

微机控制技术实验指导书

姜长城 徐静 编



华南理工大学机械与汽车工程学院
机械工程实验分中心

目录

第一章 综合实验/仿真系统简介	5
1.1 实验平台的硬件组成总览	5
1.2 AT89S52 单片机核心板	7
第二章 实验教学方式及软、硬件使用简介	8
2.1 实验教学方式	8
2.2 Proteus 仿真实验步骤	8
2.3 keil 编程步骤	12
2.4 Plink 下载程序方法	16
第三章 实验练习	19
3.1 基础实验一 I/O 口实验	19
一、实验目的	19
二、实验设备和器件	19
三、实验要求	19
四、硬件连接图	19
五、Proteus 原理图设计	20
六、程序流程图及程序	22
七、实验报告要求	23
3.2 基础实验二 流水灯实验	24
一、实验目的	24
二、实验设备和器件	24
三、实验要求	24
四、硬件连接图	24
五、Proteus 原理图设计	25
六、程序流程图	26
七、实验报告要求	26
3.3 基础实验三 外部中断计数实验	27
一、实验目的	27

二、实验设备和器件	27
三、实验要求	27
四、硬件连接图	27
五、Proteus 原理图设计	28
六、程序流程图及参考程序	29
七、实验报告要求	34
3.4 基础实验四 PWM 蜂鸣器（定时器）实验	35
一、实验目的	35
二、实验设备和器件	35
三、实验要求	35
四、实验原理	35
五、硬件连接图	36
六、Proteus 原理图设计	37
七、程序流程图及参考程序	38
八、实验报告要求	41
3.5 基础实验五 数码管驱动实验	42
一、实验目的	42
二、实验设备和器件	42
三、实验要求	42
四、实验原理	42
五、硬件连接图	43
六、Proteus 原理图设计	43
七、程序流程图	45
八、实验报告要求	45
3.6 基础实验六 矩阵键盘扫描实验	46
一、实验目的	46
二、实验设备和器件	46
三、实验要求	46

四、实验原理	46
五、硬件连接图	47
六、Proteus 原理图设计	48
七、程序流程图	50
八、实验报告要求	50
3.7 创新实验七 步进电机控制实验	51
一、实验目的	51
二、实验设备和器件	51
三、实验要求	51
四、实验原理	51
五、硬件连接图	52
六、Proteus 原理图设计	53
七、程序流程图	55
八、实验报告要求	55
3.8 创新实验八 PC 与单片机串口通讯实验	56
一、实验目的	56
二、实验设备和器件	56
三、实验要求	56
四、实验原理	56
五、硬件连接图	57
六、Proteus 原理图设计	58
七、程序流程图	60
八、实验报告要求	60

第一章 综合实验/仿真系统简介

FB-EDU-MCU-F 型微控制器仿真实训实验箱完全覆盖各大院校开设的《微机控制技术》教学大纲的要求。本实验箱将 Proteus 仿真与硬件实训紧密结合，大大提高实验实训效果。实验箱提供了计算机平台、Proteus 软件、丰富的硬件资源与接口电路、多种处理器模块、CPLD/FPGA 模块，还包括单片机的一些典型应用电路。

同时，该实验箱可支持 8051、8086、AVR、PIC、MSP430、TI DSP2000、ARM、CORTEX 等多种单片机的实验实训，还支持 CPLD/FPGA 的实验实训。不仅提供了大量的验证性实训电路，还提供了大量的可选配的实训系统，可完成许多设计性实训及考核任务。

1.1 实验平台的硬件组成总览

该实验平台由丰富的硬件资源组成，按功能划分如表 1-1 所示。

表 1-1 实验平台的主要组成电路

普通实验模块	显示实验模块	拓展实验模块
74LS373 锁存器电路	16*16 点阵显示实验电路	热敏电阻传感电路
74LS47BCD 译码器电路	LCD1602 字符液晶屏显示电路	DS18B20 单线温度传感电路
74LS245 总线收发电路	LCD12864 点阵液晶屏显示电路	温度加热控制电路
74LS04 反相器电路	八位 8 段码 LED 数码管电路	湿度检测电路
74LS164 串入并出移位电路	八位逻辑电平显示电路	红外收发电路
74LS165 并入串出移位电路	八位轻触按键电路	两路 MAX232 接口通讯电路
一路继电器控制电路	4*4 矩阵键盘带中断输出电路	步进电机及驱动电路
两路光耦隔离电路		
直流电机及驱动电路		

同时，该平台使用 USB2.0 接口的通用编程器，具备烧录 AT89S51、AT89S52、PIC16F877、PIC18F4520、PIC24FJ64、ATMEGA8、ATMEGA16、ATMEGA128 及 ATMEGA168 芯片的能力。

结合本课程的需要，配备 ICE52 专业 51 仿真器对 AT89S52 芯片进行在线仿真。实验箱的布局如图 1-1 所示。

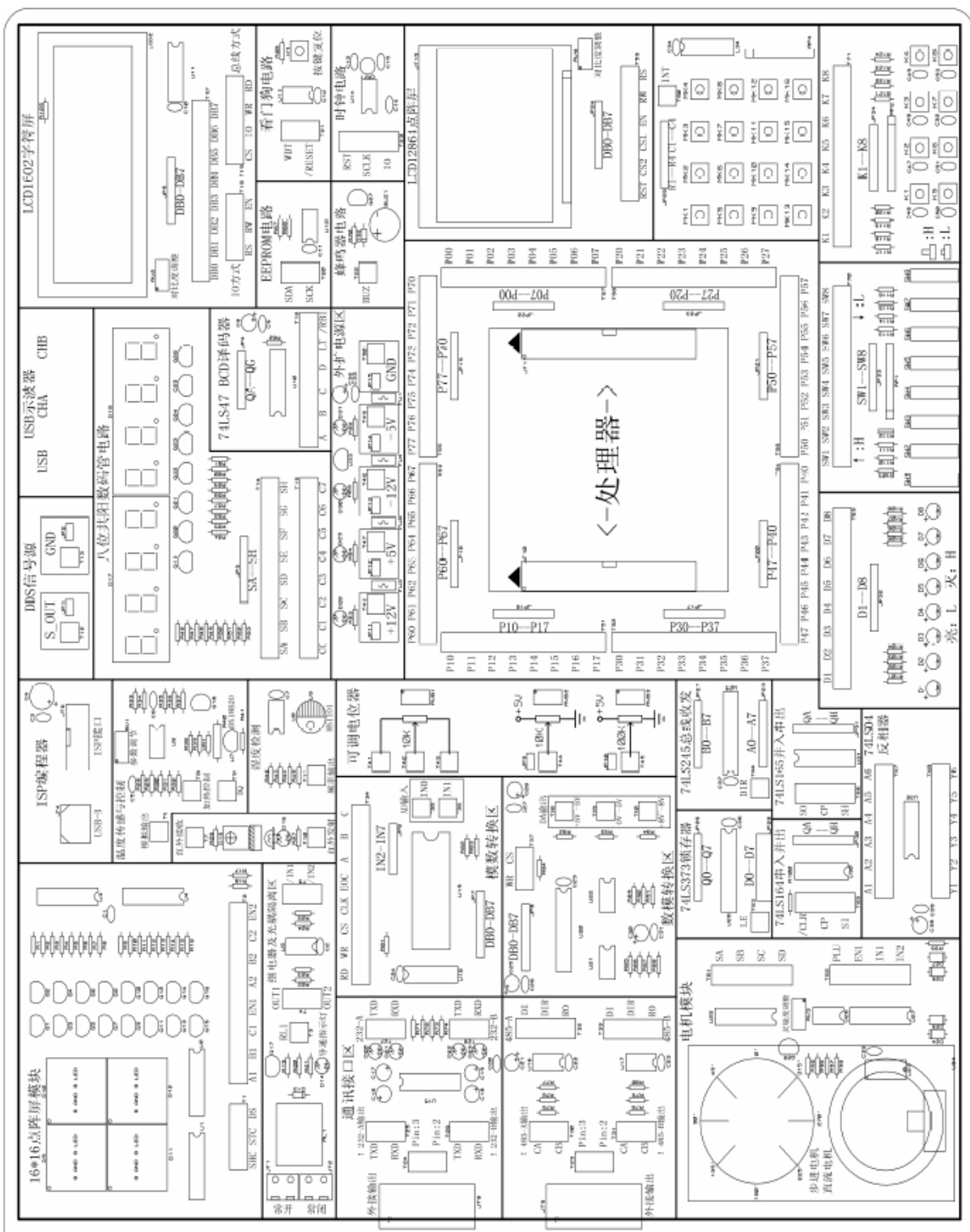


图 1-1 实验箱布局图

1.2 AT89S52 单片机核心板

(1) AT89S52 单片机最小系统

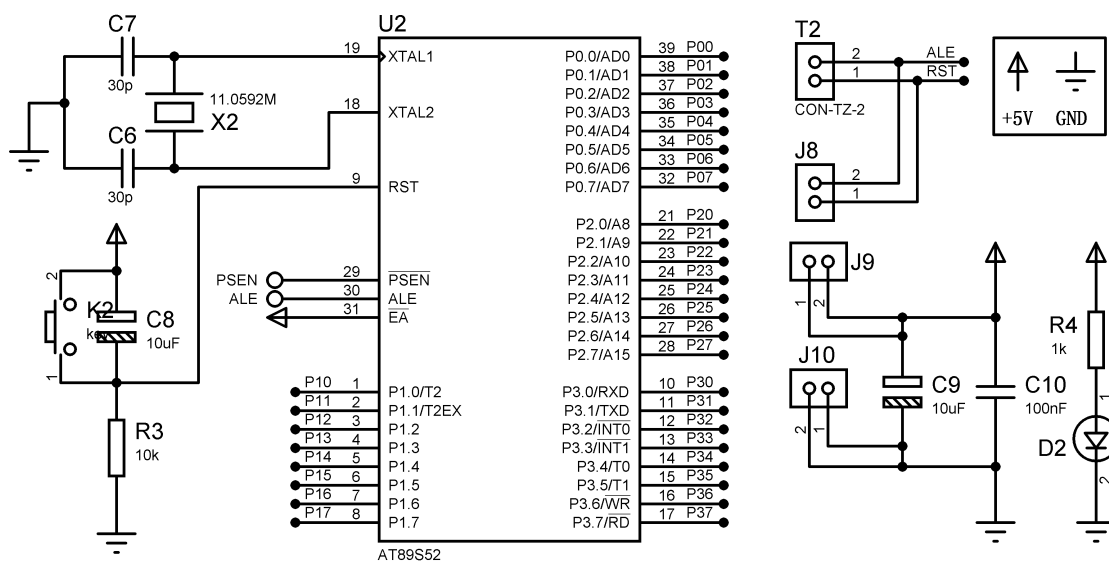


图 1-2 AT89S52 单片机最小系统

(2) AT89S52 单片机接口

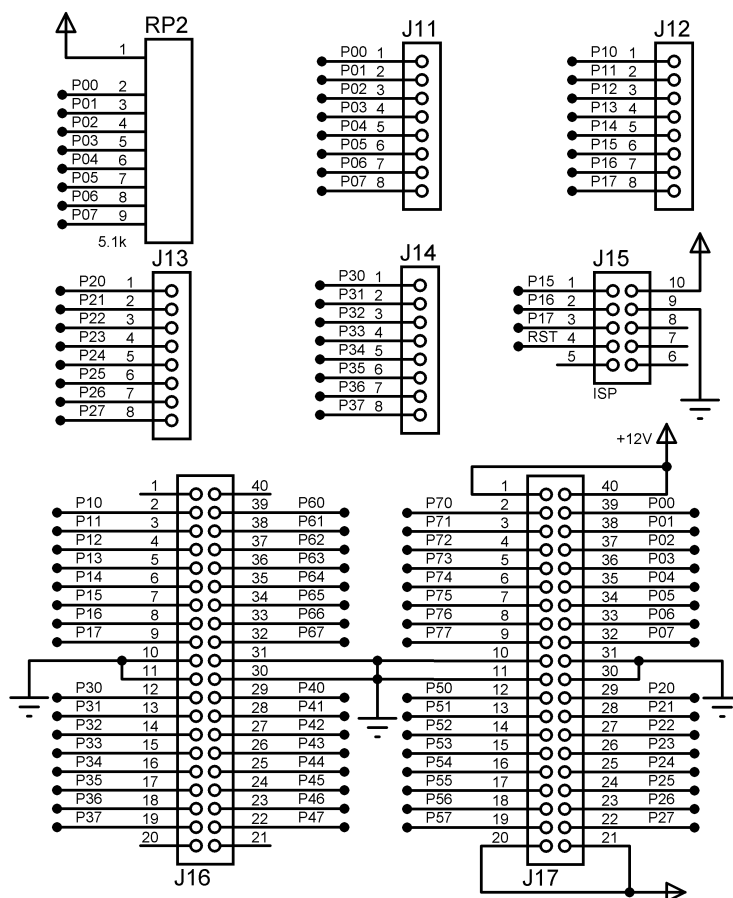


图 1-3 AT89S52 单片机接口

第二章 实验教学方式及软、硬件使用简介

2.1 实验教学方式

该课程实验采用软件仿真调试+硬件实操的方式进行教学。即：先通过 PROTEUS 软件对设计进行仿真和调试，再将同样的 HEX 文件下载到硬件实验系统的处理器中运行，最后查看实验结果，并进行比较，如图 2-1 所示。

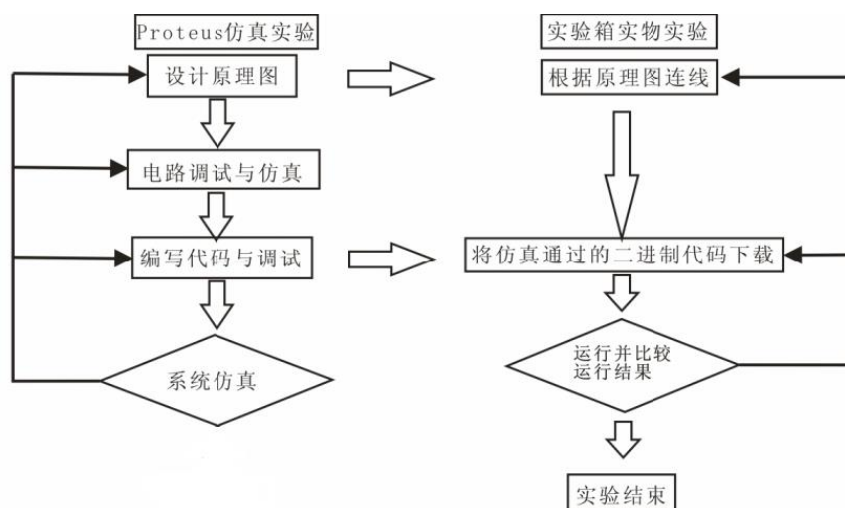


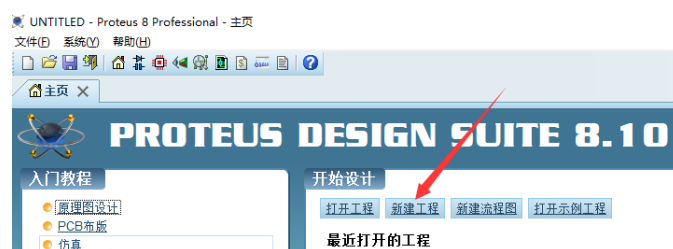
图 2-1 实验教学方式

其中，PROTEUS 软件可对设计进行仿真和调试；程序的编写可使用汇编语言或 C 语言，通过 Keil 平台实现，并生成 HEX 文件；HEX 文件可通过 Plink 下载器下载至硬件平台。

2.2 Proteus 仿真实验步骤

（1）新建原理图

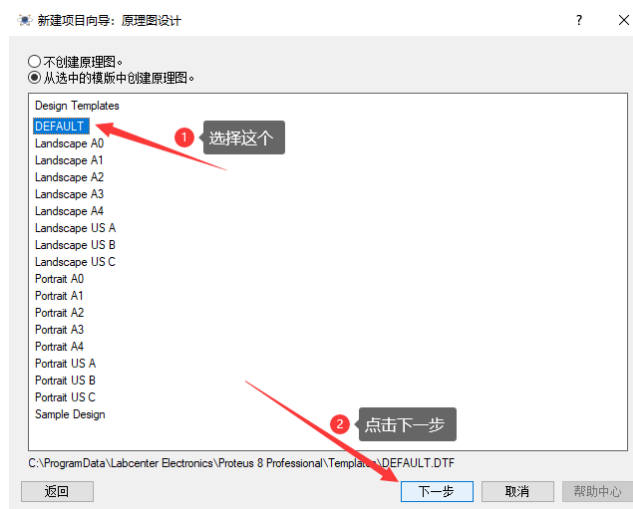
1) 打开 Proteus 8，点击新建工程；



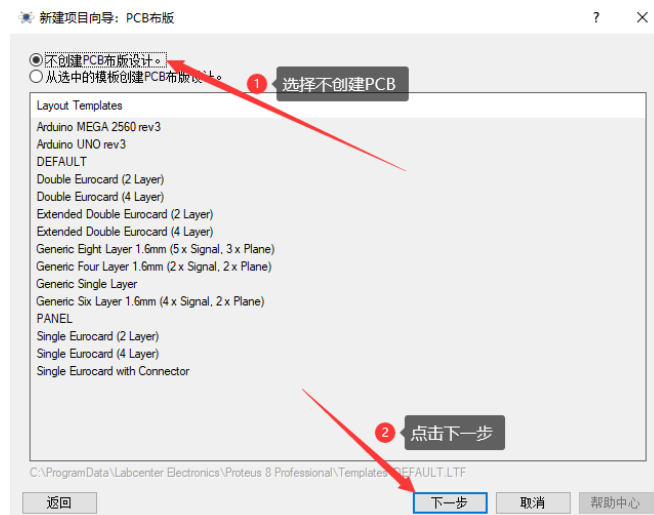
2) 工程命名，最好用英文或数字命名；



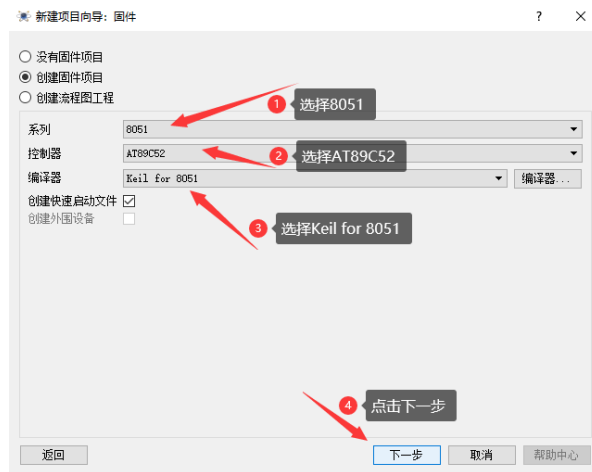
3) 创建原理图，选择 DEFAULT，默认模板；



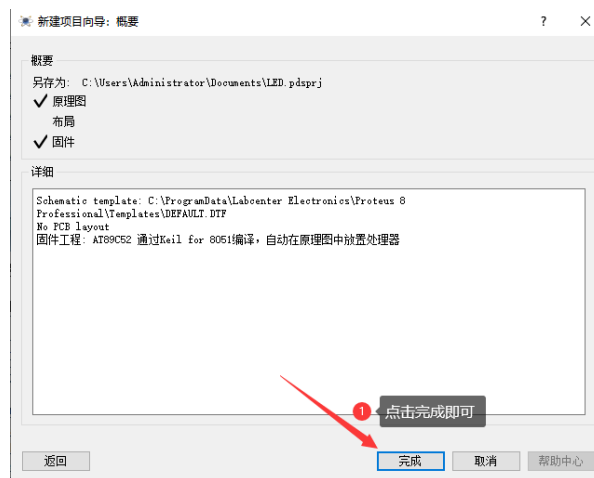
4) 选择不创建 PCB 设计；



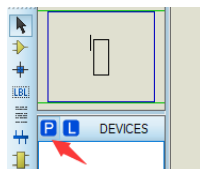
5) 创建固件项目，系列选择 8051，控制器选择 AT89C52，编译器选择 Keil for 8051，需要提前安装好 keil 软件，才能出现此编译器选项；



6) 完成原理图创建；



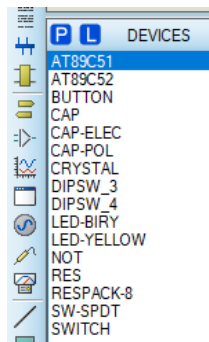
7) 在原理图设计界面，点击左侧蓝色的 P 图标进入元器件选取界面；



8) 在左上角输入所需元器件关键字，并在右侧列表双击选取所需器件，点击确定；



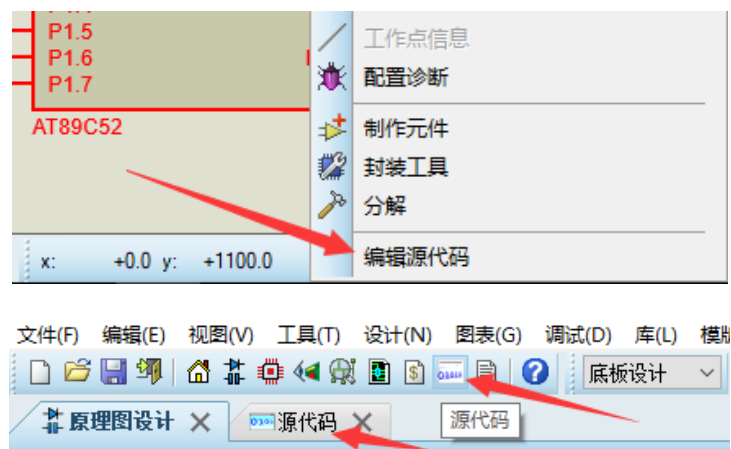
9) 选取完成后，左侧的器件栏窗口会有相应元器件列表；



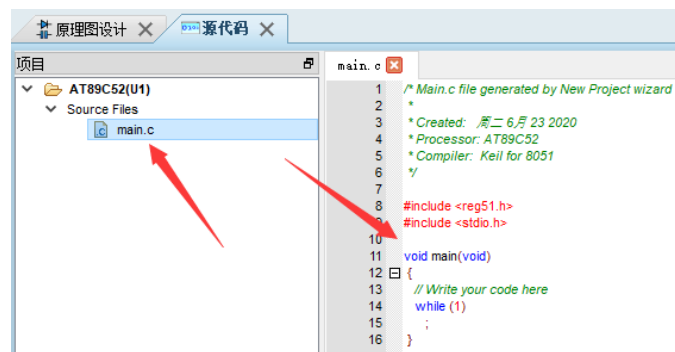
10) 参考原理图设计步骤完成器件的排布和连线。

(2) 新建程序

1) 在 51 芯片处单击右键，选择编辑源代码，也可以点击窗口切换或图标；

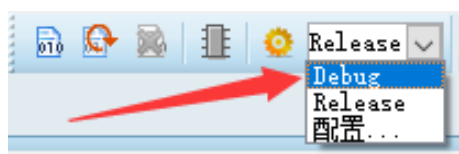


2) 根据程序流程图，在 main.c 文件中输入完整程序代码（若是.asm 文件，则需进行添加即可，并将.c 文件删除）；



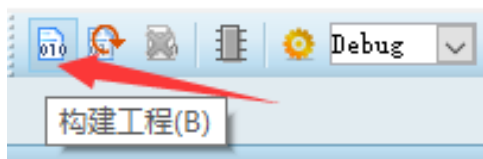
(3) Proteus 中编译和仿真

1) 在快捷工具栏中的下拉菜单 Debug /Release/配置选项中，选择 **Debug**；



2) 然后在工程中，右键单击 AT89C52，在出现的下拉列表中选择构建工程，或点

击工具栏构建工程图标（或快捷键 **Ctrl+F7**）进行构建工程，等待编译完成；



3) 编译成功后，切换到原理图状态下点击运行按钮（或快捷键 **F12**），可通过人机接口观察程序的仿真结果。

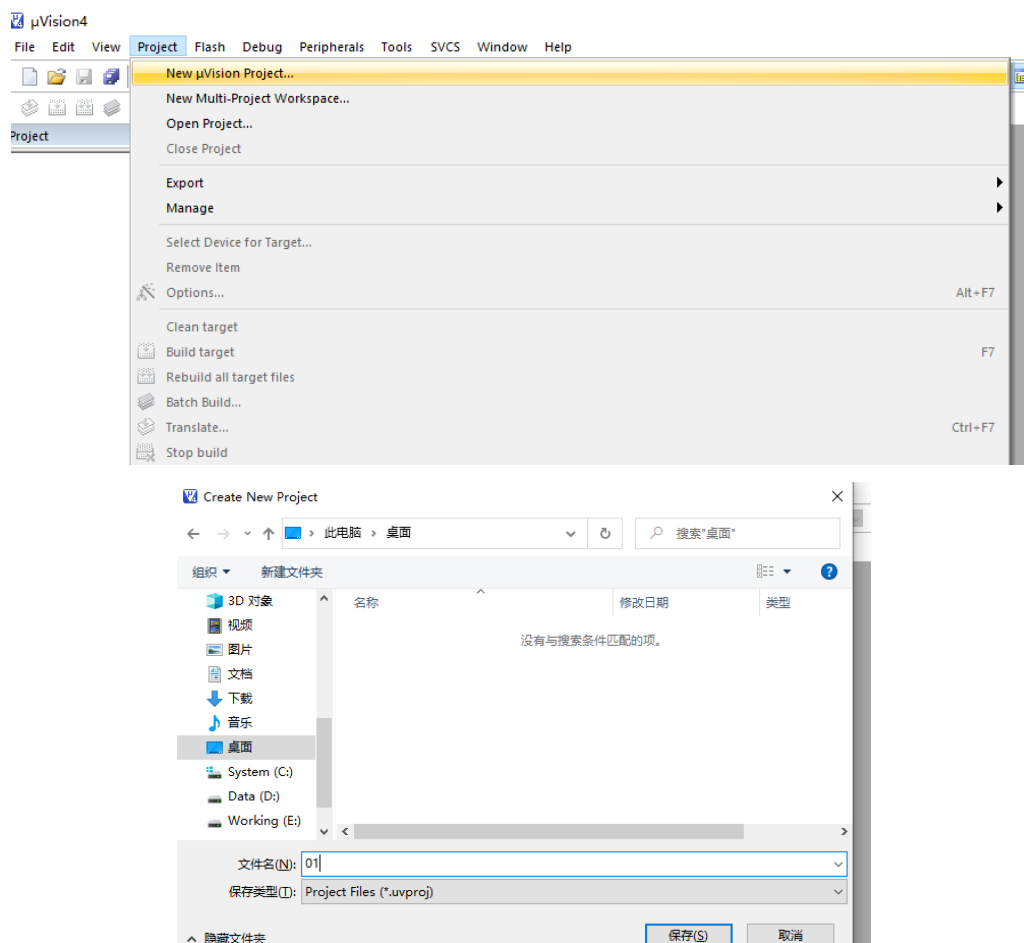
2.3 keil 编程步骤

（1）双击打开 keil 编程软件，图标如下图所示

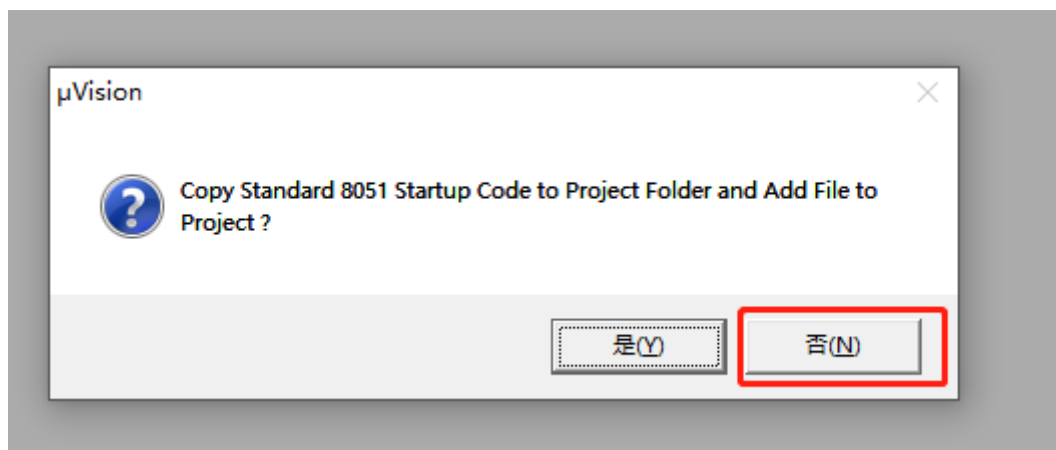
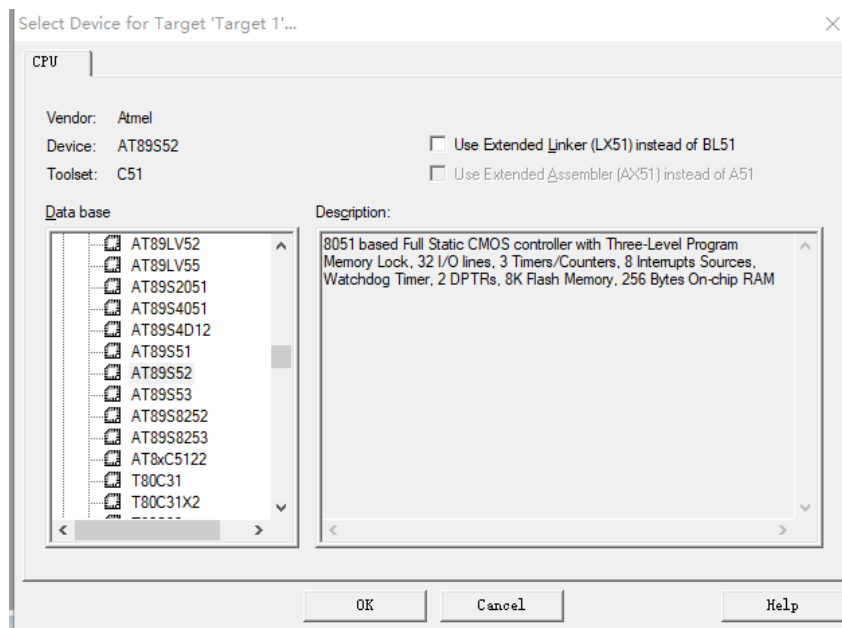


（2）新建工程

在“菜单栏” — “Project” — “New μ vision+Project” — 输入要保存的工程名。



然后选择对应 CPU 型号。此处选择 Atmel—AT89S52，点“确定”。



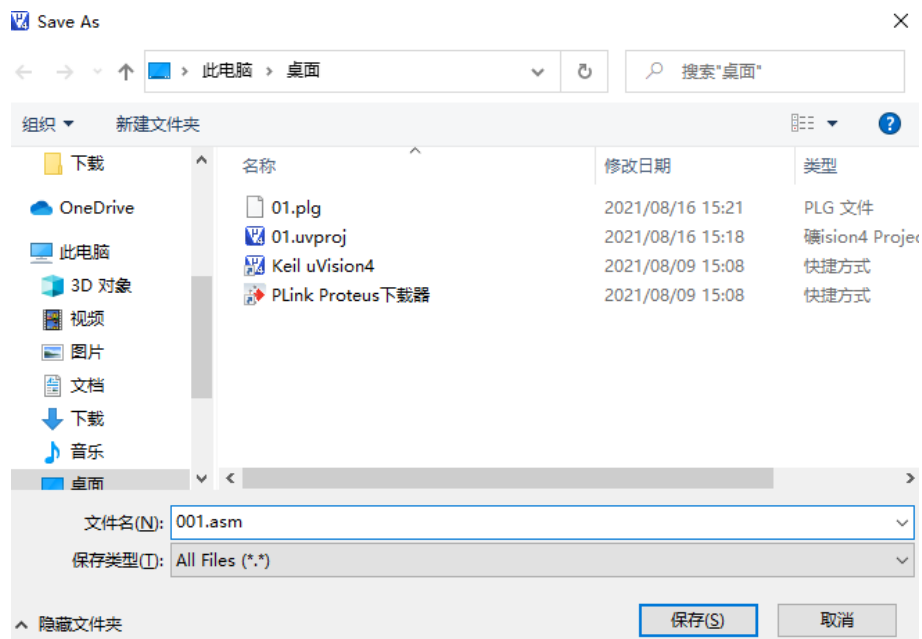
(3) 新建文件

点击新建文件图标



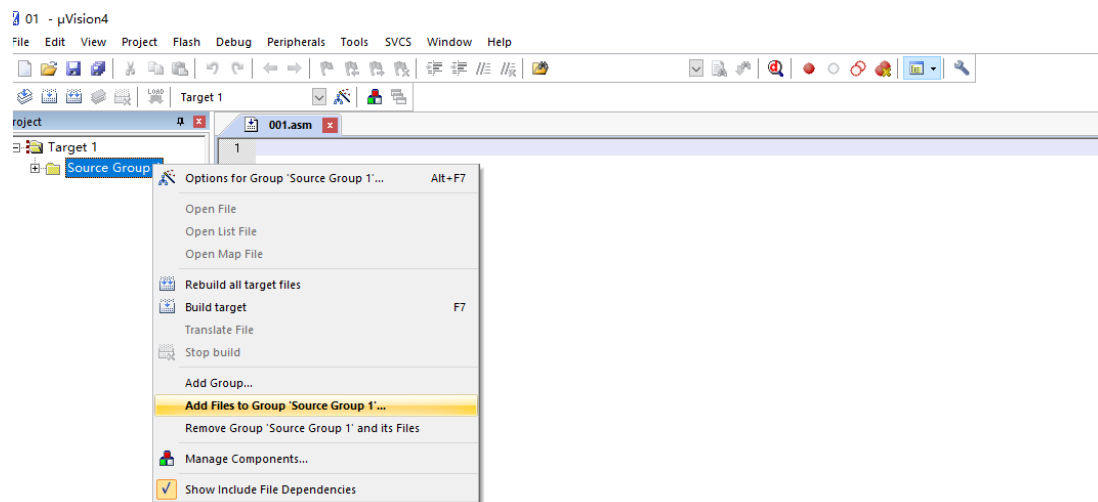
(4) 保存文件

点击“File” — “Save as”。若是用 C 语言编写的程序，扩展名为“.c”；若用汇编语言编写的程序扩展名为“.asm”。点击保存。

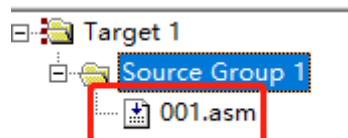


(5) 添加文件到工程

在工程管理器中，用鼠标右键点击“Source Group 1”，选择“Add File to Group Source Group 1”，如果要添加的文件是汇编文件，文件类型选择如下图所示的类型；如果是 C 语言文件，文件类型选择 “*.c”。双击要添加文件，然后点击”close”。



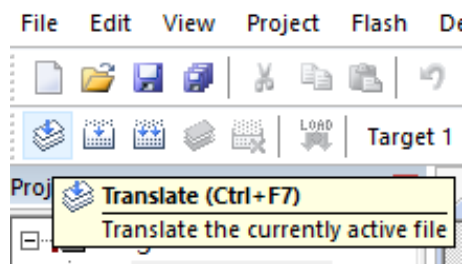
在工程管理器中，可以看到刚刚添加的文件



(6) 编写程序

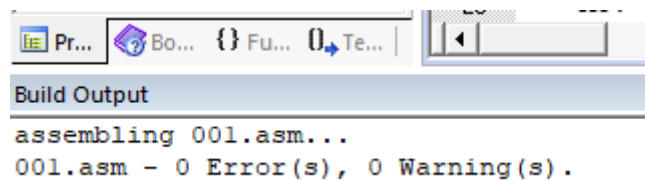
(7) 编译当前文件

点击编译按钮



(8) 修改错误

根据输出框里的提示信息修改程序中错误。

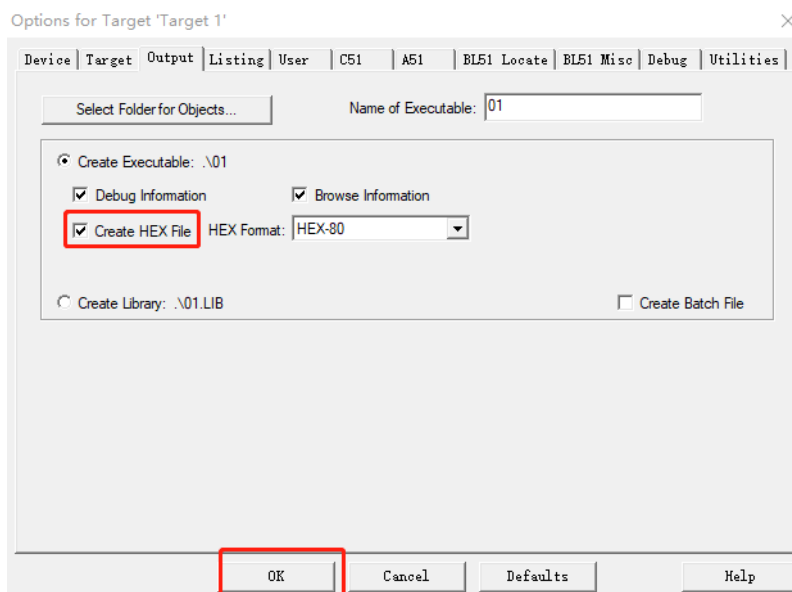


(9) 生成烧录文件

在工程管理器中，鼠标右键单击，按下图操作。



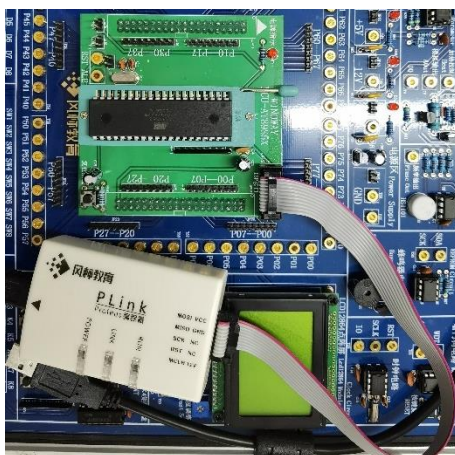
选择“Output”选项，将“create HEX file”前复选框前选上。“01”是文件名，可以改成其他的名，文件名前“select folder for objects”是此文件要保存的位置，默认情况下和当前工程保存在同一目录下，如果要保存在其他位置，可以点击此外进行修改。



点击“Build”或按 F7，生成.hex 文件（注意：必须在编译当前文件没有错误的情况下才能生成.hex 文件）。

2.4 Plink 下载程序方法

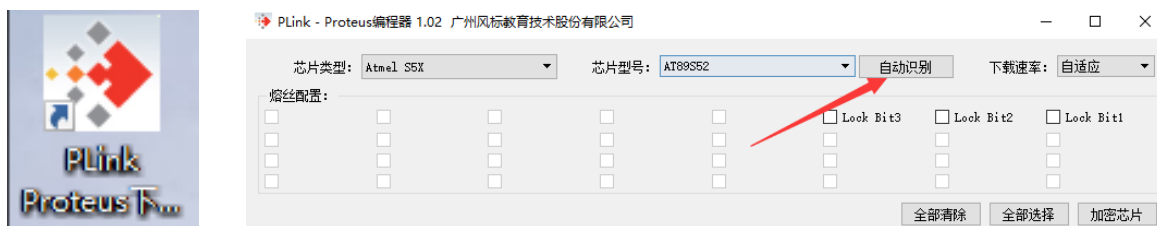
（1）连接



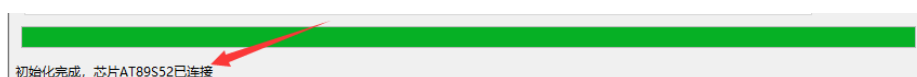
将 USB 下载器连接 PC 及目标板（USB 接口端连接到 PC 的 USB 接口，ISP 端连接到 MCU 模块 ISP 插槽）。在连接 USB 下载器之前，注意：MCU 模块或者主板应先接通电源，即实验箱通电。

（2）软件设置

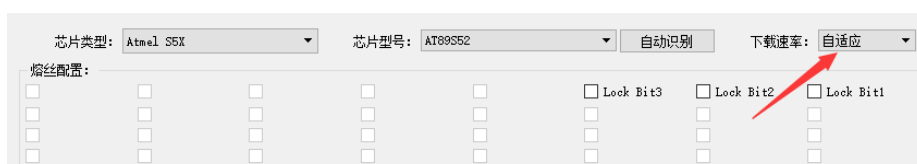
1) 打开 Plink 编程器软件，首先点击芯片型号旁的自动识别，检测芯片是否正常连接；



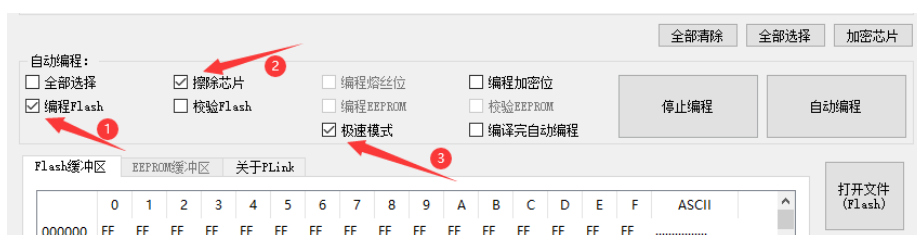
2) 识别成功后，底部出现芯片已连接提示；



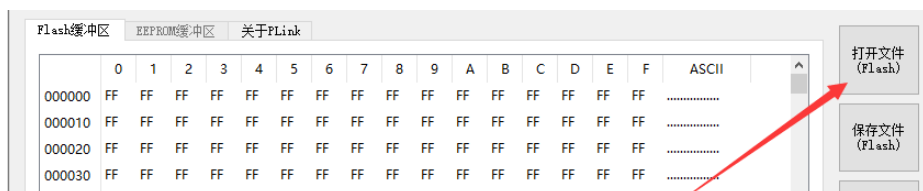
3) 点击右上角下载速率选择：自适应；



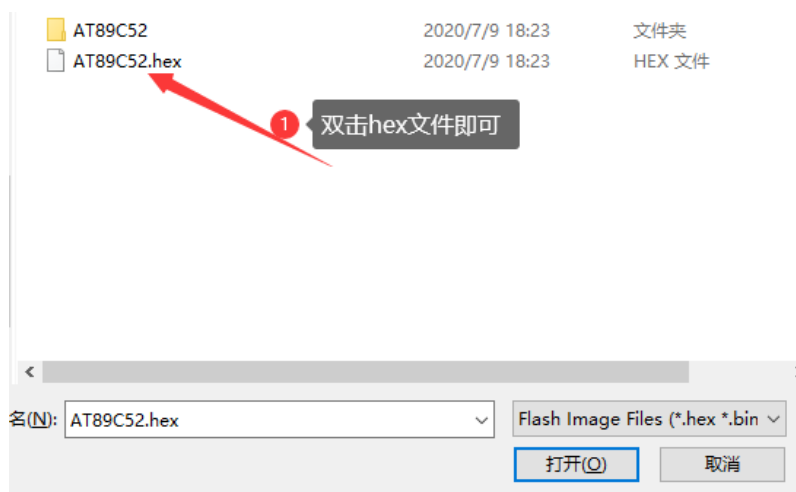
4) 勾选编程 Flash、擦除芯片、极速模式；



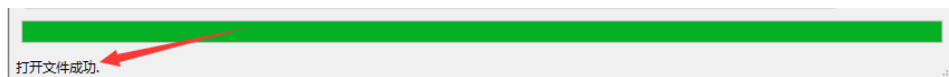
5) 点击打开文件；



6) 打开生成 hex 文件的文件夹，双击选择所需的 hex 文件；



7) 打开后底部显示打开文件成功;



8) 点击自动编程, 等待程序下载即可;



9) 提示写 Flash 完成, 表示程序下载成功;



第三章 实验练习

3.1 基础实验一 I/O 口实验

一、实验目的

- 1、熟悉单片机的最小系统，了解单片机 I/O 的结构；
- 2、掌握按键值的读入和处理；
- 3、学习简单的单片机程序的编写；
- 4、熟悉 Proteus 仿真软件的使用；
- 5、熟悉单片机实验箱的实验操作。

二、实验设备和器件

PC 机一台，PROTEUS 仿真软件一套，实验箱一台，ISP 下载器一台。

三、实验要求

- 1、利用单片机，按键和发光二极管，构成一个 LED 灯控制电路；
- 2、点亮 LED。按下 K1 时，LED 由高向低移一位，按下 K2 时，LED 由低向高移一位。

四、硬件连接图

1、硬件电路

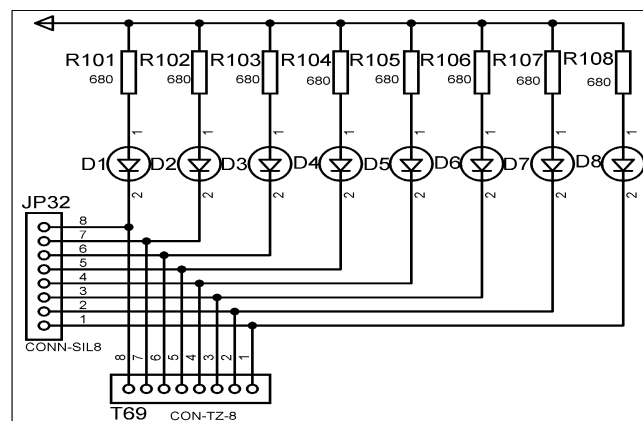


图 3-1 8 位独立 LED

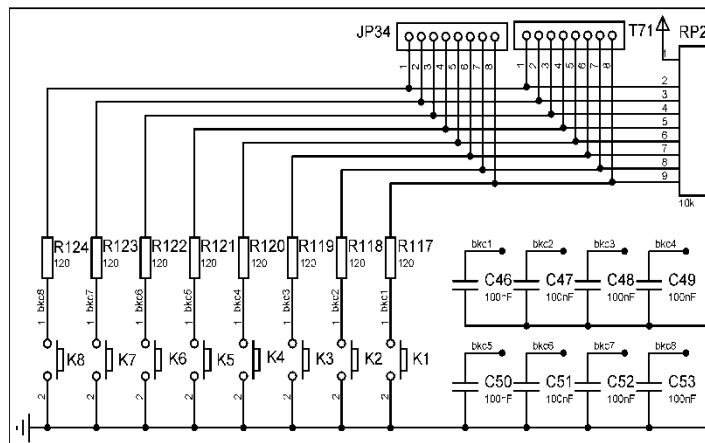


图 3-2 8 位独立按键

2、硬件连接

用连接线将 P10 与 K1 相连、P11 与 K2 相连、P00-P07 依次与 D1-D8 相连。

五、Proteus 原理图设计

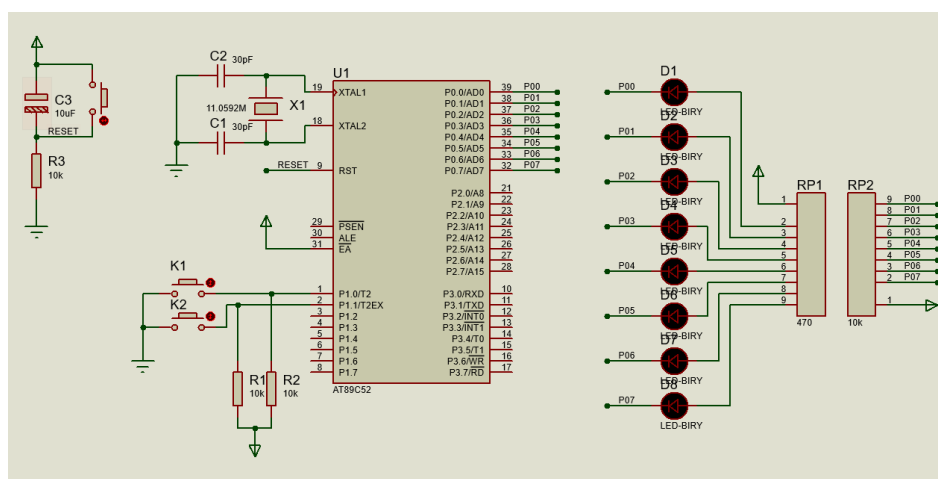


图 3-3 总原理图

步骤 1.在 Proteus 中绘制单片机最小系统，包括主控芯片、晶振电路和复位电路。

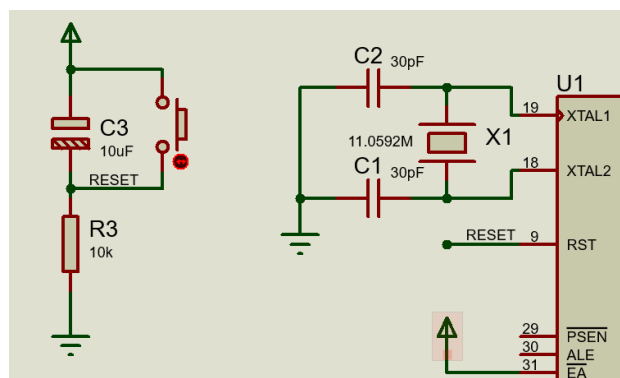


图 3-4 步骤 1 完成图

步骤 2.添加**按键**，分别连接到 **P1.0** 和 **P1.1** 口，输入端接地，连接 IO 口端添加电阻，上拉。

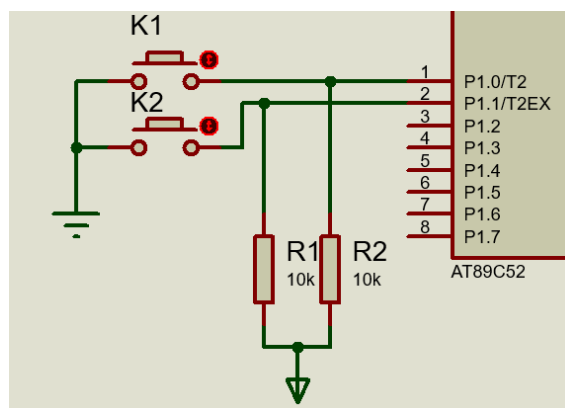


图 3-5 步骤 2 完成图

步骤 3. 添加 **8 个 LED 灯**，并添加**限流排阻**，**共阳连接**，利用标号连接到 **P0** 口。RP1 是**限流**电阻，保护 LED，避免电流过高，烧坏 LED；RP2 是**上拉**电阻，将 P0 口的电平拉高。

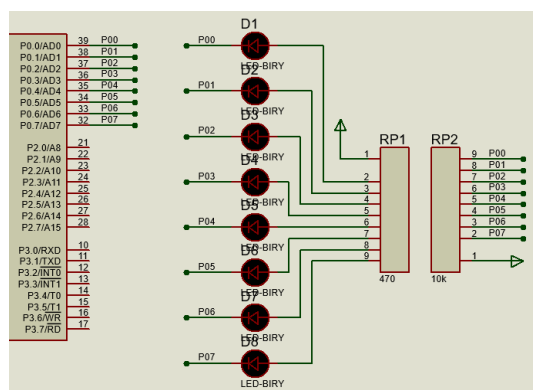


图 3-6 步骤 3 完成图

表 3-1 Proteus 使用的元器件

元器件名称	规格	说明
AT89C51		51 单片机
BUTTON		按键, 用于最小系统复位；实现输入功能
CAP	30pF	电容，用于搭建复位电路
CAP-ELEC	10uF	电解电容，用于搭建复位电路
CRYSTAL	11.0592M	晶振, 给单片机提供时钟信号
LED-YELLOW	8	黄色 LED 灯
RES	10k	电阻
RESPACK-8	470 Ω 、10k	排阻；RP1 是限流电阻，RP2 是上拉电阻

六、程序流程图及程序

1. 程序流程图

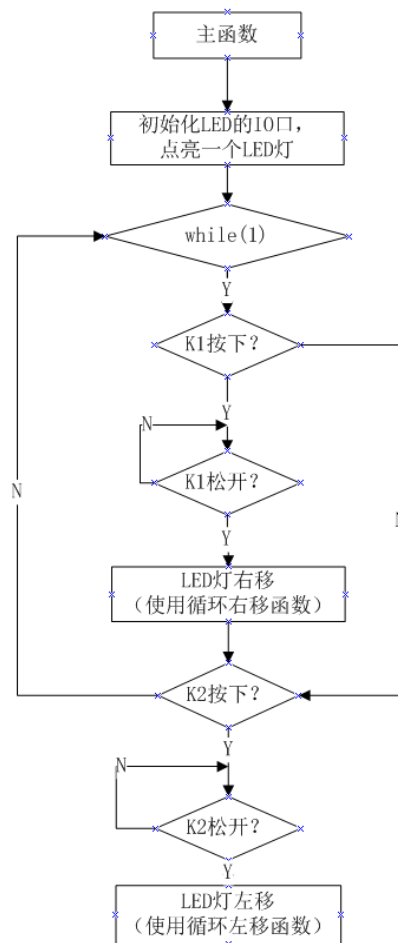


图 3-7 程序流程图

2. 参考程序

```
ORG 00H
AJMP START
ORG 0100H
START:  MOV     A,#0FEH
        MOV     P0,A      ;初始化 P0 口
MAIN:   JNB     P1.0,K1    ;判断 K1 是否按下
        JNB     P1.1,K2    ;判断 K2 是否按下
        SJMP    MAIN
K1:     LCALL   DELAYMS    ;延时消抖
        JNB     P1.0,K11   ;再次判断
        SJMP    MAIN
K11:    RR      A
```

```

        MOV    P0,A
        LCALL  DELAY
        SJMPMAIN

K2:      LCALL  DELAYMS ;延时消抖
        JNB P1.1,K22    ;再次判断
        SJMPMAIN

K22:     RL    A
        MOV    P0,A
        LCALL  DELAY
        SJMP   MAIN

DELAYMS:
        MOV R3,#60      ; 30MS
D0:      MOV R4,#248
        DJNZ R4,$
        DJNZ R3,D0
        RET

DELAY:   MOV    R5,#20           ;延时程序，延时 0.2s
D1:      MOV    R6,#20
D2:      MOV    R7,#248
        DJNZ    R7,$
        DJNZ    R6,D2
        DJNZ    R5,D1
        RET
        END

```

七、实验报告要求

1. 将程序及每一句加注释写在实验报告上。
2. 分析调试过程中遇到的问题和解决的办法。
3. 将实验步骤中观察的现象记录下来。

3.2 基础实验二 流水灯实验

一、实验目的

- 1、了解 I/O 口的电气特性和驱动能力；
- 2、了解 LED 电路中加入限流电阻的原因；
- 3、掌握程序编写的方法；
- 4、熟练 Proteus 仿真软件的使用；
- 5、熟练单片机实验箱的实验操作。

二、实验设备和器件

PC 机一台，PROTEUS 仿真软件一套，实验箱一台，ISP 下载器一台。

三、实验要求

用单片机及 8 个发光二极管等器件，构成一个单片机控制的流水灯系统。即：在实验一基础上，使其可以连续的向右/左流动。

四、硬件连接图

1、硬件电路

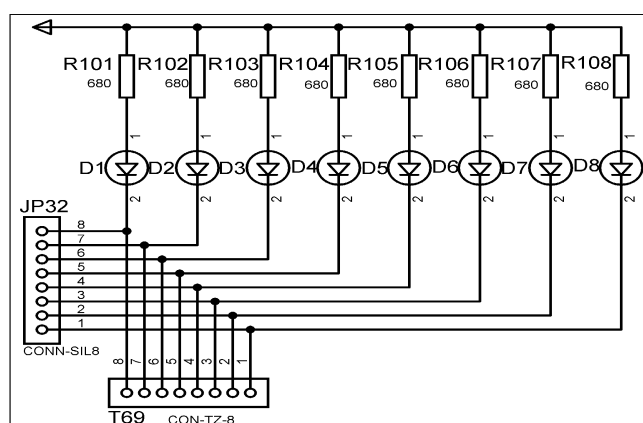


图 3-8 8 位独立 LED

2、硬件连接

用连接线将 P00-P07 依次与 D1-D8 相连。

五、Proteus 原理图设计

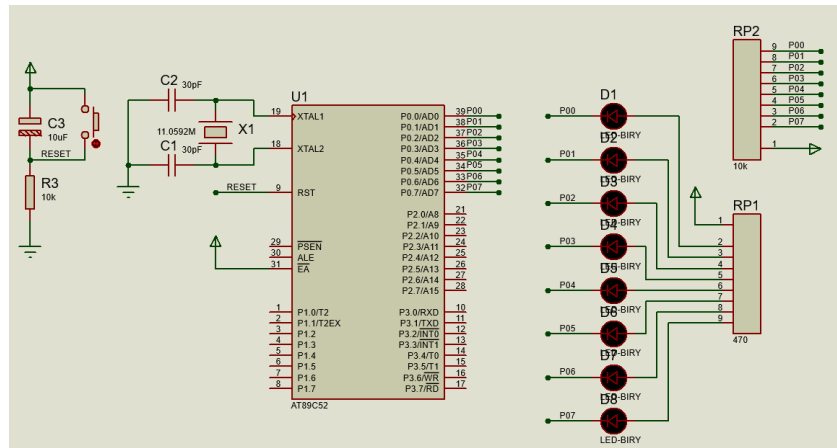


图 3-9 总原理图

步骤 1. 在 Proteus 中绘制单片机最小系统，包括主控芯片、晶振电路和复位电路。

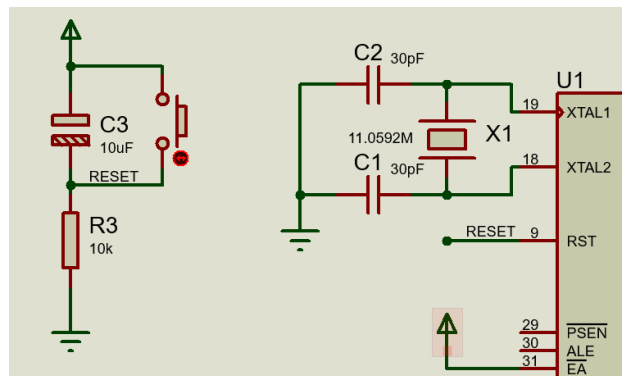


图 3-10 步骤 1 完成图

步骤 2. 添加 8 个 LED 灯，并添加限流排阻，共阳连接，利用标号连接到 P0 口。RP1 是限流电阻，保护 LED，避免电流过高烧坏 LED；RP2 是上拉电阻，将 P0 口的电平拉高。

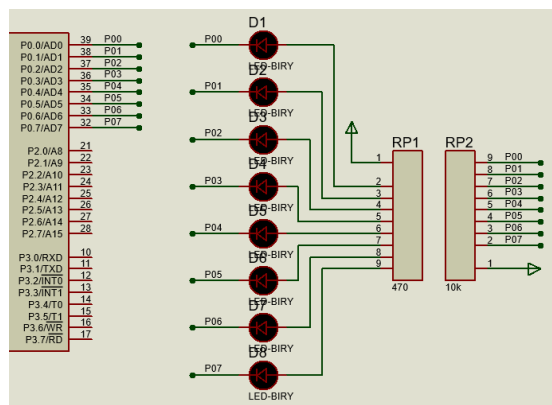


图 3-11 步骤 2 完成图

表 3-2 Proteus 使用的元器件

元器件名称	规格	说明
AT89C51		51 单片机
BUTTON		按键, 用于最小系统复位 ; 实现输入功能
CAP	30pF	电容, 用于搭建复位电路
CAP-ELEC	10uF	电解电容, 用于搭建复位电路
CRYSTAL	11.0592M	晶振, 给单片机提供时钟信号
LED-YELLOW	8	黄色 LED 灯
RES	10k	电阻
RESPACK-8	470 Ω 、 10k	排阻 ; RP1 是限流电阻, RP2 是上拉电阻

六、程序流程图

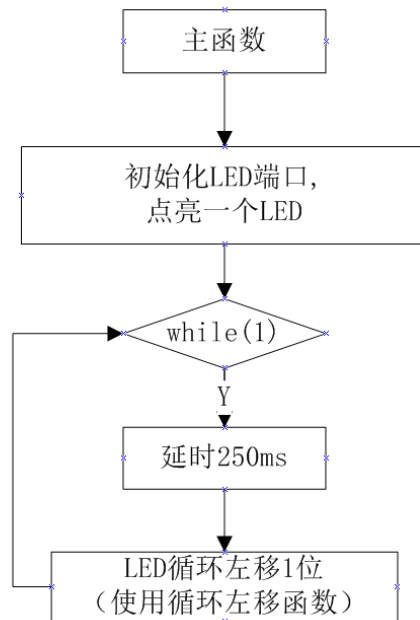


图 3-12 程序流程图

七、实验报告要求

1. 实现本实验功能的程序并将其每一句加注释。
2. 尝试改变 LED 流水灯的方向和快慢（写出延时程序的时间计算公式，并写出 100ms 延时程序）。

3.3 基础实验三 外部中断计数实验

一、实验目的

- 1、了解 I/O 口的电气特性和驱动能力；
- 2、了解 LED 电路中加入限流电阻的原因；
- 3、掌握程序编写的方法；
- 4、熟练 Proteus 仿真软件的使用；
- 5、熟练单片机实验箱的实验操作。

二、实验设备和器件

PC 机一台，PROTEUS 仿真软件一套，实验箱一台，ISP 下载器一台。

三、实验要求

在单片机接一个按键作为外部的中断输入信号，通过数码管显示中断次数

四、硬件连接图

1、硬件电路

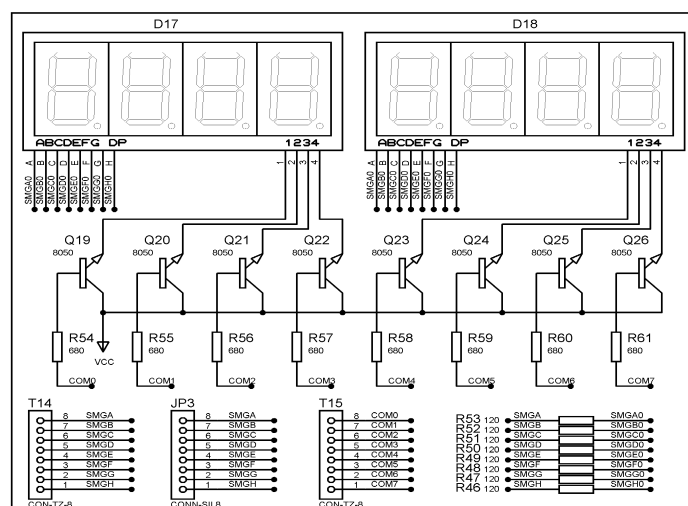


图 3-13 8 位共阳数码管

2、硬件连接

用连接线将 P00-P07 依次与 8 位共阳数码管 SA~SH 相连，P10~P13 依次与 C0~C3

五、Proteus 原理图设计

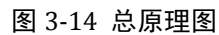


图 3-15 步骤 1 完成图

图 3-16 步骤 2 完成图

步骤 3. 添加一个按键，外部输入端连接一个 $120\ \Omega$ 电阻接地，连接 IO 口端接 $10\text{k}\Omega$

上拉电阻。

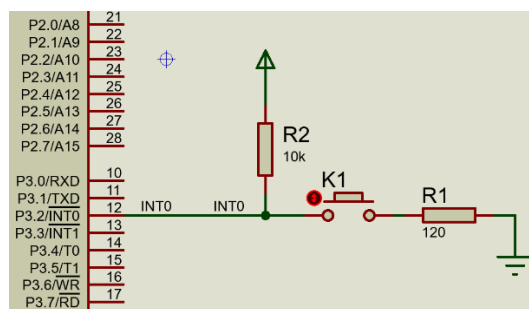


图 3-17 步骤 3 完成图

表 3-3 Proteus 使用的元器件

元器件名称	规格	说明
AT89C51		51 单片机
BUTTON		按键, 用于最小系统复位 ; 实现输入功能
CAP	30pF	电容, 用于搭建复位电路
CAP-ELEC	10uF	电解电容, 用于搭建复位电路
CRYSTAL	11.0592M	晶振, 给单片机提供时钟信号
7SEG-MPX4-CA		4 位共阳数码管
RES	10k、120	电阻
RESPACK-8	10k	排阻 ; RP2 是上拉电阻

六、程序流程图及参考程序

1. 程序流程图

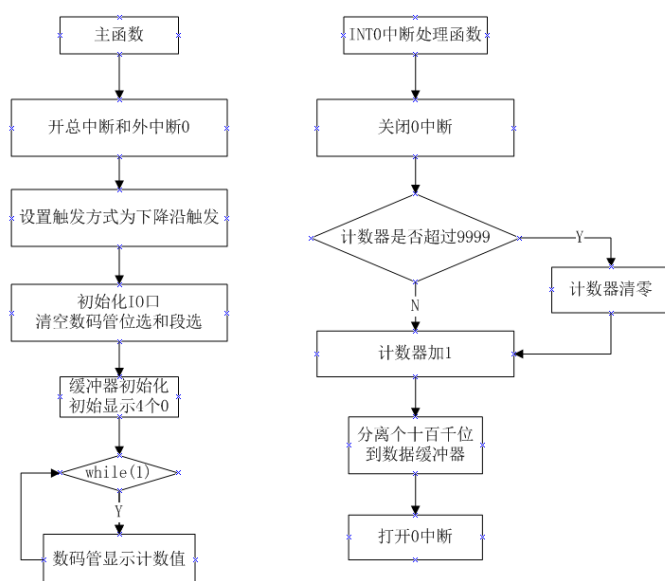


图 3-18 程序流程图

2. 参考程序

```
;*****功能说明*****  
;用 AT89S52 的 MCU 外部中断 0 记录中断次数并显示  
;连线方式为: P0 口连接数码管段选 SA-SH, P32 连接按键作为中断输入  
;程序启动时, 初始化显示为 0000 , 有按键按下进入中断, 计数加一  
;*****  
  
ORG 0000H  
AJMP MAIN  
  
  
ORG 0003H  
RETI  
ORG 000BH  
RETI  
  
ORG 0003H ;外部中断 0 入口地址  
LJMP INTT0  
  
  
ORG 001BH  
RETI  
ORG 0023H  
RETI  
;-----  
ORG 0100H  
MAIN: ;*****主程序*****  
  
MOV P1,#0FFH  
LOOP: MOV 40H,#00H ;系统初始设置计数初值: 0000 (可自行修改)  
LOOP0: MOV 39H,#00H  
MOV 38H,#00H  
MOV 37H,#00H  
  
MOV 30H,#00H ;数码管显示缓存 千位  
MOV 29H,#00H ;数码管显示缓存 百位  
MOV 28H,#00H ;数码管显示缓存 十位  
MOV 27H,#00H ;数码管显示缓存 个位  
  
MOV 30H,40H ;初值传递  
MOV 29H,39H  
MOV 28H,38H  
MOV 27H,37H
```

```

MOV    DPTR,#TAB
MOV    R1,#50H
MOV    R2,#50H
JJJ:   LCALL DISPLAY ;开始重复调用显示子程序,显示系统设置的初始值
        DJNZ R1,JJJ   ;多约 4 秒钟
;      DJNZ R2,JJJ

MOV    30H,#00H
MOV    29H,#00H
MOV    28H,#00H
MOV    27H,#00H
SETB   EA ;开启 CPU 中断
SETB   IT0 ;边沿触发
SETB   EX0 ;开启外部中断 0

KKK:   LCALL DISPLAY ;循环调用显示子程序
        AJMP KKK

;*****
;*****外部中断 1 子程序*****
;*****
INTT0:
        INC    27H      ;累加程序
        MOV    A,27H
        CJNE   A,#10,LOOP1 ;判断个位是否为 10，不相等转移
        MOV    27H,#00H ;向十位进一，个位归零

        INC    28H
        MOV    A,28H
        CJNE   A,#10,LOOP1
        MOV    28H,#00H

        INC    29H
        MOV    A,29H
        CJNE   A,#10,LOOP1
        MOV    29H,#00H

        INC    30H
        MOV    A,30H
        MOV    30H,#00H

LOOP1:  MOV    A,27H      ;比较判断程序

```

```
CLR    C           ;清除进位标志
SUBB   A,37H       ;将累加器 A 中的数据 and 计数初值比较
JNZ    LOOP2       ;累加器为 1 则转移
```

```
MOV    A,28H
CLR    C
SUBB   A,38H
JNZ    LOOP2
```

```
MOV    A,29H
CLR    C
SUBB   A,39H
JNZ    LOOP2
```

```
MOV    A,30H
CLR    C
SUBB   A,40H
JNZ    LOOP2
```

```
MOV    P0,#0FFH    ;关闭数码管的显示输出
MOV    30H,#00H    ;再给计数缓存置 0
MOV    29H,#00H
MOV    28H,#00H
MOV    27H,#00H
MOV    P1,#0FFH
```

```
LOOP2:    RETI
```

```
,*****
,*****显示子程序*****
,*****
```

```
DISPLAY:
```

```
;MOV    P1,#0FFH
```

```
MOV    A,30H
MOVC   A,@A+DPTR    ;代码字节传送到累加器
MOV    P0,A
SETB   P1.0
ACALL  DELAY
CLR    P1.0
```



```

MOV    A,29H
MOVC   A,@A+DPTR
MOV    P0,A
SETB   P1.1
ACALL  DELAY
CLR    P1.1

```

```

MOV    A,28H
MOVC   A,@A+DPTR
MOV    P0,A
SETB   P1.2
ACALL  DELAY
CLR    P1.2

```

```

MOV    A,27H
MOVC   A,@A+DPTR
MOV    P0,A
SETB   P1.3
ACALL  DELAY
CLR    P1.3
RET

```

```

;=====

```

```

TAB:    DB      0C0H;0
        DB      0F9H;1
        DB      0A4H;2
        DB      0B0H;3
        DB      099H;4
        DB      092H;5
        DB      082H;6
        DB      0F8H;7
        DB      080H;8
        DB      090H;9

```

```

;=====

```

```

DELAY:    MOV    R6,#10H    ;动态扫描延时子程序,这个时间能短就短,不然数码管会有闪烁.
DEL1:     MOV    R7,#10H    ;一般控制在 3mS 以内
          DJNZ   R7,$
          DJNZ   R6,DEL1
          RET

```

```

;=====

```

```

DELAY1:    MOV    R4,#14H    ;输出控制延时子程序
DEL2:     MOV    R3,#0C8H
DEL3:     MOV    R2,#0FAH

```

```
DJNZ R2,$  
DJNZ R3,DEL3  
DJNZ R4,DEL2  
RET
```

END

七、实验报告要求

1. 将程序及每一句加注释写在实验报告上。
2. 分析调试过程中遇到的问题和解决的办法。
3. 将实验步骤中观察的现象记录下来。
4. 按键触发时可能导致按下无反应，如何避免这种情况发生。

3.4 基础实验四 PWM 蜂鸣器（定时器）实验

一、实验目的

- 1、掌握定时器的使用；
- 2、掌握蜂鸣器的发声原理；
- 3、熟悉 PWM 波形的产生；
- 4、熟练 Proteus 仿真软件的使用；
- 5、熟练单片机实验箱的实验操作。

二、实验设备和器件

PC 机一台，PROTEUS 仿真软件一套，实验箱一台，ISP 下载器一台。

三、实验要求

用定时器设计 PWM 波形驱动蜂鸣器发声, 播放音乐。

四、实验原理

51 单片机可以用内部定时器输出 PWM，可用两个定时器实现，也可以用一个定时器实现。本实验是使用定时器 0 来实现的。下图是工作方式控制寄存器（TMOD）各个位的功能。

Table 2-4. TMOD Register - TMOD (S: 89h)
TMOD - Timer/Counter 0 and 1 Modes

7	6	5	4	3	2	1	0
GATE1	C/T1#	M11	M01	GATE0	C/T0#	M10	M00
Bit Number	Bit Mnemonic	Description					
7	GATE1	Timer 1 Gating Control Bit Clear to enable timer 1 whenever the TR1 bit is set. Set to enable timer 1 only while the INT1# pin is high and TR1 bit is set.					
6	C/T1#	Timer 1 Counter/Timer Select Bit Clear for timer operation: timer 1 counts the divided-down system clock. Set for Counter operation: timer 1 counts negative transitions on external pin T1.					
5	M11	Timer 1 Mode Select Bits					
		M11	M01	Operating mode			
		0	0	Mode 0: 8-bit timer/counter (TH1) with 5-bit prescaler (TL1).			
4	M01	0	1	Mode 1: 16-bit timer/counter.			
		1	0	Mode 2: 8-bit auto-reload timer/counter (TL1). Reloaded from TH1 at overflow.			
		1	1	Mode 3: timer 1 halted. Retains count.			
3	GATE0	Timer 0 Gating Control Bit Clear to enable timer 0 whenever the TR0 bit is set. Set to enable timer/counter 0 only while the INT0# pin is high and the TR0 bit is set.					
2	C/T0#	Timer 0 Counter/Timer Select Bit Clear for timer operation: timer 0 counts the divided-down system clock. Set for counter operation: timer 0 counts negative transitions on external pin T0.					
1	M10	Timer 0 Mode Select Bit					
		M10	M00	Operating mode			
		0	0	Mode 0: 8-bit timer/counter (TH0) with 5-bit prescaler (TL0).			
		0	1	Mode 1: 16-bit timer/counter.			
		1	0	Mode 2: 8-bit auto-reload timer/counter (TL0). Reloaded from TH0 at overflow.			
0	M00	1	1	Mode 3: TL0 is an 8-bit timer/counter. TH0 is an 8-bit timer using timer 1's TR0 and TF0 bits.			

Reset Value = 0000 0000b

图 3-19 TMOD 功能位

GATE1：门控位。0：TCON 中的 TR1 启动定时；1：外部中断引脚 INT1 和 TR1 定时启动。

C/T1：定时或者计数器选择位。0：定时工作方式；1：计数工作方式。

GATE0：门控位。0：TCON 中的 TR0 启动定时；1：外部中断引脚 INT0 和 TR0 定时启动。

C/T0：定时或者计数器选择位。0：定时工作方式；1：计数工作方式。

M11/M01：T1 的工作方式选择位。

M10/M00：T0 的工作方式选择位。

表 3-4 T0/T1 工作方式选择

M10/11	M00/01	功能描述
0	0	方式 0：13 位定时器/计数器
0	1	方式 1：16 位定时器/计数器
1	0	方式 2：初值自动重新装入的 8 位定时/计数器
1	1	方式 3：仅适用于 T0，将其分为两个 8 位计数器。对 T1 停止计数

使用定时器 T0 产生 PWM 时，首先要确定 PWM 的周期 T 和占空比 D，确定了这些以后，就可以用定时器产生一个时间基准 t，比如定时器溢出 n 次的时间是 PWM 的高电平的时间，则 $D \cdot T = n \cdot t$ ，类似的可以求出 PWM 低电平时间需要多少个时间基准 n。再给 TH0、TL1 设定初值。

五、硬件连接图

1、硬件电路

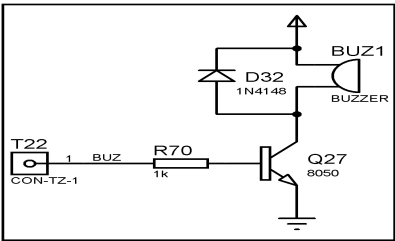


图 3-20 蜂鸣器连接

2、硬件连接

用连接线将 P00 与蜂鸣器 SOUNDER 相连。

六、Proteus 原理图设计

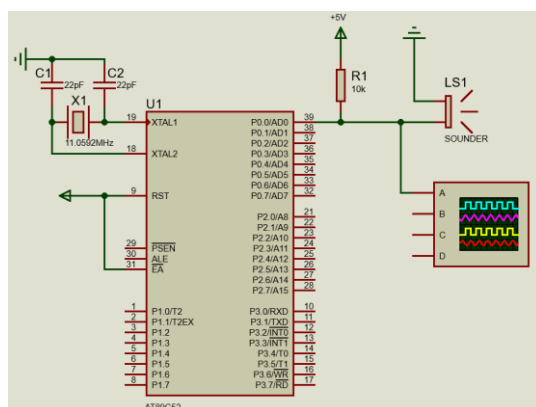


图 3-21 总原理图

步骤 1.在 Proteus 中绘制单片机最小系统，包括主控芯片、晶振电路和复位电路。

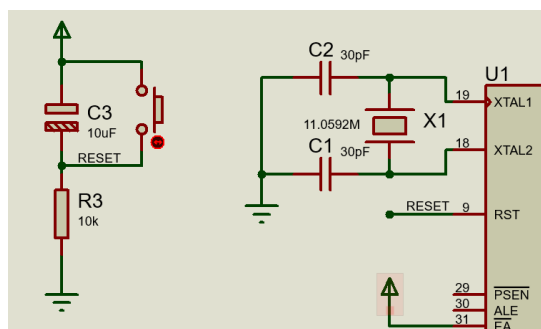


图 3-22 步骤 1 完成图

步骤 2. 添加蜂鸣器，一端接地，一端连接到 P0.0 口，P0 口需要上拉，同时该输出口连接一个示波器用来观察 PWM 波形。

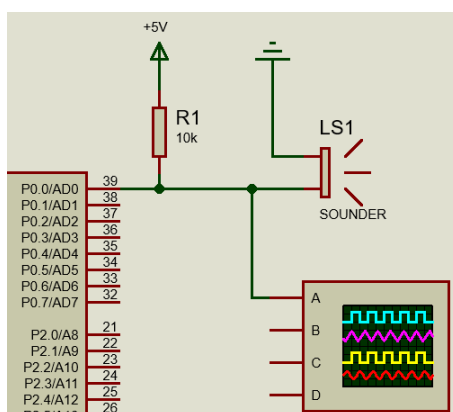


图 3-23 步骤 2 完成图

表 3-5 Proteus 使用的元器件

元器件名称	规格	说明
AT89C51		51 单片机
BUTTON		按键, 用于最小系统复位 ; 实现输入功能
CAP	30pF	电容, 用于搭建复位电路
CAP-ELEC	10uF	电解电容, 用于搭建复位电路
CRYSTAL	11. 0592M	晶振, 给单片机提供时钟信号
SOUNDER		蜂鸣器
RES	10k	电阻
RESPACK-8	10k	排阻 ; RP2 是上拉电阻

七、程序流程图及参考程序

1. 程序流程图

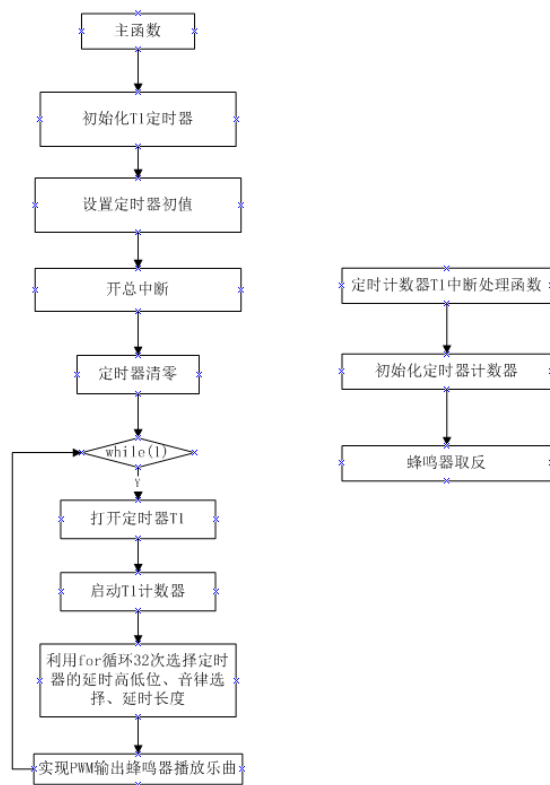


图 3-24 程序流程图

2. 参考程序

```
PP EQU P0.0
```

```
ORG 0000H
```

LJMP START

ORG 000BH

INC 20H ;中断服务,中断计数器加 1

MOV TH0,#0D8H

MOV TL0,#0EFH ;12M 晶振, 形成 10 毫秒中断

RETI

ORG 0100H

START:

MOV SP,#50H ;SP: 堆栈指针寄存器

MOV TH0,#0D8H

MOV TL0,#0EFH

MOV TMOD,#01H ;定时器 0 工作方式 1

MOV IE,#82H ;开总中断和定时器 0 中断

MUSIC0:

NOP

MOV DPTR,#DAT ;表头地址送 DPTR

MOV 20H,#00H ;中断计数器清 0

MUSIC1:

NOP

CLR A ;A;清零

MOVC A,@A+DPTR ;查表取代码

JZ END0 ;是 00H,则结束

CJNE A,#0FFH,MUSIC5 ;如果是休止符, 往下执行

LJMP MUSIC3

MUSIC5:

NOP

MOV R6,A ;R6=18H 音符的频率

INC DPTR ;DPTR;加 1

MOV A,#0

MOVC A,@A+DPTR ;取节拍代码送 R7

MOV R7,A ;R7=30H 音符发音的时间

SETB TR0 ;启动计数

MUSIC2:

NOP

CPL PP ;音乐输出口控制

```
MOV A,R6
MOV R3,A          ;R3=R6=18H
LCALL DEL
MOV A,R7
CJNE A,20H,MUSIC2 ;中断计数器(20H)=R7 否?
```

;不等,则继续循环

```
MOV 20H,#00H      ;等于,则取下一代码
INC DPTR
LJMP MUSIC1
```

```
MUSIC3:
NOP               ;休止 100 毫秒
CLR TR0
MOV R2,#0DH       ;R2=13
```

```
MUSIC4:
NOP
MOV R3,#0FFH      ;R3=255
LCALL DEL
DJNZ R2,MUSIC4
INC DPTR
LJMP MUSIC1
```

```
END0:
NOP
MOV R2,#0FFH      ;歌曲结束,延时 1 秒后继续
```

```
MUSIC6:
MOV R3,#00H
LCALL DEL
DJNZ R2,MUSIC6
LJMP MUSIC0
```

```
DEL:
NOP
```

```
DEL3:
MOV R4,#02H
```

```
DEL4:
NOP
```



```

DJNZ R4,DEL4
NOP
DJNZ R3,DEL3
RET
NOP
;音律编码，作用于定时时间
DAT:
DB 30H,20H,2BH,10H,26H,10H
DB 26H,20H,26H,20H,2BH,20H,30H,10H,2BH,10H,2BH,60H,30H,10H,2BH,10H
DB 26H,20H,26H,20H,2BH,20H,30H,20H,30H,60H,40H,10H,39H,10H
DB 40H,40H,40H,10H,40H,10H,39H,10H,30H,10H,30H,60H,30H,20H
DB 30H,20H,30H,20H,30H,10H,2BH,10H,2BH,40H,30H,20H,2BH,10H,26H,10H
DB 26H,20H,26H,20H,20H,20H,1CH,10H,2BH,10H,2BH,40H,26H,20H,2BH,10H,30H,10H
DB 30H,20H,30H,20H,2BH,20H,26H,10H,20H,10H,20H,60H,40H,10H,40H,10H
DB 39H,20H,30H,10H,30H,10H,30H,10H,26H,30H,2BH,20H,30H,20H,30H,20H,2BH,10H,30H,10H
DB 30H,0C0H,26H,20H,26H,10H,20H,10H
DB 20H,20H,20H,10H,20H,10H,20H,20H,20H,10H,1CH,10H,1CH,40H,26H,20H,2BH,20H
DB 30H,20H,30H,20H,30H,20H,2BH,10H,26H,10H,26H,60H,26H,10H,20H,10H
DB 20H,20H,20H,20H,20H,20H,1CH,10H,18H,10H,18H,20H,1CH,20H,20H,20H,26H,20H
DB 2BH,20H,30H,20H,30H,20H,39H,10H,2BH,10H,2BH,60H,26H,10H,20H,10H
DB 20H,20H,20H,20H,20H,10H,1CH,20H,20H,10H,1CH,40H,26H,20H,2BH,20H
DB 30H,20H,30H,20H,30H,20H,2BH,10H,26H,10H,26H,60H,26H,10H,20H,10H
DB 20H,20H,20H,20H,20H,20H,1CH,10H,18H,10H,18H,20H,1CH,20H,20H,20H,26H,10H,2BH,10H
DB 2BH,20H,2BH,20H,2BH,80H,30H,10H,26H,20H,24H,10H
DB 26H,20H,2BH,20H,2BH,80H,26H,20H,2BH,10H,30H,10H
DB 30H,40H,26H,20H,2BH,10H,30H,10H,30H,40H,26H,20H,2BH,20H
DB 26H,20H,30H,20H,26H,20H,1CH,10H,20H,10H,20H,60H,26H,10H,20H,10H
DB 1CH,20H,1CH,10H,20H,10H,20H,20H,1CH,10H,20H,10H,20H,40H,26H,10H,2BH,10H,30H,20H
DB 2BH,20H,2BH,10H,26H,10H,2BH,20H,30H,10H,2BH,10H,2BH,40H,26H,20H,2BH,10H,30H,10H
DB 30H,40H,26H,20H,2BH,10H,30H,10H,30H,40H,26H,20H,2BH,20H
DB 26H,20H,30H,20H,26H,10H,1CH,20H,20H,10H,20H,60H,26H,10H,20H,10H
DB 1CH,20H,1CH,20H,18H,20H,20H,10H,1CH,10H,20H,30H,20H,10H,26H,10H,2BH,10H,30H,20H
DB 2BH,40H,2BH,10H,30H,10H,2BH,10H,26H,10H,2BH,40H,26H,20H,26H,20H,2BH,60H,30H,20H,30H,0C0H
DB 00H
END

```

八、实验报告要求

1. 将程序读懂并将其每一句加注释。
2. 分析调试过程中遇到的问题和解决的办法。
3. 将实验步骤中观察的现象记录下来。

3.5 基础实验五 数码管驱动实验

一、实验目的

- 1、了解数码管显示原理；
- 2、掌握读表程序的编写；
- 3、熟练 Proteus 仿真软件的使用；
- 4、熟练单片机实验箱的实验操作。

二、实验设备和器件

PC 机一台，PROTEUS 仿真软件一套，实验箱一台，ISP 下载器一台。

三、实验要求

利用 I/O 口实现动态扫描数码管，数码管显示“12345678”。

四、实验原理

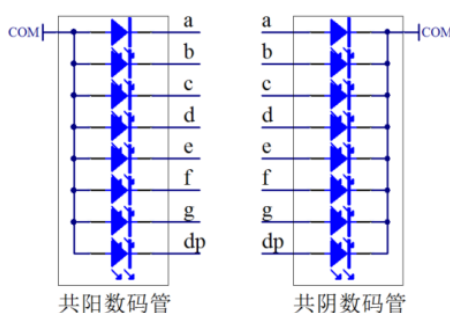


图 3-25 数码管图

数码管分为共阳和共阴两种，共阴数码管就是 8 只 LED 阴极连接在一起，阴极是公共端，由阳极来控制单个小灯的亮灭。同理，共阳数码管就是阳极接在一起。本实验使用的是 8 位共阳数码管。

数码管有两种显示方式：静态和动态。数码管静态显示是对应动态显示而言的，静态显示对于少量数码管（1-2 个）。而本实验是使用动态显示的方法。

多个数码管显示数字的时候，我们实际上是轮流点亮数码管（一个时刻内只有一个数码管是亮的），利用人眼的视觉暂留现象（也叫余辉效应），就可以做到看起来是所有

数码管都同时亮了，此为动态显示，也叫做动态扫描。只要刷新率大于 100Hz，即刷新时间小于 10ms，就可以做到无闪烁，这也就是动态扫描的硬性指标。有最小值的限制吗？理论上没有，但实际上做到更快的刷新却没有任何进步的意义了，因为已经无闪烁了，再快也还是无闪烁，只是徒然增加 CPU 的负荷而已（因为 1 秒内要执行更多次的扫描程序）。所以，通常我们设计程序的时候，取接近 10ms 的值即可。

五、硬件连接图

1、硬件电路

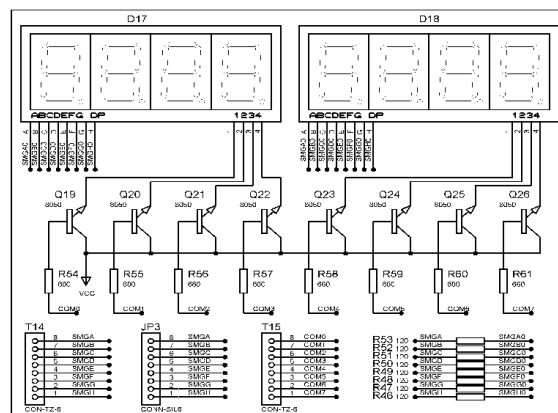


图 3-26 8 位数码管连接

2、硬件连接

用连接线将 P00-P07 依次与 8 位共阳数码管 SA-SH 相连；P20~P27 依次与 8 位共阳数码管 C0~C7 相连。

六、Proteus 原理图设计

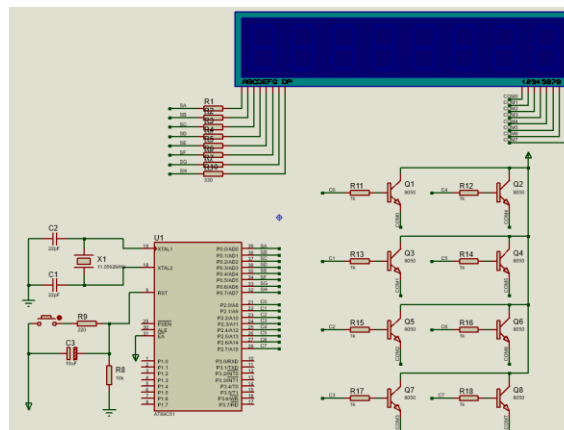


图 3-27 总原理图

步骤 1.在 Proteus 中绘制单片机最小系统，包括主控芯片、晶振电路和复位电路。

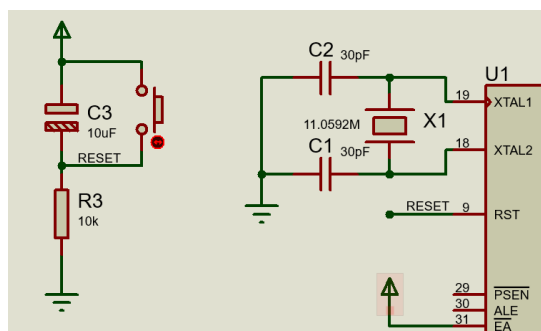


图 3-28 步骤 1 完成图

步骤 2. 添加 8 个 NPN 三极管，分别构成 8 个三极管开关选择电路，各基极连接一个 1k 电阻，通过标号分别连接到 P2 口，集电极共同连接高电平。

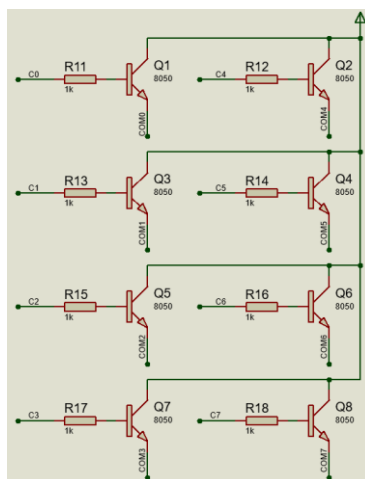


图 3-29 步骤 2 完成图

步骤 3. 添加 8 位共阳数码管，段选增加电阻后连接到 P0 口，位选分别连接到三极管开关电路的发射极。

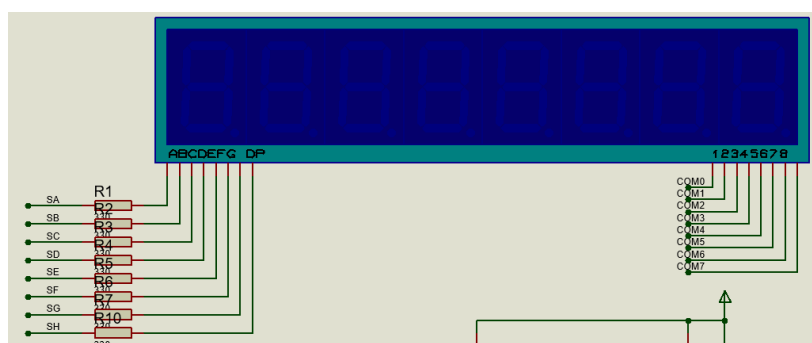


图 3-30 步骤 3 完成图

表 3-6 Proteus 使用的元器件

元器件名称	规格	说明
AT89C51		51 单片机
BUTTON		按键, 用于最小系统复位 ; 实现输入功能
CAP	30pF	电容, 用于搭建复位电路
CAP-ELEC	10uF	电解电容, 用于搭建复位电路
CRYSTAL	11. 0592M	晶振, 给单片机提供时钟信号
7SEG-MPX8-CA-BLUE		7 段 8 位蓝色共阳数码管
RES	10k、1k、 330、220	电
NPN	8050	NPN 三极管

七、程序流程图

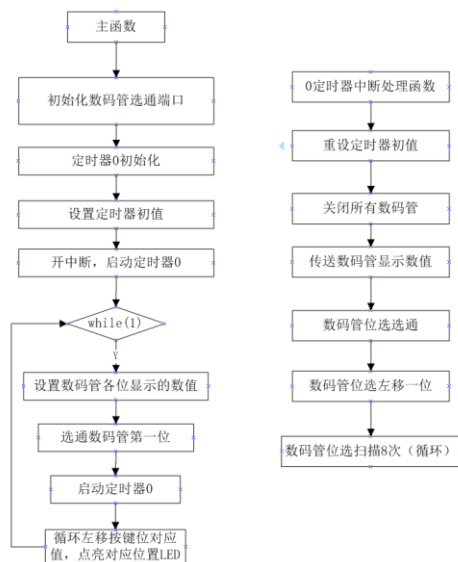


图 3-31 程序流程图

八、实验报告要求

1. 将程序及每一句加注释写在实验报告上。
2. 分析调试过程中遇到的问题和解决的办法。
3. 将实验步骤中观察的现象记录下来。

3.6 基础实验六 矩阵键盘扫描实验

一、实验目的

- 1、理解矩阵键盘扫描的原理；
- 2、掌握矩阵键盘与单片机接口的编程方法；
- 3、熟练 Proteus 仿真软件的使用；
- 4、熟练单片机实验箱的实验操作。

二、实验设备和器件

PC 机一台，PROTEUS 仿真软件一套，实验箱一台，ISP 下载器一台。

三、实验要求

读取矩阵键盘键值，对键值进行判断和处理，通过 1 位数码管显示。

四、实验原理

在某一个系统设计中，如果需要使用很多的按键时，做成独立按键会大量占用 IO 口，因此引入了矩阵按键的设计。使用 8 个 IO 口来实现了 16 个按键。本实验的矩阵键盘如下图所示。

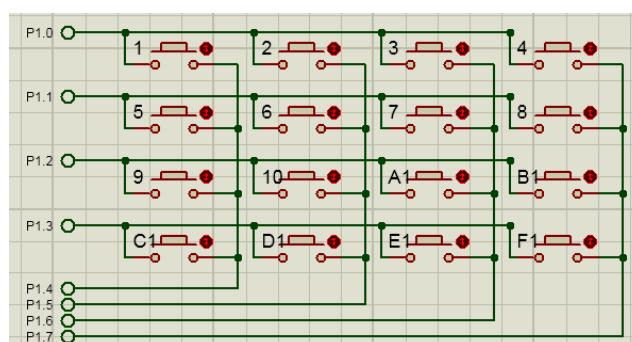


图 3-32 矩阵键盘

行列扫描法原理：

第一步，拉低所有的行线，判断列线的变化，如果有按键按下，按键按下的对应列线被拉低，否则所有的列线都为高电平。

第二步，在第一步判断有键按下后，延时 10ms 消除机械抖动，再次读取行值，如

果此行线还处于低电平状态则进入下一步，否则返回第一步重新判断。

第三步，开始扫描按键位置，采用逐列扫描，读取列值找到按键的位置，分别把行值和列值储存在寄存器里。

第四步，从寄存器中找到行值和列值并把其合并，得到按键值，对此按键值进行编码。

五、硬件连接图

1、硬件电路

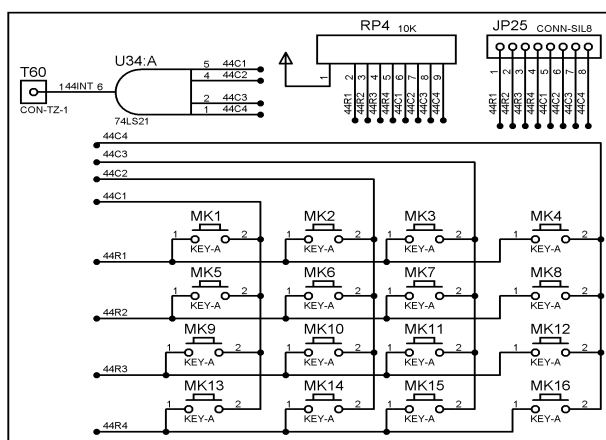


图 3-33 4*4 矩阵键盘

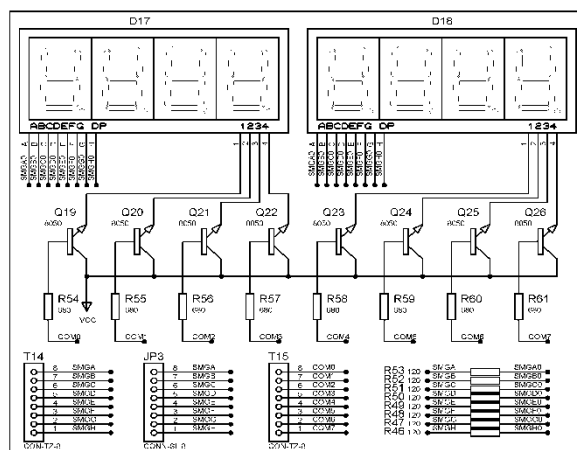


图 3-34 8 位共阳数码管

2、硬件连接

用连接线将 P00~P07 依次与 8 位共阳数码管 SA~SG 端相连；

用连接线将 P10~P13 依次与 4*4 矩阵键盘 R1~R4 端相连；

用连接线将 P14~P17 依次与 4*4 矩阵键盘 C1~C4 端相连；

用连接线将 P20 端与 8 位共阳数码管的 C0 相连。

六、Proteus 原理图设计

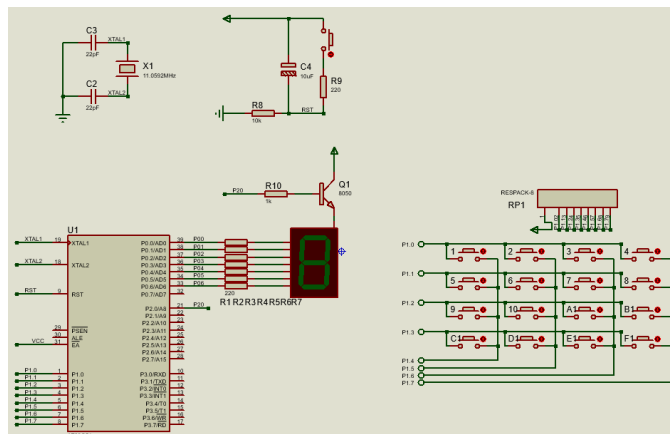


图 3-35 总原理图

步骤 1.在 Proteus 中绘制单片机最小系统，包括主控芯片、晶振电路和复位电路。

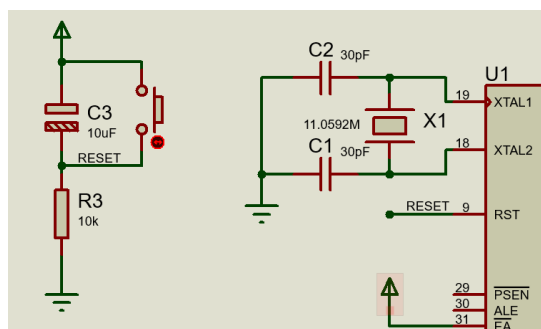


图 3-36 步骤 1 完成图

步骤 2. 添加 4*4 矩阵按键，按照行和列排好，各行和列连接到 P1 口，P1 口添加排阻上拉。

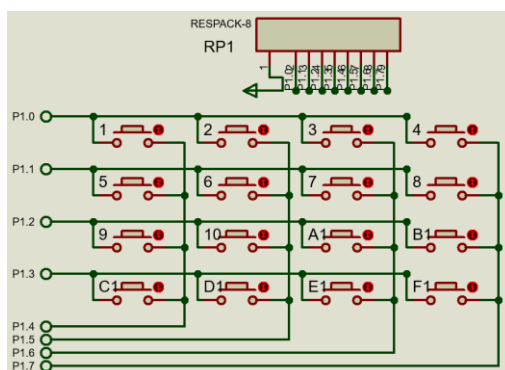


图 3-37 步骤 2 完成图

步骤 3. 添加一位共阳数码管，段选分别连接电阻后接到 P0.0-P0.6，位选连接到三

极管发射极，基极连接到 P2.0 口，作为选通开关。

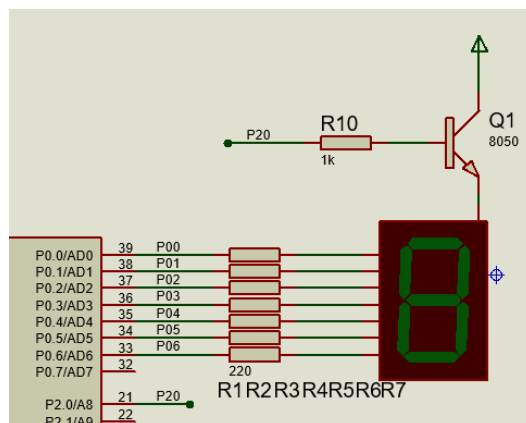


图 3-38 步骤 3 完成图

表 3-6 Proteus 使用的元器件

元器件名称	规格	说明
AT89C51		51 单片机
BUTTON		按键, 用于最小系统复位 ; 实现输入功能
CAP	30pF	电容, 用于搭建复位电路
CAP-ELEC	10uF	电解电容, 用于搭建复位电路
CRYSTAL	11.0592M	晶振, 给单片机提供时钟信号
7SEG-MPX8-CA-BLUE		7 段 8 位蓝色共阳数码管
RES	10k、220、1k	电阻
RESPACK-8	10k	排阻 ; RP1 是上拉排阻
NPN		NPN 三极管
7SEG-COM-AN-GRN		共阳数码管

七、程序流程图

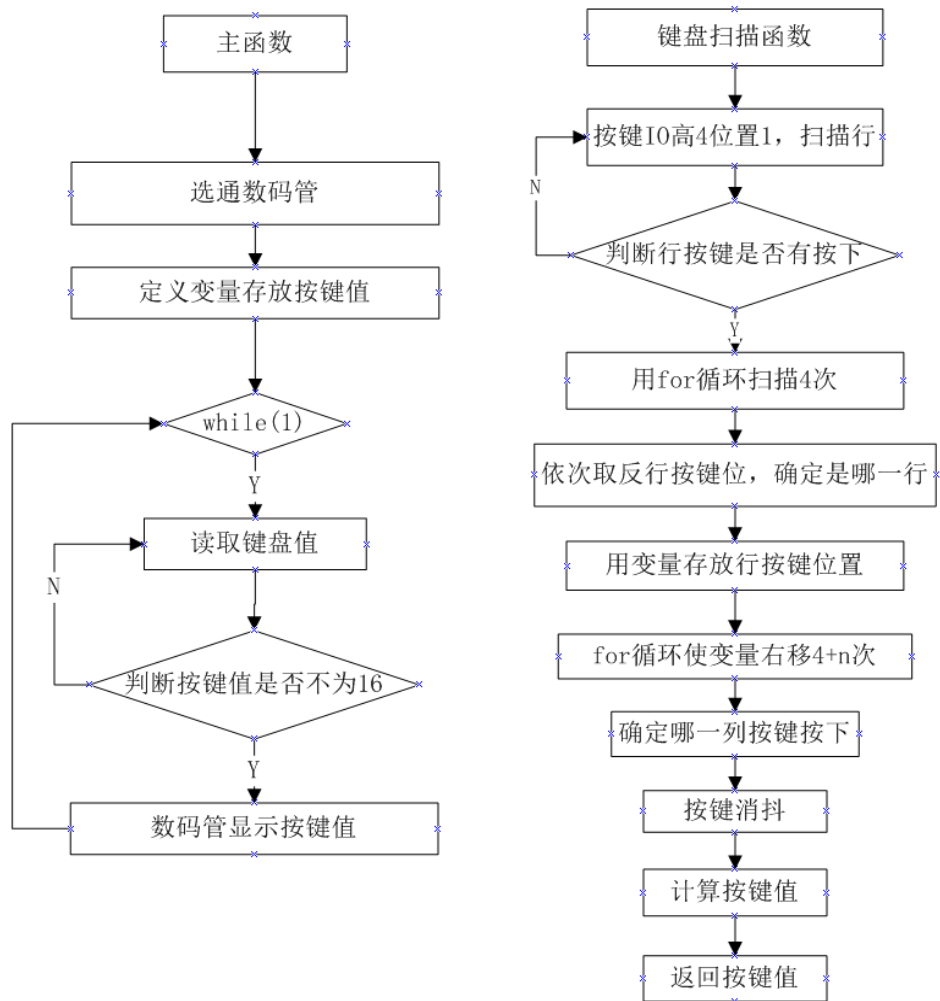


图 3-39 程序流程图

八、实验报告要求

1. 将程序及每一句加注释写在实验报告上。
2. 分析调试过程中遇到的问题和解决的办法。
3. 将实验步骤中观察的现象记录下来。

3.7 创新实验七 步进电机控制实验

一、实验目的

- 1、了解步进电机控制的基本原理；
- 2、掌握控制步进电机转动的编程方法；
- 3、熟练 Proteus 仿真软件的使用；
- 4、熟练单片机实验箱的实验操作。

二、实验设备和器件

PC 机一台，PROTEUS 仿真软件一套，实验箱一台，ISP 下载器一台。

三、实验要求

利用单片机实现对步进电机的控制，编写程序，用四路 I/O 口实现环形脉冲的分配，控制步进电机按固定方向连续转动。同时，要求按下 A 键时，控制步进电机正转；按下 B 键时，控制步进电机反转；放开按键时，电机停止转动。

四、实验原理

步进电机驱动原理是通过对每组线圈中的电流的顺序切换来使电机作步进式旋转。切换是通过单片机输出脉冲信号来实现的。所以调节脉冲信号的频率就可以改变步进电机的转速，改变各相脉冲的先后顺序，就可以改变电机的转向。步进电机的转速应由慢到快逐步加速。

ULN2003A 是高压大电流达林顿晶体管阵列系列产品，具有电流增益高、工作电压高、温度范围宽、带负载能力强等特点，适应于各类要求高速大功率驱动的系统。ULN2003A 是一个 7 路反向器电路，即当输入端为高电平时 ULN2003A 输出端为低电平，当输入端为低电平时 ULN2003A 输出端为高电平。此处作为步进电机的驱动电路。

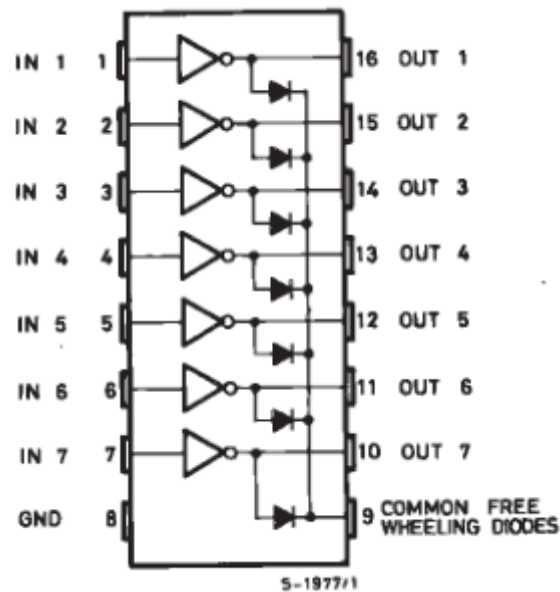


图 3-40 ULN2003A

五、硬件连接图

1、硬件电路

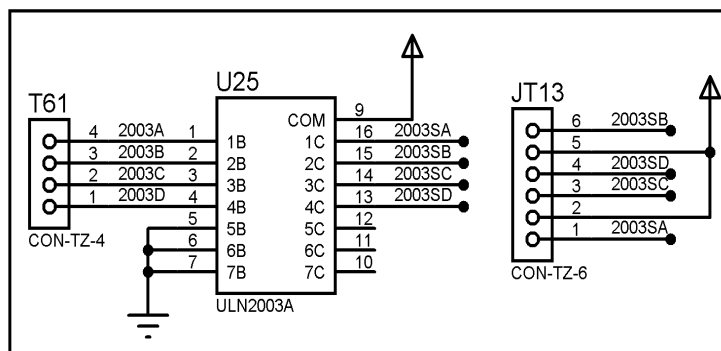


图 3-41 步进电机模块

2、硬件连接

用连接线将 P00 与 8 位独立按键 K1 相连；

用连接线将 P01 与 8 位独立按键 K2 相连；

用连接线将 P20 与电机模块 SA 端相连；

用连接线将 P21 与电机模块 SB 端相连。

用连接线将 P22 与电机模块 SC 端相连。

用连接线将 P23 与电机模块 SD 端相连。

六、Proteus 原理图设计

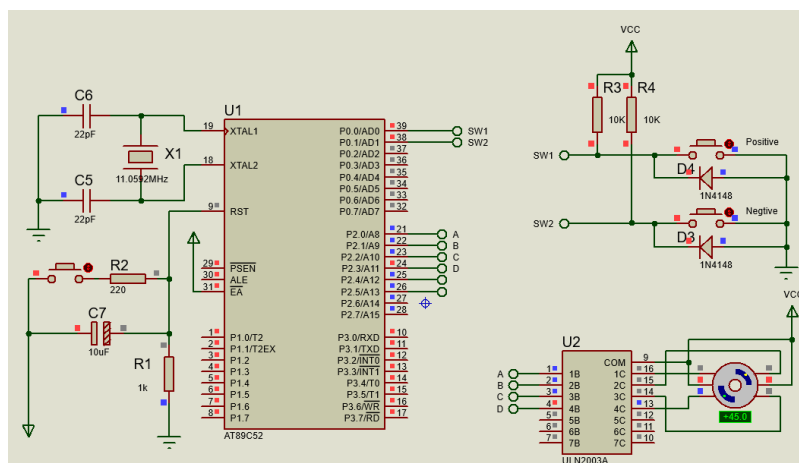


图 3-42 总原理图

步骤 1.在 Proteus 中绘制单片机最小系统，包括主控芯片、晶振电路和复位电路。

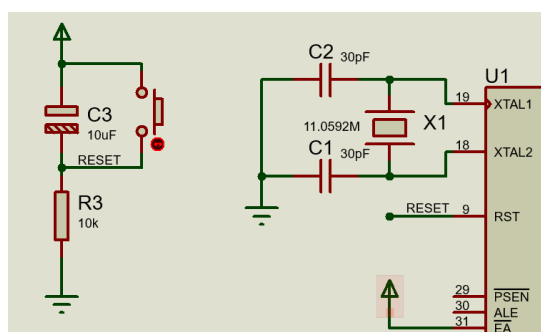


图 3-43 步骤 1 完成图

步骤 2. 添加 ULN2003A 驱动芯片和步进电机, 芯片输入端 ABCD6 单片机 P2.0-P2.3 口, 端口 COM 连接到电机的脉冲输入端并同时接高电平, 芯片输出端 1C、2C 分别接到电机的正向控制端口, 芯片输出端 3C、4C 分别接到电机的反向控制端口。

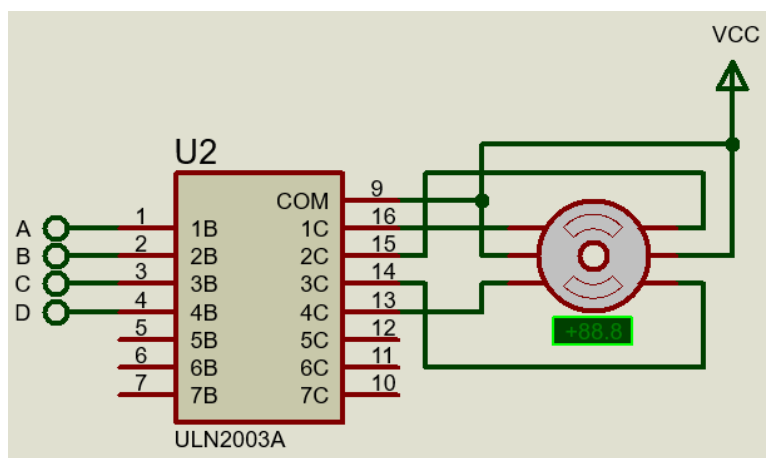


图 3-44 步骤 2 完成图

步骤 3. 添加 2 个按键、两个整流二极管以及电阻，构成以下开关控制电路，并连接到 P0.0 和 P0.1 口，注意二极管摆放方向。

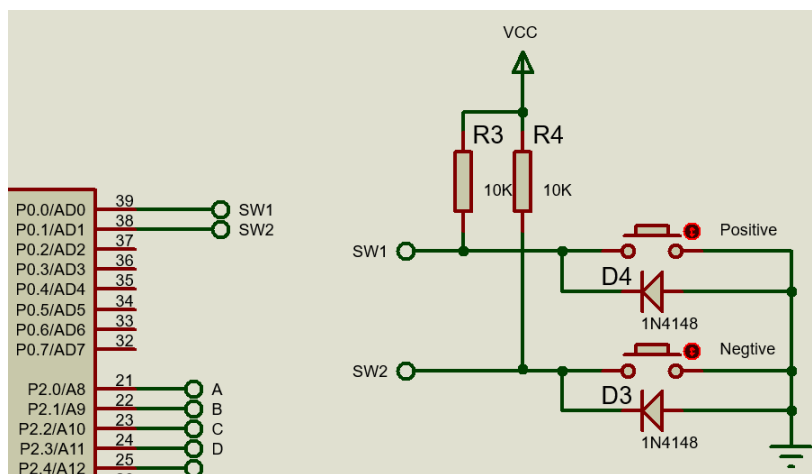


图 3-45 步骤 3 完成图

表 3-7 Proteus 使用的元器件

元器件名称	规格	说明
AT89C51		51 单片机
BUTTON		按键, 用于最小系统复位 ; 实现输入功能
CAP	30pF	电容, 用于搭建复位电路
CAP-ELEC	10uF	电解电容, 用于搭建复位电路
CRYSTAL	11. 0592M	晶振, 给单片机提供时钟信号
7SEG-MPX8-CA-BLUE		7 段 8 位蓝色共阳数码管
RES	10k	电阻
ULN2003A		高耐压、大电流复合晶体管阵列
MOTOR-STEPPER		步进电机
1N4148		整流二极管

七、程序流程图

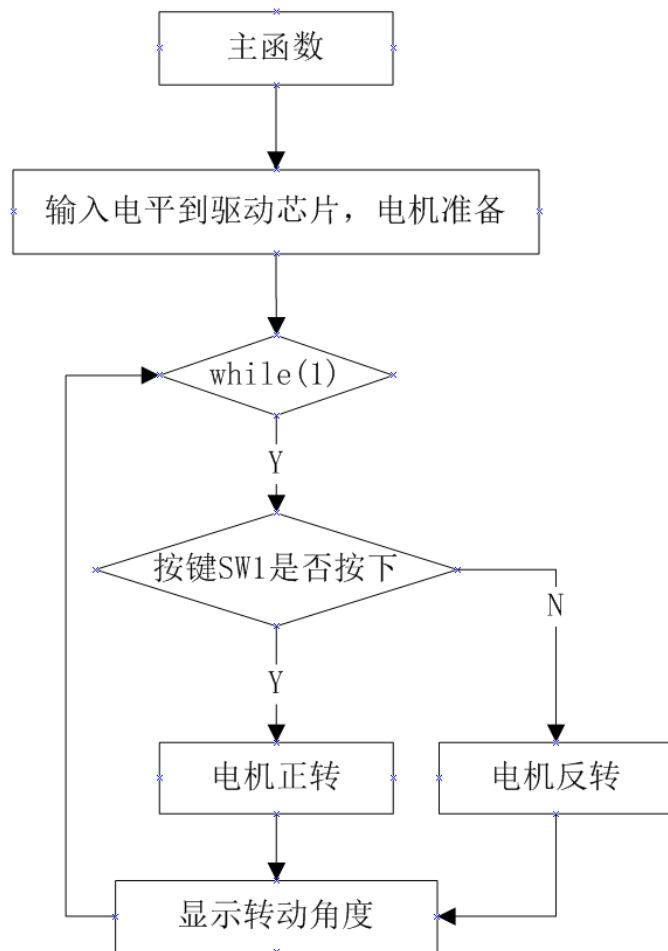


图 3-46 程序流程图

八、实验报告要求

1. 将程序及每一句加注释写在实验报告上。
2. 分析调试过程中遇到的问题和解决的办法。
3. 将实验步骤中观察的现象记录下来。
4. 若步进电机在运行过程中出现了抖动，如何消除。

3.8 创新实验八 PC 与单片机串口通讯实验

一、实验目的

- 1、理解用异步串行通信进行 RS232 通信的原理并能掌握其方法及编程；
- 2、计算波特率的计数方法；
- 3、熟练 Proteus 仿真软件的使用；
- 4、熟练单片机实验箱的实验操作。

二、实验设备和器件

PC 机一台，PROTEUS 仿真软件一套，实验箱一台，ISP 下载器一台。

三、实验要求

编写程序，实现单片机的串口同 PC 机通讯，并能传输相应的字符串。

四、实验原理

通信按照基本类型可以分为并行通信和串行通信。并行通信时数据的各个位同时传送，可以实现字节为单位通信，但是通信线多占用资源多，成本高。而串行通信，是一位一位的发送出去的，要发送 8 次才能发送完一个字节。

AT89C52 有两个引脚是专门用来做 UART 串行通信的，一个是 P3.0 一个是 P3.1，它们还分别有另外的名字叫做 RXD 和 TXD，由它们组成的通信接口就叫做串行接口，简称串口。TXD 是串行发送引脚，RXD 是串行接收引脚。

波特率：发送二进制数据位的速率，习惯上用 baud 表示，即我们发送一位二进制数据的持续时间 $=1/\text{baud}$ 。在通信之前，两台设备首先都要明确的约定好它们之间的通信波特率，必须保持一致，收发双方才能正常实现通信。

约定好速度后，我们还要考虑第二个问题，数据什么时候是起始，什么时候是结束呢？不管是提前接收还是延迟接收，数据都会接收错误。在 UART 通信的时候，一个字节是 8 位，规定当没有通信信号发生时，通信线路保持高电平，当要发送数据之前，先发一位 0 表示起始位，然后发送 8 位数据位，数据位是先低后高的顺序，数据位发完后

再发一位 1 表示停止位。这样本来要发送一个字节的 8 位数据，而实际上我们一共发送了 10 位，多出来的两位其中一位起始位，一位停止位。



图 3-47 串口数据发送示意图

RS232 是个反逻辑，也叫做负逻辑。它的 TXD 和 RXD 的电压，-3V~-15V 电压代表是 1，+3~+15V 电压代表是 0。低电平代表的是 1，而高电平代表的是 0，所以称之为负逻辑。因此电脑的 9 针 RS232 串口是不能和单片机直接连接的，需要一个电平转换芯片 MAX232 来完成，如下图所示。这个芯片就可以实现把标准 RS232 串口电平转换成我们单片机能够识别和承受的 UART 0V/5V 电平。其实 RS232 串口和 UART 串口，它们的协议类型是一样的，只是电平标准不同而已，而 MAX232 这个芯片起到的就是中间人的作用，它把 UART 电平转换成 RS232 电平，也把 RS232 电平转换成 UART 电平，从而实现标准 RS232 接口和单片机 UART 之间的通信连接。

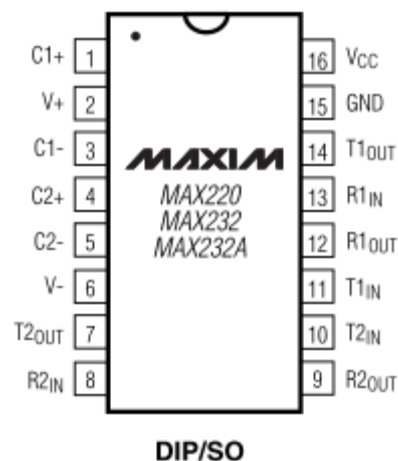


图 3-48 MAX232

五、硬件连接图

1、硬件连接

用连接线将 P30 与 RXD 相连；

用连接线将 P31 与 TXD 相连；

用串口线将实验箱串口与 PC 端串口相连；

六、Proteus 原理图设计

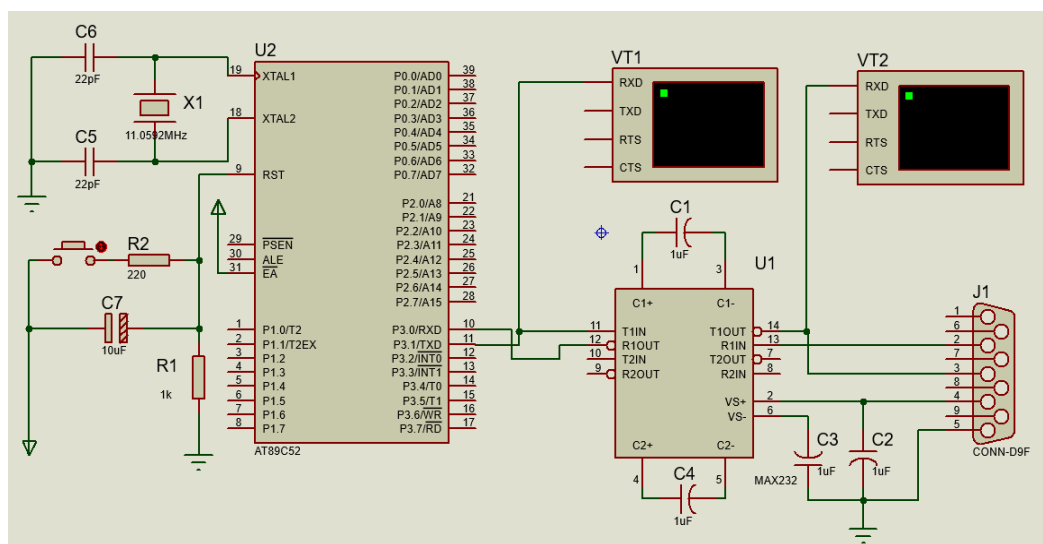


图 3-49 总原理图

步骤 1.在 Proteus 中绘制单片机最小系统，包括主控芯片、晶振电路和复位电路。

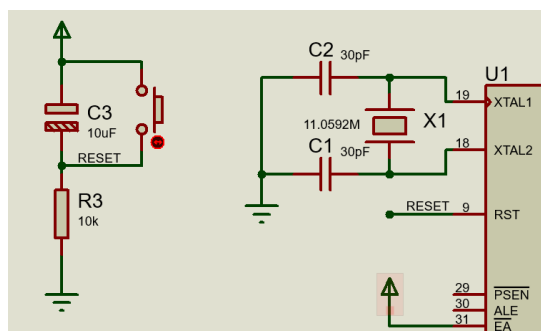


图 3-50 步骤 1 完成图

步骤 2. 添加 MAX232 通信芯片，T1IN 端口连接单片机 TXD 端口，R1OUT 端口连接单片机 RXD 端口，C1+和 C1-、C2+和 C2-都连接一个 1uF 电容，添加一个 9 孔的管脚，2 引脚与 MAX232 芯片 R1IN 连接，3 引脚与 T1OUT 连接，4 引脚与 VS+连接，5 引脚分别连接一个 1uF 电容后和 VS+、VS-连接，三者交点同时接地。

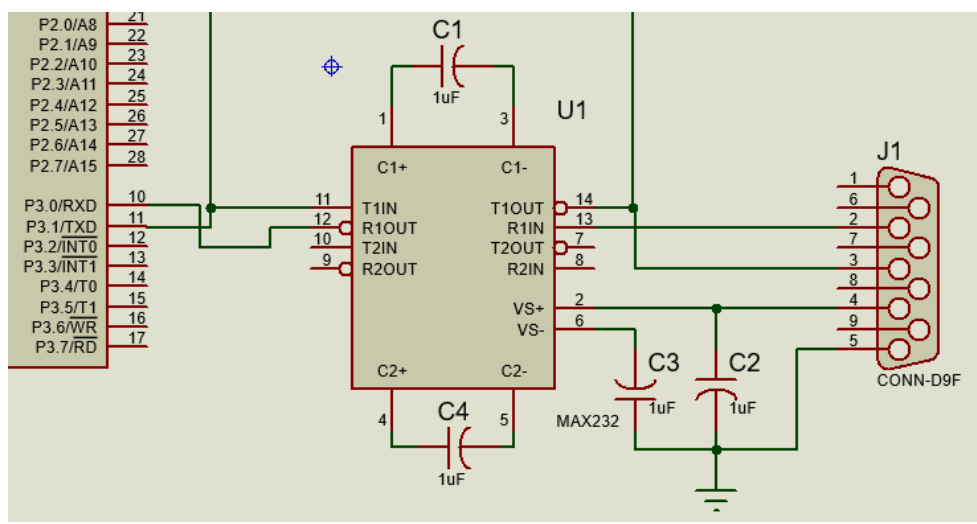


图 3-51 步骤 2 完成图

步骤 3. 添加两个虚拟终端，第一个 RXD 连接单片机 TXD 口，第二个 RXD 连接 232 芯片的 T1OUT 端口。

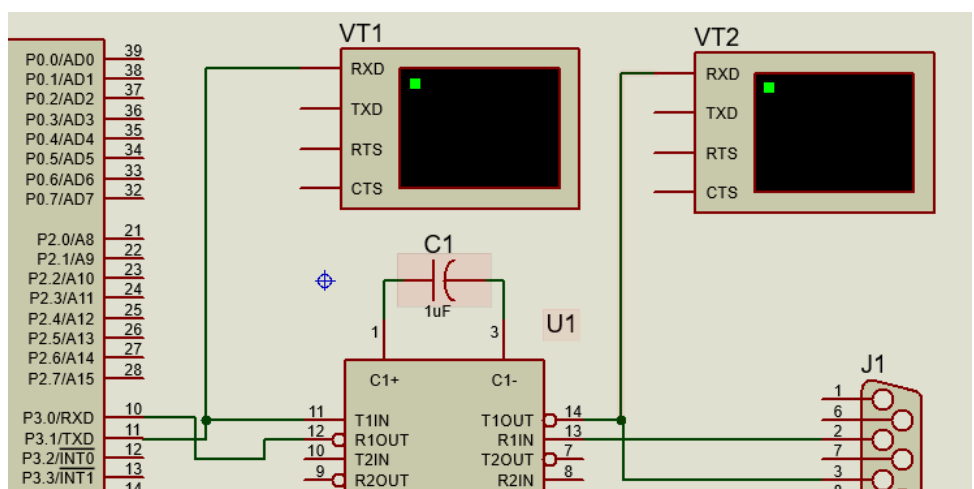


图 3-52 步骤 3 完成图

表 3-7 Proteus 使用的元器件

元器件名称	规格	说明
AT89C51		51 单片机
BUTTON		按键, 用于最小系统复位 ; 实现输入功能
CAP	30pF	电容, 用于搭建复位电路
CAP-ELEC	10uF	电解电容, 用于搭建复位电路
CRYSTAL	11.0592M	晶振, 给单片机提供时钟信号
CONN-D9F		RS232 管脚
RES	1k、220	电阻
MAX232		RS-232 标准串口设计的单电源电平转换芯片

七、程序流程图

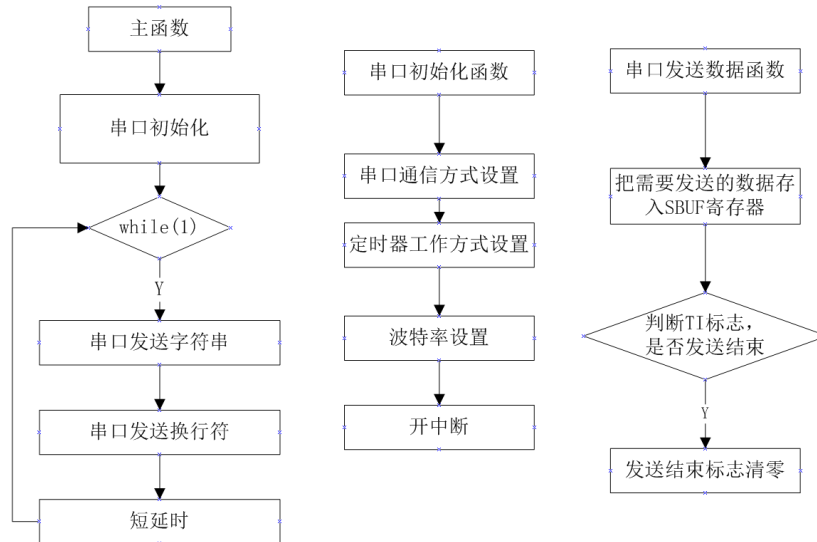


图 3-53 程序流程图

八、实验报告要求

1. 将程序及每一句加注释写在实验报告上。
2. 分析调试过程中遇到的问题和解决的办法。
3. 将实验步骤中观察的现象记录下来。
4. 如何实现两台单片机互相发送和接收信息。