聯發科

Filogic 130A (MT7933)

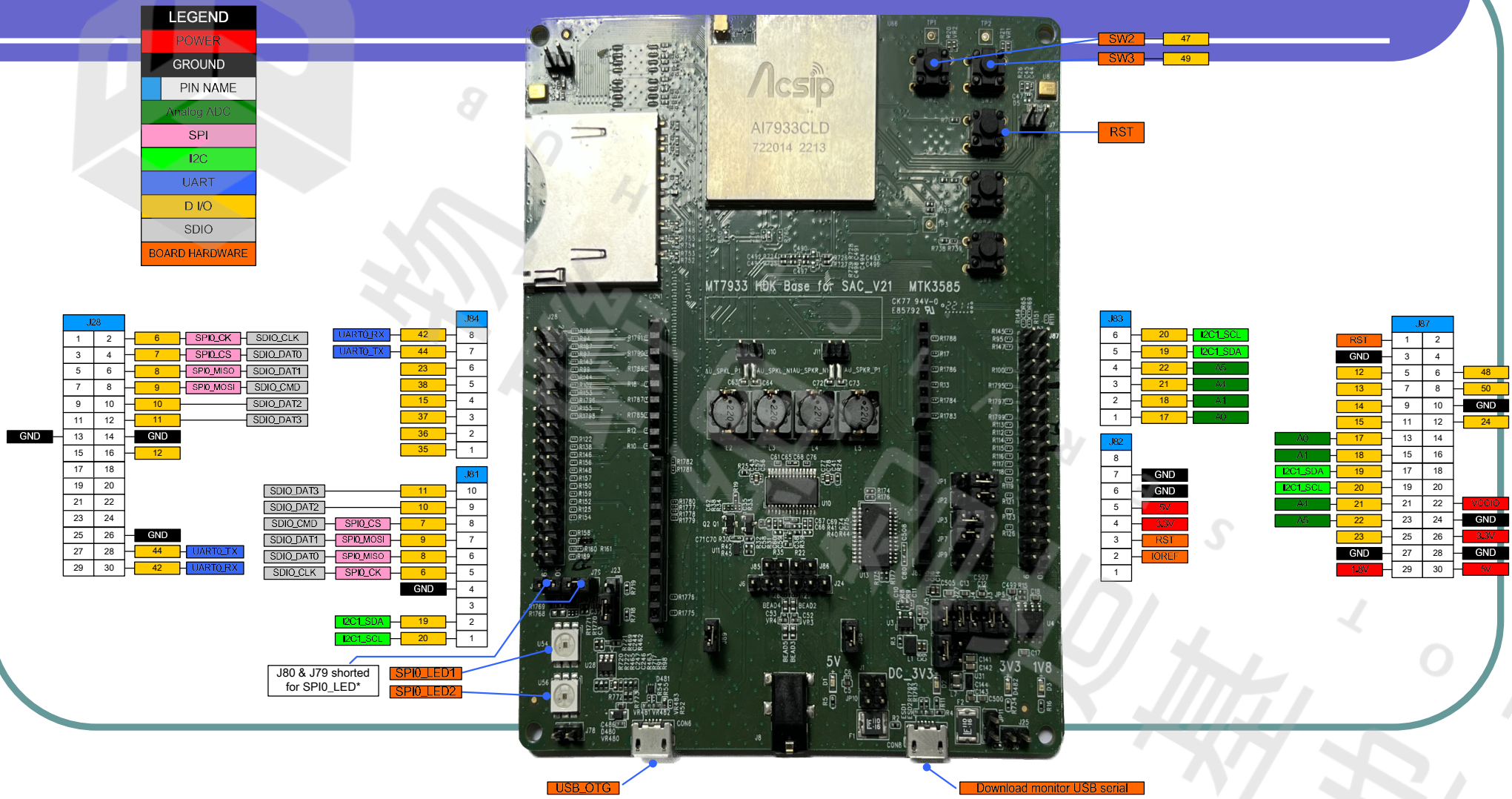
Arduino開發板教學

My Name: Tom, Yen(顏仲良)

大綱

- 開發板Filogic130A 介紹
- 開發環境：Arduino SDK安裝與韌體上傳
- 周邊介面與範例說明：
  - 溫溼度感測模組 DHT11
  - 板上RGB LED
  - MQTT 訂閱/發佈

開發板Filogic130A 介紹:開發板硬體外觀及I/O功能



開發板Filogic130A 介紹: Arduino目前可以使用I/O及硬體資源

MCU

- UART X 1
- I2C X 1
- SPI / SDIO/SD卡
- ADC X 4 (輸入範圍0~1.8V)
- PWM X 12
- USB DISK

無線連網

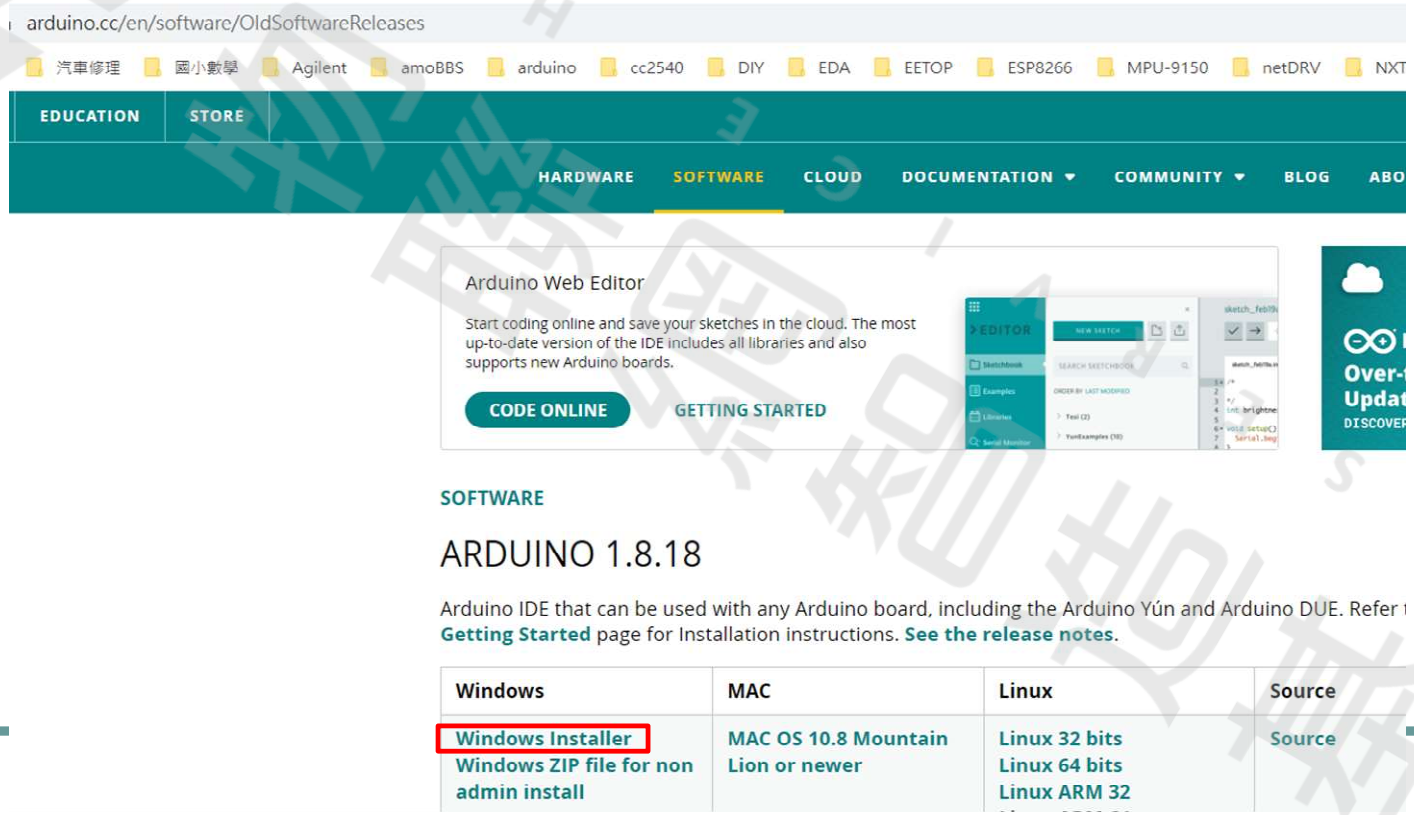
- Wi-Fi 6.0(2.4/5GHz)
- BT 5.0

音訊

- Stereo speaker / line out
- Microphone X 2

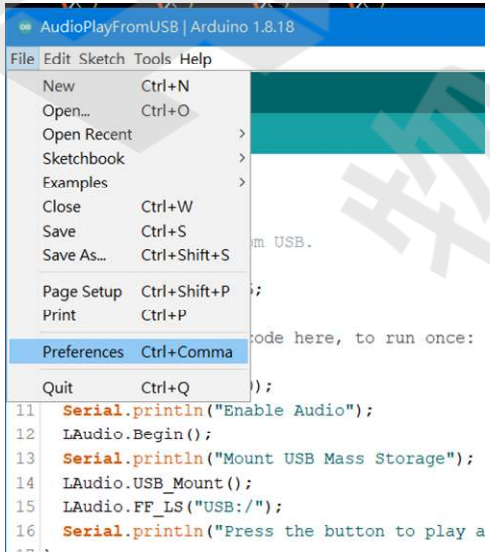
開發環境 Filogic Arduino 1.8.18 安裝(1)

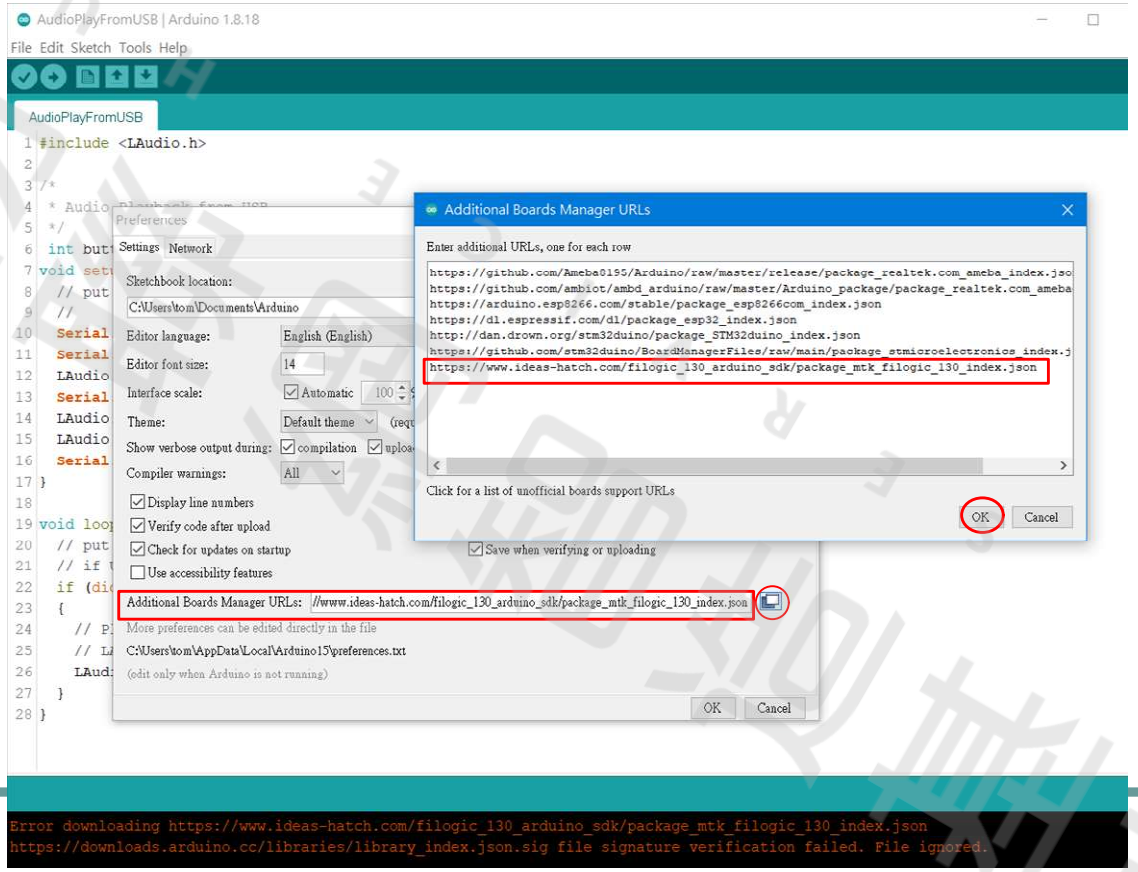
- 下載Arduino 1.8.18
  https://www.arduino.cc/en/software/OldSoftwareReleases



開發環境 Filogic Arduino 1.8.18 安裝(2)

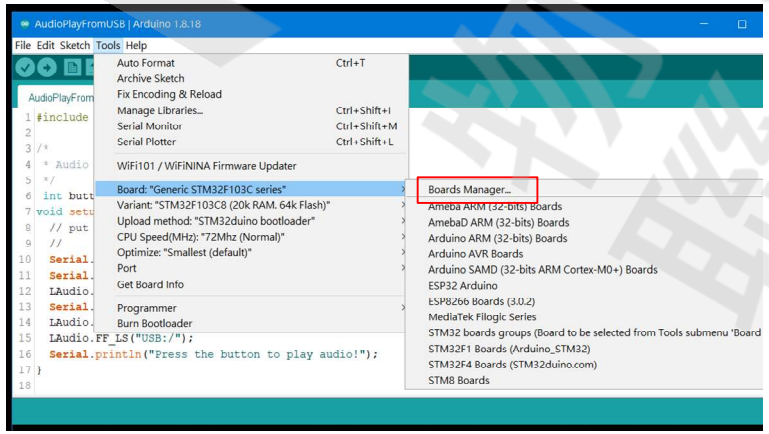
- 在開發板管理網址設定: https://www.ideas-hatch.com/filogic_130_arduino_sdk/package_mtk_filogic_130_index.json

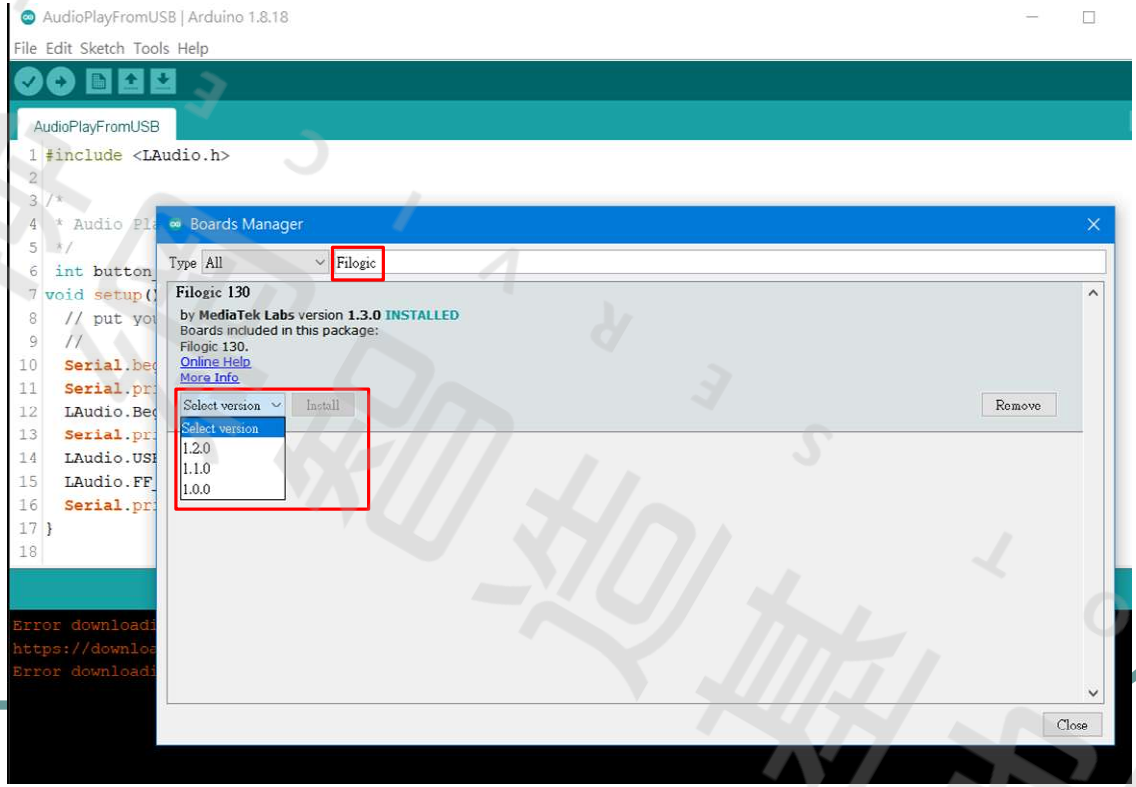




開發環境 Filogic Arduino 1.8.18 安裝(3)

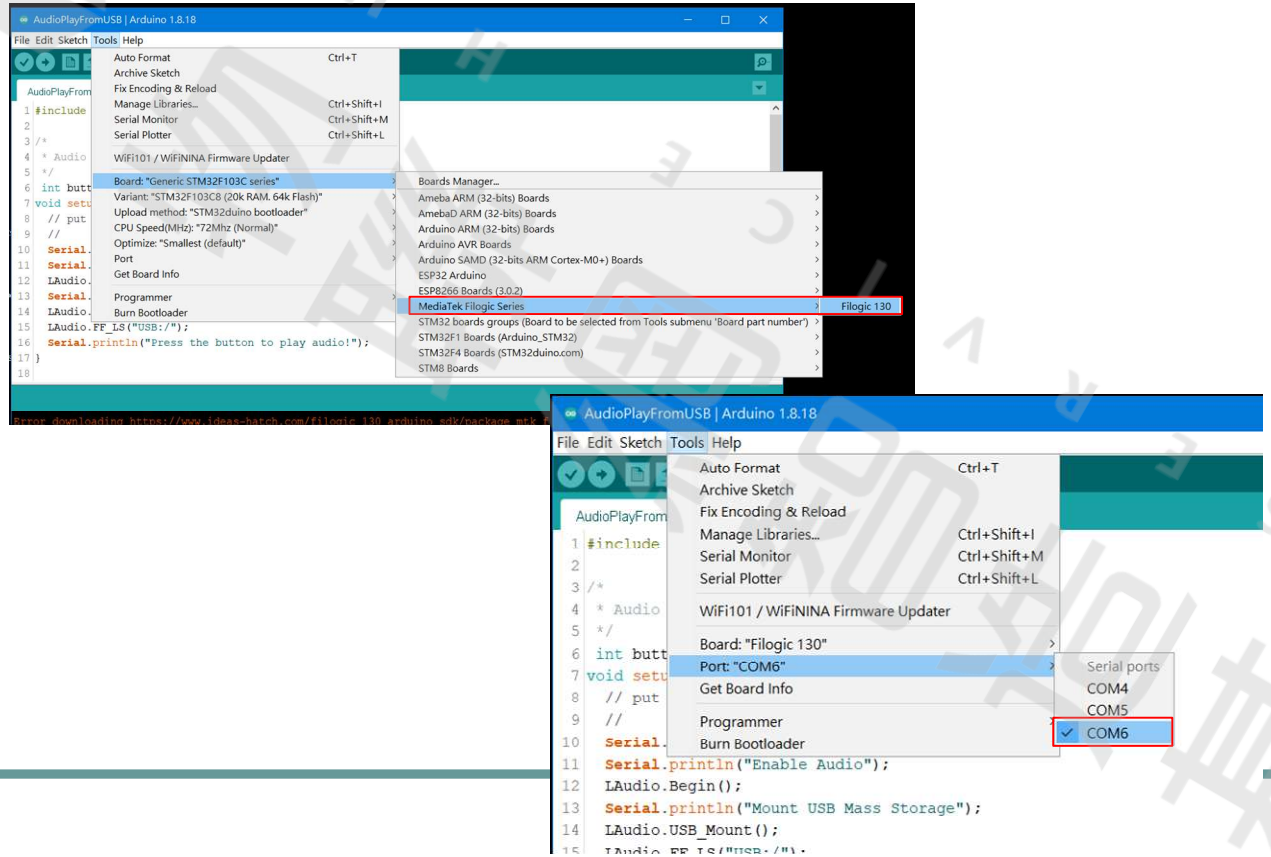
- 按裝開發板Filogic 1.0.0逐版安裝到 1.3.0





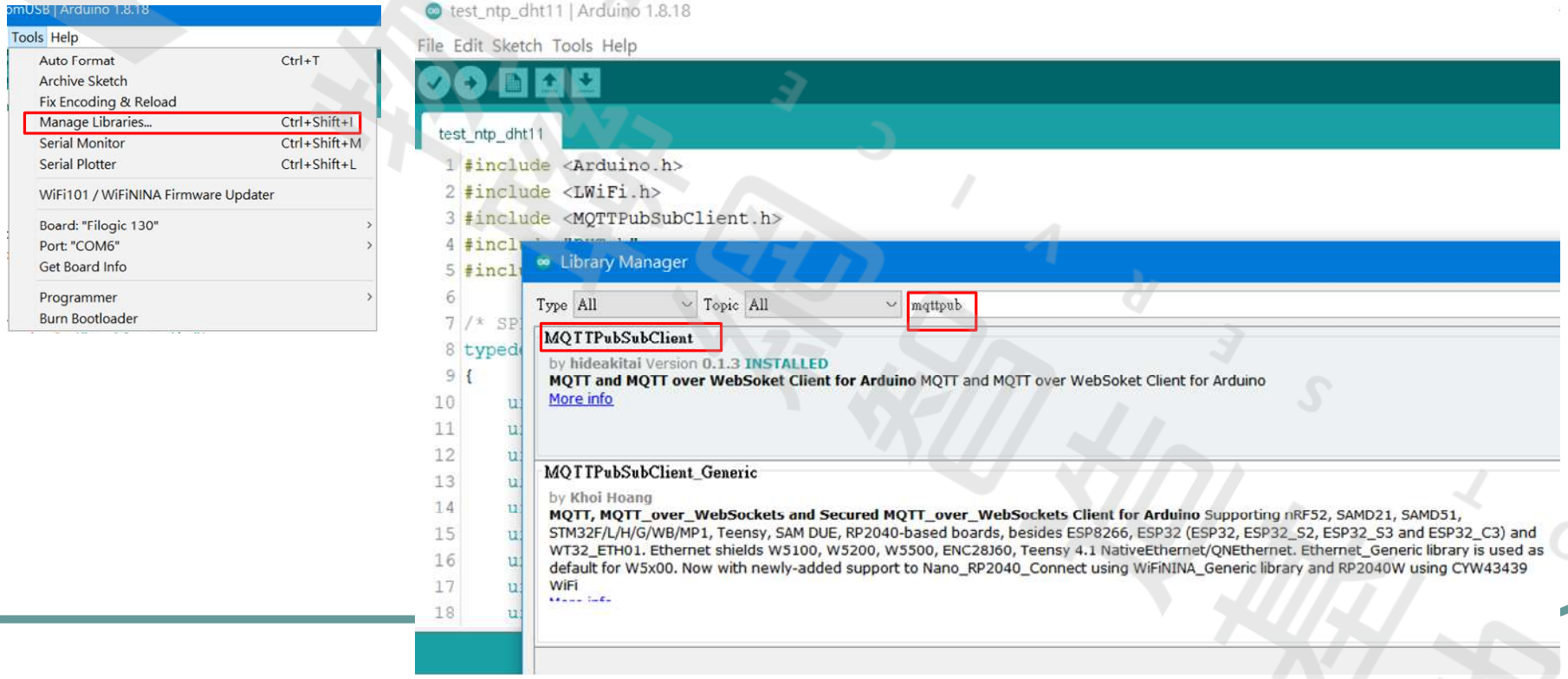
開發環境 Filogic Arduino 1.8.18 安裝(4)

- 選擇Filogic開發版及serial com port



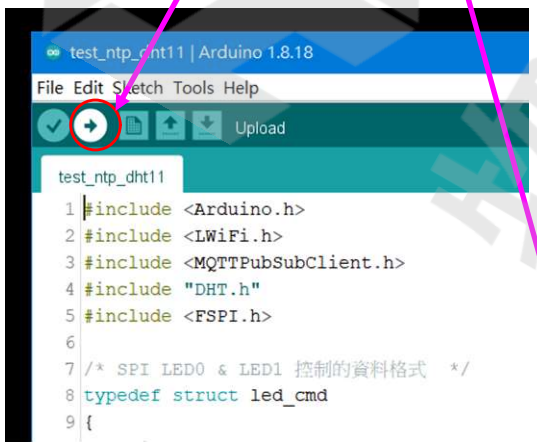
開發環境 Filogic Arduino 1.8.18 安裝(5)

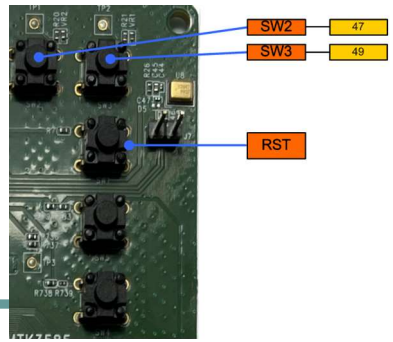
- 安裝 MQTTPubSubClient library

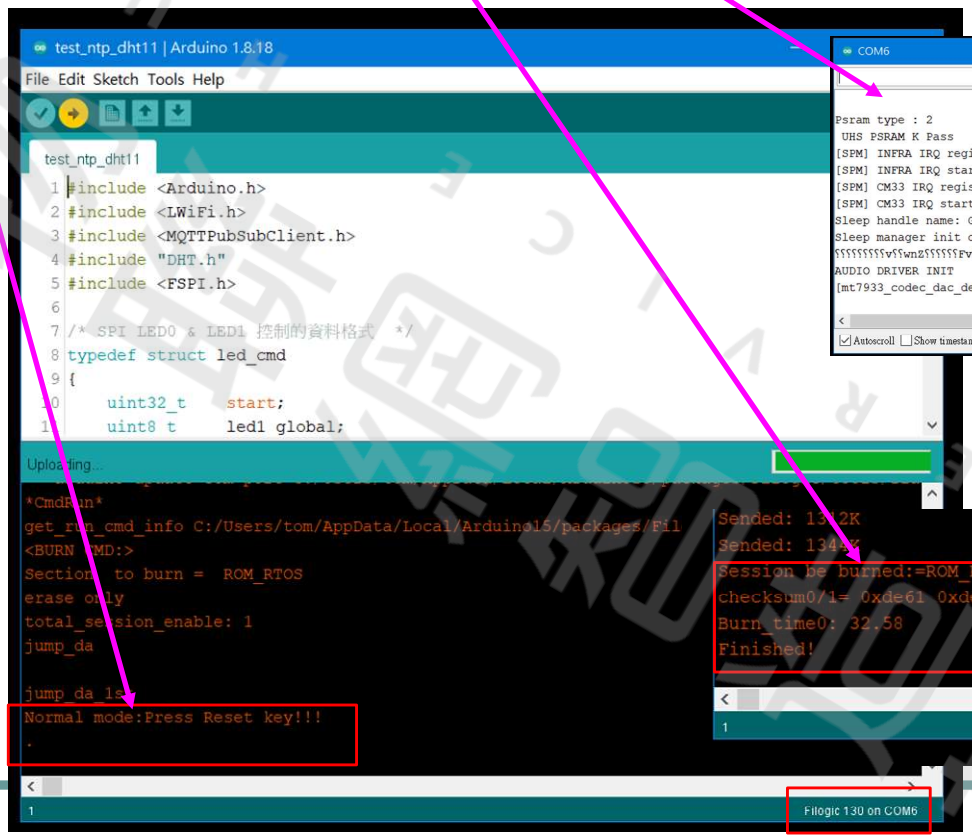


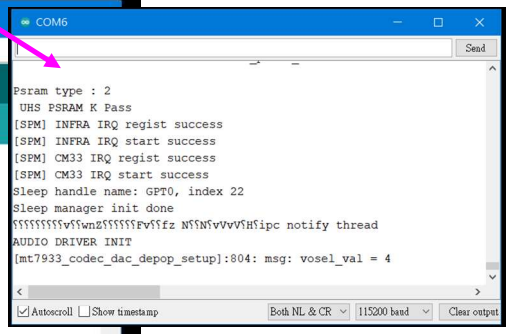
開發環境 Filogic Arduino韌體上傳

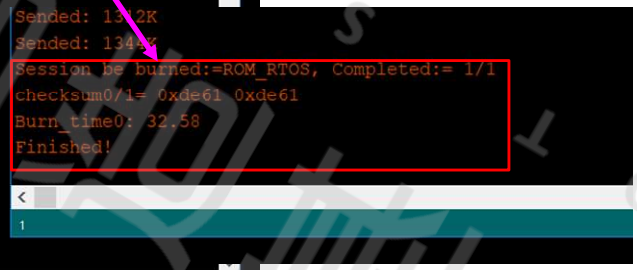
- Upload -> RST->(Finished)RST->RUN











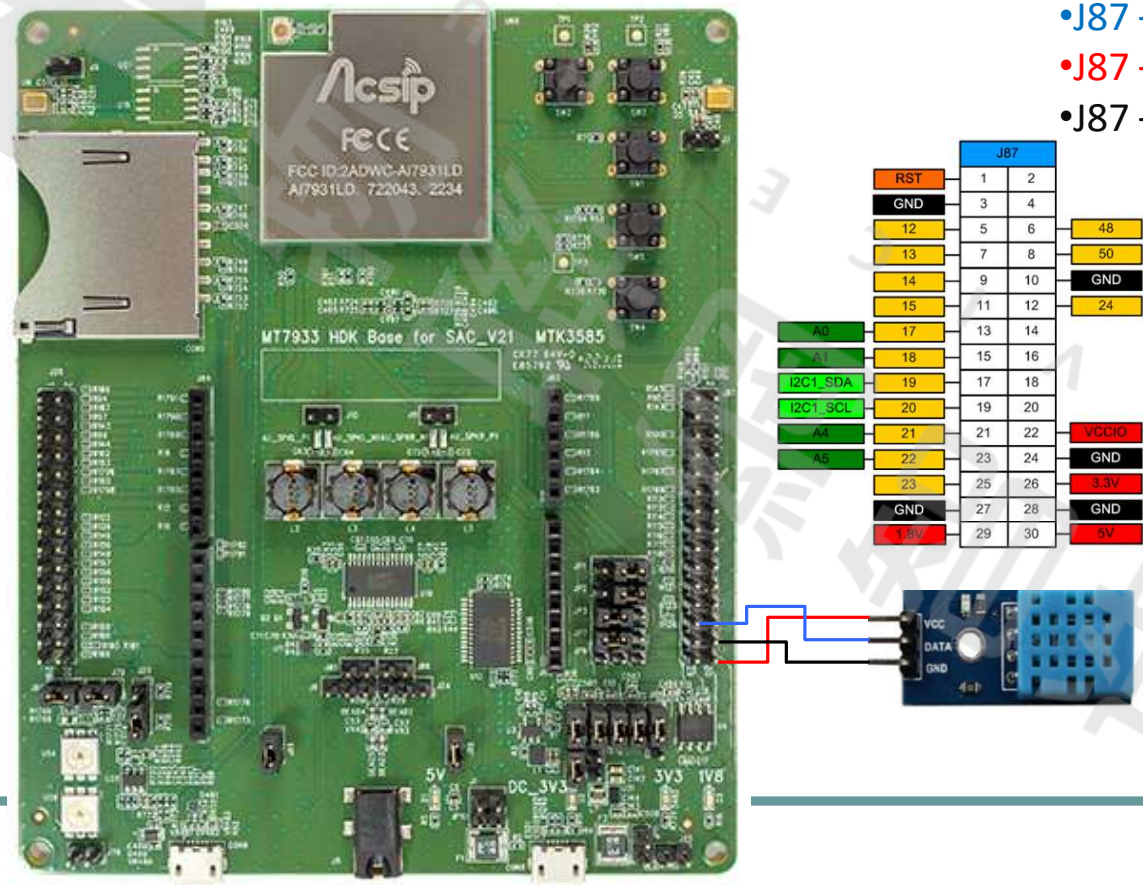
周邊介面與範例說明：溫溼度感測模組 DHT11(1)

- DHT11是一個結合濕度計和測溫元件量測週遭空氣環境，並與一個高性能八位元單晶片相連接，將所量測到的溫、濕度資料拆解成為數位訊號，再由感測器接腳將資料送出。使用上很簡單，但是抓取資料時必須要特別注意時間的掌控，而且每筆資料的抓取時間間隔要 2 秒鐘以上，不能太快。

- DHT11 的規格如下：
- 濕度測量範圍：20~90%;
- 濕度測量精度：±5%;
- 溫度測量範圍：0~50℃
- 溫度測量精度：±2℃
- 電源供應範圍：3~5V
- 頻率不可超過：0.5Hz (每2秒一次)

周邊介面與範例說明：溫溼度感測模組 DHT11(2)

電路連接圖

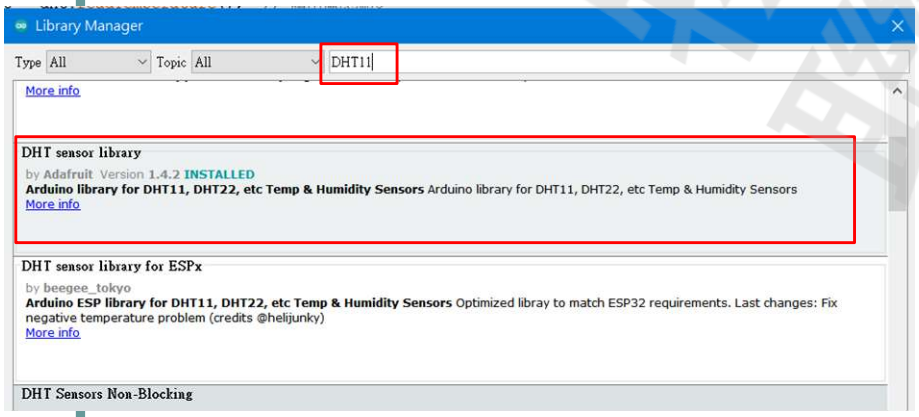


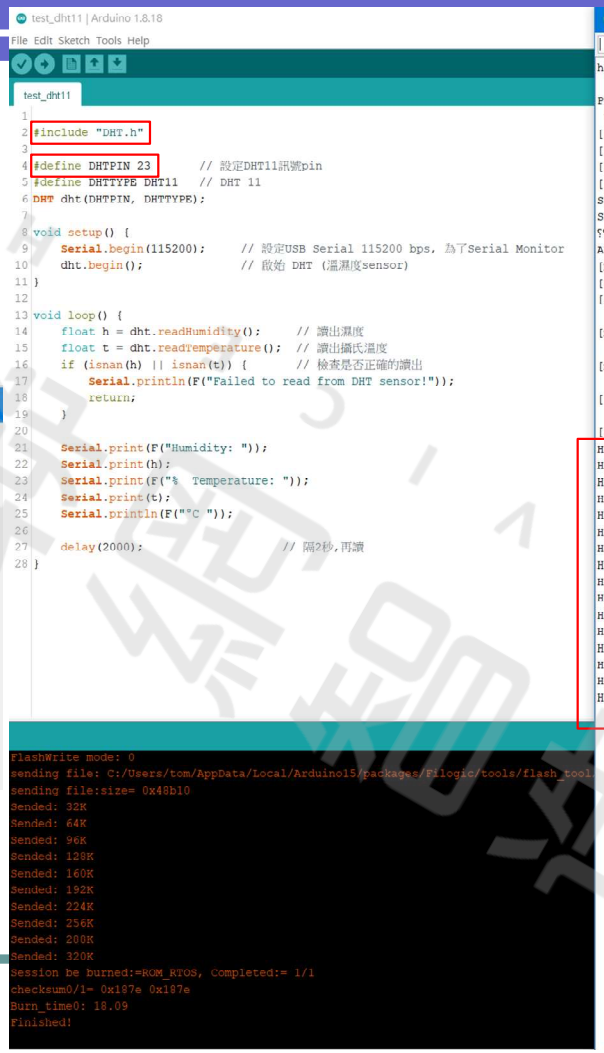
PIN：

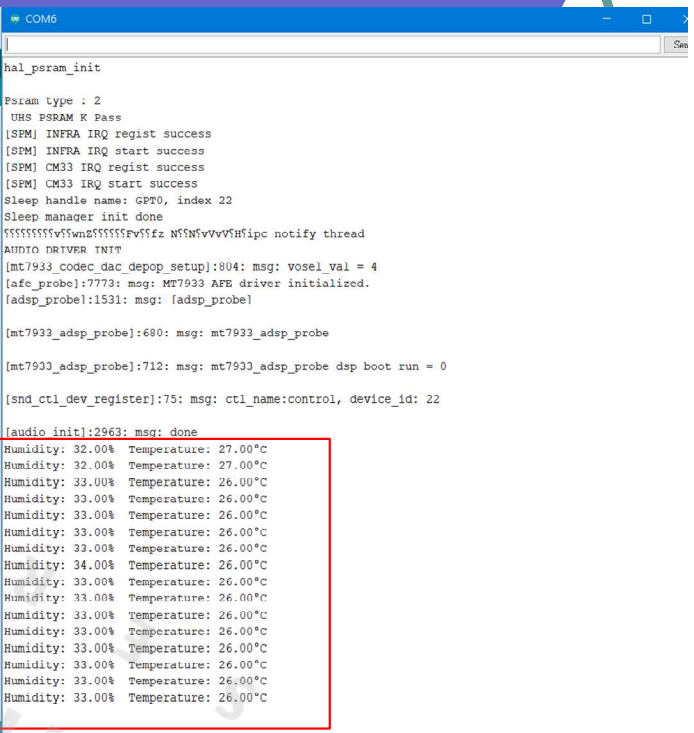
- J87 – 25 Arduino 23(左側倒數第三個)
- J87 – 30 5V (右側倒數第一個)
- J87 – 28 GND (右側倒數第二個)

周邊介面與範例說明：溫溼度感測模組 DHT11(3)

- test_dht11
- 安裝 DHT sensor library
- 修改DHTPIN 為23





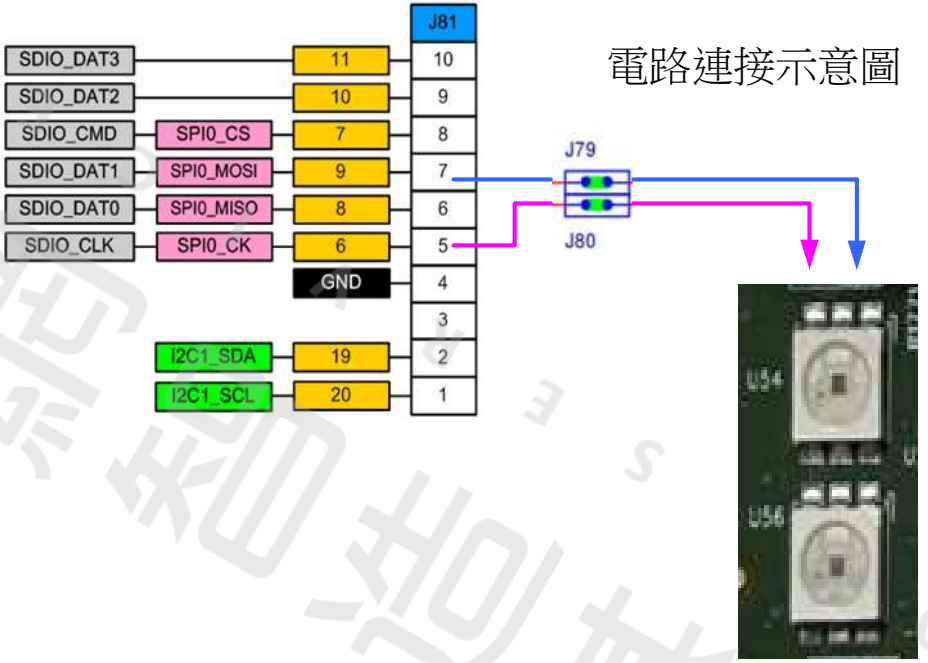


周邊介面與範例說明：板上RGB LED (1)

- 板上的2顆3色LED透過SPI調整明滅顏色
- 資料格式如下:

```
/* SPI LED0 & LED1 控制的資料格式 */
typedef struct led_cmd
{
    uint32_t    start;
    uint8_t     led1_global;
    uint8_t     led1_b;
    uint8_t     led1_g;
    uint8_t     led1_r;
    uint8_t     led2_global;
    uint8_t     led2_b;
    uint8_t     led2_g;
    uint8_t     led2_r;
    uint32_t    end;
} led_cmd_t;
```

0x00000000
0xf0
LED1_Blue
LED1_Green
LED1_Red
0xf0
LED2_Blue
LED2_Green
LED2_Red
0xffffffff



電路連接示意圖

周邊介面與範例說明：板上RGB LED (2)

- 程式說明

test_obRGBLED | Arduino 1.8.18

File Edit Sketch Tools Help

test_obRGBLED

```
#include <FSPI.h>
```
引入SPI程式庫

```
/* SPI LED0 & LED1 控制的資料格式 */
typedef struct led_cmd
{
    uint32_t    start;
    uint8_t     led1_global;
    uint8_t     led1_b;
    uint8_t     led1_g;
    uint8_t     led1_r;
    uint8_t     led2_global;
    uint8_t     led2_b;
    uint8_t     led2_g;
    uint8_t     led2_r;
    uint32_t    end;
} led_cmd_t;

led_cmd_t  led_cmd;     // 控制LED暫存區

void ledOff(){                          // 將LED off,並設定傳送的格式
    led_cmd.start       = 0x00000000;
    led_cmd.led1_global = 0xF0;
    led_cmd.led1_b      = 0x00;
    led_cmd.led1_g      = 0x00;
    led_cmd.led1_r      = 0x00;
    led_cmd.led2_global = 0xF0;
    led_cmd.led2_b      = 0x00;
    led_cmd.led2_g      = 0x00;
    led_cmd.led2_r      = 0x00;
    led_cmd.end         = 0xFFFFFFFF;
    FSPI.transfer(&led_cmd, sizeof(led_cmd));
}
```

```
void setLedColor(uint8_t r1,uint8_t g1,uint8_t b1,uint8_t r2,uint8_t g2,uint8_t b2){
    led_cmd.led1_b      = b1;
    led_cmd.led1_g      = g1;
    led_cmd.led1_r      = r1;
    led_cmd.led2_b      = b2;
    led_cmd.led2_g      = g2;
    led_cmd.led2_r      = r2;
    FSPI.transfer(&led_cmd, sizeof(led_cmd));
}
```
設定LED顏色並透過SPI傳送設定資訊

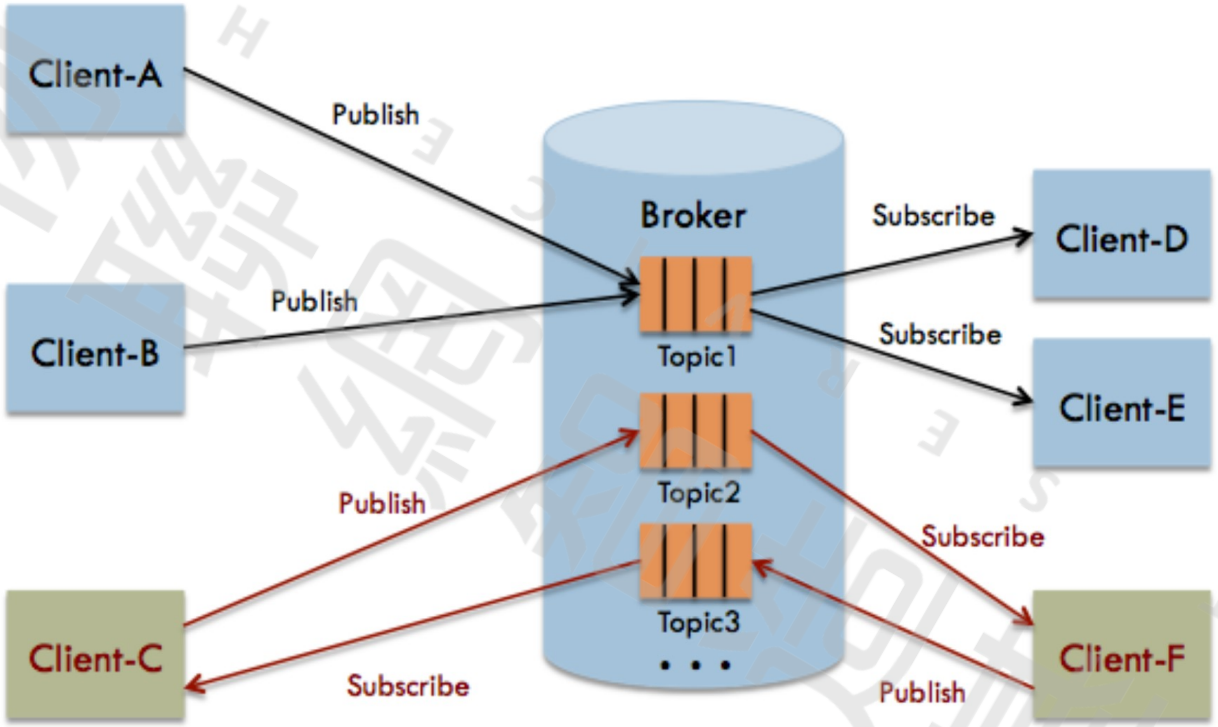
```
int colorI;
int tcolor;

void setup() {
    FSPI.begin();
    Serial.begin(115200);
    ledOff();
    colorI=0;
}
```
設定SPI、Serial及LED

```
void loop() {
    if (colorI>=1536) colorI=0;
    if (colorI<256){
        tcolor =colorI;
        setLedColor(tcolor,0,0,tcolor,tcolor,0);
    }else if (colorI< 512){
        tcolor=0xff & (255-(colorI & 0xff));
        setLedColor(tcolor,0,0,tcolor,tcolor,0);
    }else if (colorI< 768){
        tcolor = colorI & 0xff;
        setLedColor(0,tcolor,0,0,tcolor,tcolor);
    }else if (colorI< 1024){
        tcolor = 0xff & (255-(colorI & 0xff));
        setLedColor(0,tcolor,0,0,tcolor,tcolor);
    }else if (colorI< 1280){
        tcolor = colorI & 0xff;
        setLedColor(0,0,tcolor,tcolor,0,tcolor);
    }else{
        tcolor = 0xff & (255-(colorI & 0xff));
        setLedColor(0,0,tcolor,tcolor,0,tcolor);
    }
    colorI++;
    delay(50);
}
```

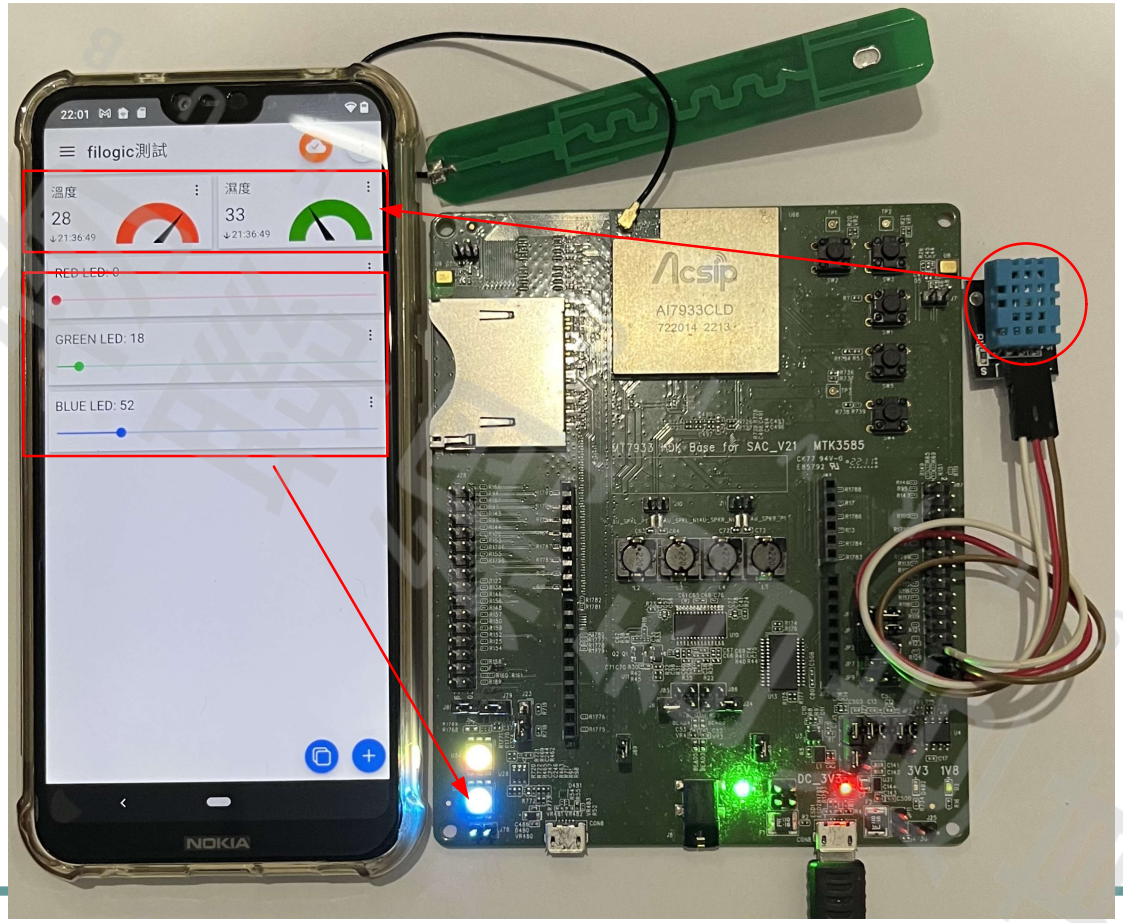
周邊介面與範例說明：MQTT 訂閱/發佈(1)

- MQTT介紹



周邊介面與範例說明：MQTT 訂閱/發佈(2)

- 硬體連線
- MQTT 操作



周邊介面與範例說明：MQTT 訂閱/發佈(3)

- 程式說明1

test_mqtt | Arduino 1.8.18

File Edit Sketch Tools Help

test_mqtt

```
#include <LWiFi.h>
#include <MQTTPubSubClient.h>
#include "DHT.h"
#include <FSPI.h>

/* SPI LED0 & LED1 控制的資料格式 */
typedef struct led_cmd
{
    uint32_t    start;
    uint8_t     led1_global;
    uint8_t     led1_b;
    uint8_t     led1_g;
    uint8_t     led1_r;
    uint8_t     led2_global;
    uint8_t     led2_b;
    uint8_t     led2_g;
    uint8_t     led2_r;
    uint32_t    end;
} led_cmd_t;

led_cmd_t  led_cmd;     // 控制LED暫存區

#define DHTPIN 23       // 設定DHT11訊號pin
#define DHTTYPE DHT11   // DHT 11
DHT dht(DHTPIN, DHTTYPE);

char ssid[] = "XXXXX";     // WIFI ID
char pass[] = "XXXXX";   // WIFI 密碼

const char* mqttServer = "XXX.com.tw";    // MQTT伺服器位址
const char* mqttUserName = "XXX";         // 使用者名稱
const char* mqttPwd = "XXXX";             // MQTT密碼
const char* deviceID = "filogic00";       // 用戶端ID，隨意設定。

WiFiClient client;
MQTTPubSubClient mqtt;

void ledOff(){                          // 將LED off,並設定傳送的格式
    led_cmd.start       = 0x00000000;
    led_cmd.led1_global = 0xF0;
    led_cmd.led1_b      = 0x00;
    led_cmd.led1_g      = 0x00;
    led_cmd.led1_r      = 0x00;
    led_cmd.led2_global = 0xF0;
    led_cmd.led2_b      = 0x00;
    led_cmd.led2_g      = 0x00;
    led_cmd.led2_r      = 0x00;
    led_cmd.end         = 0xFFFFFFFF;
    FSPI.transfer(&led_cmd, sizeof(led_cmd));
}

void setLedColor(uint8_t r,uint8_t g,uint8_t b,int led=0){
    if (led==0) {
        led_cmd.led1_b      = b;
        led_cmd.led1_g      = g;
        led_cmd.led1_r      = r;
    } else {
        led_cmd.led2_b      = b;
        led_cmd.led2_g      = g;
        led_cmd.led2_r      = r;
    }
    FSPI.transfer(&led_cmd, sizeof(led_cmd));
}
```

周邊介面與範例說明：MQTT 訂閱/發佈(4)

- 程式說明2

```
65
66 uint8_t redLed,greenLed,blueLed;
67 void setup() {
68     FSPI.begin();
69     Serial.begin(115200);
70     ledOff();
71
72     Serial.println("WIFI 連接中");
73     WiFi.begin(ssid, pass);
74
75     while (WiFi.status() != WL_CONNECTED) {
76         delay(500);
77         setLedColor(0x0,0x50,0x0);
78         Serial.print(".");
79         delay(300);
80         setLedColor(0x0,0x0,0x50);
81     }
82     Serial.println("WIFI 連接上了,連接MQTT仲介伺服器中");
83     setLedColor(0x0,0x10,0x0);
84     dht.begin();
85     while (!client.connect(mqttServer, 1883)){
86         Serial.print("連接 ");
87         Serial.print(mqttServer);
88         Serial.println(" MQTT仲介伺服器.....");
89         delay(500);
90     }
91     Serial.println("Login MQTT仲介伺服器");
92     setLedColor(0x0,0x10,0x10);
93     mqtt.begin(client);
94     while (!mqtt.connect(deviceID,mqttUserName,mqttPwd)){
95         Serial.print("Login ");
96         Serial.print(mqttServer);
97         Serial.println(" MQTT仲介伺服器.....");
98         delay(500);
99     }
100    Serial.println("MQTT仲介伺服器 login ok");
```

```
101    setLedColor(0x0,0x10,0x0);
102    redLed=0;
103    greenLed=0;
104    blueLed=0;
105    String topic;
106    topic=(String)deviceID+"/rLed";
107    mqtt.subscribe(topic, [](const String& payload, const size_t size) {
108        Serial.print("RED LED: ");
109        Serial.println(payload);
110        redLed = 0xff & payload.toInt();
111        setLedColor(redLed,greenLed,blueLed,1);
112    });
113    topic=(String)deviceID+"/gLed";
114    mqtt.subscribe(topic, [](const String& payload, const size_t size) {
115        Serial.print("GREEN LED: ");
116        Serial.println(payload);
117        greenLed = 0xff & payload.toInt();
118        setLedColor(redLed,greenLed,blueLed,1);
119    });
120    topic=(String)deviceID+"/bLed";
121    mqtt.subscribe(topic, [](const String& payload, const size_t size) {
122        Serial.print("BLUE LED: ");
123        Serial.println(payload);
124        blueLed = 0xff & payload.toInt();
125        setLedColor(redLed,greenLed,blueLed,1);
126    });
127    delay(1500);
128 }
129
```

周邊介面與範例說明：MQTT 訂閱/發佈(5)

- 程式說明3

APIs

```
void begin(ClientType& client);
bool connect(const String& client_id, const String& user = "", const String& pass = "");
bool disconnect();
bool update();

bool publish(const String& topic, const String& payload, const bool retained = false, int qos = 0);
bool publish(const String& topic, uint8_t* payload, const size_t length, const bool retained = false, const u

void subscribe(const global_callback_t& cb);
bool subscribe(const String& topic, const topic_callback_t& cb);
bool subscribe(const String& topic, const uint8_t qos, const topic_callback_t& cb);
bool unsubscribe(const String& topic);
```

周邊介面與範例說明：MQTT 訂閱/發佈(6)

- 程式說明4

```
129
130 void loop() {
131     mqtt.update();
132     float h = dht.readHumidity();          //讀出溫濕度
133     float t = dht.readTemperature();
134     if (isnan(h) || isnan(t)) {            //檢查讀出溫濕度是否有效
135         Serial.println(F("Failed to read from DHT sensor!"));
136         return;
137     }
138
139     Serial.print(F("濕度: "));
140     Serial.print(h);
141     Serial.print(F("%  溫度: "));
142     Serial.print(t);
143     Serial.println(F("°C "));
144
145     String top,msg;                                  // 發佈溫濕度
146     top=(String)deviceID+"/Humidity";
147     msg=String(round(h));
148     mqtt.publish(top.c_str(), msg.c_str());
149     top=(String)deviceID+"/Temperature";
150     msg=String(round(t));
151     mqtt.publish(top.c_str(), msg.c_str());
152
153     delay(500);
154     setLedColor(0x20,0x0,0x20);
155     delay(500);
156     setLedColor(0x20,0x20,0x5);
157     delay(500);
158     setLedColor(0x0,0x0,0x0);
159     delay(1000);
```

周邊介面與範例說明：MQTT 訂閱/發佈(7)

- Android APP

周邊介面與範例說明：MQTT 訂閱/發佈(8)

- Apple Iphone APP ?