

# 系统实验

---

基于 AT89S51 单片机的交通灯控制系统

班 级：13 通信

学号： 1342401041

姓 名：唐懂润

指导教师：陈蓉

|                       |   |
|-----------------------|---|
| 一、概述.....             | 1 |
| 二、设计任务.....           | 1 |
| *时钟（计时器） .....        | 1 |
| *交通灯： .....           | 1 |
| 三、主要器件介绍.....         | 1 |
| 1. AT89S51 芯片介绍.....  | 1 |
| 2、AT89S51 引脚说明： ..... | 2 |
| 3. 集成块 74LS47.....    | 4 |
| 4. 集成块 LM7805.....    | 5 |
| 四、电路原理图.....          | 5 |
| 五、软件流程图.....          | 6 |
| 六、调试过程.....           | 7 |
| 七、心得体会.....           | 7 |

## 一、概述

本次系统实验是基于单片机的可控交通灯系统。

## 二、设计任务

\*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 4：四只数码管低二位 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。

## 三、主要器件介绍

### 1. AT89S51 芯片介绍

AT89S51 单片机包含中央处理器、程序存储器（ROM）、数据存储器（RAM）、定时/计数器、并行接口、串行接口和中断系统等几大单元及数据总线、地址总线和控制总线等三大总线，下面分别加以说明：

#### (1). 中央处理器

中央处理器（CPU）是整个单片机的核心部件，是 8 位数据宽度的处理器，能处理 8 位二进制数据或代码，CPU 负责控制、指挥和调度整个单元系统协调的工作，完成运算和控制输入输出功能等操作。

#### (2). 数据存储器（RAM）

AT89S51 内部有 128 个 8 位用户数据存储单元和 128 个专用寄存器单元，它们是统一编址的，专用寄存器只能用于存放控制指令数据，用户只能访问，而不能用于存放用户数据，所以，用户能使用的 RAM 只有 128 个，可存放读写的数据，运算的中间结果或用户定义的字型表。

#### (3). 程序存储器（ROM）

AT89S51 共有 4096 个 8 位掩膜 ROM，用于存放用户程序，原始数据或表格。

#### (4). 定时/计数器 (ROM)

AT89S51 有两个 16 位的可编程定时/计数器，以实现定时或计数产生中断用于控制程序转向。

#### (5). 并行输入输出口 (I/O)

AT89S51 共有 4 组 8 位 I/O 口 (P0、P1、P2 或 P3)，用于对外部数据的传输。

#### (6). 全双工串行口

AT89S51 内置一个全双工串行通信口，用于与其它设备间的串行数据传送，串行口既可以用于异步通信接发器，也可以当同步移位器使用。

#### (7). 中断系统

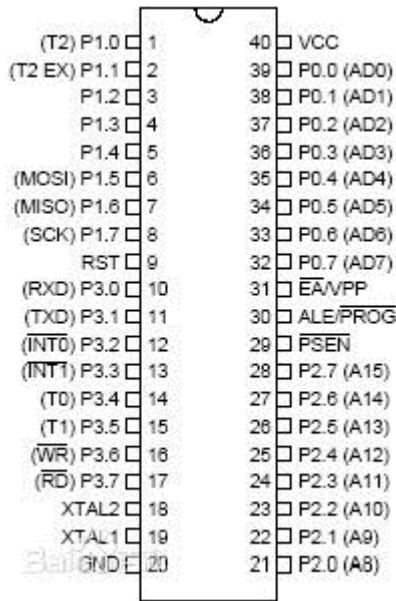
AT89S51 具备较完整的中断功能，有两个外中断、两个定时/计数器中断和一个串行中断，可满足不同的控制要求，并具有 2 级的优先级别选择。

#### (8). 时钟电路

AT89S51 内置最高频率达 12MHz 的时钟电路，用于产生整个单片机运行的脉冲时序，但 AT89S51 单片机需外置震荡电容。

单片机的结构有两种类型，一种是程序和数据存储器分开的形式，即哈佛 (Harvard) 结构，另一种是采用通用计算机广泛使用的程序存储器与数据存储器合二为一的结构，即普林斯顿 (Princeton) 结构。AT89S51 单片机采用的是哈佛结构的形式。

### 2、AT89S51 引脚说明：



(图 1)AT89S51 引脚图

P0 口：

P0 口是一个 8 位漏极开路的双向 I/O 口。作为输出口，每位能驱动 8 个 TTL 逻辑电平。对 P0 端口写“1”时，引脚用作高阻抗输入。当访问外部程序和数据存储器时，P0 口也被作为低 8 位地址/数据复用。在这种模式下，P0 不具有内部上拉电阻。在 flash 编程时，P0 口也用来接收指令字节；在程序校验时，输出指令字节。程序校验时，需要外部上拉电阻。

P1 口：

P1 口是一个具有内部上拉电阻的 8 位双向 I/O 口，p1 输出缓冲器能驱动 4 个 TTL 逻辑电平。

此外，P1.0 和 P1.1 分别作定时器/计数器 2 的外部计数输入 (P1.0/T2) 和定时器/计数器 2 的触发输入 (P1.1/T2EX)。在 flash 编程和校验时，P1 口接收低 8 位地址字节。

引脚号第二功能：

P1.0 T2 (定时器/计数器 T2 的外部计数输入)，时钟输出

P1.1 T2EX (定时器/计数器 T2 的捕捉/重载触发信号和方向控制)

P1.5 MOSI (在系统编程用)

P1.6 MISO (在系统编程用)

P1.7 SCK (在系统编程用)

P2 口：

P2 口是一个具有内部上拉电阻的 8 位双向 I/O 口，P2 输出缓冲器能驱动

4 个 TTL 逻辑电平。对 P2 端口写“1”时，内部上拉电阻把端口拉高，此时可以作为输入口使用。作为输入使用时，被外部拉低的引脚由于内部电阻的原因，将输出电流（IIL）。在访问外部程序存储器或用 16 位地址读取外部数据存储器（例如执行 MOVX @DPTR）时，P2 口送出高八位地址。在这种应用中，P2 口使用很强的内部上拉发送 1。在使用 8 位地址（如 MOVX @RI）访问外部数据存储器时，P2 口输出 P2 锁存器的内容。在 flash 编程和校验时，P2 口也接收高 8 位地址字节和一些控制信号。

P3 口：

P3 口是一个具有内部上拉电阻的 8 位双向 I/O 口，p3 输出缓冲器能驱动 4 个 TTL 逻辑电平。P3 口亦作为 AT89S52 特殊功能（第二功能）使用，如下表所示。

| 端口引脚 | 第二功能              |
|------|-------------------|
| P3.0 | RXD（串行输入口）        |
| P3.1 | TXD（串行输出口）        |
| P3.2 | /INT0（外中断 0）      |
| P3.3 | /INT1（外中断 1）      |
| P3.4 | T0（定时/计数器 0 外部输入） |
| P3.5 | T1（定时/计数器 1 外部输入） |
| P3.6 | /WR（外部数据存储器写选通）   |
| P3.7 | /WD（外部数据存储器读选通）   |

（表 1）

此外，P3 口还接收一些用于 FLASH 闪存编程和程序校验的控制信号。

RST：

复位输入。当振荡器工作时，RST 引脚出现两个机器周期以上高电平将是单片机复位。

ALE/PROG：

当访问外部程序存储器或数据存储器时，ALE（地址锁存允许）输出脉冲用于锁存地址的低 8 位字节。一般情况下，ALE 仍以时钟振荡频率的 1/6 输出固定的脉冲信号，因此它可对外输出时钟或用于定时目的。要注意的是：每当访问外部数据存储器时将跳过一个 ALE 脉冲。对 FLASH 存储器编程期间，该引脚还用于输入编程脉冲（PROG）。如有必要，可通过对特殊功能寄存器（SFR）区中的 8EH 单元的 D0 位置位，可禁止 ALE 操作。该位置位后，只有一条 MOVX 和 MOVC 指令才能将 ALE 激活。此外，该引脚会被微弱拉高，单片机执行外部程序时，应设置 ALE 禁止位无效。

PSEN:

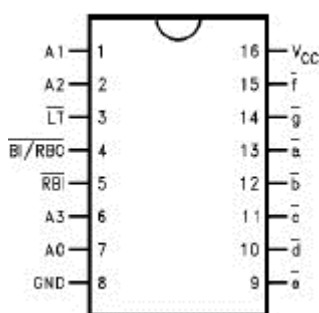
程序储存允许（PSEN）输出是外部程序存储器的读选通信号，当 AT89S52 由外部程序存储器取指令（或数据）时，每个机器周期两次 PSEN 有效，即输出两个脉冲，在此期间，当访问外部数据存储器，将跳过两次 PSEN 信号。

EA/VPP:

外部访问允许，欲使 CPU 仅访问外部程序存储器（地址为 0000H~FFFFH），EA 端必须保持低电平（接地）。需注意的是：如果加密位 LB1 被编程，复位时内部会锁存 EA 端状态。如 EA 端为高电平（接 Vcc 端），CPU 则执行内部程序存储器的指令。FLASH 存储器编程时，该引脚加上+12V 的编程允许电源 Vpp，当然这必须是该器件是使用 12V 编程电压 Vpp。

### 3. 集成块 74LS47

引脚图:



(图 2) 74LS47 引脚图

a、b、c、d、e、f、g 是七段译码器的输出驱动信号，低电平有效。可直接驱动共阳极七段数码管，是使能端，起辅助控制作用。

使能端的作用如下：

(1) 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，若驱动的数码管正常，是显示 8。

(2) BI(—)：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。

(3) 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 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。

(4) RBO(——)：灭零输出，它和灭灯输入 BI(—)共用一端，两者配合使

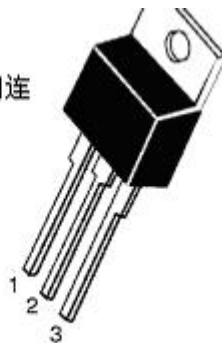
用，可以实现多位数码显示的灭零控制。

#### 4. 集成块 LM7805

LM317T

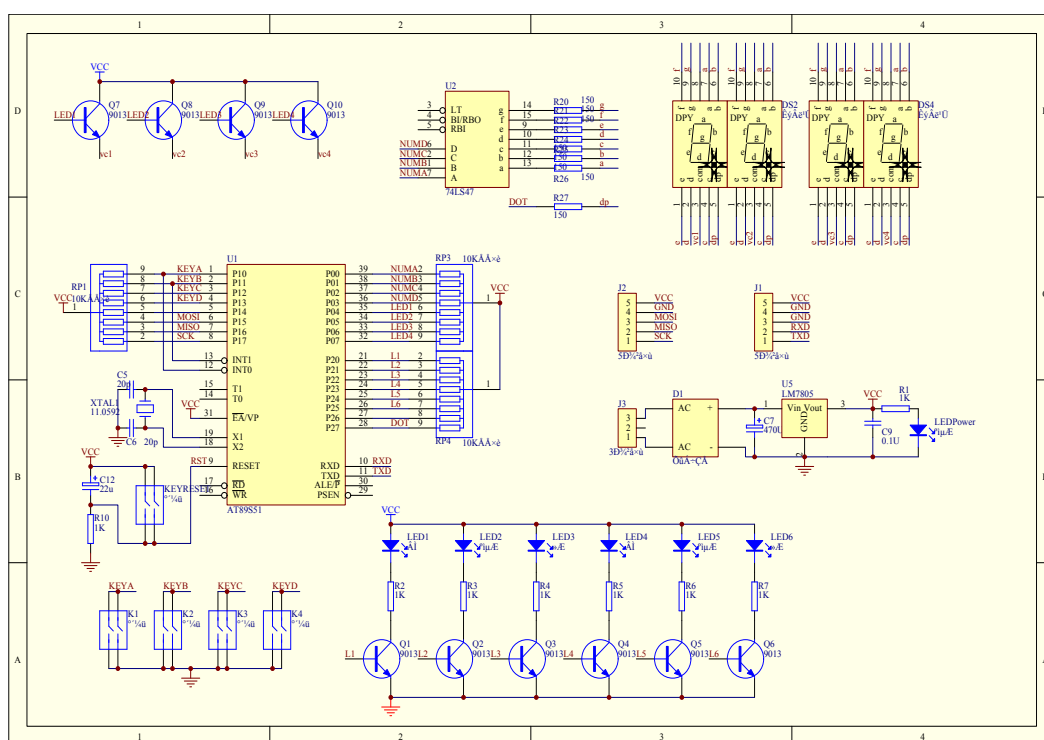
散热器表面与2脚相连

1脚: adj  
2脚: v-out  
3脚: v-in



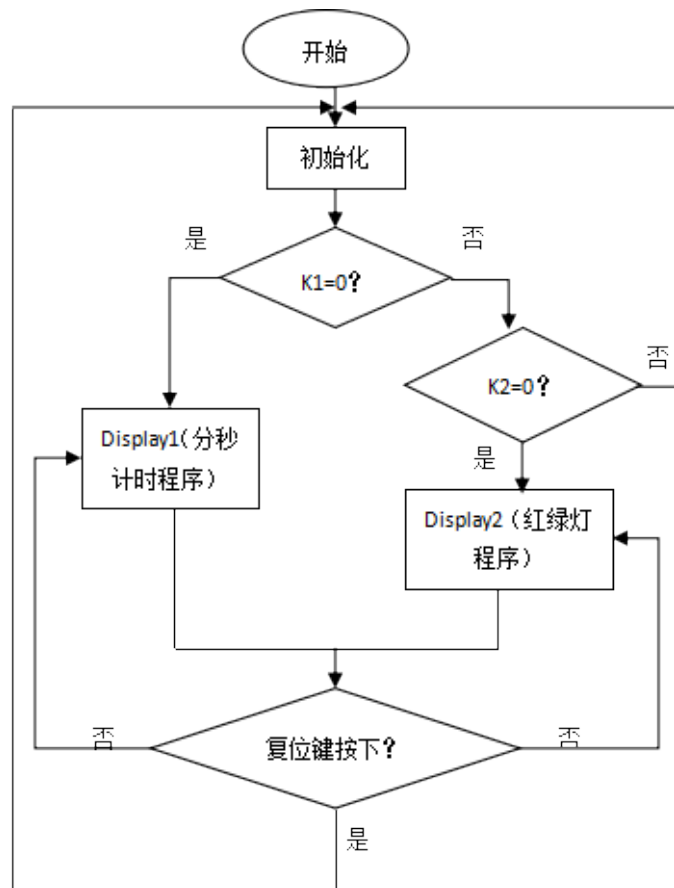
(图 3) LM7805

### 四、电路原理图



### 五、软件流程图

(1) 主程序部分:



(2) 子程序部分：



## 六、调试过程：

焊接时不小心把排阻和 LM7805 方向弄反了，焊好和同学对比才发现错误，后来在同学的帮助下用电烙铁一点一点取下来又重新焊接，后面的编程也感觉很困难，出现过灯的颜色变化不对，最后经过不断的修改才完成最终程序。

## 七、心得体会

两个星期的交通灯实验，虽然有些疲倦和困难，但带给了我很多的收获。在这次实验之前我已经做过类似电路焊接的实验，所以在开始画完原理图后进行焊接很是得心应手，但，我还是焊错了两个地方，一个是集成块 LM7805，另一个是排阻，焊的时候还以为它们没正负，后来和完成的同学比较才知道其实它们都有正负的，可惜我都焊的很完整了，没办法只能再取下来重新焊接，两个都是比较难取的，特别是排阻，在同学的帮助下，好不容易才弄下来重新焊上去，这里我要由衷感谢帮助我的同学，这真是一个非常需要耐心的操作，最后检查电路正确才松了一口气，后来发现有几个同学也焊错了，于是我就帮他们一起改正，由于有了经验，发现后来取的时候也不是那么难了，感觉自己的手越来越熟练了。这次实验最难的地方就是编程了，编的时候才发现以前课堂上学的东西都忘得差不多了，于是我又复习了一下基本的编程方法，勉强编出了基本的程序，最后经过请教同学我才完成了完整的程序，经过不断的测试烧录后，插上电路板验证才算完成了。本次实验让我的电路焊接技术又有了进一步的提高，我学会了有些错误看似很小，但纠正起来还是比较难的，所以只有一开始就用心做好每一步，不急不燥，才能把错误的概率降到最低，做的时候遇到不太清楚的也不能含糊的理所当然过去，要多请教老师同学，做的慢的话就多学习其他同学的经验教训，争取不犯同样的错误，多学会总结，这样会对自己有很大的帮助。总的来说，通过设计、焊接和制作一个电子电路系统让我了解一个电子产品的开发和制作过程，了解了平时马路上看到的交通灯是怎么运行的，锻炼了我的焊接技术、编程能力和改错技巧，培养了我理论联系实际的思维方式，也让我意识到了在动手能力和理论方面我还有不少欠缺，还需要不断的学习来填充自己，让以后的实验做得有效快速。