

利基 RF Commander

無線雙向傳輸模組

版本: V1.0



產品介紹: 利基 RF Commander 模組，整合了

兩個 RF24G 與 BC1 的部分功能，提供使用者簡單易用的無線雙向傳輸，透過 USB 直接與電腦連接，就能以單一指令直接傳送多種格式的資料，透過軟體的動態設定，可以隨時切換收發模式，變更傳輸頻道與識別碼。適合讓使用者在 PC 上直接傳輸資訊給遠端的套件。

應用方向:

- 無線傳輸各種資料。
- 傳送控制訊號，達到無線遙控的功能。
- 直接與 NB 結合，成為可隨時更改控制程式的遠端控制中心。

產品特色:

- 無線傳輸頻率範圍: 2.4 ~ 2.524 GHz。
- 無線傳輸模式: GFSK。
- 能隨時更換模組狀態為接收或發送模式。
- 可以軟體動態切換 125 個頻道。
- 輸出功率: 0 dBm。
- 資料傳輸速率: 250 Kbps。
- 無線傳輸範圍可達約 280 公尺。
- 內建天線無須再外接其他天線。
- 提供 256 組 ID 碼與 Reg 碼讓使用者可做動態設定識別，可以隨時透過軟體更換。
- 可以將想要傳輸的資料先儲存到內建暫存空間，再用指令一次傳送，最多可以儲存 40 Bytes 的資料。
- 簡易的變數傳輸指令，Byte，Word，與 Dword 都可以透過單一指令傳送。
- 提供字串與陣列傳輸指令，可以一次最多傳送 20 個字元或 20 Bytes 長度的陣列。
- 量測提醒事件，啟動後每當更新量測值，就會產生提醒事件。
- 設定傳送完成提醒事件，可以在資料傳送完成即時。
- 設定接收完成提醒事件，每當有新資料接收完成就會產生提醒事件。
- 四段傳送強度可動態調整: -20 dBm，-10 dBm，-5 dBm，0 dBm。
- 透過指令可以隨時讀回目前設定值做狀態確認，以及是否有未讀取的接收資料。
- 電源由 USB 主控端提供，不需要連接額外的電源輸入。

產品規格:

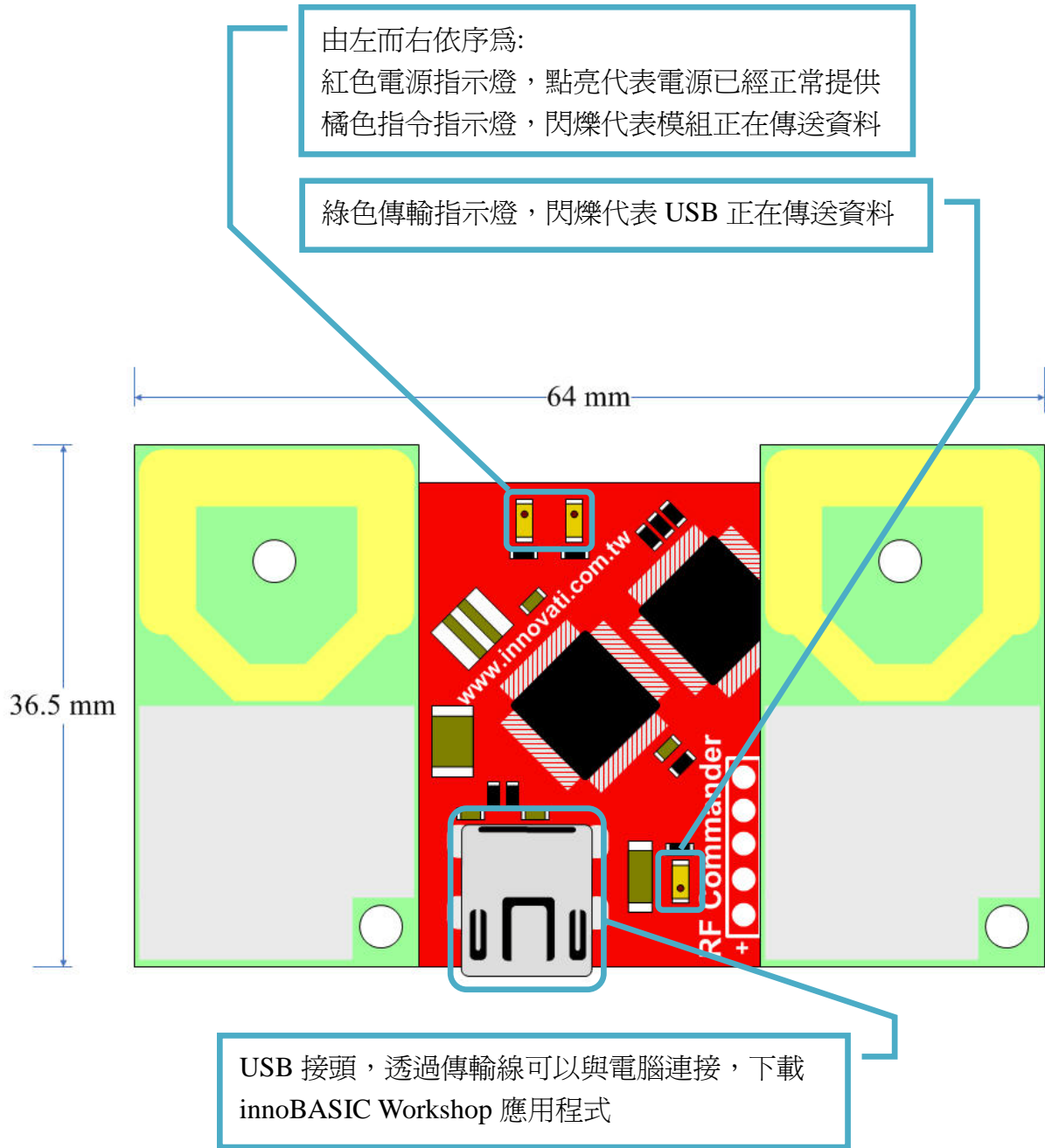


圖 1: 模組腳位與開關介紹

操作注意事項:

- 無線傳輸時，請不要碰觸天線以免影響資料收送。

模組操作溫度 0 °C ~ 70 °C

模組儲存溫度 -40 °C ~ 85 °C

指令表:

下面的指令表是專供控制 RF24G 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 RF24G 指令前，請先於程式開頭定義對應參數與編號，例:

Peripheral *ModuleName* As RF24G @ *ModuleID* ‘ *ModuleID* 固定為 2 或 3

指令格式	指令功能
資料收送相關指令…傳送模式	
SendVar (<i>Data</i>)	傳送 <i>Data</i> 的資料， <i>Data</i> 可以為任意參數
SendArray (<i>Array</i>)	傳送 <i>Array</i> 的資料， <i>Array</i> 為一個陣列參數，陣列的總 Byte 數須小於等於 20 Bytes
SendString (<i>String</i>)	傳送 <i>String</i> 的資料， <i>String</i> 為一個字串參數，字串的總字數須小於等於 20 個字元
BufferVar (<i>Data</i>)	將 <i>Data</i> 儲存到傳送資料暫存記憶體中， <i>Data</i> 可以為任意參數 * 1
BufferArray (<i>Array</i>)	將 <i>Array</i> 儲存到傳送資料暫存記憶體中， <i>Array</i> 為一個陣列參數，陣列的總 Byte 數須小於等於 20 Bytes * 1
BufferString (<i>String</i>)	將 <i>String</i> 儲存到傳送資料暫存記憶體中， <i>String</i> 為一個字串參數，字串的總字數須小於等於 20 個字元 * 1
SendBuffer ()	將儲存於傳送資料暫存記憶體的所有資料，一次傳送出去
資料收送相關指令…接收模式	
GetVar (<i>Data</i>)	取得接收資料暫存記憶體中的資料，儲存於 <i>Data</i> 中，需要根據傳送端設定的參數格式，將 <i>Data</i> 設定為對應的參數格式
GetArray (<i>Array</i>)	取得接收資料暫存記憶體中的資料，儲存於 <i>Array</i> 中，需要根據傳送端設定的陣列長度與參數格式，將 <i>Array</i> 設定為對應的陣列值
GetArray (<i>String</i>)	取得接收資料暫存記憶體中的資料，儲存於 <i>String</i> 中，需要根據傳送端設定的字串長度，將 <i>String</i> 設定為對應的長度值
資料收送相關指令…收送共用	
GetStatus (<i>Status</i>)	發送模式 <i>Status</i> = 0: 待命傳送狀態，模組此時可以接受各種傳送指令 <i>Status</i> = 1: 資料傳送狀態，模組此時僅能接受模式設定指令，不能執行傳送指令，也不能執行 Config 更改模式或狀態

	<p>接收模式</p> <p>Status = 0: 接收資料暫存記憶體中沒有收到新的資料</p> <p>Status = 1~40: 代表接收資料暫存記憶體中的資料筆數，亦即 1 代表有一筆資料，最多會有 40 筆資料儲存在記憶體中，若是在有資料未讀取的狀態下，又收到新資料，則就資料會被清空，若是有啟動 DataLostEvent，也會產生 DataLostEvent 提醒</p>
ClrBuffer(Status)	<p>發送模式</p> <p>清除所有傳送資料暫存記憶體中的資料</p> <p>接收模式</p> <p>清除所有接收資料暫存記憶體中的資料，使用 GetStatus 會讀回 0，此時不會產生 DataLostEvent 提醒事件</p>
收送狀態設定相關指令	
SetMode(Mode)	<p>根據 Mode 值設定模組為發送或接收模式</p> <p>Mode = 0 → 設定模組為發送模式</p> <p>Mode = 1 → 設定模組為接收模式</p> <p>預設值為 1 (接收模式) * 2</p>
GetMode(Mode)	<p>取得現在設定的模式值放於參數 Mode 中</p> <p>Mode = 0 → 設定模組為接收模式</p> <p>Mode = 1 → 設定模組為發送模式</p>
SetCh(Channel)	<p>根據 Channel 值設定模組為所使用的頻道，Channel 可以設定為 0~124 之件的整數值，預設值為 0 * 2</p>
GetCh(Channel)	<p>取得現在模組設定的頻道放於參數 Channel 中，Channel 會回傳 0~124 之間的整數</p>
SetRFID(ID)	<p>根據 ID 值設定模組所使用的識別碼，ID 可以設定為 0~255 之間的整數值，預設值為 0 * 2</p>
GetRFID(ID)	<p>取得現在模組設定的識別碼放於參數 ID 中，ID 會回傳 0~255 之間的整數</p>
SetRegCode(Reg)	<p>根據 Reg 值設定模組所使用的註冊碼，Reg 可以設定為 0~255 之間的整數值，預設值為 0 * 2</p>
GetRegCode(Reg)	<p>取得現在模組設定的註冊碼放於參數 Reg 中，Reg 會回傳 0~255 之間的整數</p>
Config()	<p>將設定的模式，頻道，識別碼與註冊碼，下載到模組中 * 2</p>
SetPower(Power)	<p>根據 Power 值設定模組發送時的功率，Power 可以設定為 0~3 之間的整數值</p> <p>0: -20 dBm</p>

	1: -10 dBm 2: -5 dBm 3: 0 dBm 預設值為 3
GetPower(Power)	取得現在模組設定的發送功率值放於參數 Power 中， Power 會回傳 0~3 之間的整數
SaveConfig(Num)	將頻道，識別碼與註冊碼，儲存於 Num 指定的位置， Num 可以輸入 1~10 之間的整數
LoadConfig(Num)	由 Num 指定的位置，讀取儲存的的頻道，識別碼與註冊碼值， Num 可以輸入 0~10 之間的整數，0 為回復預設值
提醒事件相關指令	
EnDataLostEvent()	啟動資料遺失事件提醒，執行後在資料遺失時會產生 DataLostEvent ，預設為啟動 * 3
DisDataLostEvent()	關閉資料遺失事件提醒，執行後在資料遺失時不會產生 DataLostEvent ，預設為啟動 * 3
EnTxReadyEvent()	啟動傳送完成事件提醒，執行後在資料傳送完成時會產生 TxReadyEvent ，預設為關閉
DisTxReadyEvent()	關閉傳送完成事件提醒，執行後在資料遺失時不會產生 TxReadyEvent ，預設為關閉
EnRxReadyEvent()	啟動接收完成事件提醒，執行後在接收到新資料時會產生 RxReadyEvent ，預設為關閉
DisRxReadyEvent()	關閉接收完成事件提醒，執行後在接收到新資料時不會產生 RxReadyEvent ，預設為關閉
EnBufferFullEvent()	啟動傳送資料暫存記憶體存滿事件提醒，執行後在傳送資料暫存記憶體存滿時，又再下達 Buffer 相關指令，就會產生 BufferFullEvent ，預設為啟動
DisBufferFullEvent()	關閉傳送資料暫存記憶體存滿事件提醒，執行後在傳送資料暫存記憶體存滿時，又再下達 Buffer 相關指令，不會產生 BufferFullEvent ，預設為啟動
EnRxErrorEvent()	啟動接收資料錯誤事件提醒，執行後，在接收到內部判斷為 CRC 或是傳輸格式錯誤的資料時，就會產生 RxErrorEvent ，預設為關閉
DisRxErrorEvent()	關閉接收資料錯誤事件提醒，執行後，在接收到內部判斷為 CRC 或是傳輸格式錯誤的資料時，不會產生 RxErrorEvent ，預設為關閉

* 1 傳送資料暫存記憶體最多可儲存 40 Bytes 的資料，當儲存滿 40 Bytes 後再執行 **Buffer** 相關的指令，會被視為無效指令

* 2 發送接收模式，頻道，識別碼與註冊碼，在設定後並不會立刻更新，而是要執行 Config 指令後，才會一次更新此四項資料並啓動設定值

* 3 此處的資料遺失，是指接收的資料尚未被讀取，又有接收到新的資料，此時原先在接收資料暫存記憶體中的資料會被清除，而以新的資料取代

模組提供應用事件:

事件名稱 (Event)	啓動條件
DataLostEvent	在執行 EnDataLostEvent () 後，在接收模式，當偵測到接收資料暫存記憶體中資料未被讀取，就被新資料覆蓋時，就會產生
TxReadyEvent	在執行 EnTxReadyEvent () 後，在發送模式，當偵測到資料傳送完畢時，就會產生
RxReadyEvent	在執行 EnRxReadyEvent () 後，在接收模式，當接收資料暫存記憶體中的資料已經被讀取過，又接收到新資料時，就會產生
BufferFullEvent	在執行 EnBufferFullEvent () 後，在發送模式，當偵測到傳送資料暫存記憶體內已有 40 Bytes 的資料，又執行 Buffer 相關指令，就會產生
RxErrorEvent	在執行 EnRxErrorEvent () 後，在接收模式，接收到的資料判斷為 CRC 錯誤或格式錯誤時，就會產生

範例程式:

```
'=====
'===          利基應用科技          ===
'===          RF Commander 範例程式          ===
'=====

Peripheral myT As RF24G @ 2          ' 設定模組編號 2
Peripheral myR As RF24G @ 3          ' 設定模組編號 3

Dim g_TxReady As Byte                ' 宣告傳送完成參數
Dim g_RxReady As Byte                ' 宣告接收完成參數

'=====
' 主程序
'=====

Sub Main()
    Dim bTxData, bRxData As Byte      ' 宣告發射與接收參數

    Debug CLS                          ' 清除終端視窗文字
    ConfigInitVal()                    ' 呼叫初始值設定程序
    myT.Config()                       ' 將設定功能寫入發射模組
    myR.Config()                       ' 將設定功能寫入接收模組
    Pause 1000
    Debug CSRXY(1, 1), "TX: "
    Debug CSRXY(1, 2), "RX: "

'-----
' For 迴圈會執行十次單一參數的傳送
'-----

    For bTxData=1 To 10
        myT.SendVar(bTxData)          ' 送出資料

'-----
' Do 迴圈等待資料送出
'-----

        Do
            Loop Until g_TxReady=1

            Debug CSRXY(5, 1), %DEC3R bTxData ' 顯示傳送值

'-----
' Do 迴圈等待資料接收
'-----
```

```

Do
    Loop Until g_RxReady=1

    myR.GetVar(bRxData)          ' 接收資料
    Debug CSRXY(5, 2), %DEC3R bRxData ' 顯示接收值
Next

Debug CR, CR, "傳送完成"
End Sub

'=====
' 初始值設定程序
'=====

Sub ConfigInitVal()
    myT.EnTxReadyEvent()        ' 啓動傳送完成事件
    myT.SetMode(0)              ' 設定爲傳送模式
    myT.SetCh(0)                ' 設定傳送頻道 0

    myR.EnRxReadyEvent()       ' 啓動接收完成事件
    myR.SetMode(1)              ' 設定爲接收模式
    myR.SetCh(0)                ' 設定接收頻道 0
End Sub

'=====
' 傳送完成事件
'=====

Event myT.TxReadyEvent()
    g_TxReady = 1
End Event

'=====
' 接收完成事件
'=====

Event myR.RxReadyEvent()
    g_RxReady = 1
End Event

```


附錄

1. 已知問題: