电机

//D6，D8控制1A，1B的电机

#define OUT1A 6

#define OUT1B 8

int value, data;

/\*

遥控器

\*/

#include <IRremote.h>

int RECV\_PIN = 10; //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() {

// put your setup code here, to run once:

Serial.begin(9600); //串口初始化

pinMode(OUT1A, OUTPUT);

pinMode(OUT1B, OUTPUT);

/\*

遥控器

\*/

irrecv.enableIRIn(); // 启动红外解码

}

void motor\_sta(int num, int fadeValue)

{

if (num == 0)

{

analogWrite(OUT1A, fadeValue);

digitalWrite(OUT1B, LOW);

}

else if (num == 1)

{

digitalWrite(OUT1A, LOW);

analogWrite(OUT1B, fadeValue);

}

}

int uiStep() //切歌

{

if (results.value==33464415) //UP

return 1;

if (results.value==33478695) //DOWN

return 2;

if (results.value==33427695) //OK

return 3;

return 0;

}

void loop() {

if (uiStep() == 1)

{

value =130;

data = 1;

}

else if (uiStep() == 2)

{

value =130;

data = 0;

}

else if (uiStep() == 3)

value=0;

motor\_sta(data, value);

if (irrecv.decode(&results)) { //解码成功，收到一组红外线信号

if (results.value!=4294967295)

Serial.println(results.value, DEC);//// 输出红外线解码结果（十六进制）

irrecv.resume(); // 接收下一个值

/\*

Serial.print(data);

Serial.print(" ");

Serial.println(value);

\*/

}

}

红外报警及Arduino将数据传至onenet

/\*#define humanHotSensor 4//PIR传感器D4

#define PIN 6//彩灯

Adafruit\_NeoPixel strip = Adafruit\_NeoPixel(1, PIN, NEO\_GRB + NEO\_KHZ800);

void setup() {

strip.begin();

Serial.begin(9600);

pinMode(humanHotSensor, INPUT);}

void loop() {

humanHotState = digitalRead(humanHotSensor);

// print out the state of the button:

//Serial.println(humanHotState);

//delay(1); // delay in between reads for stability

if (humanHotState) {

strip.setPixelColor(0, strip.Color(255, 0, 0));//红光

strip.show(); //LED显示

delay(1000); //延迟1秒输出

}

else

{ strip.setPixelColor(0, strip.Color(0, 0, 0));//红光

strip.show(); //LED显示

delay(1000); //延迟1秒输出

}

}\*/

#include <Adafruit\_NeoPixel.h>//引用头文件

#define humanHotSensor 4//PIR传感器D4

#define PIN 8//彩灯

Adafruit\_NeoPixel strip = Adafruit\_NeoPixel(2, PIN, NEO\_GRB + NEO\_KHZ800);

int humanHotState;

#define mic\_pin A6

#define buzzer\_pin 6

#include <Microduino\_RTC.h>

RTC rtc;

/\* 设置RTC启动时间

\* 年, 月, 星期, 日, 时, 分, 秒 \*/

DateTime dateTime = {2016, 6, 3, 1, 2, 30, 40};

uint16\_t tYear;

uint8\_t tMonth, tWeekday, tDay, tHour, tMinute, tSecond;

#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 <SHT2x.h>

//光照

#define sensorPin\_1 A0

#define SSID "HUAWEI nova 2" // cannot be longer than 32 characters!

#define PASSWORD "11111111"

#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

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

#define DEVICEID "20449602"

#define PROJECTID "106194"

#define HOST\_PORT (80)

String apiKey="RFeyfLJTSW6WcEeWSPIfmD5c8Xo=";

char buf[10];

#define INTERVAL\_sensor 2000

unsigned long sensorlastTime = millis();

float tempOLED, humiOLED, lightnessOLED,micOLED;

#define INTERVAL\_OLED 1000

String mCottenData;

String jsonToSend;

//3,传感器值的设置

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

char sensor\_tem\_c[7], sensor\_hum\_c[7], sensor\_lux\_c[7],sensor\_sound\_c[7] ; //换成char数组传输

#include <SoftwareSerial.h>

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

ESP8266 wifi(mySerial);

//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

void setup(void) //初始化函数

{

strip.begin();

//清除所有寄存器

rtc.begin();

rtc.clearAll();

//设置启动时间

rtc.setDateTime(dateTime);

//初始化串口波特率

Wire.begin();

Serial.begin(115200);

while(!Serial);

pinMode(sensorPin\_1, INPUT);

pinMode(humanHotSensor, INPUT);

pinMode(buzzer\_pin,OUTPUT);

pinMode(mic\_pin, INPUT);

//ESP8266初始化

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

Serial.print("FW Version:");

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

if (wifi.setOprToStationSoftAP()) {

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

} else {

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

}

if (wifi.joinAP(SSID, PASSWORD)) { //加入无线网

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

Serial.print("IP: ");

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

} else {

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

}

if (wifi.disableMUX()) {

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

} else {

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

}

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

}

void loop(void) //循环函数

{

rtc.getDate(&tYear, &tMonth, &tWeekday, &tDay);

rtc.getTime(&tHour, &tMinute, &tSecond);

humanHotState = digitalRead(humanHotSensor);

// print out the state of the button:

//Serial.println(humanHotState);

//delay(1); // delay in between reads for stability

if((tHour>=1&&tHour<=5)||(tHour>=8&&tHour<=17)) //规定运行的时间

{

if (humanHotState) {

strip.setPixelColor(0, strip.Color(255, 0, 0));//红光

strip.show(); //LED显示

delay(1000); //延迟1秒输出

for(int i=200;i<=800;i++) //用循环的方式将频率从200HZ 增加到800HZ

{

tone(buzzer\_pin,i); //在端口输出频率

delay(5); //该频率维持5毫秒

}

delay(1000); //最高频率下维持1秒钟

for(int i=800;i>=200;i--)

{

tone(buzzer\_pin,i);

delay(10); //该频率维持10毫秒

}

}

else

{ strip.setPixelColor(0, strip.Color(0, 0, 0));//红光

strip.show(); //LED显示

delay(1000); //延迟1秒输出

tone(buzzer\_pin,0);

}

}

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 = SHT2x.readT() ;

sensor\_hum = SHT2x.readRH();

//获取光照

sensor\_lux = analogRead(A0);

sensor\_sound=analogRead(A2);

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);

dtostrf(sensor\_sound, 2, 1, sensor\_sound\_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+=",\"Sound\":";

dtostrf(sensor\_sound,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");

}

}

彩灯

#include <IRremote.h>

#include <U8glib.h>

int RECV\_PIN = 10;

IRrecv irrecv(RECV\_PIN);

decode\_results results;

unsigned long lcd\_time = millis(); //OLED刷新时间计时器

U8GLIB\_SSD1306\_128X64 u8g(U8G\_I2C\_OPT\_NONE); //设置OLED型号

//-------字体设置，大、中、小

#define setFont\_L u8g.setFont(u8g\_font\_7x13)

#define setFont\_M u8g.setFont(u8g\_font\_fixed\_v0r)

#define setFont\_S u8g.setFont(u8g\_font\_fixed\_v0r)

#define setFont\_SS u8g.setFont(u8g\_font\_fub25n)

#include <Adafruit\_NeoPixel.h>//引用头文件

#define PIN 6 /\*定义了控制LED的引脚，6表示Microduino的D6引脚，可通过Hub转接出来，用户可以更改 \*/

#define PIN\_NUM 2 //定义允许接的led灯的个数

Adafruit\_NeoPixel strip = Adafruit\_NeoPixel(PIN\_NUM, PIN, NEO\_GRB + NEO\_KHZ800); //该函数第一个参数控制串联灯的个数，第二个是控制用哪个pin脚输出，第三个显示颜色和变化闪烁频率

int i=0;

void setup() {

Serial.begin(9600);

irrecv.enableIRIn(); // 启动红外解码

Serial.println("Initialisation complete.");

strip.begin();

}

void loop() {

if (irrecv.decode(&results)) {

if(results.value!=0&&results.value!=4294967295)

{

if(results.value==33456255)

{

i=50;

}

if(results.value==33439935)

{

i=120;

}

if(results.value==33472575)

{

i=255;

}

if(results.value==33464415)

{

i=i+20;

if(i>255)

{

i=255;

}

}

if(results.value==33478695)

{

i=i-20;

if(i<0)

{

i=0;

}

}

if(results.value==33441975)

{

i=0;

}

strip.setPixelColor(0, strip.Color(i, i, i));

strip.show(); //LED显示

delay(1000); //延迟1秒输出

Serial.println(results.value);

Serial.println(i);

}

u8g.firstPage();//显示光度

do {

setFont\_L;

u8g.setPrintPos(4, 16);

u8g.print(i);

}while( u8g.nextPage() );

irrecv.resume();

}

}