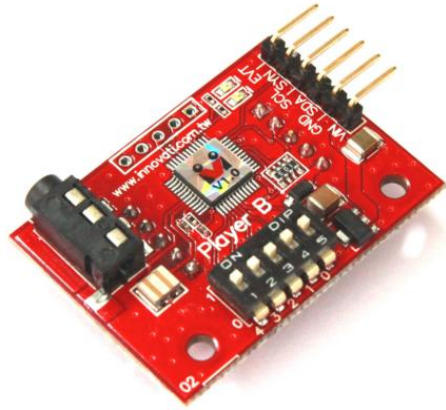


Player B

MP3 播放模組

版本: V2.0



產品介紹: 利基 Player B 模組可以分析 Micro SD 卡中的檔案，判斷 MP3 檔案並播放。由 BASIC Commander[®]傳送簡易操作指令，可以隨時播放或暫停音樂，也能指定曲目，需要 MP3 播放功能時是極佳的選擇。

應用方向:

- 接上喇叭，配合 BASIC Commander[®]外接按鍵功能，就可以播放想操作的音樂，成為小型收音機。
- 組合在機器人套件上，將語音錄成 MP3，能讓機器人套件有發聲的功能。
- 組合多個模組，可以選擇更多播放曲目，做出多片卡槽的音樂庫效果。

產品特色:

- 具有 Micro SD 卡插槽，能讀取最多 32G 的 Micro SD 卡資料內容。
- 以檔案名稱做為指標，最多可以判斷至 9999 個。
- 提供 30 階音量控制。
- 五種 EQ(等化器)設定。
- 可以設定自動重複播放開關，有單首重複播放與全資料夾重複播放兩種選項。
- 上一首與下一首快速跳首指令。
- 由讀取狀況指令，能取得播放狀況或是 Micro SD 的狀態。
- 提供 3.5mm 音源輸出孔，能直接接上小型喇叭播放，提供約 46mW 的功率。
- 可透過 I2C 方式，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。Micro SD 卡須安裝於背面的卡槽中，請注意卡片有正反面與方向性。

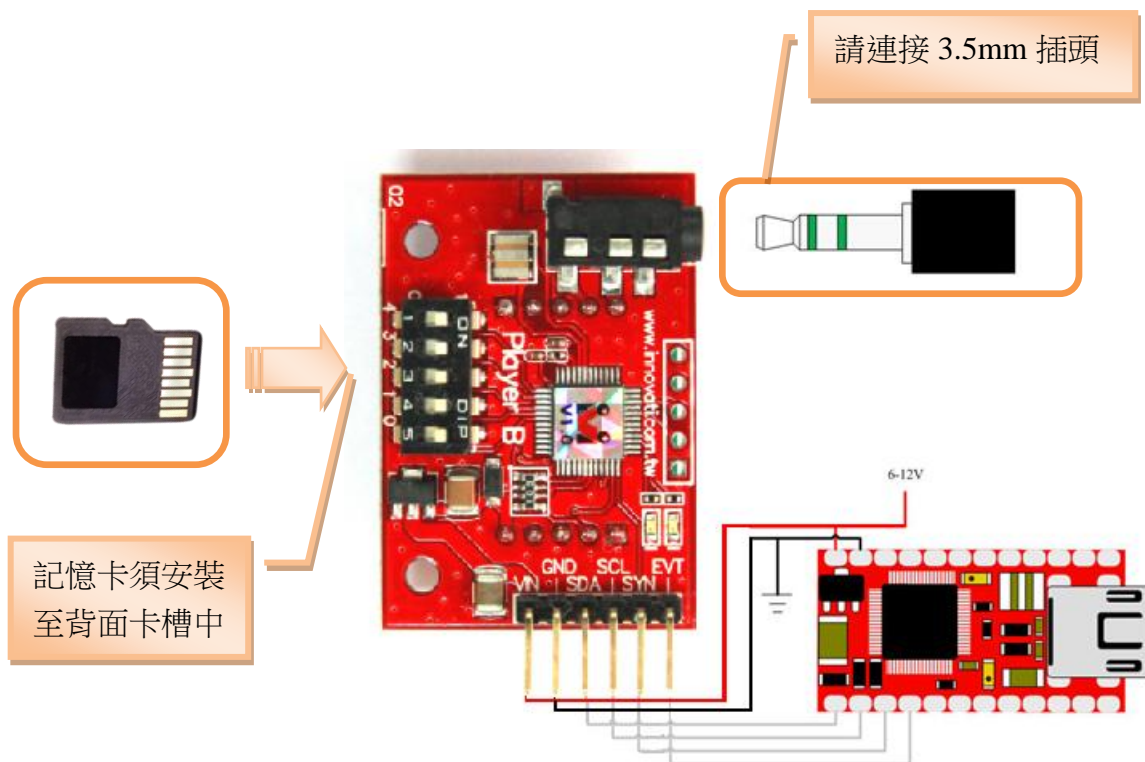


圖 1: 模組連接範例

產品規格:

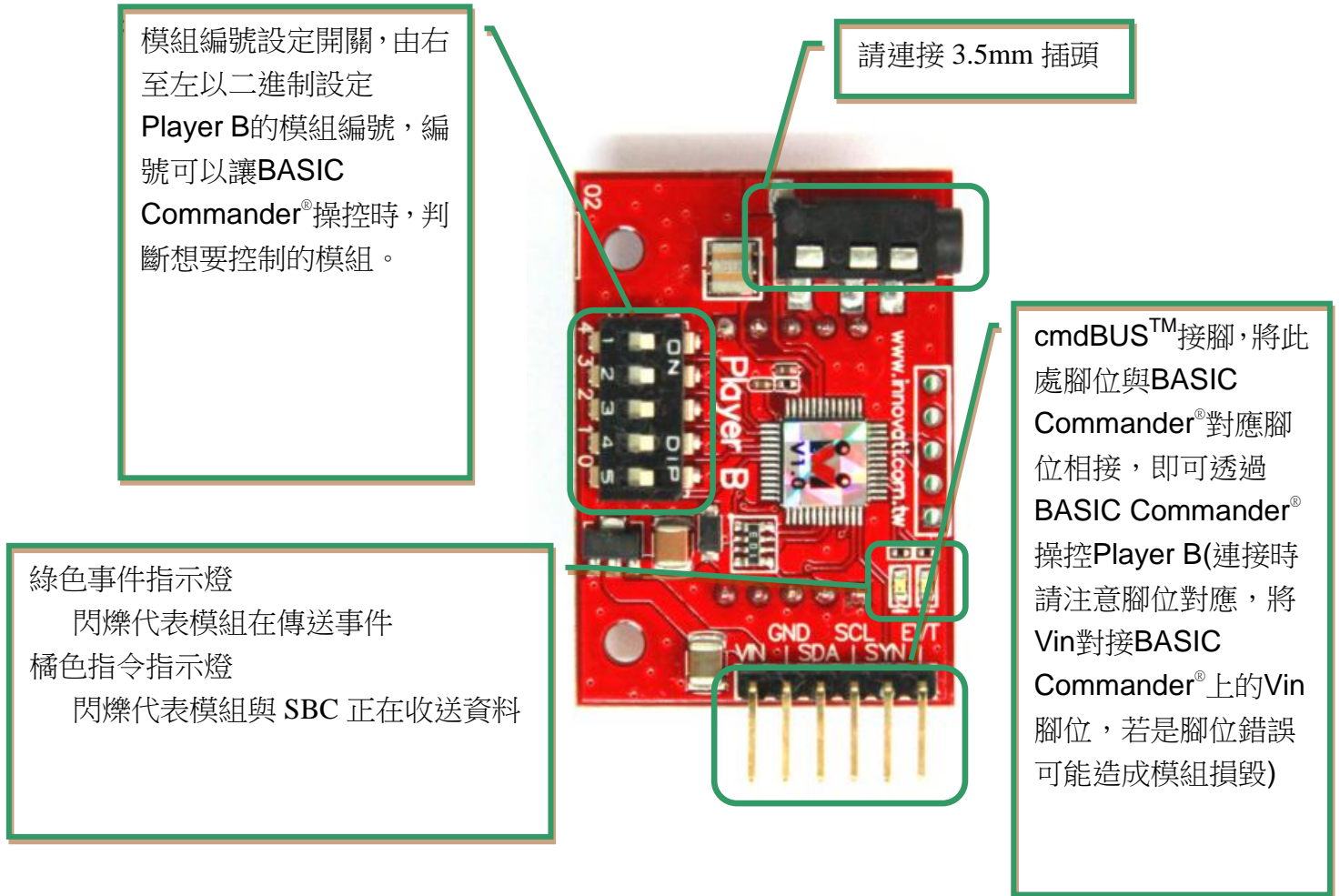


圖 2: 模組腳位與開關介紹

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V _{DD}	Conditions				
I _{DD}	Operating Current	7.5	No I/O	—	30	—	mA

表 1: 工作電流特性 (於 25 °C 之環境)

操作注意事項:

Player B 能讀取 32G 的磁碟空間。最大檔案數為 9999。檔案必須放在記憶卡根目錄的”MP3”資料夾中，檔案名稱的前四個半形字元必須是 0001~9999(檔案名稱不須連續)。錯誤的資料夾名稱、存放位置和錯誤的檔名，都會造成系統無法正常讀取。

SD Card 目錄範例



名稱	修改日期	類型
0001天黑黑	2012/7/31 上午 11:10	MP3 格式聲音
0002BluetoothMode	2012/9/12 下午 03:19	MP3 格式聲音
0003BTMode	2012/9/12 下午 03:19	MP3 格式聲音
0014GoodBye	2012/9/12 下午 03:07	MP3 格式聲音
0205HowAreYouDoing	2012/9/12 下午 03:07	MP3 格式聲音
9006iShow1	2012/9/12 下午 03:07	MP3 格式聲音
9207iShow2	2012/9/12 下午 03:07	MP3 格式聲音

模組操作溫度 0 °C ~ 70 °C (記憶卡的操作溫度需另外確認)

模組儲存溫度 -50 °C ~125°C

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 Player B 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 Player B 指令前，請先於程式開頭定義對應參數與編號，例:

Peripheral *ModuleName* As PlayerB @ *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： $255 - (Checksum1 \sim Checksum2 \text{ 之間的變數總和})$

Checksum3：驗證位元_3，空間大小為 Byte 的變數。

定義方式： $255 - MID - (MID \sim Checksum3 \text{ 之間的變數總和})$

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:

$$\text{MID} = 2$$

$$\text{CID} = 153$$

$$\text{Checksum1} = 255 - (2*2) - 153 = 98$$

$$\text{Byte} = 100$$

$$\text{Checksum2} = 255 - 100$$

$$\text{Dummy} = 0\sim 255 \text{ 之間的任意數}$$

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:

$$\text{MID} = 2$$

$$\text{CID} = 153$$

$$\text{Checksum1} = 255 - (2*2) - 153 = 98$$

$$\text{Temp_L} = 255, \text{Temp_H} = 1$$

$$\text{Checksum2} = 255 - \text{Temp_L} - \text{Temp_H} = 255$$

$$\text{Dummy} = 0\sim 255 \text{ 之間的任意數}$$

指令格式	指令功能
檔案播放相關指令	
CmdBUS : Play()	播放檔案中的曲目,會直接從頭播放目前設定編號的歌曲,若是原先為暫停狀態,則會從暫停的時間繼續播放。
I2C : MID+91+Checksum1+Dummy	
CmdBUS : Pause()	暫停播放中的歌曲,再執行 Play 指令播放時,會從暫停的時間繼續播放。
I2C : MID+92+Checksum1+Dummy	
CmdBUS : Stop()	停止播放中的歌曲,再執行 Play 指令播放時,會從歌曲開頭重新播放。
I2C : MID+93+Checksum1+Dummy	
CmdBUS : Forward()	跳至下一首曲目。
I2C : MID+100+Checksum1+Dummy	
CmdBUS : Backward()	跳至前一首曲目。

I2C : MID+99+Checksum1+Dummy	
CmdBUS : SetPlayNum(Num)	根據 Num 的設定值，設定要播放的曲目。以檔案名稱的前四個半形字元做為指標，最多可指定 0001~9999 個檔案。如
I2C : MID+122+Checksum1 +Num_L+Num_H +Checksum2+Dummy	Num=1 則播放 MP3 目錄中，檔名前四個字元為 0001 的 MP3 檔(如'0001天黑黑.MP3') Num=205 則播放 MP3 目錄中，檔名前四個字元為 0205 的 MP3 檔(如'0205HowAreYouDoing.MP3') Num=9207 則播放 MP3 目錄中，檔名前四個字元為 9207 的 MP3 檔(如'9207iShow2.MP3')
CmdBUS : VolUp()	
I2C : MID+95+Checksum1+Dummy	將音量調大一階，如果現在音量為最大，則不會動作。
CmdBUS : VolDown()	
I2C : MID+96+Checksum1+Dummy	將音量調小一階，如果現在音量為最小，則不會動作。
CmdBUS : Mute(Mode)	
I2C : MID+103+Checksum1 +Mode+Checksum2+Dummy	根據 Mode 設定啟動靜音或關閉靜音。 0: 關閉靜音，依照設定音量繼續播放。 1: 啟動靜音。
模式設定相關指令	
CmdBUS : SetVol(Vol)	
I2C : MID+97+Checksum1 +Vol+Checksum2+Dummy	依據 Vol 的數值，設定目前音量， Vol 可以輸入 0~30 之間的整數值。
CmdBUS : SetRepeat(Mode)	根據 Mode 設定重複播放的模式。
I2C : MID+105+Checksum1 +Mode+Checksum2+Dummy	0: 資料夾重複播放，當播放完資料夾中的最後一首，自動從第一首開始播放。 1: 單曲重複播放，重複播放同一首歌曲。 2: 不重複播放。 預設為 0。

<p>CmdBUS : SetEQ(EQ)</p> <hr/> <p>I2C : MID+107+Checksum1 +EQ+Checksum2+Dummy</p>	<p>根據 EQ 設定等化器的風格。</p> <p>0: 關閉等化器</p> <p>1: Pop</p> <p>2: Rock</p> <p>3: Jazz</p> <p>4: Classic</p> <p>5: Bass</p> <p>預設為 0。</p>
<p>SetAutoPlay(Mode)</p>	<p>依據 Mode 的數值，設定自動播放的模式。Mode 可以輸入 0~4 之間的整數值。</p> <p>0: 重新偵測到記憶卡，進入到暫停狀態等待指令。</p> <p>1: 重新偵測到記憶卡，自動播放上次最後曲目，若是歌曲資訊失效，則進入暫停狀態等待指令。</p> <p>2: 重新偵測到記憶卡，自動播放第一首曲目，若是歌曲資訊失效，則進入暫停狀態等待指令。</p> <p>3: 重新偵測到記憶卡，則重新掃描裝置，完成後進入暫停狀態。</p> <p>4: 重新偵測到記憶卡，則重新掃描裝置，完成後則從第一首開始播放。</p> <p>預設為 0</p>
<p>資料讀取相關指令</p>	
<p>CmdBUS : GetTotalNum(Num)</p> <hr/> <p>I2C : Out : MID+121+Checksum1+Dummy In : MID+Num_L+Num_H+Checksum3</p>	<p>取得 SD 卡中的檔案總數(同時包含 MP3 目錄外的其他 MP3 檔案)，存放於 Num 中。</p>
<p>GetPlayNum(Num)</p>	<p>取得目前播放的曲目，存放於 Num 中。0 代表沒有可播放的檔案，或是尚未解析，在解析完成而且可以有播放的檔案時，預設值為 1。</p>
<p>GetVol(Vol)</p>	<p>取得目前的音量設定值，存放於 Vol 中，Vol 會回傳 0~30 之間的整數值。</p>
<p>GetRepeat(Mode)</p>	<p>取得重複播放的模式，存放於 Mode 中。</p> <p>0: 資料夾重複播放，當播放完資料夾中的最後一首，自動從第一首開始播放。</p> <p>1: 單曲重複播放，重複播放同一首歌曲。</p> <p>2: 不重複播放。</p> <p>預設為 0。</p>
<p>GetEQ(EQ)</p>	<p>取得等化器的風格設定，存放於 EQ 中。</p>

	<p>0: 關閉等化器</p> <p>1: Pop</p> <p>2: Rock</p> <p>3: Jazz</p> <p>4: Classic</p> <p>5: Bass</p> <p>預設為 0。</p>
<p>CmdBUS :</p> <p><i>Status = GetSDStatus()</i></p> <hr/> <p>I2C :</p> <p>Out :</p> <p>MID+89+Checksum1+Dummy</p> <p>In :</p> <p>MID+Status+Checksum3</p>	<p>取得記憶卡的狀態，存放於 <i>Status</i> 參數中。 <i>Status</i> 會回傳 0~8 之間的整數值。</p> <p>0: 沒有偵測到記憶卡。</p> <p>1: 記憶卡辨識中。</p> <p>2: 偵測到錯誤裝置。</p> <p>3: 記憶卡中沒有可以播放的檔案。</p> <p>4: 記憶卡為閒置裝置。</p> <p>5: 歌曲停止播放。</p> <p>6: 歌曲暫停。</p> <p>7: 歌曲播放中。</p> <p>8: 讀取到錯誤資訊。</p> <p>安裝記憶卡後，要先確定記憶卡狀態不為 0，才可以執行播放等動作。</p>
<p>GetAutoPlay(<i>Mode</i>)</p>	<p>取得自動播放的設定，存放於 <i>Mode</i> 中。<i>Mode</i> 會回傳 0~4 之間的整數值。</p> <p>0: 重新偵測到記憶卡，進入到暫停狀態等待指令。</p> <p>1: 重新偵測到記憶卡，自動播放上次最後曲目，若是歌曲資訊失效，則進入暫停狀態等待指令。</p> <p>2: 重新偵測到記憶卡，自動播放第一首曲目，若是歌曲資訊失效，則進入暫停狀態等待指令。</p> <p>3: 重新偵測到記憶卡，則重新掃描裝置，完成後進入暫停狀態。</p> <p>4: 重新偵測到記憶卡，則重新掃描裝置，完成後則從第一首開始播放。</p> <p>預設為 0</p>

範例程式:

```

Peripheral myPlayer as PlayerB @ 0      ' 設定模組編號為 0
' 執行程式前，請確認 MP3 目錄中有檔名開頭為 0001 的 MP3 檔
Sub Main()
    Dim bStatus As Byte ' 宣告儲存狀態的參數
'-----
'--- 無窮迴圈等待記憶卡解析完成 ---
'-----


















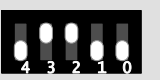














    Do
        bStatus = myPlayer.GetSDStatus()
    Loop Until bStatus>=3

    myPlayer.SetVol(30) ' 設定播放音量
    myPlayer.Play()     ' 開始播放歌曲。
End Sub

```

附錄

1. 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