

**装置设计小组报告**

设计题目 温度报警器设计

班 级 自动化1503班

小组成员 权红飞 201505060321

王 盼 201505060305

刘 欢 201505060302

指导老师 毛艳慧

2108年9月28日

**课程设计任务书**

|  |  |  |
| --- | --- | --- |
| 题 目 | 基于单片机的温度检测装置设计 | |
| 小组成员 | 权红飞、王盼、刘欢 | |
| 设  计  内  容  与  要  求 | 1. 基于单片机、晶振、液晶显示、温度传感器等模块设计并实现温度测量及显示；温度上下限可预置，超出设定范围声光报警；功能可自行扩展。 2. 以项目小组的形式，每组6人左右。项目组成员按个人意愿自由组合。能与团队成员分享信息、倾听他人意见和建议，体现团队合作精神。 3. 综合应用电路分析与电路设计的基础知识，设计原理电路并分析,能使用AD/DXP/Protel/Proteus电路设计软件进行电路原理图设计、电路仿真和PCB设计。 4. 能正确使用PCB制板设备、电路制作工具和电路调试工具，具有PCB制板、元器件焊接、电路调试的工程实践技能。具有工程实习经历。 5. 整个设计过程中，能正确处理制板过程中产生的电子废品/废液，按规程正确操作PCB制板设备，培养环境保护意识，具有可持续发展的理念。 6. 整理设计结果，撰写并提交课程设计报告。   参考资料：  [1] 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  [2] 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． | |
| 起止时间 | | 年 月 日 至 年 月 日 |
| 指导教师签名 | | 年 月 日 |
| 系（教研室）主任签名 | | 年 月 日 |
| 学生签名 | | 年 月 日 |

# 

**目录**

[1 课程设计的目的 1](#_Toc20922)

[2 设计内容以及设计方案论证 1](#_Toc11406)

[2.1 设计内容 1](#_Toc23500)

[2.2 设计方案论证 1](#_Toc24129)

[2.2.1 数字温度计设计方案论证 1](#_Toc13449)

[2.2.2 单片机的选择 1](#_Toc11819)

[3 系统整体方案设计 2](#_Toc19329)

[3.1 设计目标 2](#_Toc5566)

[3.2 系统各模块方案的选择 3](#_Toc19838)

[3.2.1 控制模块STC89C52 3](#_Toc11080)

[3.2.2 温度测量模块DS18B20 4](#_Toc27004)

[3.2.3 LCD显示模块 7](#_Toc13336)

[3.2.4 报警模块 8](#_Toc24777)

[4 系统主要单元电路设计 8](#_Toc8346)

[4.1 单片机控制模块电路的设计 8](#_Toc3372)

[4.2 DS18B20数字测温电路的设计 9](#_Toc2646)

[4.3 LCD1602显示电路的设计 10](#_Toc13669)

[4.4 报警电路的设计 10](#_Toc19203)

[4.5 电源电路的设计 10](#_Toc2684)

[4.6 键盘电路的设计 10](#_Toc15621)

[5 系统的软件设计 10](#_Toc15548)

[5.1 KEIL软件介绍 11](#_Toc30995)

[5.2 系统程序设计流程图 11](#_Toc19786)

[6 元器件清单 12](#_Toc16858)

[7 调试过程与结果 12](#_Toc10684)

[7.1 电路的调试 12](#_Toc5584)

[7.2 软件调试 13](#_Toc18715)

[8 调试中遇到的问题及解决方法 14](#_Toc24729)

[9 项目组成员及分工 15](#_Toc32302)

[10 总结 15](#_Toc20658)

[11 参考文献 17](#_Toc4365)

[附录 18](#_Toc15157)

[附录一 18](#_Toc579)

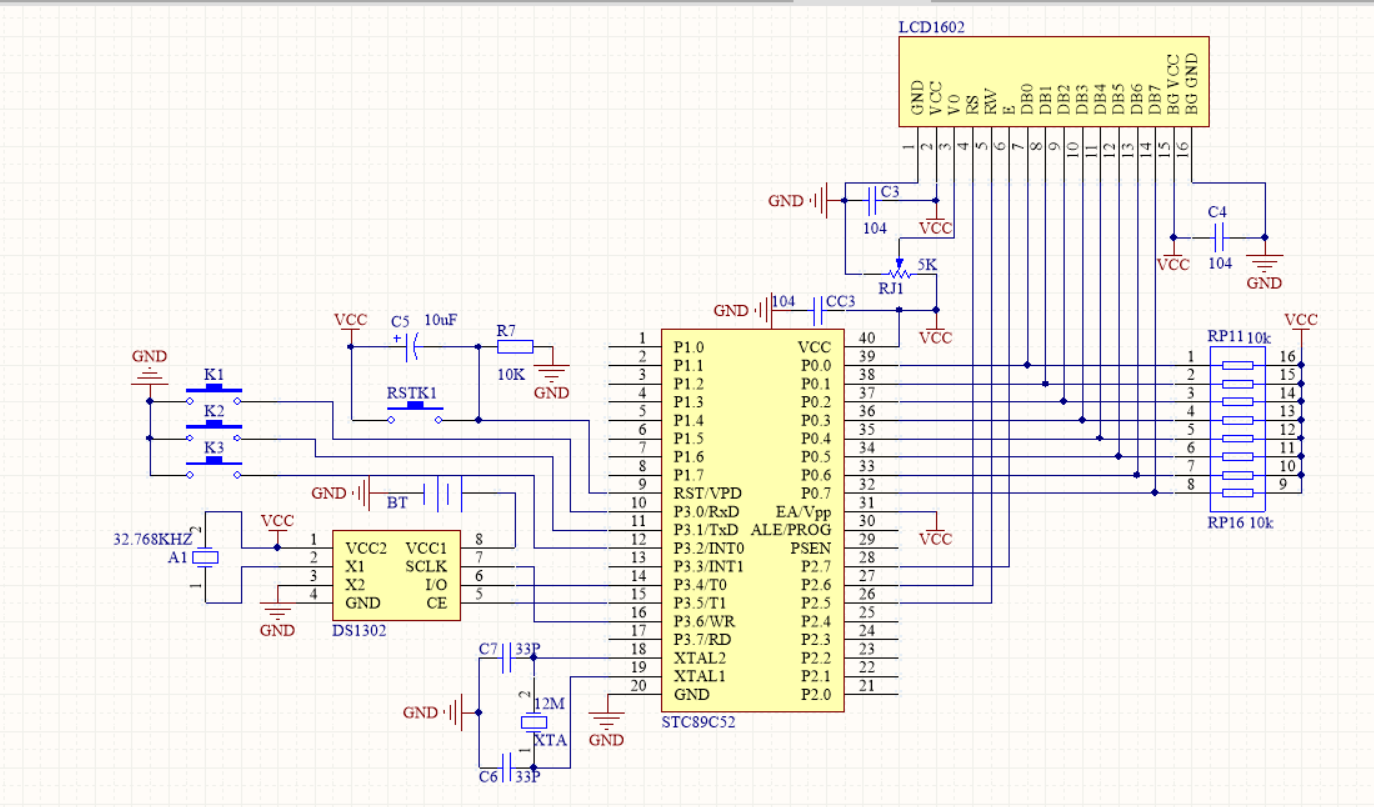
[4 实物演示效果图 19](#_Toc1778)

[附录二 21](#_Toc11277)

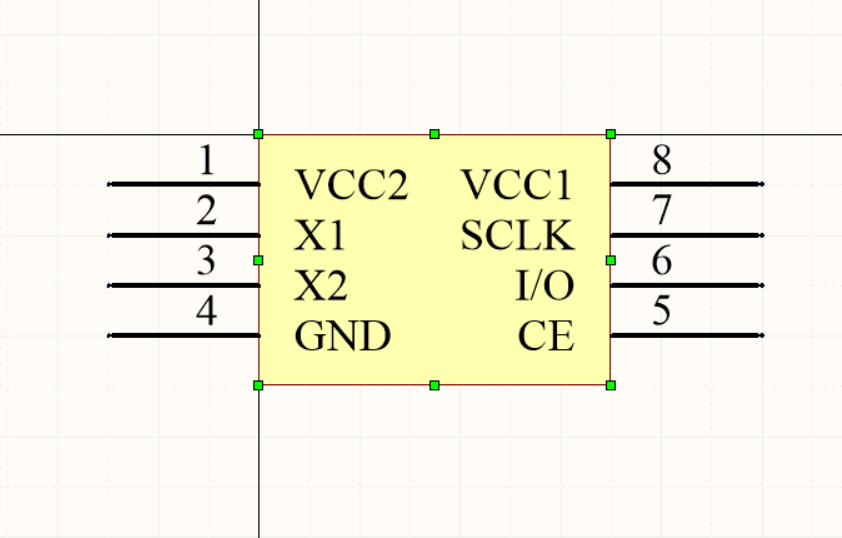
# 1 系统任务

1. 掌握单片机I/O端口的编程方法；
2. 掌握DS1302时钟的基本特点及读写方法；
3. 掌握LCD液晶显示屏的驱动方法并显示年、月、日、星期、时、分、秒；
4. 通过独立按键可以校队时间；

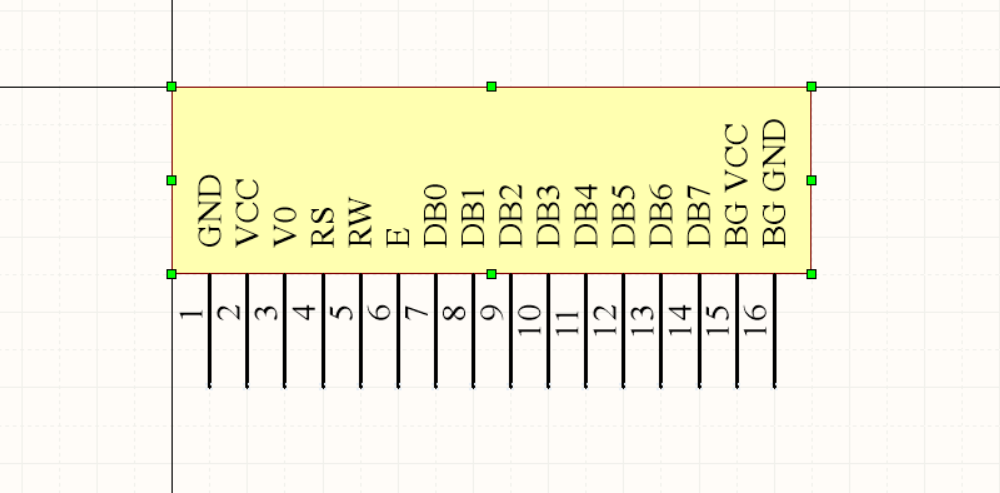
# 2 电路原理图



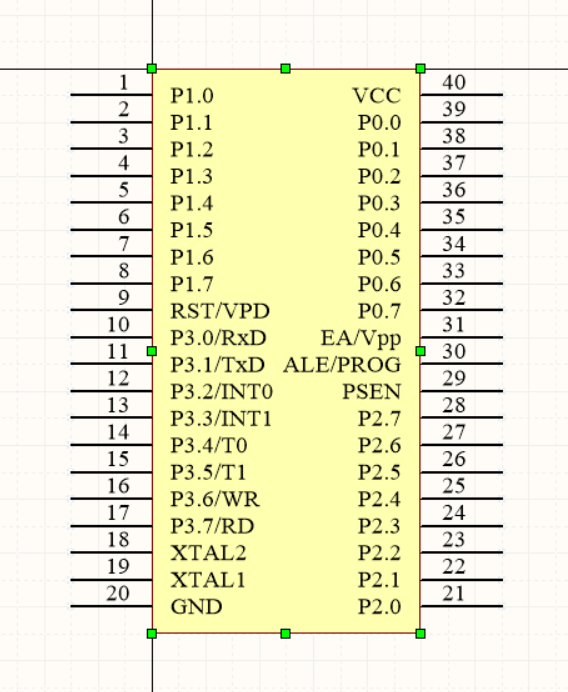
2.1总原理图



2.2 DS1302原理图



2.3 LCD1602原理图



2.4 STC89C52原理图

## 2.1 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。

## 2.2 设计方案论证

# 3 系统整体方案设计

## 3.1 设计目标

此设计采用STC80C52单片机薇主控芯片，结合外围电路时钟复位电路、LCD1602液晶显示电路、报警电路、DS18B20数字温度传感器等部分电路来实现温度的检测并在LCD1602上进行实时显示，通过设定温度上下限来检测温度是否超过上下限温度，若不符合则报警的一个数字温度检测报警器系统，可以实现:

## 3.2 系统各模块方案的选择

### 3.2.1 控制模块STC89C52

STC89C52是一种带8K字节闪烁可编程可檫除只读存储器的低功耗，高性能COMOS8的微处理器，俗称单片机。该器件采用ATMEL搞密度非易失存储器制造技术制造，与工业标准的MCS-51指令集和输出管脚相兼容。

STC89C52具体介绍如下：

① 主电源引脚（2根）

VCC(Pin40)：电源输入，接＋5V电源；

GND(Pin20)：接地线；

②外接晶振引脚（2根）

XTAL1(Pin19)：片内振荡电路的输入端；

XTAL2(Pin20)：片内振荡电路的输出端；

③控制引脚（4根）

RST/VPP(Pin9)：复位引脚，引脚上出现2个机器周期的高电平将使单片机复位；

ALE/PROG(Pin30)：地址锁存允许信号；

PSEN(Pin29)：外部存储器读选通信号；

EA/VPP(Pin31)：程序存储器的内外部选通，接低电平从外部程序存储器读指令，如果接高电平则从内部程序存储器读指令；

④可编程输入/输出引脚（32根）

STC89C52单片机有4组8位的可编程I/O口，分别位P0、P1、P2、P3口，每个口有8位（8根引脚），共32根。

P0口（P39～P32）：8位双向I/O口线，名称为P0.0～P0.7

P1口（P1～P8）：8位准双向I/O口线，名称为P1.0～P1.7

P2口（P21～P28）：8位准双向I/O口线，名称为P2.0～P2.7

P3口（P10～P17）：8位准双向I/O口线，名称为P3.0～P3.7

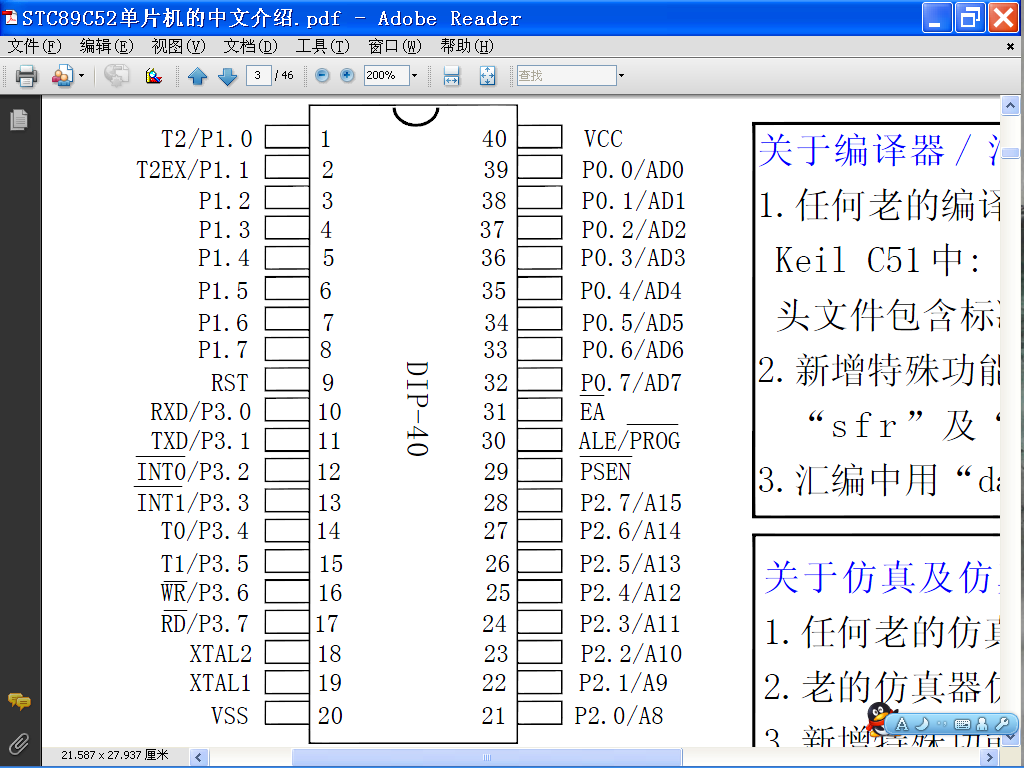


图3.2.1 STC89C52的引脚图与实物图

STC89C52包含以下部分：

（1）一个8位微处理器CPU；

（2）片内数据存储器RAM和特殊功能寄存器SFR；

（3）片内程序存储器ROM；

（4）两个定时/计数器T0、T1，可用作定时器，也可用以对外部脉冲进行计数；

（5）四个8位可编程的并行I/O端口，每个端口既可作输入，也可作输出；

（6）一个串行端口，用于数据的串行通信；

（7）中断控制系统；

（8）内部时钟电路；

### 3.2.3 LCD显示模块

1602液晶也叫1602字符型液晶，它是一种专门用来显示字母、数字、符号等的点阵型液晶模块。它由若干个5X7或者5X11等点阵字符位组成，每个点阵字符位都可以显示一个字符，每位之间有一个点距的间隔，每行之间也有间隔，起到了字符间距和行间距的作用，正因为如此所以它不能很好地显示图形（用自定义CGRAM，显示效果也不好）。

LCD1602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。

市面上字符液晶大多数是基于HD44780液晶芯片的，控制原理是完全相同的，因此基于HD44780写的控制程序可以很方便应用于市面上大部分的字符型液晶。

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。



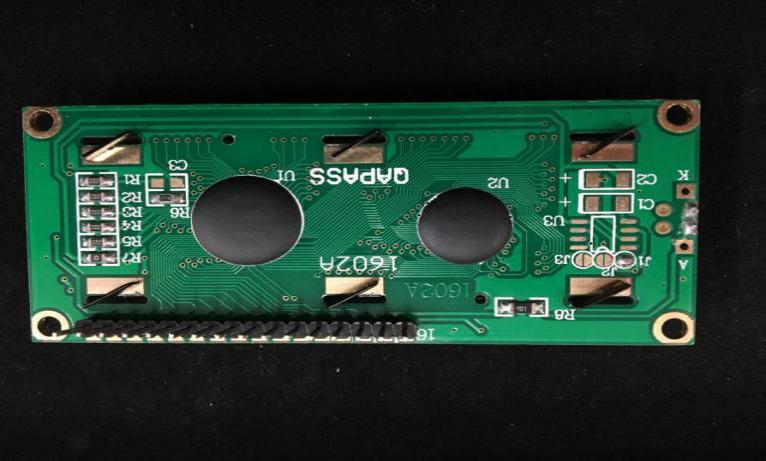


图3.2.5 LCD1602实物图

### 3.2.4 DS1302时钟模块

4 系统主要单元电路设计

## 4.1 单片机控制模块电路的设计

单片机最小系统,或者称为最小应用系统，是指用最少的元件组成的单片机可以工作的系统。对52系列单片机来说，最小系统一般应该包括:单片机、晶振电路、复位电路。故STC89C52单片机最小系统电路图如图4.2所示。STC89C52单片机具有抗干扰性能强、速度快、功耗低等特点。时钟频率最高可达80MHz。

单片机最小系统电路介绍：

52单片机最小系统复位电路的极性电容C1的大小直接影响单片机的复位时间，一般采用10～30uF，52单片机最小系统晶振Y1也可以采用6MHz或者11.0592MHz，在正常工作的情况下可以采用更高频率的晶振。52单片机最小系统起振电容C2、C3一般采用15～33pF，并且电容离晶振越近越好，晶振离单片机越近越好。

说明：

（1）晶振电路：为系统提供基本的时钟信号，晶振为12MHz。

（2）复位电路：是单片机正常运行的一个必要部分，复位电路应该保证单片机在上电瞬间进行一次有效的复位，在单片机正常工作时将RST引脚置低。此外通过一个按键进行手动复位，在单片机运行不正常时使用。

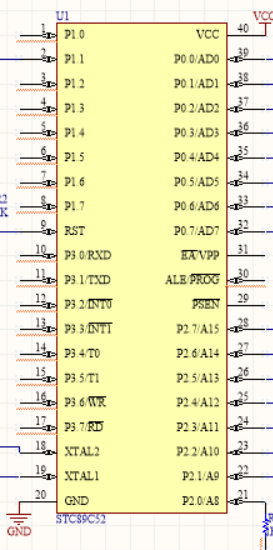
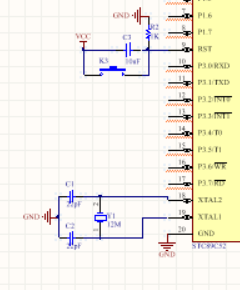
 

图4.1 单片机控制模块电路 图4.2 单片机最小系统电路

## 4.2 DS1302时钟的设计

## 4.6 键盘电路的设计

键盘电路由4个按键组成，分别对应为如下功能：

K1 +时间

K2 -时间

K3 打开修改时间开关

K4 修改位选端

**5 系统的软件设计**

## 5.2 系统程序设计流程图

通过上述原理，已对实时时钟的整体运行环境有了充分认识，绘制主程序流程图，利用Keil软件，对单片机进行编程，如图5.1为主程序功能流程图：

开始

初始化

DS1302时钟初始化

LCD初始化并显示时间

K3按键修改时间

K4按键位选

K1 +

K2 -

修改值回写

DS1302寄存器

修改值回写

DS1302寄存器

正常

显示时间

图5.1 主程序功能流程图

# 6 元器件清单

|  |  |  |
| --- | --- | --- |
| 原件名称 | 元件型号 | 数量 |
| 电容 | 100pF | 2 |
| 电容 | 10uF | 1 |
| 晶振 | 12MHZ | 1 |
| 电源开关 | SW-灰色 | 1 |
| 电容 | 22pF | 2 |
| 蜂鸣器 | 蜂鸣器 | 1 |
| 三极管 | 2N3906 | 1 |
| 温度传感器 | DS18B20 | 1 |
| 按键 | SW-PB 独立按键 | 5 |
| 电阻 | 4.7K | 1 |
| 电阻 | 1K | 2 |
| 电阻 | 2K | 2 |
| 芯片 | STC89C52 | 1 |
| 液晶显示仪 | LCD1602 | 1 |
| 电源接口 | 5V-DC | 1 |

# 

# 7 调试过程与结果

调试是指系统的调整、改进与测试。测试是在电路组装后对电路的参数与工作状态进行测量，调整则是在测试的基础上对电路的某些参数进行修正，使满足设计要求。在进行调试前应拟订出测试项目、测试步骤、调试方法和所用仪器等，做到心中有数，保证调试工作圆满完成。

## 7.1 电路的调试

电路的调试方法有两种。第一种是边安装边调试的方法。它是把复杂的电路按原理框图上的功能分成单元进行安装和调试，在单元调试的基础上逐步扩大安装和调试的范围，最后完成整机调试。这种方法在新设计的电路中比较常用。第二种方法是在整个电路系统全部焊接完毕后，实行一次性调试。这种方法比较适用于电路相对来说比较简单，系统不复杂的电路调试。

电路调节的步骤：

（1）通电观察

在电路安装没有错误的情况下接通电源（先关断电源开关，待接通电源连线之后再打开电路的电源开关）。但接通电源后不要立即进行电路功能的测试，不要急于通电，首先要根据原理电路认真对照检查电路中的接接线是否正确，包括错线（连线一端正确、另一端错误），少线（安装时漏掉的线），多线（连线的两端在电路图上都是不存在的）和短路（特别是间距很小的引脚及焊点间），并且还要检查每个元件引脚的使用端数是否与图纸相符。查线时最好用指针式万用表“Ω×1”档进行检查, 或是用数字万用表“Ω”档的蜂鸣器来测量，而且要尽可能直接测量元器件引脚，这样同时可以发现接触不良的地方。

（2）通电观察

在电路安装没有错误的情况下接通电源（先关断电源开关，待接通电源连线之后再打开电路的电源开关）。但接通电源后不要立即进行电路功能的测试，首先要充观察整个电路有无异常现象，电路中元器件是否有发热烧坏等现象，是否有漏电现象，电源是否有短路和开路现象等。如果电路在测试过程中出现异常，首先应该立即关闭电源，检查后排除故障再重新通电测试。然后再按要求测量各元器件引脚电源的电压，而不只是测量各路总电源电压，以保证元器件正常工作。

（3）单元电路调试

在调试单元电路时应明确本部分的调试要求。调试顺序应按照电路原理图中信号流向进行，这样可以把整个电路进行分步调试，把前面调试好的电路的输出信号作为后一级电路的输入信号。从而保证电路的调试更加顺利方便。单元调试包括静态和动态调试。静态调试一般是指在没有外加信号的条件下测试电路各点的电位，特别是有源器件的静态工作点。通过它可以及时发现已经损坏和处于临界状态的元器件。动态调试是用前级的输出信号或自身的信号测试单元的各种指标是否符合设计要求，对于信号产生电路一般只看动态指标。把静态和动态测试的结果与设计的指标加以比较，经深入分析后对电路与参数提出合理的修正。在调试过程中应有详尽记录。

（4）整机联调

各单元电路调试好以后，并不见得由它们组成的整体电路性能一定会好，因此还要进行整体电路调试。整体电路调试主要是观察和测量动态性能，把测量的结果与设计指标逐一对比，找出问题及解决办法，然后对电路及其参数进行修正，直到全部电路的性能完全符合设计要求为止。

## 7.2 软件调试

电路主要的模块有三个：报警模块、单片机最小系统、温度检测模块。

对AT89S52单片机进行烧录后，将程序刻入芯片内，再进行调试。

本设计是在Keil C环境下开发的，Keil C软件支持C语言的编程及调试，运用方便，是做C语言程序设计的首选。设计的首要任务是安装和学习使用这个软件，在课程设计前期已经学会了软件的使用和调试，在编译完Keil C后，再运用STC\_ISP\_V480软件烧录到开发板上，实现实物与程序的连接。在烧录前要对STC\_ISP\_V480进行一些必要的设置。

第一步：设置MCU Type为STC89C52RC；

第二步：打开编写好并编译的程序文件，它是以.hex为后缀的文件；

第三步：选择对应的COM端口，（可在我的电脑的设备管理处查看COM选项）；

第四步：点击Download/下载，等提示并给MCU上电时，打开开发板上的开关，它就自行烧录了。当烧录程序完成后,我们就可以启动硬件看看是否满足设计的要求,然后进一步的调试。

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。

# 8 调试中遇到的问题及解决方法

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。通过不断调试程序解决了此问题。

# 9 项目组成员及分工

|  |  |
| --- | --- |
| 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 |

# 10 总结

在此次“温度报警器”装置设计中，我们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：

第一阶段，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、以及大量相关资料，查找并确定相应的硬件元器件，充分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）；并通过大量的单片机的编程资料来确定软件程序的结构。

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，通过该软件的测试并制作一张连接正确的原理图及各种元器件的封装。并开始编写温度报警器的程序。

第四阶段，通过AD10软件制作PCB图，并制作电路板。

第五阶段，在已制作好的电路板上焊接元器件，并调试电路。

第六阶段，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，我们小组成员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。

# 11 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,2014.6。

[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] 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

[6] 曾一江 《单片微机原理与接口技术[M]》 北京：科技出版社 2009

[7] 康华光 《电子技术基础（模拟部分）[M]》 北京：高等教育出版社 2004

[8] 江晓安，董秀峰 《模拟电子技术[M]》 西安电子科技大学出版社 2001

[9] 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

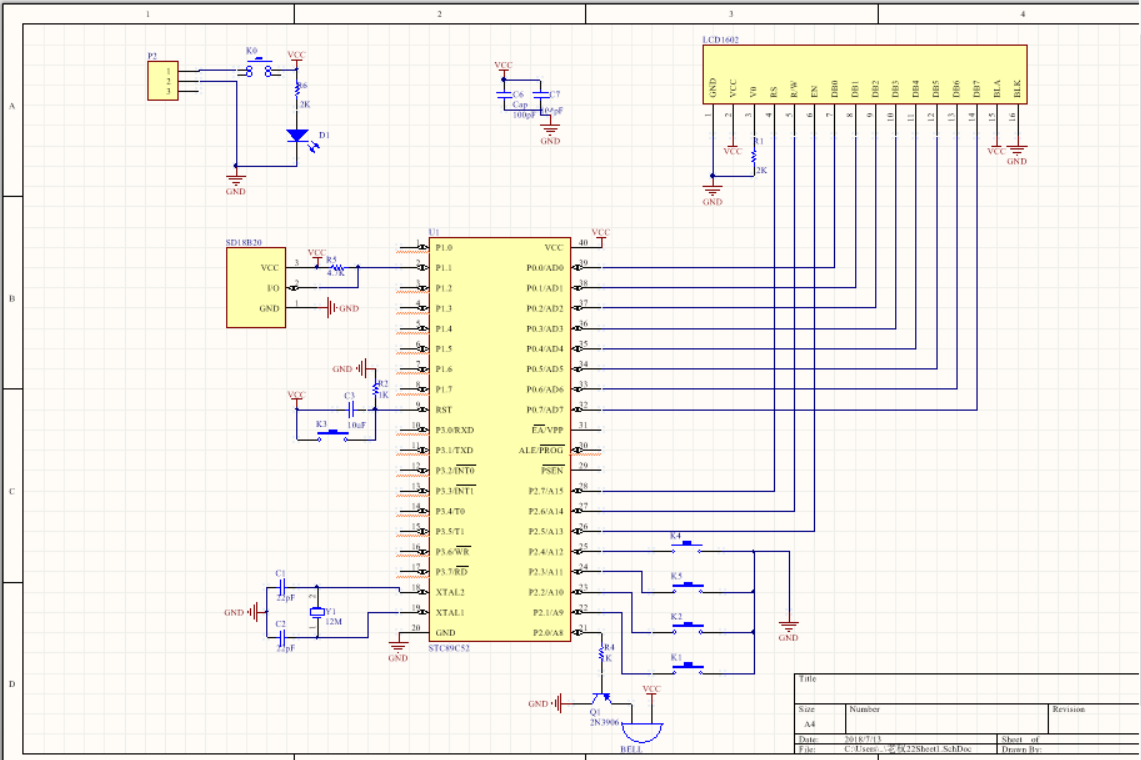
[10] 谭浩强 《C语言程序设计[M]》，北京清华出版社 2000

[11] 沙占友，孟志永 《单片机外围电路设计》 北京电子工业出版社 2006

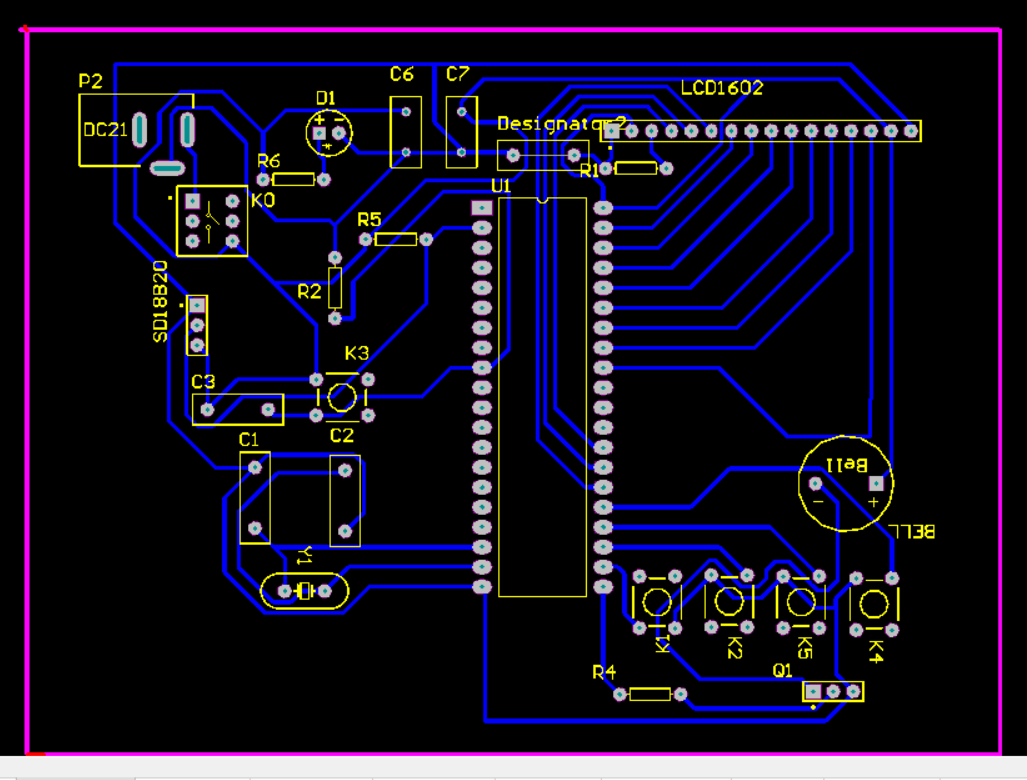
# 附录

**附录一**

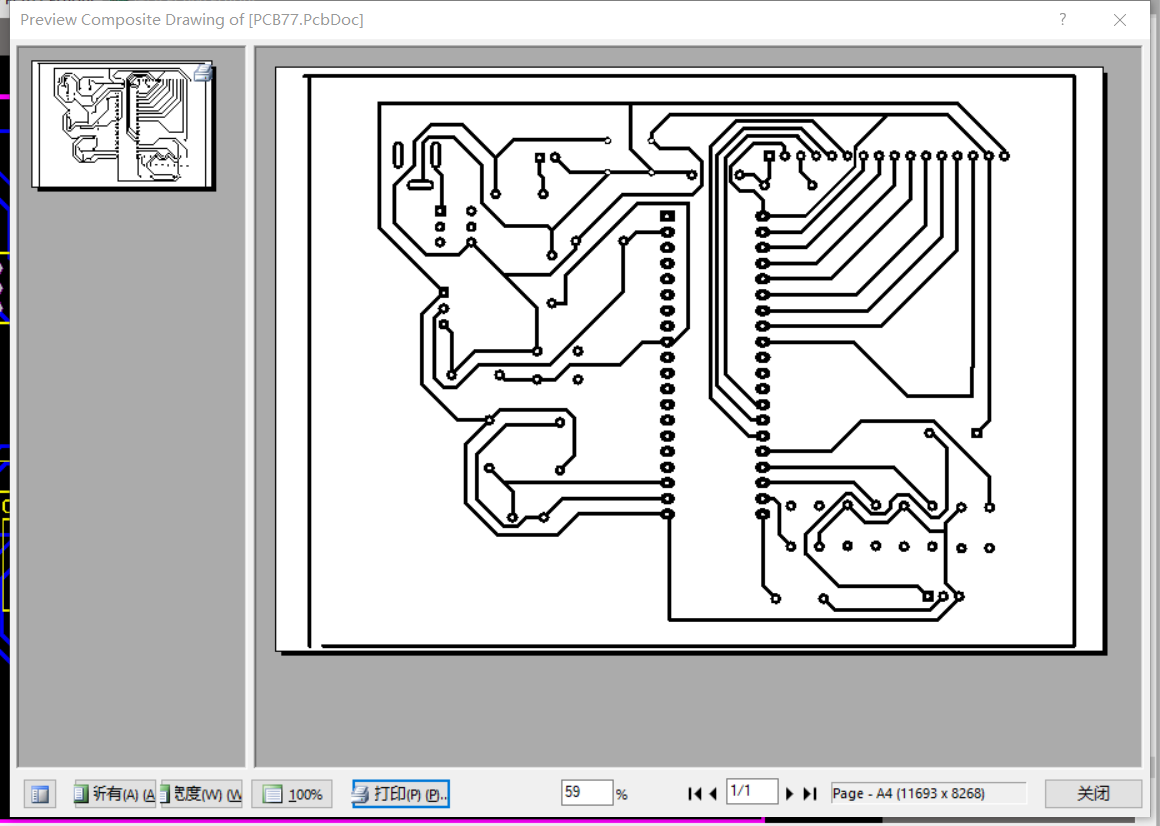
**1 电路原理图**



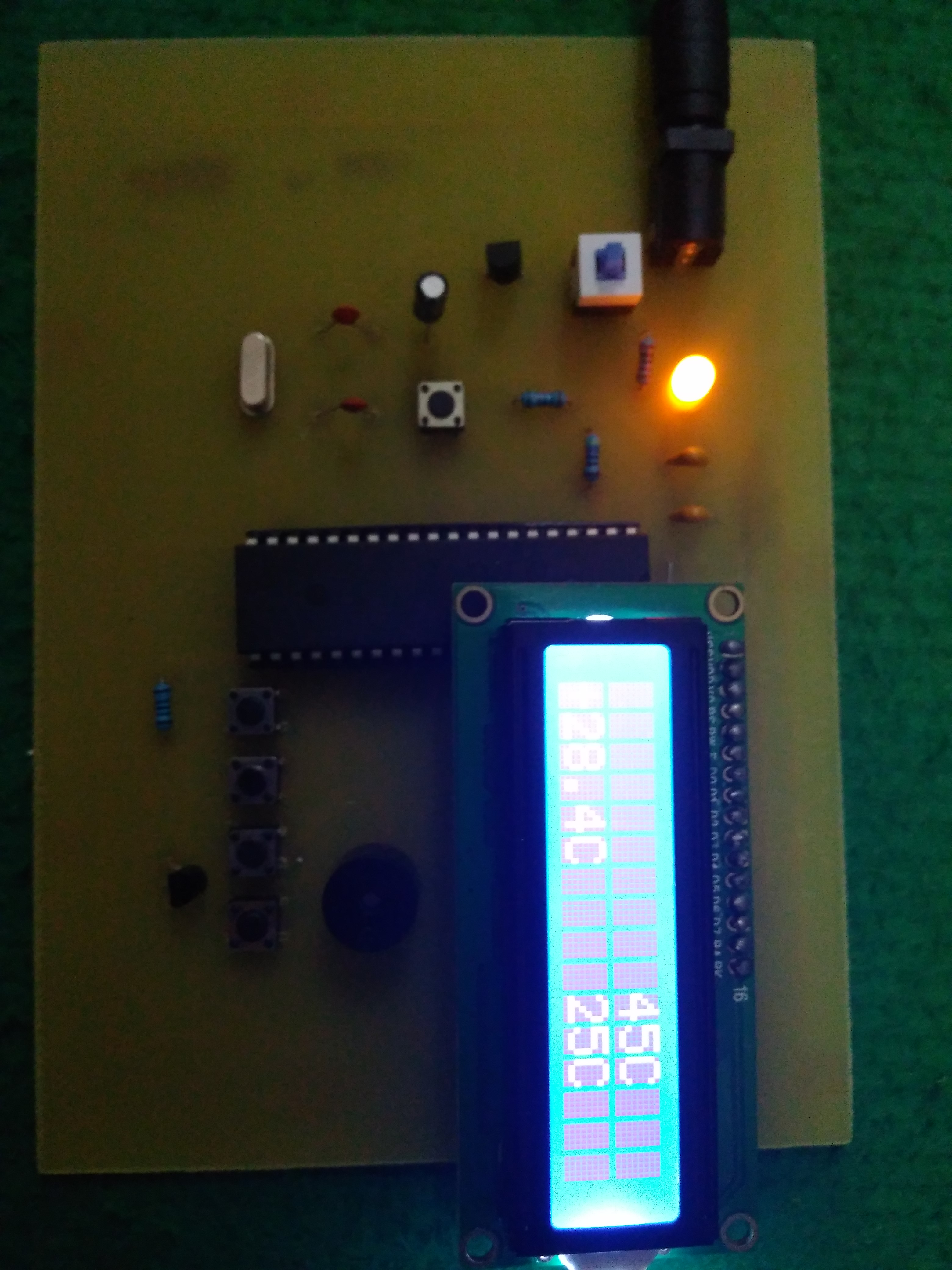
**2 PCB图**

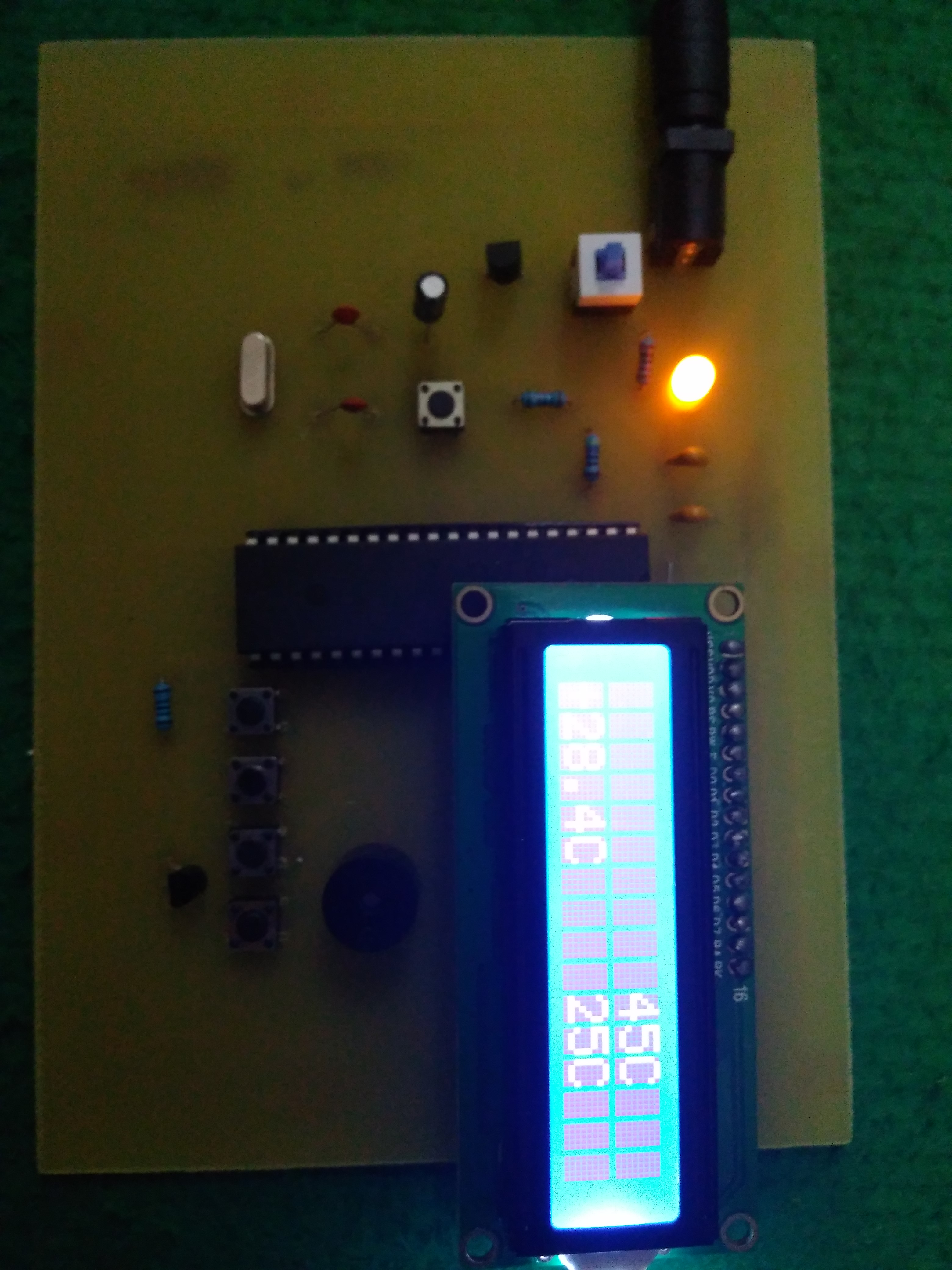


**3 打印预览图**



**4 实物演示效果图**





**附录二**

#include<reg52.h>

#define uint unsigned int

#define uchar unsigned char

sbit lcdrs=P2^7;//数据命令端口

sbit lcdrw=P2^6;//读写端口.我们在这里只写/

sbit lcden=P2^5;//使能端口。

sbit DQ=P1^1;//ds18b20 端口

sbit K1=P2^2;//按键控制报警阈值，高低温阈值的加减 高温阀值加

sbit K2=P2^1;//高温阀值减

sbit K3=P2^4;//低温阀值加

sbit K4=P2^3;//低温阀值减

sbit beep=P2^0;

int H\_WD=45,L\_WD=25,flag=1;

uint temp,Temp\_H,Temp\_L;

uchar code table1[]=" ";

uchar code table2[]=" ";

void delay(uint z)

{

uint x,y;

for(x=0;x<z;x++)

for(y=0;y<110;y++);

}

void delay1(unsigned int i)//延时函数

{

while(i--);

}

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);

}

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);

}

}

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);

}

}

}

void main()

{

init();

beep=1;

while(1)

{

temp=ReadTemperature();//温度读取

if(temp&0x8000)//判断

{

flag=0;//负号标志

temp=~temp; // 取反加1

temp +=1;

}

else

{

temp=temp;

flag=1;//温度为正标志位1

}

Temp\_H=temp>>4;//温度高4位

write\_com(0x80+0x40);

if(flag==0)

{

write\_date(0xB0);

}//负号标志

else

{

write\_date(0x20+11);

}//正号标志.

write\_com(0x80+0x41);

write\_date(0x30+Temp\_H/10);//显示温度的十位

write\_date(0x30+Temp\_H%10);//显示温度的各位

Temp\_L=temp&0x0F;

Temp\_L=Temp\_L\*16;//小数近似处理

write\_com(0x80+0x43);

write\_date(0x2e);//显示小数点

write\_date(0x30+Temp\_L/10);//显示温度的小数

write\_date(0x40+3);

keyscan();//按键扫描

if((Temp\_H\*10+Temp\_L/10)>(H\_WD\*10)&&(flag==1)||(Temp\_H\*10+Temp\_L/10)<(L\_WD\*10)&&(flag==1))//温度判断

{

beep=0;

delay(500);

}

else

{

beep=1;//蜂鸣器不报警

}

write\_com(0x80+10);//高上线

write\_date(0x30+H\_WD/10);

write\_date(0x30+H\_WD%10);

write\_date(0x40+3);

write\_com(0x80+0x40+10);//低下线

write\_date(0x30+L\_WD/10);

write\_date(0x30+L\_WD%10);

write\_date(0x40+3);

}

}