#define NOTE\_D0 -1

#define NOTE\_D1 294

#define NOTE\_D2 330

#define NOTE\_D3 350

#define NOTE\_D4 393

#define NOTE\_D5 441

#define NOTE\_D6 495

#define NOTE\_D7 556

#define NOTE\_DH1 589

#define NOTE\_DH2 661

#define NOTE\_DH3 700

#define NOTE\_DH4 786

#define NOTE\_DH5 882

#define NOTE\_DH6 990

#define NOTE\_DH7 112

#define Do 262

#define Re 294

#define Mi 330

#define Fa 349

#define Sol 392

#define La 440

#define Si 494

#define Do\_h 523

#define Re\_h 587

#define Mi\_h 659

#define Fa\_h 698

#define Sol\_h 784

#define La\_h 880

#define Si\_h 988

#define SSID "wzldwifi" //改为你的热点名称, 不要有中文

#define PASSWORD "wzldwifi333"//改为你的WiFi密码Wi-Fi密码

#define DEVICEID "561784227" //OneNet上的设备ID

String apiKey = "7pYkJuFhqYtwnZGjY4SH5fwt=a0=";//与你的设备绑定的APIKey

/\*\*\*/

#define HOST\_NAME "api.heclouds.com"

#define HOST\_PORT (80)

#define INTERVAL\_SENSOR 5000 //定义传感器采样时间间隔 597000

#define INTERVAL\_NET 5000 //定义发送时间

//传感器部分================================

#include <Wire.h> //调用库

#include <ESP8266.h>

#define sensorPin\_1 A0

#define IDLE\_TIMEOUT\_MS 3000 // Amount of time to wait (in milliseconds) with no data

// received before closing the connection. If you know the server

// you're accessing is quick to respond, you can reduce this value.

//WEBSITE

char buf[10];

#define INTERVAL\_sensor 2000

unsigned long sensorlastTime = millis();

float tempOLED, humiOLED, lightnessOLED;

#define INTERVAL\_OLED 1000

String mCottenData;

String jsonToSend;

float sensor\_lux; //传感器voice

char sensor\_lux\_c[7] ; //换成char数组传输

#include <SoftwareSerial.h>

#define EspSerial mySerial

#define UARTSPEED 9600

SoftwareSerial mySerial(2, 3); /\* RX:D3, TX:D2 \*/

ESP8266 wifi(&EspSerial);

//ESP8266 wifi(Serial1); //定义一个ESP8266（wifi）的对象

unsigned long net\_time1 = millis(); //数据上传服务器时间

unsigned long sensor\_time = millis(); //传感器采样时间计时器

//int SensorData; //用于存储传感器数据

String postString; //用于存储发送数据的字符串

//String jsonToSend; //用于存储发送的json格式参数

const int buttonPin = 6;

int buttonState =0;

int judge=0; //传送条件判断是否按了开关

int length1=0,length2=0;

int happybirthday[]={Sol,Sol,La,Sol,Do\_h,Si,

Sol,Sol,La,Sol,Re\_h,Do\_h,

Sol,Sol,Sol\_h,Mi\_h,Do\_h,Si,La,

Fa\_h,Fa\_h,Mi\_h,Do\_h,Re\_h,Do\_h };

float durt1[]=

{

0.5,0.5,1,1,1,1+1,

0.5,0.5,1,1,1,1+1,

0.5,0.5,1,1,1,1,1,

0.5,0.5,1,1,1,1+1,

};

#include <IRremote.h>//引入红外解码库

int RECV\_PIN = 0; //红外线接收器OUTPUT端接在pin 10

IRrecv irrecv(RECV\_PIN); //定义IRrecv对象来接收红外线信号

decode\_results results; //解码结果放在decode\_results构造的对象results里

void setup(void) //初始化函数

{ irrecv.enableIRIn(); // 启动红外解码

Wire.begin();

Serial.begin(115200);

while (!Serial); // wait for Leonardo enumeration, others continue immediately

Serial.print(F("setup begin\r\n"));

delay(100);

pinMode(sensorPin\_1, INPUT);

pinMode(6,INPUT);

pinMode(8,OUTPUT);

pinMode(12,OUTPUT);

length1=sizeof(happybirthday)/sizeof(happybirthday[0]);

WifiInit(EspSerial, UARTSPEED);

Serial.print(F("FW Version:"));

Serial.println(wifi.getVersion().c\_str());

if (wifi.setOprToStationSoftAP()) {

Serial.print(F("to station + softap ok\r\n"));

} else {

Serial.print(F("to station + softap err\r\n"));

}

if (wifi.joinAP(SSID, PASSWORD)) {

Serial.print(F("Join AP success\r\n"));

} else {

Serial.print(F("Join AP failure\r\n"));

}

if (wifi.disableMUX()) {

Serial.print(F("single ok\r\n"));

} else {

Serial.print(F("single err\r\n"));

}

Serial.print(F("setup end\r\n"));

}

void loop(void) {

if (irrecv.decode(&results))

{

Serial.println(results.value);

switch (results.value)

{

case 33456255:

Music1();

Serial.println("happy");

break;

case 33439935:

Music2();

Serial.println("sky");

break;

case 33472575:

Music3();

Serial.println("love");

break;

case 33431775:

/\*buttonState=digitalRead(buttonPin);

if(buttonState==0){ judge=1; };

if (sensor\_time > millis()) sensor\_time = millis();

if(millis() - sensor\_time > INTERVAL\_SENSOR) //传感器采样时间间隔

{

getSensorData(); //读串口中的传感器数据

sensor\_time = millis();

}

if (net\_time1 > millis()) net\_time1 = millis();

if (millis() - net\_time1 > INTERVAL\_NET) //发送数据时间间隔

{

if(judge==1){

digitalWrite(8,HIGH);

updateSensorData(); //将数据上传到服务器的函数

net\_time1 = millis();

delay(2000);

digitalWrite(8,LOW);

judge=0;

}

}\*/

break;

default:

break;

}

irrecv.resume();

}

}

void Music1( ){

for(int x=0;x<length1;x++)

{

tone(12,happybirthday[x]);

delay(500\*durt1[x]);

noTone(12);

}

delay(3000);

}

void Music2(){

int skycity[] =

{

NOTE\_D0,NOTE\_D6,NOTE\_D7,NOTE\_DH1, NOTE\_D7,NOTE\_DH1,NOTE\_DH3,NOTE\_D7, NOTE\_D3,

NOTE\_D6,NOTE\_D5,NOTE\_D6,NOTE\_DH1,NOTE\_D5, NOTE\_D5,NOTE\_D5,NOTE\_D3,NOTE\_D4,NOTE\_D3, NOTE\_D4,NOTE\_DH1,

NOTE\_D3,NOTE\_D3,NOTE\_D0,NOTE\_DH1,NOTE\_D7,NOTE\_D4,NOTE\_D4,NOTE\_D7,NOTE\_D7,NOTE\_D7,NOTE\_D0,NOTE\_D6,NOTE\_D7,

NOTE\_DH1,NOTE\_D7,NOTE\_DH1,NOTE\_DH3,NOTE\_D7,NOTE\_D7,NOTE\_D7,NOTE\_D3

};

float durt2[]=

{

1,0.5,0.5, 1+0.5, 0.5,1,1,1,0.5,

1+0.5,0.5,1,1, 1,1,1,1,1+0.5,0.5,1,1,

1,1,0.5,0.5, 1+0.5,0.5,1,1, 1,1,1,0.5,0.5,

1+0.5,0.5,1,1, 1,1,1,0.5

};

length2=sizeof(skycity)/sizeof(skycity[0]);

for(int x=0;x<length2;x++)

{

tone(12,skycity[x]);

delay(400\*durt2[x]);

noTone(12);

}

delay(3000);

}

void Music3(){

int loveriver[]={La,Do\_h,Re\_h,Re\_h,Re\_h, Re\_h,Re\_h,Mi\_h,Fa\_h,Fa\_h,Mi\_h, Do\_h,Re\_h,Mi\_h,Mi\_h, Sol\_h,La\_h,La\_h,La\_h,Do\_h, Do\_h, Mi\_h,Re\_h,Do\_h,Re\_h,Mi\_h };

float durt3[]= {1,1,0.7,0.7,0.7, 1,0.8,0.8,0.9,1.1, 1.1+0.5 ,0.9,0.9,0.9,0.9, 1, 1,1.5,1,1, 1,1,1.3,1.3,1.3,1.3 };

int length3=sizeof(loveriver)/sizeof(loveriver[0]);

for(int x=0;x<length3;x++)

{

tone(12,loveriver[x]);

delay(500\* durt3[x]);

noTone(12);

}

delay(3000);

}

void getSensorData(){

sensor\_lux = analogRead(A0);

dtostrf(sensor\_lux, 3, 1, sensor\_lux\_c);

}

void updateSensorData() {

if (wifi.createTCP(HOST\_NAME, HOST\_PORT)) { //建立TCP连接，如果失败，不能发送该数据

Serial.print("create tcp ok\r\n");

jsonToSend="{\"Light\":";

dtostrf(sensor\_lux,1,2,buf);

jsonToSend+="\""+String(buf)+"\"";

jsonToSend+="}";

postString="POST /devices/";

postString+=DEVICEID;

postString+="/datapoints?type=3 HTTP/1.1";

postString+="\r\n";

postString+="api-key:";

postString+=apiKey;

postString+="\r\n";

postString+="Host:api.heclouds.com\r\n";

postString+="Connection:close\r\n";

postString+="Content-Length:";

postString+=jsonToSend.length();

postString+="\r\n";

postString+="\r\n";

postString+=jsonToSend;

postString+="\r\n";

postString+="\r\n";

postString+="\r\n";

const char \*postArray = postString.c\_str(); //将str转化为char数组

Serial.println(postArray);

wifi.send((const uint8\_t\*)postArray, strlen(postArray)); //send发送命令，参数必须是这两种格式，尤其是(const uint8\_t\*)

Serial.println("send success");

if (wifi.releaseTCP()) { //释放TCP连接

Serial.print("release tcp ok\r\n");

}

else {

Serial.print("release tcp err\r\n");

}

postArray = NULL; //清空数组，等待下次传输数据

} else {

Serial.print("create tcp err\r\n");

}

}