//mcookie开发

#define SSID "HUAWEI Mate 30 5G" //改为你的Wi-Fi名称

#define PASSWORD "zzzy9120"//Wi-Fi密码

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

#define DEVICEID "562214216" //OneNet上的设备ID

#define PROJECTID "287819" //OneNet上的产品ID

#define HOST\_PORT (80)

String apiKey="8U9dfpYNgd0HsNHlL5vR1pqLdxU=";//与你的设备绑定的APIKey

#define INTERVAL\_SENSOR 200 //定义传感器采样时间间隔

#define INTERVAL\_NET 200 //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

//#include <Microduino\_SHT2x.h>

#define sensorPin\_1 A0

int level\_pin=4; //定义引脚

#define IDLE\_TIMEOUT\_MS 300 // 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 200

unsigned long sensorlastTime = millis();

String mCottenData;

String jsonToSend;

//3,传感器值的设置

float level,kaiguan; //传感器水位,用光照传感器作为开关

char level\_c[7],kaiguan\_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(level\_pin,INPUT\_PULLUP); //初始化液位感应器引脚

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(){

level=digitalRead(level\_pin); //获取水位传感器信息

kaiguan = analogRead(A0);

delay(100);

Serial.print("level= ");

Serial.println(level,DEC);

Serial.print("kaiguan= ");

Serial.println(kaiguan,DEC);

dtostrf(level, 3, 1, level\_c); //把float类型转化为char类型

dtostrf(kaiguan, 3, 1, kaiguan\_c);//把float类型转化为char类型

}

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="{\"level1\":";

dtostrf(level,1,2,buf);

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

jsonToSend+=",\"kaiguan1\":";

dtostrf(kaiguan,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");

}

}

//ESP8266开发

#include <ESP8266WiFi.h>

#include <WiFiClientSecure.h>

#include<Servo.h>

Servo myservo;

#define servoPin 2

int pos=0;

char status1;

char lastStatus1='0';

float lastStatus;

const char\* ssid = "HUAWEI Mate 30 5G"; // wifi名

const char\* password = "zzzy9120"; // wifi密码

const char\* host = "api.heclouds.com"; // 连接的主机域名

const int httpsPort = 80; // https端口

String API\_KEY = "8U9dfpYNgd0HsNHlL5vR1pqLdxU="; // onenet的 api key

String deviceId = "562214216"; // onenet的设备ID

float getDataToOnenet1();

void lightOn();

void lightClose();

char readData(String line);

void setup() {

Serial.begin(9600);

Serial.print("connecting to ");

Serial.println(ssid);

WiFi.mode(WIFI\_STA); // 设置wifi模式

WiFi.begin(ssid, password); // 连接wifi

while (WiFi.status() != WL\_CONNECTED) { //判断连接状态

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

myservo.attach(servoPin);

pinMode(4,OUTPUT);

}

void loop() {

float data1 = getDataToOnenet1();

if(lastStatus1!=status1)

{

if(status1=='0')

lightClose();

else if(status1=='1')

lightOn();

}

lastStatus=status1;

delay(100);

}

float getDataToOnenet1(){

WiFiClient client; // HTTP

Serial.print("connecting to ");

Serial.println(host);

if (!client.connect(host, httpsPort)) { // 判断连接情况

Serial.println("connection failed");

return -1;

}

Serial.print("requesting URL: ");

// 发送GET请求

// 组拼url地址

// 组拼HTTPS请求的Header

String getStrlight = String("GET /devices/")+deviceId+ "/datapoints?datastream\_id=light&limit=1 HTTP/1.1\r\n"+

"api-key: " + API\_KEY + "\r\n" +

"Host:" + host + "\r\n"

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

client.print(getStrlight);

Serial.println("request sent");

// 读取连接情况

while (client.connected()) {

String line = client.readStringUntil('\n');

if (line == "\r") {

Serial.println("headers received");

break;

}

}

// 读取数据

String line = client.readStringUntil('~');

Serial.println("reply was:");

Serial.println("==========");

Serial.println(line); // 打印接受到的数据

Serial.println("==========");

Serial.println("closing connection");

status1=readData(line);

return 0;

}

char readData(String line)

{

int len=line.length();

char a;

if (len > 0)

{

for (int i = 0; i < len; i++)

{

if((char)line[i]=='v'&&(char)line[i+1]=='a'&&(char)line[i+2]=='l'&&(char)line[i+3]=='u'&&(char)line[i+4]=='e')

{

a=line[i+8];

break;

}

}

}

return a;

}