

**装 置 设 计 个 人 报 告**

设计题目 温度报警器设计

班 级 自动化1503班

姓 名 权 红 飞

学 号 201505060321

**目录**

**[一、 设计目标](#_Toc4746_WPSOffice_Level1)** **[3](#_Toc4746_WPSOffice_Level1)**

**[二、设计内容](#_Toc22110_WPSOffice_Level1)** **[3](#_Toc22110_WPSOffice_Level1)**

**[三、项目组成员分工](#_Toc19005_WPSOffice_Level1)** **[3](#_Toc19005_WPSOffice_Level1)**

**[四、系统设计](#_Toc3583_WPSOffice_Level1)** **[4](#_Toc3583_WPSOffice_Level1)**

**[五、个人工作](#_Toc14350_WPSOffice_Level1)** **[5](#_Toc14350_WPSOffice_Level1)**

[5.1 LCD显示模块](#_Toc3583_WPSOffice_Level2) [5](#_Toc3583_WPSOffice_Level2)

[5.2 DS18B20测温模块](#_Toc14350_WPSOffice_Level2) [6](#_Toc14350_WPSOffice_Level2)

[5.3 上下限设定模块](#_Toc8079_WPSOffice_Level2) [8](#_Toc8079_WPSOffice_Level2)

**[六、硬件调试](#_Toc8079_WPSOffice_Level1)** **[10](#_Toc8079_WPSOffice_Level1)**

**[七、系统整体电路图](#_Toc9967_WPSOffice_Level1)** **[11](#_Toc9967_WPSOffice_Level1)**

[电路原理图](#_Toc19031_WPSOffice_Level2) [11](#_Toc19031_WPSOffice_Level2)

[PCB图](#_Toc6650_WPSOffice_Level2) [12](#_Toc6650_WPSOffice_Level2)

[打印预览图](#_Toc2606_WPSOffice_Level2) [12](#_Toc2606_WPSOffice_Level2)

**[八、总结](#_Toc19031_WPSOffice_Level1)** **[13](#_Toc19031_WPSOffice_Level1)**

**[九、参考文献 1](#_Toc6650_WPSOffice_Level1)4**

2. **设计目标**

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。温度的测试也已经越来越多的影响到各个领域。因此设计一个温度测试的系统是十分必要的。

二、**设计内容**

此设计主要做一个基于STC80C52单片机的数字温度检测报警器系统。本系统是基于单片机的智能温度报警控制器的设计。以STC80C52为核心，采用温度传感器DS18B20作为温度检测器，在液晶显示屏LCD1602上显示实时温度。并且设置上下限报警温度。

# **三、项目组成员分工**

|  |  |
| --- | --- |
| 成 员  任 务 | 温度检测装置整体设计 |
| 权 红 飞 | 软件编写  PCB板的制作  PCB板子的焊接  软硬件调试 |
| 王 盼 | 设计方案论证  前期各模块资料查询  PCB板的制作  PCB板子的焊接 |
| 刘 欢 | 设计方案论证  前期各模块资料查询  PCB板的制作  小组报告的整体框架撰写 |

**四、系统设计**

系统整体设计框图如图2-1所示：

时钟复电路

LCD显示

报警电路

单片机

数字时钟

测温电路

温度上下限设定

图1 系统整体设计框图

正常

低

高

开始

LCD初始化

LCD清屏

LCD显示

启动温度转换

与极限值比较

显示温度

LED 灯光报警

蜂鸣器声音报警

显示温度

LED 灯光报警

蜂鸣器声音报警

显示温度

图2 程序设计流程图

**五、个人工作**

**5.1 LCD显示模块**

1602LCD是指显示的内容为16X2,即可以显示两行，每行16个字符液晶模块（显示字符和数字）。1602液晶模块内部的字符发生存储器（CGROM)已经存储了160个不同的点阵字符图形，这些字符有：阿拉伯数字、英文字母的大小写、常用的符号、和日文假名等，每一个字符都有一个固定的代码，比如大写的英文字母“A”的代码是01000001B（41H），显示时模块把地址41H中的点阵字符图形显示出来。

特点：3.3V或5V工作电压，对比度可调；内含[复位电路](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=7725650&ss_c=ssc.citiao.link" \t "_blank)；提供各种控制命令,如：清屏、字符闪烁、光标闪烁、显示移位等多种功能；有80字节显示数据存储器DDRAM；内建有192个5X7点阵的字型的字符发生器CGROM；8个可由用户自定义的5X7的字符发生器CGRAM。

**LCD1602读写函数封装**

void write\_com(uchar com)

{

lcdrs=0; //写指令时RS=L

lcden=0;

P0=com; //给端口送指令

delay(5); //延时5MS

lcden=1; //使能端高电平有效

delay(5);

lcden=0;

}

void write\_date(uchar date)

{

lcdrs=1; //写数据时RS=L

lcden=0;

P0=date; //给端口送数据

delay(5); //延时5MS

lcden=1; //使能端高电平有效

delay(5);

lcden=0;

}

void init()

{

uchar num;

lcdrw=0;//写信号拉低

lcden=0;//使能端拉低

write\_com(0x38); //开显示

write\_com(0x0c); //0e开显示不显示光标

write\_com(0x06); //每写一个地址和光标加一

write\_com(0x01); //显示清零

write\_com(0x80); //第一行的初始地址为0x80

for(num=0;num<14;num++)//循环的数字要看上面的占多少个字符

{

write\_date(table1[num]);//多了和少了都不会对的。

delay(5);

}

write\_com(0x80+0x40); //第二行初始地址为0x80+0x40

for(num=0;num<14;num++)

{

write\_date(table2[num]);

delay(5);

}

}

**5.2 DS18B20测温模块**

在正常测温情况下，DS18B20的测温分辨力为0.5℃，可采用下述方法获得高分辨率的温度测量结果：首先用DS18B20提供的读暂存器指令(BEH)读出以0.5℃为分辨率的温度测量结果，然后切去测量结果中的最低有效位(LSB)，得到所测实际温度的整数部分Tz，然后再用BEH指令取计数器1的计数剩余值Cs和每度计数值CD。实际温度Ts可用式(2-1)计算：

Ts=（Tz-0.25℃）+(CD-Cs)/CD

**DS18B20读写函数封装**

void Init\_DS18B20(void)

{

unsigned char x=0;

DQ = 1; //DQ复位

delay1(8); //稍做延时

DQ = 0; //单片机将DQ拉低

delay1(80); //精确延时 大于 480us

DQ = 1; //拉高总线

delay1(10);

x=DQ; //稍做延时后 如果x=0则初始化成功 x=1则初始化失败

delay(5);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* 读一个字节 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

unsigned char ReadOneChar(void)

{

unsigned char i=0;

unsigned char dat = 0;

for (i=8;i>0;i--)

{

DQ = 0; // 给脉冲信号

dat>>=1;

DQ = 1; // 给脉冲信号

if(DQ)

dat|=0x80;

delay1(5);

}

return(dat);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* 写一个字节 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void WriteOneChar(unsigned char dat)

{

unsigned char i=0;

for (i=8; i>0; i--)

{

DQ = 0;

DQ = dat&0x01;

delay1(5);

DQ = 1;

dat>>=1;

}

delay1(5);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* 读取温度 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

unsigned int ReadTemperature(void)

{

unsigned char a=0;

unsigned int b=0;

unsigned int t=0;

Init\_DS18B20();

WriteOneChar(0xCC); // 跳过读序号列号的操作

WriteOneChar(0x44); // 启动温度转换

delay1(200);

Init\_DS18B20();

WriteOneChar(0xCC); //跳过读序号列号的操作

WriteOneChar(0xBE); //读取温度寄存器等（共可读9个寄存器） 前两个就是温度

a=ReadOneChar(); //低位

b=ReadOneChar(); //高位

b<<=8;

t=a+b;

return(t);

}

**5.3 上下限设定模块**

函数中先提前设置好温度上下限，也可以通过按键来改变上下限，按键在未按下之前，P1口全为高电平，为0xff，当其中按键按下后，电平发生改变，P1口不会为全1，CPU检测到电平的跳变后就可判断出是哪个按键按下：如果是第一个按键按下，为0xfd，写命令使其为温度上限的增加按键，以此类推。同时按键可能会有抖动，所以我们在设置按键时要进行消抖。DS18B20只能保证-125~55℃范围内的温度精度。

**轮询方式下扫描键盘函数**

void keyscan()

{

if(K1==0)

{

delay(20);

if(K1==0)

{

H\_WD++;

while(!K1);

}

}

if(K2==0)

{

delay(20);

if(K2==0)

{

H\_WD--;

while(!K2);

}

}

if(K3==0)

{

delay(20);

if(K3==0)

{

L\_WD++;

while(!K3);

}

}

if(K4==0)

{

delay(20);

if(K4==0)

{

L\_WD--;

while(!K4);

}

}

}

# 六、硬件调试

硬件调试主要是测试各硬件部分能否完成设计功能。

系统的性能调试以主程序为主。硬件调试比较简单，直接插上电，软件调试可以先编写显示程序并进行硬件的正确性检验，然后分别进行主程序、从程序的编写和调试，由于DS18B20与单片机采用串行数据传送，因此对DS18B20进行编程时必须严格地保证读写时序，否则将无法读取测量结果。

在焊接的过程中，每接一根线都用万用表测试是否短路或开路。这样焊接完了之后，就不必怀疑线路不通的问题了，由于线路很繁琐也免去了大量排查的工作。最初焊接最小系统，查看二极管能否正常闪烁，以此判断单片机是否正常工作。整体电路焊接完后，通电测试所有硬件芯片的输入输出电压是否在设计要求的范围内，判断单片机的工作与否。

**调试中遇到的问题：**

1. LCD引脚连接不正确： LCD默认的RS、RW、E引脚和单片机的连接需要和电路板里的连接口一样，即RS连接P2.4，RW连接P2.5，E连接P2.6，不可随意连接，否则电路板在烧录程序后不能显示温度。在我的程序中LCD1602的RS、RW、EN引脚最初是接的P2.1、P2.1、P2.3，因此不能正确显示。
2. DS18B20反接：根据硬件电路上的图像方向接，一旦接反就会立刻发热。我们第一次反接传感器感觉到发烫立刻取下，没有烧毁电路板酿成恶果。接反也是导致该传感器总是显示85℃的原因。实际操作中将正负反接，传感器立即发热，液晶屏不能显示读数，正负接好后显示85℃。另外如果使用51单片机的话，那么中间那个引脚必须接上4.7K—10K的上拉电阻，否则高电平不能正常输入/输出，要么通电后立即显示85℃，要么用几个月后温度在85℃与正常值上乱跳。

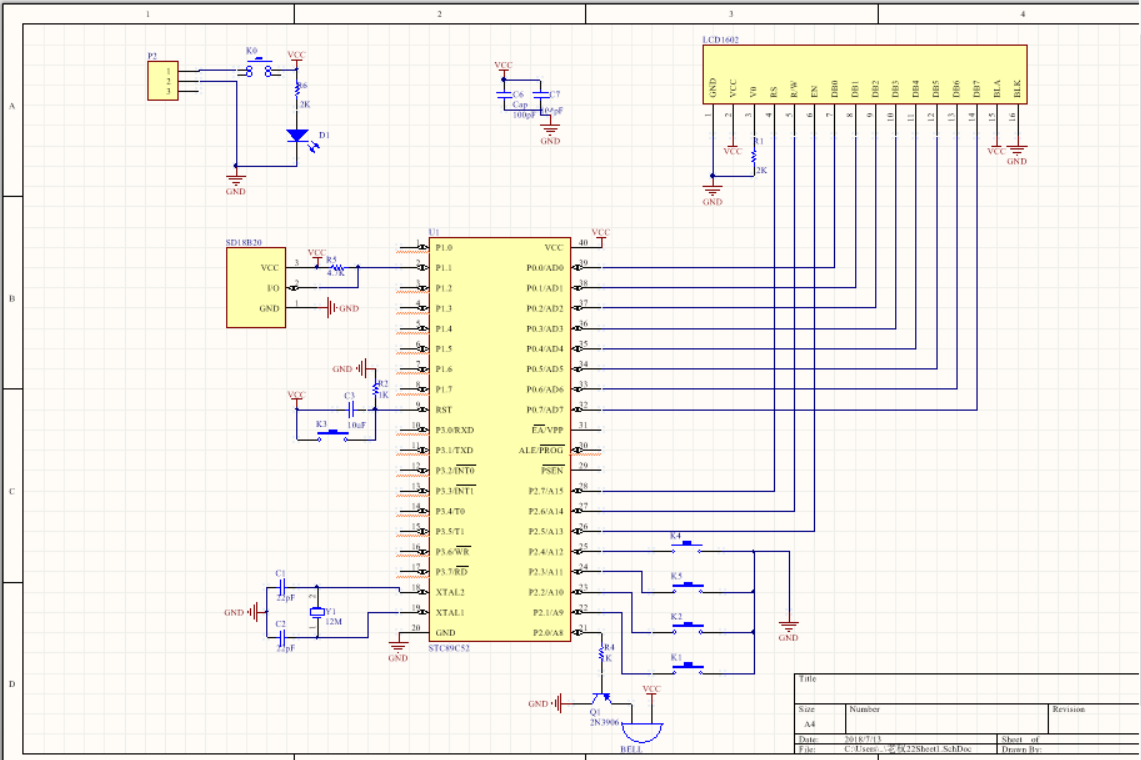
（3）编写LCD显示闪烁时：有时上下限显示会变成随机数，这是因为if\else语句位置不正确。有时候会留下上一次显示的数据，这是因为第二次语句的空隙处没有用空格进行覆盖，因此上一次的数据依然在显示。显示过快，是因为延时函数延时不够准确。

**七、系统整体电路图**

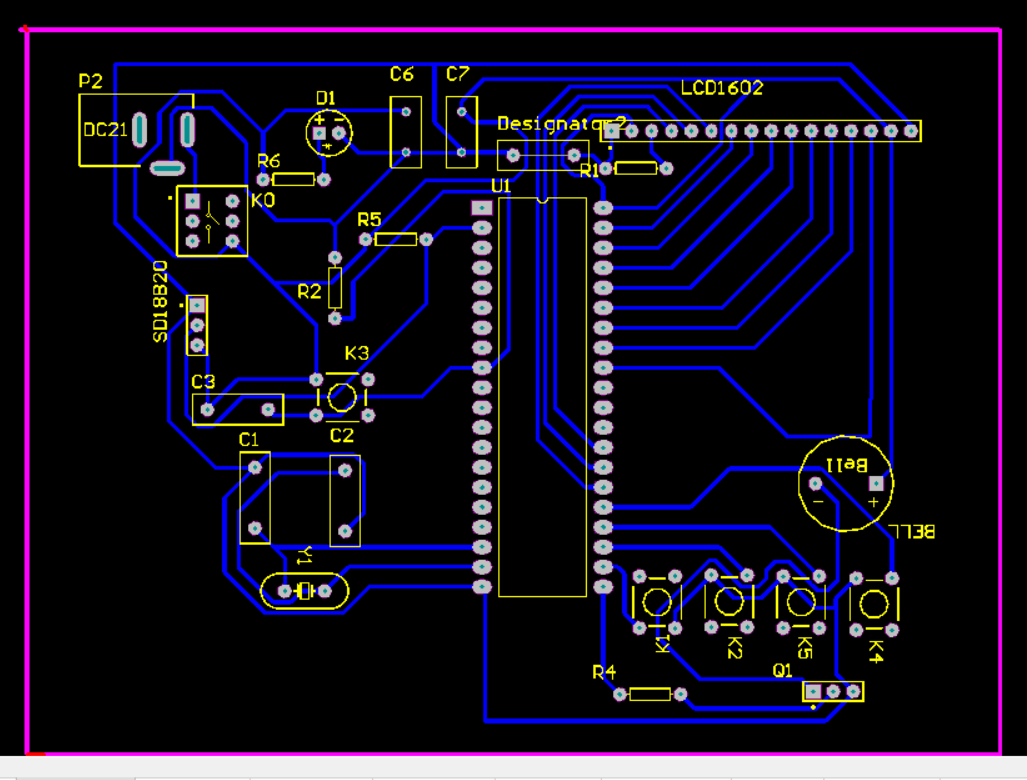
该系统电路主要包括:单片机最小系统、DS18B20温度传感器系统、报警系统、数码管显示模块电路及电源接口和数据下载接口等电路，如图2.1所示。

图2.1中有四个独立式按键可以分别调整温度计的上下限报警设置，图中蜂鸣器可以在被测温度不在上下限范围内时，发出报警呜叫声音，同时LCD1602将被测温度值显示，这时可以调整报警上下限，从而测出被测的温度值。

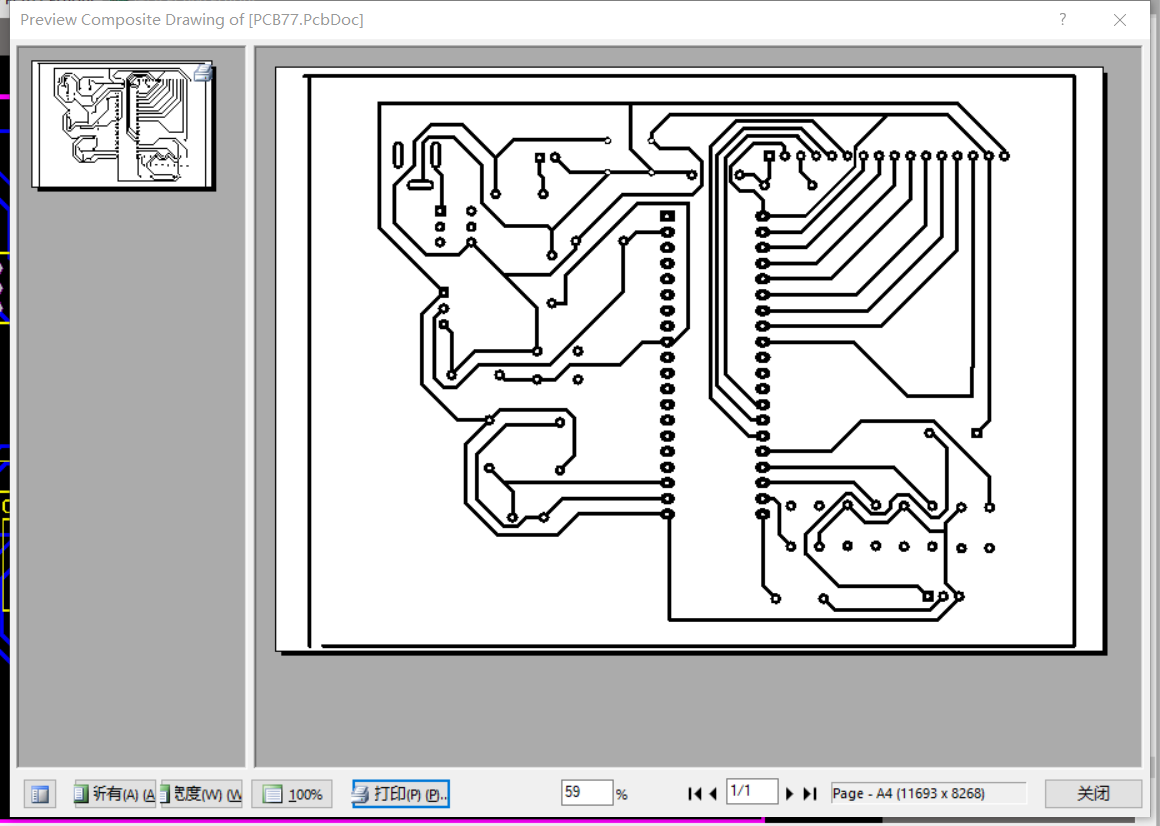
图2.1中的按健复位电路是.上电复位加手动复位，使用比较方便，在程序跑飞时，可以手动复位，这样就不用在重起单片机电源，就可以实现复位。



电路原理图



PCB图



打印预览图

# **八、总结**

“温度报警器”装置设计中，小组三人全程参与了硬件部分电路原理图的设计到软件部分软件的设计再到最后的焊接和调试的全部过程。对于设计工作过程，大致的可以分为以下几个阶段：

首要阶段，首先对课题的项目的设想与研究，通过参考大量的资料，拟定一个自己心目中理想的方案，对于自己想设计的器件实现的功能有个具体全面的认识。

第二阶段，通过网络、书籍、老师提供和的ppt,查找相应的硬件元件，全面的了解每个元器件的功能(包括各个引脚功能、总体实现的功能)与具体的工作结构（元器件的内部结构及工作时的复位、脉冲、初始化等）。

第三阶段，利用AD10等软件对该温度报警器进行电路原理图的绘制与调试，通过该软件测试制作一张连接正确的原理图及各种元器件的封装。

第四阶段，根据设计的原理图进行软件部分的设计，通过keil软件对程序进行编译与调试无误后加载到单片机中，调试到达预想的效果。

通过这次装置设计实训，我懂得了理论与实际相结合是很重要的，只有理论知识是远远不够的，只有把所学的理论知识与实践相结合起来，从理论中得出结论，从而提高自己的实际动手能力和独立思考的能力。我在设计的过程中遇到了许多问题，同时也发现了自己的不足之处，对以前所学过的知识理解得不够深刻，掌握得不够牢固，比如说不懂一些元器件的使用方法，对C语言掌握得不透彻。这次的课设也使我对单片机有了更加浓厚的兴趣，今后也会不断的去学习单片机，增加自己的知识储备，逐步的去提高自己。

# **九、参考文献**

[1] 戴佳,苗龙,陈斌.51单片机应用系统开发典型实例[M]. 中国电力出版社,2005.9.

[2] 高玉芹．单片机原理及应用及C51编程技术．北京：机械工业出版社，2011.6.

[3] 孙育才. MCS-51系列单片微型计算机及其应用[M]. 东南大学出版社, 2004.6．

[4] 沈红卫. 单片机应用系统设计实力与分析[M]. 北京:北京航空航天大学出版社，2003.

[5] 曾一江. 单片微机原理与接口技术[M]. 北京：科技出版社,2009.12．

[6] 赵文博，刘文涛．单片机语言C51程序设计[M]．北京：人民邮电出版社，2005.10．

[7] 徐爱钧, 彭秀华. 单片机高级语言C51应用程序设计[M].北京航空航天大学出版社，2006.10.

[8] 谭浩强. C语言程序设计[M],北京清华出版社 2000.