

接口与通讯实验 报告 2

Hollow Man

一、实验环境

一台带有装有 Windows XP 系统的实验室计算机，一台实验箱。

二、实验目的

(1) 学习在 PC 系统中扩展 I/O 接口的方法。掌握 8255 并行接口编程原理，体会可编程接口的设计灵活性、方便性。

(2) 学习编制数据输出程序的设计方法。

(3) 了解微机化竞赛抢答器的基本原理。

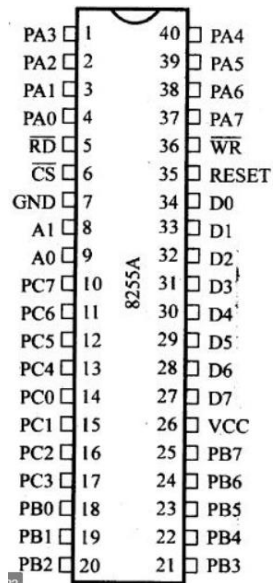
三、芯片介绍

8255 是一种通用的可编程并行 I/O 接口芯片，是专为 Intel 公司的微处理器设计的，也可用于其他系列的微型机系统中。利用 8086 汇编指令系统，编制初始化程序，可以变更 8255 的工作方式，通用性强，使用灵活。8255 具有 3 个带锁存或缓冲的数据端口，它的并行数据宽度为 8 位，可与外设并行进行数据交换。A 口和 B 口内具有中断控制逻辑，在外设与之间可用中断方式进行信息交换。8255 能与许多外部设备连接，如键盘、显示器和打印机等。

8255 是一个具有两个 8 位（A 口和 B 口）和两个 4 位（C 口高/低 4 位）并行 I/O 端口的可编程的接口芯片。PC 口除了数据口外，在方式 1 或方式 2，它的部分引脚被分配作为专用联络信号。它可以适用 CPU 和 I/O 接口之间进行多种数据传送方式的要求，如无条件传送、查询传送和中断传送。

8255 的编程命令有方式命令字，其作用就是指定 8255 的工作方式及其方式下 3 个并行口是作输入还是作输出。按位置位/复位的命令字，其作用就是指定 PC 口的某一位（或引脚）输出高电平或低电平。

另外，关于两个命令的初始化问题。方式命令是对 8255 的 3 个端口的工作方式及功能的指定，即初始化。初始化要在使用 8255 之前做。按位置位/复位命令只是对 PC 端口的输出进行控制，使用它并不破坏已建立的 3 种工作方式，而是对它们实现动态控制的一种支持。它可放在初始化程序以后的任何地方。两个命令的最高位 D7 都作了特征位，以便识别两个不同的命令。注意，按位置位/复位的命令字只能写入命令口，不能写入 C 口，如果写入 C 口，就会按 C 口的数据格式来处理。



三、实验内容

实验 1

将 C 口通过开关控制的数据输入 A 口，使得 A 口连接的 LED 灯亮起。

实验 2:

编写程序，以 8255C 口作为输出口，控制 8 个单色 LED。

四、实验原理

实验一

```

;*****
;* 8255 方式 0 的 C 口输入,A 口输出  *;
;*****
io8255a      equ 288h
io8255b      equ 28bh
io8255c      equ 28ah

code  segment
    assume cs:code
start: mov dx,io8255b      ;设 8255 为 C 口输入,A 口输出
        mov al,8bh
        out dx,al

inout:  mov dx,io8255c      ;从 C 口输入一数据
        in al,dx
        mov dx,io8255a      ;从 A 口输出刚才自 C 口
        out dx,al          ;所输入的数据
        mov dl,0ffh        ;判断是否有按键
        mov ah,06h
        int 21h
        jz inout          ;若无,则继续自 C 口输入,A 口输出
        mov ax,4c00h      ;否则返回
        int 21h

```

```
code ends
end start
```

实验二

8255 工作于方式 0,此时 PA、PB、PC 均为可独立输入输出的并行口。8255 的各寄存器对应的端口地址为:

PA 口: 288H
PB 口: 289H
PC 口: 28AH
控制口: 28BH

8255 的片选信号为 CS 连 138 译码器的 288H~28FH。由于各 PC 速度不同, 为到达较好的实验效果, 可以适当调节 LED 亮灭的延迟时间。

实现代码

```
DATA SEGMENT
MESS1 DB '左循环按 1',0DH,0AH,'$ '
      DB '右循环按 2',0DH,0AH,'$ '
      DB '间隔闪烁按 3',0DH,0AH,'$ '
      DB '退出按 4',0DH,0AH,'$ '
MESS2 DB '按无效键继续选择',0DH,0AH,'$ '
MESS3 DB '按任意键退回主菜单',0DH,0AH,'$ '
DATA ENDS
STACK SEGMENT
STA DB 20 DUP (?)
TOP EQU LENGTH STA
STACK ENDS
CODE SEGMENT
ASSUME CS:CODE,DS:DATA,SS:STACK,ES:DATA
START: MOV AX,DATA ;初始化
      MOV DS,AX
      MOV AX,STACK
      MOV SS,AX
      MOV SP,TOP
      MOV SP,AX
      MOV DX,108255D
      MOV AL,80H
      OUT DX,AL
BG:    LEA DX,MESS1
      MOV AH,09H
      INT 21H
      MOV AH,08H ;等待有键按下
      INT 21H
      PUSH AX
      LEA DX,MESS3
      MOV AH,09H
      INT 21H
      POP AX
      MOV AH,AL
      CMP AL,'1'
      JZ ZXH ;进入各种循环
      MOV AL,AH
      CMP AL,'2'
      JZ YXH
```

```

        MOV     AL, AH
        CMP     AL, '3'
        JZ      JGS
        MOV     AL, AH
        CMP     AL, '4'
        JNZ     WX
        JMT     QIT
WX:      LEA     DX, MESS2           ;无效按键提示信息
        MOV     AH, 09H
        INT     21H
        JMP     BG
ZXH:     MOV     DX, I08255C       ;左循环

        MOV     AL, OFEH
BB:      OUT     DX, AL
        MOV     CX, 0FFFFH
L1:      LOOP    L1
        ROL     AL, 1
        PUSH    AX
        PUSH    DX
        MOV     DL, 0FFH
        MOV     AH, 06H
        INT     21H
        JNZ     BG
        POP     DX
        POP     AX
        JMP     BB
YXH:     MOV     DX, I08255C       ;右循环
        MOV     AL, OFEH
EE:      OUT     DX, AL
        MOV     CX, 0FFFFH
L2:      LOOP    L2
        ROR     AL, 1
        PUSH    AX
        PUSH    DX
        MOV     DL, 0FFH
        MOV     AH, 06H
        INT     21H
        JNZ     BG
        POP     DX
        POP     AX
        JMP     EE
JGS:     MOV     DX, I08255C       ;间隔闪烁
        MOV     AL, 55H
HH:      OUT     DX, AL
        MOV     CX, 0FFFFH
L3:      LOOP    L3
        NOT     AL
        PUSH    AX
        PUSH    DX
        MOV     DL, 0FFH

```

```

        MOV     AH, 06H
        INT     21H
        JNZ     BG1
        POP     DX
        POP     AX
        JMP     HH
BG1:    JMP     BG
QIT:    MOV     AX, 4C00H
        INT     21H

CODE    ENDS
END     START

```

实验三

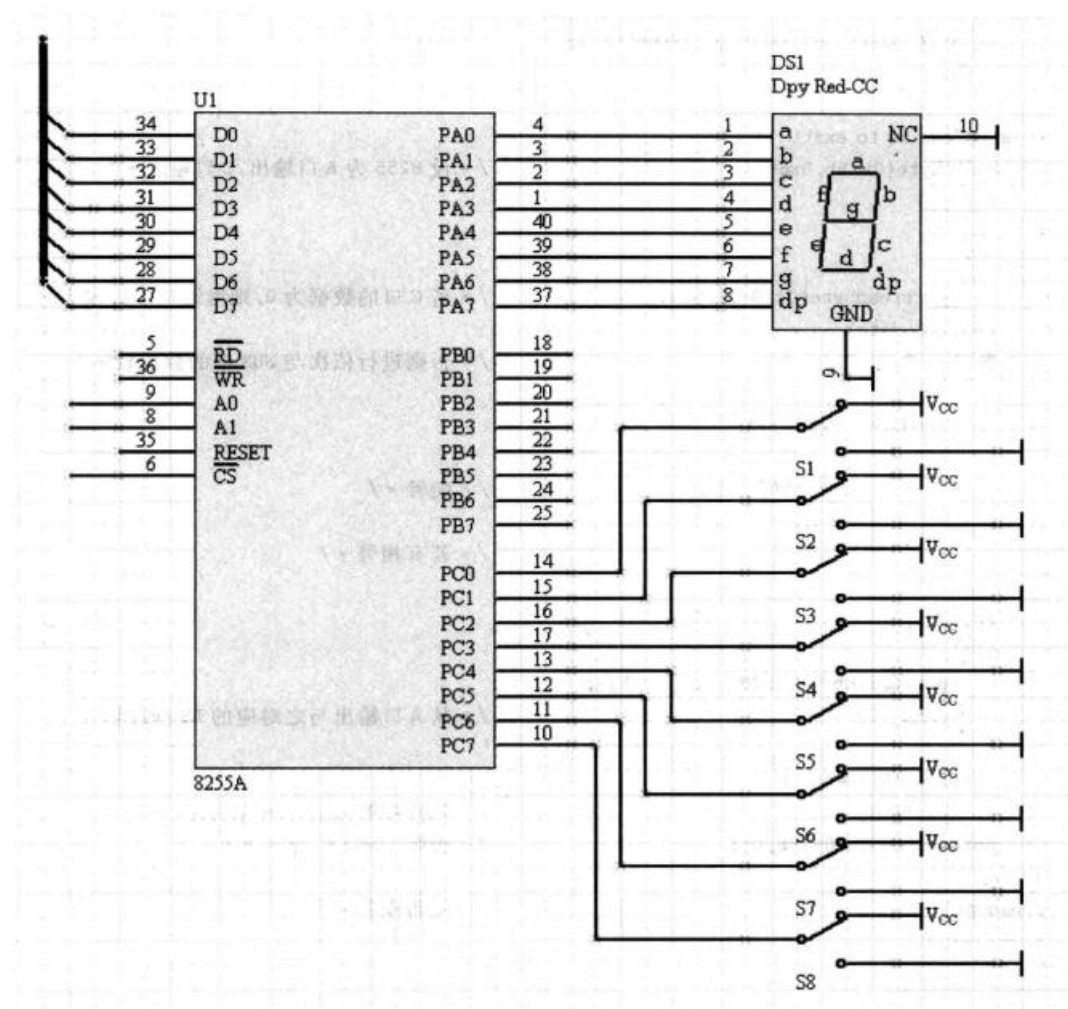
逻辑开关 K0~K7 代表竞赛抢答按钮 0~7 号，当某个逻辑电平开关置“1”时，相当某组抢答按钮按下。在七段数码管上将其组号(0~7)显示出来。置 8255 为 C 口输入、A 口输出，读取 C 口数据，若为 0 表示无人抢答，若不为 0 则有人抢答。根据读取数据可判断其组号。从键盘上按空格键开始下一轮抢答，按其他键程序退出。

进一步改编了程序，增加了键盘选择 m 进行键盘输入抢答，n 进行开关抢答。

流程图



抢答器电路图



实现代码

```

DATA    SEGMENT
LED      DB 3FH,06H,5BH,4FH,66H,6DH,7DH,07H      ;数码表
DATA     ENDS
CODE     SEGMENT
ASSUME CS:CODE,DS:DATA
START:   MOV     AX,DATA
          MOV     DS,AX
          MOV     DX,I08255D      ;设 8255 为 A 口输出,C 口输入
          MOV     AX,89H
          OUT     DX,AL
          MOV     BX,OFFSET LED   ;使 BX 指向段码管首址
SSS:     MOV     DX,I08255C
          IN      AL,DX           ;从 8255 的 C 口输入数据
          OR      AL,AL          ;比较是否为 0
          JE      SSS            ;若为 0,则表明无键按下,转 SSS
          MOV     CL,OFFH        ;CL 作计数器,初值为 -1
RR:      SHR     AL,1
          INC     CL
          JNC     RR
          MOV     AL,CL
          XLAT
          MOV     DX,I08255A
          OUT     DX,AL
          MOV     DL,7           ;响铃 ASCII 码为 07
          MOV     AH,2
          INT     21H
WAI:     MOV     AH,1
          INT     21H
          CMP     AL,20H         ;是否为空格
          JNE     EEE            ;不是,转 EEE
          MOV     AL,0           ;是,关灭灯
          MOV     DX,I08255A
          OUT     DX,AL
          JMP     SSS
EEE:     MOV     AH,4CH          ;返回
          INT     21H
CODE     ENDS
END      START

```

另附增加了键盘选择 m 进行键盘输入抢答，n 进行开关抢答的程序：

```

DATA SEGMENT
LED DB 3FH,06H,5BH,4FH,66H,6DH,7DH,07H,7FH,6FH
MESS DB 0DH,0AH,'PLEASE INPUT m OR n',0DH,0AH,'$'
DATA ENDS
CODE SEGMENT
ASSUME CS:CODE,DS:DATA
START:
MOV AX,DATA
MOV DS,AX
MOV DX,28BH
MOV AX,89H
OUT DX,AL
MOV BX,OFFSET LED
CMPP:

```

```
LEA DX,MESS
MOV AH,09H
INT 21H
MOV AH,1
INT 21H
CMP AL,6EH
JE SSS
CMP AL,6DH
JNE EEE
MOV AH,1
INT 21H
SUB AL,48
XLAT
MOV DX,288H
OUT DX,AL
MOV DL,7
MOV AH,2
INT 21H
JMP WAIK
SSS:
MOV DX,28AH
IN AL,DX
OR AL,AL
JE SSS
MOV CL,0FFH
RR:
SHR AL,1
INC CL
JNC RR
MOV AL,CL
XLAT
MOV DX,288H
OUT DX,AL
MOV DL,7
MOV AH,2
INT 21H
WAIK:
MOV AH,1
INT 21H
CMP AL,20H
JNE EEE
MOV AL,0
MOV DX,288H
OUT DX,AL
```



```
JMP CMPP  
EEE:  
MOV AH,4CH  
INT 21H  
CODE ENDS  
END START
```