



浙江大學  
ZHEJIANG UNIVERSITY

## 物联网系统设计课程报告

姓名 \_\_\_\_\_

学号 \_\_\_\_\_

院所 \_\_\_\_\_ 信息与工程学院

2024 年 6 月 28 日

## 目录

<b>1</b>	<b>实验介绍</b>	<b>3</b>
1.1	实验要求 . . . . .	3
1.2	实验平台 . . . . .	3
<b>2</b>	<b>任务一: 采样温度数据, 发送云端</b>	<b>3</b>
2.1	实验过程 . . . . .	3
2.2	实验结果 . . . . .	4
<b>3</b>	<b>任务二: 设置温度报警并推送钉钉群</b>	<b>5</b>
3.1	实验过程 . . . . .	5
3.2	实验结果 . . . . .	7
<b>4</b>	<b>任务三: 创建定时开关灯的场景, 实现在指定时间开启 LED</b>	<b>8</b>
4.1	实验过程 . . . . .	8
4.2	实验结果 . . . . .	9

## 1 实验介绍

### 1.1 实验要求

1. 在设备端直接采样温度的数据，将数据发送云端，十秒更新一次。（周期性上报温度数据）
2. 设置温度报警并推送钉钉群。当温度  $>25^{\circ}\text{C}$  时，自动推送报警信息到钉钉群。
3. 在事件响应中创建定时开关灯的场景，实现在指定时间开启 LED，在特定时间关闭 LED。

### 1.2 实验平台

- 阿里云物联网平台
- Arduino 硬件平台

## 2 任务一：采样温度数据，发送云端

### 2.1 实验过程

#### 实验代码

```
1 // 自定义的宏
2 #define My_JSON_PACK_1 "{\"id\":\"666666\",\"version\":"
  "\",\"1.0\",\"method\":\"thing.event.property.post\",\""
  "params\":{\"Temperature\":%d.%02d}}\r"
3
4 //loop中的上传温度代码
5 if (bmp.takeForcedMeasurement()) {
6     // can now print out the new measurements
7     temperature=bmp.readTemperature();
8     Serial.print(F("Temperature_="));
9     Serial.print(bmp.readTemperature());
10    Serial.println("_*C");
11 }
12 else {
13     Serial.println("Forced_measurement_failed!");
14 }
15 if(Update()) Serial.println("Update_success");
```

温度上传函数如下：

```
1 bool Update()
2 {
3     bool flag;
4     int frac;
5     int zs;
```

```

6   int len;
7   cleanBuffer(ATcmd,BUF_LEN);
8   snprintf(ATcmd,BUF_LEN,AT_MQTT_PUB_SET,ProductKey,
9           DeviceName);
10  flag = check_send_cmd(ATcmd,AT_OK,DEFAULT_TIMEOUT);
11  if(!flag) {
12      Serial.println("send_AT_MQTT_PUB_SET_false");
13      return false;
14  }
15
16  cleanBuffer(ATdata,BUF_LEN_DATA);
17  zs=(int) temperature;
18  frac=(temperature-zs)*100;
19  len = snprintf(ATdata,BUF_LEN_DATA,My_JSON_PACK_1,zs,frac);
20
21  cleanBuffer(ATcmd,BUF_LEN);
22  snprintf(ATcmd,BUF_LEN,AT_MQTT_PUB_DATA,len-1);
23  flag=0;
24  flag = check_send_cmd(ATcmd,">",DEFAULT_TIMEOUT);
25  if(flag) {
26      flag = check_send_cmd(ATdata,AT_MQTT_PUB_DATA_SUCC,20);
27  }
28  else {
29      Serial.println(">_accept_wrong");
30      return false;
31  }
32  return true;
33 }

```

- 连接硬件，烧写程序，打开串口监视器
- 在串口监视器中可以看到设备上传的温度数据
- 打开阿里云平台，可以看到设备已经成功连接
- 在监控运维中可以看到设备上传的温度数据

## 2.2 实验结果

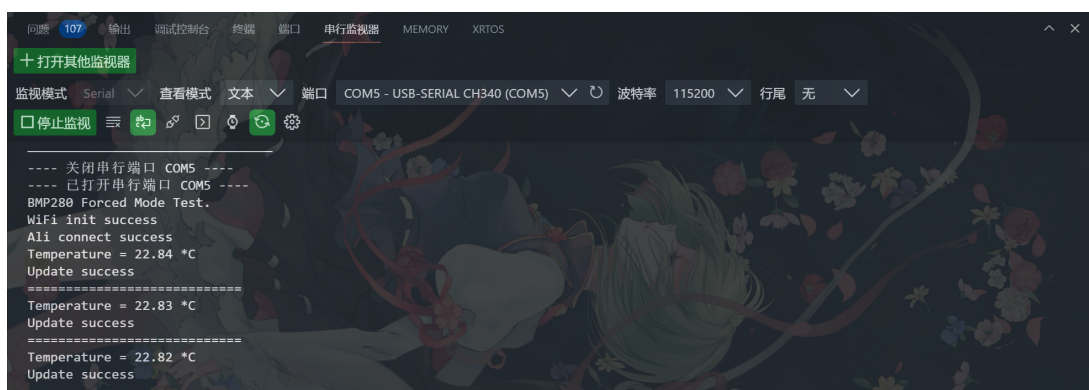


图 1: 串口显示

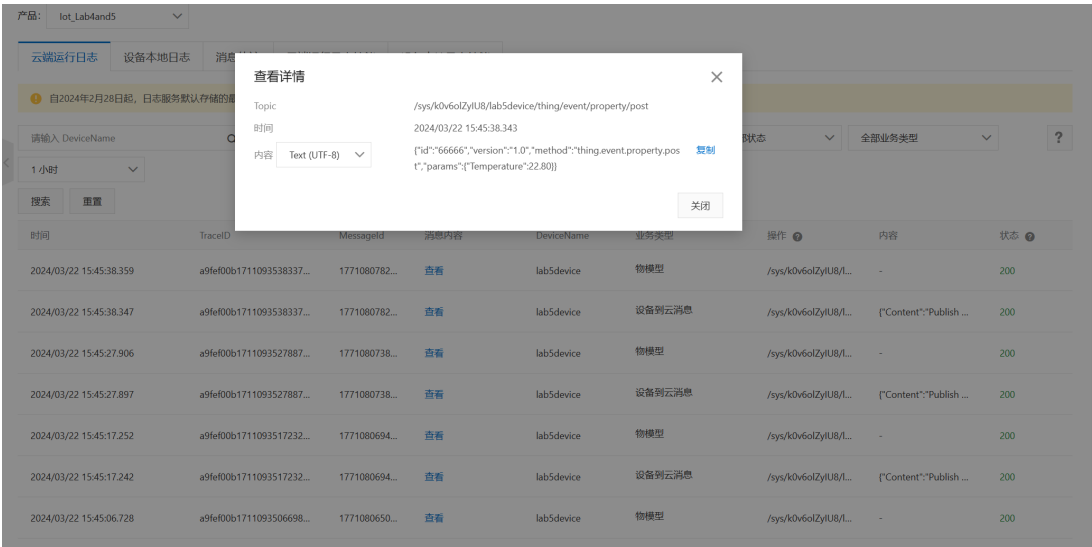


图 2: 阿里云接受温度数据

3 任务二：设置温度报警并推送钉钉群

3.1 实验过程

- 在钉钉群中设置机器人

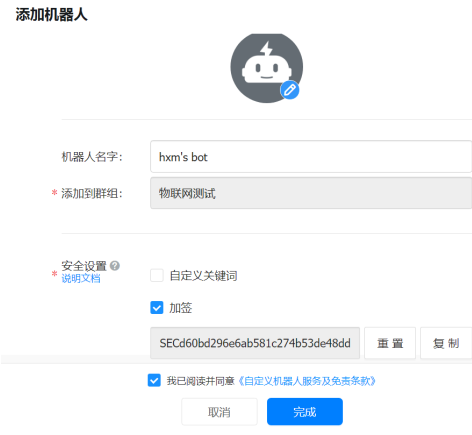


图 3: 设置钉钉机器人



图 4: 设置钉钉机器人

- 在阿里云上-事件响应-联系人管理中-创建事件联系人
- 把钉钉机器人的 webhook 地址和密钥填入
- 在事件响应，场景管理中创建温度报警场景
- 设置好场景定义和场景动作，立即启用场景

新建联系人

\* 姓名

hxms bot

\* 手机号码

请输入手机号码

钉钉机器人webhook

https://oapi.dingtalk.com/robot/send?access\_token=3b25e60bfd4023250f5

钉钉机器人token密钥

SECD60bd296e6ab581c274b53de48dd53daddb33ac9c13ea62ed44e24833a1

备注

用于lab5

确认

取消

图 5: 阿里云创建联系人

物联网平台 / iot-06x00hej1eqm2a4 / 监控运维 / 事件响应 / 场景详情

← 温度报警

未启动

编辑

场景描述

用于lab5发布温度数据

场景定义

配置说明

场景定义1

删除

当以下事件发生

\* 场景事件类型

物模型属性上报

\* 产品

iot\_Lab4and5

设备

lab5device

\* 模块

默认模块

\* 属性

BMP传感器温度

+ 新增事件过滤条件

且满足以下条件

条件1

\* 条件类型

物模型属性值

\* 产品

iot\_Lab4and5

\* 设备

lab5device

\* 模块

默认模块

\* 属性

BMP传感器温度

\* 运算符

大于等于

\* 参数

25.0

添加目标条件

删除

+ 新增或条件组

+ 新增场景定义

场景动作

删除

场景动作1

\* 场景动作类型

发送通知消息

\* 联系人类型

联系人

\* 联系人

已选择 1 个联系人

\* 通知渠道

钉钉

\* 通知模版

系统默认模版

+ 新增场景动作

确认

取消

图 6: 温度报警场景创建

6

- 加热传感器到 25 摄氏度以上，可以在钉钉中看到消息推送

## 3.2 实验结果

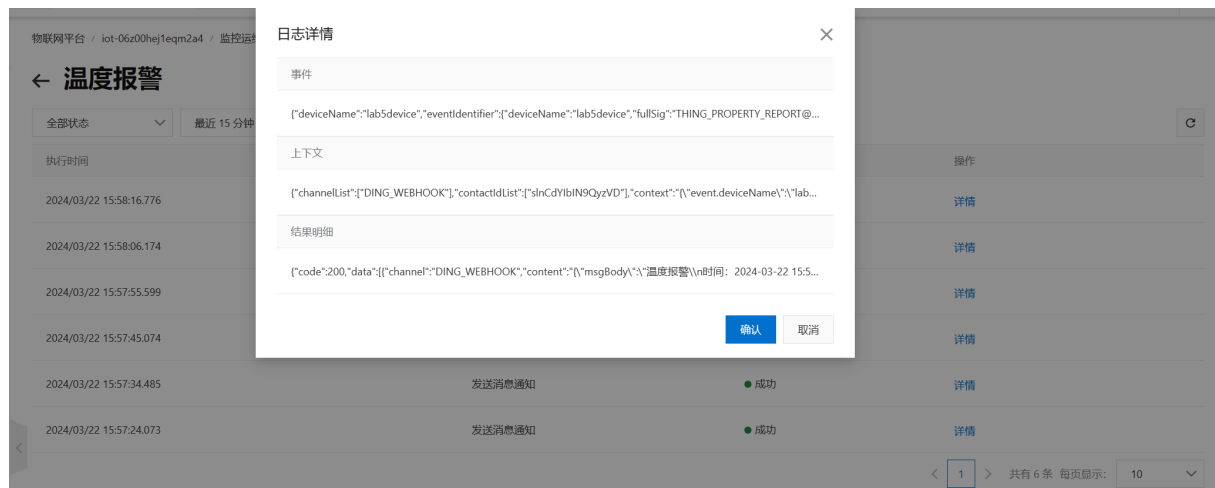


图 7: 温度报警日志

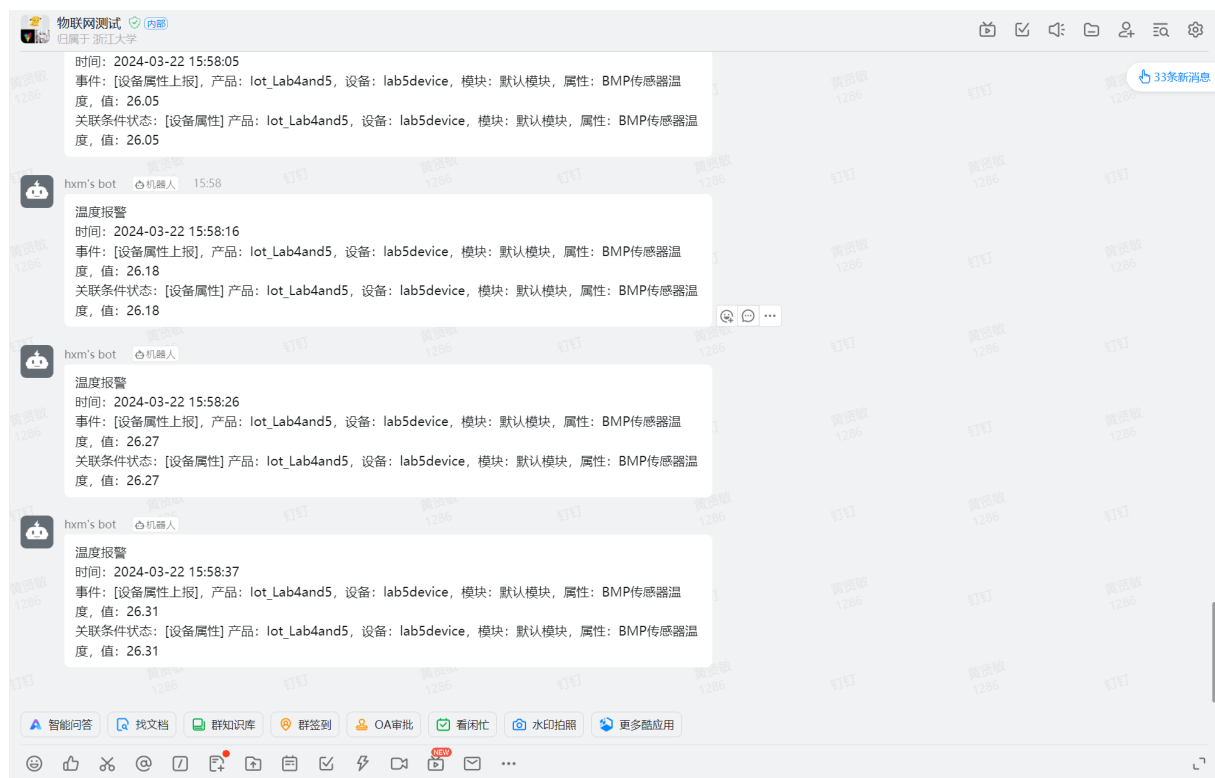


图 8: 钉钉收到温度报警

## 4 任务三：创建定时开关灯的场景，实现在指定时间开启LED

### 4.1 实验过程

#### 实验代码

```

1 //定时开启闪烁灯或关闭
2 unsigned long starmil=millis();
3 unsigned long currmil;
4 do{
5     if (Serial3.available() > 0) {
6         handleSerialData(); //串口数据处理函数
7     }
8     if (IsFlickering){ //闪烁灯实现
9         delay(delay_seconds/2);
10        analogWrite(9, ColorBlue);
11        analogWrite(7, ColorGreen);
12        analogWrite(8, ColorRed);
13
14        delay(delay_seconds/2);
15        analogWrite(9, 0);
16        analogWrite(7, 0);
17        analogWrite(8, 0);
18    }
19    currmil=millis();
20 }while (currmil-starmil < 10000); //10s计时结束后要发送一次温度数据

```

- 在事件响应中创建定时开关灯的场景，设置好参数，开启事件。

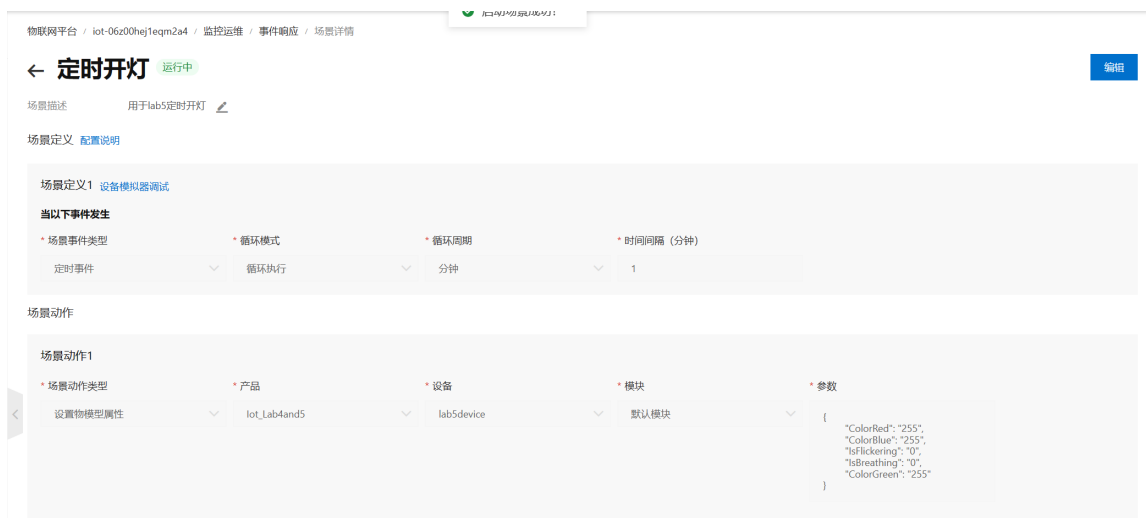


图 9: 设置开关灯场景

- 同理设置好定时关灯场景，启用场景



4.2 实验结果

可以看到定时开灯和定时关灯场景都成功执行，设备上的灯定时开启闪烁，定时关闭闪烁。

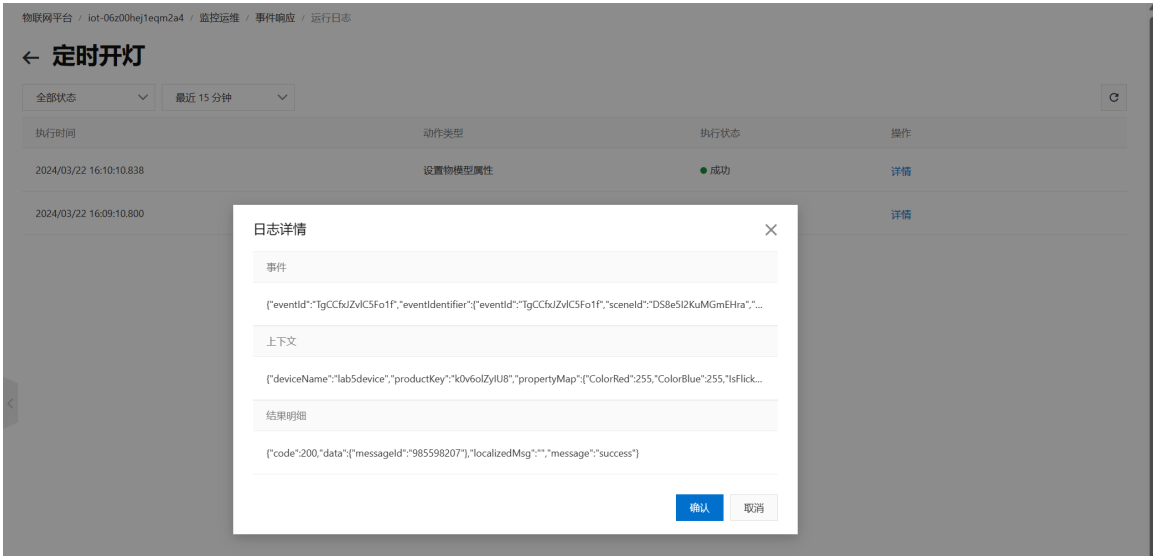


图 10: 定时开灯

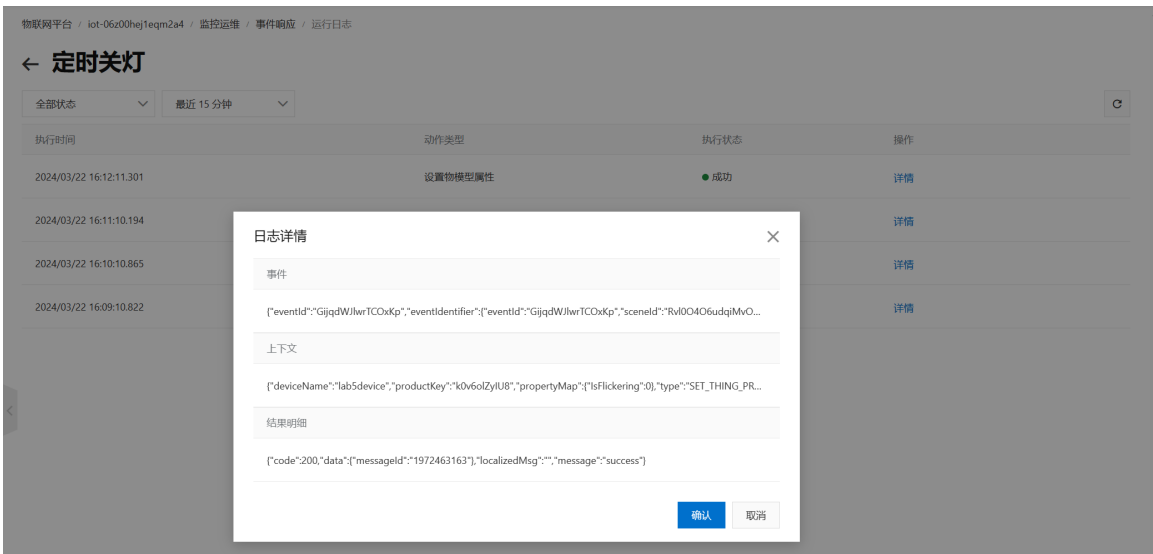


图 11: 定时关灯

## 附录

实验关键代码如下:

```

1 //setup 部分代码
2 void setup() {
3     //Serial Initial
4     Serial3.begin(115200);
5     Serial.begin(115200);
6     String inString="";
7     pinMode(7,OUTPUT); //RGB引脚
8     pinMode(8,OUTPUT);
9     pinMode(9,OUTPUT);
10    pinMode(13,OUTPUT); //板载led
11
12    // BMP module test
13    Serial.println(F("BMP280_Forced_Mode_Test."));
14    if (!bmp.begin(0x76,0x58)) {
15        Serial.println(F("Could_not_find_a_valid_BMP280_
16            sensor,_check_wiring_or_
17                "try_a_different_address!"));
18        while (1) delay(10);
19    }
20    bmp.setSampling(Adafruit_BMP280::MODE_FORCED, /*
21        Operating Mode. */
22        Adafruit_BMP280::SAMPLING_X2, /*
23            Temp. oversampling */
24        Adafruit_BMP280::SAMPLING_X16, /*
25            Pressure oversampling */
26        Adafruit_BMP280::FILTER_X16, /*
27            Filtering. */
28        Adafruit_BMP280::STANDBY_MS_500); /*
29            Standby time. */
30
31    // Pin_init();
32    BEEP(1); //蜂鸣器响一声
33
34    //wifi连接和阿里云连接
35    while(1)
36    {
37        if(!WiFi_init()) continue;
38        BEEP(2);
39        Serial.println("WiFi_init_success");
40        if(!Ali_connect()) continue;
41        break;
42    }
43    BEEP(3);
44    Serial.println("Ali_connect_success");
45
46    //

```

```
43 void loop() {
44     delay(10);
45     while (1)
46     {
47         //温度传感器上传
48         if (bmp.takeForcedMeasurement()) {
49             // can now print out the new measurements
50             temperature=bmp.readTemperature();
51             Serial.print(F("Temperature_=_"));
52             Serial.print(bmp.readTemperature());
53             Serial.println("_*C");
54         }
55         else {
56             Serial.println("Forced_measurement_failed!");
57         }
58         if(Update()) Serial.println("Update_success");
59         Serial.println("=====");
60         while(Serial3.available()){ //清楚串口缓冲区
61             Serial3.read();
62         }
63         //闪烁灯
64         unsigned long starmil=millis();
65         unsigned long currmil;
66         do{
67             if (Serial3.available() > 0) {
68                 // 读取并处理串口数据
69                 handleSerialData();
70             }
71             if (IsFlickering){//闪烁灯实现
72                 delay(delay_seconds/2);
73                 analogWrite(9, ColorBlue);
74                 analogWrite(7, ColorGreen);
75                 analogWrite(8, ColorRed);
76
77                 delay(delay_seconds/2);
78                 analogWrite(9, 0);
79                 analogWrite(7, 0);
80                 analogWrite(8, 0);
81             }
82             currmil=millis();
83         }while (currmil-starmil<10000);
84     }
85 }
86 //
87 //upload函数
88 bool Update()
89 {
90     bool flag;
91     int frac;
```

```

92     int  zs;
93     int  len;
94     cleanBuffer(ATcmd,BUF_LEN);
95     snprintf(ATcmd,BUF_LEN,AT_MQTT_PUB_SET,ProductKey,DeviceName)
96     ;
97     flag = check_send_cmd(ATcmd,AT_OK,DEFAULT_TIMEOUT);
98     if(!flag) {
99         Serial.println("send_AT_MQTT_PUB_SET_false");
100        return false;
101    }
102    cleanBuffer(ATdata,BUF_LEN_DATA);
103    zs=(int) temperature;
104    frac=(temperature-zs)*100;
105    len = snprintf(ATdata,BUF_LEN_DATA,My_JSON_PACK_1,zs,frac);
106
107    cleanBuffer(ATcmd,BUF_LEN);
108    snprintf(ATcmd,BUF_LEN,AT_MQTT_PUB_DATA,len-1);
109    flag=0;
110    flag = check_send_cmd(ATcmd,">",DEFAULT_TIMEOUT);
111    if(flag) {
112        flag = check_send_cmd(ATdata,AT_MQTT_PUB_DATA_SUCC,20);
113    }
114    else {
115        Serial.println(">_accept_wrong");
116        return false;
117    }
118    return true;
119 }
120 //
121
122 //串口数据处理函数
123 void handleSerialData() {
124     delay(10);
125     inString = Serial3.readString(); // 读取串口数据
126     if (inString != "") {
127         // 在这里处理接收到的串口数据
128         data=inString;
129         Serial.println(inString);
130     }
131     frequency=parse(data); //解析数据并打印
132     if (frequency==0) { frequency=1;}
133     Serial.print("Frequency:");
134     Serial.println(frequency);
135     delay_seconds=1.0/frequency*1000; //单位为毫秒
136     Serial.print("delay_seconds(ms):");
137     Serial.println(delay_seconds);
138     if(IsBreathing && IsFlickering){//防止两个同时为真
139         IsBreathing==0;
140         Serial.println("IsBreathing_and_IsFlickering_can't_be_

```

```
141     true_at_the_same_time , set_IsBreathing_to_false");  
142 }
```