

藏碳蘊漁

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

溶氧感測器

MQTT應用



目錄



材料

撰寫程式步驟

接線說明

寫入程式步驟

校正方法

序列埠查看資訊

MQTT查看資訊



減碳源頭

材料



ESP32

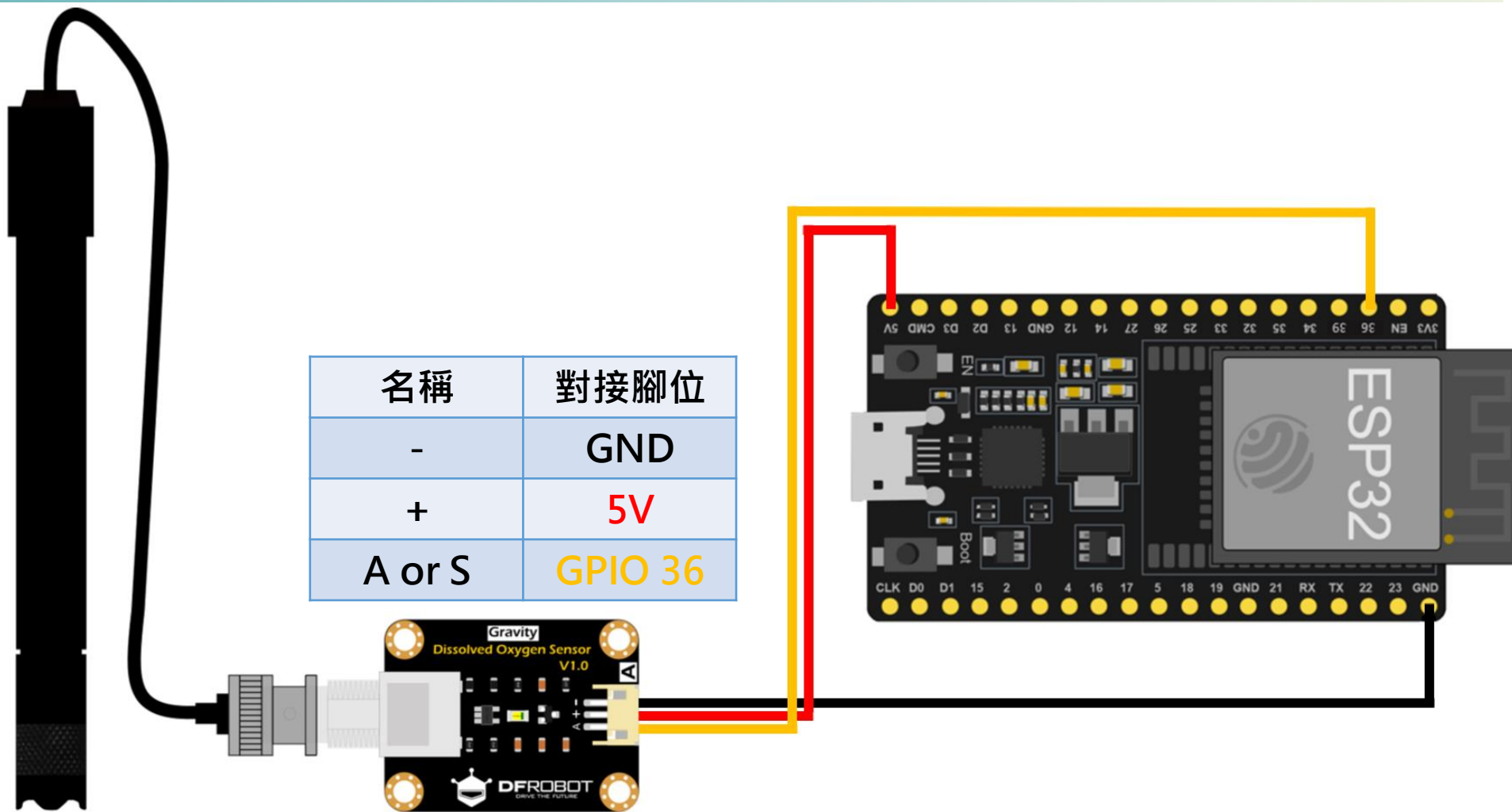


數據傳輸線 (MicroUSB)



溶解氧傳感器套件

接線說明



校正方法

- 請參考簡報「[1.溶氧感測器 校正方法.pptx](#)」

撰寫程式步驟

- 開啟記事本
「範例程式 溶氧感測器 MQTT.txt」
- 複製內容並貼上Arduino視窗中

撰寫程式步驟

```
#define DO_PIN 36 //腳位GPIO36
```

```
#define VREF 5000 //VREF (mv)
```

```
#define ADC_RES 4096 //ADC Resolution
```

```
//Single-point calibration Mode=0
```

```
//Two-point calibration Mode=1
```

```
#define TWO_POINT_CALIBRATION 0
```

使用單點校正法輸入「0」

使用兩點校正法輸入「1」

```
#define READ_TEMP (25) //修改量測溶氧水的溫度
```

撰寫程式步驟

//Single point calibration needs to be filled CAL1_V and CAL1_T

```
#define CAL1_V (1600) //mv
```

```
#define CAL1_T (25) //°C
```

//Two-point calibration needs to be filled CAL2_V and CAL2_T

//CAL1 High temperature point, CAL2 Low temperature point

```
#define CAL2_V (1300) //mv
```

```
#define CAL2_T (15) //°C
```

使用單點校正法請更改CAL1_V(校正電壓)、CAL1_T(校正的溫度)

CAL2_V及CAL2_T不須理會

使用兩點校正法請更改CAL1_V(第一杯水校正電壓)、CAL1_T(第一杯水的水溫)
更改CAL2_V(第二杯水校正電壓)、CAL2_T(第二杯水的水溫)

```
const uint16_t DO_Table[41] = {
```

```
    14460, 14220, 13820, 13440, 13090, 12740, 12420, 12110, 11810, 11530,
```

```
    11260, 11010, 10770, 10530, 10300, 10080, 9860, 9660, 9460, 9270,
```

```
    9080, 8900, 8730, 8570, 8410, 8250, 8110, 7960, 7820, 7690,
```

```
    7560, 7430, 7300, 7180, 7070, 6950, 6840, 6730, 6630, 6530, 6410};
```

撰寫程式步驟

```
uint8_t Temperaturet;
uint16_t ADC_Raw;
uint16_t ADC_Voltage;
uint16_t DO;

int16_t readDO(uint32_t voltage_mv, uint8_t temperature_c)
{
    #if TWO_POINT_CALIBRATION == 0
        uint16_t V_saturation = (uint32_t)CAL1_V + (uint32_t)35 * temperature_c - (uint32_t)CAL1_T * 35;
        return (voltage_mv * DO_Table[temperature_c] / V_saturation);
    #else
        uint16_t V_saturation = (int16_t)((int8_t)temperature_c - CAL2_T) * ((uint16_t)CAL1_V - CAL2_V) /
        ((uint8_t)CAL1_T - CAL2_T) + CAL2_V;
        return (voltage_mv * DO_Table[temperature_c] / V_saturation);
    #endif
}

double DOSensor;
```

撰寫程式步驟

➤ 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MQTT設定 -----
```

```
char* MQTTServer = "broker.mqttgo.io"; //免註冊MQTT伺服器
```

```
int MQTTPort = 1883; //MQTT Port
```

```
char* MQTTUser = ""; //不須帳號
```

```
char* MQTTPassword = ""; //不須密碼
```

```
//推播主題1:推播溶氧量(記得改Topic)
```

```
char* MQTTPubTopic1 = "YourTopic/class402/WaterD0";
```

```
long MQTTLastPublishTime; //此變數用來記錄推播時間
```

```
long MQTTPublishInterval = 10000; //每10秒推播一次
```

```
WiFiClient WifiClient;
```

```
PubSubClient MQTTClient(WifiClient);
```

自行更改路徑，
例如：TEST/class402/Water

程式撰寫步驟



```
void setup(){  
  Serial.begin(115200); //設定序列埠通訊  
  
  //開始WiFi連線  
  WifiConnecte();  
  
  //開始MQTT連線  
  MQTTConnecte();  
}
```

撰寫程式步驟



```
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10秒，則Publish溶氧量  
  if ((millis() - MQTTLastPublishTime) >= MQTTPublishInterval ) {  
    //讀取溶氧量  
    Temperaturet = (uint8_t)READ_TEMP;  
    ADC_Raw = analogRead(DO_PIN);  
    ADC_Voltage = uint32_t(VREF) * ADC_Raw / ADC_RES;  
    DOSensor = ((double)(readDO(ADC_Voltage, Temperaturet))/1000);  
    Serial.print("Temperature:\t" + String(Temperaturet) + "\t");  
    Serial.print("ADC RAW:\t" + String(ADC_Raw) + "\t");  
    Serial.print("ADC Voltage:\t" + String(ADC_Voltage) + "\t");  
    Serial.println("DO:\t" + String(readDO(ADC_Voltage, Temperaturet)) + "\t");  
    Serial.print("目前溶氧：");  
    Serial.println( String(DOSensor) + " mg/L");  
  }
```

```
  // ----- 將溶氧量送到MQTT主題 -----  
  MQTTClient.publish(MQTTPubTopic1, String(DOSensor).c_str());  
  Serial.println("溶氧量已推播到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());  
}
```

撰寫程式步驟

```
//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);
        }
    }
}
```

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

```
COM4
23:40:31.422 -> raw: 1712 Voltage(mv) 2089
23:40:32.401 -> raw: 1718 Voltage(mv) 2097
23:40:59.448 -> .....WiFi連線成功
23:41:21.149 -> IP Address:192.168.137.235
23:41:22.361 -> MQTT已連線
23:41:30.017 -> Temperature: 30 ADC RAW: 1712 ADC Voltage: 2089 DO: 7488
23:41:30.017 -> 目前溶氧: 7.49 mg/L
23:41:30.017 -> 溶氧量已推播到MQTT Broker
23:41:40.049 -> Temperature: 30 ADC RAW: 1738 ADC Voltage: 2121 DO: 7603
23:41:40.049 -> 目前溶氧: 7.60 mg/L
23:41:40.049 -> 溶氧量已推播到MQTT Broker
23:41:50.019 -> Temperature: 30 ADC RAW: 1744 ADC Voltage: 2128 DO: 7628
23:41:50.019 -> 目前溶氧: 7.63 mg/L
23:41:50.019 -> 溶氧量已推播到MQTT Broker
23:42:00.051 -> Temperature: 30 ADC RAW: 1744 ADC Voltage: 2128 DO: 7628
23:42:00.051 -> 目前溶氧: 7.63 mg/L
23:42:00.051 -> 溶氧量已推播到MQTT Broker
23:42:10.025 -> Temperature: 30 ADC RAW: 1744 ADC Voltage: 2128 DO: 7628
23:42:10.025 -> 目前溶氧: 7.63 mg/L
23:42:10.025 -> 溶氧量已推播到MQTT Broker
```

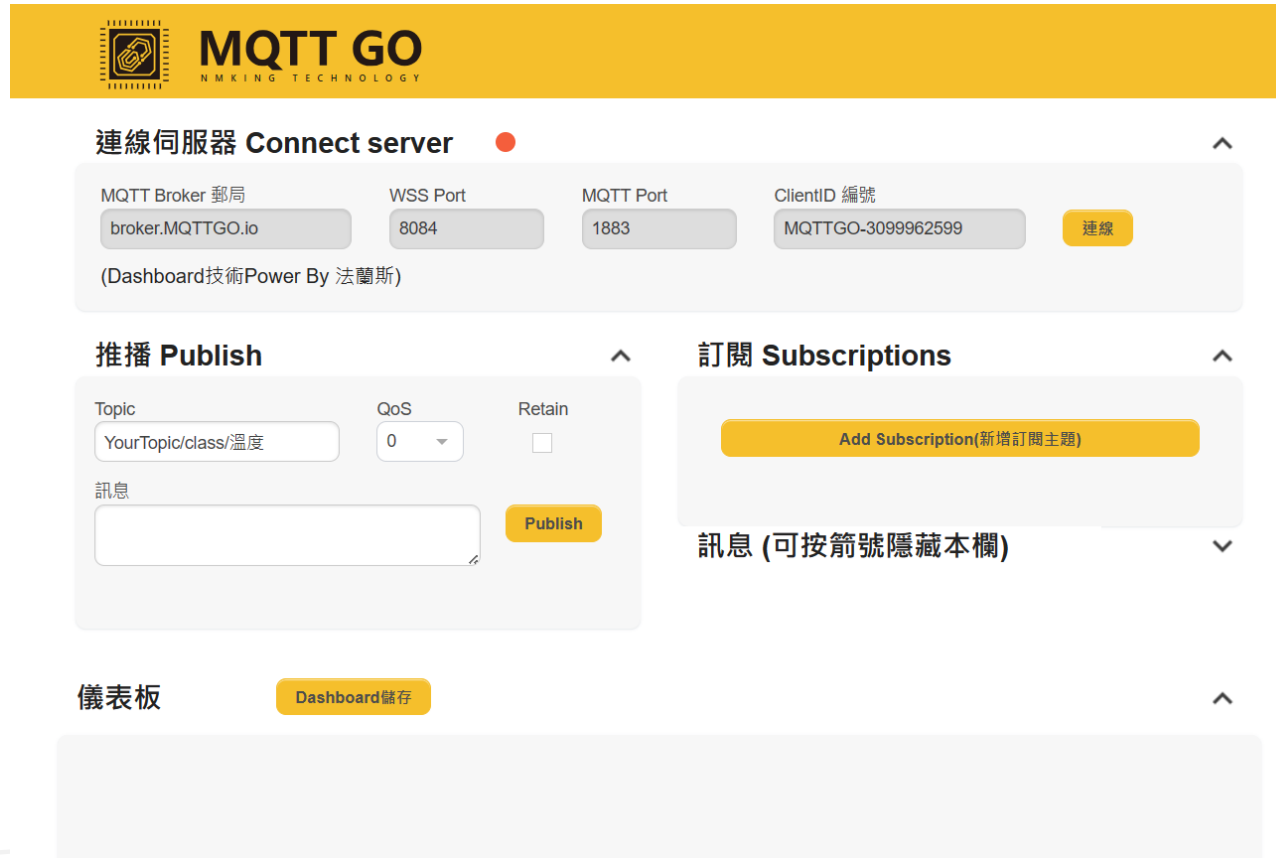
☒ 自動捲動 ☒ Show timestamp NL(newline) 115200 baud Clear output

Temperature : 溫度
 ADC RAW : 類比訊號原始值
 ADC Voltage : 電壓值
 DO : 溶氧量
 目前溶氧量 : $DO \text{ 數值} \div 1000$
 (單位mg/L)

已推播到MQTT

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 is a yellow header with the MQTT GO logo and the text "NM KING TECHNOLOGY". Below the header, there are three main sections: "連線伺服器 Connect server", "推播 Publish", and "訂閱 Subscriptions".

The "連線伺服器 Connect server" section has a red status indicator. It contains four input fields: "MQTT Broker 郵局" (broker.MQTTGO.io), "WSS Port" (8084), "MQTT Port" (1883), and "ClientID 編號" (MQTTGO-3099962599). A yellow "連線" button is to the right. Below these fields is the text "(Dashboard技術Power By 法蘭斯)".

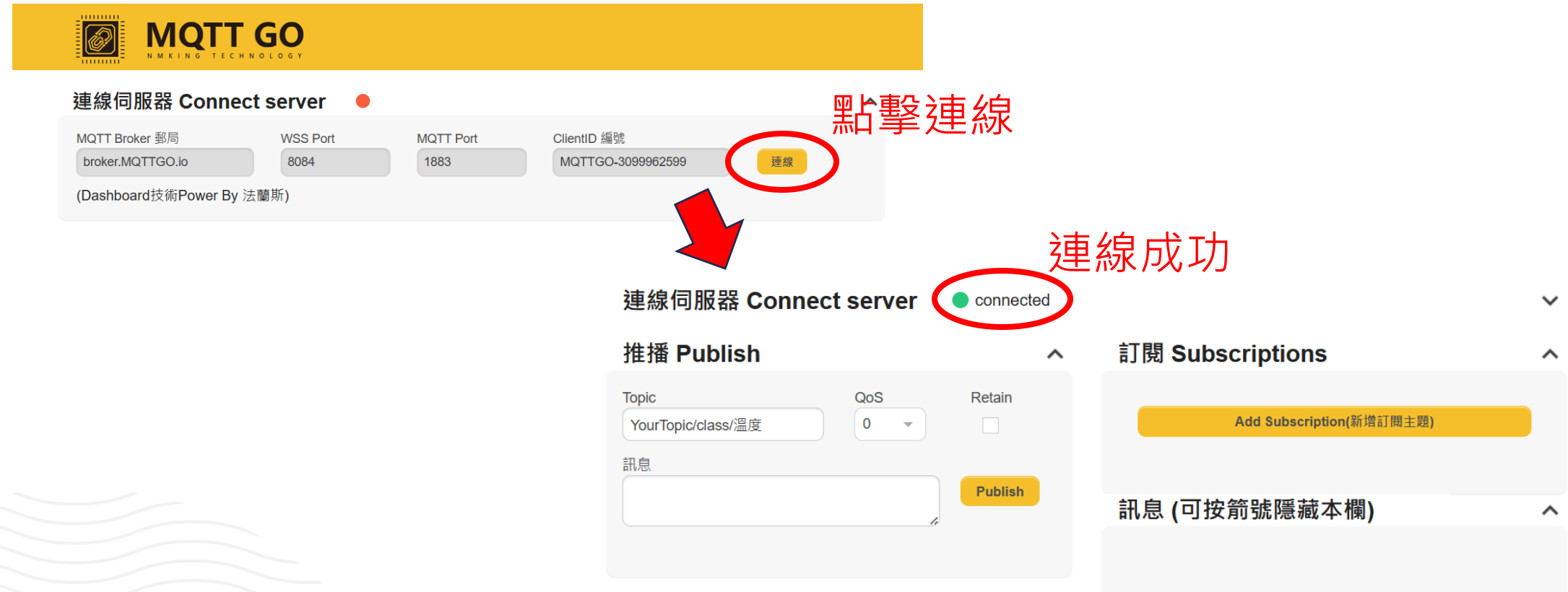
The "推播 Publish" section has a title "推播 Publish" and a collapse icon. It contains three input fields: "Topic" (YourTopic/class/溫度), "QoS" (0), and "Retain" (checkbox). Below these is a text area for "訊息" and a yellow "Publish" button.

The "訂閱 Subscriptions" section has a title "訂閱 Subscriptions" and a collapse icon. It contains a yellow button "Add Subscription(新增訂閱主題)". Below this is the text "訊息 (可按箭號隱藏本欄)".

At the bottom, there is a section titled "儀表板" with a yellow button "Dashboard儲存". Below this is a large empty box.

MQTT查看資訊

- 點擊連線，待燈號亮綠燈顯示connected即連線成功



The image shows the MQTT GO dashboard interface. At the top, there's a yellow header with the MQTT GO logo. Below it, the 'Connect server' section is highlighted. It contains fields for MQTT Broker (broker.MQTTGO.io), WSS Port (8084), MQTT Port (1883), and ClientID (MQTTGO-3099962599). A red circle highlights the '連線' (Connect) button, with a red arrow pointing to it and the text '點擊連線' (Click Connect). Below this, the 'Connect server' status is shown as 'connected' with a green dot, also circled in red with the text '連線成功' (Connection successful). The 'Publish' section is expanded, showing a topic field with 'YourTopic/class/溫度', a QoS dropdown set to 0, and a 'Publish' button. The 'Subscriptions' section is collapsed, showing an 'Add Subscription' button and a note about hiding the column.

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:推播溶氧量(記得改Topic) 複製  
char* MQTTPubTopic1 = "YourTopic/class402/WaterDO";  
long MQTTLastPublishTime;//此變數用來記錄推播時間  
long MQTTPublishInterval = 10000;//每10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



連線伺服器 **Connect server** ● connected ▼

推播 **Publish** ^

Topic

QoS

0 ▼

Retain

☐

訊息

Publish

訂閱 **Subscriptions** ^

Add Subscription(新增訂閱主題)

訊息 (可按箭號隱藏本欄) ^



藏碳蘊源

18

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 image shows a MQTT Subscribe configuration interface with the following elements and annotations:

- Color:** A blue color selection box.
- QoS:** A dropdown menu set to 2.
- Topic:** A text input field containing "YourTopic/class402/WaterDO", annotated with ① "貼上剛剛複製的Topic".
- Subscribe:** A yellow button, circled in red and annotated with ⑥ "完成後點擊Subscribe訂閱".
- 儀表板 (Dashboard):** A dropdown menu showing "甜甜圈圖" (Donut Chart), annotated with ② "選擇儀表板" (Choose dashboard) with an upward arrow.
- 名稱 (ID):** A text input field containing "溶氧量" (Dissolved Oxygen), annotated with ③ "更改名稱" (Change name).
- 數值區間 (min,max):** A text input field containing "0,10", annotated with ④ "調整區間0~10" (Adjust interval 0~10).
- 單位 (Unit):** A text input field containing "mg/L", annotated with ⑤ "更改單位:mg/L" (Change unit: mg/L).

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/WaterDO

X

訊息 (可按箭號隱藏本欄)

2024-04-24 23:41:50 Topic: YourTopic/class40... Qos: 0
7.63

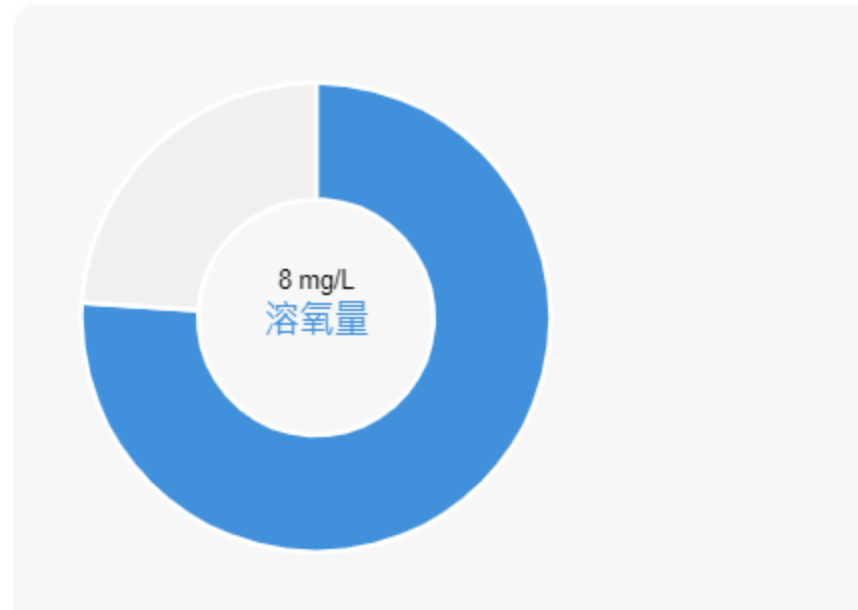
2024-04-24 23:41:40 Topic: YourTopic/class40... Qos: 0
7.60

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

儀表板

Dashboard儲存



儀表板為四捨五入結果



藏碳蘊漁

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

