

# Product Manual

## RMD-X4-P36-24

### Disclaimer

Thanks for choosing RMD series motor. Please read this statement carefully before using. Once used, this statement is deemed to be approved and accepted. Please install and use this product strictly in accordance with the manual, product description and relevant laws, regulations, policies and guidelines. In the process of using the product, the user undertakes to be responsible for his own behavior and all consequences arising therefrom.

MYACTUATOR will not be liable for any loss caused by improper use,

installation and modification of the user.

MYACTUATOR is the trademark of Suzhou Micro Actuator Technology Co., Ltd. and its affiliates. Product names, brands, etc. appearing in this document are trademarks or registered trademarks of their respective companies.

This product and manual are copyrighted by MYACTUATOR. Reproduction in any form is not permitted without permission. The final interpretation right of the disclaimer belongs to MYACTUATOR.

### Manual Version V1.0

## Contents

<b>Introduction .....</b>	<b>1</b>
<b>1. Product Features .....</b>	<b>1</b>
<b>2. Product Interface, Coordinate System Definition, and Parts Description .....</b>	<b>2</b>
<b>3. Product Installation Dimensions .....</b>	<b>2</b>
<b>4. Product Parameters, Measured Data, and Characteristic Curves .....</b>	<b>2</b>
<b>5. Product Instructions and Functionalities Descriptions .....</b>	<b>3</b>
5.1 Encoder Function Description .....	3
5.2 Instructions for Using Voltage and Power Supply .....	4
5.3 Communication Instructions .....	5
5.4 Explanation of Force Position Hybrid Control Mode .....	5
5.5 Description of Servo Position Control Mode .....	6
5.6 Description of Servo Speed Control Mode .....	6
5.7 Description of Current/Torque Control Mode .....	6
5.8 Motor Braking Control Instructions .....	6
5.9 Indicator Light Description .....	7
<b>6. Debugging using VESC tool .....</b>	<b>7</b>
<b>7. Typical Applications .....</b>	<b>12</b>
<b>8. Precautions .....</b>	<b>12</b>
<b>Appendix 1: CAN Communication Protocol .....</b>	<b>13</b>



## Introduction

EC-A4310-P2-36 is a joint module product that integrates reducer, motor, and motor drivers. The reduction method is a planetary reducer, and all gears have been tempered to ensure the reducer's high strength and long service life. The motor adopts a centralized winding brushless DC motor scheme with high power density and efficiency. After optimizing the pole arc coefficient, the motor has a lower cogging torque and runs more smoothly. At the same time, the product is equipped with a self-developed high-performance driver. A magnetic flux observation control is also added to the servo control, which can adapt to different loads and has an extensive fixed speed bandwidth. It does not need frequent parameter tuning like traditional three-loop PID control and is simple and easy to use. The drive also has functions such as temperature protection and overload protection, ensuring the stable and reliable operation of the product.

### 1. Product Features

- **Dual Encoder:**

A 14-bit encoder is installed at the output end of the motor, and a linear Hall sensor is used at the control end of the motor to enable real-time feedback on the output position of the product, making it easy for users to use and eliminating the need for continuous multi-turn memory, improving product reliability.

- **Multiple Control Modes:**

The control modes provided include force position hybrid control mode, servo position mode, servo speed mode, current (torque) control mode, and braking mode.

- **Flux observation control:**

In servo control mode, it is unnecessary to adjust the parameters frequently to adapt to the load or even to adjust parameters.

- **Wide speed control bandwidth:**

Constant speed control can range from 0 to 95% of the maximum speed. The speed control step size is 5 ERPM (ERPM is the electrical rpm = rpm \* motor pole number).

- **Accurate position control:**

In servo position control mode, within the load range the motor can withstand, the position error is less than 0.1° (limited by the encoder accuracy).

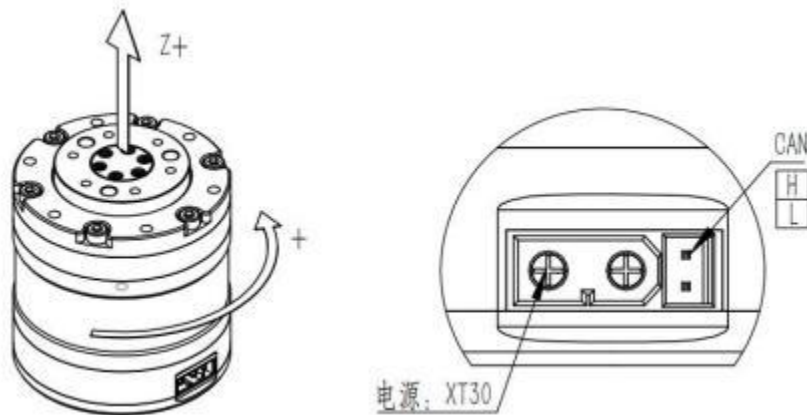
- Equipped with **over-temperature, over-voltage, and overload protection.**

- **Equipped with a USB interface:**

No other modules are required for use, and it can be directly connected to a PC for debugging and control

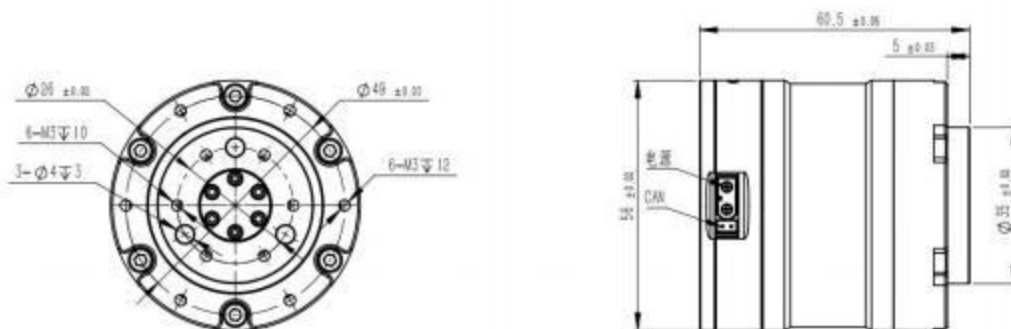
## 2. Product Interface, Coordinate System Definition, and Parts

### Description



While facing the output shaft of the module, speed is positive for counterclockwise rotation, and the angle increases as rotation continues; speed is negative for clockwise rotation, and the angle decreases as rotation continues.

## 3. Product Installation Dimensions



## 4. Product Parameters, Measured Data, and Characteristic Curves

Performance Parameters		
Reducer ratio		36
Nominal Voltage	V	24
Nominal Current	A	6.1

Nominal Power	W	146.4
Nominal Speed	rpm	78
Nominal Torque	Nm	12
Peak Speed	rpm	93
Peak Current	A	18
Peak Torque	Nm	36
Efficiency		70%
Torque Constant	Nm/A	1.4
Motor Weight	g	377
Backlash	Arcmin	10
Motor Size (OD × Thickness)	mm	56 × 60.5
Communication Method		CAN
Communication Baud Rate		1M

The speed shown in this manual refers to the motor speed at the output end unless otherwise specified.

Product Measured Data							
Voltage (V)	Bus Current (A)	Output Power (W)	Torque (Nm)	Speed (rpm)	Phase Current (A)	Efficiency	Torque / Phase Current
24	0.9	21.6	1	89	1.0	42%	1
	1.4	33.6	2	88	1.6	54%	1.25
	1.7	40.8	3	87	2.2	64%	1.36
	2.3	55.2	4	86	3.0	65%	1.34
	2.75	66	5	85	3.7	68%	1.35
	3.2	76.8	6	84	4.4	71%	1.36
	3.6	86.4	7	84	5.1	71%	1.37
	4	96	8	83	5.6	71%	1.43
	4.65	111.6	9	82	6.4	70%	1.41
	5.1	122.4	10	81	7.1	70%	1.41
	5.5	132	11	80	7.8	70%	1.41
	6.1	146.4	12	78	8.6	70%	1.40
	6.6	158.4	13	77	9.6	66%	1.35
	7.3	175.2	14	77	10.3	64%	1.36
	8	192	15	74	11.5	60%	1.30
8.9	213.6	16	70	12.7	56%	1.26	
10	240	17	65	14.7	49%	1.16	

Different individuals have certain differences; the difference range is within 5%.

The characteristic curves are still under development.

## 5. Product Instructions and Functionalities Descriptions

### 5.1 Encoder Function Description

The motor output end encoder uses a 14-bit magnetic encoder. According to actual measurements, the highest angle accuracy can only reach 0.1 degrees. Due to encoder quality issues, some only have an accuracy of 0.2 degrees. This result has been compared with several common magnetic resistance encoders on the market that

use 1-pole magnet feedback, and the common magnetic resistance encoders are all in this level of accuracy.

The motor can provide real-time feedback on the position of the output end, which is indeed 0~360°. To facilitate a unified coordinate system, it is divided into -180°~180°. However, the motor only has a single-turn memory function. For example, when the motor is powered on at a 170° angle. If the motor is kept powered on and turned counterclockwise by 20°, the motor angle is now at 190°. If the motor is powered off at 170° and turned counterclockwise by 20°, the angle is read as -170° when powered on. If, instead, the motor is rotated clockwise by 20°, the motor angle is read as 150° when powered on.

## 5.2 Instructions for Using Voltage and Power Supply

### . Use of Voltage

The voltage range of the motor is 10~57V. At 57V~60V, the driver will alarm; exceeding 60V may damage the driver; at 8V~10V, the driver can be used, but low voltage alarms are prone to occur; At 6V~8V, the driver can be turned on, but there will be a low voltage alarm; if voltage is lower than 6V, the driver cannot be turned on.

When under load, try to use battery power as much as possible to ensure sufficient power supply capacity and energy return channels. If using desktop laboratory power supply, switch power supply, self-made power management modules, or other equipment for power supply, pay attention to adding capacitors to provide sufficient withstand voltage and capacity to guarantee energy return. Otherwise, the driver may be burned easily or cause it to alarm or reboot. Reboot is a hardware protection function. If the driver does not reboot, driver hardware tends to be damaged easily and renders it unusable.

When using desktop laboratory power supplies, switch power supplies, self-made power management modules, or other devices for power supply, it tends to cause the drive to fail to start smoothly. The main reason is that the driver has a voltage protection mechanism. Therefore, when the power supply device has a voltage climbing process, the driver judges it as having an unstable power supply. The solution to this problem is to replace the current power supply with a battery power supply or contact official technical support.

When using desktop laboratory power supplies, switch power supplies, self-made power management modules, or other devices for power supply, it tends to cause the drive to fail to start smoothly. The main reason is that the driver has a voltage protection mechanism, when the power supply device has a voltage climbing process, the driver judges it as having unstable power supply. The solution to this problem is to replace current power supply to battery power supply or contact official technical support.



. Power Supply and Current

The product power supply port is an XT30 (2+2) connector produced by AMASS, with a rated current of 15A and a peak value of 30A. The power supply current is the bus current to the driver. When the motor is in actual control, the power supply current is phase current because the phase current is linearly related to the motor torque. If it is necessary to obtain the motor power, bus current, or other status, relevant reading instructions can be sent to obtain them.

### 5.3 Communication Instructions

CAN baud rate is 1M, standard frame. CAN control frequency can reach up to 2000Hz. After one frame is sent, the feedback data frame can be received, at the fastest, in 0.4ms. The system has heartbeat protection, with a time interval of 500ms. The motor will automatically stop moving if a second frame control command is not received after sending a CAN control command.

When using the motor for the first time, checking or resetting the motor ID before controlling is recommended. Most factory parameters and settings can be used directly. It is suggested that users should only modify the parameters or settings with experience. Control commands can be sent directly to the configured motor after the driver starts successfully (at least 5s). The configuration can be changed at any time during this period.

When sending the following setting commands to the drive, users should delay at least 500ms before sending a second CAN command.

- . Set communication mode
- . Set the current point to zero
- . Set motor ID

If the delay is not greater than 500ms, the setting may fail.

For other commands, users only need to avoid causing CAN busy.

CAN control and USB control cannot be used simultaneously. If used simultaneously, CAN control takes priority, but USB control cannot be completely shielded. CAN control can be used while using USB to view motor feedback data.

### 5.4 Explanation of Force Position Hybrid Control Mode

The following command uses the force position hybrid control mode. The final output current goes to the motor. The motor current is calculated as below:

$$\begin{aligned} \text{motor\_current} = & \\ & (KP * (\text{pos\_desired} - \text{pos\_current}) + \\ & KD * (\text{spd\_desired} - \text{spd\_current}) + \text{Torque}) / K_T \end{aligned}$$

The KP and KD parameters in the command can be adjusted at any time, and KT is the motor torque constant. This mode is mainly used for the control of foot robots. Without load, the reference values of KP and KD, respectively, are 15 and 0.5.

## 5.5 Description of Servo Position Control Mode

If the given position is greater than  $\pm 180$  degrees, the motor will be able to turn multiple rounds, but rebooting will cause the number of turns to reset.

The servo control mode is a three-loop control. The innermost loop is the current loop, mainly controlled by a PI controller, and the user cannot change the parameters. The outmost loop is KP, KD control. KP is the feedback compensation gain, and KD is the damping coefficient in this system. Middle loop is the speed loop, which uses servo control mode to control speed. An adaptive algorithm developed by our team is used, which allows users to have no need to adjust the parameters frequently to adapt load. KP and KD of the position loop correspond to values 0 ~ 0.1 in the VESC tool software, with reference values of KP: 0.006 and KD: 0.

The servo position control has three input parameters: expected position P, expected speed V, and current threshold I. The motor moves from the current position P0 to the expected position P at the expected speed V. During the process from P0 to P, the maximum speed is V. When P0 and P are particularly small, and V is set too large, the actual speed may not reach V. During the process of moving from P0 to P, the maximum motor current is I, and the maximum torque generated is  $T_{max} = I * \text{torque constant}$ . If the load encountered exceeds the maximum torque, then the motor will stall and maintain the torque  $T_{max}$ .

## 5.6 Description of Servo Speed Control Mode

The speed loop control is a magnetic flux observation control with many parameters. The parameters provided to the customer for modification are the **magnetic flux observation gain** and the **disturbance compensation coefficient**. The recommended value range for the magnetic flux observation gain is 0.0001 ~ 1, with a reference value 0.1. The recommended value range for the disturbance compensation coefficient is 0 ~ 1, with a reference value 0.06. The factory set values have been tested and can adapt to most load situations. It is recommended that users do not change this value arbitrarily and only change it after having relevant experience or under official technical guidance (the above range of values is for the settings on the VESC tool; please refer to the communication protocol for the setting range in CAN)).

## 5.7 Description of Current/Torque Control Mode

When using the current/torque control mode to control the motor, as long as it rotates, it will inevitably continue accelerating to the highest speed. Precautions should be made accordingly during debugging!

## 5.8 Motor Braking Control Instructions

There are three motor braking modes: variable damping braking control mode, energy consumption braking mode, and regenerative braking mode. The energy



consumption braking mode and the regenerative braking mode are recommended to be used under no-load or light load conditions. The variable damping braking mode achieves its purpose by calling the command to set the motor duty cycle to 0, similar to the effect of three-phase short circuiting. The variable damping braking mode can be used under load. The most recommended method for braking is to use speed loop control to brake.

## 5.9 Indicator Light Description

- . Green light slightly on: Normal power on.
- . Flashing or enhancing green light: Received CAN command and operating normally
- . Flashing red light: The motor encountered error. The specific error information can be viewed through CAN query or VESC tool.

## 6. Debugging using VESC tool

This driver program is a self-developed FOC algorithm that utilizes the interaction between the VESC tool and the APP. The underlying layer is entirely different from the official firmware. Therefore, many parameters on the VESC tool are not the same as the display. Therefore, special attention should be paid to parameter modification! If you are unfamiliar with VESC, it is recommended to use only the master computer as an observer to view the driver data.

The steps for debugging using the VESC tool on Windows are as follows. For Linux use, please refer to the official VESC tutorial or contact our team's technical support.

① Install the VCP driver and download VESC tool version 2.06 or higher (2.06 recommended). VCP drivers and VESC tools can be requested from our team's technical support or downloaded from the official website.

VCP download address:

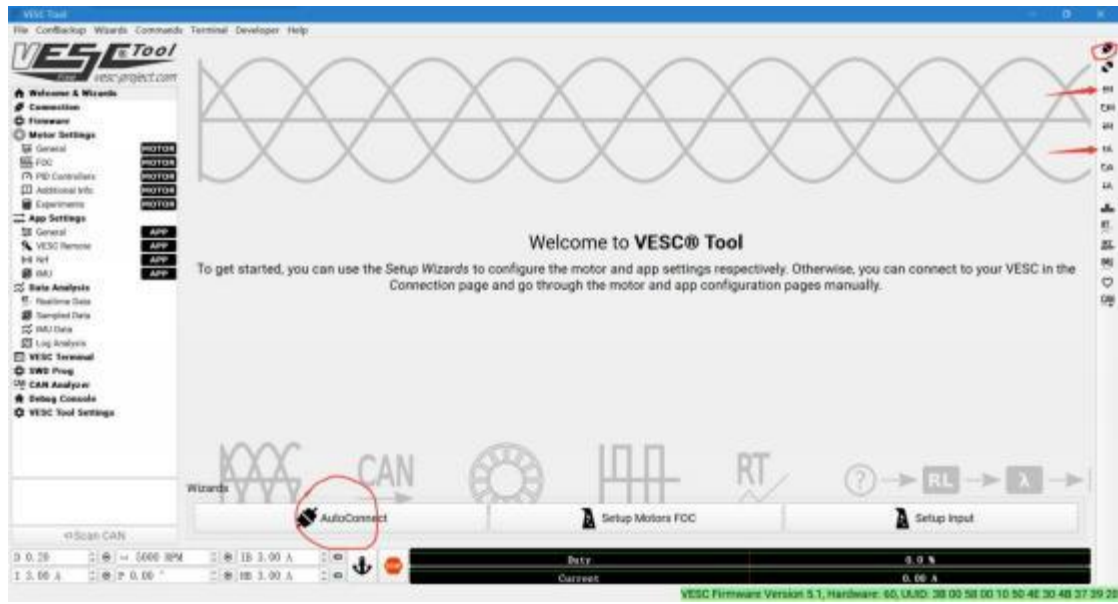
<https://www.st.com/en/development-tools/stsw-stm32102.html#get-software>

VESC tool download address:

[https://vesc-project.com/vesc\\_tool](https://vesc-project.com/vesc_tool)

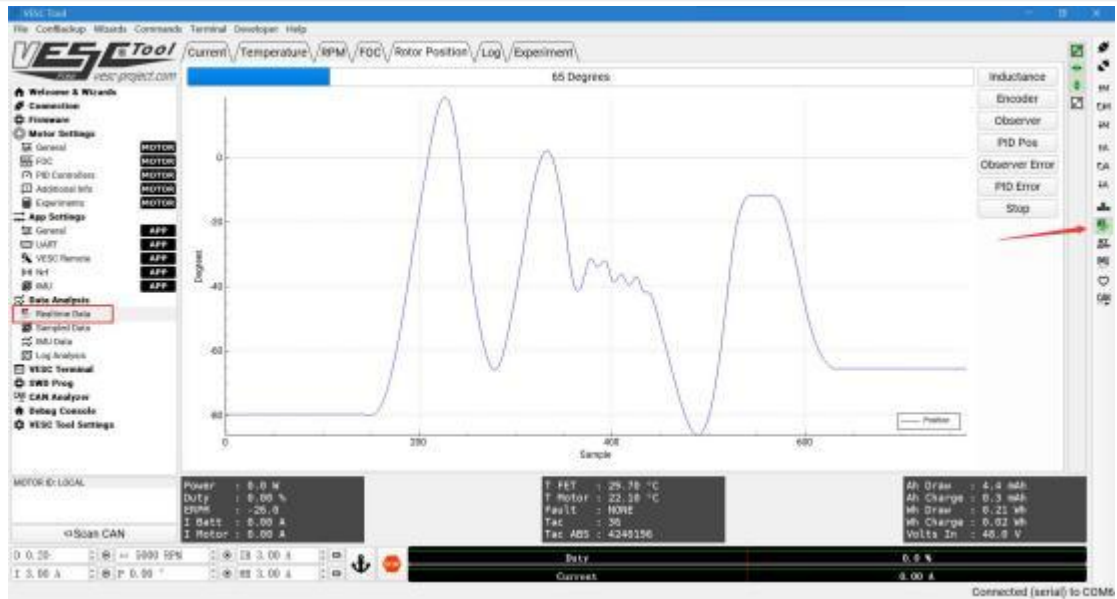
② Power on the module (if purchasing a product with detached driver and module, please connect the driver and module correctly before powering on the module). Use a micro USB cable to connect the module to the computer USB port. Open the VESC tool and click the motor button at the upper right corner to connect. After a successful connection, first, upload the motor configuration parameters and APP configuration parameters to the VESC tool (shown in the figure below with red

arrows) to avoid accidentally replacing the parameters in the motor. Please refer to the following figure for details.

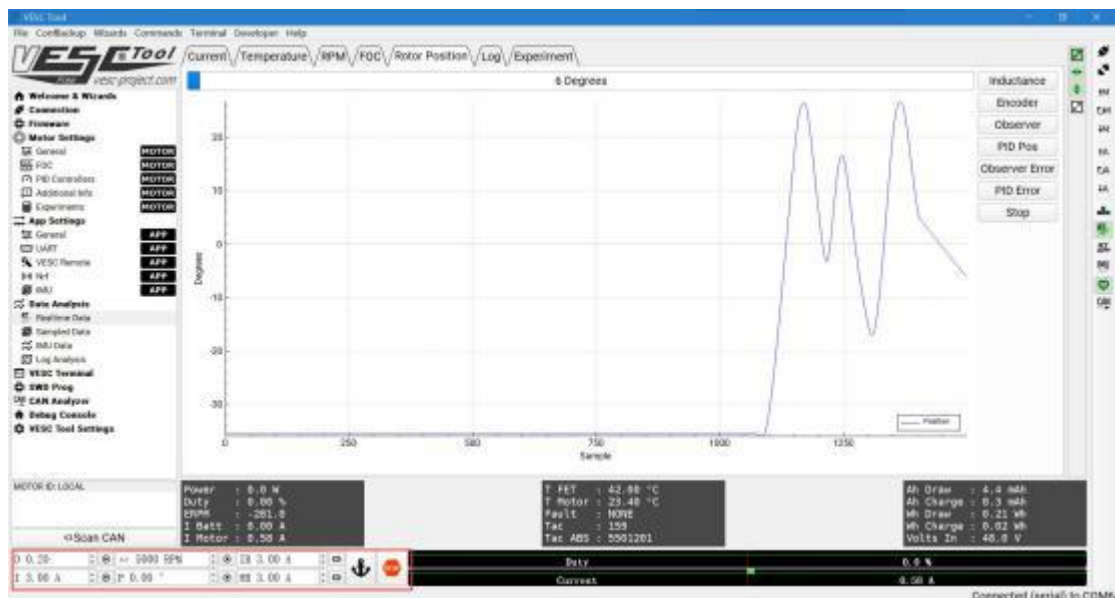


Attention! If the configuration is downloaded without uploading parameters, it will overwrite the parameters in the driver, causing the motor to fail to run smoothly or even damage it. Some parameters require actual testing to be obtained. Therefore, it may be necessary to send the motor back for repair. If only the parameters needed to be reconfigured, there would be no repair fee, and the user only needs to cover the shipping cost. If the motor has been damaged, our team has no responsibility.

③ Click the RT button on the right, then click Realtime Data under the Data Analysis directory on the left to view real-time data of the driver, such as power, voltage, error information, speed, position, phase current, etc.

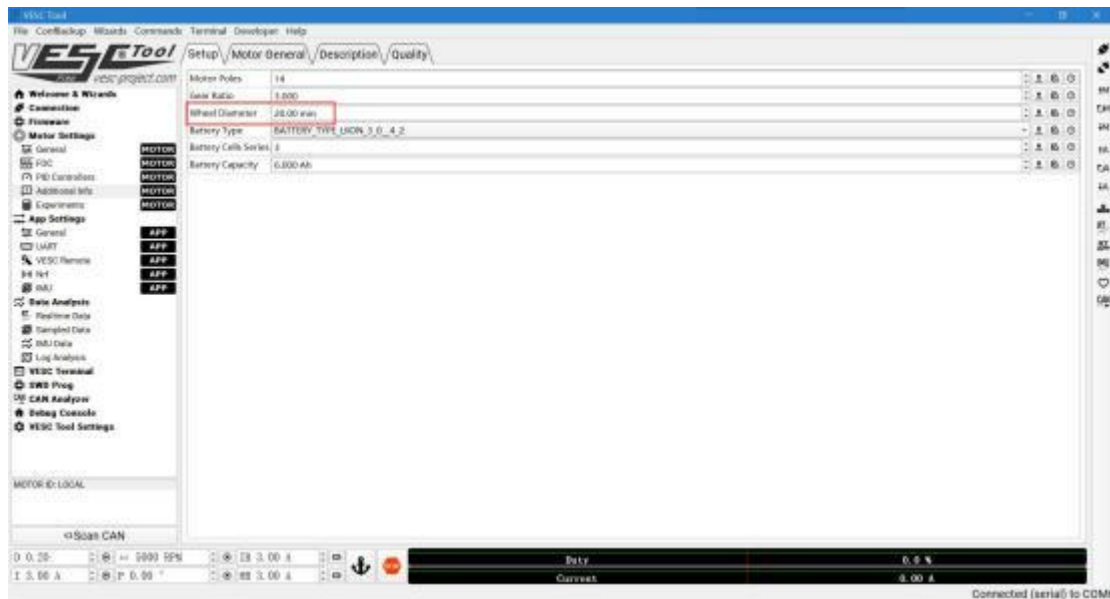


④ Enter numerical values in the input box at the bottom right corner and click the small button on the right side of the input box to control the motor. D is duty cycle control;  $\omega$  is constant speed control, which calls the magnetic flux observation control function, the same as the servo speed loop control in CAN control. Note that the input parameters are actually ERPM; I is current control; P is servo position loop control; IB is regenerative braking control; HB is energy consumption control; Anchor symbol is variable damping braking, which calls for the control command that sets duty cycle to 0.



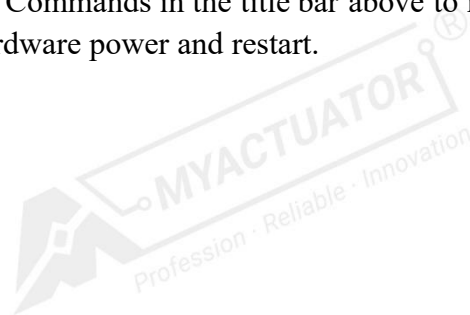
In position control, servo position control is used, and the desired speed can be given by the Setup -> Wheel Diameter on the Additional Info page under the left

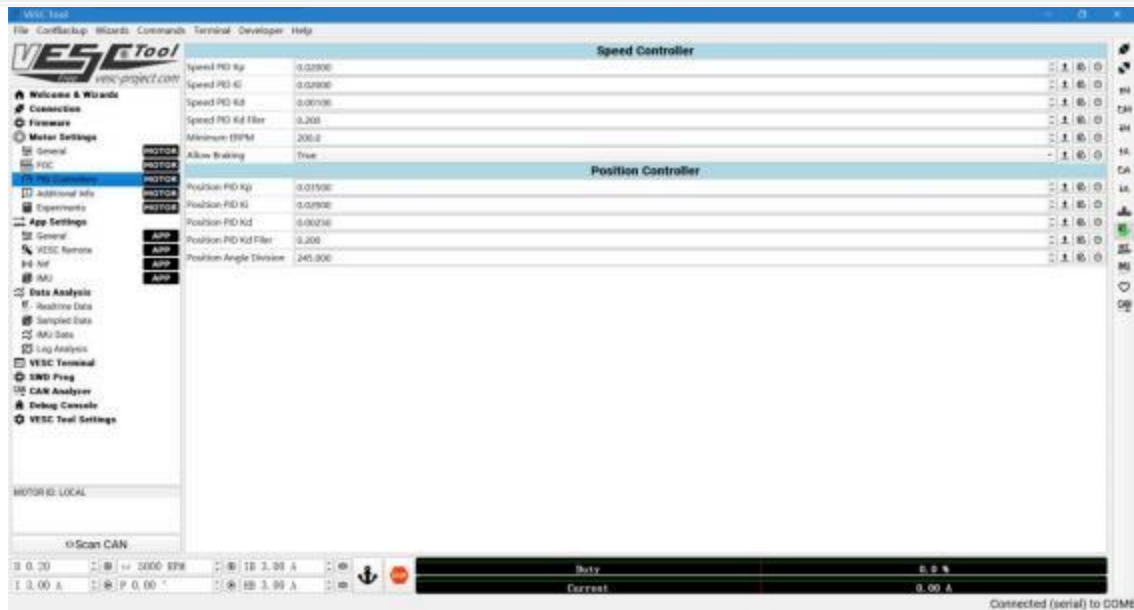
Motor Setting column. The corresponding relationship is: Wheel Diameter parameter \* 100 = desired electrical angular velocity.



⑤ Click PID controllers in the Motor setting directory on the left to modify parameters such as flux observation gain, disturbance compensation coefficient, feedback gain, etc. (it is recommended that users modify them under our team's technical guidance). Among them, speed PID K<sub>p</sub> corresponds to the disturbance compensation coefficient; speed PID K<sub>i</sub> corresponds to flux observation gain; position PID K<sub>p</sub> corresponds to the feedback compensation coefficient; position PID K<sub>d</sub> corresponds to the damping coefficient. After modifying the parameters on this page, click the ↓ M button on the right to download the parameters to the driver.

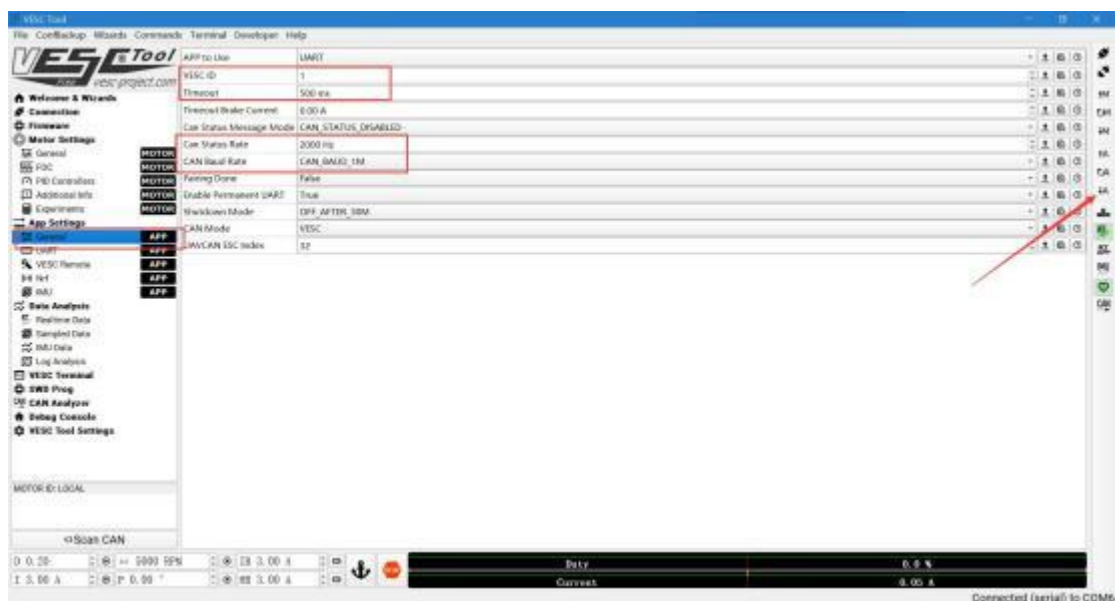
Attention! The modified parameters require restarting the driver to take effect. Restart method 1: Click reboot under Commands in the title bar above to restart the software. Restart method 2: Turn off hardware power and restart.





Attention! The flux observation gain, disturbance compensation coefficient, feedback compensation coefficient, and damping coefficient set on the VESC tool will be written into the driver FLASH as default parameters. In contrast, these three parameters modified in real-time through CAN commands will not be written into FLASH. After turning off power and restarting, the default parameters will be used.

⑥ Click General under the App Settings directory on the left to enter the settings for CAN ID, heartbeat delay, feedback frequency, and CAN baud rate (do not modify other parameters; otherwise communication may not be possible). After changing the parameters, click the ↓ A button on the right to download the App parameters to the driver.



## 7. Typical Applications

- (Biped robot)
- (Quadruped robot)
- (Biped wheeled robot)
- (Wire drive robotic arm)
- (Exoskeleton robot)
- (Pan tilt camera)

## 8. Precautions

- . This product should be used away from strong magnetic environments; otherwise, it may affect the motor encoder data and cause abnormal motor rotation!
- . When using this product, the user should ensure it is stable within 70°C . If the temperature exceeds 70°C , the motor will alarm. If the temperature exceeds 100°C, the motor will forcibly shut down and enter a standby state, and in the meantime, it cannot be controlled. After the temperature returns to normal, the motor will return to normal state.
- . When this product is operating at high torque, it will generate a lot of heat. Be careful not to get burned!
- . When debugging or working on this product, the user should ensure the casing is fixed.
- . Do not disassemble this product on your own, as it may affect its normal use. If you disassemble it without permission, the corresponding warranty service will not be provided.
- . Pay attention to secure wiring and avoid poor contact.
- . As in winter, the weather is dry. Pay attention to static interference and avoid drive loss.



## Appendix 1: CAN Communication Protocol

### 1 CAN Communication Parameters

- . Baud rate: 1M
- . Identifier: Standard identifier (0x01~0x7FF), 0x7FF is the special identifier for motor setting commands and motor query commands, and the rest are motor IDs.
- . Data length: 4-8 bytes
- . Communication mode: Q&A mode (default), automatic message mode

### 2 Motor Setting Command

#### 2.1 Motor Communication Mode Setting Command

- . Identifier: 0x7FF (all identifiers received by the master and slave are 0x7FF)
- . Master control sends data length: 4 bytes
- . Motor feedback data length: 4 bytes

CAN Command Data Domain – Set to Automatic Message Mode			
Byte 0	Byte 1	Byte 2	Byte 3
motor ID high 8 bits	motor ID low 8 bits	0x00 (represents the instruction sent by the controller)	0x01: Set to automatic message mode 0x02: Set to Q&A mode
Setting successful return (identifier: 0x7FF)			
motor ID high 8 bits	motor ID low 8 bits	0x01 (indicating the message of the motor)	0x01: Successfully set to automatic 0x02: Successfully set as Q&A
Setting failed return (identifier: 0x7FF)			
motor ID high 8 bits	motor ID low 8 bits	0x01 (indicating the message of the motor)	0x00

#### 2.2 Motor Zero Setting Command

- . Identifier: 0x7FF (all identifiers received by the master and slave are 0x7FF)
- . Master control sends data length: 4 bytes
- . Motor feedback data length: 4 bytes

CAN Command Data Field - Set the Current Position to Zero			
Byte 0	Byte 1	Byte 2	Byte 3
motor ID high 8 bits	motor ID low 8 bits	0x00 (represents the instruction sent by the controller)	0x03
Setting successful return (identifier: 0x7FF)			
Motor ID high 8 bits	Motor ID low 8 bits	0x01 (indicating the message of the motor)	0x03
Setting failed return (identifier: 0x7FF)			
Motor ID high 8 bits	Motor ID low 8 bits	0x01 (indicating the message of the motor)	0x00

Both modes are valid and still valid after switching communication modes.  
Attention! The number of rounds the motor turned is also reset to 0!

Setting failed in the following situations:

1. The motor is in operation, and the speed is not 0 (in reality, this situation can also be caused by the motor speed being less than a certain speed).
2. The motor did not complete the previous command.
3. The motor has an alarm - high temperature alarm, encoder error alarm, etc.

To successfully set the zero point, run the motor to the desired position and maintain the position before sending the zero point setting command.

### 2.3 Motor CAN Communication ID Setting Command

- . Identifier: 0x7FF (all identifiers received by the master and slave are 0x7FF)
- . Master control sends data length: 6 bytes
- . Motor feedback data length: 4 bytes

CAN Communication Data Domain - Set Motor ID					
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte5
Old motor ID high 8 bits	Old motor ID low 8 bits	0x00	0x04	New motor ID high 8 bits	New motor ID low 8 bits
Setting successful return (identifier: 0x7FF)					
New motor ID high 8 bits	New motor ID low 8 bits	0x01	0x04		
Setting failed return (identifier: 0x7FF)					
Old motor ID high 8 bits	Old motor ID low 8 bits	0x01		0x00	

### 2.4 Motor CAN Communication ID Reset Command

- . Identifier: 0x7FF (all identifiers received by the master and slave are 0x7FF)
- . Master control sends data length: 6 bytes
- . Motor feedback data length: 4 bytes

Reset Motor ID to 0x01 in Two Modes					
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte5
0x7F	0x7F	0x00	0x05	0x7F	0x7F
Set successfully and return (identifier: 0x7FF) Data length: 4 bytes					
0x7F	0x7F	0x01	0x05	0x7F	0x7F
No response if the setting fails					

It is recommended to reset the ID to 1 for the first time and then send the ID modification command to modify it to the desired ID number.

## 3 Motor Query Command

### 3.1 Query motor communication mode

- . Identifier: 0x7FF (all identifiers received by the master and slave are 0x7FF)

- . Master control sends data length: 4 bytes
- . Motor feedback data length: 4 bytes

CAN Command Data Domain - Query Motor Communication Mode			
Byte 0	Byte 1	Byte 2	Byte 3
Motor ID high 8 bits	Motor ID low 8 bits	0x00 (represents the instruction sent by the controller)	0x81
Query successful return (identifier: 0x7FF)			
Motor ID high 8 bits	Motor ID low 8 bits	0x01 (indicating the message of the motor)	0x01: Automatic message mode
Query failed and returned (identifier: 0x7FF)			
Motor ID high 8 bits	Motor ID low 8 bits	0x01 (indicating the message of the motor)	0x80

### 3.2 Query CAN Communication ID

- . Identifier: 0x7FF (all identifiers received by the master and slave are 0x7FF)
- . Master control sends data length: 4 bytes
- . Motor feedback data length: 5 bytes for success and 4 bytes for failure

CAN Command Data Domain - Query Motor Communication Mode				
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
0xFF	0xFF	0x00 (represents the instruction sent by the controller)	0x82	
Query successful return (identifier: 0x7FF), data length 5 bytes				
0xFF	0xFF	0x01 (indicating the message of the motor)	Motor ID high 8 bits	Motor ID low 8 bits
Query failed and returned (identifier: 0x7FF), data length 4 bytes				
0x80	0x80	0x01 (indicating the message of the motor)	0x80	

**Attention!**All motors mounted on a CAN bus need to process 0X7FF data frames. The ID contained in the frame data interval is used to identify whether it is a set instruction for itself. The motor that receives the set instruction sends a response, while the other motors do not send a set frame response (if the motor itself is in an automatic message state, the response will remain unchanged and sent normally). When many motors are in automatic message mode, CAN may be busy, so sending several set commands and receiving responses may be necessary to ensure successful setting.

## 4 Motor control related Commands

Data stored in the CAN data domain follow the storage method of high bit first and low bit last. For specific details and analysis of data splitting and storage, please refer to Appendix 2 or request routines from our team's technical team.

Data received by the driver is not subjected to amplitude limiting processing, and the relevant control is automatically limited by the system parameters. For example, in the current control mode, the current value given by CAN is 5000, indicating the actual control current of 50A. However, if the system has already set the maximum current to 30A, the actual motor control is given 30A. Other relevant control parameter ranges can be found in the parameter descriptions in the product manual.

For more detailed descriptions of control functions, please refer to the product description and function description in the relevant product manual.

#### 4.1 Automatic Message Communication Mode Control Command

- . Identifier: 0x1FF, 0x2FF, 0x204+MotorID
- . Master control sends data length: 8 bytes
- . Motor feedback data length: 8 bytes

Motor Receiving Message (Identifier: 0x1FF)							
Data [0]	Data [1]	Data [2]	Data [3]	Data [4]	Data [5]	Data [6]	Data [7]
Current setting value high 8 bits	Current setting value low 8 bits	Current setting value high 8 bits	Current setting value low 8 bits	Current setting value high 8 bits	Current setting value low 8 bits	Current setting value high 8 bits	Current setting value low 8 bits
MotorID: 0x01		MotorID: 0x02		MotorID: 0x03		MotorID: 0x04	

Motor Receiving Message (Identifier: 0x2FF)							
Data [0]	Data [1]	Data [2]	Data [3]	Data [4]	Data [5]	Data [6]	Data [7]
Current setting value high 8 bits	Current setting value low 8 bits	Current setting value high 8 bits	Current setting value low 8 bits	Current setting value high 8 bits	Current setting value low 8 bits	Current setting value high 8 bits	Current setting value low 8 bits
MotorID: 0x05		MotorID: 0x06		MotorID: 0x07		MotorID: 0x08	

Motor Automatic Return (Identifier: 0x204+MotorID)							
Output position high 8 bits	Output position low 8 bits	Output speed high by 8 bits	Output speed low 8 bits	Actual torque Current high 8 bits	Actual torque current low 8 bits	Motor temperature	error message

- . Given current value range: -32768~32767 corresponds to a current of -327.68A~327.67A.
- . Feedback output position value range: -18000~18000 corresponds to an angle of -180°~180°.
- . Feedback output speed value range: -32768~32767 corresponds to a speed of -3276.8rpm~3276.7rpm.
- . The actual torque current range of the feedback motor is -32768~32767, corresponding to a current of -327.68A~327.67A.

- Feedback motor temperature value: motor temperature value = actual temperature \* 2+50.

This communication protocol can control up to 4 motors in one data frame but only controls the current. The speed, position, and other parameters of the motors need to be calculated and controlled by the user on the main control. Below are examples to illustrate the effect of the control. For example, suppose there are currently two motors with motor ID distribution set to 1 and 2, and the motor has entered automatic message mode. In that case, the motor CAN interface will automatically send feedback messages to the host at 1000Hz. The message identifier for the motor with ID 1 is 0x205, and the message identifier for the motor with ID 2 is 0x206. For using the main control to control these motors, the CAN identifier is 0x1FF. Similarly, for the motor with ID 5 and 6, the identifier is 0x2FF.

## 4.2 Q&A Communication Mode Control Command

### 4.2.1 Force Position Hybrid Control Mode Command

Identifier: Motor ID Data Length: 8 bytes (64 bits in total)					
Motor Mode uint3	Parameter KP uint12	Parameter KD uint9	Expected position uint16	Expected speed uint12	Feedforward torque uint12
0x00	0-4095	0-511	0-65536	0-4095	0-4095

Sending the following command calls for force position hybrid control mode. The final output current is the motor current, which is calculated as shown below:

$$\begin{aligned}
 \text{motor\_current} = & \\
 & (KP * (\text{pos\_desired} - \text{pos\_current}) + \\
 & KD * (\text{spd\_desired} - \text{spd\_current}) + \text{Torque}) / K_T
 \end{aligned}$$

- Parameter KP value range: 0~4095 corresponds to 0.0f~500.0f.
- Parameter KD value range: 0~511 corresponds to 0.0f~5.0f.
- Expected output end position value range: 0~65536 corresponds to -12.5rad~12.5rad.
- Expected output end speed range: 0-4095 corresponds to -18.0rad/s~18.0rad/s.
- Motor feedforward torque value range: 0-4095 corresponds to -30.0Nm~30.0Nm.

When the driver receives this command, it can only return type 1 messages, and would not be able to choose to return other types of messages.

**Hint: The reference values for the no-load test experiment are as follows:  
 KP= 15.0f, KD=0.5f, expected speed: 0, expected current: 0**

### 4.2.2 Servo Position Control Mode Command

Identifier: Motor ID Data Length: 8 bytes (64 bits in total)				
Motor mode uint3	Expected position (pos) float32	Expected speed (spd) uint15	Current threshold (cur) uint12	Message return status (ack) uint2
0x01	Unit: degrees	0-32767	0-4095	0~3

- . Motor mode: 0x01, indicating that this control command is a servo position control command.
- . Explanation of the expected position value at the output end: The value is the actual angle, in degrees. For example, 120.5 represents 120.5 °.
- . Explanation of expected speed values at the output end: 0~32767 corresponds to 0~3276.7 rpm, with a ratio of 10.
- . Motor current threshold value range: 0~4095 corresponds to 0~409.5A, with a ratio of 10.
- . Message return status:
  - 0: No return
  - 1: Return message type 1
  - 2: Return message type 2
  - 3: Return message type 3

#### 4.2.3 Servo Speed Control Mode Command

Identifier: Motor ID Data Length: 7 bytes (56 bits in total)				
Motor mode uint3	Reserved control status uint3	Message return status uint2	Expected output end speed float32	Current threshold uint16
0x02	0	0~3		0-65536

- . Motor mode: 0x02, indicating that this control command is a servo speed control command.
- . Reservation control status: must be 0! If it is not 0, it can be controlled normally, but no matter how many events the return status is set, no message will be returned.
- . Message return status:
  - 0: No return
  - 1: Return message type 1
  - 2: Return message type 2
  - 3: Return message type 3
- . Explanation of the expected output end speed value: The CAN value corresponds one-to-one with the actual speed.
- . Explanation of motor current threshold value: 0~65536 corresponds to 0~6553.6A, with a ratio of 10.

#### 4.2.4 Current/Torque Control Mode and Braking Control Commands

Identifier: Motor ID Data Length: 3 bytes (24 bits in total)			
Motor mode uint3	Reserved control status uint3	Message return status uint2	Expected current/expected torque int16
0x03	0-7	0~3	-32768~32767

- . Reserved control status:



- 0: Normal current control
- 1: Torque control mode
- 2: Variable damping braking control mode, currently automatically ignoring the actual current data
- 3: Energy consumption braking control mode, currently automatically ignoring the expected current data
- 4: Regenerative braking control mode, currently the expected current value is the set braking current threshold
- 5: Reserved (currently invalid)
- 6: Reserved (currently invalid)
- 7: Reserved (currently invalid)

· Message return status:

- 0: No return
- 1: Return message type 1
- 2: Return message type 2
- 3: Return message type 3

- Expected current range: -32768~32767 corresponds to -327768~327.67A.
- Expected torque range: -32768~32767 corresponds to -327.68~327.67Nm.
- (Torque = current \* torque constant, the torque constant of this product is 1.4Nm/A)

#### 4.2.5 Reserve Control Command 1

Identifier: Motor ID Data Length: 8 bytes(64 bits in total)			
Motor mode uint3	Reserved control status uint3	Message return status uint2	Control value
0x04	0-7	0~3	

Currently invalid, reserved for traditional PID control in the future. If there is an urgent need, please contact our team.

#### 4.2.6 Reserve Control Command 2

Identifier: Motor ID Data Length: 8 bytes(64 bits in total)			
Motor mode uint3	Reserved control status uint3	Message return status uint2	Control value
0x05	0-7	0~3	

Currently invalid, reserved for ADRC or other adaptive algorithm control in the future. Please look forward to it!

### 4.3 Motor Control Parameter Configuration Instructions

configuration will fail when there are errors, such as motor overheat and overvoltage errors. As long as the operation safety is ensured, changes can be sent as frequently as other control commands, but users should pay attention to the control effect. Our team is not responsible for any losses caused by frequent modification of this parameter due to inexperienced operation. Therefore, it is recommended not to

modify parameters during motor operation or when the motor carries no load, and do not modify them frequently.

#### 4.3.1 Motor Acceleration Configuration Command

Identifier: Motor ID Data Length: 4 bytes(32 bits in total)				
Motor mode uint3	Reserved control status uint3	Message return status uint2	Configuration code uint8	Acceleration value uint16
0x06	0	0~3	0x01	Acceleration value

- Motor mode: 0x06, indicating that this control command is a motor parameter configuration command
- Reserved control status: must be set to 0, otherwise the driver will not respond, and the configuration will fail.
- Message return status:
  - 0: No return
  - 1: Return message type 4
  - 2: Failure to return and driver not responding, setting failed
  - 3: Failure to return and driver not responding, setting failed
- Configuration code:
  - 0x00: Reserved, currently invalid
  - 0x01: Indicates that the command is configuring system acceleration
- Acceleration value: 0-2000 corresponds to 0-20rad/s. Limits have been applied internally. As the acceleration decreases, the operation will become softer, however it is easy to overshoot if it is too small. The system's default value is 2000.

#### 4.3.2 Configuration Instructions for Motor Flux Observation Gain and Disturbance Compensation Coefficient

Identifier: Motor ID Data length: 6 bytes (48 bits in total)					
Motor mode uint3	Reserved control status uint3	Message return status uint2	Configuration code uint8	Parameter value uint16	Parameter value uint16
0x06	0	0 or 1	0x02	Magnetic flux observation gain value	Disturbance compensation coefficient value

- Motor mode: 0x06, indicating that this control command is a motor parameter configuration command
- Reserved control status: must be set to 0, otherwise the driver will not respond, and the configuration will fail.
- Message return status:
  - 0: No return
  - 1: Return message type 4
  - 2: Failure to return and driver not responding, setting failed

3: Failure to return and driver not responding, setting failed

- Configuration code:
  - 0x02: Indicates that the instruction is to configure the flux observation gain and disturbance compensation coefficient.
- Flux observation gain: 0-10000 corresponds to 0-1 in the VESC tool. The default value is 1000, corresponding to 0.1.
- Disturbance compensation coefficient: 0-10000 corresponds to 0-1 in the VESC tool. The default value is 60, corresponding to 0.006.

### 4.3.3 Motor Feedback Compensation Coefficient and Damping Coefficient (KD) Configuration Command

Identifier: Motor ID Data length: 6 bytes (48 bits in total)					
Motor mode uint3	Reserved control status uint3	Message return status uint2	Configuration code uint8	Parameter value uint16	Parameter value uint16
0x06	0	0 or 1	0x03	Motor feedback compensation coefficient	Damping system

- Motor mode: 0x06, indicating that this control command is a motor parameter configuration command
- Reserved control status: must be set to 0, otherwise the driver will not respond, and the configuration will fail.
- Message return status:
  - 0: No return
  - 1: Return message type 4
  - 2: No return, and driver no response, will result in configuration failure
  - 3: No return, and driver no response, will result in configuration failure
- Configuration code:
  - 0x02: Indicates that the instruction is to configure the flux observation gain and disturbance compensation coefficient.
- Motor feedback compensation coefficient: 0-10000 corresponds to 0-1 in the VESC tool. The default value is 60, corresponding to 0.006.
- Damping coefficient: 0-10000 corresponds to 0-1 in the VESC tool. The default value is 0.

### 4.4 Motor Control Parameter Query Command

Identifier: Motor ID Data length: 2-8 bytes		
Motor mode uint3	Retain data domain uint5	Query code uint8
0x07	0-31 Currently invalid	0: Reserved, currently invalid 1: Query current location 2: Query current speed 3: Query current current 4: Query current power 5: Query current acceleration 6: Query the current flux observation gain

		7: Query the current disturbance compensation coefficient 8: Query feedback compensation coefficient 9: Query damping coefficient
--	--	---

## 5 Q&A Mode Feedback Message

The return message includes the electronic control instruction, return message, configuration command range message, and the query instruction range message. The rules for storing data in the CAN data domain follow the storage method of high bit first and low bit last. For specific details and analysis of data splitting and storage, please refer to the appendix or request routines from our team's technical team.

- Message Lattice Identifier: Motor ID
- Data length: 2-8 bytes
- The format of the data field is as follows:

Byte0[5:7]	Byte0[0:4]	Byte1	Byte2~Byte7
Message type 0: Reserved 1: Control instruction return message 1 2: Control instruction return message 2 3: Control instruction return message 3	Motor error message 0: No errors 1: Motor overheating 2: Motor overcurrent 3: Motor voltage too low 4: Motor encoder error 6: Motor brake voltage too high 7: DRV drive error	Motor data	
4: Configuration instruction return message			
5: Query instruction return message		Query Code	Queried data

### 5.1 Q&A Communication Mode Return Message Type 1

Identifier: Motor ID Data length: 8 bytes (64 bits total)						
Message class uint3	Error message uint5	Motor position uint16	Motor speed uint12	Actual current uint12	Motor temperature uint8	MOS temperature uint8
0x01	0-7	Output position	Output speed	Motor phase current	0-255	0-255

- Message type: 0x01, indicating that this frame is a type 1 message.
- Motor error information: Refer to the error information table.
- Output end position value range: 0~65536 corresponds to - 12.5f~ 12.5f, in rad.
- Output end speed value range: 0-4095 corresponds to - 18.0f~ 18.0f, in rad/s.
- Actual current: 0-4095 corresponds to -30-30A.
- Motor temperature: The feedback numerical data type is unsigned 8-bit, and the value is equal to the actual temperature \* 2+50.

MOS temperature: The feedback numerical data type is unsigned 8-bit, and the value is equal to the actual temperature \* 2+50.

### 5.2 Q&A Communication Mode Return Message Type 2

Identifier: Motor ID Data length: 8 bytes (64 bits total)				
Message class uint3	Error message uint5	Motor position float32	Actual current int16	Motor temperature uint8
0x02	0-7	Output position	Motor phase current	0-255

- Message type: 0x02, indicating that this frame is a type 2 message.
- Motor error information: Refer to the error information table.
- Output position value range: The value refers to the actual angle, in radians. For example, 120.5 represents 120.5 °.
- The range of motor phase current values: -32768~32767 corresponds to -327768~327.67A.
- Motor temperature: The feedback numerical data type is unsigned 8-bit, and the value is equal to the actual temperature \* 2+50.

### 5.3 Q&A Communication Mode Return Message Type 3

Identifier: Motor ID Data length: 8 bytes (64 bits total)				
Message class uint3	Error message uint5	Motor speed float32	Actual current int16	Motor temperature uint8

- Message type: 0x03, indicating that this frame is a type 3 message.
- Motor error information: Refer to the error information table.
- Output end speed value range: The value refers to the actual speed, in rpm. For example, 100.6 represents 100.6 rpm.
- The range of motor phase current values: -32768~32767 corresponds to -327768~327.67A.
- Motor temperature: The feedback numerical data type is unsigned 8-bit, and the value is equal to the actual temperature \* 2+50.

### 5.4 Q&A Communication Mode Return Message Type 4

Identifier: Motor ID Data length: 3 bytes (24 bits in total)			
Message class uint3	Error message uint5	Configuration code uint8	Configuration status uint8
0x04	0-7	0-255	0/1

- Message type: 0x04, indicating that this frame is a type 4 message.
- Motor error information: Refer to the error information table.
- Configuration code: corresponds to the configuration code sent.
- Configuration status: 0 indicates failure, 1 indicates success.

## 5.5 Q&A Communication Mode Return Message Type 5

Identifier: Motor ID Data length: 4-6 bytes			
Message type uint3	Error message uint5	Query code uint8	Query returned data
0x05	0-31	1	Current position, float type (this message has a total of 6 bytes)
		2	Current speed, float type (this message has a total of 6 bytes)
		3	Current current, float type (this message has a total of 6 bytes)
		4	Current power, float type (this message has a total of 6 bytes)
		5	Current acceleration, uint16 type (this message has a total of 4 bytes)
		6	Current flux observation gain, uint16 type (this message has a total of 4 bytes)
		7	Current disturbance compensation coefficient, uint16 type (this message has a total of 4 bytes)
		8	Current feedback compensation gain, uint16 type (this message has a total of 4 bytes)
		9	Current damping coefficient, uint16 type (this message has a total of 4 bytes)