#define SSID "1" //改为你的热点名称, 不要有中文

#define PASSWORD "123456789"//改为你的WiFi密码Wi-Fi密码

#define DEVICEID "561985997" //OneNet上的设备ID

String apiKey = "O37QS2O6BXh7CJVi6gdmJPvt6MM=";//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>

#include <I2Cdev.h> //调用库

/\*\*\*\*\*\*\*温湿度\*\*\*\*\*\*\*/

#include <Microduino\_SHT2x.h>

/\*\*\*\*\*\*\*光照\*\*\*\*\*\*\*/

#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;

//3,传感器值的设置

float sensor\_tem, sensor\_hum, sensor\_lux; //传感器温度、湿度、光照

char sensor\_tem\_c[7], sensor\_hum\_c[7], 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

Tem\_Hum\_S2 TempMonitor;

void setup(void) //初始化函数

{

//初始化串口波特率

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);

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"));

Serial.print(F("IP:"));

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

} 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) //循环函数

{

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

{

updateSensorData(); //将数据上传到服务器的函数

net\_time1 = millis();

}

}

void getSensorData(){

sensor\_tem = TempMonitor.getTemperature();

sensor\_hum = TempMonitor.getHumidity();

//获取光照

sensor\_lux = analogRead(A0);

delay(1000);

dtostrf(sensor\_tem, 2, 1, sensor\_tem\_c);

dtostrf(sensor\_hum, 2, 1, sensor\_hum\_c);

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="{\"Temperature\":";

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

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

jsonToSend+=",\"Humidity\":";

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

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

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");

}

}