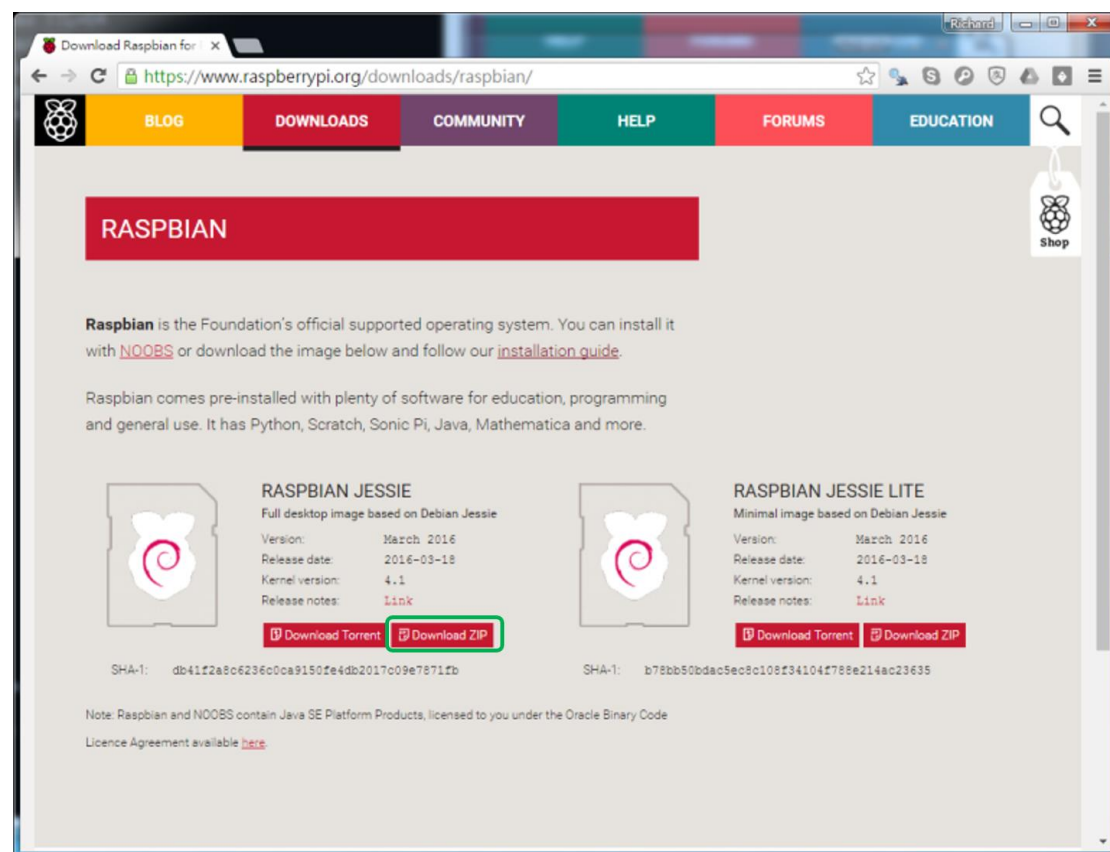
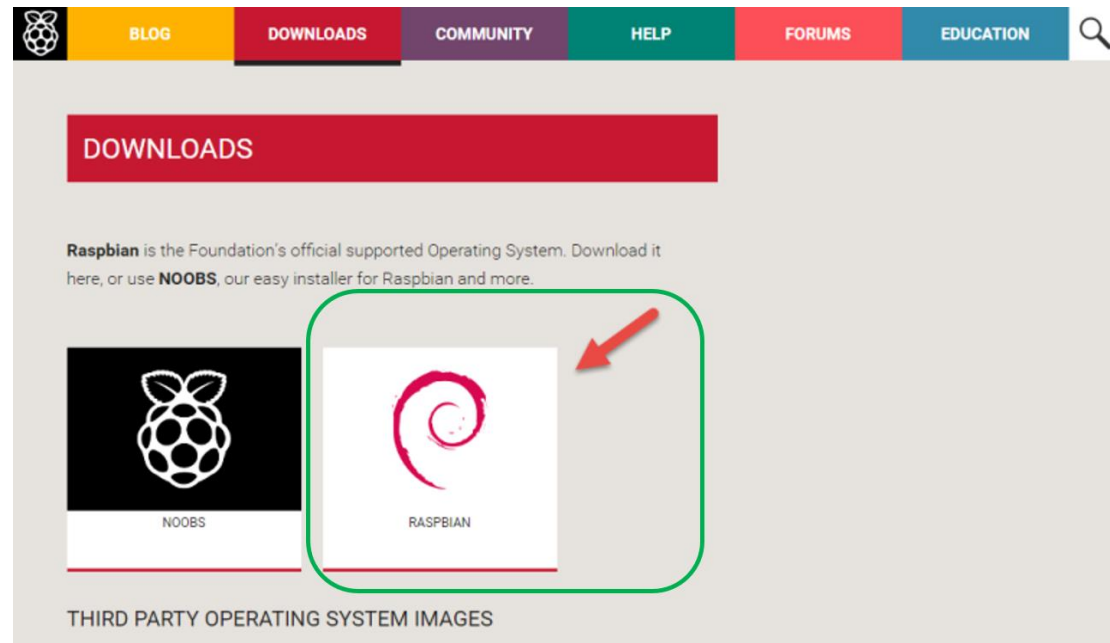


# 安裝教學

## Step1. 為 Raspberry Pi 3 安裝作業系統

在電腦端至 <https://www.raspberrypi.org/downloads/> 下載 Raspbian

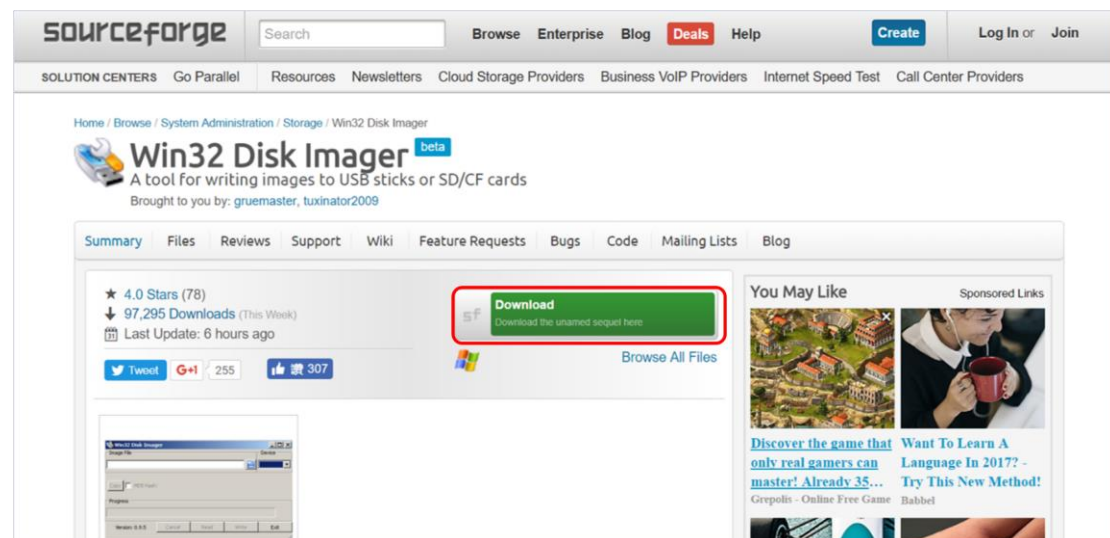


將下載而來的 2017-01-11-raspbian-jessie.zip

解壓縮可得到一個光碟映像檔.img (即樹莓派所需的 Linux OS)

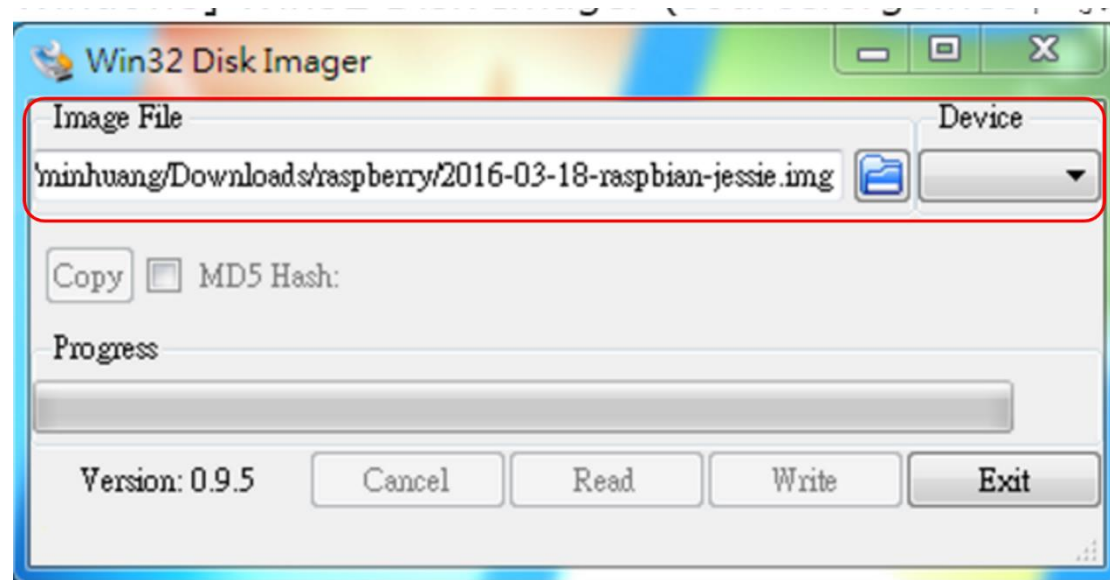
此映像檔案要透過 Win32 Disk Imager 工具軟體，  
才能寫入到 Micro SD 卡中成為可以開機的碟片，

<https://sourceforge.net/projects/win32diskimager/>



開啟 Win 32 Disk Imager 並插上 Micro SD 卡至電腦中

1. Image File 選擇剛剛下載的光碟映像檔(.img)
2. Device 則選擇 Micro SD 卡在電腦上相對應的位置



選好後按下 Write 開始寫入，需數分鐘的時間

(PS: 若中途失敗，請嘗試更換 Micro SD 卡的輸入方式，Ex:讀卡機、PC 內建 SD 卡槽...)

成功寫入後，接下來把以下設備連結至樹莓派，並接上電源進行網路的設定

1. Micro SD 卡
2. USB 鍵盤
3. USB 滑鼠
4. HDMI
5. 支援 HDMI 的螢幕

(Raspberry Pi 沒有電源開關，只要接上電源，系統就會啟動)

開機後自動安裝成功後，應可看到類似 Linux 的主頁面

### ✧ 網路的設定

須開啟 Terminal 來輸入指令(預設的帳號 **pi** 密碼 **raspberry**)

為了連上網路安裝相關套件，並使樹莓派往後一接上電源即可自動連線 WiFi

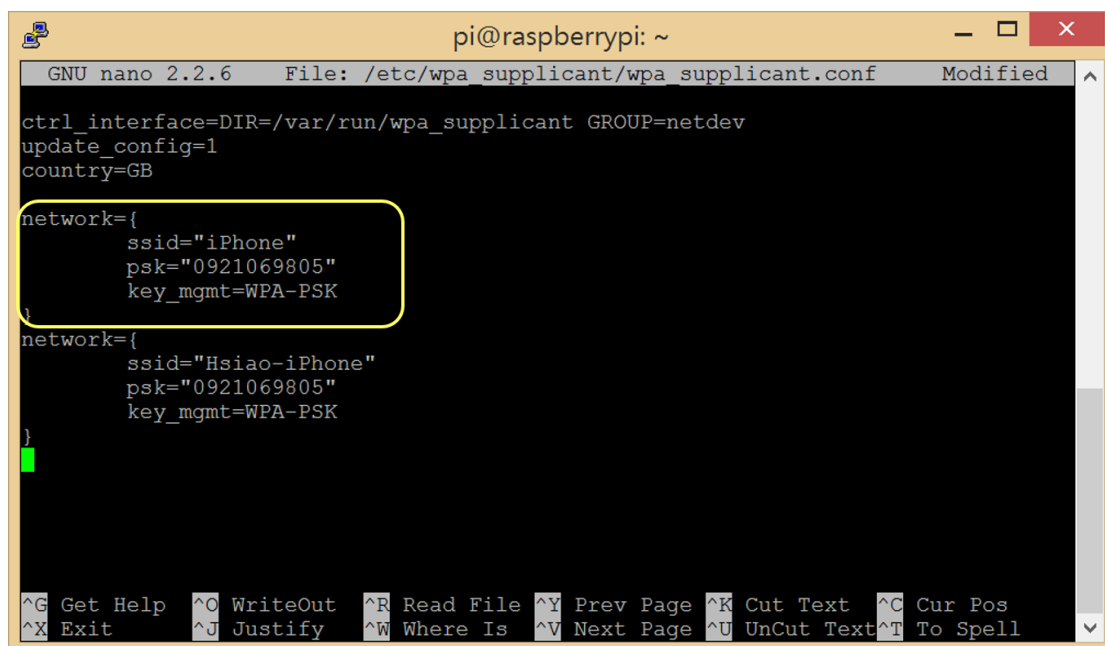
```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

新增如下指令

```
network={  
    ssid="PLEASE_KEY_YOUR_SSID"  
    psk="PLEASE_KEY_YOUR_PASSWORD"  
    key_mgmt=WPA-PSK  
}
```

按下 Ctrl + O 儲存

並按下 Enter



```
pi@raspberrypi: ~  
GNU nano 2.2.6 File: /etc/wpa_supplicant/wpa_supplicant.conf Modified  
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev  
update_config=1  
country=GB  
network={  
    ssid="iPhone"  
    psk="0921069805"  
    key_mgmt=WPA-PSK  
}  
network={  
    ssid="Hsiao-iPhone"  
    psk="0921069805"  
    key_mgmt=WPA-PSK  
}  
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos  
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

## Step2. 為 Raspberry Pi 3 安裝 Python

(PS: 須注意執行任何指令若遇到權限問題 可在指令最前面加上 **sudo** )

1. 更新系統

```
$ sudo apt-get update
```

```
$ sudo apt-get upgrade
```

2. 安裝 Python

```
$ sudo apt-get install python2.7
```

3. 查詢 Python 版本

```
$python --version
```

4. 安裝相關套件：

```
$ sudo apt-get update
```

```
$ sudo apt-get install build-essential python-dev
```

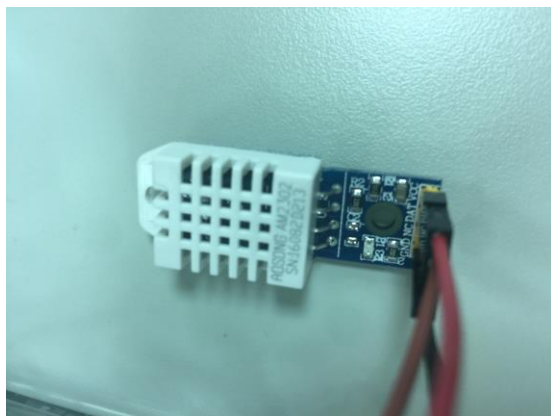
5. 下載 pm2.5 範例專案

```
git clone https://github.com/Thomas-Tsai/pms3003-g3.git
```

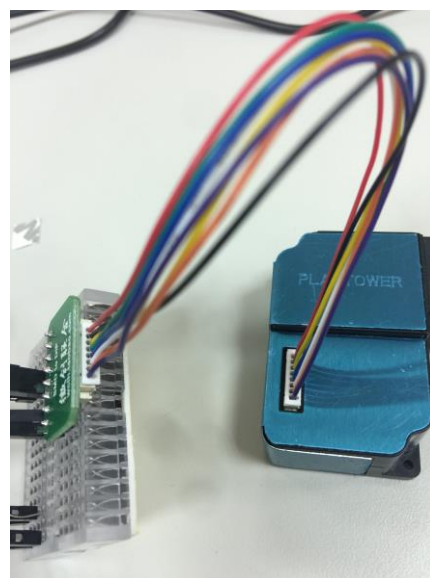
```
sudo apt-get install python-pip python-serial
```

```
python g3.py
```

## Step3. 溫溼度感測器(DHT22)與 PM2.5 感測器接線



溫溼度



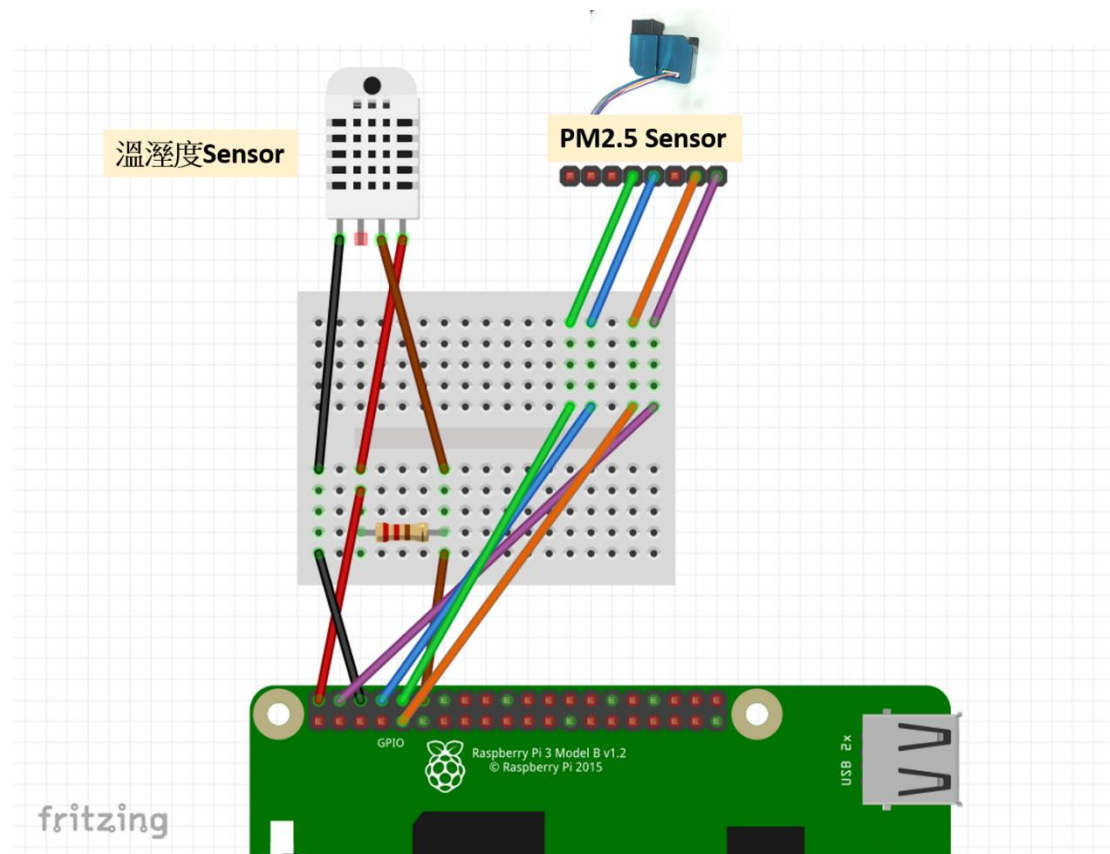
PM2.5

PM2.5 感測器之綠色板子背面有對應的 PIN

請先將 PM2.5 參考下表接成右上圖的樣子

|             |       |                     |
|-------------|-------|---------------------|
| PIN1        | VCC   | 電源正 5V              |
| PIN2        | GND   | 電源負                 |
| PIN3        | SET   | 設置管腳 /TTL 電平@3.3V   |
| PIN4        | RXD   | 串口接收管腳 /TTL 電平@3.3V |
| PIN5        | TXD   | 串口發送管腳 /TTL 電平@3.3V |
| PIN6        | RESET | 模塊復位信號 /TTL 電平@3.3V |
| PIN7 與 PIN8 | NC    | 懸空                  |

請耐心依照下圖接線



## Step4. 溫溼度感測器(DHT22)功能測試

以下為測試溫溼度感測器是否能運作的指令

開啟 Terminal 終端機並輸入

```
git clone https://github.com/adafruit/Adafruit_Python_DHT.git
```

```
cd Adafruit_Python_DHT
```

```
sudo python setup.py install
```

```
cd examples
```

```
sudo ./AdafruitDHT.py 22 18
```

(PS: 其中參數 22 代表 DHT22，18 代表把它接到 Raspberry Pi 的 GPIO 18)

```
pi@raspberrypi:~/Adafruit_Python_DHT $ ls
Adafruit_DHT      build  examples  ez_setup.pyc  README.md  source
Adafruit_DHT.egg-info  dist  ez_setup.py  LICENSE      setup.py
pi@raspberrypi:~/Adafruit_Python_DHT $ cd examples/
pi@raspberrypi:~/Adafruit_Python_DHT/examples $ sudo ./AdafruitDHT.py 22 18
Temp=23.8* Humidity=58.1%
pi@raspberrypi:~/Adafruit_Python_DHT/examples $
```

## Step5. PM2.5 感測器(PMS 3003)功能測試

修改 config.txt

```
sudo nano /boot/config.txt
```

並在最下面一行增加

```
enable_uart=1
```

重開機

```
sudo reboot
```

確認有出現 ttyS0 (截圖如下)

```
ls -l /dev
```

```
drwxr-xr-x 2 root root          60 Jan  1 1970 raw
crw-rw-r-- 1 root root    10, 58 Feb  8 09:40 rfkill
lrwxrwxrwx 1 root root          5 Feb  8 09:40 serial0 -> ttyS0
lrwxrwxrwx 1 root root          7 Feb  8 09:40 serial1 -> ttyAMA0
drwxrwxrwt 2 root root          40 Feb  8 09:40 shm
drwxr-xr-x 2 root root       140 Feb  8 09:40 snd
lrwxrwxrwx 1 root root          15 Jan  1 1970 stderr -> /proc/self/fd/2
lrwxrwxrwx 1 root root          15 Jan  1 1970 stdin  -> /proc/self/fd/0
lrwxrwxrwx 1 root root          15 Jan  1 1970 stdout -> /proc/self/fd/1
```

```
cd pms3003-g3
```

```
sudo nano g3.py
```

找到 `pmdata=air.read("/dev/ttyAMA0")` 修改成 `pmdata=air.read("/dev/ttyS0")`

執行 g3.py (先 cd 至含有 g3.py 的目錄下)



```
sudo python g3.py
```

```
pi@raspberrypi:~ $ ls
Adafruit_Python_DHT  Downloads      Pictures      python_games
Desktop             iot-python-example  pms3003-g3  Templates
Documents           Music          Public       Videos
pi@raspberrypi:~ $ cd pms3003-g3/
pi@raspberrypi:~/pms3003-g3 $ ls
g3.py  monitor.py  README.md
pi@raspberrypi:~/pms3003-g3 $ sudo python g3.py
[8, 15, 10, 8, 15, 10]
pi@raspberrypi:~/pms3003-g3 $
```

## Step6. 下載並測試 IoT 實際範例

```
sudo pip install paho-mqtt
```

```
git clone https://github.com/yuzheng/iot-python-example
```

```
cd iot-python-example/
```

```
cd g3-dht/
```

```
nano iot-lass.py
```

修改 21、23 與 24 行

```
Iothost="iot.epa.gov.tw"
```

```
iotkey="請參考下圖，輸 IoT 智慧聯網大平台裝置金鑰"
```

```
device="請參考下圖，輸入 IoT 智慧聯網大平台裝置識別碼(數字)"
```

device 資訊於 IoT 智慧聯網大平台 > 專案管理 > 設備列表中可獲知，如下



智慧聯網大平台  
SMART PLATFORM

開發者中心 專案管理 應用服務 產品登錄 帳號資訊 繁體中文 登出

首頁 / 專案管理 / 設備管理

### 設備管理

專案名稱 Raspberry Pi Demo

+ 增加感測器 + 增加設備 → 返回

快速搜尋:

| 設備編號      | 設備名稱          | 設備描述               | 設備類型    | 功能  |
|-----------|---------------|--------------------|---------|---|
| 285787084 | raspberrypi 3 | raspberrypi 3 demo | general |      |
| 290959349 | test          | test               | general |      |

device (裝置識別碼)  
顯示第 1 至 2 項結果

« 1 »

設備的 IoT Key 可從設備編輯得知，如下

## 設備管理

專案名稱 Raspberry Pi Demo

+ 增加感測器

+ 增加設備

→ 返回

快速搜尋:

| 設備編號      | 設備名稱          | 設備描述               | 設備類型    | 功能     |
|-----------|---------------|--------------------|---------|--------|
| 285787084 | rasberry pi 3 | rasberry pi 3 demo | general | 點選設備編輯 |
| 290959349 | test          | test               | general |        |

顯示第 1 至 2 項結果

« 1 »

設備管理頁面中，最底下的設備金鑰即代表 IoT Key，如下

設備管理

基本資料

擴充屬性資訊

設備名稱

rasberry pi 3

設備描述

rasberry pi 3 demo

設備類型

general

緯度

緯度 ...

經度

經度 ...

URI

Y3CHFKZ50SU91HP1

設備金鑰

DKF1PFUS4FY2499GEZ

IoT Key

取消

下一頁



```
sudo python iot-lass.py
```

成功時在命令提示字元應可看到數值  
且平台專案底下也會同步更新!

為了確保離開 ssh 後仍可以繼續執行，請輸入底下指令

```
nohup sudo python iot-lass.py &
```

並可透過底下指令，確認服務是否執行

```
ps aux
```