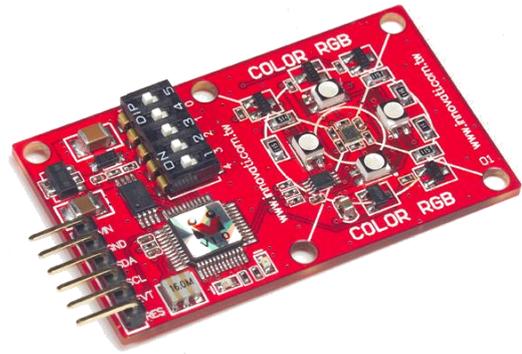


Color RGB

顏色感測模組

版本: V2.0



產品介紹: 利基 Color RGB 模組是一個簡單易用的三波長顏色感測模組，透過 cmdBUS 與 Arminno 連接，可以用簡單的指令取得感測的顏色值，包含總和的光強度，以及分離的紅藍綠三色，也可以設定顏色編號與感測值，進行自動配對比較，模組提供三色 LED，可以直接調整光源強度。

應用方向:

- 量測物體反射的三色值。
- 感測環境的光強度。

產品特色:

- 可以分別感測紅藍綠三色光的反射值(λ_p ，紅: 640 nm，綠: 524 nm，藍: 470 nm)。
- 可以感測三色光混和的光強度。
- 反映強度大於等於 $61 \mu W/cm^2$ 的能量值。
- 整合四組三色光 LED，直接以軟體即可控制開關，也能各別調整各色光源強度。
- 可以直接用比例值減弱 LED 光源強度。
- 三組光感測時間可供隨時變換，達到類似快門切換效果。
- 提供多組內建回傳值比例切換，用以取得較適當的回傳值範圍，包括四組放大比例值與七組縮小比例值，都可直接以軟體選擇切換。
- 可透過 cmdBUS，下達指令。

連接方式: 直接將 ID 開關撥至欲設定的編號，再將 cmdBUS 連接至 Arminno 上對應的腳位，就可透過 Arminno 執行操作。

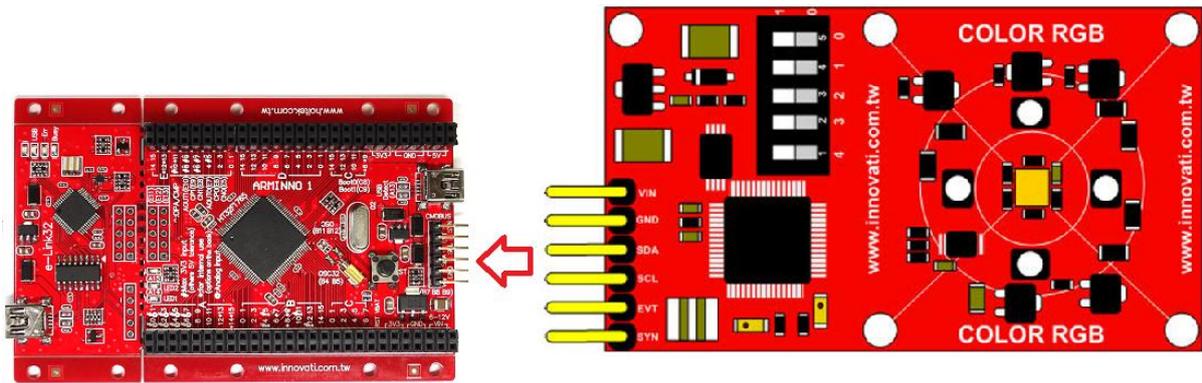
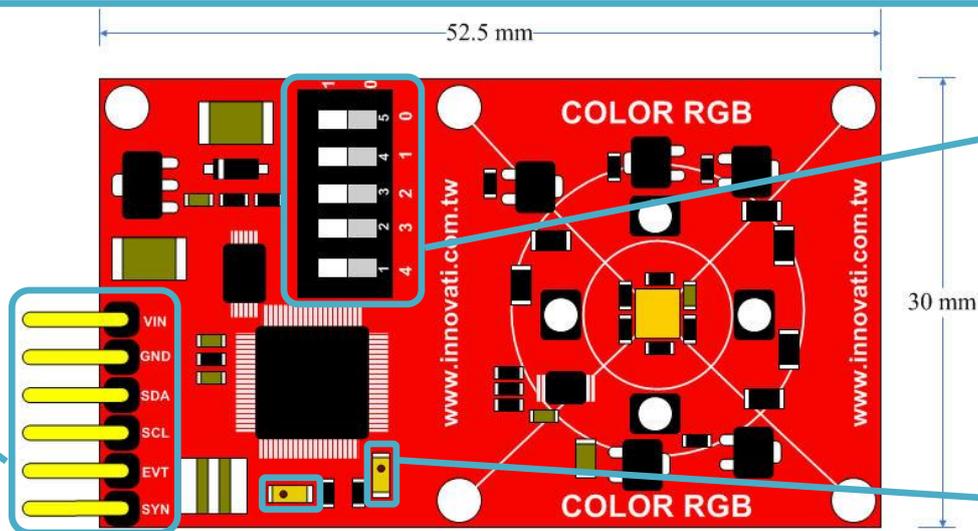


圖 1: 與 Arminno 連接

產品規格:

cmdBUS 接腳，將此處腳位與 Arminno 對應腳位相接，即可透過 Arminno 操控 Color RGB 模組(連接時請注意腳位對應，將 Vin 對接 Arminno 上的 Vin 腳位，若是腳位錯誤可能造成模組損毀)。

模組編號設定開關，由右至左以二進制設定 Color RGB 模組的模組編號，編號可以讓 Arminno 操控時，判斷想要控制的模組(請參考附錄 2)



橘色指令指示燈，閃爍代表模組與 Arminno 正在收送資料

圖 2: 模組腳位與開關介紹

操作注意事項:

模組操作溫度 0 °C ~ 70°C

模組儲存溫度 -40 °C ~ 85°C

Optical Characteristics, $V_{DD} = 3\text{ V}$, $T_A = 25^\circ\text{C}$, (unless otherwise noted)

PARAMETER	TEST CONDITIONS	Red Channel			Green Channel			Blue Channel			Clear Channel			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
R_e Irradiance responsivity, 64x / 12 ms setting	$\lambda_p = 470\text{ nm}$, 61 $\mu\text{W}/\text{cm}^2$	1%	15%	15%	40%	65%	90%	65.5			(Counts/ $\mu\text{W}/\text{cm}^2$)			
	$\lambda_p = 524\text{ nm}$, 52 $\mu\text{W}/\text{cm}^2$	1%	15%	60%	90%	10%	35%	77						
	$\lambda_p = 640\text{ nm}$, 44.4 $\mu\text{W}/\text{cm}^2$	80%	110%	1%	15%	1%	15%	90						

NOTE: The percentage shown represents the ratio of the respective red, green, or blue channel value to the clear channel value.

表 1: 光反應值參考表

SPECTRAL RESPONSIVITY

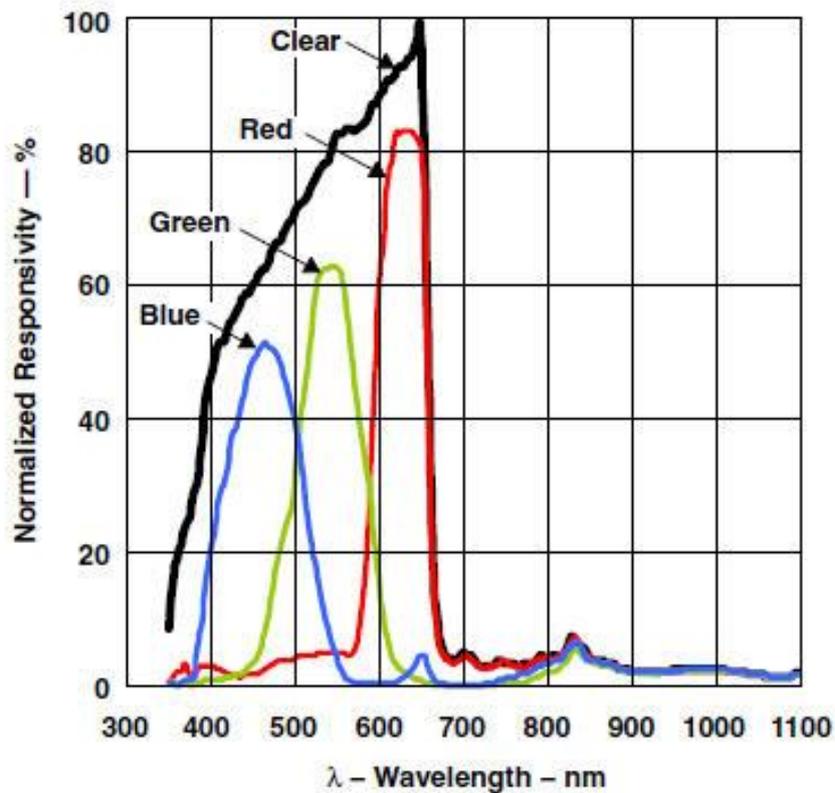


表 2: 頻譜響應

指令格式	指令功能
量測與量測值讀取相關指令	
<i>uint8_t bStatus = GetRGB(uint16_t &ValueR, uint16_t &ValueG, uint16_t &ValueB)</i>	取得量測狀態放在 <i>bStatus</i> ，0 代表上次讀值後尚未更新量測值，1 則代表此為更新的量測值，並取得紅綠藍光感測值，分別存在 <i>ValueR</i> 、 <i>ValueG</i> 、 <i>ValueB</i> ，此數值為量測到的計數值， <i>ValueR</i> 、 <i>ValueG</i> 、 <i>ValueB</i> 回傳值都為 0~65535 間的整數
<i>uint8_t bStatus = GetCRGB(uint16_t &ValueC, uint16_t &ValueR, uint16_t &ValueG, uint16_t &ValueB)</i>	取得量測狀態放在 <i>bStatus</i> ，0 代表上次讀值後尚未更新量測值，1 則代表此為更新的量測值，並取得無濾鏡與紅綠藍光感測值，分別存在 <i>ValueC</i> 、 <i>ValueR</i> 、 <i>ValueG</i> 、 <i>ValueB</i> ，此數值為量測到的計數值， <i>ValueR</i> 、 <i>ValueG</i> 、 <i>ValueB</i> 回傳值都為 0~65535 間的整數
<i>StartMeasure(void)</i>	啟動偵測，執行後會根據設定的感測時間，執行顏色的量測，須先執行此命令，並等到量測結束，才能用取得感測值相關指令，得到最新的感測值，請注意執行量測時，會自動根據設定的 LED 強度，在量測時點亮 LED 光源
量測控制相關指令	
<i>TurnOnLED(void)</i>	點亮 LED 燈直到執行 <i>TurnOffLED</i> 指令為止
<i>TurnOffLED(void)</i>	關閉 LED 燈，但執行 <i>StartMeasure</i> 仍會依據設定亮度，在量測時自動點亮
<i>SetLEDRGB(uint8_t Red, uint8_t Green, uint8_t Blue)</i>	分別以 <i>Red</i> 、 <i>Green</i> 、 <i>Blue</i> ，設定 LED 三色光(紅，綠，藍)點亮時的強度， <i>Red</i> 、 <i>Green</i> 、 <i>Blue</i> 可以輸入 0~255 間的整數值
<i>ScaleLED(uint8_t Scale)</i>	將設定的 LED 三色光強度，乘上輸入的 <i>Scale</i> 值，以百分比為單位，再以乘積更改三色光的強度值，乘積最大最小值會自動調整，設定為介於 0~255 之間的整數值， <i>Scale</i> 可以輸入 0~200 間的整數值
<i>LoadLED(void)</i>	將 LED 的光強度值，更改回預設值
各項設定相關指令	
<i>SetPeriod(uint8_t Period)</i>	根據 <i>Period</i> 設定量測的時間 0: 12 ms 1: 100 ms 2: 400 ms 預設為 0 (12 ms)
	根據 <i>Gain</i> 設定量測值的增益量

SetGain(uint8_t Gain)	0: 1X 1: 4X 2: 16X 3: 64X 預設為 0 (1X)
SetPrescaler(uint8_t Prescaler)	根據 Prescaler 設定量測值的縮小量 0: 1 1: 1/2 2: 1/4 3: 1/8 4: 1/16 5: 1/32 6: 1/64 預設為 0 (1)

範例程式:

```

#include "arminno.h"

ColorRGB myC(3);

uint16_t g_wC, g_wR, g_wG, g_wB; // 設定感測值儲存參數
uint8_t bStatus; // 設定狀態儲存參數

int main(void)
{
    myC.LoadLED();

    myC.SetPeriod(0); // 設定量測時間
    myC.SetGain(0); // 設定量測倍率
    myC.SetPrescaler(0);

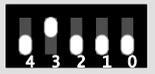
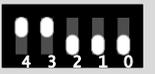
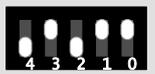
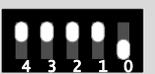
    while(1)
    {
        myC.StartMeasure(); // 開始量測顏色值
        do {
            bStatus = myC.GetCRGB(g_wC, g_wR, g_wG, g_wB); // 讀取感測的顏色值
            Pause(100);
        } while(bStatus != 1);
        printf("C=%d\r\n", g_wC);
        printf("R=%d\r\n", g_wR);
        printf("G=%d\r\n", g_wG);
        printf("B=%d\r\n", g_wB);
        Pause(2000);
    }
}

```

附錄

1. 已知問題:

2. 模組編號開關對應編號表:

	0		8		16		24
	1		9		17		25
	2		10		18		26
	3		11		19		27
	4		12		20		28
	5		13		21		29
	6		14		22		30
	7		15		23		31