



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

IDEAS Chain 網站 <https://www.ideaschain.com.tw/>

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. Ideaschain 網站註冊及概略設定
4. HTTP AT_Command 概略說明
5. HTTP 程式(函式)使用說明

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：

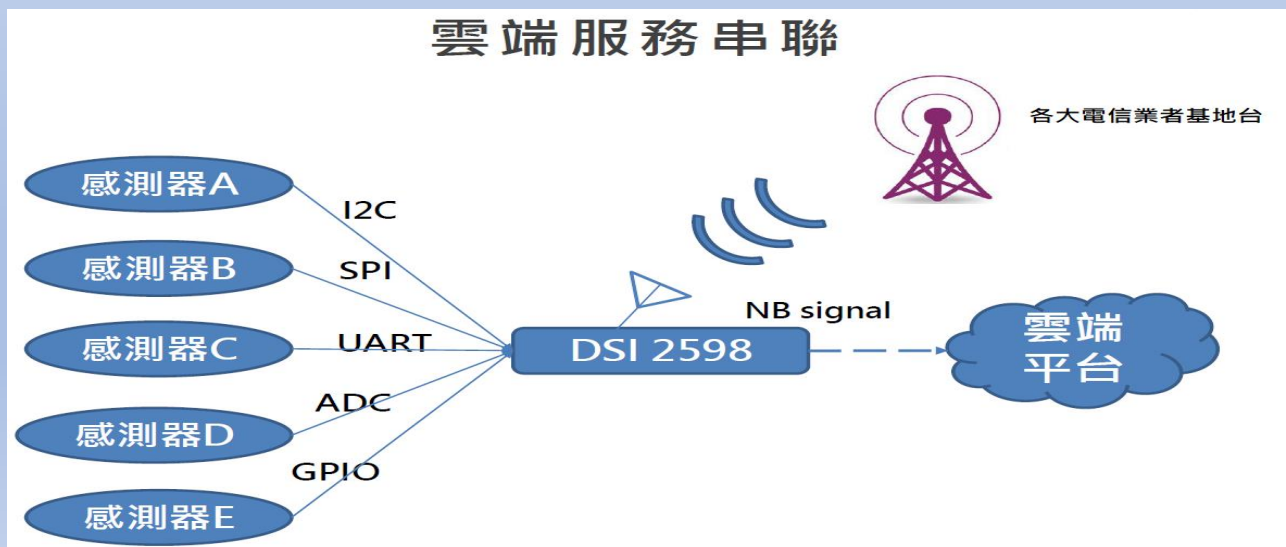
1. WiFi 透過無線基地台連上網際網路，NB-IoT 利用電信基地台連上網際網路。
2. WiFi 適用傳輸大量資料的訊息，NB-IoT 適用小資料量傳輸。
3. WiFi 連接無線基地台的距離較短，NB-IoT 由於全台基地台涵蓋率夠高，幾乎無死角。
4. WiFi 晶片耗用功率較高，NB-IoT 採用低功率晶片，使用一般 AA 電池可達 3-5 年以上。
5. WiFi 連網較易取得真實IP 位址，而NB-IoT使用的電信基地台提供的IP 位址大都為 虛擬 IP 網段。

DSI 2598+

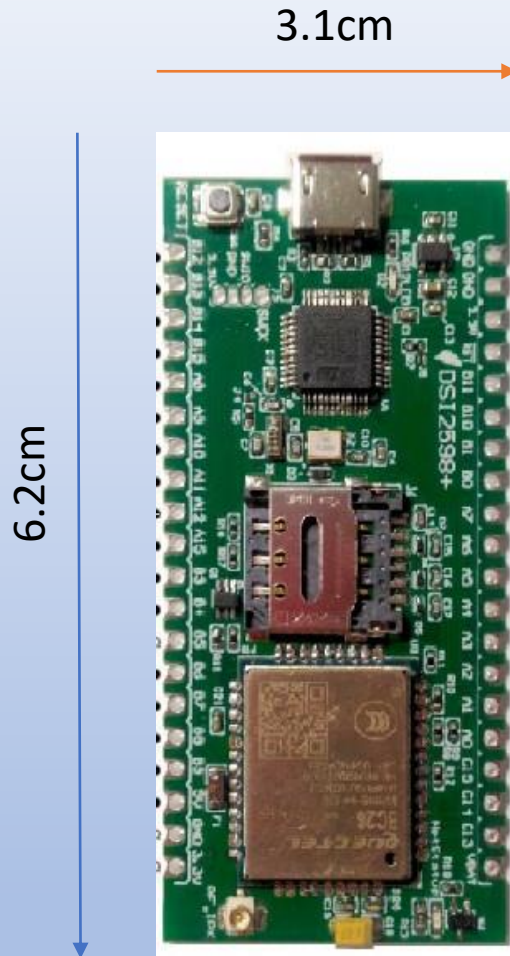
採用MT 2625

結合 STM32F103C8T6 晶片，DSI2598第二代 NB-IoT開發板

DSI2598P使用聯發科技NB-IoT晶片-MT2625模組，STM32F103C8T6晶片，有著PWM、I2C、SPI、ADC、UART等多種腳位功能，簡單但完整，可讓使用者無縫接軌任何Arduino程式庫，進行各項功能程式開發，是改善DSI2598速度及記憶體空間不足的第二代 NB-IoT開發板。



外型與尺寸說明



支持NB-IoTR14 的系統單晶片，以超高整合度為大量物聯網設備提供兼具低功耗及成本效益的解決方案，廣泛適用於家庭、城市、工業或行動應用。

高度整合NB-IoT調制解調數字信號處理器、射頻天線及前端模擬基帶，同時結合ARM Cortex-M 微控制器（MCU）、偽靜態隨機存儲器（PSRAM）、閃存與電源管理單元（PMU）。

整合一系列豐富的外圍輸入輸出介面，包括安全數字輸入輸出模塊（SDIO）、通用異步收發傳輸器（UART）、I2C 傳輸協議、I2S、序列外圍接口（SPI）及脈衝寬度調制（PWM）。

具備強大功能於小巧的封裝尺寸和少量的管腳數目，滿足物聯網設備對成本及體積的需求，並有助於廠商簡化其產品設計流程。

DSI 2598+ 基於實時操作系統（RTOS），易於針對各種不同的應用進行客製化，比如家庭自動化、雲信標（cloud beacon）、智慧型電錶及多項物聯網靜態或行動應用。

DSI 2598+ 的寬頻前端模組支持3GPP R14 規範，涵蓋超低頻/低頻/中頻/四頻的全頻段運作，可滿足全球市場需求，進而降低成本和開發時間。

DSI2598+ Arduino 開發環境

接腳及硬體功能說明:

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)

STM32F103
PINOUT DIAGRAM

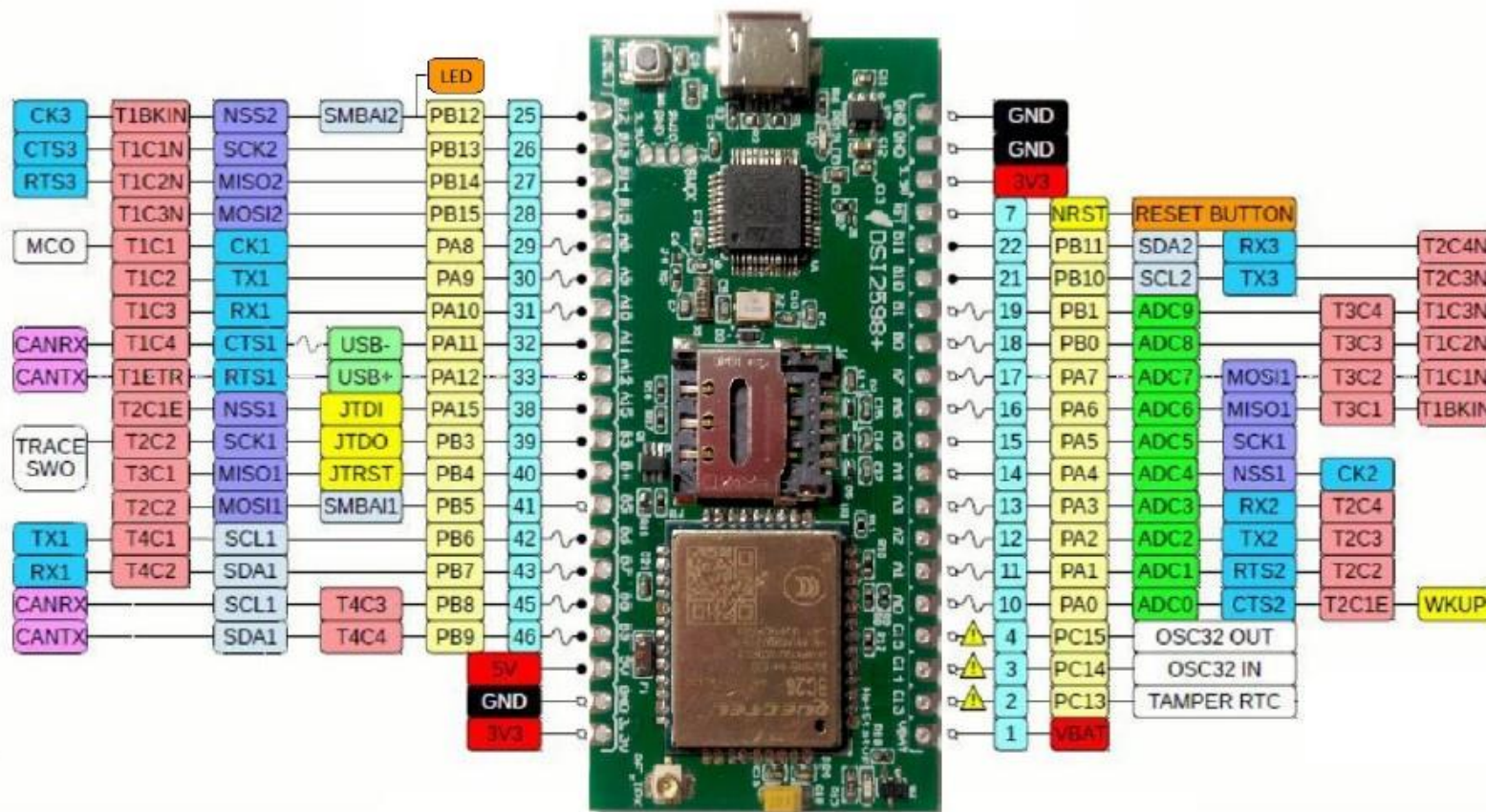
LEGEND

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, max 30pF

Absolute MAX 150mA total source/sink for entire CPU

Max ±20mA per pin, ±8mA recommended

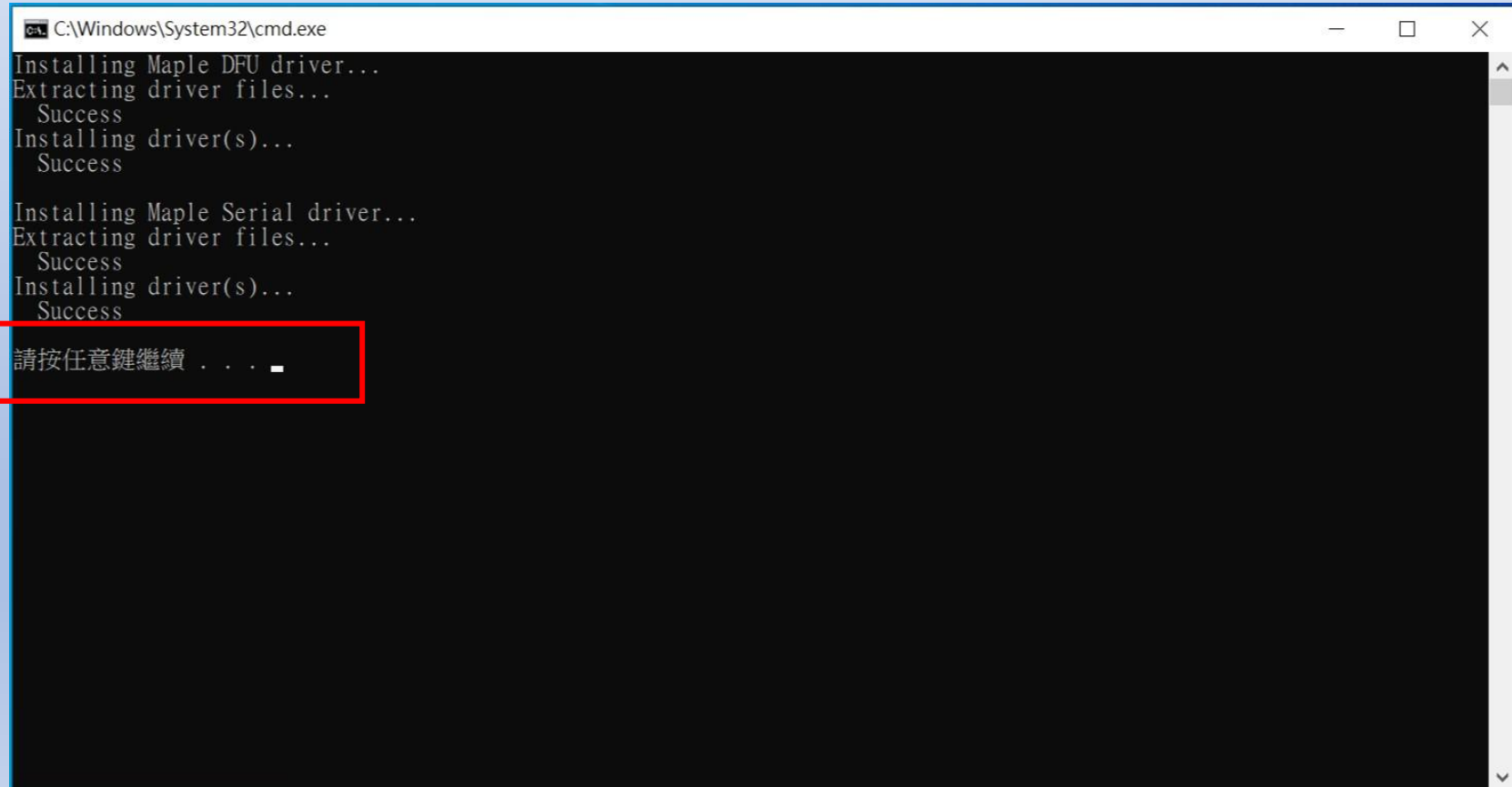
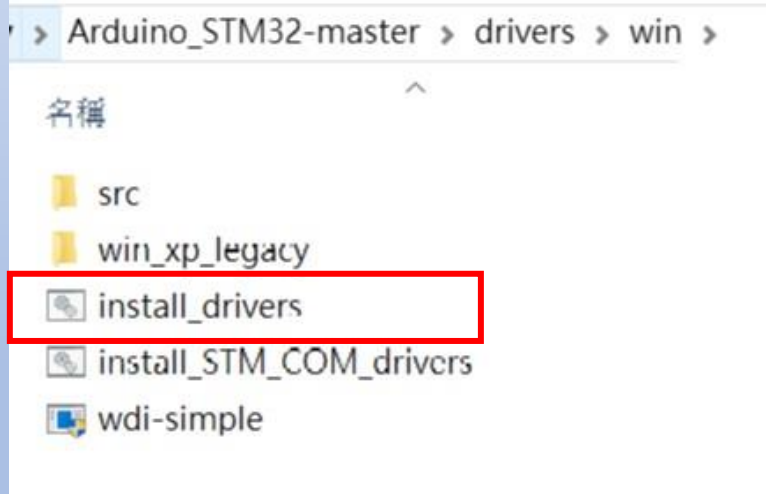


設定Arduino DSI2598+開發板的環境: (for Windows 10 作業系統)

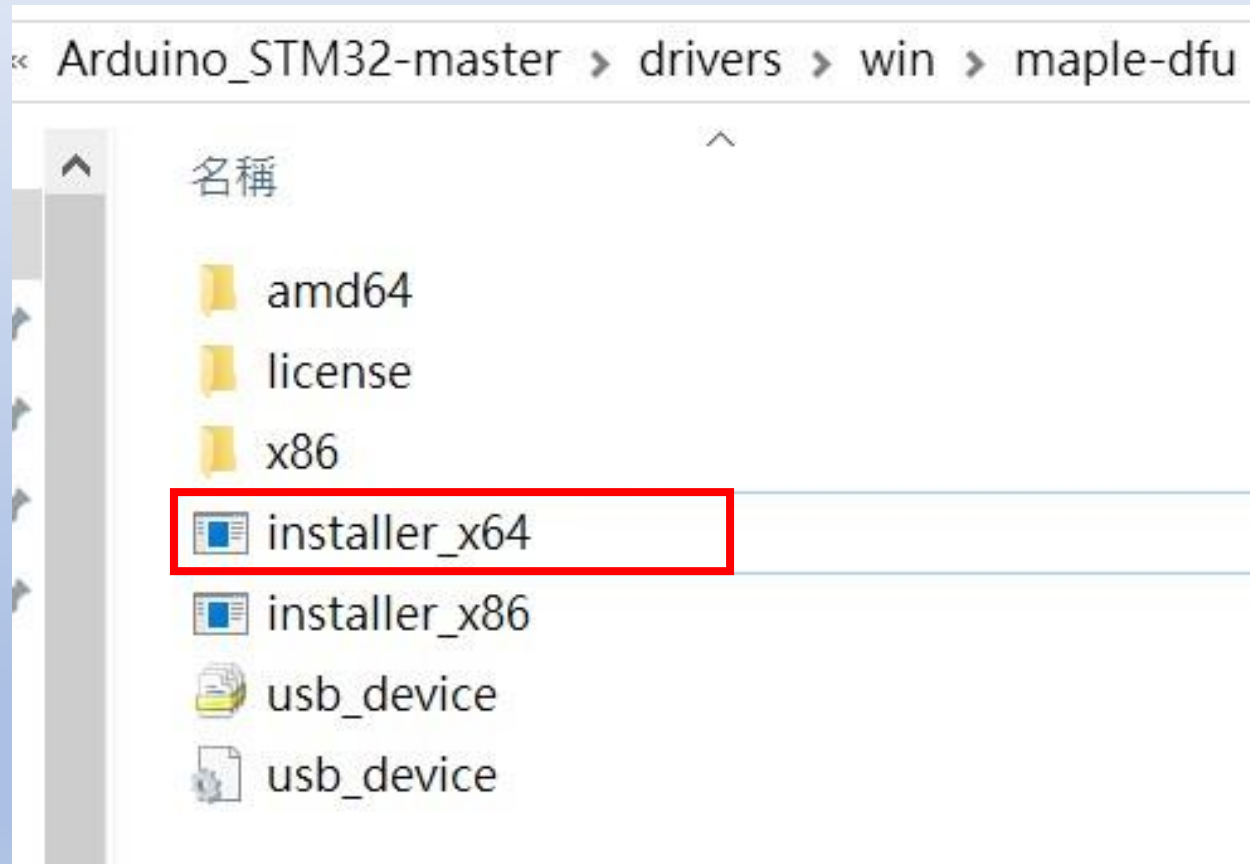
1. 安裝DFU windows 的driver :

下載目 https://github.com/rogerclarkmelbourne/Arduino_STM32

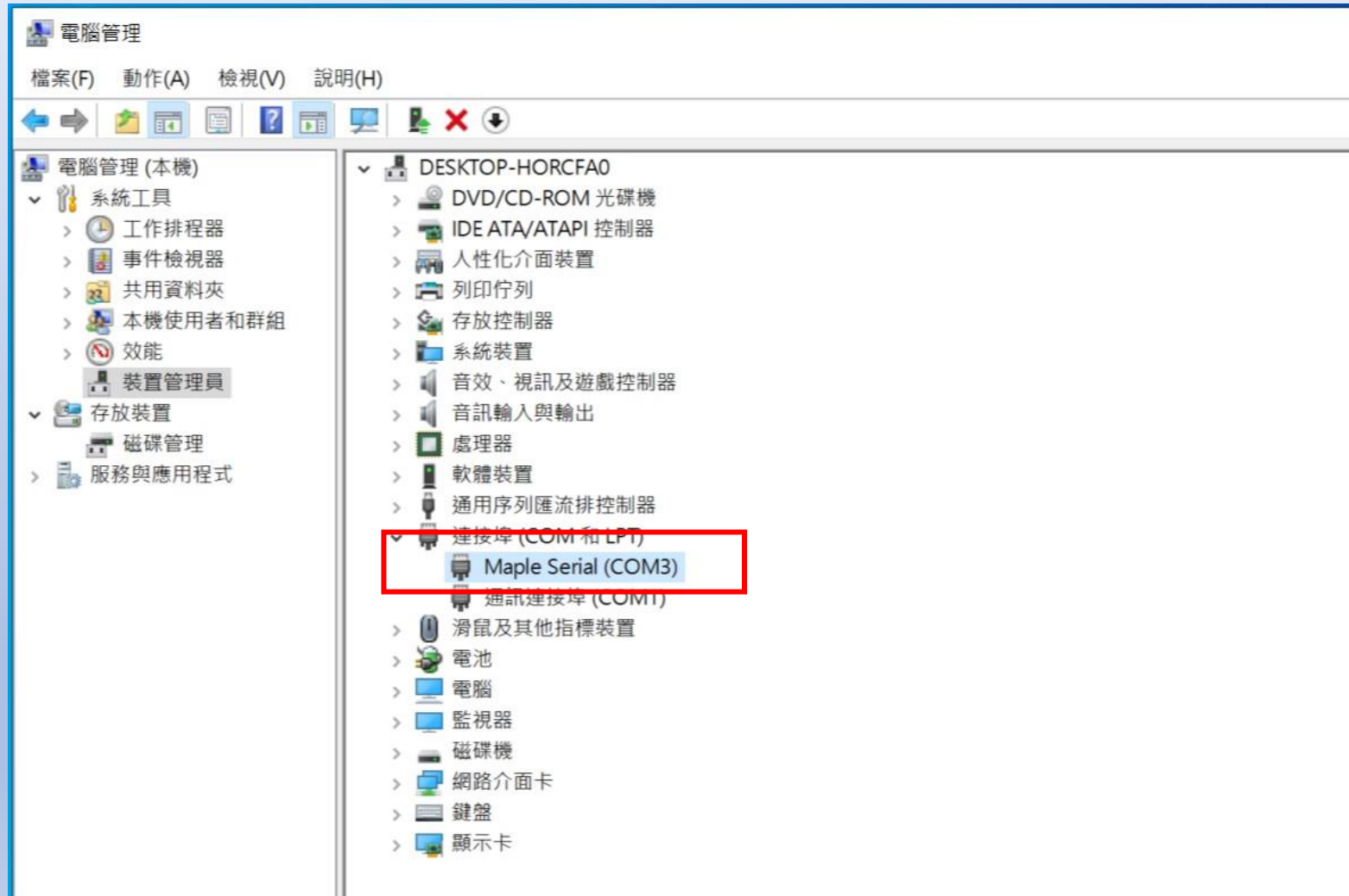
至 **Arduino_STM32-master.zip** , 解開檔案之後在目錄下用系統管理者執行 **Arduino_STM32-master\drivers\win\install_drivers.bat** , 會出現下列畫面



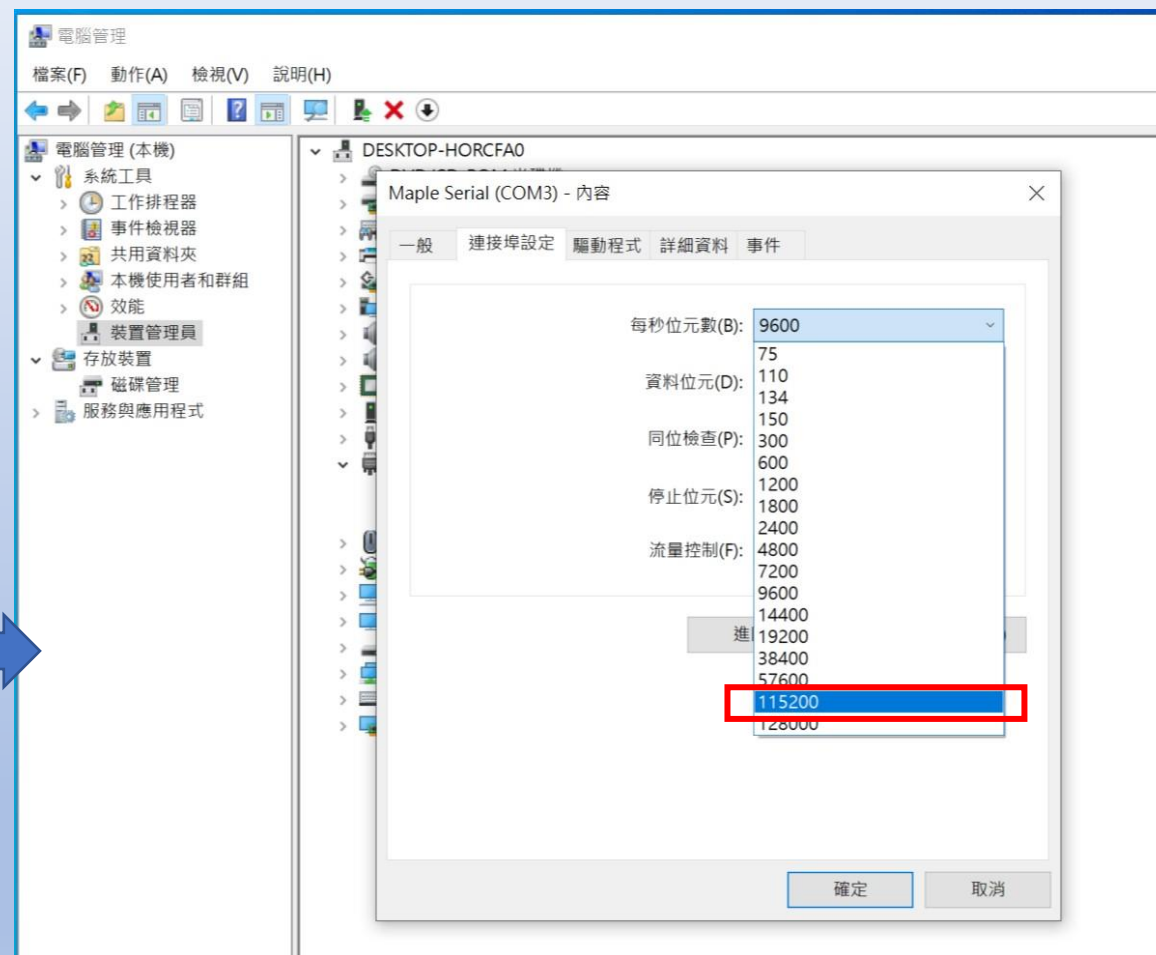
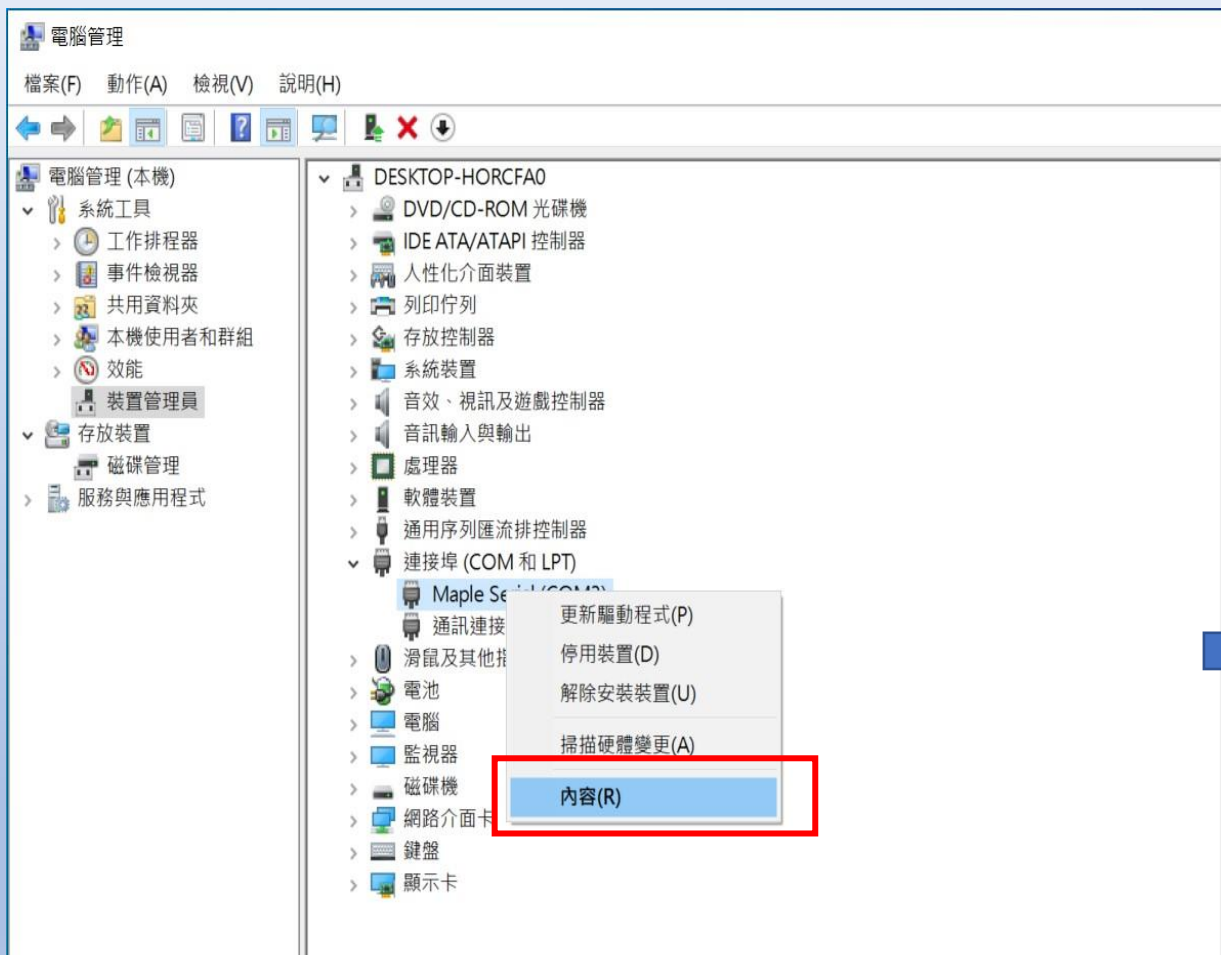
2. 在 `Arduino_STM32-master\drivers\win\maple-dfu` 下用系統管理者執行 `installer_x64.exe` (若系統為 32 位元 請執行 `installer_x86.exe`)，如下畫面：



3.將DSI2598+ 透過MicroUSB 線插入電腦 USB port 中，透過檢視電腦管理介面
確認出現以下畫面：

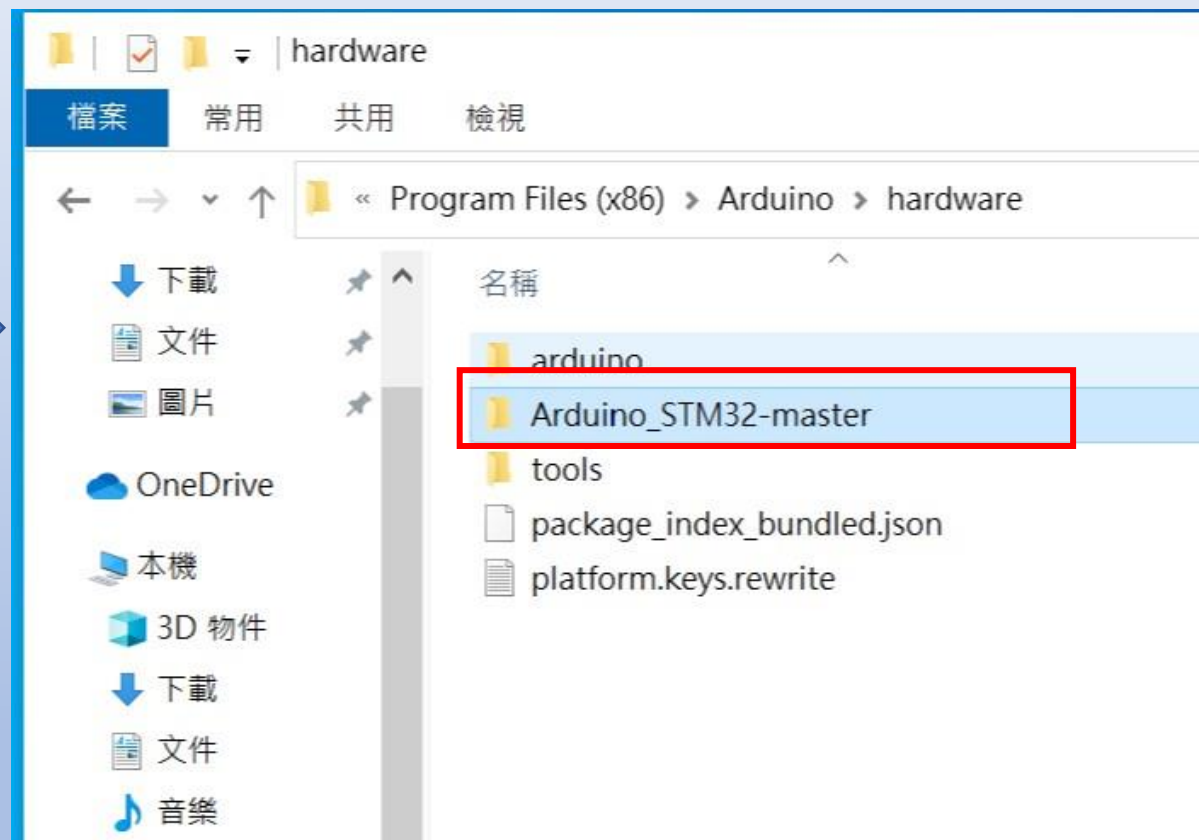
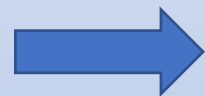
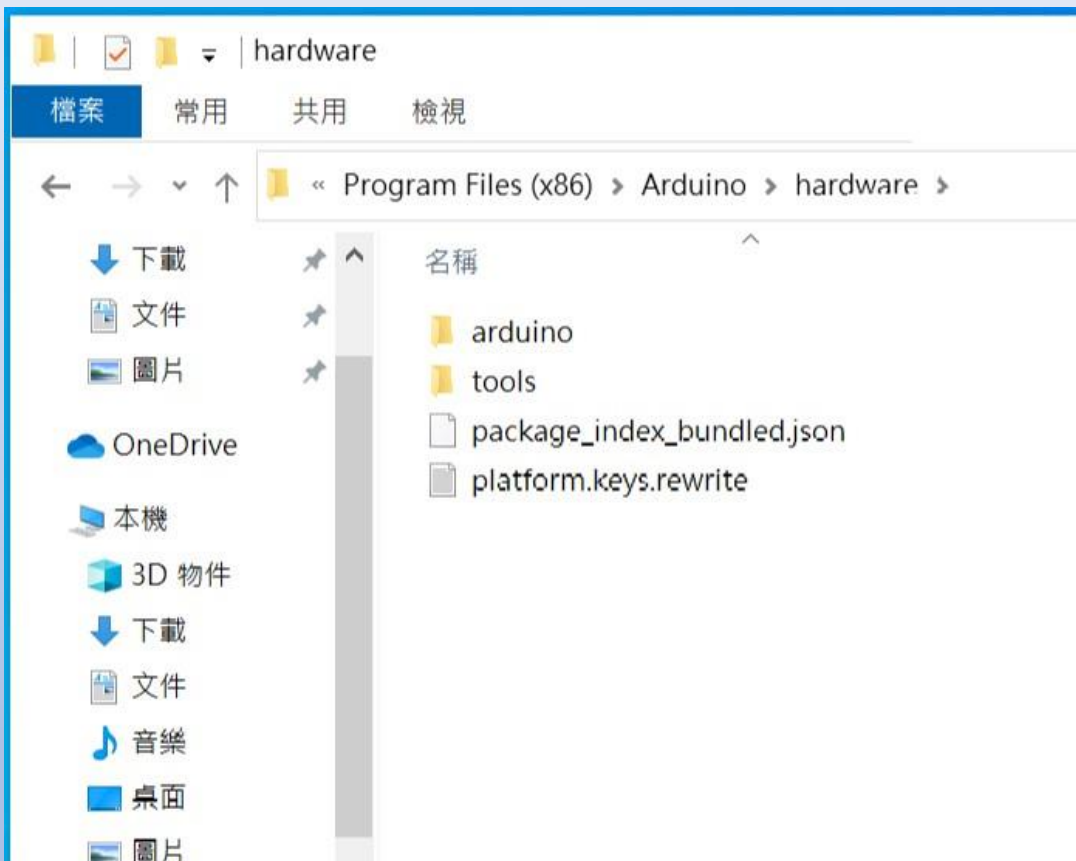


4.更改傳輸速率，加快傳輸動作！！

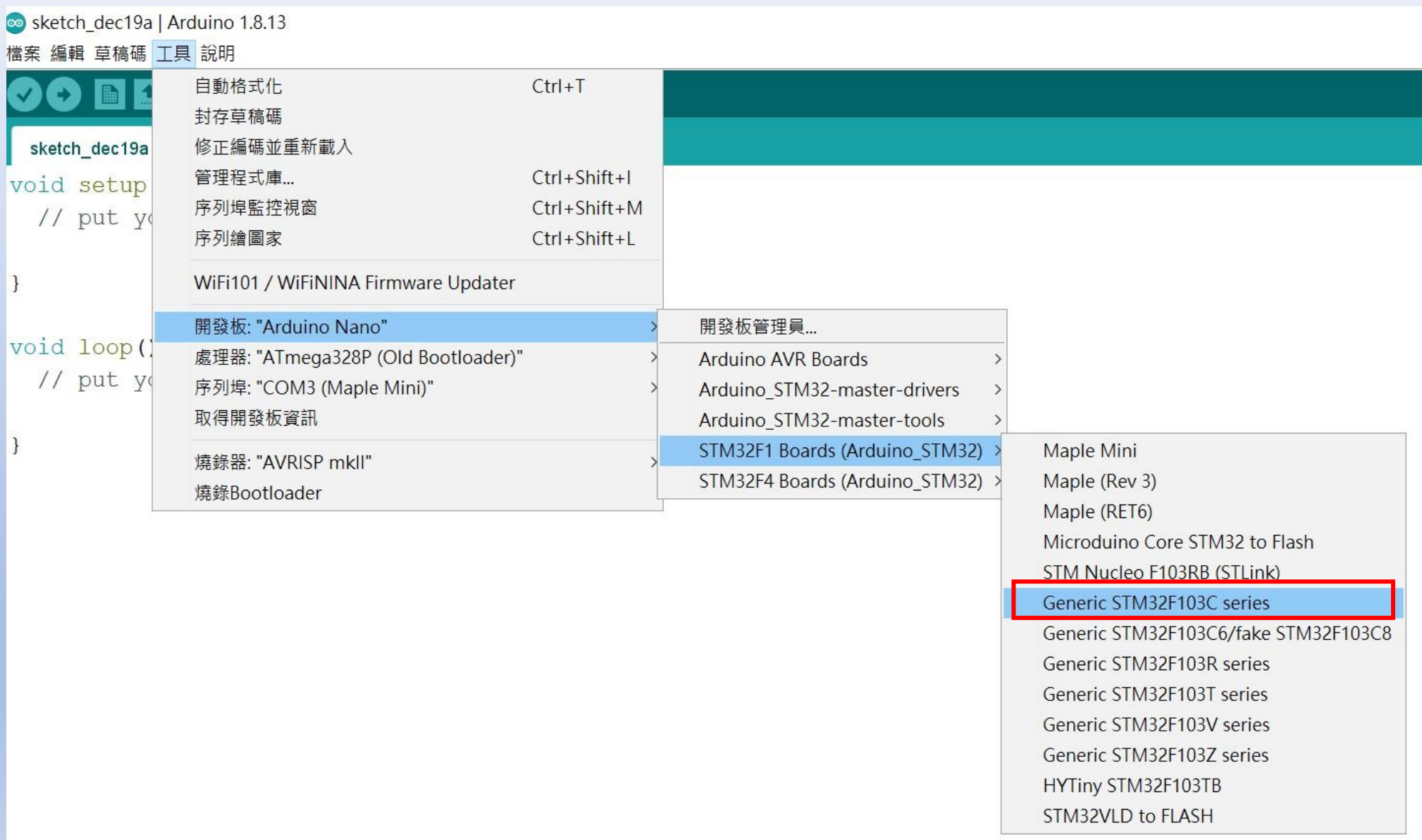


完成驅動程式的安裝

5. 將 **Arduino_STM32-master** 目錄拷貝至 Arduino IDE 的安裝目錄中，預設畫面如下：



6. 開啟 Arduino IDE 程式，設定以下動作：



7. 確認設定與以下畫面相同：



sketch_dec19a | Arduino 1.8.13

檔案 編輯 草稿碼 工具 說明

- 自動格式化 Ctrl+T
- 封存草稿碼
- 修正編碼並重新載入
- 管理程式庫... Ctrl+Shift+I
- 序列埠監控視窗 Ctrl+Shift+M
- 序列繪圖家 Ctrl+Shift+L

WiFi101 / Wi-FiNINA Firmware Updater

- 開發板: "Generic STM32F103C series" >
- Variant: "STM32F103C8 (20k RAM, 64k Flash)" >
- Upload method: "STM32duino bootloader" >
- CPU Speed(MHz): "72Mhz (Normal)" >
- Optimize: "Smallest (default)" >

- 序列埠: "COM3 (Maple Mini)" >
- 取得開發板資訊

- 燒錄器 >
- 燒錄Bootloader



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) !!

8. 執行編譯時會若出現錯誤：

```
開發板 Generic STM32F103C series 編譯錯誤。  
exec: "/bin/arm-none-eabi-g++": file does not exist  
開發板 Generic STM32F103C series 編譯錯誤。
```

請安裝 下列開發版套件



接下頁



DHT11-ideash \$ BC26Init.h

```
void setup()
{
  Serial.begin(115200);
  Serial1.begin(115200);
  pinMode(PB6, OUTPUT);
  pinMode(PB7, OUTPUT);
  pinMode(PB8, OUTPUT);
  pinMode(PC13, OUTPUT);
  digitalWrite(PB6, HIGH);

  // Rest BC26 ===
  digitalWrite(PC13, LOW);
  delay(30);
  digitalWrite(PC13, HIGH);
  delay(3000);
  // =====

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

開發板管理員

類型 全部

Arduino megaAVR Boards
by **Arduino**
此套件包含的開發板:
Arduino Uno WiFi Rev2, Arduino Nano Every.
[Online Help](#)
[More Info](#)

Arduino SAM Boards (32-bits ARM Cortex-M3)
by **Arduino**
此套件包含的開發板:
Arduino Due.
[Online Help](#)
[More Info](#)

1.6.12 安裝

Arduino SAMD Boards (32-bits ARM Cortex-M0+)
by **Arduino**
此套件包含的開發板:
Arduino MKR WiFi 1010, Arduino Zero, Arduino MKR1000, Arduino MKRZERO, Arduino MKR FOX 1200, Arduino MKR WAN 1300, Arduino MKR WAN 1310, Arduino MKR GSM 1400, Arduino MKR NB 1500, Arduino MKR Vidor 4000, Arduino Nano 33 IoT, Arduino M0 Pro, Arduino M0, Arduino Tian, Adafruit Circuit Playground Express.

完成環境設定安裝

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。

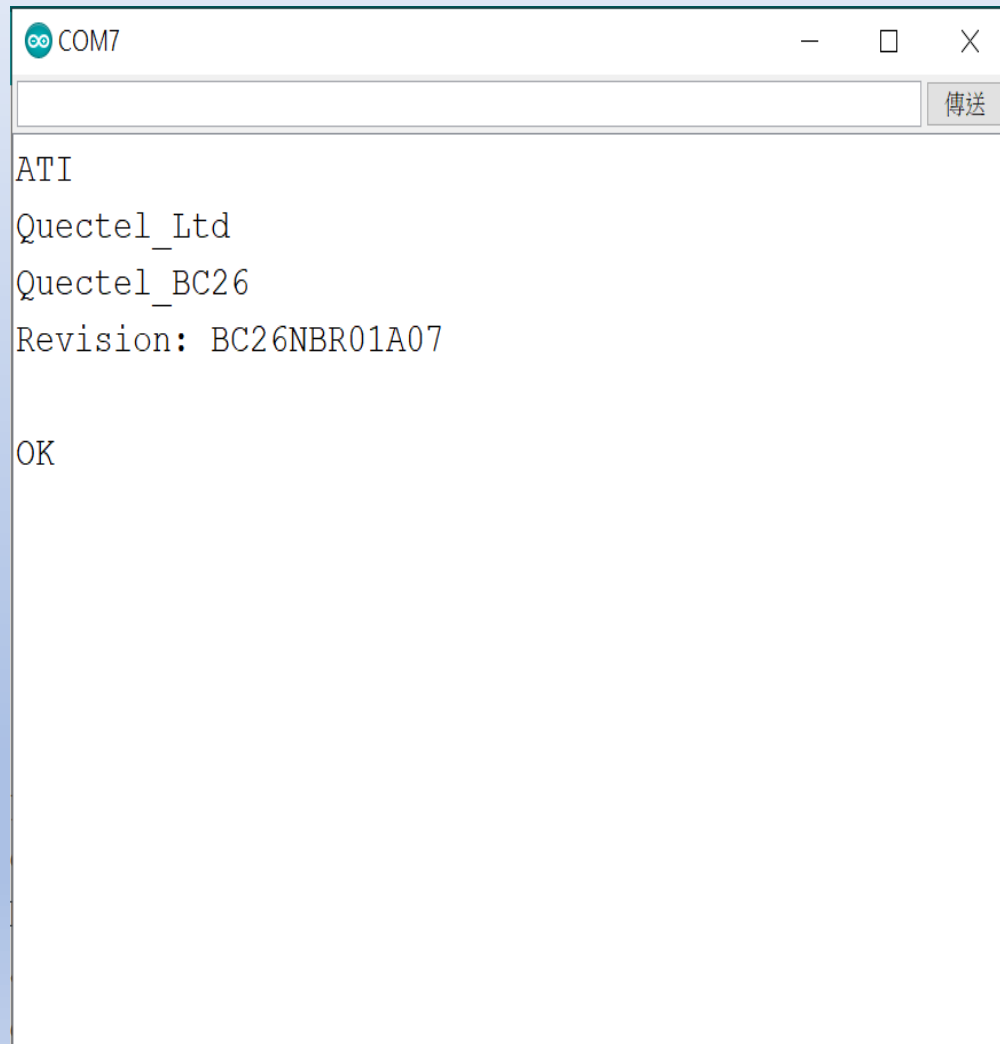
STEP 2: 選擇 工具 ->開發板->Generic STM32F103C series, 然後按下上傳 (Ctrl+U), 將
程式燒錄進去。



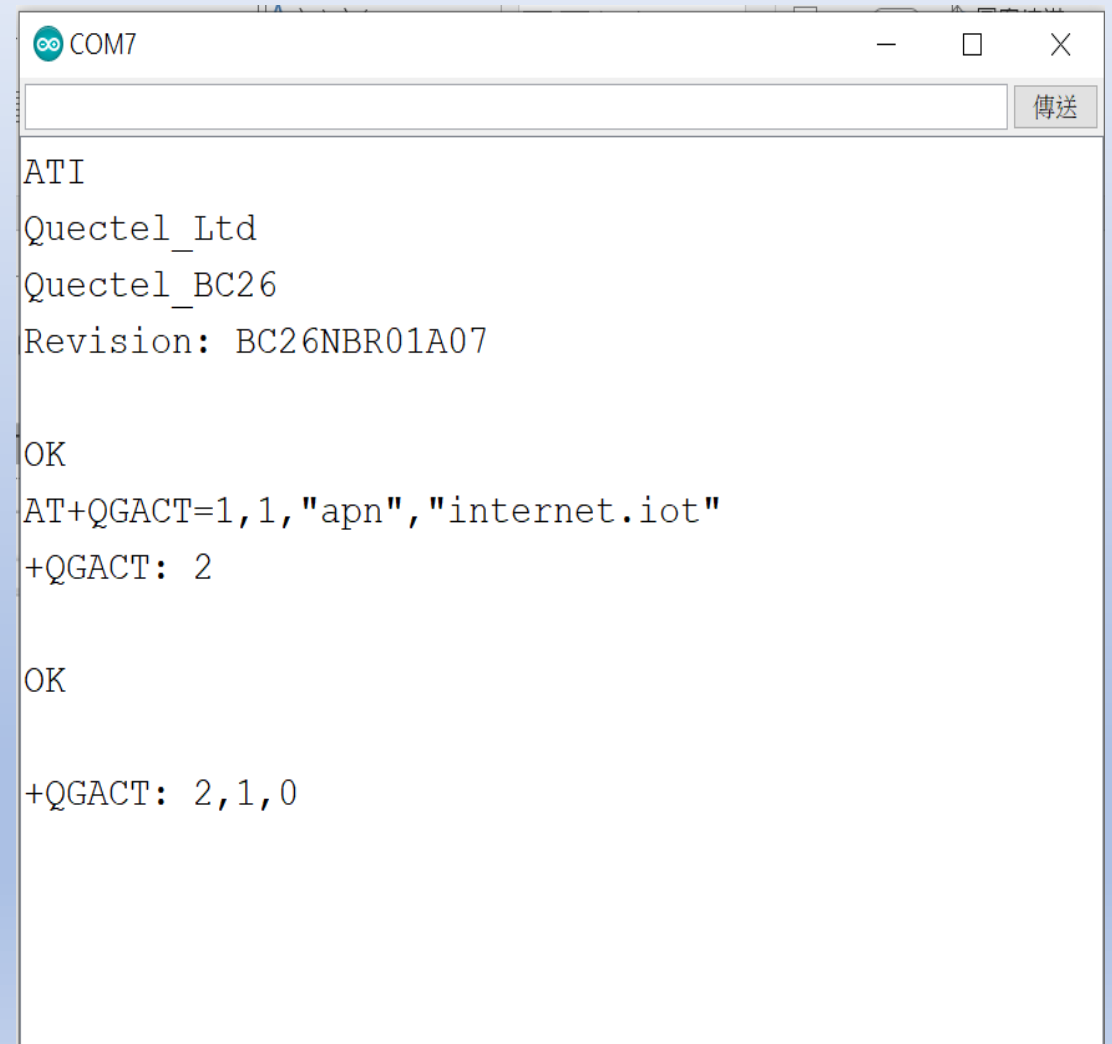
STEP 3：打開序列埠監控視窗，在上方輸入欄中輸入ATI 指令，可先輸入「ATI」，查看模組是否有回覆版本訊息。

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
ATI
Quectel_Ltd
Quectel_BC26
Revision: BC26NBR01A07
OK
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

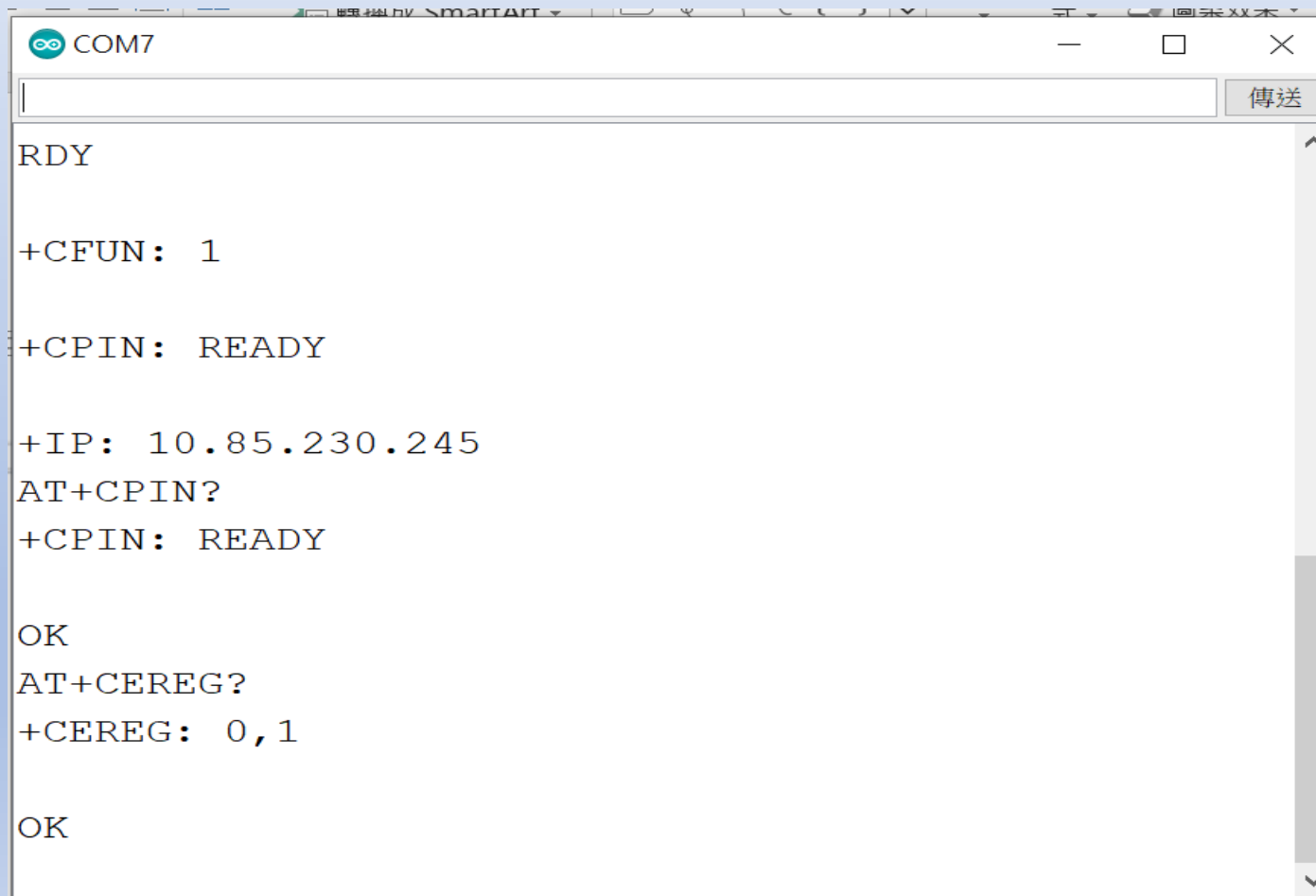
STEP 2 :

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。



The screenshot shows a serial terminal window titled 'COM7'. The text displayed is as follows:

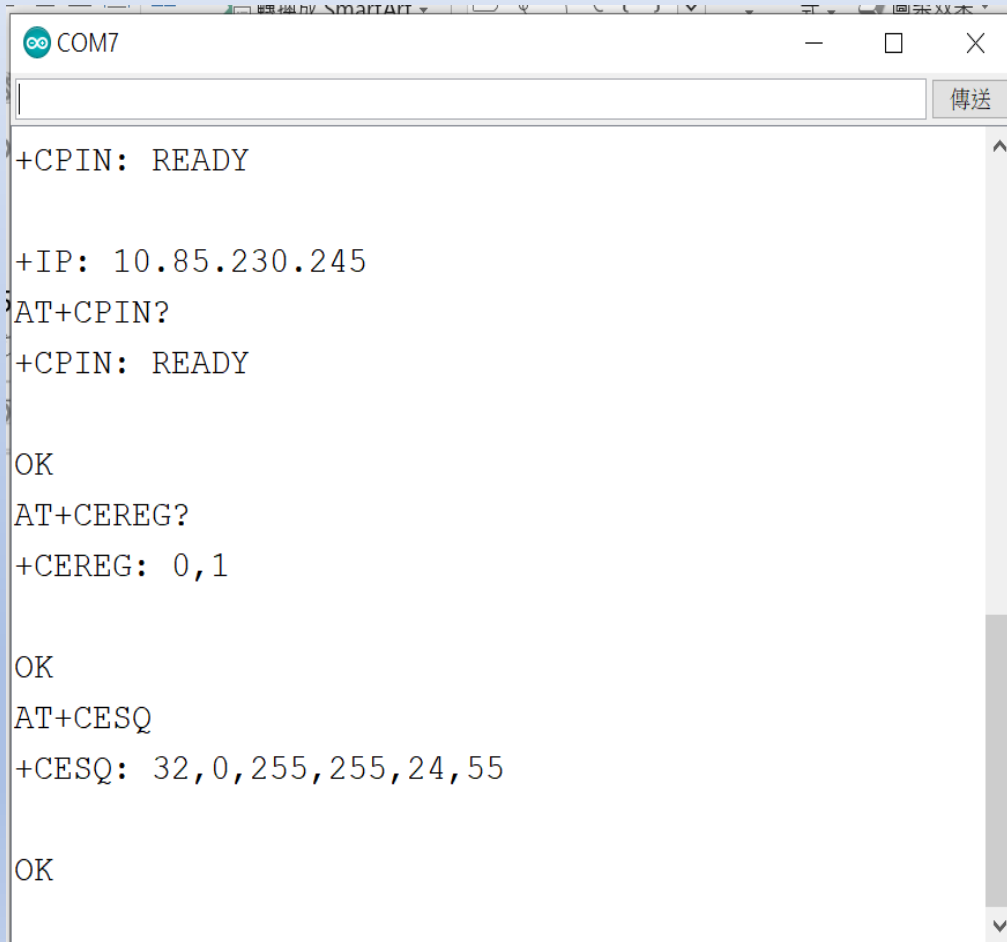
```
RDY  
  
+CFUN: 1  
  
+CPIN: READY  
  
+IP: 10.85.230.245  
AT+CPIN?  
+CPIN: READY  
  
OK  
AT+CEREG?  
+CEREG: 0,1  
  
OK
```

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

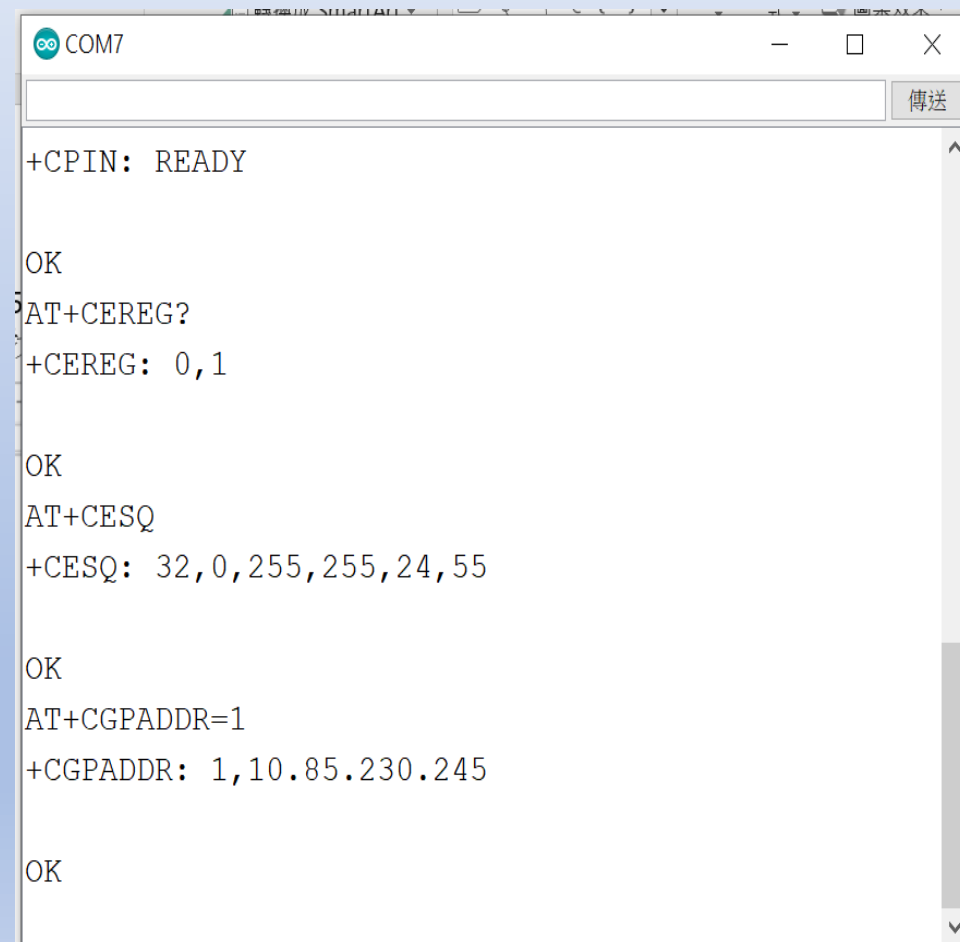
OK
AT+CESQ
+CESQ: 32,0,255,255,24,55

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

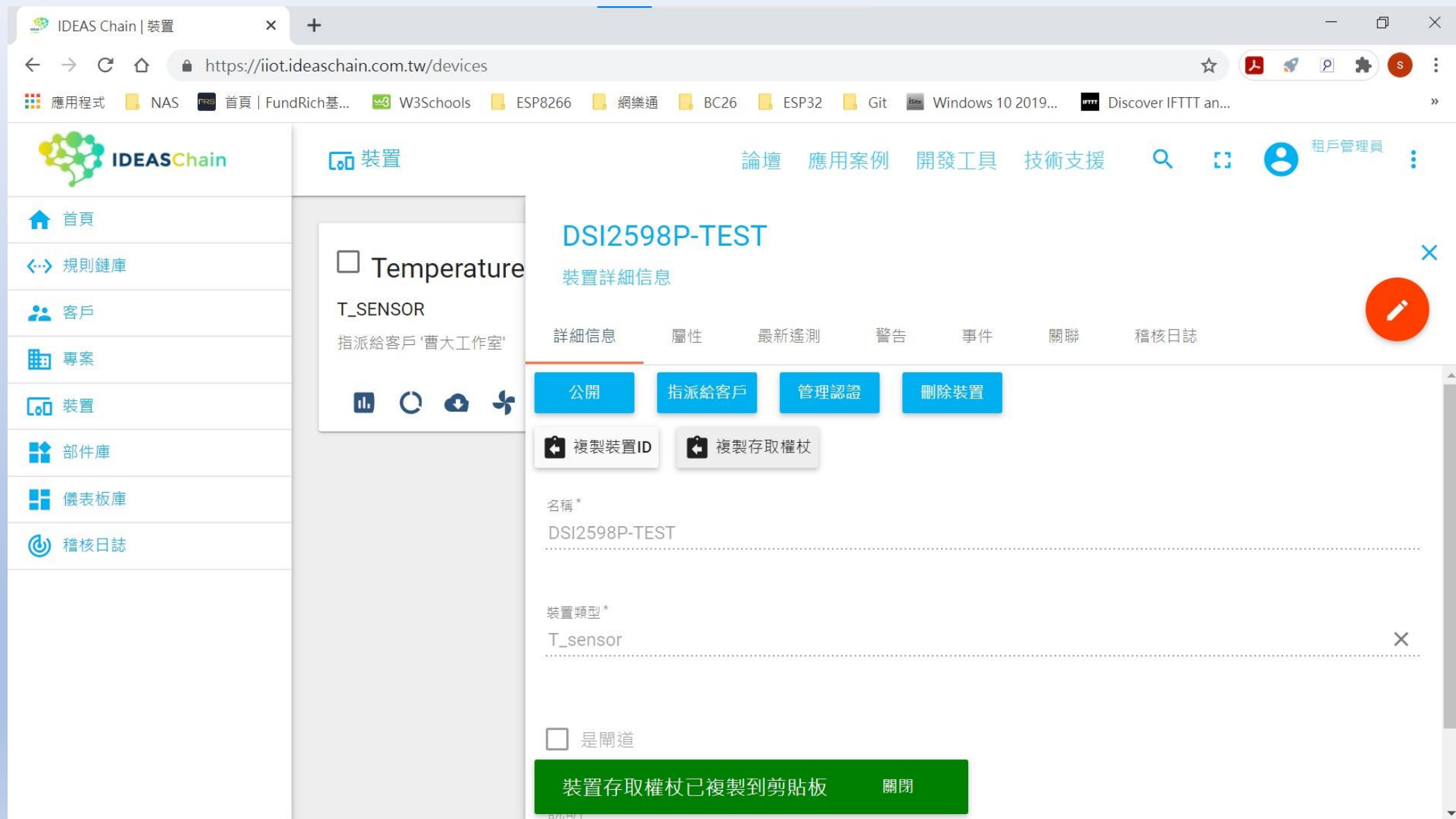
OK
```

Ideaschain 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>

於網站內先行設定下列資訊：第一步驟

The screenshot displays the '裝置' (Devices) page in the IdeasChain IoT platform. The browser address bar shows the URL <https://iiot.ideaschain.com.tw/devices>. The page features a left-hand navigation menu with options like '首頁', '規則鍵庫', '客戶', '專案', '裝置', '部件庫', '儀表板庫', and '稽核日誌'. The main content area shows three device configuration cards. The first card is for 'Temperature Sensor' (T_SENSOR), the second for 'DSI2598' (TH_SENSOR), and the third for 'DSI2598P-TEST' (TH_SENSOR). The third card is highlighted with a red border. Each card includes a checkbox, a name, a description, and a set of control icons at the bottom.

第二步驟：複製存取權杖，以便將權杖資料貼上程式



The screenshot displays the IDEASChain IoT management console. The main content area shows the details for a device named 'DSI2598P-TEST'. The device is assigned to the customer '曹大工作室' (Cao Da Gong Shi). The device type is 'T_sensor'. The interface includes several action buttons: '公開' (Public), '指派給客戶' (Assign to Customer), '管理認證' (Manage Authentication), and '刪除裝置' (Delete Device). Two buttons for copying information are visible: '複製裝置ID' (Copy Device ID) and '複製存取權杖' (Copy Access Token). A green notification banner at the bottom indicates that the device access token has been successfully copied to the clipboard.

IDEASChain | 裝置

裝置

論壇 應用案例 開發工具 技術支援

租戶管理員

DSI2598P-TEST

裝置詳細信息

Temperature

T_SENSOR

指派給客戶 '曹大工作室'

詳細信息 屬性 最新遙測 警告 事件 關聯 稽核日誌

公開 指派給客戶 管理認證 刪除裝置

複製裝置ID 複製存取權杖

名稱*
DSI2598P-TEST

裝置類型*
T_sensor

是開道

裝置存取權杖已複製到剪貼板 關閉

第三步驟：確認客戶端屬性沒有任何鍵值

The screenshot shows the IDEASChain IoT management interface. The browser address bar indicates the URL is <https://iiot.ideaschain.com.tw/devices>. The page title is "DSI2598P-TEST" and the sub-header is "裝置詳細信息". The "屬性" (Attributes) tab is selected, showing a table for "客戶端屬性" (Client Attributes). The table is currently empty, with a search icon in the top right corner. A modal window for "Temperature" is open, showing "T_SENSOR" and "指派給客戶 '曹大工作室'". The table footer shows "Page: 1", "Rows per page: 5", and "0-0 of 0".

IDEASChain | 裝置

裝置 論壇 應用案例 開發工具 技術支援

租戶管理員

DSI2598P-TEST

裝置詳細信息

詳細信息 屬性 最新遙測 警告 事件 關聯 稽核日誌

裝置屬性範圍
客戶端屬性

客戶端屬性

鍵	值
最後更新時間	

Page: 1 Rows per page: 5 0-0 of 0

3.1. ATI Display Product Identification Information

The execution command returns product identification information.

ATI Display Product Identification Information	
Execution Command ATI	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

ATE<value>

Response

OK

Maximum Response Time

300ms

Parameter

<value> 0 Echo mode OFF

1 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

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：BC26-HTTP.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 20: // Connect HTTP Server
C_temp="+QIOPEN: 0,0";
break;
}
```

Send_ATcommand

(第一部分)


```
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

```
bool BC26init() // 初始化 BC26
{
    Send_ATcommand("AT+QGACT=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;
}
```

初始化 DSI2598+

BC26Init()

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;
}
```

送資料至網站 Send_Data_Ideaschain()

```
bool Send_Data_IdeasChain(String token,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 " + token + " HTTP/1.1");
Serial1.println("POST " + token + " HTTP/1.1");
Serial.println("Host: iiot.ideaschain.com.tw");
Serial1.println("Host: iiot.ideaschain.com.tw");
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);
```

第一部分

送資料至網站 Send_Data_Ideaschain()

```
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 true;
}
```

第二部分

將讀到 Ideaschain 網站的屬性資料，進行分解取出 DEC_data()

```
//===== Get =====  
String DEC_data (String input)  
{  
    int index = input.indexOf('{');  
    int x = input.substring(0, index).toInt();  
    String temp = input.substring(index, input.length());  
    //Serial.println(temp);  
    index = temp.indexOf(':');  
    x = temp.substring(0, index).toInt();  
    temp = temp.substring(index + 1, temp.length());  
    //Serial.println(temp);  
    index = temp.indexOf(':');  
    x = temp.substring(0, index).toInt();  
    temp = temp.substring(index + 1, temp.length());  
    //Serial.println(temp);  
    index = temp.indexOf('}');  
    temp = temp.substring(0, index);  
    return temp;  
}
```

```
String Get_QIRD_DATA() // 取得回傳資料
```

```
{  
  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();  
  if (S_temp.length()>30)  
  {  
    Serial.println(S_temp);  
    return (DEC_data(S_temp));  
  }  
  else return "Empty";  
}
```

取得網站回傳資料
Get_QIRD_DATA()

取得的 Ideaschain 屬性資料 Get_Data_Ideaschain()

```
String Get_Data_IdeasChain(byte mode,String token,String key) // HTTP 方式取得網站資料
{
String Send_check="";
String Get_data="";
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();
if (mode==0)
{
Serial.println("GET " + token + "?sharedKeys=" + key + " HTTP/1.1");
Serial1.println("GET " + token + "?sharedKeys=" + key + " HTTP/1.1");
}
if (mode==1)
{
Serial.println("GET " + token + "?clientKeys=" + key + " HTTP/1.1");
Serial1.println("GET " + token + "?clientKeys=" + key + " HTTP/1.1");
}
Serial.println("Host: iiot.ideaschain.com.tw");
Serial1.println("Host: iiot.ideaschain.com.tw");
Serial1.print("\r\n\r\n\x1A");
Send_check=Check_RevData();
//Serial.println(Send_check);
```

第一部分

取得的 Ideaschain 屬性資料 Get_Data_Ideaschain()

```
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();
}
```

第二部分

主程式 : ideaschain-HTTP.ino

```
#include "BC26-HTTP.h"

String Server_Name="iiot.ideaschain.com.tw"; //HTTP Server 的 IP 位址
String Access_Token="k2L4RyvCQHKsPAoEFuHw"; //使用者資訊 ==> IDEAS Chain存取權杖
String Attrib_Key="TestValue"; //屬性的 Key
String Attrib_Data_String="telemetry";
String Attrib_Client_String="attributes";
String DATA_Attrib;
String DATA_Message;
byte Attrib_Mode=1; // 0 -> sharedKeys ; 1 -> clientKeys
int test_value= 80;

// ** iiot.ideaschain.com.tw/api/v1/$ACCESS_TOKEN/telemetry //時間序列資料上傳
// ** iiot.ideaschain.com.tw/api/v1/$ACCESS_TOKEN/attributes //上傳客戶端設備屬性值
// ** iiot.ideaschain.com.tw/api/v1/$ACCESS_TOKEN/attributes?clientKeys=key 取得客戶端設備屬性值
// ** Return of Data Format "{clientKeys:{"key":"value"}}"
```

設定通訊協定，初始化 BC26

```
void setup()
{
  Serial.begin(115200);
  Serial1.begin(115200);
  pinMode(PC13,OUTPUT);

  // Rest BC26 ===
  digitalWrite(PC13, LOW);
  delay(30);
  digitalWrite(PC13, HIGH);
  delay (3000);
  // =====
  if (!BC26init()) {delay (10000); nvic_sys_reset();}
  Serial.println("initialization OK ....");
}
```

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)

nvic_sys_reset() 是STM32
專用的重置函式

連線 Ideaschain 網站後，傳送TestValue數值至網站的客戶端屬性

```
void loop()
{
  Serial.println("Star Send Data ....");
  if (Connect_HTTP_Server(Server_Name))
  {
    DATA_Attrib="/api/v1/" + Access_Token + "/" + Attrib_Client_String;
    DATA_Message="{\" + Attrib_Key + "\":\" + String(test_value) + "\"}";
    if (Send_Data_IdeasChain(DATA_Attrib,DATA_Message))
      Serial.println("Data Send OK !!");
    Close_HTTP();
  }
  else Serial.println("Data Send Fail .....");

  Serial.println("\n\nDelay 10 second ..... \n\n");
  delay (10000);
}
```

取回網站客戶端屬性的數值

```
Serial.println("Star Get Data ....");

String G_data;
if (Connect_HTTP_Server(Server_Name))
{
  DATA_Attrib="/api/v1/" + Access_Token + "/" + Attrib_Client_String;
  G_data=Get_Data_IdeasChain(Attrib_Mode,DATA_Attrib,Attrib_Key);
  if (G_data!="Empty")
    Serial.println("\nGet " + Attrib_Key + " of Data is : " + G_data);
  Rset_Count=0;
}
Close_HTTP();
Serial.println("Data Get OK !!");
delay(3000);
exit(0);
}
```