实验例程

目录

简单 I/O（16 位）实验

一、实验目的与要求

1、了解外设的扩展方法，掌握外设的读写时序。

2、了解 74HC273、74HC244 的功能，掌握它们的使用方法。

3、掌握 CPU 对 16 位外设的访问方法

4、认真预习本节实验内容，尝试自行编写程序，填写实验报告。

二、实验设备

SUN 系列实验仪一套、PC 机一台

三、实验内容

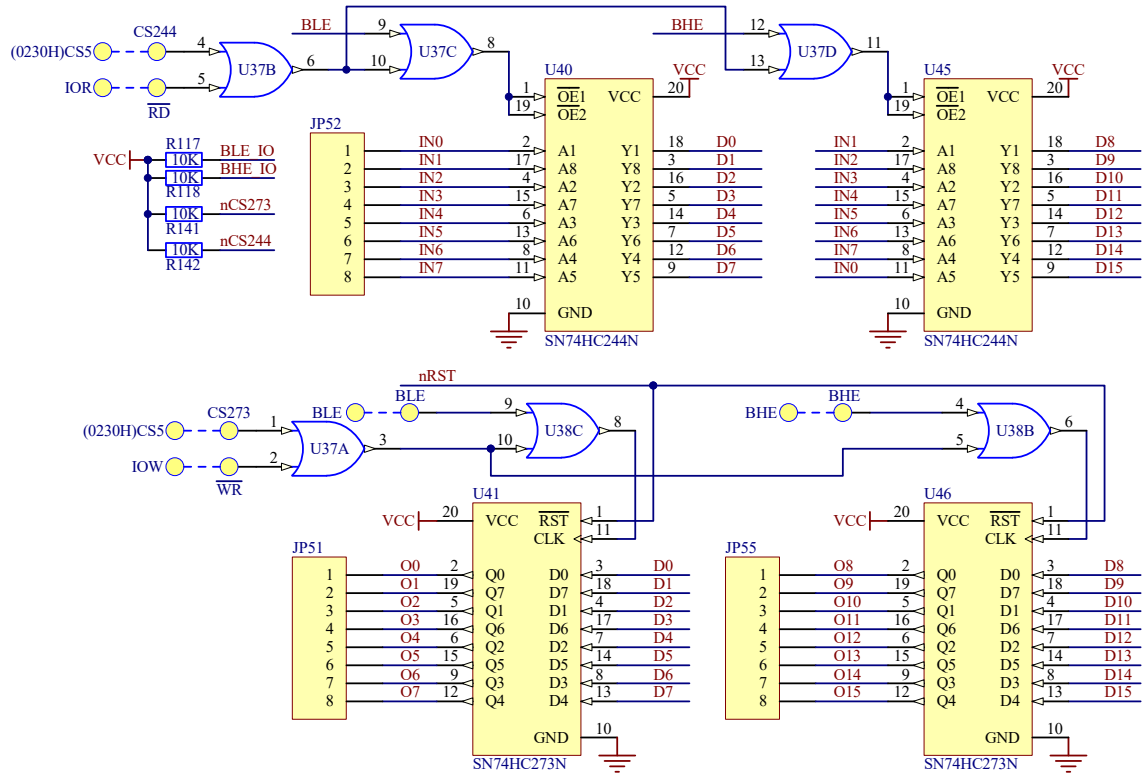
1、说明：二片 74HC244 组成 16 位的只读外设，二片 74HC273 组成 16 位的只写外设，它们都可以按字节或字方式操作。实验仪具有 16 位数据总线 D0..D15、BLE（低电平有效，选中挂在低 8 位数据总线上外设）、BHE（低电平有效，选中挂在高 8 位数据总线上外设）；BLE、BHE 同时有效，对外设字方式读写，BLE 或 BHE 有效，对外设字节方式读写。

二片 74HC273 的输出端与 F4 区的 16 个发光二极管相连；低位 74HC244 的输入端与 F4 区的 8 个拨动开关相连，8 个拨动开关循环左移一位后与高位 74HC244 的输入端相连。

2、编写程序：将 B4 区的二片 74HC244 中数据读出、写入二片 74HC273 中；然后逐一点亮挂在 74HC273 上的 16 个发光二级管；循环执行

3、连接线路验证功能，熟悉它的使用方法。

四、实验原理图



五、实验步骤

1、连线说明：

B4(I/O)区：CS273、CS244	——	A3 区：CS5、CS5
B4(I/O)区：BLE、BHE	——	A3 区：BLE、BHE
B4(I/O)区：RD、WR	——	A3 区：IOR、IOW
B4(I/O)区：JP51、JP55	——	F4 区：JP18、JP19(发光管)
B4(I/O)区：JP52	——	F4 区：JP27（开关）
B4 区：JP57(D0..D7)	——	A3 区：JP42(D0..D7)
B4 区：JP56(D8..D15)	——	A3 区：JP40(D8..D15)

2、观察实验结果，拨动开关状态是否与点亮的发光二极管一致，是否循环点亮 16 个发光二级管。

六、演示程序

```
IO244       EQU     0230H           ;244(16位)片选
IO273       EQU     0230H           ;273(16位)片选
_STACK      SEGMENT STACK
            DW      100 DUP(?)
_STACK      ENDS
_DATA       SEGMENT WORD PUBLIC 'DATA'
_DATA       ENDS
CODE        SEGMENT
START       PROC    NEAR
            ASSUME  CS:CODE, DS:_DATA, SS:_STACK
            MOV     AX,_DATA
            MOV     DX,IO244
            IN      AX,DX           ;读取开关数据(16位，K0 K7 K6 K5 K4 K3
                                    K2 K1 K7 K6 K5 K4 K3 K2 K1 K0)
            MOV     DX,IO273
            OUT     DX,AX
            CALL    Delay
            CALL    Delay
            CALL    Delay
            CALL    Delay
            CALL    Delay
            CALL    Delay
            MOV     DX,IO273
            MOV     AX,0FFFEH
START1:     OUT     DX,AX
            CALL    Delay
            TEST    AX,8000H
            JZ      START
            ROL     AX,1
            JMP     START1
Delay       PROC    NEAR            ;延时
Delay1:     XOR     CX,CX
            LOOP    $
```

```
                RET
Delay           ENDP
START           ENDP
CODE            ENDS
                END             START
```

七、实验扩展及思考

1、请按照字、字节方式画出读（74HC244）写（74HC273）的时序。

2、以上程序中，使用 16 位方式读写外设，请按照 8 位方式，重编程序。

8255 控制交通灯实验

一、实验目的与要求

1、了解 8255 芯片的工作原理，熟悉其初始化编程方法以及输入、输出程序设计技巧。学会使用 8255 并行接口芯片实现各种控制功能，如本实验（控制交通灯）等。

2、熟悉 8255 内部结构和与 8086 的接口逻辑，熟悉 8255 芯片的 3 种工作方式以及控制字格式。

3、认真预习本节实验内容，尝试自行编写程序，填写实验报告。

二、实验设备

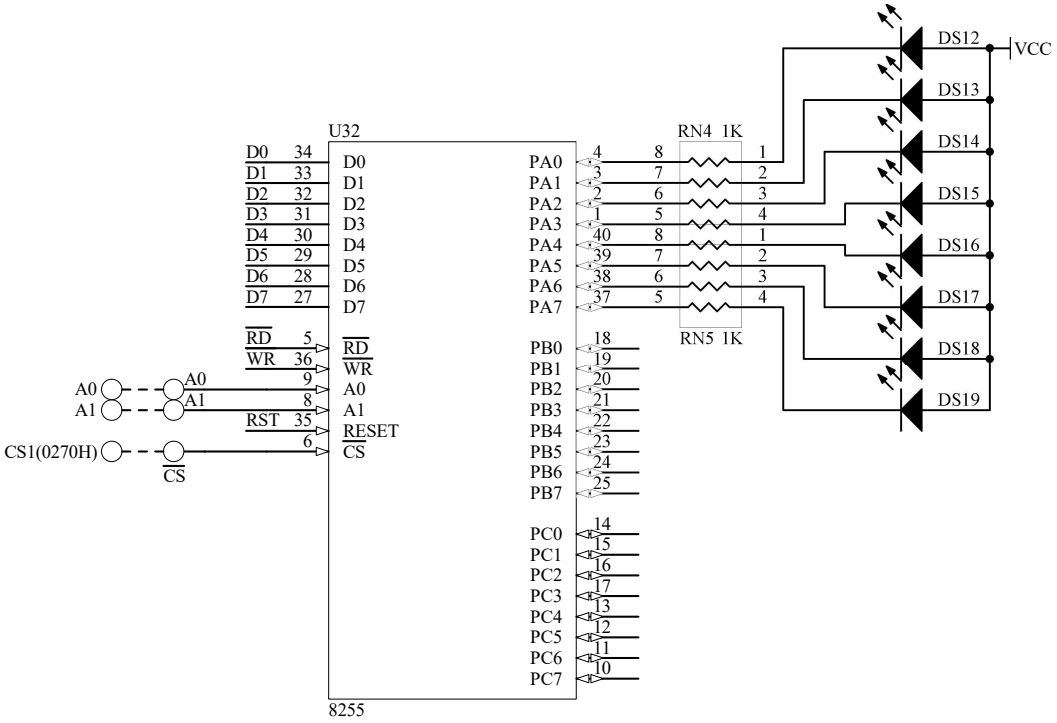
SUN 系列实验仪一套、PC 机一台

三、实验内容

1、编写程序：使用 8255 的 PA0..2、PA4..6 控制 LED 指示灯，实现交通灯功能。

2、连接线路验证 8255 的功能，熟悉它的使用方法。

四、实验原理图



五、实验步骤

1、连线说明：

D3 区：CS、A0、A1	——	A3 区：CS1、A0、A1
D3 区：JP23（PA 口）	——	F4 区：JP18

2、观察实验结果，是否能看到模拟的交通灯控制过程。

六、演示程序

```
COM_ADD         EQU         0273H
PA_ADD          EQU         0270H
PB_ADD          EQU         0271H
PC_ADD          EQU         0272H
_STACK          SEGMENT     STACK
```

```
            DW          100 DUP(?)
_STACK      ENDS
_DATA       SEGMENT     WORD PUBLIC 'DATA'
LED_Data    DB          10111110B           ;东西绿灯，南北红灯
            DB          10111111B           ;东西绿灯闪烁，南北红灯
            DB          10111101B           ;东西黄灯亮，南北红灯
            DB          11101011B           ;东西红灯，南北绿灯
            DB          11111011B           ;东西红灯，南北绿灯闪烁
            DB          11011011B           ;东西红灯，南北黄灯亮
_DATA       ENDS
CODE        SEGMENT
START       PROC        NEAR
            ASSUME      CS:CODE, DS:_DATA, SS:_STACK
            MOV         AX,_DATA
            MOV         DS,AX
            NOP
            MOV         DX,COM_ADD
            MOV         AL,80H              ;PA、PB、PC为基本输出模式
            OUT         DX,AL
            MOV         DX,PA_ADD           ;灯全熄灭
            MOV         AL,0FFH
            OUT         DX,AL
            LEA         BX,LED_Data
START1:     MOV         AL,0
            XLAT
            OUT         DX,AL               ;东西绿灯，南北红灯
            CALL        DL5S
            MOV         CX,6
START2:     MOV AL,1
            XLAT
            OUT         DX,AL               ;东西绿灯闪烁，南北红灯
            CALL        DL500ms
            MOV         AL,0
            XLAT
            OUT         DX,AL
            CALL        DL500ms
            LOOP        START2
            MOV         AL,2                ;东西黄灯亮，南北红灯
            XLAT
            OUT         DX,AL
            CALL        DL3S
            MOV         AL,3                ;东西红灯，南北绿灯
            XLAT
            OUT         DX,AL
```

```
                CALL        DL5S
                MOV         CX,6
START3:         MOV         AL,4                ;东西红灯，南北绿灯闪烁
                XLAT
                OUT         DX,AL
                CALL        DL500ms
                MOV         AL,3
                XLAT
                OUT         DX,AL
                CALL        DL500ms
                LOOP        START3
                MOV         AL,5                ;东西红灯，南北黄灯亮
                XLAT
                OUT         DX,AL
                CALL        DL3S
                JMP         START1
DL500ms         PROC        NEAR
                PUSH        CX
                MOV         CX,60000
DL500ms1:       LOOP        DL500ms1
                POP         CX
                RET
DL500ms         ENDP
DL3S            PROC        NEAR
                PUSH        CX
                MOV         CX,6
DL3S1:          CALL        DL500ms
                LOOP        DL3S1
                POP         CX
                RET
                ENDP
DL5S            PROC        NEAR
                PUSH        CX
                MOV         CX,10
DL5S1:          CALL        DL500ms
                LOOP        DL5S1
                POP         CX
                RET
                ENDP
START           ENDP
CODE            ENDS
                END         START
```

七、实验扩展及思考

1、如何对 8255 的 PC 口进行位操作？

8255 键盘显示实验

一、实验目的与要求

1、进一步掌握 8255 的设计、编程方法。

2、掌握矩阵键盘的扫描方法

3、掌握动态扫描数码块的方法

4、认真预习，做好实验前的准备工作，填写实验报告

二、实验设备

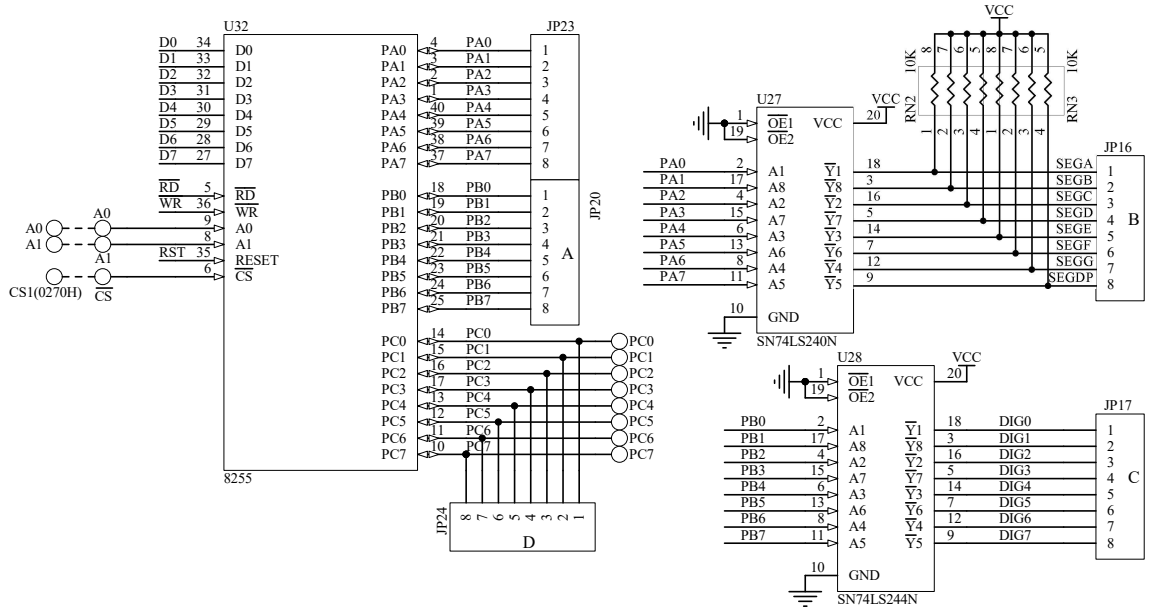
SUN 系列实验仪一套、PC 机一台

三、实验内容

1、编写程序：扫描键盘，如有按键，键号显示于数码管。

2、连接线路，验证 8255 的功能，熟悉它的使用方法。

四、实验原理图



五、实验步骤

1、连线说明：

D3 区：CS、A0、A1	——	A3 区：CS1、A0、A1
D3 区：PC0、PC1	——	F5 区：KL1、KL2
D3 区：JP20(PB 口)、JP16(B)、JP17(C)	——	F5 区：A、B、C

2、运行程序，观察实验结果（任意按下 F5 区 4X4 键盘几个键，它上面的 8 个 LED 显示器会将按键的编码从左至右依次显示出来），可依此验证对程序的正确性。

六、演示程序

```
COM_8255    EQU     0273H               ;8255 控制口
PA_8255     EQU     0270H
PB_8255     EQU     0271H
```

```
PC_8255     EQU     0272H

_STACK      SEGMENT STACK
            DW      100 DUP(?)
_STACK      ENDS
_DATA       SEGMENT WORD PUBLIC 'DATA'
buffer      DB      8 DUP(?)            ;8 个字节显示缓冲区
SEG_TAB     DB      0C0H,0F9H,0A4H,0B0H, 99H, 92H, 82H,0F8H
            DB      080H, 90H, 88H, 83H,0C6H,0A1H, 86H, 8EH,0FFH
_DATA       ENDS

CODE        SEGMENT
START       PROC    NEAR
            ASSUME  CS:CODE, DS:_DATA, SS:_STACK
            MOV     AX,_DATA
            MOV     DS,AX
            MOV     ES,AX
            NOP
            CLD                         ;0->DF, 地址自动递增
            MOV     DX,COM_8255
            MOV     AL,89H
            OUT     DX,AL               ;PA、PB 输出，PC 输入
            LEA     DI,buffer
            MOV     AL,10H
            MOV     CX,08H
            REP     STOSB
            CALL    DIR
MAIN3:      LEA     DI,buffer
MAIN2:      CALL    keyi
            STOSB
            CALL    DIR
            CMP     DI,offset buffer+8
            JNZ     MAIN2
            JMP     MAIN3
DIR         PROC    NEAR
            PUSH    AX
            PUSH    BX
            PUSH    DX
            LEA     SI,buffer           ;置显示缓冲器初值
            MOV     AH,0FEH
            LEA     BX,SEG_TAB
LD0:        MOV     DX,PA_8255
            LODSB
            XLAT                        ;取显示数据
```

```
        OUT     DX,AL           ;段数据->8255 PA 口
        INC     DX              ;扫描模式->8255 PB 口
        MOV     AL,AH
        OUT     DX,AL
        CALL    DL1             ;延迟 1ms
        MOV     DX,PB_8255
        MOV     AL,0FFH
        OUT     DX,AL
        TEST    AH,80H
        JZ      LD1
        ROL     AH,01H
        JMP     LD0
LD1:    POP     DX
        POP     BX
        POP     AX
        RET
DIR     ENDP
DL1     PROC    NEAR            ;延迟子程序
        PUSH    CX
        MOV     CX,500
        LOOP    $
        POP     CX
        RET
DL1     ENDP
KEYI    PROC    NEAR
        PUSH    BX
        PUSH    DX
LK:     CALL    AllKey          ;调用判有无闭合键子程序
        JNZ     LK1
        CALL    DIR
        CALL    DIR             ;调用显示子程序,延迟 6ms
        JMP     LK
LK1:    CALL    DIR
        CALL    DIR
        CALL    AllKey          ;调用判有无闭合键子程序
        JNZ     LK2
        CALL    DIR
        JMP     LK
LK2:    MOV     BL,0FEH         ;R2
        MOV     BH,0            ;R4
LK4:    MOV     DX,PB_8255
        MOV     AL,BL
        OUT     DX,AL
        INC     DX
```

```
            IN          AL,DX
            TEST        AL,01H
            JNZ         LONE
            XOR         AL,AL                ;0 行有键闭合
            JMP         LKP
LONE:       TEST        AL,02H
            JNZ         NEXT
            MOV         AL,08H               ;1 行有键闭合
LKP:        ADD         BH,AL
LK3:        CALL        DIR                  ;判断释放否
            CALL        AllKey
            JNZ         LK3
            MOV         AL,BH                ;键号->AL
            POP         DX
            POP         BX
            RET
NEXT:       INC         BH                   ;列计数器加 1
            TEST        BL,80H
            JZ          KND                  ;判是否已扫到最后一列
            ROL         BL,01H
            JMP         LK4
KND:        JMP         LK
KEYI        ENDP
AllKey      PROC        NEAR
            MOV         DX,PB_8255
            XOR         AL,AL
            OUT         DX,AL                ;全"0"->扫描口
            INC         DX
            IN          AL,DX                ;读键状态
            NOT         AL
            AND         AL,03H               ;取低二位
            RET
AllKey      ENDP
START       ENDP
CODE        ENDS
            END         START
```

七、实验扩展及思考

1、显示程序中延时函数起什么作用？如何调节数码块亮度？

2、重新编写软件实验二，自己编写键扫描、显示程序

8253 方波实验

一、实验目的与要求

了解 8253 的内部结构、工作原理；了解 8253 与 8086 的接口逻辑；熟悉 8253 的控制寄存器和初始化编程方法，熟悉 8253 的 6 种工作模式。

二、实验设备

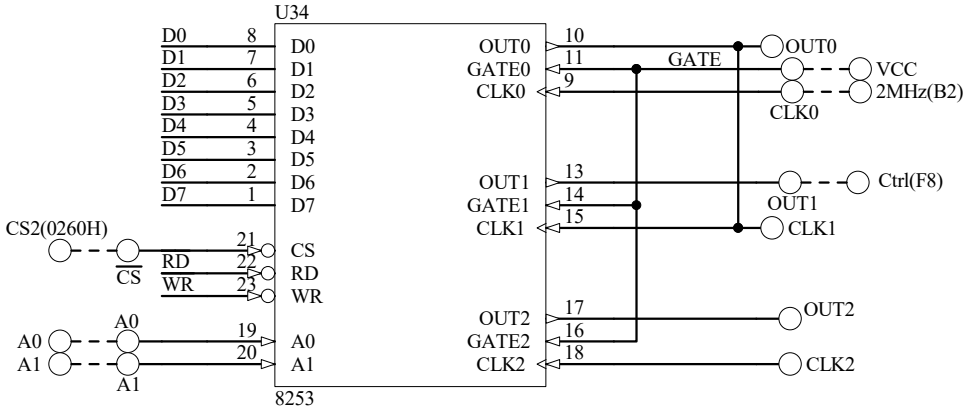
SUN 系列实验仪一套、PC 机一台

三、实验内容

1、编写程序：使用 8253 的计数器 0 和计数器 1 实现对输入时钟频率的两级分频，得到一个周期为 1 秒的方波，用此方波控制蜂鸣器，发出报警信号，也可以将输入脚接到逻辑笔上来检验程序是否正确。

2、连接线路，验证 8253 的功能，熟悉它的使用方法。

四、实验原理图



五、实验步骤

1、连线说明：

C4 区：CS、A0、A1	——	A3 区：CS2、A0、A1
C4 区：CLK0	——	B2 区：2M
C4 区：OUT0	——	C4 区：CLK1
C4 区：OUT1	——	F8 区：Ctrl(蜂鸣器)
C4 区：GATE	——	C1 区的 VCC

2、测试实验结果：蜂鸣器发出时有时无的声音；用逻辑笔测试蜂鸣器的输入端口，红绿灯交替点亮。

六、演示程序

```
COM_ADDR    EQU         0263H
T0_ADDR     EQU         0260H
T1_ADDR     EQU         0261H

_STACK      SEGMENT     STACK
```

```
            DW          100 DUP(?)
_STACK      ENDS
CODE        SEGMENT
START       PROC        NEAR
            ASSUME      CS:CODE, SS:_STACK
            MOV         DX,COM_ADDR
            MOV         AL,35H
            OUT         DX,AL               ;计数器T0设置在模式2状态,BCD码计数
            MOV         DX,T0_ADDR
            MOV         AL,00H
            OUT         DX,AL
            MOV         AL,10H
            OUT         DX,AL               ;CLK0/1000
            MOV         DX,COM_ADDR
            MOV         AL,77H
            OUT         DX,AL               ;计数器T1为模式3状态，输出方波,BCD码计数
            MOV         DX,T1_ADDR
            MOV         AL,00H
            OUT         DX,AL
            MOV         AL,10H
            OUT         DX,AL               ;CLK1/1000
            JMP         $                   ;OUT1输出1S的方波
START       ENDP
CODE        ENDS
            END         START
```

七、实验扩展及思考

1、8253 还有其它五种工作方式，其它工作模式下，硬件如何设计？程序如何编写？

2、使用 8253，编写一个实时钟程序。

8259 中断控制器实验

一、实验目的与要求

了解 8259A 的内部结构、工作原理；了解 8259A 与 8086 的接口逻辑；掌握对 8259A 的初始化编程方法，了解 8086 是如何响应中断、退出中断的。

复习本节实验内容，可尝试自行编写程序，做好实验准备工作，填写实验报告。

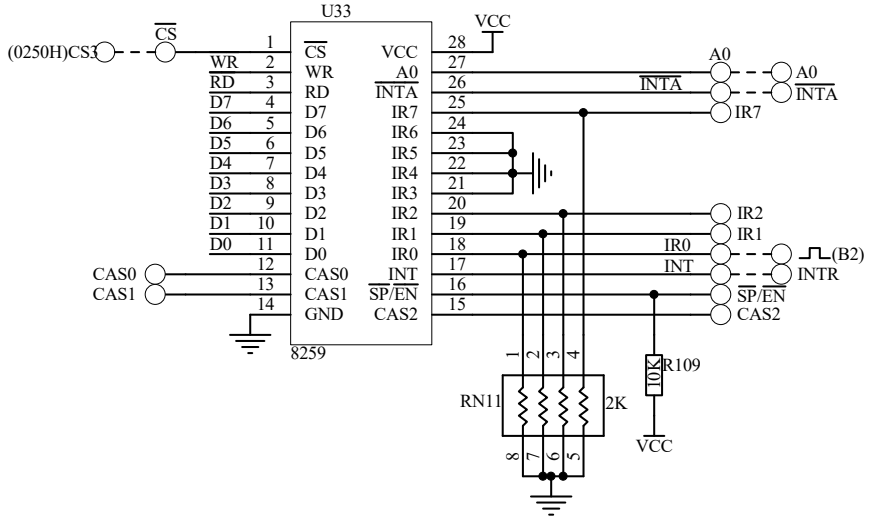
二、实验设备

SUN 系列实验仪一套、PC 机一台

三、实验内容

1、编制程序：拨动单脉冲开关，“⎍”送给 8259A 的 IR0，触发中断，8086 计数中断次数，显示于 F5 区的数码管上

四、实验原理图



五、实验步骤

1、连线说明：

B3 区：CS、A0	——	A3 区：CS3、A0
B3 区：INT、INTA	——	A3 区：INTR、INTA
B3 区：IR0	——	B2 区：单脉冲⎍
D3 区：CS、A0、A1	——	A3 区：CS1、A0、A1
D3 区：PC0、PC1	——	F5 区：KL1、KL2
D3 区：JP20、B、C	——	F5 区：A、B、C

2、运行程序

3、上下拨动单脉冲开关，拨动二次，产生一个“⎍”，观察结果，数码管上显示的次数与拨动开关次数是否对应。

六、演示程序

```
EXTRN       InitKeyDisplay:NEAR, Display8:NEAR
IO8259_0    EQU         0250H
IO8259_1    EQU         0251H
_STACK      SEGMENT     STACK
            DW          100 DUP(?)
```

```
_STACK          ENDS
_DATA           SEGMENT       WORD PUBLIC 'DATA'
BUFFER          DB            8 DUP(?)
Counter         DB            ?
ReDisplayFlag   DB            0
_DATA           ENDS

CODE            SEGMENT
START           PROC          NEAR
                ASSUME        CS:CODE, DS:_DATA, SS:_STACK
                MOV           AX,_DATA
                MOV           DS,AX
                MOV           ES,AX
                NOP
                CALL          InitKeyDisplay      ;对键盘、数码管控制器8255初始化
                CALL          Init8259
                CALL          WriIntver
                MOV           Counter,0           ;中断次数
                MOV           ReDisplayFlag,1     ;需要显示
                STI           ;开中断
START1:         LEA           SI,Buffer
                CALL          Display8
                CMP           ReDisplayFlag,0
                JZ            START1
                CALL          LedDisplay
                MOV           ReDisplayFlag,0
                JMP           START1
Init8259        PROC          NEAR
                MOV           DX,IO8259_0
                MOV           AL,13H
                OUT           DX,AL
                MOV           DX,IO8259_1
                MOV           AL,08H
                OUT           DX,AL
                MOV           AL,09H
                OUT           DX,AL
                MOV           AL,0FEH
                OUT           DX,AL
                RET
Init8259        ENDP
WriIntver       PROC          NEAR
                PUSH          ES
                MOV           AX,0
                MOV           ES,AX
```

```
                MOV             DI,20H
                LEA             AX,INT_0
                STOSW
                MOV             AX,CS
                STOSW
                POP             ES
                RET
WriIntver       ENDP
LedDisplay      PROC            NEAR
                MOV             AL,Counter
                MOV             AH,AL
                AND             AL,0FH
                MOV             Buffer,AL
                AND             AH,0F0H
                ROR             AH,4
                MOV             Buffer + 1,AH
                MOV             Buffer + 2,10H      ;高六位不需要显示
                MOV             Buffer + 3,10H
                MOV             Buffer + 4,10H
                MOV             Buffer + 5,10H
                MOV             Buffer + 6,10H
                MOV             Buffer + 7,10H
                RET
LedDisplay      ENDP
INT_0:          PUSH            DX
                PUSH            AX
                MOV             AL,Counter
                ADD             AL,1
                DAA
                MOV             Counter,AL
                MOV             ReDisplayFlag,1
                MOV             DX,IO8259_0
                MOV             AL,20H
                OUT             DX,AL
                POP             AX
                POP             DX
                IRET
START           ENDP
CODE            ENDS
                END             START
```

七、实验扩展及思考

1、从 8259A 收到上升沿，到 8086 响应中断，试画这个过程的时序图。

8250 可编程通信实验(与微机)

一、实验目的与要求

了解 8250 的内部结构、工作原理；了解 8250 与 8086 的接口逻辑；掌握对 8250 的初始化编程方法，学会使用 8250 实现设备之间的串行通信。

认真预习，做好实验前的准备工作，填写实验报告

二、实验设备

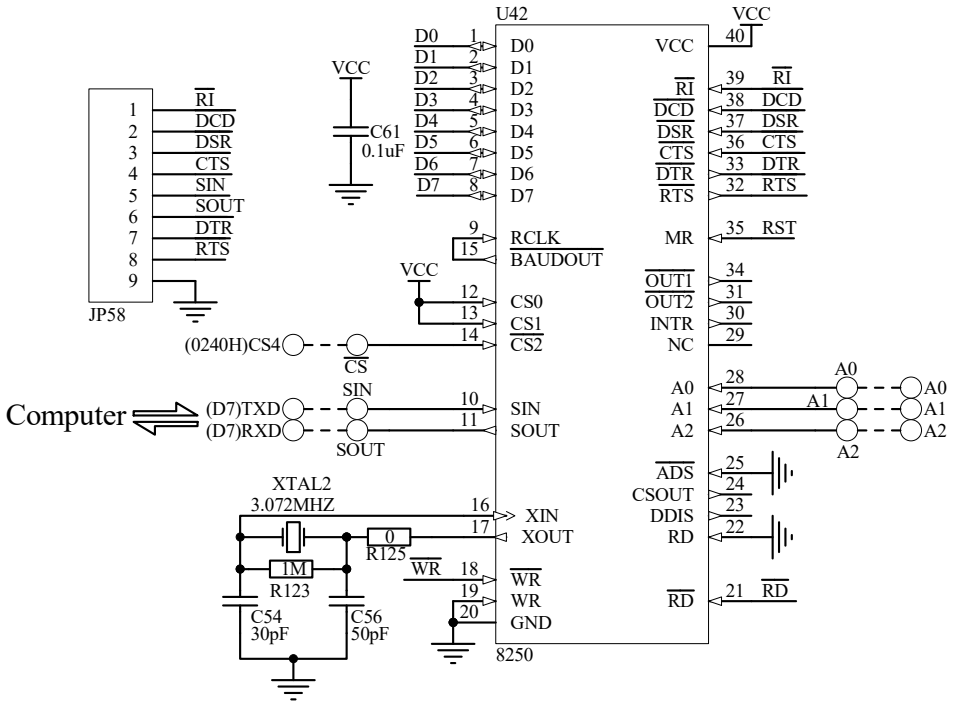
SUN 系列实验仪一套、PC 机一台

三、实验内容

1、编写程序：通过初始化 8250，设置波特率为 4800bps(或其它，但与微机部分一致)，数据格式为 8 数据位，1 停止位，偶校验；然后打开 PC 机的串行通信测试软件，向 8250 发送一批数据，8250 接收完数据之后，再将数据依次发送回去。

2、按图连线，运行程序，观察实验结果，掌握 8250 的各项功能及编程方法。

四、实验原理图



五、实验步骤

1、连线说明：

D5 区：CS、A0、A1、A2	——	A3 区：CS4、A0、A1、A2
D5 区：SIN、SOUT	——	D7 区：RxD、TxD

2、运行程序

3、运行“串口助手(ComPort.EXE)”，设置串口(波特率 4800,8 个数据位，一个停止位，偶校验)，打开串口，选择“HEX 发送”、“HEX 显示”，向 8250 发送 10 个字节数据(输入数据之间用空格分隔)，是否能接收到 10 个字节数据，接收到的数据是否与发送数据一致。

4、改变传输数据的数目，重复实验，观察结果。

六、演示程序

```
;8250和PC机通信，需要在PC上运行一个串口软件，并设置与8250相同的波特率
NS8250_Base_Address   EQU     0240H
RHR             EQU     NS8250_Base_Address     ;接收数据缓冲区
THR             EQU     NS8250_Base_Address     ;发送数据缓冲区
IER             EQU     NS8250_Base_Address+1   ;中断控制寄存器
FCR             EQU     NS8250_Base_Address+2   ;FIFO控制寄存器
ISR             EQU     NS8250_Base_Address+2   ;中断状态寄存器
LCR             EQU     NS8250_Base_Address+3   ;串行口控制寄存器
MCR             EQU     NS8250_Base_Address+4   ;MODEM控制寄存器
LSR             EQU     NS8250_Base_Address+5   ;串行口状态寄存器
MSR             EQU     NS8250_Base_Address+6   ;MODEM状态寄存器
DLL             EQU     NS8250_Base_Address     ;波特率除数锁存器低位
DLM             EQU     NS8250_Base_Address+1   ;波特率除数锁存器高位
_STACK          SEGMENT STACK
                DW      100 DUP(?)
_STACK          ENDS
_DATA           SEGMENT WORD PUBLIC 'DATA'
Receive_Buffer  DB      10 DUP(0)           ;接受缓冲器
Send_Buffer     EQU     Receive_Buffer      ;发送缓冲器
_DATA           ENDS
CODE            SEGMENT
START           PROC    NEAR
                ASSUME  CS:CODE, DS:_DATA, SS:_STACK
                MOV     AX,_DATA
                MOV     DS,AX
                MOV     ES,AX
                NOP
                CALL    INIT8250
START2:         MOV     CX,10               ;接收数据(接收完设定的数据个数）
                CALL    RECEIVE_GROUP
                MOV     CX,10               ;发送数据（发完设定的数据个数）
                CALL    SEND_GROUP
                JNC     START2
WARNING1:       JMP     $
;**************发送一组字符子程序，个数在CX中***********
Send_Group      PROC    NEAR
                LEA     SI,Send_Buffer
Send_Group1:    LODSB
                CALL    Send_Byte
                JC      Send_Group2
                LOOP    Send_Group1
```

```
                CLC
Send_Group2:    RET
Send_Group      ENDP
;*************接收一组字符子程序,存放首地址在DPTR中，个数在R6R7中***********
Receive_Group   PROC            NEAR
                LEA             DI,Receive_Buffer
Receive_Group1: CALL            Receive_Byte
                STOSB
                LOOP            Receive_Group1
                CLC
                RET
Receive_Group   ENDP
INIT8250        PROC            NEAR
                MOV             DX,ISR
                MOV             AL,06H
                OUT             DX,AL
                MOV             DX,LCR
                MOV             AL,83H              ;允许访问波特率因子寄存器
                OUT             DX,AL
                MOV             DX,DLL
                MOV             AL,40
;除数低位寄存器，波特率设为4800=(3.072*1000000/16)/DLMDLL
                OUT             DX,AL
                MOV             DX,DLM              ;00H送高字节寄存器
                MOV             AL,00H
                OUT             DX,AL
                MOV             DX,LCR              ;不允许访问波特率因子寄存器
                MOV             AL,1BH             ;数据格式为8数据位,1停止位,偶校验
                OUT             DX,AL
                RET
INIT8250        ENDP
;*********发送一个字节子程序，发送A中的数，失败置CY*********
Send_Byte       PROC            NEAR
                PUSH            CX
                PUSH            AX
                MOV             CX,1000
                MOV             DX,LSR
REP11:          IN              AL,DX
                TEST            AL,20H
                JNZ             OUTPORT1
                LOOP            REP11
                POP             AX
                STC
                JMP             EXIT8250
```

```
OUTPORT1:       POP             AX
                MOV             DX,RHR
                OUT             DX,AL
                CLC
EXIT8250:       POP             CX
                RET
Send_Byte       ENDP
;*********接收一个字节子程序,接收字节在A中，接收失败置1CY*********
Receive_Byte    PROC            NEAR
                MOV             DX,LSR
Receive1:       IN              AL,DX
                TEST            AL,1
                JZ              Receive1
Receive2:       MOV             DX,RHR
                IN              AL,DX
Receive3:       RET
Receive_Byte    ENDP

START           ENDP
CODE            ENDS
                END             START
```

七、实验扩展及思考

1、思考 8250 与 8251 有何异同之处？

2、8250 也可以做自发自收的实验，该如何连线及修改程序？

3、如何通过中断处理方式实现 8250 串行接收,需要更改哪些线路？

并行 DA 实验

一、实验目的

了解数模转换的原理；了解 0832 与 8086 的接口逻辑，掌握使用 DAC0832 进行数模转换。

二、实验设备

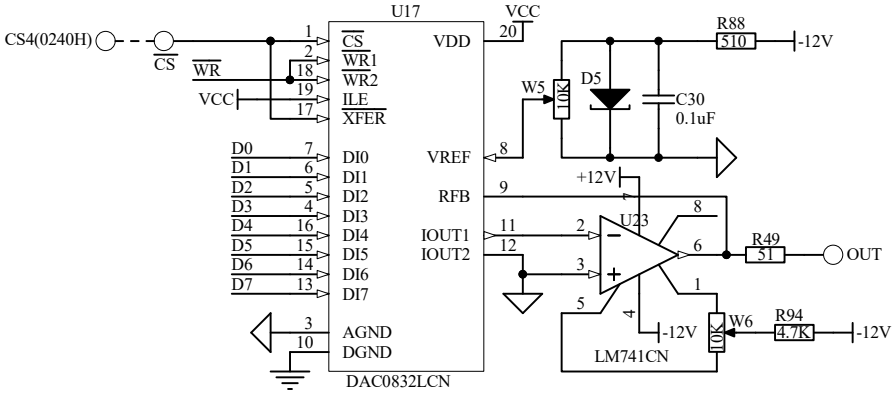
SUN 系列实验仪一套、PC 机一台、示波器一台。

三、实验内容

1、编写程序：用 0832 输出正弦波

2、按图连线，运行程序，使用示波器观察实验结果。

四、实验原理图



五、实验步骤

1、连线说明：

D2 区：CS	——	A3 区：CS4

2、运行程序，示波器的探头接 D2 区的 OUT，观察实验结果，是否产生正弦波。

六、演示程序

```
ADDR_0832       EQU         0240H               ;0832输出口地址
_STACK          SEGMENT     STACK
                DW          100 DUP(?)
_STACK          ENDS
_DATA           SEGMENT     WORD PUBLIC 'DATA'
TAB_1       DB      7FH,8BH,96H,0A1H,0ABH,0B6H,0C0H,0C9H,0D2H
            DB      0DAH,0E2H,0E8H,0EEH,0F4H,0F8H,0FBH,0FEH,0FFH,0FFH
            DB      0FFH,0FEH,0FBH,0F8H,0F4H,0EEH,0E8H,0E2H,0DAH,0D2H
            DB      0C9H,0C0H,0B6H,0ABH,0A1H,096H,08BH,07FH
            DB      74H,69H,5EH,54H,49H,40H,36H,2DH,25H,1DH,17H,11H,0BH,7,4,2,0,0
            DB      0,2,4,7,0BH,11H,17H,1DH,25H,2DH,36H,40H,49H,54H,5EH,69H,74H
_DATA           ENDS
CODE            SEGMENT
START           PROC        NEAR
```

```
                ASSUME      CS:CODE, DS:_DATA, SS:_STACK
                MOV         AX,_DATA
                MOV         DS,AX
                NOP
                MOV         DX,ADDR_0832
START1:         LEA         SI,TAB_1
                MOV         CX,72
START2:         LODSB
                OUT         DX,AL
                CALL        DELAY
                LOOP        START2
                JMP         START1
DELAY           PROC        NEAR
                PUSH        CX
                MOV         CX,80
                LOOP        $
                POP         CX
                RET
DELAY           ENDP

START           ENDP
CODE            ENDS
                END         START
```

并行 AD 实验(数字电压表实验)

一、实验目的与要求

1、了解几种类型 AD 转换的原理；掌握使用 ADC0809 进行模数转换

2、认真预习实验内容，做好准备工作，完成实验报告。

二、实验设备

SUN 系列实验仪一套、PC 机一台、万用表一个。

三、实验内容

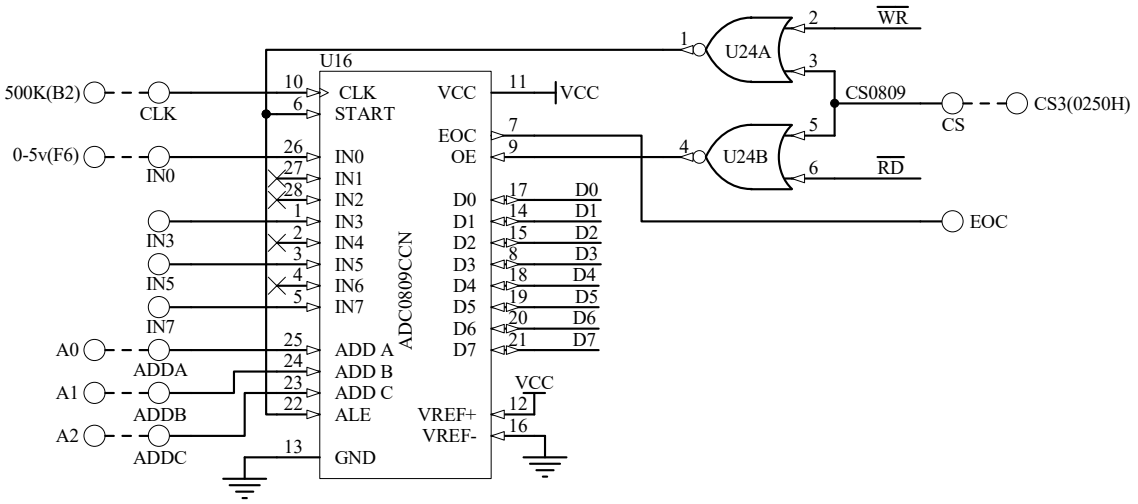
1、ADC0809(C2 区)

(1) 模数转换器，8 位精度，8 路转换通道，1 路并行输出

(2) 转换时间 100us，转换电压范围 0～5V

2、编写程序：制作一个电压表，测量 0～5V，结果显示于数码管上。

四、实验原理图



五、实验步骤

1、连线说明：

C2 区：CS、ADDA、ADDB、ADDC	——	A3 区：CS3、A0、A1、A2（选择通道）
C2 区：CLK	——	B2 区：500K
C2 区：IN0	——	F6 区：0～5V
D3 区：CS、A0、A1	——	A3 区：CS1、A0、A1
D3 区：PC0、PC1	——	F5 区：KL1、KL2
D3 区：JP20、B、C	——	F5 区：A、B、C

2、调节 0～5V 电位器（F6 区）输出电压，显示在 LED(最右边 2 位)上的电压数字量会随之改变。用万用表验证 AD 转换的结果。

六、演示程序

```
ADDR_0809       EQU             0250H
EXTRN           InitKeyDisplay:NEAR, Display8:NEAR
_STACK          SEGMENT         STACK
```

```
                DW          100 DUP(?)
_STACK          ENDS
_DATA           SEGMENT     WORD PUBLIC 'DATA'
BUFFER          DB          8 DUP(?)
LastAD          DB          0                   ;上一次AD转换值
_DATA           ENDS

CODE            SEGMENT
START           PROC        NEAR
                ASSUME      CS:CODE, DS:_DATA, SS:_STACK
                MOV         AX,_DATA
                MOV         DS,AX
                NOP
                CALL        InitKeyDisplay      ;初始化键盘数码管控制器（8255）
                XOR         AL,AL
                JMP         START6
START1:         MOV         CX,8                ;采样8次
                MOV         BX,0                ;累计8次的采样值
START2:         CALL        AD0809
                XOR         AH,AH
                ADD         BX,AX
                LOOP        START2
                MOV         AX,8
                XCHG        AX,BX
                DIV         BL                  ;8次的平均值
                CMP         AL,LastAD
                JZ          START3
START6:         MOV         LastAD,AL
                CALL        Display_Data
START3:         CALL        DLTime
                JMP         START1
AD0809          PROC        NEAR
                PUSH        CX
                MOV         AL,0
                MOV         DX,ADDR_0809
                OUT         DX,AL
                MOV         CX,100
                LOOP        $                   ;延时,等待AD转换完成
                MOV         DX,ADDR_0809
                IN          AL,DX
                POP         CX
                RET
AD0809          ENDP
DISPLAY_DATA    PROC        NEAR
```

```
                MOV             AH,AL
                AND             AL,0FH
                MOV             BUFFER + 4,AL
                MOV             AL,AH
                AND             AL,0F0H
                ROR             AL,4
                MOV             BUFFER + 5,AL
                MOV             AL,AH
                XOR             AH,AH
                MOV             BL,51                   ;255/51 (16进制的1 = 1/51V)
                DIV             BL
                OR              AL,80H                  ;加上小数点
                MOV             BUFFER + 2,AL
                MOV             AL,10
                MUL             AH
                DIV             BL
                MOV             BUFFER + 1,AL           ;第一位小数
                MOV             AL,10
                MUL             AH
                DIV             BL
                MOV             BUFFER,AL               ;第二位小数
                MOV             buffer+3,10H
                MOV             buffer+6,10H
                MOV             buffer+7,10H            ;消隐
                RET
DISPLAY_DATA    ENDP
DLTime          PROC            NEAR
                MOV             CX,10
                LEA             SI,buffer
XX:             CALL            Display8
                LOOP            XX
                RET
DLTime          ENDP
START           ENDP
CODE            ENDS
                END             START
```

七、实验扩展及思考

如何实现多路模拟量的数据采集、显示？

12864 液晶显示实验

一、实验目的与要求

了解图形液晶模块的控制方法；了解它与 8086 的接口逻辑；掌握使用图形点阵液晶显示字体和图形。

二、实验设备

SUN 系列实验仪一套、PC 机一台。

三、实验内容

1、12864J 液晶显示器

(1) 图形点阵液晶显示器，分辨率为 128X64。可显示图形和 8×4 个(16×16 点阵)汉字。

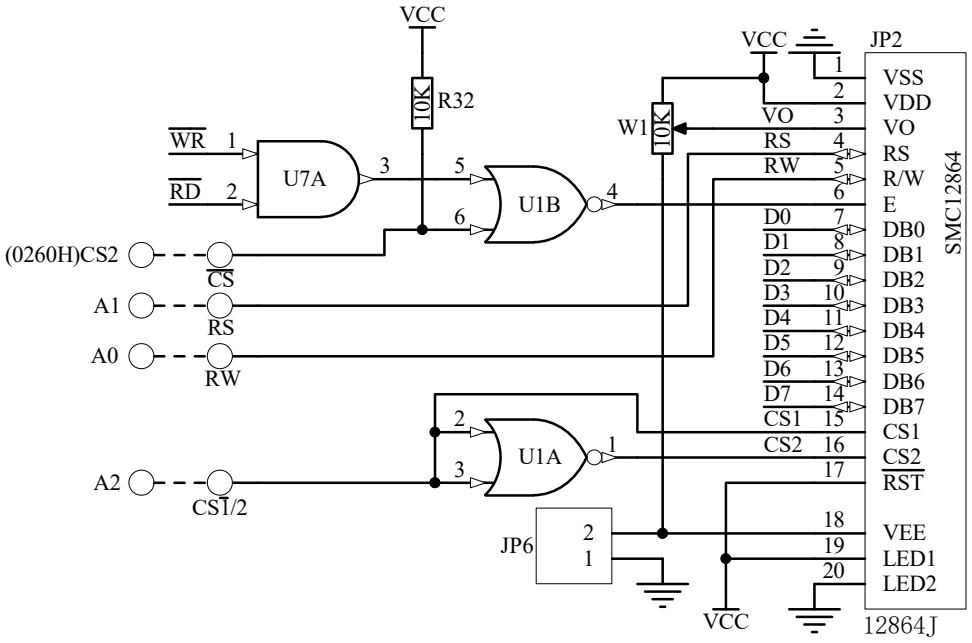
(2) 采用 8 位数据总线并行输入输出和 4 条控制线。

(3) 指令简单，7 种指令

2、实验过程

在 12864J 液晶上显示一段字，包括汉字和英文：“星研电子”、“STAR ES51PRO”、“欢迎使用”，三行字。

四、实验原理图



五、实验步骤

1、主机连线说明：

A1 区：CS、RW、RS、CS1/2	——	A3 区：CS2、A0、A1、A2

2、运行程序，验证显示结果。

六、演示程序

```
WR_COM_AD_L     EQU         0264H               ;写左半屏指令地址
WR_COM_AD_R     EQU         0260H               ;写右半屏指令地址
```

```
WR_DATA_AD_L    EQU         0266H               ;写左半屏数据地址
WR_DATA_AD_R    EQU         0262H               ;写右半屏数据地址
RD_BUSY_AD      EQU         0261H               ;查忙地址
RD_DATA_AD      EQU         0263H               ;读数据地址
X               EQU         0B8H                ;起始显示行基址
Y               EQU         040H                ;起始显示列基址
FirstLine       EQU         0C0H                ;起始显示行
_STACK          SEGMENT     STACK
                DW          100 DUP(?)
_STACK          ENDS
_DATA           SEGMENT     WORD PUBLIC 'DATA'
;--  文字:  星  --
Line1_1 DB      00H,00H,0FCH,82H,82H,0AAH,2AH,0AAH,0AAH,0AAH,02AH,  02H,02H,0FCH,00H,00H
        DB      00H,0EEH,9BH,90H,98H,94H,95H,80H,80H,80H,95H,95H,95H,95H,0FFH,00H
;--  文字:  研  --
Line1_2 DB      9EH,62H,02H,02H,02H,32H,0FEH,62H,02H,02H,32H,02H,02H,02H,62H,0DCH
        DB      03H,3CH,40H,40H,46H,40H,0F1H,8EH,80H,40H,7CH,80H,80H,80H,0FEH,03H
;--  文字:  电  --
Line1_3 DB      00H,0F8H,04H,04H,44H,44H,06H,02H,02H,46H,44H,04H,04H,0F8H,00H,00H
        DB      00H,0FH,10H,10H,11H,11H,0F0H,80H,90H,91H,91H,8CH,84H,87H,0C8H,78H
;--  文字:  子  --
Line1_4 DB      80H,40H,5EH,52H,52H,52H,32H,72H,82H,82H,42H,62H,52H,4CH,0C0H,00H
        DB      07H,04H,04H,04H,0FCH,8CH,8CH,80H,80H,7CH,04H,04H,04H,04H,07H,00H
;"STARES51PRO"
Line2_1 DB    00H,70H,88H,08H,  08H,08H,38H,00H,00H,38H,20H,21H,21H,22H,1CH,00H
Line2_2 DB    18H,08H,08H,0F8H,08H,08H,18H,00H,00H,00H,20H,3FH,20H,00H,00H,00H
Line2_3 DB    00H,00H,0C0H,38H,0E0H,00H,00H,00H,20H,3CH,23H,02H,02H,27H,38H,20H
Line2_4 DB    08H,0F8H,88H,88H,88H,88H,70H,00H,20H,3FH,20H,00H,03H,0CH,30H,20H
Line2_5 DB    00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
Line2_6 DB    08H,0F8H,88H,88H,0E8H,08H,10H,00H,20H,3FH,20H,20H,23H,20H,18H,00H
Line2_7 DB    00H,70H,88H,08H,08H,08H,38H,00H,00H,38H,20H,21H,21H,  22H,1CH,00H
Line2_8 DB    00H,0F8H,08H,  88H,88H,08H,08H,00H,00H,19H,21H,20H,20H,11H,0EH,00H
Line2_9 DB    00H,10H,10H,0F8H,00H,00H,00H,00H,00H,20H,20H,3FH,20H,20H,00H,00H
Line2_10 DB   08H,0F8H,08H,08H,08H,08H,0F0H,00H,20H,3FH,21H,01H,01H,01H,00H,00H
Line2_11 DB   08H,0F8H,88H,88H,88H,88H,70H,00H,20H,3FH,20H,00H,03H,0CH,30H,20H
Line2_12 DB   0E0H,10H,08H,08H,08H,10H,0E0H,00H,0FH,10H,20H,20H,20H,10H,0FH,00H
;--  文字:  欢  --
Line3_1 DB    14H,24H,44H,84H,64H,1CH,20H,18H,0FH,0E8H,08H,08H,28H,18H,08H,00H
        DB    20H,10H,4CH,43H,43H,2CH,20H,10H,0CH,03H,06H,18H,30H,60H,20H,00H
;--  文字:  迎  --
Line3_2 DB    40H,41H,0CEH,04H,00H,0FCH,04H,02H,02H,0FCH,04H,04H,04H,0FCH,00H,00H
        DB    40H,20H,1FH,20H,40H,47H,42H,41H,40H,5FH,40H,42H,44H,43H,40H,00H
;--  文字:  使  --
Line3_3 DB    40H,20H,0F0H,1CH,07H,0F2H,94H,94H,94H,0FFH,94H,94H,94H,0F4H,04H,00H
```

```
        DB    00H,00H,7FH,00H,40H,41H,22H,14H,0CH,13H,10H,30H,20H,61H,20H,00H
;--  文字:  用  --
Line3_4 DB    00H,00H,00H,0FEH,22H,22H,22H,22H,0FEH,22H,22H,22H,22H,0FEH,00H,00H
        DB    80H,40H,30H,0FH,02H,02H,02H,02H,0FFH,02H,02H,42H,82H,7FH,00H,00H
_DATA           ENDS
CODE            SEGMENT
START           PROC          NEAR
                ASSUME        CS:CODE, DS:_DATA, SS:_STACK
                MOV           AX,_DATA
                MOV           DS,AX
                NOP
START1:         CALL          LCD_INIT        ;液晶初始化
                CALL          DelayTime
                CALL          DisLine1        ;第2行显示”星研电子”
                CALL          DelayTime
                CALL          DisLine2        ;第3行显示”STAR ES51PRO”
                CALL          DelayTime
                CALL          DisLine3        ;第4行显示”欢迎使用”
                CALL          DelayTime
                JMP           START1
;延时程序
DelayTime       PROC          NEAR
                MOV           CX,0
                LOOP          $
                LOOP          $
                RET
DelayTime       ENDP
;第2行显示”星研电子”
DisLine1        PROC          NEAR
                LEA           SI,Line1_1
                MOV           AL,2            ;A-起始显示行地址，第2行
                MOV           AH,32           ;B-起始显示列地址，第32列，以下同
                CALL          WordDISL        ;左半屏，显示一个字子程序
                LEA           SI,Line1_2
                MOV           AL,2
                MOV           AH,48
                CALL          WordDISL
                LEA           SI,Line1_3
                MOV           AL,2
                MOV           AH,0
                CALL          WordDISR        ;右半屏，显示一个字子程序
                LEA           SI,Line1_4
                MOV           AL,2
                MOV           AH,16
```

```
                CALL        WordDISR
                RET
DisLine1        ENDP
;第3行显示”STAR ES51PRO”
DisLine2        PROC        NEAR
                LEA         SI,Line2_1
                MOV         AL,4            ;A-起始显示行地址，第4行
                MOV         AH,16           ;B-起始显示列地址，第16列，以下同
                CALL        ByteDISL        ;左半屏，显示一个字节子程序
                LEA         SI,Line2_2
                MOV         AL,4
                MOV         AH,24
                CALL        ByteDISL
                LEA         SI,Line2_3
                MOV         AL,4
                MOV         AH,32
                CALL        ByteDISL
                LEA         SI,Line2_4
                MOV         AL,4
                MOV         AH,40
                CALL        ByteDISL
                LEA         SI,Line2_5
                MOV         AL,4
                MOV         AH,48
                CALL        ByteDISL
                LEA         SI,Line2_6
                MOV         AL,4
                MOV         AH,56
                CALL        ByteDISL
                LEA         SI,Line2_7
                MOV         AL,4
                MOV         AH,0
                CALL        ByteDISR        ;右半屏字节显示数据
                LEA         SI,Line2_8
                MOV         AL,4
                MOV         AH,8
                CALL        ByteDISR
                LEA         SI,Line2_9
                MOV         AL,4
                MOV         AH,16
                CALL        ByteDISR
                LEA         SI,Line2_10
                MOV         AL,4
                MOV         AH,24
```

```
                CALL        ByteDISR
                LEA         SI,Line2_11
                MOV         AL,4
                MOV         AH,32
                CALL        ByteDISR
                LEA         SI,Line2_12
                MOV         AL,4
                MOV         AH,40
                CALL        ByteDISR
                RET
DisLine2        ENDP
;第4行显示”欢迎使用”
DisLine3        PROC        NEAR
                LEA         SI,Line3_1
                MOV         AL,6            ;A-起始显示行地址，第6行
                MOV         AH,32           ;B-起始显示列地址，第32列，以下同
                CALL        WordDISL        ;左半屏，显示一个字子程序
                LEA         SI,Line3_2
                MOV         AL,6
                MOV         AH,48
                CALL        WordDISL
                LEA         SI,Line3_3
                MOV         AL,6
                MOV         AH,0
                CALL        WordDISR        ;右半屏，显示一个字子程序
                LEA         SI,Line3_4
                MOV         AL,6
                MOV         AH,16
                CALL        WordDISR
                RET
DisLine3        ENDP
;液晶初始化
LCD_INIT        PROC        NEAR
                MOV         AL,3EH          ;初始化左半屏，关显示
                CALL        WRComL          ;写指令子程序
                MOV         AL,FirstLine    ;设置起始显示行，第0行
                CALL        WRComL
                MOV         AL,3EH          ;初始化右半屏，关显示
                CALL        WRComR          ;写指令子程序
                MOV         AL,FirstLine    ;设置起始显示行，第0行
                CALL        WRComR
                CALL        LCDClear        ;清屏
                MOV         AL,3FH          ;开显示
                CALL        WRComL
```

```
                MOV             AL,3FH          ;开显示
                CALL            WRComR
                RET
LCD_INIT        ENDP
;清屏
LCDClear        PROC            NEAR
                ;清左半屏
                MOV             AL,0            ;起始行，第0行
                MOV             AH,0            ;起始列，第0列
LCDClearL1:     PUSH            AX
                MOV             CX,64
                CALL            SETXYL          ;设置起始显示行列地址
LCDClearL2:     MOV             AL,0
                CALL            WRDATAL
                LOOP            LCDClearL2
                POP             AX
                INC             AX
                CMP             AL,8            ;共8行
                JNZ             LCDClearL1
                ;清右半屏
                MOV             AL,0            ;起始行，第0行
                MOV             AH,0            ;起始列，第0列
LCDClearR1:     PUSH            AX
                MOV             CX,64
                CALL            SETXYR          ;设置起始显示行列地址
LCDClearR2:     XOR             AL,AL
                CALL            WRDATAR
                LOOP            LCDClearR2
                POP             AX
                INC             AL
                CMP             AL,8            ;共8行
                JNZ             LCDClearR1
                RET
LCDClear        ENDP
;显示字体，显示一个数据要占用X行两行位置
;左半屏显示一个字节/字：AL-起始显示行序数X(0-7)；AH-起始显示列序数Y(0-63)；SI-显示字
数据首地址
ByteDisL        PROC            NEAR
                MOV             CX,8        ;显示8个字节数据，用于显示一个英文/符号
                CALL            DispL
                RET
ByteDisL        ENDP
WordDisL        PROC            NEAR
                MOV             CX,16           ;显示16字节数据，用于显示一个汉字
```

```
                CALL        DispL
                RET
WordDisL        ENDP
DispL           PROC        NEAR
                PUSH        AX
                PUSH        CX
                CALL        SETXYL          ;设置起始显示行列地址
                CALL        DisplayL        ;显示上半行数据
                POP         CX
                POP         AX
                INC         AL
                CALL        SETXYL          ;设置起始显示行列地址
                CALL        DisplayL        ;显示下半行数据
                RET
DispL           ENDP
;右半屏显示一个字节/字：AL-起始显示行序数X(0-7)； AH-起始显示列序数Y(0-63)； SI-显示字
数据首地址
ByteDisR        PROC        ENAR
                MOV          CX,8       ;显示8个字节数据，用于显示一个英文/符号
                CALL        DispR
                RET
ByteDisR        ENDP
WordDisR        PROC        NEAR
                MOV         CX,16           ;显示16字节数据，用于显示一个汉字
                CALL        DispR
                RET
WordDisR        ENDP
DispR           PROC        NEAR
                PUSH        AX
                PUSH        CX
                CALL        SETXYR          ;设置起始显示行列地址
                CALL        DisplayR        ;显示上半行数据
                POP         CX
                POP         AX
                INC         AL
                CALL        SETXYR          ;设置起始显示行列地址
                CALL        DisplayR        ;显示下半行数据
                RET
DispR           ENDP
;显示图形
;显示左半屏一行图形,AL-X起始行序数(0-7)， AH-Y起始列地址序数(0-63)
LineDisL        PROC        NEAR
                MOV         CX,64
                CALL        SETXYL          ;设置起始显示行列
```

```
                CALL            DisplayL        ;显示数据
                RET
LineDisL        ENDP
;显示右半屏一行图形,AL-X起始行地址序数(0-7)，AH-Y起始列地址序数(0-63)
LineDisR        PROC            NEAR
                MOV             CX,64
                CALL            SETXYR          ;设置起始显示行列
                CALL            DisplayR        ;显示数据
                RET
LineDisR        ENDP
;基本控制
;显示左半屏数据，R7-显示数据个数
DisplayL        PROC            NEAR
                LODSB
                CALL            WRDataL         ;写左半屏数据
                LOOP            DisplayL
                RET
DisplayL        ENDP
;显示右半屏数据，R7-显示数据个数
DisplayR        PROC            NEAR
                LODSB
                CALL            WRDataR         ;写左半屏数据
                LOOP            DisplayR
                RET
DisplayR        ENDP
;设置左半屏起始显示行列地址,AL-X起始行序数(0-7)，AH-Y起始列序数(0-63)
SETXYL          PROC            NEAR
                OR              AL,X            ;行地址=行序数+行基址
                CALL            WRComL
                MOV             AL,AH
                OR              AL,Y            ;列地址=列序数+列基址
                CALL            WRComL
                RET
SETXYL          ENDP
;设置右半屏起始显示行列地址,AL-X起始行序数(0-7)，AH-Y起始列序数(0-63)
SETXYR          PROC            NEAR
                OR              AL,X            ;行地址=行序数+行基址
                CALL            WRComR
                MOV             AL,AH
                OR              AL,Y            ;列地址=列序数+列基址
                CALL            WRComR
                RET
SETXYR          ENDP
;写左半屏控制指令，A-写入指令
```

```
WRComL          PROC        NEAR
                MOV         DX,WR_COM_AD_L
                OUT         DX,AL
WRComL1:        MOV         DX,RD_BUSY_AD
                IN          AL,DX
                TEST        AL,80H          ;检查液晶显示是否处于忙状态
                JNZ         WRComL1
                RET
WRComL          ENDP
;写右半屏控制指令，A-写入指令
WRComR          PROC        NEAR
                MOV         DX,WR_COM_AD_R
                OUT         DX,AL
WRComR1:        MOV         DX,RD_BUSY_AD
                IN          AL,DX
                TEST        AL,80H          ;检查液晶显示是否处于忙状态
                JNZ         WRComR1
                RET
WRComR          ENDP
;写左半屏数据，A-写入数据
WRDataL         PROC        NEAR
                MOV         DX,WR_DATA_AD_L
                OUT         DX,AL
WRDataL1:       MOV         DX,RD_BUSY_AD
                IN          AL,DX
                TEST        AL,80H          ;检查液晶显示是否处于忙状态
                JNZ         WRDataL1
                RET
WRDataL         ENDP
;写右半屏数据，A-写入数据
WRDataR         PROC        NEAR
                MOV         DX,WR_DATA_AD_R
                OUT         DX,AL
WRDataR1:       MOV         DX,RD_BUSY_AD
                IN          AL,DX
                TEST        AL,80H          ;检查液晶显示是否处于忙状态
                JNZ         WRDataR1
                RET
WRDataR         ENDP
START           ENDP
CODE            ENDS
                END         START
```

七、实验扩展及思考

实验内容：显示一幅图画，进一步熟练液晶显示的操作。

步进电机实验

一、实验目的与要求

1、了解步进电机的基本原理，掌握步进电机的转动编程方法

2、了解影响电机转速的因素有那些

二、实验设备

STAR 系列实验仪一套、PC 机一台。

三、实验内容

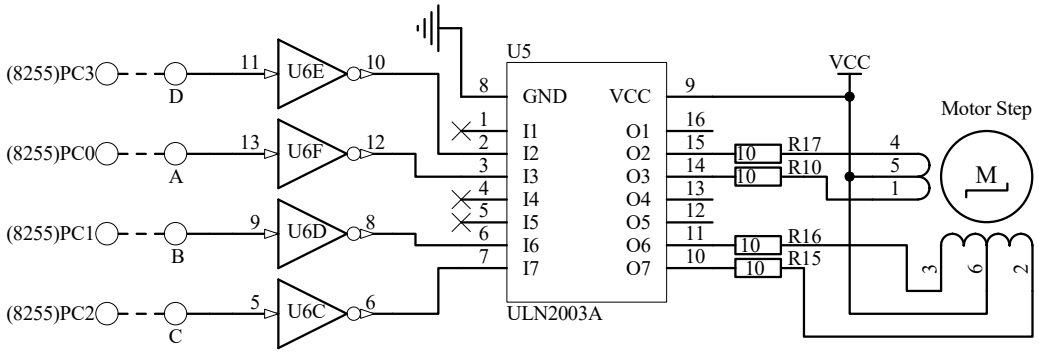
编写程序：使用 F5 区的键盘控制步进电机的正反转、调节转速，连续转动或转动指定步数；将相应的数据显示在 F5 区的数码管上。

四、控制原理

步进电机的驱动原理是通过它每相线圈的电流的顺序切换来使电机作步进式旋转，驱动电路由脉冲来控制，所以调节脉冲的频率便可改变步进电机的转速，微控制器最适合控制步进电机。另外，由于电机的转动惯量的存在，其转动速度还受驱动功率的影响，当脉冲的频率大于某一值（本实验为 f.>100hz）时，电机便不再转动。

实验电机共有四个相位（A,B,C,D），按转动步骤可分单 4 拍（A->B->C->D->A）,双 4 拍（AB->BC->CD->DA->AB）和单双 8 拍（A->AB->B->BC->C->CD->D->DA->A）.

五、实验原理图



六、实验步骤

1、主机连线说明：

D1 区：A、B、C、D	——	D3 区：PC4、PC5、PC6、PC7
D3 区：CS、A0、A1	——	A3 区：CS1、A0、A1
D3 区：PC0、PC1	——	F5 区：KL1、KL2
D3 区：JP20(PB)、B、C	——	F5 区：A、B、C
B3 区：CS、A0	——	A3 区：CS3、A0
B3 区：INT、INTA	——	A3 区：INTR、INTA
C4 区：CS(8253)、A0、A1	——	A3 区：CS2、A0、A1
C4 区：GATE	——	C1 区：VCC
C4 区：CLK0	——	B2 区：1M
C4 区：OUT0	——	B3 区：IR0

2、调试程序，查看运行结果是否正确

七、演示程序（完整程序见目录 STEP）

1、步进电机控制程序（STEP.ASM）

```
EXTRN           InitKeyDisplay:NEAR, GetKeyA:NEAR, DisPlay8:NEAR
```

```
IO8259_0        EQU     0250H
IO8259_1        EQU     0251H
Con_8253        EQU     0263H
T0_8253         EQU     0260H
IO8255_Con      EQU     0273H               ;CS3
IO8255_PC       EQU     0272H

_STACK          SEGMENT STACK
                DW      100 DUP(?)
_STACK          ENDS
_DATA           SEGMENT WORD PUBLIC 'DATA'
StepControl     DB      0                   ;下一次送给步进电机的值
buffer          DB      8 DUP(0)            ;显示缓冲区，8个字节
SpeedNo         DB      0                   ;选择哪一级速度
StepDelay       DB      0                   ;转动一步后，延时常数
StartStepDelay  DB      0;若选择速度过快,延时由长到短,最终使用对应延时常数
StartStepDelay1 DB      0                   ;StartStepDelay
bFirst          DB      0                   ;有没有转动过步进电机
bClockwise      DB      0                  ; =1 顺时针方向  =0 逆时针方向转动
bNeedDisplay    DB      0                   ;已转动一步，需要显示新步数
StepCount       DW      0                   ;需要转动的步数
StepDelayTab:   DB      250,125,83,62,50,42,36,32,28,25,22,21
_DATA           ENDS

CODE            SEGMENT WORD PUBLIC 'CODE'
START           PROC    NEAR
                ASSUME  CS:CODE, DS:_DATA, SS:_STACK
                MOV     AX,_DATA
                MOV     DS,AX
                MOV     ES,AX
                NOP
                CALL    InitKeyDisplay  ;初始化键盘、数码管控制器（8255）
                MOV     bFirst,1        ;有没有转动过步进电机
                MOV     bClockwise,1    ;顺时针方向

                MOV     StepControl,33H ;下一次送给步进电机的值
                MOV     SpeedNo,5       ;第五级速度
                CALL    Init8255
                CALL    Init8253
                CALL    Init8259
                CALL    WriIntver
                MOV     buffer,0        ;显示缓冲器初始化
                MOV     buffer+1,0
                MOV     buffer+2,0
```

```
                MOV     buffer+3,0
                MOV     buffer+4,10H
                MOV     AL,SpeedNo
                MOV     buffer+5,AL
                MOV     buffer+6,10H
                MOV     buffer+7,0
STAR2:          LEA     SI,buffer
                CALL    Display8
STAR3:          CALL    GetKeyA
                JB      STAR5
                CMP     bNeedDisplay,0
                JZ      STAR3
                MOV     bNeedDisplay,0
                CALL    Adjust_Step
                JMP     STAR2
STAR5:          CLI                     ;终止步进电机转动
                CMP     AL,10
                JNB     STAR1
                MOV     AH,buffer+2
                MOV     buffer+3,AH
                MOV     AH,buffer+1
                MOV     buffer+2,AH
                MOV     AH,buffer
                MOV     buffer+1,AH
                MOV     buffer,AL
                JMP     STAR2
STAR1:          CMP     AL,14
                JNB     STAR2
                LEA     SI,DriverTab
                SUB     AL,10
                SHL     AL,1
                XOR     AH,AH
                MOV     BX,AX
                JMP     CS:[SI+BX]
DriverTab:      DW      Direction       ;转动方向
                DW      Speed_up        ;提高转速
                DW      Speed_Down      ;降低转速
                DW      Exec            ;步进电机根据方向、转速、步数开始转动
Direction:      CMP     bClockwise,0
                JZ      Clockwise
                MOV     bClockwise,0
                MOV     buffer+7,1
AntiClockwise:  CMP     bFirst,0
                JZ      AntiClockwise1
```

```
                MOV     StepControl,91H
                JMP     Direction1
AntiClockwise1: MOV     AL,StepControl
                ROR     AL,2
                MOV     StepControl,AL
                JMP     Direction1
Clockwise:      MOV     bClockwise,1
                MOV     buffer+7,0
                CMP     bFirst,0
                JZ      Clockwise1
                MOV     StepControl,33H
                JMP     Direction1
Clockwise1:     MOV     AL,StepControl
                ROL     AL,2
                MOV     StepControl,AL
Direction1:     JMP     STAR2
Speed_up:       MOV     AL,SpeedNo
                CMP     AL,11
                JZ      Speed_up2
Speed_up1:      INC     AL
                MOV     SpeedNo,AL
                MOV     buffer+5,AL
Speed_up2:      JMP     STAR2
Speed_Down:     MOV     AL,SpeedNo
                CMP     AL,0
                JZ      Speed_Down1
                DEC     AL
                MOV     SpeedNo,AL
                MOV     buffer+5,AL
Speed_Down1:    JMP     STAR2
Exec:           MOV     bFirst,0
                CALL    TakeStepCount
                LEA     BX,StepDelayTab
                MOV     AL,SpeedNo
                XLAT
                MOV     StepDelay,AL
                CMP     AL,50
                JNB     Exec1
                MOV     AL,50
Exec1:          MOV     StartStepDelay,AL
                MOV     StartStepDelay1,AL
                STI
                JMP     STAR2
TIMER0:         PUSH    AX
```

```
                PUSH    DX
                DEC     StartStepDelay
                JNZ     TIMER0_1
                MOV     AL,StartStepDelay1
                CMP     AL,StepDelay
                JZ      TIMER0_2
                DEC     AL
                MOV     StartStepDelay1,AL
TIMER0_2:       MOV     StartStepDelay,AL
                MOV     AL,StepControl
                MOV     DX,IO8255_PC
                OUT     DX,AL
                CMP     bClockwise,0
                JNZ     TIMER0_3
                ROR     AL,1
                JMP     TIMER0_4
TIMER0_3:       ROL     AL,1
TIMER0_4:       MOV     StepControl,AL
                CMP     StepCount,0
                JZ      TIMER0_1
                MOV     bNeedDisplay,1
                DEC     StepCount
                JNZ     TIMER0_1
                add     sp,8                ;小写部分不允许使用单步、单步进入命
                                            令
                popf
                cli
                pushf
                sub     sp,8
                nop
TIMER0_1:       MOV     DX,IO8259_0
                MOV     AL,20H
                OUT     DX,AL
                POP     DX
                POP     AX
                IRET
Adjust_Step     PROC    NEAR
                MOV     CX,4
                LEA     DI,buffer
                CLD
                MOV     AX,StepCount
                MOV     BX,10
Adjust_Step1:   XOR     DX,DX
                DIV     BX
```

```
                XCHG    AX,DX
                STOSB
                MOV     AX,DX
                LOOP    Adjust_Step1
                RET
Adjust_Step     ENDP
TakeStepCount   PROC    NEAR
                MOV     AL,buffer+3     ;转动步数送入StepCount
                MOV     BX,10
                MUL     BL
                ADD     AL,buffer+2
                MUL     BL
                ADD     AL,buffer+1
                ADC     AH,0
                MUL     BX
                ADD     AL,buffer
                ADC     AH,0
                MOV     StepCount,AX
                RET
TakeStepCount   ENDP
Init8255        PROC    NEAR
                MOV     DX,IO8255_Con
                MOV     AL,81H
                OUT     DX,AL           ;8255 PC输出
                DEC     DX
                MOV     AL,0FFH
                OUT     DX,AL           ;0FFH->8255 PC
                RET
Init8255        ENDP
Init8253        PROC    NEAR
                MOV     DX,Con_8253
                MOV     AL,35H
                OUT     DX,AL           ;计数器T0设置在模式2状态,BCD码计
                                        数
                MOV     DX,T0_8253
                MOV     AL,10H
                OUT     DX,AL
                MOV     AL,02H
                OUT     DX,AL           ;CLK0/210
                RET
Init8253        ENDP
Init8259        PROC    NEAR
                MOV     DX,IO8259_0
                MOV     AL,13H
```

```
                OUT         DX,AL
                MOV         DX,IO8259_1
                MOV         AL,08H
                OUT         DX,AL
                MOV         AL,09H
                OUT         DX,AL
                MOV         AL,0FEH
                OUT         DX,AL
                RET
Init8259        ENDP
WriIntver       PROC        NEAR
                PUSH        ES
                MOV         AX,0
                MOV         ES,AX
                MOV         DI,20H
                LEA         AX,TIMER0
                STOSW
                MOV         AX,CS
                STOSW
                POP         ES
                RET
WriIntver       ENDP

START           ENDP
CODE            ENDS
                END         START
```

2、键盘、数码管扫描程序(K_D.ASM)，请参阅基础实验九“8255键盘显示实验”

八、实验扩展及思考

1、怎样改变电机的转速？

2、通过实验找出电机转速的上限，如何能进一步提高最大转速？

3、怎样能使电机反转？

直流电机测速实验

一．实验目的

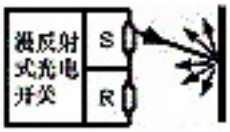
了解直流电机工作原理；了解光电开关的原理；掌握使用光电开关测量直流电机转速。

二．实验设备

SUN 系列实验仪一套、PC 机一台。

三．实验内容

1、转速测量原理：



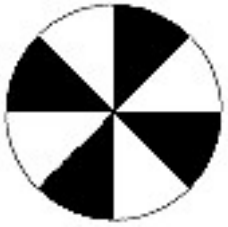


图 1 强反射　　图 2 弱反射　　图 3 转盘

本转速测量实验采用反射式光电开关，通过计数转盘通断光电开关产生的脉冲，计算出转速

(1) 反射式光开关工作原理：光电开关发射光，射到测量物体上，如果强反射，如图 1，光电开关接收到反射回来的光，则产生高电平 1；弱反射，如图 2，光电开关接收不到反射回来的光，则产生弱电平 0。

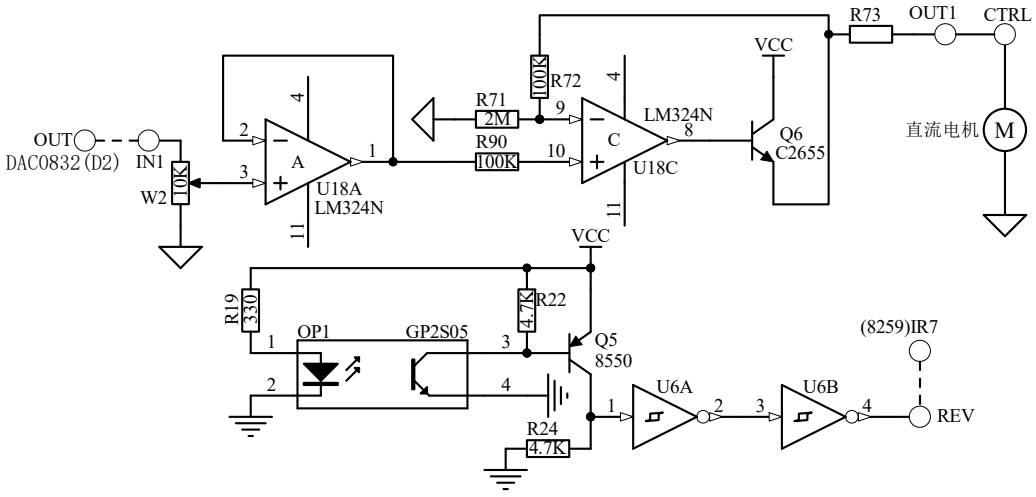
(2) 实验方法：本实验转速测量用的转盘在下表面做成如图 3 样子的转盘，白部分为强反射区，黑部分为弱反射区，转盘每转一圈，产生 4 个脉冲，每 1/4 秒计数出脉冲数，即得到每秒的转速。(演示程序中，LED 显示的是每秒钟转速)

2、实验过程

(1) 由 DAC0832 给电机供电，使用光电开关，测量电机转速，再经调整，最终将转速显示在 LED 上。

(2) 通过按键调节电机转速，随之变化的转速动态显示 LED 上

四．实验原理图



五、实验步骤

1、连线说明：

B3 区：CS、A0	——	A3 区：CS3、A0
B3 区：INT、INTA	——	A3 区：INTR、INTA
C4 区：CS、A0、A1	——	A3 区：CS2、A0、A1
C4 区：GATE	——	C1 区：VCC
C4 区：CLK0	——	B2 区：31250Hz
C4 区：CLK1	——	B2 区：1M
C4 区：OUT0	——	B3 区：IR0
D2 区：CS	——	A3 区：CS4
D2 区：OUT	——	F3 区：IN1
F3 区：OUT1	——	E1 区：CTRL
E1 区：REV	——	B3 区：IR7
D3 区：CS、A0、A1	——	A3 区：CS1、A0、A1
D3 区：JP20(PB)、B、C	——	F5 区：A、B、C
D3 区：PC0、PC1	——	F5 区：KL1、KL2

2、由 DAC0832 经功放电路驱动直流电机，计数光电开关通关次数并经过换算得出直流电机的转速，并将转速显示在 LED 上。

3、F5 区的 0、1 号按键控制直流电机转速快慢，（最大转速≈96r/s，5V，误差±1r/s）

六、演示程序

```
EXTRN           InitKeyDisplay:NEAR, DisPlay8:NEAR, GetKeyA:NEAR
IO8259_0        EQU         0250H
IO8259_1        EQU         0251H
Con_8253        EQU         0263H
T0_8253         EQU         0260H
T1_8253         EQU         0261H
DA0832          EQU         0240H
VoltageOffset   EQU         5                   ;0832 调整幅度

_STACK          SEGMENT     STACK
                DW          100 DUP(?)
_STACK          ENDS
_DATA           SEGMENT     WORD PUBLIC 'DATA'
buffer          DB          8 DUP(0)            ;显示缓冲区，8 个字节
buffer1         DB          8 DUP(0)            ;显示缓冲区，8 个字节
VOLTAGE         DB          0                   ;转换电压数字量
Count           DW          0                   ;一秒转动次数
NowCount        DW          0                   ;当前计数值
kpTime          DW          0                   ;保存上一次采样时定时器的值
bNeedDisplay    DB          0                   ;需要刷新显示
_DATA           ENDS

CODE            SEGMENT
START           PROC        NEAR
```

```
                ASSUME      CS:CODE, DS:_DATA, SS:_STACK
                MOV         AX,_DATA
                MOV         DS,AX
                MOV         ES,AX
                NOP
                CALL        InitKeyDisplay      ;对键盘、数码管扫描控制器 8255 初始化
                MOV         bNeedDisplay,1      ;显示初始值
                MOV         VOLTAGE,99H         ;初始化转换电压输入值，99H-3.0V
                MOV         Count,0             ;一秒转动次数
                MOV         NowCount,0          ;当前计数值
                MOV         kpTime,0            ;保存上一次采样时定时器的值
                CALL        DAC0832             ;初始 D/A
                CALL        Init8253
                CALL        Init8259
                CALL        WriIntver
                STI
MAIN:           CALL        GetKeyA             ;按键扫描
                JNB         Main1
                CMP         AL,00H
                JNZ         Key1
Key0:           MOV         AL,VoltageOffset;0 号键按下，转速提高
                ADD         AL,VOLTAGE
                CMP         AL,VOLTAGE
                JNB         Key0_1
                MOV         AL,0FFH             ;最大
Key0_1:         MOV         VOLTAGE,AL
                CALL        DAC0832             ;D/A
                JMP         Main2
Key1:           MOV         AL,VOLTAGE          ;1 号键按下，转速降低
                SUB         AL,VoltageOffset
                JNB         Key1_1
                XOR         AL,AL               ;最小
Key1_1:         MOV         VOLTAGE,AL
                CALL        DAC0832             ;D/A
                JMP         Main2
Main1:          CMP         bNeedDisplay,0
                JZ          MAIN
                MOV         bNeedDisplay,0      ;1s 定时到刷新转速
Main2:          CALL        RateTest            ;计算转速/显示
                JMP         MAIN                ;循环进行实验内容介绍与测速功能测试
;转速测量/显示
RateTest:       MOV         AX,Count
                MOV         BL,10
                DIV         BL
```

```
                CMP         AL,0
                JNZ         RateTest1
                MOV         AL,10H              ;高位为 0，不需要显示
RateTest1:      MOV         buffer,AH
                MOV         buffer+1,AL
                MOV         AL,VOLTAGE          ;给 0832 送的数据
                AND         AL,0FH
                MOV         buffer+4,AL
                MOV         AL,VOLTAGE
                AND         AL,0F0H
                ROR         AL,4
                MOV         buffer+5,AL
                MOV         buffer+2,10H        ;不显示
                MOV         buffer+3,10H
                MOV         buffer+6,10H
                MOV         buffer+7,10H
                LEA         SI,buffer
                CALL        Display8            ;显示转换结果
                RET
Timer0Int:      PUSH        AX
                PUSH        DX
                MOV         bNeedDisplay,1
                MOV         AX,NowCount
                SHR         AX,1
                SHR         AX,1
                MOV         Count,AX            ;转一圈产生 4 个脉冲,Count=NowCount/4
                MOV         NowCount,0
                MOV         DX,IO8259_0
                MOV         AL,20H
                OUT         DX,AL
                POP         DX
                POP         AX
                IRET
CountInt:       PUSH        AX
                PUSH        DX
                MOV         DX,Con_8253
                MOV         AL,40H
                OUT         DX,AL               ;锁存
                MOV         DX,T1_8253
                IN          AL,DX
                MOV         AH,AL
                IN          AL,DX
                XCHG        AL,AH               ;T1 的当前值
                XCHG        AX,kpTime
```

```
                SUB         AX,kpTime
                CMP         AX,100
                JB          CountInt1           ;前后二次采样时间差小于 100，判断是干扰
                INC         NowCount
CountInt1:      MOV         DX,IO8259_0
                MOV         AL,20H
                OUT         DX,AL
                POP         DX
                POP         AX
                IRET
Init8253        PROC        NEAR
                MOV         DX,Con_8253
                MOV         AL,34H
                OUT         DX,AL               ;计数器 T0 设置在模式 2 状态,HEX 计数
                MOV         DX,T0_8253
                MOV         AL,12H
                OUT         DX,AL
                MOV         AL,7AH
                OUT         DX,AL               ;CLK0=31250Hz,1s 定时
                MOV         DX,Con_8253
                MOV         AL,74H
                OUT         DX,AL               ;计数器 T1 设置在模式 2 状态,HEX 计数
                MOV         DX,T1_8253
                MOV         AL,0FFH
                OUT         DX,AL
                MOV         AL,0FFH
                OUT         DX,AL               ;作定时器使用
                RET
Init8253        ENDP
Init8259        PROC        NEAR
                MOV         DX,IO8259_0
                MOV         AL,13H
                OUT         DX,AL
                MOV         DX,IO8259_1
                MOV         AL,08H
                OUT         DX,AL
                MOV         AL,09H
                OUT         DX,AL
                MOV         AL,7EH
                OUT         DX,AL
                RET
Init8259        ENDP
WriIntver       PROC        NEAR
                PUSH        ES
```

```
                MOV         AX,0
                MOV         ES,AX
                MOV         DI,20H
                LEA         AX,Timer0Int
                STOSW
                MOV         AX,CS
                STOSW
                LEA         AX,CountInt
                MOV         DI,3CH
                STOSW
                MOV         AX,CS
                STOSW
                POP         ES
                RET
WriIntver       ENDP
;数模转换，A-转换数字量
DAC0832         PROC        NEAR
                MOV         DX,DA0832
                MOV         AL,VOLTAGE
                OUT         DX,AL
                RET
DAC0832         ENDP

START           ENDP
CODE            ENDS
                END         START
```

七．实验扩展及思考题

实验内容：在日光灯或白炽灯下，将转速调节到 25、50、75，观察转盘有什么现象出来

直流电机调速实验

一、实验目的

了解光电开关测速原理；掌握使用光电开关测量直流电机转速。

二、实验设备

SUN 系列实验仪一套、PC 机一台。

三、实验内容

1、转速控制原理

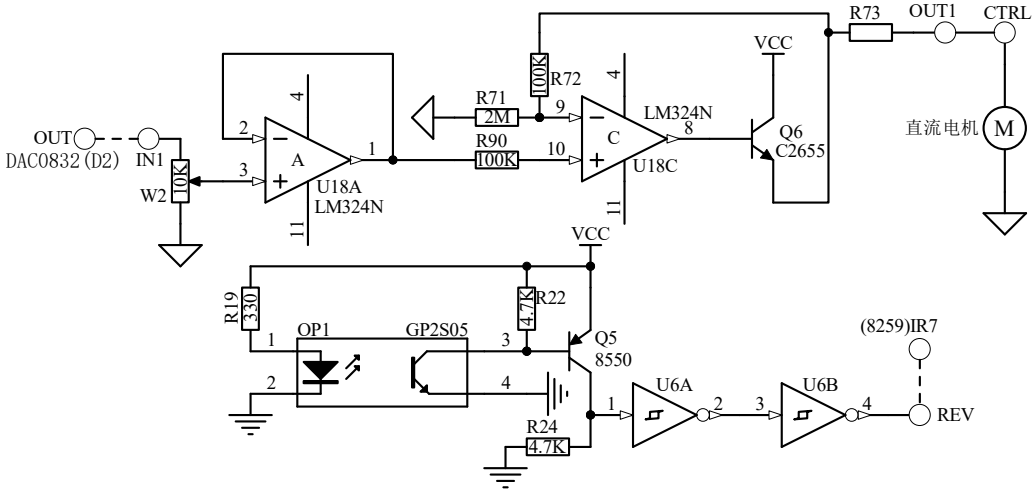
将设置的转速与当前测量的转速比较，得出差值用于控制 DAC0832 的输出电压，从而控制直流电机的转速，使转速逐渐达到设置转速。

2、实验过程

(1) 将当前转速与设置转速(要求达到的转速)相比较，得出差值来调整 DAC0832 的输出电压，逐步将转速控制到设置转速。

(2) 在 LED 上显示设置转速(左 4 位 LED)和当前转速(右 4 位 LED)，转速显示采用十进制。控制过程中，当前转速显示不断变化。　　＊　直流电机转速范围 3－99r/s,误差±1r/s

四、实验原理图



五、实验步骤

1、主机连线说明：

B3 区：CS、A0	———	A3 区：CS3、A0
B3 区：INT、INTA	———	A3 区：INTR、INTA
C4 区：CS、A0、A1	———	A3 区：CS2、A0、A1
C4 区：GATE	———	C1 区：VCC
C4 区：CLK0	———	B2 区：31250Hz
C4 区：CLK1	———	B2 区：1M
C4 区：OUT0	———	B3 区：IR0
D2 区：CS	———	A3 区：CS4
D2 区：OUT	———	F3 区：IN1
F3 区：OUT1	———	E1 区：CTRL
E1 区：REV	———	B3 区：IR7
D3 区：CS、A0、A1	———	A3 区：CS1、A0、A1
D3 区：JP20(PB)、B、C	———	F5 区：A、B、C

D3 区：PC0、PC1	——	F5 区：KL1、KL2

2、设置要求达到的转速，显示在 LED 上(左 4 位)；测量当前转速，显示在 LED 上(右 4 位)

3、比较设置转速与测量的当前转速，得出差值，用于调整 DAC0832 的输出电压，控制电机转速达到设置的转速。可以看到 LED 上显示的当前转速迅速靠近设置转速。

六．演示程序(完整程序 RateCtrl.asm)

1、基本控制程序

(1) DAC0832 程序请参阅“并行 DA 实验”；(2)键盘、数码管扫描可参阅“基础实验九”，也可以调用库函数。

2、主程序(MAIN.ASM)

(1) 转速测量

转速测量请参阅“直流电机测速实验”。（0.25s 测速一次，控制一次）

(2) 转速控制

```
REVControl    PROC    NEAR
              MOV     AL,Count            ;当前转速
              CMP     AL,RevSet           ;设置转速
              JZ      REVControl1
              JNB     RevDEC
;提高转速
RevINC:       MOV     Count500ms,1
              MOV     AL,RevSet           ;设置转速
              SUB     AL,Count            ;当前转速
RevINC1:      ADD     AL,Data_0832        ;转速差值+上一次 DAC0832 输入值=DAC0832 输入值
              JNB     RevINC2             ;判断是否超过 DAC0832 最大输入值
              MOV     AL,0FFH
RevINC2:      MOV     Data_0832,AL
              CALL    DAC0832             ;D/A,调整 DAC0832 输出电压
              JMP     REVControl1
;降低转速
RevDEC:       MOV     AL,Count            ;当前转速
              SUB     AL,RevSet           ;设置转速
              CMP     AL,40
              JNB     RevDEC1
              MOV     AH,Count500ms
              OR      AH,AH
              JZ      RevDEC3
              MOV     Count500ms,0
RevDEC1:      XCHG    AL,Data_0832        ;上一次 DAC0832 输入值-转速差值=DAC0832 输入值
              SUB     AL,Data_0832
              JNB     RevDEC2
              MOV     AL,10
RevDEC2:      MOV     Data_0832,AL
              CALL    DAC0832             ;D/A,调整 DAC0832 输出电压
              JMP     REVControl1
RevDEC3:      INC     Count500ms
```

```
REVControl1:  RET
REVControl    ENDP
```

七．实验扩展及思考题

实验内容：本实验采用差值法控制转速，现请使用其他的方法控制转速，实现更精确、快速的转速控制。

ISD1420 语音模块实验

一、实验目的

了解 ISD1420 的性能；与 8086 的接口逻辑；掌握手动和 CPU 控制两种录音、放音的基本功能。

二、实验设备

STAR 系列实验仪一套、PC 机一台。

三、实验内容

1、ISD1420 语音模块（B1 区）：

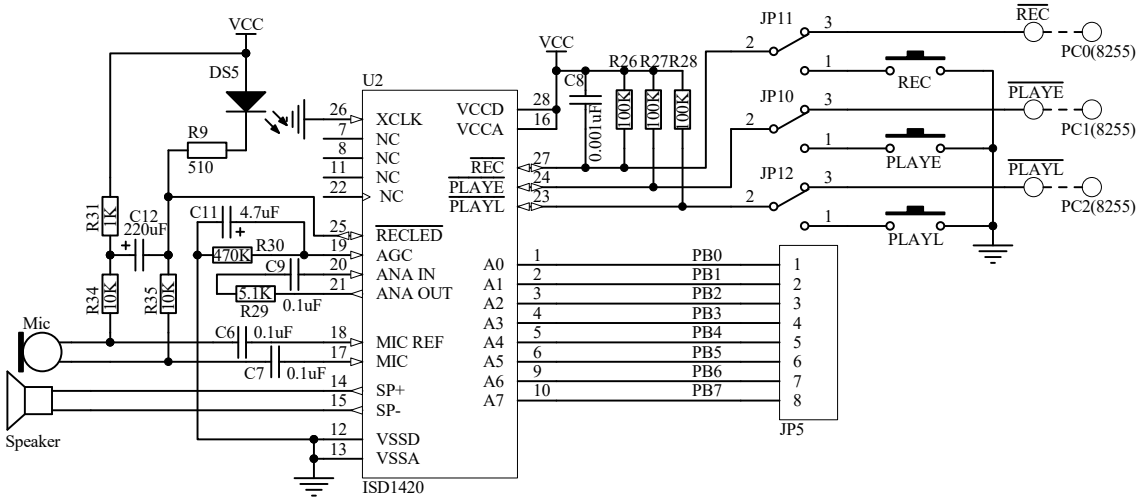
(1) 20 秒录放音长度，具有不掉电存储功能；(2) 可分 1－160 段录放音片段

2、具体操作

(1) 手动控制方式，通过 B1 区按键 REC 录音和按键 PLAYE、PLAYL 放音

(2) MCU 控制方式，通过 F5 区 8 个按键控制录、放音：0～3 号键录音各 5 秒；然后通过 4～7 号键放音，放音内容顺序对应 0～3 号键的录音内容

四．实验原理图



五．实验步骤

1、主机连线说明：

D3 区：CS(8255)、A0、A1	———	A3 区：CS1、A0、A1
D3 区：JP23(PA 口)	———	F5 区：A
F5 区：KL1	———	C1 区：GND
B1 区：REC	———	D3 区：PC0(8255) 录音控制
B1 区：PLAYE	———	D3 区：PC1(8255) 电平放音控制
B1 区：PLAYL	———	D3 区：PC2(8255) 触发放音控制，下降沿触发
B1 区：JP5	———	D3 区：JP20（PB 口）

2、将 JP10，JP11，JP12 跳向“MCU”，8086 控制，运行演示程序，0～3 号键录音，4～7 号键放音。

六．演示程序

```
ISD1420_AD0     EQU     00H             ;0号键录放音起始地址,每次录音5s
ISD1420_AD1     EQU     28H             ;1号键录放音起始地址
ISD1420_AD2     EQU     50H             ;2号键录放音起始地址
```

```
ISD1420_AD3     EQU         78H             ;3号键录放音起始地址
ISDCOMM         EQU         0271H  ;录放音地址/操作模式输入地址,由8255的PB口控制
I8255_Ctr       EQU         0273H           ;8255控制端口地址
I8255_PA        EQU         0270H           ;键盘数据输入口
I8255_PC        EQU         0272H           ;ISD1420控制输出口

_STACK          SEGMENT     STACK
                DW          100 DUP(?)
_STACK          ENDS
_DATA           SEGMENT     WORD PUBLIC 'DATA'
KeepMode        DB          7               ;保存REC、PLAYE、PLAYL当前值
bNewKey         DB          0               ;有键按下标志位，清0-无键按下
KEYno           DB          0
KeyTab          DW   KEY0, KEY1,KEY2,KEY3,KEY4,KEY5,KEY6,KEY7;录音键放音键子程序入口
_DATA           ENDS
CODE            SEGMENT
START           PROC        NEAR
                ASSUME      CS:CODE, DS:_DATA, SS:_STACK
                MOV         AX,_DATA
                MOV         DS,AX
                NOP
                CALL        MainInit        ;主程序初始化
Main:           CALL        ScanKey         ;扫描按键
                JNB         Main
Main1:          CALL        KeyRun          ;按键处理
                CMP         bNewKey,0       ;是否有新的键按下
                JZ          Main
                MOV         bNewKey,0       ;清按键标志
                JMP         Main1       ;循环进行实验内容介绍与ISD1420功能测试
;主程序初始化
MainInit        PROC        NEAR
                MOV         bNewKey,0       ;有键按下标志位，清0-无键按下
                MOV         DX,I8255_Ctr    ;8255初始化
                MOV         AL,90H          ;PA为输入，PC的低四位为输出
                OUT         DX,AL
                CALL        ISD_INIT
                RET
MainInit        ENDP
;录放音子程序
KEY0            PROC        NEAR
                MOV         AL,ISD1420_AD0    ;0号键录音首地址
                CALL        KEY_REC
                RET
KEY0            ENDP
```

```
KEY1            PROC        NEAR
                MOV         AL,ISD1420_AD1     ;1号键录音首地址
                CALL        KEY_REC
                RET
KEY1            ENDP
KEY2            PROC        NEAR
                MOV         AL,ISD1420_AD2     ;2号键录音首地址
                CALL        KEY_REC
                RET
KEY2            ENDP
KEY3            PROC        NEAR
                MOV         AL,ISD1420_AD3     ;3号键录音首地址
                CALL        KEY_REC
                RET
KEY3            ENDP
;录音子程序
KEY_REC         PROC        NEAR
                MOV         CX,20              ;录音时间长度,5s
                CALL        ISD_REC            ;调用录音子程序
KEY_REC1:       CALL        Delay_025S         ;延时
                CMP         bNewKey,0          ;检测按键是否有键按下
                JNZ         KEY_REC2
                LOOP        KEY_REC1           ;录音时间，根据CX的值决定
                CALL        ISD_STOP           ;停止录音
KEY_REC2:       RET
KEY_REC         ENDP
;放音子程序
KEY4            PROC        NEAR
                MOV         AL,ISD1420_AD0     ;4号键放音首地址
                CALL        KEY_PLAY
                RET
KEY4            ENDP
KEY5            PROC        NEAR
                MOV         AL,ISD1420_AD1
                CALL        KEY_PLAY
                RET
KEY5            ENDP
KEY6            PROC        NEAR
                MOV         AL,ISD1420_AD2     ;6号键放音首地址
                CALL        KEY_PLAY
                RET
KEY6            ENDP
KEY7            PROC        NEAR
                MOV         AL,ISD1420_AD3     ;7号键放音首地址
```

```
                CALL        KEY_PLAY
                RET
KEY7            ENDP
KEY_PLAY        PROC        NEAR
                MOV         CX,20
                CALL        ISD_PLAY            ;调用录音子程序
KEY_PLAY1:      CALL        Delay_025S          ;用于进度显示的时间参照
                CMP         bNewKey,0
                JNZ         KEY_PLAY2           ;检测按键是否有键按下
                LOOP        KEY_PLAY1
KEY_PLAY2:      RET
KEY_PLAY        ENDP
KeyRun          PROC        NEAR
                LEA         BX,KeyTab           ;有键按下,跳到相应处理程序
                MOV         AL,KEYno            ;KEYno--按键值
                SHL         AL,1                ;×2倍
                XOR         AH,AH
                ADD         BX,AX
                CALL        [BX]                ;[BX]=对应按键子程序入口地址
                RET
KeyRun          ENDP
;按键扫描
ScanKey         PROC        NEAR
                MOV         DX,I8255_PA         ;8255.PA----检测按键输入
                IN          AL,DX               ;键扫描
                CMP         AL,0FFH
                JNZ         ScanKey1
ScanKey4:       CLC                             ;无按键按下
                RET
ScanKey1:       CALL        ScanKey2            ;有按键，取抖动处理
                JNB         ScanKey4
ScanKey3:       MOV         BL,KEYno
                CALL        Delay20ms           ;消抖动
                CALL        Delay20ms
                CALL        ScanKey2
                JNB         ScanKey4
                CMP         BL,KEYno
                JNZ         ScanKey3
ScanKey5:       MOV         DX,I8255_PA
ScanKey6:       IN          AL,DX
                CMP         AL,0FFH
                JNZ         ScanKey6
                STC
                RET
```

```
ScanKey         ENDP
;按下的键（1～8）转化为对应的键值（0～7），便于多分支程序处理
ScanKey2        PROC        NEAR
                PUSH        BX
                XOR         BL,BL
                MOV         DX,I8255_PA
                IN          AL,DX
                TEST        AL,01H
                JZ          ScanKey21
                INC         BL
                TEST        AL,02H
                JZ          ScanKey21
                INC         BL
                TEST        AL,04H
                JZ          ScanKey21
                INC         BL
                TEST        AL,08H
                JZ          ScanKey21
                INC         BL
                TEST        AL,10H
                JZ          ScanKey21
                INC         BL
                TEST        AL,20H
                JZ          ScanKey21
                INC         BL
                TEST        AL,40H
                JZ          ScanKey21
                INC         BL
                TEST        AL,80H
                JZ          ScanKey21
                CLC
                JMP         ScanKey22
ScanKey21:      STC                         ;有键按下，置有键按下标志
                MOV         KEYno,BL        ;获得键值
ScanKey22:      POP         BX
                RET
ScanKey2        ENDP
;延时
Delay20ms       PROC        NEAR
                PUSH        CX
                MOV         CX,2640
                LOOP        $
                POP         CX
                RET
```

```
Delay20ms       ENDP
;延时0.25s（兼有键盘检测功能）
Delay_025S      PROC        NEAR
                PUSH        AX
                PUSH        CX
                PUSH        DX
                MOV         CX,28000
                LOOP        $
                MOV         CX,28000
                LOOP        $
                CALL        ScanKey
                JNB         DL1S_2
                MOV         bNewKey,1
DL1S_2:         POP         DX
                POP         CX
                POP         AX
                RET
Delay_025S      ENDP
;录音子程序
;AL--存放操作方式设置值，CX--录几秒
ISD_INIT        PROC        NEAR
                MOV         DX,I8255_PC
                MOV         AL,KeepMode
                OR          AL,7                ;语音模块初始化，关闭录放音功能
                OUT         DX,AL
                MOV         KeepMode,AL
                MOV         DX,ISDCOMM
                XOR         AL,AL
                OUT         DX,AL           ;允许手动录放音,当A6,A7为高时，无法手动放音
                RET
ISD_INIT        ENDP
;操作模式，AL-操作模式设置值
ISD_MODE        PROC        NEAR
                PUSH        AX
                CALL        ISD_STOP        ;初始化,REC,PLAYE,PLAYL置位，设置操作模式
                MOV         DX,ISDCOMM          ;设置操作模式:分段录音
                POP         AX
                OUT         DX,AL               ;设置操作模式命令在AL中
                MOV         DX,I8255_PC
                MOV         AL,KeepMode
                AND         AL,0FBH
                OUT         DX,AL
                OR          AL,4
                OUT         DX,AL               ;给PLAYL一个上升沿,锁存命令
```

```
                MOV         KeepMode,AL
                RET
ISD_MODE        ENDP
;录音
ISD_REC         PROC        NEAR
                MOV         DX,ISDCOMM
                OUT         DX,AL               ;设置录音起始地址
                MOV         DX,I8255_PC
                MOV         AL,KeepMode
                AND         AL,0FEH
                OUT         DX,AL               ;REC变低，即开始录音
                MOV         KeepMode,AL
                RET
ISD_REC         ENDP
;放音子程序
;AL--放哪段音
ISD_PLAY        PROC        NEAR
                PUSH        AX
                CALL        ISD_STOP            ;暂停之前的录放音操作
                POP         AX
                MOV         DX,ISDCOMM          ;设置放音起始地址
                OUT         DX,AL
                MOV         DX,I8255_PC
                MOV         AL,KeepMode
                AND         AL,0FDH
                OUT         DX,AL               ;0->PLAYE 开始放音,边沿放音模式
                OR          AL,2
                OUT         DX,AL               ;1->PLAYE
                MOV         KeepMode,AL
                RET
ISD_PLAY        ENDP
;停止录放音
ISD_STOP        PROC        NEAR
                MOV         DX,I8255_PC
                MOV         AL,KeepMode
                AND         AL,0FBH
                OUT         DX,AL               ;PLAYL:一个负脉冲停止放音
                OR          AL,4
                OUT         DX,AL
                CALL        Delay50ms
                OR          AL,3                ;1->REC,PLAYE
                OUT         DX,AL               ;关闭所有操作指令
                MOV         KeepMode,AL
                MOV         DX,ISDCOMM
```

```
                XOR         AL,AL
                OUT         DX,AL           ;允许手动录放音,当A6,A7为高时，无法手动放音
                RET
ISD_STOP        ENDP
;延时
Delay50ms       PROC        NEAR
                PUSH        CX
                MOV         CX,13200
                LOOP        $
                POP         CX
                RET
Delay50ms       ENDP

START           ENDP
CODE            ENDS
                END         START
```

七．实验扩展及思考题

实验名称：公交车的报站功能

实验内容：利用掌握分段录音和放音控制，实现公交车的报站功能，有兴趣者可自行尝试