WiFi Module

BMC81M001
Arduino Library V1.0.3 Description

Revision: V1.20 Date: December 20, 2023

www.bestmodulescorp.com

Contents

Introduction

The Best Modules BMC81M001 is a WiFi module, which uses the UART communication method. This document provides the description of the BMC81M001 Arduino Lib functions and how to install the Arduino Lib. The example demonstrates the data transmission functions of TCP, Alibaba Cloud Platform and ThingSpeak.

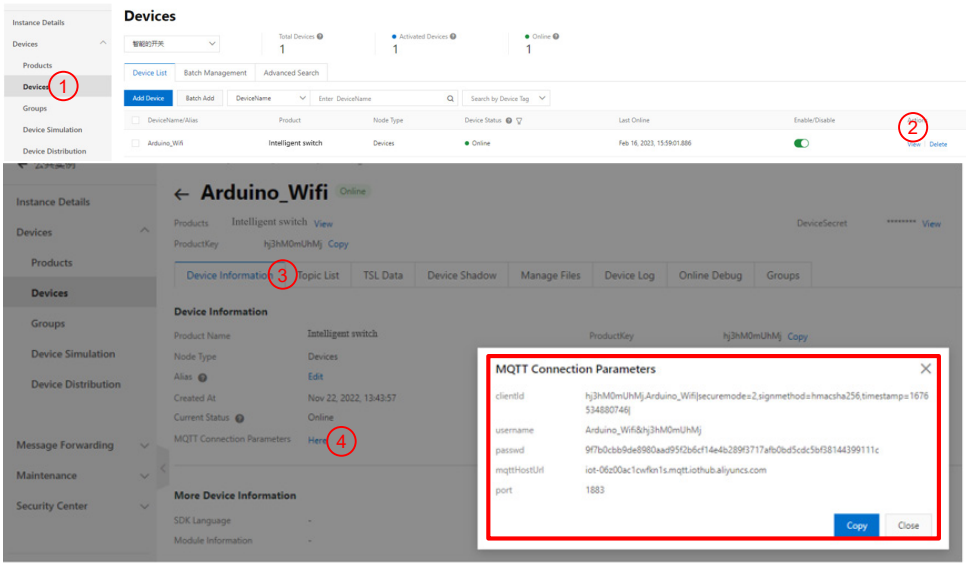
Arduino Lib Functions

Arduino Lib Name: BMC81M001		Lib Version: V1.0.3
Constructors & Initialisation		
1	BMC81M001(HardwareSerial*theSeriall = &Serial)	
	Description	Constructor, select hardware UART communication method
	Parameter	* theSerial: hardware UART communication interface selection
	Return Value	—
	Note	—
2	BMC81M001(uint16_t rxPin,uint16_t txPin)	
	Description	Constructor, select software UART communication method
	Parameter	rxPin: Software UART RX pin txPin: Software UART TX pin
	Return Value	—
	Note	—
3	void begin(uint32_t baud = BMC81M001_baudRate)	
	Description	Module initialisation
	Parameter	baud: baud rate, defaults to 115200
	Return Value	void
	Note	—
Performance Functions (TCP)		
4	bool connectToAP(String ssid,String pass)	
	Description	The module is connected to WiFi
	Parameter	Ssid: WiFi name pass: WiFi password
	Return Value	Execution result: true: Succeeded false: Failed
	Note	—
5	bool connectTCP(String ip,int port)	
	Description	Connect to TCP server
	Parameter	ip: TCP server IP address port: Port number
	Return Value	Execution result: true: Succeeded false: Failed
	Note	The IP address and port number are defined by the server

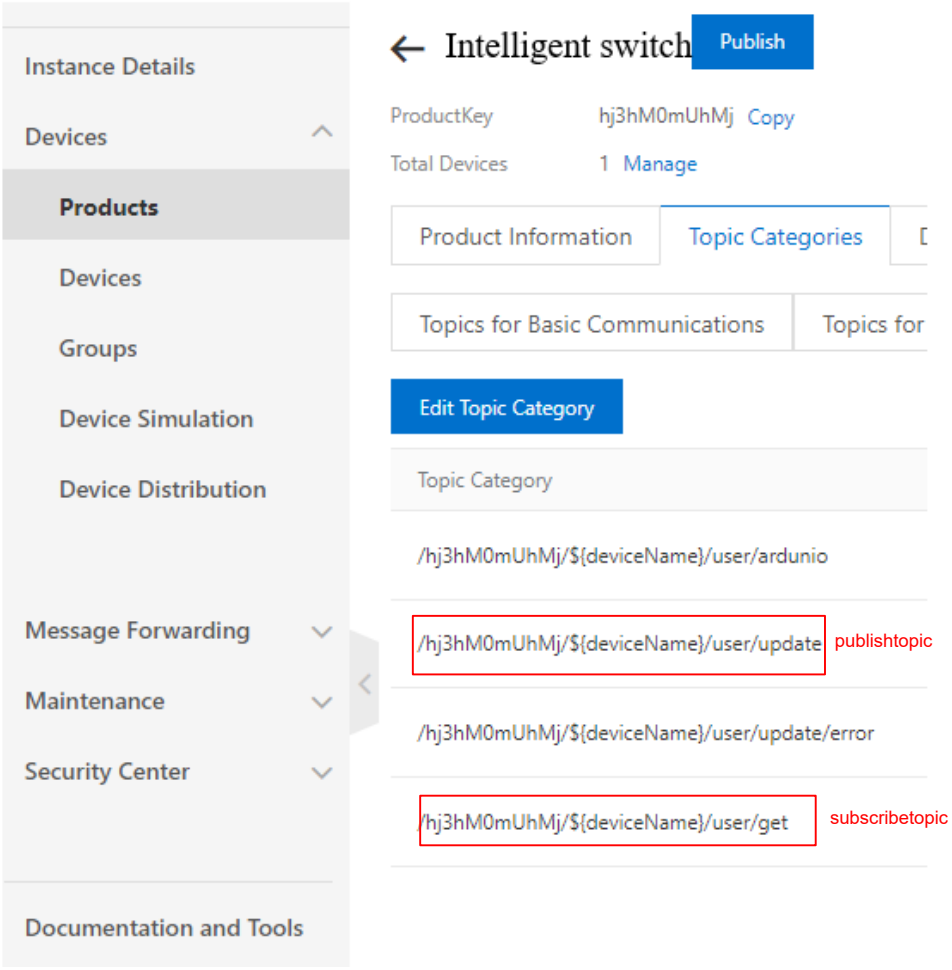
6	bool writeDataTcp(int Dlength, char Dbuffer[])	
	Description	Transmit a data to the TCP server
	Parameter	Dlength: Data length Dbuffer[]: Data
	Return Value	Execution result: true: Succeeded false: Failed
	Note	—
7	String readDataTcp ()	
	Description	Connect to TCP server and read the data transmitted by the server
	Parameter	—
	Return Value	Data transmitted by the TCP server
	Note	—
Performance Functions (Iot)		
8	bool configMqtt(String clientlid,String username,String password,String mqtt_host,int server_port);	
	Description	Configure the MQTT connection parameter
	Parameter	Clientlid: client, user ID. username: user name. password: password. mqtt_host: server address. server_port: server port.
	Return Value	Execution result: true: Succeeded false: Failed
	Note	—
9	bool setPublishTopic(String publishtopic);	
	Description	Set the publish default Topic
	Parameter	The default publish Topic in the Topic list on the IoT platform
	Return Value	Execution result: true: Succeeded false: Failed
	Note	—
10	bool setSubscribetopic(String subscribetopic);	
	Description	Set the subscribe default Topic
	Parameter	The default subscribe Topic in the Topic list on the IoT platform
	Return Value	Execution result: true: Succeeded false: Failed
	Note	—
11	bool setTopic(String topic)	
	Description	Set the custom topic categories for IoT Platform
	Parameter	topic: The transmission media between a message publisher and a subscriber
	Return Value	Execution result: true: Succeeded false: Failed
	Note	—

12	bool writeString (String Dbuffer,String topic)	
	Description	Transmit a data to IoT Platform in String data format
	Parameter	Dbuffer: A character data to be transmitted topic: The transmission media between a message publisher and a subscriber
	Return Value	Execution result: true: Succeeded false: Failed
	Note	—
13	bool writeBytes (char Dbuffer[],int Dlength,String topic)	
	Description	Transmit a data to IoT Platform in Byte data format
	Parameter	Dbuffer[]: Data Dlength: Data length topic: The transmission media between a message publisher and a subscriber
	Return Value	Execution result: true: Succeeded false: Failed
	Note	—
14	void readIotData (String *ReciveBuff,int *ReciveBufflen,String *topic)	
	Description	Connect to the IoT platform and then read the data transmitted by the platform
	Parameter	*ReciveBuff: The received data *ReciveBufflen: The received data length *topic: The topic for the received data
	Return Value	void
	Note	—
Other Functions		
15	bool reset(void)	
	Description	Software reset module
	Parameter	void
	Return Value	Execution result: true: Succeeded false: Failed
	Note	—
16	int sendATCommand(String StringstrCmd,int timeout,uint8_t reTry);	
	Description	Send AT command
	Parameter	StringstrCmd: AT command timeout: timeout reTry: timeout times
	Return Value	Execution result: true: Succeeded false: Failed
	Note	—

Note 1: Login to Alibaba Cloud IoT Platform, Enter the console → Select public instance → Device management → Device → Click to view and enter the corresponding product → Click MQTT connection parameters to view.


Devices
Instance Details
Devices
Products
Devices
Groups
Device Simulation
Device Distribution
Device List
Batch Management
Advanced Search
Add Device
Batch Add
1
2
Arduino_Wifi
Online
Intelligent switch
Device Information
Topic List
TSL Data
Device Shadow
Manage Files
Device Log
Online Debug
Groups
3
4
Product Name
Node Type
Alias
Created At
Nov 22, 2022, 13:43:57
Current Status
Online
MQTT Connection Parameters
More Device Information
SDK Language
Module Information
Message Forwarding
Maintenance
Security Center
MQTT Connection Parameters
clientId
username
Arduino_Wifi&hj3hM0mUhMj
passwd
mqttHostUrl
iot-06z00ac1cwfkn1s.mqtt.iothub.aliyuncs.com
port
1883
Copy
Close

Note 2: Login to Alibaba Cloud IoT Platform, Enter the console → Select public instance → Device management → Device → Click to view and enter the corresponding product → Topic categories → Custom topic.

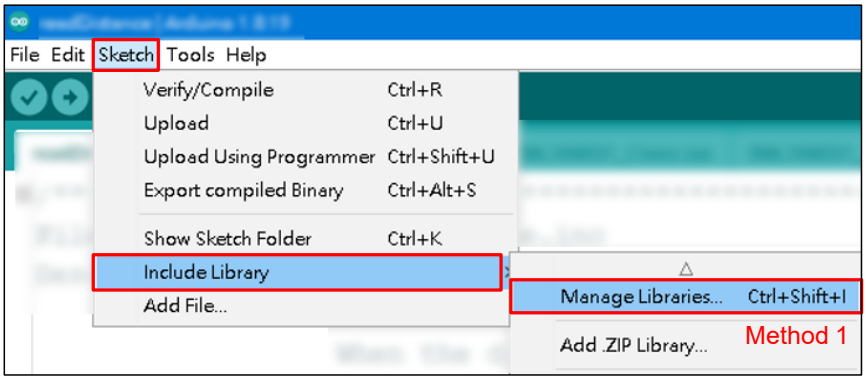


Arduino Lib Download and Installation

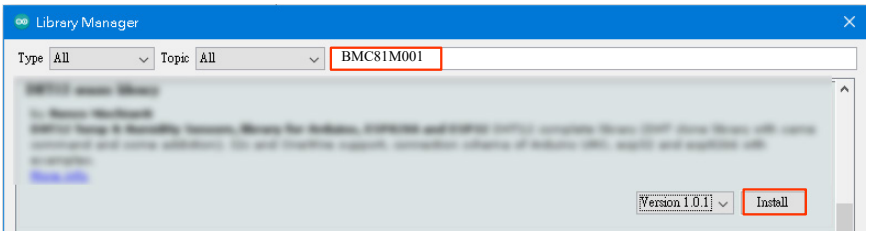
BMC81M001 Library: Refer to the following two methods to install the BMC81M001 Arduino Library.

Method 1: Search for installation

Arduino IDE → Sketch → Include Library → Manage Libraries... → Search BMC81M001 → Install



Search for Installation Step 1

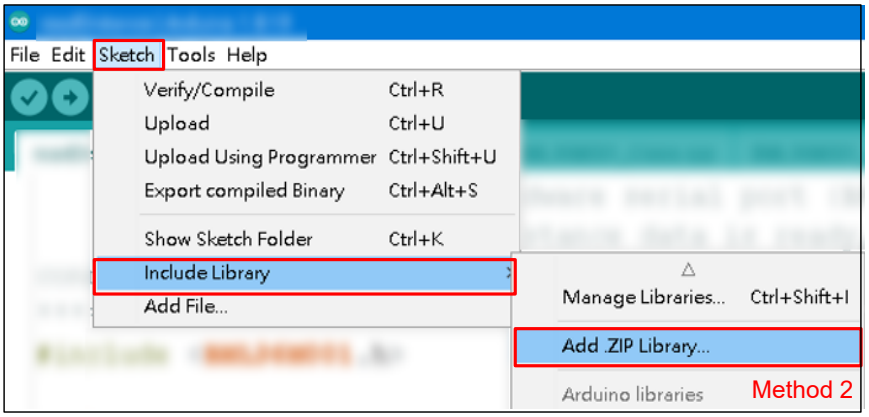


Search for Installation Step 2

Method 2: Download the .ZIP library before adding it

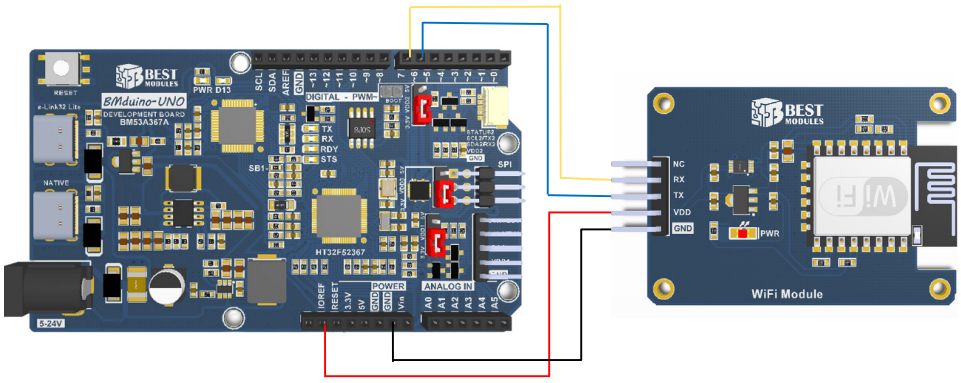
Download the Arduino example (BMC81M001 Library) under the DOCUMENTS menu from the Best Modules website (https://www.bestmodulescorp.com/bmc81m001.html).

Add .ZIP library: Arduino IDE → Sketch → Include Library → Add .ZIP Library...



Arduino Example

Example 1: TCP



Physical Connection Diagram

Example 1 function: The module is connected to the mobile phone hotspot to implement bidirection data communication with the TCP server APP on the mobile phone.

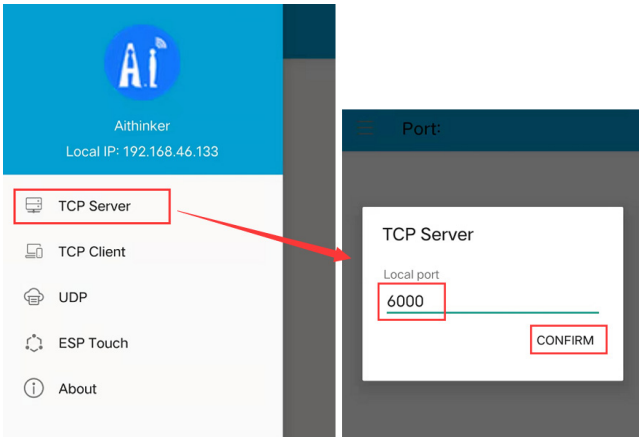
1. Open the example:

   Arduino IDE → File → Examples → Select Lib (BMC81M001) → Select example (TCP)

2. Example Description:

   Firstly, users need to modify the WiFi information and TCP server information in the TCP.h file, shown as follows:

```
// The connected WiFi account password
#define WIFI_SSID "iQOO_Neo_855"  // Mobile phone hotspot name
#define WIFI_PASS "12345678.!"    // Mobile phone hotspot password
// The connected TCP IP and port information
#define IP "192.168.46.133"       // The TCP server IP address of the APP
#define IP_Port  6000             // The TCP server port of the APP
```



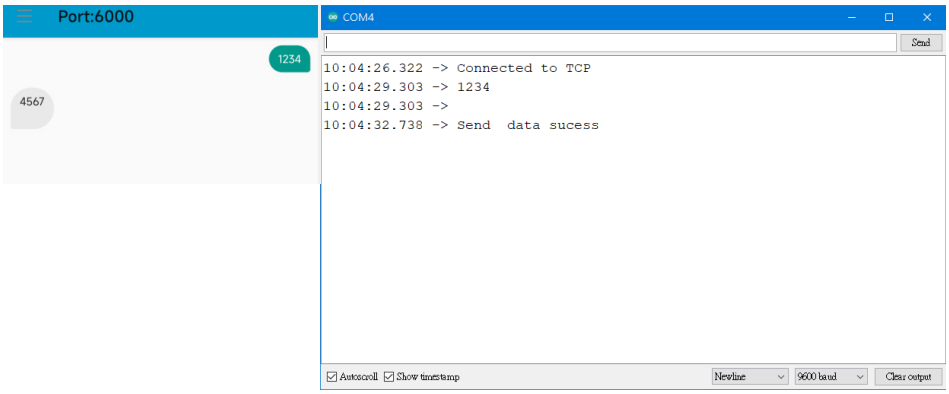
3. Connect to WiFi and TCP server

```
#include "TCP.h"
BMC81M001   Wifi (6,7);
void setup()
{digitalWrite(LED, LOW);
  Serial.begin(9600);          // Configure the serial monitor
  Wifi.begin();                // Module initialisation
  if(!WiFi.connectToAP(WIFI_SSID,WIFI_PASS))// Set the connected
                                            // hotspot name and password
  {
    Serial.println("Disconnect to WIFI");
  }
  if(!WiFi.connectTCP(IP,IP_Port))   // Connect to TCP server (APP)
  {
    Serial.println("Disconnect to TCP server");
  }
  else
  {Serial.println("Connected to TCP"); }
}
```

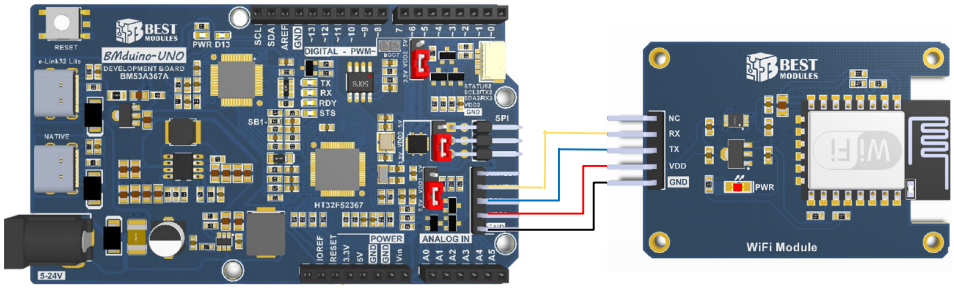
4. In this example, if the data is transmitted on the serial monitor, the data will be directly uploaded to TCP server, and the module information can be viewed on TCP/UDP network debug assistant. The transmitted data on TCP/UDP network debug assistant can be viewed on the serial monitor.

```
void loop() {
tcpBuff=Wifi.readDataTcp();     // Monitor the data received by the module
if(tcpBuff!=0)
{
  Serial.println(tcpBuff);
}
while (Serial.available() > 0)// When the data transmitted by the serial
                              // monitor is received, execute data
                              // transparent transmission
{
  SerialBuff[resLen++] = mSerial.read();
  delay(10);
}
if(resLen>0)
{
  digitalWrite(LED, HIGH);    // LED on when a data is received
  if(WiFi. writeDataTcp (resLen,SerialBuff))// Transmit a data to TCP
                                           // server
  {
    Serial.println("Send  data sucess");
    digitalWrite(LED,LOW);    // LED off when a data is transmitted
                              // successfully
  }
  clearBuff();
}
void clearBuff(){
  memset(SerialBuff,'\0',RES_MAX_LENGTH); // Clear the character array of
                                          // the received serial data
  resLen = 0;
}
```

The program execution result is displayed on the APP and the serial monitor:



Example 2: Alinyun_Iot



Physical Connection Diagram

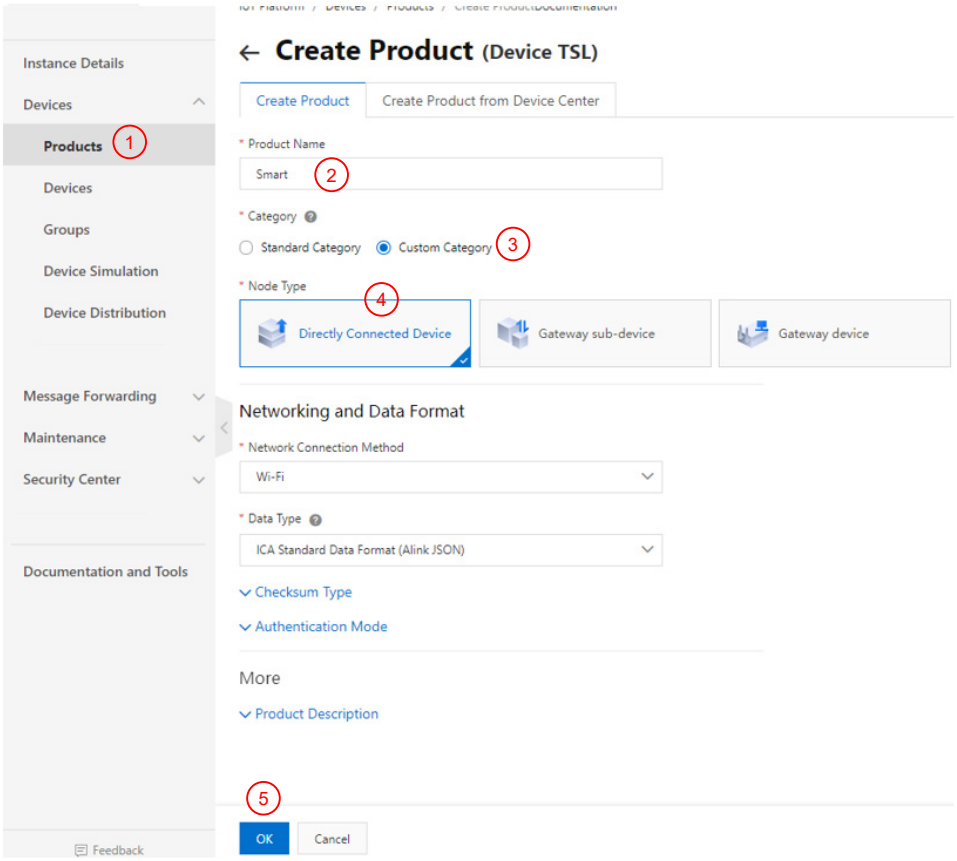
Example 2 function: The module is connected to the mobile phone hotspot and then connected to the Alibaba Cloud Platform instance, which can implement data transparent transmission.

1. Open the example:

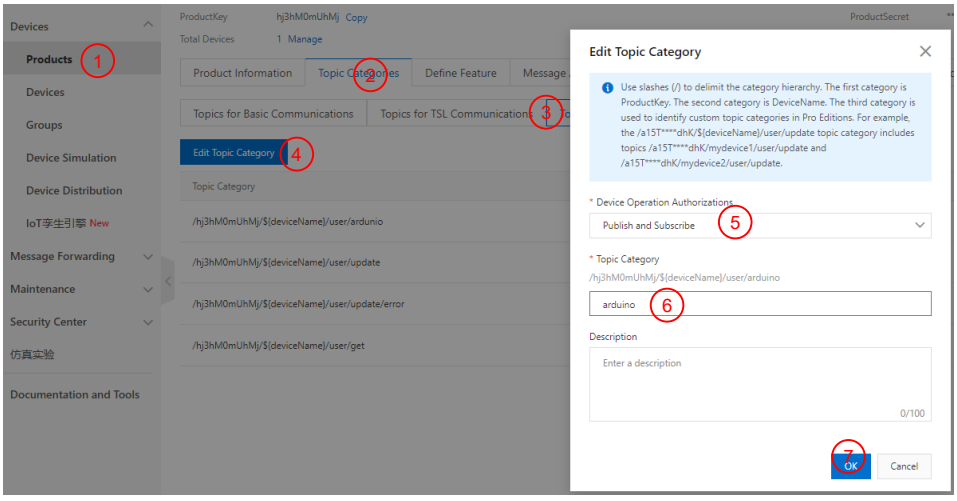
   Arduino IDE → File → Examples → Select Lib(BMC81M001) → Select the corresponding example (Alinyun_Iot).

2. Login to IoT platform (https://account.alibabacloud.com/) and register, select public instance → create products and devices → custom topic, fill MQTT parameters and various topics into the Aliyun_Iot.h file. The specific process is as follows:

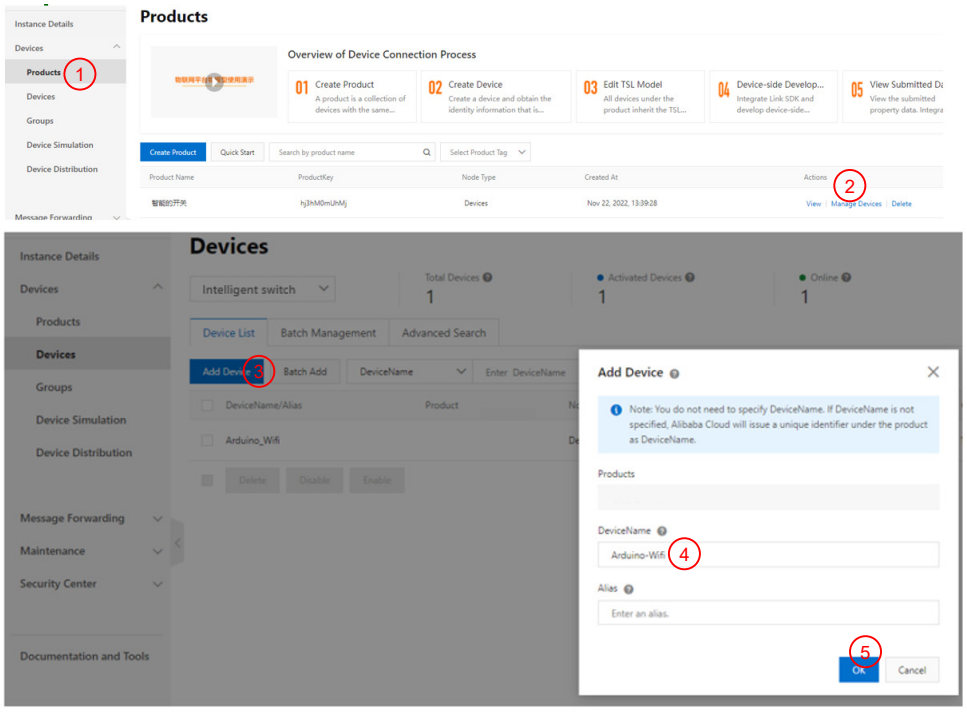
1) Create Product: After entering the public instance, click the menu device management on the left → Product → Create product



2) Custom topic, define the data transmission media: Device management → Product → View the newly created product → Topic categories → Custom topic

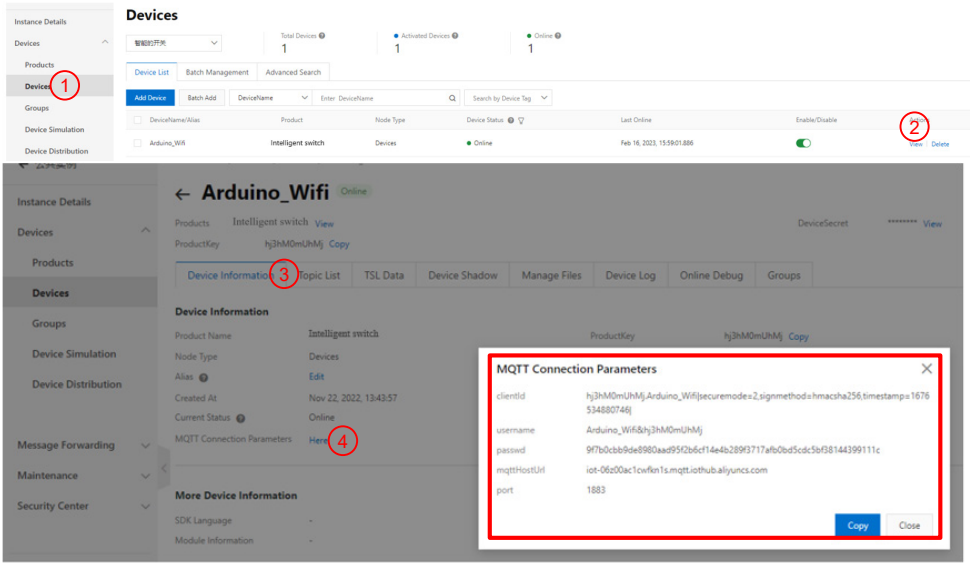


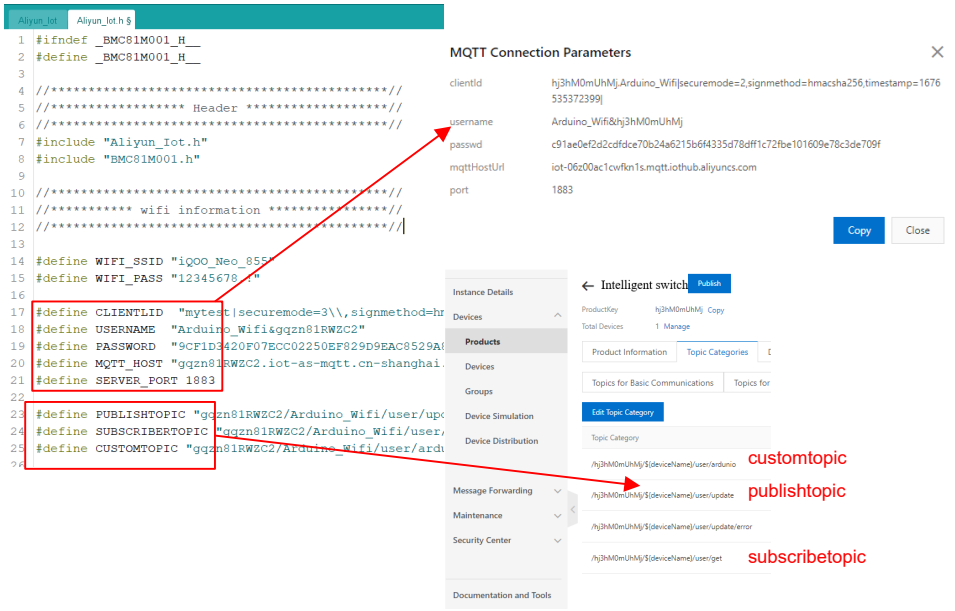
3) Return to “Product” interface → “Manage device” → “Add device” → DeviceName, Click “OK”.



4) Check the MQTT connection parameters of the device: Device → Check the newly created device Arduino-Wifi → Device information → Check the MQTT parameters. Viewing the device topic, which is in the same way as the step 2 for custom topic.

Fill the CLIENTLID, USERNAME, PASSWORD, MQTT_HOST, SERVER_PORT and other MQTT parameters into the Aliyun_Iot.h file. Fill the PUBLISHTOPIC, SUBSCRIBERTOPIC, CUSTOMTOPIC and other topic categories into the Aliyun_Iot.h file.






```
// The connected WiFi account password
#define WIFI_SSID "iQOO_Neo_855"      // Mobile phone hotspot name
#define WIFI_PASS "12345678.!"        // Mobile phone hotspot password
// aliyun MQTT information
#define CLIENTLID "mytest|securemode=3\\,signmethod=hmacsha1\\,
                   timestamp=6789|"
#define USERNAME "Arduino_Wifi&gqzn81RWZC2"
#define PASSWORD "9CF1D3420F07ECC02250EF829D9EAC8529A8D672"
#define MQTT_HOST "gqzn81RWZC2.iot-as-mqtt.cn-shanghai.aliyuncs.com"
#define SERVER_PORT 1883              // Part number
// aliyun TOPIC information
#define PUBLISHTOPIC "gqzn81RWZC2/Arduino_Wifi/user/update"
#define SUBSCRIBERTOPIC "gqzn81RWZC2/Arduino_Wifi/user/get"
#define CUSTOMTOPIC "gqzn81RWZC2/Arduino_Wifi/user/ardunio"
```

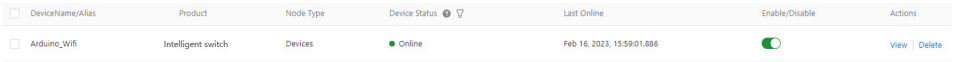
Note: When the CLIENTID was defined, the "\\" should be added in front of ",".

3. Example Description:

According to the data generated by Alibaba Cloud Platform to initialize the module. This process includes software reset module, connect to WiFi and connect to Alibaba Cloud IoT, shown as follows:

```
#include "Aliyun_Iot.h"
BMC81M001       Wifi(&Serial1);
void setup()
{
  digitalWrite(LED, LOW);
  Serial.begin(9600); // Configure the serial monitor
  Wifi.begin();       // Module initialisation and configuration
  Wifi.reset();
  Serial.print("WIFI Connection Results:");
  if(Wifi.connectToAP(WIFI_SSID,WIFI_PASS)==0) // According to the name
                      // and password to connect to WiFi
  {
    Serial.println("fail");
  }
  else {Serial.println("success");}
  Serial.print("Aliyun Connection Results:");
  Wifi.sendATCommand("AT+CIPSNTPCFG=1,8,\"ntp1.aliyun.com\"",1000,2);
  // Connect to aliyun
  if(Wifi.configMqtt(CLIENTLID,USERNAME,PASSWORD,MQTT_HOST,
     SERVER_PORT)==0) // Configure MQTT parameter
  {
    Serial.println("fail");
  }
  else
  {
    Serial.println("success");
    Wifi.setPublishTopic(PUBLISHTOPIC); // Configure the subscribe and
    // publish Topic after the connection is successful
    Wifi.setSubscribetopic(SUBSCRIBERTOPIC);
  }
  Serial.print("Topic set Results:");
  if(Wifi.setTopic(CUSTOMTOPIC)==0) // Set the custom Topic
  {
    Serial.println("fail");
  }
  else {Serial.println("success");}
  delay(200);}
}
```

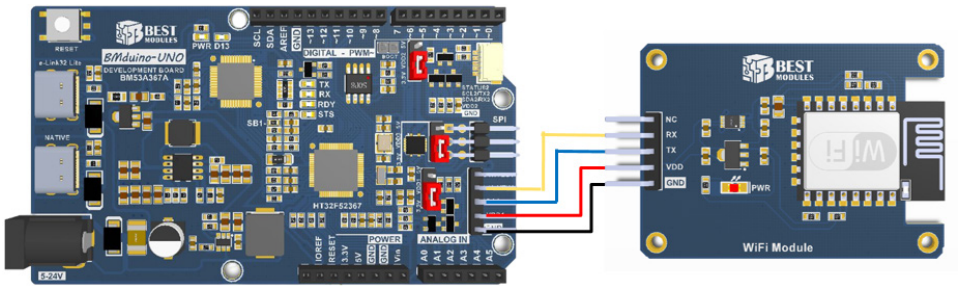
4. After executing, users can view the device status on the platform.



5. In this example, when the serial monitor transmits a data, the module will only execute the data transparent transmission function. The data received by the serial monitor is directly uploaded to the cloud platform, and the information uploaded by the module can be recorded and viewed in the platform log.

```
void loop()
{
  Wifi.readIotData(&aliyunReciveBuff,&aliyunReciveBufflen,&recTopic);
  // Monitor the data received by the module
  if(aliyunReciveBufflen)
  {
    Serial.println(aliyunReciveBufflen);
    Serial.println(aliyunReciveBuff);
  }
// When the data transmitted by the serial monitor is received, execute
// data transparent transmission
  while (Serial.available() > 0) // Receivehe the data from the serial
// monitor
  {
    SerialBuff[resLen++] = Serial.read();
    delay(10);
  }
  if(resLen>0)
  {
    digitalWrite(LED, HIGH);
    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); // Clear the received array
  // data
  resLen = 0;
}
```

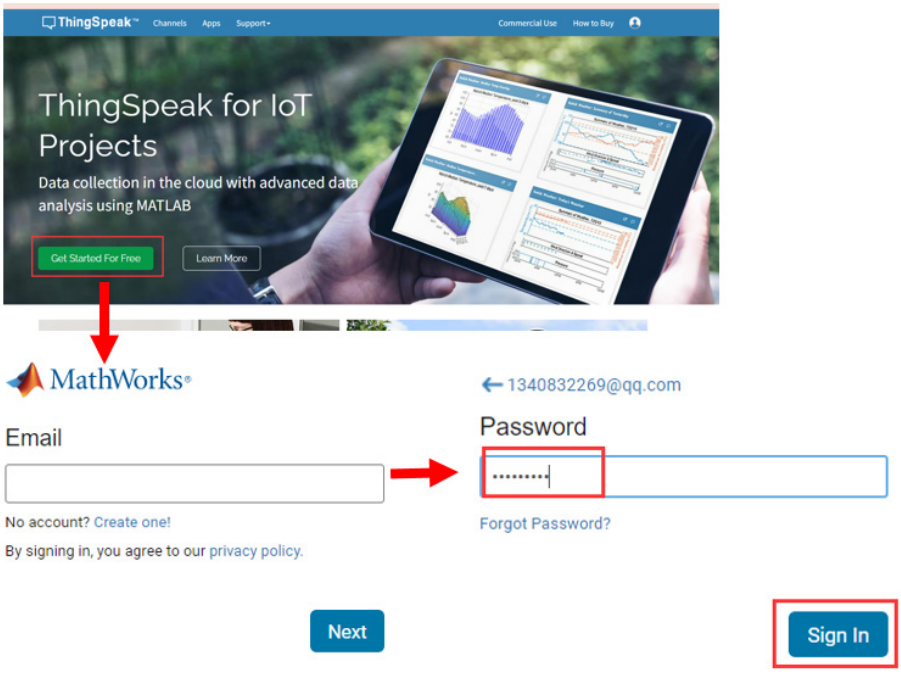
Example 3: ThingSpeak



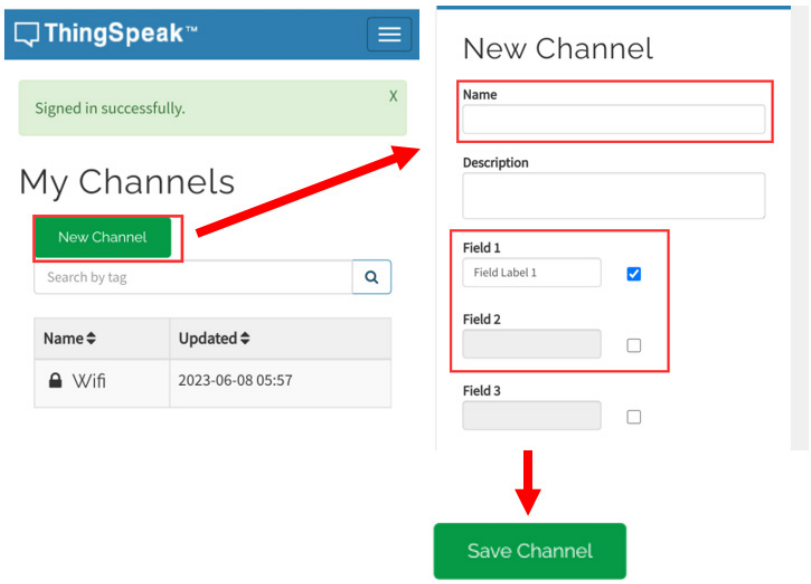
Physical Connection Diagram

Example 3 function: The module is connected to the mobile phone hotspot and then connected to the channel of the ThingSpeak platform, the data will be transmitted to display on the platform.

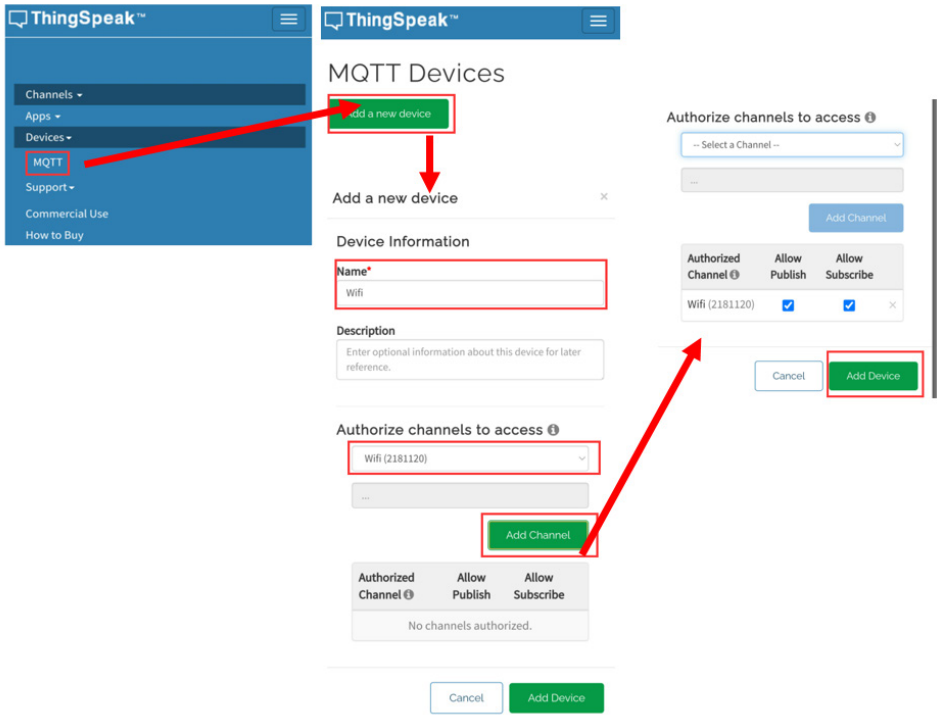
1. Open the example: Arduino IDE→File→Examples→Select Lib(BMC81M001)→Select the corresponding example (ThingSpeak).
2. Login to the ThingSpeak platform (https://thingspeak.com/), after registering the account, create a new data channel, and then create a new MQTT device, fill the MQTT parameters into the ThingSpeak.h file.
   1) Register an account and enter the platform: click "Get Start For Free" to enter the platform interface, register an account and log in.



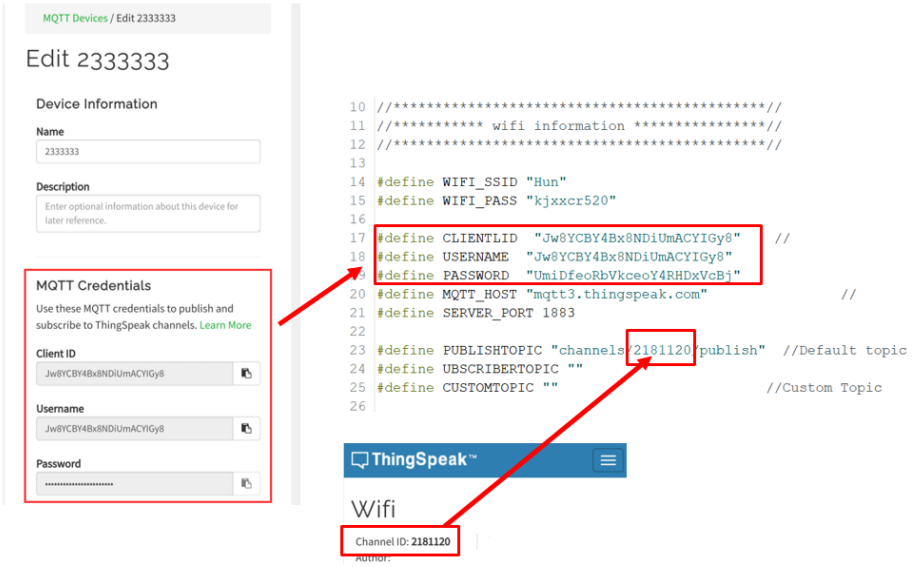
2) Create channels: click New Channel, fill in the Name and number of open data fields, and save the channel.



3) Create MQTT device: fill into the device name, select the new channel, click "Add Channel", wait for "Allow Publish" and "Sublish" is OK, then click "Add Device".



4) Click "edit", check the new MQTT device parameter, fill the data into the ThingSpeak.h file.




```
// The hotspot information to be connected
#define WIFI_SSID "iQOO_Neo_855"      // Mobile phone hotspot name
#define WIFI_PASS "12345678.!"        // Mobile phone hotspot password
// ThingSpeak MQTT information
#define CLIENTLID  "Jw8YCBY4Bx8NDiUmACYIGy8"
#define USERNAME  "Jw8YCBY4Bx8NDiUmACYIGy8"
#define PASSWORD  "UmiDfeoRbVkceoY4RHDxVcBj"
#define MQTT_HOST "mqtt3.thingspeak.com"     // Server address
#define SERVER_PORT 1883              // Port number
// ThingSpeak TOPIC information
#define PUBLISHTOPIC "channels/2181120/publish"
```

3. Example Description:

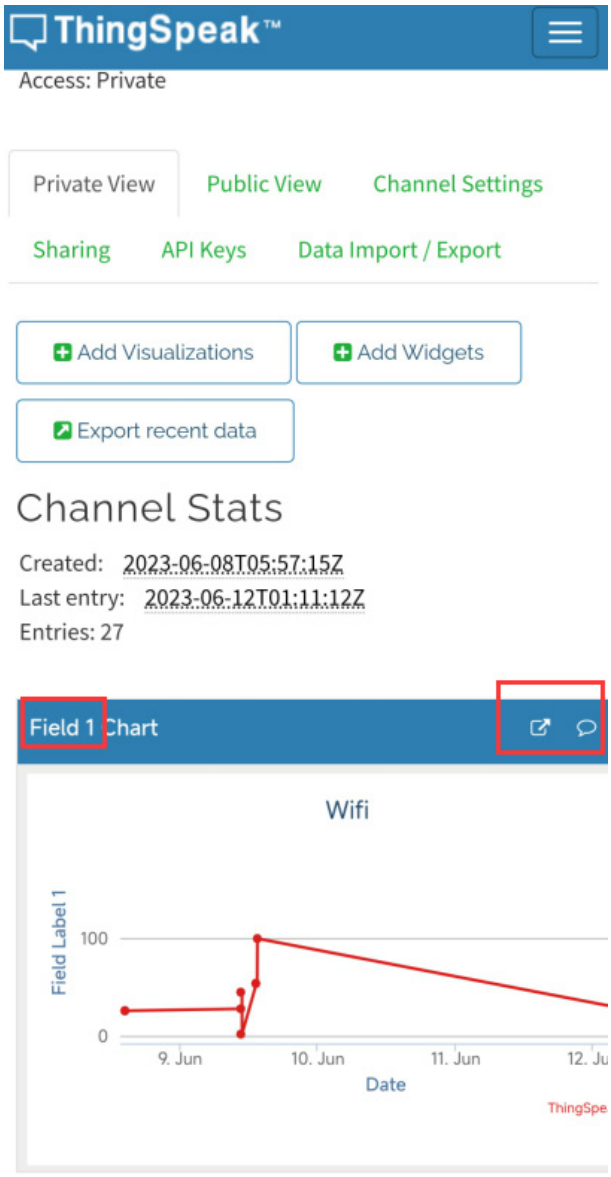
After module initialization, connect the hotspot, configure the MQTT parameters, connect the channel of the ThingSpeak platform, and the data will be transmitted to the platform. Shown as below:

```
#include "ThingSpeak.h"
BMC81M001       Wifi(&Serial1);
void setup()
{
  digitalWrite(LED, LOW);
  Serial.begin(9600);         // Configure the serial monitor
  Wifi.begin();               // Module initialisation and configuration
  Wifi.reset();
  Serial.print("WIFI Connection Results:");
  if(Wifi.connectToAP(WIFI_SSID,WIFI_PASS)==0) // According to the name
                              // and password to connect to WiFi
  {
    Serial.println("fail");
  }
  else {Serial.println("success");}
  Serial.print("ThingSpeak Connection Results:");
  if(Wifi.configMqtt(CLIENTLID,USERNAME,PASSWORD,MQTT_HOST,
     SERVER_PORT)==0) // Configure MQTT parameter
  {
    Serial.println("fail");
  }
  else
  {
    Serial.println("success");
  }
 delay(200);}
```

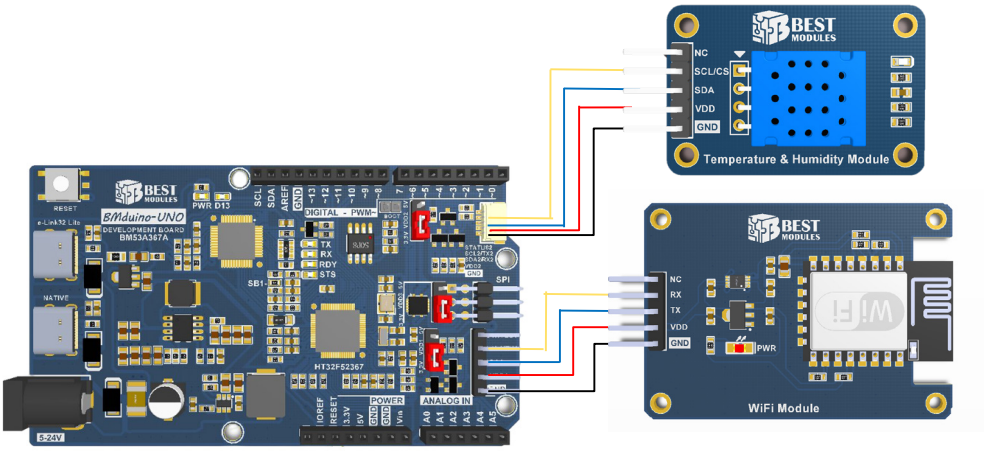
4. In this example, a data is transmitted on the serial port monitor, and the data is uploaded directly to the cloud platform, where it can be seen on the channel interface.

```
void loop()
{
  Wifi.readIotData(&ReciveBuff,&ReciveBufflen,&recTopic);// Monitor the
                                        //data received by the module
  if(ReciveBufflen)
  {
    Serial.println(ReciveBufflen);
    Serial.println(ReciveBuff);
  }// When the data tramsmitted by the serial monitor is received,
   // execute data transparent transmission
  while (Serial.available() > 0)// Receive the dta from the serial
                                // monitor
  {
    SerialBuff[resLen++] = Serial.read();
    delay(10);
  }
  if(resLen>0)
  {
    digitalWrite(LED, HIGH);
    DATA_BUF = "field1=";  // Send the channel field 1 data
    DATA_BUF += SerialBuff;
    topic = PUBLISHTOPIC;
    if(Wifi.writeString(DATA_BUF,topic))
    {
      Serial.println("Send String data sucess");
    }
    clearBuff();
    digitalWrite(LED, LOW);
    }
  }
void clearBuff(){
  memset(SerialBuff,'\0',RES_MAX_LENGTH); // Clear the received array
                                          // data
  resLen = 0;
}
```

5. The ThingSpeak will display the results and users can view the data received from the chart.



Example 4: ThingSpeakPublish

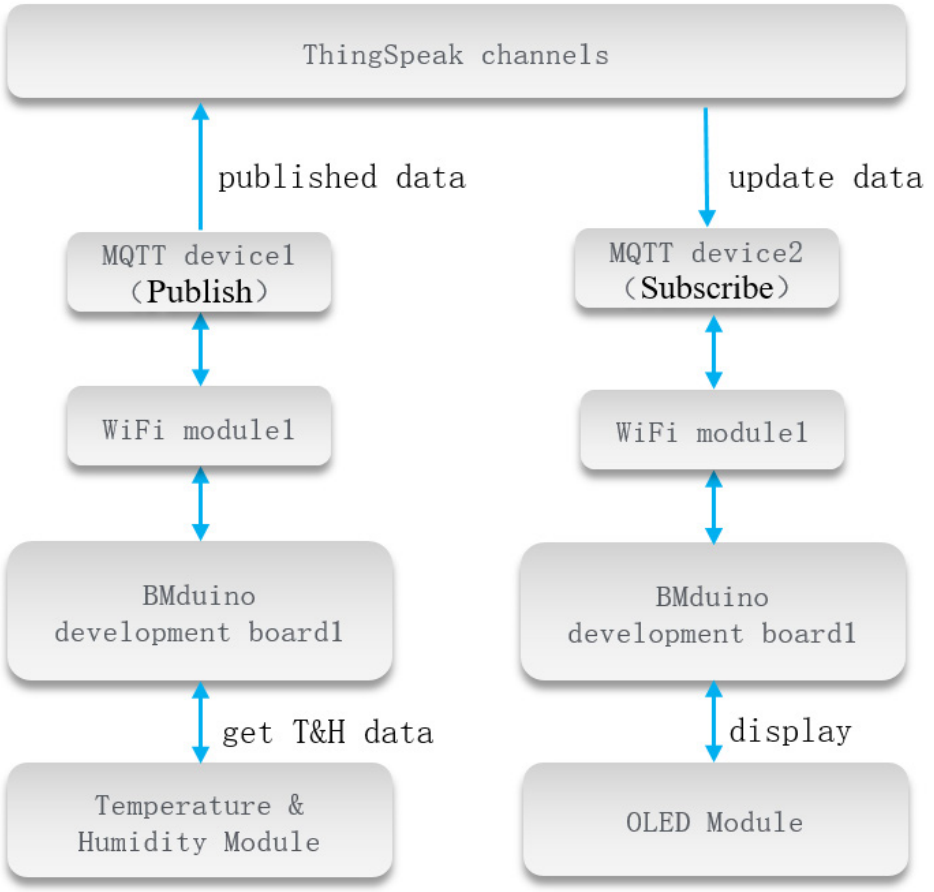


Physical Connection Diagram

Example 4 and Example 5 can be used together to achieve remote data exchange between two WiFi devices, one as a Publish end and the other as a Subscribe end. The implementation functions are as follows:

Publish end: Upload the temperature and humidity data of the Temperature and Humidity module BME33M251 to ThingSpeak;

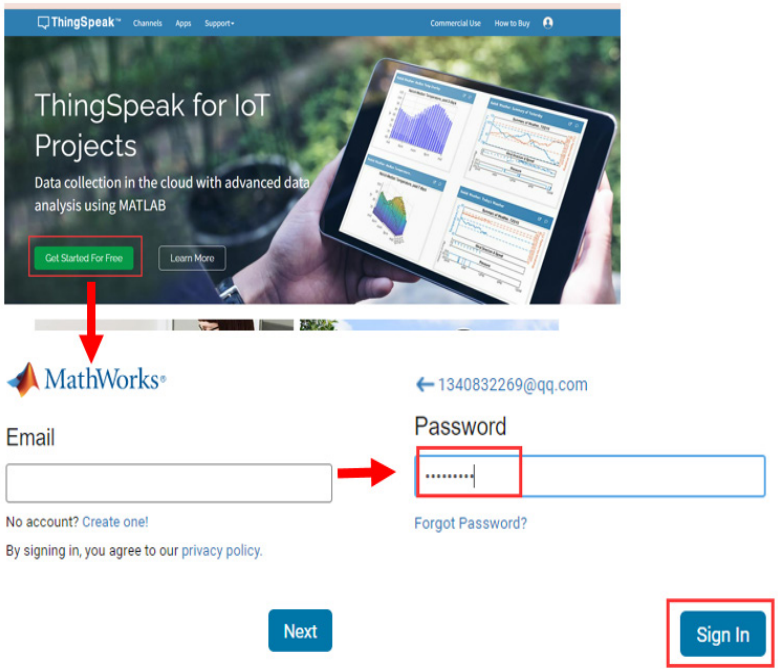
Subscribe end: Read the temperature and humidity data which is uploaded by the Publish end from ThingSpeak and then display the data on the OLED module;



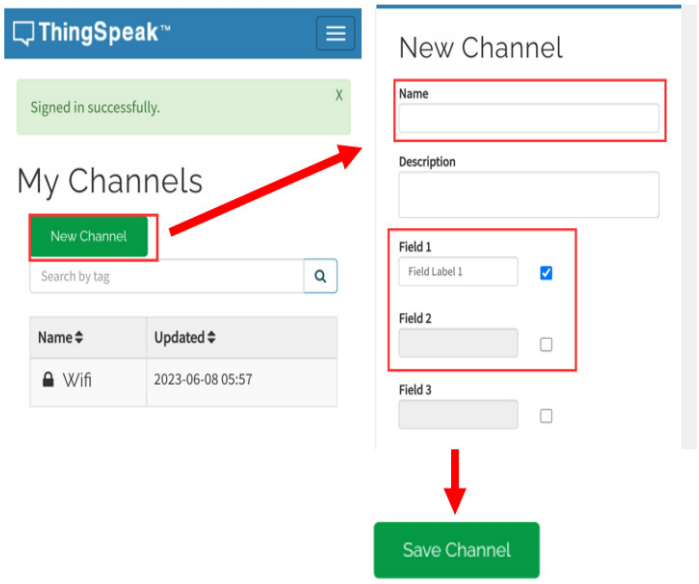
Example 4 function: Read the temperature and humidity data of the Temperature and Humidity module BME33M251 and publish the data to the channel of the ThingSpeak platform through MQTT device 1 (Publish).

1. Open the example: Arduino IDE → File → Examples → Select Lib (BMC81M001) → Select the corresponding example (ThingSpeakPublish).
2. Login to the ThingSpeak platform (https://thingspeak.com), after registering the account, create a new data channel, (wifi t&h), create a new MQTT device (Publish), fill the MQTT parameters into the ThingSpeak.h file.

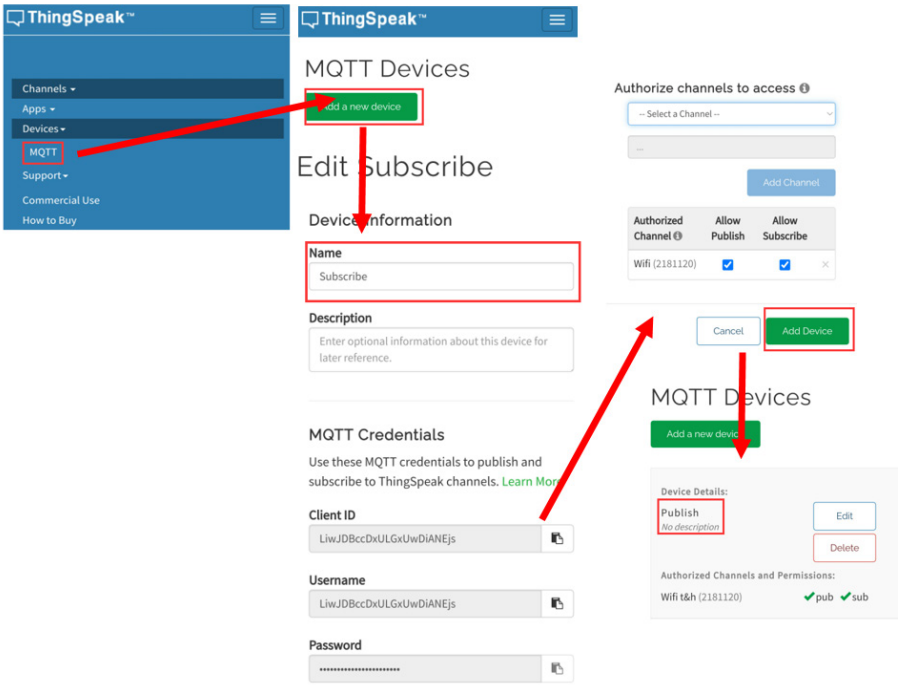
1) Register an account and enter the platform: click “Get Start For Free” to enter the platform interface, register an account and log in.



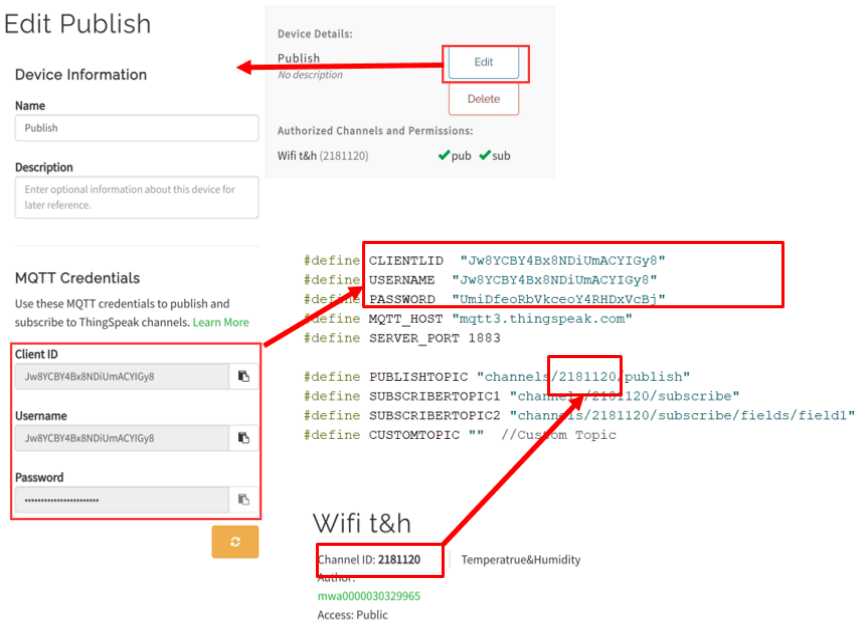
2) Create channels: click New Channel, fill into the Name (e.g. wifi t&h), Field1 (e.g. humidity) and Field2 (e.g. temperature), save the channel.



3) Create MQTT device 1: fill into the device name (Publish), authorize the publish and subscribe of the channel wifi t&h, click Add channel, wait for the Allow Publish and Sublish is OK, then click Add Device. This Publish device is used to upload the temperature and humidity module data.



4) Click the "edit" button on the right of the device (Publish), check the MQTT device (Publish) MQTT parameter, fill the data into the ThingSpeakPublish.h.



3. Example Description:

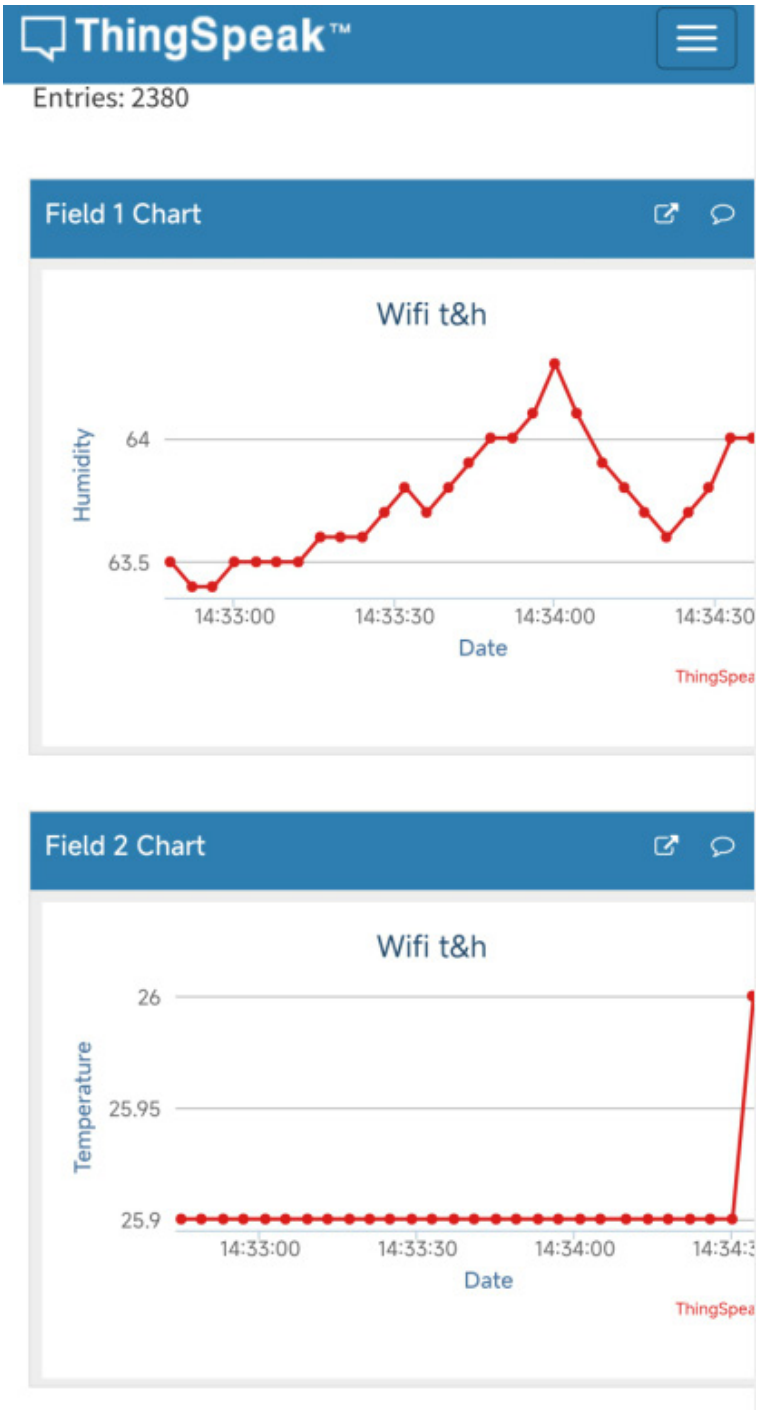
After module initialization, connect the hotspot, configure the MQTT parameters, connect the channel of the ThingSpeak platform. Shown as below:

```
#include "ThingSpeak.h"
#include <BMD31M090.h>
#include <BM25S2021-1.h>      // Temperature and Humidity module
BMC81M001   Wifi(&Serial1);
BM25S2021_1 BMht(&Wire2);
void setup()
{
  Serial.begin(9600);
  BMht.begin();
  Wifi.begin();                // WIFI module,
                    // Temperature and Humidity moduleinitialization
  Wifi.reset();
  Serial.print("WIFI Connection Results: ");
  if(Wifi.connectToAP(WIFI_SSID,WIFI_PASS)==0)// connect the hotspot
  {
    Serial.println("fail");
  }
  else {Serial.println("success");}
  Serial.print("ThingSpeak Connection Results: ");
  if(Wifi.configMqtt(CLIENTLID,USERNAME,PASSWORD,MQTT_HOST,SERVER_
     PORT)==0)
  // According the MQTT to connect the ThingSpeak platform
  {
    Serial.println("fail");
  }
  else {Serial.println("success");}
  delay(200);
  Wifi.setPublishTopic(PUBLISHTOPIC); // Subscribe Topic,
                                      // update the channel data
  Wifi.setSubscribetopic(SUBSCRIBERTOPIC2);
  Wifi.setSubscribetopic(SUBSCRIBERTOPIC1);
  topic = PUBLISHTOPIC; // Publish Topic
}
```

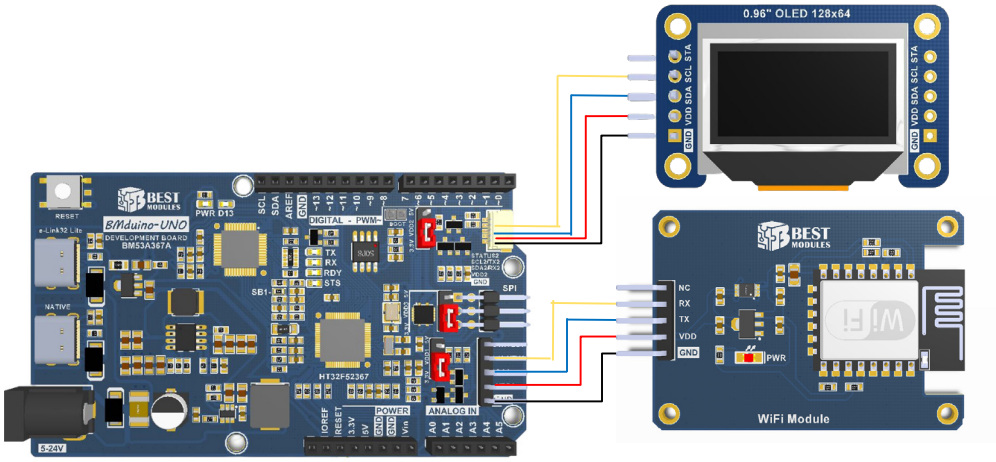
4. Polling the Temperature and Humidity module data, sending the humidity data to field 1 in the channel and the temperature data to field 2 in the channel.

```
void loop()
{
  data1=BMht.readHumidity();    // Get the humidity data
  Humidity=String(data1,2);     // Convert the floating data to a string
  DATA_BUF = "field1=";         // The channel 1 data
  DATA_BUF += Humidity;
  if(Wifi.writeString(DATA_BUF,topic)) // Publish the data to channel 1
  {
    Serial.println("Send String data sucess");
    delay(1000);
  }
  clearBuff();
  data1=BMht.readTemperature(false); // Get the temperature data
  Temperature=String(data1,2); // Convert the floating data to a string
  DATA_BUF = "field2=";         // The channel 2 data
  DATA_BUF += Temperature;
  if(Wifi.writeString(DATA_BUF,topic)) // Publish the data to channel 2
  {
    Serial.println("Send String data sucess");
    delay(1000);
  }
  clearBuff();
  delay(2000);
}
```

5. The ThingSpeak will display the results and users can view the data received from the chart.



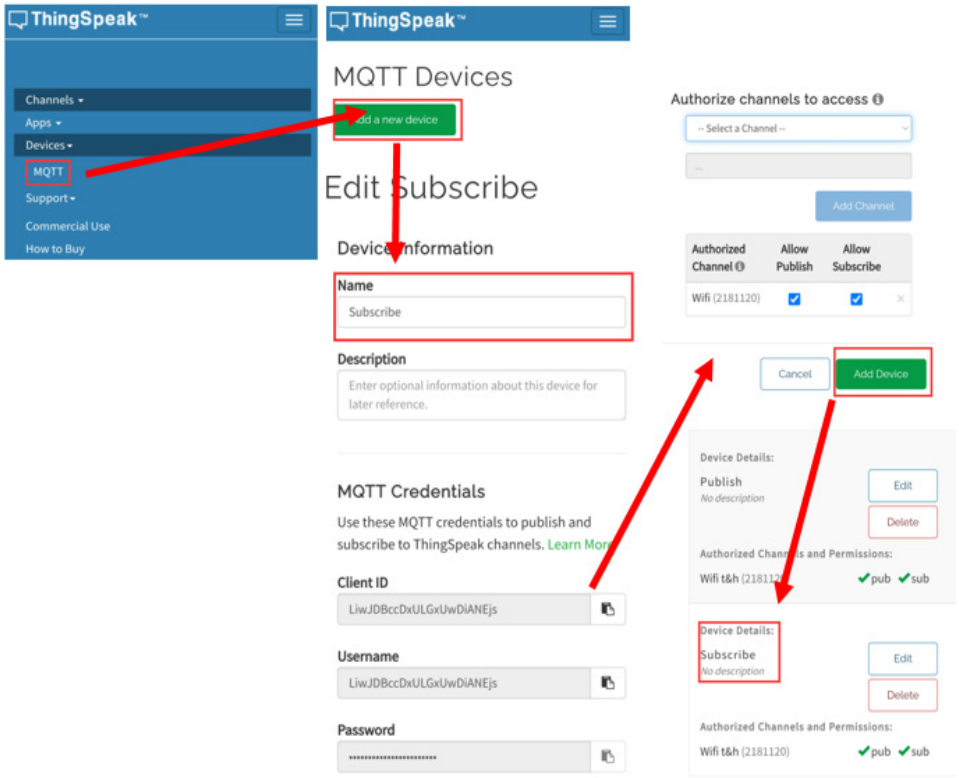
Example 5: ThingSpeakSubscribe



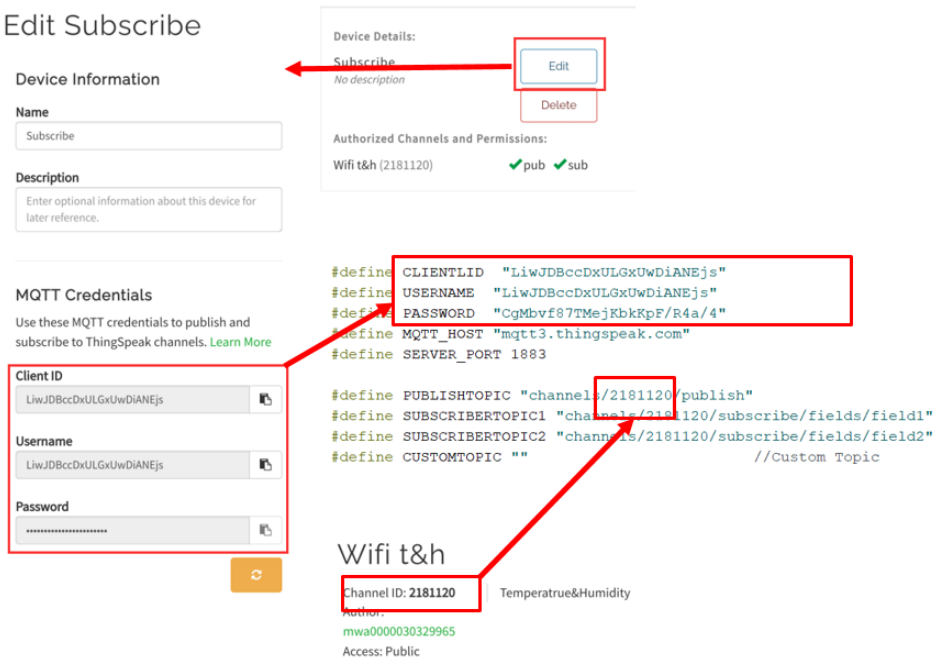
Physical Connection Diagram

Example 5 function: Subscribe to the data update Topic, receive the latest temperature and humidity data of the corresponding channel of ThingSpeak by creating and connecting MQTT device 2 (Subscribe), and display it on the OLED module.

1. Open the example: Arduino IDE → File → Examples → Select Lib (BMC81M001) → Select the corresponding example (ThingSpeakSubscribe).
2. ThingSpeak channel does not need to be created, and the wifi t&h in Example 4 can be used. Create a new MQTT device (Subscribe) and fill the parameters into the ThingSpeakSubscribe.h file.
   1) Create MQTT device 2: fill into the device name(Subscribe), authorize the publish and subscribe of the channel wifi t&h, click Add channel, wait for the Allow Publish and Sublish is OK, then click Add Device. This Subscribe device is used to receive the latest updated data from the channel.



2) Click the "edit" button on the right of the device (Subscribe), check the MQTT device (Subscribe) MQTT parameter, fill the data into the ThingSpeakSubscribe.h file.



3. Example Description:

After module initialization, connect the hotspot, configure the MQTT parameters, connect the channel of the ThingSpeak platform. Shown as below:

```
#include "ThingSpeakRead.h"
#include <BMD31M090.h>
#include <BM25S2021-1.h>                        // OLED module
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) // connect the hotspot
  {
    Serial.println("fail");
  }
  else {Serial.println("success");}
  Serial.print("ThingSpeak Connection Results: ");
  if(Wifi.configMqtt(CLIENTLID,USERNAME,PASSWORD,MQTT_HOST,SERVER_
     PORT)==0)// According the MQTT to connect the ThingSpeakplatform
  {
    Serial.println("fail");
  }
  else {Serial.println("success");}
  delay(200);
  Wifi.setPublishTopic(PUBLISHTOPIC);
  // Subscribe Topic, update the channel data
  Wifi.setSubscribetopic(SUBSCRIBERTOPIC2);
  Wifi.setSubscribetopic(SUBSCRIBERTOPIC1);
}
```

4. Polling the platform to check whether there is updated data, identify the corresponding data if there is updated data and displayed on the OLED module.

```
void loop()
{
  Wifi.readIotData(&ReciveBuff,&ReciveBufflen,&topic);
  // Receive data from the platform
  if(ReciveBufflen)
  {
    if(topic==SUBSCRIBERTOPIC1) // The data in channel 1 is humidity
    {
      ReciveBuff.toCharArray(OledBuff,7);
      BMD31.drawString(40,2,(u8*)OledBuff); // OLED display
      BMD31.drawString(85,2,(u8*)"%");
      ReciveBufflen=0;
    }
    if(topic==SUBSCRIBERTOPIC2) // The data in channel 2 is temperature
    {
      ReciveBuff.toCharArray(OledBuff,7);
      BMD31.drawString(40,6,(u8*)OledBuff); // OLED display
      BMD31.drawString(85,6,(u8*)"C");
      ReciveBufflen=0;
    }
  }
  clearBuff();                              // Clear data
}
```

5. The OLED display results are as follows: