DAMIAO | 达妙科技

DM-S2325-1EC Geared Motor

(Including DM3520-1EC Driver)

User Manual 2025.09.15

Disclaimer

Thank you for purchasing the DAMIAO DM-S2325-1EC motor (hereinafter referred to as "the Motor") and DM3520-1EC driver (hereinafter referred to as "the Driver"). Before using this product, please carefully read and comply with this document and all safety guidelines provided by DAMIAO Technology. Failure to do so may cause harm to you or others, or damage the product or surrounding items. By using this product, you acknowledge that you have read, understood, and accepted all terms and conditions of this document and any related materials. You agree to use the product solely for legitimate purposes and assume full responsibility for its use and any resulting consequences. DAMIAO Technology shall not be liable for any damages, injuries, or legal liabilities arising from the direct or indirect use of this product.

Precautions

1. The driver must be operated strictly within the specified working environment and maximum allowable temperature range of the winding; otherwise, permanent irreversible damage may occur to the product.

2. Prevent conductive foreign objects from entering the drive unit, as this may cause abnormal operation.

3. Before use, check all components for integrity. If any parts are missing, aged, or damaged, stop using the device.

4. Ensure proper wiring and secure connection to the motor.

5. Do not touch the drive during operation to prevent accidents. The drive may overheat when delivering high current, so avoid burns.

6. Do not remove the drive without permission to avoid damage.

7. Do not apply force to the three-phase wires or 4-pin wire exit points of the motor, as this may cause permanent and irreversible damage to the product.

8. For motors requiring continuous operation at rated or higher loads, it is recommended to replace the grease every seven hours to prevent grease degradation and subsequent wear of the motor's side gears.

Motor Features

1. The driver features intelligent design, enabling plug-and-play operation after motor replacement without recalibration or parameter adjustment.

2. Support firmware upgrades.

3. The host computer provides visual parameter adjustment, enabling immediate use with simple configuration.

4. Supports CAN FD functionality with a maximum baud rate of 5Mbps.

5. The CAN bus provides real-time feedback on motor parameters including speed, position, torque, and temperature.

6. Features dual-temperature protection.

7. High speed, high torque.

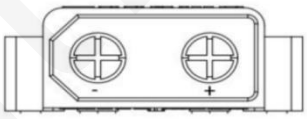
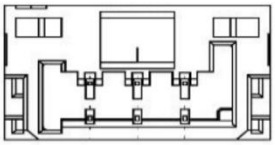
8. Flexible switching between multiple control modes.

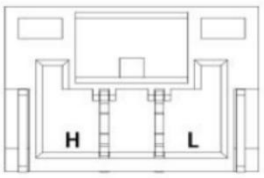
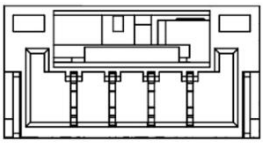
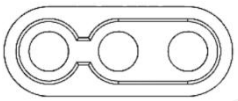
Package Contents

Class	Inventory	Key
Motor set	1 Drive × 1 2. Reducer motor × 1 3. XT30 Power Cord (Double-ended, 200mm) × 1 4. 3-pin serial cable (crossed connectors, 300mm length) × 1 5. 2-pin CAN cable (crossed, 300mm) × 1	

Single Motor	1. Reducer motor × 1	
Single Drive	1. Motor rotor × 1	

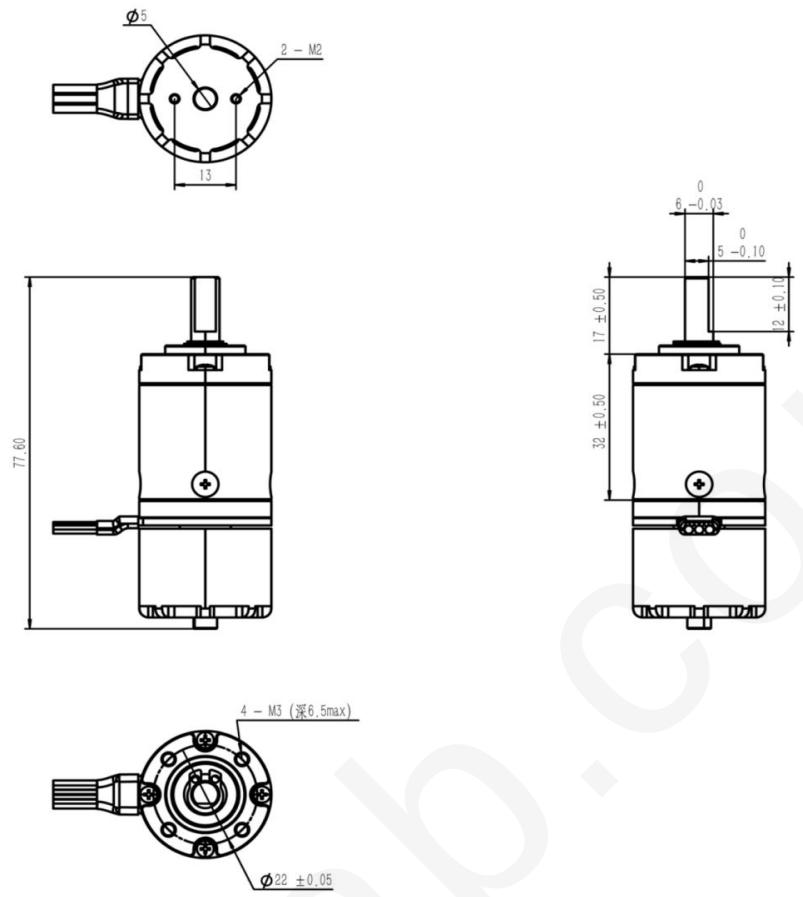
Interface and Wire Sequence Description

Specific Name-Number	Interface annotation	Explain
Power supply connection interface		Connect the power supply to the motor using the XT30-F plug's power cable, rated at 24V.
Gorge line		For serial port debugging, parameter adjustment, and other functions

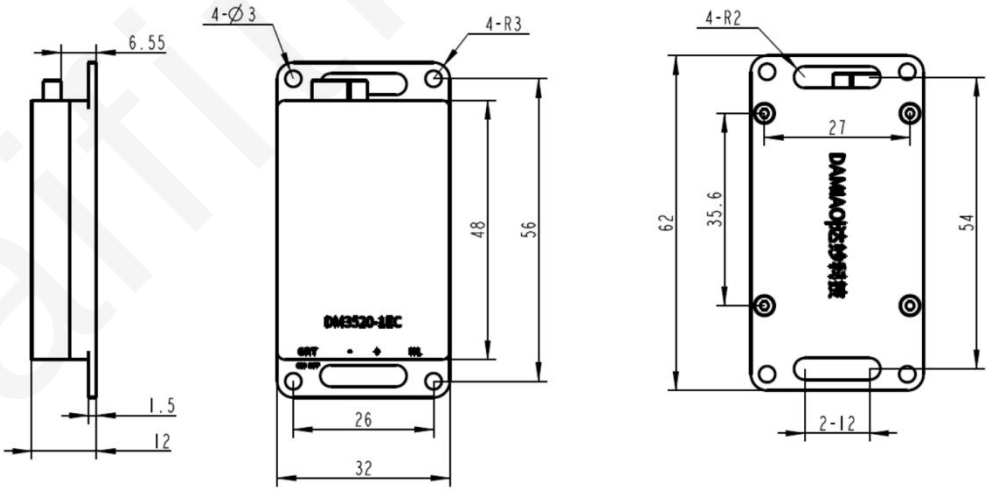
CAN interface		Control driver
Encoder interface		Connects to the DM-S2325-1EC motor using the matching 4-pin cross-type cable.
Three-phase line interface		Used to connect the three-phase wires of the motor

Dimension and Installation of Reducer & Driver

Refer to the drive installation hole size and position to install the drive on the corresponding device.



Dimension diagram of DM-S2325-1EC Geared motor

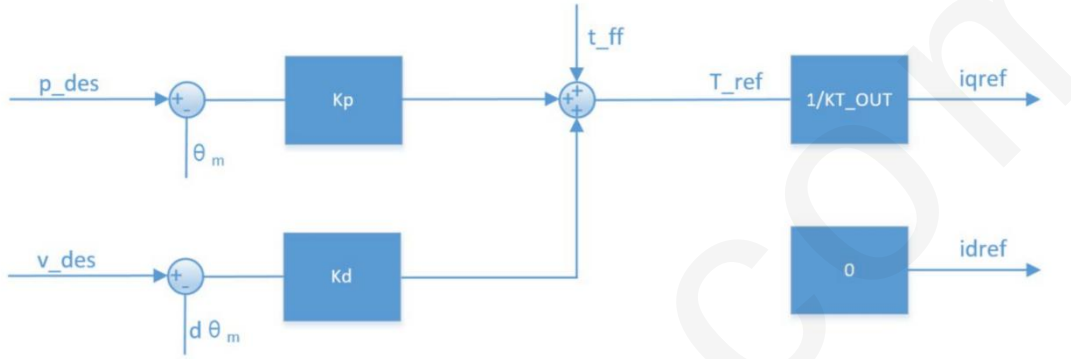


DM3520-1EC driver size diagram

Work Pattern

MIT Mode

The MIT mode is designed to be compatible with the original MIT mode, enabling seamless switching while allowing flexible control range settings (P_MAX, V_MAX, T_MAX). The driver converts received CAN data into control variables, which are then processed to generate torque values for the current loop. The current loop ultimately achieves the target torque current according to its regulation rules. The control block diagram is shown below:



The MIT model can be adapted to various control configurations. For instance, when kp=0 and kd≠0, a constant-speed rotation can be achieved by setting v_des. Similarly, when kp=0 and kd=0, torque output can be controlled by setting t_ff.

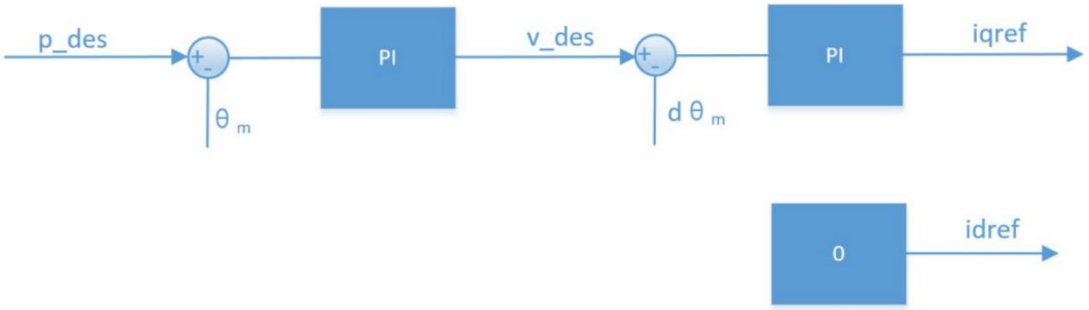
Note:

1. After power-on, the motor position is fixed at 0.0 rad.

2. When controlling position, the kd value must not be set to 0, as this may cause motor oscillation or even loss of control.

Velocity Position Mode

The position cascade control mode employs a three-loop series configuration. The outermost position loop provides the speed loop's setpoint, while the inner current loop receives the speed loop's output to regulate actual current. The control block diagram is shown below:



p_des is the target position for control, while v_des defines the maximum absolute speed during motion.

When the cascade mode is controlled using the parameters recommended by the debugging assistant, it achieves better control accuracy with a relatively smooth process, though the response time is longer. In addition to v_des, other configurable parameters include acceleration/deceleration settings.

If additional oscillations occur during the control process, the acceleration/deceleration can be increased.

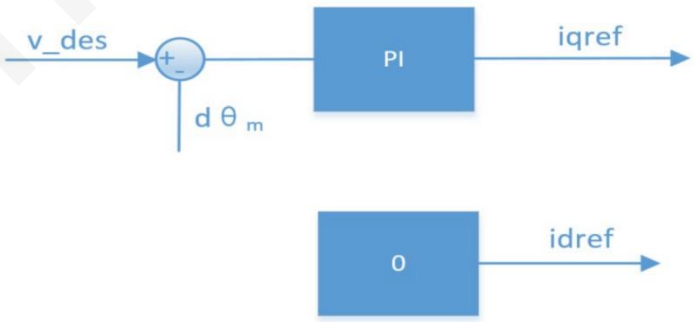
pay attention to：

1. Upon power-on, the motor position is fixed at 0.0 rad.

2. p_des and v_des are measured in radians (rad) and radians per second (rad/s) respectively, with the data type being float.

Speed mode

The speed mode enables the motor to operate stably at the set speed, as shown in the control block diagram below.



pay attention to：

1. v_des is measured in radians per second (rad/s) and is a float data type.

Control protocol description

The system employs CAN standard frame formats with a default baud rate of 1Mbps, which can be adjusted to different rates via commands (see the Baud Rate Modification section for details). Functionally, frames are categorized into receiving frames and feedback frames. Receiving frames contain control data for motor command execution, while feedback frames transmit status data from the motor to the upper controller. The feedback mechanism operates on an inquiry-based principle: when the driver detects a frame ID matching the motor's configured CAN ID (with low 8 bits for parity check and high 3 bits ignored), it sends the current status data to the bus. Although the receiving frame format and frame ID vary by motor mode, the feedback frame format and data remain consistent across all modes.

Feedback frame

The feedback frame ID is set by the debug assistant (Master ID), with a default value of 0. It primarily provides feedback on the motor's position, speed, and torque. The frame format is defined as:

feedback 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

among：

ID indicates the controller's identifier, with the lower 4 bits of CAN_ID (ERR) representing the status, corresponding to the following status types:

0-Disability;

1—Enable;

5 - Read the sensor error;

6-Read motor parameter error;

8—overpressure;

9-Under-voltage;

A—— excess current；

B—MOS overtemperature;

C-Motor coil overheating;

D-Communication loss;

E—— overload ;

POS indicates the motor's position information*.

VEL indicates the motor speed information*.

T indicates the motor's torque information*.

T_MOS is the average temperature of the upper MOS, in ℃

T_Rotor is the average temperature of the motor's internal coils, measured in ℃

The position, speed and torque are converted to fixed-point data with sign by linear mapping, where the position is 16-bit data, and the speed and torque are 12-bit data.

Note:

1. Upon power-on, the motor position is fixed at 0.0 rad.

2. The position is measured in radians (rad) and indicates the output shaft's position after deceleration. All subsequent position descriptions follow this definition, which will not be repeated here.

3. The unit of speed is rad/s (radians per second), representing the output shaft's speed after deceleration. All subsequent descriptions of speed shall follow this definition and need not be repeated.

4. The unit of torque is Nm, representing the torque of the output shaft (i.e., the torque after reduction). All subsequent descriptions of torque shall follow this definition and need not be repeated.

Control frame in MIT mode

control 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 equals the set CAN ID value

P_des: position given

V_des: Speed given

Kp: Position Ratio Coefficient

Kd: Positional differential coefficient

T_ff: Torque setpoint

All parameters adhere to the mapping relationships specified in the previous section. The values of p_des, v_des, and t_ff can be configured via the debugging assistant, while Kp ranges from 0 to 500 and Kd ranges from 0 to 5.

A standard CAN data frame is 8 bytes long. MIT's control command format encodes the five parameters—Position, Velocity, Kp, Kd, and Torque—into 8 bytes using bit combinations. Specifically, Position occupies 2 bytes (16 bits), Velocity 12 bits, Kp 12 bits, and Kd 12 bits.

Control frame in position velocity mode

control 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 given, floating-point type, with the lower bit first and the higher bit last

V_des: Speed given, floating-point, lower bits first, higher bits last

The CAN ID for the command is 0x100+ID. The speed setting is the maximum speed defined during motor operation.

give an example：

Set the motor with CAN ID 5 to rotate 180 degrees (3.14159 rad) at a maximum speed of 10 rad/s, as instructed below:

frame ID	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
0x105	0xCF	0x0F	0x49	0x40	0x00	0x00	0x20	0x42

Control frame in speed mode

control 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: A floating-point variable with fixed speed, where the lower bit is first and the higher bit is last. The CAN ID for this command is 0x200+ID.

give an example：

Set the motor with CAN ID 3 to rotate at 60rpm (6.283 rad/s) using the following command:

frame ID	D[0]	D[1]	D[2]	D[3]
0x203	0x56	0x0E	0xC9	0x40

Direction for Use

The motor is factory-calibrated and marked inside. Under normal circumstances, it can be used directly without recalibration. The following provides the calibration and marking process, which can be skipped under normal conditions.

Motor Calibration

The calibration aims to correct installation errors and sensor orientation. During calibration, the motor will rotate in a specific direction. Ensure the motor rotates freely, preferably under no-load conditions, as this may cause calibration failure.

The Damiu Debug Assistant can be used to calibrate the motor via serial communication. Any serial tool supporting a baud rate of 921600 is compatible. For example, the calibration process using USB to CAN conversion is demonstrated below:

1. Connect the motor and driver using three-phase wires and sensor wires, ensuring a secure and tight connection.

2. Connect the driver to the USB-to-CAN converter using a 3-pin serial cable, ensuring a secure and stable connection.

3. Connect the USB-to-CAN converter to the PC and enable the host computer in the debugging assistant.

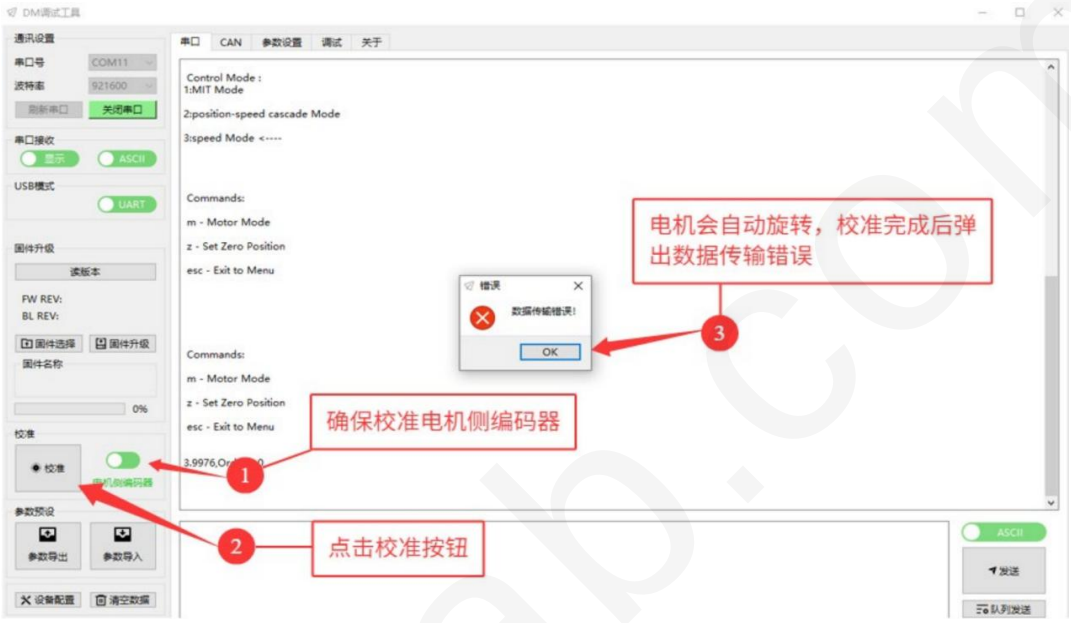
4. Use the following parameters to configure the serial port.

5. Select the right side as the motor-side encoder

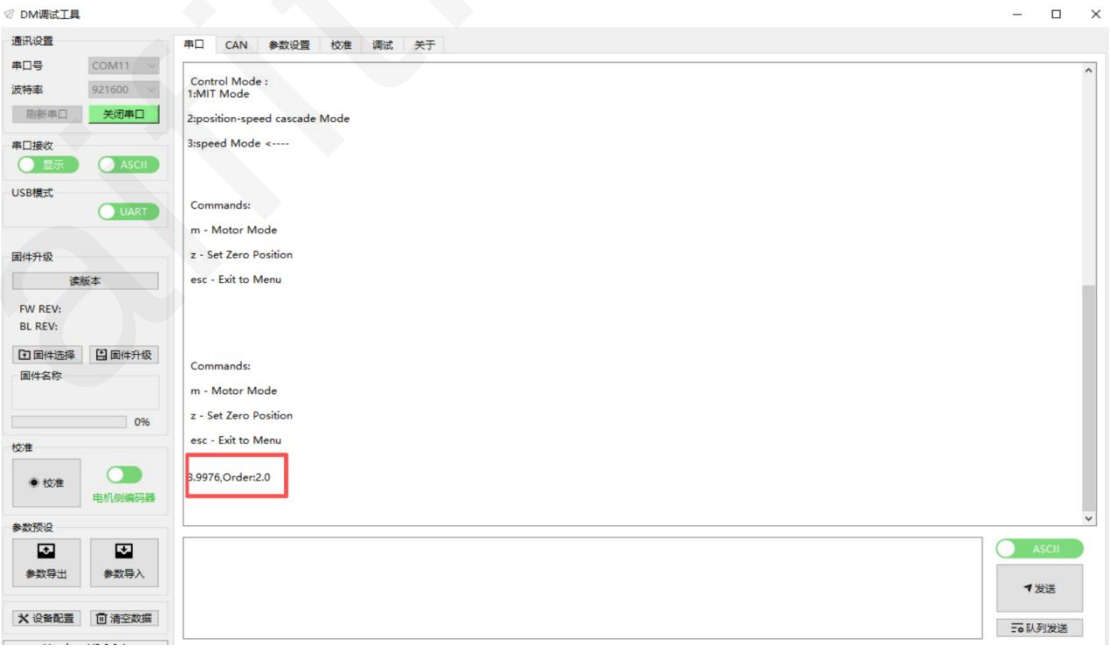
6. Supply the motor with 24V (calibration current: approximately 0.3A; use a power supply rated at least 0.3A).

7. Click Calibrate, the motor will rotate freely. The entire process takes about 2 seconds.

8. After calibration is completed, the driver sends a message to the serial port, which may cause the host computer to display a 'data transfer error' —this can be ignored.



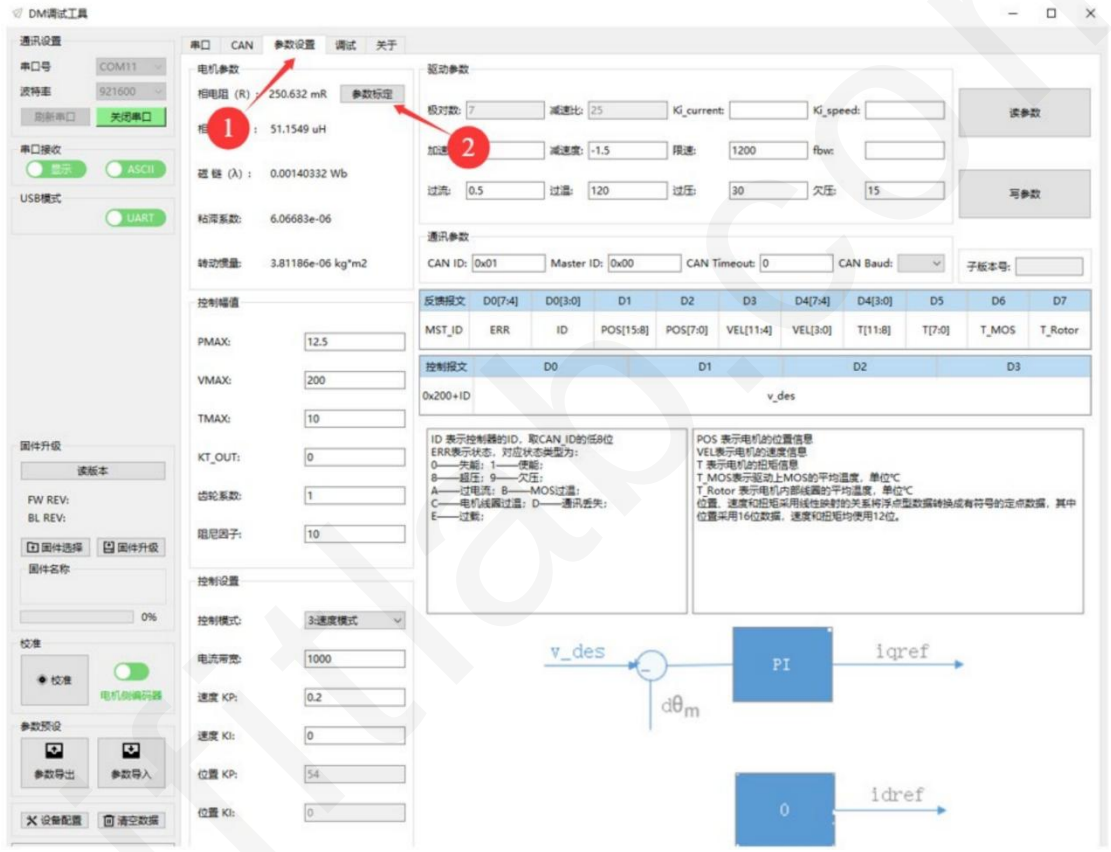
9. Back to the serial interface, the driver prints the current calibration information to the serial port, marking the completion of calibration.



Parameter Calibration

After calibration is completed, parameter identification can be performed, primarily to determine key parameters such as phase resistance, phase inductance, and magnetic flux of the motor. Keep the motor unloaded and securely fixed before proceeding with parameter calibration.

Click the "Parameter Settings" tab, then select "Parameter Calibration" to initiate the drive recognition process. The motor will first vibrate at low frequency, then accelerate to a specific speed before rotating at a constant speed, and finally perform forward and reverse rotation. This is the standard recognition sequence. If any abnormalities occur, check whether the motor has been calibrated and verify the normal rotation of the output shaft.



After recognition, the results will be automatically uploaded:

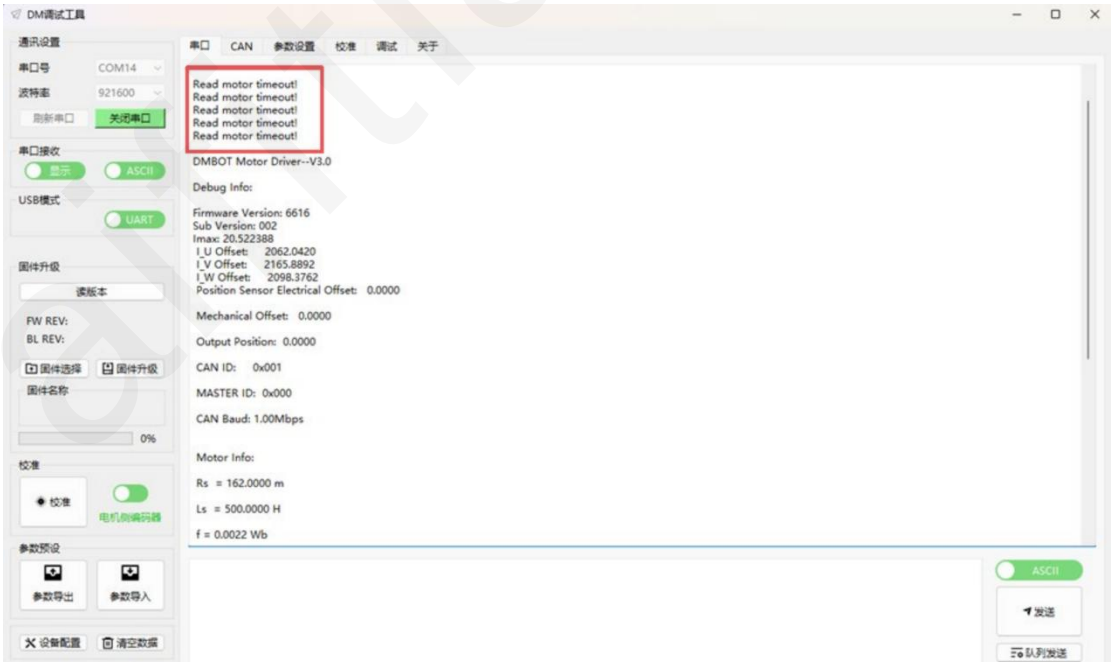
[Note] All parameters except the viscosity coefficient must be positive. If negative values occur, verify the motor status before recalibration.

Debugging Steps

Under normal conditions, the motor requires no calibration or adjustment. You can proceed with debugging as follows.

1. Connect the motor and driver

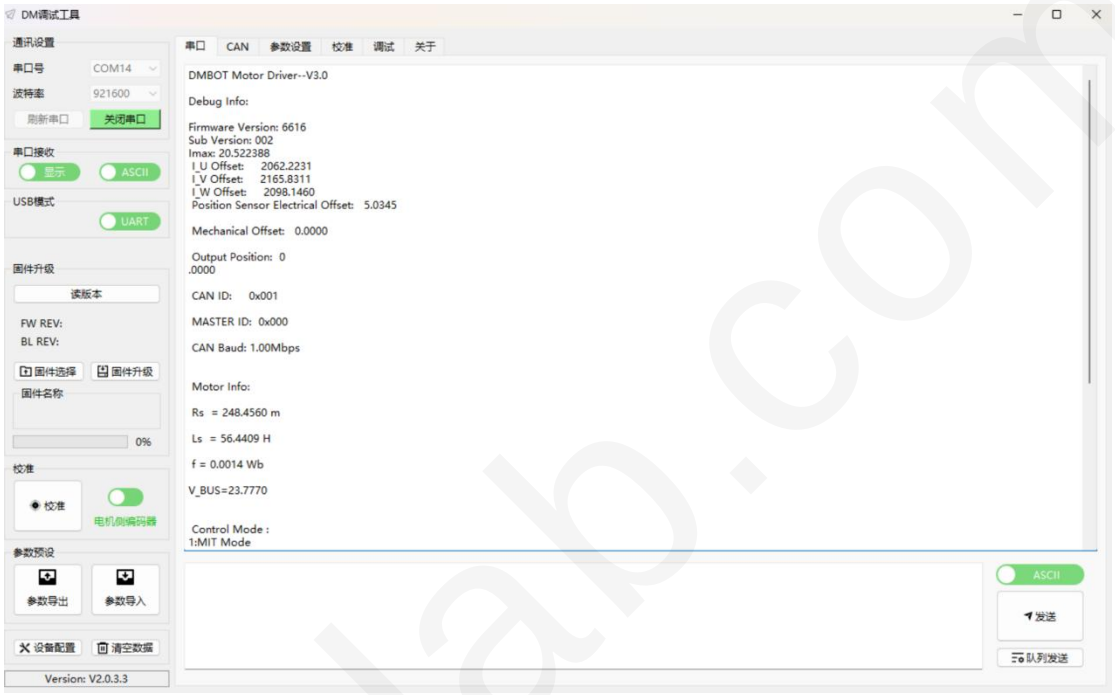
The motor and driver require a 4-pin sensor cable and a 3-phase power cable. Loose or disconnected communication cables may impair motor operation, causing the driver to display 'read timeout', error messages, and a red light indicator.



2. Connect the driver communication cable and power cable

The driver supports parameter read/write via a 3-pin serial port, while the 2-pin CAN line controls motor operation. Alternatively, CAN lines can be used for parameter exchange. For detailed specifications, see the CAN configuration commands section below.

Connect the driver and USB-to-CAN module using a serial cable and CAN cable, ensuring a secure connection without looseness. Verify the CAN cable sequence to confirm proper connections between CAN_H on the USB-to-CAN module and CAN_H on the driver, as well as CAN_L and CAN_L. After confirming the connections are correct, power the driver. Under normal conditions, the following message will be printed on the serial interface:



Includes firmware version, three-phase current offset, motor angle, driver ID configuration, baud rate, and power supply voltage.

3. Serial port read/write parameters

Switch to the parameter settings interface, click the "Read" button, and the driver will upload the current driver configuration parameters and motor parameters.

Check if the parameters meet the requirements. If needed, enter the data in the corresponding fields and click the Write Parameters button.

pay attention to：

① Do not modify the parameters of the logarithmic pole and the speed reduction ratio.

②Click 'Write Parameters' to automatically restart the driver without external power.

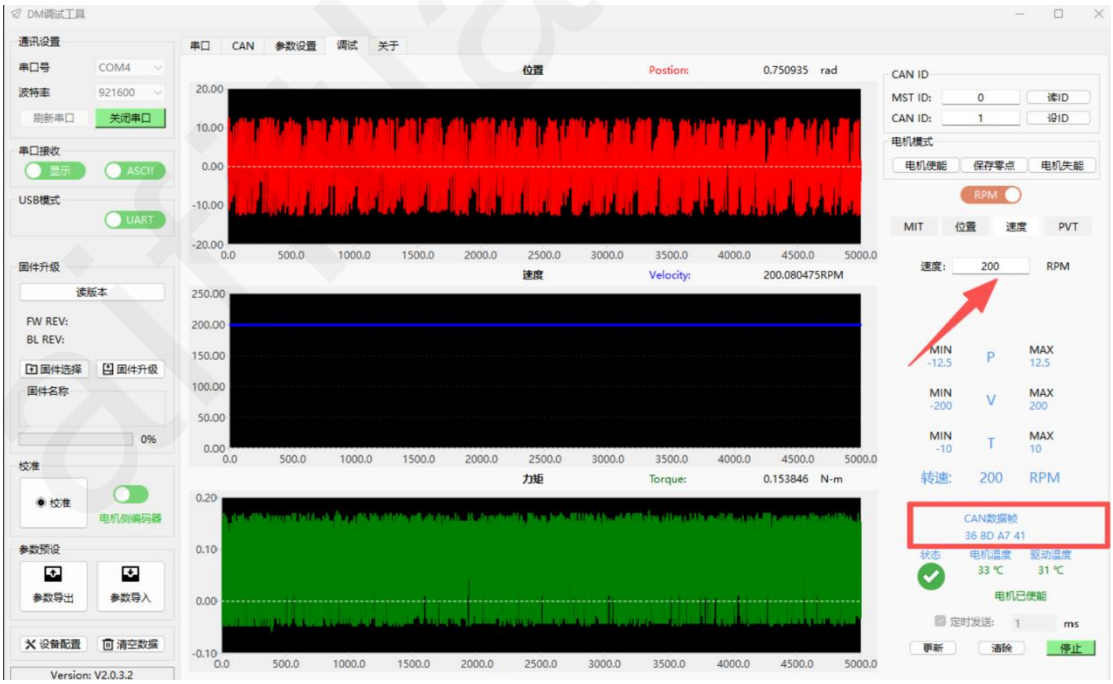
4. Debugging

After completing the initial steps correctly, you can proceed with the motor operation debugging. The debugging interface uses CAN communication. Ensure the CAN cable is properly connected before proceeding. Once enabled, the motor will rotate. This step requires securing the motor to prevent accidents.

Click the debug label to open the debug interface.

Click Send. Once communication is established, the interface will display the motor status curve.

Select the appropriate control mode tab. For instance, in speed mode, enter 20 rad/s and click update. The CAN data frame will automatically refresh the data, which is then sent to the driver. After enabling the function, the motor will rotate at 20 rad/s, with the waveform scrolling on the interface. This completes the debugging process.



CAN Configuration Command

Motor parameters can be configured via both serial port (via host computer) and CAN bus. In addition to basic parameter settings, the system now supports mode switching and baud rate adjustment.

1. Read parameters

message ID	attribute	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
0x7FF	STD	CANID_L	CANID_H	0x33	RID	xx(don’t care)			

RID is the register address, as defined in Section 6.

After successful read, the data of the register is returned, with the frame format as follows:

message ID	attribute	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 either floating-point or unsigned integer, occupying 32 bits (4 bytes), with D4 as the least significant bit and D7 as the most significant bit, and so on.

2. Write parameters

message ID	attribute	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			

As described above, after data is written, the driver performs a data range check. If the data falls within the specified range, the write operation succeeds. The driver then immediately loads the corresponding parameters and returns the written data, with the frame format identical to the original sent data.

message ID	attribute	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			

If the range is exceeded, the write operation fails, and the driver returns the original register data.

Writing register data takes effect immediately but cannot be stored. The data is lost after power failure. A command to send storage parameters is required to write all modified parameters into the chip.

3. Storage Parameters

message ID	attribute	D[0]	D[1]	D[2]	D[3]
0x7FF	STD	CANID_L	CANID_H	0xAA	XX

After successful write, the return format is:

message ID	attribute	D[0]	D[1]	D[2]	D[3]
MST_ID	STD	CANID_L	CANID_H	0xAA	01

pay attention to：

The storage parameters are only effective in the disabled mode.

② All parameters will be stored in one go.

③ This operation writes parameters to the on-chip flash memory, with each operation taking up to 30ms. Ensure you allow sufficient time.

④ The flash memory can withstand approximately 10,000 erase cycles. Avoid frequent transmission of the "storage parameters" command.

4. Mode Switch

Supports switching between multiple modes. The available control modes are:

Encoding	Mode
1	MIT
2	Velocity of position
3	Velocity

By adjusting the mode register (0x0A), the operating mode can be modified. During mode switching, the motor first resets all command values, including position, speed, and the torque feedforward and KP/KD values in MIT mode.

When switching from a mode to position control mode, to prevent shock, it is recommended to first read the precise position before considering the switch, and to perform the switch when the motor is at zero speed.

After modifying the mode, it won't be saved to the flash memory and will be lost if the power is cut. Upon reboot, the control mode will revert to the last saved configuration in the flash.

5. Port rate modification

The current CAN communication baud rate can be modified by writing specific data to the baud rate register (address 0x23). The driver supports specific baud rate modifications, with the following supported rates:

Encoding	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 modifying the baud rate, the driver first transmits data at the original baud rate before switching to the new rate. Upon power-on, the motor first checks the stored baud rate. If it exceeds 5Mbps, the system automatically defaults to 1Mbps. For rates above 1Mbps (excluding 1Mbps), the motor switches to CAN FD mode. If the baud rate is ≤1Mbps, it automatically switches to CAN 2.0B. Motors configured as CAN FD can still receive CAN 2.0B data frames but use CAN FD for feedback transmission, causing the upper-layer controller to miss feedback data and triggering continuous driver error reports. Controllers using CAN 2.0B can revert to the original baud rate by issuing a baud rate modification command after missetting the ID.

6. Register address

Address （HEX）	Address (DEC)	Variable	Description	Read-write	Scope	Type
0x00	0	UV_Value	Low voltage 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	Overcurrent protection value	RW	(0.0,1.0)	float
0x04	4	ACC	Accelerated speed	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

0x08	8	ESC_ID	Receive ID	RW	[0,0x7FF]	uint32
0x09	9	TIMEOUT	Timeout alert time	RW	[0,2^32-1]	uint32
0x0A	10	CTRL_MODE	Control model	RW	[0,4]	uint32
0x0B	11	Damp	Motor viscosity coefficient	RO	/	float
0x0C	12	Inertia	Rotating inertia of motor	RO	/	float
0x0D	13	hw_ver	Continue to have	RO	/	uint32
0x0E	14	sw_ver	Software version	RO	/	uint32
0x0F	15	SN	Continue to have	RO	/	uint32
0x10	16	NPP	Number of pole pairs of motor	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	Velocity 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 ring Kp	RW	[0.0,fmax]	float
0x1A	26	KI_ASR	Speed ring 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	Overvoltage protection value	RW	TBD	float
0x1E	30	GREF	Gear torque efficiency	RW	(0.0,1.0]	float
0x1F	31	Deta	Speed loop damping coefficient	RW	[1.0,30.0]	float
0x20	32	V_BW	Speed loop filter bandwidth	RW	(0.0,500.0)	float

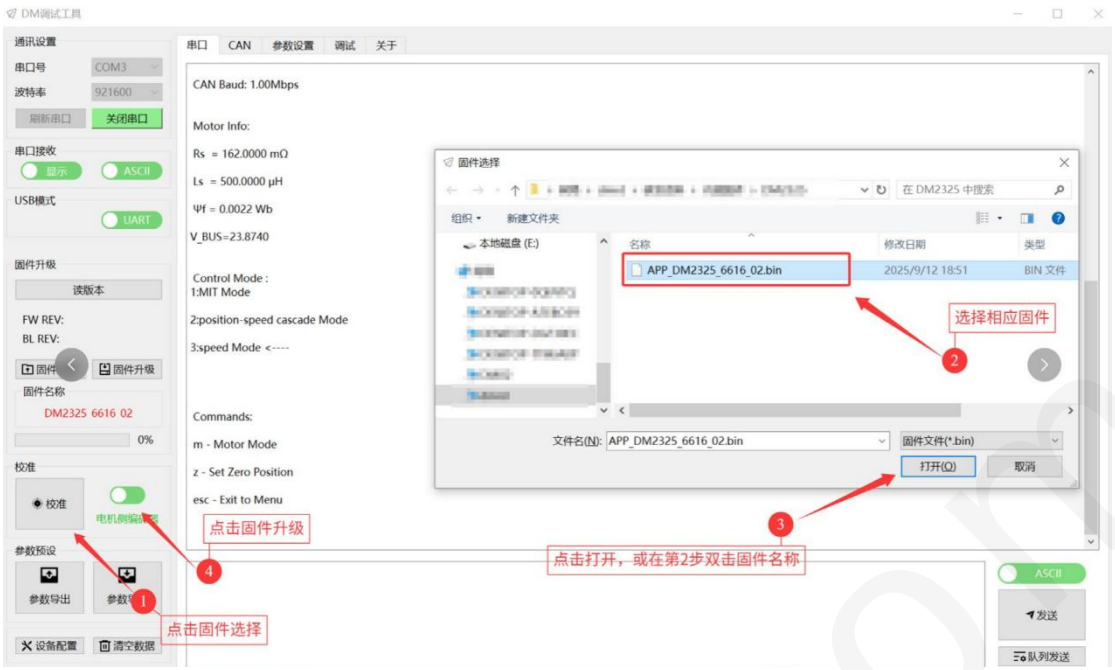
0x21	33	IQ_c1	Current loop enhancement factor	RW	[100.0, 1.0e4]	float
0x22	34	VL_c1	Speed loop enhancement coefficient	RW	(0.0,1.0e4]	float
0x23	35	can_br	CAN baud rate code	RW	[0,4]	uint32
0x24	36	sub_ver	Subversion number	RO	/	uint32
0x32	50	u_off	U phase offset	RO	/	float
0x33	51	v_off	V phase offset	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

RW: Read and write.

RO： read only.

Firmware Upgrade

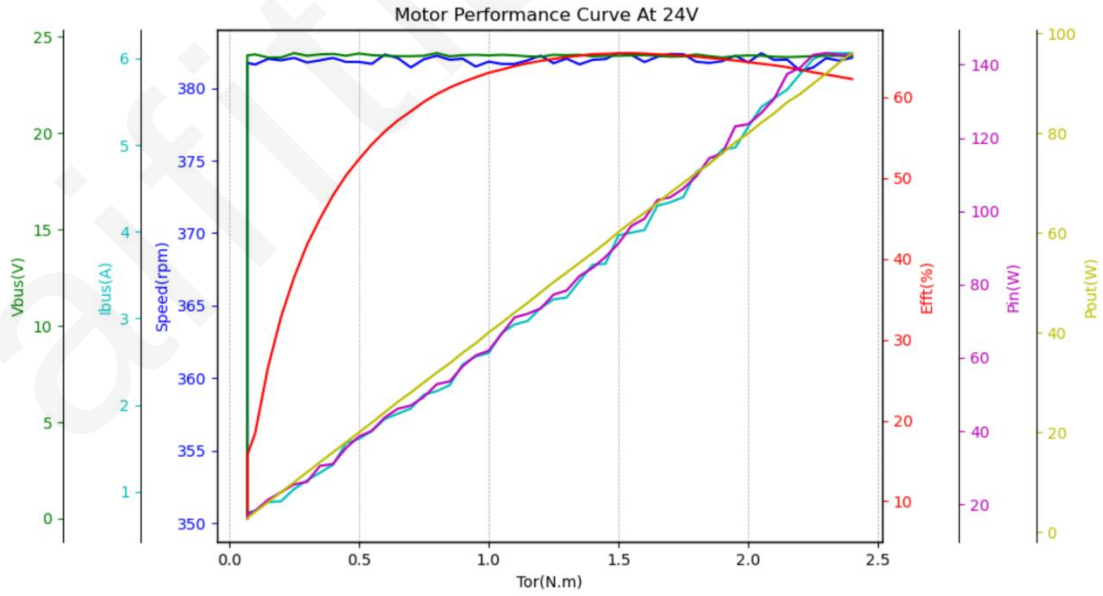
When new firmware features are added or bug fixes are released, users can perform upgrades via the serial port to resolve issues and access new functionalities. Before using the upgrade, connect the serial port, select the desired firmware version through the "Firmware Selection" menu, confirm the choice, and click the firmware upgrade button. Wait for the progress bar to complete, or check the upgrade status through the serial port interface.



Note: The firmware name must match the motor to avoid selection errors.

Characteristic Curve

Test environment: room temperature 25℃, laboratory environment.



Characteristic Parameter

Use the motor properly according to the following parameters.

Motor Parameters	Rated voltage	24V (Supports 24~48V power supply)
	Rated phase/power current	5A/3.2A
	Peak phase/power current	13A/7.5A
	Rating torque	1.35Nm
	Peak torque	5Nm
	Rated speed	380rpm
	No-load maximum speed	560rpm
Motor Characteristics	Reduction gear ratio	1：25
	Number of pole-pairs	7
	Phase inductance	75uH
	Phase resistance	0.2915Ω
	Maximum radial load (dynamic load)	395N
Structure and Weight	External diameter	Motor 28mm
	Altitude	Motor 77.6mm
	Motor weight	Approximately 172.2 grams
Encoder	Encoder type	incremental encoder
Communication	Control interface	CAN@1Mbps（Max）、FD CAN@5Mbps（Max）
	Parameter passing interface	UART@921600bps
Control and protection	Control model	MIT pattern
		Speed mode
		position mode
	Protect	Enable over-temperature protection with a protection temperature of 120°C. The motor will exit the enable mode if overheated.
		Motor over-temperature protection. Set according to usage requirements, recommended not to exceed 100°C. The motor will exit the 'enabled mode' if overheated.
		Motor overvoltage protection. Set according to usage requirements, recommended not to exceed 60V. Overvoltage will exit the "enabled mode".

		Communication loss protection. If no CAN commands are received within the set period, the system will automatically exit the enable mode.
		Motor overcurrent protection. Set according to usage requirements. It is recommended not to exceed 13A (phase current). Overcurrent will exit the 'enabled mode'.
		Motor under-voltage protection: If the power supply voltage drops below the set threshold, the system exits the enable mode. The minimum required voltage is 15V.

Download link 1: https://gl1po2nscb.feishu.cn/drive/folder/RJL7fFT4ll9PDSdvM6Pc5vntnPw

Download link 2: https://gitee.com/kit-miao/damiao/tree/master

User Help Manual 3: https://gl1po2nscb.feishu.cn/wiki/MZ32w0qnnizTpOkNvAZcJ9SlnXb