

# GL 云台电机使用说明

(含驱动器说明)

V1.0.0



## 目录

目录 .....	2
注意事项 .....	4
产品特色 .....	4
免责声明 .....	4
版本变更记录 .....	5
1. 驱动器产品信息 .....	6
1.1 驱动器外观简介&产品规格 .....	6
1.2 驱动器接口及定义 .....	7
1.2.1 驱动器接口图 .....	7
1.2.2 驱动器接口推荐品牌及型号 .....	7
1.2.3 驱动器接口引脚定义 .....	7
1.3 驱动器指示灯定义 .....	8
1.4 主要配件及规格 .....	8
2. S-link 产品信息 .....	9
2.1 S-link 外观简介&产品规格 .....	9
2.2 S-link 接口及定义 .....	10
2.2 S-link 指示灯定义 .....	11
3. 驱动器与 S-link 连接及注意事项 .....	11
4. 上位机使用说明 .....	12
4.1 上位机界面及说明 .....	12
4.1.1 主窗口 .....	13
4.2 设备连接 .....	16
4.3 驱动板校准 .....	17
4.4 控制演示 .....	19
4.4.1 控制模式设置 .....	19
4.4.2 MIT 模式 .....	22
4.4.3 位置速度模式 .....	27
4.4.4 速度模式 .....	30

---

4.4.5 PWM 模式 .....	32
4.5 固件升级 .....	34
5. 驱动板通讯协议及说明 .....	34
5.1 MIT 模式控制模式及说明 .....	35
5.2 位置速度模式控制模式及说明 .....	37
5.3 速度模式控制模式及说明 .....	38
5.4 CAN 反馈报文协议 .....	39
5.5 CAN 口控制命令实例 .....	41

## 注意事项

1. 确保电路无短路，接口按照要求正确连接。
2.  驱动板输出时，会出现发热情况，请小心使用，以免烫伤。
3.  使用前请检查各零部件是否完好。如有部件缺失、老化，请停止使用并及时联系技术支持。
4.  多种可选控制方式在驱动板运行时不可切换，且不同控制方式之间通信协议不同。如需切换，请重新启动电源，然后再进行更改。使用错误的协议控制可能会使驱动板烧毁！
5.  请严格按照本文规定的工作电压、电流、温度等参数使用，否则会对产品造成永久性的损坏！

## 产品特色

GL 云台电机驱动板采用同级别中高性能的驱动芯片，使用 Field Oriented Control（简称 FOC）算法，搭配先进自抗扰控制技术对速度和角度进行控制。可配合上位机软件进行参数设置并升级固件。

## 免责声明

感谢您购买 GL 云台电机。在使用之前，请仔细阅读本声明，一旦使用，即被视为对本声明全部内容的认可和接受。请严格遵守产品说明书和相关的法律法规、政策、准则安装和使用该产品。在使用产品过程中，用户承诺对自己的行为及因此而产生的所有后果负责。

因用户不当使用、安装、改装造成的任何损失，CubeMars 将不承担法律责任。

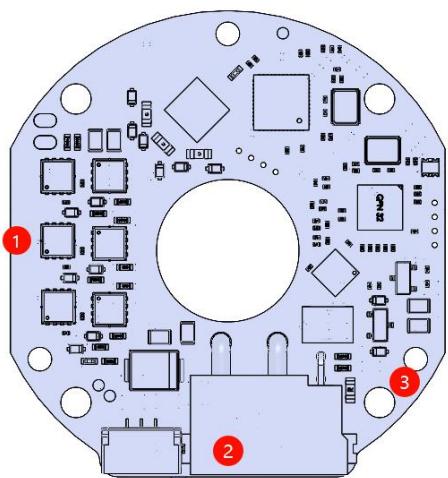
CubeMars 是南昌酷德智能科技有限公司及其关联公司的商标。本文出现的产品名称、品牌等，均为其所属公司的商标。本产品及手册为南昌酷德智能科技有限公司版权所有。未经许可，不得以任何形式复制翻印。关于免责声明的最终解释权，归南昌酷德智能科技有限公司所有。

## 版本变更记录

日期	版本	变更内容
2024.07.18	Ver. 1.0.0	创建版本

## 1. 驱动器产品信息

### 1.1 驱动器外观简介&产品规格

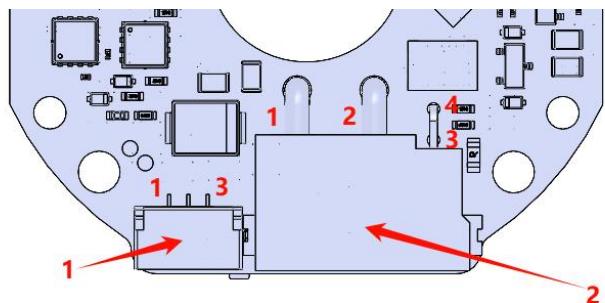


- ①三相动力线连接端
- ②连接端口
- ③安装孔

产品规格	
额定工作电压	24V
允许电压	8~32V
额定工作电流	2A
最大允许电流	10A
待机功耗	≤50mA
CAN 总线比特率	1Mbps
尺寸	41mm×38mm
工作环境温度	-20°C至 65°C
控制板最大允许温度	120°C
控制精度	0.1°

## 1.2 驱动器接口及定义

### 1.2.1 驱动器接口图



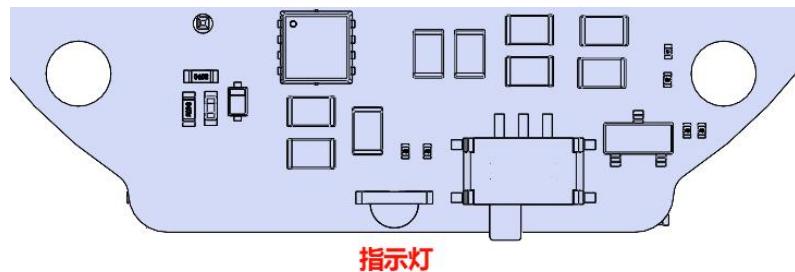
### 1.2.2 驱动器接口推荐品牌及型号

序号	板载接口型号	品牌	线端接口型号	品牌
1	A1257WR-S-3P	CJT(长江连接器)	A1257H-3P	CJT(长江连接器)
2	XT30 (2+2) PW-M	AMASS(艾迈斯)	XT30 (2+2) -F	AMASS(艾迈斯)

### 1.2.3 驱动器接口引脚定义

序号	接口功能	引脚	说明
1	串口通讯	1	串口信号地 (GND)
		2	串口信号输出 (TX)
		3	串口信号输入 (RX)
2	电源输入及CAN 通讯	1	电源正极 (+)
		2	电源负极 (-)
		3	CAN 通讯高侧 (CAN_H)
		4	CAN 通讯低侧 (CAN_L)

## 1.3 驱动器指示灯定义



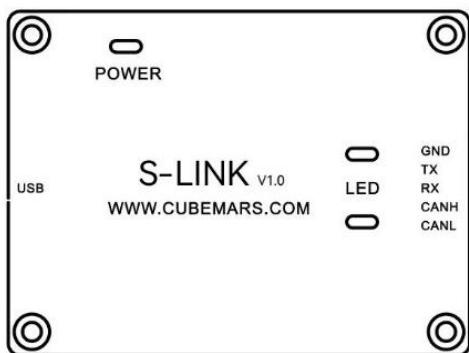
指示灯定义	
1. 电机失能指示 (亮时红灯)	失能指示灯，用于指示电机运行状态
2. 电机使能指示 (亮时绿色)	使能指示灯，用于指示电机运行状态
3. 驱动报错指示 (红灯闪烁时)	驱动报错指示灯，用于指示驱动板报错情况，正常情况下只有当驱动板出现报错才会闪烁。

## 1.4 主要配件及规格

序号	项目	规格		数量	备注
1	电源及信号线插头	电源及CAN线	16AWG 红黑硅胶线和白蓝铁氟龙 30#-OD0.64-300±10mm-4一端 XT30(2+2)-F, 一端剥皮浸锡 3±1mm	各 1PCS	±2MM
2		串口线	铁氟龙 30#线 OD0.64-300mm-3-GH1.25-3PIN 公头转 JR-3PIN 公头-NULL	各 1PCS	±2MM

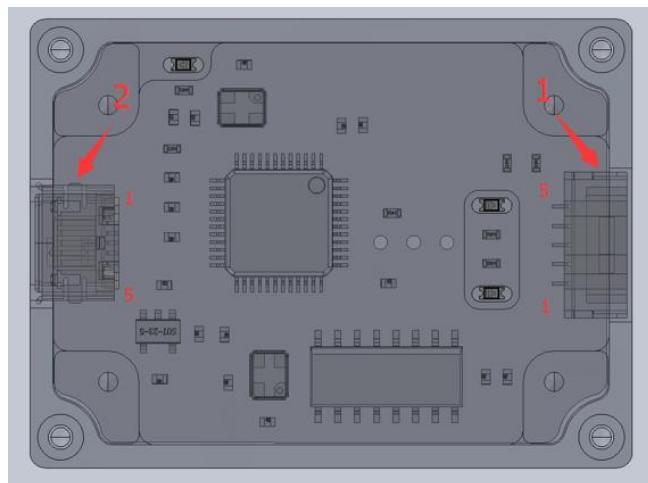
## 2. S-link 产品信息

### 2.1 S-link 外观简介&产品规格



产品规格	
额定工作电压	5V
待机功耗	≤30mA
外形尺寸	39.2x29.2x10MM
工作环境温度	-20°C 至 65°C
控制板最大允许温度	85°C

## 2.2 S-link 接口及定义

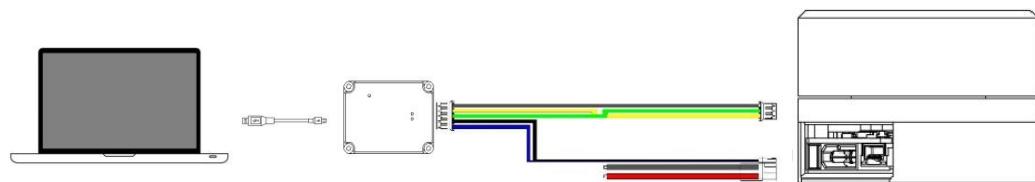


序号	接口功能	引脚	说明
1	通讯接口	1	CAN 通讯低侧 (CAN_L)
		2	CAN 通讯高侧 (CAN_H)
		3	串口信号输入 (RX)
		4	串口信号输出 (TX)
		5	串口信号地 (GND)
2	USB 接口	1	VBUS
		2	D-
		3	D+
		4	ID
		5	GND

## 2.2 S-link 指示灯定义

序号	颜色定义	说明
1	绿色	电源指示灯，用于指示 S-link 电源情况，正常情况下电源插入时都会亮起绿色，如插入电源时绿灯不亮，请立刻移除电源，切不可再上电。
2	蓝色	串口通讯输出指示灯 (TX)，常灭，当 S-link 串口有数据输出时，闪烁。
3	红色	串口通讯输入指示灯 (RX)，常灭，当 S-link 串口有数据输入时，闪烁。

## 3. 驱动器与 S-link 连接及注意事项



S-link 上的 USB 线 -----> PC 端  
2+2Pin 端子（电源及 CAN 口）-----> 电机上的 2+2Pin 口（电源及 CAN）  
3Pin 端子（UART 口）-----> 电机上的 3Pin 口（UART）

## 4.上位机使用说明

## 4.1 上位机界面及说明

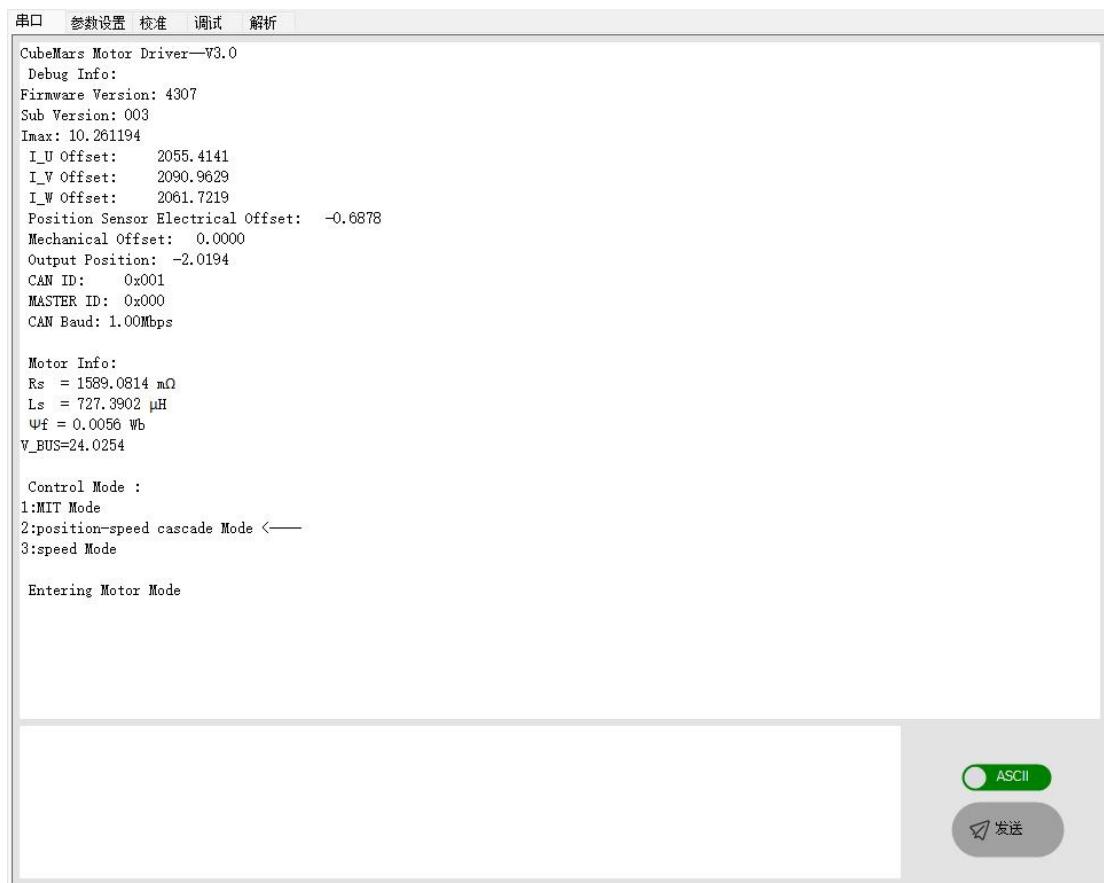


- A. 通讯设置
  - B. 固件升级
  - C. 校准
  - D. 电机模式
  - E. 主窗口
  - F. 通讯状态
  - G. 中英文切换

## 4.1.1 主窗口

### 4.1.1.1 串口

该页面主要显示串口的接收数据和发送串口数据。



### 4.1.1.2 参数设置

参数设置界面包含电机参数、控制幅值、控制设置、驱动参数、其他等。

- 1、电机参数：对电机相电阻、相电感等参数进行标定；
- 2、控制幅值：设置位置、速度和扭矩等幅值参数；
- 3、控制设置：修改电机控制模式、电流带宽、速度及位置的 KP、KI 值等，通过“暂存”写入参数在关闭电机电源后不会进行保存（“暂存”无法修改电机控制模式）；
- 4、驱动参数：修改极对数、减速比、CAN ID、Master ID、电压、电流、温度和速度等驱动参数；
- 5、其他：读写参数按键及通信格式说明等。

⚠️ 请务必严格按照规定电压、电流、功率、温度使用。因违规操作本产品导致对人体造成伤害，或对驱动板及电机造成不可逆的损伤，我司将不承担任何法律责任。

串口 参数设置 校准 调试 解析

电机参数	驱动参数			
参数标定				
相电阻 (R) : 1589.081 mΩ	极对数: 14	欠压: 15	加速度: 2	<span style="border: 1px solid blue; padding: 2px; float: right;">读参数</span>
相电感 (L) : 727.3902 uH	减速比: 1	过压: 32	减速度: -2	<span style="border: 1px solid blue; padding: 2px; float: right;">写参数</span>
磁 链 (λv) : 0.005579725 wb	CAN ID: 0x01	过温: 100	限速: 153.3178	
粘滞系数: 0.0001270785	Master ID: 0x00	CAN Timeout: 0	过流: 0.8	

控制幅值							
P <sub>MAX</sub> :	12.5	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]
V <sub>MAX</sub> :	30	MST_ID	ID ERR<<4	POS[15:8]	POS[7:0]	VEL[11:4]	T[7:0]
T <sub>MAX</sub> :	10	D[6]	D[7]	VEL[3:0]T[11:8]	T[7:0]	T_MOS	T_Rotor
K <sub>T_OUT</sub> :	0	控制报文		D[4]	D[5]	D[6]	D[7]
齿轮系数:	1	0x100+ID		p_des	v_des		
阻尼因子:	4						

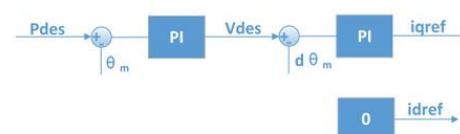
  

控制设置		<span style="border: 1px solid blue; padding: 2px; margin-right: 10px;">读取</span> <span style="border: 1px solid blue; padding: 2px;">暂存</span> <span style="border: 1px solid blue; padding: 2px; margin-top: 10px;">?</span>
控制模式:	2: 位置速度模式	
电流带宽:	1000	
速度 KP :	0.618	
速度 KI :	0.009	
位置 KP :	54	
位置 KI :	0	

计算 KP

ID 表示控制器的 ID，取 CAN\_ID 的低 8 位  
 ERR 表示故障，对应故障类型为：  
 8—超压； C—电机线圈过温  
 9—欠压； D—通讯丢失；  
 A—过电流； E—过载；  
 B—MOS 过温

POS 表示电机的位置信息  
 VEL 表示电机的速度信息  
 T 表示电机的扭矩信息  
 T\_MOS 表示驱动上 MOS 的平均温度，单位 °C  
 T\_Rotor 表示电机内部线圈的平均温度，单位 °C  
 位置、速度和扭矩采用线性映射的关系将浮点型数据转换成有符号的定点数据，其中位置采用 16 位数据，速度和扭矩均使用 12 位。



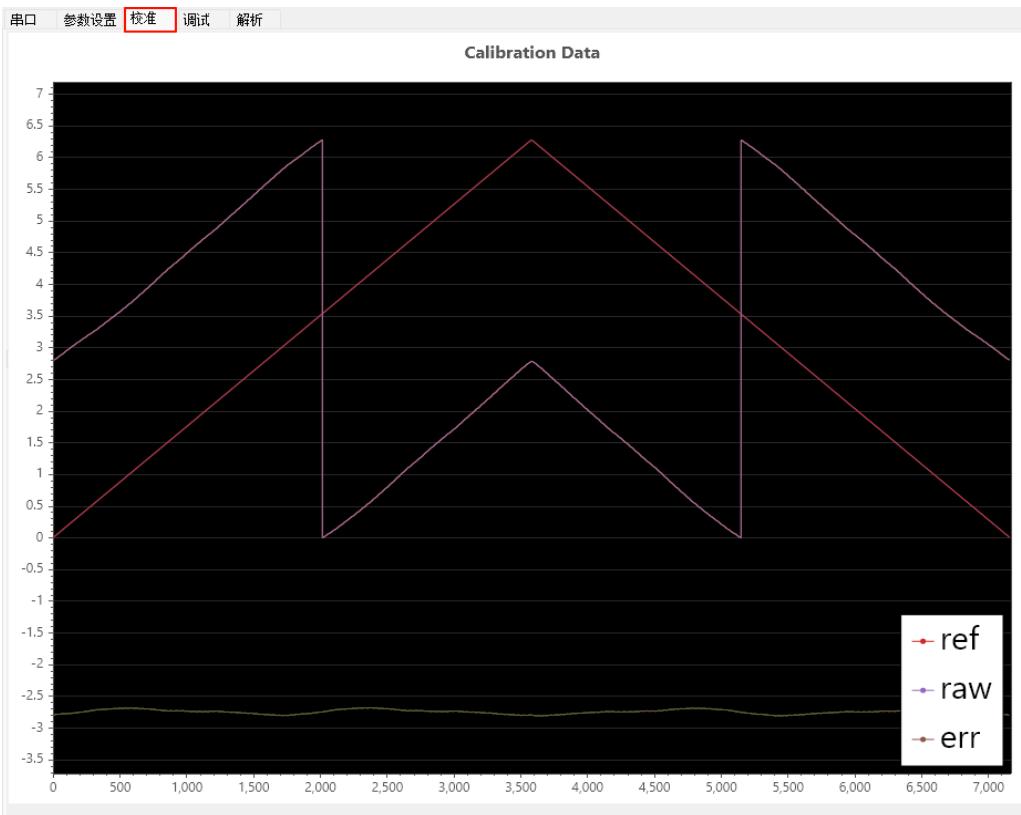
```

    graph LR
      Pdes((Pdes)) --> PI1[PI]
      PI1 --> Vdes((Vdes))
      Vdes --> PI2[PI]
      PI2 --> iqref[iqref]
      iqref --> idref[idref]
      idref --> 0[0]
  
```

⚠: 修改控制幅值后需重新点击“读参数”

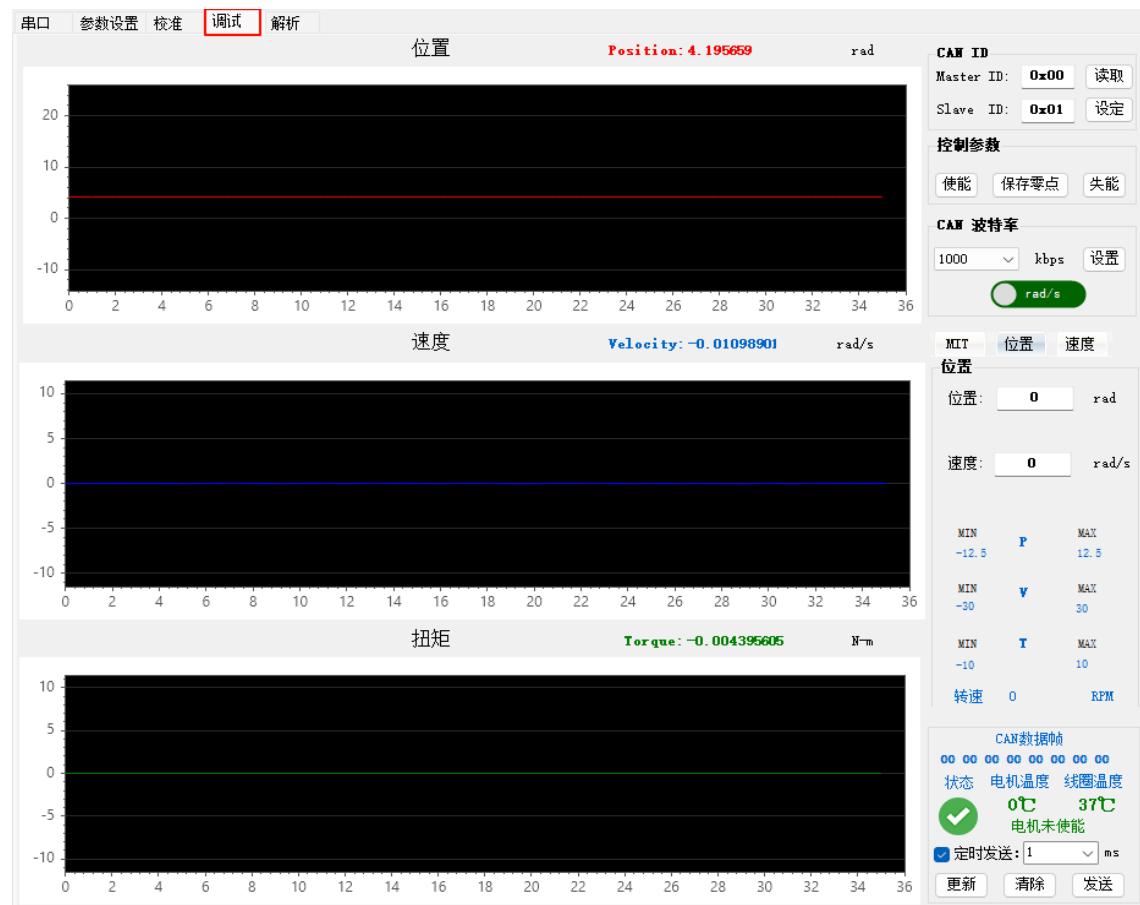
#### 4.1.1.3 校准

该页面显示校准过程波形与进度。



#### 4.1.1.4 调试

该页面主要为设置 CAN ID、波特率和修改 MIT 模式、位置速度模式、速度模式下的控制参数进行调试，并实时显示电机位置、速度、力矩、温度等状态信息。



⚠️ 请务必严格按照规定电压、电流、功率、温度使用。因违规操作本产品导致对人体造成伤害，或对驱动板及电机造成不可逆的损伤，我司将不承担任何法律责任。

#### 4.1.1.5 解析

该页面可解析 CAN 反馈报文，通过设置位置、速度、力矩范围大小将 CAN 反馈报文解析为实际数据。

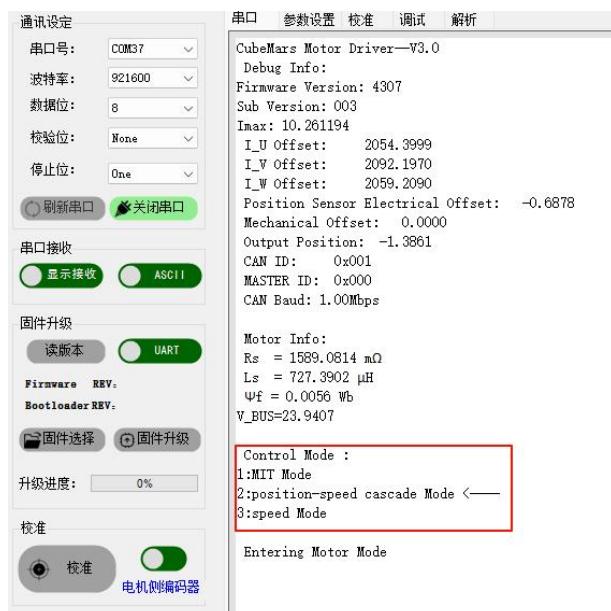


## 4.2 设备连接

首先，连接电机的串口、CAN 口以及电源接口。电脑端打开上位机软件，选择相应的串口设备并打开串口。



此时给电机供电（点），串口会打印下述信息。Control Mode :指示当前驱动模式（电机上电后自动进入此模式使能状态）。不同的模式采用不同的命令格式（参考第 5 章节驱动板通信协议及说明）。



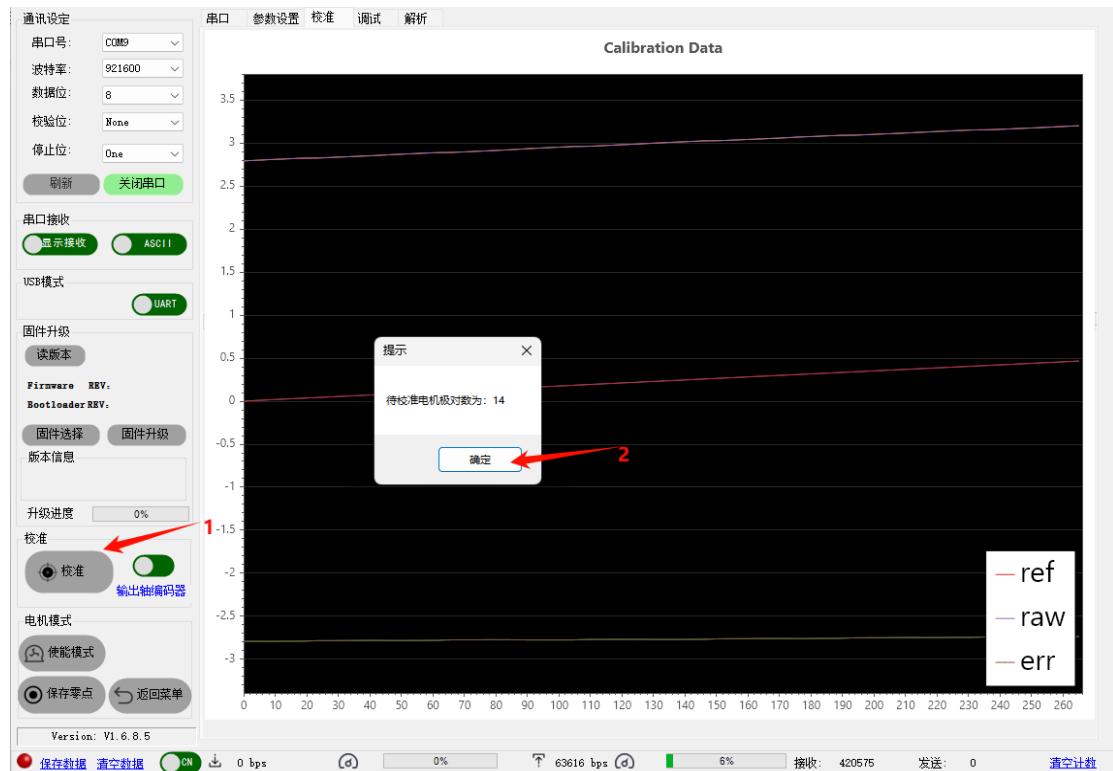
## 4.3 驱动板校准

当你重新安装了驱动板在电机上，或者更换过电机三相线的线序，亦或者更新了固件之后，都必须进行校准。校准之后，便可以正常使用电机，过程中电机会正向反向转动一个转子周期，请保证电机可以自由转动。

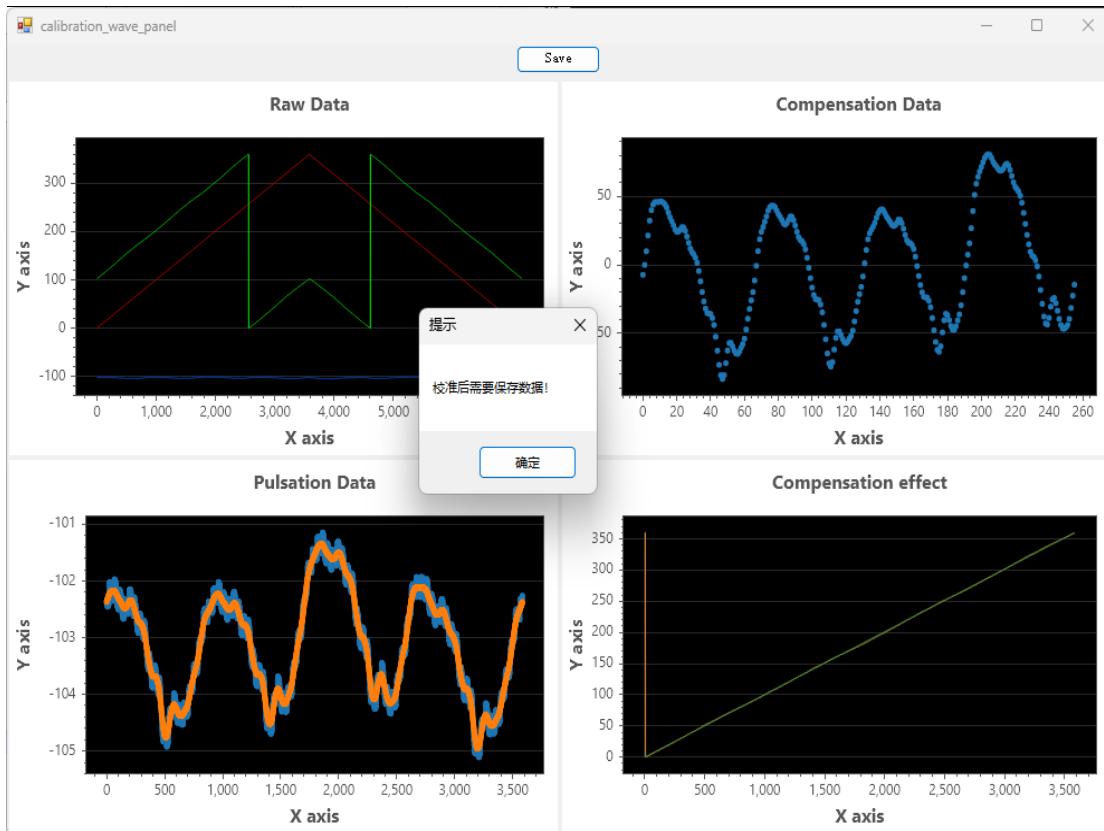
**⚠️ 确保电机空载运行，否则可能造成校准失败。**

### (1) 传感器校准

点击校准，电机会先转动，转动完毕后向上位机返回极对数。同时电机进行正反方向转动以校准传感器的数据，界面下方有校准进度条。



校准完成会，会自动弹出校准的效果图。



## (2) 校准数据排查与存储

应当特别关注“Compensation Data 补偿点云”的值，建议不要超过 $\pm 150$ ，若超过 150 可能有几个原因：

- ① 极对数识别错误；
- ② 电机阻力过大，导致卡顿；
- ③ 传感器安装不规范。

如排除以上原因，校准值依然偏大，请联系厂家。

校准数据无误后点击“保存”，上位机会将校准数据传输到驱动器中存储。

## (3) 电机参数识别

校准完成后，可以进行参数辨识，主要识别电机的相电阻、相电感以及磁链等重要参数。点击“参数设置”标签卡，再点击“参数标定”按钮，驱动器进入辨识步骤，期间电机会转动，辨识完成后，会自动上传辨识的结果。

**⚠: 保持空载状态，并固定电机。**



粘滞系数仅作为参考，可多次标定。

**⚠:** 以上参数不能出现负值，若出现负值，请确认电机状态后再行标定。

## 4.4 控制演示

### 4.4.1 控制模式设置

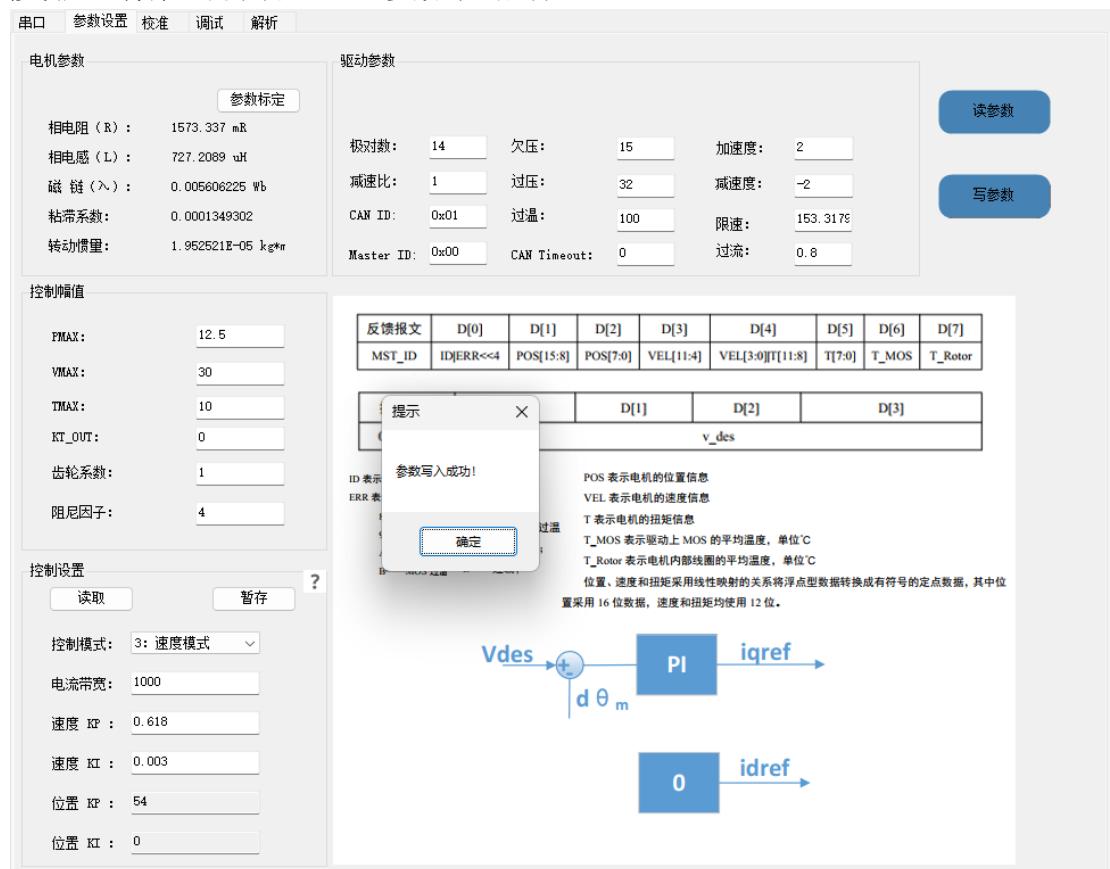
#### (1) 模式设置

确认电机输入电源稳定，S-link 连接正常，在与上位机成功连接后，依次点击上位机“参数设置”、“读参数”后，在控制设置中点击“控制模式”，可选择 MIT 模式、速度位置模式、速度模式三种模式，然后点击“写参数”进行控制模式设置。

**⚠:** 注意如无需修改控制模式，每次重启软件也需要点击“读参数”后再进行控制。



修改后, 将弹出提示窗口: “参数写入成功!”



## (2) 模式检查

每次修改控制模式后重新给驱动上电, 通过上电串口打印的数据或在参数设定页面上重新读

参数后显示的信息，箭头指向的模式为当前驱动的控制模式。

```

串口 参数设置 校准 调试 解析
CubeMars Motor Driver—V3.0
Debug Info:
Firmware Version: 4307
Sub Version: 003
I_max: 10.261194
I_U_Offset: 2054.4641
I_V_Offset: 2092.0391
I_W_Offset: 2059.3291
Position Sensor Electrical Offset: -0.6881
Mechanical Offset: 3.1720
Output Position: -1.0675
CAN ID: 0x001
MASTER ID: 0x000
CAN Baud: 1.00Mbps

Motor Info:
R_s = 1573.3369 mΩ
L_s = 727.2089 μH
Ψ_f = 0.0056 Wb
V_BUS=23.9425

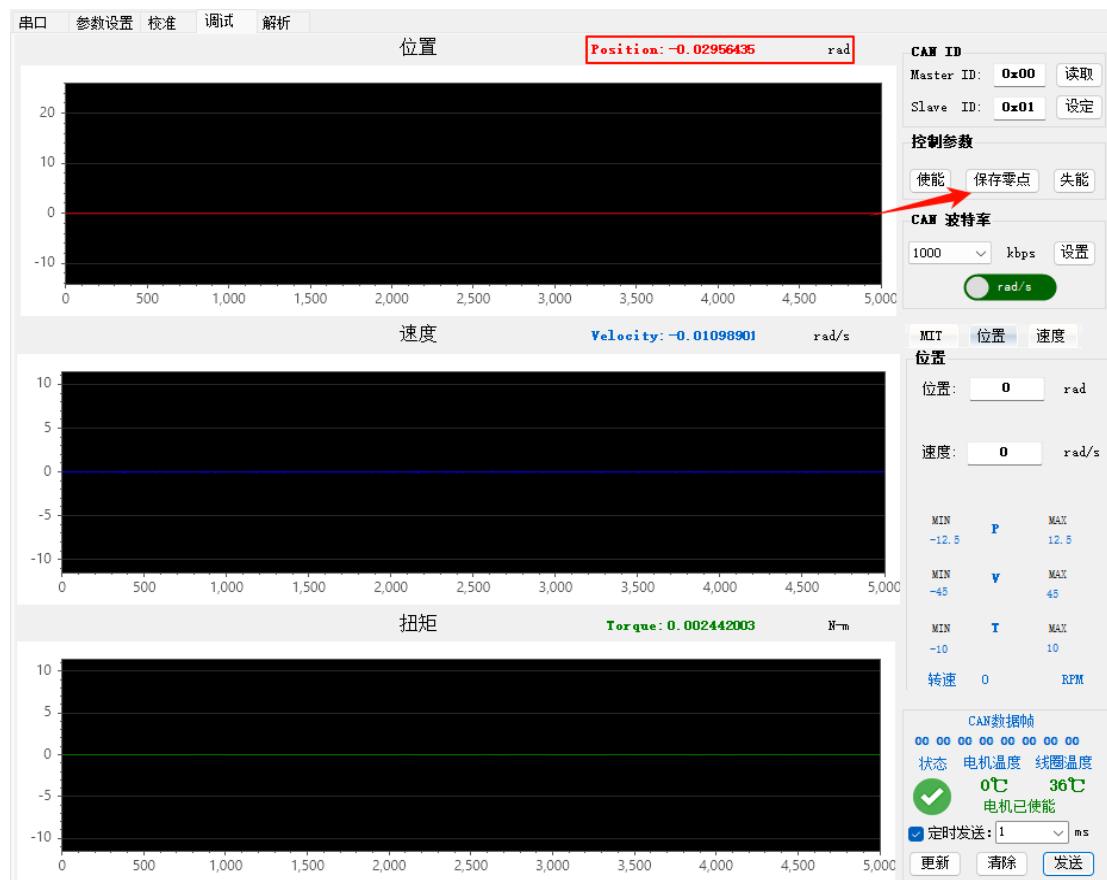
Control Mode :
1:MIT Mode
2:position-speed cascade Mode ←—
3:speed Mode

Entering Motor Mode

```

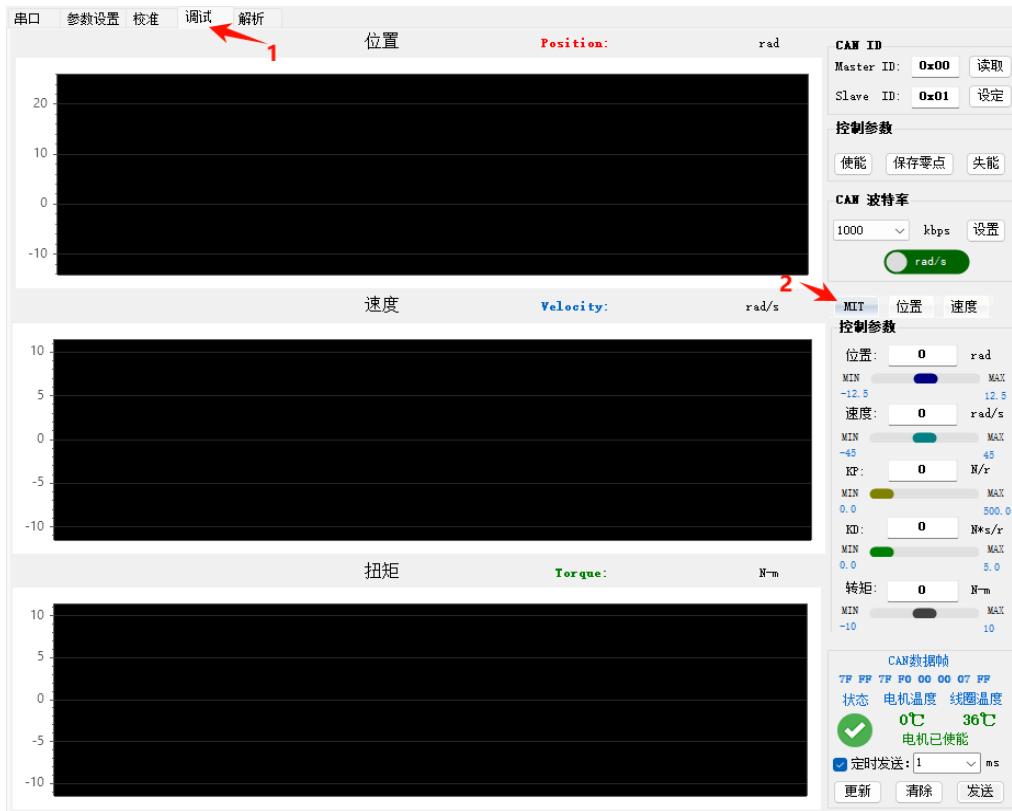
### (3) 保存零点

在主窗口中的“调试”页面中，点击电机模式下的保存零点。

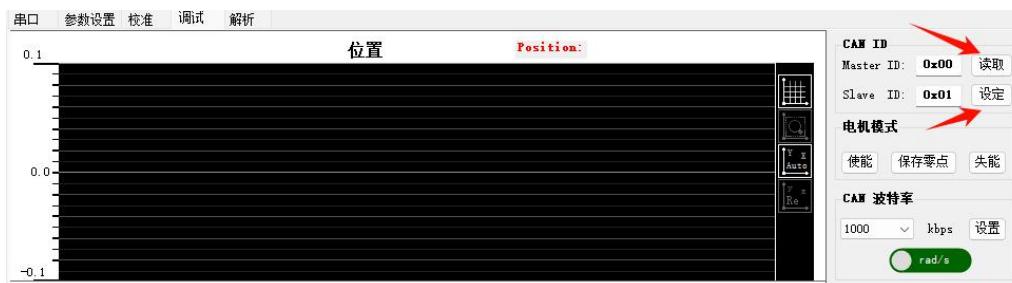


#### 4.4.2 MIT 模式

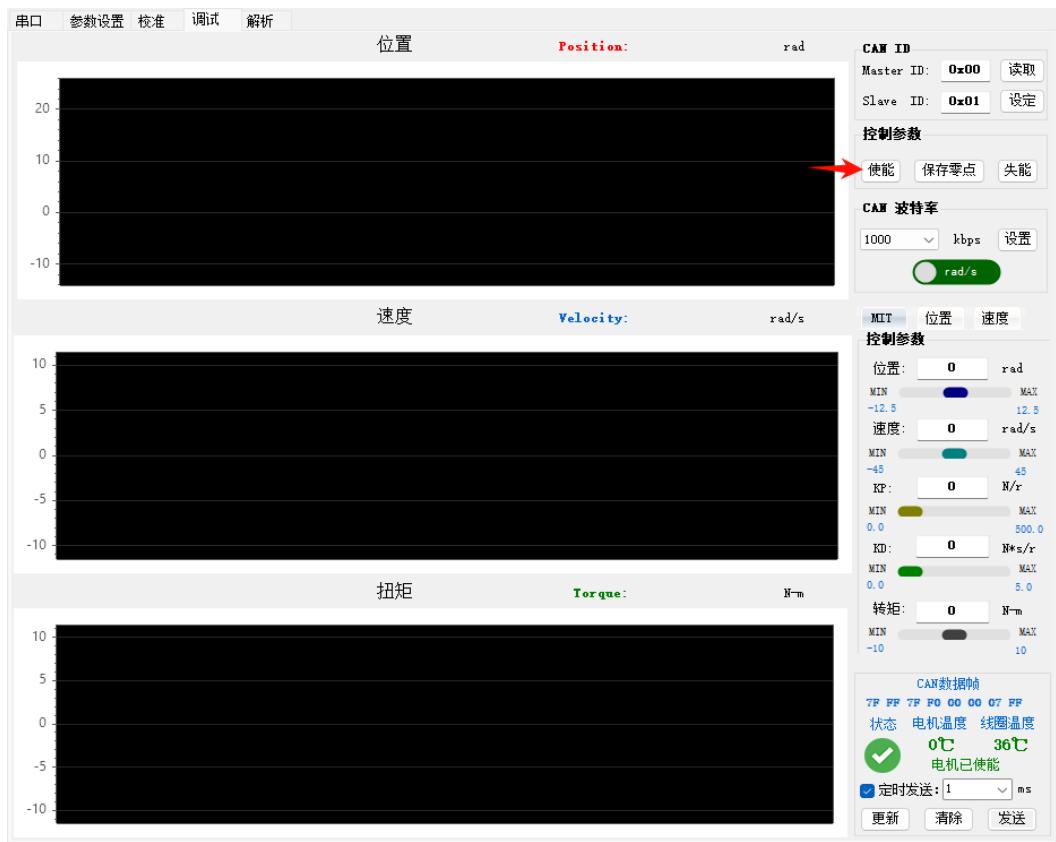
根据 4.4.1 节设置电机控制模式，确认当前控制模式为 MIT 模式，在“调试”页面中选择对应的 MIT 子标签卡。



确保 CAN ID 正确（可通过串口打印信息或者参数设置页面获取，也可以通过调试页面读取、设定按钮设定）



在电机模式栏点击“使能”按钮（电机上电默认进入使能状态）。此时驱动器绿色灯亮起，表示进入电机模式。

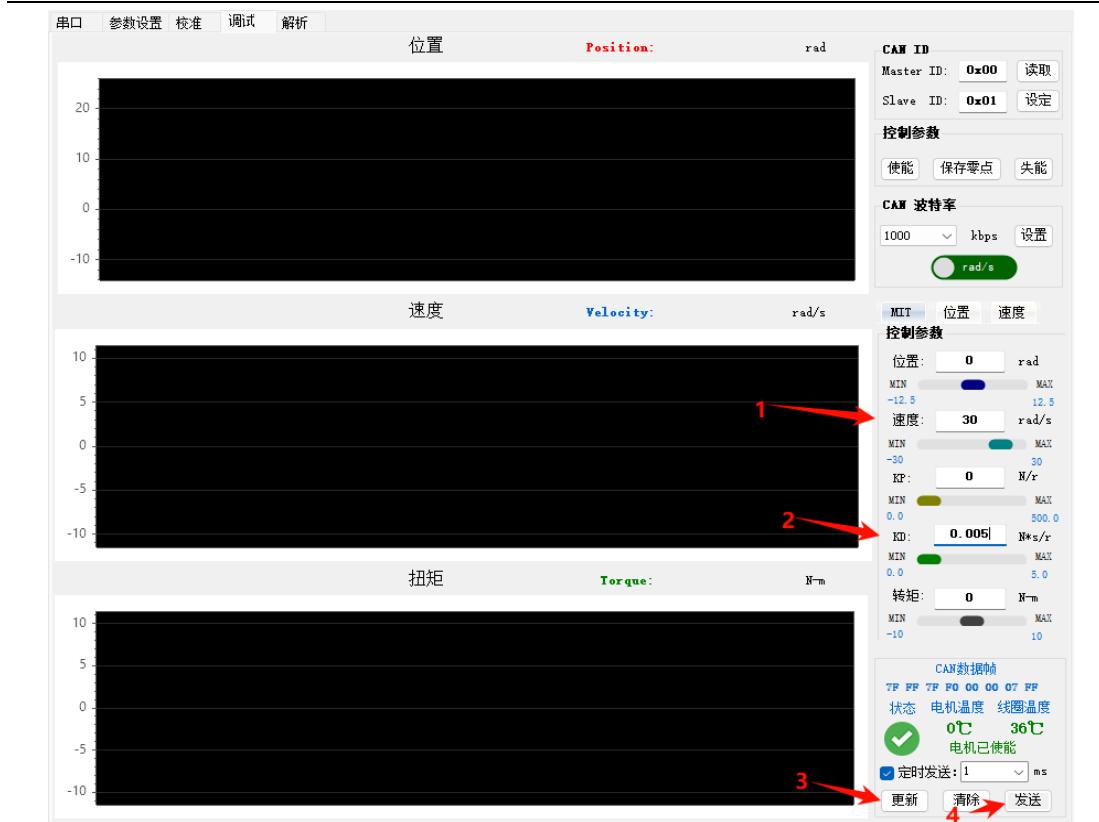


MIT 模式有三种控制方法：速度、位置、力矩（三种控制模式见 4.4.2.x）

#### 4.4.2.1 速度控制

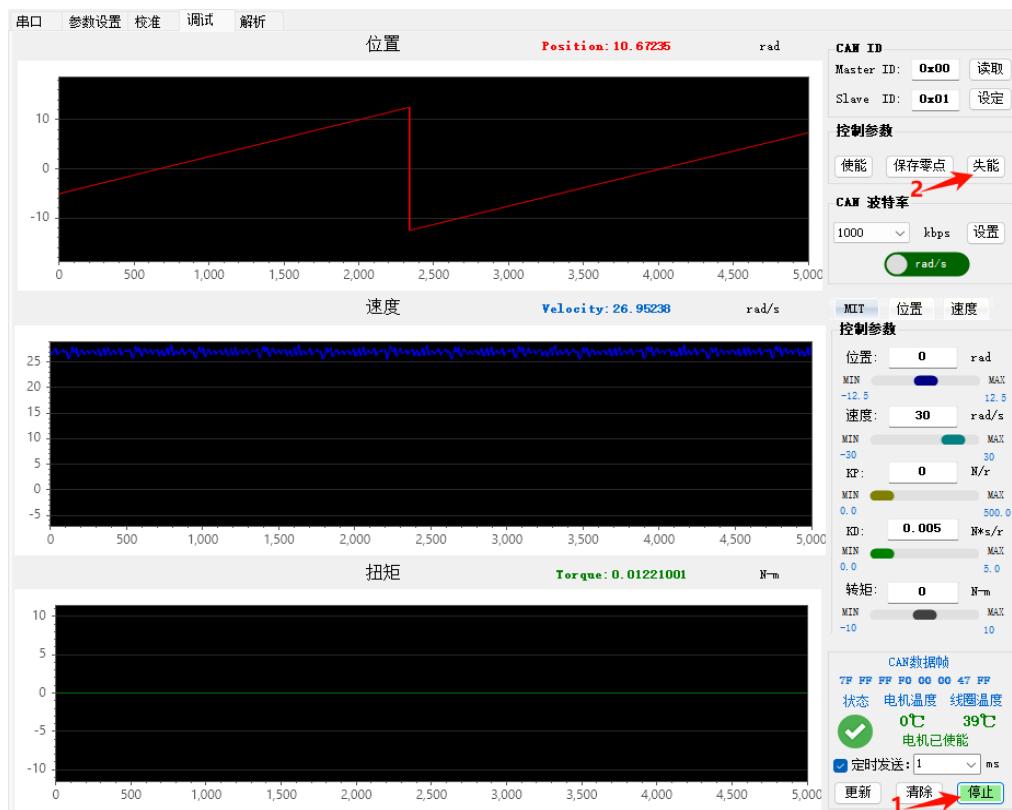
例如：在“控制参数”中速度设置为 30r/s, KD 为 0.005N\*s/r 其余全部给 0，并勾选“定时发送”框，依次点击“更新”按钮和“发送”按钮，电机将会按照期望速度运行，可在调试界面查看参数曲线变化图。

**⚠ 注意固定电机。**



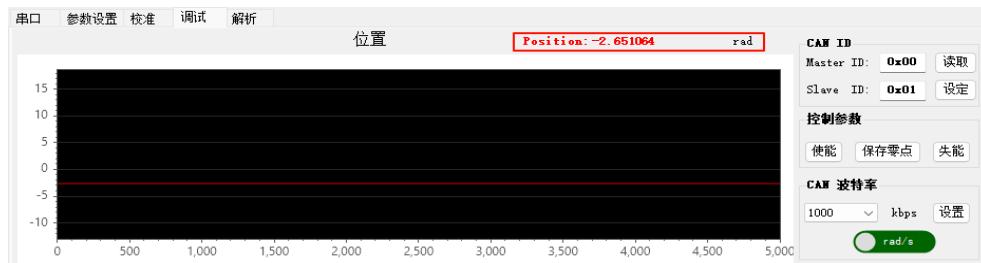
根据调试需求，需要修改控制参数查看调试变化，在原界面直接对参数进行修改，并保持勾选“定时发送”，点击“更新”按钮即可进行调试。

退出调试时，依次点击“停止”和“失能”按钮。驱动器红色灯亮起，表示退出电机模式。



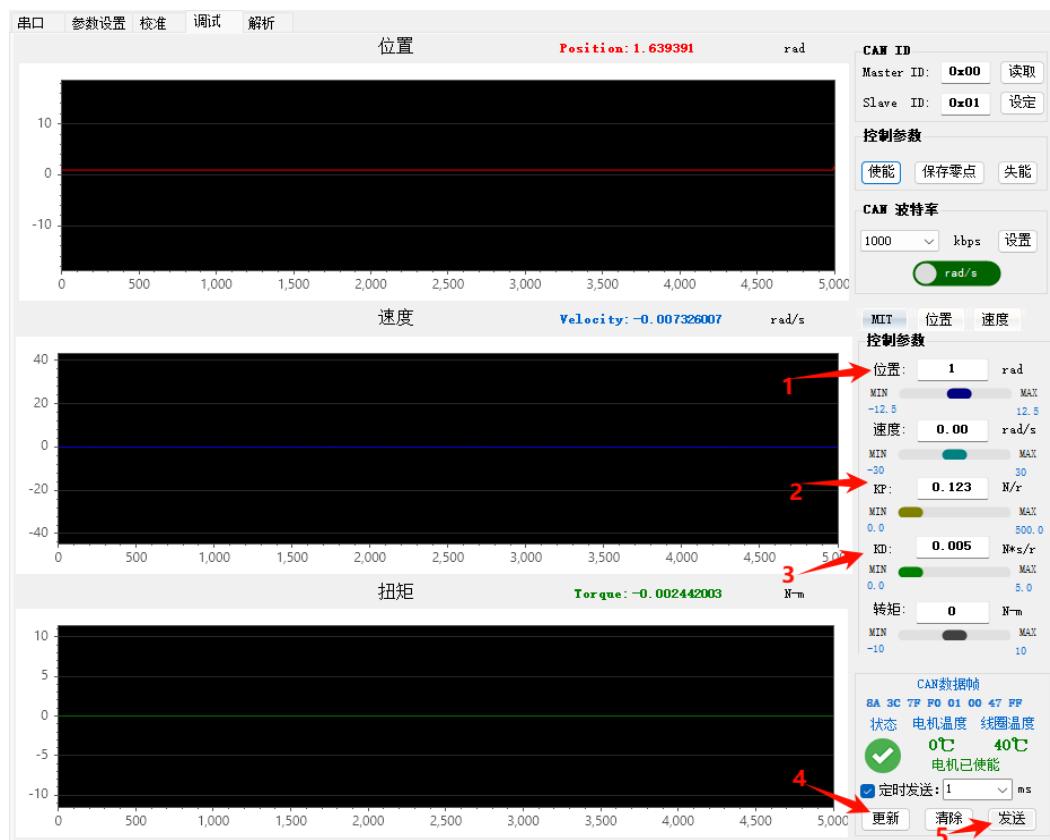
#### 4.4.2.2 位置控制

设置参数前，需注意电机的初始位置，以此为参考对参数进行设置。



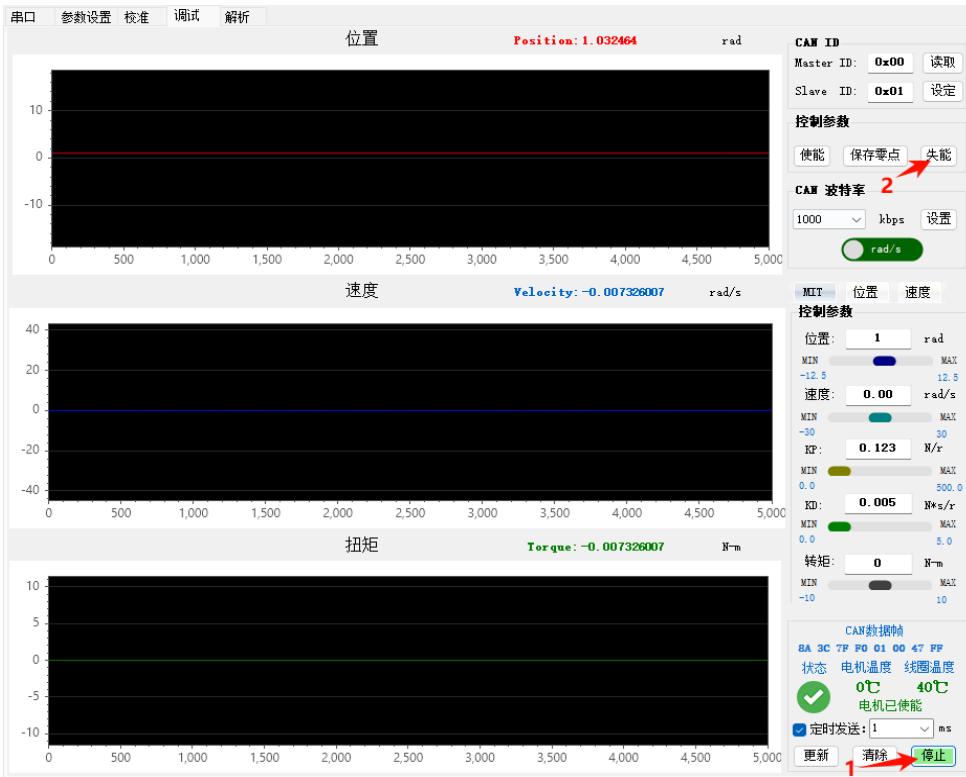
在“控制参数”中给定位置让电机旋转到指定位置，例如：在“控制参数”中位置设定 1rad，KP 为 0.123N/r，KD 为 0.005N\*s/r 其余全部为 0，并勾选“定时发送”框，依次点击“更新”按钮和“发送”按钮，可在调试界面查看参数曲线变化图。

**⚠ 注意电机初始位置，对控制参数栏中的“位置”参数给定数值时，避免与初始位置差距过大，引起电机冲击。注意固定电机。**



根据调试需求，需要修改控制参数查看调试变化，在原界面直接对参数进行修改，并保持勾选“定时发送”，点击“更新”按钮即可进行调试。

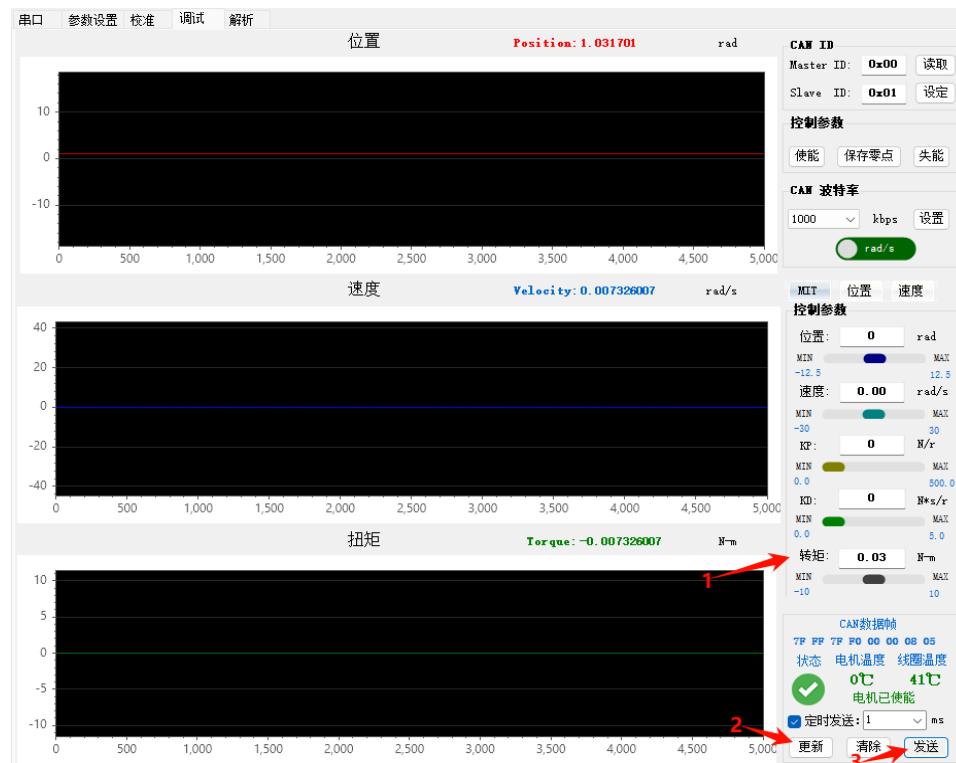
退出调试时，依次点击“停止”和“失能”按钮。驱动器红色灯亮起，表示退出电机模式。



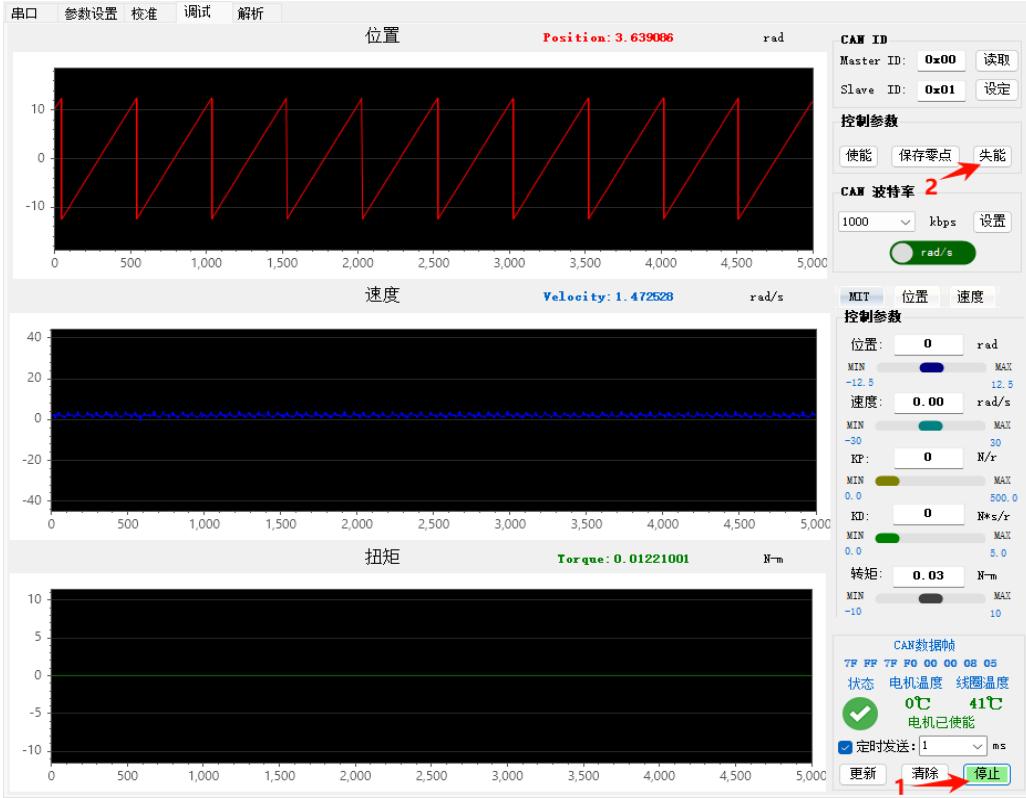
#### 4.4.2.3 力矩控制

在“控制参数”中给定期望力矩运行，例如：在“控制参数”中转矩设定为 0.03N·m，其余全部为 0，并勾选“定时发送”框，依次点击“更新”按钮和“发送”按钮，可在调试界面查看参数曲线变化图。

**⚠ 注意固定电机，在空载情况下，即使给定很小的力矩，电机也会加速到最大转速旋转。**



退出调试时，依次点击“停止”和“失能”按钮。驱动器红色灯亮起，表示退出电机模式。

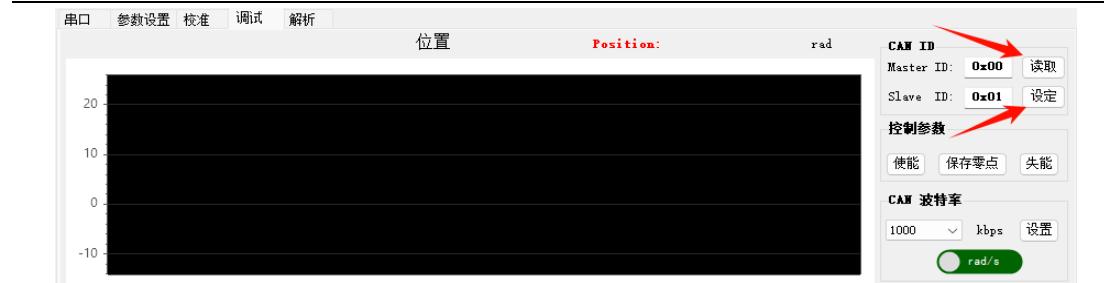


#### 4.4.3 位置速度模式

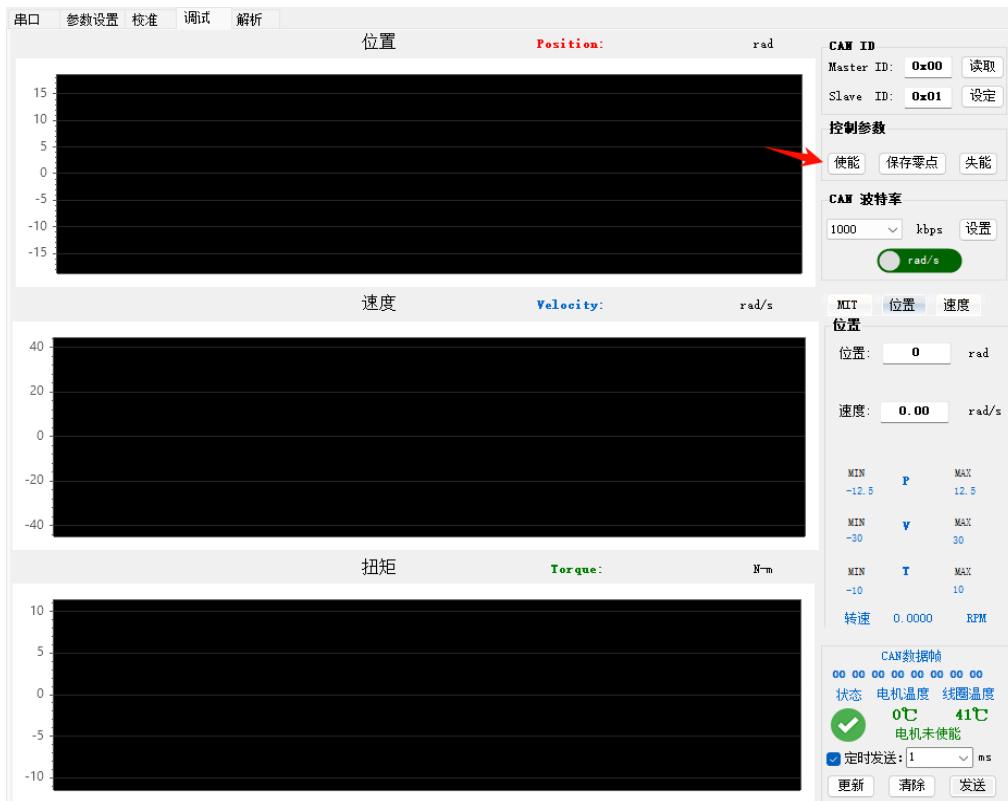
根据 4.4.1 节设置电机控制模式，确认当前控制模式为位置速度模式，在“调试”页面中选择对应的“位置”子标签卡。



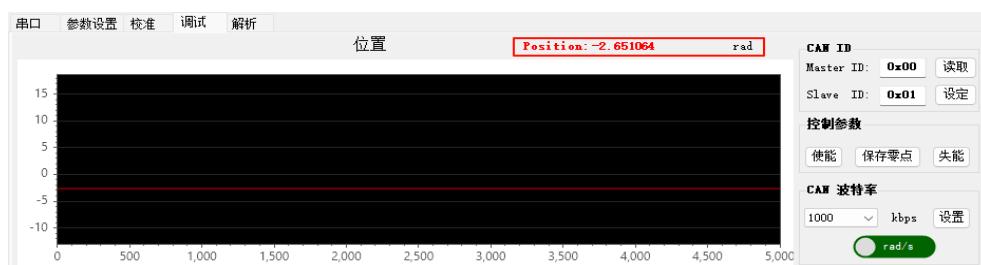
确保 CAN ID 正确（可通过串口打印信息或者参数设置页面获取，也可以通过调试页面读取、设定按钮设定）



在电机模式栏点击“使能”按钮（电机上电默认进入使能状态）。此时驱动器绿色灯亮起，表示进入电机模式。

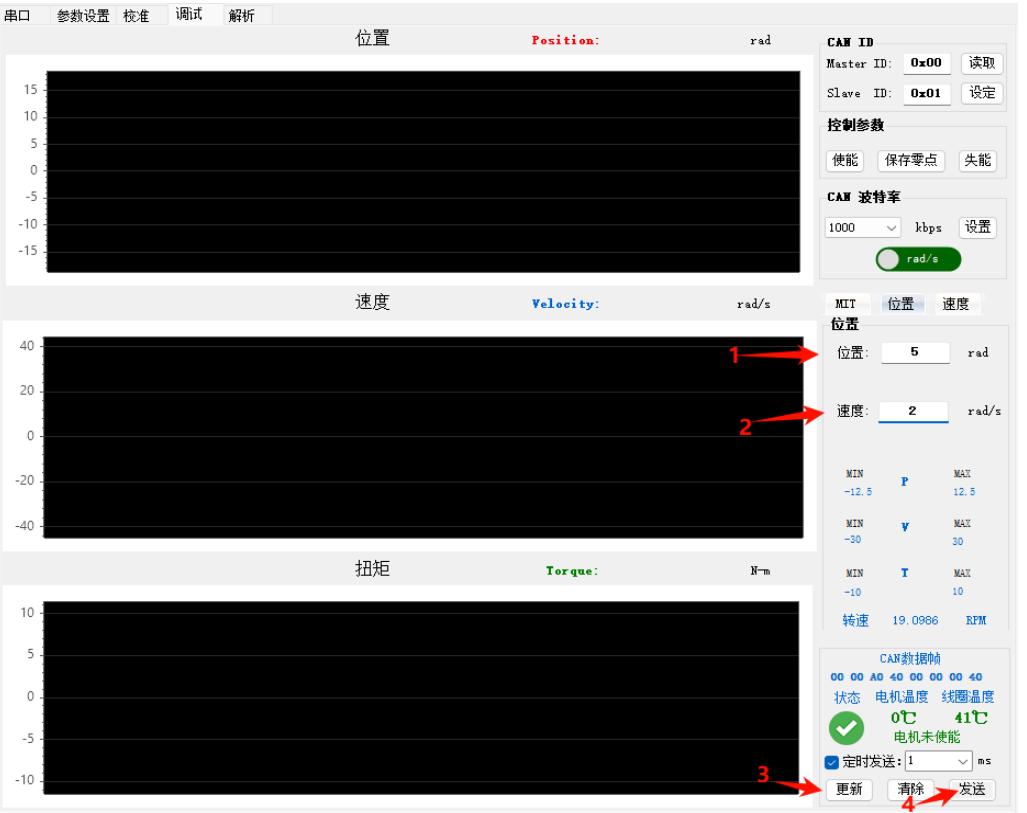


设置参数前，需注意电机的初始位置，以此为参考对参数进行设置。

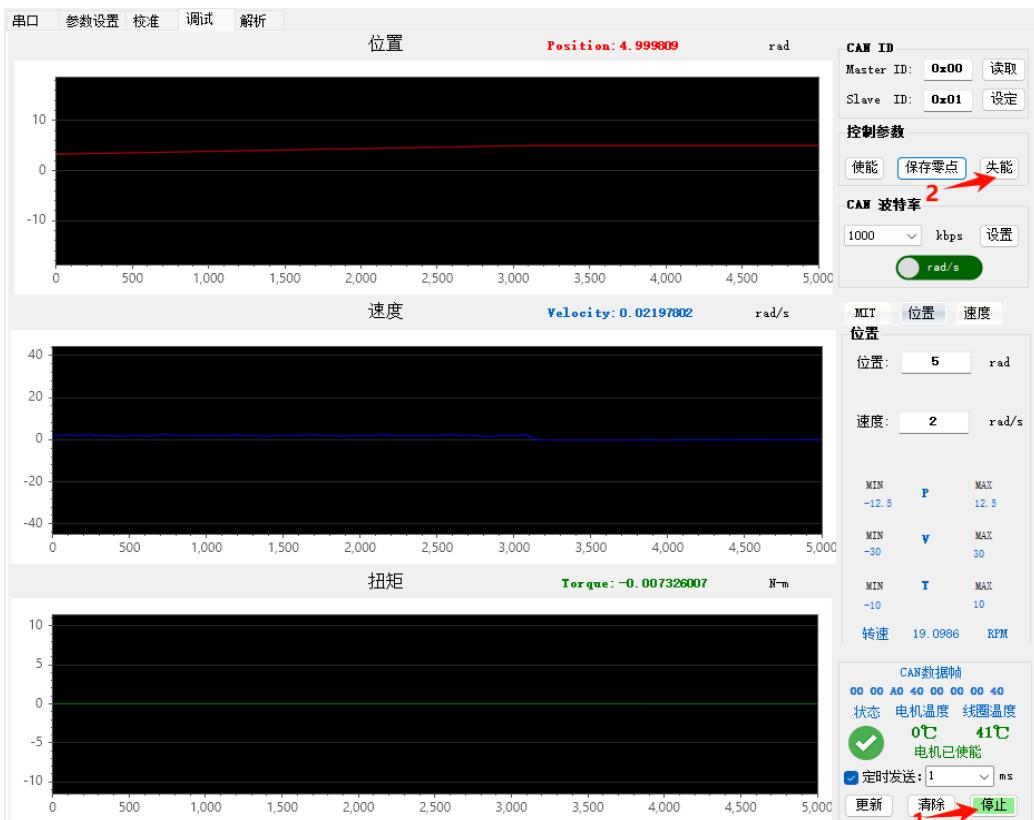


在“控制参数”中给定期望速度运转到指定位置，例如：在“控制参数”中位置设定为5rad，速度为2rad/s。并勾选“定时发送”框，依次点击“更新”按钮和“发送”按钮，可在调试界面查看参数曲线变化图。

**⚠ 注意固定电机。**



根据调试需求，需要修改控制参数查看调试变化，在原界面直接对参数进行修改，并保持勾选“定时发送”，点击“更新”按钮即可进行调试。



退出调试时，依次点击“停止”和“失能”按钮。驱动器红色灯亮起，表示退出电机模式。

#### 4.4.4 速度模式

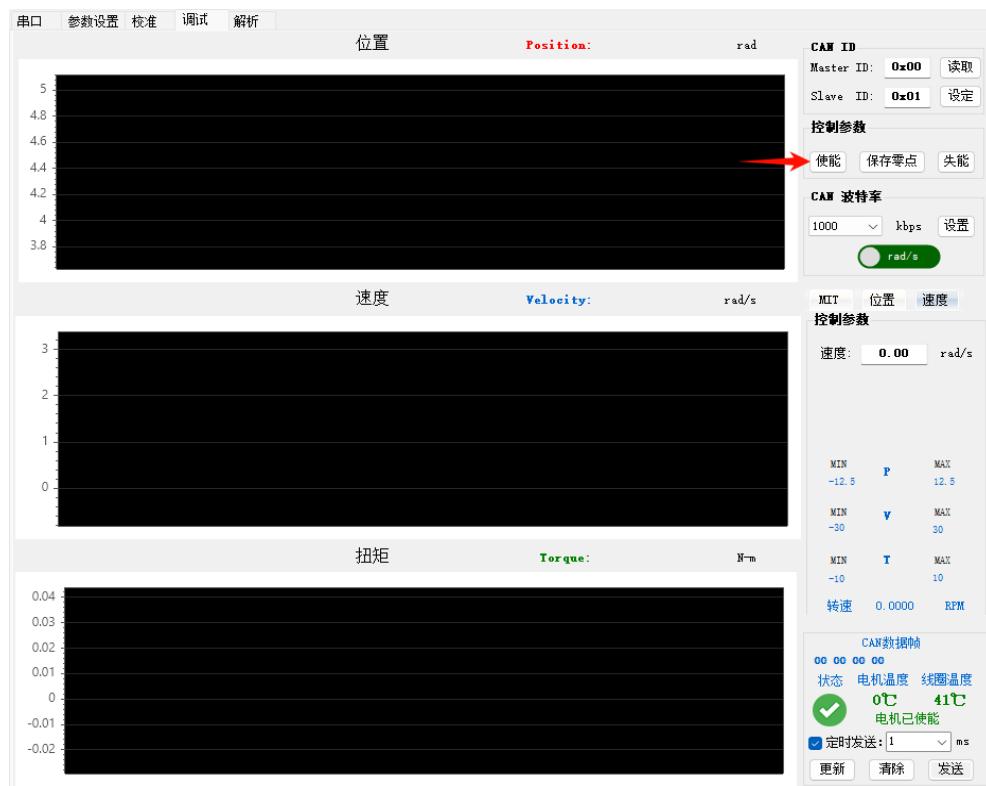
根据 4.4.1 节设置电机控制模式，在调试页面中选择对应的“速度”子标签卡。



确保 CAN ID 正确（可通过串口打印信息或者参数设置页面获取，也可以通过调试页面读取、设定按钮设定）

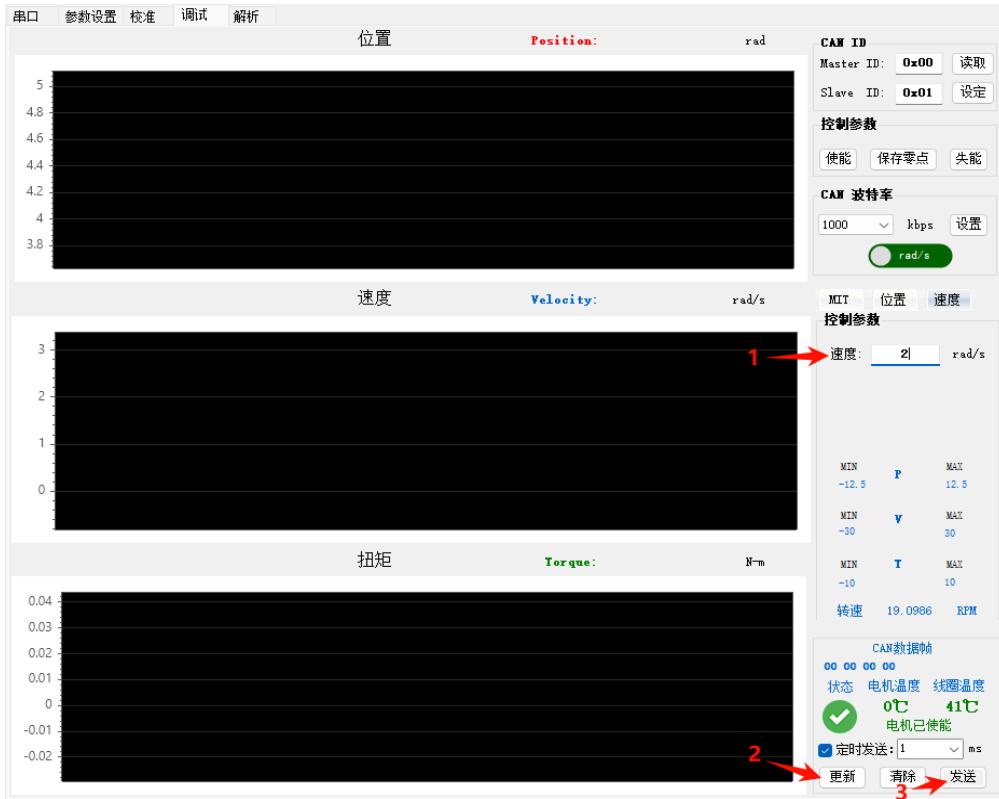


确认电机输入电源稳定，S-link 连接正常，并且电机处于速度模式时，在与上位机成功连接后，在电机模式栏点击“使能”按钮（电机上电默认进入使能状态）。此时驱动器绿色灯亮起，表示进入电机模式。



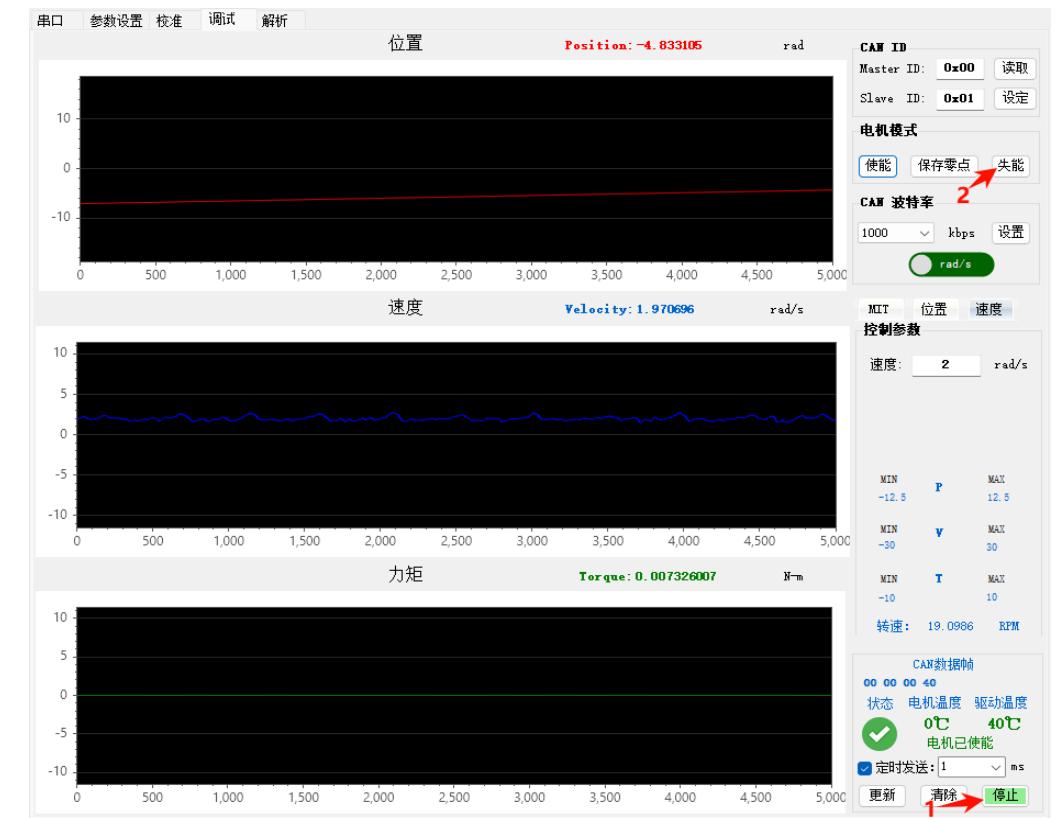
在“控制参数”中给定期望速度运转，例如：在“控制参数”中位速度设定为 2rad/s。并勾选“定时发送”框，依次点击“更新”按钮和“发送”按钮，可在调试界面查看参数曲线变化图。

**⚠ 注意固定电机。**



根据调试需求，需要修改控制参数查看调试变化，在原界面直接对参数进行修改，并保持勾选“定时发送”，点击“更新”按钮即可进行调试。

退出调试时，依次点击“停止”和“失能”按钮。驱动器红色灯亮起，表示退出电机模式。



#### 4.4.5 PWM 模式

PWM 脉冲控制允许输入频率 50Hz、脉宽行程 800-2200us 的 PWM 信号，可在上位机控制模式中切换速度和位置控制模式（参考 4.4.1 控制模式设置）。

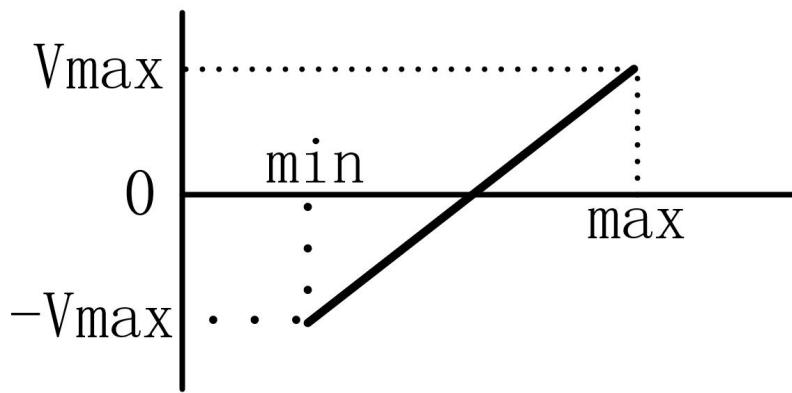
⚠ 在 PWM 与 CAN 同时存在的情况下，CAN 具有最高优先级，即此时只接收 CAN 控制命令，而忽略 PWM 控制。

##### 4.4.5.1 行程校准

确认电机输入电源稳定，PWM 连接正常，上电前将 PWM 输入推至最长脉宽档位，随后绿灯亮起，表示程序进入 PWM 输入校准功能，并在 3s 内采样最长脉宽时间待电机红灯和绿灯都灭时，表示长脉宽采集完成，进入下一环节。在 2s 内请将 PWM 输入推至最短脉宽档位之后绿灯再次亮起，此时将进入低脉宽采样等待红灯和绿灯再次都灭时，表示长短脉宽校准完成此后驱动器将数据写入驱动器中之后红灯闪烁 5s，请在此期间将 PWM 档位归中红灯闪烁完成后，将自动重启驱动器，至此校准完成。

##### 4.4.5.1 速度控制

确认电机输入电源稳定，PWM 连接正常，并且电机处于速度模式。

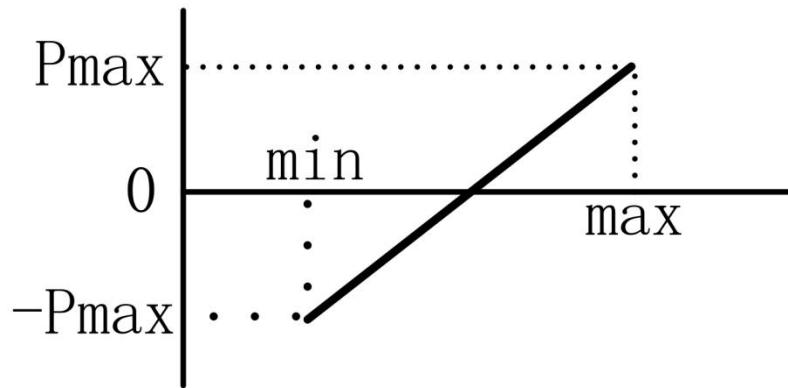


$V_{max}$  为驱动器中控制幅值设置的 VMAX 参数，Max 为输入 PWM 最大脉宽时间，Min 为输入 PWM 最小脉宽时间。



#### 4.4.5.1 位置控制

确认电机输入电源稳定，PWM 连接正常，并且电机处于位置速度模式。

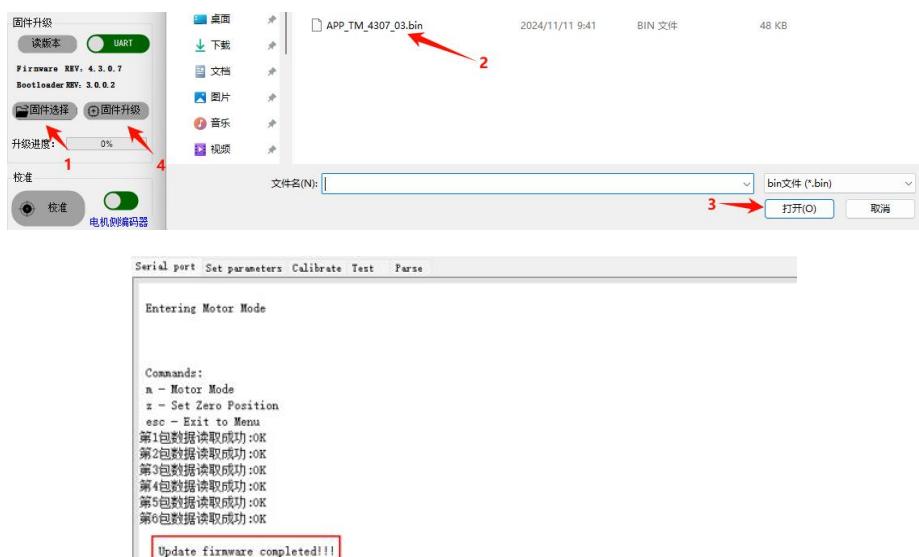


$P_{max}$  为驱动器中控制幅值设置的 PMAX 参数，此模式下速度限制为 VMAX，Max 为输入 PWM 最大脉宽时间，Min 为输入 PWM 最小脉宽时间。



## 4.5 固件升级

当固件有新增新功能，或有升级解决 BUG 版本后，用户可以通过串口实现升级，以解决问题，使用新功能。使用前需要连接上串口，然后点击“固件选择”，选择相应的固件，确认后再点击“固件升级”，等待升级进度条完成，也可通过串口界面观察是否完成升级。



⚠️ 请务必严格按照规定电压、电流、功率、温度使用。因违规操作本产品导致对人体造成伤害，或对驱动板及电机造成不可逆的损伤，我司将不承担任何法律责任。

## 5. 驱动板通讯协议及说明

电机协议为 can 协议，采用标准帧格式，固定波特率为 1Mbps，按功能可分为接收帧和反馈帧，接收帧为接收到的控制数据，用于实现对电机的命令控制；反馈帧为电机向上层控制器发送电机的状态数据。根据电机选定的不同模式，其接收帧帧格式定义以及帧 ID 各不相同，但各种模式下的反馈帧是相同的。

标准帧格式如下示：

Can ID bits	[10]-[8]	[7]-[0]
Field name (功能定义)	Control mode (控制模式)	Source node ID (从机 ID 号)

Control mode 有{0,1,2} 3 个特征值分别对应 3 种控制模式：

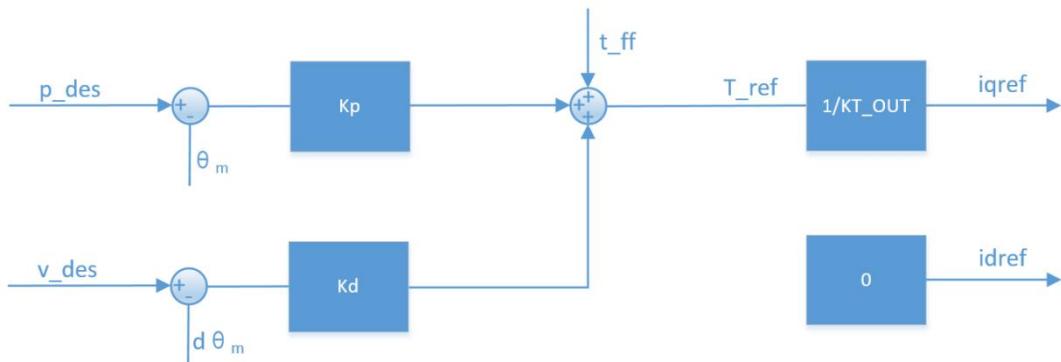
MIT 模式: 0;  
位置速度模式: 1;  
速度模式: 2。

### 通用 CAN 命令

命令	CAN 数据位
进入电机控制模式	0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0XFC
退出电机控制模式	0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0XFD
设置电机当前位置为零点	0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0XFE
清除错误	0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0XFB

## 5.1 MIT 模式控制模式及说明

MIT 模式能够灵活设定控制范围 ( $P_{MAX}, V_{MAX}, T_{MAX}$ )，驱动器将接收到的 CAN 数据转化成控制变量进行运算得到扭矩值作为电流环的电流给定，电流环根据其调节规律最终达到给定的扭矩电流。其控制示意框图如下。



根据 MIT 模式可以衍生出多种控制模式，如  $Kp=0, Kd \neq 0$  时，给定  $v_{des}$  即可实现匀速转动； $Kp=0, Kd=0$ ，给定  $t_{ff}$  即可实现给定扭矩输出。

⚠ 对位置进行控制时， $Kd$  不能赋 0，否则会造成电机震荡，甚至失控。

MIT 模式发送数据定义：

数据位	Data[0]	Data[1]	Data[2]	Data[3]
范围	0~0xff	0~0xff	0~0xff	0~0xff
对应变量	位置 15-8 位	位置 7-0 位	速度 11-4 位	速度 3-0 位 $Kp$ 11-8 位

数据位	Data[4]	Data[5]	Data[6]	Data[7]
范围	0~0xff	0~0xff	0~0xff	0~0xff
对应变量	Kp 7-0 位	Kd 11-4 位	Kd 3-0 位 t_ff 11-8 位	t_ff 7-0 位

其中，位置范围-12.5~12.5 代表-12.5~12.5rad;

其中，速度范围-200~200 代表-200~200r/s;

其中，Kp 范围 0~500 代表 0~500N/r;

其中，Kd 范围 0~5 代表 0~5N\*s/r;

其中，t\_ff 范围-10~10 代表-10~10N-m。

```

void ctrl_motor(CAN_HandleTypeDef hcan,uint16_t id, float _pos, float _vel, float _KP, float _KD,
float _torq){

    uint16_t pos_tmp,vel_tmp,kp_tmp,kd_tmp,tor_tmp;
    pos_tmp = float_to_uint(_pos, P_MIN, P_MAX, 16);
    vel_tmp = float_to_uint(_vel, V_MIN, V_MAX, 12);
    kp_tmp = float_to_uint(_KP, KP_MIN, KP_MAX, 12);
    kd_tmp = float_to_uint(_KD, KD_MIN, KD_MAX, 12);
    tor_tmp = float_to_uint(_torq, T_MIN, T_MAX, 12);

    hcan->pTxMsg->StdId = id;
    hcan->pTxMsg->IDE = CAN_ID_STD;
    hcan->pTxMsg->RTR = CAN_RTR_DATA;
    hcan->pTxMsg->DLC = 0x08;
    hcan->pTxMsg->Data[0] = (pos_tmp >> 8);
    hcan->pTxMsg->Data[1] = pos_tmp;
    hcan->pTxMsg->Data[2] = (vel_tmp >> 4);
    hcan->pTxMsg->Data[3] = ((vel_tmp&0xF)<<4)|(kp_tmp>>8);
    hcan->pTxMsg->Data[4] = kp_tmp;
    hcan->pTxMsg->Data[5] = (kd_tmp >> 4);
    hcan->pTxMsg->Data[6] = ((kd_tmp&0xF)<<4)|(tor_tmp>>8);
    hcan->pTxMsg->Data[7] = tor_tmp;

    HAL_CAN_Transmit(hcan, 100);
}

```

发包时所有的数都要经以下函数转化成整型数之后再发给电机。

```

int float_to_uint(float x, float x_min, float x_max, unsigned int bits){
/// Converts a float to an unsigned int, given range and number of bits ///
    float span = x_max - x_min;
    if(x < x_min) x = x_min;
    else if(x > x_max) x = x_max;

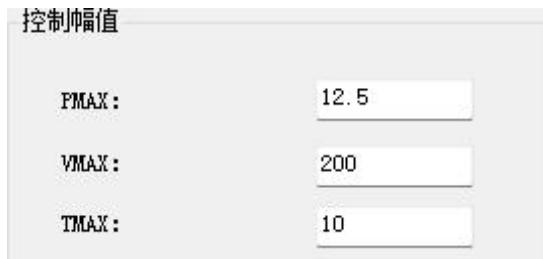
```

```

    return (int) ((x - x_min)*((float)((1<<bits)/span)));
}

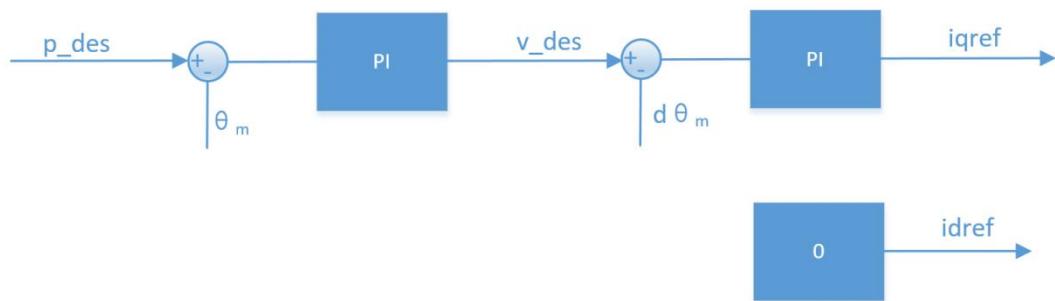
```

转换函数需确定两个等比例转换的最大最小值，这两个值在参数设定页面查询，其中 KP、KD 的最大最小值默认分别为 0.0~500.0、0.0~5.0。Pos、Vel、Torque 分别预设为 ±12.5、±200、±10，这三个参数可根据电机的实际参数进行调整。发送控制命令时，一定要与设定值保持一致。



## 5.2 位置速度模式控制模式及说明

位置速度模式是采用三环串联控制的模式，位置环作为最外环，其输出作为速度环的给定，而速度环的输出作为内环电流环的给定，用以控制实际的电流输出，其控制示意框图见下图：



$p_{des}$  为控制的目标位置， $v_{des}$  是用来限定运动过程中的最大绝对速度值。

位置速度模式如使用上位机推荐的控制参数控制，可以达到较好的控制精度，控制过程相对柔顺，但响应时间相对较长。可配置的相关参数除  $v_{des}$  外，另有加/减速度进行设定，如控制过程中产生额外的震荡可提高加/减速度。

⚠:  $p_{des}$ ,  $v_{des}$  单位分别为 rad 和 rad/s，数据类型为 float，阻尼因子必须设置为非 0 的正数，可参考速度模式的注意事项。

位置速度模式发送数据定义：

数据位	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
范围	0~0xff	0~0xff	0~0xff	0~0xff	0~0xff	0~0xff	0~0xff	0~0xff
对应变量	位置 7-0 位 位 15-8 位	位 15-8 位 位置 23-16 位	位 23-16 位 位置 31-24 位	位 31-24 位 速度 7-0 位	速度 7-0 位 速度 15-8 位	速度 15-8 位 速度 23-16 位	速度 23-16 位 速度 31-24 位	

其中，位置为 float 型，范围-12.5~12.5 代表-12.5~12.5rad;  
 其中，速度为 float 型，范围-200~200 代表-200~200r/s。

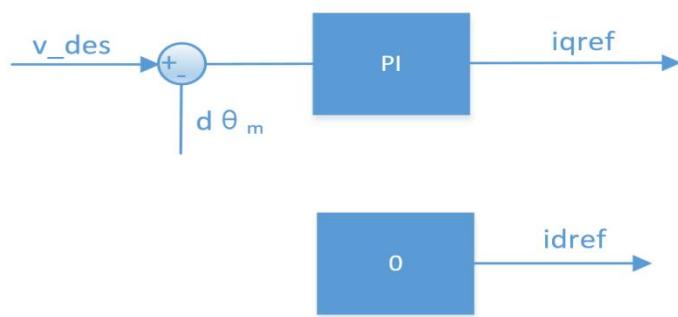
```
void ctrl_motor2(CAN_HandleTypeDefDef* hcan,uint16_t id, float _pos, float _vel){
    uint8_t *pbuf,*vbuf;
    pbuf=(uint8_t*)&_pos;
    vbuf=(uint8_t*)&_vel;

    hcan->pTxMsg->StdId = id;
    hcan->pTxMsg->IDE = CAN_ID_STD;
    hcan->pTxMsg->RTR = CAN_RTR_DATA;
    hcan->pTxMsg->DLC = 0x08;
    hcan->pTxMsg->Data[0] = *pbuf;
    hcan->pTxMsg->Data[1] = *(pbuf+1);
    hcan->pTxMsg->Data[2] = *(pbuf+2);
    hcan->pTxMsg->Data[3] = *(pbuf+3);
    hcan->pTxMsg->Data[4] = *vbuf;
    hcan->pTxMsg->Data[5] = *(vbuf+1);
    hcan->pTxMsg->Data[6] = *(vbuf+2);
    hcan->pTxMsg->Data[7] = *(vbuf+3);

    HAL_CAN_Transmit(hcan, 100);
}
```

### 5.3 速度模式控制模式及说明

速度模式能让电机稳定运行在设定的速度，其控制示意框图如下：



**⚠:**  $v_{des}$  单位为 rad/s，数据类型为 float，如需使用上位机自动计算参数，则需要设置阻尼因子为非 0 正数，通常情况下取值在 2.0~10.0，过小的阻尼因子会带来速度的震荡以及较大的过冲，过大的阻尼因子则会带来较长的上升时间，推荐的设定值为 4.0。

---

速度模式发送数据定义：

数据位	Data[0]	Data[1]	Data[2]	Data[3]
范围	0~0xff	0~0xff	0~0xff	0~0xff
对应变量	速度 7-0 位	速度 15-8 位	速度 23-16 位	速度 31-24 位

其中，速度为 float 型，范围-200~200 代表-200~200r/s;

```
void ctrl_motor3(CAN_HandleTypeDef* hcan,uint16_t id, float _vel){
    uint8_t *vbuf;
    vbuf=(uint8_t*)&_vel;

    hcan->pTxMsg->StdId = id;
    hcan->pTxMsg->IDE = CAN_ID_STD;
    hcan->pTxMsg->RTR = CAN_RTR_DATA;
    hcan->pTxMsg->DLC = 0x04;
    hcan->pTxMsg->Data[0] = *vbuf;
    hcan->pTxMsg->Data[1] = *(vbuf+1);
    hcan->pTxMsg->Data[2] = *(vbuf+2);
    hcan->pTxMsg->Data[3] = *(vbuf+3);

    HAL_CAN_Transmit(hcan, 100);
}
```

## 5.4 CAN 反馈报文协议

反馈帧 ID 由上位机设置（Master ID），默认为 0，主要反馈电机的位置，速度和扭矩信息，其帧格式定义为：

数据位	Data[0]	Data[1]	Data[2]	Data[3]
范围	0~0xff	0~0xff	0~0xff	0~0xff
对应变量	ERR 7-4 位 CAN ID 3-0 位	位置 15-8 位	位置 7-0 位	速度 11-4 位

数据位	Data[4]	Data[5]	Data[6]	Data[7]
范围	0~0xff	0~0xff	0~0xff	0~0xff
对应变量	速度 3-0 位 扭矩 11-8 位	扭矩 7-0 位	驱动温度 7-0 位	电机温度 7-0 位

其中，ID 为 int8 型，取 CAN\_ID 的低 8 位；

其中，ERR 为 int8 型，对应编码为：

- 0——失能；
- 1——使能；
- 8——超压；
- 9——欠压；
- A——过电流；
- B——MOS 过温；
- C——电机线圈过温；
- D——通讯丢失；
- E——过载。

其中，范围-12.5~12.5 代表-12.5~12.5rad；

其中，范围-200~200 代表-200~200r/s；

其中，范围-10~10 代表-10~10N·m；

其中，驱动温度为 int8 型，范围-128-127°C；

其中，驱动温度为 int8 型，范围-128-127°C。

```
void motor_receive(float *motor_pos, float *motor_spd, float *motor_t, int_8 *temp1, int_8
*temp2, int_8 *error, rx_message) {
    int_8 err_int = (rx_message)->Data[0] >> 4;
    int16_t pos_int = (rx_message)->Data[1] << 8 | (rx_message)->Data[2];
    int16_t spd_int = (rx_message)->Data[3] << 4 | (rx_message)->Data[4] >> 4;
    int16_t t_int = ((rx_message)->Data[4] & 0xF) << 8 | (rx_message)->Data[5];
    &motor_error = (rx_message)->Data[0] >> 4; // 电机故障码
    &motor_pos = uint_to_float(pos_int, P_MIN, P_MAX, 16); // 电机位置
    &motor_spd = uint_to_float(spd_int, V_MIN, V_MAX, 12); // 电机速度
    &motor_t = uint_to_float(t_int, T_MIN, T_MAX, 12); // 电机扭矩
    &motor_temp1 = (rx_message)->Data[6]; // 驱动温度
    &motor_temp2 = (rx_message)->Data[7]; // 电机温度
}
```

收包时位置、速度和扭矩要经以下函数转化成浮点型。

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

转换函数需确定两个等比例转换的最大最小值，这两个值在参数设定页面查询，其中 KP、KD 的最大最小值默认分别为 0.0~500.0、0.0~5.0。Pos、Vel、Torque 分别预设为±12.5、±200、±10，这三个参数可根据电机的实际参数进行调整。发送控制命令时，一定要与设定值保持一致。

控制幅值

PMAX:	12.5
VMAX:	200
TMAX:	10

## 5.5 CAN 口控制命令实例

帧格式：标准帧-数据帧（以电机 ID 为 0x01 示例）

模式	ID	DATA	说明
MIT 速度	00 01	7F FF 83 00 00 00 47 FF	Kd 设置为 0.005 速度设置为 6rad/s
	00 01	7F FF 7C E0 00 00 47 FF	Kd 设置为 0.005 速度设置为 -6rad/s
MIT 位置	00 01	94 7A 7F F0 01 00 47 FF	Kp 设置为 0.123 Kd 设置为 0.005 电机转动到 2rad
	00 01	6B 84 7F F0 01 00 47 FF	Kp 设置为 0.123 Kd 设置为 0.005 电机转动到 -2rad
MIT 力矩	00 01	7F FF 7F F0 00 00 08 05	力矩设置为 0.03N·m
	00 01	7F FF 7F F0 00 00 07 F9	力矩设置为 -0.03N·m
位置速度模式	01 01	00 00 40 40 00 00 80 3F	3rad 1rad/s
	01 01	00 00 40 C0 00 00 80 3F	-3rad 1rad/s
速度模式	02 01	00 00 00 40	2rad/s
	02 01	00 00 00 C0	-2rad/s