

系统实验报告书

班级： 13 通信

学号： 1342401005

姓名： 薛睿

目录

一、	设计任务.....	3
二、	器件功能及引脚图.....	4
三、	电路原理图.....	8
四、	软件设计流程.....	10
五、	已实现的功能.....	16
六、	调试过程.....	17
七、	心得小结.....	18
附录：程序源码		

一、设计任务

1、 时钟

- (1) 一只数码管 0-9 计数，一只按键控制开始、结束；
- (2) 二只数码管 0-59 计数，一只按键控制开始、结束；
- (3) 三只数码管低二位 0-59 计数后进位给高 0-9 计数，一只按键控制开始、结束；
- (4) 四只数码管低二位 0-59 计数后进位给高 0-59 计数，一只按键控制开始、结束；
- (5) 可增加按键增加功能（如开始、暂停）。

2、 交通灯

- (1) 仅亮灯，数码管不工作；
- (2) 灯和数码管相结合，模拟十字路口交通灯。

二、器件功能及引脚图

(1) 排阻，有点的一端为公共端。引脚图如图 2-1 所示。

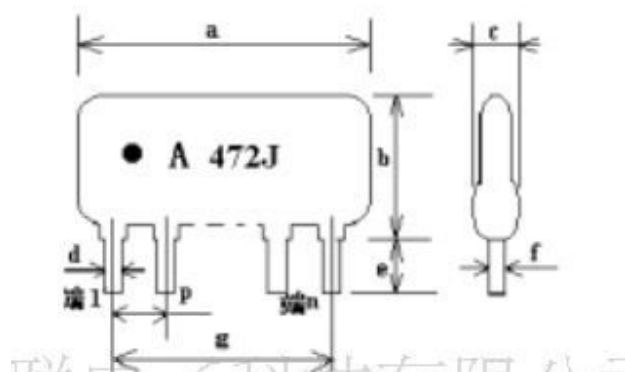


图 2-1 排阻引脚图

- (2) 电阻： $150\ \Omega$ 的驱动电阻，用于驱动数码管发光，根据数码管导通电流得出； $1\text{k}\ \Omega$ 的限流电阻，用于保护发光二极管，根据电源电压和二极管导通电流等参数得出。
- (3) C5、C6：接在晶振两边，是晶振的负载电容，参与振荡；
C7、C9：接在电源和地之间，起滤波的作用，消除干扰；
C12：接在复位按键两边，滤波作用，防止电压突变。其中有极电容引脚图如图 2-2 所示。

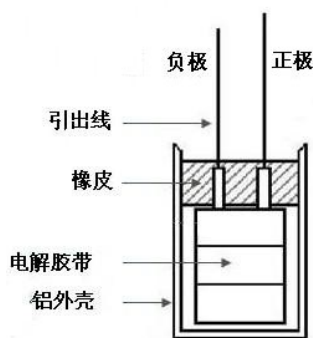


图 2-2 电容引脚图

- (4) 晶振： 11.0592MHz ，没有正负，提供单片机时钟。
- (5) 三极管：用于通过导通和截止来达到控制作用。
- (6) 74LS47：译码芯片，图 2-3 为 74LS47 引脚图。

①LT：试灯输入，是为了检查数码管各段是否能正常发光而设置的。当 $LT=0$ 时，无论输入 A_3 ， A_2 ， A_1 ， A_0 为何种状态，译码器输出均为低电平，若驱动的数码管正常，是显示 8。

②BI：灭灯输入，是为控制多位数码显示的灭灯所设置的。 $BI=0$ 时。

不论 LT 和输入 A3 , A2 , A1, A0 为何种状态, 译码器输出均为高电平, 使共阳极 7 段数码管熄灭。

③RBI: 灭零输入, 它是为使不希望显示的 0 熄灭而设定的。当对每一位 A3=A2=A1=A0=0 时, 本应显示 0, 但是在 RBI=0 作用下, 使译码器输出全 1。其结果和加入灭灯信号的结果一样, 将 0 熄灭。

④RBO: 灭零输出, 它和灭灯输入 BI 共用一端, 两者配合使用, 可以实现多位数码显示的灭零控制。表 2-1 为 74LS47 的真值表。

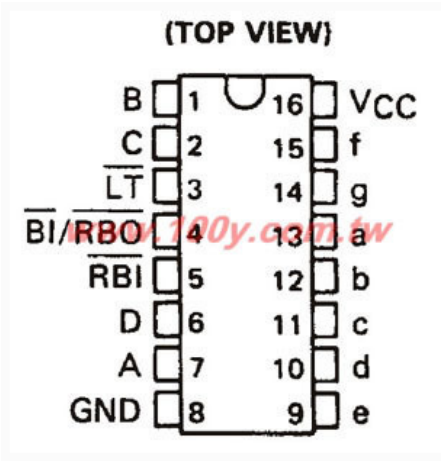


图 2-3 74LS47 引脚图

Truth Table																
Decimal or Function	Inputs							Outputs							Note	
	LT	RBI	A3	A2	A1	A0	BI/RBO	a	b	c	d	e	f	g		
0	H	H	L	L	L	L	H	L	L	L	L	L	L	H	(Note 2)	
1	H	X	L	L	L	H	H	H	L	L	H	H	H	H		(Note 2)
2	H	X	L	L	H	L	H	L	L	H	L	L	H	L		
3	H	X	L	L	H	H	H	L	L	L	L	H	H	L		
4	H	X	L	H	L	L	H	H	L	L	H	H	L	L	(Note 2)	
5	H	X	L	H	L	H	H	L	H	L	L	H	L	L		
6	H	X	L	H	H	L	H	H	H	L	L	L	L	L		
7	H	X	L	H	H	H	H	L	L	L	H	H	H	H		
8	H	X	H	L	L	L	H	L	L	L	L	L	L	L		
9	H	X	H	L	L	H	H	L	L	L	H	H	L	L		
10	H	X	H	L	H	L	H	H	H	H	L	L	H	L		
11	H	X	H	L	H	H	H	H	H	L	L	H	H	L		
12	H	X	H	H	L	L	H	H	L	H	H	H	L	L	(Note 3)	
13	H	X	H	H	L	H	H	L	H	H	L	H	L	L		
14	H	X	H	H	H	L	H	H	H	H	L	L	L	L		
15	H	X	H	H	H	H	H	H	H	H	H	H	H	H		
BI	X	X	X	X	X	X	L	H	H	H	H	H	H	H	(Note 4)	
RBI	H	L	L	L	L	L	L	H	H	H	H	H	H	H	(Note 5)	
LT	L	X	X	X	X	X	H	L	L	L	L	L	L	L		

表 2-1 74LS47 真值表

(6) AT89S51: 作为主控芯片使用, 图 2-4 是其引脚图。

P1.0	1	40	Vcc	
P1.1	2	39	P0.0/AD0	
P1.2	3	38	P0.1/AD1	
P1.3	4	37	P0.2/AD2	
P1.4	5	36	P0.3/AD3	
MOSI/P1.5	6	35	P0.4/AD4	
MISO/P1.6	7	34	P0.5/AD5	
SCK/P1.7	8	33	P0.6/AD6	
RST	9	32	P0.7/AD7	
RXD/P3.0	10	31	EA/VPP	
TXD/P3.1	11	30	ALE/PROG	
INT0/P3.2	12	29	PSEN	
INT1/P3.3	13	28	P2.7/A15	
T0/P3.4	14	27	P2.6/A14	
T1/P3.5	15	26	P2.5/A13	
WR/P3.6	16	25	P2.4/A12	
RD/P3.7	17	24	P2.3/A11	
XTAL2	18	23	P2.2/A10	
XTAL1	19	22	P2.1/A9	
PDIP	GND	20	21	P2.0/A8

图 2-4 AT89S51 引脚图

- ① P0 口：可做一般的 I/O，应在外部接上拉电阻。也可做数据总线和地址总线低 8 位分时复用。
 - ② P1 口：一般 I/O。
 - ③ P2 口：一般 I/O。也可做地址总线高 8 位。
 - ④ P3 口：一般 I/O。也可做第二功能引脚。
 - ⑤ VCC、VSS：电源正极和地。
 - ⑥ XTAL1、XTAL2：晶振的两极。
 - ⑦ 控制线引脚：RST/VPD 为复位信号输入或备用电源输入；ALE/PROG 为地址锁存允许信号输出或编程脉冲输入；EA/VPP 为内外存储器选择引脚或编程电压输入；PSEN 为外部程序存储器选通信号输出。
- (7) LM7805：三端集成稳压管，用于 9V-5V 电压转换。图 2-5 为引脚图。

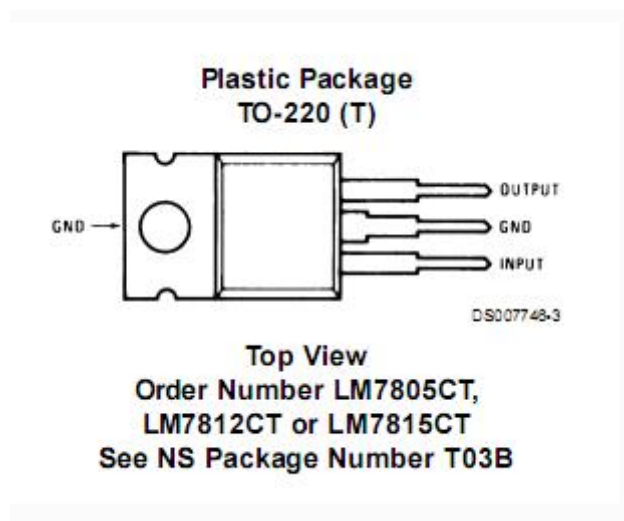


图 2-5 LM7805 引脚图

- (8) 按键：输入控制通断可用万用表测量。
- (9) LED 灯：较长的引脚为正极。
- (10) 数码管：共阳极数码管，当 com 端为高，相应段 I/O 输出低时点亮。

图 2-6 为数码管引脚图。

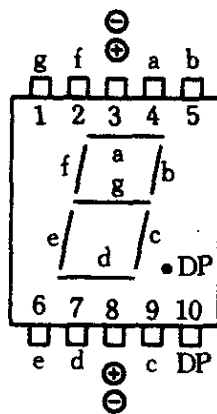


图 2-6 数码管引脚图

- (11) DC 插座：用于接插变压器。图 2-7 为引脚图。

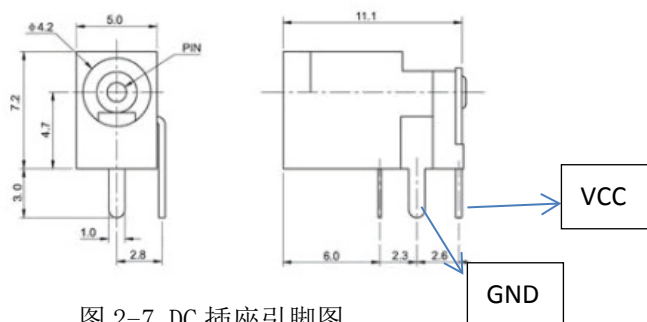


图 2-7 DC 插座引脚图

三、 电路原理图

分为按键模块、复位模块、晶振模块、数码管驱动模块、数码管选通模块、LED 驱动模块等等。图 3-1 为整体的电路原理图。图 3-2~3-4 为几个重要的模块原理图。

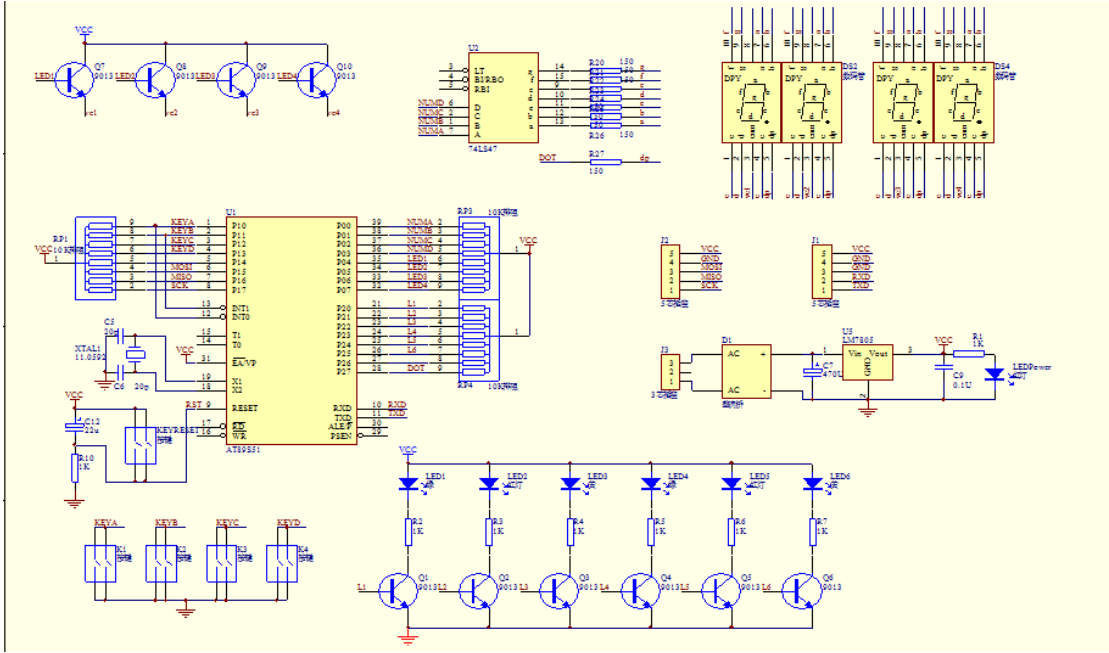


图 3-1 电路整体原理图

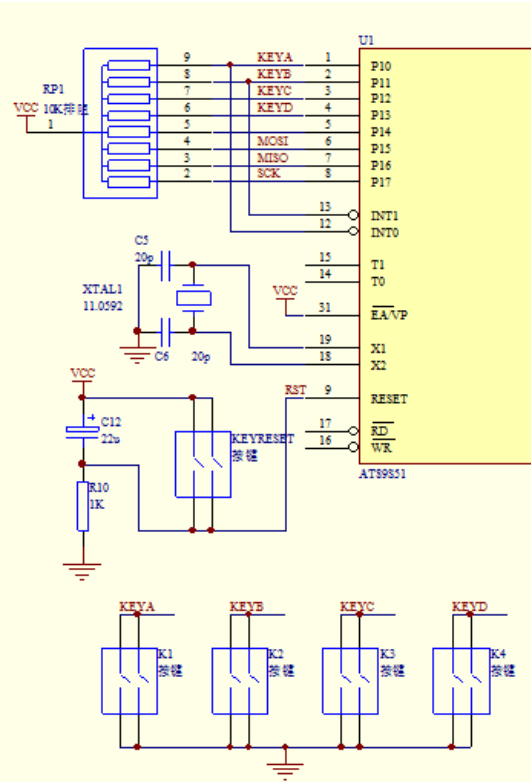


图 3-2 按键相关原理图

四、 软件设计流程

1、 时钟

时钟的设计分为以下几个模块：分秒计数、数码管动态显示、显示数字传送。其中分秒计数和动态显示选择用中断实现。图 4-1 是时钟程序的流程图。图 4-2~4-4 是三个中断服务函数。T0 中断用于 1ms 扫描显示数码管，T1 中断用于分秒计数，INT0 中断用于开始和暂停时钟，k3 按键可以控制时钟开始和停止。

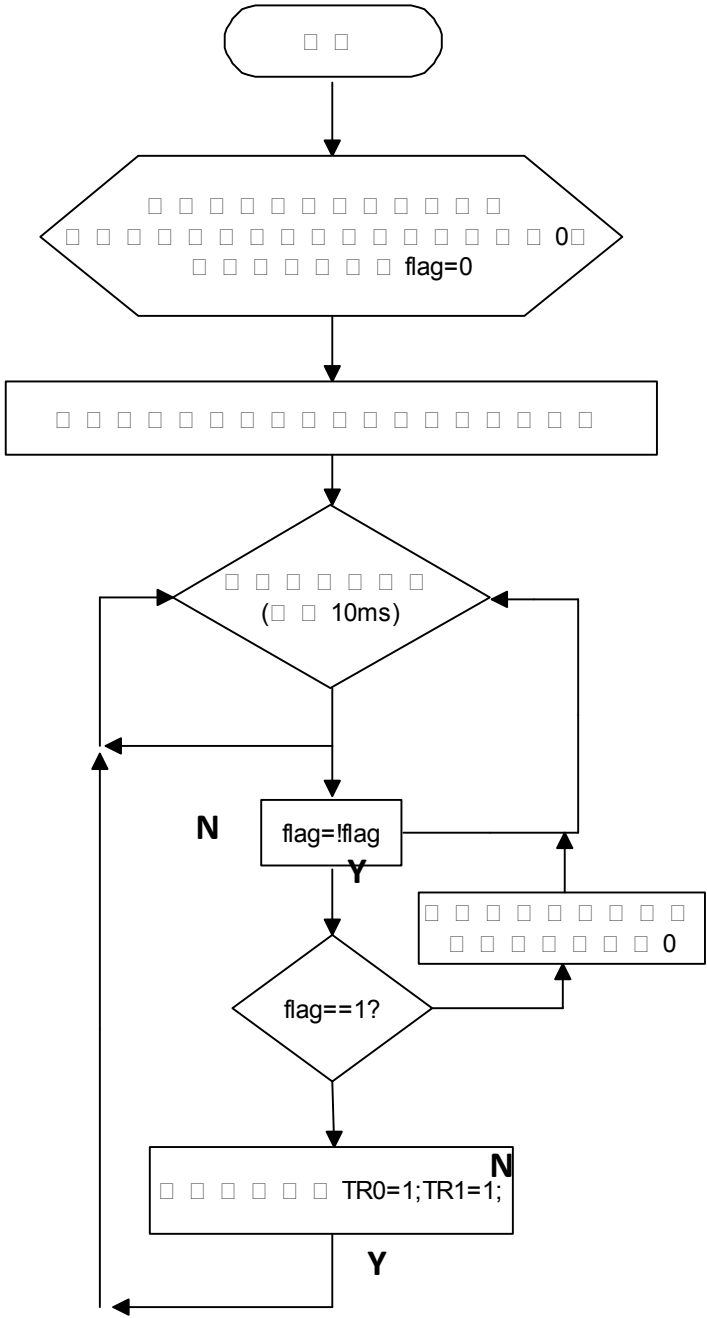


图 4-1 时钟程序流程图

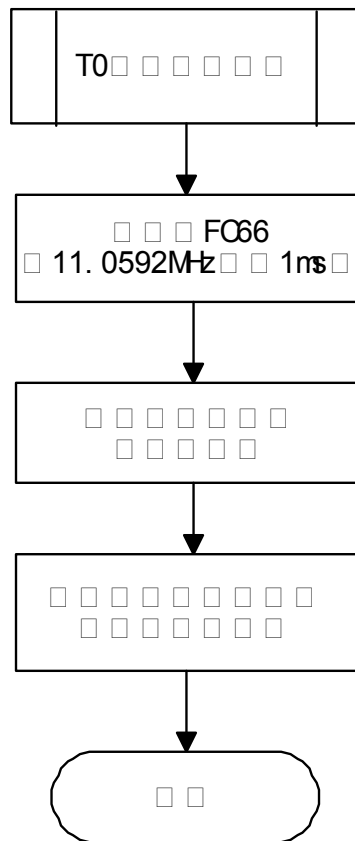


图 4-2 T0 中断服务函数

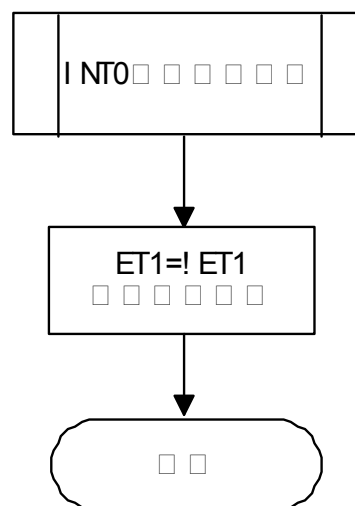


图 4-3 INT0 中断服务函数

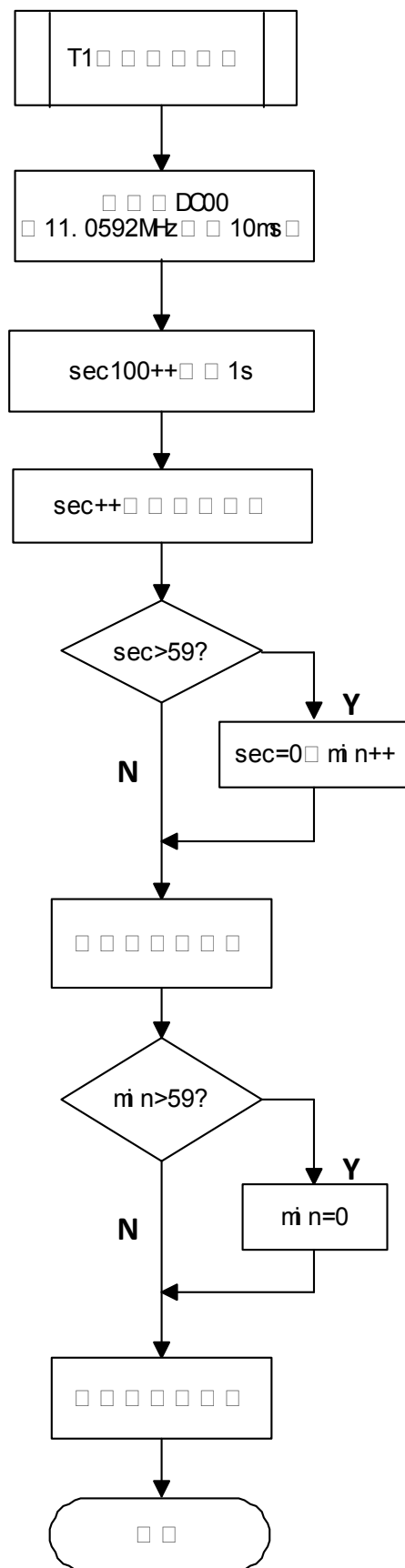


图 4-4 T1 中断服务函数

2、交通灯

交通灯的程序设计包括以下几个模块：交通灯控制程序主体、分秒计数、数码管动态显示、显示数字传送。其中计数和显示都在中断中完成，由 step 的值来在中断服务函数里区分完成相应操作。表 4-1 是实际交通灯的控制顺序。（以 20s 绿灯、5s 黄闪灯、30s 红灯为例）总函数流程图如图 4-5 所示。

	G1	Y1	R1	G2	Y2	R2
step1	20s~0s					25s~5s
step2		5s~0s 闪				5s~0s
Step3			25s~5s	20s~0s		
Step4			5s~0s		5s~0s 闪	

表 4-1 交通灯控制流程

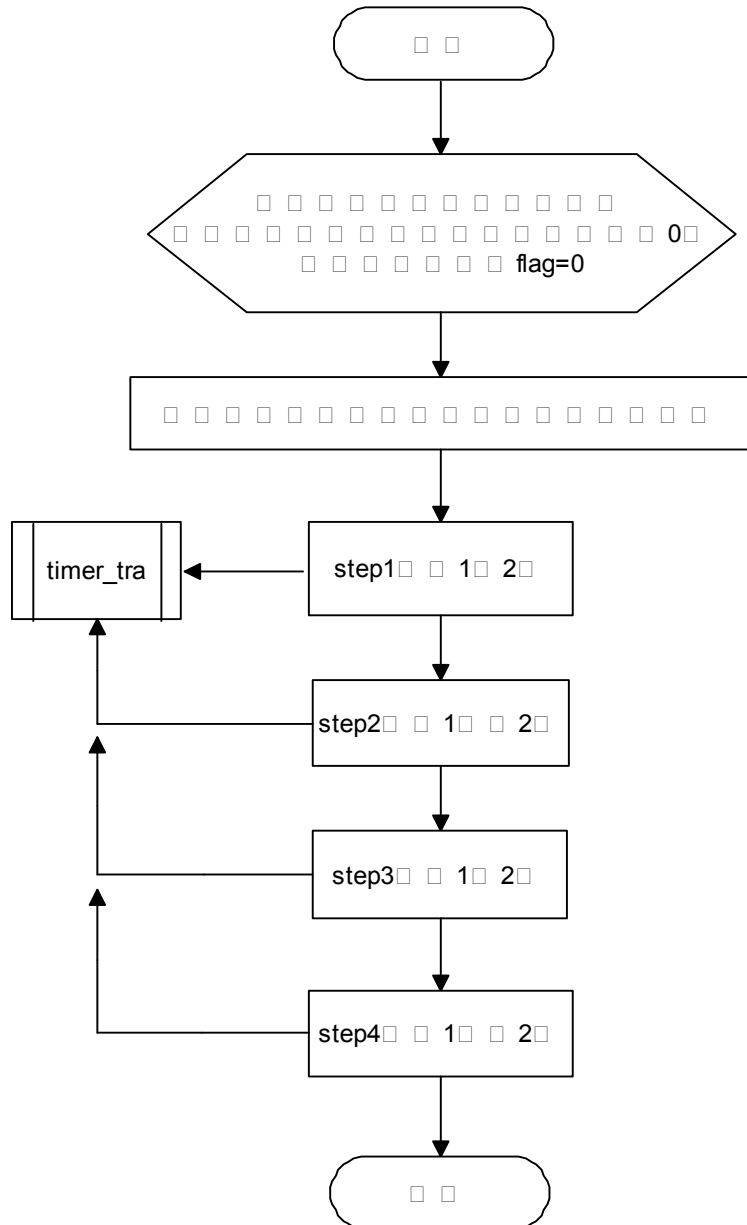


图 4-5 交通灯程序流程

这里设置一个新的函数 timer_tra，用于为不同的步设置初值，时间到时及时跳出，进行下一步的操作。子函数的流程图如 4-6 所示。

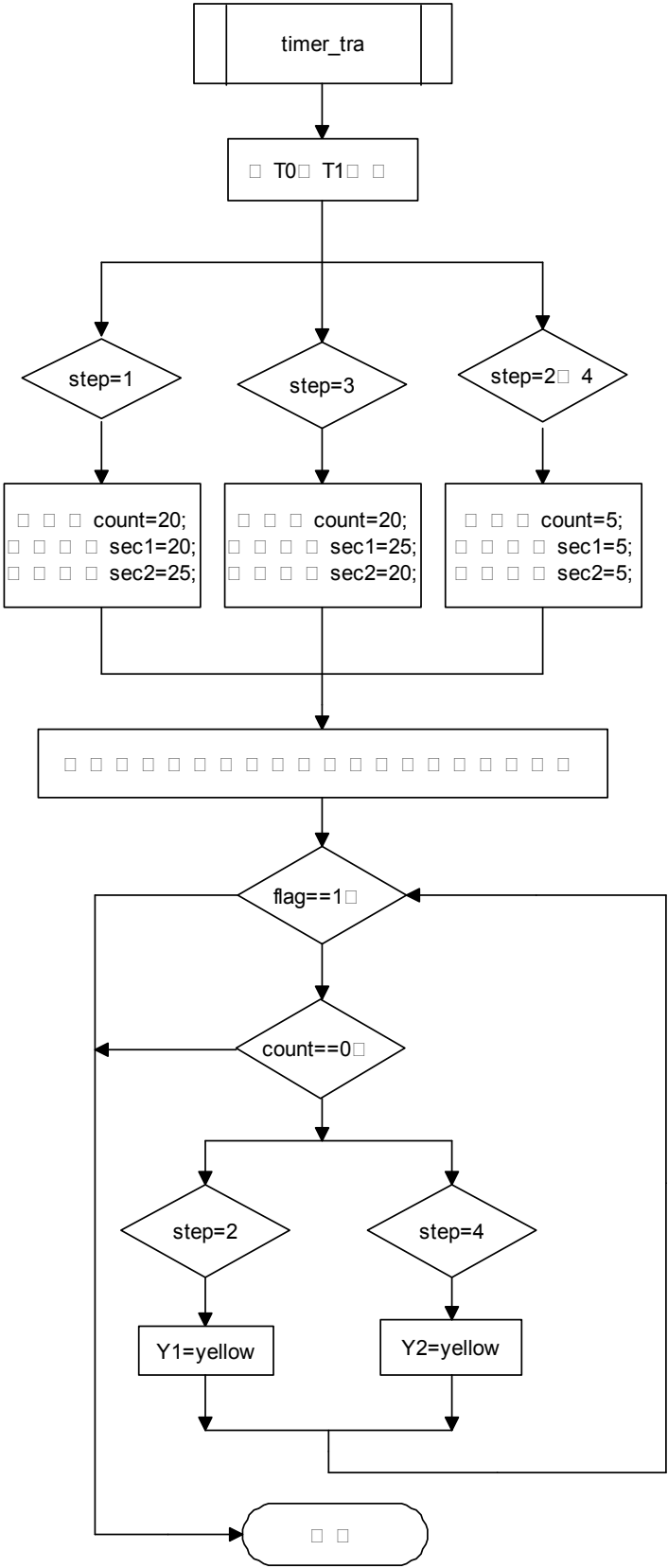


图 4-6 timer_tra 子函数流程

对应交通灯流程，应当设置 T1 中断中做倒计时工作，其他中断 T0、INT0 和时钟一样，不再赘述。T1 中断服务函数流程如图 4-7 所示。

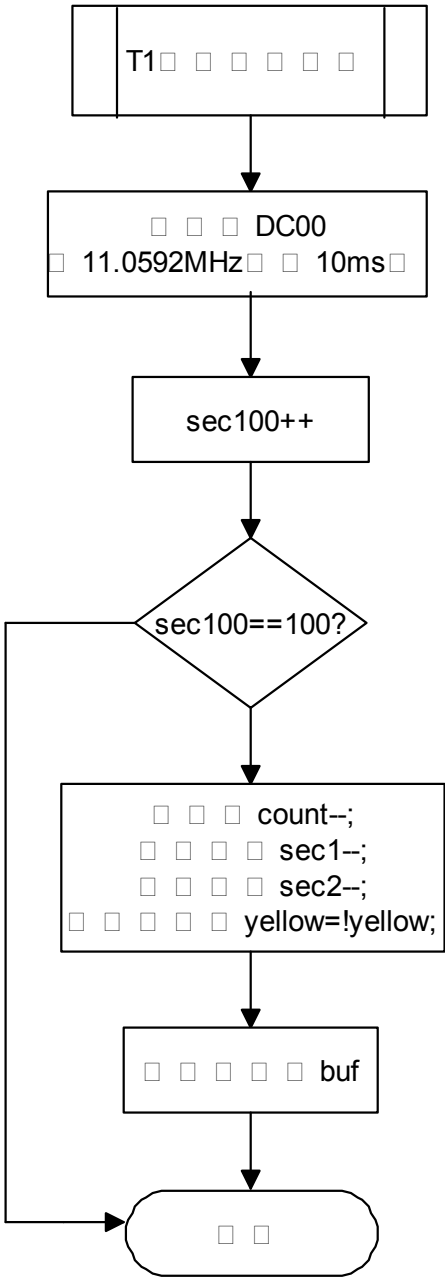


图 4-7 T1 中断服务函数

五、 已实现的功能

- 1、 时钟：四只数码管低二位 0-59 计数后进位给高 0-59 计数，满 59 小时清零从头开始。
- 2、 交通灯：模拟十字路口交通灯，红灯 25s，黄灯 5s，绿灯 20s，交替变换，数码管同步倒计时。
- 3、 按键：KEY1（中断）控制暂停和开始，KEY2（中断）控制时钟和交通灯功能无间隔切换，KEY3（扫描）控制开始和停止。

六、调试过程

1、对于时钟部分，我开始在数码管动态显示程序上出现了问题，显示的数字不会跳变，于是我在中断中加了取反 LED 的程序作为指示，发现中断可以实现，因此把问题确定在显示部分。经过观察，我没有在 1s 的时机到来时将索引值加一，这样虽然一直在扫描，但是送去显示的数是不变的，经过修改以后，可以实现功能。

2、对于交通灯部分，我分了几个阶段来实现：首先我尝试只亮灯，不显示数码管，利用 switch case 语句，加上之前使用的 t1 控制的 1s 中断，可以实现功能。然后我又尝试将时钟融入进去和交通灯一起跳变，由于都是用中断实现的，同步性较好。最后我修改了 t1 的中断函数，实现了倒计时的功能。

在此过程中，我发现一个小问题，倒计时的数缺少 1，直接从 2 跳到零。经过分析程序，我每次在显示前将显示初值置为 0，然后计数值 count 一再去做显示，这样的话，对于本该有 20 次计数的程序，其实 count 是从 19 开始计数的，所以我在主程序中先给数码管送了显示初值 20，这样就不会出现漏数的现象了，最终显示正常。

3、对于按键扩展功能部分，首先要考虑的问题就是采用扫描方法还是中断方法。显然，在实现上是中断方法更加方便和稳定，扫描方法虽然也可行，但是处理不当很容易造成无法实时切换的现象，因为正在执行的程序是不会被打断的。

原本我利用两个外部中断，将 K1 和 K2 按键分别作为开始暂停键和开始结束键，而切换键使用扫描 K3 的方法，通过复位后扫描来切换功能。但是老师提出了不允许用复位切换的方法，一开始我陷入了苦恼，因为我既想保留暂停、结束功能，又想做切换，这样就需要三个中断，貌似是不可能实现的。但是经过一番思考，我想到一个不算太好的解决方法，可以将扫描 K3 的程序插在 T1 的中断程序中，即相当于每 10ms 自动扫描一次按键，这样虽然可能无法做到像外部中断一样实时和准确，但是经过实验，若按键按的时候稍微延长一些，是可以实现功能的。于是我将 K3 作为开始结束键、K2 作为功能切换键、K1 作为开始暂停键，最终达到了老师的设计要求。

七、心得小结

这次系统实验是我们学习了单片机原理、数电、模电、C 语言等等课程后首次尝试独立编写一个比较完整的程序系统。我之前有一些调试程序的经验，但是没有独立写过较长的程序，所以我这次决定不参考别人的思路，完全依靠自己的思维去完成整个编程过程，这对我来说是一个挑战。

准备阶段，我们先要根据 PCB 板手绘出原理图。我的思路是首先从最复杂和重要的元件——51 芯片开始，先根据各个引脚本身的功能推测其用途，然后用万用表蜂鸣档去验证，首先绘出草图，然后合理分配每一个模块所在图纸的部位，工整抄写后，再按照模块一一检验是否有错绘和漏绘。焊接过程相对简单，只要注意元件的引脚顺序。焊完以后，我又用万用表按照模块测量了一下整个电路防止虚焊。

准备工作完成以后就进入了编程实现功能的部分。首先是时钟的设计，因为当时自学 51 单片机时就写过一个使用数码管的例程，虽然用的静态显示，并且硬件外设有所区别，但是原理大同小异，加上之前绘制原理图时已经理解了电路工作原理，我很顺利地完成了时钟的设计。

然后就是交通灯了，刚开始我做了一个两个方向的灯同时变化的“非标准”交通灯，并且采用了正计时方法，这个过程与时钟相仿，完成过程也比较顺利，于是我很高兴。可是事实证明，人不能太得瑟，更多时候我们还是要脚踏实地，在调整交通灯为实际情况的过程中我遇到了困难。第一是由于从同步变化为非同步的过程中我们需要理清实际的流程，即每一个循环应该分为几个阶段，因为平时缺少观察，我甚至在红灯到绿灯的过程中还加了一个黄闪灯，不过好在同学及时指出得以修改；第二就是因为流程增加的原因，中断函数的复杂度提高了，需要增加判断的过程，要费一点脑子。

最后是按键扩展功能的设计，也是费了一番功夫。大家的设计都五花八门，只有自己能看懂自己写的程序，想交流和讨论的时候才发现思路和程序结构完全不一样，我也渐渐看到自己程序中的一些冗余的部分，包括结构的一些不合理，出于时间关系可能没有机会再在课上完善了，不过我在课下仍然会继续思考的，相信能够收获更多。

在这次实训中，我编程的整体流程有了一个更深的了解认识，也巩固了前几个学期所学习的相关课程，最初的理论知识转化基本技能。看懂一

个程序不难，但是自己编写却不容易，一些思考逻辑上的不严密、对原理认识的不深刻、配置上的疏漏都会导致各种各样错误的现象。这个时候必须要从现象看本质，除了掌握常用的调试方法，例如添加指示灯、添加死循环等，最重要的是要从现象出发，分析可能的原因。例如按键失灵就要考虑是否在执行过程中不小心关掉了中断，如果是扫描方法那么可能是程序无法执行到扫描的部分，再例如数码管显示不正常，那么首先排除硬件的障碍，利用电压表去测试每一段是否都能亮起，然后在考虑是选通出了问题还是送显示数出了问题。所有的问题只有找到了原因才能够解决。

这次实验我还深切体会到了老师认真负责的精神和循循善诱的促学方式，不直接告知答案，而是引导我们去自己探索，并提出更高的要求，这种教学让我们独立思考和动手能力得到了真正的提高。

未来我也会继续认真学习基础知识，抓住机会动手实践，争取更大的进步，啊，真是一个美好的经历。