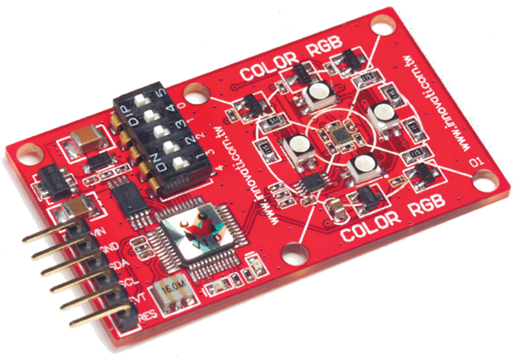
Color RGB

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 BASIC Commander 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。
- 組合儲存色碼，根據色碼提供整合套件的動作判斷。

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。
- 提供 255 組色值儲存空間，並且可以對儲存的色值，單獨給予紅藍綠的辨識值。
- 多種搜尋指令，能針對部分範圍或全部記憶色值，搜尋與偵測值相近的儲存內容，也可以回傳各色的差異值，或是總差異值。
- 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。
- 偵測完成事件提醒，能更即時掌握量測結束時間。
- 搜尋結束事件提醒，在檢視儲存資料時能快速反應搜尋結果。
- 可透過 I2C，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 BASIC Commander 上對應的腳位，就可透過 BASIC Commander 執行操作。

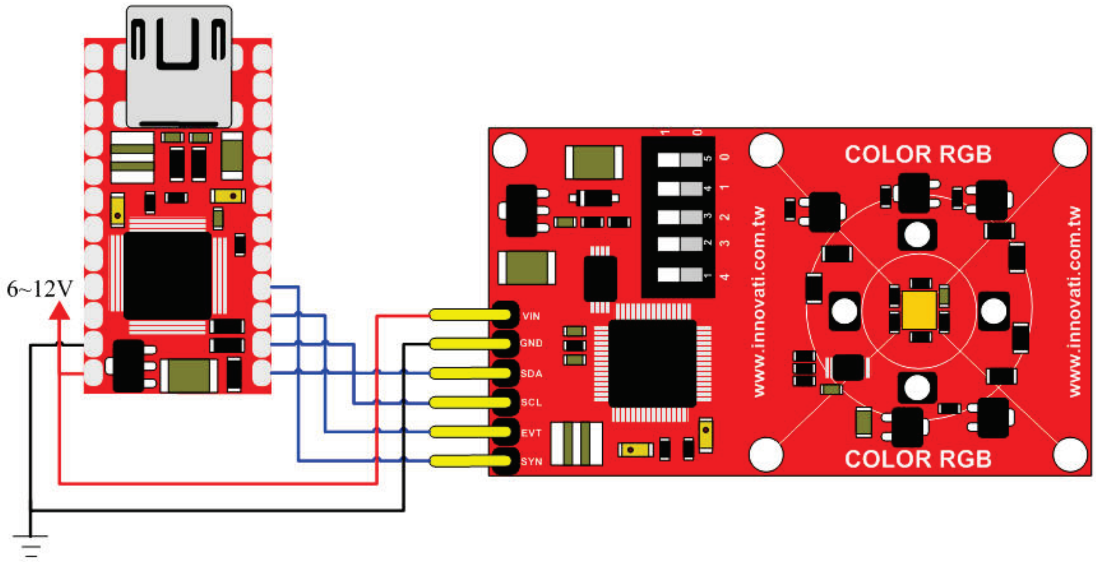


圖 1: 與 BASIC Commander 連接

產品規格:

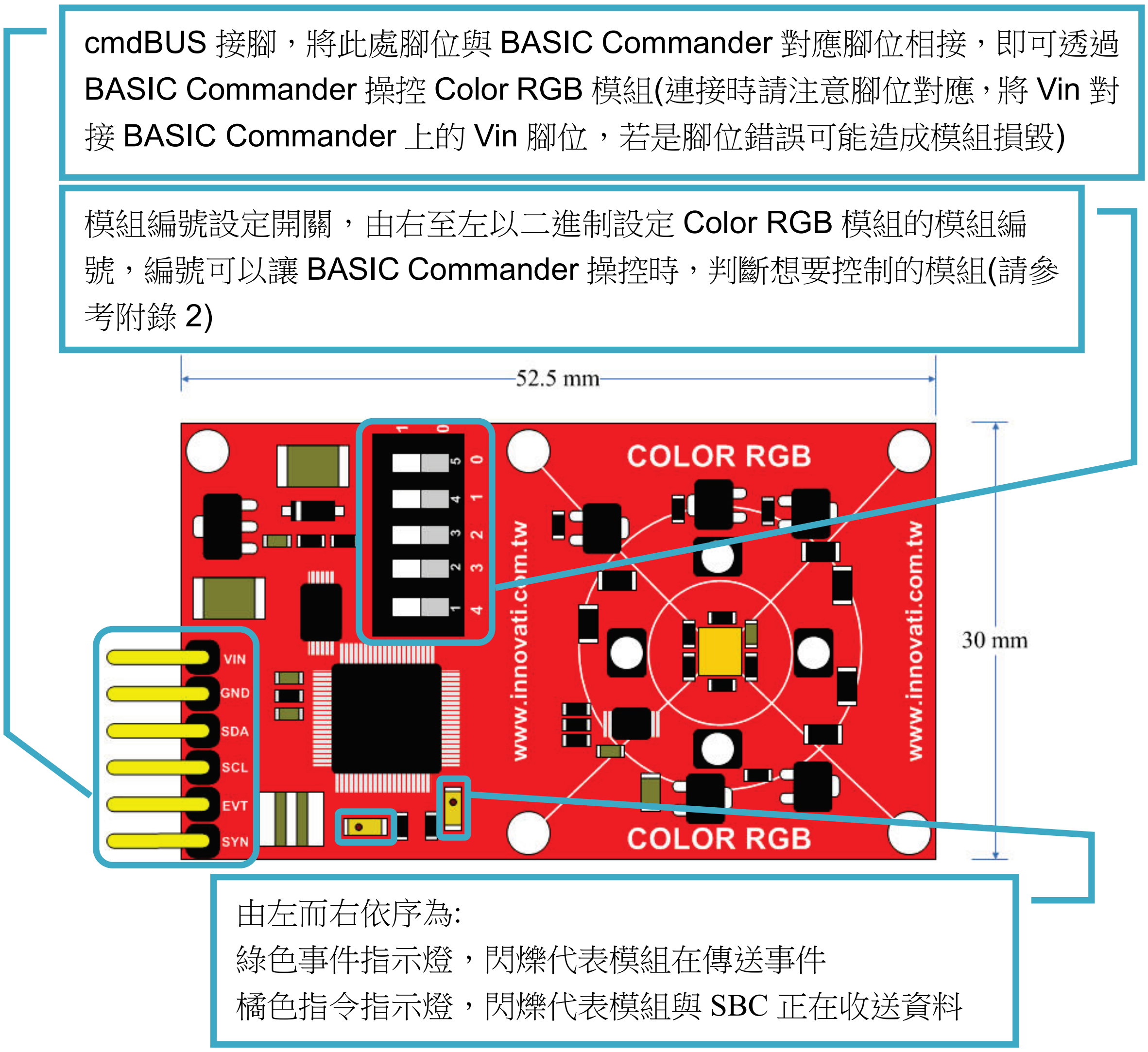


圖 2: 模組腳位與開關介紹

操作注意事項:

模組操作溫度　　0 ℃ ~ 70℃

模組儲存溫度　　-40 ℃ ~ 85℃

Optical Characteristics, V_{DD} = 3 V, T_A = 25°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, 64× / 12 ms setting	λ_p = 470 nm, 61 µW/cm^2	1%		15%	15%		40%	65%		90%		65.5		(Counts/ µW/ cm^2)
		λ_p = 524 nm, 52 µW/cm^2	1%		15%	60%		90%	10%		35%		77		
		λ_p = 640 nm, 44.4 µW/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

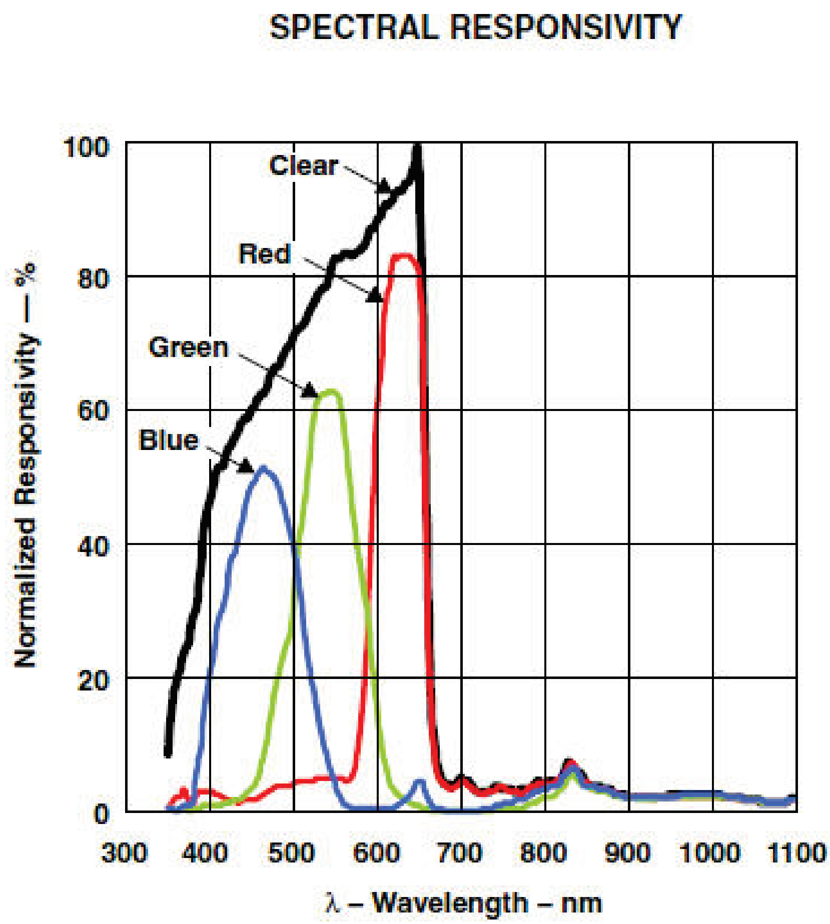


表 2: 頻譜響應

模組下達指令的方式可分為兩種：cmdBUS、I2C 控制方式

cmdBUS 指令表:

下面的指令表是專供控制 Color RGB 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。輸入時請注意 innoBASIC Workshop 大寫與小寫會視為相同字。在執行 Color RGB 模組指令前，請先於程式開頭定義對應參數與編號，例:

Peripheral *ModuleName* As ColorRGB @ *ModuleID*

I2C 通訊協議(Protocol):

為了使更廣泛的使用者能控制模組，提供了部份指令的通訊協議讓使用者應用。透過通訊規格，使用者可使用 I2C 通訊協議為模組下達命令。

通訊協議常見的封包如下：

MID：模組 ID 編號，空間大小為 Byte 的變數。對應於硬體的指播開關。

CID：命令 ID 編號，空間大小為 Byte 的變數。依不同命令而改變。

CheckSum1：驗證位元_1，空間大小為 Byte 的變數。

定義方式：255 – (MID * 2) – CID

CheckSum2：驗證位元_2，空間大小為 Byte 的變數。

定義方式：255 – (CheckSum1~CheckSum2 之間的變數總和)

CheckSum3：驗證位元_3，空間大小為 Byte 的變數。

定義方式：255 – MID-(MID~CheckSum3 之間的變數總和)

Dummy：虛設位元，可為任意變數。空間大小為 Byte 的變數。

於通訊規格每筆資料空間大小階為 Byte，若資料空間大小超過一個 Byte 時，需將資料拆開，並由 Low Byte 開始傳送。

Ex:傳送資料 Temp 為一筆空間大小為 Word 的資料,則需將 Temp 拆開,分為 Temp_L、Temp_H，並且先傳送 Temp_L。

Ex1 模組編號為 2，命令編號為 153，傳送參數 Byte 為 100，通訊協議為

MID+CID+CheckSum1+Byte+CheckSum2+Dummy 則：

MID = 2

CID = 153

CheckSum1 = 255 – (2*2) – 153 = 98

Byte = 100

CheckSum2 = 255 – 100

Dummy = 0~255 之間的任意數

Ex2 模組編號為 2，命令編號為 153，傳送參數 Temp 為 511，通訊協議為

MID+CID+CheckSum1+Temp_L+Temp_H+CheckSum2+Dummy 則：

MID = 2

CID = 153

CheckSum1 = 255 – (2*2) – 153 = 98

Temp_L = 255，Temp_H = 1

CheckSum2 = 255 – Temp_L–Temp_H = 255

Dummy = 0~255 之間的任意數

指令格式	指令功能
量測與量測值讀取相關指令	
bStatus **= GetClear(*****Value*****)**	取得量測狀態放在 ***bStatus***，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，並取得無濾鏡感測值存在 ***Value***，此數值為量測到的計數值，***Value*** 會回傳 0~65535 間的整數
bStatus **= GetRed(*****Value*****)**	取得量測狀態放在 ***bStatus***，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，並取得紅光感測值存在 ***Value***，此數值為量測到的計數值，***Value*** 會回傳 0~65535 間的整數
bStatus **= GetGreen(*****Value*****)**	取得量測狀態放在 ***bStatus***，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，並取得綠光感測值存在 ***Value***，此數值為量測到的計數值，***Value*** 會回傳 0~65535 間的整數
bStatus **= GetBlue(*****Value*****)**	取得量測狀態放在 ***bStatus***，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，並取得藍光感測值存在 ***Value***，此數值為量測到的計數值，***Value*** 會回傳 0~65535 間的整數
CmdBUS： ***bStatus*** **= GetRGB(*****ValueR*****,** ***ValueG*****,** ***ValueB*****)** **I2C：** **Out：MID+92+CheckSum1+Dummy** **In：** **MID+bStatus+ValueR_L+ValueR_H+ValueG_L+ValueG_H+ValueB_L+ValueB_H+CheckSum3**	取得量測狀態放在 ***bStatus***，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 ***ValueR***、***ValueG***、***ValueB***，此數值為量測到的計數值，***ValueR***、***ValueG***、***ValueB*** 回傳值都為 0~65535 間的整數
CmdBUS： ***bStatus*** **= GetCRGB(*****ValueC*****,** ***ValueR*****,** ***ValueG*****,** ***ValueB*****)** **I2C：** **Out：MID+93+CheckSum1+Dummy**	取得量測狀態放在 ***bStatus***，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 ***ValueC***、***ValueR***、***ValueG***、***ValueB***，此數值為量測到的計數值，***ValueR***、***ValueG***、***ValueB*** 回傳值都為 0~65535 間的整數

In： **MID+bStatus+ ValueC_L+ValueC_H +ValueR_L+ValueR_H+ ValueG_L+ValueG_H+ ValueB_L+ValueB_H+CheckSum3**	
CmdBUS： **StartMeasure()** **I2C：** **MID+128+CheckSum1+Dummy**	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
儲存感測值相關指令	
SaveCurColorVal(*Num*)	將最新偵測到的感測值，存入以 ***Num*** 指定的位置，***Num*** 可以輸入 1~255 間的整數
SaveColorDes(*Num*, *DesR*, *DesG*, *DesB*)	將 ***Num*** 指定位置的感測值，以 ***DesR***、***DesG***、***DesB*** 分別設定顏色參考值，***Num*** 可以輸入 1~255 間的整數，***DesR***、***DesG***、***DesB*** 都可以輸入 0~255 間的整數
SaveColorVal(*Num*, *ValC*, *ValR*, *ValG*, *ValB*)	將 ***ValC***、***ValR***、***ValG***、***ValB*** 設定的無濾鏡與紅綠藍色量測值，放到 ***Num*** 指定位置，***Num*** 可以輸入 1~255 間的整數，***ValC***、***ValR***、***ValG***、***ValB*** 都可以輸入 0~65535 間的整數
SaveColorInfo(*Num*, *DesR*, *DesG*, *DesB*, *ValC*, *ValR*, *ValG*, *ValB*)	將 ***ValC***、***ValR***、***ValG***、***ValB*** 設定的無濾鏡與紅綠藍色量測值，放到 ***Num*** 指定位置，並且以 ***DesR***、***DesG***、***DesB*** 分別設定顏色參考值，***Num*** 可以輸入 1~255 間的整數，***ValC***、***ValR***、***ValG***、***ValB*** 都可以輸入 0~65535 間的整數，***DesR***、***DesG***、***DesB*** 都可以輸入 0~255 間的整數
GetColorVal(*Num*, *ValC*, *ValR*, *ValG*, *ValB*)	讀取 ***Num*** 指定位置中的顏色感測值，依無濾鏡與紅綠藍順序，存放於 ***ValC***、***ValR***、***ValG***、***ValB*** 中，***Num*** 可以輸入 1~255 間的整數，***ValC***、***ValR***、***ValG***、***ValB*** 會回傳 0~65535 間的整數
GetColorDes(*Num*, *DesR*, *DesG*, *DesB*)	讀取 ***Num*** 指定位置中的顏色參考值，依紅綠藍順序，存放於 ***DesR***、***DesG***、***DesB*** 中，***Num*** 可以輸入 1~255 間的整數，***DesR***、***DesG***、***DesB*** 會回傳 0~255 間的整數

搜尋儲存值相關指令	
MatchColorDesAll(*DesR*, *DesG*, *DesB*)	搜尋儲存資料中，顏色參考值與 ***DesR***、***DesG***、***DesB*** 相同的資料，執行後請先以 **GetMatchStatus** 指令或等待到 **SearchCompleteEvent** 送達，再執行讀取指令，***DesR***、***DesG***、***DesB*** 都可以輸入 0~255 間的整數
MatchColorDesIndex(*DesR*, *DesG*, *DesB*, *Min*, *High*)	搜尋儲存資料中，顏色參考值與 ***DesR***、***DesG***、***DesB*** 相同的資料，並且只會搜尋介於 ***Min***、***High*** 之間儲存位置，執行後請先以 **GetMatchStatus** 指令或等待到 **SearchCompleteEvent** 送達，再執行讀取指令，***DesR***、***DesG***、***DesB*** 都可以輸入 0~255 間的整數，***Min***、***High*** 請輸入 1~255 間的整數值，且 ***High*** 需要大於 ***Min***
MatchCurColorAll()	根據最新的量測值作為搜尋基準，並對所有儲存位置做搜尋，執行後請先以 **GetMatchStatus** 指令或等待到 **SearchCompleteEvent** 送達，再執行讀取指令
MatchCurColorIndex(*Min*, *High*)	根據最新的量測值作為搜尋基準，並對介於 ***Min***、***High*** 之間的儲存位置做搜尋，執行後請先以 **GetMatchStatus** 指令或等待到 **SearchCompleteEvent** 送達，再執行讀取指令，***Min***、***High*** 請輸入 1~255 間的整數值，且 ***High*** 需要大於 ***Min***
MatchCurColorRGBAll()	根據最新的量測值，但扣除無濾鏡感測值，作為搜尋基準，並對所有儲存位置做搜尋，執行後請先以 **GetMatchStatus** 指令或等待到 **SearchCompleteEvent** 送達，再執行讀取指令
MatchCurColorRGBIndex(*Min*, *High*)	根據最新的量測值，但扣除無濾鏡感測值，作為搜尋基準，並對介於 ***Min***、***High*** 之間的儲存位置做搜尋，執行後請先以 **GetMatchStatus** 指令或等待到 **SearchCompleteEvent** 送達，再執行讀取指令，***Min***、***High*** 請輸入 1~255 間的整數值，且 ***High*** 需要大於 ***Min***
GetMatchNum(*Num*)	將搜尋結果的儲存位置，放於 ***Num*** 中，***Num*** 會回傳 0~255 間的整數值
GetMatchNeighborNum(*Num1*, *Num2*, *Num3*)	回傳與搜尋設定最接近的三個儲存位置，分別放於 ***Num1***、***Num2*** 與 ***Num3*** 中，***Num1***、***Num2*** 與 ***Num3*** 會回傳 0~255 間的整數值

bStatus **= GetMatchStatus()**	取得比對量指令動作的狀態值，回傳的 ***bStatus*** 值有三種不同狀態 0: 沒有搜尋中的動作，或是搜尋完畢，沒有找到對應的值 1: 搜尋中，如果此時再執行搜尋，就會自動取消這一次的搜尋，並且開啟新的搜尋 2: 搜尋結束，並且可以用取得搜尋結果相關指令，取得對應的值
GetMatchDiff(*DiffC*, *DiffR*, *DiffG*, *DiffB*)	取得比對後，最接近的儲存值，與比對值的差異，分別放於 ***DiffC***、***DiffR***、***DiffG*** 與 ***DiffB*** 中，會回傳-5000~5000 間的整數值
GetMatchDiffAll(*Diff*)	取得比對後，最接近的儲存值，與比對值的總差異量，放於 ***Diff*** 中，會回傳 0~20000 間的整數值
SetMatchMaxDiff(*Max*)	將 ***Max*** 值，設定為最大的單色配對差異量，如果儲存的資料，與量測值超過此設定值，即使總差異量小於其他儲存值，仍不會判斷為對應值，***Max*** 可以輸入 0~5000 之間的整數值，預設為 500
GetMatchMaxDiff(*Max*)	取得最大的單色配對差異量，放於 ***Max*** 中，***Max*** 會回傳 0~5000 之間的整數值，預設為 500
量測控制相關指令	
CmdBUS： **TurnOnLED()** **I2C：** **MID+111+CheckSum1+Dummy**	點亮 LED 燈直到執行 **TurnOffLED** 指令為止
CmdBUS： **TurnOffLED()** **I2C：** **MID+112+CheckSum1+Dummy**	關閉 LED 燈，但執行 **StartMeasure** 仍會依據設定亮度，在量測時自動點亮
GetLEDStatus(*bStatus*)	取得 LED 的狀態值，存放於 ***bStatus*** 中 0: LED 關閉 1: LED 開啟
CmdBUS： **SetLEDRGB(*Red*, *Green*, *Blue*)** **I2C：** **MID+124+CheckSum1+Red+Green+Blue** **+CheckSum2+Dummy**	分別以 ***Red***、***Green***、***Blue***，設定 LED 三色光(紅，綠，藍)點亮時的強度，***Red***、***Green***、***Blue*** 可以輸入 0~255 間的整數值

GetLEDRGB(*Red*, *Green*, *Blue*)	取得設定的 LED 三色光(紅，綠，藍)點亮時的強度，分別放於 ***Red***、***Green***、***Blue*** 中，***Red***、***Green***、***Blue*** 會回傳 0~255 間的整數值
CmdBUS： **ScaleLED(*Scale*)** **I2C：** **MID+126+CheckSum1+Scale** **+CheckSum2+Dummy**	將設定的 LED 三色光強度，乘上輸入的 ***Scale*** 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，***Scale*** 可以輸入 0~200 間的整數值
LoadLED()	將 LED 的光強度值，更改回預設值
各項設定相關指令	
CmdBUS： **SetPeriod(*Period*)** **I2C：** **MID+114+CheckSum1+Period** **+CheckSum2+Dummy**	根據 ***Period*** 設定量測的時間 0: 12 ms 1: 100 ms 2: 400 ms 預設為 0 (12 ms)
GetPeriod(*Period*)	取得設定的量測時間，放於 ***Period*** 中，***Period*** 會回傳 0~2 間的整數，預設為 0
CmdBUS： **SetGain(*Gain*)** **I2C：** **MID+116+CheckSum1+Gain** **+CheckSum2+Dummy**	根據 ***Gain*** 設定量測值的增益量 0: 1X 1: 4X 2: 16X 3: 64X 預設為 0 (1X)
GetGain(*Gain*)	取得設定的量測值增益量放於 ***Gain*** 中，***Gain*** 會回傳 0~3 間的整數值
CmdBUS： **SetPrescaler(*Prescaler*)** **I2C：** **MID+118+CheckSum1+Prescaler** **+CheckSum2+Dummy**	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)
GetPrescaler(*Prescaler*)	取得設定的量測值縮小量放於 ***Prescaler*** 中，***Prescaler*** 會回傳 0~6 間的整數值

提醒事件設定相關指令	
EnSearchCompleteEvent()	啟動 **SearchCompleteEvent** 提醒事件，預設為關閉
DisSearchCompleteEvent()	關閉 **SearchCompleteEvent** 提醒事件，預設為關閉
EnValReadyEvent()	啟動 **ValReadyEvent** 提醒事件，預設為開啟
DisValReadyEvent()	關閉 **ValReadyEvent** 提醒事件，預設為開啟

模組提供應用事件:

事件名稱 (Event)	啟動條件
SearchCompleteEvent	在執行 **EnSearchCompleteEvent** ()後，執行搜尋儲存值相關指令，當搜尋動作完成時，就會啟動
ValReadyEvent	在執行 **EnValReadyEvent** ()後，當量測動作完成時，就會啟動

範例程式:

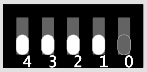
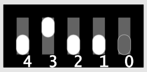
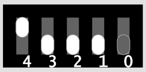
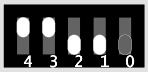
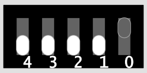
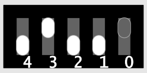
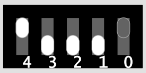
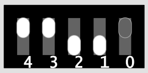
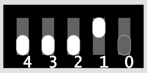
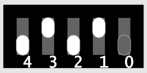
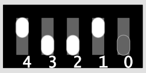
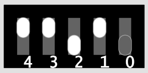
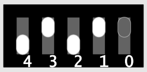
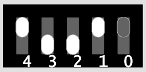
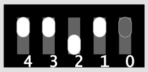
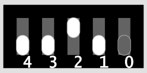
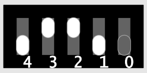
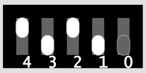
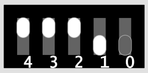
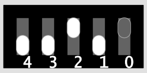
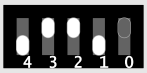
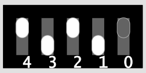
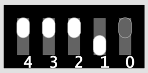
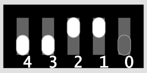
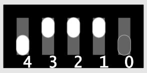
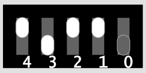
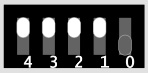
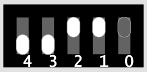
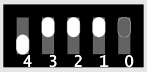
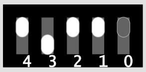
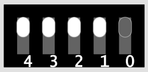
```
Peripheral myC As ColorRGB @ 0                  '   設定模組編號為 0

Dim g_wC, g_wR, g_wG, g_wB As Word              '   設定感測值儲存參數

Sub Main()
    Dim bStatus As Byte                         '   設定狀態儲存參數
    Dim cTemp As Byte                           '   設定暫存字元

    Debug CLS, "點亮紅光"
    myC.SetLEDRGB(20, 0, 0)                     '   設定只點亮 LED 的紅光
    myC.TurnOnLED()                             '   點亮 LED
    Pause 3000                                  '   暫停兩秒
    myC.TurnOffLED()                            '   關閉 LED

    Debug CLS, "點亮綠光"
    myC.SetLEDRGB(0, 20, 0)                     '   設定只點亮 LED 的綠光
    myC.TurnOnLED()                             '   點亮 LED
    Pause 3000                                  '   暫停兩秒
    myC.TurnOffLED()                            '   關閉 LED

    Debug CLS, "點亮藍光"
    myC.SetLEDRGB(0, 0, 20)                     '   設定只點亮 LED 的藍光
    myC.TurnOnLED()                             '   點亮 LED
    Pause 3000                                  '   暫停兩秒
    myC.TurnOffLED()                            '   關閉 LED
    Debug CLS

    myC.LoadLED()                               '   調整 LED 的光強度值為預設值

    myC.SetPeriod(0)                            '   設定量測時間
    myC.SetGain(0)                              '   設定量測倍率
    myC.SetPrescalar(0)                         '   設定量測值縮小量

    myC.DisValReadyEvent()                      '   關閉感測值更新事件
    myC.StartMeasure()                          '   開始量測顏色值

'----------------------------------------------------------------------------
'   Do 迴圈根據狀態值判斷量測是否結束
'----------------------------------------------------------------------------
    Do
        bStatus = myC.GetCRGB(g_wC, g_wR, g_wG, g_wB)     '   讀取感測的顏色值
```

```
    Loop Until bStatus=1

    Debug "C: ", %DEC5R g_wC, CR                   '    顯示全光量
    Debug "R: ", %DEC5R g_wR, CR                   '    顯示紅光量
    Debug "G: ", %DEC5R g_wG, CR                   '    顯示綠光量
    Debug "B: ", %DEC5R g_wB, CR                   '    顯示藍光量

    Debugin "輸入任意直開始重複偵測", %CHR cTemp
    Debug CSRXY(1, 5), CLREOL
    myC.EnValReadyEvent()                          '    啟動感測值更新事件
    myC.StartMeasure()                             '    開始量測顏色值

    Do
        Debug CSRXY(4, 1), %DEC5R g_wC             '    顯示全光量
        Debug CSRXY(4, 2), %DEC5R g_wR             '    顯示紅光量
        Debug CSRXY(4, 3), %DEC5R g_wG             '    顯示綠光量
        Debug CSRXY(4, 4), %DEC5R g_wB             '    顯示藍光量
    Loop
End Sub

Event myC.ValReadyEvent()
    myC.GetClear(g_wC)                             '    取得全光量
    myC.GetRed(g_wR)                               '    取得紅光量
    myC.GetGreen(g_wG)                             '    取得綠光量
    myC.GetBlue(g_wB)                              '    取得藍光量
    myC.StartMeasure()                             '    開始量測顏色值
End Event
```

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