



微机系统与接口课程设计

[实验报告]



2018-7-19

[欧宝源 计科七班 学号 2015211324]

[裴子祥 计科七班 学号 2015211921]

[指导老师：张杰 姚文斌]

目录

1.	实验目的	3
2.	实验内容	3
2.1.	硬件部分	3
2.2.	软件部分	3
3.	实验原理	3
3.1.	设计思想-模块设计	3
3.2.	实验原理图	6
3.3.	实验流程图	7
4.	实验结果	9
4.1.	单机通信模式	9
4.2.	双机通信模式	10
5.	问题解决及收获体会	10
5.1.	问题与解决	10
5.2.	收获与体会	11
6.	实验程序	11

北京邮电大学课程设计报告

课程设计 名称	微机系统与接口课程 设计	学院	计算机学院	指导教师	张杰
班级	班内序号	学号	学生姓名	成绩	
2015211307	14	2015211324	欧宝源		
2015211307	32	2015211921	裴子祥		
课 程 设 计 内 容	<p>实验内容：利用两台 TPC-ZK 微机原理实验平台实现单机与两机之间通信</p> <p>实验目的：</p> <ul style="list-style-type: none"> • 通过本实验了解串行通信原理； • 理解计算机软件、硬件如何配合工作，如何对计算机进行功能扩展； • 学会一般接口电路设计过程和驱动代码编写方法； • 学会如何进行软件、硬件调试。 <p>实验方法：HQFC 汇编编程和上机实验结合，小组合作</p> <p>团队分工：欧宝源负责键盘扫描、液晶屏显示、proteus 原理图绘制；裴子祥负责用 8251 发送输入的数据、数码管显示，与文档报告编写总和。两人共同完成中断部分对数据的接收和处理。</p>				
学 生 课 程 设 计 报 告 (附页)	详见课程设计报告。				
课 程 设 计 成 绩 评 定	<p>遵照实践教学大纲并根据以下四方面综合评定成绩：</p> <ol style="list-style-type: none"> 1、课程设计目的任务明确，选题符合教学要求，份量及难易程度 2、团队分工是否恰当与合理 3、综合运用所学知识，提高分析问题、解决问题及实践动手能力的效果 4、是否认真、独立完成属于自己的课程设计内容，课程设计报告是否思路清晰、文字通顺、书写规范 <p>评语:</p> <p>成绩:</p> <p>指导教师签名：</p> <p>年 月 日</p>				

1. 实验目的

- 通过本实验进一步加深和巩固对所学知识点的理解和掌握，如串行通信原理
- 综合运用所学知识、工程原理来分析和完成本课程设计
- 理解计算机软件、硬件如何配合工作，能对计算机硬件进行功能扩展
- 掌握接口电路设计过程和驱动程序的编写方法
- 学会如何区分系统中软件、硬件故障和调试手段。
- 培养计算机软硬件的综合设计能力和调试能力

2. 实验内容

2.1. 硬件部分

- 4x4 小键盘接口设计（利用 8255 C 口）
- 异步串行通信接口设计（8254 为波特率发生器）
- 4 位数码管显示器设计

2.2. 软件部分

- 键盘扫描程序。将按键 0~F 转换成 0~15 的二进制值。
- 串行接口通信程序。发送为查询方式，接收采用中断方式。
- 4 位七段数码管显示程序（显示接收到的数据）。左移位方式显示 0~9、A~F（E 并表明无数字，初始状态显示）
- 将串行接口发送端和接收端环回，当在 4x4 小键盘上敲击按键时，经串口发送、接收后将对应键值显示在 4 位数码管上。（要求：依次敲击 1、2、3、4、5，则依次显示 xxx1、xx12、x123、1234、2345）
- 两组间进行双机通信，一方用小键盘键入数字，另一方接收到后在 4 位数码管上显示。
- 在液晶显示器上实现显示功能，能够显示最近发送的数据与最近接收的数据

3. 实验原理

3.1. 设计思想-模块设计

3.1.0 主体框架

如 Figure 1. 多接口系统框图所示。

发送数据：查询方式，从 8255C 口进行键盘扫描读入按键，将其发送给 8251。

接收数据：中断方式，从 8251 接收到发送来的按键值，将其转为显示需要的七段码在液晶显示器 LCD 上进行移位显示。

主函数为一个大循环，依次查询是否有键被按下即发送是否准备好，再通过中断处理接收数据，执行相应操作。循环执行主程序，直至程序终止。

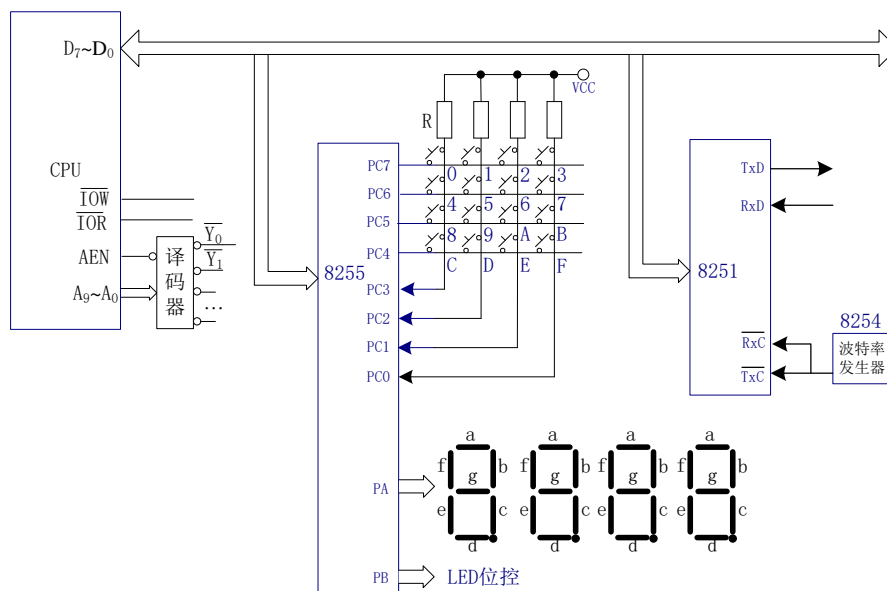


Figure 1. 多接口系统框图

3.1.1 扫描矩阵式键盘

如图 Figure 2.所示, 4x4 的矩阵式键盘, 采用一个可编程并行接口芯片 8255 完成输入/输出, 8255 的 C 口高 4 位 (PC7~PC4) 用作输出接口, C 口低 4 位 (PC3~PC0) 用作输入接口, 4 条输出行线和 4 条输入列线构成 16 个交叉点, 每个交叉点处有一个按键开关, 从而构成一个具有 16 个按键的键盘。

采用行扫描法,使键盘上某一行线为低电平,而其余行接高电平,然后读取列值;如果列值中有某位为低电平,则表明当前行与当前列交点处的键被按下;否则扫描下一行,直到扫完全部的行线为止。

分别用寄存器记录当前行和当前列，并用 $\text{行数} \times 4 + \text{列数}$ 计算键值。

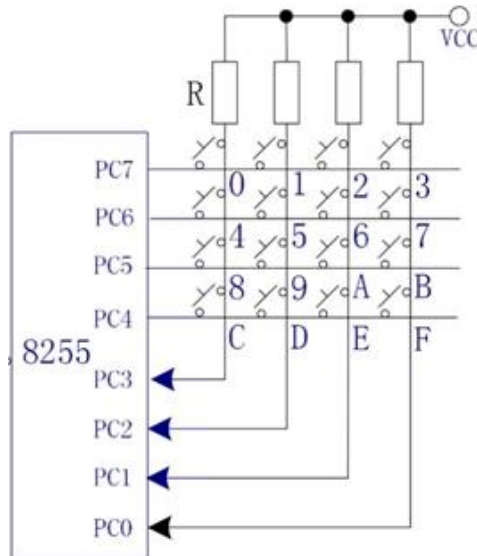


Figure 2. 8255 与键盘阵列

3.1.2 可编程并行接口 (8255)

如图 Figure 2.所示, 8255A 芯片是由数据总线缓冲器、读/写控制逻辑、ABC3 个双向 I/O 数据端口及相关控制电路组成。在该次实验中, 使用 8255A 来完成矩阵式键盘的输入/输出以及七段数码显像管/LCD 液晶屏显示的控制和数据输入。

8255 的 C 口高 4 位用作矩阵式键盘输出接口, C 口低 4 位用作矩阵式键盘输入接口。在七段数码显像管中, A 口 8

位用作数据输入接口，B口用8位用作位控输入接口。在LCD液晶屏中，A口8位用作数据输入接口，B口低3位用作控制接口。

3.1.3 串行接口 (8251)

8251 是可编程的串行通信接口芯片，它可以实现串行同步通信，也可以实现串行异步通信。在此次实验中，我们采用串行异步通信方式，因为考虑到双机通信时两个实验台使用同一时钟源较为麻烦，且单机通信时同步方式会出错，故采用异步方式更为灵活。

实验中的自发自收、双机通信皆是使用 8251 芯片进行收发，采用异步通信，1 停止位，奇校验，8 数据为，16 波特率。

发送数据口 TxD，接收数据口 RxD。当自发自收时，将自己实验台的 TxD 和 RxD 相连即可自发自收；当双机通信时，两个实验台的 TxD 和 RxD 口交叉相连即可进行双机通信。

3.1.4 定时与计数 (8254)

8254 芯片在该实验中用作频率发生器（方式 2），给串行接口 8251 提供工作时钟。

在本次实验中，使用了 8254 的计数器 0 和计数器 1，计数器 0 负责给 8251 提供收发时钟。提供收发时钟是为了让 8251 正常进行收发工作。

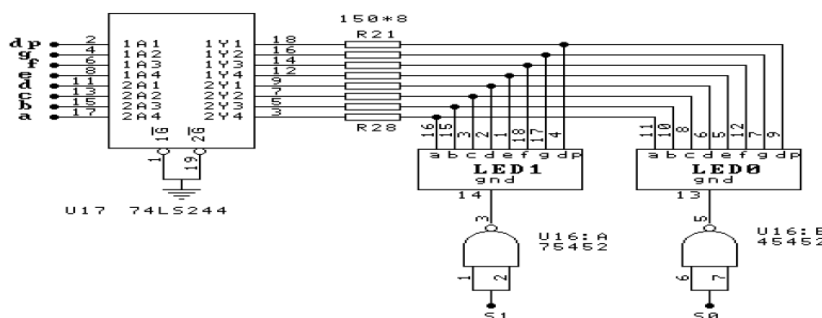


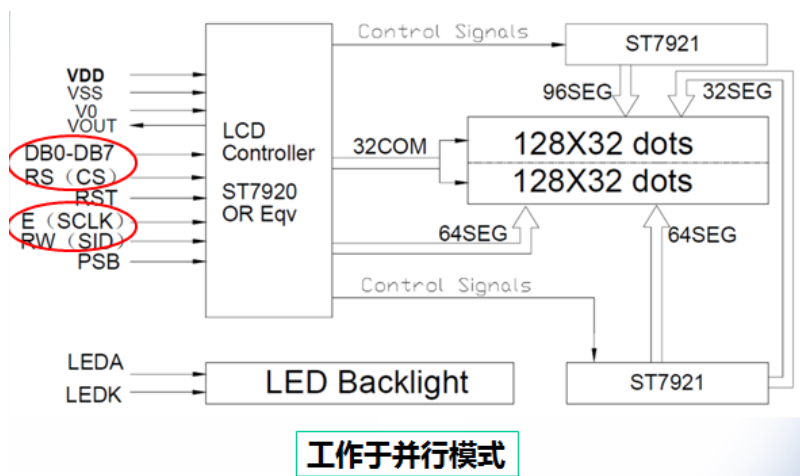
Figure 3. 七段数码管显示电路

3.1.5 七段数码管显示

8255 与七段数码管连接，编程移位显示接收的数据。在实验中，8255 通过 A 口传送数据到七段数码管，B 口传送位控信号。在送下一位数码管的数据和位控信号前需要先对之前的位控信号清零，防止数据的重叠显示。

8255 B 口输出，高四位 PB7-PB4 接 LED 位码 S3-S0，选择某一个数码管。

8255 A 口输出，PA7-PA0 接 LED 的 DP-A，将七段译码值输出到数码管，并进行显示。



工作于并行模式

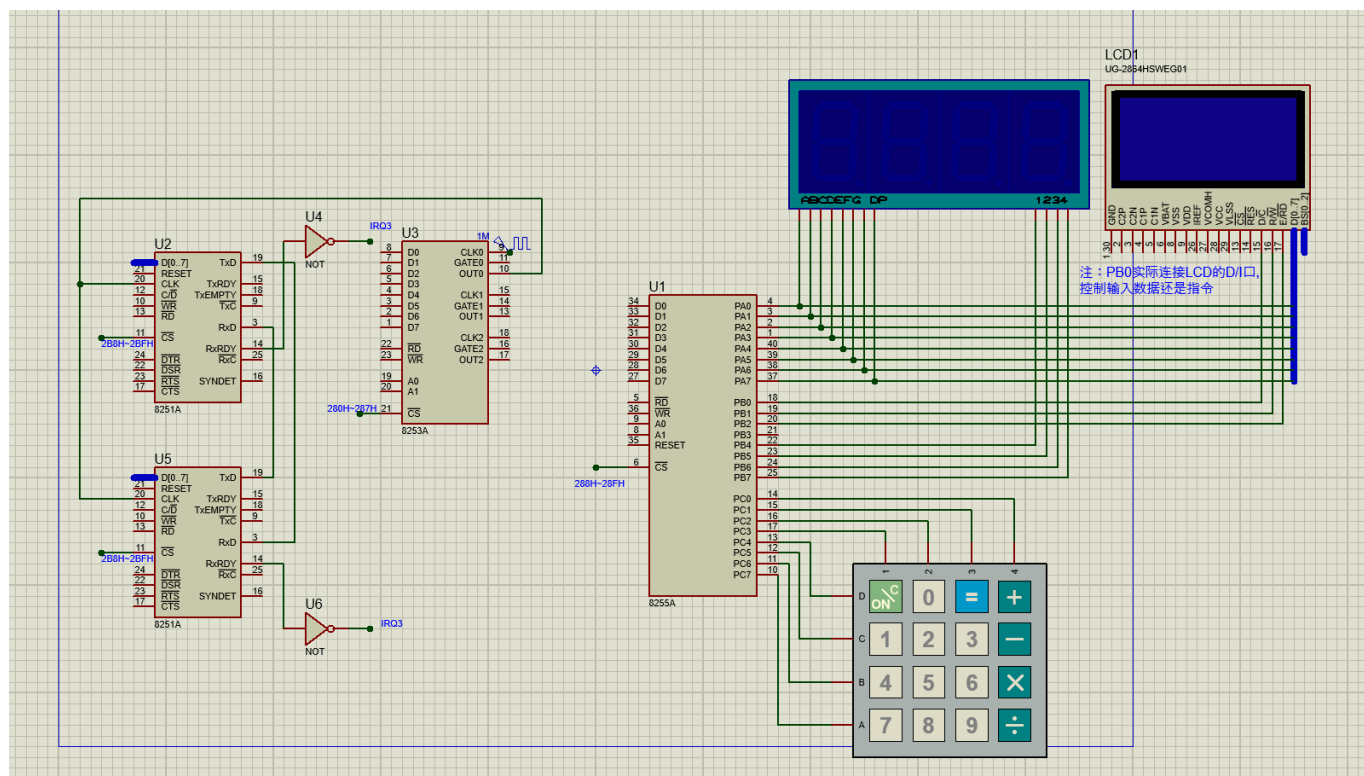
Figure 4. LCD 线路图

3.1.6 LCD 液晶屏显示

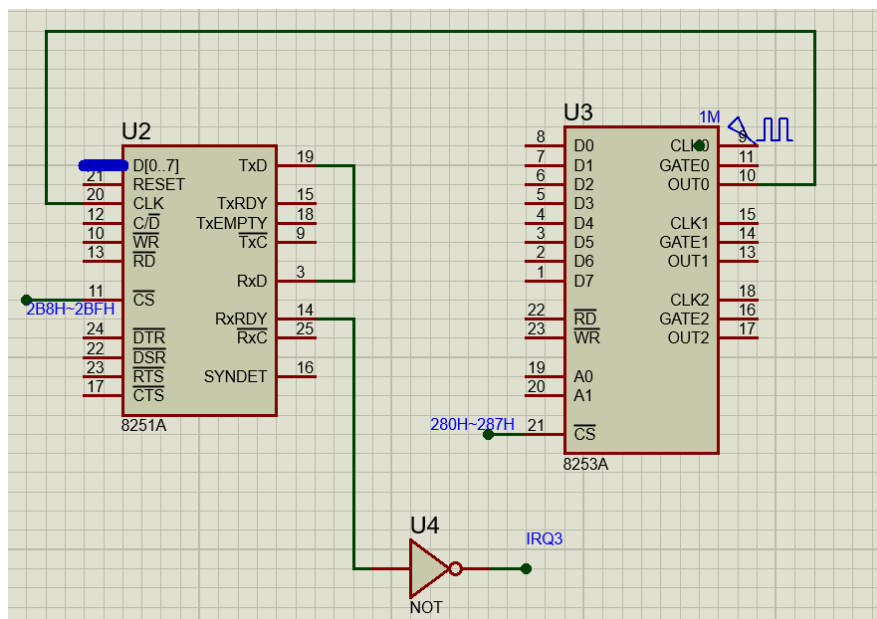
使用 8255 与 128x64LCD 显示模块连接, 编程显示接收的数据。采用半宽字型进行显示, 将 8 位元写入 DDRAM 中, 范围为 02H~7FH 的编码。其中数字 0~9 的编码为 30H—39H, 字母 A~F 的编码为 41H~46H。

3.2. 实验原理图

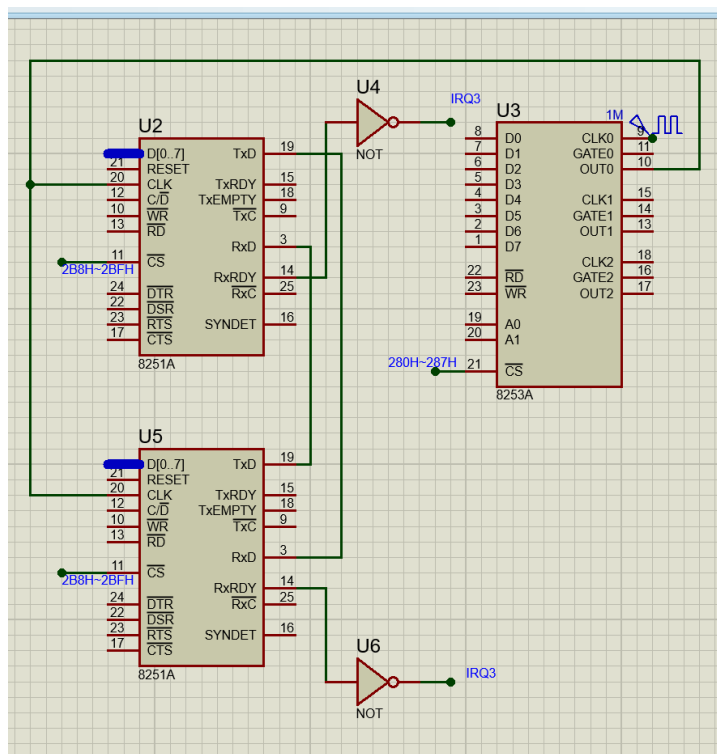
3.2.0 系统总体接线图



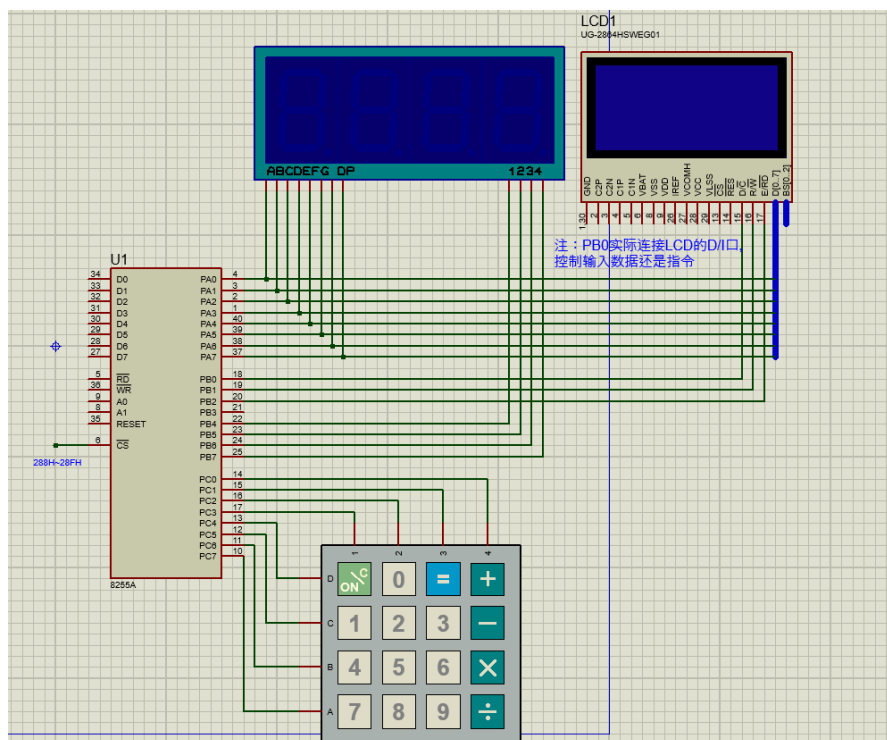
3.2.1 8251、8254 单机通信连接图



3.2.2 8251 双机通信连接图

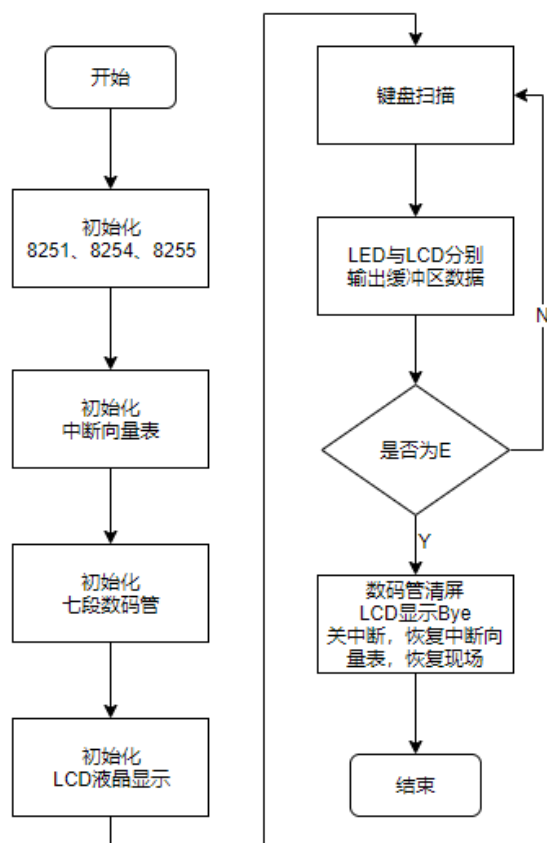


3.2.3 8255、矩阵键盘、LCD 液晶显示屏、七段数码显像管

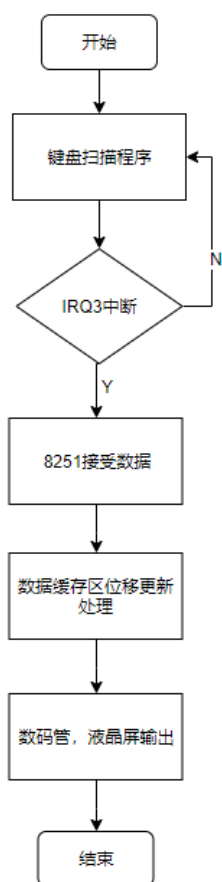


3.3. 实验流程图

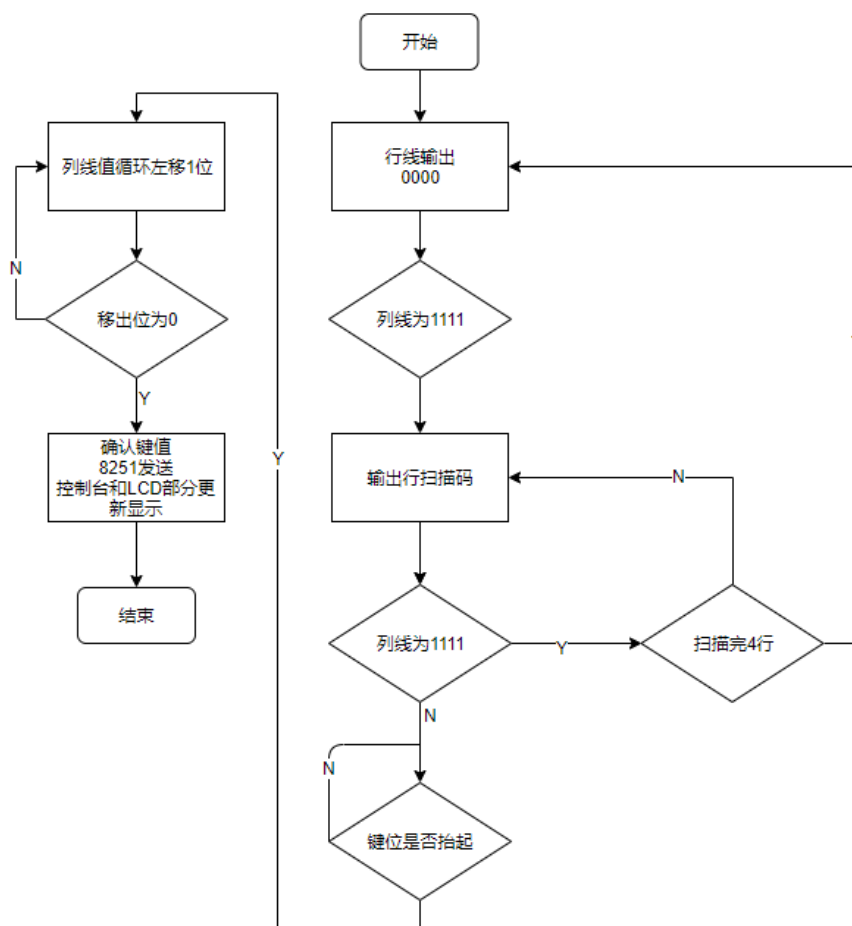
3.3.0 主程序



3.3.1 键盘扫描程序



3.3.2 中断服务子程序



4 . 实验结果

4.1. 单机通信模式

LED 数码管显示采用中断接收方式显示，LCD 发送部分采用查询方式显示，接收部分同 LED 一样采用中断接收方式显示。同时 Dos 控制台也能及时显示键盘矩阵扫描结果。

单机模式，正常接线的情况下：初始时，数码管显示“EEEE”，LCD 上显示“R EEEE”，“R”代表接收的数据和“S EEEE”，“S”代表发送的数据。之后每次按键，三个数据都向左位移显示数据。当按下“E”时，程序退出，数码管清空、全灭，LCD 显示“BYE”字样。如输入数据依次为 1、2、3、4、5、E。数码管与 LCD 显示数字部分都相同显示“EEEE”，“EEE1”，“EE12”，“E123”，“1234”，“2345”，数码管熄灭，LCD 显示“BYE”。

当拆除 IRQ 的连线时，表明取消中断响应。此时，按照上面顺序按键，可以看到数码管与 LCD 接受（“R”）部分的数据显示不会更新，始终保持“EEEE”。但 LCD 发送（“S”）部分数据显示，照常更新，还是依次显示“EEEE”，“EEE1”，“EE12”，“E123”，“1234”，“2345”，直到按下 E，数码管熄灭，LCD 显示“BYE”。表明数码管显示与 LCD 接受数据的显示是中断方式，LCD 发送数据显示是查询方式。

防抖动设计，只在按键抬起的瞬间发射扫描到的读数，长按按键在不抬起时并不会识别。

4.2. 双机通信模式

LED 数码管显示采用中断接收方式显示，LCD 发送部分采用查询方式显示，接收部分同 LED 一样采用中断接收方式显示。同时 Dos 控制台也能及时显示键盘矩阵扫描结果。这里接受数据，将从另外一台机器中接受。假设一台机器为 A 机器，一台为 B 机器。

初始状态与单机模式相同，不再赘述。正常连接情况下，A 机器点击发送数据，B 机器的数码管与 LCD 接受（“R”）部分将会更新，左移位显示，而 A 机器自身的 LCD 发送（“S”）部分也将更新显示，并且数据与 B 中数码管一致。同样的，B 机器按键，会有相类似的效果。当一台机器按下“E”时，比如 A 机器，则 A 机器数码管熄灭，LCD 显示“BYE”，退出程序，此时 B 机器再按键发送，A 机器将不会有所反应，只有 B 机器 LCD 发送（“S”）部分数据会发生变化。硬件与程序都能很好地运行。

如单机通信一般，断开 IRQ 连线，相同的，也能获得相应的效果，来体现数码管与 LCD 接受（“S”）部分是通过中断方式来更新显示的。而 LCD 的发送（“S”）部分不会受到影响，是查询方式。实验结果很好。

防抖动效果与之前相同，不再赘述。

5. 问题解决及收获体会

5.1. 问题与解决

1. 七段数码管 LED 每次只显示一位

在程序主循环中每次都要执行一下 LED 显示的函数，因为 LED 的显示是高电平使能。每次只能显示一位数字，依次输出，通过高速的循环显示使得多位数字能够同时显示。

2. RxRDY 端口直接接入 IRQ3 口，未能正常执行中断服务子程序

中断服务子程序是下降沿触发，将 8251 的 RXRDY 端口传出的信号先进行反相器转换（一个非门），最后再输出到 IRQ3，这样，下降沿信号能够准确触发中断。

3. 键盘扫描的时候容易出现连接的问题，按一次出现多个数值

代码中加入键盘扫描防抖动的机制，使得扫描结果在键抬起的时候再读取。先把按键结果存到一个变量里，在按键过程中，如果检测到有键按下（还没抬起），则不断重新扫描。在抬起以后，再从变量里面取出之前的按键。

4. 8251 状态端口读出的 TxRDY 信号位总为 0，无法发送数据

在初始化 8251 前预先向其控制字端口输出三次 0 对其进行预初始化（即软复位）。

5. 8251 状态端口读出的 RxRDY 信号位总为 0，无法接收数据

在发送完数据之后需要让程序延迟等待一下，否则无法接收到数据。

6. LCD 显示在多次输出后会产生一些不必要的输出

LCD 显示发送数据使用查询方式，应在发送数据时，与显示控制台同步一起显示在 LCD 屏幕中，发送 LCD 显示子程序需要在此调用。将原来的 LCD 控制信号和数码管位控信号分开，在 8255 的 B 口区分分成两个部分，前三位用于 LCD 控制信号，后四位用于数码管的位控。

5.2. 收获与体会

欧宝源：

通过本次微机原理与接口技术课程设计，我重新复习了诸如 8255、8254、8251 等器件的工作方式与流程，诸如初始化、收发数据等等；对按键的按下抬起检测、中断处理等问题完成解决方法。

开始对 LCD 我和队友对其工作原理与流程都非常的陌生，需要查阅很多资料才能大致掌握框架；另外，双机通信的方面着实花费了不少时间，这也是由于对 8251 的不熟悉造成的。

在完成课程设计的过程中，出现了许多的问题，需要清楚地定位问题的来源是硬件还是软件，从而解决它们。这个过程是繁琐并且枯燥的。但沉下心来，慢慢调试，发现错误并修正错误是我在此次课设中最大的收获。

裴子祥：

通过这次微机原理与接口技术课程设计，我重新复习了微机接口的课程，对汇编语言的编写有了更好的掌握。关于 4*4 键盘与数码管显示，代码逻辑较为清晰，编写起来也算比较容易，很快可以达到预期的目标；主要在于对 8251 器件的熟练掌握。

我们使用了模块化的设计完成课设。因此通过单步调试的方法，我们可以隔离各个模块，逐步确定每一个模块的输入输出是否正确，从而判断出问题究竟出现在哪一个模块中，问题是在硬件还是在软件。当硬件出现问题时，我们需要利用现有资源，去判别它是否真的损坏，还是自身逻辑问题，向老师证明，并及时汇报。这期间需要多次查询硬件的相关文档资料，也慢慢更加熟练熟悉。遇到问题，不要埋怨，多与同伴交流，相互合作，沟通想法永远是最为行之有效的方式。

本次小学期的实验内容详实有料，需要静下心来重温以前的知识，借这次课程设计，重新回顾了硬件方面的知识，巩固日后学习的基础。

6. 实验程序

```
;数据段
data    segment
;常量
c8251portio    equ 2b8h
c8251portctrl  equ 2b9h
c8254port0     equ 280h
c8254port1     equ 281h
c8254portctrl  equ 283h
c8255porta     equ 288h
c8255portb     equ 289h
c8255portc     equ 28ah
c8255portctrl  equ 28bh
c8259port0     equ 20h
c8259port1     equ 21h
disploc        equ 93h

;16个键值,对应数码管的显示,键盘序 FEDCBA9876543210;
key_cc db 71h,79h,5eh,39h,7ch,77h,6fh,7fh,07h,7dh,6dh,66h,4fh,5bh,06h,3fh
;液晶屏显示数字
```

```
    lcd_num db 46h,45h,44h,43h,42h,41h,39h,38h,37h,36h,35h,34h,33h,32h,31h,30h
    lcd_end db 42h,59h,45h
;ascii 码,屏显 rec
    asc db 46h,45h,44h,43h,42h,41h,39h,38h,37h,36h,35h,34h,33h,32h,31h,30h
;保存上一键值
    temp db ?
    rec db 01h, 01h, 01h, 01h
    buf db 01h, 01h, 01h, 01h
    linekey db ?
data ends
```

;代码段

```
code segment
    assume cs:code,ds:data
;-----数码管显示数字子程序-----
display proc
    push cx
    push ax

    mov al,00h
    mov dx,c8255portb
    out dx,al
    mov al,rec[0]
    and ax,00ffh ;取低位数据,即为相应键下标值
    mov si,ax
    mov al,key_cc[si]
    mov dx,c8255porta
    out dx,al ;传送第一个数码管数据
    mov al,10h
    mov dx,c8255portb
    out dx,al ;传送第一个数码管位控信号,点亮

    call delay1

    mov al,00h
    mov dx,c8255portb
    out dx,al
    mov al,rec[1]
    and ax,00ffh ;取低位数据,即为相应键下标值
    mov si,ax
    mov al,key_cc[si]
    mov dx,c8255porta
    out dx,al ;传送第二个数码管数据
    mov al,20h
    mov dx,c8255portb
```

```
    out dx,al      ;传送第二个数码管位控信号，点亮

    call delay1

    mov al,00h
    mov dx,c8255portb
    out dx,al
    mov al,rec[2]
    and ax,00ffh    ;取低位数据，即为相应键下标值
    mov si,ax
    mov al,key_cc[si]
    mov dx,c8255porta
    out dx,al      ;传送第三个数码管数据
    mov al,40h
    mov dx,c8255portb
    out dx,al      ;传送第三个数码管位控信号，点亮

    call delay1

    mov al,00h
    mov dx,c8255portb
    out dx,al
    mov al,rec[3]
    and ax,00ffh    ;取低位数据，即为相应键下标值
    mov si,ax
    mov al,key_cc[si]
    mov dx,c8255porta
    out dx,al      ;传送第四个数码管数据
    mov al,80h
    mov dx,c8255portb
    out dx,al      ;传送第四个数码管位控信号，点亮

    call delay1
    mov al,00h
    mov dx,c8255portb
    out dx,al      ;传送第四个数码管位控信号，点亮

    pop ax
    pop cx
    ret
display endp

;-----液晶屏清屏子程序-----
lcd_clear  proc
    mov dx,c8255porta
```

```
mov al,00001100b    ;设置打开显示命令
out dx,al
call lcd_ctrl        ;液晶屏送入指令

mov dx,c8255porta
mov al,00000001b    ;设置清屏指令
out dx,al
call lcd_ctrl        ;液晶屏送入指令

ret
lcd_clear    endp

;-----液晶屏显示数字子程序-----
lcd_display proc
    push dx
    push ax
    push si
;显示第一个数
    mov dx,c8255porta
    mov al,8dh        ;设定显示位置(0x80h+DDRAM 地址)
    out dx,al        ;设定 DDRAM 地址命令
    call lcd_ctrl    ;液晶屏送入指令

    mov dx,c8255porta
    mov al,buf[0]
    and ax,0ffh
    mov si,ax
    mov al,lcd_num[si] ;数字对应的字符编码
    out dx,al
    call lcd_data    ;液晶屏送入数据

    call delay

;显示第二个数
    mov dx,c8255porta
    mov al,8ch        ;设定显示位置(0x80h+DDRAM 地址)
    out dx,al        ;设定 DDRAM 地址命令
    call lcd_ctrl    ;液晶屏送入指令

    mov dx,c8255porta
    mov al,buf[1]
    and ax,0ffh
    mov si,ax
    mov al,lcd_num[si] ;数字对应的字符编码
    out dx,al
```

```
    call lcd_data      ;液晶屏送入数据

    call delay

;显示第三个数
    mov dx,c8255porta
    mov al,8bh         ;设定显示位置(0x80h+DDRAM 地址)
    out dx,al          ;设定 DDRAM 地址命令
    call lcd_ctrl      ;液晶屏送入指令

    mov dx,c8255porta
    mov al,buf[2]
    and ax,0ffh
    mov si,ax
    mov al,lcd_num[si] ;数字对应的字符编码
    out dx,al
    call lcd_data      ;液晶屏送入数据

    call delay

;显示第四个数
    mov dx,c8255porta
    mov al,8ah         ;设定显示位置(0x80h+DDRAM 地址)
    out dx,al          ;设定 DDRAM 地址命令
    call lcd_ctrl      ;液晶屏送入指令

    mov dx,c8255porta
    mov al,buf[3]
    and ax,0ffh
    mov si,ax
    mov al,lcd_num[si] ;数字对应的字符编码
    out dx,al
    call lcd_data      ;液晶屏送入数据

    call delay

;显示 s
    mov dx,c8255porta
    mov al,88h         ;设定显示位置(0x80h+DDRAM 地址)
    out dx,al          ;设定 DDRAM 地址命令
    call lcd_ctrl      ;液晶屏送入指令

    mov dx,c8255porta
    mov al,buf[3]
    and ax,0ffh
```



```
mov si,ax
mov al,53h ;数字对应的字符编码
out dx,al
call lcd_data ;液晶屏送入数据

call delay

;-----receive lcd-----
;显示第一个数
mov dx,c8255porta
mov al,9dh ;设定显示位置(0x80h+DDRAM 地址)
out dx,al ;设定 DDRAM 地址命令
call lcd_ctrl ;液晶屏送入指令

mov dx,c8255porta
mov al,rec[0]
and ax,0ffh
mov si,ax
mov al,lcd_num[si] ;数字对应的字符编码
out dx,al
call lcd_data ;液晶屏送入数据

call delay

;显示第二个数
mov dx,c8255porta
mov al,9ch ;设定显示位置(0x80h+DDRAM 地址)
out dx,al ;设定 DDRAM 地址命令
call lcd_ctrl ;液晶屏送入指令

mov dx,c8255porta
mov al,rec[1]
and ax,0ffh
mov si,ax
mov al,lcd_num[si] ;数字对应的字符编码
out dx,al
call lcd_data ;液晶屏送入数据

call delay

;显示第三个数
mov dx,c8255porta
mov al,9bh ;设定显示位置(0x80h+DDRAM 地址)
out dx,al ;设定 DDRAM 地址命令
call lcd_ctrl ;液晶屏送入指令
```

```
mov dx,c8255porta
mov al,rec[2]
and ax,0ffh
mov si,ax
mov al,lcd_num[si] ;数字对应的字符编码
out dx,al
call lcd_data      ;液晶屏送入数据

call delay

;显示第四个数
mov dx,c8255porta
mov al,9ah          ;设定显示位置(0x80h+DDRAM 地址)
out dx,al           ;设定 DDRAM 地址命令
call lcd_ctrl       ;液晶屏送入指令

mov dx,c8255porta
mov al,rec[3]
and ax,0ffh
mov si,ax
mov al,lcd_num[si] ;数字对应的字符编码
out dx,al
call lcd_data      ;液晶屏送入数据

call delay

;显示 R
mov dx,c8255porta
mov al,98h          ;设定显示位置(0x80h+DDRAM 地址)
out dx,al           ;设定 DDRAM 地址命令
call lcd_ctrl       ;液晶屏送入指令

mov dx,c8255porta
mov al,rec[3]
and ax,0ffh
mov si,ax
mov al,52h          ;数字 R 对应的字符编码
out dx,al
call lcd_data      ;液晶屏送入数据

call delay

pop si
pop ax
```

```
    pop dx
    ret
lcd_display endp

;-----液晶屏退出界面子程序-----
lcd_exit    proc
    call lcd_clear
    mov si,0        ;偏移量
    mov bl,92h       ;显示坐标
    mov cx,3         ;循环 4 次,输出"END"

lp: mov dx,c8255porta
    mov al,bl        ;设定显示位置(0x80h+DDRAM 地址)
    out dx,al        ;设定 DDRAM 地址命令
    call lcd_ctrl     ;液晶屏送入指令

    mov dx,c8255porta
    mov al,lcd_end[si] ;字符编码
    out dx,al        ;输出字符编码
    call lcd_data     ;液晶屏送入数据

    inc si
    inc bl
    loop lp

    ret
lcd_exit    endp

;-----液晶屏送入控制指令子程序-----
lcd_ctrl    proc
    mov dx,c8255portb ;此处 8255 的 b 端口的低三位接 LCD 的控制端口

    mov al,00000000b   ;D/I=0,R/W=0,写指令
    out dx,al
    call delay

    mov al,00000100b   ;E=1 使能
    out dx,al
    call delay

    mov al,00000000b   ;E=0 关闭使能
    out dx,al
    call delay

    ret
endp
```

```
lcd_ctrl    endp

;-----液晶屏送入数据子程序-----
lcd_data    proc
    mov dx,c8255portb    ;此处 8255 的 b 端口的低三位接 LCD 的控制端口

    mov al,00000001b    ;D/I=1,R/W=0,写数据
    out dx, al
    call delay

    mov al,00000101b    ;E=1 使能
    out dx, al
    call delay

    mov al, 00000001b    ;E=0 关闭使能
    out dx, al
    call delay

    ret
lcd_data    endp

;-----主程序-----
start:  mov ax,data
        mov ds,ax;    ds 指向 data 段

;初始化 8251、8254、8255、中断向量表、液晶屏
;初始化 8251
    mov dx,c8251portctrl
    xor al,al
    out dx,al    ;预初始化
    mov al,40h
    out dx,al    ;软复位
    nop          ;延时

    mov al,11001111b;设置方式命令字,异步 2 停止位,无校验,8 数 据位, 64 波特率
    mov dx,c8251portctrl
    out dx,al

    mov al,00110111b;正常模式,内部不复位,请求发送信号 rts 高, 错误标志复位,不发送中止符,接收器使
能,dts 低,发送器使能
    out dx,al    ;设置控制字

;初始化 8254
;收发时钟
    mov al,00110100b;设置方式控制字,计数器 0,先低后高,方式 2, 二进制计数
```

```
mov dx,c8254portctrl
out dx,al
mov ax,28h      ;计数初值 40
mov dx,c8254port0
out dx,al      ;写入计数初值低字节
mov al,ah
out dx,al      ;写入计数初值高字节
;中断触发
mov al,01110100b;设置方式控制字,计数器 1,先低后高,方式 2, 二进制计数
mov dx,c8254portctrl
out dx,al
mov ax,0ffh     ;计数初值 255
mov dx,c8254port1
out dx,al      ;写入计数初值低字节
mov al,ah
out dx,al      ;写入计数初值高字节

;初始化 8255
mov al,10000001b;a 组方式 0,a 口输出,c 口高四位输出,b 组方式 0,b 口输出,c 组低四位输入
mov dx,c8255portctrl
out dx,al

;初始化接收中断
mov ax,350bh    ;取出第 0bh 个向量表
int 21h         ;取出旧中断服务程序入口地址
push es
push bx        ;保存在堆栈中

cli           ;关中断
push ds
mov dx,offset receive ;中断服务程序 receive 的偏移地址
mov ax,seg receive
mov ds,ax
mov ax,250bh   ;写入第 0bh 个向量表
int 21h        ;将中断服务程序 receive 入口地址 填入中断向量表
pop ds

in al,c8259port1
and al,0f7h    ;OCW1,允许 IR3
out c8259port1,al
sti           ;开中断

;初始化液晶屏
call lcd_clear ;lcd 清除
```

```
;初始化 temp,temp 用于存放上一键值防重复显示
mov temp,0ffh

call lcd_display

;键盘输入检测
input:

mov al,00h
mov dx,c8255portc
out dx,al      ;00->c 口高四位

waiting:
mov al,00h
mov dx,c8255portb
out dx,al

call display
mov dx,c8255portc
in al,dx      ;c 口低四位读入列线值
and al,0fh    ;取低 4 位
cmp al,0fh    ;al 全为 1, 则 zf=1

jz waiting    ;无键按下, zf=1 跳转继续扫描

mov cx,7fffh  ;有键按下
lp0:loop lp0  ;延时

mov bl,4      ;行数
mov bh,4      ;列数
mov al,0efh   ;扫描码
mov ch,0ffh   ;按键下标值-1

row:  mov dx,c8255portc
out dx,al     ;c 口高四位输出扫描码
rol al,1      ;循环左移生成下一扫描码
mov ah,al     ;保存下一扫描码
mov dx,c8255portc
in al,dx      ;c 口低四位读入列值
mov linekey, al
and al,0fh    ;取低四位
cmp al,0fh    ;判断是否有列线为 0
jnz dead      ;有列线为 0, 转 col

add ch,bh     ;下标值改为下行首键下标值-1
```

```
    mov al,ah      ;取扫描码
    dec bl         ;行计数-1
    jnz row        ;行数不为0,继续扫描

    jmp input

dead:  mov dx, c8255portc
      call delay
      in al, dx
      cmp al, linekey
      jz dead
      mov al, linekey

col:
    inc ch         ;键下标值+1
    rcr al,1       ;带进位的循环右移位
    jc col         ;右移出的最低位为1,该列未按下,继续循环

    mov al,ch      ;al = 键下标值

;8251 发送
    cmp al,01h     ;按下e时退出
    jz exit

    push ax
    mov al,buf[2]
    mov buf[3],al  ;buf[2]->buf[3]
    mov al,buf[1]
    mov buf[2],al  ;buf[1]->buf[2]
    mov al,buf[0]
    mov buf[1],al  ;buf[0]->buf[1]
    pop ax
    mov buf[0],al  ;新数据存放到buf[0]中

    call lcd_display

    mov dx,c8251portio
    out dx,al      ;发送键值

;cmp temp,al
;jz input         ;键值重复,不重复显示屏显
```

```
;首次按键,屏显
    mov temp,al      ;保存新键值
    and ax,0fh       ;ax 高位清 0
    mov si,ax

    mov dl,asc[si]   ;按键对应 asc

    mov ah,02h
    int 21h

    jmp input        ;重新扫描是否有按键按下

;退出
exit:  call lcd_exit

    in al,c8259port1
    or al,08h        ;OCW1,禁止 IR3
    out c8259port1,al

;恢复旧中断屏蔽字
    cli              ;关中断
    pop dx           ;弹出旧中断服务程序入口的偏移地 址
    pop ds           ;弹出旧中断服务程序入口的段地址
    mov ax,250bh     ;写入第 08h 个向量表
    int 21h          ;恢复旧中断服务程序入口的地址
    sti              ;开中断

    mov ax,4c00h
    int 21h          ;返回 dos

;接收中断程序
receive proc
;保存现场
    push ax
    push dx
    push si

;接收数据
    mov dx,c8251portio
    in al,dx         ;8251 接收数据

    push ax
    mov al,rec[2]
```



```
    mov rec[3],al    ;rec[2]->rec[3]
    mov al,rec[1]
    mov rec[2],al    ;rec[1]->rec[2]
    mov al,rec[0]
    mov rec[1],al    ;rec[0]->rec[1]
    pop ax
    mov rec[0],al    ;新数据存放到 rec[0] 中

    call lcd_display

;液晶屏显示
    and ax,0fh      ;高位清 0
    mov si,ax

;恢复现场
    pop si
    pop dx

;通知 8259 中断结束
    mov al,20h      ;OCW2,中断结束
    out c8259port0,al

    pop ax

    iret            ;中断返回
receive endp

;延迟
delay proc
    push cx
    mov cx,0ffffh
lp1:loop lp1
    pop cx
    ret
delay endp

delay1 proc
    push cx
    mov cx,24h
lp2:    call delay
    loop lp2
    pop cx
    ret
delay1 endp
```

```
code ends  
end start
```