DAMIAO | 达妙科技

DM-J6248P-2EC Geared Motor

User Manual V1.0 2025.05.29

Disclaimer

Thank you for purchasing the DAMIAO Technology DM-J6248P-2EC geared motor (hereinafter referred to as the "motor"). Before using this product, please read carefully and follow this document and all safety guidelines provided by DAMIAO Technology. Failure to do so may cause harm to you and others, damage this product, or damage surrounding objects.

By using this product, you are deemed to have carefully read this document, understood, acknowledged, and accepted all terms and content of this document and all related documentation of this product. You agree to use this product only for legitimate purposes. You accept full responsibility for the use of this product and any consequences that may arise. DAMIAO Technology shall not be liable for any damage, injury, or legal responsibility resulting from direct or indirect use of this product.

Precautions

1. Please strictly adhere to the specified operating environment and the maximum allowable temperature range of the windings when using the motor. Otherwise, permanent and irreversible damage may occur to the product.

2. Avoid foreign objects entering the rotor, as this may cause abnormal operation of the rotor.

3. Before use, please check whether all components are intact. If any parts are missing, aged, or damaged, discontinue use.

4. Ensure correct wiring and that the motor is installed correctly and securely.

5. Do not touch the electronic rotor during operation to prevent accidents. The motor may generate heat during high-torque output; please be cautious to avoid burns.

6.Users are not permitted to disassemble the motor privately, as this may affect the motor's control accuracy and even cause abnormal operation.

Motor Features

1. Dual encoders provide absolute single-turn position output from the shaft, preventing loss of absolute position in the event of power loss.
2. Integrated design of motor and driver, compact structure, and high integration.
3. Supports PC-based visual debugging and CAN bus parameter reading/writing.
4. Supports CAN FD with a maximum baud rate of 5 Mbps.
5. Can feedback motor speed, position, torque, motor temperature, and other information via the CAN bus.
6. Equipped with dual-temperature protection functionality.
7. Supports firmware upgrades via CAN bus.

Specifications

Please use the motor appropriately according to the following parameters.

Motor Parameters	Rated Voltage	24V (supports 24-48V)
	Rated Phase/Supply Current	11.3A/7A@24V; 11.3A/3.7A @48V
	Peak Phase/Supply Current	38.7A/16A@24V； 38.7A/8A@48V；
	Rated Torque	30NM
	Peak Torque	97NM
	Rated Speed	40rpm
	Maximum No-load Speed	60rpm
Motor Characteristics	Reduction Ratio	48: 1
	Pole Pairs	14
	Phase Inductance	85uh(@25℃)
	Phase Resistance	0.17Ω(@25℃)
Structure & Weight	Outer Diameter	76mm
	Height	62.5mm
	Motor Weight: Approx	628g
Encoder	Encoder Resolution	16-bit
	Number of Encoders	2
	Encoder Type	Magnetic (Single-turn)
Communication	Control Interface	CAN@5Mbps(Max)
	Parameter Tuning Interface	UART@921600bps
Control & Protection	Control Modes	MIT Mode
		Speed Mode
		Position Mode
		Torque/Position Hybrid Mode
		Drive Over-temperature Protection, Threshold: 120°C. Motor will exit "Enable Mode" upon over-temperature.
		Motor Over-temperature Protection,

	Protections	user-configurable. Recommended setting not exceeding 100°C. Motor will exit "Enable Mode" upon over-temperature.
		Motor Over-voltage Protection, user-configurable. Recommended setting not exceeding 65V. Motor will exit "Enable Mode" upon over-voltage.
		Communication Loss Protection. Motor will automatically exit "Enable Mode" if no CAN command is received within the set period.
		Motor Over-current Protection, user-configurable. Recommended setting not exceeding 0.98. Motor will exit "Enable Mode" upon over-current.
		Motor Under-voltage Protection. Motor will exit "Enable Mode" if the supply voltage falls below the set value. The supply voltage should not be lower than 20V.

Operating Voltage

The driver operating voltage range is 24V to 48V. Hot-plugging is not recommended when the voltage exceeds 36V. The minimum operating voltage is 20V, and the maximum operating voltage is 65V.

Maximum Phase Current

The maximum phase current of the corresponding driver can be queried via the serial port print information during power-on:

The maximum phase current during operation can be limited by setting a percentage value via the debugging assistant. The default value is 0.8, which corresponds to 80% of the maximum current that can be sampled. It is recommended that this value does not exceed 98%.

Maximum Speed

The maximum speed is limited by various factors, including the power supply voltage (V_BUS), the magnetic flux linkage value (ψf), and the gear reduction ratio (GR). An upper limit can typically be calculated using the following formula.

$$V_{MAX}(rad/s) = 0.57735 * \frac{V_{BUS}}{N_{pp} \times GR \times \psi_f}(rad/s)$$

where:

V_BUS is the power supply voltage

Npp is the number of motor pole pairs,

ψf represents the rotor flux linkage

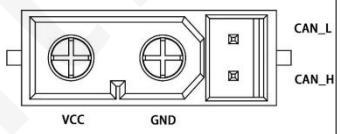
Package Contents

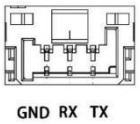
1. Motor (with integrated driver) × 1

2. Power (with CAN communication terminals) cable: XT30 (2+2)-F plug single-ended cable (100mm) × 1

3. Debugging serial port signal cable: GH1.25 cable-3pin (reversed port, 300mm) × 1

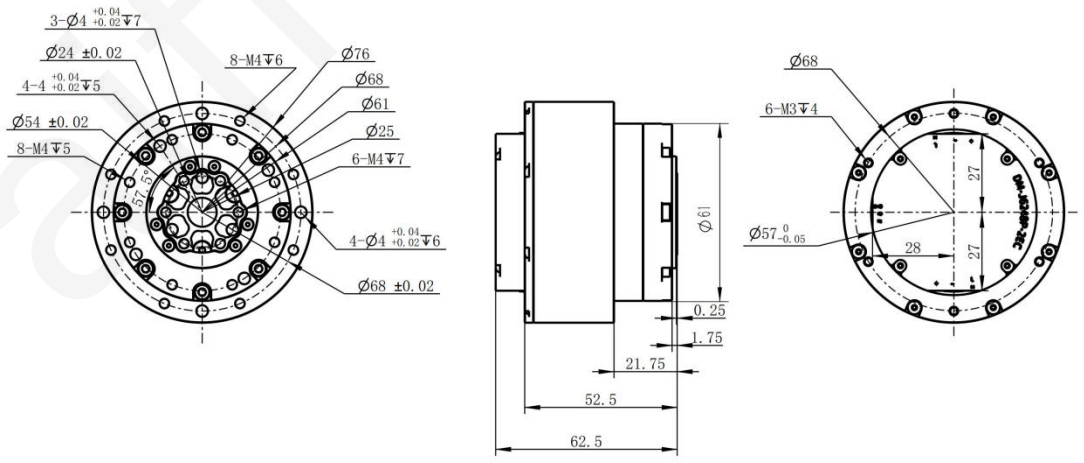
Interface and Pinout Description

Designation - No.	Interface Label	Description
Power Interface-1 (with CAN Communication Terminals) Power Interface-2 (with CAN Communication Terminals)		1. Connect to a power supply using the XT30(2+2)-F plug power cable. The rated voltage is 24V to power the motor. 2. Connect to external control devices via the CAN communication terminals to receive CAN control commands and feedback motor status information. 3. The motor includes two power interfaces. Any single interface can be used independently, or multiple motors can be

		connected in series via these interfaces for convenient cabling.
Debugging Serial Port-3		Connect to a PC using the GH1.25-3pin cable and a USB-to-CAN debugging tool (or a universal USB-to-serial module). Use the DAMIAO Technology Debugging Assistant to configure motor parameters and perform firmware updates.

Motor Dimensions and Installation

Please install the motor to the corresponding equipment by referring to the dimensions and locations of its mounting holes.



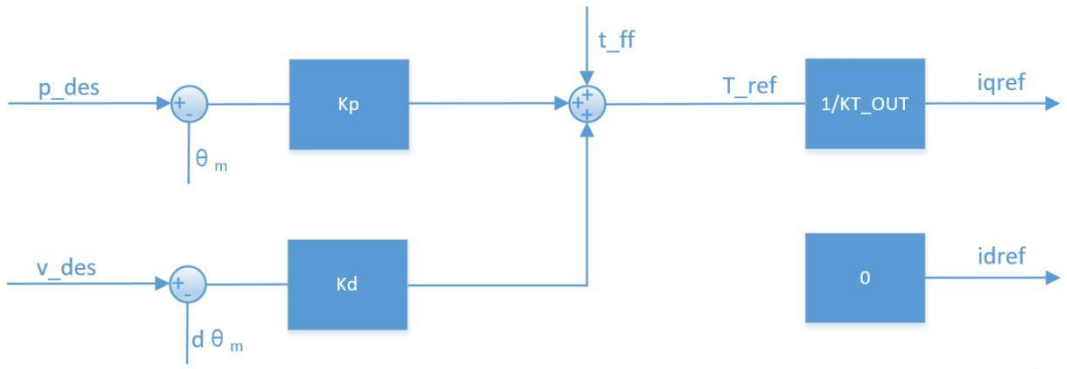
Indicator Status

Normal Status	Green LED steady on.	ERR bit = 1, indicates Enabled mode (normal operation).
	Red LED steady on.	ERR bit = 0, indicates Disabled mode.
Abnormal Status	Red LED flashing.	Indicates a fault. The corresponding fault types are: 8 – Over-voltage; 9 – Under-voltage; A – Over-current; B – MOSFET Over-temperature; C – Motor Coil Over-temperature; D – Communication Loss; E – Overload; The specific fault can be viewed via the feedback frame or displayed on the DAMIAO Technology Debugging Assistant interface.

Operating Modes

MIT Mode

The MIT Mode is designed for compatibility with the original MIT control scheme. It allows for seamless switching while enabling flexible setting of control limits (P_MAX, V_MAX, T_MAX). The electronic speed controller (ESC) converts the received CAN data into control variables, calculates a torque value, and uses this as the current reference for the current loop. The current loop then regulates the output to achieve the commanded torque current according to its control law. The block diagram of this control scheme is shown below:

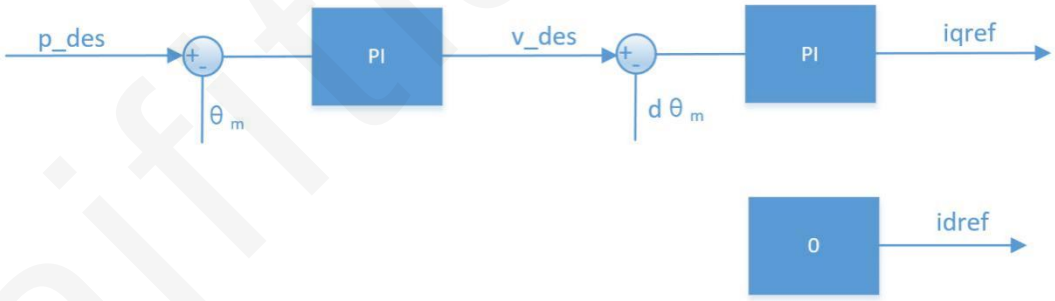


Various control modes can be derived from the MIT Mode. For example, when kp=0 and kd $\neq$ 0, setting v_des enables constant speed rotation; when kp=0 and kd=0, setting t_ff enables commanded torque output.

Note: When performing position control, kd must not be set to 0, as this may cause motor oscillation or even loss of control.

Position-Speed Mode

The Position-Speed cascade mode employs a three-loop series control structure. The position loop, as the outermost loop, provides the setpoint for the speed loop. The output of the speed loop then serves as the setpoint for the innermost current loop, which controls the actual current output. The block diagram of this control scheme is shown in the following figure:



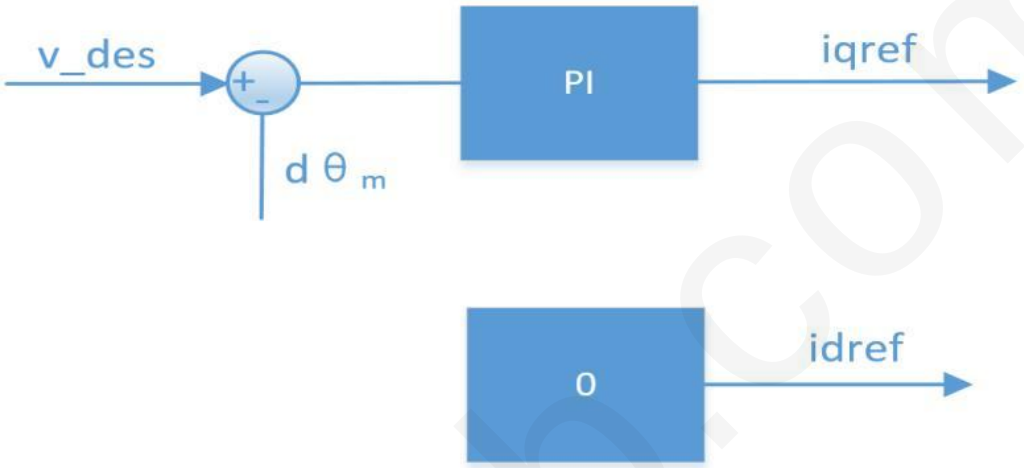
p_des is the target position for control, and v_des is used to limit the maximum absolute speed during motion.

When using the control parameters recommended by the debugging assistant, the Position-Speed cascade mode can achieve good control accuracy and relatively smooth motion, though the response time is comparatively longer. Configurable parameters, in addition to v_des, include acceleration/deceleration settings. If additional oscillation occurs during control, increasing the

acceleration/deceleration values may help.
Note: The units for p_des and v_des are radians (rad) and radians per second (rad/s) respectively, with the data type being float. The damping factor must be set to a positive non-zero value. Please also refer to the notes in the Speed Mode section.

Speed Mode

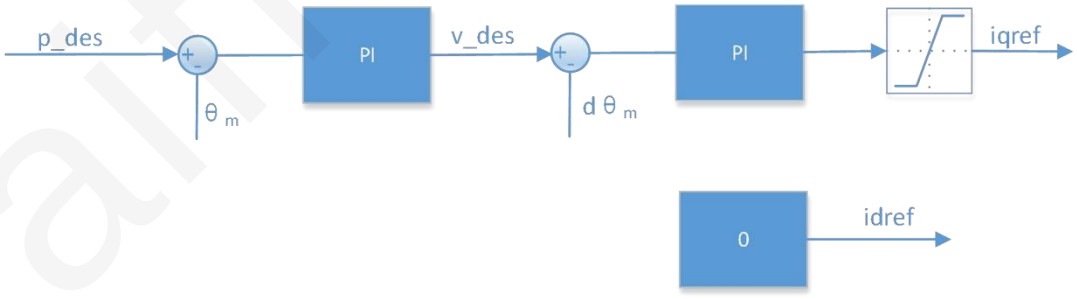
Speed Mode allows the motor to run stably at a set speed. The block diagram of this control scheme is shown below:



Note: The unit for v_des is rad/s, and its data type is float.

Torque-Position Hybrid Mode

The Torque-Position Hybrid Mode dynamically controls the output torque based on the Position-Speed control scheme. Its control block diagram is shown below:



A current command saturation block is added after the speed loop's output command to limit the reference to the current loop within a specified range.

CAN Communication

The motor is ready for use after calibration and parameter configuration. Control is implemented using the CAN Standard Frame (STD) format with a default baud rate of 1 Mbps. The baud rate can be changed to a different value via command. For details, please refer to the "CAN Baud Rate Modification" section.

Functionally, frames can be divided into Receive Frames and Feedback Frames. Receive Frames contain control data for commanding the motor. Feedback Frames contain status data sent from the motor to the upper-level controller. The feedback operates on a polled response basis: whenever the driver receives a frame whose ID matches the motor's configured CAN ID (checking the lower 8 bits, ignoring the upper 3 bits), it transmits the current status data onto the bus.Depending on the selected motor mode, the frame format definition and frame ID for Receive Frames vary. However, the Feedback Frame is identical across all modes.

Depending on the selected motor mode, the frame format definition and frame ID for Receive Frames vary. However, the Feedback Frame is identical across all modes.

Feedback Frame

The Feedback Frame ID is set via the debugging assistant (Master ID) and defaults to 0. It primarily feeds back the motor's position, speed, and torque information. Its frame format is defined as follows:

Feedback Frame (Message)	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
MST_ID	ID|ERR<<4	POS[15:8]	POS[7:0]	VEL[11:4]	VEL[3:0]|T[11:8]	T[7:0]	T_MOS	T_Rotor

where:
ID indicates the controller ID, which is the lower 8 bits of the CAN_ID.

ERR indicates the status, corresponding to the following status types:
0 – Disabled;
1 – Enabled;
8 – Over-voltage;
9 – Under-voltage;
A – Over-current;
B – MOSFET Over-temperature;
C – Motor Coil Over-temperature;
D – Communication Loss;
E – Overload;POS

POS indicates the motor's position information.

VEL indicates the motor's speed information.

T indicates the motor's torque information.

T_MOS indicates the average temperature of the MOSFETs on the driver, in °C.

T_Rotor indicates the average temperature of the coils inside the motor, in °C.

Position, speed, and torque data are converted from floating-point to signed fixed-point data using a linear mapping relationship. Position uses 16-bit data, while speed and torque both use 12-bit data, as shown in the following figure:

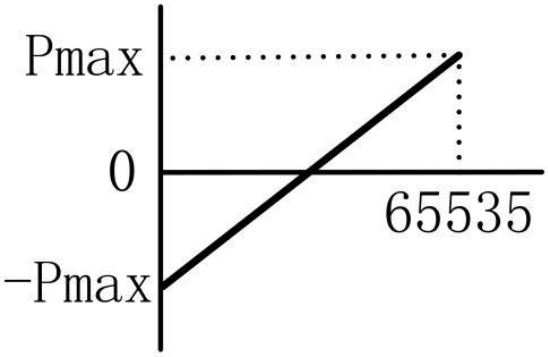


Figure: Position Linear Mapping Diagram

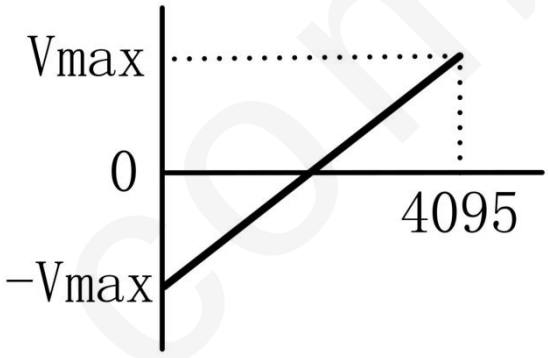


Figure: Speed Linear Mapping Diagram

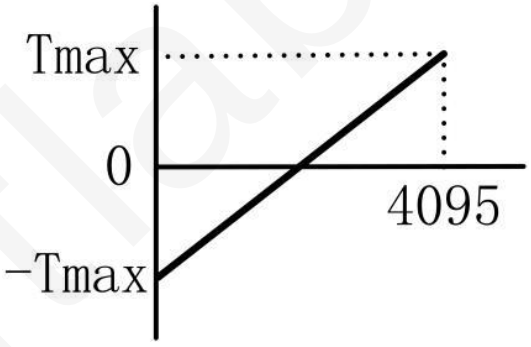


Figure: Torque Linear Mapping Diagram

Note:

① After power-on, the motor position output is limited to the range [-π, π] rad.

② The unit for position is rad (radians), and it represents the position of the output shaft, i.e., the position after gear reduction. All subsequent descriptions of position follow this definition and will not be reiterated.

③ The unit for speed is rad/s (radians per second), and it represents the speed of

the output shaft, i.e., the speed after gear reduction. All subsequent descriptions of speed follow this definition and will not be reiterated.

④ The unit for torque is N·m, and it represents the torque of the output shaft, i.e., the torque after gear reduction. All subsequent descriptions of torque follow this definition and will not be reiterated.

Control Frame in MIT Mode

Control Frame (Message)	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
ID	p_des [15:8]	p_des [7:0]	v_des [11:4]	v_des[3:0]| Kp[11:8]	Kp [7:0]	Kd [11:4]	Kd[3:0]| t_ff[11:8]	t_ff[7:0]

The Frame ID is equal to the set CAN ID value.

P_des: Position setpoint.

V_des: Velocity setpoint.

Kp: Position proportional coefficient.

T_ff: Torque feedforward value.

These parameters conform to the mapping relationships described in the previous section. The ranges for p_des, v_des, and t_ff can be set via the debugging assistant. The range for Kp is [0, 500], and the range for Kd is [0, 5].

A standard CAN data frame has only 8 bytes. The MIT control command format packs the five parameters—Position, Velocity, Kp, Kd, Torque—by bit combination into these 8 bytes. Specifically: Position occupies 2 bytes (16 bits), Velocity occupies 12 bits, Kp occupies 12 bits, and Kd occupies 12 bits.

A standard CAN data frame has only 8 bytes. The MIT control command format packs the five parameters—Position, Velocity, Kp, Kd, Torque—by bit combination into these 8 bytes. Specifically: Position occupies 2 bytes (16 bits), Velocity occupies 12 bits, Kp occupies 12 bits, and Kd occupies 12 bits.

The MIT commands are sent to the driver by proportionally converting

floating-point data into integers. The driver then proportionally converts the received integers back into floating-point data. This conversion utilizes the uint_to_float and float_to_uint functions. The specific implementation of this process is shown below:

```
float uint_to_float(int x_int, float x_min, float x_max, int bits){
    /// converts unsigned int to float, given range and number of bits ///
    float span = x_max - x_min;
    float offset = x_min;
    return ((float)x_int)*span/((float)((1<<bits)-1)) + offset;
    }
```

```
int float_to_uint(float x, float x_min, float x_max, int bits){
    /// Converts a float to an unsigned int, given range and number of bits ///
    float span = x_max - x_min;
    float offset = x_min;
    return (int) ((x-offset)*((float)((1<<bits)-1))/span);
    }
```

The use of these two conversion functions first requires determining the maximum and minimum values for the proportional scaling. These values can be queried on the parameter configuration page. The default maximum and minimum values for Kp and Kd are 0.0~500.0 and 0.0~5.0, respectively.Pos, Vel, and Torque are preset to ±12.5, ±45, and ±18 respectively, as shown in the figure below. These three parameters can be adjusted according to the motor's actual specifications.However, when sending control commands, they must correspond exactly to the set values; otherwise, the control commands will be proportionally scaled.

Control Frame in Position-Speed Mode

Control Frame (Message)	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
0x100+ID	p_des				v_des			

The Frame ID is the set CAN ID value plus an offset of 0x100.

P_des: Position setpoint, floating-point type, stored in little-endian byte order (least significant byte first).

V_des: Velocity setpoint, floating-point type, stored in little-endian byte order (least significant byte first).

The CAN ID for sending commands here is 0x100 + ID. The velocity setpoint (V_des) limits the maximum speed during operation.

Control Frame in Speed Mode

Control Frame (Message)	D[0]	D[1]	D[2]	D[3]
0x200+ID	v_des			

The Frame ID is the set CAN ID value plus an offset of 0x200.

V_des: Velocity setpoint, floating-point type, stored in little-endian byte order (least significant byte first).
The CAN ID for sending commands here is 0x200 + ID.

Control Frame in Torque-Position Hybrid Mode

Control Frame (Message)	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
0x300+ID	p_des				v_des		i_des	

P_des: Position setpoint, unit is rad, floating-point type, stored in little-endian byte order (least significant byte first).

V_des: Speed limit, unit is rad/s, amplified by a factor of 100, unsigned 16-bit integer type, stored in little-endian byte order. Range is 0-10000, values exceeding 10000 are limited to 10000, thus the corresponding actual speed limit range is 0~100 rad/s.

I_des: Torque current limit per-unit value, amplified by a factor of 10000, unsigned 16-bit integer type, stored in little-endian byte order. Range is 0-10000, values exceeding 10000 are limited to 10000, thus the corresponding actual current limit per-unit range is 0-1.0.

Current per-unit value: The actual current value divided by the maximum phase current value.

Enable

After the power-on self-test is completed, an "Enable" command must be sent to initiate control. The "Enable" frame is a type of control frame. Its Frame ID is as described previously. The difference lies in the Data Field. Regardless of the operating mode, the data definition for the "Enable" frame is the same, as follows:

D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFC

Disable

Disabled is the default state of the motor upon power-up. In this state, the voltage waveforms at the three motor terminals are identical, each being a 50% modulation wave of the power supply voltage.The "Disable" frame is a type of control frame. Its Frame ID is as described previously, and its Data Field is defined as follows:

D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFD

Save Position Zero

The "Save Position Zero" frame is a type of control frame. This command sets the current output shaft position as the zero point and resets the position setpoint to 0. Its Frame ID is as described previously, and its Data Field is defined as follows:

D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFE

Clear Error

When the motor experiences an error (e.g., overheating), sending a "Clear" command can clear the error. The "Clear" frame is a type of control frame. Its Frame ID is as described previously, and its Data Field is defined as follows:

D[0]	D[1]	D[3]	D[4]	D[5]	D[6]	D[7]
0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFB

Read Parameter

Message ID	Attributes	D[0]	D[1]	D[2]	D[3]

0x7FF	STD	CANID_L	CANID_H	0x33	RID

RID is the register address. Refer to the appendix <Register List and Ranges>.

Upon successful reading, the data of that register will be returned. The frame format is as follows:

Message ID	Attributes	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
MST_ID	STD	CANID_L	CANID_H	0x33	RID	Data			

The data is of floating-point or unsigned integer type, occupying 32 bits (4 bytes), with the least significant byte at D4 and the most significant byte at D7. The same applies to subsequent similar cases.

Write Parameter

Message ID	Attributes	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
0x7FF	STD	CANID_L	CANID_H	0x55	RID	Data			

The RID is as described above. Upon a successful write, the written data will be returned in a frame format identical to the one sent.

Message ID	Attributes	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
MST_ID	STD	CANID_L	CANID_H	0x55	RID	Data			

Writing to a register takes effect immediately, but the data is volatile and will be lost after power loss. It is necessary to send a "Store Parameters" command to write all modified parameters into the on-chip non-volatile memory.

Store Parameters

Message ID	Attributes	D[0]	D[1]	D[2]	D[3]
0x7FF	STD	CANID_L	CANID_H	0xAA	0x01

After a successful write, the return format is as follows:

Message ID	Attributes	D[0]	D[1]	D[2]	D[3]
MST_ID	STD	CANID_L	CANID_H	0xAA	0x01

Note:

① Storing parameters only takes effect when the motor is in Disabled mode.

② All parameters will be retained at once when storing.

③ This operation writes parameters to the on-chip flash, with a maximum duration of 30ms per operation. Please ensure sufficient time is allotted.

④ The flash has approximately 10,000 erase/write cycles. Please avoid frequently sending the "Store Parameters" command.

Mode Switching

Supports switching between multiple modes. The currently supported control modes are as follows:

Code	Mode
1	MIT Mode
2	Position-Speed
3	Speed
4	Torque-Position Hybrid

The mode can be changed by modifying the value of the Mode Register (0x0A). During mode switching, the motor first clears all command values, including position, speed, and the torque feedforward, Kp, and Kd values in MIT Mode.

When switching from one mode to a position control mode, to prevent impact, it is recommended to first read the accurate position (value of Register 0x50) and then consider switching, preferably when the motor speed is zero.

After a mode change, the setting is not saved to flash and will be lost upon power loss. After a power cycle, the control mode will be set to the mode last stored in flash.

CAN Baud Rate Modification

The current CAN communication baud rate can be modified by writing specific data to the Baud Rate Register (address 0x23). Specific baud rates are supported. The currently available baud rates are as follows:

Code	Baud Rate
0	125K

1	200K
2	250K
3	500K
4	1M
5	2M
6	2.5M
7	3.2M
8	4M
9	5M

After successfully changing the baud rate, the driver first sends feedback data at the original baud rate, then communicates at the new baud rate. Upon power-up, the motor first checks the stored baud rate. If it is greater than 5 Mbps, it defaults to 1 Mbps automatically. For rates greater than 1 Mbps (excluding 1 Mbps), it automatically switches to CAN FD functionality. If the baud rate is less than or equal to 1 Mbps, it automatically switches to CAN 2.0B.A motor configured for CAN FD can still receive CAN 2.0B data frames. However, when sending feedback frames, it uses CAN FD, so the upper-level controller will not receive the feedback data, and the driver will continuously report errors. If a CAN 2.0B controller has an incorrect ID set, the baud rate can still be changed back using the baud rate modification command.

Register List and Ranges

Addr(HEX)	Addr(HEX)	Variable	Description	R/W	Range	Type
0x00	0	UV_Value	Undervoltage protection value	RW	(10.0,fmax]	float
0x01	1	KT_Value	Torque coefficient	RW	[0.0,fmax]	float
0x02	2	OT_Value	Over-temperature protection value	RW	[80.0,200)	float
0x03	3	OC_Value	Over-current protection value	RW	(0.0, 1.0)	float
0x04	4	ACC	Acceleration	RW	(0.0,fmax)	float
0x05	5	DEC	Deceleration	RW	[-fmax,0.0)	float

0x06	6	MAX_SPD	Maximum speed	RW	(0.0,fmax]	float
0x07	7	MST_ID	Feedback ID	RW	[0,0x7FF]	uint32
0x0A	8	ESC_ID	Receiver ID	RW	[0,0x7FF]	uint32
0x09	9	TIMEOUT	Timeout alarm duration	RW	[0,2^32-1]	uint32
0x0A	10	CTRL_MODE	Control mode	RW	[0,4]	uint32
0x0B	11	Damp	Moto rviscous damping coefficient	RO	/	float
0x0C	12	Inertia	Motor moment of inertia	RO	/	float
0x0D	13	hw_ver	Reserved	RO	/	uint32
0x0E	14	sw_ver	Software version	RO	/	uint32
0x0F	15	SN	Reserved	RO	/	uint32
0x10	16	NPP	Motor pole pairs	RO	/	uint32
0x11	17	Rs	Motor phase resistance	RO	/	float
0x12	18	Ls	Motor phase inductance	RO	/	float
0x13	19	Flux	Motor flux linkage	RO	/	float
0x14	20	Gr	Gear reduction ratio	RO	/	float
0x15	21	PMAX	Position mapping range	RW	(0.0,fmax]	float
0x16	22	VMAX	Speed mapping range	RW	(0.0,fmax]	float
0x17	23	TMAX	Torque mapping range	RW	(0.0,fmax]	float
0x18	24	I_BW	Current loop control bandwidth	RW	[100.0, 1.0e4]	float
0x19	25	KP_ASR	Speed loop Kp	RW	[0.0,fmax]	float
0x1A	26	KI_ASR	Speed loop Ki	RW	[0.0,fmax]	float
0x1B	27	KP_APR	Position loop Kp	RW	[0.0,fmax]	float
0x1C	28	KI_APR	Position loop Ki	RW	[0.0,fmax]	float
0x1D	29	OV_Value	Over-voltage protection value	RW	TBD	float

0x1E	30	GREF	Gear torque efficiency	RW	(0.0, 1.0]	float
0x1F	31	Deta	Speed loop damping factor	RW	[1.0,30.0]	float
0x20	32	V_BW	Speed loop filter bandwidth	RW	(0.0,500.0)	float
0x21	33	IQ_c1	Current loop enhancement coefficient	RW	[100.0, 1.0e4]	float
0x22	34	VL_c1	Speed loop enhancement coefficient	RW	(0.0, 1.0e4]	float
0x23	35	can_br	CAN baud rate code	RW	[0,4]	uint32
0x24	36	sub_ver	Sub-version number	RO	/	uint32
0x32	50	u_off	U-phase offset	RO	/	float
0x33	51	v_off	V-phase offset	RO	/	float
0x34	52	k1	Compensation factor 1	RO	/	float
0x35	53	k2	Compensation factor 2	RO	/	float
0x36	54	m_off	Angle offset	RO	/	float
0x37	55	dir	Direction	RO	/	float
0x50	80	p_m	Motor current position	RO	/	float
0x51	81	xout	Output shaft position	RO	/	float

RW: Read/Write.

RO: Read-Only.

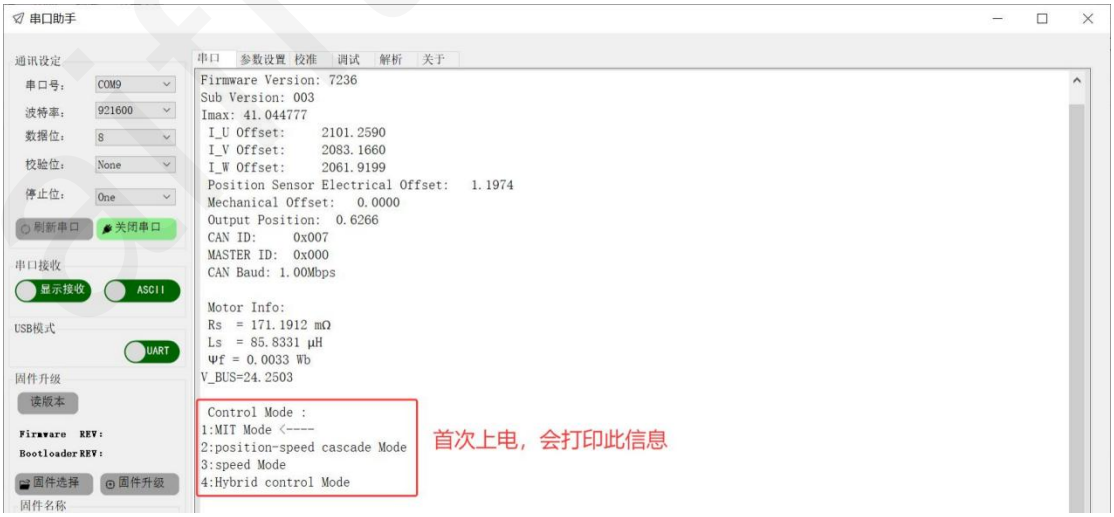
Motor Debugging Procedure

Please use PC software version V1.7.0.0 or higher for debugging. The following demonstrates the debugging procedure using version V1.7.0.0.

Connect the Device

First, connect the motor's serial port, CAN port, and power supply interface. On the computer, open the debugging assistant software, select the corresponding serial port device, and open the serial port.

When power is supplied to the motor, the serial port will print the following information. Control Mode: indicates the current drive mode. Different modes use different command formats (refer to the CAN Communication section).

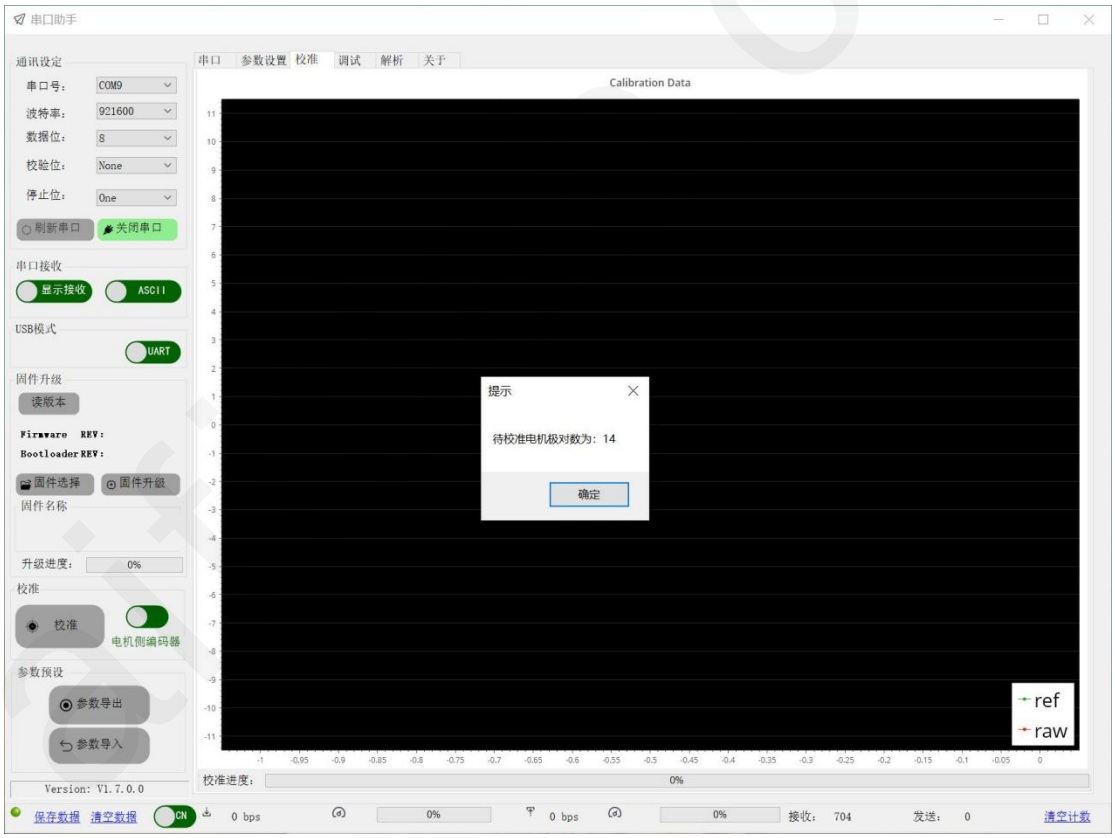


Motor-Side Encoder Calibration

The purpose of motor-side encoder calibration is mainly to correct sensor installation errors. During the process, the motor will rotate forward and reverse for one rotor electrical cycle. Ensure the motor can rotate freely; it is best to run the motor under no-load conditions, otherwise calibration may fail. The motor is calibrated at the factory, and the data is stored internally. Under normal circumstances, it can be used directly without recalibration. The following provides the calibration steps. In abnormal states such as after replacing the driver board or experiencing abnormal vibration, perform calibration before operation.

Step 1: Motor-Side Encoder Calibration

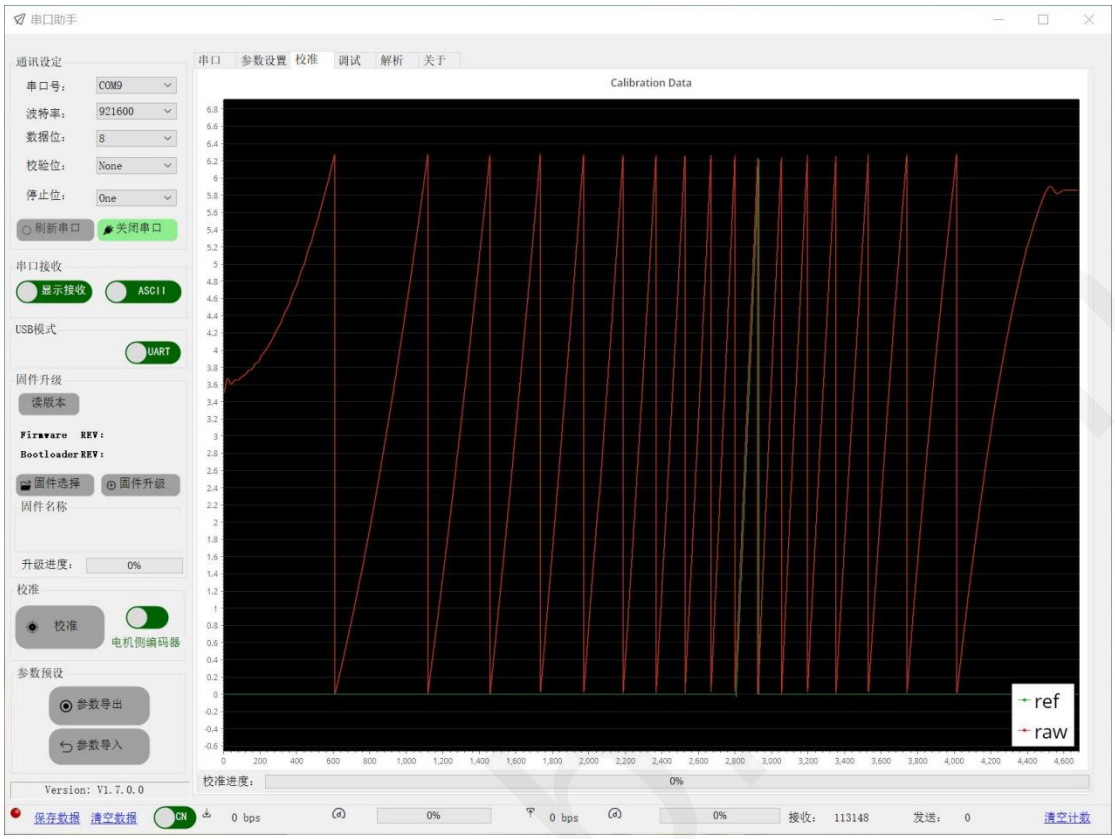
Click "Calibrate". The motor will rotate first. After rotation completes, it returns the number of pole pairs to the debugging assistant.



Afterwards, the motor will automatically perform parameter identification. During this process, the motor will have rotational movement. Please secure the motor properly to prevent accidents.

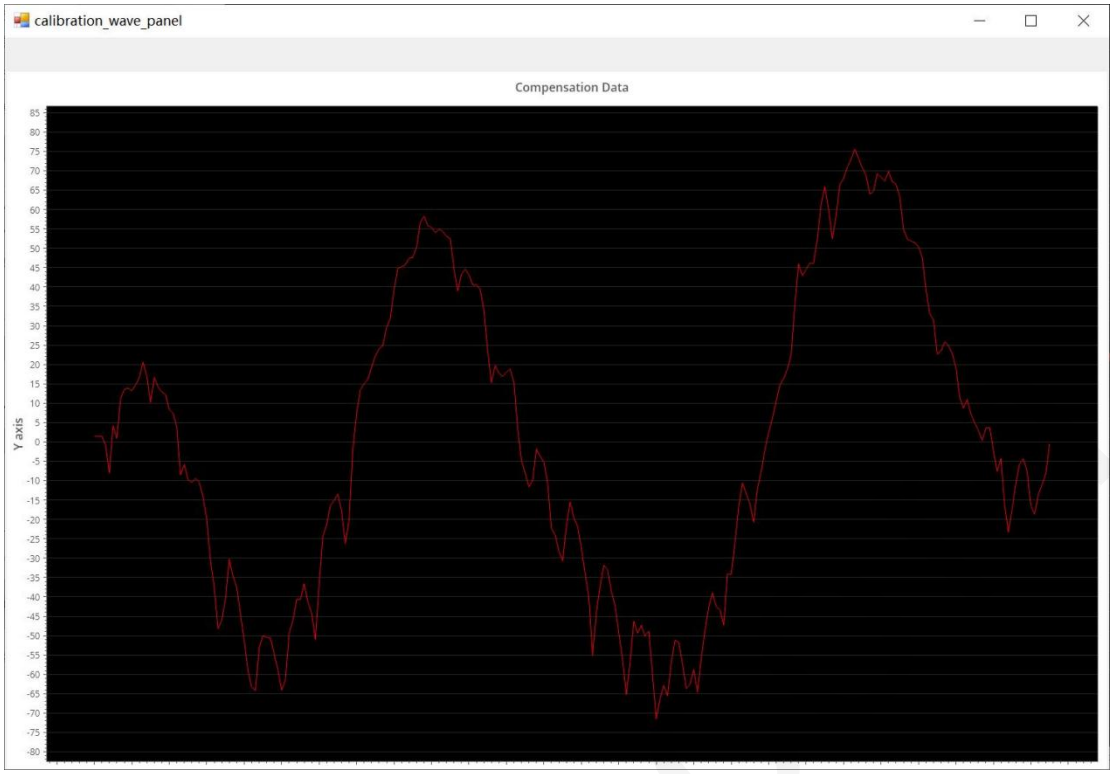
After parameter identification is complete, the motor will automatically perform

sensor calibration and display the motor's sensor waveform graph, as shown below:



Note: The maximum value of the green curve should not significantly exceed that of the red curve, otherwise it will cause failure. If failed, simply recalibrate.

Step 2: After calibration is complete, the calibration values will be uploaded automatically. Example as follows:



Step 3: Calibration Data Checking and Storage

Pay special attention to the value of "Compensation Data". It is recommended not to exceed ±1000. If it exceeds 1000, there may be several reasons:

1. Incorrect identification of pole pairs;

2. Excessive motor resistance, causing stalling;

3. Improper sensor installation;

Step 4: Troubleshoot based on the above points.

Parameter Calibration

Parameter identification mainly determines key motor parameters such as phase resistance, phase inductance, and flux linkage. The motor has been calibrated at the factory, and the data is stored internally. It can be used directly without recalibration. The calibration steps are provided below:

Click the "Parameter Settings" tab, then click the "Parameter Calibration" button. The driver will enter the identification procedure. During this process, the motor will

rotate. It should be kept under no-load conditions and properly secured.

After identification is complete, the results will be uploaded automatically.

The viscous damping coefficient is for reference only and can be calibrated multiple times.

[Note] None of the above parameters, except for the viscous damping coefficient, should have negative values. If negative values appear, verify the motor status before recalibrating.

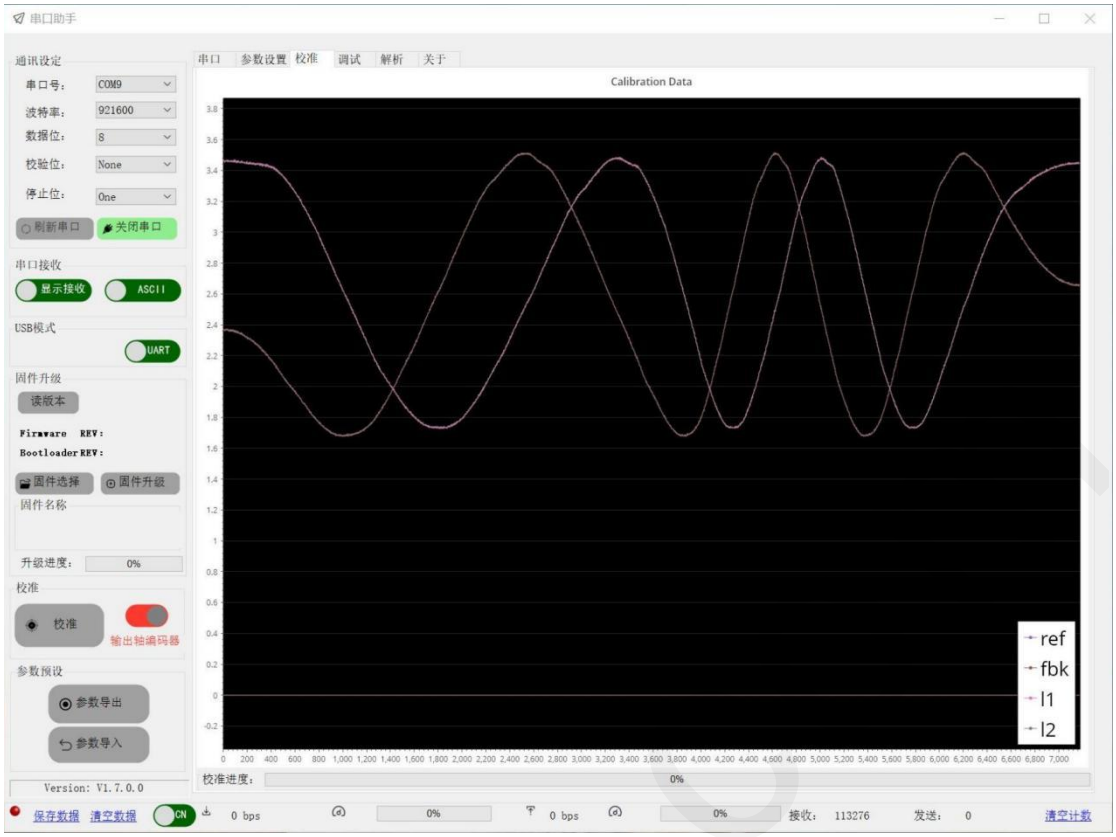
Output Shaft Encoder Calibration

Dual-encoder motors require output shaft encoder calibration to improve its accuracy. During the process, the motor will rotate forward one full revolution. Ensure the motor can rotate freely and is running under no-load conditions; otherwise, calibration errors may be excessive. The output shaft encoder has been calibrated at the factory, and the data is stored internally. Under normal circumstances, it can be used directly without recalibration. The following provides the steps for output shaft encoder calibration. In abnormal states such as after replacing the driver board or changes in output position, perform this calibration before operation.

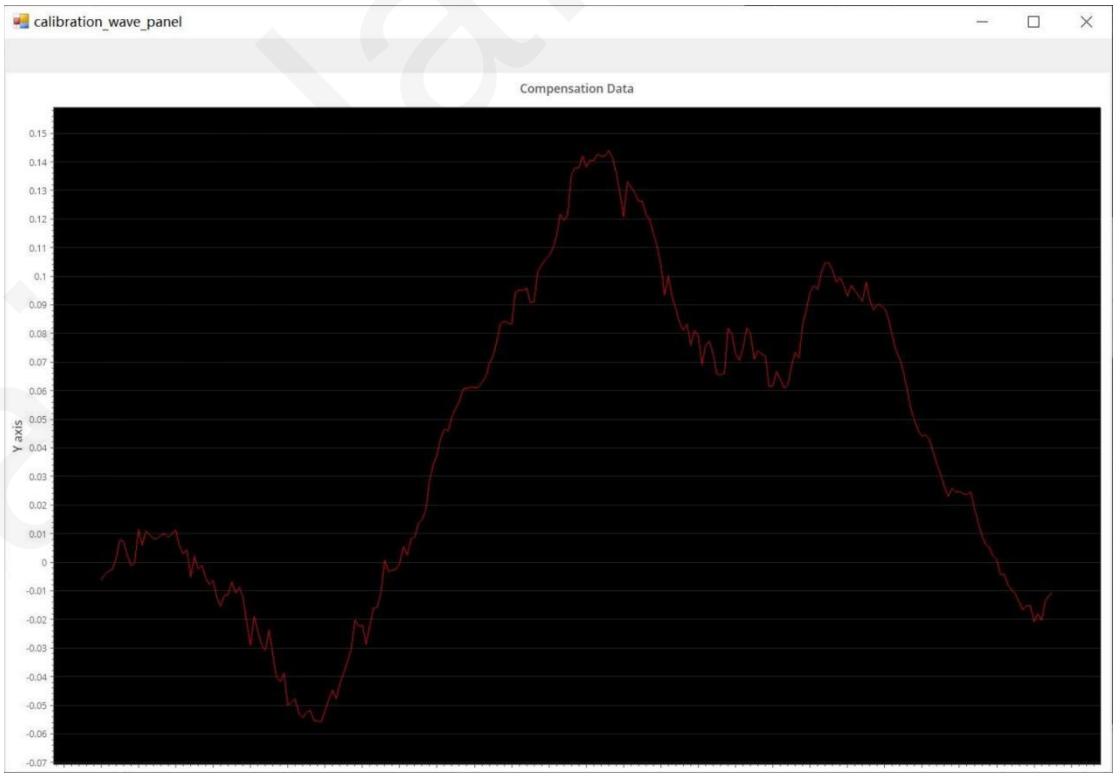
Step 1: Output Shaft Encoder Calibration

Slide the control next to the calibration button until "Output Shaft Encoder" is displayed, then click Calibrate.

The motor will begin to rotate one full revolution and upload the raw encoder data waveform, as shown in the figure below:



Step 2: After calibration is complete, the calibration values will be uploaded and displayed automatically, as follows:



Step 3: Calibration Data Verification

After calibration is complete, the driver will automatically perform data verification. If the deviation in the verification data is too large, it will report an error, error type 3, with the red LED flashing. Typically, errors are caused by output shaft encoder abnormalities and can be resolved by contacting after-sales support.

Serial Port Parameter Reading/Writing

1. Read Parameters

Under the "Parameter Settings" tab, click the "Read Parameters" button. The driver will upload the parameters stored and used by the driver to the debugging assistant. Please carefully check the parameters.

The read parameters include:

(1) Driver Parameters: The driver's parameter settings section.

l Pole Pairs: The motor's pole pairs, obtained automatically through calibration. Do not modify.
l Undervoltage: If the power supply voltage falls below the set value, the driver will be unable to control the motor. Default is 20V.
l Overvoltage: Sets the upper voltage limit for driver operation. The power supply voltage is sampled when the motor is powered on. If the sampled voltage exceeds this value, the driver will report an error and exit the enabled state. If overvoltage occurs at power-on, it will continuously report an error.
l Acceleration/Deceleration: Used in non-MIT modes to limit the driver's acceleration/deceleration. Motor rotor parameter, unit is Krad/s², where deceleration is a negative number.
l Reduction Ratio: The motor's gear reduction ratio. It primarily affects the motor's output speed and position, and indirectly affects torque feedback information. Do not modify.
l Overtemperature: Motor coil temperature protection value. Set according to usage requirements, recommended not to exceed 100°C. After overtemperature, the driver exits the enabled state, enters the disabled state, and reports an error.
l CAN_ID: Driver ID number, used for CAN commands, in hexadecimal. It is recommended to set it to less than 16 to avoid conflict with error type codes.
l Master ID: Frame ID number used for driver feedback information, in hexadecimal.
l CAN Timeout: CAN communication timeout setting, 32-bit integer. Indicates the number of counting cycles after which, if no CAN command is detected, motor

protection is triggered. One counting cycle is 50us. Only effective when the motor is enabled.
l Speed Limit: Used only in Speed Mode to limit the maximum operating speed of the motor rotor (before reduction), unit is rad/s.
l Overcurrent: Used to limit the motor's maximum phase current, as a percentage.
l CAN Baud Rate: Configures the CAN baud rate, supports 125Kbps to 5Mbps.
l Current Loop Enhancement Coefficient: Auxiliary current loop control parameter. Modification is not recommended.
l Speed Loop Filter Bandwidth: Speed filtering bandwidth, unit is Hz.
l Sub-version Number: Firmware sub-version number.
l Speed Loop Enhancement Coefficient: Auxiliary speed loop control parameter. The recommended adjustable range is 200-800.

Note: After successfully clicking "Read Parameters", the parameters on the debugging interface, such as PMAX, ID, etc., will be updated synchronously.

(2) Motor Parameters: The motor parameters section.

These parameters are automatically identified by the driver (parameter calibration is required once after replacing the driver board. These parameters are automatically saved in the driver).

(3) Control Amplitudes: Drive command parameter range settings.

l PMAX: In MIT Mode, serves as the mapping value for commands received by the motor. In other modes, it serves as the mapping value for motor feedback information. Refer to the CAN Communication chapter for mapping rules.
l VMAX: Same as PMAX.
l TMAX: Same as PMAX.
l KT_OUT: Motor torque coefficient. Set to 0 if motor parameter identification is accurate.
l Gear Coefficient: Gear torque transmission coefficient, less than or equal to 1.0.
l Damping Factor: Ratio of current loop to speed loop control bandwidth. Not used.

(4) Control Settings

l Control Mode: MIT Mode, Position-Speed Mode, Speed Mode, Torque-Position Hybrid Mode.
l Current Bandwidth: Sets the gain coefficient for the current loop. Default is 1000.
l Speed KP/KI, Position KP/KI: Parameters for the speed loop and position loop.

2. Write Parameters

Check the driver parameters, control amplitudes, control settings, and other parameters. Modify the parameters that need to be changed, then finally click "Write

Parameters" to save the modified parameters to the driver.

Clicking the "Write Parameters" button will cause the chip to automatically reset. Therefore, when using "Write Parameters", ensure the motor is in the "Disabled" state to prevent safety incidents.

[Note]:
① Do not modify the pole pairs and reduction ratio parameters.
② After clicking "Write Parameters", the driver will automatically reboot via software; no external power cycle is needed.
③ The "Temporary Save" button only takes effect for controller parameters, and the data will be lost after power loss.

3. Temporary Save

During parameter tuning, the "Temporary Save" button will automatically apply the current controller parameters, which will be lost after power loss. After debugging is complete, save the controller parameters in the driver using the "Write Parameters" button.

[Note]: Mode switching cannot be performed via Temporary Save.

Debugging

This function is only operational when the CAN interface is connected and is for debugging a single motor only. Connect the CAN cable to the driver board before

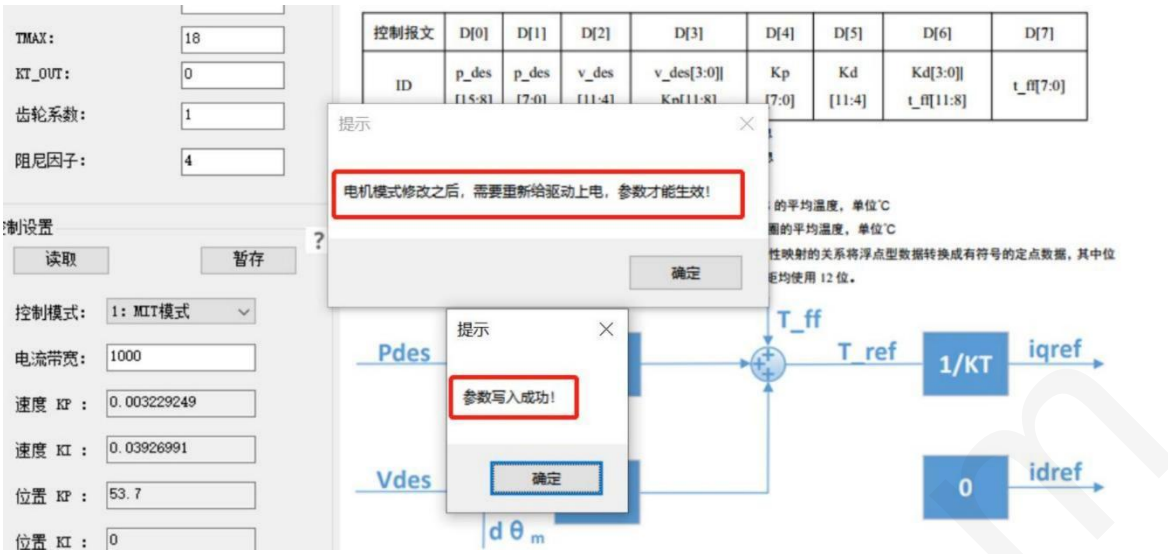
operation. Before debugging, confirm the wiring sequence and the current control mode. Select the corresponding sub-tab in the debugging page according to the different modes.

1. Selection and Confirmation of Control Mode

Click "Parameter Settings" in the debugging assistant, then click "Control Mode" under Control Settings. You can choose from four modes: MIT Mode, Position-Speed Mode, Speed Mode, and Torque-Position Hybrid Mode. Then click "Write Parameters" to set the control mode.

When modifying the control mode, a prompt window will pop up: "Parameters written successfully!".

Since the "Write Parameters" function triggers an automatic reset, there is no need to power cycle the driver for the changes to take effect. The reminder from earlier versions, "After modifying the motor mode, the driver needs to be powered on again for the parameters to take effect!" has been disabled.

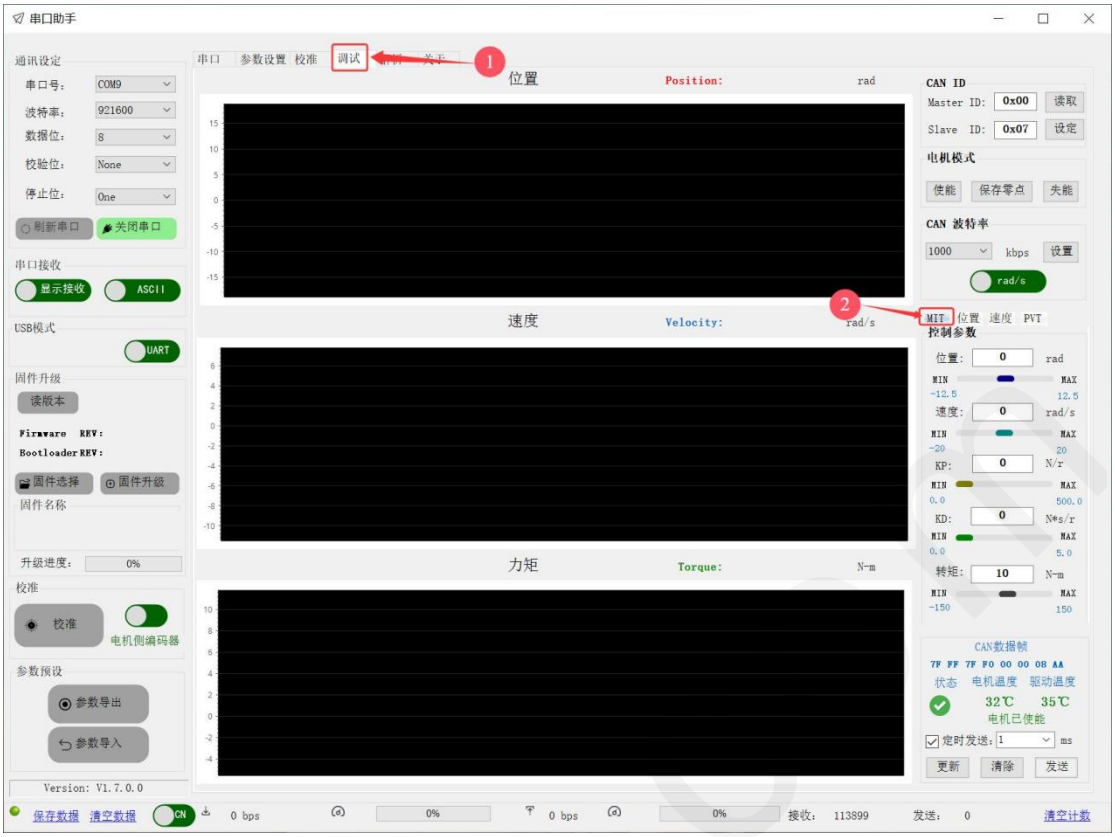


(1) Determine the control mode based on the data printed by the serial port upon power-up. The mode indicated by the arrow is the current driver control mode.

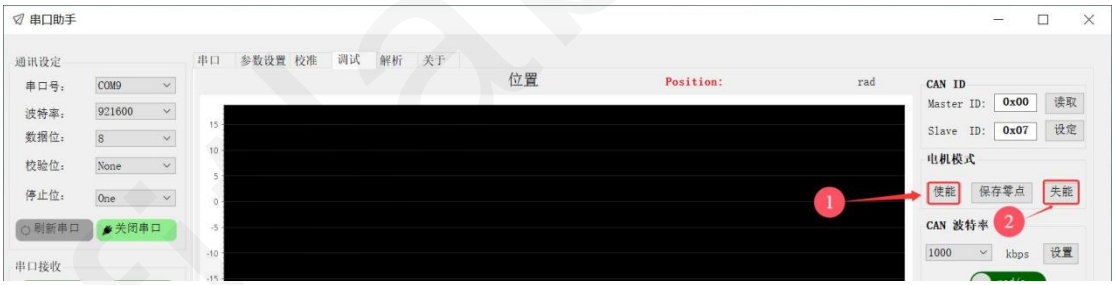
(2) Alternatively, determine the driver's control mode based on the information displayed after re-reading parameters on the parameter setting page.

2. MIT Mode

(1) Referring to Selection and Confirmation of Control Mode, ensure the current mode is set to MIT Mode. Confirm the current control mode and select the corresponding MIT sub-tab in the debugging page.



(2) Ensure the CAN ID is correct (can be obtained from serial port print information or the parameter setting page, or can be set via the read/set buttons on the debugging page).

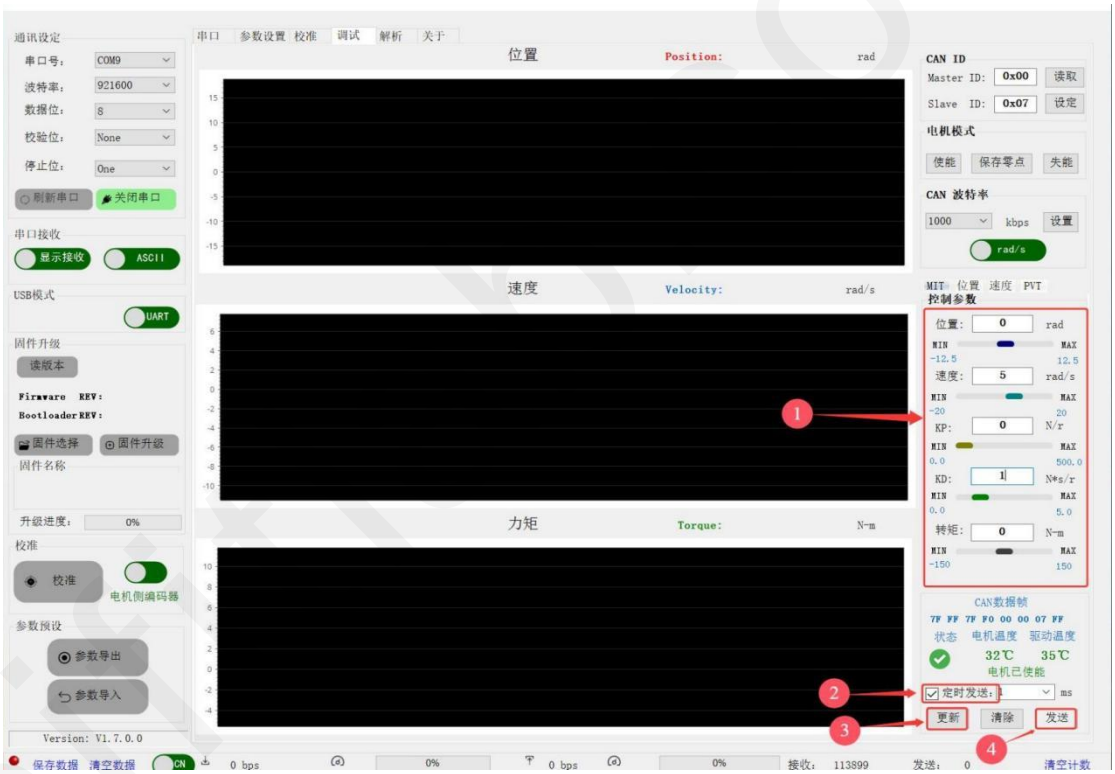


(3) MIT Mode has three control methods: Speed, Position, and Torque.

① Speed Control

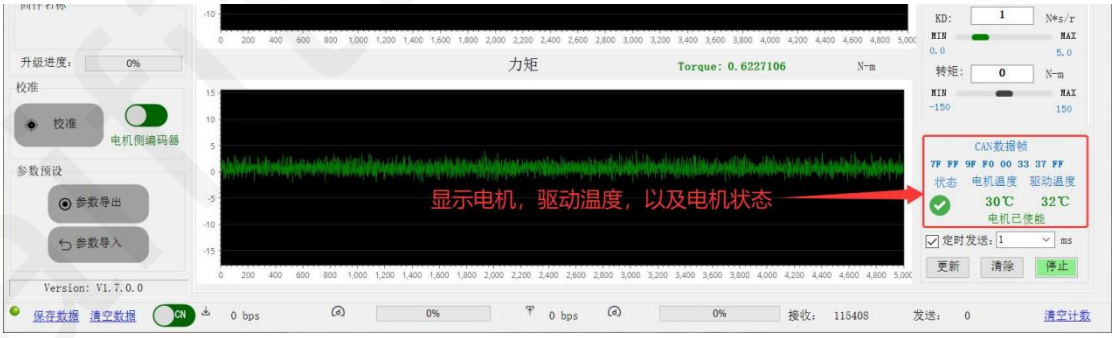
Step 1: Click the "Enable" button in the motor mode section. The driver's green LED will light up, indicating the motor is enabled.

Step 2: Provide a speed setpoint to run the motor at the given speed.
For example: Speed setpoint: 5 rad/s, KD: 1 N·s/rad, all other parameters set to 0. Check the "Timed Send" box, then click the "Update" button followed by the "Send" button. The parameter curve changes can be viewed on the debugging interface. Secure the motor.

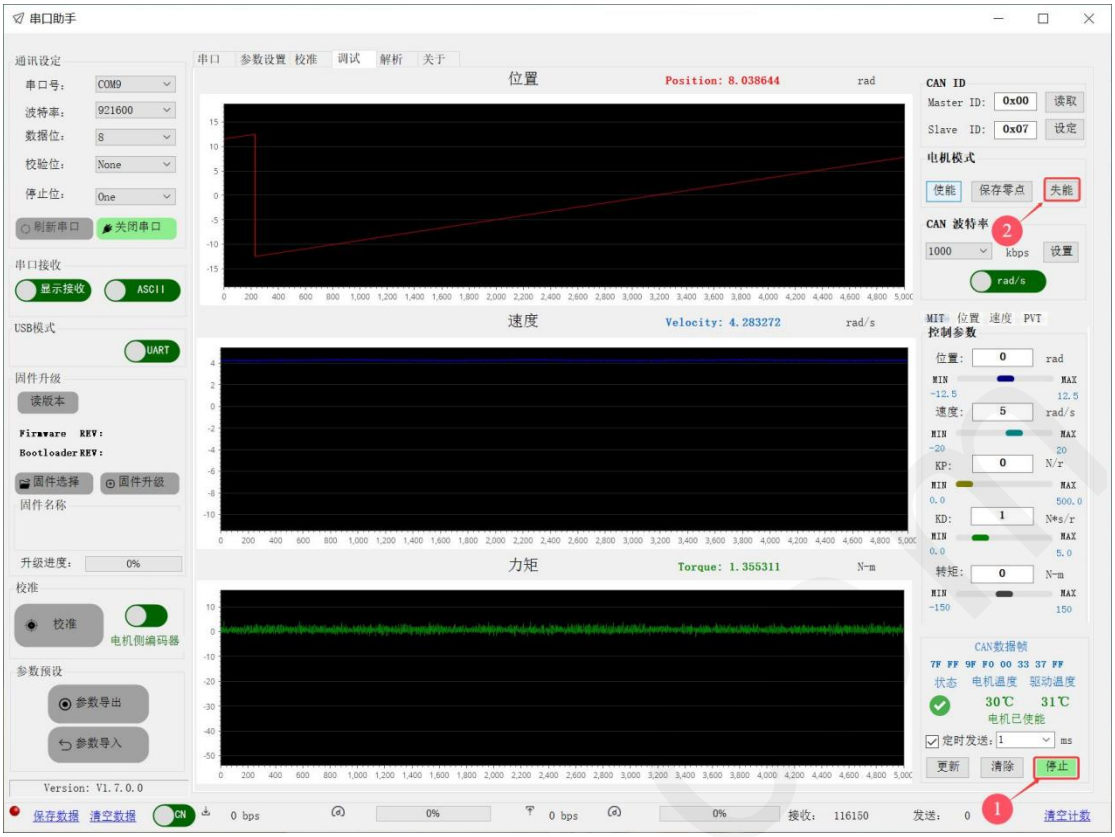


To modify control parameters during debugging, directly change the parameters on the interface, keep the "Timed Send" box checked, and click the "Update" button to proceed with debugging.

The debugging assistant interface displays the current motor and driver temperature, as well as motor operation status in real time. This information can also be viewed via the feedback frame. Refer to the "4.1 Feedback Frame" section for the feedback frame format and status types.

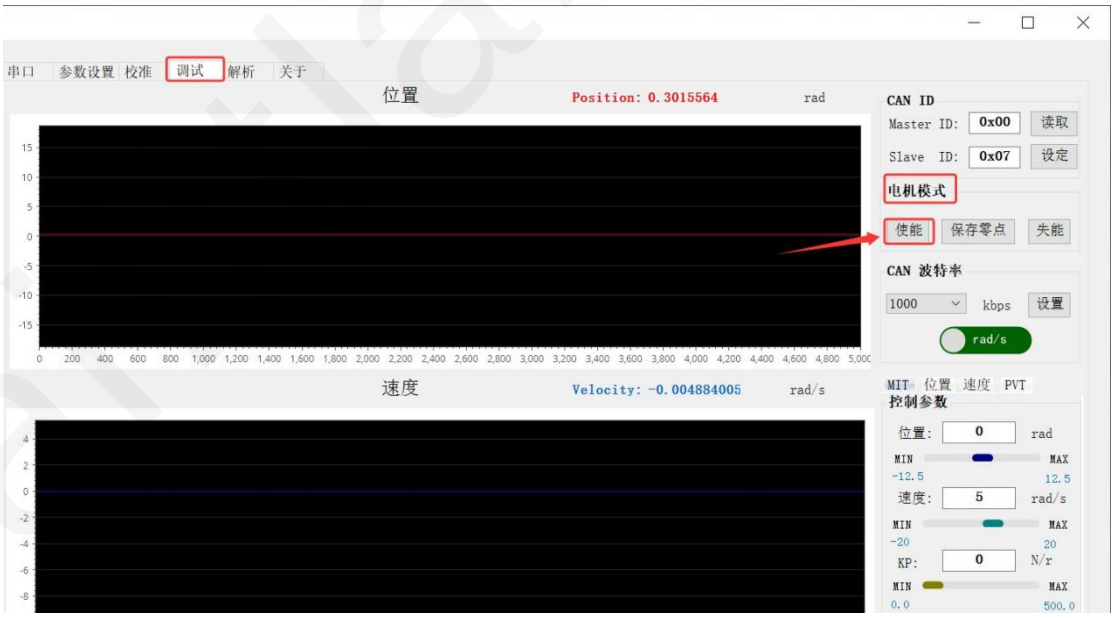


Step 3: To exit debugging, click the "Stop" and then the "Disable" buttons in sequence. The driver's red LED will light up, indicating exit from the motor operation mode.



② Position Control

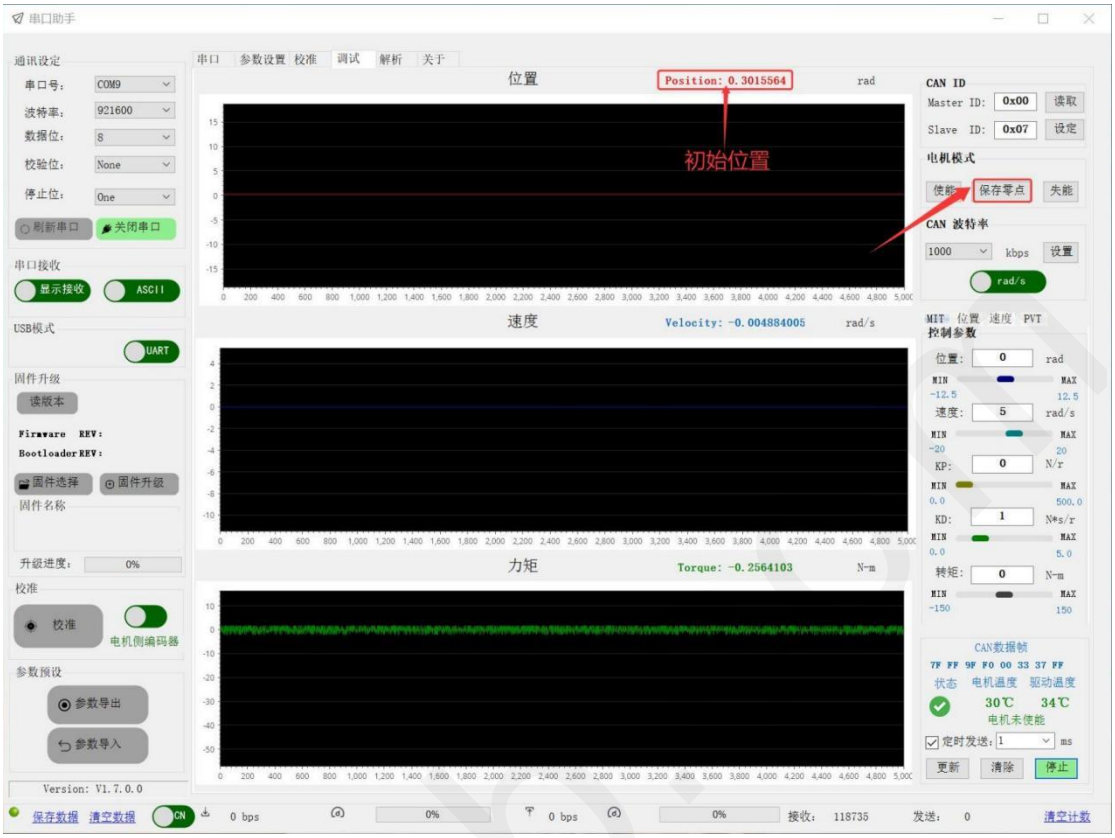
Step 1: Click the "Enable" button in the motor mode section. The driver's green LED will light up, indicating the motor is enabled.



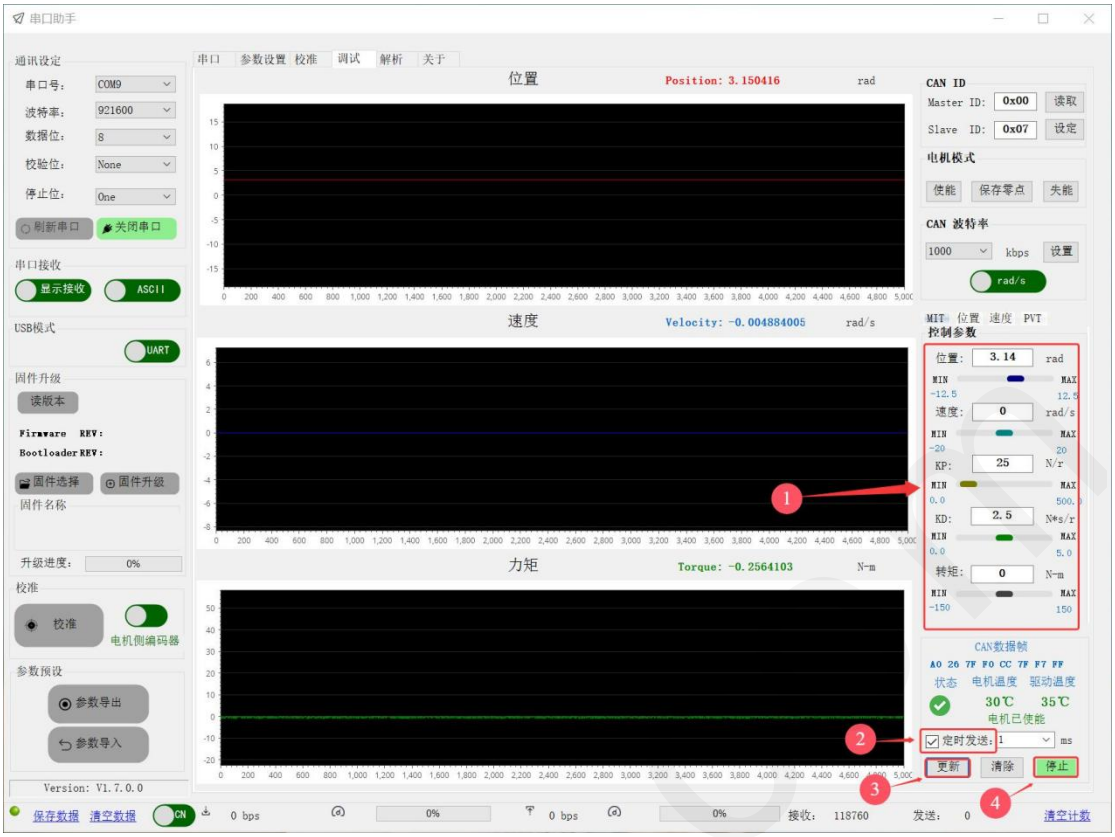
Step 2: Provide a position setpoint to rotate the motor to the specified position. Note the motor's initial position at this time. When setting the "Position" parameter in the control parameters section, avoid a large difference from the initial position to prevent motor shock.

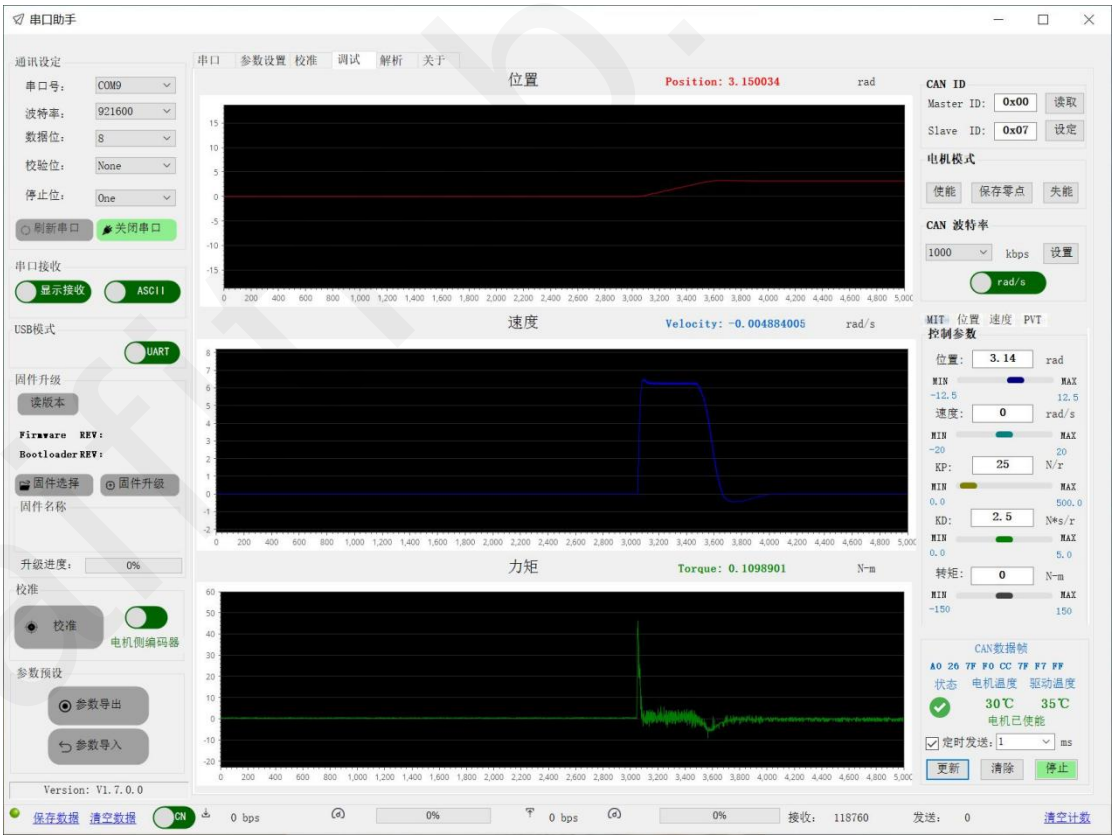
You can click "Save Zero Point" in the control command section to set the motor's

current position as the zero point, facilitating the setting of the "Position" parameter.



For example, given parameters: Position setpoint: 3.14 rad, KP: 2 N/rad, KD: 1 N·s/rad, all other parameters set to 0. Check the "Timed Send" box, then click the "Update" button followed by the "Send" button. The parameter curve changes can be viewed on the debugging interface. Secure the motor.

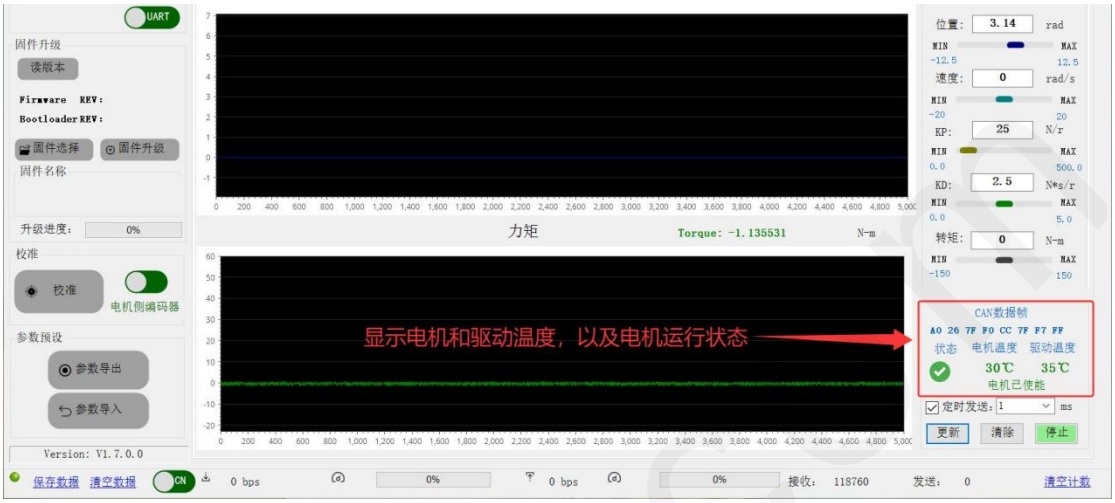




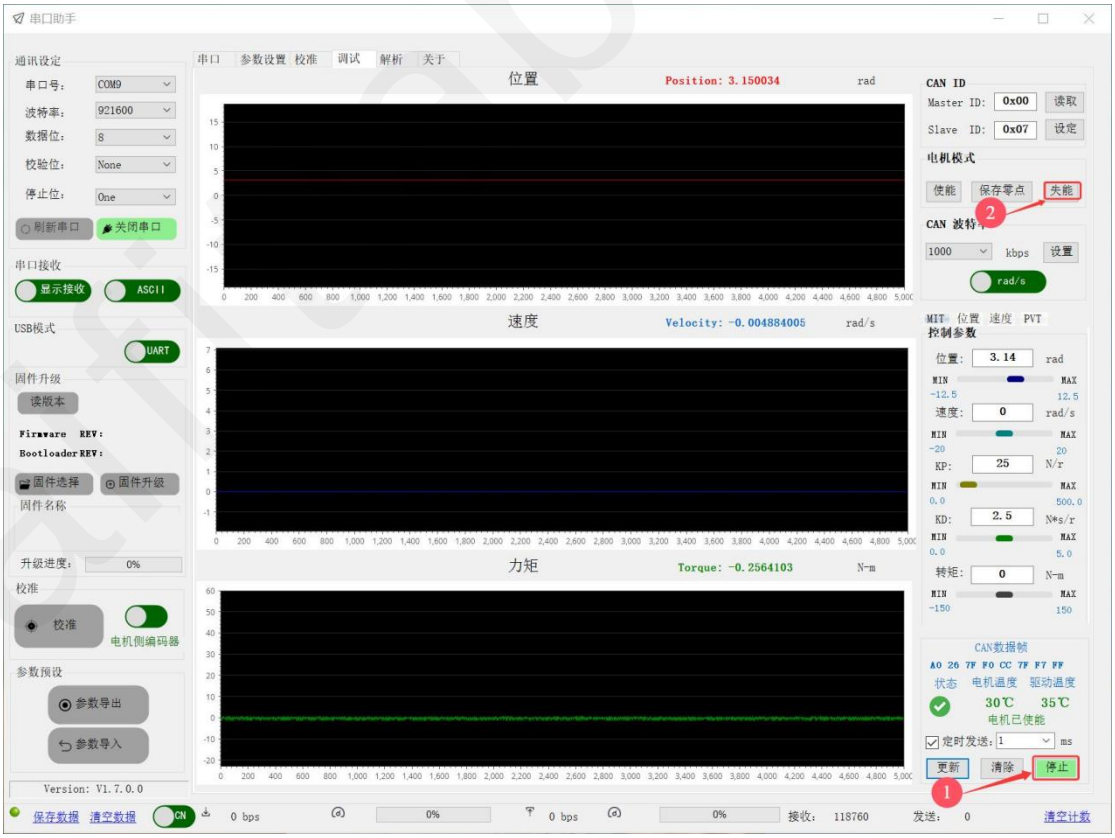
To modify control parameters during debugging, directly change the parameters on the interface, keep the "Timed Send" box checked, and click the "Update" button

to proceed with debugging.

The debugging assistant interface displays the current motor and driver temperature, as well as motor operation status in real time. This information can also be viewed via the feedback frame. Refer to the "4.1 Feedback Frame" section for the feedback frame format and status types.



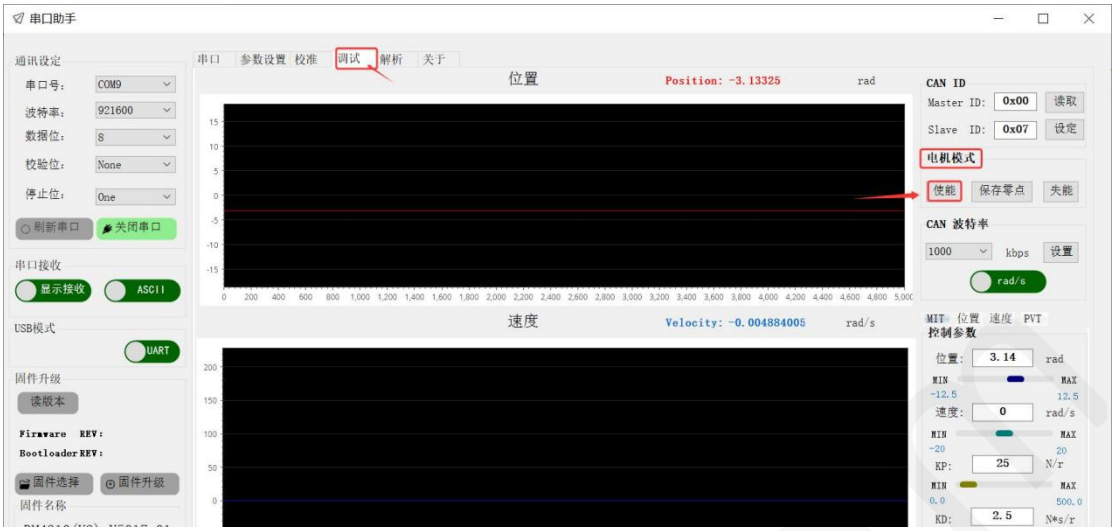
Step 3: To exit debugging, first stop the motor, then click the "Stop" and "Disable" buttons in sequence. The driver's red LED will light up, indicating exit from the motor operation mode.



③ MIT Torque Control

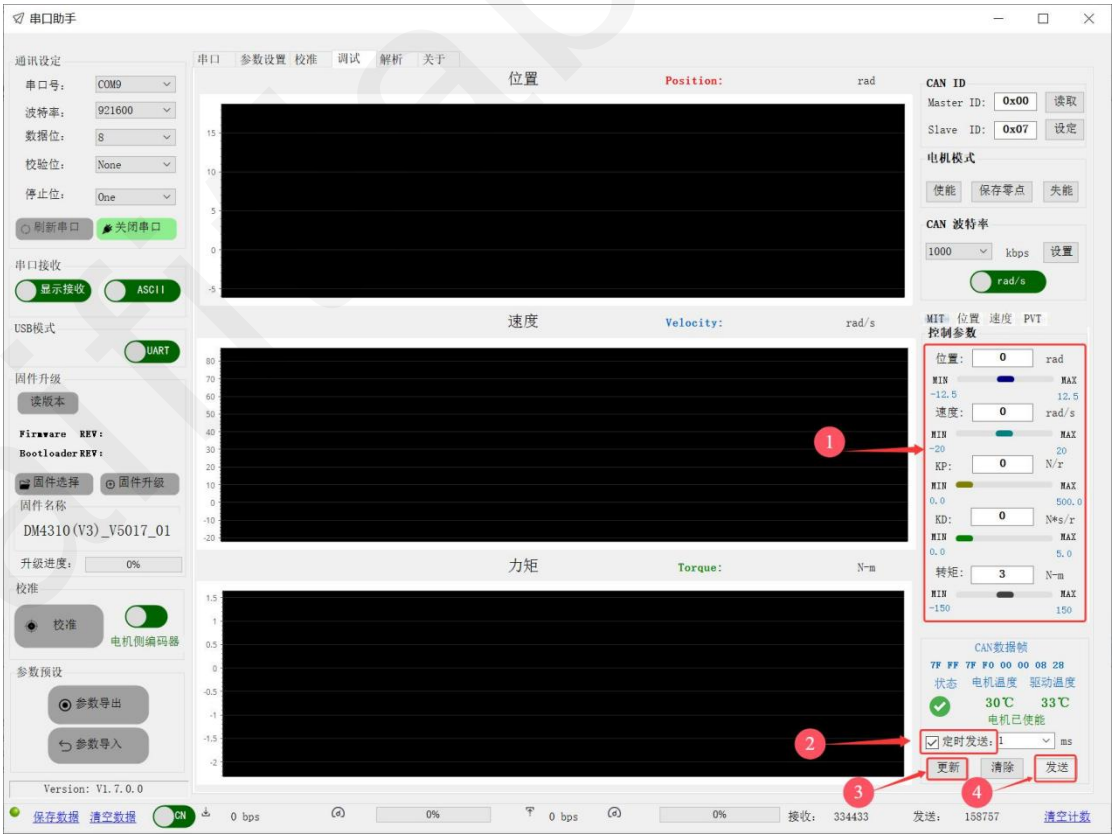
Step 1: Click the "Enable" button in the motor mode section. The driver's green

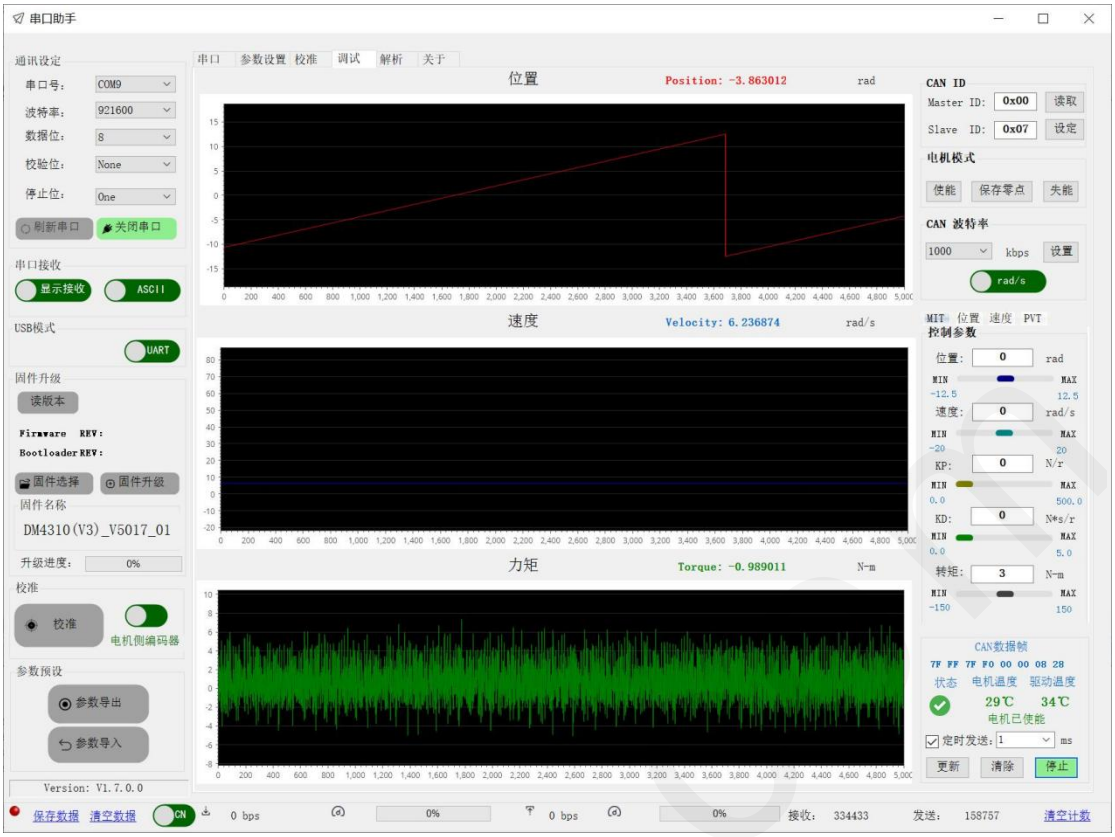
LED will light up, indicating the motor is enabled.



Step 2: Provide a torque setpoint for motor debugging.
Note: Under no-load conditions, even a very small torque setpoint will cause the motor to accelerate and rotate at its maximum speed.

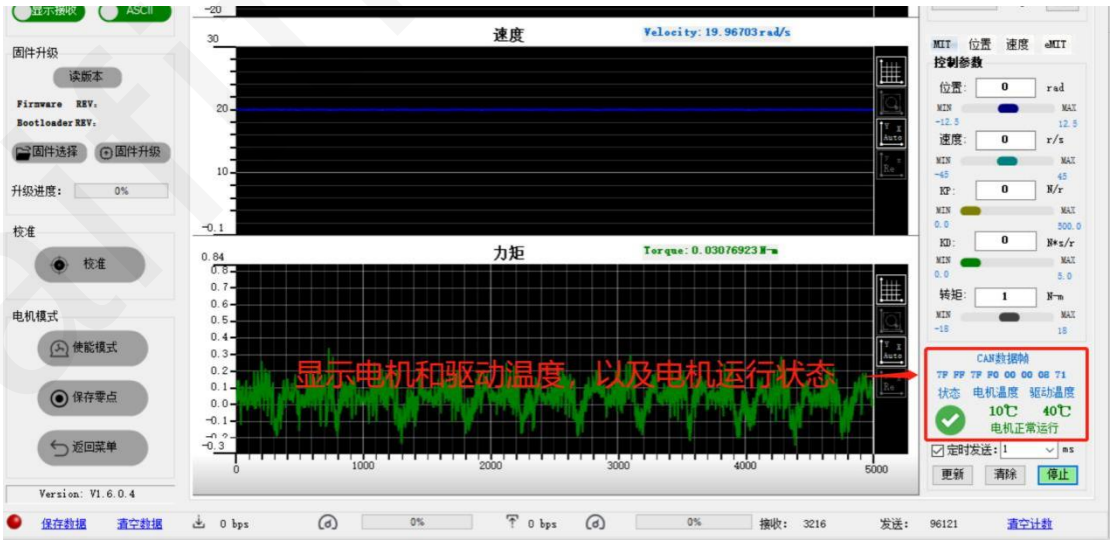
For example, given parameters: Torque setpoint: 1 N·m, all other parameters set to 0. Check the "Timed Send" box, then click the "Update" button followed by the "Send" button. The parameter curve changes can be viewed on the debugging interface. Secure the motor.



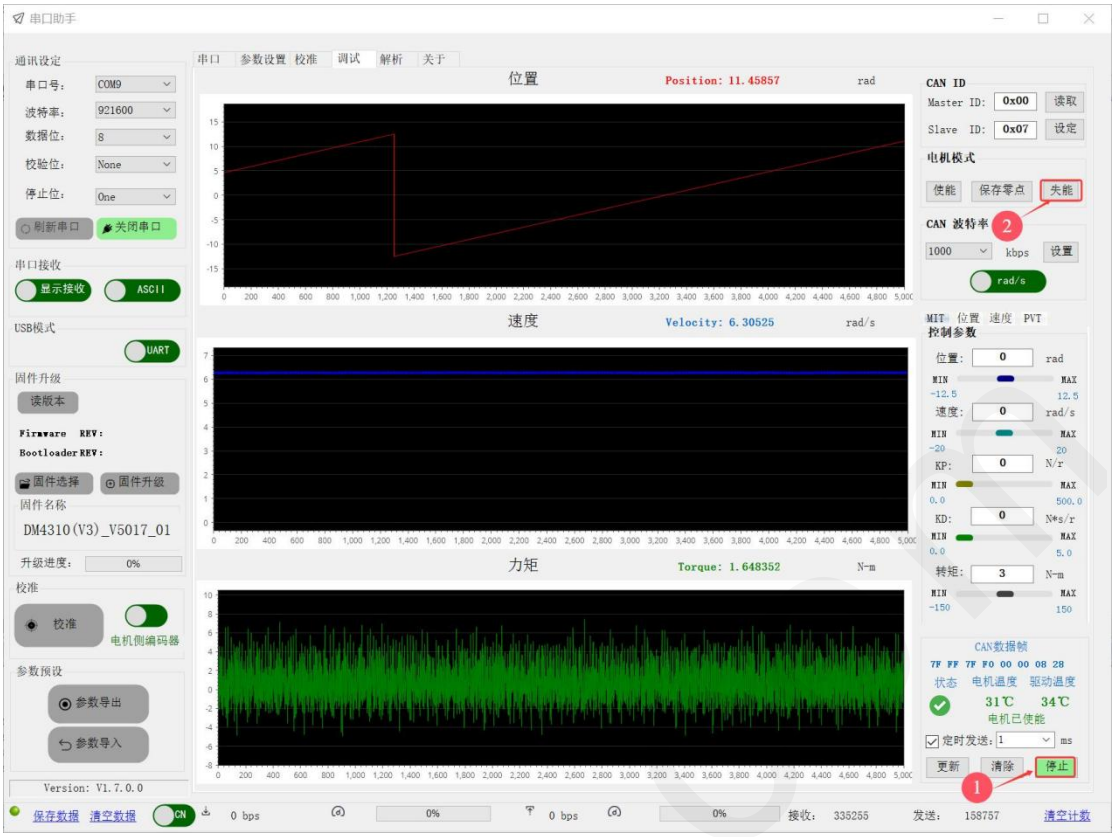


To modify control parameters during debugging, directly change the parameters on the interface, keep the "Timed Send" box checked, and click the "Update" button to proceed with debugging.

The debugging assistant interface displays the current motor and driver temperature, as well as motor operation status in real time. This information can also be viewed via the feedback frame. Refer to the "4.1 Feedback Frame" section for the feedback frame format and status types.

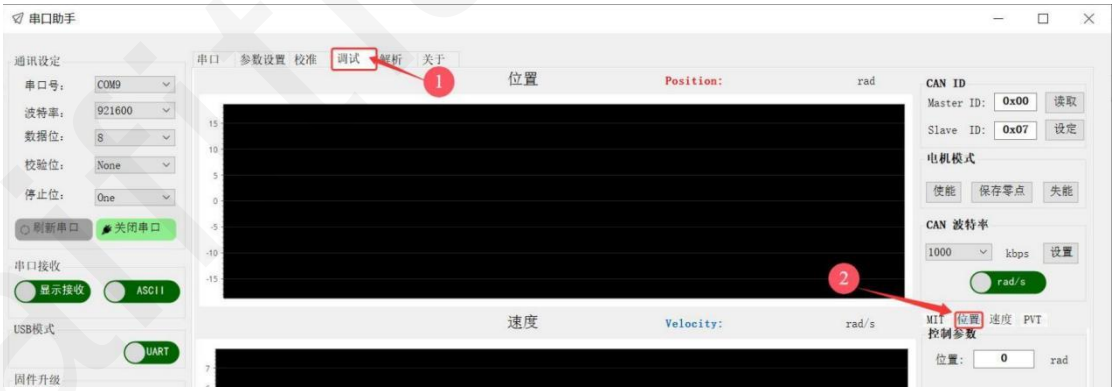


Step 3: To exit debugging, click the "Stop" and then the "Disable" buttons in sequence. The driver's red LED will light up, indicating exit from the motor operation mode.

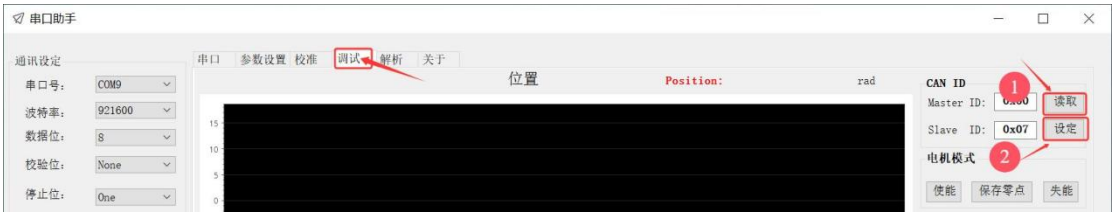


3. Position-Speed Mode

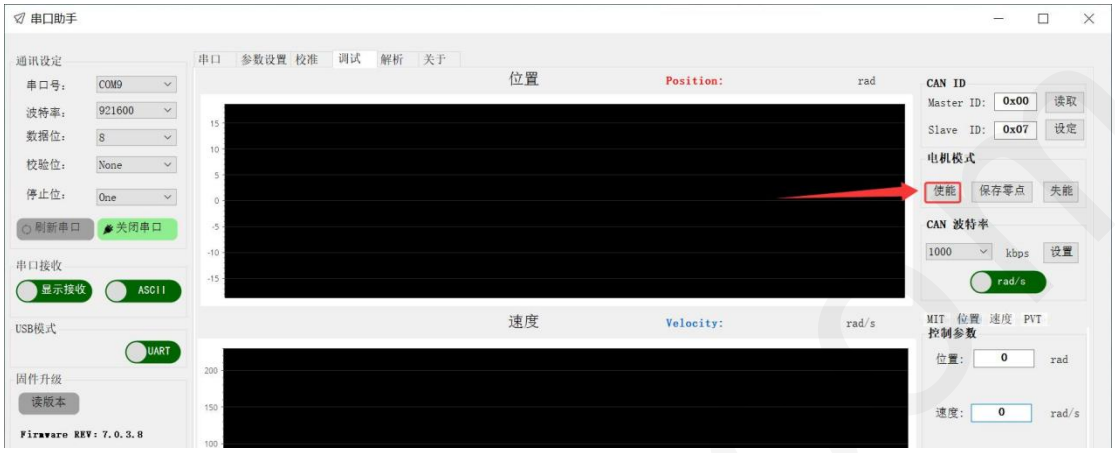
(1) To debug in Position-Speed Mode, first switch the motor mode to Position-Speed Mode on the parameter page. Click "Write Parameters" for the change to take effect. Then select the corresponding "Position" sub-tab in the debugging page.



(2) Ensure the CAN ID is correct (can be obtained from serial port print information or the parameter setting page, or can be set via the read/set buttons on the debugging page).

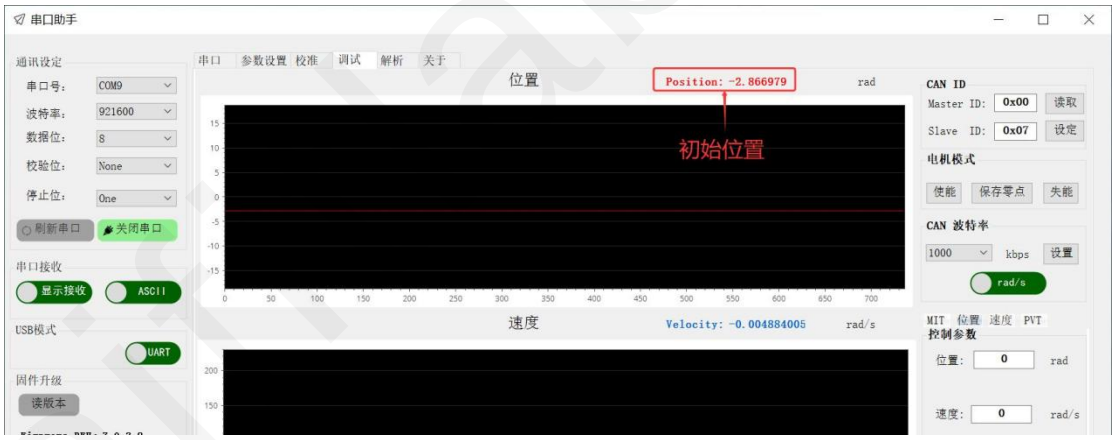


(3) Click the "Enable" button in the motor mode section. The driver's green LED will light up, indicating the motor is enabled.

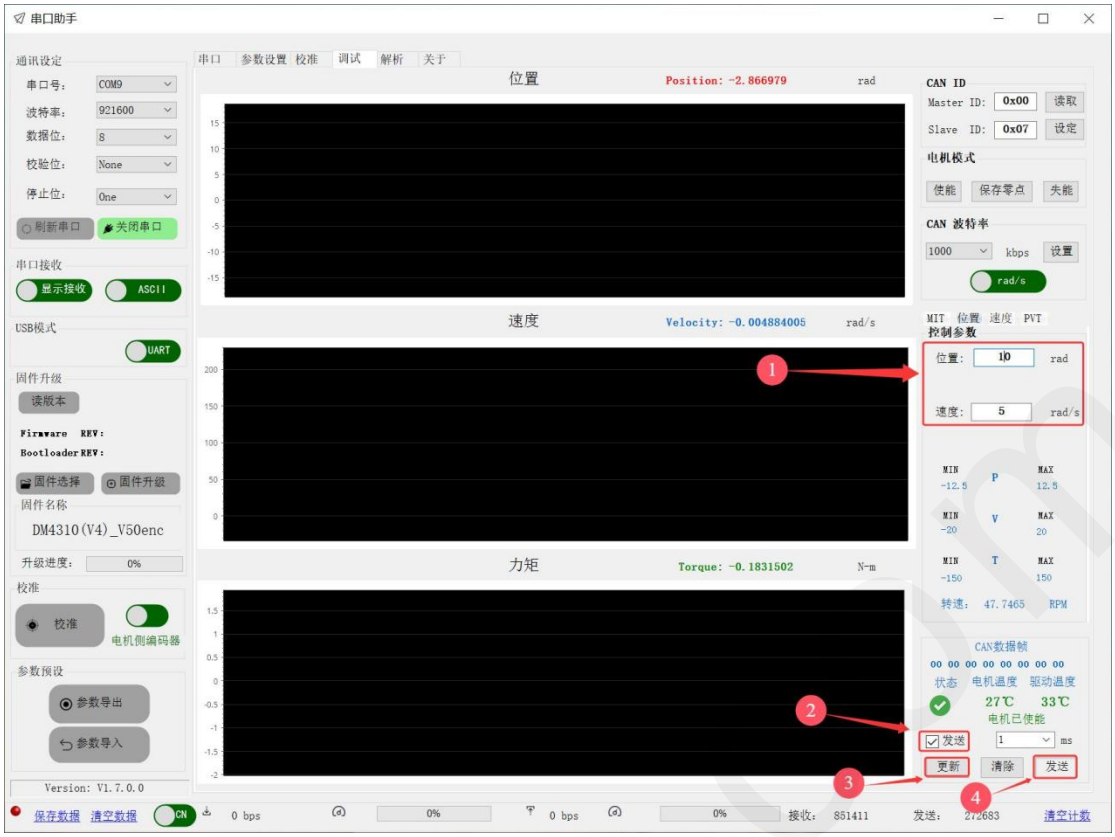


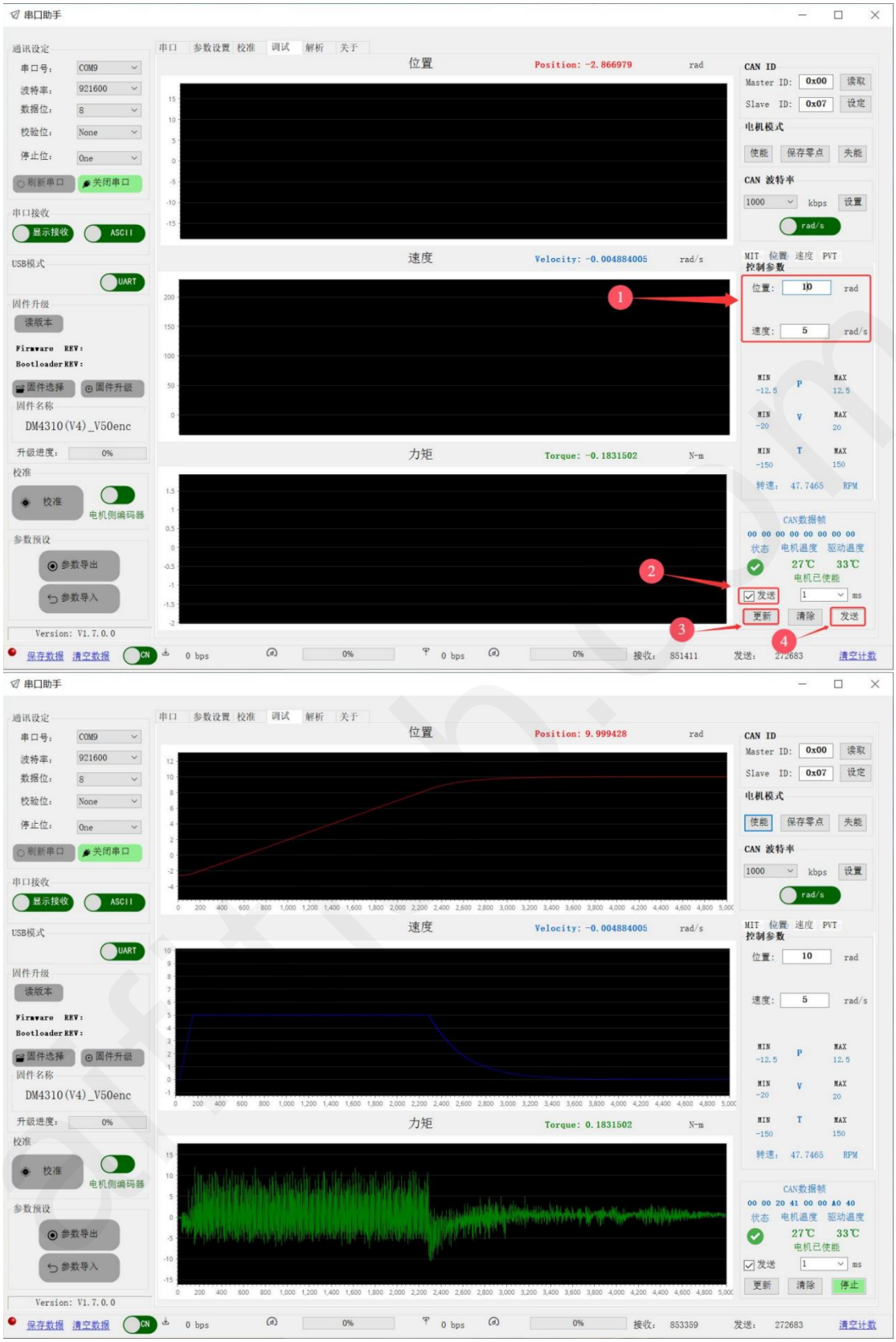
(4) Set the parameters. The motor will operate at the set speed to reach the specified position.

Before setting parameters, note the motor's initial position and use it as a reference for parameter configuration.



For example, set parameters: Position: 10 rad, Speed: 5 rad/s. Check the "Timed Send" box, then click the "Update" button followed by the "Send" button. The parameter curve changes can be viewed on the debugging interface. Secure the motor.

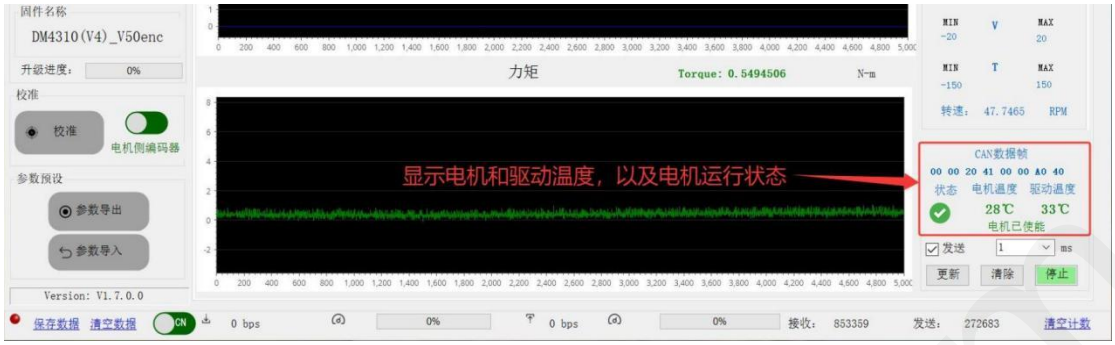

串口助手
通讯设定
串口号：
COM9
波特率：
921600
数据位：
8
校验位：
None
停止位：
One
刷新串口
关闭串口
串口接收
显示接收
ASCII
USB模式
UART
固件升级
读版本
Firmware REV:
Bootloader REV:
固件选择
固件升级
固件名称
DM4310(V4)_V50enc
升级进度：
0%
校准
校准
电机侧编码器
参数预设
参数导出
参数导入
Version: V1.7.0.0
串口
参数设置
校准
调试
解析
关于
位置
Position: -2.866979
rad
速度
Velocity: -0.004884005
rad/s
力矩
Torque: -0.1831502
N-m
CAN ID
Master ID:
0x00
读取
Slave ID:
0x07
设定
电机模式
使能
保存零点
失能
CAN 波特率
1000
kbps
设置
rad/s
MIT
位置
速度
PVT
控制参数
位置:
10
rad
速度:
5
rad/s
MIN
-12.5
P
MAX
12.5
MIN
-20
V
MAX
20
MIN
-150
T
MAX
150
转速：
47.7465
RPM
CAN数据帧
00 00 00 00 00 00 00 00
状态
电机温度
驱动温度
27℃
33℃
电机已使能
发送
1
ms
更新
清除
发送
1
2
3
4
保存数据
清空数据
CN
0 bps
0%
0 bps
0%
接收：
851411
发送：
272683
清空计数



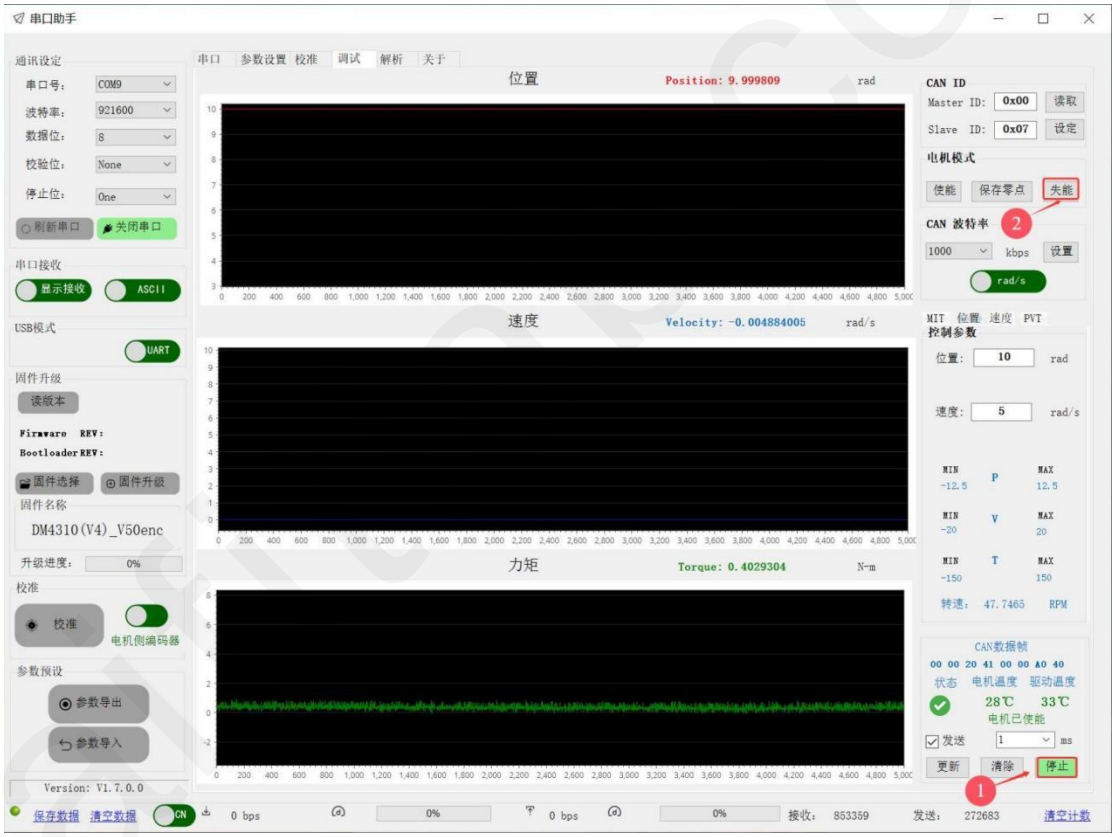
To modify control parameters during debugging, directly change the parameters on the interface, keep the "Timed Send" box checked, and click the "Update" button to proceed with debugging.

The debugging assistant interface displays the current motor and driver temperature,

as well as motor operation status in real time. This information can also be viewed via the feedback frame. Refer to the "4.1 Feedback Frame" section for the feedback frame format and status types.

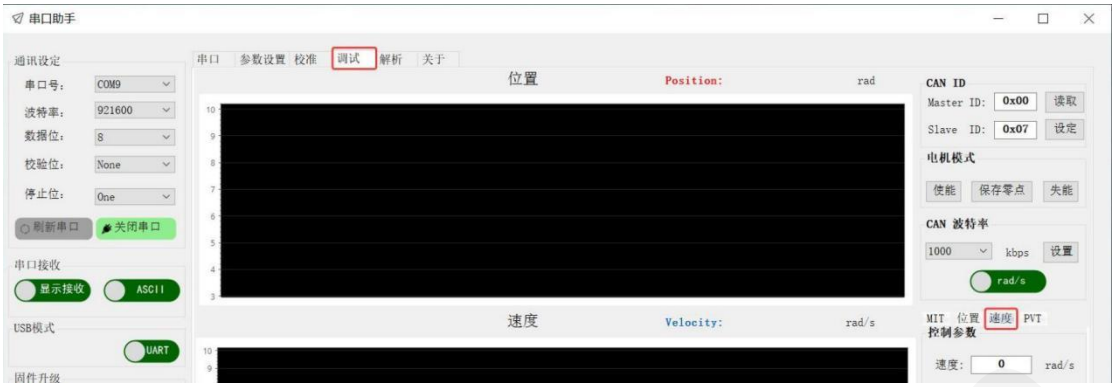


To exit debugging, click the "Stop" and then the "Disable" buttons in sequence. The driver's red LED will light up, indicating exit from the motor operation mode.

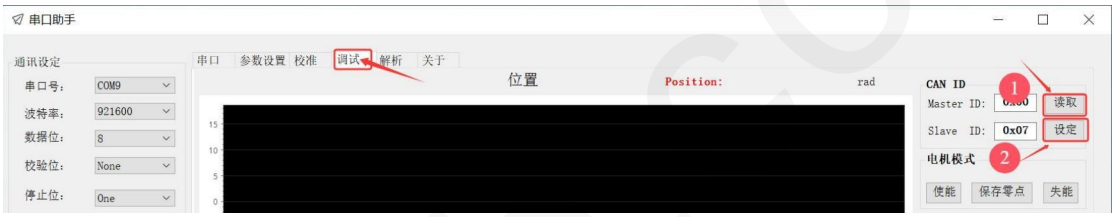


4. Speed Mode

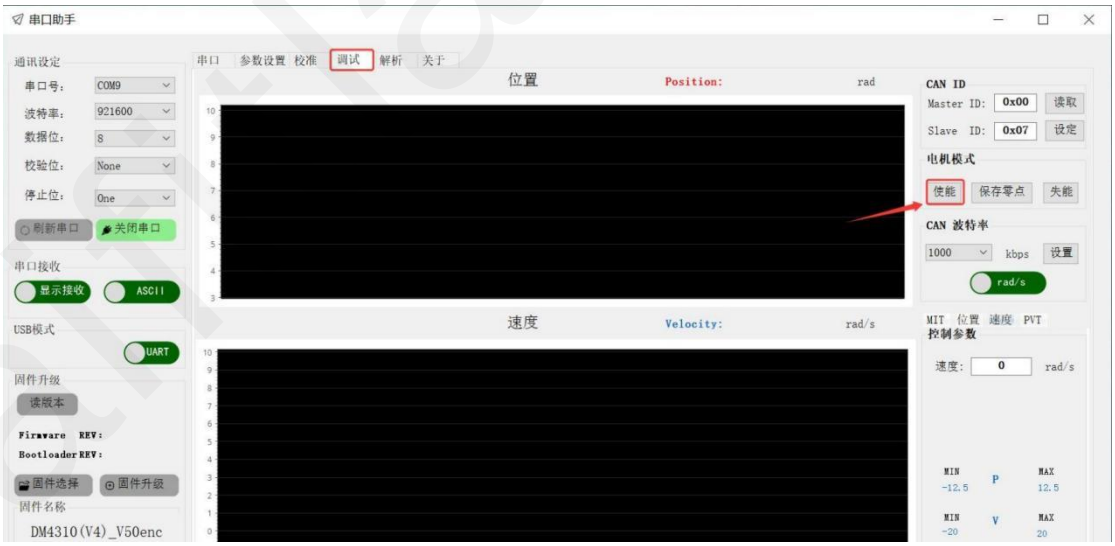
(1) Referring to Selection and Confirmation of Control Mode, ensure the current mode is set to Speed Mode. Confirm the current control mode and select the corresponding "Speed" sub-tab in the debugging page.



(2) Ensure the CAN ID is correct (can be obtained from serial port print information or the parameter setting page, or can be set via the read/set buttons on the debugging page).

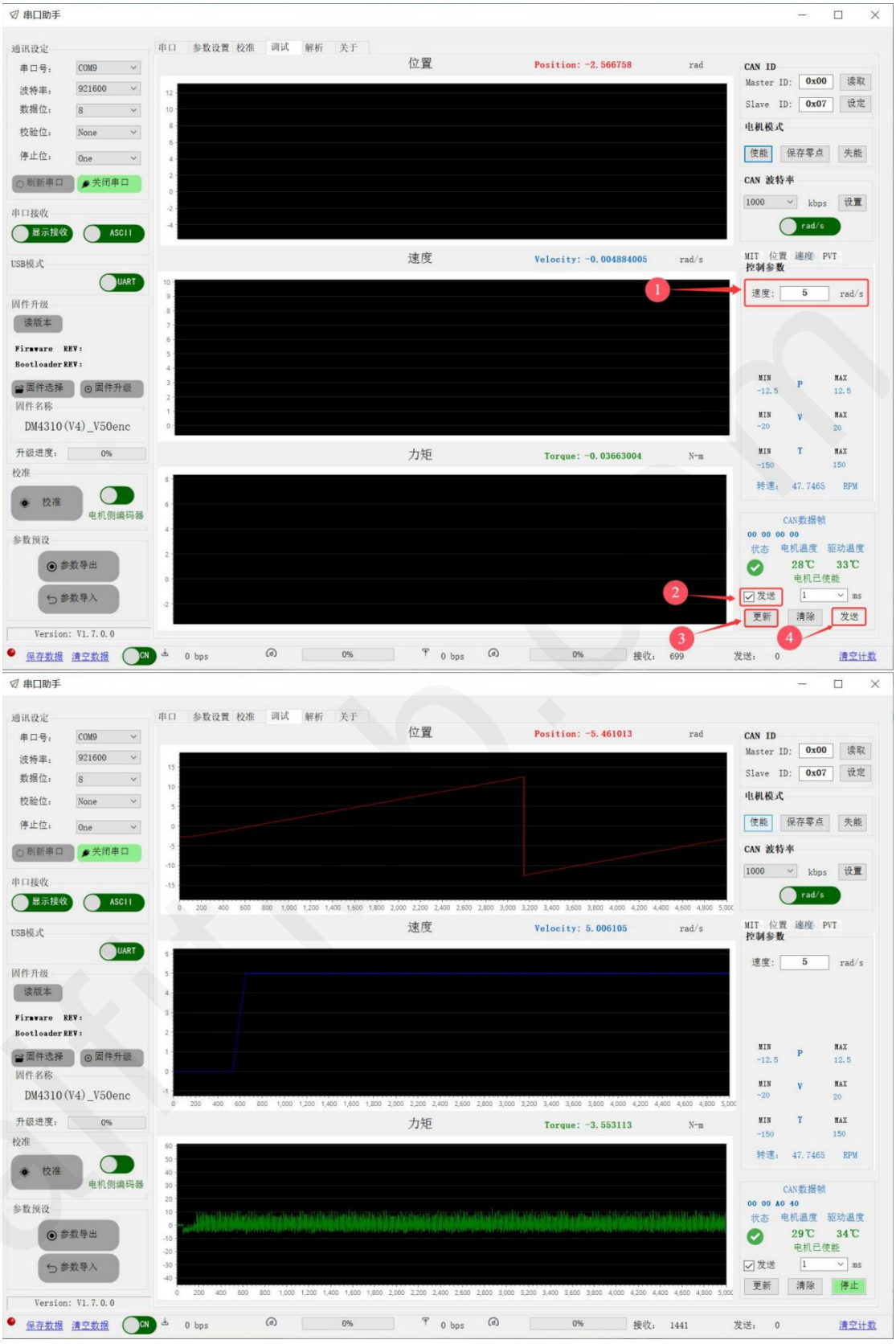


(3) Click the "Enable" button in the motor mode section. The driver's green LED will light up, indicating the motor is enabled.



(4) Set the parameters. The motor will operate at the set speed.

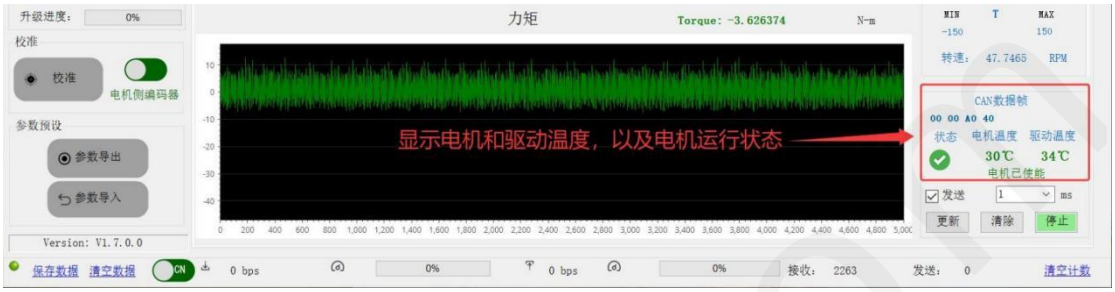
For example, set parameters: Speed: 5 rad/s. Check the "Timed Send" box, then click the "Update" button followed by the "Send" button. The parameter curve changes can be viewed on the debugging interface. Secure the motor.



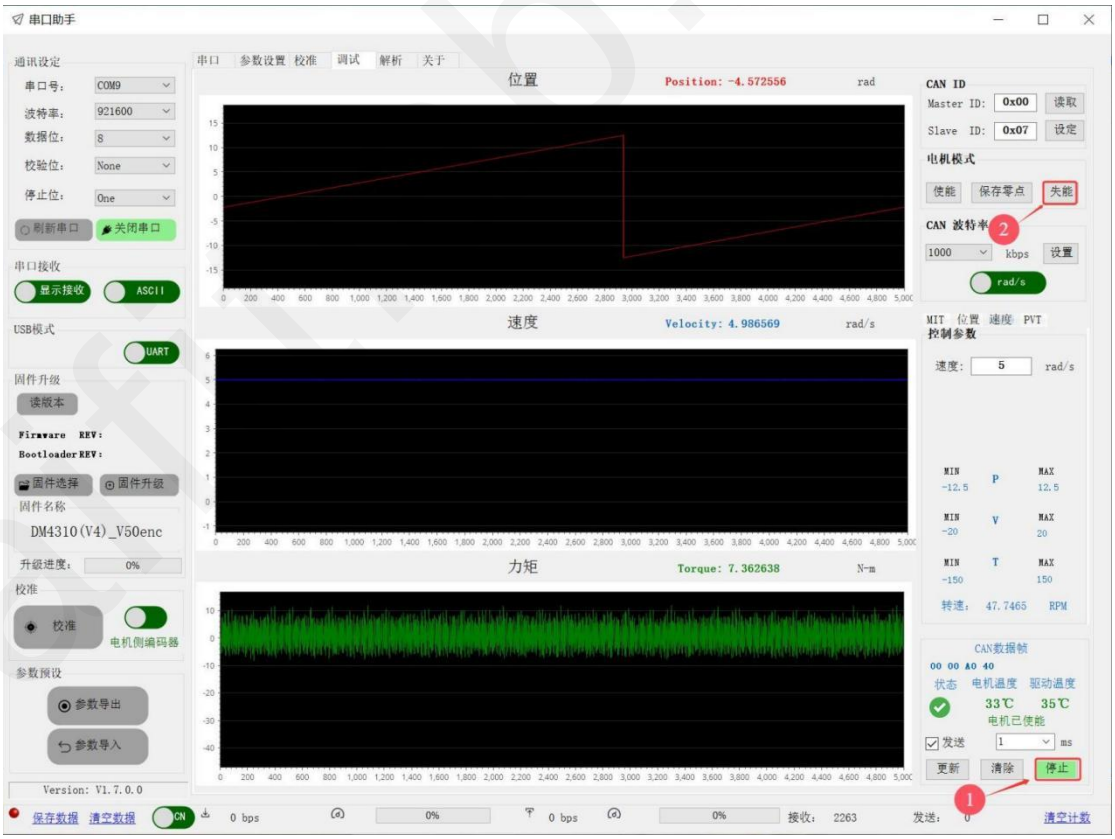
To modify control parameters during debugging, directly change the parameters on the interface, keep the "Timed Send" box checked, and click the "Update"

button to proceed with debugging.

The debugging assistant interface displays the current motor and driver temperature, as well as motor operation status in real time. This information can also be viewed via the feedback frame. Refer to the "4.1 Feedback Frame" section for the feedback frame format and status types.



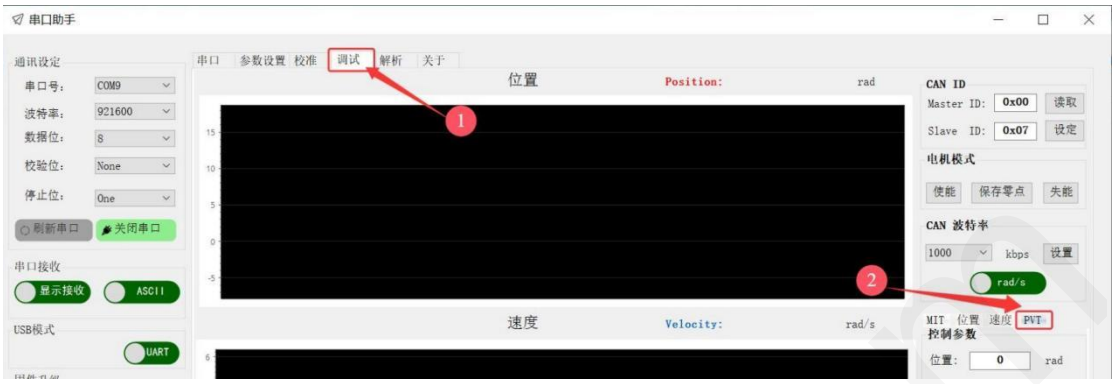
(5) To exit debugging, first send a speed command of 0 to stop the motor, then click the "Stop" and "Disable" buttons in sequence. The driver's red LED will light up, indicating exit from the motor operation mode.



5. Torque-Position Hybrid Mode

(1) To debug in Torque-Position Hybrid Mode, first switch the motor mode to PVT

Mode on the parameter page. Click "Write Parameters" for the change to take effect. Then select the corresponding "Position" sub-tab in the debugging page.。



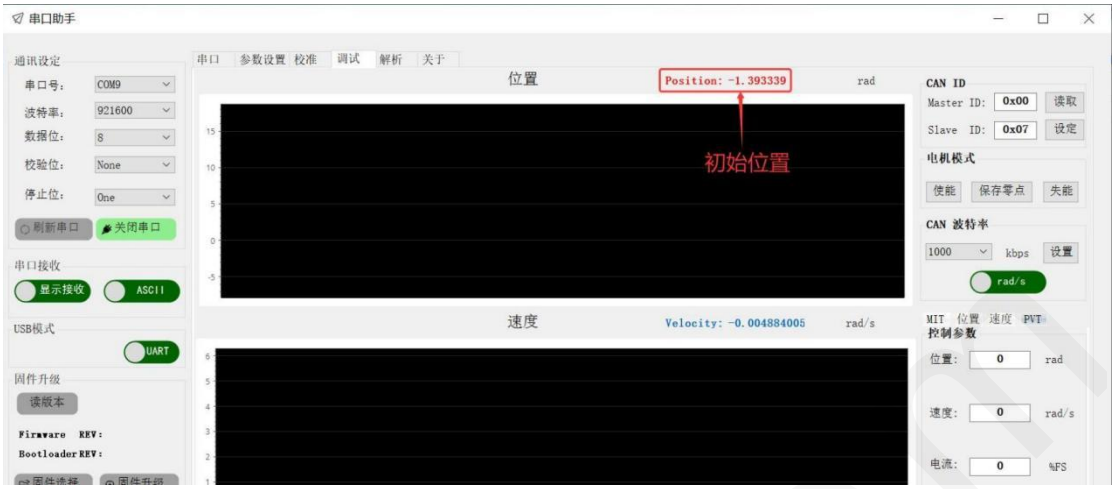
(2) Ensure the CAN ID is correct (can be obtained from serial port print information or the parameter setting page, or can be set via the read/set buttons on the debugging page).

(3) Click the "Enable" button in the motor mode section. The driver's green LED will light up, indicating the motor is enabled.

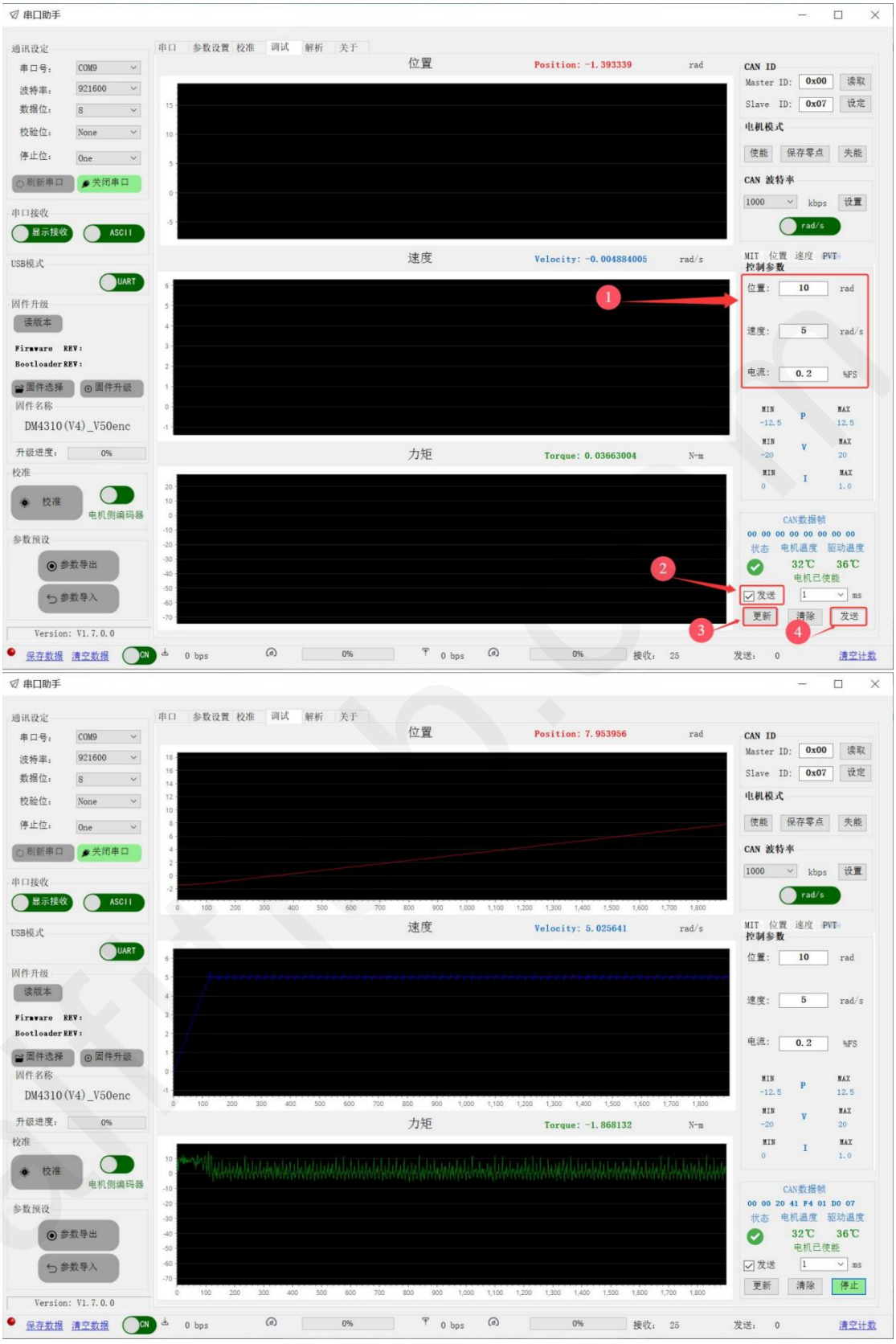
(4) Set the parameters. The motor will operate at the set speed to reach the specified position.

Before setting parameters, note the motor's initial position and use it as a

reference for parameter configuration.

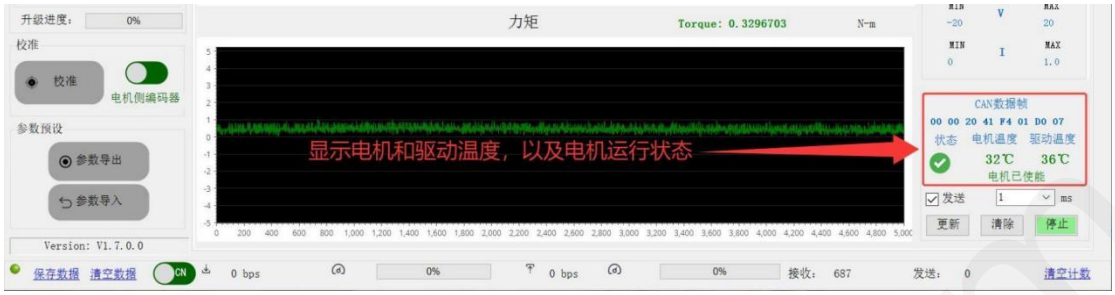


For example, set parameters: Position: 10 rad, Speed: 5 rad/s, Current: 20%. Check the "Timed Send" box, then click the "Update" button followed by the "Send" button. The parameter curve changes can be viewed on the debugging interface. Secure the motor.

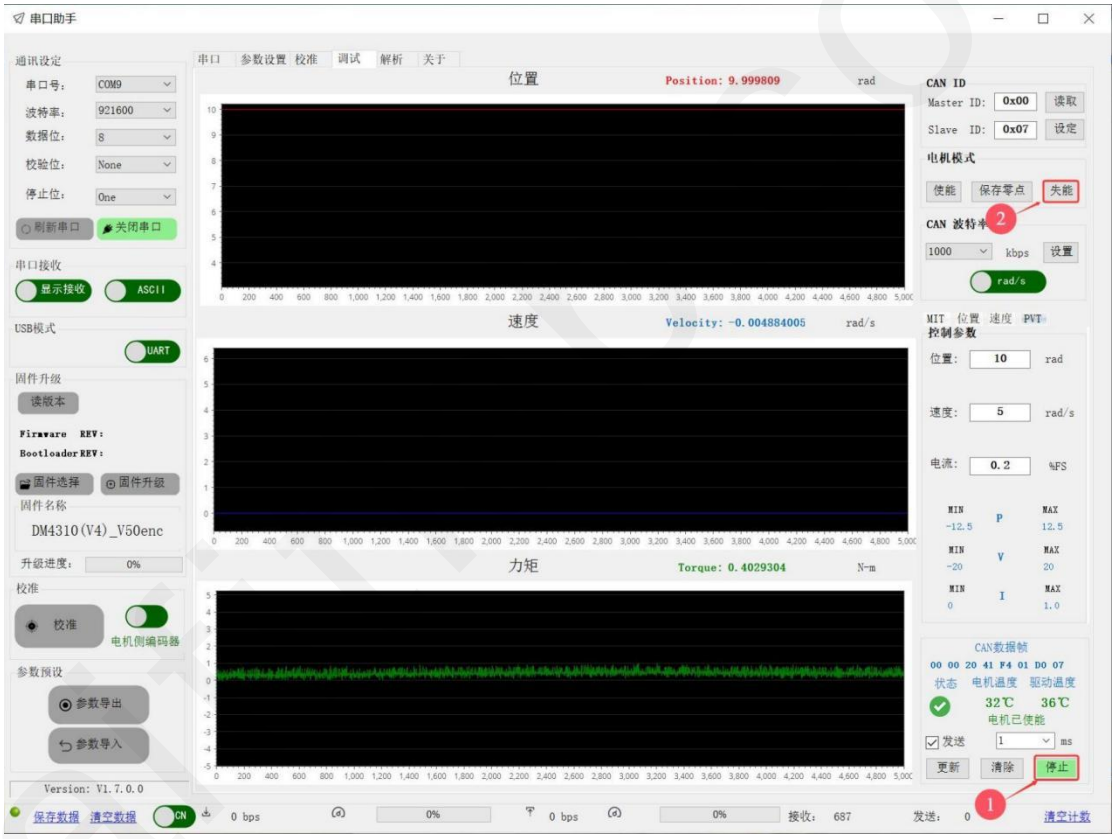


To modify control parameters during debugging, directly change the parameters on the interface, keep the "Timed Send" box checked, and click the "Update" button to

proceed with debugging.
The debugging assistant interface displays the current motor and driver temperature, as well as motor operation status in real time. This information can also be viewed via the feedback frame. Refer to the "4.1 Feedback Frame" section for the feedback frame format and status types.



To exit debugging, first ensure the motor has stopped, then click the "Stop" and "Disable" buttons in sequence. The driver's red LED will light up, indicating exit from the motor operation mode.



Firmware Version Check and Upgrade

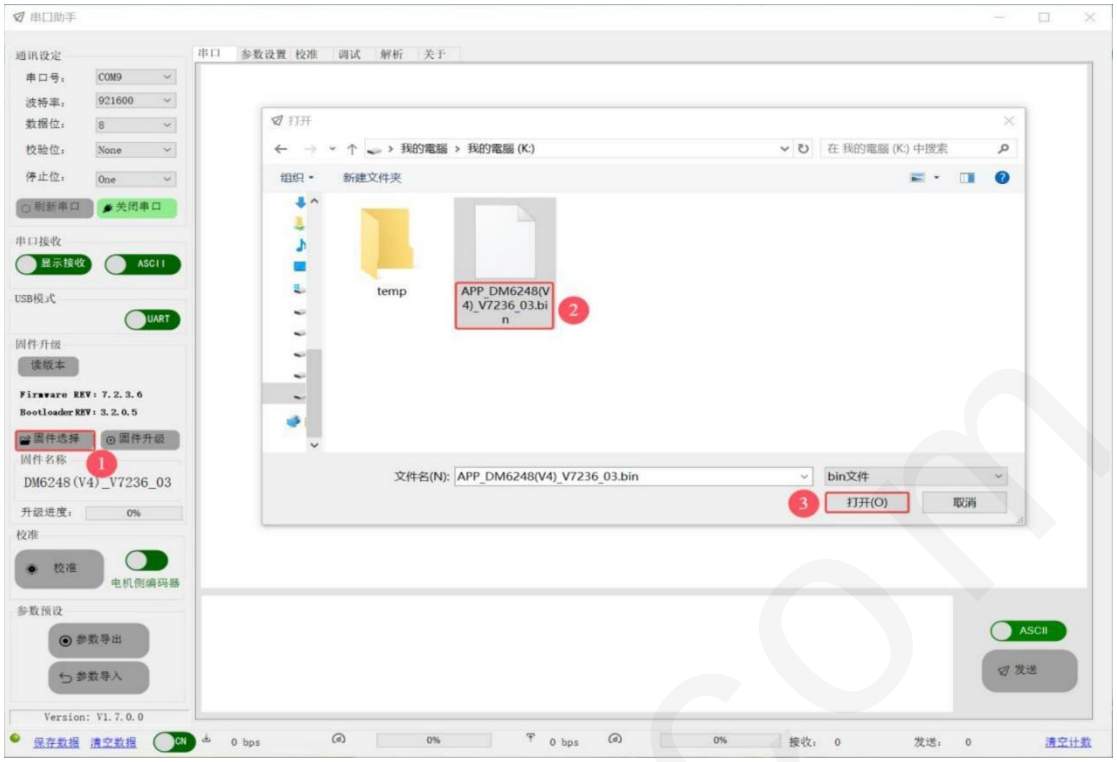
Version Check

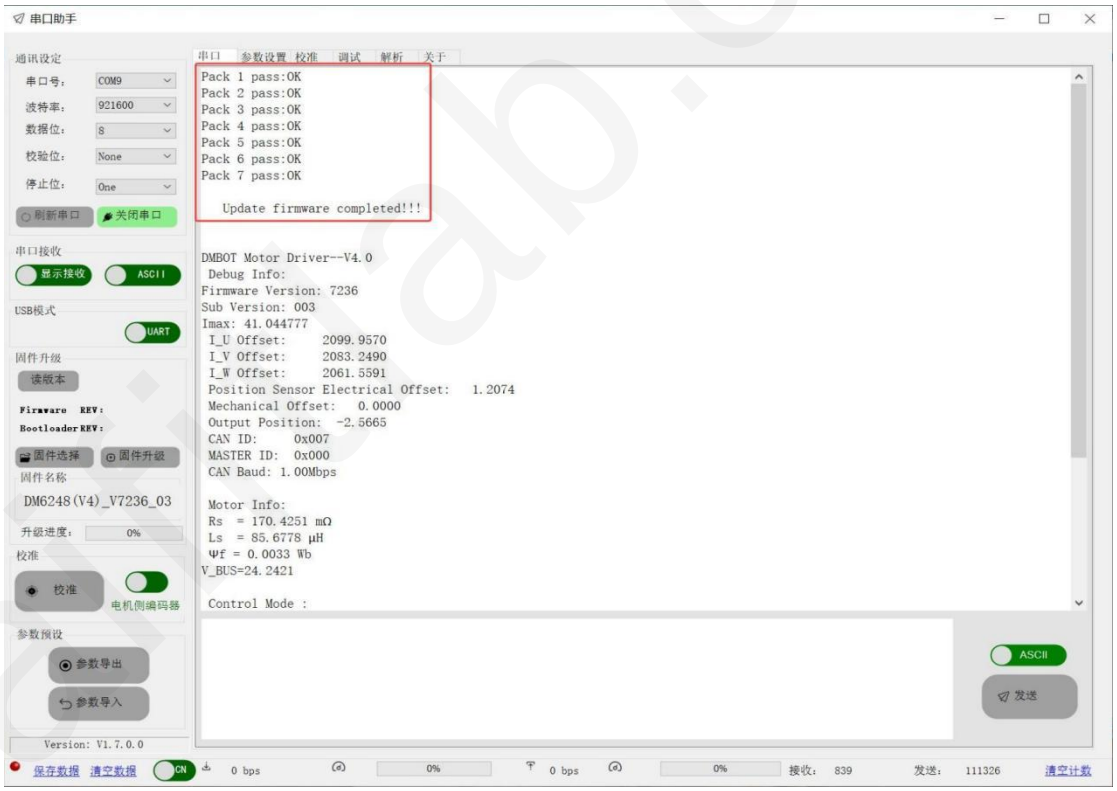
Connect the motor's serial port, CAN port, and power supply interface. Select the corresponding serial port device and open the serial port. Click "Read Version" to view the current driver's software version number.
Before upgrading the firmware, always read the current version first. Unless otherwise specified, select firmware with a version number starting with '72' for upgrade;

otherwise, it may cause unknown issues and unnecessary trouble.

Serial Port Firmware Upgrade

When firmware adds new features or provides a bug-fix version, users can upgrade via the serial port to resolve issues or use new functions. Before use, connect the serial port. Then click "Firmware Selection", choose the corresponding firmware file, and click "Open". The firmware name will be displayed. After confirming it's correct, click the "Firmware Upgrade" button and wait for the upgrade progress bar to complete. The completion can also be observed via the serial port interface.

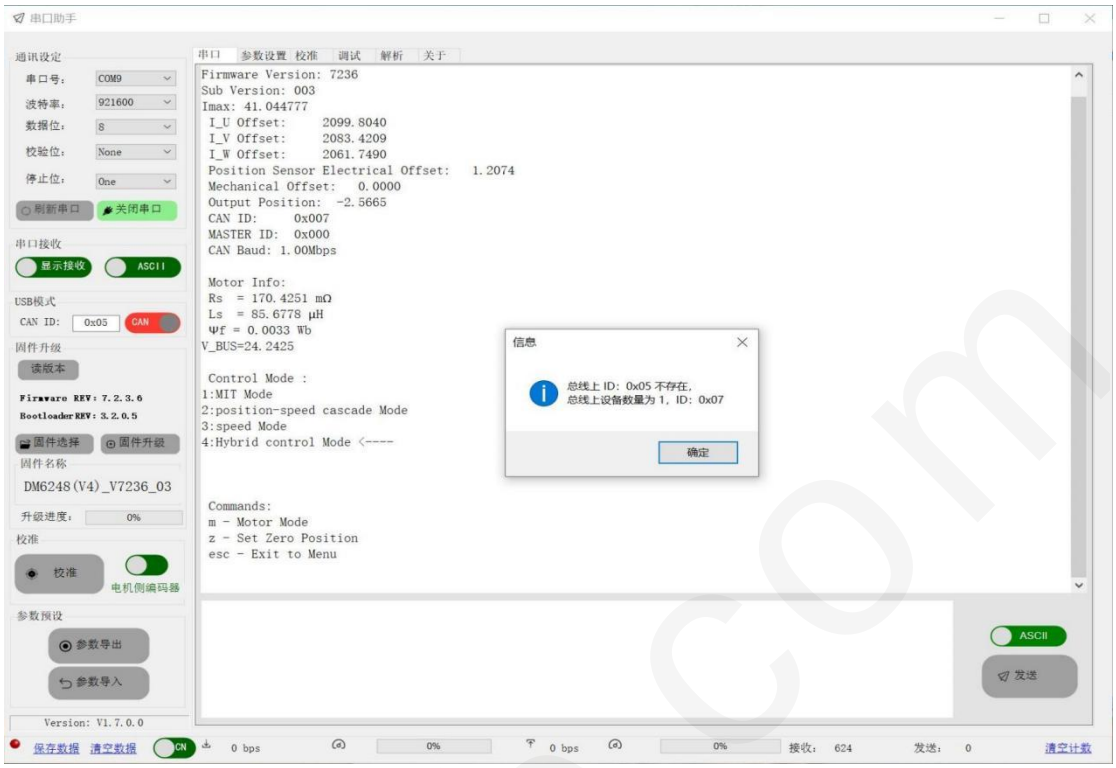




CAN Firmware Upgrade

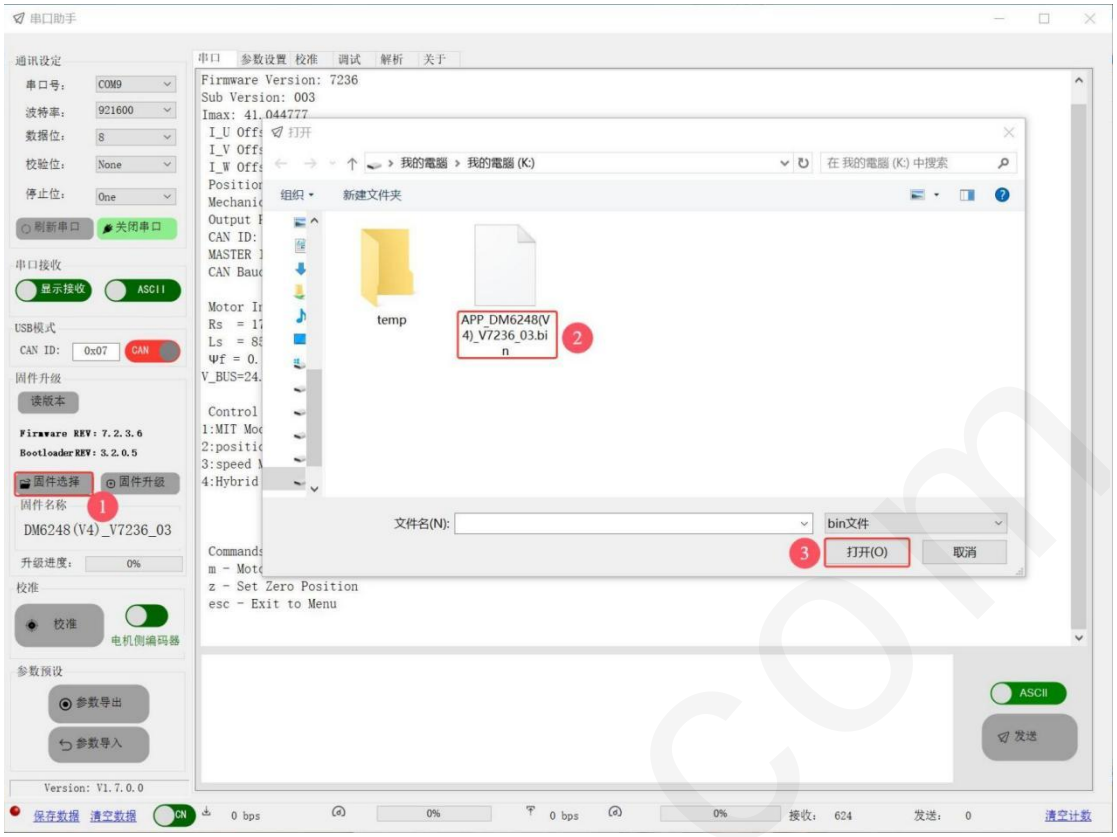
Bootloader version 3.2.0.5 and above supports firmware upgrade via CAN, at the baud rate configured for the driver board. Currently, only USB-to-CAN adapters are supported for upgrade. Before use, connect the CAN interface and ensure CAN communication is normal. Then, in the USB Mode section, switch from UART to CAN,

and enter the motor ID to be upgraded in the left box. If the bus does not exist, a corresponding prompt will appear, as shown below:



If the connection is successful, there will be no prompt.

Then click "Firmware Selection", choose the corresponding firmware file, and click "Open". The firmware name will be displayed. After confirming it's correct, click the "Firmware Upgrade" button and wait for the upgrade progress bar to complete. The completion can also be observed via the serial port interface.



After the upgrade is complete, a corresponding prompt will appear on the serial port interface: