**郑州轻工业学院**

本科毕业设计（论文）

题 目

学生姓名 徐子昂

专业班级 电子信息工程14-01

学 号 541401030147

学 院 电子信息工程学院

指导教师(职称) 徐盛

完成时间 20 年 月 日

郑州轻工业学院

**毕业设计（论文）任务书**

（空1行。小四号1.5倍行距）

**题目 基于单片机的防酒后驾驶控制系统设计**

**专业 电子信息工程 学号 541401030147 姓名 徐子昂**

**主要内容、基本要求、主要参考资料等：**

**完 成 期 限：**

**指导教师签名：**

**专业负责人签名：**

**年 月**

基于单片机的防酒后驾驶控制系统设计

摘 要

这些年以来，随着我们国家汽车的普及，解决汽车造成的各种安全问题已经迫在眉睫。根据目前最新的调查数据，现在大部分的汽车安全事故都有酒后驾驶的影子。我们国家因为酒后驾车而造成的交通事故数量每年都在增加。为了防止日益严重的酒后驾驶问题，我们必须设计开发出一种可以方便的测量汽车驾驶员身体中乙醇浓度的设备，并且能在需要时发出报警同时关闭发动机。

该设计使用了AT89C51单片机来对各种数据元器件进行操处理。同时还选用了MQ-3 型酒精传感器来检测人呼出气体中的乙醇浓度，并将其转换为电信号。再之后通过使用 A/D转换器来将模拟信号信号变为数字信号，然后再输入到微控制器中。之后由单片机通过对数字信号的处理来显示酒精浓度在数码管上。在超出了设定值时，蜂鸣器发生报警，同时控制继电器来开关电源。

关键词 AT89C51单片机；MQ-3 酒精传感器；智能电路；酒后驾车；

**Design of anti drunk driving control system based on single chip microcomputer**

**ABSTRACT**

Over the years, with the popularity of cars in our country, it is urgent to solve all kinds of safety problems caused by automobiles. According to the latest survey data, most of the automobile accidents now have the shadow of drunk driving. The number of traffic accidents caused by drunk driving in our country is increasing every year. In order to prevent the growing problem of drunk driving, we have to design and develop a device that can easily measure the concentration of ethanol in the driver's body, and can alarm and close the engine at the time of need.

The design uses AT89C51 microcontroller for data processing. At the same time, MQ-3 alcohol sensor was used to detect the concentration of ethanol in human breath and convert it into electrical signal. After that, the analog signal is converted into digital signal by using A/D converter, and then input to the microcontroller. After that, the microcontroller shows the alcohol concentration on the digital tube by processing the digital signal. When the setting value is exceeded, the buzzer will alarm and control the relay to switch the power supply.

**KEY WORDS** STC89C51 microcontroller;intelligent circuit;drunk driving

目 录

[本科毕业设计（论文） 1](#_Toc515807090)

[1 绪论 1](#_Toc515807091)

[1.1论文研究的背景 1](#_Toc515807092)

[1.2单片机的现状 2](#_Toc515807093)

[2 设计方案 2](#_Toc515807094)

[2.1功能设计 2](#_Toc515807095)

[2.2酒精浓度检测仪要求分析 2](#_Toc515807096)

[2.3系统的整体设计方案 3](#_Toc515807097)

[3 主要硬件的选取 3](#_Toc515807098)

[3.1单片机 3](#_Toc515807099)

[3.1.1 AT89C51简介 4](#_Toc515807100)

[3.2传感器的选择 5](#_Toc515807101)

[3.3 A/D转换器 7](#_Toc515807102)

[3.3.1 A/D转换器简介 7](#_Toc515807103)

[3.3.2 ADC0809转换器简介 8](#_Toc515807104)

[4 系统硬件设计 9](#_Toc515807105)

[4.1单片机最小系统设计 9](#_Toc515807106)

[4.1.1晶振电路 10](#_Toc515807107)

[4.1.2复位回路 10](#_Toc515807108)

[4.2 信号采集电路 11](#_Toc515807109)

[4.3 A/D转换电路 12](#_Toc515807110)

[4.4数码管显示电路 13](#_Toc515807111)

[4.5 声光报警电路 14](#_Toc515807112)

[4.5.1 灯光提示电路 14](#_Toc515807113)

[4.5.2声音报警电路 15](#_Toc515807114)

[4.6继电器工作电路 15](#_Toc515807115)

[4.7键盘电路 16](#_Toc515807116)

[4.7.1键盘的类型 16](#_Toc515807117)

[4.7.2键盘去抖动 16](#_Toc515807118)

[5软件设计 18](#_Toc515807119)

[5.1使用软件介绍 18](#_Toc515807120)

[5.1.1 Keil C51 18](#_Toc515807121)

[5.1.2 Protel99SE 19](#_Toc515807122)

[5.2主程序框图 20](#_Toc515807123)

[5.3数据采集部分程序框图 21](#_Toc515807124)

[6电路仿真 22](#_Toc515807125)

[7 实物的制作与调试 24](#_Toc515807126)

[结束语 25](#_Toc515807127)

[致谢 26](#_Toc515807128)

[参考文献 26](#_Toc515807129)

[附录 26](#_Toc515807130)

# 1 绪论

## 1.1论文研究的背景

如今，人民的生活水逐渐提高，购买汽车的人也变得多了起来。使用汽车出行给我们生活带来了舒适与便捷，节省了出行的时间，但随着汽车的普及，一系列的交通事故也越来越多，这对人们的生命安全造成了严重的威胁。根据调查，酒后驾驶而造成的交通事故占很大比重，死亡率更是远超其他种类事故。我国目前许多地方“无酒不欢”这个概念深入人心，大家开车聚会喝酒之后又开车离开，这是对自己与他人生命的不负责。

一旦饮酒达一定量后，会降低人的反应速度，让人对外部的刺激变得不敏感，导致紧急情况发生时的反应速度下降。酒精对驾驶员的视线也有影响，会使人变得视力模糊。经过医学研究证明，大部分的人，在饮酒之后都会在体内残留留有大量的酒精,这会导致饮酒者感觉头晕，疲乏。而大部分酒后驾车的人都不知道酒后驾车的危险，他们往往会认为自己的酒量非常好，喝完酒后和喝酒以前没有区别，所以能和正常人一样的驾驶。他们非常自信自己没有危险，于是在酒后他们依然会选择自行开车开，行驶在路上，对危险向自己靠近浑然不知。等到造成事故时，都会后悔不已，甚至失去了自己的生命。科学研究表明，未饮酒的正常驾驶员在驾驶车辆遇到前方危险时，从发现危险开始一直到踩下刹车，这之间大约会有0.75秒的时间间隔。而饮酒后的驾驶员在相同情况下遇到危险的时候，反应间隔往往会比没有饮酒的正常驾驶员长2-3倍。因此在相同的情况下驾驶，酒后驾驶的人会比未饮酒正常的驾驶员制动的过程会延长很多，这样极大的增加了交通事故发生的概率。因此得出，驾驶员如果进行酒后驾驶，会对道路交通安全造成严重的危害。

为了防止酒后驾车的情况屡屡发生，设计一款能根据驾驶员体内酒精含量从而控制汽车启动的装置已经迫在眉睫。它能防止酒后的驾驶员发动汽车，降低酒驾发生的概率。虽然目前我国已经将酒后驾驶列入了刑法，但还是有许多人存在侥幸的心理，不顾他人生命安危继续酒驾。这款装置可以从源头解决酒后驾驶问题，使喝酒后的人无法启动车辆，让因酒驾热造成的交通事故概率有效的减少。

## 1.2单片机的现状

单片机也可以叫做微控制器，相当于将微型计算机系统集成到一个芯片上。相比较于大量运用在PC上的通用微型处理器，它更加的强调不需要外接硬件与节约成本。单片机优点在于其体型小，造价不高，输入输出接口较为简单。它诞生于1971年，Intel研制出了世界上第一个单片机，之后迅速发展，往生活中各个领域渗透。当前，世界上主流的芯片设计公司都有推出自己的设计制作的单片机芯片。8位，16位和32位，应有尽有。由于简单可靠受到了广大好评。

现如今单片机在我们生活中各个领域都得到了广泛应用，几乎很难找到一个没有应用单片机的领域。智能化家用电器、工业自动化控制设备、计算机网络通讯、商业营销设备等，这些都离不开单片机。

# 2 设计方案

## 2.1功能设计

本文所设计系统的目的是为了防止司机进行酒后驾车的行为。在司机上了车之后，会接通汽车电源，此时本文所设计的系统就会自行通电打开。如果司机想点燃发动机进行驾驶，就必须先进行酒精检测，否在会无法点燃发动机。在司机对数据采集的部分呼出气体后，本系统会进行数据采集并判断驾驶员是否数据酒驾。如果属于酒驾，则会断开汽车的打火装置让驾驶员无法启动汽车，同时会发出声光报警。若驾驶员经过测试之后并不属于酒后驾车，汽车就会正常的打火启动行驶。

## 2.2酒精浓度检测仪要求分析

设计应有以下三个特点

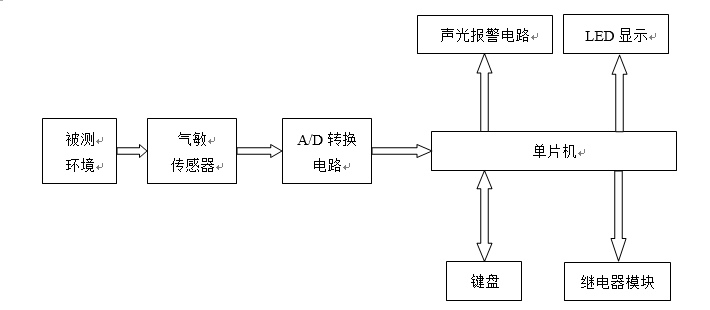
（1）利用单片机作为核心部分，采集空气中的酒精浓度数据。系统还需要包含LED显示电路数据收集电路等。用户不需要使用其他的额外设备，就可以对系统进行一些控制。

（2）考虑到设备实际所应用的场景，设计因需要考虑到便携性。用户可以通过LED显示管上出现的数字，来查看酒精浓度。还可以使用按键控制报警浓度的高低。

（3）整个系统应具有功耗低、小型轻便等特点。

## 2.3系统的整体设计方案

本系统主要分为气体数据收集模块，单片机模块，LED显示模块与继电器模块。运作模式为，首先使用酒精传感器来收集空气中的乙醇气体的浓度信息。再把通过传感器所收集的乙醇浓度数据转换为可以处理的电信号，之后将传感器输出的模拟信号通过A/D转换器进行处理成为数字信号。再将所得到的数据传入单片机进行处理，通过数码管显示浓度，并且告诉报警模块浓度是否超标，同时控制继电器的开关以保证驾驶员无法酒后驾驶。



**图2-1 系统工作框图**

# 3 主要硬件的选取

## 3.1单片机

MSC-51系列单片机是由著名的英特尔公司设计生产的。同时MSC-51也是现在应用的最多的单片机系列。8051/80C51是MSC-51这个系列单片机的核心。虽然该系列种类繁多，但这个系列另外的一些产品基本都是基于8051/80C51上所进行改装的。

其中ATMEL公司考虑到将用户需要的Flash储存器与80C51这一内核结合起来，从而设计出全新的单片机系列，这就是AT89系列。AT89系列和MCS-51系列虽然是两种系列，是由两个不同的公司生产的。但是它们两者不管是在引脚还是系统指令等方面都是可以兼容的。除了相同的部分，AT89还在MCS-51的基础上还添加了一部分新的功能，例如看门狗定时器WDT与SPI串行接口等。

对比MCS-51系列，AT89系列单片机有许多优点，是对其的改进。例如AT89系列的Flash能够通过电来进行瞬间的擦除与改写。不仅如此还可以对加密写入单片机内部的程序，进行保密处理。

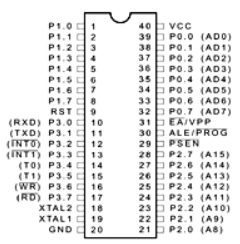
Atmel公司的AT89系列单片机型号主要有AT89C51、AT89S52与AT89C52等。本系统所选用的型号是AT89C51。

### 3.1.1 AT89C51简介

AT89C51单片机主要封装形式分为两种，一种是方形封装，另一种是双列直插式封装。

AT89C51有40个引脚如图3-2所示，可以分为：

1. 电源引脚2根
2. 时钟引脚2根
3. 控制引脚4根
4. I/O引脚32根

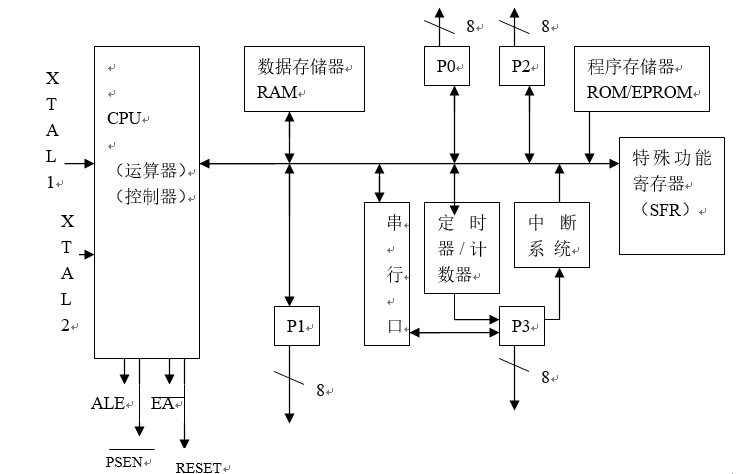


**图3-1 AT89C51引脚**

AT89C51性能很高，由于它的引脚数的目有限，故采用引脚复用技术，来让有限的引脚做出更多的功能，因此其部分引脚有两种功能。

I/O引脚一共三十二跟。[P1.0~P1.7](file:///C:\Users\XUZIAN~1\AppData\Local\Temp\Rar$EXa15328.10389\%E3%80%8A111111111111111%E3%80%8BPaperFree%E6%96%87%E6%A1%A3%E6%A3%80%E6%B5%8B%E6%8A%A5%E5%91%8A\htmls\sentence_detail\56.htm) [P2.0~P2.7](file:///C:\Users\XUZIAN~1\AppData\Local\Temp\Rar$EXa15328.10389\%E3%80%8A111111111111111%E3%80%8BPaperFree%E6%96%87%E6%A1%A3%E6%A3%80%E6%B5%8B%E6%8A%A5%E5%91%8A\htmls\sentence_detail\58.htm) [P3.0~P3.7](file:///C:\Users\XUZIAN~1\AppData\Local\Temp\Rar$EXa15328.10389\%E3%80%8A111111111111111%E3%80%8BPaperFree%E6%96%87%E6%A1%A3%E6%A3%80%E6%B5%8B%E6%8A%A5%E5%91%8A\htmls\sentence_detail\60.htm)均为双向I/O口，其中P3口除了作为I/O口之外，还有另外的功能。

|  |  |
| --- | --- |
| P3.0 | RXD（串行输入口） |
| P3.1 | TXD（串行输出口） |
| P3.2 | /INT0（外部中断0） |
| P3.3 | /INT1（外部中断1） |
| P3.4 | T0（计时器0外部输入） |
| P3.5 | T1（计时器1外部输入） |
| P3.6 | /WR（[外部数据](https://baike.baidu.com/item/%E5%A4%96%E9%83%A8%E6%95%B0%E6%8D%AE)[存储器](https://baike.baidu.com/item/%E5%AD%98%E5%82%A8%E5%99%A8)写选通） |
| P3.7 | /RD（外部数据存储器读选通） |



**图3-2 单片机的结构**

## 3.2传感器的选择

现代科技不断发展，酒精传感器也诞生了许多种类。目前主要有半导体型、红外线型、燃料电池型等。考虑到本系统目的在于测量人所呼出的气体中的乙醇浓度，之后根据呼出的气体中的乙醇浓度转换为血液中的乙醇浓度。因此采用气敏传感器。人呼出气体中不止有酒精还有二氧化碳等其他气体，所以应该选用一个只对酒精敏感的传感器，于是本文决定选择MQ-3型传感器。MQ-3型传感器对空气中的乙醇具有很高的灵敏度，其使用寿命长，制造成本低，可靠性相对较高。MQ3型酒精传感器将内部元件固定再不锈钢外壳的内部，内部一般有陶瓷管与二氧化硅敏感层。由于MQ3需要在特定的温度中工作以减小误差，所以有加热元件为其提供工作化境。当传感器工作环境中酒精气体浓度增长时，传感器的导电率也会随着增加。MQ3在实际中使用并不复杂，使用比较简单的回路就能够把传感器电阻的变化转为乙醇浓度相对应的电压进行输出。

MQ-3传感器的回路由两个部分组成。其中一条是加热回路，它用来使传感器达到相应的工作温度；另外一条是信号输出回路，功能是把MQ-3的电阻值变化转换成电信号进行输出。

MQ-3所处环境中乙醇浓度变化时，其表面的电阻RS也会改变。RS改变后，与其串联进行连接的负载电阻RL上的电压也会改变，将有效电压进行输出即可获得数据。传感器进行加热所需的电压为5v。MQ-3型酒精传感器的自身结构与回路如图3-3和图3-4所示。MQ-3的电阻值改变速率和其自身的温度高低还有乙醇在气体中的浓度大小的关系图如图3-6所示。如果直接使用误差会很大，应先加热3分钟，使传感器达到工作温度在进行测量，这时测量误差会减小很多。



**图3-3 MQ3 结构和外形**



**图3-4 MQ-3阻值变化率与乙醇气体浓度、外界温度之间的关系**

## 3.3 A/D转换器

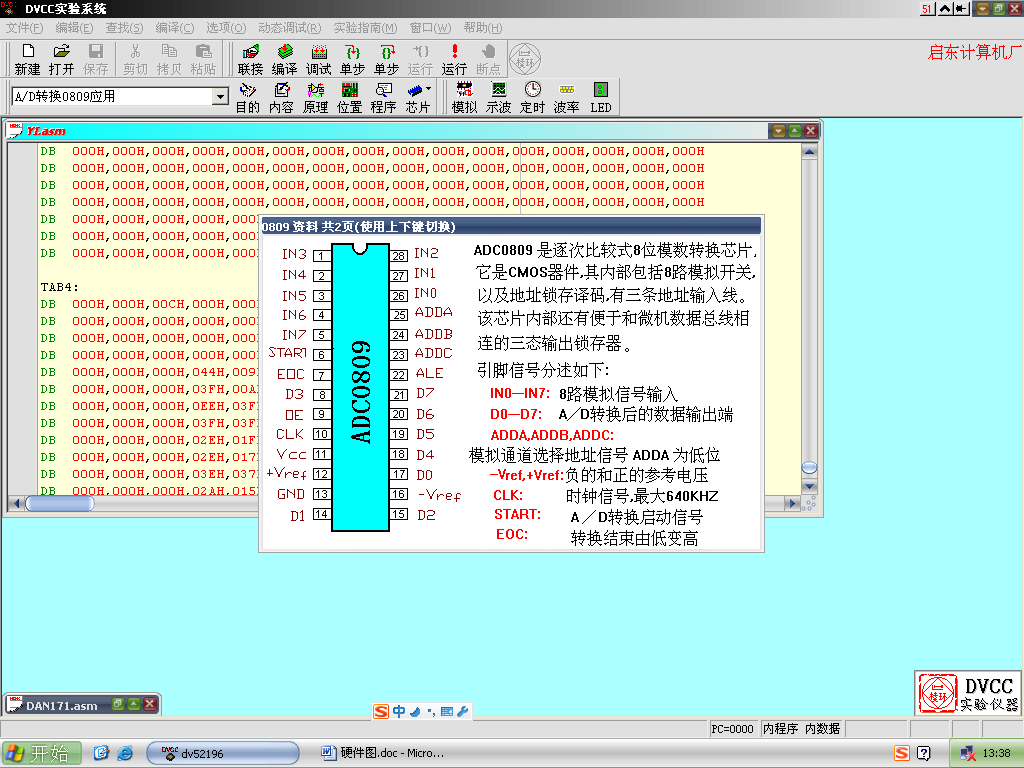
### 3.3.1 A/D转换器简介

A/D转换器是模数转换器的简称，也可以简称成为ADC。它的功能就如名字所示，是为了将模拟信号转换为数字信号而设计出来的元器件。在各种使用单片机的测量中，有许多非电的物理量，如温度、浓度和压力等。这些量需要被传感器转换成模拟信号这种电信号，不转换的话单片机是无法处理的。但是模拟信号也没有办法处理，AT89C51需要输入的是数字信号。因此须要进行转换，将模拟信号转换为数字信号。将模拟信号转换为数字信号这一过程是不可缺少的，而实现将模拟信号转变成数字信号的工具就是A/D转换器。

A/D转换器的主要类型有这么几种：一种是积分型A/D转换器，它的原理是对输入其内部的电压进行处理，转换成时间或者频率。这之后再使用定时器、计数器进行处理，测出时间间隔得到所需值。它的优点是精度较高，工作电路简单，成本不高，但其转换速率较低；一种是是并行比较型转换器。并行比较型转换器本身使用了多个比较器来进行比较，可以只需要进行一次比较就可以实现从模拟信号到数字信号的转换，由于其速度快，也成为Flash型。虽然它的转换速率非常高，但在速率高的同时，也需要多个比较器来进行比较，所以它的电路规模比较大，成本造价高，所以只在特别需要转换速度的情景；一种是逐次逼近型A/D转换器，它的原理是从大到小，按顺序对输入的电压数值进行处理和它内置的D/A转换器的输出进行对比，通过n次的对比，从而输出对应的值。其优点是精度、速度、价格都适中。

### 3.3.2 ADC0809转换器简介

综合考虑本次设计中所选用的为逐次逼近型A/D转换器，所用型号是ADC0809，它的引脚如图3-5所示。它是一个8位A/D转换器。IN0~IN7(25~28,1~5)：这八个引脚为ADC0809要进行处理的模拟信号的输入所使用的端口。START口是控制模拟信号到数字信号是否进行转换的端口，当输入高电平时就会启动转换器。EOC是发出ADC0809结束工作的端口。转换器启动的时候，该引脚输出的是低电平。但在A/D转换的过程停止之后，它又会转而输出高电平。D0~D7是经过ADC0809处理后的输出端口。OE：控制经过处理后的数据是否能输出的端口。输入高电平式有效。当停止工作的时候，若这个端口的输入信号是高电平，就会把锁存器中的储存的数据通过D0到D7端口输出。CLK：输入时钟脉冲的引脚。其频率需要控制在640HKZ以下。Vcc：为ADC0809供电的端口，需要使用的电压为5V。GND：接地端。ADDA、ADDB、ADDC，是3位地址输入端口，通过三个输入电平高低，从而用来控制从哪一路输入输出信号。ALE：控制地址是否锁存的端口，在输入高电平的时候锁住数据。



**图3-5 ADC0809的引脚**

ADC0809内部的构造图如图3-6所示。在使用的时候先Vcc通电，然后需要给ADDA、ADDB、ADDC输入3位地址选择使用端口。同时给与ALE高电平，这时地址将会存入到[锁存器](http://baike.baidu.com/view/1498669.htm)中。这个地址经过[译码](http://baike.baidu.com/view/189742.htm)器译码之后，通过模拟通道之前决定的那路输入到[比较器](http://baike.baidu.com/view/421754.htm)中。之后还要对寄存器进行复位，通过给与START高脉冲使寄存器进行复位。脉冲的下降沿将会启动ADC0809开始进行工作，并同时会使EOC的输出成为低电平。当A/D工作完成之后，EOC信号将会回到高电平，锁存器将会接受这些经过处理后的数据，告诉CPU转换已经结束。需要读取数据的时候，会通过将高电平输入OE端，从而让三态门变为可以读取的状态，这时即可从输出端读取数据。

START CLK

OE

VR(+) VR(－)

VCC

GND

EOC

D0

**.**

**.**

**.**

D7

三态输出锁存器

8位A/D转换器

地址锁存与密码

C

B

A

ALE

8 路模拟量开 关

IN7

**….**

IN0

**图3-6 ADC0809的结构框图**

# 4 系统硬件设计

## 4.1单片机最小系统设计

单片机的最小系统是指电源、晶振和复位电路等在内的系统，这些部分可以保持微控制器正常工作。这些部分是让单片机可以正常工作所必不可少的，因此这些部分被称为单片机的最小系统。在这之上，我们还可以通过添加一些内存等部件，使单片机能够完成更多任务。

### 4.1.1晶振电路

单片机中，必须要有时钟信号来进行控制，才能让工作按规定的时间来进行执行。本文所选用的AT89C51一般拥有以下几种时钟信号的生成方式。

一种是外部时钟方式。这种方式顾名思义，就是从外部引入时钟信号，做法是在单片机的XTAL1引脚输入外部生成的时钟信号。同时 XTAL2 不接电，这时XTAL1所接的频率通常情况下是小于 12MHZ 的方波。这种工作方式只有在需要数个单片机一起进行处理时才会使用。因为必须要让各个单片机互相进行连接，使它们的工作时间统一。

本文设计所采用的是另外一种内部时钟方式，结构如图3-11所示。AT89C51微控制器本身会自带一个振荡回路，直接使用即可。在使用的时候需要在微控制器的XTAL1与XTAL2端外接一个晶振，这样就能产生一个震荡电路。同时会在单片机的内部产生一个时钟脉冲信号。一般会如图所示加上两个电容，使用电容的原因是这样能让起振的速度变快。晶体振荡器所使用的振荡频率范围通常是1.2到12MHz之间，我们实际使用的时候一般会选择6MHz或者12MHz。



**图4-1 89C51内部时钟电路**

### 4.1.2复位回路

在微控制器执行程序的时候中，会有一个指针。这个指针会自动指向下一个马上会被读取的语句。在微控制器重新上电的时候，必须经过复位。这是为了让指针指向AT89C51即将读取的最开始的那条程序。在之后的过程中，微控制器每次运行一条语句，都会让指针指向下一条要运行的语句。这样就能使微控制器知道下一次需要执行什么，不会出现错误的执行顺序。微控制器上电时进行的复位操作使单片机每次重新执行程序的顺序都一样，不会出现两次执行出不同结果的状况。

当接通电源的时候，向RST端口输入高电平，如果给予RST高电平的时间维持了两个机器周期，微控制器就会开始进行复位，让指针回到程序开始。但是如果一直保持在高电平状态，微控制器就会不停的进行复位。

在设计复位电路的时候，我们需要决定使用哪种元件的参数，如果参数选择不当，可能会造成各种各样的问题。例如在选择使用电阻的过程中，我们一定要选择合适的电阻，不能选择阻值太小的电阻也不能选用阻值太大的电阻。因为如果我们选用的电阻阻值过低，会让电路电流过大，没办法对电路进行限流。如果我们使用的电阻阻值过大，可能会让RST引脚输入不了高电平，让电路无法复位。电容值同理，我们需要选择能够正常运行的电容值，如果我们的选择不当，很可能让电路无法正常复位。

除了微控制器自己默认的复位之外，在一些情况下我们可能会遇到问题，需要通过人工操作来进行复位。复位所用电路如图4-2所示，通过使用按钮来进行人工操作复位。

****

**图4-2 89C51复位电路**

## 4.2 信号采集电路

信号采集电路图如图4-3所示。

MQ-3连接+5V 直流稳压电源。它需要通过电路，来将空气中乙醇浓度转换为其阻值变化，之后再由本电路将阻值变化转化为输出电压的变化。在进入单片机之前还会再通过 ADC0809转换为相应的数字信号。



**图4-3 信号采集电路图**

## 4.3 A/D转换电路

本文所采用的ADC0809是一个八位八通道的逐次逼近型A/D转换器。它的结构框图如图4-3所示，由+5V电源供电。通过向ADDA\ADDB\ADDC这三个引脚输入的电平高低来决定使用哪个通道。ADC0809完整的完成一次从模拟信号到数字信号的转换需要100μs左右。所输出的信号不需要再经过处理，就能够直接的连接到微控制器上上。

经过一些电路连接，ADC0809可以将输入的模拟信号进行转换。它的连线图如图4-4所示：



**图4-4 ADC0809的连线图**

## 4.4数码管显示电路

本文设计所使用的八段数码管，本质就是八根LED管所摆放而成的。它相比于七段数码管来说多出了一个小数点。八段数码管点亮特定的几根LED管的时候就会构成想要的数字。

数码管按连接的方式可以分为两种。一种就是共阳极数码管，所谓的共阳数码管就是把它们的LED管的阳极连接到一起。哪一段阴极为低电平就点亮哪一段，高电平的时候就不点亮。另一种就是共阴极数码管，它正好与共阳极数码管相反，他需要将所有的LED管的阴极连接到一起。

数码管通过按显示方式区分也可以分成两类。其中一种叫做静态驱动，就是让数码管所有的段都分配一个微控制器的端口来单独操作。虽然其程序简单，数码管显示的亮度较高，但其缺点也非常明显，使用的时候需要占用大量的I/O口。若要驱动n根数码管就会使用n\*8个微控制器端口。但单片机I/O不仅需要控制数码管，还需要用作其它用途。在实际的使用过程中，我们通常会需要多根数码管一起工作，因此不适合使用静态显示。大多数时候我们会使用动态显示来解决问题。

在使用静态显示时，我们面临所需要的I/O口过多的问题，而另一种方法，通过使用动态显示则可以很好的处理所需端口过多的问题。动态显示与静态显示相反，它要求将所有的段进行并联，然后在通过使用位选线来控制点亮。单片机会轮流对数码管发送所要现实的字形码。通过使用程序来进行延时，按顺序决定各个LED段是否被点亮，这样可以让每个数码管输出不同的内容。通过给单片机编写程序，来输出信号，首先输出的是位选信号，来告知要控制的是哪一根LED管。决定好要控制的段之后，再输出段码，通过段码来告知该如何显示。不同输出之间有一个小段的延时，快速轮流对每个数码管进行控制。因此各个数码管不是同时显示的，但是由于生物上人有视觉暂留的现象，虽然不是同时显示的数据，只要通过合适的频率转换显示，就能让我们的感觉上是同时正常显示的。但是延时时间过长，则会让人感觉数码管在闪烁。延时时间过短，则会让数码管的亮度不足，而且由于视觉暂留，会让人感觉有重影。如果使用动态显示，则不需要像静态显示那样占用微控制器大量的引脚。

本文所设计的数码管显示电路如图4-5所示，利用电阻调整电流，从而调整数码管显示的亮度。



**图4-5 数码管显示电路**

## 4.5 声光报警电路

### 4.5.1 灯光提示电路

灯光报警电路如图4-6所示。当酒精浓度超标的时候，微控制器输会出高电平，这样使发光二极管被点亮。



**图4-6灯光报警提示电路**

### 4.5.2声音报警电路

声音报警电路如图4-7所示。本电路使用蜂鸣器来发出声响进行报警。它使用直流供电，是经常使用的元器件。使用它的时候通常会在电路里加上三极管、电容与二极管。蜂鸣器用来发出警报，在他的两端施加直流电压就可使它工作。图中电阻起限流作用，并且使用三极管来做为开关使用。如果基极上是高电平，就会让它导通，这时候就会开启蜂鸣器，进行发声报警。如果基极上是低电平，就会让三极管关闭，蜂鸣器就会停止工作，不再发出声响。



**图4-7声音报警电路**

## 4.6继电器工作电路

继电器也是一种常用的电子元器件。为了保护一些元器件或其他作用，使用继电器来作为开关来控制电路的闭合。继电器本身也有非常多的种类，本文所选用的是电磁继电器。继电器工作电路如图4-8所示。



**图4-8 继电器控制负载电路**

继电器可以根据单片机输出的信号进行开关操作。当乙醇气体浓度超过了设定值的时候，微控制器输出的高电平就会转换成低电平。输出电平转换会让三极管导通，让继电器吸合。

## 4.7键盘电路

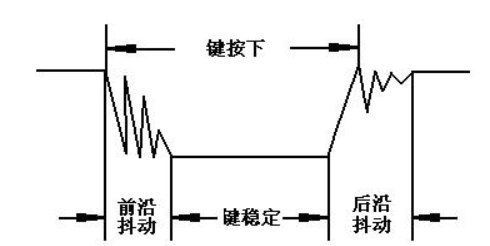
### 4.7.1键盘的类型

微控制器使用的键盘通常分为以下两种。其中一种是独立式，使用的时候每个键盘都要使用微控制器的一个端口，另一端接地。另一种是矩阵式键盘只占用微控制器一个端口，但是用起来相对复杂。本设计根据实际情况采用独立式键盘。

使用独立式键盘时，其中一端接微控制器的端口，另一端接地，微控制器输出的为高电平。当按键被按下的时候，就会让微控制器输出口就会接地，输出变成低电平。单片机检查I/O口状态就能够判断按键有没有被按下。

### 4.7.2键盘去抖动

按键一般都是机械开关。当按键按下时，按键由于各种原因不会马上就稳定接通，当按键断开时同理也不会马上就彻底断开。在闭合与断开之间，会发生一系列的抖动，如图4-9所示，可能会导致单片机多次响应。这种抖动并不能靠我们的按键方式所避免。所以必须保证在使用时，键盘的每次按下或者断开都只会让微控制器程序只执行一次，避免多次执行，于是需要进行去抖动。



**图4-9 按键抖动**

去除抖动一般分为两种方法。其中一种硬件消抖，所谓的硬件消抖是使用一个电容与按键进行并联连接，利用电容充放电的特性来去除抖动。但在实际应用当中这种消抖方法并不怎么好，还会让开发成本与电路的复杂度增加，所以实际应用当中一般选用软件消抖。

软件消抖并不是利用程序来将抖动消除，而是利用程序延时来忽略抖动。其中常用有效的，当微控制器检测到电平改变的时候，暂时先不执行程序，经过10ms上下来等待电平稳定，之后再来检测电平状态。若这时候的检测的电平与之前检测到的电平高低一样，就可以认为这次动作确实的执行了，并且电平已经稳定下来。按键的电路如图4-10所示.

****

**图4-10 按键电路**

# 5软件设计

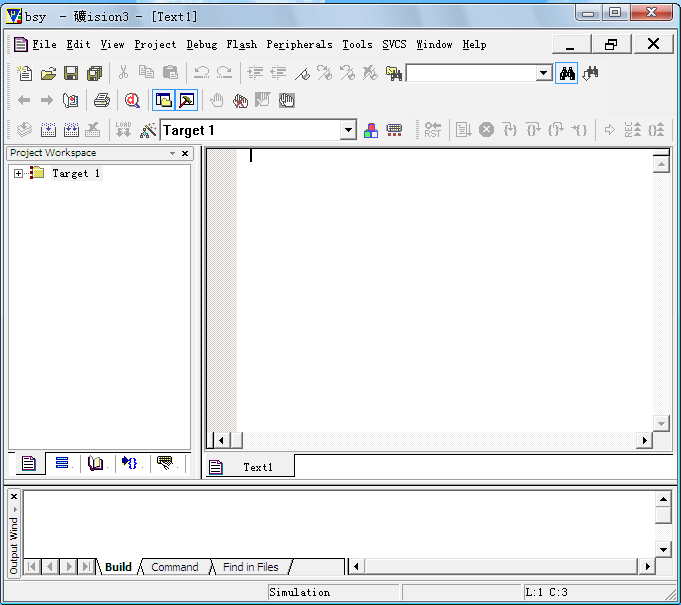
## 5.1使用软件介绍

### 5.1.1 Keil C51

本文采用了Keil C51来编写单片机源程序。Keil C51是一款由美国Keil Software专门开发的软件。它适合各个阶段的人来进行51单片机开发。这款软件可以让人在51系列及其兼容的微控制器系列上使用C语言进行编程。他适合各个阶段的人来进行51单片机开发。相对于汇编语言而言，使用C语言虽然在执行效率上可能略低于汇编语言，但使用C语言进行开发，在代码的可读性与可维护性上是强于汇编语言的。如果两者都使用过会明显的感受到C语言编写的便利性，因此本文使用C语言编写软件。

Keil C51自身就拥有数量可观的库函数方便人进行操作使用。另外还有一些方便使用的调试工具与拥有友好的可视化工作界面。虽然使用C语言等高级语言效率上不如汇编语言，但是如果使用过Keil C5，就会有一些改观。在看过Keil C5编译之后生成的汇编语言，就能知道它编译之后的汇编代码的效率很高。生成的汇编代码没有太多的多余，其中大部分语句容易理解。因此使用Keil C51编写大型程序对比汇编优势更加的明显。

Keil C51的界面如图5-1所示，是比较友好的图形界面。



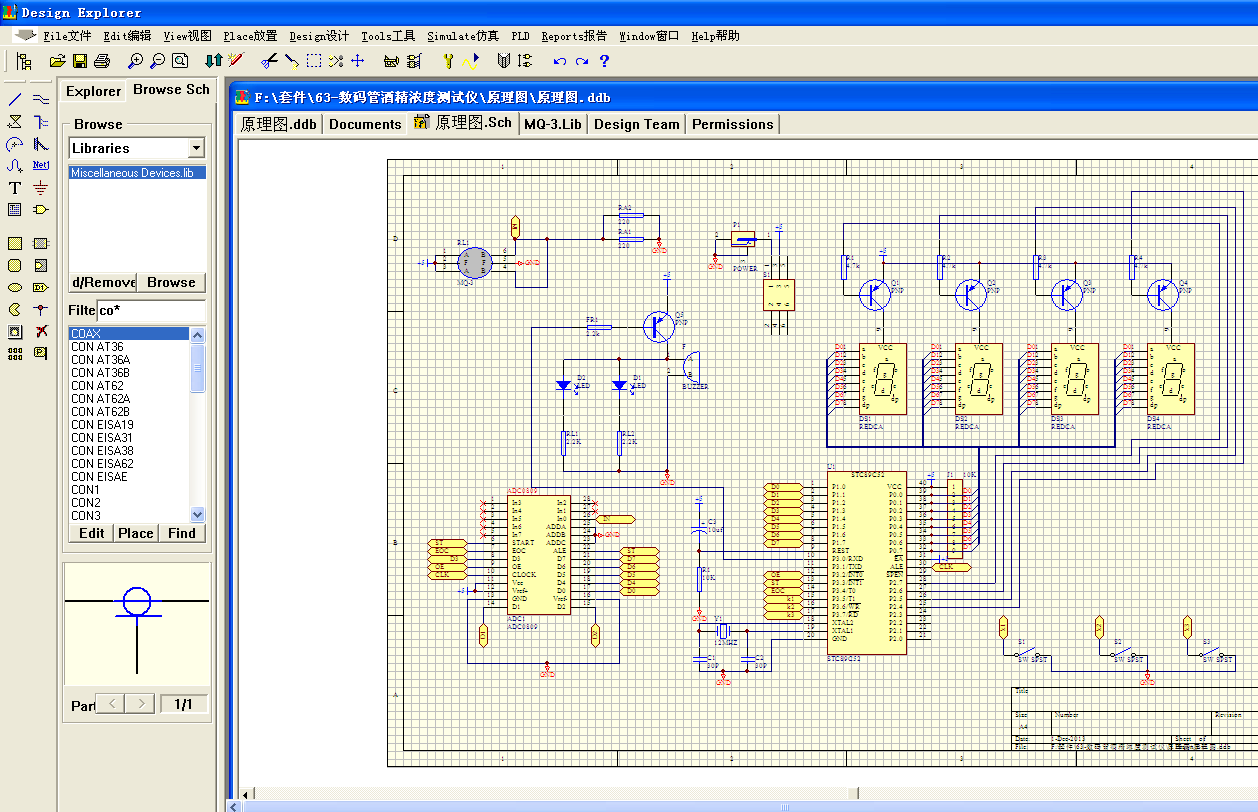
**图5-1 Keil c51软件界面**

### 5.1.2 Protel99SE

本文使用Protel99SE来进行电路设计。Protel系列软件是上世纪八十年代就诞生了的EDA软件。早在DOS时期，Protel公司就制作了许多EDA程序来让大家进行开发。随着PC的发展，随着图形界面的普及，Protel公司在windows平台上推出了Protel设计软件，这款软件程序稳定，非常好用。

作为许多人使用的Protel99SE是主要应用于可视化操作系统下的EDA软件，经过多年发展早已不是单纯的PCB设计工具。它主要有五个模块，原理图设计、自动布线器、PCB设计、PLD设计与原理图混合信号仿真。Protel99SE软件本身也有许多的特点，例如拥有许多标准电路板可以作为基础的设计模板使用；3D显示能够让人在制作出实物前就可以看到制作好后的效果等。

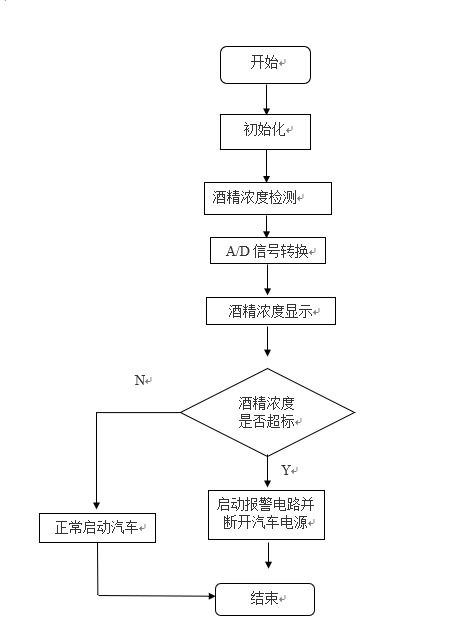
Protel99SE的工作界面如图5-2所示，是专门为了Windows这类图形界面操作系统开发的。



**图5-2 Protel 99se界面**

## 5.2主程序框图

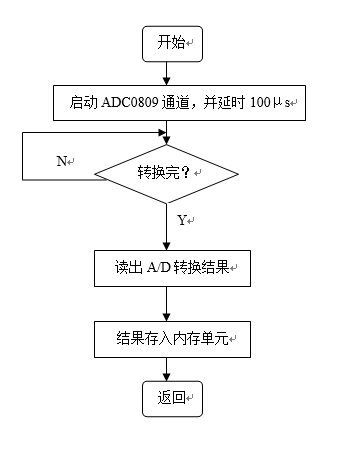
如图 5-3 所示，系统通电之后，首先需要自动进行初始化操作，在初始化完成之后，就会启动乙醇检测装置，开始对驾驶员呼出气体的乙醇浓度进行检测。检测的结果经过A/D进行转换，经过单片机处理之后显示在屏幕上。将所测到的值与之前设定的报警值进行比对，如果超过报警值则接通报警电路并且切断发动机电源，禁止汽车启动。若判断呼出气体中乙醇的浓度没有高于报警值，就会让启动汽车。



**图5-3 主程序框图**

## 5.3数据采集部分程序框图

A/D转换部分的流程图如下图5-4所示。

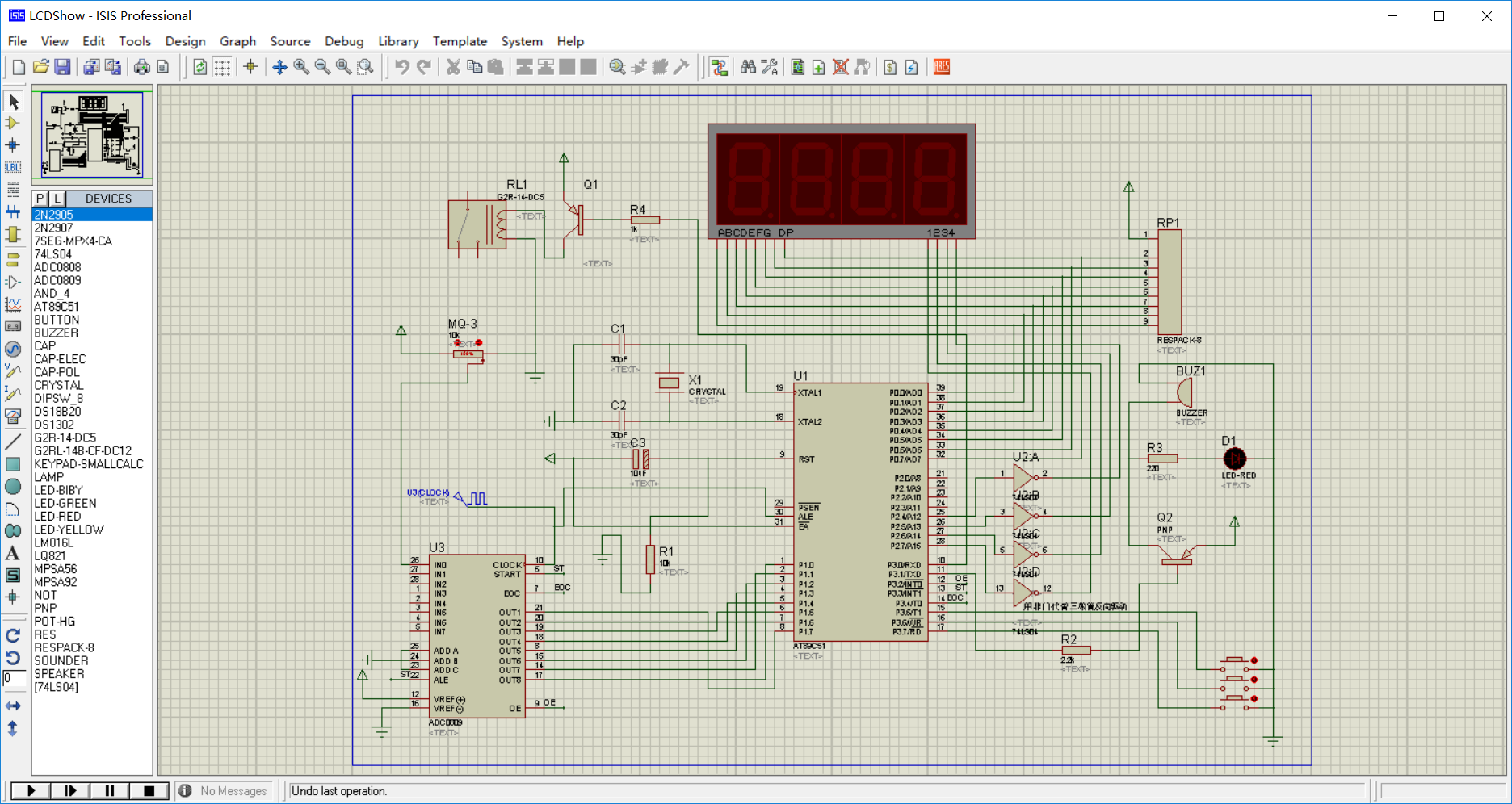


**图4-4 数据采集子程序框图**

# 6电路仿真

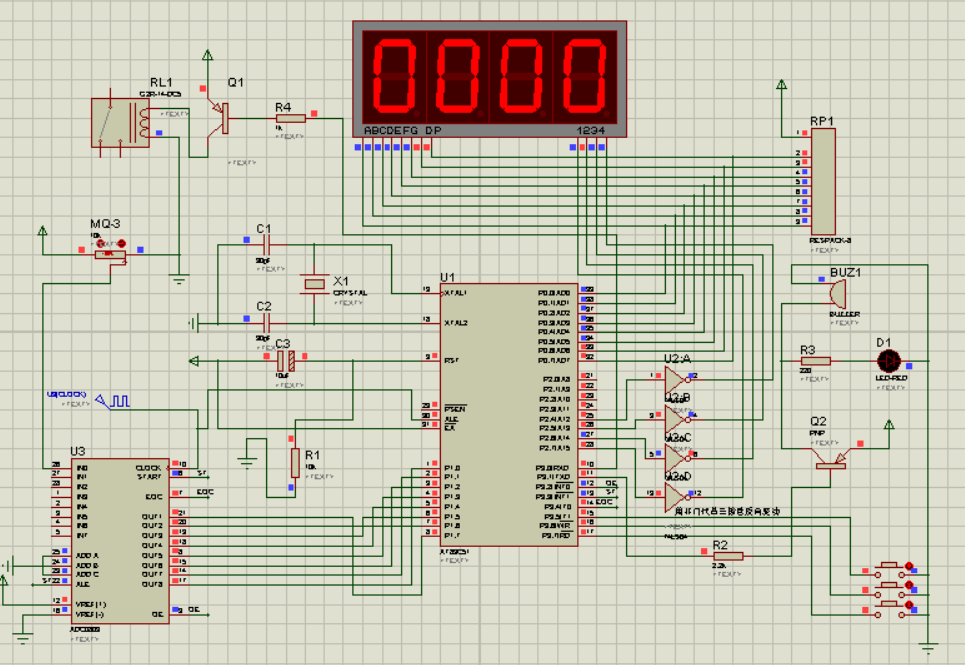
在使用Keil C51编写好程序并编译通过后，设计好电路图就可以开始进行电路仿真测试了。本文利用Proteus来对微控制器电路进行仿真测试。它拥有丰富的元件库，可以模拟设计制作出来后的实际情况。

Proteus进行仿真的原理图如图6-1所示，由于仿真中没有酒精传感器，而且也没办法模拟呼出酒精的情况。解决方法是暂时使用滑动电阻器代替MQ-3酒精传感器，模拟其根据自身环境中的酒精浓度的改变其电阻值。



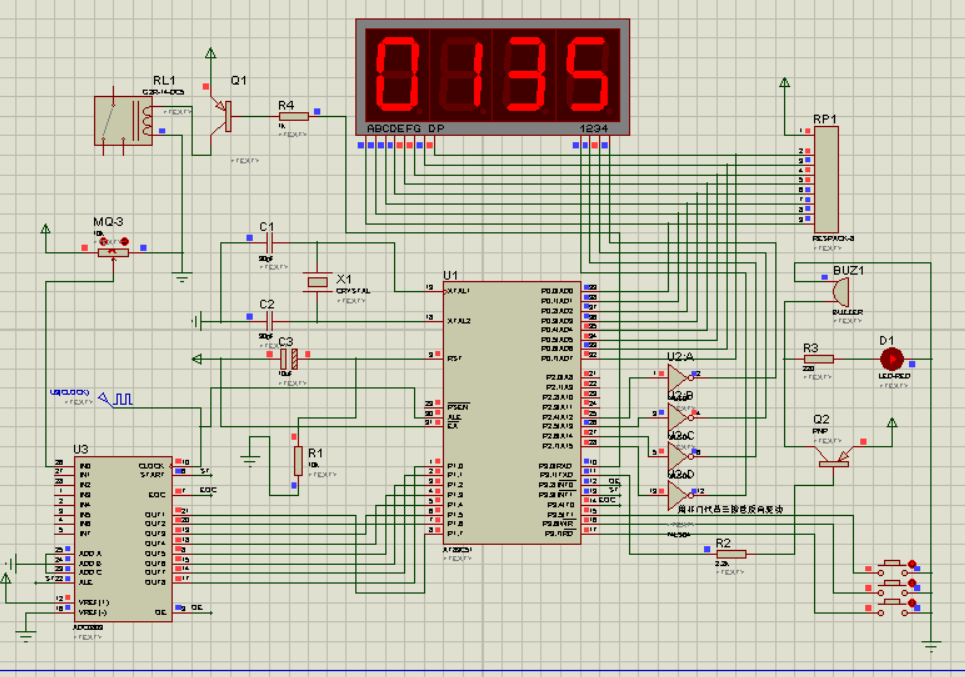
**图6-1 Proteus仿真电路原理图**

将之前在Keil C51中生成的hex文件导入到Proteus中，在debug中点start键之后再点击左下角三角键开始进行仿真模拟。



**图6-2 Proteus仿真正常情况**

如图6-2所示，通过使用滑动变阻器，我们可以对酒精浓度的高低进行操作。当乙醇浓度低于所设定值的时候，传感器输入引脚为红色高电平，报警灯处于熄灭状态。



**图6-3 Proteus仿真报警状态**

如图6-3所示，当呼出的酒精浓度高于所设定值的时候，传感器输入引脚和单片机控制继电器的引脚均为蓝色低电平。此时报警灯亮起，蜂鸣器开始工作，发动机电源被切断。

# 7 实物的制作与调试

制作实物看起来简单只需要将程序写入单片机，检查电路是否正确，然后根据电路将每个元器件焊接到PCB板上就可以了，如果自己开始亲自动手做了，就会发现并不是那么简单的事情。在焊接时要认真检查焊接的情况，有时候可能会因为一些原因虚焊，导致电路不通。这种问题不是很好检查，会耗费大量的时间。

在调试的过程中，遇到了一些问题，首先在焊接电路的时候，我没有查看好ADC8089的引脚，结果导致ADC8089整个器件接反了，又需要将其取下重新焊接好。程序在使用过程中也根据实际的情况做了一些修改，让其更好的在系统中运行，减少出错的问题。

在成功接通电路后，数码管显示酒精浓度非常很高，蜂鸣器与灯光也发出报警，后来经过一些查找才发现，MQ-3第一次使用的时候，需要先通电几小时才能正常使用。

# 结束语

大四即将步入尾声，校园生活即将结束，通过不断地的努力，我也终于完成了这款基于单片机的防酒后驾驶控制系统的设计。本设计造价便宜，电路简单，使用起来也方便，易于携带。

万事开头难，在决定毕业设计题目的时候就犹豫了很久，最后选定了这个题目。选定了题目之后却不知道如何入手。经过不懈的努力后最后完成了毕业设计，让人如释重负。

在完成毕业设计的过程中，我发现我懂得东西还是太少，需要学习的东西还是相当的多。这让我明白学习是永无止境的，俗话说逆水行舟，不进则退，在以后的工作当中，也需要不断地了解新的东西，努力提高自己的知识水平。虽然这个设计可能不太完善，但在这次设计过程当中我所获得的东西将让我受益终生。

# 致谢

光阴似箭，大学四年过得如此之快，马上就要离开学校，但这四年的学习生活让我受益匪浅。

首先要感谢徐盛老师，本文在耿鑫老师的辛勤教导下完成。在我进行选题，撰写论文的时候，他都为我提供了极大的帮助。在写作论文的时候，每次我有疑问的时候总会为我提供帮助。老师这种对工作不辞劳苦，在学术上的钻研精神都是能让我学习的。

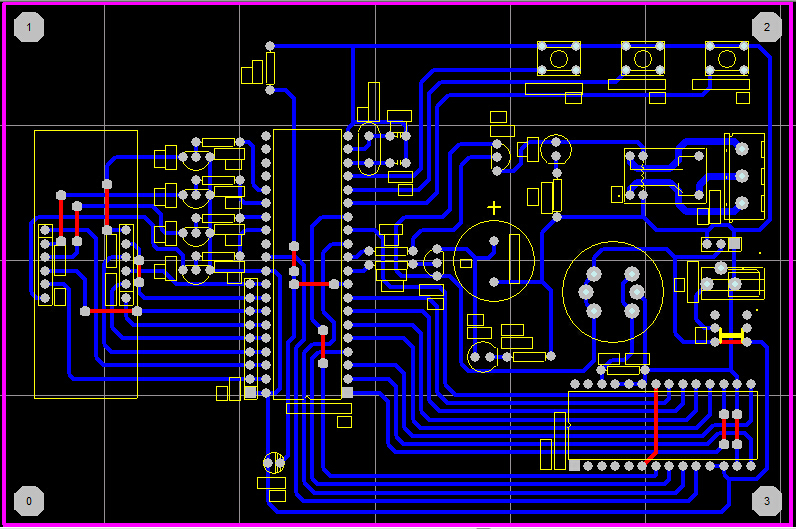
感谢在过去四年里教导我们的老师，是因为你们的耐心教导，让我学到了许多的专业知识。还要感谢这四年里一起生活的同学们，朋友们，感谢你们四年以来对我的帮助。

最后，我要感谢我的家人，是家人的帮助让我能够投入到学习之中，顺利的完成了自己的学业。也感谢自己生活了四年的母校，是母校为我提供了能让我认真学习的环境。

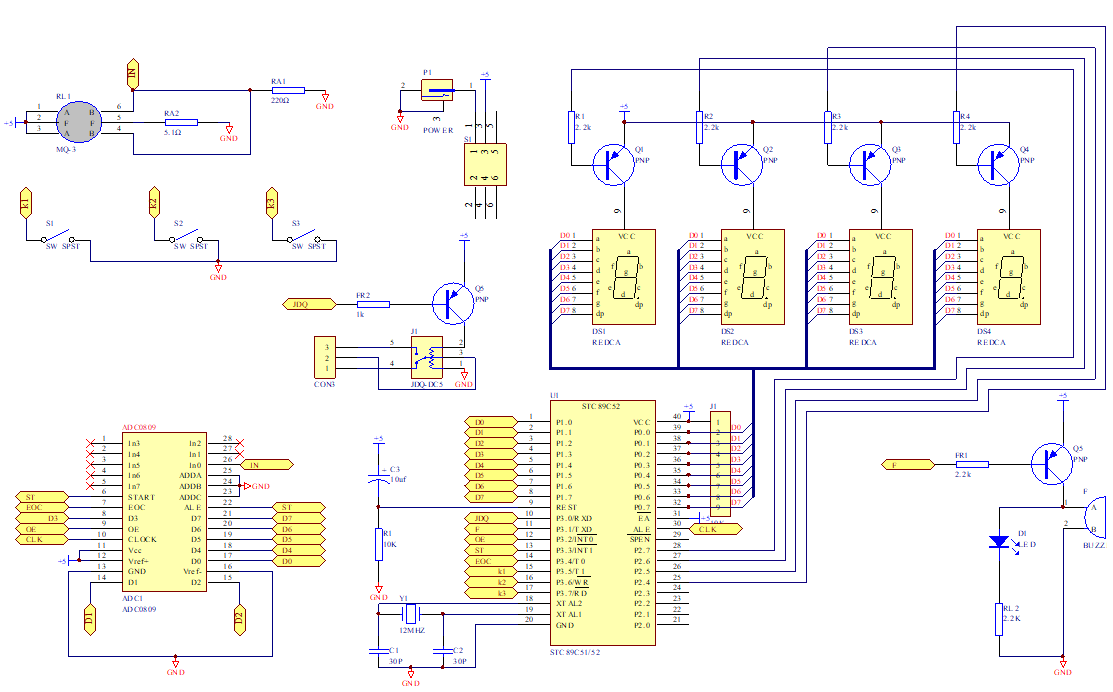
# 参考文献

# 附录

附录1 PCB板设计图



附录2 电路设计图



附录3 单片机C语言原程序

//程序头函数

#include <reg52.h>

//宏定义

#define uint unsigned int

#define uchar unsigned char

#define Data\_ADC0809 P1 //定义P1口为Data\_ADC0809

//管脚声明

sbit jdq=P3^0; //继电器

sbit Feng = P3^1; //蜂鸣器

//ADC0809

sbit ST=P3^3;

sbit EOC=P3^4;

sbit OE=P3^2;

//按键

sbit Key1=P3^5;

sbit Key2=P3^6;

sbit Key3=P3^7;

bit shan=0; //闪烁标志位

//显示数组 0-9+F

uchar Data\_[]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f,0x71,0x3f}; //共阴显示数组

sbit Wei1 = P2^7;

sbit Wei2 = P2^6;

sbit Wei3 = P2^5;

sbit Wei4 = P2^4; //四个数码管的位选端口

//函数声明

extern uchar ADC0809();

void Display(uchar X,uchar Data);

void delay(uint t);

//酒精含量变量

uchar temp=0;

//蜂鸣器变量

uchar FF=10;

//显示模式

uchar Mode=0;

uchar p;

void main() //主函数

{

TH0=0x3c;

TL0=0xb0;//赋初值50ms

TMOD=0x01;//设置工作方式

EA=1;//打开中断总开关

ET0=1;//打开允许开关

TR0=1; //打开定时器定时开关

while(1) //进入循环，括号内为1，所以程序进入循环就不会出来了，所以就叫死循环

{

//正常模式

if(Mode==0)

{

//读取AD值

temp=ADC0809();

for(p=0;p<30;p++) //调用三十次显示函数

Display(0,temp);

//判断是否报警

if(temp>FF) //大于报警值

{

jdq=0; //继电器吸和

}

if((shan==1)&&(temp>FF)) //闪烁变量为1时并且大于报警值

{

Feng=0; //蜂鸣器响

}

if(shan==0) //闪烁变量为0时

{

Feng=1; //关闭蜂鸣器

} //（闪烁变量shan受定时器控制，500ms取反一次，达到1秒响一下的效果）

if(temp<=FF) //小于报警值时

{

Feng=1; //关闭蜂鸣器和继电器

jdq=1;

}

}

//调整模式

else

{

Display(1,FF); //显示报警值界面

}

//功能键

if(Key3==0) //设置键按下（按键一端接地，一端接单片机IO口，当按键按下时，单片机的IO口就会检测到低电平）

{

Feng=0; //蜂鸣器响

jdq=1; //断开继电器

delay(100); //延时去抖

while(Key3==0) //判断按键是否按下

{

if(Mode==0) //正常模式

Display(0,temp);//显示正常界面

else //设置报警值

Display(1,FF); //显示报警值界面

}

if(Mode==0) //如果是正常模式，按下设置键，就切换到设置模式

Mode=1;

else //如果是设置模式，按下设置键，就切换到正常模式

Mode=0;

Feng=1; //关闭蜂鸣器

}

//增加

if(Key2==0&&Mode==1) //加键按下

{

Feng=0; //蜂鸣器响

delay(100);

while(Key2==0)

{

Display(1,FF); //显示设置界面

}

FF++; //报警值加

if(FF>=251) //最大加到250

FF=250;

Feng=1; //关闭蜂鸣器

}

//减少

if(Key1==0&&Mode==1) //减按键按下，注释参考加按键注释

{

Feng=0;

delay(100);

while(Key1==0)

{

Display(1,FF);

}

FF--;

if(FF==0xff) //报警值减到小于0

FF=0; //最小值就是0

Feng=1;

}

}

}

//ADC0809读取信息

uchar ADC0809()

{

uchar temp\_=0x00;

//初始化高阻太

OE=0;

//转化初始化

ST=0;

//开始转换

ST=1;

ST=0;

//外部中断等待AD转换结束

while(EOC==0)

//读取转换的AD值

OE=1;

temp\_=Data\_ADC0809;

OE=0;

return temp\_;

}

//延时

void delay(uint t)

{

uint i,j;

for(i=0;i<t;i++)

for(j=0;j<10;j++);

}

//显示 X表示状态 Data表示数据

void Display(uchar X,uchar Data)

{

Wei1=1;

Wei2=1;

Wei3=1;

Wei4=1;

P0=0xff; //关闭显示

//正常模式

if(X==0)

{

P0=~Data\_[11]; //显示0，也就是正常模式时，第一个位置显示0

}

//非正常

else

{

P0=~Data\_[10]; //设置模式，第一位显示字母F

}

Wei1=0; //选中第一位，其他位关闭

Wei2=1;

Wei3=1;

Wei4=1;

delay(10); //延时

Wei1=1; //关闭所有位

Wei2=1;

Wei3=1;

Wei4=1;

P0=~Data\_[Data/100]; //显示数值的百位，例如数据是123，除以100的商是1，这里显示的就是数字1

Wei1=1;

Wei2=0; //选中第二位

Wei3=1;

Wei4=1;

delay(10); //延时

Wei1=1; //关闭所有位

Wei2=1;

Wei3=1;

Wei4=1;

P0=~Data\_[Data%100/10];//显示数值的十位，数值是123，取余的作用就是除以100后的余数，是23,23再除以10得到的商是2，这里显示的就是2

Wei1=1;

Wei2=1;

Wei3=0;

Wei4=1;

delay(10);

Wei1=1;

Wei2=1;

Wei3=1;

Wei4=1;

P0=~Data\_[Data%10]; //显示数值的个位，数值是123，取余10就是数值除以10的余数就是3，这里显示的就是3

Wei1=1;

Wei2=1;

Wei3=1;

Wei4=0;

delay(10);

Wei1=1;

Wei2=1;

Wei3=1;

Wei4=1;

}

void time0() interrupt 1//定时器函数

{

uchar m;

TH0=0x3c;

TL0=0xb0;//重新赋初值

m++; //定时器进入一次是50ms，m加一次

if(m==10) //加10次，也就是500ms

{

shan=~shan; //控制shan变量取反，就是0变1,1变0

m=0; //m清零

}

}