

2021.6.22



DSI5168 線上工作坊



關於黑趣

黑趣活動

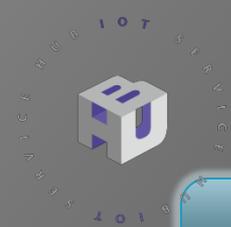
黑趣爆導

物聯網智造基地



物聯網智造基地

IoT產品邁向量產的加速站



DSI5168 線上工作坊

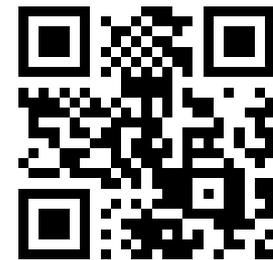
益師傅



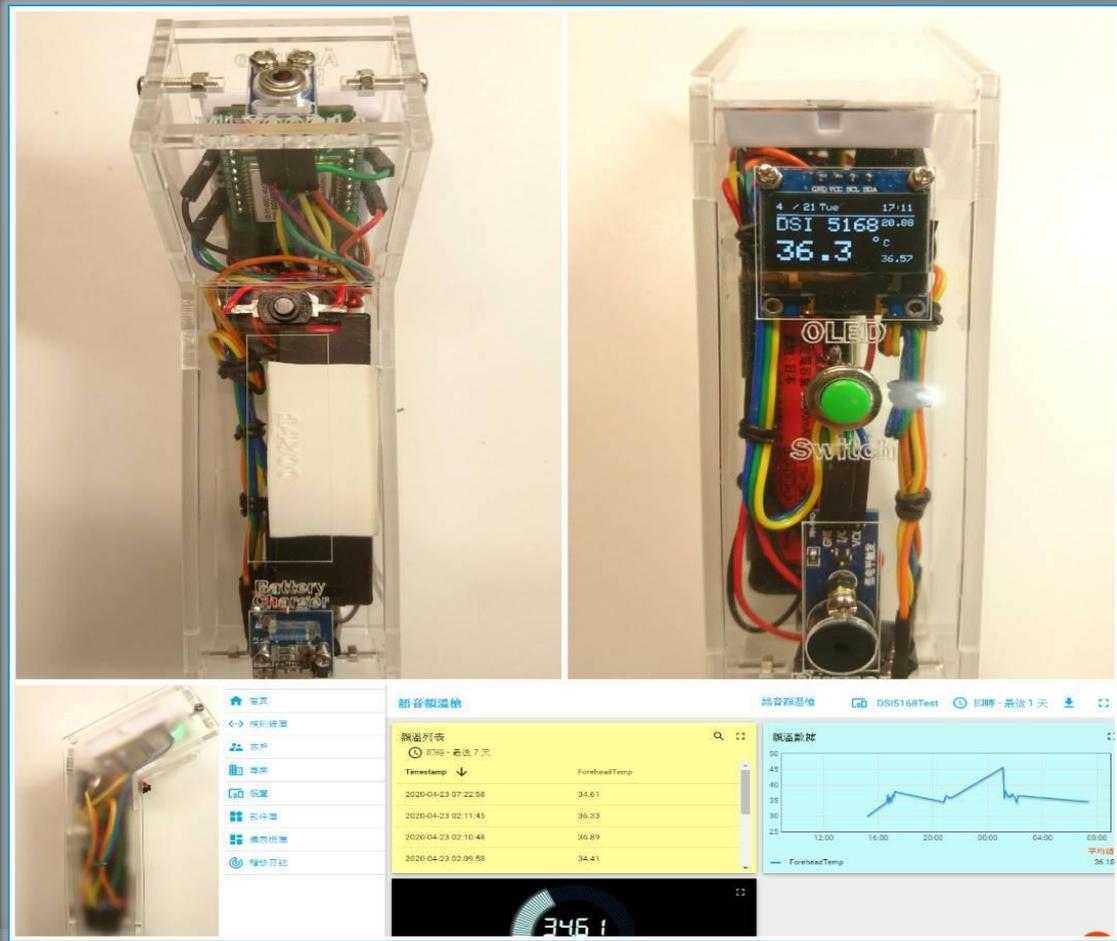
E-mai: pos_soft_king@yahoo.com.tw

程式與資料下載:

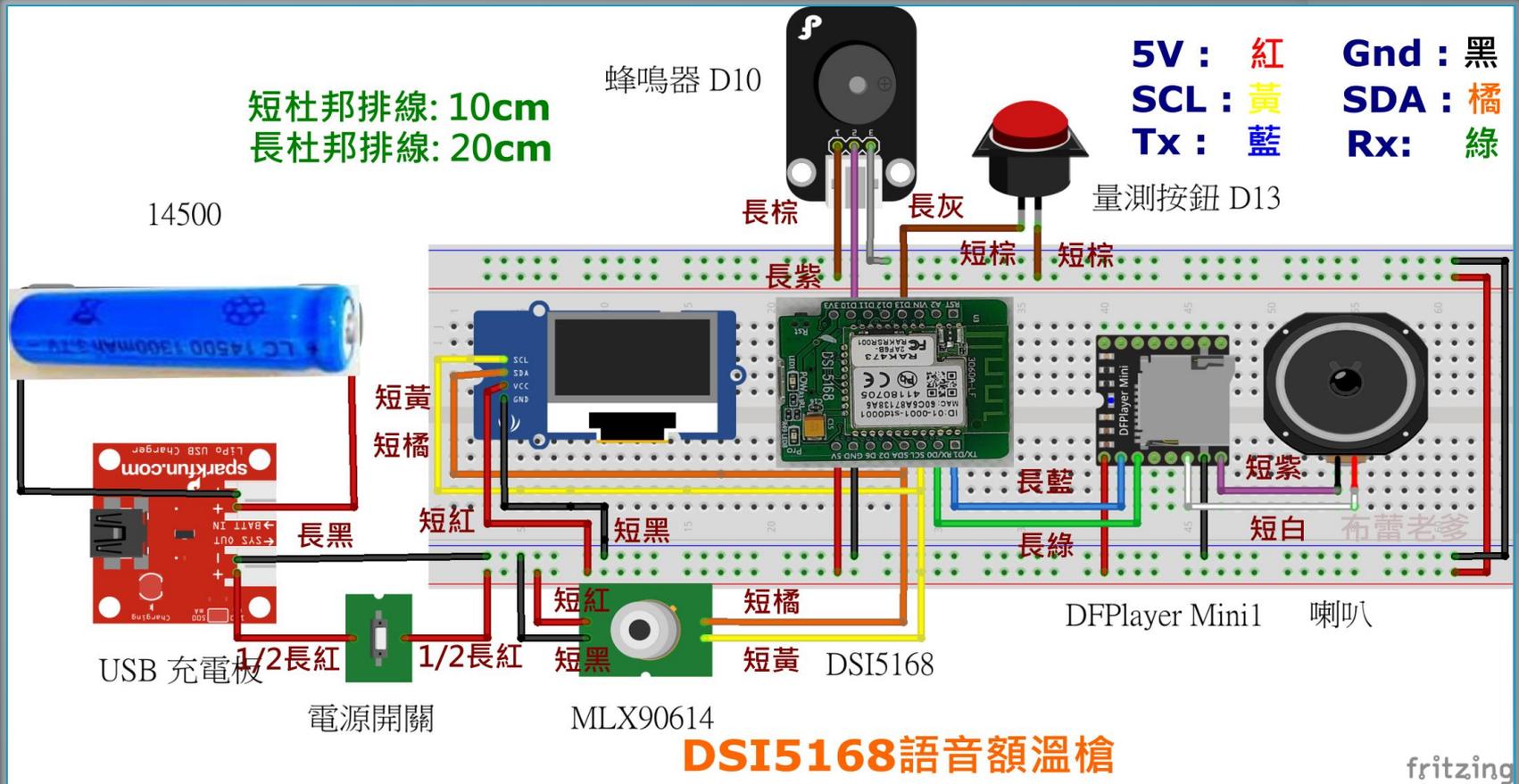
<https://reurl.cc/MA8z1W>



DSI5168 語音額溫槍



DSI5168 語音額溫槍



D1 mini 語音額溫槍

短杜邦排線 10cm
長杜邦排線 20cm

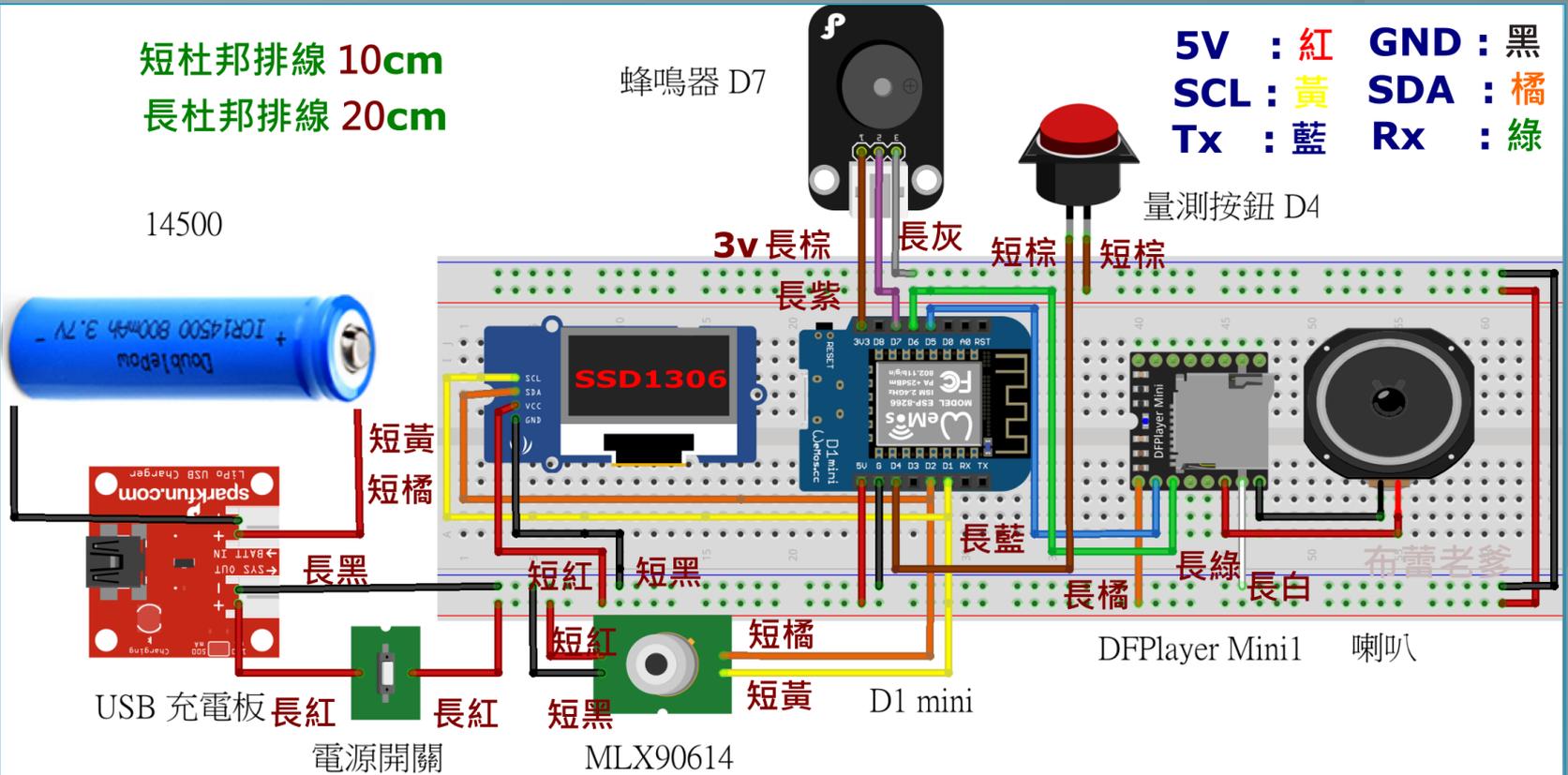
蜂鳴器 D7

5V : 紅 GND : 黑
SCL : 黃 SDA : 橘
Tx : 藍 Rx : 綠

14500

3v 長棕 長灰 短棕 短棕
長紫

量測按鈕 D4



USB 充電板

長紅

電源開關

長紅

短黑

MLX90614

短黃

D1 mini

DFPlayer Mini1

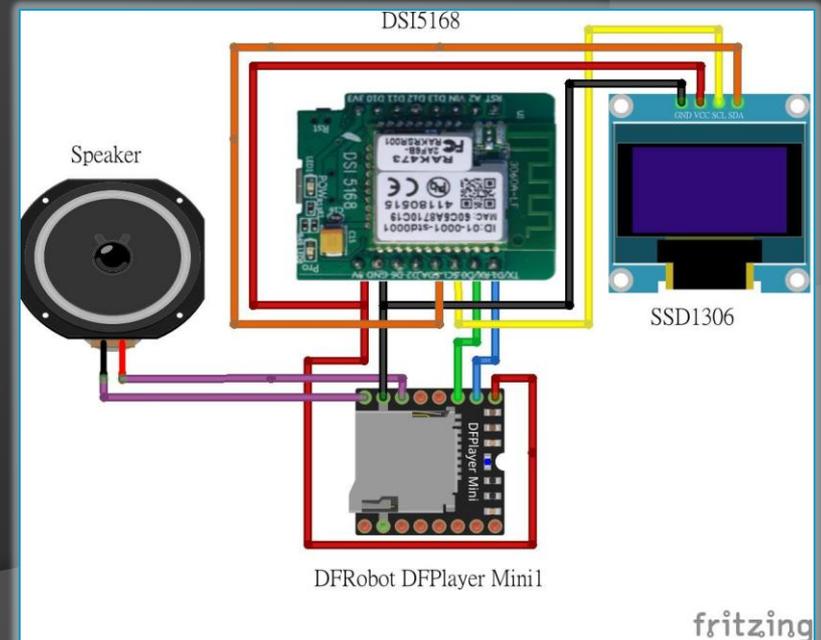
喇叭

D1 mini 語音額溫槍

fritzing



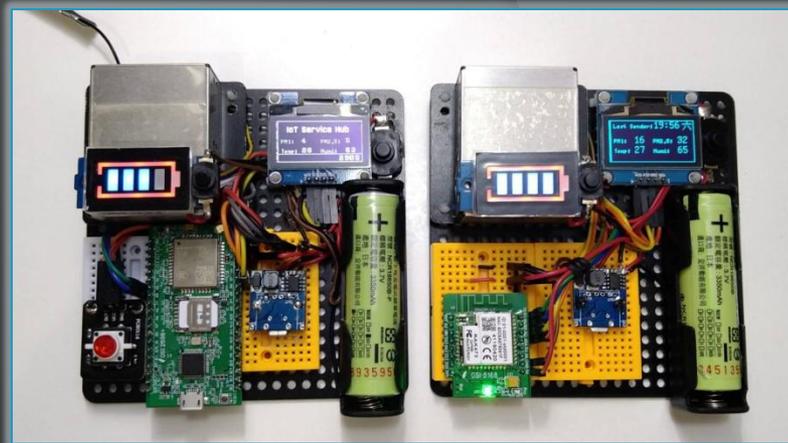
上下課鐘聲播報工具



DSI5168實作小案例



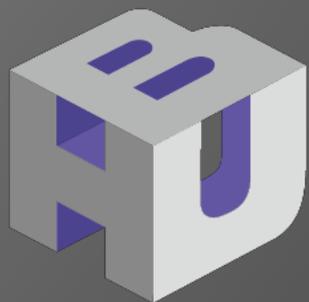
氣象預報器



MQTT 警示燈

落塵收集器





物聯網智造基地

I O T S E R V I C E H U B

國產IC開發板系列

量產導向的物聯網開發板解決方案

DSI 5168說明



MQTT解析說明

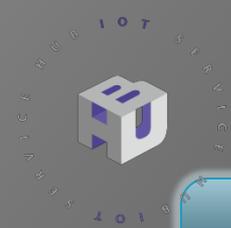
MQTT是一種基於「發布 **publish** / 訂閱 **subscribe**」機制的訊息傳輸協定，我們可以把它想成 **YouTube** 影片發行和訂閱的機制。

MQTT 訊息發送端，相當於 **YouTuber** 拍攝影片上傳到 **YouTube** 後，若有開啟訂閱小鈴鐺時，便會收到發片通知可以讓訂閱者觀看，而 **YouTube** 角色就如同代理人（**broker**），來統籌管理發行和訂閱事宜。
每一個訊息來源（影片）都有個唯一的主題名稱（哪個 **YouTuber** 的哪個影片）。

代理人是個伺服器軟體(我們這裡都採用 **broker.emqx.io**)，向伺服器發送主題的一方是發布者（**publisher**），從伺服器獲取主題的一方則是訂閱者（**subscriber**）。

傳送感測器資料的一端是所謂的發佈者(**publish**)，接收感測器資料的另一端則是訂閱者(**subscribe**)。每個 **Sensor/MCU** 的訊息都需要有個主題（**Topic**）名稱以方便識別。

MQTT的主題（**Topic**）：**MQTT**主題名稱可以是 **UTF-8**（萬國碼）編碼的字串，我們可以自行決定主題名稱，舉例來說：傳送溫度的訊息主題可命名成{**Temp**}、傳送濕度的訊息主題叫做{**Humi**}、傳送亮度的訊息主題叫做{**Lux**}...等等。
主題名稱也支援類似檔案路徑的階層式命名方式，例如：**wanhua/room1/temp**



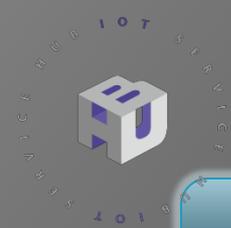
MQTT 整合運用

安裝 MQTTlens

chrome 線上應用程式商店 -> MQTTLens

參考資料:

<https://swf.com.tw/?p=1009>



MQTT Lens

MQTT LENS

Connections +

- broker.emqx.io
- Local
- IDEASChain
- Mobile

Add a new Connection

Connection Details

broker.emqx.io

connection name: broker.emqx.io

Connection color scheme:

Hostname: tcp:// broker.emqx.io Port: 1883

Client ID: lens_go6LEBCGdkwnKPMxdrG1DIVm6pJ Generate a random ID

Session: Clean Session Automatic Connection: Automatic Connection Keep Alive: 120 seconds

Credentials

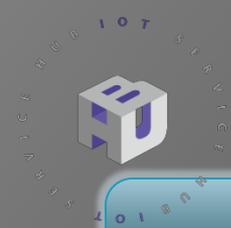
Username: emqx Password:

version 0.0.14

most once SUBSCRIBE

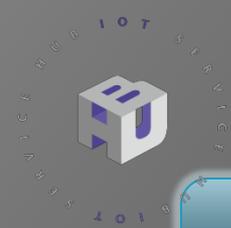
Retained PUBLISH

Messages: 0/2



MQTT Lens

The screenshot displays the MQTT Lens web interface. At the top left, the logo and name 'MQTT Lens' are visible, and at the top right, the version 'Version U.U.14' is shown. On the left side, there is a 'Connections' panel with two entries: 'broker.emqx.io' (highlighted with a yellow box) and 'IDEASChain'. The main area shows the configuration for the 'broker.emqx.io' connection. Under the 'Subscribe' section, the topic 'wanhua/room1/temp' is entered in a text field (highlighted with a red box). To the right of this field is a dropdown menu set to '0 - at most once' and a green 'SUBSCRIBE' button (highlighted with a red box). Below this, the 'Publish' section also has the topic 'wanhua/room1/temp' in its text field (highlighted with a blue box), a dropdown menu set to '0 - at most once', a 'Retained' checkbox, and a green 'PUBLISH' button (highlighted with a blue box). Two blue arrows point from the text 'wanhua/room1/temp' to the respective text fields. Below the publish section, a 'Message' section shows the value '26.5'. Further down, a 'Subscriptions' section shows the topic 'wanhua/room1/temp' with a message count of '0/1'. A table below this shows a single entry: '# Time Topic QoS' with '0 12:31:02 wanhua/room1/temp 0'. A 'Message: 26.5' is also listed. At the bottom, the 'IDEASChain' connection is partially visible. A large blue arrow on the left points downwards, indicating the flow of the interface.



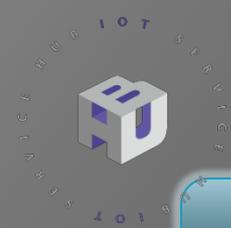
JSON整合運用

JSON 解析與練習

Node-RED 的核心就是 **Node.js** , 而他實現 **IoT** 就是用 **MQTT**, 各位若之前沒接觸過 **Node-RED** 而先接觸 **MQTT** 的人, 就會發現 其實 **Node-RED** 裡頭就是 **JSON** 結構, 用的都是 **payload** 字眼.

所以建議各位 一定先清楚 **JSON** 結構再來使用他, 會覺得很輕鬆上手 .

各位可以多試試下面的連結, 來驗證自己是否了解



JSON整合運用

JSON 解析與練習

參考資料:

<http://JSON 教學>

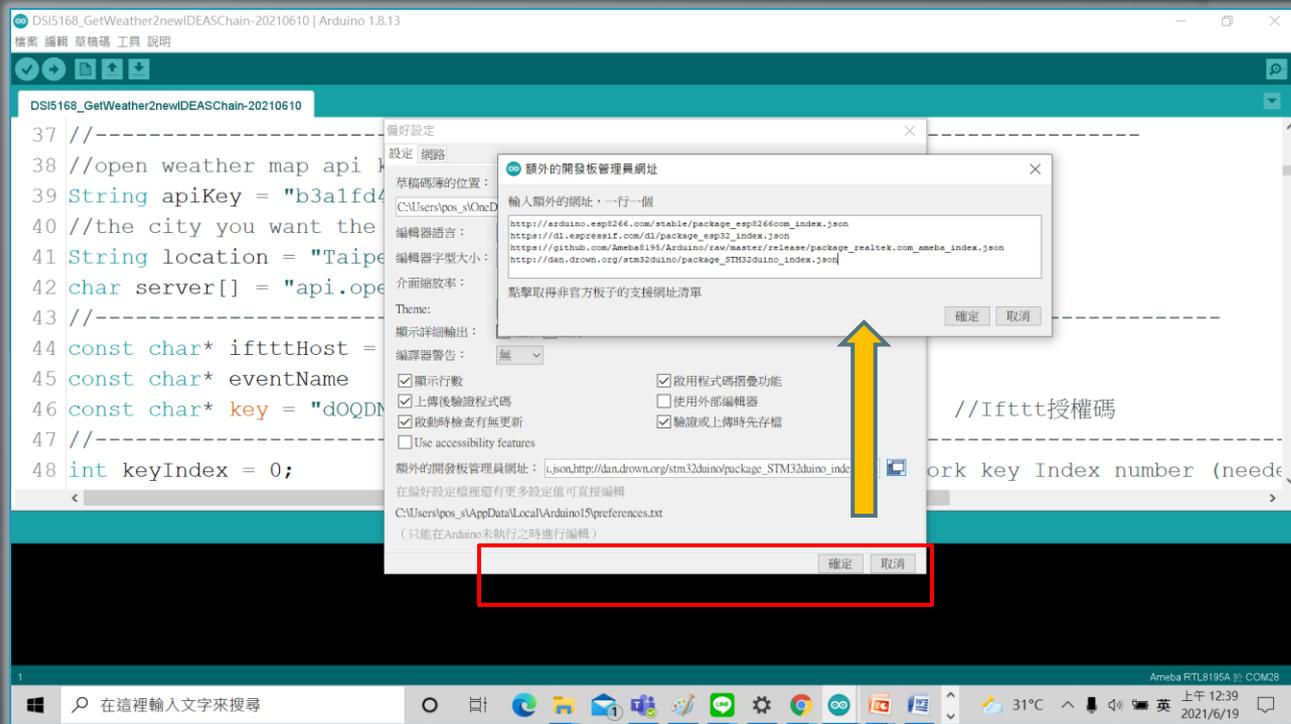
工具

<http://jsonviewer.stack.hu/>

練習

<https://reurl.cc/Dvq0zR>

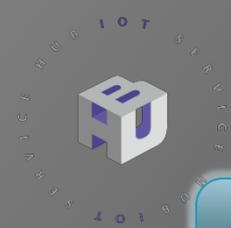
arduino 環境設定



參考步驟: <https://reurl.cc/9r4a28>

填入:

https://github.com/Ameba8195/Arduino/raw/master/release/package_realtek.com_ameba_index.json



程式解說

抓取OpenWeatherMap 天氣資料

JSON Serialize 與 Deserialize

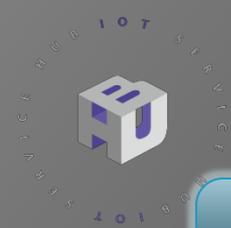
IDEASChain MQTT 傳送

NTP

程式解說

IDEASChain MQTT Topic

```
14 char ssid1[] = "*****"; // your network SSID (name)
15 char pass1[] = "*****"; // your network password
16 char mqtt_server[] = "broker.emqx.io";
17 const unsigned int mqtt_port = 1883;
18 int status = WL_IDLE_STATUS; // the Wifi radio's status
19 //-----
20 char mqttServer[] = "iiot.ideaschain.com.tw"; // new ideaschain dashboard MQTT server
21 int mqttPort = 1883;
22 char clientId[] = "DSI5168Test"; // MQTT client ID. it's better to use unique id.
23 char username[] = "6ee85EfXy5Ud7otVtULk"; // device access token(存取權杖)
24 char password[] = ""; // no need password
25 //char subscribeTopic[] = "v1/devices/me/attributes"; // Fixed topic. ***DO NOT MODIFY***
26 char subscribeTopic[] = "v1/devices/me/telemetry";
27 char publishTopic[] = "v1/devices/me/telemetry"; // Fixed topic. ***DO NOT MODIFY***
28 char subscribeTopicRPC[] = "v1/devices/me/rpc/request/+"; // RPC MQTT
29 char publishPayload[] = "{\"ForeheadTemp\": \"30\"}"; // String of stringified JSON Object (key value pairs)
30 char publishTopicStr1[] = "ForeheadTemp";
31 char publishTopicStr2[] = "NowTemp";
32 char publishTopicStr3[] = "MaxTemp";
33 char publishTopicStr4[] = "MinTemp";
34 char publishTopicStr5[] = "Humi";
```



程式解說

Deserialize

```
354 while (wifiClient.available())
355 {
356     String jsonData = wifiClient.readStringUntil('\r'); //逐列讀取
357     Serial.println(jsonData);
358     DeserializationError error = deserializeJson(doc, jsonData);
359
360     // Test if parsing succeeds.
361     if (error) {
362         Serial.print(F("deserializeJson() failed: "));
363         Serial.println(error.c_str());
364         //return;
365     }
```

Serialize

```
237     serializeJson(json_doc, json_output);
238     Serial.print("JSON Str: ");
239     Serial.println(json_output);
240     delay(1000);
241     mqttClient.publish(publishTopic, json_output);
242 }
243 mqttClient.loop();
244 delay(50);
245 }
```

MQTT程式注意事項

MQTT 程式解說

原本的寫法

```
void loop() {  
  if (!client.connected()) {  
    reconnect();  
  }  
  client.loop();  
  
  // 等待20秒  
  if (millis() - prevMillis > interval) {  
    prevMillis = millis();  
  
    // 讀取DHT11的溫濕度資料  
    int chk = DHT11.read(DHT11PIN);  
    if (chk == 0) {  
      temp = DHT11.temperature;  
      hum = DHT11.humidity;  
    }  
  
    // 組合MQTT訊息; field1填入溫度、field2填入濕度  
    msgStr=msgStr+"field1="+temp+"&field2="+hum;  
  
    // 宣告字元陣列  
    byte arrSize = msgStr.length() + 1;  
    char msg[arrSize];  
  
    Serial.print("Publish message: ");  
    Serial.println(msgStr);  
    msgStr.toCharArray(msg, arrSize); // 把String字串轉換成字元陣列格式  
    client.publish(topic, msg); // 發布MQTT主題與訊息  
    msgStr = "";  
  }  
}
```



較佳的寫法

```
void loop() {  
  
  int getLoop =0;  
  mqtt_nonblock_reconnect();  
  buttonA.loop(); //重複按鍵的觸發設定  
  buttonB.loop();  
  
  if((minute() == 30 ) && ( second() == 0))  
  {  
  
    getWeather();  
    getNtpTime();  
    tft.fillScreen(TFT_BLACK); // 用全黑清除螢幕  
  }  
  
  .....  
  .....  
  .....  
  .....  
  .....  
  
  mqttClient.loop();  
  delay(MQTT_LOOP_INTERVAL);  
}
```



MQTT Dash 說明

MQTT Dash 僅適用於 **Android** 設備，可用於 **智能手機** 和 **平板電腦**。它包含一個儀表板，可以在其中註冊多個專案，例如針對不同的房間。使用 **Icon**，**文字**，**圖片** 和類似內容，可以訂閱或發佈不同的主題。它支持 **Raspberry Pi**，並且可以 **24/7 全天候工作**。

安裝和配置 **MQTT Dash App**

於 **Google Play** 商店搜索 “ **MQTT Dash** ”



”並且安裝

安裝後打開 “ MQTT Dash 應用程序 ”
點擊右上角的 “ **加入** ” 符號。

Name: 輸入名稱 ,這是可自訂。

Address: 輸入位址 `broker.emqx.io`

Port: 端口可以保留 `1883` 值。

MQTT Dash

have just one connection configured.

If you have more than one connection, you can create home screen shortcut for every connection.
To create shortcut long press on any connection in connections list.

Keep screen on when connected to this broker

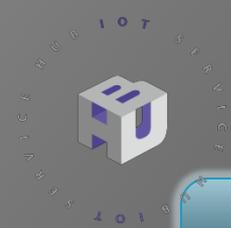
Allow metrics management. If disabled, you can't add, edit, delete or rearrange metrics.
This serves as protection from unintentional metrics changing.

Name

Address

Port

1883



MQTT Dash 說明

MQTT Dash

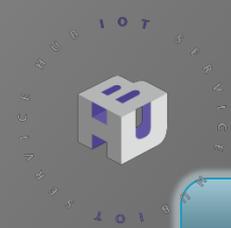


JavaScript 語法運用

ON RECEIVE
ON DISPLAY
ON TAP

參考資料:

<https://reurl.cc/8yDDmM>



MQTT Dash 練習

練習一：

從 **DSI5168** (60秒傳送一次) 抓取台北天氣預報

最高溫度 (Text)	Topic :	v1/devices/me/telemetry
	JSON:	MaxTemp.value
最低溫度 (Text)	Topic :	v1/devices/me/telemetry
	JSON:	MinTemp.value
目前溫度 (Text)	Topic :	v1/devices/me/telemetry
	JSON:	NowTemp.value
濕度 (Range)	Topic :	v1/devices/me/telemetry
	JSON:	Humi.value

