

2021年研究生专业实训报告



**实训课题 ：物联网智能硬件设计开发课程**

姓名：田晓鹏

学号：2112003361

学院：信息工程学院

指导老师：顾东袁

时间：2021年4月9日~2021年6月1日

报告提交日期：2021年06月11日

目录

[物联网智能硬件设计开发课程 实训报告 1](#_Toc74480188)

[第1章 JLINK使用 1](#_Toc74480189)

[第2章 调试F411 2](#_Toc74480190)

[1 LED调试 2](#_Toc74480191)

[2 串口通信 2](#_Toc74480192)

[2.1串口接收中断执行流程图。 3](#_Toc74480193)

[2.2主程序 3](#_Toc74480194)

[2.3效果 5](#_Toc74480195)

[3蜂鸣器调试 5](#_Toc74480196)

[3.1效果 5](#_Toc74480197)

[第3章Esp8266 6](#_Toc74480198)

[1硬件 6](#_Toc74480199)

[1.1原理图 6](#_Toc74480200)

[1.2PCB 6](#_Toc74480201)

[2软件 7](#_Toc74480202)

[2.1开发环境搭建 7](#_Toc74480203)

[3程序 9](#_Toc74480204)

[3.1库文件 9](#_Toc74480205)

[3.2WIFI 9](#_Toc74480206)

[3.3MQTT连接阿里云 9](#_Toc74480207)

[3.4连接MCU 10](#_Toc74480208)

[4问题与总结 10](#_Toc74480209)

[第4章小程序部分 11](#_Toc74480210)

[1小程序开发准备 11](#_Toc74480211)

[2 MTTQ连接准备材料 11](#_Toc74480212)

[2.1在GitHub上下载MTTQ.js库，库中有编程方法的案例供参考 11](#_Toc74480213)

[2.2去阿里云官方库下载hmac-sha1算法库hex\_hmac\_sha1.js 11](#_Toc74480214)

[2.3打开微信开发者工具创建项目 11](#_Toc74480215)

[2.4拷贝上面提供的两个mqtt.min.js和hex\_hmac\_sha1.js到项目中utils目录中去，下图为小程序默认创建的小程序项目目录 11](#_Toc74480216)

[2.5域名的设置： 11](#_Toc74480217)

[3阿里云设置 11](#_Toc74480218)

[3.1新建设备 11](#_Toc74480219)

[3.2规则引擎 12](#_Toc74480220)

[4 UI 12](#_Toc74480221)

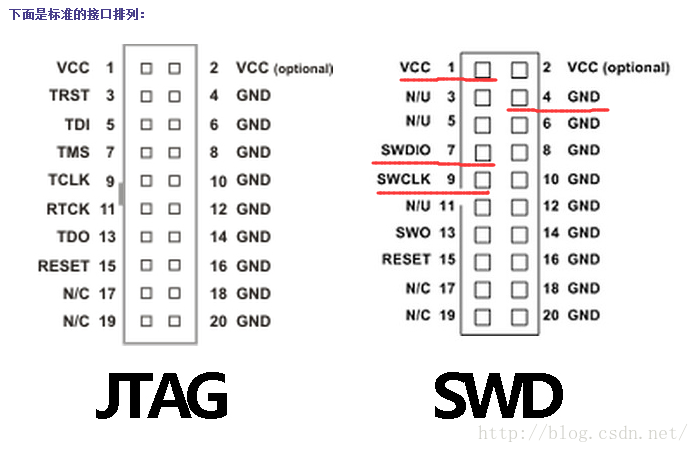
[5功能 13](#_Toc74480222)

[5.1json解析 13](#_Toc74480223)

[5.2 按键按下发送指令 13](#_Toc74480224)

物联网智能硬件设计开发课程 实训报告

# JLINK使用



按照接线图接线

## 遇到的问题：

前期接线中使用万用表量vcc3.3v于是把主板vcc接到了jlink的vcc（optinal）这个引脚有3.3v的输出，而实际上jlink是要去读主板的3.3v，并不产生3.3v。接到第一脚后即可下载程

# 调试F411

## 1LED调试

调试了两颗板载LED灯，对应引脚PC6、PC7

HAL\_GPIO\_WritePin(GPIOC,GPIO\_PIN\_7,GPIO\_PIN\_RESET);//LED0亮

HAL\_GPIO\_WritePin(GPIOC,GPIO\_PIN\_6,GPIO\_PIN\_SET);//LED1灭

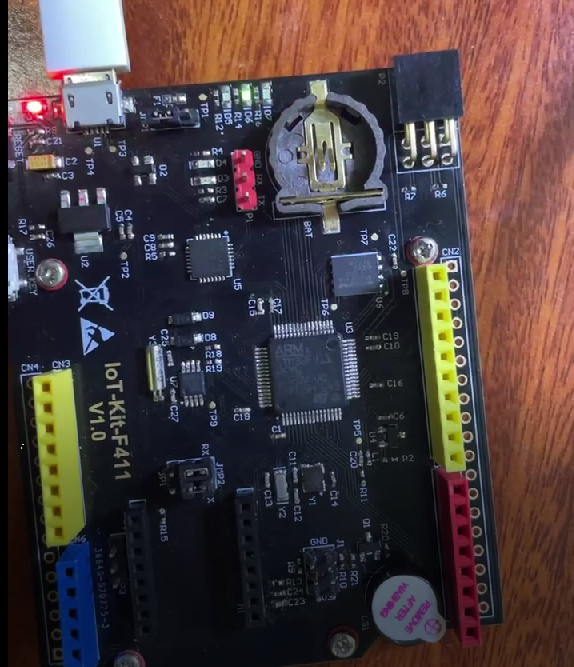
delay\_ms(500); //延时500ms

HAL\_GPIO\_WritePin(GPIOC,GPIO\_PIN\_7,GPIO\_PIN\_SET);//LED0灭

HAL\_GPIO\_WritePin(GPIOC,GPIO\_PIN\_6,GPIO\_PIN\_RESET);//LED1亮

delay\_ms(500); //延时500ms

烧写程序可以观察到两颗LED灯交替闪烁。



## 2 串口通信

411在原理图设计中使用串口一与上位机通信，对应引脚PA9、PA10。

### 2.1串口接收中断执行流程图。



HAL库一共提供了5个中断处理回调函数：

void HAL\_UART\_TxCpltCallback(UART\_HandleTypeDef\*huart);//发送完成回调函数

void HAL\_USART\_TxHalfCpltCallback(USART\_HandleTypeDef \*husart);//发送完成过半

void HAL\_UART\_RxCpltCallback(UART\_HandleTypeDef\*huart);//接收完成回调函数

Void HAL\_UART\_RxHalfCpltCallbackk(UART\_HandleTypeDef\*huart);//接收完成过半

Void HAL\_UART\_ErrorCallback(UART\_HandleTypeDef\*huart);//错误处理回调函数

### 2.2主程序

int main(void)

{

u8 len;

u16 times=0;

HAL\_Init(); //初始化HAL库

Stm32\_Clock\_Init(96,4,2,4); //设置时钟,96Mhz

delay\_init(96); //初始化延时函数

LED\_Init(); //初始化LED

uart\_init(115200); //初始化串口115200

while(1)

{

if(USART\_RX\_STA&0x8000)

{

len=USART\_RX\_STA&0x3fff;//得到此次接收到的数据长度

printf("\r\n您发送的消息为:\r\n");

HAL\_UART\_Transmit(&UART1\_Handler,(uint8\_t\*)USART\_RX\_BUF,len,1000);

//发送接收到的数据

while(\_\_HAL\_UART\_GET\_FLAG(&UART1\_Handler,

UART\_FLAG\_TC)!=SET);//等待发送结束

printf("\r\n\r\n");//插入换行

USART\_RX\_STA=0;

}else

{

times++;

if(times%5000==0)

{

printf("\r\nALIENTEK NANO STM32开发板 串口实验\r\n");

printf("555K\r\n\r\n\r\n");

}

if(times%200==0)printf("请输入数据,以回车键结束\r\n");

if(times%30==0)LED0=!LED0;//闪烁LED,提示系统正在运行.

delay\_ms(10);

}

}

}

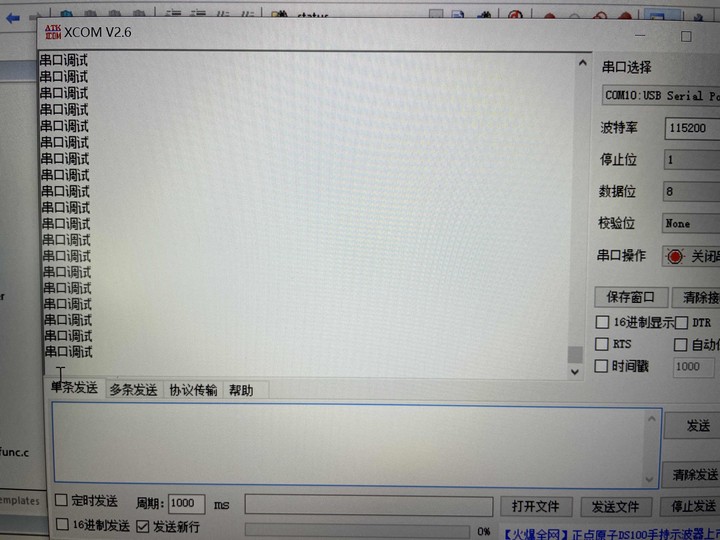
这段代码逻辑比较简单，首先判断全局变量USART\_RX\_STA的最高位是否为1，如果为1的话，那么代表前一次数据接收已经完成，接下来就是把我们自定义接收缓冲的数据发送到串口。接下来我们重点以下两句。

HAL\_UART\_Transmit(&UART1\_Handler,(uint8\*)USART\_RX\_BUF,len,1000);

While(\_\_HAL\_UART\_GET\_FLAG(&UART1\_Handler,UART\_FLAG\_TC)!=SET);

第一句，其实就是调用HAL串口发送函数HAL\_UART\_Transmit来发送一个字符到串口。第二句呢，就是我们发送一个字节之后，要检测这个数据是否已经被发送完成了。

### 2.3效果



## 3蜂鸣器调试

蜂鸣器对应引脚PC13

HAL\_GPIO\_WritePin(GPIOC,GPIO\_PIN\_6,GPIO\_PIN\_RESET); //LED0=0;

HAL\_GPIO\_WritePin(GPIOC,GPIO\_PIN\_13,GPIO\_PIN\_SET); //BEEP=1;

delay\_ms(300); //延时500ms

HAL\_GPIO\_WritePin(GPIOC,GPIO\_PIN\_6,GPIO\_PIN\_SET); //LED0=1;

HAL\_GPIO\_WritePin(GPIOC,GPIO\_PIN\_13,GPIO\_PIN\_RESET); //BEEP=0;

delay\_ms(300);

### 3.1效果

蜂鸣器每隔500ms响一次。

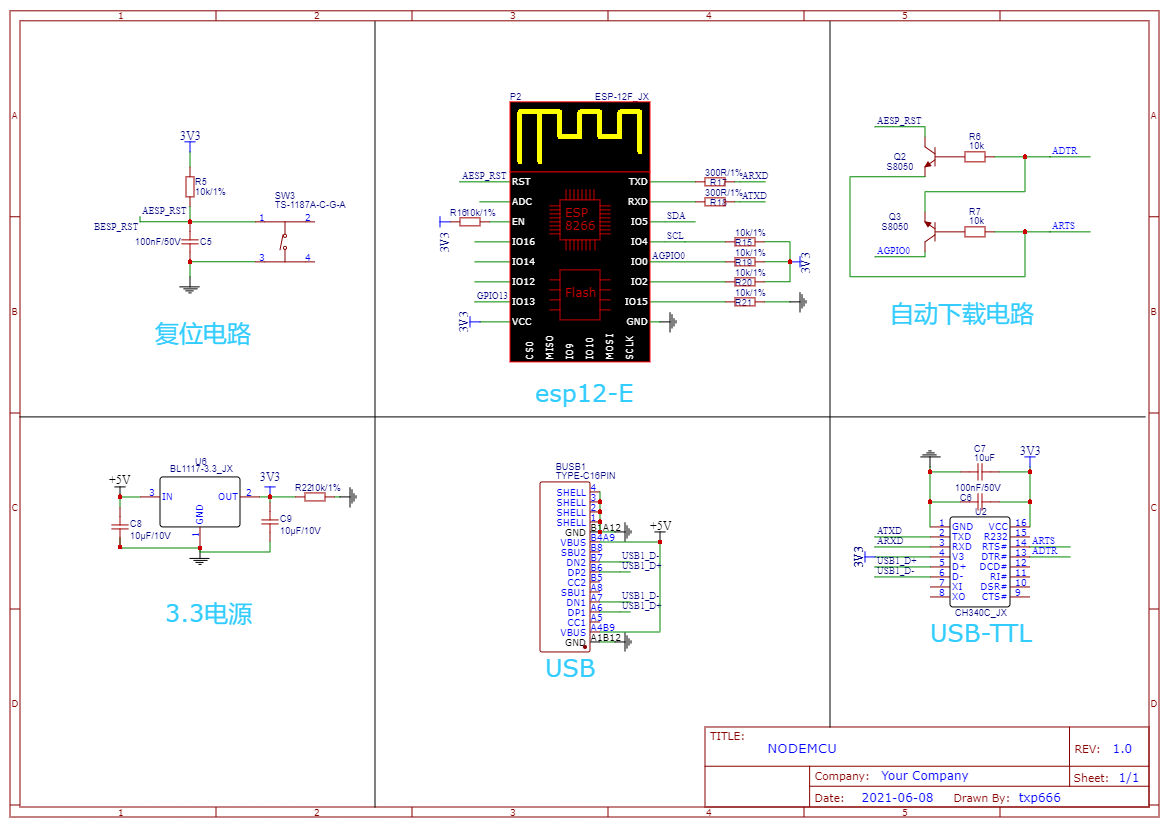
# 第3章Esp8266

使用用arduino环境进行开发

## 1硬件

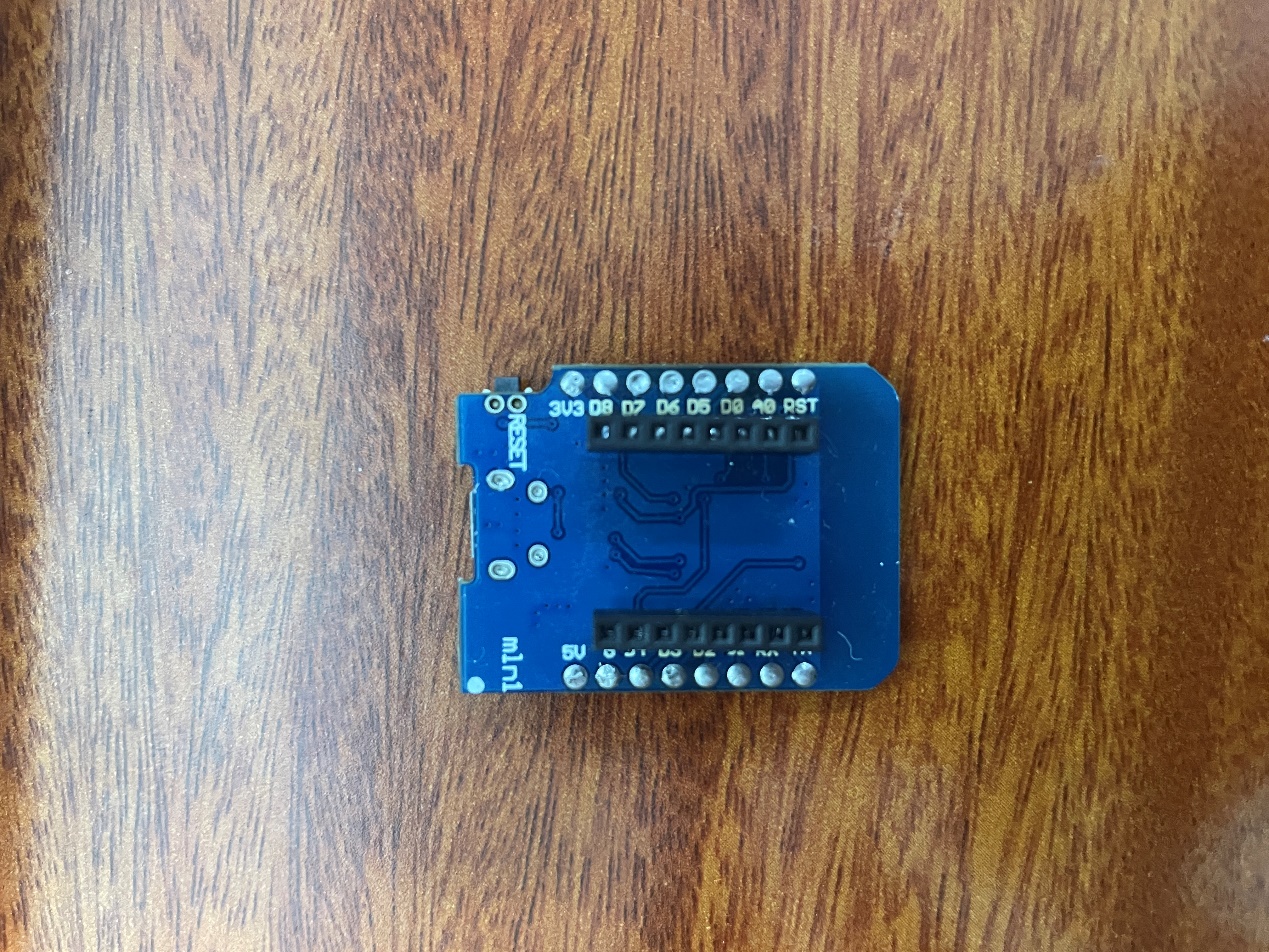
采用nodemcu方式下载程序

### 1.1原理图



### 1.2PCB

采用接插件的方式，下载程序时将esp-12E芯片插入即可



## 2软件

### 2.1开发环境搭建

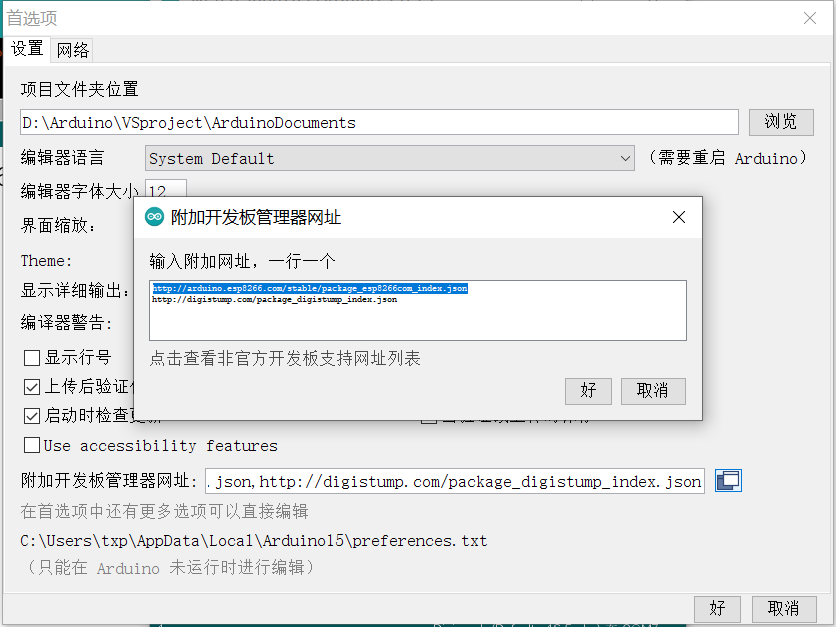
#### 2.1.1下载安装arduinoIDE（使用版本-1.8.3）

IDE如图所示

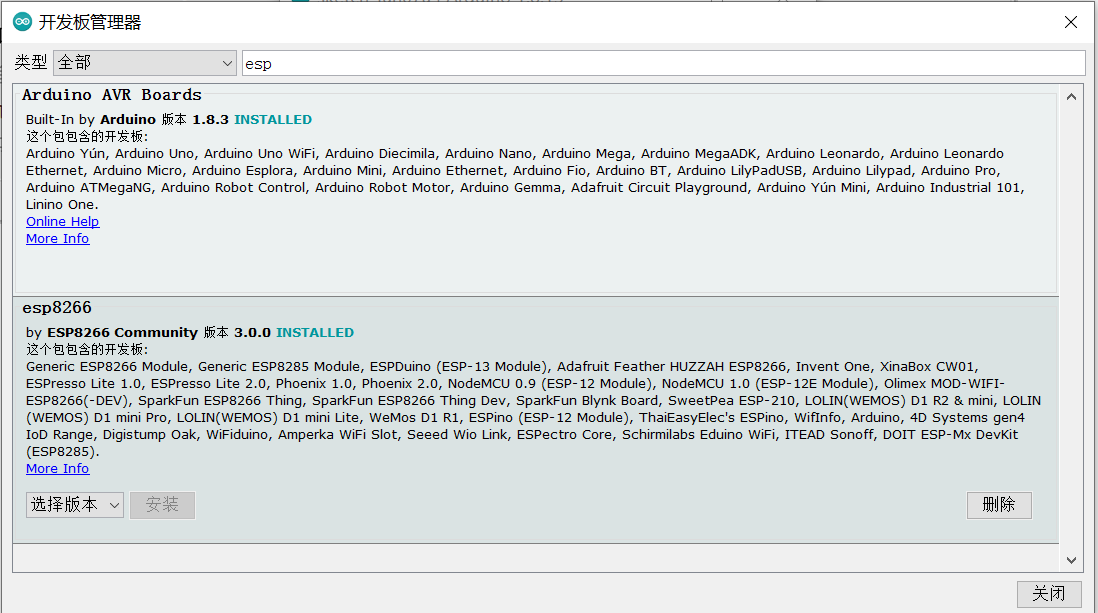


#### 2.1.2安装esp8266开发板

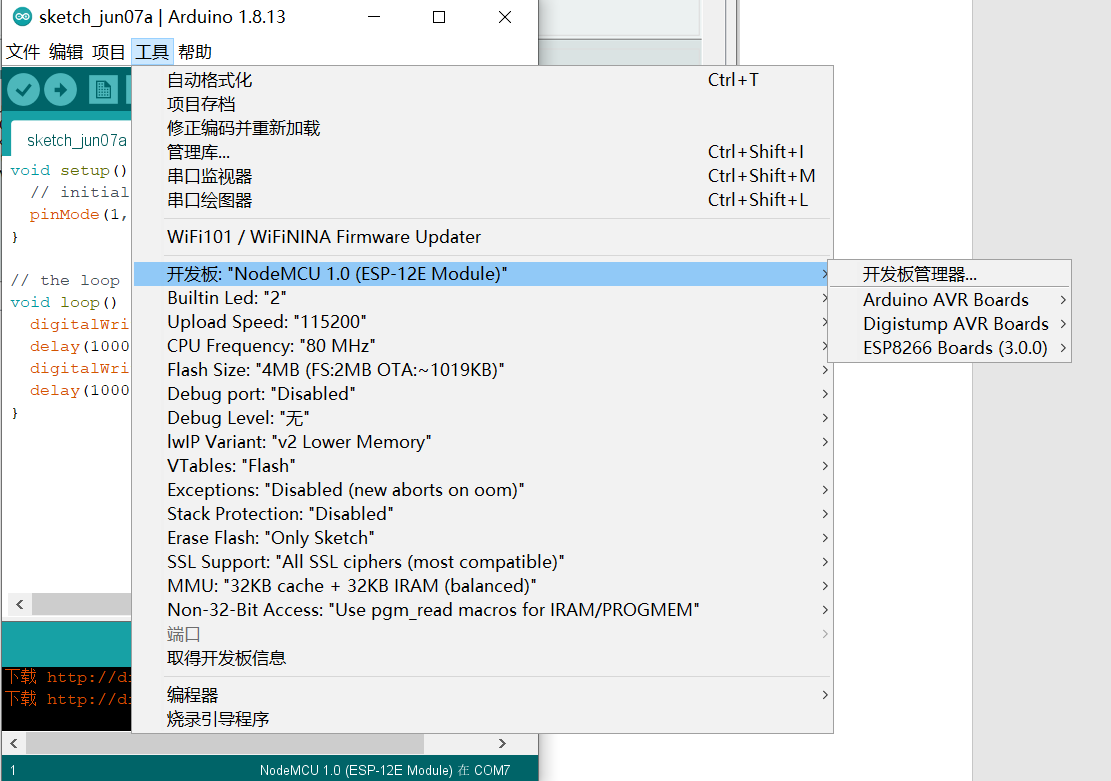
首选项中添加地址http://arduino.esp8266.com/stable/package\_esp8266com\_index.json



工具-开发板管理器中搜索esp8266进行安装



选择nodemcu1.0可进行下载



## 3程序

### 3.1库文件

在库工具中搜索以下库进行安装即可（特别注意json库使用V5版本）

#include <ESP8266WiFi.h>

主要用来连接wifi

#include <PubSubClient.h>

主要用来使用MQTT进行通信连接阿里云

#include <ArduinoJson.h>

与阿里云通信的消息是json格式需用此库进行解析和打包消息

### 3.2WIFI

在程序中直接进行wifi信息配置，调用函数即可。

#define WIFI\_SSID "666"

#define WIFI\_PASSWD "12345678"

wifiInit();

### 3.3MQTT连接阿里云

按照阿里云配置的信息，将三元组信息填入（注意要通过sha1算法生成password）

阿里云配置信息如图



### 3.4连接MCU

通过串口通信连接stm32主控

接收到CO2数值后打包发送到阿里云定义的topic主题（在mqttIntervalPost函数中）

接收到阿里云控制指令后，直接串口发送指令给stm32（在callback函数中）

## 4问题与总结

在串口通信中遇到一些问题：

接收到阿里云的控制指令后转发指令到stm32直接串口打印数字如0、1、2

Stm32接受串口数据对应的是0、1、2的ASCII码。

# 第4章小程序部分

## 1小程序开发准备

首先在小程序注册网站注册账号获取AppID为后续发布做准备

在微信开发工具网站下载开发者工具进行微信小程序开发

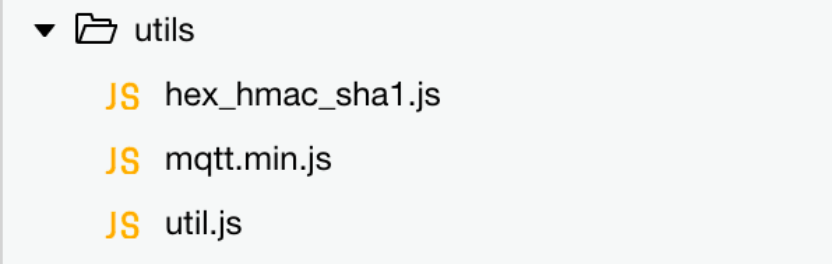
## 2 MTTQ连接准备材料

### 2.1在GitHub上下载MTTQ.js库，库中有编程方法的案例供参考

### 2.2去阿里云官方库下载hmac-sha1算法库hex\_hmac\_sha1.js

### 2.3打开微信开发者工具创建项目

### 2.4拷贝上面提供的两个mqtt.min.js和hex\_hmac\_sha1.js到项目中utils目录中去，下图为小程序默认创建的小程序项目目录



### 2.5域名的设置：

#### 2.5.1 使用不校验域名的方法，在开发者工具项目中的菜单栏的详情中勾选【不校验合法域名、web-view（业务域名）、TLS 版本以及 HTTPS 证书】

#### 2.5.2注册了小程序后，在后台可以添加socket 合法域名，格式为：wxs://productKey.iot-as-mqtt.cn-shanghai.aliyuncs.com（替换productKey为自己的产品productKey）

## 3阿里云设置

### 3.1新建设备

与esp8266相同在阿里云物联网平台同一个产品下新建一个设备，将设备三元组写入小程序代码中，如图：



### 3.2规则引擎

新建一个数据流转从微信小程序流转到esp8266实现控制功能（手动自动的切换以及开关继电器）。具体操作同阿里云设置中新建8266到微信小程序规则。

## 4 UI

本设计主要用到三个功能设计以下几个按键和显示功能如图



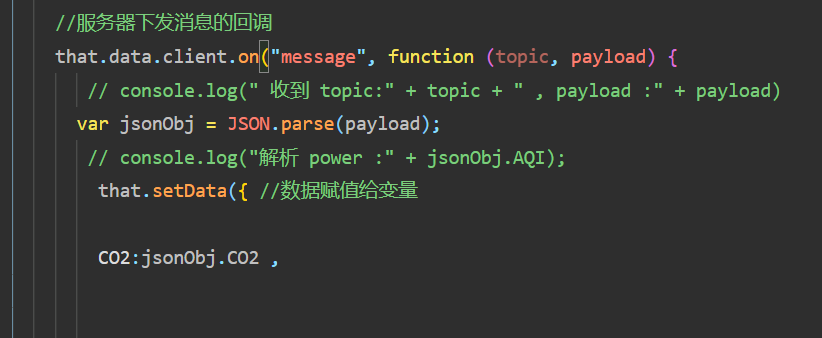
Ui代码在index.wxml中



## 5功能

实现按键控制以及CO2显示，主要涉及到接收到阿里云数据进行json解析，然后赋值，或按键按下后将对应指令json打包发送到阿里云

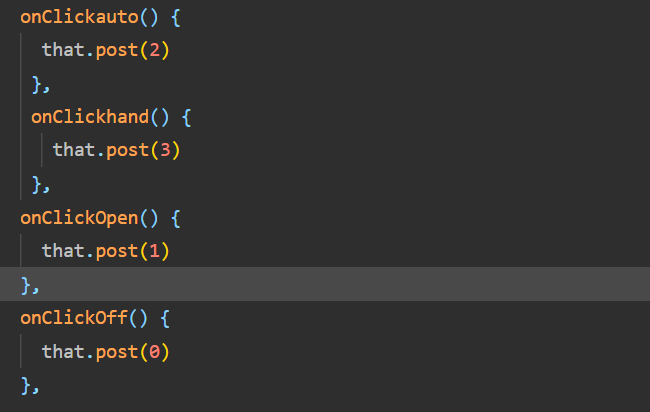
### 5.1json解析

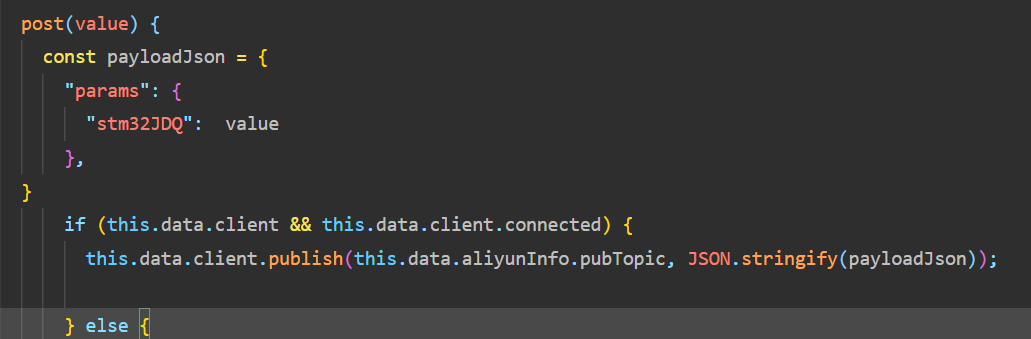


主要是接收C02的值，用标识符CO2解析得到数值赋值给变量C02

### 5.2 按键按下发送指令

2、3代表自动手动模式，0、1代表继电器开关。然后json打包，用标识符stm32JDQ.





其中params后面为阿里云指定物模型数据格式。