

机电控制课程设计论文

## 交通信号灯控制系统

论文作者 孙楠

学 号 5080209271

小组成员 张力文 陈相帆 周游

指导教师 张银桥

2011 年 6 月

# 交通信号灯控制系统

## 摘 要

机电控制技术在生产生活中发挥着巨大的作用，尤其是以单片机为核心器件的控制系统因其功耗低、可编程性强、扩展能力强等优点应用日益广泛。交通信号灯是社会维持正常交通秩序的重要工具。信号灯对交通参与者的友好性及其自身的可编程性，在机动车数量迅猛增长的当下凸显其重要性。在机电控制技术课程学习之后，我们尝试利用单片机技术及相关软硬件技术架设一交通信号灯控制系统，整合倒计时、对闯红灯者拍照等功能，并实现红绿灯持续时间的简便可调。本文将对该系统的设计和功能使用做出说明，并讨论一些可能的拓展功能。

由于本人在课程设计过程中主要负责汇编程序的编写，本文也将更为侧重软件部分：程序设计的思路、主要代码的解释、关键问题的探讨等。同时也会较对设计过程中软件设计和硬件设计之间的相互影响做一些阐述。

**关键词：**单片机 控制 汇编

# 目 录

摘要	i
目录	ii
插图索引	v
表格索引	v
第一章 设计背景及任务	1
1.1 设计背景	1
1.2 设计任务	1
1.2.1 基本设计要求	1
1.2.2 扩展设计要求	2
第二章 硬件设计	3
2.1 硬件设计总体概况	3
2.2 数码管及信号灯驱动电路	4
2.2.1 数码管	4
2.2.2 信号灯驱动电路	4
2.2.3 数码管及信号灯的选通	5
2.3 键盘	5
2.4 地感线圈输入与中断触发	6
2.5 小结	7
第三章 系统功能设计	8
3.1 信号灯交替流程	8
3.2 可调的亮灯时间	8

3.2.1	预置数据表调节 . . . . .	9
3.2.2	用户自定义亮灯时间 . . . . .	9
3.2.3	双机通信调节 . . . . .	9
3.3	键盘屏显接口使用方法 . . . . .	9
3.3.1	键位定义 . . . . .	10
3.3.2	键盘设定操作流程 . . . . .	10
3.3.3	强壮性设计 . . . . .	10
3.4	闯红灯拍照功能 . . . . .	12
3.5	仍待加入的功能 . . . . .	12
附录 A 电路图		13
参考文献		14
致谢		15

## 表格索引

3-1 键位定义 . . . . .	10
3-2 键盘输入处理 . . . . .	12

## 插图索引

2-1 电路图 (小)	3
2-2 数码管驱动电路	4
2-3 信号灯驱动电路	5
2-4 键盘电路	6
2-5 地感线圈输入电路	6
3-1 亮灯顺序循环图	8
3-2 键盘操作流程圖	11
A-1 电路图	13

# 第一章 设计背景及任务

## 1.1 设计背景

交通信号灯是城市交通行为中重要的环节，它带来有秩序的交通，引导人流车流、缓解拥堵、提高通行效率。

传统的交通信号灯简单地实现红黄绿等的切换，时间间隔难以调节，高峰时间和夜间红绿灯的间隔不变，更不可能根据不同路口的车流量做出调节。单调不变的红灯会加剧人的焦急感，人们要求信号灯能够有显示倒计时的功能。另外，对于闯红灯、超速等交通违规现象的监控，如需要整合到路口信号灯系统中。

要控制具有如上所有功能的交通信号灯系统，单片机因其运行速度快、功耗低、成本低廉、可编程性优越，接口丰富等优点成为控制核心的必然选择。

## 1.2 设计任务

结合设计背景，要求设计一个以 S52 单片机为核心的交通信号灯控制系统，同时对闯红灯车辆进行拍照。

### 1.2.1 基本设计要求

根据交通规则设计

1. 十字路口交通信号灯分为红灯、绿灯和黄灯，交替亮灭，保证车辆安全有序通行；
2. 亮灯时间可以设置；
3. 路口安装摄像监控装置，对于闯红灯车辆进行拍摄，存入录像机；
4. 操作简单。

### 1.2.2 扩展设计要求

在基本设计要求的基础上，结合日常的生活经验、以及我们对单片机技术的掌握，我们认为最终设计的系统还具有以下功能:

1. 可以实时显示两组红绿灯倒计时时间;
2. 相交的两条路应有不同的红绿灯时间;
3. 亮灯时间可以根据当前所处的时段自动调整;
4. 亮灯时间可以通过监控车流量等信息的上位机智能调节;
5. 同一路口的红绿灯之间有互锁控制;
6. 设置界面应友好易用。



## 第二章 硬件设计

本文将主要介绍软件部分的设计，通过对软件设计方案的描述，实现对整个系统功能的介绍。但是硬件部分仍是整个系统的基础，与软件设计也有着密切的关系。本章将对基本的硬件设计做一个浏览。并重点介绍与软件相关最为密切的部分。

### 2.1 硬件设计总体概况

本系统采用 AT89S52 单片机为控制核心，晶振电路产生 11.0592 MHz 的晶振频率，具有复位电路。输入部分有四个地感线圈信号（接 P1 口低四位），五个键盘按键输入信号（接 P2 口高五位）。输出部分驱动六个数码管，六个信号灯，四个拍照驱动信号。图2-1是控制部分电路图，附录A图A-1为其放大版。

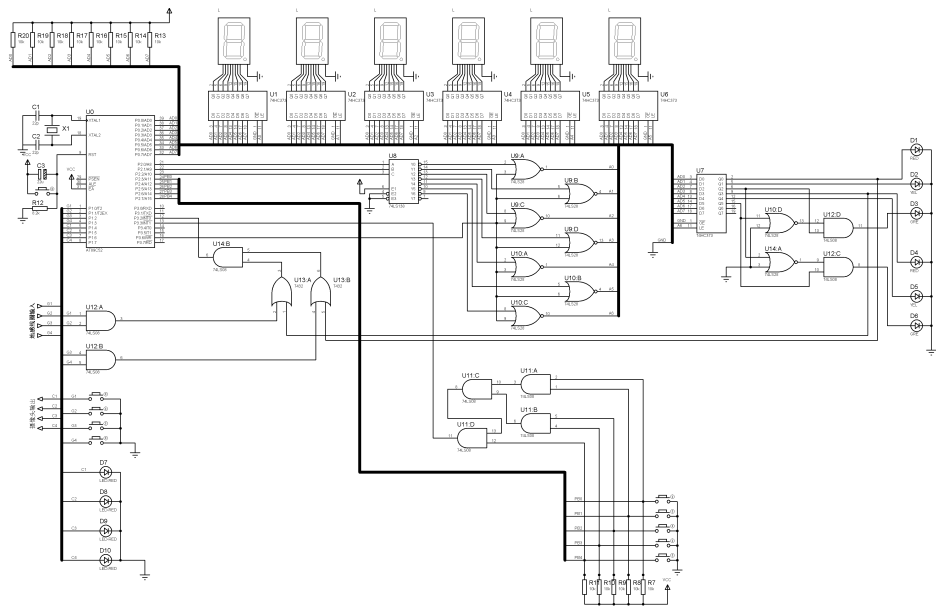


图 2-1 控制电路图

Fig 2-1 Control Circuit Figure

要实现规定的功能目标，通过估算，另外也考虑到 S52 单片机内部存储器

足够大（8K），而最终生成的程序为 1.32Kb，RAM 空间也有剩余，故不需要进行存储器的扩展。

地感线圈处的输入信号需通过继电器与控制电路连接，具体的连接参数视地感线圈而定。信号灯与数码管的控制信号需要通过光电耦合器后放大再驱动相应器件。这些电路，在以上的控制电路图中并没有体现，但在完整的系统中，它们是同样重要的。

2.2 数码管及信号灯驱动电路

2.2.1 数码管

数码管采用共阴数码管。为了提高显示质量，每一个数码管均通过一个 373 锁存器驱动，共六个。这样做可以实现静态显示，数码不会闪烁。软件实现也大为便捷，CPU 占用率也降低许多。图2-2是数码管部分的电路图。

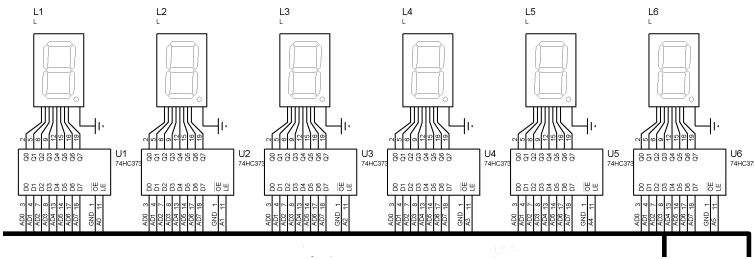


图 2-2 数码管驱动电路

Fig 2-2 Digital Led drive circuit

2.2.2 信号灯驱动电路

信号灯驱动电路如图2-3所示。因为相对的路的信号灯内容相同，所以两条路且每条路三盏灯，共六盏灯，通过一个 373 锁存器驱动。图中可见互锁电路由两个或非门（非门）和两个与门组成，则两个绿灯不能同时点亮。尽管软件中避免了两盏绿灯同时点亮的情况的出现，但硬件互锁还是必不可少的，毕竟软件在运行过程中可能会受到干扰而混乱，可能出现不可预料的情况。

图中所绘的小灯为演示所用，实际应将输出信号放大后，再驱动高电压的信号灯。

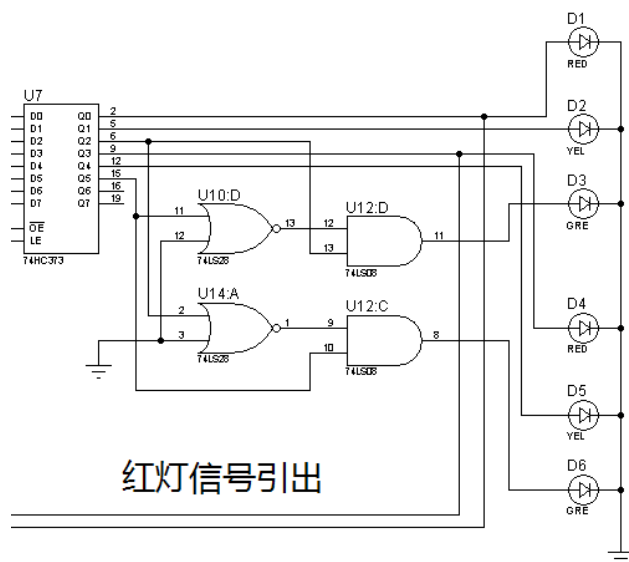


图 2-3 信号灯驱动电路

Fig 2-3 Signal light drive circuit

图中还可以看到两个红灯处引出了两路信号，这样做的目的将在2.4节详细阐述。

### 2.2.3 数码管及信号灯的选通

P0 口作为数据总线和七个 373 锁存器相连，P1 口的低三位作为地址线，再经过一个 38 译码器选通各锁存器。单片机的 WR 信号与 38 译码器的七个输出端分别或非之后再选通锁存器。在对方案的重新评估中，我们发现也可以令 WR 信号直接选通 138 译码器，但考虑到译码器的输出端仍需通过非门，器件数量并未减少太多，故仍保留最初的设计。

## 2.3 键盘

图2-4是键盘的电路图。因为 P2 口正好还有五个 IO 口可用，五个键也正好可以实现设置的要求，所以采用的独立键盘。这样对于编程工作也是很大的便利。各键的定义在3.3.1节介绍。

按键触发后 P2 口有下降沿信号，这个信号全部相与后接 P3.3 ( $\overline{INT1}$ )，可产生中断信号，在软件中再通过轮询确定具体按下的键位。

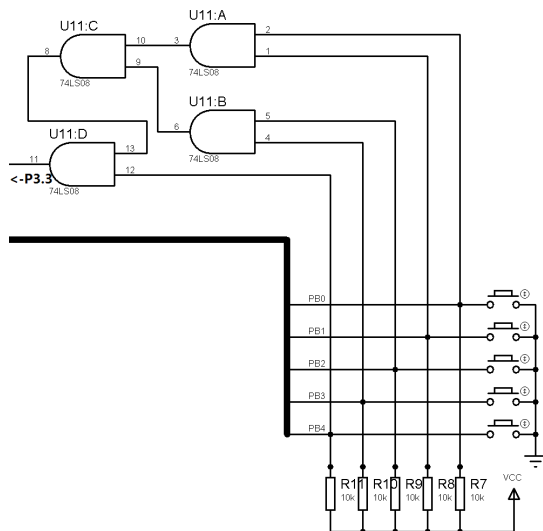


图 2-4 键盘电路

Fig 2-4 Keyboard circuit

## 2.4 地感线圈输入与中断触发

地感线圈输入信号送入 P1 口低三位，并触发  $\overline{INT0}$  中断。在电路图中地感线圈的输入以开关代替，实际信号也是一开关量，但由地感线圈转换至单片机控制电路输入信号还需要一些处理。

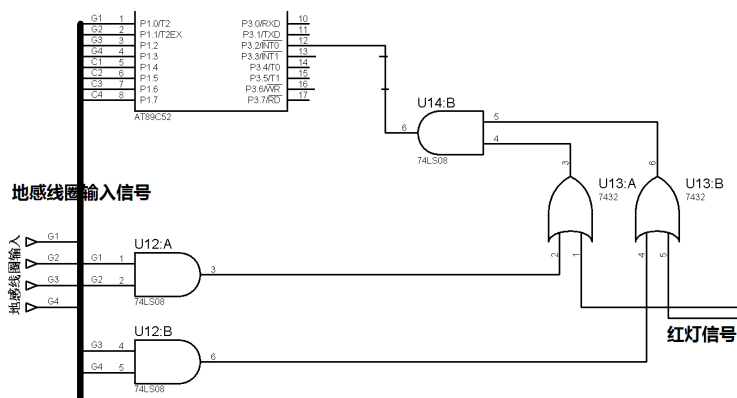


图 2-5 地感线圈输入电路

Fig 2-5 Ground wire input

由图2-5可见，此电路与普通的外部中断扩展电路稍有不同，在2.2.2节已

提到，两路红灯信号由右侧引入，分别与另一条路的地感线圈输入信号取或运算，然后在送入  $\overline{\text{INT0}}$  口。这是为了在某一路为绿灯时，屏蔽其地感线圈的信号，使之不能产生中断。只有在某一路为红灯时，闯红灯的车辆才能触发地感线圈，继而触发单片机中断。这样大大减少了中断的产生大大降低了 CPU 的工作量，提高了反映速度。编程难度也有所减轻。

## 2.5 小结

除了以上介绍的主要输入输出电路外，我们还设计了上电复位电路，晶振频率为 11.0952MHz 的时钟电路，P0 口加入了上拉电阻。由于这些电路都较为常规，在此不再详述。

通过以上的分析可见，我们的设计对单片机 IO 口的利用非常充分 P0, P1, P2 口均已占满。通过充分利用 IO 口实现了数码管和信号灯的静态显示、独立键盘、独立的地感线圈输入，减少了元器件的使用，减小了编程难度。

硬件电路设计中也通过一些逻辑电路的组合，实现了互锁、条件屏蔽等功能。

### 第三章 系统功能设计

本章起到一个功能说明书的作用，将抛开底层的硬件和具体的软件实现，面向普通用户，介绍本系统的主要功能、工作流程，以及设置界面的使用方法。

#### 3.1 信号灯交替流程

红黄绿三色信号灯按照图3-1所示的工作顺序交替闪亮。由于相交的两条路交通状况可能相差很大，其红灯持续时间是不同的。两个红灯时间  $T1$  和  $T2$  可以分别设置<sup>1</sup>，黄灯时间默认为 3 秒。由图可见一个亮灯循环只需要两个独立参数（ $T1$  和  $T2$ ）就可以确定，绿灯时间比相应红灯时间短 3 秒。

路口 A	绿灯	黄灯	红灯 T1
路口 B	红灯 T2	绿灯	黄灯

图 3-1 亮灯顺序循环图  
Fig 3-1 Signal light changing pattern

各个路口均悬挂有倒计时显示器，两组相对的路口显示内容分别相同。倒计时的内容为当前亮的灯将要持续的时间。当显示完 00 之后，灯的状态改变，倒计时显示屏也重新开始计数。

#### 3.2 可调的亮灯时间

所谓亮灯时间的可调性，就是指图3-1中的循环参数  $T1$  和  $T2$  是可变的，这是为了适应不同的交通状况，车流量大时缓解拥堵，车流量小时减少等待。调节亮灯时间的方式有以下三种。

<sup>1</sup>下文将称这两个参数为亮灯时间参数

### 3.2.1 预置数据表调节

在单片机的 **ROM** 中预先存放了一数据表，一天中的每小时<sup>2</sup>都有存储了两个数据在表中，分别代表 **A** 路口和 **B** 路口的红灯时间。则此表格共包含 48 个数据。单片机会根据当前的时间段<sup>3</sup>，自动切换亮灯时间。

### 3.2.2 用户自定义亮灯时间

本系统还提供了一人机界面（包括 5 个按键和 2 位数显示屏）供用户随时调节亮灯时间。用户可以调节任一时间段的亮灯时间，也可以重新定义当前所处的时间段。若用户调节了当前时间段的亮灯时间，则在当前红灯倒计时变为 0 后立即生效。3.3 节会详细介绍该方式的使用方法。

时间参数存储在 **RAM** 中，单片机复位后会全部清零。用户定义的时间参数优先级比内置参数表高。

### 3.2.3 双机通信调节

单片机系统中留出了接口，上位机可以通过串行接口向单片机发出指令，直接改变当前时间段的亮灯时间参数。上位机可以是自动监控车流量的单片机，也可以是交通管理网络中的计算机。这种调节方式和通过键盘的调节方式，后设置的参数会覆盖之前覆盖的参数，所以它们的优先级是一致的。

## 3.3 键盘屏显接口使用方法

用户通过键盘可以实现两个功能。

1. 设定任一时段两路口的红灯时间参数，该参数存入 **RAM** 中。
2. 当用户设定的数据为两个 00，则更改目前的时间区段为用户设定的值，这一功能可用于在单片机复位后初始化当前时区。

在设定过程中，数字显示的相应位会闪烁，提示用户目前正在操作的位的位置。

<sup>2</sup>一天中划分为 24 个时间段，从 0 到 23，分别对应一天中的第一到第 24 个小时。

<sup>3</sup>单片机复位后，用户可通过键盘设定初始时区为当前时间段，如图 3-2，之后单片机即从此时区开始自动累加

3.3.1 键位定义

表3-1详细列出了各个按键的定义，了解了这张表基本就可以尝试操作了。但是由于屏显只有两位数码管，刚开始也许还会有些费解，还需要参考3.3.2节给出的一个完整的设置流程图。

表 3-1 键位定义

Table 3-1 keyboard defination

键名	端口	功能
开关 on/off	p2.3	开关显示屏。开屏则首先显示当前时间段代码。关屏则重置键盘接口，清空所有寄存器。
确认键 ok	p2.4	确认当前选择，并前进到下一选择界面
选位键 <>	p2.5	切换选择高低位。在设定数码时，用于在个位十位间切换，被选择的位会闪烁。设定时间段时，该键不起作用。
加一键 +	p2.6	使屏显的内容加一。在红灯时间调节状态下，对闪烁的位加一。若超出最大值，则变为最小值。
减一键 -	p2.7	使屏显的内容减一。在红灯时间调节状态下，对闪烁的位减一。若小于最小值，则变为最大值。

3.3.2 键盘设定操作流程

图3-2 是键盘相关的完整流程图。从图中可以明确看到，通过键盘设定可以完成两个功能。

需要补充的一点是，在成功设置某一时段的时间参数后，程序将检查设置的是否就是当前时段，如果是的话，会重新载入该时间常数。故更改会在当前红灯倒计时完毕后立即生效。

3.3.3 强壮性设计

如果用户向信号灯系统定义了两个 0 为时间参数，显然这样的定义是没有意义的，通过软件已经将这样的输入转换为对当前时间段的重定义。本节就将



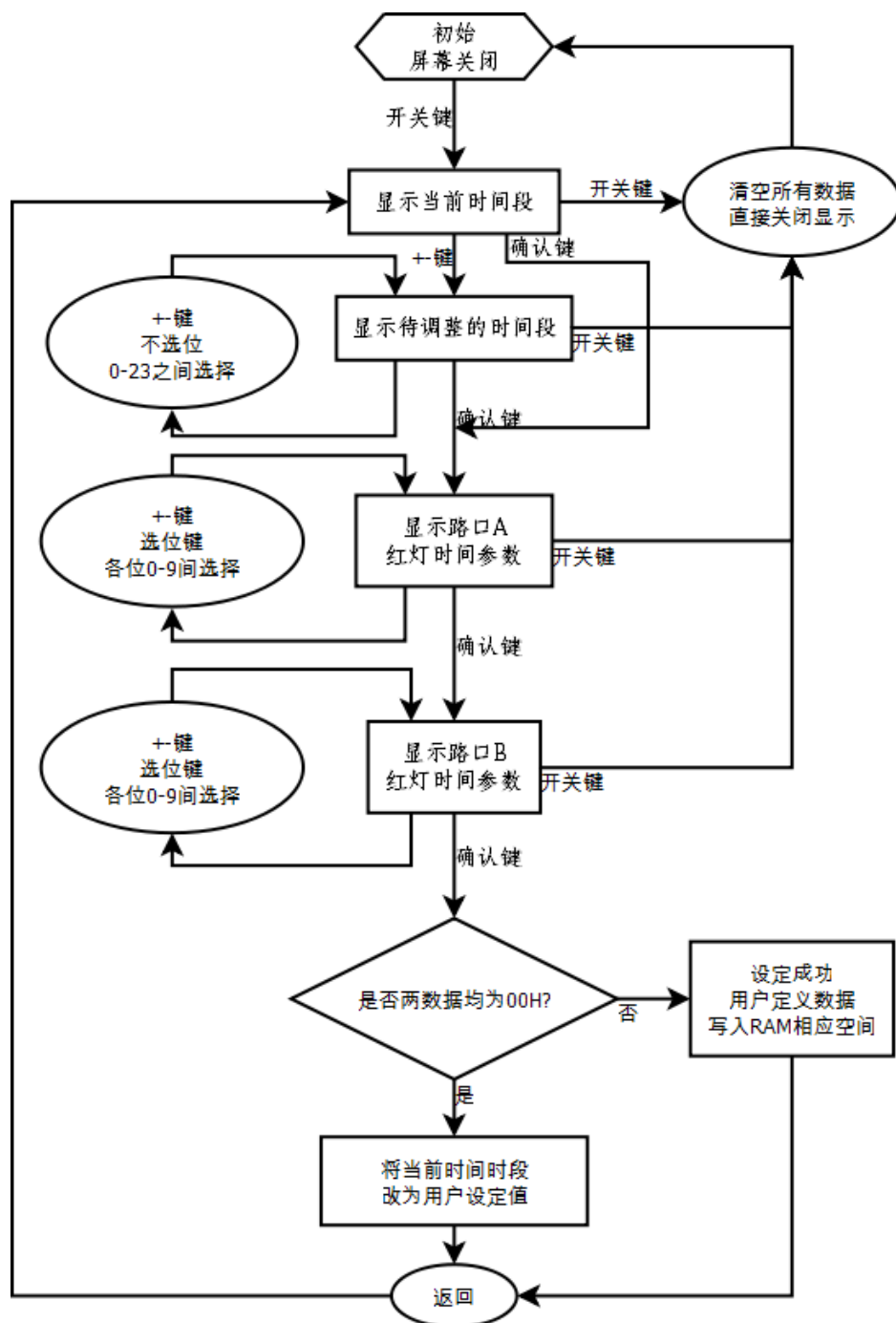


图 3-2 键盘操作流程

Fig 3-2 keyboard control flow figure

讨论用户对用户各种输入数据的处理。

因为黄灯时间需要持续三秒，所以红灯时间必须大于 3 秒，任何小于三秒的定义也是无效的。

表 3-2 键盘输入处理  
Table 3-2 keyboard input interpretation

用户输入		相应处理
T1	T2	
0	0	设置当前时间段
$\geq 0, < 3$	$\geq 0, < 3$	无效输入，不操作
$\geq 0, < 3$	有效输入 T2	T2 写入 A 路口对应 RAM
有效输入 T1	$\geq 0, < 3$	T2 写入 A 路口对应 RAM
有效输入 T1	有效输入 T2	写入 T1
		写入 T1
		写入 T2

表3-2列出了软件中对于各种输入情况的处理。由表可见，程序会自动消除用户的错误输入。默认的修正原则是：如果两个数据都错误则抛弃；如果其中有一个数据有效，则另一个路口的时间参数取相同的值。

3.4 闯红灯拍照功能

本系统还具有对闯红灯的车辆拍照的功能。需要说明的是，四个路口分别布置了地感线圈，只有在该路为红灯状态时，地感线圈发出的信号才有效。系统可以响应两个相对的路口同时闯红灯的状况，虽然这种情况的几率并不大。

3.5 仍待加入的功能

待完成

## 附录 A 电路图

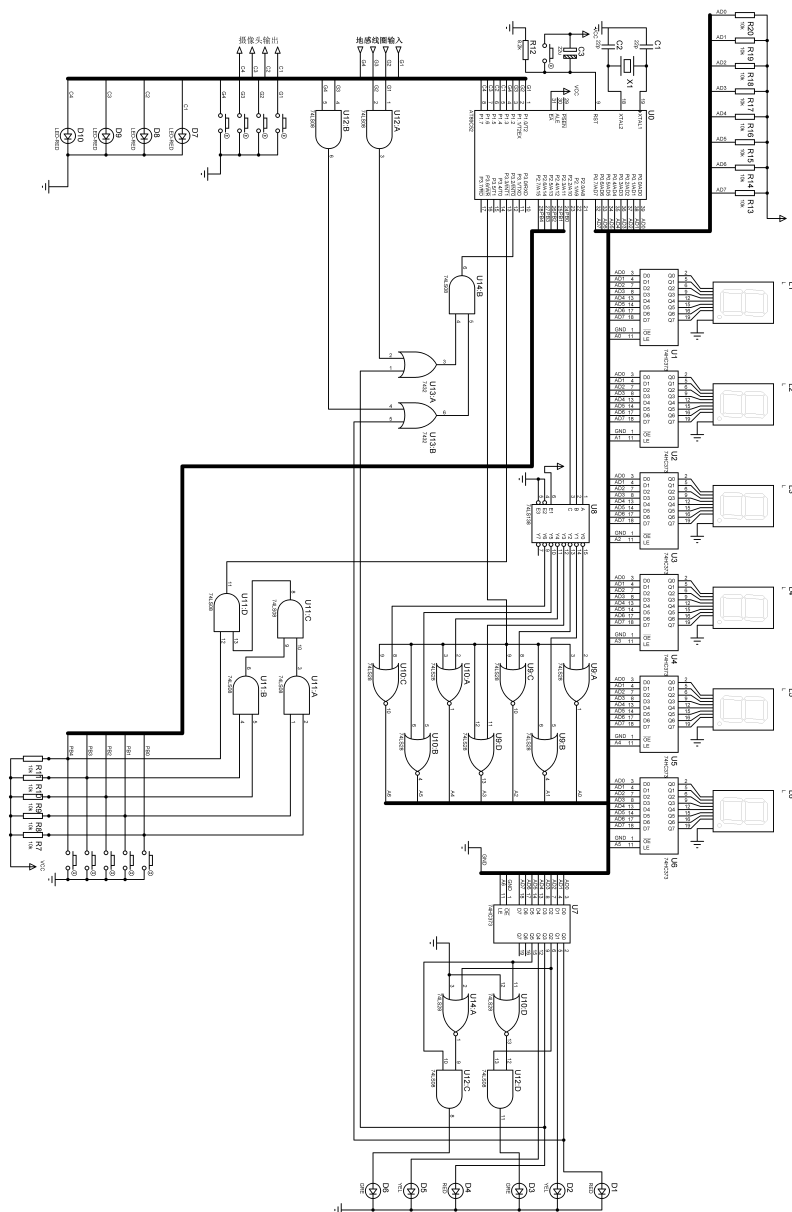


图 A-1 电路图

Fig A-1 Circuit Map

## 参考文献

## 致 谢

感谢张银桥老师的精彩授课和您在课程设计过程中给予的热心指导!

感谢张力文、陈相帆、周游同学在课设中给我的帮助和启发,和你们合作是本次课设工作圆满完成的关键!

感谢 Keil, PROTEUS 等软件的开发者,这些软件使得开发工作的难度和工作量大为降低。

感谢 L<sup>A</sup>T<sub>E</sub>X 给文档编写工作带来的便利,也感谢 William Wang 同学对 L<sup>A</sup>T<sub>E</sub>X 模板移植做出的巨大贡献。