

系统实验报告

学院：城市轨道交通学院
专业：通信工程
学号：1309401002
姓名：朱佳卉

1、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：仅灯亮，数码管不工作；按下键，红、黄、绿三色灯交替亮：红（20s）→黄（闪烁 5s）→绿（20s）→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。

在本次实验中我选择的是任务 6 和任务 4。

2、实验器件

单片机 AT89S51、集成块 74LS47、LM7805、数码管 4 只、发光二极管 6 只、稳压源、按键 4 个、电阻、排阻、三极管、电容。

1. 单片机 AT89S51:

低功耗，高性能 CMOS 8 位单片机。片内含 8KB 的 ISP 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 引脚结构。

Pin Configurations

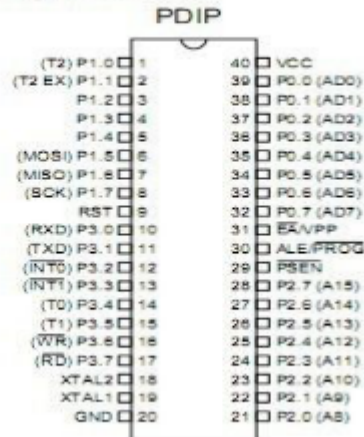


图 2.1 AT89S51 引脚图

引脚功能说明:

- **Vcc:** 电源电压
- **GND:** 地

• **P0 口:** P0 口是一组 8 位漏极开路型双向 I/O 口，也即地址/数据总线复用口。作为输出口用时，每位能驱动 8 个 TTL 逻辑门电路，对端口写 ‘1’ 可作为高阻抗输入端用。

在访问外部数据存储器或程序存储器时，这组口线分时转换地址(低 8 位)和数据总线复用，在访问期间激活内部上拉电阻。

在 Flash 编程时，P0 口接收指令字节，而在程序校验时，输出指令字节，校验时，要求外接上拉电阻。

• **P1 口:** P1 是一个带内部上拉电阻的 8 位双向 I/O 口，P1 的输出缓冲级可驱动(吸收或输出电流)4 个 TTL 逻辑门电路。对端口写 ‘1’，通过内部的上拉电阻把端口拉到高电平，此时可作输入口。作输入口使用时，因为内部存在上拉电阻，某个引脚被外部信号拉低时会输出一个电流(In)。

Flash 编程和程序校验期间 P 1 接收低 8 位地址。

• **P2 口:** P2 是一个带有内部上拉电阻的 8 位双向 I/O 口，P2 的输出缓冲级可驱动(吸收或输出电流)4 个 TTL 逻辑门电路。对端口写 ‘1’，通过内部的上拉电阻把端口拉到高电平，此时可作输入口，作输入口使用时，因为内部存在上拉电阻，某个引脚被外部信号拉低时会输出一个电流(In)。

在访问外部程序存储器或 16 位地址的外部数据存储器 (例如执行 MOVX @DPTR 指令) 时, P2 口送出高 8 位地址数据。在访问 8 位地址的外部数据存储器 (如执行 MOVX @Ri 指令) 时, P2 口线卜的内容 (也即特殊功能寄存器 (SFR) 区中 P2 寄存器的内容), 在整个访问期间不改变。

Flash 编程或校验时, P2 亦接收高位地址和其它控制信号。

• **P3 口:** P3 口是一组带有内部上拉电阻的 8 位双向 I/O 口。P3 口输出缓冲级可驱动 (吸收或输出电流) 4 个 TTL 逻辑门电路。对 P3 口写入 “1” 时, 它们被内部上拉电阻拉高并可作为输入端口。作输入端时, 被外部拉低的 P3 口将用上拉电阻输出电流 (In)。

P3 口除了作为一般的 I/O 口线外, 更重要的用途是它的第二功能。

P3 口还接收一些用于 Flash 闪速存储器编程和程序校验的控制信号。

P3 口也可作为 AT89C51 的一些特殊功能口, 如下表所示:

管脚	备选功能
P3.0	RXD (串行输入口)
P3.1	TXD (串行输出口)
P3.2	$\overline{\text{INT0}}$ (外部中断 0)
P3.3	$\overline{\text{INT1}}$ (外部中断 1)
P3.4	T0 (记时器 0 外部输入)
P3.5	T1 (记时器 1 外部输入)
P3.6	$\overline{\text{WR}}$ (外部数据存储器写选通)
P3.7	$\overline{\text{RD}}$ (外部数据存储器读选通)

• **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使单片机复位。WDT 溢出将使该引脚输出高电平, 设置 SFR AUXR 的 DISRTO 位 (地址 8EH) 可打开或关闭该功能。DISRTO 位缺省为 RESET 输出高电平打开状态。

• **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。即使不访问外部存储器, ALE 仍以时钟振荡频率的 1/6 输出固定的正脉冲信号, 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 ALE 才会被激活。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 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 AT89S51 由外部程序存储器取指令(数据)时，每个机器周期两次 PSEN 有效，即输出两个脉冲。当访问外部数据存储器，没有两次有效的 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中的指令。

Flash 存储器编程时，该引脚加上+12 V 的编程电压 Vpp。

- **XTAL1:** 振荡器反相放大器及内部时钟发生器的输入端。

- **XTAL2:** 振荡器反相放大器的输出端。

- **特殊功能寄存器:**

特殊功能寄存器的于片内的空间分布的这些地址并没有全部占用，没有占用的地址亦不可使用，读这些地址将得到一个随意的数值。而写这些地址单元将不能得到预期的结果。

- **程序存储器:**

如果 EA 引脚接地 (GND)，全部程序均执行外部存储器。

在 AT89S51, 假如 EA 接至 Vcc (电源+)，程序首先执行地址从 0000H-0FFFH (4KB) 内部程序存储器，而执行地址为 1000H-FFFFH (60KB) 的外部程序存储器。

2. 芯片 74LS47

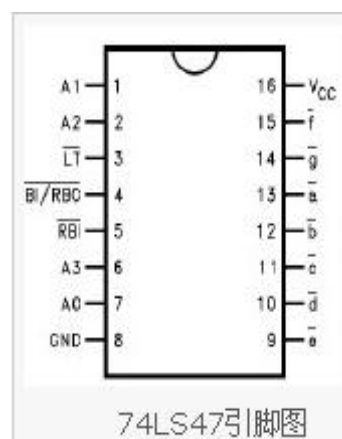


图 2.2 74LS47 引脚图

真值表：

$\overline{LT}$	$\overline{RBI}$	$\overline{BI} / \overline{RBO}$	D C B A	abcdefg	说明
0	X	1	X X X X	0000000	试灯
X	X	0	X X X X	1111111	熄灭
1	0	0	0 0 0 0	1111111	灭零
1	1	1	0 0 0 0	0000001	0
1	X	1	0 0 0 1	1001111	1
1	X	1	0 0 1 0	0010010	2
1	X	1	0 0 1 1	0000110	3
1	X	1	0 1 0 0	1001100	4

图 2.3 74LS47 真值表（1）

1	X	1	0 1 0 1	0100100	5
1	X	1	0 1 1 0	1100000	6
1	X	1	0 1 1 1	0001111	7
1	X	1	1 0 0 0	0000000	8
1	X	1	1 0 0 1	0001100	9

图 2.4 74LS47 真值表（2）

- 3. 稳压源:将输入的交流电转换为 5V 的稳定直流电压。
- 4. 发光二极管：输出部分，由 P2.0-P2.5 口控制，高电平灯亮。
- 5. 七段数码管：

将所有数码管通过分时轮流控制各个数码管的的 COM 端，就使各个数码管轮流受控显示。将所有数码管的 8 个显示笔划“a, b, c, d, e, f, g, dp”的同名端连在一起，另外为每个数码管的公共极 COM 增加位选通控制电路，位选通由各自独立的 I/O 线控制，当单片机输出字形码时，所有数码管都接收到相同的字形码，但究竟是那个数码管会显示出字形，取决于单片机对位选通 COM 端电路的控制，所以我们只要将需要显示的数码管的选通控制打开，该位就显示出字形，没有选通的数码管就不会亮。

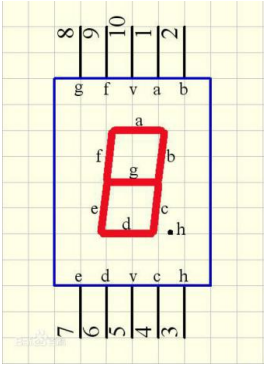


图 2.5 数码管引脚图

6. 集成块 LM7805:

LM7805 是常用的三端稳压器，一般使用的是 TO-220 封装，能提供 DC 5V 的输出电压，应用范围广，内含过流和过载保护电路。带散热片时能持续提供 1A 的电流，如果使用外围器件，它还能提供不通的电压和电流。

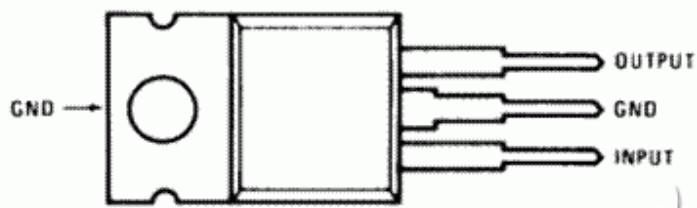


图 2.6 LM7805 引脚图

3、电路原理图

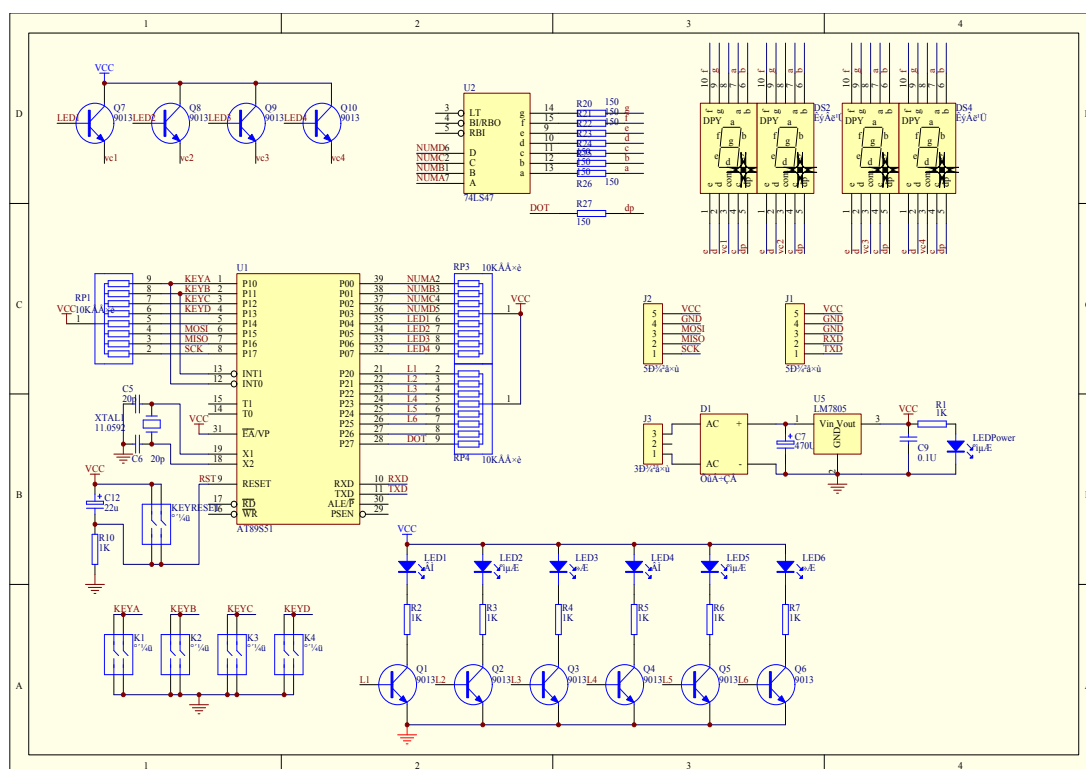
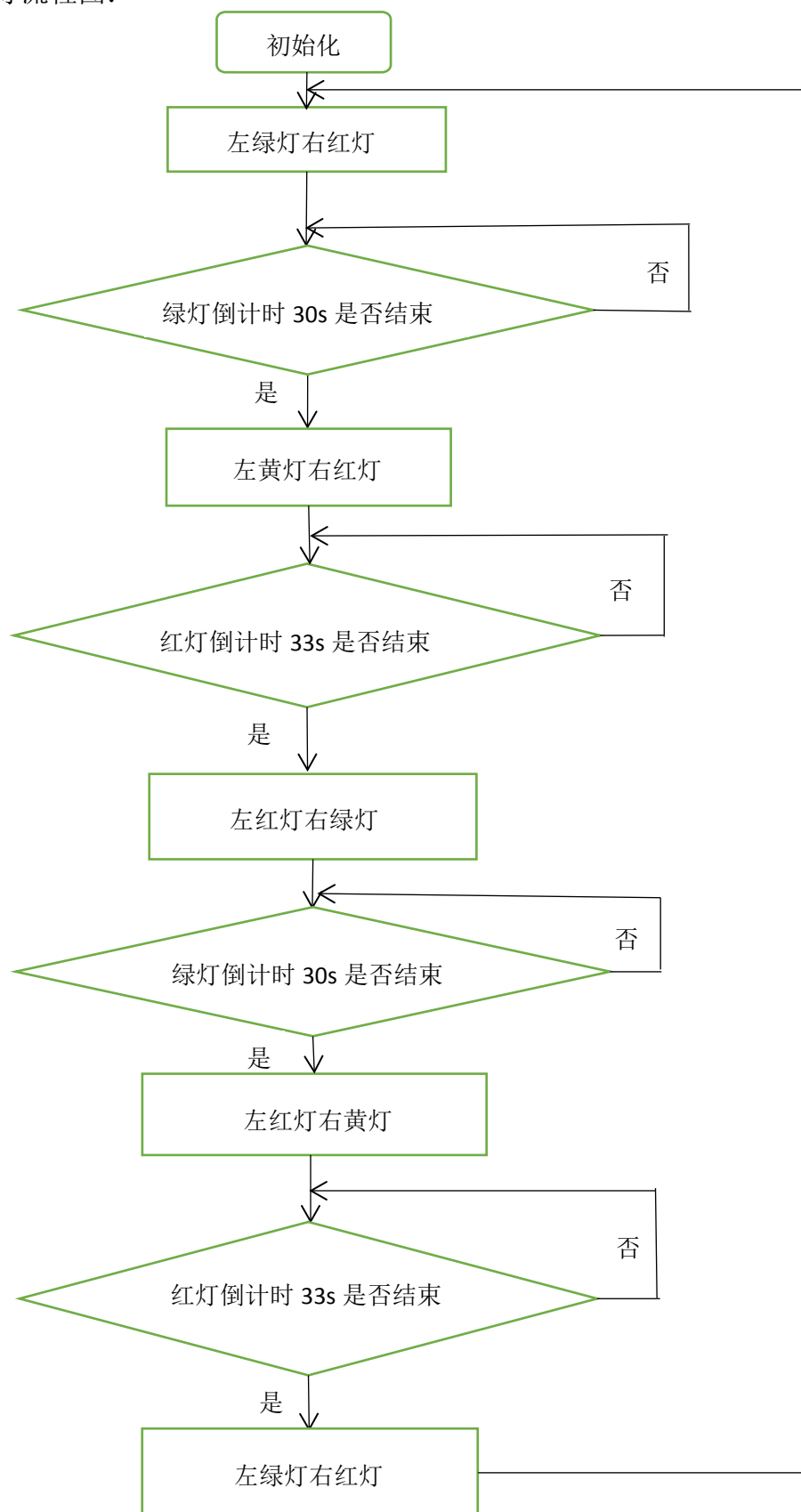


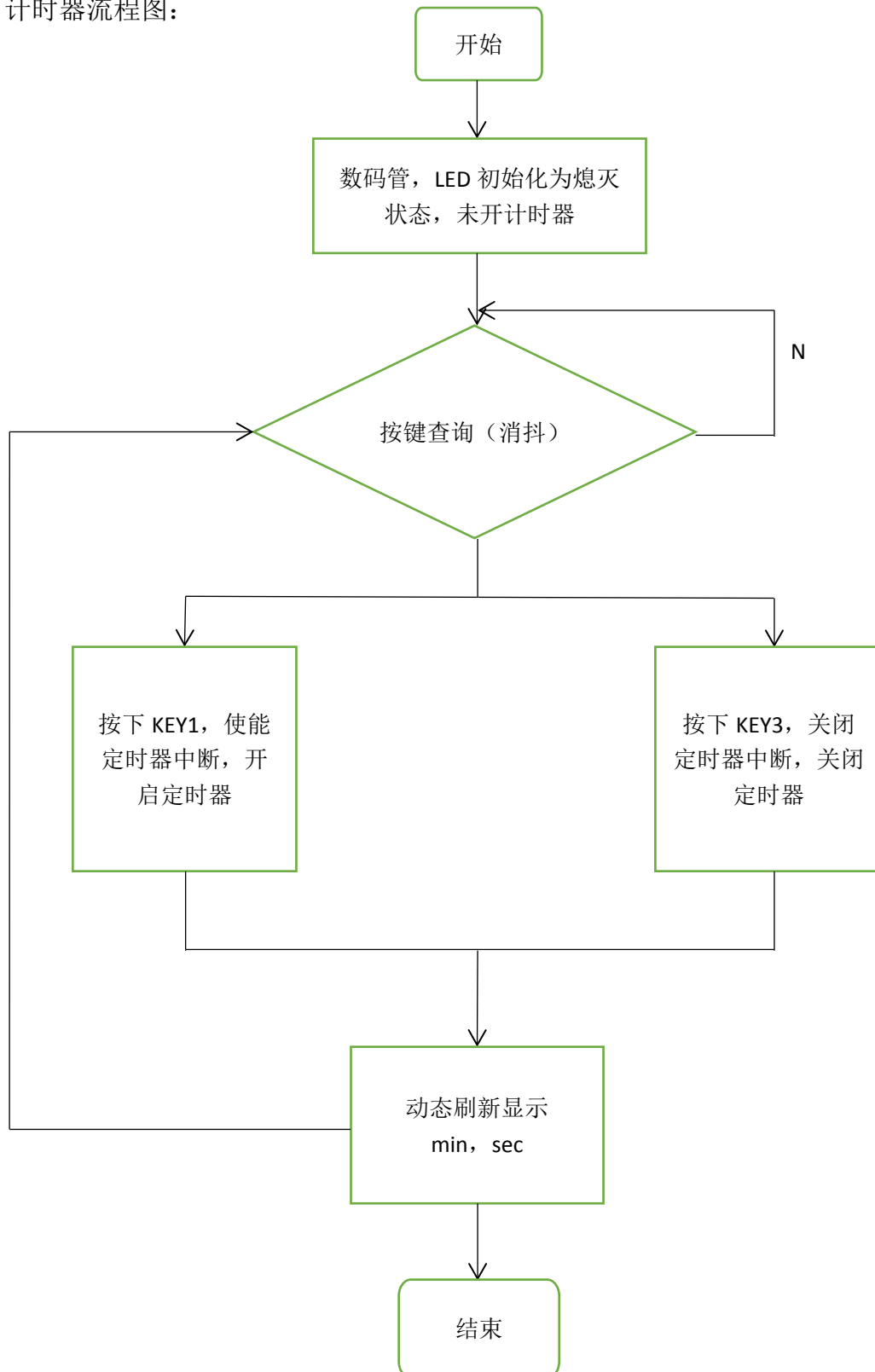
图 3.1 电路原理图

4、软件设计流程图

1. 交通灯流程图：



2. 计时器流程图：



五、已实现的功能

1、交通灯：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。

数码管 1 和数码管 2 是显示左边红绿灯的计时，数码管 3 和数码管 4 是显示右边红绿灯的计时，按键 0 是总开关，按键 1 是开始计时按钮。按下按键一便开始倒计时，数码管 1 和 2 显示 30，数码管 3 和 4 显示 33，左边亮绿灯右边亮红灯，30s 后左边亮黄灯右边亮红灯，33s 后左边亮红灯右边亮绿灯，数码管 1 和 2 显示 33，数码管 3 和 4 显示 30，30s 后左边亮红灯右边亮黄灯，33s 后左边亮绿灯右边亮红灯，并且数码管 1 和 2 显示 30，数码管 3 和 4 显示 33，如此循环。

2、计时器：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），KEY1 控制开始，KEY3 控制暂停。

6、调试过程

编写好程序后，编译生成 HEX 文件，选择器件（Device）AT89S51 文件（File）→装入文件（Load）→擦除（Erase）→空检查（Blank check）→编程（program）→校验（Verify）烧录程序。

在编写程序和调试程序时，遇到过几个问题：

（1）程序写的比较冗长，后来才将延时子程序作为调用程序放到后面调用，缩减了一些。

（2）而且在定义变量和循环嵌套时也会出错，仔细检查之后才修正。

（3）一开始设置按键时，发现按下按键后屏幕会闪烁，询问同学后加入按键去抖解决了这个问题。

（4）烧录程序进芯片后，如果芯片与板子接触不良数码管也很容易显示错误。

（5）编程时，在使用！取反的时候会弄错前后位置以及循环的判定。以及使用==还是=等细节上会出现问题。

7、心得小结

经过本次系统实验，我对 51 单片机各个引脚功能有了更加全面的了解，也

学会了使用 C 语言编写一些简单的程序，发现编程并没有想象中的那么困难，让我能够比之前更加熟练的运用 C 语言。通过硬件焊接、原理图绘制等过程，也加强了我对硬件的理解。