基于ds18b20数字温度报警器
课程设计报告书

姓　　名：_____

学　　号：_____

专业班级：_____

指导老师：_____

所在学院：_____

年　　月　　日

摘要

随着时代的进步和发展，单片机技术已经普及到我们生活、工作、科研、各个领域，已经成为一种比较成熟的技术，本文主要介绍了一个基于 89S51 单片机的测温系统，详细描述了利用数字温度传感器 DS18B20 开发测温系统的过程，重点对传感器在单片机下的硬件连接，软件编程以及各模块系统流程进行了详尽分析，特别是数字温度传感器 DS18B20 的数据采集过程。对各部分的电路也一一进行了介绍,该系统可以方便的实现实现温度采集和显示，并可根据需要任意设定上下限报警温度，它使用起来相当方便，具有精度高、量程宽、灵敏度高、体积小、功耗低等优点，适合于我们日常生活和工、农业生产中的温度测量，也可以

当作温度处理模块嵌入其它系统中，作为其他主系统的辅助扩展。

DS18B20 与AT89C51 结合实现最简温度检测系统，该系统结构简单，抗干扰能力强，适合于恶劣环境下进行现场温度测量，有广泛的应用前景。

关键词： 单片机

DS18B20 温度传感器 数字温度计 AT89S52

目　录

1 概述

1.1 课程设计的意义

本次课程设计是对于我们所学的传感器原理知识所进行的一次实际运用，通过自主的课程设计和实际操作，可增加我们自身的动手能力。特别是对温度传感这方面的知识有了实质性的了解，对进一步学习传感器课程起到很大的作用。本课程设计由4个人共同完成，在锻炼了自我的同时也增强了自己的团队意识和团队协作精神。

1.2 设计的任务和要求

1、基本范围-50℃-110℃

2、精度误差小于 0.5℃

3、LED 数码直读显示

4、可以任意设定温度的上下限报警功能

2 系统总体方案及硬件设计

2.1 数字温度计设计方案论证

2.1.1 方案一2.1.1

由于本设计是测温电路，可以使用热敏电阻之类的器件利用其感温效应，在将随被测温度变化的电压或电流采集过来，进行 A/D 转换后，就可以用单片机进行数据的处理，在显示电路上，就可以将被测温度显示出来，这种设计需要用到 A/D 转换电路，其中还涉及到电阻与温度的对应值的计算，感温电路比较麻烦。而且在对采集的信号进行放大时容易受温度的影响出现较大的偏差。

2.1.2 方案二2.1.1.2

进而考虑到用温度传感器，在单片机电路设计中，大多都是使用传感器，所以这是非常容易想到的，所以可以采用一只温度传感器 DS18B20，此传感器，可以很容易直接读取被测温度值，进行转换，电路简单，精度高，软硬件都以实现，而且使用单片机的接口便于系统的再扩展，满足设计要求。

从以上两种方案，很容易看出，采用方案二，电路比较简单，费用较低，可靠性高，软件设计也比较简单，故采用了方案二。

2.2 系统总体设计

温度计电路设计总体设计方框图如图 1 所示，控制器采用单片机AT89S51，温度传感器采用 DS18B20，用 3 位 LED 数码管以串口传送数据实现温度显示。

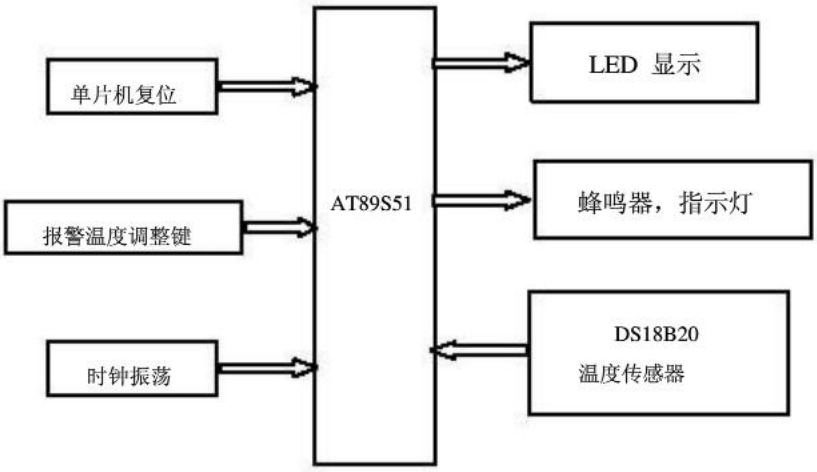

图 2.2—1　总体设计方框图

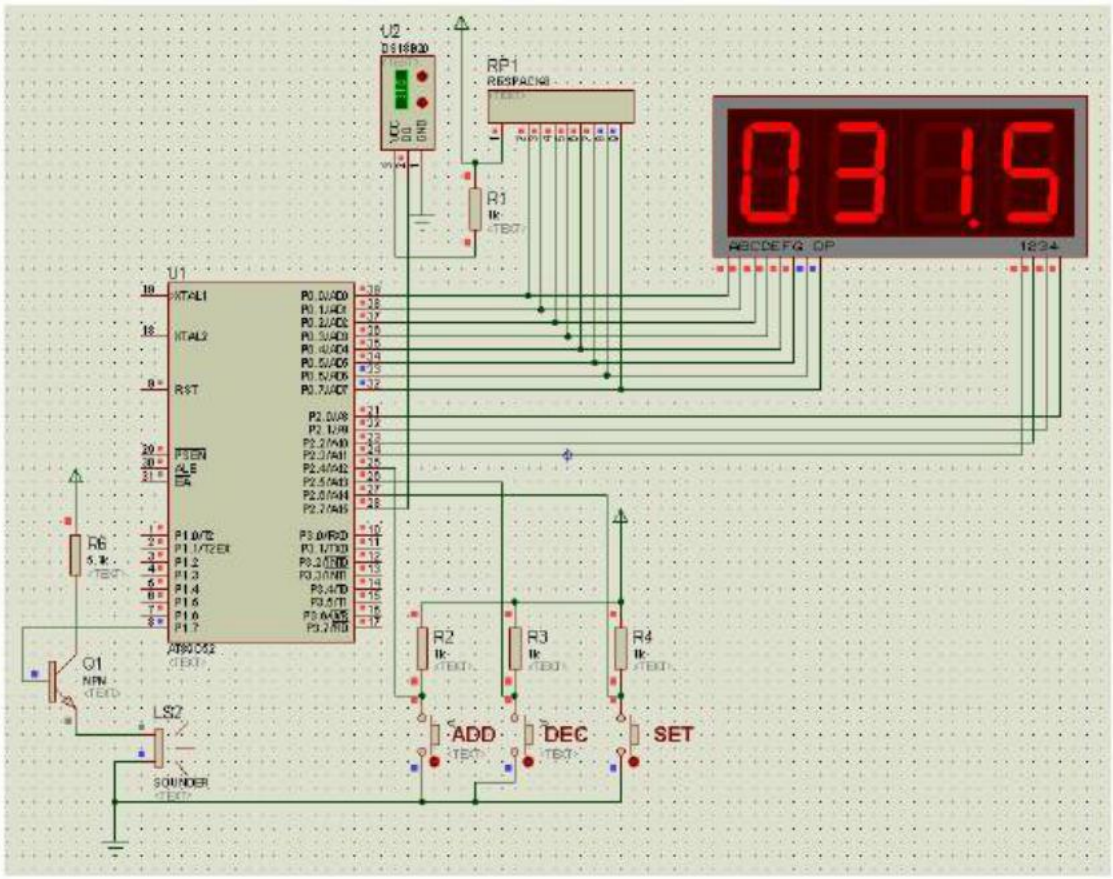

图 2.2—2 系统仿真图

2.3 系统模块

系统由单片机最小系统、显示电路、按键、温度传感器等组成。

2.3.1 主控制器2.3.1

单片机 AT89S51 具有低电压供电和体积小等特点，四个端口只需要两个口就能满足电路系统的设计需要，很适合便携手持式产品的设计使用系统可用二节电池供电。晶振采用 12MHZ。复位电路采用上电加按钮复位。

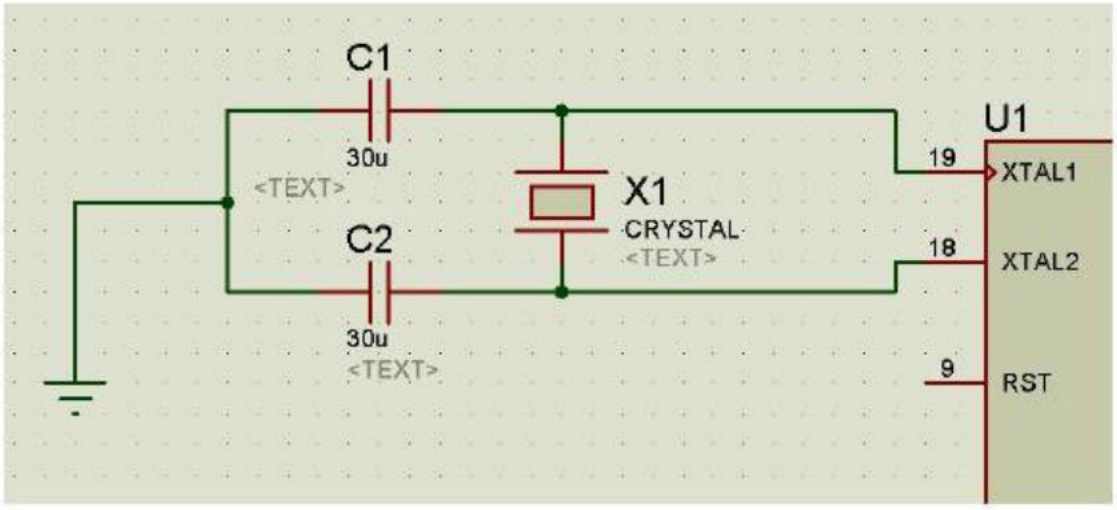

图 2.3.1—1 晶振电路

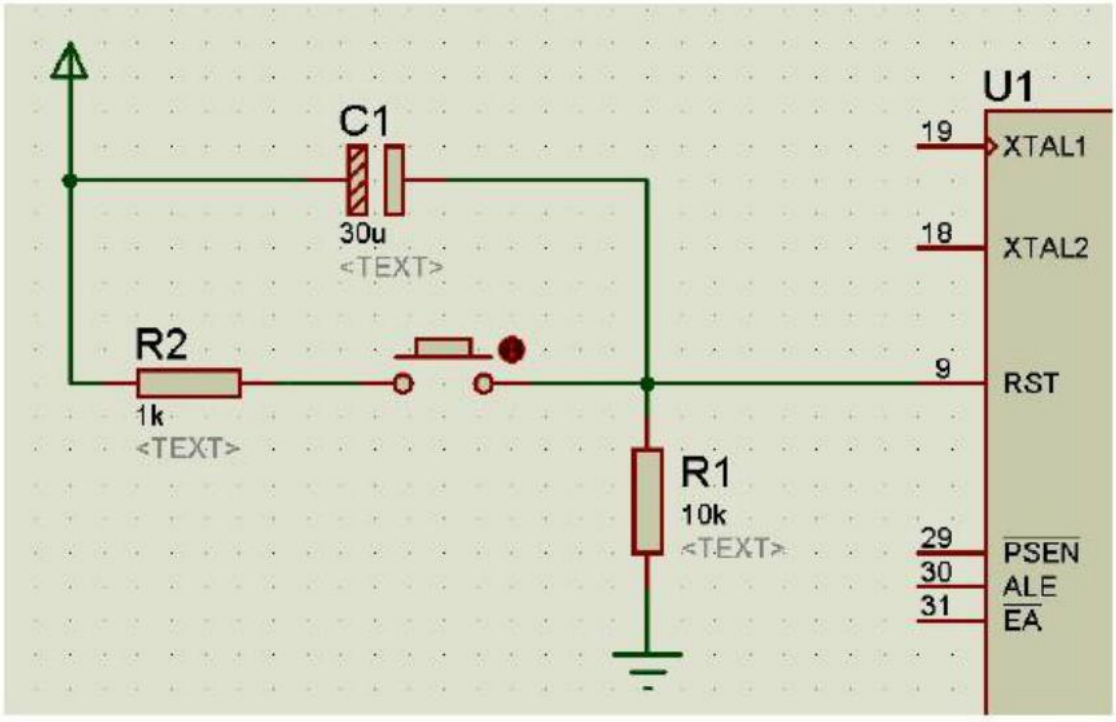

图 2.3.1—2 复位电路

2.3.2 显示电路2.3.2

显示电路采用 4 位共阴极 LED 数码管，P0 口由上拉电阻提高驱动能力，作为段码输出并作为数码管的驱动。P2 口的低四位作为数码管的位选端。采用动态扫描的方式显示。

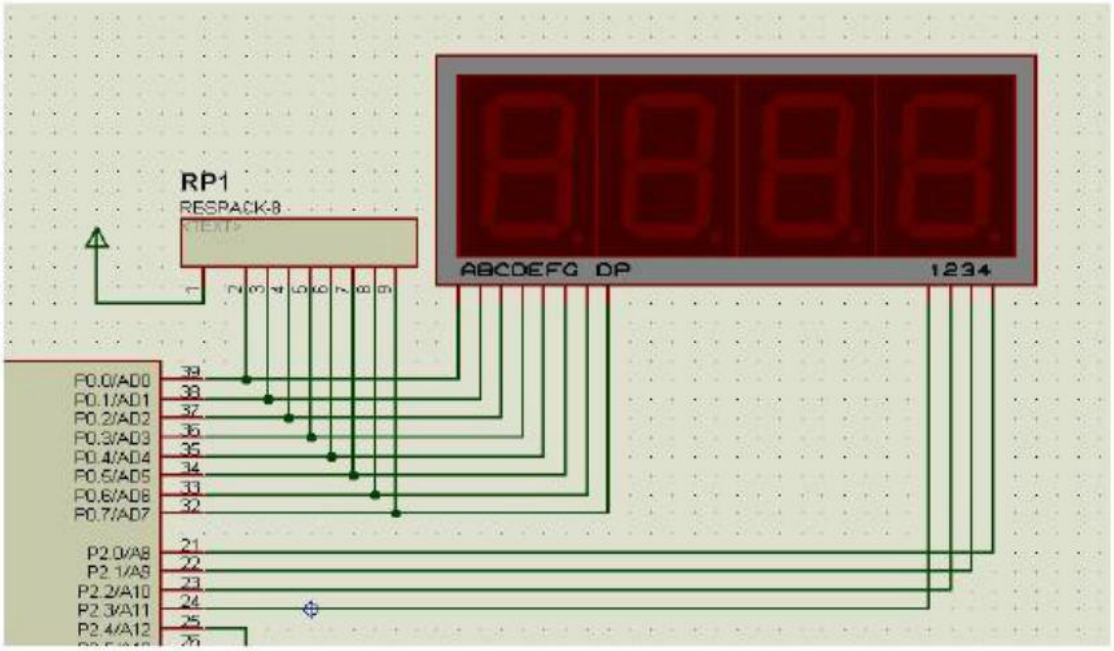

图 2.3.2 数码管显示电路

2.3.3 温度传感器2.3.3

DS18B20 温度传感器是美国 DALLAS 半导体公司最新推出的一种改进型智能温度传感器，与传统的热敏电阻等测温元件相比，它能直接读出被测温度，并且可根据实际要求通过简单的编程实现 9 ～ 1 2 位的数字值读数方式。DS18B20 的性能特点如下：

1、独特的单线接口仅需要一个端口引脚进行通信；2、多个 DS18B20 可以并联在惟一的三线上，实现多点组网功能；3、无须外部器件；4、可通过数据线供电，电压范围为 3.0~5.5V；5、零待机功耗；6、温度以 9 或 1 2 位数字；7、用户可定义报警设置；8、报警搜索命令识别并标志超过程序限定温度（温度报警条件）的器件；9、负电压特性，电源极性接反时，温度计不会因发热而烧毁，但不能正常工作；DS18B20 可以采用两种方式供电，一种是采用电源供电方式，此时 DS18B20 的 1 脚接地，2 脚作为信号线，3 脚接电源。另一种是寄生电源供电方式，如图 4 所示单片机端口接单线总线，为保证在有效的 DS18B20 时钟周期内提供足够的电流，可用一个 MOSFET 管来完成对总线的上拉。

当 DS18B20 处于写存储器操作和温度 A/D 转换操作时，总线上必须有强的上拉，上拉开启时间最大为 10us。采用寄生电源供电方式时 VDD 端接地。由于单线制只有一根线，因此发送接口必须是三态的。

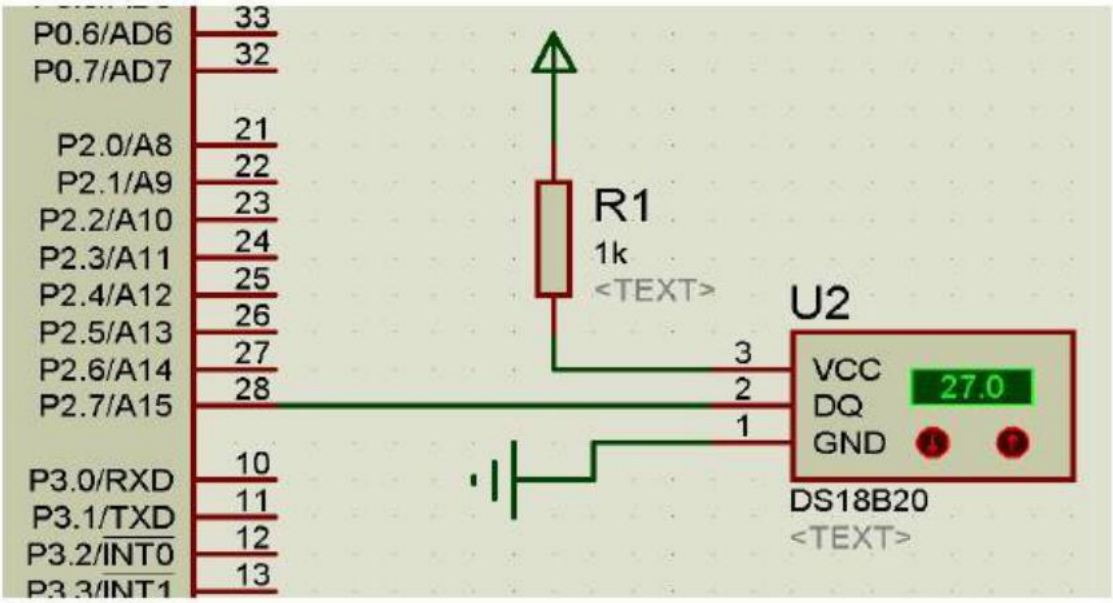

图 2.3.3 温度传感器与单片机的连接

2.3.4 报警温度调整按键

本系统设计三个按键，采用查询方式，一个用于选择切换设置报警温度和当前温度，另外两个分别用于设置报警温度的加和减。均采用软

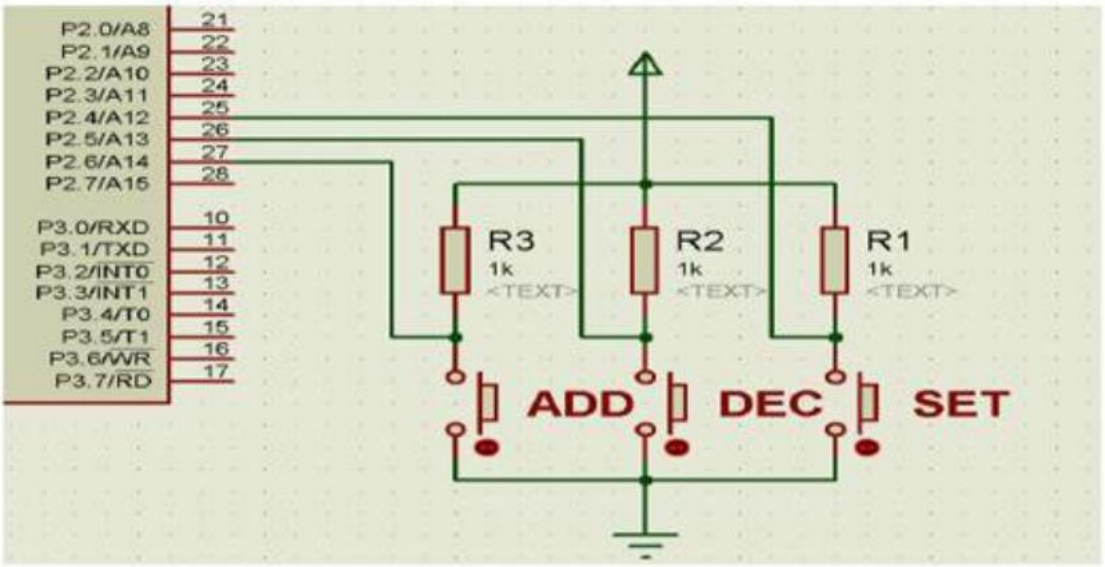

图 2.3.4 按键电路

3 系统软件算法分析

系统程序主要包括主程序，读出温度子程序，温度转换命令子程序，计算温度子程序，显示数据刷新子程序，按键扫描处理子程序等。

3.1 主程序流程图

主程序的主要功能是负责温度的实时显示、读出并处理 DS18B20 的测量的当前温度值，温度测量每 1s 进行一次。这样可以在一秒之内测量一次被测温度，其程序流程见图 3.1 所示。

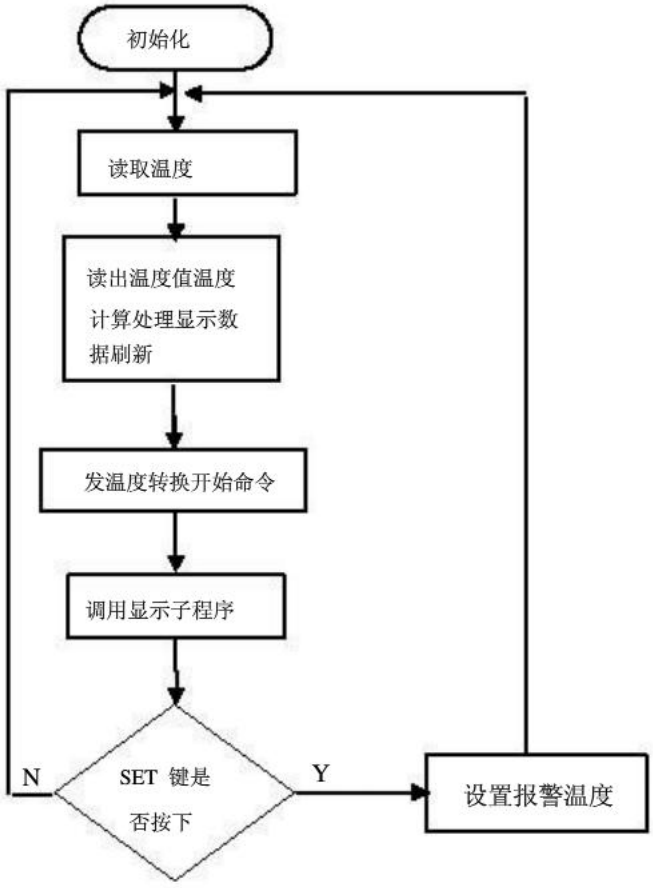

图 3.1 主程序流程图

3.2 读出温度子程序

读出温度子程序的主要功能是读出 RAM 中的 9 字节，在读出时需进行 CRC 校验，校验有错时不进行温度数据的改写。其程序流程图如图 3.2 示

3.3 温度转换命令子程序

温度转换命令子程序主要是发温度转换开始命令，当采用 12 位分辨率时转换时间约为750ms，在本程序设计中采用 1s 显示程序延时法等待转换的完成。温度转换命令子程序流程图如上图，图 3.3 所示

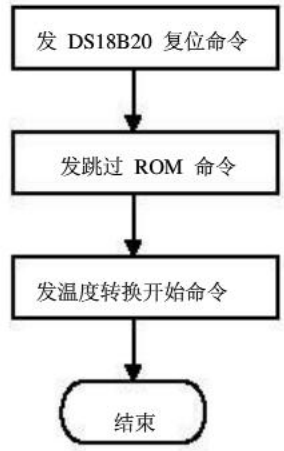

图 3.3 温度转换流程图

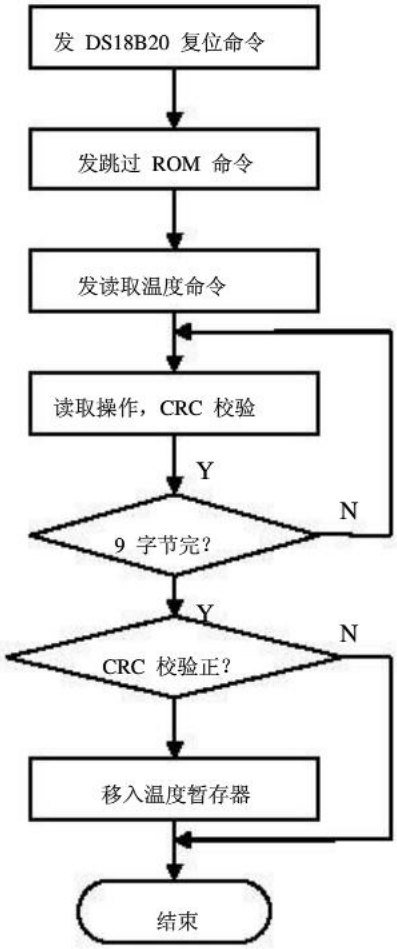

图 3.2 读温度流程图

3.4 计算温度子程序

计算温度子程序将 RAM 中读取值进行 BCD 码的转换运算，并进行温度值正负的判定，其程序流程图如图 3.4 所示。

3.5 显示数据刷新子程序

显示数据刷新子程序主要是对分离后的温度显示数据进行刷新操作，当标志位位为 1时将符号显示位移入第一位。程序流程图如图 3.5。

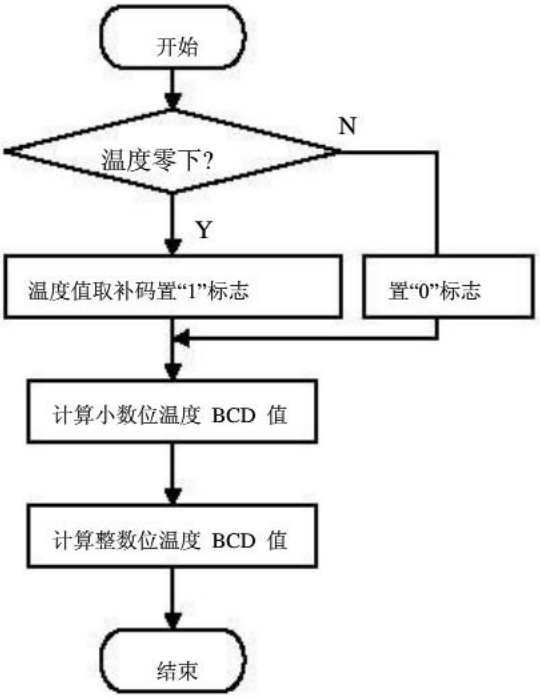

图 3.4 计算温度流程图

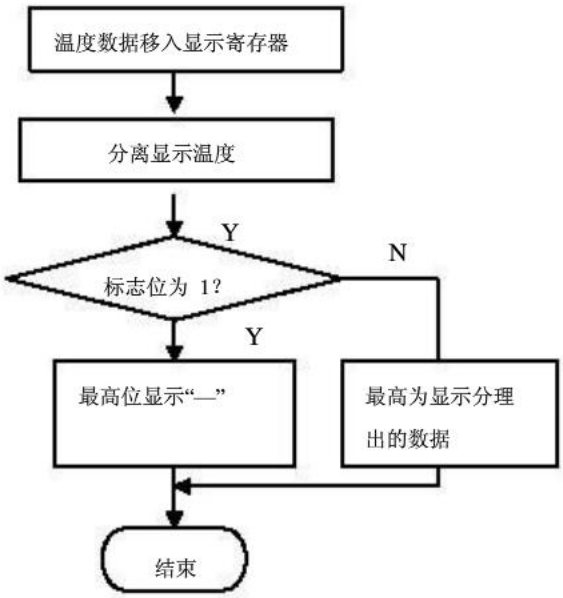

图 3.5 显示数据刷新流程图

3.6 按键扫描处理子程序

按键采用扫描查询方式，设置标志位，当标志位为 1 时，显示设置温度，否则显示当前温度。如下图 3.6 示。

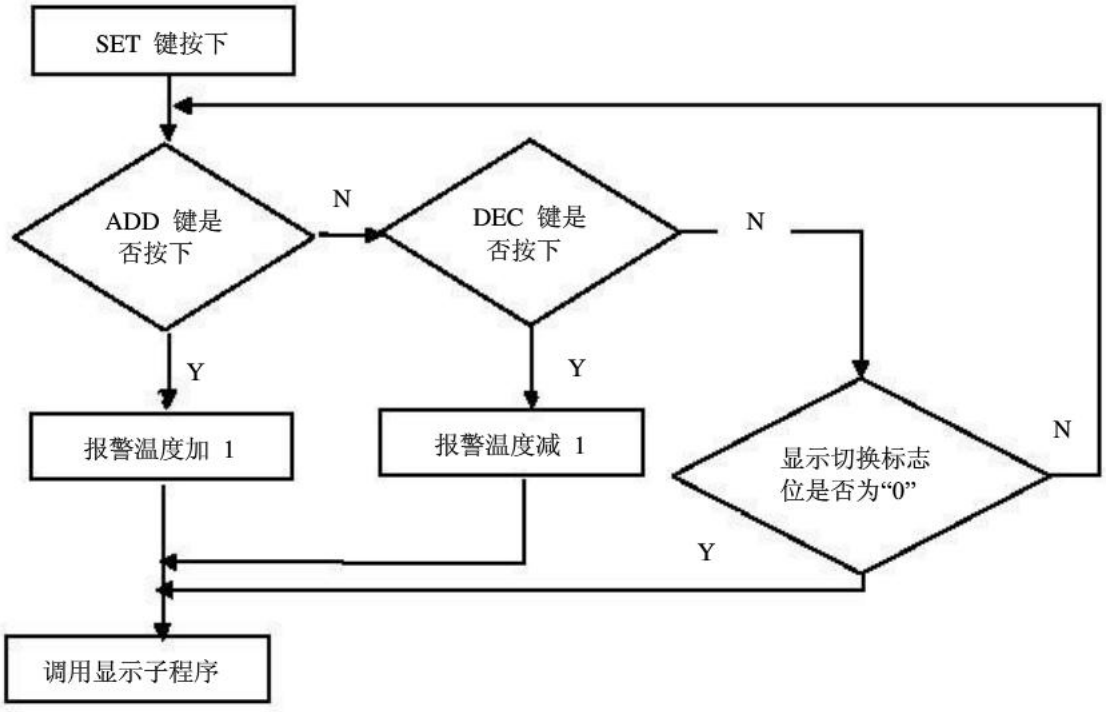

图 3.6 按键扫描处理子程序

4 实验仿真

进入 protuse 后，连接好电路，并将程序下载进去。将 DS18B20 的改为 0.1 数码管显示温度与传感器的温度相同。

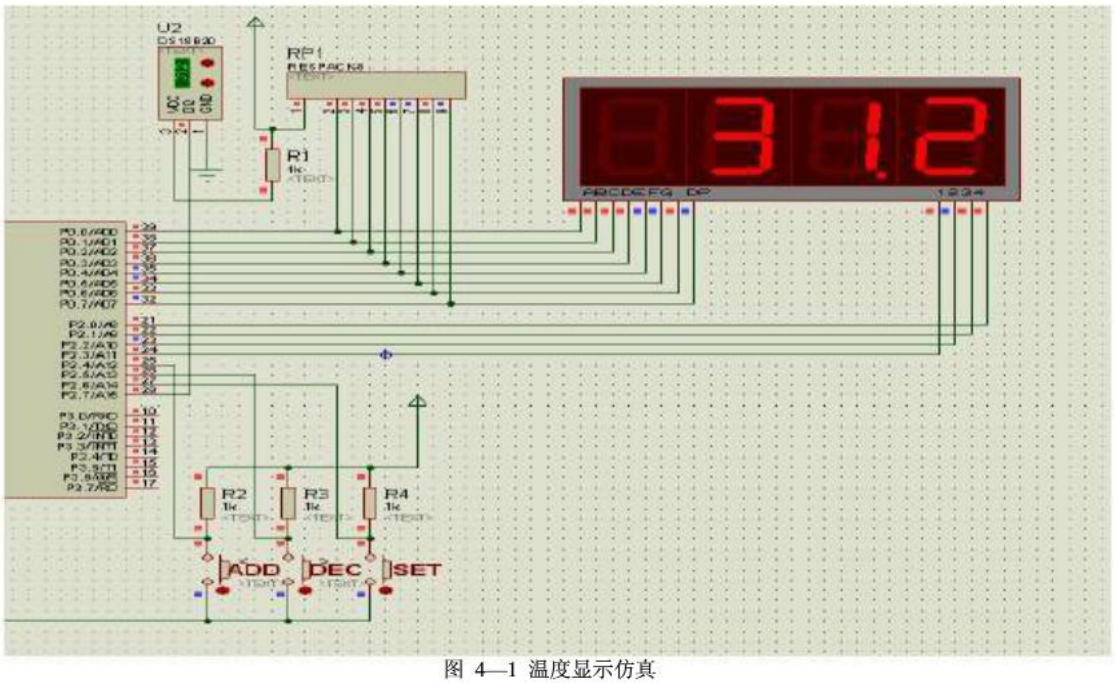

图 4—1 温度显示仿真

当按下 SET 键一次时，进入温度报警上线调节，此时显示软件设置的温度报警上线， ADD按或 DEC 分别对报警温度进行加一或减一。

当再次按下 SET 键时，进入温度报警下线调节，此时显示软件设置的温度报警下线， ADD按或 DEC 分别对报警温度进行加一或减一。

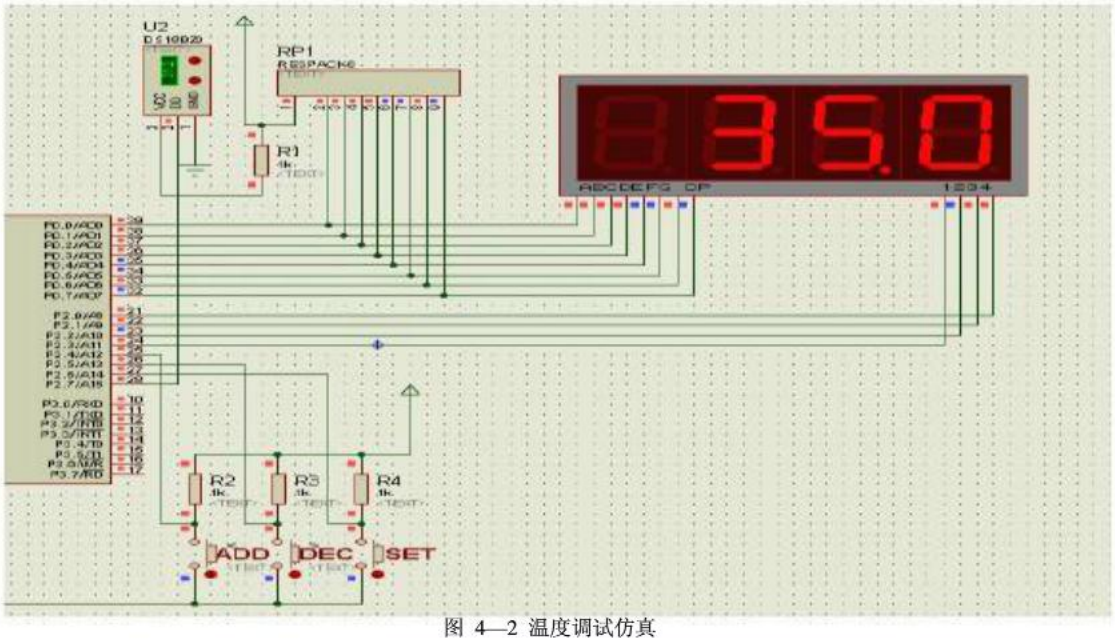

图 4—2 温度调试仿真

当第三次按下 SET 键时，退出温度报警线设置。显示当前温度。

5 总结与体会

本次基于“ds18b20 数字温度报警器”的传感器课程设计大致可以分为：资料收集→程序编辑→电路设计→模拟仿真→电板焊接。每个过程相辅相成却又相互独立。

通过这次对数字温度计的设计与制作，让我了解了设计电路的程序，也让我了解了关于数字温度计的原理与设计理念，要设计一个电路总要先用仿真仿真成功之后才实际接线的。但是最后的成品却不一定与仿真时完全一样因为，再实际接线中有着各种各样的条件制约着。而且，在仿真中无法成功的电路接法，在实际中因为芯片本身的特性而能够成功。所以，在设计时应考虑两者的差异，从中找出最适合的设计方法。

通过一个一个步骤的跟进，让我对很多电子元器件的结构和基本特性有了一定的了解，对电路的实际操作让我对电路有了深刻的理解。焊接过程是一个很有趣的过程，通过小心翼翼的一个个引脚的焊接，最终成就我们的温度传感器，每一步都那么的谨慎以防与相邻的电路短接。在很大的程度上锻炼了我的耐心，同时也能够对整个电路设计及走向有一个深刻的了解、理解。

当然，由于种种原因：元器件缺失、系统本身及电路的影响等导致所得的结果不够精确，无法达到预想的理想状态，让人很是遗憾。从这次的课程设计中，我真真正正的意识到，在以后的学习中，要理论联系实际，把我们所学的理论知识用到实际当中，学习传感器更是如此，任何元件、程序等只有在反复的学习和使用过程中才能在运用过程中得心应手，这就是我在这次课程设计中的最大收获。

查考文献

【1】马忠梅，张凯，等. 单片机的 C 语言应用程序设计(第四版) 北京航空航天大学出版社

【2】薛庆军，张秀娟，等.单片机原理实验教程 北京航天航空大学出版社

【3】廖常初.现场总线概述［J］.电工技术，1999.

【4】传感器原理 设计与应用 刘迎春 叶湘滨 国防科技大学出版社

【5】常用电子元器件及应用电路 赵春云 曹经稳 赵春强 电子工业出版社

【6】单片机原理及应用 杨恢先 黄辉先 人民邮电出版社

附 1 源程序代码

```
//DS18B20 的读写程序,数据脚 P2.7                                    //
//温度传感器 18B20 汇编程序,采用器件默认的 12 位转化              //
//最大转化时间 750 微秒,显示温度-55 到+125 度,显示精度 //
//为 0.1 度，显示采用 4 位 LED 共阳显示测温值                       //
//P0 口为段码输入,P34~P37 为位选                                    //
/************************************************/

#include "reg51.h"
#include "intrins.h"                //_nop_();延时函数用
#define dm P0                       //段码输出口
#define uchar unsigned char
#define uint       unsigned int
sbit DQ=P2^7;              //温度输入口
sbit w0=P2^0;              //数码管 4
sbit w1=P2^1;              //数码管 3
sbit w2=P2^2;              //数码管 2
sbit w3=P2^3;              //数码管 1
sbit beep=P1^7;            //蜂鸣器和指示灯
sbit set=P2^6;             //温度设置切换键
sbit add=P2^4;             //温度加
sbit dec=P2^5;             //温度减
int temp1=0;               //显示当前温度和设置温度的标志位为 0 时显示当前温度
uint h;
uint temp;
uchar r;
uchar high=35,low=20;
uchar sign;
uchar q=0;
uchar tt=0;
uchar scale;
//**************温度小数部分用查表法***********//
uchar code ditab[16]={0x00,0x01,0x01,0x02,0x03,0x03,0x04,0x04,0x05,0x06,0x06,0x07,0x08,0x08,0x09,0x09};
      //小数断码表
uchar code table_dm[12]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f,0x00,0x40};
//共阴 LED 段码表              "0"   "1"   "2"   "3"   "4"   "5"   "6"   "7"   "8"   "9" "不亮" "-"
uchar table_dm1[]={0xbf,0x86,0xdb,0xcf,0xe6,0xed,0xfd,0x87,0xff,0xef};          //个位带小数点的断码表

uchar data temp_data[2]={0x00,0x00};                    //读出温度暂放
uchar data display[5]={0x00,0x00,0x00,0x00,0x00};       //显示单元数据，共 4 个数据和一个运算暂用
/*****************11us 延时函数************************/
```

```
void delay(uint t)
{
   for (;t>0;t--);
}

void scan()
{
int j;
        for(j=0;j<4;j++)
        {
                switch (j)
                {
                        case 0: dm=table_dm[display[0]];w0=0;delay(50);w0=1;//xiaoshu
                        case 1: dm=table_dm1[display[1]];w1=0;delay(50);w1=1;//gewei
                        case 2: dm=table_dm[display[2]];w2=0;delay(50);w2=1;//shiwei
                        case 3: dm=table_dm[display[3]];w3=0;delay(50);w3=1;//baiwei
                                //      else{dm=table_dm[b3];w3=0;delay(50);w3=1;}
        }
        }
}

//***************DS18B20 复位函数***********************/
ow_reset(void)
{
char presence=1;
while(presence)
{
   while(presence)
    {
      DQ=1;_nop_();_nop_();//从高拉倒低
DQ=0;
delay(50);                  //550 us
DQ=1;
delay(6);                   //66 us
presence=DQ;                  //presence=0 复位成功,继续下一步
    }
    delay(45);                      //延时 500 us
    presence=~DQ;
}
DQ=1;                                 //拉高电平
}
/***************DS18B20 写命令函数***********************/
//向 1-WIRE 总线上写 1 个字节
```

```
void write_byte(uchar val)
{
   uchar i;
   for(i=8;i>0;i--)
   {
    DQ=1;_nop_();_nop_();                          //从高拉倒低
    DQ=0;_nop_();_nop_();_nop_();_nop_();          //5 us
    DQ=val&0x01;                                   //最低位移出
    delay(6);                                      //66 us
    val=val/2;                                     //右移 1 位
    }
    DQ=1;
    delay(1);
}
/***************DS18B20 读 1 字节函数*********************/
//从总线上取 1 个字节
uchar read_byte(void)
{
uchar i;
uchar value=0;
for(i=8;i>0;i--)
{
   DQ=1;_nop_();_nop_();
   value>>=1;
   DQ=0;_nop_();_nop_();_nop_();_nop_();           //4 us
   DQ=1;_nop_();_nop_();_nop_();_nop_();           //4 us
   if(DQ)value|=0x80;
   delay(6);                                       //66 us
}
DQ=1;
return(value);
}

/****************读出温度函数*********************/
read_temp()
{
   ow_reset();                          //总线复位
   delay(200);
   write_byte(0xcc);                    //发命令
   write_byte(0x44);                    //发转换命令
   ow_reset();
   delay(1);
   write_byte(0xcc);                    //发命令
   write_byte(0xbe);
```

```
    temp_data[0]=read_byte();          //读温度值的第字节
    temp_data[1]=read_byte();          //读温度值的高字节
    temp=temp_data[1];
    temp<<=8;
    temp=temp|temp_data[0];             // 两字节合成一个整型变量。
 eturn temp;                           //返回温度值
}
/***************温度数据处理函数**********************/
//二进制高字节的低半字节和低字节的高半字节组成一字节,这个
//字节的二进制转换为十进制后, 就是温度值的百、十、个位值,而剩
//下的低字节的低半字节转化成十进制后,就是温度值的小数部分
/*****************************************************/
work_temp(uint tem)
{
uchar n=0;
if(tem>6348)                                  // 温度值正负判断
        {tem=65536-tem;n=1;}                   // 负温度求补码,标志位置 1
         display[4]=tem&0x0f;                  // 取小数部分的值
   display[0]=ditab[display[4]];       // 存入小数部分显示值
         display[4]=tem>>4;                    // 取中间八位,即整数部分的值
display[3]=display[4]/100;             // 取百位数据暂存
   display[1]=display[4]%100;            // 取后两位数据暂存
   display[2]=display[1]/10;             // 取十位数据暂存
   display[1]=display[1]%10;             //个位数据
  =display[1]+display[2]*10+display[3]*100;
/////符号位显示判断/////
   if(!display[3])
   {
      display[3]=0x0a;                   //最高位为 0 时不显示
      if(!display[2])
{
    display[2]=0x0a;                 //次高位为 0 时不显示
}
   }
   if(n){display[3]=0x0b;}             //负温度时最高位显示"-"
}

void BEEP()
{
       if((r>=high&&r<129)||r<low)
       {
            beep=!beep;
       }
       else
```

```
        {
            beep=0;
        }
}
//*********设置温度显示转换************//
void xianshi(int horl)
{
        int n=0;
        if(horl>128)
        {
            horl=256-horl;n=1;
        }
        display[3]=horl/100;
        display[3]=display[3]&0x0f;
        display[2]=horl%100/10;
        display[1]=horl%10;
        display[0]=0;
        if(!display[3])
        {
            display[3]=0x0a;                    //最高位为 0 时不显示
            if(!display[2])
             {
                display[2]=0x0a;                //次高位为 0 时不显示
             }
        }
          if(n)
          {
              display[3]=0x0b; //负温度时最高位显示"-"
          }

}

//*********按键查询程序**************//
void keyscan()
{
        int temp1;      //最高温度和最低温度标志位
        if(set==0)
        {
            while(1)
            {
                delay(500);//消抖
                if(set==0)
                {
                    temp1++;
```

```
            while(!set)
            scan();
        }
        if(temp1==1)
        {
            xianshi(high);
            scan();
            if(add==0)
            {
                while(!add)
                scan();
                high+=1;
            }
            if(dec==0)
            {
                while(!dec)
                scan();
                high-=1;
            }
        }

        if(temp1==2)
        {
            xianshi(low);
            if(add==0)
            {
                while(!add)
                scan();
                low+=1;
            }
            if(dec==0)
            {
                while(!dec)
                scan();
                low-=1;
            }
            scan();
        }
        if(temp1>=3)
        {
            temp1=0;
            break;
        }
    }
```

```
        }
}

/****************主函数************************/
void main()
{

        dm=0x00;                            //初始化端口
        w0=0;
        w1=0;
        w2=0;
        w3=0;
        for(h=0;h<4;h++)                     //开机显示"0000"
        {
            display[h]=0;
        }
        ow_reset();                          //开机先转换一次
        write_byte(0xcc);                   //Skip ROM
        write_byte(0x44);                   //发转换命令
        for(h=0;h<100;h++)                   //开机显示"0000"
        {
            scan();
        }
        while(1)
        {
             if (temp1==0)
            {
                  work_temp(read_temp());          //处理温度数据
                BEEP();
                  scan();                          //显示温度值
                keyscan();
            }
              else
                 keyscan();
             }
}

//**********************结束*************************//
```