

# 物聯網應用實作 —以溫溼度感測為主



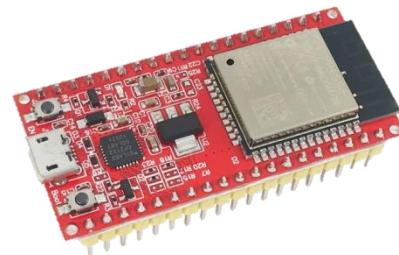
指導老師：尹廉輝老師  
學生：2年11班 許顯勳



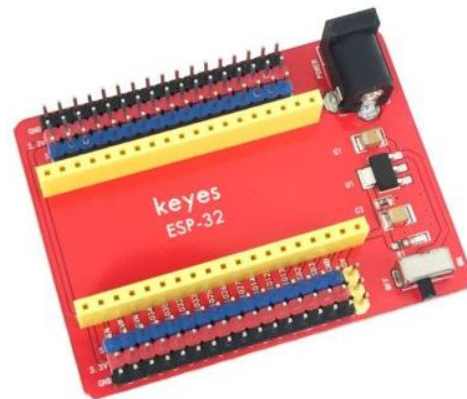


# ESP32介紹

- ESP32 是一系列整合 Wi-Fi 和雙模式藍牙的單晶片**微控制器**
- 擁有多個開放原始碼開發環境以及程式庫
- 除了常見的 GPIO 輸入輸出，ESP32 也支援許多通訊方式，並內建Wi-Fi 晶片，能夠使用網路功能

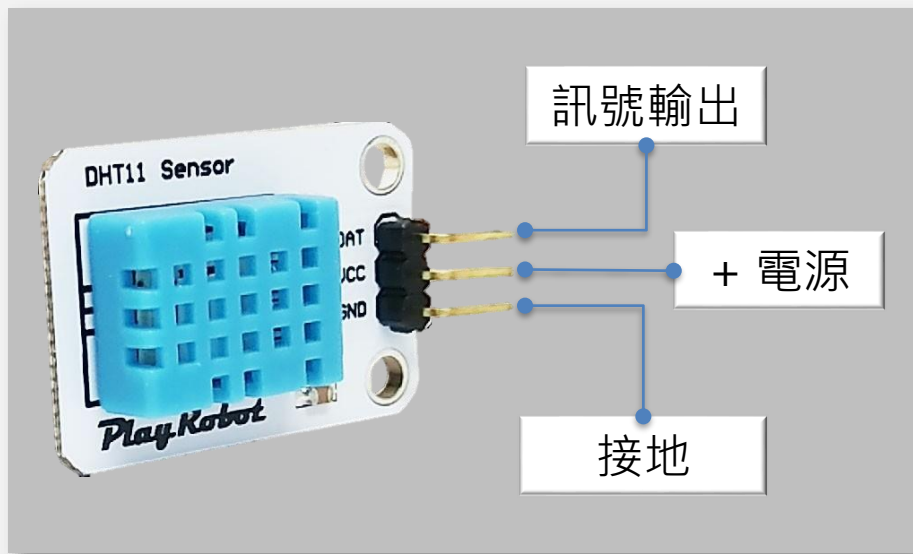


▲ ESP32 主板



▲ ESP32 擴充板

# DHT 溫溼度感測



DHT 感測器包含：

1. 電阻式感濕元件\*1

➔ 根據濕度變化使感測元件電阻值改變，來判斷濕度

2. NTC測溫元件\*1

➔ 隨溫度變高使感測電阻變低，以輸出換算後的溫度

3. 與一個高性能8位單晶片相連

➔ 用來將上述兩者整合在一塊積體電路晶片上的微型計算機

# Wi-Fi 連接應用

- ESP32 內建了 wifi 晶片，可透過引入 <WiFi.h> 函式庫來呼叫 `WiFi.begin(ssid,password)` 入程式碼以連上網路

## 我們將使用到的通訊協定

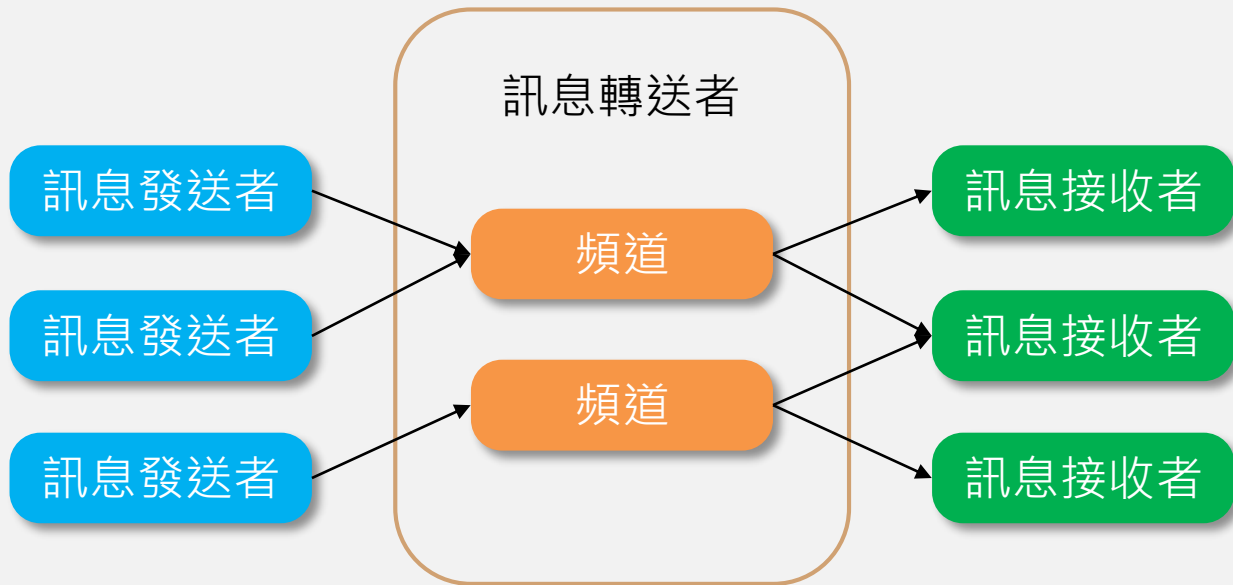
### 1. Http Request

用來使用網路資源，  
包含爬蟲、Line 通知、  
Blink 等

### 2. MQTT

讓我們在區域網或外  
網使用發布/訂閱機  
制來通訊

# MQTT 網路控制



# ◆ MQTT應用：開關燈控制 — 使用 MQTTBox

- 目的為了控制遠端的硬體，我們可利用MQTT來進行遠端操控

預先設定

The screenshot displays the MQTTBox web interface with the following elements and annotations:

- 1. 新增訂閱頻道**: A blue callout box pointing to the gear icon in the top right navigation bar.
- 2. 輸入發送者名稱**: A blue callout box pointing to the "playrobot" text in the "Topic to publish" input field.
- 3. 輸入接收者名稱**: A blue callout box pointing to the "playrobot" text in the "Topic to subscribe" input field.
- 4. 訂閱**: A blue callout box pointing to the orange "Subscribe" button.

The interface includes a top navigation bar with "Menu", "Connected" status, and "Add publisher" / "Add subscribe" buttons. The main content area is split into two panels: "Topic to publish" (left) and "Topic to subscribe" (right). Both panels have a "QoS" dropdown set to "0 - Almost Once" and a "Retain" checkbox. The "Payload Type" dropdown is set to "Strings / JSON / XML / Characters". The "Payload" field is empty. A "Publish" button is located at the bottom of the "Topic to publish" panel.

# ◆ 編寫程式

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
```

引入函式庫

```
3
4 // 請按照各位的網路設定輸入
5 const char* ssid = " ";
6 const char* password = " ";
7 const char* mqtt_server = "broker.hivemq.com";
8 #define mqtt_port 1883
9 #define MQTT_SERIAL_PUBLISH_CH "playrobotToESP32"
10 #define MQTT_SERIAL_RECEIVER_CH "playrobot"
11
12 WiFiClient wifiClient;
13 PubSubClient client(wifiClient);
```

網路資料設定  
(宣告變數)

```
14
15 void setup_wifi() {
16     delay(10);
17     // wifi連線
18     WiFi.begin(ssid, password);
19     while (WiFi.status() != WL_CONNECTED) delay(500);
20     randomSeed(micros());
21 }
```

Wifi 連線

```
22
23 void reconnect() {
24     // MQTT重新連線
25     while (!client.connected()) {
26         // 連線需要一個ID，用亂數解決
27         String clientId = "ESP32Client-";
28         clientId += String(random(0xffff), HEX);
29
30         if (client.connect(clientId.c_str())) {
31             Serial.println("connected");
32             //連上之後發一個訊息證明一下。
33             client.publish("playrobotToESP32", "hello world");
34             //訂閱的頻道
35             client.subscribe(MQTT_SERIAL_RECEIVER_CH);
36         }
37         else {
38             delay(5000);
39         }
40     }
41 }
```

MQTT  
重新連線

```
43 void callback(char* topic, byte *payload, unsigned int length) {
44     Serial.print("data:");
45     String messageTemp;
46
47     for (int i = 0; i < length; i++) {
48         Serial.print((char)payload[i]);
49         messageTemp += (char)payload[i];
50     }
51
52     if(messageTemp=="on") digitalWrite(13, 1);
53     if(messageTemp=="off") digitalWrite(13, 0);
54 }
```

LED 燈控制

```
55
56 void setup() {
57     Serial.begin(115200);
58     Serial.setTimeout(500);
59     setup_wifi();
60     client.setServer(mqtt_server, mqtt_port);
61     client.setCallback(callback);
62     reconnect();
63     pinMode(13,OUTPUT);
64 }
```

接腳與網路設定

```
65
66 void publishSerialData(char *serialData){
67     if (!client.connected()) {
68         reconnect();
69     }
70     client.publish(MQTT_SERIAL_PUBLISH_CH, serialData);
71 }
```

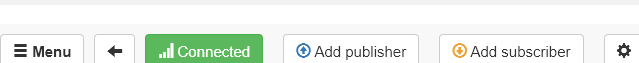
發送訊息

```
72
73 void loop() {
74     client.loop();
75     if (Serial.available() > 0) {
76         char mun[501];
77         memset(mun,0, 501);
78         Serial.readBytesUntil( '\n',mun,500);
79         publishSerialData(mun);
80     }
81 }
```

主要程式  
(重複執行)



# ◆ 執行結果



Playrobot - mqtt://broker.hivemq.com

Topic to publish  
playrobot

QoS  
0 - Almost Once

Retain

Payload Type  
Strings / JSON / XML / Characters

e.g: {'hello':'world'}

Payload  
off

**輸入on或off**

**Publish**

**按下發布**

off  
topic:playrobot, qos:0, retain:false

on

playrobot

off

qos : 0, retain : false, cmd : publish, dup : false, topic : playrobot, messageId : , length : 14

on

qos : 0, retain : false, cmd : publish, dup : false, topic : playrobot, messageId : , length : 13

Hello Playrobot!

qos : 0, retain : false, cmd : publish, dup : false, topic : playrobot, messageId : , length : 27

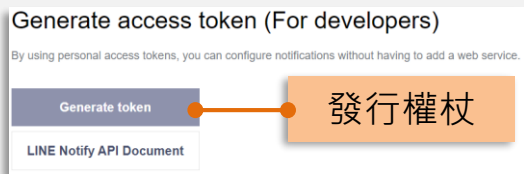
燈暗 ; 序列視窗  
顯示: "data:off"

燈亮 ; 序列視窗  
顯示: "data:on"

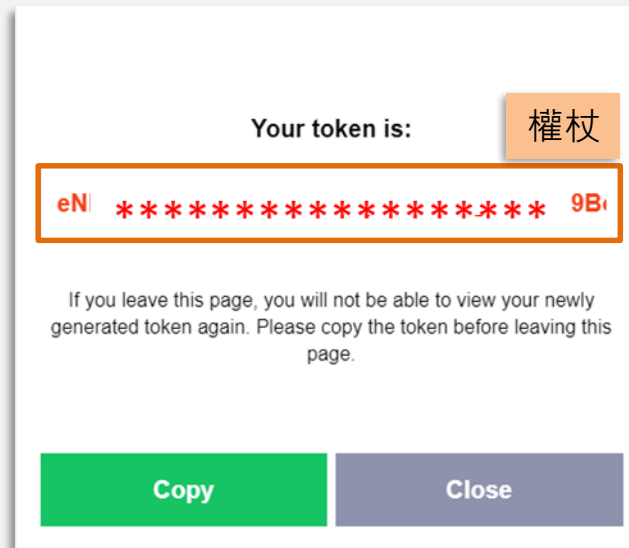
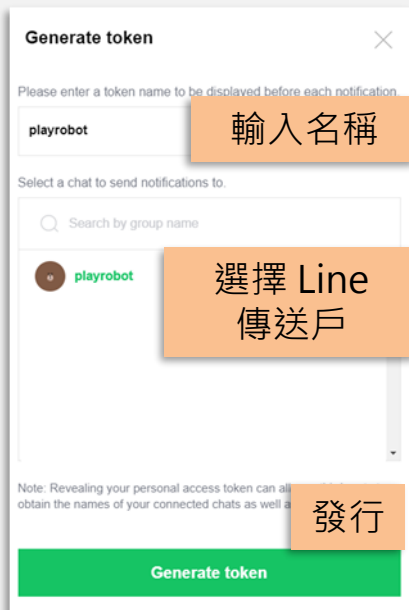
# LINE 通知

- 要使用LINE服務，首先須先取得**權杖**，搜尋LINE Notify：

登入個人頁面後：



- 因為我們無法永遠盯著遠端監控，所以利用Line 通知來告訴我們遠端裝置的狀況



# ● 編寫程式

```
1 #include <WiFi.h>
2 #include <WiFiClientSecure.h>
3 #include <SimpleDHT.h>
```

引入函式庫

```
4
5 //請依照網路環境修改
6 const char* SSID = "";
7 const char* PASSWORD = "";
8 const String Linetoken = "";
9
10 int pinDHT11 = 15;
11 SimpleDHT11 dht11(pinDHT11);
12 WiFiClientSecure client;//網路連線物件
13 const char* host = "notify-api.line.me";//LINE Notify API網址
14
```

網路資料設定  
(宣告變數)

```
15 void setup() {
16     Serial.begin(115200);
17     //連線到WiFi
18     WiFi.begin(SSID, PASSWORD);
19     while (WiFi.status() != WL_CONNECTED) {
20         Serial.print(".");
21         delay(500);
22     }
23     Serial.println("WiFi connected");
24 }
25
```

Wifi 連線

```
26 void loop() {
27     //讀取溫度內容
28     byte temperature = 0;
29     byte humidity = 0;
30     dht11.read(&temperature, &humidity, NULL);
31
```

讀取溫濕度

```
32     String message = "";
33     message += "溫度=" + String(((int)temperature)) + "C";
34     message += "濕度=" + String(((int)humidity));
35     Serial.println(message);
36     client.setInsecure();
37     if (client.connect(host, 443)) {
38         int LEN = message.length();
39         //利用http傳送
40         String url = "/api/notify";
41         client.println("POST " + url + " HTTP/1.1");
42         client.print("Host: "); client.println(host);
43         client.print("Authorization: Bearer "); client.println(Linetoken);
44         client.println("Content-Type: application/x-www-form-urlencoded");
45         client.print("Content-Length: "); client.println( String((LEN + 8) ));
46         client.println();
47         client.print("message="); client.println(message);
48         client.println();
49         client.stop(); //斷線, 否則只能傳5次
50     }
51     //每5秒發送一次, 覺得太頻繁可以把時間拉長...
52     delay(5000);
53 }
```

以 http 傳送



## ● 執行結果

透過網路傳輸溫溼度的讀值，每隔5秒傳送到 LINE Notify 加入的群組中

## 總結

- 從上課的實作中，我們實際體驗了硬體設備是如何經由網路來遠端操控，得知遠距離的訊息、資料，並了解感測數據是如何一步步從讀取、數據化、傳送再重複或有條件的送到我們手上
- 實作應用內容有許多複雜的地方，常會遇到各方面的阻礙，如程式的語意錯誤、運作原理與概念的澄清等等
- 經過課程所有的實作練習，有更進一步貼近物聯網的感覺，了解現代的趨勢，以及未來持續研究的方向

感謝聆聽