**云南大学软件学院**

**实 验 报 告**

课程： 物联网技术 任课教师： 陈清毅 实验指导教师（签名）：

**实验七 Wifi网络数据传输实验**

1. **实验目的**

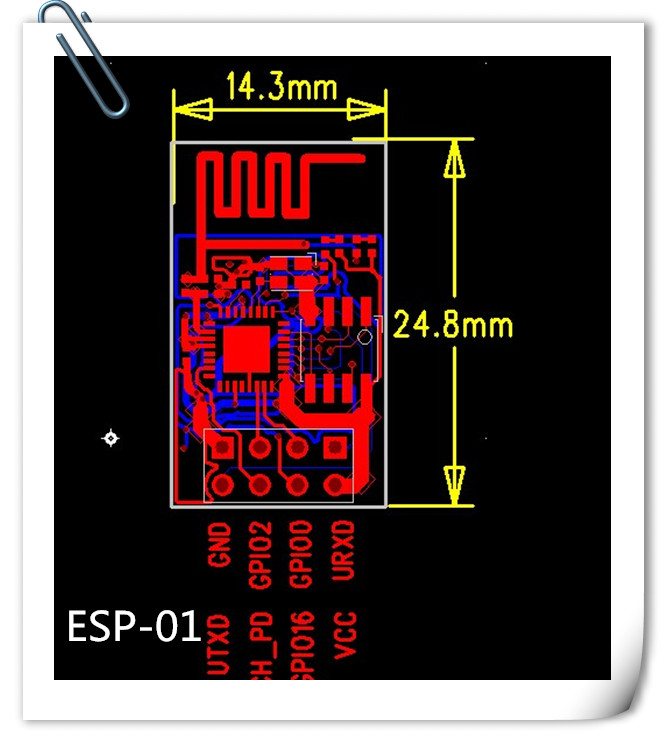
（1）学习如何使用 Arduino（树莓派）控制WIFI芯片；

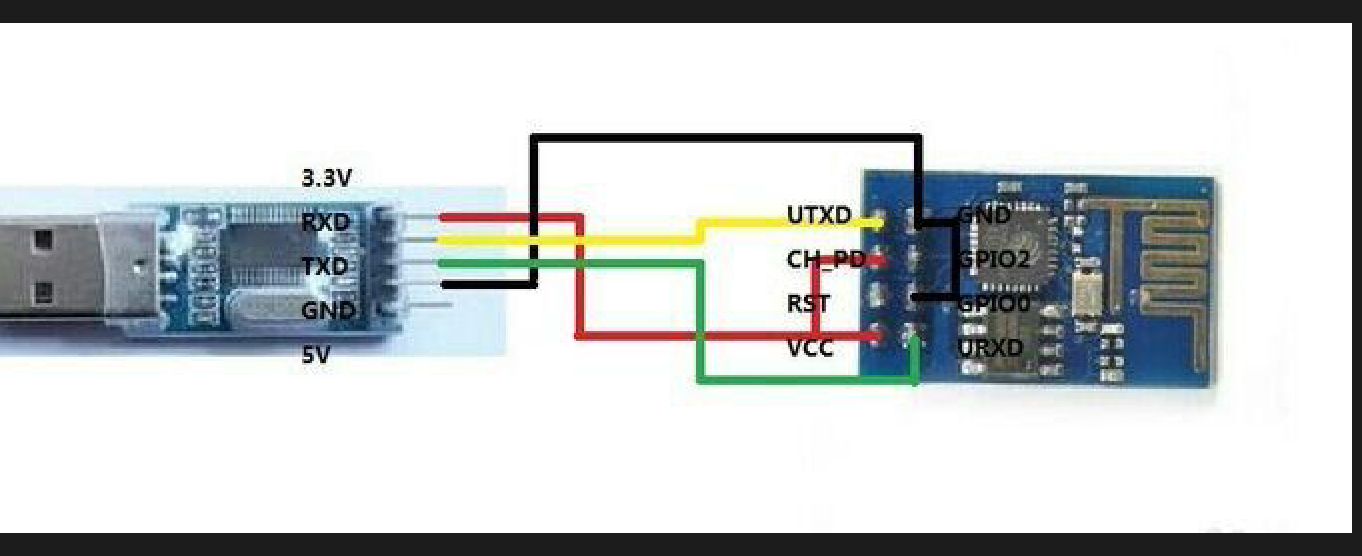
（2）学习如何进行通过Arduino将AT指令集发送给WIFI芯片；

（3）学习通过Arduino像网络上的服务器发送数据（服务器TCP代码在网盘）。

**二、实验内容**

学习如何使用 Arduino（树莓派）控制WIFI芯片

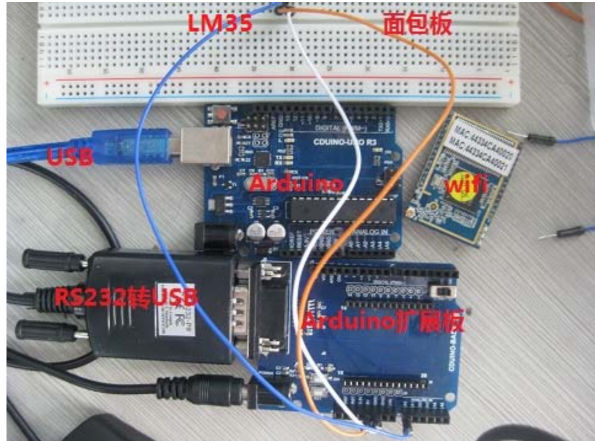




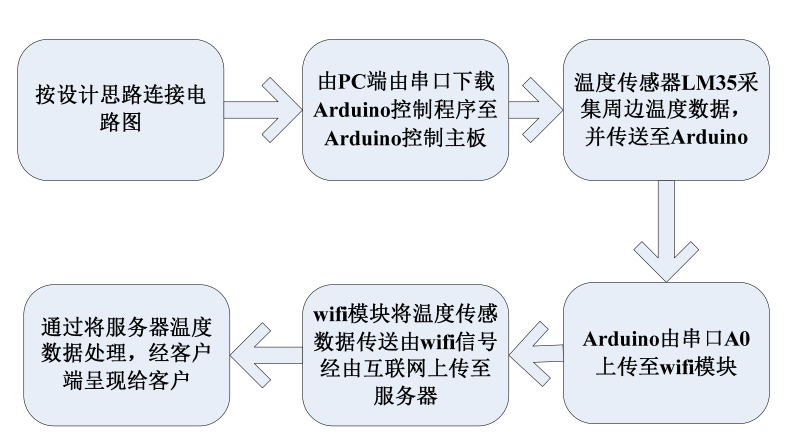
**参考接线1：**

****

**参考接线2：**

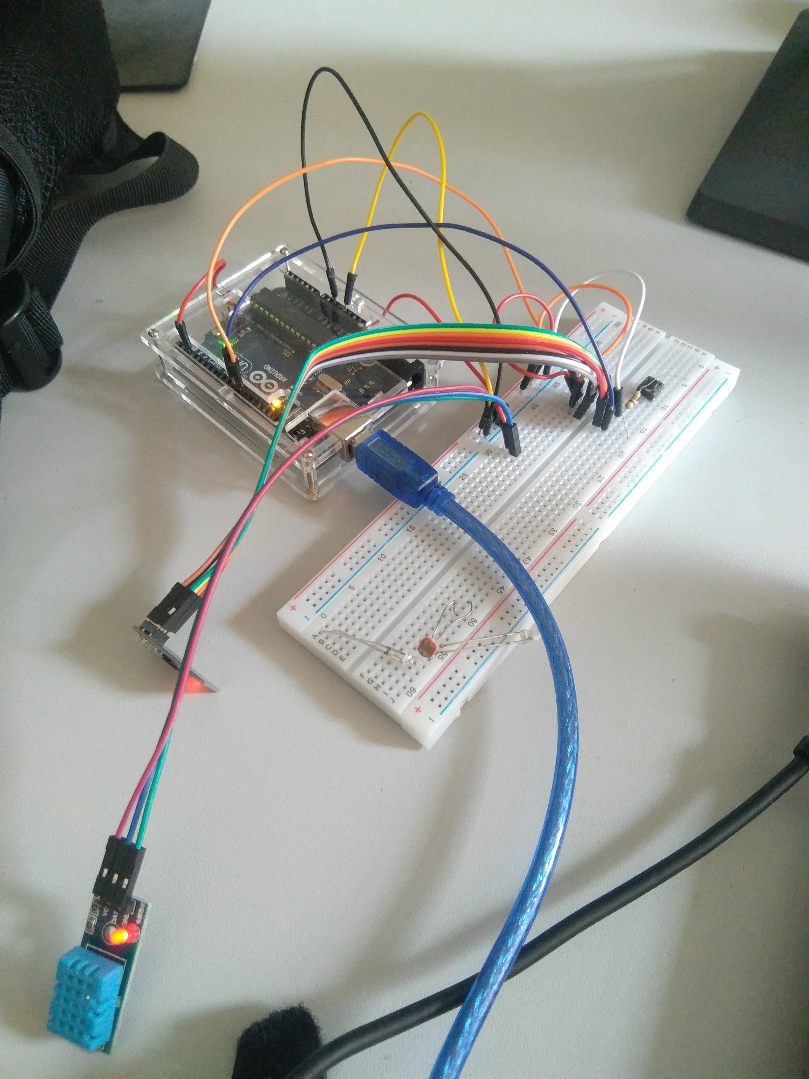


**工作思路：**

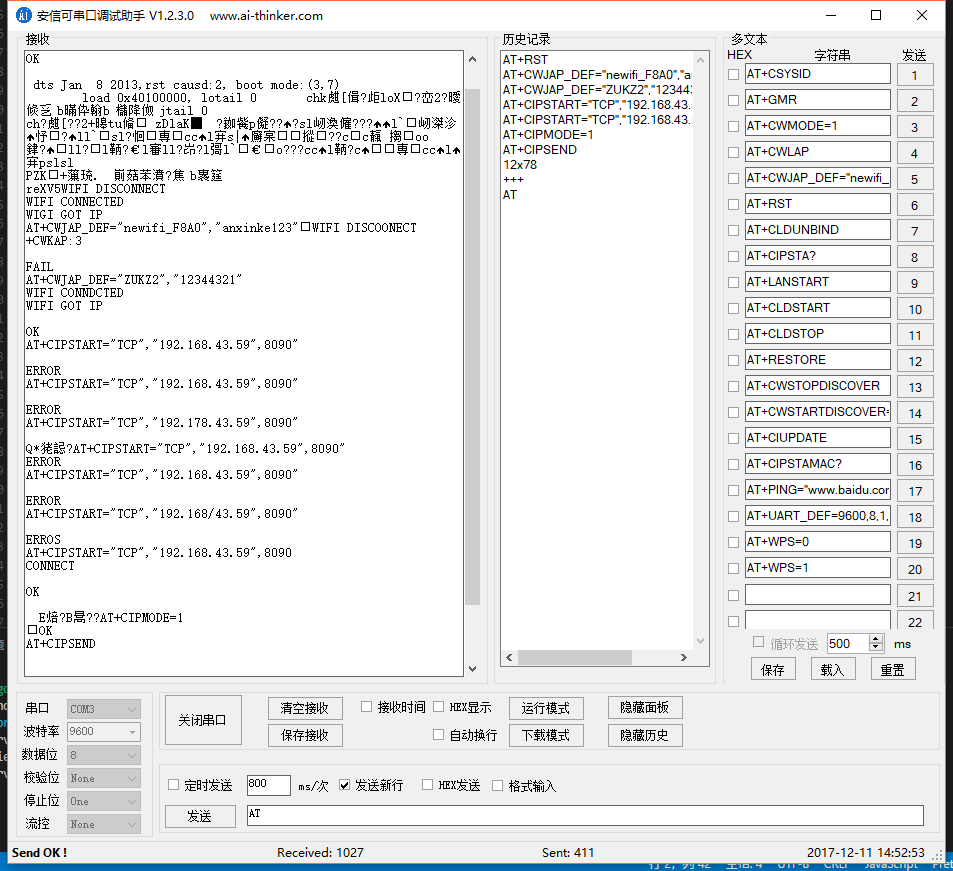


**三、关键实验步骤（请粘贴关键步骤、代码、实验结果）**

**1、按图示连接线路：**



**2、使用软串口转发配置ESP8266.**



**3、编写arduino代码，通过tcp向服务器发送数据。**

#include <SimpleDHT.h>

#include <SoftwareSerial.h>

SoftwareSerial mySerial(11, 10); // RX, TX

int c = 0;

int h = 0;

int pinDHT11 = 2;

SimpleDHT11 dht11;

void setup() {

// Open serial communications and wait for port to open:

Serial.begin(9600);

while (!Serial) {

; // wait for serial port to connect. Needed for native USB port only

}

// set the data rate for the SoftwareSerial port

mySerial.begin(115200);

mySerial.println("AT+CIPSTART=\"TCP\",\"192.168.43.59\",8090"); //连接tcp

delay(1000);

mySerial.println("AT+CIPMODE=1"); //设为透传模式

delay(500);

mySerial.println("AT+CIPSEND"); //开始传输

delay(500);

}

void loop() { // run over and over

// if (mySerial.available()) {

// Serial.write(mySerial.read());

// }

// if (Serial.available()) {

// mySerial.write(Serial.read());

// }

byte temperature = 0;

byte humidity = 0;

int err = SimpleDHTErrSuccess;

if ((err = dht11.read(pinDHT11, &temperature, &humidity, NULL)) != SimpleDHTErrSuccess) {

Serial.print("Read DHT11 failed, err="); Serial.println(err);delay(1000);

return;

} //读取dht11的温度。

String str="";

str.concat((int)temperature);

str.concat("x");

str.concat((int)humidity);//构造字符串，用”x”分割温湿度

mySerial.print(str); //发送温度到tcp服务器。

// DHT11 sampling rate is 1HZ.

delay(1500);

}

**4、编写服务器代码**

var app = require("express")();

var server = require("http").Server(app);

var io = require("socket.io")(8088);

server.listen(8087);

app.get("/", function(req, res) {

res.sendfile(\_\_dirname + "/index.html");

});

io.on("connection", function(socket) {

socket.emit("news", { hello: "world" });

socket.on("data", function(data) {

console.log(data);

});

var net = require("net");

// 创建TCP服务器

var server2 = net.createServer(function(socket2) {

console.log("client connected");

// 监听客户端的数据

socket2.on("data", function(data) {

d = data.toString().split("x");

socket.emit("news", {

c: d[0],

h: d[1]

});//监听esp8266发送的数据并通过socketio发送到网页上。

console.log("server got data from client: ", data.toString());

});

// 监听客户端断开连接事件

socket2.on("end", function(data) {

console.log("connection closed");

});

// 发送数据给客户端

socket2.write("Hello\r\n");

});

// 启动服务

server2.listen(8090, "192.168.43.59", function() {

console.log("server bound");

});

});

**5、网页端重要代码：**

var socket = io.connect('http://localhost:8088');//连接node tcp服务器、

socket.on('news', function(data) {

console.log(data);

option.series[0].data[0].value = parseFloat(data.c)

option2.series[0].data[0].value = parseFloat(data.h)

myChart.setOption(option, true);

myChart2.setOption(option2, true);

socket.emit('my other event', {

my: 'data'

});//监听数据 并动态更新页面图标。

});

**6、结果**

