浙江大学 实验报告

课程名称：微机原理及应用

实验名称：实验一数码管循环显示实验

指导老师：田翔/马永昌

实验类型：仿真验证实验

专业：生物医学工程

姓名：

学号：

日期：2024 年 3 月 21 日

地点：东 1B-416

目录

1. 实验目的和要求

1.1. 实验要求

1.熟悉 MCS-51 教学实验系统硬件结构。

2.看懂 C 语言程序，编写汇编代码，用 proteus 仿真 。

3.双数码管实现秒表功能要求实验前，对实验任务进行认真分析，写出需求分析报告和系统设计报告。

1.2. 实验任务

•数码管变化由 00->99，不断循环，表示从 0.0 秒到 9.9 秒变化。99 过后回到 00，重新循环。

•Proteus 器件选择:AT89C52/AT89C51、7SEG-MPX2-CA/7SEG-MPX2-CC（共阳与共阴数码管均可)、其他外围器件。(注意电路设计的合理性)

2. 实验内容和原理

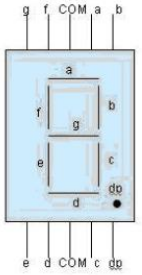
1.数码管显示

最常用的是七段式和八段式 LED 数码管，八段比七段多了一个小数点。

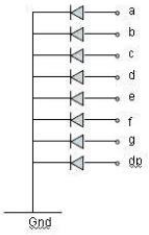
•所谓的八段就是指数码管里有八个小 LED 发光二极管，通过控制不同的 LED 的亮灭来显示出不同的字形。

•数码管又分为共阴极和共阳极两种类型：共阴极就是将八个 LED 的阴极连在一起，让其接地，这样给任何一个 LED 的另一端高电平，它便能点亮；而共阳极就是将八个 LED 的阳极连在一起。

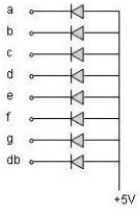
•数码管的 8 段，对应一个字节的 8 位，a 对应最低位，dp 对应最高位。



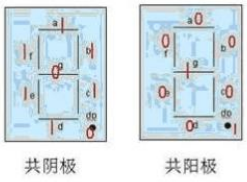
引脚图



共阴极



共阳极


共阴极　　共阳极

- 假设要让数码管显示数字0：
  - 共阴数码管的字符编码：00111111，即0x3F
  - 共阳数码管的字符编码：11000000，即0xC0

3. 主要仪器设备

计算机、Proteus 8.9 仿真软件

4. 操作方法和实验步骤

4.1. 需求分析

数码管变化由 00->99，不断循环，表示从 0.0 秒到 9.9 秒变化。99 过后回到 00，重新循环。

4.2. 系统设计

· 实验装置以单片机为核心

· 输入控制： 无外部输入

· 输出控制： 数字显示：使用七段显示器作为输出设备，通过 P1.6 和 P1.7 两个端口控制两个七段显示器的启动与关闭，从而显示数字的十位和个位。

显示接口：具体到硬件连接，P2 端口用于向七段显示器发送具体的显示编码，由程序中的 DISP 过程控制。

· 控制程序：

主控程序：控制程序以一个无限循环的方式运行，持续更新和显示数字。通过调用 PRODATA、DISP 和 PROADD 三个子过程，实现数字的计算、显示和更新。

PRODATA 过程：负责计算当前要显示的数字，并将数字转换成对应的七段显示编码。

DISP 过程：负责驱动七段显示器，将计算得到的显示编码输出到显示器上。包括延时控制以调整显示的持续时间。

PROADD 过程：负责更新显示的数字，实现从 0 递增至 99 再回到 0 的循环。

延时处理：使用 DELAY 子程序来实现基本的时间延迟，确保数字能在显示器上稳定显示足够的时间。

4.3. 硬件设计

系统核心采用 8051 单片机，该单片机在不进行外扩数据存储器和程序存储器的条件下，共有 32 个 I/O 可供使用。功能强，应用面广，价格低。

Proteus 器件选择: AT89C51、7SEG-MPX2-CA

用 Proteus 新建一个工程，绘出 8051 单片机应用系统电路，在 P1.6 、P1.7 两个端口分别连接控制数码管的开关， P 2.0~2.7 分别连接数码管 A~P。

4.3.1. 输入

本实验无外部输入

4.3.2. 输出

4.3.2.1. 参数选择

数码管选择：使用两个七段显示器（型号：7SEG-MPX2-CA）来分别显示十位和个位数字。

控制端口：

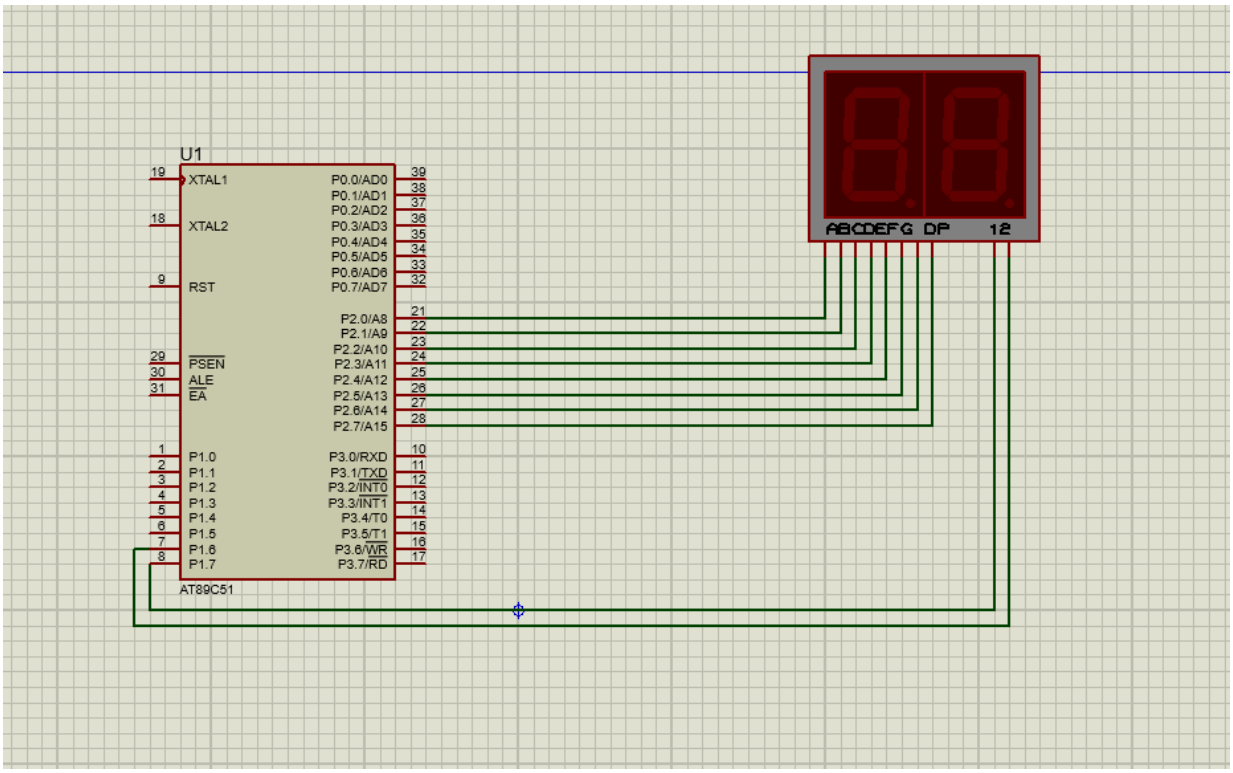
P1.6 和 P1.7：这两个端口分别用于控制两个数码管的开关。通过设置或清除这两个端口的值，可以控制相应数码管的显示状态。

P2.0 至 P2.7：这些端口连接到数码管的 A 至 P 段，用于传送具体的显示编码，控制数码管显示对应的数字。

4.3.2.2. 参数数值

显示编码：七段显示器的每个段（A 至 G 及 DP）通过 P2 端口的 P2.0 至 P2.7 分别控制。每个端口对应数码管的一个特定段。

4.3.3. 电路



4.4. 软件设计

4.4.1. 硬件资源规划

R0

参数资源基地址

始终为 0x30H

R0 在此程序中用于 DELAY 过程中作为循环计数寄存器。

R1

临时寄存器

在 DELAY 过程中用作内部循环的计数寄存器。

R6

参数状态 输出信号

初始值: 0b11111110

R6 用于存储从 TAB 表中读取的七段显示编码（十位部分）。程序中没有直接使用 0b11111110，这里的初始值描述可能是一个示例。

低电平有效，0 对应端口的 LED 亮

R6 的值最终会移动到端口 P2，控制七段显示器的显示（十位部分）。

R7

参数状态 序号

初始值: 0x00H

范围[0, 1, 2, 3]，大于等于 4 时置零

在实际程序中，R7 用于遍历数字 0-99，然后循环。每次循环结束时，如果 R7 的值等于 99，则重置为 0。描述中的范围[0, 1, 2, 3]可能需要根据实际应用进行调整，或者是一个示例。

每按一次 SEL，R7+1

在此程序中，R7 的值是通过 PROADD 自动增加，而不是通过按键操作。如果要实现按键操作，需要添加外部中断或轮询按键状态的代码。

R5

参数状态 输出信号

R5 用于存储从 TAB 表中读取的七段显示编码（个位部分）。

低电平有效，0 对应端口的 LED 亮

R5 的值同样移动到端口 P2，控制七段显示器的显示（个位部分）。

P2

参数数值 显示

端口 P2 用于向七段显示器输出当前的显示编码，由 R5 或 R6 的值提供。

4.4.2. 流程图

本实验较为简单，故不绘制

4.4.3. 代码

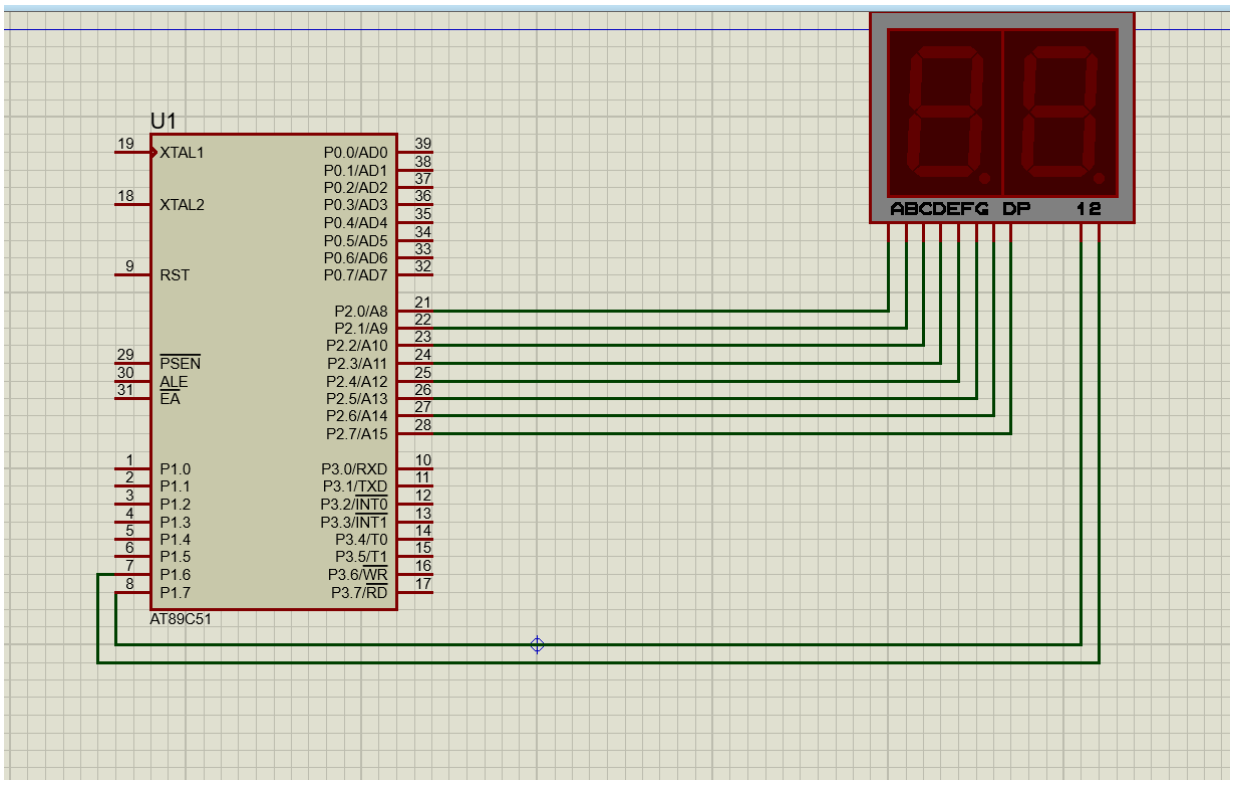
```
ORG 0000                ; 设置程序的起始地址为 0000
MOV DPTR, #TAB          ; 将数据指针 DPTR 设置为指向 TAB 表，TAB 表包含了七段显示的数字编码
MOV R7, #00             ; 初始化 R7 寄存器，R7 用于存储当前显示的数字（0-99）

START: CALL PRODATA     ; 调用 PRODATA 子程序，计算当前数字对应的七段显示码，并存储到 R5 和 R6
    MOV R3, #13         ; 设置循环计数器 R3 为 13，控制显示的持续时间
N2:   CALL DISP         ; 调用显示子程序，用于控制七段显示器显示数字
    DJNZ R3, N2         ; 循环调用显示子程序，直到 R3 计数器减至 0
    CALL PROADD         ; 调用 PROADD 子程序，更新 R7 的值，准备显示下一个数字
    JMP  START          ; 无条件跳转回 START，继续循环显示下一个数字

PROADD: MOV A, R7       ; 将 R7 的值移入累加器 A
     CJNE A, #99, LOOP  ; 比较 A 与 99，如果不等，则跳转到 LOOP 标签
     MOV A, #00         ; 如果 A 等于 99，将 A 重置为 0
     MOV R7, A          ; 将 A 的值（0）移回到 R7
     JMP LOOP1          ; 跳转到 LOOP1，结束子程序
LOOP:  INC A            ; 如果 A 不等于 99，增加 A 的值
     MOV R7, A          ; 将增加后的 A 值存回 R7
LOOP1:  RET             ; 返回到调用点

PRODATA: MOV A, R7      ; 将 R7 的值（当前数字）移入累加器 A
```

```
        MOV B, #10          ; 将 B 寄存器设置为 10，用于之后的除法运算
        DIV AB              ; 用 A 除以 B，结果：A 是商，B 是余数
        MOVC A, @A+DPTR     ; 通过查找 TAB 表获取 A（十位）的显示码，并将其存入 A
        MOV R6, A           ; 将 A（十位的显示码）存入 R6
        MOV A, B            ; 将 B（个位）移入 A
        MOVC A, @A+DPTR     ; 通过查找 TAB 表获取 A（个位）的显示码，并将其存入 A
        MOV R5, A           ; 将 A（个位的显示码）存入 R5
        RET             ; 返回到调用点

DISP:   SETB P1.6           ; 设置 P1.6 位，控制显示器的一部分（可能是多位显示器的选择）
        MOV A, R5           ; 将个位的显示码移入 A
        MOV P2, A           ; 将 A 的值输出到 P2 端口，控制七段显示
        CALL DELAY          ; 调用延时子程序
        CLR P1.6        ; 清除 P1.6 位
        SETB P1.7       ; 设置 P1.7 位，切换到显示另一部分
        MOV A, R6           ; 将十位的显示码移入 A
        MOV P2, A           ; 将 A 的值输出到 P2 端口
        CALL DELAY          ; 再次调用延时子程序
        CLR P1.7        ; 清除 P1.7 位
        RET             ; 返回到调用点

DELAY:  MOV R0, #100        ; 设置延时循环的外层计数器 R0 为 100
N1:     MOV R1, #13             ; 设置延时循环的内层计数器 R1 为 13
        DJNZ R1, $          ; 内层延时循环
        DJNZ R0, N1         ; 外层延时循环
        RET             ; 返回到调用点

TAB:    DB  0c0h,0f9h,0a4h,0b0h,99h,92h,82h,0f8h,80h,90h
        ; 数据表，包含 0-9 数字的七段显示编码
END                     ; 程序结束
```

5. 实验结果和分析

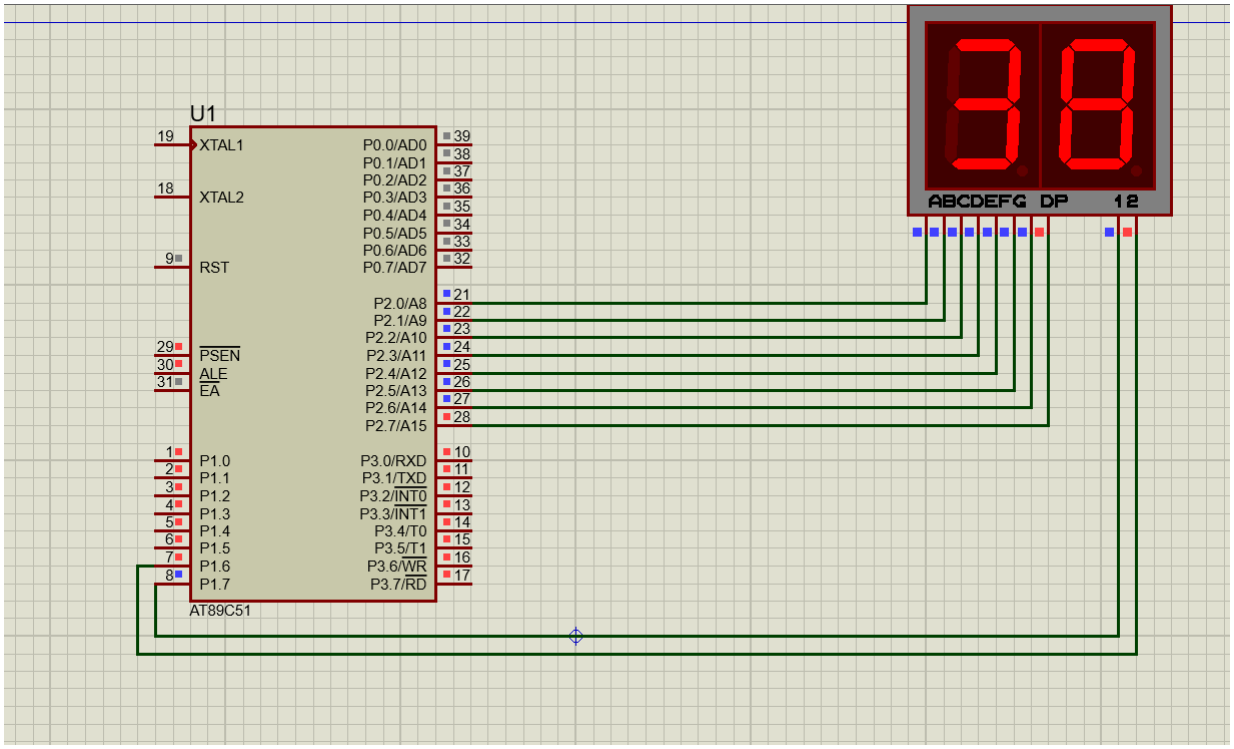
5.1. 系统测试

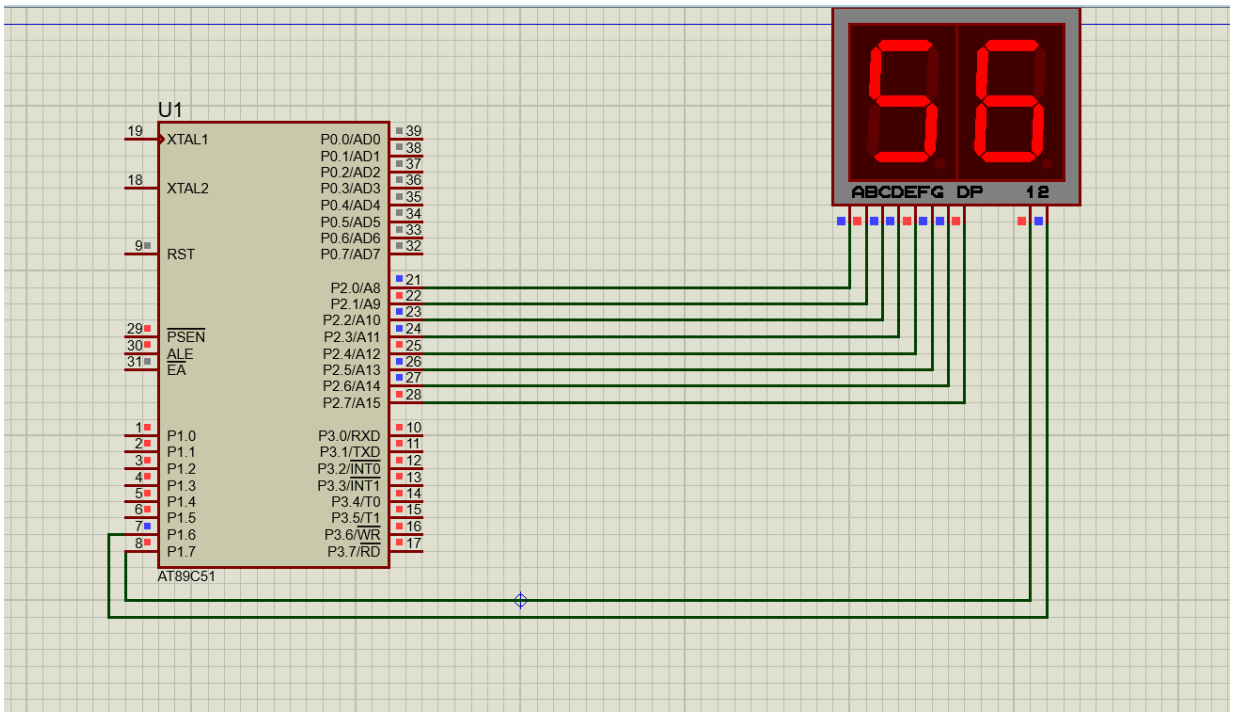
· 硬件电路设计图


U1
XTAL1
XTAL2
RST
PSEN
ALE
EA
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7
P2.0/A8
P2.1/A9
P2.2/A10
P2.3/A11
P2.4/A12
P2.5/A13
P2.6/A14
P2.7/A15
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
AT89C51
ABCDEFG DP
1 2

开始运行

运行截图


U1
XTAL1
XTAL2
RST
PSEN
ALE
EA
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7
P2.0/A8
P2.1/A9
P2.2/A10
P2.3/A11
P2.4/A12
P2.5/A13
P2.6/A14
P2.7/A15
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
AT89C51
ABCDEFG DP
1 2



Keil5 运行截图

```
linking...
Program Size: data=8.0 xdata=0 code=79
creating hex file from ".\Objects\Test"...
".\Objects\Test" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:00
```

5.2. 分析

系统运行结果表明，本实验设计的基于 51 单片机的硬件电路和软件组成的系统，采用数码管, IO 硬件和循环、条件程序，实现了

数码管变化由 00->99，不断循环，表示从 0.0 秒到 9.9 秒变化。99 过后回到 00，重新循环。

完成了实验任务。

6. 思考和讨论

6.1. DEBUG

由于是第一次实验，刚开始接线发生了错误，错误把 P1.6,P1.7 接线接反，故无法正确显示循环、

后修正，能够正确显示循环

6.2. 总结

作为第一次实验，较为系统的领略到了微机课程老师所说的软硬件结合，不仅仅是代码上的 bug 会导致实验失败，硬件上连接的错误也会导致实验出错。故在分析故障时需要软硬件同时结合考虑。以上是我这次实验最主要的收获。