

## 1 设计概述

### 1.1 设计预期

- (1) 设计出超声波测距器的硬件结构电路。
- (2) 设计出超声波测距器的系统程序。
- (3) 对系统硬件和软件进行联调，使系统能在一定距离范围内精确的测距。

### 1.2 设计指标

- (1) 超声波测距器能够测量的距离范围达到 1 米。
- (2) 超声波测距器的误差不超过 5 厘米。
- (3) 能对测得的距离数据进行实时显示。
- (4) 能暂存测得的距离数据。
- (5) 当测得的距离小于 10 厘米时，蜂鸣器能发声报警。
- (6) 整个系统能完全通过电池供电稳定工作。

## 2 系统硬件设计

### 2.1 单片机主控模块

单片机也叫做“微控制器”或者“嵌入式微控制器”。它不是完成某一个逻辑功能的芯片，而是把一个微型计算机系统集成到一个芯片上。概括的讲，一块芯片就成了一台计算机。它体积小、质量轻、价格便宜，为学习、应用和开发提供了便利条件。近年来，微处理器已广泛应用于多种领域，尤其是在智能仪器仪表中的应用更是如此，这不仅引起了产品本身的变革，也深深地影响设计的理念的变革。智能仪器仪表作为一种智能系统，其核心在于微处理器。基于微处理器的智能系统设计，已成为目前电子设计领域的一个热点。智能系统是一个复杂的系统，一般包含微处理器、按键与显示人机界面、A/D 转换、D/A 转换等基本功能部件，同时也包含与应用领域相关的其他特殊部件。智能系统一般需要在恶劣的环境下长期连续地工作，因此在满足功能的基础上，其可靠性也是设计时需要考虑的一个方面，目前已经普遍应用于通信、雷达、遥控和自动控制等各个领域。在本次课程设计中我选用的是 AT89S52 单片机，是一种带 8K 字节 FLASH 存储器的低电压、高性能的 CMOS 微处理器。

51 单片机芯片原理图如图 2-1 所示。

#### 2.1.1 单片机时钟电路

对于单片机时钟电路，用于产生单片机工作所需要的时钟信号，而时序所研究的是指令执行中各信号之间的相互关系。单片机本身就如一个复杂的同步时序电路，为了保证同步工作方式的实现，电路应在唯一的时钟信号的严格控制下。

单片机内部有一个高增益反相放大器，其输入端为芯片引脚 XTAL1，其输出端为引脚 XTAL2。而在芯片的外部，XTAL1 和 XTAL2 之间跨接晶体振荡器和微调电容，从而构成一个稳定的自激振荡器。

---

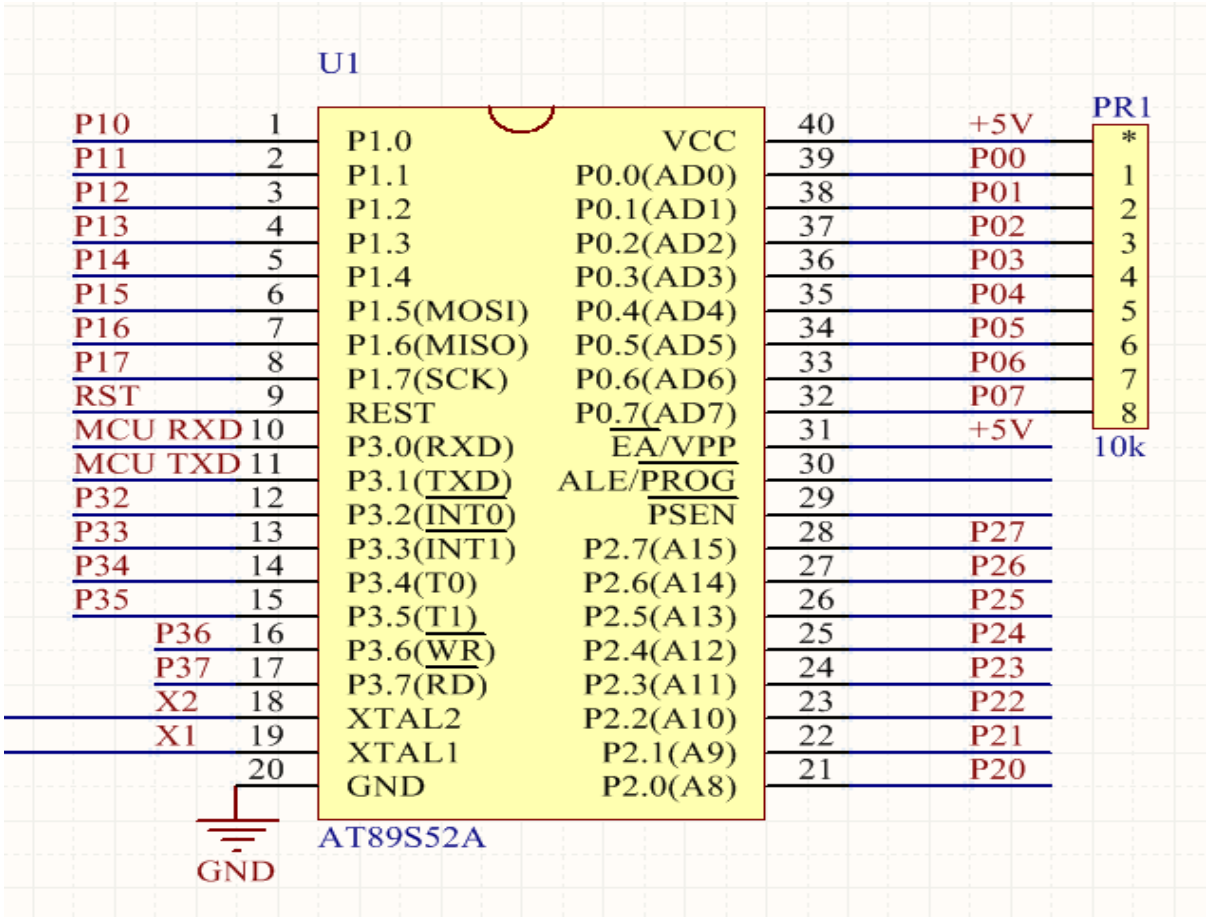


图 2-1 51 单片机芯片原理图

电容器 C1 和 C2 的作用是稳定频率和快速起振，电容值的范围在 5pF 到 30pF, 典型值为 30pF。晶振的频率为 22.1184MHz。只要在单片机的 XTAL1 和 XTAL2 引脚外接晶体振荡器就构成了自激振荡器并在单片机内部产生时钟脉冲信号。

单片机时钟电路如图 2-2 所示。

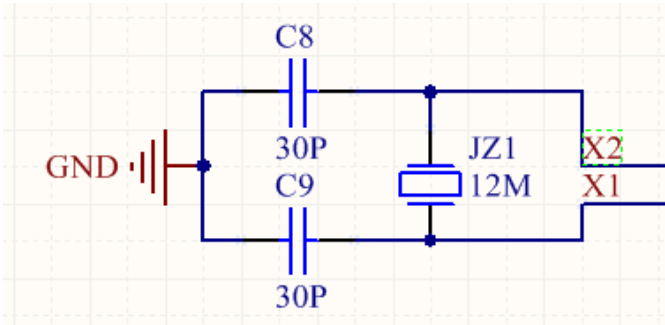


图 2-2 单片机时钟电路

2.1.2 单片机复位电路

复位电路是使单片机的 CPU 或系统中的其他部件处于某一确定的初始状态，并从这上状态开始工作。

本单片机的复位电路有两种，即上电复位电路和按键复位电路。上电复位是单片机上电时复位操作，保证单片机上电后立即进入规定的复位状态。按键复位电路即要实现复位的话，只要按下 RESET 键即可。它主要是利用电阻的分压来实现的。

上电复位要求接通电源后，单片机自动实现复位操作。上电瞬间 RESET 引脚获得高电平，随着电容的充电，RESET 引脚的高电平将逐渐下降。RESET 引脚的高电平只要能保持足够的时间（2 个机器周期），单片机就可以进行复位操作。

上电与按键均有效的复位电路不仅在上电时可以自动复位，而且在单片机运行期间，利用按键也可以完成复位操作。单片机复位电路的原理图如图 2-3 所示。

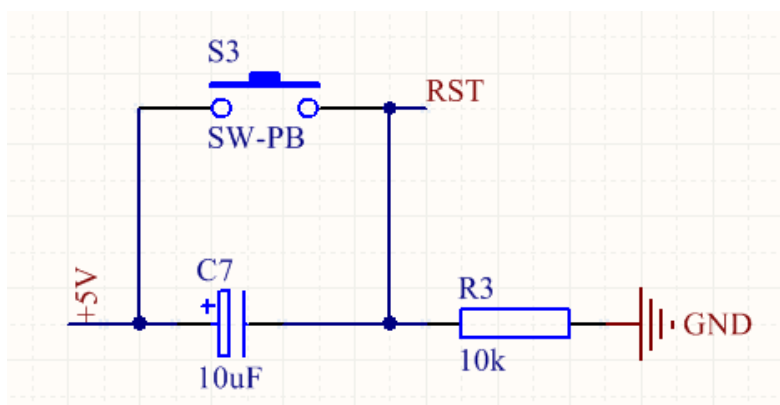


图 2-3 单片机复位电路

### 2.1.3 蜂鸣器警报电路

蜂鸣器警报电路用于发声警报，当测得距离在警报区间（小于 10 厘米）时，报警提醒用户此时距离过近。由于 51 单片机管脚驱动能力不足，无法直接输出或输入足够大的电流来驱动蜂鸣器，故采用 8550PNP 管对蜂鸣器进行扩流。当蜂鸣器控制管脚 P10 输出低电平时，三极管饱和导通，蜂鸣器上流过较大的电流，发声报警。当 P10 输出高电平时三极管停止报警。

蜂鸣器报警电路如图 2-4 所示。

## 2.2 液晶显示模块

液晶显示部分主要采用 LCD1602 液晶显示器。该模块具有如下特点：可分为 2 行共显示 32 个字符，每行 16 个字符；其可与 8 位或 4 位微处理器，其可与 8 位或 4 位微处理器连接；内藏式字符发生器 ROM，可提供 160 种工业标准字符，包括全部大小写字母、阿拉伯数字及片假名，以及 32 个特殊字符与符号的显示；需 +5V 单电源供电；低功耗。

对于 LCD1602 液晶显示模块，它的指令系统包括清显示指令、光标返回指令、输入模式设置指令、显示开关控制指令、光标或字符移位指令、功能设置指令、字符发生存储器地址设置指令、数据存储器地址设置指令、读忙标志或光标地址指令、写数据指令和读数据指令。

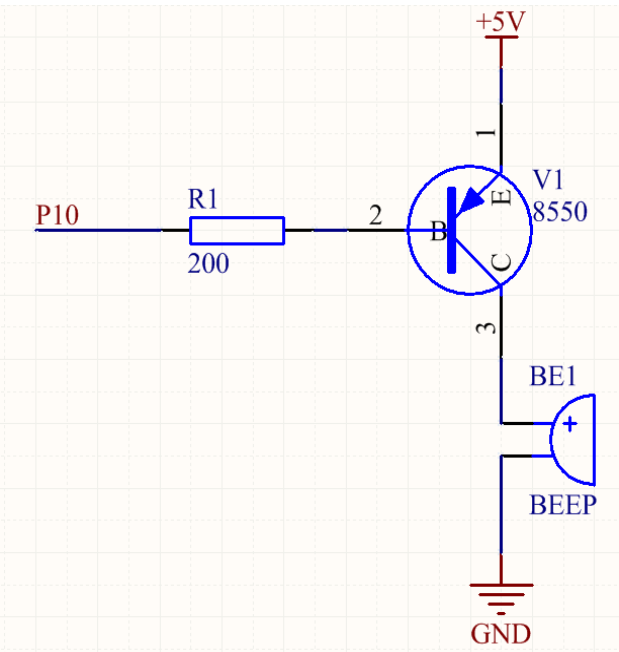


图 2-4 蜂鸣器警报电路

本次课程设计的内容主要是显示超声波测量的距离，使用 LCD1602 不但可以成功用数字表示距离显示，还可以显示温度以确认测距精准度。模块功耗小、体积小、显示内容丰富、超薄轻巧。可在本系统中作为显示设备显示出准确、有序而美观的测量结果

液晶驱动电路原理图如图 2-5 所示。

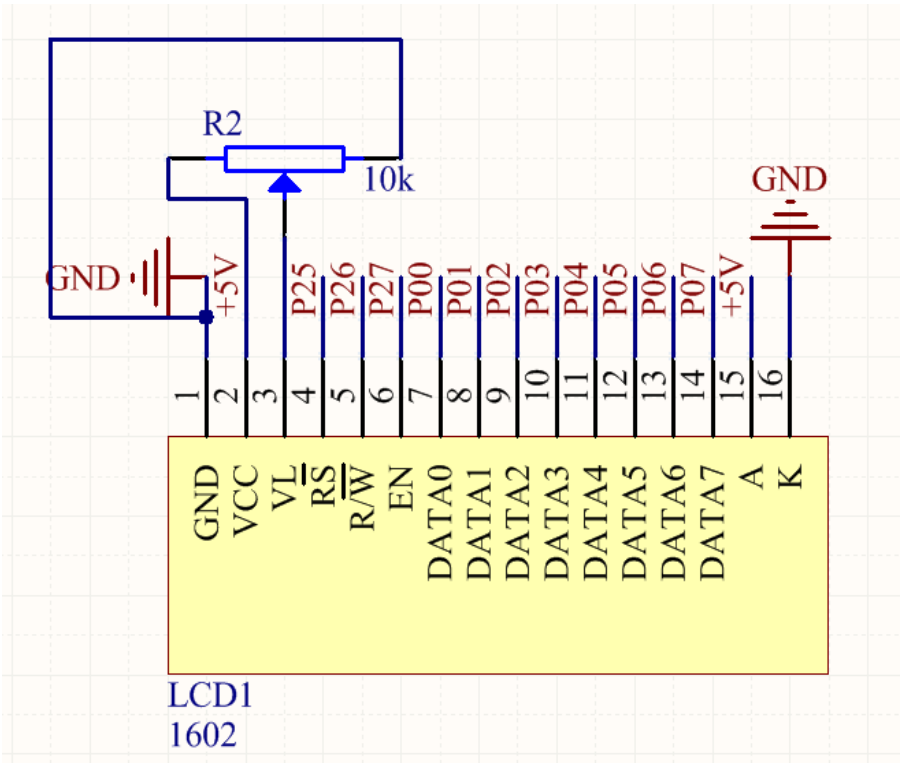


图 2-5 液晶显示模块原理图

## 2.3 超声波测量模块

### 2.3.1 超声波的特性

声音是与人类生活紧密相关的一种自然现象。当声的频率高到超过约 20kHz 时，人耳听觉的频率达到极限，人们就会觉察不出周围声的存在，因而称这种高频率的声为“超”声。

超声波的特性有以下四个方面：

首先是束射特性。由于超声波的波长短，超声波射线可以和光线一样，能够反射、折射，也能聚焦，而且遵守几何光学上的所有定律。即超声波射线从一种物质表面反射时，入射角等于反射角，当射线透过一种物质进入另一种密度不同的物质时就会产生折射现象，也就是要改变它的传播方向，两种物质的密度差别愈大，则折射率也愈大。

其次是吸收特性。声波在各种介质中传播时，随着传播距离的增加，其强度会逐渐减弱，这是因为介质要吸收掉它的部分能量。对于同一介质，声波的频率越高，介质吸收就越强。对于一个频率一定的声波，在气体中传播时吸收尤为厉害，在液体中传播时吸收就比较弱，在固体中传播时吸收是最小的。

再次是能量传递特性。超声波之所以能在各个工业部门中得到广泛的应用，主要原因还在于比声波具有强大得多的功率。为什么有这么强大的功率呢？因为当声波进入某一介质中时，由于声波的作用使物质中的分子也随之振动，振动的频率和声波频率一样，分子振动的频率决定了分子振动的速度。频率愈高速度愈大。物质分子由于振动所获得的能量除了与分子本身的质量有关外，主要是由分子的振动速度的平方决定的，所以如果声波的频率愈高，也就是物质分子愈能得到更高的能量。超声波的频率比普通声波要高出很多，所以它可以使物质分子获得很大的能量；换句话说，超声波本身就可以供给物质分子足够大的功率。

最后是声压特性。当声波进入某物体时，由于声波振动使物质分子相互之间产生压缩和稀疏的作用，将使物质所受的压力产生变化。由于声波振动引起附加压力现象叫声压作用。

### 2.3.2 超声波发射部分

发射电路主要由反相器 CD4069 和超声波发射换能器 T 构成。由于在距离较远的情况下，超声波的回波很弱，因而转换为电信号的幅值也较小，为此要求将信号放大。单片机端口输出的 40kHz 的方波信号一路经一级反相器后送到超声波换能器的一个电极，另一路经两级反相器后送到超声波换能器的另一个电极，用这种推挽形式将方波信号加到超声波换能器的两端，可以提高超声波的发射强度。输出端采用两个反相器并联，用以提高驱动能力。上位电阻 R10、R11 一方面可以提高反向器输出高电平的驱动能力，另一方面可以增加超声波换能器的阻尼效果，缩短其自由振荡时间。

超声波发射部分如图 2-6 所示。

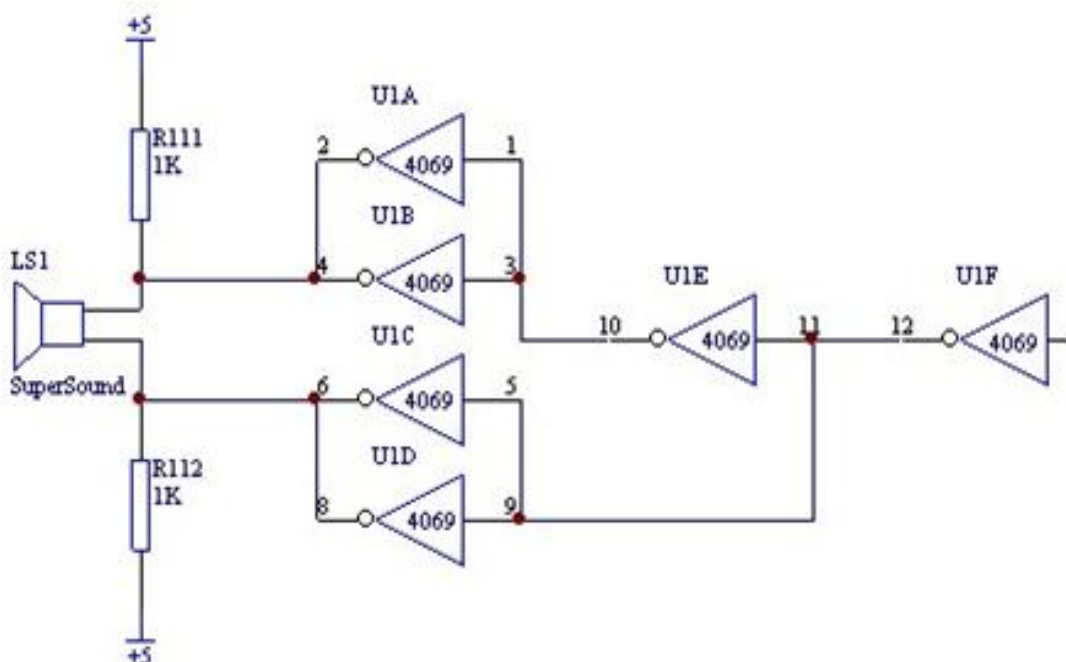


图 2-6 超声波发送部分电路图

### 2.3.3 超声波接收部分

超声波检测接收电路主要是由集成电路 CX20106 组成，这是一款应用广泛的红外线检波接收的专用芯片，其具有功能强、性能优越、外围接口简单、成本低等优点，常用于电视机红外遥控接收器。考虑到红外遥控常用的载波频率 38kHz 与测距的超声波频率 40kHz（超声波传感器的声压能级、灵敏度在 40kHz 时最大）较为接近，可以利用它制作超声波检测接收电路。适当更改电容 C4 的大小，就可以改变接收电路的灵敏度和抗干扰能力。

CX20106 内部由前置放大器、限幅放大器、带通滤波器、检波器、积分器及整形电路构成。使用 CX20106A 集成电路对接收探头受到的信号进行放大、滤波，其总放大增益 80db。当 CX20106 接收到 40KHz 的信号时，会在第 7 脚 OUT 脚产生一个低电平下降脉冲，单片机外部中断接到这个信号并采取相应动作，使测距结果在 1602 液晶上显示出来。

超声波接收部分如图 2-7 所示。

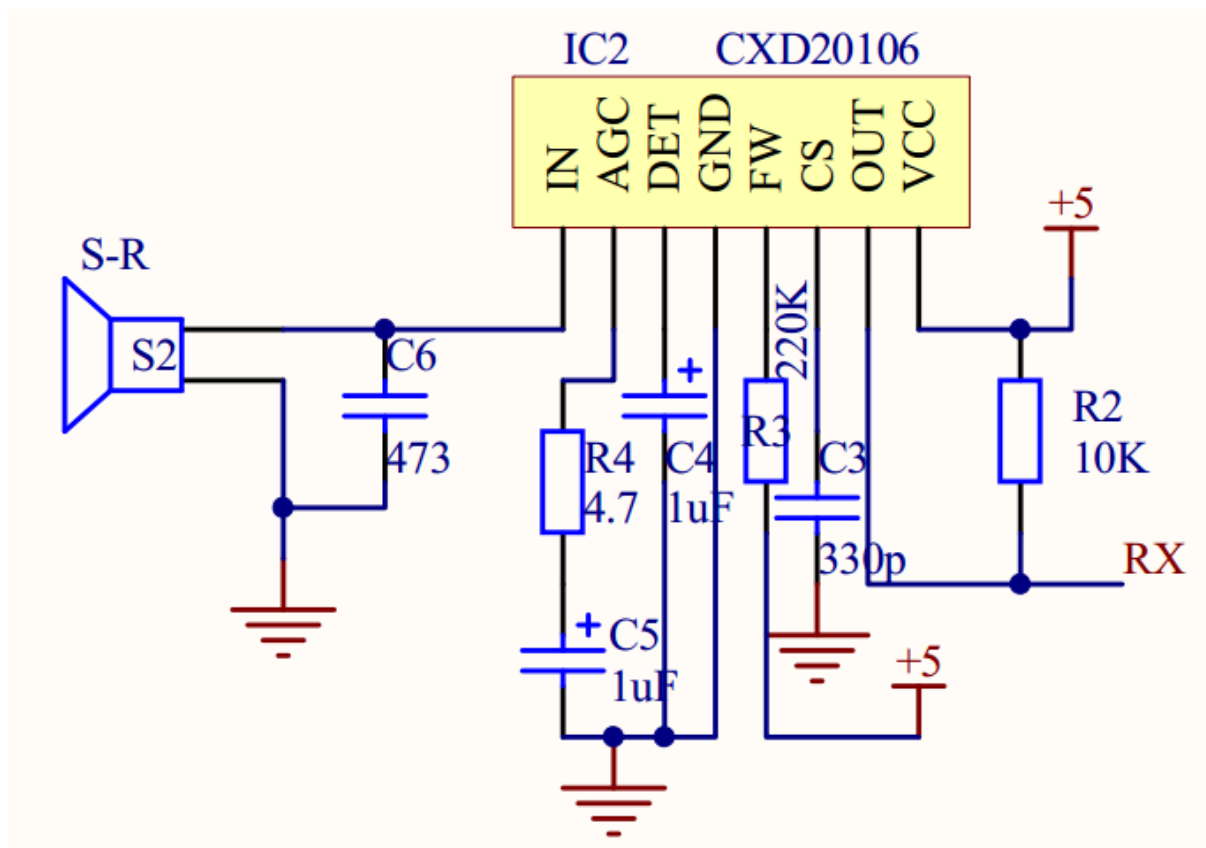


图 2-7 超声波接收部分电路图

## 2.4 电池供电模块

出于便携性的考虑，全部系统都采用电池进行供电，采用两节 18650 锂电池串联工作，串联后的电压约为 7.6V，由于单片机、液晶和超声波模块的工作电压均为 5V，所以在供电模块内加入了 7805 稳压芯片进行降压，电池电压经降压后能稳定地给其他模块进行供电，同时满足了便携性上的要求。

图 2-8 为 7805 降压电路原理图。

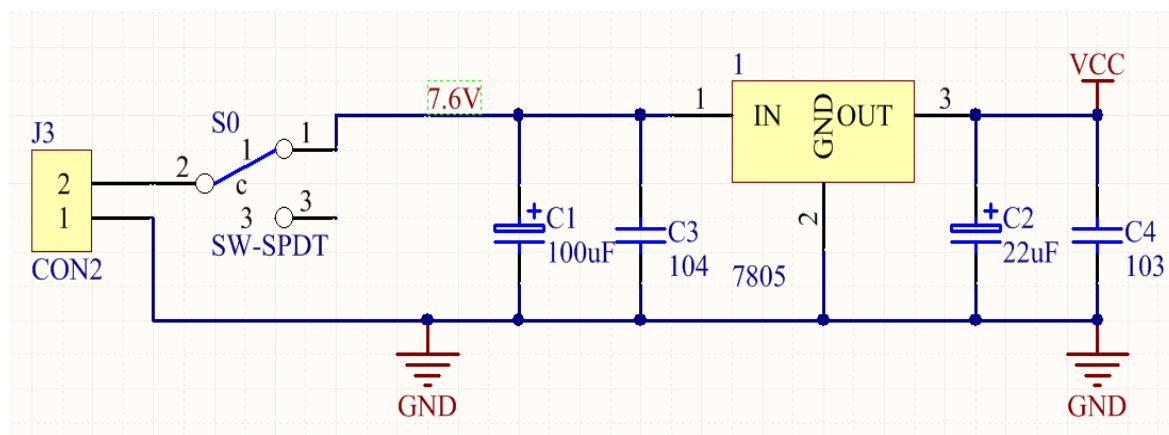


图 2-8 7805 降压电路原理图

### 3 系统软件设计

#### 3.1 软件设计方法

本次单片机的软件采用 Keil uVision4 软件进行 C 语言编程。Keil 提供了丰富的库函数和功能强大的集成开发调试工具，可以完成编辑、编译、连接、调试、仿真等整个开发流程。Keil 软件的优点主要有以下几点：

首先，Keil C51 生成的目标代码效率非常之高，多数语句生成的汇编代码很紧凑，容易理解。在开发大型软件时更能体现高级语言的优势。

Keil C51 软件提供丰富的库函数和功能强大的集成开发调试工具，全 Windows 界面。操作十分方便。

其次，汇编语言是一种面向机器的低级的程序设计语言。它直接利用机器提供的指令系统编写程序，该类程序的可执行指令是与机器语言程序的指令一一对应的。汇编语言由于是面向机器的程序设计语言，与具体的计算机硬件有着密切的关系，因此，可移植性差。但由于汇编指令与机器语言指令一一对应，即一条汇编语言的可执行指令对应着一条机器语言指令，反之亦然。因此，汇编语言可直接利用机器硬件系统的许多特性，如寄存器、标志位以及一些特殊指令等，具有执行速度快、占用内存少等优点。汇编语言的缺点是程序的通用性和可移植性差；程序比较繁琐，调试困难；目标程序比较庞大，运行速度慢。

而 C 语言是一种编译型的程序设计语言，它兼顾了多种高级语言的特点，并具备汇编语言的功能。C 语言有功能丰富的库函数，运算速度快，编译效率高，有良好的可移植性，而且可以直接实现对系统硬件的控制。C 语言是一种结构化设计语言，支持支持由顶向下结构化程序设计技术。C 语言的模块化程序结构可以使程序模块实现共享。在 C 语言的可读性方面更容易借鉴前人的开发经验，提高程序的开发水平。

C 语言应用于单片机编程出了上述特点外，还有以下突出特点：编译器可以自动完成变量存储单元的分配，省去了分配和纪录存储单元的繁琐；不必对单片机和硬件接口的结构有很深入的了解，省去了单片机漫长的学习时间；具有良好的可移植性，只要将程序略加改动就可以将其应用与其他类型的单片机，省去了更改单片机型号时重新编写程序的无奈。因此利用 C 语言编写程序可以大大缩短目标系统软件的开发周期，程序的可读性明显增加，便于改进、扩充、研制规模更大、性能更完备的系统。

对于本次编程，使用 keil 生成的 hex 文件可以用软件直接转化成机械码编入 STC89C52 单片机，简单方便。

#### 3.2 软件流程介绍

超声波测距的原理为超声波发生器 T 在某一时刻发出一个超声波信号，当这个超声波遇到被测物体后反射回来，就被超声波接收器 R 所接收到。这样只要计算出从发出超声波信号到接收到返回信号所用的时间，就可算出超声波发生器与反射物体的距离。

在启动发射电路的同时启动单片机内部的定时器 T1，利用定时器的计数功能记录超



声波发射的时间和收到反射波的时间。当收到超声波反射波时，接收电路输出端产生一个负跳变，在 INT0 端产生一个中断请求信号，单片机响应外部中断请求，执行外部中断服务子程序，读取时间差，计算距离。

主程序首先是对系统环境初始化，设置定时器 T0 工作模式为 16 位定时计数器模式。置位总中断允许位 EA 并给显示端口 P0 和 P2 清 0。然后调用超声波发生子程序送出一个超声波脉冲，为了避免超声波从发射器直接传送到接收器引起的直射波触发，需要延时约 0.1 ms（这也就是超声波测距仪会有一个最小可测距离的原因）后，才打开外部中断 0 接收返回的超声波信号。由于采用的是 12 MHz 的晶振，计数器每计一个数约  $1\mu\text{s}$ ，当主程序检测到接收成功的标志位后，将计数器 T0 中的数（即超声波来回所用的时间）计算即可得被测物体与测距仪之间的距离。

整个系统的流程如图 3-1 所示。

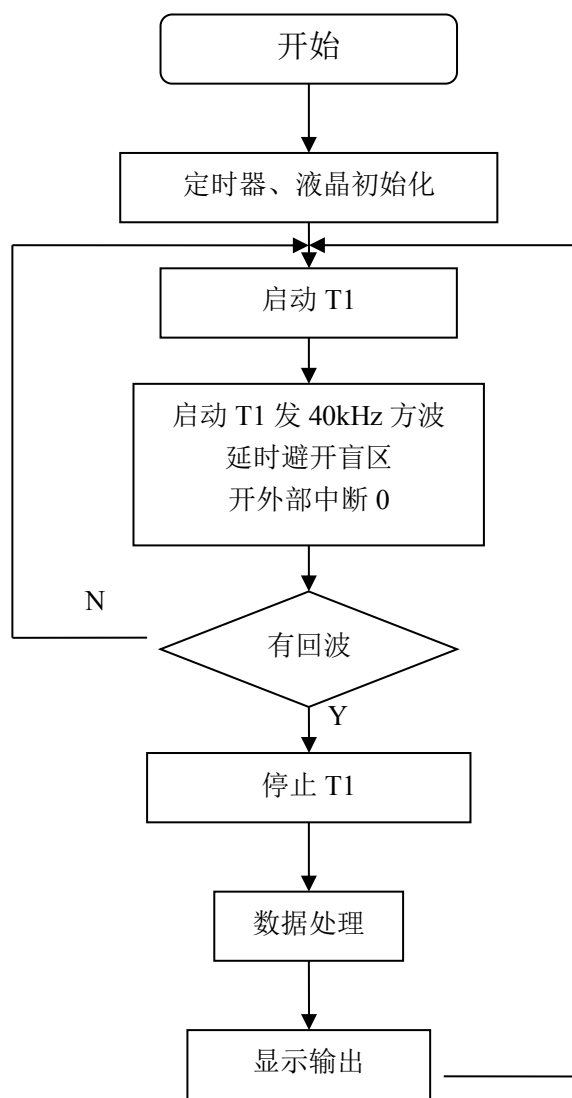


图 3-1 系统软件流程图

## 4 系统的测试与分析

### 4.1 元件的焊接与整板调试

#### 4.1.1 元件的焊接

焊接前对电子元器件要进行测量、选择，严格要求器件参数值（电阻、电容的量值），注意某些有极性要求的器件接法和集成芯片的引脚排列顺序。本系统中元件大部分都是直插式的，焊接时掌握每个焊点的时间，要防止引脚因焊接不当短路或断路。

#### 4.1.2 整板测试

整板测试是在元件焊接完成但未加电前对电路板进行的检查。该过程是系统上电前的检查工作，需要对每个器件逐个引脚进行检查，一方面是检查有没有引脚虚焊或与其他信号线短路，另一方面是对器件引脚功能的再检查，查看设计是否正确。整板测试可以按照先地和电源，再逐个器件引脚的顺序测试。

### 4.2 硬件调试

调试过程是一个非常重要的环节，这是整个电路板成功与否的关键。

首先应断开电源，用数字万用表认真检查电路连线是否有误，是否有短路或者断路和虚焊等的现象。经过初步检查以后，方可接通电源。此时还不应该把芯片装上。

接上电源时应该明确电源的正负极性，切不可接反，一接反则马上可能会出现烧毁重要元件的情况！时刻用手触摸各元件有没有发热现象，并注意其是否冒烟等。待过一段时间稳定后，如没有出现不良的现象，则电路基本上正常。

### 4.3 软件调试

在确认硬件无问题之后，先往单片机内写入一个简单的程序，让所有 IO 口输出低电平，并用万用表进行检测，如此可以确认单片机最小系统已经正常工作，程序可以跑起来了。之后依次调试蜂鸣器、液晶模块和超声波测距模块，用模块化的编程思维控制各个模块，最后再把程序汇总进行联调。经测试，整个系统可以在 1 米之内精确地进行超声波测距及显示，误差小于 5 厘米。同时，能对测得的距离数据进行暂存，一次能暂存两个距离数据，当距离小于 10 厘米时蜂鸣器能进行报警，整个系统能在电池供电时稳定的工作。软件和硬件能协调地进行工作，完全达到了设计指标。

## 5 结束语

经过了一个多月的努力钻研，我终于完成了《基于单片机的超声波测距器》课程设计。从开始选题到完成系统硬件设计，再到论文文章的完成，每走一步对我来说都是新的尝试与挑战。在这段时间里，开始了独立地学习和试验，查看相关的资料和书籍，我学到了很多知识也有很多感受，虽然我的论文作品不是很成熟，还有很多不足之处，但是确是对我自己的一次考验，也是我自己的劳动成果。当看着自己的设计能够无差错地运行，真是莫大的幸福和欣慰。我相信其中的酸甜苦辣最终都会化为甜美的甘泉。这次做论文的经历也会使我终身受益，我感受到做论文是要用心去做的一件事情，是真正的自己学习的过程和研究的过程，没有学习就不可能有研究的能力，没有自己的研究，就不会有所突破，那就失去课程设计的意义了。希望这次的经历能让我在以后学习中激励我继续进步。

## 6 谢辞

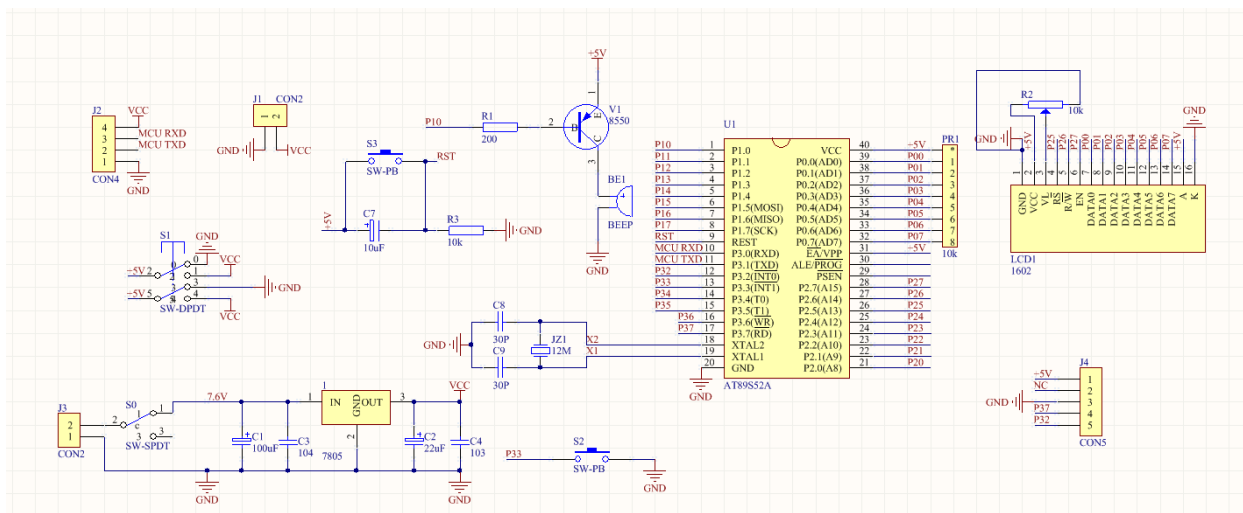
经过努力，我终于把本次课程设计完成了！在此首先要感谢我的指导老师给了我一个选择机会和空间，让我制作了自己想做的东西，也让我对所学的单片机知识有了更深一层的认识。我觉得一个人就是在完成一个又一个实践的过程中逐渐获得经验并得到自我提升的。在这次课程设计过程中。通过资料的查阅，制定方案，设计硬件电路，软件编程，系统调试等一系列过程，极大的丰富了我的专业知识，使我的理论与实际动手能力又有了一个很大的提高，特别是在单片机编程方面又学到了不少新的东西。课程设计使我有机会把自己平时理论学习能够运用到实际中去，理论与实践相结合使我对自己的课程设计更加得心应手，在设计过程中同样也发现自己在各方面还存在不少问题，对各方面的知识掌握得不够透彻。经过这次也使我真正的认识到单片机编程虽然繁琐，但编译成功之后的喜悦是难以言说的，我真正感受到了电子信息工程专业的乐趣。

在此，向王老师致以衷心的感谢！

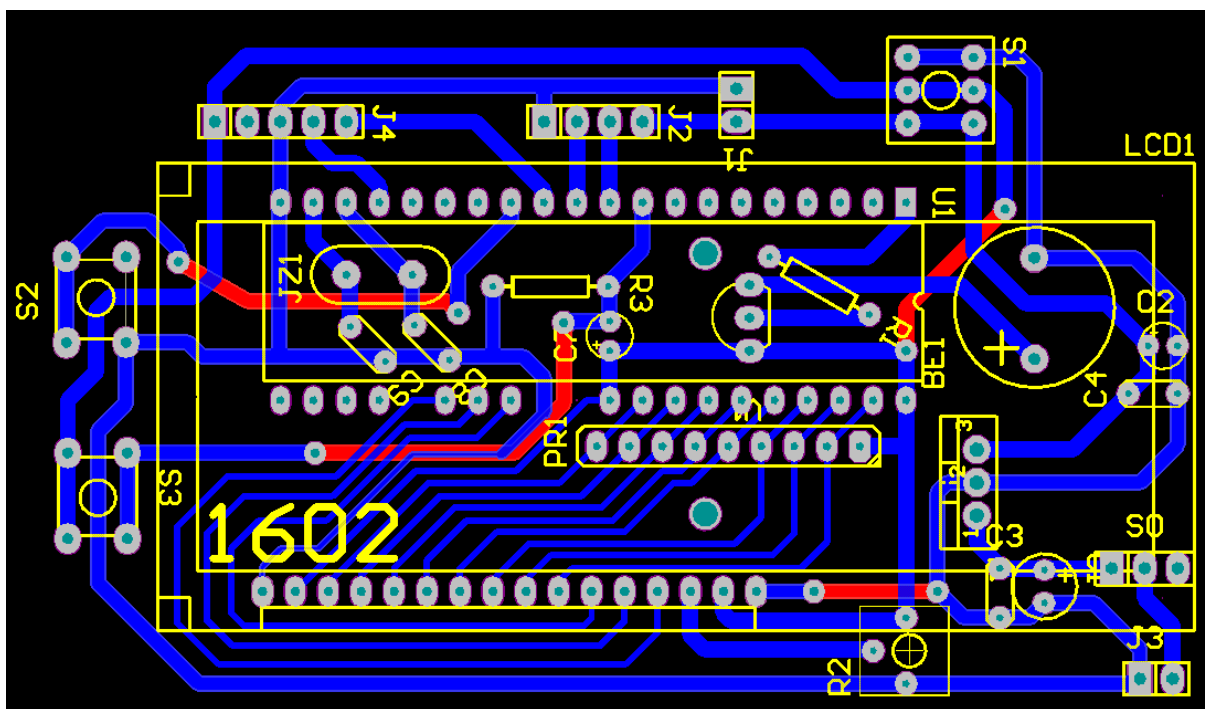
## 参考文献

- [1] 曹月真. 基于 51 单片机的超声波系统的设计. 电子世界. 2011.
  - [2] 张毅刚、彭喜元、姜守达. 《新编 MCS-51 单片机应用设计（第 3 版）》. 哈尔滨工业大学出版社
  - [3] 王晓利. 汽车防追尾无线动态测距系统的研究与应用. 计算机与现代化. 2011
  - [4] 郭清. 基于 STC89C52 的超声波测距防撞系统设计. 仪表技术与传感器. 2011
  - [5] 王安敏、张凯. 基于 AT89C52 单片机的超声波测距系统. 仪表技术与传感器. 2006
-

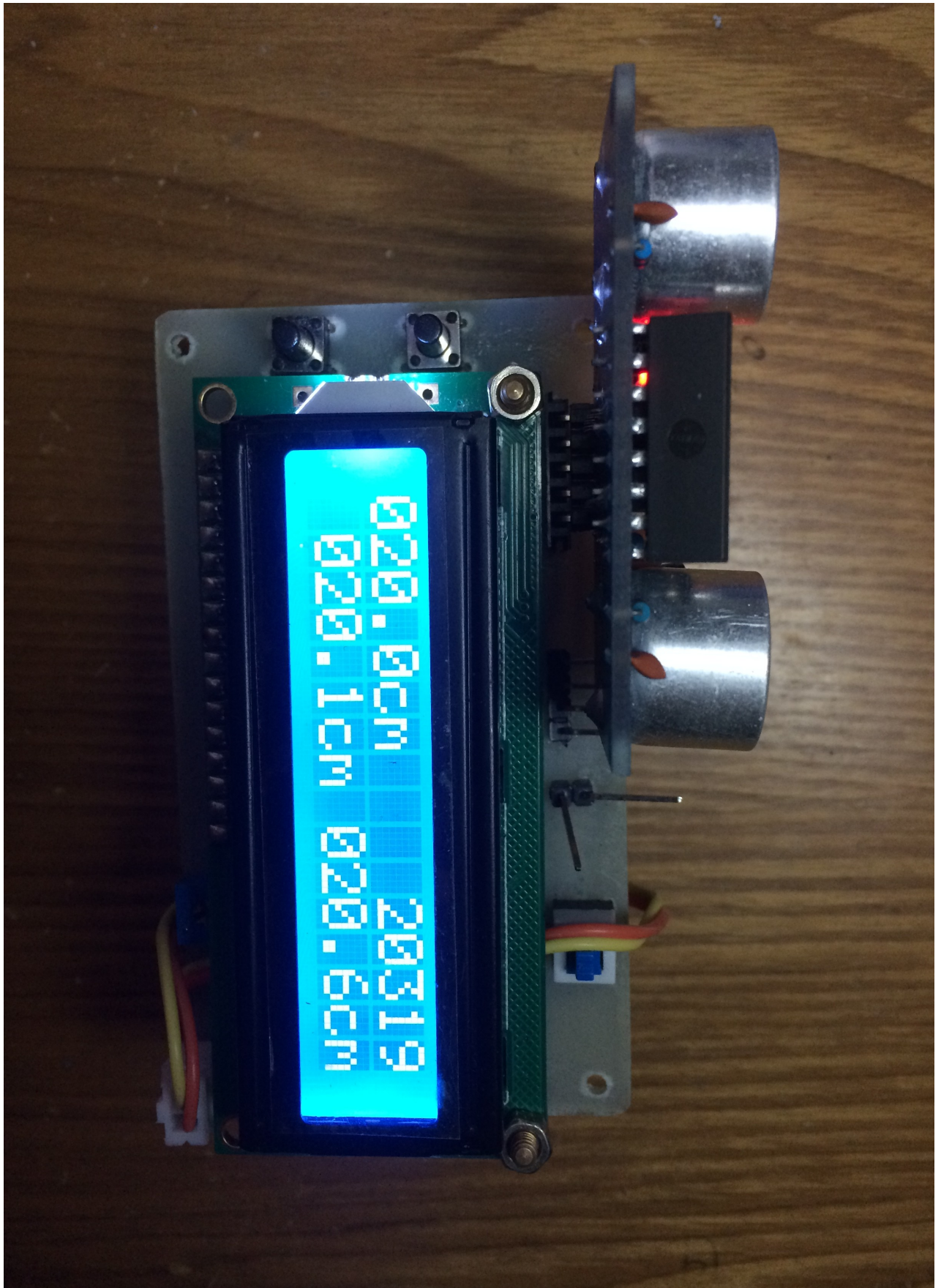
## 附录一 系统电路原理图



## 附录二 系统电路 PCB 图



## 附录三 系统电路实物图



## 附录四 系统源程序

```
//超声波模块显示程序
#include <reg52.h> //包括一个 52 标准内核的头文件
#include<intrins.h> //包含_nop_()函数定义的头文件
#define uchar unsigned char //定义一下方便使用
#define uint unsigned int
#define ulong unsigned long
sbit Tx = P3^7; //产生脉冲引脚
sbit Rx = P3^2; //回波引脚
sbit KEY =P3^3;//按键
sbit BEEP =P1^0;//按键
sbit RS=P2^5; //寄存器选择位，将 RS 位定义为 P2.0 引脚
sbit RW=P2^6; //读写选择位，将 RW 位定义为 P2.1 引脚
sbit E=P2^7; //使能信号位，将 E 位定义为 P2.2 引脚
sbit BF=P0^7; //忙碌标志位，将 BF 位定义为 P0.7 引脚
unsigned char code digit[ ]={"0123456789"}; //定义字符数组显示数字
uint distance[4]; //测距接收缓冲区
uchar ge,shi,bai,qian,temp,flag,outcomeH,outcomeL,i,j; //自定义寄存器
bit succeed_flag; //测量成功标志
//*****函数声明
void conversion(uint temp_data);
void delay_20us();
void pai_xu();

/*****
函数功能：延时 1ms
(3j+2)*i=(3×33+2)×10=1010(微秒)，可以认为是 1 毫秒
*****/
void delay1ms()
{
    unsigned char i,j;
    for(i=0;i<10;i++)
        for(j=0;j<33;j++)
            ;
}

/*****
函数功能：延时若干毫秒
入口参数：n
*****/
void delay(unsigned char n)
{
    unsigned char i;
    for(i=0;i<n;i++)
        delay1ms();
}

/*****
函数功能：判断液晶模块的忙碌状态
返回值：result。result=1，忙碌;result=0，不忙
*****/
unsigned char BusyTest(void)
{
    bit result;
    RS=0; //根据规定，RS 为低电平，RW 为高电平时，可以读状态
    RW=1;
    E=1; //E=1，才允许读写
}
```

```

        _nop_(); //空操作
        _nop_();
        _nop_();
        _nop_(); //空操作四个机器周期，给硬件反应时间
        result=BF; //将忙碌标志电平赋给 result
        E=0; //将 E 恢复低电平
        return result;
    }
}
/*****
函数功能：将模式设置指令或显示地址写入液晶模块
入口参数：dictate
*****/
void WriteInstruction (unsigned char dictate)
{
    while(BusyTest()==1); //如果忙就等待
    RS=0; //根据规定，RS 和 R/W 同时为低电平时，可以写入

指令
    RW=0;
    E=0; //E 置低电平(根据表 8-6，写指令时，E 为高脉冲，
        // 就是让 E 从 0 到 1 发生正跳变，所以应先置“0”

        _nop_();
        _nop_(); //空操作两个机器周期，给硬件反应时间
        P0=dictate; //将数据送入 P0 口，即写入指令或地址
        _nop_();
        _nop_();
        _nop_();
        _nop_(); //空操作四个机器周期，给硬件反应时间
        E=1; //E 置高电平

        _nop_();
        _nop_();
        _nop_();
        _nop_(); //空操作四个机器周期，给硬件反应时间
        E=0; //当 E 由高电平跳变成低电平时，液晶模块开始执

行命令
}
/*****
函数功能：指定字符显示的实际地址
入口参数：x
*****/
void WriteAddress(unsigned char x)
{
    WriteInstruction(x|0x80); //显示位置的确定方法规定为“80H+地址码 x”
}
/*****
函数功能：将数据(字符的标准 ASCII 码)写入液晶模块
入口参数：y(为字符常量)
*****/
void WriteData(unsigned char y)
{
    while(BusyTest()==1);
    RS=1; //RS 为高电平，RW 为低电平时，可以写入数据
    RW=0;
    E=0; //E 置低电平(根据表 8-6，写指令时，E 为高脉冲，
        // 就是让 E 从 0 到 1 发生正跳变，所以应先置“0”
    P0=y; //将数据送入 P0 口，即将数据写入液晶模块
    _nop_();
    _nop_();
    _nop_();

```



```

        _nop_();          //空操作四个机器周期，给硬件反应时间
        E=1;              //E 置高电平
        _nop_();
        _nop_();
        _nop_();
        _nop_();          //空操作四个机器周期，给硬件反应时间
        E=0;              //当 E 由高电平跳变成低电平时，液晶模块开始执行命令
    }
/*****
函数功能：对 LCD 的显示模式进行初始化设置
*****/
void LcdInitiate(void)
{
    delay(15);            //延时 15ms，首次写指令时应给 LCD 一段较长的反应
时间
    WriteInstruction(0x38); //显示模式设置：16×2 显示，5×7 点阵，8 位数
据接口
    delay(5);             //延时 5ms ，给硬件一点反应时间
    WriteInstruction(0x38);
    delay(5);
    WriteInstruction(0x38); //连续三次，确保初始化成功
    delay(5);
    WriteInstruction(0x0c); //显示模式设置：显示开，无光标，光标不闪烁
    delay(5);
    WriteInstruction(0x06); //显示模式设置：光标右移，字符不移
    delay(5);
    WriteInstruction(0x01); //清屏幕指令，将以前的显示内容清除
    delay(5);
}
void main(void) // 主程序
{
    uint distance_data,a,b;
    uchar CONT_1;
    LcdInitiate();      //调用 LCD 初始化函数
    delay(10);           //延时 10ms，给硬件一点反应时间
    WriteAddress(0x0b);
    WriteData('2');
    WriteData('0');
    WriteData('3');
    WriteData('1');
    WriteData('9');
    i=0;
    flag=0;
    Tx=0;               //首先拉低脉冲输入引脚
    TMOD=0x10;          //定时器 0，定时器 1，16 位工作方式
    IT0=0;              //由高电平变低电平，触发外部中断
    EX0=0;              //关闭外部中断
    EA=1;               //打开总中断 0

    while(1)            //程序循环
    {
        WriteAddress(0x00); // 从第 2 行第 6 列开始显示
        WriteData(digit[qian]);
        WriteData(digit[bai]);
        WriteData(digit[shi]);
        WriteData('.')
        WriteData(digit[ge]);
        WriteData('c');
    }

```

```

    WriteData('m');

    EA=0;
    Tx=1;
    delay_20us();
    Tx=0;          //产生一个 20us 的脉冲，在 Tx 引脚
    while(Rx==0); //等待 Rx 回波引脚变高电平
    succeed_flag=0; //清测量成功标志
    EX0=1;          //打开外部中断
    EX1=1;
    TH1=0;          //定时器 1 清零
    TL1=0;          //定时器 1 清零
    TF1=0;          //
    TR1=1;          //启动定时器 1
    EA=1;

    while(TH1 < 30); //等待测量的结果，周期 65.535 毫秒（可用中断实现）
    TR1=0;           //关闭定时器 1
    EX0=0;           //关闭外部中断

    if(succeed_flag==1)
    {
        distance_data=outcomeH;          //测量结果的高 8 位
        distance_data<<=8;              //放入 16 位的高 8 位
        distance_data=distance_data|outcomeL; //与低 8 位合并成为 16 位
        distance_data*=12;               //因为定时器默认为 12 分
        distance_data/=58;               //微秒的单位除以 58 等于
        //为什么除以 58 等于厘米，
        // X 秒=( 2*Y 米)/344 ==》
        Y 米= (X 秒*344) /2
        X 秒=0.0058*Y 米 ==》 厘米=微秒/58
        if(succeed_flag==0)
        {
            distance_data=0;             //没有回波则清零
        }
        distance[i]=distance_data; //将测量结果的数据放入缓冲区
        i++;
        if(i==3)
        {
            distance_data=(distance[0]+distance[1]+distance[2]+distance[3])/4;

            pai_xu();
            distance_data=distance[1];
            a=distance_data;
            if(b==a) CONT_1=0;
            if(b!=a) CONT_1++;
            if(CONT_1>=3)
            { CONT_1=0;
              b=a;
              conversion(b);
            }
            i=0;
        }
    }

```

```
}
//*****
//外部中断 0，用做判断回波电平
INT0_() interrupt 0 // 外部中断是 0 号
{
    outcomeH =TH1; //取出定时器的值
    outcomeL =TL1; //取出定时器的值
    succeed_flag=1; //至成功测量的标志
    EX0=0; //关闭外部中断
}
//*****
//外部中断 1，用做记录距离
INT1_() interrupt 2 // 外部中断是 2 号
{
    delay(15);
    if(KEY==0)
    {
        while(!KEY);
    if(j==2) j=0;
    if(j==0)
    {
        WriteAddress(0x40);
        WriteData(' ');
        WriteData(' ');
        WriteData(' ');
        WriteData(' ');
        WriteData(' ');
        WriteData(' ');
        WriteData('c');
        WriteData('m');

        WriteAddress(0x40);
        WriteData(' ');
        WriteData(digit[qian]);
        WriteData(digit[bai]);
        WriteData(digit[shi]);
        WriteData('.');
        WriteData(digit[ge]);
    }
    if(j==1)
    {

        WriteAddress(0x48);
        WriteData(' ');
        WriteData(' ');
        WriteData(' ');
        WriteData(' ');
        WriteData(' ');
        WriteData(' ');
        WriteData(' ');
        WriteData('c');
        WriteData('m');

        WriteAddress(0x49);
        WriteData(digit[qian]);
        WriteData(digit[bai]);
        WriteData(digit[shi]);
        WriteData('.');
        WriteData(digit[ge]);
    }
}
```

```
    }

    j++;
}
}
//*****
//定时器 0 中断, 用做显示
timer0() interrupt 1 // 定时器 0 中断是 1 号
{
    // TH0=0xfd; //写入定时器 0 初始值
    // TL0=0x77;

}

//显示数据转换程序
void conversion(uint temp_data)
{
    uchar ge_data, shi_data, bai_data, qian_data ;
    qian_data=temp_data/1000;
    temp_data=temp_data%1000;
    bai_data=temp_data/100 ;
    temp_data=temp_data%100;    //取余运算
    shi_data=temp_data/10 ;
    temp_data=temp_data%10;    //取余运算
    ge_data=temp_data;
    EA=0;
    qian=qian_data;
    bai = bai_data;
    shi = shi_data;
    ge = ge_data ;
    if(qian||bai||shi||ge!=0)
    BEEP=qian||bai;
    EA=1;
}
//*****
void delay_20us()
{
    uchar bt ;
    for(bt=0;bt<60;bt++);
}

void pai_xu()
{
    uint t;
    if (distance[0]>distance[1])
        {t=distance[0];distance[0]=distance[1];distance[1]=t;}
    if(distance[0]>distance[2])
        {t=distance[2];distance[2]=distance[0];distance[0]=t;}
    if(distance[1]>distance[2])
        {t=distance[1];distance[1]=distance[2];distance[2]=t;}
}
```