



DSI5168 天氣預報器



關於黑趣

黑趣活動

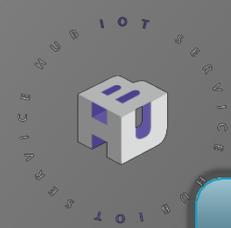
黑趣爆導

物聯網智造基地



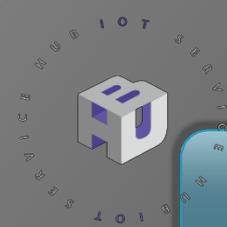
物聯網智造基地

IoT產品邁向量產的加速站



DSI5168 天氣預報器

1. 抓取OpenWeatherMap 天氣資料
2. 透過按鈕 播報天氣預報並傳送資料到 LINE
3. 定時透過 MQTT傳送資料
4. 透過 IDEASChain dashboard MQTT
RPC 做雙向控制



系統功能

IDEASChain MQTT RPC 運用

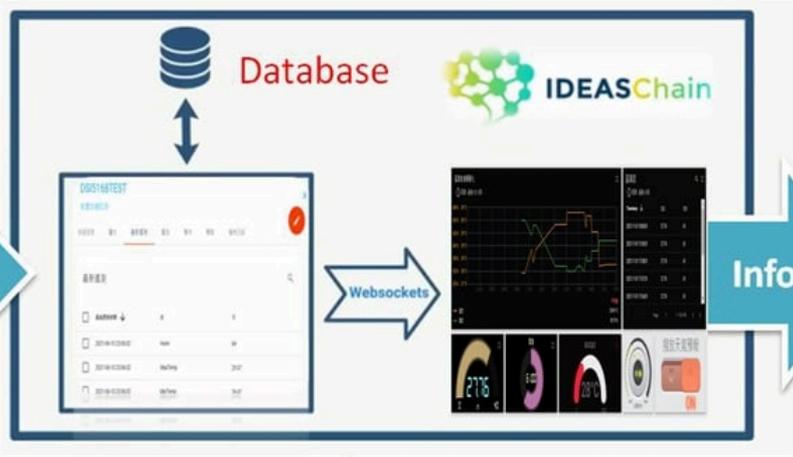
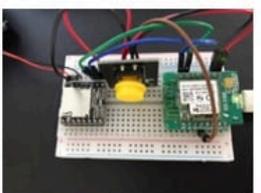
用簡單的天氣語音播報器來與 dashboard 做雙向互控機制.

1. 用 **DSI5168** 抓取 **OpenWeatherMap** 天氣預報, 透過 **MQTT** 將感測值轉程 **JSON** 格式丟出, 由數據平台接收並呈現 **dashboard**.
2. 透過播放器上的按鈕, 播放目前三小時內的天氣預報, 並透過 **LINE** 傳送預報資訊.
3. 可以由 **dashboard** 上的旋鈕與播放鈕 以 **RPC** 方式來控制 播報器上的音量大小與啟動播放預報.

MQTT RPC: 透過一組階層式 **Topic**, 每個 **control widgets** 有自己的 **getvalue** 與 **setvalue**, 透過不同的 **method** 取出便能知道哪個 **widgets** 被觸發到. 進而解析執行對應的 **function**. 這樣的作法就無須每個 **widgets** 都要有獨立的 **MQTT Topic** 了, 管理上也較便利.

系統架構

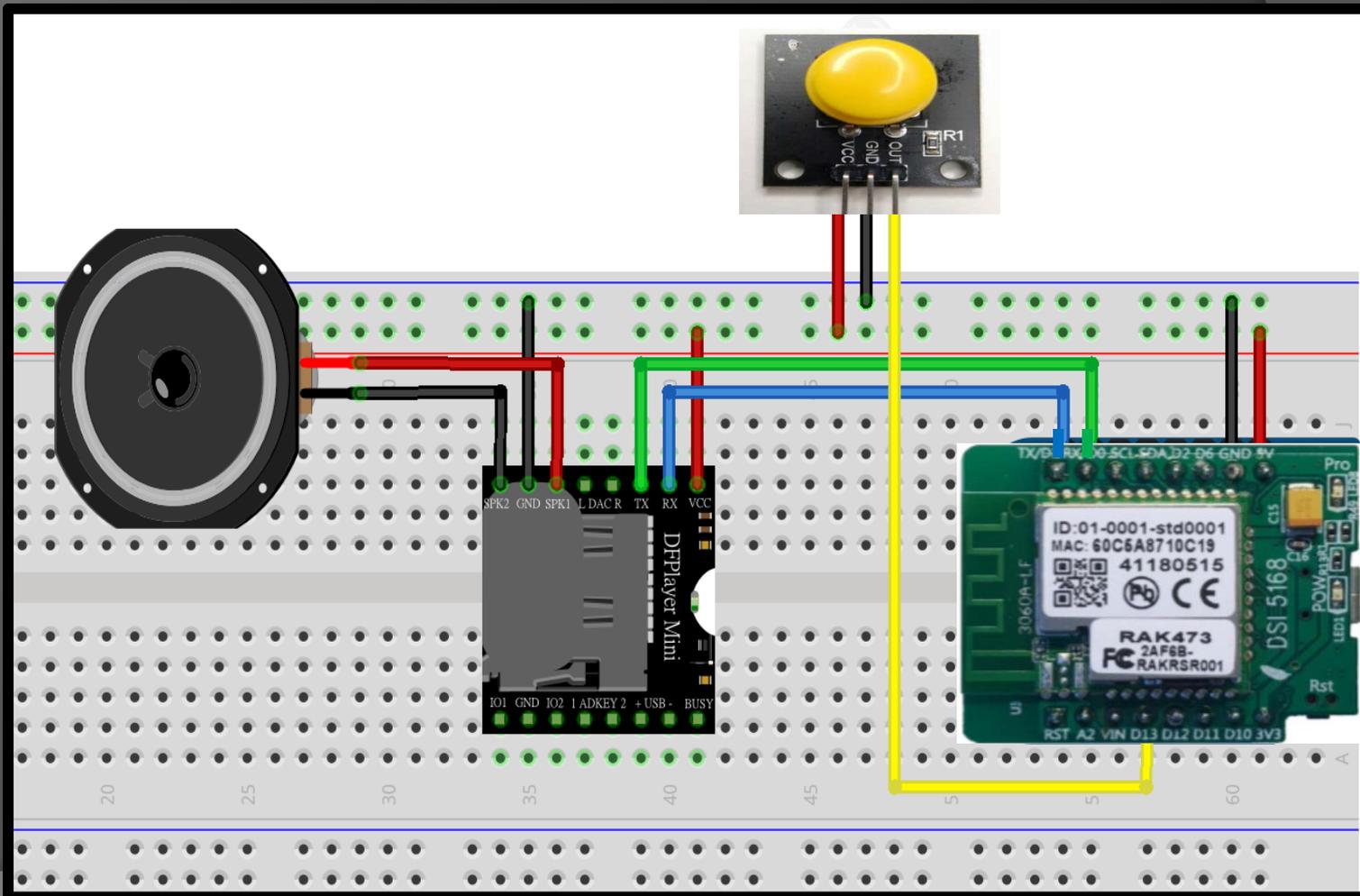
OpenWeatherMap



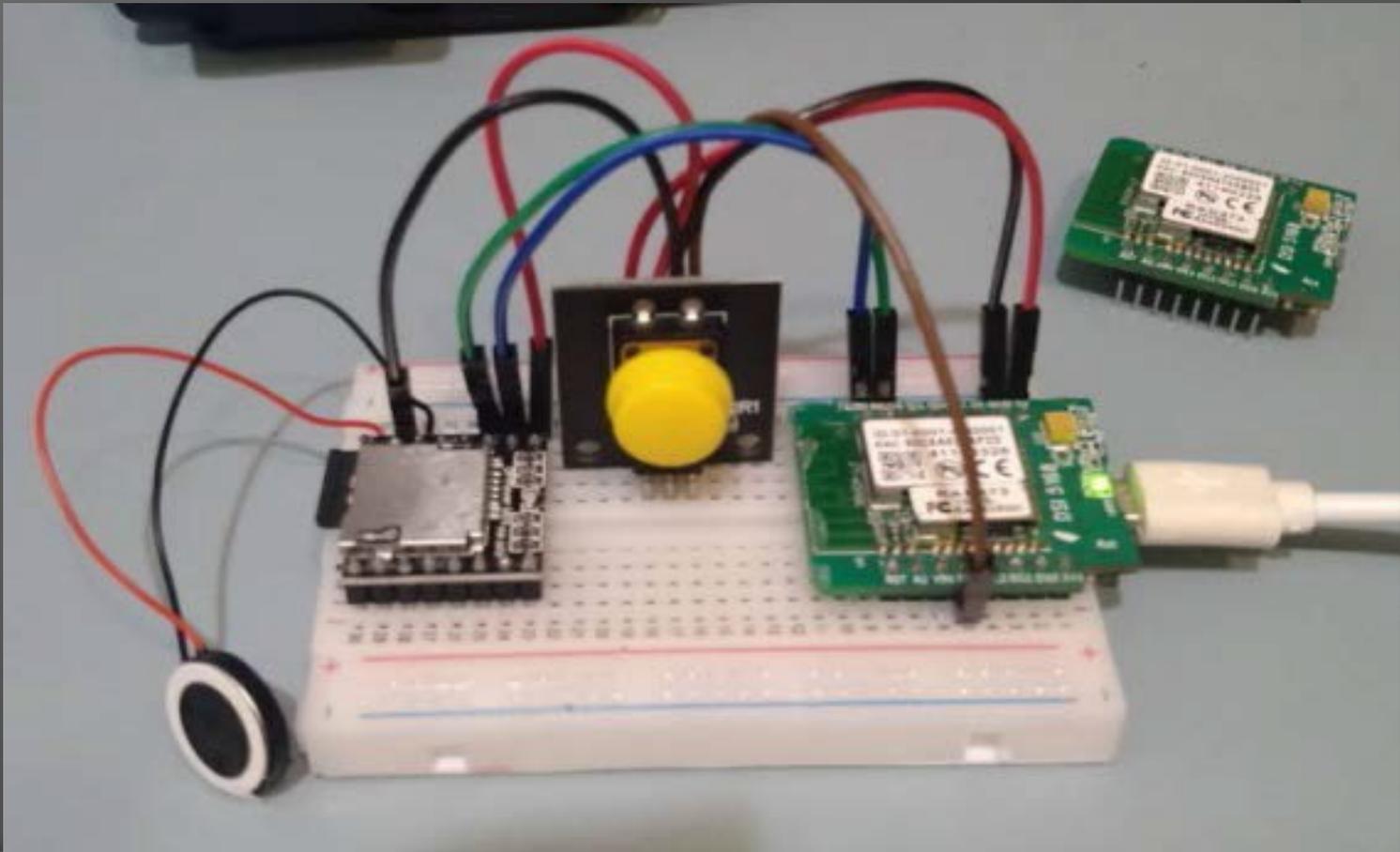
user



硬體接線



硬體完成品





MQTT解析說明

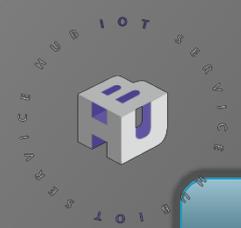
MQTT是一種基於「發布 **publish** / 訂閱 **subscribe**」機制的訊息傳輸協定，我們可以把它想成 **YouTube** 影片發行和訂閱的機制。

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)。

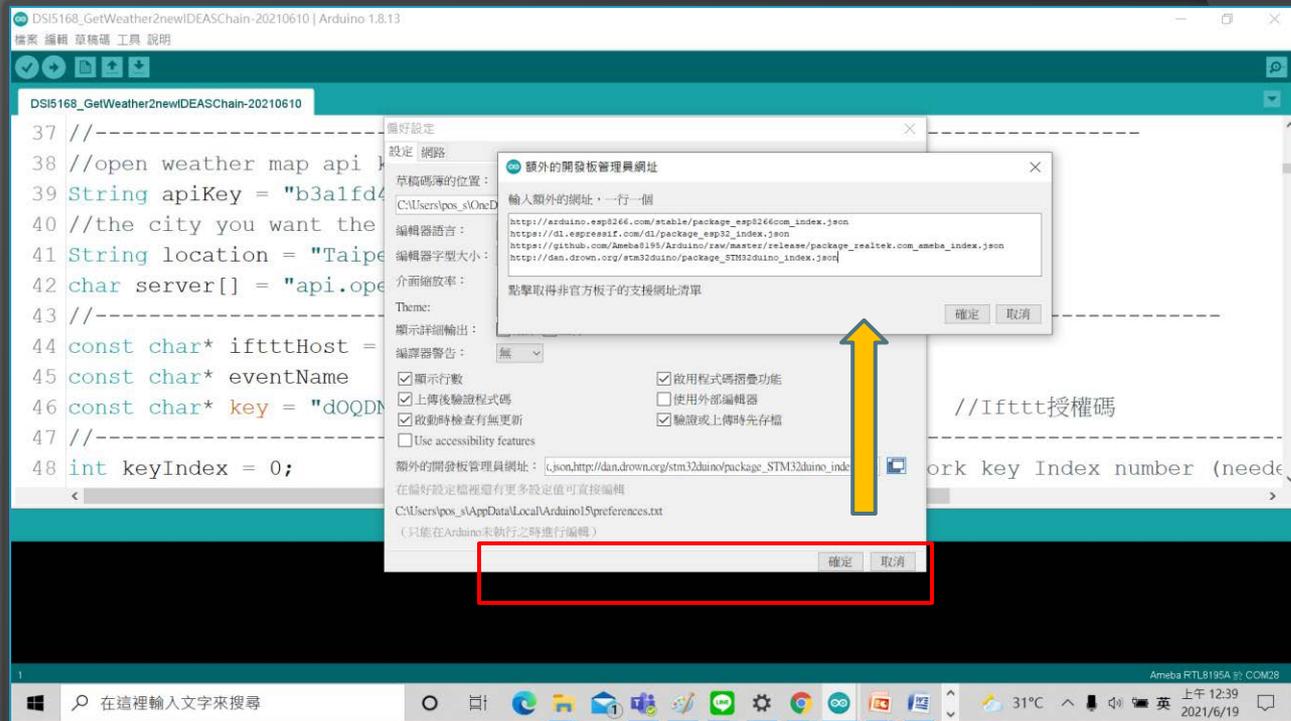
代理人是個伺服器軟體(我們這裡都採用 **broker.emqx.io**)，向伺服器發送主題的一方是發布者 (**publisher**)，從伺服器獲取主題的一方則是訂閱者 (**subscriber**)。

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**



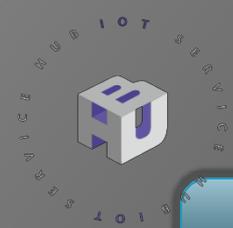
DSI5168 arduino 環境設定



參考步驟: <https://reurl.cc/9r4a28>

填入:

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



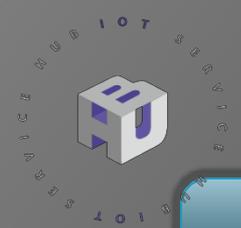
MQTT Client 測試

安裝 MQTTlens

[chrome 線上應用程式商店 -> MQTTLens](#)

參考資料:

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



MQTTLens

MQTTLens

Connections +

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

Add a new Connection

Connection Details

broker.emqx.io

Connection name: broker.emqx.io

Connection color scheme: [Blue bar]

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

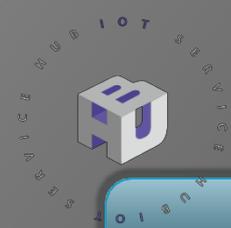
most once

SUBSCRIBE

Retained

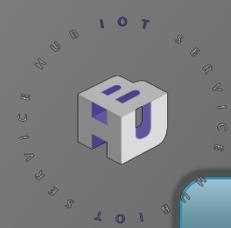
PUBLISH

Messages: 0/2



MQTTLens

The screenshot displays the MQTTLens web interface. At the top left, the logo and name 'MQTTLens' are visible, and at the top right, the version 'Version 0.0.14' 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), and a 'SUBSCRIBE' button is visible (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) and a 'PUBLISH' button (highlighted with a blue box). Two blue arrows point from the text fields to the topic name 'wanhua/room1/temp' written in red above them. A large blue arrow on the left points downwards towards the 'Subscriptions' section. In the 'Subscriptions' section, the topic 'wanhua/room1/temp' is listed with a message count of '0/1'. Below this, a table shows a message received at '12:31:02' with the topic 'wanhua/room1/temp' and a QoS of '0'. The message content is '26.5'. At the bottom, the 'IDEASChain' connection is partially visible.



JSON整合運用

JSON 解析與練習

參考資料:

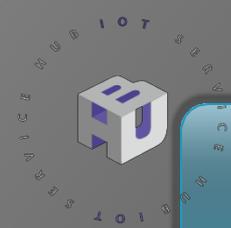
[http://JSON 教學](http://JSON教學)

工具

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

練習

<https://reurl.cc/Dvq0zR>



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** ”



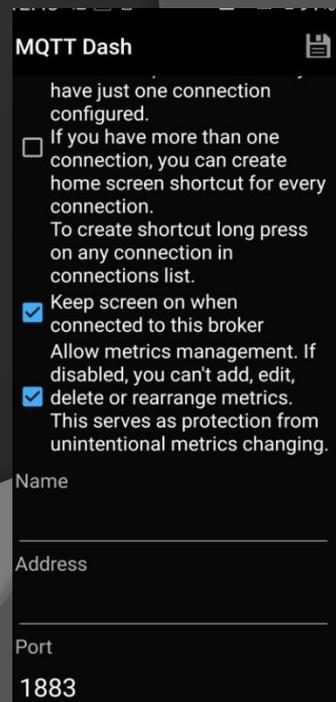
”並且安裝

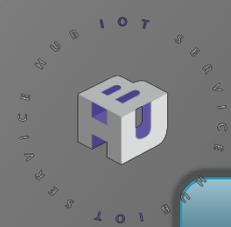
安裝後打開 “ **MQTT Dash** 應用程序 ”
點擊右上角的 “ **加入** ” 符號。

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 設定

從 **DSI5168** 抓取天氣預報

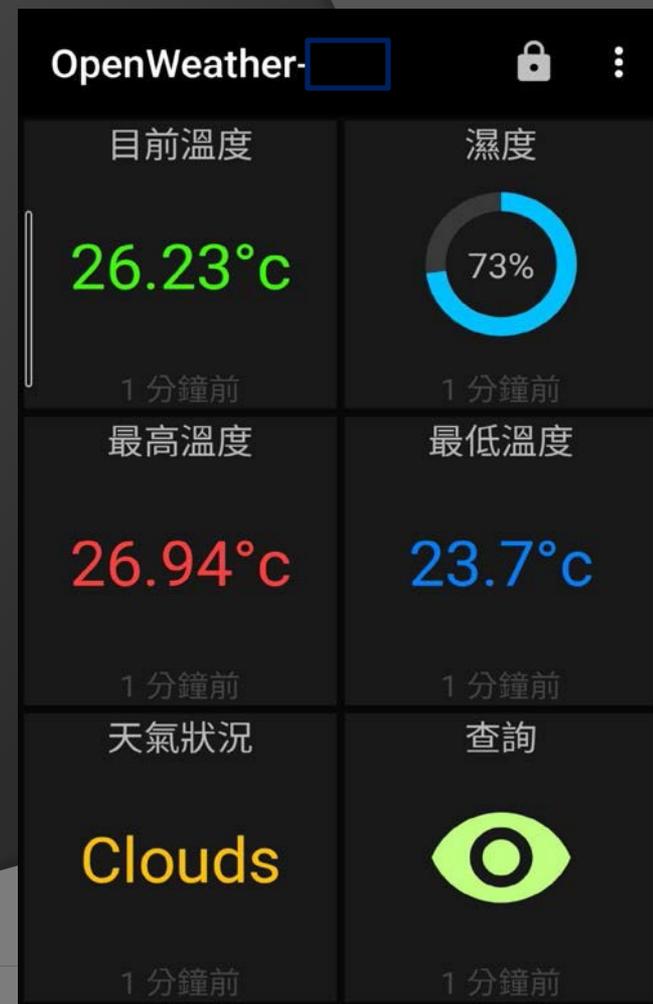
Topic :
v1/devices/me/telemetry

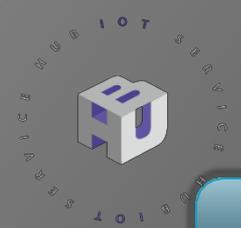
最高溫度 (Text) JSON: **MaxTemp.value**

最低溫度 (Text) JSON: **MinTemp.value**

目前溫度 (Text) JSON: **NowTemp.value**

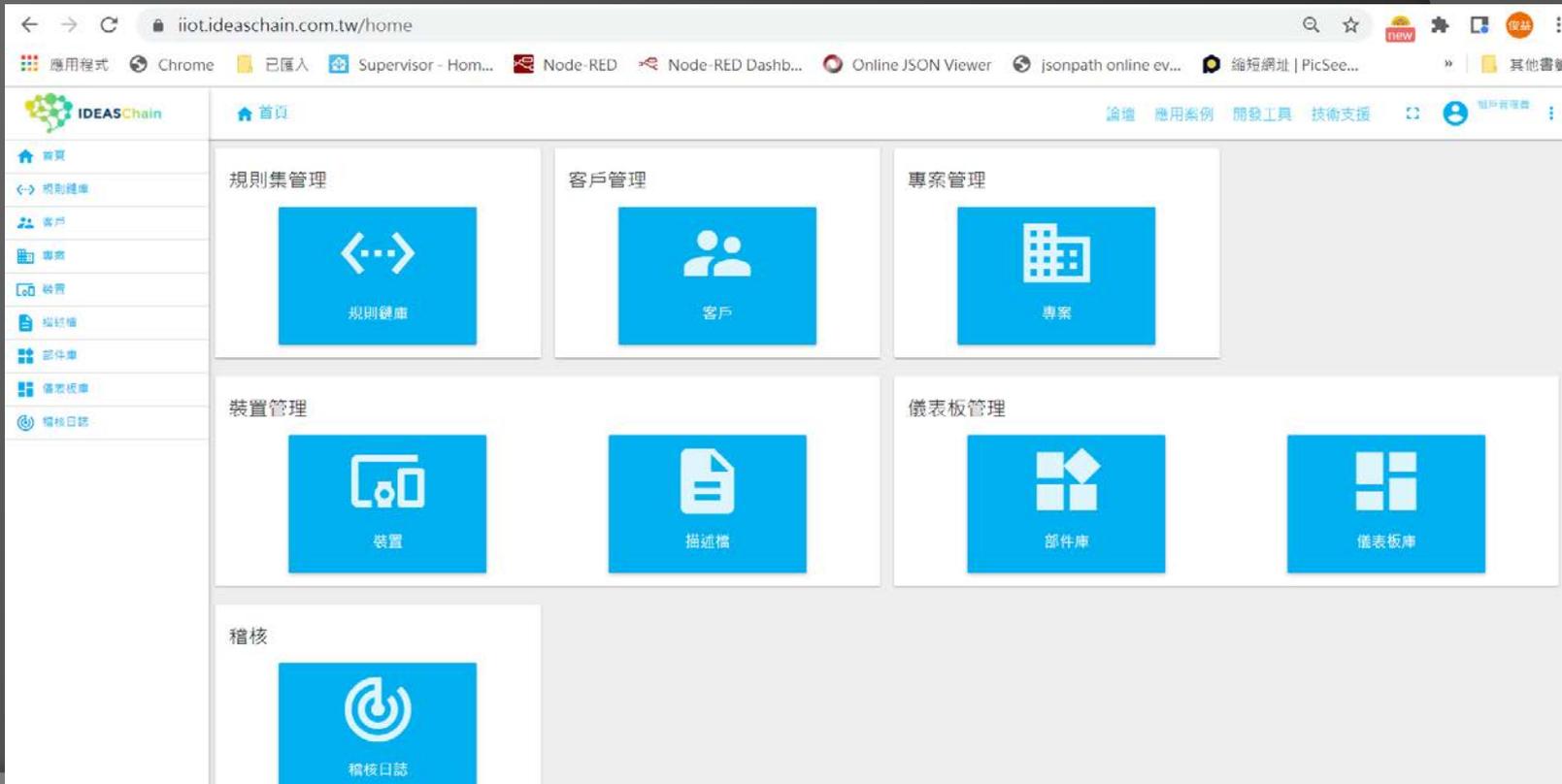
濕度 (Range) JSON: **HumiTemp.value**

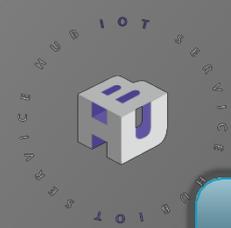




IDEASChain 設定

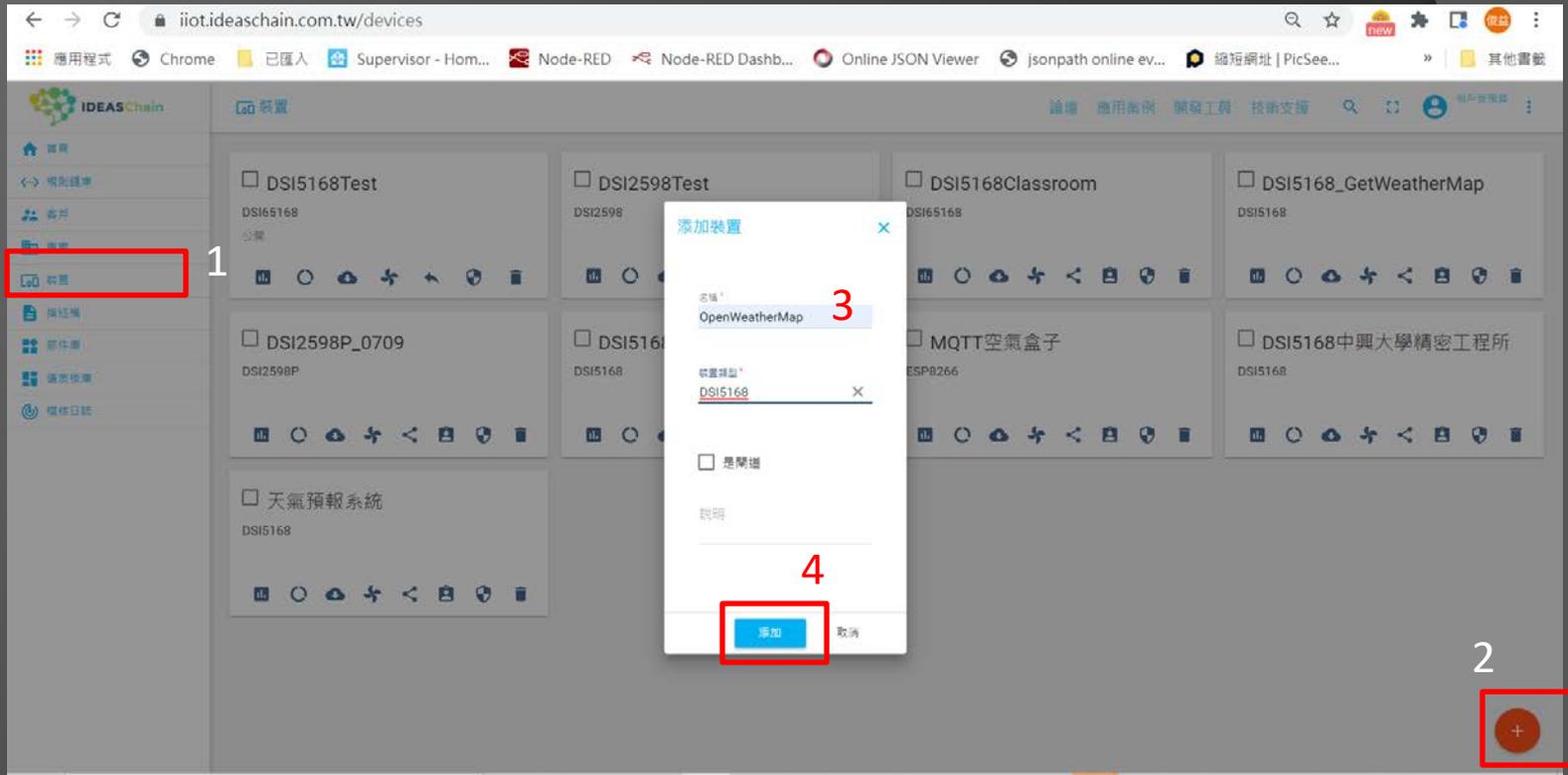
Step1. 至IDEAS Chain並點選數據平台: <https://iiot.ideaschain.com.tw/home> (請先建立帳號)在此平台建立專屬專案，並連接儀表版

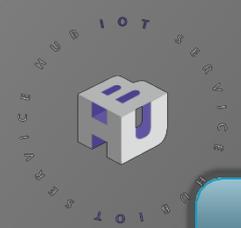




IDEASChain 設定

Step2. 點選左側裝置:，再點選右下角的+，添加裝置，填寫名稱類型後，點選添加

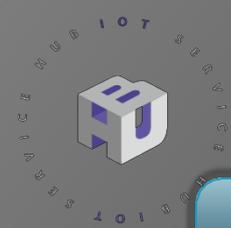




IDEASChain 設定

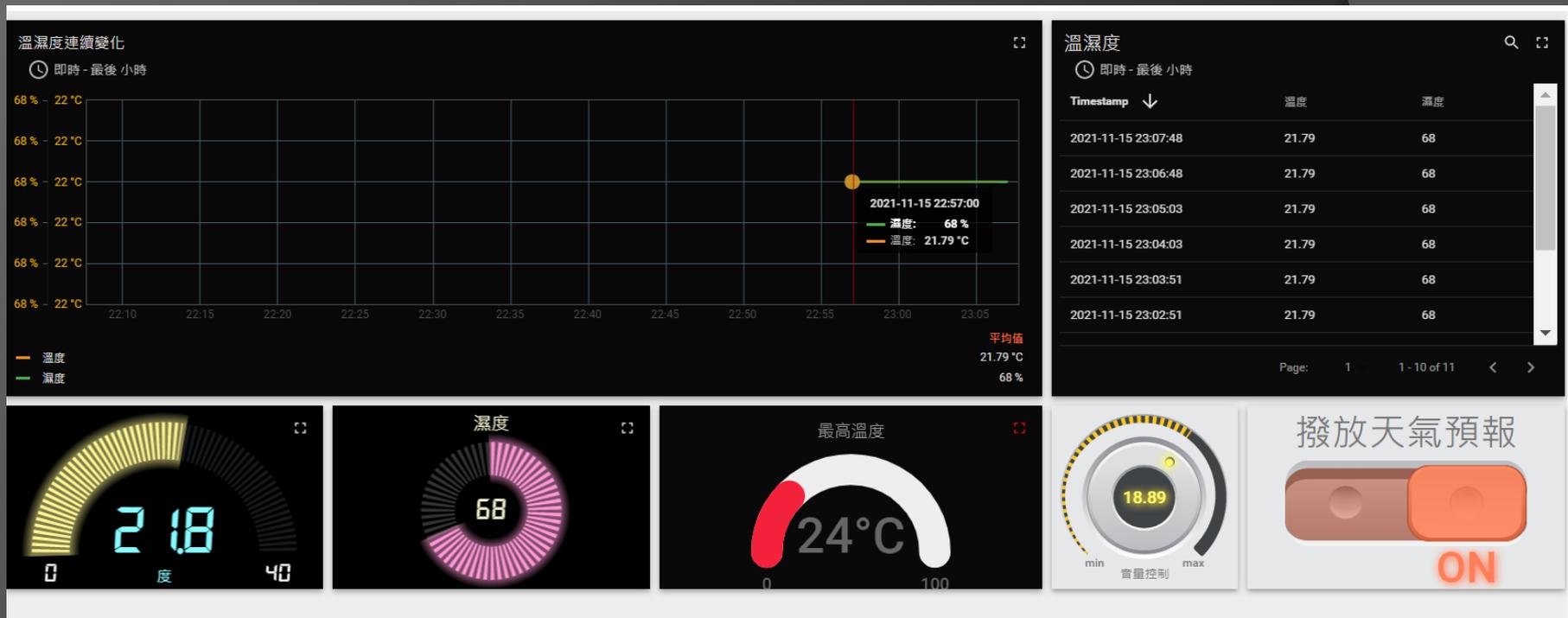
Step3. 點選左側:，再點選剛才新增的裝置，並複製存取權杖，貼上於程式碼中

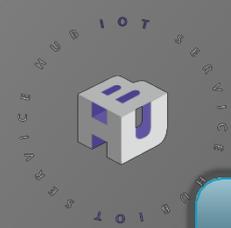
The screenshot displays the IDEASChain web interface. On the left is a navigation menu with options like '首頁', '規則匯庫', '客戶', '專案', '裝置', '描述檔', '部件庫', '儀表板庫', and '稽核日誌'. The main area shows a list of devices under the '裝置' tab. The selected device, '天氣預報系統' (Weather Forecast System), is shown in a detailed modal window. This modal window includes tabs for '詳細信息', '屬性', '最新遙測', '警告', '事件', '關聯', and '稽核日誌'. Below these tabs are buttons for '公開', '指派給客戶', '管理認證', and '刪除裝置'. A red box highlights the '複製存取權杖' (Copy Access Token) button. The modal also shows fields for '名稱*' (Name: 天氣預報系統) and '裝置類型*' (Device Type: DSI5168).



IDEASChain 設定

Step4. 點選左側:，再點選右下角的+，建立新的儀表板





程式架構

使用 NTP 服務

抓取OpenWeatherMap 天氣資料

Sounf of Text → MP3 IDEASChain

MQTT Broker 傳送

LINE Notify

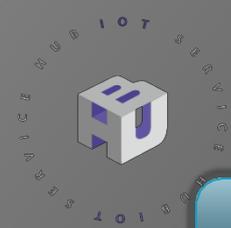
MQTT RPC 服務

程式解說

程式修改處

填寫裝置的存取權杖

```
DSI5168_GetWeather2IDEASChain-20211103-mp3
22
23 //-----
24 char mqttServer[]    = "iiot.ideaschain.com.tw"; // new ideaschain dashboard MQTT server
25 int mqttPort        = 1883;
26 char clientId[]     = "GetOpenWeatherMapxxx"; // xxx 為你的電話末3碼
27 char username[]     = "XhFtLYyjoNDp91k "; // device access token(存取權杖)
28 char password[]     = ""; // no need password
29 char subscribeTopic[] = "v1/devices/me/telemetry";
30 char publishTopic[]  = "v1/devices/me/telemetry"; // Fixed topic. ***DO NOT MODIFY***
31 char subscribeTopicRPC[] = "v1/devices/me/rpc/request/+"; // RPC MQTT
32 char publishPayload[] = "{\"RealTemp\":\"30\"}"; // String of stringified JSON Object (key value pair)
33
34 #define MQTT_RECONNECT_INTERVAL 100 // millisecond
35 #define MQTT_LOOP_INTERVAL      50 // millisecond
36 #define BUTTON_PIN              13 // 按鈕Pin
37 //
```



程式解說

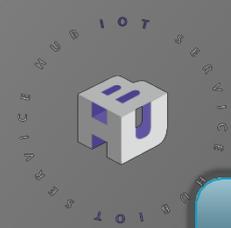
程式 Publish

組合成 JSON 字串

Publish to IDEASChain
MQTT Broker

```
214
215   Buff += nowTemp ;           //數值轉為 String
216   json_doc["NowTemp"] = Buff;
217
218   Buff = "";
219   Buff += humidity ;
220   json_doc["Humi"] = Buff;
221
222   Buff = "";
223   Buff += maxTemp ;
224   json_doc["MaxTemp"] = Buff;
225
226   Buff = "";
227   Buff += minTemp ;
228   json_doc["MinTemp"] = Buff;
229
```

```
serializeJson(json_doc, json_output);
Serial.print("JSON Str: ");
Serial.println(json_output);
mqttClient.publish(publishTopic, json_output);
delay(800);
```



程式解說

程式 MQTT RPC

當調整音量旋鈕時,在序列埠視窗中會看到 RPC method 變數 **vol** 的 params 數值會變動
當播放天氣預報 按鈕啟動在序列埠視窗中會看到 RPC method 變數 **music** 的 params 數值會變 **true**

The screenshot shows a serial port monitor window (COM28) displaying MQTT RPC messages. The messages are as follows:

```
Message arrived with Topic [v1/devices/me/rpc/request/758]
Data Length: [33], Payload: [{"method":"vol","params":"18.89"}]
params = 18.89
method = vol
nowTemp: 21.79
maxTemp: 23.79
minTemp: 20.31
JSON Str: {"NowTemp":"21.79","Humi":"68","MaxTemp":"23.79","MinT
Message arrived with Topic [v1/devices/me/rpc/request/759]
Data Length: [33], Payload: [{"method":"music","params":false}]
params = false
method = music
Message arrived with Topic [v1/devices/me/rpc/request/760]
Data Length: [32], Payload: [{"method":"music","params":true}]
params = true
method = music
sending NTP packet...
```

Red arrows point from the text above to the corresponding lines in the serial monitor output. A blue arrow points from the text above to the 'params' field in the second message.

The background shows a weather forecast interface with a table of data:

Timestamp	溫度	濕度
2021-11-15 23:04:03	21.79	68
2021-11-15 23:03:51	21.79	68
2021-11-15 23:02:51	21.79	68
2021-11-15 23:01:31	21.79	68
2021-11-15 22:59:39	21.79	68
2021-11-15 22:58:39	21.79	68

Below the table is a volume control knob showing 18.89 and a toggle switch labeled '撥放天氣預報' (ON).