

西安邮电大学

毕业设计（论文）

题目：基于 51 单片机倒车防撞系统设计

学院：自动化学院

专业：自动化

班级：自动 1403

学生姓名：马栎雯

学号：06141085

导师姓名：侯铁双 职称：讲师

起止时间：2017 年 12 月 5 日 至 2018 年 6 月 10 日

毕业设计（论文）声明书

本人所提交的毕业论文《基于 51 单片机倒车防撞系统设计》是本人在指导教师指导下独立研究、写作的成果，论文中所引用他人的文献、数据、图件、资料均已明确标注；对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式注明并表示感谢。

本人完全理解《西安邮电大学本科毕业设计（论文）管理办法》的各项规定并自愿遵守。

本人深知本声明书的法律责任，违规后果由本人承担。

论文作者签名：

日期： 年 月 日

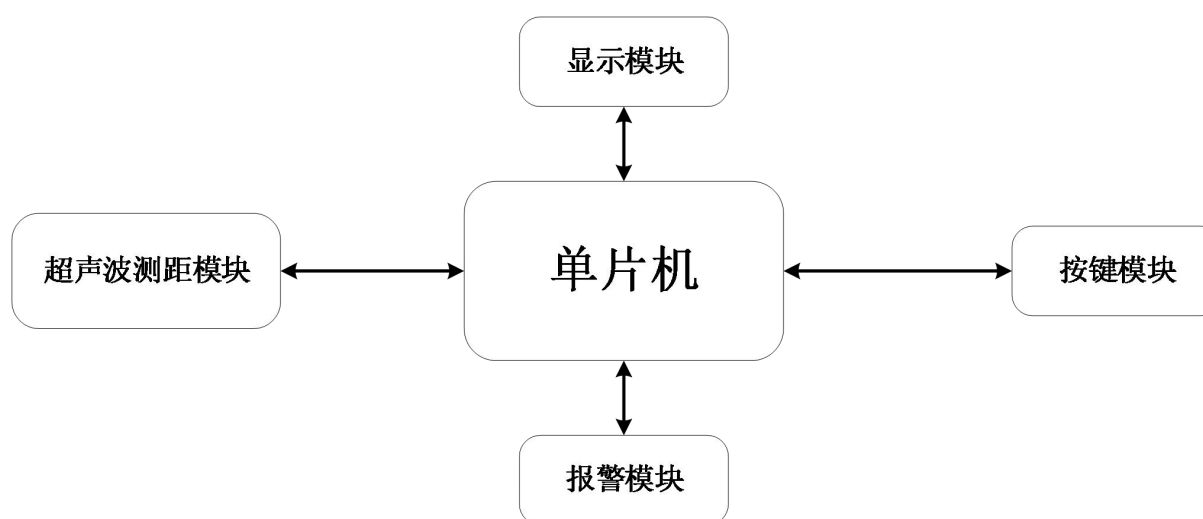
西安邮电大学本科毕业设计(论文)选题审批表

申 报 人	侯铁双	职 称	讲师	学 院	自动化学院		
题目名称	基于 51 单片机倒车防撞系统设计						
题目来源	科研				教学		其它 <input checked="" type="checkbox"/>
题目类型	硬件设计 <input checked="" type="checkbox"/>	软件设计		论文		艺术作品	
题目性质	应用研究 <input checked="" type="checkbox"/>		理论研究				
题目简述	本设计以单片机作为报警装置控制器，利用超声波对汽车进行测距，充分发挥单片机的数据处理和实时控制功能，实现汽车倒车防撞报警的作用。						
对学生知识与能力要求	1. 熟悉单片机技术 2. 熟悉汽车倒车防撞系统； 3. 应用单片机设计控制系统						
具体任务以及预期目标	1. 熟悉单片机技术； 2. 熟悉倒车防撞系统组成； 3. 应用单片机设计倒车防撞控制系统						
时间进度	2017-12-05 — 2018-01-06 查找资料，写开题报告 01-07 — 3-21 学习并掌握单片机技术 3-22 - 4-12 学习并掌握倒车防撞控制系统的组成 4-13 — 4-30 查阅资料设计控制系统 5-1 — 5-12 实现硬件仿真 5-11 — 5-25 完成系统调试，实现系统正常工作 5-26 — 6-10 毕业论文的准备与撰写 6-11— 6-20 毕业答辩						
系（教研室）主任 签字	年 月 日			主管院长 签字	年 月 日		

西安邮电大学本科毕业设计（论文）开题报告

学生姓名	马栋雯	学号	06141085	专业班级	自动 1403
指导教师	侯铁双	题目	基于 51 单片机倒车防撞系统设计		
<p>选题目的（为什么选该课题）</p> <p>随着我国经济飞速发展，人们的生活质量大幅度提高，私家车几乎进入了大多数家庭，成为了一种平民化交通工具。然而私家车平民化带来方便的同时也带来了危险。据最近一份调查报告显示，由于司机不当驾驶或者驾驶技术不合格造成的交通事故越来越多，汽车给人们生活带来方便的同时也存在着一定的安全问题。随着汽车产业的快速发展，汽车的智能化水平快速提升，越来越多的辅助驾驶功能被应用在汽车中，越来越多的人也依赖车上的辅助行车设备进行驾驶。调查显示，很多驾驶员呼吁车上能够配置更多的行车辅助设备，他们不仅要求辅助设备灵活好用，还要价格低廉，能够被大多数人所接受。</p> <p>因此本文以基于 51 单片机超声波测距的倒车防撞报警系统为课题，提出以超声波测距为基础，结合其他必要模块设计一款倒车防撞报警系统。超声波技术不仅可用于倒车系统，还可以应用到自动泊车、自动驾驶等行车辅助系统上。</p>					
<p>前期基础（已学课程、掌握的工具，资料积累、软硬件条件等）</p> <p>已学课程：《8051 单片机的 C 语言开发》《单片机原理及应用 A》《计算机控制技术》《高级语言程序设计》《传感器原理及应用》《模拟电子技术基础》；</p> <p>掌握工具：示波器、万用表、信号发生器、电烙铁、电脑；</p> <p>资料积累：《新概念 51 单片机 C 语言教程》（郭天祥 电子工业出版社） 《C 程序设计》（谭浩强 清华大学出版社） 《电路分析》（胡翔骏 高等教育出版社）</p> <p>软硬件条件：</p> <p>软件：Keil、protues、Altium Designer；</p> <p>硬件：51 单片机、超声波模块、显示模块、报警模块等。</p>					
<p>要研究和解决的问题（做什么）</p> <p>距离采集问题：使用超声波模块进行距离的测量。超声波测距原理是在超声波发射装置发出超声波，它的根据是接收器接到超声波时的时间差，与雷达测距原理相似。超声波发射器向某一方向发射超声波，在发射时刻的同时开始计时，超声波在空气中传播，途中碰到障碍物就立即返回来，超声波接收器收到反射波就立即停止计时（超声波在空气中的传播速度为 340m/s，根据计时器记录的时间 t，就可以计算出发射点距障碍物的距离(s)，即：$s=340t/2$）。</p> <p>报警问题：在设定好的距离报警，是此本课题的基本功能。这里使用蜂鸣器，完成报警功能。</p> <p>显示问题：采用数码管对当前的距离以及相关设置进行显示，为方使用书获知当前距离，同时进行设计报警距离。</p>					

工作思路和方案（怎么做）



超声波倒车雷达实现对距离的测量、数据处理、数据显示、报警，配合用户自定义报警距离等功能。该系统主要模块有：单片机最小系统、测距模块、显示模块、报警模块以及按键设置模块。

STC89C51 是采用 8051 核的 ISP（In System Programming）在系统可编程芯片，最高工作时钟频率为 80MHz，片内含 4K Bytes 的可反复擦写 1000 次的 Flash 只读程序存储器，器件兼容标准 MCS-51 指令系统及 80C51 引脚结构，芯片内集成了通用 8 位中央处理器和 ISP Flash 存储单元，具有在系统可编程（ISP）特性，配合 PC 端的控制程序即可将用户的程序代码下载到单片机内部。STC89C51 系列单片机是单时钟/机器周期(1T)的兼容 8051 内核单片机，是高速/ 低功耗的新一代 8051 单片机，全新的流水线/精简指令集结构, 内部集成 MAX810 专用复位电路

计划：

第 1 周一第 2 周：查找相关资料，选择各大模块所需芯片，进行电路基础方面的学习；

第 3 周一第 4 周：学习编译环境，如 Keil、Altium Designer 等；

第 5 周一第 7 周：各个模块硬件设计；

第 8 周一第 10 周：编写各个功能程序；

第 11 周一第 12 周：整合所有模块，进行各个模块的相互调试；

第 13 周一第 14 周：撰写毕业论文。

指导教师意见

签字

年 月 日

西安邮电大学毕业设计（论文）成绩评定表

学生姓名	马栎雯	性别	女	学号	06141085	专业 班级	自动 1403
课题名称	基于 51 单片机倒车防撞系统设计						
指导教师 意见	评分（百分制）： 指导教师(签字) 年 月 日						
评阅 教师 意见	评分（百分制）： 评阅教师(签字) 年 月 日						
验收 小组 意见	评分（百分制）： 评阅教师(签字)： 年 月 日						
答辩 小组 意见	评分（百分制）： 答辩小组组长(签字) 年 月 日						
评分比例	指导教师评分 （20%） 评阅教师评分 （30%） 验收小组评分(30%) 答辩小组评分 （20%）						
学生总评 成绩	百分制成绩				等级制成绩		
答辩委员 会意见	毕业论文(设计)最终成绩(等级)： 学院答辩委员会主任(签字、学院盖章)： 年 月 日						

目 录

摘 要.....	I
ABSTRACT.....	II
第一章 引言.....	1
1.1 课题背景.....	1
1.2 课题任务.....	1
1.3 论文结构.....	2
第二章 背景知识.....	3
2.1 单片机知识概述.....	3
2.1.1 STC89C51 外部结构及特性.....	3
2.2.2 STC89C51 内部组成.....	4
2.2 超声波基础理论.....	5
2.2.1 超声波简介.....	5
2.2.2 超声波的特质.....	5
2.2.3 超声波的应用.....	6
第三章 系统硬件设计.....	7
3.1 系统整体硬件设计.....	7
3.2 单片机最小系统设计.....	7
3.2.1 时钟电路设计.....	8
3.2.2 复位电路设计.....	9
3.2.3 下载电路设计.....	9
3.2.4 电源电路设计.....	10
3.3 超声波测距模块硬件设计.....	10
3.4 显示模块硬件设计.....	11
3.5 报警模块硬件设计.....	12
3.6 按键模块硬件设计.....	13
3.7 硬件设计总结.....	13
第四章 软件设计.....	14
4.1 系统整体软件设计思路介绍.....	14

4.2 超声波测距程序设计.....	15
4.2.1 超声波测距原理简介.....	15
4.2.2 超声波模块时序.....	15
4.2.3 超声波测距代码实现.....	16
4.3 显示模块程序设计.....	16
4.3.1 显示模块程序设计思想.....	16
4.3.2 显示模块代码实现.....	18
4.4 报警模块程序设计.....	19
4.4.1 报警模块程序设计思想.....	19
4.4.2 报警模块代码实现.....	20
4.5 按键模块程序设计.....	20
4.5.1 按键模块程序设计思想.....	20
4.5.2 按键模块代码实现.....	22
4.6 软件设计小结.....	23
第五章 调试与总结.....	24
5.1 软件编译环境.....	24
5.1.1 硬件设计环境.....	24
5.1.2 软件设计环境.....	25
5.1.3 仿真环境.....	25
5.2 超声波模块调试与误差分析.....	26
5.3 仿真调试.....	27
5.4 硬件实物及其工作状态.....	28
5.4.1 硬件实物.....	28
5.4.2 硬件工作状态.....	28
5.5 调试小结.....	29
结束语.....	30
致 谢.....	31
参考文献.....	32
附录.....	33
附录 A 系统原理图.....	33
附录 B 部分程序代码.....	34

摘 要

本毕业设计采用单片机（STC89C51）作为其核心控制器，设计一种超声波无线测距仪倒车雷达防撞报警系统。本系统使用所学习的嵌入式技术、模拟电子相关技术、C 语言编程技术、电路仿真等技术，实现距离测量，声光报警（高于预设的门限），按键设置等功能。

本毕业设计使用模块化设计的设计思想。主要包含单片机（STC89C51）最小系统、信息采集模块（超声波测量距离）、显示模块（4 位数字显示）、报警模块（声光报警）、按键设置调节模块等。信息采集模块主要实现的功能是使用超声波对距离进行测量；显示模块用数码管进行相关数据的显示；报警模块实现功能为当距离超过设计的阈值时对操作人员进行提示；按键模块实现相关参数及功能的用户设置；按键模块与显示模块相配合实现人机交互。

本毕业设计实现了倒车距离的测量、显示、报警，实现的倒车的辅助，很大程度上提高了泊车和倒车的安全性和可操作性，为驾驶员提供一定的帮助。本系统还可以移植为其他的驾驶辅助系统，具有很强的使用价值。

关键字：单片机；超声波；数码管；

ABSTRACT

The graduation design uses the STC89C51 microcontroller as the core controller to design an ultrasonic wireless rangefinder reversing radar anti-collision alarm system. This system uses the embedded technology, analog electronic related technology, C language programming technology, and circuit simulation technology to achieve ultrasonic distance measurement, sound and light alarms, and key setting adjustments.

The graduation design system adopts a modular design method and mainly includes a single-chip microcontroller minimum system module, an information acquisition module (ultrasonics), a display module, an audible and visual alarm module, and a button module. The acquisition module uses ultrasonic waves to measure the distance; the display module uses the digital tube to display the relevant data; the audible and visual alarm module is used to alert the user; the key module implements the user's settings of related parameters and functions; the key module and the display module cooperate Human-computer interaction. The system's hardware and software adopt a modular design concept. First, the hardware and software debugging of each module, simulation through the software, and then the actual system to debug the entire system.

The graduation design realizes the measurement, display and alarm of the reversing distance, and the assistance of reversing the vehicle, greatly improves the safety of parking and reversing, and provides certain help for the driver. The system can also be transplanted to other driving assistance systems and has a strong use value.

Keywords: Microcontroller; Ultrasonic; Digital Tube

第一章 引言

1.1 课题背景

伴随着全球经济的快速发展，人们日常的生活水平不断提升，汽车已经变成人们日常的必需品之一。然而，由于不断增加的汽车数量，也带来了许多的问题。比如汽车数量的增加所引发的交通事故也越来越多，给人们的生活带了一定的危害，汽车辅助驾驶系统应运而生。

在当今汽车中，应用的电子信息技术越来越多，汽车的智能程度越来越高。技术的需要不断个性，由汽车工业技术和电子信息技术相结合而产生了汽车电子技术。汽车行业技术与电子信息技术不断相互促进，共同发展，共同发展。汽车的智能化水平越来越高，越来越多的驾驶辅助系统应运而生，给人们的驾驶带来很好的帮助。应用最多的就是倒车辅助装置。汽车发生碰撞的原因有很多种，大多都是由于驾驶员对于距离的判断不准确而导致错误判断刹车时间，为了避免这样的情况，就需要由辅助装置实时的提醒当前的距离，留给驾驶员更多的反应时间。

汽车要避免由于距离过近而发生的碰撞，就需要实时的检测当前的距离，并反馈给驾驶员，让其做出相应的判断。即使在突发的危险情况下，它也可以执行自动制动的操作以避免交通事故。

1.2 课题任务

本毕业设计课题要求设计一种于基于超声波无线测距的倒车雷达防撞报警系统，本毕业设计系统的整体硬件框架如图 1.1 所示。该系统利用嵌入式技术、传感器技术，超声波避障相关技术等，来实现此系统的测距、报警、显示等行为。本系统的控制中心为单片机，距离测量则用的是超声波模块，4 位一体数码管来实现相关信息的显示以及于按键形成人机交互，用蜂鸣器来实现报警功能，电源模块提供稳定的电源等模块构成完整系统。

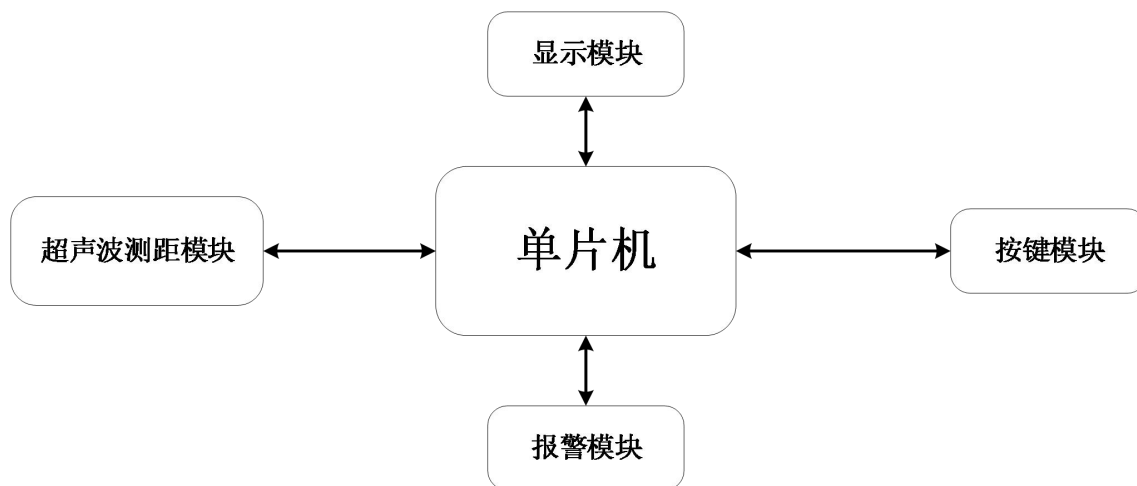


图 1.1 系统框图

1.3 论文结构

本毕业设计论文将按章节来介绍整体系统的设计。

第一章，引言：介绍课题背景、课题任务等。

第二章，背景知识：介绍本设计中所用的主要器件模块的相关资料，为之后的设计工作打下基础；

第三章，系统硬件设计：介绍系统硬件所用到的相关知识和各个模块以及整体电路的设计，整体采用总分总的结构来介绍。

第四章，软件设计：讲解系统各个模块的软件设计思想及其实现整体，采用总分总的结构来介绍。

第五章，调试与误差分析：介绍软硬件的设计环境及其各个功能的调试方法及误差分析与解决；

第六章，思想总结：总结在毕业设计过程中的所感所想以及对未来无限的展望。

第二章 背景知识

2.1 单片机知识概述

基于 51 单片机超声波无线测距仪倒车雷达防撞报警系统毕业设计使用的芯片是 STC89C51 单片机。STC89C51 单片机是一种相对较为常见的微型控制器，它是采用 8051 核的单片机，80MHz 是它可达到的最高工作时钟频率。STC89C51 微控制器包含 4K 字节的闪存，可重写 1000 次，只读程序存储器地址 0000H-0FFFH，片外最多可扩展至 64KBROM，地址为 1000H-FFFFH。

2.1.1 STC89C51 外部结构及特性

STC89C51 的外形封装有：PDIP-28，SOP-28，PDIP-20，SOP-20，PLCC-32，TSSOP-20 等多种形式。其中最多见的为直插的 40Pin 封装，实物如图 2.1 所示



图 2.1 51 单片机实物图

引脚排列和外部总线结构如图 2.2 所示

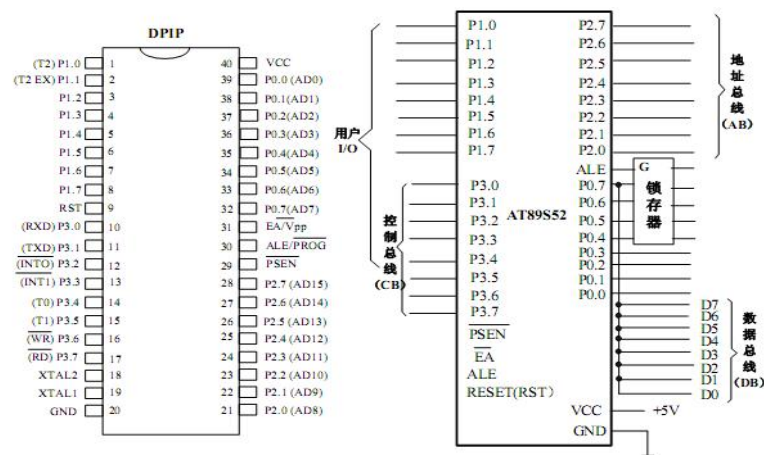


图 2.2 微控制器引脚排列和外部总线结构

2.2.2 STC89C51 内部组成

STC89C51 微控制器集成了多种 I/O 设计，包括 CPU、存储器（RAM、ROM）、定时器/计数器、看门狗等。具备了一台电子计算机所需要的所有的基本功能单元。具体内部包含如图 2.3 所示：

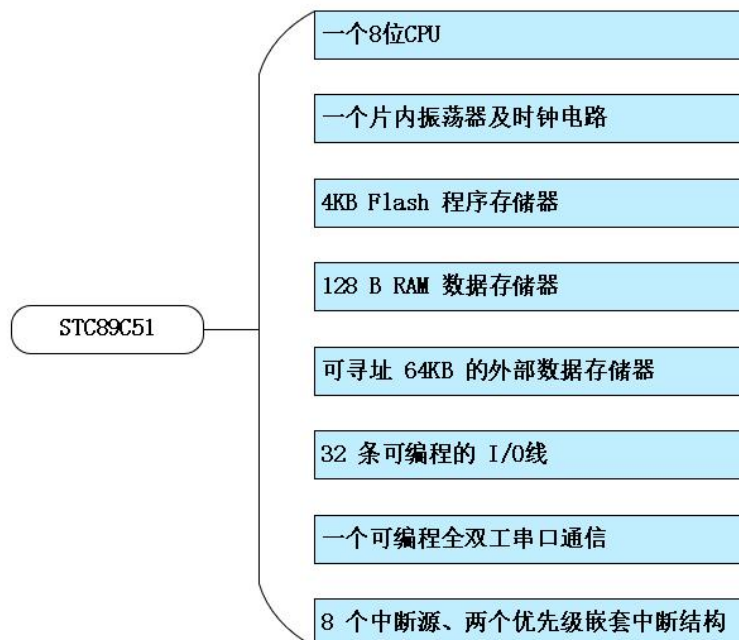


图 2.3 单片机内部功能单元

STC89C51 微控制器的内部结构框图如图 2.4 所示。

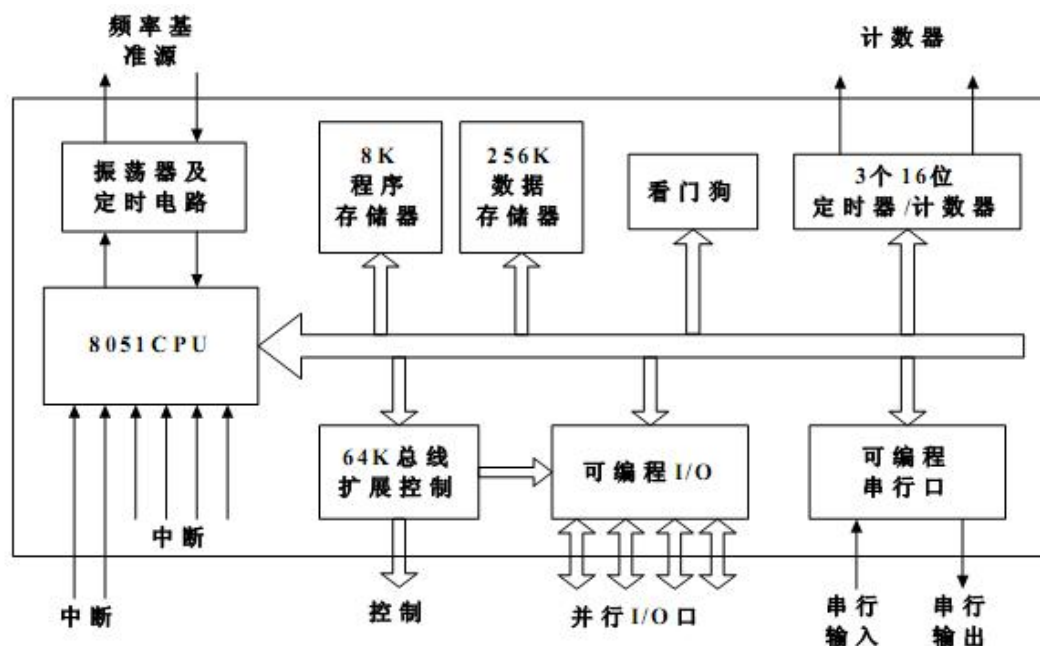


图 2.4 STC89C51 单片机内部框图

2.2 超声波基础理论

2.2.1 超声波简介

声波是由发声体的振动而产生的。声波可以分为很多种。按照不同的振动频率可分为不同的声波。频率低于二十赫兹(20Hz)称为次声波或低声波；频率范围在二十赫兹到二十千赫(20Hz~20kHz)之间的声波称为可闻声波；频率范围在二十千赫到一吉赫(20kHz~1GHz)的声波称为超声波；频率大于一吉赫(1GHz)的声波称为特超声或微波超声。在这些声波中，超声波是运用最为普及的一种。

2.2.2 超声波的特质

- 1) 超声波与其他的机械波一样，它的传播途径具有较强的定向性，并且它的能量更容易集中；
- 2) 超声波在传递时可将能量较为集中的传递出去；
- 3) 超声波在不同传播媒质（固体、液体、气体等）中都可以进行传播且传播距离较长；
- 4) 超声波与其他的声波、机械波相比较其可以传递的能量很强；
- 5) 与普通声波一样，超声波也具有反射、干涉、叠加和共振等的物理特性；

- 6) 超声波拥有超声效应。

2.2.3 超声波的应用

超声波已被广泛应用，主要应用如下：

- 1) 超声波测量
- 2) 超声波清洗
- 3) 超声波除鳞、除油
- 4) 超声波医学检查
- 5) 工业自动控制
- 6) 超声波化学合成
- 7) 超声波制药

第三章 系统硬件设计

3.1 系统整体硬件设计

本毕业设计的 System block 如图 3.1 所示。本毕业设计系统硬件设计包含：STC89C51 单片机最小系统硬件电路设计、超声波测距模块硬件电路设计、报警模块硬件电路设计、显示模块硬件电路设计和按键模块硬件电路设计。

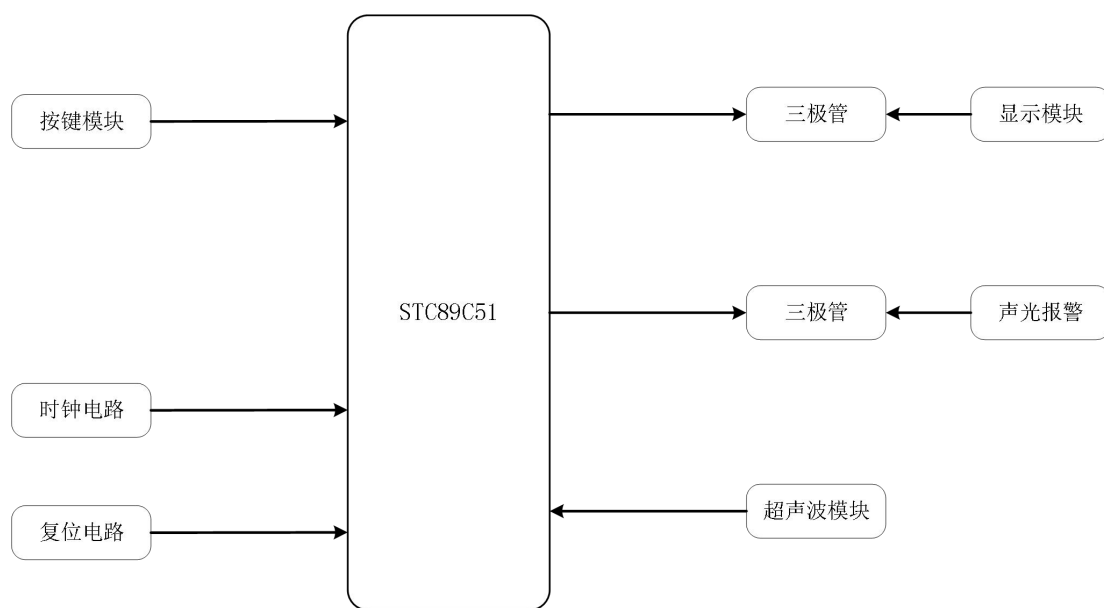


图 3.1 系统框图

3.2 单片机最小系统设计

单片机最小系统的稳定与否将直接影响整个系统能否正常运行。复位电路、时钟电路、下载电路等基础电路构成了单片机最小系统电路，如图 3.2 所示。

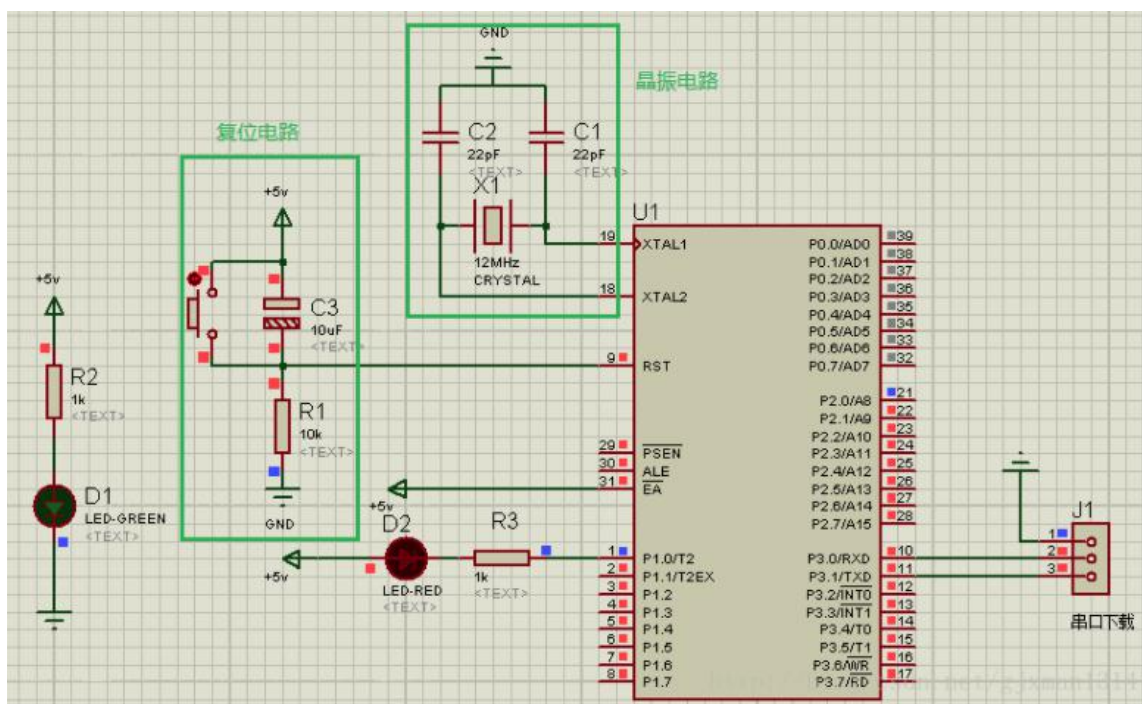


图 3.2 最小系统电路图

3.2.1 时钟电路设计

需要外部时钟输入的 IC（包括微控制器）内部由高增益的反相放大器和它的反馈电阻组成一个振荡器。在一般情况下，由单片机的时钟输入引脚，XTAL1 和 XTAL2 之间连接两个补偿电容和一个石英晶体，由此就可构成一个自激振荡器。其电路图如图 3.3 所示。可根据器件手册和实际情况选择相应频率的石英晶体（典型值 11.0592MHz），再结合石英晶体的相关参数计算补偿电容的值。

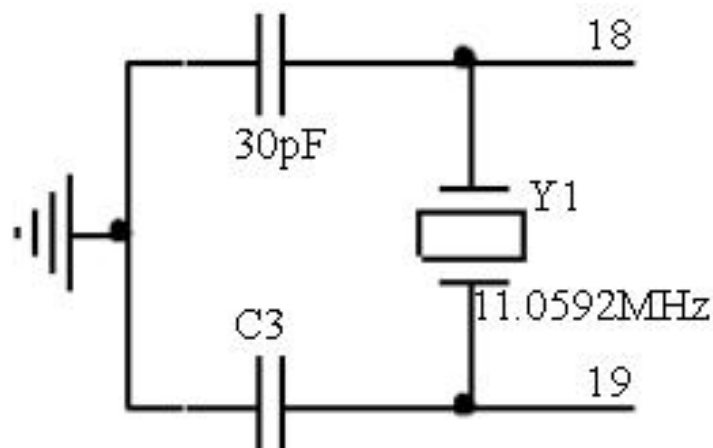


图 3.3 单片机时钟电路图

3.2.2 复位电路设计

IC 的常用的复位方式有很多种。如上电自动复位、按键手动复位、使用专门的 IC 实现过压或欠压复位等多种方式。其中，上电自动复位要求系统开机之后单片机自动实现复位操作；而手动按键复位需要通过外部按键在单片机工作的过程中进行复位操作。如图 3.4 所示，上电自动复位时通过系统上电之后对电容充电，因为电容两端的电压不能发生突变来实现，所以需要手动按键复位通过按键来接通电阻与电源，在需要复位时通过按下按键进行复位操作。

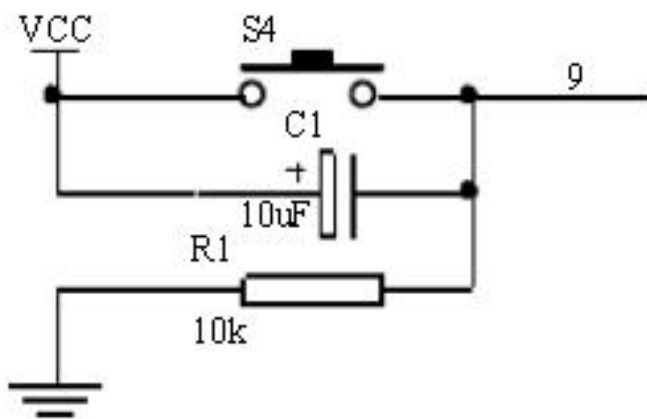


图 3.4 单片机复位电路图

3.2.3 下载电路设计

使用 UART 进行程序下载。STC89C51 只有一组 UART (P3.0RX、P3.1TX) 外接到的接口上。为防止 UART 误操作或被干扰，可以对其外部加上拉电阻；在 Layout 时适当拉开两条线的间距（最少保证 3W）。如图 3.5 所示。

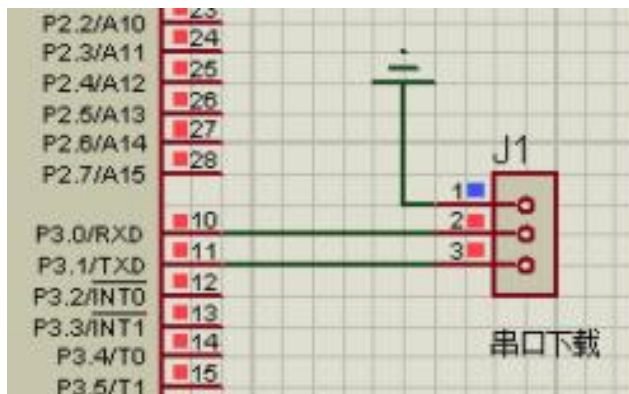


图 3.5 下载接口电路图

3.2.4 电源电路设计

STC89C51 微控制器的工作电压为 5.0V。本毕业设计直接使用直流 5V 对其供电（USB 接口）。如系统使用其他电源供电，则需加相应的电源电路，根据外围电路的实际需求选用满足系统功耗的电源 IC (DC-DC、LDO)。

3.3 超声波测距模块硬件设计

本毕业设计使用 HC-SR04 超声波模块进行距离测量。超声波模块实物如图 3.6 所示

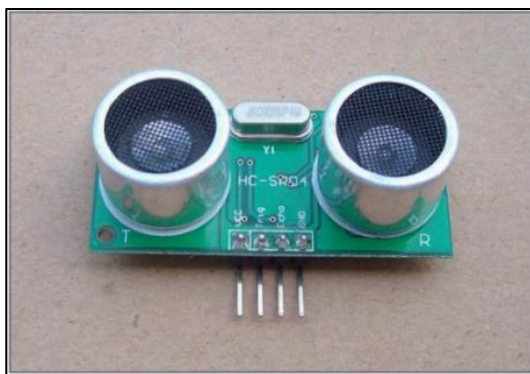


图 3.6 超声波模块实物

其基本工作原理：

- 1) 通过 51 单片机的 I/O 口给超声波模块的 TRIG 输出一个至少 10 μ s 的高电平信号，触发超声波模块开始工作；
- 2) 超声波模块出发之后可以自动发送八个频率位 10KHz 方波，然后开始检测是否有信号返回；
- 3) 当超声波模块检测到有信号返回时，超声波模块中的 ECHO 会输出一个高电平，单片机通过 I/O 去检测此高电平信号，此信号的持续时间就是超声波传播的时间。

表 3.1 超声波模块技术参数

工作电压	DC 5V
静态电流	小于 2mA
电平输出 MAX	5V
电平输出 MIN	0V
感应角度	不大于 15 度
探测距离	2cm-500cm
探测精度	0.3cm

超声波模块对外接口如图 3.7 所示。其接口为间距是 2.54mm 的 4Pin 排针，接口定义如表 3.2 所示，为其预留间距为 2.54mm 的 4Pin 排母。需要注意超声波模块与其他器件的干涉问题。

表 3.2 超声波模接口定义

接口名称	说明
VCC	电源输入（5V）
TRIG	触发信号输入
ECHO	回响信号输出
GND	地

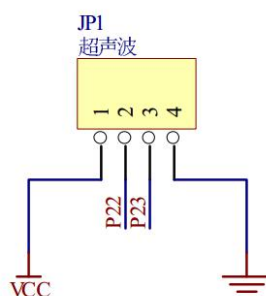


图 3.7 超声波模块接口电路

3.4 显示模块硬件设计

本毕业设计的显示模块使用一个 4 位一体的 7 段共阳极数码管，显示模块的主要功能是显示当前的实时距离以及配合按键实现相关设置。4 位数码管是由四个单独的数码管封装到一起，每一位数码管的 a, b, c, d, e, f, g 和 dp 端在其内部连接并连接到微控制器的 P0 口。四位一体数码管有四个位选端 S1, S2, S3, S4 分别代表每一位，连接到单片机的 P2 口。

在软件设计中采用动态扫描的方法，数码管的 a, b, c, d, e, f, g 和 dp 同名端连在一起，但用独立的单片机 I/O 口对每一个数码管的公共极 COM 进行控制。当单片机输出相关代码时，所有的数码管都可以接收到，且所有的数码管所受到的相同；通过单片机控制其公共端，决定哪一个数码管被点亮。单片机重复上述工作，不断的点亮不同的数码管，整个过程时一个动态扫描的过程。

在整个动态显示的过程中，由于扫描点亮时间极短，人类有暂时滞留的视觉现象，人眼看起来就很稳定。显示模块的原理图如图 3.8 所示。

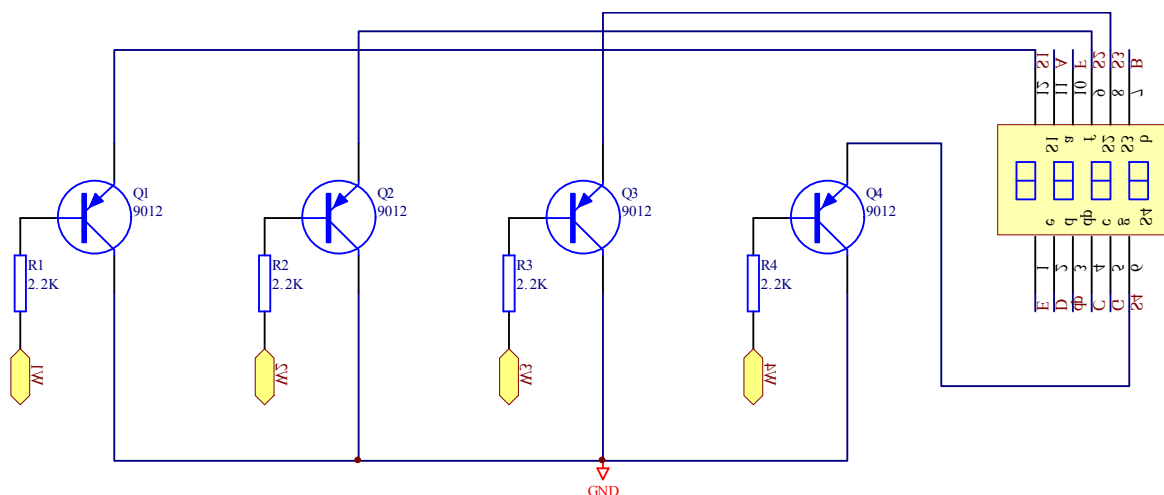


图 3.8 显示模块原理图

3.5 报警模块硬件设计

本毕业设计使用蜂鸣器配合一个 LED 二极管进行声光报警。蜂鸣器是一种应用较为广泛的电子发声器件；二极管为红色发光二极管。

单片机 I / O 口（STC89C51）提供的输出电流不足以蜂鸣器进行正常的工作，需要对单片机输出的信号进行放大后输出给蜂鸣器。电流放大有多种方式，可以使用运放芯片或者分立器件，在毕业设计中，使用三极管组成的放大电路对单片机的输出电流进行放大，从而使蜂鸣器正常工作。其放大电路图如图 3.9 所示

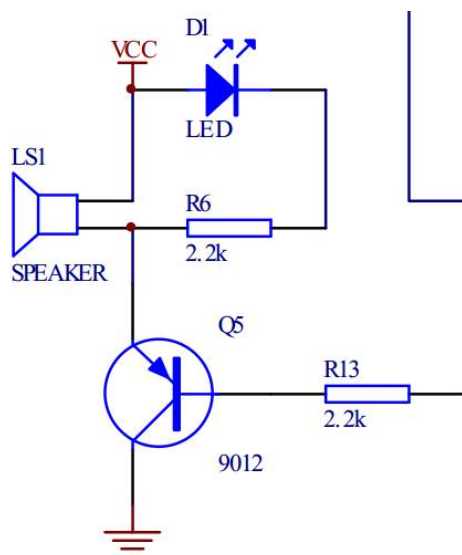


图 3.9 报警模块电路图

3.6 按键模块硬件设计

单片机的键盘可分为两种；独立键盘和矩阵键盘。独立键盘是指 MCU 的每个 I / O 只接一个按钮，而按钮的另一端连接到 VCC 或 GND；独立键盘的优点是在程序处理方面较为简单容易，其缺点也较为明显，如果按键较多需要很多的 I/O 资源。矩阵键盘以矩阵的形式将按键连接，程序处理较为复杂，但占用 I/O 资源少。经综合考虑本毕业设计使用三个独立按键组成的按键模块。按键模块的电路图如图 3.10 所示

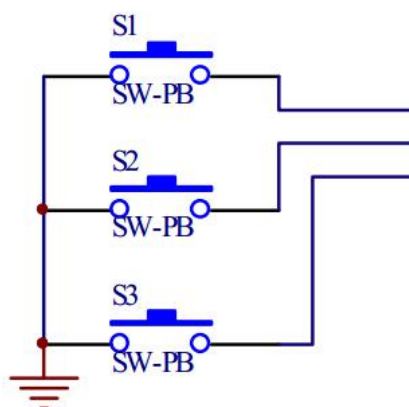


图 3.10 按键模块电路图

3.7 硬件设计总结

该系统所涉及的硬件有 STC89C51、超声波模块、数码管、蜂鸣器及相关驱动电路，通过阅读各个模块 IC 的 DataSheet 了解其性能、各类参数及使用注意事项等。作为系统最重要的部分，微控制器不仅需要了解 IC 本身的特性，还需要了解其最小系统的设计。晶体选择，补偿电容选择，复位电路选择等；显示模块和报警模块不仅需要名称数码管、蜂鸣器自身的参数，还需要知道其驱动电路的设计；按键电路也需要考虑硬件去抖等等。只有对每个器件完全了解才能设计出稳定可靠的系统，只有硬件系统稳定可靠，才能为软件打好基础。

第四章 软件设计

4.1 系统整体软件设计思路介绍

本毕业设计的软件设计编程思想为系统模块化。先将整体分成模块，在根据需求对各个模块进行程序编写，这样可以保证各模块能够独立的进行正常的工作，再对主程序的框架进行编写，为每个模块程序调用级联调试，并通过对所有模块进行集成来完成整个软件设计。主程序的流程图如图 4.1 所示。

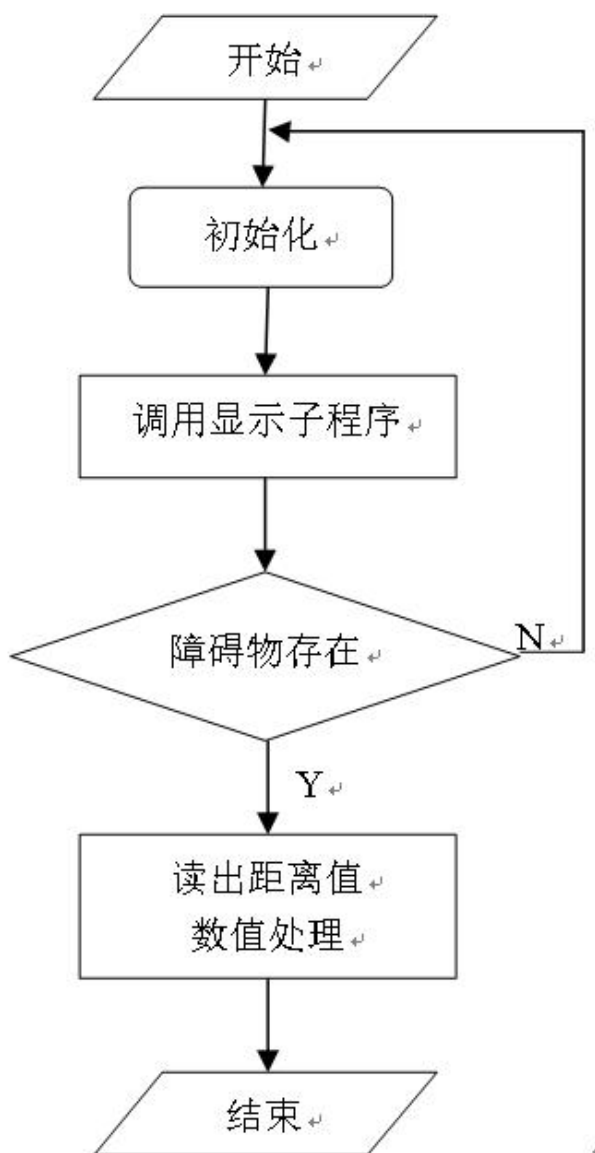


图 4.1 主程序流程图

4.2 超声波测距程序设计

4.2.1 超声波测距原理简介

超声波测距原理是通过单片机计时器记录超声波模块在发出信号后直至模块输出端输出高电平的时间，再使用超声波传播速度（超声波传播速度在环境空气中约为 340m/s ），根据测得时间就可以算出所要测量距离。超声波测距原理如图 4.2 所示。测量距离 S 等于传播时间与 超声波的传播速度之积，要注意式中所用的时间为测得时间的一半，由于超声波需要进行反射，所以传播时间实际为所得总时间的一半。

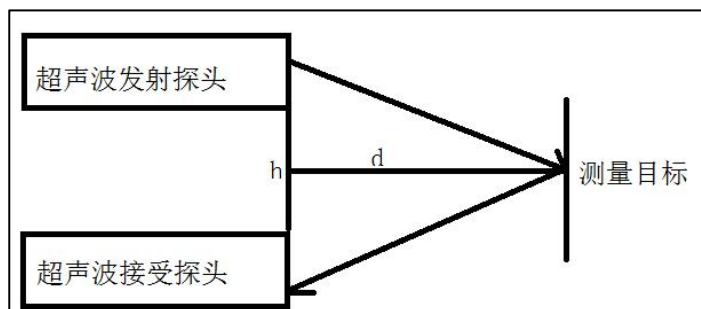


图 4.2 超声波测距原理

单片机连接超声波模块控制端的 I/O 口给超声波模块的控制端输出一个启动信号（ $10\mu\text{s}$ 以上的高电平），然后通过连接超声波模块输出端的 I/O 去检测其输出信号（高电平输出）。当 MCU 检测到超声波模块输出的混响信号的高电平时，定时器开启以进行定时。当超声波模块混响信号的输出信号变低时，停止计时并读取计时器值。记录的时间是超声波传播的时间，从中可以计算出测量的距离。

4.2.2 超声波模块时序

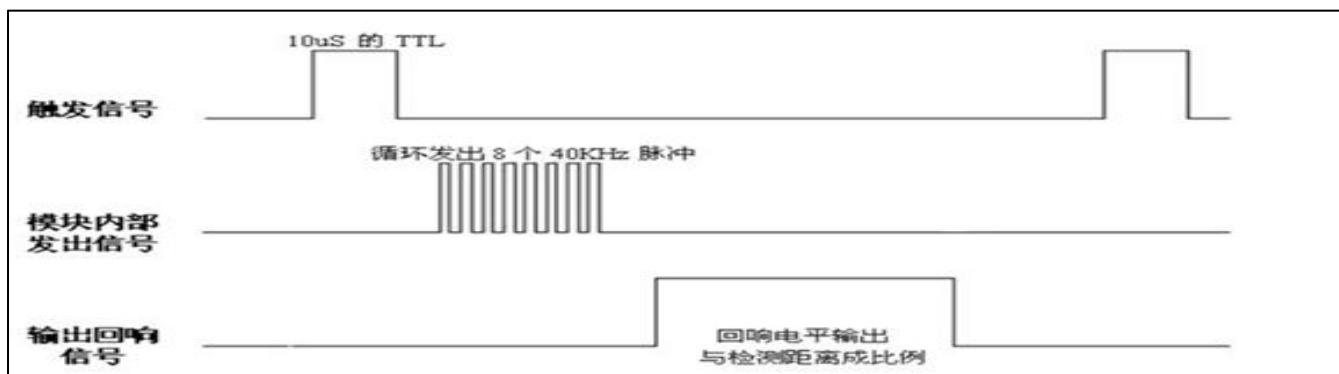


图 4.3 超声波时序图

超声波模块的具体工作时序图如图 4.3 所示。由其工作时序图可以清楚的看出其工作原理。

4.2.3 超声波测距代码实现

```
void Conut(void)
{
    time=TH0*256+TLO;    //读出T0的计时数值
    TH0=0;
    TLO=0;                //清空计时器
    S=(time*1.7)/100;      //算出来是CM
    //声音的速度是340m/s, 时间的单位是us, 计算到秒需要将时间数据/1000000,
    //长度=速度*时间, 340*time/1000000, 长度数据单位是m转换成cm需要乘以100得到340*time/10000,
    //小数点都向左移两位得到3.4*time/100, 因为超声波是往返了, 所以再除以2,得到距离数据(time*1.7)/100
    if(Mode==0)            //非设置状态时
    {
        if((S>=700)||flag==1) //超出测量范围显示“-”
        {
            Feng=0;          //蜂鸣器报警
            flag=0;
            disbuff[0]=10;    //“-”
            disbuff[1]=10;    //“-”
            disbuff[2]=10;    //“-”
        }
        else
        {
            //距离小于报警距
            if(S<=BJS)
            {
                Feng=0; //报警
            }
            else //大于
            {
                Feng=1;    //关闭报警
            }
            disbuff[0]=S%1000/100;    //将距离数据拆成单个位赋值
            disbuff[1]=S%1000%100/10;
            disbuff[2]=S%1000%10 %10;
        }
    }
    else
    {
        Feng=1;
        disbuff_BJ[0]=BJS%1000/100;
        disbuff_BJ[1]=BJS%1000%100/10;
        disbuff_BJ[2]=BJS%1000%10 %10;
    }
}
```

4.3 显示模块程序设计

4.3.1 显示模块程序设计思想

显示模块程序的主要功能就是将实时测量的数据用数码管显示出来, 以及显示设置时的相关信息。数码管的工作原理就是给数码管相应的段加上电压后, 这一段就会发光, 根

据自身需求给不同的段加电压就会有不同的组合就可以呈现出不同的效果。真值表如表 4.1 所示。

表 4.1 数码管真值表

显示	段号								十六进制代码	
	dp	g	f	e	d	c	b	a	共阴极	共阳极
0	0	0	1	1	1	1	1	1	3FH	C0H
1	0	0	0	0	0	1	1	0	06H	F9H
2	0	1	0	1	1	0	1	1	5BH	A4H
3	0	1	0	0	1	1	1	1	4FH	B0H
4	0	1	1	0	0	1	1	0	66H	99H
5	0	1	1	0	1	1	0	1	6DH	92H
6	0	1	1	1	1	1	0	1	7DH	82H
7	0	0	0	0	0	1	1	1	07H	F8H
8	0	1	1	1	1	1	1	1	7FH	80H
9	0	1	1	0	1	1	1	1	6FH	90H
A	0	1	1	1	0	1	1	1	77H	88H
b	0	1	1	1	1	1	0	0	7CH	83H
C	0	0	1	1	1	0	0	1	39H	C6H
d	0	1	0	1	1	1	1	0	5EH	A1H
E	0	1	1	1	1	0	0	1	79H	86H
F	0	1	1	1	0	0	0	1	71H	8EH
H	0	1	1	1	0	1	1	0	76H	89H
P	0	1	1	1	0	0	1	1	F3H	8CH

4.3.2 显示模块代码实现

```

void Display(void)
{
    //正常显示
    if(Mode==0)          //正常测距模式
    {
        num++;           //计数切换不同显示数码管
        if(num==1)       //第一个数码管
        {
            W3=1;         //关闭第二位
            W0=1;         //关闭第二位
            P0=~discode[disbuff[0]]; //显示整数位，因为段码数组为共阴数码管码表，所以在使用时要取反
            DIAN=0;        //点亮小数点
            W1=0;         //打开位
        }
        else if(num==2)   //第二位
        {
            W1=1;         //关闭第一位
            P0=~discode[disbuff[1]]; //显示小数点后一位
            W2=0;         //打开第二位
        }
        else if(num>=3)   //第三个位置
        {
            W2=1;         //关闭第二位
            P0=~discode[disbuff[2]]; //显示小数点后二位
            W3=0;         //打开第三位
            num=0;         //清零
        }
    }
    ,
    //报警显示
    else
    {
        num++;
        if(num==1)
        {
            W3=1;
            P0=~0xCE;     //11001110 显示字母H
            W0=0;
        }
        else if(num==2)
        {
            W0=1;
            P0=~discode[disbuff_BJ[0]]; //显示报警值
            DIAN=0;
            W1=0;
        }
        else if(num==3)
        {
            W1=1;
            P0=~discode[disbuff_BJ[1]];
            W2=0;
        }
        else if(num>=4)
        {
            W2=1;
            P0=~discode[disbuff_BJ[2]];
            W3=0;
            num=0;
        }
    }
}

```

4.4 报警模块程序设计

4.4.1 报警模块程序设计思想

报警模块所要实现的功能就是当实时测量的距离小于所设定的报警阈值时，单片机输出的信号经过外部驱动使得蜂鸣器发声以及点亮二极管，从而达到声光报警的目的。程序流程图如图 4.4 所示

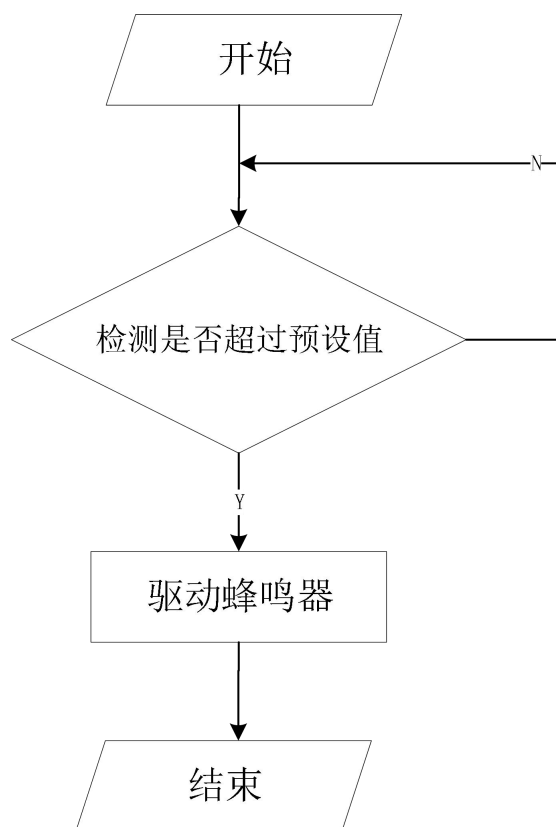


图 4.4 报警模块程序流程图

4.4.2 报警模块代码实现

```

if(Mode==0)           //非设置状态时
{
    if((S>=700)||flag==1) //超出测量范围显示“-”
    {
        Feng=0;          //蜂鸣器报警
        flag=0;
        disbuff[0]=10;    //“-”
        disbuff[1]=10;    //“-”
        disbuff[2]=10;    //“-”
    }
    else
    {
        //距离小于报警距
        if(S<=BJ8)
        {
            Feng=0; //报警
        }
        else //大于
        {
            Feng=1;  //关闭报警
        }
    }
}

```

4.5 按键模块程序设计

4.5.1 按键模块程序设计思想

按键模块主要实现功能是使得报警距离可以进行设计调节，从而使得设备可以适用于不同的情况，程序流程图如图 4.5 所示。

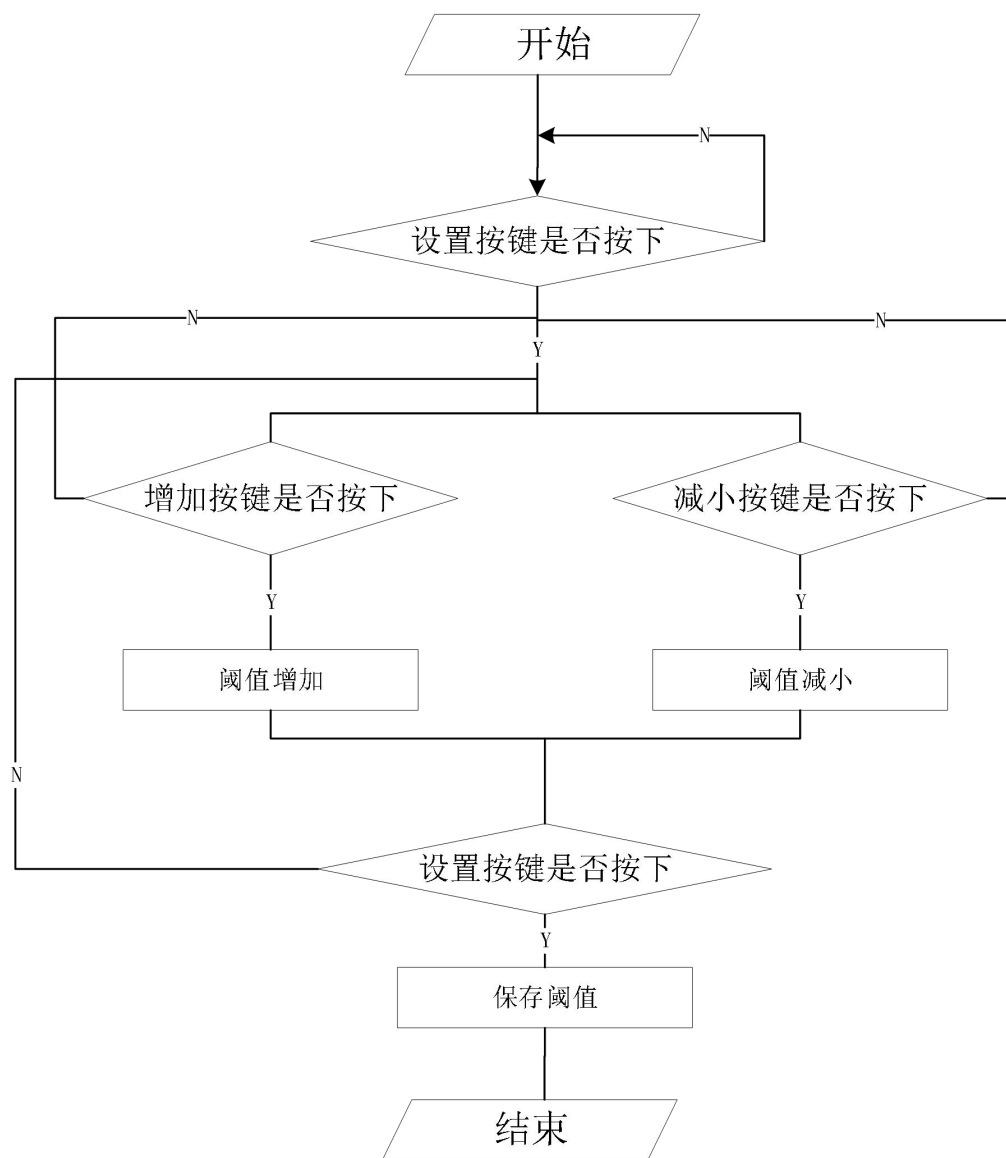


图 4.5 按键模块程序流程图

按钮模块需要去抖动以防止误操作。抖动的关键可以分为软件去抖和硬件去抖。在硬件电路上增加电容为硬件去抖；软件设置相应的延时为软件去抖。本毕业设计使用去抖方式作为软件去抖。

4.5.2 按键模块代码实现

```

//按键扫描
void Key_()
{
    //+
    if(S1==0&&Mode==1)    //设置状态时,按下加按键
    {
        delay();    //延时去抖
        delay();    //延时去抖
        while(S1==0)    //按键松开
        {
            P1=P1|0x0f;    //不显示
        }
        BJ3++;    //报警值加
        if(BJ3>=151) //最大150,加到大干等于151就会变成0
        {
            BJ3=0;    //清零
        }
        write_eeprom();    //保存数据
    }
    //-
    if(S2==0&&Mode==1)
    {
        delay();
        delay();    //延时去抖
        while(S2==0)
        {
            P1=P1|0x0f;
        }
        BJ3--;    //报警值减
        if(BJ3<=1)    //最小1
        {
            BJ3=150;
        }
        write_eeprom();    //保存数据
    }
    //功能
    if(S3==0)    //设置键
    {
        delay();
        delay();    //延时去抖
        while(S3==0)
        {
            P1=P1|0x0f;
        }
        Mode++;    //模式加
        num=0;
        if(Mode>=2)    //加到2时清零
        {
            Mode=0;    //退出设置状态
        }
    }
}
}

```


4.6 软件设计小结

使用 C 语言在 KIEL 环境中调试和调试微控制器。首先，对每个模块进行程序设计，根据流程图完成模块程序，以实现相应的功能；通过每个模块程序的主要功能调用，调试，编译，最终完成整体软件调试测试。在级联调试中，必须特别注意每个模块互相调用，I/O 端口初始化，寄存器配置和变量定义之间的关系。确保级联调试正常，然后执行最终的软件和硬件调试。

第五章 调试与总结

5.1 软件编译环境

5.1.1 硬件设计环境

本毕业设计硬件电路设计使用 Altium Designer 软件。该软件可进行较多功能的设计（原理图库设计、封装库设计、电路原理图设计、PCB 设计），软件使用较为简单，但是功能强大；使用的软件版本是 Altium Designer（13.0）版本；设计界面如图 5.1-5.5 所示。

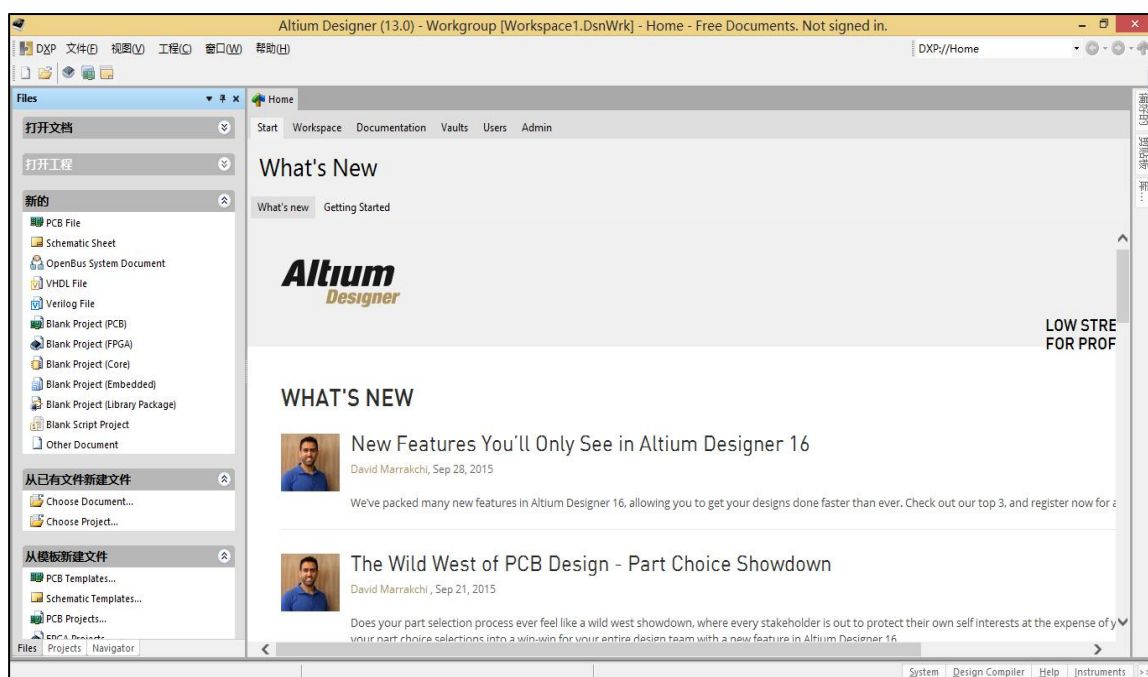


图 5.1 Altium Designer 硬件设计环境界面

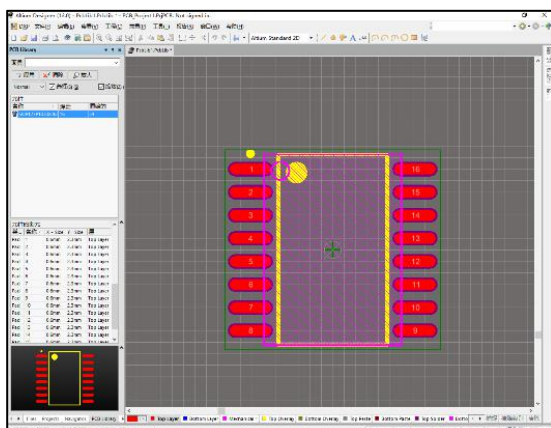


图 5.2 封装库设计界面

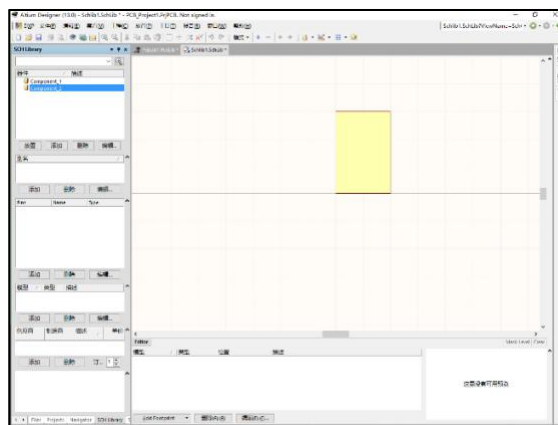


图 5.3 原理图库设计界面

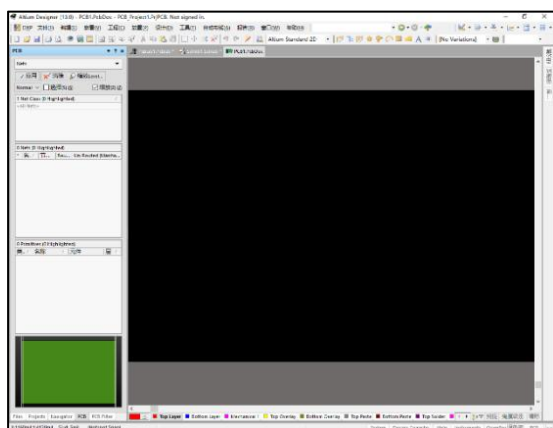


图 5.4 PCB 设计界面

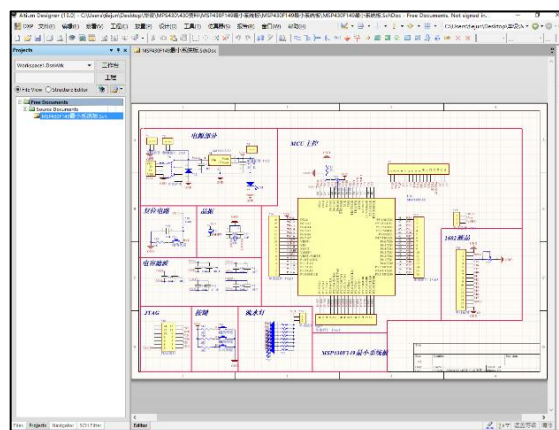


图 5.5 原理图设计界面

5.1.2 软件设计环境

本毕业设计的软件编写及编译使用的是 Keil 软件。使用此软件进行代码编写、调试以及下载等操作简单且功能强大实用，本毕业设计所使用的是 Keil uVision5 版本。软件界面如图 5.6 所示。

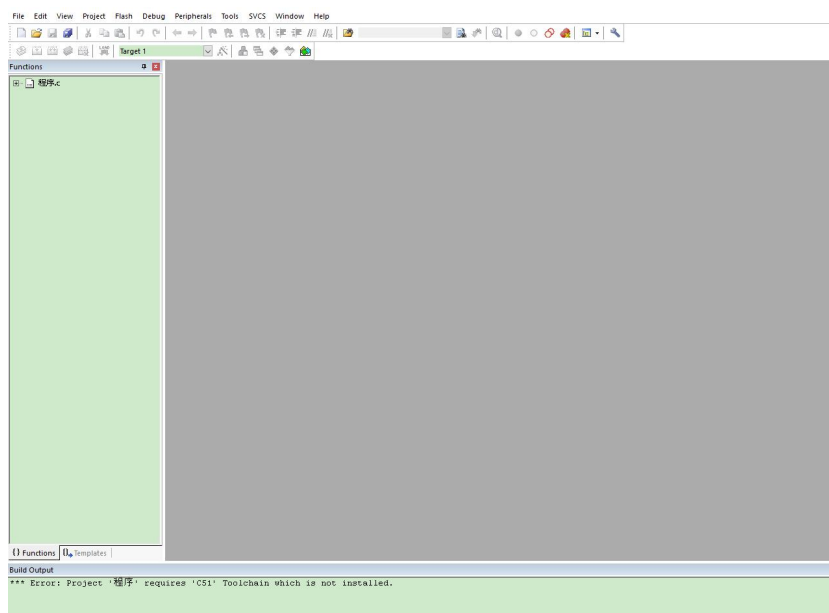


图 5.6 软件设计环境界面

5.1.3 仿真环境

本毕业设计所使用的仿真环境为 Proteus 软件。Proteus 是一个著名的 EDA 工具（仿真软件）。可以执行实时仿真来调试硬件电路和软件代码。界面如图 5.7 所示。

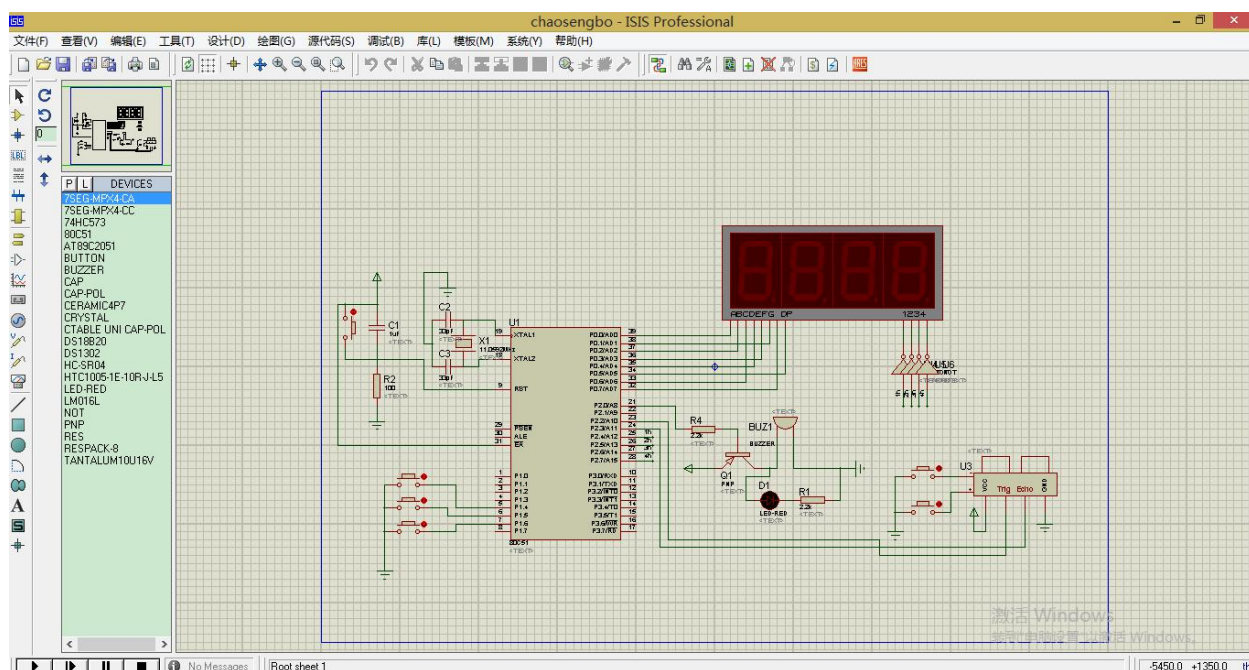


图 5.7 仿真环境界面

5.2 超声波模块调试与误差分析

依据超声波模块的工作原理以及实际使用的情况，分析其误差来源于一下几个方面：

- 理论上发射和计时是同时进行的，但实际两者之间会存在一定的偏差；
- 从收到回波信号到被检测电路检测出信号之间存在一定时间的滞后；
- 从单片机的指令操作间存在时间上微小的误差；
- 计时器本身的误差；
- 超声波的传播速度会受到实际工况下环境的影响。。

对于以上可能会出现的误差问题分析如下：

a、c、d 项的误差都是来自于 MCU 指令处理的问题。由于 MCU 处理每一次只能处理一条指令，不能并行处理，虽然机器的计算速度很快，但是理论上还是存在一定量的时间误差。超声波模块与单片机一样，不能同时处理两个事件，也就是计时器实际记录的时间是与超声波实际传播的时间是存在误差的。选用误差小的电子元器件，在软件上进行合理的优化，就可以将这些误差降低。

b 项误差是由于超声波模块的内部电路所造成的，收到回波信号和被检测电路检测出信号这两个动作在实际操作是不可能按照理想状态同时进行，从而使得收到实际回波到电路确认并输出相应信号之间存在一定的时间差，此误差由超声波模块的内部电路造成，只能通过选择合理的超声波模块将误差降低[5]。

e 项误差是由于工作环境的差异对超声波的影响，由于温度湿度等元素对声波传输速

度有很大的影响,不同的温湿度会对应不同的传播速度,对从而使计时的频率受到了影响,检测电路的灵敏度也随之变化,环境对其的影响较为复杂,但是其影响量有限。本项误差也是实际中影响最大的一项[5]。

表 5.1 超声波测距误差分析表

测量温度 25℃		测量温度 30℃	
实测距离	实测距离（50 次取平均值）	实测距离	实测距离（50 次取平均值）
50mm	51mm	100mm	101mm

5.3 仿真调试

通过使用 Proteus 软件构建电路图,然后将软件环境 Keil 生成的十六进制文件加载到相关器件中,即可执行仿真。通过仿真可以初步调试代码,验证硬件电路,为系统的稳定运行打好基础。仿真结果如图 5.8 所示。

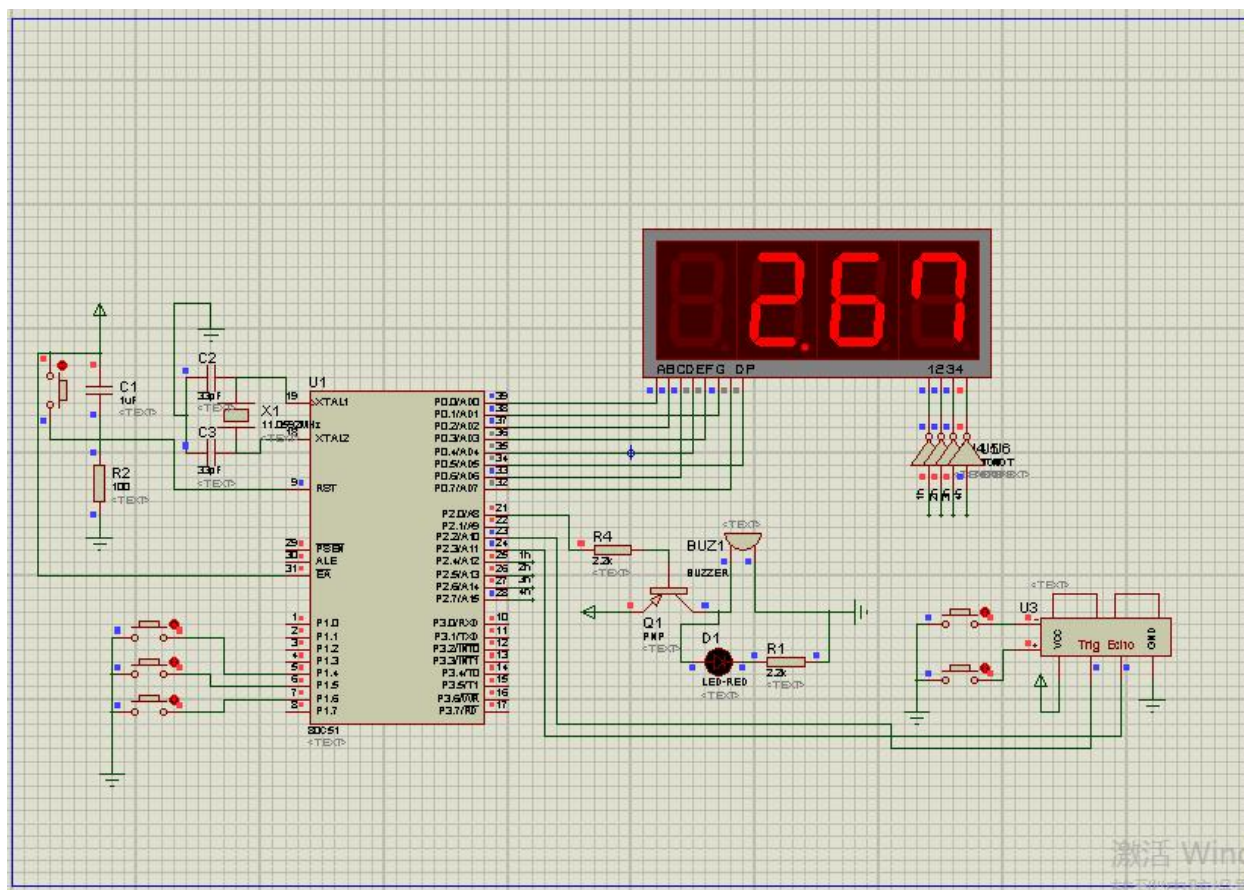


图 5.8 仿真结果图

5.4 硬件实物及其工作状态

5.4.1 硬件实物

通过焊接将各个硬件模块连接在一起，并对每个模块的连接点进行检查。确保每个连接点能够导通，以使得整体硬件能够正常工作。硬件实物如图 5.9 所示

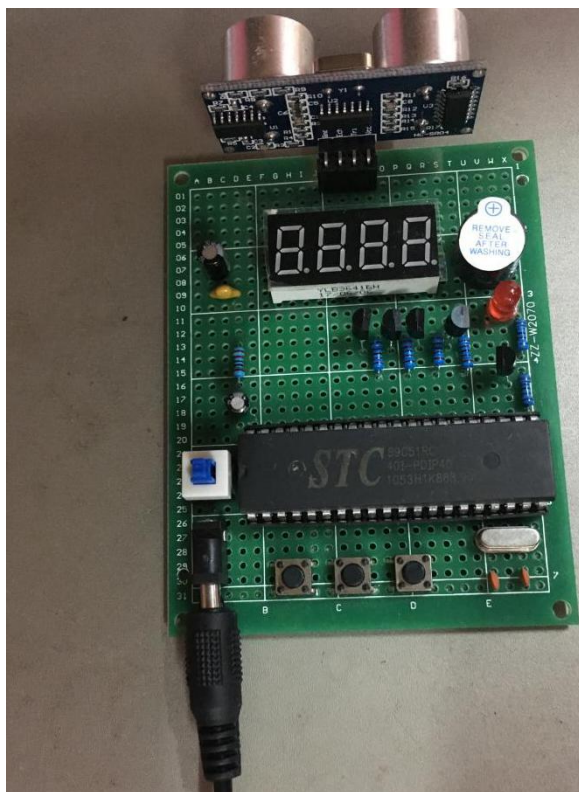


图 5.9 硬件实物图

5.4.2 硬件工作状态

调试好硬件后，通过 USB 对硬件进行供电。将模拟测量物放在超声波模块后，打开电源。硬件开始正常工作，测得距离与实际距离误差较小，硬件调试结束。其工作状态如图 5.10 所示

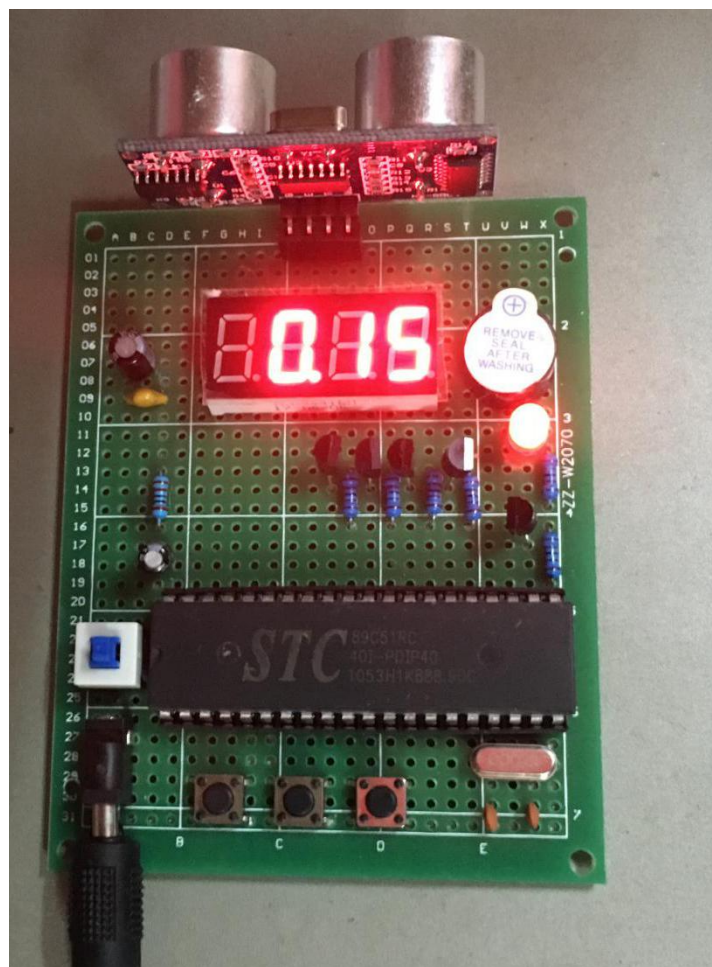


图 5.10 硬件工作状态图

5.5 调试小结

理论的计算以及通过软件的仿真都与实际的系统运行存在这一定的差距。通过初期的仿真调试可以初步得到系统的参考的数据，不断的改变仿真参数，使其最大限度的接近实际的应用环境与工作状态。数据的理论计算与实际采集也从在一定的误差，调试软件参数，将误差减小到最小。分析各个误差出现的原因，寻在相应的结局方案，调试是的系统工作在最稳定的状态。

结束语

随着时间流逝，我走到了我学生生涯的终点。毕设是我学生生涯的最后一部分，它不但是将我们所学的专业知识综合的运用，更是检验和升华我们所学的知识，促使我们的学习能力进一步提升。

当没有进行毕设之前，觉得它可能只是运用这几年的所学知识来研究一个科学课题，但是通过这次经历发现自己的考虑的太不全面了。毕设不单是对自己大学所学知识的一种检查，也是对自己综合能力的一种提升。通过这次毕设，我更深刻地认识到学习需要长期积累。

当选定了题目后，我便立刻开始了资料的收集。虽然我的题目中用到的各种元器件对我而言是比较熟悉的，但是我还是第一次这么仔细地去了解它们。我翻阅了各类的相关资料，将这些收集来的资料进行分类，筛选和细化。尽可能的使我的资料更加全面一点，这样方便之后论文的书写与硬件的制作。

在编写软件的过程中，由于缺乏知识，我遇到了很大的困难。幸运的是，我的指导老师给了我很多帮助。

在硬件调试过程中，数码管的显示出现了问题。我检查了一遍又一遍的电路原理图，但是仿真没有出现问题。我又开始检查焊接点，焊接也是没有问题的。在同学的提醒下我才发现是在元器件的选择上出现了问题。在显示模块中，我用四位一体的数码管来进行显示，由于许多元器件都是在网上购买的，所以并没有注意它是共阳极数码管还是共阴极数码管。最后错买成共阴极数码管，导致单片机输出的电平不能带动数码管显示。通过这个问题，我对数码管的了解又加深了，我想这就是毕业设计的意义吧！

、

致 谢

大学四年的时光如白驹过隙一般飞速流逝。当我回过神时我的学生时代已经结束了大学四年是精彩的也是平淡的。在大学这个舞台上，我到了谢幕的时刻。而这篇毕业论文，仿佛就是我为这个舞台准备的最后一句台词了。当然，我最要感谢的就是陪我一起谢幕的毕业设计指导老师——侯铁双老师。

在整个毕设的过程中有过失败，有过烦躁，同学们给我鼓励是我遇到困难瓶颈时最强有力的动力。他们总是给予我帮助，安慰我、鼓励我，让我一次次从心情的低谷爬出来，重新斗志昂扬的完成任务。大家在在段日子里相互扶持、相互鼓励，有遇到难题一起讨论并互相帮助的解决掉难题，遇到困难也丝毫不气馁，一起度过这段艰苦奋斗、枯燥乏味但充满希望的日子。

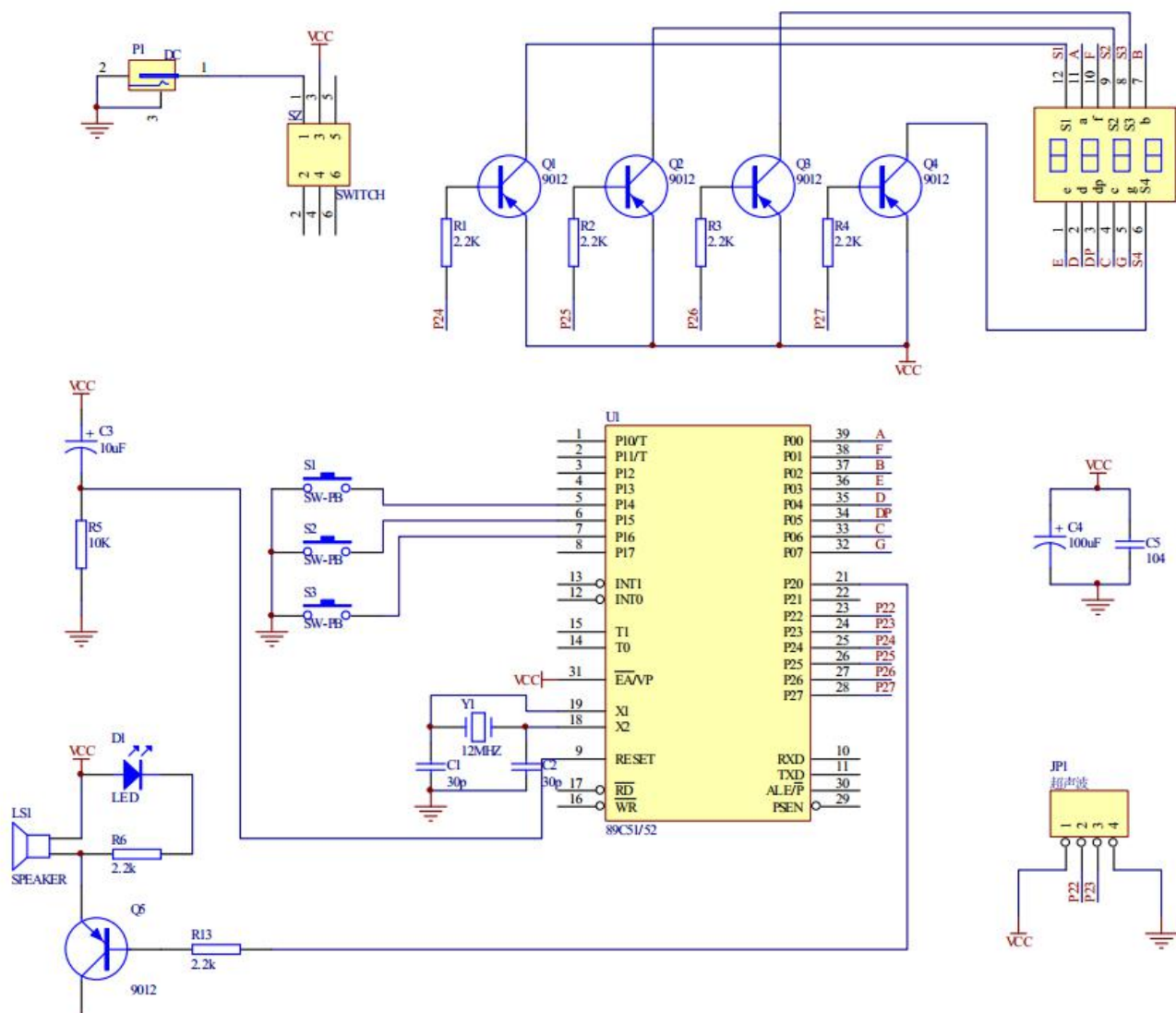
在一次又一次的成功与失败中，我通过努力的学习，终于完成了最后的毕业设计和论文写作。老师的指导、同学的帮助是我毕设过程中最大的动力。通过努力工作，不辜负老师的期望，老师精心教导的场景激励我不断努力工作。毕设中有很多新的知识，自己的查阅资料学习、老师的指导、同学的交流讨论使得我很快的掌握新的知识。我衷心感谢老师的指导，并感谢老师的支持。

参考文献

- [1] 计长安, 工业控制计算机[M], 北京: 清华大学出版社, 2014 年
- [2] 段鸿斌, 超速报警器[M], 西安: 西北工业大学硕士学位论文, 2010 年
- [3] 刘卓娅, 基于 STC89C52 的无线超声波测距系统的研究与设计[M], 广州市: 华南理工大学博士学位论文, 2017 年
- [4] 杨 扬, 超声波测距仪的研究与设计[J], 电子测试, 2017 年, 第三卷 22 期: 34-36 页
- [5] 李方旭, 基于 HC—SR04 超声波传感器的智能避障小车设计[M], 北京, 北京理工大学硕士学位论文, 2017 年
- [6] 寻艳芳, 超声波传感器[J], 电脑迷, 2014 年, 第三卷 22 期: 36-40 页
- [7] 张宁宁, 非接触多模态手成像采集系统设计, 重庆: 重庆邮电大学硕士学位论文, 2014 年
- [8] Bikeev, Simulation of VVER-1000 startup physics tests using the MC U Monte Carlo code[J], ANNALS OF NUCLEAR ENERGY, 2018, 117 卷 13 期: 60-66 页
- [9] TANG YANG, Digital System for Designing E-linac Tube[J], Annual Report of China Institute of Atomic Energy 2017, 75 卷 11 期: 32-35 页
- [10] 刘正刚, Study of errors in ultrasonic heat meter measurements caused by impurities of water based on ultrasonic attenuation[J], Journal of Hydrodynamics, 2015, 36 卷 20 期: 102-109 页码

附录

附录 A 系统原理图



附录 B 部分程序代码

```

#include <reg52.h> // 器件配置文件
#include <intrins.h>
#include "eeprom52.h"
// 传感器接口
sbit RX = P2^3;
sbit TX = P2^2;
// 按键声明
sbit S1 = P1^4;
sbit S2 = P1^5;
sbit S3 = P1^6;

sbit DIAN=P0^5; // 小数点引脚
// 蜂鸣器
sbit Feng=P2^0;

// 变量声明
unsigned int time=0;
unsigned int timer=0;
unsigned char posit=0;
unsigned long S=0;
unsigned long BJS; // 报警距离
char num=0;
// 模式 0 正常模式 1 调整
char Mode=0;
bit flag=0;

unsigned char const discode[] = {0x5F, 0x44, 0x9D, 0xD5, 0xC6, 0xD3, 0xDB, 0x47, 0xDF, 0xD7, 0x80}; // 数码管显示码 0123456789- 和不显示

unsigned char disbuff[4] = {0, 0, 0, 0}; // 数组用于存放距离信息
unsigned char disbuff_BJ[4] = {0, 0, 0, 0}; // 报警信息
sbit W0=P2^4;
sbit W1=P2^5;
sbit W2=P2^6;
sbit W3=P2^7; // 数码管的位选引脚

/*****把数据保存到单片机内部eeprom中*****/
void write_eeprom()
{
/*****把数据从单片机内部eeprom中读出来*****/
void read_eeprom()
{
/*****开机自检eeprom初始化*****/
void init_eeprom()
{
// 延时100ms (不精确)
void delay(void)
{
// 按键扫描
void Key_()
{
/*****扫描数码管*****/
void Display(void)
{
/*****计算*****/
void Conut(void)
{
/*****定时器0*****/
// 定时器0
void zd0() interrupt 1 // T0中断用来计数器溢出, 超过测距范围
{
/*****定时器1*****/
// 定时器1
void zd3() interrupt 3 // T1中断用来扫描数码管和计800ms启动模块
{
/*****主函数*****/
void main(void)
{

```

```

void write_eeprom()
{
    SectorErase(0x2000);    //先清除
    byte_write(0x2000, BJS); //写数据
    byte_write(0x2060, a_a);
}

/*****把数据从单片机内部eeprom中读出来*****/
void read_eeprom()
{
    BJS = byte_read(0x2000);
    a_a = byte_read(0x2060);
}

/*****开机自检eeprom初始化*****/
void init_eeprom()
{
    read_eeprom(); //先读
    if(a_a != 1) //新的单片机初始单片机内eeprom
    {
        BJS = 25;
        a_a = 1;
        write_eeprom(); //保存数据
    }
}

//延时100ms (不精确)
void delay(void)
{
    unsigned char a,b,c;
    for(c=10;c>0;c--)
        for(b=38;b>0;b--)
            for(a=130;a>0;a--);
}

//主函数
void main(void)
{
    TMOD=0x11;    //设T0为方式1, GATE=1;
    TH0=0;
    TL0=0;
    TH1=0xf8;    //2MS定时
    TL1=0x30;
    ET0=1;    //允许T0中断
    ET1=1;    //允许T1中断
    TR1=1;    //开启定时器
    EA=1;    //开启总中断
    init_eeprom(); //开始初始化保存的数据
    while(1)
    {
        while(!RX); //当RX为零时等待
        TR0=1;    //开启计数
        while(RX); //当RX为1计数并等待
        TR0=0;    //关闭计数
        Conut();    //计算
    }
}

```

```

//按键扫描
void Key_()
{
    //+
    if(S1==0&&Mode==1)          //设置状态时，按下加按键
    {
        delay();      //延时去抖
        delay();      //延时去抖
        while(S1==0)    //按键松开
        {
            P1=P1|0x0f;      //不显示
        }
        BJS++;    //报警值加
        if(BJS>=151) //最大150，加到大于等于151就会变成0
        {
            BJS=0;      //清零
        }
        write_eeprom();    //保存数据
    }
    //-
    if(S2==0&&Mode==1)
    {
        delay();
        delay();      //延时去抖
        while(S2==0)
        {
            P1=P1|0x0f;
        }
        BJS--;    //报警值减
        if(BJS<=1)    //最小1
        {
            BJS=150;
        }
        write_eeprom();    //保存数据
    }
    //功能
    if(S3==0)    //设置键
    {
        delay();
        delay();      //延时去抖
        while(S3==0)
        {
            P1=P1|0x0f;
        }
        Mode++;    //模式加
        num=0;
        if(Mode>=2)    //加到2时清零
        {
            Mode=0;      //退出设置状态
        }
    }
}

```