微机系统与接口技术

课程设计

**姓名**： 何顺发

**班级**： 计算机183

**学号**： 201805120612

[**程序总体介绍 3**](#_Toc26436)

[**大体功能模块 3**](#_Toc26436)

[**运用到的所学的知识 3**](#_Toc26436)

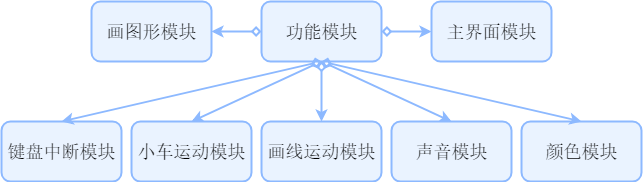
[**操作实现流程图 4**](#_Toc26436)

[**心得体会 27**](#_Toc26436)

**1、程序总体介绍**

该系统能实现多任务，键盘控制，界面绘制，播放声音等功能。根据不同的键盘操作，能响应相应的功能和效果。运行程序后，首先是一个初始页面，按下不同按键，可以分别绘制小车，控制小车运动，画线运动，播放声音，更改界面背景等功能。

**2、大体功能模块**



**3、运用到的所学的知识**

1、感觉最主要的键盘中断，由键盘事件响应相应的动作功能。

2、在指定位置打点，由点连成线，由线组成一个简单的基本图形，根据给给定的01点阵打点形成指定图案，还有界面的绘制。

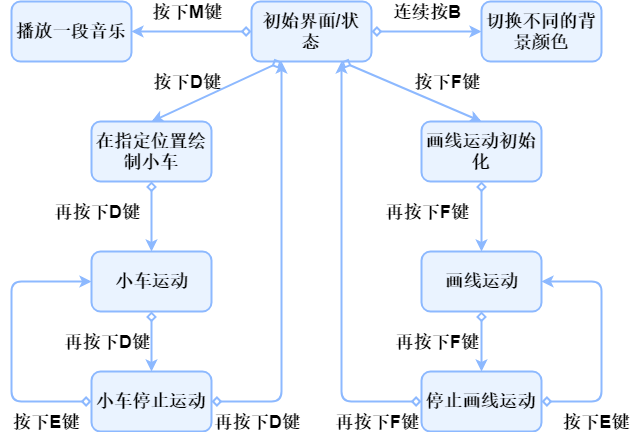
3、颜色RGB的配置

4、声音的频率的设置、开启、关闭，设置延迟等，8253和8255的使用。

5、普通中断的应用，使用普通中断完成各种动态任务，还有任务之间的切换。

6、各种寄存器的应用，简单的基本运算等。

**4、操作实现流程图**



注：小车运动和画线运动可以同时进行。

**功能说明：**

按下B，切换界面背景颜色

按下C，重置，回到系统最开始状态

按下M，播放一段声音

按下D，开始任务1，小车运动。第一次按下D，准备小车运动，再次按下D，小车开始运动，再按下D，小车暂停，如果此时再按D，会退出，回到页面初始状态。如果此时按E，小车会继续运动。

按下F，开始任务2，画线，第一次按下F，准备画线，再次按下F，开始画线，再按下F，停止画线。如果此时按F，会退出，回到页面初始状态。如果此时按E，会继续画线。

小车运动和画线可以同时进行，如果小车和画线都处于暂停状态，按下E，两个任务都会继续进行。

**代码：**

org 0x8400

*start:*

    mov al,0x13

    mov ah,0x00

    int 0x10

    mov ax,0x0a000

    mov es,ax

    cli

    mov word [ds:0x24],*int\_key*

    mov word [ds:0x26],0

    sti

    ;==========颜色=======

    push si

    mov si,*pallete*

    call *colors*

    pop si

    ;=====================

    ;==========底部栏=====

    call *bottom*

    ;=====================

    jmp *buttonplay*

*buttonplay:*

    mov al,0B6H

    out 43H,al

*lop:*                    ;一个死循环

    mov dx,18

    mov ah,7

    int 21H

    cmp word[*play*],1    ;播放声音

    je *play1*

    cmp word[*clear*],1

    je *play2*

    cmp word[*over*],1

    je *play3*

    cmp word[*over2*],1

    je *play5*

    cmp word[*carshow*],0

    jnz *play4*

    jmp *lop*

*play1:*

    call *tiger*

    mov word[*play*],0

    jmp *lop*

*play2:*

    sti

    mov word [ds:0x20],*int\_time1*

    mov word [ds:0x22],0

    mov word[*clear*],0

    mov word[*carshow*],0

    mov word[*start2*],0

    jmp *start*

*play3:*

    mov word[*over*],0

    sti

    mov word [ds:0x20],*int\_time1*

    mov word [ds:0x22],0

    jmp *lop*

*play4:*

    cmp word[*start2*],0

    je *lop*

    cmp word[*carshow*],0

    je *lop*

    sti

    mov word [ds:0x20],*int\_time0*

    mov word [ds:0x22],0

    sti

    mov word [ds:0x20],*int\_time2*

    mov word [ds:0x22],0

    jmp *play4*

*play5:*

    mov word[*over2*],0

    sti

    mov word [ds:0x20],*int\_time1*

    mov word [ds:0x22],0

    jmp *lop*

*int\_key:*            ;键盘中断

    mov dx,0x60     ;读取键盘缓冲区数据

    in  al,dx       ;out  cpu发送消息给输出设备

    cmp al,0x32     ;m  按下m播放一段声音

    je  *work1*

    cmp al,0x1e     ;a

    je  *work3*

    cmp al,0x30     ;b  更换背景

    je  *work4*

    cmp al,0x2e     ;c  清屏，回到最初始状态

    je  *work2*

    cmp al,0x20     ;d  小车运动

    je  *work5*

    cmp al,0x12     ;e  小车或划线暂停后，继续运动

    je  *work6*

    cmp al,0x21     ;f  画线

    je  *work7*

    jmp *kexit*

*work1:*                 ;播放声音

    mov word[*play*],1

    jmp *kexit*

*work2:*                 ;清屏，回到最开始状态

    mov word[*carshow*],0

    mov word[*start2*],0

    mov word[*clear*],1

    jmp *kexit*

*work3:*

    jmp *kexit*

*work4:*                 ;更换页面背景颜色

    cmp word[*siancolor*],0

    je *work4\_1*

    jmp *work4\_2*

*work4\_1:*

    mov word[*ncolor*],1

    mov word[*siancolor*],1

    jmp *work4\_4*

*work4\_2:*

    inc word[*ncolor*]

    cmp word[*ncolor*],6

    ja *work4\_3*

    jmp *work4\_4*

*work4\_3:*

    mov word[*siancolor*],0

    jmp *work4*

*work4\_4:*

    mov word[*slidex*],0

    mov word[*slidey*],0

    mov word[*slideh*],180

    mov word[*slidew*],320

    call *slide*

    jmp *kexit*

*work5:*                 ;小车运动

    cmp word[*carshow*],0

    je  *work5\_1*

    cmp word[*carshow*],1

    je  *work5\_2*

    cmp word[*carshow*],2

    je  *work5\_3*

    cmp word[*carshow*],3

    je  *work5\_4*

*work5\_1:*

    mov word[*carshow*],1

    mov word[*stop1*],0

    push di

    mov word[*mcolor*],6

    mov word[*countmax*],20

    mov word[*rowmax*],9

    mov word[*x*],300

    mov word[*y*],100

    mov di,*car*

    call *transit1*

    pop di

    mov word[*carshow*],1

    jmp *kexit*

*work5\_2:*            ;开始运动

    sti

    mov bx,0

    mov word [ds:0x20],*int\_time0*

    mov word [ds:0x22],0

    mov word[*carshow*],2

    jmp *kexit*

*work5\_3:*          ;暂停

    mov word[*stop1*],1

    mov word[*carshow*],3

    jmp *kexit*

*work5\_4:*          ;退出

    mov word[*carshow*],0

    mov word[*stop1*],0

    mov word[*clear*],1

    sti

    mov word [ds:0x20],*int\_time1*

    mov word [ds:0x22],0

    jmp *kexit*

*work6:*                 ;继续运动

    cmp word[*stop1*],1

    je *work6\_1*

    cmp word[*stop2*],1

    je *work6\_2*

    jmp *kexit*

*work6\_1:*

    mov word[*stop1*],0

    mov word[*carshow*],2

    jmp *work6*

*work6\_2:*

    mov word[*stop2*],0

    mov word[*start2*],1

    jmp *kexit*

*work7:*                 ;画线任务

    cmp word[*start2*],0

    je *work7\_0*

    cmp word[*start2*],1

    je *work7\_1*

    cmp word[*start2*],2

    je *work7\_2*

    cmp word[*start2*],3

    je *work7\_3*

    jmp *kexit*

*work7\_0:*            ;任务初始化

    mov word[*over2*],0

    mov word[*start2*],1

    mov word[*i*],0

    mov word[*j*],10

    jmp *kexit*

*work7\_1:*              ;任务开始

    mov word[*start2*],2

    sti

    mov word [ds:0x20],*int\_time2*

    mov word [ds:0x22],0

    jmp *kexit*

*work7\_2:*           ;任务暂停

    mov word[*start2*],3

    mov word[*stop2*],1

    jmp *kexit*

*work7\_3:*           ;任务结束

    mov word[*stop2*],0

    mov word[*clear*],1

    mov word[*start2*],0

    sti

    mov word [ds:0x20],*int\_time1*

    mov word [ds:0x22],0

    jmp *kexit*

*kexit:*

    mov dx,0x20 ;键盘中断结束

    mov al,0x61

    out dx,al

    iret

;===========任务1==============

*int\_time0:*      ;进程1  小车运动

    cmp word[*stop1*],1

    je *intexit*

    cmp word[*y*],0

    jnz *int1*

    cmp word[*x*],0

    je  *int3*

    jmp *int1*

*int1:*

    cmp word[*x*],0

    jnz *int2*

    mov word[*x*],290

    sub word[*y*],10

*int2:*

    dec word[*x*]

    push di

    mov word[*mcolor*],6

    mov word[*countmax*],20

    mov word[*rowmax*],9

    mov di,*car*

    call *draw*

    pop di

    jmp *intexit*

*int3:*

    mov word[*over*],1

    jmp *intexit*

*intexit:*

    mov al,0x20

    mov dx,0x20

    out dx,al

    iret

;=============================

;=========空进程==============

*int\_time1:*      ;空进程

    mov al,0x20

    mov dx,0x20

    out dx,al

    iret

;=============================

;===========任务2=============

*int\_time2:*      ;进程2  画线

    cmp word[*stop2*],1

    je  *int2\_exit*

    cmp word[*over2*],1

    je  *int2\_exit*

    cmp word[*i*],320

    jnz *int2\_1*

    cmp word[*j*],160

    je  *int2\_2*

    add word[*j*],10

    mov word[*i*],0

    jmp *int2\_1*

*int2\_1:*

    push bx

    push ax

    push di

    push dx

    mov bx,word[*i*]

    mov ax,word[*j*]

    call *cal*

    mov byte[es:bx],1

    inc word[*i*]

    pop dx

    pop di

    pop ax

    pop bx

    jmp *int2\_exit*

*int2\_2:*

    mov word[*over2*],1

    mov word[*i*],0

    mov word[*j*],0

    jmp *int2\_exit*

*int2\_exit:*

    mov al,0x20

    mov dx,0x20

    out dx,al

    iret

;=============================

;========界面底部==============

*bottom:*

    mov word[*slidex*],0

    mov word[*slidey*],0

    mov word[*slideh*],180

    mov word[*slidew*],320

    call *slide*

    mov word[*ncolor*],8

    mov word[*slidex*],0

    mov word[*slidey*],180

    mov word[*slideh*],200

    mov word[*slidew*],320

    call *slide*

    mov word[*ncolor*],1

    mov word[*slidex*],140

    mov word[*slidey*],180

    mov word[*slideh*],200

    mov word[*slidew*],1

    call *slide*

    mov word[*ncolor*],1

    mov word[*slidex*],180

    mov word[*slidey*],180

    mov word[*slideh*],200

    mov word[*slidew*],1

    call *slide*

    mov word[*ncolor*],1

    mov word[*x*],140

    mov word[*y*],180

    mov word[*w*],40

    call *linexy*

    mov word[*ncolor*],1

    mov word[*x*],140

    mov word[*y*],199

    mov word[*w*],40

    call *linexy*

    mov word[*ncolor*],7

    push di

    mov word[*countmax*],30

    mov word[*rowmax*],13

    mov word[*x*],145

    mov word[*y*],183

    mov di,*hsf*

    mov word[*mcolor*],6

    call *transit1*

    pop di

    ret

;=============================

;===========画四边形=============

*slide:*

    push bx

    push ax

    push di

    push bp

    mov di,0

*s0:*

    mov bp,word[*slidey*] ;60

    cmp bp,word[*slideh*] ;75

    je *s2*

    jmp *s1*

*s1:*

    mov bx,word[*slidex*]  ;20

    mov ax,word[*slidey*]  ;

    inc word[*slidey*]

    call *cal*

    add di,word[*slidew*]

    call *xy*

*xy:*

    mov dl,[*ncolor*]

    mov byte[es:bx],dl

    inc bx

    cmp bx,di

    jae *s0*

    jmp *xy*

*s2:*

    pop bp

    pop di

    pop ax

    pop bx

    ret

;==============================

;=======画横线=================

*linexy:*

    push ax

    push bx

    push di

    push dx

    mov bx,word[*x*]

    mov ax,word[*y*]

    call *cal*

    add di,word[*w*]

*line1:*

    cmp bx,di

    ja *line2*

    mov dl,[*ncolor*]

    mov byte[es:bx],dl

    inc bx

    jmp *line1*

*line2:*

    pop dx

    pop di

    pop bx

    pop ax

    ret

;=============================

;==========播放一段声音==========

*tiger:*          ;播放一段声音模块

    push ax

    push bx

    push cx

    push dx

    push di

    mov di,*twotiger*

    mov si,0

*m0:*

    mov ah,0bH

    int 21H

    cmp si,28

    je *m2*

    cmp si,32

    je *m2*

    cmp si,40

    je *m2*

    cmp si,44

    je *m2*

    jmp *m3*

*m2:*

    mov dx,12

    jmp *m4*

*m3:*

    mov dx,12

*m4:*

    mov ax,[di]

    call *sounder*

    call *soundon*

*m1:*

    call *delay*

    dec dx

    jnz *m1*

    call *soundoff*

    add si,2

    add di,2

    cmp ax,0

    jne *m0*

*exit:*

    pop di

    pop dx

    pop cx

    pop bx

    pop ax

    ret

;=========================

;======声音部分模块==========

*sounder:*    ;扬声器设置子程序

    push ax

    out 42H,al   ;计数初值写入计时器2

    mov al,ah

    out 42H,al

    pop ax

    ret

*soundon:*   ;开启扬声器子程序

    push ax

    in al,61H

    or al,03H

    out 61H,al

    pop ax

    ret

*soundoff:*   ;关闭扬声器子程序

    push ax

    in al,61H

    and al,0FDH

    out 61H,al

    pop ax

    ret

*delay:*      ;延迟系统

    push ax

    push cx

    mov cx,7000

*w1:*

    in al,61H

    and al,10H

    cmp al,ah

    je *w1*

    mov ah,al

    loop *w1*

    pop cx

    pop ax

    ret

;===========================

;======画图形模块==========

*transit1:*   ;用点阵画字或图形模块1

*locat:*

    push bx

    push ax

    push di

    mov bx,word[*x*]

    mov ax,word[*y*]

    call *cal*

    mov word[*location*],di

    pop di

    pop ax

    pop bx

    push bx

    push ax

    push dx

    push bp

    push cx

*judegment:*   ;判断0/1

    mov bp,[di]

    cmp bp,1

    je  *print*

    add di,2    ;下一个

    inc word[*count*]

    jmp *pexit*

*print:*

    mov bx,0

    mov ax,word[*row*]  ;乘法计算打印位置

    mov dx,320

    mul dx

    add bx,ax

    add bx,word[*count*]

    add bx,word[*location*]

    mov dl,[*mcolor*]

    mov byte[es:bx],dl   ;画点

    add di,2        ;下一个

    inc word[*count*]

    jmp *pexit*

*pexit:*

    mov cx,word[*countmax*]

    cmp word[*count*],cx  ;判断当前行是否遍历完

    ja  *transit2*               ;当前行遍历完切换到下一行

    jmp *judegment*

*transit2:*

    mov word[*count*],1   ;当前行从1开始

    inc word[*row*]

    mov cx,word[*rowmax*]

    cmp word[*row*],cx   ;判断是否跑完最后一行

    ja  *fexit*          ;跑完，退出

    jmp *judegment*

*fexit:*

    mov word[*count*],1

    mov word[*row*],1

    pop cx

    pop bp

    pop dx

    pop ax

    pop bx

    ret

*draw:*       ;用点阵画字或图形模块2

*locat\_draw:*

    push bx

    push ax

    push di

    mov bx,word[*x*]

    mov ax,word[*y*]

    call *cal*

    mov word[*location*],di

    pop di

    pop ax

    pop bx

    push bx

    push ax

    push dx

    push bp

    push cx

*judegment\_draw:*   ;判断0/1

    mov bp,[di]

    cmp bp,1

    je  *print1*

*print0:*

    mov bx,0

    mov ax,word[*row*]  ;乘法计算打印位置

    mov dx,320

    mul dx

    add bx,ax

    add bx,word[*count*]

    add bx,word[*location*]

    mov dl,[*ncolor*]

    mov byte[es:bx],dl   ;画点

    jmp *next\_draw*

*next\_draw:*

    add di,2    ;下一个

    inc word[*count*]

    jmp *pexit\_draw*

*print1:*

    mov bx,0

    mov ax,word[*row*]  ;乘法计算打印位置

    mov dx,320

    mul dx

    add bx,ax

    add bx,word[*count*]

    add bx,word[*location*]

    mov dl,[*mcolor*]

    mov byte[es:bx],dl   ;画点

    add di,2        ;下一个

    inc word[*count*]

    jmp *pexit\_draw*

*pexit\_draw:*

    mov cx,word[*countmax*]

    cmp word[*count*],cx  ;判断当前行是否遍历完

    ja  *transit2\_draw*               ;当前行遍历完切换到下一行

    jmp *judegment\_draw*

*transit2\_draw:*

    mov word[*count*],1   ;当前行从1开始

    inc word[*row*]

    mov cx,word[*rowmax*]

    cmp word[*row*],cx   ;判断是否跑完最后一行

    ja  *fexit\_draw*         ;跑完，退出

    jmp *judegment\_draw*

*fexit\_draw:*

    mov word[*count*],1

    mov word[*row*],1

    pop cx

    pop bp

    pop dx

    pop ax

    pop bx

    ret

;==========================

;========设置颜色===========

*colors:*     ;颜色模块

    push bp

    mov bp,0

*c1:*

    mov dx,0x3c8

    mov al,[si]

    out dx,al

    mov dx,0x3c9

    mov al,[si+2]

    out dx,al

    mov al,[si+4]

    out dx,al

    mov al,[si+6]

    out dx,al

    add bp,8

    add si,8

    cmp bp,72

    jae *cexit*

    jmp *c1*

*cexit:*

    pop bp

    ret

;=======================

;==============================================变量定义==============================================

*cal:*        ;计算位置

    mov dx,320

    mul dx

    add bx,ax

    mov di,bx

    ret

*car* dw 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2

    dw 0,0,0,0,0,0,0,1,1,1,1,1,1,1,0,0,0,0,0,2

    dw 0,0,0,0,0,0,1,1,0,0,1,0,0,1,1,0,0,0,0,2

    dw 0,0,0,0,0,1,1,1,0,0,1,0,0,1,1,1,0,0,0,2

    dw 0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2

    dw 0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2

    dw 0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2

    dw 0,0,0,0,1,1,1,1,0,0,0,0,0,1,1,1,1,0,0,2

    dw 0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,0,0,0,2

    dw 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2

*twotiger* DW 11C9H,0FBEH,0E1FH,11C9H,11C9H,0FBEH,0E1FH,11C9H,0E1FH,0D5AH,0BE3H,0E1FH,0D5AH,0BE3H

         DW 0BE3H,0A97H,0BE3H,0D5AH,0E1FH,11C9H,0BE3H,0A97H,0BE3H,0D5AH,0E1FH,11C9H

         DW 11C9H,0BE3H,11C9H,11C9H,0BE3H,11C9H,00H

*pallete*     dw 1,70,90,50

            dw 2,207,207,207

            dw 3,169,169,169

            dw 4,223,223,223

            dw 5,255,255,255

            dw 6,0,0,0

            dw 7,153,153,153

            dw 8,246,246,246

            dw 9,255,0,0

*hsf* dw 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

    dw 0,1,1,1,0,0,1,1,1,0,0,0,0,1,1,1,1,1,0,0,0,0,1,1,1,1,1,1,1,1

    dw 0,1,1,1,0,0,1,1,1,0,0,0,1,1,1,0,1,1,1,0,0,0,1,1,1,1,1,1,1,1

    dw 0,1,1,1,0,0,1,1,1,0,0,0,1,1,1,0,1,1,1,0,0,0,1,1,1,0,0,0,0,0

    dw 0,1,1,1,0,0,1,1,1,0,0,0,1,1,1,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0

    dw 0,1,1,1,1,1,1,1,1,0,0,0,1,1,1,1,1,0,0,0,0,0,1,1,1,0,0,0,0,0

    dw 0,1,1,1,1,1,1,1,1,0,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,1,1,1,0

    dw 0,1,1,1,0,0,1,1,1,0,0,0,0,0,0,0,1,1,1,0,0,0,1,1,1,1,1,1,1,0

    dw 0,1,1,1,0,0,1,1,1,0,0,1,1,1,0,0,0,1,1,1,0,0,1,1,1,0,0,0,0,0

    dw 0,1,1,1,0,0,1,1,1,0,0,0,1,1,1,0,0,1,1,1,0,0,1,1,1,0,0,0,0,0

    dw 0,1,1,1,0,0,1,1,1,0,0,0,1,1,1,1,1,1,1,0,0,0,1,1,1,0,0,0,0,0

    dw 0,1,1,1,0,0,1,1,1,0,0,0,0,1,1,1,1,1,0,0,0,0,1,1,1,0,0,0,0,0

    dw 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

*count*   dw 1  ;列

*row* dw 1  ;行

*countmax*    dw 30

*rowmax*  dw 13

*location*    dw 0

*ncolor* dd 7

*siancolor* dd 0

*mcolor* dd 0

*stop* dw 0

*stop1* dw 0

*stop2* dw 0

*clear* dw 0

*play* dw 0

*over* dw 0

*start2* dw 0

*over2* dw 0

*together* dw 0

*i* dw 0

*j* dw 0

*x* dw 0

*y* dw 0

*w* dw 0

*slidex*  dw 10

*slidey*  dw 120

*slideh*  dw 140

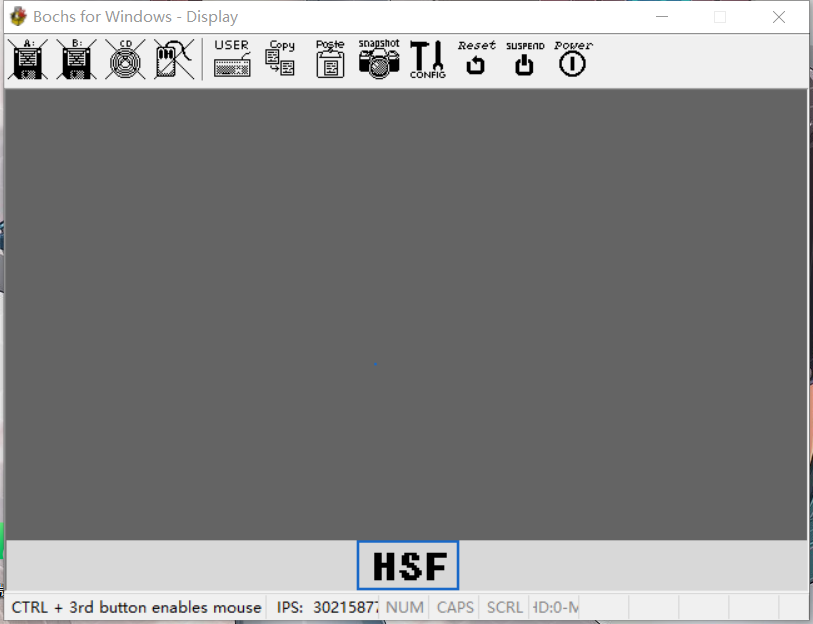
*slidew*  dw 40

*carshow* dw 0

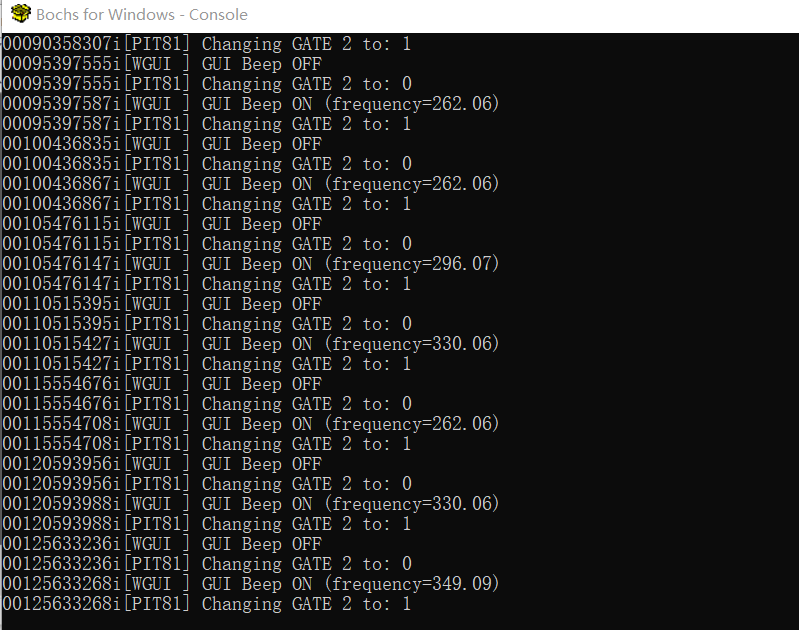
;========================================================================================

**贴图：**

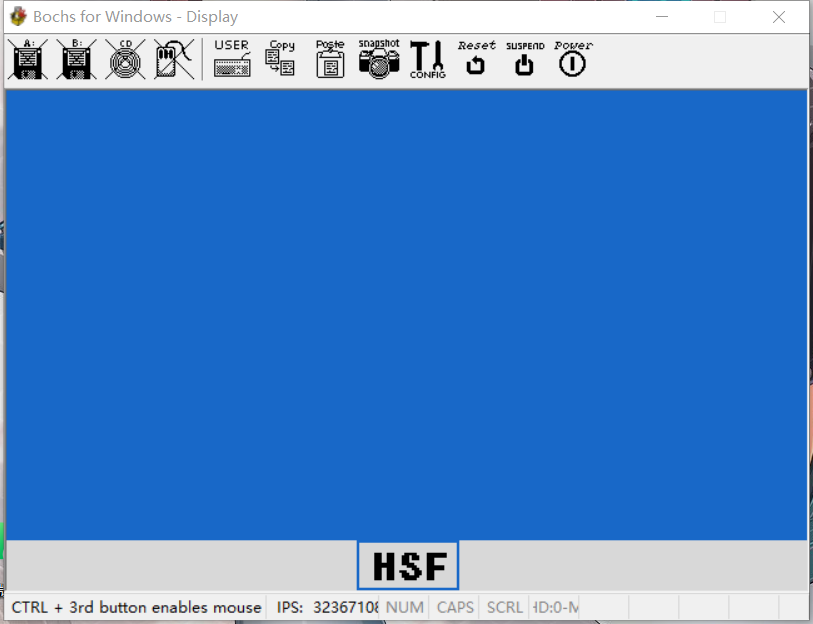
**初始界面：**

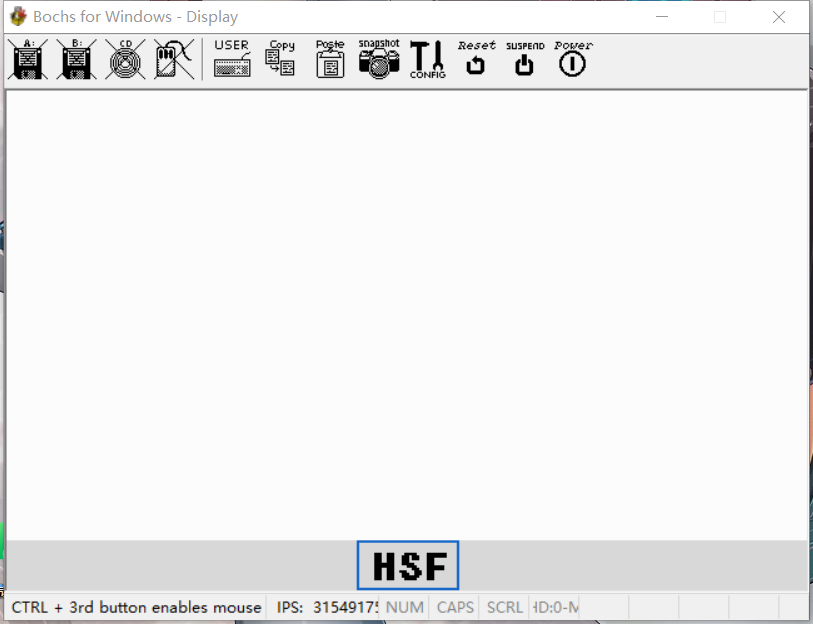


**按m会播放一段设置好的声音**

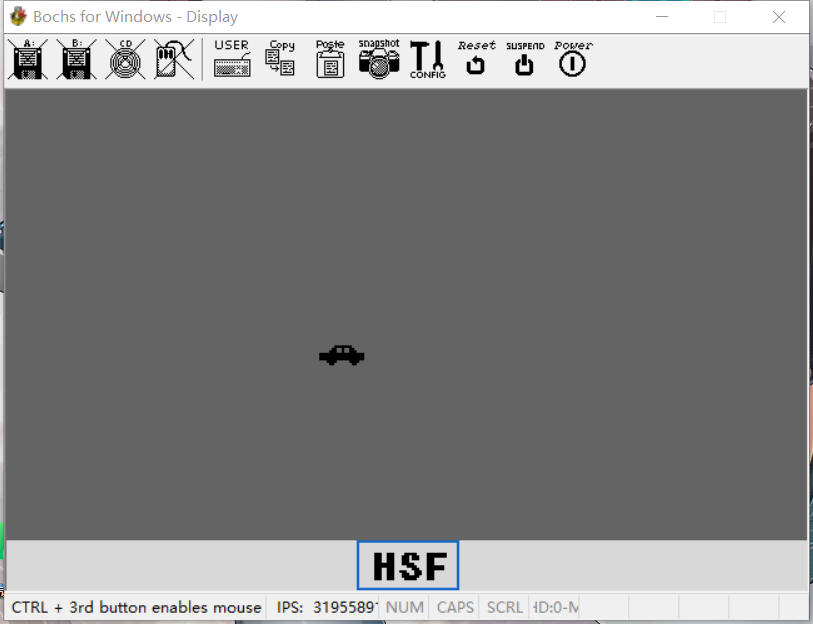


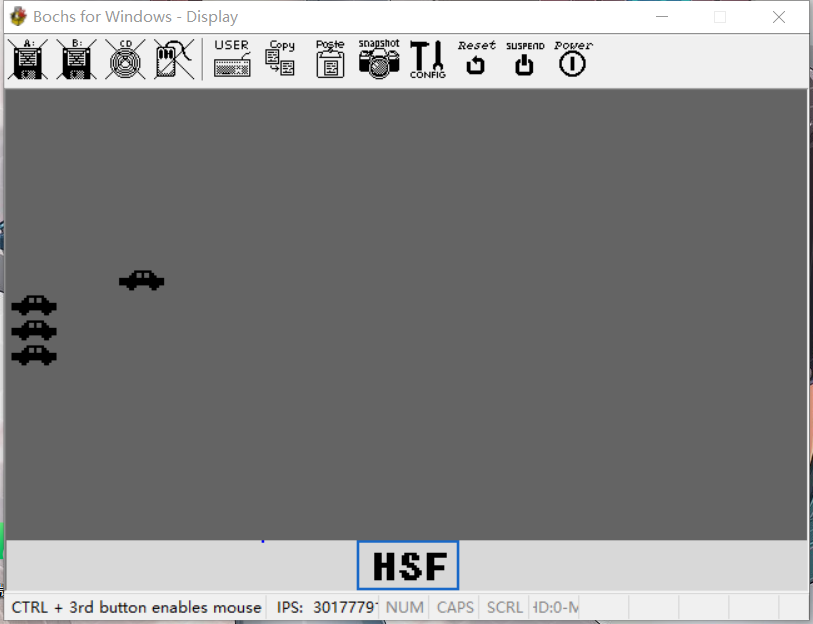
**按B切换背景颜色**



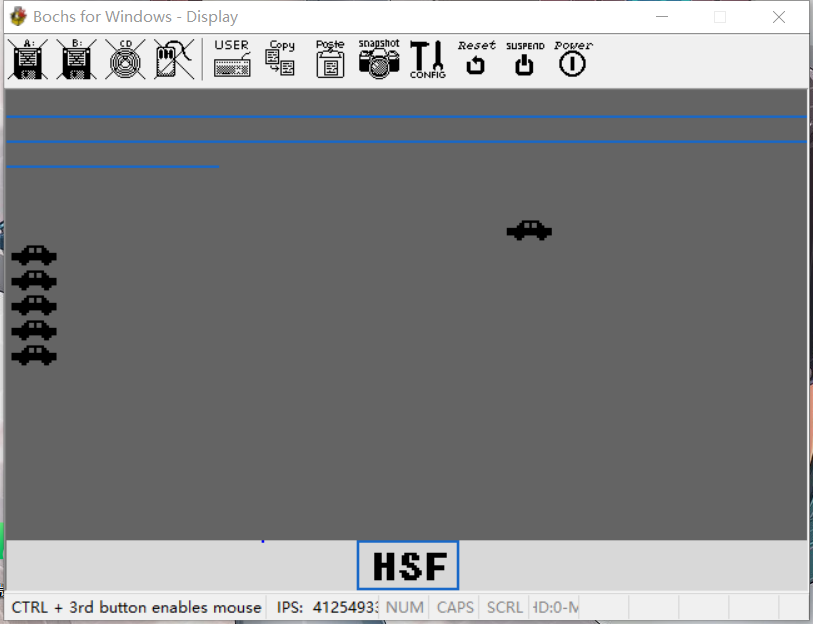


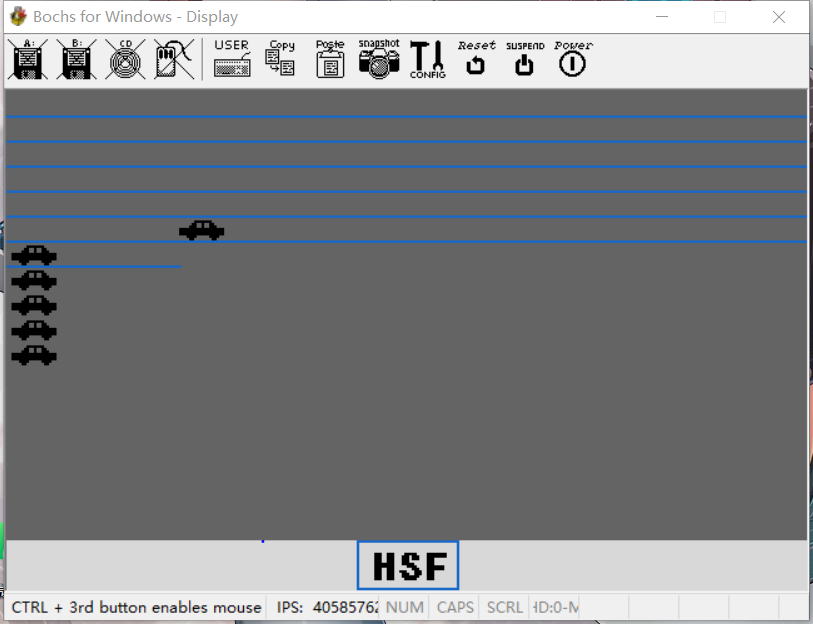
**按D小车运动**





**按F画线任务：**





**5、心得体会**

需要在考试之前做完，而且还有留有复习时间所有做的有点匆忙，有些地方还不够完善，但花费的时间也是不少的。

一开始是想先做个播放一首歌曲，但不知道为什么，设置一个声音频率数组给想让它按顺序播放，但效果是一直只播放一个音，但是另外一边的面板上，那些声音频率也都是按着我给的数组来一个个遍历的，不是很懂，就去网上学了学别人的写法，加了定时器，再改改延时，才终于有最后的效果，虽然怎么听都不像一首歌，但觉得声音效果也比之前好多了，而且就搞这个声音就做了好几天，也就没深究，去写别的模块。其他模块都没什么大问题，有了自己之前就做过的作业为基础，改成一个模块，再使用，花些时间就能写出来，但也用了两三天，先是一直思考要做什么，再来一个一个模块的试试效果，再除掉一些小小的问题，比如是不知道打图案时，左上角总会打个多余的点出来，最后再整理整理就已经星期五了，时间还真紧迫。还有一些问题不是很懂，比如我小车从右向左行驶，每次换了一行运动时，最左边会多画出一个多余的小车，每次换行都有一个，但是更换一下背景就又没掉了，不是很懂。感觉汇编学起来很麻烦，晚上又资料少，有些问题都查查不到，写条件判断的时候也怪麻烦的，一不小心就会有问题，然后找半天才找到。寄存器又优先，不过这样画时间写东西的过程比复习考试有意思多了。