

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

水位感測器

NB-IoT與MQTT連線



目錄



材料

模組腳位說明

接線說明

程式撰寫說明

寫入程式步驟

序列埠查看資訊

MQTT查看資訊

材料



ESP32

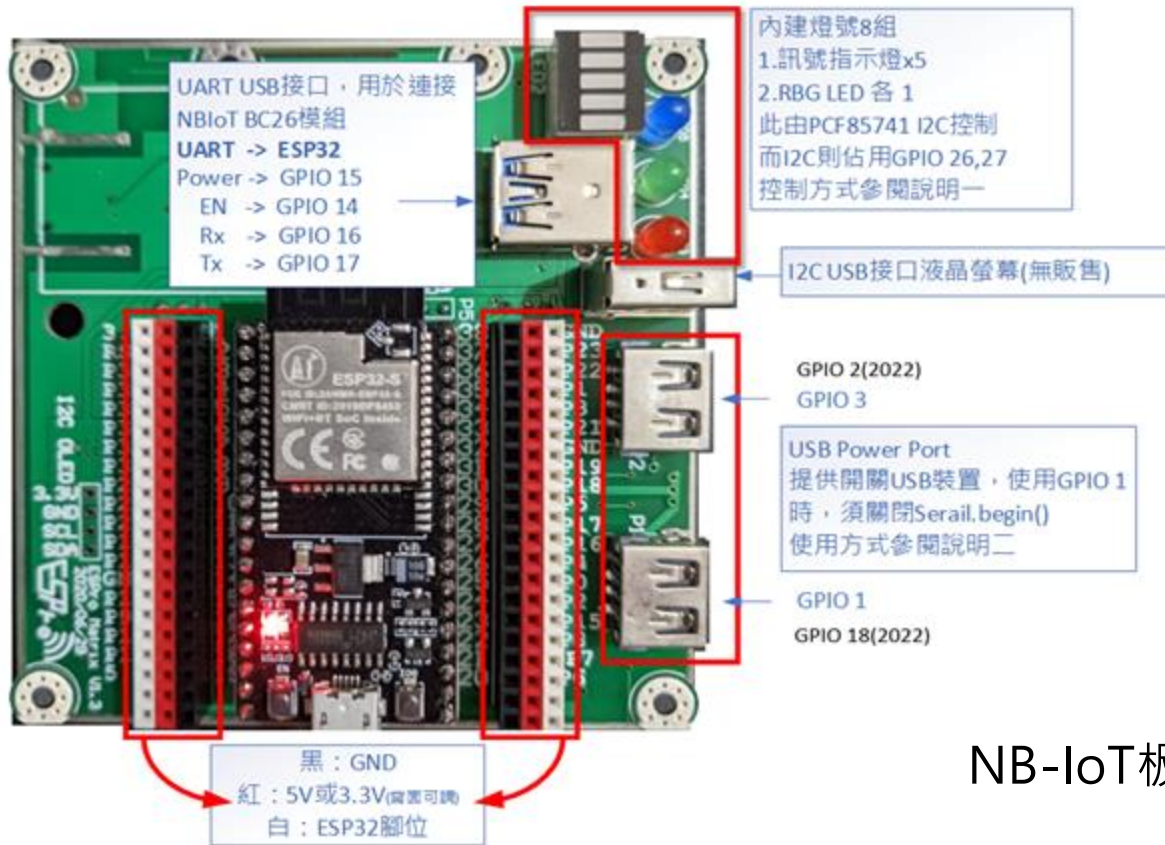


數據傳輸線 (MicroUSB)



水位感測器模組

材料



NB-IoT板

注意事項:

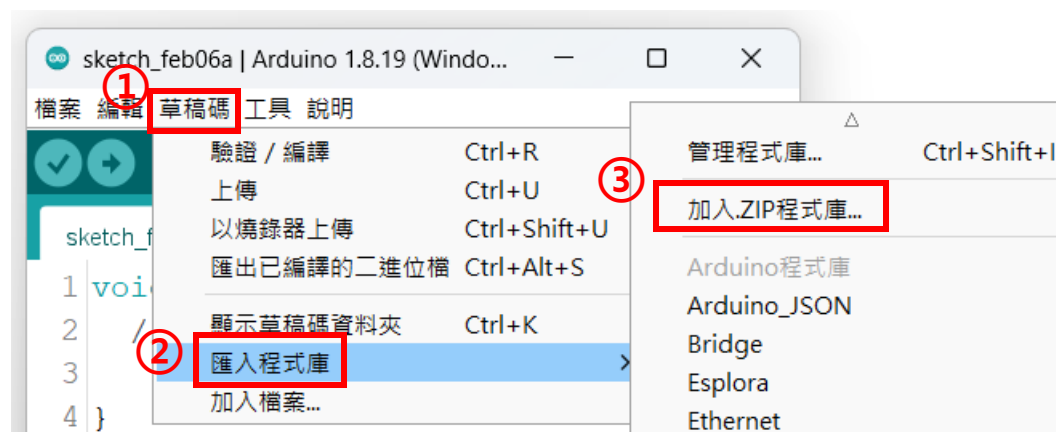
移除更換sim卡，請務必關閉電源，否則會短路冒煙損毀

匯入程式庫

- 先下載MatrixInt程式庫

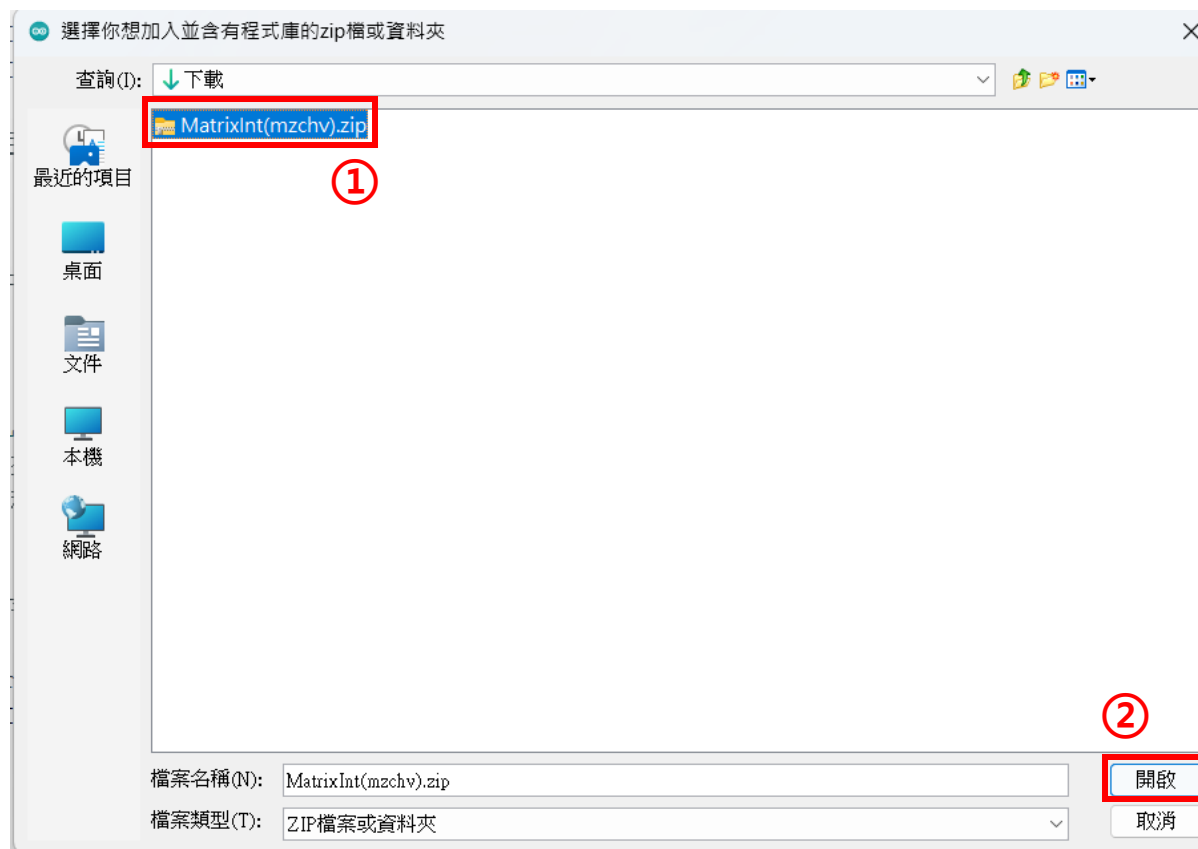
下載連結：[MatrixInt\(mzchv\).zip - Google 雲端硬碟](#)

- 下載後進行安裝，到Arduino IDE的草稿碼/匯入程式庫/加入ZIP程式庫



匯入程式庫

- 找到剛剛下載的檔案，選擇開啟，即完成安裝步驟



程式撰寫步驟(1/10)

- 開啟記事本
「範例程式 水位感測器 NB-IoT.txt 」
- 複製內容並貼上Arduino IDE視窗中

程式撰寫步驟(2/10)

```
#include <WiFi.h>
#include <Wire.h>
#include "MatrixInt.h" //請先安裝

long WNB303Timeout = 10000; //返回0=Timeout
int water_sensor=36; // 水位感測器 S 信號腳 連接到ESP32的 GPIO36
//使用小霸王Matrix板
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
int a;
```

程式撰寫步驟(3/10)

- 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 更改路徑，例如：TEST/class402/WaterLevel

String MQTTPubTopic = "YourTopic/class402/WaterLevel";//推播主題:推播土壤濕度

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(4/10)

```
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(water_sensor, INPUT); //設置water_sensor對應的腳GPIO36為輸入
  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);
}
```

程式撰寫步驟(5/10)

```
void loop() {
```

```
//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");
```

程式撰寫步驟(6/10)

```
int val= analogRead(soil_sensor); //土壤濕度值給val
a=map(val,0,4095,0,100);      //將val轉換成百分比顯示5
Serial.print("水位狀態：");
Serial.print(val);
Serial.println(" val");
Serial.print("水位百分比：");
Serial.print(a);
Serial.println(" %");
String payload = String((int)a);
//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(7/10)

```
//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+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;
}

//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;
}
```

程式撰寫步驟(8/10)

//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;  
    MatrixLEDrssi(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.x.x.x/update?api_key=xxxxx&field1=60  
    //拆解成Protocol="http" host="x.x.x.x" port="80" Url="/update?api_key=CxxxxxxJ&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+EHTTSEND=" + Url, "OK");  
        if (!(result == "OK")) result = "Error: Can't Send to Server";  
    }  
    delay(1000);  
}
```

程式撰寫步驟(9/10)



```
//關閉連線
AT2WNB303("AT+EHHTPDISCON=" + clientid, "OK");
//Serial.println(result);
delay(1000);
AT2WNB303("AT+EHHTPDESTROY=" + clientid, "OK");
//Serial.println(result);
delay(1000);
return result;
}

//HTTP POST
String HTTPPOST(String Protocol, String Host, String port, String Url, String contentType, 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+EHHTPCREATE=0," + String(PIPlen) + "," + String(PIPlen) + "\",\"PIP + \"\", \"EHHTPCREAT");
result.trim(); String clientid = split(result, ':', 1);
if (result == "") return "Error: Can't Create Connection";
else {
//3.開啟連線
result = AT2WNB303("AT+EHHTPCON=" + 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,\" + contentType.length() + "," + contentType + ",";
//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+EHHTPSEND=1," + String(totalLenHexData) + "," + PostCommand[j];
else if (j == i - 1) PostCommand[j] = "AT+EHHTPSEND=0," + String(totalLenHexData) + "," + PostCommand[j];
else PostCommand[j] = "AT+EHHTPSEND=1," + String(totalLenHexData) + "," + PostCommand[j];
result = AT2WNB303(PostCommand[j], "OK");
//Serial.println("POST" + String(j) + " " + result);
}
}
delay(1000);
//關閉連線
AT2WNB303("AT+EHHTPDISCON=" + clientid, "OK");
//Serial.println(result);
delay(1000);
AT2WNB303("AT+EHHTPDESTROY=" + clientid, "OK");
//Serial.println(result);
delay(1000);
return result;
}
```

程式撰寫步驟(10/10)



```
//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

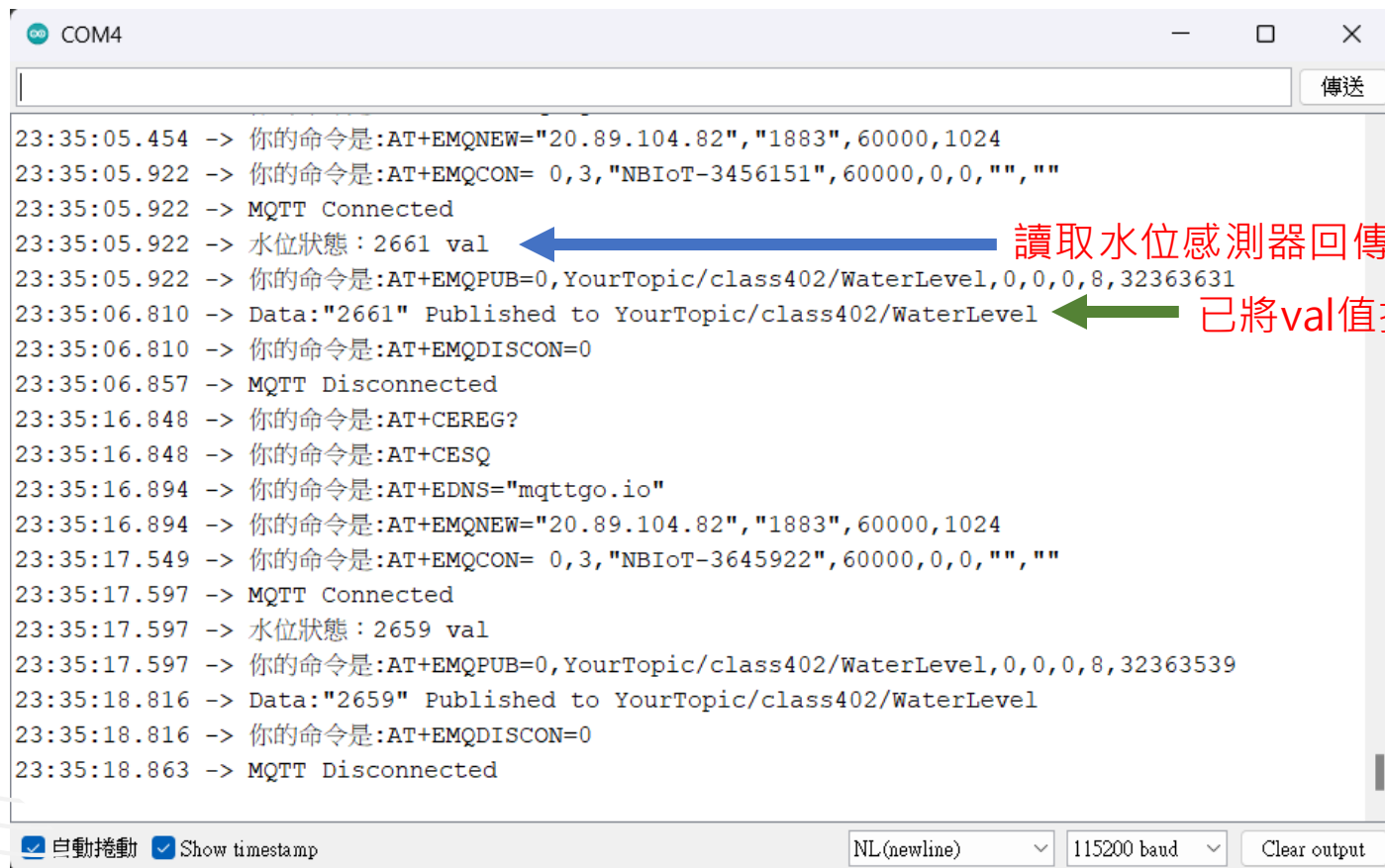


➤ 3.等待底下出現此字串即成功

```
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

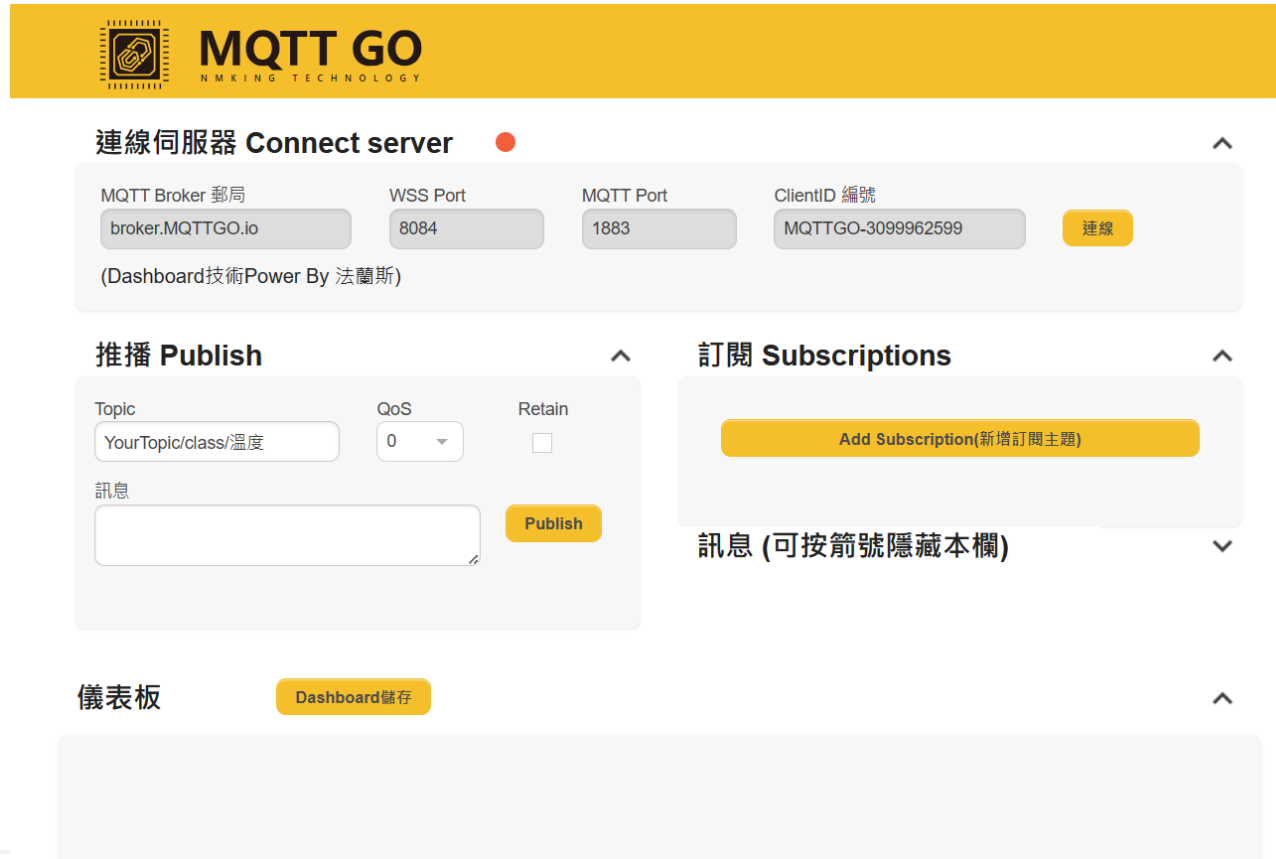


```
COM4
23:35:05.454 -> 你的命令是:AT+EMQNEW="20.89.104.82","1883",60000,1024
23:35:05.922 -> 你的命令是:AT+EMQCON= 0,3,"NBIoT-3456151",60000,0,0,"",""
23:35:05.922 -> MQTT Connected
23:35:05.922 -> 水位狀態: 2661 val
23:35:05.922 -> 你的命令是:AT+EMQPUB=0,YourTopic/class402/WaterLevel,0,0,0,8,32363631
23:35:06.810 -> Data:"2661" Published to YourTopic/class402/WaterLevel
23:35:06.810 -> 你的命令是:AT+EMQDISCON=0
23:35:06.857 -> MQTT Disconnected
23:35:16.848 -> 你的命令是:AT+CEREG?
23:35:16.848 -> 你的命令是:AT+CESQ
23:35:16.894 -> 你的命令是:AT+EDNS="mqttgo.io"
23:35:16.894 -> 你的命令是:AT+EMQNEW="20.89.104.82","1883",60000,1024
23:35:17.549 -> 你的命令是:AT+EMQCON= 0,3,"NBIoT-3645922",60000,0,0,"",""
23:35:17.597 -> MQTT Connected
23:35:17.597 -> 水位狀態: 2659 val
23:35:17.597 -> 你的命令是:AT+EMQPUB=0,YourTopic/class402/WaterLevel,0,0,0,8,32363539
23:35:18.816 -> Data:"2659" Published to YourTopic/class402/WaterLevel
23:35:18.816 -> 你的命令是:AT+EMQDISCON=0
23:35:18.863 -> MQTT Disconnected

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 "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 contains four input fields: "MQTT Broker 郵局" (broker.MQTTGO.io), "WSS Port" (8084), "MQTT Port" (1883), and "ClientID 編號" (MQTTGO-3099962599). There is a "連線" (Connect) button and a note "(Dashboard技術Power By 法蘭斯)".

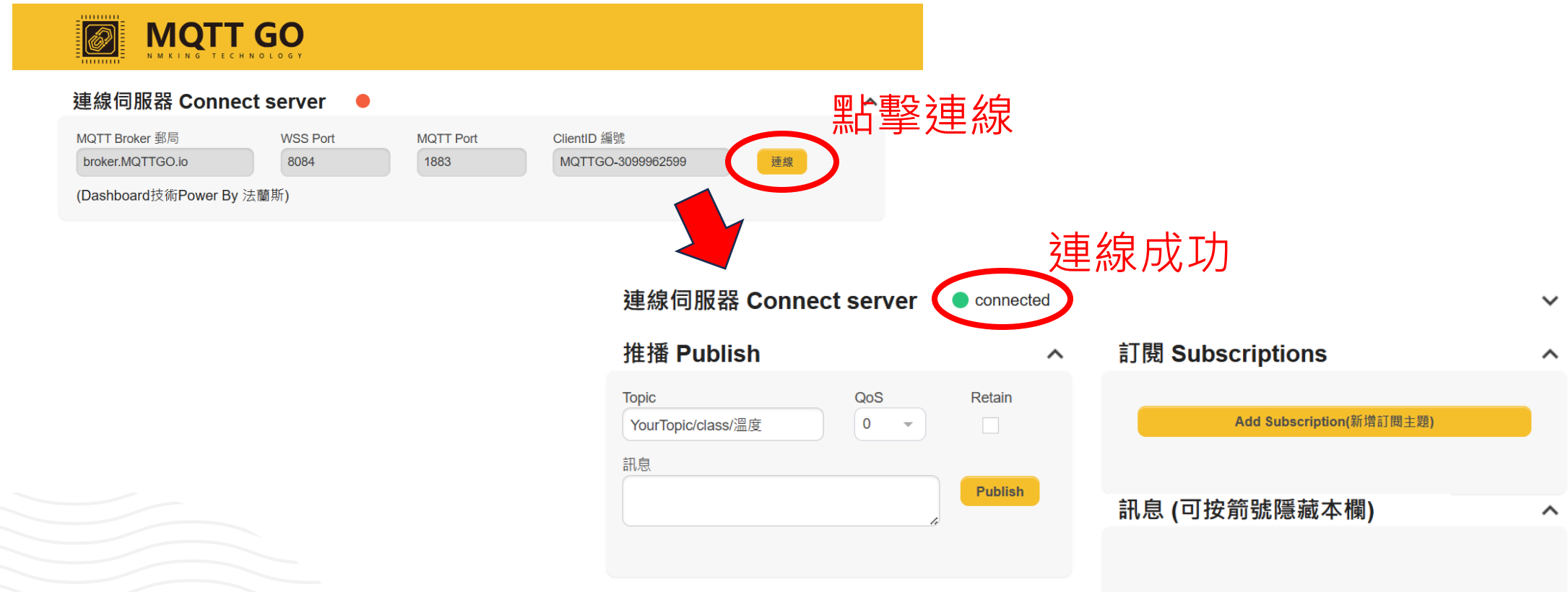
The "推播 Publish" section has a "Topic" field (YourTopic/class/溫度), a "QoS" dropdown (0), a "Retain" checkbox, a "訊息" (Message) input field, and a "Publish" button.

The "訂閱 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)".

At the bottom, there is a "儀表板" (Dashboard) section with a "Dashboard儲存" (Save Dashboard) 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 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 status 'connected' is shown with a green dot and a red circle, with the text '連線成功' (Connection successful) next to it. The 'Publish' section includes fields for Topic (YourTopic/class/溫度), QoS (0), and Retain (checkbox), along with 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)' (Message (can be hidden by arrow)).

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
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);
```

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 web interface for subscribing to an MQTT topic. The interface includes a 'Color' selector (blue), a 'QoS' dropdown (set to 2), and a 'Subscribe' button. The 'Topic' field contains 'YourTopic/class402/WaterLevel'. The '儀表板' (Dashboard) dropdown is set to '水位圖' (Water Level Chart). The '數值區間 (min,max)' (Value Range) field is set to '0,100'. The '單位' (Unit) field is set to '%'. Annotations with red boxes and numbers 1-4 guide the user through the steps: 1. Topic, 2. 儀表板, 3. 數值區間, and 4. Subscribe button.

Color QoS 2 Subscribe

① Topic 貼上剛剛複製的Topic
YourTopic/class402/WaterLevel

② 儀表板 水位圖
選擇儀表板

③ 數值區間 (min,max) 0,100
調整區間0~100

④ 單位 %
完成後點擊Subscribe訂閱

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 12:30:11 Topic: YourTopic/class40... Qos: 0

59

2024-04-07 12:30:08 Topic: YourTopic/class40... Qos: 0

59

2024-04-07 12:30:05 Topic: YourTopic/class40... Qos: 0

59

2024-04-07 12:30:02 Topic: YourTopic/class40... Qos: 0

59

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

