Part

在 Raspberry Pi 上使用 Python 來開發

本書第 I 篇要談的是物聯網的「網」。

我們會從如何正確設定 Python 開發環境開始，接著就會透過 Pyhthon 來實際操作各種網路通訊技術來製作物聯網服務與程式。我們會製作簡易網頁介面來操作所要學習的技術與範例。

不過啦，我相信你在閱讀本書時一定迫不及待，想要趕快學會如何製作與操作各種電子元件，對吧？我自己就是這樣！所以第 2 章就會告訴你如何從頭製作簡易的物聯網專案（包含電路與所有必要內容）。這樣到了後續章節就有很多範例可以參考了（也會有很多東西可以改！）

現在就開始吧！

本篇包含以下章節：

Chapter 01

設定開發環境

Python 程式設計中一項重要但也經常被忽視的一環，就是如何正確地設定和維護 Python 專案與其執行階段環境。經常被忽視的原因是它之於整個 Python 生態系來說並非必要的步驟。儘管這件事對於還在打基礎時可能還不錯，然而對於需要維護多份獨立函式庫和相依套件來確保專案不會彼此干擾的複雜專案時，這很快就會變成大問題了；更糟糕的是可能會損壞作業系統的一些工具與公用程式，後續都會談到。

因此，在進入後面章節的物聯網範例之前，一個必要的重要步驟就是好好帶你做一遍設定 Python 專案與其執行階段環境所需的步驟。

本章主題如下：

- 了解你的 Python 安裝
- 設定 Python 虛擬環境
- 用 `pip` 安裝 Python GPIO 套件
- 執行 Python 腳本的替代方法
- Raspberry Pi GPIO 介面設定

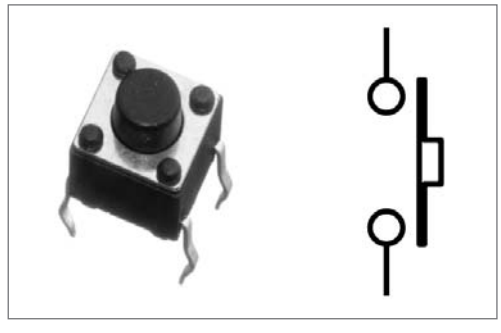
- E30 與 F30 彼此未連接（它們在不同的區塊）。
- 左側電源軌上的第三個 + 孔（從麵包板上往下算）與左側電源軌上的最後一個 + 孔（它們在同一個直行）是彼此連接的。
- 左側電源軌上的第三個 + 孔（從麵包板頂部）與右側電源軌上的第三個 + 孔彼此未連接（它們位於不同的電源軌上）。

本節開頭有提到，所有麵包板基本上是一樣的，但是有一個小例外就是電源軌。有些全尺寸的麵包板可把電源軌拆成兩個獨立的直排（因此，在同一軌中的孔並非全部相連）。一下子可能看不太出來電源軌是有間隔的，因此需要根據你實際採用的麵包板型號來看看。之所以提到這一點是為了防止你在使用全尺寸麵包板的電源軌時碰到接線的問題。

介紹了麵包板，也了解各個孔彼此如何連接之後，現在可以把元件和電線插入麵包板來建立第一個電路。我們就從按鈕開始。

2.2.2 安裝按鈕與接線

在此使用簡單的開關按鈕，也稱為單刀單擲（Single Pole, Single Throw / SPST）瞬時型開關，如圖 2-2：

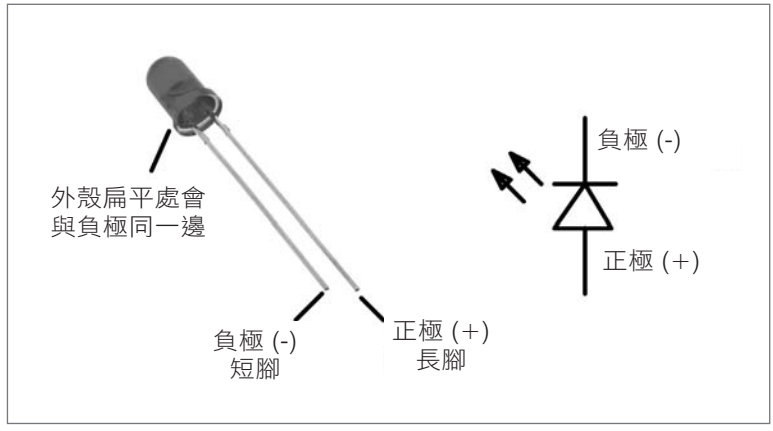


▲ 圖 2-2 按鈕與其電路符號

2.2.3 安裝 LED 與接線

LED 是一種體積小巧但亮度很高的照明裝置。它由微小的晶體構成，通電之後會發出特定的顏色。

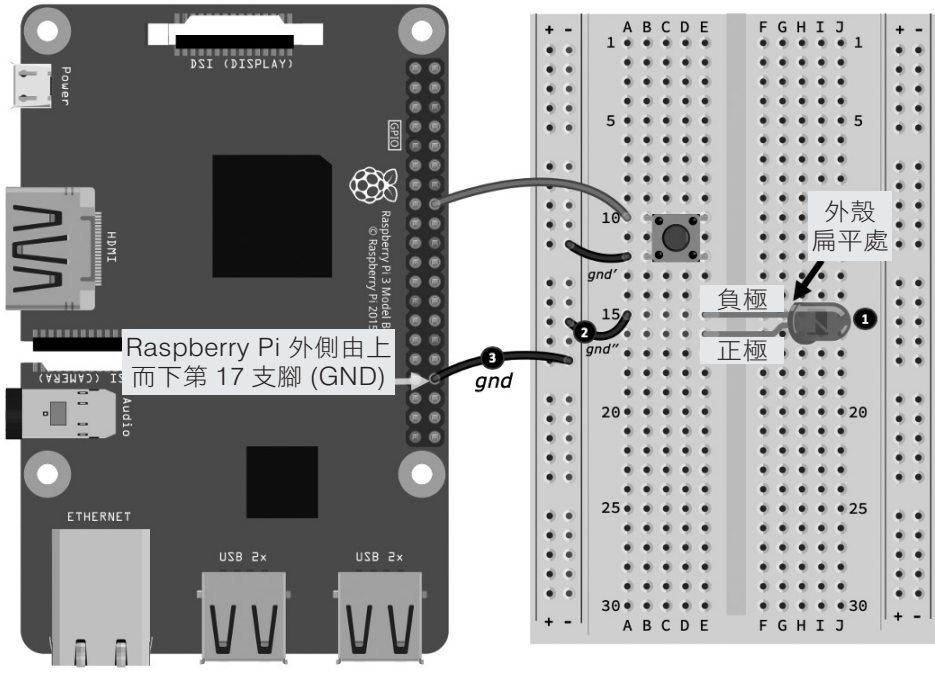
圖 2-4 是常見的 LED，左側為實體照片，右側為其電路符號：



▲ 圖 2-4 LED 與其電路符號

LED 一定要以正確的方式連接，否則無法運作。仔細看一下 LED，你會發現到 LED 外殼的有一側是平的。在這一側的腳為負極，要接到電源負極（接地）。負極腳位也是 LED 兩支腳中較短的。另一支腳稱為正極，並要接到電源正極。請看一下 LED 的電路符號，你會發現到 LED 負極那一側有一條橫過三角形頂點的線—如果你能把這條線看成一個超大負號，這有助於你記住符號的哪一邊是負極。

圖 2-5 是在麵包板上插上 LED 的結果。執行後續步驟時，請參考此圖：



▲ 圖 2-5 將 LED 接上麵包板

以下是把 LED 接到麵包板與 Raspberry Pi 的步驟，步驟編號對應於圖 2-5 中的黑色圓圈號碼：

01 如上圖，把 LED 插上麵包板，特別注意 LED 不要裝錯。圖中負極腳位於 E15 孔中，正極腳則是在 E16 孔。

Tips

有時可能需要折彎 LED 的腳會更好連接。在安裝 LED 時，注意兩支腳不要碰到！如果碰到的話，會導致所謂的電氣短路，電路中的 LED 就會無法運作或損壞。

02 接下來，使用一條線（上圖標示為 gnd" 的那條線）將 LED 的負極腳接到按鈕所接的同一個共地軌。這條 gnd" 的一端是接到 A15 孔，另一端則是接到麵包板左側的負極（-）共地軌。

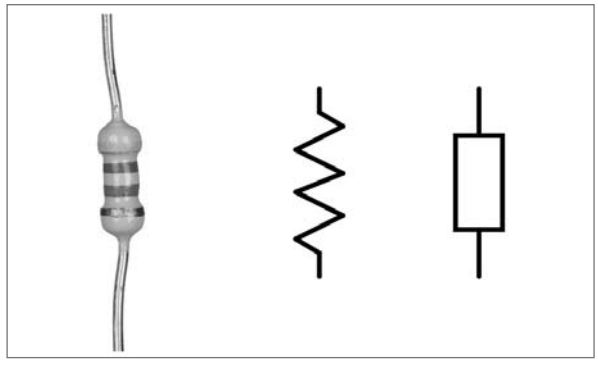
03 最後，使用另一跳線（上圖中的 gnd），從共地軌接到 Raspberry Pi 外側由上往下數的第 17 支腳位，這是 Raspberry Pi 的其中一支接地（GND）腳位。

做得好！ LED 接好了。接下來再加入電阻，電路就完成了。

2.2.4 安裝電阻與接線

電阻是用來限制（也就是阻擋）電流流動以及切分電壓的常見電子元件。

圖 2-6 左側是電阻的實體照片，右側則是兩個電路符號。兩個電路符號沒有實際差別，只是不同的文件慣例而已，而且你會發現繪圖者只會選用一種符號。本書將使用鋸齒符號。



▲ 圖 2-6 電阻與其電路符號

電阻有許多形狀、大小及顏色。一般來說，其實際形狀及大小與其物理性質與功能有關，但至少就性質來說，外殼的顏色通常不太重要。不過，電阻上的色環就很重要了，因為它們是用來判讀電阻的電阻值。值得一提的是，小型通用電阻（就是後續要用到的）會用色環來判讀其電阻值，然而高功率的較大型電阻則會把電阻值印在外殼上。

▲ 圖 3-2 Web Socket 客戶端網頁

確認可否使用網頁上的滑桿來改變 LED 的亮度。

Tips

開啟另一個網路瀏覽器並開啟 `http://localhost:5000`，現在打開兩個網頁，拉動滑桿，你可以看到這兩個網頁即時同步了！相信你很快就知道 Web Socket 相較於 RESTful API 的獨特優點。

03 在網頁上找到 **Connected to server: Yes** 這一行，然後做以下動作：

- 在終端機中按下 Ctrl + C 來終止伺服器，你會發現這行變成了 **Connected to server: No**
- 再度重啟伺服器，這行訊息又會變回 **Connected to server: Yes**

這展示了 Web Sockets 的雙向性質。後續看到其 JavaScript 內容時就會知道，我們將看到如何在網頁上實作這件事，但首先要來看看構成 Web Socket 伺服器的 Python 程式碼。

3.5.2 伺服器程式碼

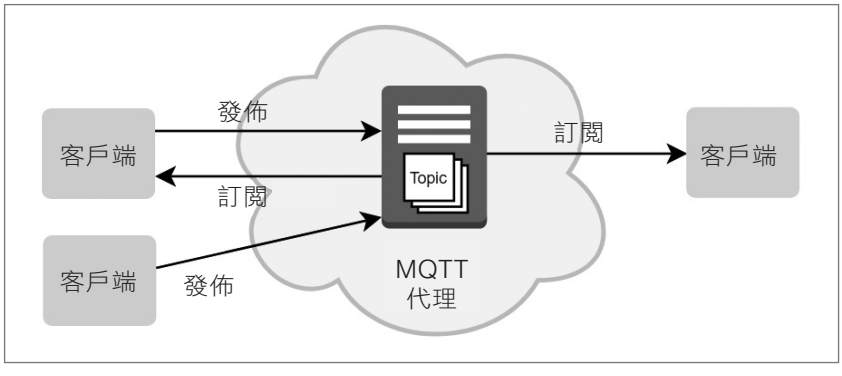
本節要介紹 Python 伺服器的原始程式碼並討論一些重要的地方，在此同樣會跳過在先前章節所提過的程式碼與概念。首先來看看匯入了哪些東西。

4.3 以範例來學習 MQTT

MQTT 是以代理（broker）為基礎的訊息通訊協定，可以發佈與訂閱訊息（常改寫為 pub/sub），而所謂 MQTT 代理（如同上一節所安裝的 Mosquitto MQTT 代理）就是一台實作 MQTT 通訊協定的伺服器。利用基於 MQTT 的架構，你的應用大致上就能把所有複雜的訊息傳遞處理和路由邏輯都交給代理去負責，讓應用能專注於自身的主要功能。

MQTT 客戶端（例如之後要用到的 Python 程式與命令列工具）會與代理建立訂閱關係，並訂閱他們感興趣的訊息主題。客戶端會發佈訊息給某個主題，而代理負責所有的訊息路由與運送保證。任何客戶端都可作為訂閱端、發佈方，或兩者兼具。

圖 4-2 是一個包含了泵浦、水箱與控制器的 MQTT 的簡易概念系統。



▲ 圖 4-2 MQTT 範例

以下是系統元件的高階描述：

- 把水位感測器 MQTT 客戶端視為連接到水箱中之水位感測器的軟體。此客戶端在 MQTT 範例中是作為發佈方。它會定期發送（也就是發佈）關於水箱有多滿的訊息給 MQTT 代理。
- 把泵浦 MQTT 客戶端視為能夠啟動或關閉水泵浦的軟體驅動器。此客戶端在範例中同時作為發佈方與訂閱端。

以下為給你參考的一些範例：

表 4-3 發佈方與訂閱端 QoS 範例

發佈方發佈訊息之等級	訂閱端訂閱訊息之等級	訂閱端得到什麼
QoS 2	QoS 0	傳輸遵循 QoS 0 的訊息（訂閱端得到訊息 0 或 1 次）
QoS 2	QoS 2	傳輸遵循 QoS 2 的訊息（訂閱端只得到訊息 1 次）
QoS 0	QoS 1	傳輸遵循 QoS 0 的訊息（訂閱端得到訊息 0 或 1 次）
QoS 1	QoS 2	傳輸遵循 QoS 1 的訊息（訂閱端得到訊息 1 或更多次）
QoS 2	QoS 1	傳輸遵循 QoS 1 的訊息（訂閱端得到訊息 1 或更多次）

從這些範例中得出的結論是，實務上在設計或整合物聯網解決方案時，你需要留意在主題兩端之發佈方與訂閱端採用了何種等級的 QoS，而非單獨從某一端來解釋。

請根據以下步驟來操作 QoS 情境，並看看客戶端與代理的即時互動：

01 在終端機中執行以下指令來啟動訂閱端：

```
# Terminal 1 (Subscriber)
$ mosquitto_sub -d -v -q 2 -h localhost -t 'pyiot'
```

02 在第二終端機中執行以下指令來發佈訊息：

```
# Terminal 2 (Publisher)
$ mosquitto_pub -d -q 1 -h localhost -t 'pyiot' -m 'hello!'
```

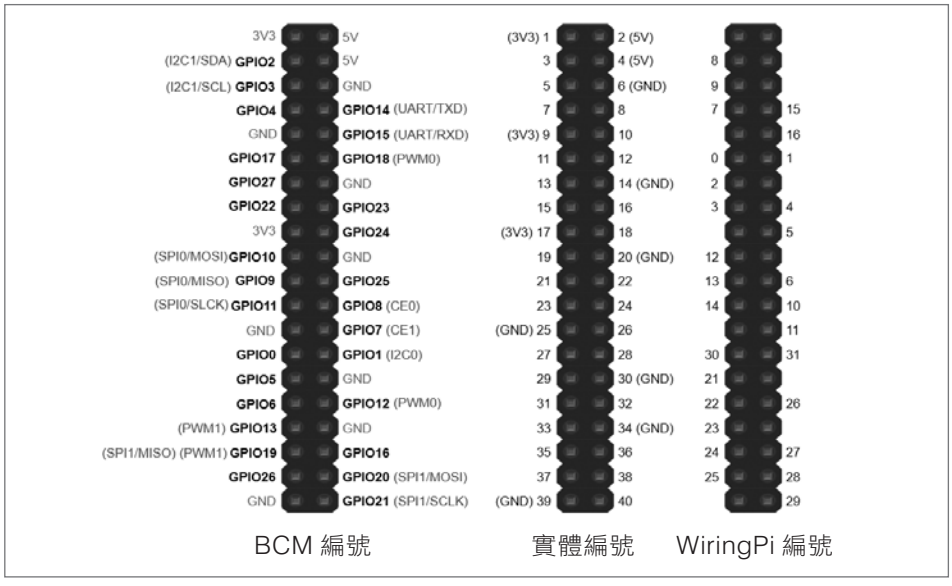
除了上述軟體之外，本章範例還需要以下實體電子元件：

- 5 mm 紅光 LED，1 個
- 200 Ω 電阻，一個。色環為紅、黑、棕，然後是金色或銀色
- ADS1115 ADC 轉接模組，1 個（例如，`https://www.adafruit.com/product/1085`）
- 10 kΩ 電位計，2 個（10KΩ - 100KΩ 之間的任何數值都適用）
- 麵包板，1 個
- 公對母和公對公跨接線，1 批（也稱為杜邦線）

5.2 了解 Raspberry Pi 腳位編號

注意到 Raspberry Pi 那些伸出的腳位了嗎？自第 2 章開始，我們已知道如何透過像是 GPIO Pin 23 的說法來引用這些腳位，但這是什麼意思？現在是詳細說明的時候了。

有三種常用方式來引用 Raspberry Pi 的 GPIO 腳位，如圖 5-1 所示：

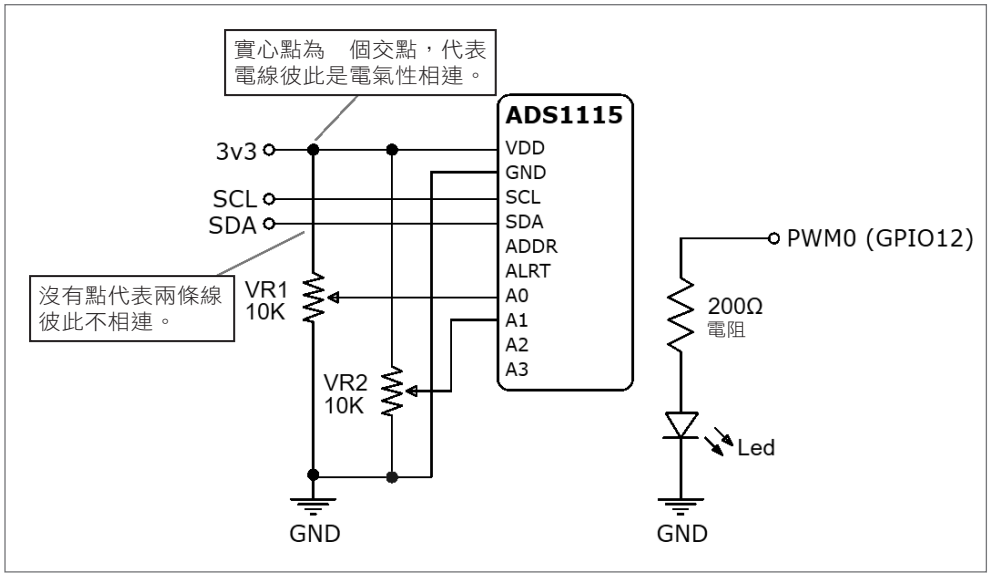


▲ 圖 5-1 GPIO 腳位編號方案

做得好！電路完成了。別忘了，以下圖 5-7 是對應本範例麵包板電路的電路圖。

提醒一下，我們在第 2 章有解說過如何閱讀電路圖喔！

建議你參照麵包板佈線去對照這個示意圖，了解圖中線條與標籤如何對應到麵包板上的元件和配線。花時間了解電路圖和麵包板電路之間的對應關係，將有助於你直接從電路圖去製作麵包板電路的能力：



▲ 圖 5-7　ADC 電路示意圖

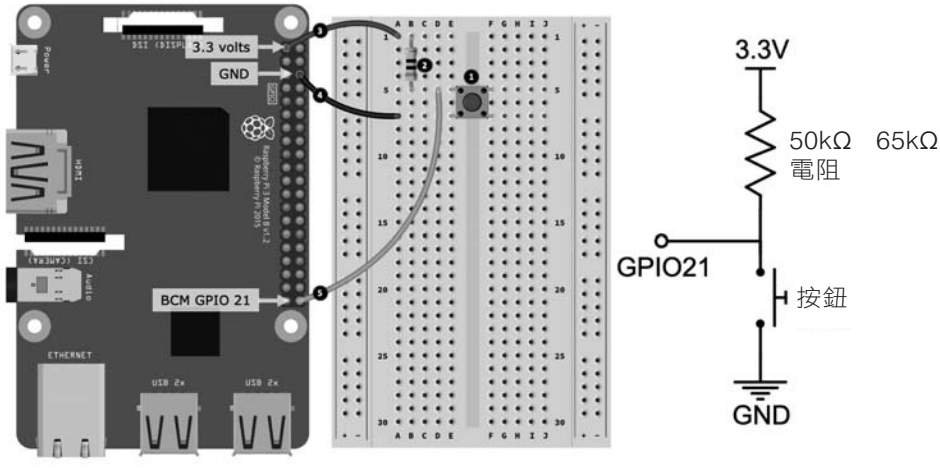
電路完成後，檢查一下 Raspberry Pi 是否可看到 ADS1115。

5.5.2 確認 ADS1115 已接上 Raspberry Pi

主機（就是 Raspberry Pi）是透過一個唯一的位址來識別 I2C 裝置，ADS1115 的預設位址為 0x48。由於 I2C 裝置已被定址，因此多個裝置可共享 Raspberry Pi 的同一個 I2C 通道（腳位）。

◉ 使用電阻來解決腳位浮動

如果在電路中多加一個電阻，如下圖，代表加入所謂的上拉電阻。「拉」代表連接，而「上拉」是指將 GPIO 21 腳位的電位拉上去（連接到正電壓）到 3.3V：



▲ 圖 6-6 帶有上拉電阻的按鈕電路

請根據以下步驟把各元件接到麵包板，步驟編號對應於圖 6-6 中的黑色圓圈號碼：

01 把按鈕接上麵包板。

02 將電阻（50kΩ - 65kΩ 之間）接上麵包板。電阻一端與按鈕的左上腳（如圖中的 B5 孔）接在同一橫列。電阻另一端則接到任一空白橫列。

03 將電阻的另一端接到 Raspberry Pi 的 3.3V 腳位。

04 將按鈕的左下腳接到 Raspberry Pi 上的 GND 腳位。

05 最後用一條電線，從按鈕的左上腳與電阻下方腳位（如圖中的 D5 孔）所共享的同一橫列接到 Raspberry Pi 的 GPIO 21 腳位。