

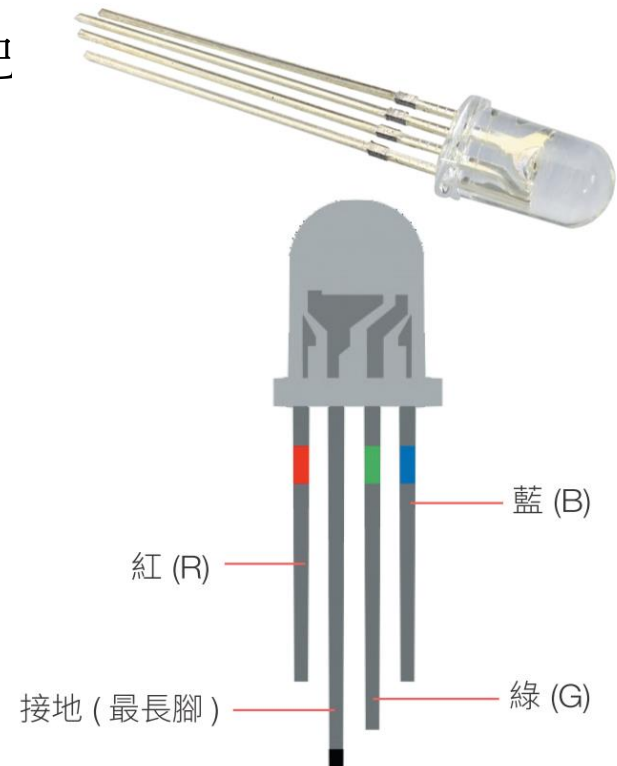
Ch05 燈光控制 – RGB LED

5-1 認識 RGB LED 與電阻

- **RGB LED**

LED (發光二極體) 是一種通電就會發光的半導體元件，效率高、可靠且省電，普遍用於照明用途。

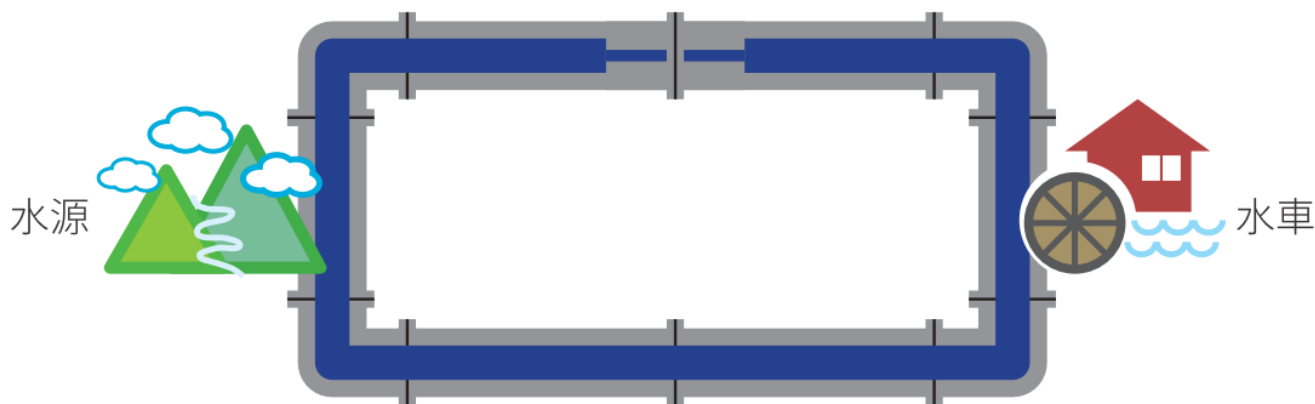
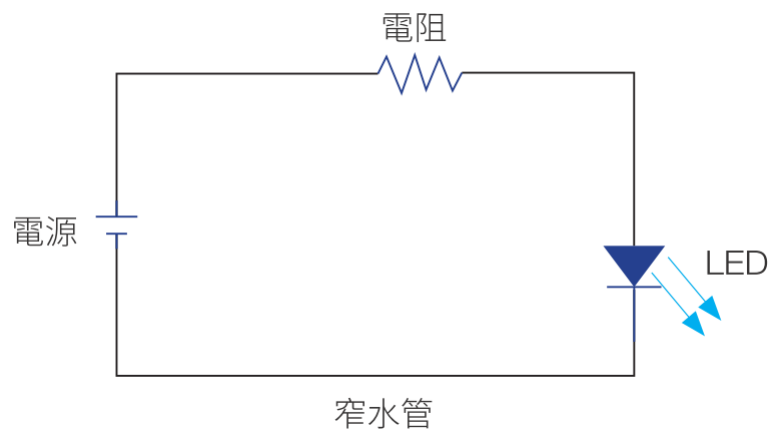
這裡使用的 RGB LED 只要對特定顏色該顏色的光就會點亮。



5-1 認識 RGB LED 與電阻

- 電阻

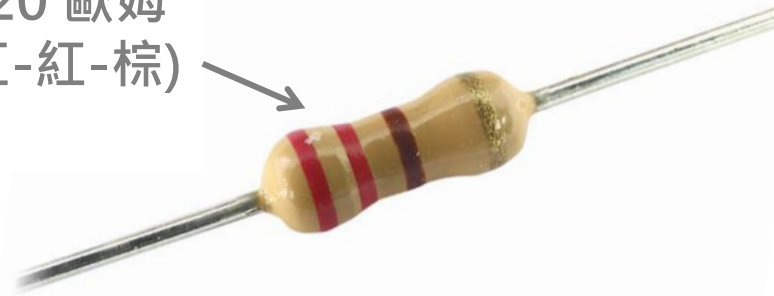
LED 能承受的電流量有限，為保護 LED 之間加入「電阻」，以限制流經的電流量。



5-1 認識 RGB LED 與電阻

歐姆 (ohms) 為衡量電阻值的單位。

220 歐姆
(紅-紅-棕)



5-2 控制 RGB LED

- Lab14

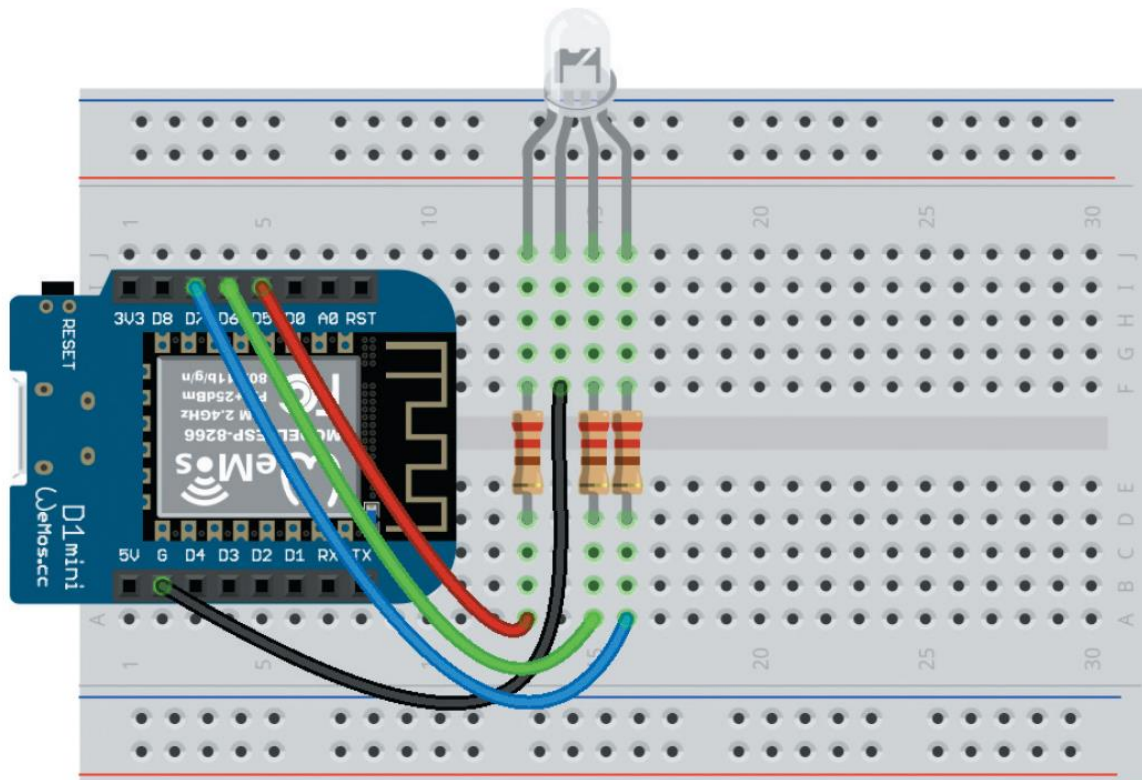
隨機炫彩燈	
實驗目的	使 RGB LED 隨機亮起不同顏色
材料	<ul style="list-style-type: none">• D1 mini• RGB LED• 220 歐姆電阻 x 3

- 接線圖

RGB LED 腳位	D1 mini 對應腳位
R	D5 (14 號腳位)
接地	G
G	D6 (12 號腳位)
B	D7 (13 號腳位)

5-2 控制 RGB LED

控制板腳位與 RGB LED 的顏色控制腳位之間必須接上電阻。



5-2 控制 RGB LED

- 設計原理

匯入 `urandom` (亂數) 函式庫，
並呼叫 `urandom.getrandbits()` 方法取得亂數：

```
import urandom
value = urandom.getrandbits(1)
```

`urandom.getrandbits()` 的參數代表要產生幾位元的亂數。
可用 `urandom.getrandbits(1)` 來決定 RGB LED 某顏色是否開啟。

5-2 控制 RGB LED

為避免三個顏色都剛好關閉，程式內會判斷，若紅與綠色都被關閉，藍色就一定得打開

想知道某個腳位目前處於高電位還是低電位，可讀取：

```
value = r.value() # 讀取紅燈腳位的電位 (傳回 0 或 1)
```


5-2 控制 RGB LED

- 程式設計

```
from machine import Pin
import utime, urandom

r = Pin(14, Pin.OUT)
g = Pin(12, Pin.OUT)
b = Pin(13, Pin.OUT)

while True:

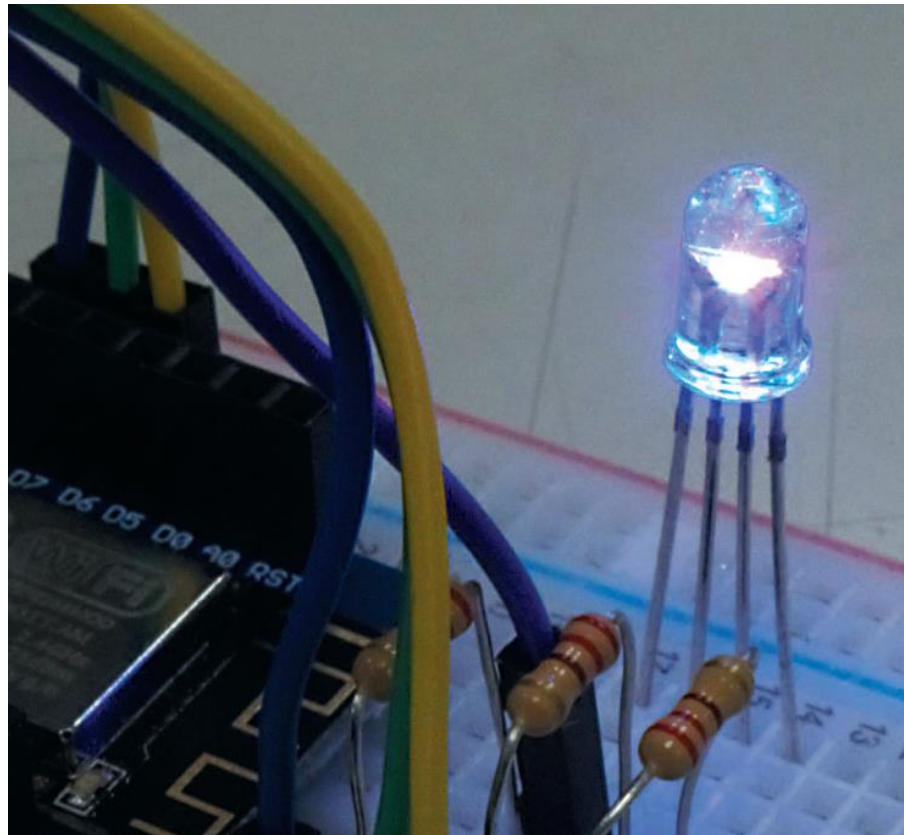
    r.value(urandom.getrandbits(1))
    g.value(urandom.getrandbits(1))
    b.value(1 if r.value() == 0 and g.value() == 0 else
            urandom.getrandbits(1))

    utime.sleep_ms(500)
```

5-2 控制 RGB LED

- 實測

執行程式，可看到 RGB LED 每半秒會改變一次顏色：



5-3 控制 LED 明暗度

- **Lab15**

七彩呼吸燈	
實驗目的	以 PWM 控制 RGB LED, 變成能逐漸點亮或熄滅的七彩燈
材料	<ul style="list-style-type: none">• D1 mini• RGB LED• 220 歐姆電阻 x3

- **接線圖**

同 Lab 14 。

5-3 控制 LED 明暗度

- 設計原理

D1 mini 的 PWM 功能可調整輸出電壓和頻率；

藉由改變 PWM 工作週期改變電壓，來改變 LED 的亮度。

```
from machine import Pin, PWM
import utime, urandom

# 初始化 PWM, 設為最高電壓 (等於高電位或數位信號 1, 100% 亮度)
led = PWM(Pin(14, Pin.OUT), freq=1000, duty=1023)
led.duty(512) # 電壓 1/2 (50% 亮度)
led.duty(128) # 電壓 1/8 (12.5% 亮度)
led.duty(0)   # 電壓 0 (低電位或數位信號 0, 完全不亮)
led.deinit() # 關閉 PWM
```

5-3 控制 LED 明暗度

- 程式設計

```
from machine import Pin, PWM
import utime, urandom

r = PWM(Pin(14, Pin.OUT), freq=1000, duty=0)
g = PWM(Pin(12, Pin.OUT), freq=1000, duty=0)
b = PWM(Pin(13, Pin.OUT), freq=1000, duty=0)

while True:

    # 隨機決定要顯示的顏色
    r_switch = urandom.getrandbits(1)
    g_switch = urandom.getrandbits(1)
    b_switch = 1 if r_switch == 0 and g_switch == 0 else
                urandom.getrandbits(1)
```

5-3 控制 LED 明暗度

```
# 漸層點亮
for i in range(1024):
    if r_switch == 1:
        r.duty(i)
    if g_switch == 1:
        g.duty(i)
    if b_switch == 1:
        b.duty(i)
    utime.sleep_ms(1)

# 漸層轉暗
for i in reversed(range(1024)):
    if r_switch == 1:
        r.duty(i)
    if g_switch == 1:
        g.duty(i)
    if b_switch == 1:
        b.duty(i)
    utime.sleep_ms(1)
```

5-3 控制 LED 明暗度

第一個 for 迴圈的 range(1024) 代表讓迴圈中變數 i 從 0 遞增到 1023 。

第二個 for 迴圈加上 reversed()，讓變數 i 從 1023 遞減至 0。迴圈中把 RGB LED 各腳位的 PWM 工作週期設為變數 i，燈的特定顏色就會漸亮再漸滅。

- **實測**

執行程式，觀察 RGB LED 慢慢明滅的效果。

5-4 隨環境變化明暗度

- Lab16

智慧小夜燈	
實驗目的	用超音波模組及亮度感測模組打造能在太暗或有人靠近時就亮起的小夜燈
材料	<ul style="list-style-type: none">• D1 mini• 220 歐姆電阻 x3• 亮度感測模組• RGB LED• 超音波模組

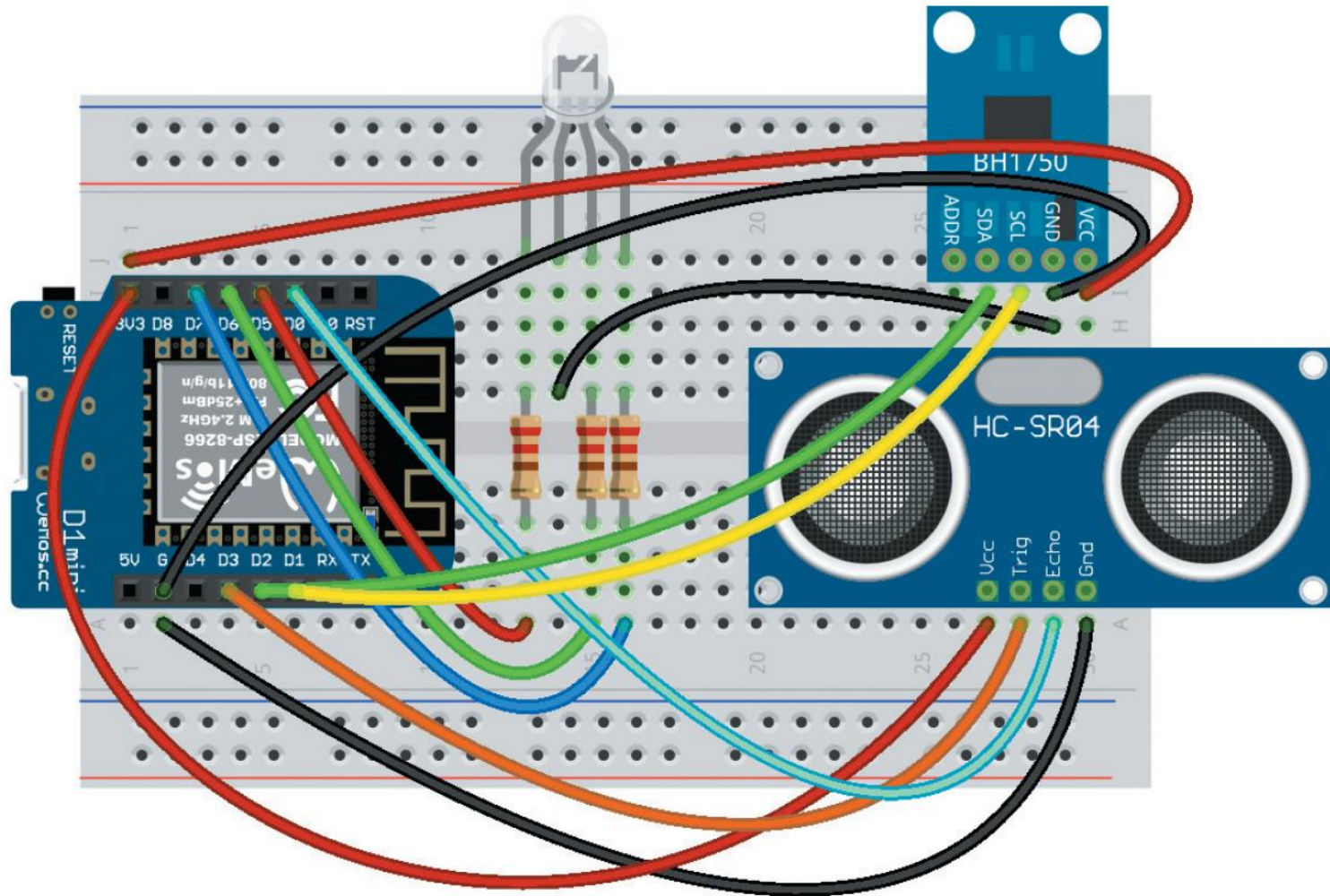
5-4 隨環境變化明暗度

- 接線圖

超音波模組腳位	意義	D1 mini 對應腳位
Vcc	電源	3V3
Trig	觸發腳	D3 (0 號腳位)
Echo	接收腳	D0 (16 號腳位)
Gnd	接地	G

亮度模組腳位	意義	D1 mini 對應腳位
Vcc	電源	3V3
GND	接地	G
SCL	串列時脈線	D1 (5 號腳位)
SDA	串列資料線	D2 (4 號腳位)
ADDR	位址切換	不接線

5-4 隨環境變化明暗度



5-4 隨環境變化明暗度

- 設計原理

希望小夜燈有以下功能：

- ✓ 超音波模組感應到有東西靠近時，讓 LED 完全點亮。
- ✓ 若沒有，就根據環境亮度調整燈光，周遭越暗 LED 越亮。
(只有環境亮度低於 256 lux 時才會亮，
LED 亮度設為用 256 減去環境亮度再乘以 4 。)

5-4 隨環境變化明暗度

- 程式設計

```
from machine import Pin, I2C, PWM
from hcsr04 import HCSR04
import bh1750fvi, utime

sonar = HCSR04(trigger_pin=0, echo_pin=16)

r = PWM(Pin(14, Pin.OUT), freq=1000, duty=0)
g = PWM(Pin(12, Pin.OUT), freq=1000, duty=0)
b = PWM(Pin(13, Pin.OUT), freq=1000, duty=0)

while True:

    distance = sonar.distance_cm()
    light_level = bh1750fvi.sample(I2C(scl=Pin(5),
                                     sda=Pin(4)), mode=0x23)

    if 2 <= distance <= 30:
        led_light_value = 1023
```

5-4 隨環境變化明暗度

```
else:
    if light_level > 256:
        light_level = 256
    led_light_value = (256 - light_level) * 4 - 1

    r.duty(led_light_value)
    g.duty(led_light_value)
    b.duty(led_light_value)

    utime.sleep_ms(50)
```

- 實測

執行程式，將手放在超音波模組前面，LED 會完全點亮；接著確保超音波模組前 30 公分沒有東西，遮住亮度感測器，燈光隨偵測到的環境亮度而漸亮。