



物聯網智造基地

I O T S E R V I C E H U B

NB-IoT DSI2598+ 開發板講解

中華電信IoT智慧聯網大平台網站 <https://iot.cht.com.tw/iot/index>

Arduino IDE 1.8.12 for Windows : <https://t.ly/Ccwp>

1. DSI2598+ 開發板基礎介紹
2. APN (Access Point Name) 設定
3. 中華電信 IoT 網站概略設定
4. HTTP AT_Command 概略說明
5. HTTP 程式(函式)使用說明
6. 溫溼度範例使用說明及線路圖

請注意：由於 NB-IoT 當初就是以小資料傳輸為設計原則，因此並不適合HTTP 方式的傳輸 !!

NB-IoT :窄帶物聯網(Narrow Band Internet of Things, NB-IoT)

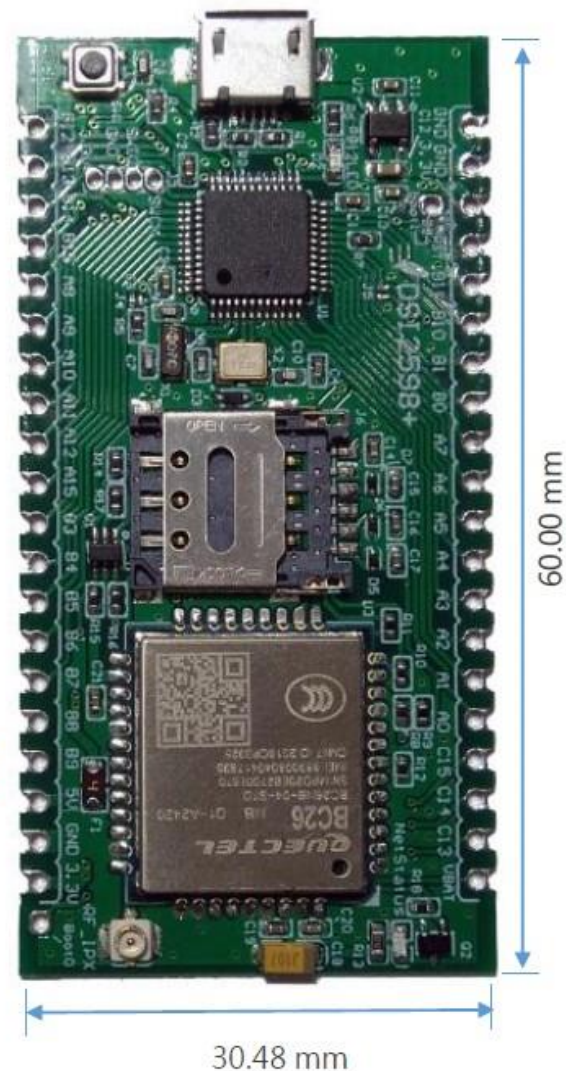
1. 構建於蜂窩網絡，只消耗大約180KHz的帶寬，可直接部署於GSM網絡、UMTS網絡或LTE網絡。
2. 是IoT領域一個新興的技術，支持低功耗設備在廣域網的蜂窩數據連接，也被叫作低功耗廣域網(LPWAN)。
3. 待機時間長、設備電池壽命提高至少5年以上。
4. 可透過各大電信業者提供的 NB-IoT / SIM 卡，利用電信基地台連到網際網路。
5. 其特性可增加覆蓋範圍提升 20dB，使原本透過 4G LTE網路收不到的地方(如地下室、地下管道等)也能收到訊號。

NB-IoT 與 WiFi 之差異：

1. WiFi 透過無線基地台連上網際網路，NB-IoT 利用電信基地台連上網際網路。
2. WiFi 適用傳輸大量資料的訊息，NB-IoT 適用小資料量傳輸。
3. WiFi 連接無線基地台的距離較短，NB-IoT 由於全台基地台涵蓋率夠高，幾乎無死角。
4. WiFi 晶片耗用功率較高，NB-IoT 採用低功率晶片，使用一般 AA 電池可達 3-5 年以上。
5. WiFi 連網較易取得真實IP 位址，而NB-IoT使用的電信基地台提供的IP 位址大都為 虛擬 IP 網段。

DSI 2598+

- 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
- 郵票式電路板設計

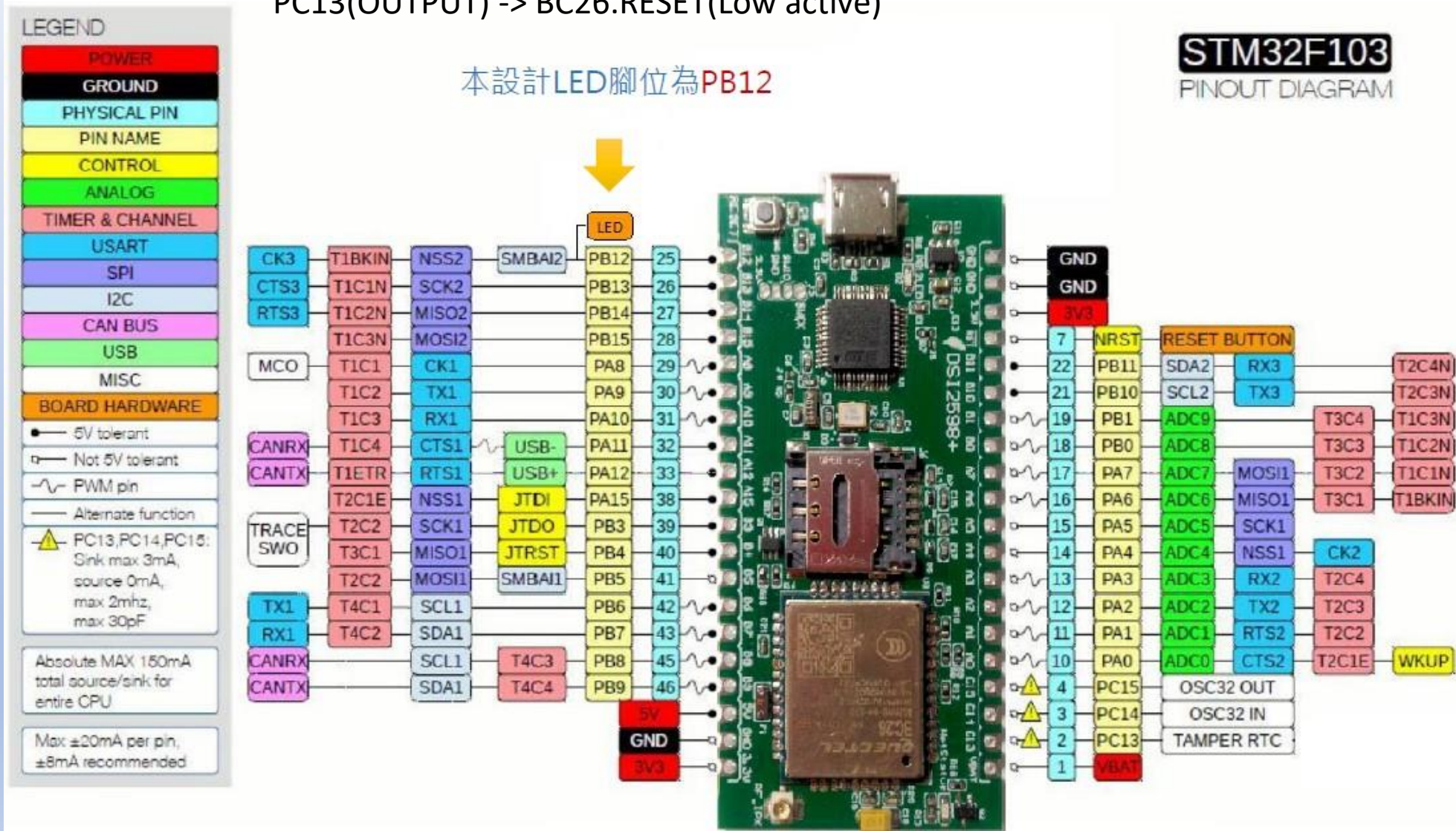


為了控制NB-IOT BC26 通訊模組，下面列的PIN，盡量不要與其他功能並用：

PA10(Serial1.RXD) <- BC26.TX

PA9(Serial1.TXD) -> BC26.RX

PC13(OUTPUT) -> BC26.RESET(Low active)



設定Arduino DSI2598+開發板的環境: (for Windows 10 作業系統)

請參考

<https://iforum.ideaschain.com.tw/iforum/devtool/board.do?board=2>

說明手冊

或參考 <https://iforum.ideaschain.com.tw/iforum/techmatch/solution.do?solution=34>

之說明手冊內容

確認設定與以下畫面相同：



請注意：若之後任意更改設定，有可能讓板子啟動方式毀損，導致系統再也找不到該序列埠 (Maple Mini) !!

APN 設定

程式碼下載: <https://t.ly/wN5u>

資料來源：資策會

STEP 1：打開DSI2598P_ATcommand.ino，並至Arduino的工具->序列埠中找到USB
模組的COM PORT編號，可至控制台確認。

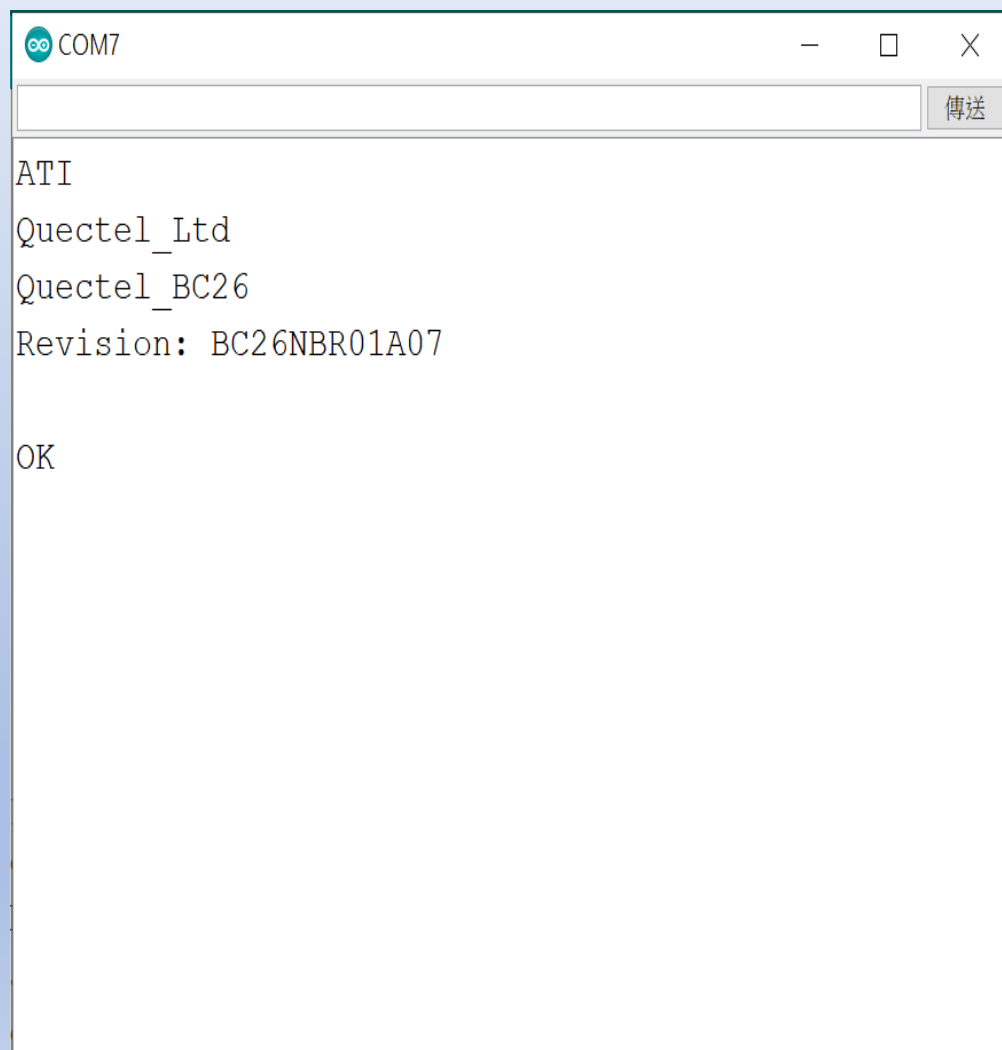
STEP 2：選擇 工具 ->開發板->Generic STM32F103C series，然後按下 上傳 (Ctrl+U)，將
程式燒錄進去。



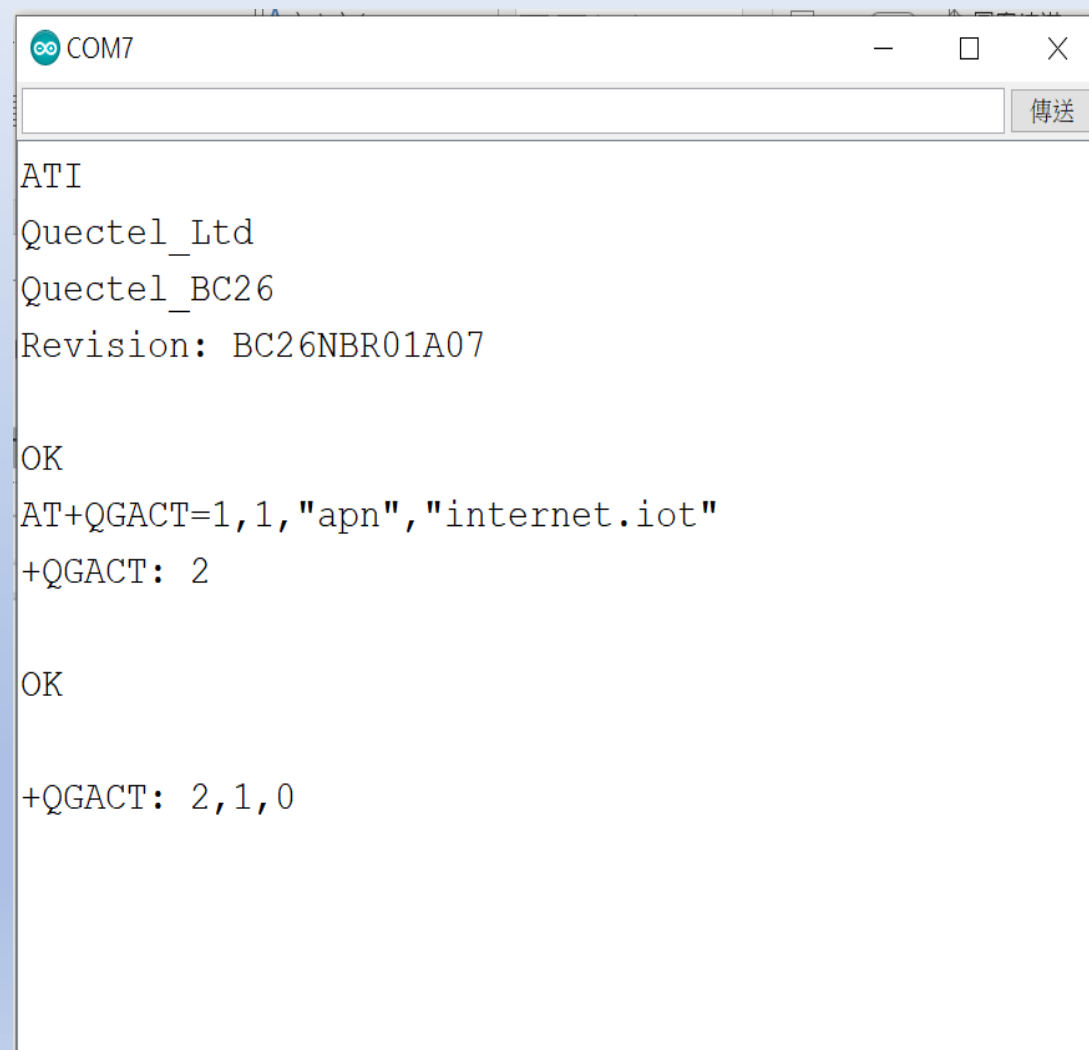
STEP 3：打開 序列埠監控視窗，在上方輸入欄中輸入**ATI** 指令，可先輸入「**ATI**」，查看模組是否有回覆版本訊息。

資料來源：資策會

STEP 4：啟用APN：**AT+QGACT=1,1,"apn","internet.iot"**



```
COM7
ATI
Quectel_Ltd
Quectel_BC26
Revision: BC26NBR01A07
OK
```



```
COM7
AT+QGACT=1,1,"apn","internet.iot"
+QGACT: 2
OK
+QGACT: 2,1,0
```

STEP 5：註冊APN：AT+QCGDEFCONT="IP","internet.iot"

資料來源：資策會

STEP 6：頻寬設定：AT+QBAND=1,8

STEP 7：重新啟動模組：AT+QRST=1

2019年 台灣頻段支援表						
電信\頻段	FDD - LTE 700MHz(B28)	FDD - LTE 900MHz(B8)	FDD - LTE 1800MHz(B3)	FDD - LTE 2100MHz(B1)	FDD - LTE 2600MHz(B7)	TDD - LTE 2600(B38/B41)
中華電信		○	○	○	○	
遠傳電信	○		○	○	○	○
台灣大哥大	○		○	○		
台灣之星		○		○	○	
亞太	○	◎ (通話限定)				○

```
COM7
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
```

```
COM7
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
```

```
COM7
OK
AT+QBAND=1,8

OK
AT+QRST=1
RbRQBBER ?tY?
RbRQBBER ?tY?

RDY

+CFUN: 1

+CPIN: READY

+IP: 10.85.230.245
```

設定檢查

資料來源：資策會

STEP 1：

Sim卡狀態查詢：AT+CPIN?

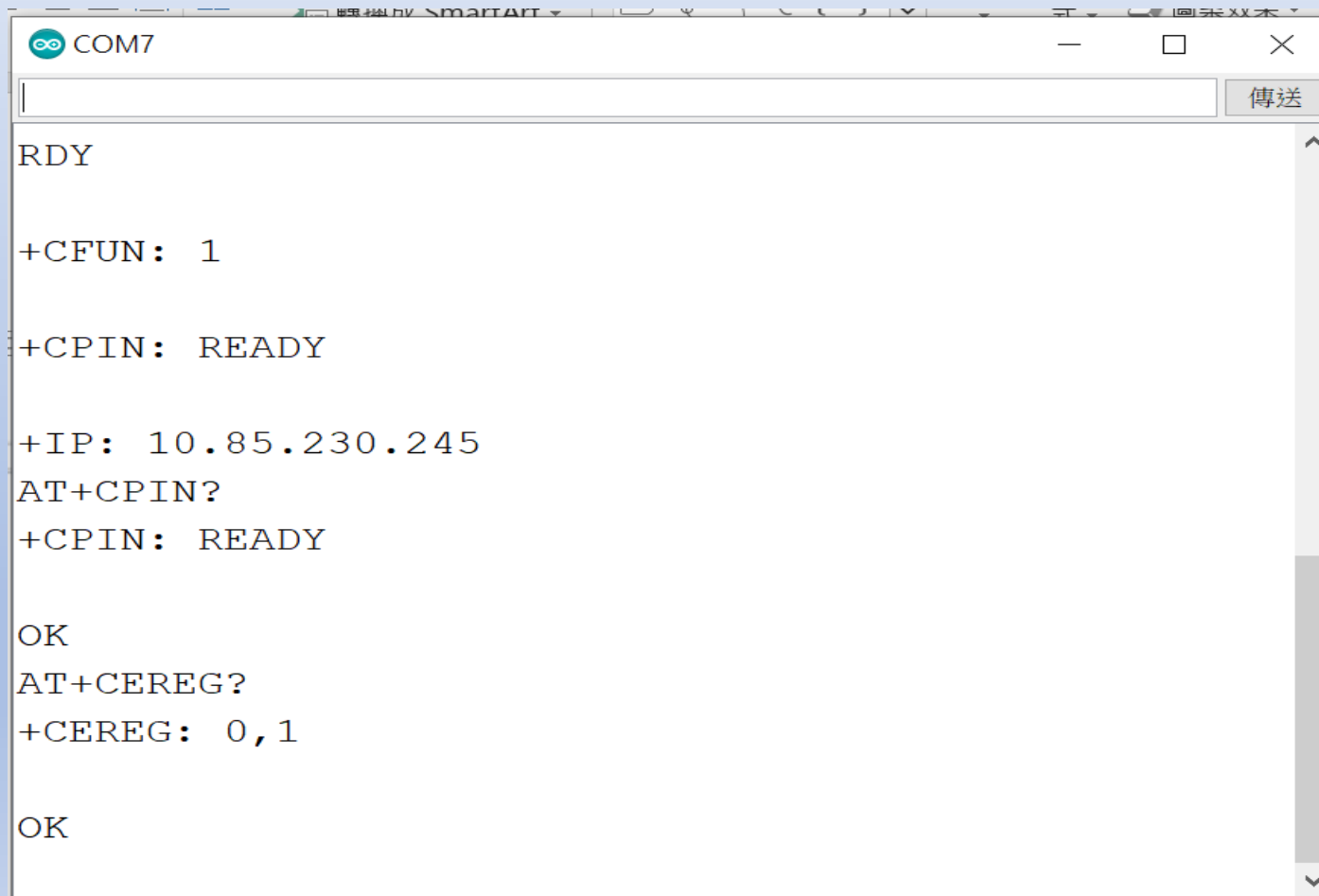
回覆：READY，表示有找到SIM卡
回覆：ERROR，表示沒有SIM卡

STEP 2：

APN狀態查詢：AT+CEREG?

回覆：+CEREG= 0,1，表示已經進入APN的網域。

回覆：+CEREG= 0,2，表示已經尚未註冊入APN的網域。
回覆：+CEREG= 0,0，表示沒有SIM卡。



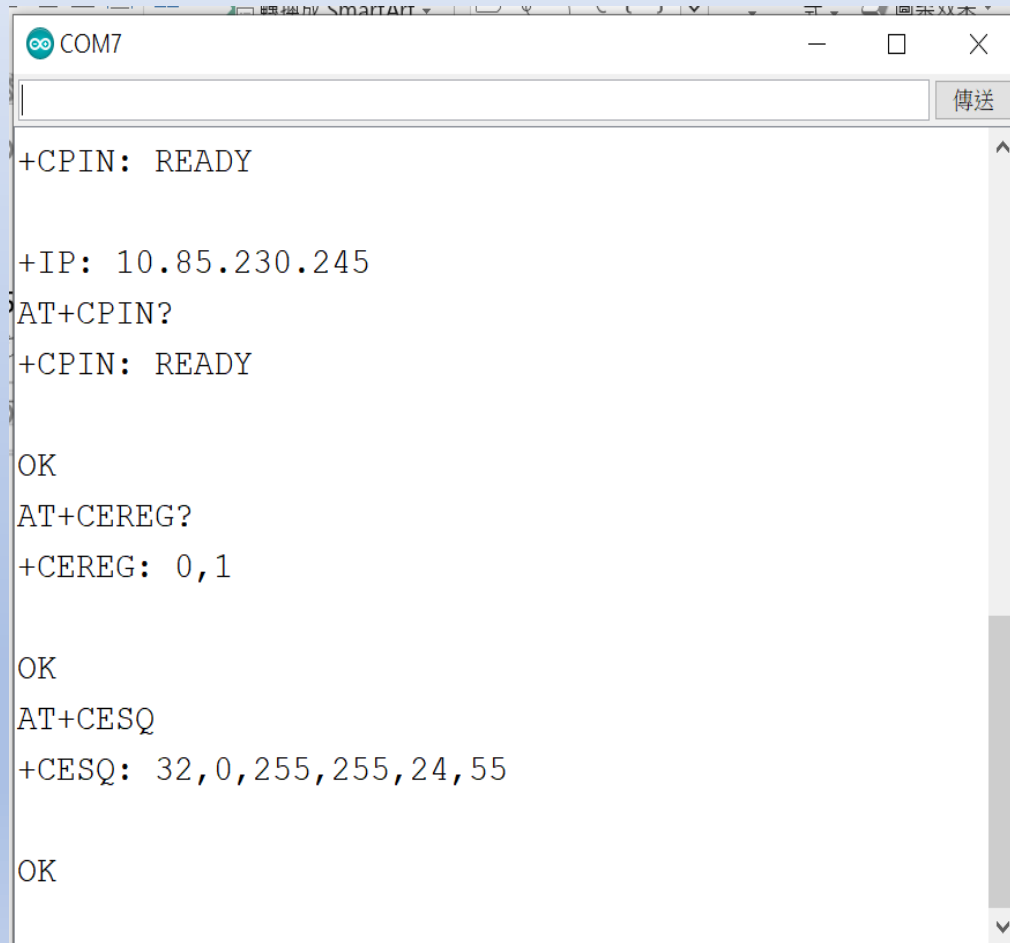
```
COM7
RDY
+CFUN: 1
+CPIN: READY
+IP: 10.85.230.245
AT+CPIN?
+CPIN: READY
OK
AT+CEREG?
+CEREG: 0,1
OK
```

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



```
COM7
+CPIN: READY

+IP: 10.85.230.245
AT+CPIN?
+CPIN: READY

OK
AT+CEREG?
+CEREG: 0,1

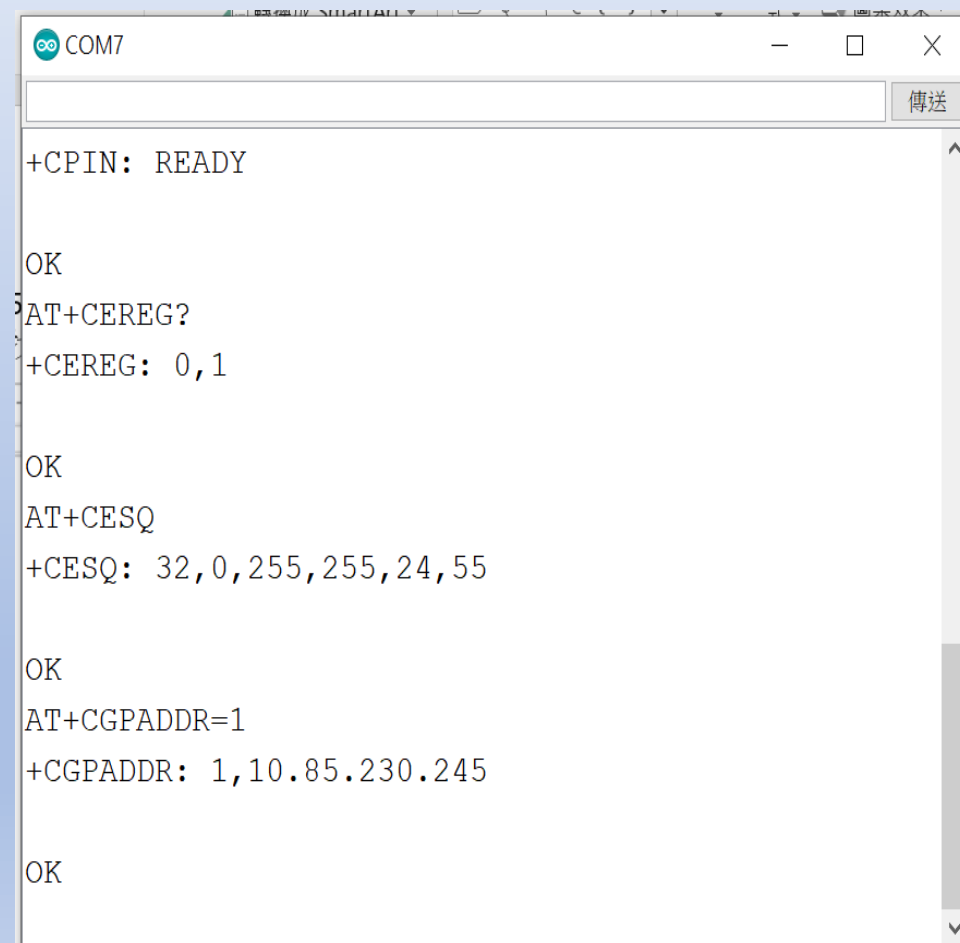
OK
AT+CESQ
+CESQ: 32,0,255,255,24,55

OK
```

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



```
COM7
+CPIN: READY

OK
AT+CEREG?
+CEREG: 0,1

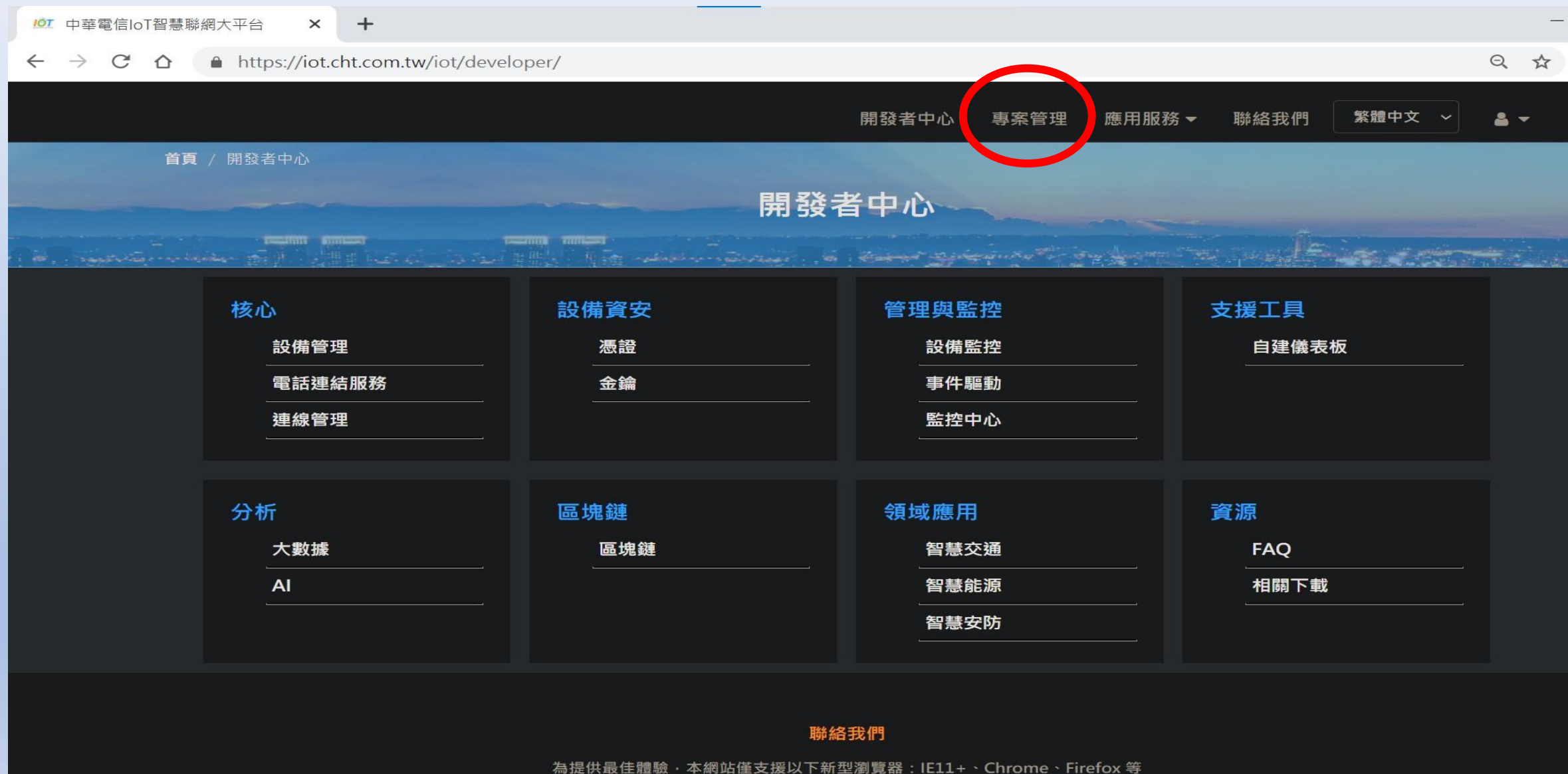
OK
AT+CESQ
+CESQ: 32,0,255,255,24,55

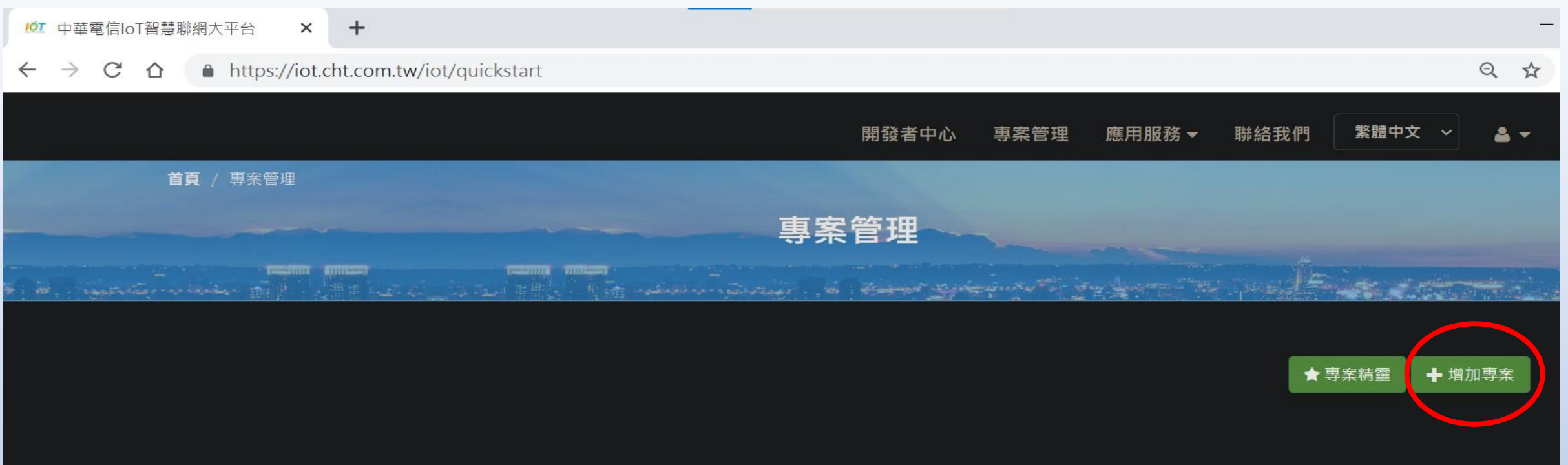
OK
AT+CGPADDR=1
+CGPADDR: 1,10.85.230.245

OK
```


中華電信IoT智慧聯網大平台概略設定： 請先註冊為中華電會員

於網站內先行設定下列資訊：





專案管理

專案名稱：

DSI2598P

專案描述：

NB-IoT 開發板專案

應用領域：

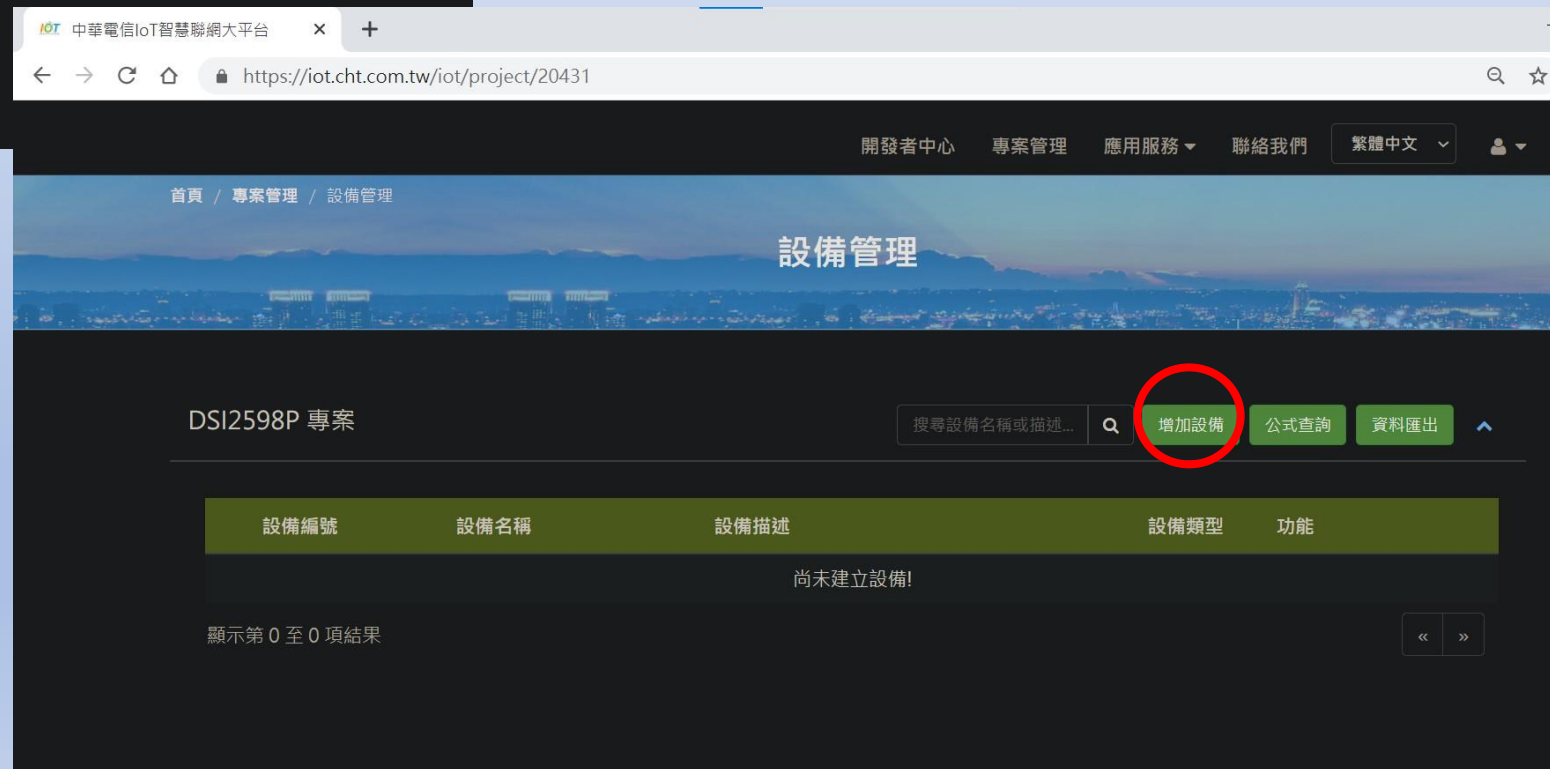
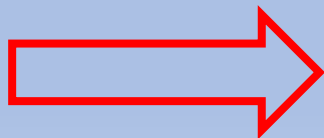
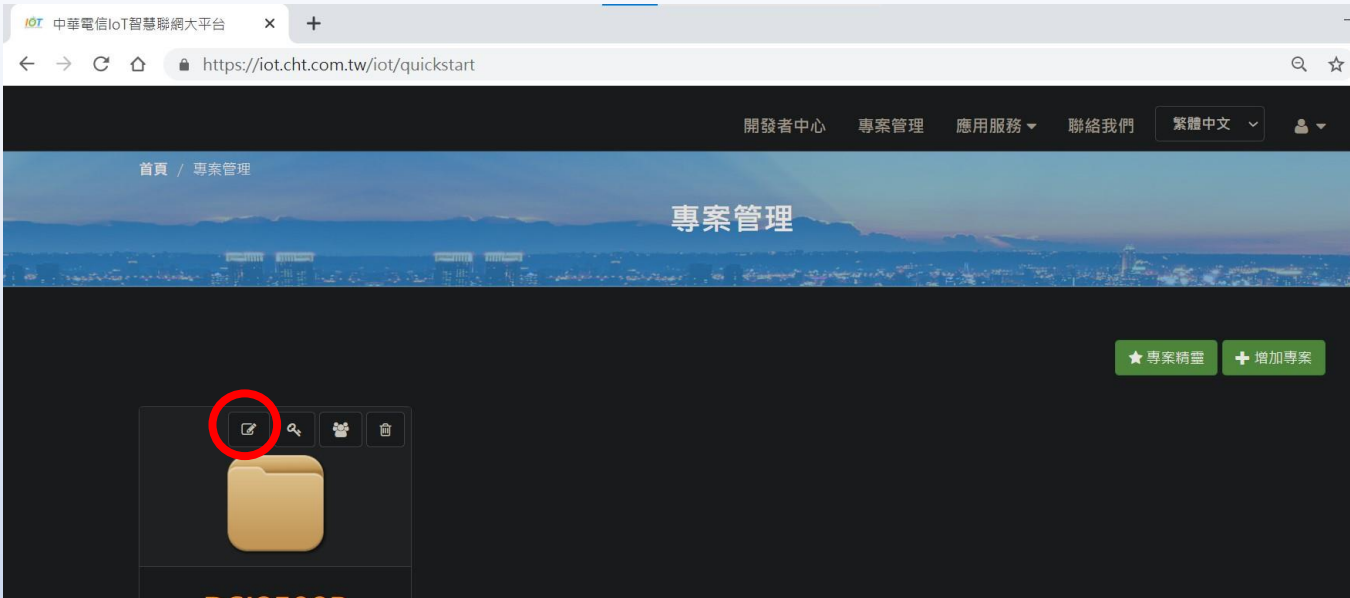
家庭/建築

統計資訊

設備數量：0 | 感測器數量：0

關閉

儲存



設備管理

基本資料

擴充屬性資訊

新增模式

☒ 使用者設定 ☐ 從領域模板匯入

設備名稱

溫溼度偵測

設備描述

DHT11

設備類型

☒ 通用設備 ☐ Modbus工業設備 ☐ UDP

地理位置

經度

經度 ...

緯度

緯度...

URI

URI ...

專案最多允許建立 4096 個設備

關閉

下一頁



設備管理

基本資料

擴充屬性資訊

提供客製化擴充屬性設置，滿足額外所需的設備屬性資訊

屬性名稱(key)	屬性數值(value)	
<div>Key</div>	<div>Value</div>	<div>+</div>

專案最多允許建立 4096 個設備

關閉

儲存

設備編號	設備名稱	設備描述	設備類型	功能
25620325408	溫溼度偵測	DHT11	通用設備	<div><div></div><div></div><div></div></div>

顯示第 1 至 1 項結果

『 溫溼度偵測 』設備資訊 (編號:25620325408)

- 感測器
- 設備內容
- 事件驅動
- 憑證申請
- 存取統計
- 連線管理
- 主動監控

溫溼度偵測 設備，共有 0 個感測器

數據顯示時間格式: UTC

增加感測器

感測器管理

基本資料

其他資料



識別編號(ID)

Temperature

識別編號只允許輸入英文或數字或底線符號

顯示名稱

溫度

描述

感測器描述...

類型

☒ 數值 ☐ 文字 ☐ 開關 ☐ 圖像

單位

單位 ...

公式

公式 ...

選擇平行公式

設備最多允許建立 128 個感測器

關閉

下一頁



感測器管理

基本資料

其他資料

個人客製化屬性設置，您可以於底下增加感測器屬性資訊

屬性名稱(key)

屬性數值(value)

Key

Value

+

設備最多允許建立 128 個感測器

關閉

儲存

感測器管理

基本資料

其他資料



識別編號(ID)

Humidity

識別編號只允許輸入英文或數字或底線符號

顯示名稱

濕度

描述

感測器描述...

類型

☒ 數值 ☐ 文字 ☐ 開關 ☐ 圖像

單位

單位 ...

公式

公式 ...

選擇平行公式

設備最多允許建立 128 個感測器

關閉

下一頁



感測器管理

基本資料

其他資料

個人客製化屬性設置，您可以於底下增加感測器屬性資訊

屬性名稱(key)

屬性數值(value)

Key




Value

+

設備最多允許建立 128 個感測器

關閉

儲存

設備編號	設備名稱	設備描述	設備類型	功能
25620325408	溫溼度偵測	DHT11	通用設備	  

顯示第 1 至 1 項結果

« 1 »

『溫溼度偵測』設備資訊 (編號:25620325408)

感測器

設備內容

事件驅動

憑證申請

存取統計

連線管理

主動監控

溫溼度偵測 設備，共有 2 個感測器

數據顯示時間格式: Asia/Taipei

增加感測器

>_ ↺ ✎ ✕

溫度

>_ ↺ ✎ ✕

濕度

基本資料

擴充屬性資訊

設備描述

DHT11

設備類型

general

地理位置

經度

經度 ...

緯度

緯度...

URI

X54BT3TGF04CX0S0

模組元件

☐ 設備是否採用中華電信硬體安全元件(ChipSim或eSIM)



設備金鑰

DKSHPK0U3HU279EUX2

3.1. AT! Display Product Identification Information

The execution command returns product identification information.

AT! Display Product Identification Information	
Execution Command AT!	Response Quectel_Ltd <Object Id> Revision: <revision> OK
Maximum Response Time	300ms

Parameter

<Object Id>	Identifier of device type
<revision>	Revision of software release

3.2. ATE Set Command Echo Mode

The execution command determines whether or not the UE echoes characters received from external MCU during command state.

ATE Set Command Echo Mode

Execution Command	Response
ATE<value>	OK
Maximum Response Time	300ms

Parameter

<value>	0	Echo mode OFF
	<u>1</u>	Echo mode ON

3.4. AT+IPR Set TE-TA Fixed Local Rate

Please refer to **Chapter 6** for possible <err> values.

AT+IPR Set TE-TA Fixed Local Rate	
Test Command AT+IPR=?	Response +IPR: (list of preferred auto baud <rate>s),(list of supported fixed-only <rate>s) OK
Read Command AT+IPR?	Response +IPR: <rate> OK
Write Command AT+IPR=<rate>	Response OK If there is any error, response: ERROR or +CME ERROR: <err>
Maximum Response Time	300ms

Parameter

<rate>	Baud rate per second
	<u>0</u> (Auto baud)
	110
	300
	1200
	2400
	4800
	9600
	19200
	38400
	57600
	115200
	230400
	460800
	921600

NOTES

1. The setting will apply to all channels routed through one connection level for UART.
2. The command is not applicable for USB interface.

2.1.1. AT+QIOPEN Open a Socket Service

This command is used to open a socket service. The service type can be specified by <service_type>, and the data access mode can be specified by <access_mode>. The URC “+QIOPEN: <connectID>,<err>” indicates whether the socket service has been opened successfully.

AT+QIOPEN Open a Socket Service	
Test Command AT+QIOPEN=?	Response +QIOPEN: (1-3),(0-4),"TCP/UDP","<IP_address>/<domain_name>",<remote_port>,<local_port>,(0-1)[,(0-1)] OK
Write Command AT+QIOPEN=<contextID>,<connectID>,<service_type>,<IP_address>/<domain_name>,<remote_port>[,<local_port>[,<access_mode>]][,<protocol_type>]]	Response OK +QIOPEN: <connectID>,<err> If there is an error related to ME functionality: ERROR
Maximum Response Time	300ms

2.1.4. AT+QISEND Send Text String Data

The command is used to send socket data in text string format via specified connection.

AT+QISEND Send Text String Data

Test Command

AT+QISEND=?

Response

+QISEND: (0-4),(1-1024),<data>

OK

Write Command

AT+QISEND=<connectID>,<send_length>,<data>

Response

If data is sent successfully:

OK

SEND OK

Otherwise:

OK

SEND FAIL

AT+QIRD Retrieve the Received TCP/IP Data

Test Command

AT+QIRD=?

Response

+QIRD: (0-4),(1-512)

OK

Write Command

AT+QIRD=<connectID><read_length>

Response

**+QIRD: <actual_read_length>[,<remaining_length>]
<data>**

OK

If there is no data:

+QIRD: 0

OK

If there is an error related to ME functionality:

ERROR

2.1.2. AT+QICLOSE Close a Socket Service

The command is used to close the specified socket service.

AT+QICLOSE Close a Socket Service

Test Command
AT+QICLOSE=?

Response
+QICLOSE: (0-4)

OK

Write Command
AT+QICLOSE=<connectID>

Response
If closed successfully:
OK

CLOSE OK

If there is an error related to ME functionality:
ERROR

Maximum Response Time

300ms

函式檔：BC26Init-H.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(); //Conduct a serial read
        data+=c; //Shorthand for data = data + c
        if (c=='\n') break;
    }
    data.trim();
    return data;
}
```

由於資料接收是屬於串列方式，因此透過該函式Check_RevData將字元整合成字串 !!


```
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);
    Long StartTime=millis();
    switch (stepnum)
    {
        case 0: // Reset BC26
            C_temp="+IP:";
            break;
        case 1: // Other Data
            C_temp="OK";
            break;
        case 2: // Check IPAddress
            C_temp="+CGPADDR:";
            break;
        case 10: // build MQTT Server
            C_temp="+QMTOPEN: 0,0";
            break;
        case 11: // Connect to MQTT server by username and password
            C_temp="+QMTCONN: 0,0,0";
            break;
    }
}
```

Send_ATcommand

(第一部分)

```

    case 12: // Publisher MQTT Data
        C_temp="+QMTPUB: 0,0,0";
        break;
    case 13: // Sub MQTT Data
        C_temp="+QMTSUB: 0,1,0,0";
        break;
    case 20: // Connect HTTP Server
        C_temp="+QIOPEN: 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;
}

```

Send_ATcommand

(第二部分)

若回傳資料正確則返回 99

初始化 DSI2598+ BC26Init()

```
bool BC26init() // 初始化 BC26
{
    Send_ATcommand("AT+QGACTION=1,1,\"apn\",\"internet.iot\",1);
    Send_ATcommand("AT+QCGDEFCONT=\"IP\",\"internet.iot\",1);
    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;
}
```

HTTP 連線及關閉 Connect_HTTP_Server() & Close_HTTP()

```
bool Connect_HTTP_Server(String S_name) // 建立 HTTP 連線通道
{
    String S_temp="";
    S_temp="AT+QIOPEN=1,0,\"TCP\", \"\" + S_name + "\",80,0,0";
    //Serial.println(S_temp);
    if (Send_ATcommand(S_temp,20)!=99) return false;
    //Send_ATcommand("AT+QICFG=\"viewmode\",1,4);
    delay(100);
    return true;
}

bool Close_HTTP()
{
    if (Send_ATcommand("AT+QICLOSE=0",1)!=99) return false;
    delay(100);
    return true;
}
```



```
bool Send_Data_CHT(String token,String id, String D_Message) // HTTP 方式傳送資料給網站
{
    String Send_check="";
    Serial.println("AT+QISEND=0");
    Serial1.println("AT+QISEND=0");
    Send_check=Check_RevData();
    Serial.println(Send_check);
    while (!Send_check.startsWith(">"))
    {
        Send_check=Check_RevData();
    }
    Serial.println("POST /iot/v1/device/" + id + "/rawdata HTTP/1.1");
    Serial1.println("POST /iot/v1/device/" + id + "/rawdata HTTP/1.1");
    Serial.println("Host: iot.cht.com.tw");
    Serial1.println("Host: iot.cht.com.tw");
    Serial.println("CK: " + token);
    Serial1.println("CK: " + token);
    Serial.println("Content-Type: application/json");
    Serial1.println("Content-Type: application/json");
    Serial.println("Content-Length:" + String(D_Message.length()));
    Serial1.println("Content-Length:" + String(D_Message.length()));
    Serial1.println();
    Serial.println(D_Message);
    Serial1.print(D_Message); //JSON Data
    Serial1.print("\x1A");

    Send_check=Check_RevData();
    Serial.println(Send_check);
    while (!Send_check.startsWith("SEND OK"))
    {
        Send_check=Check_RevData();
    }
    Serial.println(Send_check);
    return true;
}
```

```
String Get_QIRD_DATA() // 取得回傳資料
{
    int len;
    String S_temp="";
    S_temp="AT+QIRD=0,512";
    Serial.println(S_temp);
    Serial1.println(S_temp);
    while(!Serial1.available());
    while(Serial1.available())
        S_temp=Serial1.readString();
    Serial.println("資料長度：" + (String(S_temp.length())));
    if (S_temp.length()>30 && S_temp.length()<=512)
    {
        Serial.println(S_temp);
        return "OK";
    }
    return "Empty";
}
```



```

String Get_Data_CHT(String token,String id, String D_name) // HTTP 方式取得網站資料
{
    String Send_check="";

    Serial.println("AT+QISEND=0");
    Serial1.println("AT+QISEND=0");
    Send_check=Check_RevData();
    Serial.println(Send_check);
    while (!Send_check.startsWith(">"))
    {
        Send_check=Check_RevData();
    }
    Serial.println("GET /iot/v1/device/" + id + "/sensor/" + D_name + "/rawdata HTTP/1.1");
    Serial1.println("GET /iot/v1/device/" + id + "/sensor/" + D_name + "/rawdata HTTP/1.1");
    Serial.println("Host: iot.cht.com.tw");
    Serial1.println("Host: iot.cht.com.tw");
    Serial.println("CK: " + token);
    Serial1.println("CK: " + token);
    Serial1.println();
    Serial1.print("\x1A");

    Send_check=Check_RevData();
    //Serial.println(Send_check);
    while (!Send_check.startsWith("SEND OK"))
    {
        Send_check=Check_RevData();
        Serial.println(Send_check);
        Send_check=Check_RevData();
        while (!Send_check.startsWith("+QIURC: \"recv\"",0))
        {
            Send_check=Check_RevData();
        }
        Serial.println(Send_check);
    }
    return Get_QIRD_DATA();
}

```

主程式：DHT11-CHT-HTTP.ino

```
#include <SimpleDHT.h> // Arduino 內建
#include "BC26Init-H.h"

#define SLED PB12 // low 時,LED亮
#define BCRset PC13 // BC26 Reset 腳位

String Server_Name="iot.cht.com.tw"; //HTTP Server 的 IP 位址
String HTTP_Port="80"; //使用的埠
String Device_Key="DKSHPK0U3HU279EUX2"; //設備金鑰
String DeviceId="25620325408"; //設備編號
String DeviceName1="Temperature" ; //識別編號(ID1)
String DeviceName2="Humidity"; //識別編號(ID2)
String DATA_Value="";

String DelayTime="60"; // 暫停發送的時間設定

const int pinDHT11 = PB9; // DHT11 溫濕度傳感器 使用 PB9 腳位
SimpleDHT11 dht11(pinDHT11);
int err = SimpleDHTErrSuccess;
float temperature = 0;
float humidity = 0;
float CorrectionTEMP = 0.0; // 溫度校正值
float CorrectionHUMI = 20.0; // 溼度校正值
```

設定通訊協定，初始化 BC26

```
void setup()
{
  Serial.begin(115200);
  Serial1.begin(115200);
  pinMode(SLED, OUTPUT);
  pinMode(BCRset, OUTPUT);
  digitalWrite(SLED, HIGH);

  // 設為程式執行時的確認燈號

  // Rest BC26 ===
  digitalWrite(BCRset, LOW);
  delay(30);
  digitalWrite(BCRset, HIGH);
  delay (3000);
  // =====

  if (!BC26init()) {delay (10000); nvic_sys_reset();}
  Serial.println("初始化完成 ....");
}
```

BC26與STM32 溝通使用
Serial1 (PA9,PA10)

```
Serial1.begin(115200); //TX=PA9,RX=PA10
Serial2.begin(115200); //TX=PA2,RX=PA3
Serial3.begin(115200); //TX=PB10,RX=PB11
```

BC26 使用STM32 的
PC13腳位的高低電壓
做為 Rest 的動作 !!
(Low 30ms)

連線後，讀取DHT11最新偵測資料傳送至網站

```
void loop()
{
  Serial.println("=====");
  Serial.println("溫溼度模組 DHT11 檢測中....");
  int err = SimpleDHTErrSuccess;
  if ((err = dht11.read2(&temperature, &humidity, NULL)) != SimpleDHTErrSuccess)
  {
    Serial.print("Read DHT11 failed, err=");
    Serial.println(err);
    delay(1500);
    return;
  }
  delay(500);
  Serial.print("攝氏溫度：");Serial.print((int)temperature - CorrectionTEMP); Serial.println(" °C");
  Serial.print("環境溼度：");Serial.print((int)humidity - CorrectionHUMI); Serial.println(" %");
  String DHTtemp = String(temperature - CorrectionTEMP);
  String DHThumi = String(humidity - CorrectionHUMI);
  if (Connect_HTTP_Server(Server_Name))
  {
    digitalWrite(SLED, LOW);
    DATA_Value="[{\"id\":\""+ DeviceName1 + "\",\"value\":["+ DHTtemp + "]}]";
    if (Send_Data_CHT(Device_Key,DeviceId,DATA_Value))
    {
      Serial.println("攝氏溫度 資料上傳成功 !!\n");
    }
    DATA_Value="[{\"id\":\""+ DeviceName2 + "\",\"value\":["+ DHThumi + "]}]";
    if (Send_Data_CHT(Device_Key,DeviceId,DATA_Value))
    {
      Serial.println("環境溼度 資料上傳成功 !!\n");
    }
    Close_HTTP();
    digitalWrite(SLED, HIGH);
  }
}
```

將資料上傳後，等候10秒嘗試將網站資料讀回 !!

```
Serial.println("延遲 10 秒後讀回 CHT IOT 平台資料 .....");
delay (10000);

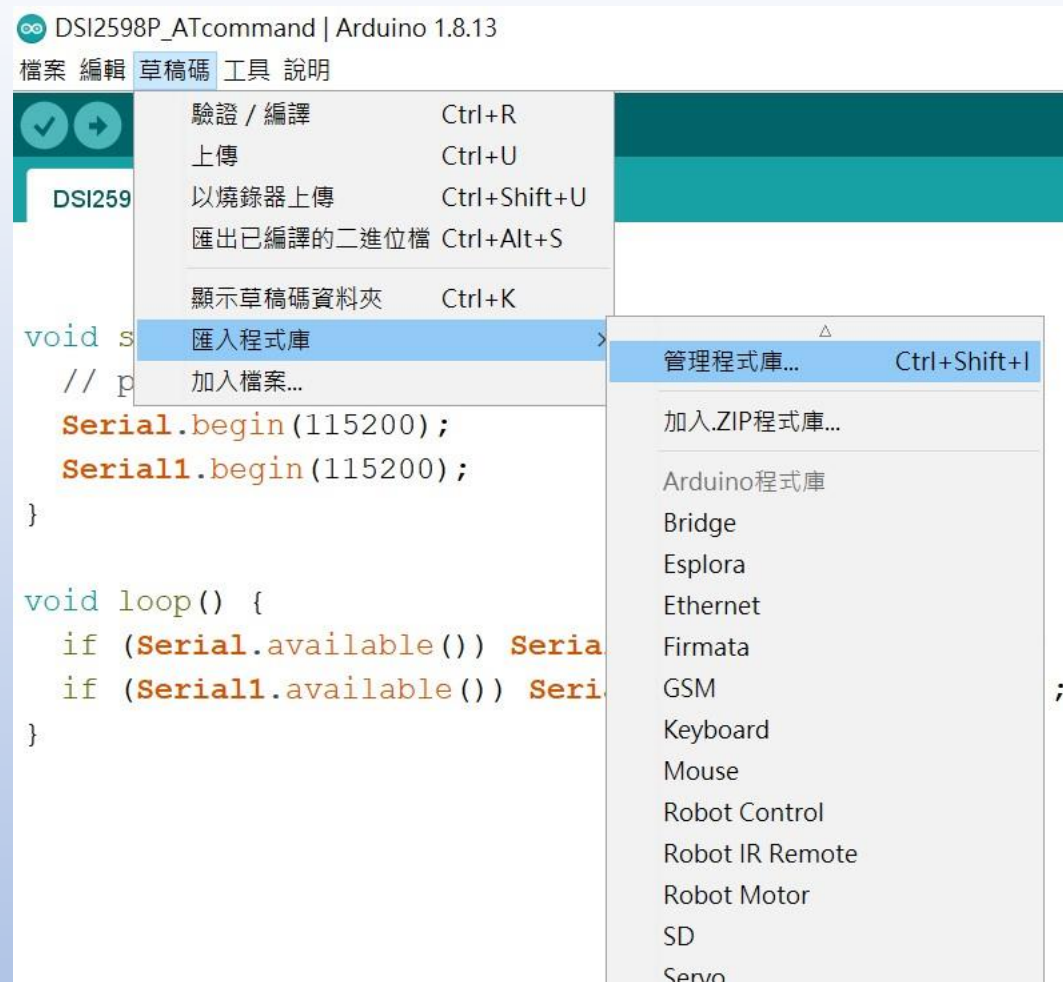
if (Connect_HTTP_Server(Server_Name))
{
    digitalWrite(SLED, LOW);
    if (Get_Data_CHT(Device_Key,DeviceId,DeviceName1)!="Empty")
    {
        Serial.println("資料下載成功 !!\n");
    }
    else Serial.println("可惜資料大於 512 byte 無法正常讀取 !!\n");
    Close_HTTP();
    digitalWrite(SLED, HIGH);
}

Serial.println("暫停 " + DelayTime +" 秒後再傳送 !!");
delay (DelayTime.toInt()*1000);
}
```

BC26僅能接收小於
512 Bytes 的資料，
由於網站回傳資料
大於 512 Bytes，
因此無法取得資料

程式編譯時，請先安裝所需的程式庫：

SimpleDHT



電路圖

