國產IC開發套件
Filogic 130 (MT7933)
示範案例

溫濕度感測器 DHT11 + ThingSpeak Cloud

大綱

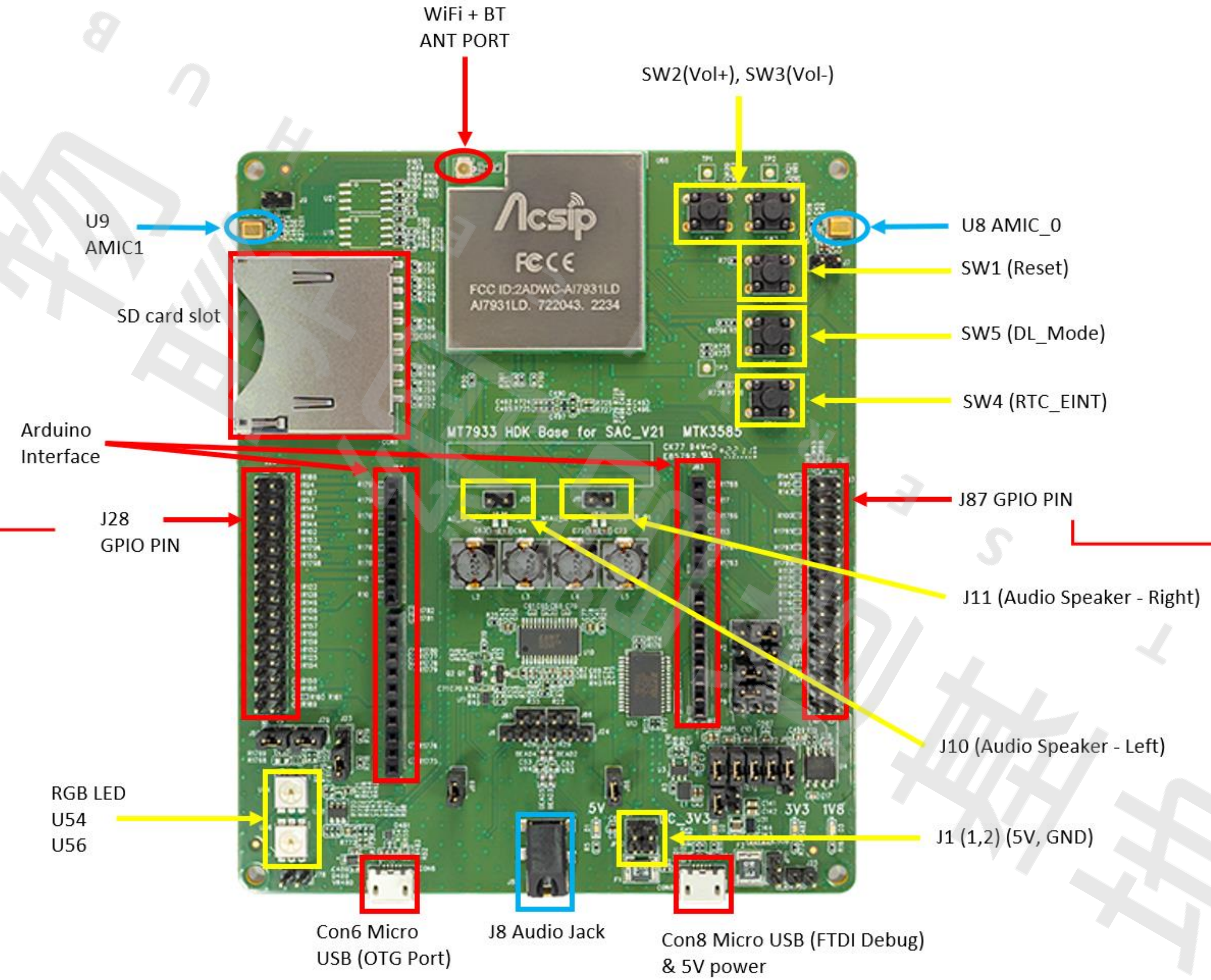
- 開發板Filogic130 介紹
- 開發環境 （Arduino SDK安裝與韌體上傳）
- 周邊介面與範例說明：
  - ◆ 溫溼度感測模組 DHT11
  - ◆ ThingSpeak
  - ◆ DHT11 ＋ ThingSpeak

開發板硬體外觀

J28 GPIO

1			2
	NC	6	
	NC	7	
	NC	8	
	NC	9	
	NC	10	
	NC	11	
	GND	GND	
	NC	C_12	
	NC	NC	
	NC	NC	
	NC	NC	
	NC	NC	
	NC	GND	
	NC	44	
29	NC	42	30

GPIO pin-put extension connectors



J87 GPIO

1			2
	RST	C_15	
	GND	C_13	
	12	48	
	13	50	
	14	GND	
	15	24	
	17	NC	
	18	NC	
	19	NC	
	20	NC	
	21	VCCIO	
	22	GND	
	23	3V3	
	GND	GND	
29	1V8-OUT	5V-OUT	30

GPIO pin-put extension connectors

GPIO pin-out extension connector

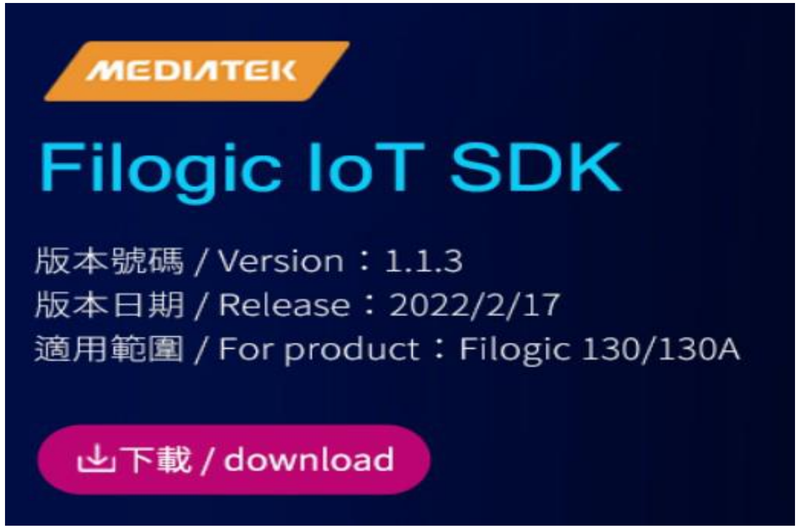
Signal Name	Connector Pin Number	Signal Name	Connector Pin Number
GPIO_0	Reserve for flash	GPIO_14	J87 – 9
GPIO_1	Reserve for flash	GPIO_15	J87 – 11
GPIO_2	Reserve for flash	GPIO_17	J87 – 13
GPIO_3	Reserve for flash	GPIO_18	J87 – 15
GPIO_4	Reserve for flash	GPIO_19	J87 – 17 Reserve for Arduino:I2C1_SDA
GPIO_5	Reserve for flash	GPIO_20	J87 – 19 Reserve for Arduino:I2C1_SCL
GPIO_6	J28 – 2 Reserve for Arduino:SPI0_CSK	GPIO_21	J87 – 21
GPIO_7	J28 – 4 Reserve for Arduino:SPI0_CSN	GPIO_22	J87 – 23
GPIO_8	J28 – 6 Reserve for Arduino:SPI0_MSIO	GPIO_23	J87 – 25
GPIO_9	J28 – 8 Reserve for Arduino:SPI0_MOSI	GPIO_24	J87 – 12
GPIO_10	J28 – 10	GPIO_42	J28 – 30 Reserve for Arduino:UART1_RX
GPIO_11	J28 – 12	GPIO_44	J28 – 2 Reserve for Arduino:UART1_TX
GPIO_12	J87 – 5	GPIO_48	J28 – 2 Reserve for CM33 UART
GPIO_13	J87 – 7	GPIO_50	J28 – 2 Reserve for CM33 UART

Source: https://www.acsip.com.tw/index.php?action=download_pro&perm=d&id=527

開發環境建制

Filogic SDK

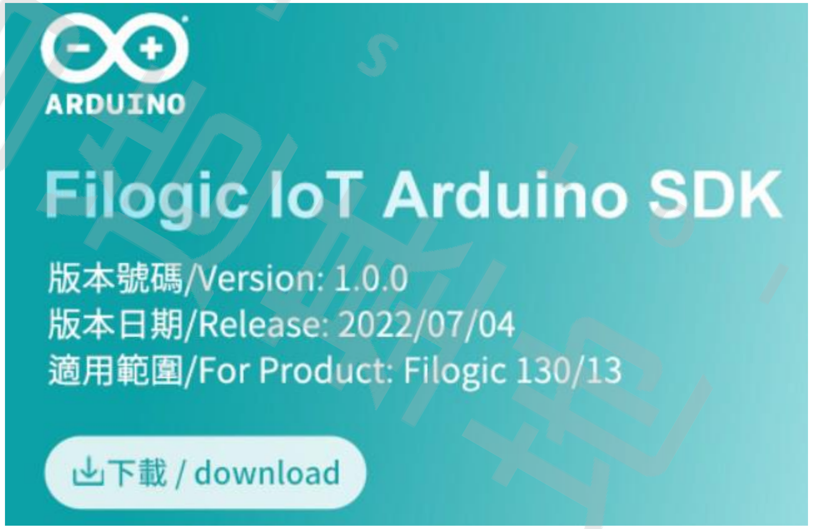
- 大聯大的大大通網站 www.wpgdadatong.com
  - ◆ 技術論壇：提問、討論、文件、技術分享...等。
- 建立開發環境
  - ◆ IOT SDK (Linux編譯環境)：基於Ubuntu v20.04
  - ◆ Arduino SDK：Arduino IDE v1.6.13以上



MEDIATEK

Filogic IoT SDK

版本號碼 / Version：1.1.3
版本日期 / Release：2022/2/17
適用範圍 / For product：Filogic 130/130A



ARDUINO

Filogic IoT Arduino SDK

版本號碼/Version: 1.0.0
版本日期/Release: 2022/07/04
適用範圍/For Product: Filogic 130/13

Arduino IDE 編譯環境

- **優點:**
  - ✓ 快速入手，編輯介面簡潔易懂
  - ✓ 建立統一的編譯環境
  - ✓ 在軟體架構下，更多市面的模組能支援各類開發板
  - ✓ 網路資源豐富，開發過程更為便利與快速

- **操作前的準備**
  - ◆ 安裝Arduino IDE v1.6.13以上，勿用v2.0

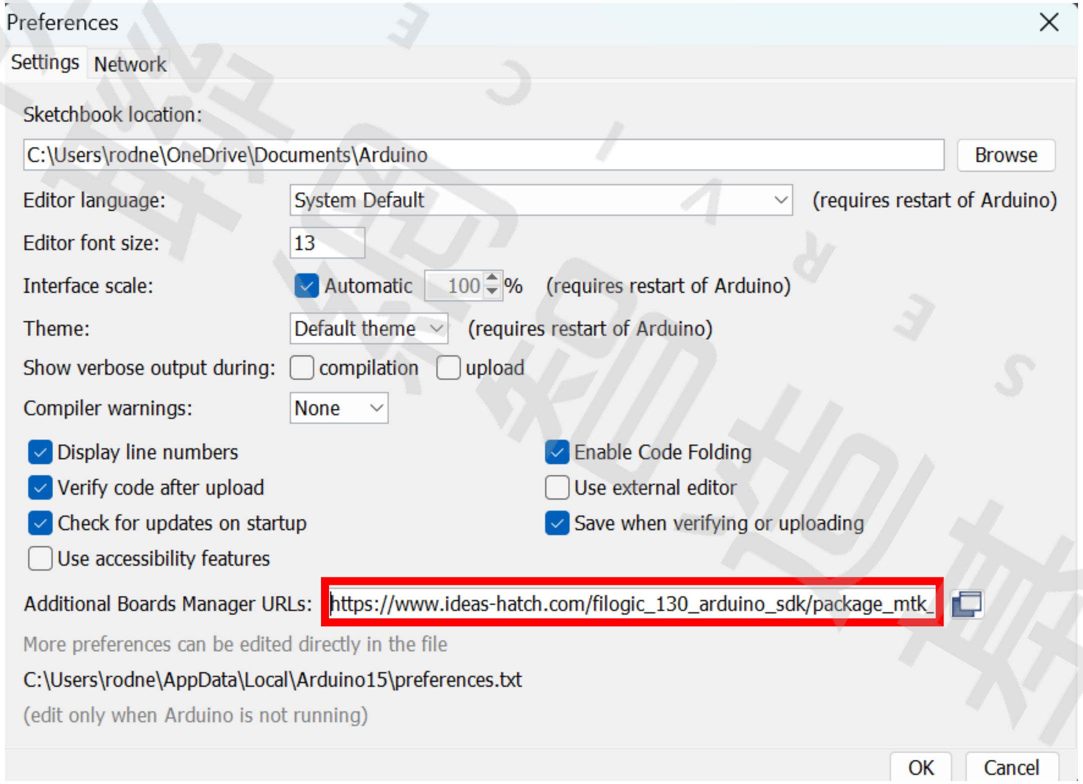
Filogic SDK 下載

- 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



Preferences

Settings | Network

Sketchbook location:

C:\Users\rodne\OneDrive\Documents\Arduino [Browse]

Editor language: System Default (requires restart of Arduino)

Editor font size: 13

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Theme: Default theme (requires restart of Arduino)

Show verbose output during: □ compilation □ upload

Compiler warnings: None

☑ Display line numbers
☑ Verify code after upload
☑ Check for updates on startup
□ Use accessibility features

☑ Enable Code Folding
□ Use external editor
☑ Save when verifying or uploading

Additional Boards Manager URLs: https://www.ideas-hatch.com/filogic_130_arduino_sdk/package_mtk_

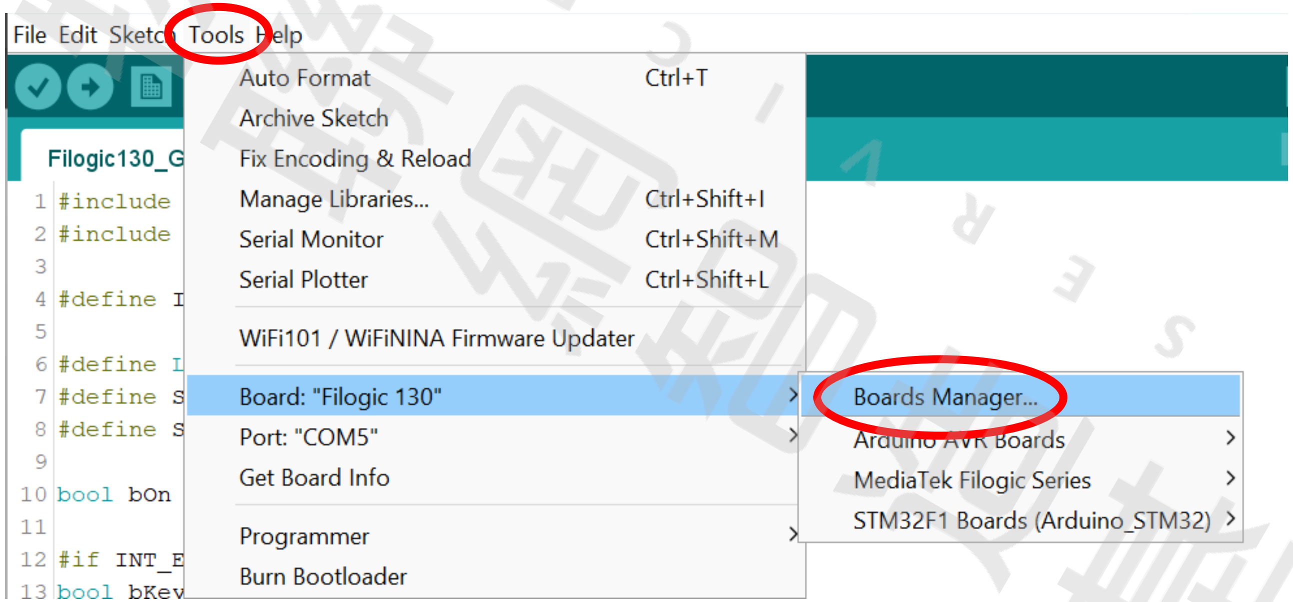
More preferences can be edited directly in the file

C:\Users\rodne\AppData\Local\Arduino15\preferences.txt

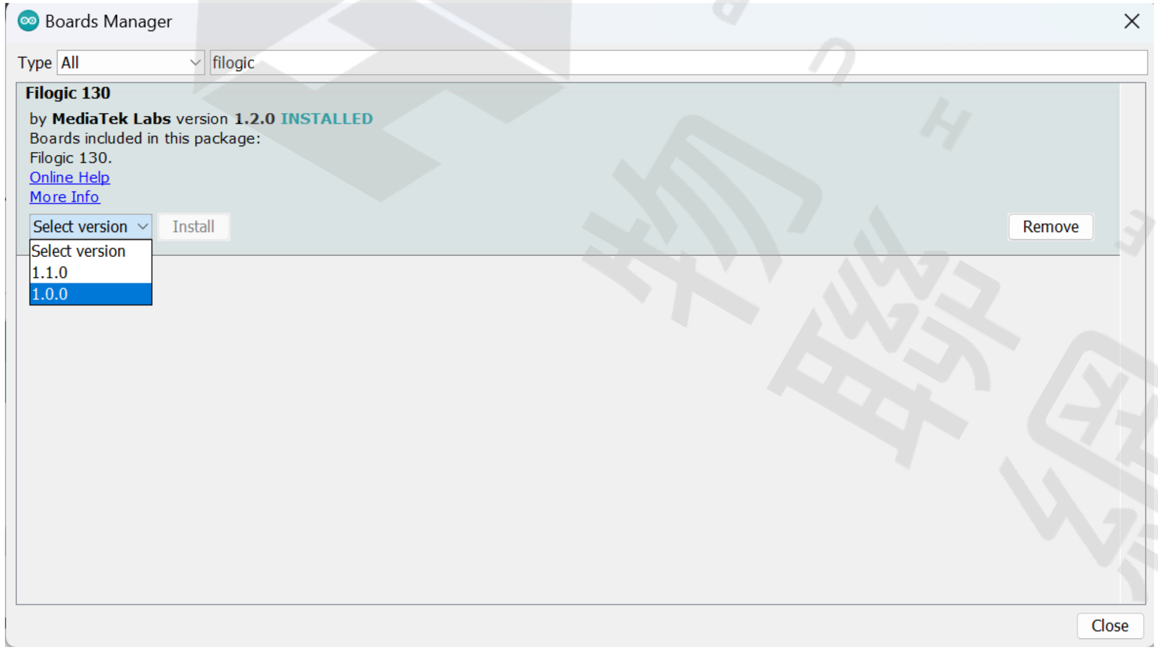
(edit only when Arduino is not running)

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

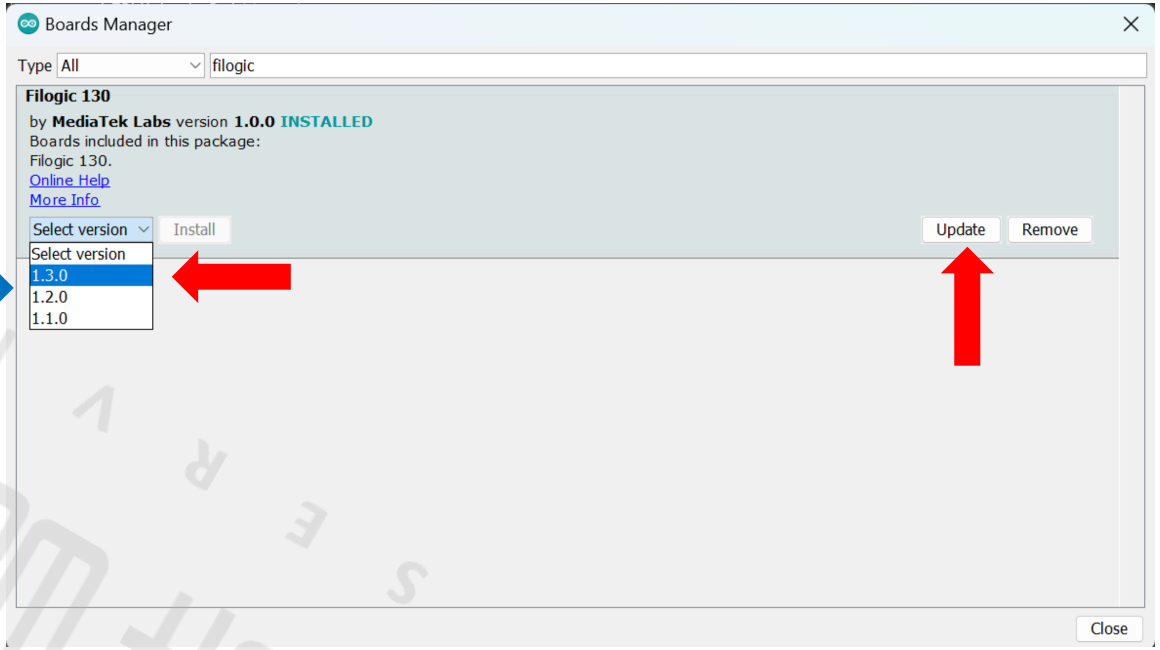


安裝步驟說明（3）


Boards Manager

Type All filogic

Filogic 130
by **MediaTek Labs** version **1.2.0** **INSTALLED**
Boards included in this package:
Filogic 130.
Online Help
More Info

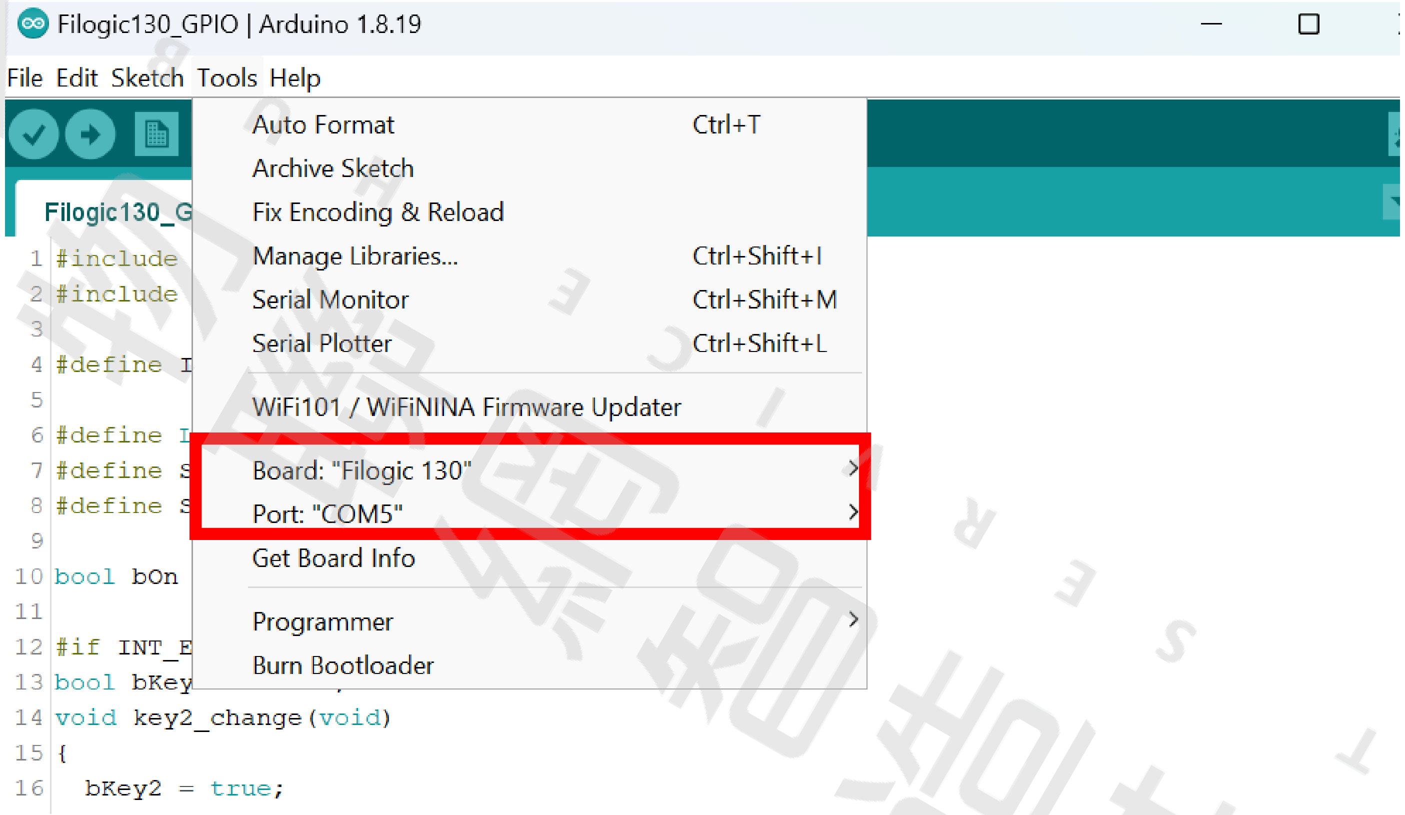

Boards Manager

Type All filogic

Filogic 130
by **MediaTek Labs** version **1.0.0** **INSTALLED**
Boards included in this package:
Filogic 130.
Online Help
More Info

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）


Filogic130_GPIO | Arduino 1.8.19
File Edit Sketch Tools Help
Filogic130_G
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Get Board Info
Programmer
Burn Bootloader

Board: "Filogic 130"

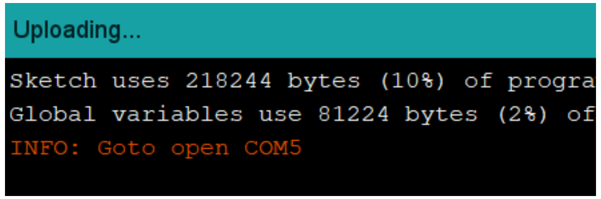
Port: "COM5"

```
#include
#include

#define I

#define I
#define S
#define S

bool bOn

#if INT_E
bool bKey
void key2_change(void)
{
  bKey2 = true;
```

- 安裝完畢後確認Board 與 Port

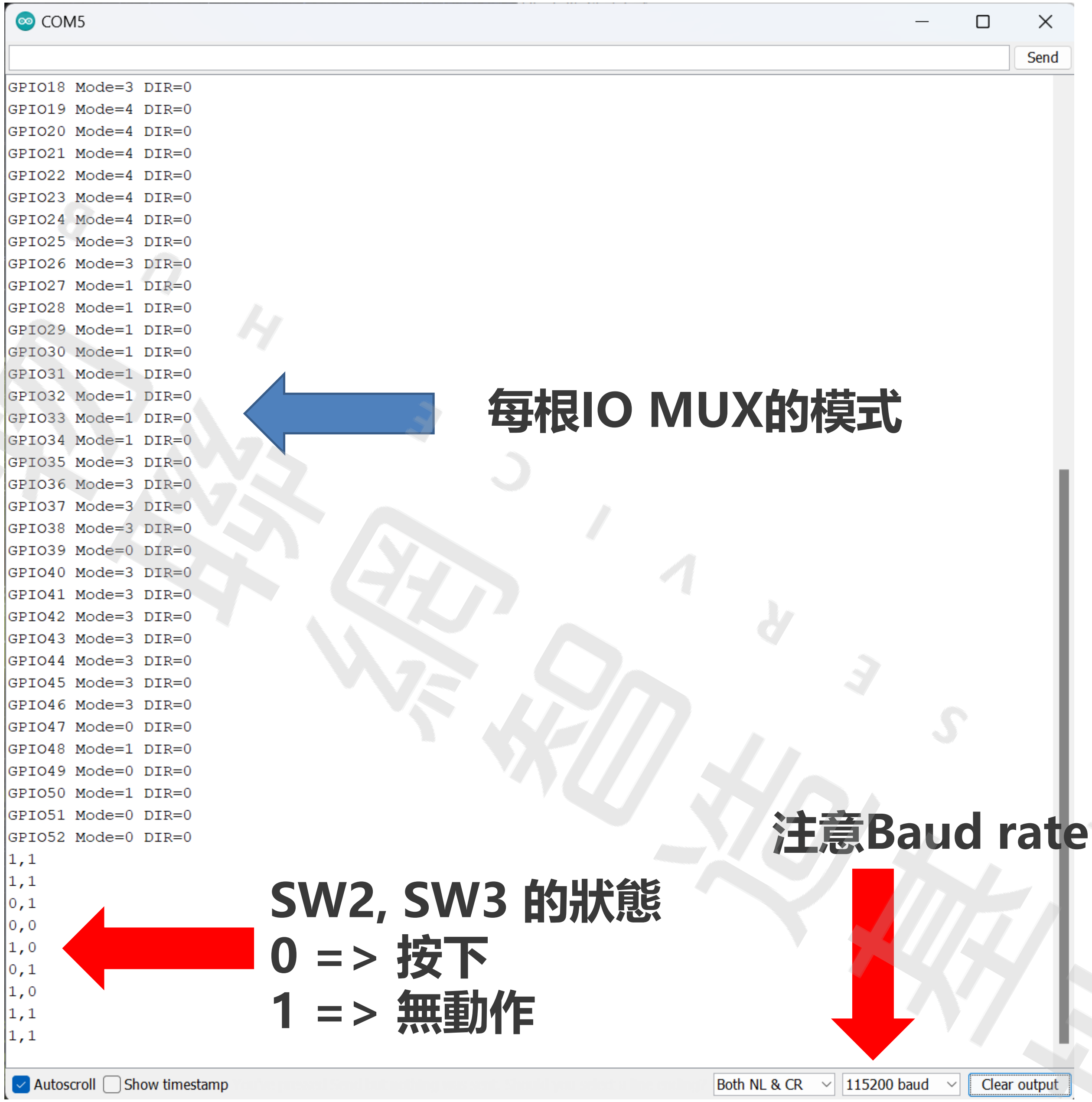
安裝步驟說明（4）

- 下載測試程式 - GPIO
  - https://github.com/yijenlu1971/Filogic_onArduino/tree/main/GPIO



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

- 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。



```
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
```

☑ Autoscroll ☐ Show timestamp   Both NL & CR   115200 baud

每根IO MUX的模式

SW2, SW3 的狀態
0 => 按下
1 => 無動作

注意Baud rate!

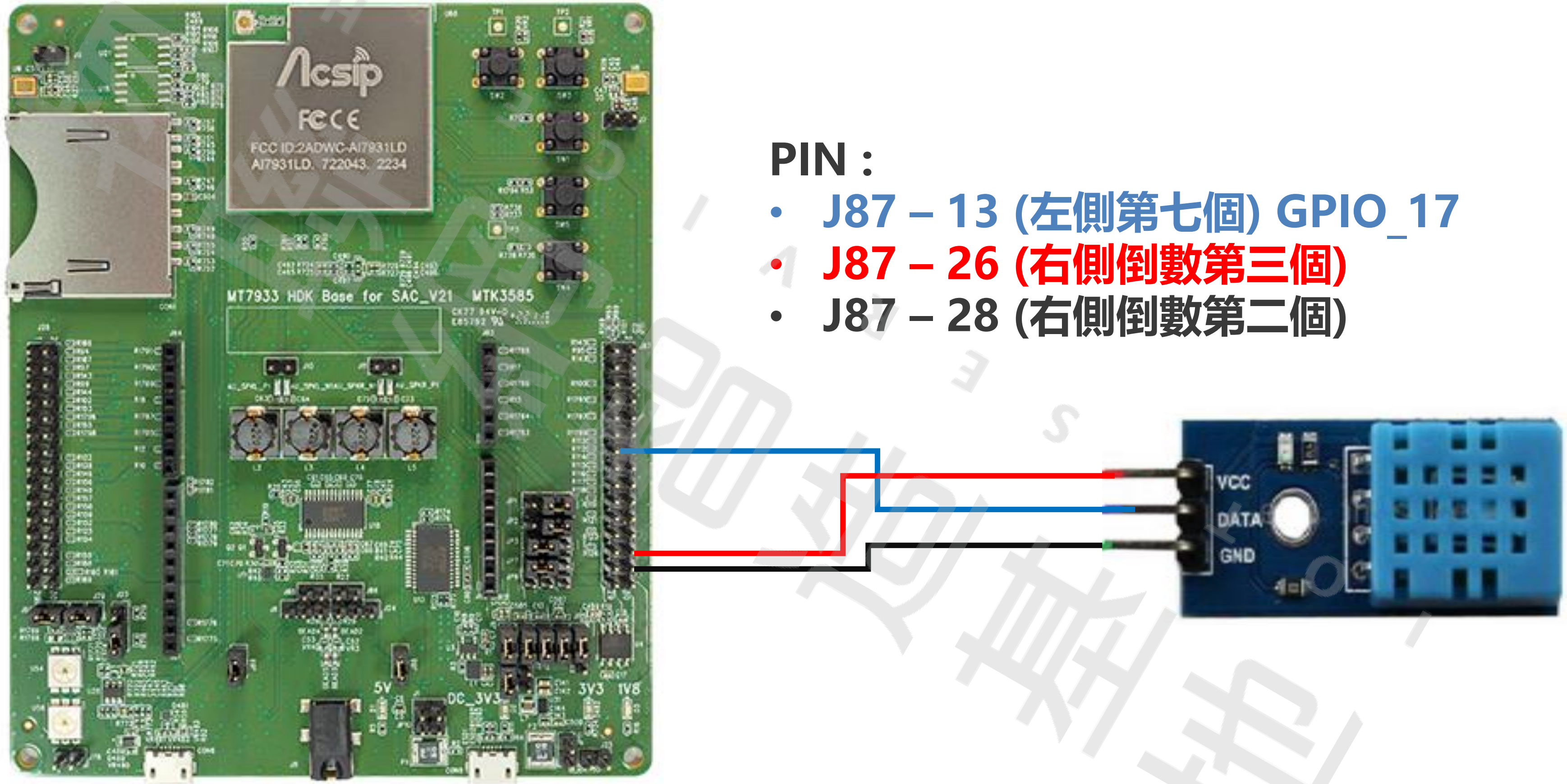
示範案例

DHT11 溫度與溼度感測器

- DHT11是一個結合濕度計和測溫元件量測週遭空氣環境。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)

電路圖

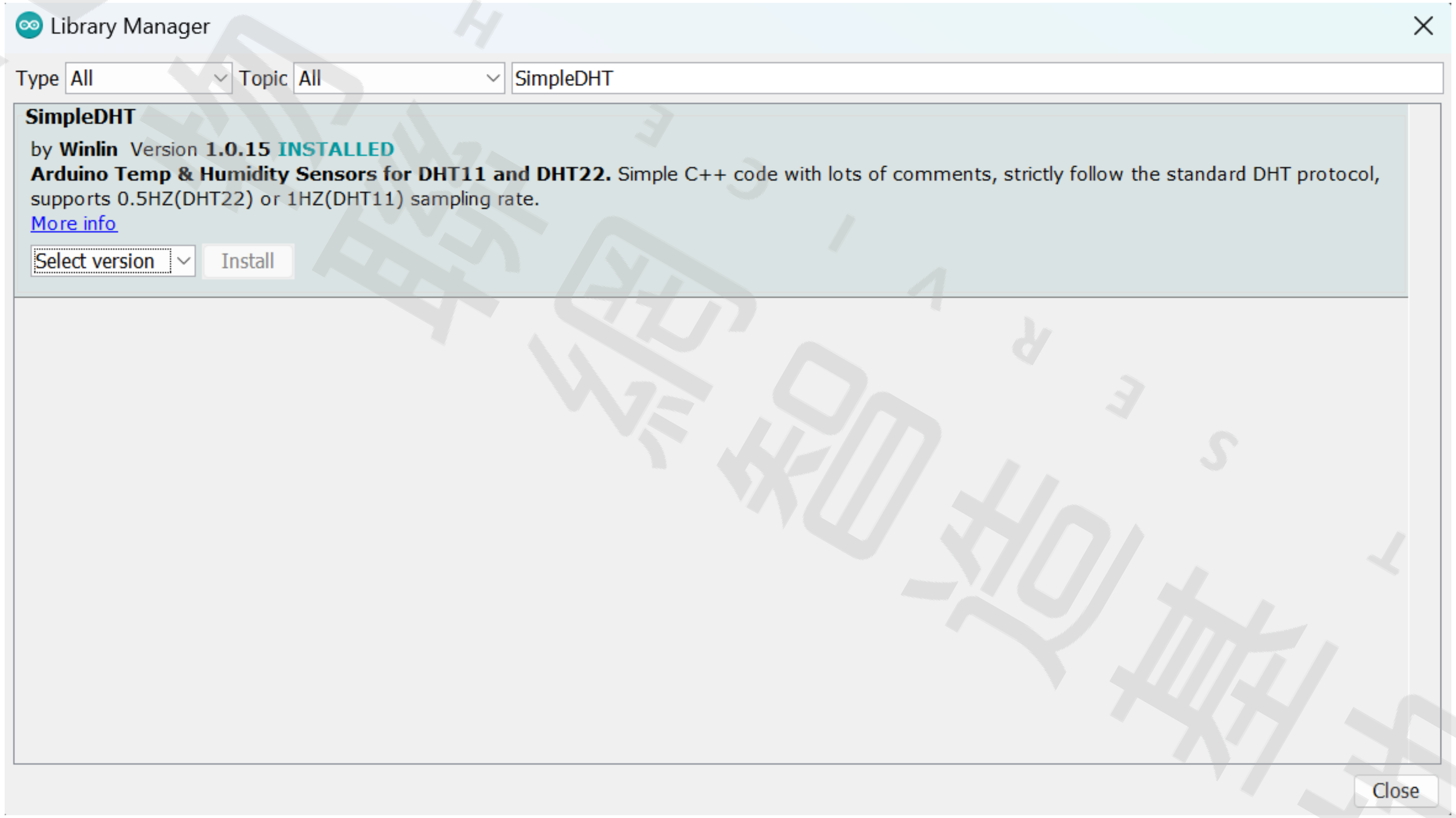


PIN :

- J87 – 13 (左側第七個) GPIO_17
- J87 – 26 (右側倒數第三個)
- J87 – 28 (右側倒數第二個)

安裝SimpleDHT Library

- 開啟IDE的“開發板管理員”(Sketch > Include Library > Manage Libraries)
- 安裝SimpleDHT Library



SimpleDHT

SimpleDHT

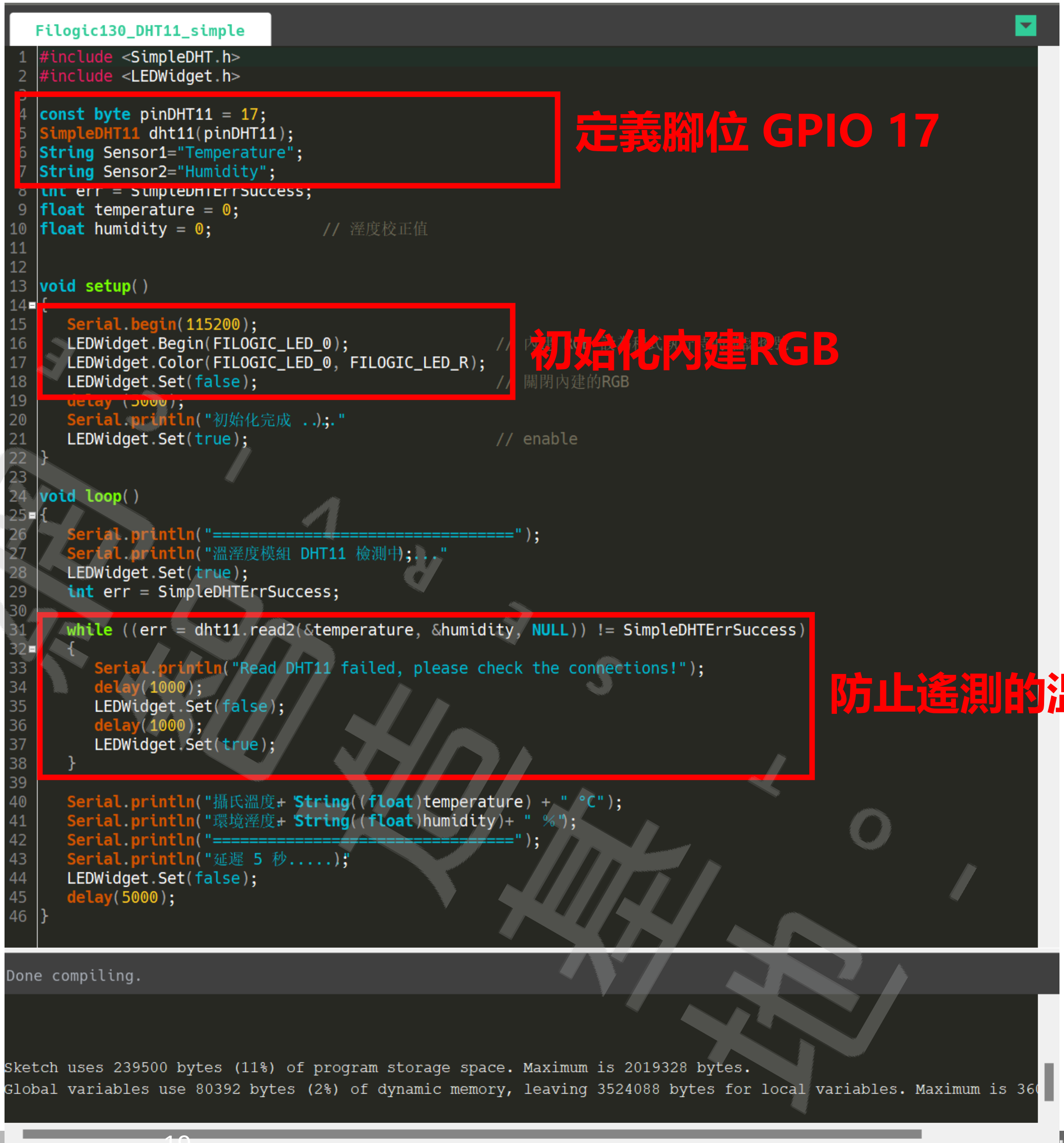
by **Winlin** Version **1.0.15** **INSTALLED**

Arduino Temp & Humidity Sensors for DHT11 and DHT22. Simple C++ code with lots of comments, strictly follow the standard DHT protocol, supports 0.5HZ(DHT22) or 1HZ(DHT11) sampling rate.

More info

程式簡說

- 監測溫濕度時，板子上內建的RGB會亮起。
- 每五秒監測一次。



Filogic130_DHT11_simple

```
#include <SimpleDHT.h>
#include <LEDWidget.h>

const byte pinDHT11 = 17;
SimpleDHT11 dht11(pinDHT11);
String Sensor1="Temperature";
String Sensor2="Humidity";
int err = SimpleDHTErrSuccess;
float temperature = 0;
float humidity = 0;            // 溼度校正值


void setup()
{
  Serial.begin(115200);
  LEDWidget.Begin(FILOGIC_LED_0);                          // 內建RGB設為程式執行時的確認燈號
  LEDWidget.Color(FILOGIC_LED_0, FILOGIC_LED_R);
  LEDWidget.Set(false);                                    // 關閉內建的RGB
  delay (3000);
  Serial.println("初始化完成 ..);."
  LEDWidget.Set(true);                                     // enable
}

void loop()
{
  Serial.println("=================================");
  Serial.println("溫溼度模組 DHT11 檢測中);..."
  LEDWidget.Set(true);
  int err = SimpleDHTErrSuccess;

  while ((err = dht11.read2(&temperature, &humidity, NULL)) != SimpleDHTErrSuccess)
  {
    Serial.println("Read DHT11 failed, please check the connections!");
    delay(1000);
    LEDWidget.Set(false);
    delay(1000);
    LEDWidget.Set(true);
  }

  Serial.println("攝氏溫度+ String((float)temperature) + " °C");
  Serial.println("環境溼度+ String((float)humidity)+ " %");
  Serial.println("=================================");
  Serial.println("延遲 5 秒.....);"
  LEDWidget.Set(false);
  delay(5000);
}
```

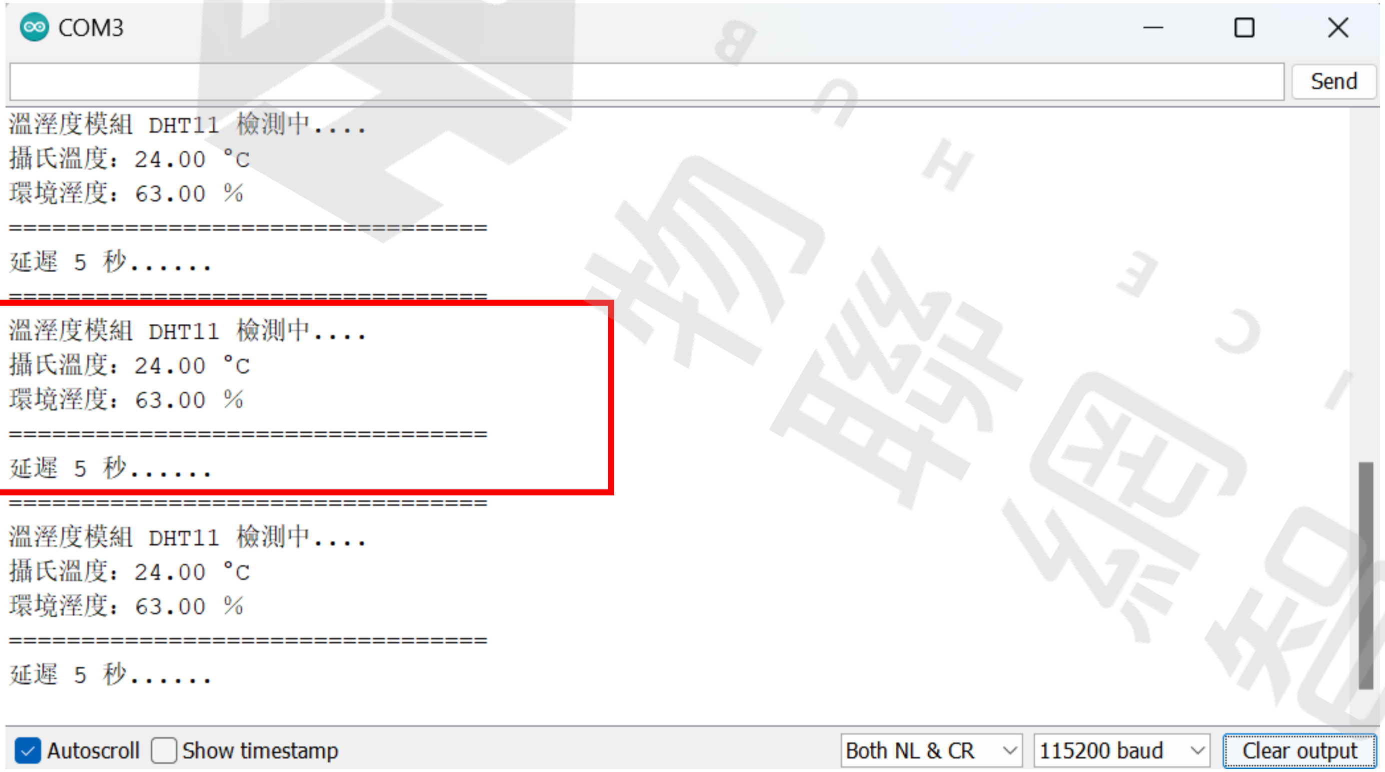
```
Done compiling.

Sketch uses 239500 bytes (11%) of program storage space. Maximum is 2019328 bytes.
Global variables use 80392 bytes (2%) of dynamic memory, leaving 3524088 bytes for local variables. Maximum is 36
```

定義腳位 GPIO 17

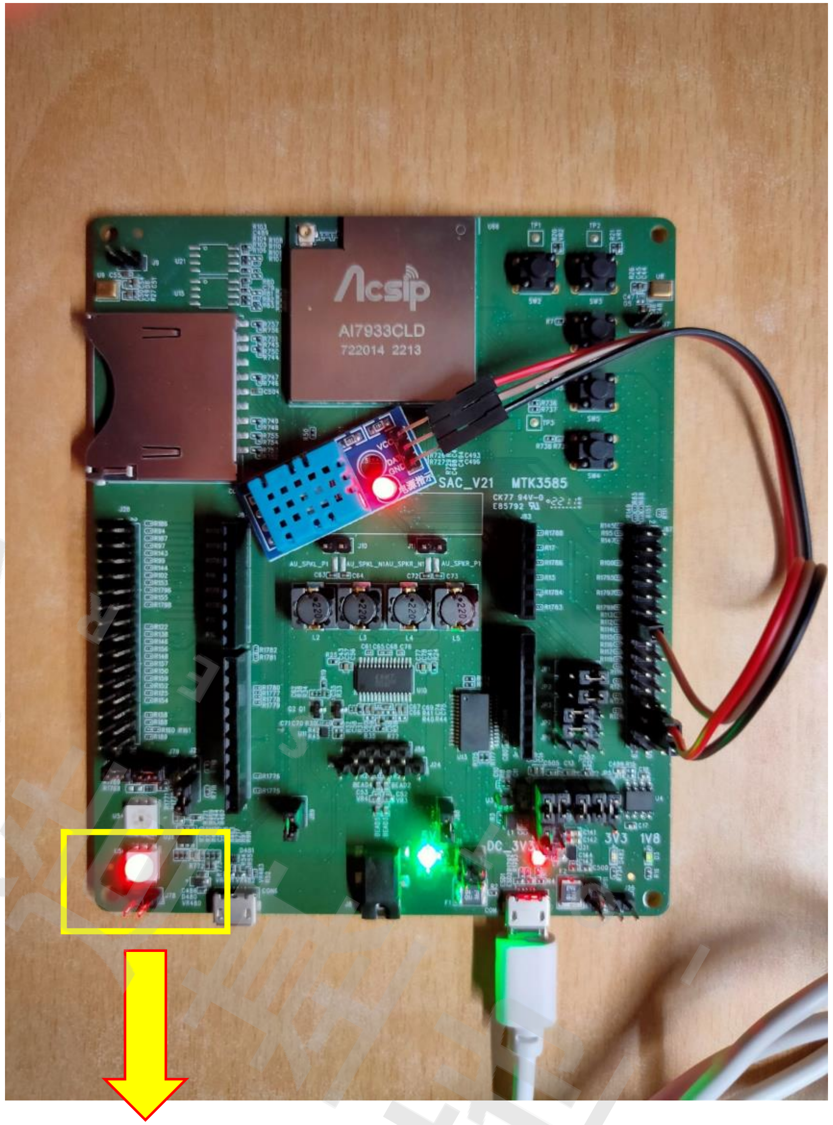
初始化內建RGB

防止遙測的溫濕度為空

執行結果



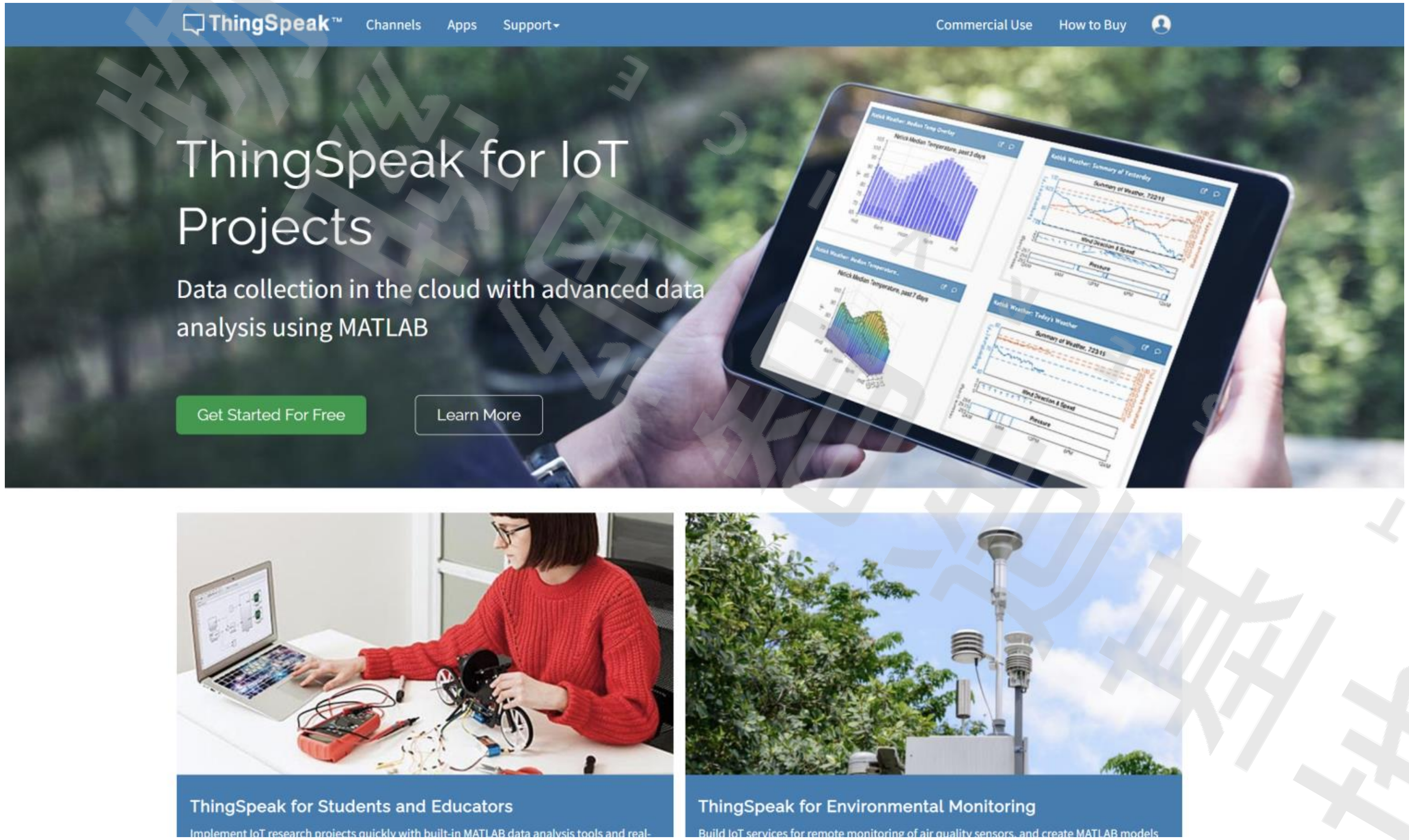
```
溫溼度模組 DHT11 檢測中....
攝氏溫度: 24.00 °C
環境溼度: 63.00 %
=================================
延遲 5 秒......
=================================
溫溼度模組 DHT11 檢測中....
攝氏溫度: 24.00 °C
環境溼度: 63.00 %
=================================
延遲 5 秒......
=================================
溫溼度模組 DHT11 檢測中....
攝氏溫度: 24.00 °C
環境溼度: 63.00 %
=================================
延遲 5 秒......
```



監測時會亮起RGB燈

ThingSpeak IoT數據平台

1. 在ThingSpeak上創建一個帳戶。
   https://thingspeak.com/



ThingSpeak for IoT Projects

Data collection in the cloud with advanced data analysis using MATLAB

ThingSpeak for Students and Educators

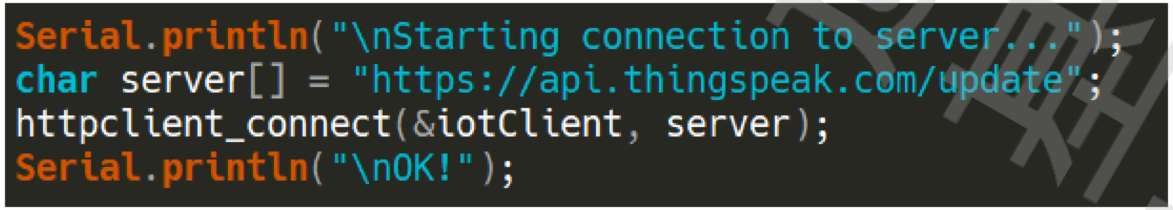
Implement IoT research projects quickly with built-in MATLAB data analysis tools and real-

ThingSpeak for Environmental Monitoring

Build IoT services for remote monitoring of air quality sensors, and create MATLAB models

ThingSpeak

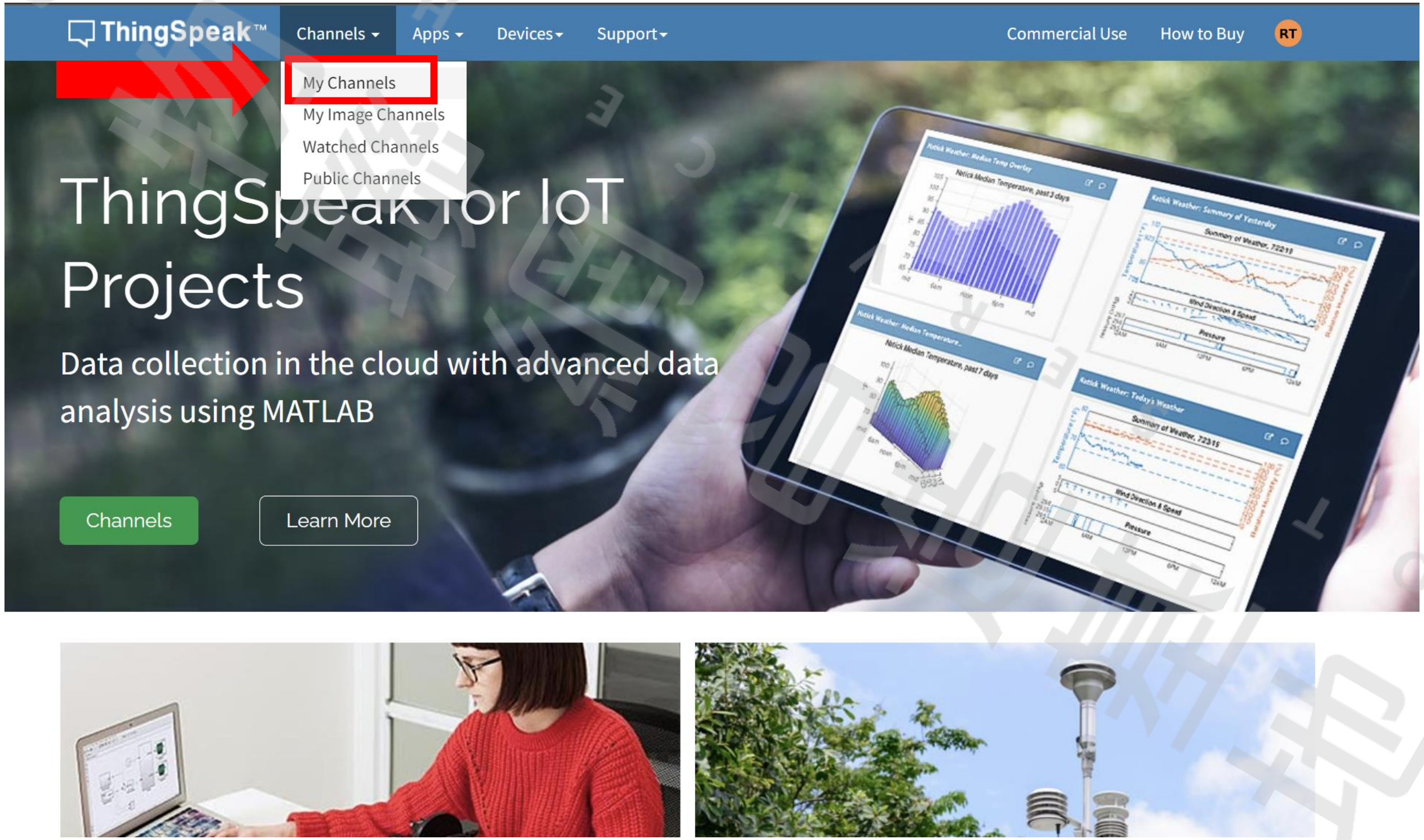
- ThingSpeak 是一種物聯網分析平台服務，提供雲端整合、數據視覺化和分析實時數據流。
- ThingSpeak 可將感測設備的數據即時視覺化。
- ThingSpeak 提供REST API 和MQTT API 兩種方式：
  - 本範例以REST為主。
- 本範例HTTP POST方式連線ThingSpeak 網站，建立連線httpclient_connect，如下



```
Serial.println("\nStarting connection to server...");
char server[] = "https://api.thingspeak.com/update";
httpclient_connect(&iotClient, server);
Serial.println("\nOK!");
```

ThingSpeak – New Channel

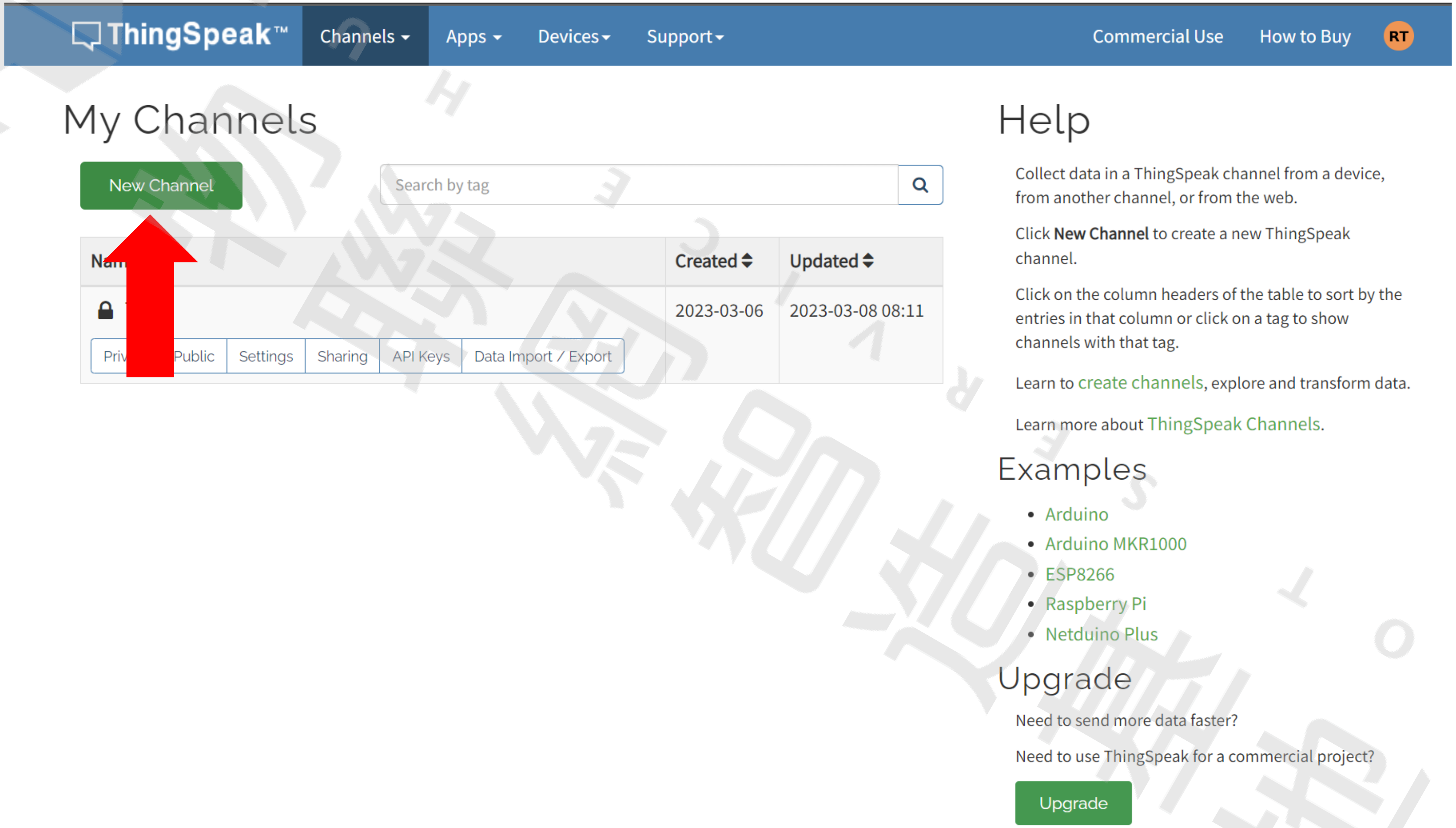
2. 建立新頻道(Channel)並添加頻道名稱(Name)和描述(Description)。



ThingSpeak for IoT Projects

Data collection in the cloud with advanced data analysis using MATLAB

ThingSpeak – Create New Channel



My Channels

Name	Created	Updated
🔒	2023-03-06	2023-03-08 08:11

Help

Collect data in a ThingSpeak channel from a device, from another channel, or from the web.

Click **New Channel** to create a new ThingSpeak channel.

Click on the column headers of the table to sort by the entries in that column or click on a tag to show channels with that tag.

Learn to create channels, explore and transform data.

Learn more about ThingSpeak Channels.

Examples

- Arduino
- Arduino MKR1000
- ESP8266
- Raspberry Pi
- Netduino Plus

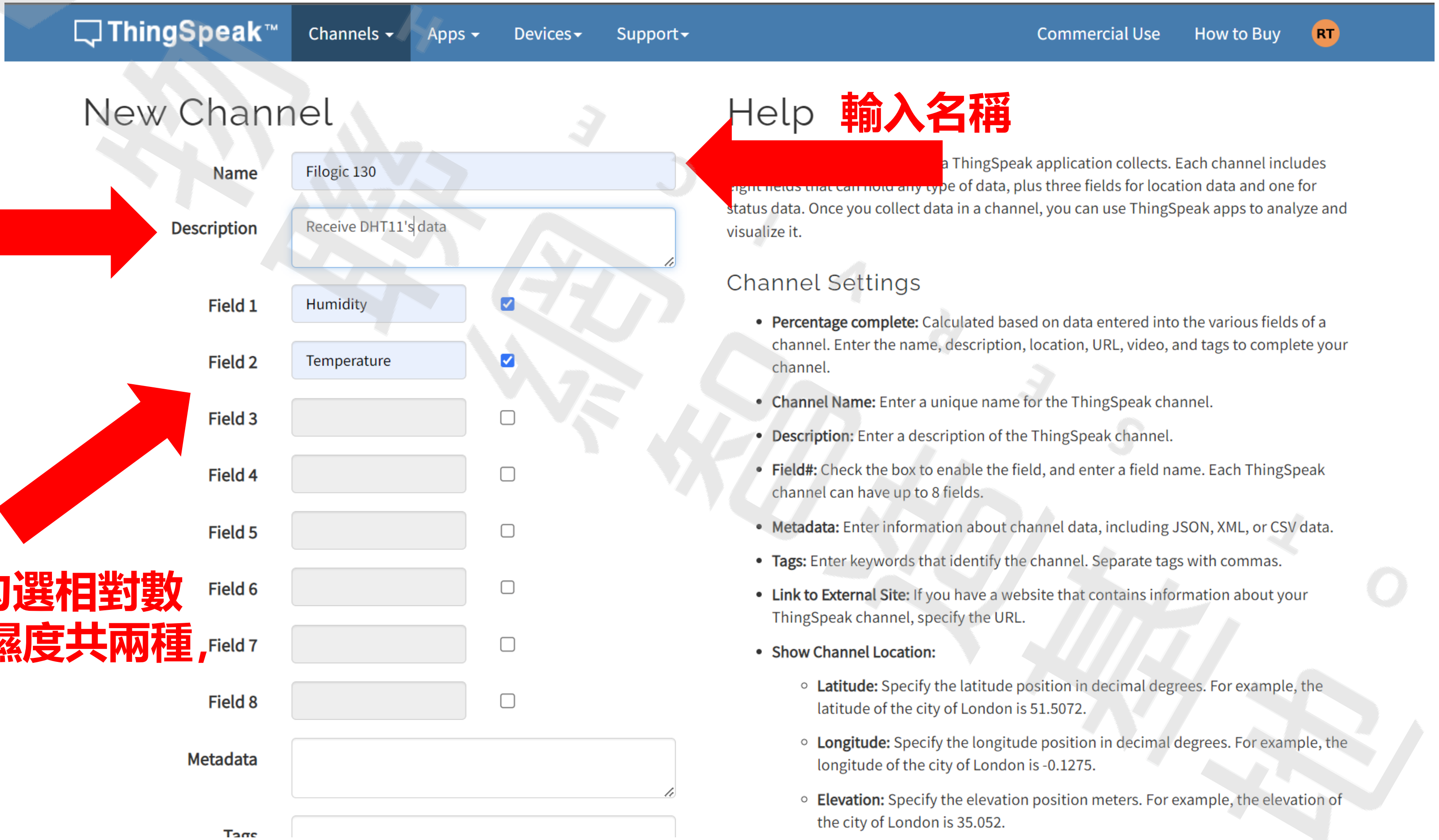
Upgrade

Need to send more data faster?

Need to use ThingSpeak for a commercial project?

ThingSpeak – Create New Channel

3. 設置頻道。 在本範例中，有兩個數據字段，設置兩個Fields。



New Channel

Name: Filogic 130

Description: Receive DHT11's data

Field 1: Humidity ☑

Field 2: Temperature ☑

Field 3: □

Field 4: □

Field 5: □

Field 6: □

Field 7: □

Field 8: □

Metadata:

Tags

Help

a ThingSpeak application collects. Each channel includes ight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

Channel Settings

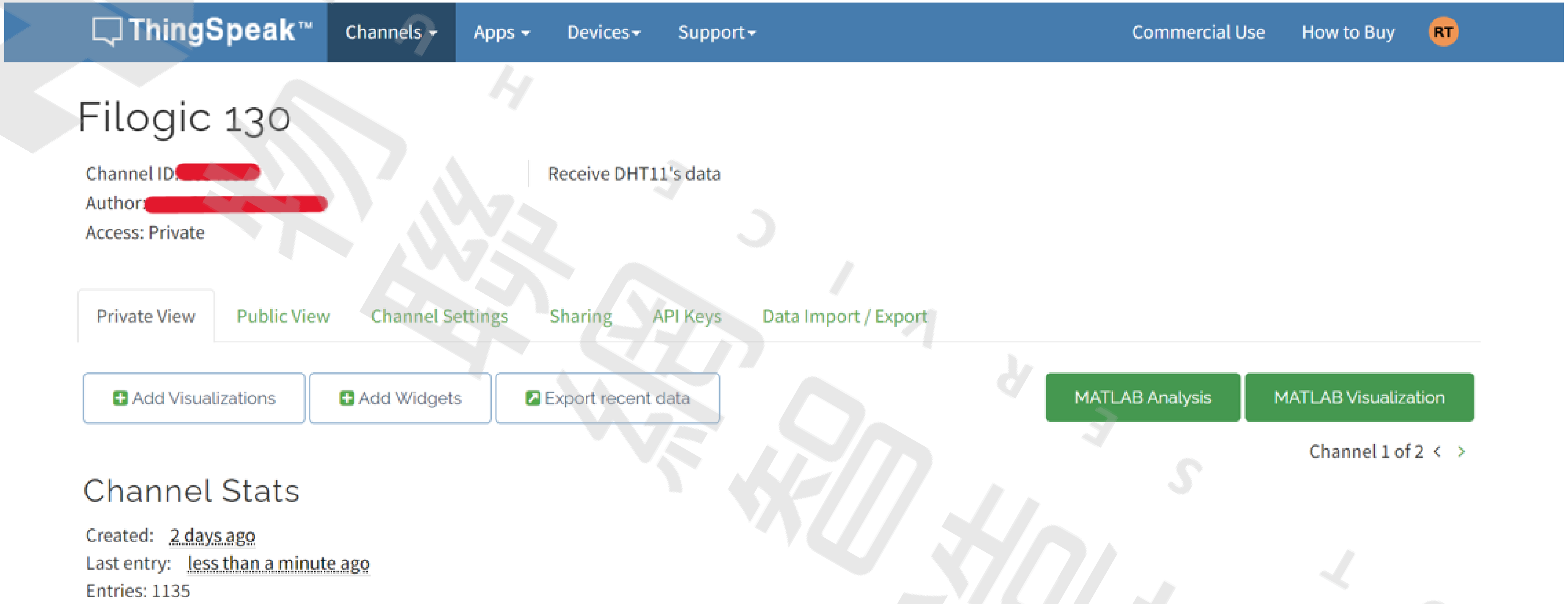
- **Percentage complete:** Calculated based on data entered into the various fields of a channel. Enter the name, description, location, URL, video, and tags to complete your channel.
- **Channel Name:** Enter a unique name for the ThingSpeak channel.
- **Description:** Enter a description of the ThingSpeak channel.
- **Field#:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- **Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- **Tags:** Enter keywords that identify the channel. Separate tags with commas.
- **Link to External Site:** If you have a website that contains information about your ThingSpeak channel, specify the URL.
- **Show Channel Location:**
  - **Latitude:** Specify the latitude position in decimal degrees. For example, the latitude of the city of London is 51.5072.
  - **Longitude:** Specify the longitude position in decimal degrees. For example, the longitude of the city of London is -0.1275.
  - **Elevation:** Specify the elevation position meters. For example, the elevation of the city of London is 35.052.

輸入名稱

輸入敘述

- 勾選Field 並命名
- 有幾筆不同數據就勾選相對數量的Fields, ex: 溫濕度共兩種，勾選兩個Fields。

ThingSpeak – New Channel Created



Filogic 130

Channel ID: [redacted]
Author: [redacted]
Access: Private

Receive DHT11's data

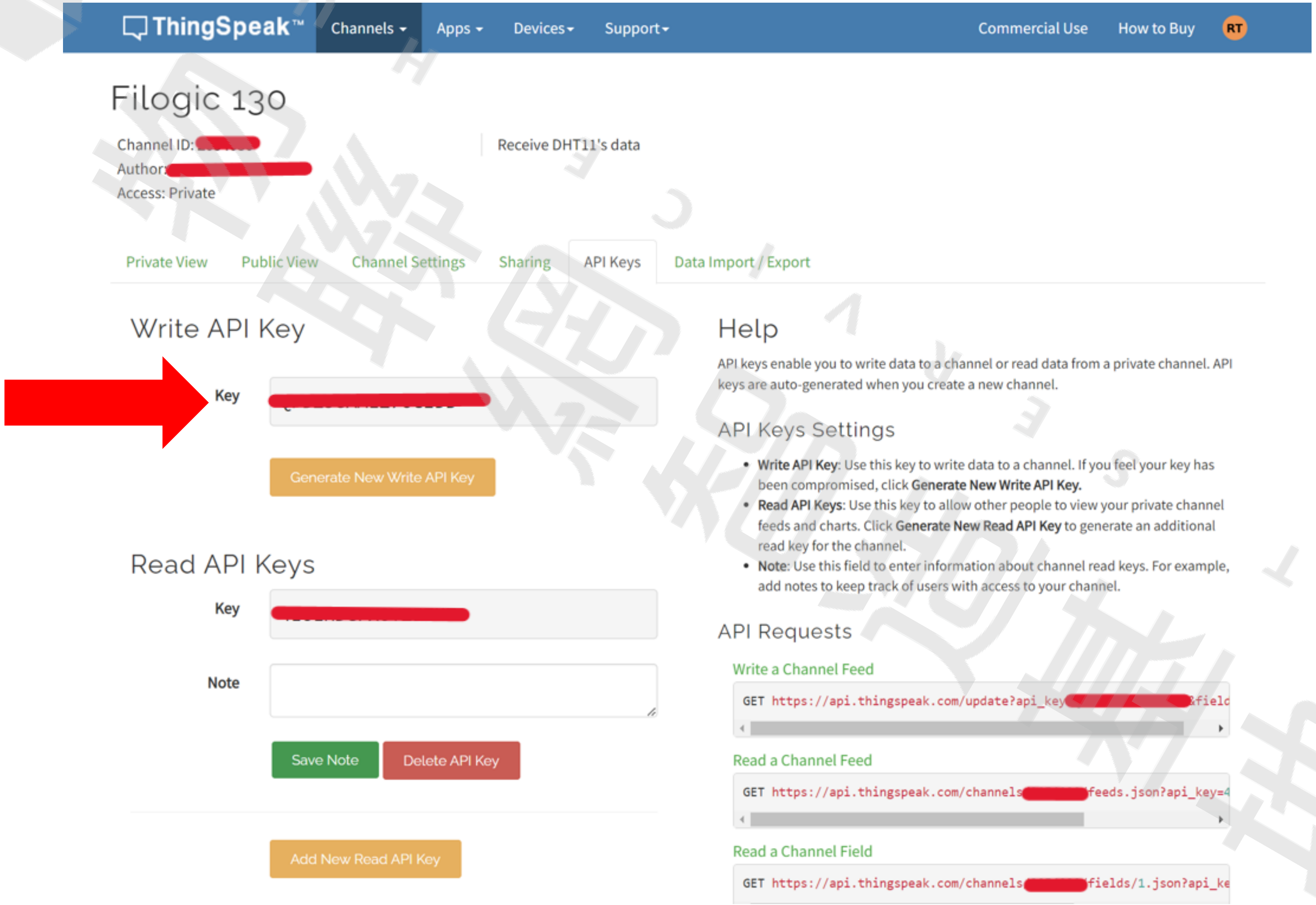
Channel Stats

Created: 2 days ago
Last entry: less than a minute ago
Entries: 1135

ThingSpeak – Get API Keys

4. 選擇" API Keys"，然後複製Write API Key。

複製API Key



Filogic 130

Channel ID: [redacted]
Author: [redacted]
Access: Private

Receive DHT11's data

Write API Key

Key [redacted]

Read API Keys

Key [redacted]

Note

Help

API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.

API Keys Settings

- **Write API Key**: Use this key to write data to a channel. If you feel your key has been compromised, click **Generate New Write API Key.**
- **Read API Keys**: Use this key to allow other people to view your private channel feeds and charts. Click **Generate New Read API Key** to generate an additional read key for the channel.
- **Note**: Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.

API Requests

Write a Channel Feed

```
GET https://api.thingspeak.com/update?api_key[redacted]&field
```

Read a Channel Feed

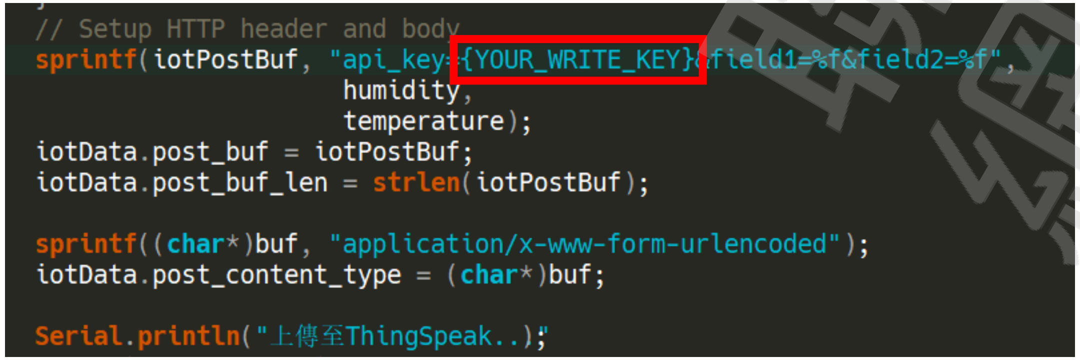
```
GET https://api.thingspeak.com/channels[redacted]feeds.json?api_key=
```

Read a Channel Field

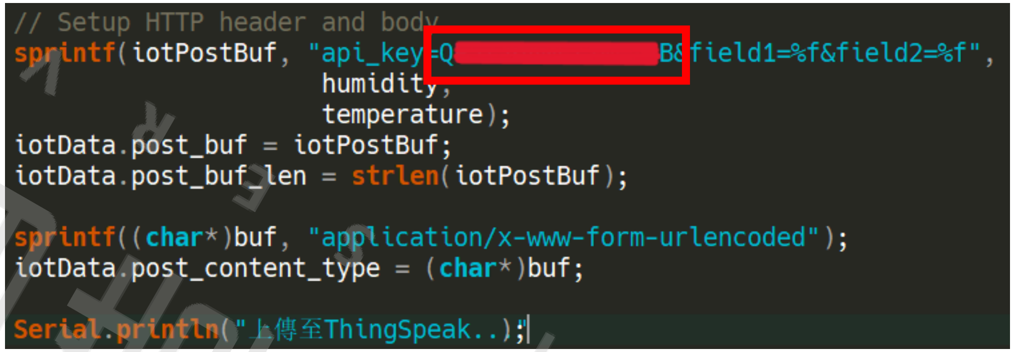
```
GET https://api.thingspeak.com/channels[redacted]fields/1.json?api_ke
```

DHT11 + ThingSpeak

5. 開啟範例程式，並替換範例程式中的Key - {YOUR_WRITE_KEY}。（請連同“{ }” 一同替換）



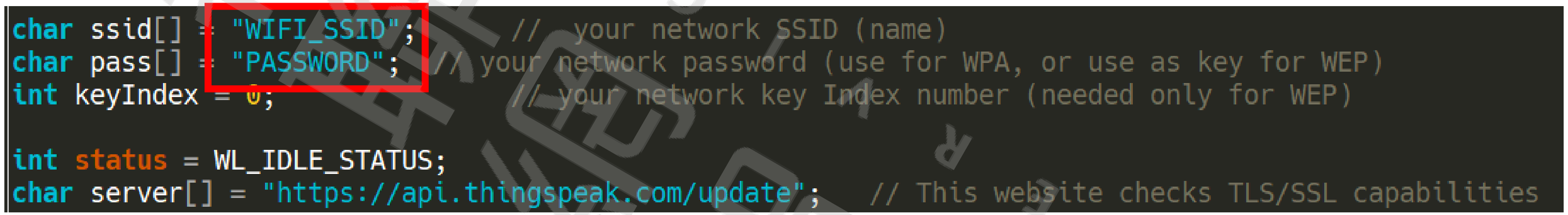
```
// Setup HTTP header and body
sprintf(iotPostBuf, "api_key={YOUR_WRITE_KEY}&field1=%f&field2=%f",
                     humidity,
                     temperature);
iotData.post_buf = iotPostBuf;
iotData.post_buf_len = strlen(iotPostBuf);

sprintf((char*)buf, "application/x-www-form-urlencoded");
iotData.post_content_type = (char*)buf;

Serial.println("上傳至ThingSpeak..);
```



```
// Setup HTTP header and body
sprintf(iotPostBuf, "api_key=[redacted]&field1=%f&field2=%f",
                     humidity,
                     temperature);
iotData.post_buf = iotPostBuf;
iotData.post_buf_len = strlen(iotPostBuf);

sprintf((char*)buf, "application/x-www-form-urlencoded");
iotData.post_content_type = (char*)buf;

Serial.println("上傳至ThingSpeak..);
```

DHT11 + ThingSpeak

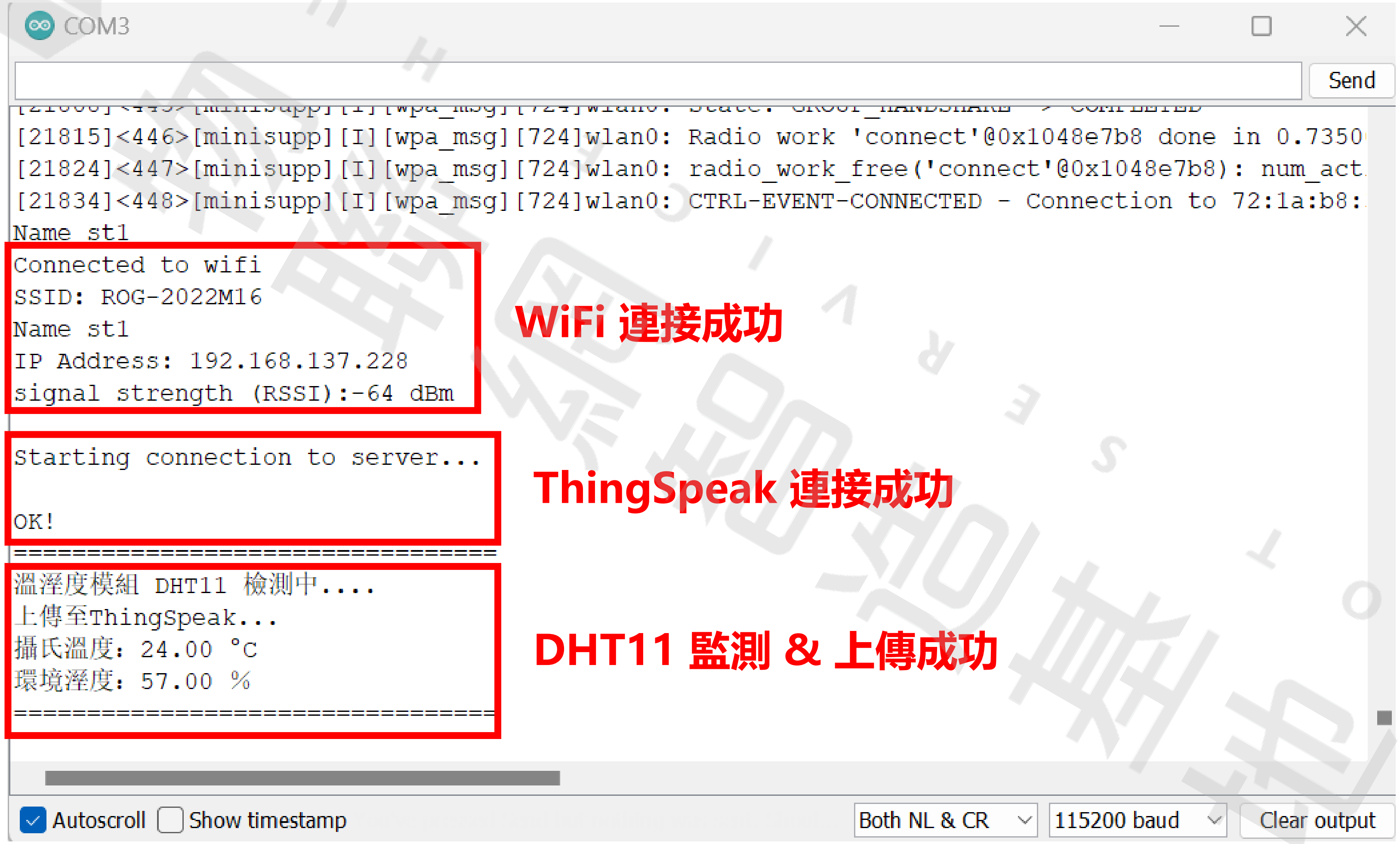
6. 把"WIFI_SSID" 及 "PASSWORD" 替換成欲連結的WIFI。



```
char ssid[] = "WIFI_SSID";        //  your network SSID (name)
char pass[] = "PASSWORD";   // your network password (use for WPA, or use as key for WEP)
int keyIndex = 0;              // your network key Index number (needed only for WEP)

int status = WL_IDLE_STATUS;
char server[] = "https://api.thingspeak.com/update";    // This website checks TLS/SSL capabilities
```

執行結果

7. 編譯並上傳，開啟監控視窗(Serial Monitor)查看執行結果。



```
[21806]<445>[minisupp][I][wpa_msg][724]wlan0: State: GROUP_HANDSHAKE -> COMPLETED
[21815]<446>[minisupp][I][wpa_msg][724]wlan0: Radio work 'connect'@0x1048e7b8 done in 0.7350
[21824]<447>[minisupp][I][wpa_msg][724]wlan0: radio_work_free('connect'@0x1048e7b8): num_act
[21834]<448>[minisupp][I][wpa_msg][724]wlan0: CTRL-EVENT-CONNECTED - Connection to 72:1a:b8:
Name st1
Connected to wifi
SSID: ROG-2022M16
Name st1
IP Address: 192.168.137.228
signal strength (RSSI):-64 dBm

Starting connection to server...

OK!
==================================
溫溼度模組 DHT11 檢測中....
上傳至ThingSpeak...
攝氏溫度: 24.00 °C
環境溼度: 57.00 %
==================================
```

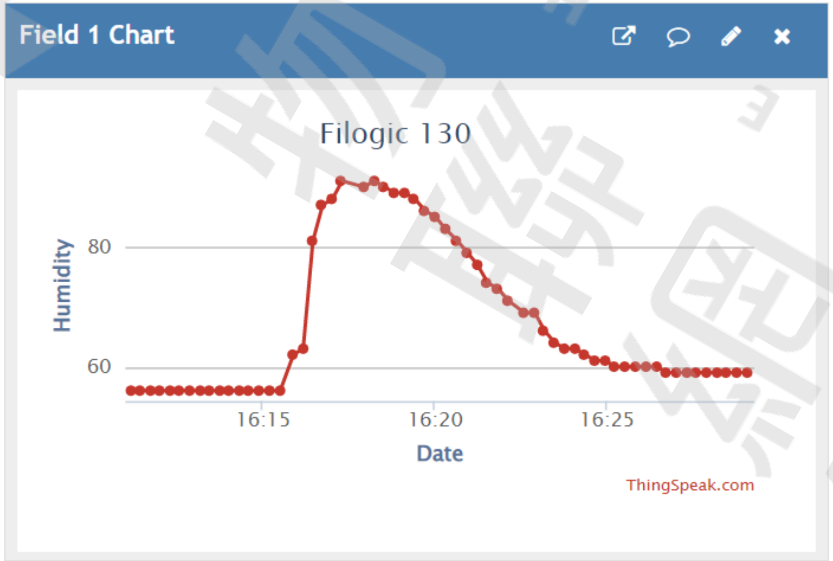
WiFi 連接成功

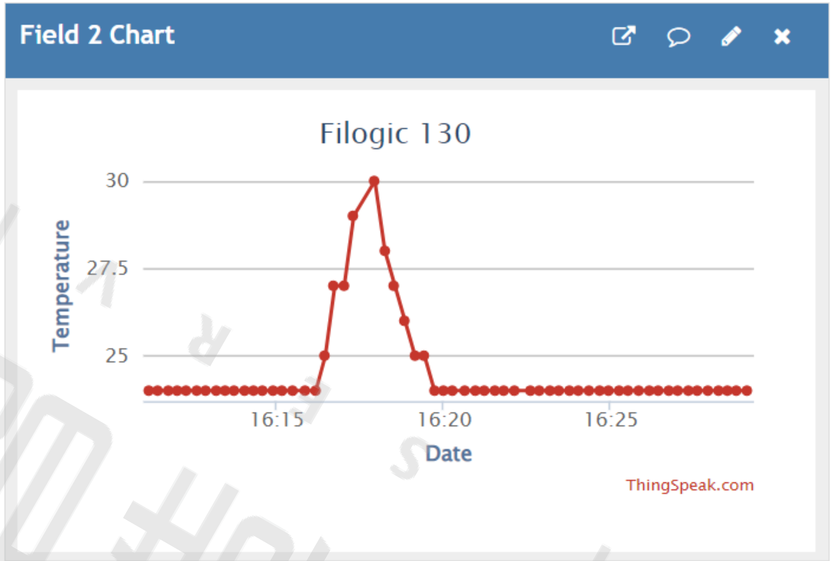
ThingSpeak 連接成功

DHT11 監測 & 上傳成功

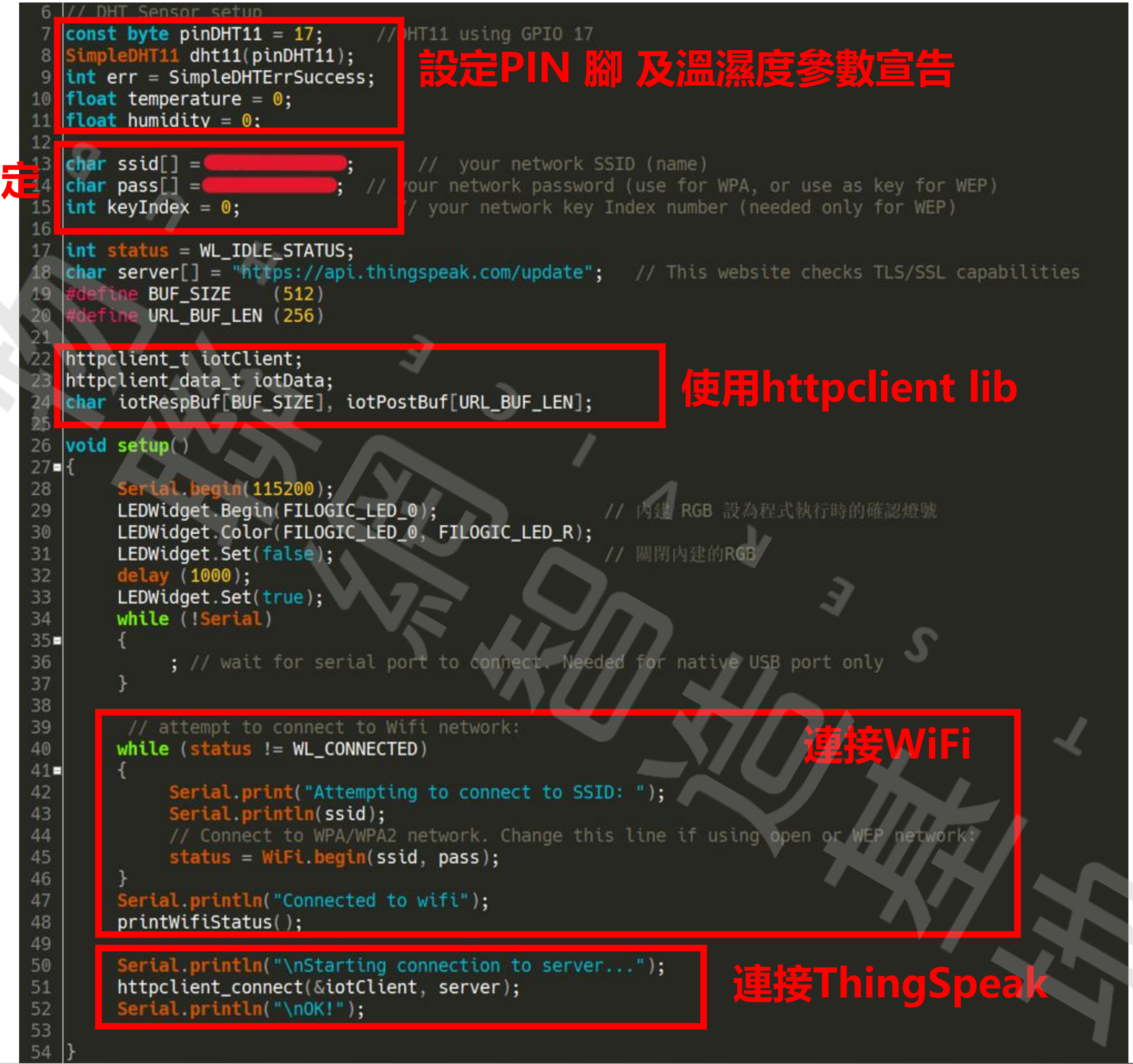
ThingSpeak 圖表展示

8. 在網站上可看到更新數據。





範例程式簡說



```
// DHT Sensor setup
const byte pinDHT11 = 17;       //DHT11 using GPIO 17
SimpleDHT11 dht11(pinDHT11);
int err = SimpleDHTErrSuccess;
float temperature = 0;
float humidity = 0;

char ssid[] = [redacted];    //  your network SSID (name)
char pass[] = [redacted];   // your network password (use for WPA, or use as key for WEP)
int keyIndex = 0;            // your network key Index number (needed only for WEP)

int status = WL_IDLE_STATUS;
char server[] = "https://api.thingspeak.com/update";   // This website checks TLS/SSL capabilities
#define BUF_SIZE     (512)
#define URL_BUF_LEN (256)

httpclient_t iotClient;
httpclient_data_t iotData;
char iotRespBuf[BUF_SIZE], iotPostBuf[URL_BUF_LEN];

void setup()
{
     Serial.begin(115200);
     LEDWidget.Begin(FILOGIC_LED_0);                   // 內建 RGB 設為程式執行時的確認燈號
     LEDWidget.Color(FILOGIC_LED_0, FILOGIC_LED_R);
     LEDWidget.Set(false);                             // 關閉內建的RGB
     delay (1000);
     LEDWidget.Set(true);
     while (!Serial)
     {
          ; // wait for serial port to connect. Needed for native USB port only
     }

      // attempt to connect to Wifi network:
     while (status != WL_CONNECTED)
     {
          Serial.print("Attempting to connect to SSID: ");
          Serial.println(ssid);
          // Connect to WPA/WPA2 network. Change this line if using open or WEP network:
          status = WiFi.begin(ssid, pass);
     }
     Serial.println("Connected to wifi");
     printWifiStatus();

     Serial.println("\nStarting connection to server...");
     httpclient_connect(&iotClient, server);
     Serial.println("\nOK!");

}
```

設定PIN 腳 及溫濕度參數宣告

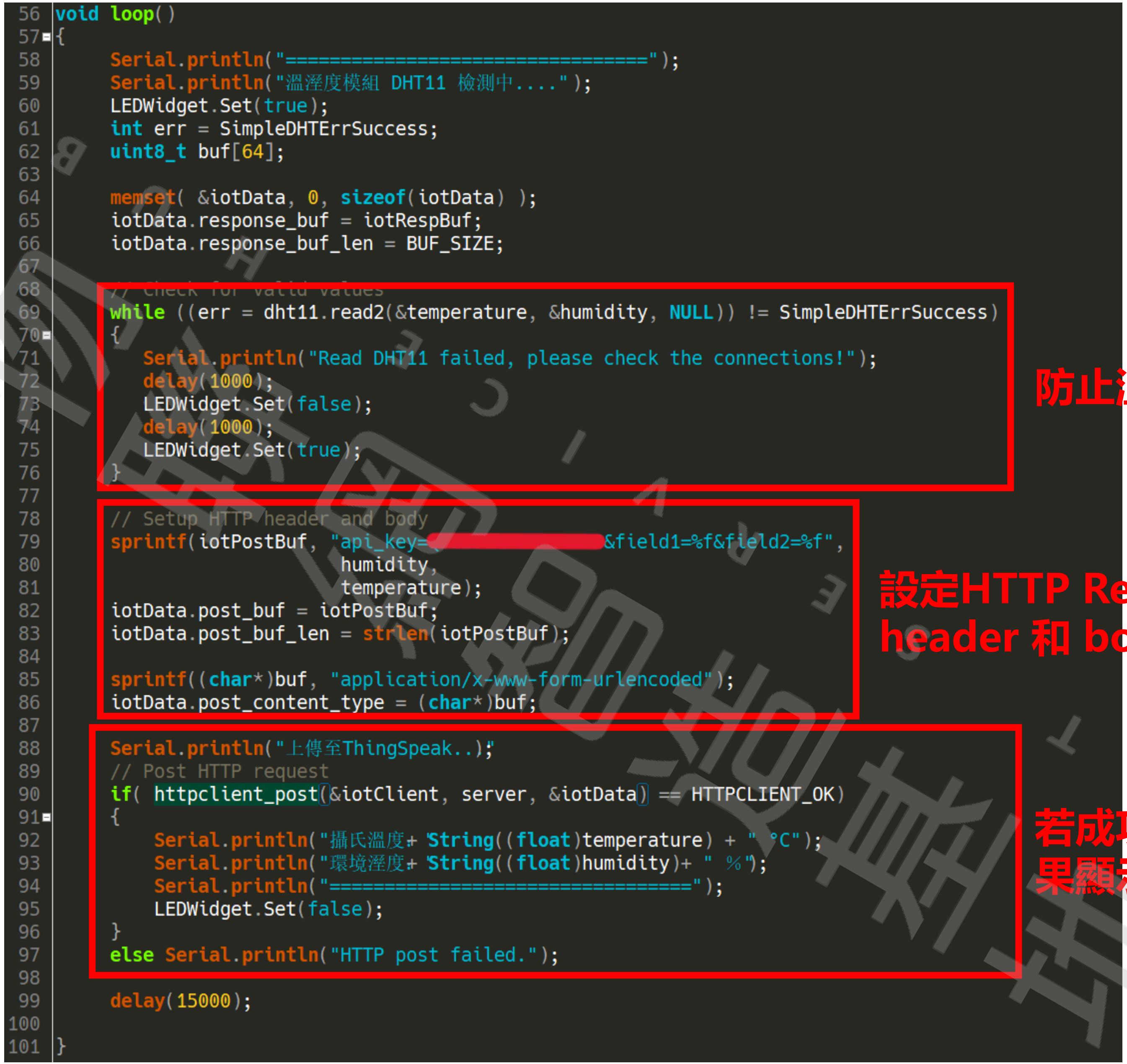
WIFI 設定

使用httpclient lib

連接WiFi

連接ThingSpeak

範例程式簡說



```
void loop()
{
    Serial.println("==================================");
    Serial.println("溫溼度模組 DHT11 檢測中....");
    LEDWidget.Set(true);
    int err = SimpleDHTErrSuccess;
    uint8_t buf[64];

    memset( &iotData, 0, sizeof(iotData) );
    iotData.response_buf = iotRespBuf;
    iotData.response_buf_len = BUF_SIZE;

    // Check for valid values
    while ((err = dht11.read2(&temperature, &humidity, NULL)) != SimpleDHTErrSuccess)
    {
      Serial.println("Read DHT11 failed, please check the connections!");
      delay(1000);
      LEDWidget.Set(false);
      delay(1000);
      LEDWidget.Set(true);
    }

    // Setup HTTP header and body
    sprintf(iotPostBuf, "api_key=[redacted]&field1=%f&field2=%f",
                        humidity,
                        temperature);
    iotData.post_buf = iotPostBuf;
    iotData.post_buf_len = strlen(iotPostBuf);

    sprintf((char*)buf, "application/x-www-form-urlencoded");
    iotData.post_content_type = (char*)buf;

    Serial.println("上傳至ThingSpeak..");
    // Post HTTP request
    if( httpclient_post(&iotClient, server, &iotData) == HTTPCLIENT_OK)
    {
        Serial.println("攝氏溫度"+ String((float)temperature) + " °C");
        Serial.println("環境溼度"+ String((float)humidity)+ " %");
        Serial.println("==================================");
        LEDWidget.Set(false);
    }
    else Serial.println("HTTP post failed.");

    delay(15000);

}
```

防止溫濕度為空

設定HTTP Request header 和 body

若成功POST，將結果顯示在監控視窗

Thank you

物聯網智造基地
IOT SERVICE HUB