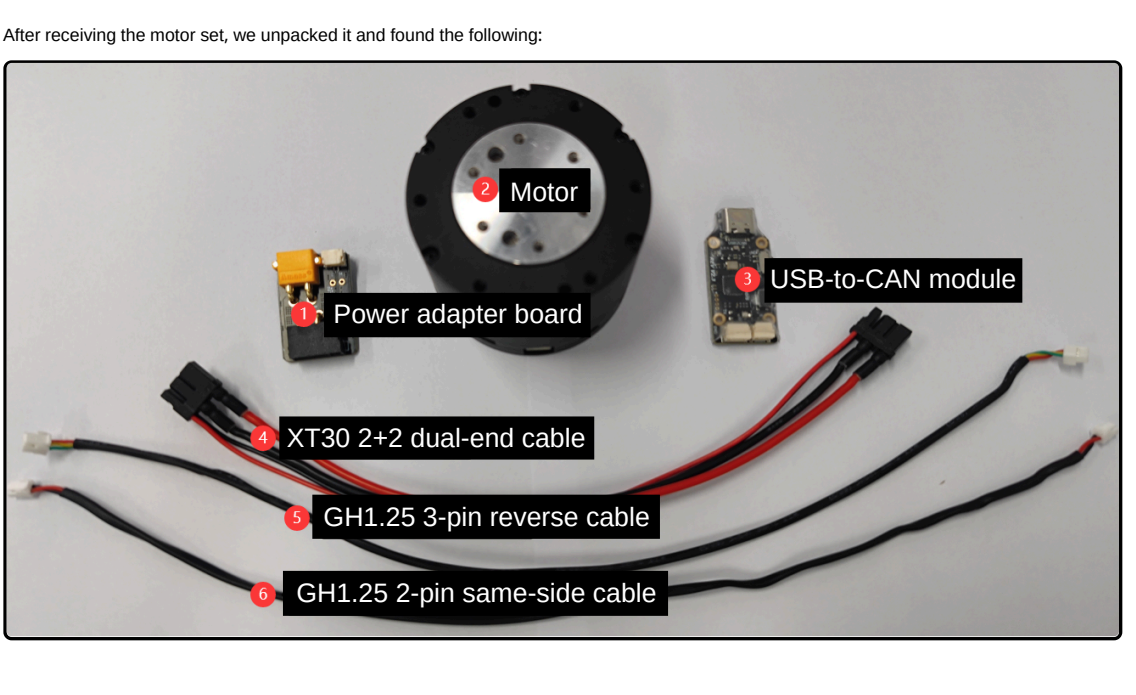
Motor Handling Process (I)

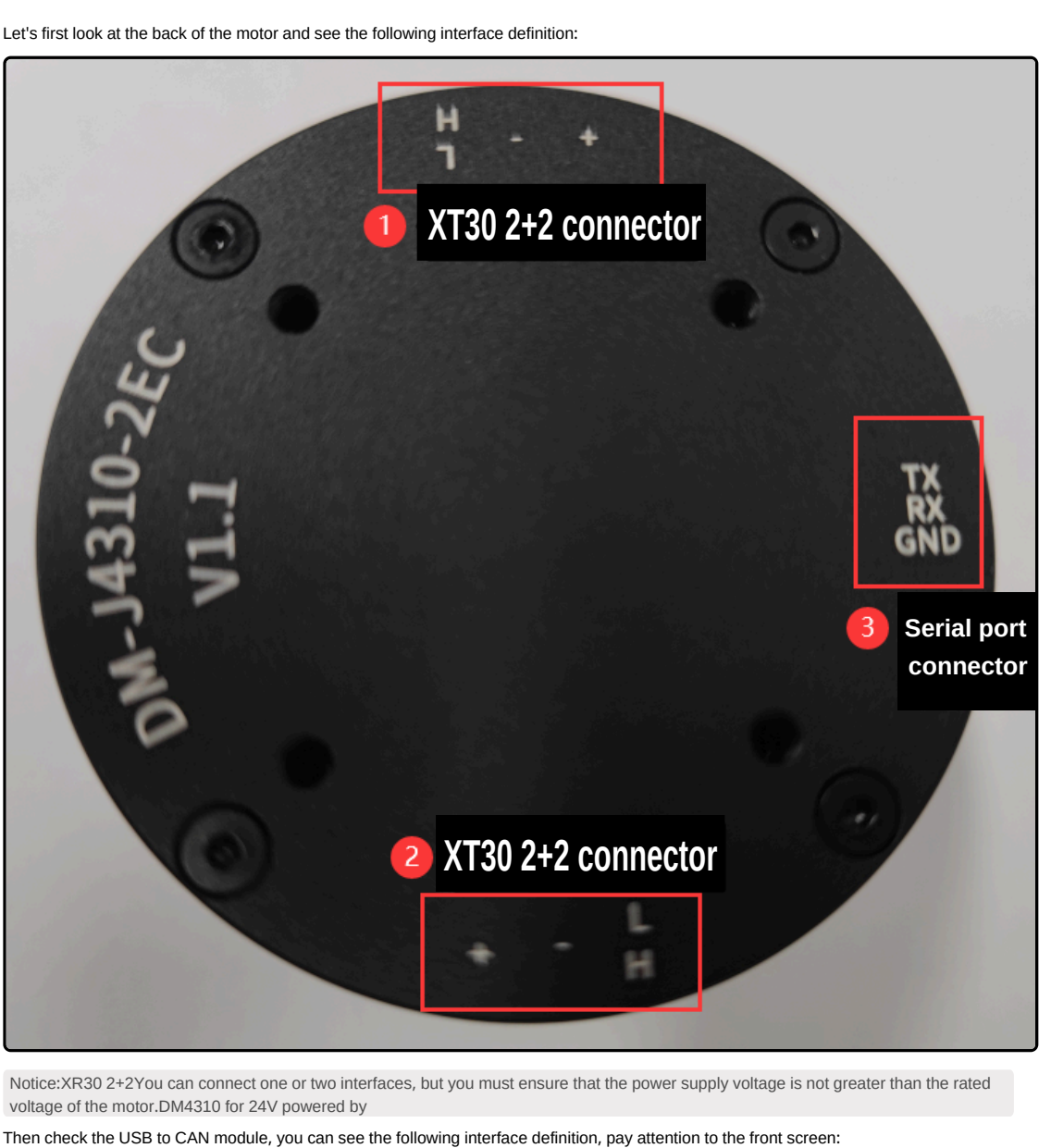
1. Motor set

After receiving the motor set, we unpacked it and found the following:


1 Power adapter board
2 Motor
3 USB-to-CAN module
4 XT30 2+2 dual-end cable
5 GH1.25 3-pin reverse cable
6 GH1.25 2-pin same-side cable

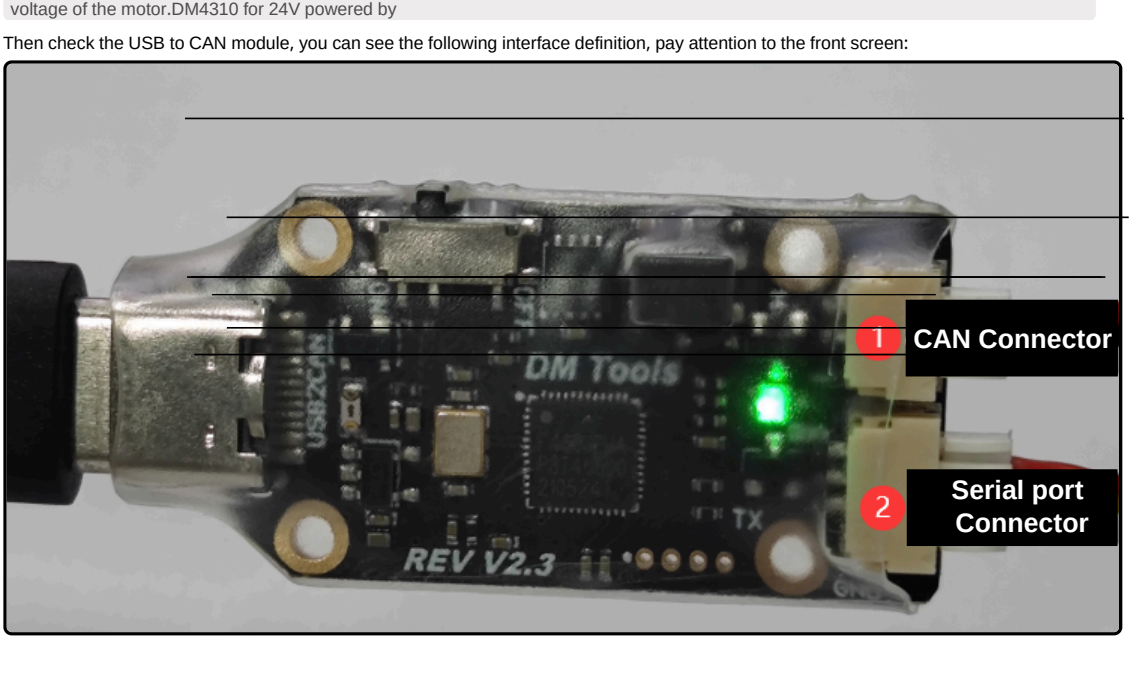
2. Definition of the interface between the motor and the USB-to-CAN module

Let's first look at the back of the motor and see the following interface definition:



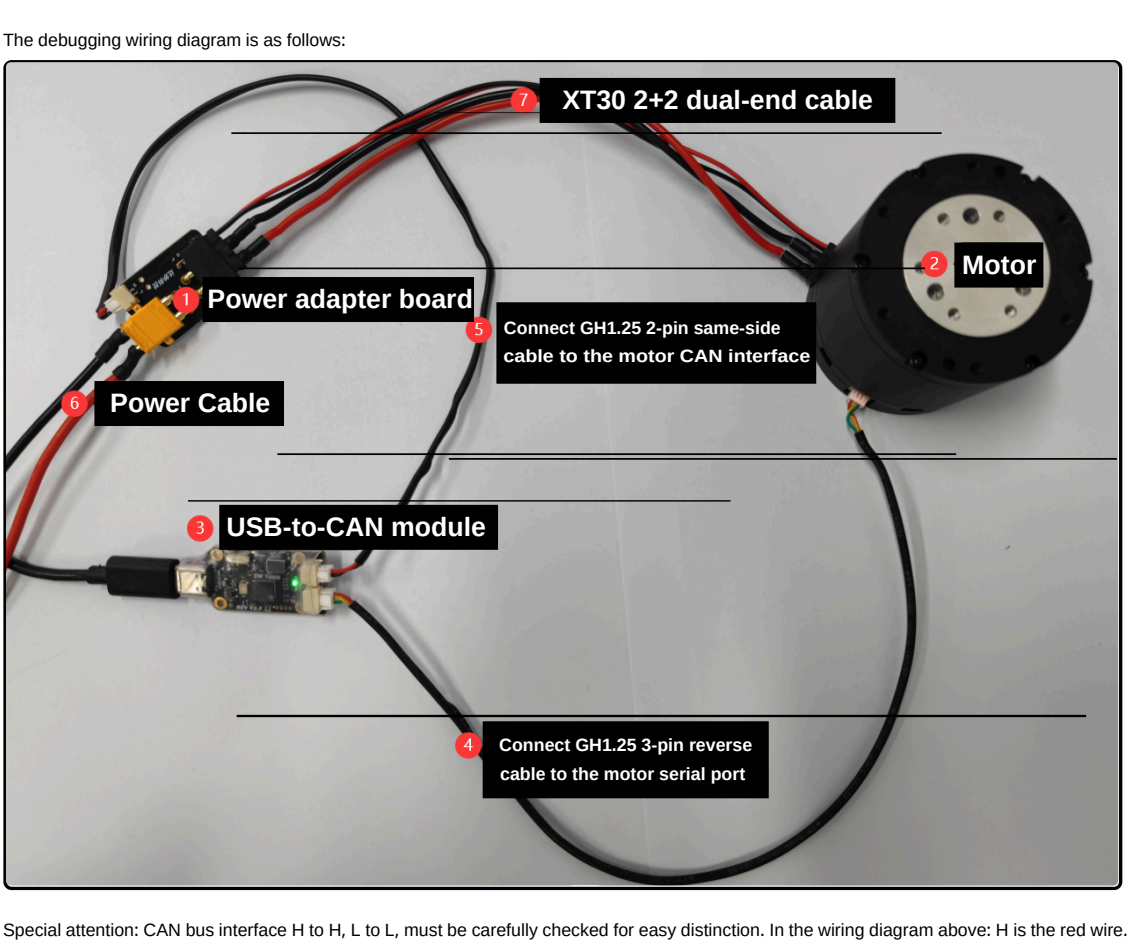
Notice:XR30 2+2You can connect one or two interfaces, but you must ensure that the power supply voltage is not greater than the rated voltage of the motor.DM4310 for 24V powered by

Then check the USB to CAN module, you can see the following interface definition, pay attention to the front screen:



3. Wiring Diagram

The debugging wiring diagram is as follows:


7 XT30 2+2 dual-end cable
2 Motor
1 Power adapter board
5 Connect GH1.25 2-pin same-side cable to the motor CAN interface
6 Power Cable
3 USB-to-CAN module
4 Connect GH1.25 3-pin reverse cable to the motor serial port

Special attention: CAN bus interface H to H, L to L, must be carefully checked for easy distinction. In the wiring diagram above: H is the red wire.

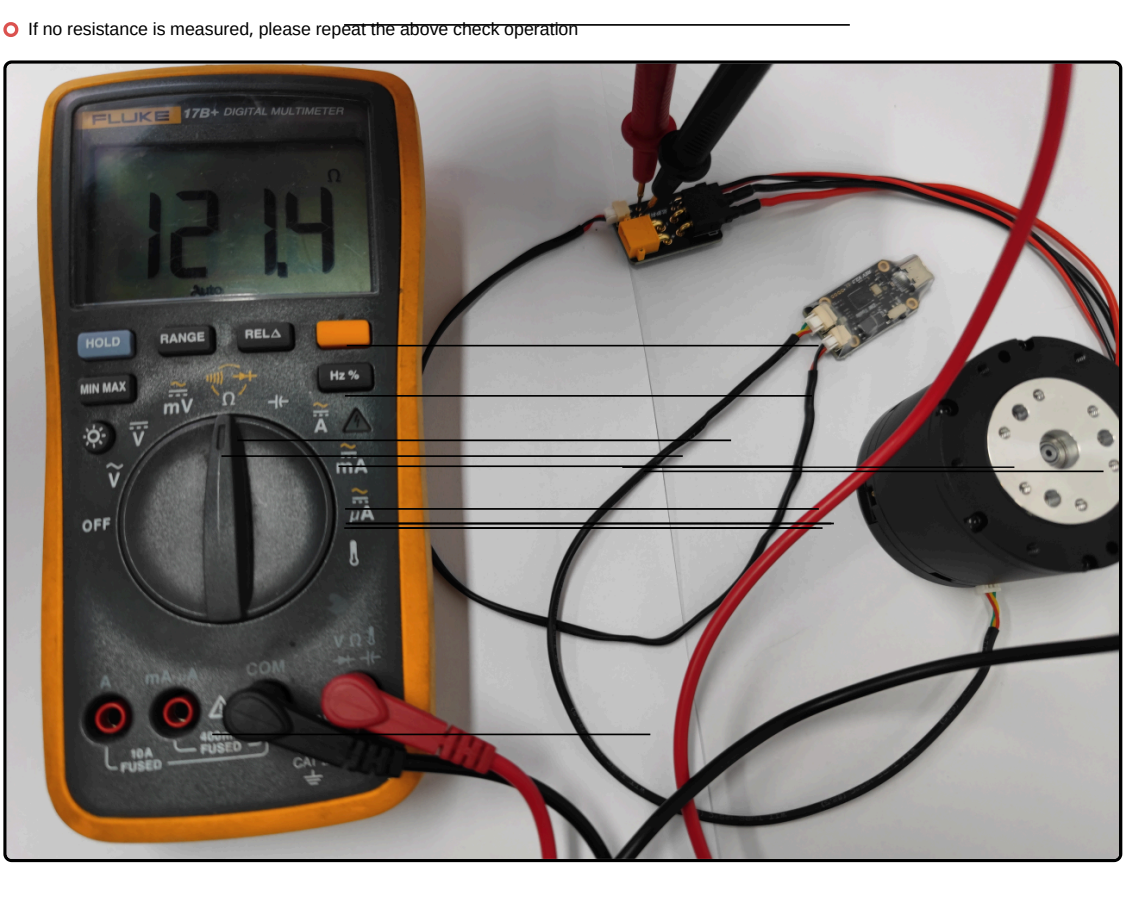
Special attention: The serial port interfaces TX to RX, RX to TX, must be carefully checked, and the serial port line also needs to be connected to the GND line.

4. CAN bus resistance measurement

To ensure communication quality, whether debugging this motor kit or debugging for secondary development, please be sure to add CAN bus resistors. At least one resistor is required, and it is best to have a 120R resistor on both the sending and receiving ends, that is, a 60R resistor on the bus.

The measurement method is as follows:

- Power off all the above modules.
- Make sure the wiring sequence is correct and the line is not broken.
- Use a multimeter to measure the H and L of the CAN interface
- If no resistance is measured, please repeat the above check operation

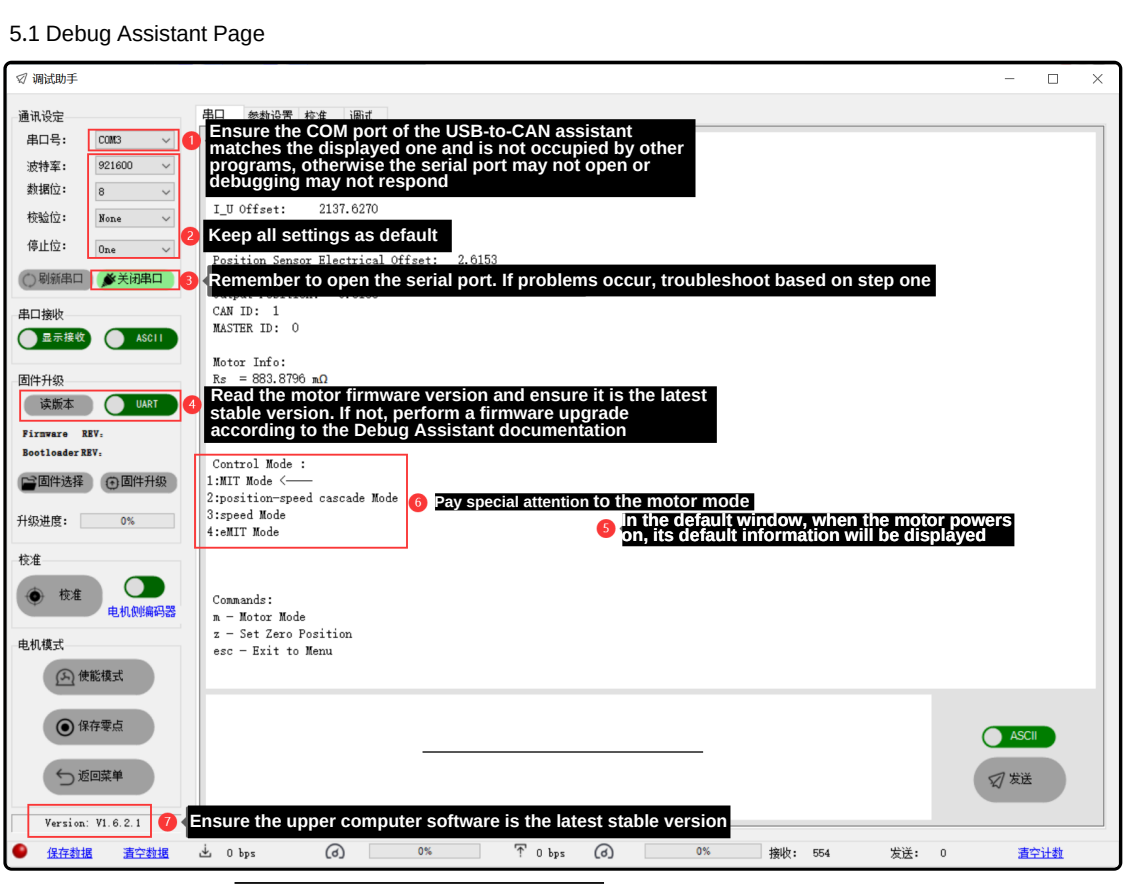


5. Start debugging with the debugging assistant

After all operations are performed and checked to ensure that there are no problems, you can start debugging

For detailed debugging process, please refer to: Debugging Assistant User Manual (Damiao Drive Control Protocol) V1.3.pdf

5.1 Debug Assistant Page



1 Ensure the COM port of the USB-to-CAN assistant matches the displayed one and is not occupied by other programs, otherwise the serial port may not open or debugging may not respond

2 Keep all settings as default

3 Remember to open the serial port. If problems occur, troubleshoot based on step one

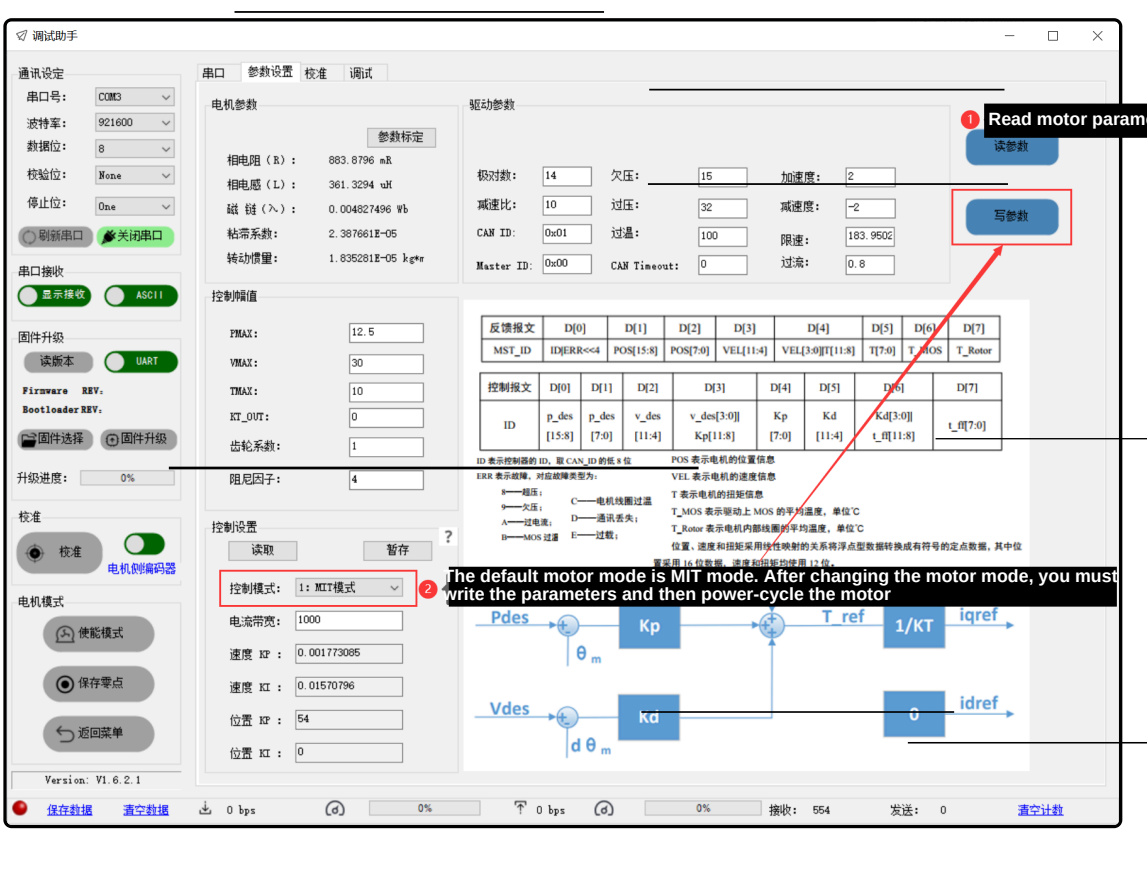
4 Read the motor firmware version and ensure it is the latest stable version. If not, perform a firmware upgrade according to the Debug Assistant documentation

```
Control Mode :
1:MIT Mode <----
2:position-speed cascade Mode
3:speed Mode
4:eMIT Mode
```

6 Pay special attention to the motor mode

5 In the default window, when the motor powers on, its default information will be displayed

7 Ensure the upper computer software is the latest stable version



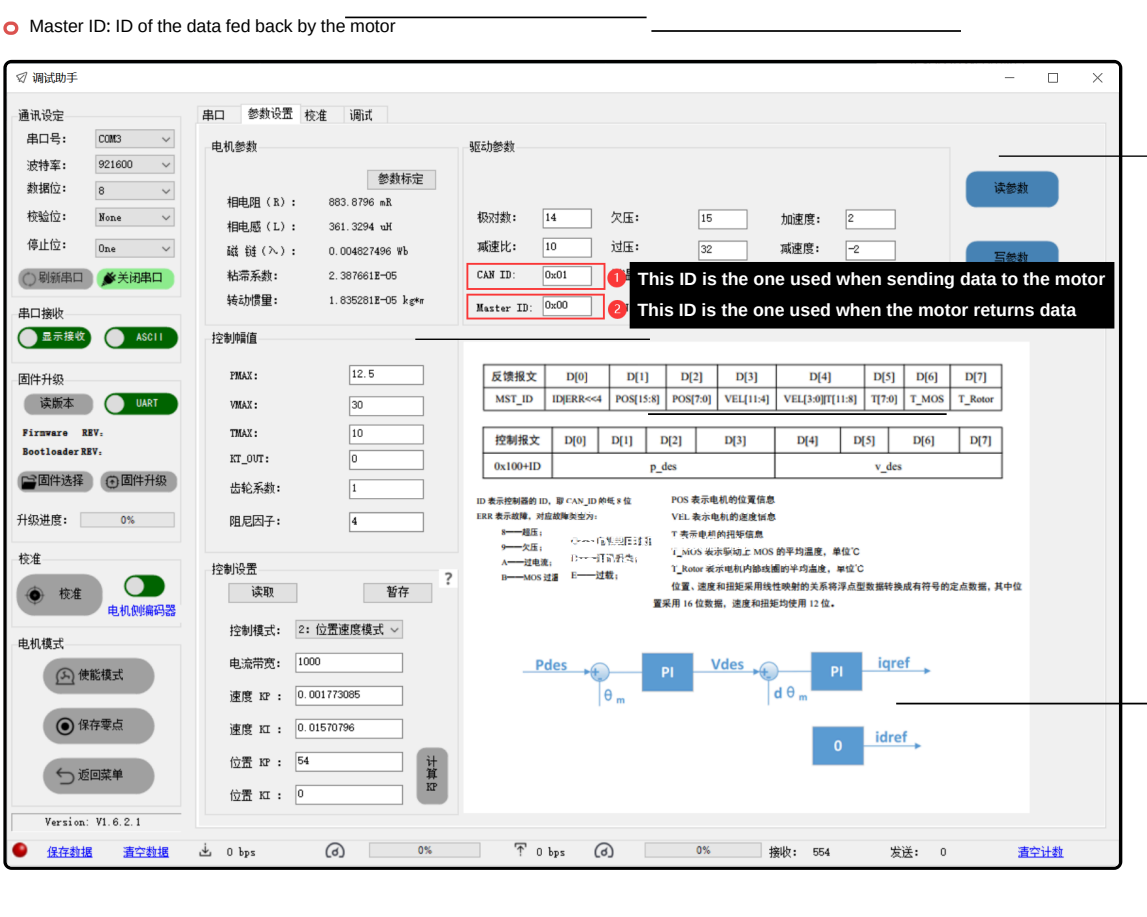
1 Read motor parameters

2 The default motor mode is MIT mode. After changing the motor mode, you must write the parameters and then power-cycle the motor

5.2 Differences between CAN ID and Master ID

When we need to modify these IDs, we need to understand what they mean:

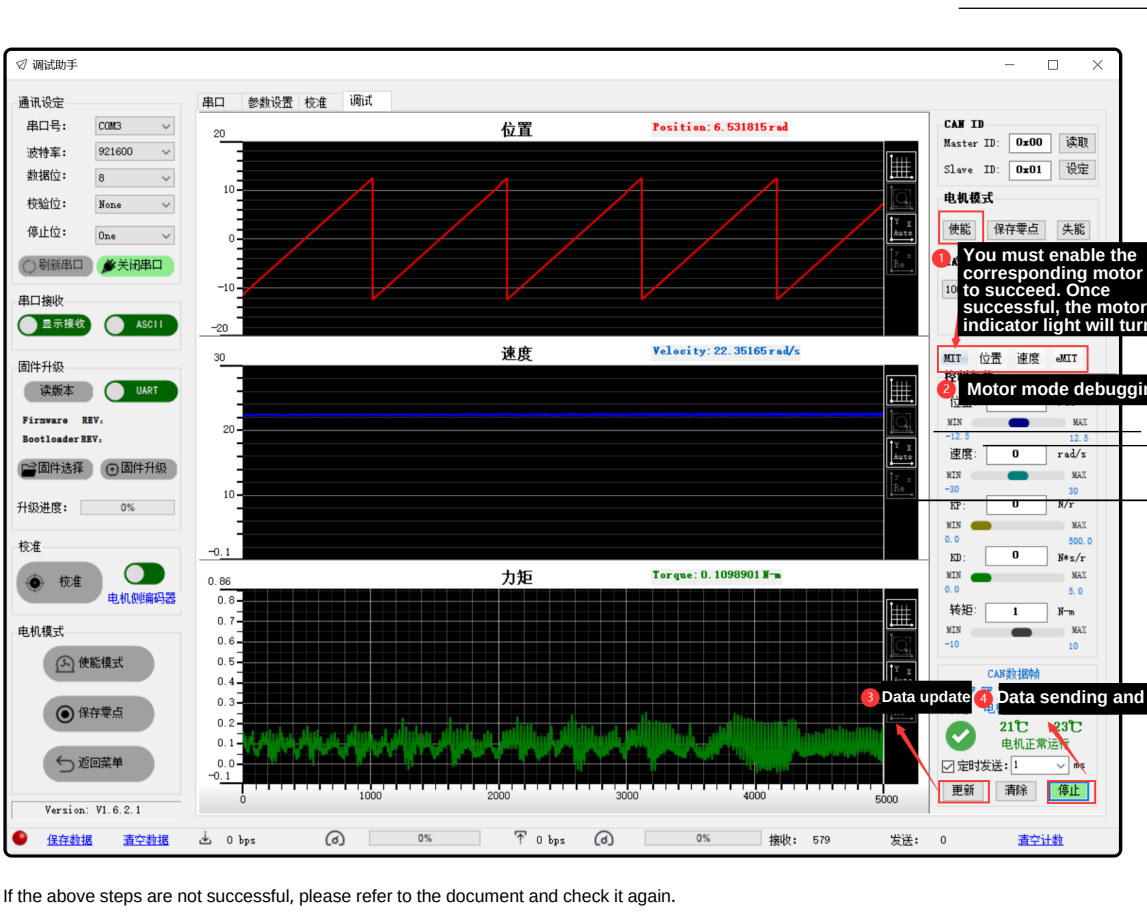
- CAN ID: ID for sending data to the motor
- Master ID: ID of the data fed back by the motor



1 This ID is the one used when sending data to the motor

2 This ID is the one used when the motor returns data

5.3 Motor Start



1 You must enable the corresponding motor mode to succeed. Once successful, the motor indicator light will turn green

2 Motor mode debugging

3 Data update

4 Data sending and stopping

If the above steps are not successful, please refer to the document and check it again.

Special attention: Wiring

6 Understanding and using communication protocols

When we are proficient in using the Damiao Technology debugging assistant, if we need to perform secondary development of the motor's motion control, we need to master some tools to troubleshoot the communication problems encountered.During the secondary development process, we will use some development boards to control multiple motors. Before starting to control multiple motors, please carefullyDebugging Assistant User Manual (Damiao Drive Control Protocol)Read it thoroughly. First,master the use of the debugging assistant and become familiar with the communication protocol of the Damiao motor. We also have related motor control routines for reference.
It is not recommended to start with the development board to control multiple motors. Let's first familiarize ourselves with the protocol andUSB to CAN Assistant Use.

6.1 Communication Protocol

For detailed communication protocol, please refer to: Debugging Assistant User Manual (Damiao Drive Control Protocol). Here we only provide a brief introduction and usage.

- MIT mode: the frame ID is the set CAN ID value
- Position speed mode: Frame ID is the set CAN ID value plus an offset of 0x100
- Speed mode: Frame ID is the set CAN ID value plus an offset of 0x200

Enable frame

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

DisabledFrame

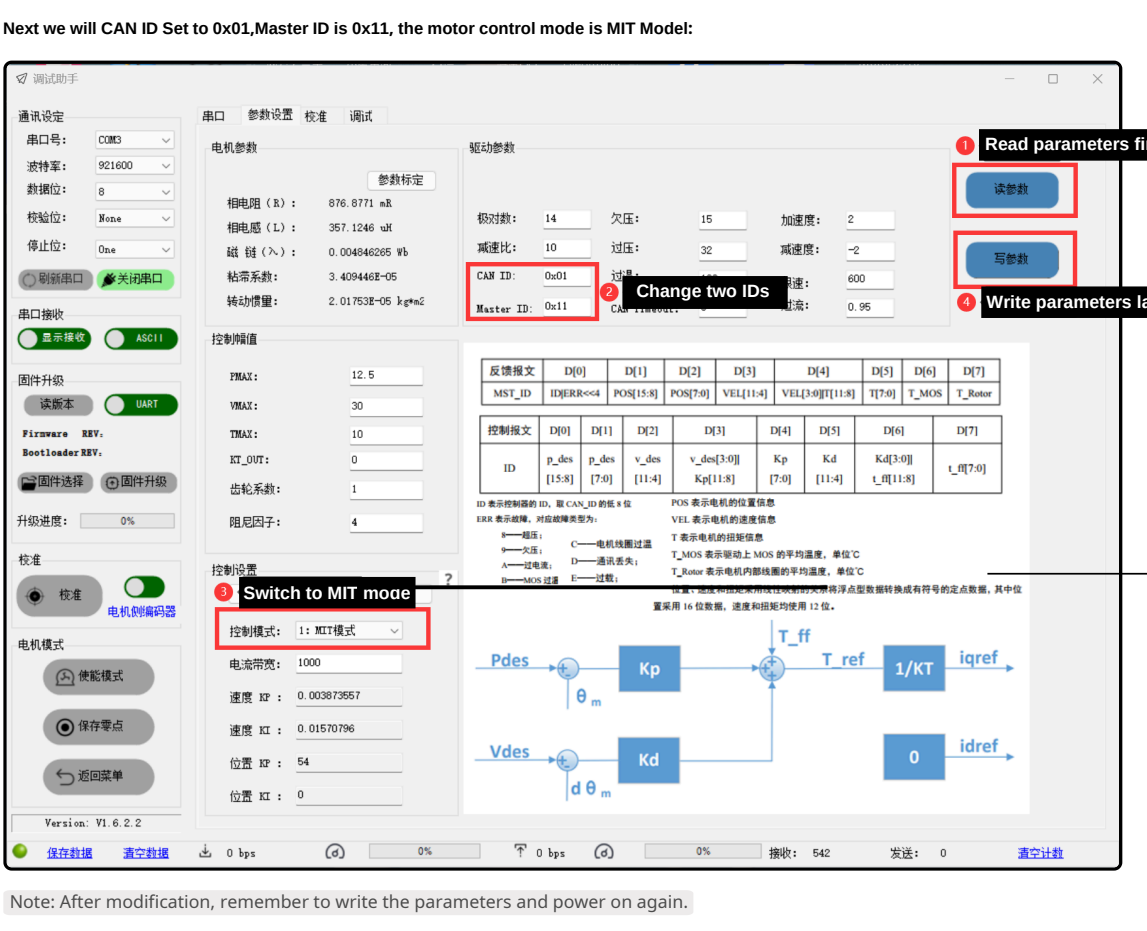
D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFD

For now, we will use these ~~protocol commands. For~~ more details, please refer to Routines and Debugging Assistant User Manual (Damiao Drive Control Protocol)

7. MIT Model Control

7.1 Set the motor mode and ID

Next we will CAN ID Set to 0x01,Master ID is 0x11, the motor control mode is MIT Model:



1 Read parameters first

2 Change two IDs

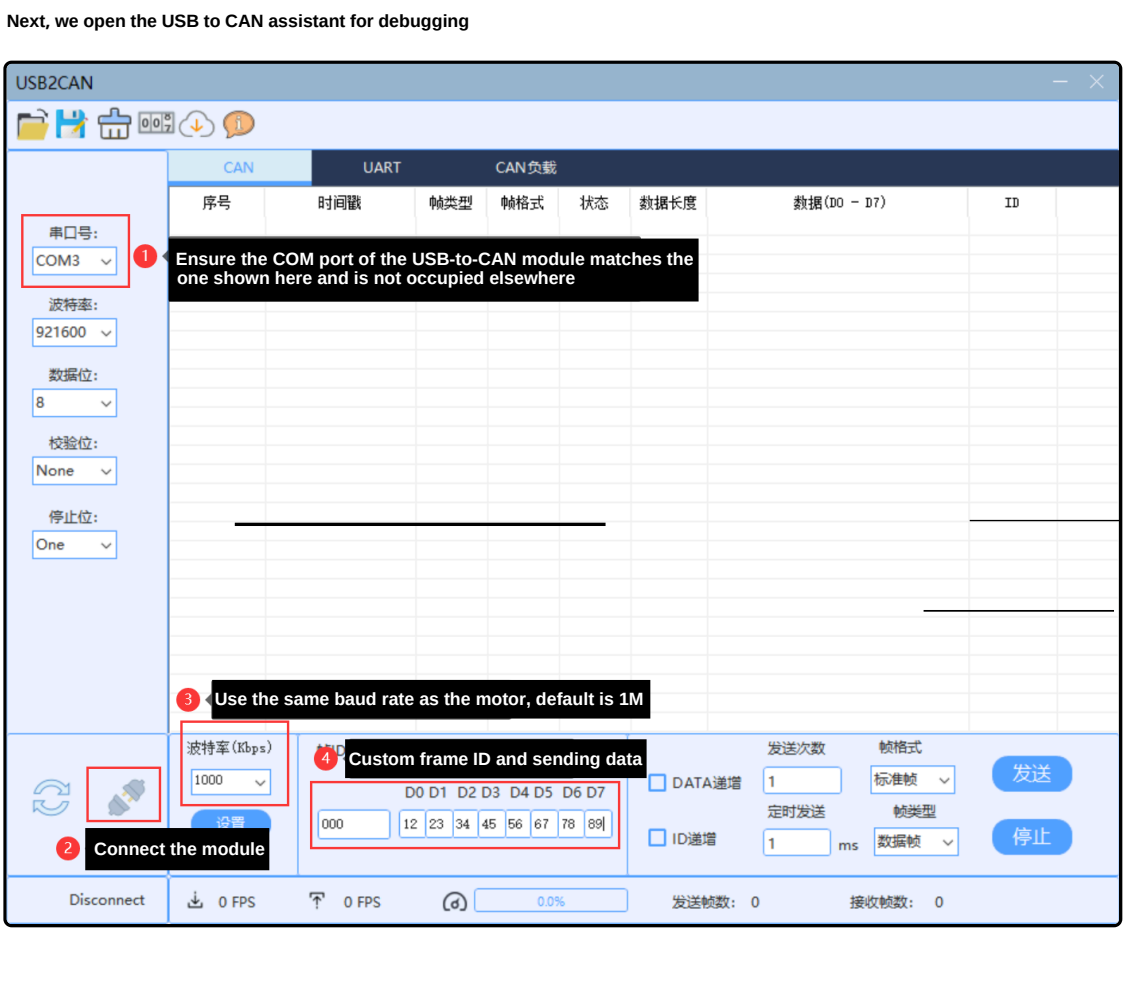
3 Switch to MIT mode

4 Write parameters last

Note: After modification, remember to write the parameters and power on again.

7.2 USB to CAN Assistant Connection

Next, we open the USB to CAN assistant for debugging



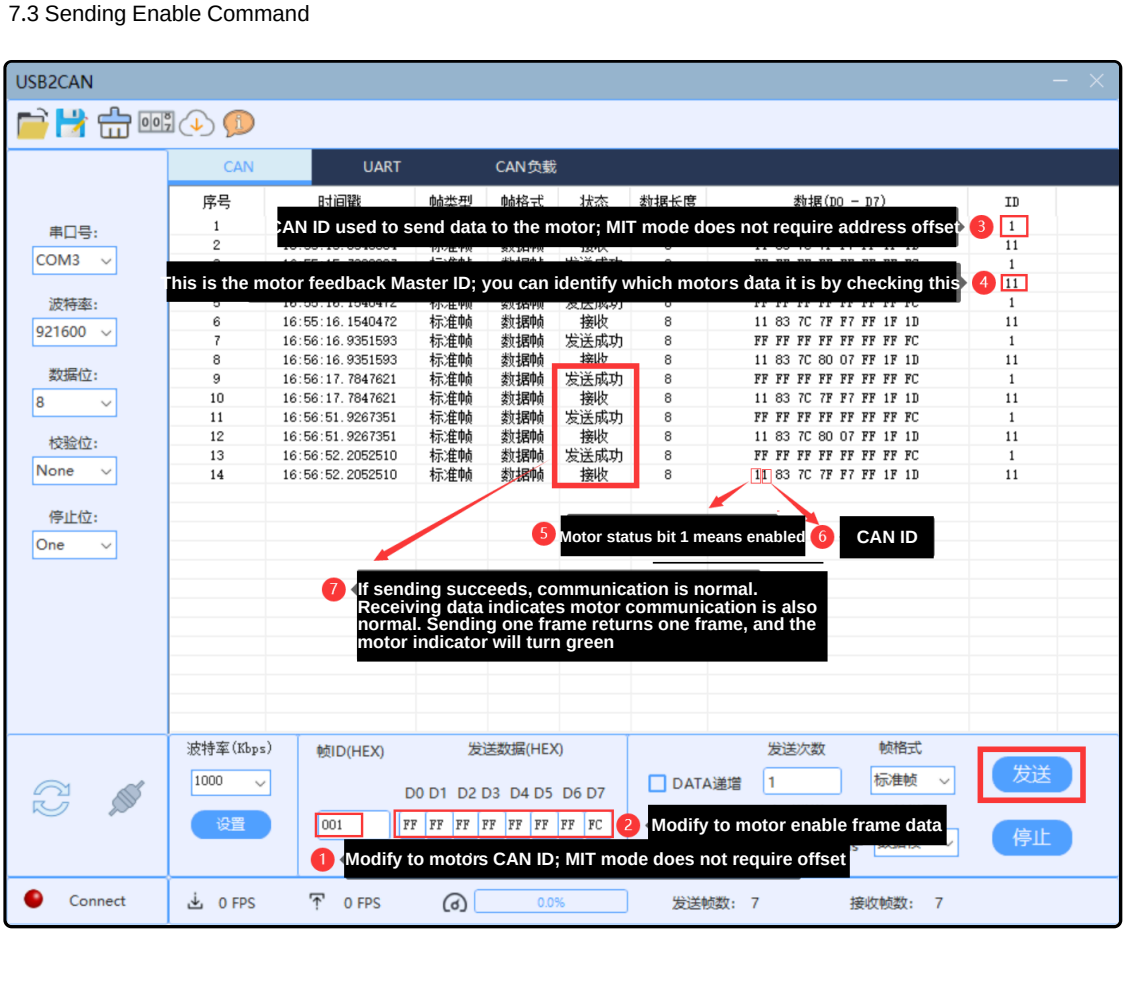
1 Ensure the COM port of the USB-to-CAN module matches the one shown here and is not occupied elsewhere

3 Use the same baud rate as the motor, default is 1M

4 Custom frame ID and sending data

2 Connect the module

7.3 Sending Enable Command



3 CAN ID used to send data to the motor; MIT mode does not require address offset

4 This is the motor feedback Master ID; you can identify which motors data it is by checking this

5 Motor status bit 1 means enabled

6 CAN ID

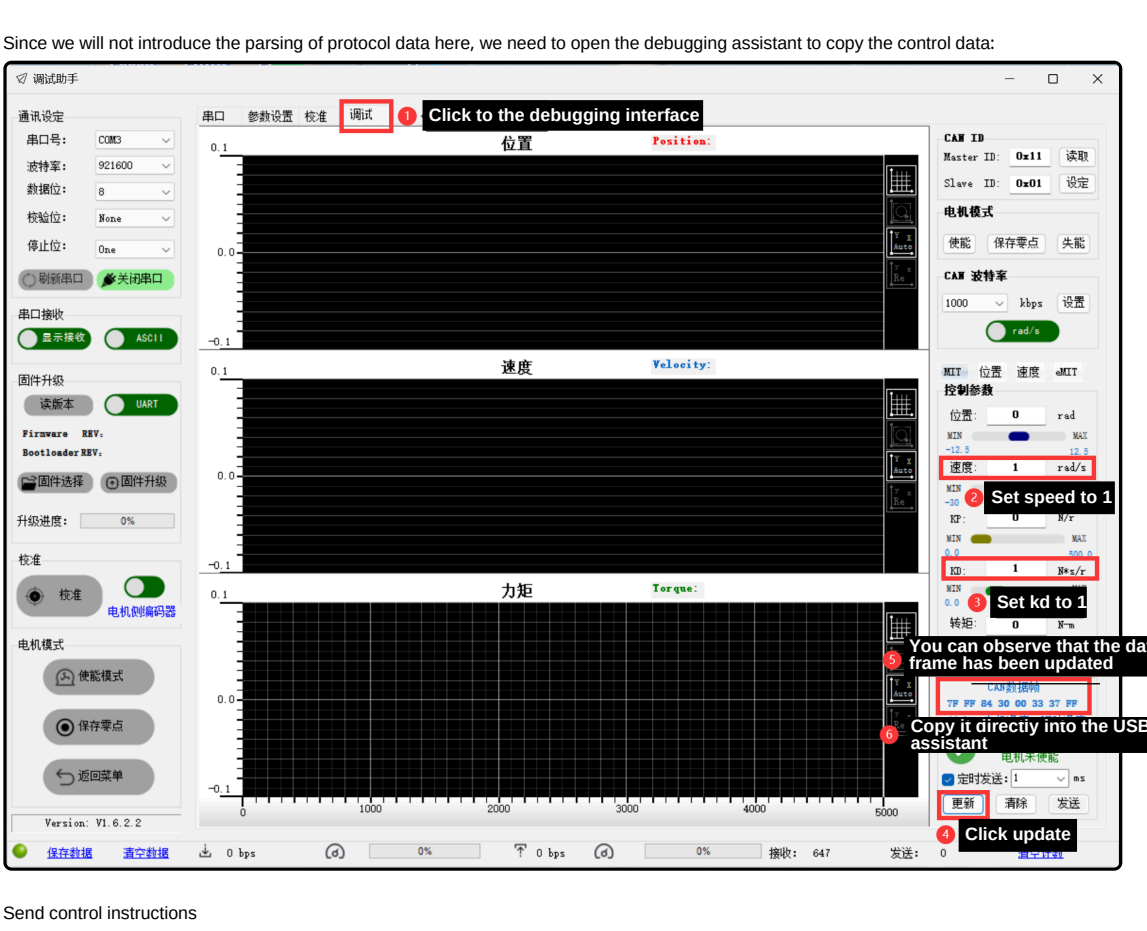
7 If sending succeeds, communication is normal. Receiving data indicates motor communication is also normal. Sending one frame returns one frame, and the motor indicator will turn green

2 Modify to motor enable frame data

1 Modify to motors CAN ID; MIT mode does not require offset

7.4 Sending Motor Control Instructions

Since we will not introduce the parsing of protocol data here, we need to open the debugging assistant to copy the control data:



1 Click to the debugging interface

2 Set speed to 1

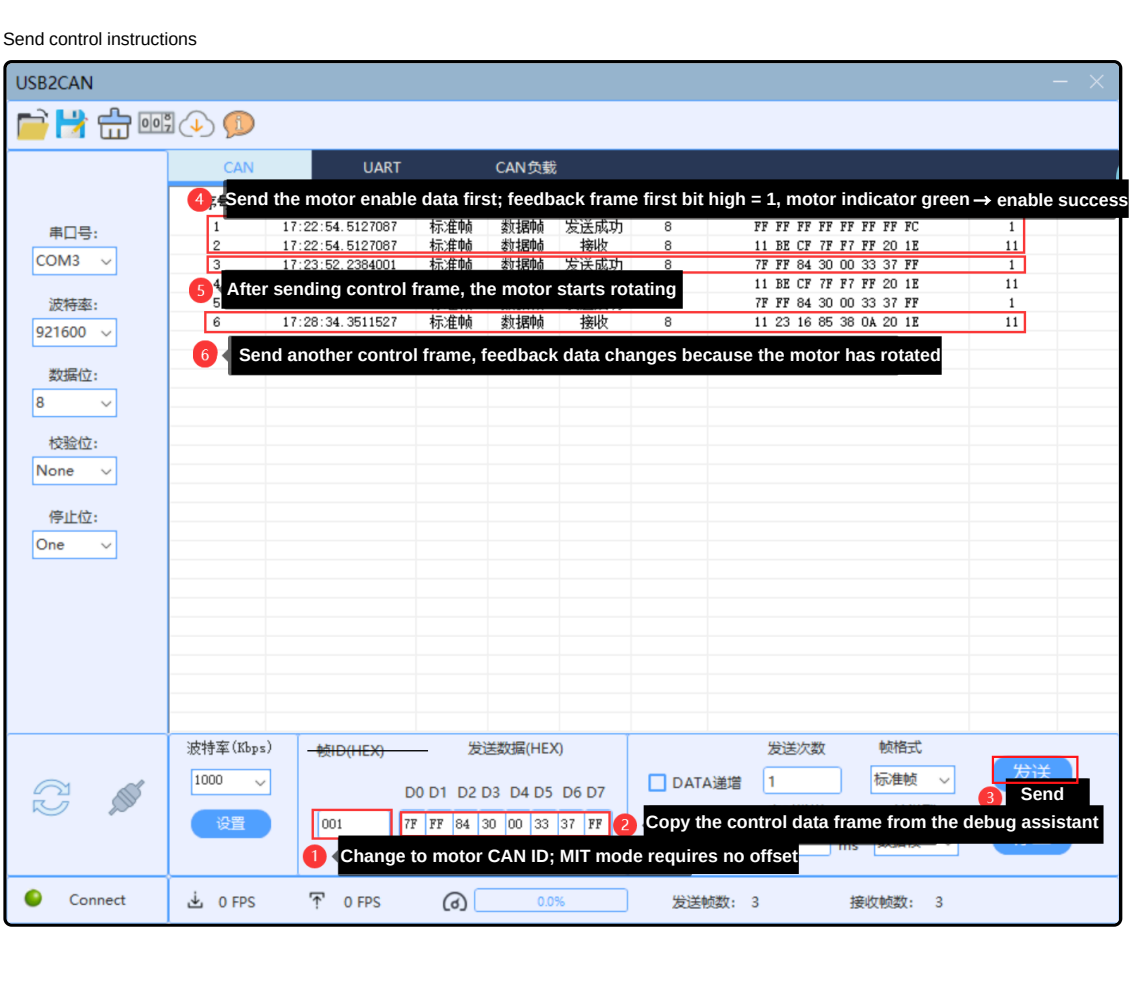
3 Set kd to 1

5 You can observe that the data frame has been updated

6 Copy it directly into the USB2CAN assistant

4 Click update

Send control instructions



4 Send the motor enable data first; feedback frame first bit high = 1, motor indicator green → enable success

5 After sending control frame, the motor starts rotating

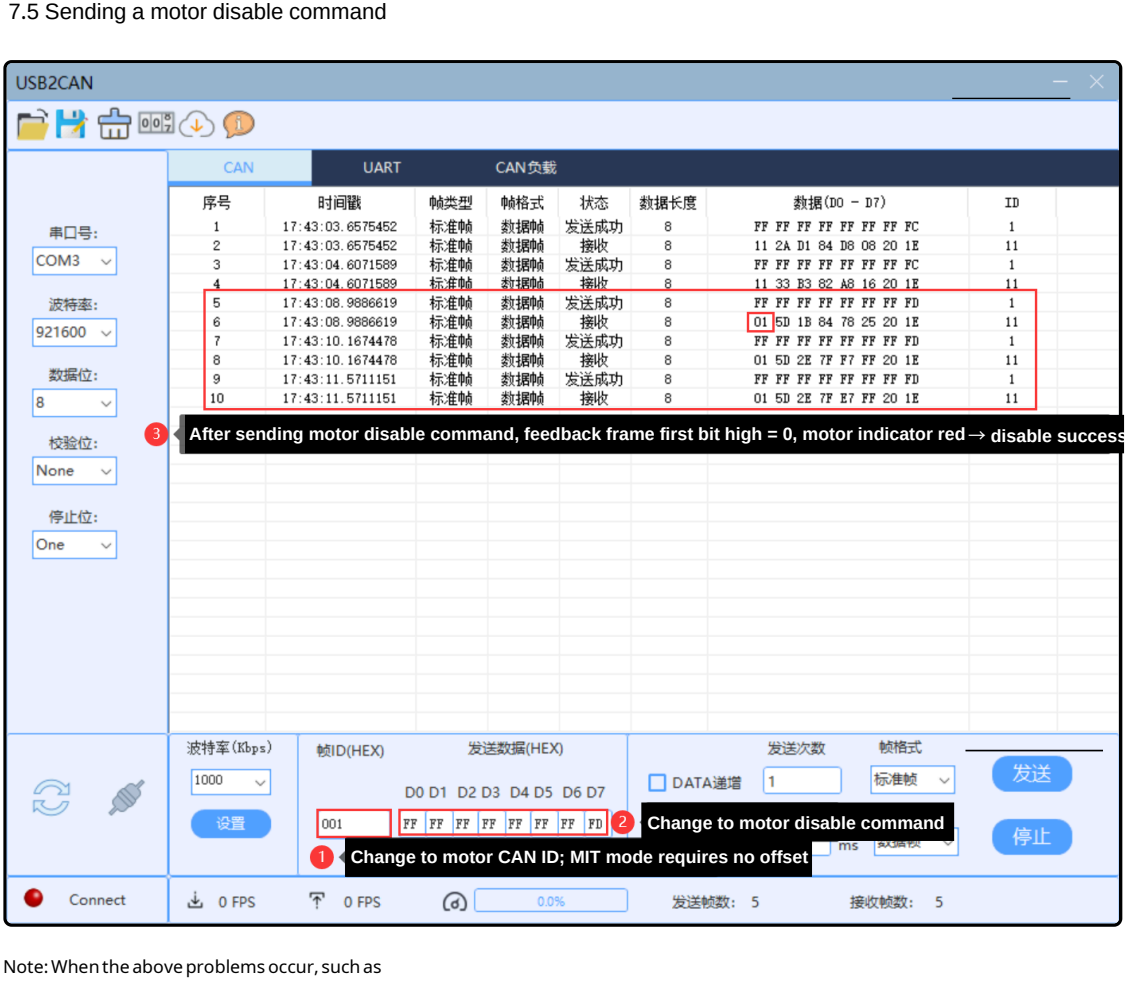
6 Send another control frame, feedback data changes because the motor has rotated

2 Send

3 Copy the control data frame from the debug assistant

1 Change to motor CAN ID; MIT mode requires no offset

7.5 Sending a motor disable command



3 After sending motor disable command, feedback frame first bit high = 0, motor indicator red → disable success

2 Change to motor disable command

1 Change to motor CAN ID; MIT mode requires no offset

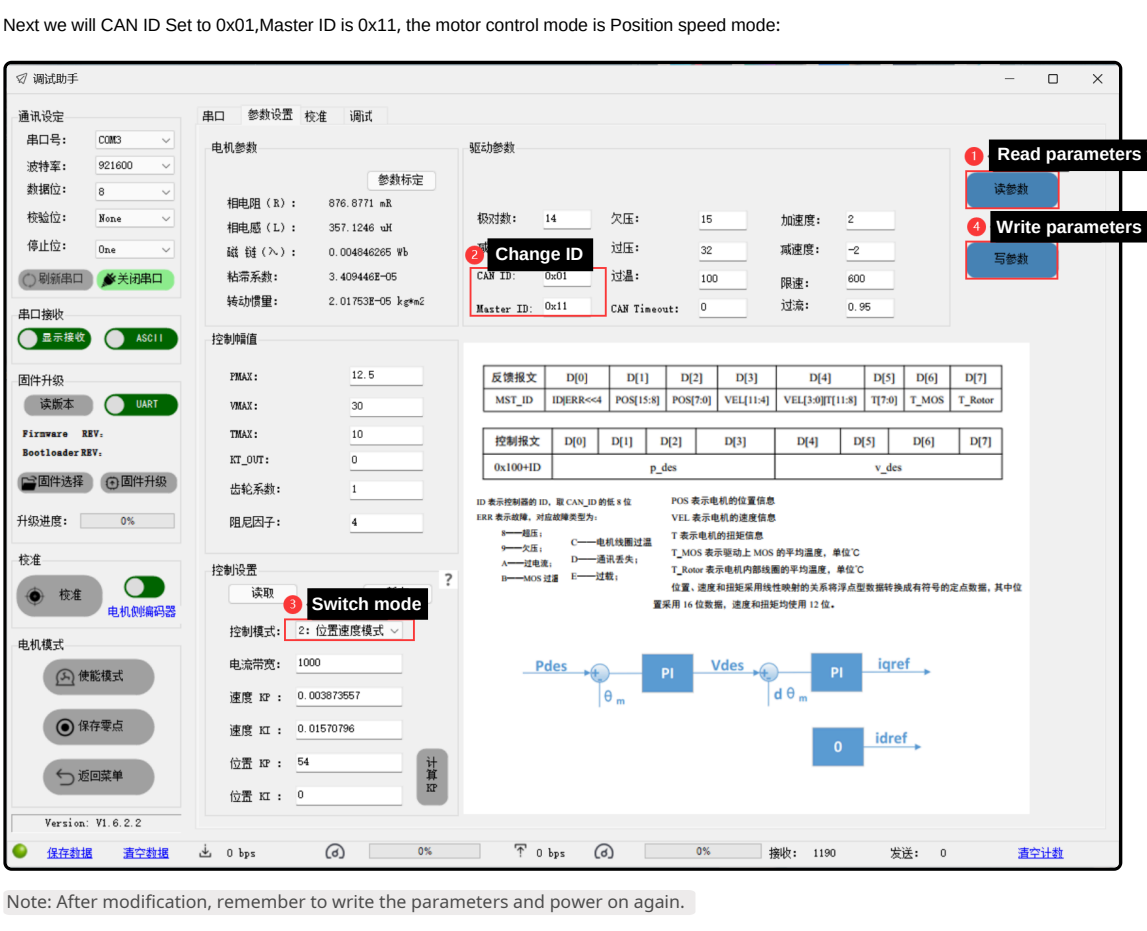
Note: When the above problems occur, such as

- Sending failure: You need to check whether the line sequence is correct, whether the CAN baud rate is set incorrectly, or whether the COM port is wrong.
- The message is sent successfully but the motor does not return any information and the red light does not turn green: Check whether the CAN ID and motor control mode are set correctly.
- Send disable but the motor does not return information and the green light does not turn red: Check whether the CAN ID and motor control mode are set correctly.

8. Position and speed mode control

8.1 Setting the motor mode and ID

Next we will CAN ID Set to 0x01,Master ID is 0x11, the motor control mode is Position speed mode:



1 Read parameters first

2 Change ID

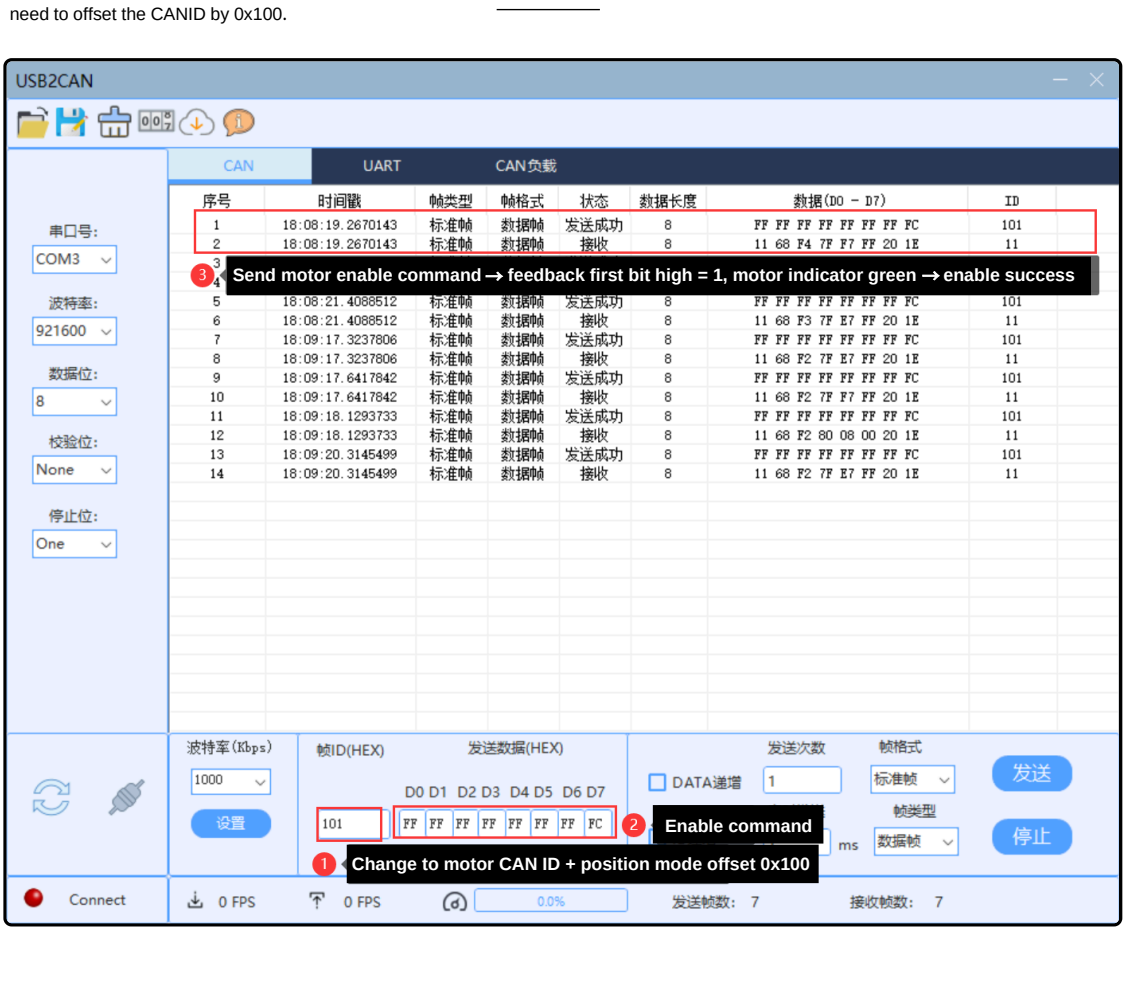
3 Switch mode

4 Write parameters last

Note: After modification, remember to write the parameters and power on again.

8.2 Sending Motor Enable Command

Since we already know the enable and disable frame commands, we can just refer to the MIT mode process to send them. The only difference is that we need to offset the CANID by 0x100.



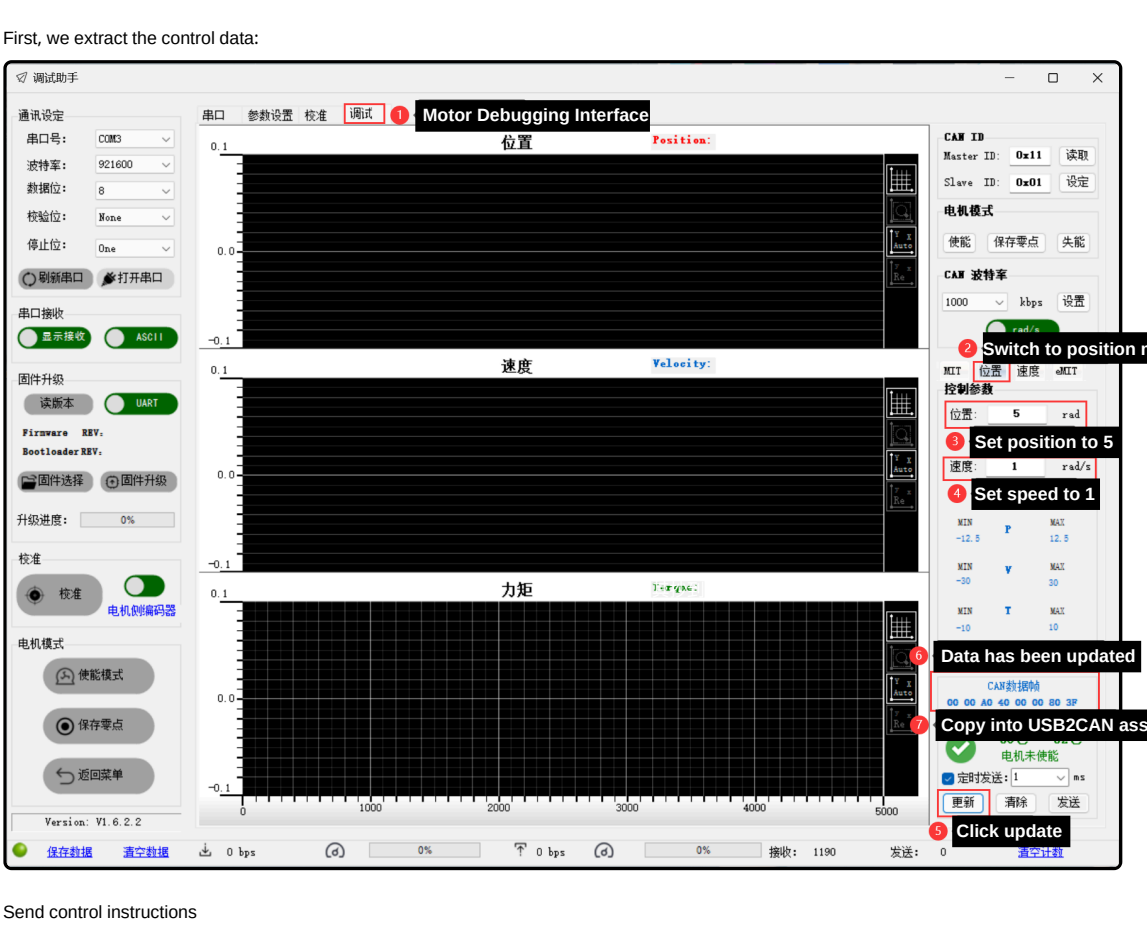
3 Send motor enable command → feedback first bit high = 1, motor indicator green → enable success

2 Enable command

1 Change to motor CAN ID + position mode offset 0x100

8.3 Sending Motor Control Instructions

First, we extract the control data:



1 Motor Debugging Interface

2 Switch to position mode

3 Set position to 5

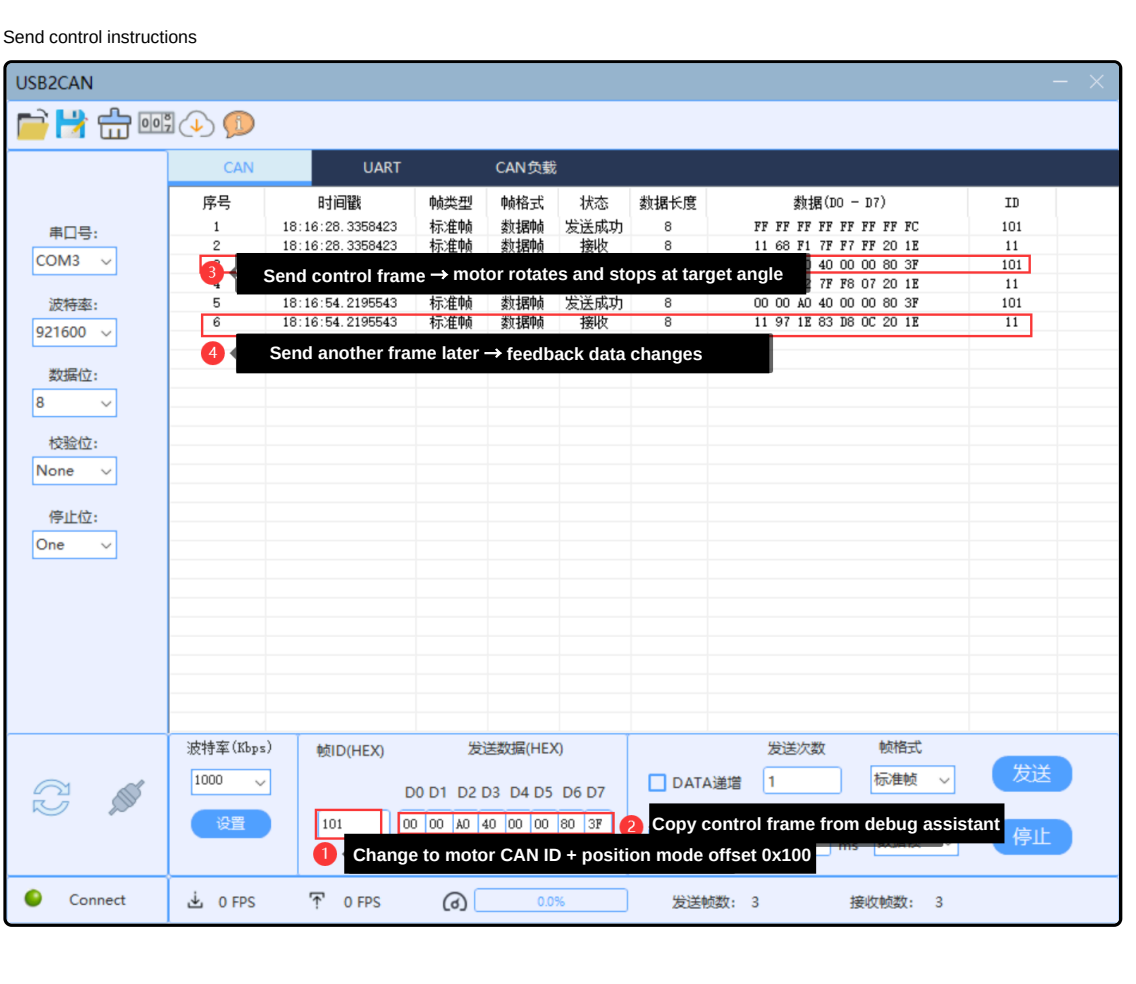
4 Set speed to 1

6 Data has been updated

7 Copy into USB2CAN assistant

5 Click update

Send control instructions



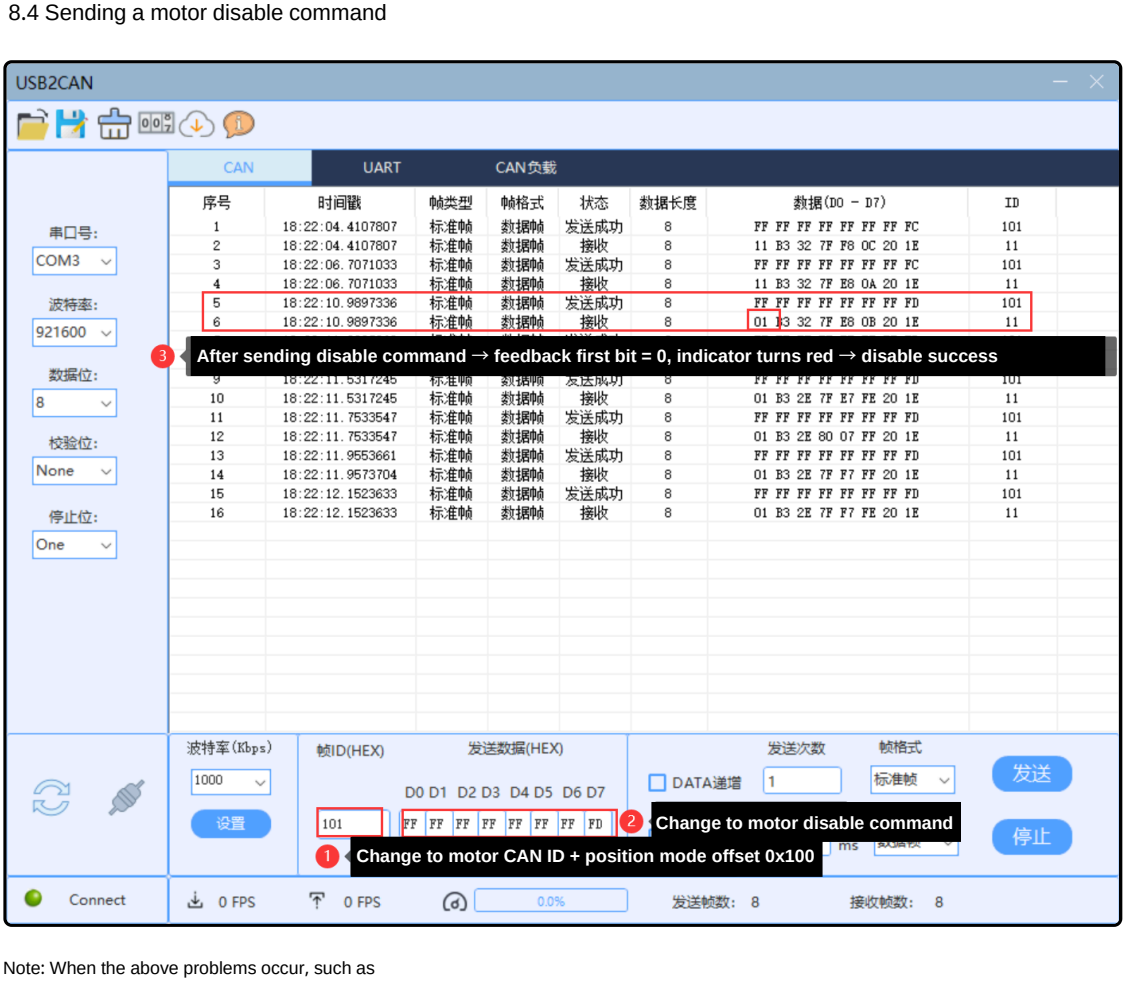
3 Send control frame → motor rotates and stops at target angle

4 Send another frame later → feedback data changes

2 Copy control frame from debug assistant

1 Change to motor CAN ID + position mode offset 0x100

8.4 Sending a motor disable command



3 After sending disable command → feedback first bit = 0, indicator turns red → disable success

2 Change to motor disable command

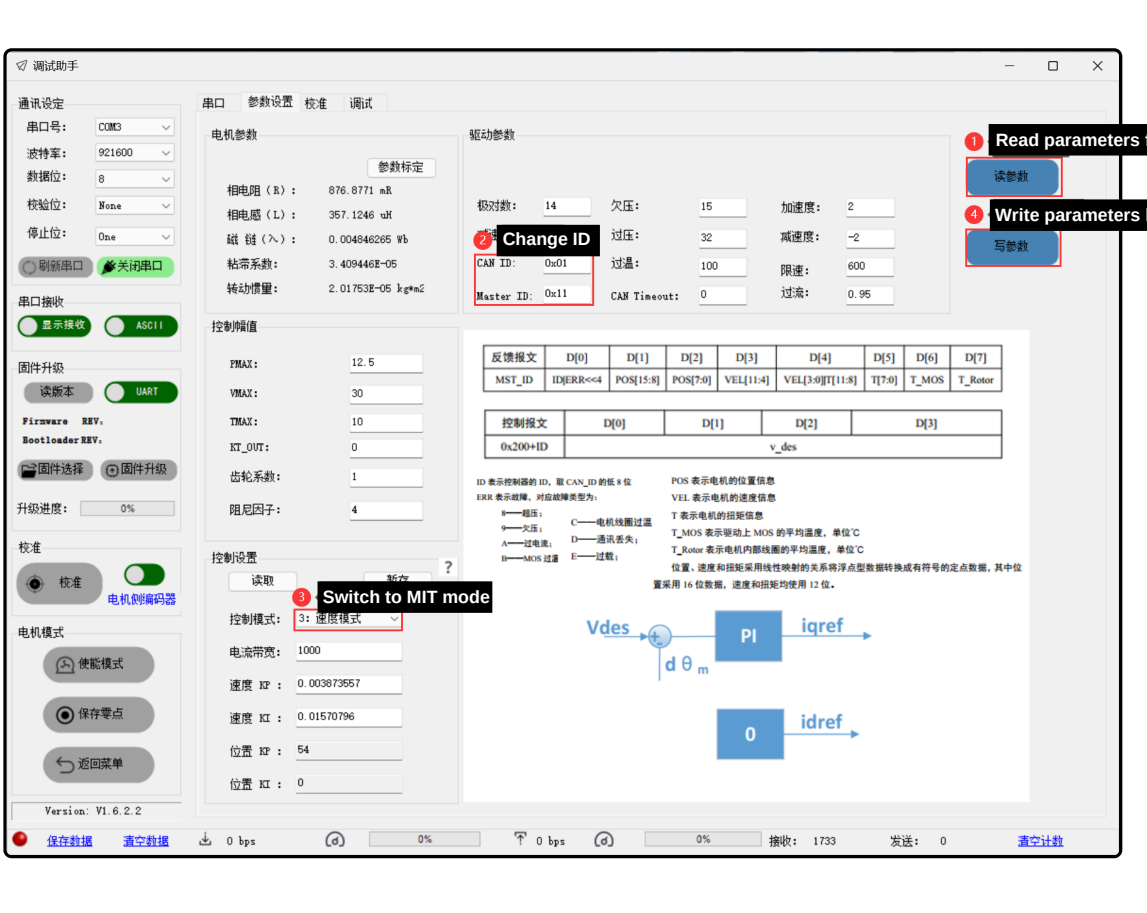
1 Change to motor CAN ID + position mode offset 0x100

Note: When the above problems occur, such as

- Sending failure: You need to check whether the line sequence is correct, whether the CAN baud rate is set incorrectly, or whether the COM port is wrong.
- The message is sent successfully but the motor does not return any information and the red light does not turn green: Check whether the CAN ID and motor control mode are set correctly.
- Send disable but the motor does not return information and the green light does not turn red: Check whether the CAN ID and motor control mode are set correctly.

9. Speed mode control

9.1 Setting the motor mode and ID



1 Read parameters first

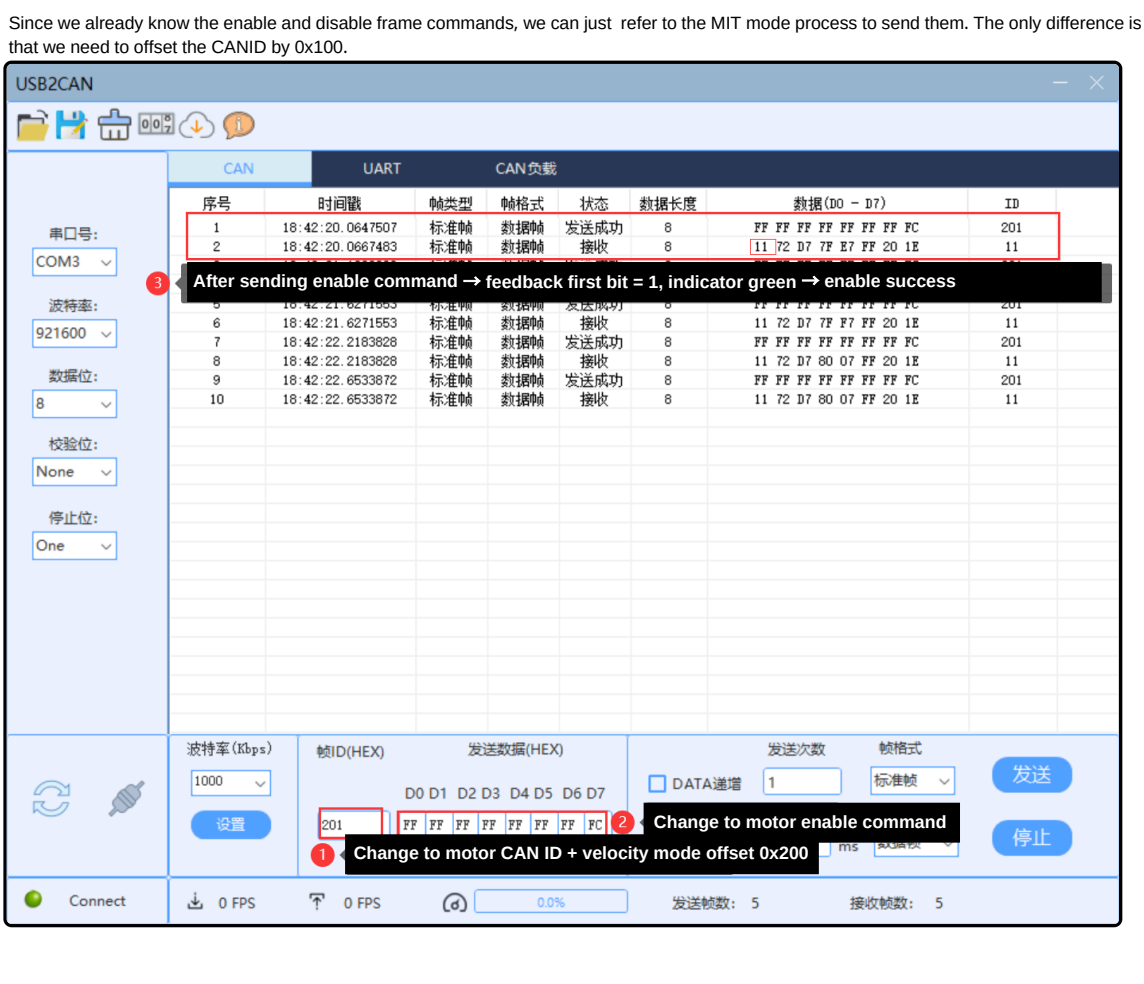
2 Change ID

3 Switch to MIT mode

4 Write parameters last

9.2 Sending Motor Enable Command

Since we already know the enable and disable frame commands, we can just refer to the MIT mode process to send them. The only difference is that we need to offset the CANID by 0x100.



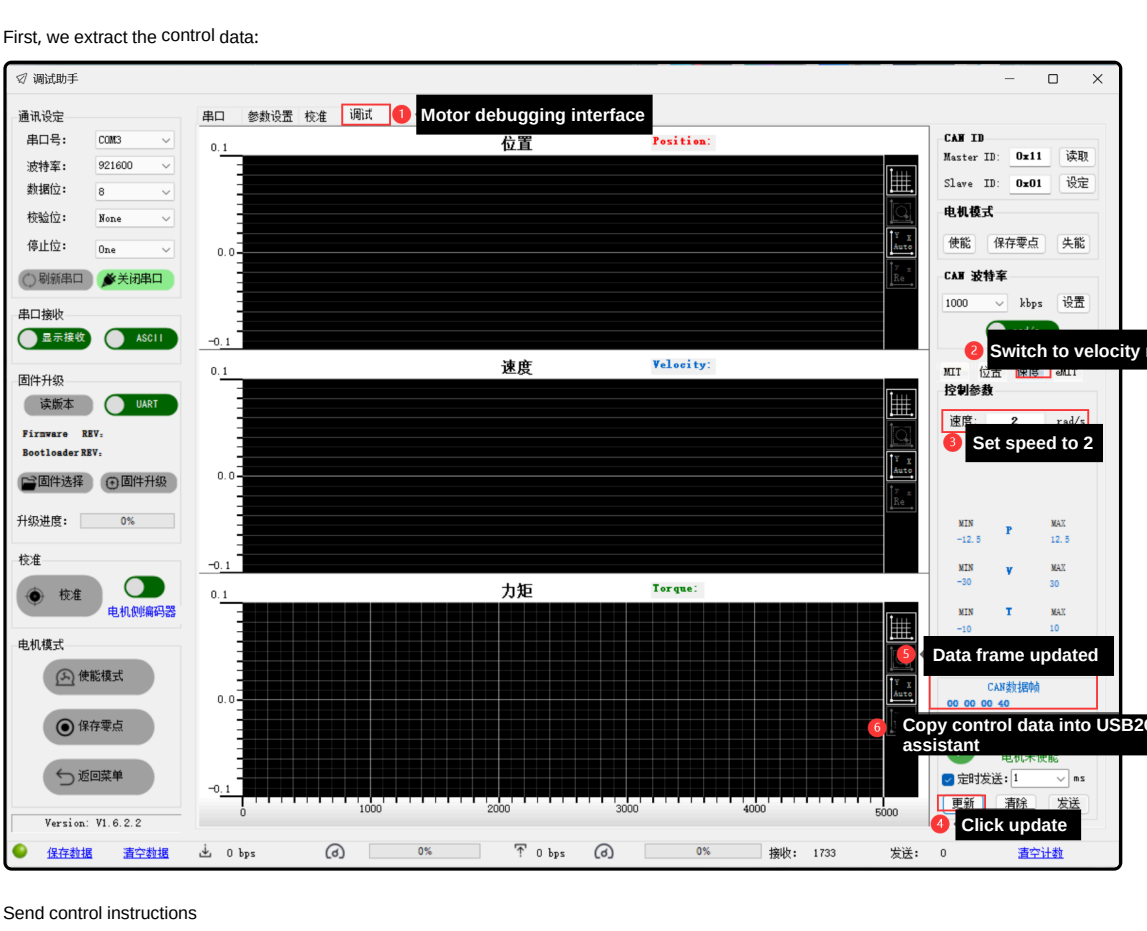
3 After sending enable command → feedback first bit = 1, indicator green → enable success

2 Change to motor enable command

1 Change to motor CAN ID + velocity mode offset 0x200

9.3 Sending Motor Control Commands

First, we extract the control data:



1 Motor debugging interface

2 Switch to velocity mode

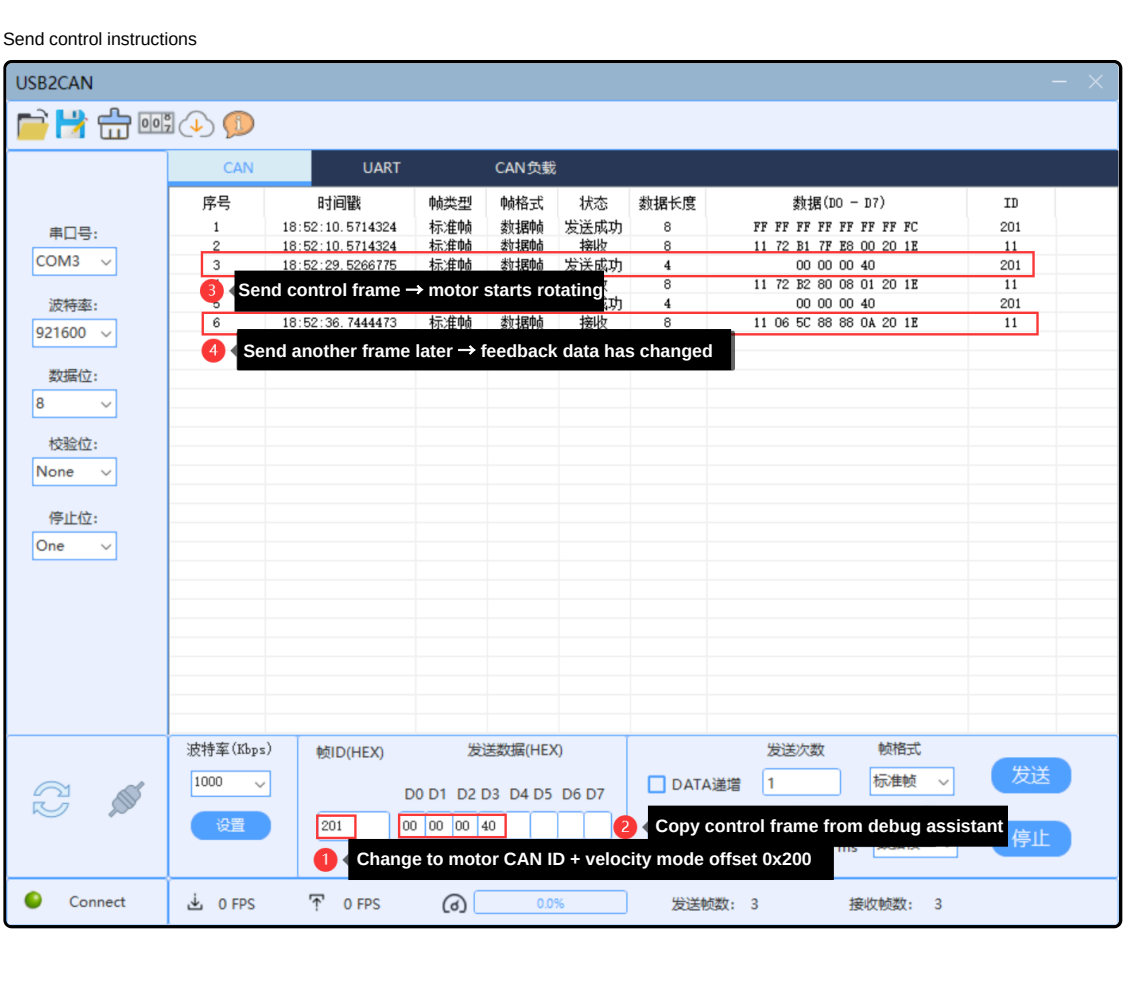
3 Set speed to 2

4 Data frame updated

6 Copy control data into USB2CAN assistant

5 Click update

Send control instructions



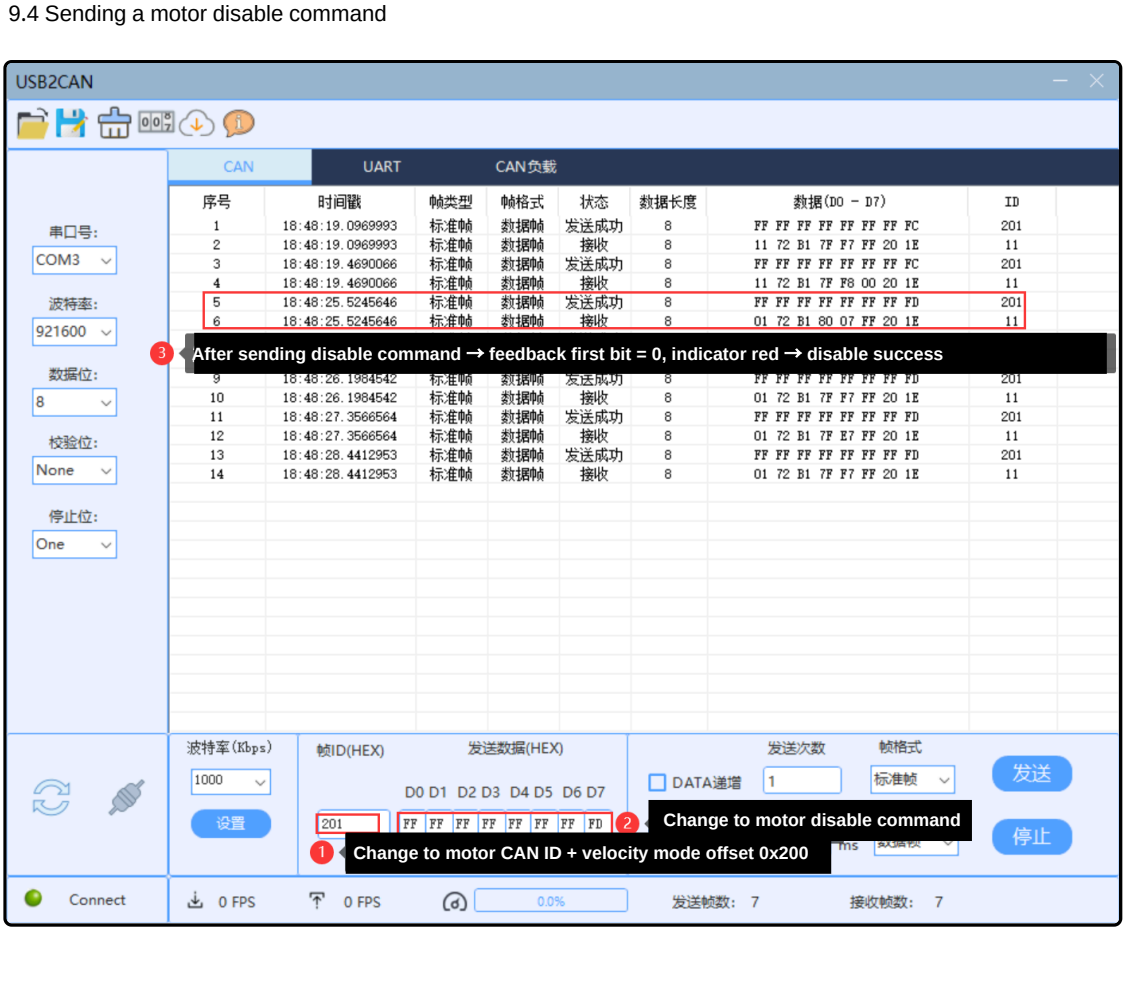
3 Send control frame → motor starts rotating

4 Send another frame later → feedback data has changed

2 Copy control frame from debug assistant

1 Change to motor CAN ID + velocity mode offset 0x200

9.4 Sending a motor disable command



3 After sending disable command → feedback first bit = 0, indicator red → disable success

2 Change to motor disable command

1 Change to motor CAN ID + velocity mode offset 0x200

10. Motor register operation

10.1 Read Command Format

Message ID	property	data[0]	data[1]	data[2]	data[3]
0x7FF	STD	CANID_L	CANID_H	0x33	RID

After successful reading, the data of the register will be returned, and the frame format is as follows:

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

The returned data has the low bit first and the highest bit last.

10.2 Write Command Format

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

After writing successfully, the written data will be returned, and the frame format is the same as that sent

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

The data is low-order first and the highest-order second

10.3 Storage Command Format

Message ID	property	data[0]	data[1]	data[2]	data[3]
0x7FF	STD	CANID_L	CANID_H	0xAA	RID

After successful writing, the return format is:

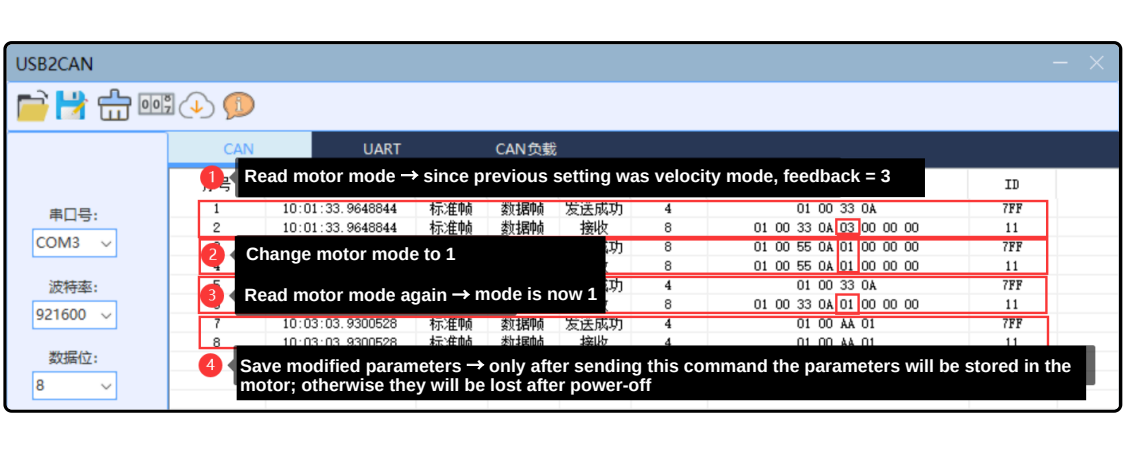
Message ID	property	data[0]	data[1]	data[2]	data[3]
MST_ID	STD	CANID_L	CANID_H	0xAA	0x01

10.4 Register Table 1

RID is a register. For a detailed register table, please refer to the control protocol document. Here we only take one register variable as an example.

Register address	variable	describe	Read and Write	scope	Data Type
10(0x0A)	CTRL_MODE	Control Mode	RW	[1,4]	uint32_t

10.5 Reading, Writing, and Saving Register Data



1 Read motor mode → since previous setting was velocity mode, feedback = 3

2 Change motor mode to 1

3 Read motor mode again → mode is now 1

4 Save modified parameters → only after sending this command the parameters will be stored in the motor; otherwise they will be lost after power-off

11. Use the development board to control the motor

- Here you can carefully study the DM control board routines
- Be proficient in using an external USB-to-CAN tool to check whether the data sent by your development board is consistent with the communication protocol.
- Be proficient in using the motor debugging assistant to set the motor's CAN ID, Master ID, and control mode to ensure that the corresponding program blocks in your development board match the parameters you set.
  For detailed debugging process, please refer to: Debugging Assistant User Manual (Damiao Drive Control Protocol) V1.4.pdf and Motor Getting Started Process (Part 2)