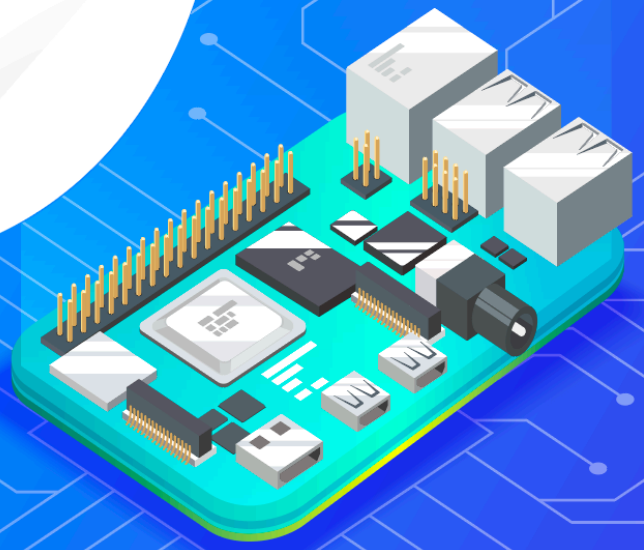
# AIOT智慧物聯網學習馬拉松

▸ 作業解答篇



陪跑專家：李盛安

# 作業1


**問題：**

- 實際安裝gpiozero，先安裝RPi.GPIO，再安裝pigpio，觀察安裝過程系統顯示的訊息。比較直接啟動raspi-config的interfacing選項，透過啟動Remote GPIO，直接安裝GPIOZero。

# 作業1


- gpiozero是新型態的GPIO開發架構，現在可以很簡單的透過raspi-config直接啟動Remote GPIO選項來安裝，為了讓同學了解架構間的相依性與歷史脈絡，同學可以練習自己裝裝看各個套件，之後在使用GPIOZero的時候會更清楚背後的觀念。

# 作業1

CUPOY

- 安裝RPI.GPIO的方法，使用apt install python3-rpi.gpio

```
root@raspberrypi:/home/shengan# apt install python3-rpi.gpio
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libnginx-mod-http-auth-pam libnginx-mod-http-dav-ext libnginx-mod-http-echo libnginx-mod-http-geoip libnginx-mod-http-image-filter libnginx-mod-http-subs-filter
  libnginx-mod-http-upstream-fair libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream nginx-common
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  rpi.gpio-common
The following NEW packages will be installed:
  rpi.gpio-common
The following packages will be upgraded:
  python3-rpi.gpio
1 upgraded, 1 newly installed, 0 to remove and 286 not upgraded.
Need to get 26.4 kB of archives.
After this operation, 11.3 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://archive.raspberrypi.org/debian buster/main armhf rpi.gpio-common armhf 0.7.0-0.1~bpo10+1 [5,776 B]
Get:2 http://archive.raspberrypi.org/debian buster/main armhf python3-rpi.gpio armhf 0.7.0-0.1~bpo10+1 [20.7 kB]
Fetched 26.4 kB in 1s (23.2 kB/s)
Reading changelogs... Done
Selecting previously unselected package rpi.gpio-common:armhf.
(Reading database ... 155013 files and directories currently installed.)
Preparing to unpack .../rpi.gpio-common_0.7.0-0.1~bpo10+1_armhf.deb ...
Unpacking rpi.gpio-common:armhf (0.7.0-0.1~bpo10+1) ...
Preparing to unpack .../python3-rpi.gpio_0.7.0-0.1~bpo10+1_armhf.deb ...
Unpacking python3-rpi.gpio (0.7.0-0.1~bpo10+1) over (0.7.0~buster-1) ...
Setting up rpi.gpio-common:armhf (0.7.0-0.1~bpo10+1) ...
Setting up python3-rpi.gpio (0.7.0-0.1~bpo10+1) ...
root@raspberrypi:/home/shengan#
```

# 作業1

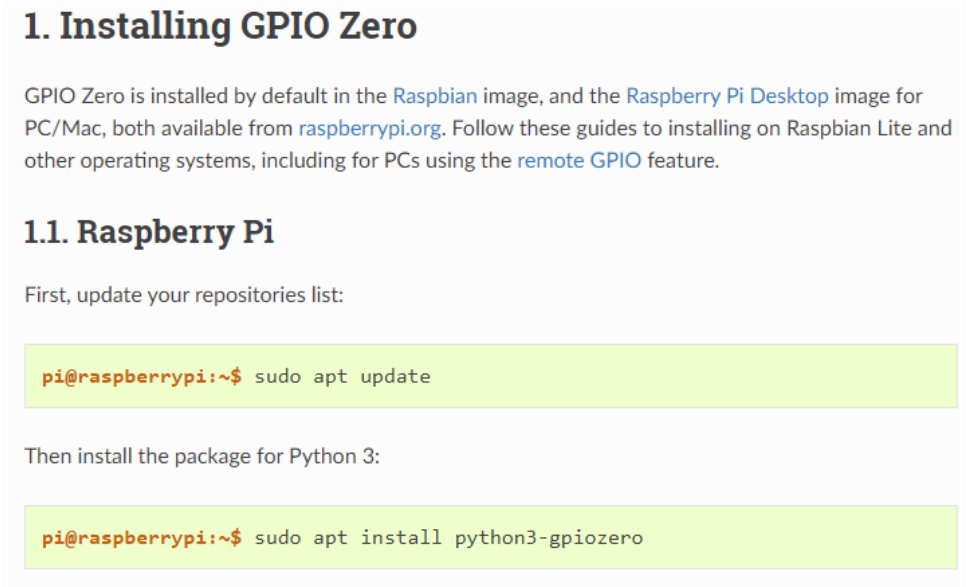

- 安裝pigpio的方法，使用apt install pigpio python3-pigpio

```
root@raspberrypi:/home/shengan# apt install pigpio python3-pigpio
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libnginx-mod-http-auth-pam libnginx-mod-http-dav-ext libnginx-mod-http-echo libnginx-mod-http-geoip
  libnginx-mod-http-image-filter libnginx-mod-http-subs-filter libnginx-mod-http-upstream-fair libnginx-mod-http-xslt-filter
  libnginx-mod-mail libnginx-mod-stream nginx-common
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  pigpio python3-pigpio
0 upgraded, 2 newly installed, 0 to remove and 286 not upgraded.
Need to get 38.6 kB of archives.
After this operation, 185 kB of additional disk space will be used.
Get:1 http://archive.raspberrypi.org/debian buster/main armhf pigpio armhf 1.71-0~rpt1 [3,320 B]
Get:2 http://archive.raspberrypi.org/debian buster/main armhf python3-pigpio all 1.71-0~rpt1 [35.3 kB]
Fetched 38.6 kB in 1s (33.4 kB/s)
Selecting previously unselected package pigpio.
(Reading database ... 154965 files and directories currently installed.)
Preparing to unpack .../pigpio_1.71-0~rpt1_armhf.deb ...
Unpacking pigpio (1.71-0~rpt1) ...
Selecting previously unselected package python3-pigpio.
Preparing to unpack .../python3-pigpio_1.71-0~rpt1_all.deb ...
Unpacking python3-pigpio (1.71-0~rpt1) ...
Setting up python3-pigpio (1.71-0~rpt1) ...
Setting up pigpio (1.71-0~rpt1) ...
```

# 作業1

CUPOY

- 如果想要按照官方網站的文件來安裝GPIOZero
- 文件在這裡https://gpiozero.readthedocs.io/en/stable/installing.html



## 1. Installing GPIO Zero

GPIO Zero is installed by default in the Raspbian image, and the Raspberry Pi Desktop image for PC/Mac, both available from raspberrypi.org. Follow these guides to installing on Raspbian Lite and other operating systems, including for PCs using the remote GPIO feature.

### 1.1. Raspberry Pi

First, update your repositories list:

```
pi@raspberrypi:~$ sudo apt update
```

Then install the package for Python 3:

```
pi@raspberrypi:~$ sudo apt install python3-gpiozero
```

- 現在已經全部都支援python3。

# 作業2

CUPOY

**問題：**

- 實際練習GPIOZero控制LED，確定單獨的GPIO控制LED亮跟暗交替閃爍完成，並且PWM控制LED明亮的完成後，嘗試依序改變led.value的值，分別設定0.1, 0.3, 0.5, 0.7觀察差異。

# 作業2


- 安裝完GPIOZero之後，最重要的是了解它的操作邏輯，直接學會PWM形式的操控GPIO接腳作業方式，是最有效的學習方式，因此同學記得一定要觀察PI如何透過GPIOZero控制GPIO的PWM。

# 作業2

CUPOY

- 本單元內教材的範例程式碼如下, 可以按照範例程式碼改成作業2要求的段落，原始範例程式碼如下：

```
from gpiozero import PWMLED
from time import sleep

led = PWMLED(17)

while True:
    led.value = 0  # off
    sleep(1)
    led.value = 0.5  # half brightness
    sleep(1)
    led.value = 1  # full brightness
    sleep(1)
```

# 作業2

CUPOY

- 按照上面的程式碼片段，可以很容易的加上各個ＰＷＭ參數的程式碼，執行並且觀察結果。

```
from  gpiozero  import  PWMLED
from  time  import  sleep

#  sleepInterval  每個指令之間的等待時間
sleepInterval  =  1
#  定義使用GPIO  17作為GPIO的接腳
led  =  PWMLED(17)
while  True:
    led.value  =  0.1
    sleep(sleepInterval)
    led.value  =  0.3
    sleep(sleepInterval)
    led.value  =  0.5
    sleep(sleepInterval)
    led.value  =  0.7
    sleep(sleepInterval)
```

# 作業3

CUPOY

**問題：**

- 實際練習GPIOZero透過Button控制LED，在按鈕的過程中，觀察實際按下按鈕的次數，LED點亮的次數，是否一致。練習修改程式，讓按鈕按下是全亮，按鈕放開後是30%的亮度

# 作業3


- 本題的目的是練習整合教材內的兩個範例，會做了以後就會發現其實很多功能，可以透過許多網路上的小片段程式碼組合而成，進而之後可以練習有能力透過github上面的程式片段，了解如何與思考的過程整合的能力。

# 作業3

CUPOY

- 首先作業2已經學會如何使用PWM控制LED的亮度，學會將只有高電位與低電位狀態的GPIO接腳，達成類似類比訊號的介於0-1之間的輸出能力，例如本題的範例，如何輸出0.3。

- 接下來在了解事件驅動的概念，按鈕程式碼已經被GPIOZero架構成事件驅動的架構，只要更改lightOn與lightOff的程式碼內容就可以達到對應的動作。

```
from gpiozero import LED, Button
from singal import pause

led = LED(17)

def lightOn():
    led.value = 1
    print("Light value is 1")

def lightOff():
    led.value = 0.3
    print("Light value is 0.3")

button = Button(2)
button.when_pressed = lightOn
button.when_released = lightOff

pause()
```