

浙江大学实验报告

专业: _____
姓名: _____
学号: 3230103696
日期: 2025.3.25
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: 三键多信息输入实验 成绩: _____ 签名: _____

一、实验目的和要求

1. 实验目的

- (1). 掌握和熟悉循环与多分支汇编程序设计与开发技巧。
- (2). 学会 I/O 端口的复杂信息输入与输出方法。
- (3). 通过本实验, 学习三键多参数设置方法, 加深对控制方法基本应用操作的理解。
- (4). 理解并掌握单片机在实际应用中的硬件和软件设计方法。

2. 实验要求

- (1). 利用 proteus 软件, 设计一个基于三键输入的 4 参数信息输入应用系统, 每个参数值可以进行单位增减操作。
- (2). 使用 4 个 LED 灯分别标识 4 个参数, 仅当前选中的参数对应的 LED 灯亮起, 其余熄灭。(3). 参数的数值范围为 0-8, 与 LED 灯亮的个数对应。
- (4). 在实验前, 对任务进行详细的需求分析, 并撰写需求分析报告和系统设计报告。
- (5). 绘制详细的硬件电路图, 进行软件设计并绘制软件流程图。
- (6). 制定系统测试方案, 并在测试完成后撰写测试分析报告。
- (7). 完成实验后, 总结实验心得。

二、实验内容和原理

1. 硬件电路设计

(1) 按键输入电路原理: 采用三个按钮开关电路连接到 8051 单片机的 I/O 口作为开关量输入, 防止灌入单片机引脚的电流过大, 需串联限流电阻。(2)LED 显示电路原理: LED 正常点亮时的电流一般都在 20mA 左右, P2 和 P3 口直接驱动能力不够, 低电平时灌电流最大是 6mA, 而高电平时的拉电流最大也只有 0.22mA。因此在驱动 LED 时采用三极管来提高 P2 和 P3 口的驱动能力。

2. 控制软件设计

对于参数选择按键, 用一变量记录按键次数, 并对参数个数取模操作, 可循环参数选择, 用一个 I/O 口控制 LED 对应一个参数, 用 LED 灯的亮/熄灭来表示是否选中该参数。每个参数各用一个参数记录参数数值, 用 8 位 I/O 端口驱动 LED 灯, 显示每个参数的数值。当按增量或减量参数时, 由参数选择变量的记录值, 选择要执行操作的参数, 并对该参数的数值进行加减操作, 同时设置数据显示 I/O 口, 显示修改后的值。

三、主要仪器设备

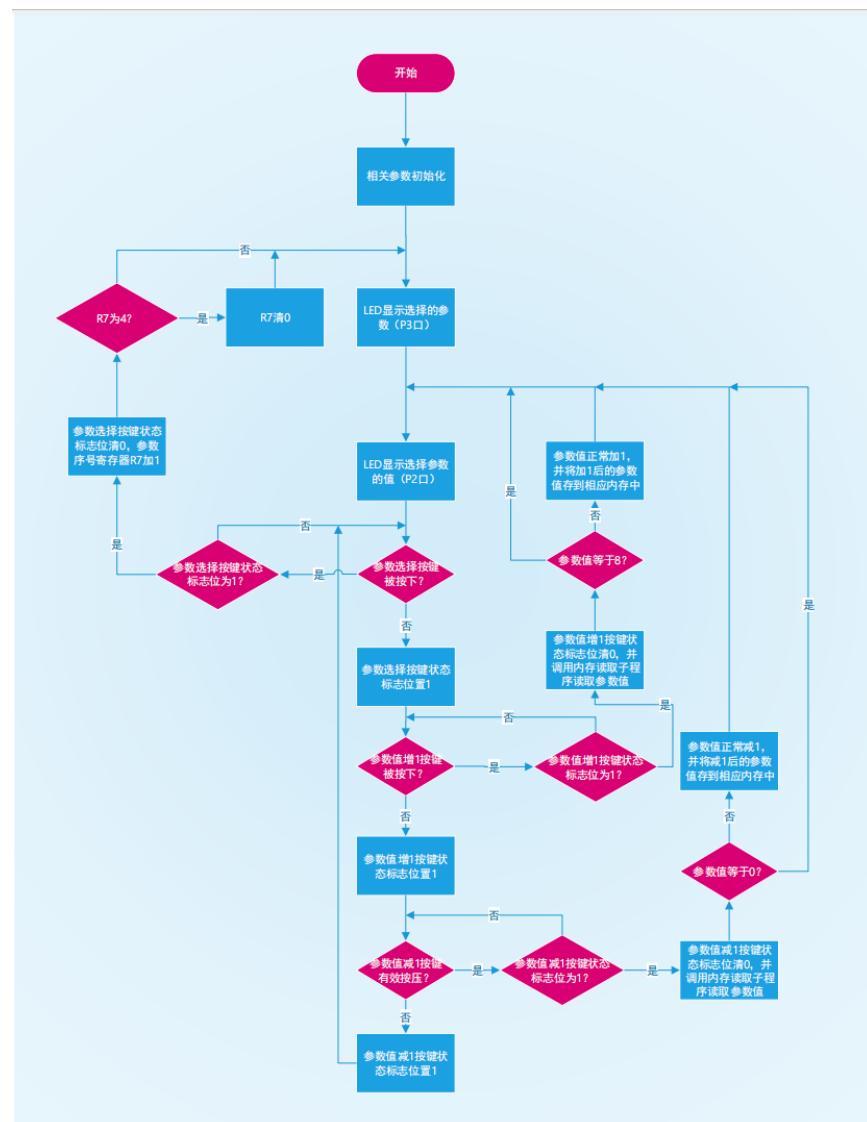
微型计算机、单片机开发工具软件 Keil C51, Proteus

四、操作方法和实验步骤

1. 硬件电路设计

LED 和三极管组成 LED 阵列电路，其中 P2 口驱动 8 个 LED 阵列用于显示参数值，P3 口驱动 4 个 LED 阵列用于显示选择的是何种参数。

2. 软件流程图



3. 程序代码

```

LASTSEL BIT 00H; 参数 SEL 选择键原状态
LASTINC BIT 01H; 参数 INC 增 1 键原状态
LASTDEC BIT 02H; 参数 DEC 减 1 键原状态

BUTTONSEL BIT P1.0; 新读入参数 SEL 选择键状态
BUTTONINC BIT P1.1; 新读入参数 INC 增 1 键状态
BUTTONONDEC BIT P1.2; 新读入参数 DEC 减 1 键状态

ORG 0000h
JMP Start

ORG 0100H
Start: MOV P2,#0FFH
       MOV P3, #0FEH

       MOV 20H,#0FFH ; 置位, 初始值为 1
       MOV 30H,#00H ; 30H-33H 地址存四个参数的值
       MOV 31H,#00H
       MOV 32H,#00H
       MOV 33H,#00H

       MOV R0, #30H ; 赋 R0 初值用于寄存器间接寻址

       SETB LASTSEL ; 置位, 原来按键都处于开状态
       SETB LASTINC
       SETB LASTDEC

       MOV R6, #00H
       MOV R7, #00H

STARTKEY:MOV P1,#0FFH
       JB BUTTONSEL, CSELKEY ; 按键选择按键没被按下, 直接跳转到高电平处理模块
       JNB LASTSEL, CSELKEY ; 前一刻按键选择按键被按下, 跳转到高电平处理模块
       JMP STARTKEY ; 循环

ADDR_VALUE_GET: MOV A,RO
       ADD A,R7 ; 将 R0 基地址和 R7 当前选择参数的序号相加得到当前选择参数值的存储地址
       MOV R1,A
       MOV A, @R1 ; 寄存器间接寻址, 得到选择参数的参数值
       RET

SEL_DISPLAY: MOV A,R7 ; 查表显示选择的参数, 对应 P3 口控制的 LED
       MOV DPTR, #TABLE1

```

```

MOV C A, @A+DPTR
MOV P3, A
LJMP DISPLAY

```

DISPLAY: ACALL ADDR_VALUE_GET ; 获得当前参数的值, 存储在 A 种, 查表对应 P2 口 LED

```

MOV DPTR, #TABLE
MOV C A, @A+DPTR
MOV P2, A

```

KEYSEL_DETECT: MOV P1, #OFFH ; 读按键状态, 先置端口为 1

```

JB BUTTONSEL, CSELKEY ; 参数选择按钮高电平为开状态, 转参数选择按钮高电平处理模块, 并从高电平
JNB LASTSEL, KEYSEL_DETECT; 参数选择按钮低电平为闭合状态, 若前一次按钮也是闭合状态, 则转回继续
INC R7 ; 确认是按键按下动作, 存储参数序号寄存器 R7 加 1,
CLR LASTSEL ; 前次状态清零
CJNE R7, #4, SEL_DISPLAY ; 计数次数不是 4, 转显示循环入口
MOV R7, #0 ; 若是 4, 则给参数选择寄存器赋值 0 后转显示
LJMP SEL_DISPLAY ; 计数次数不是 4, 转显示查询循环入口

```

CSELKEY: MOV C, BUTTONSEL
MOV LASTSEL, C

KEYINC_DETECT: MOV P1, #OFFH ; 读按键状态, 先置端口为 1

```

JB BUTTONINC, CINCKEY ; 参数增值按钮高电平为开状态, 转参数增值按钮高电平处理模块, 并从高电平
JNB LASTINC, KEYINC_DETECT; 参数选择按钮低电平为闭合状态, 若前一次按钮也是闭合状态, 则转回继续
CLR LASTINC ; 前次状态清零
CJNE A, #8, KEYINC; 计数次数不是 8, 转增值处理系统
LJMP DISPLAY ; 计数次数是 8, 转 P2 显示查询循环入口

```

KEYINC: ACALL ADDR_VALUE_GET; 获取当前参数值
INC A ; 当前参数值 +1
MOV @R1, A ; 改变对应地址的参数值
LJMP DISPLAY ; 转 P2 口显示模块

CINCKEY: MOV C, BUTTONINC
MOV LASTINC, C

KEYDEC_DETECT: MOV P1, #OFFH ; 读按键状态, 先置端口为 1

```

JB BUTTONDEC, CDECKEY ; 参数减值按钮高电平为开状态, 转参数减值按钮高电平处理模块, 否则顺序
JNB LASTDEC, KEYDEC_DETECT; 参数减值按钮低电平为闭合状态, 若前一次按钮也是闭合状态, 则转回继续
CLR LASTDEC ; 前次状态清零
ACALL ADDR_VALUE_GET
CJNE A, #0H, KEYDEC; 计数次数不是 0, 转显示循环入口
LJMP DISPLAY ; 计数次数是 0, 转显示查询循环入口

```

KEYDEC: ACALL ADDR_VALUE_GET
DEC A
MOV @R1, A
LJMP DISPLAY

CDECKEY: MOV C, BUTTONDEC
MOV LASTDEC, C
JMP KEYSEL_DETECT

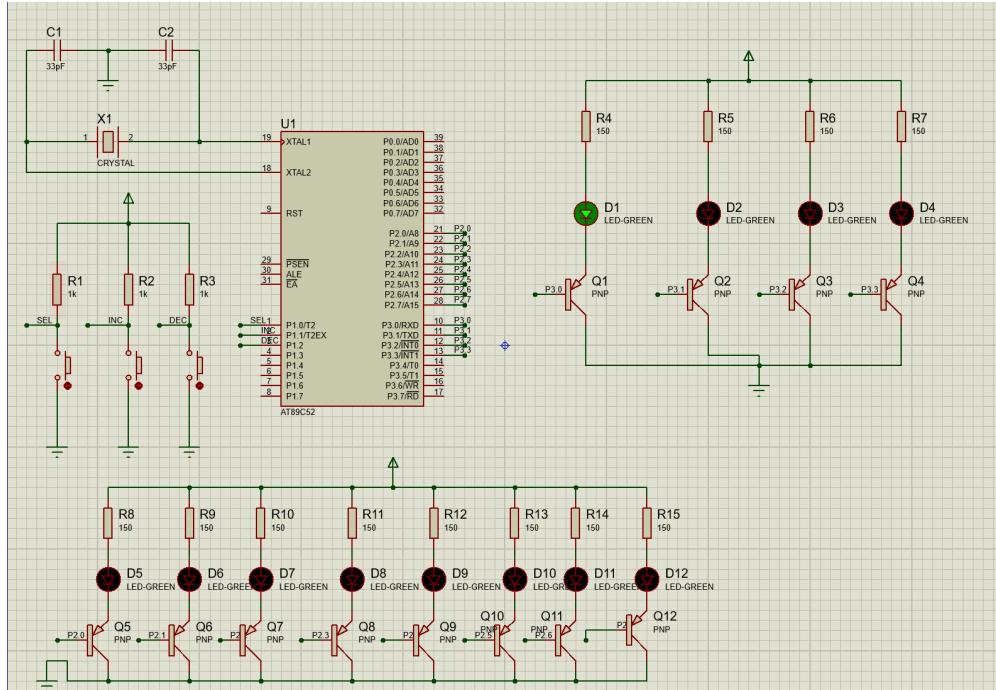
```

ORG 0200H
TABLE:DB 0FFh,0FEh,0FCh,0F8h,0F0h,0E0h,0C0h,80h,00h
TABLE1:DB 0EH,0DH,0BH,07H
END

```

五、实验结果与分析

1. 硬件电路设计图



2. 软件编写经历

我这次代码的编写过程是在实验方案给出的例程基础上完善的，主要是补充了赋初值部分，P2 口显示模块，增值和减值处理模块。经过不断的 debug 最终完成了完整的代码。

我一开始没有正确理解 30-33H 四个连续地址存储参数的具体含义，也并没有正确理解 ADDRGETVALUE 模块的作用，只用了一个寄存器 R6 来存储参数，增值和减值操作时也都直接对 R6 操作，导致当按下 SEL 按键时，P2 口输出维持上一状态，不符合设计要求。于是我又在 P2 口显示模块处最开始的地方加上给 R6 赋 0 的代码，期望达到一种清零效果。一开始仿真后我以为符合要求，但我发现这样的结果就是无论如何转换按键都无法实现存储参数的目的，转换参数后所有的灯都会熄灭，这显然也不符合设计要求。

最后我详细看了例程对于代码的说明，才发现需要用四个连续地址存储四个参数，并用 ADDRGETVALUE 模块获取当前的参数值，再对当前参数值进行相应操作，才最终得到了正确的结果。

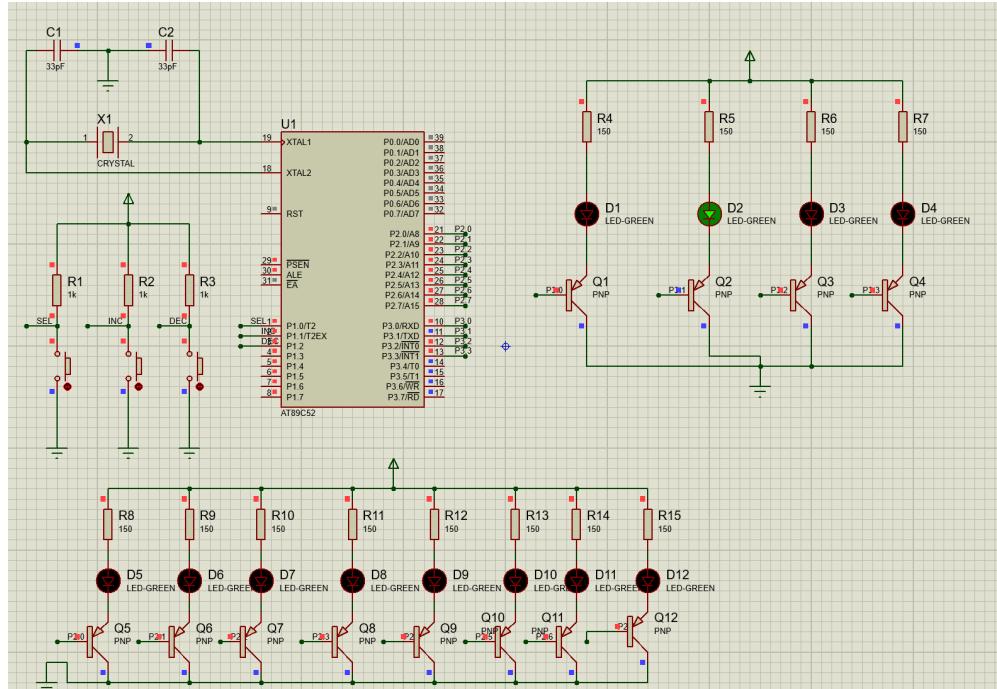
3. 系统测试方案

1. 按下按键转换按钮，观察 P2 口参数 LED 灯显示是否正确
2. 按下参数增 1 按钮，观察 P3 口 LED 灯亮灯数是否正确增 1，以及到达 8 后能否保持
3. 按下参数减 1 按钮，观察 P3 口 LED 灯亮灯数是否正确减 1，以及到达 0 后能否保持

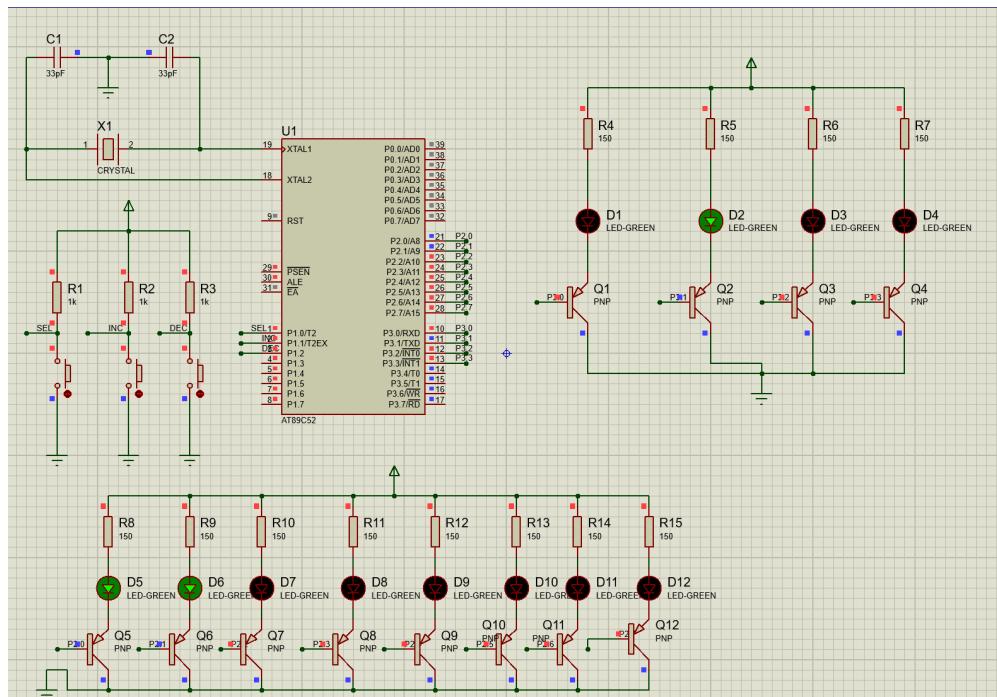
4. 按下按键转换按钮, 观察转换后的参数 2 是否与前次设置是否一致

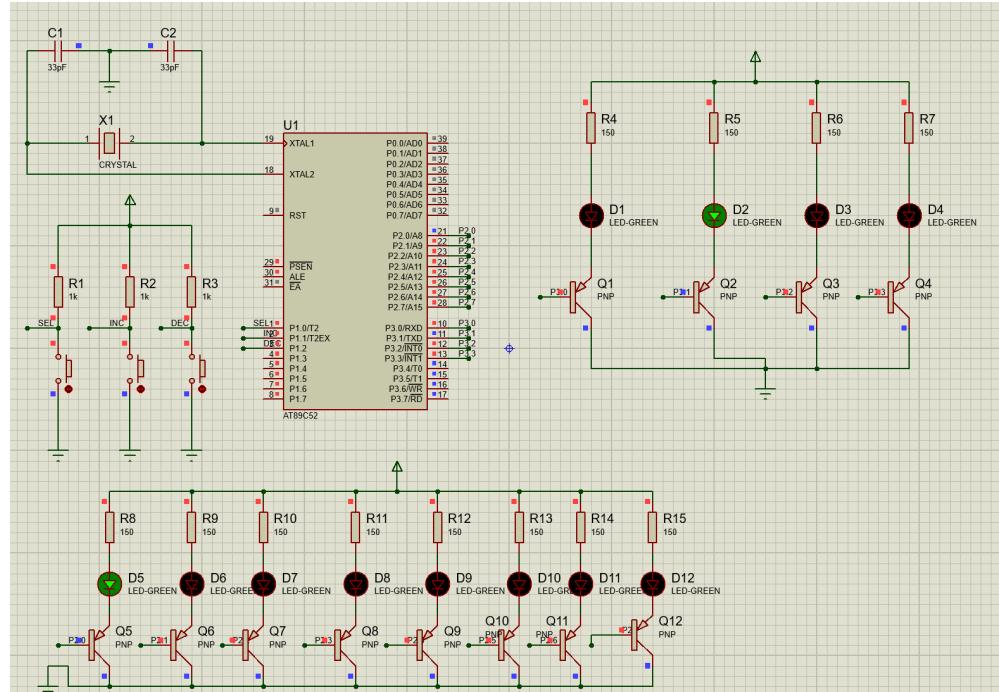
4. 软件运行结果

按下参数转换按钮, 转换参数, 对应 P3 口 LED 灯亮灭



按下参数值增 1/减 1 按钮, 参数值增 1/减 1, 对应 P2 口 LED 灯亮的个数





六、讨论、心得

本次实验我是相对独立地完成了整段代码的编写，虽然是在例程基础上完善的，但是许多模块都是自己独立完成的，期间遇到了不少问题，也反复琢磨代码进行仿真调试，最后不断修改完善，使得问题得到了解决。我觉得微机原理的学习还是需要自己多动手写代码加深理解，才能自己领会到软硬件结合的思想。在不断 debug 的过程中能提升自己对代码的理解和编写代码的能力。本次实验的三键多输入系统也比较有意义，是在第一个实验流水灯基础上完善的，加入了多个按键控制开关，来实现对流水灯的控制。