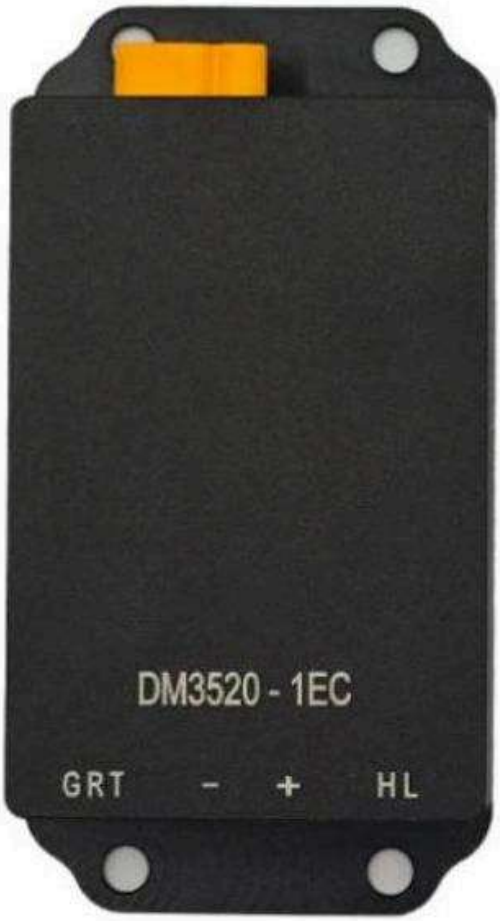
DAMIAO | Damiao Technology

# DM-S3519-1EC Geared Motor (including DM3520-1EC Driver)

User Manual V1.1 2024.11.18


| Date | Version | Changes |
|---|---|---|
| 2024.11 .06 | V 1.0 | First Edition Created |
| 2024.11 .18 | V 1.1 | Added Characteristic Curve |

## Disclaimer

Thank you for purchasing the DAMIAO DM-S3519-1EC motor (hereinafter referred to as "motor") / DM3520-1EC driver (hereinafter referred to as "driver") from Damiao Technology. Before using this product, please read and follow this document and all safety guidelines provided by Damiao Technology carefully. Failure to do so may cause harm to you and those around you, and damage this product or other surrounding items. By using this product, you are deemed to have carefully read this document and understood, acknowledged, and accepted all terms and contents of this document and all related documents for this product. You promise to use this product only for legitimate purposes. You promise to take full responsibility for the use of this product and any potential consequences. Damiao Technology is not responsible for any damage, injury, or legal liability caused by the direct or indirect use of this product.

## Precautions

1. Please strictly use the driver within the specified operating environment and the maximum allowable winding temperature range; otherwise, it will cause permanent and irreversible damage to the product.

2. Avoid conductive debris entering the driver, as this may cause abnormal operation of the driver.
3. Before use, please check if all components are intact. If any components are missing, aged, or damaged, please stop using the product.
4. Ensure correct and secure wiring and connection to the motor.
5. Do not touch the driver during use to avoid accidents. When the driver outputs high current, it may heat up. Please be careful to avoid burns.

6. Users should not disassemble the driver without authorization to avoid damaging it.

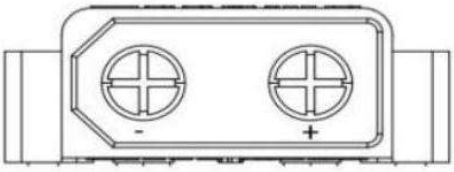
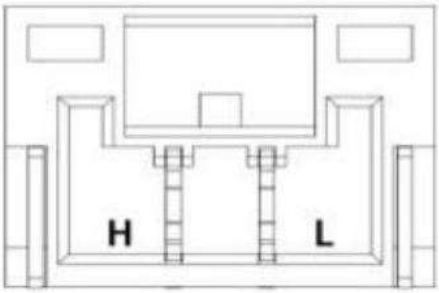
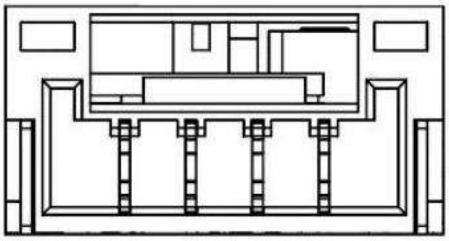
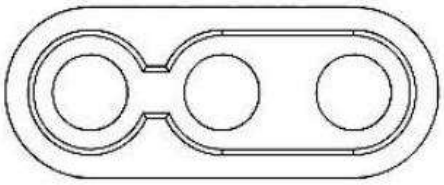
## Product Features

1. The driver features an intelligent design, allowing for plug-and-play functionality when replacing motors, without the need for recalibration or parameter adjustment.
2. Supports firmware upgrades.
3. Visual parameter adjustment via PC software, enabling use with simple configuration.
4. Supports CAN FD functionality, with a maximum baud rate of up to 5 Mbps.
5. Motor speed, position, torque, motor temperature, and other information can be fed back via the CAN bus.
6. Features dual temperature protection.
7. Low speed, high torque.
8. Flexible switching between multiple control modes.

## Item List

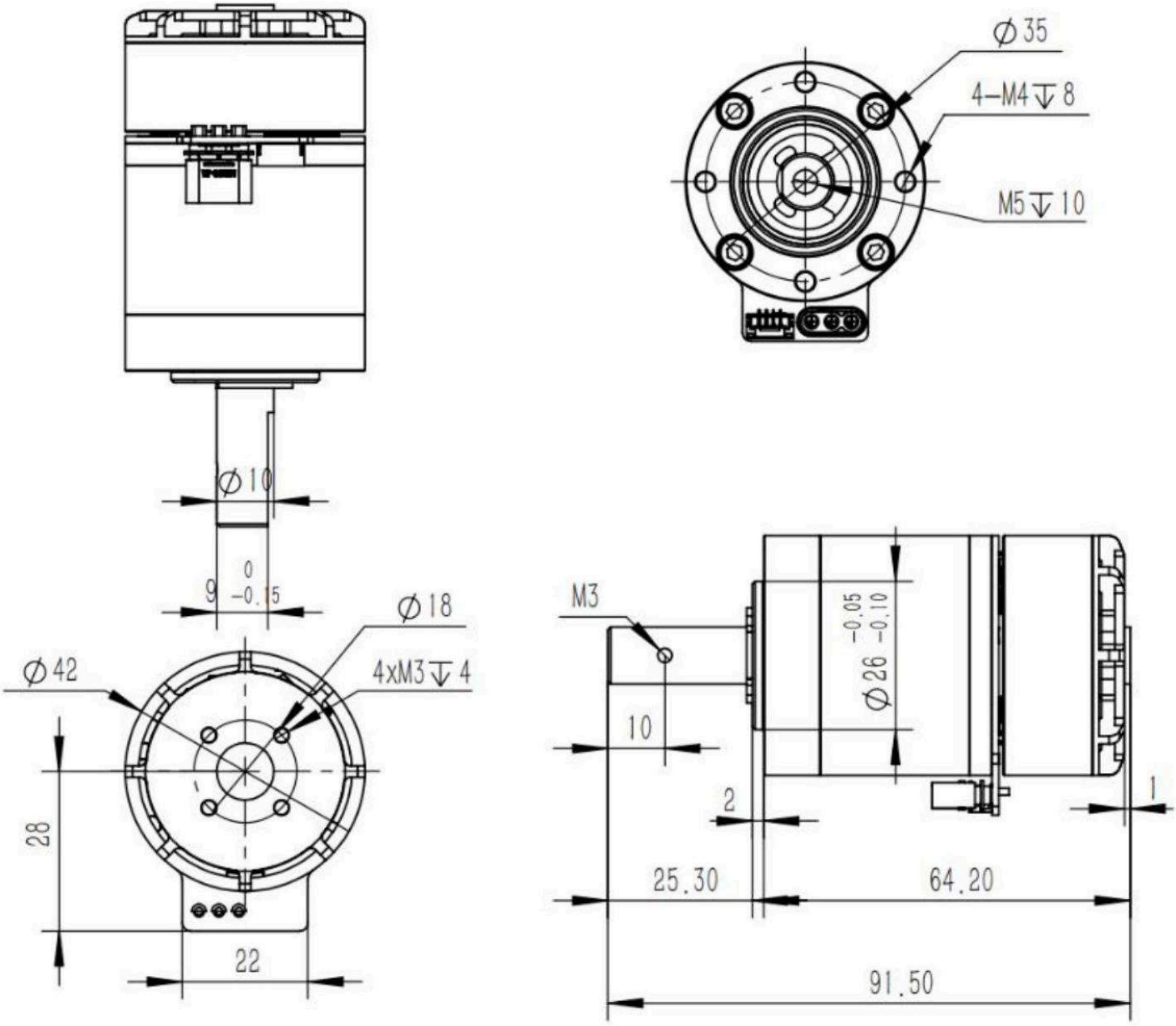
| Category | Checklist | Legend |
| --- | --- | --- |
| Motor Set | 1. Driver ×1 2. Geared Motor ×1 3. XT30 Power Cable (dual-ended, 200 mm) ×1 4. 3 Pin Serial Cable (non-planar, 300 mm) ×1 5. 2 Pin CAN Cable (non-planar, 300 mm) ×1 6. 4 Pin Encoder Cable (non-planar, 250 mm) ×1 7. 3 Pin Motor Phase Cable (non-planar, 200 mm) ×1 |  |
| Single Motor | 1. Geared Motor ×1 |  |
| Single Driver | 1. Driver ×1 2. XT30 Power Cable (double-ended, 200 mm) ×1 3. 3-Pin Serial Cable (different sides, 300 mm) ×1 4. 2-Pin CAN Cable (different sides, 300 mm) ×1 5. 4-Pin Encoder Cable (different sides, 250 mm) ×1 6. 3-Pin Motor Phase Cable (different sides, 200 mm) ×1 |  |

## Interface and Wiring Description

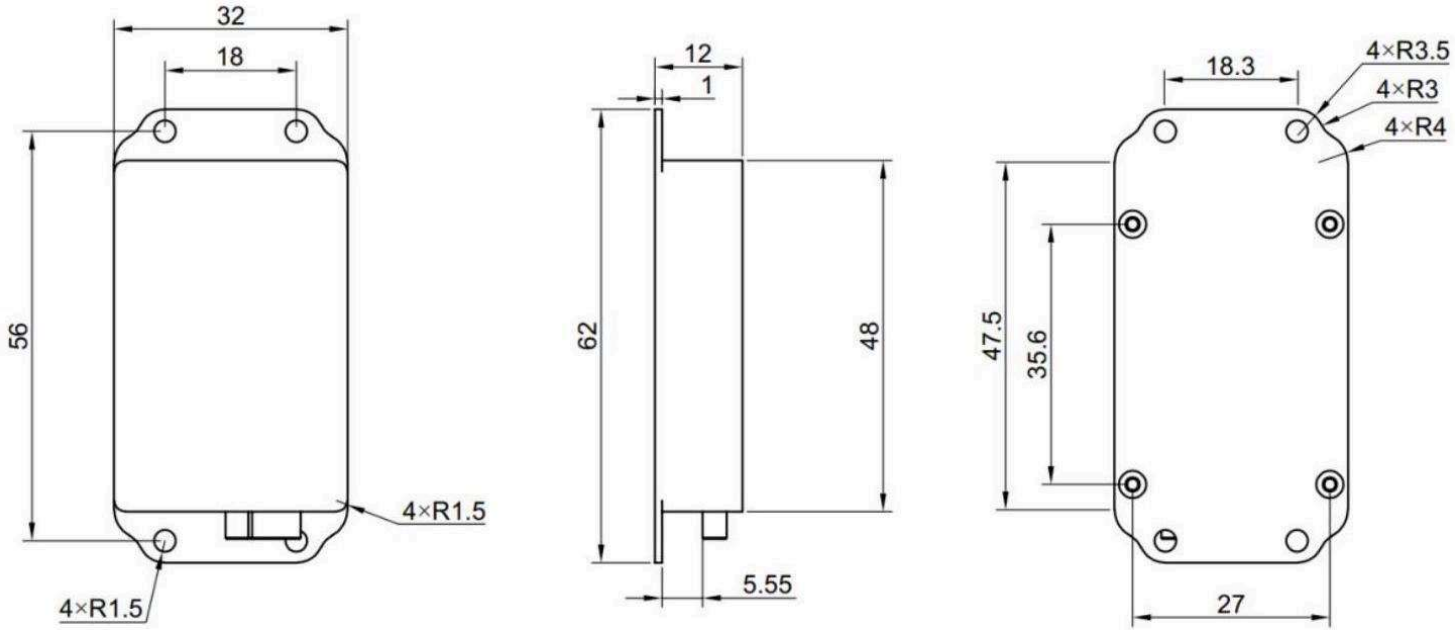
| Specific Name - Number | Interface Markings | Description |
| --- | --- | --- |
| Power Connection Interface |  | Connect the power supply via the XT30-F plug power cable. The rated voltage is 24 V, which powers the motor. |
| Serial Port | GND RX TX | Used for serial port debugging, parameter adjustment, and other functions |
| CAN Interface |  | Used to control the driver |
| Encoder Interface |  | Used to connect the DM-S3519-1EC motor, using the accompanying 4-pin twisted pair cable. |
| Three-Phase Wire Interface |  | Used for connecting the motor's three-phase wires with the accompanying cable |

## Geared Motor & Driver Dimensions and Installation

Please refer to the driver mounting hole dimensions and positions to install the driver onto the corresponding equipment.




DM-S3519-2EC Geared Motor Dimension Drawing

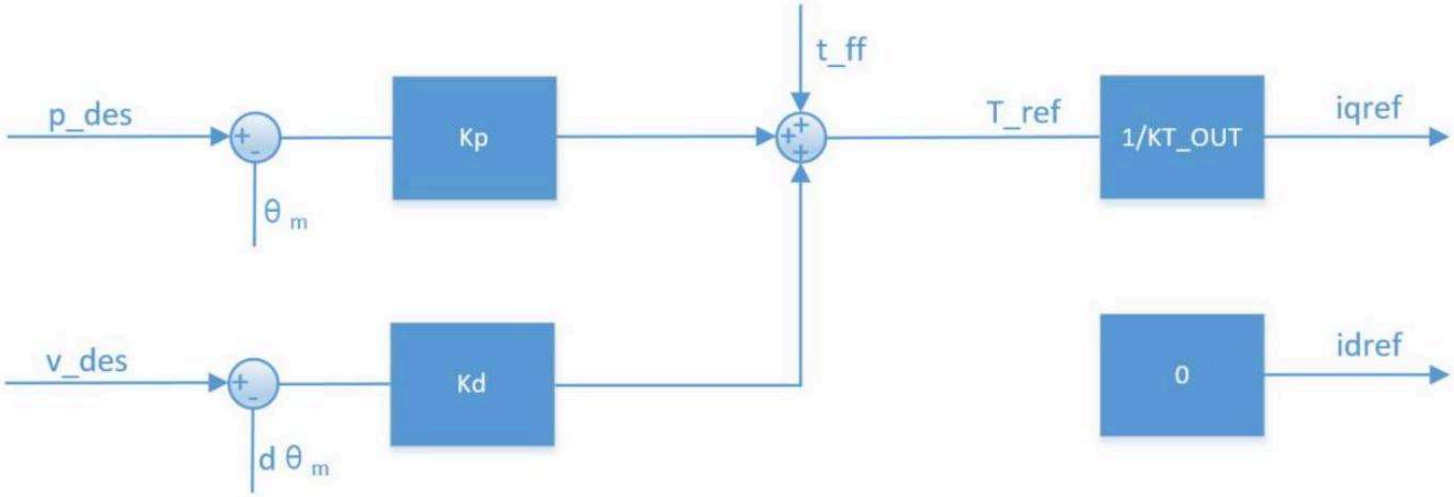



DM3520-1EC Driver Dimension Drawing

Operating Modes

*MIT Mode

MIT mode is designed to be compatible with the original MIT mode, allowing for seamless switching while flexibly setting the control range (P_MAX, V_MAX, T_MAX). The driver converts the received CAN data into control variables for calculation to obtain a torque value, which is then used as the current setpoint for the current loop. The current loop ultimately reaches the given torque current according to its regulation rules. The control block diagram is as follows:

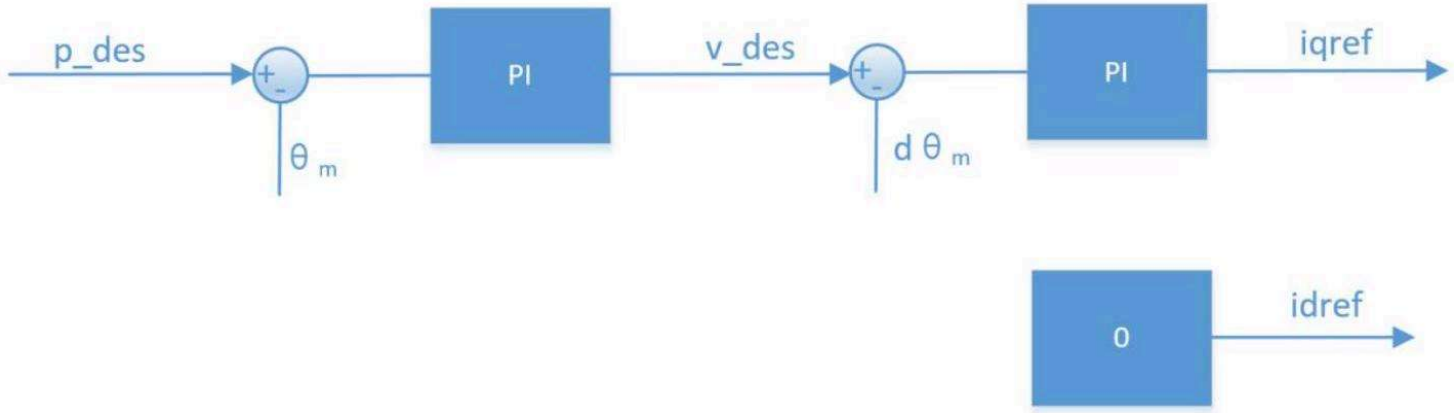



Based on MIT mode, various control modes can be derived. For example, when $kp = 0, kd$ is not 0, providing v_ des can achieve constant speed rotation; $kp = 0, kd = 0$ , providing t_ff can achieve a given torque output.

Note: 1. After power-on, the motor position is fixed at 0.0 rad.
2. When controlling the position, *kd* cannot be set to 0, otherwise it will cause motor oscillation or even loss of control.

Position Velocity Mode

Position cascade mode adopts a three-loop series control. The position loop acts as the outermost loop, its output serving as the input for the velocity loop. The output of the velocity loop then serves as the input for the inner current loop, which controls the actual current output. The control schematic diagram is shown below:

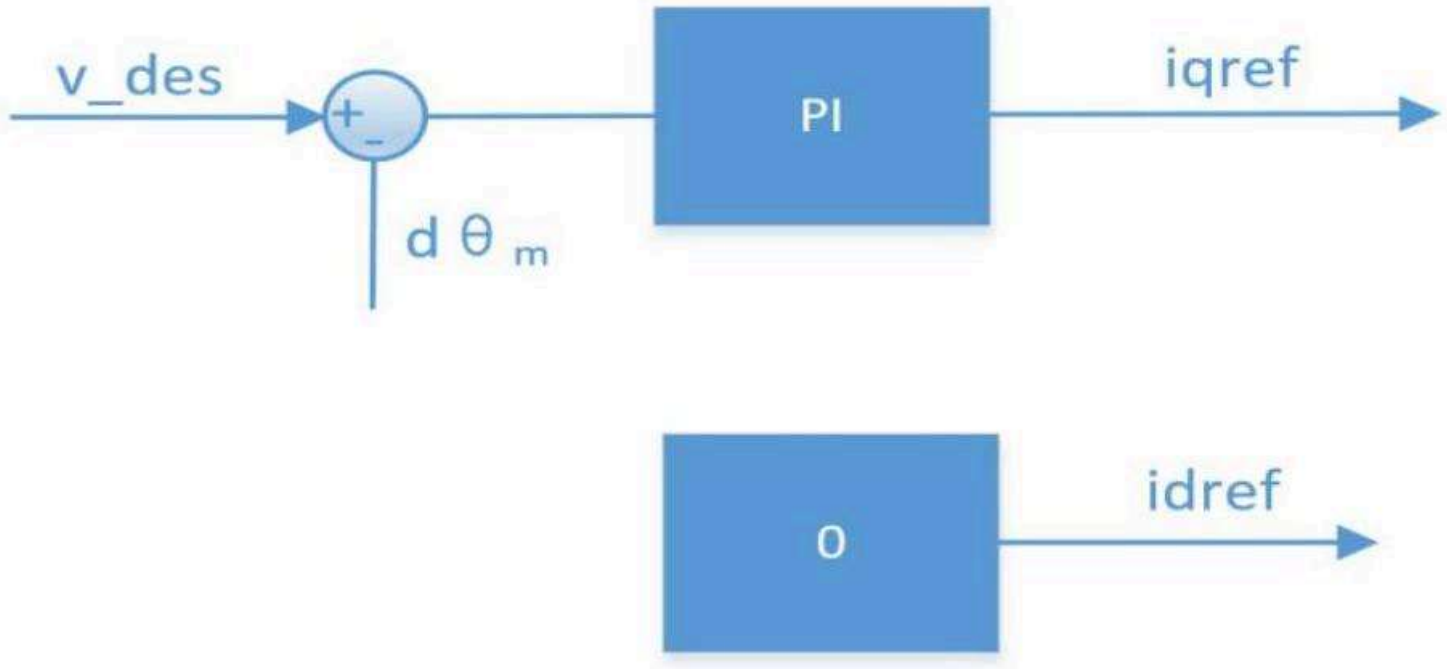



p_ des is the target position for control, and v_ des is used to limit the maximum absolute velocity value during motion. If the position cascade mode uses the control parameters recommended by the debugging assistant, it can achieve better control accuracy and a relatively smooth control process, but the response time is relatively long. In addition to v_des, other configurable parameters include acceleration/deceleration.

fixed. If additional oscillation occurs during the control process, the acceleration/deceleration can be increased.
Note:
1. After power-on, the motor position is fixed at 0.0 rad.
2. $p_-$ des and $v_-$ des units are rad and $\mathrm{rad/s}$ respectively, and the data type is float.

Speed Mode

The speed mode allows the motor to operate stably at the set speed. The control block diagram is as follows:




Note:

1. The unit of v_des is $\mathrm{rad/s}$ , and the data type is float.

Control Protocol Description

Control uses the CAN standard frame format, with a default baud rate of 1 Mbps. The baud rate can be changed to a different one using commands; refer to the section on baud rate modification for details. By function, it can be divided into reception frames and feedback frames. Reception frames are received control data used to implement command control of the motor; feedback frames are motor status data sent by the motor to the upper-level controller. The feedback behavior is inquiry-based: as long as the frame ID received by the driver matches the CAN ID set for the motor (the lower 8 bits are checked, the upper 3 bits are ignored), the driver sends the current status data to the bus. Depending on the different modes selected for the motor, the definition of the reception frame format and the frame ID vary, but the feedback frame format and data are the same across all modes.

Feedback Frame

The feedback frame ID is set by the debugging assistant (Master ID), defaulting to 0. It primarily provides feedback on the motor's position, speed, and torque information. Its 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 |

Where:

ID represents the controller's ID, taking the lower 4 bits of CAN_ID

ERR represents the status, corresponding to the status type:

0 - Disabled;
1 - Enabled;
5 - Read sensor error;
6 - Read motor parameter error;
8 - Overpressure;
9 - Undervoltage;
A - Overcurrent;
B - MOS over-temperature;
C - Motor coil over-temperature;
D - Communication loss;
E - Overload;

POS indicates motor position information*.

VEL indicates the motor's speed information*.

T indicates the motor's torque information*.

T_MOS indicates the average temperature of the upper MOS of the driver, unit °C

T_Rotor indicates the average temperature of the motor's internal coils, unit °C

Position, velocity, and torque use a linear mapping to convert floating-point data into signed fixed-point data, with position using 16-bit data, and both velocity and torque using 12-bit data.

*Note:

1. After power-on, the motor position is fixed at 0.0 rad.
2. The unit of position is rad (radians), and it represents the position of the output shaft, i.e., the position after reduction. All subsequent descriptions of position adhere to this definition and will not be reiterated.

3. The unit of speed is $\mathrm{rad/s}$ (radians per second), and it represents the speed of the output shaft, i.e., the speed after reduction. All subsequent descriptions of speed adhere to this definition and will not be reiterated.

4. The unit of torque is Nm, and it represents the torque of the output shaft, i.e., the torque after reduction. The following descriptions of torque all adhere to this definition and will not be reiterated.

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] |

Frame ID equals the set CAN ID value

P_des: Position command

V_des: Velocity command

Kp: Position proportional coefficient

Kd: Position differential coefficient

T_ff: Torque Setpoint

Each parameter conforms to the mapping relationship in the previous section, where the ranges of P_ des, v_ des, and t_ff can be set by the debugging assistant. The range of Kp is $[0, 500]$, Kd , and the range of $[0, 5]$ is $[0, 5]$.

A standard CAN data frame only has 8 bytes. The MIT control command format combines five parameters—Position, Velocity, Kp、Kd、 Torque—bit-wise into 8 bytes. Specifically: Position occupies 2 bytes (16 bits), Velocity occupies 12 bits, Kp occupies 12 bits, and Kd occupies 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] |
|---|---|---|---|---|---|---|---|---|
| $0 \times 100 + \mathrm{ID}$ | p_des | | | | v_des | | | |

The frame ID is the set CAN ID value plus an offset of $0 \times 100$

P_des: Position command, float type, low byte first, high byte last.

V_des: Velocity command, float type, low byte first, high byte last.

The CAN ID for sending commands here is $0x100+$ ID. The speed setting is the maximum speed limited during motor operation.

For example:

To set the motor with CAN ID 5 to reach a position of 180 degrees (3.14159 rad) at a maximum speed not exceeding $10\mathrm{rad/s}$ , the command is as follows:

| 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] |
|---|---|---|---|---|
| $0 \times 200 + \mathrm{ID}$ | v_des | | | |

The frame ID is the set CAN ID value plus an offset of $0 \times 200$

V_des: Velocity command, float type, low byte first, high byte last.

The CAN ID for sending commands here is $Ox200 + ID$ .

For example:

To set the motor with CAN ID 3 to rotate at a speed of $60\mathrm{rpm}\ (6.283\mathrm{rad/s})$ , the command is as follows:

| Frame ID | D[0] | D[1] | D[2] | D[3] |
|---|---|---|---|---|
| 0x203 | 0x56 | 0x0E | 0xC9 | 0x40 |

Instructions

The motor has been calibrated and set at the factory and the settings are stored in the motor. Unless there is an abnormal situation, it can be used directly without further adjustment

New calibration and adjustment. The following provides the calibration and adjustment process, which can normally be skipped.

※ Motor Calibration

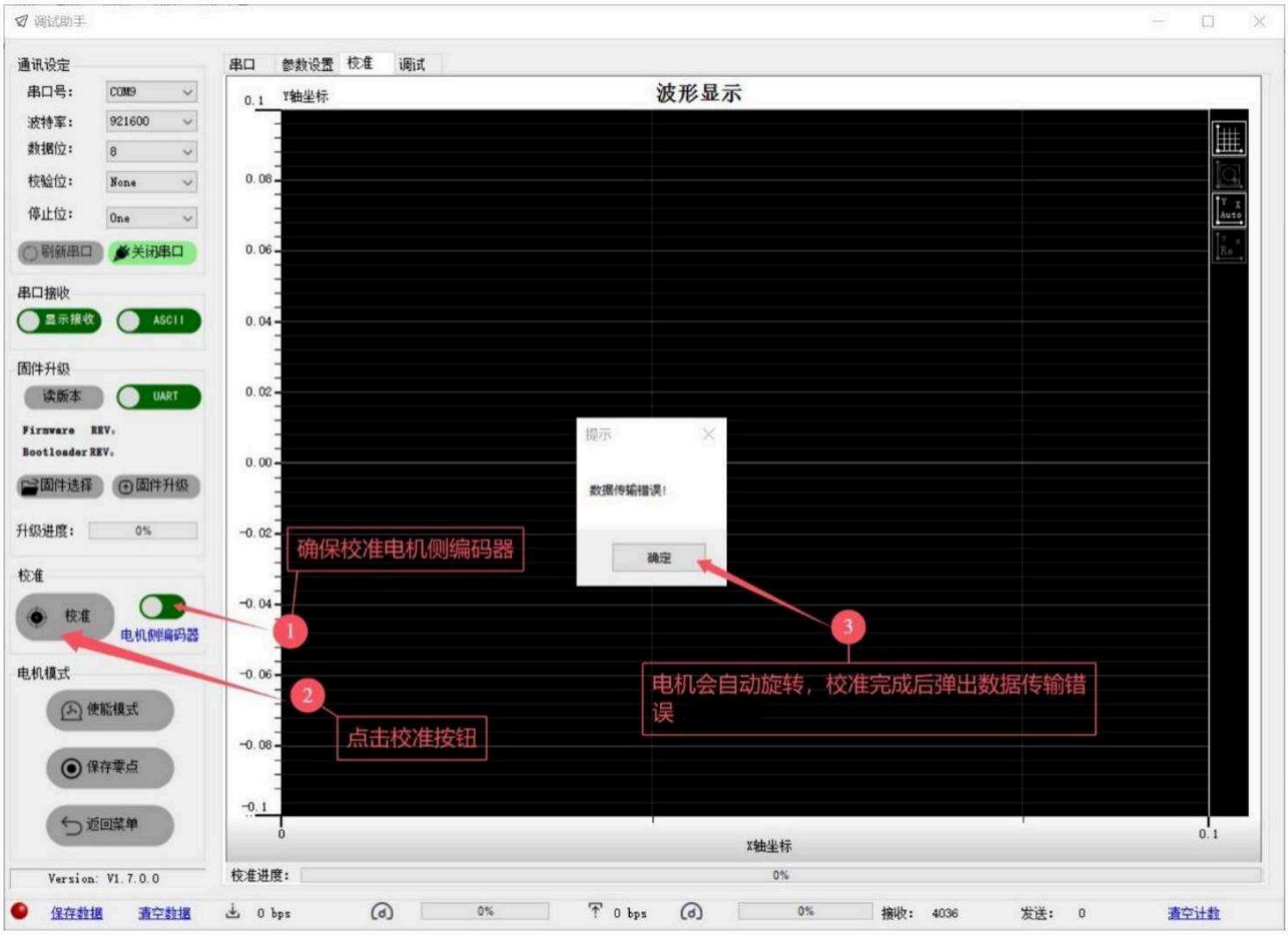
The main purpose of calibration is to correct sensor installation errors and sensor orientation. During the process, the motor will rotate in a certain direction. Please ensure that the motor can rotate freely, preferably under no-load conditions, otherwise calibration may fail.

You can use the Damiao Debugging Assistant to calibrate the motor. Calibration uses serial communication, and any serial tool that supports a baud rate of 921600 can be used. Taking USB to CAN as an example for calibration instructions:

1. Use the three-phase wires and sensor wires to connect the motor and driver, ensuring a firm and secure connection without looseness.
2. Connect the driver and the USB to CAN adapter with a 3-pin serial cable, ensuring a secure and stable connection.
3. Connect the USB to CAN adapter to the PC and open the debugging assistant software.
4. Use the following parameters to configure and open the serial port.


5. Select "Motor Side Encoder" on the right.


6. Power the motor with 24 V (the calibration process current is about 1 A, please use a power supply with 1 A or more).
7. Click "Calibrate", and the motor will rotate freely. The entire process takes about 2 seconds.
8. After calibration is complete, the driver will send a message to the serial port, which will cause the host computer to prompt "Data transmission error". This can be ignored.




9. Return to the serial port interface, and you can see the driver printing the current calibration information to the serial port. At this point, the calibration is complete.

| | | | |
|---|---|---|---|
| - Debugging Assistant | | | - |
| Communication Settings | Serial Port Parameter Settings Calibration Debugging | | |
| Serial Port: COM. 9 COM. 9 | | | |
| | Mechanical offset: 0.0000 Output Position: 0.0000 CAN ID: 0x001 | | |
| | | | |
| | | | |
| Stop Bit: ☐ One | Motor Info: $\mathrm{Rs} = 162.0000\ \mathrm{m}\Omega$ $\mathrm{Ls} = 500.0000\mu\mathrm{H}\ \psi\mathrm{f} = 0.0022\psi$ b V_BUS = 23.7966 | | |
| Read Version | Control Mode: 1: MIT Mode | | |
| Upgrade Progress: 類類 0% | Commands: m - Motor Mode $z$ - Set Zero Position esc - Exit to Menu | | |
| | | | |
| Calibration Calibration Motor Encoder | Commands: m - Motor Mode $z$ - Set Zero Position | | |
| | 1. 1658, Order: 1.0 | | |
| | | | |
| Motor Mode Enable Mode Return to Menu | ☐ | | |
| | | |   |
| Version: V1.7.0.0 | | | |
| Data Preservation Qingkong Data | T 0 bps (d) | Sent: 0 | Sky Count |

## Parameter Calibration

After calibration, parameter identification can be performed to identify important parameters such as the motor's phase resistance, phase inductance, and magnetic flux linkage.
Please ensure the motor is unloaded and securely fixed before proceeding with parameter calibration.
Click the "Parameter Settings" tab, then click the "Parameter Calibration" button, and the drive will enter the identification step. During this period, the motor will first vibrate at a low frequency, then accelerate to a certain speed and rotate at a constant speed, and finally rotate forward and backward. The above are correct identification phenomena. If abnormal situations occur, please check whether the motor has been calibrated and whether the output shaft rotation is normal.


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

| Motor Parameters | Parameter Calibration |
|---|---|
| Phase Resistance (R): | 182.5193 mR |
| Phase Inductance (L): | 57.28119 uH |
| Magnetic Flux ( λ ): | 0.002268364 wb |
| Viscous friction coefficient: | 4.780932E − 05 |
| Moment of inertia: | 1.7697E − 05 kg ∗ m2 |

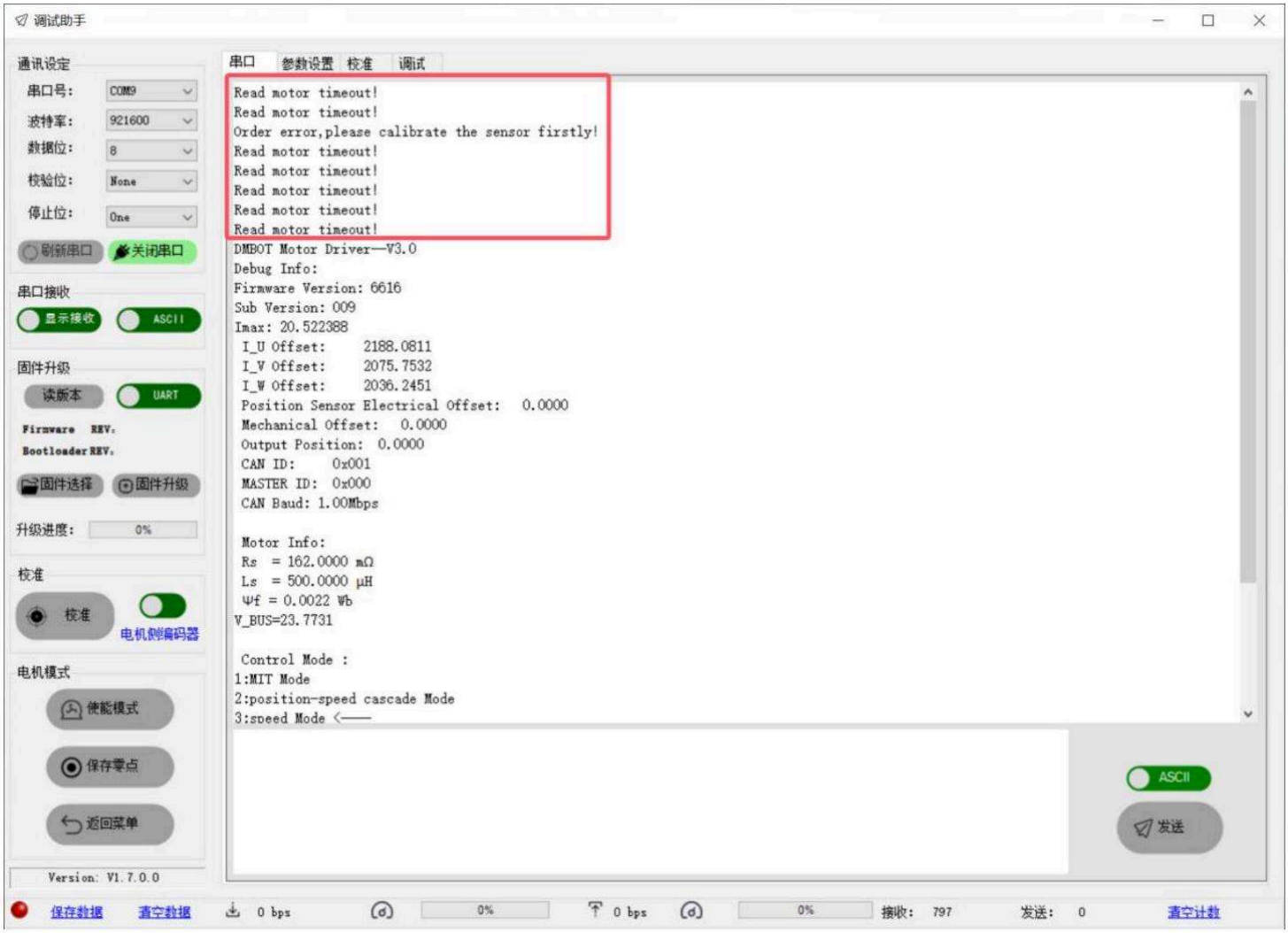
[Note] The above parameters, except for the viscous friction coefficient, cannot be negative. If negative values appear, please confirm the motor status before proceeding with calibration.

*Debugging Steps

Under normal circumstances, the motor does not require calibration or standardization operations. Debugging can be performed according to the following steps.

1. Connect the motor and driver

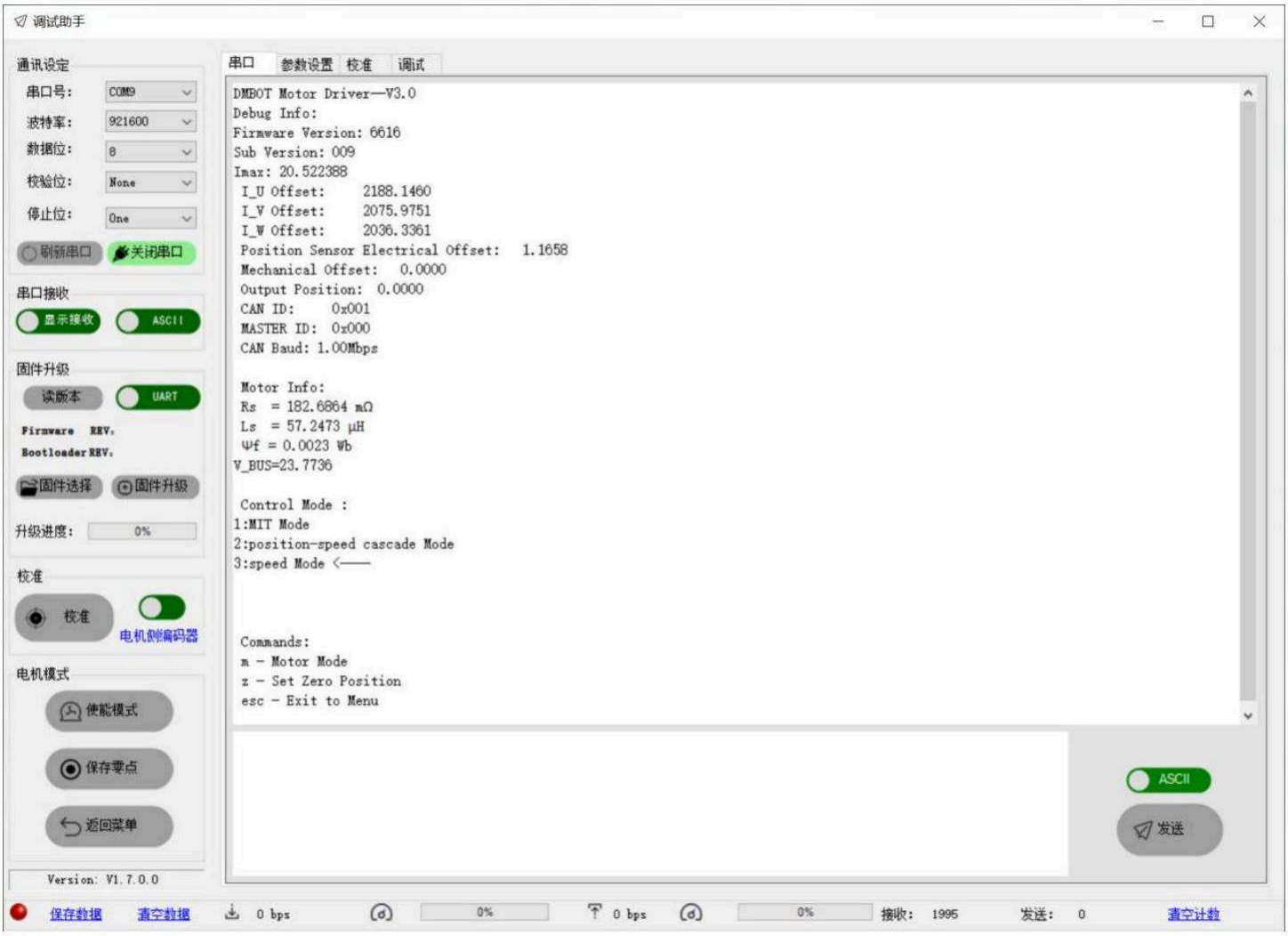
The motor and driver require a 4-pin sensor cable and a 3-phase power cable connection. If the communication cable connection is loose or not connected, it will affect the normal use of the motor. When the driver is powered on, it will print a read timeout, communication error, and the red light will flash.




2. Connect the drive communication cable and power cable

The drive can use a 3-pin serial cable for parameter reading and writing, and a 2-pin CAN cable to control motor operation. The CAN cable can also be used for parameter reading and writing, as detailed in the "CAN Configuration Commands" section below.

Use the serial cable and CAN cable to connect the drive and the USB to CAN module. Ensure the connection is secure and not loose. Pay attention to the CAN cable wiring sequence: ensure that CAN_H on the USB to CAN module is connected to CAN_H on the drive, and CAN_L is connected to CAN_L on the drive. After verifying the connections, power on the drive. Under normal circumstances, the following message will be printed on the serial interface:




This includes information such as firmware version, three-phase current offset, motor angle, drive 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 currently configured driver and motor parameters.

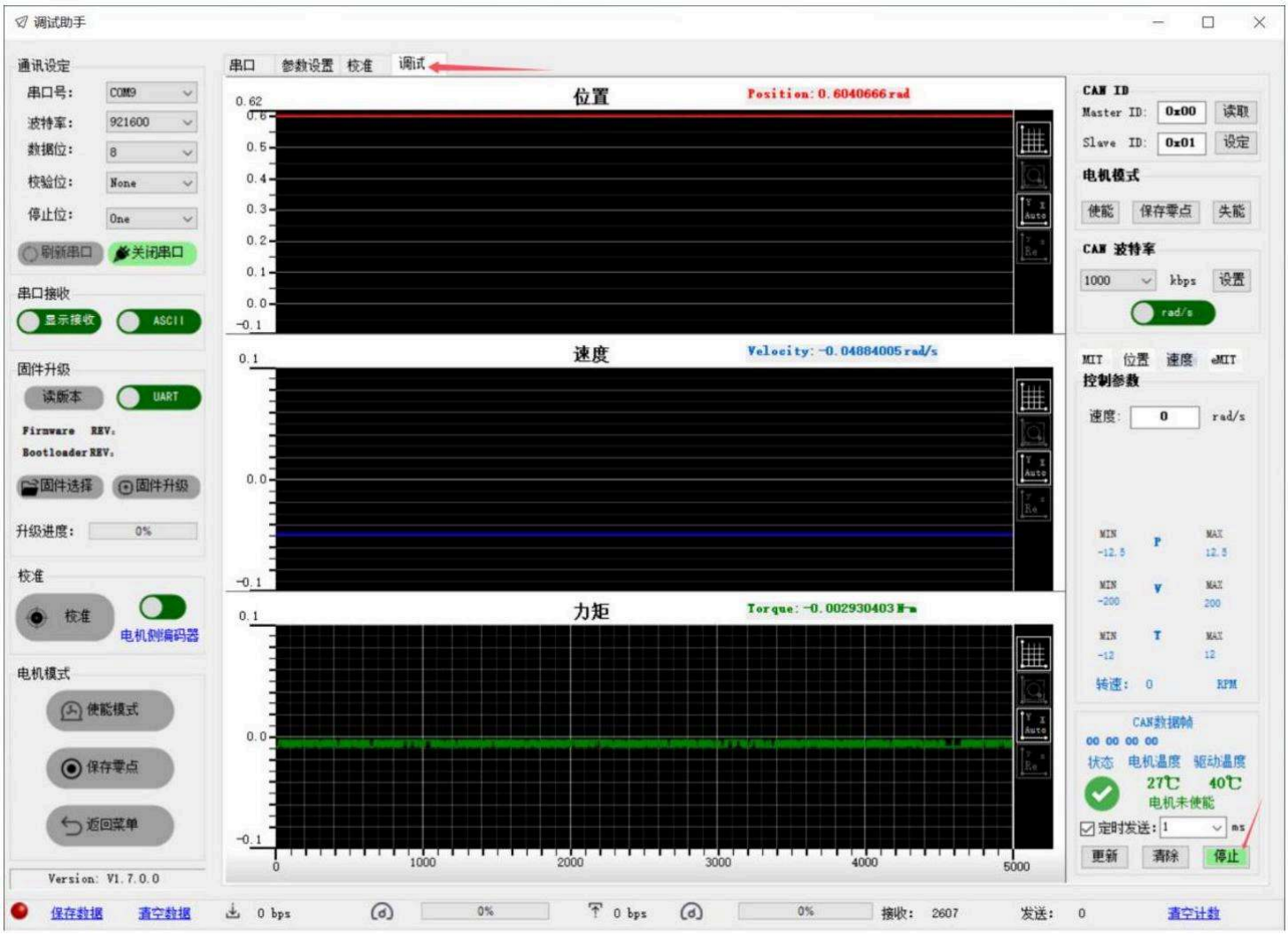

Please check whether the relevant parameters meet the requirements. If modifications are needed, enter the corresponding data in the respective fields and then click the "Write Parameters" button.

Note:
(1) Do not modify the pole pair count and reduction ratio parameters.
(2) After clicking "Write Parameters", the driver will automatically soft reboot, without the need for an external power cycle.
(3) The "Stash" button is only effective for controller parameters and will be lost after power-off.

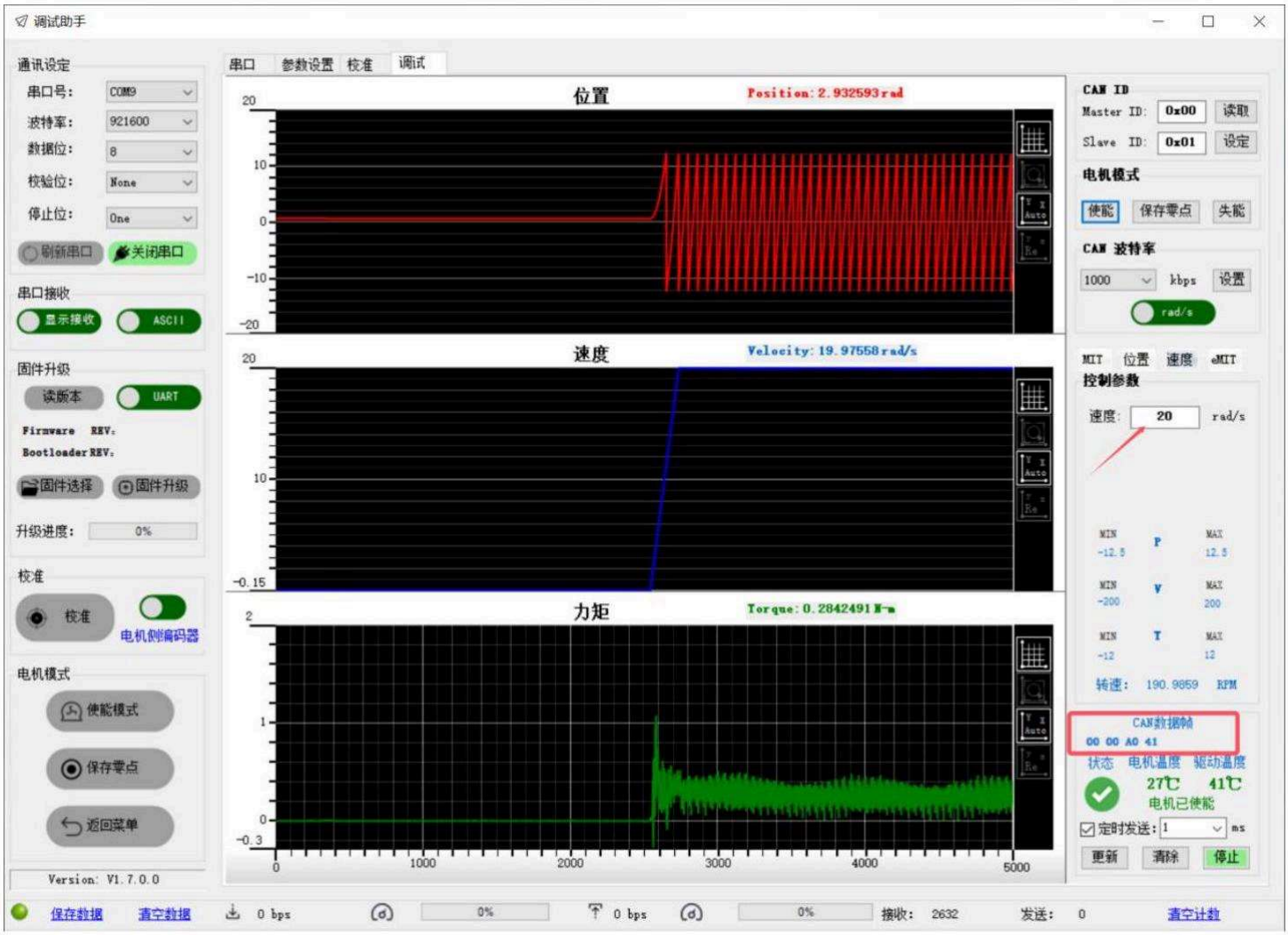
4. Debugging

After correctly performing the previous steps, you can proceed with motor operation debugging. The debugging interface uses CAN communication, so please ensure the CAN cable is correctly connected before proceeding. The motor will rotate after enabling, so it is necessary to secure the motor in this step to prevent accidents.

Click the "Debug" tab to enter the debugging interface.




Click "Send". After successful communication, the interface will display the motor status curve.
Select the corresponding control mode tab. Taking speed mode as an example, enter the speed $20\mathrm{rad/s}$ . After clicking "Update", the CAN data frame will automatically update the data. At this point, the data will be sent to the driver. After clicking "Enable", the motor will rotate at a speed of $20\mathrm{rad/s}$ , and the waveform on the interface will scroll, indicating that debugging is complete.




CAN Configuration Commands

Motor parameters can be configured not only via serial port using a host computer but also via the CAN bus. In addition to basic parameter configuration, mode changes and baud rate modifications have been added.

1. Read Parameters

| Message ID | Attributes | D[0] | D[1] | D[2] | D[3] | D[4] | D[5] | D[6] | D[7] |
|---|---|---|---|---|---|---|---|---|---|
| 0 × 7FF | STD | CANID_L | CANID_H | 0 × 33 | RID | xx (don't care) | | | |

RID is the register address, as defined in Section 6.
Upon successful reading, the data from that register will be returned, with the frame format 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 | 0 × 33 | RID | Data | | | |

The data is either floating-point or unsigned integer, occupying 32 bits (4 bytes) in total, with the lowest byte being D4 and the highest being D7. The same applies below.

2. Write Parameters

| 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 | | | |

With the RID as above, after writing the data, the driver will perform data range validation. If it is within the specified range, the write will be successful, the driver will immediately load the corresponding parameters, and return the written data. The frame format is the same as that 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 | 0 × 33 | RID | Data | | | |

If out of range, the write fails, and the driver returns the original register data.

Writing register data takes effect immediately but cannot be stored; it will be lost after power-off. You need to send a command to store parameters to write all modified parameters to the chip.

3. Storing Parameters

| Message ID | Attributes | D[0] | D[1] | D[2] | D[3] |
|---|---|---|---|---|---|
| 0x7FF | STD | CANID_L | CANID_H | 0xAA | XX |

After successful writing, the return format is:

| Message ID | Attributes | D[0] | D[1] | D[2] | D[3] |
|---|---|---|---|---|---|
| MST_ID | STD | CANID_L | CANID_H | 0xAA | 01 |

Note:
(1) Storing parameters only takes effect in disabled mode.
(2) All parameters will be saved at once when storing parameters.
(3) This operation writes parameters to the on-chip flash, with a maximum operation time of 30 ms each time. Please ensure sufficient time is allocated.
(4) The flash memory can be erased and written approximately 10,000 times. Please do not frequently send the "store parameters" command.

4. Mode Switching

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

| Encoding | Mode |
|---|---|
| 1 | MIT |
| 2 | Position Velocity |
| 3 | Velocity |

By modifying the value of the mode register (0x0A), the mode can be changed. When switching modes, the motor first clears the command values, including position, velocity, and the torque feedforward, KP, and KD values in MIT mode.

When switching from one mode to position control mode, to prevent impact, it is recommended to read the precise position first before considering switching, and to switch when the motor speed is zero if possible.

After the mode is modified, it will not be saved in the flash and will be lost after power-off. After repowering, the control mode will be set to the mode previously saved in the flash.

5. Baud Rate Modification

By writing specific data to the baud rate register (address $0 \times 23$ ), the current CAN communication baud rate can be modified. The driver supports specific baud rate modification, and the currently supported baud rates are as follows:

| Encoding | Baud Rate |
|---|---|
| 0 | 125 K |
| 1 | 200 K |
| 2 | 250 K |
| 3 | 500 K |
| 4 | 1 M |
| 5 | 2 M |
| 6 | 2.5 M |
| 7 | 3.2 M |
| 8 | 4 M |
| 9 | 5 M |

After successfully modifying the baud rate, the driver will first feedback data at the original baud rate, and then communicate at the new baud rate. After power-on, the motor first determines the stored baud rate. If it is greater than 5 Mbps, it automatically defaults to 1 Mbps. For baud rates greater than 1 Mbps (not including 1 Mbps), it will automatically switch to CAN FD function; if the baud rate is less than or equal to 1 Mbps, it automatically switches to CAN 2.0 B. Motors set to CAN FD can still receive CAN 2.0B data frames, but when sending feedback frames, CAN FD is used, so the upper-layer controller will not receive feedback data, and the driver will continuously report errors. If a CAN 2.0 OB controller sets the wrong ID, it can still change the baud rate back by using the command to modify the baud rate.

## 6. Register Address

| Address (HEX) | Address (DEC) | Variable | Description | Read/Write | Range | Type |
|---|---|---|---|---|---|---|
| 0x00 | 0 | UV_Value | Under-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 |
| 0 × 03 | 3 | OC_Value | Overcurrent protection value | RW | (0.0, 1.0) | float |
| 0x04 | 4 | ACC | Acceleration | RW | (0.0, fmax) | float |
| 0 × 05 | 5 | DEC | Deceleration | RW | [-fmax, 0.0) | float |
| 0 × 06 | 6 | MAX_SPD | Maximum Speed | RW | (0.0, fmax] | float |
| 0 × 07 | 7 | MST_ID | Feedback ID | RW | [0, 0x7FF] | uint32 |
| 0 × 0 A | 8 | ESC_ID | Receive ID | RW | [0, 0x7FF] | uint32 |
| 0 × 09 | 9 | TIMEOUT | Timeout alarm time | RW | [0, 2^32-1] | uint32 |
| 0 × 0 A | 10 | CTRL_MODE | Control Mode | RW | [0, 4] | uint32 |
| 0x0B | 11 | Damp | Motor Viscosity Coefficient | Ro | / | float |
| 0x0C | 12 | Inertia | Motor Rotation Inertia | Ro | / | float |
| 0 × 0D | 13 | hw_ver | Reserved | Ro | / | uint32 |
| 0x0E | 14 | sw_ver | Software Version Number | Ro | / | uint32 |
| 0 × 0 F | 15 | SN | Reserved | Ro | / | uint32 |
| 0 × 10 | 16 | NPP | Motor Pole Pairs | Ro | / | uint32 |
| 0 × 11 | 17 | Rs | Motor phase resistance | Ro | / | float |
| 0 × 12 | 18 | Ls | Motor phase inductance | Ro | / | float |
| 0 × 13 | 19 | Flux | Motor Flux Value | Ro | / | float |
| 0 × 14 | 20 | Gr | Gear Reduction Ratio | Ro | / | float |
| 0 × 15 | 21 | PMAX | Position Mapping Range | RW | (0.0, fmax] | float |
| 0 × 16 | 22 | VMAX | Speed mapping range | RW | (0.0, fmax] | float |
| 0 × 17 | 23 | TMAX | Torque Mapping Range | RW | (0.0, fmax] | float |
| 0 × 18 | 24 | I_BW | Current Loop Control Bandwidth | RW | [100.0, 1.0e4] | float |
| 0 × 19 | 25 | KP_ASR | Speed Loop Kp | RW | [0.0, fmax] | float |
| 0 × 1 A | 26 | KI_ASR | Speed Loop Ki | RW | [0.0, fmax] | float |
| 0 × 1 B | 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 |
| 0 × 1 F | 31 | Deta | Speed Loop Damping Coefficient | RW | [1.0, 30.0] | float |
| 0 × 20 | 32 | V_BW | Speed Loop Filter Bandwidth | RW | (0.0, 500.0) | float |
| 0 × 21 | 33 | IQ_c1 | Current loop enhancement coefficient | RW | [100.0, 1.0e4] | float |
| 0 × 22 | 34 | VL_c1 | Speed loop enhancement coefficient | RW | (0.0, 1.0e4] | float |
| 0 × 23 | 35 | can _ br | CAN Baud Rate Code | RW | [0, 4] | uint32 |
| 0 × 24 | 36 | sub_ver | Sub-version number | Ro | / | uint32 |
| 0 × 32 | 50 | u_off | u-phase offset | Ro | / | float |
| 0 × 33 | 51 | v_off | v-phase offset | Ro | / | float |
| 0 × 34 | 52 | k1 | Compensation Factor 1 | Ro | / | float |
| 0 × 35 | 53 | k2 | Compensation Factor 2 | Ro | / | float |
| 0 × 36 | 54 | m_off | Angle Offset | Ro | 1 | float |
| 0 × 37 | 55 | dir | Direction | Ro | 1 | float |
| 0 × 50 | 80 | p_m | Motor current position | Ro | / | float |
| 0 × 51 | 81 | xout | Output shaft position | Ro | / | float |

RW: Read/Write.
RO: Read Only.

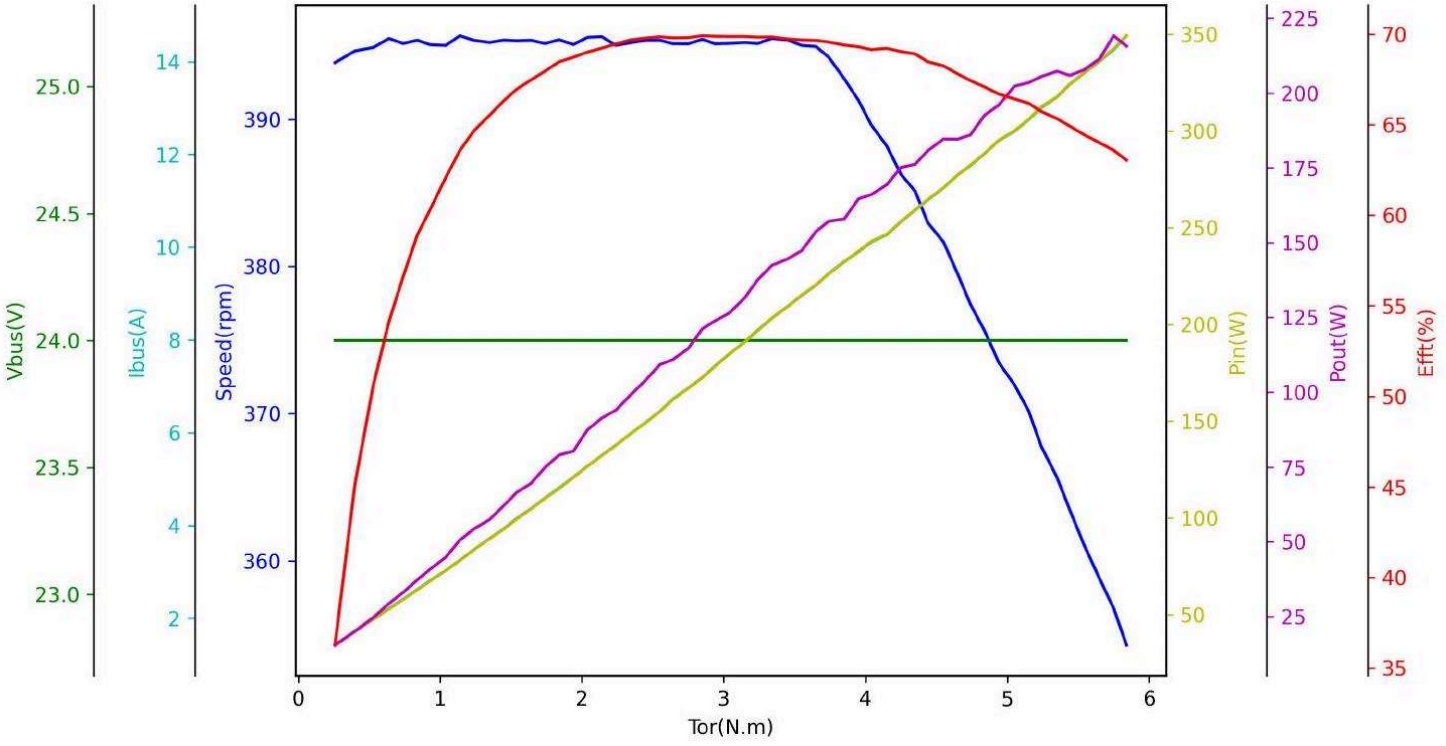
*Firmware Upgrade

When new features are added to the firmware, or an upgraded version is released to fix bugs, users can perform an upgrade via the serial port to resolve issues and use new features. Before use, connect to the serial port, then click "Firmware Selection", select the corresponding firmware, confirm, and then click the firmware upgrade button. Wait for the upgrade progress bar to complete, or observe the serial port interface to see if the upgrade is finished.


\captionsetup{labelformat=empty}
Figure 1: Note: The firmware name must match the motor. Ensure you do not select the wrong one.

Characteristic Curve

Test Environment: Room temperature 25°C , laboratory environment.



## Characteristic Parameters

Please use the motor appropriately based on the following parameters.

| | | |
|---|---|---|
| Motor Parameters | Rated Voltage | 24 V (Driver supports 15 ~ 52 V power supply) |
| | Rated Phase Current (Power Supply Current) | 9.2A (8.6A) |
| | Peak Phase Current (Power Supply Current) | 20.5A (16.1A) |
| | Rated Torque | 3.5 Nm |
| | Peak Torque | 7.8 Nm |
| | Rated Speed | 395 rpm |
| | No-load Max Speed | 435 rpm |
| Motor Characteristic Values | Reduction Ratio | 3591/187 (1:19.2) |
| | Number of Pole Pairs | 7 |
| | Phase Inductance | 55uH |
| | Phase Resistance | 0.2Ω |
| Structure and Weight | Outer Diameter | Motor 42 mm |
| | Height | Motor 91.5 mm |
| | Motor Weight | Approx. 396 g |
| Encoder | Encoder Type | Incremental Encoder |
| Communication | Control Interface | CAN |
| | Parameter Tuning Interface | UART@921600bps |

| | | |
|---|---|---|
| Control and Protection | Control Mode | MIT Mode |
| | | Speed Mode |
| | | Position Mode |
| | Protection | Driver over-temperature protection, protection temperature: 120°C , the over-temperature motor will exit "enable mode" |
| | | Motor over-temperature protection, set according to usage requirements, recommended not to exceed 100°C , the over-temperature motor will exit "enable mode" |
| | | Motor overvoltage protection, set according to usage requirements, it is recommended not to exceed 60 V. Overvoltage will exit "enable mode". |
| | | Communication loss protection. If no CAN command is received within the set period, the motor will automatically exit "enable mode" |
| | | Motor overcurrent protection, set according to usage requirements, it is recommended not to exceed 20A (phase current). Overcurrent will exit "enable mode". |
| | | Motor undervoltage protection: If the power supply voltage is lower than the set value, it will exit "enable mode". The power supply voltage should not be lower than 15 V. |