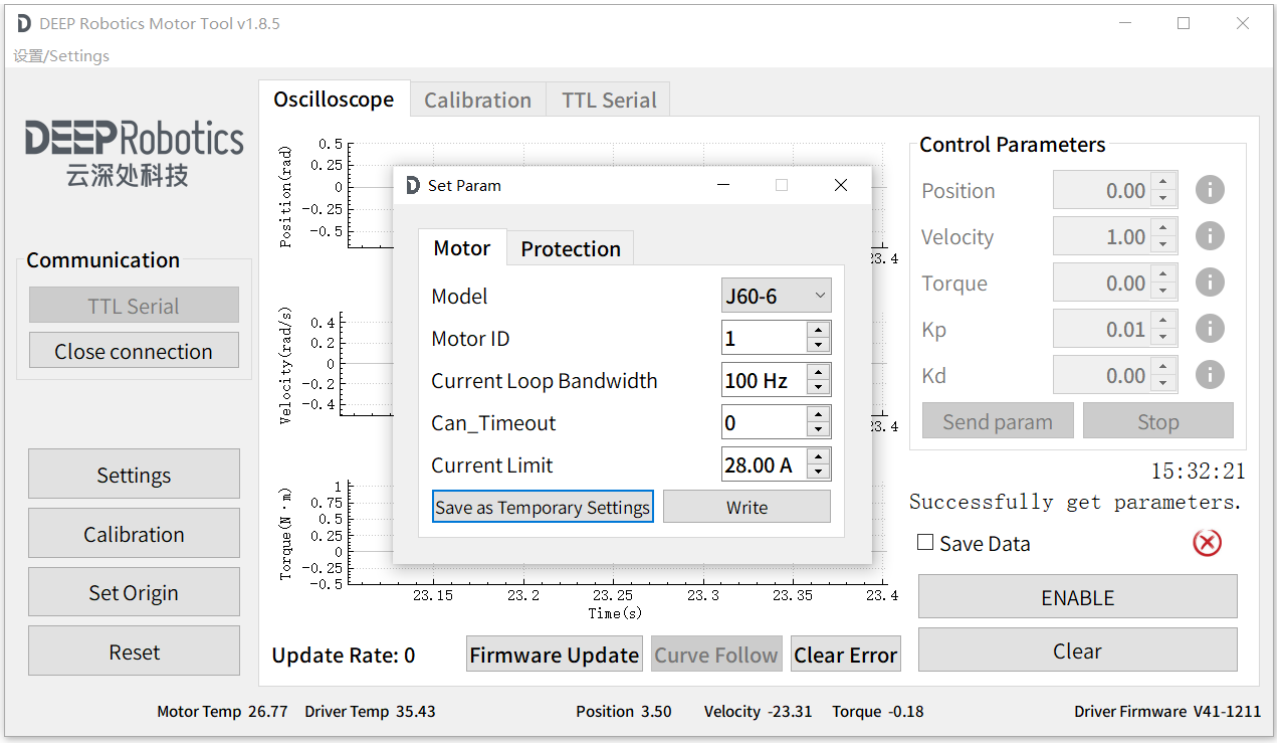


Function Module		Function Description
Communication		Establish and close TTL Serial or CAN connections
Basic Functions	Settings	Configure motor parameters and view protection thresholds (See 3.2 for details)
	Calibration	Calibrate the encoder error caused by transportation
	Set Origin	Set the current position of the motor as the origin
	Reset	Software reboot
	ENABLE/DISABLE	Activate or deactive the motor (See 3.3 for the status of ENABLE/DISABLE)
Oscilloscope	Control Parameters	Input and send desired control parameters (See 3.4 for details)
	Real-time Waveform	Display and clear waveforms
	Save Data	Record and save running data after next ENABLE
Status Bar		Real-time display of status and driver version (See 3.5 for details)

Function Module	Function Description
Exception Protection	Prompt users when an exception occurs and protect the joint

3.2 Settings

3.2.1 Motor Settings



Parameter	Description	Range
Model	Select the actual model of the joint currently connected	/
Motor ID	Set the ID number for this motor, which can distinguish the communication target when communicating via CAN	0~15
Current Loop Bandwidth	Set the current loop gain coefficient	0~2000
CAN_Timeout	Set timeout for CAN communication: if driver doesn't receive CAN frame within the timeout duration, protection will be triggered (actual timeout duration=CAN_Timeout×25μs)	/
Current Limit	Limit the target current issued by the driver to the motor (default value is 28A, and when exceeding, the current command is issued at 28A)	0~28

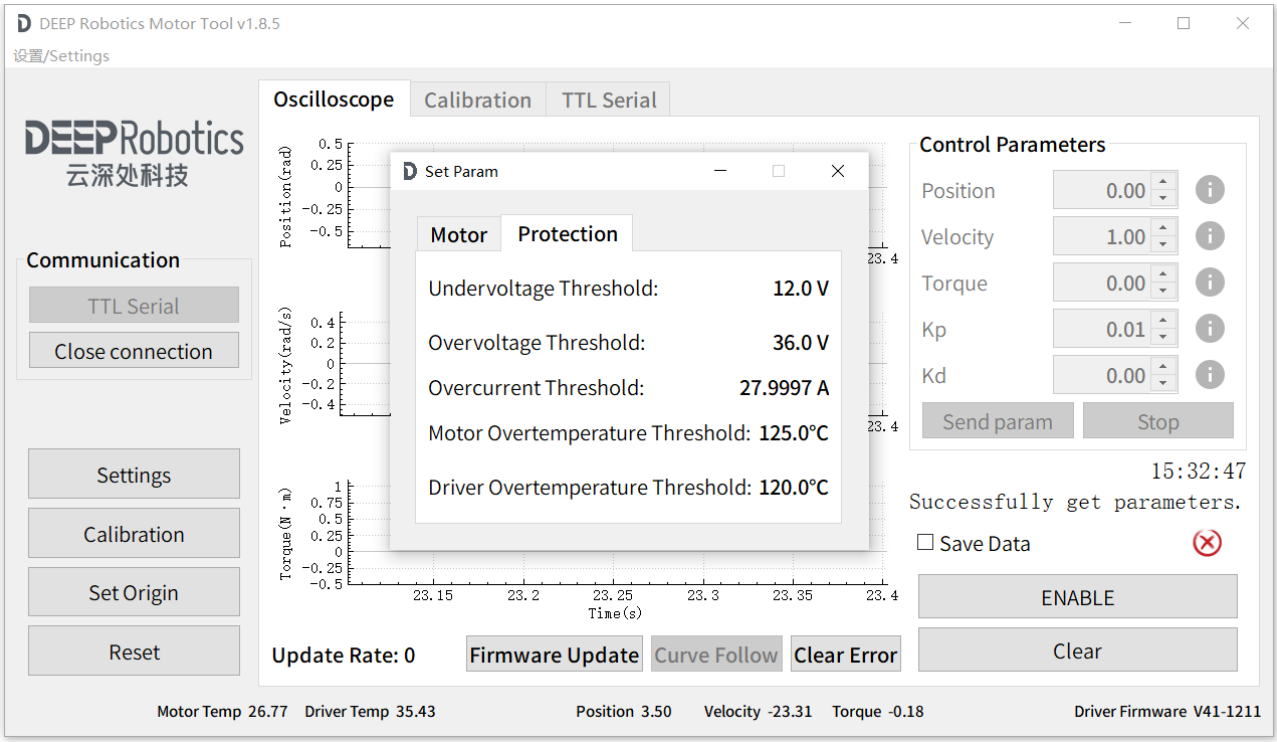
	If the current limit is set to the maximum value of 28A, it may trigger overtemperature protection during use.

There are two ways to save motor settings:

Save Mode	Description
Save as Temporary Settings	Temporary effective, and restore after restart
Write	Save to the driver, and still be effective after restart

3.2.2 Protection

Users can view the protection trigger thresholds, but can not modify them.



Parameter	Description
Undervoltage Threshold	When the operating voltage is lower than the undervoltage threshold, the motor will be disabled
Overvoltage Threshold	When the operating voltage is higher than the overvoltage threshold, the motor will be disabled
Overcurrent Threshold	Overcurrent threshold is determined by Current Limit in the motor settings
Motor Overtemperature Threshold	The motor will be disabled if the internal coil temperature of the motor exceeds the threshold

Parameter Name	Parameter Description
Driver Overtemperature Threshold	The motor will be disabled if the temperature of the driver MOS exceeds the threshold

3.3 ENABLE and DISABLE Status

The icon above the [ENABLE]/[DISABLE] button in the interface will display the motor operation status:

Icon	Meaning
	Motor is enabled
	Motor is disabled

3.4 Motion Parameters

Parameter	Unit	Data Type	Value Range
Position	rad	float	[-40.0,40.0]
Velocity	rad/s	float	[-40.0,40.0]
Torque	N·m	float	[-40.0,40.0]
Kp	/	float	[0,1023.0]
Kd	/	float	[0,51.0]

3.5 Status Bar

The status bar at the bottom of the interface will display the input voltage, motor temperature, driver temperature in real-time, as well as the real-time position, angular velocity, and torque (of output shaft), and the driver version currently used(the real-time input voltage is not displayed when communicating via CAN).

When using serial communication, the real-time angular velocity value displayed in the status bar at the bottom of the interface is 2π times the actual angular velocity value.

4 Debug & Use

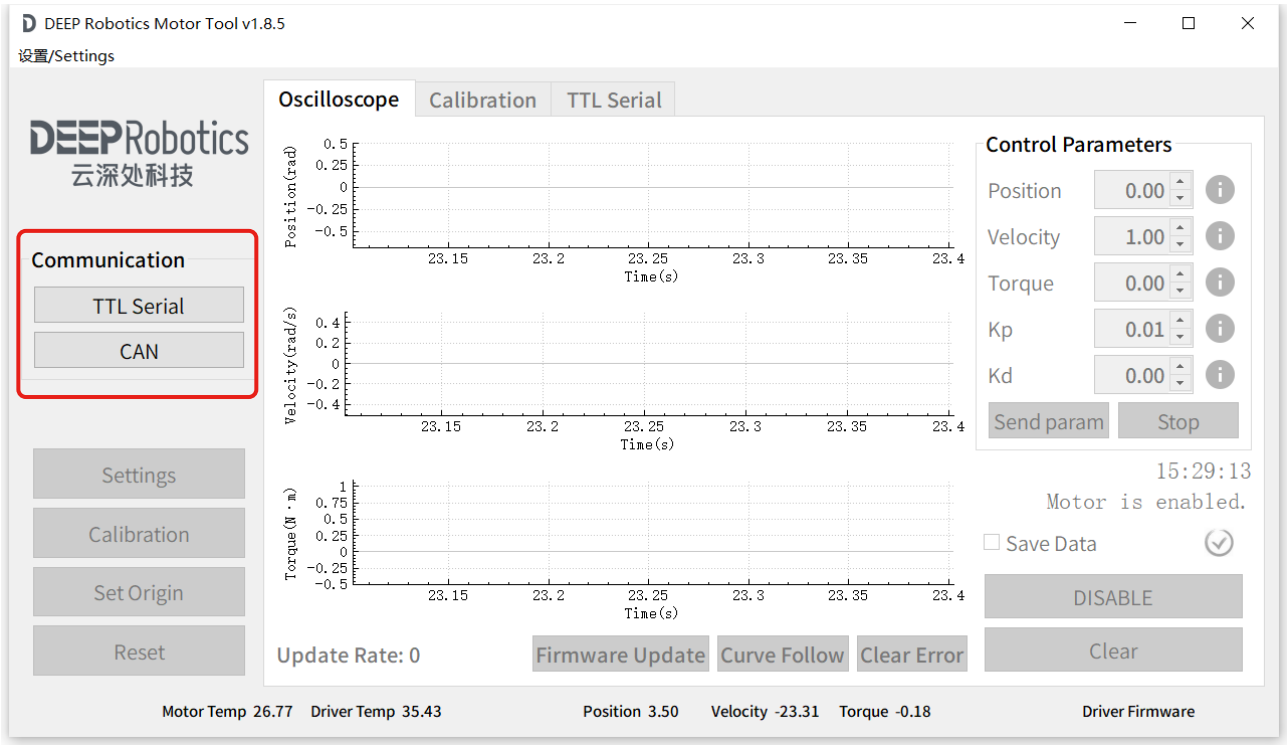
Users can use the official debugger software to test and debug the joint.
Before debugging, please use the XT30 Power Cable in the Accessories Kit to connect the power supply with the joint, and it is recommended to use a 24V stabilized power supply.
There are two communication modes for debugging the J60 joint: CAN Bus and TTL Serial.

	This section explains the debugging process on Windows. If using Ubuntu, you may need to enter the system password when connecting.

4.1 Communication Settings

4.1.1 TTL Serial

1. If debugging on Windows, you need to download and install the driver of USB to TTL Adapter.
2. According to "2 Interfaces and Cables", use the TTL Serial Cable to connect the USB to TTL Adapter with the joint.
3. Insert the USB to TTL Adapter into the USB port of the host with the debugger software installed, and open the software.
4. Power up the joint.
5. In the Communication settings, click the [TTL Serial] button, then select the serial port you need to connect to for the connection.

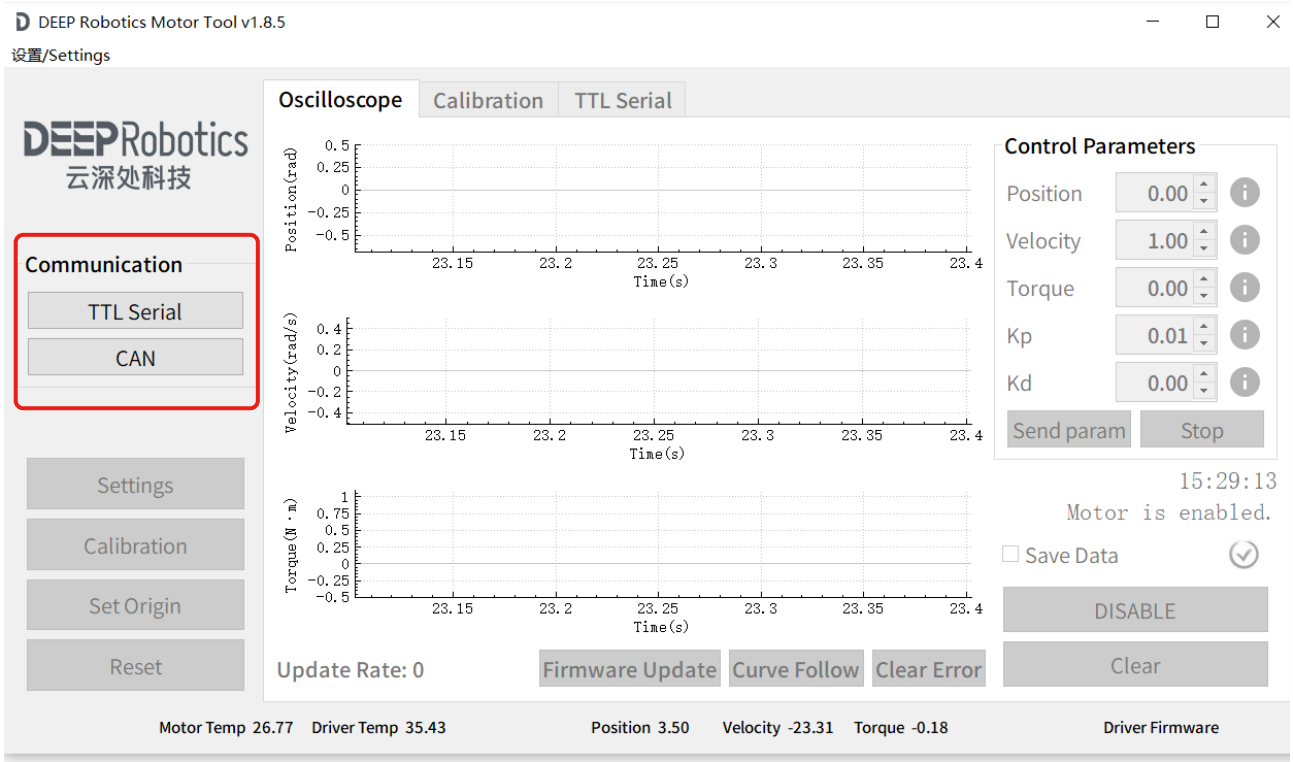


After establishing a successful connection, the input voltage, motor temperature, and driver temperature of the joint will be displayed in real-time in the status bar and the driver version will be displayed in the lower right corner. You can then debug and use the joint motor.

4.1.2 CAN

1. According to "2 Interfaces and Cables", use the CAN Bus Cable to connect the USB to CAN Adapter and the joint.
2. Connect the USB to CAN tool to the host with the joint debugging tool software installed via the Type-C data cable, and start the joint debugging tool software.
3. Power up the joint.
4. In the Communication settings, click the [CAN] button, then select the CAN Device and enter the Motor ID you need to connect to, and click to open the device.

	• The default ID for each motor is 1. If you need to modify it, please connect to the joint via the TTL Serial, change it in the motor settings, and save it. • When using debugger software for CAN communication, you need to set the Can_Timeout to 0 in the motor settings.



After a successful connection, the motor temperature and driver temperature of the joint will be displayed in real-time in the status bar, and the driver version will be displayed in the lower right corner. You can then debug and use the joint motor.

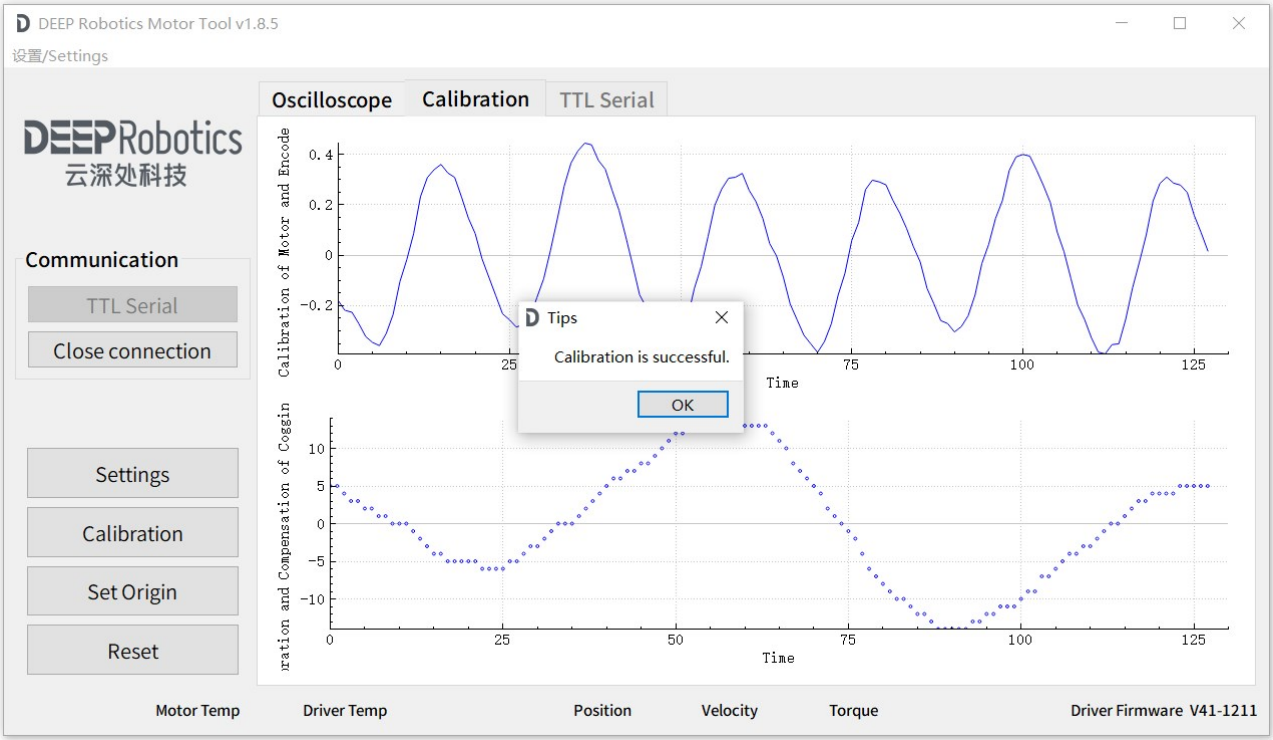
4.2 Parameters Settings

Click the [Settings] button to enter the settings window. Users can modify and save motor parameters according to the needs of debugging and development, and the definition of each parameter is referred to in 3.2.1.

	• Please modify the parameters when motor is disabled. • When using the debugger software to test the J60 joint for the first time, please select the correct model in the motor settings and click the [Write] button to save.

4.3 Calibration

Click the [Calibration] button to start the encoder calibration process, and the interface will display the calibration data and real-time waveform of the encoder compensation value.

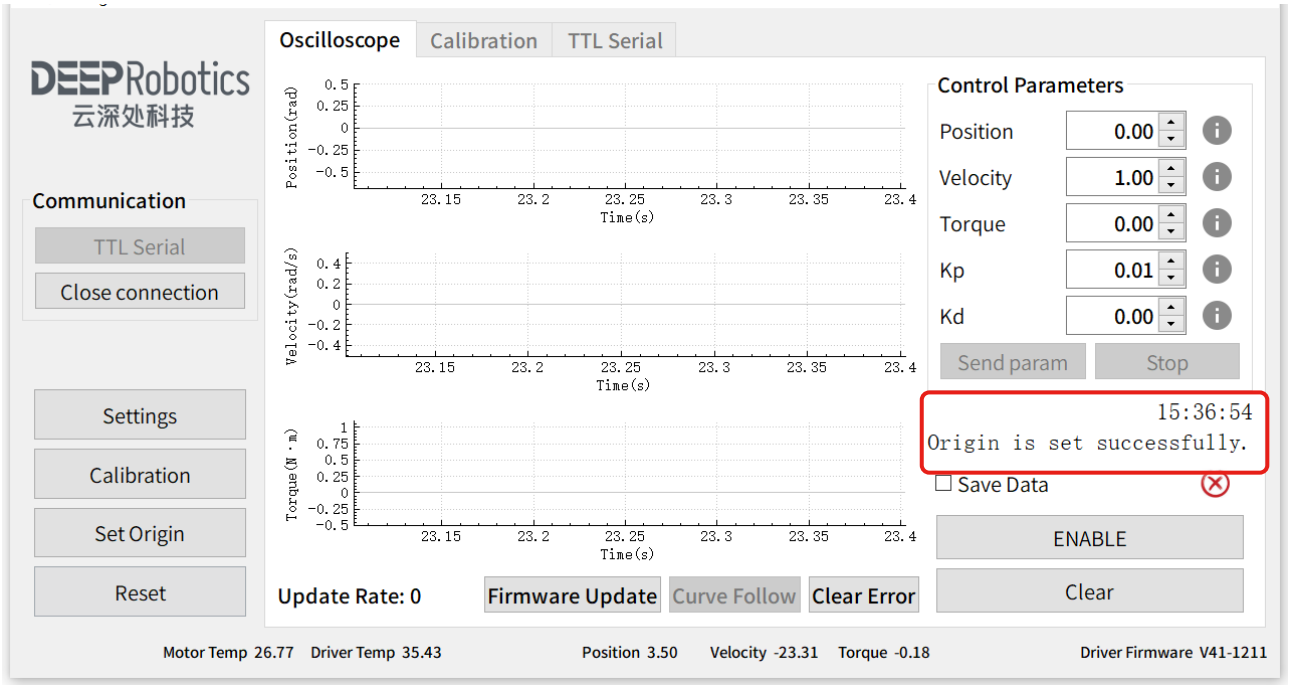


	• J60 joint has completed the encoder calibration before it left the factory, but long-distance transportation may cause encoder error. Users can calibrate the encoder when using the joint first time. • Users can use the calibration function to eliminate the encoder error caused by self-disassembly or modification, but warranty will be immediately invalidated after disassembly or modification. • If the encoder compensation value exceeds ±50 with Driver Firmware versions prior to V51, or surpasses ±200 in Driver Firmware version V51 or later, please contact after-sales service for troubleshooting. • Please perform calibration when motor is disabled.

4.4 Setting Origin

Users can set the origin position for the output shaft according to the needs of development and debugging when the motor is disabled.

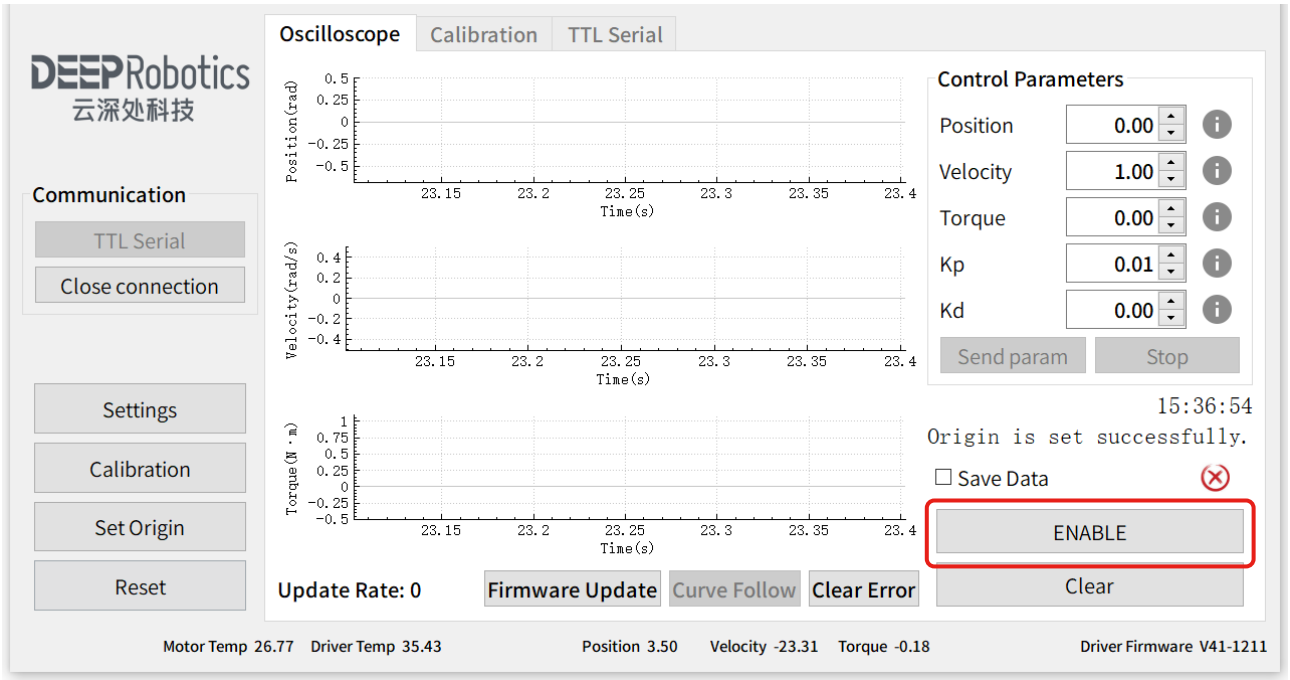
1. Rotate the output shaft to the position you want to set as origin.
2. Power off the joint and power it back on.
3. Click the [Set Origin] button to save the current position as the origin.



	J60 joint only has one single-turn absolute encoder at the motor end. After power off, the origin position of the output shaft will be lost.

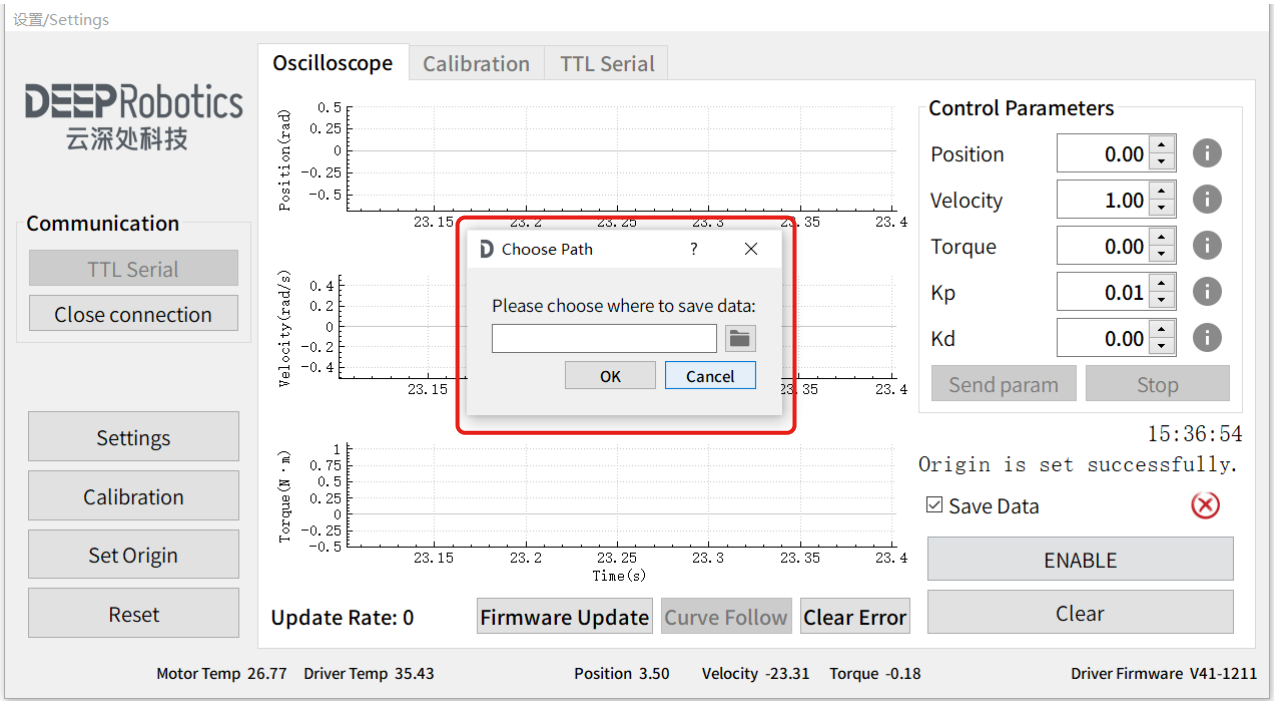
4.5 Motor ENABLE

Before starting the debugging, please click the [ENABLE] button to activate the motor, allowing it to execute the received motor control commands. If you need to save the data, you can tick the "Save Data" checkbox before enabling, see "4.6 Save Data" for details.



4.6 Save Data

Users can tick the "Save Data" checkbox before enabling the motor, and when clicking [ENABLE], please choose where to save the data in the pop-up window.



After enabling, the motion control data will be saved as a .csv file, named after the time that the motor is enabled, for example, "dr-motor-tool-log-2023-09-27-16-10-03.csv". Data saving stops after the motor is disabled.

4.7 Control Parameters

	TTL Serial communication performance is lower than CAN, so it is recommended to use CAN to send motion control parameters.

J60 joint provides 5 control parameters (of output shaft), including Position, Velocity, Torque, Kp, and Kd. The final joint target torque is:

$$T = kp \times \left(p_{des} - p_{fbk} \right) + kd \times \left(v_{des} - v_{fbk} \right) + \tau_{ff}$$

Where the parameters p_{fbk} and v_{fbk} represent the actual position and actual agular velocity of the joint, respectively.

The driver will convert the final joint target force into the desired current and perform closed-loop control at a frequency of 40kHz.

There are several control modes to control the motor. First, enable the motor, then enter the target parameters in the interface, click [Send param], and the motor will receive and operate.

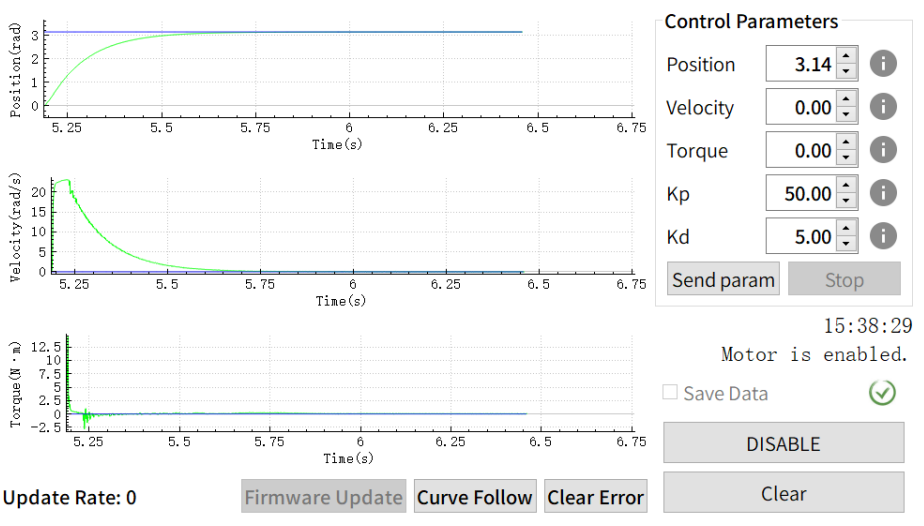
The control parameters for different control modes can refer to the cases introduced in this section.

4.7.1 Position Control Mode

Position Control Mode is a control method to make the output shaft of the motor stabilize at a specific position. For example, if the target position is 3.14 rad, you can send:

$$p_{des} = 3.14, v_{des} = 0, \tau_{ff} = 0, kp = 50, kd = 5$$

	The derivative gain kd is recommended to be given between 1.0 and 15.0, otherwise, it may easily cause the motor to oscillate continuously near the target position, or even lose control.



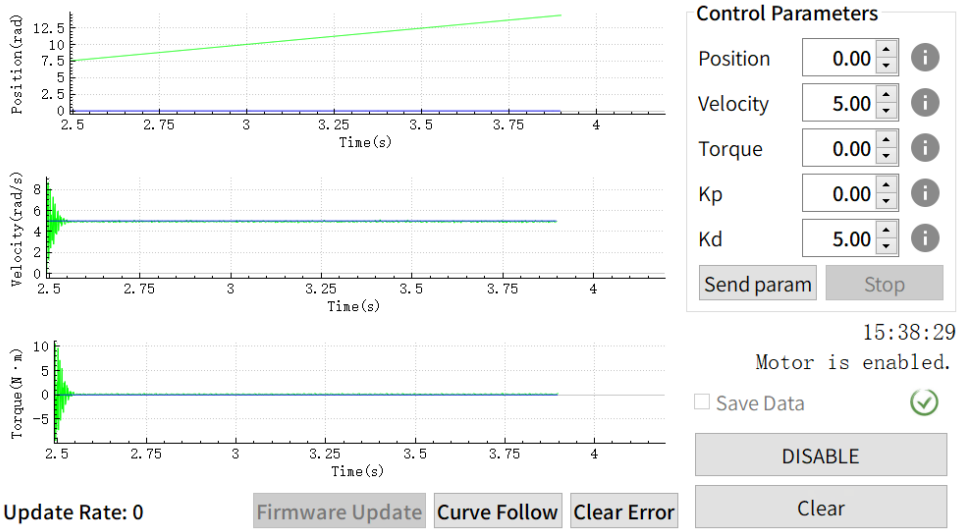
4.7.2 Velocity Control Mode

In Velocity Contorl Mode, the primary objective is to achieve a desired velocity setpoint. For

example, if desired angular velocity (of output shaft) v_{des} is 5 rad/s, you can send:

$$p_{des} = 0, v_{des} = 5, \tau_{ff} = 0, kp = 0, kd = 5$$

	The derivative gain kd is recommended to be given between 1.0 and 15.0, otherwise, it may easily cause the motor to oscillate continuously near the target velocity, or even lose control.

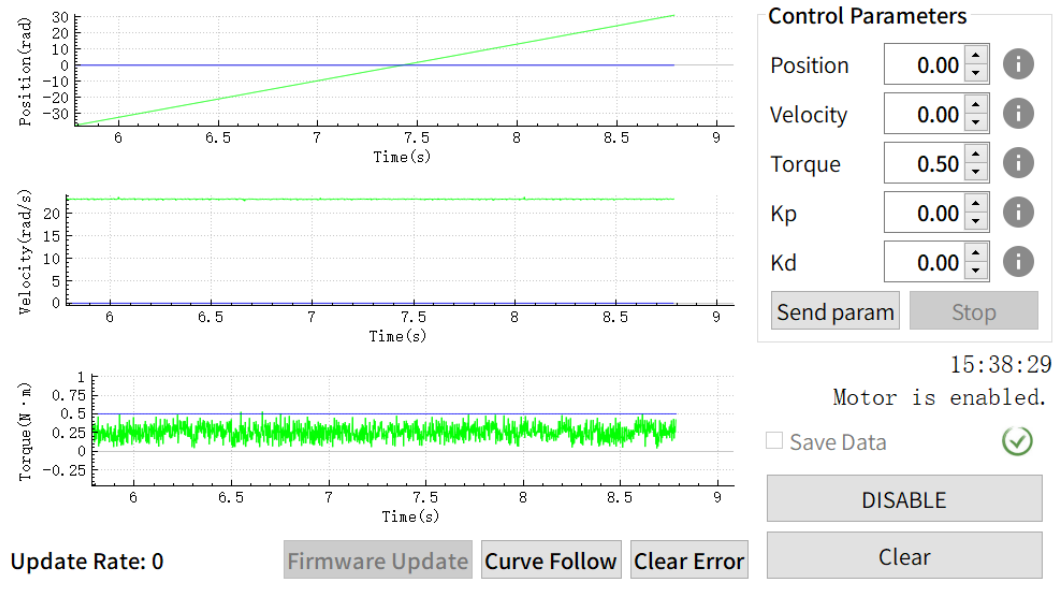


4.7.3 Torque Control Mode

Torque Control Mode aims to regulate the torque of output shaft. For example, if the feedforward torque is 0.5 N·m, you can send:

$$p_{des} = 0, v_{des} = 0, \tau_{ff} = 0.5, kp = 0, kd = 0$$

	Under no-load conditions, even giving a small feedforward torque will cause the motor to accelerate to the maximum speed and rotate.



4.7.4 Damping Mode

Damping mode is usually used to control the motor's deceleration, braking, or damping operation. In damping mode, the goal of the motor is to produce a resistance, braking, or deceleration effect by applying an appropriate motor output. By controlling the motor output, you can adjust the damping force to achieve expected deceleration effect. For example, given kd is not 0, and the rest of the control parameters are set to 0:

$$p_{des} = 0, v_{des} = 0, \tau_{ff} = 0, kp = 0, kd = 5$$

After sending the parameters, the motor will not rotate on its own. If the motor is driven to rotate by an external force, it will generate a resisting torque. The direction of the resisting torque is opposite to the direction of the rotation driven by the external force, and the resisting torque is proportional to the rotation speed.

4.7.5 Zero Torque Mode

In zero-torque mode, the motor aims to maintain the output torque at zero or close to zero. It actively generates torque to resist its own frictional torque. Set all parameters to 0:

$$p_{des} = 0, v_{des} = 0, \tau_{ff} = 0, kp = 0, kd = 0$$

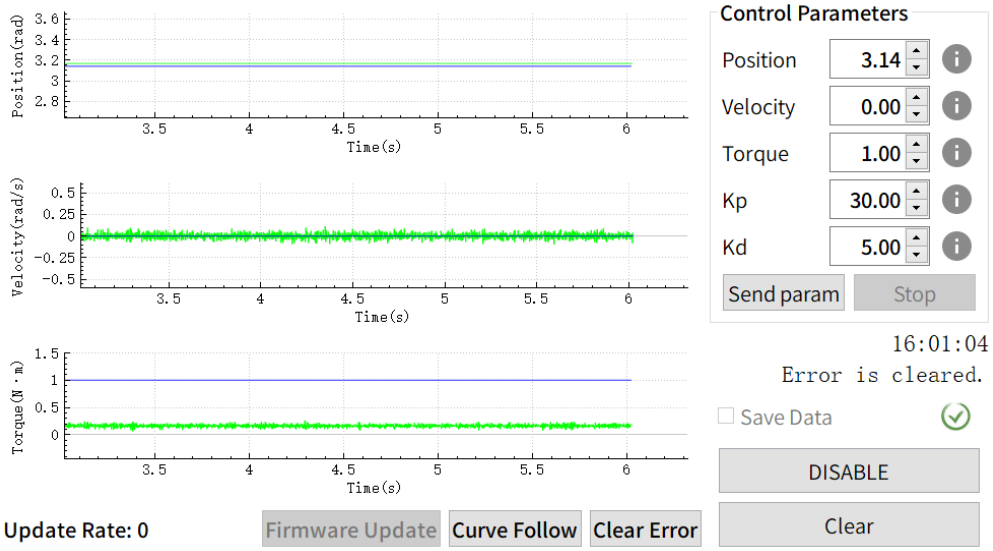
After sending the motion parameters, the motor will not rotate on its own. If you try to rotate the output shaft, you will feel that the resistance of the output shaft is significantly less than the resistance before the motor is powered up.

4.7.6 Hybrid Torque-Position Mode

In the actual control of quadruped robots, the upper controller often sends the desired position, desired angular velocity, and feedforward torque to the motor at the same time. This is Hybrid Torque-Position Control.

$$p_{des} = 3.14, v_{des} = 0, \tau_{ff} = 1, kp = 30, kd = 5$$

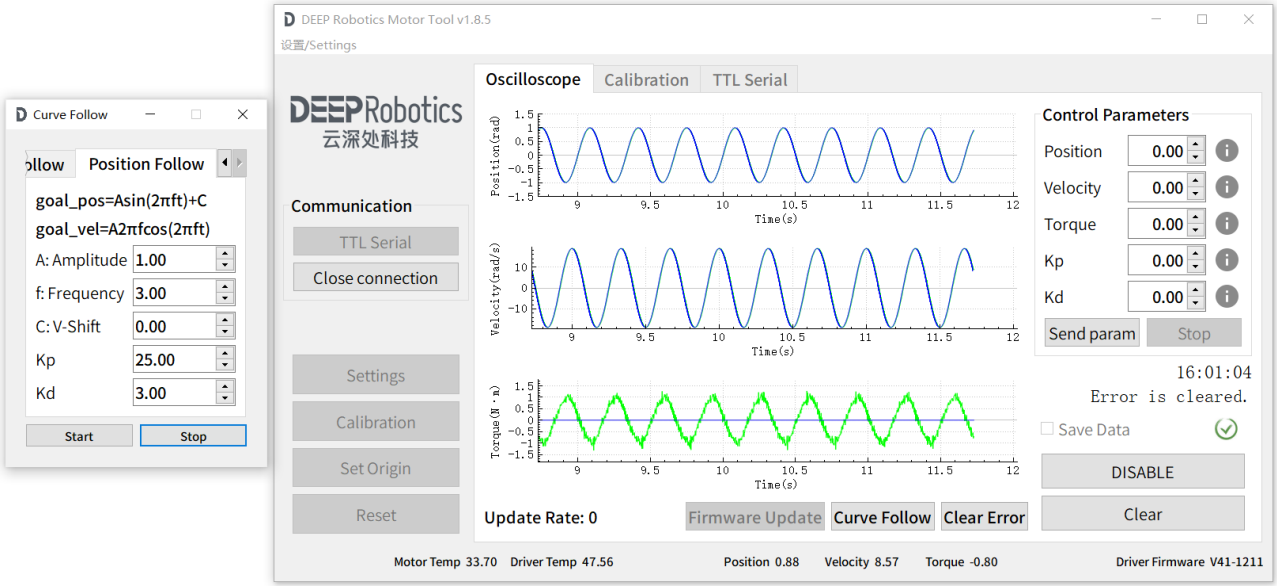
	Please be sure to fix the joint in Hybrid Torque-Position Control Mode.



4.8 Curve Follow

When communicating with the joint via CAN Bus, the debugger software supports sending target control parameters that vary in real-time according to a sinusoidal or cosinoidal curve, thereby testing the control performance of the motor.

	Please use the Curve Follow function after the motor is enabled.



4.9 View and Clear Waveforms

When the motor is rotating, the debugger software will display the real-time waveforms of position, angular velocity, and torque. After the motor is disabled, the waveforms stop generating. Dragging the mouse in the waveform box can move the time axis forward and backward. Rolling the scroll wheel in the waveform box can zoom in and out. After the waveforms stop generating, click the [Clear] button to clear the generated waveform curves in the waveform box.

4.10 Motor DISABLE

After disconnecting and powering off, please first click the [DISABLE] button to disable the motor, changing the motor from the operating state to the stopped state. At this time, the motor will no longer respond to the control signals, and then you can disconnect the power supply to the joint.

4.11 Reset

Click the [Reset] button to perform a software reboot of microcontroller.

4.12 Firmware Update

If you need to update the firmware, you can click the [Firmware Update] button when using CAN communication, select the downloaded firmware package for installation and update.

4.13 Clear Error

After an error occurs, first restore it to normal state and then click the [Clear Error] button to continue use.

4.14 Exception Handling

4.14.1 Temperature Abnormality

When the motor temperature or driver temperature in the status bar is displayed in red, and prompt text appears on the right side of the window, it indicates that the motor or driver temperature has reached the overtemperature protection threshold. At this time, the motor will automatically be disabled, and users should wait for it to cool down and click [Clear Error] before continuing to use it.

4.14.2 Voltage Abnormality

When the voltage in the status bar is displayed in red and prompt text appears on the right side of the window, it indicates that the voltage is lower than the under-voltage threshold or higher than the over-voltage threshold. The motor will automatically be disabled. Please check whether the power supply voltage is normal. After the power supply voltage is restored to between 12V and 36V, you can click [Clear Error] and continue to use it.

4.14.3 Communication Abnormality

When the CAN communication times out, the motor will be disabled.

4.15 Language Settings

Users can click the [Settings] button to change the display language.

5 CAN Communication Protocol

After setting parameters, calibrating, and setting origin for the motor, you can use the CAN communication protocol to control the motor.

The driver stores data in little-endian byte order. CAN frame uses the standard frame format, and the baud rate is 1Mbps. CAN data frame consists of CAN ID, CAN DLC, and CAN DATA:

1. CAN ID (11 Bits): Bit0~3 is the motor ID (default is 1, which can be modified according to 4.2), Bit4 is the flag bit to determine the direction of transmission, and Bit5~10 is the command code used to specify the operation that this CAN frame is intended to perform.
2. CAN DLC (4 Bits): The number of bytes in CAN DATA. Different commands carry CAN DATA of different lengths.
3. CAN DATA (8 Bytes): Data carried by the frame. Different commands carry different data.

5.1 Disable Motor

CAN_CMD_MOTOR_DISABLE sent to motor		
CAN ID	Bit0~Bit4	Target Motor ID
	Bit5~Bit10	1
CAN DLC	Bit0~Bit3	0
CAN DATA	Byte0~Byte7	/

CAN_CMD_MOTOR_DISABLE received from motor		
CAN ID	Bit0~Bit4	Motor ID+0x10
	Bit5~Bit10	1
CAN DLC	Bit0~Bit3	1
CAN DATA	Byte0	0 = successful, 1 = failed
	Byte1~Byte7	/

5.2 Enable Motor

CAN_CMD_MOTOR_ENABLE sent to motor		
CAN ID	Bit0~Bit4	Target Motor ID
	Bit5~Bit10	2

CAN_CMD_MOTOR_ENABLE sent to motor		
CAN DLC	Bit0~Bit3	0
CAN DATA	Byte0~Byte7	/

CAN_CMD_MOTOR_ENABLE received from motor		
CAN ID	Bit0~Bit4	Motor ID+0x10
	Bit5~Bit10	2
CAN DLC	Bit0~Bit3	1
CAN DATA	Byte0	0 = successful, 1 = failed
	Byte1~Byte7	/

5.3 Control Motor

CAN_CMD_MOTOR_CONTROL sent to motor		
CAN ID	Bit0~Bit4	Target Motor ID
	Bit5~Bit10	4
CAN DLC	Bit0~Bit3	8
CAN DATA	Bit0~Bit15	Target Position
	Bit16~Bit29	Target Velocity
	Bit30~Bit39	Kp
	Bit40~Bit47	Kd
	Bit48~Bit63	Target Torque

CAN_CMD_MOTOR_CONTROL received from motor		
CAN ID	Bit0~Bit4	Motor ID+0x10
	Bit5~Bit10	4
CAN DLC	Bit0~Bit3	8

CAN_CMD_MOTOR_CONTROL received from motor		
CAN DATA	Bit0~Bit19	Current Position
	Bit20~Bit39	Current Velocity
	Bit40~Bit55	Current Torque
	Bit56	Temperature Flag
	Bit57~Bit63	Current Temperature

The parameters involved in this set of commands are defined as follows:

Parameter	Parameter Meaning	Value Range
Target Position	Target position value mapped proportionally from [-40rad,40rad] to [0,65535]	[0,65535]
Target Velocity	Target velocity value mapped proportionally from [-40rad/s,40rad/s] to [0,16383]	[0,16383]
Kp	Kp value mapped proportionally from [0,1023] to [0,1023]	[0,1023]
Kd	Kd value mapped proportionally from [0.0,51.0] to [0,255]	[0,255]
Target Torque	Target torque value mapped proportionally from [-40N·m,40N·m] to [0,65535]	[0,65535]
Current Position	The current position mapped proportionally from [-40rad,40rad] to [0,1048575]	[0,1048575]
Current Velocity	The current velocity value mapped proportionally from [-40rad/s,40rad/s] to [0,1048575]	[0,1048575]
Current Torque	The current torque value mapped proportionally from [-40N·m,40N·m] to [0,65535]	[0,65535]
Temperature Flag	0 indicates Bit57~63 is the MOSFET temperature 1 indicates Bit57~63 is the motor temperature	/
Current Temperature	The current temperature mapped proportionally from [-20°C,200°C] to [0,127]	[0,127]

During development or debugging, the target control parameters sent to motor and the real-time status received from motor via debugger software or CAN Bus are both related to output shaft.

5.4 Get Current Configuration

CAN_CMD_GET_CONFIG sent to motor		
CAN ID	Bit0~Bit4	Target Motor ID
	Bit5~Bit10	24
CAN DLC	Bit0~Bit3	0
CAN DATA	Byte0~Byte7	/

CAN_CMD_GET_CONFIG received from motor		
CAN ID	Bit0~Bit4	Motor ID+0x10
	Bit5~Bit10	24
CAN DLC	Bit0~Bit3	8
CAN DATA	Byte0~Byte1	Reserved Parameter (internal use only)
	Byte2	Motor ID
	Byte3	CAN Timeout Interval (0~255ms)
	Byte4~Byte5	Current Loop Bandwidth (0~65535Hz)
	Byte6~Byte7	Current Limit mapped proportionally from [0A,40A]to [0,65535]

5.5 Change CAN Timeout

	After sending CAN_CMD_SET_CAN_TIMEOUT, the new configuration will take effect temporarily and restored after restart. If you want to make it effective even after restart, please send CAN_CMD_SAVE_CONFIG to save configuration to the driver.

CAN_CMD_SET_CAN_TIMEOUT sent to motor		
CAN ID	Bit0~Bit4	Target Motor ID
	Bit5~Bit10	9
CAN DLC	Bit0~Bit3	1
CAN DATA	Byte0	CAN Timeout Interval (0~255ms)
	Byte1~Byte7	/

CAN_CMD_SET_CAN_TIMEOUT received from motor		
CAN ID	Bit0~Bit4	Motor ID+0x10
	Bit5~Bit10	9
CAN DLC	Bit0~Bit3	1
CAN DATA	Byte0	0 = successful, 1 = failed
	Byte1~Byte7	/

5.6 Change Current Loop Bandwidth

CAN_CMD_SET_BANDWIDTH sent to motor		
CAN ID	Bit0~Bit4	Target Motor ID
	Bit5~Bit10	10
CAN DLC	Bit0~Bit3	2
CAN DATA	Byte0~Byte1	Current Loop Bandwidth (0~65535Hz)
	Byte2~Byte7	/

CAN_CMD_SET_BANDWIDTH received from motor		
CAN ID	Bit0~Bit4	Motor ID+0x10
	Bit5~Bit10	10
CAN DLC	Bit0~Bit3	1
CAN DATA	Byte0	0 = successful, 1 = failed
	Byte1~Byte7	/

After sending CAN_CMD_SET_BANDWIDTH, the new configuration will take effect temporarily and restored after restart. If you want to make it effective even after restart, please send CAN_CMD_SAVE_CONFIG to save configuration to the driver.

5.7 Save Configuration

CAN_CMD_SAVE_CONFIG sent to motor		
CAN ID	Bit0~Bit4	Target Motor ID
	Bit5~Bit10	16
CAN DLC	Bit0~Bit3	0
CAN DATA	Byte0~Byte7	/

CAN_CMD_SAVE_CONFIG received from motor		
CAN ID	Bit0~Bit4	Motor ID+0x10
	Bit5~Bit10	16
CAN DLC	Bit0~Bit3	1
CAN DATA	Byte0	0 = successful, 1 = failed
	Byte1~Byte7	/

5.8 Get Error Code

In the messages returned back, every bit of CAN DATA stands for one kind of error: if the error does not occurs, bit = 0; or bit = 1.

CAN_CMD_GET_STATUSWORD sent to motor		
CAN ID	Bit0~Bit4	Target Motor ID
	Bit5~Bit10	23
CAN DLC	Bit0~Bit3	0
CAN DATA	Byte0~Byte7	/

CAN_CMD_GET_STATUSWORD received from motor		
CAN ID	Bit0~Bit4	Motor ID+0x10
	Bit5~Bit10	23

CAN_CMD_GET_STATUSWORD received from motor		
CAN DLC	Bit0~Bit3	2
CAN DATA	Byte0~Byte1	Bit0 = 1: Overvoltage Bit1 = 1: Undervoltage Bit2 = 1: Overcurrent Bit3 = 1: Motor Overtemperature Bit4 = 1: Driver Overtemperature Bit5 = 1: CAN Timeout Bit6 = 1: Temperature Sensor Disconnection Bit7~Bit15: Reserved
	Byte2~Byte7	/

5.9 Clear Error

If some errors in CAN_CMD_GET_STATUSWORD occurred, please troubleshooting first and then clear errors to continue use.

CAN_CMD_ERROR_RESET sent to motor		
CAN ID	Bit0~Bit4	Target Motor ID
	Bit5~Bit10	17
CAN DLC	Bit0~Bit3	0
CAN DATA	Byte0~Byte7	/

CAN_CMD_ERROR_RESET received from motor		
CAN ID	Bit0~Bit4	Motor ID+0x10
	Bit5~Bit10	17
CAN DLC	Bit0~Bit3	1
CAN DATA	Byte0	0 = successful, 1 = failed
	Byte1~Byte7	/

6 Precautions

	• After the motor is powered off, please do not rotate the motor, as this may cause motor damage (when the motor is rotated by external forces at a speed exceeding the rated speed). • Please do not disassemble the joint without authorization, as it will void the warranty! • When debugging, please secure the joint to prevent it from moving or falling during motor rotation.

7 Warranty policy

7.1 Warranty Period

The warranty period of this product is 6 months. The warranty period starts from the date of receipt. Products or parts that meet the warranty policy will receive free after-sales service. If the product you purchased is beyond the warranty period, you can also get help from us by purchasing a separate service.

7.2 Warranty Coverage

Depending on the specific situation, we will repair or replace parts accordingly for the product you purchased. However, the following cases will not be covered by the free warranty, but you can still choose to have paid after-sales service, for which please consult the after-sales support for details.

- Damage caused by man-made problems but not by quality problems of the product itself occurs.
- Private modification, disassembly or opening of the shell occurs.
- Damage caused by incorrect installation, use and operation in accordance with the manual.
- Damage caused by using beyond the maximum torque.
- Damage caused by self-installation of third party products.
- Failure or damage due to force majeure factors such as typhoon, earthquake, fire, lightning strike, abnormal voltage, etc.

7.3 Repair Instructions

- Before getting after-sales service, please make sure to backup all data and delete important data to prevent data loss or leakage. DEEP Robotics is not responsible for the loss or leakage of any data contained in the product.
- When you obtain after-sales service from DEEP Robotics, you authorize DEEP Robotics to make any modification, delete data or restore factory settings for the purpose of after-sales service.
- Before sending it for repair, please contact after-sales support, DEEP Robotics will try to diagnose and solve your problem remotely.
- If the above methods cannot solve your problem, you can send back the product for repair after verifying with after-sales support. You need to pay for the postage first when you send the product to DEEP Robotics. After DEEP Robotics receives the product in need of maintenance, the product will be tested to determine the problem and responsibility.
- If the problem is caused by defects in quality of the product itself, DEEP Robotics will be responsible for the testing fee, material fee, labor fee and the postage for sending back.

- If the product does not meet the conditions of free repair, you can choose to pay for repair, and the corresponding testing fee, material fee, labor fee and the postage for sending back will be paid by you. You can also choose not to repair and to send back the product, the corresponding postage and insurance fee will be paid by you.
- If you provide an incorrect delivery address which results in non-delivery or rejection by the recipient, the adverse consequences and losses shall be afforded by you.
- To ensure your rights and interests, when you sign for the after-sale products sent by DEEP Robotics, please check carefully whether the products are intact. If there is any abnormality, please immediately take video or photos on the spot and contact DEEP Robotics to get the solution. If there are unresolved after-sale problems, please also contact DEEP Robotics immediately, otherwise it is regarded as the end of this after-sale service without dispute.

※The final interpretation of these after-sales terms and conditions belongs to DEEP Robotics.
※Please contact us if you have any questions before obtaining after-sales service.
※These after-sales terms and conditions are only used in mainland China, and the after-sales policies of other countries or regions are subject to local laws.

DEEPRobotics

Hangzhou Yunshenchu Technology Co.,Ltd.

ADD: Building 3, Zijin Dream Plaza, No.36 Xinran Street, Sandun Town,
Xihu District, Hangzhou, Zhejiang, China
TEL: 400-0559-095
WEB: www.deeprobotics.cn