基于超声波车辆运动探测系统

摘 要

随着经济的发展，汽车进入寻常百姓家，停车成了又一大难题。目前GPS的定位追踪发展十分发达但对于地下停车场则无法追踪为了能够对地下停车场车辆进行定位我们提出了基于超声波的车辆运动探测设计。利用超声波的测距原理当有车辆经过时进过一系列的判断对车辆进行定位，当车辆停放在停车位时能够实时将信息发送到PC端管理员进行实时监控引导，从而有效的管理地下停车场。

**关键字：** 超声波 车辆运动探测 地下停车场

**abstracted**

With the development of the economy, the car into the homes of ordinary people, parking has become another major problem. Currently GPS positioning and tracking development is very developed but for the underground parking is not able to track in order to be able to locate the vehicle in the underground parking lot we put forward the design of vehicle motion detection based on ultrasonic. The use of ultrasonic ranging principle when a vehicle passes into a series of judgment to position the vehicle. When the vehicle is parked in the parking lot can real-time will information sent to the PC side administrator for real-time monitoring guidance, and effective management of underground parking lot.

**Key words：**  ultrasonic vehicle motion detection underground parking lot

# 前 言

随着社会经济的发展，人们生活水平的提高，汽车已越来越多的进入家庭，汽车消费时代已悄然来临，停车场的建设也是方兴未艾，停车场智能管理系统也已经在大部分停车场发挥着重要作用，在为人们停车带来方便的同时也具有良好的社会效益和经济效益。我国汽车拥有量急剧增加。停车场作为交通设施的组成部分，随着交通运输的繁忙和不断发展，人们对其管理的要求也不断提高，都希望管理能够达到方便、快捷以及安全的效果。停车场的规模各不相同，对其进行管理的模式也有不同之处，管理者需要根据自身的条件，选择应用经济、稳定的管理程序，以免选择了高成本的管理系统。本设计旨在设计一个简洁、稳定、实用的停车场管理信息系统，希望在容错性、实用性、易操作性等方面具有自己的特色，并且保持一定的可扩展性，以满足不同停车场的信息管理需求。停车场管理系统是利用高度自动化的机电设备对停车场进行安全、有效的管理。由于尽量减少人工的参与，从而最大限度的减少人员费用和人为失误造成的损失，大大提高整个停车场的安全性与使用效率。为了更好地解决这样的问题，我设计了关于地下停车场的车辆运动探测的方案，有效的引导车主快速高效的停车，实现停车场的智能化管理。

目录

[前 言 1](#_Toc447628963)

[第1章 绪 论 2](#_Toc447628964)

[1.1基本原理 3](#_Toc447628965)

[1.1.1测距的基本原理 3](#_Toc447628966)

[1.2 系统方案 3](#_Toc447628967)

[1.2.1系统功能 3](#_Toc447628968)

[1.2.2系统方案 3](#_Toc447628969)

[1.3调试方案 12](#_Toc447628970)

[1.4 论文章节安排 12](#_Toc447628971)

[第2章 硬件的设计 13](#_Toc447628972)

[2.1系统硬件的功能 13](#_Toc447628973)

[2.2 硬件总体设计 14](#_Toc447628974)

[2.3硬件详细设计 14](#_Toc447628975)

[2.3.1 单片机最小系统设计 15](#_Toc447628976)

[2.3.2超声波模块电路 17](#_Toc447628977)

[2.3.3 LCD1602与单片机接口电路 17](#_Toc447628978)

[2.4 系统整体原理图设计 18](#_Toc447628979)

[第3章 软件设计 19](#_Toc447628980)

[3.1 软件的功能 19](#_Toc447628981)

[3.2 软件总体设计 20](#_Toc447628982)

[3.3软件详细设计 20](#_Toc447628983)

[3.3.1主程序设计 20](#_Toc447628984)

[3.3.2超声波测距子程序设计 21](#_Toc447628985)

[3.3.3显示子程序设计 22](#_Toc447628986)

[3.4 程序代码 24](#_Toc447628987)

[第4章 调 试 25](#_Toc447628988)

[4.1调试目的 26](#_Toc447628989)

[4.1.1验证超声波测距功能 26](#_Toc447628990)

[4.1.2验证显示功能 26](#_Toc447628991)

[4.1.3整体调试 26](#_Toc447628992)

[4.2调试方案 26](#_Toc447628993)

[4.2.1验证单个超声波测距模块的工作状态 26](#_Toc447628994)

[4.2.2整体调试 27](#_Toc447628995)

[4.3调试过程与结果 27](#_Toc447628996)

[4.3.1超声波测距调试 27](#_Toc447628997)

[4.3.2整个系统的调试 28](#_Toc447628998)

[4.4 结果与分析 28](#_Toc447628999)

[第5章 结论与展望 29](#_Toc447629000)

[5.1结论 29](#_Toc447629001)

[5.2展望 29](#_Toc447629002)

[参考文献 31](#_Toc447629003)

[致 谢 32](#_Toc447629004)

[附录 33](#_Toc447629005)

[附录一最小系统电路图 33](#_Toc447629006)

[附录二超声波测距原理图 34](#_Toc447629007)

[附录三主程序 35](#_Toc447629008)

[附录四实物图 46](#_Toc447629009)

# 第1章 绪 论

本章主要介绍了该设计基本原理，整体方案的设计，然后细化到每个模块的方案选择，最后根据设计方案进行了整个论文的章节安排。

## 1.1基本原理

### 1.1.1测距的基本原理

目前测距的方式大致分为激光测距、红外测距和超声波测距三大类，不管是哪种测试方式和测试仪器，其基本原理都是通过控制传感器发射端发射信号，然后信号通过被测物体反射，接收端接收反射回来的信号（如图1-1）。然后计算出从发射到接收所用时间，再根据该信号的传播速度就可以计算出物体和到发射接收端之间的距离。

发

收

被

测

物体

图1-1 测距原理图

## 1.2 系统方案

### 1.2.1系统功能

### 1.2.2系统方案

#### 1.2.2.1系统构架

显示输出

处理器

灯光提示

距离测量

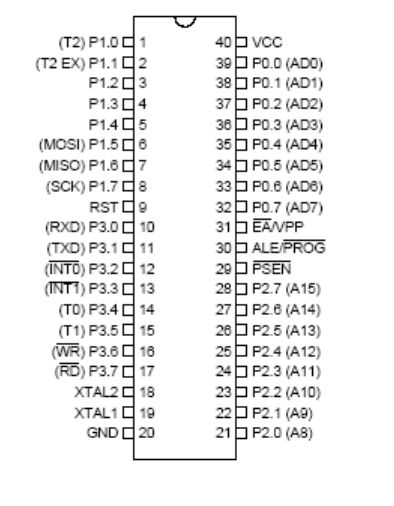
实时PC显示

图1-2 系统框图

#### 1.2.2.2器件的选择

1、处理器的选择

STC89C52单片机是宏晶科技推出的新一代高速/低功耗/超强干扰的高性能CMOS8位微控制器，指令代码完全兼容传统8051单片机，12时钟/机器周期和6时钟/机器周期可以任意选择。具有 8K 在系统可编程Flash 存储器。在单芯片上，拥有灵巧的8 位CPU 和在系统可编程Flash，使得STC89C52为众多嵌入式控制应用系统提供高灵活、超有效的解决方案。 它具有以下标准功能： 8k字节Flash，512字节RAM， 32 位I/O 口线，[看门狗定时器](http://baike.baidu.com/view/1313309.htm)，内置4KB EEPROM，MAX810复位电路，三个16 位 [定时器](http://baike.baidu.com/view/281961.htm)/计数器，一个6向量2级中断结构，全双工串行口。另外 STC89C52 可降至0Hz 静态逻辑操作，支持2种[软件](http://baike.baidu.com/view/37.htm)可选择节电模式。空闲模式下，CPU 停止工作，允许RAM、定时器/计数器、串口、中断继续工作。掉电保护方式下，RAM内容被保存，振荡器被冻结，单片机一切工作停止，直到下一个中断或硬件复位为止。最高运作频率35MHz，6T/12T可选。而且STC89C52价格低廉，适合我们设计的初衷，最为关键的一点，STC89C52单片机的程序下载器便宜实惠，而且网上关于STC89C52的教程也很多，对于我们初学者而言很容易上手，鉴于STC89C52以上的特点，我们的控制器件最终选择STC89C52.



(a)STC89C52RC实物图 (b)STC89C52RC引脚图

图1-3 STC89C52RC单片机

2、测距方案的选择

目前测距的方式有激光测距、红外线测距、超声波测距三大类，但是各有特点，我们要根据我们设计的需要来权衡这三类测距方式的利与弊，最终选择一个合适、有效的测距方式。

方案一：激光测距

由于激光在亮度、方向性、单色性和相干性等方面有独特之处，它就被广泛应用在工业生产、国防军事、科研机构、防盗安全等各个行业各个领域。

在测距领域，激光的作用更是不容忽视的，甚至可以说，激光测距是激光应用最早的领域，例如：世界上第一台激光器，是由[美国](http://baike.baidu.com/view/2398.htm)休斯飞机公司的科学家[梅曼](http://baike.baidu.com/view/451011.htm)于1960年，首先研制成功的，而在1962年就被应用于地球与月球之间距离的测量。测量精度高、分辨率高、抗干扰能力强、体积小同时重量轻的激光测距仪受到了大多数具有测距需求的企业、机构和个人的青睐，其市场需求空间大，应用领域广阔，并且起着日益重要的作用[5]。

激光测距仪一般采用两种方式来测量距离：脉冲法和相位法。脉冲法测距的过程是测距仪发射出的激光经被测量物体的反射后又被测距仪接收，测距仪同时记录激光往返的时间。光速和往返时间的乘积的一半，就是测距仪和被测量物体之间的距离。脉冲法测量距离的精度一般是在+/- 1米左右。另外，此类测距仪的测量盲区一般是15米左右。

激光测距是光波测距中的一种测距方式，如果光以速度c在空气中传播在A、B两点间往返一次所需时间为t，则A、B两点间距离D可用下列表示。

D=ct/2 （1-1）

式中：

D——测站点A、B两点间距离；

c——光在大气中传播的速度；

t——光往返A、B一次所需的时间。

由式（1-1）可知，要测量A、B距离实际上是要测量光传播的时间t，根据测量时间方法的不同，因此激光测距仪通常可分为脉冲式和相位式两种测量形式。

但是激光测距仪的成本过高，而且用于人体和电脑之间测距过于奢侈，还有一个重要原因，激光对人眼有一定的伤害，故不可取。

方案二：红外线测距

红外线测距是利用红外光来传送控制指令信号，因此，作为红外测距中的红外光发射器件的红外发光二极管和红外光接收器件的红外光敏管，是构成红外测距系统的基本器件。

红外线发射器件是最长用的为红外发光二极管，它与普通发光二极管的结构 原理以及制作工艺基本相同，是只有一个PN结的半导体器件，只是所有的材料不同，制造红外发光二极管砷化钾，砷铝钾等，其中应用最多的是砷化钾。

红外发光二极管一般采用环氧树脂，玻璃，塑料等封装，除白色透明材料封装外，还可见到用蓝色透明材料封装的。红外发光二极管按发光功率的大小，可分为小功率，中功率，大功率三种。另外，红外发光二极管除顶面发光型外，还有侧面发光型。小功率管一般采用全塑封装，也有部分是采用陶瓷底座，顶端用玻璃或环氧树脂透镜封装的，中大功率管一般采用带螺纹金属底座，以便安装散热片。随着发光功率得提高，相应体积的管子也增大。

影响红外发光二极管工作状态的主要参数为：正向工作电流I、光功率P、峰值波长、反向漏电流、相应时间t0。其中在测距方面应用时，对红外发射管工作状态影响最大的是：正向工作电流I、光功率P、峰值波长。正向电流的大小直接影响管子能否正常工作，只有正常管子正常工作时才能发出红外光；光功率是指输入到发光二极管的电功率转化为光输出功率的那一部分。光功率越大，发射距离越远，因此光功率会影响到红外测距的测距长度。还有峰值波长指红外发光管所发出近红外光中，光强最大值所对应的发光波长，在选用红外接收管时要注意，其峰值波长应尽量靠近红外发射管的峰值波长。



我们知道半导体具有光电效应，即用光照半导体，可使半导体的电阻率发生变化。利用半导体的光电效应可以制成光电二极管，不同的半导体材料对不同波长的入射光的响应是不同的。

光敏二极管有顶面受光和侧面受光两种形式。它也是采用塑料、玻璃、环氧树脂等材料封装。

其中影响光敏二极管工作状态的主要参数为：光电流IL、暗电流ID、反向工作电压UR、峰值波长。在测距方面应用时，对红外发射管工作状态影响最大的是：光电流IL和峰值波长，光电流指在一定反向电压下，入射光强为某一定值时流过管子的电流。光敏二极管的光电流一般为几十μA，并与入射光强成正比。所以可以利用这个原理把光敏二极管作为红外接收管，当接收到一定的红外光时，电流会发生变化，可以作为系统的响应信号。峰值波长指光敏二极管光谱响应最灵敏的波长范围，一般为0.88μM-0.94μM。这个是接收管能接收到红外信号的关键，只有光敏二极管的峰值波长和红外发射管的峰值波长匹配时，光敏二极管才能很好的接收到红外光。



红外传感器的测距基本原理为红外发射电路的红外发光管发出红外光，经障碍物反射后，由红外接收电路的光敏接收管接收前方物体反射光，据此判断前方是否有障碍物。根据发射光的强弱可以判断物体的距离，由于接收管接收的光强随是随反射物体的距离变化而变化的。因而，距离近则反射光强，距离远则反射光弱。

因为红外线是介于可见光和微波之间的一种电磁波，因此，它不仅具有可见光直线传播、反射、折射等特性，还具有微波的某些特性，如较强的穿透能力和能贯穿某些不透明物质等。红外传感器包括红外发射器件和红外接收器件。自然界的所有物体只要温度高于绝对零度都会辐射红外线，因而，红外传感器须具有更强的发射和接收能力。

由以上红外测距的发射器件、接收器件的参数和其测距基本原理得知，其测距方式不适合本设计。

方案三：超声波测距

超声波传感器是利用压电效应的原理，压电效应有逆效应和顺效应，超声波传感器是可逆元件，超声波发送器就是利用压电逆效应的原理。所谓压电逆效应，是在压电元件上施加电压，元件就发生形变，既逆变。已极化的压电陶瓷上施加极性的电压，外部⊕电荷与压电陶瓷的极化⊕正电荷相斥，同时外部⊖负电荷与极化⊖负电荷相斥。由于相斥的作用，压电陶瓷在厚度方向上缩短，在长度方向上伸长。若外部施加电压的极性变反，电压陶瓷在厚度方向上伸长，在长度方向上缩短。超声波测距模块上就有两个这样的传感器，一个作为发射，一个作为接受。

如果需要测距，则测距模块电路控制超声波发射装置向某一个方向发出超声波并且开始计时，超声波在空气中的传播速度为，当超声波传播到被测物体时就会立即被反射，然后通过超声波接收装置接收，接收器接到超声波时暂停计时，根据时间差t就可以知道距离了，这与雷达测距原理相似。



超声波方向性强，在介质中传播的距离较远，因而超声波经常用于距离的测量，如测距仪和物位测量仪等都可以通过超声波来实现。利用超声波检测往往比较迅速、方便、计算简单、易于做到实时控制，并且在测量精度方面能达到工业实用的要求，因此在移动机器人的研制上也得到了广泛的应用[6-13]。

例如HR-SR04超声波测距模块（如图1-5），它可以提供2cm-400cm的非接触距离感测功能，精度可以达到3mm，模块包括超声波发射器、接收器、控制电路和接口电路。此模块接口有4个，分别为VCC、GND、Trig、Echo。VCC和GND给模块提供电源和地，那么Trig和Echo则分别和单片机I/O口相连。其工作方式为触发测距，和Trig相连的I/O口发送一个至少10us的高电平信号，模块内部自动发送8个40KHz的方波，自动检测是否有信号返回，若有信号返回，通过和I/O口相连的Echo输出一个高电平，高电平持续的时间就是超声波从发射到返回的时间，（如图1-4）。

测试距离=（高电平时间\*声速（340m/s））/2 （1-3）

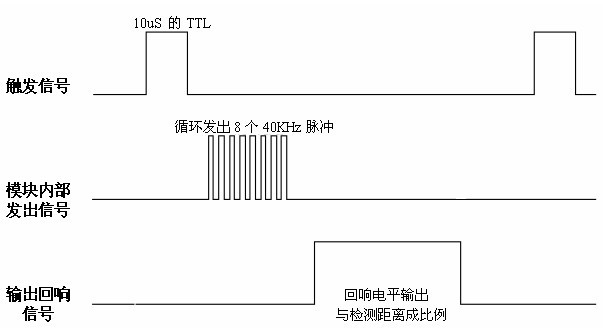


图1-4 HR-SR04超声波模块测距原理

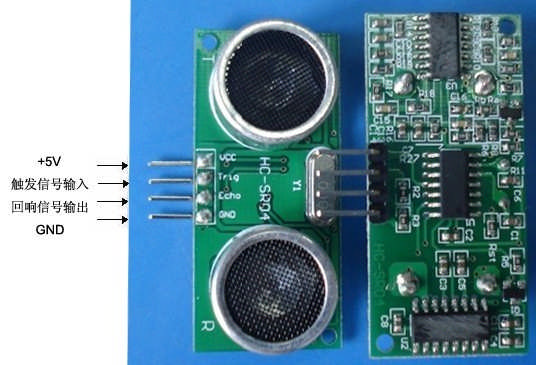


图1-5 HR-SR04超声波模块实物图

3、报警方式的选择

通常我们所见的报警方式为声和光报警，例如警车上有警灯和喇叭，当遇到紧急状况时就会同时工作，以表警示。

方案一：光电报警

光电报警常用的光电报警器件为LED发光二级管，稍微大一点的为警灯闪烁，通过延时开断这些光电器件让其达到闪烁的功能，便有警示、提示功能，即构成光电报警装置。



（a）LED发光二极管 （b）警灯

图1-6光电报警器件

但是我们的设计主要是在人们使用电脑工作的，人们的目光主要在电脑显示器上，如果使用光电报警则会影响人们正常使用电脑；还有我们的报警方式有很多种，如果使用光电报警，则需要用不同光线颜色区分报警方式，所以要使用不同的光电器件，这样会增大面板面积、增加成本，有悖于我们的设计初衷，所以此方案不可取。

方案二：语音报警

语音报警主要是通过声音信号去提示和警告人的报警方式，其原理首先通过麦克风采集语音信号，由于语音信号为模拟信号，所以要用A/D芯片将其进行模数转换，转换成便于存储和处理的二进制数字信号，然后存储在计算机的存储器里面，如果报警需要则通过指令，调出相应的报警语音文件，然后通过D/A处理，还原成原来的模拟信号，然后再用功率放大器放大D/A输出的信号，再通过扬声器就可以听到报警声音，达到报警效果。

但是由于我们的设计初衷是简单和实用，因此这样的语音报警过于复杂，还有就是成本过高，不适合本设计，故不能选择。

方案三：蜂鸣器报警

蜂鸣器是一种一体化结构的电子讯响器，采用的是直流电压供电，被广泛的应用于计算机、打印机、报警器、电子玩具、汽车电子设备、电话机、定时器等电子产品中作为发声器件。蜂鸣器主要分为压电式蜂鸣器和电磁式蜂鸣器两种类型，我们一般使用压电式蜂鸣器。压电式蜂鸣器主要由多谐振荡器、压电蜂鸣片、阻抗匹配器及共鸣箱、外壳等组成。多谐振荡器由晶体管或集成电路构成。当接通电源后，多谐振荡器起振，输出1.5-2.5kHz的音频信号，阻抗匹配器推动压电蜂鸣片发声。蜂鸣器发出的声音可以通过控制它的I/O口改变频率而改变。最主要的是这种蜂鸣器驱动电路简单，有的通过一个三极管直接驱动，有的通过驱动芯片驱动，例如ULN2003。而且这种蜂鸣器的体积小、价格低，便于我们使用其作为本设计的报警器。

故选择蜂鸣器作为本设计的报警器。

6、显示方式的选择

LED数码管是我们常见的显示器件，主要用于数字的显示，但是我们本设计需要显示的数据比较多，比如设置提示、测距数据、温度数据、时间数据，因此数码管不能够满足显示要求。依据设计的显示要求，那么我们常见的液晶显示器LCD1602比较适合，它通过字符显示，既能显示数字又能显示字符，而且接法简单，因此选择LCD1602作为显示器件（如图1-7）。

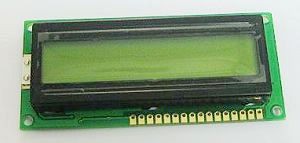


图1-7 LCD1602实物图

#### 1.2.2.3相关软件工具

1、单片机开发软件Keil C51简介

单片机开发中除必要的硬件外，同样离不开软件，我们要把实现的功能通过编程来在硬件上实现，因此我们需要一款适合我们的单片机开发软件。用于MCS-51单片机的汇编软件有早期的A51，随着单片机开发技术的不断发展，从普遍使用汇编语言到逐渐使用高级语言开发，单片机的开发软件也在不断发展，Keil软件是目前最流行开发MCS-51系列单片机的软件。Keil提供了包括C编译器、宏汇编、连接器、库管理和一个功能强大的仿真调试器等在内的完整开发方案，通过一个集成开发环境（uVision）将这些部份组合在一起。掌握这一软件的使用对于单片机的开发来说是十分必要的。由于现在C语言编程简单易懂，我们也采用C语言编程，因此Keil就是我们的首选,即使不使用C语言而仅用汇编语言编程，其方便易用的集成环境、强大的软件仿真调试工具也会使开发工作事半功倍。

2、电路设计软件Altium Designer Summer 09简介

Altium Designer Summer 09是Altium公司新出的一款EDA开发工具。Altium Designer Summer 09的发布延续了连续不断的新技术的应用过程。这必将帮助用户更轻松地创建下一代电子设计。同时，我们将令Altium Designer更符合电子设计师的要求。Altium的一体化设计结构将硬件、软件和变成硬件集合在一个单一的环境中，这将使用户自由地探索新的设计构想。在整个设计构成中，每个人都使用同一个设计界面。Summer 09版本解决了大量历史遗留问题。其中就包括了增加更多的机械层设置、增强原理图网络类定义。可以再Windows 7上运行，增加了很多类似于Windows的快捷键，大大缩短了绘图周期。对于初学者而言更加容易上手。

## 1.3调试方案

对于硬件而言，调试工作主要使用示波器和万用表去测量相应的引脚或者位置的波形或者电压，来验证硬件工作状态是否正常；对于软件主要是下载在硬件上面，让其运行，来验证是否达到预期效果。

## 1.4 论文章节安排

本论文共有五章。第一章是绪论，主要阐述了本设计中的基本原理和系统总体方案设计。第二章是硬件设计，详细阐述了硬件总体功能、超声波测距模块、单片机最小系统的硬件设计。第三章是软件设计，详细阐述了测距、显示、通信等功能的软件设计。第四章是调试与分析，按照调试方案，验证各个模块的功能，并对调试过程中产生的问题进行了分析与总结。第五章是系统设计的结论与展望，在结论里对整个系统的设计结果作了简单的总结，展望则根据系统工作中存在的问题提出了一些相应的改进方法。

# 第2章 硬件的设计

本章主要从硬件的功能出发，对整个系统的硬件做出详细的设计，主要包括：单片机最小系统的设计、超声波模块电路设计、通讯电路设计、光电提示电路设计和LCD1602与单片机接口电路设计。

## 2.1系统硬件的功能

根据第一章的确定的方案，我们首先要进行整个系统的硬件设计。硬件的设计则主要根据硬件所要达到的功能而进行，以设计方案提出的功能确定硬件所要实现的功能。

1、单片机最小系统的功能

单片机最小系统为整个系统提供控制功能，也是后面软件运行的载体，是整个系统的大脑。

2、超声波模块电路的功能

超声波测距电路也是整个系统的关键，它为测量人与电脑之间的距离提供了硬件支撑。由于这是个模块化电路，那么主要设计其和单片机的接口电路，保证供电外还要和单片机I/O口进行连接，以便控制。

2、LCD1602与单片机接口电路功能

在本系统中还有个不可或缺的部分就是显示电路部分，由于LCD1602液晶显示模块是集成的电路只需要设计接口电路即可工作，那么其主要功能就是保证该接口能给LCD显示模块供电，可以通过单片机和其连接保证数据读写。

## 2.2 硬件总体设计

本设计的硬件部分主要由：单片机最小系统、超声波测距电路、光电显示电路、LCD1602显示部分组成（如图2-1）。

其中单片机最小系统的设计和超声波测距的设计最为关键，他们的硬件设计之间关系到本设计的成败，因此硬件的设计主要先从单片机最小系统电路入手，由于超生测距模块是焊接好的，只需要在单片机上设计相应的接口即可。

如果测试单片机最小系统工作正常，接下来再加上要求的光电显示电路、串口通信电路和LCD1602显示接口。最后完成整个系统硬件的设计。

LCD1602显示接口电路

单片机最小系统

报警电路

超声波测距电路

实时PC显示

图2-1硬件总体框图

## 2.3硬件详细设计

### 2.3.1 单片机最小系统设计

#### 2.3.1.2时钟电路

和AT89C51类似，STC89C52的振荡器也有两种组成方式：片内振荡器和片外振荡器。本系统采用片内振荡器。单片机芯片引脚XTAL1、XTAL2内部有一个反相器，在这两脚之间接上一个谐振器和两个电容，就形成一个振荡器，振荡频率取决于外接谐振器的固有频率。单片机晶体谐振器可用石英晶体和陶瓷两种，常用石英晶体谐振器，并因此把振荡频率称为晶振频率。片内振荡器的组成电路如图2-1所示。

STC89C52单片机晶振频率可从0Hz～80MHz。不过，单片机晶振频率的数值通常选择为=6MHz，11.0592MHz，12MHz等三种，或者选=1MHz。晶振频率越高，电源电流越大，功率消耗越大。应当在保证控制器性能的前提下选择尽可能低的晶振频率。对石英晶体，电容C1，C2=30pF±10pF，典型值33pF；对陶瓷谐振器，电容C1，C2=40pF±10pF，典型值39pF或47pF。



本设计选用石英晶体振荡器，=12MHz，C1，C2=30pF。



图2-2片内振荡器时钟电路

#### 2.3.1.3复位电路

单片机的复位包括初始化和从头开始工作这样连续的两步。单片机复位引脚Reset接收高电平进行初始化；接收低电平，开始工作。也就是说单片机接收正脉冲开始复位，在正脉冲的下降沿启动单片机。单片机正常工作期间，复位引脚Reset需要一直保持低电平。工作过程中引脚Reset一旦接收到一个正脉冲，就会再次进行复位启动。

为可靠完成复位，单片机要求Reset引脚施加的正脉冲脉宽不小于2个机器周期（2µs）。设计复位电路的要求就是确定电阻电容值，使其时间常数达到2个机器周期的复位最小正脉宽要求。本设计中晶振频率fosc=12MHz时，机器周期T=1µs，要求加在Reset引脚的正脉宽不小于2µs。

当单片机上电后，因为电容两端的电压不能突变就会使RST端瞬间产生一个大约为+5V的电压，而CMOS单片机最小输入高电平电压Umin=3.5V，它瞬间产生的电压是大于3.5V的，因此RST接收高电平进行初始化。此后+5V对电容C充电导致RST端电压迅速下降使它变为低电平，单片机开始工作。根据上述要求，本设计R23=10K，C值取10µF。

系统上电运行后需要复位，一般都是通过手动复位即带电复位来完成的。根据以上考虑，本设计选用手动复位和上电复位相结合的方式，如图2-3所示。



图2-3复位电路

#### 2.3.1.4电源电路

USB简单易懂与自制的变压器产生的5V供电系统相比，USB供电电压为5V更加安全，最重要的是制作过程要比稳压电源5V供电简单的多，它所提供的功率不超过2.25W，最大输出电流为500mA，电压一般为，这里的偏差可以通过接入旁路电容来消除。本系统供电要求符合以上特点，尤其符合本系统简单实用的特点，即插即用。只需要从USB接口中接出VCC和GND即可给系统供电，注意在电路上接入两个电容为旁路电容，作用是稳定电压，通过一个自锁开关可以实现系统开关自如，LED是电源指示灯，1K电阻是限流的作用[14-19]。



图2-4电源电路

### 2.3.2超声波模块电路

超声波测距电路是一个模块化的电路，它有4个接口，分别为VCC、GND、Trig、Echo。VCC和GND给模块供电，那么Trig和Echo则分别和单片机I/O口相连。其工作方式为触发测距，和Trig相连的I/O口发送一个至少10us的高电平信号，模块内部自动发送8个40KHz的方波，自动检测是否有信号返回，若有信号返回，通过和I/O口相连的Echo输出一个高电平，高电平持续的时间就是超声波从发射到返回的时间。

测试距离=（高电平时间\*声速（340m/s））/2。 （2-1）

详图见附录。

### 2.3.3 LCD1602与单片机接口电路

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有若干个或者等点阵字符位组成，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有间隔起到了字符间距和行距的作用，正因为如此所以它不能显示图形。



1602采用标准的16脚接口，接口简单，即插即用，端口与单片机直接相连接，1脚为电源地，2脚为电源正极接5V，3脚可通过电位器VR可以调节它的显示字符的清晰度，4脚为RS数据命令选择，5脚为R/W读写选择，6脚为E使能信号端7-14为数据端，15背光源正极，16背光源负极（如图2-5）。



图2-5 LCD1602接口电路

## 2.4 系统整体原理图设计

整个系统的原理图采用Altium Designer Summer 09软件绘制，由于本设计主要由超声波测距模块和单片机最小系统组成，但是超声波模块需要安装在电脑显示器上面，所以不在一个电路板上，他们之间用导线连接。超声波测距模块原理图和单片机最小系统原理图分别绘制在两张图纸上面[20-21]。

详图见附录。

# 第3章 软件设计

本章主要讲解软件的详细设计，主要针对系统的功能提出了软件要实现的功能，然后给出了软件总体设计方案，接下来对各个模块的软件进行了详细的设计，并给出了程序流程图。

## 3.1 软件的功能

没有纯粹的硬件，也没有纯粹的软件，所以电脑近距离报警器设计的关键也在于软件设计。通过编写C51程序来将所有硬件整合起来，使其发挥作用，最后实现我们所要的功能。软件的编写主要从单片机初始化、超声波测距、与上位机通信、显示等各个模块出发，先编写各个模块，再整合在一起，发挥整体功能。

1、超声波波测距子程序

由于我们的设计主要是确定人的位置，那么首先得实现的功能是测距功能，那么接下来根据测量的距离可以判断该位置是否有车辆进过。

2、灯光提示子程序

通过灯光提示子程序就可以人为的去设置适合自己的灯光提示距离参数。

3、显示子程序

这里其实是和前面的功能相对应的，每一步操作都有相应的显示，显示实时距离的。

4、串口通信子程序

通过串口通信子程序将超声波采集处理后的信息实时发送到PC端，每个超声波当前检测的状态。

## 3.2 软件总体设计

软件的设计在整个系统的设计中至关重要，该系统的软件主要由主程序统领下面的超声波测距子程序、RS485通信子程序、和显示子程序。超声波采集的数据通过RS485与PC机进行通信数据传输并将显示在LCD显示屏上。

主程序

初始化

超声波测距子程序

灯光提示子程序

显示子程序

报警程序

串口通信子程序

图3-1 软件总体框图

## 3.3软件详细设计

### 3.3.1主程序设计

主程序统领各个子程序的工作，是程序设计的核心部分，也是难度比较大的一部分，所以必须根据设计要求仔细推敲，保证设计思路的正确，为后面的子程序设计做好铺垫，要在时序上安排好每个子程序的工作顺序，保证各个子程序能够更好的工作。

主程序开始

初始化显示

初始化定时器、初始化中断

超声波初始化

超声波测距并计算子程序

串口通信子函数

灯光提示

液晶显示

图3-2主程序流程图

由以上流程图可以看出我们整个系统的工作流程，先进入主程序，液晶初始化显示，定时器初始化，然后进行超声波测距初始化，就会立即进入测距阶段，再运行串口通信子程序，后进入灯光提示程序，再将测量的距离显示在LCD，主程序整个流程结束。依次循环执行。

### 3.3.2超声波测距子程序设计

根据前面讲到的超声波测距原理，只要能计算出超声波发出然后遇到被测物体返回等到接收的这段时间，就可以得到被测物体与超声波测距模块之间的距离，因此超声波测距子程序的设计要从此入手，在启动发射电路的同时启动单片机内部的定时器T0，利用定时器的计数功能记录超声波发射的时间和接收到反射波的时间。当收到超声波反射波时，接收电路输出端产生高电平，在INT0或INT1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[24-26]。

超声波测距子程序开始

I/O输出至少10us的高电平信号

发出超声波定时器计时

等超声波返回时停止计时

读取定时器值计算距离

没有超出

超出范围

距离判断

输出fault

输出距离

超声波测距子程序结束

图3-3超声波测距子程序流程图

### 3.3.3显示子程序设计

显示子程序也是基于以上子程序而设计，本设计的显示采用的是LCD1602，因此它的程序主要在于通过I/O口控制好它的读、写入口，然后再确定好其地址就可以了。

LCD命令函数如下：

void wait()

{

P0 = 0xFF;

do

{

RS = 0; //数据命令选择

RW = 1; //读写选择

EN = 0; //使能信号

EN = 1;

}

while(BUSY == 1);

EN = 0;

}

LCD写地址函数如下：

void write\_cmd(uchar com)

{

wait();

P0 = com; //数据端口

EN = 0; //使能信号

RS = 0; //数据命令选择

RW = 0; //读写选择

EN = 1;

EN = 0;

}

LCD写数据函数如下：

void write\_data(uchar data0)

{

wait();

P0 = data0; //数据端口

EN = 0; //使能信号

RS = 1; //数据命令选择

RW = 0; //读写选择

EN = 1;

EN = 0;

}

#### 3.3.4显示测量距离

在LCD1602屏幕上实时动态显示五个超声波测得的距离。

显示子程序开始

LCD初始化

Y

LCD是否忙

N

单片机向LCD写命令

单片机向LCD写数据

显示数据

显示子程序结束

图3-4显示子程序流程图

## 3.4 程序代码

本设计的程序代码是在Keil软件环境下编写的，其中包括主程序、超声波测距子程序、RS485通信子程序、光电提示子程序、显示子程序。

具体程序代码见附录。

# 第4章 调 试

本章主要介绍了整个系统的调试，先根据设计之初的要求和功能，明确调试目的，然后设计出调试方案，逐步测试验证各个模块的功能，再整体调试验证系统整体运行功能，最后根据调试中出现的问题进行改进优化。

## 4.1调试目的

通过调试来验证我们设计之初确定的功能，如果有问题再针对问题进行解决，要使系统的功能都能完全实现，并且经过调试使系统的工作达到最优化。

### 4.1.1验证超声波测距功能

地下停车场车辆运动探测其主要部分为测量距离，那么衡量的第一个标准就是距离，验证每个模块能测量的范围，接下来再研究其精度，还有一个关键环节就是验证五个超声波测距模块是否能够很好的同时工作，因为它关系到系统中确定车在地下停车场位置。

### 4.1.2验证显示功能

验证LCD1602显示功能是否正常，要能够动态显示超声波状态功能，进入运行阶段到能够正确各个超声波的状态从而判断是否有车辆。

### 4.1.3整体调试

通过整体调试，去验证整个系统的工作状态，看是否达到设计之初的设计要求，并且通过整体调试，使整个系统达到最优化，工作起来顺畅无阻。

## 4.2调试方案

### 4.2.1验证单个超声波测距模块的工作状态

首先编写一个单独的超声波测距子程序，采用一个超声波测距模块，测量结果让其在LCD上显示，看其测量的距离是否正确，测量范围和精度是否符合本设计的要求。如果单个的测距功能达到要求了，然后将多个个模块都接入到系统，编写相应的测距程序，测试多个个模块同时工作的性能。

### 4.2.2整体调试

将整个系统安装起来，将多个个超声波测距模块安装顶部位置，调整好角度，然后将线连接好，将整合好的完整程序下入单片机。测试当有车辆经过时判断是否正确灯光会有相应的提示。当车辆进入停车位是将该停车位的状态实时反应到PC端。进入最终测试如果以上各项指标正常则整个系统工作状态良好。

## 4.3调试过程与结果

### 4.3.1超声波测距调试

将单个超声波模块接入单片机最小系统，按照测试方案编写超声波测距测试程序，然后运行程序测试最远距离为600cm，测试最短距离为1cm 。

将多个个超声波测距模块全部接入单片机最小系统，按照测试方案编写三个超声波同时测距的程序，然后运行程序，可以分别显示出多个个模块所测试的距离。（如图4-1所示）。



图4-1三个超声波测距模块测试结果

### 4.3.2整个系统的调试

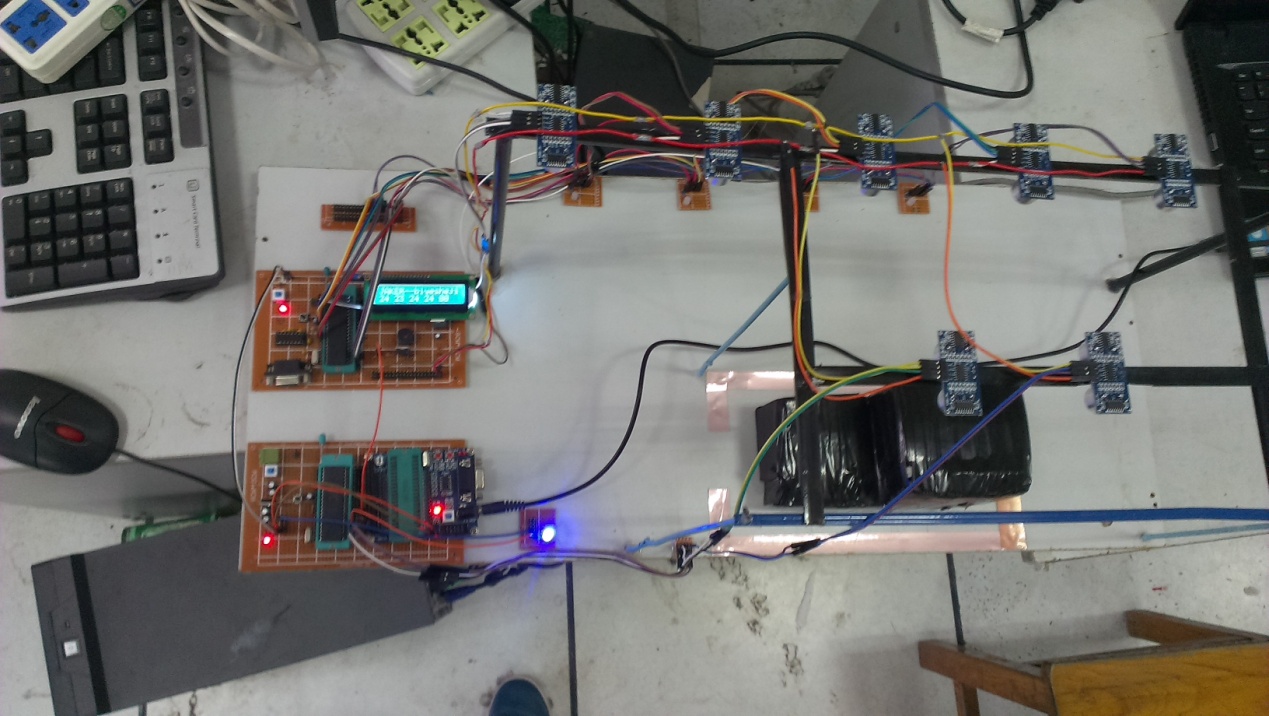


图4-3整个系统在工作时的状态

## 4.4 结果与分析

在系统的调试过程中，产生的问题多种多样。下面就对主要问题进行分析，对相应的解决方法作简单的阐述。

1、多个超声波测距模块不能同时工作。

分析：这个问题主要是由于软件问题引起的，故先要检查软件。

解决：最终发现是软件编写问题，没有注意到各个模块要分步工作，最终重新修改程序，让各个模块分步工作，然后读取每个模块测距的值，最终实现多个个模块同时工作。

2、超声波测距测量距离实时性不好。

分析：由于软件问题导致定时器值不对，或者触发信号问题。

解决：在检查过程中发现是因为触发信号过长引起，触发信号过长，再加上超声波分步工作，导致测量距离实时性不好，最后将触发信号减小，问题得到解决。

3、RS485通信如何实现控制发射顺序，通过上位机设置相应的通信地址，实现多个单片机一次发送超声波的数据。

# 第5章 结论与展望

本章主要对整个系统的设计过程做出了总结，得出了系统工作的最后结论，并且根据不足做出了展望。

## 5.1结论

在硬件设计上，完成了原理图的设计，并且根据设计的需求，设计了适合该系统的单片机最小系统，经过调试可以很好的工作。在软件上，针对设计初期所定的要求，实现了五个超声波同时测距，并且根据测量的距离进行计算得知该停车场位置的状态有相应的灯光提示，并通过RS485与PC机进行通信将采集的数据实时显示到PC端。经过测试以上所提到的功能都基本能实现。总之，本设计制作完成了设计的要求。

在这次毕业设计过程中，我对单片机这一领域有了进一步的认识，通过自己动手实践去做，去体会所学的知识，也加强了我的动手能力，而且也增强了如何从各个方面考虑问题的能力。通过这次毕业设计，使我对新事物的认识，了解及应用能力都有了明显的提高。通过这次毕业设计，我不但开阔了视野，而且拓宽了思维。

总之，这次毕业设计实践使我受益匪浅，认识到了自己有很多不完善之处：基础还不够扎实，运用知识的能力还不强，思考的深度不够，等等。所以在毕业后的工作当中，我还要继续学习，充实自己，并且注意在实际工作中积累经验。

## 5.2展望

本系统经测试基本完成了它的功能达到了预期的指标，但仍然存在着诸多缺陷与不足，需要对系统进行进一步开发与完善。这里列举了一些改进的方法，使系统更加的完善，并能使系统更有实用价值。

1、整个系统还比较零散，在用户使用时还不够方便，所以后期可以将整个系统的器件选择成贴片封装，然后PCB布线采用多层布线减少电路板面积，使整个系统体积减少，看起来美观，用起来方便。

2、本系统的设置、显示和报警功能还是在单片机最小系统上实现的，希望后期可以通过单片机和PC机之间的通信，将设置、显示和报警在计算机上通过软件来实现。

3、优化设计，采用是上位机进行数据处理，单片机作为下位机实行数据采集开发友好的人机交互界面，从实际意义上实现智能化的停车场监控、规划系统。

# 参考文献

[19] 郭天祥.51单片机C语言教程—入门﹑提高﹑开发﹑拓展[M],北京：电子工业出版社，2009：3-9

# 致 谢

首先，在此衷心感谢钱峰老师对我毕业设计的关心和支持。钱峰老师严谨的治学态度，科学的工作方法，对知识孜孜以求的态度，以及对学术前沿动态的敏锐洞察力，都给了我极大的影响。感谢多年来辛勤培养和教育过我的所有老师，感谢曾经给我无私帮助的所有朋友，感谢你们在生活和学习上给予我的很多帮助!

# 附录

附录一 最小系统电路图

附录二 超声波测距原理图

附录三 主程序

## 附录一最小系统电路图

## 附录二超声波测距原理图



## 附录三主程序

#include<reg52.h>

#include<intrins.h>

#include<math.h>

#define uchar unsigned char

#define uint unsigned int

#define AddressID 0x31

sbit RS485E=P3^7;

sbit MAX485\_DIR=P3^7;

sbit RX\_1=P1^2; //超声波接收

sbit TX\_1=P1^3; //超声波发射

sbit RX\_2=P1^4;

sbit TX\_2=P1^5;

sbit RX\_3=P1^6;

sbit TX\_3=P1^7;

sbit RX\_4=P3^2;

sbit TX\_4=P3^3;

sbit RX\_5=P1^0;

sbit TX\_5=P1^1;

sbit led0=P2^0;

sbit led1=P2^1;

sbit led2=P2^2;

sbit led3=P2^3;

sbit led4=P2^4;

uchar ch; //用于存放串口接收到的数据

bit read\_flag= 0 ; //取数标志位

unsigned long time=0; //超声波测的时间

unsigned long timer=0;

unsigned char posit=0;

unsigned long D\_1=0; //超声波测的距离

unsigned long D\_2=0;

unsigned long D\_3=0;

unsigned long D\_4=0;

unsigned long D\_5=0;

unsigned int flag =0;

unsigned char A='1'; //待显示字符。

unsigned char kong='\*';

unsigned char C='2'; //待显示字符。

unsigned char D='3'; //待显示字符。

unsigned char E='4'; //待显示字符。

void StartModule() ;

void send\_char( unsigned char ch1) ;

sbit lcd\_rs\_port = P3^4; /\*定义LCD控制端口\*/

sbit lcd\_rw\_port = P3^5;

sbit lcd\_en\_port = P3^6;

#define lcd\_data\_port P0

unsigned char code mcustudio[] ={"JAKER--biyesheji"};

unsigned char code ASCII[15] = {'0','1','2','3','4','5','6','7','8','9','C','-','M'};

unsigned char disbuff[8] ={ 0,0,0,0,};

///////////////////////////////////////

//---------------------------------------------

void lcd\_delay(uchar ms) /\*LCD1602 延时\*/

{

uchar j;

while(ms--){

for(j=0;j<250;j++)

{;}

}

}

void lcd\_busy\_wait() /\*LCD1602 忙等待\*/

{

lcd\_rs\_port = 0;

lcd\_rw\_port = 1;

lcd\_en\_port = 1;

lcd\_data\_port = 0xff;

while (lcd\_data\_port&0x80);

lcd\_en\_port = 0;

}

void lcd\_command\_write(uchar command) /\*LCD1602 命令字写入\*/

{

lcd\_busy\_wait();

lcd\_rs\_port = 0;

lcd\_rw\_port = 0;

lcd\_en\_port = 0;

lcd\_data\_port = command;

lcd\_en\_port = 1;

lcd\_en\_port = 0;

}

void lcd\_system\_reset() /\*LCD1602 初始化\*/

{

lcd\_delay(20);

lcd\_command\_write(0x38);

lcd\_delay(100);

lcd\_command\_write(0x38);

lcd\_delay(50);

lcd\_command\_write(0x38);

lcd\_delay(10);

lcd\_command\_write(0x08);

lcd\_command\_write(0x01);

lcd\_command\_write(0x06);

lcd\_command\_write(0x0c);

lcd\_data\_port = 0xff; /\*释放数据端口\*/

}

void lcd\_char\_write(uchar x\_pos,y\_pos,lcd\_dat) /\*LCD1602 字符写入\*/

{

x\_pos &= 0x0f; /\* X位置范围 0~15 \*/

y\_pos &= 0x01; /\* Y位置范围 0~ 1 \*/

if(y\_pos==1) x\_pos += 0x40;

x\_pos += 0x80;

lcd\_command\_write(x\_pos);

lcd\_busy\_wait();

lcd\_rs\_port = 1;

lcd\_rw\_port = 0;

lcd\_en\_port = 0;

lcd\_data\_port = lcd\_dat;

lcd\_en\_port = 1;

lcd\_en\_port = 0;

lcd\_data\_port = 0xff; /\*释放数据端口\*/

}

//按指定位置显示一串字符

void DisplayListChar(unsigned char X, unsigned char Y, unsigned char code \*DData)

{

unsigned char ListLength;

ListLength = 0;

Y &= 0x1;

X &= 0xF; //限制X不能大于15，Y不能大于1

while (DData[ListLength]>0x19) //若到达字串尾则退出

{

if (X <= 0xF) //X坐标应小于0xF

{

lcd\_char\_write(X, Y, DData[ListLength]); //显示单个字符

ListLength++;

X++;

}

}

}

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

void tance() //探测位置函数

{

disbuff[0]=D\_1%100/10;

disbuff[1]=D\_1%100%10;

disbuff[2]=D\_2%100/10;

disbuff[3]=D\_2%100%10;

disbuff[4]=D\_3%100/10;

disbuff[5]=D\_3%100%10;

disbuff[6]=D\_4%100/10;

disbuff[7]=D\_4%100%10;

disbuff[8]=D\_5%100/10;

disbuff[9]=D\_5%100%10;

lcd\_char\_write(0, 1, ASCII[disbuff[0]]);

lcd\_char\_write(1, 1, ASCII[disbuff[1]]);

lcd\_char\_write(3, 1, ASCII[disbuff[2]]);

lcd\_char\_write(4, 1, ASCII[disbuff[3]]);

lcd\_char\_write(6, 1, ASCII[disbuff[4]]);

lcd\_char\_write(7, 1, ASCII[disbuff[5]]);

lcd\_char\_write(9, 1, ASCII[disbuff[6]]);

lcd\_char\_write(10, 1, ASCII[disbuff[7]]);

lcd\_char\_write(12, 1, ASCII[disbuff[8]]);

lcd\_char\_write(13, 1, ASCII[disbuff[9]]);

if((D\_1>=700)||flag==1)//超出测量范围显示“-”

{

flag=0;

}

if((D\_2>=700)||flag==1)//超出测量范围显示“-”

{

flag=0;

}

if((D\_3>=700)||flag==1)//超出测量范围

{

flag=0;

}

if((D\_4>=700)||flag==1)//超出测量范围

{

flag=0;

}

if((D\_5>=700)||flag==1)//超出测量范围

{

flag=0;

}

if(D\_1<16&&D\_2<19)

{

led0=0;

send\_char(A);

lcd\_delay(2000);

}

else

{

led0=1;

send\_char(kong);

lcd\_delay(2000);

}

if(D\_2<16&&D\_3<19)

{

led1=0;

send\_char(C);

lcd\_delay(2000);

}

else

{

led1=1;

send\_char(kong);

lcd\_delay(2000);

}

if(D\_3<16&&D\_4<20)

{

led2=0;

send\_char(D);

lcd\_delay(2000);

}

else

{

led2=1;

send\_char(kong);

lcd\_delay(2000);

}

if(D\_4<=15&&D\_5<19)

{

led3=0;

send\_char(E);

lcd\_delay(2000);

}

else

{

led3=1;

send\_char(kong);

}

}

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

//初始化串口

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

void init\_com( void )

{

TCLK=1;

RCLK=1;

TH2=0xFF; //9600bps 11.0592MHz的时钟频率

TL2=0xDC;

RCAP2H=0xFF; //方式1和方式3的波特率＝fosc/(32-(65535-(RCAP2H,RCAP2L))

RCAP2L=0xDC;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*串口设置\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

SCON=0x50; //SM0 SM1 SM2 REN TB8 RB8 TI RI

PCON=0x00; //TMOD=0;

TR2=1;

}

//////////////////////////////////////////////////

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

\*\*\*\*\*\*\*\*485串口中断函数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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

void serial () interrupt 4 using 2

{

while(RI==0);

RI=0;

if(SBUF==AddressID)

{

while(RI==0);

RI=0;

}

if(SBUF==0x01)

{

MAX485\_DIR=1;//开发送

SBUF==AddressID;

while(TI==0);

TI=0;

send\_char;

}

}

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

////串口中断函数

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

//void serial () interrupt 4 using 3

//

//{

// if (RI)

// {

// RI = 0 ; //此步必不可少,因为硬件不能将其置0

// ch=SBUF; //从SBUF中取出数据

// read\_flag= 1 ; //就置位取数标志

// }

//

//}

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

//串口发送函数

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

void send\_char( unsigned char ch1)

{

SBUF=ch1; //将待发数据放到SBUF

while (TI== 0); //等待TI＝1（表示帧发送结束）发送

TI= 0 ; //此步必不可少,因为硬件不能将其置0

}

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

void main()

{

unsigned int i;

//////////////////////////

lcd\_system\_reset(); /\*LCD1602 初始化\*/

DisplayListChar(0, 0, mcustudio);

///////////////////////////////

init\_com() ;

TMOD=0x11;

TH0=0;

TL0=0;

TH1=0xf8;

TL1=0x30;

ET0=1;

ET1=1;

TR1=1;

EA=1;

while(1)

{

if(timer%100==0)

{

StartModule() ;

while(!RX\_1);

TR0=1;

while(RX\_1);

TR0=0;

time=TH0\*256+TL0;

TH0=0;

TL0=0;

D\_1=(time\*1.75)/100+2;

StartModule() ;

while(!RX\_2);

TR0=1;

while(RX\_2);

TR0=0;

time=TH0\*256+TL0;

TH0=0;

TL0=0;

D\_2=(time\*1.75)/100+1;

StartModule() ;

while(!RX\_3);

TR0=1;

while(RX\_3);

TR0=0;

time=TH0\*256+TL0;

TH0=0;

TL0=0;

D\_3=(time\*1.75)/100+1;

StartModule() ;

while(!RX\_4);

TR0=1;

while(RX\_4);

TR0=0;

time=TH0\*256+TL0;

TH0=0;

TL0=0;

D\_4=(time\*1.75)/100+1;

// StartModule() ;

//

// while(!RX\_5);

// TR0=1;

// while(RX\_5);

// TR0=0;

// time=TH0\*256+TL0;

// TH0=0;

// TL0=0;

// D\_5=(time\*1.75)/100+2;

timer=0 ;

EA=0;

tance();//探测位置

EA=1;

}

}

}

void zd0() interrupt 1

{

flag=1;

}

void zd3() interrupt 3

{

TH1=0xf8;

TL1=0x30;

timer++;

}

void StartModule() //启动模块

{

TX\_1=1;

TX\_2=1;

TX\_3=1;

TX\_4=1;

TX\_5=1; //启动一次模块

\_nop\_();

\_nop\_();

\_nop\_();

\_nop\_();

\_nop\_();

\_nop\_();

\_nop\_();

\_nop\_();

\_nop\_();

\_nop\_();

\_nop\_();

\_nop\_();

\_nop\_();

\_nop\_();

\_nop\_();

\_nop\_();

\_nop\_();

\_nop\_();

\_nop\_();

\_nop\_();

\_nop\_();

TX\_1=0;

TX\_2=0;

TX\_3=0;

TX\_4=0;

TX\_5=0;

}

## 附录四实物图

