一、课程设计目的及任务

1．课程设计目的

本设计是基于CC2530和DHT11温湿度传感器的温湿度采集系统。即该系统是采用Zigbee无线通信技术结合传感器，并通过运用Zigbee协议架构组建无线传感网络，来实现主从节点的数据采集和传输。同时，需要在网络层通过路由协议来进行节点间的连接以及数据的收发。总之，基于无线传感技术的无线网络传感器是一种将传感器、控制器、计算能力、通信能力完美的结合于一身的嵌入式设备。它们跟外界的物理环境交互，实时的采集信息，并且将收集到的信息通过无线传感器网络传送给远程用户。

无线网络传感器一般是由一个低功耗的微控制器(MCU)和若干个存储器、无线电/光通信装置、传感器等组件所集成的，通过传感器、动臂机构、以及通信装置和它们所处的外界物理环境进行交互。由此而引入的无线传感网络更是一种开创了新的应用领域的新兴概念和技术，广泛被应用于医疗领域、大规模环境监测、智能建筑、战场监视、智能家装、工业自动化和大区域内的目标追踪等领域。如在医院Zigbee网络可以帮助医生及时并准确的收集急诊病人的信息及相关度检查结果，从而快速准确的作出诊断；同时，对一些特殊的病人，可让其携带Zigbee终端而对其进行24小时的体温、脉搏控制。又如，在工业控制领域内，就可以很好的通过无线传感网络实现厂房内不同区域的温湿度的检测及控制，以及相关机器运转状况是否正常的有关信息，统计库存量等等。

通过课题的设计进一步巩固所学的知识，同时学习课程以外的相关知识，培养综合应用知识的能力。锻炼动手能力与实际工作能力，将所学的理论与实践结合起来。

1. 课程设计任务

本设计是以CC2530芯片结合实验箱作为主要控制核心，利用实验箱搭载该芯片对DTH11温湿度传感器采集数据，将采集到的数据从串口发送到上位机并进行数据处理，并且通过实验箱上搭载的继电器进行控制，该继电器可作为风扇或降温系统、水龙头等等，从而可以达到客户的需求。  
 本次设计主要完成以下内容：  
 (1) 能够使用该实验箱搭载的芯片作为传感器节点，利用Zigbee协议栈，将主从节点进行组网连接。

1. DHT11温湿度传感器模块可实现数据实时监测。
2. 实验箱搭载的Zigbee节点作为协调器，即主节点，能通过路由协议对各节点进行数据的接收与发送。即接收搭载有DHT11温湿度模块的Zigbee分节点发送的数据。当搭载有DHT11温湿度模块的分节点发送数据时，LED1灯闪烁为红色。
3. 协调器主节点接收到搭载DHT11温湿度模块的分节点发来的数据，并将数据通过串口线发送给上位机。
4. 上位机读取串口发来的数据且进行数据处理，而且上位机实时读取串口数据并显示在上位机界面。
5. 将串口读取到数据进行处理判断，若该数据中的温度大于客户在上位机所定的阈值，则上位机将弹出的窗口警报，提示温度超标，温度数据显示为红色，并向串口写入数据并发送到协调器主节点。协调器收到数据之后，通过组播组网发送指令给搭载有继电器的Zigbee分节点,该分节点接收到指令后，会立马打开继电器（模拟风扇或降温系统）回应；若该数据小于客户在上位机所定的阈值，则在上位机发送指令通知继电器关闭并在上位界面取消弹窗，温度数据将显示为绿色。

二、课程设计内容

1. 电路设计

本设计主要使用物联网综合教学实验箱搭载模块、DHT11温湿度传感器、继电器、Zigbee节点等元器件，使用IAR Embedded Workbench for 8051软件利用Zigbee协议栈进行对物联网综合教学实验箱编程控制。

* 1. **CC2530芯片**

CC2530芯片主要介绍以及特点如下：

CC2530是用于2.4GHz IEEE 802.15.4与Zigbee应用的Soc解决方案。内涵模块大致分三类：CPU和内存相关模块；外设、时钟和电源管理相关的模块以及射频相关的模块。

CC2530在单个芯片上整合了8051兼容微控制器、ZigBee射频(RF)前端、内存和Flash存储器等，还包含串行接口(UART)、模/数转换器(ADC)、多个定时器(Timer)、AES128安全协处理器、看门狗定时器(Watehdog Timer)、32kHz品振的休眠模式定时器、上电复位电路、掉电检测电路以及21个可编程I/O接口等外设接口单元。

CC2530主要特点如下：

高性能、低功耗、带程序预取功能的8051微控制器内核。

32KB/64KB/128KB或256KB的系统可编程Flash。

8KB在所有模式都带记忆功能的RAM。

2.4GHz IEEE 802.15.4兼容RF收发器。

优秀的接收灵敏度和强大的抗干扰性。  
具有电池监测和温度感测功能。  
集成AES安全协处理器,硬件支持的CSMA/CA功能。

精确的数字接收信号强度(RSSI)指示/链路质量指示(LQI)支持。

最高到4.5dBm的可编程输出功率。

具有8路输人和可配置分辨率的12位ADC。强大的5通道DMA。

带有两个强大的支持几组协议的UART等等。

CC2530的无线收发器是中低频接收器，通过变频器来完成发送，得到的射频信号被功率放大器发大，并往天线匹配。  
 Zigbee节点实物图如图1所示。



图1 CC2530芯片实物图

**1.2 DHT11温湿度传感器**  
1.2.1 DHT11温湿度传感器简介  
 DHT11数字温湿度传感器是一款含有已校准数字信号输出的温湿度复合传感器。传感器包括一个电阻式感湿元件和一个NTC测温元件,并与一个高性能8位单片机相连接。实物图如图2所示。

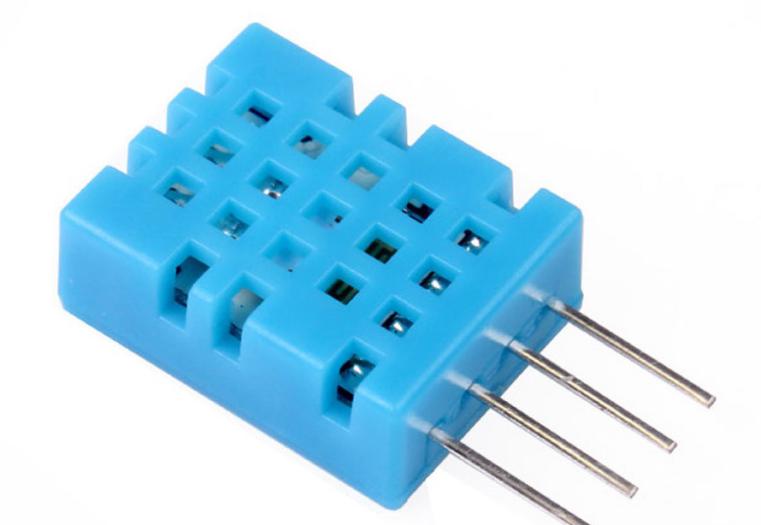


图2 DHT11温湿度传感器

1.2.2 DHT11温湿度传感器的主要特性：

（1）相对湿度和温度测量

（2）全部校准，数字输出

（3）卓越的长期稳定性

（4）无需额外部件

（5）超长的信号传输距离

（6）超低能耗

（7）引脚安装

（8）完全互换  
1.2.3 DHT11温湿度传感器测量分辨率

测量分辨率分别为8bit（温度）、8bit（湿度）。

1.2.4 串行接口(单线双向)

采用单总线数据格式，一次通讯时间4ms左右,数据分小数部分和整数部分，操作流程如下:

一次完整的数据传输为40bit，高位先出。

数据格式:8bit湿度整数数据+8bit湿度小数数据+8bi温度整数数据+8bit温度小数数据+8bit校验和

数据传送正确时校验和数据等于“8bit湿度整数数据+8bit湿度小数数据

+8bi温度整数数据+8bit温度小数数据”所得结果的末8位。

1.2.5 DTH11采集数据工作过程

DHT11从低功耗模式转换到高速模式,等待主机开始信号结束后,DHT11发送响应信号,送出40bit的数据,并触发一次信号采集，如图3所示。

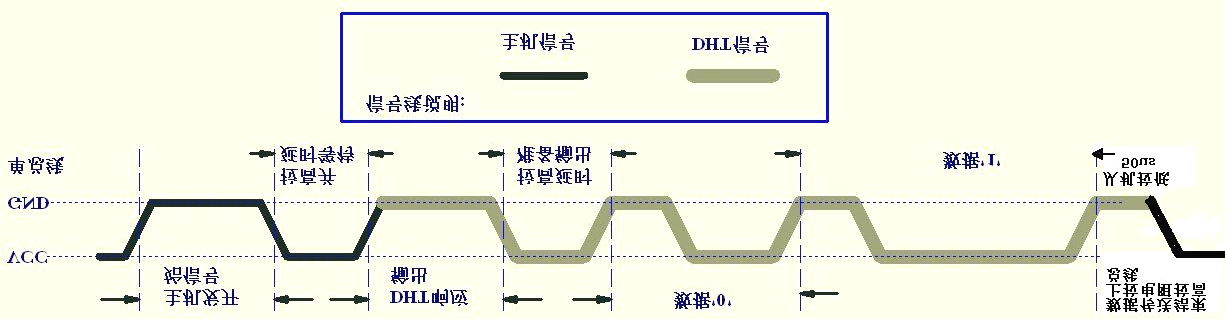


图3通讯过程图

总线空闲状态为高电平，主机把总线拉低等待DHT11响应，主机把总线拉低必须大于18毫秒，保证DHT11能检测到起始信号。DHT11接收到主机的开始信号后，等待主机开始信号结束，然后发送80us低电平响应信号。主机发送开始信号结束后,延时等待20-40us后,读取DHT11的响应信号，主机发送开始信号后，可以切换到输入模式，或者输出高电平均可，总线由上拉电阻拉高。如图4所示。

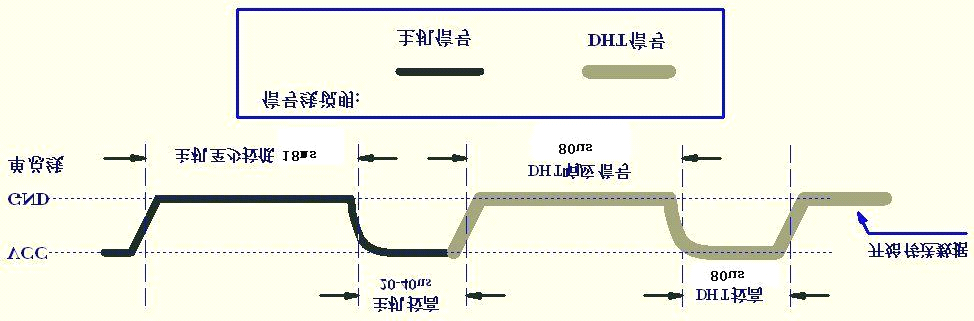


图4 高电平状态

高电平的长短定了数据位是0还是1，如果读取响应信号为高电平,则DHT11没有响应,请检查线路是否连接正常.当最后一bit数据传送完毕后，DHT11拉低总线50us,随后总线由上拉电阻拉高进入空闲状态。

1.2.5 DHT11温湿度传感器电路图

DHT11温湿度传感器电路图如图5所示。

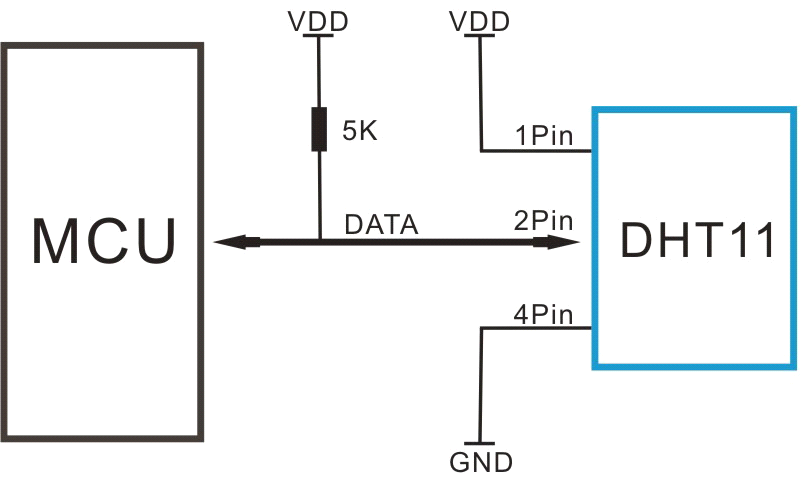


图5 DHT11温湿度传感器电路图

**1.3 继电器**

1.3.1 继电器简要介绍

（1）模块采用小家电通用的继电器，原装松乐SRD系列继电器10W次开关

（2）控制型号TTL，控制信号电压3-24V。

（3）带电源指示灯和工作指示灯（吸合亮、断开灭）

（4）控制脚加下拉电路，防止继电器误动作

（5）控制简单，模块供电，然后给控制高电平，驱动继电器吸合，低电平，继电器断开（NPN三极管）

（6）输入端

VCC:继电器模块工作电压正极

GND:继电器模块工作电压负极

IN:控制信号输入口

（7）输出端

NO:继电器常开端

COM:继电器公共端

OFF:继电器常闭端

1.3.2 继电器实物图

继电器实物图如图6所示。

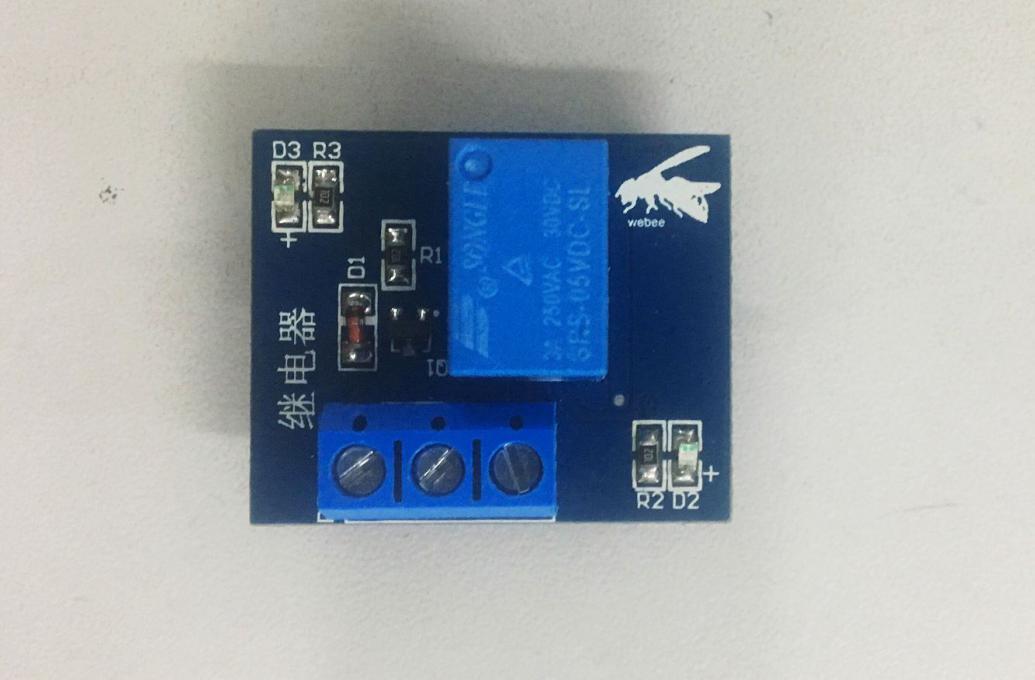


图6 继电器

1.3.3 Zigbee协议栈

ZStack由一组子层组成，每一层为其上层提供一定的特定服务：一个数据实体提供数据传输服务，一个管理实体提供管理、维护等服务。Zigbee协议栈中包括应以下各个子层：

（1）APP（Application Programming）：应用层目录，这是用户创建各种不同工程的区域，在这个目录中包含了应用层的内容和这个项目的主要内容，在协议栈里面一般是以操作系统的任务实现的。

（2）HAL（Hardware(H/W)Abstraction Layer）：硬件层目录，包含有与硬件相关的配置和驱动及操作函数。ZigBee硬件层通过射频固件和射频硬件提供了一个从MAC层以硬件层无线信道的接口。在硬件层中，包含一个硬件层管理实体（PLME），该实体通过调用硬件层的管理功能函数，为硬件层管理服务提供其接口，同时，还负责维护由硬件层所管理的目标数据库，该数据库包含有硬件层个域网络的基本信息。

（3）MAC：介质接入控制子层，包含了MAC层的参数配置文件及其MAC的

LIB库的函数接口文件。实现的功能有：

① 能产生网络信标。

② 支持PAN的连接和断开连接。

③ 同信标保持同步。

④ 在对等的MAC实体之间提供一个可靠的通信链路。

⑤ 处理和维护GTS机制。

⑥ 信道接入采用CSMA-CA接入机制。

⑦ 支持设备的安全性。介质访问控制层(MAC)帧被称为MAC协议数据单元(MPDU)，其长度不超过127个字节。它具有四种不同的帧形式，即信标帧、数据帧、确认帧和命令帧。

（4）MT（Monitor Test）：实现通过串口可控各层，与各层进行直接交互。

（5）NWK（ZigBee Network Layer）：网络层目录，含网络层配置参数文件及网络层库的函数接口文件，APS层库的函数接口。

（6）OSAL（Operating System(OS)Abstraction Layer）：Z-stack协议栈的操作系统。

（7）Profile：AF（Application work）层目录，包含AF层处理函数文件。

（8）Security：安全层目录，安全层处理函数，比如加密函数等。

（9）Services：地址处理函数目录，即包括着地址模式的定义及地址处理函数。

（10）0Tools：工程配置目录，包括空间划分及ZStack相关配置信息。

（11）ZDO（ZigBee Device Objects）：ZDO目录。

（12）ZMac：MAC层目录，包括MAC层参数配置及MAC层LIB库函数回调处理函数。

（13）ZMain：主函数目录，包括入口函数及硬件配置文件。

（14）Output：输出文件目录，这个EW8051 IDE自动生成的。

**1.4 硬件系统的结构图**

1.4.1 硬件系统的结构图

通过几个上述模块介绍，包含有该CC2530 Zigbee实验箱平台搭载的Zigbee节点若干个，节点上搭载DTH11温湿度传感器、继电器等，综合得本次设计的整个硬件系统，如图7所示。



图7 硬件系统的结构图

**1.5 节点组网**

硬件系统采用组播组网，把主节点即协调分别和搭载有DHT11模块的分节点、搭载有继电器的分节点进行组网通信。组播通信原理图如图8所示。

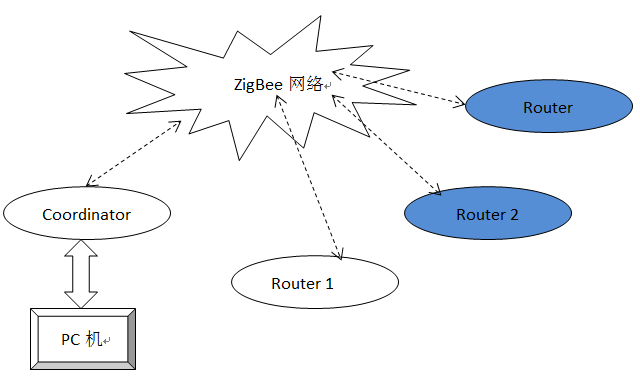


图8 组播通信原理图

1. 软件设计与实现

**3.1系统程序流程图**

整个系统的整体程序流程图，如图9所示。



图9 系统程序总流程图

**3.2 协调器工作流程图**

协调器作为Zigbee主节点，主要负责通过组播对分节点进行组网，而且对分节点的消息接收与发送，与上位机进行串口通信等等。

协调器工作流程图如图10所示。

协调器通过串口与上位机通信流程图如图11所示。

协调器向继电器发消息指令流程图如图12所示。

图10 协调器工作流程图 图11 与上位机通信流程图 图12 发送消息流程图

协调器工作关键代码实现如下：

void GenericApp\_MessageMSGCB(afIncomingMSGPacket\_t\*pkt)

{

unsigned char buffer[4]="";

switch(pkt->clusterId)

{

case GENERICAPP\_CLUSTERID:

osal\_memcpy(buffer,pkt->cmd.Data,4);//将接收到的数据拷贝到缓冲区

HalUARTWrite(0,buffer,2);//往串口发送温度数据

HalUARTWrite(0,&buffer[2],2);//往串口发送湿度数据

HalUARTWrite(0,"\r\n",2);

break;

}

LED1OFF;

}

读取串口数据关键代码如下：

void GenericApp\_SendTheMessage\_to\_relay(uint8 SerialPort,uint8 Events)

{

unsigned char uartbuf;//从串口接收数据、则发送消息（继电器）

if(Events&(HAL\_UART\_RX\_FULL|HAL\_UART\_RX\_ABOUT\_FULL|HAL\_UART\_RX\_TIMEOUT))//接收到串口数据

{

HalUARTRead(0,&uartbuf,1);//读取数据并存放到uartbuf数组中

if(uartbuf=='1')

{

unsigned char\*theMessageData="111";//消息发送函数继电器

afAddrType\_t my\_DstAddr;

my\_DstAddr.addrMode=(afAddrMode\_t)AddrGroup;

my\_DstAddr.endPoint=GENERICAPP\_ENDPOINT;

my\_DstAddr.addr.shortAddr=GenericApp\_Group.ID;//网络地址为组ID

AF\_DataRequest(&my\_DstAddr,&GenericApp\_epDesc,6,3,

theMessageData,

&GenericApp\_TransID,AF\_DISCV\_ROUTE,AF\_DEFAULT\_RADIUS);

} //发送命令给继电器6为命令即case6

if(uartbuf=='0')

{

unsigned char\*theMessageData="000";//消息发送函数继电器

afAddrType\_t my\_DstAddr;

my\_DstAddr.addrMode=(afAddrMode\_t)AddrGroup;

my\_DstAddr.endPoint=GENERICAPP\_ENDPOINT;

my\_DstAddr.addr.shortAddr=GenericApp\_Group.ID;//网络地址是组ID

AF\_DataRequest(&my\_DstAddr,&GenericApp\_epDesc,6,3, //3长度

theMessageData, //内容

&GenericApp\_TransID,AF\_DISCV\_ROUTE,AF\_DEFAULT\_RADIUS);

}

}

}

**3.3 DHT11温湿度传感器**

使用该模块时需要在IAR软件中编程使用，并初始化模块。DHT11温湿度传感器模块，流程图如图13所示。



图13 DHT11温湿度传感器

DHT11温湿度传感器模块实现关键代码如下：

unsigned char dht11\_temp(void)

{return sTemp;} //返回温度数据unsigned char dht11\_humidity(void)

{return sHumidity;} //返回湿度数据

void dht11\_update(void){

int flag=1;

unsigned char dat1,dat2,dat3,dat4,dat5,ck;

dht11\_io\_init();

COM\_CLR; //主机拉低18ms

halWait(18);COM\_SET;flag=0;

while(COM\_R&&++flag);

if(flag==0)return;//主机设为输入判断从机响应信号

//总线由上拉电阻拉高主机延时20us

flag=0;//判断从机是否有低电平响应信号如不响应则跳出，响应则向下运行

while(!COM\_R&&++flag);

if(flag==0)return;

flag=0;

while(COM\_R&&++flag);

if(flag==0)return;

dat1=dht11\_read\_byte();//读取第一组8bit位湿度整数数据

dat2=dht11\_read\_byte();//第二组8bit湿度小数数据

dat3=dht11\_read\_byte();//第三组8bit温度整数数据

dat4=dht11\_read\_byte();//第四组8bit温度小数数据

dat5=dht11\_read\_byte();

//第五组校验和与数据传送正确时校验和，等于前面四组的和

ck=dat1+dat2+dat3+dat4;

//验证第五组数据是否与前四组数据之和相等，

if(ck==dat5){

sTemp=dat3; //如果相等则将温湿度进行赋值

sHumidity=dat1;

}

}

**3.4 继电器工作流程图**

继电器刚开始是处于关闭状态，当串口数据发送到协调器，协调器发送指令给继电器时，继电器才被触发打开，或者触发关闭。继电器工作流程图如图14所示。



图14 继电器工作流程图

继电器工作实现关键代码如下：

void GenericApp\_MessageMSGCB(afIncomingMSGPacket\_t \*pkt) //消息处理函数(接受协调器发来的数据)

{

unsigned char\* recvbuf;

osal\_memcpy(recvbuf,pkt->cmd.Data,3);

switch(pkt->clusterId) //获取协调器发送的命令2

{

case 6:

if(osal\_memcmp(recvbuf,"111",3))//取数据//消息处理，若真并作出应答

{

HalLedBlink(HAL\_LED\_1,0,50,150); //LED1闪烁

HalLedBlink(HAL\_LED\_2,0,50,150);

OPEN\_Relay; //打开继电器

}

if(osal\_memcmp(recvbuf,"000",3))

{

HalLedBlink(HAL\_LED\_1,1,0,0); //LED1关

HalLedBlink(HAL\_LED\_2,1,0,0); //LED2关

CLOSE\_Relay; //关闭继电器

}

break;

}

LED1OFF;

}

**3.5上位机工作流程图**

上位机通过串口线与实验箱的协调器连接通信，实时监测与处理协调器发来的数据并显示在上位机界面，若收到的温度数据大于阈值则往串口发送数据1，并弹出警报窗口，且显示的温度数据为红色；否则往串口发送数据0，取消弹出窗口，且显示的温度数据为绿色。显示湿度数据一直未蓝色。上位机工作流程图如图15所示。



图15 上位机工作流程图

上位机工作关键代码实现如下：

读取串口数据代码：

String PortData = serialPort.ReadLine(); //读取串口数据

string temperature = temper.Substring(0, 2); //取串口数据前两位

string humidity = temper.Substring(2, 2); //取串口数据后两位

textBox1.Text += "温度：" + temperature + "℃";

textBox2.Text += "湿度：" + humidity + "%";

往串口写数据代码：：

if(int.Parse(temperature)>=int.Parse(comboBoxValue.Text))

{

textBox1.ForeColor=Color.Red;

serialPort.Write("1");

TimingMessageBox messageBox=new TimingMessageBox("超标温度："+temperature+"℃",1); //警报弹窗2秒

messageBox.ShowDialog();

}else

{ serialPort.Write("0"); }

三、调试及结论

**1.调试**

1.1 打开上位机运行程序。选好串口号、波特率等并点击打开串口。每5秒将收到一次温湿度数据。调试结果截图1，如图16所示。

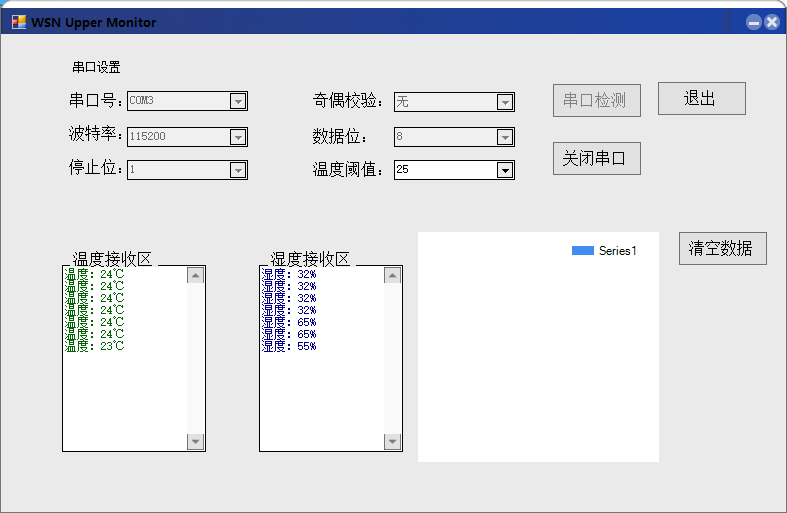


图16 调试截图1

1.2 调试结果截图2，如图17所示。

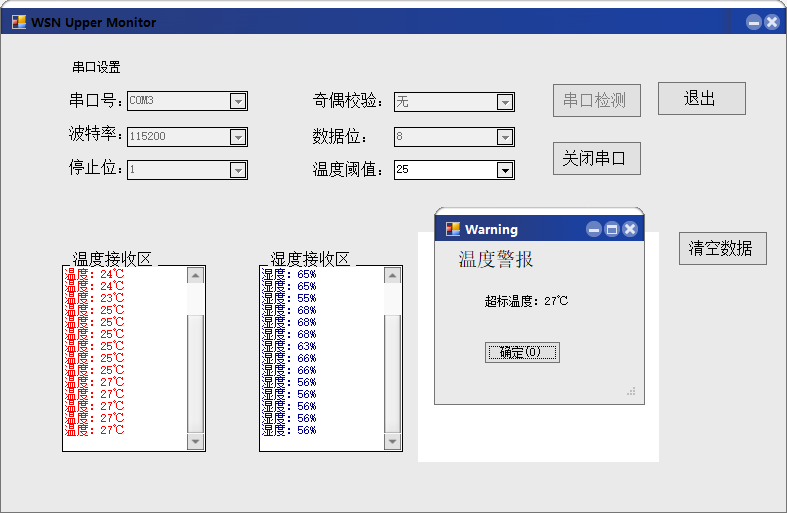


图17 调试截图2

1.3 调试结果截图3，如图18所示。

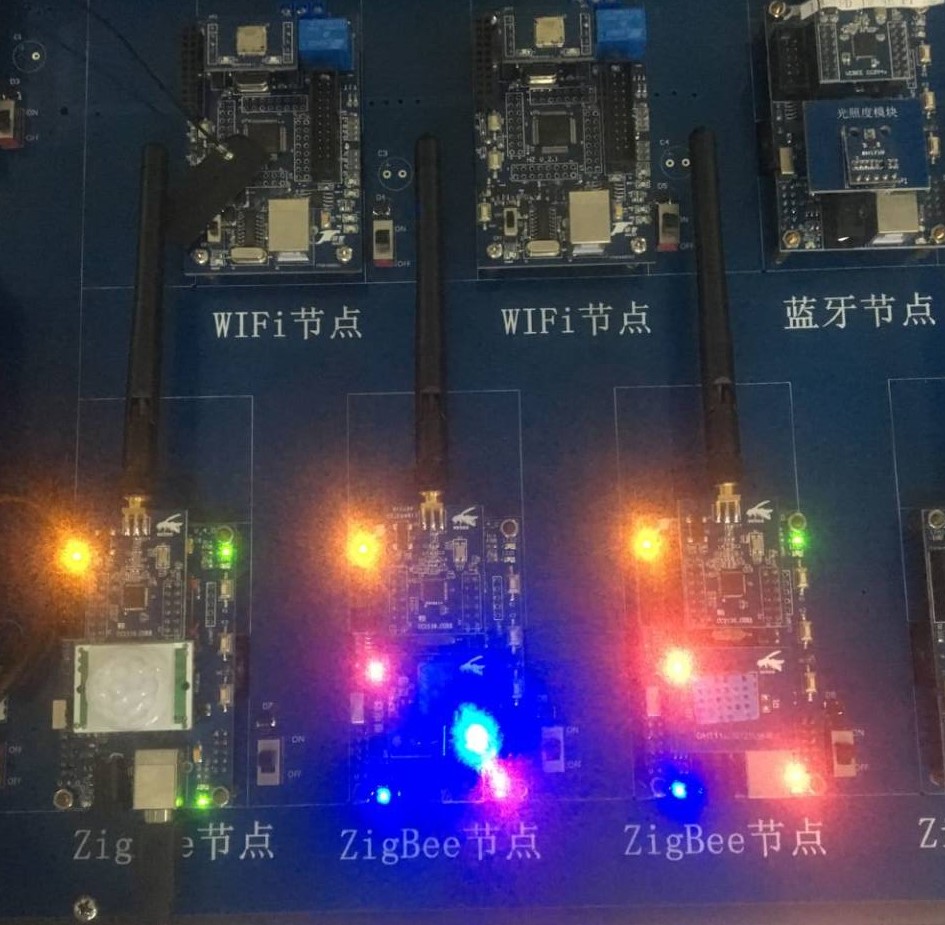


图18 调试截图3

**2.结论**

通过本次课程设计，不断改进与测试，最终实现了本设计系统的主要功能。在测试截图1中，在串口正确的配置下并打开串口，将每5秒得到温湿度数据，进行实时监测。测试温度阈值为25℃，实时监测温度小于阈值，所以在上位机不弹出窗口。在调试图2中，实时监测的温度值大于所设定的阈值，所以在上位机弹出警报窗口，显示数据将为红色。并往串口写入1，协调器将发指令给继电器且命令继电器打开，否则小于所设定阈值，则上位机取消弹窗，显示数据为绿色，并往串口写入0，协调器将发指令给继电器，命令继电器关闭。调试截图3为节点工作状态。从左到右，第一节点为协调器，第二节点为继电器，第三为搭载有DHT11温湿度传感器节点。如组网成功，三者黄灯常亮，第一第三节点绿灯闪烁。第二节点的效果为继电器处于打开状态，最终效果见图调试截图。

参考文献

1. 陈海宴.51单片机原理及应用第三版[M].北京：北京航空航天大学出版社，

2017.5.

[2] 许毅、陈立家.无线传感器网络技术原理及应用[M].北京：清华大学出版社，2019.4.

# [3] [YuQiao0303](https://www.jianshu.com/u/c9336d741115).DHT11温湿度传感器[EB/OL].(2019-03-12)[2019-12-22].

https://www.jianshu.com/p/bc38be3e2df1

# 张涛.[基于CC2530的温度监测模块设计与应用](http://cdmd.cnki.com.cn/Article/CDMD-11902-1012428148.htm" \t "http://cdmd.cnki.com.cn/Article/_blank)[D].南昌大学.2012年

1. 宋甜.[基于ZigBee的无线传感器网络的研究](http://cdmd.cnki.com.cn/Article/CDMD-10212-1012409156.htm" \t "http://cdmd.cnki.com.cn/Article/_blank)[D].黑龙江大学.2012年