



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. MQTT AT_Command 概略說明
5. MQTT 程式(函式)使用說明
6. 溫溼度範例使用說明及線路圖

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：

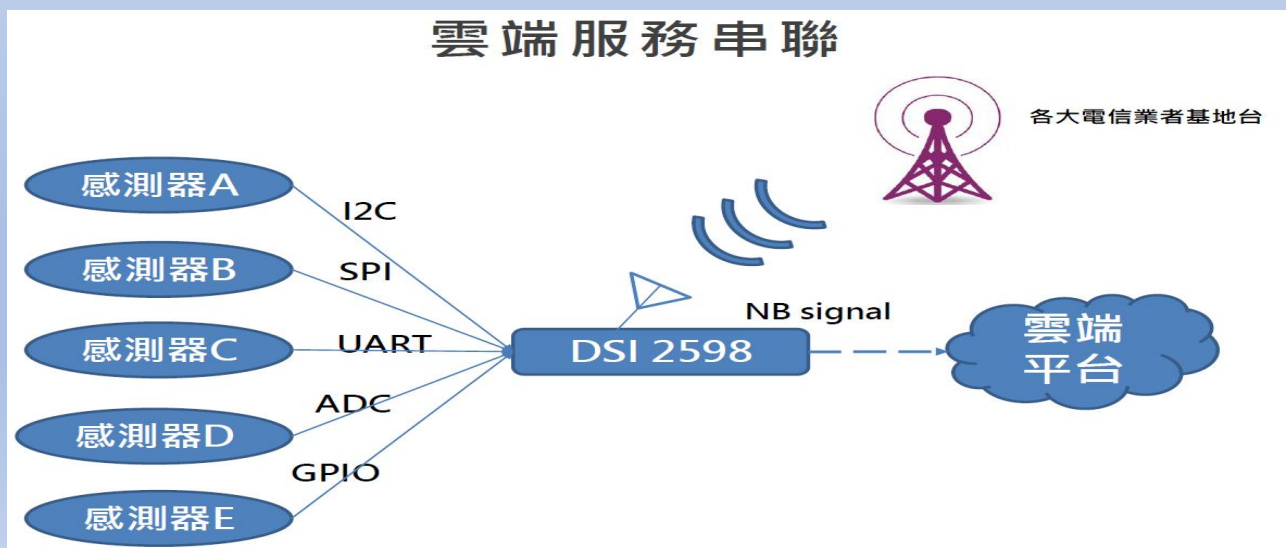
1. WiFi 透過無線基地台連上網際網路，NB-IoT 利用電信基地台連上網際網路。
2. WiFi 適用傳輸大量資料的訊息，NB-IoT 適用小資料量傳輸。
3. WiFi 連接無線基地台的距離較短，NB-IoT 由於全台基地台涵蓋率夠高，幾乎無死角。
4. WiFi 晶片耗用功率較高，NB-IoT 採用低功率晶片，使用一般 AA 電池可達 3-5 年以上。
5. WiFi 連網較易取得真實IP 位址，而NB-IoT使用的電信基地台提供的IP 位址大都為 虛擬 IP 網段。

DSI 2598+

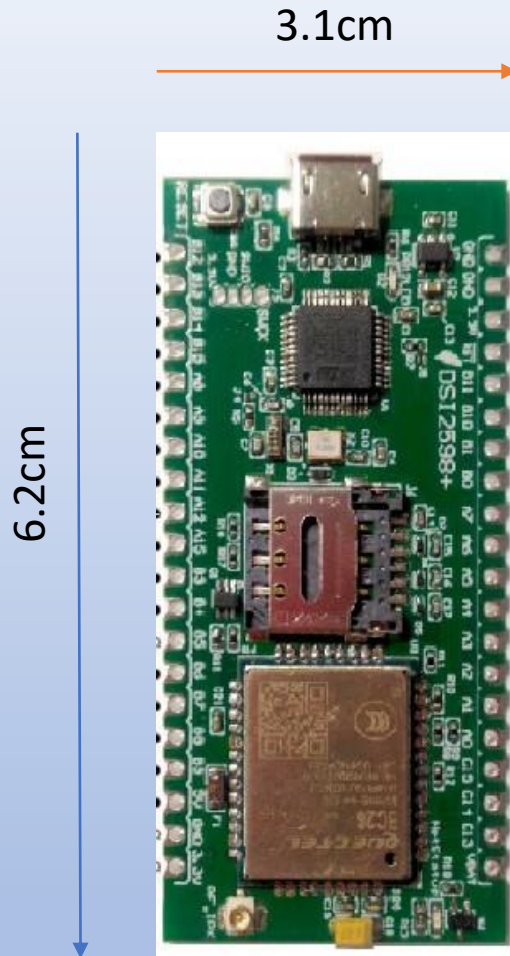
採用MT 2625

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。

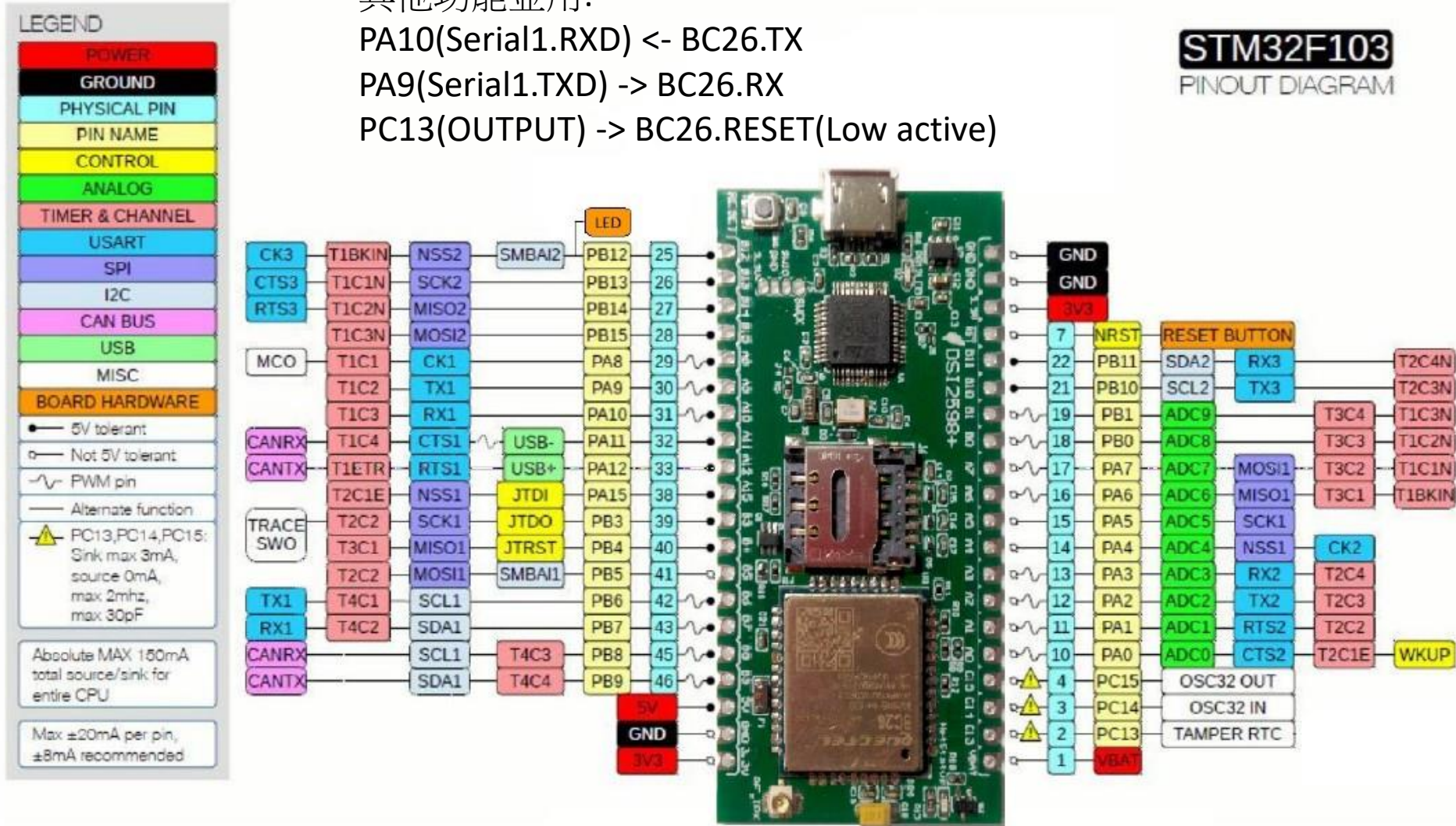
DSI2598+ Arduino 開發環境

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

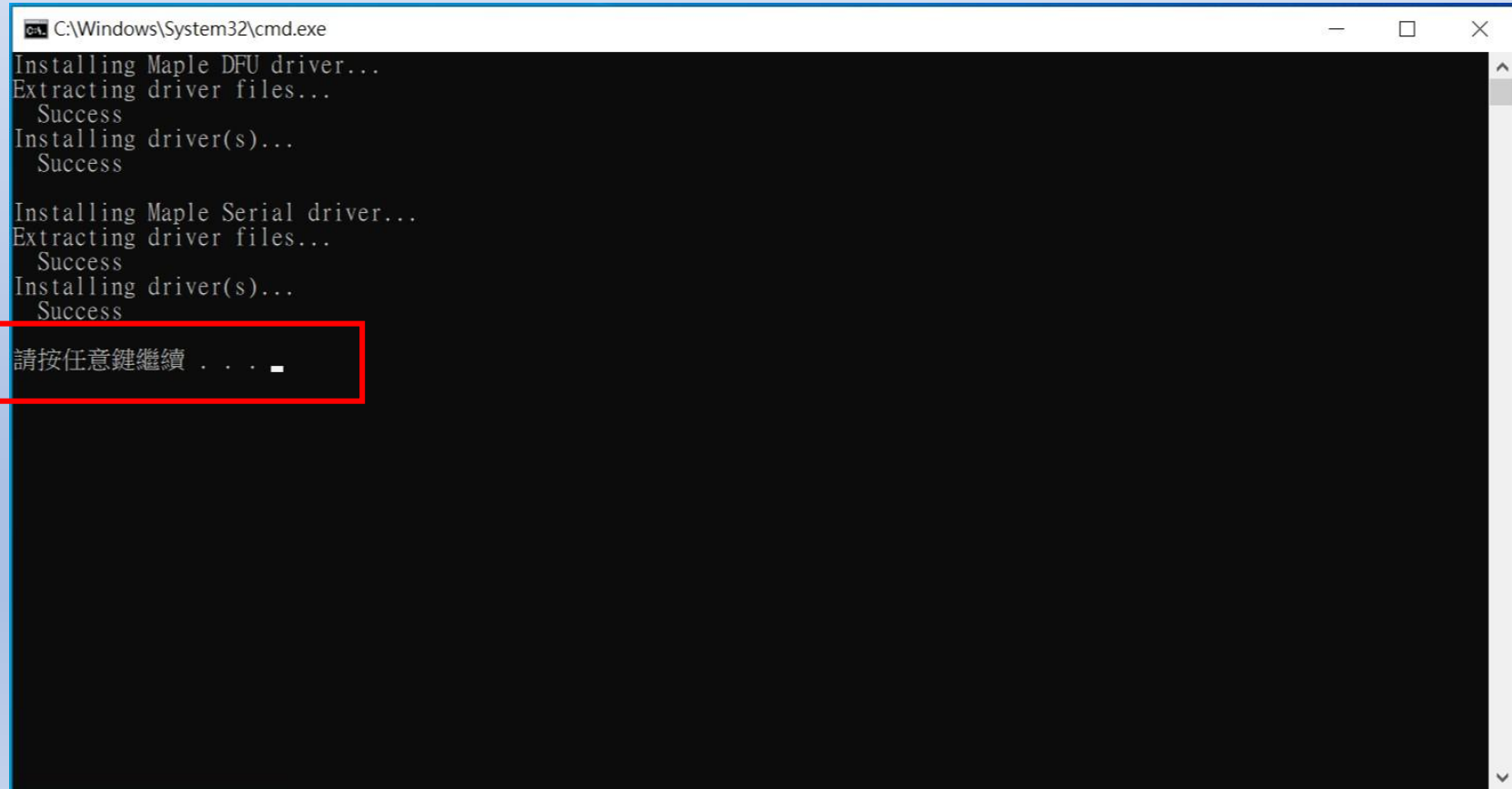
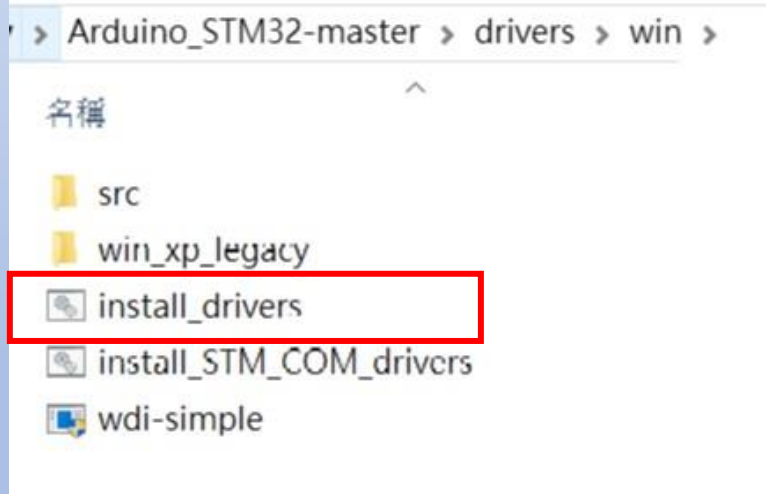


設定Arduino DSI2598+開發板的環境: (for Windows 10 作業系統)

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

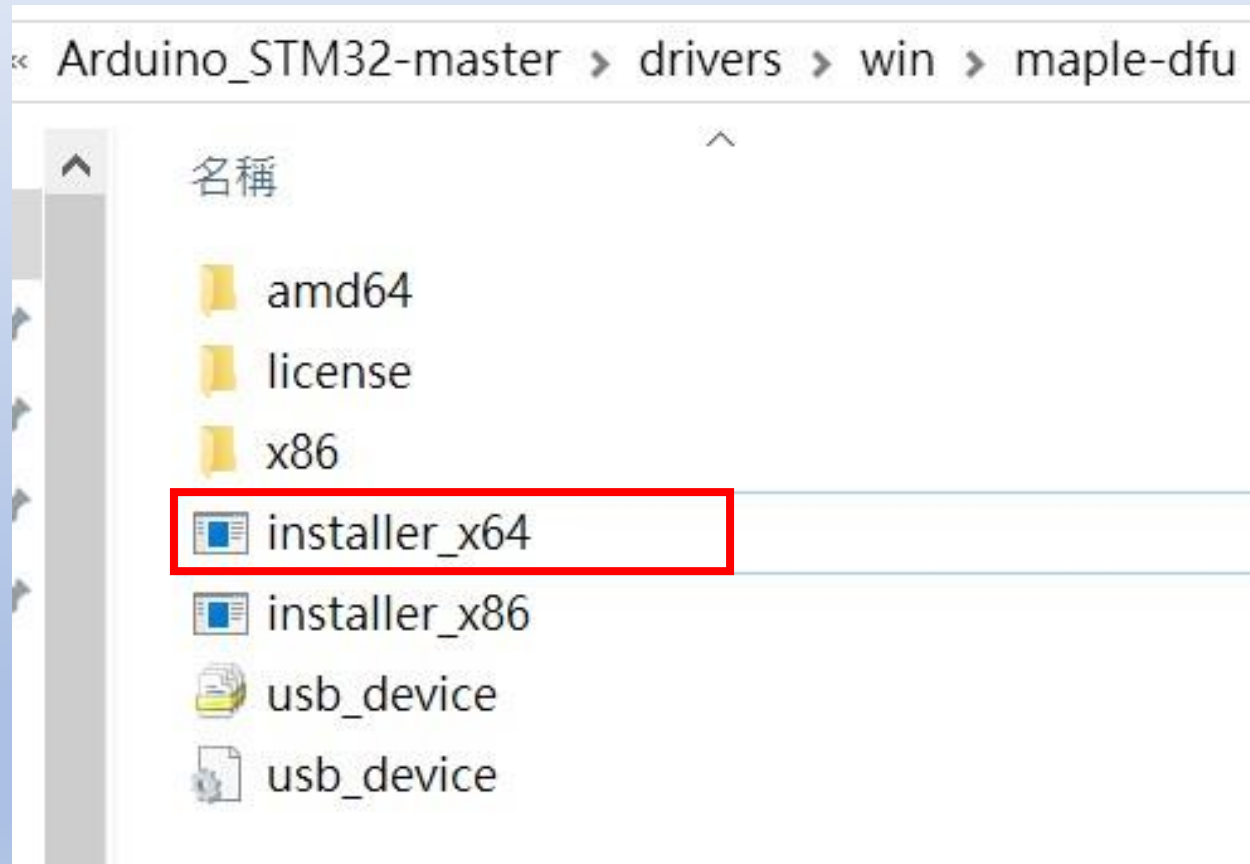


```
C:\Windows\System32\cmd.exe
Installing Maple DFU driver...
Extracting driver files...
Success
Installing driver(s)...
Success

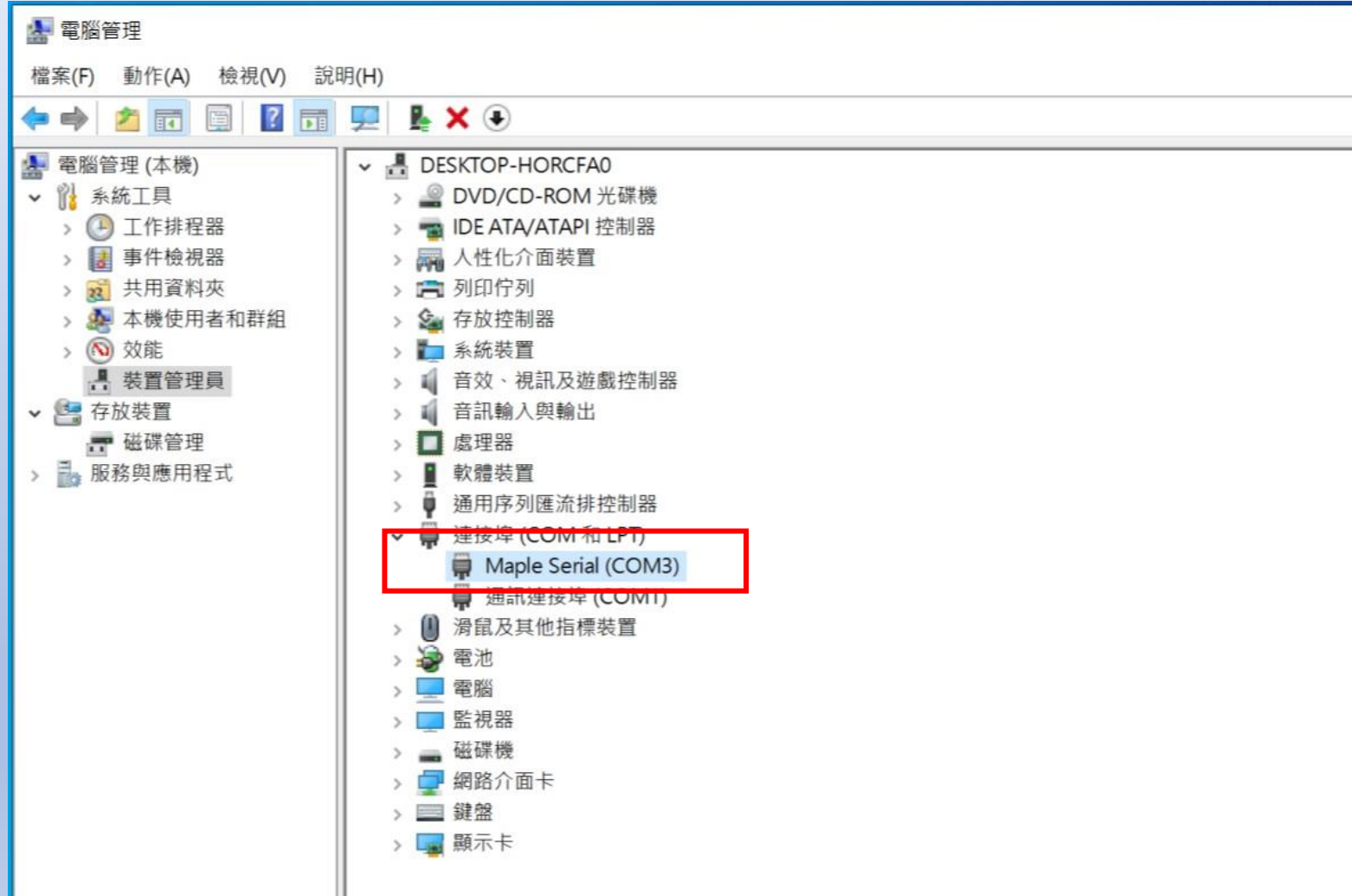
Installing Maple Serial driver...
Extracting driver files...
Success
Installing driver(s)...
Success

請按任意鍵繼續 . . . .
```

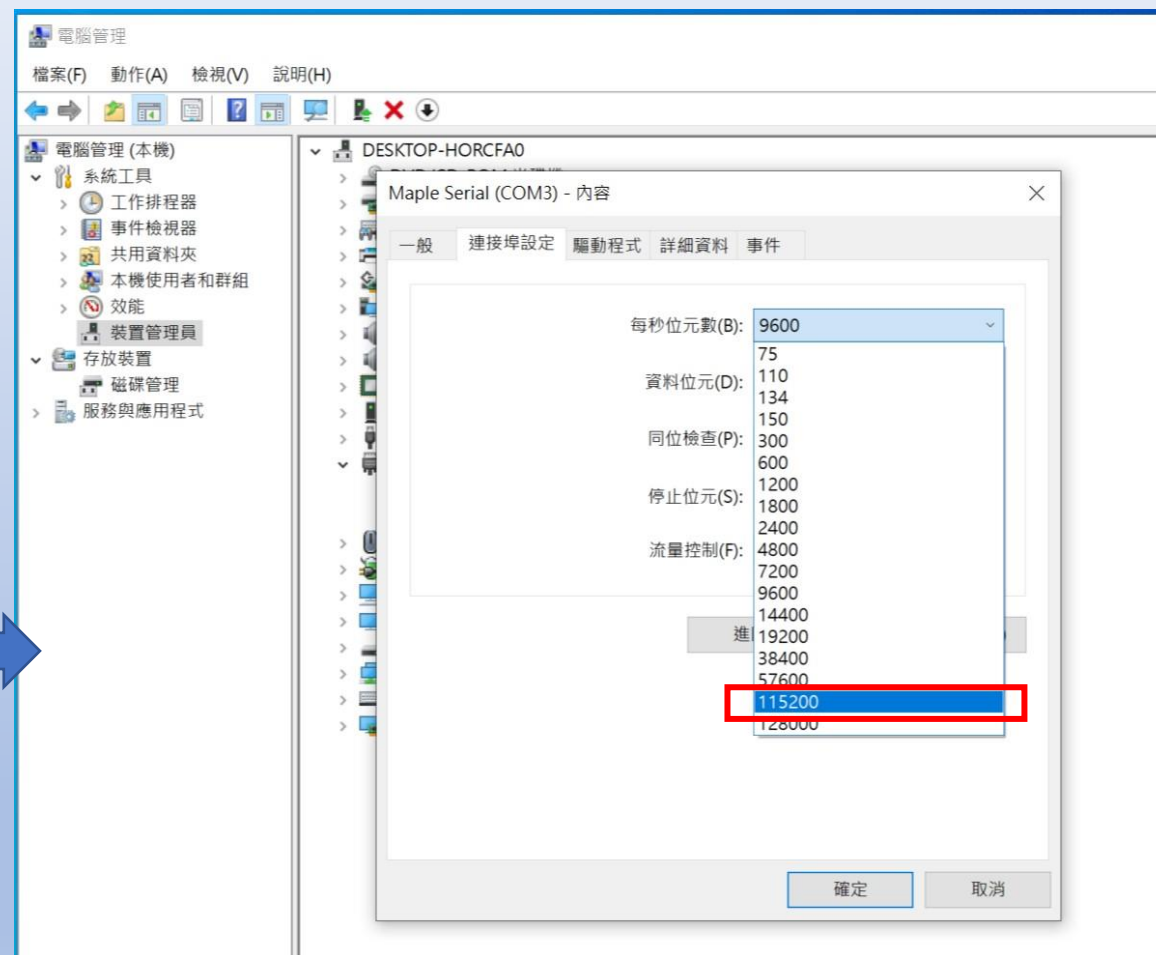
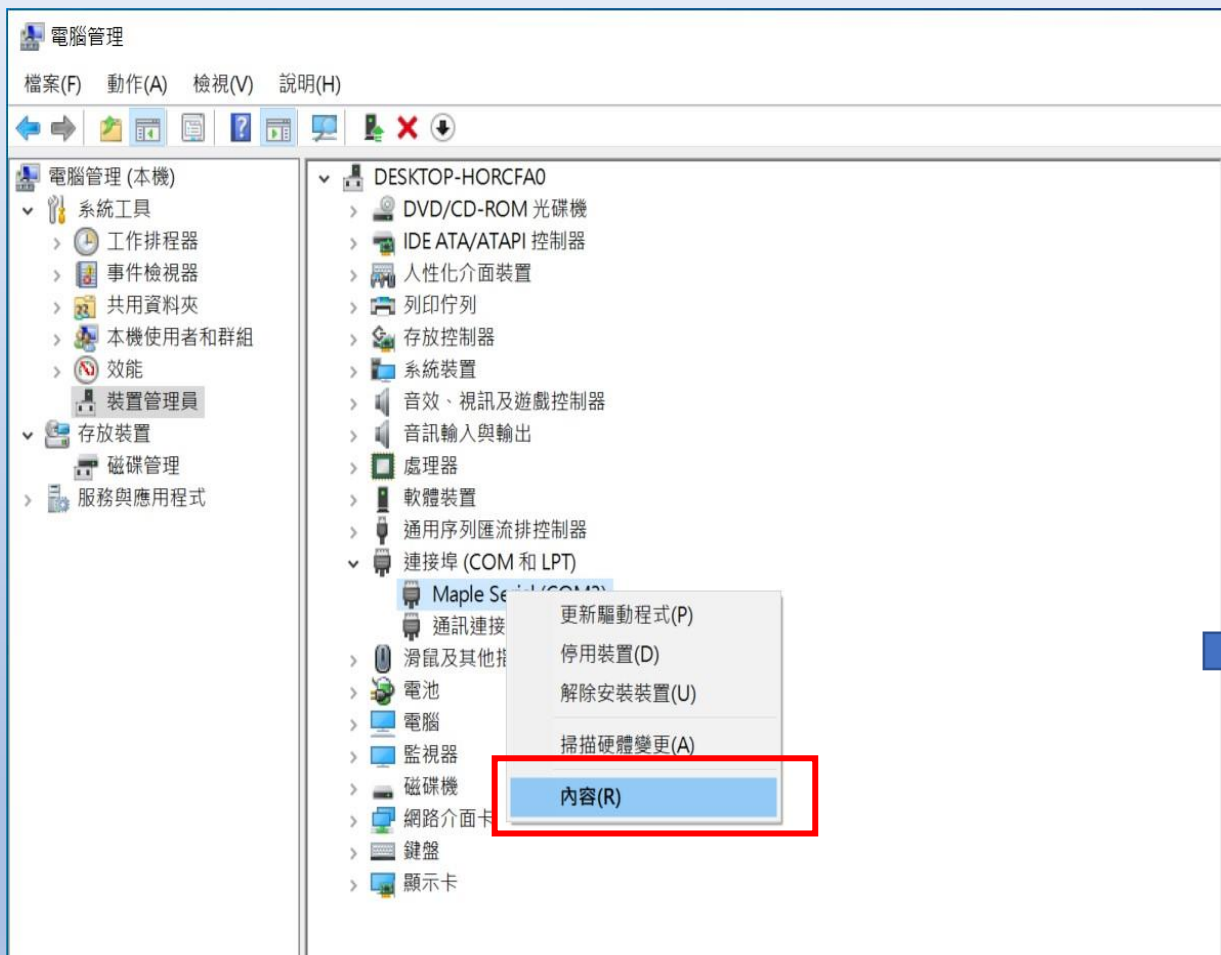
2. 在 `Arduino_STM32-master\drivers\win\maple-dfu` 下用系統管理者執行 `installer_x64.exe` (若系統為 32 位元 請執行 `installer_x86.exe`)，如下畫面：



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：

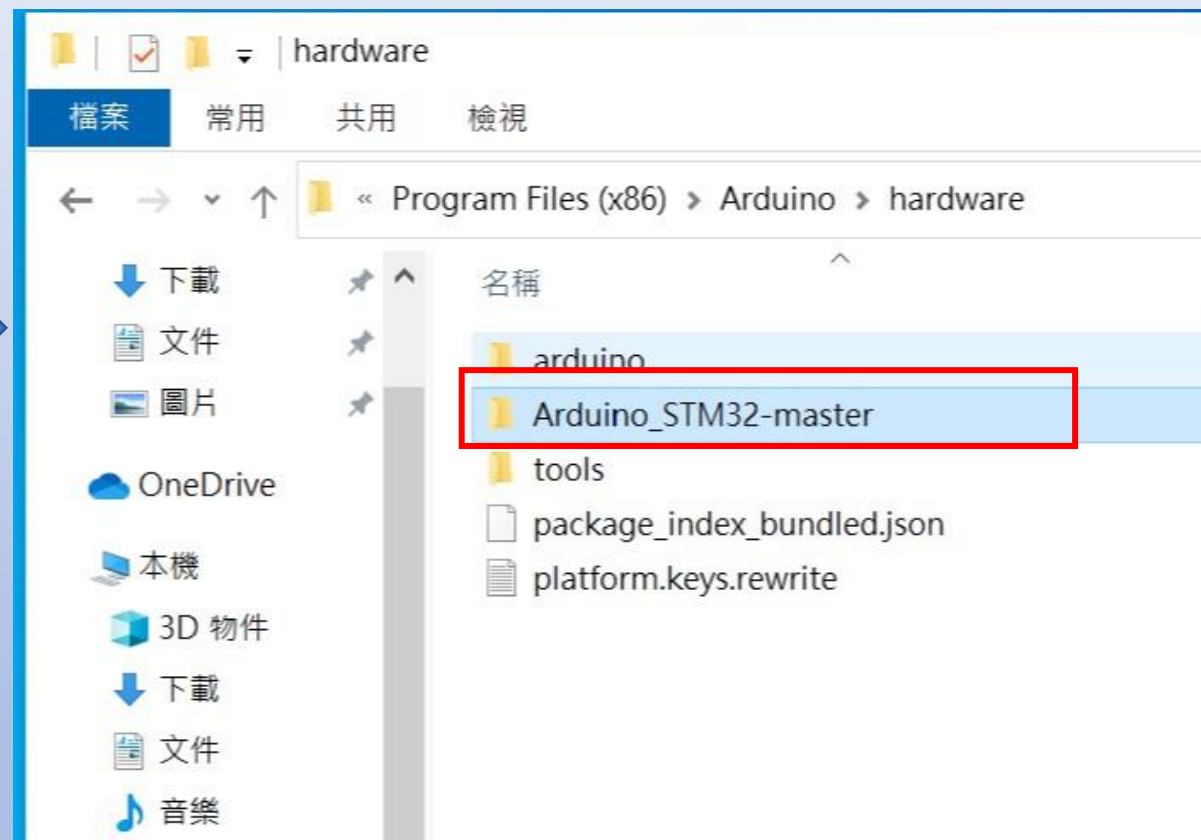
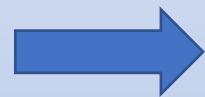
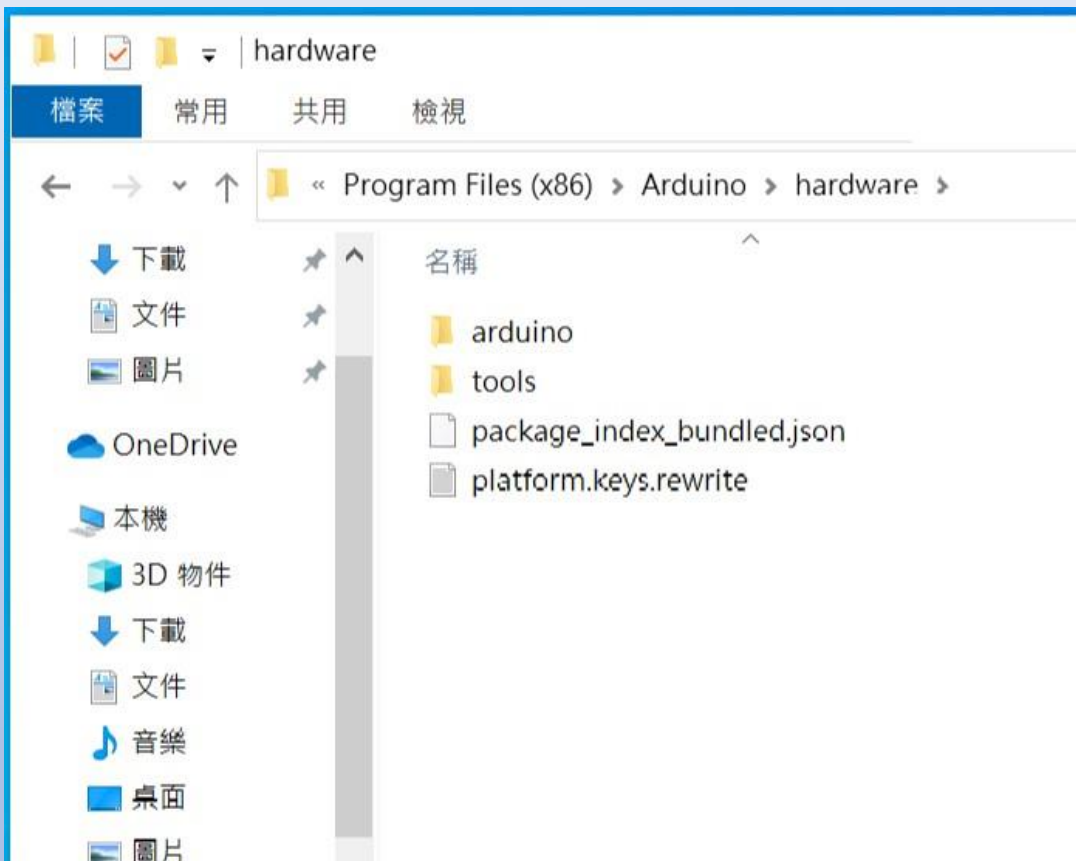


4.更改傳輸速率，加快傳輸動作！！

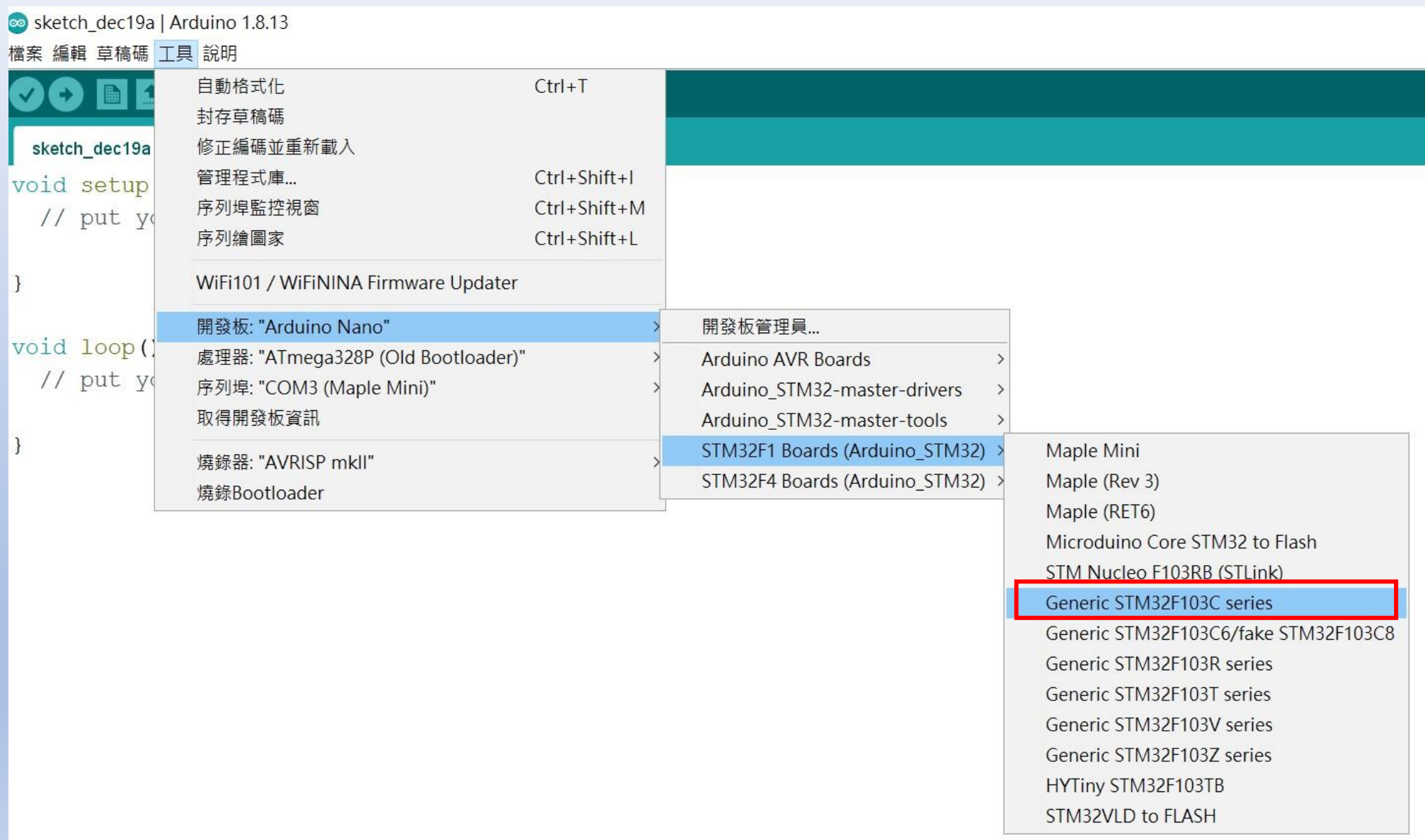


完成驅動程式的安裝

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：



The screenshot shows the Arduino IDE interface with the 'Tools' menu open. The menu items are as follows:

- 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-Fi NINA 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

The settings for the development board, variant, upload method, CPU speed, and optimization are highlighted with a red box. A blue arrow points from the warning box to this red box.



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

The screenshot shows the Arduino IDE interface with the 'Tools' menu open. The 'Board' menu item is selected, and a sub-menu is displayed. The sub-menu contains the following items:

- 開發板: "Generic STM32F103C series" (selected)
- 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

The sub-menu also includes a '開發板管理員...' option, which is highlighted. Below this, a list of board packages is shown:

- Arduino AVR Boards
- Arduino_STM32-master-drivers
- Arduino_STM32-master-tools
- STM32F1 Boards (Arduino_STM32)
- STM32F4 Boards (Arduino_STM32)



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。

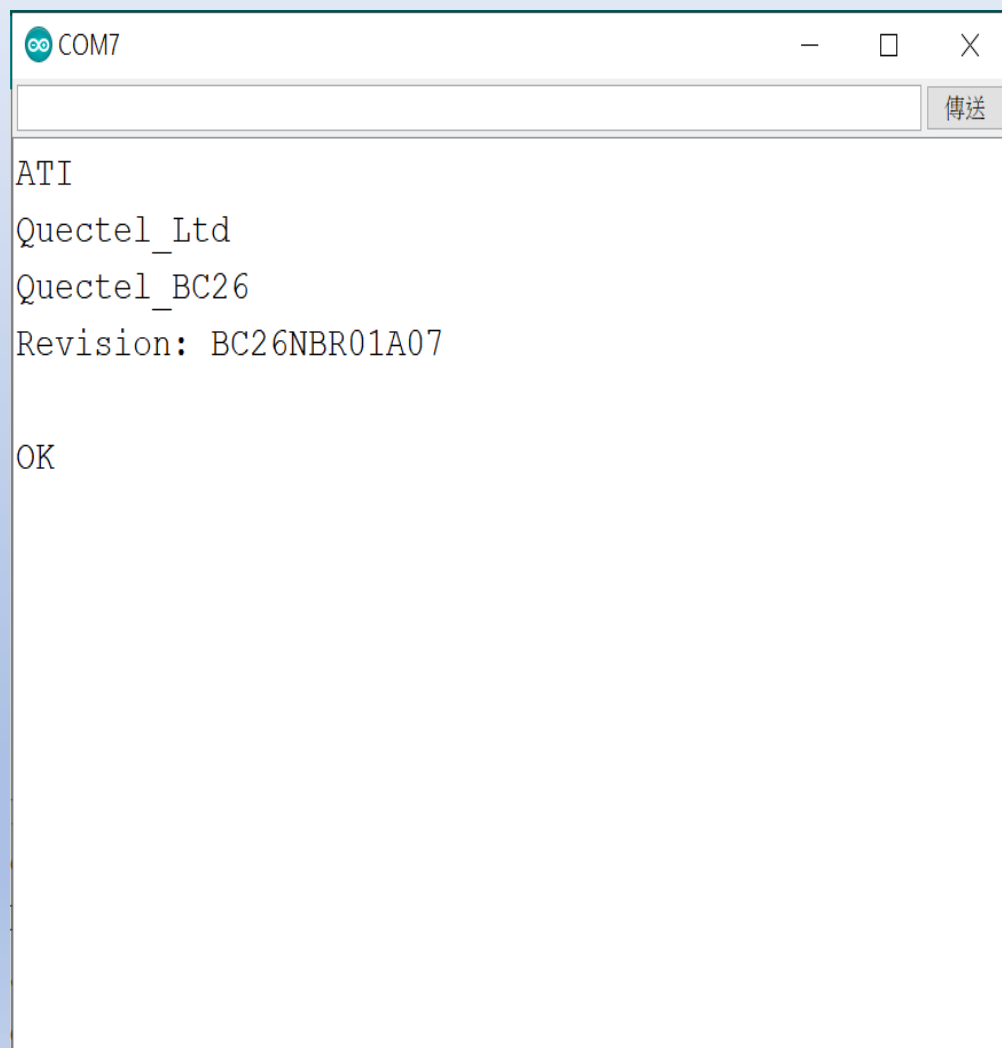
STEP 2: 選擇 工具 ->開發板->Generic STM32F103C series，然後按下上傳 (Ctrl+U)，將
程式燒錄進去。



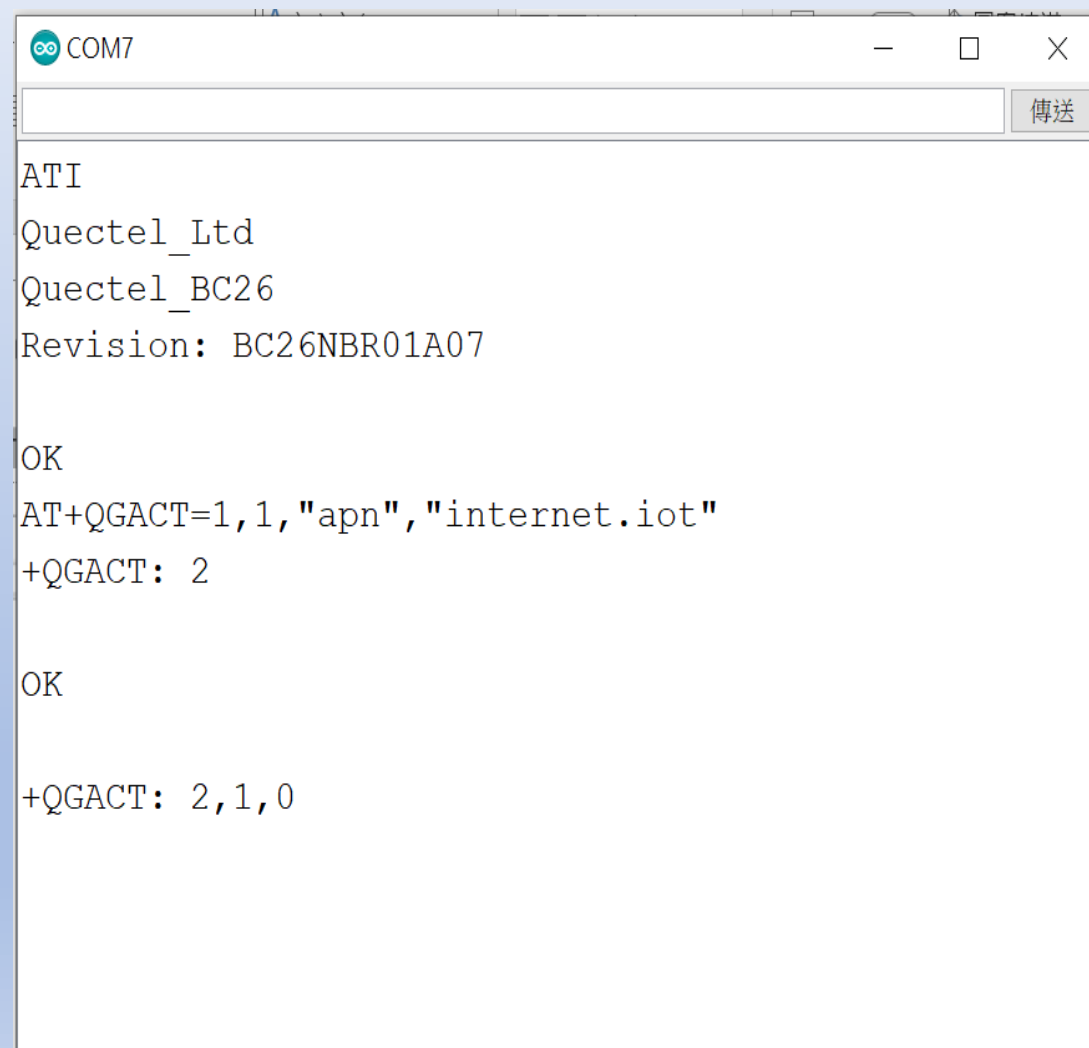
STEP 3：打開序列埠監控視窗，在上方輸入欄中輸入ATI 指令，可先輸入「ATI」，查看模組是否有回覆版本訊息。

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

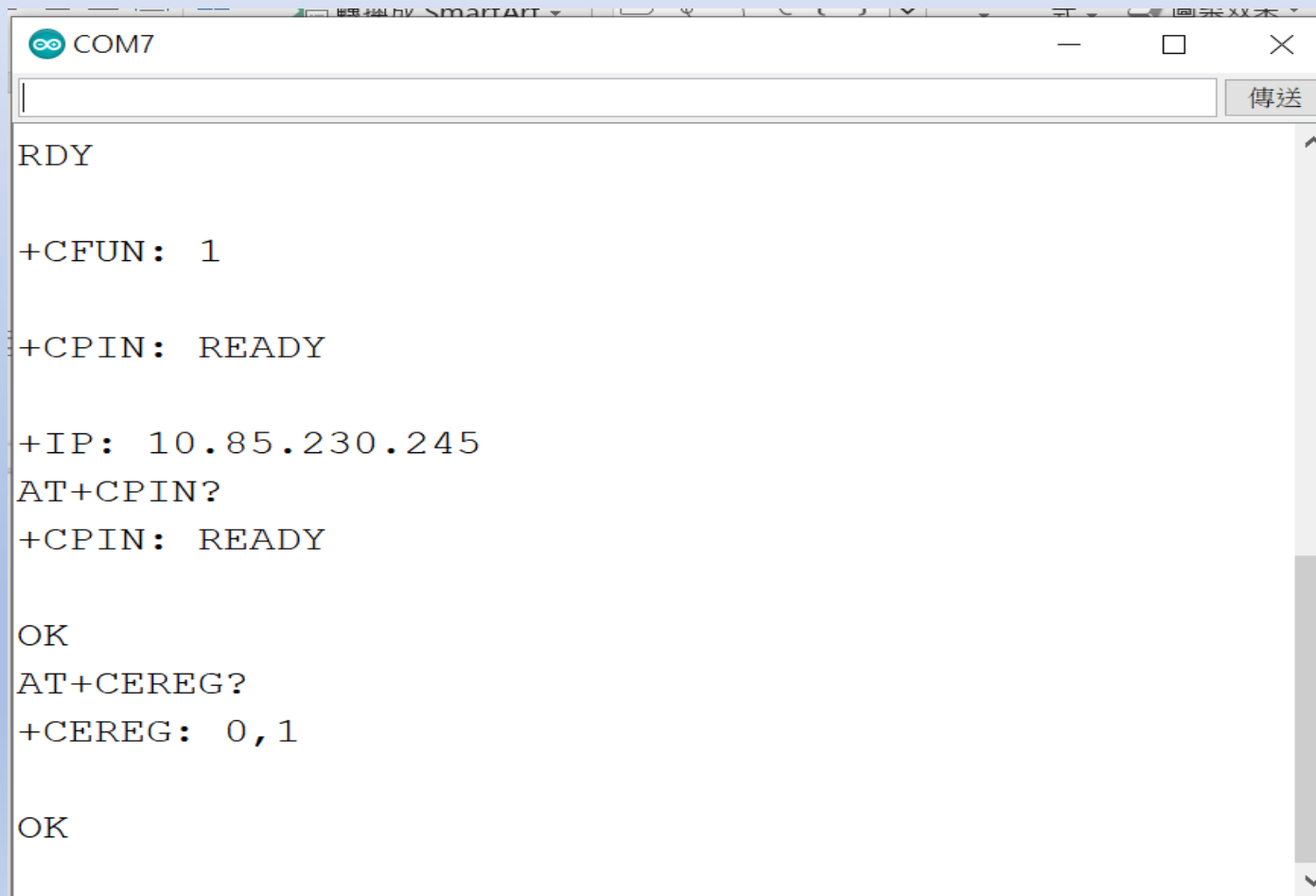
STEP 2：

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:

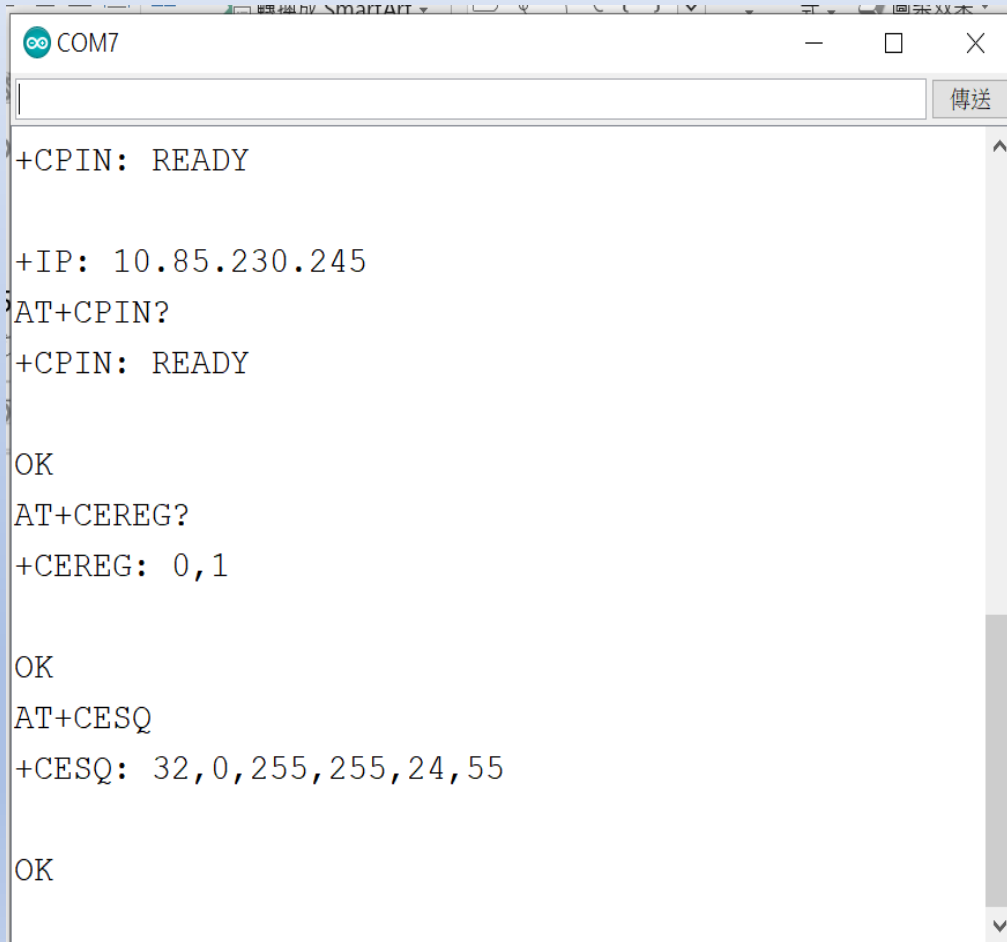
```
RDY  
  
+CFUN: 1  
  
+CPIN: READY  
  
+IP: 10.85.230.245  
AT+CPIN?  
+CPIN: READY  
  
OK  
AT+CEREG?  
+CEREG: 0,1  
  
OK
```

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

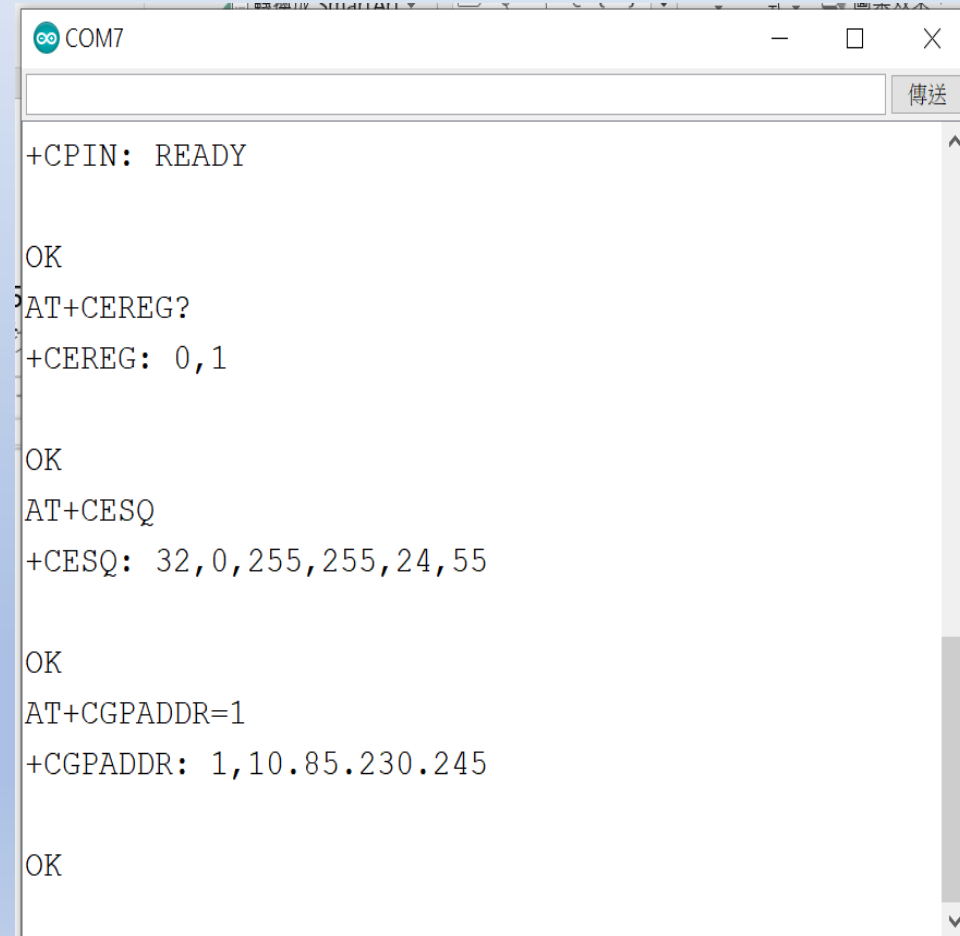
OK
AT+CESQ
+CESQ: 32,0,255,255,24,55

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) assigned to '曹大工作室'. The second card is for 'DSI2598' (TH_SENSOR). The third card, 'DSI2598P-TEST' (TH_SENSOR), is highlighted with a red border and includes a '+' icon in its bottom-left corner, indicating a new configuration step.

第二步驟

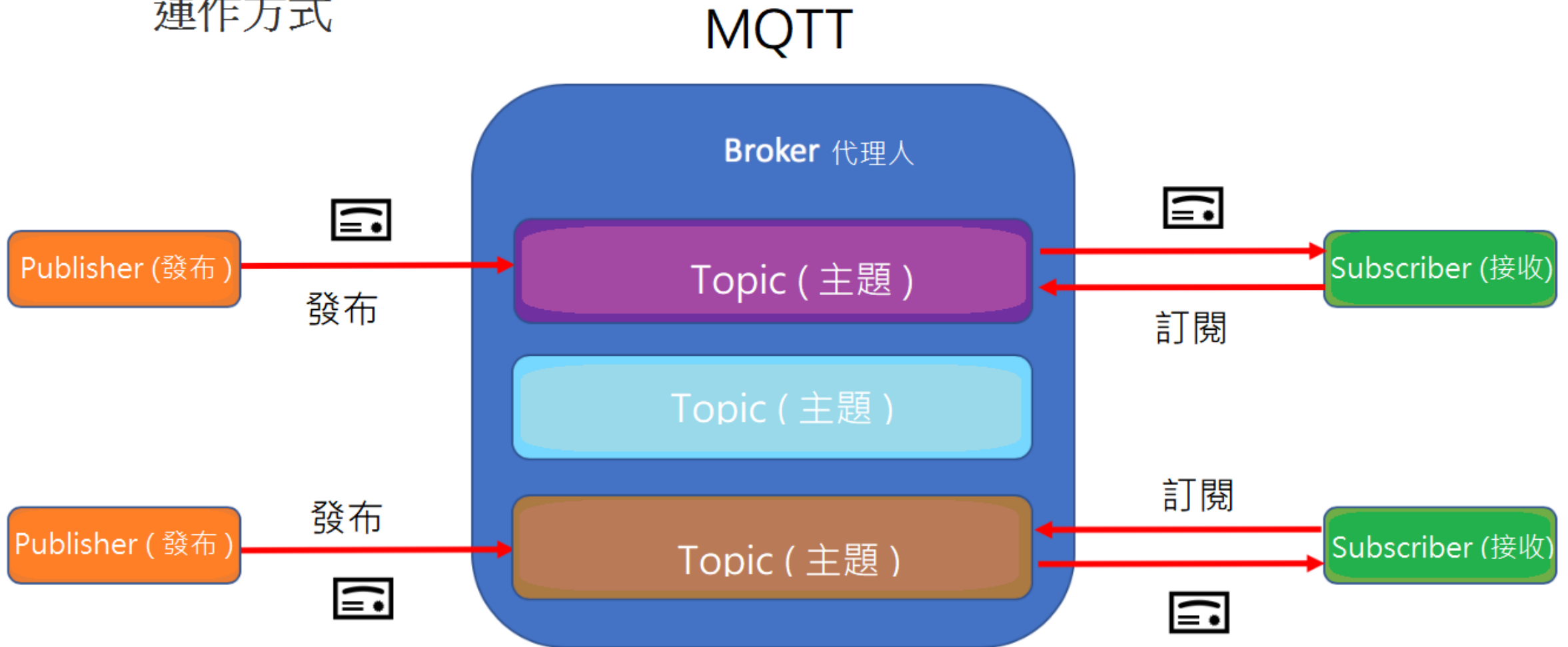
1. 在詳細信息中先複製存取權杖，方便後續程式使用 !!
2. 於共享屬性設定上先新增資料如下： DelayTime => 60 , TempHigh => 30 , TempLow =>20

The screenshot displays the IDEASChain IoT platform interface. The main content area shows the configuration for a device named "DSI2598P-TEST". The "Shared Properties" (共享屬性) section is highlighted with a red box, showing a table with columns for "Last Update Time" (最後更新時間), "Key" (鍵), and "Value" (值). The table contains three entries:

最後更新時間	鍵	值
2020-12-15 16:03:57	DelayTime	60
2020-12-08 21:20:05	TempHigh	30
2020-12-08 21:20:09	TempLow	20

MQTT 基本傳輸圖示

運作方式



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.

3.2.2. AT+QMTOPEN Open a Network for MQTT Client

The command is used to open a network for MQTT client.

AT+QMTOPEN Open a Network for MQTT Client

Test Command AT+QMTOPEN=?	Response +QMTOPEN: (list of supported <tcpconnectID>s),“ <host_name> ”,(list of supported <port>s) OK
Read Command AT+QMTOPEN?	Response [+QMTOPEN: <tcpconnectID>,”<host_name>”,<port>] OK
Write Command AT+QMTOPEN=<tcpconnectID>,”<host_name>”,<port>	Response OK +QMTOPEN: <tcpconnectID>,<result> If there is an error related to ME functionality: +CME ERROR: <err>
Maximum Response Time	75s, determined by network

3.2.4. AT+QMTCONN Connect a Client to MQTT Server

The command is used when a client requests a connection to MQTT server. When a TCP/IP socket connection is established from a client to a server, a protocol level session must be created using a CONNECT flow.

AT+QMTCONN Connect a Client to MQTT Server	
Test Command AT+QMTCONN=?	Response +QMTCONN: (list of supported <tcpconnectID>s), “<clientID>” [, “<username>”][, “<password>”] OK
Read Command AT+QMTCONN?	Response [+QMTCONN: <tcpconnectID>,<state>] OK
Write Command AT+QMTCONN=<tcpconnectID>,”<clientID>” [, “<username>”][, “<password>”] OK	Response +QMTCONN: <tcpconnectID>,<result>[,<ret_code>] If there is an error related to ME functionality: +CME ERROR: <err>
Maximum Response Time	<pkt_timeout> (default 10s), determined by network

3.2.6. AT+QMTSUB Subscribe to Topics

The command is used to subscribe to one or more topics. A SUBSCRIBE message is sent by a client to register an interest in one or more topic names with the server. Messages published to these topics are delivered from the server to the client as PUBLISH messages.

AT+QMTSUB Subscribe to Topics

Test Command
AT+QMTSUB=?

Response

+QMTSUB: (list of supported <tcpconnectID>s),(list of supported <msgID>s),“<topic>”,(list of supported <qos>s)

OK

Write Command
AT+QMTSUB=<tcpconnectID>,<msgID>,”<topic1>”,<qos1>[,”<topic2>”,<qos2>...]

Response

OK

+QMTSUB: <tcpconnectID>,<msgID>,<result>[,<value>]

If there is an error related to ME functionality:

+CME ERROR: <err>

Maximum Response Time

<pkt_timeout> * <retry_times> (default 40s), determined by network

3.2.8. AT+QMTPUB Publish Messages

The command is used to publish messages by a client to a server for distribution to interested subscribers. Each PUBLISH message is associated with a topic name. If a client subscribes to one or more topics, any message published to those topics are sent by the server to the client as a PUBLISH message.

AT+QMTPUB Publish Messages

Test Command
AT+QMTPUB=?

Response

+QMTPUB: (list of supported <tcpconnectID>s),(list of supported <msgID>s),(list of supported <qos>s),(list of supported <retain>s),“<topic>”,“<msg>”

OK

Write Command
AT+QMTPUB=<tcpconnectID>,<msgID>,<qos>,<retain>,”<topic>”,“<msg>”

Response

OK

+QMTPUB: <tcpconnectID>,<msgID>,<result>[,<value>]

If there is an error related to ME functionality:

+CME ERROR: <err>

Maximum Response Time

<pkt_timeout> * <retry_times> (default 40s), determined by network

3.2.3. AT+QMTCLOSE Close a Network for MQTT Client

The command is used to close a network for MQTT client.

AT+QMTCLOSE Close a Network for MQTT Client

Test Command

AT+QMTCLOSE=?

Response

+QMTCLOSE: (list of supported **<tcpconnectID>**s)

OK

Write Command

AT+QMTCLOSE=<tcpconnectID>

Response

OK

+QMTCLOSE: **<tcpconnectID>**,**<result>**

If there is an error related to ME functionality:

+CME ERROR: **<err>**

Maximum Response Time

300ms

函式檔：BC26Init.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(); //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;
}
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+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;
}
```

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;
  //Serial.println(S_temp);
  if (Send_ATcommand(S_temp,10)!=99) return false;
  //delay(100);
  S_temp = "\" + user + "\" + \",\" + "\" + pass + "\"";
  S_temp="AT+QMTCONN=0,0," + S_temp;
  //Serial.println(S_temp);
  if (Send_ATcommand(S_temp,11)!=99) return false;
  //delay(100);
  return true;
}
```

發布資料 Publish_MQTT ()

```
bool Publish_MQTT(String topic, String message) // 發佈資料
{
  String S_temp;
  S_temp = "\"" + topic + "\"" + "," + message ;
  S_temp = "AT+QMTPUB=0,0,0,0," + S_temp ;
  //Serial.println(S_temp);
  if (Send_ATcommand(S_temp,12)!=99) return false;
  //delay(100);
  return true;
}
```



```
bool Sub_MQTT(String topic) // 訂閱資料
{
  String S_temp;
  S_temp = "\" + topic + "\" + "," + "0";
  S_temp = "AT+QMTSUB=0,1," + S_temp;
  //Serial.println(S_temp);
  if (Send_ATcommand(S_temp,13) != 99) return false;
  //delay(100);
  return true;
}
```

訂閱資料 Sub_MQTT () 函式

```
bool Close_MQTT() // 關閉連線
{
  String S_temp;
  S_temp="AT+QMTCLOSE=0";
  if (Send_ATcommand(S_temp,1) != 99) return false;
  //delay(100);
  return true;
}
```

關閉與 MQTT Broker 的連線
Close_MQTT()

將讀到 Ideaschain 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());
    //Serial.println(json);
    index = json.indexOf(':');
    x = json.substring(0, index).toInt();
    json = json.substring(index + 1, json.length());
    //Serial.println(json);
    DynamicJsonDocument doc(1024);
    deserializeJson(doc, json);
    JsonObject obj = doc.as<JsonObject>();
    return obj[findData];
}
```


取出 Ideaschain 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; // Qos 0
  Serial.println(S_temp);
  Serial1.println(S_temp);
  delay (2000);
  return true;
}
```

訂閱後所取得的 Ideaschain 屬性資料 Get_Publish_MQTT()

```
String Get_Publish_MQTT(byte mode,String attreqtopic , String message) // 取得 Ideaschain 屬性資料
{
  String Showmsg;
  String S_temp,T_temp;
  //delay (1000);
  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(100);
    Showmsg=Check_RevData();
    if (Showmsg.length()>30) break;
    //Serial.println(Showmsg);
    if ((StartTime+waitingTime) < millis()) return "error";
  }
  //Serial.println(Showmsg);
  return JSON_DEC_data (Showmsg,message);
}
```

主程式 : DHT11-ideaschain.ino

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

String MQTT_Server="iiot.ideaschain.com.tw"; //MQTT Server 的 IP 位址
String MQTT_Port="1883"; //MQTT 使用的埠
String MQTT_Access_token="AKuyx8mztEYxMGKdVYpF"; //使用者密碼 ==> IDEAS Chain存取權杖
String MQTTtopic="v1/devices/me/telemetry"; //固定路徑 IDEAS Chain
String MQTTmessage="";
String MQTTattrestopic="v1/devices/me/attributes/response/+"; //訂閱資料路徑 IDEAS Chain
String MQTTattreqtopic="v1/devices/me/attributes/request/1"; //訂閱資料路徑 IDEAS Chain
String DelayTime_Text="DelayTime"; // 共享屬性的字串資料來取得暫停傳送的秒數
String TempHigh_Text="TempHigh"; // 共享屬性的字串資料來設定高溫參考值
String TempLow_Text="TempLow"; // 共享屬性的字串資料來設定低溫參考值
byte mode=0; // 0 => 讀取共享屬性資料 1 => 讀取客戶端屬性資料

String DelayTime;
String TempHigh;
String TempLow;

String Sensor1="Temperature";
String Sensor2="Humidity";

// "AT+QMTPUB=0,0,0,0,\"v1/devices/me/telemetry\", \"{ \"key1\":data, \"key2\":data } \""
```



```
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
```

讀取共享屬性資料 Get_ideaschain_data()

```
bool Get_ideaschain_data(String attrestopic, String attreqtopic) //讀取在網站設定的共享屬性資料
{
  Sub_Ideaschain(attrestopic);
  DelayTime=Get_Publish_MQTT(mode,attreqtopic,DelayTime_Text);
  TempHigh=Get_Publish_MQTT(mode,attreqtopic,TempHigh_Text);
  TempLow=Get_Publish_MQTT(mode,attreqtopic,TempLow_Text);

  if (DelayTime.toInt()!=0 && TempHigh.toInt()!=0 && TempLow.toInt()!=0 )
  {
    Serial.println("取得暫停秒數 : " + DelayTime + " 秒");
    Serial.println("取得設定最高溫 : " + TempHigh + " °C");
    Serial.println("取得設定最低溫 : " + TempLow + " °C");
    return true;
  }
  return false;
}
```

設定通訊協定，初始化 BC26

```
void setup()
{
  Serial.begin(115200);
  Serial1.begin(115200);
  pinMode(PB6, OUTPUT);           // 設為程式執行時的確認燈號
  pinMode(PB7, OUTPUT);           // 溫度若低於設定溫度時的確認燈號
  pinMode(PB8, OUTPUT);           // 溫度若高於設定溫度時的確認燈號
  pinMode(PC13, OUTPUT);
  digitalWrite(PB6, HIGH);

  // Rest BC26 ===
  digitalWrite(PC13, LOW);
  delay(30);
  digitalWrite(PC13, 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 Ideaschain 網站後，先讀取網站設定的資料再傳送DHT11最新偵測資料至網站

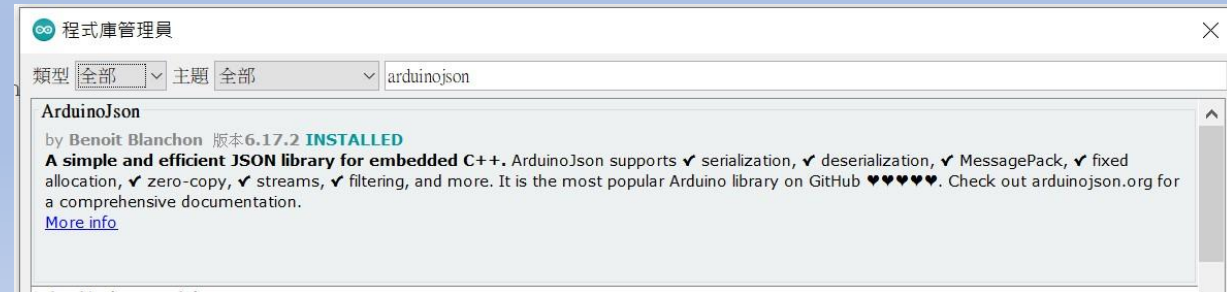
```
void loop()
{
  if (connect_MQTT(MQTT_Server, MQTT_Port, MQTT_Access_token, MQTT_Access_token))
  {
    Rset_Count++;
    if (Get_ideaschain_data(MQTTattrestopic, MQTTattreqtopic))
    {
      Serial.println("=====");
      Serial.println("溫溼度模組 DHT11 檢測中....");
      int err = SimpleDHTErrSuccess;
      if ((err = dht11.read2(&temperature, &humidity, NULL)) != SimpleDHTErrSuccess)
      {
        Serial.print("Read DHT11 failed, err=");
        Serial.println(err);
        Close_MQTT();
        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);
      MQTTmessage="{\" + Sensor1 + "\": " + DHTtemp + ", " + "\" + Sensor2 + "\": " + DHThumi + "}"
```


將資料上傳後，判斷溫度是否高於或低於設定溫度，進而顯示燈號代表可能的運作設備 !!

```
if (Publish_MQTT(MQTTtopic,MQTTmessage))
{
  if (TempHigh.toInt() < (temperature - CorrectionTEMP)) digitalWrite(PB8, HIGH);
  else digitalWrite(PB8, LOW);
  if (TempLow.toInt() > (temperature - CorrectionTEMP)) digitalWrite(PB7, HIGH);
  else digitalWrite(PB7, LOW);
  Rset_Count=0;
}
}
Serial.println("暫停 " + DelayTime +" 秒後再傳送 !!");
Close_MQTT();
delay (DelayTime.toInt()*1000);
if (Rset_Count>20) {delay (10000);nvic_sys_reset();}
}
}
```

程式編譯時，請先安裝所需的程式庫：

SimpleDHT & ArduinoJson



電路圖

