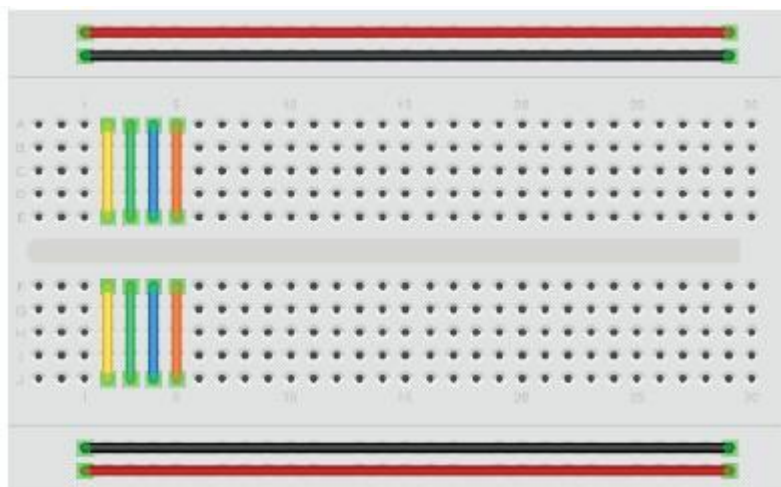
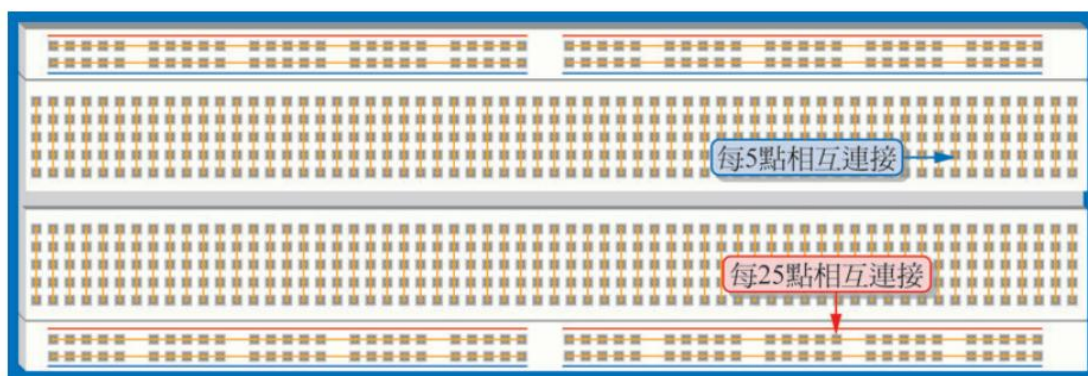


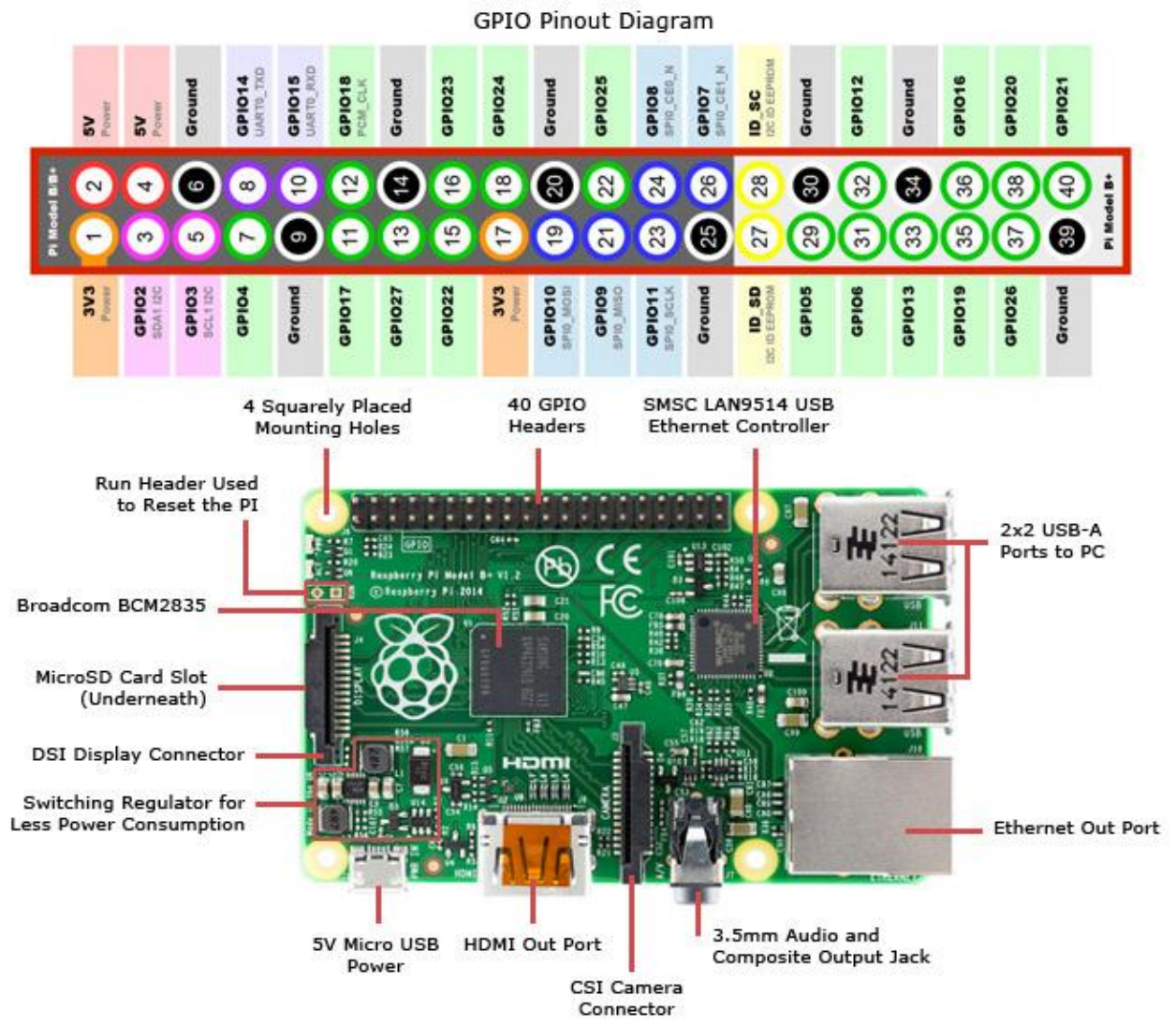
## 第四章 透過 GPIO 控制 LED 燈


### ➤ 認識麵包板認識麵包板

麵包板是不需要經由焊接過程，就可以將電路中所使用的電子元件加以連接，進而進行電路特性的量測，以驗證電路的功能是否正常的實驗室工具。



### ➤ raspberry pinout





The diagram shows a Raspberry Pi B+ board with the P1 GPIO header. Blue arrows indicate the locations of pins 1, 2, 39, and 40. A label 'GPIO pins' points to the header area.

| Raspberry Pi B<br>Rev 2 P1 GPIO Header |         |    |        | Raspberry Pi B+<br>B+ J8 GPIO Header |         |    |        |
|--|---------|----|--------|--------------------------------------|---------|----|--------|
|  | Pin No. |    |        |                                      | Pin No. |    |        |
| 3.3V                                   | 1       | 2  | 5V     | 3.3V                                 | 1       | 2  | 5V     |
| GPIO2                                  | 3       | 4  | 5V     | GPIO2                                | 3       | 4  | 5V     |
| GPIO3                                  | 5       | 6  | GND    | GPIO3                                | 5       | 6  | GND    |
| GPIO4                                  | 7       | 8  | GPIO14 | GPIO4                                | 7       | 8  | GPIO14 |
| GND                                    | 9       | 10 | GPIO15 | GND                                  | 9       | 10 | GPIO15 |
| GPIO17                                 | 11      | 12 | GPIO18 | GPIO17                               | 11      | 12 | GPIO18 |
| GPIO27                                 | 13      | 14 | GND    | GPIO27                               | 13      | 14 | GND    |
| GPIO22                                 | 15      | 16 | GPIO23 | GPIO22                               | 15      | 16 | GPIO23 |
| 3.3V                                   | 17      | 18 | GPIO24 | 3.3V                                 | 17      | 18 | GPIO24 |
| GPIO10                                 | 19      | 20 | GND    | GPIO10                               | 19      | 20 | GND    |
| GPIO9                                  | 21      | 22 | GPIO25 | GPIO9                                | 21      | 22 | GPIO25 |
| GPIO11                                 | 23      | 24 | GPIO8  | GPIO11                               | 23      | 24 | GPIO8  |
| GND                                    | 25      | 26 | GPIO7  | GND                                  | 25      | 26 | GPIO7  |
|  |         |    |        | DNC                                  | 27      | 28 | DNC    |
|  |         |    |        | GPIO5                                | 29      | 30 | GND    |
|  |         |    |        | GPIO6                                | 31      | 32 | GPIO12 |
|  |         |    |        | GPIO13                               | 33      | 34 | GND    |
|  |         |    |        | GPIO19                               | 35      | 36 | GPIO16 |
|  |         |    |        | GPIO26                               | 37      | 38 | GPIO20 |
|  |         |    |        | GND                                  | 39      | 40 | GPIO21 |

| Key              |      |
|------------------|------|
| Power +          | UART |
| GND              | SPI  |
| I <sup>2</sup> C | GPIO |

### ● I<sup>2</sup>C (Inter-Integrated Circuit)

I<sup>2</sup>C (Inter-Integrated Circuit) 字面上的意思是積體電路之間，它其實是 I<sup>2</sup>C Bus 簡稱，所以中文應該叫積體電路匯流排，它是一種串列通訊匯流排，使用內送流量備援容錯機制從架構，由飛利浦公司在 1980 年代為了讓主機板、嵌入式系統或手機用以連接低速週邊裝置而發展。I<sup>2</sup>C 的正確讀法為 "**I-squared-C**"，而 "**I-two-C**" 則是另一種錯誤但被廣泛使用的讀法。自 2006 年 11 月 1 日起，使用 I<sup>2</sup>C 協定已經不需要支付專利費，但製造商仍然需要付費以取得 I<sup>2</sup>C 從屬裝置位址。

應用：

I<sup>2</sup>C 被應用在簡單的週邊且其製造成本較傳輸速度更為要求。

一些常見的應用如下：

1. 為了儲存使用者的設定而存取 NVRAM 晶片。
2. 存取低速的數位類比轉換器 (DAC)。
3. 存取低速的類比數位轉換器 (ADC)。
4. 改變監視器的對比度、色調及色彩平衡設定 (視訊資料通道)。
5. 改變音量大小。
6. 取得硬體監視及診斷資料，例如中央處理器的溫度及風扇轉速。
7. 讀取即時時鐘 (Real-time clock)。
8. 在系統裝置中用來開啟或關閉電源供應。



I<sup>2</sup>C 的另一個強大用途在於微控制器的應用，利用兩根通用的輸入輸出接腳及軟體的規劃，可以讓微控制器控制一個小型網路。週邊可以在系統仍然在運作的同時加入或移出匯流排，這代表對於有熱插拔需求的裝置而言是個理想的匯流排。像 I<sup>2</sup>C 這樣的匯流排之所以流行起來，是因為電腦工程師發現到對於整合電路設計而言，許多的製造成本源自於封裝尺寸及接腳數量。更小的包裝通常能夠減少重量及電源的消耗，這對於行動電話及手持式電腦而言格外重要。

### ● SPI 序列周邊介面匯流排

SPI (Serial Peripheral Interface Bus)，類似 I<sup>2</sup>C，是一種 4 線同步序列資料協定，適用於可攜式裝置平臺系統，但使用率較 I<sup>2</sup>C 少。

SPI 匯流排定義四組 logic signals：

- SCLK—Serial Clock (自 master 輸出)
- MOSI/SIMO—Master Output, Slave Input (自 master 輸出)
- MISO/SOMI—Master Input, Slave Output (自 slave 輸出)
- SS—Slave Select (active low; 自 master 輸出)

### ● UART (Universal Asynchronous Receiver/Transmitter)

實作序列傳輸介面的硬體積體電路，將資料在序列與並列形式之間做轉換。早期知名的晶片有 8250、16550。個人電腦以之實作出序列埠，在 DOS/Windows 下被標示為 COM port，至於其外接線路與連接埠，通常採用 RS-232 與後繼標準。若具備同步功能則稱為 USART (Universal Synchronous/Asynchronous Receiver/Transmitter)。

### ● PWM (Pulse-Width Modulation)

Pulse-Width Modulation 脈波寬度調變，常用於電子式電壓調整，例如燈光亮度、馬達控制、螢幕亮度、等等。其原理是藉由調整脈波的工作週期 (Duty cycle) 並影響平均值來控制電壓。

### ➤ GPIO (General-purpose input/output)

Raspberry Pi 上面擁有 40 pin 的 GPIO 接腳，我們可以使用這些接腳來進行硬體控制。目前用來控制 Raspberry Pi 上的 GPIO，網路資源中常見使用有 Python、Java 以及 C 語言，其他如 Ruby、Perl、PHP 等在網路上也有人寫出函式庫提供使用。另外也有人直接使用 Shell script 來寫 GPIO 控制程式。

Raspberry Pi 上面的 40 pin GPIO 接腳定義如下圖：

| Raspberry Pi B+<br>B+ J8 GPIO Header |             |         |    |    |        |  |  |            |                             |
|--------------------------------------|-------------|---------|----|----|--------|--|--|------------|-----------------------------|
| Alt Fn                               |             | Pin No. |    |    |        |  |  | Alt Fn     |                             |
|                                      |             | 3.3V    | 1  | 2  | 5V     |  |  |            |                             |
|                                      |             | GPIO2   | 3  | 4  | 5V     |  |  |            |                             |
| I <sup>2</sup> C {                   | SDA1        | GPIO3   | 5  | 6  | GND    |  |  |            |                             |
|                                      | SCL1        | GPIO4   | 7  | 8  | GPIO14 |  |  |            |                             |
|                                      | GPCLK0      | GND     | 9  | 10 | GPIO15 |  |  | TXD        | UART(serial)                |
|                                      |             |         |    |    | GPIO18 |  |  | RXD        |                             |
|                                      | CE1_1       | GPIO17  | 11 | 12 | GND    |  |  | PWM0/CE0_1 |                             |
|                                      |             | GPIO27  | 13 | 14 |        |  |  |            |                             |
|                                      |             | GPIO22  | 15 | 16 | GPIO23 |  |  |            |                             |
|                                      |             | 3.3V    | 17 | 18 | GPIO24 |  |  |            |                             |
| SPI {                                | MOSI_0      | GPIO10  | 19 | 20 | GND    |  |  |            |                             |
| Serial Peripheral Interface          | MISO_0      | GPIO9   | 21 | 22 | GPIO25 |  |  |            |                             |
|                                      | SCLK_0      | GPIO11  | 23 | 24 | GPIO8  |  |  | CE0_0      | SPI                         |
|                                      |             | GND     | 25 | 26 | GPIO7  |  |  | CE1_0      | Serial Peripheral Interface |
|                                      |             | DNC     | 27 | 28 | DNC    |  |  |            |                             |
|                                      | GPCLK1      | GPIO5   | 29 | 30 | GND    |  |  |            |                             |
|                                      | GPCLK2      | GPIO6   | 31 | 32 | GPIO12 |  |  | PWM0       |                             |
|                                      | PWM1        | GPIO13  | 33 | 34 | GND    |  |  |            |                             |
|                                      | PWM1/MISO_1 | GPIO19  | 35 | 36 | GPIO16 |  |  | CE2_1      |                             |
|                                      |             | GPIO26  | 37 | 38 | GPIO20 |  |  | MOSI_1     |                             |
|                                      |             | GND     | 39 | 40 | GPIO21 |  |  | SCLK_1     |                             |

需要注意的是 Raspberry Pi 的 GPIO:

- 1.有電流輸出的限制；每個接腳輸出最大為 16mA，全部接腳同時最大輸出為 50mA，所以通常會透過電流放大電路來驅動裝置建議不要直接驅動負載。
- 2.GPIO 為 3.3V 可容忍的電壓上限為 3.3V，千萬不要輸入 5V 電壓到接腳上，保護電路部分通常做法是接一個暫存器 (Register)或緩衝器(Buffer)來做緩衝。
- 3.GPIO 不是即插即用的介面；除了要非常小心避免電路接錯外，再開啟 Raspberry Pi 電源情況下不要隨意增加或修改電路。

## ➤ 補充:

電壓公式  $V = IR$

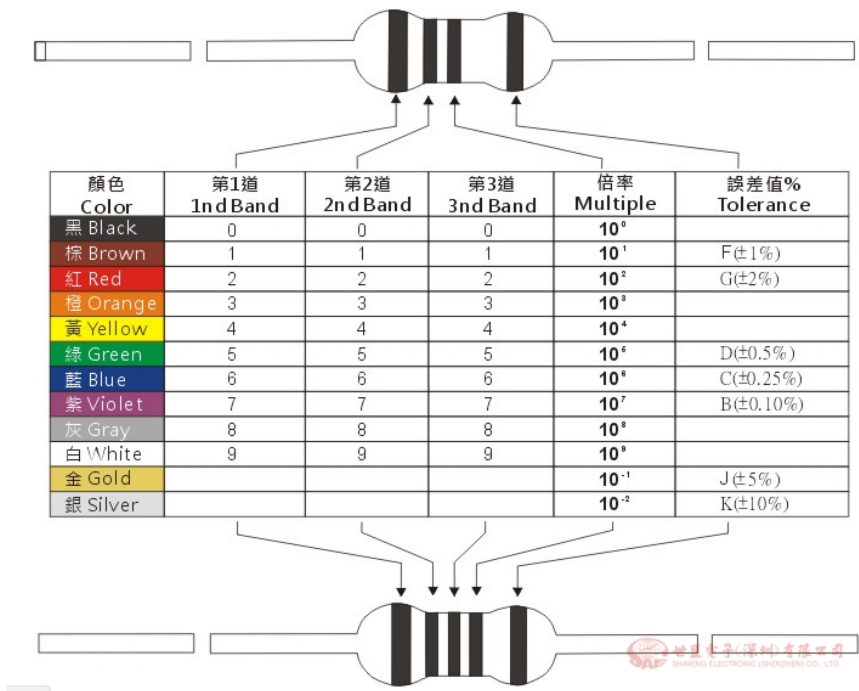
電壓值為  $V$  (Voltage)，電流值為  $I$  (Current)，電阻值為  $R$  (Resistance)，公式的意義是說電壓為通過電阻的電流，在電阻二端所產生的電壓。

電阻  $R$ (Resistor):

電流通過一個物體，會遭受不同程度的阻抗，這個特性我們稱之為電阻，不導電的物質其電阻值非常大，導電的物質，依其導電特性，會有不同的電阻值，金屬中應屬黃金其導電特性最好，然而價格太貴，我們都是用銅做為電線導體。在電子電路中，我們為了不同的電路特性(限制不同的電流大小通過電路)，設計了特殊的電子零件-電阻：

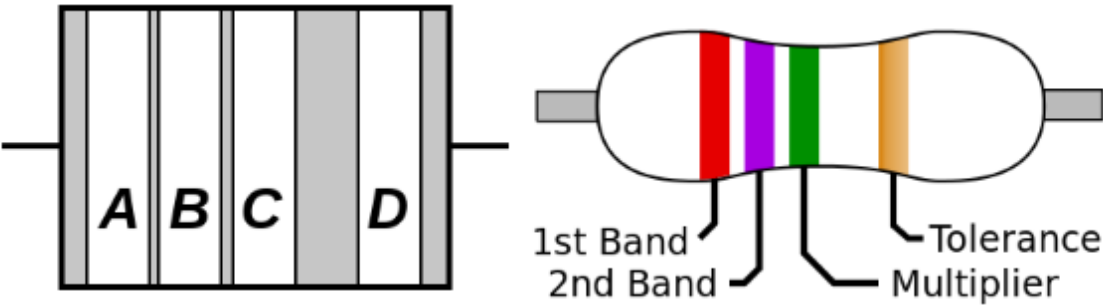


電阻其阻抗值從數歐姆到數 M 歐姆，電阻大小的識別以電阻上之色碼來標示，其色碼表示方式為：



電阻色碼:

電阻色碼，是一種以色彩碼標示出電阻器的電阻值與誤差範圍的方式，電容及電感也可用相同方式標示其容值（或感值）及誤差範圍。



其中色碼標示的電阻值其單位為歐姆。

- 色碼 A 為其數值的第一位數。
- 色碼 B 為其數值的第二位數。
- 色碼 C 為其倍率，若數字為  $c$ ，其倍率為。
- 色碼 D 若存在，則其表示數值的誤差範圍，若沒有色碼 D，其誤差範圍為 20%。

例如：一個電阻所用的色碼順序為黃, 紫, 紅, 金

第一個數字 4 (黃色)，第二個數字 7 (紫色)，倍率  $10^2$  (紅色)，電阻為  $4,700\ \Omega$ 。

誤差  $\pm 5\%$  (金色)， $\pm 10\%$  (銀色)，因此實際的電阻值在  $4,465$  與  $4,935\ \Omega$  之間。

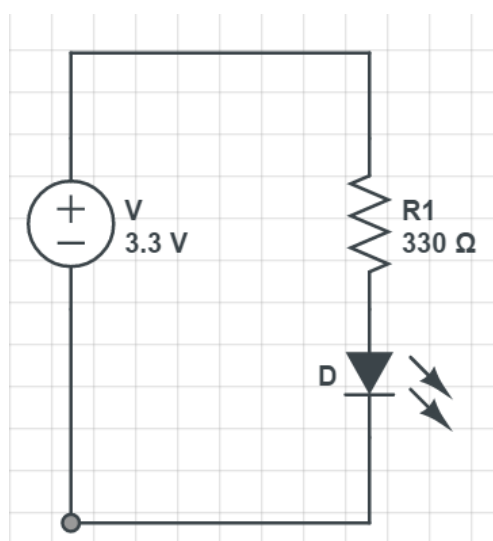
| 色碼(Color) | 代表數字<br>(Significant figures) | 倍率<br>(Multiplier) | 誤差<br>(Tolerance) |   | 溫度系數<br>(Temp. Coefficient)<br>(ppm/K) |   |
|-----------|-------------------------------|--------------------|-------------------|---|--|---|
| 黑(Black)  | 0                             | $\times 10^0$      | —                 |   | 250                                    | U |
| 棕(Brown)  | 1                             | $\times 10^1$      | $\pm 1\%$         | F | 100                                    | S |
| 紅(Red)    | 2                             | $\times 10^2$      | $\pm 2\%$         | G | 50                                     | R |
| 橙(Orange) | 3                             | $\times 10^3$      | —                 |   | 15                                     | P |
| 黃(Yellow) | 4                             | $\times 10^4$      | —                 |   | 25                                     | Q |
| 綠(Green)  | 5                             | $\times 10^5$      | $\pm 0.5\%$       | D | 20                                     | Z |
| 藍(Blue)   | 6                             | $\times 10^6$      | $\pm 0.25\%$      | C | 10                                     | Z |
| 紫(Violet) | 7                             | $\times 10^7$      | $\pm 0.1\%$       | B | 5                                      | M |
| 灰(Gray)   | 8                             | $\times 10^8$      | $\pm 0.05\%$      | A | 1                                      | K |
| 白(White)  | 9                             | $\times 10^9$      | —                 |   | —                                      |   |
| 金(Gold)   | —                             | $\times 10^{-1}$   | $\pm 5\%$         | J | —                                      |   |
| 銀(Silver) | —                             | $\times 10^{-2}$   | $\pm 10\%$        | K | —                                      |   |
| 透明(None)  | —                             | —                  | $\pm 20\%$        | M | —                                      |   |

## ➤ LED 閃爍與明暗度控制

說明：

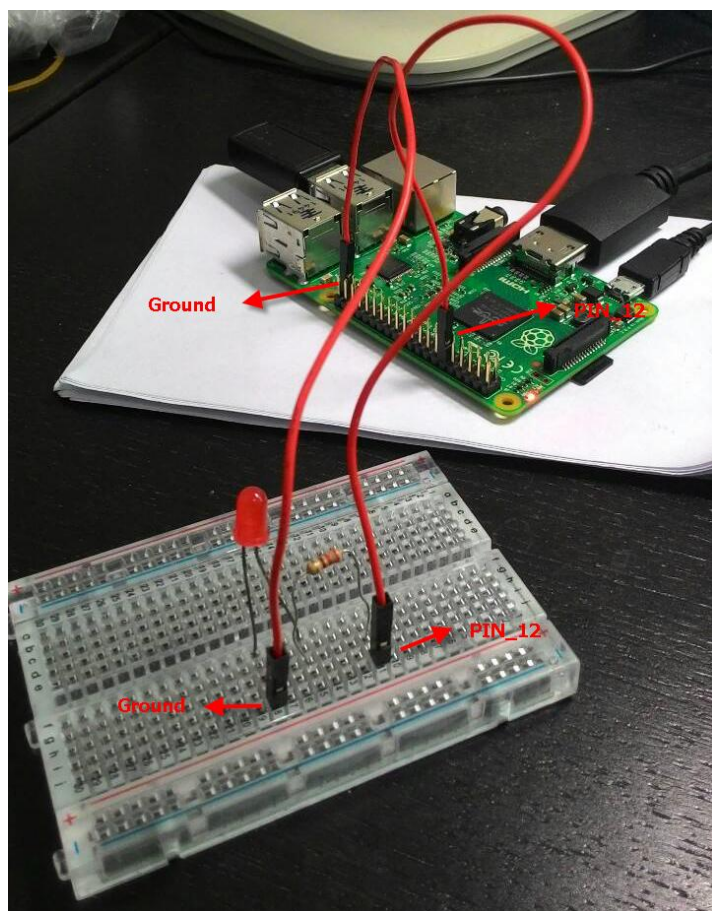
一般學習程式言時，第一支程式會是輸出 Hello World 到螢幕上，而基本電子電路則從 LED 明暗閃爍著手。

- 1.按圖施工將電路先佈置好，記得開機後不要隨意插拔電路元件。
- 2.確定開機完成，在 Windows 作業系統上透過遠端連線軟體(Putty)連線到 Raspberry Pi。
- 3.新版本的 Raspbian 作業系統已經內建 RPi.GPIO 這是針對 Python 的 GPIO 函式庫。
- 4.用文字編輯器(nano 或 vi)將下列 Python 程式碼命名為 HelloLED.py。



實際電路圖：





GPIO.setmode :

Board Pin 實體的針腳編號 : `GPIO.setmode(GPIO.BOARD)`

BCM GPIO 編號 : `GPIO.setmode(GPIO.BCM)`

範例（使用實體的針腳編號）：

範例: `ledblink_board.py`

範例（使用 GPIO 的編號）：

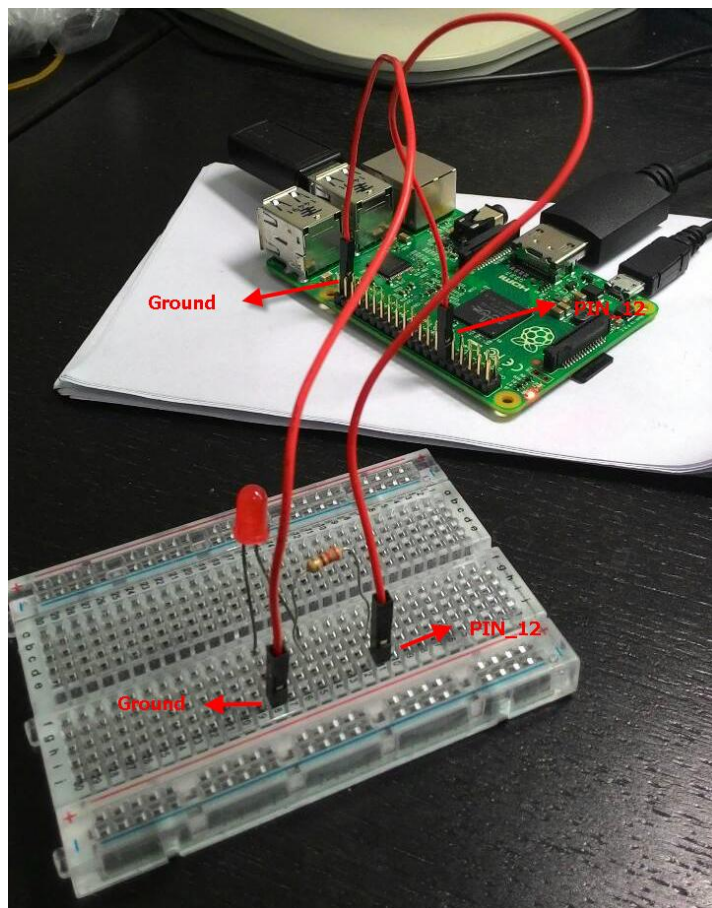
範例: `ledblink_gpio.py`

## ➤ 控制 LED 明暗度

說明：

讓紅色 LED 從亮到暗、再從暗到亮。

實際電路圖：



範例：

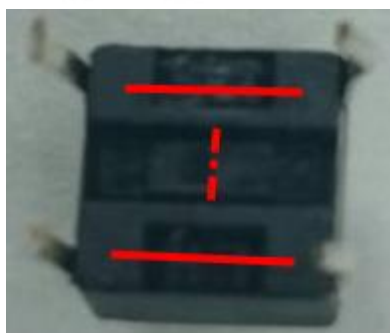
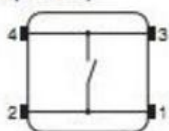
範例: `ledpwm_board.py`

### ➤ 數位輸入 + 加入按鈕 (彈跳問題)

說明：

利用主板上的 GPIO 擴充埠連結到麵包板上，讓外接的按鈕能透過玩家輸入按下狀態顯示在電腦螢幕上，因此要做一些電路上的配置。

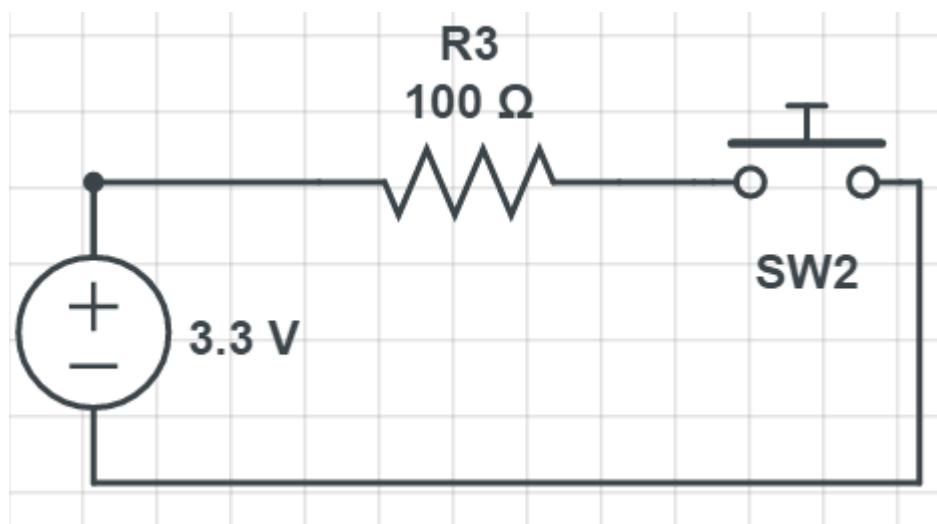
Terminal Arrangement/  
Internal Connections  
(Top View)



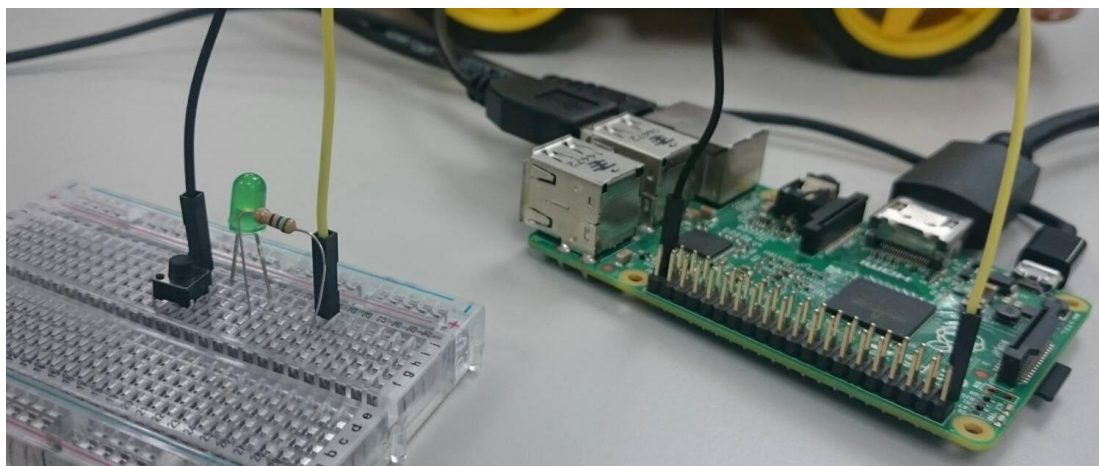
1-2、4-5 是通的，按下按鈕，則 1-2-3-4 都是通的。

測試 button 的原理 1：

電路圖：

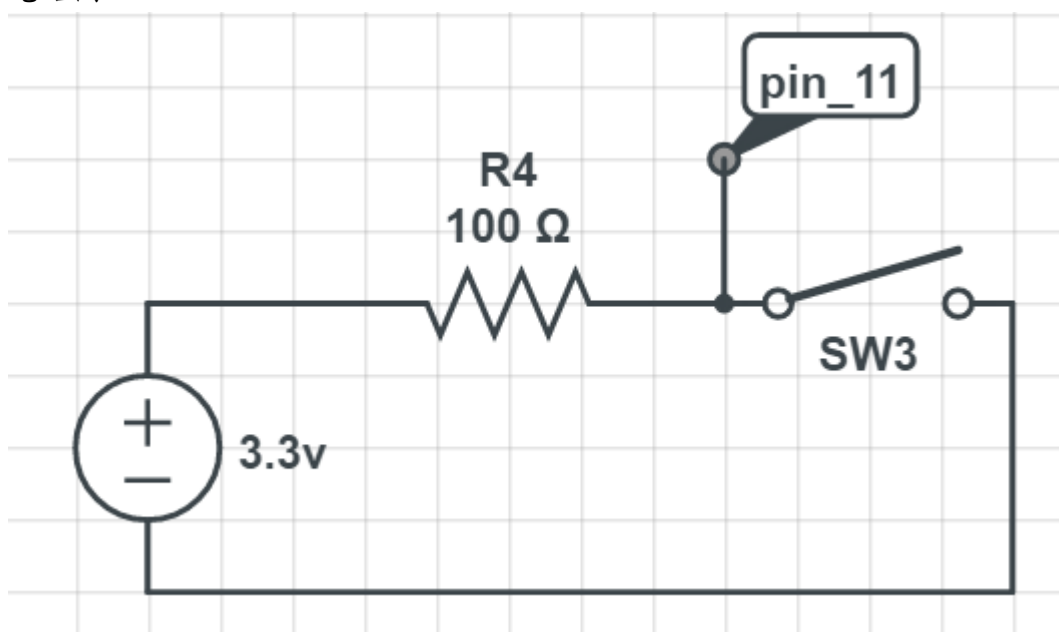


實際電路圖：

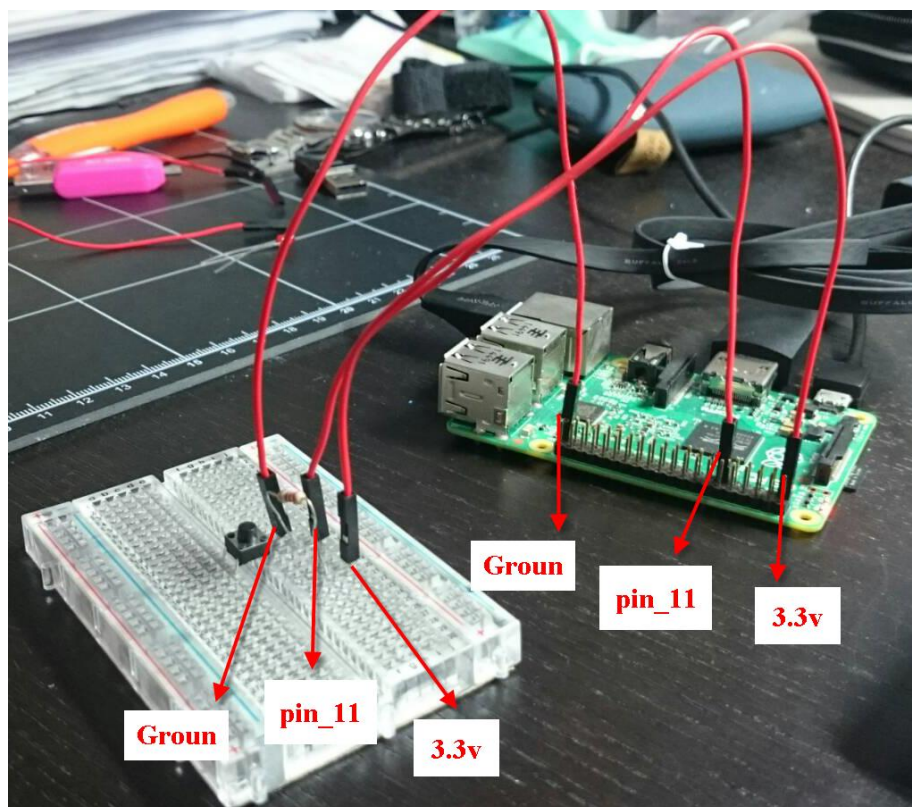


測試 button 的原理 2：

電路圖：



實際電路圖：



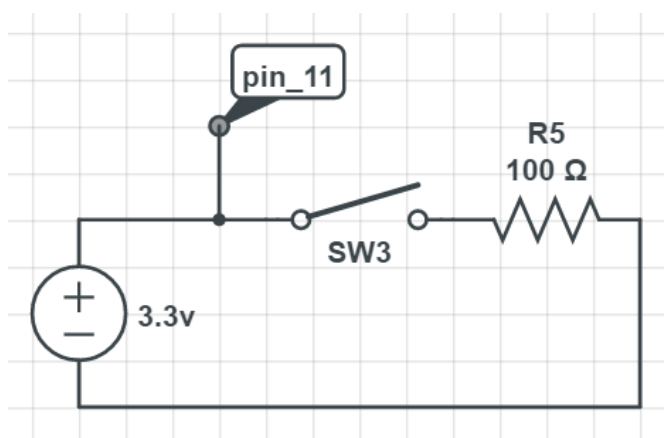
範例：

範例: `button_board.py`

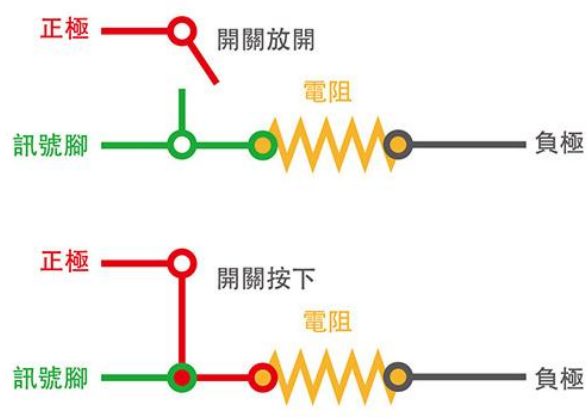
範例: `button_board_time.py`

測試 button 的原理 3：

電路圖：







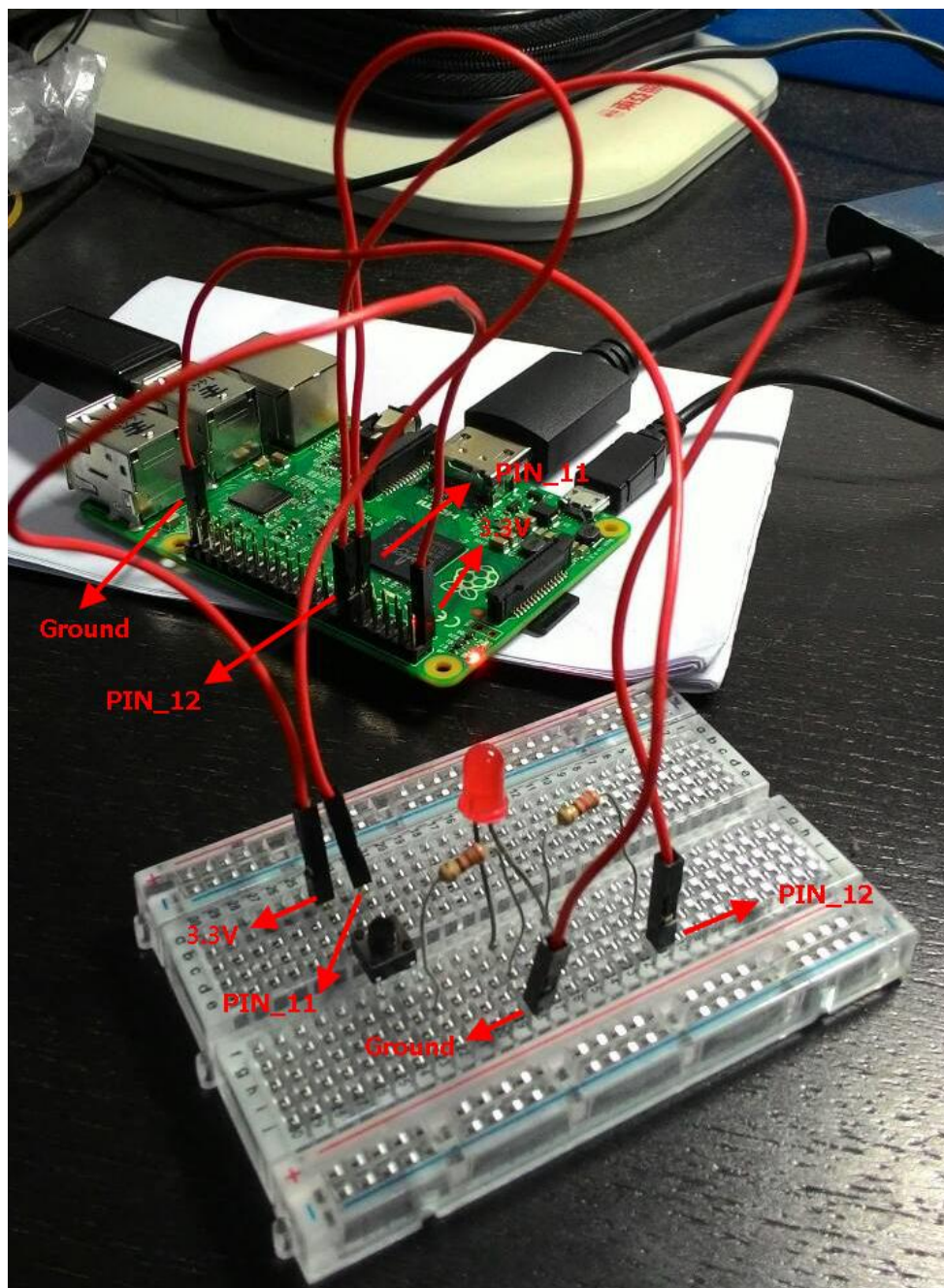
實際電路圖：

## ➤ 以開關控制 LED

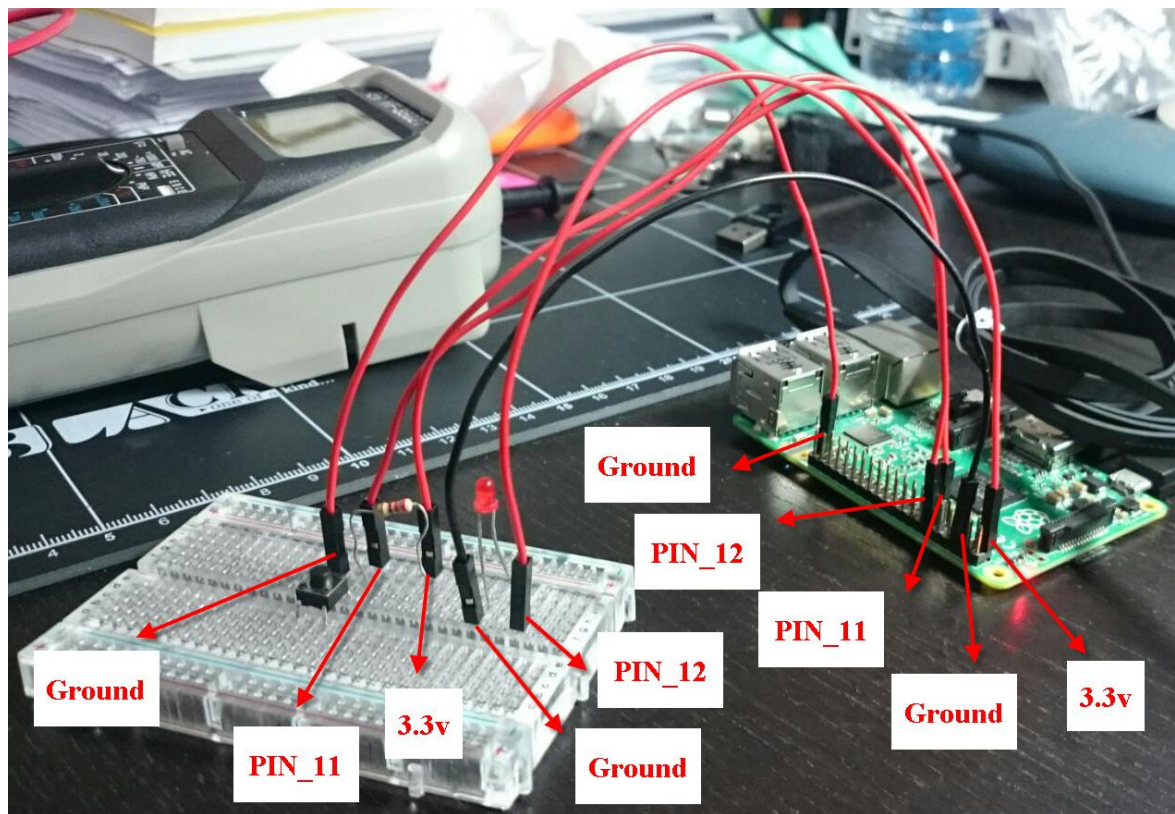
說明：

當使用者按下按鍵，若 led 本來為暗的狀態下，則 led 轉變為亮的狀態；若 led 本來為亮的狀態下，則 led 轉換成暗的狀態。

實際電路圖，第一種方法：



實際電路圖，第二種方法：



範例：

範例: `switch_led_board_button.py`

範例: `switch_led_board_button_time.py`

## ➤ 作業

材料如下：

- Raspberry Pi Model B x 1
- 麵包版 x 1
- LED 綠色 x 1
- LED 紅色 x 1
- 連接線 x 3 條

接線

將 Raspberry Pi 的第 pin 12(GPIO18)接到綠色 LED 的一支腳，第 32pin (GPIO12)接到紅色 LED 的一支腳，LED 的另一支腳則接到第 6pin (Ground)。

第一題：

讓紅色及綠色 LED 各閃 10 次，每次間隔 0.5 秒

第二題：

讓紅色及綠色 LED 燈分成 0-100 等級亮度，且讓使用者輸入亮度。