MCU Quick Start

Device Development > Tuya Development Boards > Tuya Sandwich Evaluation Kits > Development Guide

Version: 20200214

Online Version

Contents

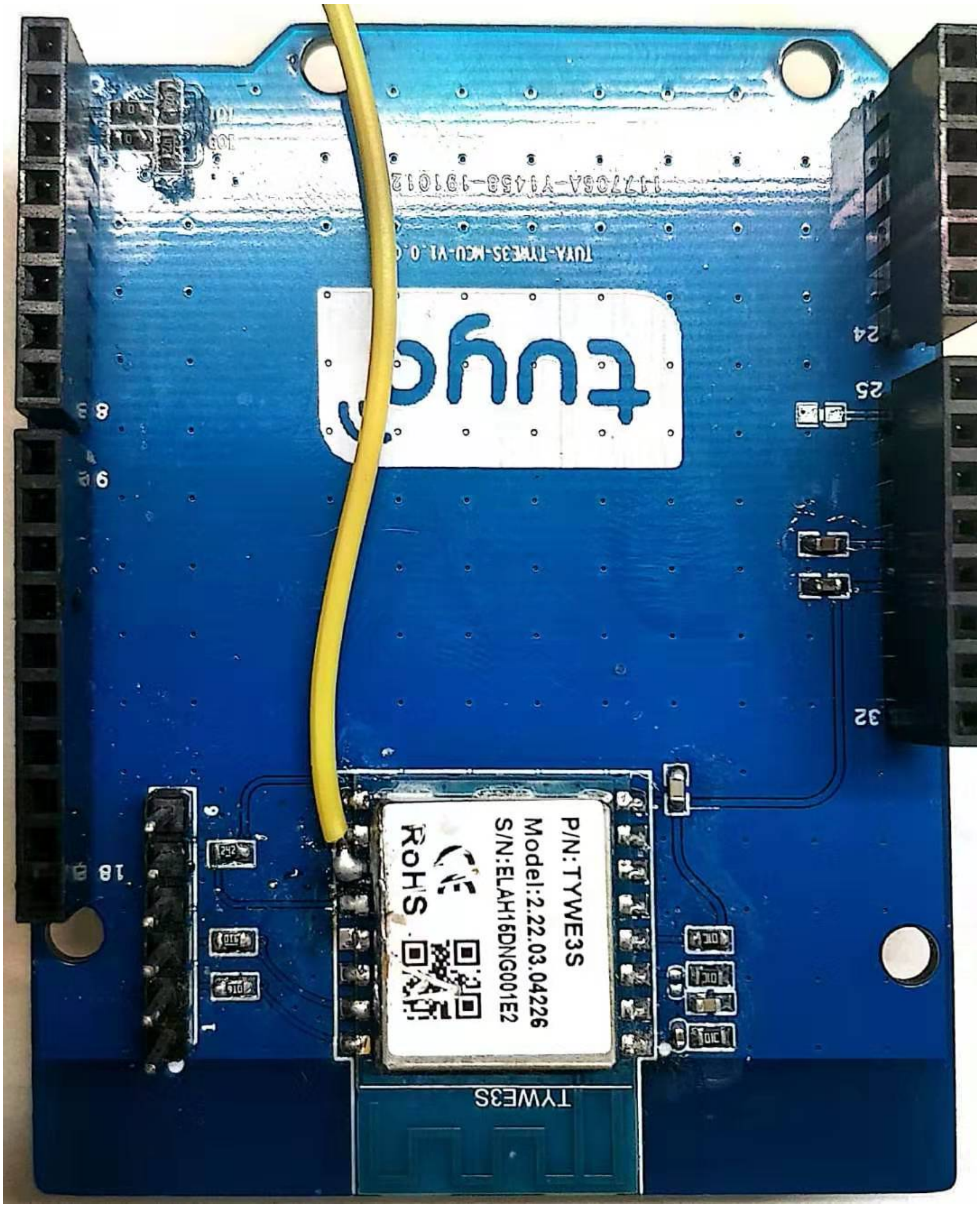
1 Introduction

After the development environment of Tuya Sandwich Development Board is set up, this article will introduce the connection between Tuya Sandwich development board and each end in development, how to create a development project based on specific products, and how to download the developed program to Tuya Sandwich Development Board. .

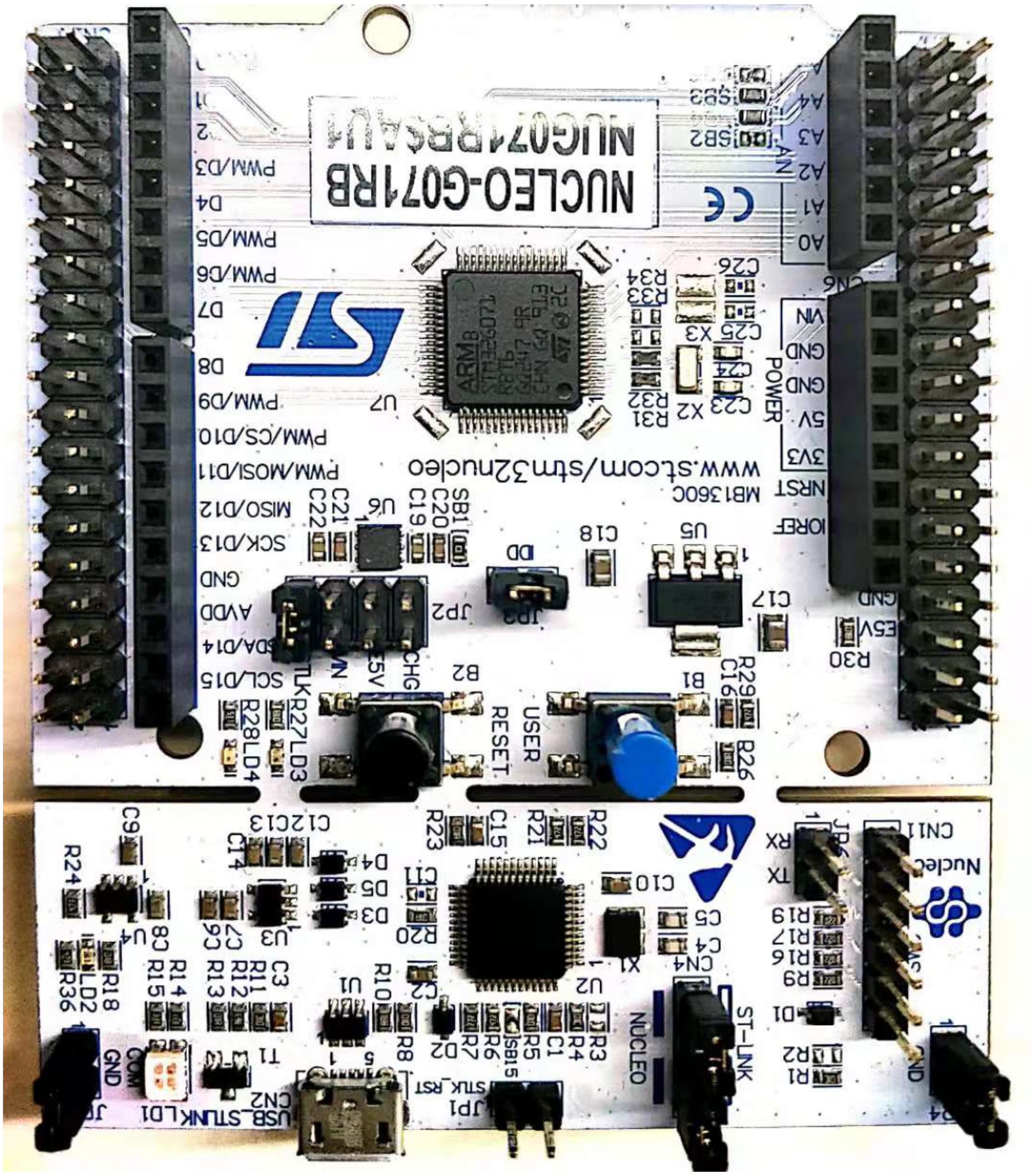
2 Bill of Materials

2.1 Hardware

1、Tuya Sandwich Wi-Fi MCU Communication Board (E3S) ;



2、ST Nucleo Development Board;



3、Micro-USB cable with data transmission function.

2.2 Software

Arduino IDE.

3 Project creation

As developers, we need to create projects to do 2 jobs first:

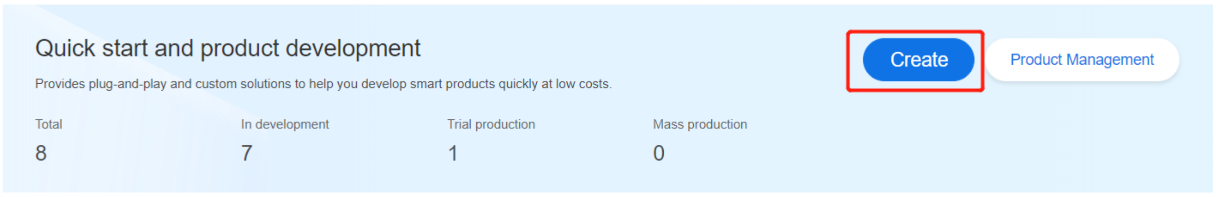
- Implement the most basic communication functions between Tuya Sandwich Main control board and Tuya Sandwich Wi-Fi MCU Communication Board (E3S);
- Writing the Tuya Sandwich's specific DP functions according to their own needs.

3.1 Get SDK

In order to realize the communication between Tuya Sandwich Development Board and Tuya E3S Wi-Fi module, we need to use the SDK package generated by Tuya IoT platform according to the product.

Taking the realization of the product socket as an example, the steps to obtain the SDK development kit for the product socket are as follows:

3.1.1 Create product

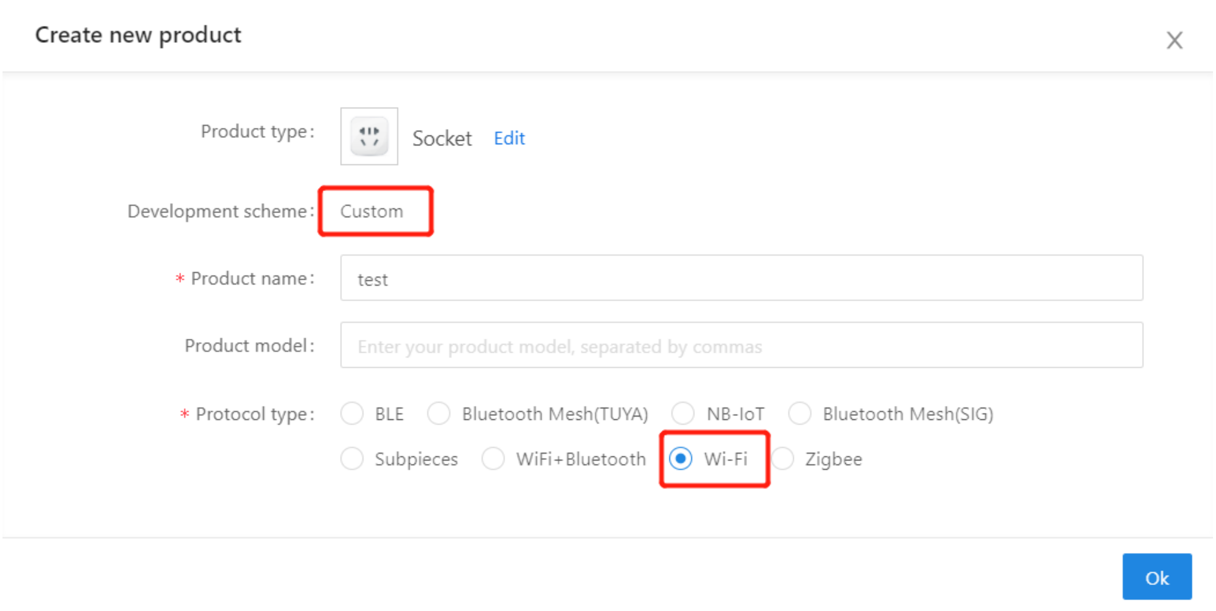

Quick start and product development

Provides plug-and-play and custom solutions to help you develop smart products quickly at low costs.

Total	In development	Trial production	Mass production
8	7	1	0

Figure 1: 微信截图 _20200212181245.png

Choose a development-free solution


Create new product

Product type: Socket Edit

Development scheme: Custom

* Product name: test

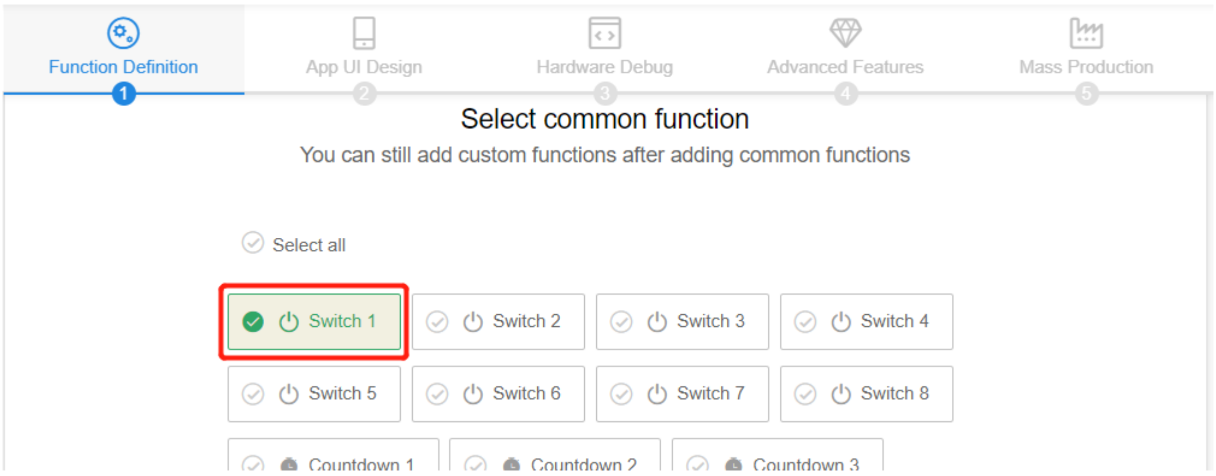
Product model: Enter your product model, separated by commas

* Protocol type: BLE Bluetooth Mesh(TUYA) NB-IoT Bluetooth Mesh(SIG) Subpieces WiFi+Bluetooth [✓ Wi-Fi] Zigbee

Ok

Figure 2: 微信截图 _20200212182741.png

Choose the function we need



Select common function

You can still add custom functions after adding common functions

Select all

[✓ Switch 1] Switch 2 Switch 3 Switch 4

Switch 5 Switch 6 Switch 7 Switch 8

Countdown 1 Countdown 2 Countdown 3

Figure 3: 微信截图 _20200212182808.png

3.1.2 Download MCU SDK

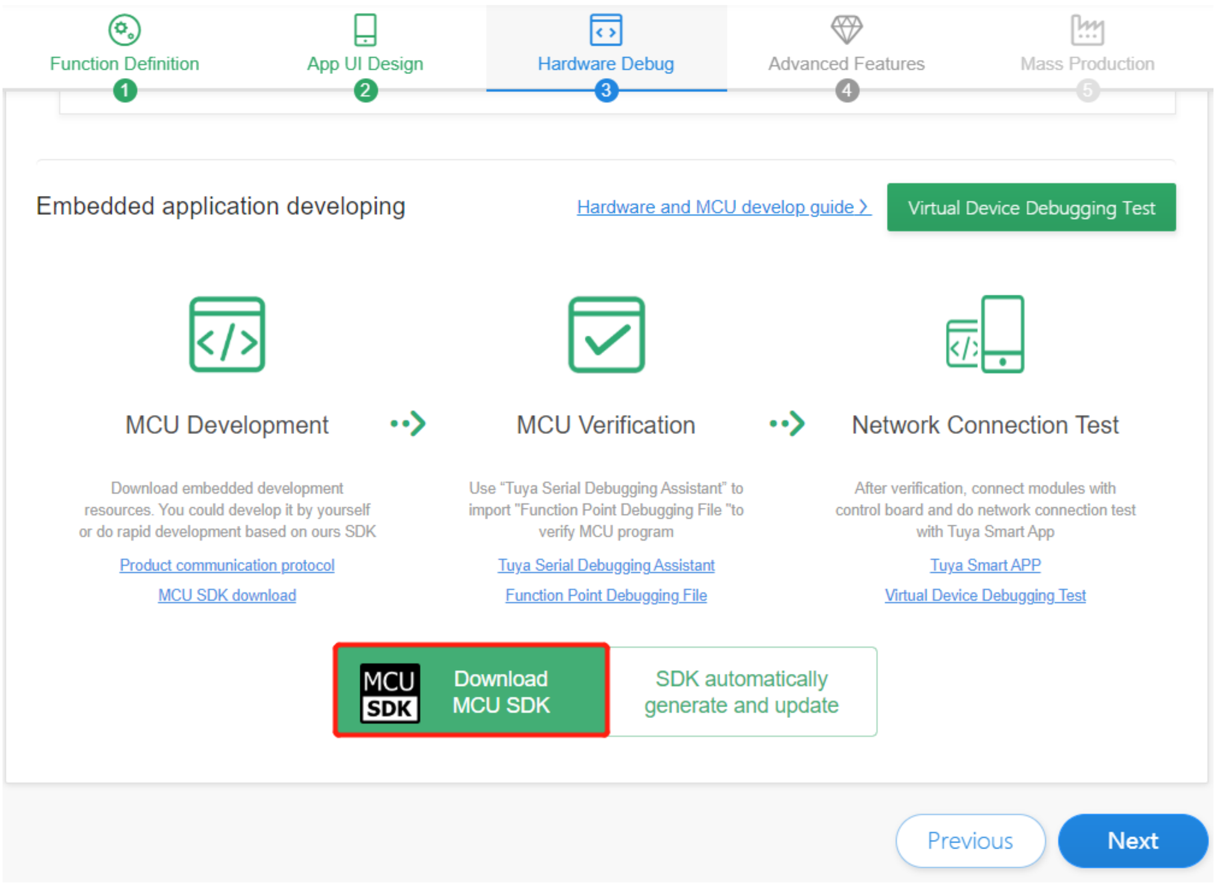
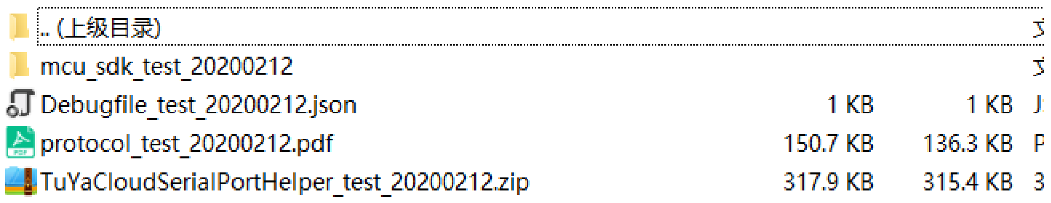

Function Definition
1
App UI Design
2
Hardware Debug
3
Advanced Features
4
Mass Production
5
Embedded application developing
Hardware and MCU develop guide >
Virtual Device Debugging Test
MCU Development
MCU Verification
Network Connection Test
Download embedded development resources. You could develop it by yourself or do rapid development based on ours SDK
Product communication protocol
MCU SDK download
Use "Tuya Serial Debugging Assistant" to import "Function Point Debugging File "to verify MCU program
Tuya Serial Debugging Assistant
Function Point Debugging File
After verification, connect modules with control board and do network connection test with Tuya Smart App
Tuya Smart APP
Virtual Device Debugging Test
MCU SDK
Download MCU SDK
SDK automatically generate and update
Previous
Next

Figure 4: 微信截图 _20200212182924.png

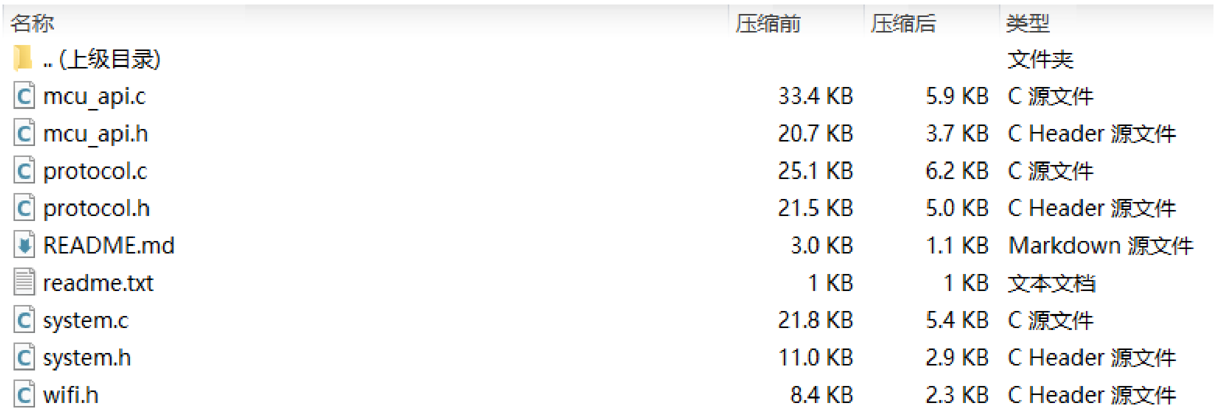
Downloaded file's directory is as follows:



.. (上级目录)			文
mcu_sdk_test_20200212			文
Debugfile_test_20200212.json	1 KB	1 KB	J
protocol_test_20200212.pdf	150.7 KB	136.3 KB	P
TuYaCloudSerialPortHelper_test_20200212.zip	317.9 KB	315.4 KB	3

Figure 5: 企业微信截图 _15816757319222.png

It includes a protocol file, Tuya serial port helper and its debugfile, and MCU SDK we need here:



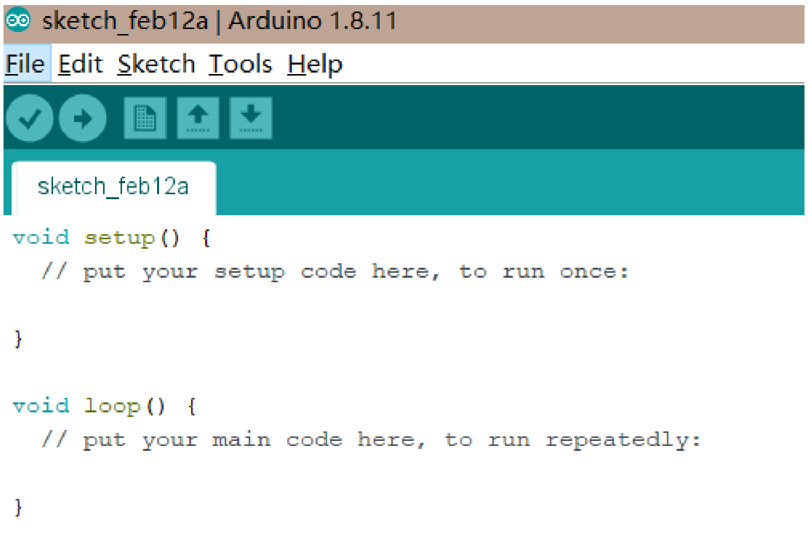
名称	压缩前	压缩后	类型
.. (上级目录)			文件夹
mcu_api.c	33.4 KB	5.9 KB	C 源文件
mcu_api.h	20.7 KB	3.7 KB	C Header 源文件
protocol.c	25.1 KB	6.2 KB	C 源文件
protocol.h	21.5 KB	5.0 KB	C Header 源文件
README.md	3.0 KB	1.1 KB	Markdown 源文件
readme.txt	1 KB	1 KB	文本文档
system.c	21.8 KB	5.4 KB	C 源文件
system.h	11.0 KB	2.9 KB	C Header 源文件
wifi.h	8.4 KB	2.3 KB	C Header 源文件

Figure 6: 4f327b19dbcff9c05ae6f2727b16600.png

When we perform the above operations, we have successfully obtained the SDK development kit. Next, we need to port the SDK development kit to our project.

3.2 MCU SDK package porting

Open Arduino IDE and save the new project.



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

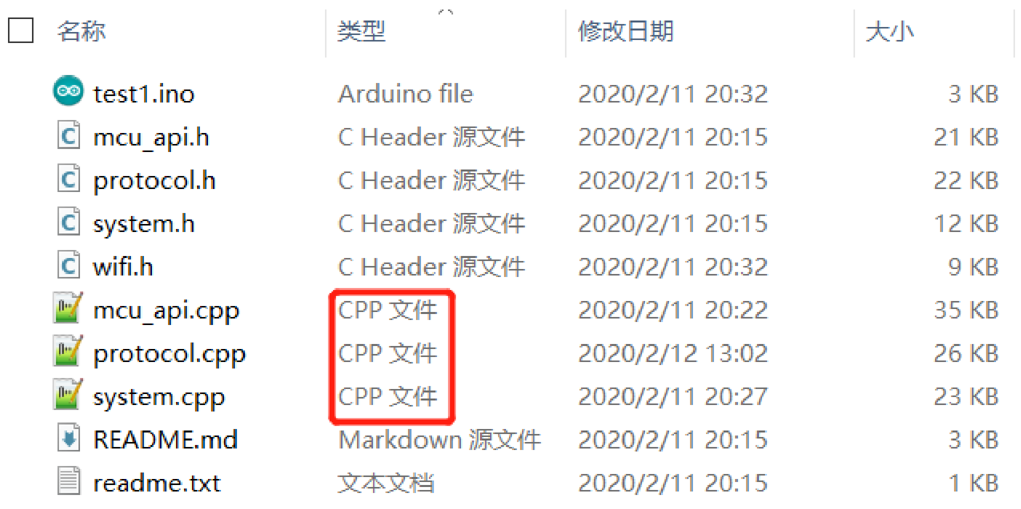
Figure 7: 微信截图 _20200212184556.png

At this time, there are two functions in our project:

`setup ()` is generally used for initialization and executed only once.

`loop ()` is executed in a loop.

All the following `.C` and `.H` files in the MCU development kit are copied to the newly created one. The same level directory under the sandwich project path, and change the suffix `.C` to `.CPP` . As shown below:



名称	类型	修改日期	大小
test1.ino	Arduino file	2020/2/11 20:32	3 KB
mcu_api.h	C Header 源文件	2020/2/11 20:15	21 KB
protocol.h	C Header 源文件	2020/2/11 20:15	22 KB
system.h	C Header 源文件	2020/2/11 20:15	12 KB
wifi.h	C Header 源文件	2020/2/11 20:32	9 KB
mcu_api.cpp	CPP 文件	2020/2/11 20:22	35 KB
protocol.cpp	CPP 文件	2020/2/12 13:02	26 KB
system.cpp	CPP 文件	2020/2/11 20:27	23 KB
README.md	Markdown 源文件	2020/2/11 20:15	3 KB
readme.txt	文本文档	2020/2/11 20:15	1 KB

Figure 8: cf39370893ce6ae3dcaf6fb6a8d8439.png

Close the Arduino IDE, click on the extension ending with `Yourprojectname.ino` and reopen to see the added files:

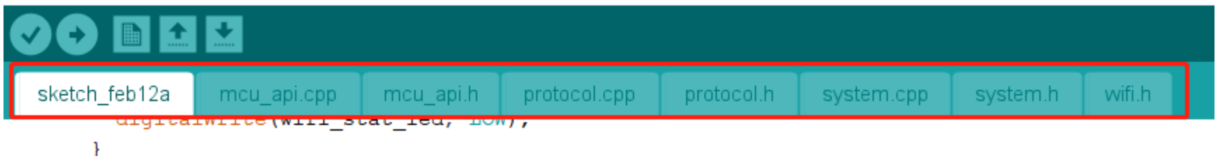


Figure 9: 企业微信截图 _15816734077680.png

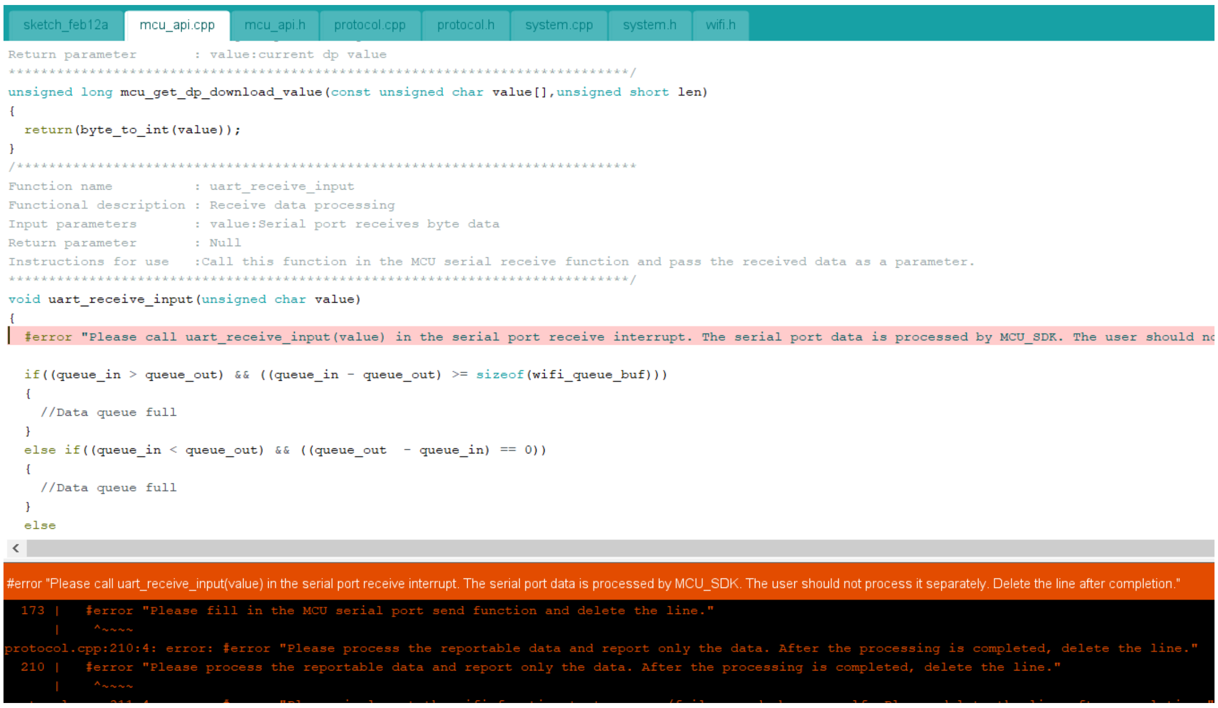
Click the check mark in the upper left corner to verify.

3.2.1 Fix error

After the first verification, an error message will appear. Follow the prompts to make corrections, and click Verify until there are no errors.

Possible errors are as follows:

Error 1:



```
Return parameter        : value:current dp value
*****************************************************************************/
unsigned long mcu_get_dp_download_value(const unsigned char value[],unsigned short len)
{
  return(byte_to_int(value));
}
/*****************************************************************************
Function name           : uart_receive_input
Functional description : Receive data processing
Input parameters        : value:Serial port receives byte data
Return parameter        : Null
Instructions for use    :Call this function in the MCU serial receive function and pass the received data as a parameter.
*****************************************************************************/
void uart_receive_input(unsigned char value)
{
  #error "Please call uart_receive_input(value) in the serial port receive interrupt. The serial port data is processed by MCU_SDK. The user should no

  if((queue_in > queue_out) && ((queue_in - queue_out) >= sizeof(wifi_queue_buf)))
  {
    //Data queue full
  }
  else if((queue_in < queue_out) && ((queue_out  - queue_in) == 0))
  {
    //Data queue full
  }
  else
```

```
#error "Please call uart_receive_input(value) in the serial port receive interrupt. The serial port data is processed by MCU_SDK. The user should not process it separately. Delete the line after completion."
```

```
  173 |   #error "Please fill in the MCU serial port send function and delete the line."
      |    ^~~~~
protocol.cpp:210:4: error: #error "Please process the reportable data and report only the data. After the processing is completed, delete the line."
  210 |   #error "Please process the reportable data and report only the data. After the processing is completed, delete the line."
      |    ^~~~~
```

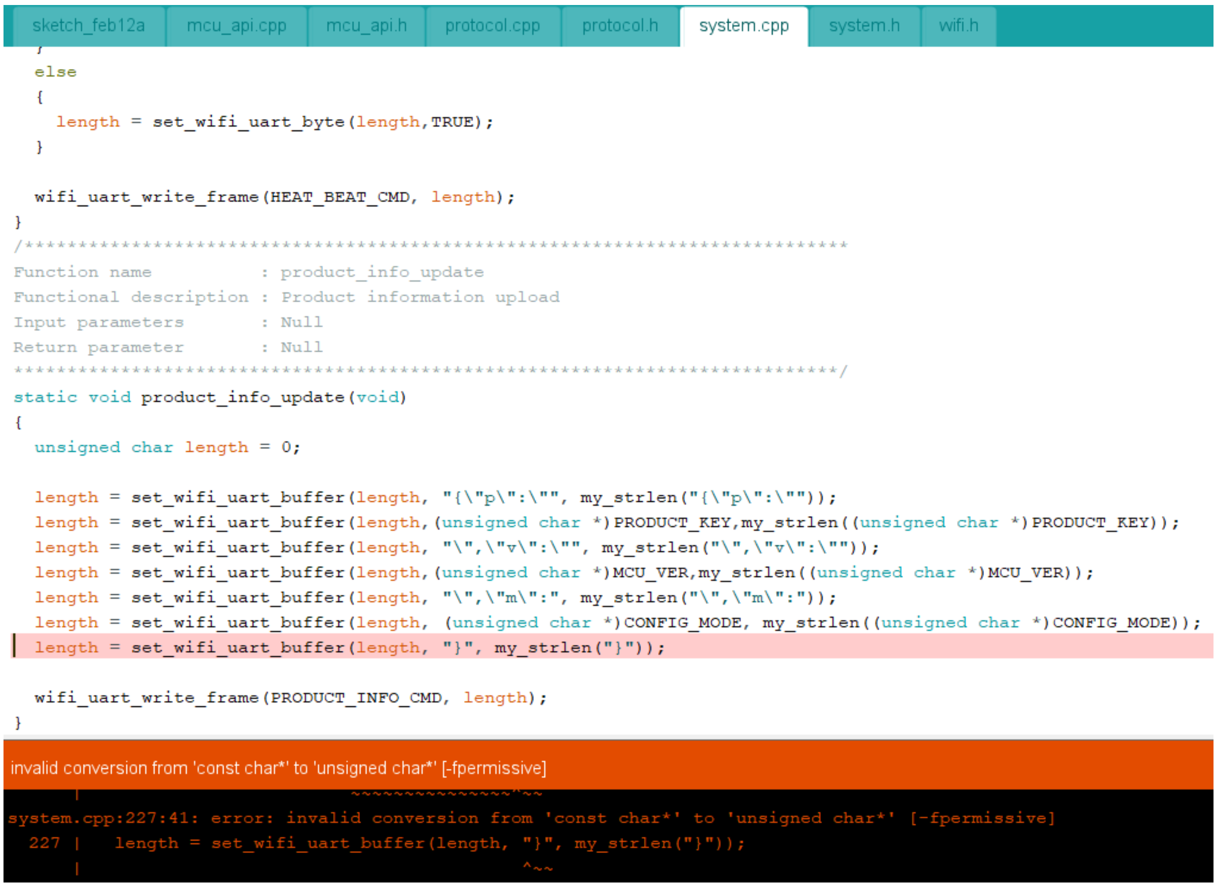
Figure 10: 企业微信截图 _15816693633170.png

Reason: Every #error in SDK is for reminding developer to edit the necessary parts of code.

Correction method: Add // before #error to make it in a comment state temporarily, click Verify, if successful, the next #error will be displayed.

Note: After verify successfully , please back to edit these necessary functions.

Error 2:



sketch_feb12a | mcu_api.cpp | mcu_api.h | protocol.cpp | protocol.h | system.cpp | system.h | wifi.h

```
  else
  {
    length = set_wifi_uart_byte(length,TRUE);
  }

  wifi_uart_write_frame(HEAT_BEAT_CMD, length);
}
/*****************************************************************************
Function name          : product_info_update
Functional description : Product information upload
Input parameters       : Null
Return parameter       : Null
*****************************************************************************/
static void product_info_update(void)
{
  unsigned char length = 0;

  length = set_wifi_uart_buffer(length, "{\"p\":\"", my_strlen("{\"p\":\""));
  length = set_wifi_uart_buffer(length,(unsigned char *)PRODUCT_KEY,my_strlen((unsigned char *)PRODUCT_KEY));
  length = set_wifi_uart_buffer(length, "\",\"v\":\"", my_strlen("\",\"v\":\""));
  length = set_wifi_uart_buffer(length,(unsigned char *)MCU_VER,my_strlen((unsigned char *)MCU_VER));
  length = set_wifi_uart_buffer(length, "\",\"m\":", my_strlen("\",\"m\":"));
  length = set_wifi_uart_buffer(length, (unsigned char *)CONFIG_MODE, my_strlen((unsigned char *)CONFIG_MODE));
  length = set_wifi_uart_buffer(length, "}", my_strlen("}"));

  wifi_uart_write_frame(PRODUCT_INFO_CMD, length);
}
```

invalid conversion from 'const char*' to 'unsigned char*' [-fpermissive]

```
system.cpp:227:41: error: invalid conversion from 'const char*' to 'unsigned char*' [-fpermissive]
  227 |   length = set_wifi_uart_buffer(length, "}", my_strlen("}"));
      |                                         ^~~
```

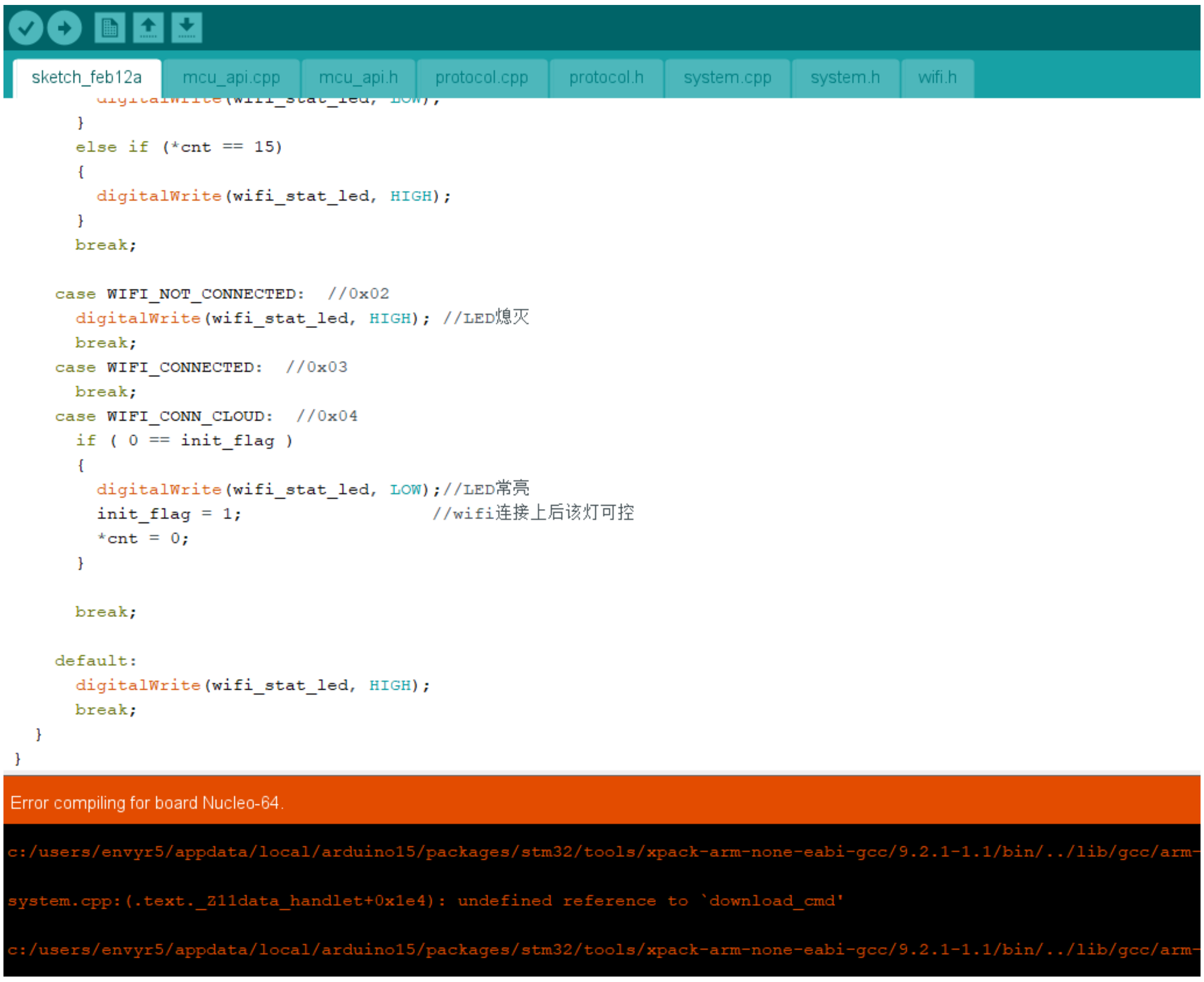
Figure 11: 企业微信截图 _15816700212641.png

Reason: Compilation environment.

Correction method: If the compilation fails, a cast is required. You can find it in `system.cpp` file, the code reference is as follows:

```c
static void product_info_update(void)
{
  unsigned char length = 0;
  length = set_wifi_uart_buffer(length,(unsigned char *)"{\"p\":\"",
     my_strlen((unsigned char *)"{\"p\":\""));
  length = set_wifi_uart_buffer(length,(unsigned char *)PRODUCT_KEY,
     my_strlen((unsigned char *)PRODUCT_KEY));
  length = set_wifi_uart_buffer(length,(unsigned char *) "\",\"v\":\"",
      my_strlen((unsigned char *)"\",\"v\":\""));
  length = set_wifi_uart_buffer(length,(unsigned char *)MCU_VER,
     my_strlen((unsigned char *)MCU_VER));
  length = set_wifi_uart_buffer(length,(unsigned char *)"\",\"m\":",
     my_strlen((unsigned char *)"\",\"m\":"));
  length = set_wifi_uart_buffer(length,(unsigned char *)CONFIG_MODE,
     my_strlen((unsigned char *)CONFIG_MODE));
  length = set_wifi_uart_buffer(length,(unsigned char *) "}", my_strlen
     ((unsigned char *)"}"));

  wifi_uart_write_frame(PRODUCT_INFO_CMD, length);
}
```

Error 3:



```
          digitalWrite(wifi_stat_led, LOW);
        }
        else if (*cnt == 15)
        {
          digitalWrite(wifi_stat_led, HIGH);
        }
        break;

      case WIFI_NOT_CONNECTED:   //0x02
        digitalWrite(wifi_stat_led, HIGH); //LED熄灭
        break;
      case WIFI_CONNECTED:   //0x03
        break;
      case WIFI_CONN_CLOUD:   //0x04
        if ( 0 == init_flag )
        {
          digitalWrite(wifi_stat_led, LOW);//LED常亮
          init_flag = 1;                  //wifi连接上后该灯可控
          *cnt = 0;
        }

        break;

      default:
        digitalWrite(wifi_stat_led, HIGH);
        break;
    }
  }
```

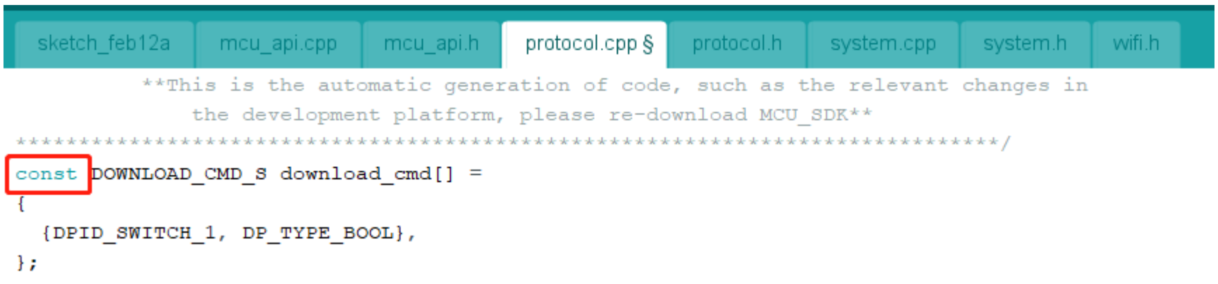
Error compiling for board Nucleo-64.

```
c:/users/envyr5/appdata/local/arduino15/packages/stm32/tools/xpack-arm-none-eabi-gcc/9.2.1-1.1/bin/../lib/gcc/arm-

system.cpp:(.text._Z11data_handlet+0x1e4): undefined reference to `download_cmd'

c:/users/envyr5/appdata/local/arduino15/packages/stm32/tools/xpack-arm-none-eabi-gcc/9.2.1-1.1/bin/../lib/gcc/arm-
```

Figure 12: 企业微信截图 _15816715329106.png

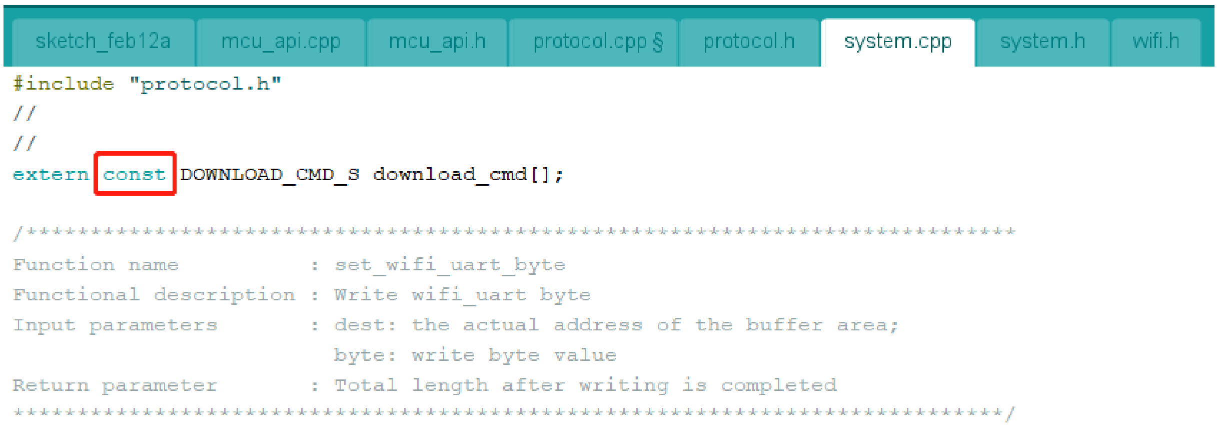
Reason: Compilation environment.

Modification method: Remove followed `const` at the screenshots in the `protocol.cpp` and `system.cpp` files.



```
           **This is the automatic generation of code, such as the relevant changes in
               the development platform, please re-download MCU_SDK**
******************************************************************************/
const DOWNLOAD_CMD_S download_cmd[] =
{
  {DPID_SWITCH_1, DP_TYPE_BOOL},
};
```

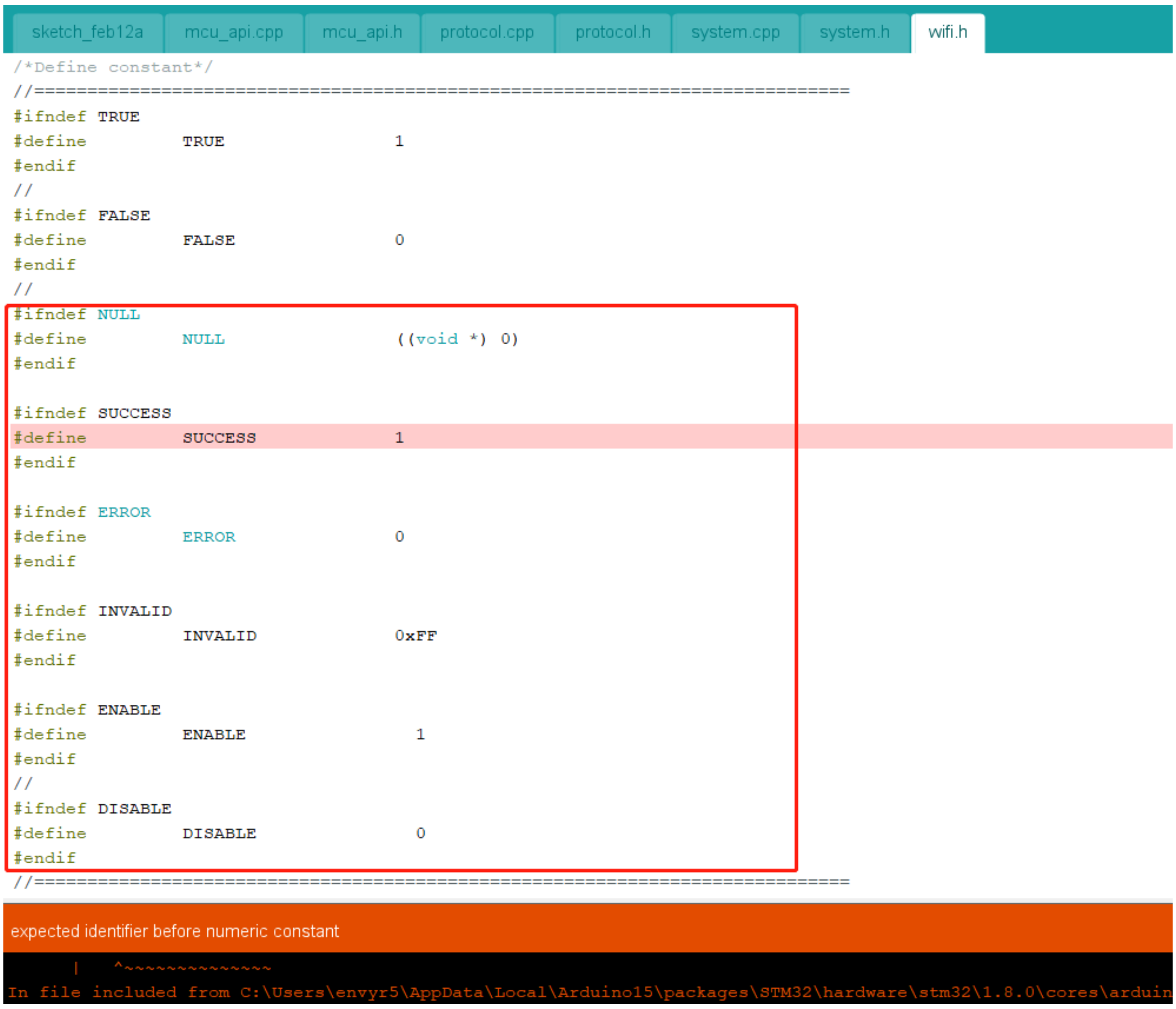
Figure 13: 企业微信截图 _15816717844034.png



```
#include "protocol.h"
//
//
extern const DOWNLOAD_CMD_S download_cmd[];

/*****************************************************************************
Function name          : set_wifi_uart_byte
Functional description : Write wifi_uart byte
Input parameters       : dest: the actual address of the buffer area;
                         byte: write byte value
Return parameter       : Total length after writing is completed
*****************************************************************************/
```

Figure 14: 企业微信截图 _15816718107043.png

Error 4:



```
/*Define constant*/
//==============================================================================
#ifndef TRUE
#define         TRUE                1
#endif
//
#ifndef FALSE
#define         FALSE               0
#endif
//
#ifndef NULL
#define         NULL                ((void *) 0)
#endif

#ifndef SUCCESS
#define         SUCCESS             1
#endif

#ifndef ERROR
#define         ERROR               0
#endif

#ifndef INVALID
#define         INVALID             0xFF
#endif

#ifndef ENABLE
#define         ENABLE                1
#endif
//
#ifndef DISABLE
#define         DISABLE               0
#endif
//==============================================================================
```

```
expected identifier before numeric constant
      |    ^~~~~~~~~~~~~~~
In file included from C:\Users\envyr5\AppData\Local\Arduino15\packages\STM32\hardware\stm32\1.8.0\cores\arduin
```

Figure 15: 企业微信截图 _1581672504653.png

Reason: Repeated definition.

Correction method: To avoid this errors, remove the red box in the figure above.

After the error is resolved, compile and pass. We need to implement the serial port sending and receiving functions between the sandwich development board and the Wi-Fi module communication. For other porting details, please refer to MCU SDK Porting.

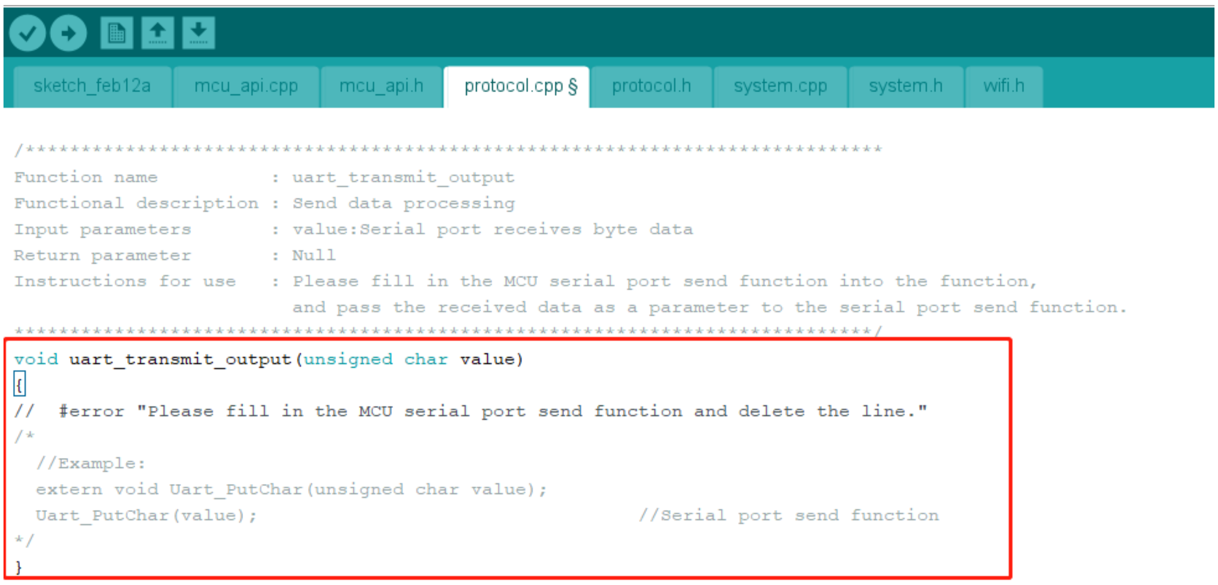
3.2.2 Function implementation of serial port

The serial port of the sandwich development board uses the Arduino serial port API. For instructions on using the Arduino API, developers can check on the Arduino website.

Refer to the following code to implement the serial port receiving function in the main file:

```
#include "wifi.h"
#include <SoftwareSerial.h>

SoftwareSerial mySerial(0, 1); // RX, TX
#define _SS_MAX_RX_BUFF 300
#define relay 10
int time_cnt = 0, cnt = 0, init_flag = 0;



void setup() {

  pinMode(relay, OUTPUT);   //init the output IO
  digitalWrite(relay, LOW);

  pinMode(PC13, INPUT);     //reset the button of wifi configration
  pinMode(8, OUTPUT);       //Wi-Fi configration light

  mySerial.begin(9600);     //init the serial port
  mySerial.println("myserial init successful!");
  Serial.begin(115200);     //PA3 RX   PA2 TX
  Serial.println("serial init successful!");

  wifi_protocol_init();
}

void loop() {
  if (init_flag == 0) {
    time_cnt++;
    if (time_cnt % 6000 == 0) {
      time_cnt = 0;
      cnt ++;
    }
    wifi_stat_led(&cnt);   //Wi-Fi statues process
  }
  wifi_uart_service();
  myserialEvent();      //receiving data
  key_scan();           //Wi-Fi reset button scan


}


void myserialEvent() {
  if (mySerial.available()) {
    unsigned char ch = (unsigned char)mySerial.read();
    uart_receive_input(ch);
  }
}

void key_scan(void)
{
  static char ap_ez_change = 0;
  unsigned char buttonState  = HIGH;
  buttonState = digitalRead(PC13);
  if (buttonState == LOW) {
    delay(3000);
    buttonState = digitalRead(PC13);
      printf("------%d",buttonState);
    if (buttonState == LOW) {
```

Transmit function rewrite:



```
/*****************************************************************************
Function name          : uart_transmit_output
Functional description : Send data processing
Input parameters       : value:Serial port receives byte data
Return parameter       : Null
Instructions for use   : Please fill in the MCU serial port send function into the function,
                         and pass the received data as a parameter to the serial port send function.
*****************************************************************************/
void uart_transmit_output(unsigned char value)
{
//  #error "Please fill in the MCU serial port send function and delete the line."
/*
  //Example:
  extern void Uart_PutChar(unsigned char value);
  Uart_PutChar(value);                                  //Serial port send function
*/
}
```

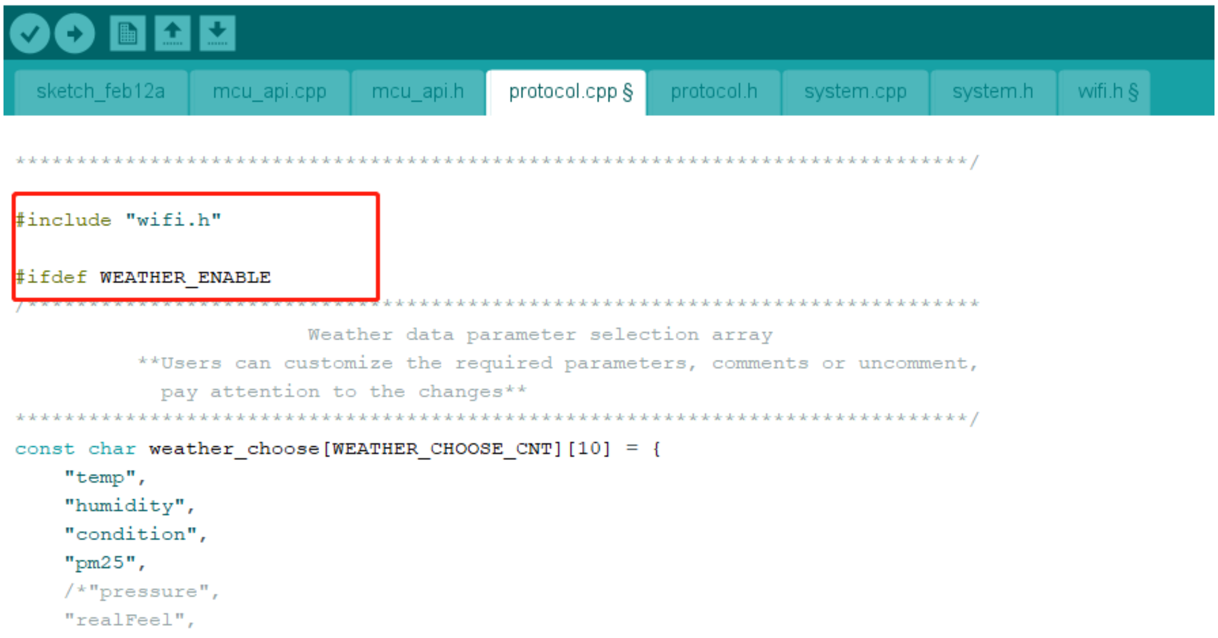
Figure 16: 企业微信截图 _1581672886471.png

Change to:

```
void uart_transmit_output(unsigned char value)
{
//  #error "Please fill in the MCU serial port send function and delete
     the line."
     extern SoftwareSerial mySerial;
     mySerial.write(value);
}
```

Import Arduino corresponding library function header files:

- Change the screenshot part from `protocol.cpp` file



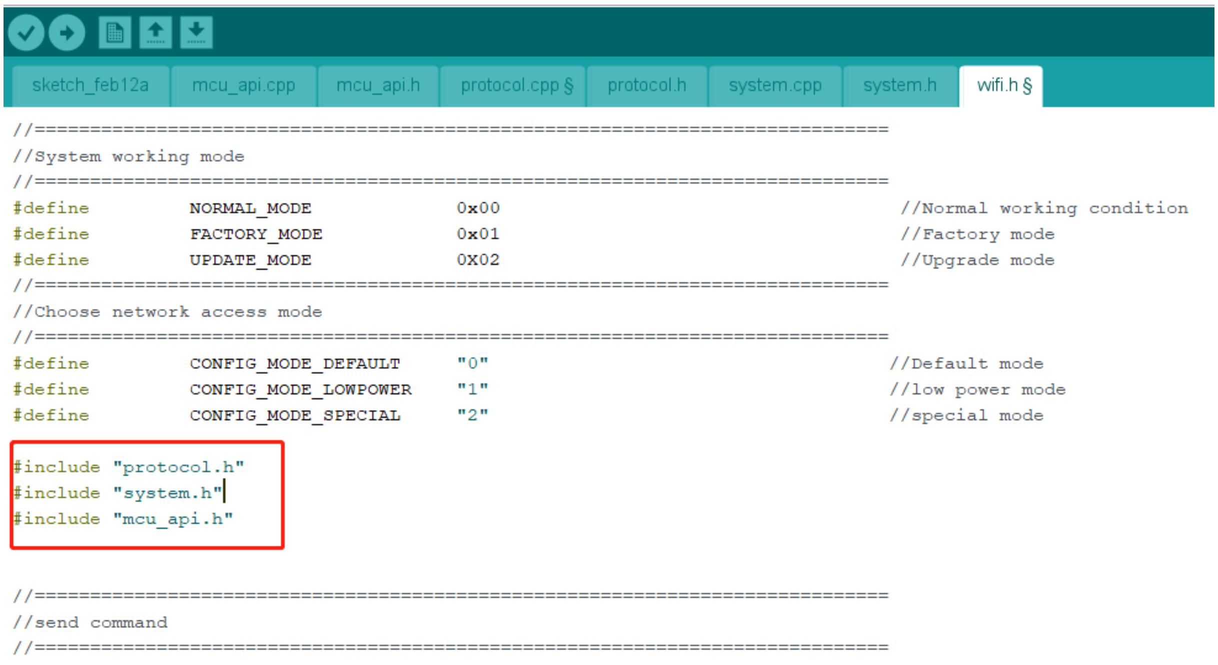
```
*****************************************************************************/

#include "wifi.h"

#ifdef WEATHER_ENABLE
/******************************************************************************
                        Weather data parameter selection array
          **Users can customize the required parameters, comments or uncomment,
            pay attention to the changes**
******************************************************************************/
const char weather_choose[WEATHER_CHOOSE_CNT][10] = {
    "temp",
    "humidity",
    "condition",
    "pm25",
    /*"pressure",
    "realFeel",
```

Figure 17: 企业微信截图 _15816721221421.png

To:

```
#include "wifi.h"
#include <SoftwareSerial.h>
#include "Arduino.h"
#ifdef WEATHER_ENABLE
```

- The screenshot part from the `wifi.h` file



```
//=============================================================================
//System working mode
//=============================================================================
#define         NORMAL_MODE             0x00                                    //Normal working condition
#define         FACTORY_MODE            0x01                                    //Factory mode
#define         UPDATE_MODE             0X02                                    //Upgrade mode
//=============================================================================
//Choose network access mode
//=============================================================================
#define         CONFIG_MODE_DEFAULT     "0"                                    //Default mode
#define         CONFIG_MODE_LOWPOWER    "1"                                    //low power mode
#define         CONFIG_MODE_SPECIAL     "2"                                    //special mode

#include "protocol.h"
#include "system.h"
#include "mcu_api.h"

//=============================================================================
//send command
//=============================================================================
```

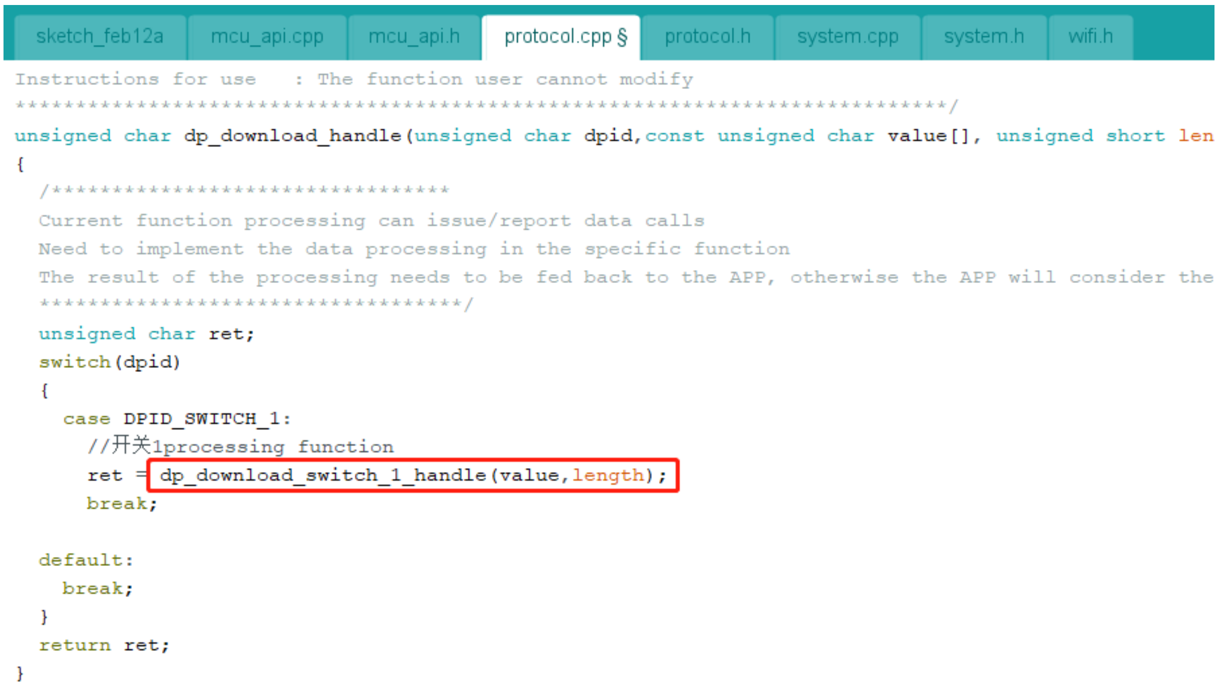
Figure 18: 企业微信截图 _15816721492245.png

Add a line below:

```
#include "Arduino.h"
```

3.3 Product function implementation

After the serial port sending and receiving functions are implemented, developers need to implement the specific functions of the product. We generated the corresponding DP function functions from the SDK package downloaded by Tuya IoT platform. When creating the product, we selected the switch function, and the specific function was implemented in the `protocol. c` function.



```
Instructions for use   : The function user cannot modify
*****************************************************************************/
unsigned char dp_download_handle(unsigned char dpid,const unsigned char value[], unsigned short len
{
  /*********************************
  Current function processing can issue/report data calls
  Need to implement the data processing in the specific function
  The result of the processing needs to be fed back to the APP, otherwise the APP will consider the
  ************************************/
  unsigned char ret;
  switch(dpid)
  {
    case DPID_SWITCH_1:
      //开关1processing function
      ret = dp_download_switch_1_handle(value,length);
      break;

  default:
    break;
  }
  return ret;
}
```

Figure 19: 企业微信截图 _15816730201595.png

For serial port related transplantation, please refer to 《MCU SDK Migration》 For detailed implementation details of the communication protocol, please refer to Tuya Cloud Universal Serial Port Access Protocol.

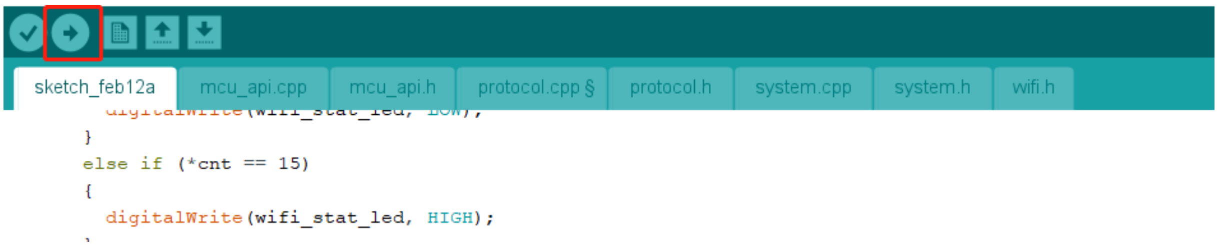
4 Download and debug

4.1 Download

After the developer has written the application for the product, the following steps can be used to download the code to the Tuya Sandwich Development Board.

1、Connect the sandwich development board to the computer and select the corresponding port;

2、Click Upload.

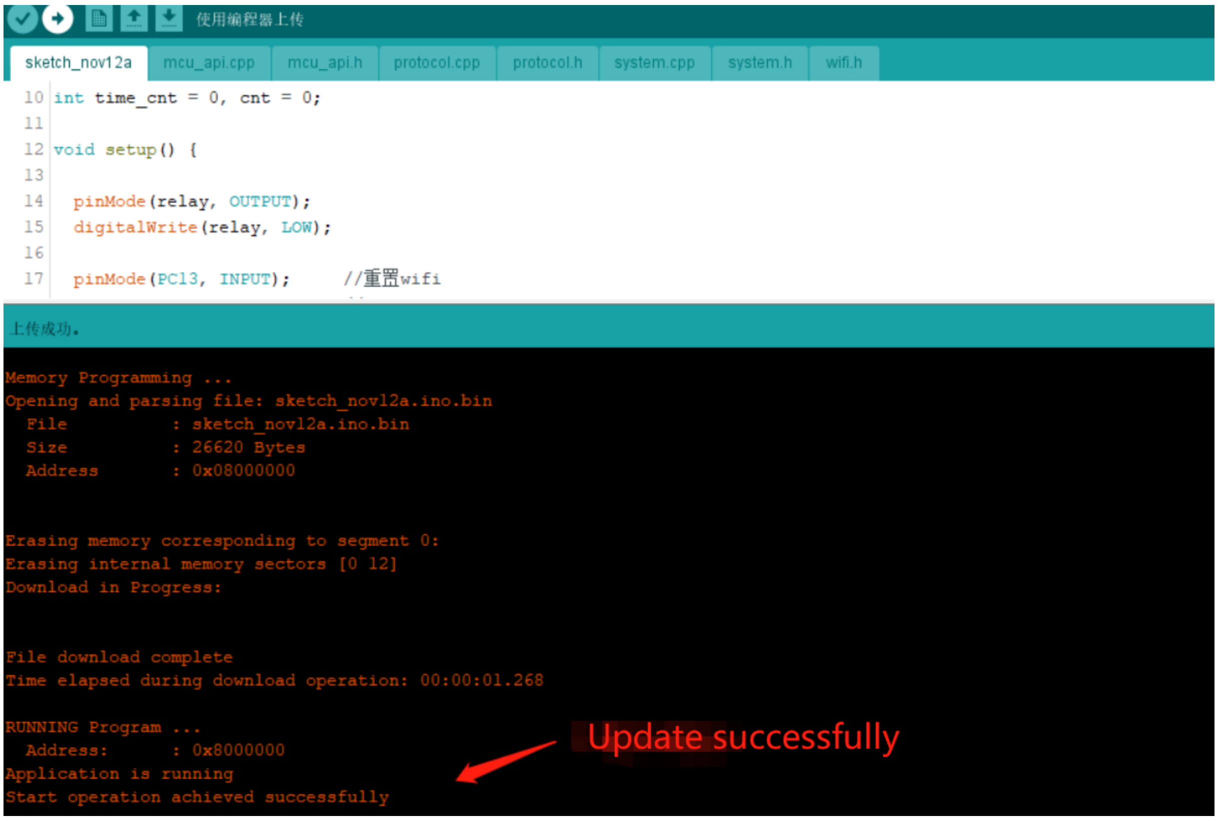


```
    digitalWrite(wifi_stat_led, LOW);
  }
  else if (*cnt == 15)
  {
    digitalWrite(wifi_stat_led, HIGH);
```

Figure 20: 企业微信截图 _15816730933277.png

5 Results display

After downloading the program to our development board through the above steps, the following page appears after downloading successfully:



```
int time_cnt = 0, cnt = 0;

void setup() {

  pinMode(relay, OUTPUT);
  digitalWrite(relay, LOW);

  pinMode(PC13, INPUT);      //重置wifi
```

上传成功。

```
Memory Programming ...
Opening and parsing file: sketch_nov12a.ino.bin
  File          : sketch_nov12a.ino.bin
  Size          : 26620 Bytes
  Address       : 0x08000000


Erasing memory corresponding to segment 0:
Erasing internal memory sectors [0 12]
Download in Progress:


File download complete
Time elapsed during download operation: 00:00:01.268

RUNNING Program ...
  Address:      : 0x8000000
Application is running
Start operation achieved successfully
```

Figure 21: 企业微信截图 _15816749444711.png

5.1 Debug

The graffiti sandwich development board debugging can use the printf function to format and output the information we need to debug

1、Initialize the hardware serial port `Serial.begin (115200);` ;

2、Since the `printf ()` function has been redirected to our serial port in the Tuya Sandwich Development Board library, you can use the `printf ()` function directly.

6 Distribution

Download the program to the development board and power it on again. At this time, the Wi-Fi module is in EZ network configuration mode by default. Use Tuya Smart App for network configuration.

Network indicator:

Status of light	Network status
Fast flashing	EZ network configuration status, waiting for network configuration
Slow flashing	AP network configuration status, waiting for network configuration
Off	WIFI is configured and connected to the router
Always lit	Connected to the router and connected to the cloud

Networking steps:

1、Open Tuya Smart, click icon;

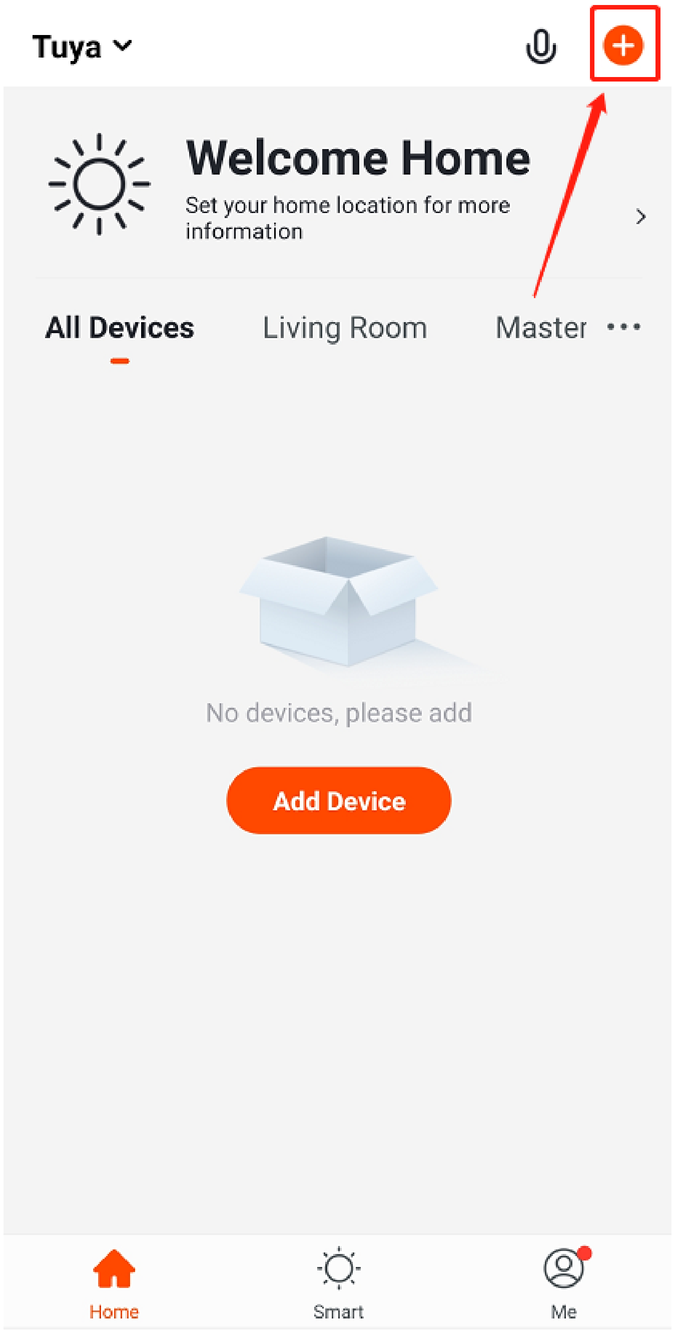


Figure 22: 企业微信截图 _15816779813590.png

2、Add device;

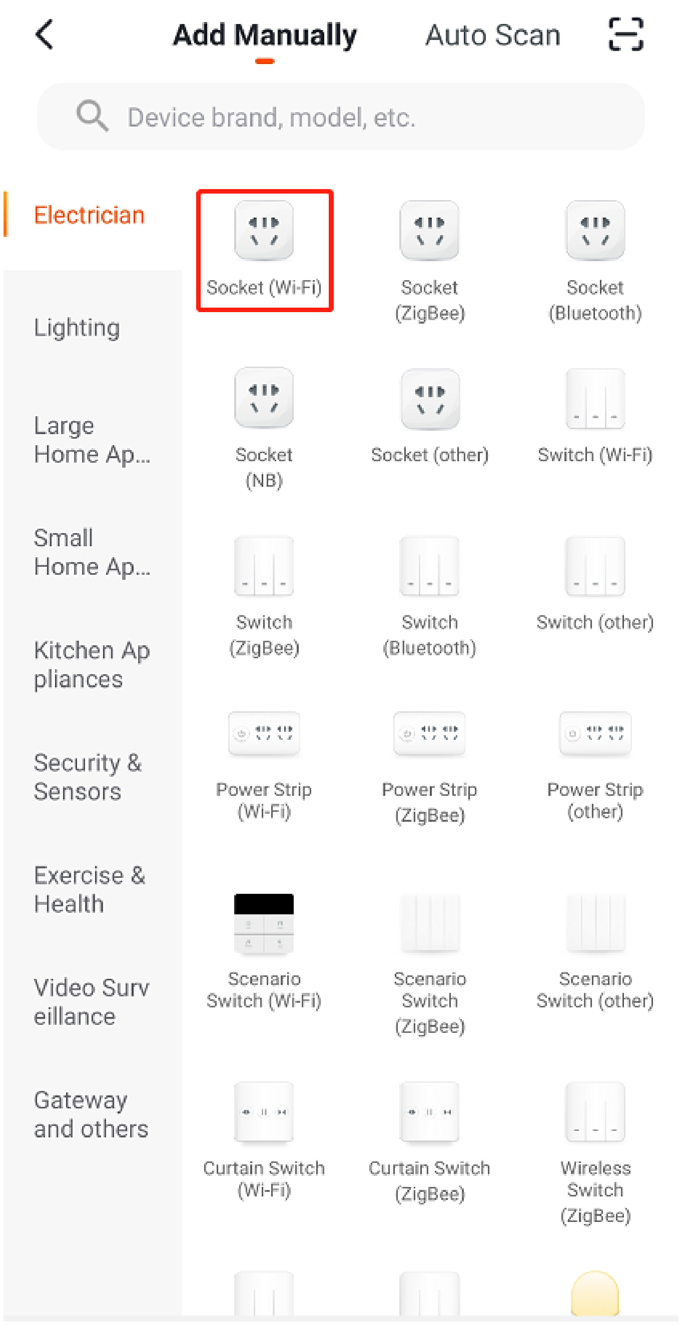
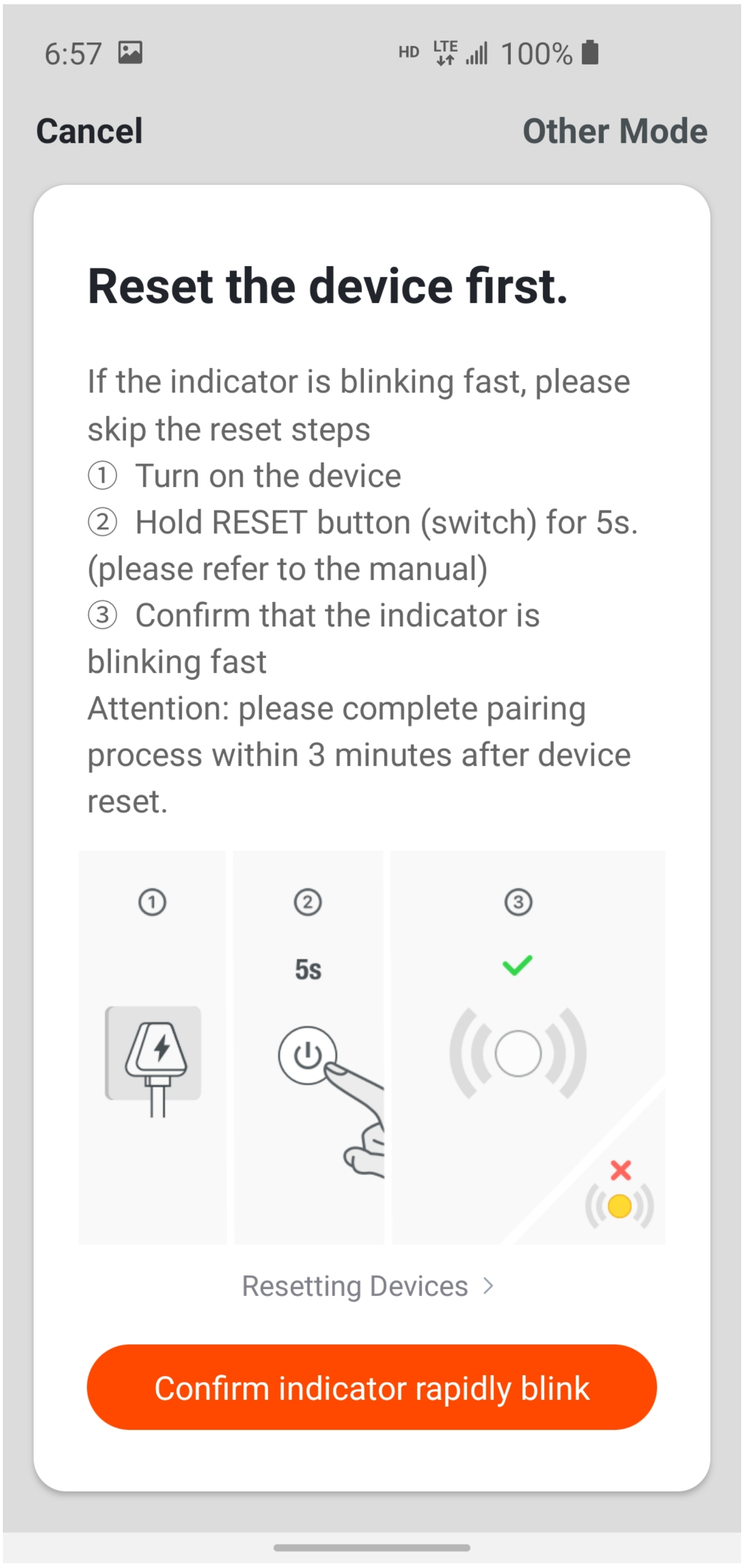


Figure 23: 企业微信截图 _15816780064780.png



6:57 HD LTE 100%

Cancel Other Mode

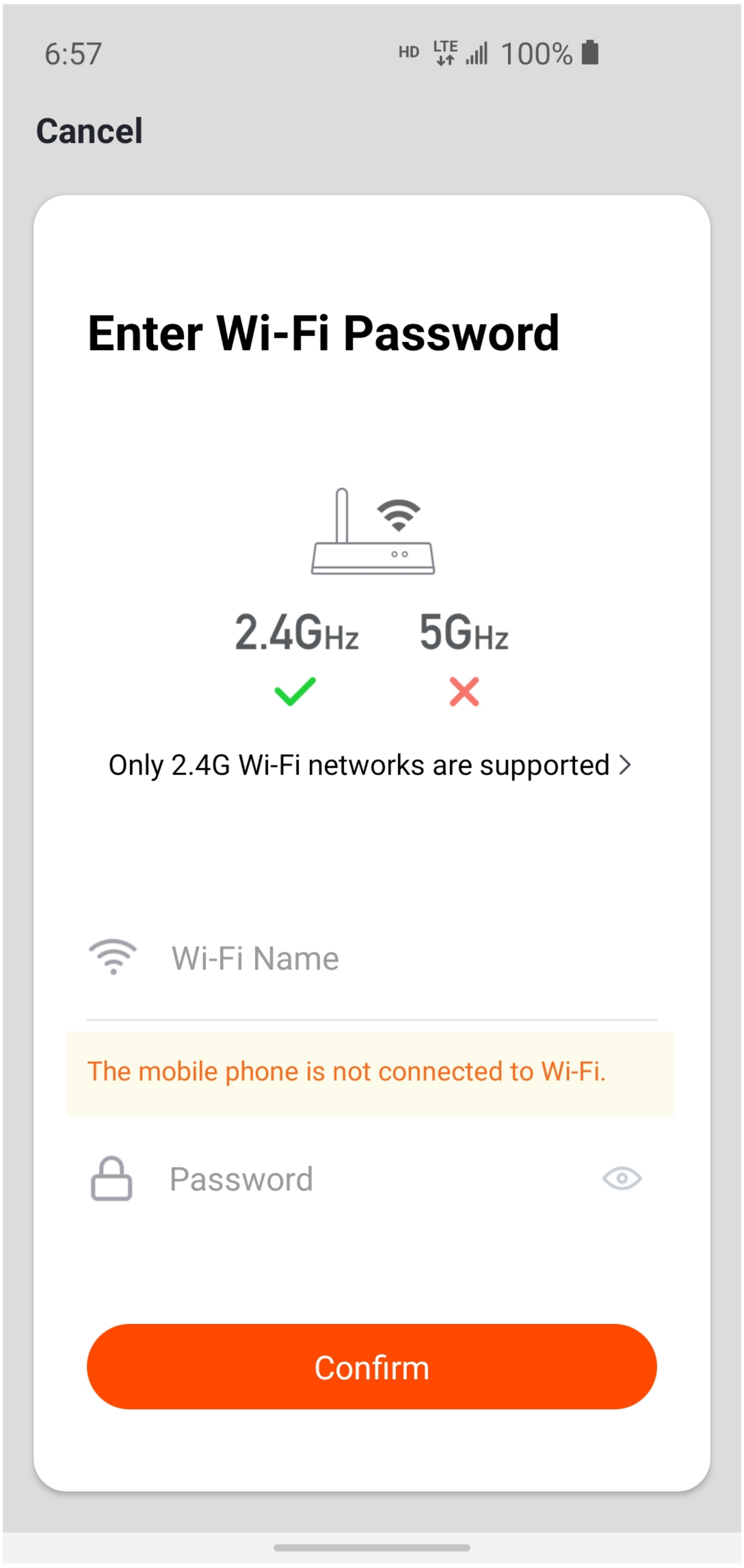
Reset the device first.

If the indicator is blinking fast, please skip the reset steps
① Turn on the device
② Hold RESET button (switch) for 5s. (please refer to the manual)
③ Confirm that the indicator is blinking fast
Attention: please complete pairing process within 3 minutes after device reset.

Resetting Devices >

Confirm indicator rapidly blink

Figure 24: Screenshot_20200214185728_TuyaSmart1.jpg



6:57 HD LTE 100%

Cancel

Enter Wi-Fi Password

Only 2.4G Wi-Fi networks are supported >

Wi-Fi Name

The mobile phone is not connected to Wi-Fi.

Password

Confirm

Figure 25: Screenshot_20200214185752_TuyaSmart.jpg

3、Wait for the network to complete;

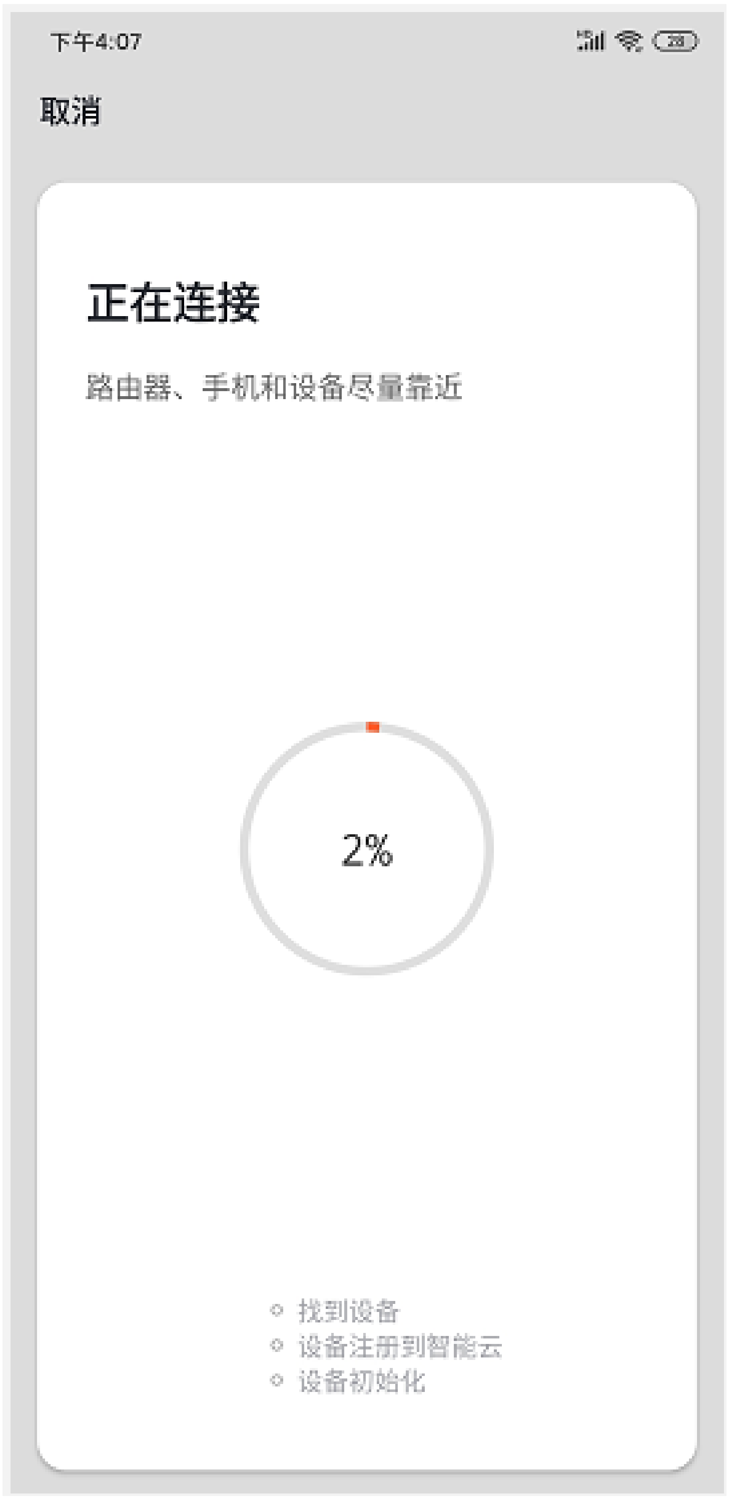
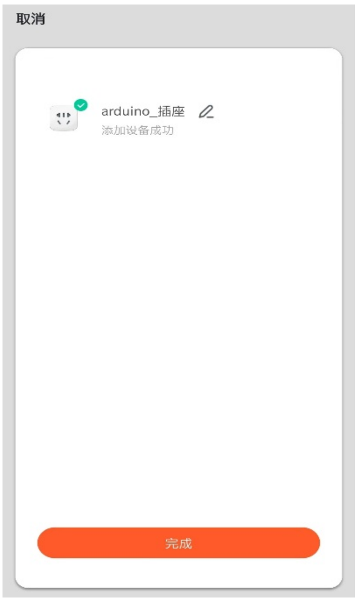


Figure 26: image.png



4、After the equipment network is successfully configured, we can control it through APP.

6.1 References

Tuya technical glossary See Explanation of Terminology.

Tuya serial port access protocol See Tuya Cloud Universal Serial Port Access Protocol.

MCU SDK migration on STM32 board SeeMCU SDK Migration.