

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 系统功能设计	1
1.1 信号灯交替流程	1
1.2 可调的亮灯时间	1
1.2.1 预置数据表调节	2
1.2.2 用户自定义亮灯时间	2
1.2.3 双机通信调节	2
1.3 键盘屏显接口使用方法	2
1.3.1 键位定义	3
1.3.2 键盘设定操作流程	3
1.3.3 强壮性设计	3
1.4 闯红灯拍照功能	5
1.5 仍待加入的功能	5
第二章 软件设计及问题研讨	6
2.1 程序概览	6
2.1.1 红灯时间参数的存储和调用	6
2.1.2 中断的应用	6
2.1.3 子程序一览	8
2.2 问题研讨及程序分析	10

附录 A 电路图	11
参考文献	12
致谢	13

表格索引

1-1	键位定义	3
1-2	键盘输入处理	5
2-1	时间参数表	7
2-2	中断应用表	8

插图索引

1-1 亮灯顺序循环图	1
1-2 键盘操作流程圖	4
A-1 电路图	11

第一章 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。本章也作为下一章软件设计的先导，软件设计都是围绕着以下功能的实现而进行的。

合理的硬件设计也是以下功能实现的重要基础。

1.1 信号灯交替流程

红黄绿三色信号灯按照图1-1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 T_1 和 T_2 可以分别设置¹，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（ T_1 和 T_2 ）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 T_1
路口 B	红灯 T_2	绿灯	黄灯

图 1-1 亮灯顺序循环图

Fig 1-1 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。

1.2 可调的亮灯时间

所谓亮灯时间的可调性，就是指图1-1中的循环参数 T_1 和 T_2 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。

¹下文将称这两个参数为亮灯时间参数

1.2.1 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²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³，自动切换亮灯时间。

1.2.2 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。1.3 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。

1.2.3 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。

1.3 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。

²一天中划分为 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。

³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 1-2，之后单片机即从此时区开始自动累加

1.3.1 键位定义

表1-1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1.3.2节给出的一个完整的设置流程图。

表 1-1 键位定义

Table 1-1 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。

1.3.2 键盘设定操作流程

图1-2 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。

1.3.3 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

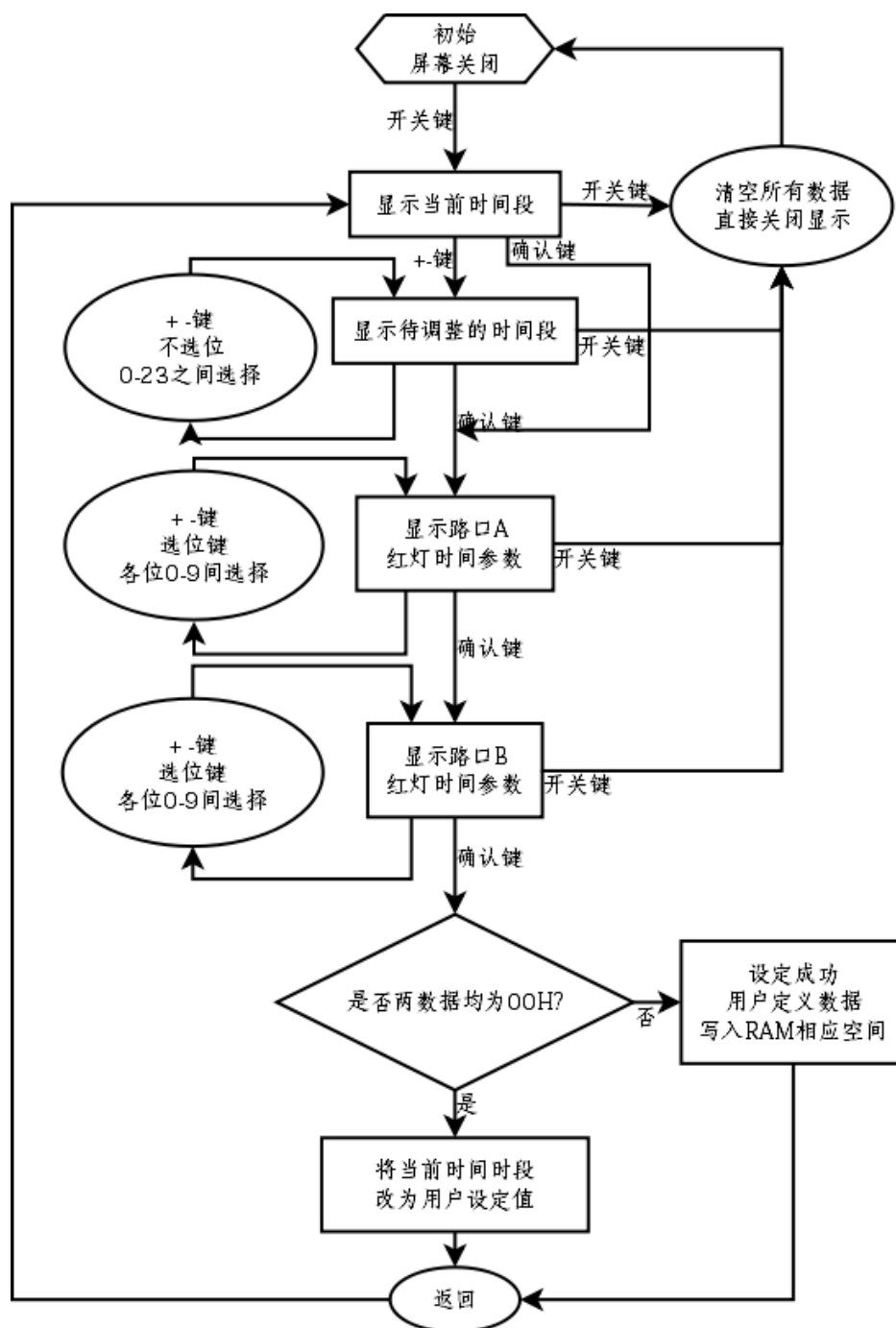


图 1-2 键盘操作流程

Fig 1-2 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 1-2 键盘输入处理
Table 1-2 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1-2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。

1.4 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。

1.5 仍待加入的功能

待完成

第二章 软件设计及问题研讨

本章是本篇论文的重点，因为本人在本次课程设计工作中主要做的的就是程序设计，对于这一部分有许多的了解最为深刻。首先我将阐述程序设计的基本思路，对重要的变量和子程序做解释。另外还将就程序设计中一些比较关键的问题做分析。

2.1 程序概览

本程序用全部用汇编语言编写，整体大约六百四十余行，编译后约占1.3Kb。附录??提供了完整的源代码。

2.1.1 红灯时间参数的存储和调用

本程序中最重要参数就是两个路口的两个红灯时间参数，倒计时、切换灯的状态、设置时间参数等都是围绕这两个参数进行的。要解释清楚该程序，首先就需要解释这个参数的不同形式，和它们的存储和调用方式。表2-1列出了相关的参数作用和存储方式。在这些存储空间中，数据都是以BCD码的形式存储的，这是为了用户输入和显示的方便¹。

这个表中数据是从底部行流向顶部行描述的存储空间的。而最后两行展示了ROM表与RAM表之间的优先级关系。

2.1.2 中断的应用

为了提高程序响应的速度，减少CPU的工作量，所有功能都移入中断处理程序中，主函数只起到初始化的作用。表2-2列出了中断的应用方式。中断优先级的设置将在??中讨论。此外，若要实现双机多机通讯，串口中断自然也需要应用。

¹但是本程序中标志时区的量都是以十六进制的形式存储的，这是因为时区参数与查表有关，而表格时连续的，16进制的时区参数对于查表是便捷的。

表 2-1 时间参数表
Table 2-1 Time registers

存储形式	参数功能
R2 R3	分别存储当前两个倒计时数码管组显示的内容，R2 为红灯，R3 为绿灯 (或黄灯)
30H 31H	存储当前时段两个红灯时间参数，倒计时到 0 后，R2 和 R3 会从这两个字节重新载入倒计时时间。三种情况改变这两个字节的值： <ul style="list-style-type: none">• 当一个小时结束时;• 用户通过键盘定义了当前时段的时间参数时，立即刷新，立即生效;• 上位机发送包要求改变这两个位时。
ROM 中的表 #TAB_LIGHT_TIME	是预置在 ROM 中的时间参数表，按小时排列，每小时两个，共 48 个字节。当一小时计数到时，30H 和 31H 从这个表中载入新的数据。
RAM 中 50H 到 7FH	这 48 个字节存储了用户通过键盘定义的时间参数数据。它的优先级比 ROM 中的表要高。但如果这个表中的数据为 0。则视为未定义，程序仍从 ROM 表中读取数据存入 30H 和 31H。

表 2-2 中断应用表

Table 2-2 Interrupt vectors

中断优先级	中断名	中断处理程序	中断功能
1	计时器 0	INT_T0	计时 $\frac{1}{144}$ 秒，清零看门狗，取反计数器 2 的输入端
2	计数器 1	INT_C1	计数 36，即 0.5 秒，中断处理程序完成倒计时等工作
3	外部中断 0	INT_EX0	响应地感线圈输入信号
4	外部中断 1	INT_EX1	响应按键输入信号

2.1.3 子程序一览

序号	子程序名	主要功能	主要参量	包含子程序
1	INT_T0	清零看门狗，取反计数器 2 的输入端		
2	INT_C1	倒计时减一显示，切换灯状态，计时，重载时间参数	R0 灯状态，R6 秒计数，R7 分计数	SUBBCD, CLOSEDIG, DIS-PLAY_NUMBER, CHANGE-LIGHT, GET_TIME_LIGHT 等
3	INT_EX0	响应地感线圈输入信号，向摄像头输出开关量控制信号		DELAY10
4	INT_EX1	响应按键输入信号，完成键盘操作流程所要求的功能。	R4 存储当前屏幕状态（显示内容的标志），3AH 当前选择的时间段，3BH 路口 A 的正在被设定的数据，3CH 相应的路口 B 的数据位 7F 标志当前调整的是高位还是低位	DELAY10, DELAY4MS5, HEX2BCD, DIS-PLAY_NUMBER, GET_TIME_LIGHT, BCDINC, BCD-DEC
5	BCDINC	根据选位标志 7F 对高位或低位做加一运算，对 9 加一得到 0	3DH 传递被处理的数据	

6	BCDDEC	根据选位标志 7F 对高位或低位做减一运算, 对 0 减一得到 9	3DH 传递被处理的数据	
7	CLOSEDIG	根据选位标志, 关闭某一位的显示	3EH 传递被处理的数据	通过调用 DISPLAY _NUMBER 送 显示
8	HEX2BCD	将 16 进制数转为 BCD 码, 主要用于时区的显示	38H 传递被处理的数据	
9	GET TIME LIGHT	根据当前的时区, 刷新 30H 和 31H 中的数据	3DH 传递被处理的数据	
10	CHANGE LIGHT	根据目前的灯状态, 选通灯锁存器, 点亮相应灯	R0 指向存储亮灯数值的单元 (32H31H)	
11	DISPLAY NUMBER	将 BCD 码显示到数码管上, 一次显示相邻的两位	38H 存放 BCD 码, 39H 存放低位数码管的地址	GETDIGIT, DISDIGIT
12	GETDIGIT	查表取数码管段码	取得段码存入 38H	
13	DISDIGIT	将 38H 中的段码送入数码管显示		
14	SUBBCD	BCD 码减一	对 36H 中的 BCD 码操作	

2.2 问题研讨及程序分析

附录 A 电路图

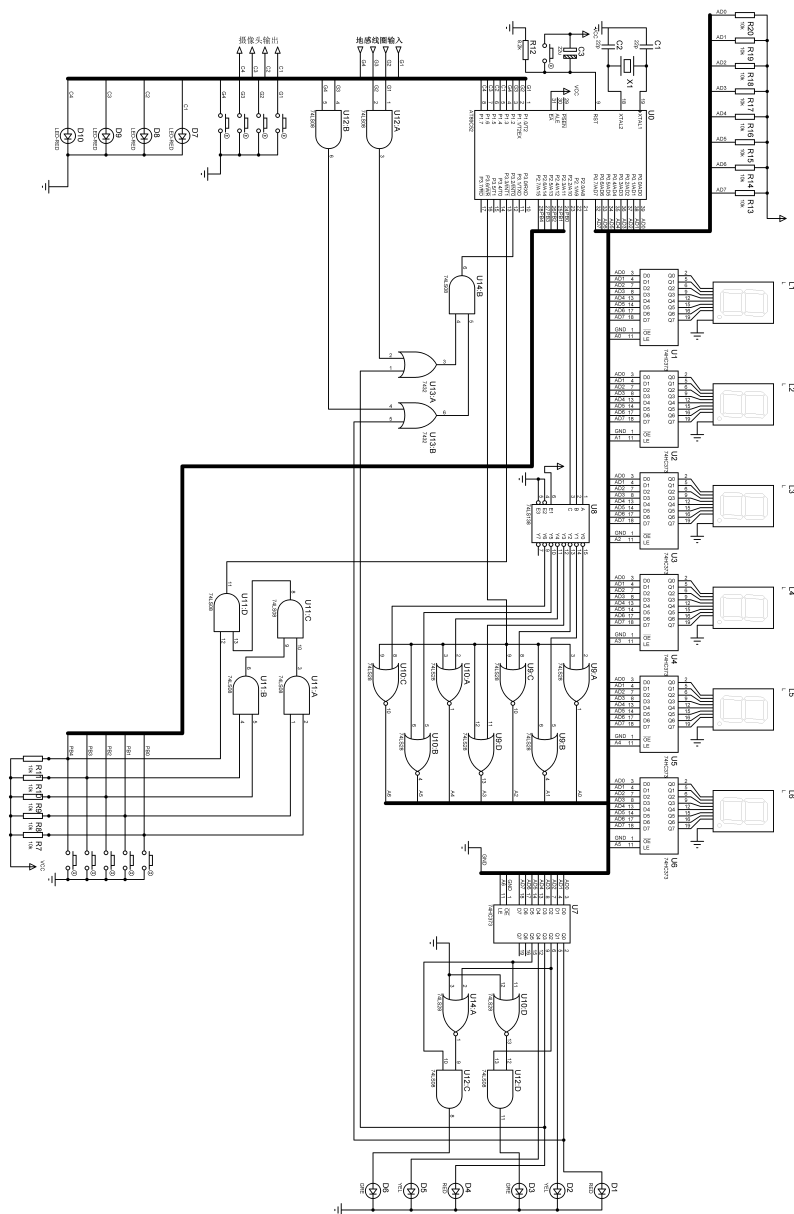


图 A-1 电路图

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^AT_EX 给文档编写工作带来的便利,也感谢 William Wang 同学对 L^AT_EX 模板移植做出的巨大贡献。