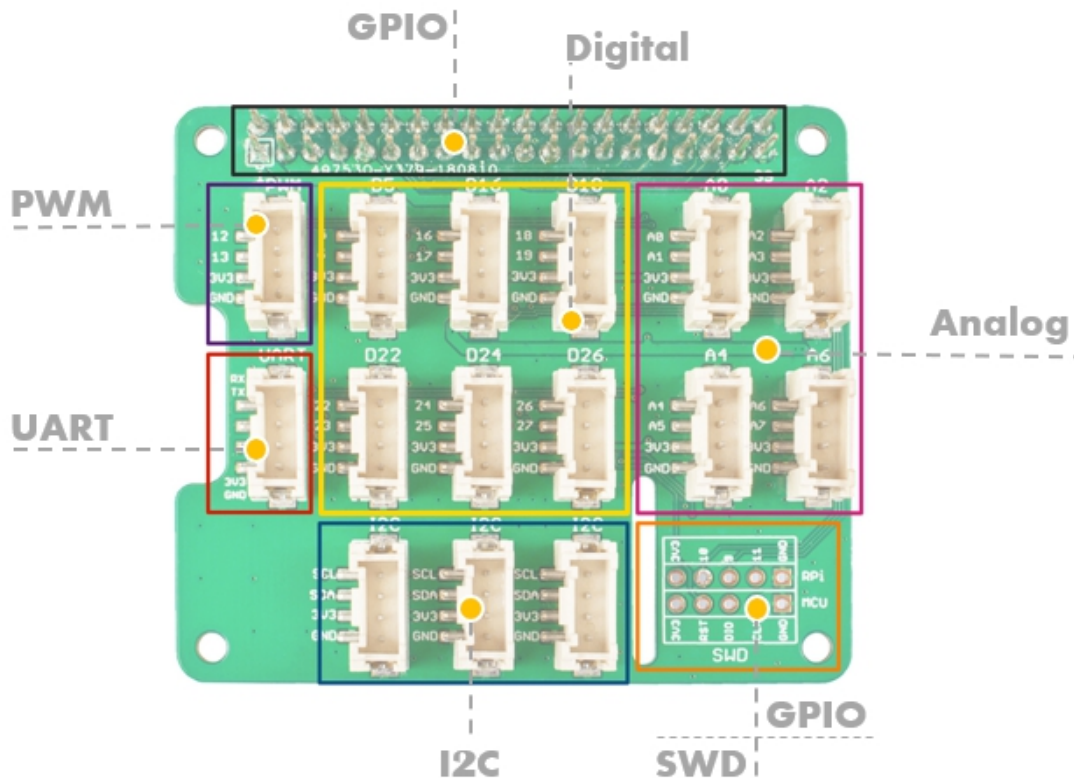


## 特征

- 支持Raspberry 2/3B/3B+/Zero
- 内置MCU
- 12位ADC
- 多个Grove接口

## 硬件概述



**GPIO:** 与树莓派的Pin输出一样。

**PWM:** 树莓派PWM部分在GPIO/BCM pin12(PWM0)和GPIO/BCM pin13(PWM1)的引脚上，它们是树莓派硬件PWM引脚，另外，你通过软件设计使用任意的GPIOPin来模拟PWM。

!!!Note

- 除Grove端口之外的所有丝印层引脚编号都是BCM引脚编号请参考[这里] (<https://www.raspberrypi.org/forums/viewtopic.php?p=726435>)
- 软件PWM没有硬件PWM精准而且在频率很高的时候容易出现问題
- GPIO/BCM Pin18也被标记为PWM0，实际上GPIO/BCM 12和GPIO/BCM 18共享一个定时器，所以他们不能设置不同的频率
- audio接口的输出也是使用PWM0和PWM1，所以你不能在audio输出音频时使用那两个PWM

**UART:** Grove Base Hat for Raspberry Pi的UART部分与GPIO14(UART0 TX)和GPIO15(UART0 RX)相连。由于UART默认被打开，所以我们使用UART去访问树莓派的内核启动信息是非常方便的。同时树莓派也可以通过UART去与Arduino, bootloaded ATmega, ESP8266等等设备通信。

**Digital:** Grove Base Hat for Raspberry Pi上有六个Digital接口，通常Grove排线中黄色那根线是信号线，因此我们把digital Grove命名为D5/D16/D18/D22/D24/D26。

**Analog:** 由于树莓派是没有Analog接口的，所以一般情况下树莓派是不能驱动模拟传感器的。不过我们可以通过Grove Base Hat for Raspberry Pi中内置的MCU来进行模拟输入输出对树莓派赋予使用模拟传感器的功能。更

让我们兴奋的是，Grove Base Hat for Raspberry Pi上面有四个模拟Grove接口可以使用。模拟传感器可以输入电压经过Grove Base Hat for Raspberry Pi，然后通过Grove Base Hat for Raspberry Pi内置的12位AD转化变成数字信号，通过I2C将数字信号传给树莓派。

I2C: Base Hat for Raspberry Pi 存在三个I2C接口，都是跟树莓派直接连接的。你可以认为Base Hat for Raspberry Pi是一个I2C分线器。目前Seeed有很多的新模块都是I2C接口。所以三个I2C是很有必要的。

SWD: 我们可以使用SWD部分去更新Base Hat for Raspberry Pi的固件。另外，你可以在SWD部分看到三个GPIO引脚。这个三个GPIO引脚没有被Grove部分使用，所以你也可以把它们当成GPIO来使用。

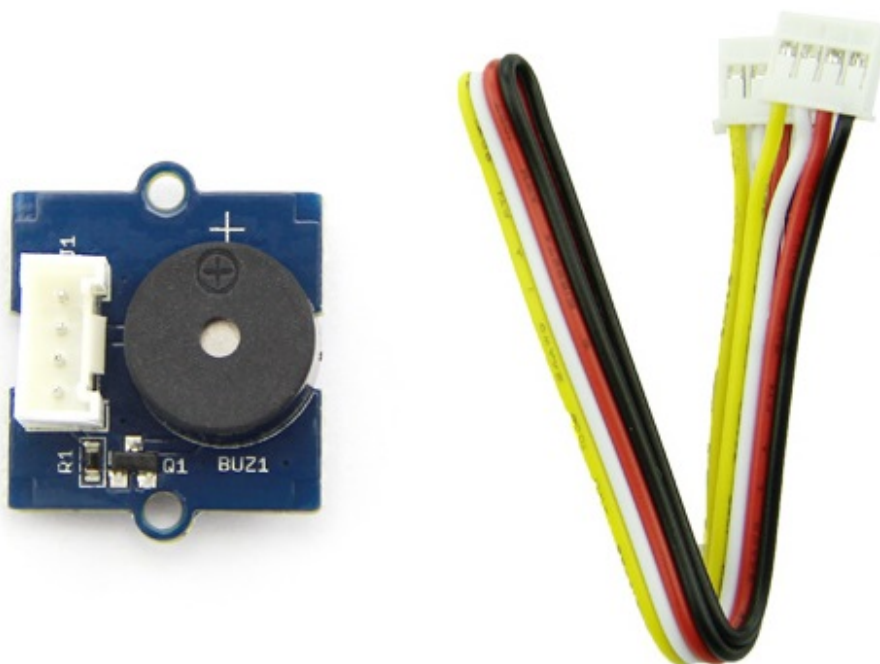
## Grove模块

### Grove - Buzzer

Grove - Buzzer使用压电蜂鸣器作为主要元器件，当digital输出是高电平时，压电蜂鸣器发生鸣叫。反之亦反。另外，当使用PWM控制它时，我们可以产生不同的频率PWM使其声调发生变化。

#### !!!Note

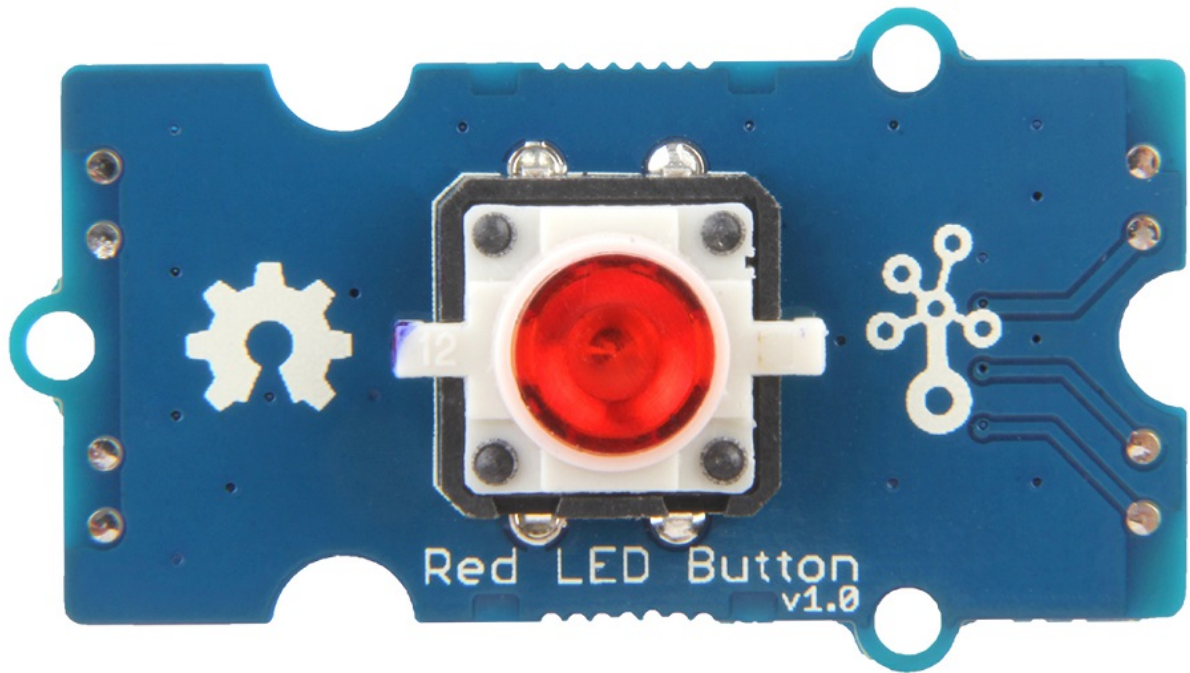
通常人耳只能分辨20Hz到20KHz声音频率



### Grove - LED Button

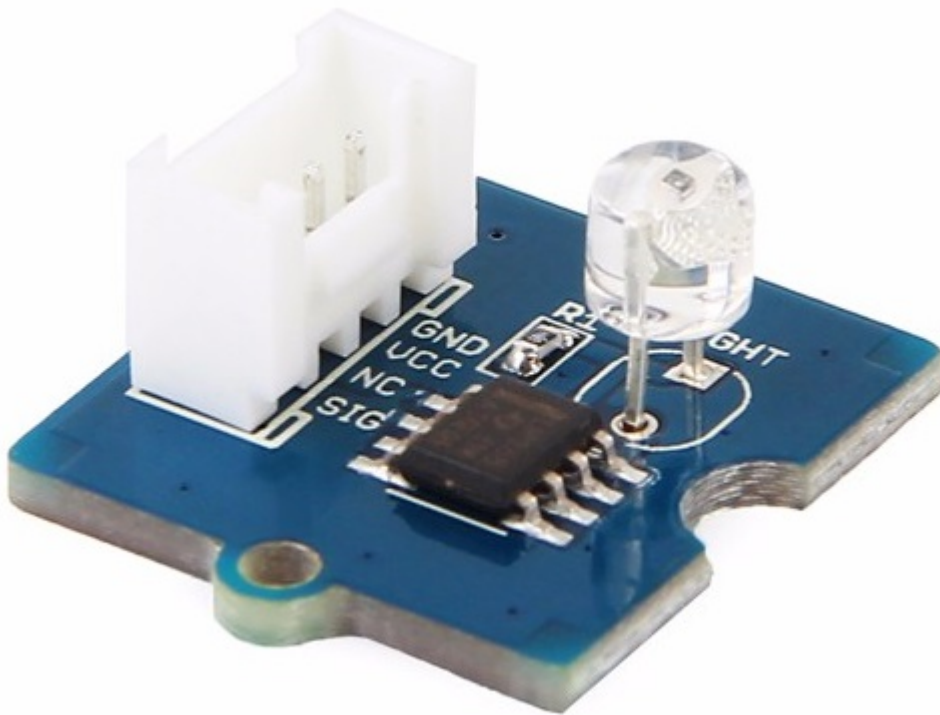
Grove - LED Button是非常稳定和坚固的，其至少可以使用100000次。通过内置的LED，你可以制作一些非常有趣的项目，比如你可以制作一个监测按键状态的设备。





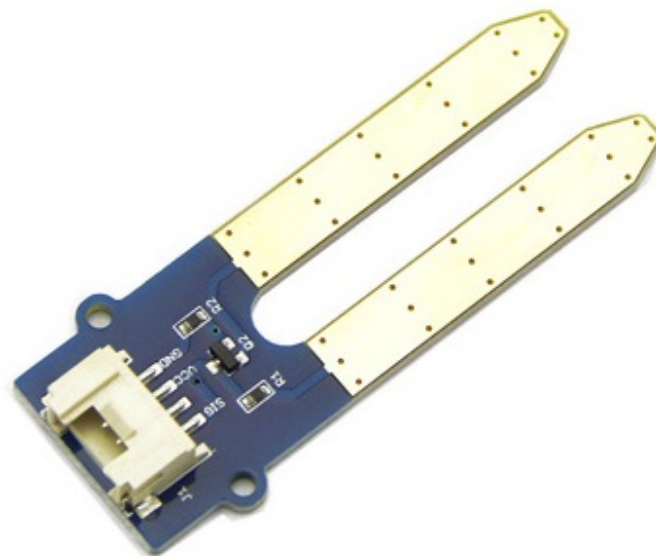
### Grove - Light sensor

Grove - Light sensor内部集成了一个光敏电阻去探测光的强度。当光照强度变大时，光敏电阻的阻值也随之变大，反之亦反。其内部集成双通道运放LM358去放大光敏电阻的阻值，所以输出信号是模拟值，光照越强对应的值就越大。



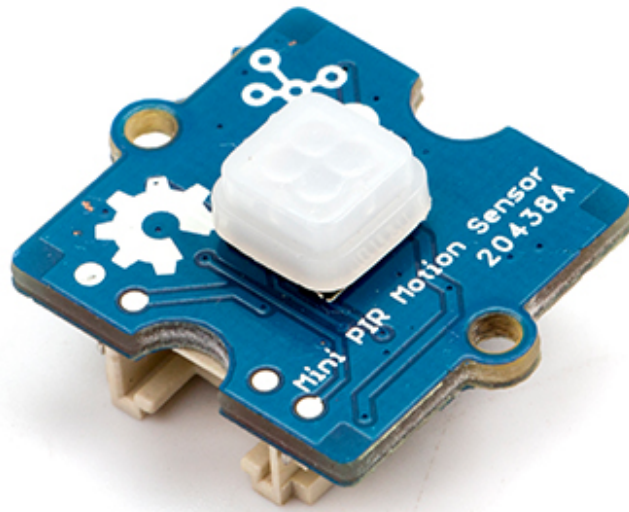
### Grove - Moisture Sensor

Grove - Moisture Sensor可以探测土壤的湿度或者创感器周围是否有水，它使用起来非常简单仅仅只需要你将它插到土壤中并采集它输出的值。典型应用是可以让你知道什么时候你该给你的植物浇水。



### Grove - mini PIR motion sensor

Grove - mini PIR motion sensor可以在一定范围内探测人类是否在移动。当人类在Grove - mini PIR motion sensor探测范围移动时，Grove - mini PIR motion sensor将会通过SIG引脚输出高电平。



## Grove - Servo

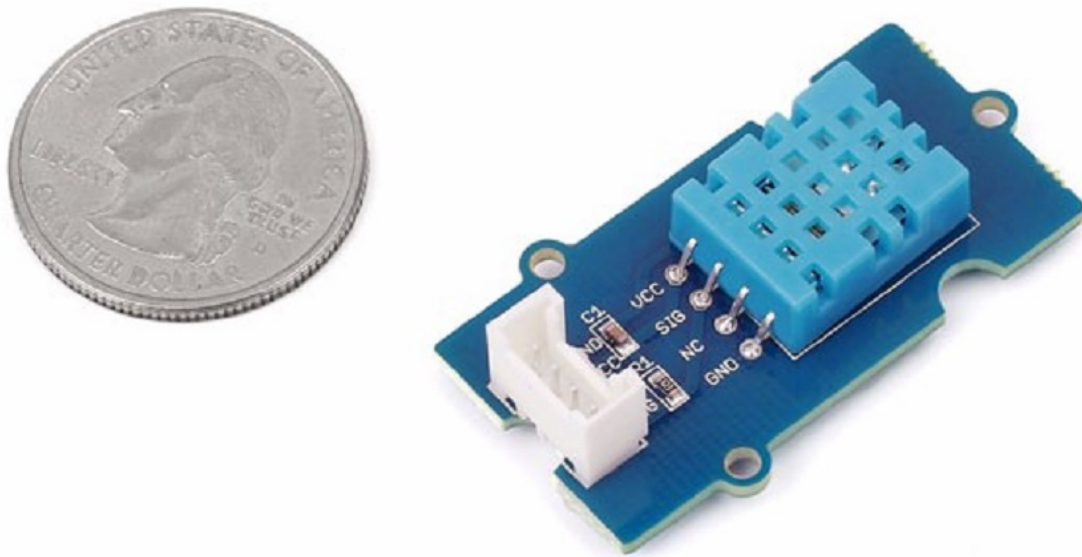
Grove - Servo是一个带有传动和反馈系统的直流电机。它常常被用来控制机器人动作。我们将三线的伺服电机重新定义为Grove的标准接口。你可以将它直接插入到Grove接口上面，不需要其他的转接。



### Grove - temperature & humidity sensor

Grove - temperature & humidity sensor提供预校准输出。独特的电容式传感器元件测量相对湿度，温度由负温度系数（NTC）热敏电阻测量。它具有出色的可靠性和长期稳定性。请注意，此传感器不适用于0度以下的温度。





## Grove - Relay

Grove - Relay模块是一个数字常开开关，可以通过5V弱电电路去控制220V的强电电路。为了方便的看到效果，Grove - Relay模块上面有一个LED灯，当继电器关闭时LED将被点亮。



## 快速入门

### 材料准备

- micro USB线

- 树莓派3B
- SD卡
- Grove Base Kit for Raspberry Pi

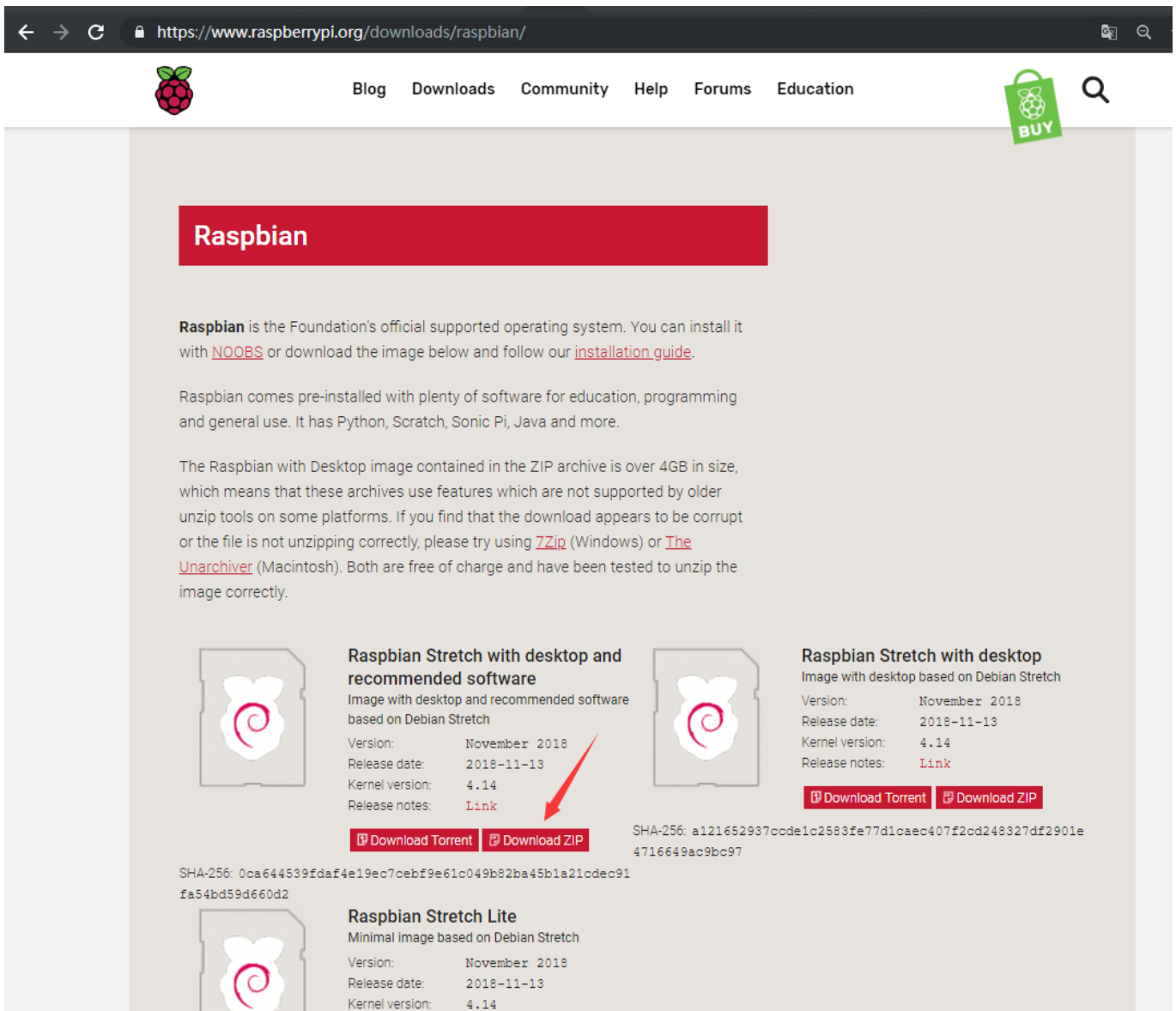
## 基础指南

### Arduino IDE 基本设置

### 怎样刷写树莓派固件

#### 第一步. 树莓派固件下载

通过访问树莓派的官方网站去下载固件，选择[with desktop and recommended software](#)版本



The screenshot shows the official Raspbian download page on the Raspberry Pi website. The page is titled "Raspbian" and provides information about the operating system. It lists three download options:

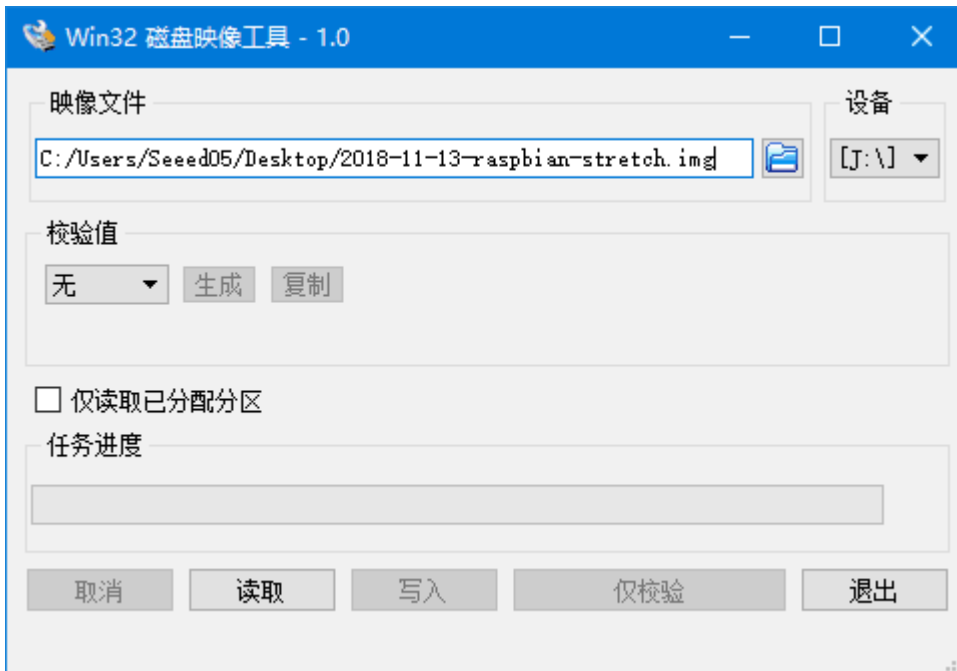
- Raspbian Stretch with desktop and recommended software**: Image with desktop and recommended software based on Debian Stretch. Version: November 2018, Release date: 2018-11-13, Kernel version: 4.14, Release notes: [Link](#). Includes "Download Torrent" and "Download ZIP" buttons.
- Raspbian Stretch with desktop**: Image with desktop based on Debian Stretch. Version: November 2018, Release date: 2018-11-13, Kernel version: 4.14, Release notes: [Link](#). Includes "Download Torrent" and "Download ZIP" buttons.
- Raspbian Stretch Lite**: Minimal image based on Debian Stretch. Version: November 2018, Release date: 2018-11-13, Kernel version: 4.14.

A red arrow points to the "Download ZIP" button for the first option. The page also includes a search bar, a "BUY" button, and a navigation menu with links for Blog, Downloads, Community, Help, Forums, and Education.

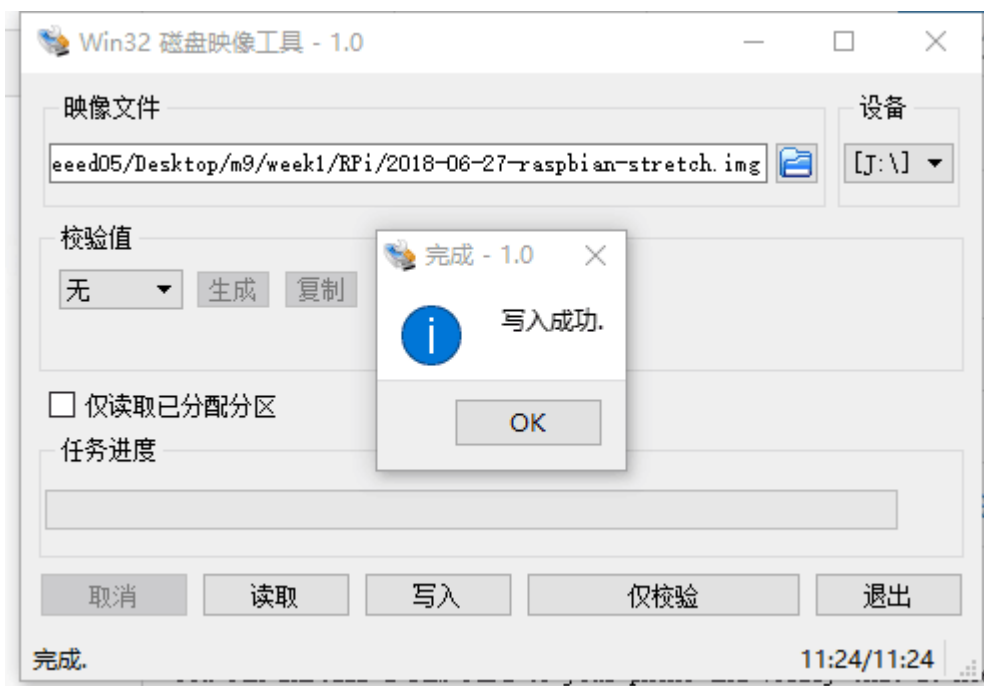
#### 第二步. Win32磁盘映像工具

- 在Sourceforge中下载[Win32磁盘映像工具](#)，并安装
- 将你的内存卡插入你的读卡器，然后将读卡器插入电脑
- 在你的桌面或者菜单中打开Win32磁盘映像工具这个软件

- 在驱动盒子中，选择对应SD卡的驱动
- 将你下载好的镜像放入镜像文件中
- 点击"写入"并等待写入完成



- 完成



- 退出软件并拔出SD卡

## 基本配置

### 无线连接和SSH

第一步. 在/boot文件夹中创建一个名为wpa\_supplicant.conf的文件，复制下面的代码到这个文件中

```
country=CN
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

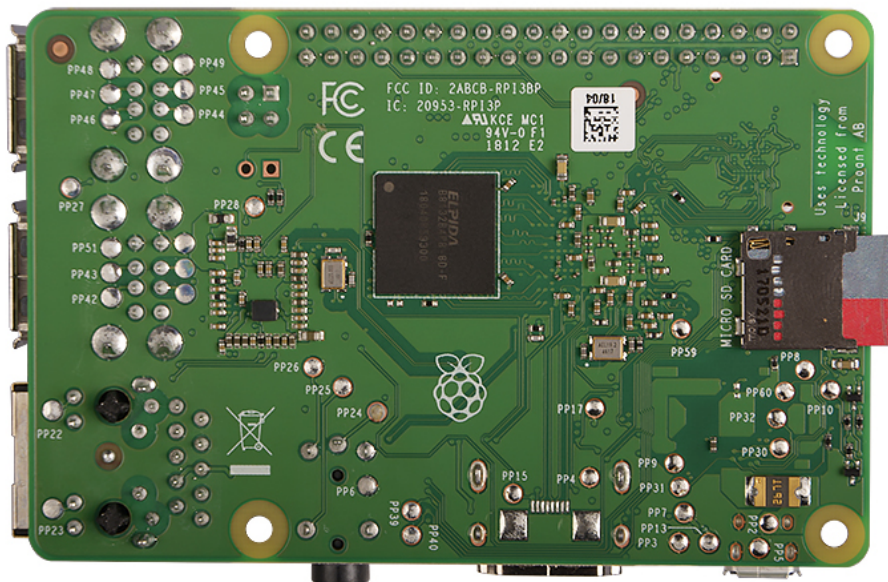
network={
ssid="WiFi-name"
psk="WiFi-password"
key_mgmt=WPA-PSK
priority=1
}
```

### !!!Note

Wi-Fi的名字和密码需要与你当前的电脑所连接的Wi-Fi的名字和密码一样

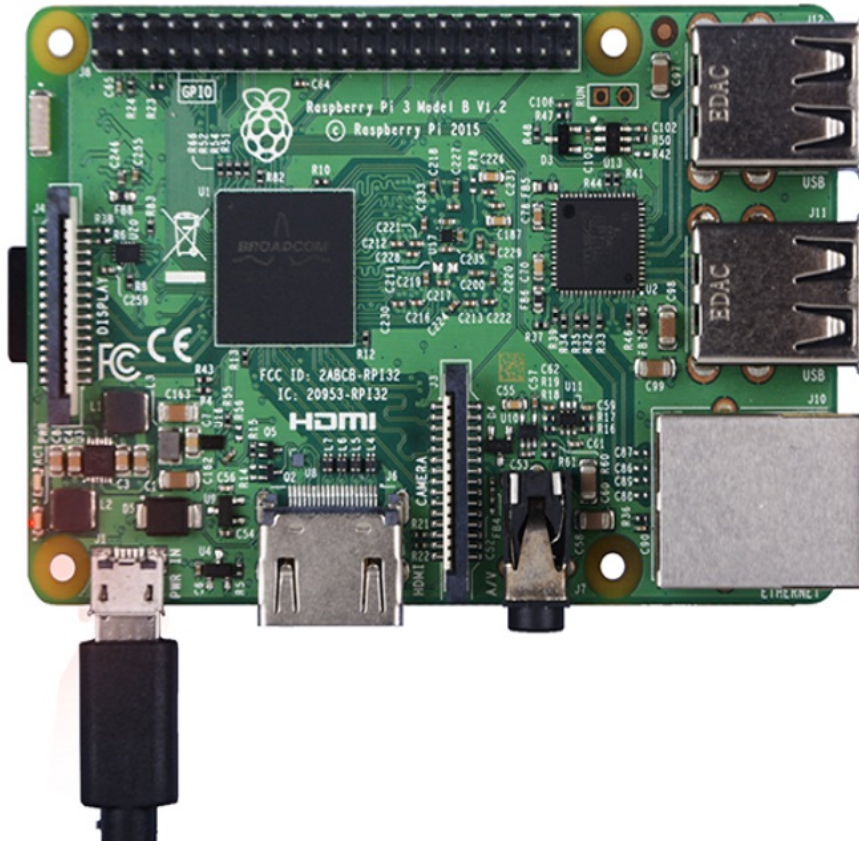
第二步. 在/boot文件夹下面创建一个名为"ssh"的空文件

第三步. 将SD卡插入树莓派的SD卡的卡槽中

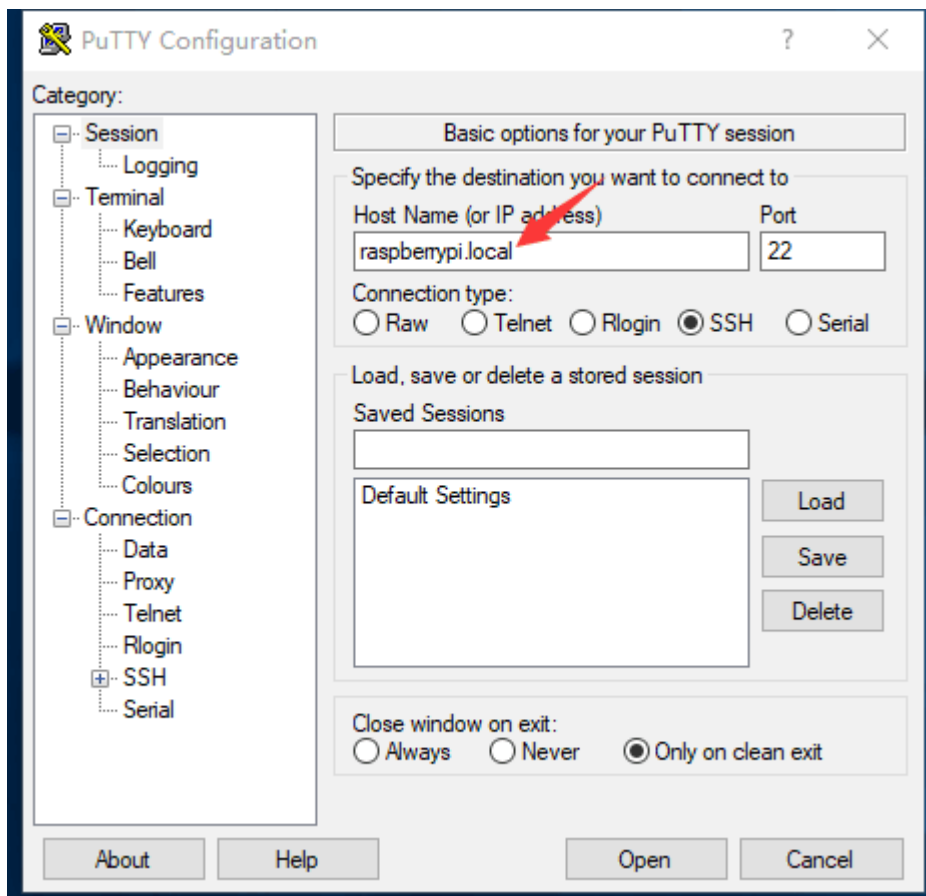


第四步. 通过USB给树莓派供电





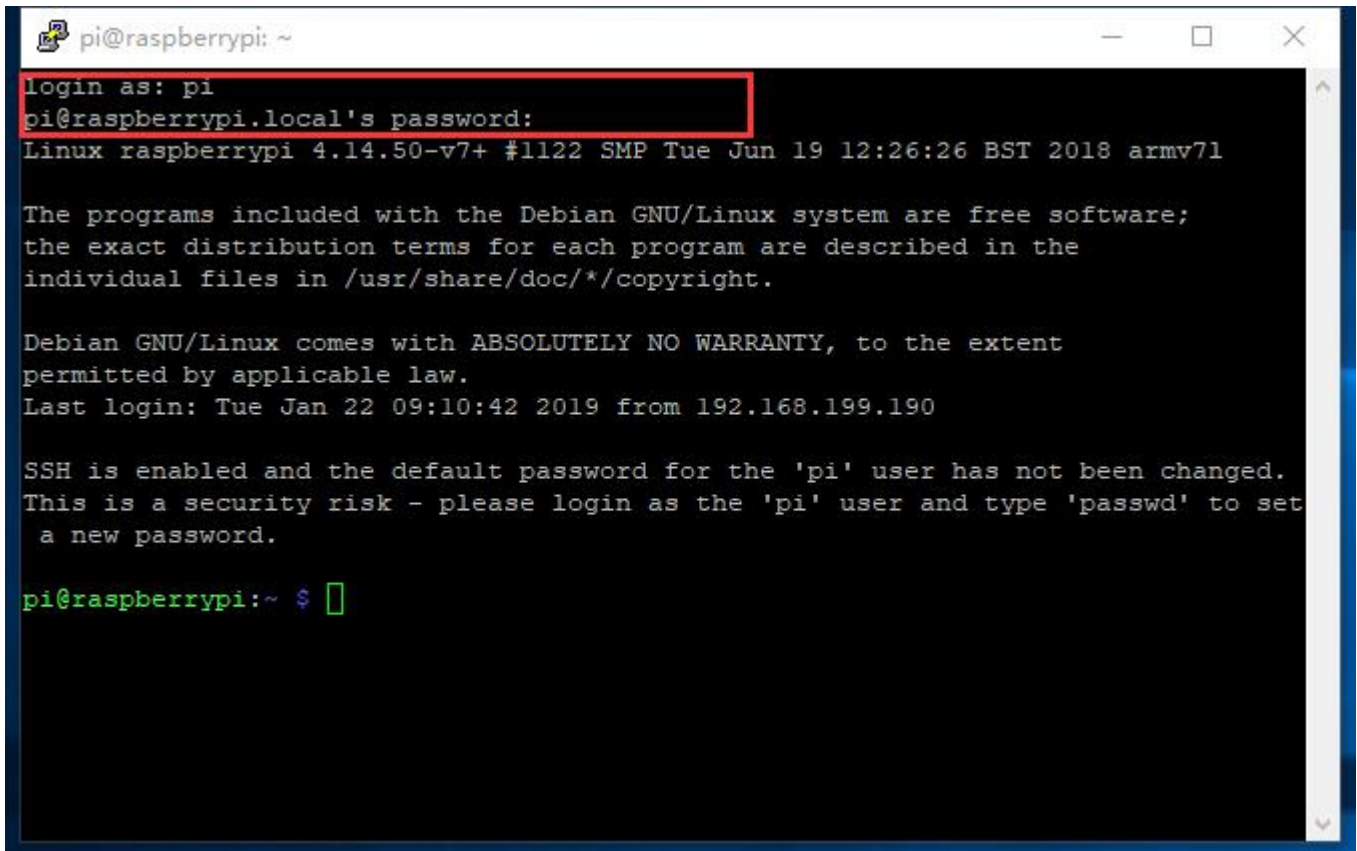
第五步. 通过putty将树莓派和PC进行远程连接



登入树莓派系统

默认用户名 : pi

默认密码 : raspberry



```
pi@raspberrypi: ~
login as: pi
pi@raspberrypi.local's password:
Linux raspberrypi 4.14.50-v7+ #1122 SMP Tue Jun 19 12:26:26 BST 2018 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jan 22 09:10:42 2019 from 192.168.199.190

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

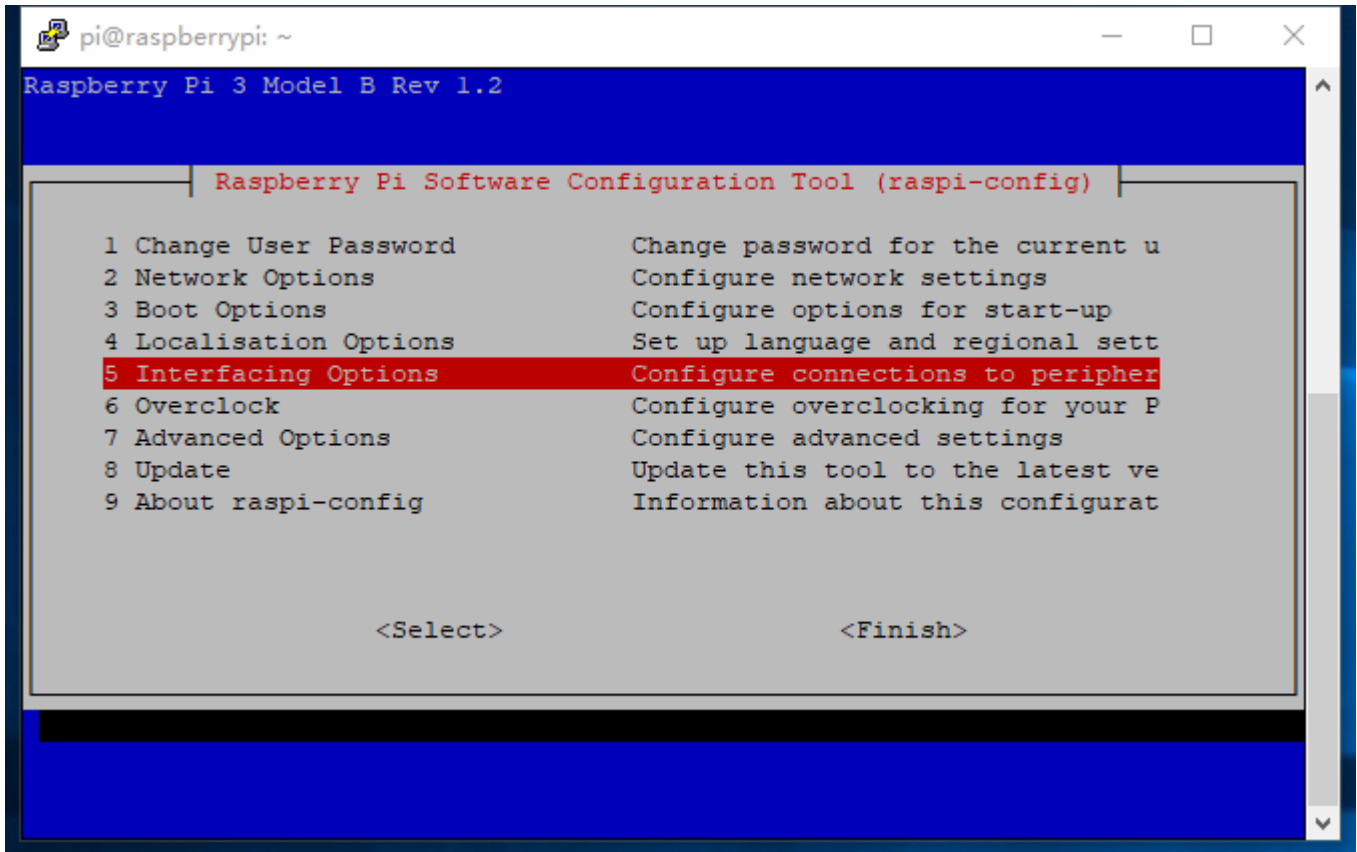
pi@raspberrypi:~ $
```

## VNC配置

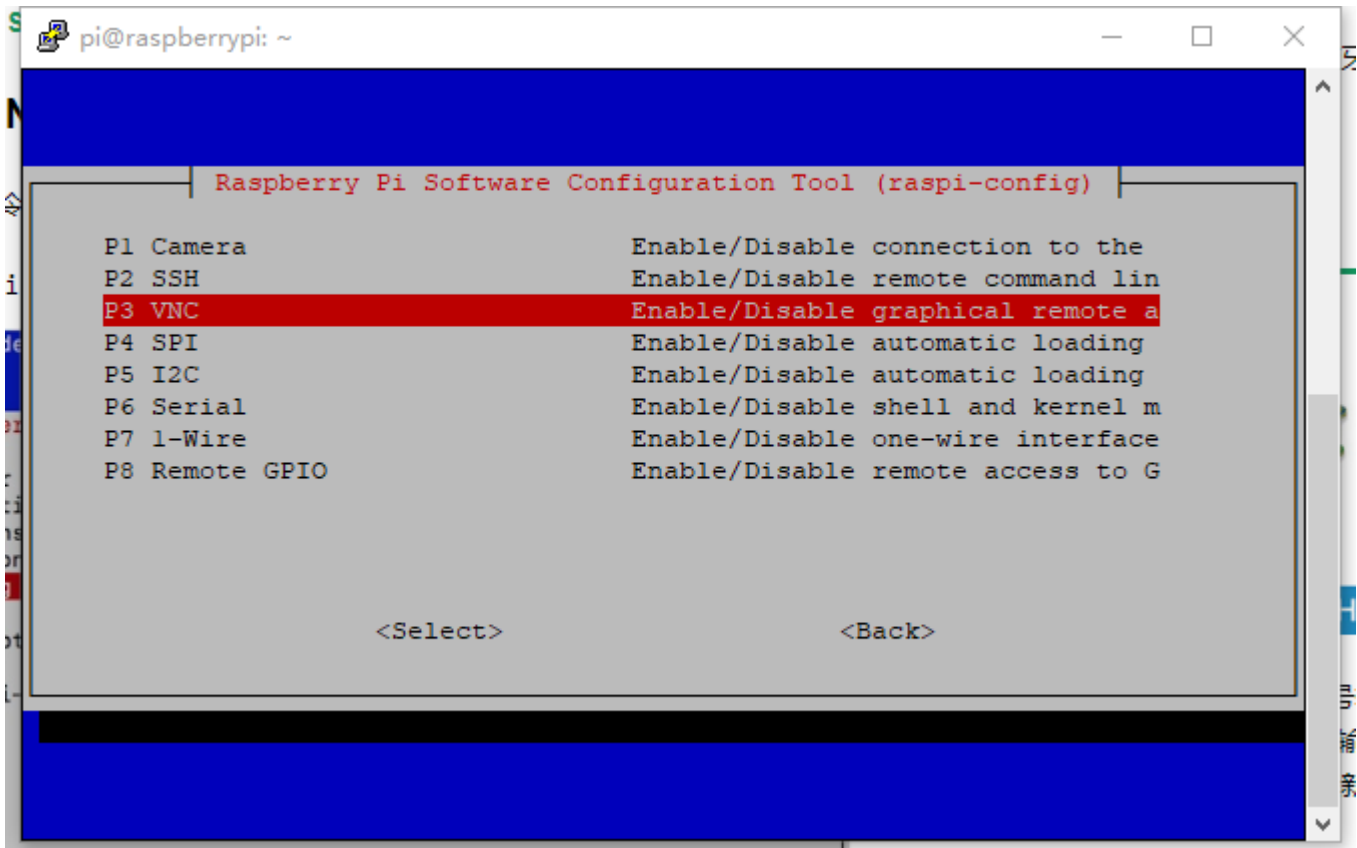
第一步. 通过输入下面的命名行打开raspi-config的界面

```
sudo raspi-config
```

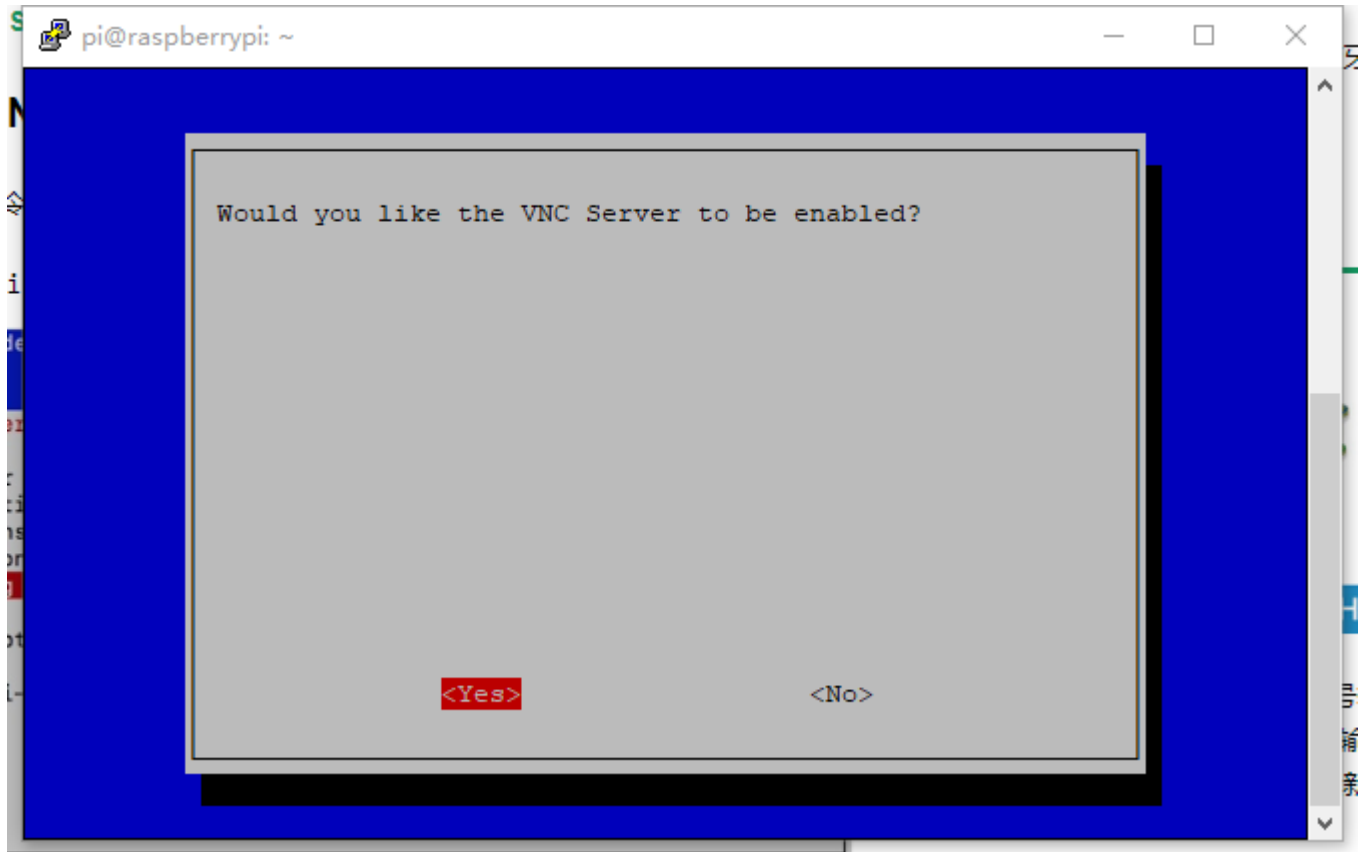
选择"5 Interfacing Options"并按"Enter"键



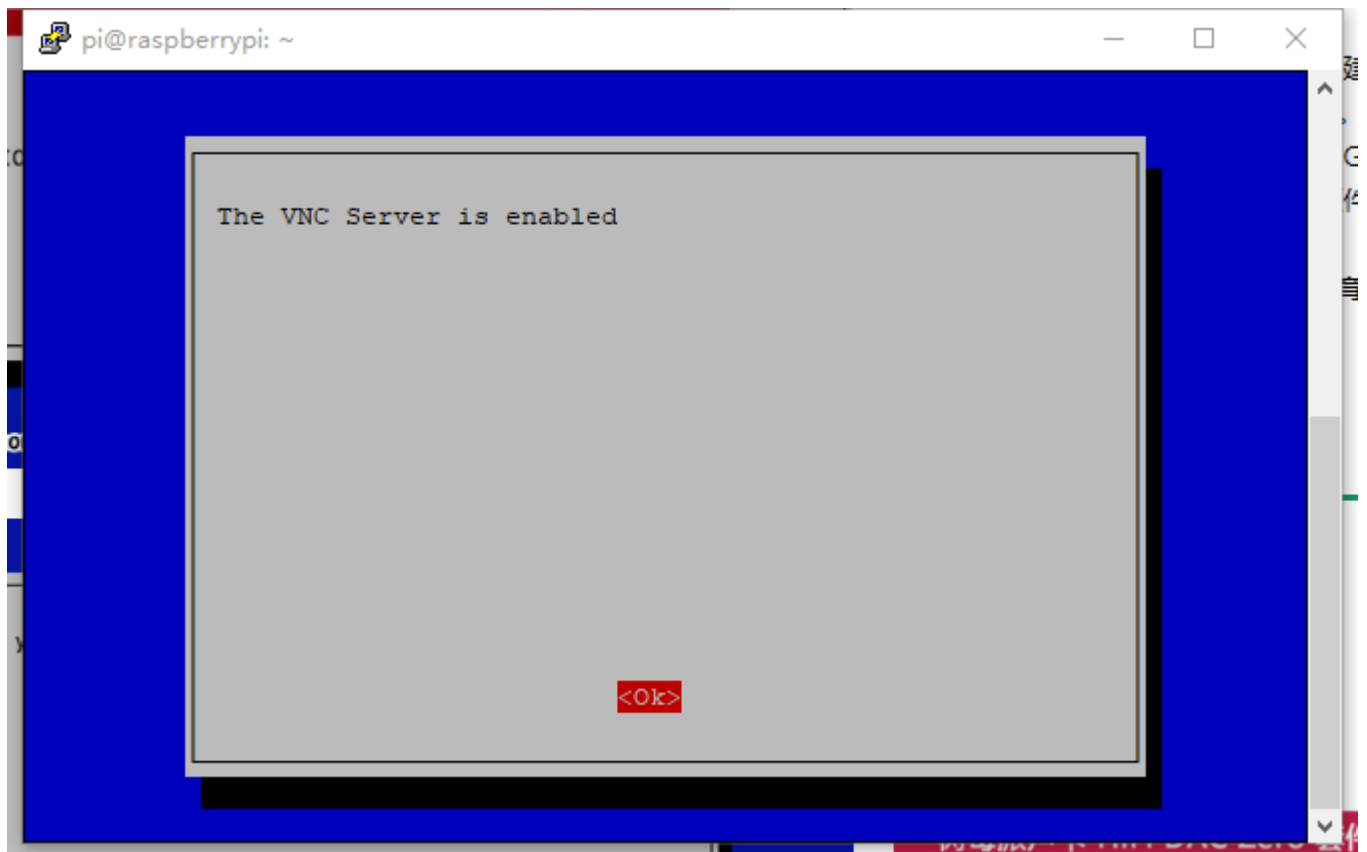
选择"P3 VNC"并按"Enter"键



选择"Yes"并按"Enter"键去打开VNC服务



选择"Ok"



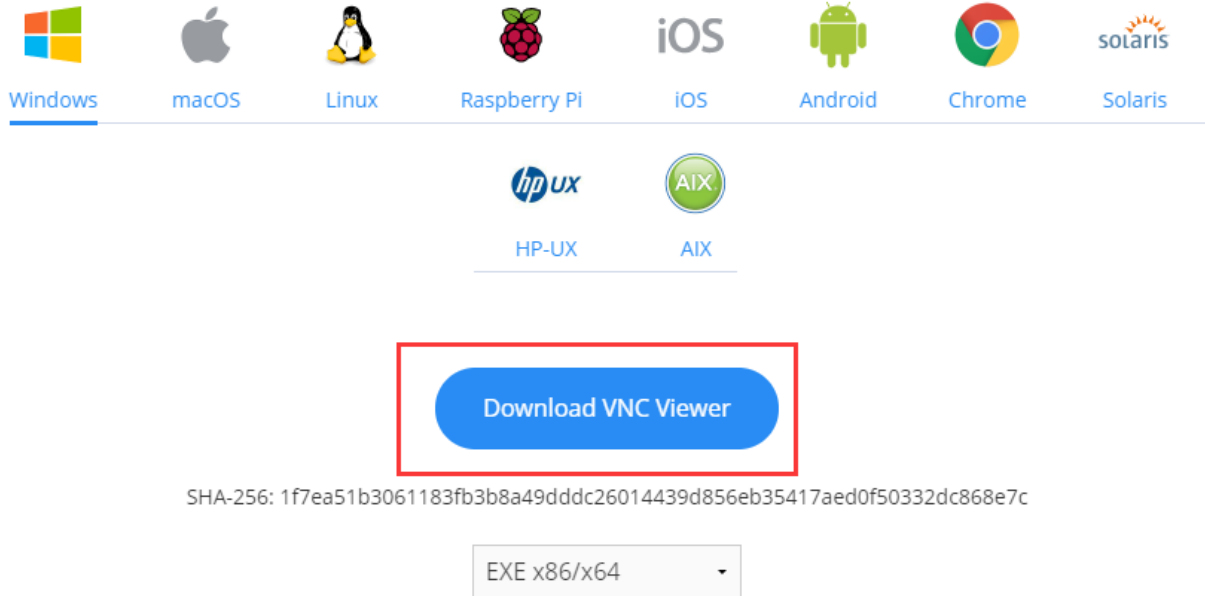
第二步. 安装VNC Viewer

下载 [VNC Viewe](#)



## Download VNC Viewer to the device to control from

Make sure you've downloaded [VNC Connect](#) to the computer you want to control.



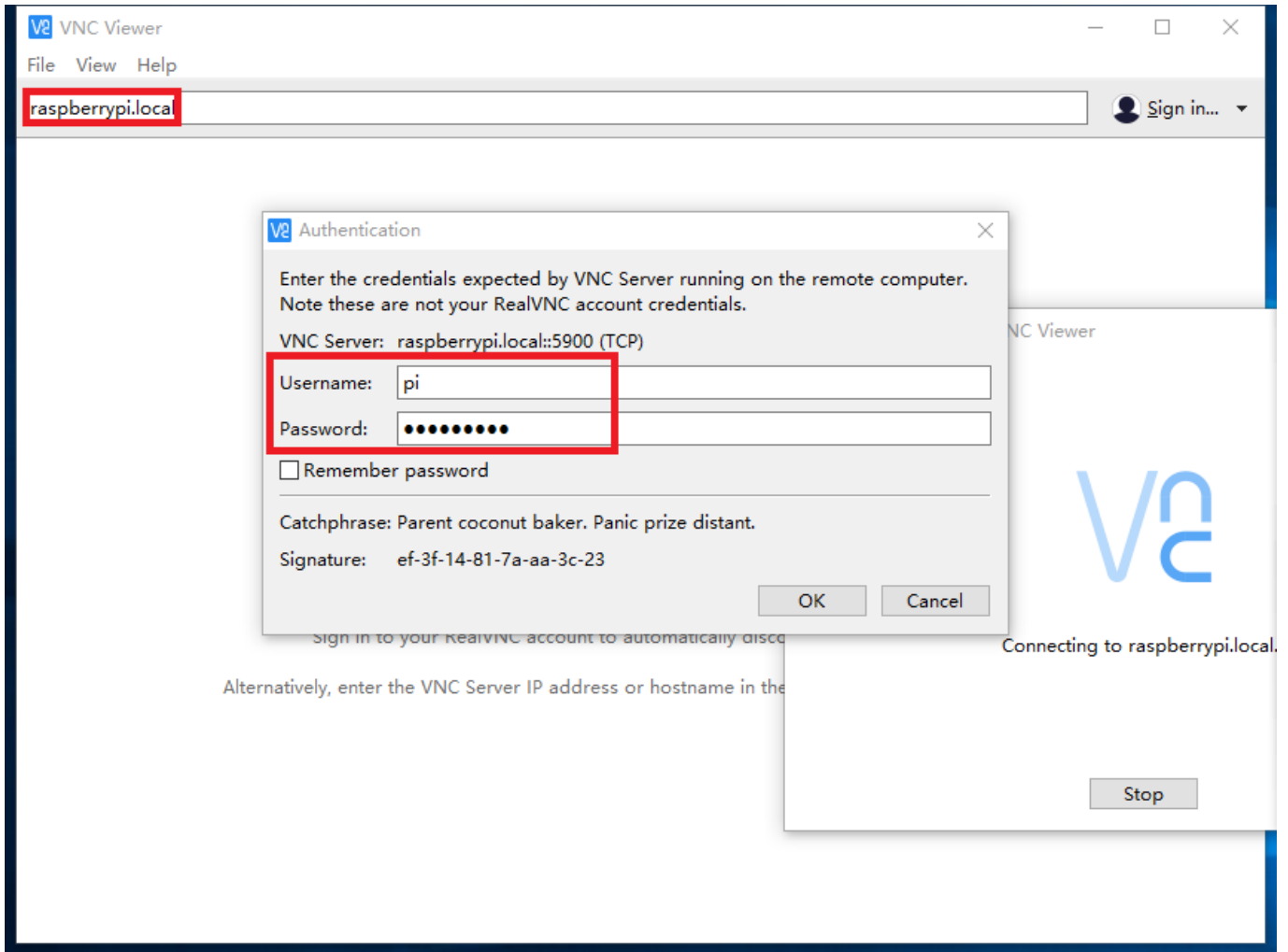
The screenshot shows the VNC Viewer download interface. At the top, there are icons for various operating systems: Windows, macOS, Linux, Raspberry Pi, iOS, Android, Chrome, and Solaris. Below these are icons for HP-UX and AIX. A blue button labeled 'Download VNC Viewer' is highlighted with a red rectangle. Below the button, the SHA-256 hash is displayed: 1f7ea51b3061183fb3b8a49dddc26014439d856eb35417aed0f50332dc868e7c. At the bottom, there is a dropdown menu showing 'EXE x86/x64'.

打开VNC Viewer并输入树莓派的地址，你可以在putty上面输入命令“ifconfig”去查看树莓派终端的IP的地址。

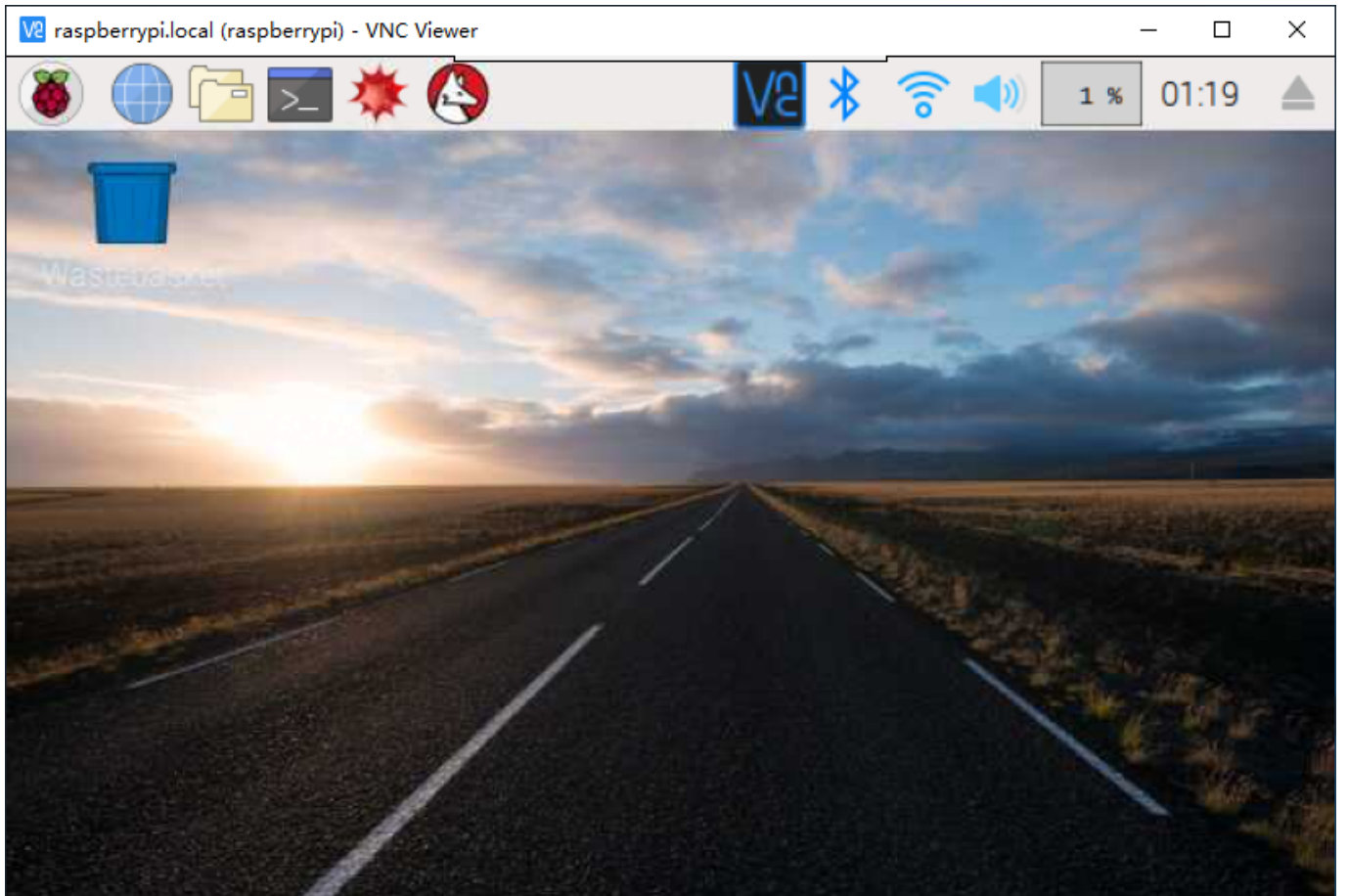
### !!!Note

如果用户使用`raspberrypi.local`登入你的树莓派的话，我们必须必须确保你的局域网内只有一块树莓派

输入默认的用户名和密码，现在你就可以通过VNC访问树莓派的桌面了！



成功!

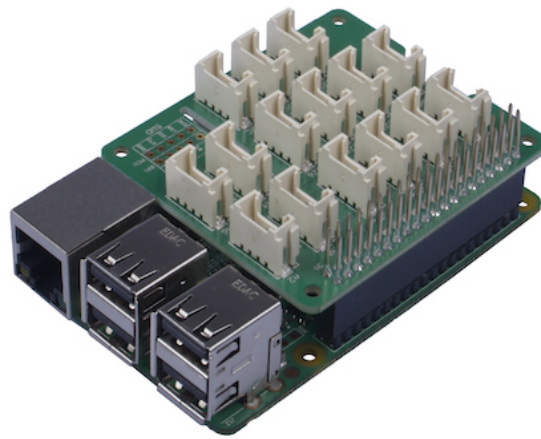


## Base Hat 配置

第一步. 关闭树莓派

```
sudo shutdown -h now
```

将Base Hat插到树莓派上

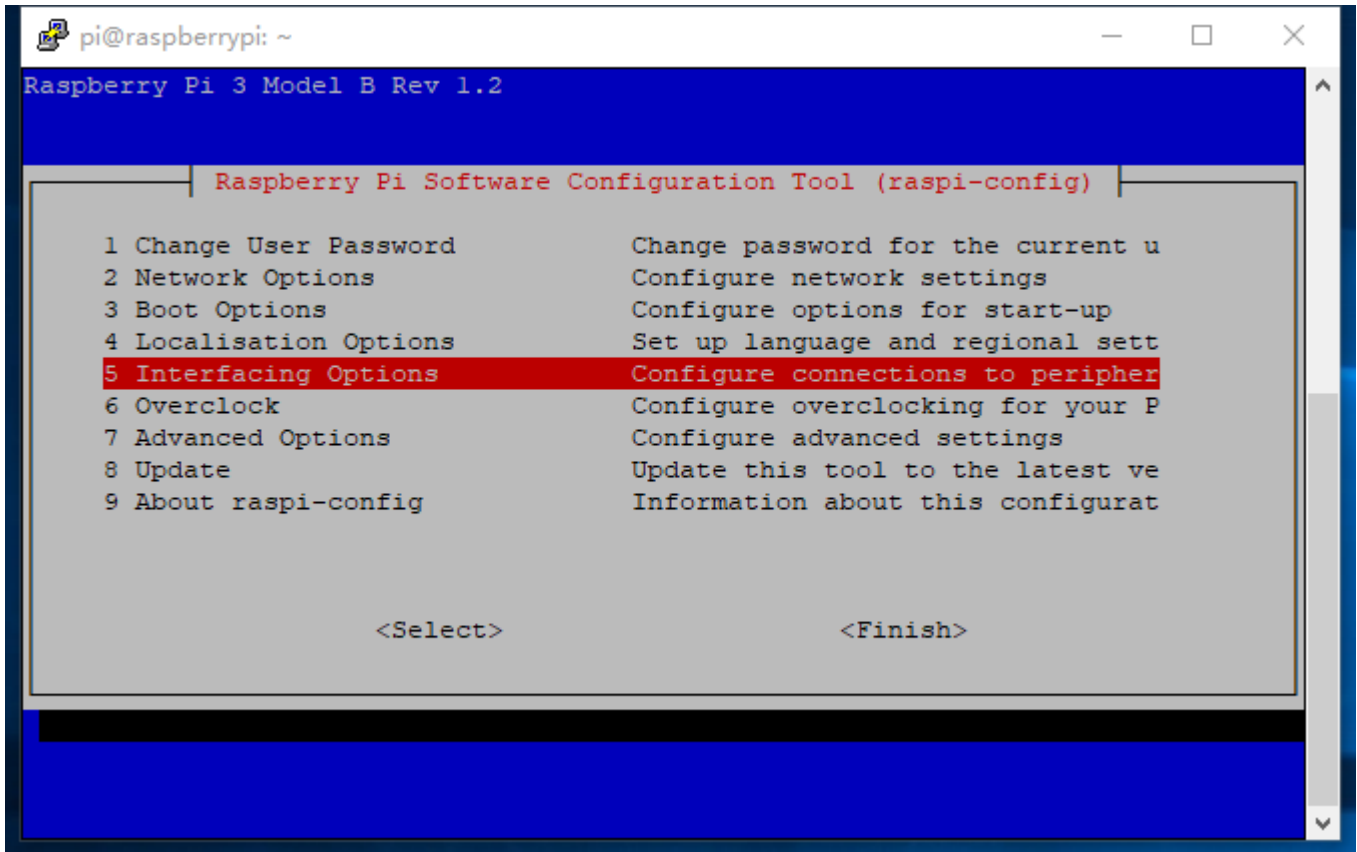


第二步. 通过micro-usb线给树莓派重新上电并始能I2C接口

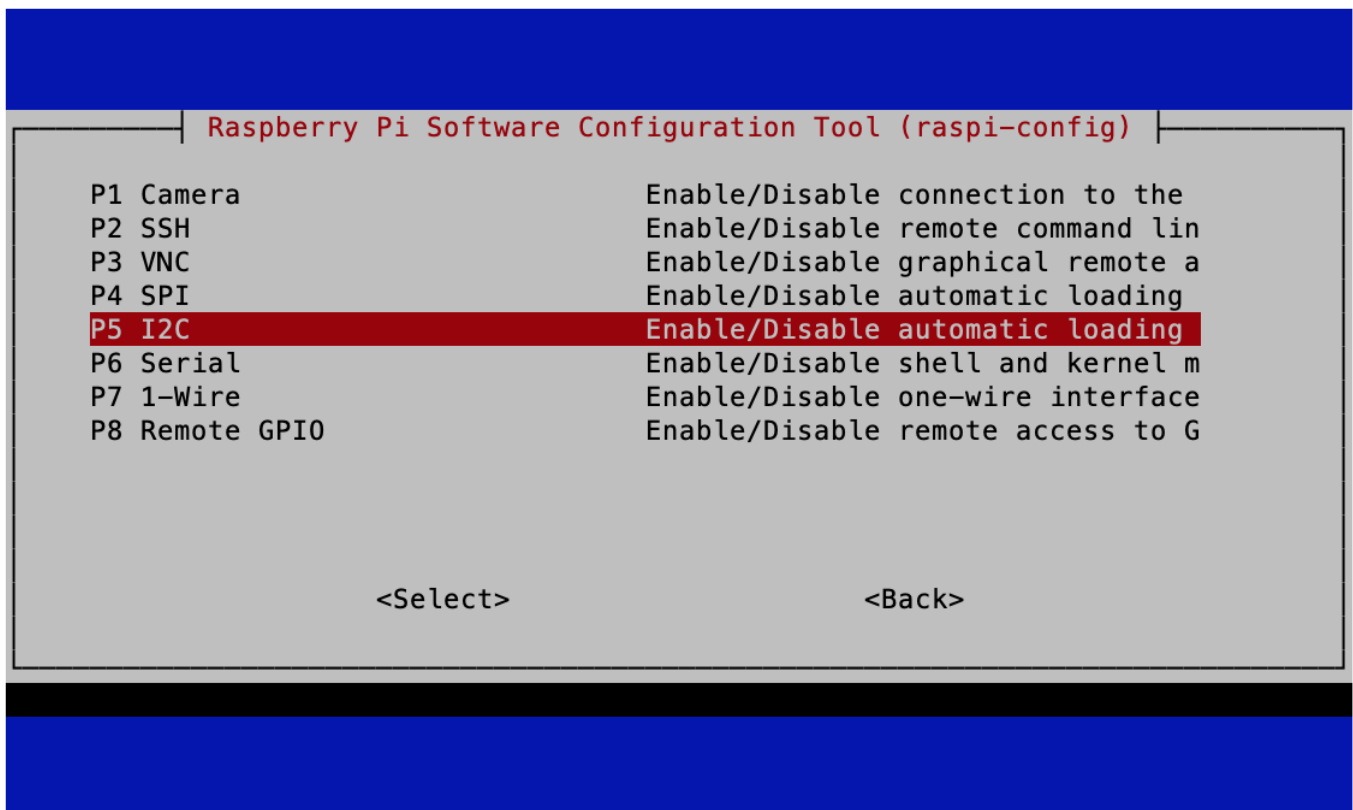
打开raspi-config

```
sudo raspi-config
```

选择"5 Interfacing Options"并按"Enter"键

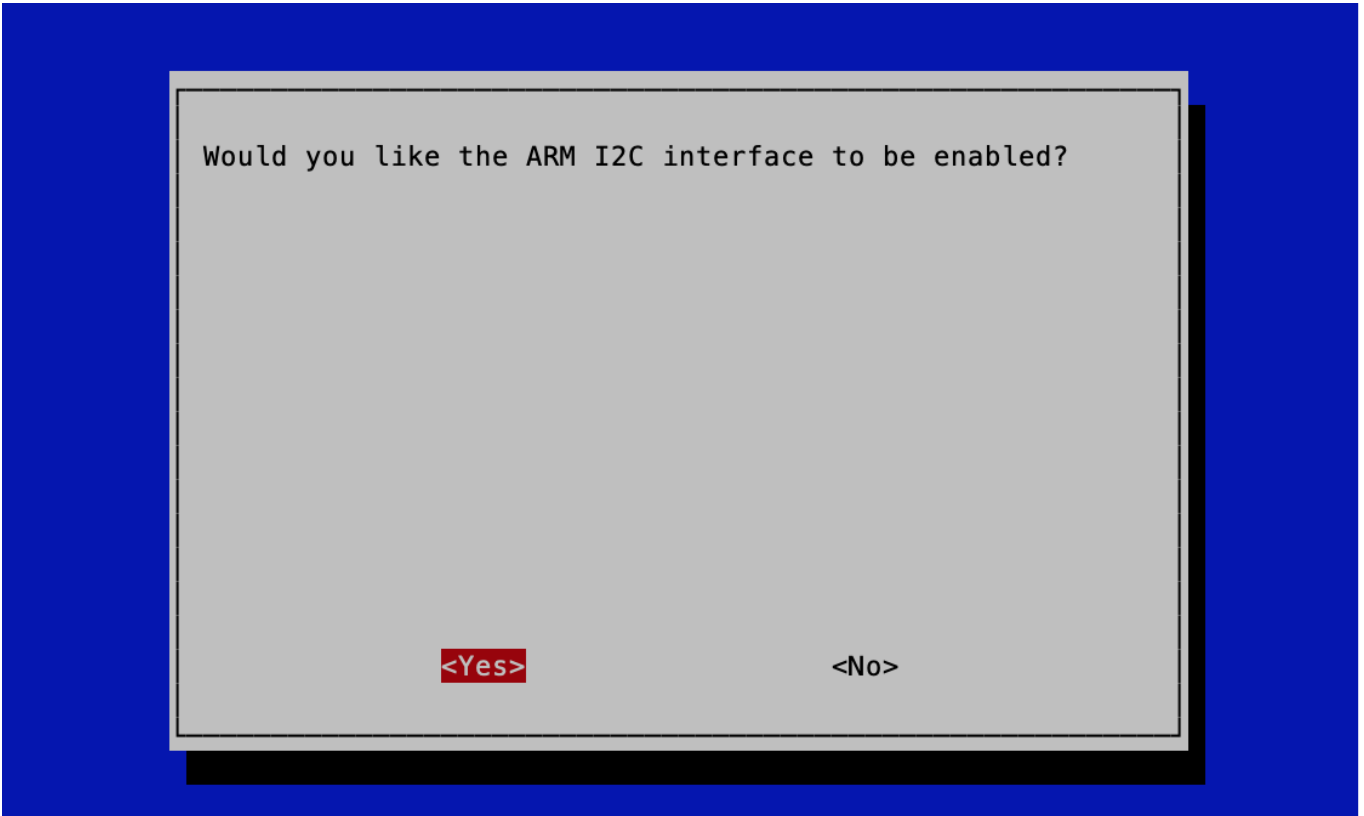


选择"P5 I2C"并按"Enter"键

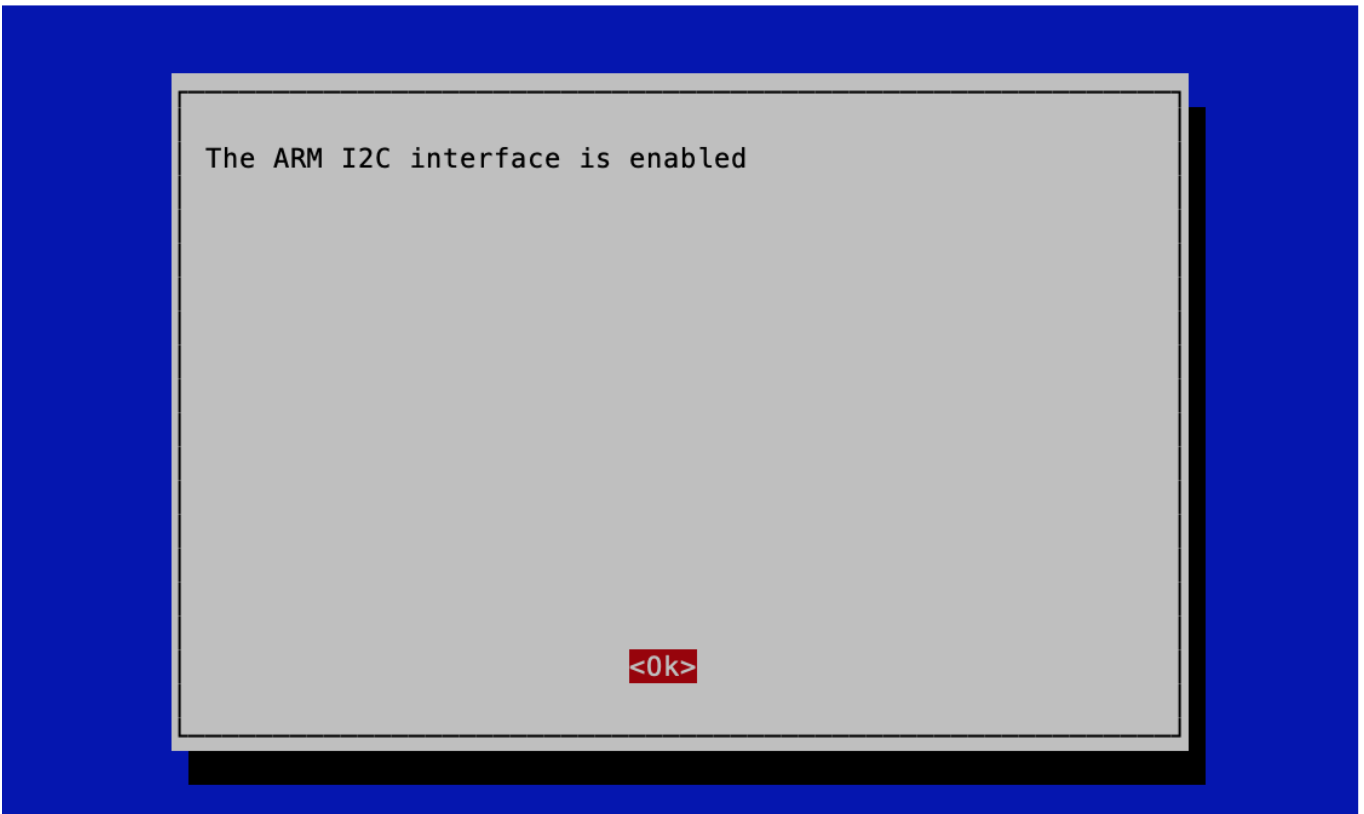


选择"Yes"并按"Enter"键去打开I2C接口





选择"Ok"



选择"Finish"去保存当前配置

Raspberry Pi 3 Model B Rev 1.2

Raspberry Pi Software Configuration Tool (raspi-config)

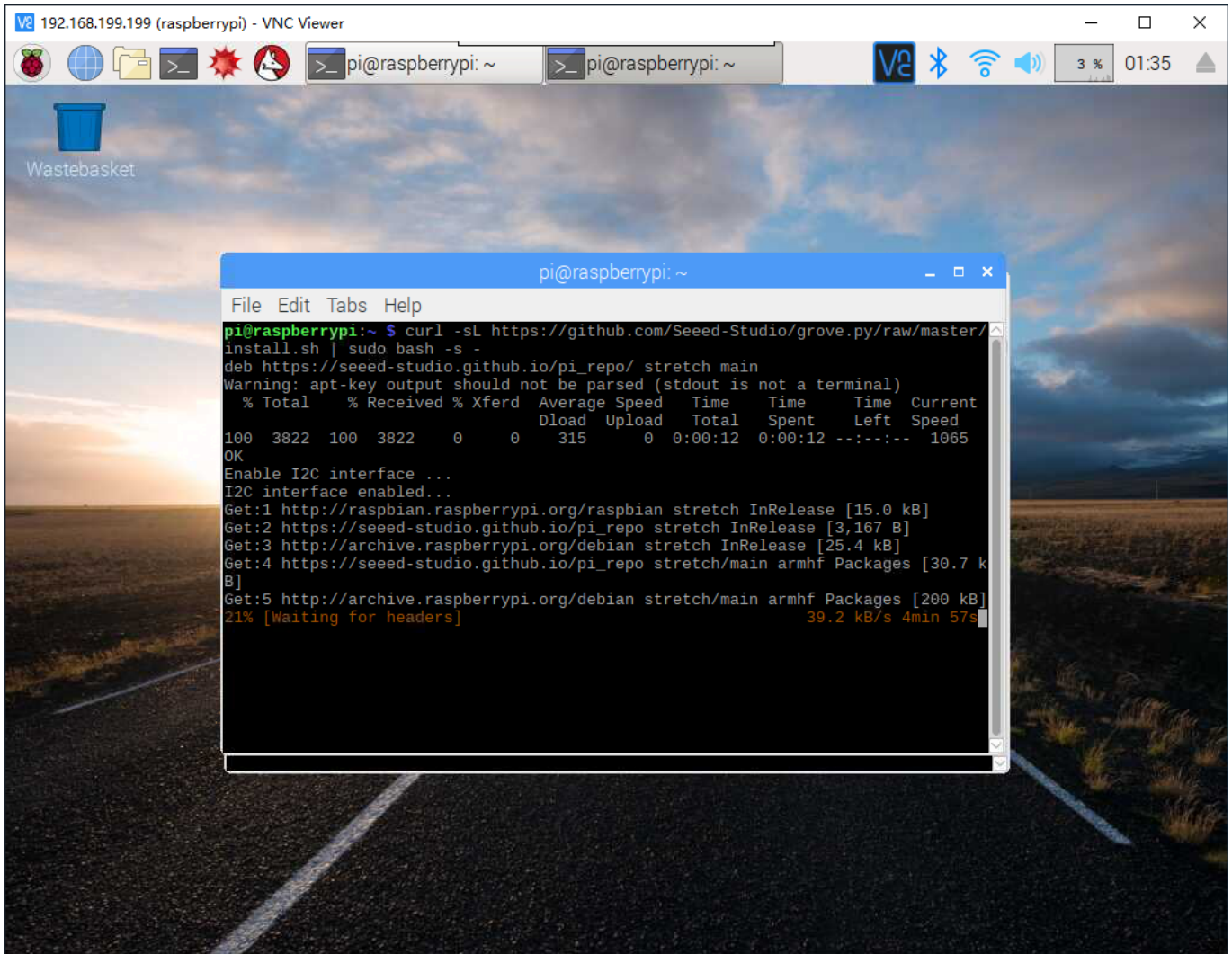
1 Change User Password	Change password for the current u
2 Network Options	Configure network settings
3 Boot Options	Configure options for start-up
4 Localisation Options	Set up language and regional sett
5 Interfacing Options	Configure connections to peripher
6 Overclock	Configure overclocking for your P
7 Advanced Options	Configure advanced settings
8 Update	Update this tool to the latest ve
9 About raspi-config	Information about this configurat

<Select>

<Finish>

第三步. 执行下面的命令去一键安装所有依赖项和最新版的 Grove.py

```
curl -sL https://github.com/Seeed-Studio/grove.py/raw/master/install.sh | sudo  
bash -s -
```



如果所有都顺利完成，你将看到下面的通知

```

Successfully installed grove.py-0.6
#####
Lastest Grove.py from github install complete !!!!!
#####

```

第四步. 假如不使用一键安装，你也可以自己安装所有[依赖项](#)

第五步. 克隆最新版的 python.py的库

```
git clone https://github.com/Seeed-Studio/grove.py
```

## Grove – LED button 历程

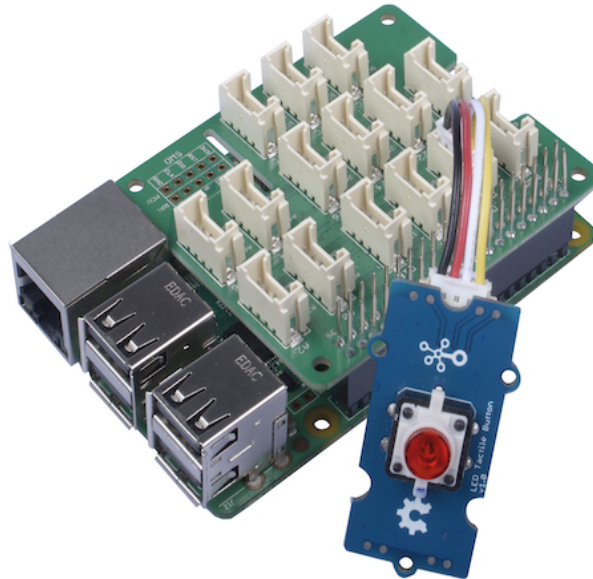
树莓派的所有配置完成后，我们现在可以运行LED历程代码。

### 硬件连接

第一步. 将Grove - Red LED Button和Base Hat的D5进行连接

第二步. 将Base Hat插到树莓派上面

第三步. 通过micro USB将树莓派上电



更新代码

第一步. 运行下面的命令行去创建python文件

```
cd grove.py
nano example.py
```

第二步. 将下面的代码复制到刚刚创建的python文件中

!!!Caution

请确定文本编辑器是UNIX格式的

```
#!/usr/bin/env python

import time
from grove.grove_ryb_led_button import GroveLedButton
```

```
def main():
    ledbtn = GroveLedButton(5)

    while True:
        time.sleep(1)

if __name__ == '__main__':
    main()
```

### 第三步. 运行代码

```
python example.py
```

单击LED按钮时，LED将变为“ON”模式此时LED点亮，如果长按则会变为“OFF”此时LED熄灭。如果双击LED按钮进入Blink模式此时LED将闪烁。

```
pi@raspberrypi:~/grove.py $ python example.py
turn on LED
turn on LED
turn off LED
turn on LED
blink LED
^CTraceback (most recent call last):
  File "./example.py", line 17, in <module>
    main()
  File "./example.py", line 14, in main
    time.sleep(1)
KeyboardInterrupt
pi@raspberrypi:~/grove.py $
```

### 闪烁代码的解释

在python中，由于模块相互引用，不同的模块可能有不同的“main”定义，但是每次只能有一个入口程序。入口程序的选择取决于 **name** 的值。if **name=='main'** 相等，这意味着它是python真正的入口。

```
if __name__ == '__main__':
    main()
```

## 基于树莓派的Grove Base Kit

让我们现在马上去探索Grove系统吧！为了更好的使用基础Grove模块我们为你设计了八个基本Grove模块的指南。本节介绍如何在实际应用程序中组合和应用模块。

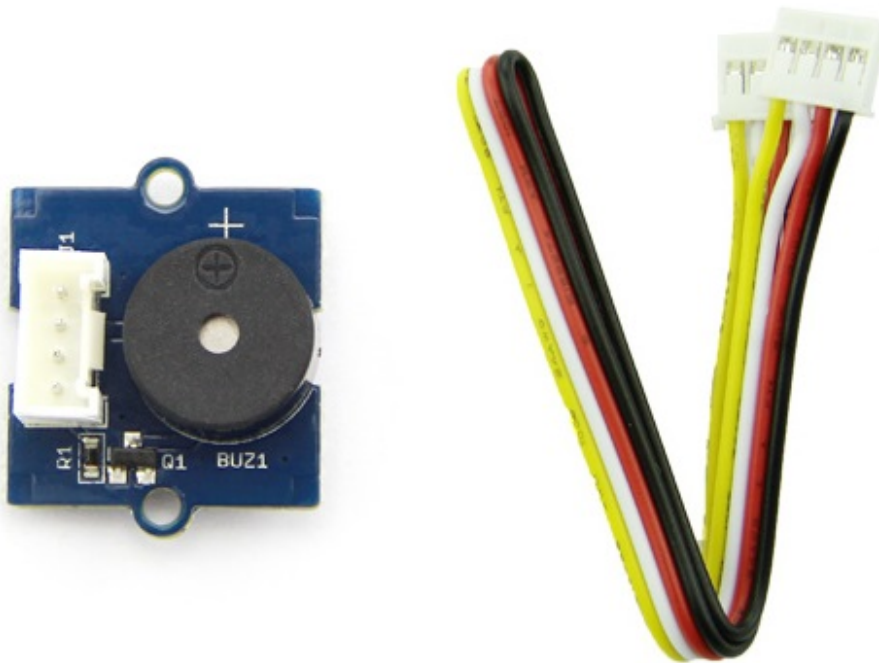
### 条件

在开始学习Grove之前，你需要树莓派和Python编程语言的基本知识。请确保你已经完成了前面的基础建立指南和LED闪烁历程同时你还需要确定Grove Base Hat在树莓派上面是否完美工作。

## 学习目标

- 能够通过Grove Base Hat和Grove模块去设计一些简单应用
- 能够展示Grove Starter Kit中的所有模块并将相关模块用于您自己的项目
- 能够识别Grove Starter Kit和对应的那些应用中的模块类型
- 理解数字和模拟量的差异

## 课时1: Buzzer



## 目的

使用Buzzer去产生一些噪声和一些特殊频率的声调

## 硬件准备

### 自己准备

- micro-USB线
- 树莓派3b
- 电脑

### Kit中的模块

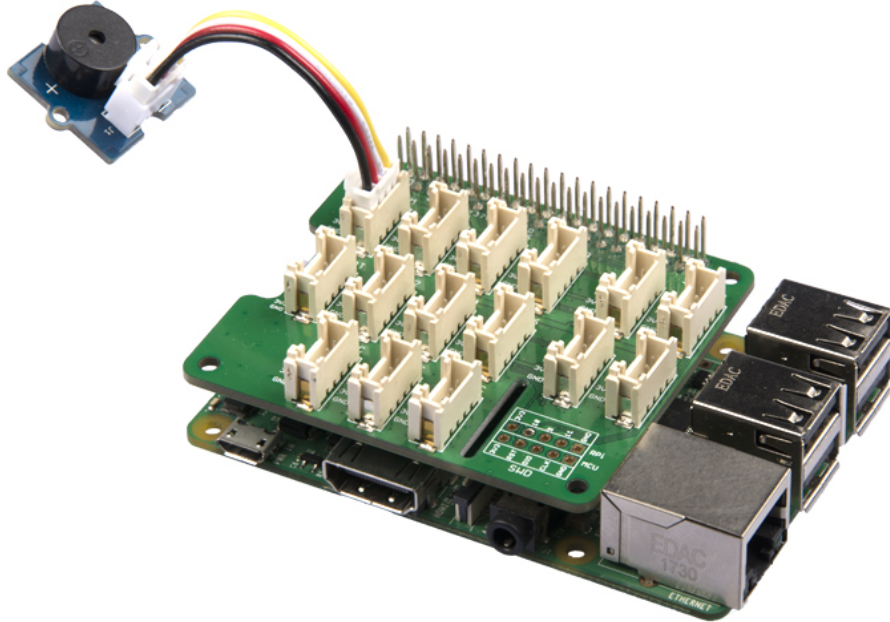
- Grove Base Hat
- Grove cable
- Grove – Buzzer

## 硬件连接

第一步. 将Grove - Buzzer通过Grove线连接到Base Hat的PWM接口上，同时将Base Hat插到树莓派上



第二步. 通过micro USB将树莓派上电



软件编程

!!!Note

请确定你已经将 `python.py` 的库已经克隆到了你的树莓派上面了

第一步. 运行下面的命令

```
cd grove.py
nano lesson_1.py
```

第二步. 复制下面的代码

```
#!/usr/bin/env python
import time
from mraa import getGpioLookup
from upm import pyupm_buzzer as upmBuzzer
```

```
def main():
    # Grove - Buzzer connected to PWM port
    buzzer = upmBuzzer.Buzzer(getGpioLookup('GPIO12'))

    CHORDS = [upmBuzzer.BUZZER_DO, upmBuzzer.BUZZER_RE, upmBuzzer.BUZZER_MI,
              upmBuzzer.BUZZER_FA, upmBuzzer.BUZZER_SOL, upmBuzzer.BUZZER_LA,
              upmBuzzer.BUZZER_SI]
    for i in range(0, len(CHORDS)):
        buzzer.playSound(CHORDS[i], 500000)
        time.sleep(0.1)

    del buzzer
    print('application exiting...')

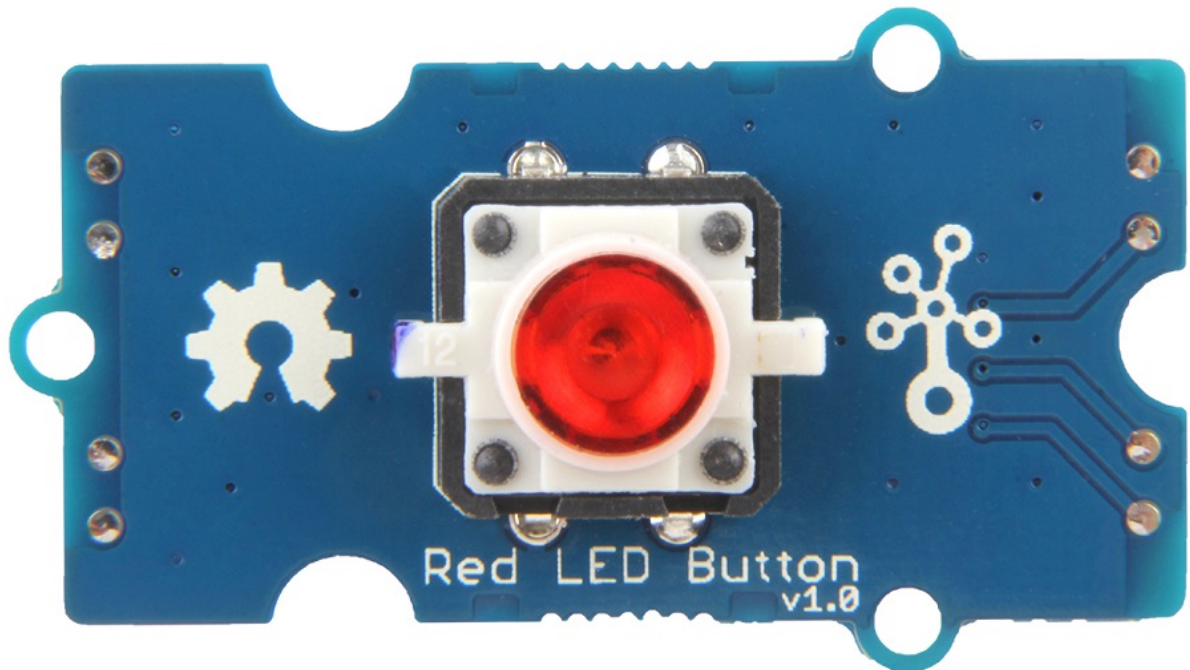
if __name__ == '__main__':
    main()
```

### 第三步. 运行程序

```
sudo chmod +x lesson_1.py
sudo ./lesson_1.py
```

如果一切都顺利，那么你将听到蜂鸣器发出"Do Re Mi Fa So La Si"的声音

### 课时2: Red LED Button



## 目的

使用Grove - Red LED Button去控制Grove - Red LED Button的LED灯闪烁同时让Grove - Buzzer发出不同音效

## 硬件准备

### 自己准备

- micro-USB线
- 树莓派3B
- 电脑

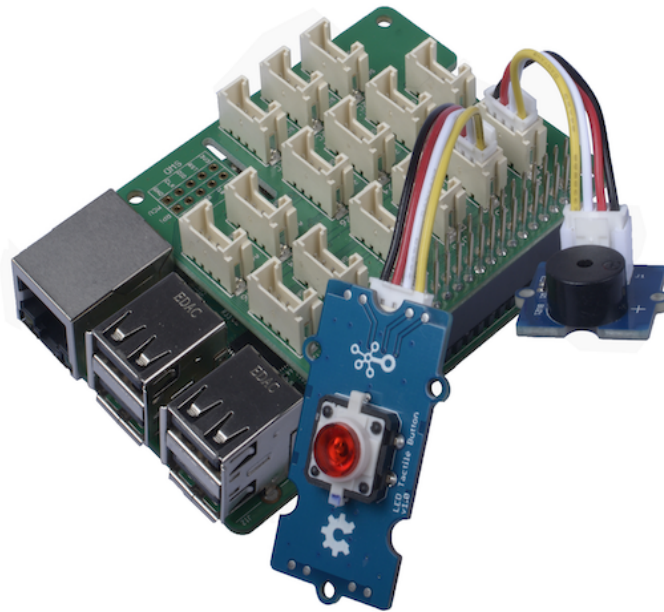
### Kit中的模块

- Grove Base Hat
- Grove cable
- Grove - Red LED Button
- Grove - Buzzer

## 硬件连接

第一步. 使用Grove电缆将Grove - Buzzer连接到PWM端口，使用Grove -Red LED Button连接到Base Hat的D5，然后插入Raspberry Pi

第二步. 通过micro USB将树莓派上电



## 软件编程

### !!!Note

请确定你已经将 `python.py` 的库已经克隆到了你的树莓派上面了

第一步. 运行下面的命令行去创建python文件

```
cd grove.py
nano lesson_2.py
```

第二步. 复制下面的代码

```
#!/usr/bin/env python

import time
from mraa import getGpioLookup
from upm import pyupm_buzzer as upmBuzzer

from grove.button import Button
```

```
from grove.grove_ryb_led_button import GroveLedButton

def main():
    # Grove - LED Button connected to port D5
    button = GroveLedButton(5)

    # Grove - Buzzer connected to PWM port
    buzzer = upmBuzzer.Buzzer(getGpioLookup('GPI012'))

    def on_event(index, event, tm):
        if event & Button.EV_SINGLE_CLICK:
            print('single click')
            button.led.light(True)
            buzzer.playSound(upmBuzzer.BUZZER_DO, 500000)

        elif event & Button.EV_LONG_PRESS:
            print('long press')
            button.led.light(False)
            buzzer.playSound(upmBuzzer.BUZZER_DO, 1000000)

    button.on_event = on_event

    while True:
        time.sleep(1)

if __name__ == '__main__':
    main()
```

### 第三步. 运行代码

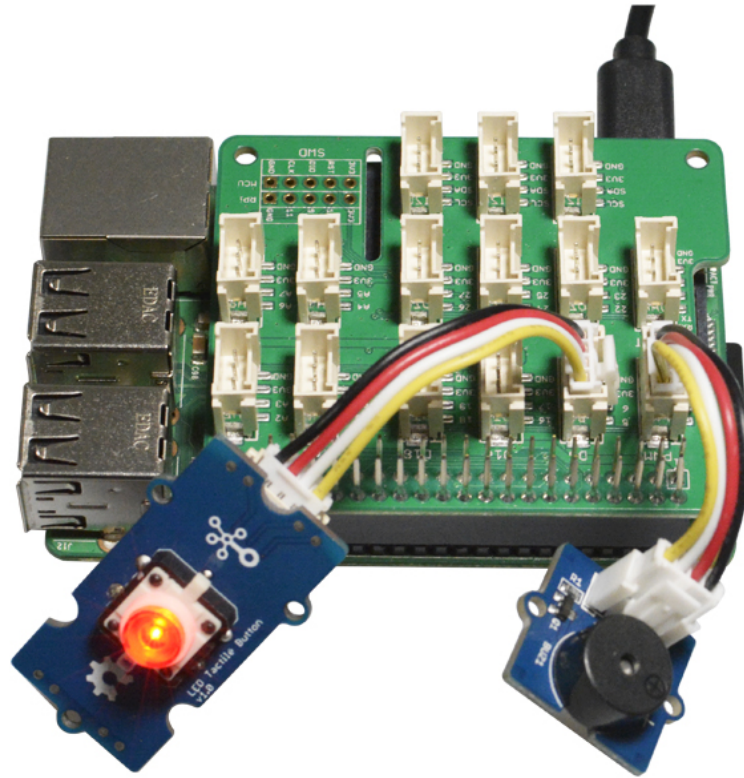
```
sudo chmod +x lesson_2.py
sudo ./lesson_2.py
```

!!!Success

如果所以都顺利完成了，你将发现当你长按LED按键时，蜂鸣器将发出一个较长的"Do"的声音并你LED灯将会熄灭，当你点击LED按键时，蜂鸣器将发出一个较短的"Do"的声音并且你的LED灯将会常亮

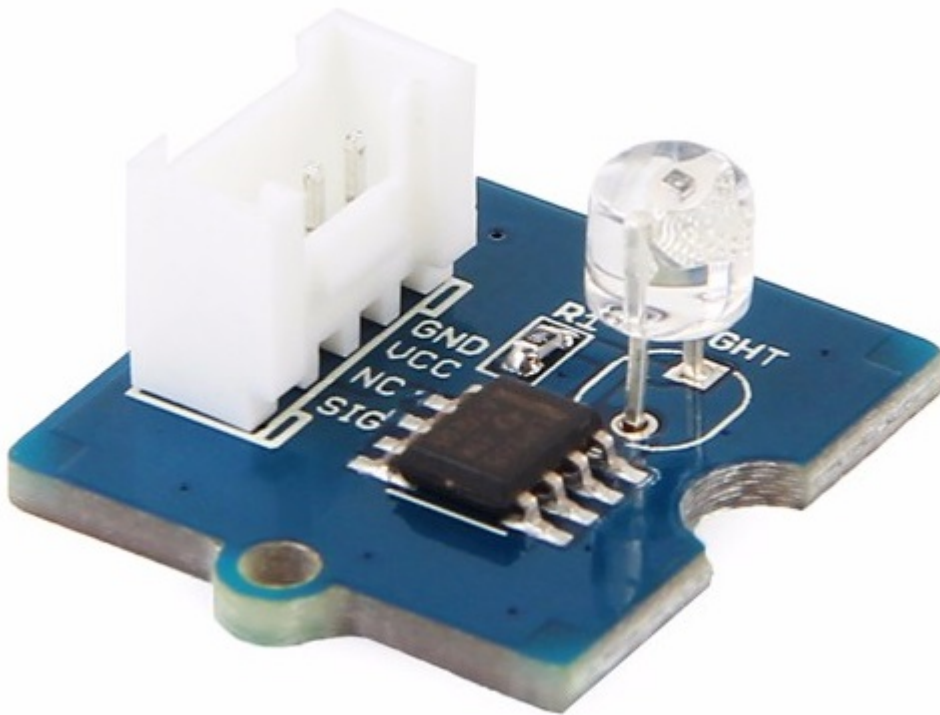
```
pi@raspberrypi:~/grove.py $ sudo ./lesson_2.py
single click
single click
single click
long press
single click
long press
long press
Traceback (most recent call last):
```

```
File "./lesson2.py", line 34, in <module>
    main()
File "./lesson2.py", line 31, in main
    time.sleep(1)
KeyboardInterrupt
^Cpi@raspberrypi:~/grove.py $
```



课时3: Grove - Light Sensor





## 目的

在这一课时， we will show you how to use Grove - Light Sensor to control Grove - Servo. In this case, servo rotation angle varies with light intensity. 在这堂课，我们将给您展示怎么使用Grove - Light Sensor去控制Grove - Servo。实物效果是servo的旋转角度随着光照强度的改变而改变。 **硬件准备**

## 自己准备

- micro-USB线
- 树莓派3B
- 电脑

## Kit中的模块

- Grove Base Hat
- Grove cable
- Grove - Light Sensor
- Grove - Servo

## 硬件连接

第一步. 将Grove - Light Sensor连接到A0， Grove - Servo连接到PWM接口

第二步. 将Base Hat插到树莓派上面

第三步. 通过micro USB将树莓派上电



## 软件编程

### !!!Note

请确定你已经将 `python.py` 的库已经克隆到了你的树莓派上面了

第一步. 运行下面的命令行去创建python文件

```
cd grove.py
nano lesson_3.py
```

第二步. 复制下面的代码

```
#!/usr/bin/env python

import time

from grove.grove_servo import GroveServo
from grove.grove_light_sensor_v1_2 import GroveLightSensor

def main():
    # Grove - Servo connected to PWM port
    servo = GroveServo(12)
```

```
# Grove - Light Sensor connected to port A0
sensor = GroveLightSensor(0)

while True:
    angle = sensor.light * 180 / 1000
    print('light value {}, turn to {} degree.'.format(sensor.light, angle))
    servo.setAngle(angle)

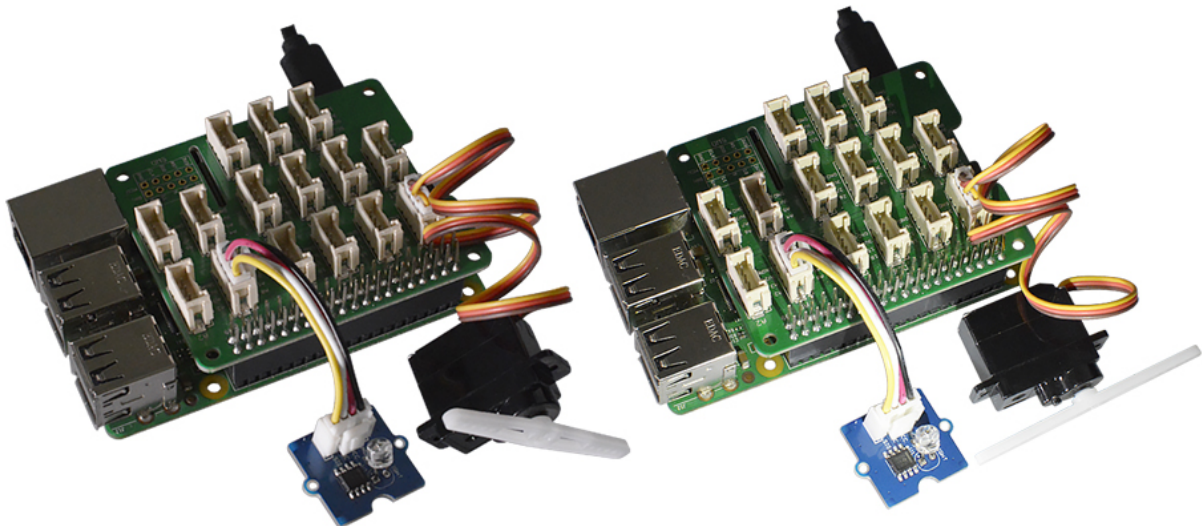
    time.sleep(1)

if __name__ == '__main__':
    main()
```

### 第三步. 运行代码

```
sudo chmod +x lesson_3.py
sudo ./lesson_3.py
```

如果一切都顺利，随着光照强度的改变不同的角度将会随之改变



```
pi@raspberrypi:~/grove.py $ sudo ./lesson_3.py
light value 300, turn to 113 degree.
light value 80, turn to 80 degree.
```

```

light value 166, turn to 165 degree.
light value 498, turn to 132 degree.
light value 601, turn to 60 degree.
light value 200, turn to 21 degree.
light value 459, turn to 99 degree.
light value 172, turn to 173 degree.
light value 319, turn to 138 degree.
^CTraceback (most recent call last):
  File "./lesson3.py", line 23, in <module>
    main()
  File "./lesson3.py", line 20, in main
    time.sleep(1)
KeyboardInterrupt
pi@raspberrypi:~/grove.py $

```

## 课时4: Motion Sensor & Relay



### 目的

使用Grove - mini PIR motion sensor去探测人类的活动，当探测到有人时照明灯点亮。

### 硬件准备

#### 自己准备

- micro-USB线
- 树莓派3B
- 电脑

#### Kit中的模块

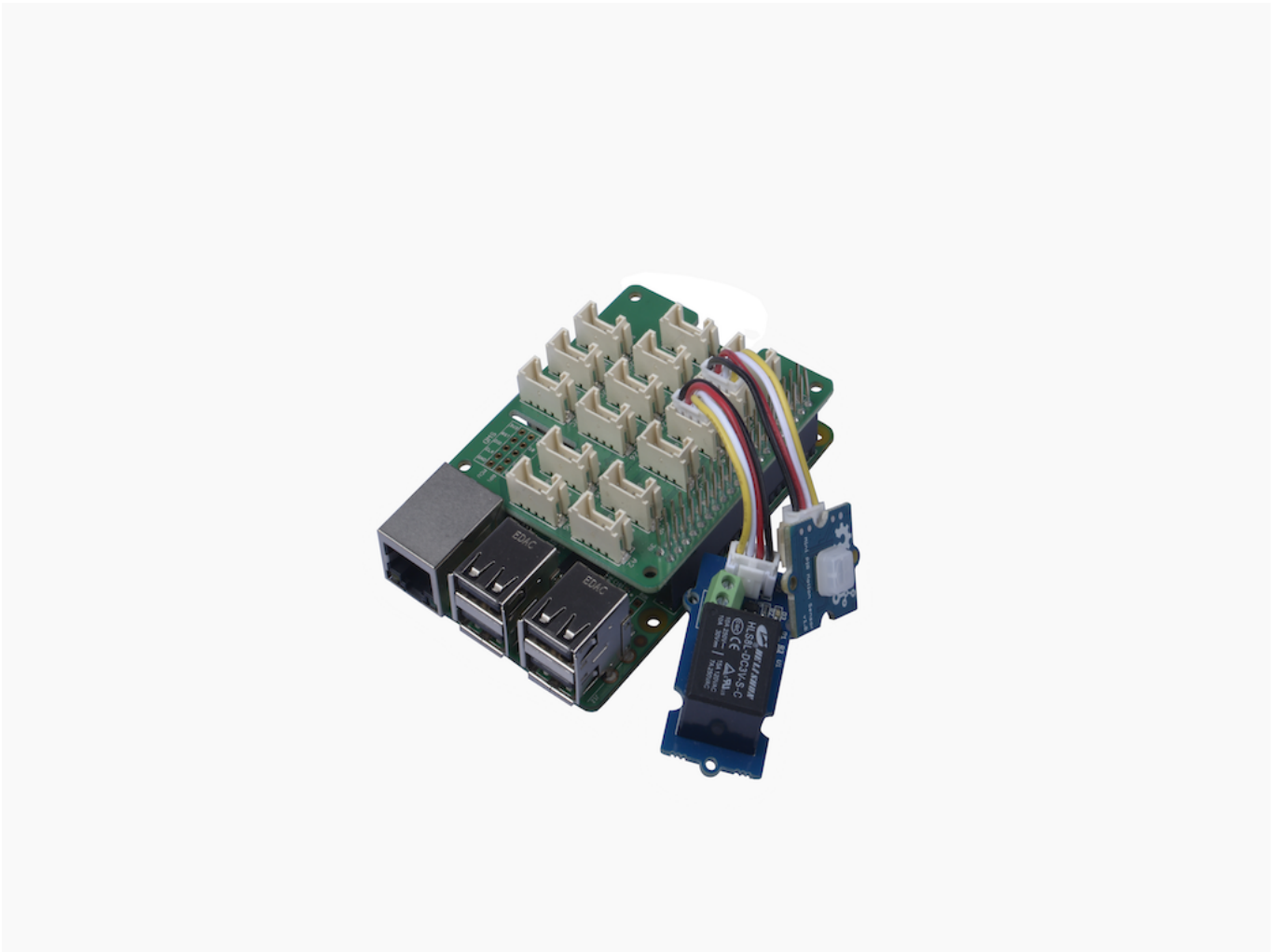
- Grove Base Hat
- Grove cable
- Grove - mini PIR motion sensor
- Grove - Relay

### 硬件连接

第一步. 将Grove - mini PIR motion sensor连接到D5， Grove - Relay连接到D16

第二步. 将Base Hat插到树莓派上面

第三步. 通过micro USB将树莓派上电



软件编程

!!!Note

请确定你已经将 `python.py` 的库已经克隆到了你的树莓派上面了

第一步. 运行下面的命令行去创建python文件

```
cd grove.py  
nano lesson_4.py
```

第二步. 复制下面的代码

```
#!/usr/bin/env python  
  
import time
```

```
from grove.grove_mini_pir_motion_sensor import GroveMiniPIRMotionSensor
from grove.grove_relay import GroveRelay

def main():
    # Grove - mini PIR motion sensor connected to port D5
    sensor = GroveMiniPIRMotionSensor(5)

    # Grove - Relay connected to port D16
    relay = GroveRelay(16)

    def on_detect():
        print('motion detected')

        relay.on()
        print('relay on')

        time.sleep(1)

        relay.off()
        print('relay off')

    sensor.on_detect = on_detect

    while True:
        time.sleep(1)

if __name__ == '__main__':
    main()
```

### 第三步. 运行代码

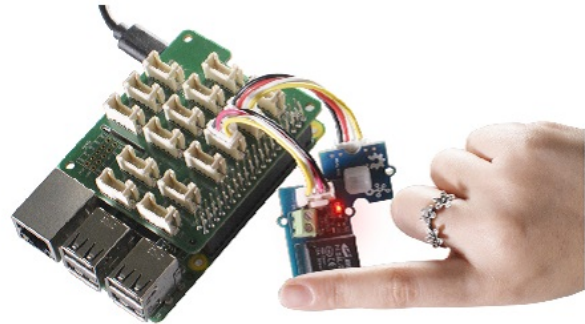
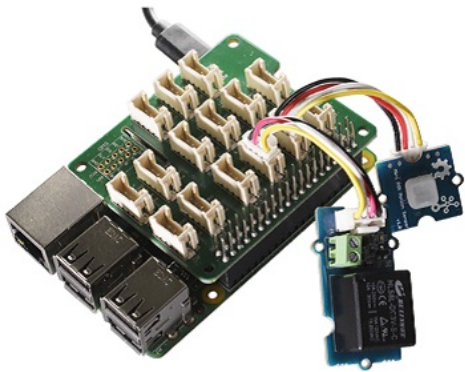
```
sudo chmod +x lesson_4.py
sudo ./lesson_4.py
```

如果一切都顺利，你观察到当有人接近时照明灯将打开

```
pi@raspberrypi:~/grove.py $ sudo ./lesson_4.py
motion detected
relay on
relay off
motion detected
relay on
relay off
^CTraceback (most recent call last):
  File "./lesson_4.py", line 33, in <module>
    main()
  File "./lesson_4.py", line 30, in main
    time.sleep(1)
```



```
KeyboardInterrupt
pi@raspberrypi:~/grove.py $
```



## 课时5: Ultrasonic Sensor & Relay



### 目的

在这一课时，我们使用Grove - Ultrasonic Ranger去检测距离，当有人靠近时Grove - Relay将被打开

### 硬件准备

#### 自己准备

- micro-USB线
- 树莓派3B
- 电脑

#### Kit中的模块

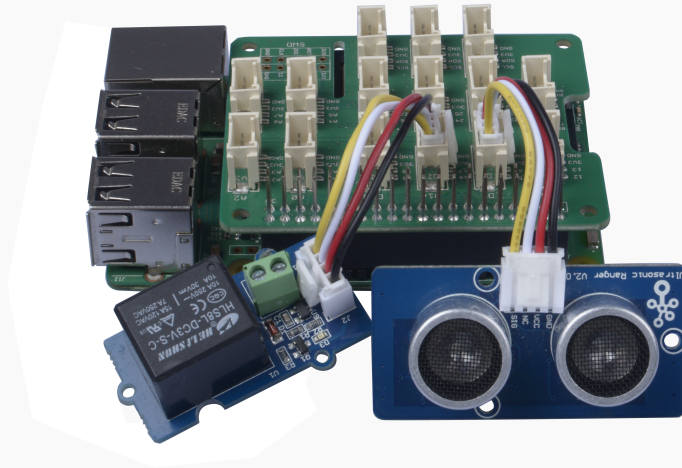
- Grove Base Hat
- Grove cable
- Grove - Ultrasonic Ranger
- Grove - Relay

## 硬件连接

第一步. 将Grove - Ultrasonic Ranger连接到D5, 将Grove - Relay连接到D16

第二步. 将Base Hat插到树莓派上面

第三步. 通过micro USB将树莓派上电.



## 软件编程

### !!!Note

请确定你已经将 `python.py` 的库已经克隆到了你的树莓派上面了

第一步. 运行下面的命令行去创建python文件

```
cd grove.py
nano lesson_5.py
```

第二步. 复制下面的代码

```
#!/usr/bin/env python

import time
```

```
from grove.grove_relay import GroveRelay
from grove.grove_ultrasonic_ranger import GroveUltrasonicRanger

def main():
    # Grove - Ultrasonic Ranger connected to port D5
    sensor = GroveUltrasonicRanger(5)

    # Grove - Relay connected to port D16
    relay = GroveRelay(16)

    while True:
        distance = sensor.get_distance()
        print('{} cm'.format(distance))

        if distance < 20:
            relay.on()
            print('relay on')

            time.sleep(1)

            relay.off()
            print('relay off')

            continue

        time.sleep(1)

if __name__ == '__main__':
    main()
```

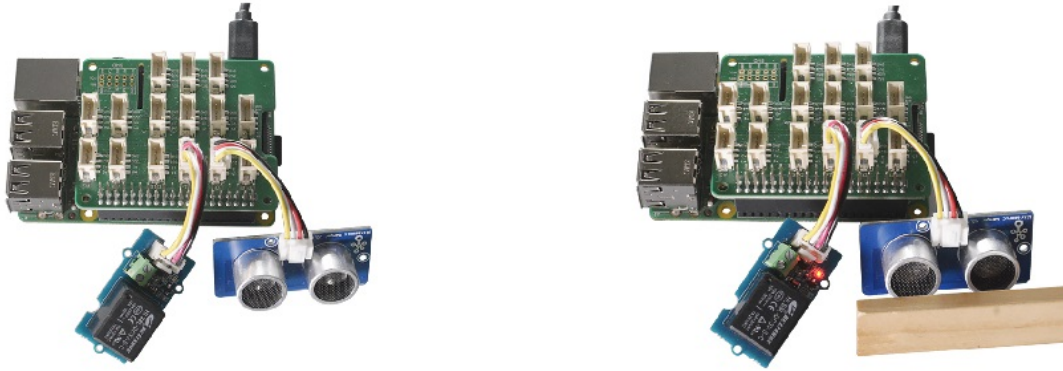
### 第三步. 运行代码

```
sudo chmod +x lesson_5.py
sudo ./lesson_5.py
```

如果一切都顺利，当物体接近Grove - Ultrasonic Ranger时，Grove -Relay将被打开

```
pi@raspberrypi:~/grove.py $ sudo ./lesson_5.py
253.722585481 cm
253.739028141 cm
252.896341784 cm
1.20442489098 cm
relay on
relay off
4.51762100746 cm
relay on
relay off
253.985668051 cm
```

```
^CTraceback (most recent call last):
  File "./lesson_5.py", line 34, in <module>
    main()
  File "./lesson_5.py", line 31, in main
    time.sleep(1)
KeyboardInterrupt
pi@raspberrypi:~/grove.py $
```



通过课时4和课时5，你可以分析出来Grove - mini PIR motion sensor和Grove Ultrasonic Ranger的优势和劣势吗？

课时6: LCD



## 目的

使用Grove - 16\*2 LCD屏幕去显示“Hello World”

## 硬件准备

### 自己准备

- micro-USB线
- 树莓派3B
- 电脑

### Kit中的模块

- Grove Base Hat
- Grove cable
- Grove - 16\*2 LCD

## 硬件连接

第一步. 将Grove - 16\*2 LCD屏幕连接到I2C接口

第二步. 将Base Hat插到树莓派上面

第三步. 通过micro USB将树莓派上电



## 软件编程

### !!!Note

请确定你已经将 `python.py` 的库已经克隆到了你的树莓派上面了

第一步. 运行下面的命令行去创建python文件

```
cd grove.py
nano lesson_6.py
```

第二步. 复制下面的代码

```
#!/usr/bin/env python

import time

from grove.display.jhd1802 import JHD1802

def main():
```



```
# Grove - 16x2 LCD(White on Blue) connected to I2C port
lcd = JHD1802()

lcd.setCursor(0, 0)
lcd.write('hello, world!!!')

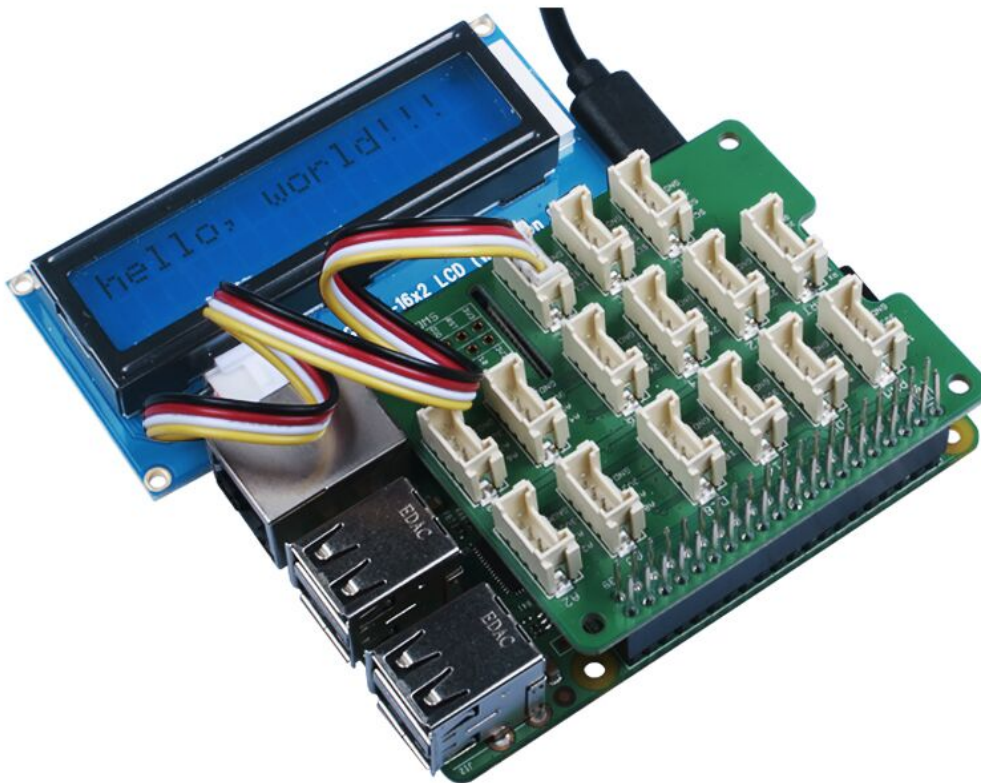
print('application exiting...')

if __name__ == '__main__':
    main()
```

第三步. 运行代码

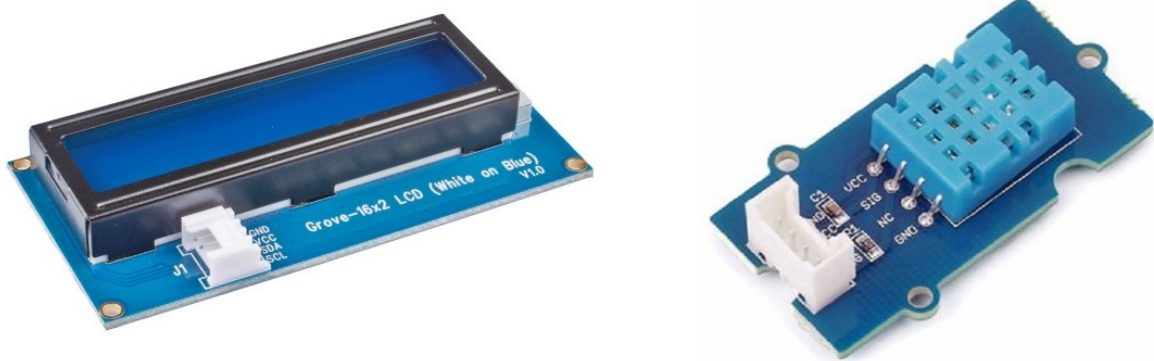
```
sudo chmod +x lesson_6.py
sudo ./lesson_6.py
```

你可以在屏幕上看到“Hello World”的字样



如果你想在Grove - 16\*2 LCD屏幕上显示一些字符，你可以改变通过`lcd.write`函数来修改

课时7: LCD & Temperature and Humidity Sensor



## 目的

将Grove - Temperature and Humidity Sensor的数据在Grove - 16\*2 LCD screen上面显示

## 硬件准备

自己准备

- micro-USB线
- 树莓派3B
- 电脑

## Kit中的模块

- Grove Base Hat
- Grove cable
- Grove - 16\*2 LCD
- Grove - Temperature and Humidity Sensor

## 硬件连接

第一步. 将Grove - 16\*2 LCD 连接到I2C, 将Grove - Temperature and Humidity Sensor 连接到D5.

第二步. 将Base Hat插到树莓派上面

第三步. 通过micro USB将树莓派上电



## 软件编程

### !!!Note

请确定你已经将 `python.py` 的库已经克隆到了你的树莓派上面了

第一步. 运行下面的命令行去创建python文件

```
cd grove.py
nano lesson_7.py
```

第二步. 复制下面的代码

```
#!/usr/bin/env python

import time

from grove.grove_temperature_humidity_sensor import DHT
from grove.display.jhd1802 import JHD1802
```

```
def main():
    # Grove - 16x2 LCD(White on Blue) connected to I2C port
    lcd = JHD1802()

    # Grove - Temperature&Humidity Sensor connected to port D5
    sensor = DHT('11', 5)

    while True:
        humi, temp = sensor.read()
        print('temperature {}C, humidity {}'.format(temp, humi))

        lcd.setCursor(0, 0)
        lcd.write('temperature: {0:2}C'.format(temp))

        lcd.setCursor(1, 0)
        lcd.write('humidity: {0:5}%'.format(humi))

        time.sleep(1)

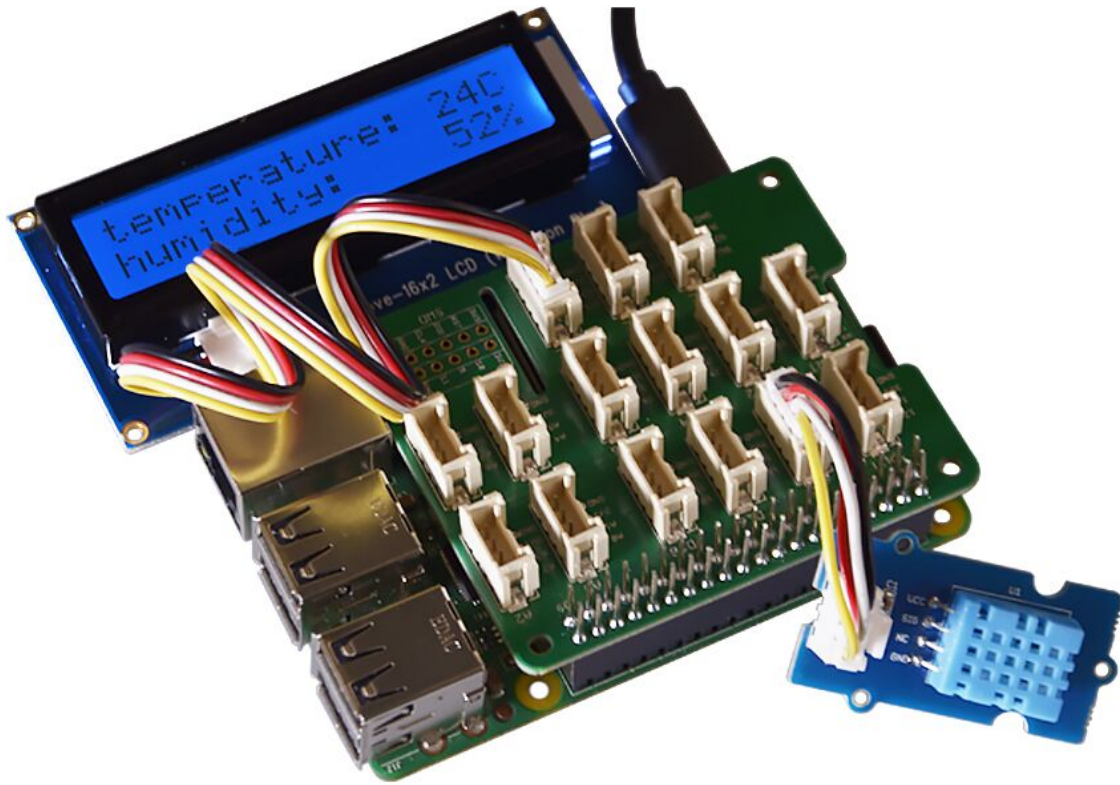
if __name__ == '__main__':
    main()
```

### 第三步. 运行代码

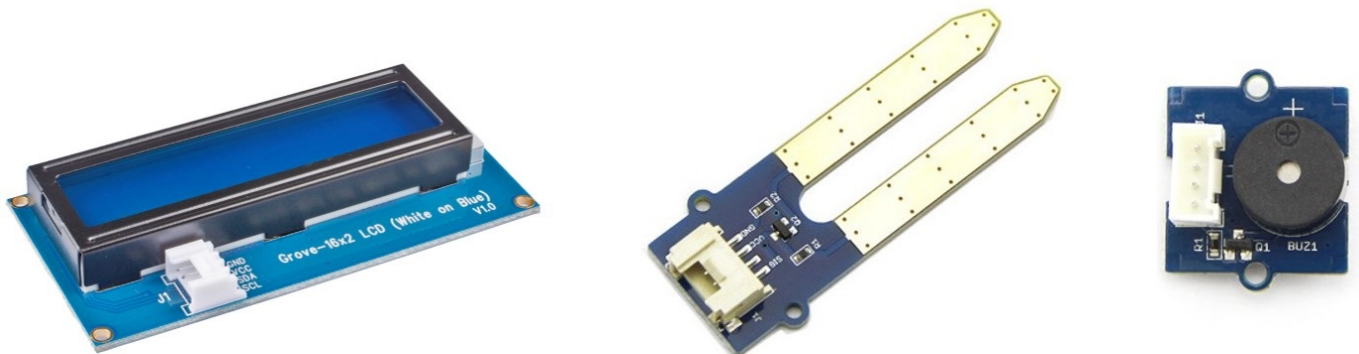
```
sudo chmod +x lesson_7.py
sudo ./lesson_7.py
```

如果一切都顺利，你将可以在LCD屏幕上面看看当前的环境的温湿度

```
pi@raspberrypi:~/grove.py $ sudo ./lesson_7.py
temperature 23C, humidity 16%
temperature 22C, humidity 17%
temperature 22C, humidity 17%
^CTraceback (most recent call last):
  File "./lesson_7.py", line 28, in <module>
    main()
  File "./lesson_7.py", line 25, in main
    time.sleep(1)
KeyboardInterrupt
pi@raspberrypi:~/grove.py $
```



## 课时8: LCD & Moisture Sensor & Buzzer



### 目的

通过Grove - 16 \* 2 LCD显示当前的湿度.当当前湿度是"wet", Grove - Buzzer将会提醒你.

### 硬件准备

### 自己准备

- micro-USB线
- 树莓派3B
- 电脑

## Kit中的模块

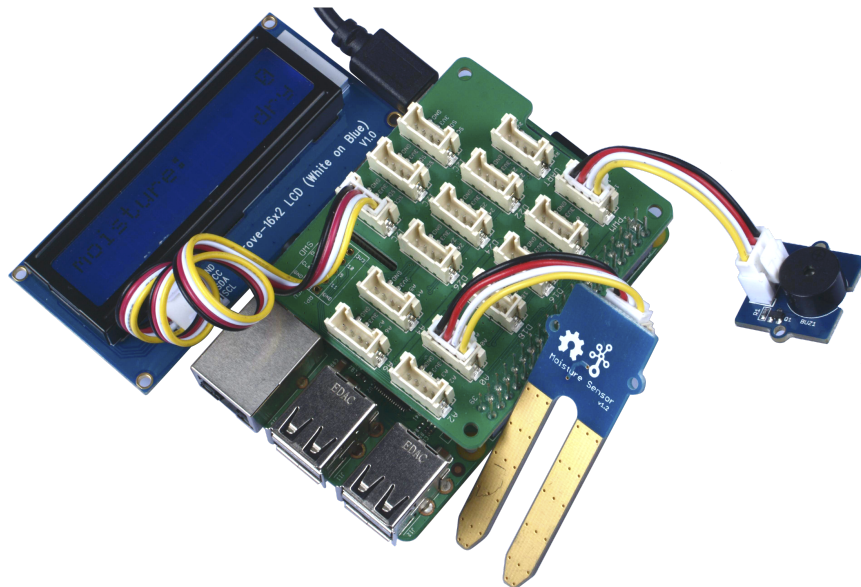
- Grove Base Hat
- Grove cable
- Grove - 16\*2 LCD
- Grove - Moisture Sensor
- Grove - Buzzer

## 硬件连接

第一步. 将Grove - 16\*2 LCD接到I2C, Grove - Moisture Sensor接到A0, Grove - Buzzer 接到PWM.

第二步. 将Base Hat插到树莓派上面

第三步. 通过micro USB将树莓派上电



## 软件编程

### !!!Note

请确定你已经将 `python.py` 的库已经克隆到了你的树莓派上面了

第一步. 运行下面的命令行去创建python文件



```
cd grove.py
nano lesson_8.py
```

第二步. 复制下面的代码

```
#!/usr/bin/env python

import time
from mraa import getGpioLookup
from upm import pyupm_buzzer as upmBuzzer

from grove.grove_moisture_sensor import GroveMoistureSensor
from grove.display.jhd1802 import JHD1802

def main():
    # Grove - 16x2 LCD(White on Blue) connected to I2C port
    lcd = JHD1802()

    # Grove - Moisture Sensor connected to port A0
    sensor = GroveMoistureSensor(0)

    # Grove - Buzzer connected to port PWM
    buzzer = upmBuzzer.Buzzer(getGpioLookup('GPIO12'))

    while True:
        mois = sensor.moisture
        if 0 <= mois and mois < 300:
            level = 'dry'
        elif 300 <= mois and mois < 600:
            level = 'moist'
        else:
            level = 'wet'
            buzzer.playSound(upmBuzzer.BUZZER_D0, 200000)

        print('moisture: {}, {}'.format(mois, level))

        lcd.setCursor(0, 0)
        lcd.write('moisture: {0:>6}'.format(mois))

        lcd.setCursor(1, 0)
        lcd.write('{0:>16}'.format(level))

        time.sleep(1)

if __name__ == '__main__':
    main()
```

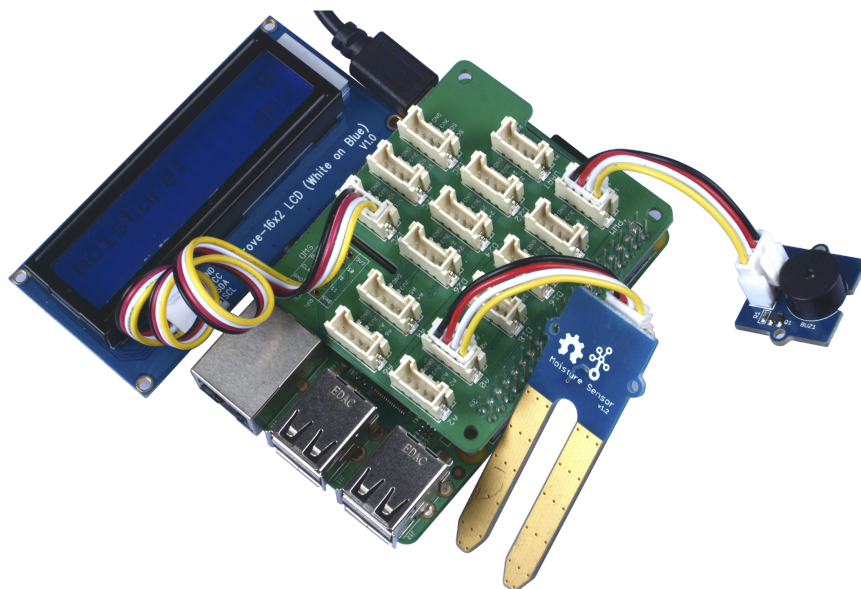
第三步. 运行代码



```
sudo chmod +x lesson_8.py
sudo ./lesson_8.py
```

如果一切都顺利，你将在LCD看到当前是湿度和湿度等级，而且当湿度等级为"wet"时，蜂鸣器将发生警报

```
pi@raspberrypi:~/grove.py $ sudo ./lesson_8.py
moisture: 0, dry
moisture: 0, dry
moisture: 396, moist
moisture: 398, moist
moisture: 407, wet
moisture: 418, wet
^CTraceback (most recent call last):
  File "./lesson_8.py", line 41, in <module>
    main()
  File "./lesson_8.py", line 38, in main
    time.sleep(1)
KeyboardInterrupt
pi@raspberrypi:~/grove.py $
```



## 资源下载

- [\[Zip\] 基于树莓派的Grove Base Hat的Eagle文件](#)

- **[Zip]** [Seeed Grove.py库](#)
- **[Zip]** [固件](#)