



WiFi 模組

BMC81M001

Arduino Library V1.0.5 說明

版本：V1.40 日期：2025-08-11

www.bestmodulescorp.com

目录

簡介	3
Arduino Lib 函式	3
Arduino Lib 下載及安裝	9
Arduino 範例	10
範例 1: TCP	10
範例 2: Alinyun_Iot	12
範例 3: ThingSpeak	18
範例 4: ThingSpeakPublish	23
範例 5: ThingSpeakSubscribe	29
範例 6: OneNet	32
範例 7: http_get	37

簡介

BMC81M001 是倍創推出的 WiFi 模組，使用 UART 通訊方式。本文檔對 BMC81M001 的 Arduino Lib 函式、Arduino Lib 安裝方式進行說明；範例演示了 TCP 和平台以及 ThingSpeak 的資料傳輸。

Arduino Lib 函式

Arduino Lib 名稱：BMC81M001		Lib 版本：V1.0.5
構造函式 & 初始化		
1	BMC81M001(HardwareSerial*theSeriall = &Serial)	
	描述	構造函式，選擇硬體 UART 通訊方式
	參數	* theSerial：硬體 UART 通訊介面選擇
	回傳值	—
2	BMC81M001(uint16_t rxPin,uint16_t txPin)	
	描述	構造函式，選擇軟體 UART 通訊方式
	參數	rxPin：軟體 UART 的 RX 腳位 txPin：軟體 UART 的 TX 腳位
	回傳值	—
3	void begin(uint32_t baud = BMC81M001_baudRate)	
	描述	模組初始化
	參數	baud：鮑率，預設 115200
	回傳值	void
功能函式 (TCP)		
4	bool connectToAP(String ssid,String pass)	
	描述	模組連接 WiFi
	參數	ssid：WiFi 名稱 pass：WiFi 密碼
	回傳值	執行情況： true：成功 false：失敗
5	bool connectTCP (String ip,int port)	
	描述	接入 TCP 伺服器
	參數	ip：TCP 服務端的 IP 地址 port：埠號
	回傳值	執行情況： true：成功 false：失敗
		備註
		IP 位址和埠號由伺服器定義

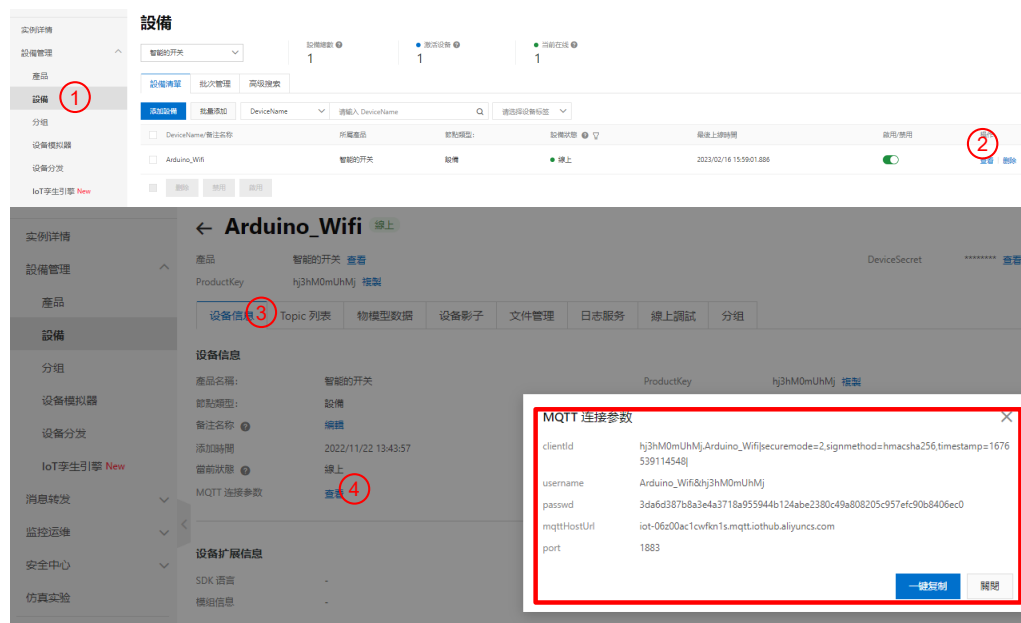
6	bool writeDataTcp(int Dlength, char Dbuffer[])	
	描述	向 TCP 伺服器發送資料
	參數	Dlength：資料長度 Dbuffer[]：資料
	回傳值	執行情況： true：成功 false：失敗
	備註	—
7	String readDataTcp ()	
	描述	連接 TCP 伺服器，讀取伺服器發送的資料
	參數	—
	回傳值	TCP 伺服器發送的資料
	備註	—
功能函式 (Iot)		
8	bool configMqtt(String clientid,String username,String password,String mqtt_host,int server_port);	
	描述	設定 MQTT 連接參數
	參數	clientid：客戶端、用戶 ID。 username：用戶名稱。 password：密碼。 mqtt_host：伺服器位址。 server_port：伺服器埠。
	回傳值	執行情況： true：成功 false：失敗
	備註	—
9	bool setSubscribetopic(String subscribetopic);	
	描述	設定訂閱的預設 Topic
	參數	IoT 平台上 Topic 列表中的預設 subscribe Topic
	回傳值	執行情況： true：成功 false：失敗
	備註	—
10	bool writeString (String Dbuffer,String topic)	
	描述	採用 String 資料格式向 IoT 平台發送資料
	參數	Dbuffer：要發送的字元型資料 topic：消息發佈者和訂閱者之間的傳輸中介
	回傳值	執行情況： true：成功 false：失敗
	備註	—

11	bool writeBytes (char Dbuffer[],int Dlength,String topic)	
	描述	採用位元組資料格式向 IoT 平台發送資料
	參數	Dbuffer[]：資料 Dlength：資料長度 topic：消息發佈者和訂閱者之間的傳輸中介
	回傳值	執行情況： true：成功 false：失敗
	備註	—
12	void readIotData (String *RecvBuff,int *RecvBufflen,String *topic)	
	描述	連接 IoT 平台後，讀取平台發送的資料
	參數	*RecvBuff：接收到的資料 *RecvBufflen：接收資料長度 *topic：接收到的資料所屬 Topic
	回傳值	void
	備註	—
功能函式 (HTTP)		
13	int http_begin(String serverURL,int port,String subURL="")	
	描述	初始化網頁資料 get 功能
	參數	serverURL：功能變數名稱，eg："iot.arduino.org.tw" port：埠號，eg：8888 subURL：路徑，預設 "/"， eg："/bigdata/dhtdata/dhDatatadd. php?MAC=AABBCCDDEEFF&T=34&H=34"
	回傳值	執行情況： 0(HTTP_GET_BEGIN_SUCCESS)：參數格式 OK -1(HTTP_GET_URL_ERROR)：參數格式錯誤
	備註	—
14	int http_get(void)	
	描述	開始進行 HTTP get 操作
	參數	void
	回傳值	執行情況： 0(HTTP_GET_OP_SUCCESS)：get 成功 -2(HTTP_GET_OP_TIMEOUT)：超時未獲取到資料 可能是網址錯誤，埠錯誤，路徑錯誤，伺服器沒有回應等原因造成
	備註	—
15	String http_getString(void)	
	描述	執行 http get 獲取到的網頁資料
	參數	void
	回傳值	網頁回傳的資料
	備註	—

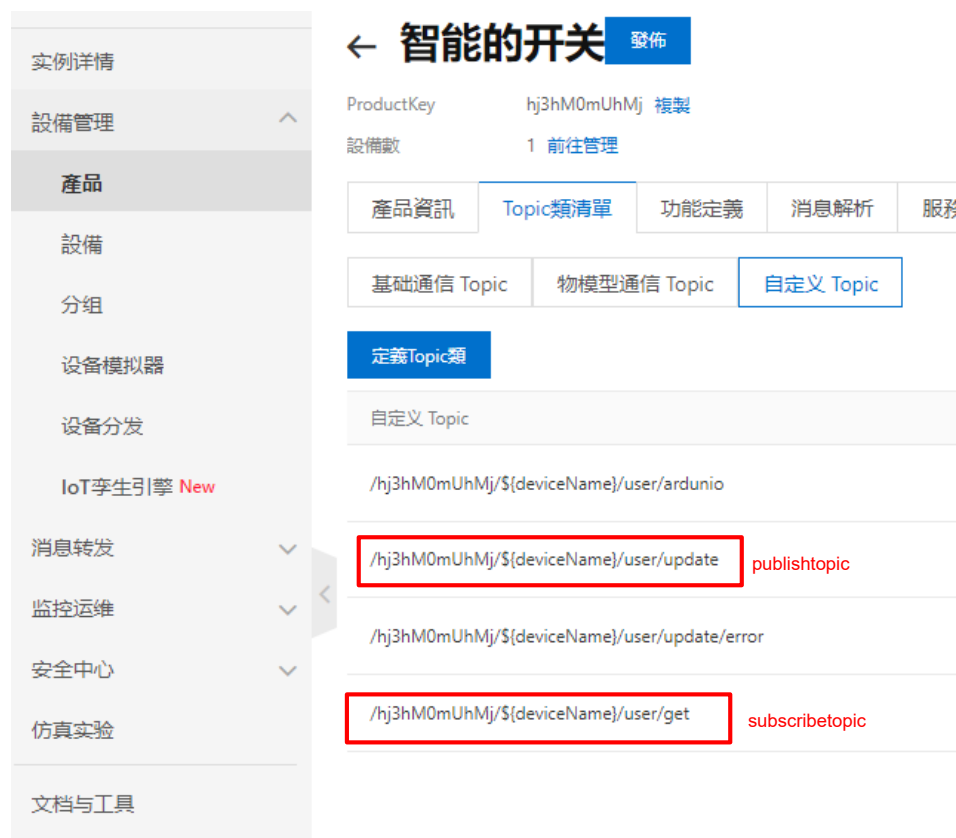
16	void http_end(void)	
	描述	斷開 http 網頁
	參數	void
	回傳值	—
	備註	要執行下一次 http get 時必須先調用一次 end
其他函式		
17	bool reset(void)	
	描述	軟體重置模組
	參數	void
	回傳值	執行情況： true：成功 false：失敗
	備註	—
18	int sendATCommand(String StringstrCmd,int timeout,uint8_t reTry)	
	描述	發送 AT 指令
	參數	StringstrCmd：AT 指令 timeout：超時時間，單位 ms reTry：超時次數
	回傳值	執行情況： true：成功 false：失敗
	備註	—
19	String sendATCmd(String StringstrCmd,int timeout,uint8_t reTry)	
	描述	發送 AT 指令
	參數	StringstrCmd：AT 指令 timeout：超時時間，單位 ms reTry：超時次數
	回傳值	AT 回傳的資料
	備註	—
20	String SSID(void)	
	描述	獲取周圍 WiFi 的資訊
	參數	void
	回傳值	周圍可檢測到的 WiFi 資訊
	備註	最大支援 1000 位元組的存儲空間
21	int getStatus(void)	
	描述	獲取當前 WiFi 狀態
	參數	void
	回傳值	WiFi 狀態： 2(WIFI_STATUS_GOT_IP)：已連接 AP 3(WIFI_STATUS_CONNETED)：已連接 AP，並連接 TCP 4(WIFI_STATUS_DISCONNECTED)：已連接 AP，TCP 連接斷開 5(WIFI_STATUS_NO_CONNET)：未連接 AP
	備註	—

22	String getSSID (void)	
	描述	獲取已連接的 WiFi 名稱
	參數	void
	回傳值	當前連接的 WiFi 名稱
	備註	如 "BestModules"
23	String getIP (void)	
	描述	獲取模組 IP 位址
	參數	void
	回傳值	IP 位址
	備註	—
24	String getGateway (void)	
	描述	獲取模組網關資訊
	參數	void
	回傳值	網關
	備註	必須連接 AP 後才有資訊，否則都為 0。如 "172.20.10.1"
25	String getMask (void)	
	描述	獲取模組子網路遮罩資訊
	參數	void
	回傳值	網關
	備註	必須連接 AP 後才有資訊，否則都為 0。如 "255.255.255.0"
26	String getMacAddress (void)	
	描述	獲取模組的 MAC 位址
	參數	void
	回傳值	MAC 位址
	備註	—
27	String getATVersion (void)	
	描述	獲取模組的 AT 版本
	參數	void
	回傳值	AT 版本
	備註	如 "2.3.0.0"

註 1：登錄物聯網平台，進入控制台 → 選擇公共實例 → 設備管理 → 設備 → 點擊查看進入對應的產品 → 點擊 MQTT 連接參數查看即可。



註 2：登錄物聯網平台，進入控制台 → 選擇公共實例 → 設備管理 → 產品 → 點擊查看進入對應的產品 → Topic 類清單 → 自定義 Topic

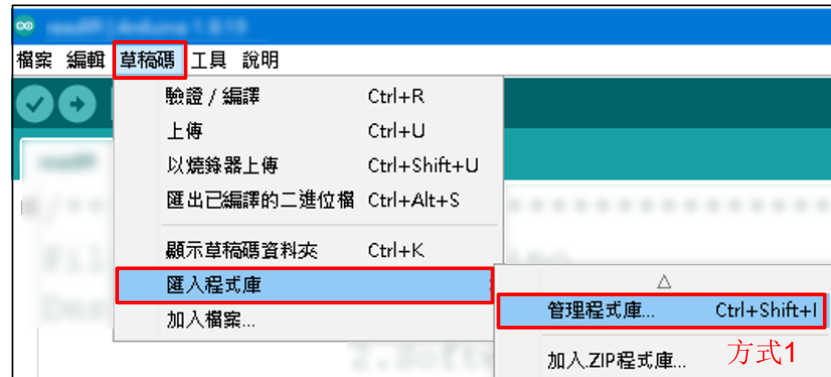


Arduino Lib 下載及安裝

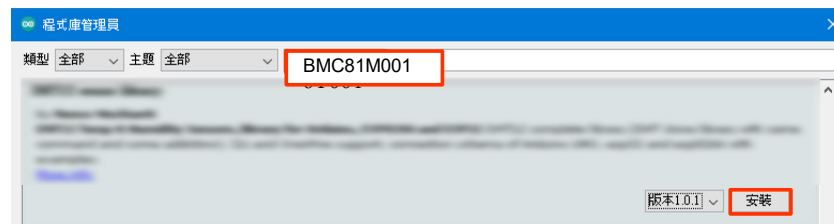
BMC81M001 Library：可參考下面兩種方法安裝 BMC81M001 的 Arduino Library

方式 1：搜索安裝

搜索安裝：Arduino IDE → 草稿碼 → 匯入程式庫 → 管理程式庫 → 搜索 BMC81M001 → 安裝



搜索安裝流程 1

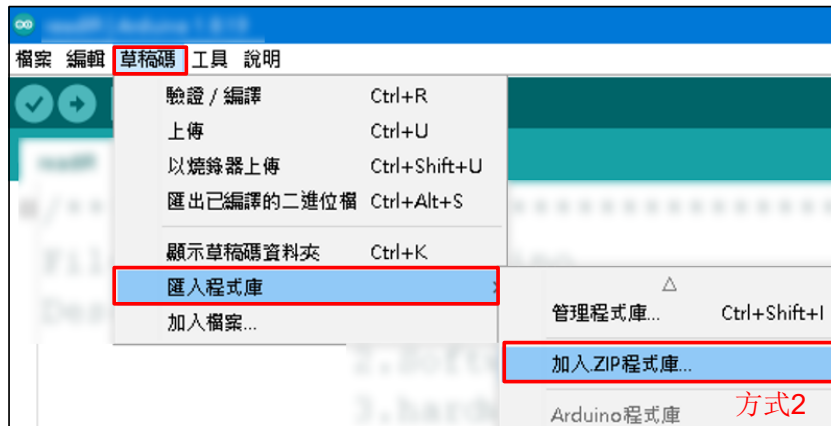


搜索安裝流程 2

方式 2：添加 .ZIP 程式庫，需提前下載 .ZIP 程式庫

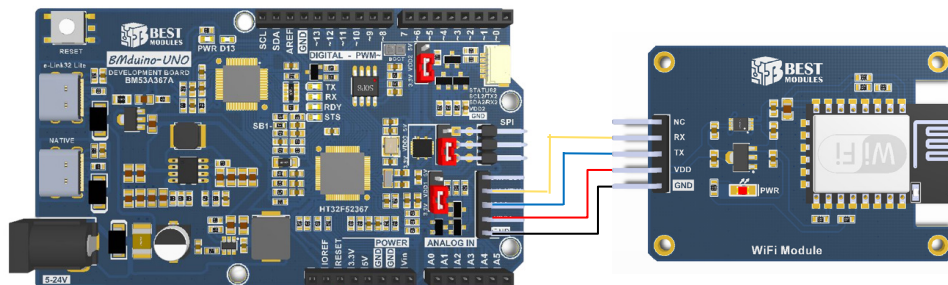
下載方法：打開倍創官方網站 (<https://www.bestmodulescorp.com/bmc81m001.html>) 檔案目錄下的 Arduino 範例程式 (BMC81M001 Library)。

添加 .ZIP 程式庫：Arduino IDE → 草稿碼 → 匯入程式庫 → 加入 .ZIP 程式庫 ...



Arduino 範例

範例 1: TCP



實物連接示意圖

範例 1 實現功能：模組連接手機熱點，實現與手機上的 TCP 服務端 APP 雙向資料通訊。

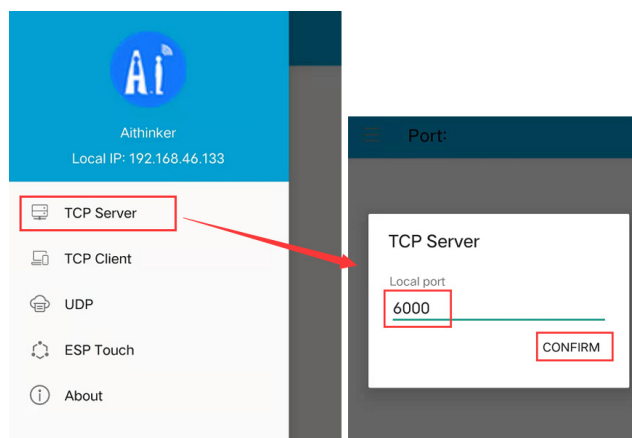
1. 範例打開方式：

Arduino IDE → 檔案 → 範例 → Lib 選擇 (BMC81M001) → 選擇範例 (TCP)

2. 範例說明：

首先需要在 TCP.h 檔中修改 WiFi 資訊和 TCP 伺服器資訊如下：

```
// 連接的 WiFi 的帳戶密碼
#define WIFI_SSID "iQOO_Neo_855" // 手機熱點名稱
#define WIFI_PASS "12345678.!" // 手機熱點密碼
// 連接的 TCP IP 及埠資訊
#define IP "192.168.46.133" // APP 的 TCP 伺服器 IP 地址
#define IP_Port 6000 // APP 的 TCP 伺服器埠
```



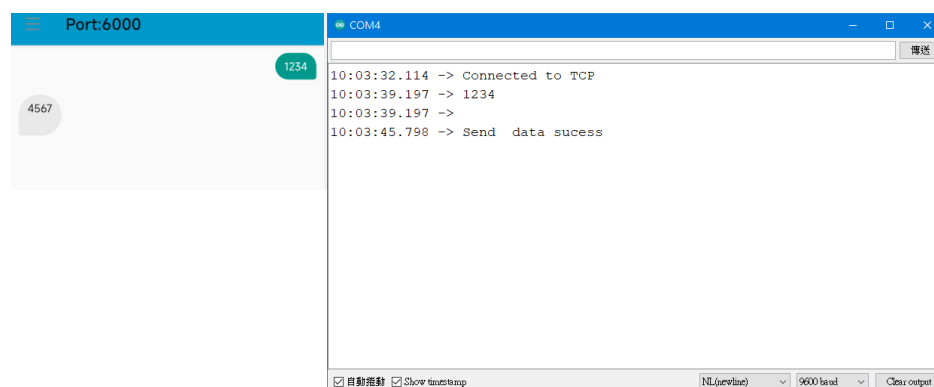
3. 連接 WiFi · 接入 TCP 伺服器

```
#include "TCP.h"
BMC81M001 Wifi(&Serial1);
void setup()
{digitalWrite(LED, LOW);
  Serial.begin(9600);          // 設定序列埠監視視窗
  Wifi.begin();                // 初始化模組
  if(!Wifi.connectToAP(WIFI_SSID,WIFI_PASS)) // 設定連接的熱點名稱和密碼
  {
    Serial.println("Disconnect to WIFI");
  }
  if(!Wifi.connectTCP(IP,IP_Port)) // 連接到 TCP 伺服器 (APP)
  {
    Serial.println("Disconnect to TCP server");
  }
  else
  {Serial.println("Connected to TCP"); }
}
```

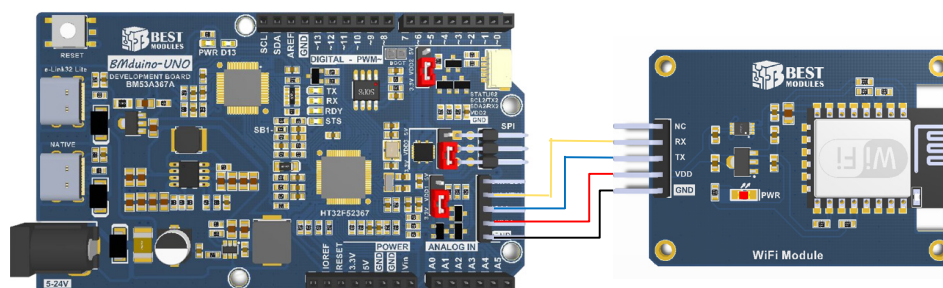
4. 在此範例中，在序列埠監視視窗發送資料，將直接上傳到 TCP 伺服器，在 TCP/UDP 網路調試助手上可以看到模組發送的資訊；在 TCP/UDP 網路調試助手上發送的資料可以在序列埠監視視窗查看。

```
void loop() {
tcpBuff=Wifi.readDataTcp();    // 監聽模組接收的資料
if(tcpBuff!=0)
{
  Serial.println(tcpBuff);
}
while (Serial.available() > 0) // 接收到監視序列埠發送的資料，透傳發送資料
{
  SerialBuff[resLen++] = mSerial.read();
  delay(10);
}
if(resLen>0)
{
  digitalWrite(LED, HIGH);    // 有資料亮燈
  if(Wifi. writeDataTcp (resLen,SerialBuff)) // 向 TCP 伺服器發送資料
  {
    Serial.println("Send data sucess");
    digitalWrite(LED,LOW);    // 資料發送成功滅燈
  }
  clearBuff();
}
void clearBuff(){
  memset(SerialBuff,'\0',RES_MAX_LENGTH); // 清空接收序列埠資料的字元型陣列
  resLen = 0;
}
```

程式執行結果在 APP 和序列埠監視視窗結果顯示：



範例 2: Alinyun_Iot



實物連接示意圖

範例 2 實現功能：模組連接手機熱點，連接平台上的實例，實現資料透傳功能。

1. 範例打開方式：

Arduino IDE → 檔案 → 範例 → Lib 選擇 (BMC81M001) → 選擇範例 (Alinyun_Iot)

2. 登錄物聯網平台 (aliyun.com) 註冊後選擇公共實例，設定產品及設備，自定義 Topic，將 MQTT 參數以及各類 Topic 填入 Alinyun_Iot.h 檔，具體流程如下：

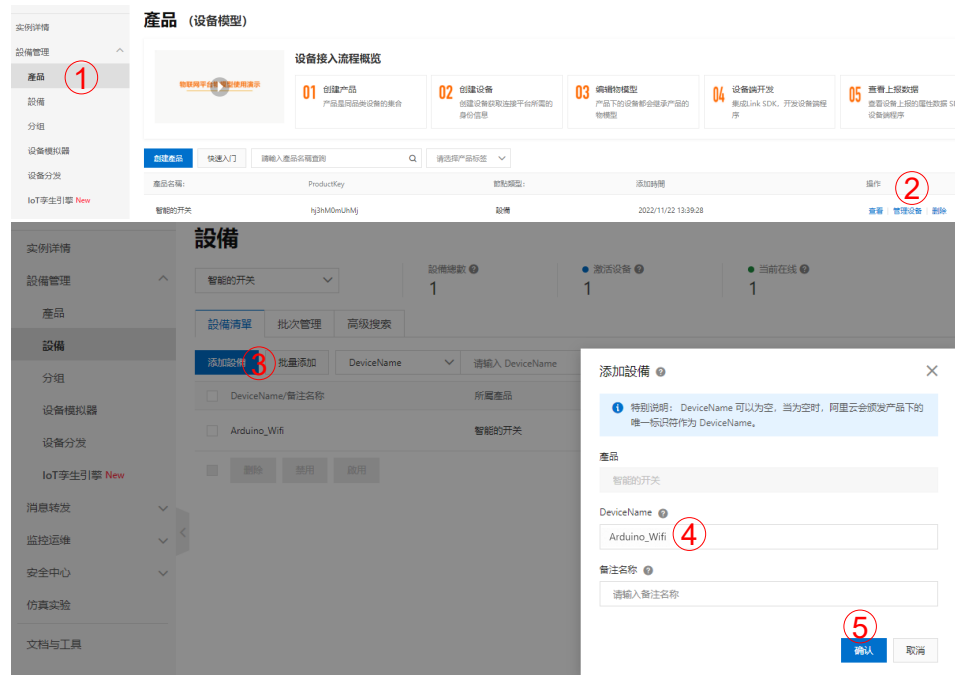
1) 設定產品：進入公共實例後，左側菜單設備管理 → 產品 → 設定產品



2) 自定義 Topic，定義資料傳輸中介：設備管理 → 產品 → 查看剛剛新建的產品 → Topic 類清單 → 自定義 Topic

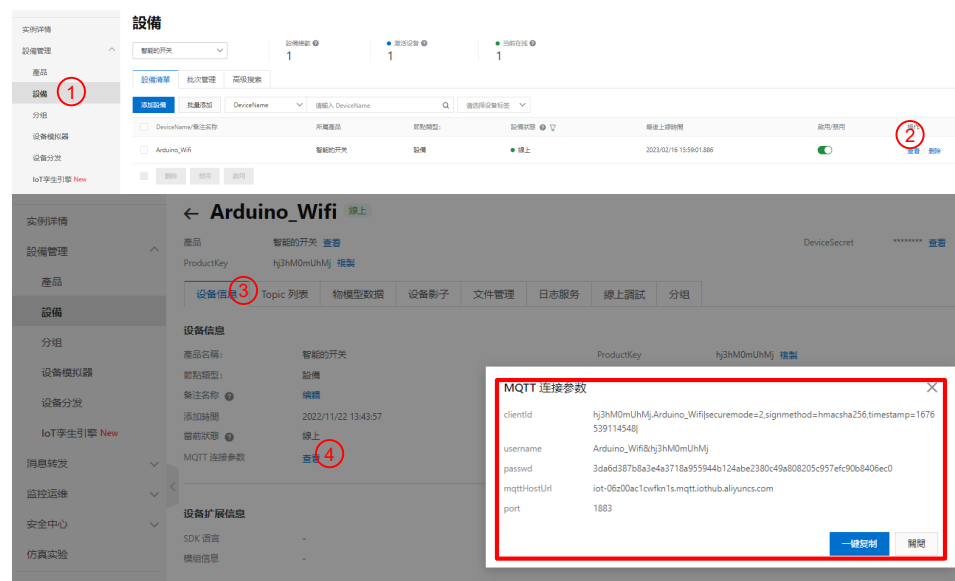


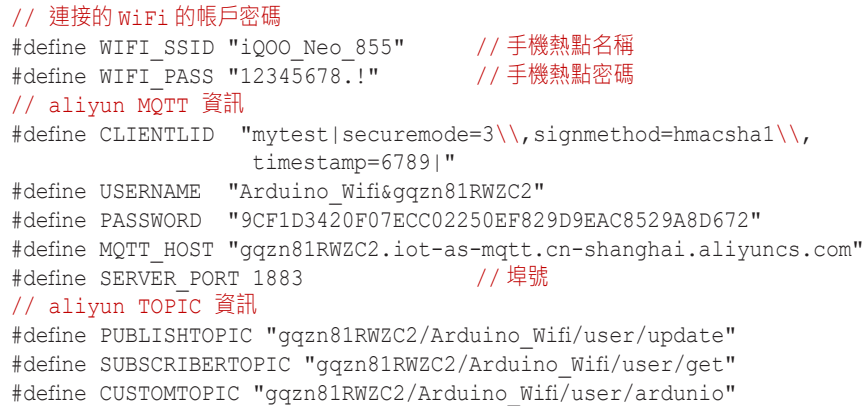
3) 回到“產品”介面 → “管理設備” → “添加設備” → DeviceName，點擊確認。



4) 查看設備的 MQTT 連接參數：設備 → 查看剛剛設定的設備 Arduino_Wifi → 設備資訊 → MQTT 參數查看；查看設備的 Topic 同步步驟 2 中的自定義 Topic 步驟相同。

將 CLIENTLID、USERNAME、PASSWORD、MQTT_HOST、SERVER_PORT 等 MQTT 參數填入 Aliyun_Iot.h 文件裡，將 PUBLISHTOPIC、SUBSCRIBERTOPIC、CUSTOMTOPIC 等類 Topic 也填入 Aliyun_Iot.h 文件裡。





15 2025-08-11

3. 範例說明：

根據平台產生的資料，對模組進行初始化。其中包括軟體重置模組，接入 WiFi，接入 IoT，如下：

```
#include "Aliyun_Iot.h"
BMC81M001      Wifi(&Serial1);
void setup()
{
  digitalWrite(LED, LOW);
  Serial.begin(9600);           // 設定序列埠監視視窗
  Wifi.begin();                 // 模組初始化及設定
  Wifi.reset();
  Serial.print("WIFI Connection Results:");
  if(Wifi.connectToAP(WIFI_SSID,WIFI_PASS)==0) // 根據名稱和密碼連接 WiFi
  {
    Serial.println("fail");
  }
  else {Serial.println("success");}
  Serial.print("Aliyun Connection Results:");
  Wifi.sendATCommand("AT+CIPSENTPCFG=1,8,\"ntp1.aliyun.com\"",1000,2);
  // 連接阿里雲
  if(Wifi.configMqtt(CLIENTLID,USERNAME,PASSWORD,MQTT_HOST,
    SERVER_PORT)==0) // 設定 MQTT 參數
  {
    Serial.println("fail");
  }
  else
  {
    Serial.println("success");
    Wifi.setSubscribetopic(SUBSCRIBERTOPIC);
  }
  Serial.print("Topic set Results:");
  delay(200);}
}
```

4. 運行後，即可在平台上查看到設備狀態。



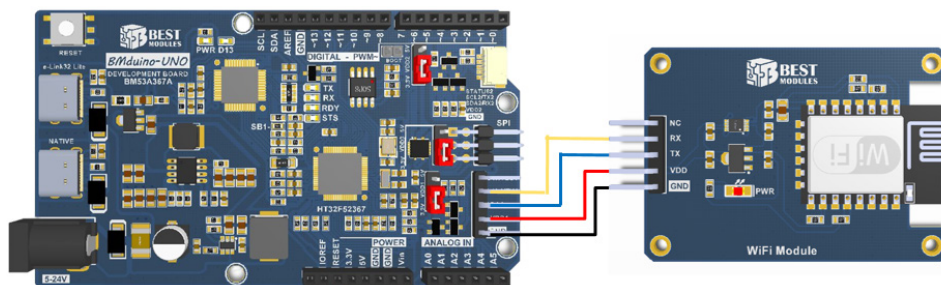
5. 在此範例中，當序列埠監視視窗發送資料，模組只做透傳功能，將接收到序列埠監視視窗發送的資料直接上傳到雲平台，可以在平台日誌記錄查看到模組上傳的資訊。

```
void loop()
{
    Wifi.readIotData(&aliyunReciveBuff,&aliyunReciveBufflen,&recTopic); // 監聽
                                     // 模組接收的資料

    while (Serial.available() > 0) // 接收監視序列埠發送的資料
    {
        SerialBuff[resLen++] = Serial.read();
        delay(10);
    }
    if(resLen>0)
    {
        DATA_BUF = (String )SerialBuff;
        topic = PUBLISHTOPIC;
        if(Wifi.writeString(DATA_BUF,topic))
        {
            Serial.println("Send String data sucess");
        }
        if(Wifi.writeBytes(SerialBuff,resLen,topic))
        {
            Serial.println("Send byte data sucess");
        }
        clearBuff();
    }
}

void clearBuff(){
    memset(SerialBuff,'\0',RES_MAX_LENGTH); // 清空資料接收陣列
    resLen = 0;
}
```

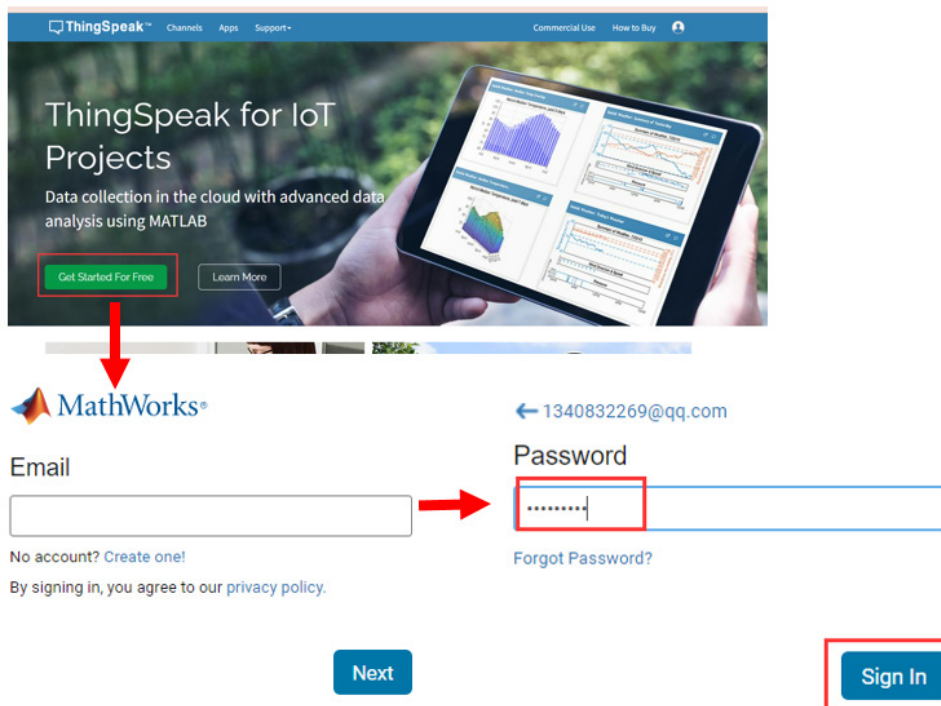
範例 3: ThingSpeak



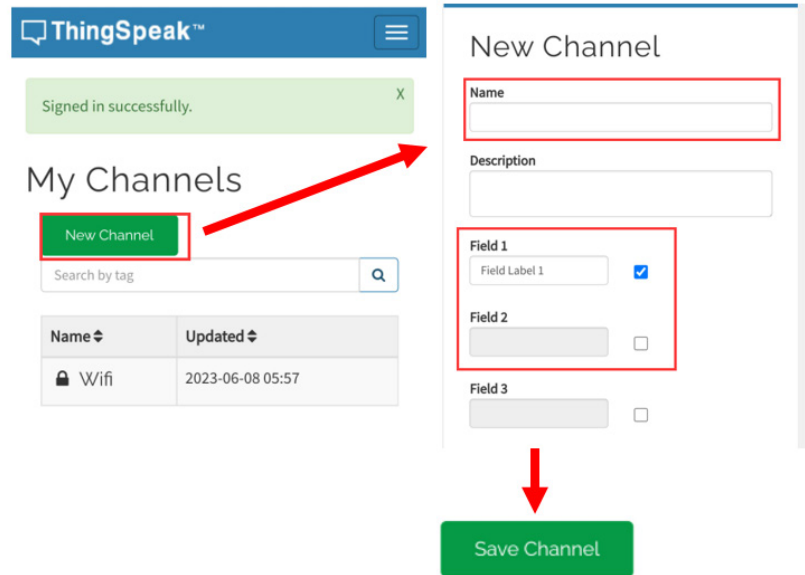
實物連接示意圖

範例 3 實現功能：模組連接手機熱點，連接 ThingSpeak 平台上的通道，發送資料到平台上顯示。

1. 範例打開方式：Arduino IDE → 文件 → 範例 → Lib 選擇 (BMC81M001) → 選擇範例 (ThingSpeak)
2. 登錄 ThingSpeak 平台 (<https://thingspeak.com/>)，註冊賬號後，新建資料通道，再新建 MQTT 設備，將 MQTT 參數填入 ThingSpeak.h 文件中。
 - 1) 註冊賬號並進入平台：點擊 Get Start For Free，進入平台介面，註冊賬號並登入。



- 2) 設定 channels：點擊 New Channel，填入 Name 和開啟資料字段數量，保存通道即可。



ThingSpeak™

Signed in successfully. X

My Channels

New Channel

Search by tag

Name	Updated
Wifi	2023-06-08 05:57

New Channel

Name

Description

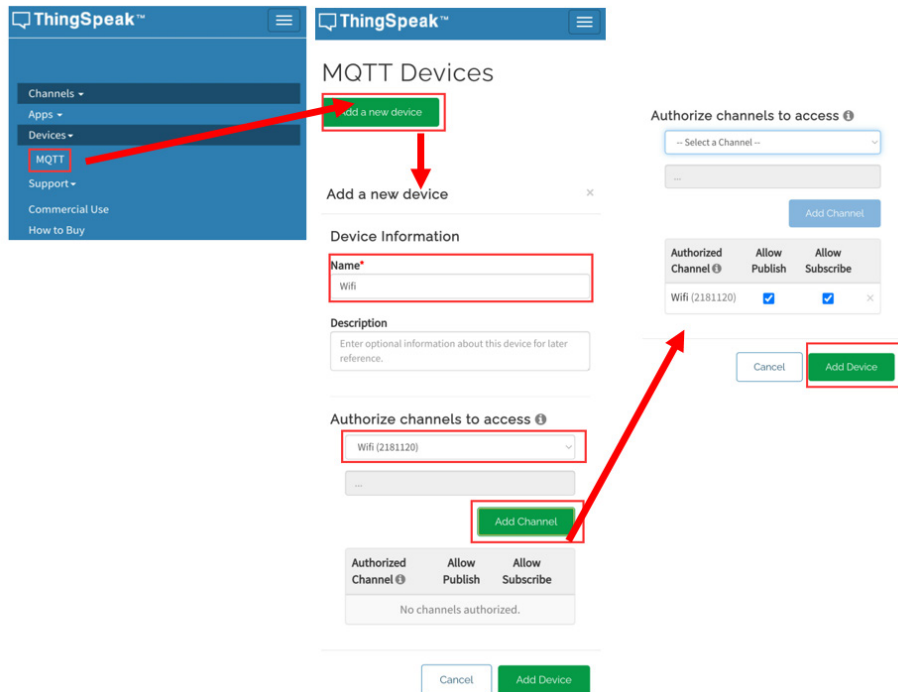
Field 1 ☒

Field 2 ☐

Field 3 ☐

Save Channel

- 3) 新建 MQTT 設備：填入設備名稱 Wifi，選擇新建的通道 Wifi，點擊 Add channel，顯示 Allow Publish 和 Subublish OK 後 Add Device。



ThingSpeak™

MQTT Devices

Add a new device

Add a new device

Device Information

Name* Wifi

Description

Authorize channels to access

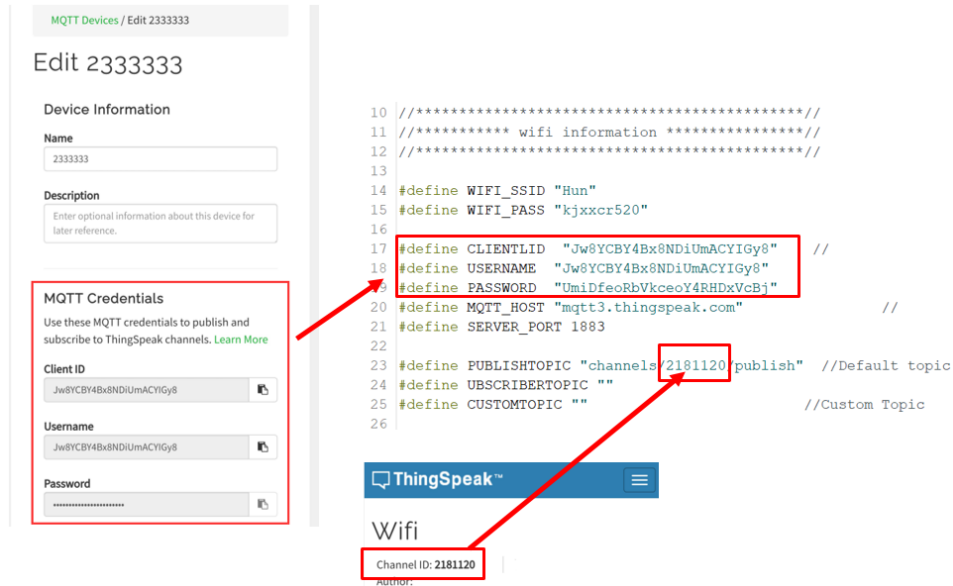
Wifi (2181120)

Add Channel

Authorized Channel	Allow Publish	Allow Subscribe
Wifi (2181120)	☒	☒

Cancel Add Device

- 4) 點擊 edit，查看新建 MQTT 設備的 MQTT 參數，將資料填入 ThingSpeak.h 文件中。



MQTT Devices / Edit 2333333

Edit 2333333

Device Information

Name
2333333

Description
Enter optional information about this device for later reference.

MQTT Credentials
Use these MQTT credentials to publish and subscribe to ThingSpeak channels. [Learn More](#)

Client ID
Jw8YCBY4Bx8NDiUmACYIGy8

Username
Jw8YCBY4Bx8NDiUmACYIGy8

Password

```

10 //*****
11 //***** wifi information *****
12 //*****
13
14 #define WIFI_SSID "Hun"
15 #define WIFI_PASS "kxxcr520"
16
17 #define CLIENTID "Jw8YCBY4Bx8NDiUmACYIGy8" //
18 #define USERNAME "Jw8YCBY4Bx8NDiUmACYIGy8"
19 #define PASSWORD "UmiDfeoRbVkceoY4RHDxVcBj"
20 #define MQTT_HOST "mqtt3.thingspeak.com" //
21 #define SERVER_PORT 1883
22
23 #define PUBLISH TOPIC "channels/2181120/publish" //Default topic
24 #define UNSUBSCRIBE TOPIC ""
25 #define CUSTOM TOPIC "" //Custom Topic
26

```

ThingSpeak™

Wifi

Channel ID: 2181120

```

// 要連接的熱點資訊
#define WIFI_SSID "iQOO_Neo_855" // 手機熱點名稱
#define WIFI_PASS "12345678.!" // 手機熱點密碼
// ThingSpeak MQTT 資訊
#define CLIENTID "Jw8YCBY4Bx8NDiUmACYIGy8"
#define USERNAME "Jw8YCBY4Bx8NDiUmACYIGy8"
#define PASSWORD "UmiDfeoRbVkceoY4RHDxVcBj"
#define MQTT_HOST "mqtt3.thingspeak.com" // 伺服器位址
#define SERVER_PORT 1883 // 埠號
// ThingSpeak TOPIC 資訊
#define PUBLISH TOPIC "channels/2181120/publish"

```

3. 範例說明：

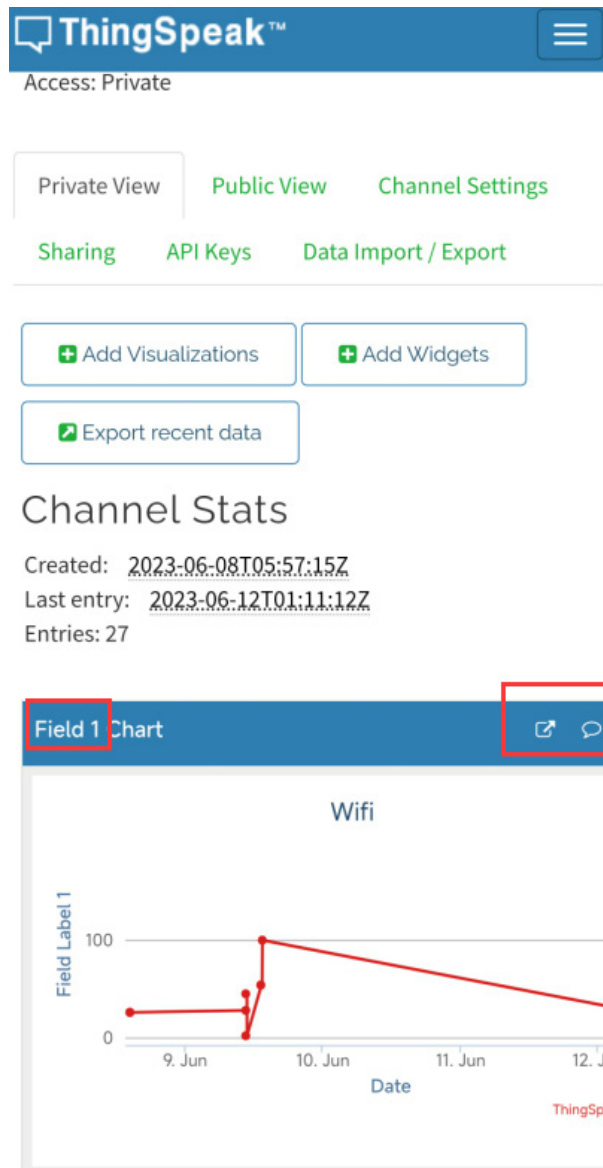
模組初始化後，連接熱點，設定 MQTT 參數，連接 ThingSpeak 平台的通道，發送資料至平台。如下：

```
#include "ThingSpeak.h"
BMC81M001      Wifi(&Serial1);
void setup()
{
  Serial.begin(9600);           // 設定序列埠監視視窗
  Wifi.begin();                 // 模組初始化及設定
  Wifi.reset();
  Serial.print("WIFI Connection Results:");
  if(Wifi.connectToAP(WIFI_SSID,WIFI_PASS)==0) // 根據名稱和密碼連接 WiFi
  {
    Serial.println("fail");
  }
  else {Serial.println("success");}
  Serial.print("ThingSpeak Connection Results:");
  if(Wifi.configMqtt(CLIENTLID,USERNAME,PASSWORD,MQTT_HOST,
    SERVER_PORT)==0)           // 設定 MQTT 參數
  {
    Serial.println("fail");
  }
  else
  {
    Serial.println("success");
  }
  delay(200);}
}
```

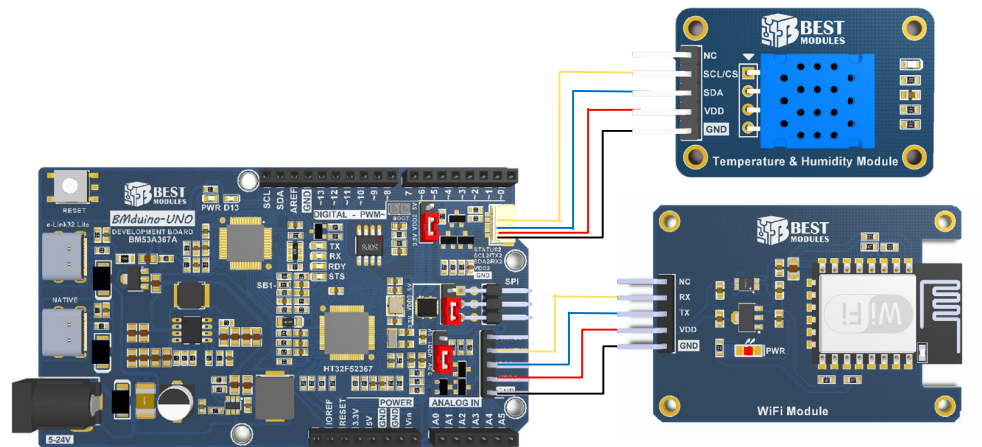
4. 在此範例中，將資料 100 上傳到雲平台，可以在通道介面看到資料。

```
void loop()
{
  Wifi.readIotData(&ReciveBuff,&ReciveBufflen,&recTopic); // 監聽模組接收
                                                         // 的資料
  if(ReciveBufflen)
  {
    Serial.println(ReciveBufflen);
    Serial.println(ReciveBuff);
  }
  DATA_BUF = "field1="; // 發送通道字段 1 資料
  DATA_BUF += "100";
  topic = PUBLISHTOPIC;
  if(Wifi.writeString(DATA_BUF,topic))
  {
    Serial.println("Send String data sucess");
  }
}
```

5. ThingSpeak 顯示結果，可以從圖表中看到資料的接收。



範例 4: ThingSpeakPublish



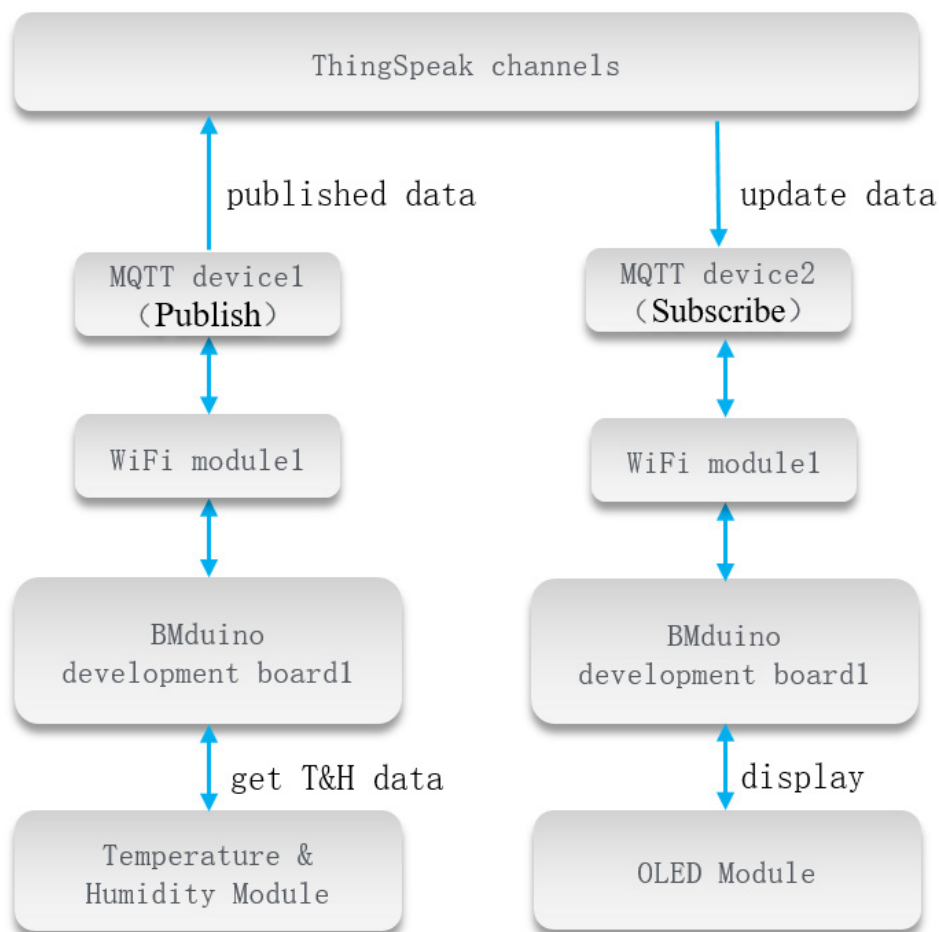
實物連接示意圖

範例 4 和範例 5 可搭配使用，實現兩個 WiFi 設備的遠端資料交換，一個作為 Publish 端，另一個作為 Subscribe 端。實現功能如下：

Publish 端：將溫濕度模組 BME33M251 的溫度、濕度資料上傳到 ThingSpeak；

Subscribe 端：從 ThingSpeak 讀取 Publish 端剛上傳的溫度、濕度資料，並顯示於 OLED 模組上；

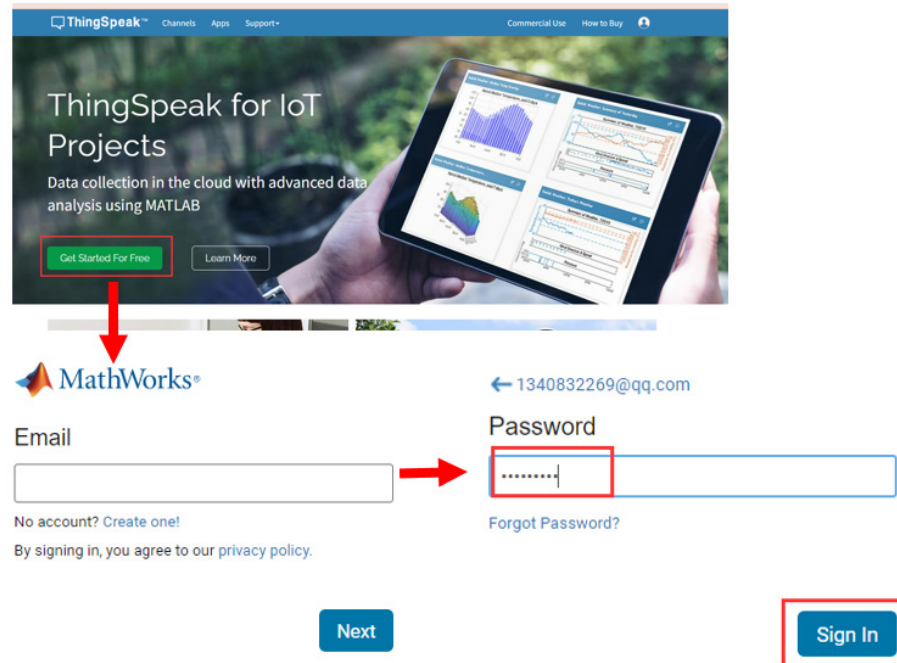
注意：使用範例 4 和範例 5 需先安裝 Arduino Library：BMD31M090，BM25S2021-1。



範例 4 實現功能：讀取溫濕度模組 BME33M251 的溫度、濕度資料，通過 MQTT 設備 1(Publish) 發佈到 ThingSpeak 平台的通道中。

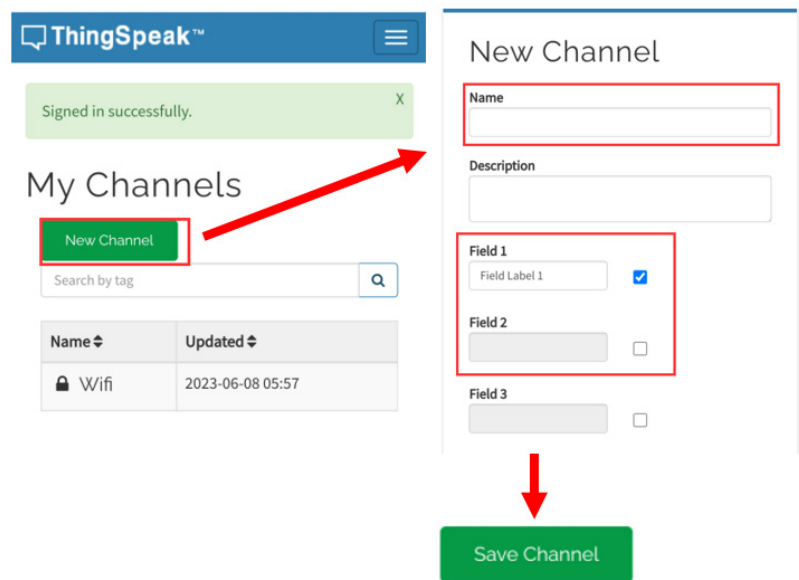
1. 範例打開方式：Arduino IDE → 文件 → 範例 → Lib 選擇 (BMC81M001) → 選擇範例 (ThingSpeakPublish)。
2. 登錄 ThingSpeak 平台 (<https://thingspeak.com>)，註冊帳號後，新建資料通道(wifi t&h)，新建 MQTT 設備 (Publish)，將 MQTT 參數填入 ThingSpeakPublish.h 文件中。

- 1) 註冊帳號並進入平台：點擊 Get Start For Free，進入平台介面，註冊帳號並登入。



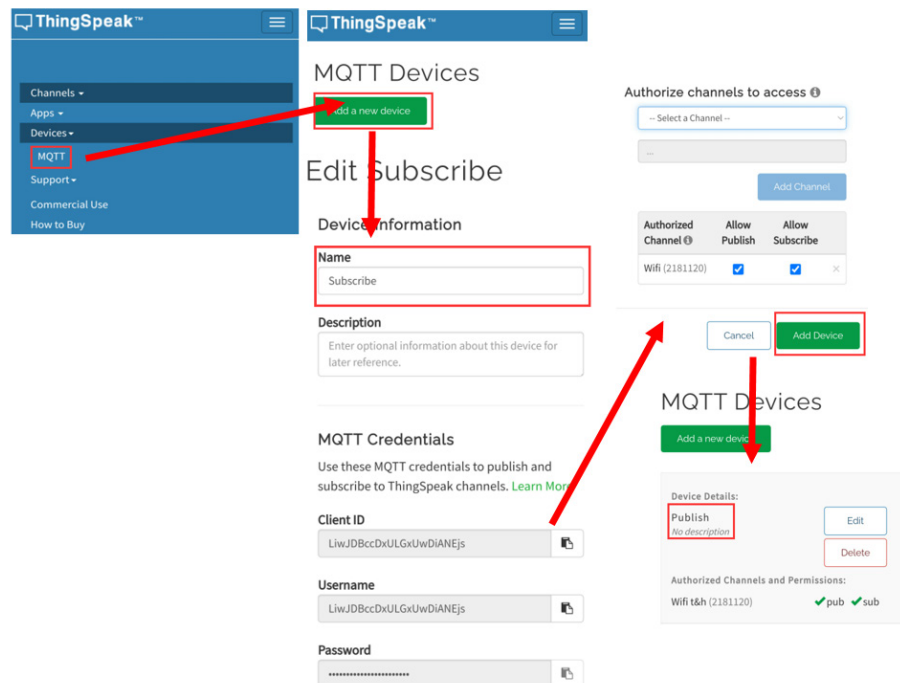
The screenshot shows the ThingSpeak website. The 'Get Started For Free' button is highlighted with a red box. Below it, the registration form is shown with the 'Email' and 'Password' fields. The 'Password' field is highlighted with a red box. The 'Next' and 'Sign In' buttons are also visible.

- 2) 建立 channels：點擊 New Channel，填入自定義 Name (例 wifi t&h) 和 Field1 (例 humidity) 和 Field2 (例 temperature)，保存通道即可。



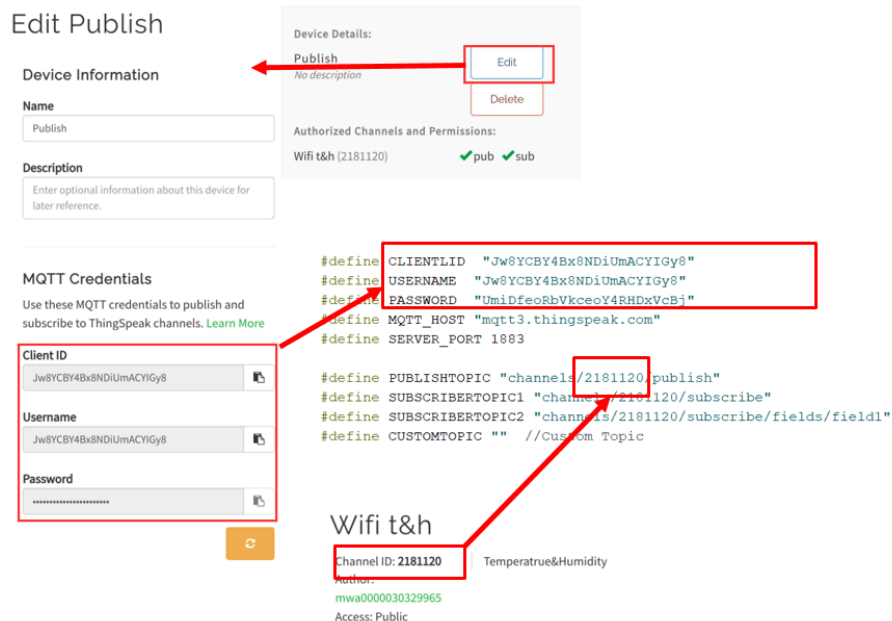
The screenshot shows the 'New Channel' form in the ThingSpeak interface. The 'New Channel' button is highlighted with a red box. The form fields include 'Name', 'Description', 'Field 1', 'Field 2', and 'Field 3'. The 'Field 1' checkbox is checked. The 'Save Channel' button is highlighted with a red box.

- 3) 新建 MQTT 設備 1：填入設備名稱 (Publish)，授權通道 wifi t&h 的發佈和訂閱，點擊 Add channel，顯示 Allow Publish 和 Sublish OK 後 Add Device。此 Publish 設備用作上傳溫濕度模組資料。



The screenshot shows the ThingSpeak interface. On the left, the 'MQTT' option is selected under the 'Devices' menu. The main area displays the 'MQTT Devices' page with an 'Add a new device' button. A red arrow points to this button, which leads to the 'Edit Subscribe' form. In this form, the 'Name' field is set to 'Subscribe'. A red arrow points to the 'Add Device' button. Another red arrow points to the 'MQTT Devices' list, which now shows a device named 'Publish' with the description 'No description'. The 'Authorized Channels and Permissions' section shows 'Wifi t&h (2181120)' with 'pub' and 'sub' permissions checked.

- 4) 點擊設備 (Publish) 右側的“edit”按鈕，查看 MQTT 設備 (Publish) 的 MQTT 參數，將資料填入 ThingSpeakPublish.h 文件中。



The screenshot shows the 'Edit Publish' form. The 'Name' field is set to 'Publish'. The 'Description' field is empty. The 'MQTT Credentials' section shows the 'Client ID', 'Username', and 'Password' fields. A red arrow points from the 'Edit' button to the 'MQTT Credentials' section. Another red arrow points from the 'Client ID' field to the code file. The code file shows the following definitions:

```
#define CLIENTID "Jw8YCBY4Bx8NDiUmACYIGy8"
#define USERNAME "Jw8YCBY4Bx8NDiUmACYIGy8"
#define PASSWORD "UmiDfeoRbVkcceY4RHDxVcbi"
#define MQTT_HOST "mqtt3.thingspeak.com"
#define SERVER_PORT 1883

#define PUBLISH_TOPIC "channels/2181120/publish"
#define SUBSCRIBERTOPIC1 "channels/2181120/subscribe"
#define SUBSCRIBERTOPIC2 "channels/2181120/subscribe/fields/field1"
#define CUSTOMTOPIC "" //Custom Topic
```

A red arrow points from the 'Wifi t&h' channel details to the code file. The channel details show the 'Channel ID: 2181120' and the 'Access: Public'.

3. 範例說明：

模組初始化後，連接熱點，設定 MQTT 參數，連接 ThingSpeak 平台的通道，如下：

```
#include "ThingSpeak.h"
#include <BMD31M090.h>
#include <BM25S2021-1.h>    // 溫濕度模組
BMC81M001  Wifi(&Serial1);
BM25S2021_1 BMht(&Wire2);
void setup()
{
  Serial.begin(9600);
  BMht.begin();
  Wifi.begin();              //WIFI 模組、溫濕度模組初始化
  Wifi.reset();
  Serial.print("WIFI Connection Results:");
  if(Wifi.connectToAP(WIFI_SSID,WIFI_PASS)==0)    // 連接熱點
  {Serial.println("fail");}
  else {Serial.println("success");}
  Serial.print("ThingSpeak Connection Results:");
  if(Wifi.configMqtt(CLIENTLID,USERNAME,PASSWORD,MQTT_HOST,SERVER_PORT)==0)
  // 根據 MQTT 連接 ThingSpeak 平台
  {Serial.println("fail");}
  else {Serial.println("success");}
  delay(200);
  Wifi.setSubscribetopic(SUBSCRIBERTOPIC2);
  Wifi.setSubscribetopic(SUBSCRIBERTOPIC1);
  topic = PUBLISHTOPIC;    // 發佈 Topic
}
```

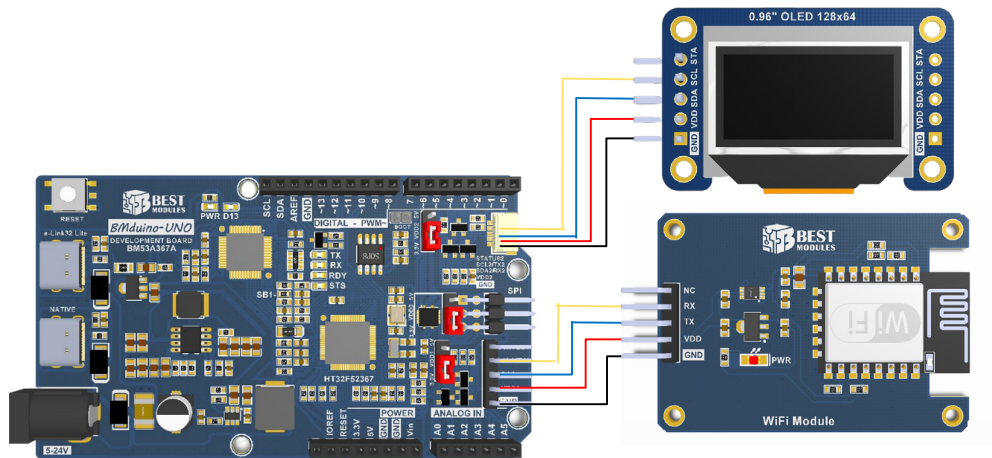
4. 輪詢溫濕度模組的資料，將濕度發送至通道中的 field1，將溫度發送至通道的 field2。

```
void loop()
{
  data1=BMht.readHumidity();    // 獲取濕度資料
  Humidity=String(data1,2);     // 浮點型資料轉成字串
  DATA_BUF = "field1=";       // 通道 1 資料
  DATA_BUF += Humidity;
  if(Wifi.writeString(DATA_BUF,topic))    // 發佈資料至通道 1
  {
    Serial.println("Send String data sucess");
    delay(1000);
  }
  clearBuff();
  data1=BMht.readTemperature(false);    // 獲取溫度資料
  Temperature=String(data1,2);           // 浮點型資料轉成字串
  DATA_BUF = "field2=";                 // 通道 2 資料
  DATA_BUF += Temperature;
  if(Wifi.writeString(DATA_BUF,topic))    // 發佈資料至通道 2
  {
    Serial.println( "Send String data sucess" );
    delay(1000);
  }
  clearBuff();
  delay(2000);
}
```

5. ThingSpeak 顯示結果，可以從圖表中看到資料的接收。



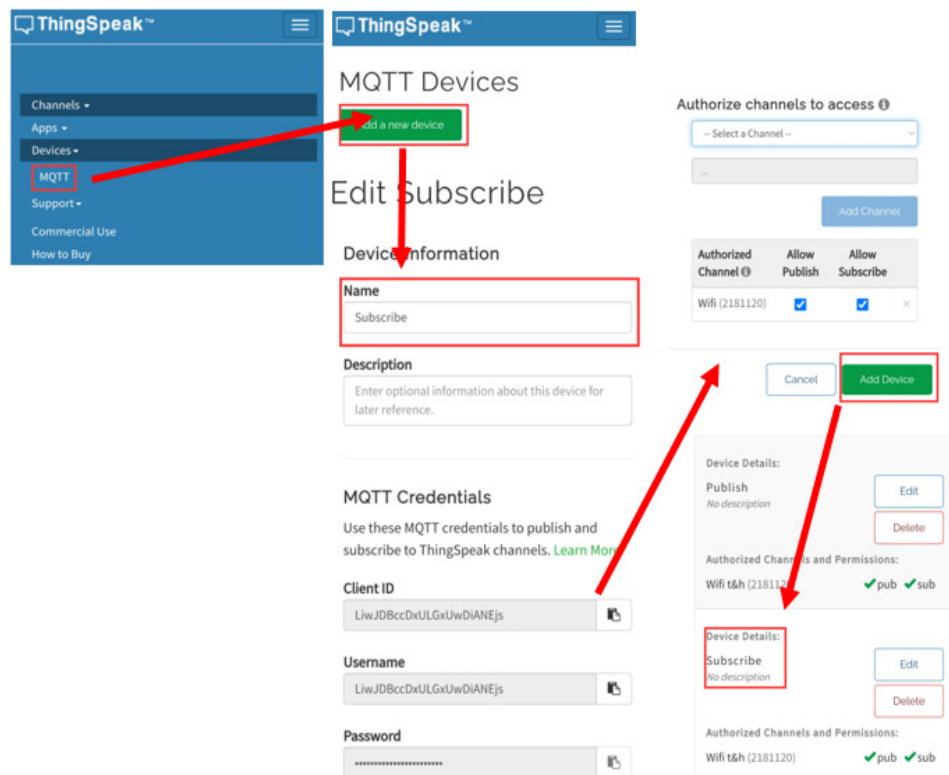
範例 5: ThingSpeakSubscribe



實物連接示意圖

範例 5 實現功能：訂閱資料更新 Topic，通過新建並連接 MQTT 設備 2 (Subscribe) 接收 ThingSpeak 對應通道最新發佈的溫濕度資料，並顯示在 OLED 模組上。

1. 範例打開方式：Arduino IDE → 文件 → 範例 → Lib 選擇 (BMC81M001) → 選擇範例 (ThingSpeakSubscribe)。
2. ThingSpeak 通道無需新建，使用範例 4 的 wifi t&h。下面新建 MQTT 設備 (Subscribe) 以及填寫參數至 ThingSpeakSubscribe.h 文件。
 - 1) 新建 MQTT 設備 2：填入設備名稱 (Subscribe)，授權通道 wifi t&h 的發佈和訂閱，點擊 Add channel，顯示 Allow Publish 和 Sublsh OK 後 Add Device。此 Subscribe 設備用於接收通道更新的資料。



ThingSpeak™

Channels ▾
Apps ▾
Devices ▾
MQTT
Support ▾
Commercial Use
How to Buy

MQTT Devices

[Add a new device](#)

Edit Subscribe

Device Information

Name:

Description:

MQTT Credentials

Use these MQTT credentials to publish and subscribe to ThingSpeak channels. [Learn More](#)

Client ID:

Username:

Password:

Authorize channels to access

-- Select a Channel --

[Add Channel](#)

Authorized Channel ID	Allow Publish	Allow Subscribe
Wifi (2181120)	☒	☒

[Cancel](#) [Add Device](#)

Device Details:

Publish: No description [Edit](#) [Delete](#)

Authorized Channels and Permissions:

Channel	pub	sub
Wifi t&h (2181120)	✓	✓

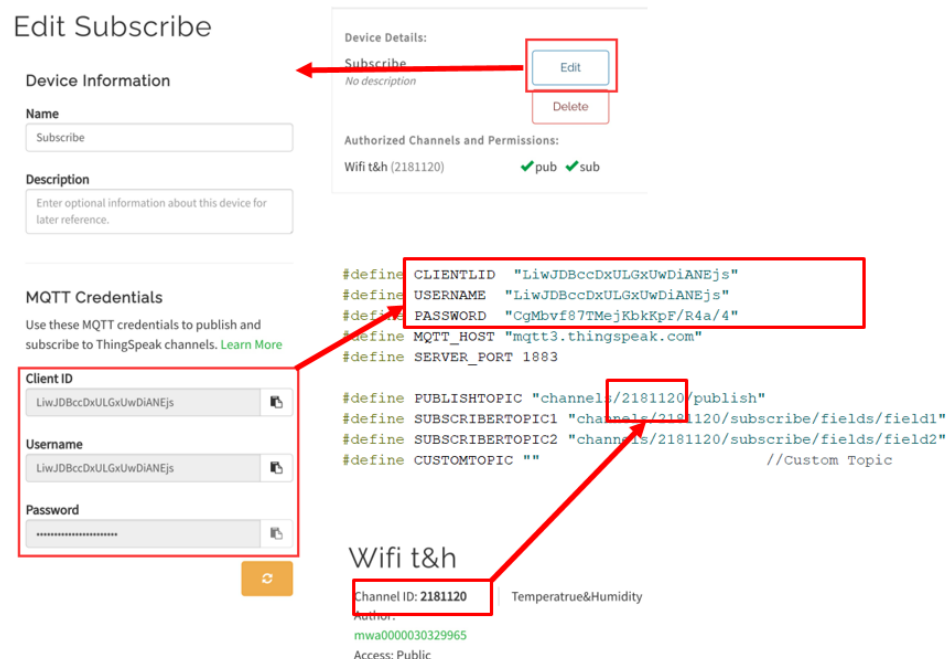
Device Details:

Subscribe: No description [Edit](#) [Delete](#)

Authorized Channels and Permissions:

Channel	pub	sub
Wifi t&h (2181120)	✓	✓

2) 點擊設備 (Subscribe) 右側的 “edit” 按鈕，查看 MQTT 設備 (Subscribe) 的 MQTT 參數，將資料填入 ThingSpeakSubscribe.h 文件中。



Edit Subscribe

Device Information

Name:

Description:

MQTT Credentials

Use these MQTT credentials to publish and subscribe to ThingSpeak channels. [Learn More](#)

Client ID:

Username:

Password:

[Edit](#) [Delete](#)

Authorized Channels and Permissions:

Channel	pub	sub
Wifi t&h (2181120)	✓	✓

```
#define CLIENTID "LiwJDBccDxULGxUwDIANEjs"
#define USERNAME "LiwJDBccDxULGxUwDIANEjs"
#define PASSWORD "CgMbvF87TMejKbkPf/R4a/4"
#define MQTT_HOST "mqtt3.thingspeak.com"
#define SERVER_PORT 1883

#define PUBLISH_TOPIC "channel/2181120/publish"
#define SUBSCRIBE_TOPIC1 "channel/2181120/subscribe/fields/field1"
#define SUBSCRIBE_TOPIC2 "channel/2181120/subscribe/fields/field2"
#define CUSTOM_TOPIC "" //Custom Topic
```

Wifi t&h

Channel ID: Temperature&Humidity

Author: mwa000030329965

Access: Public

3. 範例說明：

模組初始化後，連接熱點，設定 MQTT 參數，連接 ThingSpeak 平台的通道。
如下：

```
#include "ThingSpeakRead.h"
#include <BMD31M090.h>
#include <BM25S2021-1.h> //OLED 模組
BMC81M001 Wifi(&Serial1);
BMD31M090 BMD31(128,64, &Wire2);
float data1;
String Humidity;
String Temperature;
void Display();
void setup()
{
    BMD31.begin(0x3C);
    BMD31.setFont(FontTable_8X16);
    delay(100);
    Serial.begin(9600);
    Wifi.begin();
    Wifi.reset();
    Display();
    Serial.print("WIFI Connection Results:");
    if(Wifi.connectToAP(WIFI_SSID,WIFI_PASS)==0) // 連接熱點
    {
        Serial.println("fail");
    }
    else {Serial.println("success");}
    Serial.print("ThingSpeak Connection Results:");
    if(Wifi.configMqtt(CLIENTLID,USERNAME,PASSWORD,MQTT_HOST,SERVER_PORT)==0)
    // 根據 MQTT 連接 ThingSpeak 平台
    {
        Serial.println("fail");
    }
    else {Serial.println("success");}
    delay(200);
    Wifi.setSubscribetopic(SUBSCRIBERTOPIC2);
    Wifi.setSubscribetopic(SUBSCRIBERTOPIC1);}
```

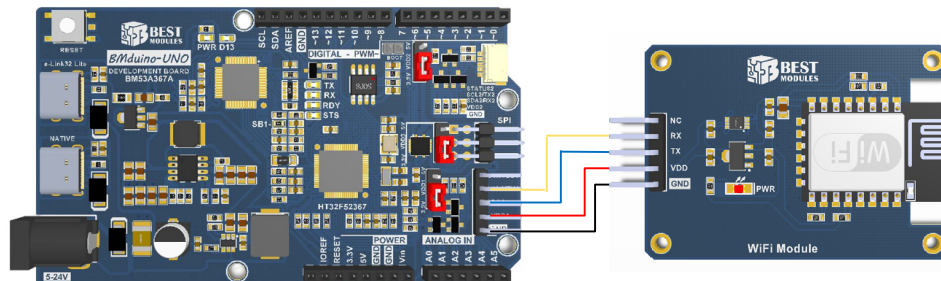
4. 輪詢平台是否有更新資料，有資料則識別對應資料，顯示在 OLED 模組上。

```
void loop()
{
  Wifi.readIotData(&RecvBuff, &RecvBufflen, &topic); // 接收平台下發的資料
  if (RecvBufflen)
  {
    if (topic == SUBSCRIBE TOPIC1) // 通道 1 資料為濕度
    {
      RecvBuff.toCharArray(OledBuff, 7);
      BMD31.drawString(40, 2, (u8*)OledBuff); // OLED 顯示
      BMD31.drawString(85, 2, (u8*)" %");
      RecvBufflen = 0;
    }
    if (topic == SUBSCRIBE TOPIC2) // 通道 2 資料為溫度
    {
      RecvBuff.toCharArray(OledBuff, 7);
      BMD31.drawString(40, 6, (u8*)OledBuff); // OLED 顯示
      BMD31.drawString(85, 6, (u8*)" C");
      RecvBufflen = 0;
    }
  }
  clearBuff(); // 清空資料
}
```

5. OLED 顯示結果如下：



範例 6: OneNet



實物連接示意圖

範例 6 實現功能：模組連接 OneNet 物聯網平台，使用 MQTT 連接對應產品的設備，演示模組上報資料至對應的屬性，以及接收平台下發的屬性資料。

- 範例打開方式：Arduino IDE → 檔案 → 範例 → Lib 選擇 (BMC81M001) → 選擇範例 (OneNet)。
- 登錄 OneNet 官網 (<http://ha.cmccconenet.com>) 註冊後登錄，選擇產品服務 → 物聯網開發平台，創建產品及設備，設定物模型屬性，將 MQTT 參數以及各

類 Topic 填入 OneNet.h 檔，具體流程如下：

1) 創建物模型產品及設備：

<https://open.iot.10086.cn/doc/v5/fuse/detail/922>

記錄設備名稱、產品 ID、設備秘鑰

2) 下載工具生成 token 參數，記錄 token 參數

<https://open.iot.10086.cn/doc/v5/fuse/detail/1487> 下載工具

<https://open.iot.10086.cn/doc/v5/fuse/detail/1486> token 參數說明

<http://shijianchuo.wiicha.com/> 時間戳線上生成連結

3) 添加產品物模型的自定義功能點

<https://open.iot.10086.cn/doc/v5/fuse/detail/987>

參考連結中的產品開發—物模型—什麼是物模型 / 物模型管理

功能點 1：功能類型：屬性；識別字：String；資料類型：String；資料長度：256

功能點 2：功能類型：屬性；識別字：humi；資料類型：int32；資料長度：1~100

功能點 3：功能類型：屬性；識別字：temp；資料類型：float；取值範圍：1~100

4) 將步驟一 & 二記錄的產品 ID、設備名稱、token 填入 OneNet.h 檔的對應參數。

產品 ID——USERNAME

設備名稱——CLIENTID

token——PASSWORD

```
1 #ifndef _BMC81M001_H_
2 #define _BMC81M001_H_
3
4 //***** Header *****//
5 //***** Header *****//
6 //***** Header *****//
7 #include "ONENET.h"
8 #include "BMC81M001.h"
9
10 //***** wifi information *****//
11 //***** wifi information *****//
12 //***** wifi information *****//
13
14 #define WIFI_SSID "Hun"
15 #define WIFI_PASS "kxxcr520"
16
17 #define CLIENTID "wifi"
18 #define USERNAME "cekQZv1EU5"
19 #define PASSWORD "version=2018-10-31&res=products%2FcekQZv1EU5%2Fdevices%2Fwifi&et=1748654171&method=md5&sign=3Yucz76x73Hfuhe0%2FrE8hAK30%3D"
20 #define MQTT_HOST "mqtt.mecclouds.com"
21 #define SERVER_PORT 1883
22 #define ONENET_TOPIC_OTA "$sys/cekQZv1EU5/wifi/ota/inform"
23 #define ONENET_TOPIC2 "$sys/cekQZv1EU5/wifi/customize/up_reply"
24 #define ONENET_TOPIC_PROP_POST "$sys/cekQZv1EU5/wifi/thing/property/post"
25 //设备属性上报请求,设备---->OneNET
26 #define ONENET_TOPIC_PROP_SET "$sys/cekQZv1EU5/wifi/thing/property/set"
27 //设备属性设置请求,OneNET---->设备
```

```
// 連接的 WiFi 的帳戶密碼
#define WIFI_SSID "iQOO_Neo_855" // 手機熱點名稱
#define WIFI_PASS "12345678.!" // 手機熱點密碼
// MQTT 資訊
#define CLIENTID "XXXXXX" // XXXXXX 為設備名稱
#define USERNAME "YYYYYY" // YYYYYY 為產品 ID
#define PASSWORD "token"
#define MQTT_HOST "mqtt.heclouds.com"
#define SERVER_PORT 1883 // 埠號
// TOPIC 資訊
#define ONENET_TOPIC_PROP_POST "$sys/cekQZv1EU5/wifi/thing/property/post"
#define ONENET_TOPIC_PROP_SET "$sys/cekQZv1EU5/wifi/thing/property/set"
#define ONENET_TOPIC_PROP_POST_REPLY "$sys/cekQZv1EU5/wifi/thing/property/post/reply"
#define ONENET_TOPIC_PROP_SET_REPLY "$sys/cekQZv1EU5/wifi/thing/property/set/reply"
```

3. 範例說明：模組進行初始化。其中包括：軟體重置模組，連接手機熱點，連接 OneNet 平台的設備，訂閱對應的 Topic 如下：

```
#include "ONENET.h"
BMC81M001 Wifi(&Serial1); // 通訊介面選擇 Serial1
String Data_String="Hello"; // 屬性 1
int Data_int=30; // 屬性 2
float Temperature=27.5; // 屬性 3
char pub[200]; // 上報資料存儲陣列
void setup()
{
    Serial.begin(9600);
    Wifi.begin();
    Wifi.reset();// 模組初始化及重置
    Serial.print("WIFI Connection Results:");
    if(Wifi.connectToAP(WIFI_SSID,WIFI_PASS)==0) // 連接手機熱點
    {
        Serial.println("fail");
    }
    else {Serial.println("success");}
    Serial.print("ONENET Connection Results:");// 連接 OneNet 平台
    if(Wifi.configMqtt(CLIENTID,USERNAME,PASSWORD,MQTT_HOST,SERVER_PORT)==0)
    {
        Serial.println("fail");
    }
    else {Serial.println("success");}
    delay(200);
    Wifi.setSubscribetopic(ONENET_TOPIC_PROP_SET);// 訂閱屬性設定請求以及上報相應 Topic
    Wifi.setSubscribetopic(ONENET_TOPIC_PROP_GET);
    Wifi.setSubscribetopic(ONENET_TOPIC_PROP_GET_REPLY);
    Wifi.setSubscribetopic(ONENET_TOPIC_PROP_SET_REPLY);
    Wifi.setSubscribetopic(ONENET_TOPIC_PROP_POST_REPLY);
```

4. 創建字串，整理資料格式，轉化為 char 陣列格式通過上報屬性 Topic 上報至平台。平台接收後，下發上報成功結果。資料格式參考連結：

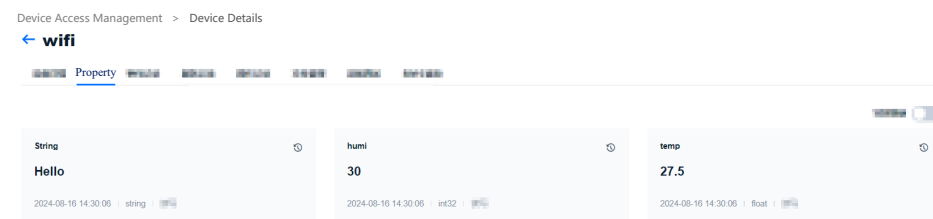
(<https://open.iot.10086.cn/doc/v5/fuse/detail/902>)

```
String temp=String(Temperature,1);
SET_BUF="{\"id\": \"\"";
SET_BUF +=ID;
ID++;
SET_BUF += "\", \"";
SET_BUF += "\"version\": \"1.0\", \"params\": \"";
SET_BUF += \"{ \"temp\": { \"value\": \"";
SET_BUF += temp;
SET_BUF += \"}, \"humid\": { \"value\": \"";
SET_BUF += Data_int;
SET_BUF += \"}, \"String\": { \"value\": \"\""; // 雙引號需要使用 \\ 轉義
SET_BUF += Data_String;
SET_BUF += \"}\"}"; // 資料整理 OK
SET_BUF.toCharArray(pub,SET_BUF.length()+1); // 資料轉 char 陣列
// 上報 3 種屬性資料
if(Wifi.writeBytes(pub,SET_BUF.length(),ONENET_TOPIC_PROP_POST))
{
    Serial.println("Send String data sucess");
}
delay(1000);
void loop()
{
    Wifi.readIotData(&ReciveBuff,&ReciveBufflen,&topic); // 接收平台下發的上報成功
// 結果
    Serial.println(Wifi.OneNetReciveBuff); // 結果列印至序列埠監視視窗
    clearBuff();
    delay(1000);
}
void clearBuff()
{
    memset(pub,'\\0',200);
    resLen = 0;
}
```

5. 模組上報屬性資料到 OneNet 的運行結果序列埠監視視窗顯示結果以及平台資料更新如下：

```
14:30:03.651 -> WIFI Connection Results:sucess
14:30:04.509 -> ONENET Connection Results:sucess
14:30:06.618 -> Send String data sucess
14:30:07.616 ->
14:30:07.647 ->
14:30:07.647 -> +MQTTFUB:OK
14:30:07.647 -> +MQTTSUBRCV:0,"says/cekQ2v1EU5/wifi/thing/property/post/reply",39,{"id":"123","code":200,"msg":"sucess"}
```

平台上顯示結果如下：設備接入管理—設備詳情——屬性



6. 模組接收 OneNet 平台通過設備模擬器發送的屬性資料

參考連結 (<https://open.iot.10086.cn/doc/v5/fuse/detail/922>) 的 “5. 設備屬性設定” 章節內容。

填入 String 資料 "Hello from onenet" ; int32 資料 "30" ; 浮點型資料 "27.5" 後按屬性設定按鈕發送。



The screenshot shows a web interface for configuring a device. It has three input fields with checkboxes and type selectors:

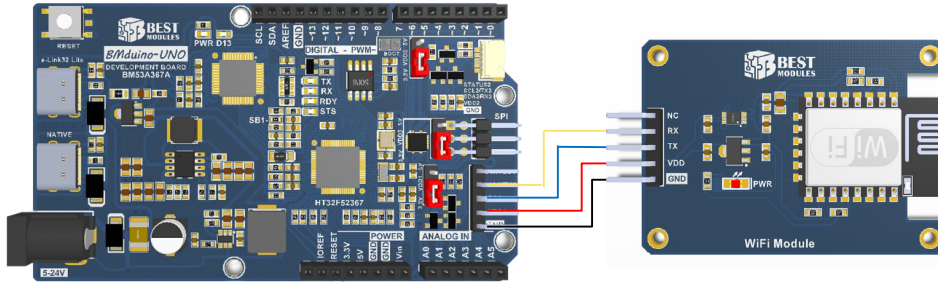
- String (String) string**: The input field contains "Hello from onenet".
- humi (humi) int32**: The input field contains "30".
- temp (temp) float**: The input field contains "27.5".

At the bottom right, there is a blue button labeled "Send" which is highlighted with a red box.

模組接收 OneNet 平台通過設備模擬器發送的屬性資料，序列埠監視視窗列印接收到的資料。

```
14:30:07.783 ->
14:30:15.648 -> +MQTTF02BRCV0,"key/oneNet/10086/wifi/shang/property/ent",88,({"id":1,"version":"1.0","payload":{"string":"Hello From onenet","humi":30,"temp":27.5}})
14:30:15.816 ->
```

範例 7: http_get



實物連接示意圖

範例 7 實現功能：模組連接手機熱點，實現 get 網頁內容。

1. 範例打開方式：

Arduino IDE → 檔案 → 範例 → Lib 選擇 (BMC81M001) → 選擇範例 (http_get)

2. 範例說明：

首先要正確設置埠、功能變數名稱、路徑這三個參數 (請瞭解標準 HTTP 協議組成)

```
#define HTTPGET_PORT_HTTP 80
#define HTTPGET_PORT_HTTPS 443
#define PORT 8888 // 埠
#define URL "http://iot.arduino.org.tw" // 功能變數名稱
#define SUBURL "/bigdata/dhtdata/dhDataatadd.php?MAC=AABBCCDDEEFF&T=34&H=34"
// 路徑
```

3. 連接 WiFi，進行一次 HTTP get

```
void setup()
{
  Serial.begin(115200);
  Wifi.begin();
  Wifi.connectToAP("zengdebin","123456789");// 連接 WiFi 熱點
  Wifi.http_begin(URL,PORT,SUBURL); // 切換參數就需要進行一次 begin
}

void loop() {
  Wifi.http_get();// 執行一次 http get
  Serial.println(Wifi.http_getString());//get http result
  Wifi.http_end();// 完成後就需要執行 end 才能下一次執行 http get
  delay(10000);
}
```

4. 程式執行結果在序列埠監視視窗結果顯示：

```
Output  Serial Monitor x
Message (Enter to send message to 'BM53A367A (BMduino-UNO HT32F52367)' on 'COM1588')

13:03:18.587 ->
13:03:30.378 -> de
13:03:30.378 ->
13:03:30.378 -> ip:27.149.102.242<br>GET DATA passed <br>(4417933E9BEC)<br>insert into big.dhtdata
13:03:30.413 ->
13:03:30.413 ->
```

Copyright© 2025 by BEST MODULES CORP. All Rights Reserved.

本文件出版時倍創已針對所載資訊為合理注意，但不保證資訊準確無誤。文中提到的資訊僅是提供作為參考，且可能被更新取代。倍創不擔保任何明示、默示或法定的，包括但不限於適合商品化、令人滿意的品質、規格、特性、功能與特定用途、不侵害第三人權利等保證責任。倍創就文中提到的資訊及該資訊之應用，不承擔任何法律責任。此外，倍創並不推薦將倍創的產品使用在會因故障或其他原因而可能會對人身安全造成危害的地方。倍創特此聲明，不授權將產品使用於救生、維生或安全關鍵零組件。在救生 / 維生或安全應用中使用倍創產品的風險完全由買方承擔，如因該等使用導致倍創遭受損害、索賠、訴訟或產生費用，買方同意出面進行辯護、賠償並使倍創免受損害。倍創 (及其授權方，如適用) 擁有本文件所提供資訊 (包括但不限於內容、資料、範例、材料、圖形、商標) 的智慧財產權，且該資訊受著作權法和其他智慧財產權法的保護。倍創在此並未明示或暗示授予任何智慧財產權。倍創擁有不事先通知而修改本文件所載資訊的權利。如欲取得最新的資訊，請與我們聯繫。