

系统实验报告

1342401019

王嘉玺

一、 本设计拟实现的功能

时钟（计时器）

1. 状态 1：一只数码管 0-9 计数，一只按键控制开始、结束；
2. 状态 2：二只数码管 0-59 计数，一只按键控制开始、结束；
3. 状态 3：三只数码管低二位 0-59 计数后进位给高位 0-9 计数，一只按键控制开始、结束；
4. 状态 4：四只数码管第二位 0-59 计数后进位给高二位 0-59 计数（即分秒计时），一只按键控制开始、结束；

交通灯

5. 状态 1：仅亮灯、数码管不工作；

按下键，红/黄/绿三色灯交替亮：红(20 秒)黄→闪烁 5 秒
绿←20 秒)黄（闪烁5 秒）红。 →

6. 状态 2：灯和数码管相结合，模拟十字路口的交通灯；

在以上功能的基础上数码管倒计时显示时间。

二、 所用器件的功能、引脚图

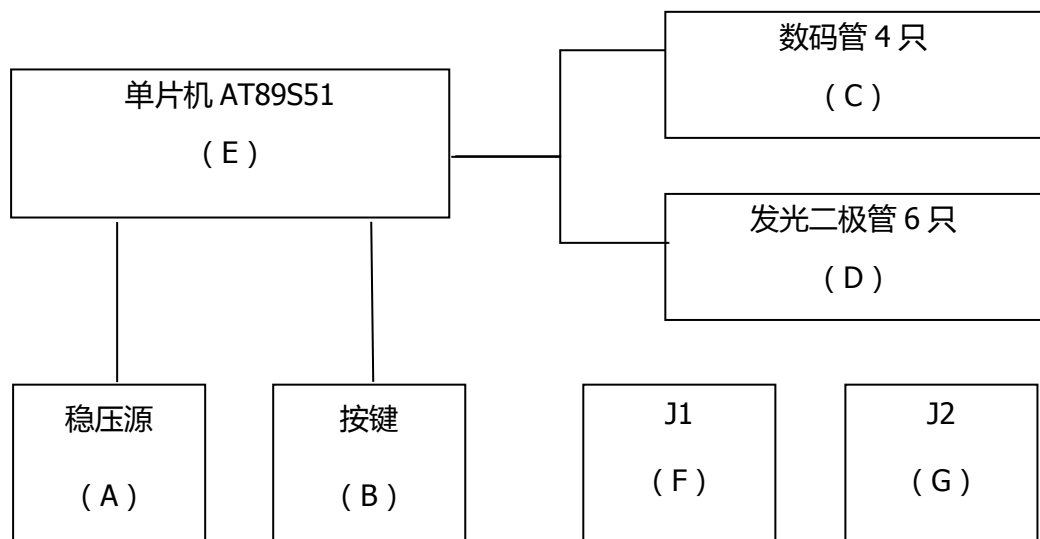


图 1 电路功能模块说明

(A) 稳压源部分将输入的交流电转换为稳定的 5V 直流电压；由于目前提供的已经是直流变压器，所以整流桥可以不用（若不接整流桥 D1，需将电路板上相应的引脚连通，使得接入的 9V 电源能够作用到 7805 上，此处需注意正负极）；

(B) 作为单片机的输入部分（P1.0–P1.3）起控制作用；

(C) 作为单片机的输出部分，P0.0–P0.3 经由 7 段译码器（74LS47）控制数码管的显示结果（0–9），P0.4–P0.7 数码管的选通控制口，提供数码管工作电源（公共端），P2.7 控制数码管的小数点是否点亮；

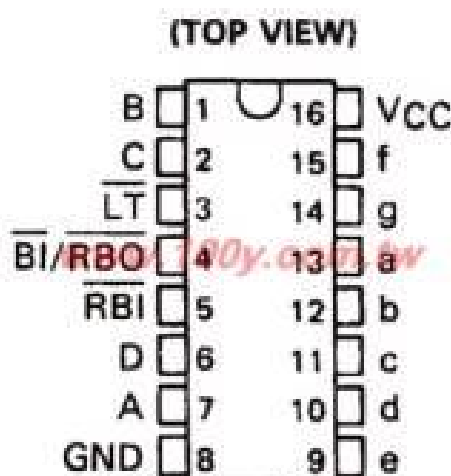


图 2 74LS47 引脚图

LT(——)：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 A3，A2，A1，A0 为何种状态，译码器输出均为低电平，也就是七段将全亮，若驱动的数码管正常，是显示 8。

BI(—)：灭灯输入，是为控制多位数码显示的灭灯所设置的。当 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，使共阳极数码管熄灭。

RBI(——): 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 $A_3=A_2=A_1=A_0=0$ 时, 本应显示 0, 但是在 RBI(——)=0 作用下, 使译码器输出全为高电平。其结果和加入灭灯信号的结果一样, 将 0 熄灭。RBO(——): 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。

74LS47 是 BCD-7 段数码管译码器驱动器, 74LS47 的功能用于将 BCD 码转化成数码块中的数字, 通过它来进行解码, 可以直接把数字转换为数码管的数字。

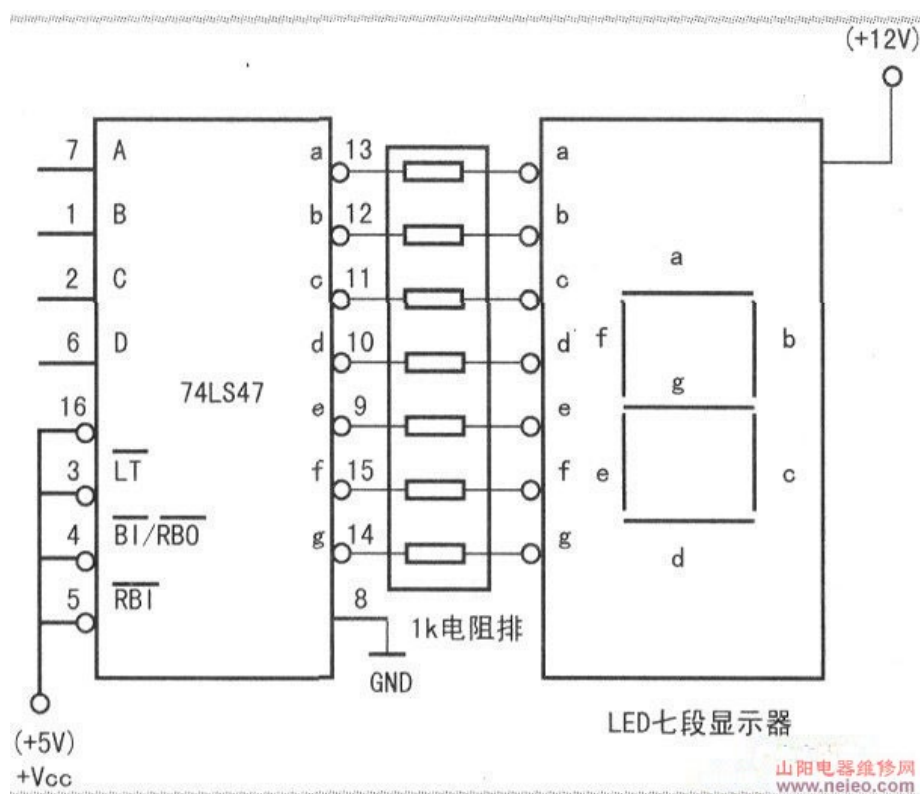


图 3 74LS47 与数码管的关系

译码器的逻辑功能是将每个输入的二进制代码译成对应的输出的高、低电平信号。常用的译码器电路有二进制译码器、二—十进制译码器和显示译码器。译码为编码的逆过程。它将编码时赋予代码的含义“翻译”过来。实现译码的逻辑电路成为译码器。译码器输出与输入代码有唯一的对应关系。74LS47 是输出低电平有效的七段字形译码器, 它在这里与数码管配合使用。将 BCD 码转化成数码

块中的数字, 通过它解码, 可以直接把数字转换为数码管的显示数字。

数码管共阳编码（共阴编码取反）：

显示——HGFE, DCBA——编码

0 ——1100, 0000——0xC0

1 ——1111, 1001——0xE9

2 ——1010, 0100——0xA4

3 ——1011, 0000——0xB0

4 ——1001, 1001——0x99

5 ——1001, 0010——0x92

6 ——1000, 0010——0x82

7 ——1111, 1000——0xE8

8 ——1000, 0000——0x80

9 ——1001, 0000——0x90

(D) 发光二极管：输出部分，由 P2.0-P2.5 口控制，高电平灯亮；

(E) 单片机 AT89S51：低功耗，高性能 CMOS 8 位单片机。片内含 8KB 的 ISP（IN-SYSTEM-PROGRAMMABLE）的可反复擦写 1000 次的 FLASH 只读程序存储器，器件采用 ATMEL 公司的高密度、非易失性存储技术制造，兼容标准 MCS-51 指令系统及 80C51 引脚结构。其特点可概括为：40 个引脚；8KB FLASH 片内存储器（可反复擦写大于 1000 次）；256BYTES 的随机存储器（RAM）；32 个外部双向输入/输出(I/O)口；5 个中断优先级，2 层中断嵌套中断；2 个全双工串行通信口；看门狗（WDT）电路；片内时钟振荡器（0-33MHZ）；4.5-5.5V 工作电压。



图 4 AT89S51 引脚图

(F) J1 口留作通信口使用，其中 J1.1 (TXD) 由 P3.1 (11 脚) 控制，J1.2 (RXD) 由 P3.0 (10 脚) 控制；

(G) J2 留作在线烧录程序使用。

三、 本系统设计电路原理图

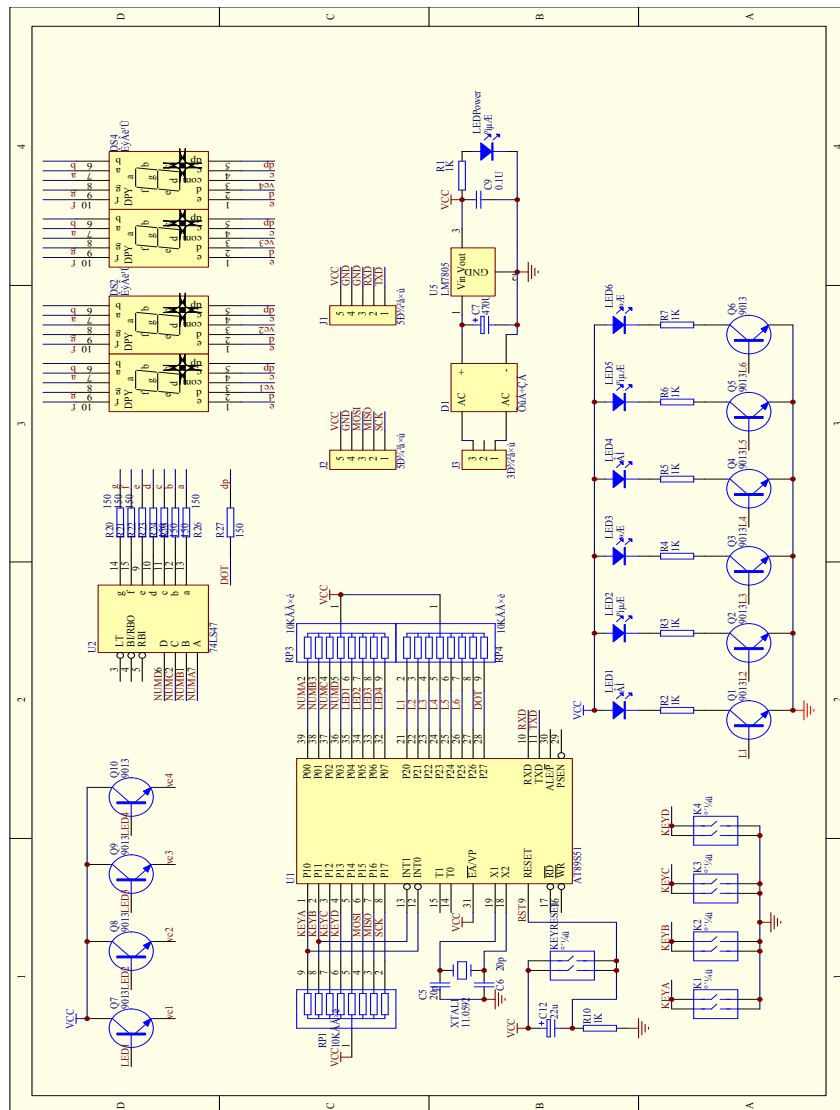
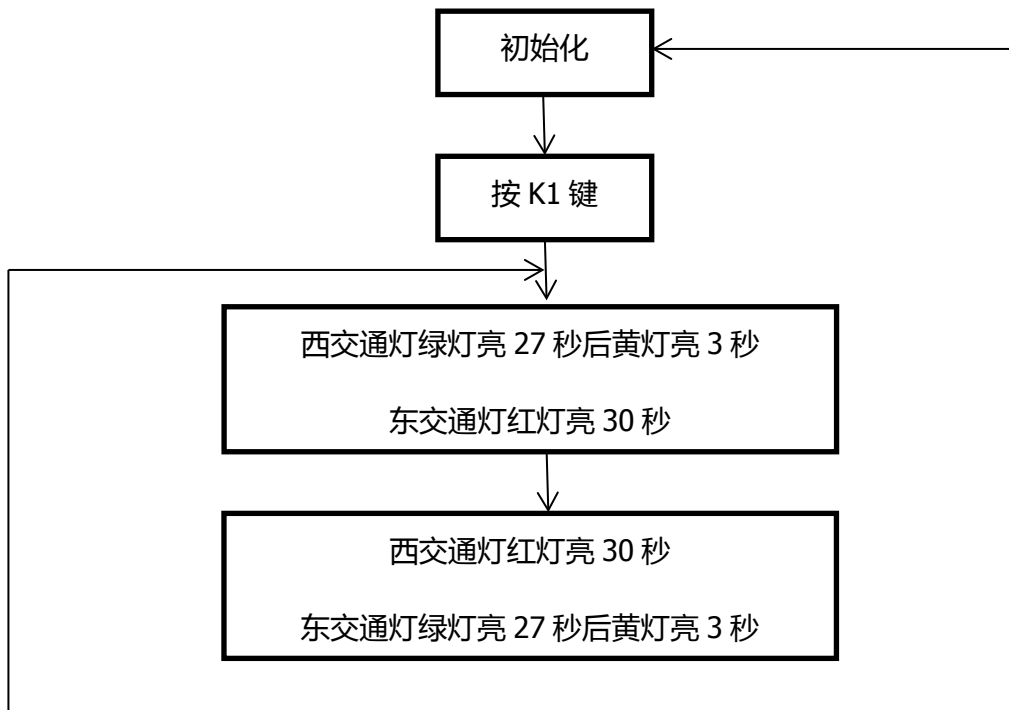


图 5 电路原理图

四、 软件设计流程图



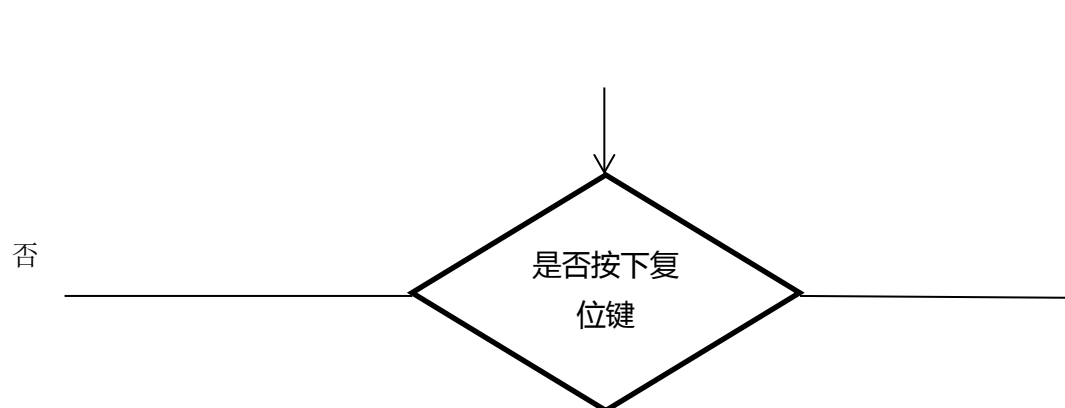


图 6 状态 6 设计流程图

五、已实现的功能

时钟（计时器）

1. 状态 1：一只数码管 0-9 计数，一只按键控制开始、结束；
2. 状态 2：二只数码管 0-59 计数，一只按键控制开始、结束；
3. 状态 3：三只数码管低二位 0-59 计数后进位给高位 0-9 计数，一只按键控制开始、结束；
4. 状态 4：四只数码管第二位 0-59 计数后进位给高二位 0-59 计数（即分秒计时），一只按键控制开始、结束；

交通灯

5. 状态 1：仅亮灯、数码管不工作；

按下键，红/黄/绿三色灯交替亮：红(20 秒)黄(闪烁 5 秒)绿(20 秒)黄(闪烁 5 秒)红。 →

6. 状态 2：灯和数码管相结合，模拟十字路口的交通灯；

在以上功能的基础上数码管倒计时显示时间。

六、 调试过程

通过对已编程序的烧录、测试，发现了不少问题，比如当数码管在需要运作时不亮，经检查发现是排阻 RP3、RP4 未完成焊接，焊接完成后仍然存在相似问题，检查得焊接时有部分地方出现短路。在解决完硬件问题后，功能上仍然有一些问题亟待解决，检查软件问题发现是：参数设置错误、循环嵌套时流程出错等，依次解决后实现了要求的功能。

七、 心得小结

通过本次系统实验，我对电路板焊接工艺更加熟练，了解了单片机 AT89S51 的结构和功能，掌握了译码器 74LS47 和数码管各自的功能与运用，对单片机 keil4 的开发环境有了更深的理解，并且学会运用该软件编写程序以实现既定要求。

在实验的过程中，遇到了不少问题，最后耐心地逐一检查、解决才得以实现交通灯的功能，这也让我意识到编写代码的过程并不是一蹴而就的，而是需要不断地调试，才能达到预期效果。