

东南大学

《微机系统与接口实验》

实验报告

实验十 键盘中断

姓 名：薛宇飞

学 号：04020235

同 组：

学 号：

专 业：信息工程

实 验 室：金智楼硬件实验室

实验时间：2022 年 5 月 12 日

报告时间：2022 年 5 月 12 日

评定成绩：

评阅教师：裴文江

目录

1 实验目的与内容	3
2 实验任务	3
1 基本功能	3
2 附加任务	3
3 实验原理(预习报告部分)	3
4 预备知识	7
5 实验代码	7
6 附加任务阐述	12
1 附加任务 1	12
2 附加任务 2	12
3 附加任务 3	14
7 结果验证	14
8 思考题	14
9 实验总结	15
参考文献	15

一. 实验目的与内容

1. 结合实验教材^[1-2], 了解 Intel 8086CPU 的中断处理功能以及 IBM-PC 的中断结构.
2. 了解 8259 中断控制器的使用.
3. 掌握键盘中断的编程, 观察中断的执行情况.

二. 实验任务

(一) 基本功能

要求每按下任意一个键就向 CPU 发出中断请求信号, 该信号由 8259 的 IRQ1 引入, 中断类型号为 09H, CPU 响应中断后转入执行 KEYINTS 中断服务程序, 并在屏幕上显示 OK!, 按下 10 次键后返回 DOS.

(二) 附加任务

1. 通过 DOS 系统功能调用的 25H, 35H 功能实现中断向量的设置和读取;
2. 在显示 OK! 的前面增加显示按键次数;
3. 按键 10 次后, 不等 25 行太阳图标显示完, 立即返回 DOS;
4. 修改显示字符的属性, 如, 红底白字, 蓝底黄字……

三. 实验原理(预习报告部分)

键盘与主机是通过 5 芯螺旋形的电缆相连的, 其中包括数据线、时钟线、复位线、+5V 电源线和地线.(电缆插入系统板后部的插座)

每当有键按下或释放时, 键盘以串行方式向系统板的键盘接口电路传送数据, 即扫描码. 一个扫描码移位传送完, 键盘接口电路便向主机发出中断请求信号 IRQ1(中断类型码为 09H), 此信号送到 8259A 产生中断请求.

CPU 响应中断请求时, 查中断向量表, 从 09H×4 开始的连续四个单元中取出中断向量 (IRQ1 中断服务程序 KEYINTS 的入口地址指针), 转去执行中断服务程序 KEYINTS.

主程序和键盘中断服务程序的流程图如图 1 所示. 请根据流程图编写主程序和键盘中断服务程序.

在主程序中应先读取并保存中断类型号 t 的原中断向量, 然后再设置新的中断向量, 即将中断服务程序 KEYINTS 入口地址的偏移量和段基址存入以 09H×4 为起始地址的四个单元内.

在键盘中断程序 KEYINTS 中, 保护现场、开中断之后, 就通过 8255A 的 PA 口 (PA 口地址为 60H) 读取键盘扫描码, 接着从 8255A PB 口 (PB 口地址为 61H) 的 PB7 输出一个正脉冲 (即 PB7 先输出高电平, 再输出低电平), 先输出的高电平信号反相之后控制键盘状态触发器的清零端, 使 IRQ1 清零, 撤消中断请求信号. 再输出的低电平信号允许位寄存器输出数据, 这样就为传递下一个键盘扫描码作好了准备.

8255A PB 口的 PB7 输出正脉冲来发键盘状态复位命令的具体指令如下:

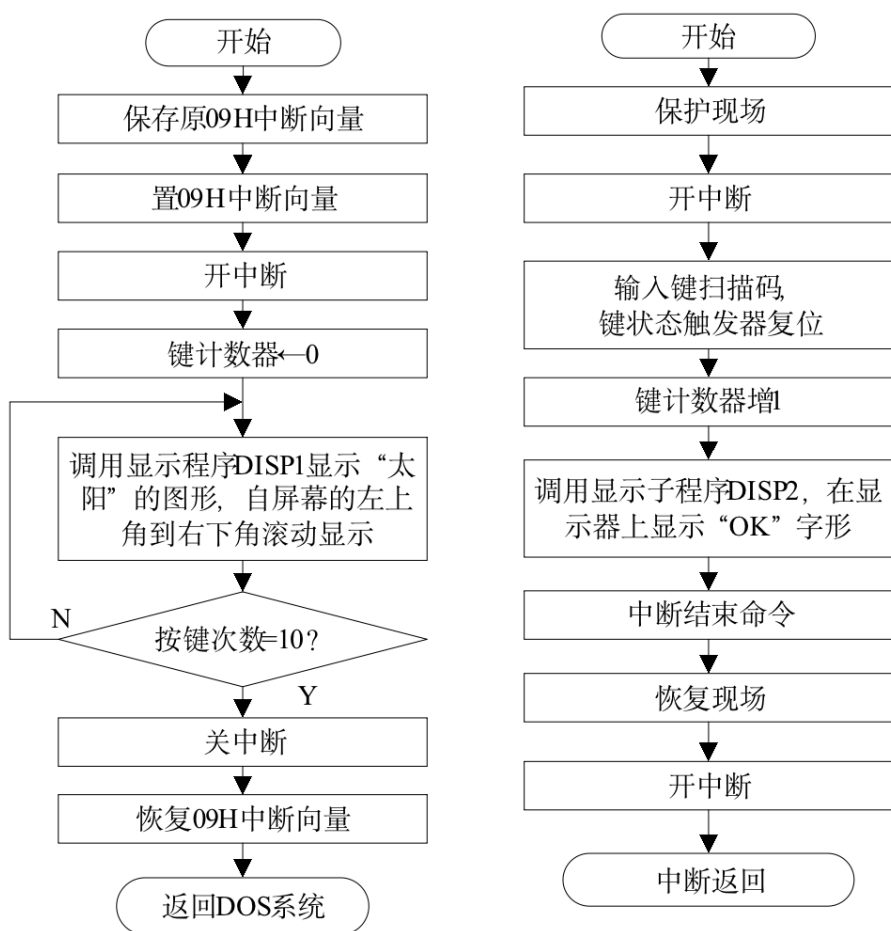


图 1: 任务流程图

code

```

1      IN    AL,61H      ; 输入PB口的当前值
2      OR     AL,80H      ; PB7置1
3      OUT    61H,AL
4      AND    AL,7FH      ; PB7清零
5      OUT    61H,AL

```

在键盘中断程序结束前,应发出“中断结束”(EOI)命令(控制字为20H),给8259A的操作命令字OCW1(端口地址为20H).具体指令如下:

code

```

1      MOV    AL,20H
2      OUT    20H,AL

```

然后,实现中断返回.

另外,当键按下时,送向主机的扫描码是键编号,而键释放时,扫描码为键编号加80H(即第7位置1).例如,按下和释放“A”键将向主机发送两个扫描码:1EH和9EH.

以下是主程序中显示“太阳”图形的子程序DISP1,仅供参考.

code

```

1      DISP1  PROC  FAR
2      PUSH  AX
3      PUSH  BX
4      PUSH  CX
5      PUSH  DX
6      MOV   AH,15      ; 读当前显示状态
7      INT   10H
8      MOV   AH,0       ; 设置显示方式
9      INT   10H
10     MOV   DX,0       ; 行号为0,列号为0
11     REPT:
12     MOV   AH,2       ; 设置光标位置
13     INT   10H
14     MOV   AL,0FH      ; 0FH--太阳图形的ASCII码
15     MOV   CX,1       ; 显示字符个数
16     MOV   AH,10      ; 写字符
17     INT   10H
18     CALL  DELAY
19     SUB   AL,AL
20     MOV   AH,10      ; 清除原图形
21     INT   10H
22     INC   DH         ; 行号+1

```

```
23      ADD    DL,2           ; 列号+2
24      CMP    DH,25         ; 是否到25行?
25      JB     REPT
26      POP     DX
27      POP     CX
28      POP     BX
29      POP     AX
30      RET
31      DISP1   ENDP
```

以下是中断服务程序中显示 OK! 字符子程序 DISP2, 仅供参考.

code

```
1      DISP2   PROC   FAR
2      PUSH    CX
3      PUSH    BX
4      PUSH    AX
5      MOV     CX, 3
6      NEXTC: LODSB           ; 字符串"OK!"在数据段中定义,AL←[SI]
7      MOV     AH,0EH         ; 写字符,并移动光标
8      MOV     BX,01
9      INT     10H
10     CALL    DELAY
11     LOOP    NEXTC
12     POP     AX
13     POP     BX
14     POP     CX
15     RET
16     DISP2   ENDP
```

以下是延时 1 秒子程序:

code

```
1      DELAY   PROC
2      PUSH    CX
3      PUSH    DX
4      MOV     DX,20
5      DL500:
6      MOV     CX,2801
7      DL10ms:
8      LOOP   DL10ms
9      DEC     DX
```

```
10      JNZ      DL500
11      POP      DX
12      POP      CX
13      RET
14      DELAY    ENDP
```

四. 预备知识

注意 1: INT 10H 的 13H 功能

BH= 页码

BL= 属性 (若 AL=00H 或 01H)

CX= 显示字符串长度

(DH、DL) = 坐标 (行、列)

ES:BP= 显示字符串的地址 AL= 显示输出方式

0: 字符串中只含显示字符, 其显示属性在 BL 中. 显示后, 光标位置不变

1: 字符串中只含显示字符, 其显示属性在 BL 中. 显示后, 光标位置改变

2: 字符串中含显示字符和显示属性. 显示后, 光标位置不变

3: 字符串中含显示字符和显示属性. 显示后, 光标位置改变

五. 实验代码

code

```
1      DATA SEGMENT
2      KEY_COUNTS DB ?
3      OK_PRINT DB "      OK!"
          ; 字符串中的空格为按键次数预留(两个数字+两个空格+OK!)
4      DATA ENDS
5
6      STACK SEGMENT                ; 为保护现场操作建立栈段空间
7          DW 100 DUP(?)
8      STACK ENDS
9
10     CODE SEGMENT
11     ASSUME CS:CODE,DS:DATA,ES:DATA,SS:STACK
12     START:
13         MOV AX,STACK
14         MOV SS,AX
15         MOV AX,DATA
```

```
16      MOV DS,AX
17
18      MOV AL,09H
           ;用中断类型21H的35H功能取中断向量保存
19      MOV AH,35H
20      INT 21H
21      PUSH ES
22      PUSH BX
23
24      MOV KEY_COUNTS,00H           ;按键次数置零
25
26      MOV DX,OFFSET KEYINTS
           ;用中断类型21H的25H功能设置中断向量
27      MOV AX,SEG KEYINTS
28      MOV DS,AX
29      MOV AL,09H
30      MOV AH,25H
31      INT 21H
32
33      MOV AX,DATA           ;恢复数据段地址
34      MOV DS,AX
35
36      STI           ;开中断
37
38      SUN:
39      CALL DISP1           ;输出太阳符号
40      CMP KEY_COUNTS,10     ;比较按键次数
41      JB SUN
           ;未到10,超过一页,退出了DISP1,继续输出太阳
42
43      CLI           ;关中断
44
45      POP DX
           ;用中断类型21H的25H功能恢复中断向量,回顾line21 22
46      POP DS
47      MOV AL,09H
48      MOV AH,25H
49      INT 21H
50
51      MOV AH,4CH           ;返回DOS
52      INT 21H
```



```
53
54 KEYINTS PROC NEAR
55     PUSH AX                ;保护现场
56     PUSH BX
57     PUSH DX
58     PUSH SI
59
60     STI                    ;开中断
61
62     IN AL,60H
        ;读取键盘扫描码(ASCII存放在AL)
63     MOV AH,AL              ;保护键盘扫描码
64     IN AL,61H
        ;输入PB口的当前键盘ASCII值
65     OR AL,80H
        ;PB7置1,产生中断请求信号(脉冲信号)
66     OUT 61H,AL
67     AND AL,7FH
        ;PB7置0,为下一次读取扫描码做准备
68     OUT 61H,AL
69
70     TEST AH,80H
        ;根据扫描码判断按键为按下还是松开
71     JNE PASS
        ;不等于,为松开状态,则不计按键数,不输出OK,跳到PASS
72     INC KEY_COUNTS         ;按键数+1
73     CMP KEY_COUNTS,10
        ;保证按键达到十次后结束程序
74     JA PASS                ;高于
75     MOV AL, KEY_COUNTS
76     ADD AL, 30H
        ;将按键次数转换为ASCII码,以便输出
77     CMP AL, 3AH
78     JNE PRINT
79
80     MOV OK_PRINT, 31H
        ;按键次数为10的ASCII码,10是两位数,单独设置一下
81     MOV OK_PRINT+1, 30H
82     JMP CHGSI
83 PRINT:
84     MOV OK_PRINT, AL
```

```
85      CHGSI:
86          LEA BP,OK_PRINT                ;为 DISP2 中进行 INT
            10H 中断作预处理
87          MOV AX,SEG OK_PRINT
88          MOV ES,AX
89          CALL DISP2
90      PASS:
91          MOV AL,20H                    ;发出中断结束命令
92          OUT 20H,AL
93          POP SI                        ;恢复现场
94          POP DX
95          POP BX
96          POP AX
97          IRET                          ;中断返回
98      KEYINTS ENDP
99
100     DISP1 PROC NEAR
101         PUSH AX                        ;保护现场
102         PUSH BX
103         PUSH CX
104         PUSH DX
105
106         MOV AH,15
            ;读当前显示状态,放入AL
107         INT 10H
108
109         MOV AH,0                      ;设置显示方式:
            AL=显示方式号;起到清屏的作用
110         INT 10H
111
112         MOV CX,1                      ;设置显示字符的个数
113         MOV DX,0                      ;设置行列为0
114
115     REAPT:
116         MOV AH,2                      ;设置光标位
117         INT 10H
118
119         CMP KEY_COUNTS,10             ;比较按键次数
120         JNB PASS1
            ;大于等于,次数达到要求则不再输出太阳
121
```

```
122      MOV AL,0FH                                ;读出太阳图形
123
124      MOV AH,10                                ;设置功能号,写字符
125      INT 10H
126
127      CALL DELAY                                ;调用延时子程序
128
129      SUB AL,AL
130      MOV AH,10                                ;清除原图形
131      INT 10H
132
133      INC DH
134      ;设置下一次输出图形的位置
135      ADD DL,2
136      CMP DH,25
137      ;未输出完1页(25行)以前,持续输出
138      JNE REAPT
139
140      PASS1:
141      POP DX                                    ;恢复现场
142      POP CX
143      POP BX
144      POP AX
145      RET
146      DISP1 ENDP
147
148      DISP2 PROC NEAR
149      PUSH CX                                ;保护现场
150      PUSH BX
151      PUSH AX
152      PUSH DX
153
154      MOV BH, 00H
155      MOV AH, 03H                                ;获取光标位置:
156      ;BH=page number(default=0), DH=row number, DL=column number
157      INT 10H
158      MOV CX,7
159      ;显示字符串长度,包含了按键次数和空格,故需要七个(与数据段对应)
160      MOV AH,13H                                ;用AH=13H的INT
161      ;10H中断改变字体颜色
162      MOV BL,4FH                                ;certain color
```

```
158      MOV BH,00H                                ;Page number
159      MOV AL,01H
160      INT 10H
161      CALL DELAY
162      POP DX
163      POP AX                                ;恢复现场
164      POP BX
165      POP CX
166      RET
167  DISP2 ENDP
168
169  DELAY PROC NEAR                                ;延时子程序
170      PUSH CX
171      PUSH DX
172      MOV DX,120
173  DL500:
174      MOV CX,2801
175  DL10ms:
176      LOOP DL10ms
177      DEC DX
178      JNZ DL500
179      POP DX
180      POP CX
181      RET
182  DELAY ENDP
183
184  CODE ENDS
185  END START
```

六. 附加任务阐述

(一) 附加任务 1

任务代码如图2.

(二) 附加任务 2

任务代码如图3.

```
1  MOV AL, KEY_COUNTS
2  ADD AL, 30H           ;将按键次数转换为ASCII码,以便输出
3  CMP AL, 3AH
4  JNE PRINT
5
6  MOV OK_PRINT, 31H     ;按键次数为10的ASCII码,10是两位数,单独设置一下
7  MOV OK_PRINT+1, 30H
8  JMP CHGSI
```

图 2: 附加任务 1

```
1  REAPT:
2  MOV AH, 2             ;设置光标位
3  INT 10H
4
5  CMP KEY_COUNTS, 10    ;比较按键次数
6  JNB PASS1            ;大于等于,次数达到要求则不再输出太阳
7
8  MOV AL, 0FH           ;读出太阳图形
9
10 MOV AH, 10            ;设置功能号,写字符
11 INT 10H
12
13 CALL DELAY            ;调用延时子程序
14
15 SUB AL, AL
16 MOV AH, 10            ;清除原图形
17 INT 10H
18
19 INC DH                ;设置下一次输出图形的位置
20 ADD DL, 2
21 CMP DH, 25            ;未输出完1页(25行)以前,持续输出
```

图 3: 附加任务 2

(三) 附加任务 3

任务代码如图4.



图 4: 附加任务 3

七. 结果验证

运行程序, 按键 10 次后效果图如图 5

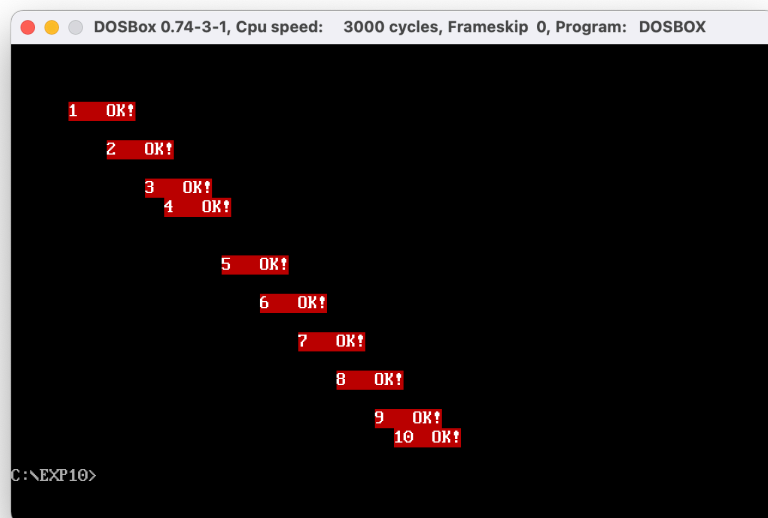


图 5

八. 思考题

键盘上某个键按下和释放时都会向 8259 发出中断请求, 要求只在键按下时显示 OK!, 键释放时不显示, 则中断服务程序 KEYINTS 应如何修改?

可以根据键盘上按键按下和释放时的键盘扫描码第七位不同, 进而判断按键的状态, 具体操作如下: 读取键盘扫描码 IN AL, 60H 后保留 MOV AH, AL, 之后根据扫描码的第七位 TEST AH, 80H: 是 0 则

为按下状态, 计数加 1 并输出 OK!; 是 1 则为松开, 不再计数及输出 OK!(JNZ PASS)。这样即可实现要求功能。

九. 实验总结

1. 在实现附加功能四, 即修改字符的属性时, 起初想要利用功能号为 09H 的 INT 10H 中断, 但是按照 INT 10H 中断操作设置不同颜色后, 输出的字符始终是黑白色。经过不断的调研, 发现功能号为 09H 的 INT 10H 中断中, 若要设置字符颜色和背景颜色, 需要在图形模式下进行, 但是本实验其他功能的实现需要在文本模式下实现, 于是无法使用功能号为 09H 的 INT 10H 中断。调研后发现可以用功能号为 13H 的 INT 10H 中断 (输出字符串) 代替功能号为 09H 的 INT 10H 中断, 且该中断设置颜色时没有工作模式限制, 于是, 使用功能号为 13H 的 INT 10H 中断终于实现了设置字符颜色和背景颜色的功能。
2. 在实现附加功能二的过程中, 起初观察到输出的 OK! 字符不全, 发现原因是字符串长度改变, 但进行 INT 10H 中断时未调整 CH。调整后解决了问题。
3. 实验任务流程图中右图倒数第二行所给的最后一个开中断没有用, 应去除。
4. 其他实验总结已随文附在“注意”、“思考”、“分析”中。

参考文献

- [1] 李继灿. 新编 16/32 位微型计算机原理及应用 (第五版) [M]. 5 版. 北京: 清华大学出版社, 2013.
- [2] 微机教学组. 《微计算机实验讲义》[A]. 南京: 东南大学, 2015.