

# 中国地质大学

CHINA UNIVERSITY OF GEOSCIENCES

北京 · BEIJING

## 课程实验报告

课程名称: 八路温度采集循环显示系统

班 级:	<u>10041532</u>
学 号:	<u>1004153213</u>
姓 名:	<u>鲁家峰</u>
指导教师:	<u>李卓, 刘伟</u>
报告日期:	<u>2018 年 7 月 20 日</u>

## 摘 要

### 暑期任务：

制作一个八路温度采集循环显示系统，循环显示每路信号的温度值。同时将结果发送到计算机，利用 labview 设计计算机显示系统，对温度进行显示。

### 步骤：

1、利用 Altium Designer 软件完成八路温度采集系统硬件电路的设计，包括画原理图和 PCB 板；

2、完成电路的焊接；

3、编写单片机程序，实现温度读取，数据转换，数码显示，计算机显示

### 技术指标：

1、温度显示精确到  $0.1^{\circ}\text{C}$ ；

2、显示温度范围  $-55^{\circ}\text{C}$ — $+125^{\circ}\text{C}$ ；

3、数码管需循环显示当前的测温通道及相应温度；

4、电路布局整体美观、合理；

5、电路焊接工艺较好，不能出现虚焊、短路问题；

6、Labview 显示系统界面美观，功能完整。

# 目 录

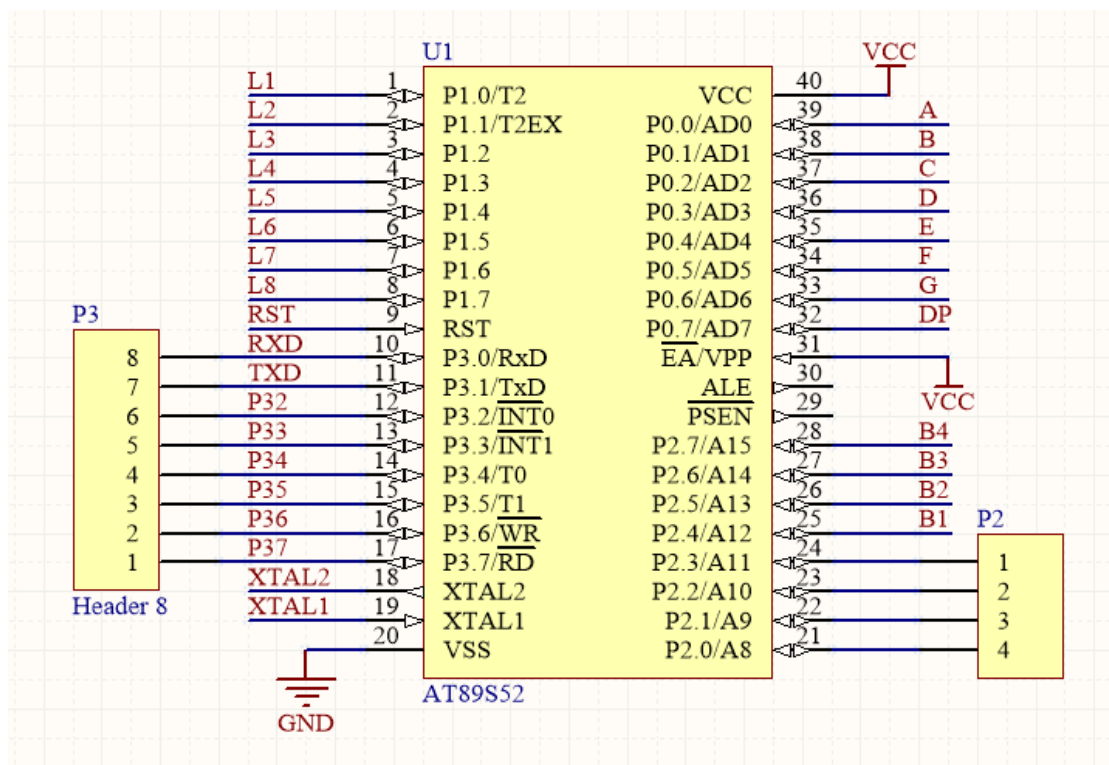
摘 要 .....	2
目 录 .....	1
正 文 .....	1
一、 电路设计 .....	1
二、 原理图 .....	3
三、 焊接 .....	3
四、 DS18B20 温度传感器 .....	4
五、 单片机程序 .....	6
六、 LabView 系统设计 .....	17
七、 总结 .....	18

# 正文

## 一、电路设计

AT89S52 是美国 ATMEL 公司生产的低电压，高性能 CMOS8 位单片机，可提供以下标准功能：4K 字节闪存，128 字节内部 RAM，32 个 I/O 口线，两个 16 位定时 / 计数器，一个 5 向量两级中断结构，一个全双工串行通信口，片内振荡器及时钟电路。同时，80C51 可降至 0HZ 的静态逻辑操作，并支持两种软件可选的节电工作模式。空闲方式停止 CPU 的工作，但允许 RAM，定时 / 计数器，串行通信口及中断系统继续工作。掉电方式保存 RAM 中的内容，但振荡器停止工作并禁止其它所有部件工作直到下一个硬件复位。

首先要根据电路设计通过 Altium Designer 软件画好原理图。



P0.0-P0.7 用来驱动数码管 a-h 的显示，采用共阳极显示。

P1.0-P1.7 口与温度传感器 DS18B20 进行连接，完成数据收发功能；

P2.4-P2.7 用来对数码管中的四位显示进行选择。

P2.0-P2.3 备用。

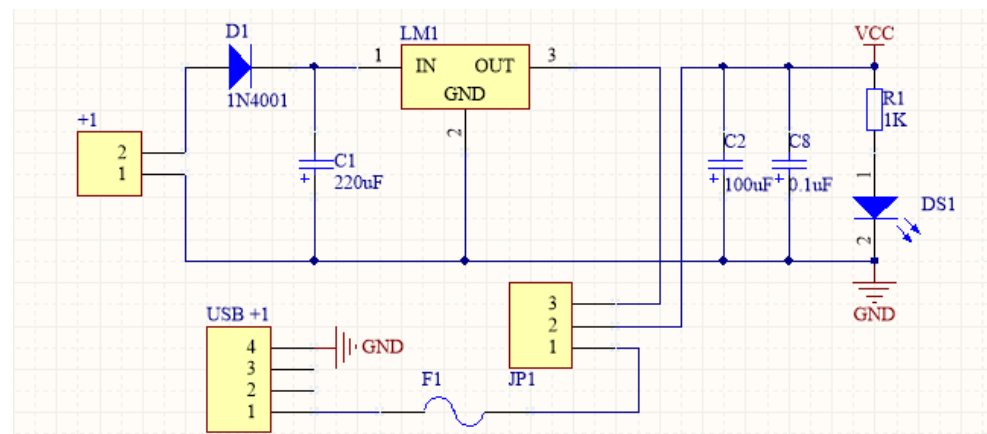
P0 和 P2 口分配给显示模块数码管的段选和位选。

P3.0 与 P3.1 口用作第二功能即串口数据收发；

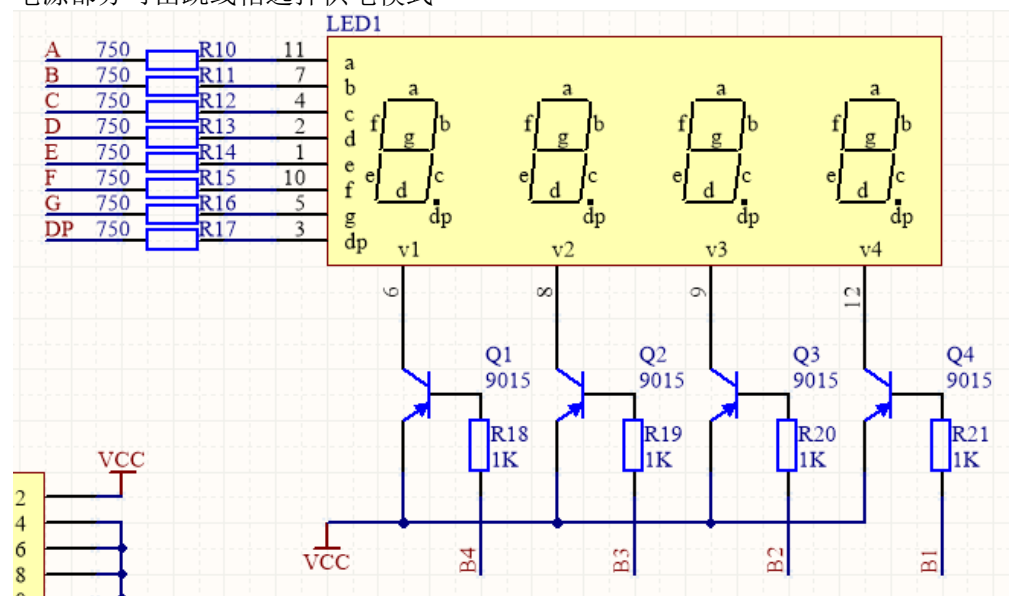
RST：复位输入当振荡器工作时，RST 引脚出现两个机器周期以上高电平将使单片机复位。

XTAL1：振荡器反相放大器及内部时钟发生器的输入端。

XTAL2 :振荡器反相放大器的输出端。



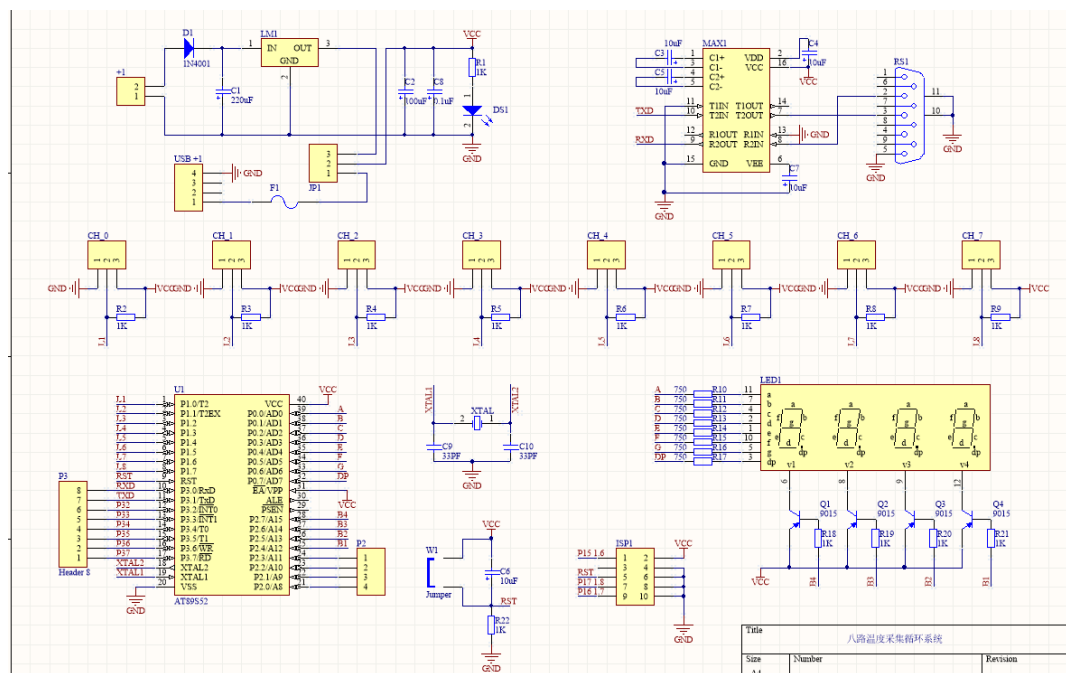
电源部分可由跳线帽选择供电模式



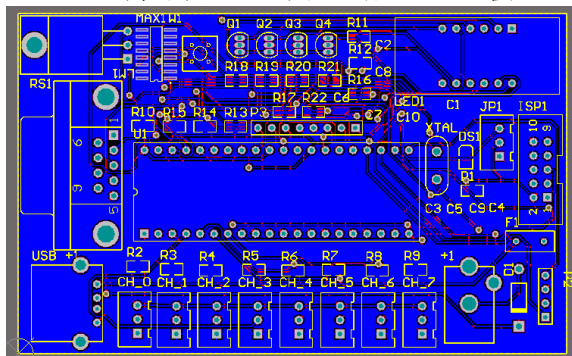
数码管采用的是共阳极接法

<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
0xc0	0xf9	0xa4	0xb0	0x99
<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>
0x92	0x82	0xf8	0x80	0x90

## 二、原理图



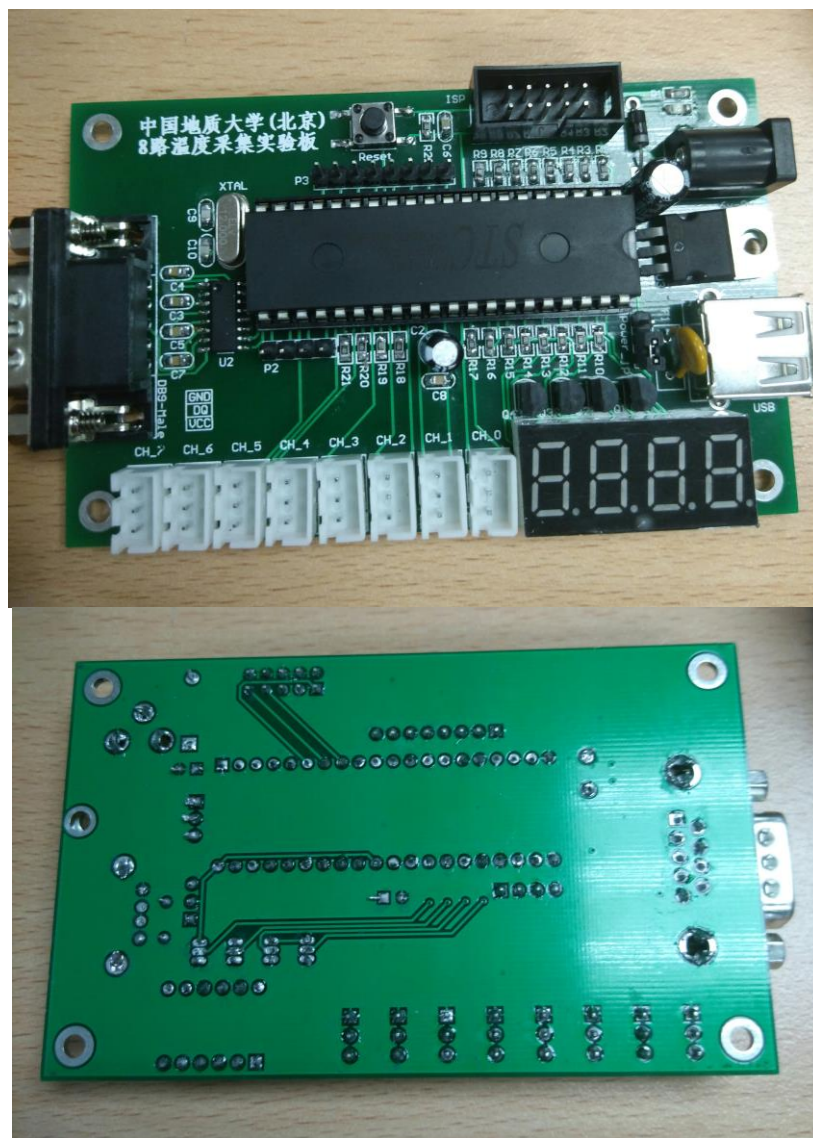
通过画好的原理图生成 PCB 板，将控件合理放置后，布线敷铜。



## 三、焊接

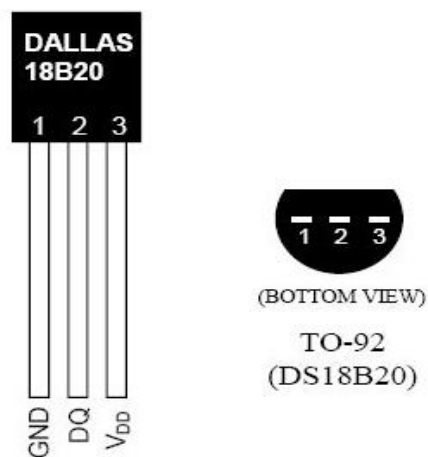
焊接过程中，焊接技术很重要。元器件安装焊接质量的好坏，将直接影响电路性能，电解电容，二极管正负级性不能接反(长管脚接正极、短管脚接负极)，最后用万用表测试避免虚焊，短路。

焊接结果展示：



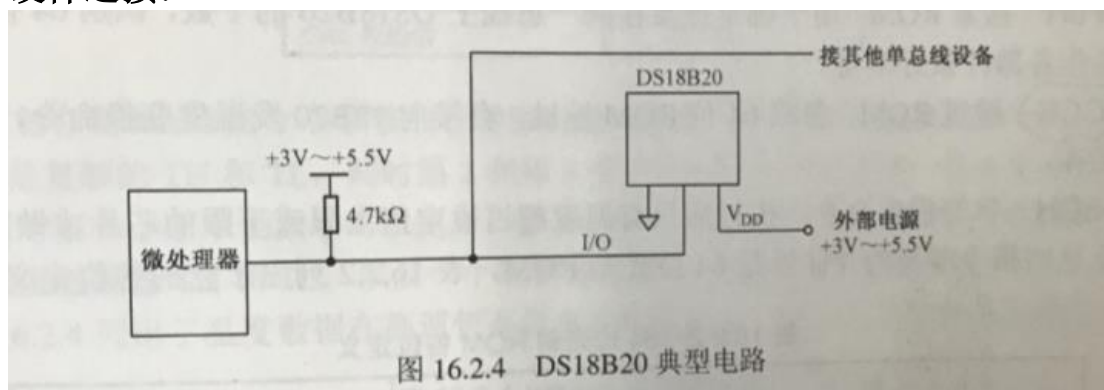
#### 四、DS18B20 温度传感器

温度传感器是各种传感器中最常用的一种，早期使用的是模拟温度传感器，如热敏电阻，随着环境温度的变化，它的阻值也发生线性变化，用处理器采集电阻两端的电压，然后根据某个公式就可以计算出当前环境温度。随着科技的进步，现代的温度传感器已经走向数字化，外形小，接口简单，广泛应用在生产实践的各个领域，为我们的生活提供便利。随着现代仪器的发展，微型化、集成化、数字化成为传感器发展的一个重要方向。美国 DALLAS 半导体公司推出的数字化温度传感器 DS18B20 采用单总线协议，即与单片机接口仅需占用一个 I/O 端口，无须任何外部元件，直接将环境温度转化成数字信号，以数字码方式串行输出，从而大大简化了传感器



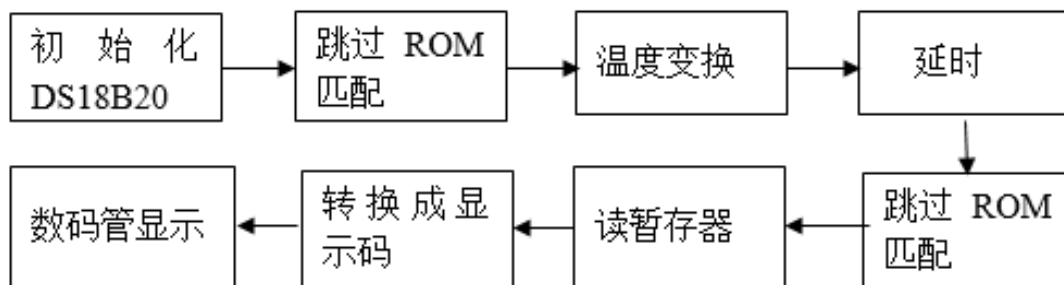
与微处理器的接口。

### 硬件连接：



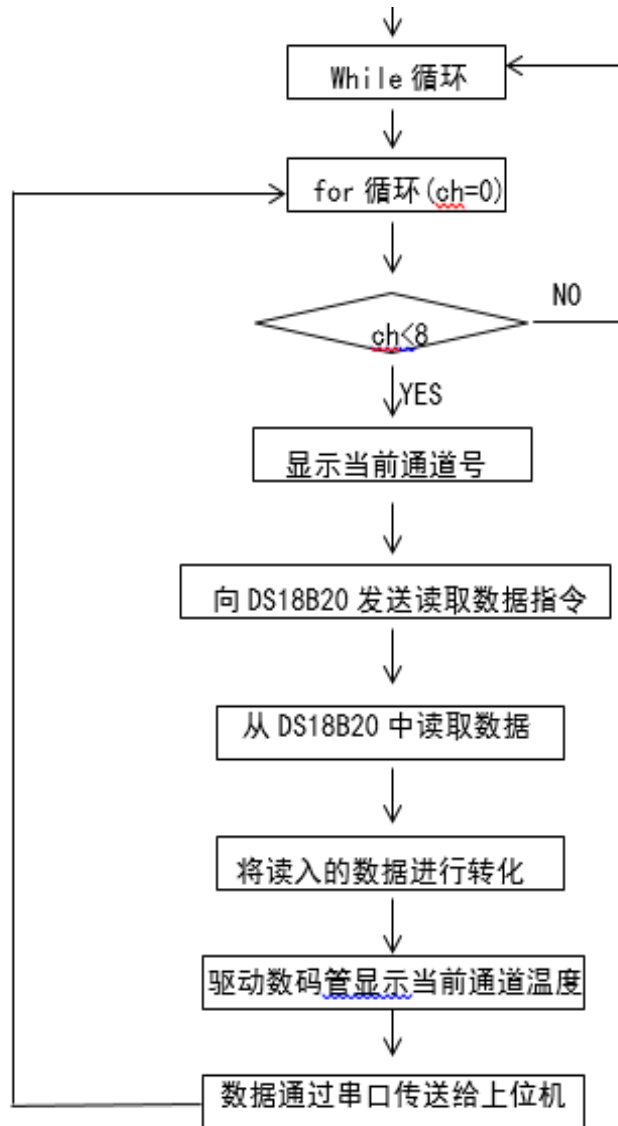
DS18B20 采用单条信号线，既可以传输时钟，又可传输数据，而且数据传输是双向的，因而这种单总线技术具有线路简单，硬件开销少，成本，便于总线扩展和维护等优点。单总线通常要求外接一个约为  $5\text{K}\Omega$  的上拉电阻。

### 温度传感器使用流程图：





## 五、单片机程序



```

#include <reg52.h>
#include <stdio.h>
#include <math.h>
#include <intrins.h>
#define uchar unsigned char
#define uint unsigned int
#define fosc 12000000 //
uint temp;
float f_temp;
int ch,zf,w0,w1,w2,w3,time=0,num=0;
uchar rbuff,sbuff,flag;
sbit B4=P2^4;sbit B3=P2^5;sbit B2=P2^6;sbit B1=P2^7;
sbit L1=P1^0;sbit L2=P1^1;sbit L3=P1^2;sbit L4=P1^3;sbit L5=P1^4;sbit L6=P1^5;sbit
L7=P1^6;sbit L8=P1^7;
unsigned char code table[]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90};
  
```

```
void delay(uint z)
{
    uint x,y;
    for(x=z;x>0;x--)
        for(y=110;y>0;y--);
}
void dsreset(void)//DS18B20 复位，初始化函数
{
    uint i;
    P1=0x00;
    i=103;
    while(i>0)i--;
    P1=0xff;
    i=4;
    while(i>0)i--;
}
bit tempreadbit(int ch)    //读 1 位数据函数
{
    uint i;
    bit dat;
    switch(ch)
    {
        case 0:
            L1=0;i++;
            L1=1;i++;i++;
            dat=L1;
            i=8;while(i>0)i--;
            return (dat);
        break;
        case 1:
            L2=0;i++;
            L2=1;i++;i++;
            dat=L2;
            i=8;while(i>0)i--;
            return (dat);
        break;
        case 2:
            L3=0;i++;
            L3=1;i++;i++;
            dat=L3;
            i=8;while(i>0)i--;
            return (dat);
        break;
        case 3:
            L4=0;i++;
            L4=1;i++;i++;
            dat=L4;
```

```
        i=8;while(i>0)i--;
        return (dat);
    break;
    case 4:
        L5=0;i++;
        L5=1;i++;i++;
        dat=L5;
        i=8;while(i>0)i--;
        return (dat);
    break;
    case 5:
        L6=0;i++;
        L6=1;i++;i++;
        dat=L6;
        i=8;while(i>0)i--;
        return (dat);
    break;
    case 6:
        L7=0;i++;
        L7=1;i++;i++;
        dat=L7;
        i=8;while(i>0)i--;
        return (dat);
    break;
    case 7:
        L8=0;i++;
        L8=1;i++;i++;
        dat=L8;
        i=8;while(i>0)i--;
        return (dat);
    break;
}
}
uchar tempread(int ch)    //读 1 字节
{
    uchar i,j,dat;
    dat=0;
    for(i=1;i<=8;i++)
    {
        j=tempreadbit(ch);
        dat=(j<<7)|(dat>>1);
    }
    return(dat);
}
void tempwritebyte(int ch,uchar dat)    //向 DS18B20 写一个字节
{
    uint i=0;
```

```
uchar j;
bit testb;
switch(ch)
{
    case 0:
        for(j=1;j<=8;j++)
        {
            testb=dat&0x01;
            dat=dat>>1;
            if(testb)    //写 1
            {
                L1=0;
                i++;i++;
                L1=1;
                i=8;while(i>0)i--;
            }
            else
            {
                L1=0;    //写 0
                i=8;while(i>0)i--;
                L1=1;
                i++;i++;
            }
        }
        break;
    case 1:
        for(j=1;j<=8;j++)
        {
            testb=dat&0x01;
            dat=dat>>1;
            if(testb)    //写 1
            {
                L2=0;
                i++;i++;
                L2=1;
                i=8;while(i>0)i--;
            }
            else
            {
                L2=0;    //写 0
                i=8;while(i>0)i--;
                L2=1;
                i++;i++;
            }
        }
        break;
    case 2:
```

```
        for(j=1;j<=8;j++)
        {
            testb=dat&0x01;
            dat=dat>>1;
            if(testb)    //写 1
            {
                L3=0;
                i++;i++;
                L3=1;
                i=8;while(i>0)i--;
            }
            else
            {
                L3=0;    //写 0
                i=8;while(i>0)i--;
                L3=1;
                i++;i++;
            }
        }
    }
    break;
case 3:
    for(j=1;j<=8;j++)
    {
        testb=dat&0x01;
        dat=dat>>1;
        if(testb)    //写 1
        {
            L4=0;
            i++;i++;
            L4=1;
            i=8;while(i>0)i--;
        }
        else
        {
            L4=0;    //写 0
            i=8;while(i>0)i--;
            L4=1;
            i++;i++;
        }
    }
}
break;
case 4:
    for(j=1;j<=8;j++)
    {
        testb=dat&0x01;
        dat=dat>>1;
        if(testb)    //写 1
```

```
        {
            L5=0;
            i++;i++;
            L5=1;
            i=8;while(i>0)i--;
        }
    else
    {
        L5=0;          //写 0
        i=8;while(i>0)i--;
        L5=1;
        i++;i++;
    }
}
break;
case 5:
    for(j=1;j<=8;j++)
    {
        testb=dat&0x01;
        dat=dat>>1;
        if(testb)      //写 1
        {
            L6=0;
            i++;i++;
            L6=1;
            i=8;while(i>0)i--;
        }
        else
        {
            L6=0;          //写 0
            i=8;while(i>0)i--;
            L6=1;
            i++;i++;
        }
    }
break;
case 6:
    for(j=1;j<=8;j++)
    {
        testb=dat&0x01;
        dat=dat>>1;
        if(testb)      //写 1
        {
            L7=0;
            i++;i++;
            L7=1;
            i=8;while(i>0)i--;
```

```
        }
        else
        {
            L7=0;          //写 0
            i=8;while(i>0)i--;
            L7=1;
            i++;i++;
        }
    }
    break;
case 7:
    for(j=1;j<=8;j++)
    {
        testb=dat&0x01;
        dat=dat>>1;
        if(testb)        //写 1
        {
            L8=0;
            i++;i++;
            L8=1;
            i=8;while(i>0)i--;
        }
        else
        {
            L8=0;          //写 0
            i=8;while(i>0)i--;
            L8=1;
            i++;i++;
        }
    }
    break;
}

}

void tempchange(int ch) //DS18B20 开始获取温度并转换
{
    dsreset();
    delay(1);
    tempwritebyte(ch,0xcc);
    tempwritebyte(ch,0x44);
}

uint get_temp(int ch)
{
    uchar a,b;
    dsreset();
    delay(1);
    tempwritebyte(ch,0xcc);
```

```
tempwritebyte(ch,0xbe);
a=tempread(ch);
b=tempread(ch);
temp=b;
temp<<=8;
temp=temp|a;

if(temp<2048)//为正
{f_temp=temp*0.0625;zf=0;} //温度在寄存器中为 12 位，分辨率为 0.0625
if(temp>2048)//为负
{f_temp=(~temp+1)*0.0625;zf=1;} //温度在寄存器中为 12 位，分辨率为 0.0625

temp=f_temp*10+0.5; //乘以 10 表示小数点后只取一位，加 0.5 是四舍五入
f_temp=f_temp+0.05;

return temp; //temp 是整型
}
void display(uchar num,int dat)
{
    switch(num)
    {
        case 0:
            B4=0;B3=1;B2=1;B1=1;
            P0=table[dat];
            delay(1);
            break;
        case 1:
            B4=1;B3=0;B2=1;B1=1;
            P0=table[dat];
            delay(1);
            break;
        case 2:
            B4=1;B3=1;B2=0;B1=1;
            P0=table[dat];
            P0&=0x7f;//小数点
            delay(1);
            break;
        case 3:
            B4=1;B3=1;B2=1;B1=0;
            P0=table[dat];
            delay(1);
            break;
    }
}
void dis_temp(uint temp)
{
```



```
        w0=temp/1000%10;
        w1=temp/100%10;
        w2=temp%100/10;
        w3=temp%10;
    time=0;num=0;
    while(1)
    {
        B4=0;B3=1;B2=1;B1=1;
        if(zf==0)
        {P0=0xff;} //不显示
        if(zf==1)
        {P0=0xbf;} //显示负号
        delay(1);

        display(1,w1);
        display(2,w2);
        display(3,w3);
        if(time) break;
    }
}
void dis_channel(int io)
{
    time=0;num=0;
    while(1)
    {
        B4=0;B3=1;B2=1;B1=1;
        P0=0xc6;
        delay(1);

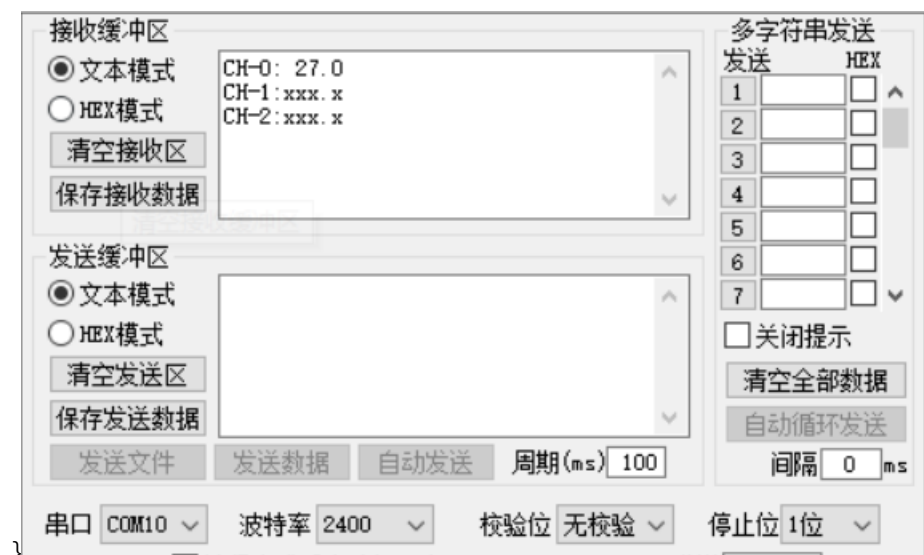
        B4=1;B3=0;B2=1;B1=1;
        P0=0x89;
        delay(1);

        B4=1;B3=1;B2=0;B1=1;
        P0=0xbf;
        delay(1);

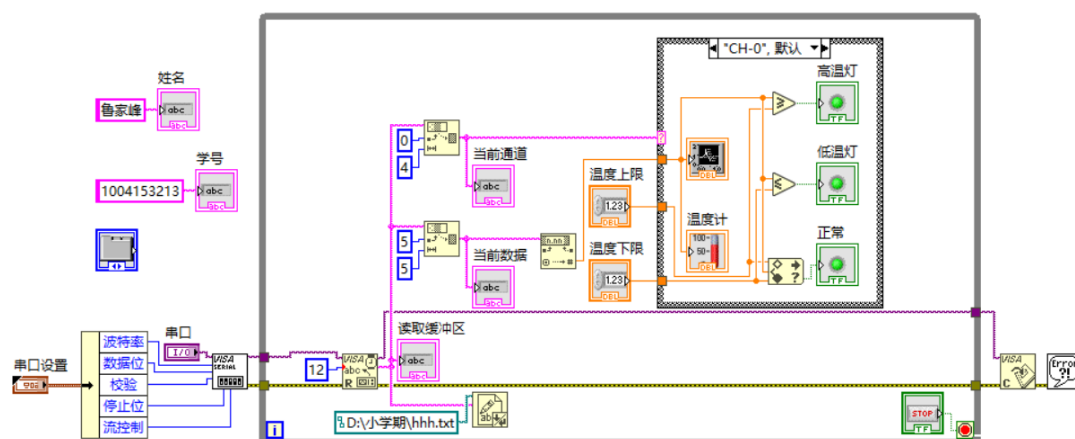
        B4=1;B3=1;B2=1;B1=0;
        P0=table[io];
        delay(1);
        if(time) break;
    }
}
void init_com(void)
{
```

```
TMOD=0x21;
TH1=256-fosc/32/12/2400;//高波特率会出错
TH0=(65536-10000/(12000000/fosc))/256;
TL0=(65536-10000/(12000000/fosc))%256;
EA=1;
ET0=1;
ES=1;
TR0=1;
TR1=1;
REN=1;
SM0=0;
SM1=1;
}
void send_bit(uchar val)
{
    SBUF=val;
    while(!TI);
    TI=0;
}
void send_string(char *pchar)
{
    int i=0;
    while(pchar[i]!='\0')
        send_bit(pchar[i++]);
}
void sendtuart(int ch)
{
    uchar sendstring[12]="CH-x:xxx.x\r\n";
    int i;
    if(!zf)
    {
        sendstring[3]=ch+'0';
        sendstring[5]=w0+'0';
        sendstring[6]=w1+'0';
        sendstring[7]=w2+'0';
        sendstring[9]=w3+'0';
        if(sendstring[5]=='0')
            {sendstring[5]=' ';}
    }
    else
    {
        sendstring[3]=ch+'0';
        sendstring[5]='x';
        sendstring[6]='x';
        sendstring[7]='x';
        sendstring[9]='x';
    }
}
```

```
    ES=0;
    send_string(sendstring);
    ES=1;
}
void main()
{
    init_com();
    while(1)
    {
        for(ch=0;ch<3;ch++)
        {
            dis_channel(ch); //显示测温通道
            tempchange(ch);  //温度转换
            dis_temp(get_temp(ch)); //读取并处理温度数据
            // dis_temp(get_temp(ch)); //读取并处理温度数据
            //dis_temp(get_temp(ch)); //读取并处理温度数据
            sendtuart(ch);      //串口发送
        }
    }
}
void UART_ISR() interrupt 4
{
    if(RI==1)
    {
        RI=0;
        rbuff=SBUF; //接收的数据
        flag=1;
    }
}
void T0_time() interrupt 1
{
    TF0=0;
    TH0=(65536-10000/(12000000/fosc))/256;
    TL0=(65536-10000/(12000000/fosc))%256;
    if(++num>=100)
    {
        num=0;
        time=1;
    }
}
```



## 六、LabView 系统设计



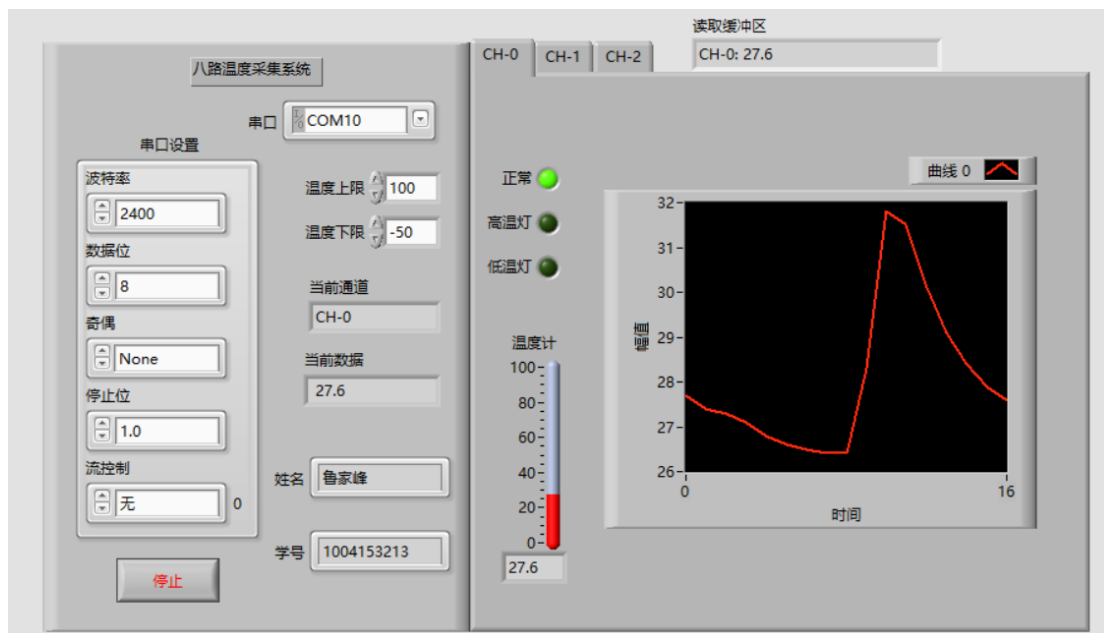
先将串口传进来的 12 位字符截取前 4 位（即通道编号），判断当前通道编号与预置的通道编号中的哪一个相等，则执行该分支。将截取到的当前温度与通道号在前面板进行显示。同时判断当前温度所处的范围，控制指示灯的亮灭，其中温度的范围可由用户通过前面板进行输入。将数值转换后的温度接入“波形图表”器件，即可观测各路波形。

hhh.txt - 记事本

文件(E) 编辑(E) 格式(O) 查看(V) 帮助(H)

CH-0: 26.3

根据写入文本函数可以把数据保存到文本



LabView 温度显示系统主要包括 CH-0~CH-7 八路温度显示选项卡及左侧的基本信息设置与显示。其中各部分具体功能如下：八路温度显示及预警：以温度计及具体数值的方式显示八路温度传感器所测得的数值，并且每一路均有高温、低温及正常指示灯来指示这一路温度是否超过规定的温度上限或下限。

## 七、总结

在这几周的课程设计和实习中，我从中学到了很多很多。

首先，感谢老师对我们的指导和对我们严格的要求。起初的两天，我们查阅资料，从书中找，上网查，但是始终没有一个具体方案，经过老师对我们的指点，我们可以从宏观把握整个实验，大体分成四部分做：1 电路设计；2 电路焊接；3 数码管动态方式实时显示温度；4 labview 系统温度显示。

其中也遇到了一些问题，例如 1. 程序没插传感器时温度显示结果的格式不正确，应该判断数据范围或判断初始化是否成功，2. 单片机串口乱码，因为晶振是 12MHz 精确度不高，需要调低波特率。3. labview 程序串口无法使用，需要安装 VISA 驱动。之后两个星期，我们逐个问题攻破，把每块都制作出来，然后整合成我们所需要的程序，刚开始还调试不出来，经过努力，最后我们终于弄好了。

老师还告诉我们，设计要注重软件和硬件的结合，尤其是硬件，有了硬件，软件程序很好写。这次课程设计让我受益匪浅，也实实在在的学到了不少东西，尤其是那严谨的态度。