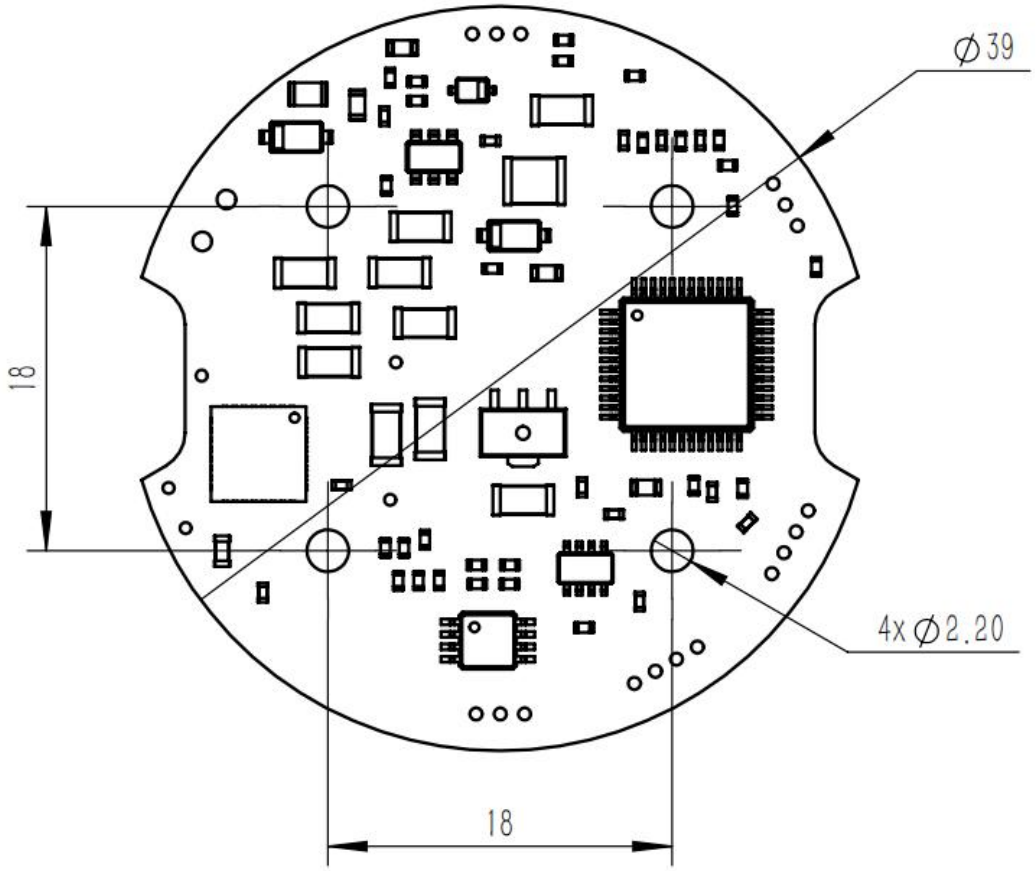
SensoredFoc24V-GD32

驱动器使用说明书

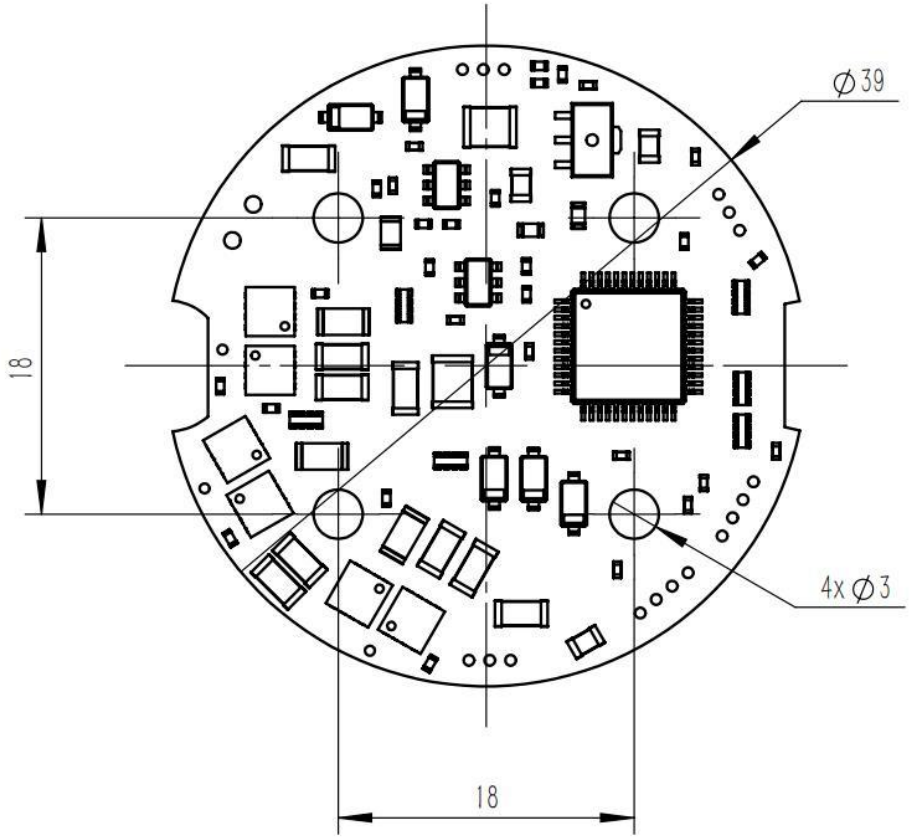
目录

一、概况

1、驱动器尺寸参数

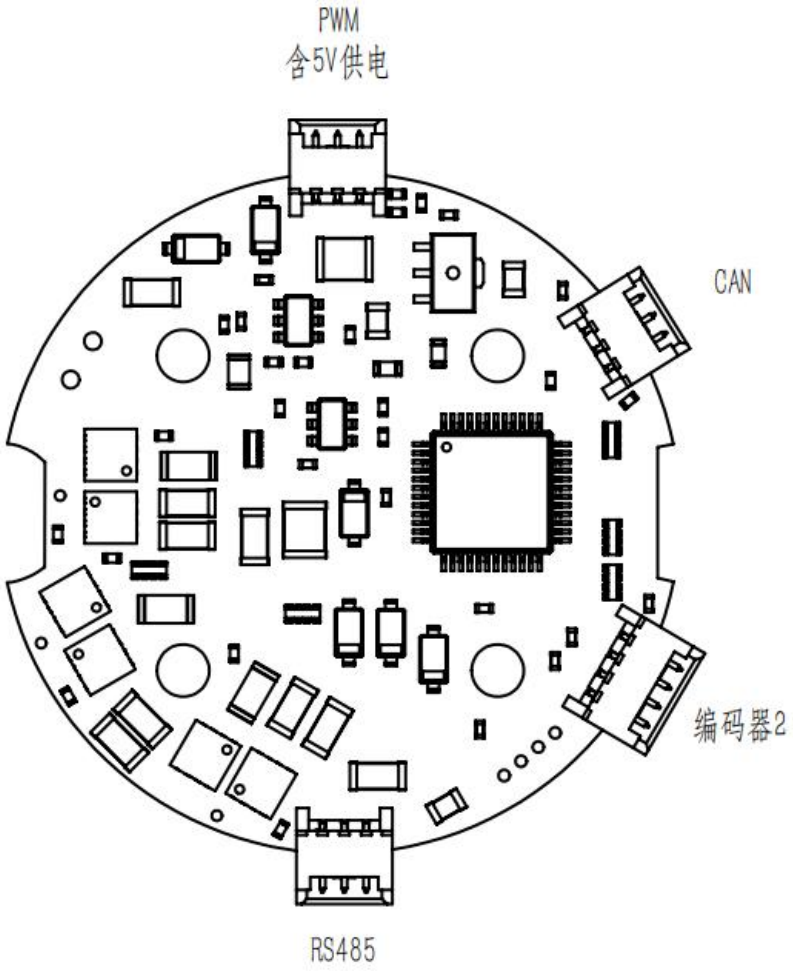


标准版



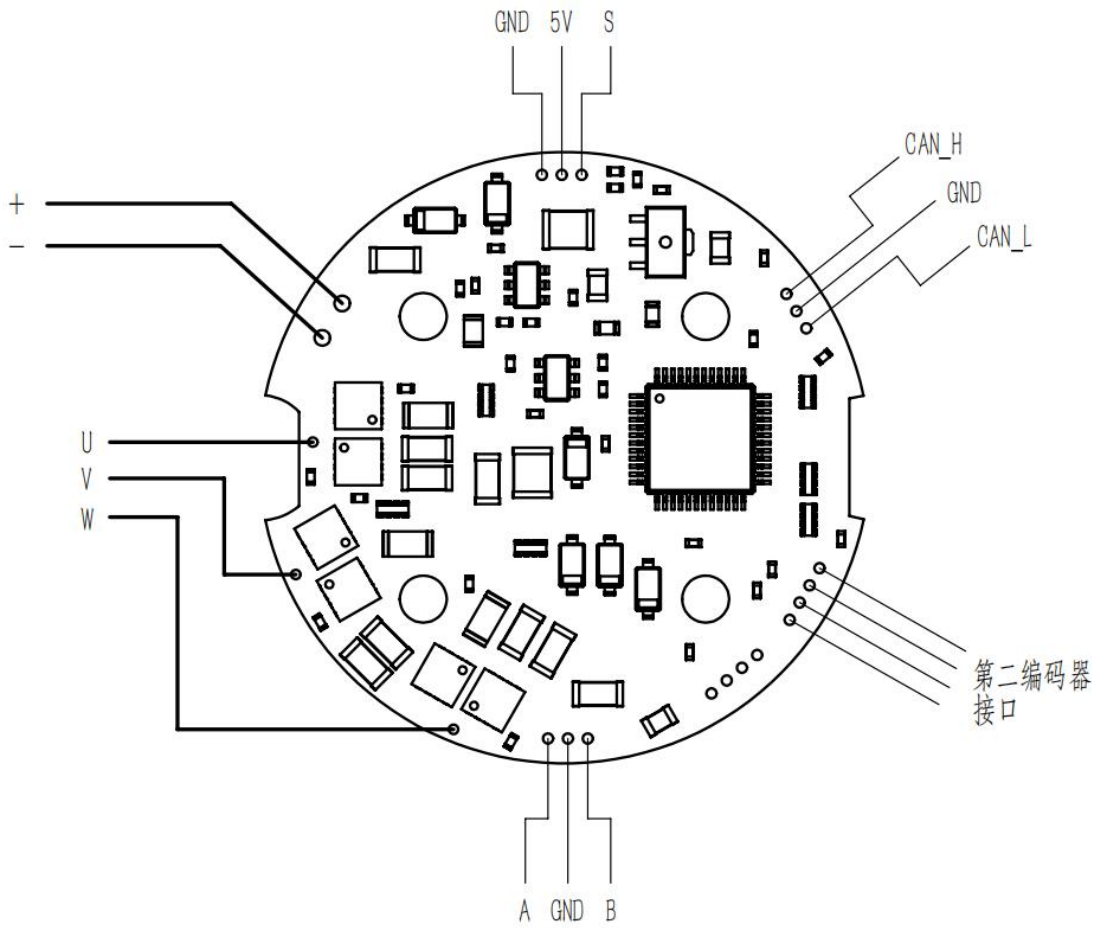
高功率版

2、驱动器接口定义



*标准版与高功率版接口定义一致

3、驱动器管脚定义



*标准版与高功率版管脚定义一致

4、电气参数

参数	数值	说明
额定电压	0-26V	
额定电流	（2A）3A	
峰值电压	30V	
峰值电流	（4A）6A	*工作电流设置大于（2A）3A 需要增加散热片
单圈位置精度	14 位	
主控芯片	GD32	
接口参数	说明	
PWM	根据配置模式（位置/速度/力矩）控制电机旋转，有 5V 供电	
ENCODER2	第二路编码器接口，用于组建绝对位置	

RS485	驱动器调参接口，固件升级接口
CAN	总线控制接口，兼容 MIT 控制协议

*括号中的参数为标准版参数

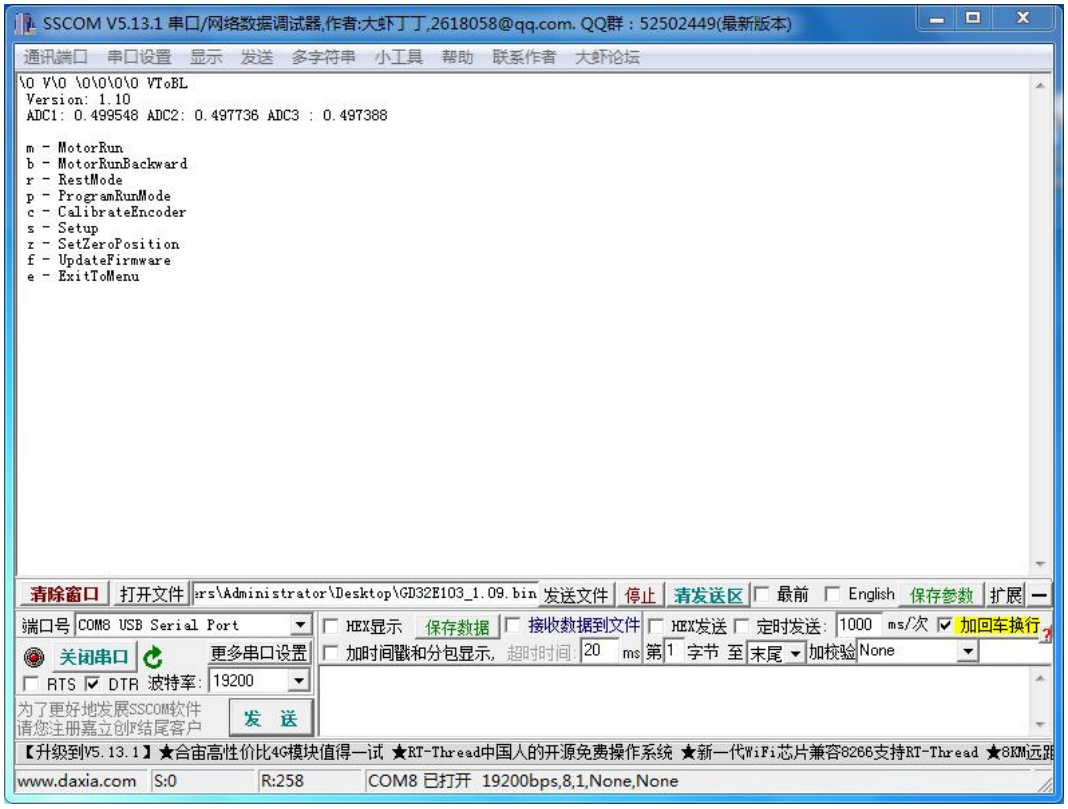
二、通过 RS485 配置说明

RS485 波特率固定 19200， 8 位数据，无校验，1 位停止位

所有命令以 0x0D 0x0A 结尾（回车—换行）

通过 RS485 接口写入的所有数据均保存在芯片内，掉电不丢失

1、一级目录介绍



m - Motor Run Mode

电机正向工作

速度模式 - 32RPM

力矩模式 - 1NM

位置模式 - 锁定在当前位置

b - Motor Run Backward Mode

电机反向旋转

速度模式 - （-32RPM）

力矩模式 - 1NM

位置模式 - 锁定在当前位置

r - Rest Mode

电机停止

p - Program Run Mode

设定运行模式

c - Calibrate Encoder

标定电机编码器 1

s - Setup

驱动器参数设置

z - Set Zero Position

设定电机位置零位

f -UpdateFirmware

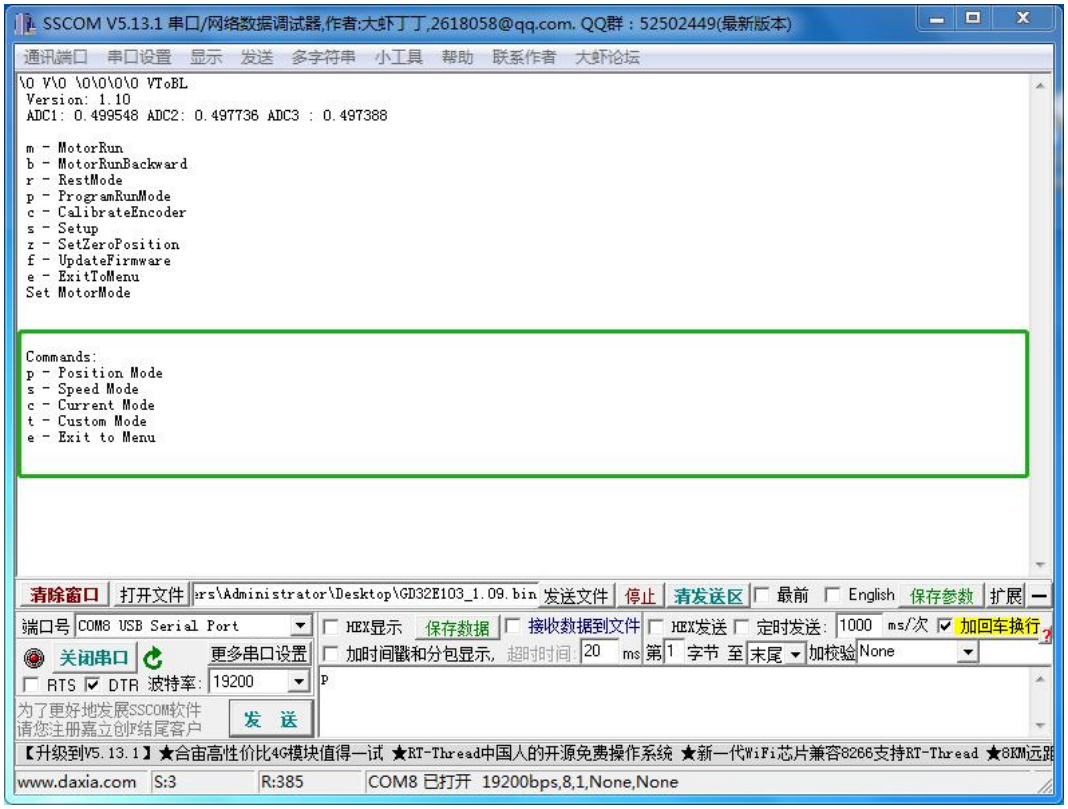
升级系统固件

e - Exit to Menu

退出菜单

2、二级目录介绍

1）p - Program Run Mode



p - Position Mode

位置模式

s - Speed Mode

速度模式

c - Current Mode

力矩模式

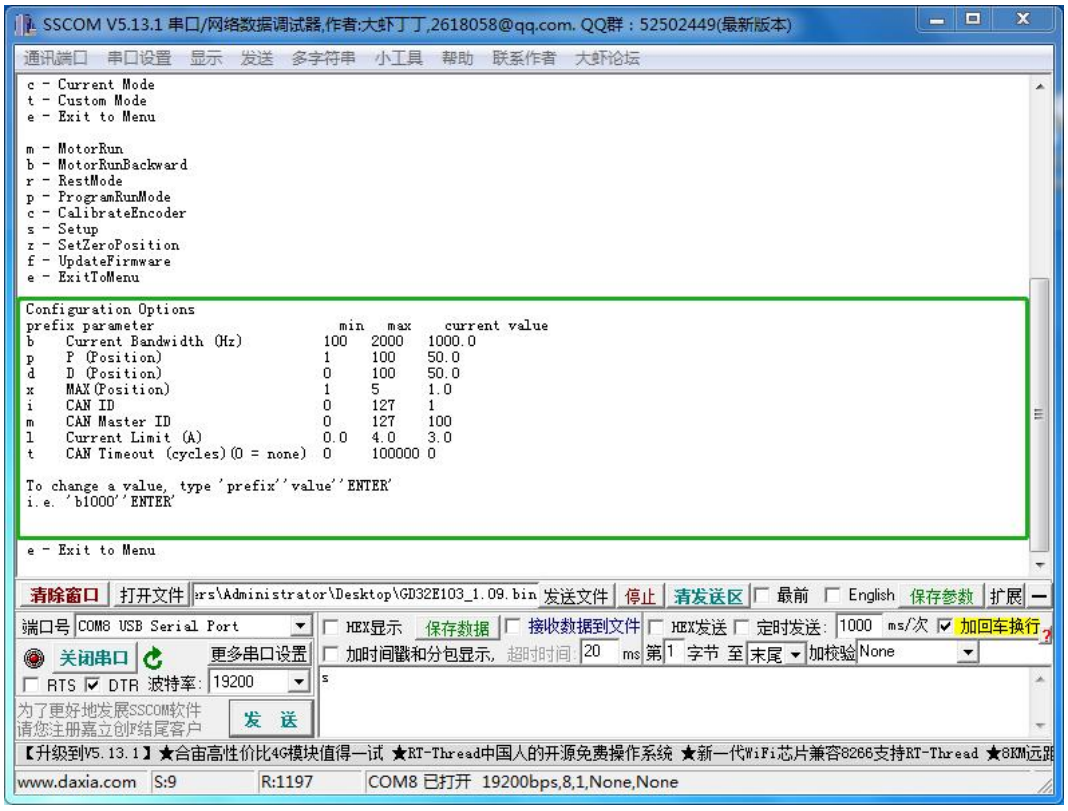
t - Custom Mode

*定制模式（**零售版此模式无效**）

e - Exit to Menu

返回上级菜单

2）s - Setup



b - Current Bandwidth (Hz)

指定电流环带宽 范围 100 -2000，单位 Hz，这个值越大，电机动态响应越高

p - P (Position)

PWM 控制位置环时的 P 参数,P 越大电机硬度越大

d - D (Position)

PWM 控制位置环时的 D 参数,D 越大电机阻尼越大

***注意以上两项在 Can 控制时无效,只针对 PWM 控制**

x - MAX(Position) Pmax

Can 控制的最大圈数

无末端编码器时 Pmax = 1 时 Can 控制的最大圈数为+-1 圈

有末端编码器时 Pmax = 1 Can 控制的最大圈数为+- 0.5 圈,并且 Pmax 需固定配置为 1

i - CAN ID

本机 can ID 0---127

m - CAN Master ID

主机 can ID 0---127

Can 总线上的 ID 不可重复，**因此注意不要把设备 ID 设置成相同的值**

l - Current Limit (A)

最大电流 0 - 4A (标准版) 0-6A(高功率版)

t - CAN Timeout (cycles)(0 = none)

can 通讯超时时间设置，单位 ms

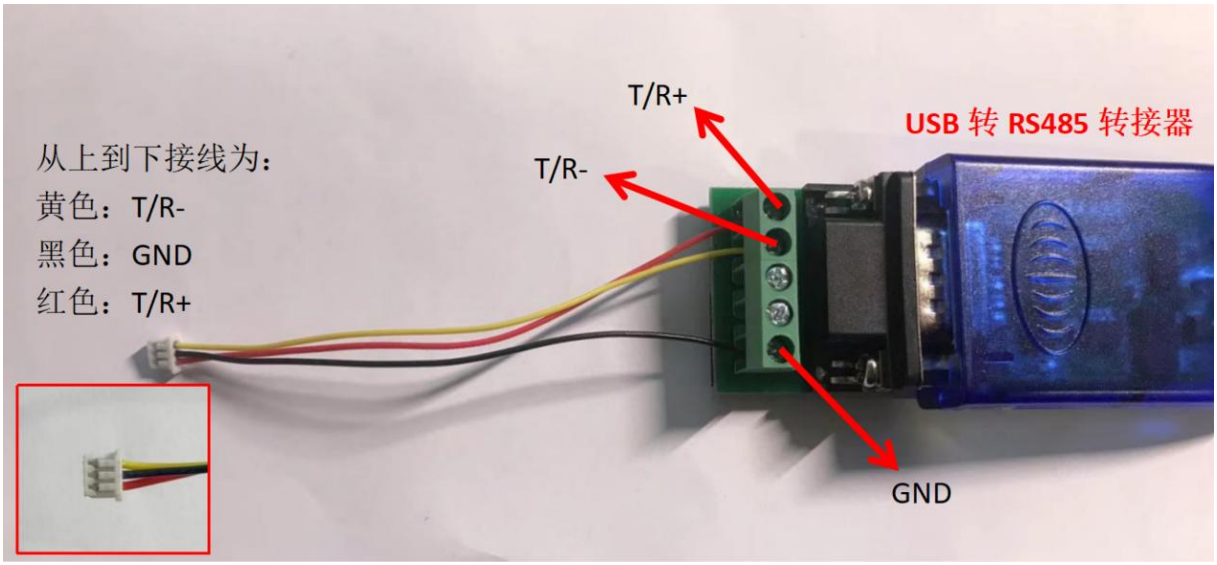
超过设定时间后，设备自动停止电机动作，此功能用于防止通讯故障引起的电机误动作

e - Exit to Menu

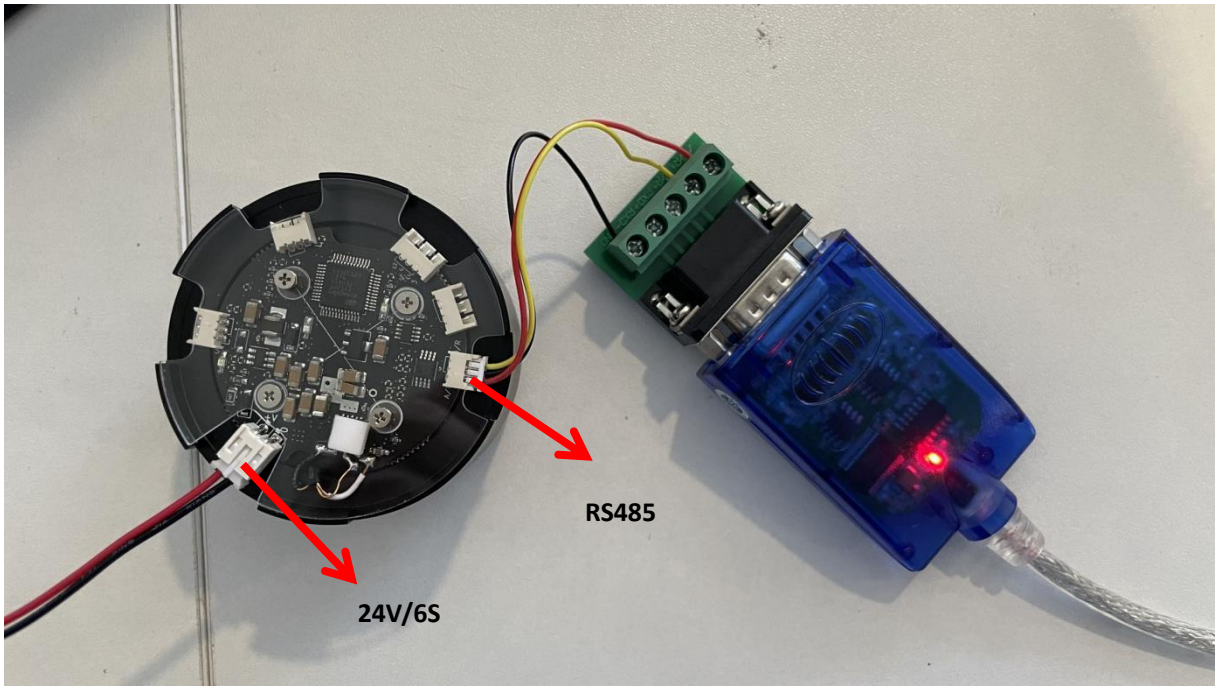
返回上级菜单

3、串口助手配置驱动器的接线方法

1）准备 USB 转 RS485 转换器并接线



2）链接驱动器



3）24V 通电，通过串口助手配置 CAN_ID 和电流等相关参数

三、驱动方式介绍

1、PWM 控制模式介绍

PWM 信号频率 50-1000Hz，脉宽 0.8ms-2ms，1.45ms-1.55ms 为中位死区，电机不工作

根据电机工作模式

1）恒定扭矩控制模式

脉宽时间 = t 单位 ms,t>1.55 正方向工作，t<1.45 反方向工作

正转：

Torque = (t - 1.55) * 2.5 单位 NM

反转：

Torque = (1.45 - t) * 2.5 单位 NM

2）速度控制模式

脉宽时间 = t 单位 ms,t>1.55 正方向工作，t<1.45 反方向工作

正转：

Speed = (t - 1.55) * 1000 单位 RPM

反转：

Speed = (1.45 - t) * 1000　　单位 RPM

3）位置控制模式

始终保持位置，电机持续工作

Position= (t - 1.5) * 1.952　单位 圈

Position 符号代表电机偏转方向

PWM 信号有效时，电机只响应 PWM 信号

PWM 信号无效时，电机响应 CAN 命令

2、CAN 总线控制模式介绍

CAN 速率　固定 1Mbps

CAN 从机地址：由串口指定 slave addr ，代表本机地址

CAN 主机地址：由串口指定 master addr，代表控制端地址

CAN 工作模式为从机接收---应答，从机不主动往总线上发送数据，只有当接收到数据时，才应答主机，主控制器主动发送数据到所有的从设备

CAN 数据帧格式：11 位 ID 的 CAN 标准帧

接收数据帧　长度 8Byte，分为四种数据帧

1） 控制电机

0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xfc

启动电机

0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xfd

停止电机

2）设定电机位置零点

0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xfe

3）切换电机运行模式

0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xf9 切换到力矩模式

0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xfA 切换到速度模式

0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xfB 切换到位置模式

4）电机运行参数设定

指定位置，16 位数，范围 0-65536，代表电机旋转到-12.5rad 到 12.5rad （-1*PMax 圈到+1*PMax 圈）范围内的指定位置，并保持

计算公式 position = (p - 32768)/32768*12.5*0.5*Pmax 单位 rad

Byte0 指定位置 p 高 8 位

Byte1 指定位置 p 低 8 位

指定速度，12 位数，范围 0-4096，指定电机转速

计算公式 speed = (v-2048)/2048*65 单位 rad/s

Byte2 指定速度 v 高 8 位

Byte3(高四位) 指定速度 v 低 4 位

设定位置环 PD 控制 P 值，12 位数，范围 0-4096

计算公式 kp =set_kp/4096 * 500

Byte3(低四位) 设定 Kp 高 4 位

Byte4 设定 Kp 低 8 位

设定位置环 PD 控制 D 值，12 位数，范围 0-4096

计算公式 kd =set_kd/4096 * 5

Byte5 设定 Kd 高 8 位

Byte6(高四位) 设定 Kd 低 4 位

指定扭矩，12 位数，范围 0-4096，指定电机恒定扭矩

计算公式 torque = (t-2048)/2048*4 单位安培

Byte6 (低四位) 指定扭矩 t 高 4 位

Byte7 指定扭矩 t 低 8 位

5）CAN 应答

CAN 应答帧 ，长度 6 Byte

Byte 0 本机 ID

Byte 1 当前位置 p 高 8 位

Byte 2 当前位置 p 低 8 位

Byte 3 当前速度 v 高 8 位

Byte 4（高 4 位） 当前速度 v 低 4 位

Byte 4（低 4 位） 当前扭矩 t 高 4 位

Byte 5 当前扭矩 t 低 8 位

计算公式

计算公式:

torque = (t-2048)/2048*4 单位安培

speed = (v-2048)/2048*65 单位 rad/s

position = (p - 32768)/32768*12.5*0.5*Pmax 单位 rad

a. 恒定扭矩控制模式

脉宽时间 = t 单位 ms,t>1.55 正方向工作，t<1.45 反方向工作

正转：

Torque = (t - 1.55) * 2.5 单位 NM

反转：

Torque = (1.45 - t) * 2.5 单位 NM

b. 速度控制模式

脉宽时间 = t 单位 ms,t>1.55 正方向工作，t<1.45 反方向工作

正转：

Speed = (t - 1.55) * 1000 单位 RPM

反转：

Speed = (1.45 - t) * 1000 单位 RPM

c. 位置控制模式

始终保持位置，电机持续工作

Position= (t - 1.5) * 1.952 单位 圈

Position 符号代表电机偏转方向

3、CAN 总线控制模式介绍及例程

```
uint16_t  float_to_uint(float v,float v_min,float v_max,uint32_t width)
{
    float temp;
```

```
    int32_t utemp;
    temp = ((v-v_min)/(v_max-v_min))*((float)width);
    utemp = (int32_t)temp;
    if(utemp < 0)
        utemp = 0;
    if(utemp > width)
        utemp = width;
    return utemp;
}
Void CanCmdSend(void)
{
    Switch(CanMode)
    {
    Case 0:  //Motor Run
        g_transmit_message.tx_data[0]==0xFF;
        g_transmit_message.tx_data[1]==0xFF;
        g_transmit_message.tx_data[2]==0xFF;
        g_transmit_message.tx_data[3]==0xFF;
        g_transmit_message.tx_data[4]==0xFF;
        g_transmit_message.tx_data[5]==0xFF;
        g_transmit_message.tx_data[6]==0xFF;
        g_transmit_message.tx_data[7]==0xFC;
        CanSend();
        Break;
    Case 1:   //Motor Rest
        g_transmit_message.tx_data[0]==0xFF;
        g_transmit_message.tx_data[1]==0xFF;
        g_transmit_message.tx_data[2]==0xFF;
        g_transmit_message.tx_data[3]==0xFF;
        g_transmit_message.tx_data[4]==0xFF;
        g_transmit_message.tx_data[5]==0xFF;
```

```
        g_transmit_message.tx_data[6]==0xFF;
        g_transmit_message.tx_data[7]==0xFD;
        CanSend();
        Break;
    Case 2:   //Set Zero Position(temp)
        g_transmit_message.tx_data[0]==0xFF;
        g_transmit_message.tx_data[1]==0xFF;
        g_transmit_message.tx_data[2]==0xFF;
        g_transmit_message.tx_data[3]==0xFF;
        g_transmit_message.tx_data[4]==0xFF;
        g_transmit_message.tx_data[5]==0xFF;
        g_transmit_message.tx_data[6]==0xFF;
        g_transmit_message.tx_data[7]==0xFE;
        CanSend();
        Break;
    Case 3:
        s_p_int = float_to_uint(f_position, P_MIN, P_MAX, 65535);
        s_v_int = float_to_uint(f_velocity, V_MIN, V_MAX, 4096);
        s_Kp_int = float_to_uint(f_kp, 0 , KP_MAX, 4096);
        s_Kd_int = float_to_uint(f_kd, 0 , KD_MAX, 4096);
       s_c_int = float_to_uint(f_current, -C_MAX, C_MAX, 4096);
        g_transmit_message.tx_data[0] = s_p_int>>8;
        g_transmit_message.tx_data[1] = s_p_int&0xFF;
        g_transmit_message.tx_data[2] = s_v_int>>4;;
        g_transmit_message.tx_data[3] = ((s_v_int&0xF)<<4) + (s_Kp_int >>8);
        g_transmit_message.tx_data[4] = s_Kp_int &0xFF;
        g_transmit_message.tx_data[5] = s_Kd_int>>4;
        g_transmit_message.tx_data[6] = ((s_Kd_int &0xF)<<4) + (s_c_int >>8);
        g_transmit_message.tx_data[7] = s_c_int&0xFF;;
        CanSend();
        Break;
```

```
        }
    }
```

4、第二编码器

第二编码器可实现掉电以后，末端位置即使改变，再次上电后也可以找回掉电前的位置。

第二编码器出厂已标定，请勿拆卸第二编码器电路板以及磁环。

***若拆卸第二编码器配件，须返厂重新标定。**

带第二编码器状态下,Pmax 需固定配置为 1