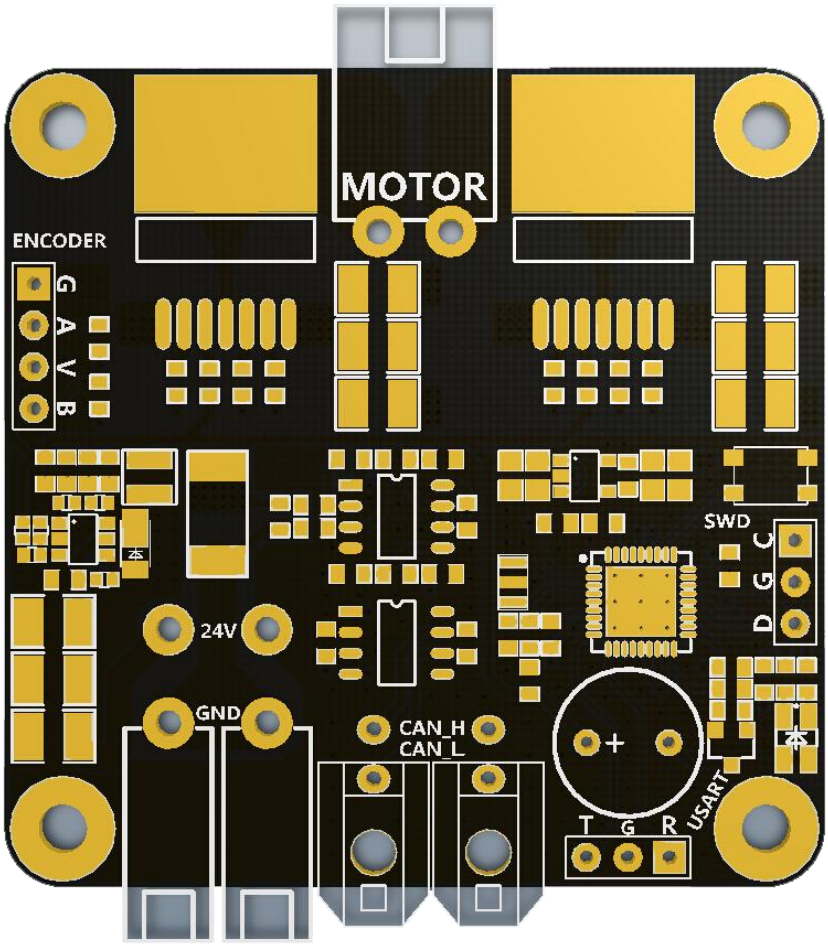
RoboModule

直流伺服电机驱动器 用户手册

V2.0 2015.06.05



免责声明和警告：

感谢您购买本产品。请严格遵守本手册的要求来使用您的产品。

请在首次使用本产品之前仔细本声明。一旦使用，即视为对本声明的全部内容表示认可和接受。在使用产品的过程中，使用者承诺对自己的行为及因此而产生的后果负责。使用者承诺是出于正当目的使用本产品，并且同意遵守本条款及 RoboModule 可能制定的任何相关政策或者准则。

鉴于本公司无法控制用户的具体使用、安装和改装以及使用不当等情况，由以上所造成的损害或损伤，本公司将不承担相应的损失及赔偿责任，相应的后果将由用户承担。因使用本产品而造成的直接或者间接的损失和伤害，本公司概不负责。

如果您在使用本产品的过程中遇到无法解决的问题，请从本公司的官方技术支持处获取帮助，联系方式见本手册最后一页。

对于本产品，需要特别注意以下几点：

1．请勿带电插拔串口线、码盘线、电机线、CAN 线等线材。如需插拔，请先让驱动器断开电源。因带电插拔以上所述线材造成的驱动器损坏，本公司对此不承担任何责任。请严格遵守本条操作，不可存在侥幸心理。

2．请务必确认，连接串口所用的串口线是 TTL 电平的，不可以是 RS232 电平的。因选错串口线造成的驱动器损坏，本公司对此不承担任何责任。

3．驱动器的电源接口已经具有防插反功能。但仍然需要注意电源不能接反。因电源接反导致的板子烧毁，本公司对此不承担任何责任。

4．如多个驱动器使用同一条 CAN 线进行通信，请参考“接口说明部分”的第 2 点所述内容，拆掉 CAN 总线上多余的 120 欧姆的电阻，否则 CAN 通信将不正常。

5．本驱动器不需要用户编写程序，所以 SWD 接口不需要用到，请不要连接 SWD 接口下载第三方程序。对于违反此条例造成驱动器无法使用的，本公司对此不承担任何责任。

产品综合介绍

感谢您使用本产品。

本产品是一款当前市面上性价比最高的一款直流伺服电机驱动器 ,它使用了黑色阻焊和表面沉金的四层板来制作 PCB，并采用了世界一流的 FUJI 贴片机进行无铅工艺的生产贴片。良好的加工工艺，是优良性能的基础。

它集成了开关电源降压电路、线性电源降压电路、功率驱动电路、电流采样电路、CAN 通信电路、蜂鸣器报警电路、LED 指示灯电路、STM32 最小系统电路等在一块 50mm*50mm 的 PCB 电路板上。体积小、质量轻、性能好。

它具有如下接口：电源接口、CAN 通信接口、增量式编码器数据采集接口、电机接口、TTL 串行通信接口、SWD 调试接口。

普通用户只需通过配套的“RoboModule 伺服电机驱动器调试软件”来进行电机的配置，即可完成所有的设置。不需要写一行代码就可以得到一个高效、稳定、合适的直流伺服电机驱动器。

结合 RoboModule 伺服电机驱动器调试软件，本驱动器具有如下七种工作模式：

1．PWM 模式 1（速度反馈），即控制电机的开环转动，PWM 占空比越大，转动越快，同时可以在调试软件窗口查看编码器所反馈的速度值的变化。
2．PWM 模式 2（电流反馈），即控制电机的开环转动，PWM 占空比越大，转动越快，同时可以在调试软件窗口查看流经电机的电流值的变化。
3．速度环模式，即可以控制电机精确的转动速度，同时可以查看实际电机的转动速度的曲线图。
4．位置环模式，即可以控制电机以某个指定的转动速度转动到某个指定的位置。
5．电机参数设置模式，即可以设置电机速度环和位置环的 PID 参数，电机转动的正方向，编码器反馈的正方向。
6．驱动器编号设置模式，即可以设置驱动器在 CAN 总线上的各种功能对应的 ID 号。
7．固件升级模式，即可以直接在配套的 RoboModule 伺服电机驱动器调试软件上进行固件升级。

配套的 RoboModule 伺服电机驱动器调试软件，还具备显示速度曲线，电流曲线、位置曲线的功能。用户可以无需安装 Matlab 等数学软件，就可以清晰的观察到电机运行过程中的数据波形。

典型应用场景

RoboModule 伺服电机驱动器可以适用于任何有刷直流电机的控制的场合。

1．适用于 RoboCon、RoboTac、RoboMasters、RoboCup、电子设计竞赛等比赛场合的驱动电机的应用。
2．可以应用于流水线传送带。
3．可以应用于各种自动玻璃门、地铁屏蔽门、地铁刷卡通道门等。
4．可以应用于三角形全向轮机器人底盘、正方向全向轮机器人底盘、麦克纳姆全向轮机器人底盘、履带式机器人底盘，普通差速轮机器人底盘。
5．可用于并联机械臂、XYZ 三轴机械臂的电机控制。
6．可用于钢丝绳拉线伸缩臂的电机控制。
7．可用于 XYZ 三轴雕刻机、3D 打印机、高速贴片机、平面切割机、绘图仪、下棋机器人等应用。
8．可用于 LED 旋转显示屏的电机精确控制。

产品硬件介绍

感谢您使用本产品。本产品所有硬件电路的设计都经过精确计算和反复验证，保证稳定可靠。相比于其他的直流伺服电机驱动器，它具有如下优点：

1. 使用 QFN36 封装的 STM32F103T8U6，性能和可靠程度大大优于 LQFP 封装的 STM32。体积小、重量轻、底部中央大面积外露焊盘用来良好散热，无外凸的引脚焊盘与内部导电路径短，自感系数小，封装体内的布线电阻小。所以这种封装的 STM32 相比 LQFP 封装而言，具有显著优越的电性能。

2. 超低开关电源纹波，24V 对 5V 的转换，只产生了 10mV 左右的纹波。超小体积（8mm*8mm）的开关电源电路设计，可以输出 600mA 的电流。稳定可靠的电源，是一切电路的基础。

3. 使用 9 个 X7R 村田陶瓷电容并联的高成本方案替代常规的单个铝电解电容来进行储能，储能和放电性能优越 10 倍以上。

4. 采用 CAN 总线通讯方式，专业汽车电子通讯方案，通讯误码率为 0%。

5. 使用德州仪器最新的 6.5MHz 采样频率的 OPA2374AID 轨对轨运放芯片做电流采样，更精确，更稳定。

6. 使用英飞凌的 BTN7971B，PN 结构的电机驱动芯片，PWM 占空比可达 100%，最大瞬间电流可达 70A，对付普通直流电机绰绰有余。（一般 MAXON 伺服电机线圈内阻在 2.4Ω左右，24V 下通过电机的峰值电流为 10A）

7. 采用进口的阿尔卑斯 ALPS 按键 SKRPACE010，2.55N 的稳定按键力度，5 万次的按键寿命，低抖动，绝非杂牌山寨按键可比。

8. 台湾亿光 11-22SURSYGC/S530-A2/TR8 红绿双色 LED，作为状态指示灯，质量保证，杜绝山寨杂牌。

9. 增加小体积的有源蜂鸣器作为状态提醒和异常报警，可以用听觉知晓驱动器的特殊状态，不用把注意力一直放在指示灯上。

10. 选择深圳一流的 PCB 加工厂代工生产 PCB，高精度，军工级品质。黑色沉金板四层板，有完整的参考地平面，大面积铺铜走线用来过电流电流，可以满足持续大电流的需求。

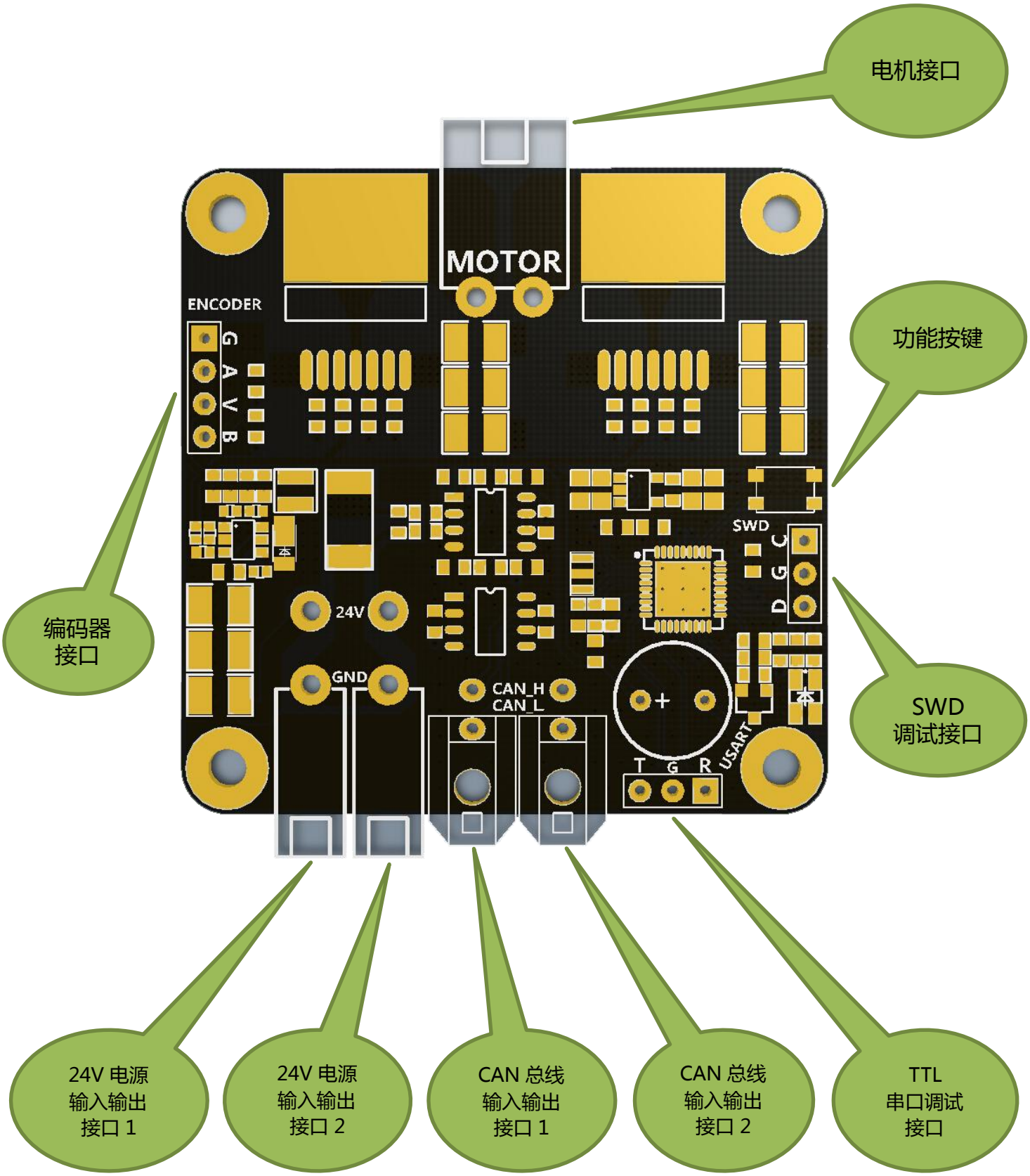
11. 选择深圳一流的 SMT 加工厂代工贴装，使用世界顶级的富士贴片机无铅 SMT 焊接，贴片精度可达到 0.03mm，焊接完美可靠，杜绝手工焊接的不良状况出现。

参数指标

电气参数	
整板供电电压范围	8V~28V
整板典型供电电压	24V
整板最大峰值电流	70A
整板最大持续电流	10A
适合驱动电机功率	120W
开关电源输出电压电流	5V@600mA
线性电源输出电压电流	3.3V@150mA
MCU 供电电压	3.3V
MCU 工作频率	72MHz
PWM 频率/周期	14.4KHz/69.4us
PWM 占空比范围	0%-100%
电流环控制周期	1ms
速度环控制周期	1ms
位置环控制周期	10ms
编码器的类型	增量式普通编码器，差分编码器需要转换
编码器的供电电压	5V
编码器的倍频数	4 倍频（相当于 500 线编码器变成 2000 线）
串口输入输出电平	TTL 电平
串口波特率	115200
CAN 总线波特率	1Mbit/s
按键力度	2.55N

机械参数	
PCB 尺寸	50mm*50mm*1.2mm
PCBA 尺寸	58mm*50mm*15mm
PCB 层数	4 层
PCB 表面处理	黑色阻焊层、白色丝印层、沉金
安装孔内径	3mm
安装孔铜环	6.3mm
相邻安装孔间距	42.888mm
整板重量	18 克

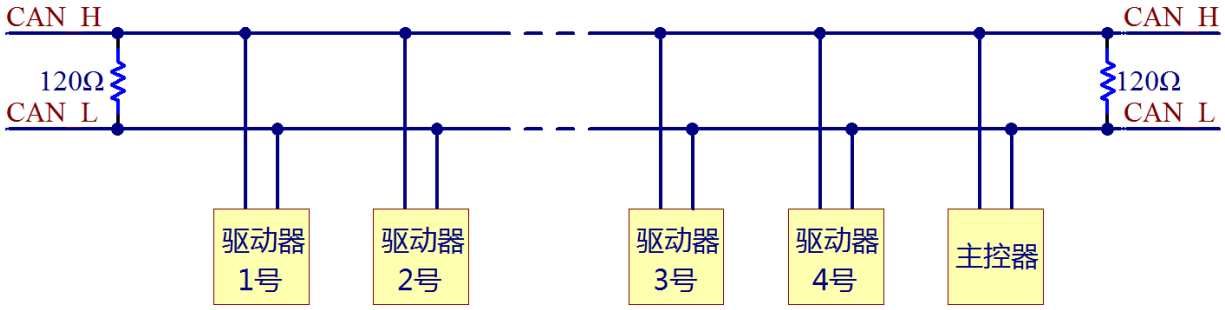
接口说明


电机接口
MOTOR
ENCODER
G A V B
功能按键
编码器
接口
SWD
D G C
24V
GND
CAN_H
CAN_L
T G R
USART
SWD
调试接口
24V 电源
输入输出
接口 1
24V 电源
输入输出
接口 2
CAN 总线
输入输出
接口 1
CAN 总线
输入输出
接口 2
TTL
串口调试
接口

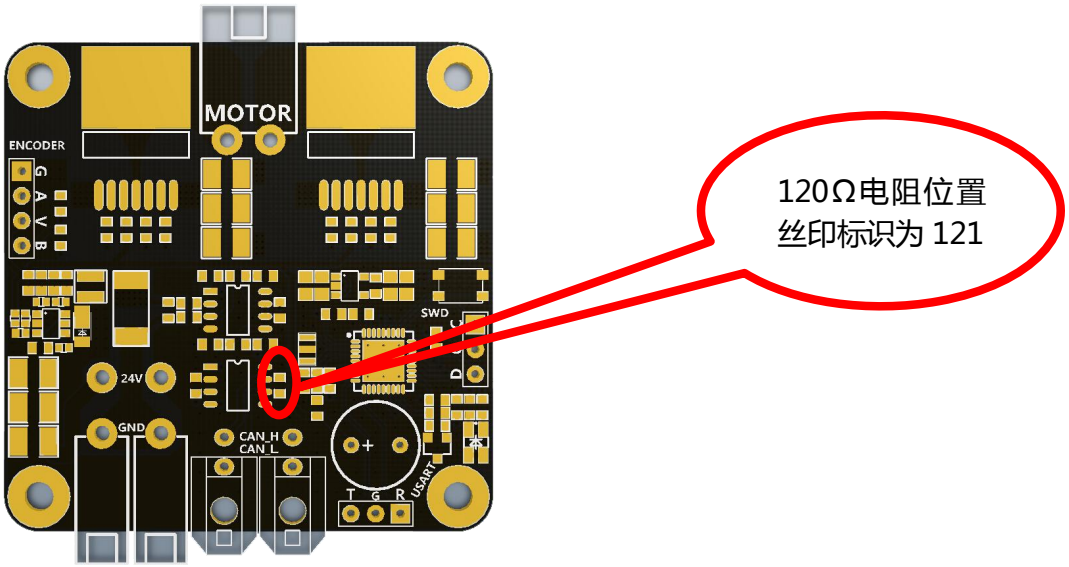
下面对上述所有接口进行详解：

1. **24V 电源输入输出接口 1，24V 电源输入输出接口 2：**
   这两个接口功能是一样的，为什么留了两个？
   考虑到一般应用环境下，用到的电机驱动器不止 1 个，比如机器人底盘，可能用到 4 个驱动器，为了整车布线的方便，可以将所有的电机驱动器的电源串联起来，2 个 24V 的电源输入输出接口，一个作为电源的输入，一个作为电源的输出，将 24V 的电源传到下一个电机驱动器。

2. **CAN 总线输入输出接口 1，CAN 总线输入输出接口 2：**
   这两个接口的功能也是一样的，为什么也留了两个？
   跟上述电源接口一样，同样考虑到串联应用的连线方便。不过要注意，CAN 总线的硬件电路拓扑，一定是如下图的:



实际连接的时候，要把 120Ω的两个终端电阻放在最端点的两个电路板上。如上图方式放置的时候，需要在驱动器 1 号和主控器上保留 120Ω电阻，驱动器 2 号、驱动器 3 号、驱动器 4 号的原有的 CAN 总线上的 120Ω电阻应该拆掉。(建议使用小口热风枪吹下，温度控制在 360℃左右即可，或者用两只电烙铁点120Ω电阻的焊盘，便可拆掉)。120Ω电阻在本驱动器上的位置在如下图示位置：



3. **TTL 串口调试接口：**
   其中丝印层上的 T 代表 USART_TXD，G 代表 GND，R 代表 USART_RXD。
   因为串口的特性，在外接其他串口设备的时候，需要按如下连接方式连接本驱动器和其他的串口设备：

本驱动器的 T~~~~其他串口设备的 R
本驱动器的 G~~~~其他串口设备的 G
本驱动器的 R~~~~其他串口设备的 T（备注：在所有的通信接口中，只有串口需要 T 和 R 交叉连接）
此串口调试接口，可以与 USB 转串口小板相连后与 PC 机通信，配合配套的“RoboModule 伺服电机驱动调试软件”来进行电机的编号设置、电机方向设置、编码器方向设置、各环路 PID 参数设置等。
此串口调试接口 ,同时也可以与主控设备相连。用户不打算用 CAN 总线来操作驱动器来控制电机的时候，就可以选择使用串口来操作驱动器从而控制电机。

4. SWD 下载接口：

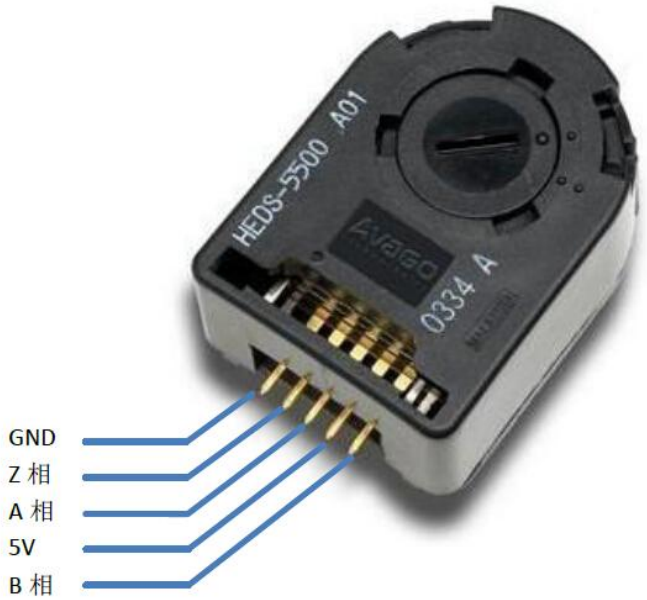
其中 D 代表 SWDIO，C 代表 SWCLK，G 代表 GND。本接口用于用户有兴趣自行编写程序定制特殊需求的情况，与本公司所售的“RoboModule Jlink 转 SWD 小板”连接来调试下载程序。“RoboModule Jlink 转 SWD 小板”单独出售，不作为标配配件赠送。
连线时候，D 对 D 连接，G 对 G 连接，C 对 C 连接。

5. 电机接口：

此接口有两个触点，与电机的两个触点相连，不分正负。可以在“RoboModule 伺服电机驱动器调试软件“上定义电机转动方向的正方向、编码器反馈的正方向，所以此处可以随意连接。

6. 编码器接口：

其中 G 代表 GND，A 代表 A 相信号，V 代表 5V 的输出，B 代表 B 相信号。
编码器（encoder），当装在电机上的时候，可以测量电机的速度和位置。
RoboModule 伺服电机驱动器可以直接支持的编码器是增量式普通输出编码器，此类编码器有 5 个引脚或者 4 个引脚，如果是 5 个引脚，则悬空不用的 Z 相的引脚。如果是 10 根引出线的差分编码器，请联系我们购买转换器。可以直接接入本驱动器的增量编码器如下图所示：



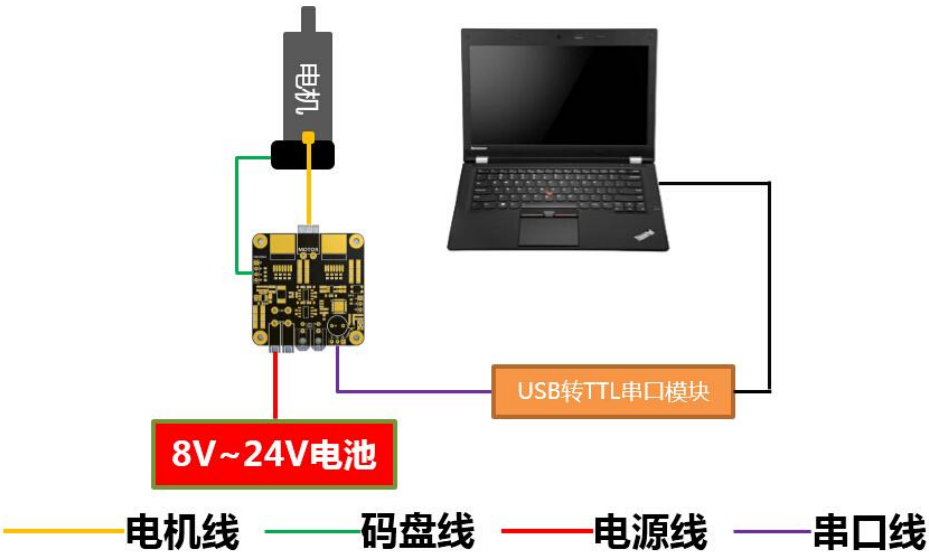
用户所使用的编码器因为品牌不同可能会有一些差别，但基本上引脚都是如图所示的排列方式。从左往右，第一引脚为 GND，第二引脚为 Z 相，第三引脚为 A 相，第四引脚为 5V 的电源输入，第五引脚为 B 相。一般应用下，只用到 A、B 相，Z 相悬空不用。

连线时候，按照 GND 连 GND，A 相连 A 相，5V 连 5V，B 相连 B 相的方式连接。

调试软件使用说明

本段介绍如何使用这个驱动器。
每个驱动器在出货前都会下载好官方程序，所以用户不需要编写一行代码便可操作驱动器。

请先使用电脑和 USB 转 TTL 串口模块连接本驱动器，按照下面的连接方式，并使用“RoboModule 伺服电机驱动器调试软件”来进行驱动器和新电机的匹配调试。
下面介绍如何使用“RoboModule 伺服电机驱动器调试软件”进行驱动器和电机的匹配。



按照如上图的方式，进行连接。
注意两点问题：
1．使用 USB 转 TTL 串口模块，不能使用 RS232 串口模块，因为此处串口电平是 TTL 的。
2．请确认连接好串口线、码盘线、电机线，再进行 24V 通电，请不要在 24V 通电的过程中插拔串口线、码盘线、电机线，否则有可能损坏驱动器。
确认好以上两点后，打开“RoboModule 伺服电机驱动器调试软件”。

如果电脑上已经插有 USB 转串口模块或者自带串口，则在串口号上会显示当前有效的串口号。如果没有，图示中串口号为空，无法继续进入调试模式。

点击“进入调试模式”按钮，则页面切换到如下界面，同时伴随着驱动器上蜂鸣器的一声长鸣，如果蜂鸣器没有一声长鸣，则关掉此窗口，重新打开软件，重新进入。

在此页面上，用户可以选择图示“模式选择”的 7 种模式，选中其中一种模式，则可以进入下一个对应的界面。

首先要选择 PWM 模式 1（速度反馈）进入。

如下图是 PWM 模式的界面，进入此界面时，驱动器上的蜂鸣器会有一声短鸣。

PWM 模式下，数据接收窗口反馈的是编码器在 1ms 时间内的转过的线数，即所谓的速度，两个相邻数据的间隔为 10ms。

PWM 模式是开环的模式。有两种作用：

其一，用于辅助调整电机转动的正方向和编码器反馈的正方向。

其二，用于开环控制不带编码器的有刷直流电机。

如何调整电机转动的正方向：

假如给定正数的 PWM 数值后，电机往顺时针方向转动，而用户想要让电机在给定正数的 PWM 数值的时候往逆时针方向转，则需要重启软件，切换到“驱动器参数设置模式”，点击电机转动方向取反的按钮，然后再重启软件，再次进入“PWM 模式 1（速度反馈）”，则可以看到，给定正数的 PWM 数值后，电机转动方向为逆时针，同理，给定负数的 PWM 数值后，电机转动方向为顺时针。

如何调整编码器反馈的正方向：

在调整好电机转动的正方向后，给定一个正数的 PWM 数值，观察“数据接收”窗口中反馈的编码器数据是否为正数，如果编码器的方向不是正数，则需要重启软件，再切换到“驱动器参数设置模式”，点击编码器方向取反的按钮，然后再重启软件，再次进入“PWM 模式 1（速度反馈）”，则可以看到，给定正数的 PWM 数值后，编码器反馈的数值也为正数。给定负数的 PWM 数值后，编码器反馈的方向也为负值。

此处，给定+1000 的数值表示 1000/5000=1/5 的占空比的 PWM 输出。

此处，编码器反馈的数值的意义为：1ms 的时间里面，编码器被带动转过的线数。

注意，必须使给定 PWM 的数值和编码器反馈的数值同为正数或者同为负数，即符号相同，否则进入速度模式和位置模式后有可能烧坏电机或者驱动器。

如下图是 PWM 模式 2（电流反馈）的界面，进入此界面时，驱动器上的蜂鸣器会有两声短鸣。

（如果无需用到堵转保护功能，可以直接跳过本功能的测试，本功能与速度位置两环相对独立，无关联）

PWM 模式 2（电流反馈）下，功能与 PWM 模式 1（速度反馈）基本完全一致，唯一的区别在于，“数据窗口”中反馈的数据意义不同。

PWM 模式 1(速度反馈)中，“数据窗口”反馈的数据意义是 1ms 的时间内，编码器被带动转过的线数。

PWM 模式 2（电流反馈）中，“数据窗口”反馈的数据意义是当前 AD 采样的值。

数据反馈的时间间隔都是固定的 10 毫秒。

如下图是速度环模式的界面，进入此界面时，驱动器上的蜂鸣器会有三声短鸣。

速度环模式下，数据接收窗口反馈的是编码器在 1ms 时间内的转过的线数，即所谓的速度，两个相邻数据的间隔为 10ms。

速度环模式需要建立在电机转动的正方向和编码器反馈的正方向一致的条件下，即：在 PWM 模式 1（速度反馈）下，给正数的 PWM 数值后，必须得到正数的编码器反馈数值。给负数的 PWM 数值，必须得到负数的编码器反馈数值。

发送指令包含两部分，

第一部分是限制 PWM 的最大值。如果担心在运行过程中发生堵转烧毁电机，则可以限制 PWM 占空比在很小的数值。如果需要快速响应给定速度值，则在限制 PWM 占空比的文本框中填入比较大的数值，比如最大值 5000。此处限制值为绝对值，只支持正数的输入。

第二部分是给定速度值，这里的速度值的含义是：在 1ms 的时间内，编码器被带动转过的线数。每个电机的最大速度值都有上限。如果给定速度值超出上限，则按能达到的最大速度值运转。

注意：

1．假如进入后，给定速度值没有反应或者反应异常，则请点击“驱动器参数设置模式”下的“读取电机参数”，如果都为零，请点击“读取出厂参数”，再点击“写入电机参数”。

2．当用户对当前的速度调节状态不满意的时候，可以重启软件，切换到“驱动器参数设置模式”下，修改当前速度环下的 PID 参数值，根据反馈的曲线来寻找最优解。

如下图是位置环模式的界面，进入此界面时，驱动器上的蜂鸣器会有四声短鸣。

位置环模式下，数据接收窗口反馈的是电机当前的位置值。

位置环模式是建立在速度环模式之上，也就是速度环模式调试 ok 之后，才能去调试位置环模式，否则会出现振荡，甚至烧毁电机和驱动器。

发送的指令包含 3 部分：

第一部分是限制的 PWM 值，文本框中所填的值越大，响应速度则越快，同时输出电流也越大。当可能存在堵转的情况下，用户可以适当的下调限制 PWM 值的数值。从而减少堵转情况下烧毁电机的可能，但同时也会降低其响应速度。

第二部分是限制的速度值，与速度环中所调试的内容意义一致。速度越大，到达目标的位置时间越短。速度越小，到达目标位置的时间越长。

第三部分是给定位置值，范围是-5000000 线到+5000000 线。

如何计算位置：

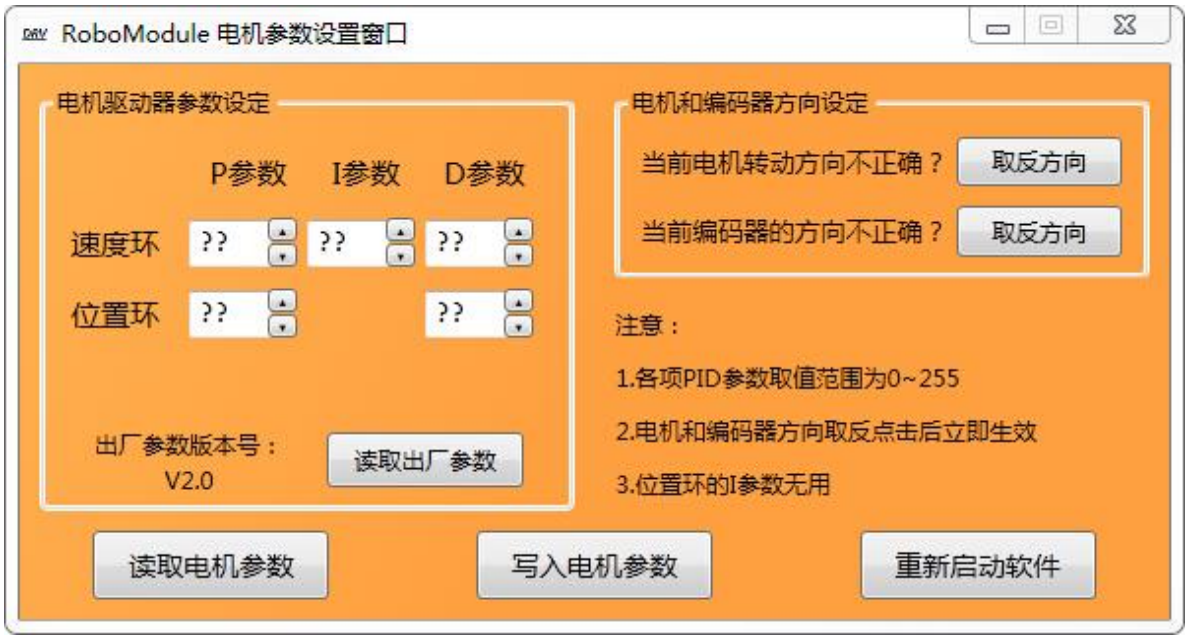
假如一个电机接了 500 线的编码器，带了 1：16 的减速箱。

如果要让电机输出轴转动一圈，则需要填入多少的数值呢？

500 线的编码器，实际转动 1 圈是 2000 线的数值（此处涉及编码器四倍频的问题），并且 1：16 的减速，

则需要 2000*16 = 32000 线，这就是让输出轴转动一圈要给定的位置值。

如下图是驱动器参数设置模式窗口，进入此界面时，驱动器上的蜂鸣器会有五声短鸣。



本窗口在前面内容已经有所涉及。
窗口左侧，用以调试速度和位置环路的 PID 参数值。
先读取电机的参数，然后修改参数，修改后写入电机参数。
如果调试到错乱的地步，可以读取出厂参数，然后写入电机参数，之后重新开始调试。

调试位置环之前，需要先调试稳定速度环。因为位置环是建立在速度环之上的。

窗口右侧，对应于 PWM 模式下，调整设置电机转动的正方向和编码器的正方向。
点击取反方向后立即生效，无需再点击 “写入电机参数”。

如下是驱动器编号设置模式的窗口，进入此界面时，驱动器上的蜂鸣器会有六声短鸣。

比如要设置当前的驱动器为 2 号驱动器，则需要先选中 2 号。选中之后，后面打了问号的文本框会变成**红色带*号字体**，并展示了所有对应的 CAN 的 ID 号，如下图所示。

然后点击**“写入电机编号“**按钮，如果写入成功，则后面文本框的**红色带*号字体**会变成**黑色普通字体。**如果写入失败，则**红色带*号字体**不会发生变化。同样，可以通过点击“**读取电机编号**”按钮来读取验证的当前写入的编号值正确与否，如果正确，则会在文本框中显示**黑色普通字体**，如下图所示：

如果脱离电脑使用，是用串口的方式来控制驱动器的运转，则不需要对驱动器进行编号。驱动器编号适用于 CAN 总线来控制驱动器的场合。

如本界面的窗口左侧显示，驱动器编号范围是 1 号到 15 号，则表明当前设计的 CAN 总线通信方案，最多可以支持 15 个驱动器在同一条 CAN 总线上一起使用。另外必须保证，同一条 CAN 总线上所挂的驱动器，编号不能一样，比如不能有两个 2 号驱动器共存。

本窗口的右侧部分的 CAN_ID 的编码，与下文 CAN 通信协议部分所述的 CAN_ID 编号方式一一对应。

在 PWM 模式 1、PWM 模式 2、速度环模式、位置环模式，这四种模式下，可以使用“RoboModule 伺服电机驱动器调试软件”自带的绘图功能，将数据接受窗口中的接收数值绘制成曲线。

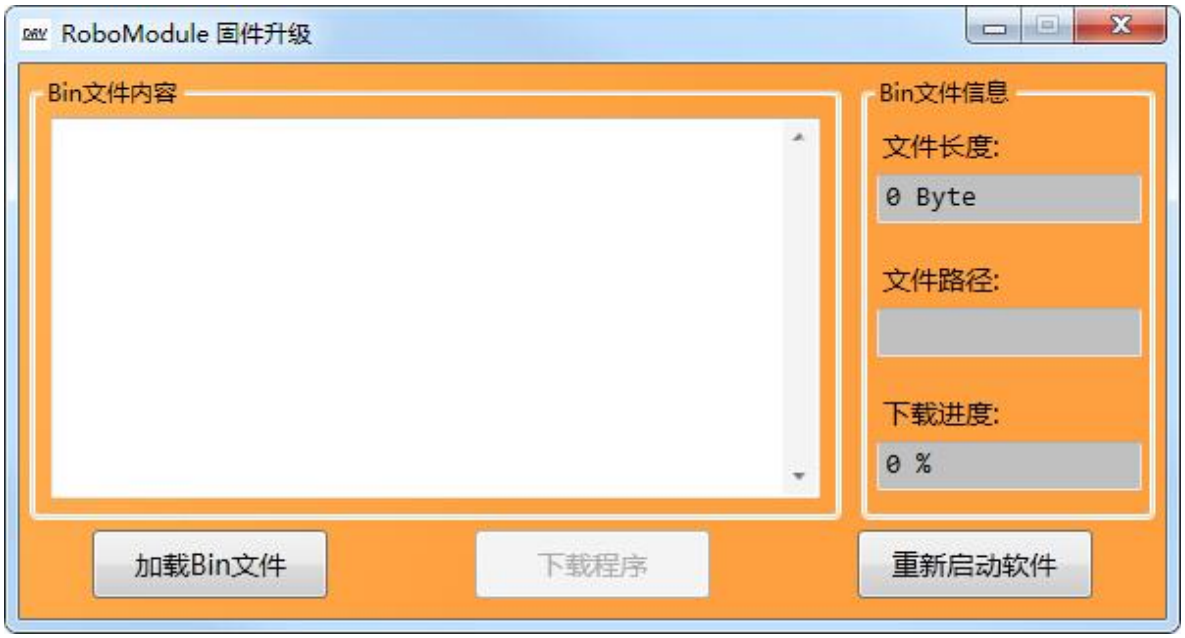
调试软件的固件升级功能说明

调试软件上自带的固件升级功能可以让驱动器在不使用 Jlink 的情况下也能成功的下载新程序。

具体使用方法如下：

给驱动器连接上串口线到电脑，然后给驱动器的总电源上电。打开软件，进入到“驱动器固件升级模式”，此时如果连接成功，蜂鸣器会有 7 声短鸣。

如下图是固件升级的界面：



当需要固件升级的时候，请进入此界面。

点击“加载 Bin 文件”，然后从弹出的打开框中找到相应的 bin 文件并打开后，界面会变化到如下形式：

然后点击“下载程序”，则驱动器将开始下载程序，同时，蜂鸣器会有十多声鸣叫，图示界面的“下载进度”会显示当前下载程序的进度。

更新完程序后，请点击“重新启动软件”，或者直接关闭软件，则更新固件成功。

另外请注意，所述 bin 文件必须确认是从 RoboModule 官方处获得，否则，驱动器有可能在更新新固件后无法使用。

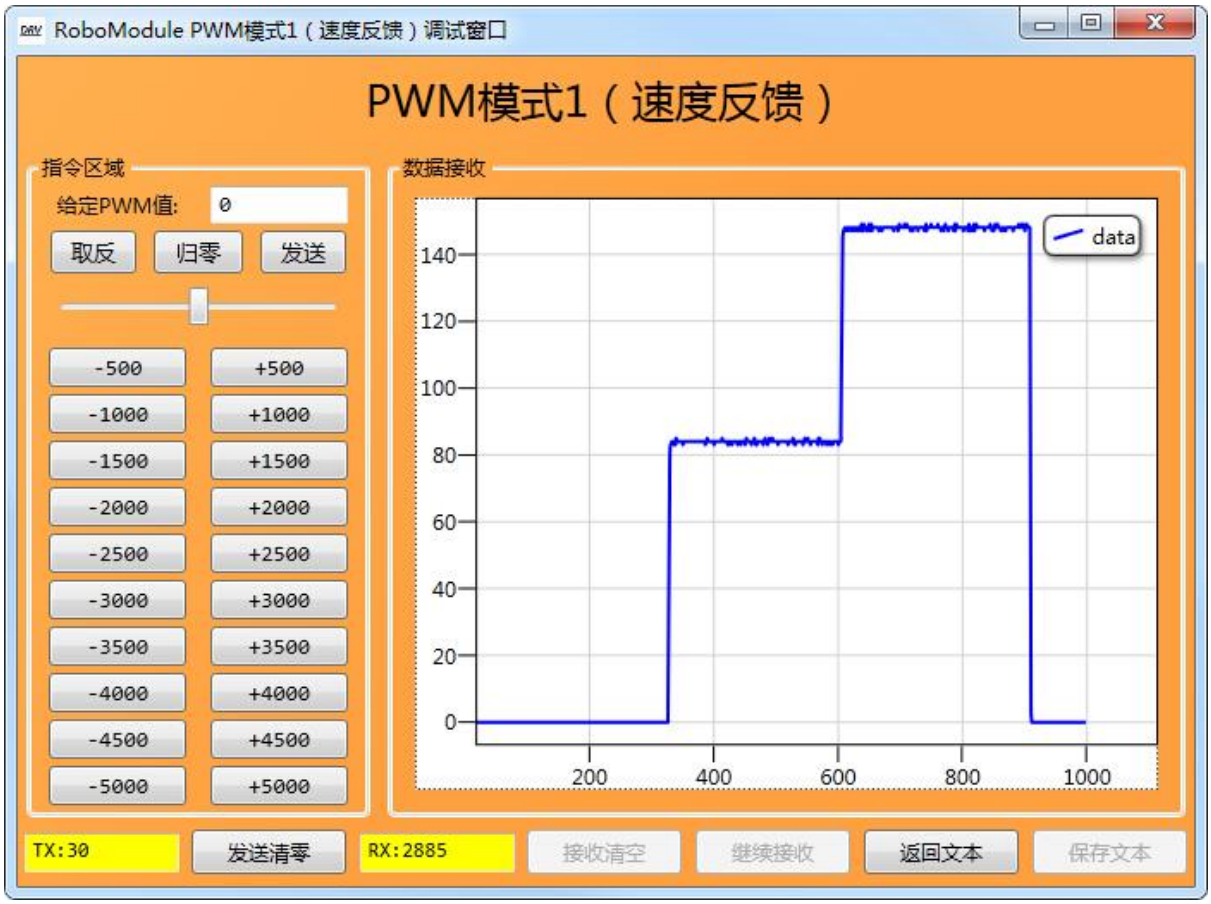
上位机绘图功能使用说明

在 PWM 模式 1（速度反馈）、PWM 模式 2（电流反馈）、速度环模式、位置环模式，这四种模式下，可以使用“RoboModule 伺服电机驱动器调试软件”自带的绘图功能，将数据接受窗口中的接收数值绘制成曲线。

以下举例说明：
如下图是文本接收：

当点击“绘制曲线”按钮后，稍等一会，曲线绘制完成：

操作方法是：在窗口中有数据的时候，先点击“暂停接收”，然后点击“绘制曲线”，然后等待一段时间，就会出现如图所示画面。



本绘图功能十分强大，在 X 轴上滚动滚轮，可以放大缩小 X 轴的精度。在 Y 轴在滚动滚轮，可以放大缩小 Y 轴的精度。同样在窗口中间滚动滚轮，可以同时放大缩小 X 和 Y 轴的精度。最小的精度，可以到 1。

在窗口中间按住鼠标左键，挪动鼠标，可以拖动曲线。

窗口可以任意放大，缩小，可以最大化。

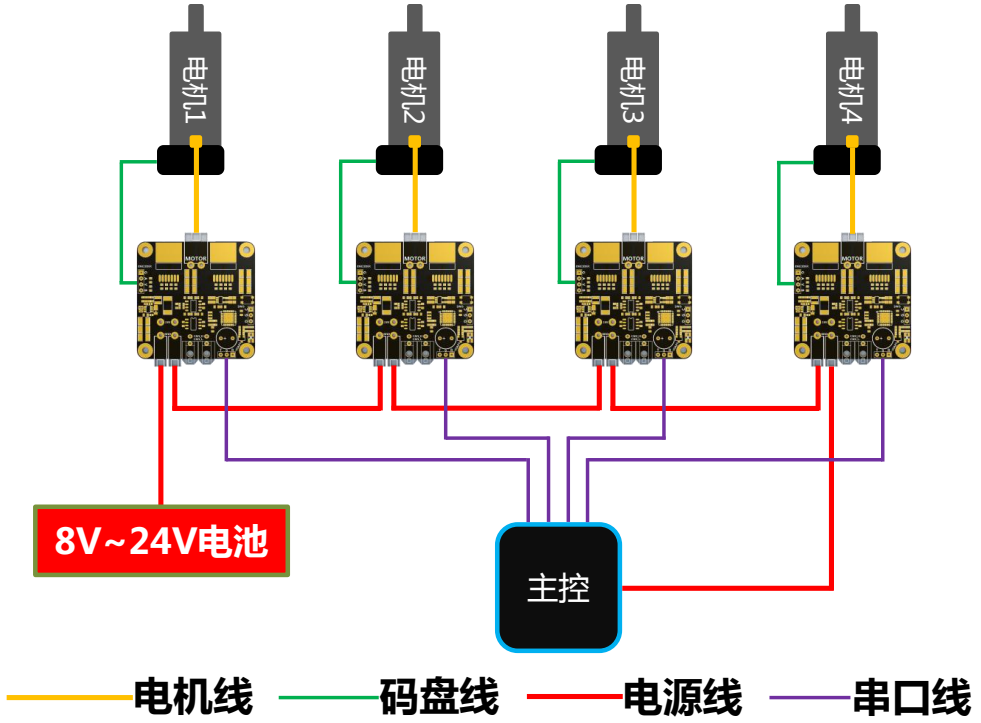
在曲线窗口点击鼠标右键，可以一键调整到适合屏幕显示、可以一键保存截图。

如果需要保存原始的数据，可以点击“返回文本”按钮，则界面会切换到文本数据，然后再点击“保存文本”，按照提示，即可将“数据接收”窗口的原始数据保存为 TXT 文本。

注意：文本解析成曲线需要一定的时间，如果“数据接收”窗口中数据量很大，则需要等待更长一点的时间。

使用串口控制驱动器操作说明

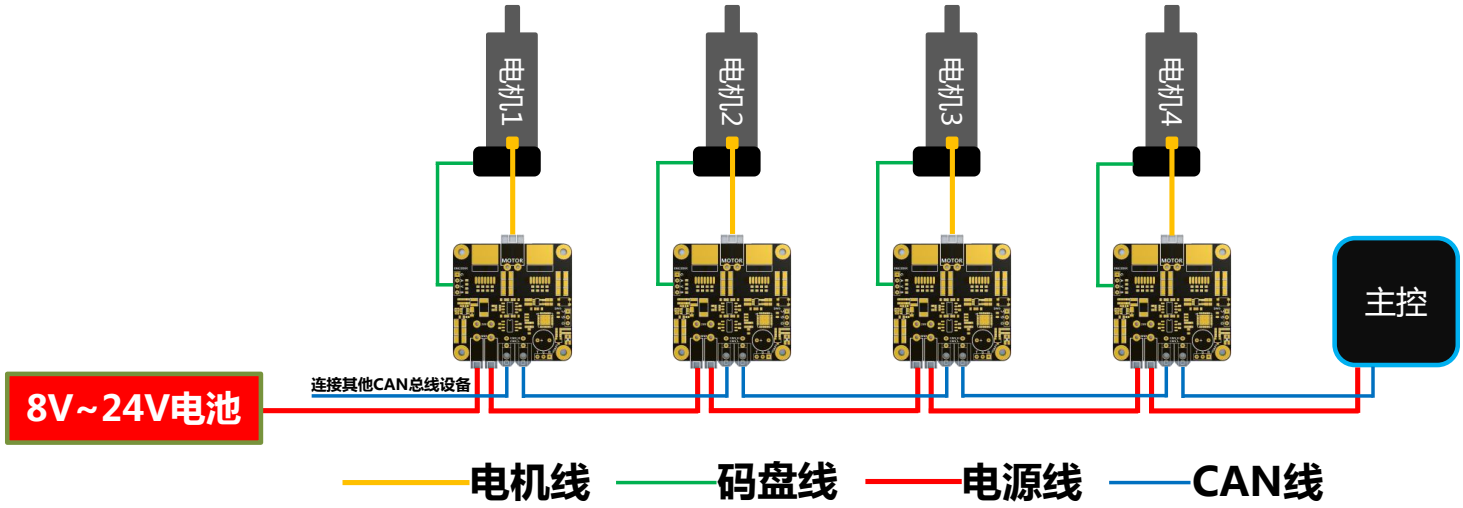
用户调参完毕，使用串口的方式来控制驱动器的连线图如下所示。



如图所示，使用四个电机驱动器，并且用四个独立串口来单独控制。则对应的主控，需要具备 4 个独立串口，所涉及的通信协议，遵循下文的串口通信协议。

使用 CAN 总线控制驱动器操作说明

用户调参完毕，使用 CAN 总线的方式来控制驱动器的连线图如下所示。



如图所示，使用四个电机驱动器，并且用 CAN 口来控制，则只需要占用一个 CAN 的资源就可以将所有的驱动器连接起来。

所涉及的通信协议，遵循下文的 CAN 通信协议。

注意：

一条 CAN 总线上的 120Ω 电阻只能留下 2 个，即留下 CAN 总线两个端点上的电阻，其余的 120Ω必须全部拆掉。请参考“接口说明”段的第 2 点。

USART 串口通信协议

本段介绍如何使用串口的方式来控制驱动器。

使用串口的方式来控制有两个用途：

用途一：最主要的用途是配合配套的“RoboModule 伺服电机驱动器调试软件”来进行电机的编号设置、电机正方向设置、编码器正方向设置、各环路 PID 参数设置等。

用途二：当用户不打算用 CAN 总线来操作驱动器来控制电机的时候，就可以选择使用串口来操作驱动器来控制电机。但相对于 CAN 通信方式，串口控制的方式有如下几个缺点：

1． MCU 与 MCU 之间使用 TTL 电平的串口通信，可通信距离较短，抗干扰能力较差。

2． USART 串口只支持点对点通信，就是说，假如主控器自身只有 1 个通信串口，那么这个主控器只能接 1 个电机驱动器。如果主控器串口数量比较多，有 5 个，则可以接 5 个电机驱动器。

实际上两个用途都共用一套 USART 通信协议，本段只涉及用户使用串口来控制驱动器来操作电机部分，不涉及驱动器参数设置和驱动器编号部分。

下面正式介绍 USART 串口通信协议：

因为串口指令只能是一个主机（电脑作为主机或者 MCU 主控器作为主机）和一个从机（当前连接的驱动器作为从机），所以，所发送的所有指令，都只有本串口所连接的驱动器能收到，所以不存在选中哪一个驱动器的问题，所以不需要像 CAN 总线那样用标识符来区分谁是谁。

每段串口命令都是由 10 个字节组成，有如下一些指令：

1． 让当前驱动器复位。

2． 让当前驱动器进入某种模式，有四种模式：PWM 模式 1、PWM 模式 2、速度环模式、位置环模式。

3． 在 PWM 模式 1 下给驱动器发送目标 PWM 的指令，注意在此模式下，驱动器将通过串口以 10ms 的周期对外发送速度值。所谓速度值，是指编码器在 1ms 的时间内转动的线数。

4． 在 PWM 模式 2 下给驱动器发送目标 PWM 的指令，注意在此模式下，驱动器将通过串口以 10ms 的周期对外发送电流值，所谓电流值，是指驱动器母线流过的电流大小，单位是 mA。

5． 在速度环模式下给驱动器发送限制 PWM 和速度的指令，注意在此模式下，驱动器将通过串口以 10ms 的周期对外发送速度值。所谓速度值，是指编码器在 1ms 的时间内转动的线数。

6． 在位置环模式下给驱动器发送限制 PWM、限制速度和目标位置的指令，注意在此模式下，驱动器将通过串口以 10ms 的周期对外发送位置值。所谓位置值，是指编码器相对于进入模式时候的起点的编码器偏移量。

下面来分解每个指令的具体内容：

让驱动器复位

Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]
0x55	0x55	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff

发送本指令后，驱动器会立即复位。

让驱动器进入某种模式

Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]
0x55	模式值	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff

如果 Data[1]=0x56，让驱动器进入 PWM 模式 1
如果 Data[1]=0x57，让驱动器进入 PWM 模式 2
如果 Data[1]=0x58，让驱动器进入速度环模式
如果 Data[1]=0x59，让驱动器进入位置环模式

如果驱动器已经进入 PWM 模式 1

Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]
0x55	0x56	PWM	PWM	0x00	0x00	0x00	0x00	0x00	0x00

让当前驱动器连接的电机在 PWM 模式 1 下，让它以 temp_pwm 的占空比转动：
则 Data[2] = (unsigned char)((temp_pwm>>8)&0xff);
Data[3] = (unsigned char)(temp_pwm&0xff);
其中 temp_pwm 的取值范围为-5000~+5000。

注意：驱动器进入 PWM 模式 1 之后，驱动器就会向串口以 10ms 的间隔，周期性的对外发送其当前的实际速度值。所谓速度值，是指编码器在 1ms 的时间内转过的线数。

如果驱动器已经进入 PWM 模式 2

Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]
0x55	0x57	PWM	PWM	0x00	0x00	0x00	0x00	0x00	0x00

让当前驱动器连接的电机在 PWM 模式 2 下，让它以 temp_pwm 的占空比转动：
则 Data[2] = (unsigned char)((temp_pwm>>8)&0xff);
Data[3] = (unsigned char)(temp_pwm&0xff);
其中 temp_pwm 的取值范围为-5000~+5000。

注意：驱动器进入 PWM 模式 2 之后，驱动器就会向串口以 10ms 的间隔，周期性的对外发送其当前的实际电流值。所谓电流值，是指当前流过电机母线的电流，单位是 mA。

如果驱动器已经进入速度环模式

Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]
0x55	0x58	PWM	PWM	速度值	速度值	0x00	0x00	0x00	0x00

让当前驱动器连接的电机以 temp_pwm 的限制占空比，以 temp_speed 的速度转动：
则 Data[2] = (unsigned char)((temp_pwm>>8)&0xff);
Data[3] = (unsigned char)(temp_pwm&0xff);
Data[4] = (unsigned char)((temp_speed>>8)&0xff);
Data[5] = (unsigned char)(temp_speed&0xff);
其中 temp_pwm 的取值范围为 0~+5000。
temp_speed 的取值范围为：-1000~+1000。

注意：驱动器进入速度环模式之后，驱动器就会向串口以 10ms 的间隔，周期性的对外发送其当前的实际速度值。所谓速度值，是指编码器在 1ms 的时间内转过的线数。

如果驱动器已经进入位置环模式

Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]
0x55	0x59	PWM	PWM	速度值	速度值	位置值	位置值	位置值	位置值

让当前驱动器连接的电机以 temp_pwm 的限制占空比，temp_speed 的限制速度，转动到 temp_location 的位置：

则 Data[2] = (unsigned char)((temp_pwm>>8)&0xff);
Data[3] = (unsigned char)((temp_pwm)&0xff);
Data[4] = (unsigned char)((temp_speed>>8)&0xff);
Data[5] = (unsigned char)(temp_speed&0xff);
Data[6] = (unsigned char)((temp_location>>24)&0xff);
Data[7] = (unsigned char)((temp_location>>16)&0xff);
Data[8] = (unsigned char)((temp_location>>8)&0xff);
Data[9] = (unsigned char)(temp_location&0xff);

其中 temp_pwm 的取值范围为 0~+5000。
temp_speed 的取值范围为：0~+1000。
temp_location 的取值范围为：-5000000~+5000000。

注意：驱动器进入位置环模式之后，驱动器就会向串口以 10ms 的间隔，周期性的对外发送其当前的实际位置值。所谓位置值，是指相对于刚进入位置环模式时候，当前编码器转过的线数。

CAN 通信协议

本段介绍如何使用 CAN 总线的方式来操作本驱动器来控制电机的各种方式转动。

在使用本段 CAN 通信协议之前，必须先使用“RoboModule 伺服电机驱动器调试软件”来对驱动器进行编号、设置电机转动的正反方向，设置编码器的正反方向，调整各项 PID 参数。在完成上述操作之后，下面的 CAN 通信协议才有意义，否则会出现异常，甚至烧毁驱动器。

对于本段协议，CAN 模块的设置，需要满足以下四个条件：
1．1M 波特率
2．数据帧
3．标准帧
4．帧数据长度为 8 个字节

在 CAN 通信协议下，对驱动器的操作命令有如下 3 类：
1. 让驱动器复位。
2. 让驱动器进入某种控制方式：第一种是 PWM 模式 1，第二种是 PWM 模式 2，第三种是速度环控制方式，第四种是位置环控制方式。
3. 在驱动器进入第 2 点所述的四种模式后，发送相应的目标参数值，与第 2 点一一对应。

以下是本驱动器 CAN 通信的所有协议：

0．复位指令：

本指令在任何状态下都会直接生效。发送本指令后，驱动器会立即复位，即程序从头开始运行。

CAN_ID	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
待选择	0x55	0x55	0x55	0x55	0x55	0x55	0x55	0x55

举例说明：
让所有的驱动器进行复位的 CAN_ID：0x00（广播）
让 01 号驱动器进行复位的 CAN_ID：0x10
让 02 号驱动器进行复位的 CAN_ID：0x20
让 03 号驱动器进行复位的 CAN_ID：0x30
让 04 号驱动器进行复位的 CAN_ID：0x40
让 05 号驱动器进行复位的 CAN_ID：0x50
让 06 号驱动器进行复位的 CAN_ID：0x60
让 07 号驱动器进行复位的 CAN_ID：0x70
让 08 号驱动器进行复位的 CAN_ID：0x80
让 09 号驱动器进行复位的 CAN_ID：0x90
让 10 号驱动器进行复位的 CAN_ID：0xA0
让 11 号驱动器进行复位的 CAN_ID：0xB0

让 12 号驱动器进行复位的 CAN_ID：0xC0
让 13 号驱动器进行复位的 CAN_ID：0xD0
让 14 号驱动器进行复位的 CAN_ID：0xE0
让 15 号驱动器进行复位的 CAN_ID：0xF0

当复位完成后，所选的驱动器会自动对外发送如下 CAN 数据包：

CAN_ID	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x7FF	待选择	0x55	0x55	0x55	0x55	0x55	0x55	0x55

所发送的数据包的 CAN_ID 为 0x7FF，Data[0]为当前驱动器的编号。
比如，2 号驱动器复位完成，则发送：
CAN_ID = 0x7FF;
Data[0] = 0x02; //当前驱动器的编号
Data[1] = 0x55; //固定填充值
Data[2] = 0x55; //固定填充值
Data[3] = 0x55; //固定填充值
Data[4] = 0x55; //固定填充值
Data[5] = 0x55; //固定填充值
Data[6] = 0x55; //固定填充值
Data[7] = 0x55; //固定填充值
用户可以利用此项功能检测所有驱动器是否在线。

一．模式选择指令：

本指令只在驱动器未进入任何模式的情况下生效。如果驱动器已经进入某种模式，再发送此指令则会报错。所以在发送本指令前，建议先发送复位指令，等待驱动器复位完成（大约 500ms），再发送本指令。

CAN_ID	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
待选择	待选择	0x55	0x55	0x55	0x55	0x55	0x55	0x55

举例说明：
让所有驱动器进入某种模式的 CAN_ID：0x01（广播）
让 01 号驱动器进入某种模式的 CAN_ID：0x11
让 02 号驱动器进入某种模式的 CAN_ID：0x21
让 03 号驱动器进入某种模式的 CAN_ID：0x31
让 04 号驱动器进入某种模式的 CAN_ID：0x41
让 05 号驱动器进入某种模式的 CAN_ID：0x51
让 06 号驱动器进入某种模式的 CAN_ID：0x61
让 07 号驱动器进入某种模式的 CAN_ID：0x71
让 08 号驱动器进入某种模式的 CAN_ID：0x81
让 09 号驱动器进入某种模式的 CAN_ID：0x91
让 10 号驱动器进入某种模式的 CAN_ID：0xA1
让 11 号驱动器进入某种模式的 CAN_ID：0xB1
让 12 号驱动器进入某种模式的 CAN_ID：0xC1
让 13 号驱动器进入某种模式的 CAN_ID：0xD1

让 14 号驱动器进入某种模式的 CAN_ID：0xE1
让 15 号驱动器进入某种模式的 CAN_ID：0xF1

让 xx 号驱动器进入 PWM 模式 1 的 Data[0] = 0x11
让 xx 号驱动器进入 PWM 模式 2 的 Data[0] = 0x22
让 xx 号驱动器进入速度环模式的 Data[0] = 0x33
让 xx 号驱动器进入位置环模式的 Data[0] = 0x44

二．PWM 模式 1 下的数据指令：

本指令只有在驱动器进入 PWM 模式 1 之后才有效，其他任何状态下发送本指令都会让驱动器报错。
支持连续发送本指令来修改 PWM 的值，但连续发送的时间间隔不能小于 2 毫秒，建议以 10 毫秒为周期。

CAN_ID	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
待选择	待计算	待计算	0x55	0x55	0x55	0x55	0x55	0x55

举例说明：
给所有的驱动器在 PWM 模式 1 下发参数指令的 CAN_ID：0x02（广播）
给 01 号驱动器在 PWM 模式 1 下发参数指令的 CAN_ID：0x12
给 02 号驱动器在 PWM 模式 1 下发参数指令的 CAN_ID：0x22
给 03 号驱动器在 PWM 模式 1 下发参数指令的 CAN_ID：0x32
给 04 号驱动器在 PWM 模式 1 下发参数指令的 CAN_ID：0x42
给 05 号驱动器在 PWM 模式 1 下发参数指令的 CAN_ID：0x52
给 06 号驱动器在 PWM 模式 1 下发参数指令的 CAN_ID：0x62
给 07 号驱动器在 PWM 模式 1 下发参数指令的 CAN_ID：0x72
给 08 号驱动器在 PWM 模式 1 下发参数指令的 CAN_ID：0x82
给 09 号驱动器在 PWM 模式 1 下发参数指令的 CAN_ID：0x92
给 10 号驱动器在 PWM 模式 1 下给参数指令的 CAN_ID：0xA2
给 11 号驱动器在 PWM 模式 1 下给参数指令的 CAN_ID：0xB2
给 12 号驱动器在 PWM 模式 1 下给参数指令的 CAN_ID：0xC2
给 13 号驱动器在 PWM 模式 1 下给参数指令的 CAN_ID：0xD2
给 14 号驱动器在 PWM 模式 1 下给参数指令的 CAN_ID：0xE2
给 15 号驱动器在 PWM 模式 1 下给参数指令的 CAN_ID：0xF2

让 xx 号驱动器连接的电机在 PWM 模式 1 下，让它以 temp_pwm 的占空比转动：
则 Data[0] = (unsigned char)((temp_pwm>>8)&0xff);
Data[1] = (unsigned char)(temp_pwm&0xff);
其中 temp_pwm 的取值范围为-5000~+5000。

三．PWM 模式 2 下的数据指令：

本指令只在驱动器进入 PWM 模式 2 之后才有效，其他任何状态下发送本指令都会让驱动器报错。
支持连续发送本指令来修改 PWM 的值，但连续发送的时间间隔不能小于 2 毫秒，建议以 10 毫秒为周期。

PWM 模式 2 在 CAN 协议下的工作方式与 PWM 模式 1 没有任何区别。如果使用开环模式来控制电机，两者任选其一即可。

CAN_ID	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
待选择	待计算	待计算	0x55	0x55	0x55	0x55	0x55	0x55

举例说明：
给所有的驱动器在 PWM 模式 2 下发参数指令的 CAN_ID：0x03（广播）
给 01 号驱动器在 PWM 模式 2 下发参数指令的 CAN_ID：0x13
给 02 号驱动器在 PWM 模式 2 下发参数指令的 CAN_ID：0x23
给 03 号驱动器在 PWM 模式 2 下发参数指令的 CAN_ID：0x33
给 04 号驱动器在 PWM 模式 2 下发参数指令的 CAN_ID：0x43
给 05 号驱动器在 PWM 模式 2 下发参数指令的 CAN_ID：0x53
给 06 号驱动器在 PWM 模式 2 下发参数指令的 CAN_ID：0x63
给 07 号驱动器在 PWM 模式 2 下发参数指令的 CAN_ID：0x73
给 08 号驱动器在 PWM 模式 2 下发参数指令的 CAN_ID：0x83
给 09 号驱动器在 PWM 模式 2 下发参数指令的 CAN_ID：0x93
给 10 号驱动器在 PWM 模式 2 下给参数指令的 CAN_ID：0xA3
给 11 号驱动器在 PWM 模式 2 下给参数指令的 CAN_ID：0xB3
给 12 号驱动器在 PWM 模式 2 下给参数指令的 CAN_ID：0xC3
给 13 号驱动器在 PWM 模式 2 下给参数指令的 CAN_ID：0xD3
给 14 号驱动器在 PWM 模式 2 下给参数指令的 CAN_ID：0xE3
给 15 号驱动器在 PWM 模式 2 下给参数指令的 CAN_ID：0xF3

让 xx 号驱动器连接的电机在 PWM 模式 2 下，让它以 temp_pwm 的占空比转动：
则 Data[0] = (unsigned char)((temp_pwm>>8)&0xff);
Data[1] = (unsigned char)(temp_pwm&0xff);
其中 temp_pwm 的取值范围为-5000~+5000。

四．速度环模式下的数据指令：

本指令只在驱动器进入速度模式之后才有效，其他任何状态下发送本指令都会让驱动器报错。
支持连续发送本指令来修改 PWM 的限制值和给定的速度值，但连续发送的时间间隔不能小于 2 毫秒，建议以 10 毫秒为周期。

CAN_ID	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
待选择	待计算	待计算	待计算	待计算	0x55	0x55	0x55	0x55

举例说明：
给所有的驱动器在速度环模式下发参数指令的 CAN_ID：0x04（广播）
给 01 号驱动器在速度环模式下发参数指令的 CAN_ID：0x14
给 02 号驱动器在速度环模式下发参数指令的 CAN_ID：0x24
给 03 号驱动器在速度环模式下发参数指令的 CAN_ID：0x34
给 04 号驱动器在速度环模式下发参数指令的 CAN_ID：0x44
给 05 号驱动器在速度环模式下发参数指令的 CAN_ID：0x54

给 06 号驱动器在速度环模式下发参数指令的 CAN_ID：0x64
给 07 号驱动器在速度环模式下发参数指令的 CAN_ID：0x74
给 08 号驱动器在速度环模式下发参数指令的 CAN_ID：0x84
给 09 号驱动器在速度环模式下发参数指令的 CAN_ID：0x94
给 10 号驱动器在速度环模式下发参数指令的 CAN_ID：0xA4
给 11 号驱动器在速度环模式下发参数指令的 CAN_ID：0xB4
给 12 号驱动器在速度环模式下发参数指令的 CAN_ID：0xC4
给 13 号驱动器在速度环模式下发参数指令的 CAN_ID：0xD4
给 14 号驱动器在速度环模式下发参数指令的 CAN_ID：0xE4
给 15 号驱动器在速度环模式下发参数指令的 CAN_ID：0xF4

让 xx 号驱动器连接的电机以 temp_pwm 的限制 PWM，以 temp_speed 的速度转动：

```
则  Data[0] = (unsigned char)((temp_pwm>>8)&0xff);
    Data[1] = (unsigned char)(temp_pwm&0xff);
    Data[2] = (unsigned char)((temp_speed>>8)&0xff);
    Data[3] = (unsigned char)(temp_speed&0xff);
```

其中 temp_pwm 的取值范围为 0~+5000。
temp_speed 的取值范围为：-1000~+1000。

五．位置环模式下的参数指令：

本指令只在驱动器进入位置模式之后才有效，其他任何状态下发送本指令都会让驱动器报错。
支持连续发送本指令来修改 PWM 的限制值，运行速度值和给定的目标位置值，但连续发送的时间间隔不能小于 2 毫秒，建议以 10 毫秒为周期。

CAN_ID	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
待选择	待计算	待计算	待计算	待计算	待计算	待计算	待计算	待计算

举例说明：
给所有的驱动器在位置环模式下发参数指令的 CAN_ID：0x05（广播）
给 01 号驱动器在位置环模式下发参数指令的 CAN_ID：0x15
给 02 号驱动器在位置环模式下发参数指令的 CAN_ID：0x25
给 03 号驱动器在位置环模式下发参数指令的 CAN_ID：0x35
给 04 号驱动器在位置环模式下发参数指令的 CAN_ID：0x45
给 05 号驱动器在位置环模式下发参数指令的 CAN_ID：0x55
给 06 号驱动器在位置环模式下发参数指令的 CAN_ID：0x65
给 07 号驱动器在位置环模式下发参数指令的 CAN_ID：0x75
给 08 号驱动器在位置环模式下发参数指令的 CAN_ID：0x85
给 09 号驱动器在位置环模式下发参数指令的 CAN_ID：0x95
给 10 号驱动器在位置环模式下发参数指令的 CAN_ID：0xA5
给 11 号驱动器在位置环模式下发参数指令的 CAN_ID：0xB5
给 12 号驱动器在位置环模式下发参数指令的 CAN_ID：0xC5
给 13 号驱动器在位置环模式下发参数指令的 CAN_ID：0xD5
给 14 号驱动器在位置环模式下发参数指令的 CAN_ID：0xE5

给 15 号驱动器在位置环模式下发参数指令的 CAN_ID：0xF5

让 xx 号驱动器连接的电机以 temp_pwm 的限制电流，temp_speed 的限制速度，转动到 temp_location 的位置：

则 Data[0] = (unsigned char)((temp_pwm>>8)&0xff);
Data[1] = (unsigned char)((temp_pwm)&0xff);
Data[2] = (unsigned char)((temp_speed>>8)&0xff);
Data[3] = (unsigned char)(temp_speed&0xff);
Data[4] = (unsigned char)((temp_location>>24)&0xff);
Data[5] = (unsigned char)((temp_location>>16)&0xff);
Data[6] = (unsigned char)((temp_location>>8)&0xff);
Data[7] = (unsigned char)(temp_location&0xff);

其中 temp_pwm 的取值范围为 0~+5000。
temp_speed 的取值范围为：0~+1000。
temp_location 的取值范围为：-5000000~+5000000。

六．配置指令：

配置指令可以决定是否让驱动器以某个固定的时间间隔通过 CAN 总线对外发送电机当前的实时电流、速度、位置值等信息。
本指令在任何状态下都可以生效。但驱动器只在进入 PWM 模式 1、PWM 模式 2、速度模式、位置模式等四种模式之后，对外发送以上所述的电流、速度、位置等信息。
驱动器通过 CAN 总线发送以上信息的周期，以 10 毫秒为最小单位。

CAN_ID	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
待选择	待给定	0x55	0x55	0x55	0x55	0x55	0x55	0x55

举例说明：
给所有驱动器发配置指令的 CAN_ID：0x06（广播）
给 01 号驱动器发配置指令的 CAN_ID：0x16
给 02 号驱动器发配置指令的 CAN_ID：0x26
给 03 号驱动器发配置指令的 CAN_ID：0x36
给 04 号驱动器发配置指令的 CAN_ID：0x46
给 05 号驱动器发配置指令的 CAN_ID：0x56
给 06 号驱动器发配置指令的 CAN_ID：0x66
给 07 号驱动器发配置指令的 CAN_ID：0x76
给 08 号驱动器发配置指令的 CAN_ID：0x86
给 09 号驱动器发配置指令的 CAN_ID：0x96
给 10 号驱动器发配置指令的 CAN_ID：0xA6
给 11 号驱动器发配置指令的 CAN_ID：0xB6
给 12 号驱动器发配置指令的 CAN_ID：0xC6
给 13 号驱动器发配置指令的 CAN_ID：0xD6
给 14 号驱动器发配置指令的 CAN_ID：0xE6
给 15 号驱动器发配置指令的 CAN_ID：0xF6

举例：
让 02 号驱动器以 100 毫秒为周期的对外发送电流、速度、位置等信息的指令为：
CAN_ID：0x26
Data[0] = 10; //给定数据*10 毫秒 = 发送周期，单位为毫秒。
Data[1] = 0x55; //固定填充值
Data[2] = 0x55; //固定填充值
Data[3] = 0x55; //固定填充值
Data[4] = 0x55; //固定填充值
Data[5] = 0x55; //固定填充值
Data[6] = 0x55; //固定填充值
Data[7] = 0x55; //固定填充值

七．数据反馈：

以下是驱动器对外发送电流、速度、位置等信息的 CAN 消息的格式。特别注意，这条 CAN 消息是由驱动器发出，发出的周期可以通过上述的配置指令来确定。

CAN_ID	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
待接收	待接收	待接收	待接收	待接收	待接收	待接收	待接收	待接收

举例说明：（由于接收方是主控，所以没有广播发送的指令）
01 号驱动器数据反馈的 CAN_ID：0x17
02 号驱动器数据反馈的 CAN_ID：0x27
03 号驱动器数据反馈的 CAN_ID：0x37
04 号驱动器数据反馈的 CAN_ID：0x47
05 号驱动器数据反馈的 CAN_ID：0x57
06 号驱动器数据反馈的 CAN_ID：0x67
07 号驱动器数据反馈的 CAN_ID：0x77
08 号驱动器数据反馈的 CAN_ID：0x87
09 号驱动器数据反馈的 CAN_ID：0x97
10 号驱动器数据反馈的 CAN_ID：0xA7
11 号驱动器数据反馈的 CAN_ID：0xB7
12 号驱动器数据反馈的 CAN_ID：0xC7
13 号驱动器数据反馈的 CAN_ID：0xD7
14 号驱动器数据反馈的 CAN_ID：0xE7
15 号驱动器数据反馈的 CAN_ID：0xF7

举例：
02 号驱动器当前的电流值是 real_current，当前的速度值是 real_speed，当前的位置是 real_location，则驱动器则会对外发送如下的 CAN 消息：
CAN_ID:0x27
Data[0] = (unsigned char)((real_current>>8)&0xff);
Data[1] = (unsigned char)(real_current&0xff);
Data[2] = (unsigned char)((real_speed>>8)&0xff);

```
Data[3] = (unsigned char)(real_speed&0xff);
Data[4] = (unsigned char)((real_location>>24)&0xff);
Data[5] = (unsigned char)((real_location>>16)&0xff);
Data[6] = (unsigned char)((real_location>>8)&0xff);
Data[7] = (unsigned char)(real_location&0xff);
```

对于主控而言，还原电流、速度、位置的反馈值，可以如下：

```
int real_current = (Data[0]<<8)|Data[1];
int real_speed = (Data[2]<<8)|Data[3];
int real_location = (Data[4]<<24)| (Data[5]<<24)| (Data[6]<<24)| Data[7];
```

此项功能为新固件增加的功能，用户可以利用此项功能完成更丰富的功能，例如如下所述：

1．可以利用电流反馈值来监测母线电流的值，以此可以在主控上设计一个长时堵转保护功能。

2．可以利用速度反馈，来分析带负载情况下速度的变化曲线。

3．可以利用位置反馈，来检测位置环的执行程度，监测位置是否到位，以便设计一个时间紧凑的执行流程。

安装尺寸说明

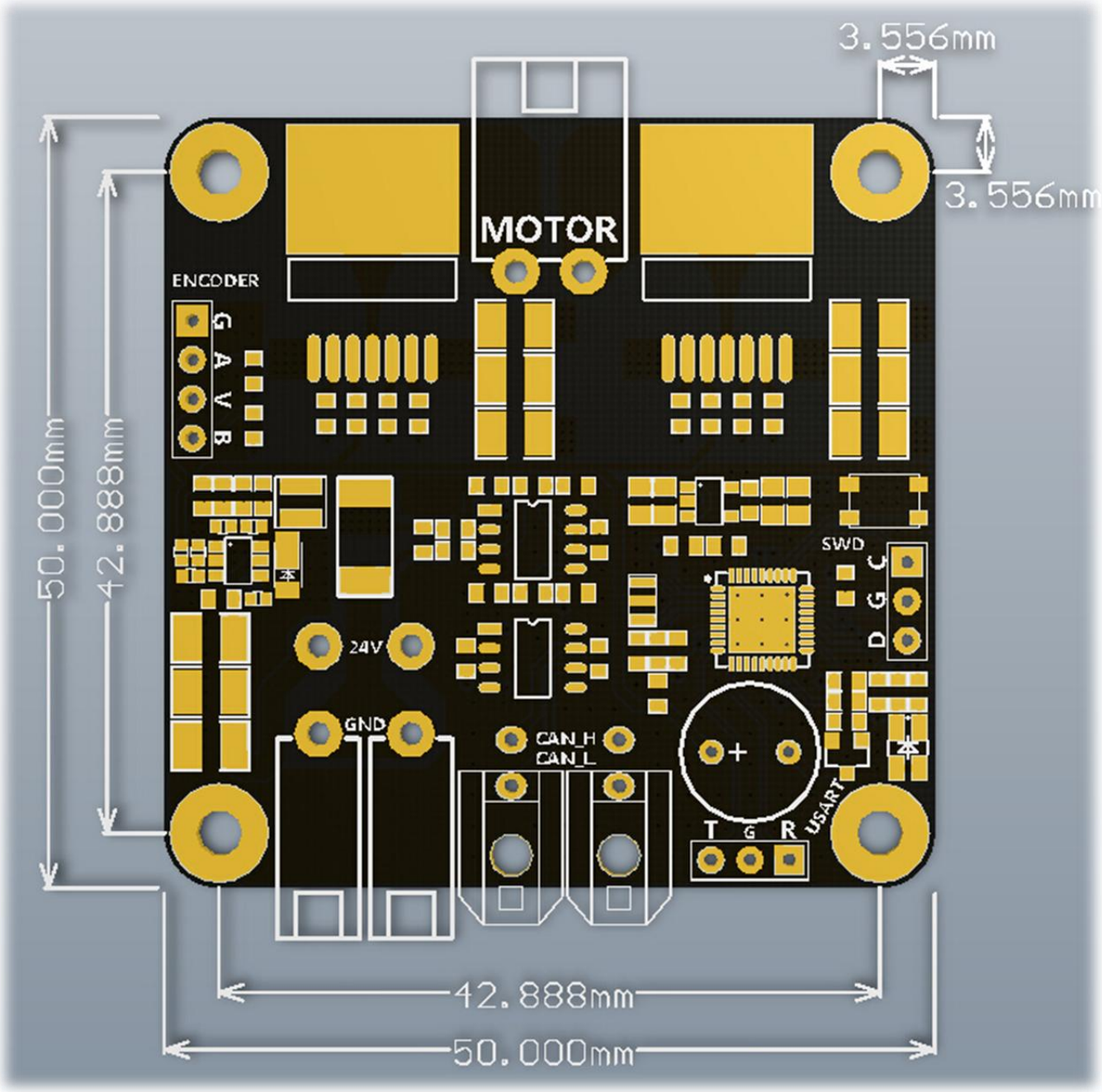


图 10 驱动器安装尺寸图

图 10 所示，为本产品的安装尺寸图。

本产品为 50mm*50mm 正方形的外形，上下左右都对称的安放四个固定孔，固定孔的内径为 3mm，外径的铜环为 6.3mm，适用于普通的 M3 型号的螺丝螺母。

相邻的固定孔与固定孔之间的间距为 42.888mm，固定孔的中心与板边界的距离为 3.556mm。

PCB 板的厚度为 1.2mm，正面最高的器件突出高度为 12mm，背面最高的器件突出高度为 2mm。

材料清单

一、实物部分

1. RoboModule 伺服电机驱动器主板 x1
2. 电源连接座的插头 x2
3. CAN 口连接座的插头 x2
4. 对应的电机和电源连接座插头的冷压端子 x6
5. 对应的 CAN 口连接座插头的冷压端子 x4
6. 配套的编码器线（2.54mm 间距的杜邦线 5pin 转 4pin）x1

二、资料部分

1. RoboModule 伺服电机驱动器用户手册
2. 用于控制本驱动器的 CAN 库函数（STM32F103 和 STM32F405 两种版本）
3. RoboModule 伺服电机驱动器调试软件

另外，需要自备的东西如下：

1．USB 转串口（TTL）小板（PL2303,CP2102,FT232 驱动的皆可。推荐后两种，因为 PL2303 芯片不稳定）

2．直流有刷伺服电机（带编码器可以控制速度环和位置环，不带编码器的只能控制 PWM 模式开环）

3．3A 或 5A 的电源箱（如果没有，直接用电池也可以）

4．压线钳（用于做线材，如果没有，就用普通尖嘴钳）

FAQ 问题解答

1．在配套的 RoboModule 伺服电机驱动器调试软件上设置好参数之后，断电会保存吗？

答：RoboModule 伺服电机驱动器调试软件上面配置的所有参数都是写入 Flash 的，可以在掉电以后永久保存，所以配置一遍即可。但是下次如果更换了电机的线或者编码器的线，或者更换了电机，则需要运行 RoboModule 伺服电机驱动器调试软件进行重新配置，因为不能保证电机接线方向和编码器接线方向与原来的完全一致。

2．CAN 通信，指令发不出去是什么情况？

答：检查驱动器的 CAN_H 是否连在主控的 CAN_H 上，驱动的 CAN_L 是否连在主控的 CAN_L 上，CAN 线的两根线，最好类似双绞线一样缠绕，以保证差分电平的稳定性。

断电后，用万用表测量已经连好的 CAN 线上，CAN_H 和 CAN_L 之间的电阻值是否是 60 欧姆左右，如果不是 60 欧姆，则检查 CAN 总线上是否只有首尾两个 120 欧姆的电阻，多余的电阻必须拆掉。

3．PWM 开环、速度环、位置环存在什么耦合关系？

答：根据实际的调试效果，我们的控制环路耦合关系确定如下：

PWM 开环—速度环—位置环。

也就是说，位置环是基于速度环的基础上建立的，而速度环是基于 PWM 开环来建立的。

4．差分式编码器，如何接入到 RoboModule 伺服电机驱动器？

答：如果用户使用的编码器是 10 根灰色排线的差分编码器，是不可以直接把信号输入到驱动器的。可以购买本公司的“差分编码器电平信号转换器”来转换电平后再输入给电机驱动器。

5．使用 CAN 发送的指令，指令顺序是什么样的？

答：首先使用发送复位指令，此时驱动器进入复位状态。

等待 1 秒钟，驱动器复位完毕。

第二步，发送模式选择指令，让驱动器进入 PWM 模式、或者速度、位置模式。

等待 500ms，驱动器进入模式就绪。

第三步，周期性发送数据指令。（数据指令需要更新的时候就可以继续发送，注意数据指令的发送间隔，推荐使用 50Hz 或者 100Hz）

6．在 RoboModule 伺服电机驱动器调试软件上运行电机时候，电机已经转动，但窗口返回的编码器数据一直为零是什么原因？

答：你的编码器线可能接反了。按照上面编码器章节所示，重新检查。如果还是不行，那么很遗憾的告诉你，你的编码器可能烧了。

联系方式

RoboModule 研发小组
官方网站： http://www.robomodule.net/
淘宝店铺网址：http://shop109954302.taobao.com/

售后热线：0755-86225277
联系邮箱：carey.lin@robomodule.net
阿里旺旺：cc_robot