



減碳靈感

古都土城仔綠電創能與智動養殖
之跨界整合永續淨零發展計畫

水位感測器

MQTT應用



目錄



材料

程式撰寫說明

模組腳位說明

寫入程式步驟

接線說明

序列埠查看資訊



減碳靈感

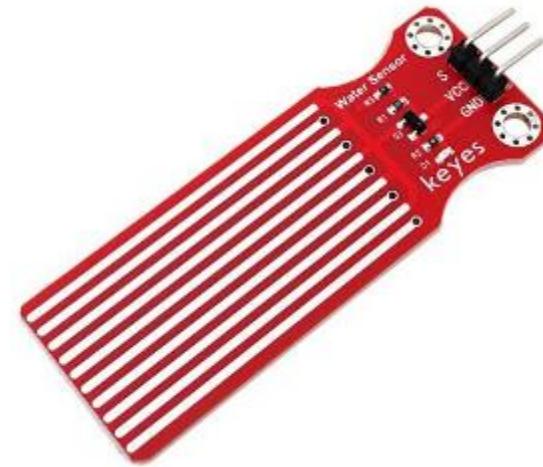
材料



ESP32



數據傳輸線 (MicroUSB)



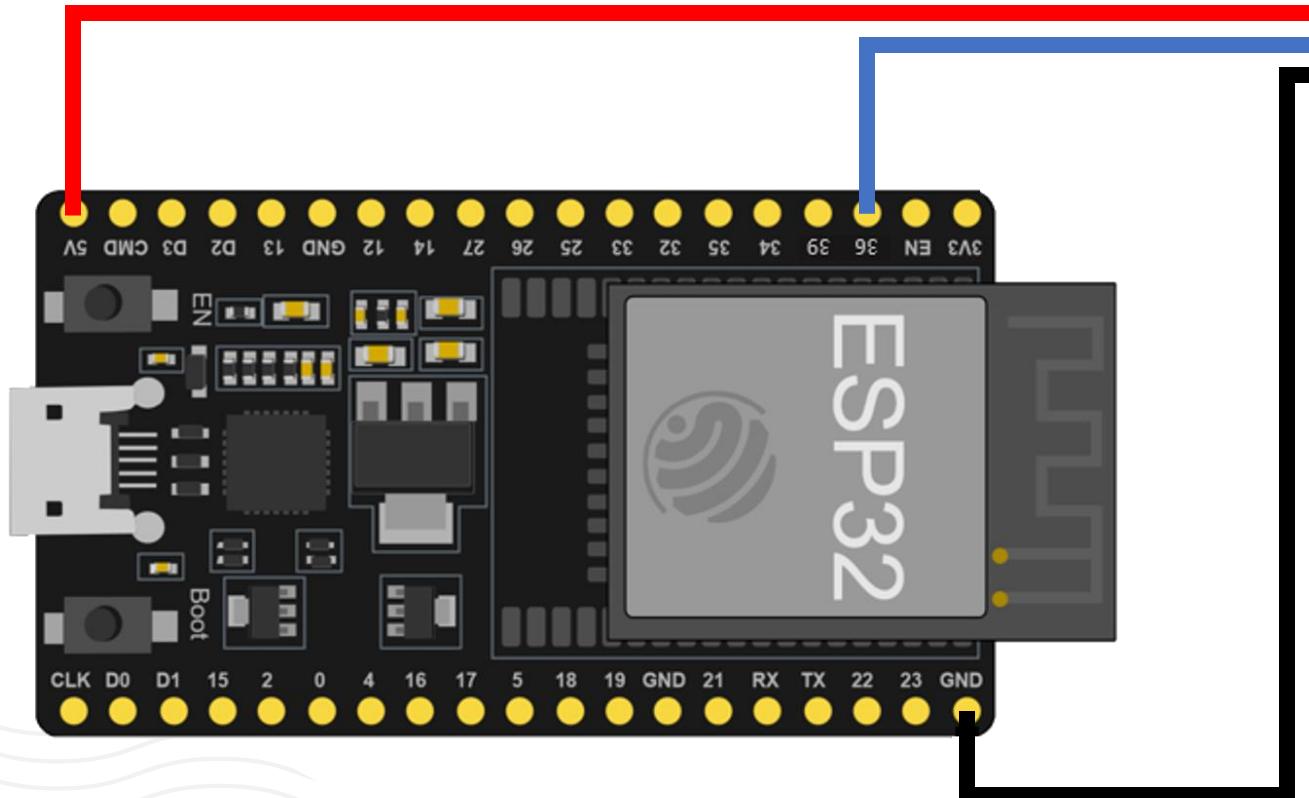
水位感測器模組

模組腳位說明



S	+	-
DATA	VCC	GND

接線說明



S	GPIO 36
+	3.3V or 5V
-	GND

程式撰寫步驟



- 開啟記事本
「範例程式 水位感測器 MQTT.txt 」

- 複製內容並貼上Arduino視窗中



減碳綠能

程式撰寫步驟

➤ 修改Wi-Fi資訊

ssid = "名稱"
password = "密碼"

```
#include <WiFi.h>
#include <PubSubClient.h> //請先安裝PubSubClient程式庫

// ----- 以下修改成你自己的WiFi帳號密碼 -----
char* ssid = "YourSSID";
char* password = "YourPASSWORD";

//----- 以下修改成你的水位感測器腳位 -----
int water_sensor = 36; //水位感測器 S 信號腳 連接到ESP32的 GPIO36
int a;

// ----- 以下修改成你MQTT設定 -----
char * MQTTServer = "broker.mqtgo.io"; //免註冊MQTT伺服器
int MQTTPort = 1883; //MQTT Port
char * MQTTUser = ""; //不須帳密
char * MQTTPassword = ""; //不須帳密
```

程式撰寫步驟

➤ 更改MQTTPubTopic，避免與他人重複

```
//推播主題1:推播水位    自行更改路徑，例如：TEST/class402/WaterLevel
char* MQTTPubTopic1 = "YourTopic/class402/WaterLevel";
long MQTTLastPublishTime;//此變數用來記錄推播時間
long MQTTPublishInterval = 3000;//每3秒推撥一次
WiFiClient WifiClient;
PubSubClient MQTTClient(WifiClient);
```

程式撰寫步驟

```
void setup() {  
    Serial.begin(115200); //設定通訊鮑率  
    pinMode(water_sensor,INPUT); //設置water_sensor對應的腳GPIO36為輸入  
  
    //開始WiFi連線  
    WifiConnecte();  
  
    //開始MQTT連線  
    MQTTConnecte();  
}
```

程式撰寫步驟

```

void loop() {
    //如果WiFi連線中斷，則重啟WiFi連線
    if (WiFi.status() != WL_CONNECTED) { WifiConnecte(); }

    //如果MQTT連線中斷，則重啟MQTT連線
    if (!MQTTClient.connected()) { MQTTConnecte(); }

    //如果距離上次傳輸已經超過1秒，則Publish距離
    if ((millis() - MQTTLastPublishTime) >= MQTTPublishInterval ) {

        int val=analogRead(water_sensor); //從水位感測器讀出類比數值
        a=map(val,0,4095,0,100);        //將val轉換成百分比顯示
        Serial.print("水位狀態：");
        Serial.print(val);
        Serial.println(" val");
        Serial.print("水位百分比：");
        Serial.print(a);
        Serial.println(" %");
    }
}

```

程式撰寫步驟

```

// ----- 將水位送到MQTT主題 -----
MQTTClient.publish(MQTPubTopic1, String((int)a).c_str());
Serial.println("水位已推播到MQTT Broker");
MQTTLastPublishTime = millis(); //更新最後傳輸時間
}

MQTTClient.loop();//更新訂閱狀態
delay(50);

}

//開始WiFi連線
void WifiConnecte() {
//開始WiFi連線
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("WiFi連線成功");
Serial.print("IP Address:");
Serial.println(WiFi.localIP());
}

//----- 將水位送到MQTT主題 -----
void MQTTConnecte() {
MQTTClient.setServer(MQTTServer, MQTTPort);
while (!MQTTClient.connected()) {
//以亂數為ClientID
String MQTTClientid = "esp32-" + String(random(1000000, 9999999));
if (MQTTClient.connect(MQTTClientid.c_str(), MQTTUser, MQTTPassword)) {
//連結成功，顯示「已連線」。
Serial.println("MQTT已連線");

} else {
//若連線不成功，則顯示錯誤訊息，並重新連線
Serial.print("MQTT連線失敗,狀態碼=");
Serial.println(MQTTClient.state());
Serial.println("五秒後重新連線");
delay(5000);
}
}
}

```

寫入程式步驟

➤ 1.確定工具欄位下的選項有正確選擇

➤ 2.確認後點擊上傳

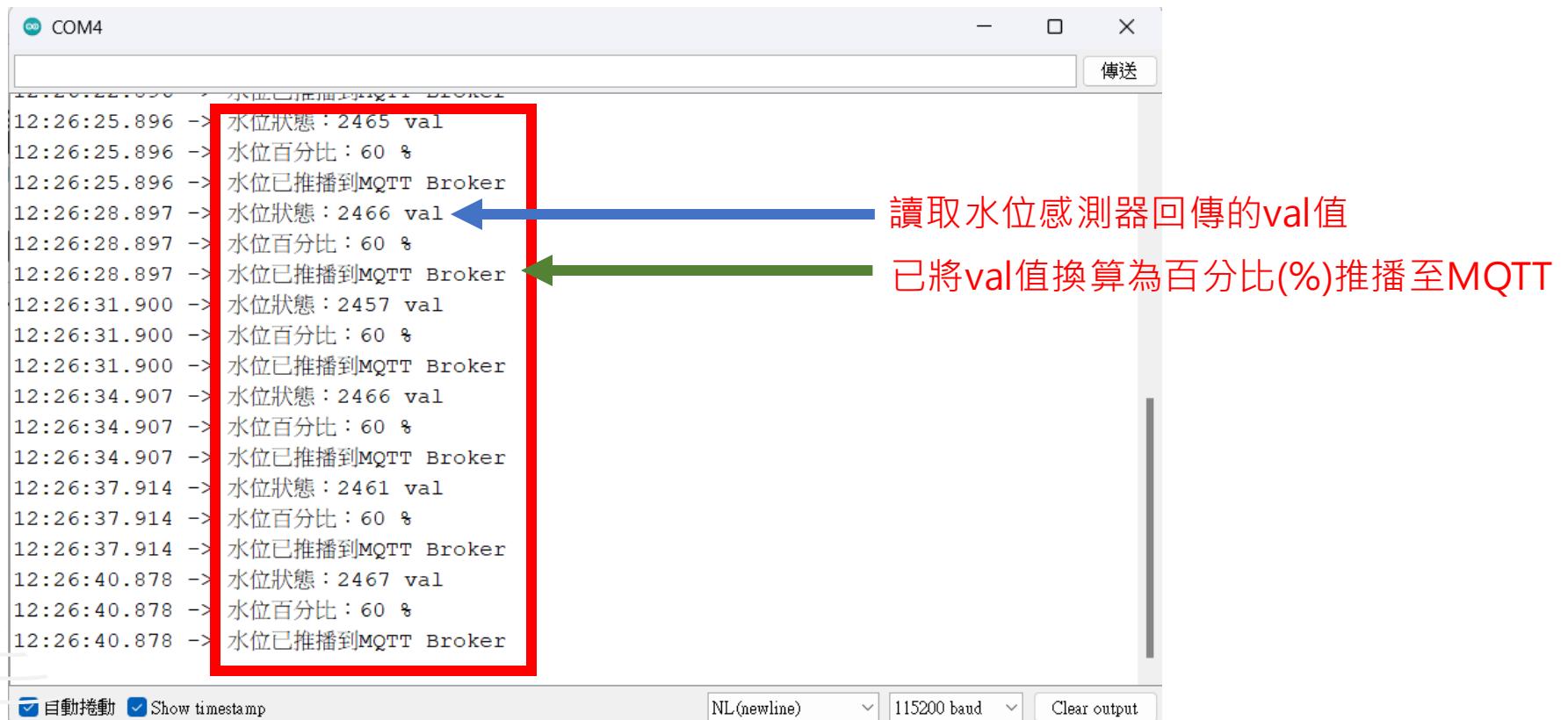


➤ 3.等待底下出現此字串即成功

```
Leaving...
Hard resetting via RTS pin...
```

查看資訊

- 開啟右上角序列埠監控視窗即可查看水位資訊
- 水位感測器需要**至少30秒數值才會穩定**，觀測前先淨置一段時間



```

COM4
12:26:25.896 -> 水位狀態: 2465 val
12:26:25.896 -> 水位百分比: 60 %
12:26:25.896 -> 水位已推播到MQTT Broker
12:26:28.897 -> 水位狀態: 2466 val
12:26:28.897 -> 水位百分比: 60 %
12:26:28.897 -> 水位已推播到MQTT Broker
12:26:31.900 -> 水位狀態: 2457 val
12:26:31.900 -> 水位百分比: 60 %
12:26:31.900 -> 水位已推播到MQTT Broker
12:26:34.907 -> 水位狀態: 2466 val
12:26:34.907 -> 水位百分比: 60 %
12:26:34.907 -> 水位已推播到MQTT Broker
12:26:37.914 -> 水位狀態: 2461 val
12:26:37.914 -> 水位百分比: 60 %
12:26:37.914 -> 水位已推播到MQTT Broker
12:26:40.878 -> 水位狀態: 2467 val
12:26:40.878 -> 水位百分比: 60 %
12:26:40.878 -> 水位已推播到MQTT Broker

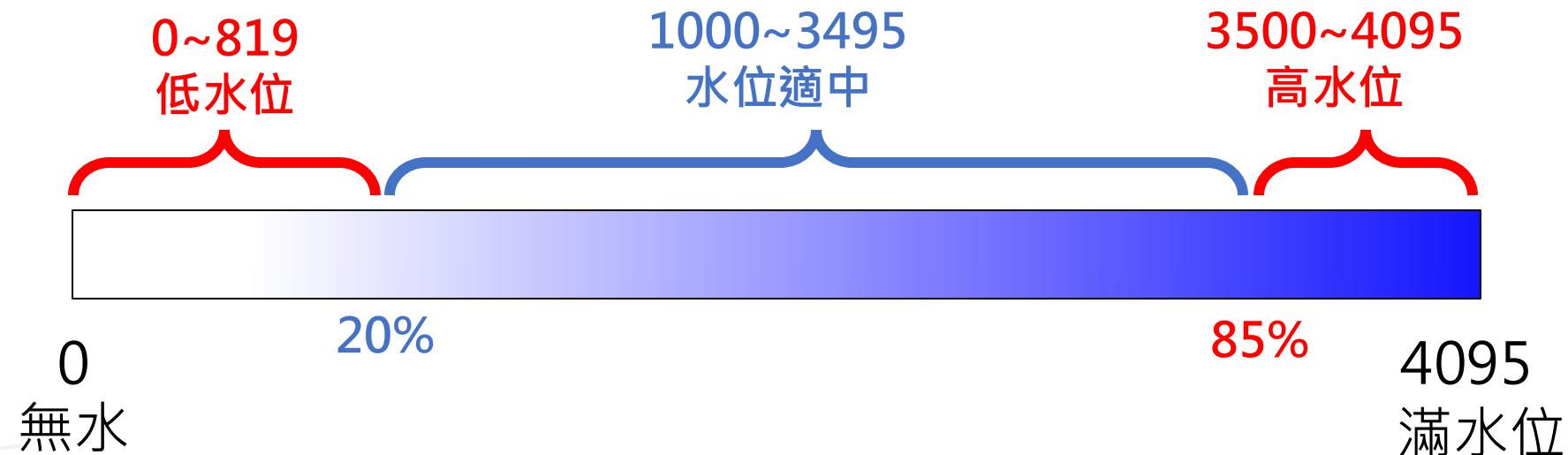
```

自動捲動 Show timestamp NL(newline) 115200 baud Clear output

查看資訊

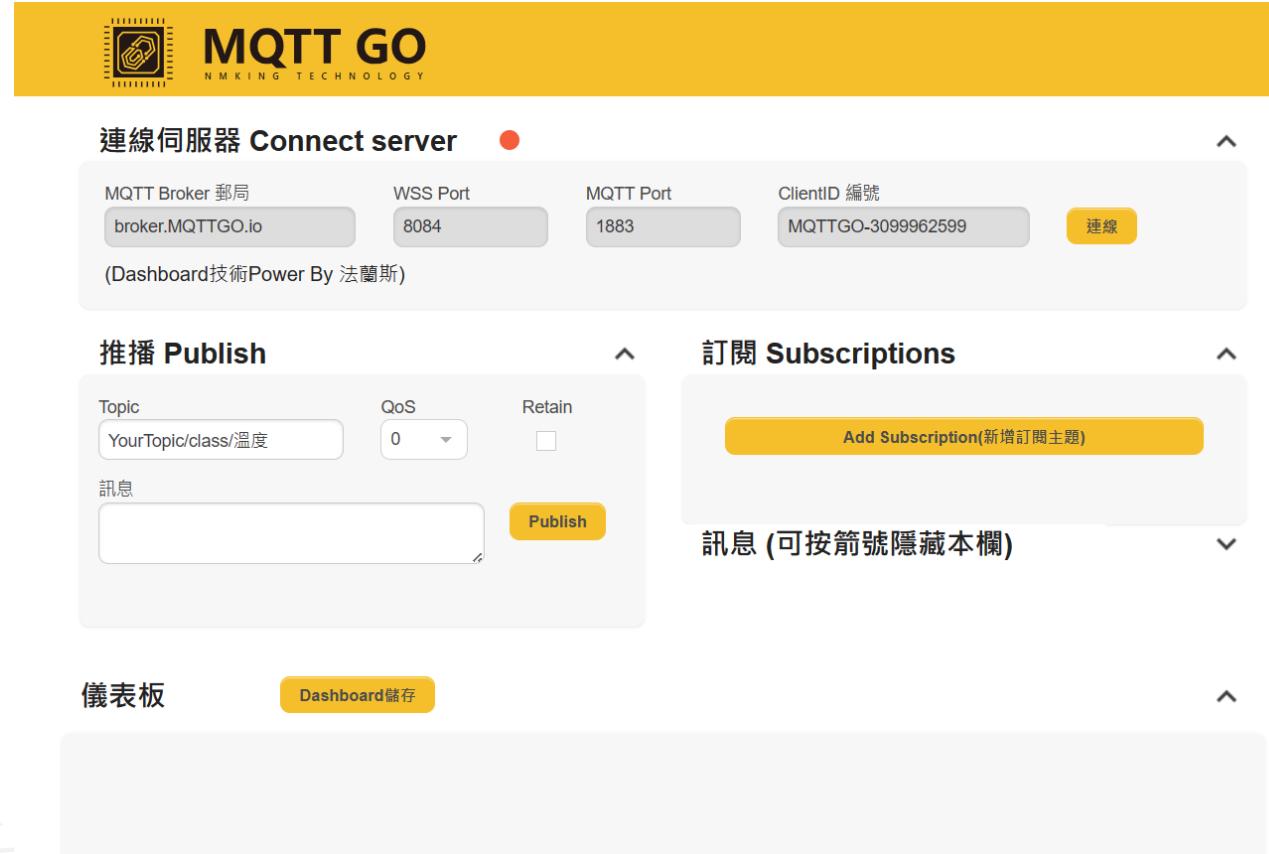


➤ ESP32 感測器VCC在5V下的val值判斷對照



MQTT查看資訊

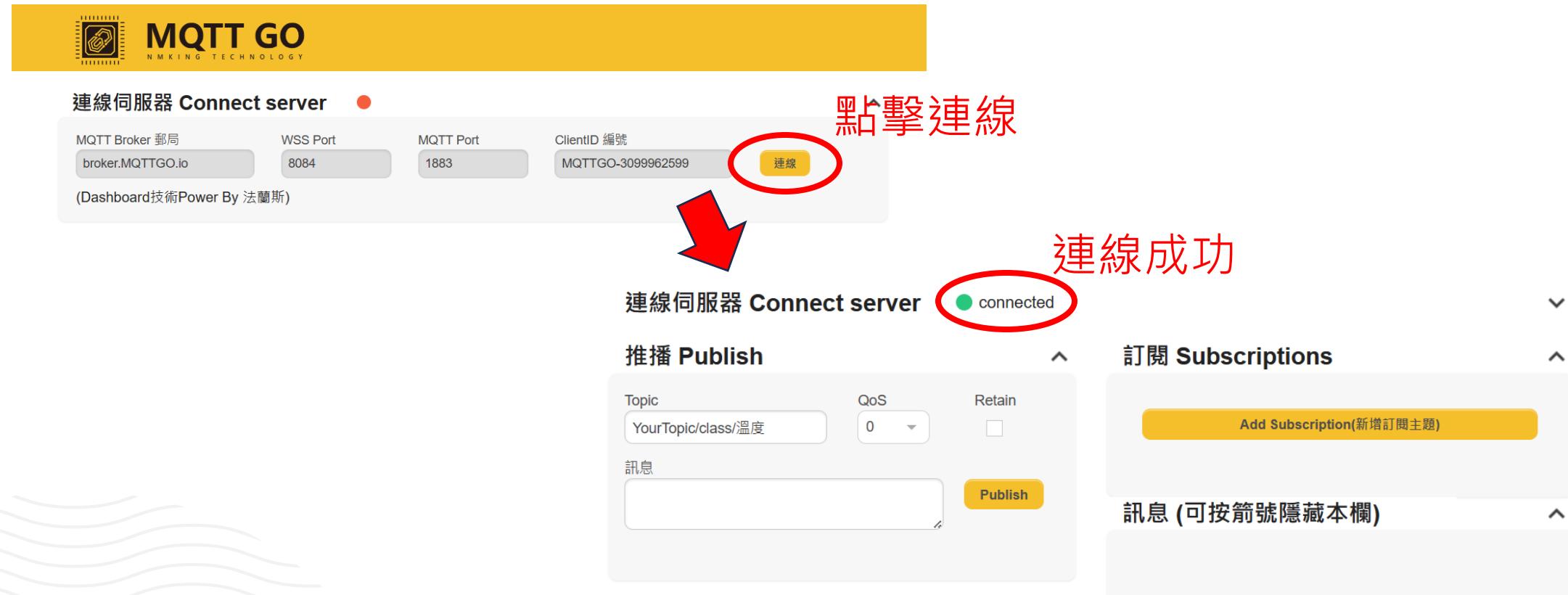
➤ 於瀏覽器開啟網站：<https://broker.mqttgo.io/>



The screenshot shows the MQTT GO web interface. At the top, there's a yellow header bar with the 'MQTT GO' logo and 'N MKING TECHNOLOGY'. Below it, a 'Connect server' section shows the MQTT Broker as 'broker.MQTTGO.io', WSS Port as '8084', MQTT Port as '1883', and ClientID as 'MQTTGO-3099962599'. A yellow '連線' (Connect) button is visible. The main area has two expandable sections: 'Publish' and 'Subscriptions'. The 'Publish' section allows users to enter a Topic (e.g., 'YourTopic/class/溫度'), set QoS (0), and choose Retain. It also has a message input field and a 'Publish' button. The 'Subscriptions' section has a 'Add Subscription(新增訂閱主題)' button and a message list placeholder '訊息 (可按箭號隱藏本欄)'. At the bottom, there's a 'Dashboard' section with a 'Dashboard儲存' button.

MQTT查看資訊

➤ 點擊連線，待燈號亮綠燈顯示connected即連線成功



The screenshot shows the MQTT GO dashboard interface. At the top, there's a yellow header bar with the title "MQTT GO" and the subtitle "NMKING TECHNOLOGY". Below the header, there's a "Connect server" section with fields for "MQTT Broker 郵局" (set to "broker.MQTTGO.io"), "WSS Port" (set to "8084"), "MQTT Port" (set to "1883"), and "ClientID 編號" (set to "MQTTGO-3099962599"). A red arrow points from the "MQTT Port" field down to the "連線" (Connect) button, which is highlighted with a red circle. To the right of this section, the text "點擊連線" (Click to connect) is written in red. A large red arrow points downwards from the "連線" button to the "Connect server" status area. In this area, the word "connected" is displayed next to a green circular icon, also highlighted with a red circle. To the right of this, the text "連線成功" (Connection successful) is written in red. Below the "Connect server" section, there are two main functional areas: "Publish" and "Subscriptions". The "Publish" area contains fields for "Topic" (set to "YourTopic/class/溫度"), "QoS" (set to "0"), and "Retain" (unchecked). It also has a "訊息" (Message) input field and a "Publish" button. The "Subscriptions" area has a "Add Subscription(新增訂閱主題)" button.



MQTT查看資訊

➤ 回到程式碼，將以下框中Topic文字複製下來

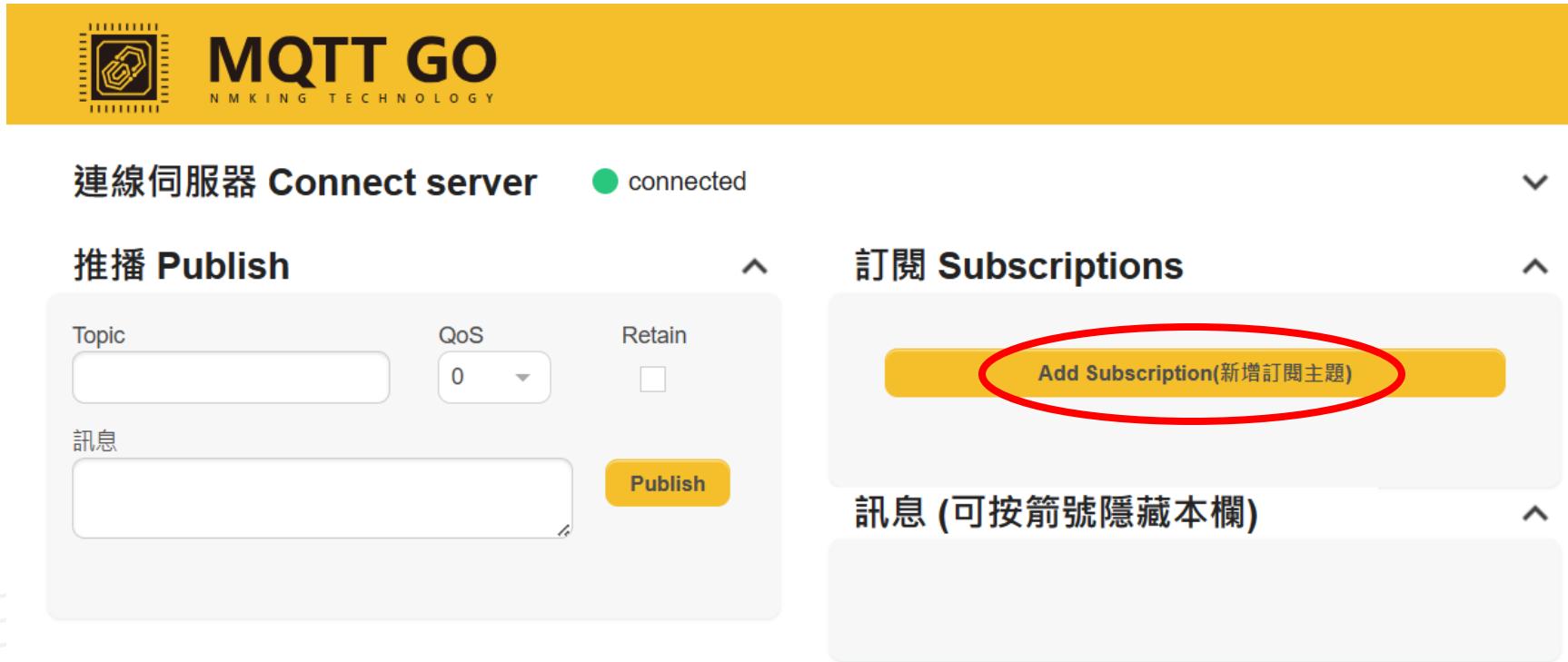
```
//推播主題1:推播土壤濕度          複製
char* MQTTPubTopic1 = "YourTopic/class402/WaterLevel";
long MQTTLastPublishTime;//此變數用來記錄推播時間
long MQTTPublishInterval = 1000;//每1秒推撥一次
WiFiClient WiFiClient;
PubSubClient MQTTClient(WIFIClient);
```



減碳綠能

MQTT查看資訊

➤ 回到MQTT GO，點選新增訂閱主題



The screenshot shows the MQTT GO interface. At the top, there's a yellow header bar with the logo "MQTT GO N M K I N G T E C H N O L O G Y". Below the header, the status "connected" is displayed next to a green dot icon. The interface is divided into two main sections: "Push Publish" on the left and "Subscription Subscriptions" on the right.

Push Publish section:

- Topic input field.
- QoS dropdown menu set to 0.
- Retain checkbox.
- Message input field.
- Publish button.

Subscription Subscriptions section:

- A yellow button labeled "Add Subscription(新增訂閱主題)" is circled in red.
- Below it, a message box says "訊息 (可按箭號隱藏本欄)".

MQTT查看資訊

➤ 將剛剛複製的路徑貼到Topic，按步驟修改完成後點擊Subscribe



The screenshot shows a configuration panel for an MQTT topic. The fields are as follows:

- Color:** A small blue square.
- QoS:** A dropdown set to 2.
- Topic:** A text input field containing "YourTopic/class402/WaterLevel".
- 儀表板 (Dashboard):** A dropdown menu with "水位圖" selected.
- 名稱 (ID):** An empty text input field.
- 數值區間 (min,max):** A text input field containing "0,100".
- 單位:** A text input field containing "%".
- Subscribe Button:** A yellow button labeled "Subscribe" with a red oval and number ④.

Red annotations with numbers ① through ④ provide instructions:

- ① Topic 貼上剛剛複製的Topic (Paste the just copied Topic)
- ② 選擇儀表板 (Select Dashboard)
- ③ 調整區間0~100 (Adjust range 0~100)
- ④ 完成後點擊Subscribe訂閱 (After completing, click the Subscribe button to subscribe)

MQTT查看資訊

➤ 訊息欄可看到接收到的資訊

訂閱 Subscriptions

Add Subscription(新增訂閱主題)

Qos: 2

YourTopic/class402/WaterLevel X

訊息 (可按箭號隱藏本欄)

2024-04-07 Topic: YourTopic/class40... Qos: 0

12:30:11

59

2024-04-07 Topic: YourTopic/class40... Qos: 0

12:30:08

59

2024-04-07 Topic: YourTopic/class40... Qos: 0

12:30:05

59

2024-04-07 Topic: YourTopic/class40... Qos: 0

12:30:02

59



減碳綠能

MQTT查看資訊

➤ 在下方儀表板也可清楚以圖示的方式得知資訊





減碳藍漁

古都土城仔綠電創能與智動養殖
之跨界整合永續淨零發展計畫

感謝聆聽
給予指導

