

RGB LED

RGB LED 模組

版本： V1.0



產品介紹：

利基 RGB LED 模組是一個簡單易用模組，體積最小(18 mm x18 mm)，採用高亮度 RGB LED，並利用標準 I2C 訊號溝通，控制容易，可串聯 63 組 RGB LED 模組，另外可設定 RGB 比例控制 LED 顏色，最多顏色可達 1677 萬，具備自動閃爍功能，適用於各種燈光裝飾場合，變化豐富。

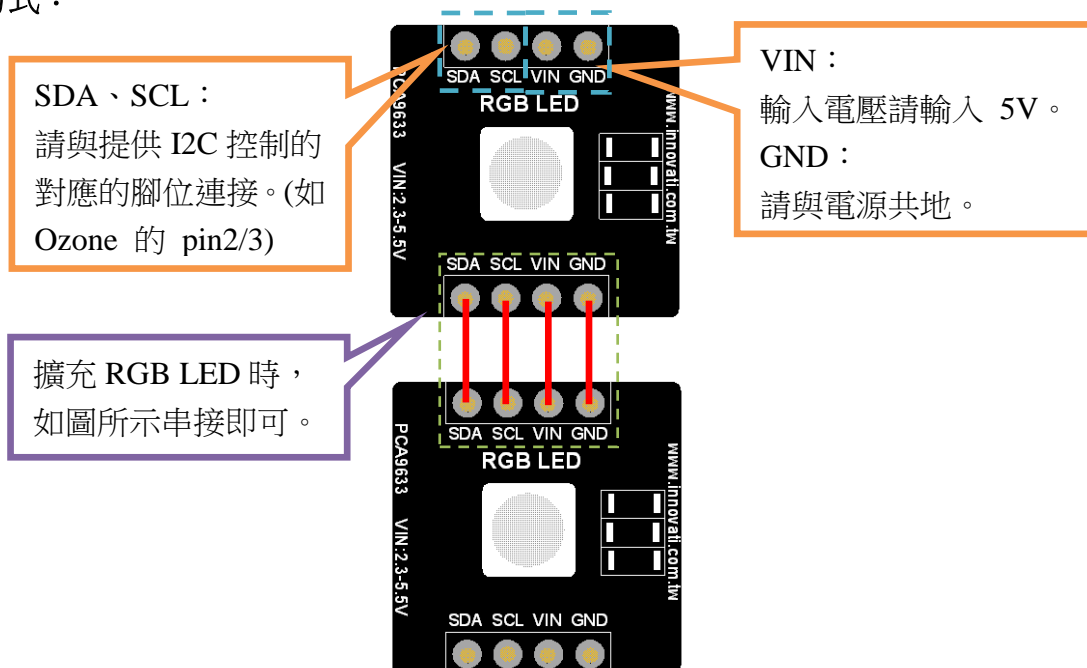
應用方向：

- 安裝於各種機構上，增加顯示的效果。

產品特色：

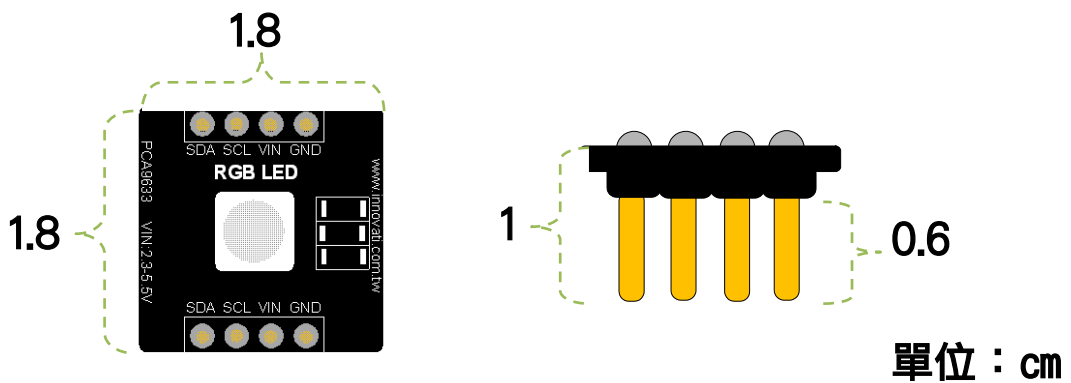
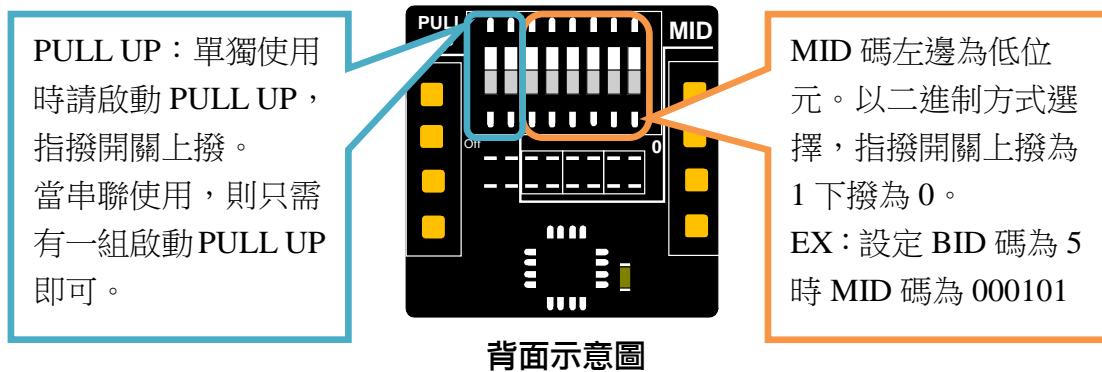
- 使用 I2C 控制方式，可同時串聯 62 組 RGB LED 模組。
- 可依照 R、G、B 比例定義出所需的顏色。
- 可設定調光、閃爍模式。
- 可定義群組。
- 尺寸小，易於安裝。
- 輸入電壓 5V。

連接方式：



正面示意圖

產品規格：



SUB 設定：RGB_LED 可自行定義四個群組，定義方式如下。(預設為群組 1)

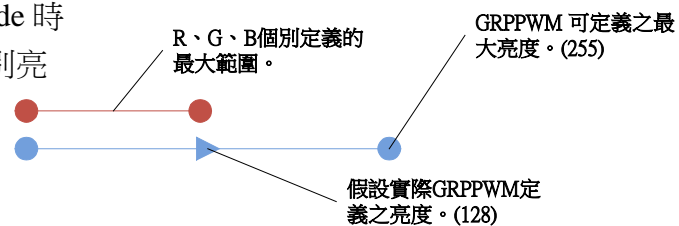
群組名稱	SUB4	SUB3	SUB2	SUB1
代碼	8	4	2	1

EX：若 SUB 設定為 10，則指定的 RGB_LED 隸屬於 SUB4、SUB2。

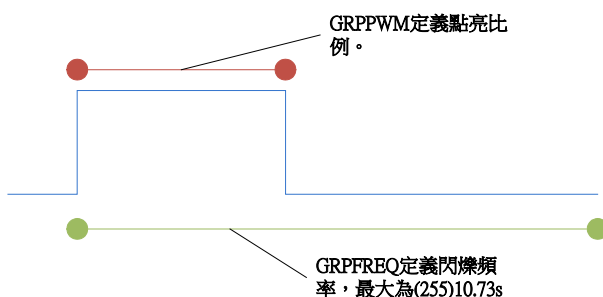
若 SUB 設定為 15，則指定的 RGB_LED 隸屬於 SUB1、SUB2、SUB3、SUB4。

調光模式 (Dimming Mode)：

依照右圖所示，於 DimmingMode 時 GRPPWM 可定義 R、G、B 個別亮度的最大範圍。



閃爍模式 (Blinking Mode)：



依照左圖所示，若 GRPFREQ 定義為最大 255(10.73s), GRPPWM 定義為 128(50%) 則 RGB LED 執行結果為點亮 5.36s、點滅 5.36s。

操作注意事項：

- ④ 模組操作溫度 -40 °C ~ 85 °C
- ④ 模組儲存溫度 -40 °C ~ 125 °C
- ④ 當串聯使用時，只需一組 RGB LED 啟動 PULL UP。
- ④ MID 碼 3 為內部使用，請勿設定。

指令表：

下面的指令表是專供控制 RGB LED 模組的各種指令，必要輸入的指令名稱與參數，以粗底或粗斜體表示，粗體的文字在輸入時請不要更改，粗斜體的文字請自行定義適當格式的參數填入。

指令格式		指令功能
啟動相關指令		
RGB_LED_PowerOff(<i>ID</i>)		將 <i>ID</i> 所指定的 RGB LED 模組，進入休眠模式。
RGB_LED_PowerOn(<i>ID</i>)		將 <i>ID</i> 所指定的 RGB LED 模組，離開休眠模式。
RGB_LED_TurnOn(<i>ID</i>)		將 <i>ID</i> 所指定的 RGB LED 模組點亮。
RGB_LED_TurnOff(<i>ID</i>)		將 <i>ID</i> 所指定的 RGB LED 模組熄滅。
設定相關指令		
RGB_LED_SetRGB(<i>ID,R,G,B</i>)		將 <i>ID</i> 所指定的 RGB LED 模組，依照 <i>R</i> 、 <i>G</i> 、 <i>B</i> 的內容設定三原色比例。 <i>R</i> 、 <i>G</i> 、 <i>B</i> 可定義的範圍為 0~255。

```

#include <Wire.h>

void RGB_LED_TurnOn(unsigned char ID) {
    Wire.beginTransmission(ID);
    Wire.write(0x88);
    Wire.write(0xAA);
    Wire.endTransmission();
}

void RGB_LED_TurnOff(unsigned char ID) {
    Wire.beginTransmission(ID);
    Wire.write(0x88);
    Wire.write(0x00);
    Wire.endTransmission();
}

void RGB_LED_PowerOn(unsigned char ID) {
    Wire.beginTransmission(ID);
    Wire.write(0x80);
    Wire.write(0x0F);
    Wire.endTransmission();
}

void RGB_LED_PowerOff(unsigned char ID) {
    Wire.beginTransmission(ID);
    Wire.write(0x80);
    Wire.write(0x10);
    Wire.endTransmission();
}

void RGB_LED_SetRGB(unsigned char ID, unsigned char R, unsigned char G, unsigned char B) {
    Wire.beginTransmission(ID);
    Wire.write(0x82);
    Wire.write(B);
    Wire.write(G);
    Wire.write(R);
    Wire.endTransmission();
}

void setup(void)
{
    Wire.begin();           // SDA/SCL 接至 Ozone pin2/3
    RGB_LED_PowerOn(63);   // 將 ID 63 的 RGB LED 啟動
    RGB_LED_TurnOn(63);    // 將 ID 63 的 RGB LED 點亮
}

void loop(void)

```

```
{  
  RGB_LED_SetRGB(63,255,0,0); //將 ID 63 的設定為紅色  
  delay(1000); //等待 1 秒  
  
  RGB_LED_SetRGB(63,0,255,0); //將 ID 63 的設定為綠色  
  delay(1000); //等待 1 秒  
  
  RGB_LED_SetRGB(63,0,0,255); //將 ID 63 的設定為藍色  
  delay(2000); //等待 2 秒  
}
```