**目录**

[摘 要 1](#_Toc16256)

[前言 1](#_Toc4795)

[一、 设计目标 2](#_Toc16566)

[二、 设计所用仪器和器件 2](#_Toc13233)

[1）处理器的选择 2](#_Toc18478)

[2）显示电路 2](#_Toc21980)

[3）芯片选择 2](#_Toc20754)

[三、 设计内容 3](#_Toc8611)

[四、 基本原理简介 3](#_Toc7302)

[① 8255A 3](#_Toc3207)

[② 8253 4](#_Toc26418)

[③0809 4](#_Toc12568)

[五、 设计过程 6](#_Toc30689)

[（一） 总体设计阶段 6](#_Toc24452)

[1． 设计方案（逻辑框图，程序流程图） 6](#_Toc14419)

[① 功能模块图 6](#_Toc29613)

[（二） 硬件设计 6](#_Toc20859)

[1. 设计系统的硬件电路图 6](#_Toc20151)

[① 硬件结构图 6](#_Toc25834)

[② 显示系统硬件电路 7](#_Toc10185)

[ 要求 7](#_Toc20018)

[ 分析 7](#_Toc20378)

[ 硬件设计 7](#_Toc26730)

[ 软件设计 8](#_Toc5645)

[③ 加速系统电路设计 9](#_Toc30559)

[ 要求 9](#_Toc23912)

[ 分析 9](#_Toc29500)

[ 硬件设计 9](#_Toc26663)

[ 软件设计 9](#_Toc31650)

[④ 声音系统电路设计 10](#_Toc27181)

[ 要求 10](#_Toc25736)

[ 分析 10](#_Toc8366)

[ 硬件设计 10](#_Toc13064)

[ 软件设计 10](#_Toc3661)

[⑤ 控制/模拟/灯光系统 12](#_Toc20854)

[ 要求 12](#_Toc9535)

[ 分析 12](#_Toc15751)

[ 硬件设计 12](#_Toc14197)

[ 软件设计 13](#_Toc18507)

[⑥ 完整的电路 13](#_Toc1301)

[（三） 软件设计 13](#_Toc32577)

[1. 设计代码流程 13](#_Toc17361)

[① 程序流程图 13](#_Toc20182)

[1) 主函数的流程图 13](#_Toc3546)

[2） 一档函数的流程图 14](#_Toc12807)

[2） 油门函数流程图 14](#_Toc23936)

[3） 喇叭流程图 15](#_Toc8202)

[2． 根据硬件电路图编写软件代码，在 Proteus 环境下调试仿真 15](#_Toc8893)

[① //主函数 15](#_Toc12316)

[② //左转向灯 16](#_Toc10821)

[③//油门函数 17](#_Toc27884)

[④//监视油门 18](#_Toc2501)

[⑤//倒车函数 18](#_Toc23612)

[3． 观测并记录仿真实验结果（要求客观、真实） 19](#_Toc9087)

[（四）系统使用说明书 21](#_Toc9061)

[① 声明 21](#_Toc19165)

[② 运行环境 21](#_Toc10843)

[③ 用户可以操作的模块介绍 21](#_Toc2078)

[④显示模块 22](#_Toc16464)

[⑤ 用户可调节模块 24](#_Toc11336)

[六、 分析与讨论 24](#_Toc3131)

[1． 步进电机运行速度的控制 24](#_Toc2446)

[2. 喇叭的声音比较刺耳，非常难听 24](#_Toc23164)

[3. 编译器会出现问题 25](#_Toc20518)

[4. 八段数码管问题 25](#_Toc11984)

[七、 个人体会与小结 25](#_Toc12787)

[八、 参考材料 26](#_Toc16596)

摘 要

随着电子技术的发展，计算机在现代科学技术的发展中起着越来越重要的

作用。多媒体技术、网络技术、智能信息处理技术、自适用控制技术、数据挖掘与处理技术等都离不开计算机。本课程设计是基于微机原理与接口技术的简单应用。运用所学的微机原理和接口技术知识完成模拟小车控制系统。

设计一个模拟汽车控制系统，用简单逻辑电平控制车的档位（前进1，前进2，前进3，后退），用步进电机的转速模拟车的速度（若步进电机不能用，也可以用流水灯的刷新方向和速度，或数码管显示的数值来模拟车的速度），用A/D转换器模拟车的油门大小。

关键词： 模拟小车，微机接口，数码管，步进电机，A/D转换

**前言**

课程设计是让自己动手动脑，亲手设计与调试的。它将基本技能训练、基本工艺知识和创新启蒙有机结合，培养我们的实践和创新能力。课程设计的意义，不仅仅是让我们把所学的理论知识与实践相结合起来，提高自己的实际动手能力和独立思考的能力。作为信息时代的大学生，基本的动手能力是一切工作和创造的基础和必要条件。

课程设计是培养和锻炼学生在学习完本门课后综合应用所学理论知识解决实际工程设计和应用问题的能力的重要教学环节，它具有动手、动脑和理论联系实际的特点，是培养在校工科大学生理论联系实际、敢于动手、善于动手和独立自主解决设计实践中遇到的各种问题能力的一种较好方法。

《微机原理及应用》是一门应用性、综合性、实践性较强的课程，没有实际的有针对性的设计环节，学生就不能很好的理解和掌握所学的技术知识，更缺乏解决实际问题的能力。所以通过有针对性的课程设计，使学生学会系统地综合运用所学的理论知识，提高学生在微机应用方面的开发与设计本领，系统的掌握微机硬软件设计方法。

通过课程设计实践，不仅要培养学生的实际动手能力，检验学生对本门课学习的情况，更要培养学生在实际的工程设计中查阅专业资料、工具书或参考书，掌握工程设计手段和软件工具，并能以图纸和说明书等表达设计思想和结果的能力。培养学生事实求是和严肃认真的工作态度。

通过设计过程，要求学生熟悉和掌握微机系统的设计的方法、设计步骤，使学生得到微机开发应用方面的初步训练。让学生独立或集体讨论设计题目的系统方案论证设计、编程、软件调试、查阅资料、编写说明书等问题，真正做到理论联系实际，提高动手能力和分析问题、解决问题的能力，实现由学习知识到应用知识的初步过渡。通过本次课程设计使学生熟练的熟练掌握微机系统的设计方法，熟练应用C语言或8086汇编语言编写应用程序和实际设计中的软件调试方法和步骤，熟悉微机系统的软件开发工具的使用方法。

通过《微机原理与接口》课程设计，使学生能够进一步了解微型计算机工作原理, 微型计算机的硬件结构及微型计算机软件编程。 要求学生根据接口电路的硬件要求进行计算机的C语言或汇编语言程序设计，使学生的软件编程能力得到加强,对接口电路的综合应用能力有较大提高。  利用所学微机的理论知识进行软硬件整体设计，提高综合应用能力。通过设计掌握综合使用基本输入设备，通用接口芯片，专用接口芯片的方法；熟悉并行接口芯片8255的使用与硬件接口方法，熟悉8255的各种工作方式，掌握8255的编程方法；掌握实时处理程序的编制和调试方法。从而帮助学生系统地掌握微机原理的接口技术的相关知识，达到将知识融会贯通的目的。同时给学生提供更多、更好的训练机会，培养学生的创新能力。

1. **设计目标**
2. 设计一个模拟汽车控制系统，有档位（前进1，前进2，前进 3，后退），油门大小，喇叭，转向灯等基本功能。

2、 充分利用所学芯片（如8253,8255等）和已经讲过的例题。

3、 熟练掌握各种芯片的基本原理和使用方法，了解如何去使用一款以前从未接触过的可编程接口芯片及其他元件。

1. **设计所用仪器和器件**

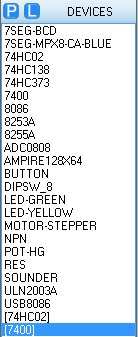
****

图 1 元件清单

此次设计是在proteus软件上仿真，用digital Mars C 编译器。

1）处理器的选择

微型机具有体积小、重量轻、耗电少、价格低廉、可靠性高、结构灵活等特点，所以选择8086系统

2）显示电路

显示可通过彩灯和数码管来实现。如果用彩灯作为显示功能，则不是很直观。而数码管具有显示亮度高，使用寿命长，且能直观方便的看到倒计时数字，和选手编号，因此选用数码管显示。用Led-red表示车尾灯，转向灯。

3）芯片选择

8255作为并行I/0接口，能满足10个输入按键的输入功能，和LED数码管的输出。并且8255的每个接口还有锁存和数据缓冲作用。所以选择8255芯片。

因为需要一个合适的脉冲来驱动蜂鸣器，所以声音系统的输出，采用8253芯片。

又因为需要即可以控制一定范围的数字信号，所以才用0809进行A/D转换，实现油门的功能。

还有373锁存器，38译码器等必要芯片。

1. **设计内容**

1、 设计一个模拟汽车控制系统，用简单逻辑电平控制车的档位（前进 1，前进2，前进3，后退），用步进电机的转速模拟车的速度（也可以用数码管显示的数值来模拟车的速度），用 A/D 转换器模拟车的油门大小。

2、按键选项

内容包括：①1234四个档位，倒档；②油门；③喇叭

3、汽车档位

选定档位之后，通过步进电机模拟汽车的行驶，通过数码管显示相应的档位的提示信息。

4、油门

选定油门之后，通过数码管闪烁显示加速度信息，步进电机的速度也会发生改变

5、喇叭

按下喇叭之后，会通过蜂鸣器发出声音。

6、主要使用接口芯片8253（定时控制器芯片），8255（并口芯片），0809（AD转换），以及38译码器和373芯片。

1. **基本原理简介**

**① 8255A**

8255A是通用的8位并行输入/输出接口芯片，使用灵活，功能强大，是应用最广的典型可编程并行接口芯片。共有40个引脚，采用双列直插式封装，各引脚功能如下：与微处理器连接的信号线     D7～D0：数据线，三态双向8位，与系统的数据总线相连。     CS：片选信号，低电平有效。     WR：写信号，低电平有效。     RD：读信号，低电平有效。     A1,A0：端口地址选择信号。用于选择8255A的3个数据端口和一个控制端口。      8255A与外部设备连接的信号线     PA7～PA0：A口数据线。     PB7～PB0：B口数据线。     PC7～PC0：C口数据线

引脚图如下：

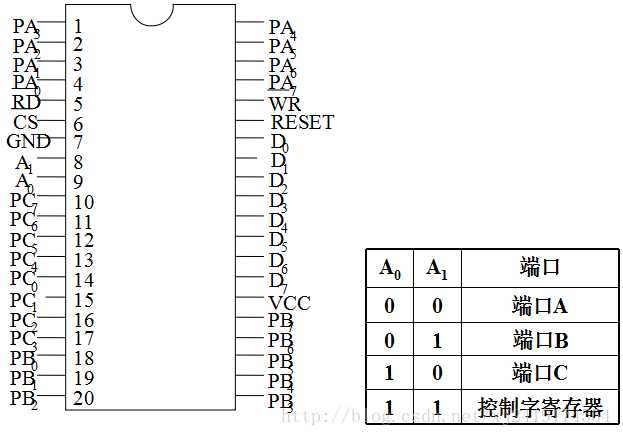


图 2 8255原理图

② 8253

8253芯片是具有24个引脚的双列直插式集成电路芯片，其引脚分布如下图所示。

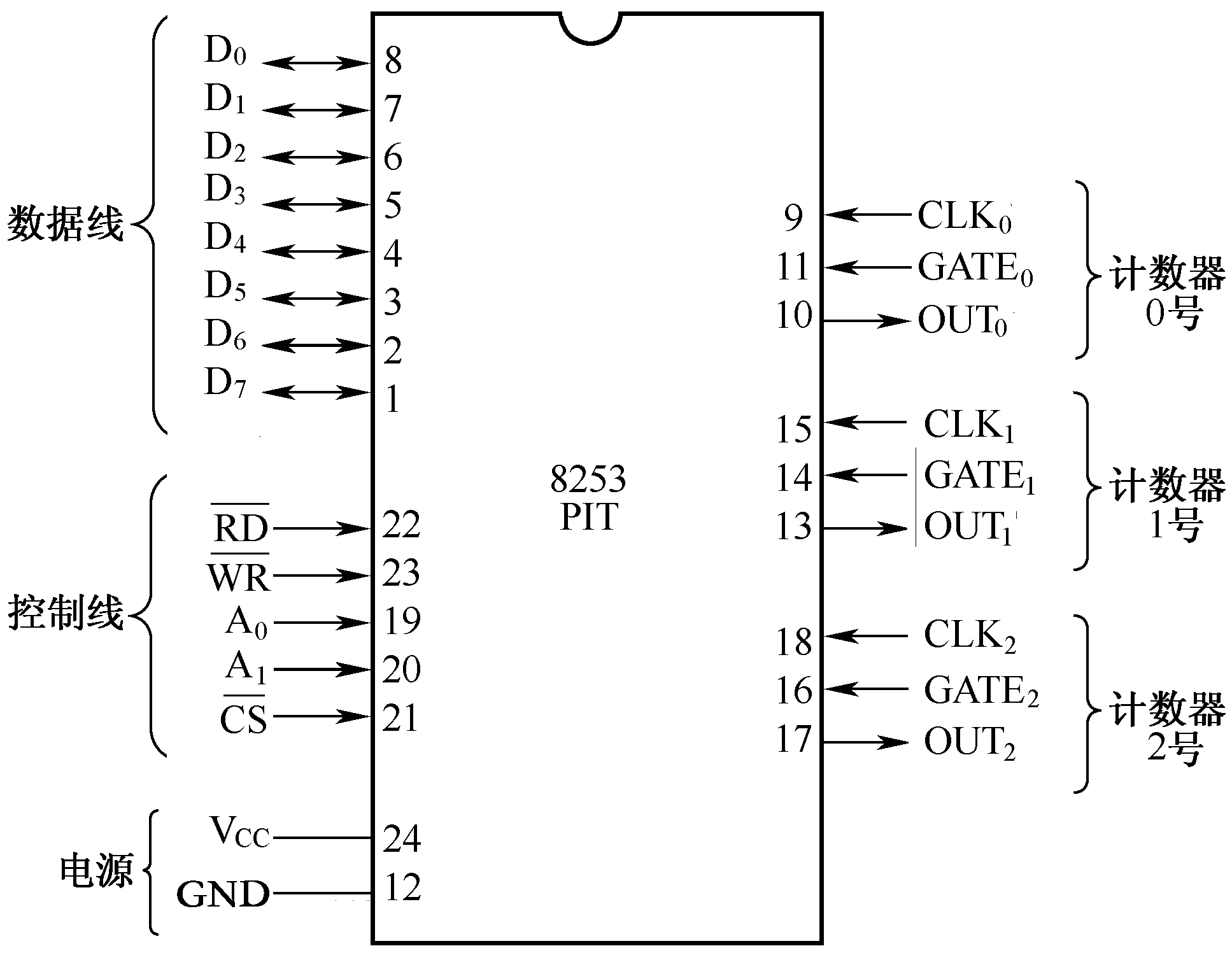


图 3 8253原理图

8253的工作方式3：方波速率发生器。

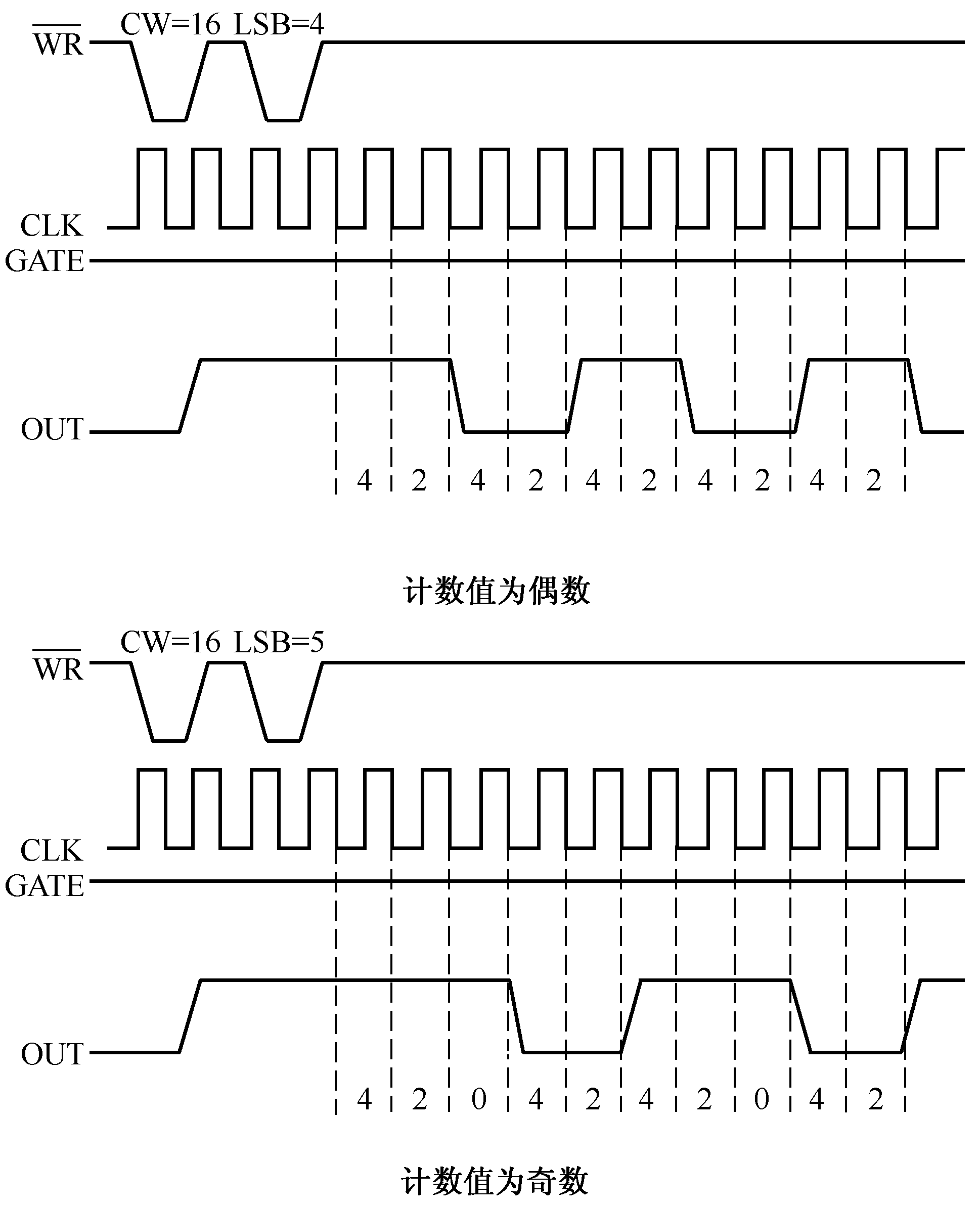


图 4 方式3的时序关系

与方式2类似，只是进行减2操作，直到0时， OUT变低，并自动重新写入计数值减2操作，直到0时， OUT变高。一般作为方波发生器使用。

③0809

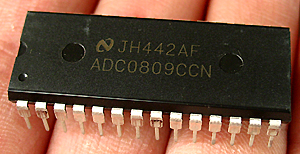
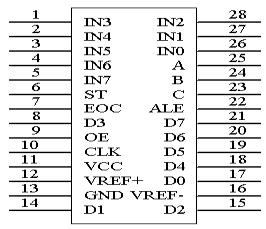


图 5 0809外部引脚

IN0－IN7：8条模拟量输入通道

    ADC0809对输入模拟量要求：信号单极性，电压范围是0－5V，若信号太小，必须进行放大；输入的模拟量在转换过程中应该保持不变，如若模拟量变化太快，则需在输入前增加采样保持电路。

地址输入和控制线：4条

    ALE为地址锁存允许输入线，高电平有效。当ALE线为高电平时，地址锁存与译码器将A，B，C三条地址线的地址信号进行锁存，经译码后被选中的通道的模拟量进转换器进行转换。A，B和C为地址输入线，用于选通IN0－IN7上的一路模拟量输入。通道选择表如下表所示。

|  |  |  |  |
| --- | --- | --- | --- |
| C | B | A | 选择的通道 |
| 0 | 0 | 0 | IN0 |
| 0 | 0 | 1 | IN1 |
| 0 | 1 | 0 | IN2 |
| 0 | 1 | 1 | IN3 |
| 1 | 0 | 0 | IN4 |
| 1 | 0 | 1 | IN5 |
| 1 | 1 | 0 | IN6 |
| 1 | 1 | 1 | IN7 |

数字量输出及控制线：11条

    ST为转换启动信号。当ST上跳沿时，所有内部寄存器清零；下跳沿时，开始进行A/D转换；在转换期间，ST应保持低电平。EOC为转换结束信号。当EOC为高电平时，表明转换结束；否则，表明正在进行A/D转换。OE为输出允许信号，用于控制三条输出锁存器向单片机输出转换得到的数据。OE＝1，输出转换得到的数据；OE＝0，输出数据线呈高阻状态。D7－D0为数字量输出线。

CLK为时钟输入信号线。因ADC0809的内部没有时钟电路，所需时钟信号必须由外界提供，通常使用频率为500KHZ，

VREF（＋），VREF（－）为参考电压输入。

1. **设计过程**
2. 总体设计阶段

本阶段的任务是通过调查研究，查阅资料来初步确定系统结构的总体方案，其中主要涉及到硬件和软件的功能划分。

应用系统中硬件和软件具有一定的互换性，即某些功能既可以用硬件实现也可以用软件实现。一般说来，用硬件实现的优点是可以提高系统的工作速度，但会增加电路的复杂性和硬件成本；而用软件代替某些硬件的功能可以使电路简化，便于修改设计，降低硬件成本，但软件工作量增大。

总体设计时，必须在硬件和软件之间权衡，确定方案，分工明确，然后分头开始设计

1. **设计方案（逻辑框图，程序流程图）**

采取以8255为主的设计方案，通过8255连接开关，控制步进电机的转速，控制指示灯的闪烁，通过再加一片8255控制数码管的显示；用8253产生方波，用来是蜂鸣器发声；通过0809将电压电流等模拟量转化成数字信号，用来作为加速度，即控制油门的大小。

**① 功能模块图**

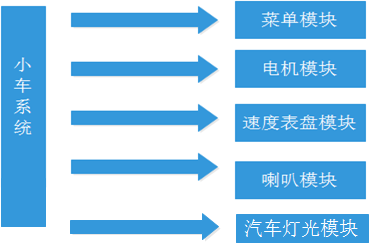
****

图 6 功能模块

1. 硬件设计

利用实验箱的硬件资源，结合课程设计题目，设计硬件原理图，搭建硬件电路。因为此次课程设计不强制要求在试验箱上做，只要软件仿真能通过即可，因此可以发挥的余地更大，尽少的减少实验箱的限制。

**1. 设计系统的硬件电路图**

**① 硬件结构图**

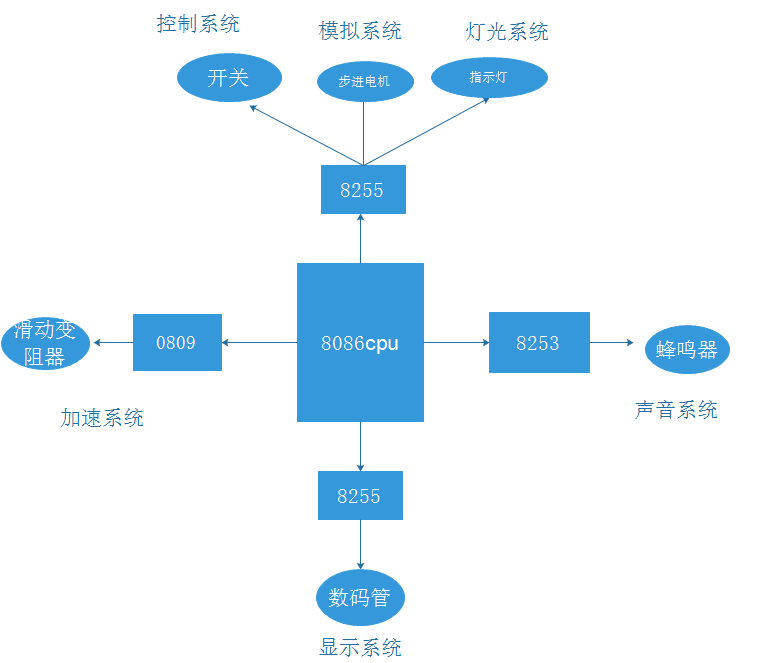
****

图 7 硬件结构图

**② 显示系统硬件电路**

* **要求**

显示一些小车系统的信息，比如速度，加速度等。

* **分析**

1. **LED 数码显示原理**

七段 LED 显示器内部由七个条形发光二极管和一个小圆点发光二极管组成，根据各 管的极管的接线形式，可分成共阴极型和共阳极型。 LED 数码管的 g~a 七个发光二极管因加正电压而发亮，因加零电压而不以发亮，不同亮暗的组合就能形成不同的字形，这种组合称之为字形码，下面给出共阳极的字形码 见下表

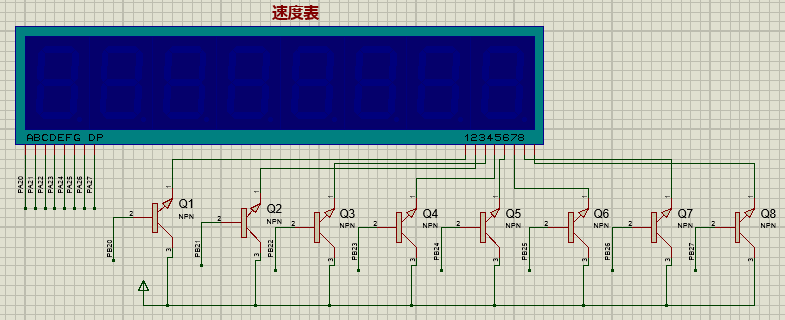
表格 1 数码管显示对应表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **“0”** | **0C0H** |  | **“8”** | **80H** |
| **“1”** | **0F9H** |  | **“9”** | **90H** |
| **“2”** | **0A4H** |  | **“A”** | **88H** |
| **“3”** | **0B0H** |  | **“b”** | **80H** |
| **“4”** | **99H** |  | **“C”** | **0B6H** |
| **“5”** | **92H** |  | **“d”** | **0B0H** |
| **“6”** | **82H** |  | **“E”** | **86H** |
| **“7”** | **F8H** |  | **“F”** | **8EH** |

1. **段码表格**

由于显示的数字 0－9 的字形码没有规律可循，只能采用查表的方式来完成我们所需的要求了。这样我们按着数字 0－9 的顺序，把每个数字的笔段代码按顺序排好！建 立的表格如下所示：TABLE DB 0c0h,0f9h,0a4h,0b0h,99h,92h,82h,0f8h,80h,90h

* **硬件设计**



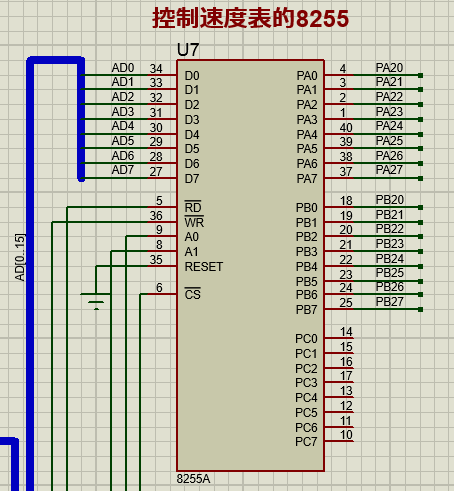


图 8 显示部分电路图

8255的A口接数码管的左边8个引脚，B口接数码管的右边8个引脚.8255的其余引脚按照其外特性进行连接。

* 软件设计

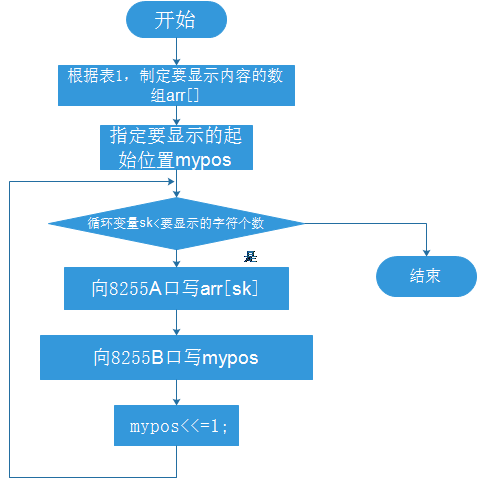


图 9 数码管驱动程序流程图

数码管驱动程序（局部）如下：

for(int sk=0;sk<3;sk++)

{

outp(IOA8255,daoche[sk]);//A口控制现实的数字

outp(IOB8255,mypos);//b口控制显示的位置

mypos<<=1;

}

③ 加速系统电路设计

* 要求

通过AD转换，获得一个数字量来表示加速度

* 分析

首先，我们有必要先了解一下ADC0809的外部特性，可以查看“基本原理简介”中的ADC0809的介绍。

A/D 转换器大致有三类：一是双积分 A/D 转换器，优点是精度高，抗干扰性好， 价格便宜，但速度慢；二是逐次逼近 A/D 转换器，精度、速度、价格适中；三是并行 A/D 转换器，速度快，价格也昂贵。

本次实验用的 ADC0809 属第二类，是 8 位 A/D 转换器，每采集一次一般需 s µ 100 。本 实验可采用延时方式或查询方式读入 A/D 转换结果，也可以采用中断方式读入结果，在 中断方式下，A/D 转换结束后会自动产生 EOC 信号，将其与 CPU 的外部中断相接。调 整电位计，得到不同的电压值，转换后的数据通过发光二级管和数码管输出

* 硬件设计

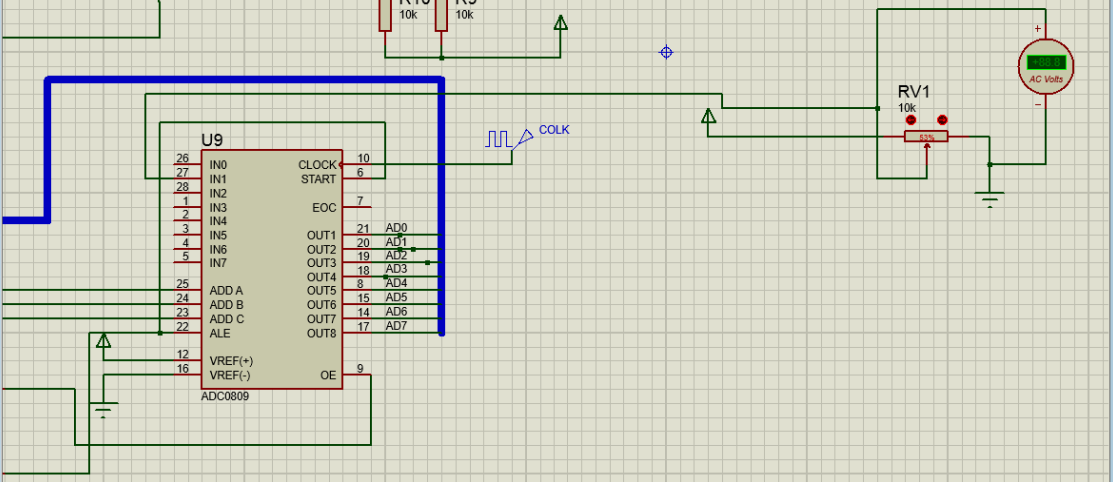


图 10 AD转换

IN1作为模拟号的输入，接一个电流源，0809的其余引脚按照其外特性连接即可。

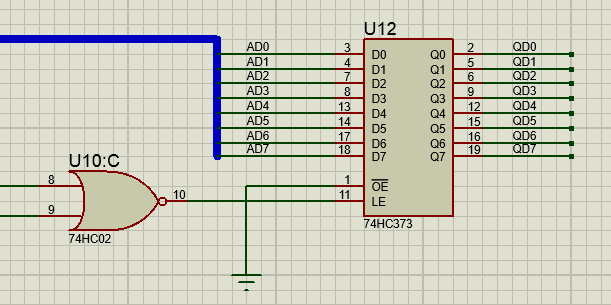


图 11 AD转换结果输出

因为要向373写入，所以为373分配了一个地址，因为373没有cs#信号，所以通过一个或非门，对cpu的写信号和地址信号进行或非实现cs#的功能。

* 软件设计

程序流程图见“图 12 油门函数流程图”；程序代码见软件设计部分的“**③//油门函数**”

**④ 声音系统电路设计**

* **要求**

通过8253，输出一个频率适中的波形，驱动蜂鸣器发出声音。

* **分析**

**8253的外部特性在**“基本原理简介”中有介绍

* **硬件设计**

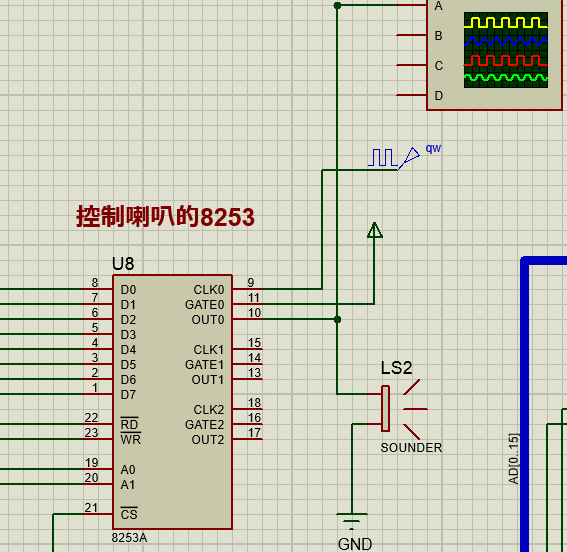


图 13 声音系统

* **软件设计**

流程图见“图 14 喇叭函数流程图”；

驱动程序如下：

**//此处省略了乐曲的频谱部分**

//初始化8253

outp(TCONTRO,0x36);//计数器0，只写计算值低8位，方式3，二进制计数

//初始化速度显示表8255

outp(IOCON8255,0x80);

while(1) //无限循环

{

i=0; //从第1个音符f[0]开始播放

unsigned char button=inp(IOA);

if(watchLaba(button))

{

outp(TCONTRO,0x10);

break;

}

//显示“laba”，提醒倒车

unsigned char mypos=pos;

for(int sk=0;sk<4;sk++)

{

outp(IOA8255,laba[sk]);//A口控制现实的数字

outp(IOB8255,mypos);//b口控制显示的位置

if(watchLaba(button))

{

outp(TCONTRO,0x10);

break;

}

mypos<<=1;

}

if(watchLaba(button))

{

outp(TCONTRO,0x10);

break;

}

while(cf[i]!=0xff) //只要没有读到结束标志就继续播放

{

//如果按下了其他按键就停止

button=inp(IOA);

if(watchLaba(button))

{

outp(TCONTRO,0x10);

break;

}

ch=cf[i]/256;

cl=cf[i]%256;

outp(TCON0,cl);

outp(TCON0,ch);

for(j=0;j<JP[i];j++) //控制节拍数

{

button=inp(IOA);

if(watchLaba(button))

{

outp(TCONTRO,0x10);

break;

}

delayLaba(); //延时1个节拍单位

}

i++; //播放下一个音符

}

button=inp(IOA);

if(watchLaba(button))

{

outp(TCONTRO,0x10);

break;

}

}

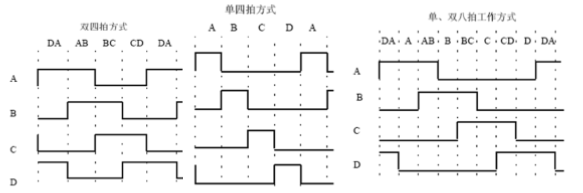
⑤ 控制/模拟/灯光系统

* 要求

通过开关对整个小车系统进行控制，用步进电机模拟车轮的转动，用led灯来模拟车的尾灯，转向灯等。

* 分析

步进电机驱动原理是通过对每组线圈中的电流的顺序切换来使电机作步进式旋转。切换 是通过单片机输出脉冲信号来实现的。所以调节脉冲信号的频率就可以改变步进电机的转速， 改变各相脉冲的先后顺序，就可以改变电机的转向。步进电机的转速应由慢到快逐步加速。 电机驱动方式可以采用双四拍（AB →BC→CD→DA→AB方式，也可以采用单四拍（A→B→C→D→A）方式。为了旋转平稳，还可以采用单、双八拍方式（ A→AB→B→B→BC→C→CD→D→DA→A）。各种工作方式的时序图如下：（高电平有效）：



上图中示意的脉冲信号是高电平有效，但实际控制时公共端是接在 VCC 上，所以实际 控制脉冲是低电平有效。

* 硬件设计

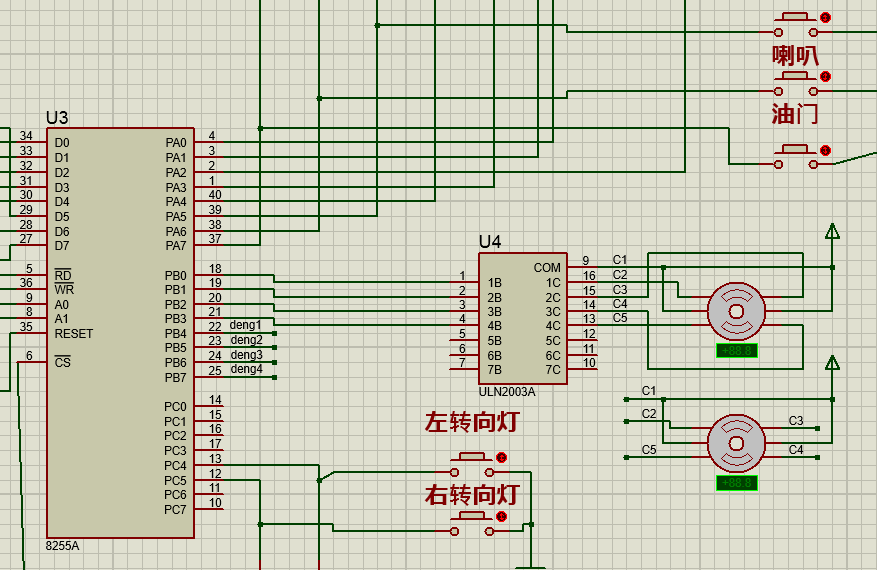


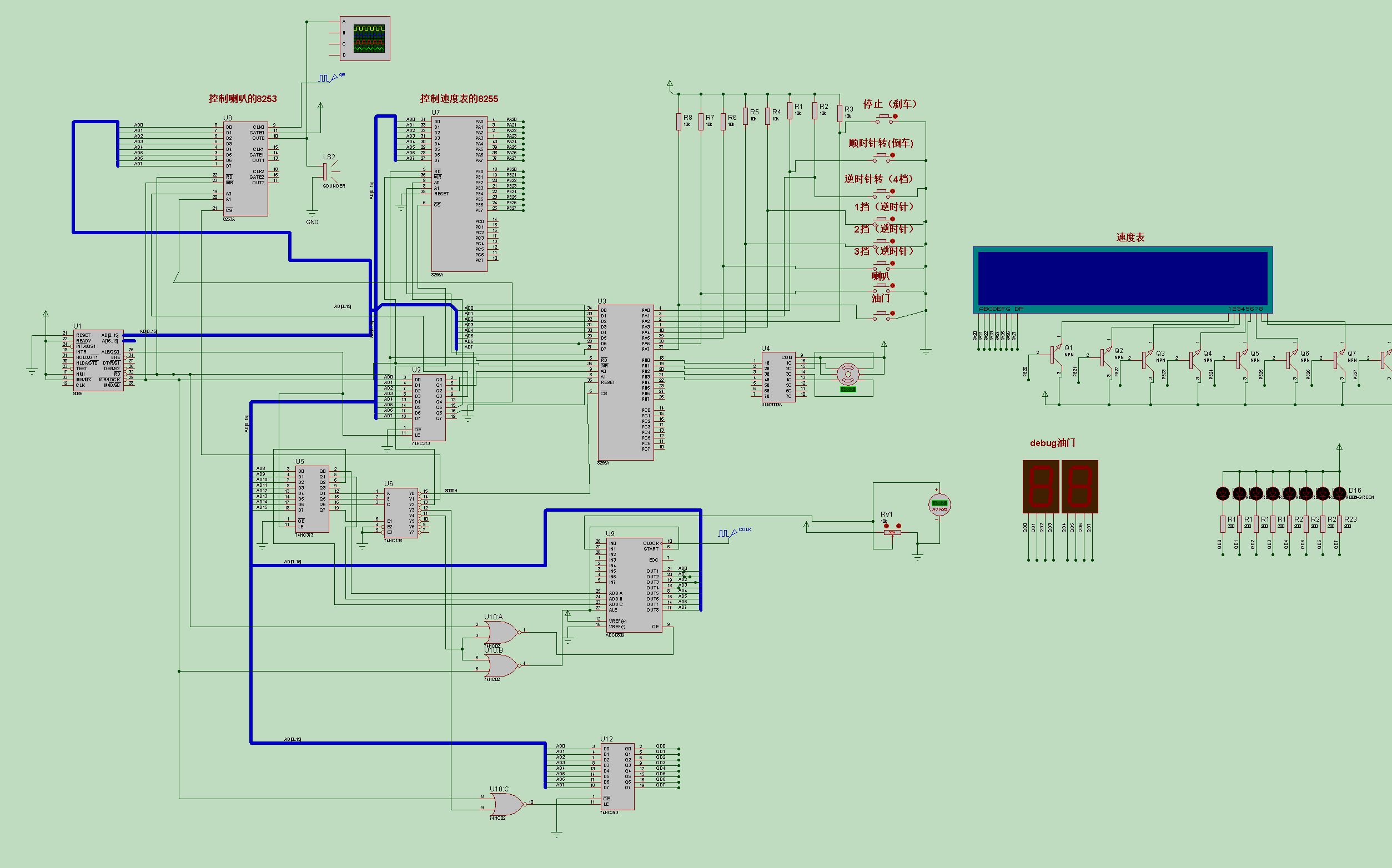
图 15 三大系统

A组作为输入，接开关作为控制信号；B组作为输出，控制步进电机和灯光。其余各个引脚按其外部特性进行连接。

* 软件设计

流程图见“图 16 一档等几个档位”；程序见（三）软件设计“**② //左转向灯**”和“**⑤//倒车函数**”等

⑥ 完整的电路

****

**声明：因为纸张太小，就是放大也可能比较模糊，所以在proteus工程所在目录下查看“步进电机.pdf”文件查看细节。**

1. **软件设计**
2. 根据设计的硬件图纸，采用模块化程序结构设计软件，可将整个软件分成若干功能模块。
3. 画出程序流程图。
4. 根据流程图，编写源程序。

**1. 设计代码流程**

**① 程序流程图**

1. **主函数的流程图**

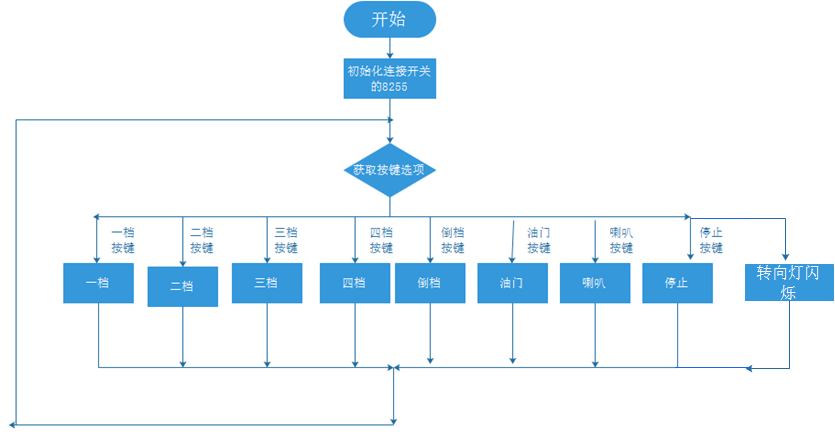
****

图 17 系统程序流程图

2） 一档函数的流程图

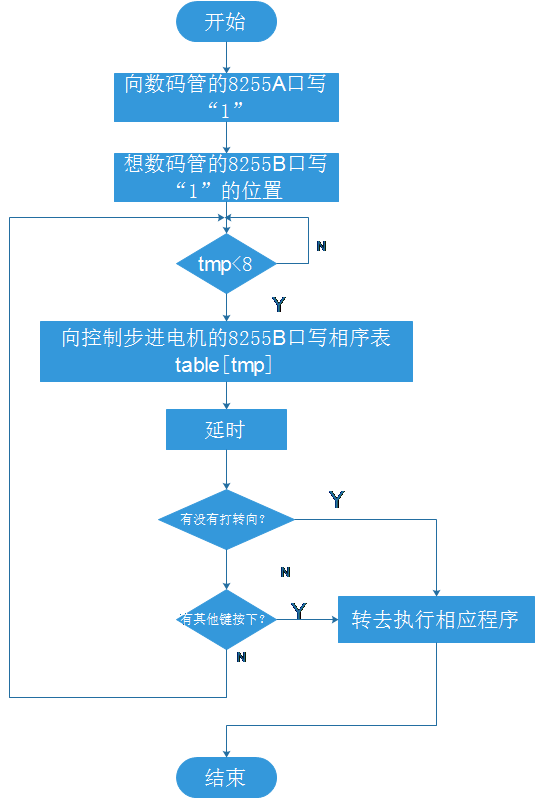


图 18 一档等几个档位

因为1、2、3、4、倒着五个档位是一样的，这里只画出1档的流程图。

1. 油门函数流程图

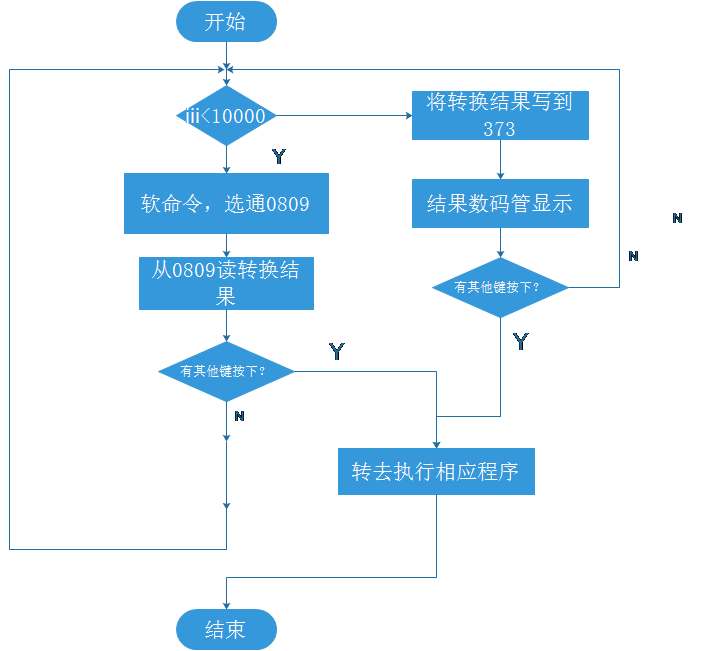


图 19 油门函数流程图

1. 喇叭流程图

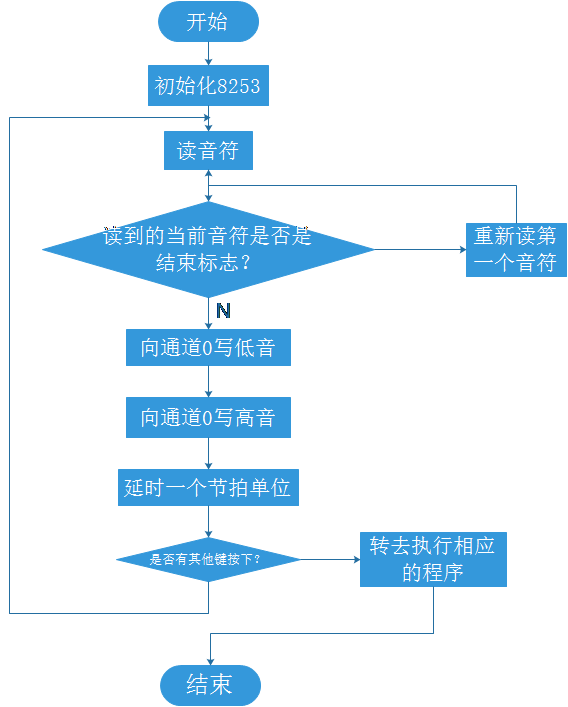


图 20 喇叭函数流程图

1. **根据硬件电路图编写软件代码，在 Proteus 环境下调试仿真**

主要代码如下：

**① //主函数**

从主函数可以看出，从程序运行后开始，初始化连有菜单的8255芯片，等待用户按键，以相应响应的事件。具体流程，见图 21 系统程序流程图。

**void main(void)**

**{**

**unsigned char button,i,tmp;**

**outp(IOCON,0x99);**

**i=0; button=0;**

**while(1)**

**{**

**//从A口都如数据**

**button=inp(IOA);**

**turn=inp(IOC);**

**//如果逆时针转(4档)**

**if(button==0xfd){sidang();}**

**//顺时针转**

**if(button==0xfe){backdang();}**

**//按下停止(熄火，刹车)**

**if(button==0xfb)**

**{**

**//初始化速度表的8255芯片**

**outp(IOA8255,zero);//A口控制现实的数字**

**outp(IOB8255,pos);//b口控制显示的位置**

**outp(IOB,0xf0);**

**}**

**//一档**

**if(button==0xf7){yidang();}**

**//二档**

**if(button==0xef){erdang();}**

**//三档**

**if(btnsandang==button){sandang();}**

**//喇叭**

**if(btnlaba==button){pressLaba();}**

**//油门**

**if(btnyoumen==button)**

**{**

**//我的AD转换，油门**

**myYoumenAcc=youmen();**

**}**

**}**

**}**

**② //左转向灯**

在汽车运行期间，读取c口是否有键按下，如果转向相关的键按下之后，转向灯闪烁。

**void turnLeft(unsigned char tmp)**

**{**

**//读取转向**

**turn=inp(IOC);**

**if(turn==0xef)**

**{**

**for(int sk=0;sk<50;sk++)**

**{**

**outp(IOB,0x40|table2[tmp]);**

**for(int j=0;j<20;j++){}**

**outp(IOB,table2[tmp]);**

**}**

**}**

**}**

**③//油门函数**

就是通过AD转换来获取一定范围的数据，作为加速度，改变速度。

**unsigned char youmen()**

**{**

**unsigned char button;**

**unsigned char in;**

**while(1)**

**{**

**for(char iii=10000;iii>0;iii--)**

**{**

**//从A口读入按键数据**

**button=inp(IOA);**

**if(watchYoumen(button))**

**{**

**return in;**

**break;**

**}**

**outp(AD0809,0);**

**in=inp(AD0809);**

**}**

**outp(OUT373,in);**

**myYoumenAcc=in;**

**//从A口读入按键数据**

**button=inp(IOA);**

**if(watchYoumen(button))**

**{**

**return in;**

**break;**

**}**

**//用数码管显示**

**/\*(inp(my373))\*/**

**unsigned char tempNum=(unsigned char)in/10;**

**unsigned char weishu[2];**

**weishu[0]=tempNum/10;**

**weishu[1]=tempNum%10;**

**unsigned char mypos=pos<<2;**

**for(unsigned char kkk=0;kkk<2;kkk++)**

**{**

**outp(IOA8255,number[weishu[kkk]]);//A口控制现实的数字**

**outp(IOB8255,mypos);//b口控制显示的位置**

**//从A口读入按键数据**

**button=inp(IOA);**

**delayYoumen();**

**mypos<<=1;**

**if(watchYoumen(button))**

**{**

**return in;**

**break;**

**}**

**}**

**}**

**return in;**

**}**

**④//监视油门**

判断是否有除油门外的其他按键按下

**unsigned char watchYoumen(unsigned char button)**

**{**

**if(btnlaba==button || btnsandang==button || button==0xef || button==0xf7 || button==0xfb || button==0xfe || button==0xfd)**

**{**

**return 1;**

**}**

**return 0;**

**}**

**void delayYoumen()**

**{**

**unsigned char m;**

**for(int i=0;i<350;i++)**

**{**

**m=inp(IOA);**

**if(m==0xfe||m==0xfb || m==0xfd || m==0xef || btnsandang==m || btnlaba==m || 0xfe==m || 0xf7==m)**

**{**

**break;**

**}**

**}**

**}**

**⑤//倒车函数**

**控制步进电机顺时针旋转，并且让数码管显示“dao”提示正在倒车。**

**void backdang()**

**{**

**unsigned char button,i,tmp;**

**outp(IOCON8255,0x80);**

**while(1)**

**{**

**//显示“dao”，提醒倒车**

**unsigned char mypos=pos;**

**for(int sk=0;sk<3;sk++)**

**{**

**outp(IOA8255,daoche[sk]);//A口控制现实的数字**

**outp(IOB8255,mypos);//b口控制显示的位置**

**if(button==0xfd||button==0xfb || button==0xf7 || i==0xef || btnsandang==button ||btnlaba==button || btnyoumen==button)break;**

**delayDao();**

**mypos<<=1;**

**}**

**for(tmp=0;tmp<8;tmp++)**

**{**

**outp(IOB,table1[tmp]);**

**outp(IOB,0x30|table1[tmp]);**

**button=inp(IOA);**

**if(watchdaoche(button)){break;}**

**delay();**

**}**

**if(watchdaoche(button)){break;}**

**}**

**}**

1. **观测并记录仿真实验结果（要求客观、真实）**

仿真结果基本满足预期。

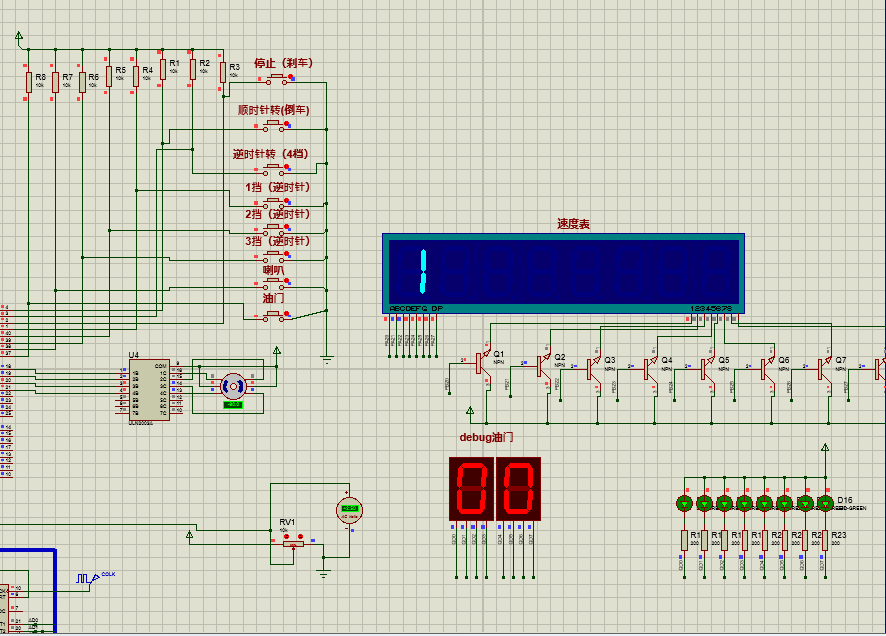


图 22 1档结果

当按下“1档”的时候，步进电机转动，数码管显示1.其他几个档位类似。

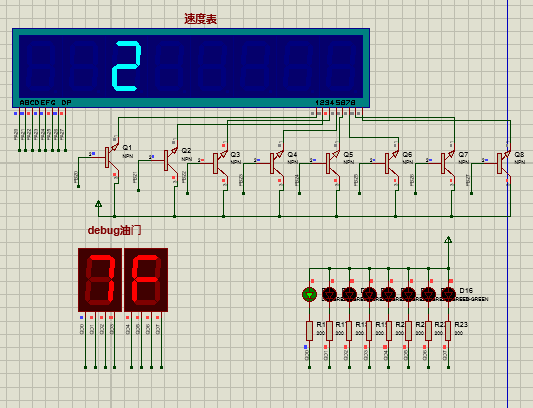


图 23 油门结果

当按下油门的时候，数码管闪烁加速度，红色数码管显示BCD码，led灯显示二进制数。

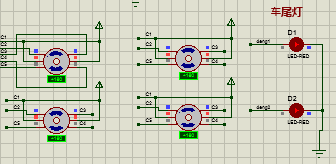


图 24 车尾灯

当在倒车的时候，车尾灯闪烁，表示正在倒车。

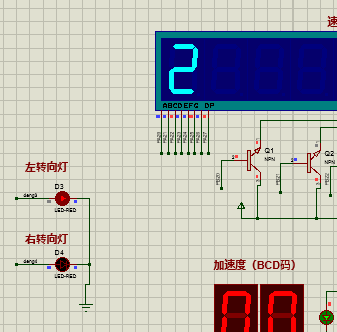
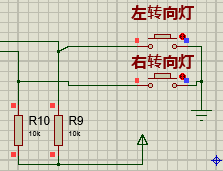
 

图 25 转向灯

当车在行驶的时候（例如在二档），按下左转向灯的开关，左转向灯会闪烁，右转向灯的效果也是一样的，在这里就不再赘述。





图 26 喇叭

当按下喇叭键，只是等会闪烁“LABA”，蜂鸣器会发出声音。

**声明：如果有不明白的地方可以参看报告所在目录下的三个视频**

**（四）系统使用说明书**

**① 声明**

使用本系统之前请先阅读使用说明书。

应课程设计要求进行设计，如需转载，请注明出处。

本套系统是以文件的形式进行保存，一般不会有安全方面的隐患。

**② 运行环境**

本系统在proteus8.3上运行仿真通过，使用digital mars c编译器，主要用c语言编写。对端口的读写（inp和outp函数）部分采用汇编。

**③ 用户可以操作的模块介绍**

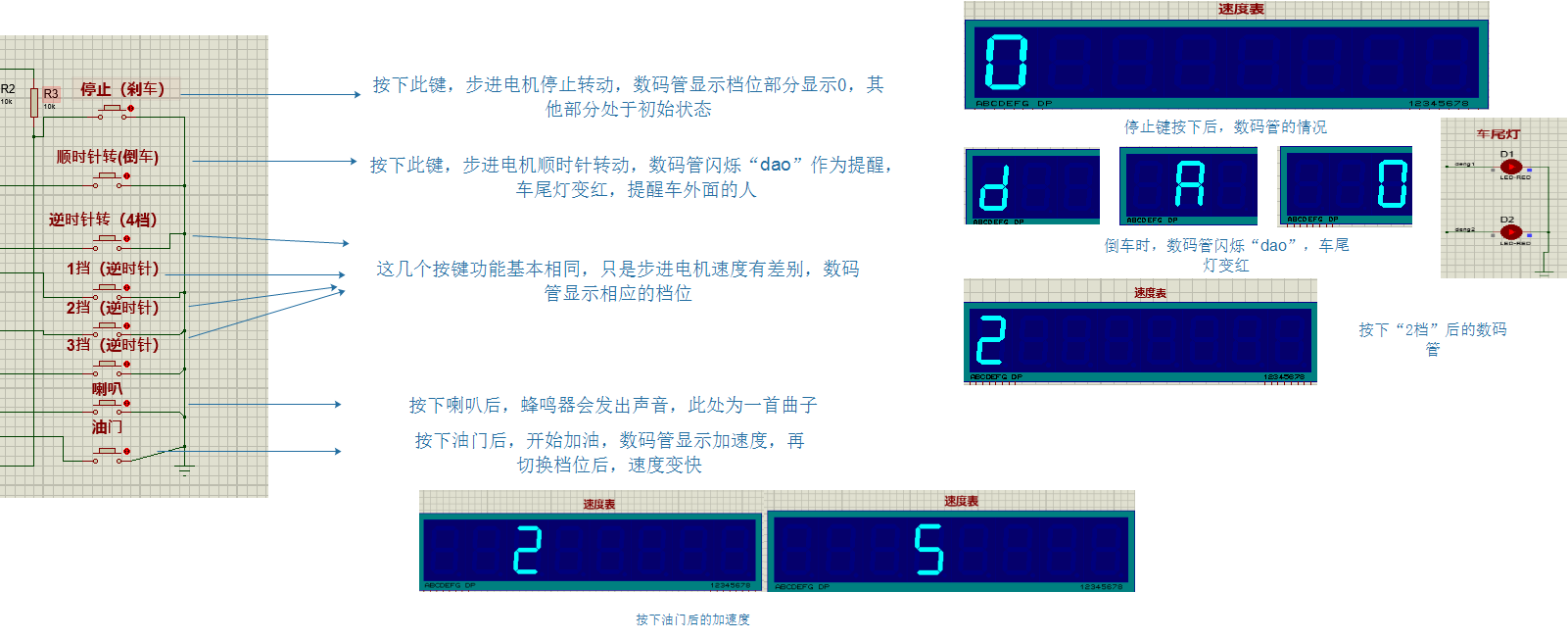
****

图 27 用户按键模块1

上图所示的用户可操作的模块在图纸的右上角。

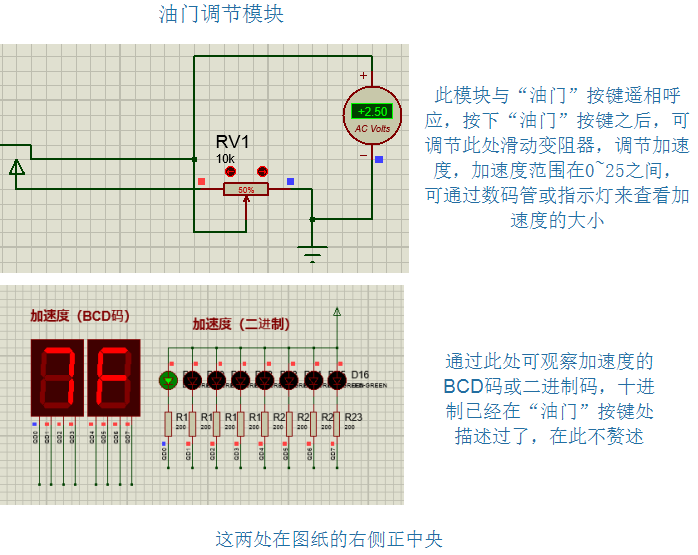


图 28 用户可操作模块2

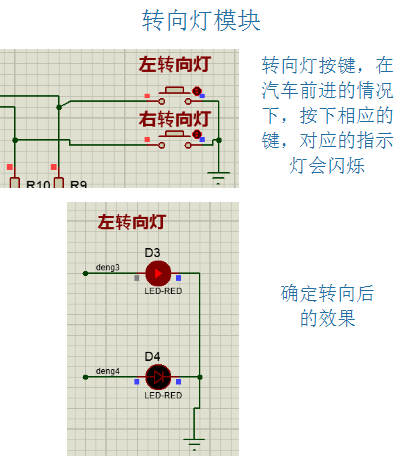


图 29 用户可操作模块3

**④显示模块**

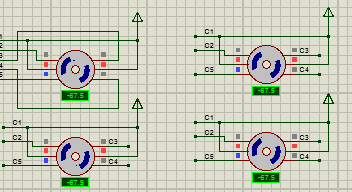


图 30 车轮

此处四个步进电机模拟汽车的四个车轮，更加形象，其转速模拟车的行驶速度

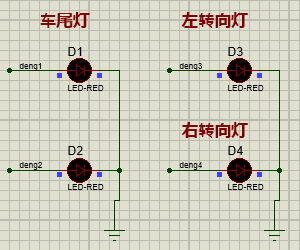


图 31 灯光系统

此处四个LED灯模拟车的灯光系统，包括尾灯和转向灯



图 32 速度指示系统1

由数码管和指示灯组成车的速度指示系统，显示车的档位，油门大小，倒车时的提醒。

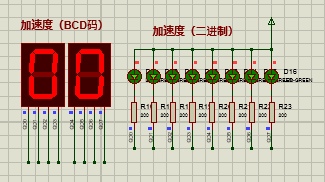
****

图 33 速度指示系统2

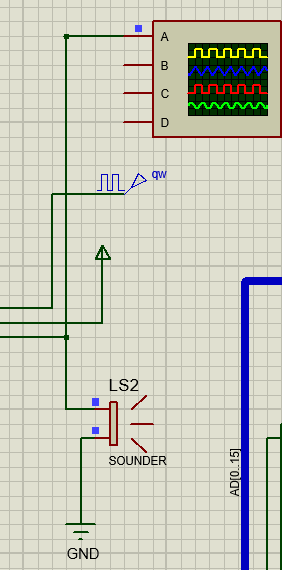
****

图 34 声音系统

蜂鸣器可以发出声音，示波器可以观察波形。

**⑤ 用户可调节模块**

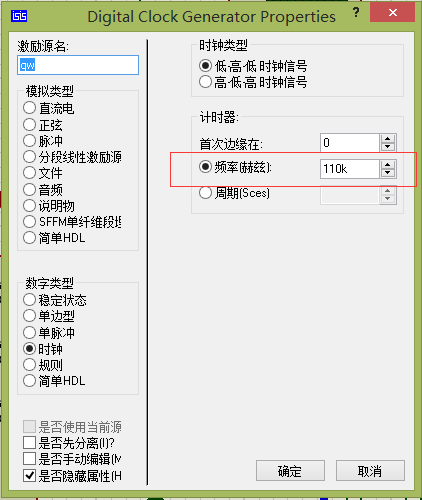
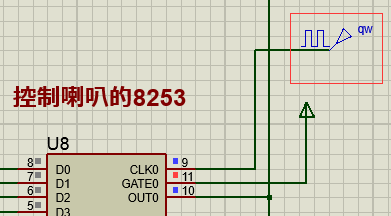


图 35 可进行设置的模块

可以编辑波形的属性，设置波形的频率，调节声音的质量

**声明：如有不明白地方，可以参照报告所在目录下的三个avi视频。**

1. **分析与讨论**
2. **步进电机运行速度的控制**

控制步进电机的速度有两种途径：一是改变输入脉冲的频率，通过对定时器（如8253）定时常数的设定，使其升频、降频或恒频；二是软件延时，或调用延时子程序。

采用软件延时的方法来改变步进电机的速度，虽然简便易行，但延时受cpu主频的影响，导致在主频较低的微机上开发的步进电机控制程序换到主频较高的微机上，就不能正常运行，甚至由于频率太高，步进电机干脆不动了。应该指出的是，步进电机的速度还收到本身矩-频特性的限制，设计时应满足运行频率与负载力矩之间的确定关系，否则，就会产生失步或无法工作的现象。

在此次设计中，我仍然采用的是软件延时法。如上文所述，软件延时有着种种的弊端，为什么还要用软件延时呢？

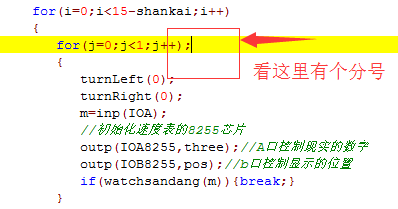
因为此次设计是作为课程设计，主要目的是引导我们掌握接口编程的基本方法，是学习为主，并不是为了商业用途，采用的也是已经停产n多年的8086cpu，程序也不会移植到另外一台机器上，所以为简单起见，降低绘制的图纸的复杂层度，采用软件延时。

**2. 喇叭的声音比较刺耳，非常难听**

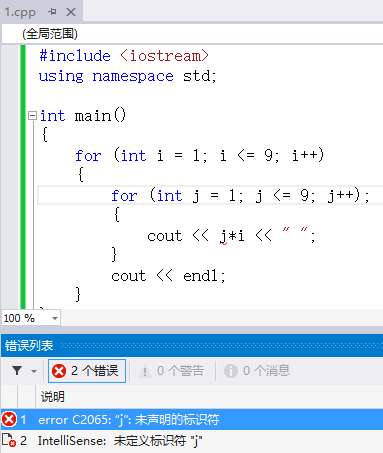
用8253的工作方式3产生的方波带有高频的部分，声音比较难听，可以用D/A转换产生一个相对比较平滑的脉冲来驱动蜂鸣器发声。

1. **编译器会出现问题**

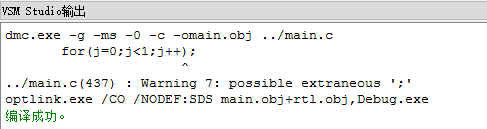
我们用的是digital Mars C 这一款编译器，和我们以前用的vc的编译器是有一些不同的。举个例子，看下图：



这种情况在vs编译器等高级语言编译器中是坚决不允许的，vs编译结果如下图：



但是在digital C中不会报错，只是报了一个警告，如下图：



但是程序的运行结果，是与预期不相符的，但是我现在也不明白它会怎么去执行这段代码，等有机会去研究研究。

1. **八段数码管问题**

预期效果：数码管的显示的数字或字母应该是连续（利用人的视觉暂留）的，

比如“dao”等。

出现的问题：数字无法同时显示，只能一个字符一个字符的显示

处理方法：如果是利用八段数码管的话，目前是无法解决的。可以采用另一种个数是4个的数码管

1. **个人体会与小结**

这次微机原理课程设计历时一个星期，虽然老师只安排了两天的时间，在整整两星期的日子里，可以说得是苦多于甜，但是可以学到很多很多的的东西，同时不仅可以巩固了以前所学过的知识，而且学到了很多在书本上所没有学到过的知识。期间还深刻体会到了画程序方框图是很有必要的。因为通过程序方框图，在做设计的过程中，我们每一步要做什么，每一步要完成什么任务都有一个很清楚的思路，而且在程序测试的过程中也有利于查错。

其次，以前对于编程工具的使用还处于一知半解的状态上，但是经过一段上机的实践，对于怎么去排错、查错，怎么去看每一步的运行结果，怎么去了解每个寄存器的内容以确保程序的正确性上都有了很大程度的提高。

通过这次课程设计使我懂得了理论与实际相结合是很重要的，只有理论知识是远远不够的，只有把所学的理论知识与实践相结合起来，从理论中得出结论，才能真正为社会服务，从而提高自己的实际动手能力和独立思考的能力。在设计的过程中遇到问题，可以说得是困难重重，这毕竟第一次做的，难免会遇到过各种各样的问题，同时在设计的过程中发现了自己的不足之处，对以前所学过的知识理解得不够深刻，掌握得不够牢固。

这次课程设计终于顺利完成了，在设计中遇到了很多编程问题，最后在指导老师的辛勤指导下，终于游逆而解。在指导老师的身上我学得到很多实用的知识，在此对给过我帮助的所有同学和墙老师表示忠心的感谢！

1. **参考材料**

[1] 微型计算机接口技术及应用

[2] 课程PPT

等