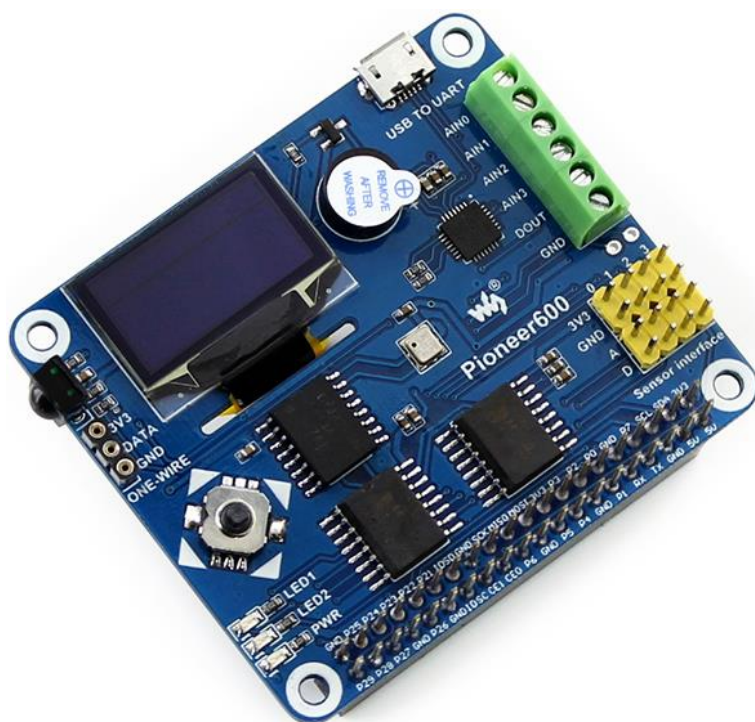




# PIONEER600 树莓派扩展板用户手册

## 产品介绍

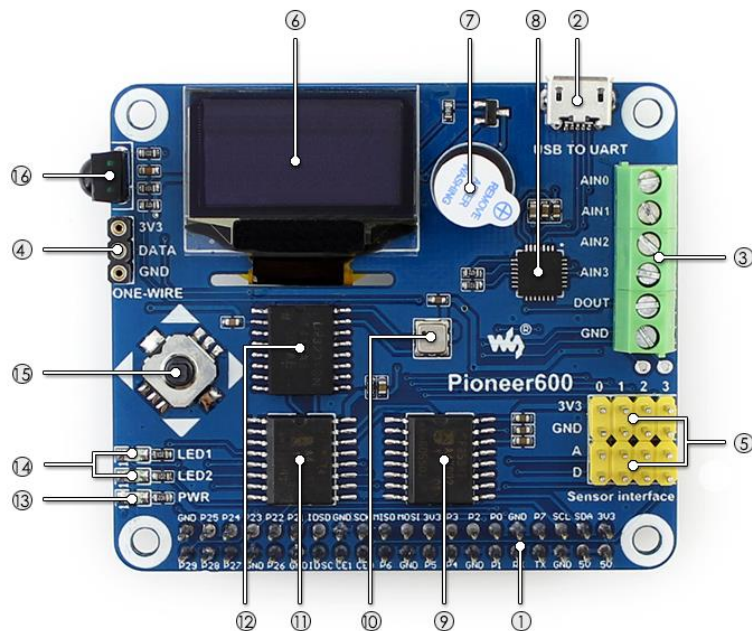
Pioneer600 是一款 Raspberry Pi A+ / B+ / 2 / 3 代 B 的外围扩展板，拥有丰富的板载资源，是你学习 Raspberry Pi 控制编程的好拍档。



- **【简易 I/O 设备】** 双 LED、五向摇杆、蜂鸣器，Raspberry Pi 入门常用（标准姿势）
- **【USB TO UART】** CP2102，支持串口终端控制 Raspberry Pi（实用主义）
- **【显示设备】** 0.96 寸 OLED，小屏亦显大精彩（开始玩花的）
- **【RTC】** DS3231，高精度，并带备用电池座（贴心）
- **【AD/DA】** PCF8591，8 位精度，并带接线端子输入输出接口（友尽）
- **【GPIO 扩展】** PCF8574，确保 GPIO 够用（妈妈不再担心）
- **【红外接收】** LFN0038K，支持遥控 Raspberry Pi（单手控制）
- **【压力传感器】** BMP180，可检测气压与温度（压力山大）
- **【WIRE 接口】** 可接入 1-WIRE 器件，配有 DS18B20（爽一下）
- **【传感器接口】** 方便接入各类传感器（爽歪歪）



## 资源介绍



### [ 核心接口简介 ]

1. **Raspberry Pi GPIO 接口**  
方便接入 RPi
2. **USB TO UART 接口**  
方便通过串口终端控制 Raspberry Pi
3. **AD/DA 输入输出接口（接线端子）**  
方便在各种场合使用
4. **ONE-WIRE 接口**  
可接入 DS18B20 等单总线设备
5. **传感器接口**  
方便接入各类传感器

### [ 器件简介 ]

6. **0.96 寸 OLED**  
驱动芯片为 SSD1306，128×64 分辨率，SPI 接口

7. **蜂鸣器**
8. **CP2102**  
USB 转串口芯片
9. **PCF8591**  
8 位精度 AD/DA 芯片，I2C 接口
10. **BMP180**  
压力传感器，I2C 接口
11. **PCF8574**  
I/O 扩展芯片，I2C 接口
12. **DS3231**  
高精度 RTC 芯片，I2C 接口
13. **电源 LED**
14. **用户 LED**
15. **五向摇杆**
16. **LFN0038K 红外接收头**



## 目录

产品介绍.....	1
资源介绍.....	2
目录 3	
1. 准备工作 - 安装必要的函数库.....	6
1.1. 安装必要的函数库 .....	6
1.2. 连接扩展板和树莓派 .....	6
2. LED 示例程序 - 改变 LED 灯状态演示.....	7
2.1. bcm2835 程序.....	7
2.2. wiringPi 程序.....	7
2.3. sysfs 程序.....	8
2.4. python 程序.....	8
3. 按键示例程序 - 按下摇杆.....	9
3.1. bcm2835 程序.....	9
3.2. wiringPi 程序.....	9
3.3. python 程序.....	10
4. PCF8574 示例程序 - I/O 扩展演示.....	11
4.1. bcm2835 程序.....	11
4.2. python 程序.....	11
4.3. fs 程序.....	12
4.4. wiringPi 程序.....	12





5.	BMP180 示例程序 - 气压传感器演示 .....	14
5.1.	bcm2835 程序 .....	14
5.2.	wiringPi 程序 .....	14
5.3.	python 程序 .....	15
6.	DS3231 示例程序 - 高精度 RTC 演示 .....	16
6.1.	bcm2835 程序 .....	16
6.2.	wiringPi 程序 .....	16
6.3.	python 程序 .....	17
7.	DS18B20 示例程序 - 温度传感器演示 .....	18
7.1.	sysfs 程序 .....	18
7.2.	python 程序 .....	18
8.	IRM 示例程序 - 红外感应程序 .....	19
8.1.	bcm2835 程序 .....	19
8.2.	wiringPi 程序 .....	19
8.3.	python 程序 .....	20
9.	UART 示例程序 - 发送串口接收到的数据 .....	21
9.1.	wiringPi 程序 .....	21
9.2.	python 程序 .....	21
10.	OLED 示例程序 .....	22
10.1.	bcm2835 程序 .....	22
10.2.	wiringPi 程序 .....	22
10.3.	python 程序 .....	22



11. Pioneer600 连接传感器套件（需另外选购） .....	24
12. Color Sensor 示例程序 - 颜色传感器演示 .....	24
13. Flame Sensor 示例程序 - 火焰传感器演示 .....	24
14. Hall Sensor 示例程序 - 霍尔传感器演示 .....	25
15. Infrared Reflective Sensor 示例程序 - 一体式红外发送接收管演示 .....	25
16. Laser Sensor 示例程序 - 激光传感器演示 .....	26
17. Moisture Sensor 示例程序 - 土壤湿度传感器演示 .....	26
18. Rotation Sensor 示例程序 - 旋转编码器演示 .....	27
19. Sound Sensor 示例程序 - 声音探测器演示 .....	27
20. Temperature-Humidity Sensor 示例程序 - 温湿度传感器演示 .....	28
21. MQ-5 Gas Sensor 示例程序 - MQ-5 气体传感器演示 .....	28
22. Tilt Sensor 示例程序 - 倾斜传感器演示 .....	29
23. UV Sensor 示例程序 - 紫外线传感器演示 .....	29
24. Liquid Level Sensor 示例程序 - 液位传感器演示 .....	30



## 1. 准备工作 - 安装必要的函数库

### 1.1. 安装必要的函数库

在使用本扩展板之前，我们需要给树莓派安装 `bcm2835`，`wiringPi` 和 `python` 函数库，以给树莓派提供官方原生系统之外的 API 接口。在安装完函数库之后，还需要设置启动 I<sup>2</sup>C，SPI，UART 等内核驱动。树莓派的库函数安装和配置过程请参见[为树莓派安装必要的函数库](#)。

安装好函数库并且启动了 I<sup>2</sup>C，SPI，UART 等内核驱动之后，您还需要下载产品的示例程序。这些示例程序都需要解压并放在 `/home/pi` 目录下（您可以通过 U 盘把它们复制到树莓派中）。部分示例程序拥有不同的实现方式（例如 `bcm2835`，`wiringPi`，`sysfs`，`python` 等方式），即使实现同一个功能也可能依赖不同的函数库。因此使用之前请务必安装好 `bcm2835`，`wiringPi` 和 `python` 函数库。那么之后就可以开始探索本产品提供的各种扩展功能了。

更加详细的说明请看[树莓派系列教程](#)。

如果安装完函数库之后依然无法执行以下的示例程序，那么您可以尝试使用 `chmod +x filename` 命令给程序赋予可执行权限。

### 1.2. 连接扩展板和树莓派

Pioneer600 扩展板可以支持树莓派 A+ / B+ / 2 / 3 代 B，在使用示例程序之前，请把扩展板和树莓派连接好，树莓派需要插上 USB 电源供电。如果只有本扩展板而没有树莓派主板的话，那么示例程序是无法使用的。

连接树莓派 A+：



链接树莓派 B+ / 2 / 3 代 B





## 2. LED 示例程序 - 改变 LED 灯状态演示

### 2.1. bcm2835 程序

进入 Linux 终端，在终端执行以下命令。

编译程序：

```
pi@raspberrypi ~/Pioneer600/LED/bcm2835 $ make
```

执行程序：

```
pi@raspberrypi ~/Pioneer600/LED/bcm2835 $ sudo ./led
```

预期结果：可以看到 LED1 闪烁。

按下 **Ctrl+C** 结束程序。

### 2.2. wiringPi 程序

进入 Linux 终端，在终端执行以下命令。

编译程序：

```
pi@raspberrypi ~/Pioneer600/LED/wiringPi $ make
```

执行程序：

```
pi@raspberrypi ~/Pioneer600/LED/wiringPi $ sudo ./led
```

预期结果：可以看到 LED1 闪烁。

按下 **Ctrl+C** 结束程序。

进入 Linux 终端，在终端执行以下命令。

编译程序：

```
pi@raspberrypi ~/Pioneer600/LED/wiringPi $ make
```

执行程序：

```
pi@raspberrypi ~/Pioneer600/LED/wiringPi $ sudo ./pwm
```

预期结果：可以看到 LED1 亮度逐渐改变。

按下 **Ctrl+C** 结束程序。





### 2.3. sysfs 程序

进入 Linux 终端，在终端执行以下命令。

编译程序：

```
pi@raspberrypi ~/Pioneer600/LED/fs $ make
```

执行程序：

```
pi@raspberrypi ~/Pioneer600/LED/fs $ sudo ./led
```

预期结果：LED1 闪烁 10 次后自动退出程序。

按下 **Ctrl+C** 结束程序。

### 2.4. python 程序

在终端输入以下命令执行程序：

```
pi@raspberrypi ~/Pioneer600/LED/python $ sudo python led.py
```

预期结果：可以看到 LED1 闪烁。

按下 **Ctrl+C** 结束程序。

在终端输入以下命令执行程序：

```
pi@raspberrypi ~/Pioneer600/LED/python $ sudo python pwm.py
```

预期结果：可以看到 LED1 亮度逐渐改变。

按下 **Ctrl+C** 结束程序。





### 3. 按键示例程序 - 按下摇杆

#### 3.1. bcm2835 程序

进入 Linux 终端，在终端执行以下命令。

编译程序：

```
pi@raspberrypi ~/Pioneer600/KEY/bcm2835 $ make
```

执行程序：

```
pi@raspberrypi ~/Pioneer600/KEY/bcm2835 $ sudo ./key
```

预期结果：

按下摇杆中间键，终端会输出以下信息。

```
Key Test Program!!!  
KEY PRESS  
KEY PRESS  
KEY PRESS
```

按下 **Ctrl+C** 结束程序。

#### 3.2. wiringPi 程序

进入 Linux 终端，在终端执行以下命令。

编译程序：

```
pi@raspberrypi ~/Pioneer600/KEY/wiringPi $ make
```

执行程序：

```
pi@raspberrypi ~/Pioneer600/KEY/wiringPi $ sudo ./key
```

预期结果：

按下摇杆中间键，终端会输出以下信息。

```
Key Test Program!!!  
KEY PRESS  
KEY PRESS  
KEY PRESS
```



按下 **Ctrl+C** 结束程序。

### 3.3. python 程序

在终端输入以下命令执行程序：

```
pi@raspberrypi ~/Pioneer600/KEY/python $ sudo ./key.py
```

预期结果：

按下摇杆中间键，终端会输出以下信息。

```
Key Test Program
KEY PRESS
KEY PRESS
KEY PRESS
```

按下 **Ctrl+C** 结束程序。



## 4. PCF8574 示例程序 - I/O 扩展演示

### 4.1. bcm2835 程序

进入 Linux 终端，在终端执行以下命令。

编译程序：

```
pi@raspberrypi ~/Pioneer600/PCF8574/bcm2835 $ make
```

执行程序：

```
pi@raspberrypi ~/Pioneer600/PCF8574/bcm2835 $ sudo ./led
```

预期结果：可以看到 LED2 闪烁。

按下 **Ctrl+C** 结束程序。

进入 Linux 终端，在终端执行以下命令。

编译程序：

```
pi@raspberrypi ~/Pioneer600/PCF8574/bcm2835 $ make
```

执行程序：

```
pi@raspberrypi ~/Pioneer600/PCF8574/bcm2835 $ sudo ./pcf8574
```

预期结果：按下方向按键时，LED2 会点亮，蜂鸣器响，终端会输出以下信息。

```
PCF8574 Test Program !!!
```

```
up
```

```
left
```

```
down
```

```
right
```

按下 **Ctrl+C** 结束程序。

### 4.2. python 程序

在终端输入以下命令执行程序：

```
pi@raspberrypi ~/Pioneer600/PCF8574/python $ sudo python led.py
```

预期结果：可以看到 LED2 闪烁。



按下 **Ctrl+C** 结束程序。

在终端输入以下命令执行程序：

```
pi@raspberrypi ~/Pioneer600/PCF8574/python $ sudo python pc8574.py
```

预期结果：按下方向按键时，LED2 会点亮，蜂鸣器响，终端会输出以下信息。

```
PCF8574 Test Program !!!
up
left
down
right
```

按下 **Ctrl+C** 结束程序。

### 4.3. fs 程序

进入 Linux 终端，在终端执行以下命令。

编译程序：

```
pi@raspberrypi ~/Pioneer600/PCF8574/fs $ make
```

执行程序：

```
pi@raspberrypi ~/Pioneer600/PCF8574/fs $ sudo ./led
```

预期结果：可以看到 LED2 闪烁。

按下 **Ctrl+C** 结束程序。

### 4.4. wiringPi 程序

进入 Linux 终端，在终端执行以下命令。

编译程序：

```
pi@raspberrypi ~/Pioneer600/PCF8574/wiringPi $ make
```

执行程序：

```
pi@raspberrypi ~/Pioneer600/PCF8574/wiringPi $ sudo ./led
```

预期结果：可以看到 LED2 闪烁。





按下 **Ctrl+C** 结束程序。

进入 Linux 终端，在终端执行以下命令。

编译程序：

```
pi@raspberrypi ~/Pioneer600/PCF8574/wiringPi $ make
```

执行程序：

```
pi@raspberrypi ~/Pioneer600/PCF8574/wiringPi $ sudo ./LED
```

预期结果：可以看到 LED2 闪烁。

按下 **Ctrl+C** 结束程序。



## 5. BMP180 示例程序 - 气压传感器演示

### 5.1. bcm2835 程序

进入 Linux 终端，在终端执行以下命令。

编译程序：

```
pi@raspberrypi ~/Pioneer600/BMP180/bcm2835 $ make
```

执行程序：

```
pi@raspberrypi ~/Pioneer600/BMP180/bcm2835 $ sudo ./BMP180
```

预期结果：终端会输出以下信息。

```
BMP180 Test Program ...
Temperature: 34.20 C
Pressure:    1005.12 Pa
Altitude:    67.66 m
```

按下 **Ctrl+C** 结束程序。

### 5.2. wiringPi 程序

进入 Linux 终端，在终端执行以下命令。

编译程序：

```
pi@raspberrypi ~/Pioneer600/BMP180/wiringPi $ make
```

执行程序：

```
pi@raspberrypi ~/Pioneer600/BMP180/wiringPi $ sudo ./BMP180
```

预期结果：终端会输出以下信息。

```
BMP180 Test Program ...
Temperature: 34.20 C
Pressure:    1005.12 Pa
Altitude:    67.66 m
```

按下 **Ctrl+C** 结束程序。



### 5.3. python 程序

在终端输入以下命令执行程序：

```
pi@raspberrypi ~/Pioneer600/BMP180/python $ sudo python BMP180_example.py
```

预期结果：终端会输出以下信息。

```
Temperature: 34.20 C
Pressure:    1005.12 Pa
Altitude:    67.66 m
```

按下 **Ctrl+C** 结束程序。



## 6. DS3231 示例程序 - 高精度 RTC 演示

### 6.1. bcm2835 程序

进入 Linux 终端，在终端执行以下命令。

编译程序：

```
pi@raspberrypi ~/Pioneer600/DS3231/bcm2835 $ make
```

执行程序：

```
pi@raspberrypi ~/Pioneer600/DS3231/bcm2835 $ sudo ./ds3231
```

预期结果：终端会输出以下信息。

```
start.....
2015/08/12 18:00:00 Wed
2015/08/12 18:00:01 Wed
2015/08/12 18:00:02 Wed
2015/08/12 18:00:03 Wed
```

按下 **Ctrl+C** 结束程序。

### 6.2. wiringPi 程序

进入 Linux 终端，在终端执行以下命令。

编译程序：

```
pi@raspberrypi ~/Pioneer600/DS3231/wiringPi $ make
```

执行程序：

```
pi@raspberrypi ~/Pioneer600/DS3231/wiringPi $ sudo ./ds3231
```

预期结果：终端会输出以下信息。

```
start.....
2015/08/12 18:00:00 Wed
2015/08/12 18:00:01 Wed
2015/08/12 18:00:02 Wed
2015/08/12 18:00:03 Wed
```





按下 **Ctrl+C** 结束程序。

### 6.3. python 程序

在终端输入以下命令执行程序：

```
pi@raspberrypi ~/Pioneer600/DS3231/python $ sudo python ds3231.py
```

预期结果：终端会输出以下信息。

```
2015/08/12 18:00:00 Wed
2015/08/12 18:00:01 Wed
2015/08/12 18:00:02 Wed
2015/08/12 18:00:03 Wed
2015/08/12 18:00:04 Wed
```

按下 **Ctrl+C** 结束程序。



## 7. DS18B20 示例程序 - 温度传感器演示

DS18B20 程序需在树莓派/boot/config.txt 文件最后面添加 `dtoverlay=w1-gpio-pullup` 这一句，重启生效后才可使用，详细内容请看[树莓派系列教程](#)。

### 7.1. sysfs 程序

在终端输入以下命令执行程序：

```
pi@raspberrypi ~/Pioneer600/DS18B20/fs $ sudo ./ds18b20
```

预期结果：终端会输出以下信息。

```
rom: 28-00000674869d
temp : 30.437 °C
temp : 30.375 °C
```

按下 **Ctrl+C** 结束程序。

### 7.2. python 程序

在终端输入以下命令执行程序：

```
pi@raspberrypi ~/Pioneer600/DS18B20/python $ sudo python ds18b20.py
```

预期结果：终端会输出以下信息。

```
rom: 28-00000674869d

C=29.687 F=85.437
C=29.687 F=85.437
C=29.687 F=85.437
```

按下 **Ctrl+C** 结束程序。



## 8. IRM 示例程序 - 红外感应程序

### 8.1. bcm2835 程序

进入 Linux 终端，在终端执行以下命令。

编译程序：

```
pi@raspberrypi ~/Pioneer600/IRM/bcm2835 $ make
```

执行程序：

```
pi@raspberrypi ~/Pioneer600/IRM/bcm2835 $ sudo ./irm
```

预期结果：按下红外遥控器按键，终端会输出信息接收到按键的键值。

```
irm test start:
Get the key: 0x0c
Get the key: 0x18
Get the key: 0x5e
```

按下 **Ctrl+C** 结束程序。

### 8.2. wiringPi 程序

进入 Linux 终端，在终端执行以下命令。

编译程序：

```
pi@raspberrypi ~/Pioneer600/IRM/wiringPi $ make
```

执行程序：

```
pi@raspberrypi ~/Pioneer600/IRM/wiringPi $ sudo ./irm
```

预期结果：按下红外遥控器按键，终端会输出信息接收到按键的键值。

```
irm test start:
Get the key: 0x0c
Get the key: 0x18
Get the key: 0x5e
```

按下 **Ctrl+C** 结束程序。



### 8.3. python 程序

在终端输入以下命令执行程序：

```
pi@raspberrypi ~/Pioneer600/IRM/python $ sudo python irm.py
```

预期结果：按下红外遥控器按键，终端会输出信息接收到按键的键值。

```
IRM Test Start ...  
Get the key: 0x0c  
Get the key: 0x18  
Get the key: 0x5e
```

按下 **Ctrl+C** 结束程序。





## 9. UART 示例程序 - 发送串口接收到的数据

注意：树莓派的串口默认为终端调试，如要启用这个示例程序，那么您将无法通过串口调试树莓派。如果您之前是使用串口调试树莓派的话，使用这个示例程序之前应该使用其他方式调试树莓派，例如直接插入 HDMI 显示器或者使用 SSH 方式。

### 9.1. wiringPi 程序

进入 Linux 终端，在终端执行以下命令。

编译程序：

```
pi@raspberrypi ~/Pioneer600/UART/wiringPi $ make
```

执行程序：

```
pi@raspberrypi ~/Pioneer600/UART/wiringPi $ sudo ./UART
```

预期结果：把扩展板的 USB TO UART 连接到电脑，设置电脑的串口软件（可使用 PuTTY 等软件监视串口），选择正确的串口号，波特率设置为 115200。电脑通过串口向树莓派发送数据，树莓派在接收到这些数据之后，会把数据通过串口传送回电脑。因此，从电脑端向树莓派发送的数据会回显在电脑端的串口监视软件中。

### 9.2. python 程序

在终端输入以下命令执行程序：

```
pi@raspberrypi ~/Pioneer600/UART/python $ sudo python uart.py
```

预期结果：同上。



## 10. OLED 示例程序

运行 bcm 的 OLED 程序后再运行 wiringpi 或者 python 的 OLED 程序，此时 OLED 会不能显示。只要重启系统之后再次执行示例程序，OLED 就可以显示了。该问题可能和函数库有关，树莓派的 bcm 库比 wiringpi 库和 python 库“更加底层”。我们暂时没有一个好的解决办法。如果您有一个合适的解决方案，欢迎联系我们。联系方式见 [www.waveshare.net](http://www.waveshare.net) 或者 [www.waveshare.com](http://www.waveshare.com)

### 10.1. bcm2835 程序

进入 Linux 终端，在终端执行以下命令。

编译程序：

```
pi@raspberrypi ~/Pioneer600/OLED/bcm2835 $ make
```

执行程序：

```
pi@raspberrypi ~/Pioneer600/OLED/bcm2835 $ sudo ./main
```

预期结果：OLED 会显示微雪电子的 LOGO 两秒。然后显示当前系统时间。

按下 **Ctrl+C** 结束程序。

### 10.2. wiringPi 程序

进入 Linux 终端，在终端执行以下命令。

编译程序：

```
pi@raspberrypi ~/Pioneer600/OLED/wiringPi $ make
```

执行程序：

```
pi@raspberrypi ~/Pioneer600/OLED/wiringPi $ sudo ./main
```

预期结果：OLED 会显示微雪电子的 LOGO 两秒。然后显示当前系统时间。

按下 **Ctrl+C** 结束程序。

### 10.3. python 程序



使用 python 程序控制 OLED 需另外安装 python-imaging 库。把树莓派连接到网络，执行下面语句安装 python-imaging 库：

```
sudo apt-get install python-imaging
```

预期结果：在终端输入以下不同命令执行程序会显示不同的内容。

```
pi@raspberrypi ~/Pioneer600/OLED/python $ sudo python oled.py
```

```
pi@raspberrypi ~/Pioneer600/OLED/python $ sudo python dispchar.py
```

```
pi@raspberrypi ~/Pioneer600/OLED/python $ sudo python image.py
```

```
pi@raspberrypi ~/Pioneer600/OLED/python $ sudo python animate.py
```

```
pi@raspberrypi ~/Pioneer600/OLED/python $ sudo python waveshare.py
```

更加详细教程请看[树莓派系列教程](#)。

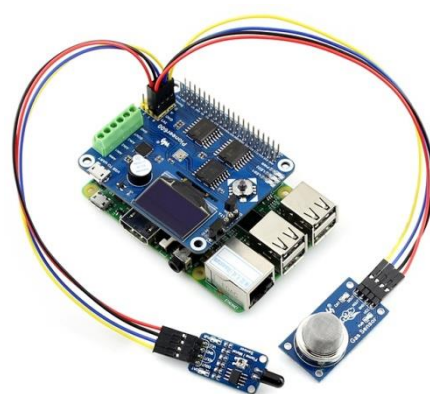


## 11. Pioneer600 连接传感器套件（需另外选购）

Pioneer600 提供了 4 个外部传感器接口，因此您除了使用板载的传感器之外，还可以为它附加其他的传感器功能。

在使用外部传感器之前，您需要：

- 1) 把传感器套件连接到扩展板，同时扩展板需要插入到树莓派才能使用。
- 2) 提供 AD/DA 功能的 PCF8591 芯片使用了 I<sup>2</sup>C 接口，因此请确保安装好函数库并且启动了 I<sup>2</sup>C 内核驱动。
- 3) 您还需要下载产品的示例程序。这些示例程序都需要解压并放在 /home/pi 目录下（您可以通过 U 盘把它们复制到树莓派中）。



## 12. Color Sensor 示例程序 - 颜色传感器演示

按照下表连接传感器和 Pioneer600 的管脚：

Color Sensor 管脚	Pioneer600 管脚
LED	3.3V
OUT	P0
S3	D3( P26)
S2	D2(P23)
S1	D1(P22)
S0	D0(P21)
GND	GND
VCC	3.3V

进入/Pioneer600/Sensor/Color\_Sensor 文件夹中，终端执行：

```
sudo ./Color_Sensor
```

预期结果：程序会对芯片的白平衡进行调整，时间大概为 2s，调整结束以后即可把三原色的频率经过终端输出，对照 RGB 颜色对照表，即可知道所测得颜色。

按下 **Ctrl+C** 结束程序。

## 13. Flame Sensor 示例程序 - 火焰传感器演示





按照下表连接传感器和 Pioneer600 的管脚：

Flame Sensor 管脚	Pioneer600 管脚
DOUT	D3
AOUT	A3
GND	GND
VCC	3.3V

进入/Pioneer600/Sensor/Flame\_Sensor 文件夹中，终端执行：

```
sudo ./General_Sensor
```

预期结果：传感器靠近火焰时，模块上的信号指示灯点亮。传感器远离火焰时，模块上的信号指示灯熄灭。随着传感器与火焰距离的改变，终端输出的数据也会发生改变。

按下 **Ctrl+C** 结束程序。

注意：该传感器主要用于感知火焰，但其自身并不防火。因此使用时请与火焰保持一定距离，以免烧坏传感器。

14. Hall Sensor 示例程序 - 霍尔传感器演示

按照下表连接传感器和 Pioneer600 的管脚：

Hall Sensor 管脚	Pioneer600 管脚
DOUT	D3
AOUT	A3
GND	GND
VCC	3.3V

进入/Pioneer600/Sensor/ Hall Sensor 文件夹中，终端执行：

```
sudo ./General_Sensor
```

预期结果：传感器靠近磁铁时，模块上的信号指示灯点亮。传感器远离磁铁时，模块上的信号指示灯熄灭。随着传感器与磁铁接触与分离，终端输出的数据会发生相应改变。

按下 **Ctrl+C** 结束程序。

15. Infrared Reflective Sensor 示例程序 - 一体式红外发送接收管演示

按照下表连接传感器和 Pioneer600 的管脚：



Infrared Reflective Sensor 管脚	Pioneer600 管脚
DOUT	D3
AOUT	A3
GND	GND
VCC	3.3V

进入/Pioneer600/Sensor/ Infrared\_Reflective\_Sensor 文件夹中，终端执行：

```
sudo ./General_Sensor
```

预期结果：传感器靠近障碍物时，模块上的信号指示灯点亮。传感器远离障碍物时，模块上的信号指示灯熄灭。随着传感器与障碍物距离的变化，终端输出的数据也会发生改变。

按下 **Ctrl+C** 结束程序。

## 16. Laser Sensor 示例程序 - 激光传感器演示

按照下表连接传感器和 Pioneer600 的管脚：

Laser Sensor 管脚	Pioneer600 管脚
DOUT	D3
GND	GND
VCC	3.3V

进入/Pioneer600/Sensor/Laser\_Sensor 文件夹中，终端执行：

```
sudo ./General_Sensor
```

预期结果：把障碍物置于激光传感器上方，此时模块上的信号指示灯会被点亮，把障碍物远离激光传感器上方，此时模块上的信号指示灯熄灭。由此可知激光传感器是否探测到障碍物。

按下 **Ctrl+C** 结束程序。

## 17. Moisture Sensor 示例程序 - 土壤湿度传感器演示

按照下表连接传感器和 Pioneer600 的管脚：

Moisture Sensor 管脚	Pioneer600 管脚
DOUT	D3
AOUT	A3
GND	GND



VCC	3.3V
-----	------

进入/Pioneer600/Sensor/Moisture\_Sensorr 文件夹中，终端执行：

```
sudo ./General_Sensor
```

预期结果：把传感器插入土壤中，然后逐渐往土壤中加水，终端输出数据变化。

按下 **Ctrl+C** 结束程序。

### 18. Rotation Sensor 示例程序 - 旋转编码器演示

按照下表连接传感器和 Pioneer600 的管脚：

Rotation Sensor 管脚	Pioneer600 管脚
SIA	D0
SIB	D1
SW	D2
GND	GND
VCC	3.3V

进入/Pioneer600/Sensor/Rotation\_Sensor 文件夹中，终端执行：

```
sudo ./Rotation_Sensor
```

预期结果：分别顺时针旋转，逆时针旋转和按下编码器，端口分别输出数据：

Turn right!

Turn left!

Turn down!

按下 **Ctrl+C** 结束程序。

### 19. Sound Sensor 示例程序 - 声音探测器演示

按照下表连接传感器和 Pioneer600 的管脚：

Sound Sensor 管脚	Pioneer600 管脚
DOUT	D3
AOUT	A3
GND	GND
VCC	3.3V



进入/Pioneer600/Sensor/Sound\_Sensor 文件夹中，终端执行：

```
sudo ./General_Sensor
```

预期结果：当模块的咪头靠近发声源时，模块上的信号指示灯点亮。当模块的咪头远离发声源时，模块上的信号指示灯熄灭。随着传感器与发声源距离的变化，终端输出数据有相应的变化。

按下 **Ctrl+C** 结束程序。

20. Temperature-Humidity Sensor 示例程序 - 温湿度传感器演示

按照下表连接传感器和 Pioneer600 的管脚：

Temperature-Humidity Sensor 管脚	Pioneer600 管脚
DOUT	D3
GND	GND
VCC	3.3V

进入/Pioneer600/Sensor/Temperature-Humidity\_Sensorr 文件夹中，终端执行：

```
sudo ./DHT11
```

预期结果：终端输出温度和湿度。例如：

```
Humidity=33
Temperature=28
```

按下 **Ctrl+C** 结束程序。

21. MQ-5 Gas Sensor 示例程序 - MQ-5 气体传感器演示

按照下表连接传感器和 Pioneer600 的管脚：

MQ-5 Gas Sensor 管脚	Pioneer600 管脚
DOUT	D3
AOUT	A3
GND	GND
VCC	3.3V

进入/Pioneer600/Sensor/MQ-5\_Gas\_Sensor 文件夹中，终端执行：

```
sudo ./General_Sensor
```





预期结果：首先让传感器先预热一分钟。把传感器放入含有敏感气体（该气体传感器对液化气，天然气和煤气敏感）的装置中，模块上的信号指示灯点亮。把传感器从敏感气体装置中取出，模块上的信号指示灯熄灭。由此可判断敏感气体的浓度是否超标。

按下 **Ctrl+C** 结束程序。

## 22. Tilt Sensor 示例程序 - 倾斜传感器演示

按照下表连接传感器和 Pioneer600 的管脚：

Tilt Sensor 管脚	Pioneer600 管脚
DOUT	D3
GND	GND
VCC	3.3V

进入/Pioneer600/Sensor/Tilt\_Sensor 文件夹中，终端执行：

```
sudo ./General_Sensor
```

预期结果：晃动模块或使模块倾斜时，模块上的信号指示灯点亮。模块平行放置时，模块上的信号指示灯熄灭。由此可判断模块的状态是否发生晃动或倾斜。

按下 **Ctrl+C** 结束程序。

## 23. UV Sensor 示例程序 - 紫外线传感器演示

按照下表连接传感器和 Pioneer600 的管脚：

UV Sensor 管脚	Pioneer600 管脚
AOUT	A3
GND	GND
VCC	3.3V

进入/Pioneer600/Sensor/UV\_Sensor 文件夹中，终端执行：

```
sudo ./General_Sensor
```

预期结果：让传感器靠近阳光，终端输出的数据会发生改变。

按下 **Ctrl+C** 结束程序。



## 24. Liquid Level Sensor 示例程序 - 液位传感器演示

按照下表连接传感器和 Pioneer600 的管脚：

Liquid Level Sensor 管脚	Pioneer600 管脚
AOUT	A3
GND	GND
VCC	3.3V

进入/Pioneer600/Sensor/ Liquid\_Level\_Sensor 文件夹中，终端执行：

```
sudo ./General_Sensor
```

预期结果：把传感器插入一定深度的水中，终端输出的数据会发生变化。

按下 **Ctrl+C** 结束程序。