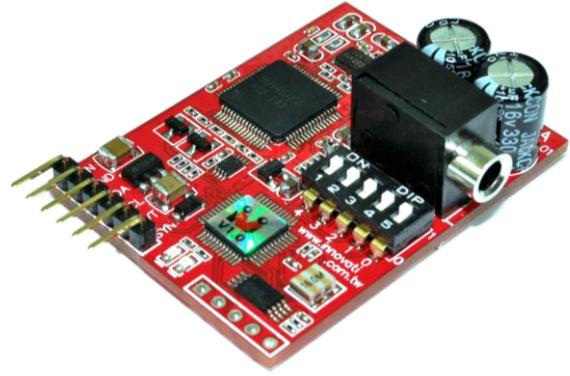


Player A

MP3 播放模組

版本: V2.0



產品介紹: 利基 Player A 模組可以分析 Micro SD 卡中的檔案，判斷 MP3 檔案並播放。由 Arminno 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:

- 接上小型麥克風，配合 Arminno 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 2G 的 Micro SD 卡資料內容。
- 最大排序檔案可以判斷至 1000 個。
- 提供 32 階音量控制。
- 十二種 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 cmdBUS 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 Arminno 上對應的腳位，就可透過 Arminno 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。

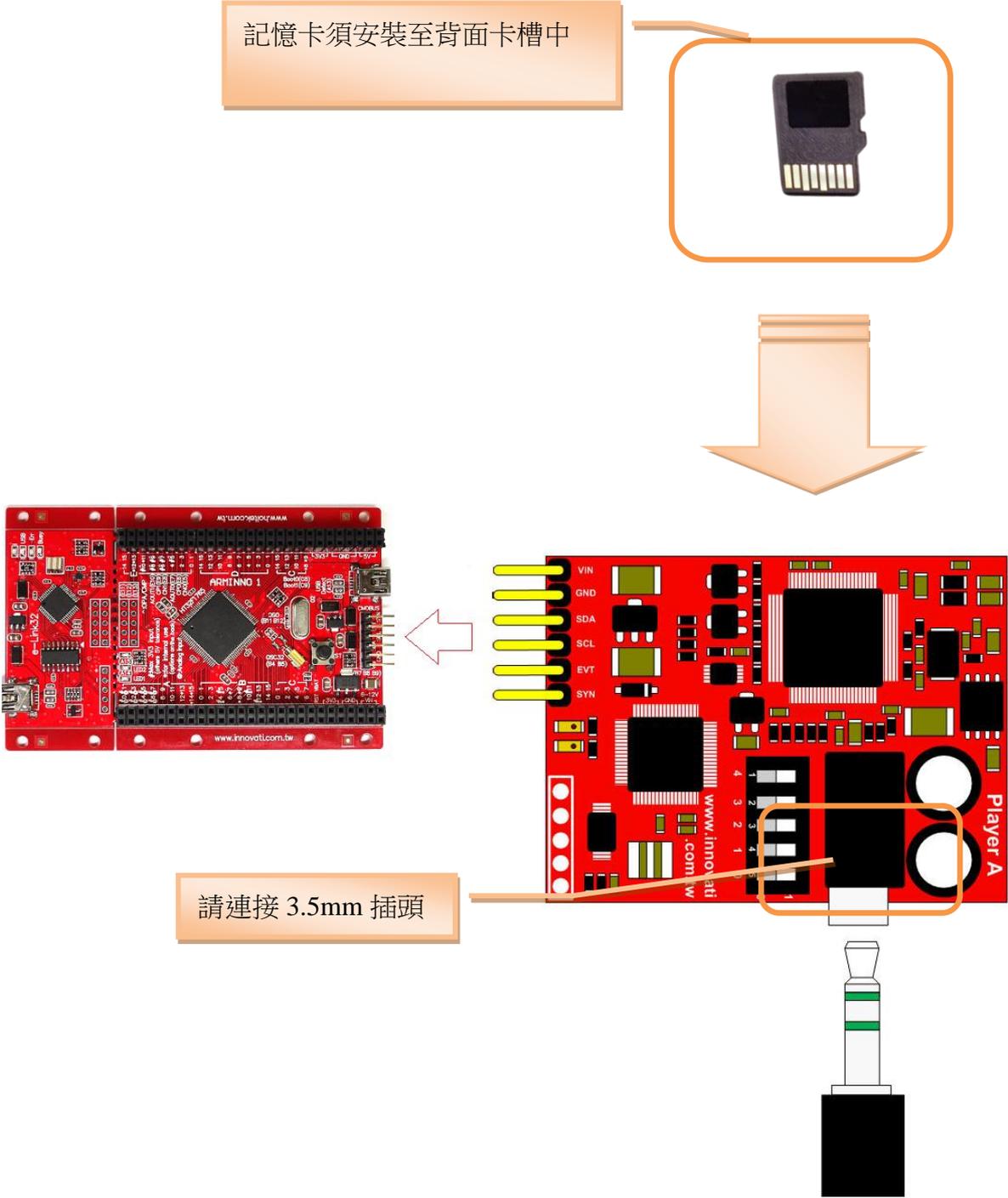


圖 1: 模組連接範例

操作注意事項:

Player A 僅能讀取 2G 的磁碟空間，如果使用超過 2G 的記憶卡，可能只能解析前半段的磁區，放在後半段的檔案將無法判別。最大排序檔案數為 1000，如果存放了超過 1000 首曲目，也只會顯示 1000 首。檔案必須放在記憶卡跟目錄的”InnovatiMusic”資料夾中，錯誤的資料夾名稱與存放位置，都會造成系統無法正常讀取。

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

指令格式	指令功能
檔案播放相關指令	
Play(void)	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。
Pause(void)	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。
Stop(void)	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。
Forward(void)	跳至下一首曲目。
Backward(void)	跳至前一首曲目。
SetPlayNum(uint16_t Num)	根據 <i>Num</i> 的設定值，設定要播放的曲目。
VolUp(void)	將音量調大一階，如果現在音量為最大，則不會動作。
VolDown(void)	將音量調小一階，如果現在音量為最小，則不會動作。
Mute(uint8_t Mode)	根據 <i>Mode</i> 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。
AnalSD(void)	解析記憶卡內容，只有在記憶卡已經安裝至模組，並且可以辨識，此指令才有效。執行後會開始解析記憶卡中的 MP3 檔案數量，並自動排序儲存相關位置。指令解析中無法執行播放等相關動作，請以 GetSDStatus 指令確認是否解析完成。
模式設定相關指令	
SetVol(uint8_t Vol)	依據 <i>Vol</i> 的數值，設定目前音量， <i>Vol</i> 可以輸入 0~31 之間的整數值。
SetRepeat(uint8_t Mode)	根據 <i>Mode</i> 設定重複播放的模式。 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。
SetEQ(uint8_t EQ)	<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: Jazz</p> <p>3: Rock</p> <p>4: Classic</p> <p>5: R&B</p> <p>6: Bass boost</p> <p>7: Pop + Bass</p> <p>8: Jazz + Bass</p> <p>9: Rock + Bass</p> <p>10: Classic + Bass</p> <p>11: R&B + Bass</p> <p>12: Bass boost2</p> <p>預設為 0。</p>
資料讀取相關指令	
GetTotalNum(uint16_t &Num)	<p>取得可以解析到的檔案總數，存放於 Num 中。Num 會回傳 0~1000 間的整數值。</p>
uint8_t Status = GetSDStatus(void)	<p>取得記憶卡的狀態，存放於 Status 參數中。Status 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>

範例程式:

```
#include "arminno.h"
```

```
PlayerA myPlayer(4);
```

```
int main(void)
```

```
{
```

```
    uint8_t bStatus;    // 宣告儲存狀態的參數
```

```
    //-----
```

```
    //--- 無窮迴圈等待記憶卡插上    ---
```

```
    //-----
```

```
    do {
```

```
        bStatus = myPlayer.GetSDStatus();
```

```
    } while(bStatus <= 1);
```

```
    myPlayer.AnalSD();                // 開始解析記憶卡
```

```
    //-----
```

```
    //--- 無窮迴圈等待記憶卡解析完成    ---
```

```
    //-----
```

```
    do {
```

```
        bStatus = myPlayer.GetSDStatus();
```

```
    } while(bStatus <= 1);
```

```
    myPlayer.SetVol(15);    // 設定播放音量
```

```
    myPlayer.Play();        // 開始播放歌曲
```

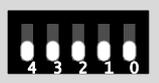
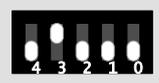
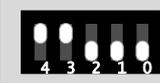
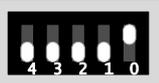
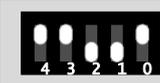
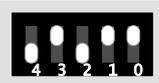
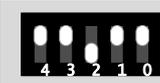
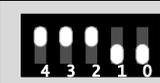
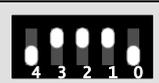
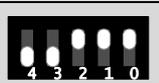
```
}
```

附錄

1. 已知問題:

- 使用時請以較大音量播放，並使用可調音量的喇叭，以免將雜訊放大。

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