

陶瓷学院科技艺术学院

本科生毕业设计（论文）

中文题目 基于单片机的多功能显示屏设计

英文题目 DESIGN OF MULTI-FUNCTION DISPLAY BASED ON MCU

院 系 景德镇陶瓷学院 科技艺术学院 工程系

专 业 自动化

姓 名 吕冬冬

学 号 201030455120

指导教师 吴德道

完成时间 2014-05-15

摘 要

随着出租车行业的发展，出租车已经是城市交通的重要组成部分，具有良好功能显示屏无论是对广大出租车司机朋友还是乘客来说都是很必要的。而随着单片机性能不断提高而价格不断下降，单片机控制在越来越多的领域得以应用。本设计按照传统的模式，在整个项目开发过程中，先根据控制系统要求设计原理图，软件编程，然后通过仿真器对系统硬件和软件调试，元器件的焊接，最后将调试成功的程序固化到单片机中。这一过程中的主要问题是，应用程序需要在硬件完成的情况下才能进行调试。本系统是由 STC89C52 单片机、LCD1602 液晶显示屏、DS18B20 温度传感器、按键、蜂鸣器等组成。软件在 keil 环境下用 C51 语言编写，包括日历显示与设置、定时闹钟、温度显示、计费相关功能显示等。

关键词：STC89C52单片机、LCD1602液晶显示屏、DS18B20温度传感器、计费显示

ABSTRACT

With the development of the taxi industry, the taxi is an important part of city traffic, has good function display both are necessary for the majority of taxi drivers or passengers. But along with the single chip constantly improve the performance and falling prices, MCU control has been applied in more and more fields. This design according to the traditional mode, throughout the project development process, according to the requirement of control system design schematics, software programming, and then through the emulator debug the hardware and software of the system, components of the welding, the success of the program debugging curing to the mcu. The main problem in this process is, applications need to complete the case for debugging hardware. The system is composed of STC89C52 single chip computer, LCD1602 display, DS18B20 temperature sensor, buttons, buzzer and so. Software in keil environment using C51 language, including the calendar display and settings, alarm clock, temperature display, charging function display.

KEYWORDS: STC89C52 MCU, LCD1602 screen, DS18B20 temperature sensor, charge display

目 录

1. 绪论.....	1
1.1 研究的内容及设计的目标.....	1
1.1.1 研究的内容.....	1
1.1.2 系统设计的目标.....	2
1.2 系统整体的方案.....	2
2. 硬件电路设计.....	4
2.1 单片机最小系统.....	4
2.2 温度采集电路设计.....	4
2.2.1 DS18B20 的主要特性.....	4
2.2.2 温度测量的步骤.....	5
2.2.3 DS18B20 的操作时序.....	5
2.3 LCD1602 液晶显示屏.....	6
2.3.1 1602 字符型 LCD 简介.....	6
2.3.2 1602LCD 的指令说明及时序.....	6
2.3.3 1602LCD 的一般初始化（复位）过程.....	8
2.3.4 1602LCD 的电路连接.....	8
2.4 蜂鸣器闹铃电路.....	9
2.5 按键调整电路.....	9
2.6 红外模块.....	10
2.7 电源模块.....	11
3. 软件部分设计.....	12
3.1 主程序流程.....	12
3.2 日期、时间设置子程序流程.....	12
3.3 闹钟设置子程序流程.....	13
3.4 红外接收装置.....	14
3.4.1 51 单片机解码红外遥控器原理.....	14
3.4.2 红外接收装置程序流程.....	15
3.5 程序设计问题.....	16
3.5.1 按键抖动问题.....	16
3.5.3 液晶显示的设置.....	18
3.5.4 中断设置.....	18
4. 系统仿真及调试.....	21
5. 结 论.....	24

6. 经济分析报告.....	25
致谢.....	26
参考文献.....	27
附录.....	28

1. 绪论

随着社会科技信息的进步与发展，国民经济的大幅提升以及生活水平的不断抬高，越来越多的人进城务工，这促使各个城市的交通日渐繁荣、紧张，然而也越来越广泛的应用出租车作为普通市民的交通工具。在这城市私家车年年递增，交通工具的拥挤，最重要的是停车位都难以解决，这已经成为很多城市的心头病，由此，许多人为求方便还是选择乘坐公交车、出租车等方式，，由此可见我国的出租车行业在城市交通运输中承担着重要的角色。不仅如此，出租车上的许多个显示屏也为我们提供了很大的方便。例如日历时间的显示、里程、速度的显示。还有更值得一提的是计费器的显示。出租车上的计费器是通过传感器与车子连接，计费器显示的总金额是通过车子行驶的里程与在此时间段的单价和起步价的函数。计费器通过乘客以上的信息，进行计费，然后直接显示其费值。

在我们日常乘坐出租车就可以发现出租汽车上的显示屏有日历时间显示、里程、速度显示、计费显示等多个显示屏。不仅占用很大的空间而且成本高。

现今我国生产计价器的企业有上百家，主要是集中在北京，上海，沈阳和广州等地。

所以如果能把日历时间、里程、速度、温度、闹钟和计费器这些人们日常生活中比不可少的东西合并起来做一个整体的改进，不但能减少汽车空间的利用，还能减少成本。

1.1 研究的内容及设计的目标

1.1.1 研究的内容

出租车上的计费器是通过传感器与车子连接，计费器显示的总金额是通过车子行驶的里程与在此时间段的单价和起步价的函数。然后由于出租车在日常使用中，有不同时间段、不同季节，它的起步价和每公里的价格不一样，所以在设计的过程中，也加入了可以调节起步价和每公里的价格。日期时间是通过定时计数算法来实现，闹钟是通过设置的时间与一直在走的时间不断进行比较，一旦相同，立马中断驱动蜂鸣器发声的原理工作的。温度的显示采用的是 DS18B20 电阻型传感器，原理是根据金属丝的电阻随温度的变化而工作的。本文设计的出租车多功

能显示器是利用 STC89C52 单片机作为核心控制，使其具有时间、日期、星期、即时温度、闹钟、速度、价格、里程数显示等相关功能。

1.1.2 系统设计的目标

本文是以液晶显示屏LCD1602和STC89C52单片机为主要研究对象，主要是利用STC89C52作为控制核心着重对LCD1602液晶显示屏显示信息的控制及其内部信息的研究。以及温度检测传感器DS18B20的温度计算，读取等实时研究。主要内容包括：

- 1) 实时温度显示（在不同情况下，可观察到明显变化）；
- 2) 年月日星期时分秒显示；
- 3) 年月日星期时分秒调整；
- 4) 闹钟定时小时分钟调整；
- 5) 计费显示行驶的里程和即时的费用（可根据不同季节、不同时间段调节起步价和单价；

1.2 系统整体的方案

首先：多功能出租车的显示屏第初始界面设置的是一个电子数字万年历的显示。可以看到界面显示年月日+星期+时分秒+温度等信息，当然也具有时间校准功能。本电路设计采用的核心控制芯片是 STC89C52 单片机，电压供电选用的是 5V 电压。

第二：闹钟界面的显示，通过闹钟的时间设置，然后通过时间的走动，不断进行验证，直到和所定的闹钟时间时、分一模一样的时候，中断启动蜂鸣器发声提醒人们，于此同时时间还是正常的走动。

第三：计费器界面的显示，这块显示有起步价(S)、单价(P)、即时速度(R)、里程、价钱的显示，本系统设计的计费器除了能实现一般出租车上的最基本的计费功能，而且还能根据白天，黑夜，中途等待来调节单价。单片机计算总价的公式为：总价=起步价+单价×（总里程-起步里程）

第四：由于考虑到按键按多了容易失灵，一般外行人都不能够自己去维修，为了一个按键去找专业人员维修，成本比较高，故在此采用红外遥控控制，如果遥控坏了的话，按键作为备用。然后只需要到超市里买一个万能遥控，只需要几

块钱，这样不但减去了麻烦还减少了成本。

本系统设计的硬件部分是以单片机 STC89C52 作为核心控制，液晶显示器 LCD1602 和温度传感器 DS18B20 作为显示输出设备，利用按键、蜂鸣器、红外遥控发送接收装置等构件进行控制。软件设计部分是在 keil UV2 环境下采用 C51 语言编写，包括时间日期、定时、闹钟、红外接收解码、起步价、单价的设置及显示和温度、里程、价钱的计算及显示。

该系统的系统框图如图1.1示：

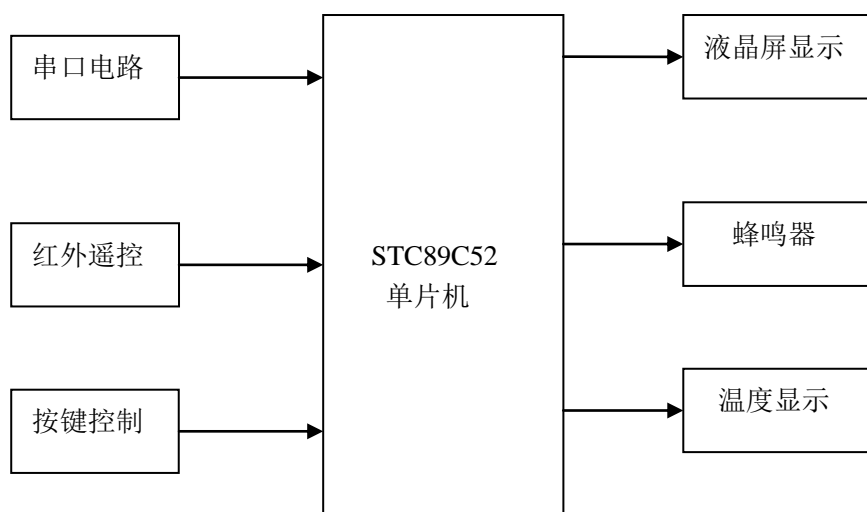


图1-1系统框图

2. 硬件电路设计

2.1 单片机最小系统

本系统设计的核心采用的是单片机STC89C52芯片，为了使单片机有合理的运行速度，所以选用的晶振是12MHZ。振荡器的电容33pF稳定性的振动频率，振荡器和快速冲击振动是适用于高水平的关键，还有其复位电路。这里利用的是按下按键，即为高电平，使得系统掉电恢复初始状态。

最小系统的电路设计如图2.1所示

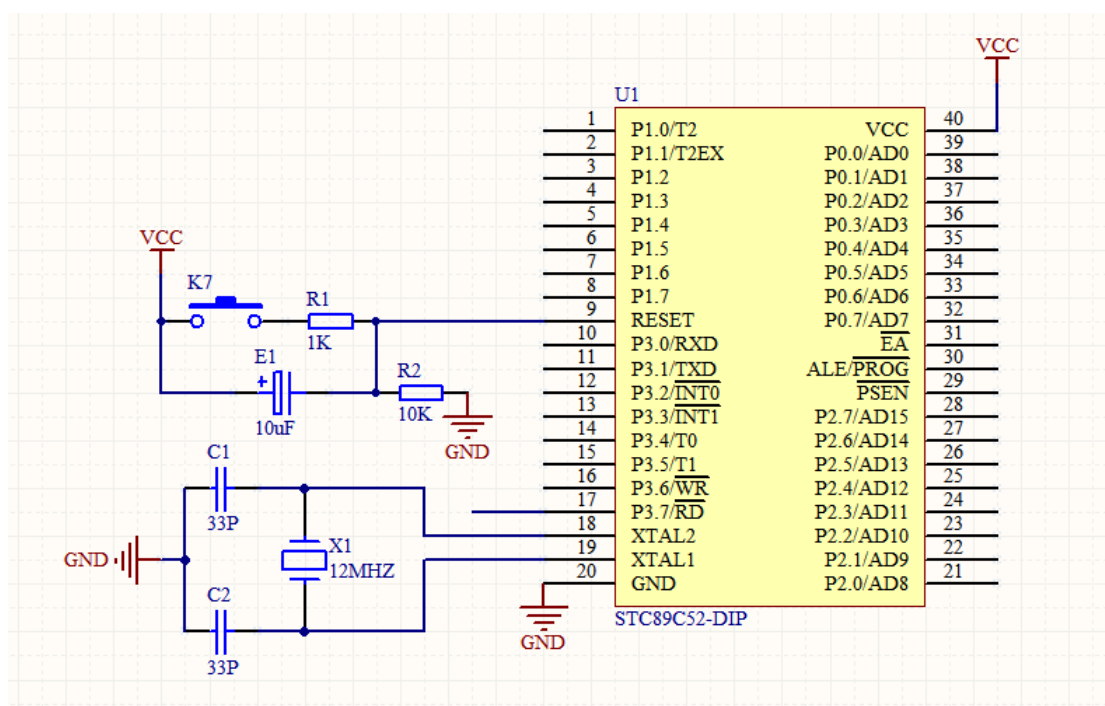


图2-1 单片机最小系统

2.2 温度采集电路设计

2.2.1 DS18B20 的主要特性

①一个更广泛的电压范围，适应电压范围：3 ~ 5.5V，可以由数据线在寄生电源模式。

②独特的单线接口方式，如果要实现 DS18B20 和微处理器的双向通讯，只需要一条口线即可。

③DS18B20 可以实现多点测温，并且唯一的三线上可以并联多个 DS18B20。

2.2.2 温度测量的步骤

(1) 为了确定温度传感器已经和总线相连，所以每次操作之前都要对其进行初始化工作，软件编程采用的是 read rom (33 h) 指令。

主要目的在于确定传感器已经连接到单总线上。

(2) 为了辨别该传感器是否已经连接到总线上，利用用指令 search rom (f0h) 进行操作。

(3) 指令发送成功后，还要对其电平进行查询，如果是高电平才进行转换温度，这里采用的是 convert t (44h) 这条操作指令。

(5) 读指令被发送出去后，利用 read scratchpad (beh) 操作指令，即可读出对应的二进制数显示温度。

2.2.3 DS18B20 的操作时序

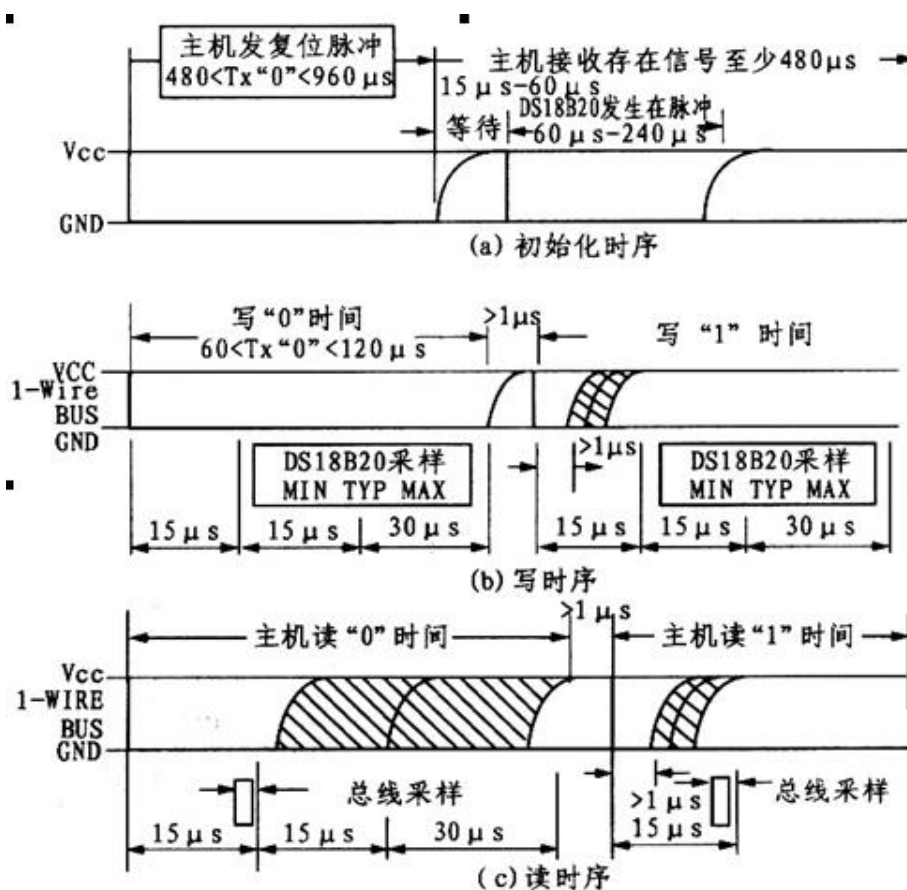


图 2-2 DS18B20 的操作时序

为了保证 DS18B20 的严格 I/O 时序。需要做较精确的延时。在温度传感器的操作中，为了严格的保证其时序信号输出，还需要对其作出较精准的延时，如

上图所示。下图为温度采集硬件电路接线图。

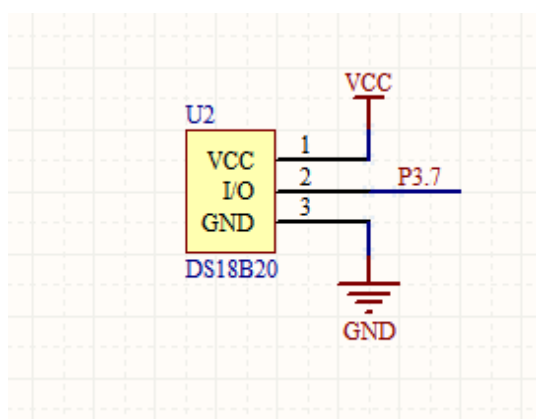


图 2-3：温度采集电路

2.3 LCD1602液晶显示屏

2.3.1 1602字符型LCD简介

本毕业设计采用的是LCD1602液晶显示屏，顾名思义即是16列×2行的字符型液晶显示屏，当然只能显示先关的字符数字等信息。

1602LCD主要技术参数：

显示容量:16×2个字符

芯片工作电压:4.5—5.5V

工作电流:2.0mA

工作电压:5.0V

字符尺寸:3.05×4.25(W×H)mm

各引脚接口说明如表所示：

编号	符号	引脚说明	编号	符号	引脚说明
1	VSS	电源地	9	D2	数据
2	VDD	电源正极	10	D3	数据
3	VL	液晶显示偏压	11	D4	数据
4	RS	数据/命令选择	12	D5	数据
5	RW	读/写选择	13	D6	数据
6	E	使能信号	14	D7	数据
7	D0	数据	15	BLA	背光源正极
8	D1	数据	16	BLK	背光源负极

表2-1 引脚接口说明

2.3.2 1602LCD的指令说明及时序

LCD1602液晶显示器的控制指令，如表2-2所示共有11条：

序号	指令	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0					
1	清显示	0	0	0	0	0	0	0	0	0	1					
2	光标返回	0	0	0	0	0	0	0	0	1	*					
3	置输入模式	0	0	0	0	0	0	0	1	I/D	S					
4	显示开/关控制	0	0	0	0	0	0	1	D	C	B					
5	光标或字符移位	0	0	0	0	0	1	S/C	R/L	*	*					
6	置功能	0	0	0	0	1	DL	N	F	*	*					
7	存贮器字符发生地址	0	0	0	1	0										
8	置数据存贮器地址	0	0	1	显示数据存贮器地址											
9	读忙标志或地址	0	1	BF	计数器地址											
10	CGRAM或DDRAM) 写数据	1	0	0												
11	从CGRAM或DDRAM读数	1	1	读出的数据内容												

表2-2 控制指令说明

LCD1602液晶显示屏都是通过指令编程来实现其相应的读写、屏幕和光标的操作的。（下面的解释高电平为1、低电平为0）

指令1：清除信息显示，并且，指令代码为01H，对应的光标恢复到初始地址位置00H。

指令2：复位光标，对应的光标恢复到初始地址位置00H。

指令3：设置显示和光标模式：低电平时，光标向左移动，高电平时，光标向右方移动，

指令4：显示开关控制。显示整体的开、关控制用字符D表示，开显示用高电平来表示，关显示利用低电平来表示。C：开关光标的控制用字符C表示，有光标用高电平来表示，无光标用低电平来表示。闪烁光标的控制用B来表示，闪烁用高电平表示，不闪烁用低电平表示。

指令5：移位显示或光标。显示的文字移动时用高电平表示，光标移动时用低电平表示。

指令6：设置功能命令，总线为4位时用高电平来表示，总线为8位时用低电平来表示，字符代码为DL。显示单行用低电平表示，显示双行用高电平来表示，字符代码为N。5x7的点阵字符的显示用低电平来表示，5x10的点阵字符的显示用高电平来表示，字符代码为F。

指令7：设置RAM字符地址。

指令8：DDRAM地址设置。

指令9：读地址光标和忙信号，字符代码为BF，忙状态用高电平表示，不忙状态

用低电平来表示。

指令10：写数据。

指令11：读数据。

2.3.3 1602LCD的一般初始化（复位）过程

1. 延时15mS
4. 写指令38H（不检测忙信号）
5. 延时5mS
8. 写指令38H（不检测忙信号）
9. 延时5mS
13. 写指令38H；第一次忙信号不检测，以后每次读/写数据、写指令操作都需要检测忙信号
15. 写指令38H：显示模式设置
17. 写指令08H：显示关闭
19. 写指令01H：显示清屏
22. 写指令06H：显示光标移动设置
25. 写指令0CH：显示开及光标设置

2.3.4 1602LCD的电路连接

读/写端RW在这里由于我们不需要读取数据，只需要写指令和显示数据信息，故在此始终作为写状态，在软件设计程序中可以看出其功能，有效电平为低电平。还有通过信号的操作传输室利用此液晶显示屏的6端口，EN使能信号端。如图2-7所示是其电路：

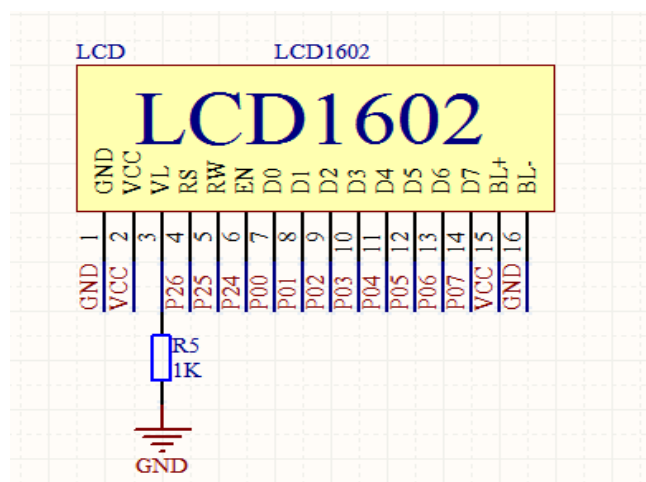


图2-5: LCD1602的连接电路

2.4 蜂鸣器闹铃电路

蜂鸣器采用的供电电压是直流电压，结构是一种一体化的电子发声器件，在通常的生活中电子玩具、计算机、电话机、复印机、定时器、打印机、报警器、汽车电子等设备中应用比较多。蜂鸣器的驱动方式有两种：一种直接驱动是利用PWM 输出口来实现的，另一种是使用单片机的定时系统对其电平实现定时翻转来产生脉冲信号，然后利用对应的管脚I/O对蜂鸣器进行控制。本毕业设计采用的是利用单片机I/O控制方式进行设计的，由于本设计买的是无源蜂鸣器，不能直接促使器发声工作，所以除此之外，还要外接一个8550的三极管来对其电流进行放大，这时只需要单片机的I/O给其一个低电流作为基极触发三极管工作的电流即可。当蜂鸣器被单片机I/O管脚给予一个低电平时，三极管即可驱动蜂鸣器促使其发出声音作为闹钟，图2-8所示电路：

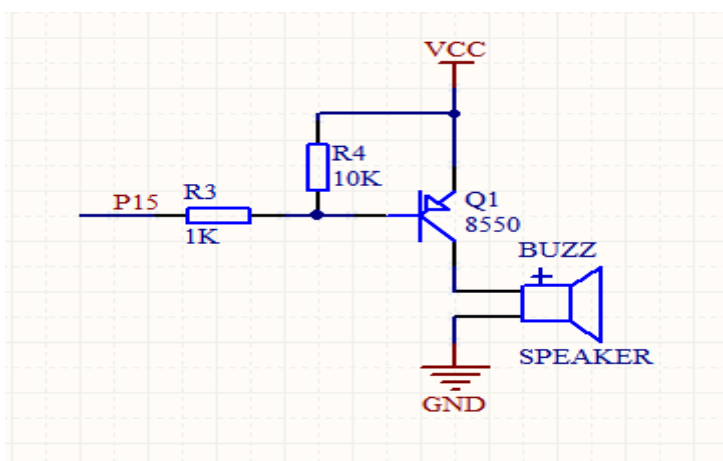


图 2-6: 蜂鸣器闹铃电路

2.5 按键调整电路

按键可分为矩阵按键和独立按键，一个按键控制一个接口的被称作独立按键，而矩阵按键是被排成N列M行，将其利用十字型全部并联在行、列的交点上。矩阵按键主要优点在于节省I/O口，由于单片机管脚有限的I/O口，并且其编程太过复杂。而独立按键编程简单，针对本课题所用的按键较少，故采用独立按键。两端系统六个独立键盘均采用查询方式，分别为设置、确认、加、减、闹钟设置、计费显示，其电路图如2-9所示：

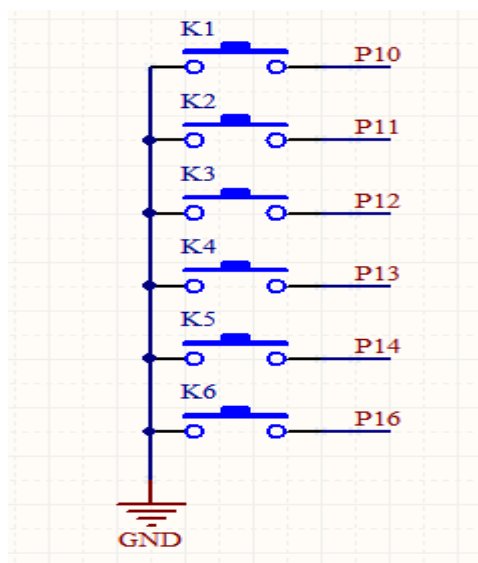


图 2-7: 按键调整电路

2.6 红外模块

本文是利用STC89C52单片机作为主控制单元，INT0中断口相连红外接受管，接收红外的脉冲信号，8个发光二极管作为显示解码输出。

如图 2-10 显示：远程控制的波形是这样的（经过反复试验结果）

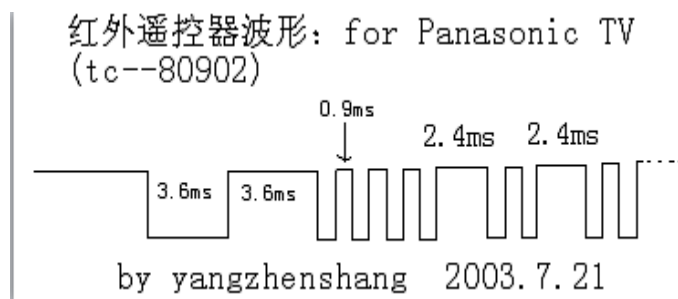


图 2-8: 红外遥控器波形图

开始位3.6ms低和3.6ms高电平，然后0.9ms低电平和0.9ms高电平周期为1.8ms的数据表示形式是“0”，“1”是用2.4ms高电平和0.9ms低电平周期为3.3ms表示，程序编写时，起始位是以大于3.4ms小于3.8ms高电平来表示，“1”是以大于2.2ms小于2.7ms高电平来表示，“0”是以大于0.84ms小于1.11ms高电平来表示。因此，我们主要使用高水平是通过测量确定的“1单片机”或“0”。定时器0设置为方式1: MOV TMOD, # 09h, 所以设置定时器0是一个1门，16位计数器，最大值为 2^{16} 个机器周期，这种方式的外部中断INT0控制，INT0允许计数器，其电路图如2-11所示:

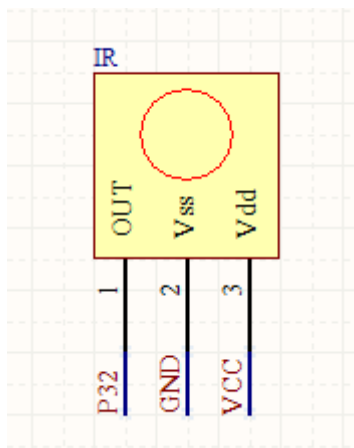


图2-9：红外接收模块

2.7 电源模块

当前，像闪存盘、移动硬盘、数码相机等越来越多使用各种外部设备USB接口。USB其中两根线是串行通道用来传送数据的，另两根为设备下游提供电源，总共是四线电缆。

USB 信号各自使用双绞线传输差分信号利用半双工的方式并协同工作的，分别标记为 D+ 和 D-，另外还用来消除长导线的电磁干扰的。本设计只用了其中供电的两根线，使用的是 5V 直流源。原理图如 2-12 所示

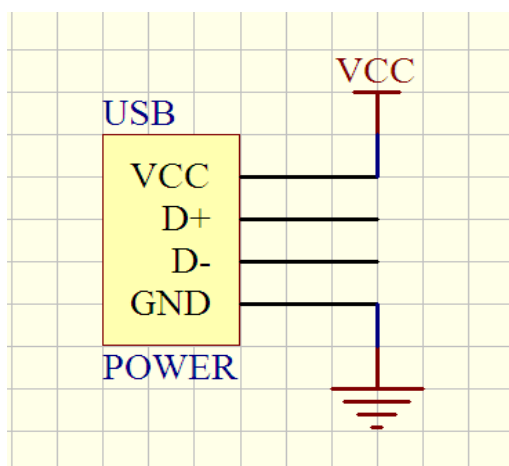


图2-10：电源模块

3. 软件部分设计

3.1 主程序流程

首先开始初始化主程序，同时中断打开，然后对闹钟、按键进行扫描及读取温度值。当有K1键按下时，执行设置，当有K5键按下时，则进入闹钟设置。当有K6键按下时，进行出租计费，不管是日期、时间、闹铃还是计费，设置完后需要退出就按K2确认键，就可实时恢复显示万年历。当然，还有外部中断红外遥控，可随时替代进行 按键设置。如图3-1所示是主程序流程图：

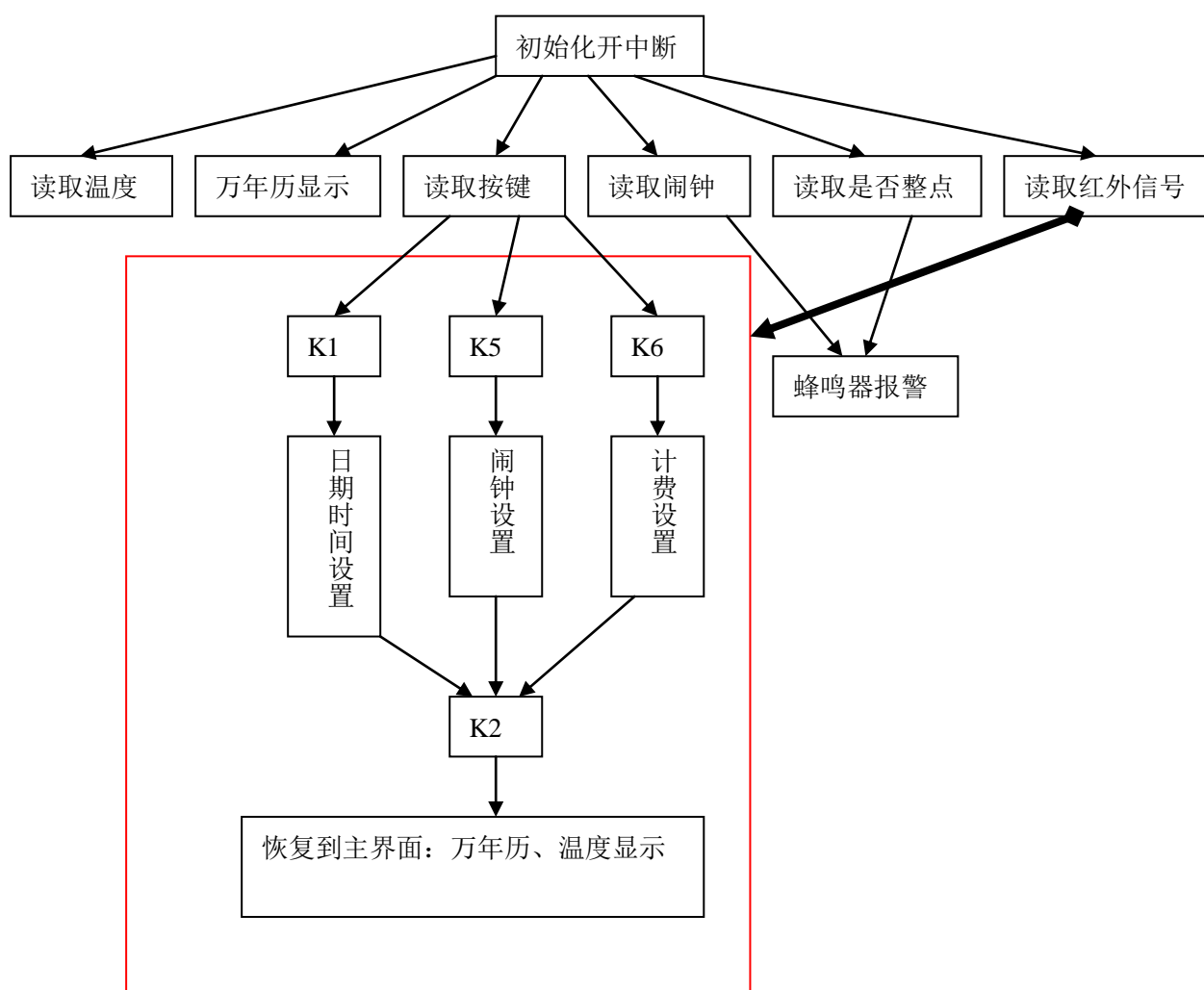


图3-1 主程序流程框图

3.2 日期、时间设置子程序流程

执行命令用if语句来判断，由num标志位来记录K1按下的次数，通过单片机控制来设置。扫描键盘是由系统程序的不断动作，低电平是由K1键按下后产生的，即是num+1。首先，所有的变量初始化在规定时间之前，和设置开始时间，在数据读取同一时间。当K1 = 1 M年进入设置，地址指针年显示位置控制显示年，加号和减号循环通过两个if语句。当K1=2时，地址指针指向month显示位置处，变量最大值为12。当K1=3时，地址指针指向day显示位置处，变量最大值设为31。这时week会随着所设定的年月日变化而变化，无需进行week设置，周分别由周一，周二，周三，周四，周五，周六，周日的英文来显示一个字符串。当K1 = 4，地址指针时显示的位置，最大值为23。当K1=5时，地址指针指向min显示位置处，最大值设为59。

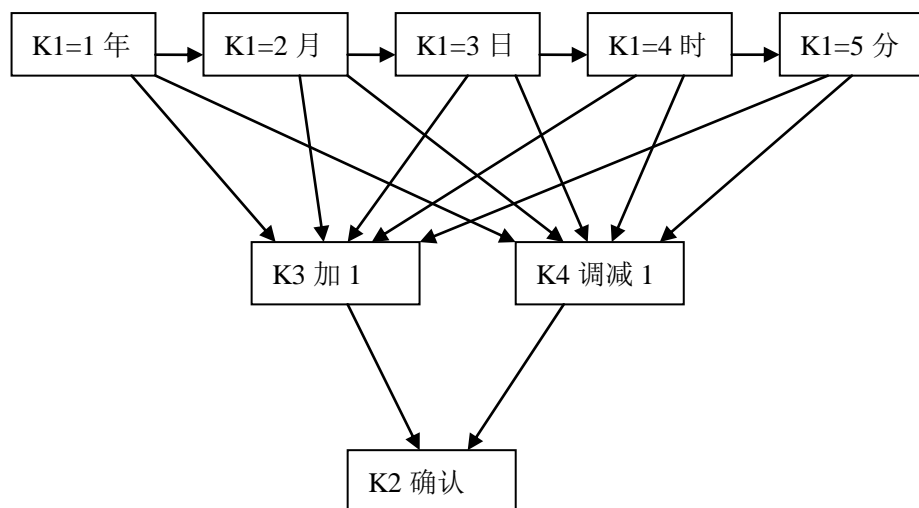


图3-2 日期、时间设置子程序流程框图

3.3 闹钟设置子程序流程

如图3-3所示是时间程序设置流程图。开始时间设定程序之前关闭全局中断，直到结束中断，将重新开放，进入时间状态。

闹钟设置通过K1按下的键数的判断。当单片机检测到K5键按下到闹钟时间设置界面，当k1按下按钮，然后在报警时间设置，按K2，确认设置并返回到主界面。

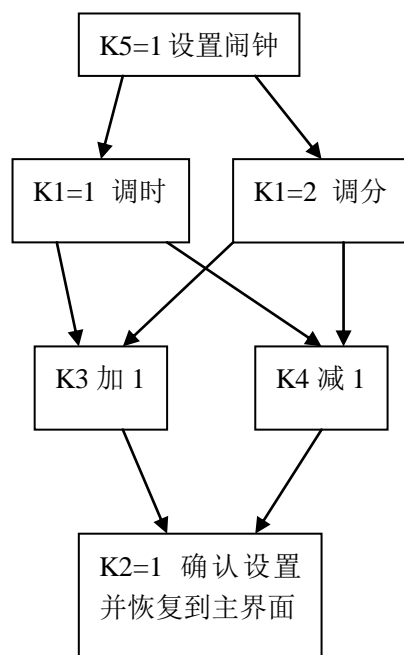


图3-3 闹钟设置流程框图

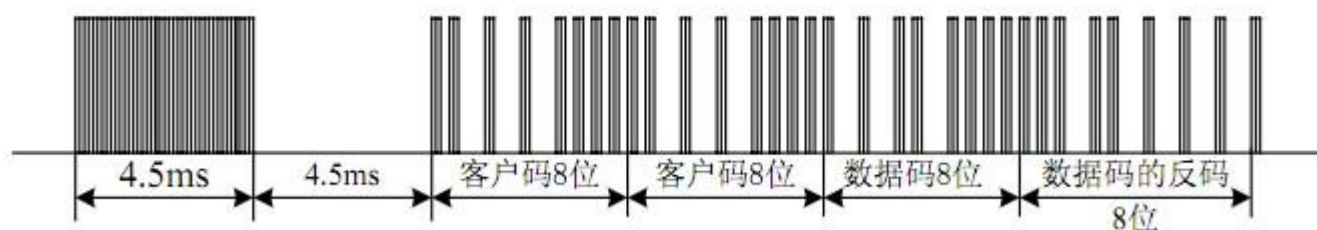
3.4 红外接收装置

3.4.1 51单片机解码红外遥控器原理

电视遥控器如东芝 TC9012，飞利浦 SAA3010T 等遥控码的发射使用的就是靠专用集成发射芯片来实现，通常发射遥控信号的彩电，就是在 38KHz 的载波上用对应的控制指令和系统码（由 0 和 1 组成的序列）控制某个按键来调制，然后将信号经驱动放大利用红外发射管发射出去，遥控码所采用的格式也会根据不同公司的遥控芯片而不一样。NEC 标准和 PHILIPS 标准是比较普遍的两种。

NEC 标准：遥控载波的频率为 38KHz（占空比为 1:3）；当某个按键按下时，系统首先发射一个完整的全码，如果键按下超过 108ms 仍未松开，接下来发射的代码（连发代码）将仅由起始码（9ms）和结束码（2.5ms）组成。一个完整的全码=引导码+用户码+用户码+数据码+数据反码。其中，引导码高电平 4.5ms，低电平 4.5ms；系统码 8 位，数据码 8 位，共 32 位；其中前 16 位为用户识别码，能区别不同的红外遥控设备，防止不同机种遥控码互相干扰。后 16 位为 8 位的操作码和 8 位的操作反码，用于核对数据是否接收准确。收端根据数据码做出应该执行什么动作的判断。连发代码是在持续按键时发送的码。它告知接收端，某键是在被连续地按着。

NEC标准下的发射码表示



对于 0.56ms 低电平 + 0.565ms 高电平=来说用发射数据 0 表示：高传输数据的时间，0.56ms 低水平+ 1.69ms 高水平= 2.25ms 用数据 0 来表示。数据红外线 0.56ms 用发射码“0”发射 38khz 来表示的，0.565ms 的停止发射，红外线 0.56ms 用发射码“1”发射 38khz 来表示的，1.69ms 的停止发射。

当集成接收机接收红外信号为 38KHz 时，需要注意的是这时输出低电平，否则高水平。按下按键的时候，在 1 和 0 之间的切换控制代码，如果连续按钮，然后控制相同的代码。。全码=启动代码“11”+控制代码和用户代码+用户代码。数据 0 用“低电平 1.778ms+高电平 1.778ms”表示；用“高电平 1.778ms+低电平 1.778ms”来表示数据 1。用延时 114ms 来表示连续码重复。

NEC，东芝，该编码格式有共同的特点：远程控制代码是“引导代码+代码+系统代码（或系统的代码从反）+数据码+数据码组成”；数据“0”和“1”的定义相同；不同的是，启动代码级的持续时间不同，系统代码的数字的长度，第一个简码和代码的最后一位之间的延迟，脉冲速度的不同等，所以你可以把相同的部分做成一个子程序，包括生成的数据“0”和“1”子程序还有“zero”和“one”9ms、4.5ms、2.25ms、22ms、45ms 等时间控制子程序... 按键的处理，用于外部中断方式，按下按钮，然后发送到 CPU 的中断请求，它还包括一个处理部分的抖动，同时多个密钥只处理一个按键。由单片机的 P3.0、P3.1、P3.4、P3.5 四位的输入数值来区分判断机芯的不同状态，比如东芝 TC90××系列工作状态用“0000”表示，NEC 公司的 PD61××系列工作状态用“0001”来表示。

判断哪一个键按下，在什么样的状态下，与相应的遥控编码传输的远程控制代码，严格按照程序，这是一个特定时期的水平，一定数量的脉冲调制对 38KHZ 的载波，一个一个地发射出去。以一体化接收头输了的波形是与发射波形是反向的。

3.4.2 红外接收装置程序流程

红外接收子程序流程图如图3-4所示。在系统进行初始化的时候首先打开全局中断，保证红外接收装置在全程都能接收信号进行遥控设置，使其可以代替按键进行操作。

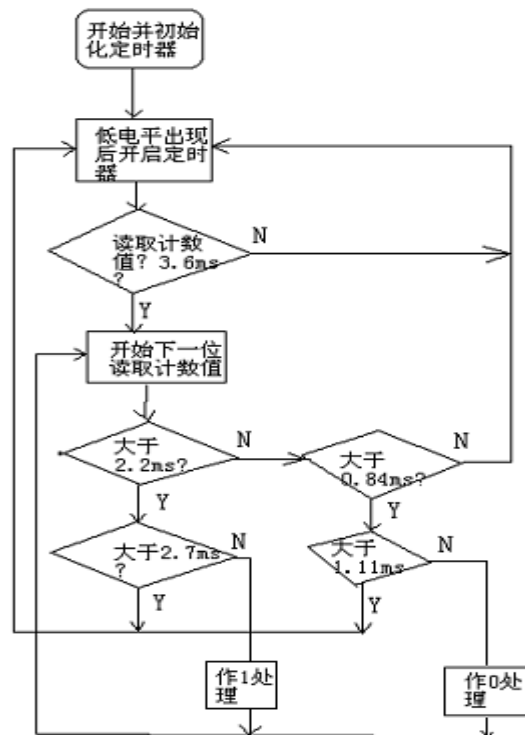


图3-4： 红外遥控器解码流程框图

3.5 程序设计问题

3.5.1 按键抖动问题

在按下按钮，机械原因会产生抖动，抖动的结果是，当一个按钮按下，因为抖动的原因使系统读取次数多键按下，导致错误的设置。以下是解决方案：当按下按钮时，读取键值，如果5ms后按确认键，是相等的，则消除了按键抖动的问题。

3.5.2 蜂鸣器设置

a) 初始化设置BEEP=1;

b) 整点报时函数

```
void clockhourcall()
```

```
{
```

```
    if(min==0&&sec==0&&hour>6&&hour<23)
```

```

{
    EX0=0;
    for(rp=175;rp<190;rp++)
    {
        p4=music_tab[rp];
        if(p4==0xff) { delayms(100), TR1=0;}
if(p4!=0xff) {m4=music_tab[rp], rp=rp+1, n1=music_tab[rp];} //取频率
常数 和 节拍常数
        TR1=1;
        while(n1!=0) BEEP=~BEEP, delay4(m4);
TR1=0;
    }
    EX0=1;
}
}

```

c) 音乐播放

```

void musicplay()
{ unsigned int a;
if(sec==sec1&&hour==hour1&&min==min1)//&&hour1!=0&&min1!=0&&sec1!=0)
{EX0=0;
    for(rp=0;rp<190;rp++)
    {p4=music_tab[rp];
        if(p4==0xff) { delayms(100), TR1=0;}
        if(p4!=0xff)
{m4=music_tab[rp], rp=rp+1, n1=music_tab[rp];} //取频率常数 和 节拍常数
        TR1=1;//开定时器1
        while(n1!=0) BEEP=~BEEP, delay4(m4);
        TR1=0; //关定时器1
        if(rp%6==0 || rp%6==1)

```

```

        {
            dispd2();
            dispt2();
        }
        if(rp%12==0 || rp%12==1)
        {

            delay(20);
            delay(20);
            for(a=0;a<16;a++)
            {
                delay(20);
            }
            dispt2();
        }
    }
    EX0=1;}
}

```

3.5.3 液晶显示的设置

- a) 初始化设置
- b) 显示模式设置
- c) 显示/开关及光标设置
- d) 写入指令码函数write_com（）写指令过程：输入：rs=0，D0-D7=指令码，E=高脉冲
- e) 写入指令和数据时rw端已经接地。

3.5.4 中断设置

```

EA=1;  //开全局中断
ET0=1;  //开定时1中断
TR0=1;//启动定时器

```

```

{
    TH0=(65536-50000)/256;
    TL0=(65536-50000)%256;//50ms
    dd++;
    if(dd==100)
    dd=0;
    ..... }

```

3.5.5 红外解码中断设置

```

void IR_IN() interrupt 0 using 0
{unsigned char j,k,N=0;
    EX0 = 0;
    delay3(15);
    if (IRIN==1)
    { EX0 =1;
      return; }
    while (!IRIN)
        {delay3(1);}
    for (j=0;j<4;j++)          //收集四组数据
    { for (k=0;k<8;k++)        //每组数据有8位
    {
        while (IRIN)
        {delay3(1);}
        while (!IRIN)
        {delay3(1);}
        while (IRIN)
        {delay3(1);
          N++;
          if (N>=30)
          { EX0=1;

```



```
        return;}

    }

    if (N>=8) {IRCOM[j] = IRCOM[j] | 0x80;} //数据最高位补"1"

    N=0;

} //end for k

} //end for j

if (IRCOM[2] != ~IRCOM[3])

{ EX0=1;

    return; }

IRCOM[5]=IRCOM[2] & 0x0F; //取键码的低四位

IRCOM[6]=IRCOM[2] >> 4; //右移4次，高四位变为低四位

if (IRCOM[5]>9)

{ IRCOM[5]=IRCOM[5]+0x37;}

else

    IRCOM[5]=IRCOM[5]+0x30;

if (IRCOM[6]>9)

{ IRCOM[6]=IRCOM[6]+0x37;}

else

    IRCOM[6]=IRCOM[6]+0x30;

    redcontrol=IRCOM[6]+IRCOM[5];

    if (redcontrol==0x6A || redcontrol==0x68 || redcontrol==0x67 || redcontrol==0x64 || redcontrol==0x6b || redcontrol==0x69)

        beep();

    EX0 = 1; }
```

4. 系统仿真及调试

根据系统设计方案，系统的仿真分为两大部分：利用 protel99se 进行硬件设计和 Keil UV2、Proteus 软硬件联调仿真。由于采用了模块化设计方法，在系统的设计中，在每个电路模块的功能如此简单的一步一步的测试。首先是万年历模块仿真调试如图 4-1：经过软件的仿真，能够正常的显示日历、时间及温度，并且能过按照程序的要求进行调节。

其次闹钟模块仿真调试如图 4-2：经过软件的仿真，能够正常的显示闹钟调节界面，并且能过按照程序的要求进行调节。

然后出租车计费显示模块仿真调试如图 4-3：经过软件的仿真，能够正常的显示起步价、单价、速度、里程及总费用，并且能过按照程序的要求进行调节。

系统总体仿真图如4-4：经过软件的仿真，能够正常的显示各个功能界面，并且能过按照程序的要求进行调节。最后进行面包板的插线调试，无误之后即对实物焊接，通过最后的实物系统调试，最终能正常的实现各个功能，实物如图4-5所示：

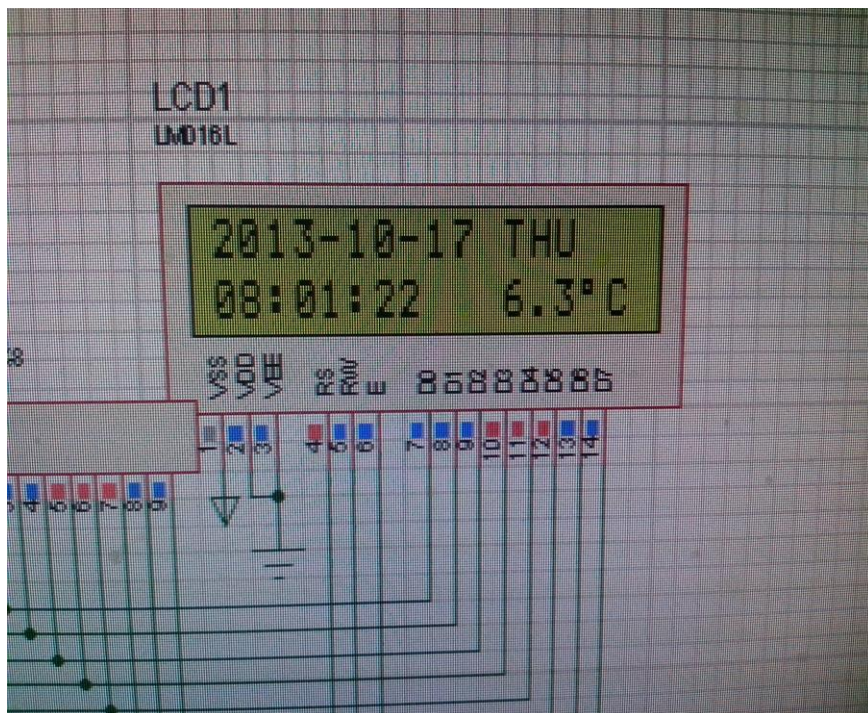


图 4-1：万年历模块仿真调试

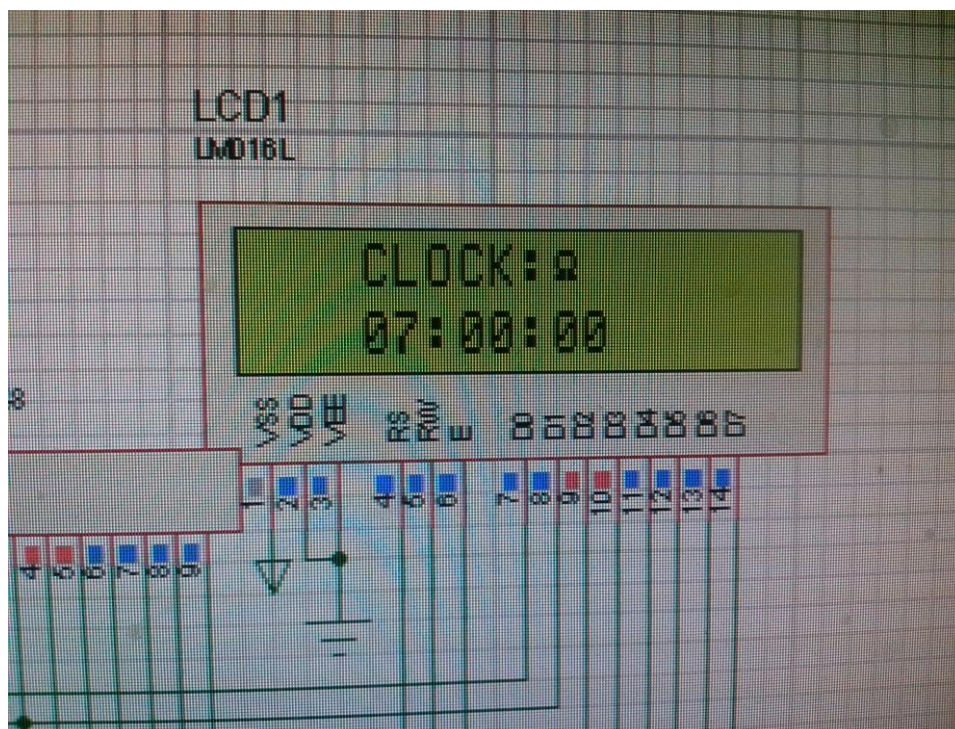


图 4-2：闹钟模块仿真调试

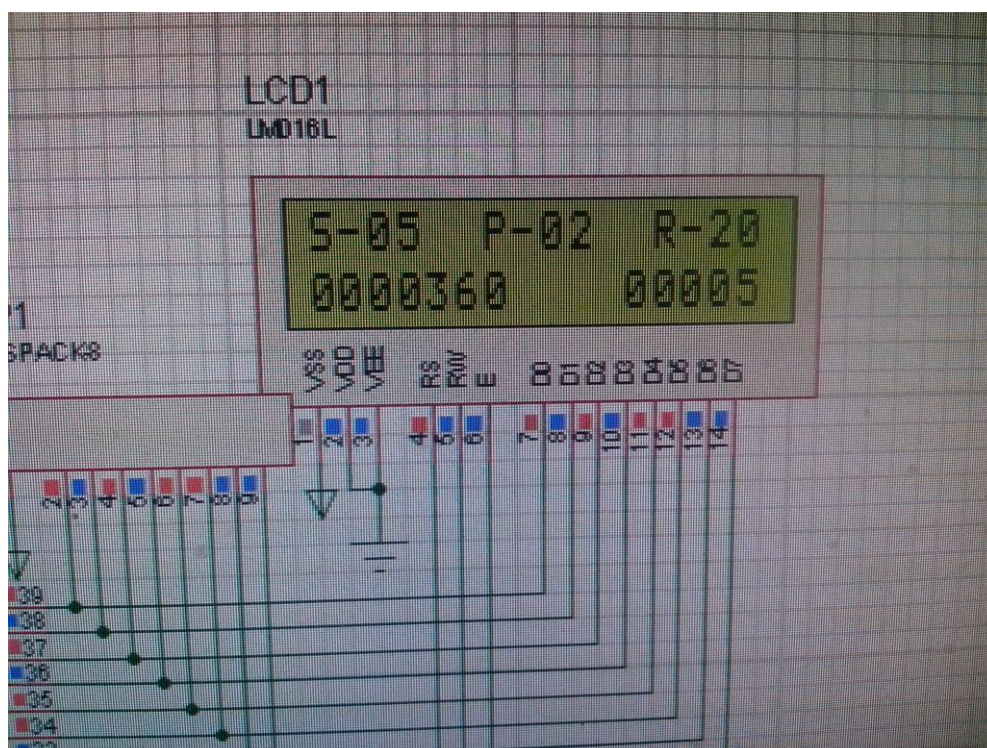
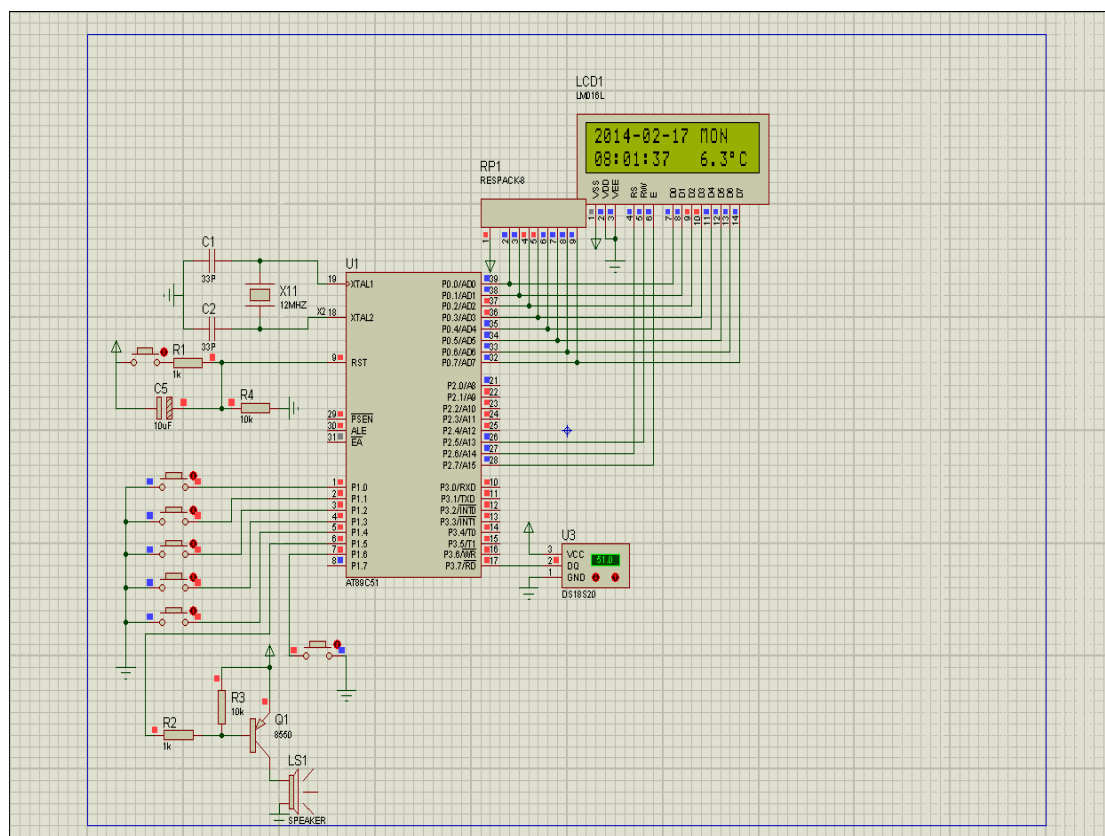
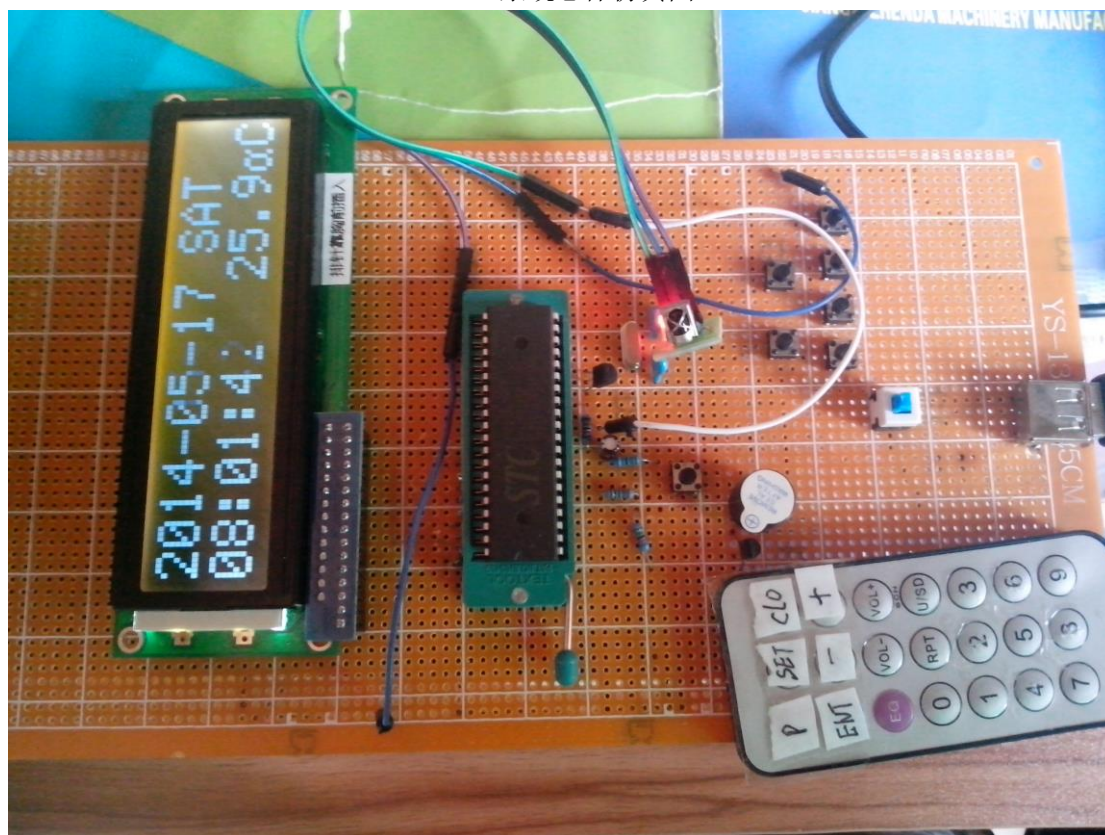


图 4-3：出租车计费显示模块仿真调试



4-4：系统总体仿真图



4-5：实物图

5. 结 论

本文是一篇关于用单片机实现集万年历、计费器等多重功能叠加的论文，设计中主要用到了STC89C52和LCD1602液晶显示屏。程序的设计作为本课题的重点设计，所以通过这次毕业的设计，不但巩固了之前学习的C语言课程，而且还提高了自己在程序编写方面的能力，对以后工作中解决实际问题的能力有所帮助，当然虽然硬件的设计比较简单，但是也不容忽视这块，因为复杂都是从一个简单的步骤中演变而来的，相信以后自己能在这个行业有所发展。

通过对本课题的研究，显示屏能够按照预期的效果显示日历时间、温度、闹钟和计费里程、速度、单价等功能，并能够按照按键对LCD显示进行调节。虽然此设计还达不到一个实际的应用效果，但是其突出的优点在于，很大的程度上降低了成本，为后者对此相关设计提供了宝贵的参考意见。

6. 经济分析报告

本设计系统采用单片机控制。以单片机为核心, 设计上采用 STC89C52 单片机为主控器, 利用单片机丰富的 IO 端口, 及其控制的灵活性, 不仅能够实现所要求的一些功能, 而且还可以更方便的进行系统的升级, 况且还具有扩展功能。比数字电路实现性能更加稳定, 而且还能更好的调节。

本设计的优点在于能对显示屏进行重复利用, 这也是没有采用数码管显示的重要原因之一, 并且数码管只能显示数字或者一些简单是字母, 在此多功能的显示上有局限性, 然而用LCD液晶显示器显示具有零辐射、低功耗、体积小、字符显示精确等优点。设计元器件费用清单如下所示:

名称	数量	单价
STC89C52单片机	1	3.00
单片机锁紧座	1	1.65
液晶显示屏	1	7.80
蜂鸣器	1	0.45
温度传感器	1	4.15
红外接收装置	1	4.00
红外遥控	1	3.8
电阻 (1K)	3	0.12
电阻 (10K)	2	0.12
晶振 (12MHZ)	1	2.00
极性电容 (10uF)	1	0.15
电容 (33P)	2	0.15
按钮	7	0.30
开关	1	0.30
PNP三极管8550	1	0.20
USB数据线	1	0.95
USB接口	1	0.21
线仔	5	10.00
洞洞板	1	3.70
总计		43.05

致谢

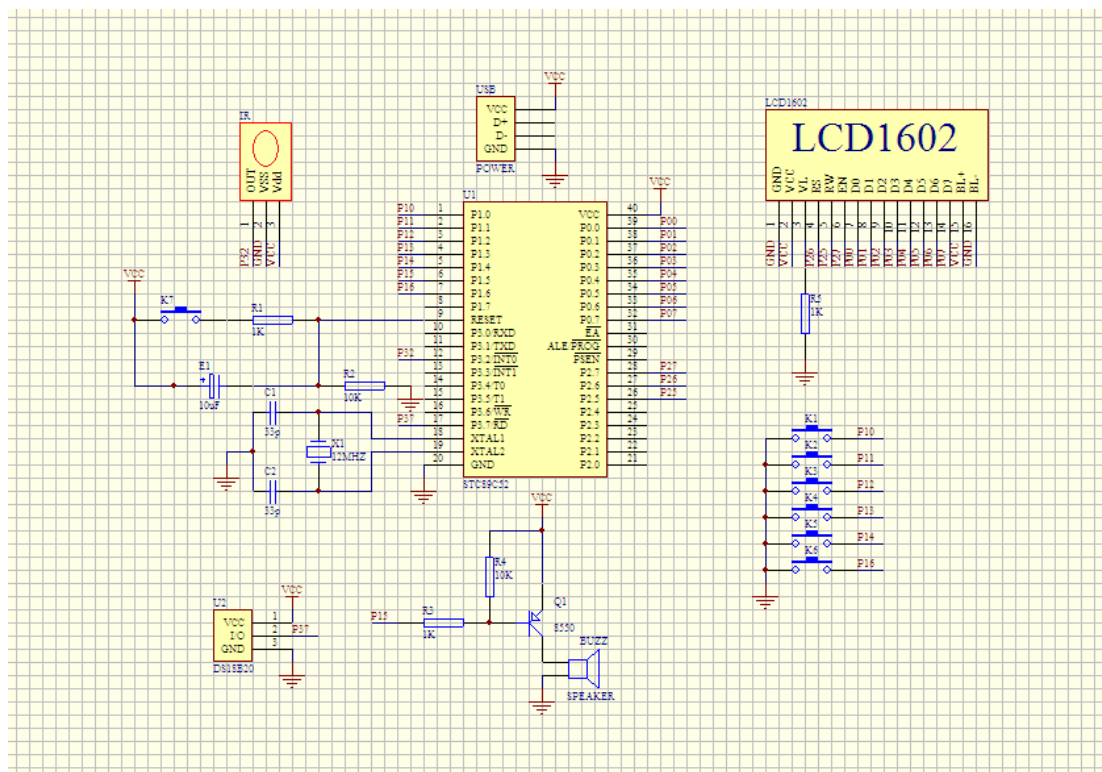
本设计论文的完成，在一定意义上，我要感谢我的学校，如果没有学校给我提供这次的学习机会和毕业设计的平台，我根本不可能完成本次毕业设计。在此设计中，老师无论是从选题的指导，还是论文框架的细节的修改，都给予了我详尽的指点，而且还提出了良多的宝贵意见与创议，他们以其严谨和务实的教学态度、表现出其高度敬业的精神对我及同学们产生及其重要的影响。

从最开始不知道怎么写毕业论文，到开始进入课题到论文的完成，再到毕业论文答辩稿顺利的完成，其中有很多同学，老师，朋友向我提供的帮助，在论文即将完成之际，我无法用语言来表达我此时的心情，在这里我以崇高的敬意表示由衷的感谢！

参考文献

- [1]王平. 电子 CAD 应用教程[M]. 第一版. 浙江: 东南大学出版社, 2009
- [2]杨忠宝, 董晓明. C 语言程序设计[M]. 北京: 北京大学出版社, 2010
- [3]邱关源. 电路[M]. 第 5 版. 北京: 高等教育出版社, 2006
- [4]章彬宏, 吴青萍. 模拟电子技术基础[M]. 北京: 北京理工大学出版社, 2008
- [5]阎石. 数字电子技术基础[M]. 北京: 中国电力出版社, 2007
- [6]李金平, 沈明山. 电子系统设计[M]. 北京: 电子工业出版社, 2007
- [7]何小艇. 电子系统设计[M]. 第四版. 浙江: 浙江大学出版社, 2008
- [8]郭天祥, 新概念51单片机c语言教程. 北京: 电子工业出版社, 2009
- [9]冯育长. 单片机系统设计与实例指导[M]. 第三版. 西安: 西安电子科技大学, 2007
- [10]谢维成, 杨加国. 单片机原理与应用及 C51程序设计[M]. 第2版. 北京: 清华大学出版社, 2009
- [11]李朝青. 单片机原理及接口技术[M]. 第三版. 北京: 北京航空航天大学出版社, 2005
- [12]W. Simpson, Editor, The Point-to-Point Protocol, RFC1661 [EB/OL]. July 1994
- [13]WAVECOM, AT Commands Interface Guide, Revision 002[EB/OL]. 6th November 2003
- [14]于永权. 单片机在控制系统中的应用[M]. 第二版. 北京: 北京航空航天大学出版社, 2010
- [15]谭浩强, C程序设计（第二版）[M]. 北京: 清华大学出版社, 2003
- [16]张伟, 王力, 赵晶, ProtelDXP 入门与提高. 北京: 人民邮电出版社, 2003
- [17]弘道工作室, 融会贯通 Protel99电路设计. 北京: 人民交通出版社, 2000
- [18]郭天祥, 新概念51单片机C语言教程. 北京: 电子工业出版社, 2009
- [19]谭浩强. MCS-51 单片机应用教程[M]. 北京: 清华大学出版社, 2004

附录



总体设计电路图

源程序：

```
#include <reg52.h>
#include <intrins.h> //库函数头文件，代码中引用了_nop_()函数
// 定义控制信号端口
typedef bit BOOL ;
sbit RS=P2^6;    //P2.4
sbit RW=P2^5;    //P2.5
sbit E=P2^7;     //P2.6
sbit PWM=P2^0;   //P2.0
sbit tab1=P1^4;  //切换闹钟键
sbit tab2=P1^6;  //切换计费键
/*年历参数声明*/
unsigned char timer1,n1=0;
unsigned int num,i,j,s0,dd,temp,temp1=0,temp2=0,temp3=0;
/*温度参数声明*/
unsigned char data disdata[5];
unsigned char flagdat=0x20;
//红外参数
unsigned char IRCOM[7];
unsigned char redcontrol;
```

```
unsigned char lcdd[]="0123456789";
/*年历声明调用函数*/
void dispt2();//闹钟显示程序二行
void dispd2();//闹钟显示程序一行
void dispd3();
void dispt3();
void motordeal();//电机处理程序
/*关于三个的主体函数*/
void promain();
/*主函数*/
void main()
{
    BEEP=1;
    PWM=0;
    delay(250);//延时
    IE = 0x89;

    TMOD&=0xf0;
    TMOD|=0x10;
    TMOD=TMOD|0x01;

    TL0=(65536-50000)%256;
    TH1=0xd8;TL1=0xef;
    EA=1; //开全局中断
    ET0=1; //开定时1中断
    TR0=1;//启动定时器
    //TR1=0;
    min=1;
    hour=8;
    day=17; //日
    month=10; //月
    year=2013; //年
    delay(255);
    while(1)
    {
        clockhourcall(); //整点报时函
        motordeal(); //电机处理程序
    }
}
void promain()
{
    {
        redcontrol=0xff;
```

```
TR0=0;
k=1;
if(f==0)
{
    n=n+3;
    if(n==9)
    {
        n=0;
        m=0;
        f=1;
    }
}
else
{
    m=m+3;
    if(m==9)
    {
        m=0;
        n=0;
        f=0;
    }
}
if(f==0)
{
    lcd_w_cmd(0x0d);
    lcd_w_cmd(0x83+n);
}
else
{
    lcd_w_cmd(0x0d);
    lcd_w_cmd(0xc1+m);
}
}

{
    redcontrol=0xff;
    k=0;
    TR0=1;
    n=-3;
    m=-3;
    f=0;
}

{
```

```

        redcontrol=0xff;
        if(k==1)
        {
            if(f==0)
            {
                if(n==0){year=year+1;if(year==1000) year=0;}
                else if(n==3) {month=month+1;if(month==13)
month=1;}

                else
                {
                    day=day+1;

                    if(day>=29)
                        daydecide1();
                }
                dispd(); //调用第一行显示函数

                lcd_w_cmd(0x83+n); //返回设置值显示地址
            }
            else
            {
                if(m==0){hour++;if(hour==24) hour=0;}
                else if(m==3) {min++;if(min==60) min=0;}
                else {sec++;if(sec==60) sec=0;}
                dispt(); //调用第二行显示函数
            }
        }
    }

    {
        redcontrol=0xff;
        if(k==1)
        {
            if(f==0)
            {
                if(n==0){year--;if(year<0) year=999;}
                else if(n==3) {month--;if(month==0) month=12;}
                else
                {
                    day=day-1;
                    if(day==0)
                        daydecide2();
                }
            }
        }
    }

```

```

        dispd();    //调用第一行显示函数

        lcd_w_cmd(0x83+n); //返回设置值显示地址
    }
    else
    {
        if(m==0) {hour--;if(hour<0) hour=23;}
        else if(m==3) {min--;if(min<0) min=59;}
        else {sec--;if(sec<0) sec=59;}
        dispt();    //调用第二行显示函数
    }
    } //if(k==1)
    } //if() //按下减1键
    if(k==0) {dispd();dispt();}
}
{

    if(min==0&&sec==0&&hour>6&&hour<23)
    {
        EX0=0;
        for(rp=175;rp<190;rp++)
        {
            p4=music_tab[rp];
            if(p4==0xff) { delayms(100), TR1=0;}
            if(p4!=0xff) {m4=music_tab[rp], rp=rp+1, n1=music_tab[rp];} //取频率
            常数 和 节拍常数
            TR1=1;
            while(n1!=0) BEEP=~BEEP, delay4(m4);
            TR1=0;
        }
        EX0=1;
    }
}
/*****音乐播放*****/
void musicplay()
{

    unsigned int a;

    if(sec==sec1&&hour==hour1&&min==min1)//&&hour1!=0&&min1!=0&&sec1!=0)
    {
        EX0=0;
        for(rp=0;rp<190;rp++)

```

```

{
    p4=music_tab[rp];
    if(p4==0xff) { delayms(100), TR1=0;}
    if(p4!=0xff)
{m4=music_tab[rp], rp=rp+1, n1=music_tab[rp];} //取频率常数 和 节拍常数
    TR1=1; //开定时器1
    while(n1!=0) BEEP=~BEEP, delay4(m4);
    TR1=0; //关定时器1
    if(rp%6==0 || rp%6==1)
    {
        dispd2();
        dispt2();
    }
    if(rp%12==0 || rp%12==1)
    {
        lcd_w_cmd(0x0c); //设置光标不显示、不闪烁
        delay(20);

        delay(20);
        for(a=0; a<16; a++)
        {

            delay(20);
        }
        dispt2();
    }
    EX0=1;
}
}

{ TH1=0xd8;
  TL1=0xef;
  n1--;
}
/*****控制频率延时*****/

{
    unsigned i4=m4*3;
    while(--i4);
}
/*****豪秒延时子程序*****/

```

```

{
    while(--a4);                //采用while(--a) 不要采用while(a--);
}
/*****中断函数*****/
void IR_IN() interrupt 0 using 0
{
    unsigned char j,k,N=0;
    EX0 = 0;
    delay3(15);
    if (IRIN==1)
    { EX0 =1;
      return;
    }

    //确认IR信号出现
    while (!IRIN)
        {delay3(1);}

    for (j=0;j<4;j++)           //收集四组数据
    {
        for (k=0;k<8;k++)       //每组数据有8位
        {
            while (IRIN)
                {delay3(1);}

            {delay3(1);}

            {
                delay3(1);
                N++;
                if (N>=30)
                {
                    EX0=1;
                    return;}
            }

            //高电平计数完毕

            if (N>=8) {IRCOM[j] = IRCOM[j] | 0x80;} //数据最高位补"1"
            N=0;
        } //end for k
    } //end for j

    if (IRCOM[2] != ~IRCOM[3])
    { EX0=1;
      return; }
    IRCOM[5]=IRCOM[2] & 0x0F;    //取键码的低四位

```

```

    IRCOM[6]=IRCOM[2] >> 4;          //右移4次，高四位变为低四位
    if (IRCOM[5]>9)
    { IRCOM[5]=IRCOM[5]+0x37;}
    else
        IRCOM[5]=IRCOM[5]+0x30;
    if (IRCOM[6]>9)
    { IRCOM[6]=IRCOM[6]+0x37;}
    else
        IRCOM[6]=IRCOM[6]+0x30;
        redcontrol=IRCOM[6]+IRCOM[5];
    if (redcontrol==0x6A || redcontrol==0x68 || redcontrol==0x67 || redcontrol==
0x64 || redcontrol==0x6b || redcontrol==0x69)
        beep();
        EX0 = 1;
}
/*****/
void beep()
{
    unsigned char i;
    for (i=0;i<100;i++)
    {
        delay3(4);

    }
    BEEP=1;          //关闭蜂鸣器
}
/*****/
{
    while(x--)
    {
        for (i = 0; i<13; i++) {}
    }
}
/*****/
/*延时函数*/

{
    unsigned char j,i;
    for(i=0;i<t;i++)
        for(j=0;j<20;j++);
}
/*延时函数1*/
void delay1()
{

```



```
    _nop_();
    _nop_();
    _nop_();
}
BOOL lcd_r_start()
{
    BOOL s;
    RS=0;
    E=1;    //E端时序
    delay1();
    s=(BOOL) (P0 & 0x80);    //从LCD的数据口读状态
    E=0;
}
/*LCD写命令函数*/
void lcd_w_cmd(unsigned char com)
{
    while(lcd_r_start());
    RS=0;

    E=0;    //E端时序
    delay1();
    delay1();
    E=1;
    delay1();
    E=0;
}
/*LCD写数据函数*/
void lcd_w_dat(unsigned char dat)
{
    while(lcd_r_start());
    RS=1;
    E=0;    // E端时序
    delay1();

    delay1();
    E=1;
    delay1();
    E=0;
}
/*****定时中断函数*****/
{
    TH0=(65536-50000)/256;
    TL0=(65536-50000)%256;//50ms
```

```
dd++;
if(dd==100)
dd=0;

if(year%400==0||year%4==0&&year%100!=0) num=1;
count1++;
if((count1%2==0)&&count1!=0)
{
    count++;
    count1=0;
}
if(count==10)
{
    count=0;
    sec++;
    if(sec==60)
    {
        sec=0;
        min++;
        if(min==60)
        {
            min=0;
            hour++;
            if(hour==24)
            {
                hour=0;
                day++;
                if(day==29&&month==2&&num==0) {day=1;month++;}
                else if(day==30&&month==2&&num==1) {day=1;month++;}
                else if(day==31&&(month==4||month==6||month==9||month==11))
                    {day=1;month++;}
                else if(day==32&&
                    {day=1;month++;}
                    if(month==13)
                    {
                        month=1;
                        year++;
                        if(year==1000) year=0;
                    }
                }
            }
        }
```

```
    }
}
}
/*按键扫描函数*/
{

    P1=0x7f;
    cod=P1&0x7f;    //读入P1口键值
    {
        if(cod!=0x7f)
        {
            cod=P1&0x7f; //读入键值
            do            //等待键释放
            {
                del=P1&0x7f;
            }while(del!=0x7f);

        }
    }
    return(0x7f);    //返回该值
}
/*ds1820复位*/
void ds1820rst()
{ unsigned char x=0;
    DQ = 1;          //DQ复位

    DQ = 0;          //DQ拉低
    delay(10); //delay_18B20(100); //精确延时大于480us
    DQ = 1;          //拉高
    delay(4); //delay_18B20(40);
}
/*ds1820读数据*/
{ 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;
        delay(1); //delay_18B20(10);
    }
    return(dat);
}
```

```

/*ds1820写数据*/
{unsigned char i=0;
  for (i=8; i>0; i--)
  { DQ = 0;
    DQ = wdata&0x01;
    delay(1); //delay_18B20(10);
    DQ = 1;
    wdata>>=1;
  }
}
/*读取温度值并转换*/
int read_temp()
{
  ds1820rst();

  ds1820rst();

  a=ds1820rd();
  b=ds1820rd();
  tvalue=b;
  tvalue<<=8;
  tvalue=tvalue|a;

  tflag=0;
  else
  {tvalue=~tvalue+1;
    tflag=1;
  }

  return(tvalue);
}
/*****/
{
  disdata[0]=tvalue/1000+0x30; //百位数

  disdata[3]=tvalue%10+0x30; //小数位

  if(tflag==0)
    flagdat=0x20; //正温度不显示符号
  else
    flagdat=0x2d; //负温度显示负号:-

  if(disdata[0]==0x30)

```

```
        if(disdata[1]==0x30)

            }

        }

    }
/*第一行显示日期函数*/
void dispd()
{

    delay(20);
    lcd_w_dat(lcdd[year/1000]);
    delay(2);
    lcd_w_dat(lcdd[year%1000/100]);
    delay(2);
    lcd_w_dat(lcdd[year%100/10]);
    delay(2);
    lcd_w_dat(lcdd[year%10]);
    delay(2);
    lcd_w_dat(' ');
    delay(2);
    lcd_w_dat(lcdd[month/10]);
    delay(2);
    lcd_w_dat(lcdd[month%10]);
    delay(2);
    lcd_w_dat(' ');
    delay(2);
    lcd_w_dat(lcdd[day/10]);
    delay(2);
    lcd_w_dat(lcdd[day%10]);
    delay(2);
    lcd_w_dat(' ');
    delay(2);
    weekdecide();
    switch(week)
    {
        case 0:
        {
            lcd_w_dat('S');
            delay(2);
            lcd_w_dat('U');
            delay(2);
            lcd_w_dat('N');
            delay(2);
            break;
        }
    }
}
```

```
}  
case 1:  
{  
    lcd_w_dat('M');  
    delay(2);  
    lcd_w_dat('O');  
    delay(2);  
    lcd_w_dat('N');  
    delay(2);  
    break;  
}  
case 2:  
{  
    lcd_w_dat('T');  
    delay(2);  
    lcd_w_dat('U');  
    delay(2);  
    lcd_w_dat('E');  
    delay(2);  
    break;  
}  
case 3:  
{  
    lcd_w_dat('W');  
    delay(2);  
    lcd_w_dat('E');  
    delay(2);  
    lcd_w_dat('D');  
    delay(2);  
    break;  
}  
case 4:  
{  
    lcd_w_dat('T');  
    delay(2);  
    lcd_w_dat('H');  
    delay(2);  
    lcd_w_dat('U');  
    delay(2);  
    break;  
}  
case 5:  
{  
    lcd_w_dat('F');
```

```
        delay(2);
        lcd_w_dat('R');
        delay(2);
        lcd_w_dat('I');
        delay(2);
        break;
    }
    case 6:
    {
        lcd_w_dat('S');
        delay(2);
        lcd_w_dat('A');
        delay(2);
        lcd_w_dat('T');
        delay(2);
        break;
    }
}
lcd_w_dat(' ');
delay(2);
lcd_w_dat(' ');
delay(2);
}
/*第二行显示时间、星期函数*/
void dispt()
{
    lcd_w_cmd(0x0c); //设置光标不显示、不闪烁
    delay(20);
    delay(20);

    lcd_w_dat(lcdd[hour/10]);
    delay(2);
    lcd_w_dat(lcdd[hour%10]);
    delay(2);
    lcd_w_dat(':');
    delay(2);
    lcd_w_dat(lcdd[min/10]);
    delay(2);
    lcd_w_dat(lcdd[min%10]);
    delay(2);
    lcd_w_dat(':');
    delay(2);
    lcd_w_dat(lcdd[sec/10]);
    delay(2);
```

```

    lcd_w_dat(lcdd[sec%10]);
    delay(2);
    lcd_w_dat(' ');
    delay(2);
    delay(2);
    lcd_w_dat(disdata[1]); //显示十位
    delay(2);
    delay(2);
    delay(2);

    delay(2);
    lcd_w_dat(0xdf);
    delay(2);
    lcd_w_dat('C');
    delay(2);
}
{
    unsigned char num=0;
    if(year%400==0 || year%4==0 && year%100!=0) num=1;
    if(day==29 && month==2 && num==0) {day=1;}
        else if(day==30 && month==2 && num==1) {day=1;}
        else
if(day==31 && (month==4 || month==6 || month==9 || month==11)) {day=1;}
        else if(day==32 &&
                        {day=1;}
}
{
    unsigned char num1=0;
    if(year%400==0 || year%4==0 && year%100!=0) num1=1;
    if(month==2 && num1==0) {day=28;}
        else if(month==2 && num1==1) {day=29;}
                        {day=30;}
                        {day=31;}
}
{
    unsigned int i, day1=0, num=0;
    i=year;
    if(i>=2011)
    {
        for(i=2011; i<year; i++)
        {
            if(i%400==0 || i%4==0 && i%100!=0)
            {
                day1=day1+366;
            }
        }
    }
}

```



```
    }
    else day1=day1+365;
}
if(year%400==0 || year%4==0&&year%100!=0)
num=1;
switch(month)
{
    case 1: { day1=day1+day;break;}
    case 2: { day1=day1+31+day;break;}
    case 3: { day1=day1+59+num+day;break;}
    case 4: { day1=day1+90+num+day;break;}
    case 5: { day1=day1+120+num+day;break;}
    case 6: { day1=day1+151+num+day;break;}
    case 7: { day1=day1+181+num+day;break;}
    case 8: { day1=day1+212+num+day;break;}
    case 9: { day1=day1+243+num+day;break;}
    case 10: { day1=day1+273+num+day;break;}
    case 11: { day1=day1+304+num+day;break;}
    case 12: { day1=day1+334+num+day;break;}
}
week=(5+day1)%7;
}
if(i<2011)
{
    day1=0;
    num=0;
    for(i=year+1;i<2011;i++)
    {
        if(i%400==0 || i%4==0&&i%100!=0)
        {
            day1=day1+366;
        }
        else day1=day1+365;
    }
    if(year%400==0 || year%4==0&&year%100!=0)
        num=1;
    switch(month)
    {
        case 1: { day1=day1+365+num-day;break;}
        case 2: { day1=day1+334+num-day;break;}
        case 3: { day1=day1+306-day;break;}
        case 4: { day1=day1+275-day;break;}
        case 5: { day1=day1+245-day;break;}
        case 6: { day1=day1+214-day;break;}
```

```
        case 7: { day1=day1+184-day;break;}
        case 8: { day1=day1+153-day;break;}
        case 9: { day1=day1+122-day;break;}
        case 10: { day1=day1+92-day;break;}
        case 11: { day1=day1+61-day;break;}
        case 12: { day1=day1+31-day;break;}
    }
    week=6-(day1+1)%7;
}
}
/*****闹钟显示程序*****/
void dispt2()
{
    char clocknum1;
    unsigned int p;

    delay(20);

    delay(20);

    delay(20);
    for(clocknum1=0;clocknum1<3;clocknum1++)
    {
        lcd_w_dat(' ');
        delay(2);
    }
    lcd_w_dat(lcdd[hour1/10]);
    delay(2);
    lcd_w_dat(lcdd[hour1%10]);
    delay(2);
    lcd_w_dat(':');
    delay(2);
    lcd_w_dat(lcdd[min1/10]);
    delay(2);
    lcd_w_dat(lcdd[min1%10]);
    delay(2);
    lcd_w_dat(':');
    delay(2);
    lcd_w_dat('0');
    delay(2);
    lcd_w_dat('0');
    delay(2);
    for(p=0;p<5;p++)
```

```

    {
        lcd_w_dat(' ');
        delay(2);
    }
}
/*****闹钟显示第一行*****/
void dispd2()
{
    unsigned char code clock1[]={ "    CLOCK:" };
    unsigned int clocknum1,p1;

    delay(20);

    delay(20);
    for(clocknum1=0;clocknum1<9;clocknum1++)
    {
        lcd_w_dat(clock1[clocknum1]);
        delay(2);
    }
    lcd_w_dat(0xf4);
    delay(2);
    for(p1=0;p1<6;p1++)
    {
        lcd_w_dat(' ');
        delay(2);
    }
}
/*****闹钟处理程序*****/
void clockdeal()
{
    unsigned char clocknum=0;
    unsigned int k1=0,p1,b;
    if(key==0x6f||redcontrol==0x6b)//*****将程序中的 2f全部
        // *****一定要记得
    {
        clocknum=clocknum+1;
        if(clocknum==2) clocknum=0;
        else
        {
            while(p)
            {
                dispd2();
                dispt2();
            }
        }
    }
}

```

```
{
    redcontrol=0xff;
    m1=m1+3;
    if(m1==6)
        m1=0;

    k1=1;
    lcd_w_cmd(0x0d);
    lcd_w_cmd(0xc4+m1);
    for(b=0;b<100;b++)
        delay(100);
}

{
    redcontrol=0xff;
    p=0;
    m1=-3;
    k1=0;
}

{
    if(k1==1)
    {
        redcontrol=0xff;
        if(m1==0){hour1++;if(hour1==24) hour1=0;}
        else {min1++;if(min1==60) min1=0;}
        dispt2();    //调用第二行显示函数

        for(b=0;b<100;b++)
            delay(100);
    }
}

{
    if(k1==1)
    {
        redcontrol=0xff;
        if(m1==0){hour1--;if(hour1<0) hour1=23;}
        else {min1--;if(min1<0) min1=59;}
        dispt2();    //调用第二行显示函数

        for(b=0;b<100;b++)
            delay(100);
    }
}
```

```
        }
        }//if() //按下减1键
    } //while
} //else
} //if
}
/*****计费显示界面*****/
void dispt3()
{
    //unsigned char code motor1[]={ " MOTOR:" };
    unsigned int p1,s0=20,s1=5,s2=2,s=0;
    unsigned char i,j,k,p=1,k1=0;
    int money=0;//假设起步价为5元，500m/1元
    delay(20);
    delay(20);

    delay(20);
    while(p)
    {
        lcd_w_cmd(0xc0);
        delay(2);
        s=s+s0;
        lcd_w_dat(lcdd[s/1000000]);
        delay(2);
        lcd_w_dat(lcdd[s%1000000/100000]);
        delay(2);
        lcd_w_dat(lcdd[s%1000000%100000/10000]);
        delay(2);
        lcd_w_dat(lcdd[s%1000000%100000%10000/1000]);
        delay(2);
        lcd_w_dat(lcdd[s%1000000%100000%10000%1000/100]);
        delay(2);
        lcd_w_dat(lcdd[s%1000000%100000%10000%1000%100/10]);
        delay(2);
        lcd_w_dat(lcdd[s%10]);
        delay(2);
        for(p1=0;p1<4;p1++) //中间空四格
        {
            lcd_w_dat(' ');
            delay(2);
        }
        if(s<=2500)
            lcd_w_dat(lcdd[s1/10000]);
        delay(2);
        lcd_w_dat(lcdd[s1%10000/1000]);
        delay(2);
    }
}
```

```

lcd_w_dat(lcdd[s1%10000%1000/100]);
delay(2);
lcd_w_dat(lcdd[s1%10000%100/10]);
delay(2);
lcd_w_dat(lcdd[s1%10]);
delay(2);
if(s>2500)    //车行2500m以内只需支付起步价5元，超过按照500m/元
{
    lcd_w_cmd(0xcb);
    delay(2);
    money=s1+(s-2000)*s2/1000;
    lcd_w_dat(lcdd[money/10000]);
    delay(2);
    lcd_w_dat(lcdd[money%10000/1000]);
    delay(2);
    lcd_w_dat(lcdd[money%10000%1000/100]);
    delay(2);
    lcd_w_dat(lcdd[money%10000%1000%100/10]);
    delay(2);
    lcd_w_dat(lcdd[money%10]);
    delay(2);
}
PWM=1;
for(i=0;i<255;i++)
    for(j=0;j<255;j++)
        for(k=0;k<2;k++){

            redcontrol=0xff;
            m1=m1+3;
            if(m1==9)
                m1=0;
            k1=1;
            lcd_w_cmd(0x0d);
            lcd_w_cmd(0x83+2*m1);
            delay(2);
        }

    { redcontrol=0xff;
      if(k1==1)
      {
        if(m1==0)
        {s1++;}
        else if(m1==3)

```

```

        {s2++;}
    else
        {s0++;}
/*****调用dispt3*****/

    delay(2);

    delay(2);
    lcd_w_dat('S');
    delay(2);
    lcd_w_dat('-');
    delay(2);
    lcd_w_dat(lcdd[s1/10]);
    delay(2);
    lcd_w_dat(lcdd[s1%10]);
    delay(2);
    for(p1=0;p1<2;p1++)
    {
        lcd_w_dat(' ');
        delay(2);
    }
    lcd_w_dat('P');
    delay(2);
    lcd_w_dat('-');
    delay(2);
    lcd_w_dat(lcdd[s2/10]);
    delay(2);
    lcd_w_dat(lcdd[s2%10]);
    delay(2);
    for(p1=0;p1<2;p1++)
    {
        lcd_w_dat(' ');
        delay(2);
    }
    lcd_w_dat('R');
    delay(2);
    lcd_w_dat('-');
    delay(2);
    lcd_w_dat(lcdd[s0/10]);
    delay(2);
    lcd_w_dat(lcdd[s0%10]);
    delay(2);

    lcd_w_cmd(0x0d);

```

```

        lcd_w_cmd(0x83+2*m1);
        delay(2);
    }
}

{
    redcontrol=0xff;
    if(k1==1)
    {
        if(m1==0)
        {s1--;}
    else if(m1==3)
        {s2--;}
    else
        {s0--;}
}
/*****调用dispt3*****/

    delay(2);
    lcd_w_dat('S');
    delay(2);
    lcd_w_dat('-');
    delay(2);
    lcd_w_dat(lcdd[s1/10]);
    delay(2);
    lcd_w_dat(lcdd[s1%10]);
    delay(2);
    for(p1=0;p1<2;p1++)
    {
        lcd_w_dat(' ');
        delay(2);
    }
    lcd_w_dat('P');
    delay(2);
    lcd_w_dat('-');
    delay(2);
    lcd_w_dat(lcdd[s2/10]);
    delay(2);
    lcd_w_dat(lcdd[s2%10]);
    delay(2);
    for(p1=0;p1<2;p1++)
    {
        lcd_w_dat(' ');
        delay(2); //空2格
    }
}

```



```

        lcd_w_dat('R');
        delay(2);
        lcd_w_dat('-');
        delay(2);
        lcd_w_dat(lcdd[s0/10]);
        delay(2);
        lcd_w_dat(lcdd[s0%10]);
        delay(2);
        lcd_w_cmd(0x0d);
        lcd_w_cmd(0x83+2*m1);
        delay(2);
    }
}

{
    redcontrol=0xff;
    p=0;
    PWM=0;
    for(i=0;i<255;i++)
    for(j=0;j<255;j++)
    for(k=0;k<30;k++);
    m1=-3;
    k1=0;
}

}

}

/*****计费显示第一行*****/
void dispd3()
{
    //unsigned char code motor1[]={ " MOTOR:" };
    unsigned int p1,s0=20,s1=5,s2=2;
    delay(20);
    delay(20);
    lcd_w_dat('S');
    delay(2);
    lcd_w_dat('-');
    delay(2);
    lcd_w_dat(lcdd[s1/10]);
    delay(2);
    lcd_w_dat(lcdd[s1%10]);
    delay(2);    //s1表示出租起步价
    for(p1=0;p1<2;p1++)
    {
        lcd_w_dat(' ');
    }
}

```

```

        delay(2); //空2格
    }
    lcd_w_dat('P');
    delay(2);
    lcd_w_dat('-');
    delay(2);
    lcd_w_dat(lcdd[s2/10]);
    delay(2);
    lcd_w_dat(lcdd[s2%10]);
    delay(2); //s2表示每公里多少钱
    for(p1=0;p1<2;p1++)
    {
        lcd_w_dat(' ');
        delay(2); //空2格
    }
    lcd_w_dat('R');
    delay(2);
    lcd_w_dat('-');
    delay(2);
    lcd_w_dat(lcdd[s0/10]);
    delay(2);
    lcd_w_dat(lcdd[s0%10]);
    delay(2);
}
/*****计费显示界面处理程序*****/
void motordeal()
{
    unsigned char motornum=0;

    unsigned int k1=0,p=1,s0=20;
    if(key==0x3f||redcontrol==0x69)
    {

        motornum=motornum+1;
        if(motornum==2) motornum=0;
        else
        {
            // if(dd<s0)
            PWM=1;
            dispd3();
            dispt3();
        }
    }
}

```