

**《微型计算机技术及应用》**

**课程设计报告**

专业班级： 电子信息工程171班

小组组别： 第二组

小组成员： 阮君韬 张佳钦 张玉杰

指导老师： 舒红波

2018 年 7 月 5日

**小组分工： 程序设计：张佳钦、阮君韬**

**报告编写：张玉杰**

1. **功能简介**

本程序实现文章阅读功能，具有开始欢迎界面和再见界面。首先，启动程序后暂停在已刷入固件的界面，等待回车键确认。若不确认，一直停留在此处。若按键响应中断检测到回车按键被按下，进入欢迎界面，显示“welcome to my computer”这样一句话。然后，执行延时函数，大约等候4秒后进入，正文显示，显示“Youth is not a time……by deserting our ideals.” 的界面。其采用定时器中断的输出方式。为了方便阅读，每行之间采取空一行。最后，进入图形模式，在320\*200的屏幕边缘画一个红色的边框。画框结束后，系统清屏，出现“再见”界面。

1. **代码简介**

主要分为三大模块，分别为欢迎模块，文本显示模块和画框、再见模块。代码中有很多的功能代码被包装成模块形式，通过call可以直接调用。模块化的编程，带来了很多方便。

1. 欢迎模块

由于uboot已经刷入部分固件代码，在启动程序后，会显示“My OS booting…”。但此刻，按键中断一直在执行，等待回车按键被按下。当回车键被按下时，系统进入文本模式，进行“welcome to my computer”显示。但是，显示欢迎语句之前对前面“My OS booting…”地方通过输出空格的方式进行清除。最后，进行一小段时间的延时，停留一会时间。

1. 文本显示模块

此部分依旧采用文本模式。在上一部分显示欢迎语句之后，我们在文本模式下，对屏幕进行清屏，这样可以避免显示混乱。通过定时器的功能进行显示，只要触发一次定时器中断，就输出显示一个字母。

当一行显示满以后，对bx加上160，进行隔行显示。当所有文本显示完毕后，延时了一小段时间，做一个画面停留。

1. 画框、再见模块

画框和再见的显示均在图形模式下进行。对于画框，并不复杂，在VGA320x200x8bit显示模式下，通过调用事先编写好的HengXian和ShuXian两函数实现。对于显示汉字“再见”，通过取字模软件，提取16×16的汉字信息。

1. 程序流程图



1. **程序代码**

org 0x8400

jmp start

string db 'welcome to my computer'

string1 db 'Youth is not a time of life; it is a state of mind; it is not a matter of rosy cheeks, red lips and supple knees; it is a matter of the will, a quality of the imagination, a vigor of the emotions; it is the freshness of the deep springs of life.Youth means a temperamental predominance of courage over timidity, of the appetite for adventure over the love of ease. This often exists in a man of 60 more than a boy of 20. Nobody grows old merely by a number of years. We grow old by deserting our ideals. '

num db 0

key db 0

string2 db 00h, 08h, 7Fh,0FCh, 01h, 00h, 01h, 10h, 1Fh,0F8h, 11h, 10h, 11h, 10h, 1Fh,0F0h

db 11h, 10h, 11h, 14h,0FFh,0FEh, 10h, 10h, 10h, 10h, 10h, 10h, 10h, 50h, 10h, 20h

db 00h, 10h, 1Fh,0F8h, 10h, 10h, 11h, 10h, 11h, 10h, 11h, 10h, 11h, 10h, 11h, 10h

db 11h, 10h, 12h, 10h, 12h, 90h, 04h, 80h, 04h, 82h, 08h, 82h, 30h, 7Eh,0C0h, 00h

;再

;见

start:

sti

mov ax,0xb800

mov es,ax

mov ax,0

mov ds,ax

mov byte[ds:0x8200],0

mov word [ds:0x24],int\_key

mov word [ds:0x26],0

m: cmp byte[key],0x1c

jne m

display\_key:

;mov bx,0

mov byte[es:bx],' '

mov byte[es:bx+1],0x0c

add bx,2

cmp bx,100

jne display\_key

call Welcome

call Delay

sti

mov ax,0xb800 ;指向文本模式的显示缓冲区

mov es,ax

mov ax,0

mov ds,ax

mov si,0

mov bx,0

call clear

mov dx,0

mov word [ds:0x20],int\_time0

mov word [ds:0x22],0

x9:

jmp $

x6:

cli

call GoodBye

jmp x9

;\*\*\*\*\*\*\*\*\*\*\*

display2:

mov ax,0a000h

mov es,ax

cld

mov cx,2

lea si,[string2]

dis1:

push cx

push si

push di

mov cx,16

disword:

movsw

add di,78

loop disword

pop di

pop si

pop cx

add si,32

add di,2

loop dis1

ret

GoodBye:

call Delay

mov al,0x13 ;vga320x200x8bit显示

mov ah,0x00

int 0x10

mov ax,0x0a000 ;往显存（0xa0000开始）

mov ds,ax

call ChangeColor

call DrawBorder

call Delay

mov al,0x12

mov ah,0x00

int 0x10

mov ax,0

mov ds,ax

mov di,4820

call display2

ret

int\_key:

push ax

;push bx

push ds

mov ax,0

mov dx,ax

mov dx,0x60 ;采集扫描码

in al,dx

mov [key],al

mov dx,0x20

mov al,0x61

out dx,al

pop ds

;pop bx

pop ax

iret

int\_time0:

mov al,[ds:string1+si]

mov byte[es:bx],al

mov byte[es:bx+1],0x0c

add bx,2

add si,1

cmp si,80

je space

cmp si,160

je space

cmp si,240

je space

cmp si,320

je space

cmp si,400

je space

cmp si,480

je space

x7: cmp bx,1970

je x6

x5: cmp si,504

je gl

x4: mov al,0x20

mov dx,0x20

out dx,al ;EOI=1,中断结束

iret

gl: mov si,503

jmp x4

space: add bx,160

jmp x5

clear: mov byte[es:bx],al

inc bx

mov byte[es:bx],0x0c

inc bx

cmp bx,4000

jbe clear

mov bx,0

ret

Delay:

push bx

push cx

mov cx,0

x1:mov bx,500

x2:

dec bx

cmp bx,0

jne x2

loop x1

pop cx

pop bx

ret

Welcome:

mov bx,0

aa: mov byte [es:bx],' '

mov byte [es:bx+1],0x0c

add bx,2

cmp bx,100

jne aa

mov si,0

mov bx,0

bb: mov al,[ds:string+si]

mov byte [es:bx+1800],al

mov byte [es:bx+1801],0x0c

add bx,2

inc si

cmp si,22

jne bb

ret

DrawBorder:

mov ax,320

mov bx,0

mov cx,319

mov dx,0

call HengXian

mov ax,320

mov bx,199

mov cx,319

mov dx,0

call HengXian

mov ax,320

mov bx,0

mov cx,199

mov dx,0

call ShuXian

mov ax,320

mov bx,319

mov cx,199

mov dx,0

call ShuXian

ret

HengXian:

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;ax=320

;bx=y

;cx=x2

;dx=x1

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

push ax ;堆栈

push bx

push cx

push dx

sub cx,dx ;cx-dx得到循环次数

mov di,dx

mul bx ;bx\*ax

add di,ax

inc cx

a: mov byte[ds:di],1 ;往显存里放数据（颜色）

inc di

dec cx

jnz a

pop dx

pop cx

pop bx

pop ax

ret

ShuXian:

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;ax=320

;bx=x=50

;cx=y1=150

;dx=y2=50

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

push ax

push bx

push cx

push dx

sub cx,dx ;cx-dx循环次数

mul dx

add bx,ax

inc cx

c:

mov byte[ds:bx],1 ;往显存里放数据（颜色）

add bx,320

dec cx

jnz c

pop dx

pop cx

pop bx

pop ax

ret

ChangeColor:

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

mov dx,0x3c8

mov al,1

out dx,al

mov dx,0x3c9 ;R

mov al,255

out dx,al

mov dx,0x3c9 ;G

mov al,0

out dx,al

mov dx,0x3c9 ;B

mov al,0

out dx,al

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

mov dx,0x3c8

mov al,2

out dx,al

mov dx,0x3c9 ;R

mov al,0

out dx,al

mov dx,0x3c9 ;G

mov al,255

out dx,al

mov dx,0x3c9 ;B

mov al,0

out dx,al

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

mov dx,0x3c8

mov al,3

out dx,al

mov dx,0x3c9 ;R

mov al,0

out dx,al

mov dx,0x3c9 ;G

mov al,0

out dx,al

mov dx,0x3c9 ;B

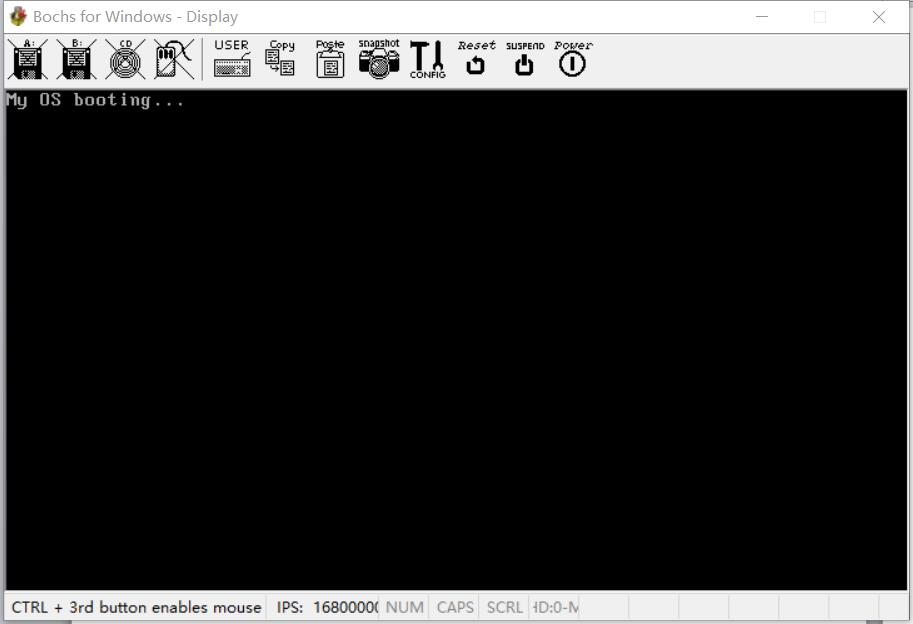
mov al,255

out dx,al

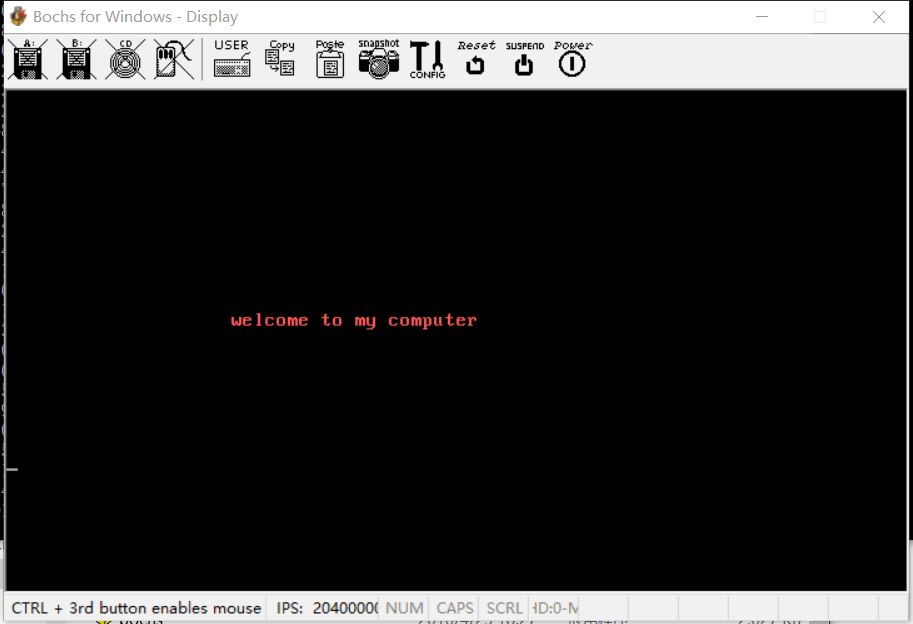
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ret

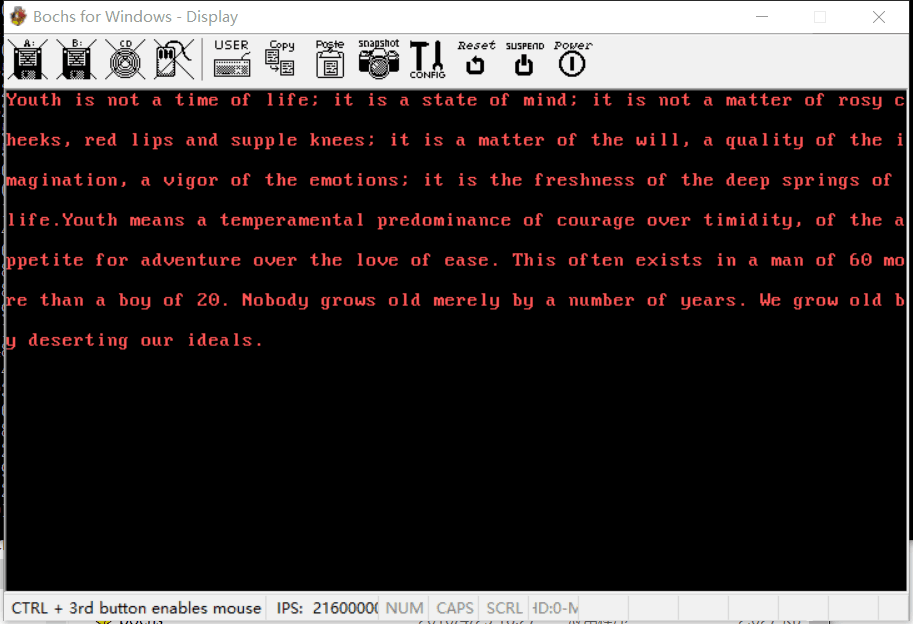
1. **测试结果**
2. 程序起始界面



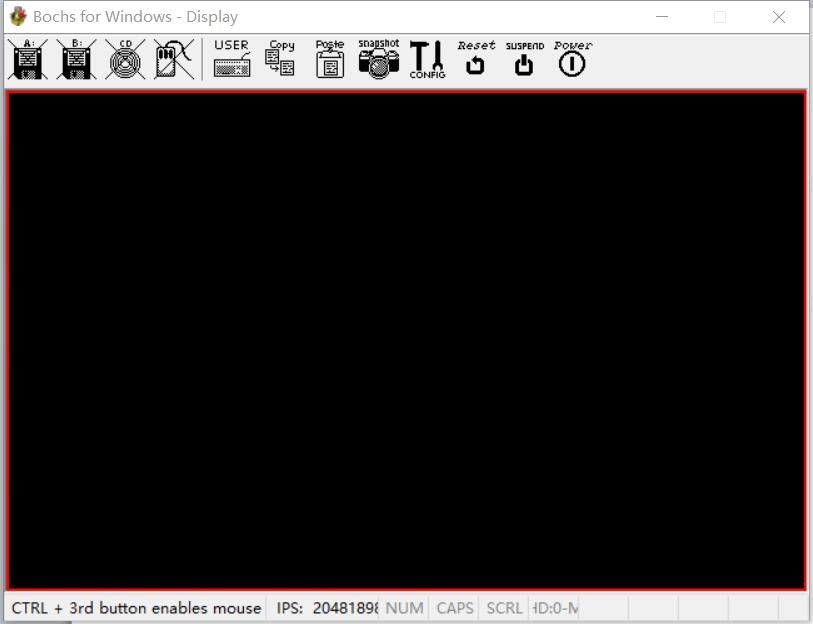
1. 欢迎界面



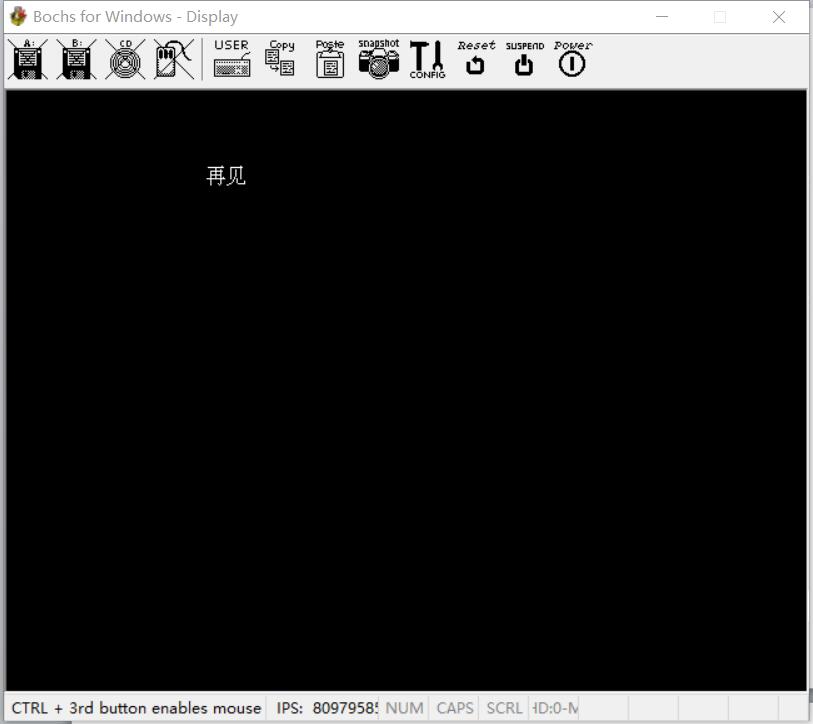
1. 文本显示界面



1. 画框界面



1. 再见界面



1. **心得体会**

在这次课程设计中，我们收获了许多。在做的过程中，小组之间首先进行讨论，构建设计思路，然后进行代码编写。代码实现是我们在作业过程中较为困难的一个部分，微机编程语言与C语言有很大不同，在C语言中，模块函数可以直接调用，但在微机汇编语言中，模块函数没有模板，需要自行编写，这是一个比较困难的过程，我们在进行代码编写时，通过在网上查资料，参考别人的代码，把学到的东西融入到自己的代码中来实现系统的功能。同时，我们理解到，自行编写模块化语言也是一种较为有效的学习方式，能够锻炼我们编写程序的能力。语言模块化也缩短了语句，节省了存储空间，增强了代码的可利用性，使代码看起来整洁规范。通过课设我们对系统的操作有了一个宏观的认识，在对文本模式、图形模式、中断、键盘输入等的操作实现中对它们有了更加清晰的理解。我们意识到，语言的神奇性和无限制性，只要有想法，语言就可以用代码的形式来实现它，要学好微机语言，我们还需要学习更多。

最后，感谢舒老师在课程设计上给予我们的指导、提供给我们的支持和帮助，这是我们能顺利完成这次任务的主要原因，更重要的是老师帮我们解决了许多技术上的难题。