**《课程名称》**

**课程设计报告**

　　　　　　　　　　（2016—2017学年第 二 学期）

题 目 汽车信号灯控制系统

学生姓名 李佳

专业班级 物联网1503

学生学号 311509080306

教师姓名 李静

成 绩**：**

评 语：

教师签名：

日期：

目录

[摘要 3](#_Toc486168703)

[1.设计题目 4](#_Toc486168704)

[2.设计要求 4](#_Toc486168705)

[3.设计目的 4](#_Toc486168706)

[4.设计方案 4](#_Toc486168707)

[4.1汽车驾驶操作与信号灯的明暗情况 4](#_Toc486168708)

[4.2单片机的系统设计 5](#_Toc486168709)

[4.2.1开关状态检测 5](#_Toc486168710)

[4.2.2输出控制 5](#_Toc486168711)

[4.2.3定时器和计数器 6](#_Toc486168712)

[4.2.4汽车转弯时灯显示 6](#_Toc486168713)

[4.2.5汽车转弯灯控制 6](#_Toc486168714)

[4.3设计方案 7](#_Toc486168715)

[5.硬件设计 7](#_Toc486168716)

[5.1单片机控制系统中功能模块的设计 7](#_Toc486168717)

[5.1.1时钟电路 7](#_Toc486168718)

[5.2.2信号灯电路 8](#_Toc486168719)

[5.2.3故障监控电路 8](#_Toc486168720)

[5.2元件清单 9](#_Toc486168721)

[5.3单片机介绍 9](#_Toc486168722)

[5.3.1单片机引脚介绍 9](#_Toc486168723)

[5.3.2单片机硬件功能实现 10](#_Toc486168724)

[6.软件分析 11](#_Toc486168725)

[6.1汽车转弯系统流程图 11](#_Toc486168726)

[6.1.1汽车转弯控制系统主流程图 11](#_Toc486168727)

[6.1.2中断服务程序流程图 12](#_Toc486168728)

[6.1.3控制系统功能键流程图 12](#_Toc486168729)

[6.2程序说明 12](#_Toc486168730)

[7.总结与展望 13](#_Toc486168731)

[参考文献 13](#_Toc486168732)

[附录：程序清单 14](#_Toc486168733)

摘要

此设计是一个单片机控制系统，在汽车做左转弯，右转弯，紧急刹车，停靠等，实现对汽车信号指示灯的控制，主要是对单片机的并行输入/输出口电路的应用，通过I/O口控制发光二极管的亮﹑灭﹑闪烁，加上一些复位电路﹑按键电路﹑驱动电路来模拟汽车灯的功能。

汽车在驾驶时有左转弯、右转弯、刹车、合紧急开关、停靠等操作。在左转弯或右转弯时，通过转弯操作杆应使左转开关或右转开关合上，从而使左头灯、仪表板左转弯灯、左尾灯或右头灯、仪表板右转弯灯、右尾灯闪烁；合紧急开关时要求前面所述的6个信号灯全部闪烁；汽车刹车时，两个尾灯点亮；如正当转

弯时刹车，则转弯时原应闪烁的信号灯仍应闪烁。以上闪烁，都是频率为1Hz的低频闪烁；在汽车停靠而停靠开关合上时，左头灯、右头灯、左尾灯、右尾灯按频率为30Hz的频率闪烁。通过做编写程序和做proteus仿真，使我对单片机的各元件有了更深刻的认识。

关键字：单片机、汽车信号转弯灯

1.设计题目

汽车信号灯控制系统

2.设计要求

1. 汽车上有一转弯控制杆。此控制杆有三个位置：

中间位置时，汽车不转弯；

1. 向上时，汽车左转；
2. 向下时，汽车右转；
3. 汽车转弯时，要求左右尾灯、左右头灯和仪表板上的2个指标灯相应地发出闪烁信号；
4. 当应急开关合上时，所有6个信号灯都应闪烁；
5. 汽车刹车时，2个尾灯发出不闪烁的信号；
6. 如刹车时正在转弯，则相应的转弯闪烁信号不受影响。
7. 汽车转弯或应急状态下，外部信号灯和仪表板指示灯的闪烁频率为1HZ。
8. 停靠(合上停靠开关)：头灯、尾灯以30Hz的频率闪烁。

3.设计目的

汽车在驾驶时有左转弯、右转弯、刹车、合紧急开关、停靠等操作。在左转弯或右转弯时，通过转弯操作杆应使左转开关或右转开关合上，从而使左头灯、仪表板左转弯灯、左尾灯或右头灯、仪表板右转弯灯、右尾灯闪烁；合紧急开关时要求前面所述的6个信号灯全部闪烁；汽车刹车时，两个尾灯点亮；如正当转

弯时刹车，则转弯时原应闪烁的信号灯仍应闪烁。以上闪烁，都是频率为1Hz的低频闪烁；在汽车停靠而停靠开关合上时，左头灯、右头灯、左尾灯、右尾灯按频率为30Hz的频率闪烁。通过编程和proteus仿真模拟汽车驾驶时信号灯的情况，以来更好的学习，了解单片机。

4.设计方案

## 4.1汽车驾驶操作与信号灯的明暗情况

汽车上有一个转弯控制杆，其中有三个位置：中间位置，汽车不转弯；向上，汽车左转；向下汽车右转。转弯时，规定左右尾灯、左右头灯仪表板上2个指示灯相应地发出闪烁信号。应急开关合上时， 6个信号灯都应闪烁。汽车刹车时，2个尾灯发出不闪烁信号。如正当转弯时刹车，转弯时原应闪烁的信号仍应闪烁。它们都是频率为1Hz低频闪烁，在汽车停靠而停靠开关合上时，左头灯、右头灯、左尾灯、右尾灯按频率为30Hz的高频闪烁。由上所述，各种情况作操作时，信

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 驾 驶 操 作 | 输 出 信 号 | | | | | |
| 仪表灯 | 仪表灯 | 左头灯 | 右头灯 | 左尾灯 | 右尾灯 |
| 左转弯（合上左转开关） | 闪烁 | —— | 闪烁 | —— | 闪烁 | —— |
| 右转弯（合上右转开关） | —— | 闪烁 | —— | 闪烁 | —— | 闪烁 |
| 合紧急开关 | 闪烁 | 闪烁 | 闪烁 | 闪烁 | 闪烁 | 闪烁 |
| 刹车（合上刹车开关） |  |  |  | —— | 亮 | 亮 |
| 左转弯时刹车 | 闪烁 | —— | 闪烁 | —— | 闪烁 | 亮 |
| 右转弯时刹车 | —— | 闪烁 | —— | 闪烁 | 亮 | 闪烁 |
| 刹车，并合紧急开关 | 闪烁 | 闪烁 | 闪烁 | 闪烁 | 亮 | 亮 |
| 左转弯时刹车，并合紧急开关 | 闪烁 | 闪烁 | 闪烁 | 闪烁 | 闪烁 | 亮 |
| 右转弯时刹车，并合紧急开关 | 闪烁 | 闪烁 | 闪烁 | 闪烁 | 亮 | 闪烁 |
| 停靠（合停靠开关） | —— | —— | 30Hz  闪烁 | 30Hz  闪烁 | 30Hz  闪烁 | 30Hz  闪烁 |

## 4.2单片机的系统设计

### 4.2.1开关状态检测

开关状态检测，对AT89S52来说是输入关系，可轮流检测每个开关状态，以每个开关的状态让相应的发光二极管指示，采用JNB P1.X，REL 指令来完成；也可以一次性检测五路开关状态，让它指示，可以用MOV A，P1 指令一次把P1 端口的状态全部读入，取低5位的状态来指示

### 4.2.2输出控制

以发光二极管D1—D6 来指示，此设计用SETB P0.X 和CLR P0.X 指令来完成，也可以用指令MOV P0，＃111XXXXXB 方法来实现。

### 4.2.3定时器和计数器

根据任务设计要求：会用到定时器。信号的控制是定时器与中断系统的联合使用得以实现。单片机的控制系统应用中，定时器是必需的，在汽车转弯灯的控制中也是必不可少。定时有三种选择方法。

(1)软件的定时

它是靠执行一个循环程序以进行时间的延迟。软件定时的优点是时间精确，且不需外加硬件电路。但它要增加CPU开销，因此软件定时的时间不能太长。此外，软件定时方法有时候无法使用。

(2)硬件的定时

时间较长的定时，常使用硬件电路完成。硬件定时方法的优点是定时功能全部由硬件电路完成，不需要占CPU的时间。用元件参数来调节定时时间，这方面使用上不够灵活方便。

(3)可编程定时器的定时

它是通过对系统时钟脉冲的计数来实现的。计数值由程序设定，改变计数值，同时也改变了定时时间，用起来既灵活且方便。此外，采用计数方法实现定时，可编程定时器都兼有计数功能，能对外来脉冲进行计数。

在AT89S52内部除了有并行和串行I/O接口外，在单片机内部共有2个可编程的定时器和计数器，称定时器/计数器0和定时器/计数器1，这两个计数器由TH0，TL0，TH1，TL1两个8位的RAM单元组成，即每个计数器都是16位的计数器，最大的计数量时65536。

### 4.2.4汽车转弯时灯显示

在汽车转弯或应急状态下，外部信号灯和仪表板它们指示灯的闪烁频率为1HZ，称低频信号。当停靠开关合上时，外部信号灯以30HZ频率闪烁此时为高频信号。

### 4.2.5汽车转弯灯控制

汽车转弯灯设计5个按键控制信号灯的转向、停靠、应急等。按键安排见下：

S1键为刹车开关；

S2键为紧急开关；

S3键为停靠开关；

S4键为左转弯开关；

S5键为右转弯开关。

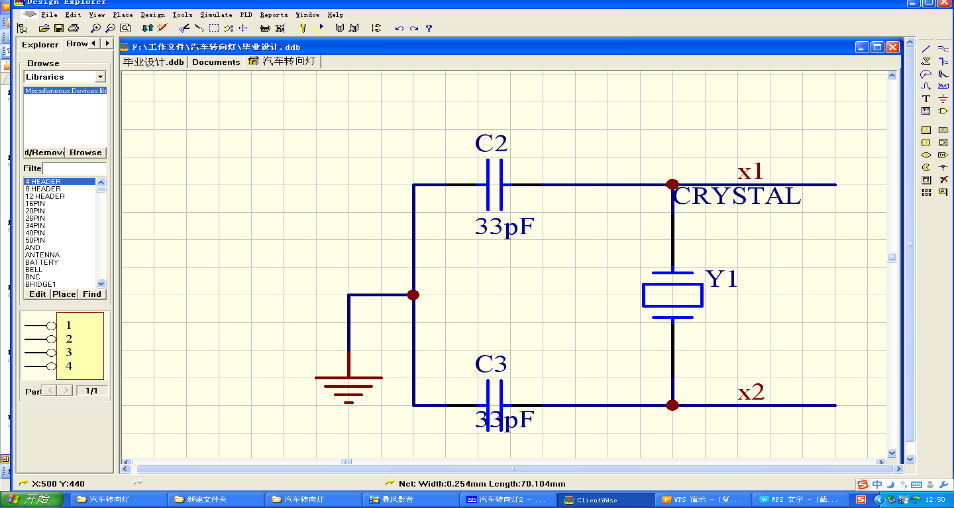
## 4.3设计方案

汽车转弯灯主要有单片机、按键、复位、时钟、电源、LED显示电路组成最基本的单片机系统。单片机本身的功能强大，汽车转弯灯的驱动用单片机本身的驱动来驱动。使得单片机的功能得到充分的运用。

# 5.硬件设计

## 5.1单片机控制系统中功能模块的设计

### 5.1.1时钟电路

采用单片机内部晶振。如下图所示。在MCS-51系列单片机内部有一个高增益反向放大器，其输入端为芯片引脚XTAL1，输出端为引脚XTAL2。而在芯片外部XTAL1和 XTAL2之间跨接晶体振荡器和微调电容，形成反馈电路，就构成了一个稳定的自激振荡器。外接晶体（石英或陶瓷，陶瓷的精度不高，但价格便宜）振荡器以及电容C1和C2构成并联谐振电路，接在放大器的反馈回路中， C1和C2的大小会对振荡器频率的高低、振荡器的稳定性、起振的快速性和温度特性有一定的影响。因此建议在采用石英晶体振荡器时C=30+/-10pF，陶瓷振荡器时，C=40+/-10pF，典型值为40pF。在设计电路板时，振荡器和电容应尽量安装得与单片机靠近，以减小寄生电容的存在更好的保障振荡器稳定、可靠的工作。

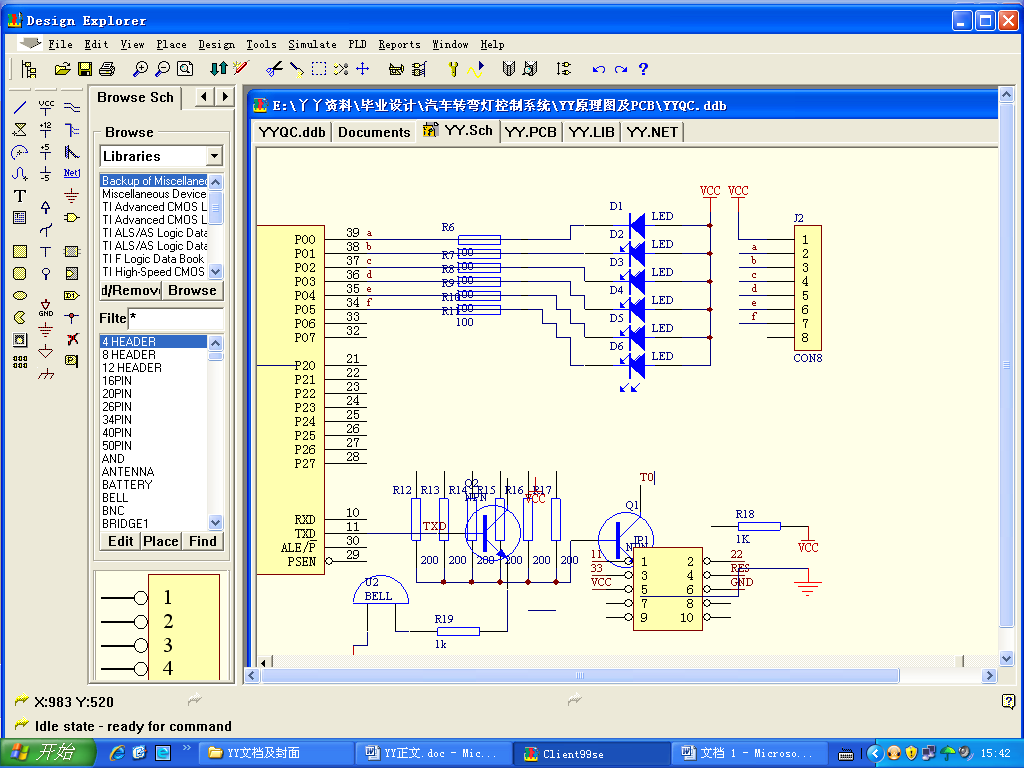
### 5.2.2信号灯电路

LED具有二极管的特性，但在导通之后会发光，称之为发光二极管。与普通的灯泡一样，LED导通后，随着其俩端电压的增加，电流急剧增加，所以，必须给LED串联一个限流电阻，否则一旦通电，LED会被烧坏。要用89S52单片机来控制LED，显然这个LED必须要与89S52单片机的某个脚相连。89S52单片机上除了基本连线必须用

图

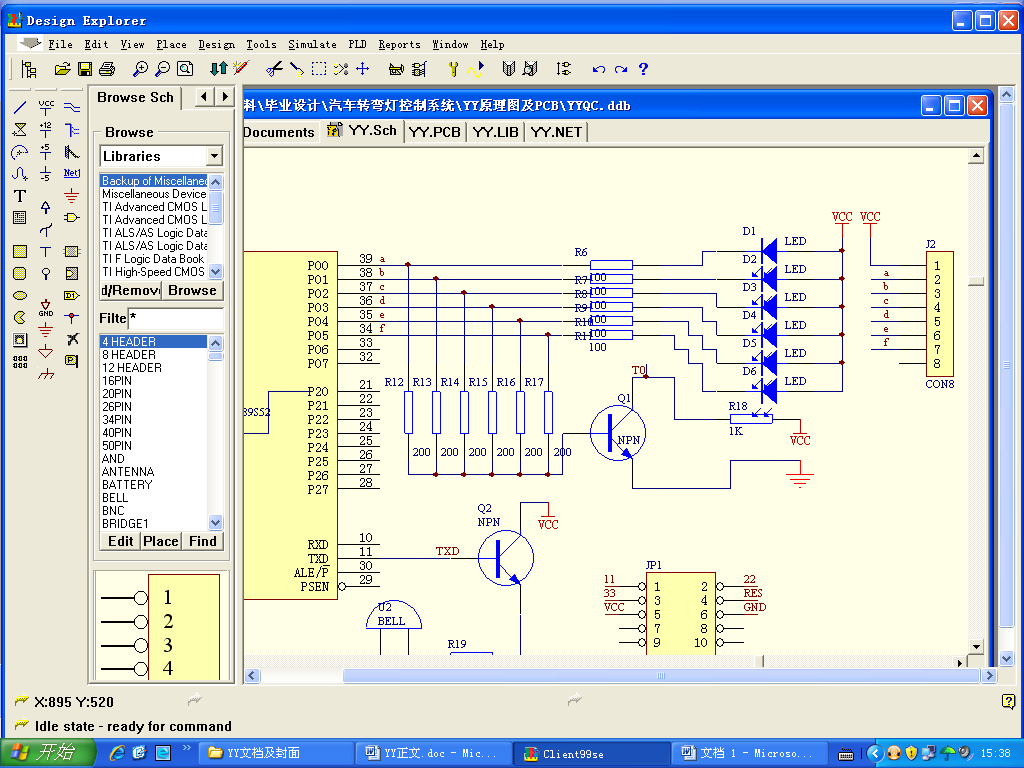
到的6个脚，还有34个引脚。这里把LED与89S52单片机的P0脚相连。本次设计中按照图4.11的接法，当AT89S52单片机的第1脚是高电平时，LED不亮，当第1脚是低电平时，LED亮。但是在汽车转弯灯里要根据汽车方向来控制信号

实现灯的亮暗。



### 5.2.3故障监控电路

如下图所示。是这类故障监控电路的方案之一，它利用T0作检测输入，只增加1个晶体管和几个电阻。假定其中一个信号灯是受控断开的（输出口线送高电平），而其余信号灯皆受控接通。这时晶体管Q1的6个输入端中有5个是低电平。



## 5.2元件清单

|  |  |  |  |
| --- | --- | --- | --- |
| LED | 7 | 限流电阻（100Ω） | 6 |
| 芯片插座（40PIN） | 1 | 电阻（1K） | 3 |
| 电源插针（2PIN） | 1 | 上拉电阻（5.1K） | 5 |
| 整流桥 | 1 | 电容（30pF） | 2 |
| 电阻（200Ω） | 2 | 电容（22μF） | 1 |
| 拨码开关 | 5 |

## 5.3单片机介绍

### 5.3.1单片机引脚介绍

(1) VCC：电源。

(2) GND：接地。

(3) P0口：是一个8位漏极开路的双向I/O口。

(4) P1口：是一个具有内部上拉电路的8位双向I/O口。

(5) P2口：是一个具有内部上拉电路的8位双向I/O口，P2口输出缓冲器能吸入/放出4个TTL输入。访问外部程序存储器及使用16位地址的是数据存储器（MOVX@DPTR）时，P2口输出高8位地址。这种情况下，当不置“1”时，P2口使用强大的内部上拉电路。访问使用8位地址的外部存储器（MOVX @RI）时，P2口输出P2口锁存器的内容。

(6) P3口：是一个具有内部上拉电路的8位双向I/O口，P3输出缓冲器能吸入/放出4个TTL输入。Flash编程及检验时，P3口也接收一些控制信号

(7) RST：复位端。当振荡器工作时，此时高电平将系统复位。

(8) ALE/PROG：当访问外部存储器时，ALE（允许地址锁存）是一个用于锁存地址的低8位字节的输出脉冲。

(9) PSEN：外部程序存储器读选取通信号。

（10) EA/Vpp：访问外部程序存储器允许端。

(11) XTAL1：振荡器反向放大器输入端和内部时钟发生器的输入端。XTAL2：振荡器反相放大器输出端。

### 5.3.2单片机硬件功能实现

1、利用单片机的P0口连接6个LED作为数据线输出使用；

2、利用单片机的P1口连接5个开关作为数据线输入使用；

3、定时器/计数器的使用

定时器/计数器0,定时功能,工作方式1,提供30ms的定时时间。

4、中断源的使用

定时器/计数器中断0。

5、按键设定

P1.0=刹车；P1.1=紧急；P1.2=停靠；P1.3=左转；P1.4=右转

6、显示设定

P0.0-P0.5控制线，接LED显示，信号灯顺序点亮是左前灯、右前灯、左仪表板、右仪表板、左后灯、右后灯。

# 6.软件分析

## 6.1汽车转弯系统流程图

### 6.1.1汽车转弯控制系统主流程图



### 6.1.2中断服务程序流程图



### 6.1.3控制系统功能键流程图



## 6.2程序说明

原理图按键连线：

P1.0=刹车；

P1.1=紧急；

P1.2=停靠；

P1.3=左转；

P1.4=右转；

说明：键值是根据P1的状态来确定的。

例P1=00011110，表明刹车键按下，它的键值是0x1e[4]（只看后面五位），程序清单见附录。

# 7.总结与展望

本次单片机课程设计中，无论是在程序设计方面还是实践动手能力都得到了一个非常大的锻炼，并对硬件知识的了解得以加深。

通过这次课程设计，也非常的清楚的认识了单片机这门课程的重要性，也意识到了自己在程序设计方面的薄弱。希望在以后的学习和工作中能进一部的加强自己专业素质和实践动手能力。

本次设计充分利用了51单片机的各引脚功能，同时有效利用了中断、查询、定时器、计数器，使得汽车转向信号灯控制得以实现。通过这次课程设计，我们对于单片机应用有了更深的了解，单片机应用技术发展迅速，有着广阔的应用前景，涉及面广，内容丰富。它用软件的方法设计硬件，并用软件方式设计的软件系统到硬件系统的转换是由有关的开发软件自动完成的。在设计过程中可用有关软件进行各种仿真，同时系统可现场编程，在线升级等。整个系统可集成在一个芯片上，体积小，功耗低，可靠性高。其技术以大规模可编程逻辑器件为设计载体，以硬件描述语言为系统逻辑描述的主要表达方法，以计算机、大规模可编程逻辑器件的开发软件及实验开发系统为设计工具，通过有关的开发软件，自动完成用软件方式设计的电子系统到硬件系统的逻辑编译，逻辑化简，逻辑分割，逻辑综合及优化，逻辑布局布线，逻辑仿真，直至特定目标芯片的适配便宜，逻辑映射，编程下载等工作，为系统的设计开发带来了极大地方便。

在课程设计过程中，我利用网络查找了大量资料，同时得到了指导老师的耐心指导。通过这次课程设计的实践，我对C语言的掌握有了更进一步的提高，在单片机的编程应用方面也更加熟练了。总的来说，这是一次利用专业基础知识解决实际问题的实践活动，我受益匪浅。特别是在程序的编写调试，以及仿真电路的绘制方面，使我得到了一次用专业知识、专业技能分析和解决问题的全面的系统的锻炼。这次课程设计使我在单片机的基本原理、单片机应用系统开发过程，以及在常用编程设计思路技巧的掌握方面都能向前迈了一大步，为日后嵌入式程序开发的学习打下了良好的基础。此次设计使我对单片机的认识加深了，知识的涵盖也广了，锻炼了动脑设计、动手完成的实践能力。

# 参考文献

[1]李广弟，朱月秀，王秀山.单片机基础［M］.北京：航空航天大学出版社，2000.147～156

[2]康华光，陈大钦.电子技术基础模拟部分[M].武汉：高等教育出版社，1998.57～116

[3]戴佳，戴卫恒.51单片机C 语言应用程序设计实例精讲[M].北京：电子工业出版社，2006.25～31

[5]徐爱钧，彭秀华。Keil Cx51 V7.0单片机高级语言编程与μVision2应用实践[M].北京：电子工业出版社，2006.133～187

[6] 吴凌燕,王剑舵,邵红等.基于protuce的汽车转向灯[J]计量与测试技术,2011,(11):56-59.

[7] 徐爱均.单片机原理实用教程-基于Proteus虚拟仿真[M].北京:电子工业出版社,2009.1.

# 附录：程序清单

#include <reg52.h>

unsigned char jianzhi;

unsigned char time; //记录中断次数，time=30时产生1HZ的

sbit LF=P0^0; //左前灯

sbit RF=P0^1; //右前灯

sbit LY=P0^2; //左仪表

sbit RY=P0^3; //右仪表

sbit LB=P0^4; //左后灯

sbit RB=P0^5; //右后灯

sbit P3\_1=P3^1;

sbit P3\_4=P3^4;

void main(void)

{

TMOD=0x01; //定时器0，方式1

TH0=(65536-30000)/256; //置初值，产生30ms定时

TL0=(65536-30000)%256;

EA=1;

ET0=1;

TR0=1;

P3\_1=0; //置成低电平

while(1)

{

if(jianzhi==0x1f) //检查按键是否按下，如果没有则关灯

{P1=0xff;

if(P3\_4==1)

{P3\_1=1;}

else

{P3\_1=0;}

}

}

}

void time0(void) interrupt 1

{

TH0=(65536-30000)/256;

TL0=(65536-30000)%256;

P1=0xff; //P1先送0xff,P1中保存是按键的值

jianzhi=P1; //暂存键值到jianzhi

jianzhi=jianzhi&0x1f; //因为高三位不用，去掉

switch(jianzhi)

{

case 0x1e: //刹车

{

LB=0;

RB=0;

}break;

case 0x1d: //紧急

{

if(time==15)

{

LF=!LF;

RF=!RF;

LY=!LY;

RY=!RY;

LB=!LB;

RB=!RB;

time=0;

}

else

{time=time+1;}

}break;

case 0x1B: //停靠

{

LF=!LF;

RF=!RF;

LB=!LB;

RB=!RB;

}break;

case 0x17: //左转弯

{

if(time==15)

{

LF=!LF;

LB=!LB;

LY=!LY;

time=0;

}

else

{time=time+1;}

}break;

case 0x0f: //右转弯

{

if(time==15)

{

RF=!RF;

RB=!RB;

RY=!RY;

time=0;

}

else

{time=time+1;}

} break;

case 0x16: //刹车左转弯

{

RB=0;

if(time==15)

{

LB=!LB;

LF=!LF;

LY=!LY;

time=0;

}

else

{time=time+1;}

}break;

case 0x0e: //刹车右转弯

{

LB=0;

if(time==15)

{

RB=!RB;

RF=!RF;

RY=!RY;

time=0;

}

else

{time=time+1;}

}break;

case 0x1c: //刹车紧急

{

LB=0;

RB=0;

if(time==15)

{

RF=!RF;

LF=!LF;

RY=!RY;

LY=!LY;

time=0;

}

else

{time=time+1;}

}break;

case 0x14: //左转刹车紧急

{

RB=0;

if(time==15)

{

LB=!LB;

RF=!RF;

RY=!RY;

LF=!LF;

LY=!LY;

time=0;

}

else

{time=time+1;}

}break;

case 0x0c: //右转刹车紧急

{

LB=0;

if(time==15)

{

RB=!RB;

RF=!RF;

RY=!RY;

LF=!LF;

LY=!LY;

time=0;

}else

{time=time+1;}

}break;

default:break;

}

}