



減碳靈鷲

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

土壤溼度感測器

MQTT應用



目錄



材料

寫入程式步驟

模組腳位說明

序列埠查看資訊

接線說明

MQTT查看資訊

程式撰寫說明



減碳靈感

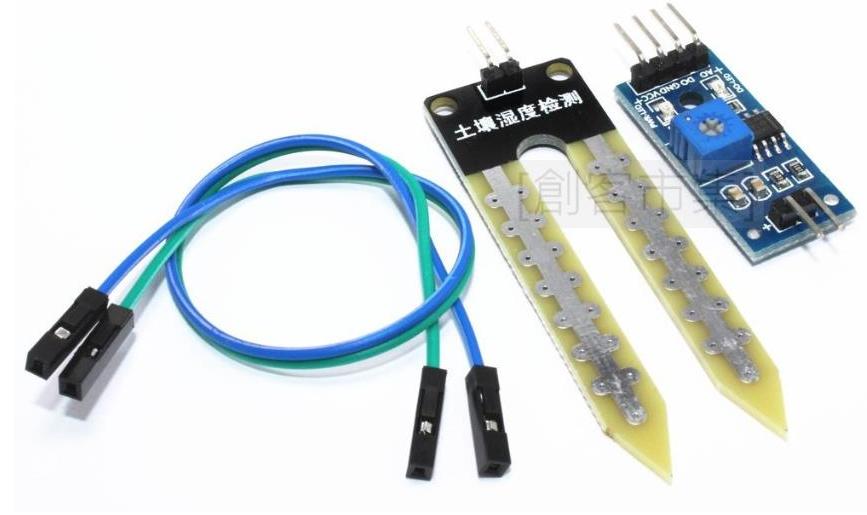
材料



ESP32



數據傳輸線 (MicroUSB)



土壤溼度感測器模組

模組腳位說明



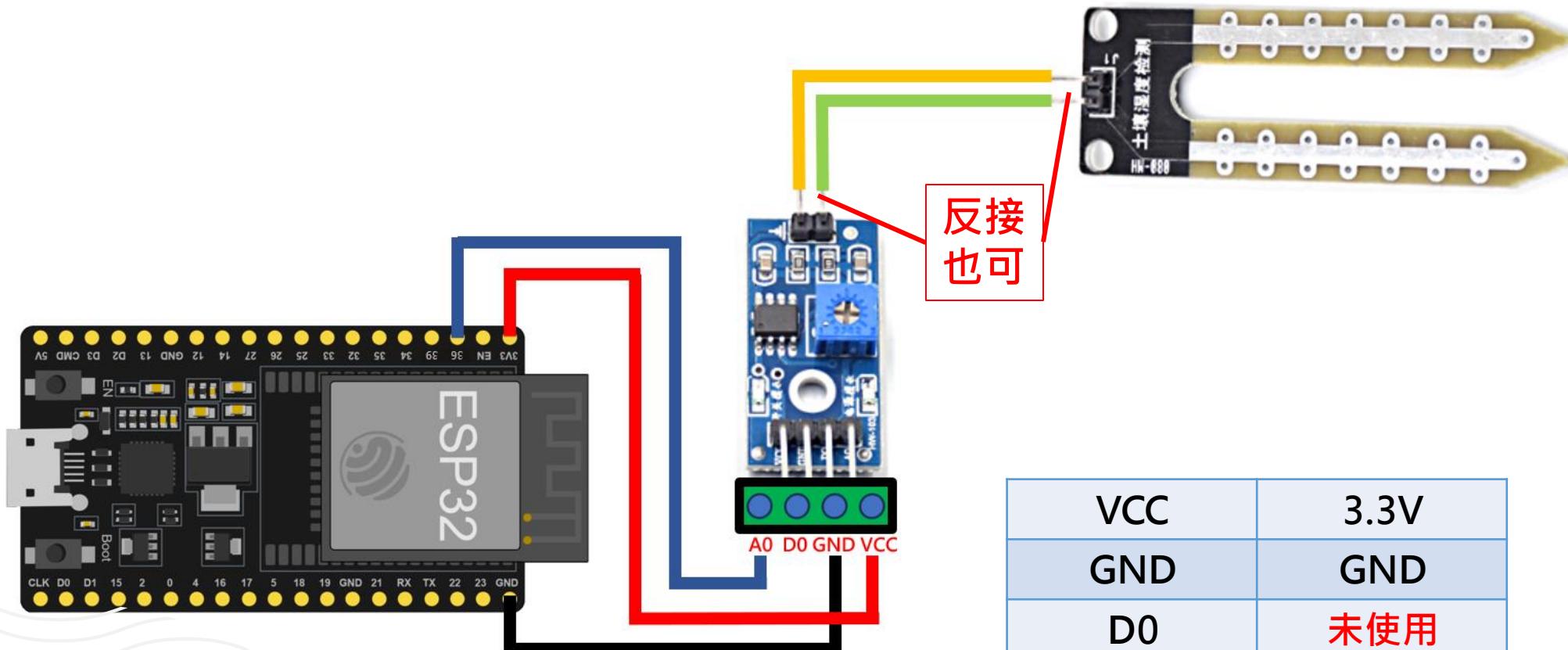
+	對接傳感器
-	對接傳感器

傳感器無正負極之分，與傳感器對接時可不必注意正負極！

靈敏度調整鈕，用於調整D0開關訊號的靈敏度

VCC	3.3V~5V(+)
GND	GND(-)
D0	開關訊號腳位
A0	模擬訊號腳位

接線說明



程式撰寫步驟



- 開啟記事本
「範例程式 土壤濕度感測器 MQTT.txt 」
- 複製內容並貼上Arduino IDE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";

//----- 以下修改成你的土壤濕度感測器腳位 -----
int soil_sensor = 36; //土壤濕度感測器AO信號腳 連接到ESP32 GPIO36

// ----- 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:推播土壤濕度 自行更改路徑，例如：TEST/class402/SoilWater
char* MQTTPubTopic1 = "YourTopic/class402/SoilWater";
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(soil_sensor,INPUT); //定義土壤濕度感測器接口為輸入接口。  
  
    //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(soil_sensor); //土壤濕度值給val
        Serial.print(val);
        Serial.println(" val"); //序列埠顯示val值
        // ----- 將距離送到MQTT主題 -----
        MQTTClient.publish(MQTTPubTopic1, String((int)val).c_str());
        Serial.println("土壤濕度已推播到MQTT Broker");
        MQTTLastPublishTime = millis(); //更新最後傳輸時間
    }
}

```

程式撰寫步驟



```
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());  
}
```

```
//開始MQTT連線  
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);  
        }  
    }  
}
```



減碳競賽

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

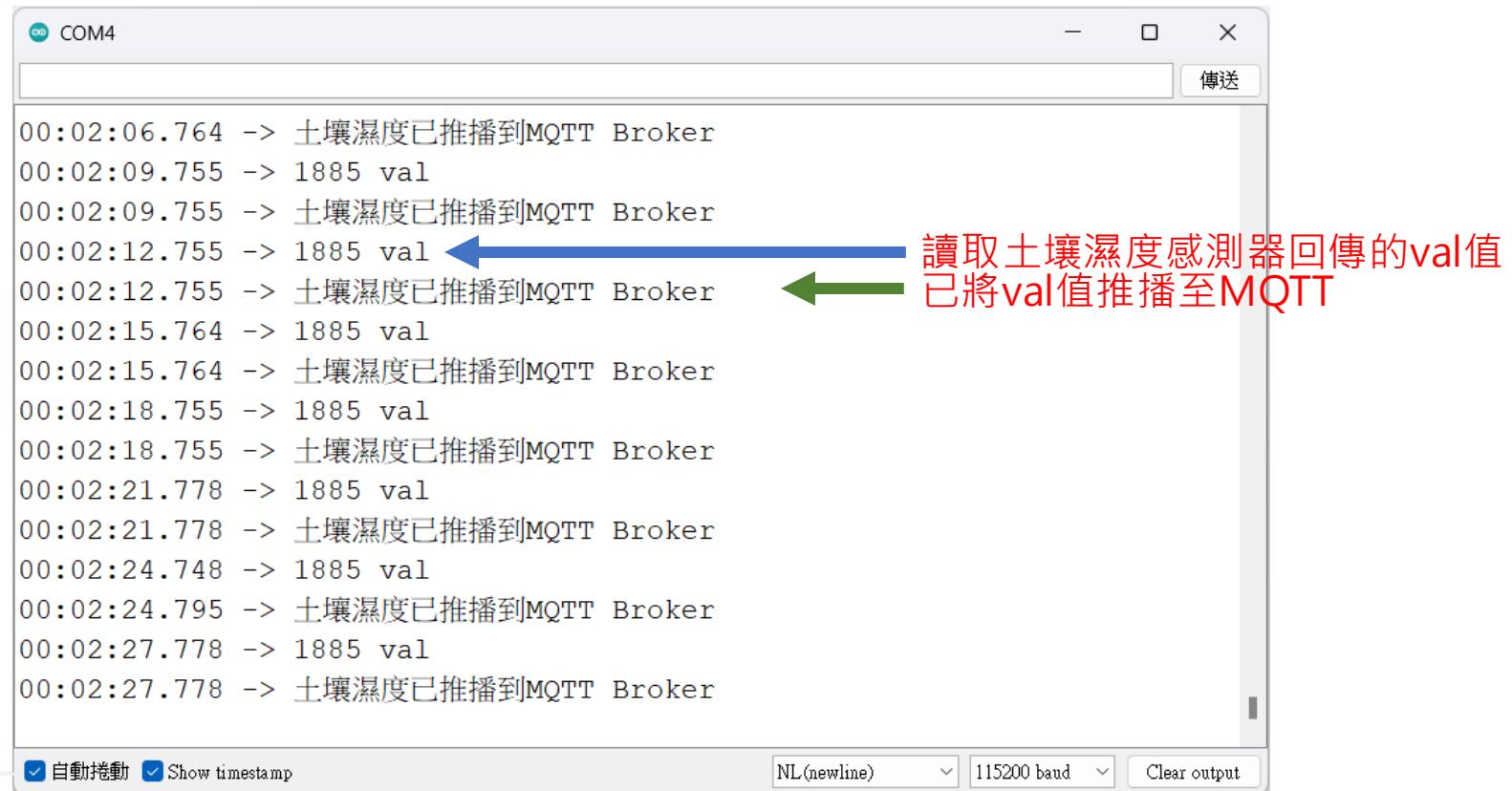


➤ 3.等待底下出現此字串即成功

```
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土壤濕度資訊



```

COM4
00:02:06.764 -> 土壤濕度已推播到MQTT Broker
00:02:09.755 -> 1885 val
00:02:09.755 -> 土壤濕度已推播到MQTT Broker
00:02:12.755 -> 1885 val ← 讀取土壤濕度感測器回傳的val值
00:02:12.755 -> 土壤濕度已推播到MQTT Broker ← 已將val值推播至MQTT
00:02:15.764 -> 1885 val
00:02:15.764 -> 土壤濕度已推播到MQTT Broker
00:02:18.755 -> 1885 val
00:02:18.755 -> 土壤濕度已推播到MQTT Broker
00:02:21.778 -> 1885 val
00:02:21.778 -> 土壤濕度已推播到MQTT Broker
00:02:24.748 -> 1885 val
00:02:24.795 -> 土壤濕度已推播到MQTT Broker
00:02:27.778 -> 1885 val
00:02:27.778 -> 土壤濕度已推播到MQTT Broker

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

```

查看資訊

- 土壤濕度偵測計的偵測原理是透過「導電度」，即透過金屬探針之間的導電度回傳類比訊號。



金屬探針

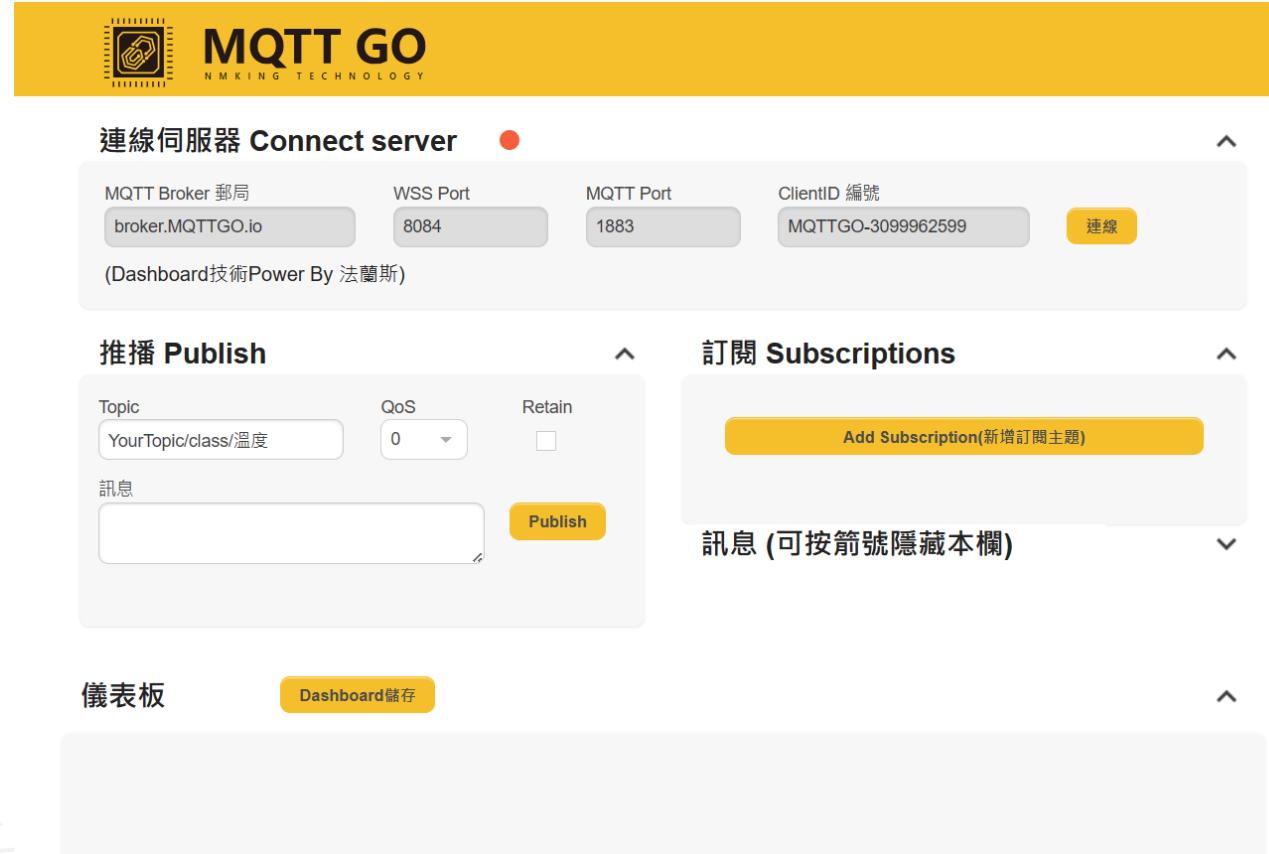
查看資訊

- ESP32類比訊號解析度為4096，即0~4095
- val值判斷方式



MQTT查看資訊

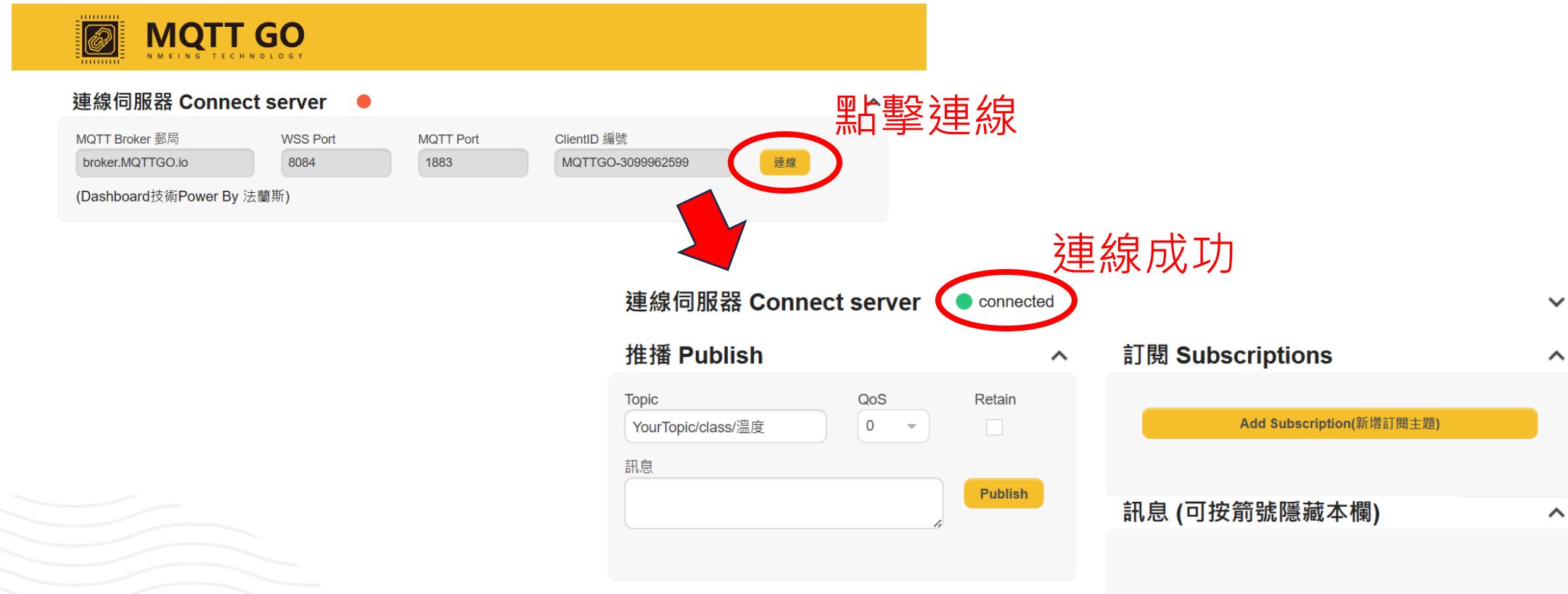
➤ 於瀏覽器開啟網站：<https://broker.mqttago.io/>



The screenshot displays 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: 'Push Publish' and 'Subscriptions'. The 'Push Publish' section allows users to enter a topic like 'YourTopic/class/溫度', set QoS to 0, and retain messages. It also includes 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. 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 sections: "Publish" and "Subscriptions". The "Publish" section has a "Topic" input field containing "YourTopic/class/溫度", a "QoS" dropdown set to "0", and a "Retain" checkbox. Below these are "訊息" (Message) input fields and a "Publish" button. The "Subscriptions" section has a "Add Subscription(新增訂閱主題)" button.



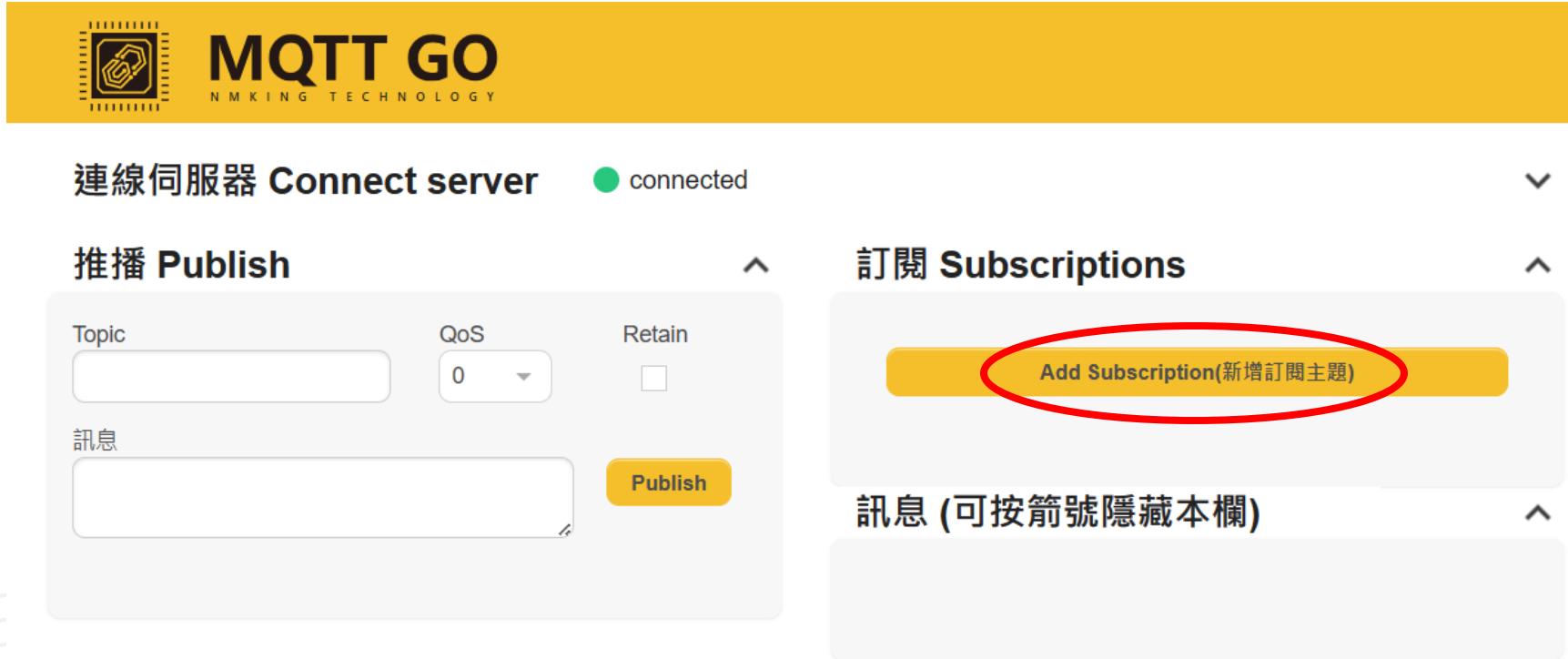
MQTT查看資訊

➤ 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/SoilWater";
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);
```

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



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/SoilWater X

訊息 (可按箭號隱藏本欄)

2024-03-12 Topic: YourTopic/class40... Qos: 0

09:59:25

1885

2024-03-12 Topic: YourTopic/class40... Qos: 0

09:59:19

1886

2024-03-12 Topic: YourTopic/class40... Qos: 0

09:59:12

1887

2024-03-12 Topic: YourTopic/class40... Qos: 0

09:59:05

1885

2024-03-12 Topic: YourTopic/class40... Qos: 0

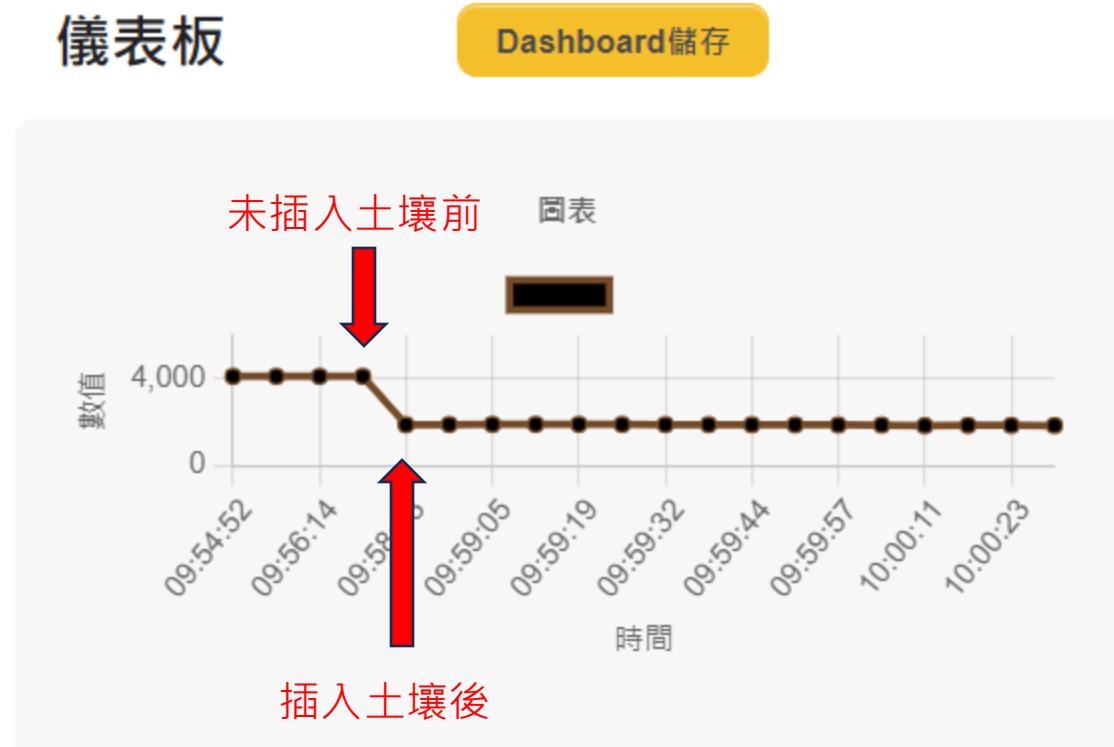
09:58:59



減碳績效

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

