

温度传感器设计

陆毅



2023-2-4

华中科技大学

船舶与海洋工程学院

目录

[一、设计内容 2](#_Toc126435592)

[二、设计原理 2](#_Toc126435593)

[三、设计过程 2](#_Toc126435594)

[热敏电阻测量电路 2](#_Toc126435595)

[A/D转换电路 3](#_Toc126435596)

[数据处理部分 4](#_Toc126435597)

[报警部分 4](#_Toc126435598)

[显示部分 5](#_Toc126435599)

[四、问题分析 5](#_Toc126435600)

[五、总结 8](#_Toc126435601)

[六、参考文献 8](#_Toc126435602)

# 温度传感器设计

陆毅 轮机2001班 U202010827

[摘要]以热敏电阻为敏感元件，使用半桥单臂连接法，将温度信号转化为电压信号，再通过模数转换元件和51单片机进行数据处理得到温度值，并显示在数码管上。此外，还设置了红温报警和风扇降温。

## 一、设计内容

选用电阻作为敏感元件，测试信号通过A/D转换电路后使用51单片机进行数据处理，并将结果显示于数码管屏幕上，完成一个智能传感器的基本设计。

## 二、设计原理

半桥单臂电桥可以将电阻变化转化为电压值，然后通过A/D转换电路将电压值转化为数字量。

由于本次实验中的值在0~5V范围，而ADC0809的允许输入电压也是0~5V，故不需要放大电路，直接转换后将数字量传入单片机I/O口。

ADC0809是8位A/D转换器，转换后的值D为0~即0~255，则

使用拟合方程计算U对应的温度值T

(该方程通过个人记录实验数据进行多项式拟合获得，为简便计算，仅取5阶，有一定误差)

在数码管上显示时对温度四舍五入取整(正加负减)，忽略小数部分的显示，在实际应用中影响不大。

对TT进行取余、除10等操作获得其各位数值，选择数码管对应段码即可显示温度。对于正值温度，直接显示即可，负值温度则在其前显示负号“-”。

一般使用过程更多时候需要的是温度上限限制，下限使用得少，所以该系统还设置了红温警告，当温度大于时触发蜂鸣器，并启动风扇进行散热降温。该功能只需要在软件中对温度与60作比较，选用单片机的某两个I/O口和三极管作为蜂鸣器、风扇电机的开关电路，当时输出1导通三极管，触发蜂鸣器和风扇。

降温过程中如果想要蜂鸣器停止报警，可设置一常开按钮连接单片机的中断口和GND，使用低电平触发，在中断处理程序中关闭蜂鸣器即可。

## 三、设计过程

### 热敏电阻测量电路

由于仿真中不便同步控制多个热敏电阻的环境温度，故采用半桥单臂连接法来测量变化量。

图示, 示意图

描述已自动生成

### A/D转换电路

采用ADC0809转换器，将电压值转换为数字量。表格

中度可信度描述已自动生成

本次试验仅使用通道0，可直接将ADD C/B/A3个位选信号接地，选中IN0，将转化结果输出至51单片机的P0口。由于ADC0809需要提供外部时钟脉冲，将“CLOCK”连接至P3.3口，在软件中做如下处理：

void t1(void) interrupt 3 using 0{

TH1=(65536-200)/256;

TL1=(65536-200)%256;

CLK=~CLK;

}

在定时处理函数中对P3.3口输出值取反，即可得到一定频率的方波信号。

以下部分是执行一次转换操作：

1. ST=1; EOC=1; OE=0; //一些使能操作
2. ST=0;\_nop\_();ST=1;\_nop\_();ST=0; //开始转换
3. while(EOC==0); //等待转换
4. OE=1; //OUTPUT ENABLE
5. digital=P0; //记录转化值
6. OE=0;

### 数据处理部分

图示, 示意图

描述已自动生成ADC0809的通道输入的电压值为电压表正极的电位值，而其负极电位并非为0，所以直接转换得到的数据与电压表显示的数据是不同的，需要做一定的处理。

显然可由R7=R8得出电压表负极电位为2.5V，所以只需要用正极电位减去2.5即可。

(该修正方法并不好，若电压不稳定，会导致结果误差较大。合理的方法应该对电压表负极电位值也进行一次数模转换，可将其接到通道IN1，将位选接口连接至P2剩余I/O口控制选择ADC通道。在设计过程中我尝试过该方案，但存在一些问题，将于下文“问题分析”中详述。)

U=digital\*5.0/255.0-2.5;

T=0.4705\*func(U,5)+0.2617\*func(U,4)-0.1288 \*func(U,3) +0.7678\*func(U,2) +18.38\*U+24.687;

func函数为自定义的乘方函数，func(U,5)表示U的5次方。

温度T是由热敏电阻的特征曲线拟合方程计算得出。

//对T四舍五入取整(TT为int型)

if(T>0) TT=T+0.5;

else  TT=T-0.5;

//根据温度值设置符号位段码(正：不显示，负：-)

if(TT>=0) sign=0xC0;

else{sign=0xBF;TT=0-TT;}

### 报警部分

温度大于60摄氏度时触发蜂鸣器报警，并打开风扇降温，可手动停止该次蜂鸣器报警声。

1. if(T>60){
2. MOTOR=1; //打开风扇
3. if(!flag) ALARM=1; //如果没手动停止，打开蜂鸣器
4. }
5. else ALARM=MOTOR=flag=0;

手动停止报警采用外部中断低电平触发方式，中断函数如下：

1. void int0(void) interrupt 0 using 0{
2. ALARM=0;
3. flag=1; //标志手动停止过本次报警
4. }

### 显示部分

1. While(1){
2. /\*其余代码\*/
3. Display(TT);
4. }

Display函数定于如下：

1. void Display(TT){
2. dispbuf[0]=TT%10;    *//取个位*
3. dispbuf[1]=(TT/10)%10; *//取十位*
4. dispbuf[2]=(TT/100)%10; *//取百位*
5. P2=com[0]; //数码管位选信号
6. P1=sign; //符号位
7. Delay(5);
8. P2=com[1];
9. P1=dispbitcode[dispbuf[2]]; //段码
10. Delay(5);
11. P2=com[2];
12. P1=dispbitcode[dispbuf[1]];
13. Delay(5);
14. P2=com[3];
15. P1=dispbitcode[dispbuf[0]];
16. Delay(5);
17. }

## 四、问题分析

对于“数据处理部分”提及的的分别对两通道的模拟量转换后相减出现的问题做了如下实验：

图示, 示意图

描述已自动生成

1. setChannel(0);
2. ST=1;
3. EOC=1;
4. OE=0;
5. ST=0;\_nop\_();ST=1;\_nop\_();ST=0;
6. while(EOC==0);
7. OE=1;
8. digital=P0;
9. OE=0;
10. setChannel(1);
11. ST=1;
12. EOC=1;
13. OE=0;
14. ST=0;\_nop\_();ST=1;\_nop\_();ST=0;
15. while(EOC==0);
16. OE=1;
17. digital=P0;
18. OE=0;
19. TT=digital\*500.0/255.0;

以上代码为main函数中执行A/D转换的部分代码，1~9行和11~19行是相同的内容，只是选择的通道不一致，21行的TT就是电压(因TT是整型而U是浮点型，为方便实验而直接使用TT)，其余部分代码未做修改，仍然是取余、除十等操作获取各位然后显示，显示结果如下：

屏幕上有字

描述已自动生成

“8.”是未作处理的符号位，无意义。“411”是通道0电压表正极电位值。反观上述代码，明明是通道1的转换在后，第18行的digital理应覆盖第8行digital的值，而结果并没有。

将第1行和第11行互换，先转换通道1，再转换通道0：

1. setChannel(1);
2. /\*其余代码\*/
3. setChannel(0);
4. /\*其余代码\*/

屏幕上有字

描述已自动生成

显然，“250”就是通道1电压表负极电位值。问题显而易见：后一次的转换值并没有覆盖前一次的转换值。

对比修改之前的程序：

while(1){

*/\*A/D转换\*/*

*/\*数据处理\*/*

*/\*显示\*/*

}

后一次循环中的转换值是可以覆盖前一次转换的，而在同一次循环中对两个通道进行转换，后一次的结果却不能覆盖前一次，有些令人费解。

采取过的无效措施：

1、每次读取数据前先给P0口置1：

1. P0=0xFF;
2. OE=1;
3. digital=P0;
4. OE=0;

2、两次转换之间延时1秒；

3、在不同的循环中选择不同的通道：

1. /\*前面定义了int ch=0\*/
2. while(1)
3. {
4. setChannel(ch);
5. ST=1;
6. EOC=1;
7. OE=0;
8. ST=0;\_nop\_();ST=1;\_nop\_();ST=0;
9. while(EOC==0);
10. P0=0xFF;
11. OE=1;
12. digital=P0;
13. OE=0;
15. if(ch<1) ch++;
16. else ch=0;
18. TT=digital\*500.0/255.0;
19. /\*显示部分\*/
20. }

最后经过阅读了许多网络上使用ADC0809的代码示例，偶然发现一篇案例中，在while循环中第一行先进行一个5毫秒的延时，结果发现能解决上述问题！但究竟为什么是这样，尚未找到原因。

但是，可能由于ADC0809本身转换存在一定误差的原因，使用此法测得的电压差反而不如直接减2.5V来的准确，而是有0.02V左右的误差，转换成温度时会有的误差。在实际应用中，一般也会采用稳压芯片，所以干脆直接减2.5V，以上就当作是一个学习和探索。

## 五、总结

本次设计完成了一个最基本的电阻式传感器，其组成很简单，就是用电桥将电阻变化转换为电压信号，然后经过A/D转换输入51单片机，再有电压值计算温度值，最终在数码管中显示温度，附加红温报警和散热功能。

经Proteus仿真调试，能达到预期效果。

## 六、参考文献

[1]刘焕平,韩树新.ADC0809与AT89C51的一种接口方法[J].石家庄师范专科学校学报,2002(02):52-53.

[2]施媛媛.基于Proteus的负温度系数热敏电阻温度调理电路[J].中国高新科技,2020(04):39-41.

[3]董亮,雷良育,李雪原,刘兵,陈喜庭.基于Proteus的惠斯通电桥测温电路仿真设计[J].广西民族大学学报(自然科学版),2017,23(02):74-77.