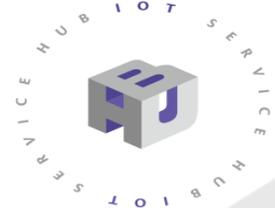


國產IC開發套件 Filogic 130 (MT7933) 示範案例

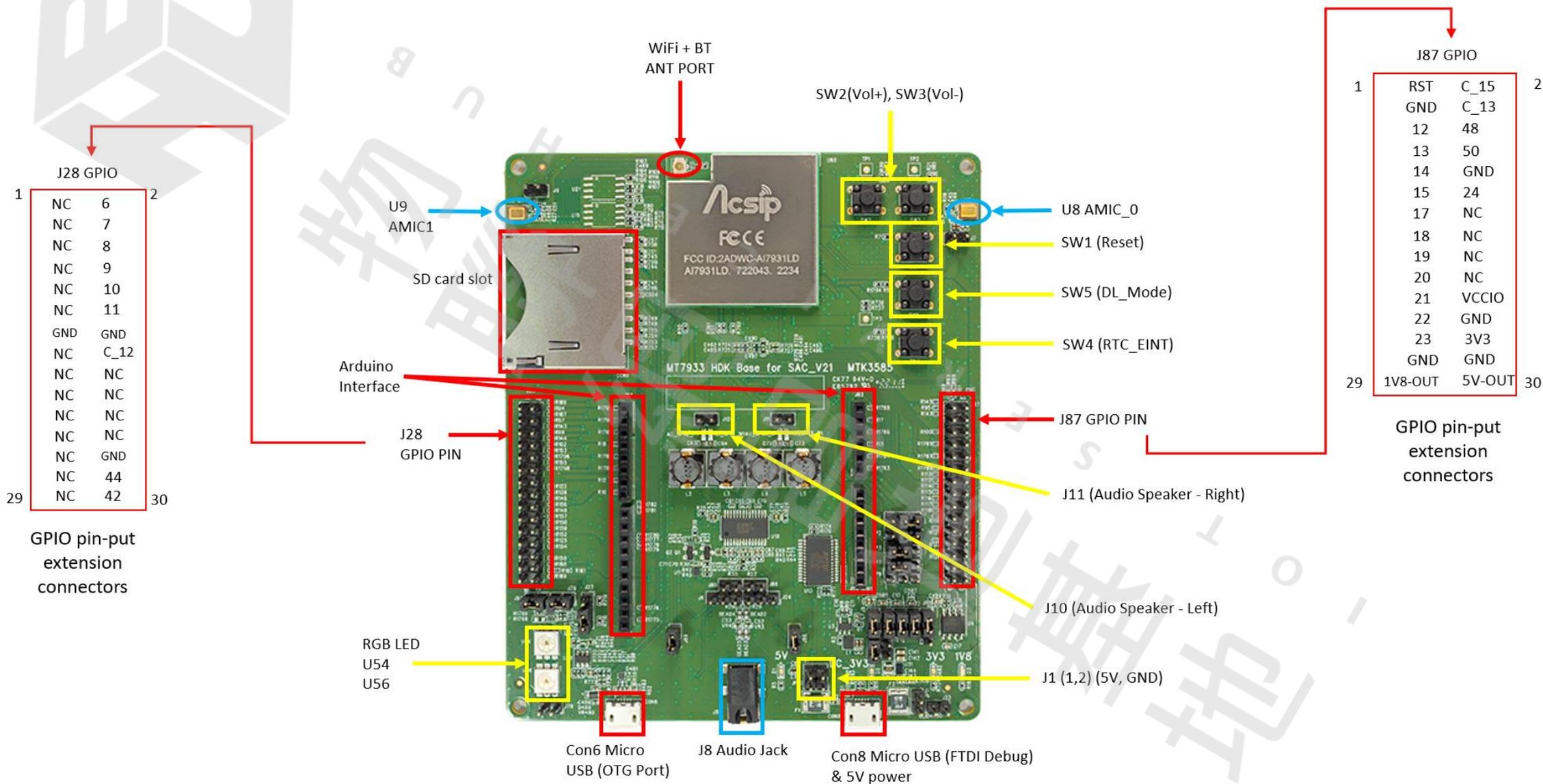
SDIO SD Card+ I2C LCD 1602 + Speaker

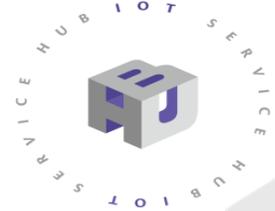


大綱

- 開發板Filologic130 介紹
- 周邊介面與範例說明：
 - ◆ SDIO SD Card (外接SD卡槽)
 - ◆ I2C LCD 1602

開發板硬體外觀





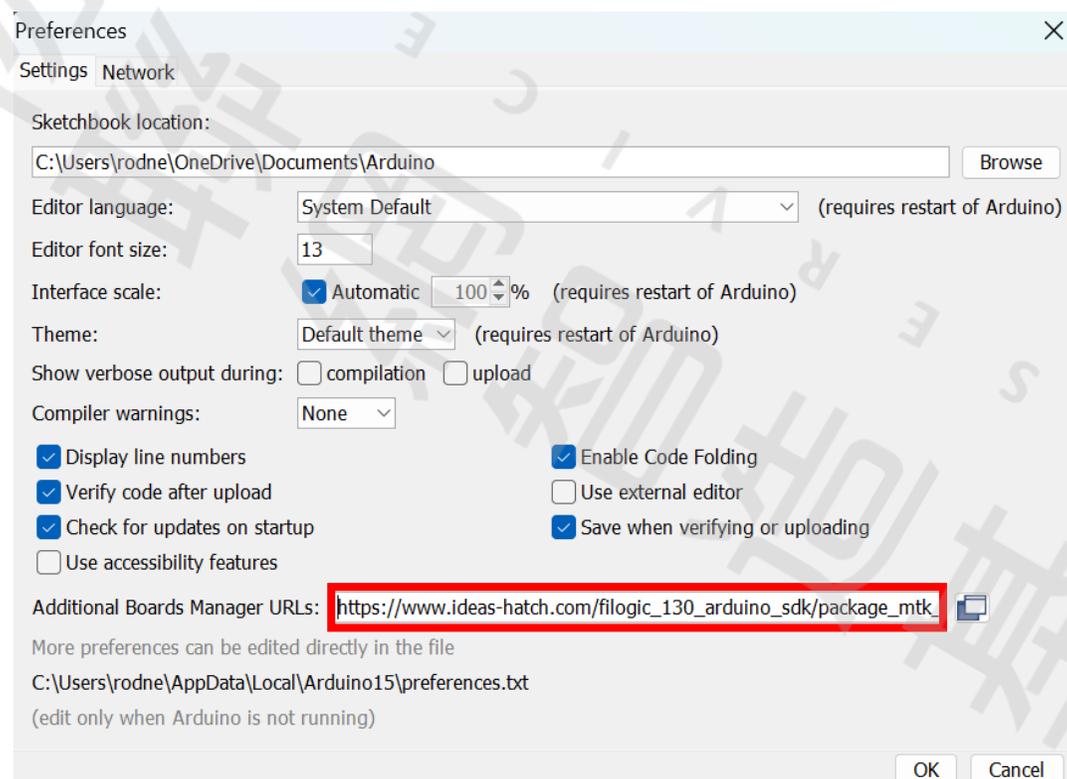
Filogic SDK 下載

- Filogic130 Arduino SDK的各個版本
 - ◆ 從資策會雲端硬碟下載
 - ◆ <https://drive.google.com/drive/folders/1z7HscksgsalRfHW7T49nVbl60gY-weV7>
 - ◆ 在SDK v1.0.0裡面，提供開發板的電路圖

安裝步驟說明 (2)

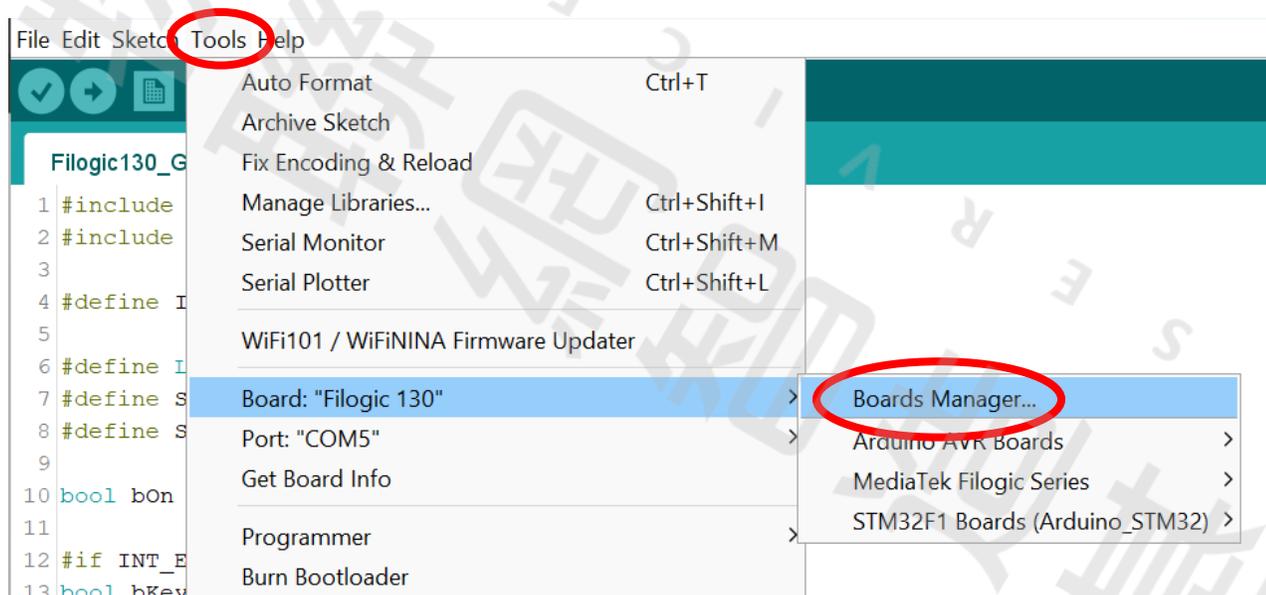
- 安裝IDE後，點選「偏好設定」(Preferences)，並設定以下SDK下載網址

https://www.ideas-hatch.com/filogic_130/arduino_sdk/package_mtk_filogic_130_index.json

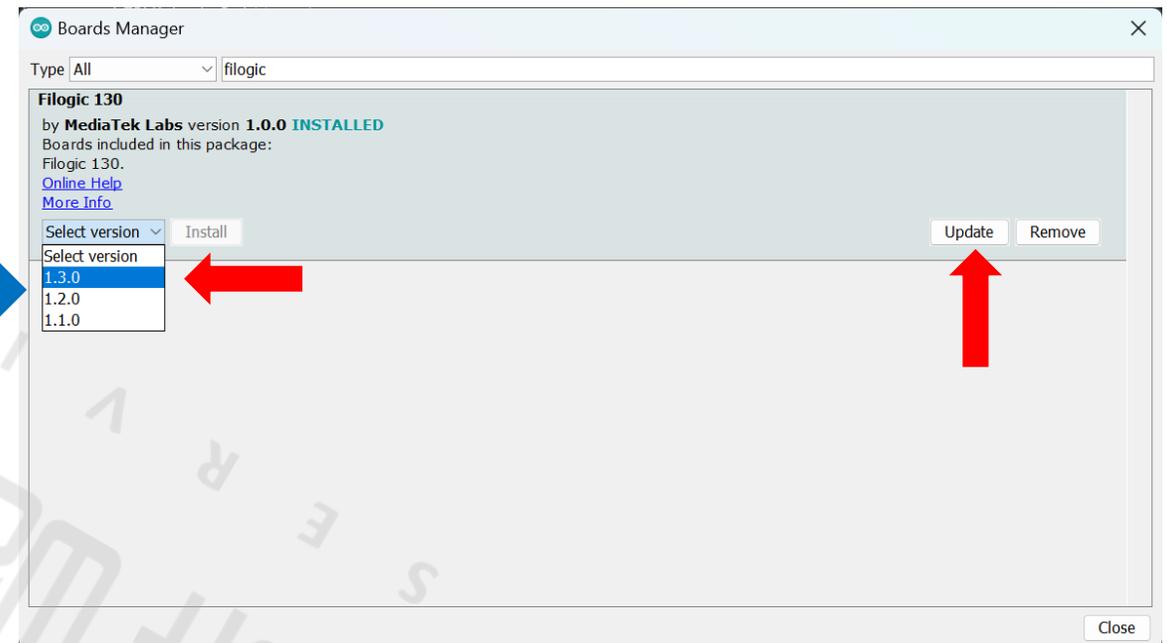
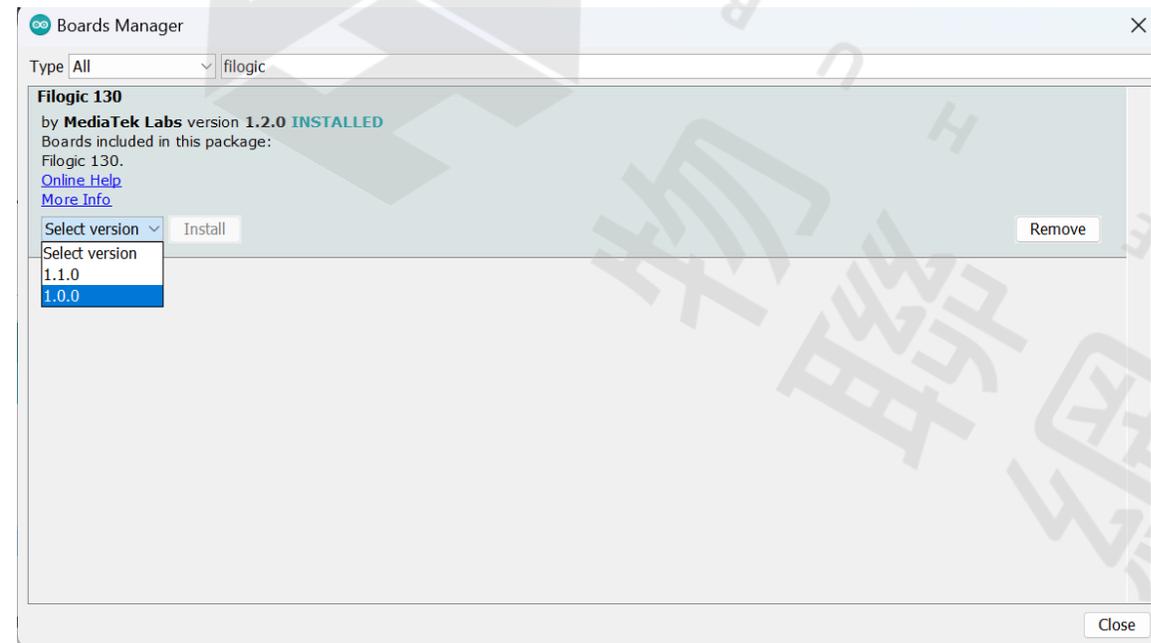
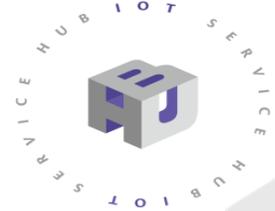


安裝步驟說明 (3)

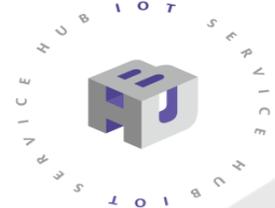
- 開啟IDE的“開發板管理員”(Tool > Board > Boards Manager)
- 安裝Filogic 130開發板的SDK



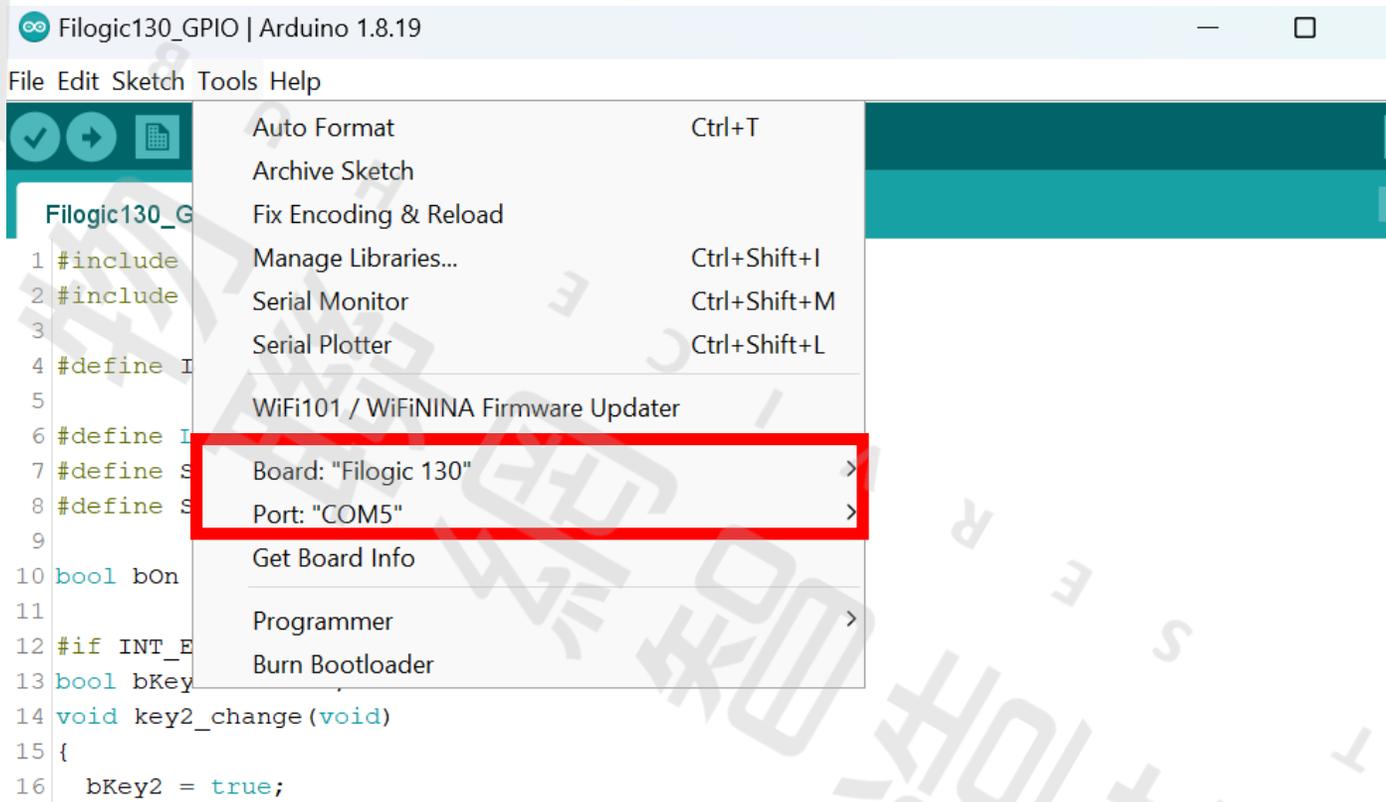
安裝步驟說明 (3)



1. 搜尋Filogic 130
2. 安裝1.0.0版本
3. 再選擇1.3.0進行update



安裝步驟說明 (3)



- 安裝完畢後確認Board 與 Port

安裝步驟說明 (4)

- 下載測試程式 - GPIO

- ◆ https://github.com/yijenlu1971/Filologic_onArduino/tree/main/GPIO

- 留意上傳過程：

- 按著SW1 Reset 鍵，當看到訊息“INFO: Goto open COM X”的時候，再放開SW1 鍵。.....開始上傳，直到看到訊息“Finished!”。
- 再按一次SW1 鍵(Reset)，重啟系統。

- 當系統運行後：

- 開啟監控視窗(Serial Monitor)
- 按一按板子上的SW2 或SW3 鈕，結果以0, 1表示按鈕狀態。

```
Uploading...
```

```
Sketch uses 218244 bytes (10%) of progra  
Global variables use 81224 bytes (2%) of  
INFO: Goto open COM5
```



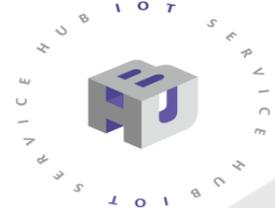
```
COM5
GPIO18 Mode=3 DIR=0
GPIO19 Mode=4 DIR=0
GPIO20 Mode=4 DIR=0
GPIO21 Mode=4 DIR=0
GPIO22 Mode=4 DIR=0
GPIO23 Mode=4 DIR=0
GPIO24 Mode=4 DIR=0
GPIO25 Mode=3 DIR=0
GPIO26 Mode=3 DIR=0
GPIO27 Mode=1 DIR=0
GPIO28 Mode=1 DIR=0
GPIO29 Mode=1 DIR=0
GPIO30 Mode=1 DIR=0
GPIO31 Mode=1 DIR=0
GPIO32 Mode=1 DIR=0
GPIO33 Mode=1 DIR=0
GPIO34 Mode=1 DIR=0
GPIO35 Mode=3 DIR=0
GPIO36 Mode=3 DIR=0
GPIO37 Mode=3 DIR=0
GPIO38 Mode=3 DIR=0
GPIO39 Mode=0 DIR=0
GPIO40 Mode=3 DIR=0
GPIO41 Mode=3 DIR=0
GPIO42 Mode=3 DIR=0
GPIO43 Mode=3 DIR=0
GPIO44 Mode=3 DIR=0
GPIO45 Mode=3 DIR=0
GPIO46 Mode=3 DIR=0
GPIO47 Mode=0 DIR=0
GPIO48 Mode=1 DIR=0
GPIO49 Mode=0 DIR=0
GPIO50 Mode=1 DIR=0
GPIO51 Mode=0 DIR=0
GPIO52 Mode=0 DIR=0
1,1
1,1
0,1
0,0
1,0
0,1
1,0
1,1
1,1
```

每根IO MUX的模式

注意Baud rate!

SW2, SW3 的狀態
0 => 按下
1 => 無動作

Autoscroll Show timestamp Both NL & CR 115200 baud Clear output



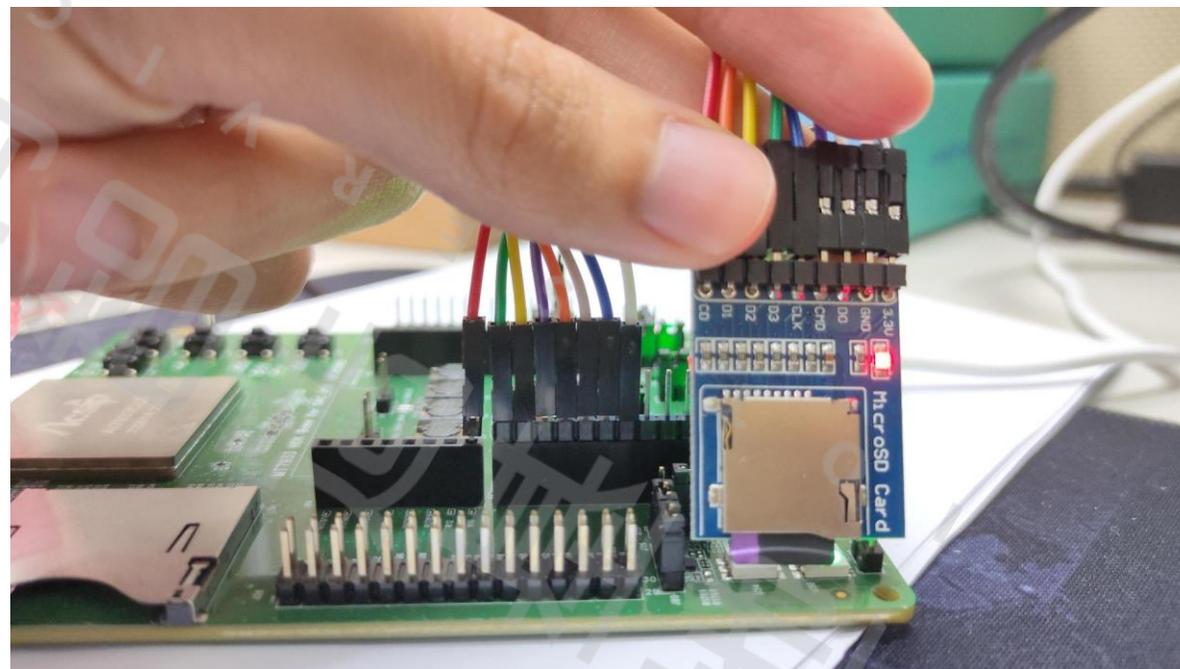
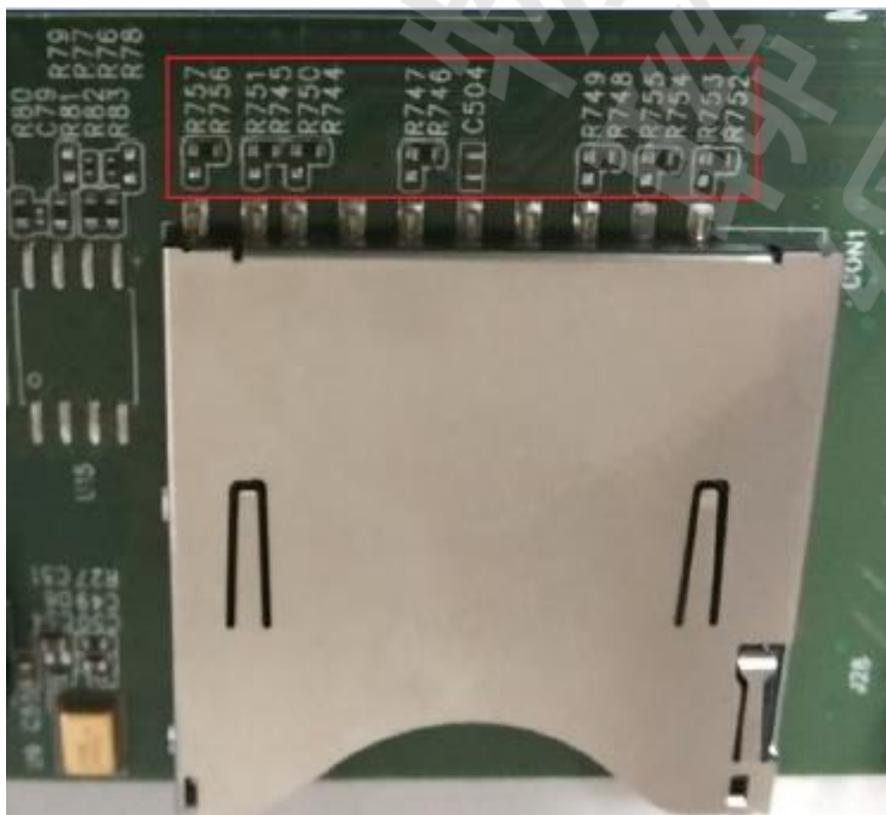
物
聯
網
基
礎
地
盤

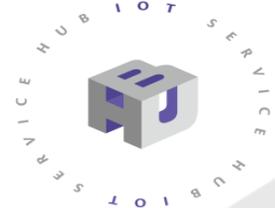
物
聯
網
基
礎
地
盤

示範案例

SDIO Interface

- 開發板上的SD 插槽需要焊接才能接通界面
- 外接SD 插槽，接到J81,J84(Arduino Interface) 排針上





SDIO Interface

- **開發板上的SD卡槽**

- ◆ R747, R749, R750, R751, R753, R755, R757七個點要焊接導通。

- **本範例使用外接SD卡槽到J81, J84, 以下為對應腳位:**

- ◆ J81 - pin4 GND

- ◆ J81 - pin5 SDIO_CLK

- ◆ J81 - pin6 SDIO_DAT0

- ◆ J81 - pin7 SDIO_DAT1

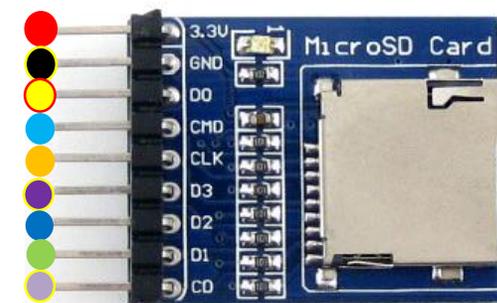
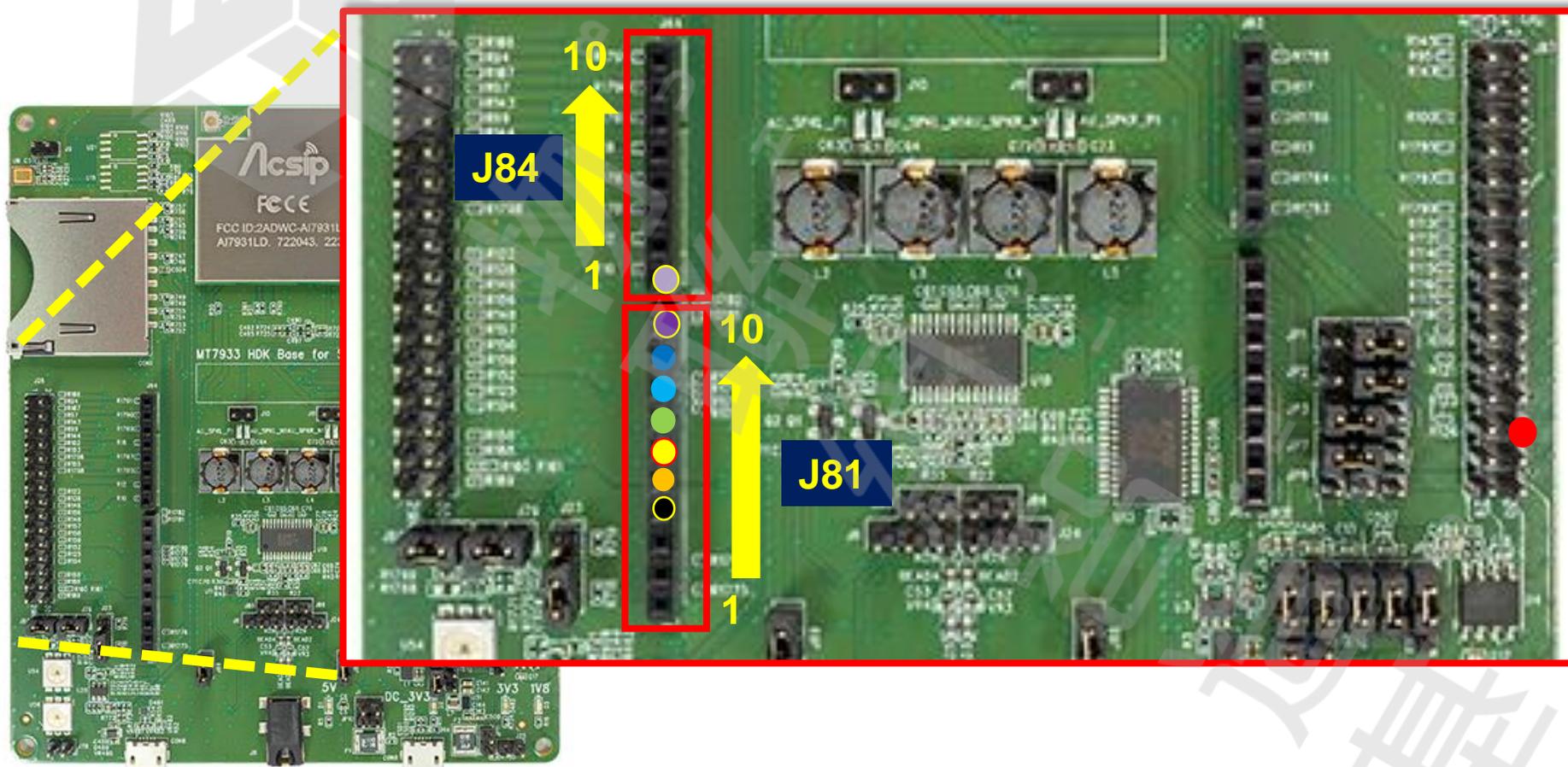
- ◆ J81 - pin8 SDIO_CMD

- ◆ J81 - pin9 SDIO_DAT2

- ◆ J81 - pin10 SDIO_DATA3

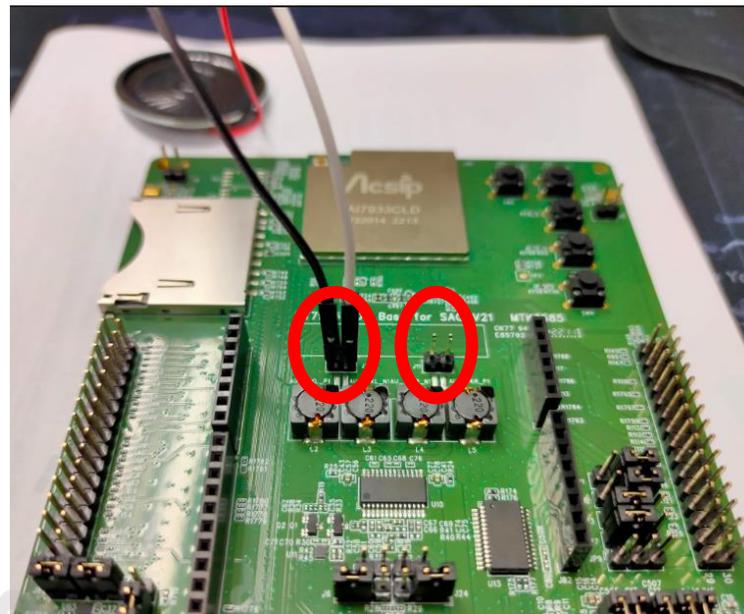
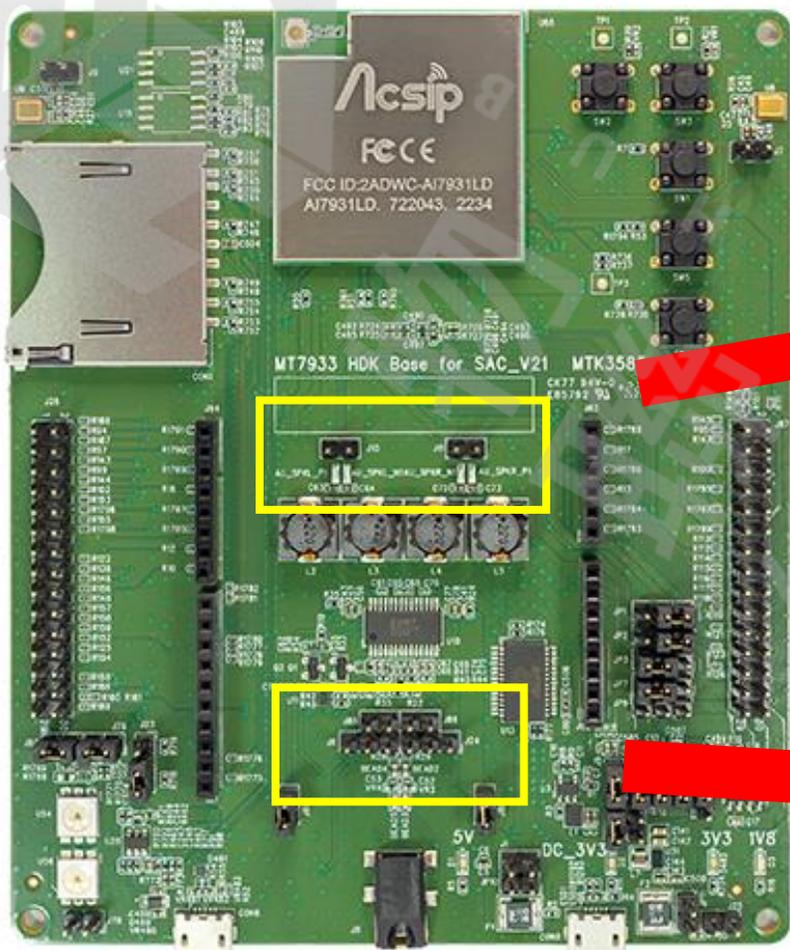
- ◆ J84 - pin 1 CD

電路圖

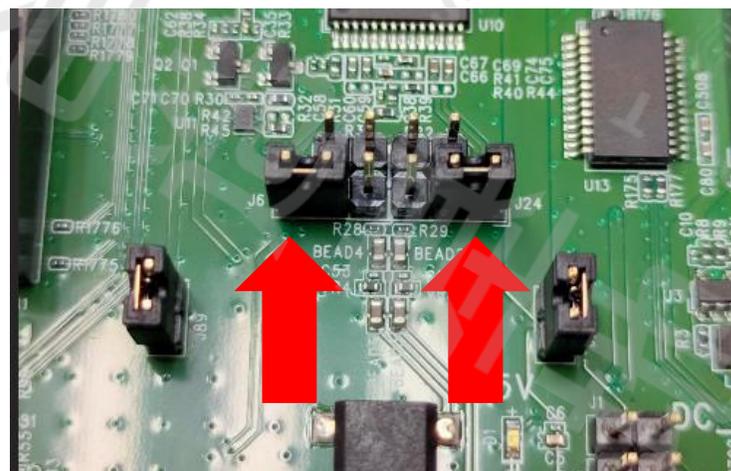


◆ 把相同顏色的圓點連接起來即可

喇叭與Jumper

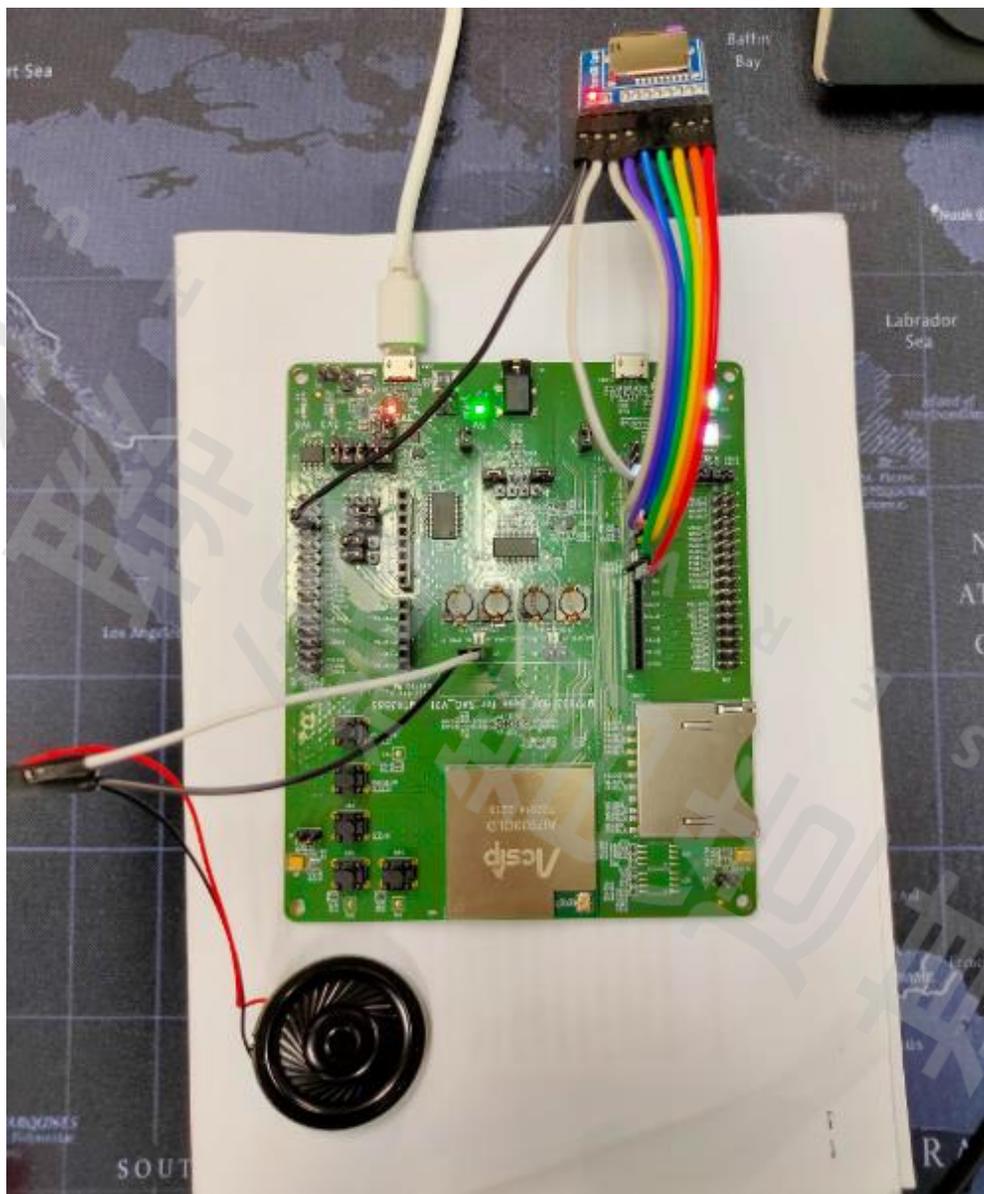


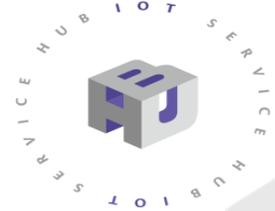
- ◆ 左右聲道
- ◆ 無正負極之分



- ◆ J6, J24 各外側兩個pin用Jumper連接起來，左右聲道才會有聲音

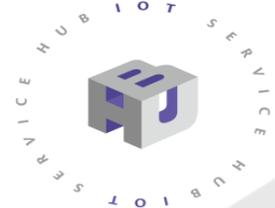
接好後



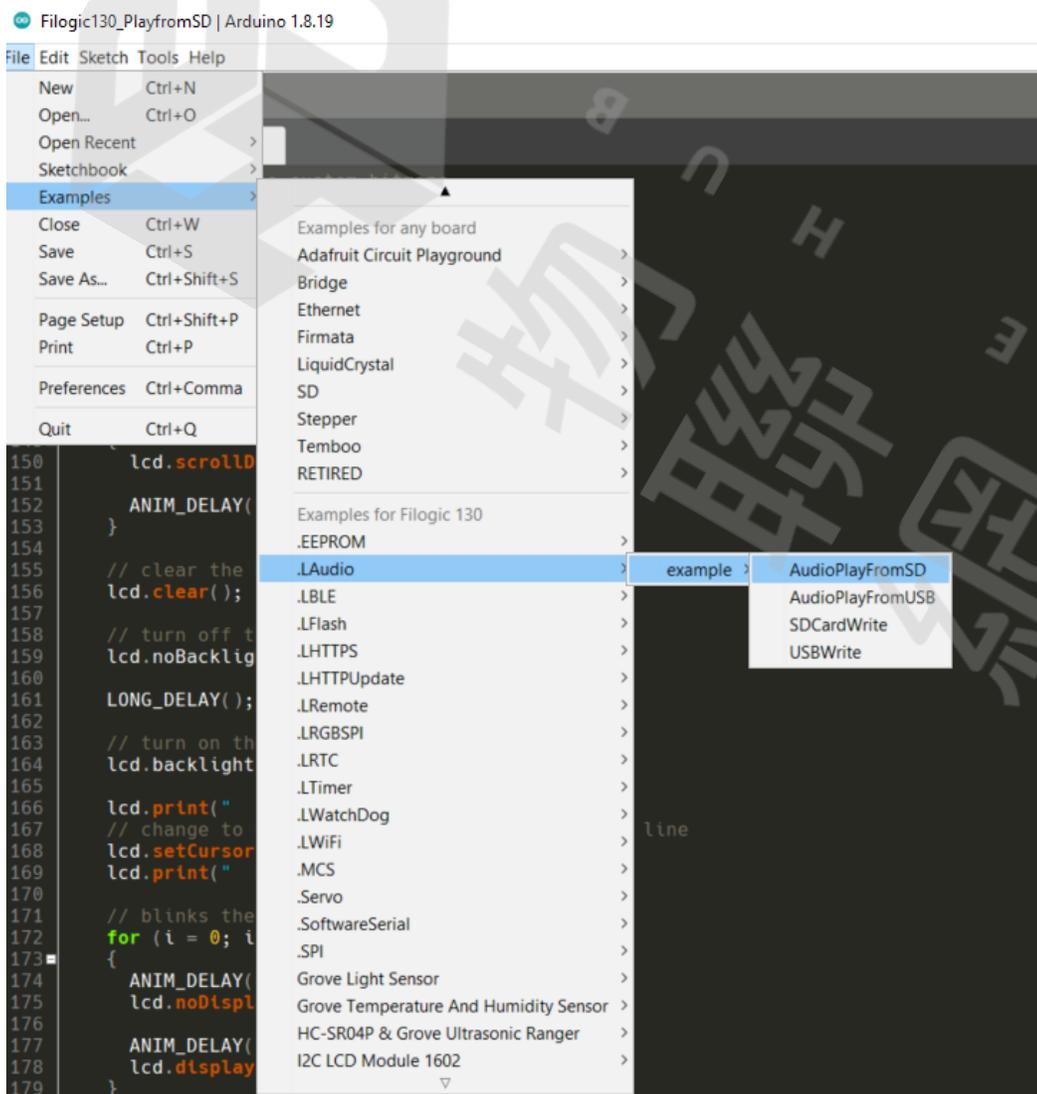


SDIO 檔案管理

- SDK v1.2.0版本已經整合第三方開源庫fatfs(FAT File System)
 - ◆ 標準化的檔案管理函數：`f_open`、`f_read`、`f_write`、`f_close`...等。
 - ◆ 可寫入/讀取數據



打開範例程式



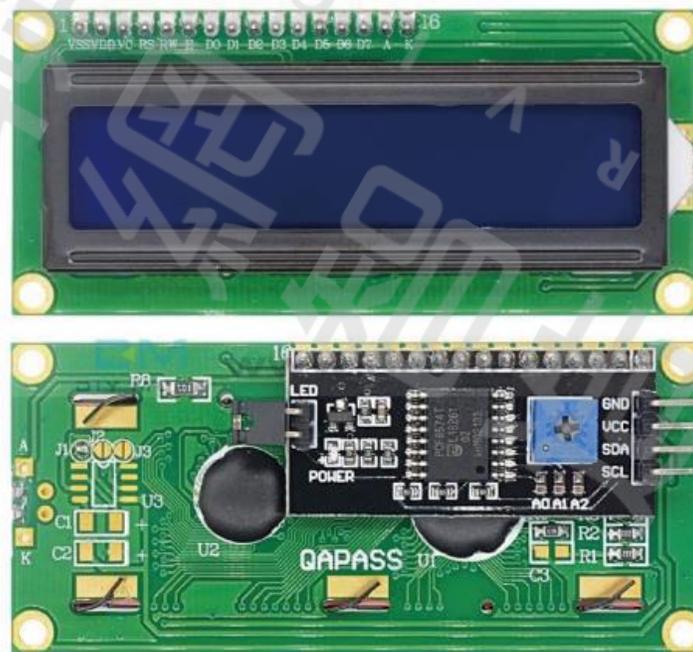
```
AudioPlayFromSD $
1 #include <LAudio.h>
2
3 /*
4  * Audio Playback from SD.
5  */
6 int button_gpio = 15;
7 void setup() {
8   // put your setup code here, to run once:
9   //
10  Serial.begin(115200);
11  Serial.println("Enable Audio");
12  LAudio.Begin();
13  Serial.println("Mount SD card");
14  LAudio.SD_Mount();
15  LAudio.FF_LS("SD:/");
16  Serial.println("Press the button to play audio!");
17 }
18
19 void loop() {
20   // put your main code here, to run repeatedly:
21   // if USB button is pressed, play the music.
22   if (digitalRead(button_gpio))
23   {
24     // Play audio data from SD Card
25     // LAudio.PlayFromSD([filename]);
26     LAudio.PlayFromSD("hello_filogic.pcm");
27   }
28 }
```

可把if 判斷式註解，注意掛號 { }

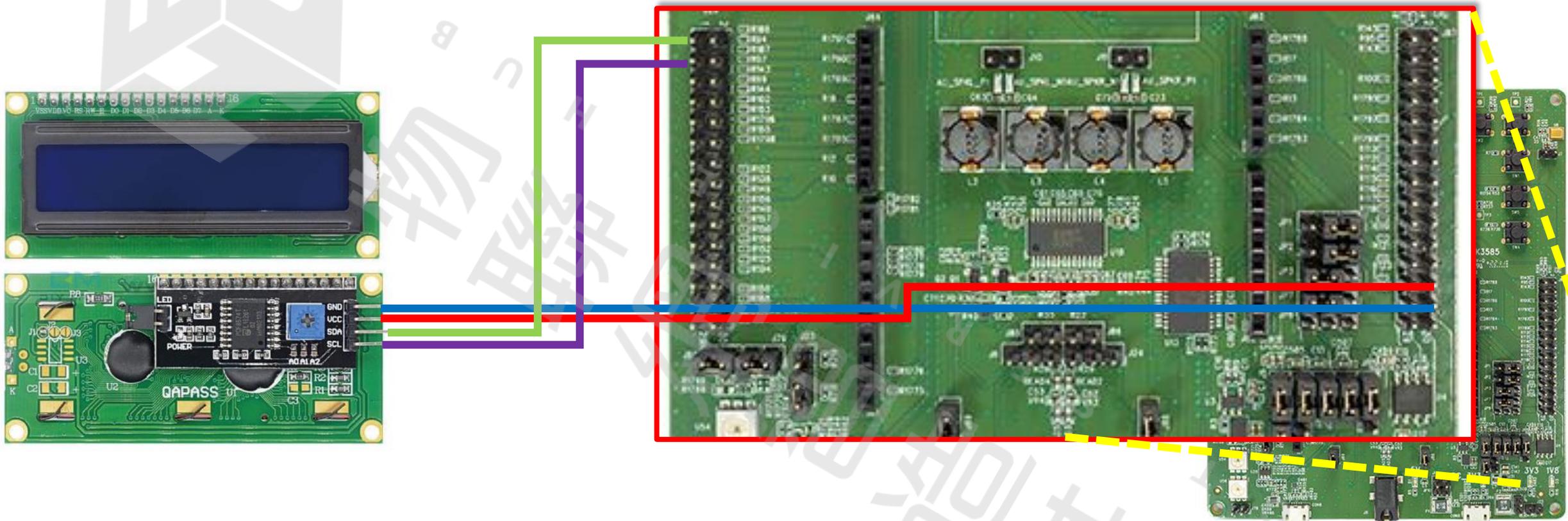
LAudio.PlayFromSD 可播放 .pcm .wav

I2C LCD 1602

- LCD1602的液晶顯示器，可以顯示單一系列16個字元的長度，一次可以顯示兩列，I2C匯流排控制。
- 背面可焊接adapter（如下圖）。

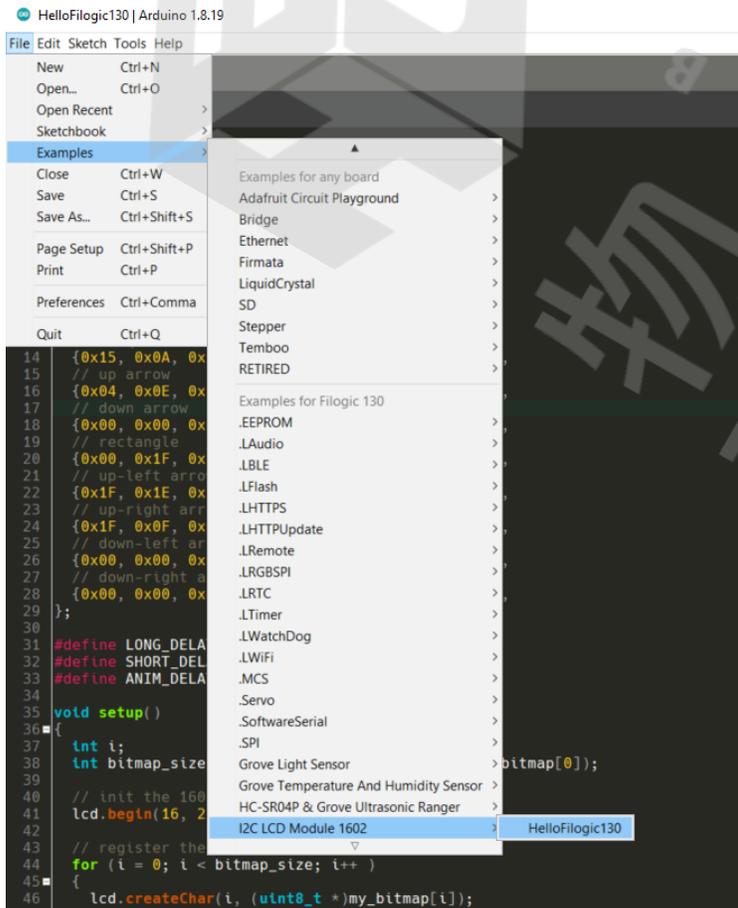


電路圖



◆ 預設使用J28 Pin1 作為 I2C0_SDA,
J28 Pin 3 作為I2C0_SCL

打開範例程式



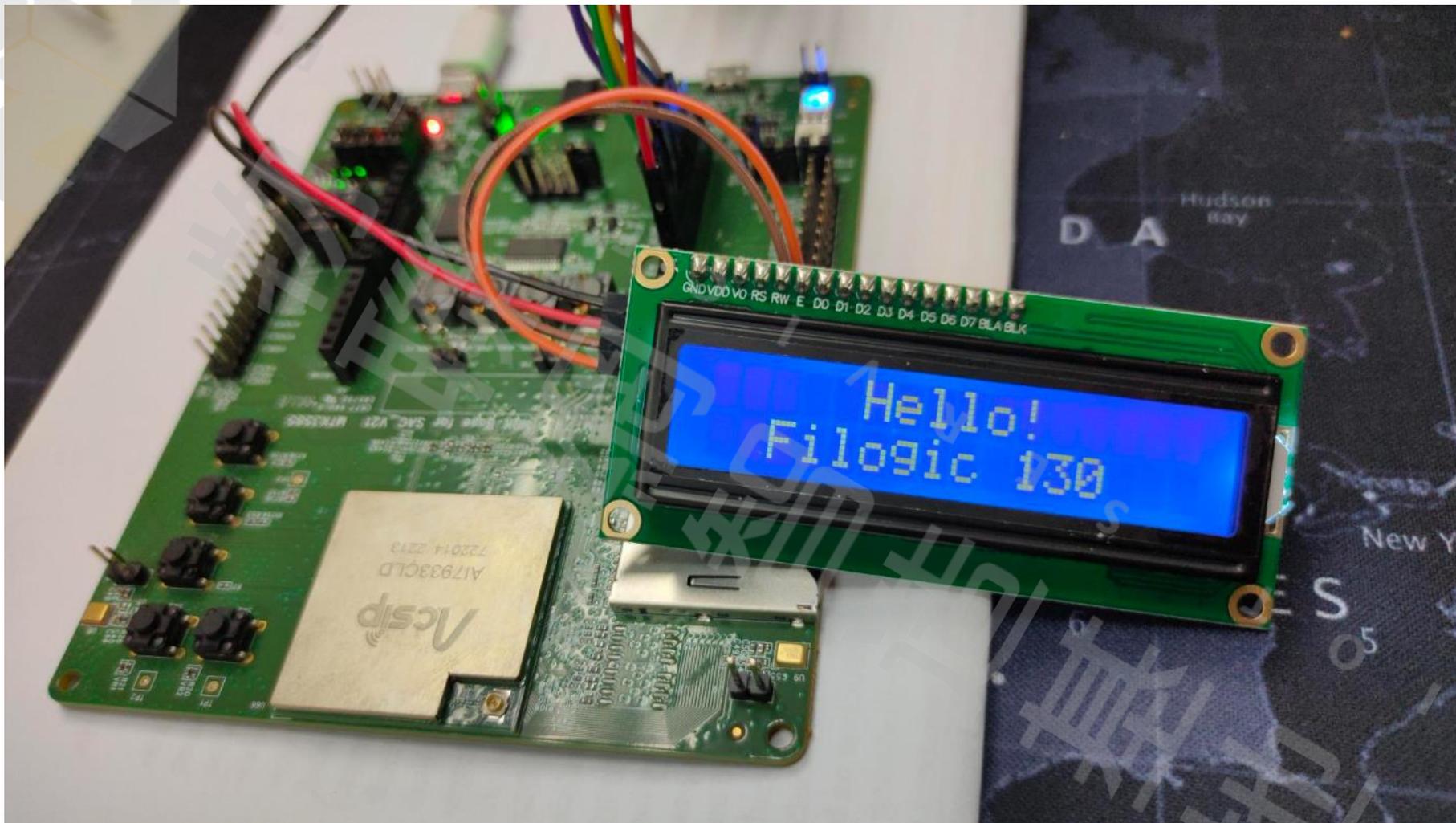
```
HelloFllotc130 $
1 #include <Wire.h>
2 #include <LiquidCrystal_I2C.h>
3
4 // set the I2C address of the LCD controller
5 //   0x3F for PCF8574A
6 //   0x27 for PCF8574
7 LiquidCrystal_I2C lcd(0x27);
8
9 // define the custom bitmaps
10 // up to 8 bitmaps are supported
11 const uint8_t my_bitmap[][8] =
12 {
13   // chequer
14   {0x15, 0x0A, 0x15, 0x0A, 0x15, 0x0A, 0x15, 0x00},
15   // up arrow
16   {0x04, 0x0E, 0x1F, 0x04, 0x04, 0x04, 0x00, 0x00},
17   // down arrow
18   {0x00, 0x00, 0x04, 0x04, 0x04, 0x1F, 0x0E, 0x04},
19   // rectangle
20   {0x00, 0x1F, 0x11, 0x11, 0x11, 0x11, 0x1F, 0x00},
21   // up-left arrow
22   {0x1F, 0x1E, 0x1C, 0x1A, 0x11, 0x00, 0x00, 0x00},
23   // up-right arrow
24   {0x1F, 0x0F, 0x07, 0x0B, 0x11, 0x00, 0x00, 0x00},
25   // down-left arrow
26   {0x00, 0x00, 0x00, 0x11, 0x1A, 0x1C, 0x1E, 0x1F},
27   // down-right arrow
28   {0x00, 0x00, 0x00, 0x11, 0x0B, 0x07, 0x0F, 0x1F},
29 };
30
31 #define LONG_DELAY() delay(1000)
32 #define SHORT_DELAY() delay(500)
33 #define ANIM_DELAY() delay(400)
34
35 void setup()
36 {
37   int i;
38   int bitmap_size = sizeof(my_bitmap) / sizeof(my_bitmap[0]);
39
40   // init the 1602 (2 rows / 16 columns) LCD
41   lcd.begin(16, 2);
```

若無顯示任何字，可更改address試試

或者調整亮度

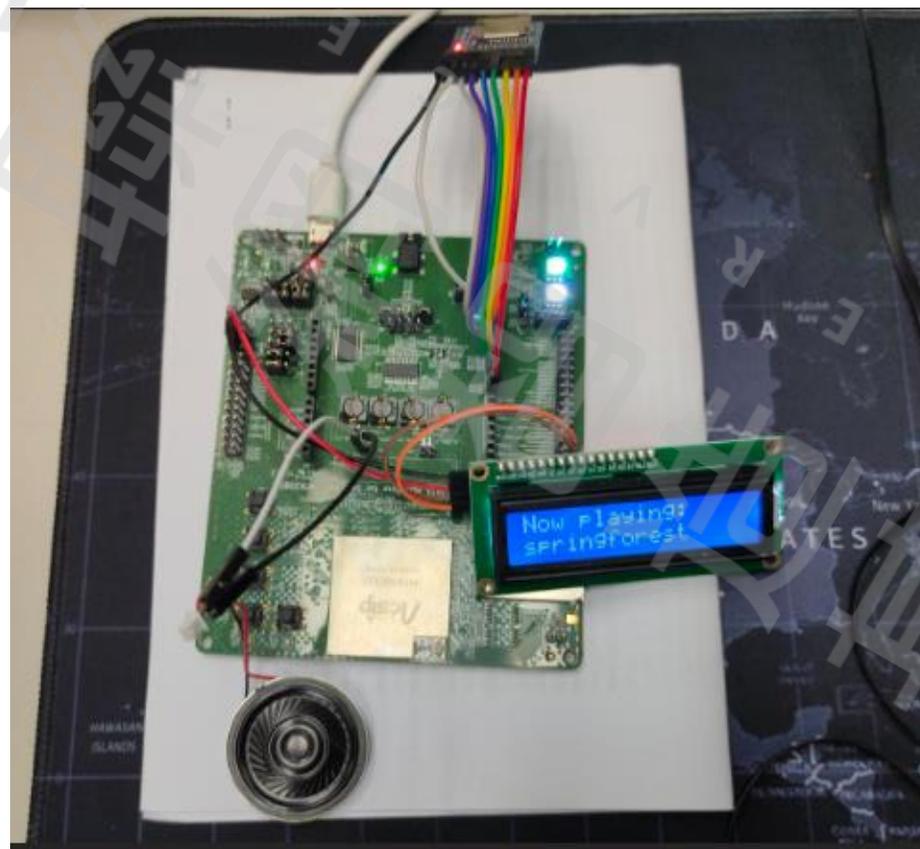


執行結果



播放音樂 + 顯示目前播放歌曲

- SDIO + LCD 1602設備腳位相同
- SD卡上可存入多首歌曲 (.pcm .wav)



執行結果



執行結果

```
COM3
Send
----ARC 2023/03/15 16:39 34922908 lost.pcm
----ARC 2023/03/30 16:50 1024000 hello_filologic.pcm
System Volume Information
The Messenger.wav
Perfect.wav
Lost.wav
perfect.pcm
moonlightsonata.pcm
MoonlightSonata.mp3
cinematic.pcm
springforest.pcm
intomyself.pcm
bach.pcm
spring-forest.mp3
peaceful-cinematic.mp3
bach.mp3
into-myself.mp3
lost.pcm
hello_filologic.pcm
List of PCM file in SD Card:
perfect.pcm
moonlightsonata.pcm
cinematic.pcm
springforest.pcm
intomyself.pcm
bach.pcm
lost.pcm
hello_filologic.pcm
Now playing:
perfect.pcmPlayFromSD, file = SD:/perfect.pcm
[audio_play_file_cmd]:327: msg: play_type = 1
 Autoscroll  Show timestamp
Both NL & CR 115200 baud Clear output
```

在監控視窗可看到
SD卡中所有的檔案

列出.pcm檔案

顯示現在播放歌曲

程式簡述

```
1 #include <LAudio.h>
2 #include "ff.h"
3 #include <Wire.h>
4 #include <LiquidCrystal_I2C.h>
5
6 FRESULT res;
7 DIR dir;
8 FILINFO Finfo;
9
10 #define MAX_FILES 10
11 char pcmFile[MAX_FILES][20] = {};
12
```

Fatfs 函數宣告，用以後續讀取SD卡資料

用以儲存pcm檔案名字，MAX_FILES 是為了避免memory overflow而設定

```
44 void setup()
45 {
46     Serial.begin(115200);
47     initLCD();
48     Serial.println("Enable Audio");
49     LAudio.Begin();
50     Serial.println("Mount SD card");
51     LAudio.SD_Mount();
52     LAudio.FF_LS("SD:/");
53     listFile();
54
55     Serial.println("List of PCM file in SD Card: ");
56     for (int i = 0; i < sizeof(pcmFile)/sizeof(pcmFile[0]); i++)
57     {
58         Serial.println(pcmFile[i]);
59     }
60
61 }
62
```

初始化LAudio，並mount SD卡後，讀取檔案名字

列出pcmFile裡存的檔案名字

程式簡述

```

88 // Store target file name in array and print out
89 void listFile()
90 {
91     int idx = 0; // for array indexing
92     res = f_opendir(&dir, "SD:/");
93     if (res)
94     {
95         printf("[FS]: Folder Open Fail - res(%u)\r\n", res);
96     }
97
98     for(;;)
99     {
100         res = f_readdir(&dir, &Finfo);
101         if (res || !Finfo.fname[0])
102         {
103             break;
104         }
105
106         // Determine .pcm file and store in array
107         if(getPCMFile(Finfo.fname))
108         {
109             strncpy(pcmFile[idx], Finfo.fname, 20); // copy fname into pcmFile[idx]
110             idx++;
111         }
112         printf("%s\n", Finfo.fname);
113     }
114     f_closedir(&dir);
115 }
116
117
118 //Determine PCM file
119 boolean getPCMFile(char fName[])
120 {
121     char* ext;
122
123     ext = strchr(fName, '.');
124     if(strcmp(ext, ".pcm") == 0) return true;
125     else
126         return false;
127 }

```

listFile -> 讀出SD卡內的檔案並存pcm檔名

getPCMFile -> 判斷是否為 pcm 檔案

程式簡述

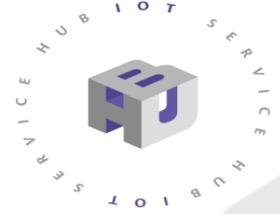
```

64 void loop()
65 {
66     int i = 0;
67     char tmp[20];
68     lcd.clear();
69     for(i = 0; i < sizeof(pcmFile)/sizeof(pcmFile[0]); i++)
70     {
71         if (pcmFile[i][0] != '\0')
72         {
73             Serial.println("Now playing: ");
74             lcd.leftToRight();
75             lcd.print("Now playing: ");
76             Serial.print(pcmFile[i]);
77             lcd.setCursor(0, 1);
78             strncpy(tmp, pcmFile[i], 20);
79             lcd.print( strtok(tmp, ".") );
80             LAudio.PlayFromSD(pcmFile[i]);
81         }
82         delay(1000);
83         lcd.clear();
84     }
85 }

```

用tmp 來確保 pcmFile裡的東西不會被更改

PlayFromSD 會根據檔名去播放音樂



物聯網製造基地

Thank you

