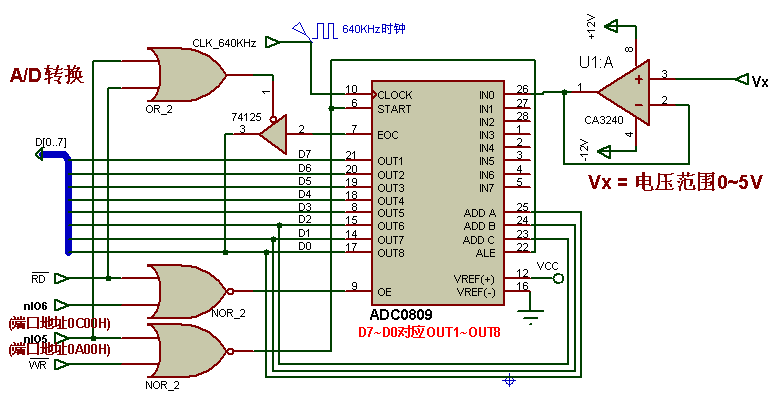
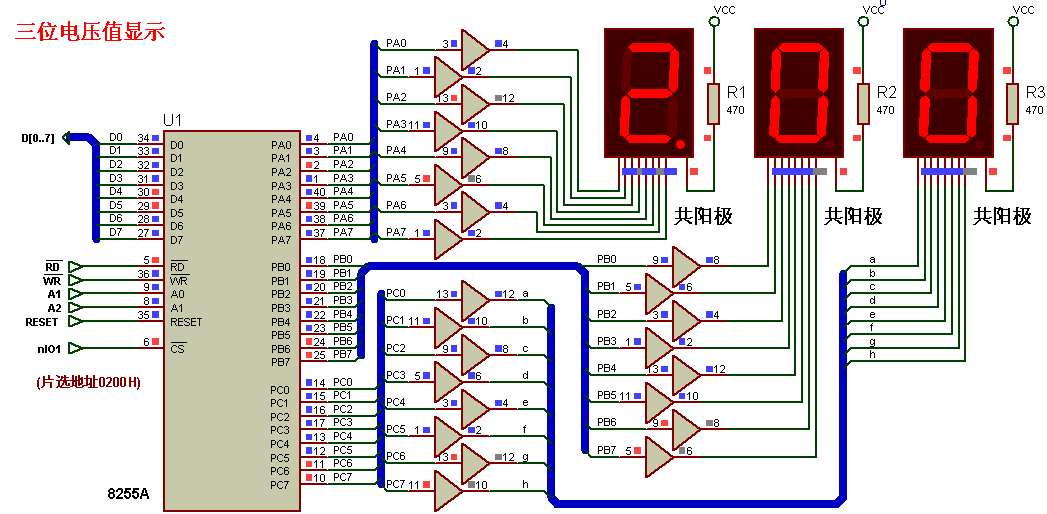
**五. 综合设计题(20分)**

给出如下一种简易的电压表硬件电路原理图，其目的是利用ADC0809，检测输入电压，并显示电压值的三位电压表。Vx是输入电压（范围是0V到5V），每次AD转换后，把与其相对应的电压值显示到三位七段数码管上（显示格式如“2.00”，表示检测到的输入电压是2.00V）。信号nIO1、nIO5、nIO6是经过和地址线A15~A9译码后，得到片选信号。8255A的A端口地址为nIO1=0200H，8255A的B端口地址为nIO1+2=0202H, 8255A的C端口地址为nIO1+4=0204H， 8255A的控制端口地址为nIO1+6=0206H。nIO5是写入选择AD通道和启动AD转换的端口地址，同时也是查询AD是否转换结束的端口地址。nIO6是AD转换结束时，读取AD值的端口地址。

试完成下列要求：

1. 写出显示0对应的七段数码管译码值；(5分)
2. 编写8255A初始化子程序(init\_8255)；AD转换值到显示电压值变换子程序(calc\_vol)。(5分)
3. 编写显示电压子程序(display\_vol)和主程序(main)。(10分)





**8255A编程字**：



**五. 综合设计题(20分)**

**（1）0c0h** (**5**分)

**（2） (5分)**

init\_8255 proc near

mov dx, PORT\_CTR\_8255

mov al, 10000000b ;初始化8255控制字

out dx, al

ret

init\_8255 endp

;---3.8253初始化--------------------------------

init\_8253 proc near

mov dx, PORT\_CTR\_8253

mov al, 00110100b ;初始化8253控制字,计数0，r/w低8位 ,高8位，方式2，二进制计数

out dx, al

mov dx, PORT\_COUNTER0\_8253

mov ax, 20000 ;计数常数 ; 20ms/(1/1Mhz) =20000

out dx, al ;写低8位

mov al, ah

out dx, al ;写高8位

ret

init\_8253 endp

**（3）(10分)**

.data

;软件七段译码表

led\_table db 0c0h,0f9h,0a4h,0b0h,99h,92h,82h,0f8h,80h,98h,88h,83h,0c6h,0a1h,86h,8eh; 0~f段码

adc\_val db 0 ;ADC值

vol\_x100 dw 0 ; Vx \* 100= (5\*Nx/256)\*100 = 1.95\*Nx 小于65535

;---代码段-----------------------------------------------

.code

;---0.主程序---------------------------------

.startup

main:

call init\_vct\_table ;中断时用

call init\_8255 ;8255A初始化

call init\_8253 ;中断时不用

;call init\_8259 ;仿真时不用

next\_loop:

;call isr\_20ms\_adc ;中断时不用

call calc\_vol\_x100;计数电压值\*100

call display\_vol ;显示电压值

;delay 0ffffh ; 仅仿真时用

jmp next\_loop

ret

;---1.中断向量表初始化--------------------------------

init\_vct\_table proc near

;mov bx, 4\*12h ; n = 12H, IR2有中断请求输入, ICW2=10H

mov bx, 4\*2 ; n = 2, NMI非屏蔽中断

mov ax, 0

mov es, ax

mov ax, offset isr\_20ms\_adc

mov es:[bx], ax

mov ax, seg isr\_20ms\_adc

mov es:[bx+2], ax

ret

init\_vct\_table endp

;---2.8255初始化--------------------------------

init\_8255 proc near

mov dx, PORT\_CTR\_8255

mov al, 10000000b ;初始化8255控制字

out dx, al

ret

init\_8255 endp

;---3.8253初始化--------------------------------

init\_8253 proc near

mov dx, PORT\_CTR\_8253

mov al, 00110100b ;初始化8253控制字,计数0，r/w低8位 ,高8位，方式2，二进制计数

out dx, al

mov dx, PORT\_COUNTER0\_8253

mov ax, 20000 ;计数常数 ; 20ms/(1/1Mhz) =20000

out dx, al ;写低8位

mov al, ah

out dx, al ;写高8位

ret

init\_8253 endp

;---4.8259初始化--------------------------------

init\_8259 proc near

mov dx, PORT\_8259\_0

mov al, 00010010b ;初始化8259控制ICW1

out dx, al

mov dx, PORT\_8259\_1

mov al, 08H; ICW2

out dx, al

mov al, 01H; ICW4

out dx, al

ret

init\_8259 endp

;---5.中断服务程序-------------------------------

isr\_20ms\_adc proc far

push ax

push dx

mov dx, PORT\_DAC2ADC

mov ah, -1

mov cx,255

next\_count:

inc ah

mov al, ah

out dx, al; 输出DAC值

in al, dx; 读回比较器状态

and al, 01h

cmp al, 0

loopne next\_count

mov adc\_val,ah;ah中是ADC值, 保存

pop dx

pop ax

;ret ;不用中断时用

iret ; 中断时用

isr\_20ms\_adc endp

;---计算电压值\*100--------------

calc\_vol\_x100 proc near

push ax

push bx

mov ah, 0

mov al, adc\_val

mov bx, 195 ; Vx \* 10000= (5\*Nx/256)\*10000 = 195\*Nx, 小于65535

mul bx ;dxax = ax\*bx

mov bx,100

div bx ; ax = dxax/bx

mov vol\_x100, ax ; 保存电压值\*100

pop bx

pop ax

ret

calc\_vol\_x100 endp

;---7.七段数码管显示电压值-------------------------------

display\_vol proc near

push ax

push bx

push cx

push dx

mov bx, offset led\_table

mov cx, vol\_x100

mov al, 0

to\_w100:

cmp cx, 100

jb display\_w100

inc al

sub cx, 100

jmp to\_w100

display\_w100:

xlat ;al <-- [bx + al]

and al,01111111b;显示小数点

mov dx, PORT\_A\_8255

out dx, al ;显示百位

mov al, 0

to\_w10:

cmp cx, 10

jb display\_w10

inc al

sub cx, 10

jmp to\_w10

display\_w10:

xlat

or al,10000000b;不显示小数点

mov dx, PORT\_B\_8255

out dx, al ;显示十位

display\_w1:

mov al, cl

xlat

or al,10000000b;不显示小数点

mov dx, PORT\_C\_8255

out dx, al ;显示个位

pop dx

pop cx

pop bx

pop ax

ret

display\_vol endp

end

1. 选用如下图给出的元器件设计一个恒温箱温度采集控制系统。该系统有两个状态：设置状态和控制状态。在设置状态时，通过键盘可以修改恒温箱的**设定温度**；在控制状态时，用开关量输出进行简单控制。**检测温度**与**设定温度**进行比较，当**检测温度**小于**设定温度**时，控制继电器加热；当**检测温度**大于**设定温度**时，关闭加热。当有按键时，发出**1kHz声音**，用于按键提示。

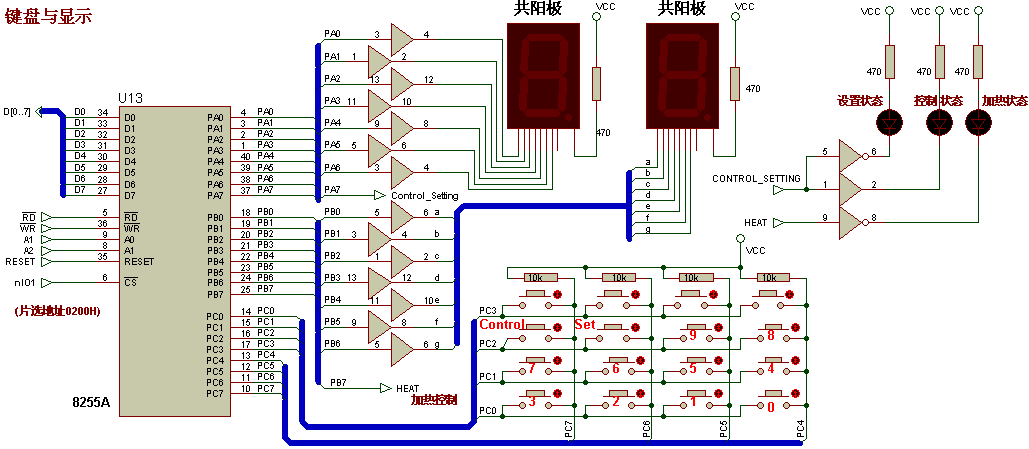
系统有两位七段数码管显示温度值（0~99℃）。在设置状态时，系统显示设定温度；在控制状态时，系统显示当前检测温度。系统通过4x4键盘输入设定温度值和启动控制，键盘有0~9键、Setting键和Control键共12个键可用。

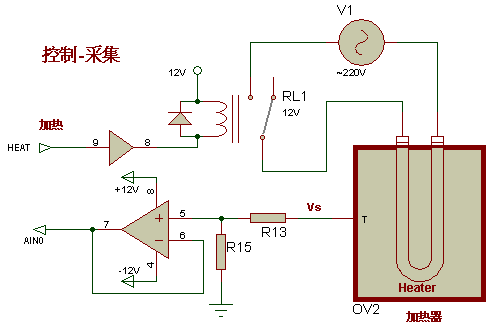
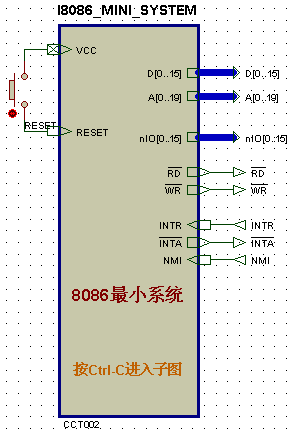
画出系统的硬件连接原理图，并标明分配给各元器件的端口地址。

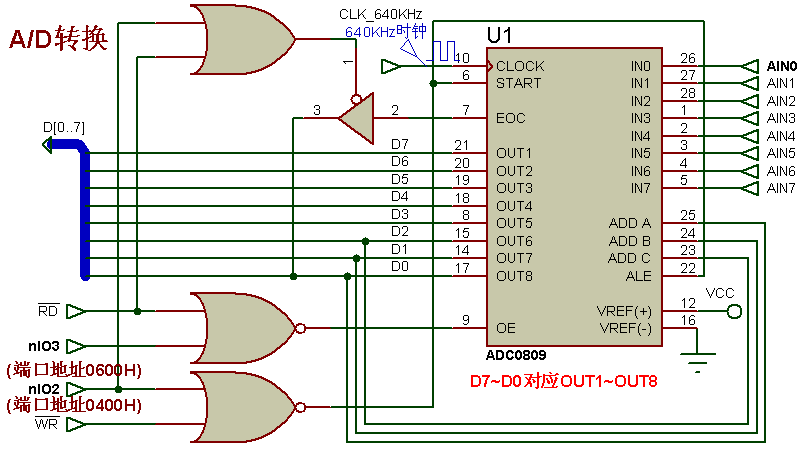
写出“0”对应的七段数码管译码值；编写8255、8253初始化程序；

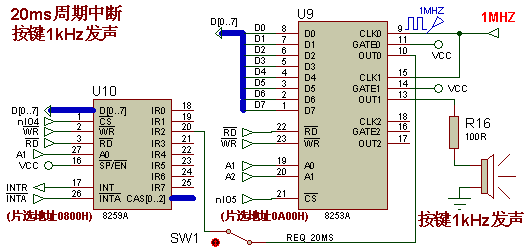
（3）编写AD转换子程序（adc）、显示子程序（display）、按键识别子程序（key）和主程序（main）。

（1）硬件原理图如下









（2）写出“0”对应的七段数码管译码值；编写8255、8253初始化程序；

“0”对应的七段数码管译码值是：0C0H

;8255初始化

init\_8255 proc near

mov dx, PORT\_CTR\_8255

mov al, 10001000b ;初始化8255控制字

out dx, al

ret

init\_8255 endp

;8253\_计数器0初始化---20ms中断请求

init\_8253 proc near

mov dx, PORT\_CTR\_8253

mov al, 00110100b ;初始化8253控制字,计数0，r/w低8位 ,高8位，方式2，二进制计数

out dx, al

mov dx, PORT\_COUNTER0\_8253

mov ax, 20000 ;计数常数 ; 20ms/(1/1Mhz) =20000

out dx, al ;写低8位

mov al, ah

out dx, al ;写高8位

ret

init\_8253 endp

;开始发声，8253计数器1，方式3， 1kHz方波----

startBeep proc near

mov dx, PORT\_CTR\_8253

mov al, 01110110b ; 初始化8253控制字,计数1，r/w低8位 ,高8位，方式3，二进制计数

out dx, al

mov dx, PORT\_COUNTER1\_8253

mov ax, 1000 ; 计数常数 ; 1mhz/1khz =1000

out dx, al ; 写低8位

mov al, ah ; 写8位

out dx, al

ret

startBeep endp

;----5.2停止发声，方式0----

stopBeep proc near

mov dx, PORT\_CTR\_8253

mov al, 01110000b ; 初始化8253控制字，计数1,方式0

out dx, al

ret

stopBeep endp

;----5.3发声100ms----

beep\_200ms proc near

push dx

call startBeep

delay 0ffh

call stopBeep

pop dx

ret

beep\_200ms endp

（3）编写AD转换子程序（adc）、显示子程序（display）、按键识别子程序（key）和主程序（main）。

Adc proc near

; 启动AD转换

mov dx, PORT\_START\_0809

mov al, 0

out dx, al ;选择通道0

mov is\_adc\_started, 1; 下次是查询状态

sample\_ch0:

; 查询转换结束引脚eoc，非阻塞方式

mov dx, PORT\_EOC\_0809

in al, dx

test al, 00000001b ;eoc引脚为高电平？

jz exit\_isr

;eoc变高到此，输入adc转换结束数值

mov dx, PORT\_DATA\_0809

in al, dx

mov [sample\_ch0\_val], al; 保存AIN0采集数字量

mov is\_adc\_started, 0 ; 下次是启动AD状态

call calc\_ad2tmp; 计算当前温度值

exit\_isr:

ret

Adc endp

display proc far

push ax

push bx

push cx

mov bx, offset led\_table

cmp is\_in\_control\_state, 1

je get\_cur\_tmp\_val

mov cl, setting\_tmp\_val

jmp start\_display

get\_cur\_tmp\_val:

mov cl, sample\_tmp\_val

start\_display:

mov al, 0

to\_w10:

cmp cl, 10

jb display\_w10

inc al

sub cl, 10

jmp to\_w10

…

cmp is\_on\_heatting\_state, 1; 在加热时

je on\_heatting

and al, 01111111b

jmp out\_w1

on\_heatting:

or al, 10000000b

out\_w1:

mov dx, PORT\_B\_8255

out dx, al ;显示个位

pop cx

pop bx

pop ax

ret

display endp

key\_identify proc near

;(1)判断是否有键按下

mov dx, PORT\_C\_8255

mov al,0

out dx, al; PC3~0行输出全0

nop

in al, dx ;读入列值PC7~4

and al, 11110000b

cmp al, 11110000b

jne re\_confident

mov is\_new\_key, 0; 无新键值

jmp error\_exit

;(2)有键按下，软延时，再次判断

re\_confident:

delay 0ffh

in al, dx ;读入列值PC7~4

and al, 11110000b

cmp al, 11110000b

jne has\_key

mov is\_new\_key, 0; 无新键值

jmp error\_exit

;(3)识别按键。确实有键按下，开始扫描，

has\_key:

mov ah, 11111110b

mov cx,4

scan\_next\_row:

mov al, ah

out dx, al

nop

in al, dx

and al, 11110000b

cmp al, 11110000b

jne find\_key ; 行值ah, 列值al

rol ah, 1

loop scan\_next\_row

jmp error\_exit

find\_key:

and ah, 00001111b

mov cl,4

shr al, cl

; 计算键位值，行值ah, 列值al

;计算行计数值

…

mov ah, bl ; 保存行计数值ah

;计算列计数值

mov bl, -1 ;计数0位置

mov cx, 4

next\_column:

inc bl; 列号计数

shr al, 1

jnc find\_column

loop next\_column

mov is\_new\_key, 0; 无新键值

jmp error\_exit

find\_column:

mov al, bl;保存列计数值al

shl ah, 1

shl ah, 1; x4

add al, ah ;al键位置值

mov cur\_key, al; 保存当前键值

mov is\_new\_key, 1; \*\*\*有新键值

call beep\_200ms;发声

;(4)判断是否键释放，行输出全0

no\_release\_wait:

mov al, 0

out dx, al ; PC7~4列输入，PC3~0 行输出

nop

in al, dx; 读入行值

and al, 11110000b

cmp al, 11110000b

jne no\_release\_wait

delay 0ffh ;键释放，软延时

error\_exit:

ret

key\_identify endp

main:

;----(0)初始化

call init\_vct\_table ;中断时用

call init\_8255 ;8255A初始化;!!!初始化8255时，端口数据消失

call init\_8253 ;中断时不用

call init\_8259 ;仿真时不用

call display

; --->转到两个状态

main\_loop:

cmp is\_in\_control\_state, 1

…

;----(1)设置状态---------------

loop\_in\_setting\_state:

call key\_identify;识别按键

cmp is\_new\_key, 1

jne loop\_in\_setting\_state

cmp cur\_key, 0Bh; 0Bh,进入控制状态

je change2control\_state

…

jmp main\_loop

;----(2)控制状态-----------

loop\_in\_control\_state:

call key\_identify

cmp cur\_key, 0AH; 0AH,设置按键

je change2setting\_state; 转到设置状态

…

call display

jmp main\_loop

ret

1. 选择使用如下元器件，设计一个多路传感器电压采集显示系统。

设计要求：

（1）由两位共阳极七段数码管组成静态显示器，由八个按键组成线性键盘，8通道的ADC连接8个传感器（由可变电阻代替），一路DAC的输出连接一个电压表， 并由8259和8253组成定时中断请求和按键提示音。

（2）每20毫秒由8253产生一次中断请求。设8253时钟CLK等于1MHZ，8253的OUT0连接到8259的IR2端，已知写入8259的ICW2是08H，每次中断时，中断服务程序对ADC的8个通道传感器电压值采集一遍，中断服务程序函数名称为isr\_20ms\_adc。

（3）当有按键按下时，发出1kHz提示声音，两位七段数码管显示与按键对应的ADC通道电压值（保留小数点后1位，单位为V），K0键对应第一个ADC通道，...,K7对应第8个ADC通道，并用DAC0832输出这个通道传感器电压的值，用电压表可以测量。

（4）当无按键按下时， 不发提示音，用两位七段数码管显示八个传感器平均电压值（电压范围为0.0V~5.0V，保留小数点后1位数，单位为V），并用DAC0832输出八个传感器电压的平均值，用电压表可以测量。

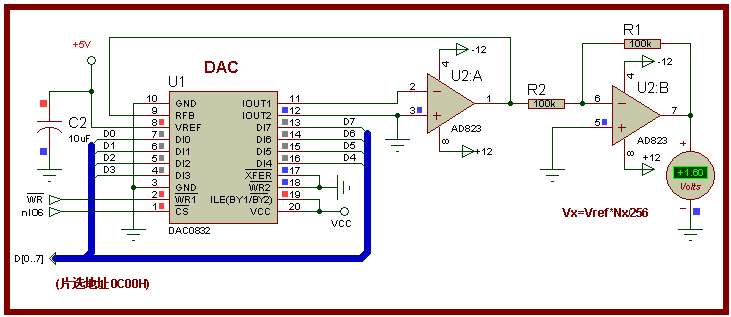
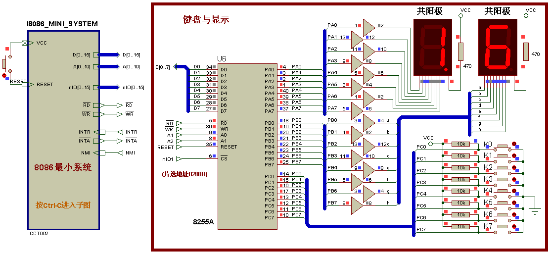
试完成如下设计：

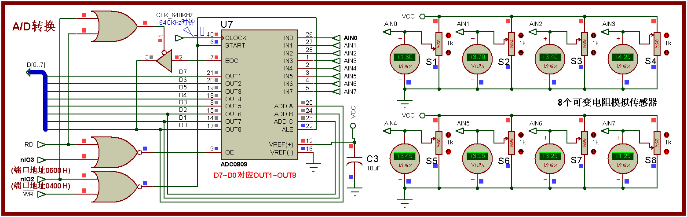
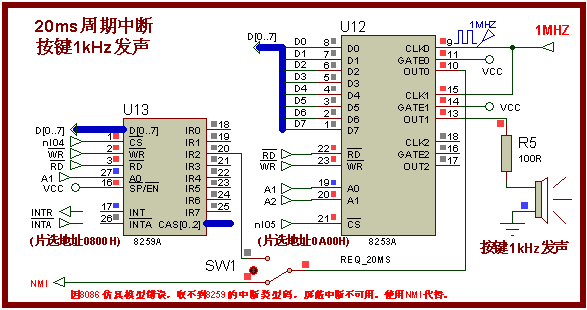
1. 画出与8086连接的硬件原理图。

2. 编写8255、8253和中断向量表的初始化程序；写出七段数码管段码译码表；并编写多路传感器电压采集显示系统的程序。

答：

（1）





（2）

; 由MASM32编译, 并开始使用.IF .REPEAT .WHILE .UNTIL 等宏, 可以不用低层的分支结构，

; 结构化更好。

; Author: hyp@jlu.edu.cn , QQ:hyper/790516

;----0.主程序-----------------------------------------------------------------

main proc far

; (1)初始化

call init\_vct\_table ;中断向量初始化

call init\_8255 ;8255A初始化

call init\_8253 ;8253初始化

call init\_8259 ; 8259初始化

enable\_isr; 开中断

; (2)循环

.WHILE 1

call key\_identify ;判断和识别按键

.IF key\_is\_pressed == 0 ;无键按下时

call display\_AV\_Vx ;显示8个通道平均电压

.ELSE ;有键按下时

call display\_CH\_Vx ;显示当前按键对应Kn通道电压

.ENDIF

.ENDW

ret

main endp

;1.中断向量表初始化

init\_vct\_table proc near

;mov bx, 4\*12h ; n = 12H, IR2有中断请求输入, ICW2=10H

mov bx, 4\*2 ; n = 2, NMI非屏蔽中断

mov ax, 0

mov es, ax

mov ax, offset isr\_20ms\_adc

mov es:[bx], ax

mov ax, seg isr\_20ms\_adc

mov es:[bx+2], ax

ret

init\_vct\_table endp

;-----------------------------------------------------------------------------

;2.8255初始化

init\_8255 proc near

out\_port PORT\_CTR\_8255, 10001001b ;初始化8255控制字

ret

init\_8255 endp

;-----------------------------------------------------------------------------

;3.8253\_计数器0初始化---20ms中断请求

init\_8253 proc near

out\_port PORT\_COUNTER\_CTR, 00110100b ;初始化8253控制字,计数0，r/w低8位 ,高8位，方式2，二进制计数

mov ax, 20000 ;计数常数 ; 20ms/(1/1Mhz) =20000, ; 40ms/(1/1Mhz) =40000

out\_port PORT\_COUNTER0, al ;写低8位

out\_port PORT\_COUNTER0, ah ;写高8位

ret

init\_8253 endp

;-----------------------------------------------------------------------------

;4.8259初始化

init\_8259 proc near

out\_port PORT\_8259\_0, 00010010b ;初始化8259控制ICW1

out\_port PORT\_8259\_1, 08H; ICW2

out\_port PORT\_8259\_1, 01H; ICW4

ret

init\_8259 endp

;-----------------------------------------------------------------------------

;5.发声xxms----

beep\_start proc near

push ax

out\_port PORT\_COUNTER\_CTR, 01110110b ; 开始发声：初始化8253控制字,计数1，r/w低8位 ,高8位，方式3，二进制计数

mov ax, 1000 ; 计数常数 ; 1mhz/1khz =1000

out\_port PORT\_COUNTER1, al ; 写低8位

out\_port PORT\_COUNTER1, ah ; 写8位

pop ax

ret

beep\_start endp

beep\_stop proc near

push ax

out\_port PORT\_COUNTER\_CTR, 01110000b ; 停止发声：初始化8253控制字，计数1,方式0，

pop ax

ret

beep\_stop endp

;6.中断服务程序--- 双状态，非阻塞方式

; 进两次采集一次AD值，40ms采集一次, 循环采集8个通道AD值,并计算8个Vx\_x10电压值

isr\_20ms\_adc proc far

.IF ( I\_Flag == 1)

push ax

push bx

push cx

push dx

push si

pushf

.IF is\_adc\_started == 0 ; 没有启动，则启动

out\_port PORT\_ADC\_START, cur\_sensor\_channel ;启动当前通道

mov is\_adc\_started, 1; 标志已经启动

.ELSE

; 查询转换结束引脚eoc，非阻塞

in\_port PORT\_ADC\_EOC ; 输入值在al中

test al, 00000001b ;eoc引脚为高电平？

.IF !zero? ; AD转换结束？

in\_port PORT\_ADC\_DATA ; 输入值在al中

mov si, offset sensor\_Nx

mov bh, 0

mov bl, cur\_sensor\_channel

add si, bx

mov [si], al ; 保存当前通道ADC数字量

;计算Vx\_x100电压，并保存。sensor\_Vx\_x100 = 100\*5\*Nx/256

mov si, offset sensor\_Vx\_x100

shl bx, 1 ; x2

add si, bx

mov ah, 0

mov cx, 500

mul cx ; dxax = ax \*cx

mov cx, 256

div cx ; ax = dxax/cx

mov [si], ax

; 下次采集下一个通道

inc cur\_sensor\_channel

.IF cur\_sensor\_channel == 8

mov cur\_sensor\_channel, 0

.ENDIF

mov is\_adc\_started, 0 ; 下次是启动AD状态

.ENDIF

.ENDIF

popf

pop si

pop dx

pop cx

pop bx

pop ax

.ENDIF

iret

isr\_20ms\_adc endp

;-----------------------------------------------------------------------------

;7.识别键盘---8个线性按键，非阻塞。出口：保存按键位置值到当前按键变量cur\_key

key\_identify proc near

in\_port PORT\_C\_8255 ; ;读入列值PC7~0

;(1)判断是否有键按下

.IF al == 11111111b

mov key\_is\_pressed, 0 ;无按键

call beep\_stop ;停止发声

.ELSE

;(2)有键按下，软延时，再次判断

;delay 0ffh

in\_port PORT\_C\_8255 ;读入列值PC7~0

.IF al == 11111111b ;再次判断

mov key\_is\_pressed, 0 ;无按键,是干扰

call beep\_stop ;停止发声

;(3)确实有键按下，识别具体按键。

.ELSE

mov ch, 8 ;共8个按键

mov cl, 0 ;按键序号

.WHILE ch != 0

shr al, 1

.IF carry? ;C=1?

inc cl

.ELSE

.BREAK ;C=0?

.ENDIF

dec ch

.ENDW

mov cur\_key, cl ;保存键序号值到当前按键变量

mov key\_is\_pressed, 1 ;有按键

call beep\_start ;发声

.ENDIF

.ENDIF

ret

key\_identify endp

;-----------------------------------------------------------------------------

;8.Kn按键按下时，显示ADC n通道的电压值，并输出这个电压值到DAC

display\_CH\_Vx proc near

mov si, offset sensor\_Vx\_x100

mov bh, 0

mov bl, cur\_key

shl bx, 1 ; x2

add si, bx

disable\_isr ;关中断

mov ax, [si] ;取某通道的sensor\_Vx\_x100

enable\_isr ;开中断

;计算百位和十位数电压数值

mov bx, offset led\_table

mov cl, 100

div cl ; ahal = ax/cl

xlat ;al <-- [bx + al]

and al,01111111b ;加小数点

out\_port PORT\_A\_8255, al ;显示带小数点的百位数

mov al, ah

mov ah, 0

mov cl, 10

div cl ; ahal = ax/cl

xlat ;al <-- [bx + al]

out\_port PORT\_B\_8255, al ;显示十位数

; DAC输出当前通道的电压

mov si, offset sensor\_Nx

mov bh, 0

mov bl, cur\_key

add si, bx

disable\_isr ;关中断

mov al, [si] ;取当前通道Nx值

enable\_isr ;开中断

out\_port PORT\_DAC\_DATA, al

ret

display\_CH\_Vx endp

;-----------------------------------------------------------------------------

;9.显示8个通道平均电压

display\_AV\_Vx proc near

; 计算8个通道的平均Nx值

mov bx, offset sensor\_Nx

mov cl, 8

mov ax, 0

mov dx, 0

clc

.WHILE (cl != 0)

disable\_isr ;关中断

mov dl, [bx]

enable\_isr ;开中断

adc ax, dx

inc bx

dec cl

.ENDW

mov cl, 8

div cl; al = ax/cl

mov sensor\_Nx\_AV, al

out\_port PORT\_DAC\_DATA, al; DAC输出平均电压

; Vx\_x100 = 500\*Nx/256

mov ah, 0

mov cx, 500

mul cx ; dxax = ax\*cx

mov cx, 256

div cx ; ax = dxax/cx, sensor\_Vx\_AV\_x100

; 计算百位和十位数电压数值

mov cl, 100

div cl ; ah(余)al(商) = ax/cl

mov bx, offset led\_table

xlat ;al <-- [bx + al]

and al,01111111b;加小数点

out\_port PORT\_A\_8255, al ;显示百位

mov al, ah ; ah(余)

mov ah, 0

mov cl, 10

div cl ; ahal=ax/cl

xlat ;al <-- [bx + al]

out\_port PORT\_B\_8255, al ;显示十位数

ret

display\_AV\_Vx endp

;-----------------------------------------------------------------------------

end

