代码：

void setup(void)

{

Serial.begin(9600);

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)) {

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)

{

uint8\_t buffer[128] = {0};

if (wifi.createTCP(HOST\_NAME, HOST\_PORT)) {

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

//char \*hello = "Hello, this is client!";

// wifi.send((const uint8\_t\*)hello, strlen(hello));

while(true){

String str="";

unsigned int irSignal[140];

uint32\_t len = wifi.recv(buffer, sizeof(buffer), 10000);

if (len > 0) {

Serial.print("Received:[");

for(uint32\_t i = 0; i < len-1; i++) {

if((char)buffer[i]!=','){

Serial.print((char)buffer[i]);

str=str+((char)buffer[i]);

}else{

irSignal[i]=str.toInt();

str="";

}

}

Serial.print("]\r\n");

}

int khz = 38;

irsend.sendRaw(irSignal, sizeof(irSignal), khz);

}

} else {

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

}

delay(5000);

}

String sendData(String command, const int timeout, boolean debug)

{

String response = "";

ser.print(command); // send the read character to the esp8266

long int time = millis();

while( (time+timeout) > millis())

{

while(ser.available())

{

char c = ser.read(); // read the next character.

response+=c;

}

} if(debug)

{

Serial.print(response);

} return response;

}

void loop() {

// read the value from LM35.read 10 values for averaging.

int val = 0;

for(int i = 0; i < 10; i++) {

val += analogRead(lm35Pin);

delay(500);

} // 转换成温度数据

float temp = val\*50.0f/1023.0f;

// convert to string

char buf[16];

String strTemp = dtostrf(temp, 4, 1, buf);

Serial.println(strTemp);

// 和Web服务器建立连接

String cmd = "AT+CIPSTART=\"TCP\",\"10.18.55.240\",8080\r\n";

sendData(cmd,5000,DEBUG);

// 准备 GET string

String getStr = "GET /update/?field1="+String(strTemp)+"\r\n\r\n";

//发送数据长度

cmd = "AT+CIPSEND=" + String(getStr.length()) + "\r\n";

ser.print(cmd);

if(ser.find('>')){

//发送数据

ser.print(getStr);

}

delay(15000);

}

try {

//构建服务器对象

ServerSocket ss = new ServerSocket(1234);

//构建 用户集合

list = new ArrayList<Socket>();

System.out.println("服务器准备就绪 ...");

//循环监听

while(true){

//上线用户

s = ss.accept();

//上线的人都添加到 集合中

list.add(s);

//获取 Socket IP

ip = s.getInetAddress().getHostAddress();

System.err.println( ip + " 用户上线了 , 当前在线用户为: " + list.size() + "人 !" );

//构建 发送信息线程

M2MSend send = new M2MSend(s);

send.start();

}

} catch (IOException e) {

//用户下线

list.remove(s);

System.err.println(ip + " 已下线 , 当前在线人数为: " + list.size() + " 人 !");

}

BufferedReader reader = new BufferedReader(new InputStreamReader(s.getInputStream()));

//不断的读取写出数据

while(true){

//接收数据

String info = null;

info=reader.readLine();

//如果读取信息不为空

//if((info=reader.readLine()) != null){

String sql="select comd from order where name ="+info+";";

String sql2="delete from order where name ="+info+";";

try{

Statement stmt = null;

Connection conn = null;

conn=DB.getConn();

stmt=DB.createStmt(conn);

ResultSet rs = stmt.executeQuery(sql);

while(rs.next()){

info=rs.getString("comd");

}

stmt.execute(sql2);

}

catch(Exception e){

e.printStackTrace();

}

//获取对象的输出流

PrintWriter pw;

pw = new PrintWriter(s.getOutputStream());

//写入信息

pw.println(info);

pw.flush();

System.out.println(info);

Thread.sleep(5000);

//}

}

BufferedReader is=new BufferedReader(new InputStreamReader(socket.getInputStream()));

PrintWriter os=new PrintWriter(socket.getOutputStream());

System.out.println("Client:"+is.readLine());

while(true){

if(is.readLine()!="\r"){

String temp;

temp=is.readLine();

System.out.println("Client:"+temp);

String sql="insert into esp8266 values('"+temp+"');";

try{

Statement stmt = null;

Connection conn = null;

conn=DB.getConn();

stmt=DB.createStmt(conn);

stmt.execute(sql);

}

catch(Exception e){

e.printStackTrace();

}

}







