1. 机械安装

拆掉Gluon末端电机的限位螺丝，然后将机械爪与末端电机的连接处对齐，用M3螺丝固定



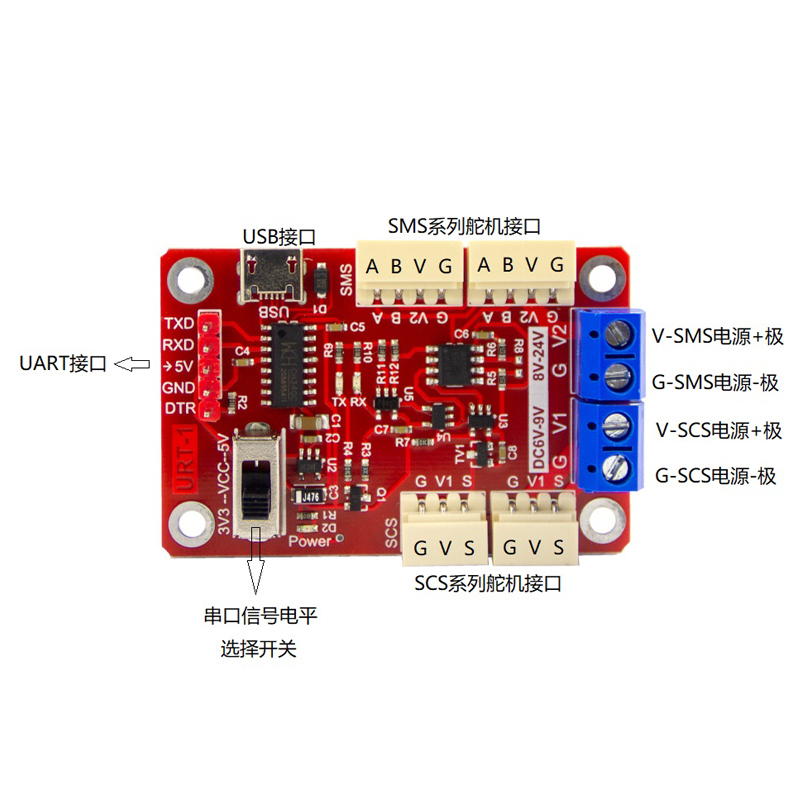
1. 电气连接

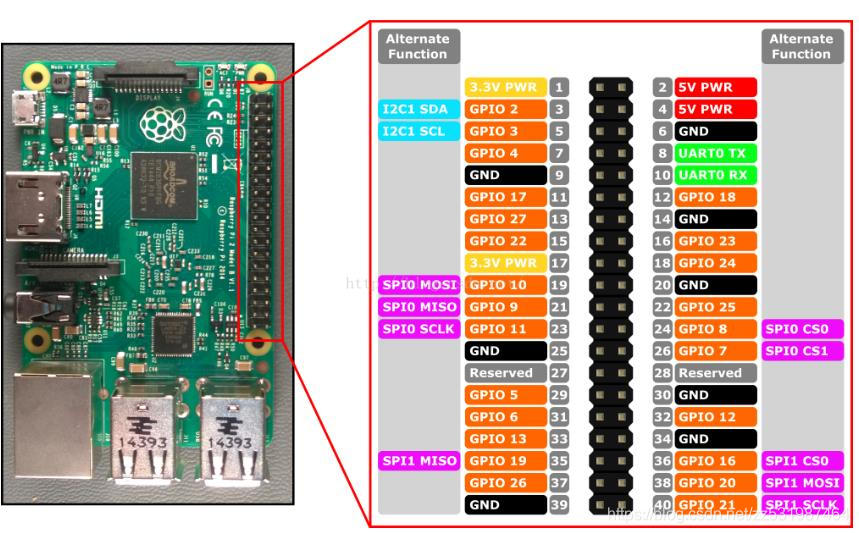
将机械爪电机的引出线接入转接板对应接口，连接电源（SMS电源接口），将转接板与树莓派的对应串口接口相连接

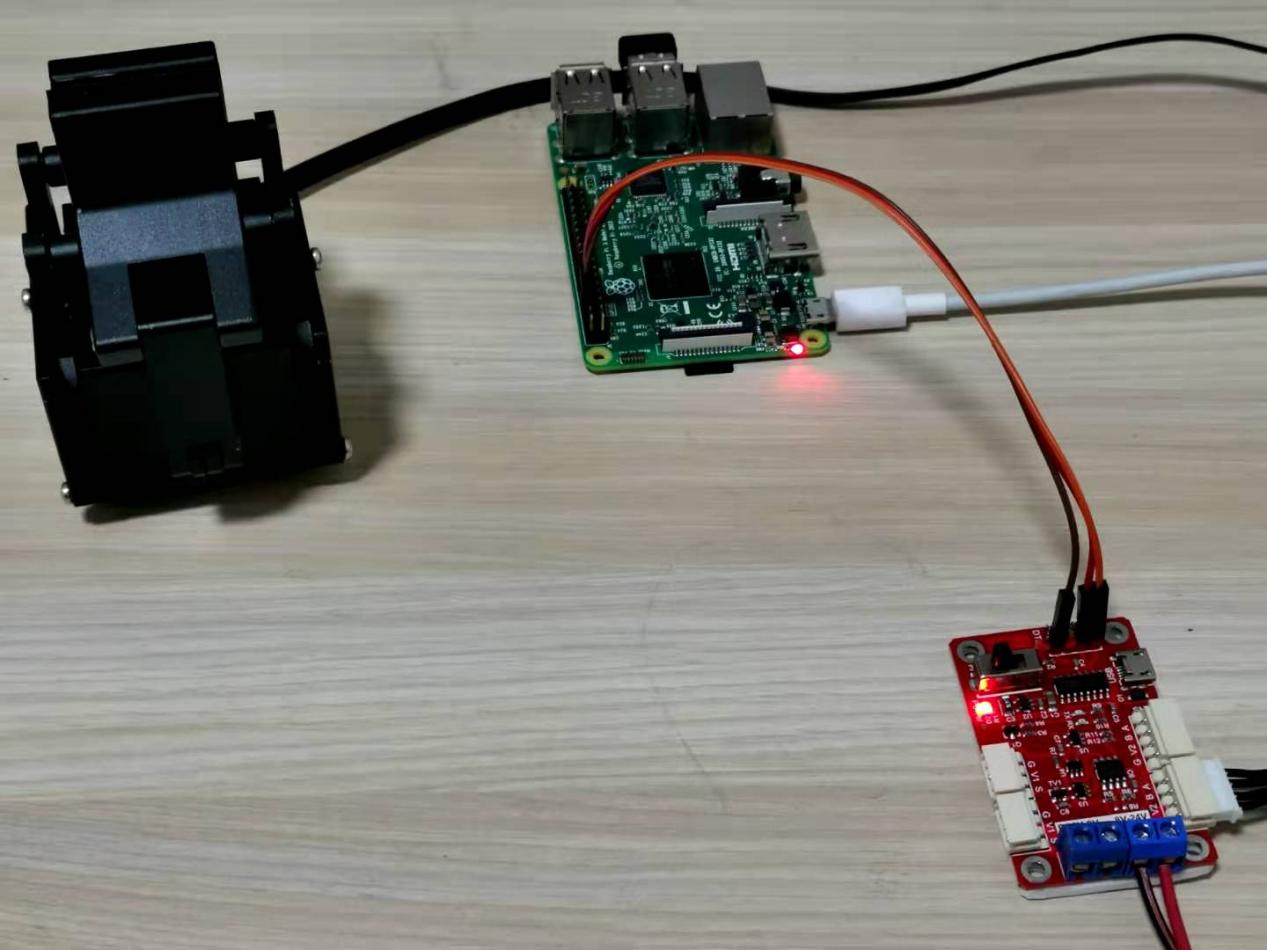
GND——GND

TXD——RX（pin 10）

RXD——TX（pin 8）





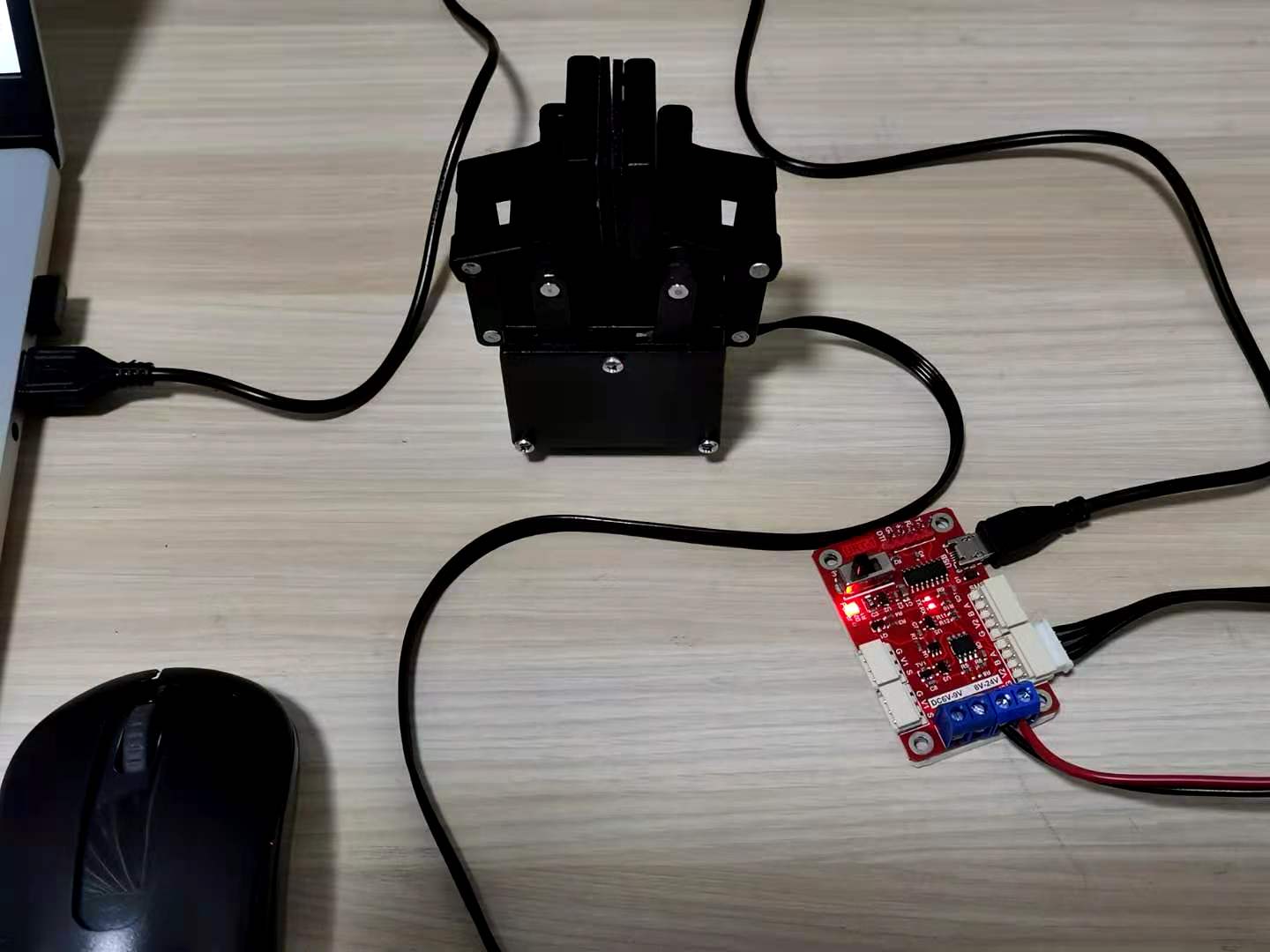


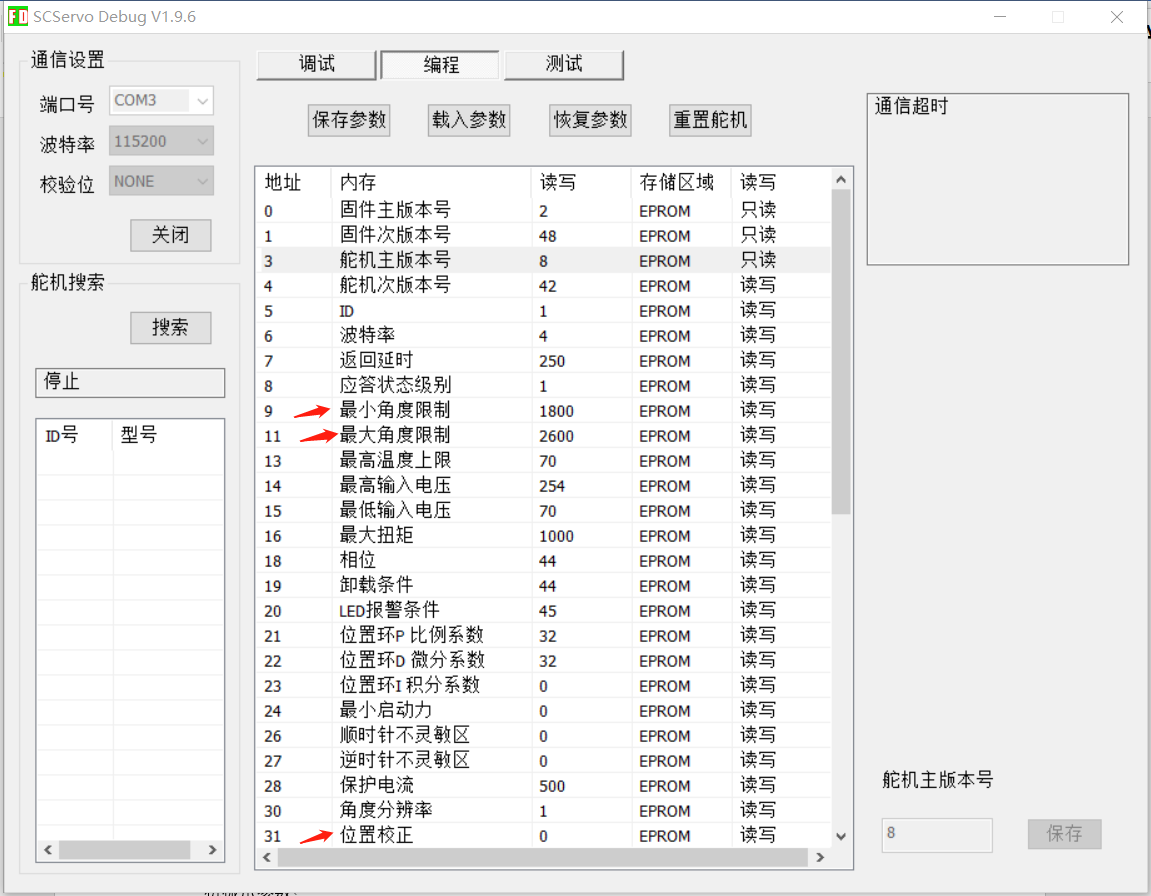
1. SDK使用

见附件

四、机械夹调试

将转接板USB接口接入PC USB接口，打开控制软件，打开串口，搜索，电机搜索出的电机，电机位置进度条(位置限制在0-800,其中0代表机械夹闭合状态，800代表机械夹开口最大状态)，可以通过修改最小角度限制、最大角度限制来控制机械夹最大开合时的位置参数。使用SDK时根据控制旋转角度大小来控制机械夹开合大小，还可以通过修改保护电流和保护时间来控制机械夹夹持力大小，若机械夹开合状态偏离对应的角度，可以通过修改位置校正参数来校正对应关系。具体参数修改可参照“SMS1.0内存表(180130)”





机械爪参数

|  |  |  |  |
| --- | --- | --- | --- |
| 重量 | 370g | 供电电压 | 12~24V |
| 尺寸 | 68mm\*56mm\*123mm | 静态电流 | 17mA@24V |
| 最大夹持力 | 30N | 空载电流 | 160 mA@24V |
| 最大行程 | 90mm | 堵转电流 | 2300 mA@24V |
| 电机 | 无刷减速电机 | 控制频率 | 333Hz |
| 行程分辨率 | 90mm/800 | 通信接口 | UART |
| 材质 | 铝合金 | 支持系统 | Linux/Windows |
| 最大开合时间 | 0.5s | 支持语言 | C++/Python/Arduino |

**附录：**

如何让树莓派与其他设备使用串口进行通信呢？

配置这个功能可以让我们的树莓派与其他设备例如stm32进行数据的交互和通信，实现两者之间的互相控制功能，这个还是非常实用的！

使用以下步骤即可：

树莓派包含两个串口

1.硬件串口（**/dev/ttyAMA0**）,硬件串口由硬件实现，有单独的波特率时钟源，性能高，可靠。一般优先选择这个使用。

2.mini串口（**/dev/ttyS0**），mini串口时钟源是由CPU内核时钟提供，波特率受到内核时钟的影响，不稳定。

想要通过树莓派的GPIO引脚进行稳定的串口通信，需要修改串口的映射关系。

serial0是GPIO引脚对应的串口，serial1是蓝牙对应的串口，可以使用 **ls -l /dev/serial\***查看当前的映射关系

**1.sudo raspi-config**

Interfacing Options -> P6 Serial, 第一个选项**（would you like a login shell to be accessible  over serial?）**选择NO，第二个选项**（would you like the serial port hardware to be enabled?）**选择 YES

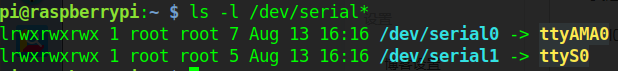
然后我们可以通过使用ls -l /dev/serial\* 查看串口映射关系发现

serial0（GPIO串口）默认使用的是ttyS0(mini串口)，serial1（蓝牙）使用的是ttyAMA0（硬件串口）

如果想使用稳定可靠的硬件串口，就要将树莓派3的硬件串口与mini串口默认映射对换。而这个需求官方也考虑到了，在系统中放了一个实现这个功能的文件---stretch版本系统中的文件为/boot/overlays/pi3-miniuart-bt.dtbo。使用该文件发挥功能只需在/boot/config.txt文件末尾添加一行代码  
**sudo nano /boot/config.txt**

在文件最后面添加一行：**dtoverlay=pi3-miniuart-bt**

修改完成后**sudo reboot**重启树莓派后使用 **ls -l /dev/serial\***查看映射如下：



如上图所示，serial0（GPIO串口）使用的是ttyAMA0(硬件串口)，而serial1（蓝牙）使用的是ttys0(mini串口)

**2.禁用串口的控制台功能**

前面步骤已经交换了硬件串口与mini串口的映射关系，但是，现在还不能使用树莓派串口模块与电脑进行通信，因为，树莓派io口引出串口默认是用来做控制台使用的，即是为了用串口控制树莓派，而不是通信。所以我们要禁用此默认设置。

首先执行命令如下：

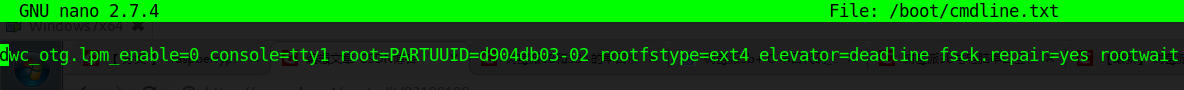
1. sudo systemctl stop serial-getty@ttyAMA0.service
2. sudo systemctl disable serial-getty@ttyAMA0.service

然后执行命令行：

sudo nano /boot/cmdline.txt

删除语句 console=serial0,115200  （没有的话就不需要此步骤）

保存之后其内容如下：



接下来再次重启树莓派即可。

最终的/boot/config.txt文件内容如下：



**3.测试验证串口功能**

代码如下：

1. #include <stdio.h>
2. #include <wiringPi.h>
3. #include <wiringSerial.h>
5. **int** **main**()
6. {
7. **int** fd;
8. **if**(wiringPiSetup()<0) {
9. **return** 1;
10. }
12. **if**((fd=serialOpen("/dev/ttyAMA0",115200))<0) {
13. **return** 1;
14. }
16. printf("serial test start ...\n");
17. serialPrintf(fd,"hello world!!\n");
19. serialClose(fd);
20. **return** 0;
21. }

编译命令：

gcc test.c -o test -lwiringPi

代码中使用到wiringPi库，具体该库的下载和使用可以参考官网：<http://wiringpi.com/>，里面描述了如何使用该库，这里不做介绍

执行：./test

我们可以将串口模块通过杜邦线插到树莓派的UART0 TX RX接入到PC机，在PC机上打开串口调试助手查看 执行上面命令后会打印 hello world 的字符串，说明通信成功。

同理，我们可以在树莓派上安装minicom，配置好 /dev/ttyAMA0 后在串口调试助手上发送字符串，树莓派minicom串口也能收到对应的字符串，说明两者通信成功。