南臺科技大學
Southern Taiwan University of Science and Technology

減碳蘊漁

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

TDS Sensor水質硬度傳感器

NB-IoT與MQTT連線

目錄

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。

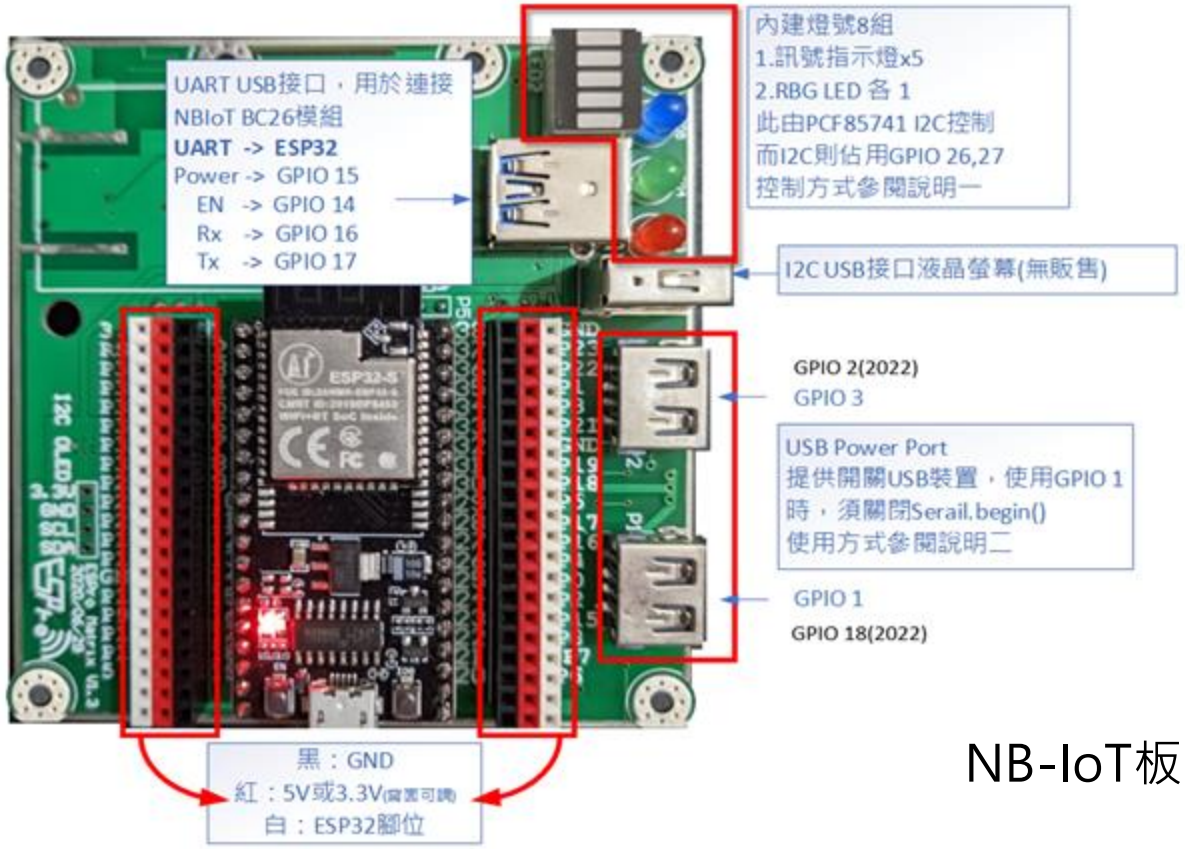
材料

ESP32

數據傳輸線 (MicroUSB)

TDS Meter及防水探頭

材料

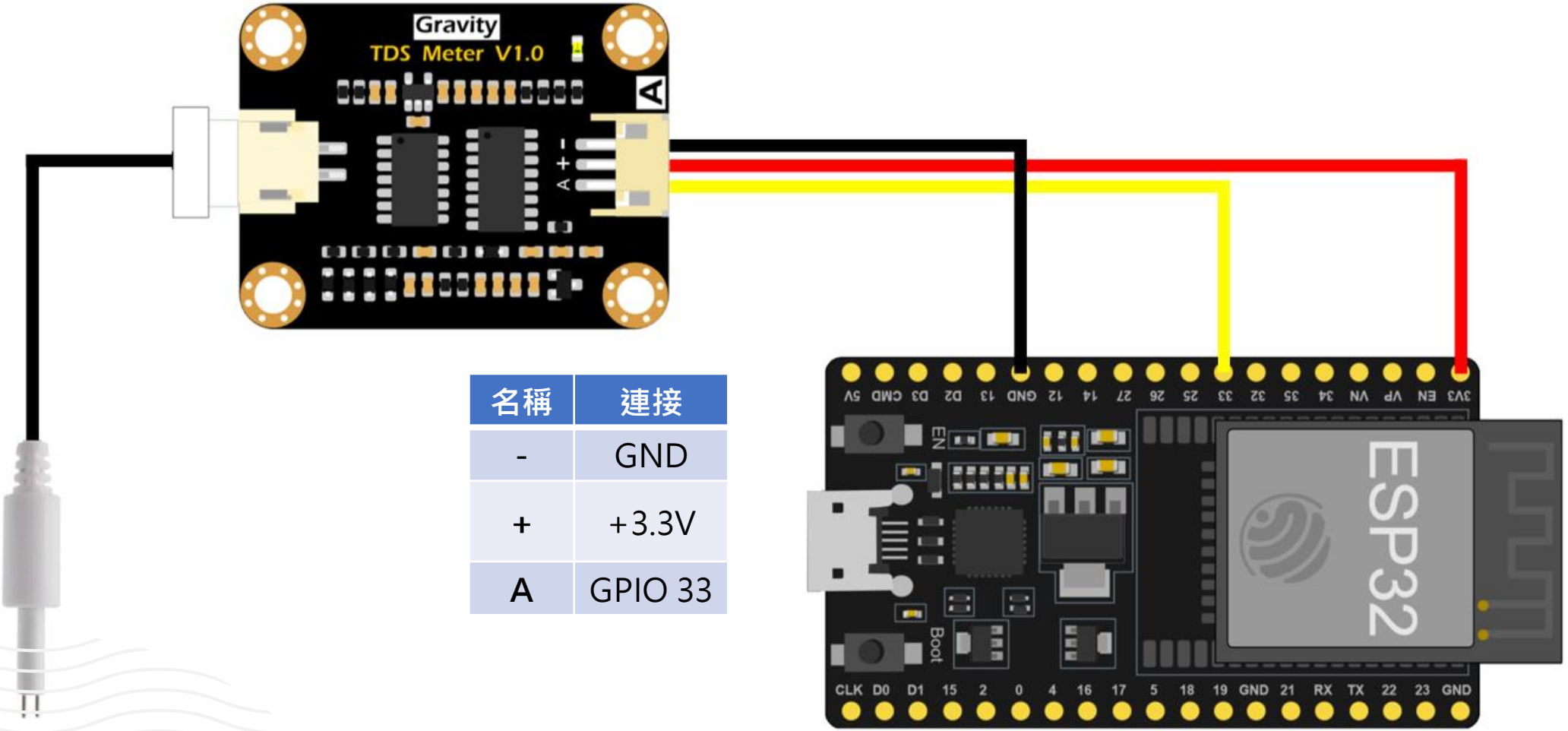


NB-IoT板

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



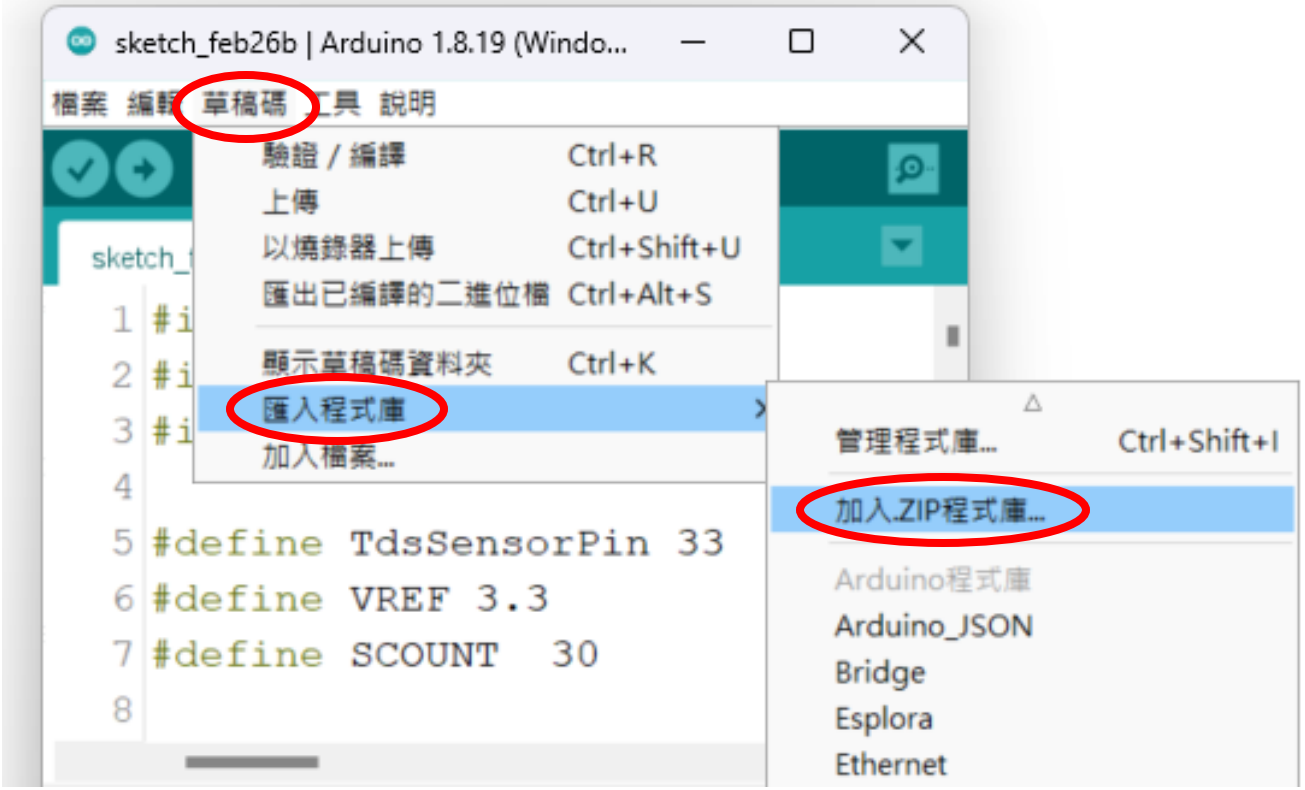
名稱	連接
-	GND
+	+3.3V
A	GPIO 33

安裝函式庫

➢ 至網站下載ZIP檔　　網址： https://twgo.io/mzchv

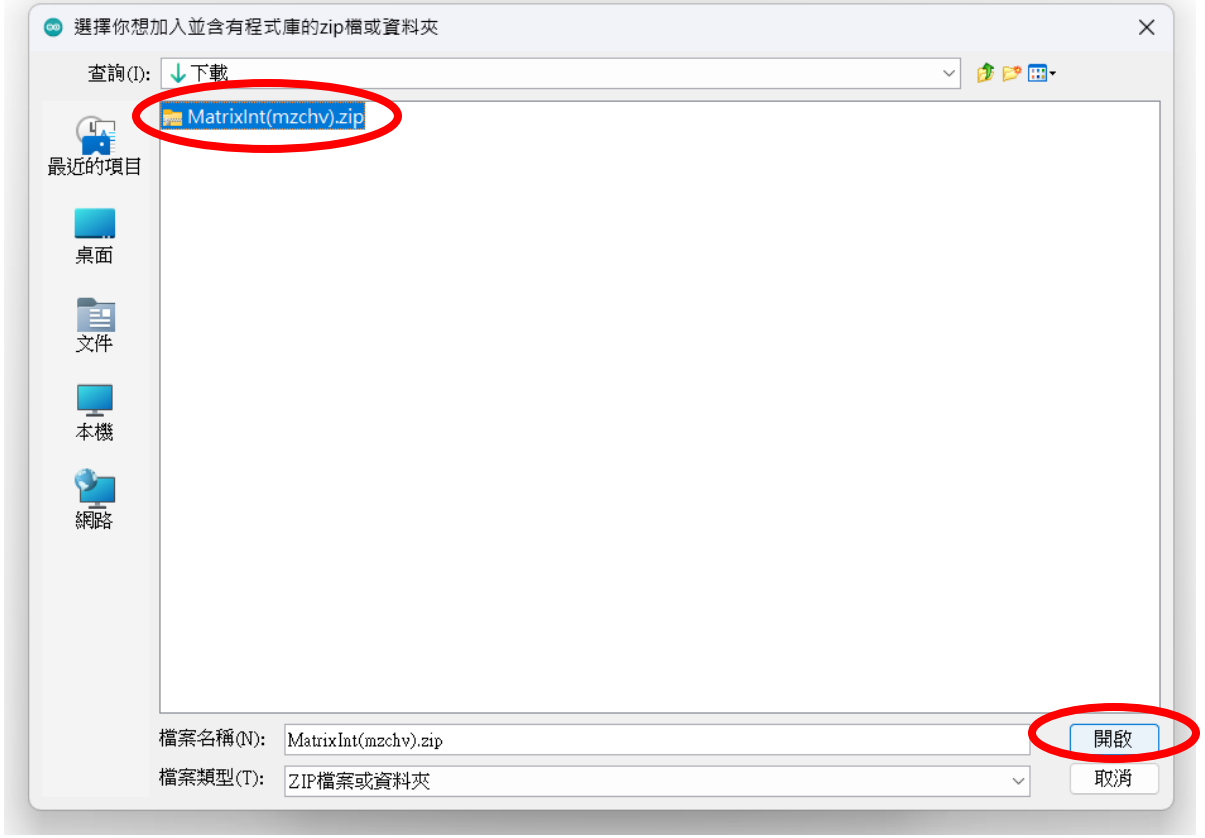
➢ 開啟Arduino IDE

➢ 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



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
//推播主題1:推播PPM(記得改Topic)
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
```

自行更改路徑，
例如：TEST/class402/TDSPPM

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\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 > 40U){    //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 > 800U){
  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 = "NBIoT-" + 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+EMQPUB=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)

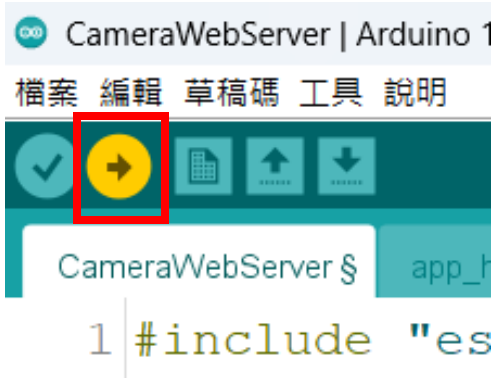
```
//檢查 網路註 冊狀態 return true or false
boolean WNB303CheckReg() {
 String result = "";
 boolean CEREG = false;
 result = AT2WNB303("AT+CEREG?", "+CEREG:");
 if (result.indexOf("0,1") >= 0 || result.indexOf("1,1") >=0) { //註冊 成功
  MatrixInt(5,0);//亮紅 燈
  MatrixInt(6,1);//亮綠 燈
  CEREG =true;
 }
 else {
  MatrixInt(5,1);//亮紅 燈
  MatrixInt(6,0);//亮綠 燈
  CEREG =false;
 }
 return CEREG;
}

//檢查 網路訊 號品質 return RSSI，當 RSSI=0或99代表 沒訊號
int WNB303CheckRSSI() {
 String result = "";
 int RSSI = 0;
 result = AT2WNB303("AT+CESQ", "+CESQ:");
 //處理+CESQ:
 if (!(result == "-1")){
  int CESQ = split(split(result, ':', 1), ',', 0).toInt();
  RSSI = CESQ - 111;
 }
 if (RSSI==-111 || RSSI == 210) RSSI = 0;
 MatrixLEDrss(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 Url) {
 //例如 http://x.xx.x/update?api_key=xxxxx&field1=60
 //拆解 成Protocol="http" host="x.x.x" port="80" Url="/update?api_key=Cxxxxxx&field1=60"
 //1.轉換 網址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
 String PIP = Protocol + "://" + IP + ":" + port + "/";
 int PIPlen = PIP.length();
 //Serial.println(" PIP=" +PIP + " ,len=" +String(PIPlen));
 result = AT2WNB303("AT+EHTTPCREATE=0," + String(PIPlen) + "," + String(PIPlen) + ",\"" +PIP +"\"", "+EHTTPCREAT");
 result.trim(); String clientid = split(result, ':', 1);
 if (result == "") return "Error: Can't Create Connection";
 else {

//3.開啟 連線
  result = AT2WNB303("AT+EHTTPCON=" + clientid, "OK");
  result.trim();
  if (!(result == "OK")) result = "Error: Can't Connect to Server";
  //4.組成 網址並 傳送
  int LenUrl = Url.length();
  Url = clientid + ",0," + String(LenUrl) + "," + Url + ",0,0,0,";
  Url = "0," + String(Url.length()) + "," + String(Url.length()) + "," + Url;
  result = AT2WNB303("AT+EHTTPSEND=" +Url, "OK");
  if (!(result == "OK")) result = "Error: Can't Send to Server";
 }
 delay(1000);
 //關閉 連線
 AT2WNB303("AT+EHTTPDISCON=" + clientid, "OK");
 //Serial.println(result);
 delay(1000);
 AT2WNB303("AT+EHTTPDESTROY=" + clientid, "OK");
 //Serial.println(result);
 delay(1000);
 return result;
}
```

```
//HTTP POST
String HTTPPOST(String Protocol, String Host, String port, String Url, String contType , String Data) {
 //1.轉換 網址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
 String PIP = Protocol + "://" + IP + ":" + port + "/";
 int PIPlen = PIP.length();
 //Serial.println(" PIP=" +PIP + " ,len=" +String(PIPlen));
 result = AT2WNB303("AT+EHTTPCREATE=0," + String(PIPlen) + "," + String(PIPlen) + ",\"" +PIP +"\"", "+EHTTPCREAT");
 result.trim(); String clientid = split(result, ':', 1);
 if (result == "") return "Error: Can't Create Connection";
 else {
  //3.開啟 連線
  result = AT2WNB303("AT+EHTTPCON=" + 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 PostCommand[10];
  PostCommand[i++] = clientid + ",1," + Url.length() + "," +Url + ",0,," + contType.length() + "," +contType + ",";
  //Serial.println(" command1=" + PostCommand[i - 1]);
  String hexData = Str2Hex(Data);
  int lenHexData = hexData.length();
  PostCommand[i++] = String(lenHexData) + ",";
  //Serial.println(" command2=" + PostCommand[i - 1]);
  int num = 30;
  for (int n = 0; n < lenHexData; n = n + num ) {
   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);
   }
   PostCommand[i++] = DataSend;
   //Serial.println(" command2=" + PostCommand[i - 1]);
  }
  //求出 每條命 令長度
  int totalLenHexData =0;
  for (int j = 0; j<= i - 1; j++) {
   totalLenHexData =totalLenHexData + PostCommand[j].length();
   PostCommand[j] = String(PostCommand[j].length()) + "," + PostCommand[j];
  }

  //完成 命令組 合
  for (int j = 0; j<= i - 1; j++) {
   if (j == 1) PostCommand[j] = "AT+EHTTPSEND=1," + String(totalLenHexData) + "," + PostCommand[j];
   else if (j== i - 1) PostCommand[j] = "AT+EHTTPSEND=0," + String(totalLenHexData) + "," + PostCommand[j];
   else PostCommand[j] = "AT+EHTTPSEND=1," +String(totalLenHexData) + "," +PostCommand[j];
   result = AT2WNB303(PostCommand[j], "OK");
   //Serial.println(" POST" + String(j) + ":" + result);
  }

 }
 delay(1000);
 //關閉 連線
 AT2WNB303("AT+EHTTPDISCON=" + clientid, "OK");
 //Serial.println(result);
 delay(1000);
 AT2WNB303("AT+EHTTPDESTROY=" + 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 AT2WNB303(String ATdata, String StartWith) {
 Serial.println("你的 命令是:" + ATdata);
 if (ATdata.length() > 0) { //送出AT命令
  ATdata.trim();
  String command = ATdata;
  command.toUpperCase();
  if (command == "RESET") { //重置 WNB303
   digitalWrite(WNB303ResetPIN, HIGH);
   delay(10000);
   digitalWrite(WNB303ResetPIN, LOW);
   return "RESET OK";
  }
  else if (command == "POWERON") { //開啟 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
   Serial2.println(ATdata);
  }
 }
 else return "";
 String result = "";
 //等候 回應資 料
 long StartTime = millis();
 while (1) {
  result = "";
  while (Serial2.available()){ //WNB303有資 料回傳
   char c = Serial2.read();  //從WNB303讀取 一個位 元組
   result += c;              //將讀 到的字 元 c 加進 字串 Xfer
   if (c== '\n') break;
  }
  result.trim();
  if (result.startsWith(StartWith)) break; //結尾 OK返回
  if (millis() - StartTime) >= WNB303Timeout ) { //Timeout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 = 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) - 1; i += 2) {
  char temp[3];
  temp[0] = input[i];
  temp[1] = input[i + 1];
  int val = ASCIIHexToInt(temp[0]) * 16 + ASCIIHexToInt(temp[1]);
  c[i] = toascii(val);
  a = a+ String(c[i]);
 }
 return a;
}

//ASC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 a1=split( "aa,bb,cc" ,',' ,0);
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[0] = strIndex[1] + 1;
   strIndex[1] = (i==maxIndex)? i+ 1 : i;
  }
 }
 return found > index ? data.substring(strIndex[0], strIndex[1]) :"";
}
```

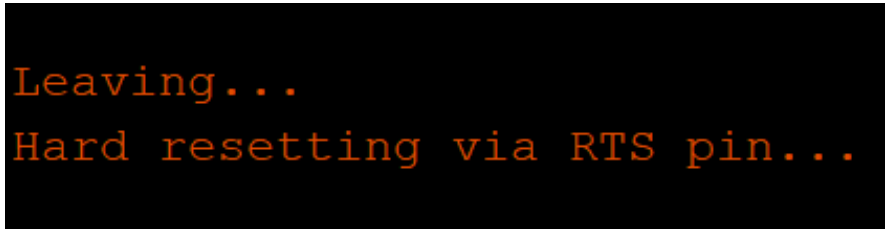
寫入程式步驟

➢ 1.確定工具欄位下的選項有正確選擇

➢ 2.確認後點擊上傳

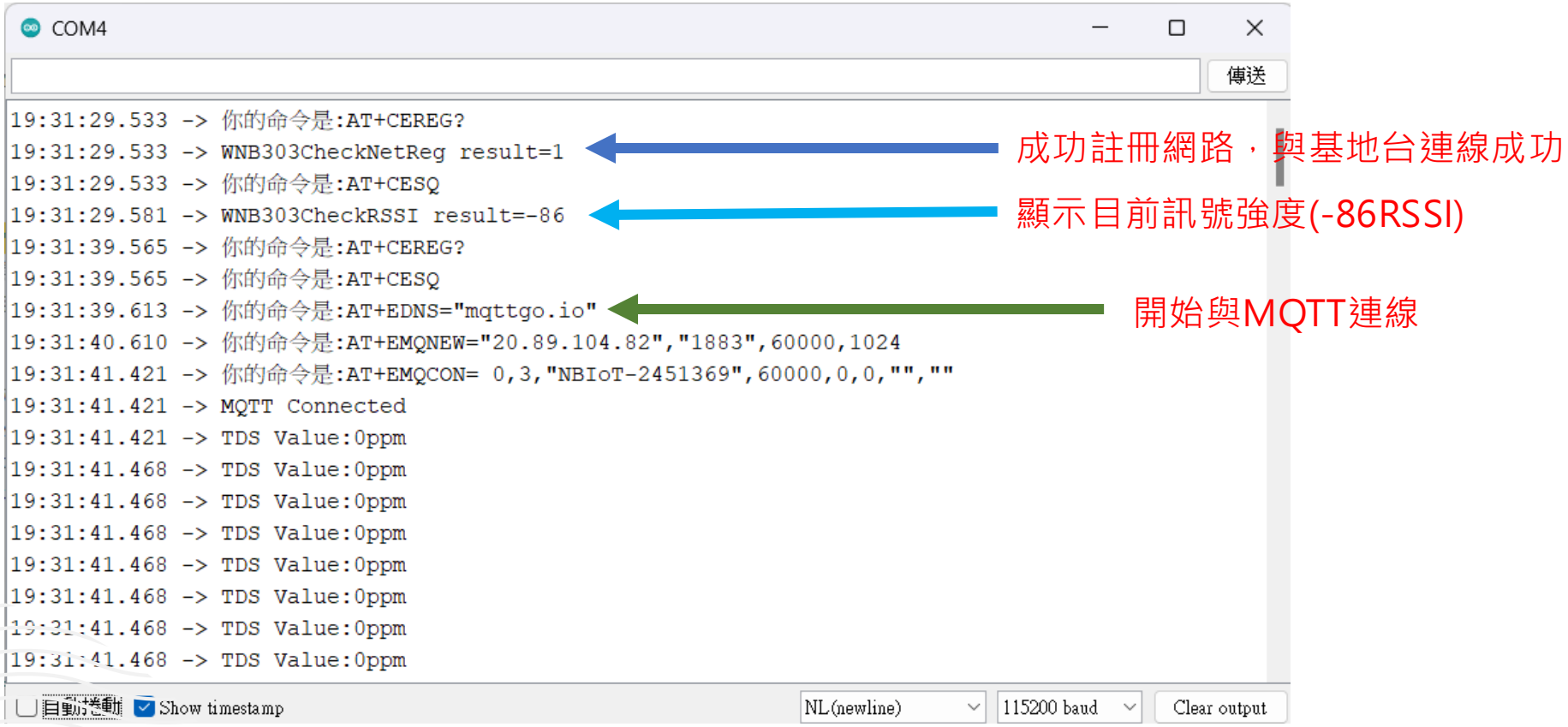


➢ 3.等待底下出現此字串即成功



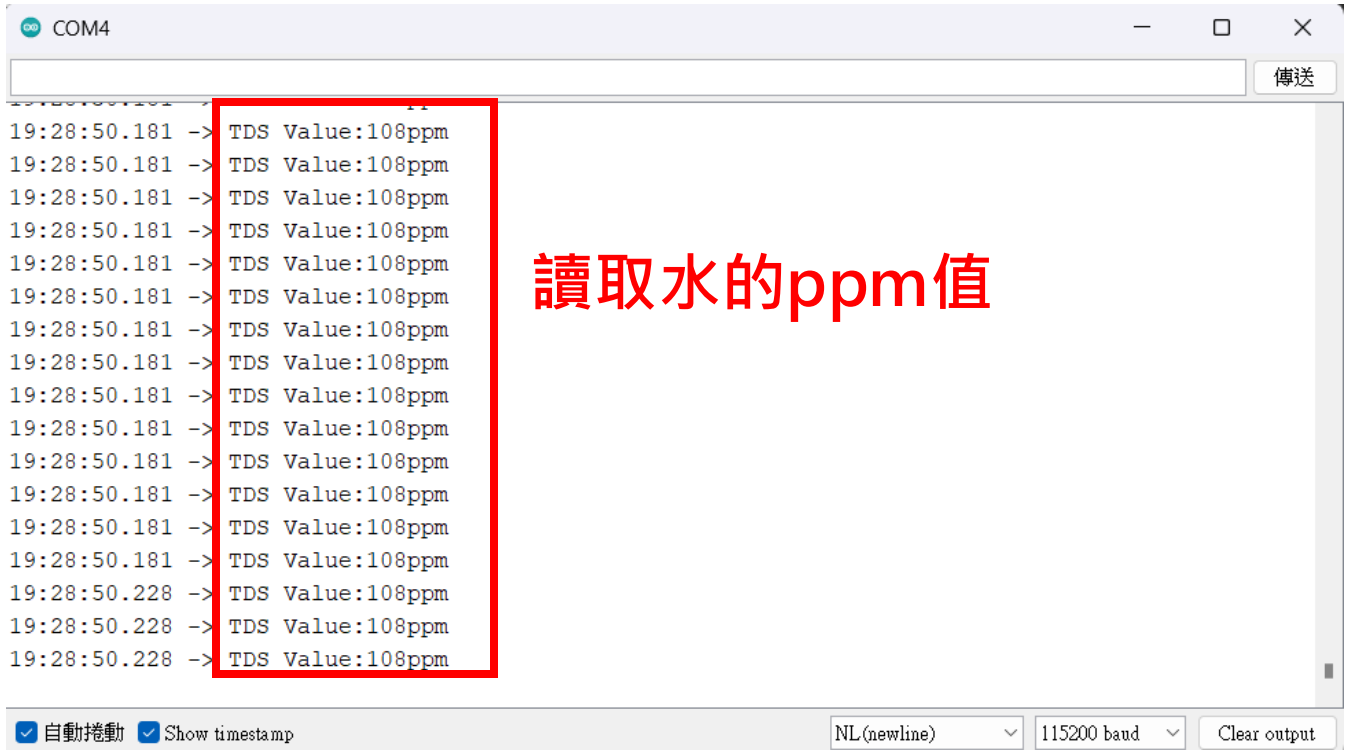
查看資訊

➢ 開啟序列埠視窗，查看連線狀態



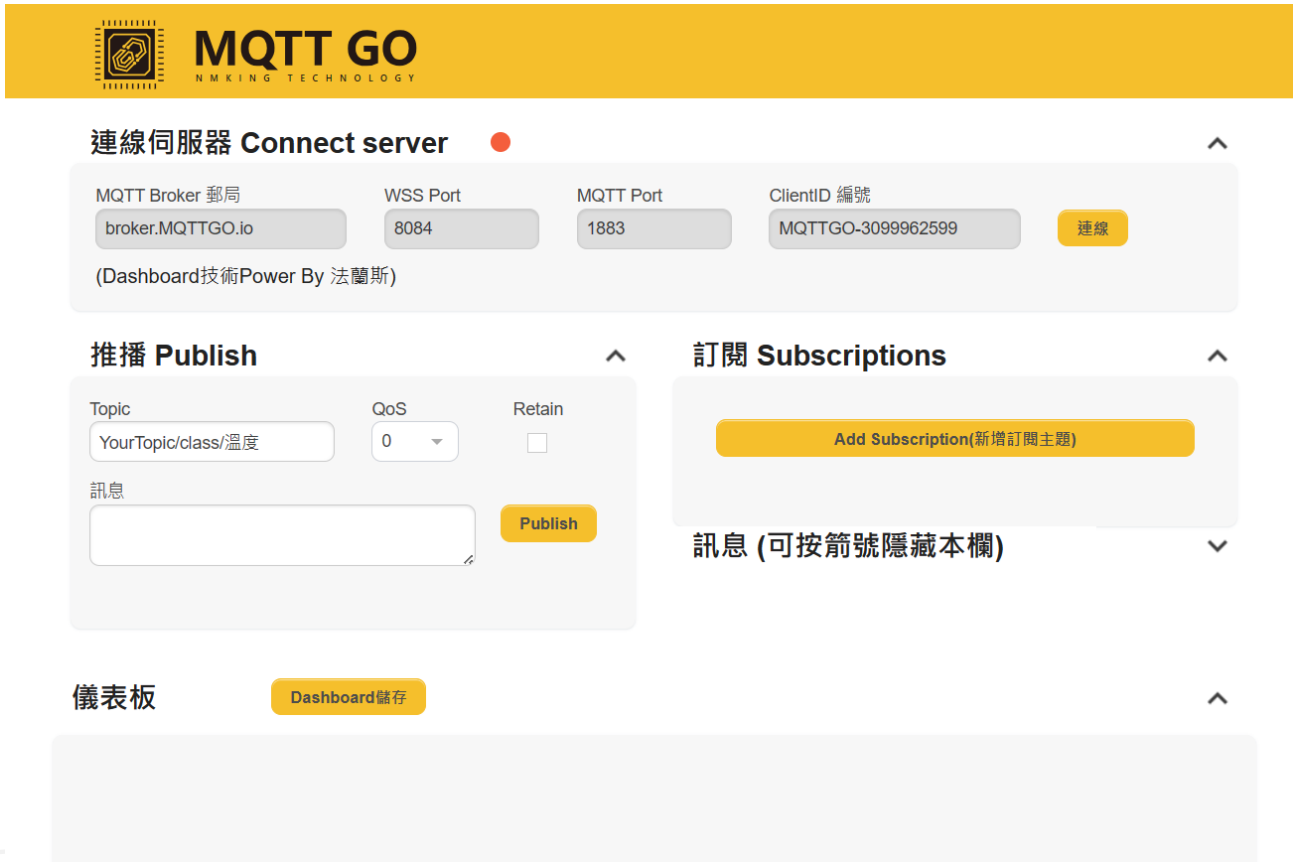
查看資訊

➢ 持續讀取水的硬度PPM並推播到MQTT



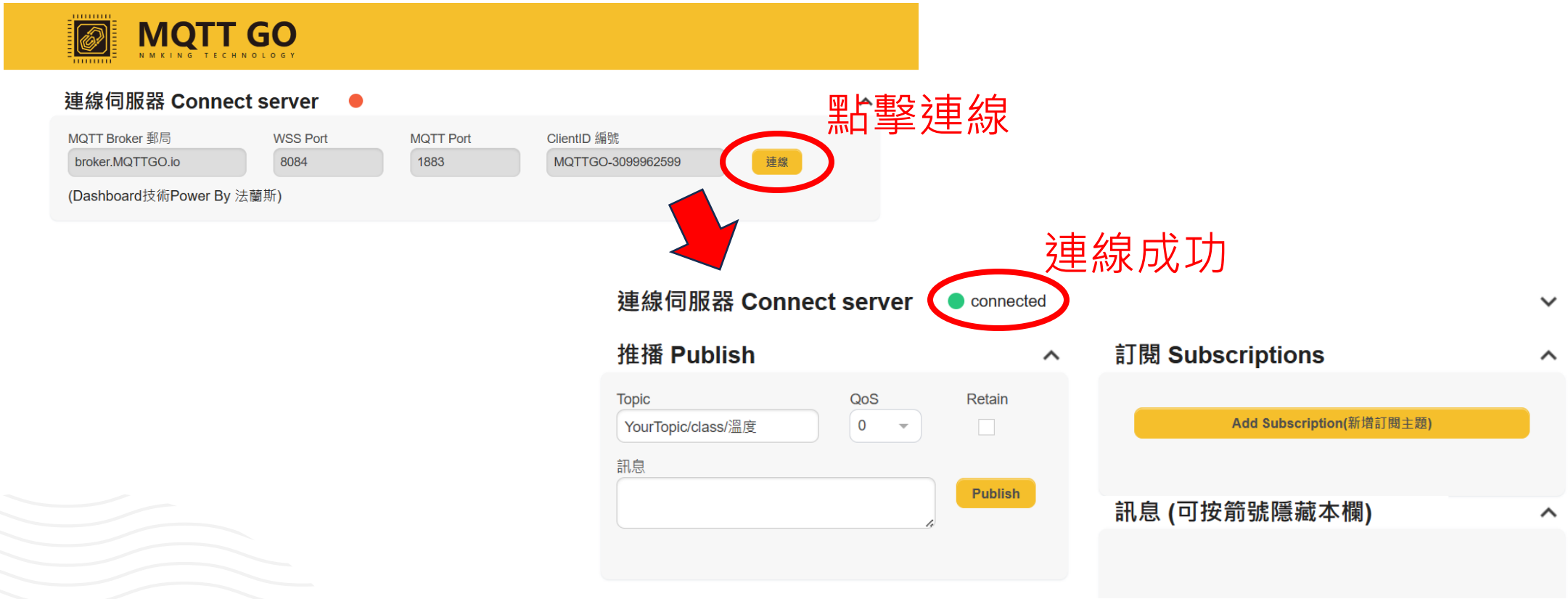
MQTT查看資訊

➢ 於瀏覽器開啟網站：https://broker.mqttgo.io/



MQTT查看資訊

➢ 點擊連線，待燈號亮綠燈顯示connected即連線成功



MQTT查看資訊

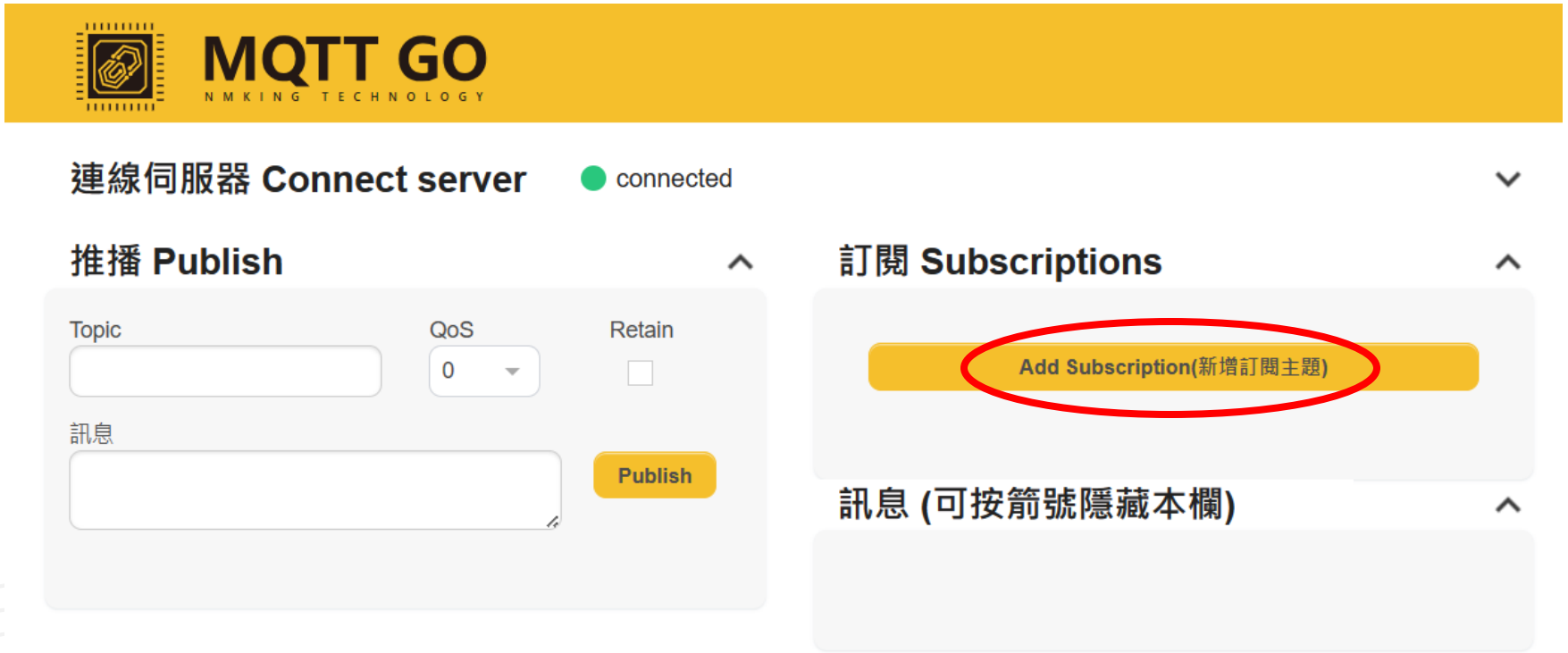
➢ 回到程式碼，將以下框中Topic文字複製下來

```
// ------ 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)
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
```

複製

MQTT查看資訊

➢ 回到MQTT GO，點選新增訂閱主題

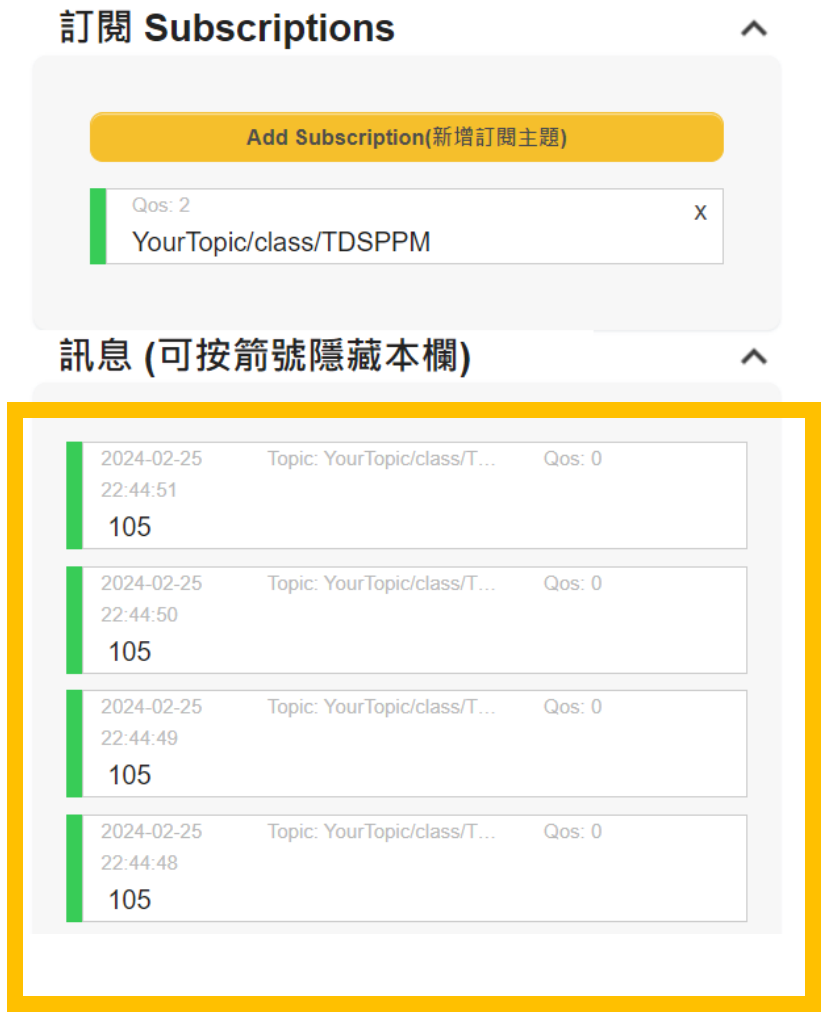


MQTT查看資訊

➢ 將剛剛複製的路徑貼到Topic，按步驟修改完成後點擊Subscribe

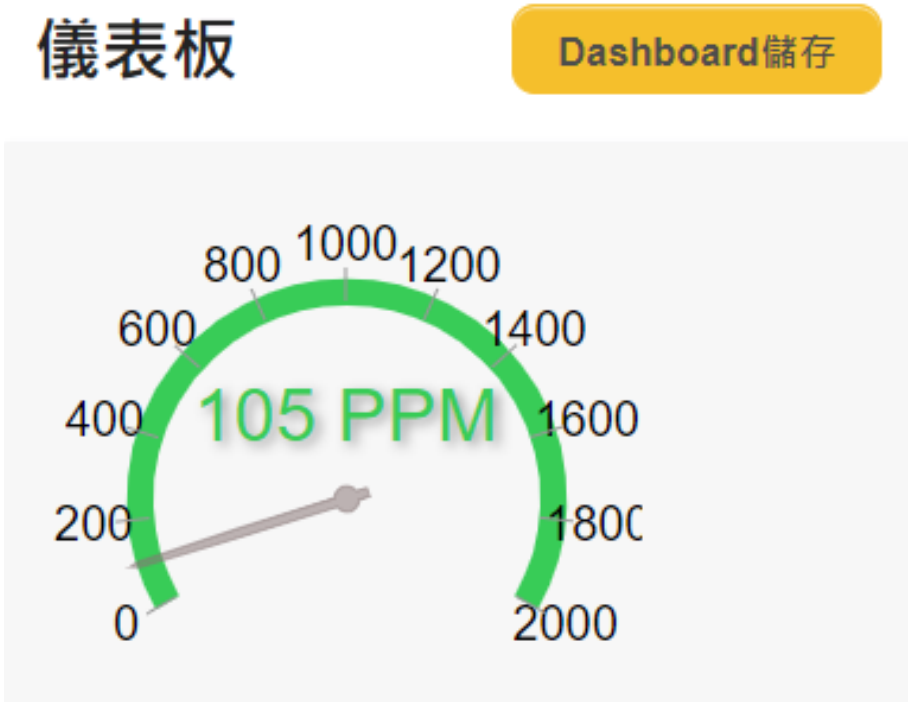
MQTT查看資訊

➢ 訊息欄可看到接收到的資訊



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



減碳蘊福

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

STUST
1969