# IoT

# 1硬件ESP8266介绍

ESP8266芯片是一款串口转无线模芯片，内部自带固件，用户操作简单，无需编写时序信号等。本次实验我用的是ESP8266-12F系列。这款芯片使用了3.3V的直流电源，体积小，功耗低，支持透传，丢包现象不严重，价格低。ESP8266还允许用户自己编写rom，不仅可以实现数据传输功能，还可控制建立wifi热点，或者作为wifi客户端连接到某指定路由器，同时还可编程控制所以的gpio。

【图1】

# 2配置Arduino环境

## 2.1首选项设置

打开arduino，点击“文件”-“首选项”，更换开发板管理器地址：<http://arduino.esp8266.com/stable/package_esp8266com_index.json>

【图2】

## 2.2添加ESP8266模组支持

点击“工具”-“开发板：Arduino UNO”-“开发板管理器，搜索“esp8266”，找到对应的并点击下载安装。

【图3】

【图4】

## 2.3添加相关依赖库

点击“项目”-“加载库”-“管理库界面”

【图5】

搜索“PubSubClient”，找到“EspMQTTclient”进行安装

【图6】

## 2.4修改PubSubClient源文件

通过“首选项”找到arduino的“library”的位置

【图7】

打开PubSubClient.h文件

【图8】

配置修改如图

【图9】

## 2.5ArduinoJson & U8g2lib & DHT sensor library的安装

在ArduinoJson

【图10】

安装U8g2

【图11】

安装DHT sensor library

【图12】

# 3配置阿里云

在阿里云搜索“物联网平台”并选择，点击“管理控制台”，进入控制台

【图13】【图14】

在控制台中点击“公共实例”

【图15】

在左侧里面有个“设备管理”，点击“产品”，再点击“创建产品”

【图16】

点击该项目，进入项目并在项目中新增一个产品，并成功创建，获得三元信息

【图17】

创建一个设备“pre-test”

【图18】

“编辑草稿”-“添加自定义功能”，添加控制主灯开关功能和电位器功能

【图19】【图20】

添加成功

【图21】

# 4连接esp8266

使用代码参考，修改并编译代码，编译成功。根据设备的三元信息修改代码，并在WiFi中选择了手机热点，这样避免了校园网需要用户账号登陆的问题以及可以在手机端快速查看到热点连接的设备状态。

【图22】【图23】

代码：

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

#include <ArduinoJson.h>

#include <aliyun\_mqtt.h>

#define SENSOR\_PIN 13 //pin define

#define LED D4

#define LED2 D5

#define PRODUCT\_KEY "a1d9dhNb6fp"//exchange PRODUCT\_KEY

#define DEVICE\_NAME "led1"//exchange DEVICE\_NAME

#define DEVICE\_SECRET "82b6da7e47de212fb63918ba3717a1e0"//exchange DEVICE\_SECRET

#define DEV\_VERSION "S-TH-WIFI-v1.0-20190220"

#define WIFI\_SSID "abc"//exchange WIFI

#define WIFI\_PASSWD "hl20011115"//exchange WIFI password

#define ALINK\_BODY\_FORMAT "{\"id\":\"123\",\"version\":\"1.0\",\"method\":\"%s\",\"params\":%s}"

#define ALINK\_TOPIC\_PROP\_POST "/sys/" PRODUCT\_KEY "/" DEVICE\_NAME "/thing/event/property/post"

#define ALINK\_TOPIC\_PROP\_POSTRSP "/sys/" PRODUCT\_KEY "/" DEVICE\_NAME "/thing/event/property/post\_reply"

#define ALINK\_TOPIC\_PROP\_SET "/sys/" PRODUCT\_KEY "/" DEVICE\_NAME "/thing/service/property/set"

#define ALINK\_METHOD\_PROP\_POST "thing.event.property.post"

#define ALINK\_TOPIC\_DEV\_INFO "/ota/device/inform/" PRODUCT\_KEY "/" DEVICE\_NAME ""

#define ALINK\_VERSION\_FROMA "{\"id\": 123,\"params\": {\"version\": \"%s\"}}"

unsigned long lastMs = 0;

int val=0;

WiFiClient espClient;

PubSubClient mqttClient(espClient);

void init\_wifi(const char \*ssid, const char \*password)

{

WiFi.mode(WIFI\_STA);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED)

{

Serial.println("WiFi does not connect, try again ...");

delay(500);

}

Serial.println("Wifi is connected.");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

}

void mqtt\_callback(char \*topic, byte \*payload, unsigned int length)

{

Serial.print("Message arrived [");

Serial.print(topic);

Serial.print("] ");

payload[length] = '\0';

Serial.println((char \*)payload);

Serial.println("");

Serial.println((char \*)payload);

Serial.println("");

if (strstr(topic, ALINK\_TOPIC\_PROP\_SET))

{

StaticJsonBuffer<100> jsonBuffer;

JsonObject &root = jsonBuffer.parseObject(payload);

int params\_LightSwitch = root["params"]["LightSwitch"];

if(params\_LightSwitch==1)

{ Serial.println("Led on");

digitalWrite(LED,HIGH);}

else

{ Serial.println("Led off");

digitalWrite(LED,LOW);}

if (!root.success())

{

Serial.println("parseObject() failed");

return;

}

}

}

void mqtt\_version\_post()

{

char param[512];

char jsonBuf[1024];

//sprintf(param, "{\"MotionAlarmState\":%d}", digitalRead(13));

sprintf(param, "{\"id\": 123,\"params\": {\"version\": \"%s\"}}", DEV\_VERSION);

// sprintf(jsonBuf, ALINK\_BODY\_FORMAT, ALINK\_METHOD\_PROP\_POST, param);

Serial.println(param);

mqttClient.publish(ALINK\_TOPIC\_DEV\_INFO, param);

}

void mqtt\_check\_connect()

{

while (!mqttClient.connected())//mqttδ����

{

while (connect\_aliyun\_mqtt(mqttClient, PRODUCT\_KEY, DEVICE\_NAME, DEVICE\_SECRET))

{

Serial.println("MQTT connect succeed!");

//client.subscribe(ALINK\_TOPIC\_PROP\_POSTRSP);

mqttClient.subscribe(ALINK\_TOPIC\_PROP\_SET);

Serial.println("subscribe done");

mqtt\_version\_post();

}

}

}

void mqtt\_interval\_post()

{

char param[512];

char jsonBuf[1024];

//sprintf(param, "{\"MotionAlarmState\":%d}", digitalRead(13));

sprintf(param, "{\"LightSwitch\":%d,\"range\":%d}",!digitalRead(LED),val);

sprintf(jsonBuf, ALINK\_BODY\_FORMAT, ALINK\_METHOD\_PROP\_POST, param);

Serial.println(jsonBuf);

mqttClient.publish(ALINK\_TOPIC\_PROP\_POST, jsonBuf);

}

void setup()

{

pinMode(SENSOR\_PIN, INPUT);

pinMode(LED, OUTPUT);

pinMode(LED2, OUTPUT);

/\* initialize serial for debugging \*/

Serial.begin(115200);

Serial.println("Demo Start");

init\_wifi(WIFI\_SSID, WIFI\_PASSWD);

mqttClient.setCallback(mqtt\_callback);

}

// the loop function runs over and over again forever

void loop()

{

val= analogRead(A0);

if (val>500)

{digitalWrite(LED2,HIGH);}

else

{digitalWrite(LED2,LOW);}

if (millis() - lastMs >= 5000)

{

lastMs = millis();

mqtt\_check\_connect();

/\* Post \*/

mqtt\_interval\_post();

}

mqttClient.loop();

unsigned int WAIT\_MS = 2000;

if (digitalRead(SENSOR\_PIN) == HIGH)

{

Serial.println("Motion detected!");

}

else

{

Serial.println("Motion absent!");

}

delay(WAIT\_MS); // ms

Serial.println(millis() / WAIT\_MS);

}

将代码上传至开发板，当开发板的蓝色小灯常亮时，代表着WiFi已连接

【图24】【图25】【图26】

通过阿里云实现灯的控制

【图27】

# 参考

<https://blog.csdn.net/JAVA_EE_J/article/details/126691230>

<https://blog.csdn.net/qq_51963216/article/details/121408579>