

# DSI2598+

## 淹水感測及防水閘門監測

設計者：張軒翊

指導單位：經濟部工業局

執行單位：資策會數位服務創新研究所

# 發想動機

---



摘要



硬體介紹



程式介紹



成果展示

# 第一章

# 摘要

1-1 案例簡介與應用

1-2 所需材料



## I-I 案例簡介與應用

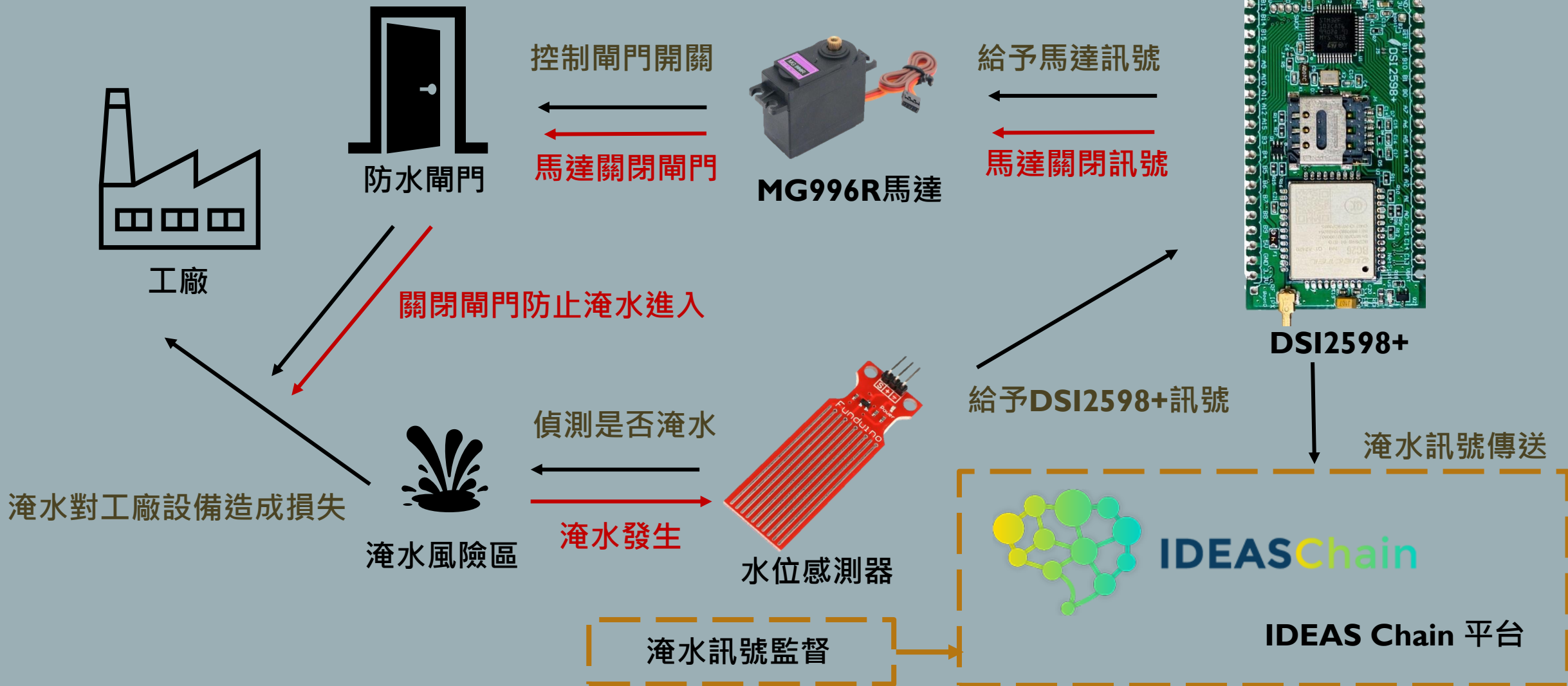
---

本範例適用於無人化工廠，在無人員於工廠時能有效監測工廠之環境參數，及對突發狀況作出應對以預防設備及廠區損壞。

其應用為透過水位感測器偵測廠區附近易淹水區域，傳遞是否淹水的訊號給防水閘門以在適當時機驅動馬達。最後利用DSI2598+開發板回傳淹水資訊，以便遠距離之監測。

# 案例簡介與應用

示意圖





## 1-2 所需材料

---

1. **DSI2598+ x1**
2. 水位感測器 x1
3. **MG 996R**伺服馬達 x1
4. 排線x少許

## 第二章

# 硬體介紹

### 2-1 DSI2598+基礎介紹

2-1-1 腳位介紹

2-1-2 接線介紹

### \*感測器介紹

2-2-1 水位感測器

2-2-2 MG 996R

## 2-1 DSI2598+基礎介紹



- NB-IoT使用MTK MT2625晶片
- STM32 F103 32 bit核心
- 相容Arduino IDE開發環境
- Keil C / STM32Cube 開發環境
- 多種韌體燒錄方式
- 更多功能腳位，12 bit ADC解析度
- 郵票式電路板和排針雙用模組設計，可使用排針或直接SMT在主板上
- 內含時鐘功能(RTC)

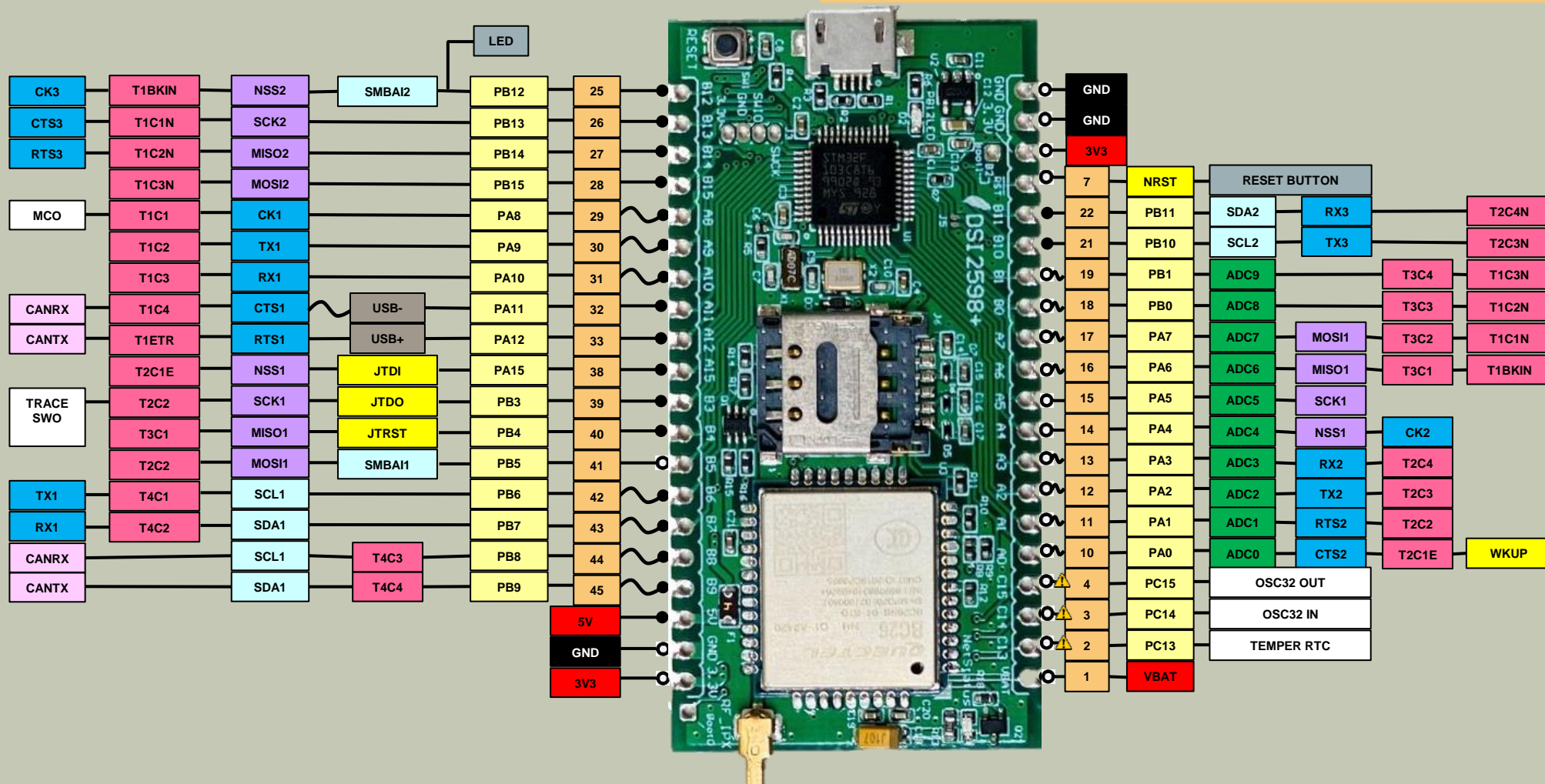


LEGEND
POWER
GROUND
PHYSICAL PIN
PIN NAME
CONTROL
ANALOG
TIMER & CHANNEL
USART
SPI
I2C
CAN BUS
USB
MISC
BOARD HARDWARE
● 5V tolerant
○ Not 5V tolerant
~ PWM pin
— Alternate function
⚠ PC13,PC14,PC15: Sink max 3mA, Source 0mA, Max 2mHz, Max30pF
Absolute MAX 150mA total source/sink for entire CPU
Max ±20mA per pin, ±8mA recommend

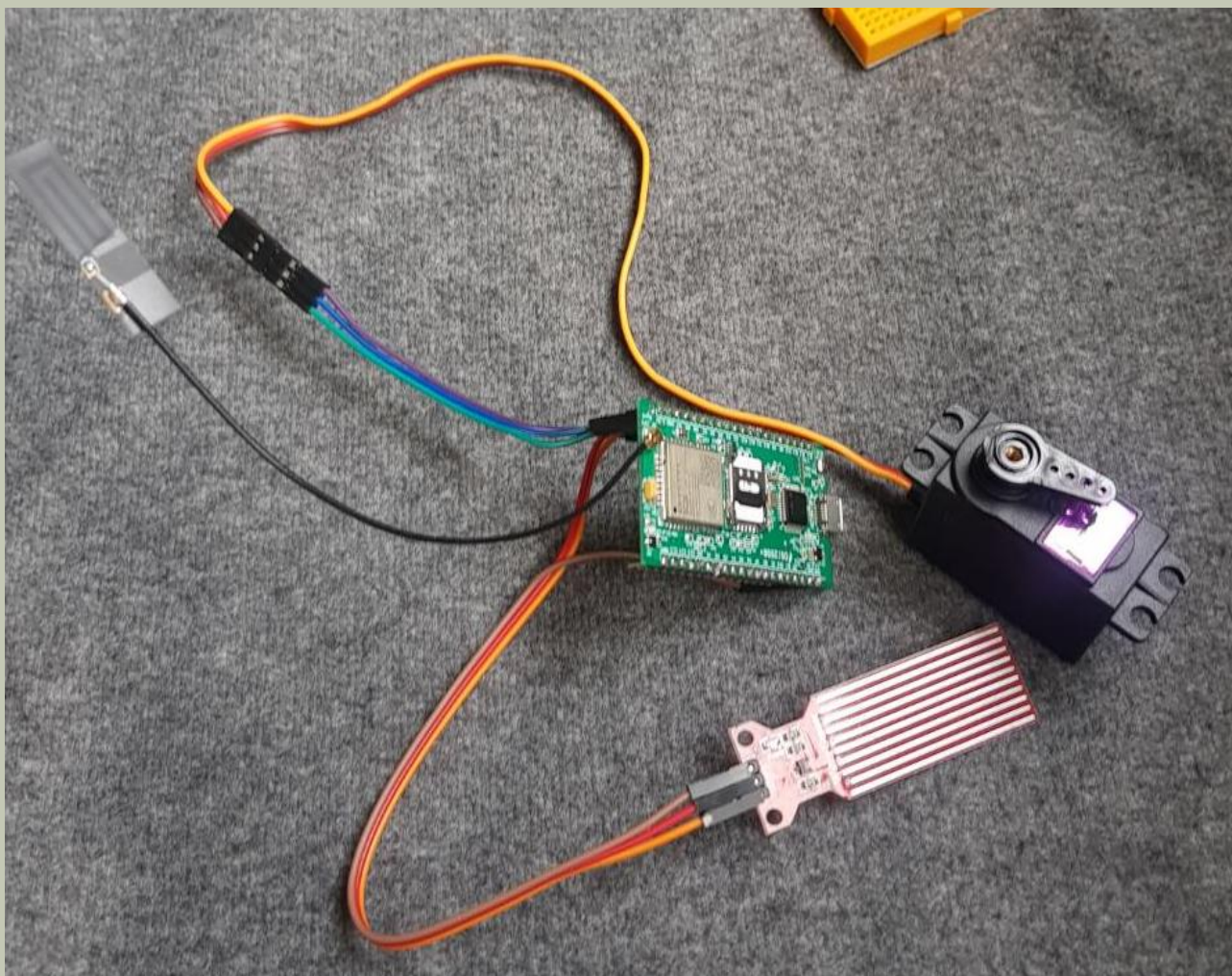


# 2-1-1 腳位介紹

為了控制NBIOT BC26 通訊模組，下面列的PIN，盡量不要與其他功能並用:  
 PA10(Serial1.RXD) <- BC26.TX  
 PA9(Serial1.TXD) -> BC26.RX  
 PC13(OUTPUT) -> BC26.RESET(Low active)



## 2-1-2 接線介紹



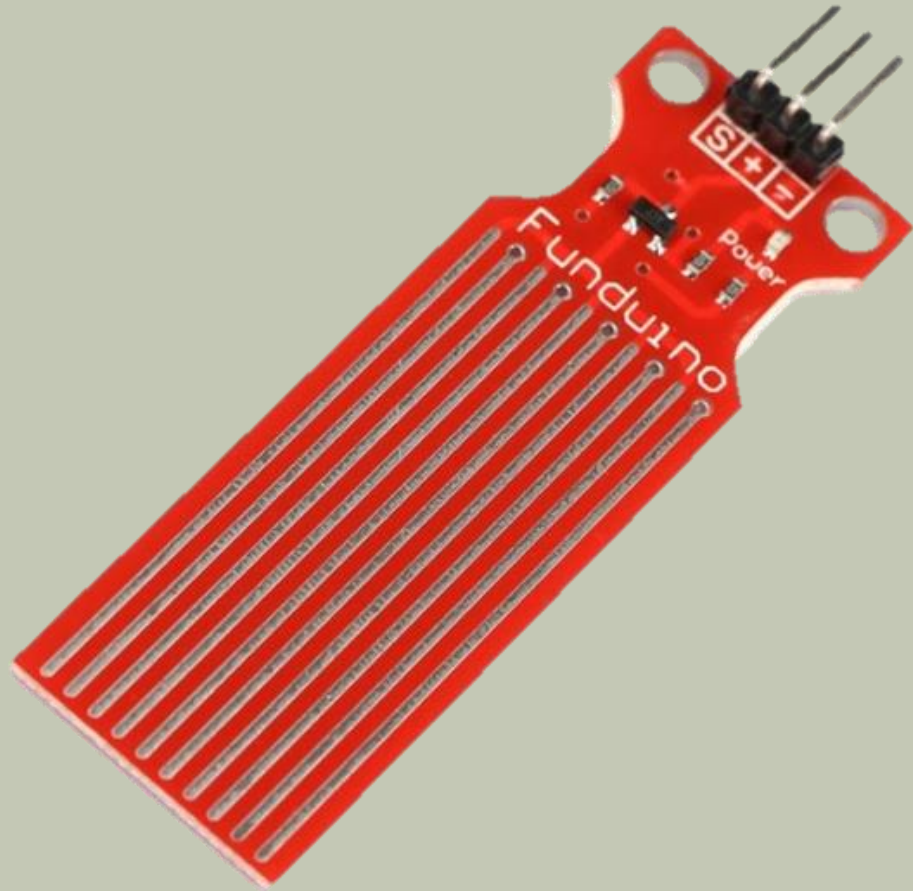
本次範例之連接圖如左

使用元件分別為：

1. DS12598+ x1
2. 水位感測器 x1
3. MG 996R 伺服馬達 x1
4. 排線 x 少許



## 2-2-1 感測器介紹-水位感測器



### 工作資訊：

工作電壓：DC3-5V

工作電流：小於 20mA

檢測面積：40mm x 16mm

工作溫度：10 °C -30 °C

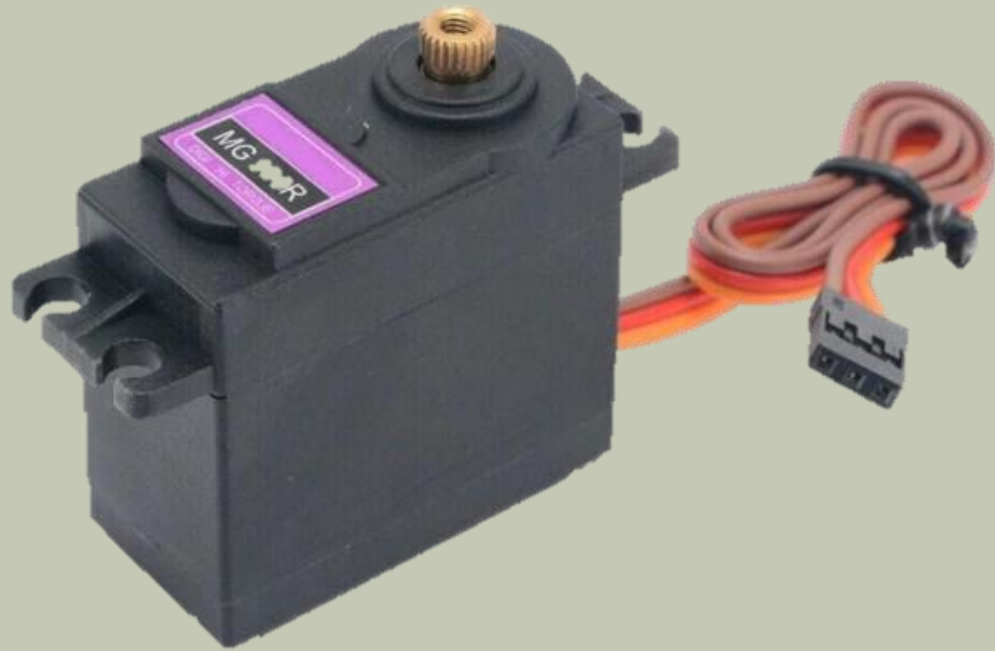
工作濕度：10%-90% 無凝結

重量：3.5g

尺寸：62mm x 20mm x 8mm



## 2-2-2 感測器介紹- MG 996R



### 工作資訊：

優質高速馬達及金屬傳動齒輪，採用雙耐震滾珠軸承

工作電壓：4.8V~7.2V

運行速度：0.17秒/60度(4.8V)；0.14秒/60度(6.0V)

舵機扭矩：

9.4kg\*cm(4.8V)/ 11kg\*cm(6.0V) / 13.5kg\*cm (7.2V)

迴旋角度：180度

重量：55g

紅線: 5V 電源正極

棕線: 地線

橙線: 訊號線

## 第三章

# 軟體介紹

### 3-1 開發環境及APN設定

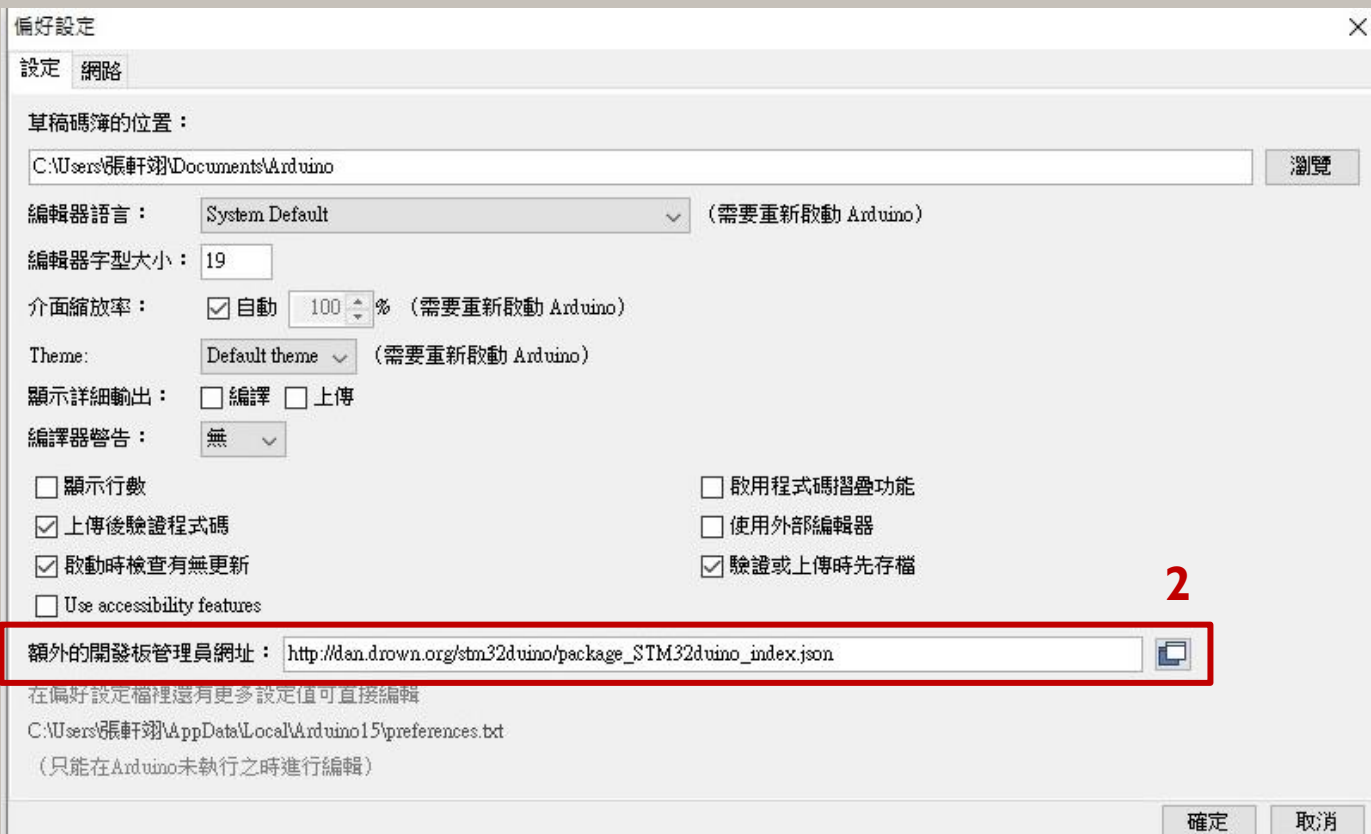
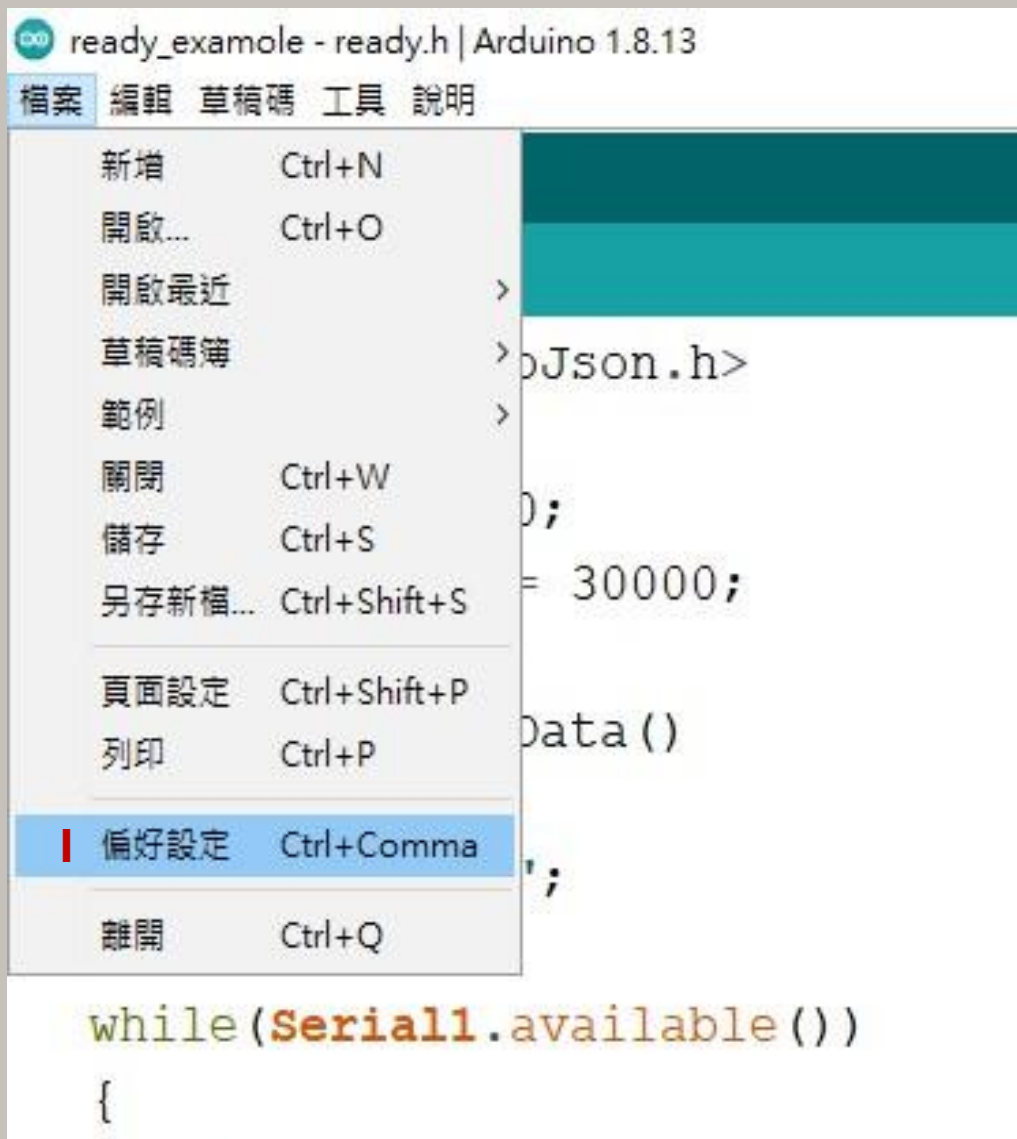
#### 3-1-1 平台設定及配置

#### \*程式介紹

#### 3-2-1 BC26\_init.h

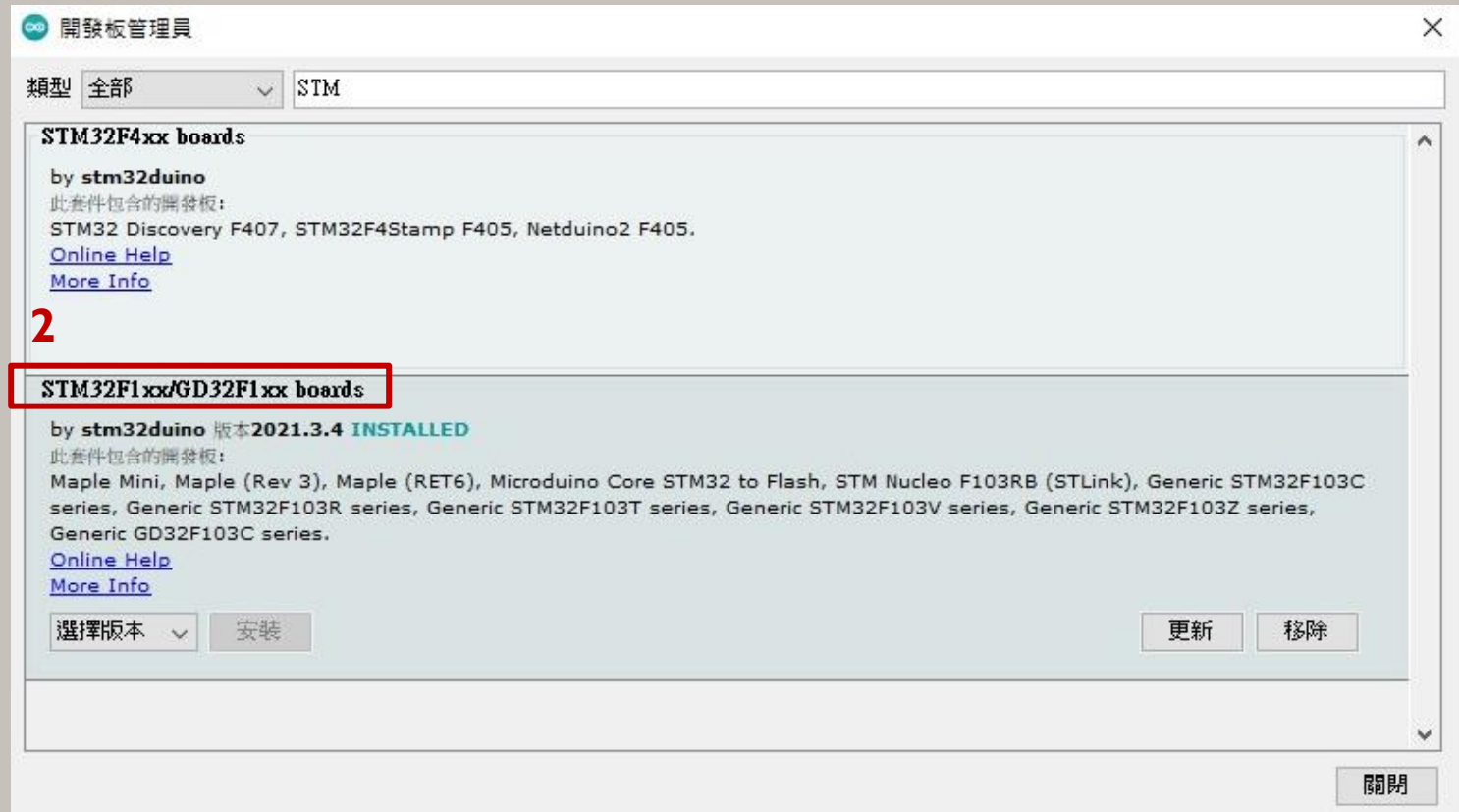
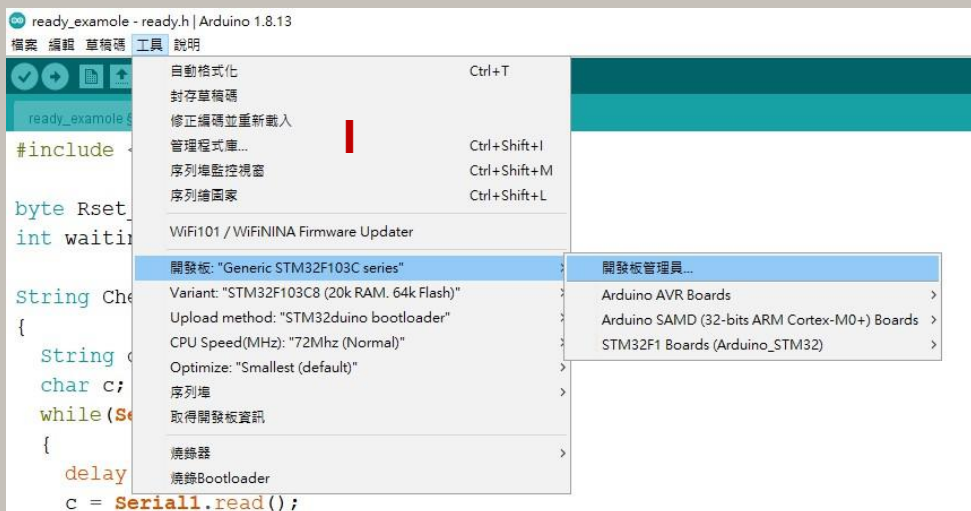
#### 3-2-2 main.ino

# 3-1 開發環境及APN設定



首先從檔案中點選偏好設定後可看到上方之視窗，並在額外增加開發板管理員中輸入以下網址：  
[http://dan.drown.org/stm32duino/package\\_STM32duino\\_index.json](http://dan.drown.org/stm32duino/package_STM32duino_index.json)

# 3-1 開發環境及APN設定



點選工具 -> 開發板 -> 開發板管理員  
以下載開發板資訊



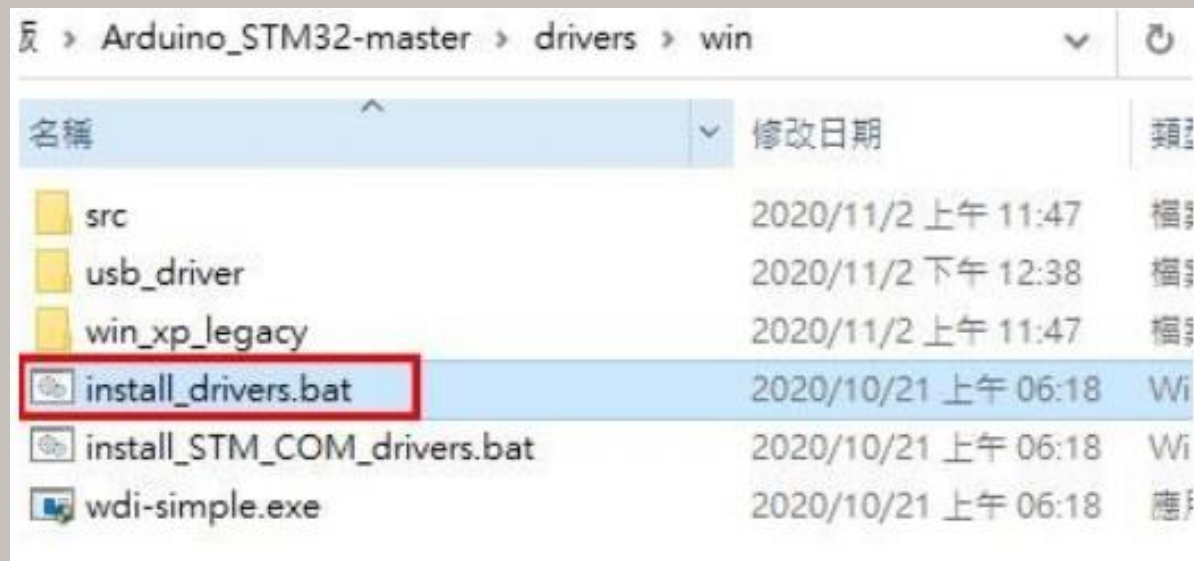
## 3-1 開發環境及APN設定

### Driver安裝(抓不到開發板時安裝)

安裝DFU windows的driver，從以下網址下載：

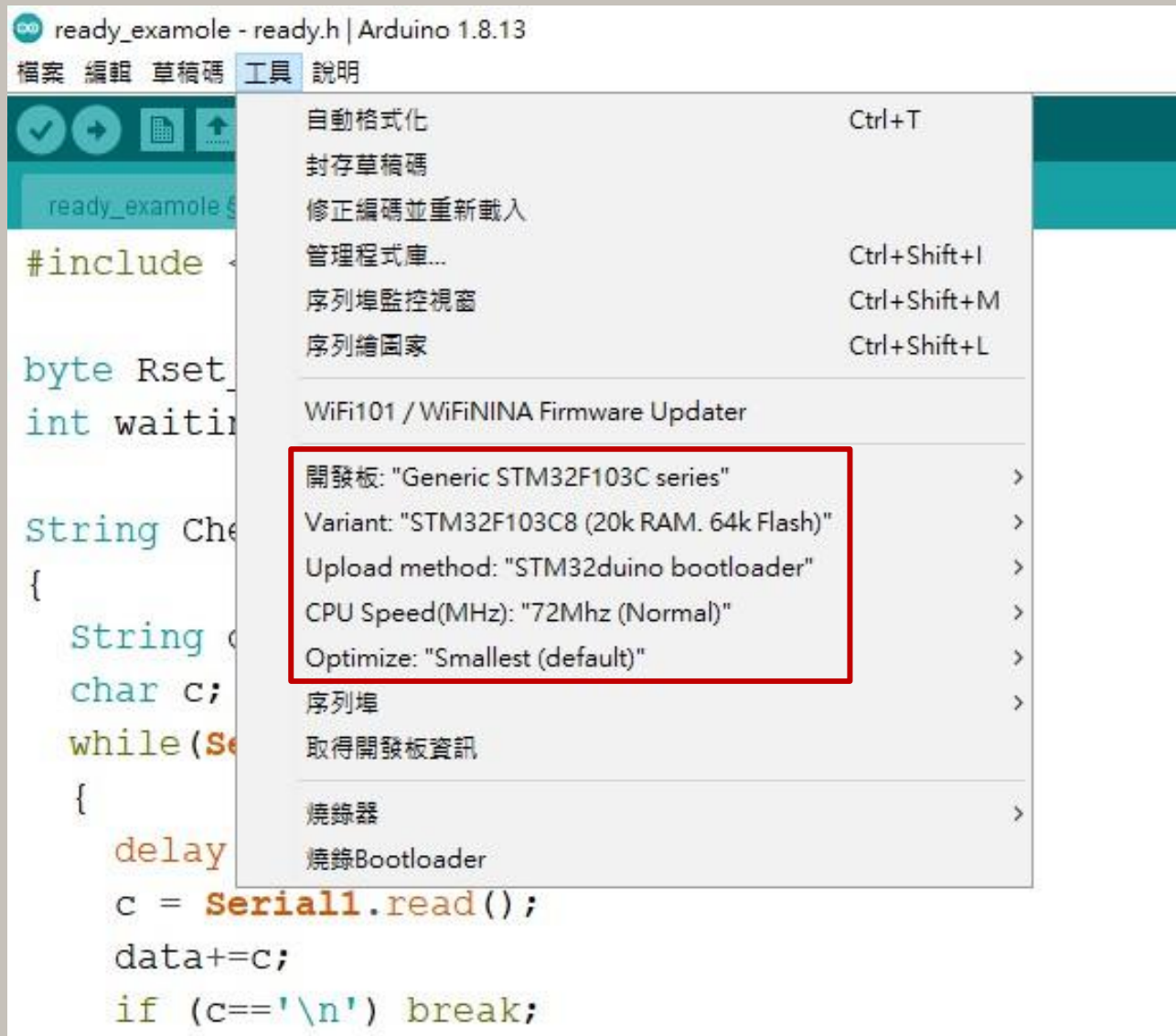
[https://github.com/rogerclarkmelbourne/Arduino\\_STM32](https://github.com/rogerclarkmelbourne/Arduino_STM32)

解開Arduino\_STM32-master.zip 之後，到目錄下Arduino\_STM32-master\drivers\win以系統管理者執行這程式install\_drivers.bat來自動安裝DFU driver。

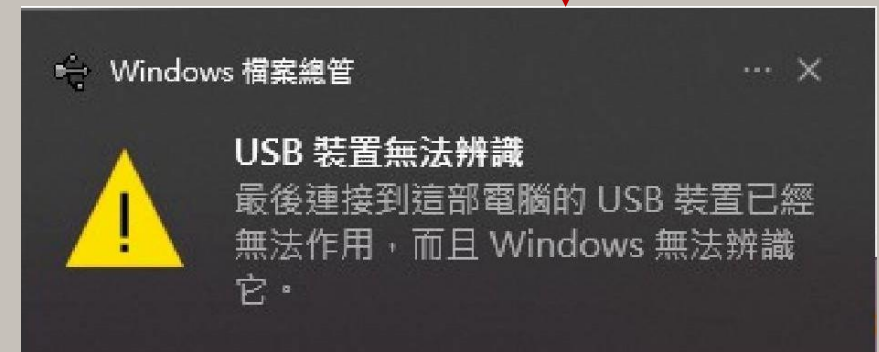




# 3-1 開發環境及APN設定



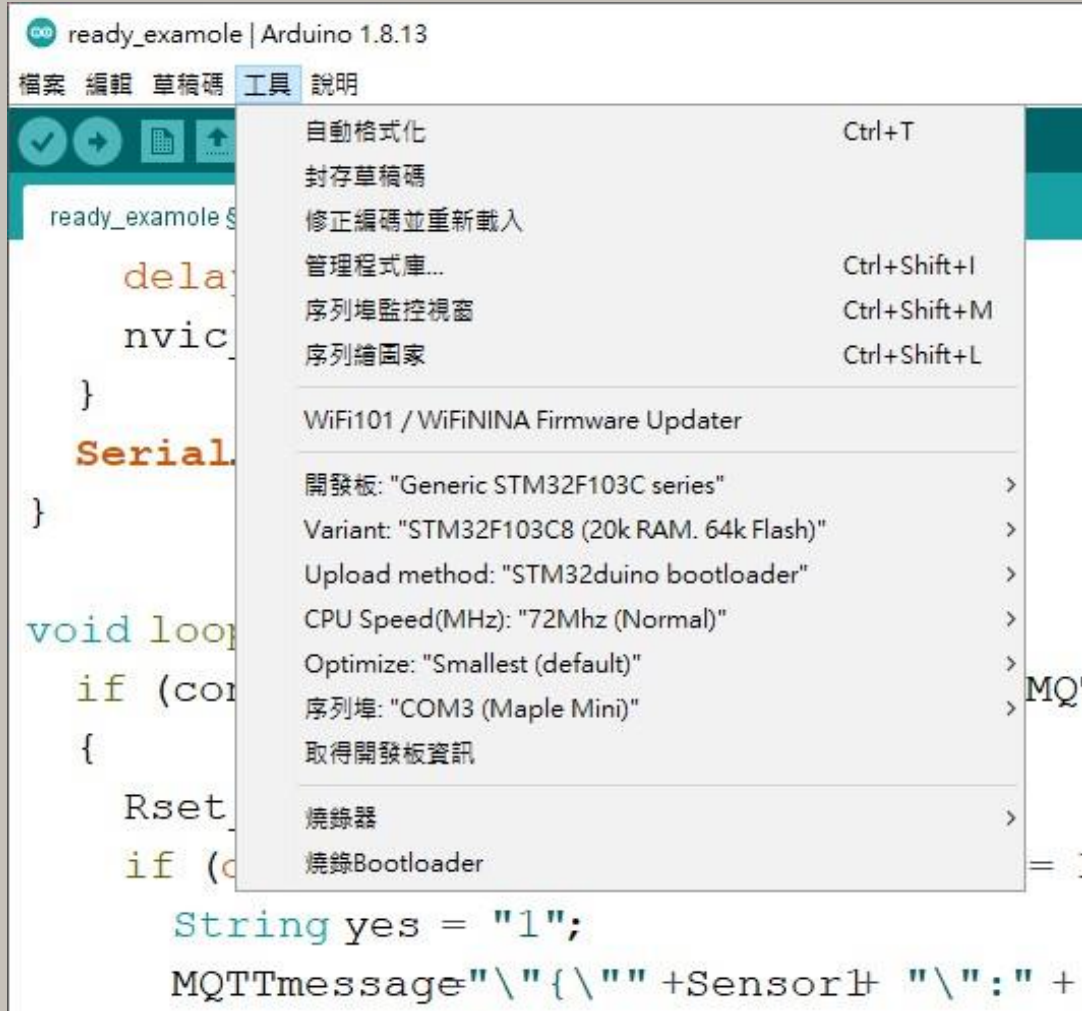
請勿任意更改設定，會導致系統無法取得開發板資訊 (Maple Mini)



# 3-1 開發環境及APN設定

資料來源：資策會

程式碼下載: <https://t.ly/lbip>

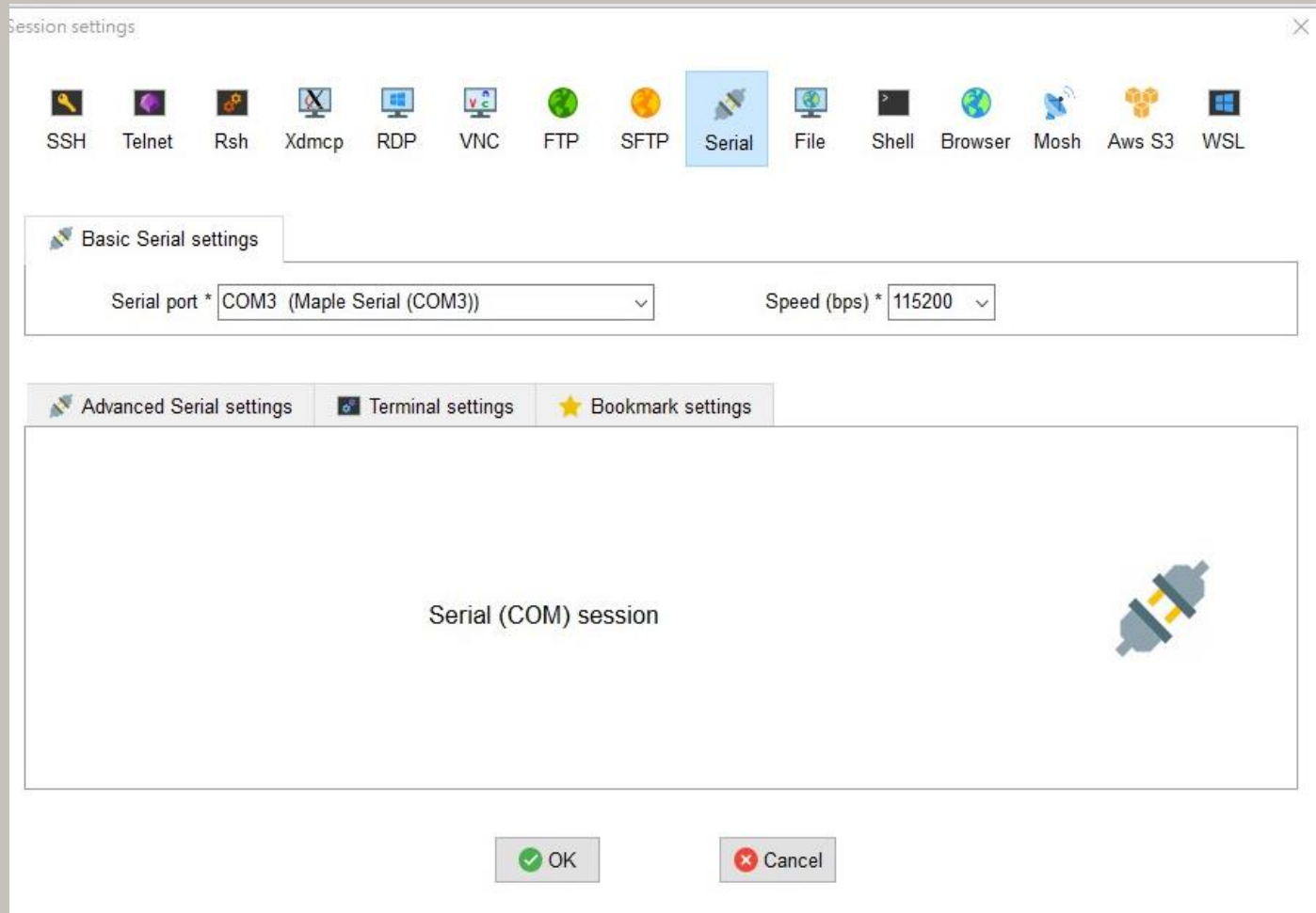


STEP 1：打開main.ino，並至Arduino的工具->序列埠中找到相對應之COM PORT編號。

STEP 2：  
工具 ->開發板->Generic STM32F103C series，  
並且按下上傳 將程式燒錄進去。

# 3-1 開發環境及APN設定

資料來源：資策會



MobaXterm有連接較穩定之優點，因此本操作範例利用MobaXterm來代替Arduino之Serial序列埠監控視窗。

操作上於Serial中選擇對應之port和Speed即可觀測。


# 3-1 開發環境及APN設定

## Home Edition

### Free

Full **X server** and **SSH** support  
Remote desktop (RDP, VNC, Xdmcp)  
Remote terminal (SSH, telnet, rlogin, Mosh)  
X11-Forwarding  
Automatic SFTP browser  
Master password protection  
Plugins support  
Portable and installer versions  
Full documentation  
Max. **12** sessions  
Max. **2** SSH tunnels  
Max. **4** macros  
Max. **360** seconds for Tftp, Nfs and Cron

 Download now

 MobaXterm Home Edition v21.2  
(Installer edition)

MobaXterm之取得方法十分簡單

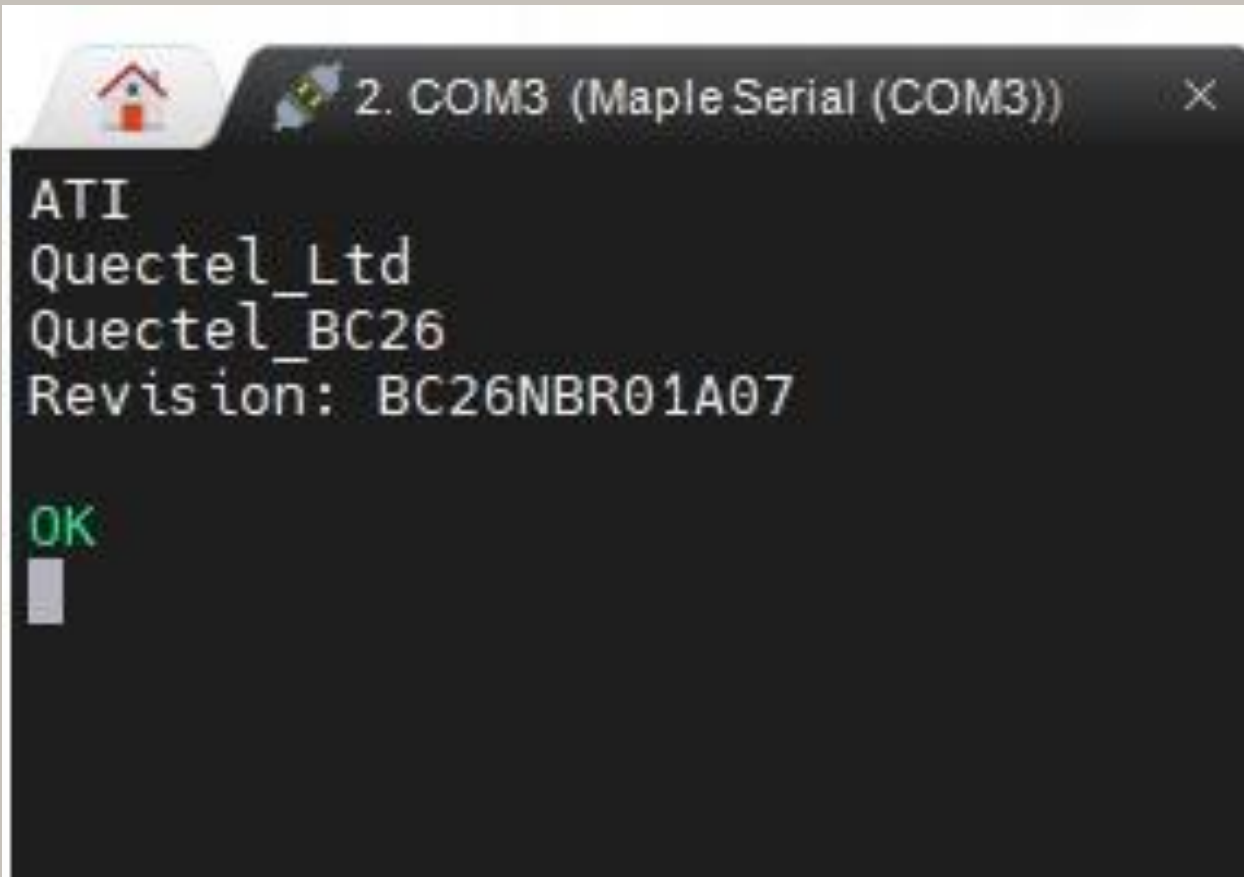
步驟一：搜尋MobaXterm網頁。

步驟二：選擇下載免費版本。

步驟三：點選綠色底部之版本開始下載。

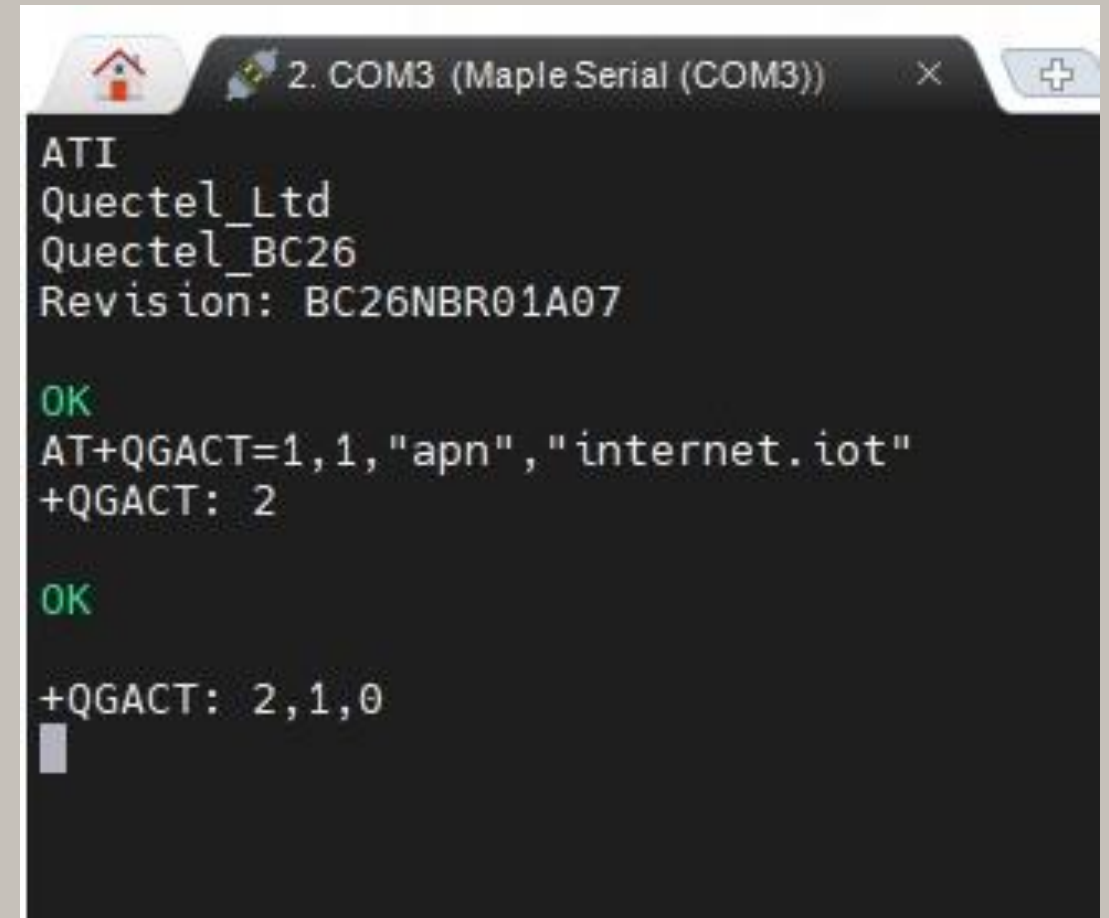
# 3-1 開發環境及APN設定

資料來源：資策會



```
ATI
Quectel_Ltd
Quectel_BC26
Revision: BC26NBR01A07

OK
```



```
ATI
Quectel_Ltd
Quectel_BC26
Revision: BC26NBR01A07

OK
AT+QGACT=1,1,"apn","internet.iot"
+QGACT: 2

OK

+QGACT: 2,1,0
```

STEP 3：開啟MobaXterm，在上方輸入欄中輸入ATI 指令，可先輸入「ATI」，查看模組是否有回覆版本訊息。

STEP 4：啟用APN：AT+QGACT=1,1,"apn","internet.iot"(此為中華電信SIM卡代碼)

# 3-1 開發環境及APN設定

資料來源：資策會

```
ATI
Quectel_Ltd
Quectel_BC26
Revision: BC26NBR01A07

OK
AT+QGACT=1,1,"apn","internet.iot"
+QGACT: 2

OK

+QGACT: 2,1,0
AT+QCGDEFCONT="IP","internet.iot"
OK
█
```

```
ATI
Quectel_Ltd
Quectel_BC26
Revision: BC26NBR01A07

OK
AT+QGACT=1,1,"apn","internet.iot"
+QGACT: 2

OK

+QGACT: 2,1,0
AT+QCGDEFCONT="IP","internet.iot"
OK
AT+QBAND=1,8

OK
█
```

```
AT+QRST=1
F1: 0000 0000
V0: 0000 0000 [0001]
00: 0006 000C
01: 0000 0000
U0: 0000 0001 [0000]
T0: 0000 00B4
Leaving the BROM

F1: 0000 0000
V0: 0000 0000 [0001]
00: 0006 000C
01: 0000 0000
U0: 0000 0001 [0000]
T0: 0000 00B4
Leaving the BROM

RDY

+CFUN: 1

+CPIN: READY

+IP: 10.176.84.80
█
```

STEP 5 : 註冊APN :

AT+QCGDEFCONT="IP","internet.iot"

STEP 6 : 頻寬設定 : AT+QBAND=1,8

STEP 7 : 重新啟動模組 : AT+QRST=1



## 3-1 開發環境及APN設定

資料來源：資策會

### 設定檢查：

STEP 1：Sim卡狀態查詢：AT+CPIN?

回覆：

READY，表示有找到SIM卡回覆

ERROR，表示沒有SIM卡。

STEP 2：APN狀態查詢：AT+CEREG?

回覆：

+CEREG= 0,1，表示已經進入APN的網域。

+CEREG= 0,2，表示尚未註冊APN的網域。

+CEREG= 0,0，表示沒有SIM卡。

```
RDY
+CFUN: 1
+CPIN: READY
+IP: 10.176.84.80
AT+CPIN?
+CPIN: READY

OK
AT+CEREG?
+CEREG: 0,1

OK
```

# 3-1 開發環境及APN設定

資料來源：資策會

STEP 3：訊號強度查詢：  
AT+CESQ 回覆：+CESQ：  
xx, 0, 255, 255, 255 xx：0~99，  
0：未有訊號，99：找不到  
訊號。

STEP 4：IP查詢：  
AT+CGPADDR=1 回覆：OK：  
尚未找到IP 回覆：  
+CGPADDR: 1, IP(四位)：表  
示已有IP說明：若設定期間  
連上網路會自動回傳IP位址  
+IP：IP位址。

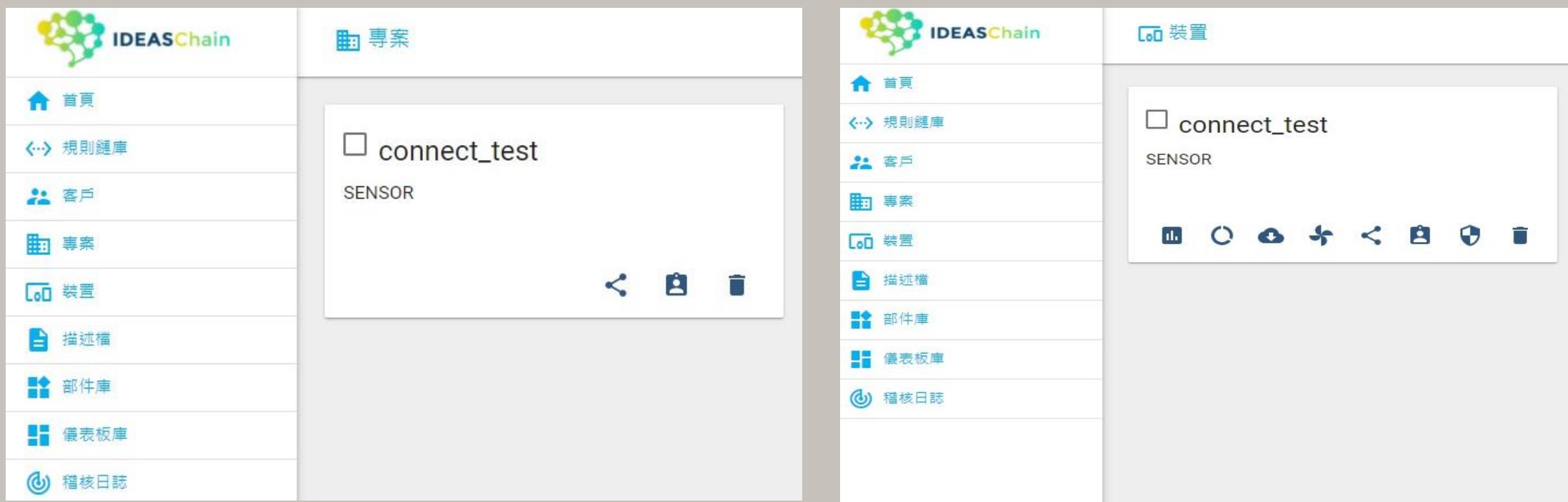
```
+CPIN: READY
+IP: 10.176.84.80
AT+CPIN?
+CPIN: READY
OK
AT+CEREG?
+CEREG: 0,1
OK
AT+CESQ
+CESQ: 23,0,255,255,22,45
OK
```

```
RDY
+CFUN: 1
+CPIN: READY
+IP: 10.176.84.80
AT+CPIN?
+CPIN: READY
OK
AT+CEREG?
+CEREG: 0,1
OK
AT+CESQ
+CESQ: 23,0,255,255,22,45
OK
AT+CGPADDR=1
+CGPADDR: 1,10.176.84.80
OK
```



# 3-1-1 平台設定及配置

IDEAS Chain 網站的平台設定與 API 使用教學 請參考下列網址：  
<https://iforum.ideaschain.com.tw/iforum/devtool/board.do?board=3>



於IDEAS Chain 網站的平台建立專案及裝置

## 3-2-1 程式介紹-BC26\_init.h

```
#include <ArduinoJson.h>

byte Rset_Count=0;           //系統重新啟動計時器宣告
int waitingTime = 30000;    //等候30秒的回覆

String Check_RevData()      //讀取收到的每一字元資料，彙整成一個字串
{
    String data= "";
    char c;
    while(Serial1.available())
    {
        delay(50);
        c = Serial1.read(); //讀取開發板的回應
        data+=c;           //將讀取到的回應存在字串data中
        if (c=='\n') break;
    }
    data.trim();           //將字串中多餘的字串去除，包括空格、Enter、Tab等
    return data;          //返回data字串
}
```

Check\_RevData函式將字元轉字串，使其符合平台接收格式(串列)

## 3-2-1 程式介紹-BC26\_init.h 指令傳送

```
byte Send_ATcommand(String msg, byte stepnum) //傳送AT command,並加以判斷
{
    String Showmsg, C_temp;
    Serial.println(msg);
    Serial1.println(msg);
    Showmsg=Check_RevData();
    //Serial.println(Showmsg); //將開發版的回應顯示於監視窗

    long StartTime=millis();
    switch (stepnum)
    {
        case 0: //重置 BC26
            C_temp = "+IP:";
            break;
        case 1: //其他Data
            C_temp = "OK";
            break;
        case 2: //確認獲得的IP位置
            C_temp = "+CGPADDR:";
            break;
        case 10: //建立MQTT Server
            C_temp = "+QMTOPEN: 0,0";
            break;
        case 11: //以username和password連線MQTT Server
            C_temp = "+QMTCONN: 0,0,0";
    }
}
```

建立指令：使操作者能從指令得知哪階段有問題或是程式進行到哪裡

## 3-2-1 程式介紹-BC26\_init.h 指令傳送

```
case 12:                //Publisher MQTT Data
    C_temp = "+QMTPUB: 0,0,0";
    break;
case 13:                //Subscribe MQTT Data
    C_temp = "+QMTPSUB: 0,1,0,0";
    break;
}
while (!Showmsg.startsWith(C_temp))
{
    Showmsg = Check_RevData();
    if (Showmsg.startsWith("+")) Serial.println(Showmsg);
    if ((StartTime + waitingTime) < millis()) return stepnum;
}
return 99;
}
```

建立指令：  
使操作者能  
從指令得知  
哪階段有問  
題或是程式  
進行到哪裡

## 3-2-1 程式介紹-BC26\_init.h

初始化 DSI2598+ 之 BC26init()

```
bool BC26init()    // 初始化 BC26
{
    Send_ATcommand("AT+QGACT=1,1,\"apn\",\"internet.iot\"", 1); // "internet.iot" 隨基地台不同而不同
    Send_ATcommand("AT+QCGDEFPCOUNT=\"IP\",\"internet.iot\"", 1); // "internet.iot" 隨基地台不同而不同
    Send_ATcommand("AT+QBAND=1,8", 1);
    Send_ATcommand("AT+QRST=1", 0);
    if (Send_ATcommand("ATE0", 1) == 99)
    if (Send_ATcommand("AT+CGPADDR=1", 2) == 99) return true;
    return false;
}
```

- 中華電信NB-IoT服務的APN : "internet.iot"
- 遠傳電信APN : "nbiot"
- 台灣大哥大APN : "twm.nbiot"



## 3-2-1 程式介紹-BC26\_init.h

連線 MQTT Broker connect\_MQTT()

```
bool connect_MQTT(String Serverx, String port, String user, String pass)//建立MQTT連線通道
{
    String S_temp;
    S_temp = "\"" + Serverx + "\" + \",\" + port;
    S_temp = "AT+QMTOPEN=0," + S_temp;
    if (Send_ATcommand(S_temp, 10) != 99) return false;
    S_temp = "\"" + user + "\" + \",\" + "\"" + pass + "\"";
    S_temp = "AT+QMTCONN=0,0," + S_temp;
    if (Send_ATcommand(S_temp, 11) != 99) return false;
    return true;
}
```

## 3-2-1 程式介紹-BC26\_init.h

```
bool Publish_MQTT(String topic, String message)//發布資料
{
    String S_temp;
    S_temp = "\"" + topic + "\"" + "," + message;
    S_temp = "AT+QMTPUB=0,0,0,0,"+S_temp;
    if (Send_ATcommand(S_temp,12) !=99) return false;
    return true;
}
```

發布資料 Publish\_MQTT ()  
將想要的資料傳至平台

```
bool Sub_MQTT(String topic)//訂閱資料
{
    String S_temp;
    S_temp="\""+topic+"\"+\",\"+\"0\";
    S_temp="AT+QMTPSUB=0,1,\""+S_temp;
    if(Send_ATcommand(S_temp,13) !=99) return false;
    return true;
}
```

訂閱資料 Sub\_MQTT () 函式  
從平台上獲得資料



## 3-2-1 程式介紹-BC26\_init.h

---

關閉與 MQTT Broker 的連線 Close\_MQTT()

```
bool Close_MQTT() //關閉連線
{
    String S_temp;
    S_temp="AT+QMTCLOSE=0";
    if(Send_ATcommand(S_temp,1)!=99) return false;
    return true;
}
```





## 3-2-1 程式介紹-BC26\_init.h

將讀到 IDEAS Chain 網站的屬性資料 JSON 格式，進行分解取出 JSON\_DEC\_data()

```
String JSON_DEC_data(String input, String findData)//將包含JSON格式的資料取出
{
    int index=input.indexOf(',');
    int x=input.substring(0, index).toInt();
    String json=input.substring(index+1,input.length());
    index=json.indexOf(':');
    x=json.substring(0,index).toInt();
    json=json.substring(index+1,json.length());
    DynamicJsonDocument doc(1024);
    deserializeJson(doc, json);
    JsonObject obj=doc.as<JsonObject>();
    return obj[findData];
}
```

MQTT需使用JSON  
格式進行傳輸



## 3-2-1 程式介紹-BC26\_init.h

取出 IDEAS Chain 網站資料時，必須先執行訂閱的動作 Sub\_Ideaschain()

```
bool Sub_Ideaschain(String attrestopic)//訂閱Ideaschain屬性資料
{
    String S_temp;
    S_temp = "\"" + attrestopic + "\" + "," + "0";
    S_temp = "AT+QMTSUB=0,1," + S_temp;
    Serial.println(S_temp);
    Serial1.println(S_temp);
    delay (2000);
    return true;
}
```

```

String Get_Publish_MQTT(byte mode, String attreqtopic, String message)//取得Ideaschain屬性資料
{
  String Showmsg;
  String S_temp, T_temp;
  if (mode==0) T_temp="sharedKeys";
  if (mode==1) T_temp="clientKeys";
  S_temp="\\"+attreqtopic+"\\"+", "+\\"{"+T_temp+"\\":\\""+message+"\\"}\\"";
  S_temp="AT+QMQTPUB=0,0,0,0,"+S_temp;
  Serial.println(S_temp);
  Serial1.println(S_temp);
  Showmsg=Check_RevData();
  long StartTime=millis();
  while(!Showmsg.startsWith("+QMQTRECIV:"))
  {
    delay(1000);
    Showmsg=Check_RevData();
    if(Showmsg.length()>30) break;
    if((StartTime+waitingTime)<millis()) return "error";
  }
  return JSON_DEC_data (Showmsg,message);
}

```

訂閱後所取得的  
 IDEAS Chain 屬性資料  
 Get\_Publish\_MQTT()

\*本範例不會使用

## 3-2-2 程式介紹- main.ino

```
#include "BC26_init.h"
#include <Servo.h>           //Arduino內建           內建函式庫，用於控制伺服馬達

#define Grove_Water_Sensor PB9      //水位感測器腳位設定
int waterVal;

Servo myservo;
int theta;

String MQTT_Server="iiot.ideaschain.com.tw"; //MQTT Server的IP位址
String MQTT_Port="1883"; //MQTT使用的port (固定的勿更動)
String MQTT_Access_token="qQBgVSaCxeVBsUNdKi4w"; //使用者密碼-->Ideaschain裝置的存取權杖
String MQTTtopic="v1/devices/me/telemetry"; //Ideaschain固定路徑
String MQTTmessage=""; //發佈到Ideaschain的字串

String Sensor1="flooded"; //平台接收的變數名稱
```

權杖可由IDEAS Chain之平台裝置獲取



## 3-2-2 程式介紹- main.ino

```
void setup() {  
    Serial.begin(115200);  
    Serial1.begin(115200);  
    pinMode(Grove_Water_Sensor, INPUT);           //水位感測器腳位宣告  
    myservo.attach(PB8);                          //MG996R伺服馬達腳位設定  
  
    if(!BC26init())                               //引用BC26初始化函式  
    {  
        delay(30000);  
        nvic_sys_reset();  
    }  
    Serial.println("初始化完成....");  
}
```

## 3-2-2 code introduction- main.ino

```
void loop() {
  if (connect_MQTT(MQTT_Server,MQTT_Port,MQTT_Access_token,MQTT_Access_token))//以MQTT連線Ideaschain平台
  {
    Rset_Count++;
    if (digitalRead(Grove_Water_Sensor) == 1){ //讀取水位感測器的資訊
      String yes = "1";
      MQTTmessage="\{" + Sensor1 + "\":" + yes + "\}"; //設定發佈資料
      myservo.write(0);
      delay(500);
    }else{
      String no = "0";
      MQTTmessage="\{" + Sensor1 + "\":" + no + "\}"; //設定發佈資料
      myservo.write(180);
      delay(500);
    }
    Publish_MQTT(MQTTtopic,MQTTmessage); //引用發佈函示以發佈資料

    Close_MQTT();
    delay(500);
    if (Rset_Count>20) {
      delay(10000);nvic_sys_reset();
    }
  }
}
```

定義淹水與否時對應之感測器訊號

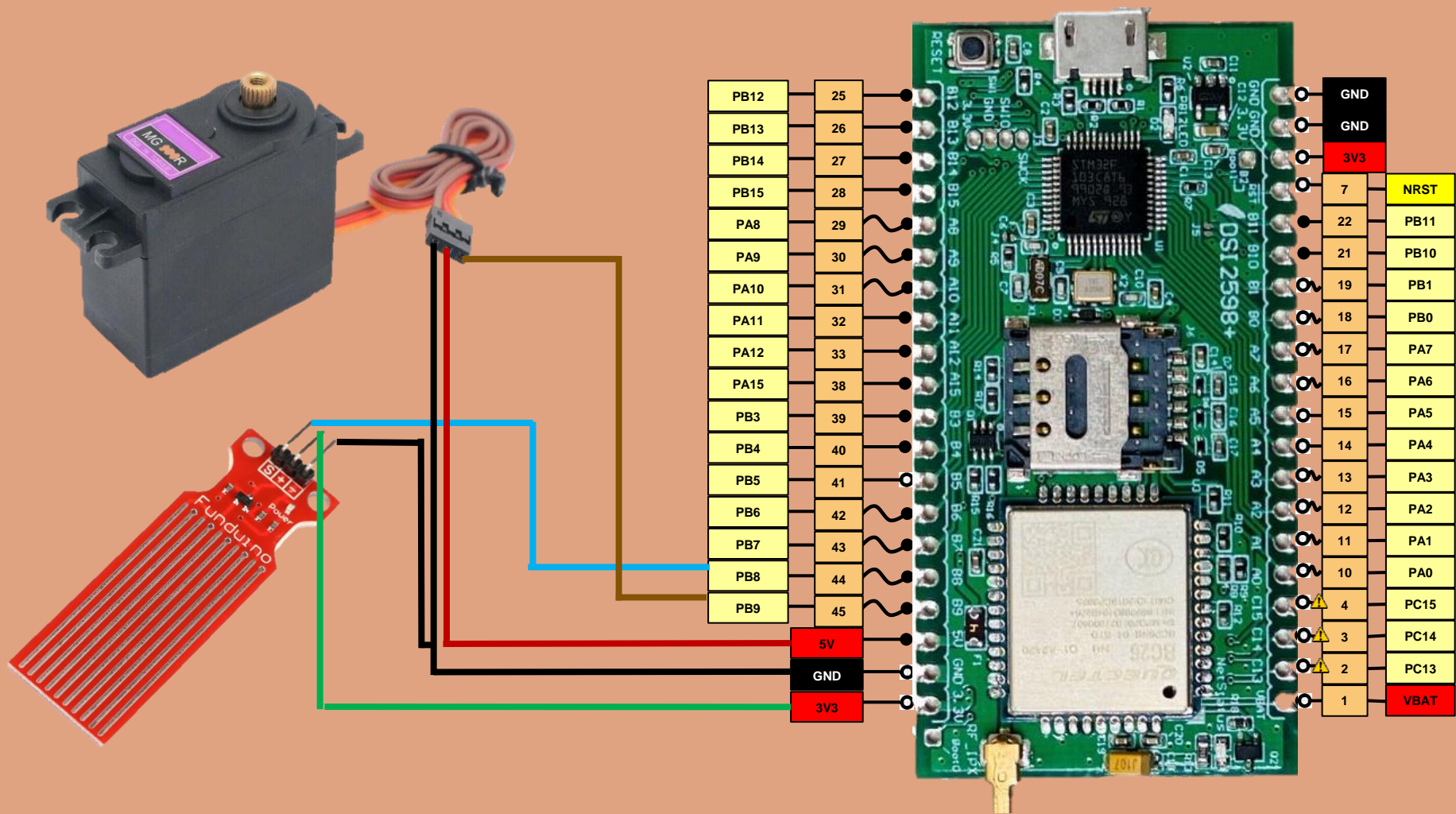
0表示無偵測到淹水

1表示偵測到淹水

## 第四章

# 成果展示

# 電路圖





# 訊號顯示成果

MobaXterm之訊號顯示結果：

“flooded”：0 表示無淹水

“flooded”：1 表示偵測到淹水

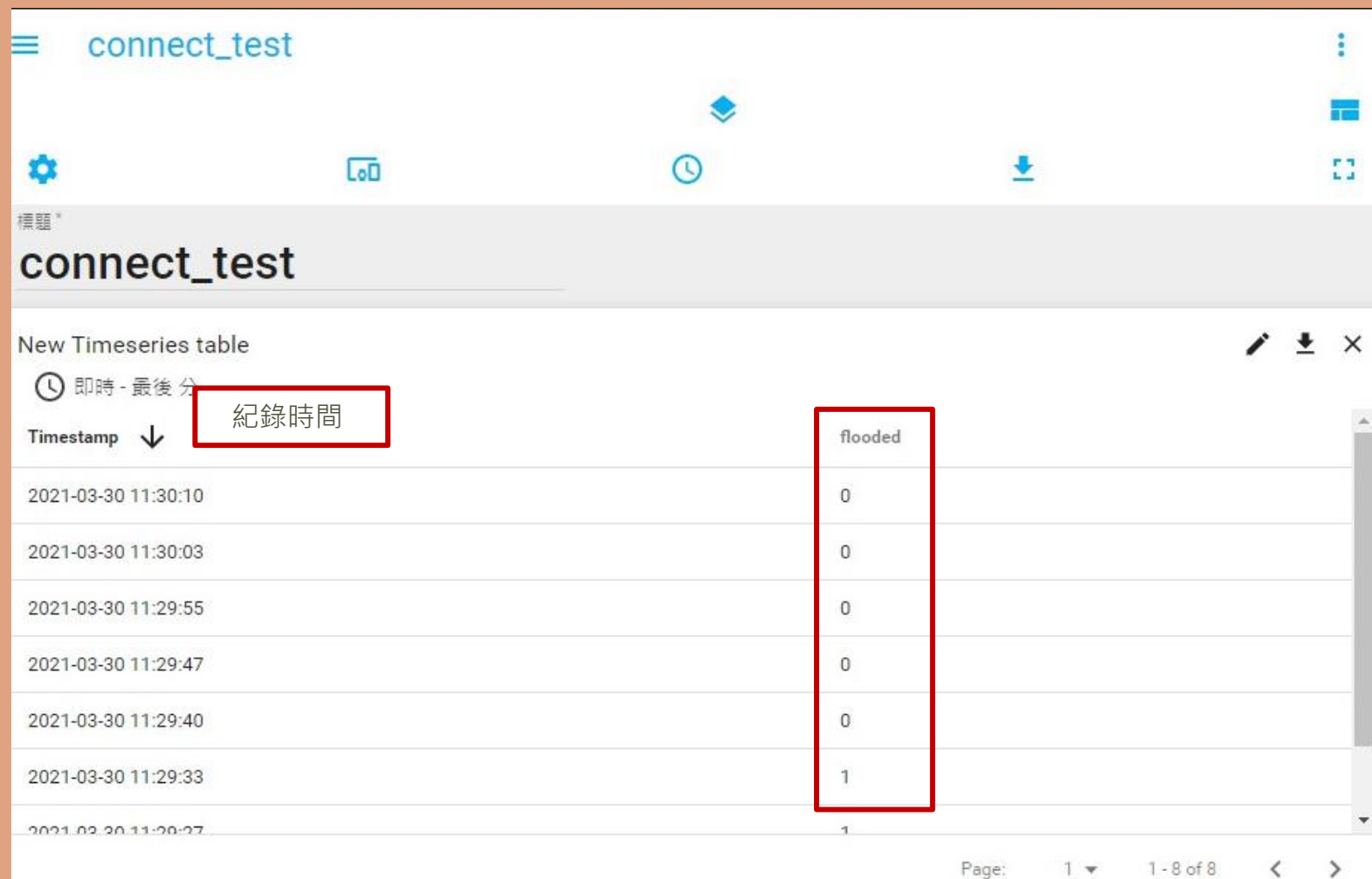
```
2. COM3 (Maple Serial (COM3))
+QGACT: 1
AT+QCGDEFPCOUNT="IP","internet.iot"
+QGACT: 1,1,0
+IP: 10.211.179.58
AT+QBAND=1,8
AT+QRST=1
+CFUN: 1
+CPIN: READY
+IP: 10.211.193.185
ATE0
AT+CGPADDR=1
+CGPADDR: 1,10.211.193.185
初始化完成...
AT+QMTOPEN=0,"iiot.ideaschain.com.tw",1883
+QMTOPEN: 0,0
AT+QMTCONN=0,0,"qQBgVSaCxeVBsUNdKi4w","qQBgVSaCxeVBsUNdKi4w"
+QMTCONN: 0,0,0
AT+QMT PUB=0,0,0,0,"v1/devices/me/telemetry",{"flooded":0}
+QMT PUB: 0,0,0
AT+QMTCLOSE=0
AT+QMTOPEN=0,"iiot.ideaschain.com.tw",1883
+QMTCLOSE: 0,0
+QMTOPEN: 0,0
AT+QMTCONN=0,0,"qQBgVSaCxeVBsUNdKi4w","qQBgVSaCxeVBsUNdKi4w"
+QMTCONN: 0,0,0
AT+QMT PUB=0,0,0,0,"v1/devices/me/telemetry",{"flooded":1}
+QMT PUB: 0,0,0
AT+QMTCLOSE=0
AT+QMTOPEN=0,"iiot.ideaschain.com.tw",1883
+QMTCLOSE: 0,0
+QMTOPEN: 0,0
AT+QMTCONN=0,0,"qQBgVSaCxeVBsUNdKi4w","qQBgVSaCxeVBsUNdKi4w"
+QMTCONN: 0,0,0
AT+QMT PUB=0,0,0,0,"v1/devices/me/telemetry",{"flooded":1}
+QMT PUB: 0,0,0
```

# 平台成果

IDEAS Chain平台之訊號顯示結果：

0 表示無淹水

1 表示偵測到淹水



The screenshot displays a web interface for a 'connect\_test' table. The table is titled 'New Timeseries table' and shows a list of data points. The 'Timestamp' column is highlighted with a red box and labeled '紀錄時間'. The 'flooded' column is also highlighted with a red box and contains values 0 or 1. The data points are as follows:

Timestamp	flooded
2021-03-30 11:30:10	0
2021-03-30 11:30:03	0
2021-03-30 11:29:55	0
2021-03-30 11:29:47	0
2021-03-30 11:29:40	0
2021-03-30 11:29:33	1
2021-03-30 11:29:27	1

# 結果示意圖

