

藏碳繭漁

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

TDS Sensor水質硬度傳感器

NB-IoT與MQTT連線



目錄



TDS介紹

程式撰寫說明

材料

寫入程式步驟

模組腳位說明

查看資訊

TDS感測器接線說明

MQTT查看資訊



藏碳蘊源

TDS介紹

- TDS中文名「溶解性總固體值」
- 用於監測水的硬度(混濁度)，通常採用"PPM"(partaper million，百萬分之一)為單位，即以每一百萬克含有多少碳酸鈣之量來決定。

TDS in PPM 使用範圍標準



材料



ESP32

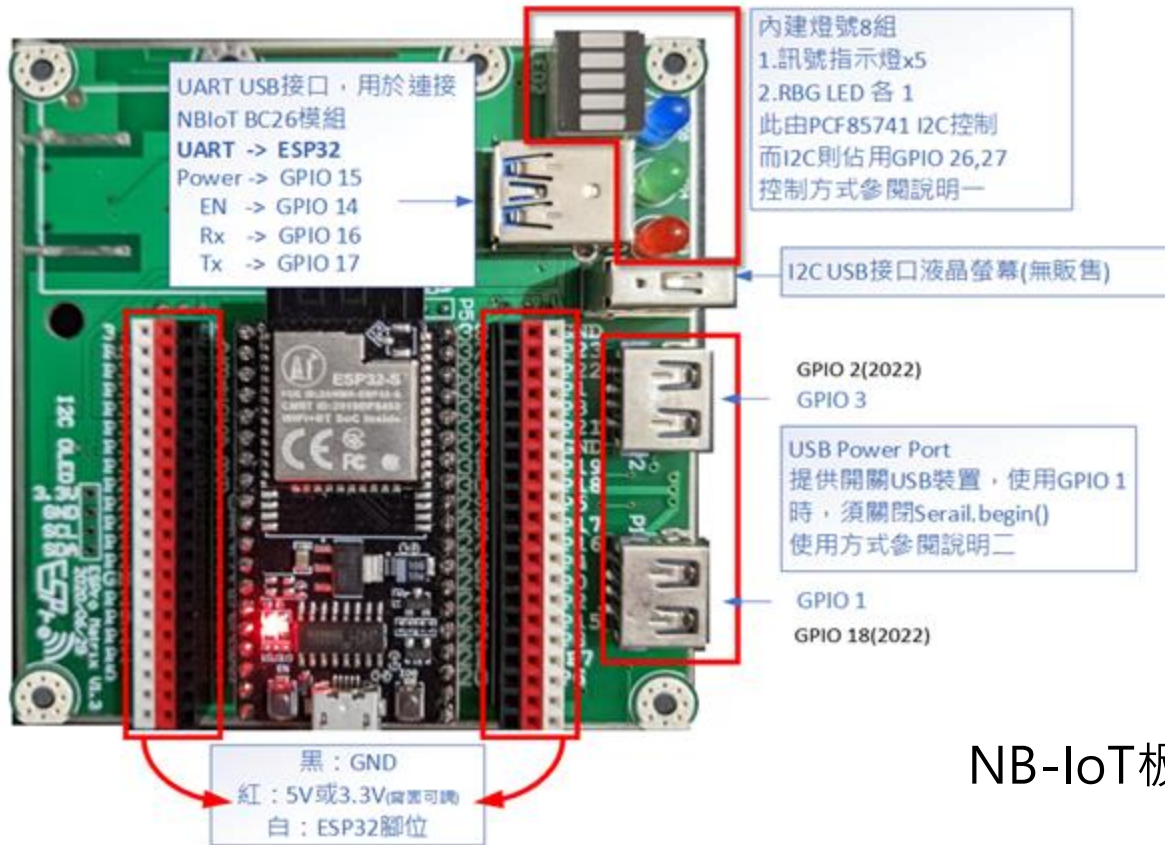


數據傳輸線 (MicroUSB)



TDS Meter及防水探頭

材料

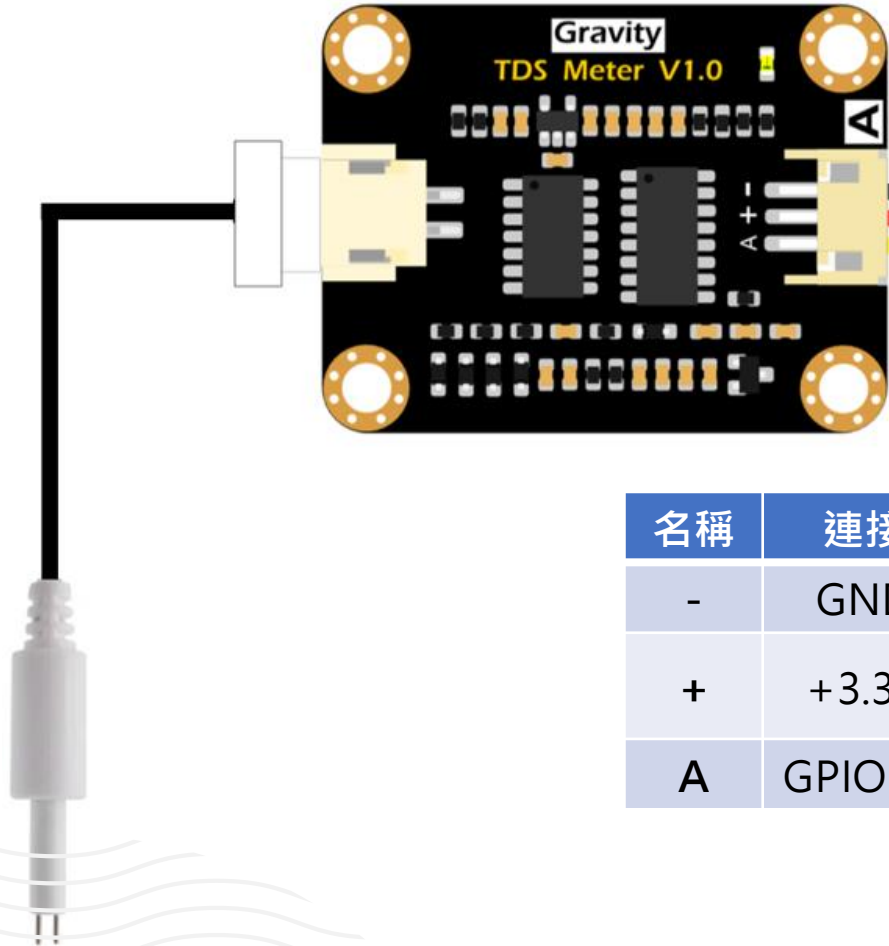


NB-IoT板

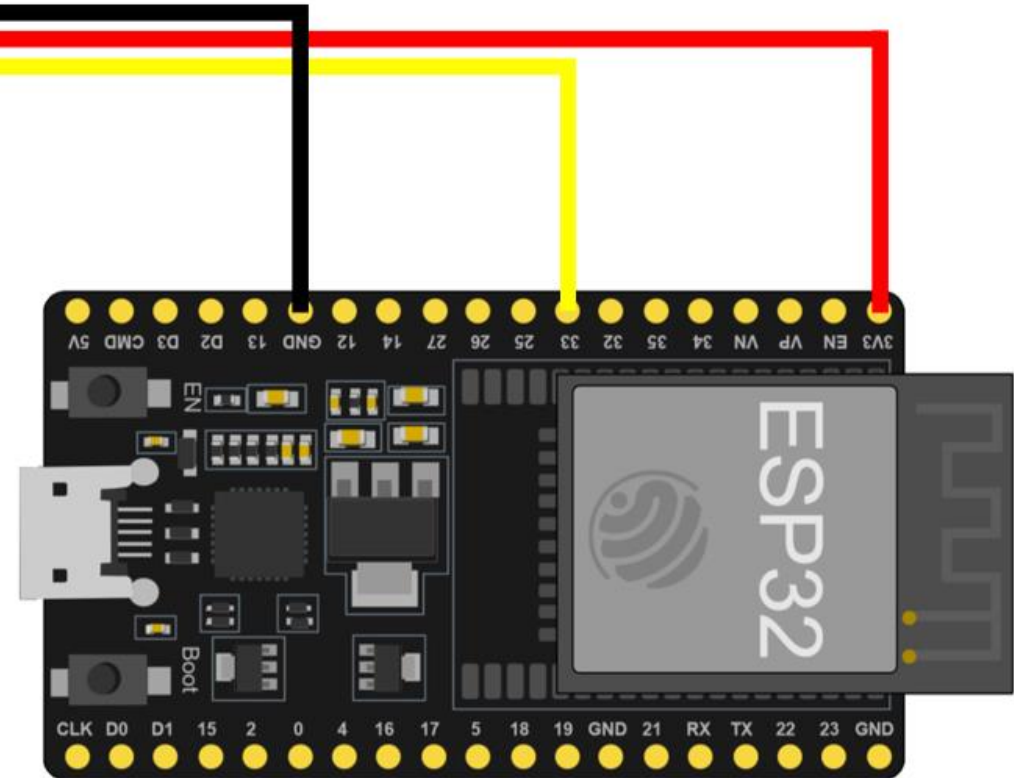
注意事項:

移除更換sim卡，請務必關閉電源，否則會短路冒煙損毀

TDS感測器接線說明

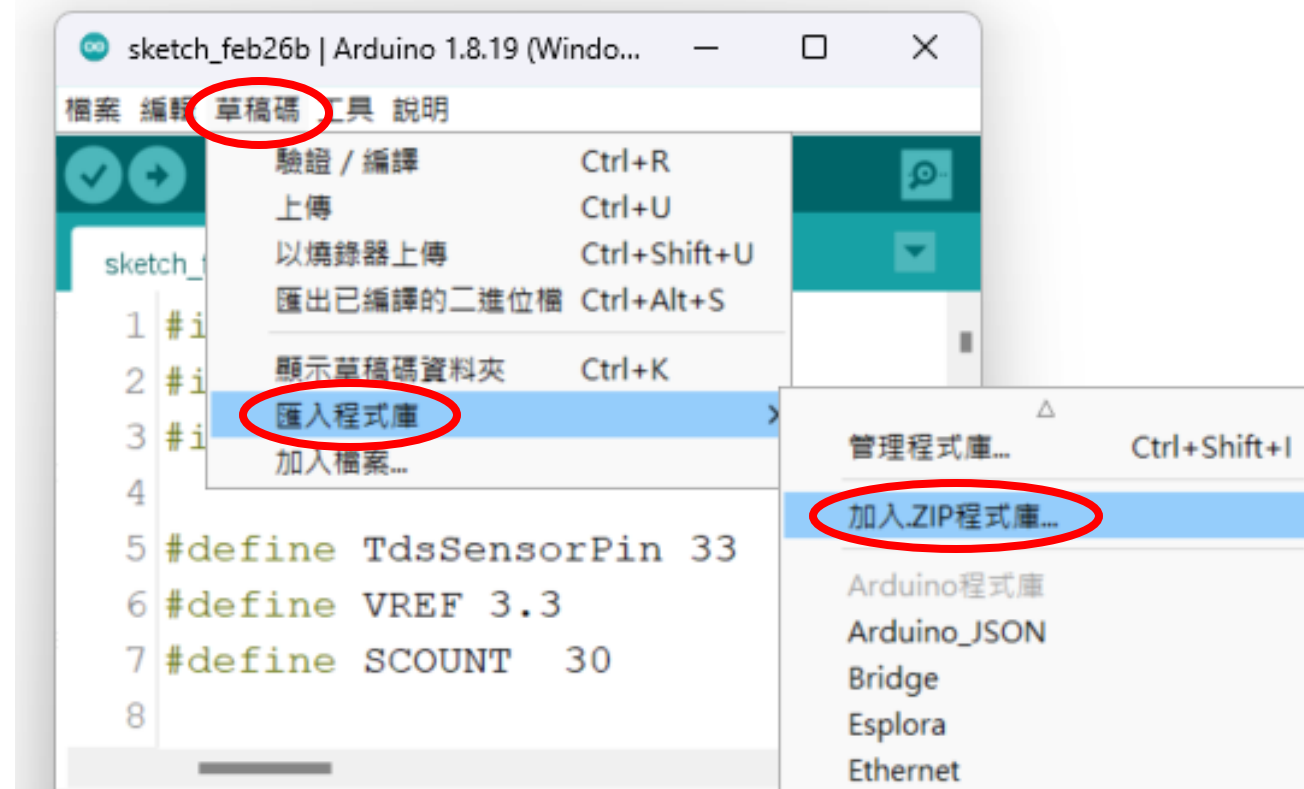


名稱	連接
-	GND
+	+3.3V
A	GPIO 33



安裝函式庫

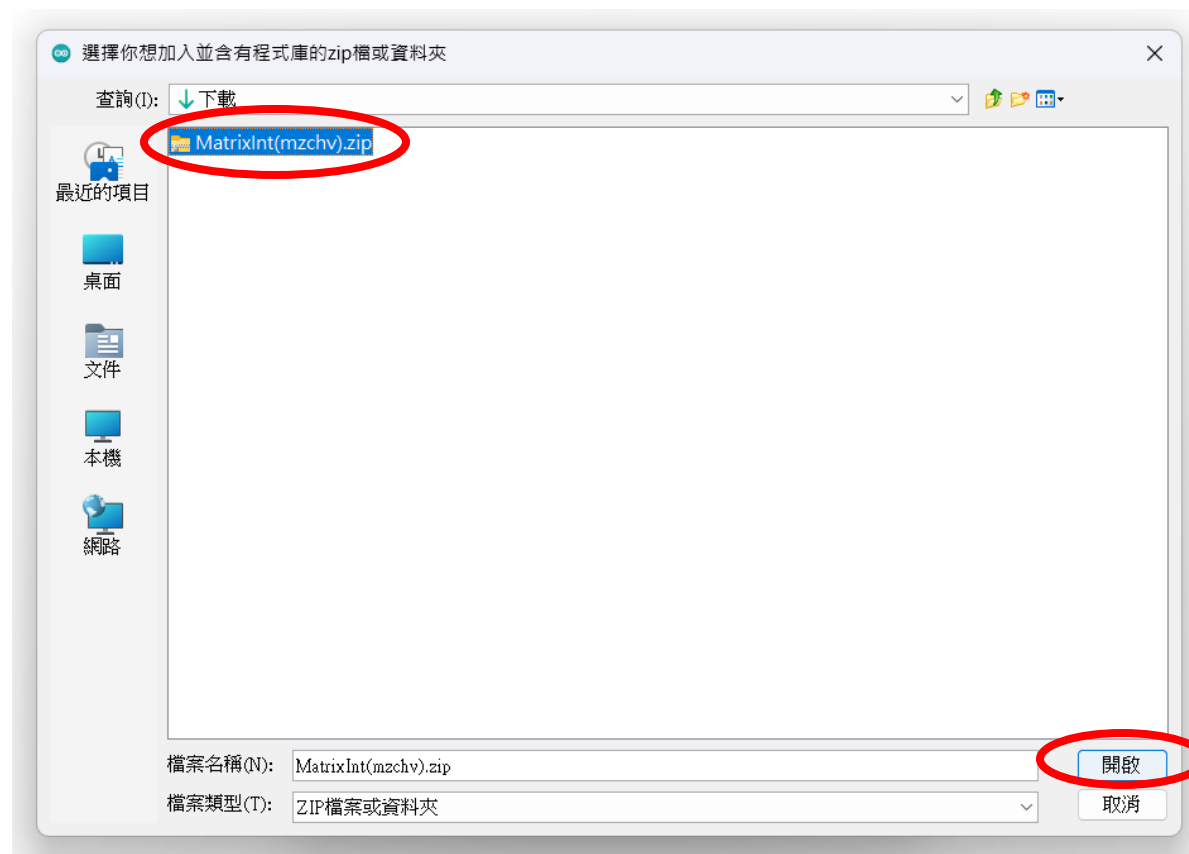
- 至網站下載ZIP檔 網址：<https://twgo.io/mzchv>
- 開啟Arduino IDE
- 草稿碼>匯入程式庫
 > 加入.ZIP程式庫



安裝函式庫

- 選擇剛剛下載的MatrixInt(mzchv).zip
- 點擊開啟
- 匯入成功

已加入程式庫。請檢查選單「匯入程式庫」



程式撰寫步驟(1/6)

- 開啟「範例程式TDS Sensor NB-IoT.txt」
- 複製內容並貼上Arduino視窗中

程式撰寫步驟(2/6)

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

```
#define TdsSensorPin 33 //TDS感測器接入腳位GPIO 33
#define VREF 3.3 // analog reference voltage(Volt) of the ADC
#define SCOUNT 30 // sum of sample point
```

```
int analogBuffer[SCOUNT]; // store the analog value in the array, read from ADC
int analogBufferTemp[SCOUNT];
int analogBufferIndex = 0;
int copyIndex = 0;
long WNB303Timeout = 10000; //返回0=Timeout
int WNB303ResetPIN = 14; // 設定 WNB303 重置腳位為輸出腳位
int WNB303PowerPIN = 15; // 設定 WNB303 電源控制腳位為輸出腳位
```

```
float averageVoltage = 0;
float tdsValue = 0;
float temperature = 25; // 設定當前環境溫度 (計算PPM需使用溫度資訊)
// 根據當下氣溫 自行更改氣溫數據(溫度單位：攝氏)
```

程式撰寫步驟(3/6)

```
// median filtering algorithm
int getMedianNum(int bArray[], int iFilterLen){
    int bTab[iFilterLen];
    for (byte i = 0; i<iFilterLen; i++)
        bTab[i] = bArray[i];
    int i, j, bTemp;
    for (j = 0; j < iFilterLen - 1; j++) {
        for (i = 0; i < iFilterLen - j - 1; i++) {
            if (bTab[i] > bTab[i + 1]) {
                bTemp = bTab[i];
                bTab[i] = bTab[i + 1];
                bTab[i + 1] = bTemp;
            }
        }
    }
    if ((iFilterLen & 1) > 0){
        bTemp = bTab[(iFilterLen - 1) / 2];
    }
    else {
        bTemp = (bTab[iFilterLen / 2] + bTab[iFilterLen / 2 - 1]) / 2;
    }
    return bTemp;
}
```

➤ 更改MQTTPubTopic，避免與他人重複

// ----- 以下修改成你MQTT設定 -----

String MQTTServer = "mqttgo.io";//免註冊MQTT伺服器

String MQTTPort = "1883";//MQTT Port

String MQTTUser = "";//不須帳密

String MQTTPassword = "";//不須帳密 自行更改路徑，

//推播主題1:推播PPM(記得改Topic) 例如：TEST/class402/TDSPPM

String MQTTPubTopic = "YourTopic/class/TDSPPM";

long MQTTLastPublishTime = 0;//此變數用來記錄推播時間

long MQTTPublishInterval = 5000;//每5秒推撥一次

程式撰寫步驟(4/6)

```
void setup() {
  Serial.begin(115200); // 設定序列埠監控視窗的鮑率
  Serial2.begin(115200); // 設定 WNB303 的鮑率
  Wire.begin(26, 27); // Matrix I2C
  for (int i = 0; i <= 7; ++i) MatrixInt(i, 0); // 關閉所有燈號
  pinMode(TdsSensorPin, INPUT);
  pinMode(WNB303ResetPIN, OUTPUT); // 設定 WNB303 電源控制腳位為輸出腳位
  pinMode(WNB303PowerPIN, OUTPUT); // 設定 WNB303 重置腳位為輸出腳位
  delay(1000);

  Serial.println("\r\n\r\n\r\n===== \r\nSystem Starting....");
  String result = "";
  // 開啟 Modem 的電源，開啟關閉之間須間隔10秒(延時電路)
  WNB303Restart(10);
  // 檢查網路註冊狀態，5分鐘內無法註冊，則重新啟動WNB303
  int tryCount = 0;
  while (1) {
    boolean resultb = WNB303CheckReg();
    Serial.print("WNB303CheckNetReg result="); Serial.println(resultb);
    if (resultb == false) {
      Serial.println("時間內無法註冊網路(" + String(tryCount) + ")");
      delay(10000);
    }
    else break;
    if (tryCount++ >= 30) { // 無法註冊已經超過5分鐘
      tryCount = 0;
      WNB303Restart(30);
    }
  }
  // 檢查網路訊號品質
  result = WNB303CheckRSSI();
  Serial.print("WNB303CheckRSSI result="); Serial.println(result);
  delay(10000);
}
```

```
void loop() {
  ReadPPM();
  // 檢查時間，傳輸步驟：檢查網路、讀取距離、MQTT連線、推播、關閉
  if ((millis() - MQTTLastPublishTime) >= MQTTPublishInterval) {
    MatrixInt(7, 1);
    // 沒有網路，則重新啟動WNB303
    String result = "";
    // 檢查網路狀態
    if (!WNB303CheckReg()) {
      WNB303Restart(60); // 60秒後重新檢測網路
      return;
    } else result = WNB303CheckRSSI();

    // 連線mqtt
    result = mqttConnect(MQTTServer, MQTTPort, MQTTUser, MQTTPassword);
    if (result == "OK") Serial.println("MQTT Connected");
    ReadPPM();
    byte tds1 = 0;
    tds1 = tdsValue, 0;
    String payload = String((int)tds1);
    // 推播訊息
    result = mqttPublish(MQTTPubTopic, "0", "0", "0", payload);
    if (result == "OK") Serial.println("Data:" + payload + "\" Published to " + MQTTPubTopic);
    // 關閉mqtt
    result = mqttDisconnect();
    if (result == "OK") Serial.println("MQTT Disconnected");
    MQTTLastPublishTime = millis(); // 更新最後傳輸時間
    MatrixInt(7, 0);
  }
  delay(1000);
}
```

程式撰寫步驟(5/6)



```
void ReadPPM() {
    static unsigned long analogSampleTimepoint = millis();
    if(millis()-analogSampleTimepoint > 400){ //every 40 milliseconds,read the analog value from the ADC
        analogSampleTimepoint = millis();
        analogBuffer[analogBufferIndex] = analogRead(TdsSensorPin); //read the analog value and store into the buffer
        analogBufferIndex++;
        if(analogBufferIndex == SCOUNT){
            analogBufferIndex = 0;
        }
    }

    static unsigned long printTimepoint = millis();
    if(millis()-printTimepoint > 8000){
        printTimepoint = millis();
        for(copyIndex=0; copyIndex<SCOUNT; copyIndex++){
            analogBufferTemp[copyIndex] = analogBuffer[copyIndex];

            // read the analog value more stable by the median filtering algorithm, and convert to voltage value
            averageVoltage = getMedianNum(analogBufferTemp,SCOUNT) * (float)VREF / 4096.0;

            //temperature compensation formula: fFinalResult(25°C) = fFinalResult(current)/(1.0+0.02*(fTP-25.0));
            float compensationCoefficient = 1.0+0.02*(temperature-25.0);
            //temperature compensation
            float compensationVoltage=averageVoltage/compensationCoefficient;

            //convert voltage value to tds value
            tdsValue=(133.42*compensationVoltage*compensationVoltage*compensationVoltage -
            255.86*compensationVoltage*compensationVoltage + 857.39*compensationVoltage)*0.5;

            //Serial.print("voltage:");
            //Serial.print(averageVoltage,2);
            //Serial.print("V ");
            Serial.print("TDS Value:");
            Serial.print(tdsValue,0);
            Serial.println("ppm");
        }
    }
}
```

```
//連線mqtt
String mqttConnect(String Host, String Port, String MQTTUser, String MQTTPassword) {
    //1.取得IP
    String result = "";
    String IP = AT2WNB303("AT+EDNS=\\" + Host + "\\", "+EDNS:");
    IP.trim(); IP = split( IP, ':', 1);
    if (IP.length() <= 7) return "Error: Can't Get IP";
    //2.建立連線
    result = AT2WNB303("AT+EMQNEW=\\" + IP + "\\", Port + "\\", "+ 60000 + \", " + 1024, "+EMQNEW:");
    String mqttid = split( result, ':', 1);
    //Serial.print("MQTTCreate result="); Serial.println(mqttid);
    //以亂數為ClientID
    String MQTTClientid = "NBloT-" + String(random(1000000, 9999999));
    result = AT2WNB303("AT+EMQCON=" + mqttid + ",3,\\" + MQTTClientid + "\\", "+ 60000 + ",0,0,\\" + MQTTUser + "\\", \\" +
    MQTTPassword + "\\", "OK");
    //Serial.print("MQTTConnect result="); Serial.println(result);
    return result;
}

//推播訊息
String mqttPublish(String Topic, String QoS, String retained, String dup, String Payload) {
    String result = "";
    String hexPayload = Str2Hex(Payload);
    String lenHexPayload = String(hexPayload.length());
    result = AT2WNB303("AT+EMQ PUB=0,\" + Topic + \",\" + QoS + \",\" + retained + \",\" + dup + \",\" + lenHexPayload + \",\" + hexPayload, "OK");
    return result;
}

//關閉mqtt連線
String mqttDisconnect() {
    String result = "";
    result = AT2WNB303("AT+EMQDISCON=0", "OK");
    return result;
}
```

程式撰寫步驟(6/6)



```
//檢查網路註冊狀態 return true or false
boolean WNB303CheckReg() {
    String res ut = "";
    boolean CERE G = false;
    result = AT2WNB303("AT+CEREG?", "+CEREG:");
    if (result.indexOf("0,1") >= 0 || result.indexOf("1,1") >= 0) { //註冊成功
        Matrix(5, 0); //完經 燈
        Matrix(6, 1); //完綠 燈
        CERE G = true;
    }
    else {
        Matrix(5, 1); //完紅 燈
        Matrix(6, 0); //完綠 燈
        CERE G = false;
    }
    return CERE G;
}

//檢查網路訊號品質 return RSSI，當RSSI=0或99代表沒訊號
int WNB303CheckRSSI() {
    String res ut = "";
    int RSSI = 0;
    result = AT2WNB303("AT+CEQ?", "+CEQ:");
    //處理 +CEQ
    if (result == "-1") {
        int CEQ = split(result, ',', 1); // 0 to int()
        RSSI = CEQ - 11;
    }
    if (RSSI == -11 || RSSI == 210) RSSI = 0;
    MatrixLEDs(RSSI);
    return RSSI;
}

//重新啟動WNB303(單位為秒)
void WNB303Restart(int delayTime) {
    AT2WNB303("POWEROFF", "");
    delay(5000);
    AT2WNB303("POWERON", "");
    delay(delayTime * 1000);
}

//HTTP GET
String HTTPGET(String Protocol, String Host, String port, String Uri) {
    //例如: http://x.x.x:update.asp?key=xxx&field=0
    //拆解成Protcol="http" host="x.x.x" port="80" Uri="/update.asp_key=xxx&field=0"
    //1.轉換網址
    String res ut = "";
    String IP = AT2WNB303("AT+EDNS=" + Host + "\n", "+EDNS:");
    IP.trim(); IP = split(IP, '\n', 1);
    if (IP.length() <= 7) return "Error: Can't Get IP";
    //2.建立連線
    String PIP = Protocol + "/" + IP + "/" + port + "/";
    int PIPlen = PIP.length();
    //Serial.println("PIP=" + PIP + " len=" + String(PIPlen));
    result = AT2WNB303("AT+HTTPCREAT=" + String(PIPlen) + ",\n" + PIP + "\n", "+BHTTPCREAT");
    result.trim(); String clientid = split(result, '\n', 1);
    if (result == "") return "Error: Can't Create Connection";
    else {
        //3.開啟連線
        result = AT2WNB303("AT+HTTPCONN=" + clientid, "OK");
        result.trim();
        if (result == "OK") result = "Error: Can't Connect to Server";
        //4.組成網址並傳送
        int i = 0;
        String PostCom mand[30];
        PostCom mand[i++] = clientid + ",1," + Uri.length() + ", " + Uri + ",0," + contType.length() + ", " + contType + ";";
        //Serial.println("command=" + PostCom mand[i-1]);
        String hexData = Str2Hex(Data);
        int lenHexData = hexData.length();
        PostCom mand[i++] = String(lenHexData) + ";";
        //Serial.println("command2=" + PostCom mand[i-1]);
        int num = 30;
        for (int n = 0; n < lenHexData; n = n + 1) {
            String DataSend = "";
            if (n + num < lenHexData) {
                //切割字串
                DataSend = hexData.substring(n, n + num);
            } else if (lenHexData % num > 0) {
                int remainder = lenHexData % num;
                //切割字串
                DataSend = hexData.substring(n, n + remainder);
            }
            PostCom mand[i++] = DataSend;
            //Serial.println("command3=" + PostCom mand[i-1]);
        }
        //求出 每條命令長度
        int totalLenHexData = 0;
        for (int j = 0; j < i-1; j++) {
            totalLenHexData += totalLenHexData + PosCom mand[j].length();
            PostCom mand[j] = String(PostCom mand[j].length()) + "," + PostCom mand[j];
        }
        //完成命令組合
        for (int i = 0; i < i-1; i++) {
            if (i == 1) PostCom mand[i] = "AT+BHTTPSEND=1" + String(totalLenHexData) + "," + PosCom mand[i];
            else if (i == i-1) PostCom mand[i] = "AT+BHTTPSEND=0" + String(totalLenHexData) + "," + PostCom mand[i];
            else PostCom mand[i] = "AT+BHTTPSEND=1" + String(totalLenHexData) + "," + PostCom mand[i];
            result = AT2WNB303("PostCom mand[i]", "OK");
            //Serial.println("POST" + String(i) + " " + result);
        }
    }
    //關閉連線
    delay(1000);
    AT2WNB303("AT+BHTTPDISCONN=" + clientid, "OK");
    //Serial.println(result);
    delay(1000);
    AT2WNB303("AT+BHTTPDESTROY=" + clientid, "OK");
    //Serial.println(result);
    delay(1000);
    return result;
}

//將訊息傳到WNB303，並讀取回傳訊息 0代表timeout
String AT2WNB303String ATdata, String StatWith() {
    Serial.println("你的命令是:" + ATdata);
    if (ATdata.length() > 0) { //送出AT命令
        ATdata.trim();
        String comm and = ATdata;
        comm and.toLowerCase();
    }
}
```

```
//HTTP POST
String HTTPPOST(String Protocol, String Host, String port, String Uri, String contentType, String Data) {
    //1.轉換網址IP
    String res ut = "";
    String IP = AT2WNB303("AT+EDNS=" + Host + "\n", "+EDNS:");
    IP.trim(); IP = split(IP, '\n', 1);
    if (IP.length() <= 7) return "Error: Can't Get IP";
    //2.建立連線
    String PIP = Protocol + "/" + IP + "/" + port + "/";
    int PIPlen = PIP.length();
    //Serial.println("PIP=" + PIP + " len=" + String(PIPlen));
    result = AT2WNB303("AT+HTTPCREAT=" + String(PIPlen) + ",\n" + PIP + "\n", "+BHTTPCREAT");
    result.trim(); String clientid = split(result, '\n', 1);
    if (result == "") return "Error: Can't Create Connection";
    else {
        //3.開啟連線
        result = AT2WNB303("AT+HTTPCONN=" + clientid, "OK");
        result.trim();
        if (result == "OK") result = "Error: Can't Connect to Server";
        //4.組成網址並傳送
        int i = 0;
        String PostCom mand[30];
        PostCom mand[i++] = clientid + ",1," + Uri.length() + ", " + Uri + ",0," + contType.length() + ", " + contType + ";";
        //Serial.println("command=" + PostCom mand[i-1]);
        String hexData = Str2Hex(Data);
        int lenHexData = hexData.length();
        PostCom mand[i++] = String(lenHexData) + ";";
        //Serial.println("command2=" + PostCom mand[i-1]);
        int num = 30;
        for (int n = 0; n < lenHexData; n = n + 1) {
            String DataSend = "";
            if (n + num < lenHexData) {
                //切割字串
                DataSend = hexData.substring(n, n + num);
            } else if (lenHexData % num > 0) {
                int remainder = lenHexData % num;
                //切割字串
                DataSend = hexData.substring(n, n + remainder);
            }
            PostCom mand[i++] = DataSend;
            //Serial.println("command3=" + PostCom mand[i-1]);
        }
        //求出 每條命令長度
        int totalLenHexData = 0;
        for (int j = 0; j < i-1; j++) {
            totalLenHexData += totalLenHexData + PosCom mand[j].length();
            PostCom mand[j] = String(PostCom mand[j].length()) + "," + PostCom mand[j];
        }
        //完成命令組合
        for (int i = 0; i < i-1; i++) {
            if (i == 1) PostCom mand[i] = "AT+BHTTPSEND=1" + String(totalLenHexData) + "," + PosCom mand[i];
            else if (i == i-1) PostCom mand[i] = "AT+BHTTPSEND=0" + String(totalLenHexData) + "," + PostCom mand[i];
            else PostCom mand[i] = "AT+BHTTPSEND=1" + String(totalLenHexData) + "," + PostCom mand[i];
            result = AT2WNB303("PostCom mand[i]", "OK");
            //Serial.println("POST" + String(i) + " " + result);
        }
    }
    //關閉連線
    delay(1000);
    AT2WNB303("AT+BHTTPDISCONN=" + clientid, "OK");
    //Serial.println(result);
    delay(1000);
    AT2WNB303("AT+BHTTPDESTROY=" + clientid, "OK");
    //Serial.println(result);
    delay(1000);
    return result;
}
```

```
//將訊息傳到WNB303，並讀取回傳訊息 0代表timeout
String AT2WNB303String ATdata, String StatWith() {
    Serial.println("你的命令是:" + ATdata);
    if (ATdata.length() > 0) { //送出AT命令
        ATdata.trim();
        String comm and = ATdata;
        comm and.toLowerCase();
    }
}
```

```
if (comm and == "RESET") { //重置 WNB303
    digitalWrite(WNB303ResetPIN, HIGH);
    delay(10000);
    digitalWrite(WNB303ResetPIN, LOW);
    return "RESET OK";
}
else if (command == "POWERON") { //關閉 WNB303 的電源
    digitalWrite(WNB303PowerPIN, HIGH);
    return "POWERON OK";
}
else if (command == "POWEROFF") { //關閉 WNB303 的電源
    digitalWrite(WNB303PowerPIN, LOW);
    return "POWEROFF OK";
}
else { //送出AT命令
    Serial.println(ATdata);
}
}
else return "";
String res ut = "";
//等候回應資料
long StartTime = millis();
while (1) {
    result = "";
    while (Serial.available()) { //WNB303有資料回傳
        char c = Serial.read(); //從WNB303讀取一個位元組
        result += c;
        if (c == '\n') break;
    }
    result.trim();
    if (result.startsWith(Str2Hex)) break; //結尾OK返回
    if (lenHexData - StartTime >= WNB303Timeout) { //Time out返回
        result = "0";
        break;
    }
}
return result;
}
```

```
//字串轉HEX
String Str2Hex(String msg) {
    String a = "";
    for (int i = 0; i < msg.length(); i++) {
        a += String(msg.charAt(i), HEX);
    }
    return a;
}
```

```
//HEX轉字串
String Hex2Str(String msg) {
    char input[msg.length() + 1];
    msg.toCharArray(input, msg.length() + 1);
    char c[sizeof input];
    String a = "";
    for (int i = 0; i < sizeof(input); i += 2) {
        char tem p[3];
        temp[0] = input[i];
        temp[1] = input[i + 1];
        int val = ASCIIHexToInt(temp[0] * 16 + ASCIIHexToInt(temp[1]));
        c[i] = toascii(val);
        a += String(c[i]);
    }
    return a;
}
```

```
//ASCII轉INT
int ASCIIHexToInt(char c) {
    int ret = 0;
    if ((c >= '0') && (c <= '9')) ret = (ret << 4) + c - '0';
    else ret = (ret << 4) + toupper(c) - 'A' + 10;
    return ret;
}
```

```
//split拆解 範例：String at=split("aabbcc","");
String split(String data, char separator, int index) {
    int found = 0;
    int strIndex[] = {0, -1};
    int maxIndex = data.length() - 1;
    for (int i = 0; i <= maxIndex && found <= index; i++) {
        if (data.charAt(i) == separator || i == maxIndex) {
            found++;
            strIndex[i] = strIndex[i] + 1;
            strIndex[i] = (i == maxIndex) ? i + 1 : i;
        }
    }
    return found > index ? data.substring(strIndex[0], strIndex[i]) : "";
```



藏碳蘊源

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

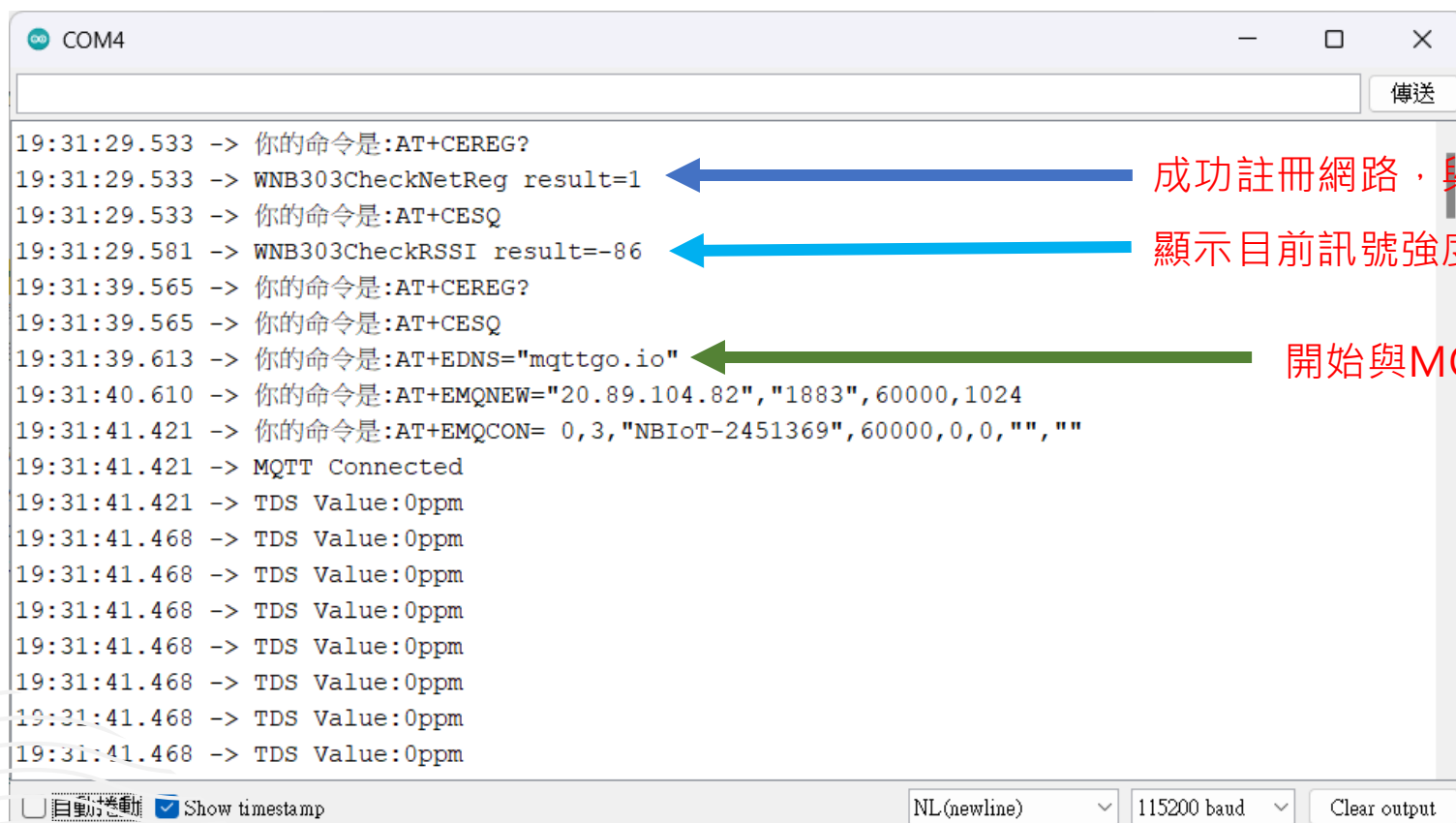


➤ 3.等待底下出現此字串即成功

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

查看資訊

➤ 開啟序列埠視窗，查看連線狀態



```
COM4
19:31:29.533 -> 你的命令是:AT+CEREG?
19:31:29.533 -> WNB303CheckNetReg result=1
19:31:29.533 -> 你的命令是:AT+CESQ
19:31:29.581 -> WNB303CheckRSSI result=-86
19:31:39.565 -> 你的命令是:AT+CEREG?
19:31:39.565 -> 你的命令是:AT+CESQ
19:31:39.613 -> 你的命令是:AT+EDNS="mqttgo.io"
19:31:40.610 -> 你的命令是:AT+EMQNEW="20.89.104.82","1883",60000,1024
19:31:41.421 -> 你的命令是:AT+EMQCON= 0,3,"NB IoT-2451369",60000,0,0,"",""
19:31:41.421 -> MQTT Connected
19:31:41.421 -> TDS Value:0ppm
19:31:41.468 -> TDS Value:0ppm
19:31:41.468 -> TDS Value:0ppm
19:31:41.468 -> TDS Value:0ppm
19:31:41.468 -> TDS Value:0ppm
19:31:41.468 -> TDS Value:0ppm
19:31:41.468 -> TDS Value:0ppm
19:31:41.468 -> TDS Value:0ppm
19:31:41.468 -> TDS Value:0ppm
```

傳送

☐ 自動換行 ☒ Show timestamp

NL(newline) 115200 baud Clear output

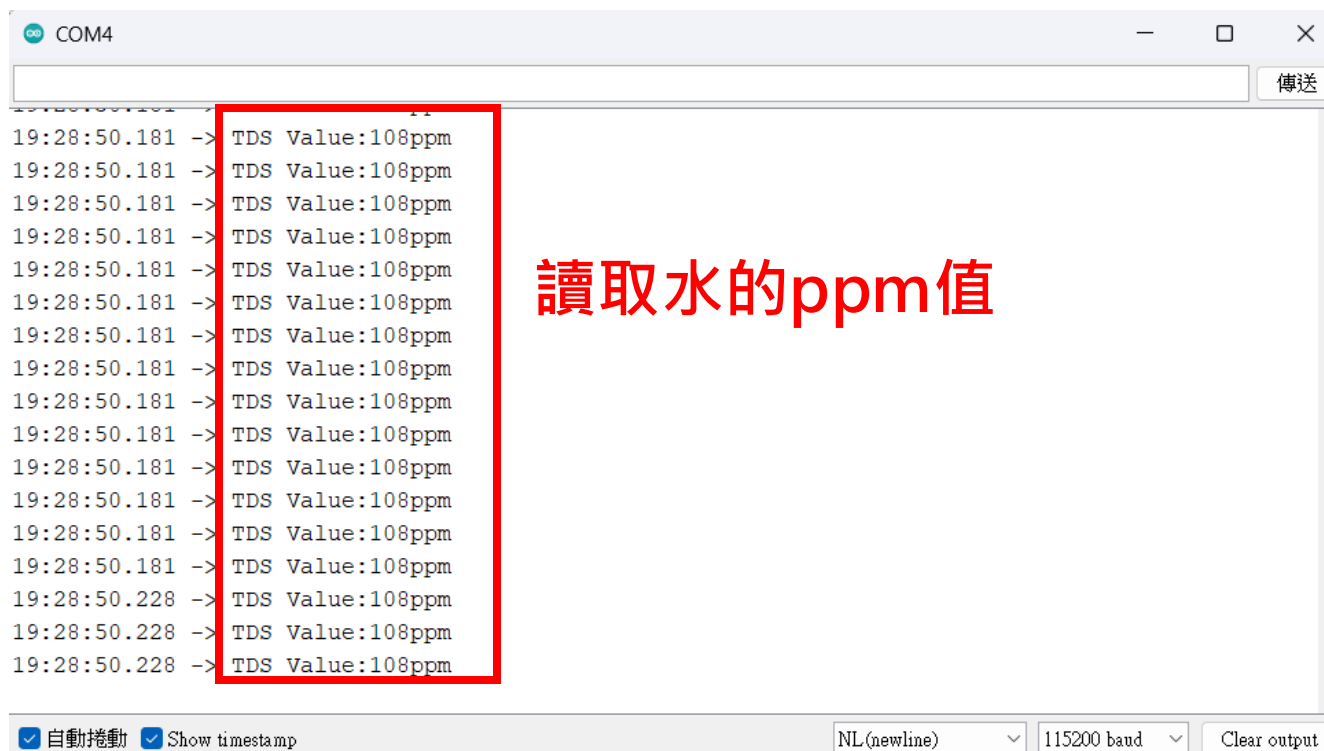
成功註冊網路，與基地台連線成功

顯示目前訊號強度(-86RSSI)

開始與MQTT連線

查看資訊

- 持續讀取水的硬度PPM並推播到MQTT



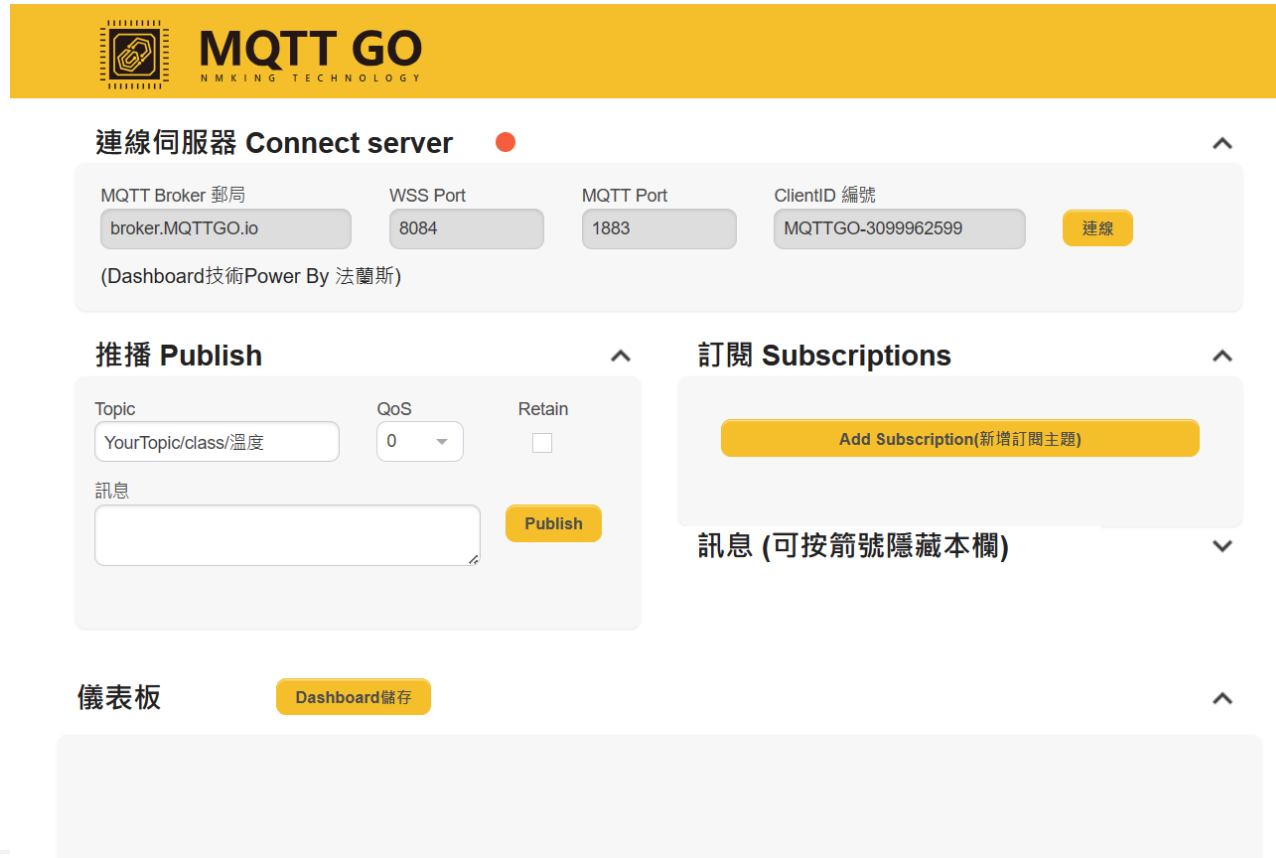
```
COM4
19:28:50.181 -> TDS Value:108ppm
19:28:50.181 -> TDS Value:108ppm
19:28:50.181 -> TDS Value:108ppm
19:28:50.181 -> TDS Value:108ppm
19:28:50.181 -> TDS Value:108ppm
19:28:50.181 -> TDS Value:108ppm
19:28:50.181 -> TDS Value:108ppm
19:28:50.181 -> TDS Value:108ppm
19:28:50.181 -> TDS Value:108ppm
19:28:50.181 -> TDS Value:108ppm
19:28:50.181 -> TDS Value:108ppm
19:28:50.181 -> TDS Value:108ppm
19:28:50.181 -> TDS Value:108ppm
19:28:50.181 -> TDS Value:108ppm
19:28:50.181 -> TDS Value:108ppm
19:28:50.228 -> TDS Value:108ppm
19:28:50.228 -> TDS Value:108ppm
19:28:50.228 -> TDS Value:108ppm
```

讀取水的ppm值

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

MQTT查看資訊

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



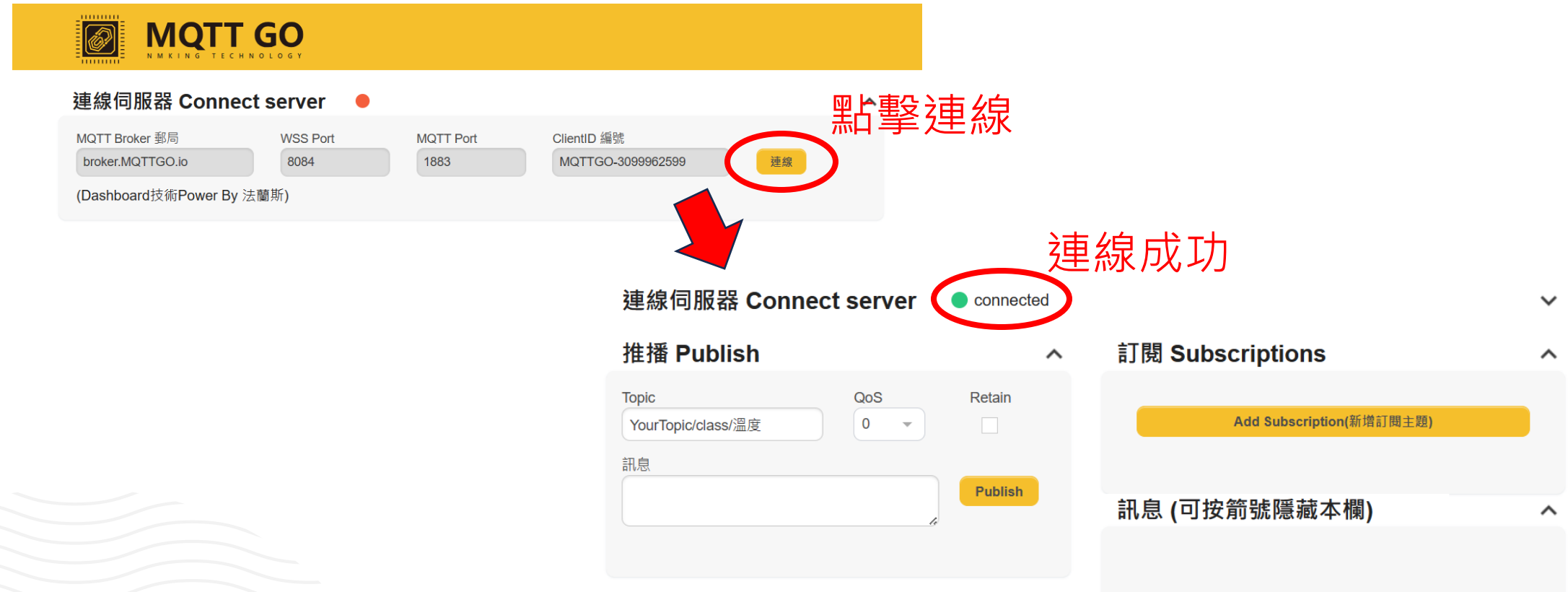
The screenshot shows the MQTT GO web interface. At the top is a yellow header with the MQTT GO logo and the text "MQTT GO" and "NM KING TECHNOLOGY". Below the header is a section titled "連線伺服器 Connect server" with a red status indicator. It contains four input fields: "MQTT Broker 郵局" (broker.MQTTGO.io), "WSS Port" (8084), "MQTT Port" (1883), and "ClientID 編號" (MQTTGO-3099962599). A yellow "連線" button is to the right. Below this is a note "(Dashboard技術Power By 法蘭斯)".

Below the connection section are two main panels. The left panel is titled "推播 Publish" and contains a "Topic" input field (with placeholder "YourTopic/class/溫度"), a "QoS" dropdown (set to 0), a "Retain" checkbox, a "訊息" input field, and a yellow "Publish" button. The right panel is titled "訂閱 Subscriptions" and contains a yellow "Add Subscription(新增訂閱主題)" button and a section for "訊息 (可按箭號隱藏本欄)".

At the bottom is a section titled "儀表板" with a yellow "Dashboard儲存" button and a large empty box for the dashboard.

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 displays 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 includes fields for Topic (YourTopic/class/溫度), QoS (0), Retain (checkbox), and a 'Publish' button. The 'Subscriptions' section has an 'Add Subscription(新增訂閱主題)' button and a note '訊息 (可按箭號隱藏本欄)'.

MQTT查看資訊

- 回到程式碼，將以下框中Topic文字複製下來

```
// ----- 以下修改成你MQTT設定 -----  
String MQTTServer = "mqttgo.io";//免註冊MQTT伺服器  
String MQTTPort = "1883";//MQTT Port  
String MQTTUser = "";//不須帳密  
String MQTTPassword = "";//不須帳密  
//推播主題1:推播PPM(記得改Topic) 複製  
String MQTTPubTopic = "YourTopic/class/TDSPPM";  
long MQTTLastPublishTime = 0;//此變數用來記錄推播時間  
long MQTTPublishInterval = 5000;//每5秒推撥一次
```

MQTT查看資訊

- 回到MQTT GO，點選新增訂閱主題

**MQTT GO**
NM KING TECHNOLOGY

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) ▲

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 configuration interface with several fields and a 'Subscribe' button. Red boxes and numbers 1 through 6 highlight specific elements, with red text annotations explaining their purpose.

- ① Topic 貼上剛剛複製的Topic: Points to the Topic input field containing "YourTopic/class/TDSPPM".
- ② 選擇儀表板: Points to the "儀表板" (Dashboard) dropdown menu, which is currently set to "類比圖" (Analog Chart).
- ③ 更改名稱: Points to the "名稱 (ID)" (Name/ID) input field.
- ④ 調整區間0~2000: Points to the "數值區間 (min.max)" (Value range) input field, which contains "0,2000".
- ⑤ 更改單位:PPM: Points to the "單位" (Unit) input field, which contains "PPM".
- ⑥ 完成後點擊Subscribe訂閱: Points to the yellow "Subscribe" button.

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/class/TDSPPM

X

訊息 (可按箭號隱藏本欄)

2024-02-25 22:44:51
105

Topic: YourTopic/class/T...

Qos: 0

2024-02-25 22:44:50
105

Topic: YourTopic/class/T...

Qos: 0

2024-02-25 22:44:49
105

Topic: YourTopic/class/T...

Qos: 0

2024-02-25 22:44:48
105

Topic: YourTopic/class/T...

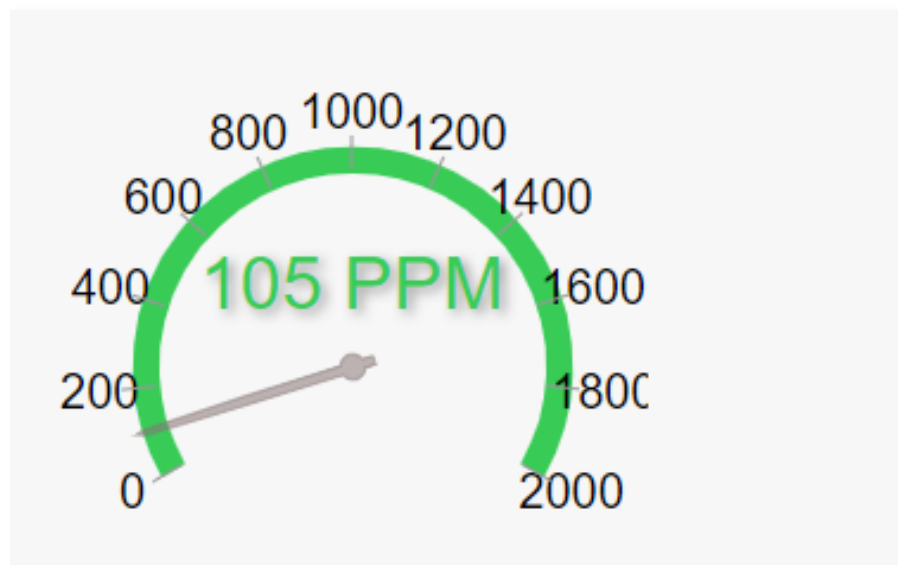
Qos: 0

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





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

