

# 浙江大学实验报告

专业: 生物医学工程  
姓名: \_\_\_\_\_  
学号: \_\_\_\_\_  
日期: 2025.3.11  
地点: 东 1B 416

课程名称: 微机原理及其应用 指导老师: 张恒义 实验类型: 微机实验  
实验名称: 键控数字显示 成绩: \_\_\_\_\_ 签名: \_\_\_\_\_

## 一、实验目的和要求

### 1. 实验目的

(1). 通过基于 PROTEUS 软件仿真设计 51 单片机系统, 深入理解单片机的基本工作原理及其在实际应用中的操作流程。

(2). 熟练掌握简单的查表程序编写方法, 这有助于在处理数据查找与对应操作时能够高效准确地实现功能。

(3). 全面掌握 I/O 口开关量输入方法, 包括如何正确设置 I/O 口的初始状态、读取输入状态等技能。

(4). 透彻掌握数码管显示器的显示方法, 涵盖数码管与单片机的电路连接方式、显示控制字的设置以及根据不同数字显示需求输出正确控制信号等知识要点。

### 2. 实验要求

(1). 使用 PROTEUS 软件仿真构建一个 51 单片机系统。

(2). 该系统需具备记录按键按下次数 (范围为 0 - 9) 的功能, 当次数达到 10 次后, 清零并重新计数。

(3). 将按键按下的次数在一位七段式共阳极数码管上准确显示出来。

## 二、实验内容和原理

### 1. 硬件电路设计

根据实验要求, 以 8051 单片机为核心, 设计系统硬件电路, 并以按钮开关电路连接到 8051 的 I/O 口作为开关量输入, 以连接到 I/O 的 7 段数码管作为显示输出电路。

### 2. 控制软件设计

(1) 键盘输入控制程序: 用 i/o 口输入开关量时, 应先向 I/O 口写高电平, 然后再读入 I/O 口的状态。程序需要循环查询 I/O 端口状态, 当发现端口状态从高电平变成低电平时, 则认为是按键被按下, 按键按下次数加 1。需要注意的是, 按键动作较慢, 而查询按键状态的循环很快, 一次按下过程可能被查询很多次, 所以需要正确识别按下动作。设置一个按钮状态记录变量, 初始设置为 1, 标示 I/O 为高电平状态, 当查询 I/O 状态为低电平且标示变量为 1 时, 才认为是按下了按键, 按键次数加 1, 并将标示变量置为 0。当查询 I/O 状态为高电平时, 标示变量设置为 1。等待下一次的按钮行为。本实验装置要求仅显示一位数字, 当计数 10 次后, 计数器清零, 重新计数。

(2) 数字显示控制程序: 按照 7 段共阳极数码显示器的显示要求, 以及与单片机的电路联接, 当向 P2 口输出相应的控制字, 即可显示 0-9 数字。端口控制字设为常数表, 当设定常数表首地址于 DPTR, 待显示数字置于累加器 A 中, 则指令 MOVC A,@A+DPTR 获得相应的控制字, 将控制字输出到 P2 口, 完成显示。

### 3. 元件列表

80c51 单片机, 按钮 BUTTON, 7 段共阳极数码显示器 7SEG Common Anode, 通用电阻 1K resistor generic

## 三、 主要仪器设备

计算机一台, 用于安装 keil C51 和 proteus 软件, 并进行程序编写、电路绘制以及仿真操作。

## 四、 操作方法和实验步骤

### 1. 硬件电路设计

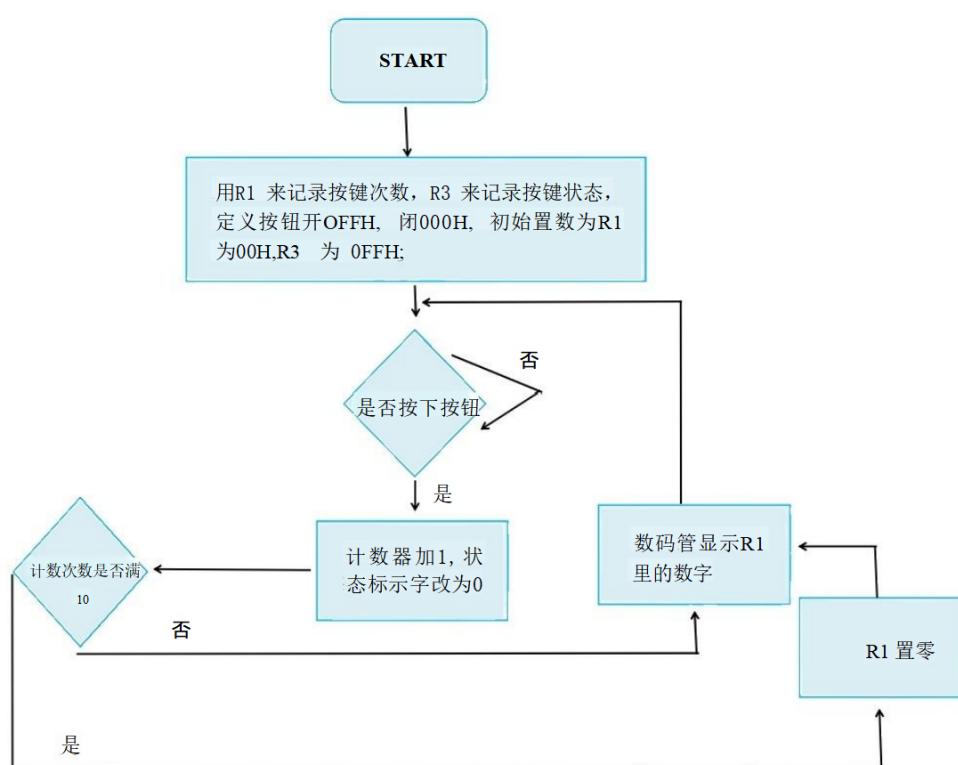
(1) 说明按钮开关量输入:

按钮 BUTTON 与一限流电阻 (RESISTOR,1k) 串联, 电阻上接电源, 按钮下接地, 从按钮与电阻中间引线接入单片机 I/O 口, 按钮开时, 输入高电平, 按钮闭合, 输入低电平。

(2) 数码管显示电路:

数码管显示器分共阳极 (7SEG Common Anode) 和共阴极 (7SEG CommonCathode) 两种。7 段数码管显示器由 7 个 LED 发光二极管共用电源或地组成, 每个控制引脚控制一个二极管, 通过不同的亮暗组合控制, 可显示数字 0~9; 用 7 个 i/o 端口的输出来控制数码管。如用 P2 口的低 7 位, P2.0 P2.7 对应共阳极数码管的 a g, 则 0~9 数字对应的控制字为: 0ffH, 0c0H, 0f9H, 0a4H, 0b0H, 099H, 092H, 082H, 0f8H, 080H, 090H。

### 2. 软件流程图



### 3. 程序代码

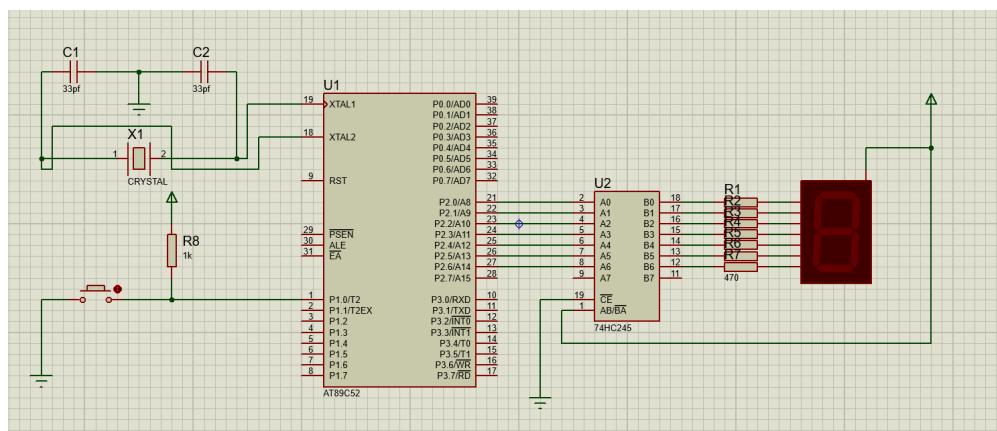
```

ORG 0000H
AJMP MAIN;
ORG 0100h
Start:MOV R3,#0FFH
Loop0:MOV R1, #00H
Loop: MOV A,R1 ; 查表显示保存在 R1 中的计数
      MOV DPTR, #0200H
      MOVC A, @A+DPTR
      MOV P2,A
      MOV P1, #0FFH ; 读按键状态, 先置端口为 1
      JB P1.0, LNext ;P1.0 高电平按钮为开状态, 转高电平处理模块
      CJNE R3,#0FFH,Loop ;P1.0 低电平按钮为闭合状态, 若前一次循环也是 闭合状态, 则转显示查询循环入口
      INC R1
      MOV R3,#00H ; 确认是按键按下动作, 计数器加 1, 状态标示字改 为 0
      CJNE R1,#0AH,Loop ; 计数次数不是 10, 转显示查询循环入口
      JMP Loop0 ; 计满 10 次, 转计数器置 0
LNext: MOV R3,#0FFH ;P1.0 高电平按钮为开状态, 状态标示字改为 offh
      JMP Loop
ORG 0200H
DB 0c0H,0f9H,0a4H,0b0H,099H,092H,082H,0f8H,080H,090H,0ffH ;7 段共阳极数码管显示器 0-9 状态字表
END

```

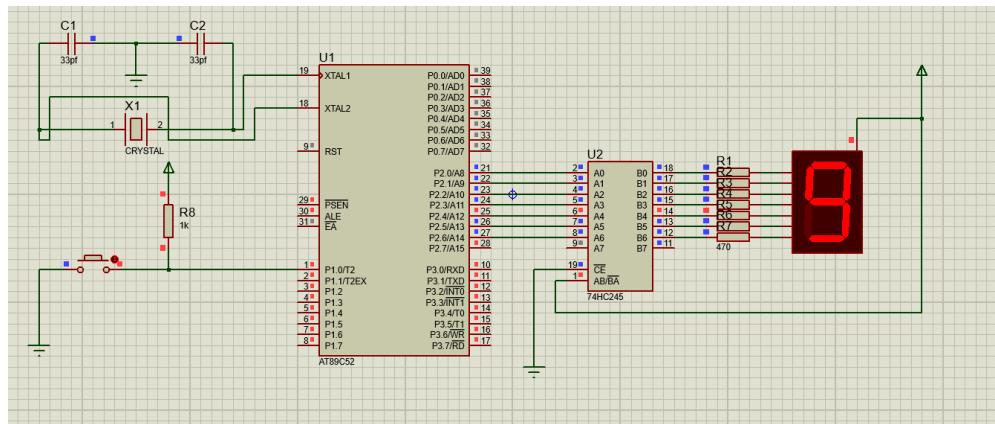
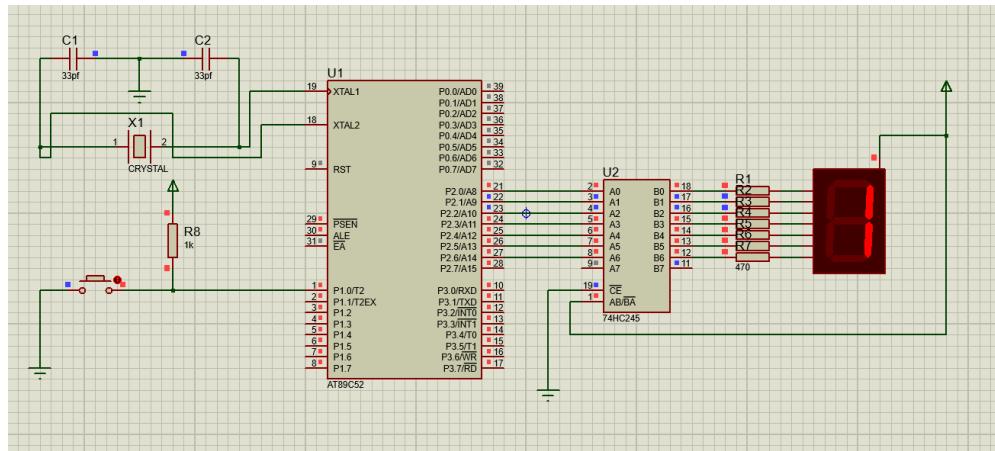
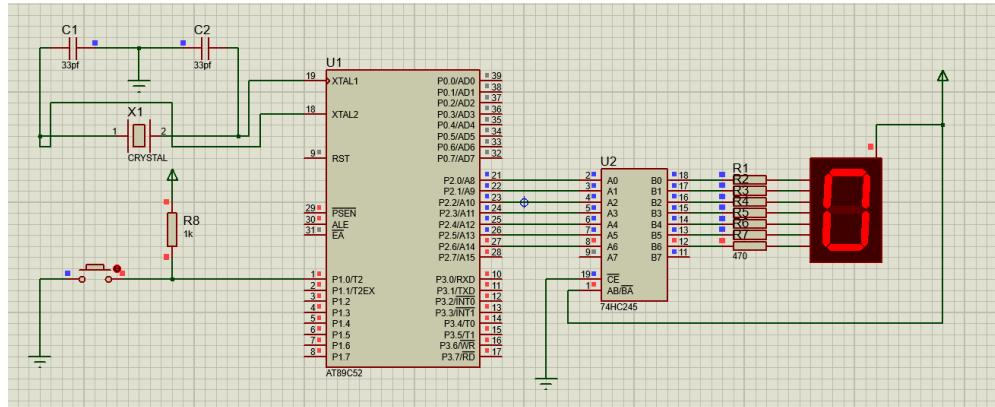
## 五、 实验结果与分析

### 1. 硬件电路设计图



## 2. 软件运行结果

初始状态下, 数码管显示 0, 此后每按一次按钮, 数码管显示的数字 +1, 最高达到 9, 此时再按下按钮, 数码管数字清零



## 六、 拓展实验

### 1. 实验要求

将个位数字的数码管显示扩展到两位, 范围为 00-99

### 2. 硬件电路扩展

需要增加一个数码管, 同时需要实现两个功能, 一是个位从 9 到 0 向十位进位的功能, 二是到达 99 时个位与十位都清零的功能。

### 3. 软件扩展思路

需要增加一个寄存器用来判断是否是个位第一次置 0 循环, 以及另一个寄存器存储十位的数字。如果不是第一次进入置 0 循环, 则在个位清零同时进位。当十位也到达 10 时, 将个位和十位都置 0

### 4. 程序代码

---

```

ORG 0000H
    JMP Start
    ORG 0100H
Start:MOV R3,#0FFH

Loop2:MOV R5,#00H    ;    进行十位置零, 判断是否第一次进入个位置 0 循环
    MOV R2,#00H
Loop0:MOV R1, #00H    ; 个位置零, 判断是否第一次进入循环
    CJNE R2,#00H,Loop3
    jmp loop

Loop: MOV A,R1 ; 个位显示
    MOV DPTR, #0200H
    MOVC A, @A+DPTR
    MOV P2,A

    MOV A,R5 ; 十位显示
    MOV DPTR, #0200H
    MOVC A, @A+DPTR
    MOV P3,A
    MOV R2,01H

    MOV P1, #0FFH
    JB P1.0, LNext
    CJNE R3,#0FFH,Loop
    INC R1
    MOV R3,#00H
    CJNE R1,#0AH,Loop

```

实验名称: 键控数字显示

姓名:

学号:

```
JMP Loop0
LNext: MOV R3,#0FFH
        JMP Loop
Loop3: INC R5 ; 进位与判断十位是否满 10
        CJNE R5,#0AH,loop
        jmp loop2

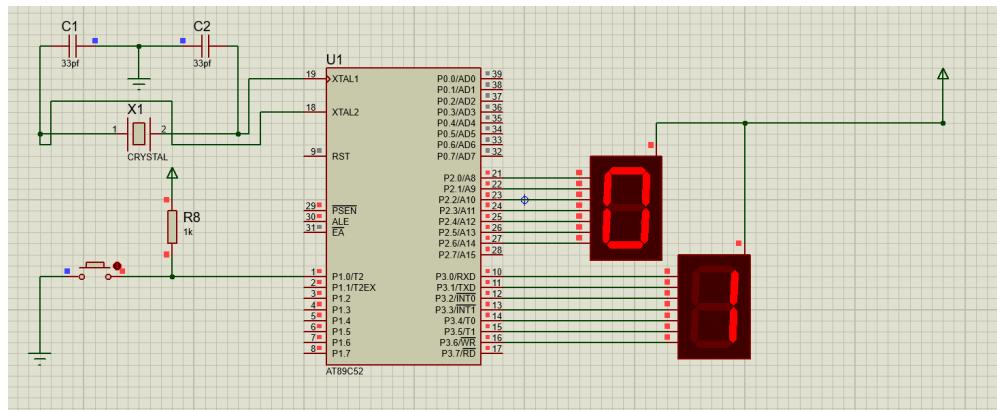
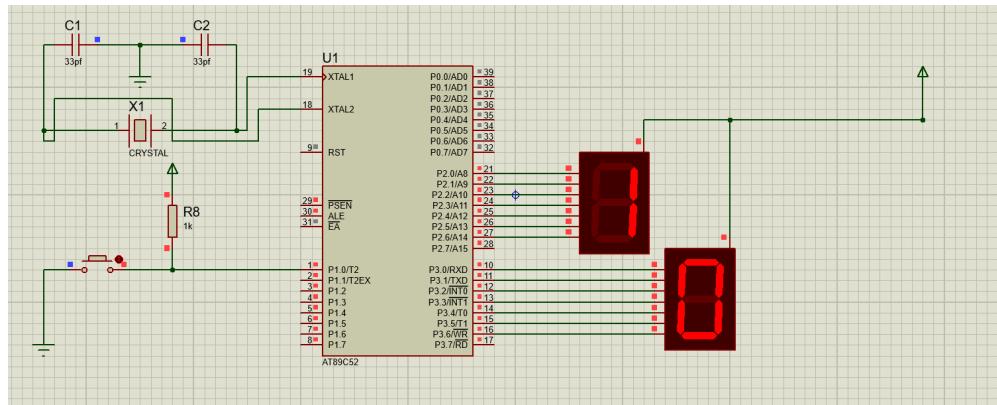
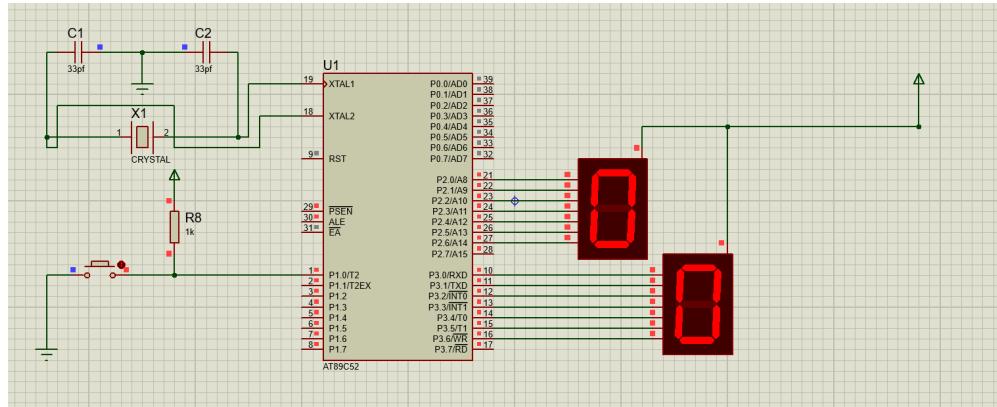
ORG 0200H
DB 0c0H,0f9H,0a4H,0b0H,099H,092H,082H,0f8H,080H,090H,0ffH
END
```

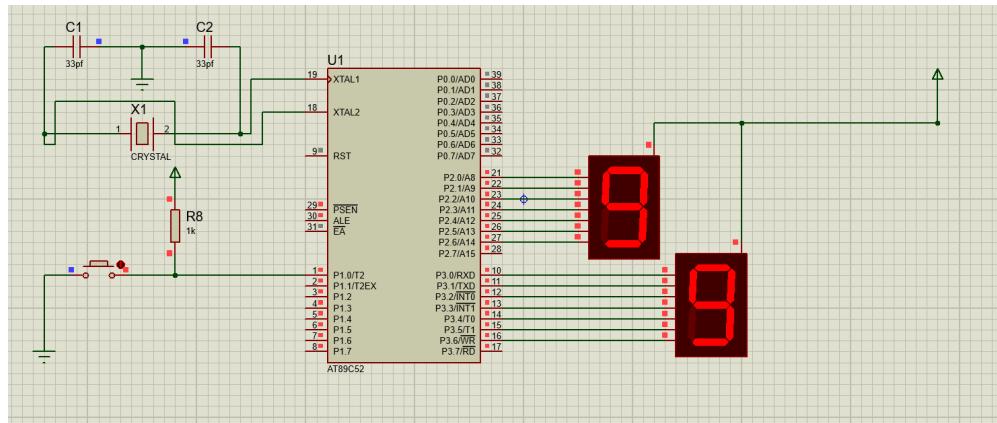
---

线  
订  
装

## 5. 软件运行结果

初始状态 00, 按一次按钮后个位加 1, 个位达到 10 时, 实现向十位进位, 达到最大状态 99 后, 再次按下按钮, 清零





## 七、 讨论、心得

本次实验让我继续熟悉了 keil、proteus 软件开发环境并熟悉了单片机结构，熟悉了简单的的程序编写和硬件电路绘制。本次实验仍然是在老师给出的示例程序的框架下进行的，扩展实验部分自己也是了解了大概的思路和做法后，在原有的代码上面进行了补充。不同的是进行了理论课的学习，我已经能理解代码的大致执行流程，希望下次能够具备独立编写汇编代码的能力。

本次实验也让我对单片机软硬件结合的特点有了更深的体会，想要学好单片机，一定要具备软硬件结合的思想，并且自己独立完成代码编写，思维能力才会不断地提高。