

Control Puerto Paralelo : Yoel Ocmin

;Control del Puerto Paralelo 378H  
;Hecho Por Yoel Ocmin 24/10/2003  
;Cualquier duda consultar en la Pagina Web  
;o al Correo yoelocmin@hotmail.com

.286  
.model small  
.stack 64h  
;Para poder Programar en Assembler con mayor Facilidad  
include macros.inc; Incluyo mi libreria de Macros

.data  
;EL Titulo que aparece en Pantalla, que los uso como plantilla de Fondo  
TITULO DB 10,13  
DB " Manejo del Puerto Paralelo",10,13  
db " =====",10,13,10,13  
db " Datos : D0 D1 D2 D3 D4 D5 D6 D7",10,13  
db " [] [] [] [] [] [] [] []",2 Dup(10,13)  
DB "  
db " Control : C0 C1 C2 C3 Estado : S3 S4 S5 S6 S7",10,13  
db " [] [] [] [] [] [] [] []\$"  
;Datos del Menu de Cambio  
Dat1 db " Para Realizar cambios pulse :",10,13,10,13  
db " Numeros 0 ",126," 7 Para cambiar Bits de Datos",10,13  
db " [F1][F2][F3][F4]=> Para Cambiar Bits de Control",10,13  
db " [F5] => Para Actualizar",10,13  
db " [Enter] => Efectuar Cambios",10,13  
db " [Esc] => Para Salir ",10,13,'\$'

;Aqui guardo mis datos del Registro de Datos luego de leer el Puerto Base=378H  
;Puedo Cambiar  
datos label byte  
D0 DB '0'  
D1 DB '0'  
D2 DB '0'  
D3 DB '0'  
D4 DB '0'  
D5 DB '0'  
D6 DB '0'  
D7 DB '0'

;Aqui Guardo mis datos del registro de Control (Base +2) 378h+2=37Ah  
;Puedo Cambiar  
ctrl label byte  
C0 DB '0'  
C1 DB '0'  
C2 DB '0'  
C3 DB '0'

;Aqui guardo mis datos del registro de Estado (Base + 1) 378h+1 =379h  
;No puedo cambiar  
estado label byte  
E3 DB '0'  
E4 DB '0'  
E5 DB '0'  
E6 DB '0'  
E7 DB '0'

;Archivo con las coordenadas dentro de la pantalla donde voy a situar mis datos  
include lugares.inc

.code  
MOV AX,@DATA  
MOV DS,AX  
CLS ;Limpio la pantalla  
CALL INGRES\_INI ;Inicializo los valores, es decir leo mis datos  
;y los coloco en DATOS, CTRL y Estado  
cadena titulo ;Macro Poner titulo en Pantalla

;Aqui es donde se pone con cada coordenada, el respectivo dato

<http://protectos-fie.tk>

```

include actdatos.inc
;Luego examino la ultima tecla presionada,
;con el servicio 0 de la int16
;Recordemos que nos da como resultado
;AH=opcode tecla y AL=codigo ASCII
;y luego ejecuto la rutina revisando que se encuentra abajo
salir: CLS
terminar

;*****
;***** MIS - RUTINAS *****
;*****

;*****
;*** RUTINA DE ACTUALIZACION DE DATOS ***
;*****
;Esta Rutina Lee los estados del Puerto Paralelo 378H
;En sus direcciones de Datos, Control y Estado
;Y en cada una guarda el contenido convertido en ASCII
;en el vector VARIABLE[ x] que Creamos en el segmento .DATA
;Si no se acuerdan tiene la forma
;Data Label Byte
;D0 db '0' ... D7 db '0'
;Donde para guardar en el Dato D6 aqui se pondria
; mov byte ptr DATA[6], AL ;Suponiendo que el dato se encuentra en AL

INGRES_INI PROC NEAR
PUSHA
MOV BX,1h ;Inicializo mi Puntero dentro del bit
MOV DI,0 ;Inicializo mi Puntero
REPITE_in_DAT:MOV DX,378H
IN AL,DX
AND AL,BL ;Me quedo con el valor del bit que quiero
MOV CX,DI ;Muevo el numero de bits a regresar
ROR AL,CL ;Roto para quedarme solo con el bit que quiero
ADD AL,30H ;Convierto al bit en Ascii
MOV DATOS[di],AL ;Lo guardo en Datos
ROL BL,1 ;Roto para incrementar BI y pasar a otro bit
INC DI ;Incremento mi puntero
cmp DI,8 ;Comparo si ya llegue al bit 8vo
jnz REPITE_in_DAT

MOV BX,00001000b;Lo inicializo aqui, ya que comienzo
;a revizar a partir del bit S3
MOV DI,0 ;Mi Puntero de datos
REPITE_in_ESTADO:MOV DX,379H
IN AL,DX
AND AL,BL
MOV CX,DI
ADD CX,3
ROR AL,CL
ADD AL,30H
MOV ESTADO[di],AL
ROL BL,1
INC DI
cmp DI,5
jnz REPITE_in_ESTADO

MOV BX,00000001b;Lo inicializo aqui, ya que comienzo
;a revizar a partir del bit C1
MOV DI,0 ;Mi Puntero de datos
REPITE_in_Ctrl:MOV DX,37AH
IN AL,DX
AND AL,BL
MOV CX,DI
ROR AL,CL
ADD AL,30H
MOV CTRL[di],AL
ROL BL,1
INC DI

```

## Control Puerto Paralelo : Yoel Ocmin

```

cmp DI,4
jnz REPITE_in_CTRL
popa
RET
INGRES_INI ENDP

```

```

.*****
;
;**** RUTINA DE REVISION DE TECLA ****
;
.*****
;

```

```

;Revisa que tecla se ha pulsado y de acuerdo a eso
;Cambia los valores de nuestras variables:
;Data y CTRL ; a Estado no lo toca ya que ese no se
;Puede cambiar, es de solo Lectua Puerto In
;Mientras que DATA Puede ser IN-OUT , de acuerdo como se configure
;CTRL es IN-OUT

```

```
revisando proc near
```

```

cmp ah,01 ;Si es Escape ;[ESC] opcode 01
jz retorna_revi_salir ;Salta a salir

```

```

cmp ah,3fh ;Si es Actualizar ;[F5] opcode 3Fh
jz refrescar ;Salta a refrescar
cmp al,0Dh ;Si es ENTER ;[ENTER] opcode 0Dh
jz ES_ENTER ;Salta a Es_enter

```

```

cmp al,30h ;Para ver si esta fuera del rango de los numero
;Ademas los [Fx] no tienen código ASCII
; y si se presiono aparece 0, y es menor que 30H
jl s_reb;Si son F1,F2,F3,F4;Es decir si alteran algo dentro de control
cmp al,37h;Abarcamos todos los numeros ya que los que no son menores de 30
; y son menores a 37h son los numeros que estan en el rango [30h -37h]
;Solo de '0' a '7', ya que nuestros datos son [D0 ... D7]

```

```

jl s_numeros;saltar a son numeros
ret ;Si no es ninguno, entonces retornar
retorna_revi_salir: mov ax,0
ret

```

```
refrescar: call INGRES_INI ;F5
```

```
ret
```

```
es_ENTER: call actualizando
```

```
ret
```

```
s_numeros:
```

```

mov bl,al
sub bl,30h
mov bh,0
mov di,bx
mov al,byte ptr datos[di]
xor al,1 ;Cambiamos ultimo bit de Datos[i]
mov byte ptr datos[di],al
ret

```

```
s_reb:
```

```

mov al,ah
mov ah,0
mov bx,3Bh
sub ax,bx
mov di,ax
mov al,byte ptr ctrl[di]
xor al,1;Cambiamos ultimo bit de CTRL[i]
mov byte ptr ctrl[di],al
ret

```

```
revisando endp
```

```
actualizando proc near
```

```
MOV DI,0;Inicializo mi puntero
```

```
//Hago que mis datos,control vuelvan a binario
```

```
mov CX,0
```

```
ACT_1:MOV AL,BYTE PTR Datos[di]
```

```
AND AL,1h
```

```
MOV BYTE PTR DATOS[di],AL
```

```
INC DI
```

Control Puerto Paralelo : Yoel Ocmin

```
inc cx
cmp cx,12 ;hago 12 veces ya que se encuentran juntos DATOS y CTRL
           ;y coincide DATOS[8] con CTRL[0]
jnz act_1

MOV DI,0;Inicializo mi Puntero
MOV BL,0;Dato Actual
MOV CX,0;Contador de bits
ACT_2: MOV AL,byte ptr Datos[Di]
        ADD BL,AL
        ROR BL,1
        inc di
        inc cx
        cmp cx,8
        jnz act_2

mov dx,378h
mov al,bl
out dx,al

MOV DI,0;Inicializo mi Puntero
MOV BL,0
MOV CX,0
ACT_3: MOV AL,byte ptr CTRL[Di]
        ADD BL,AL
        ROR BL,1
        inc di
        inc cx
        cmp cx,4
        jnz act_3

ROR BI,4;Hago que rote 4 veces para obtener un rotamiento total
           ;de Bits desde que empezo y regrese con
           ;BL=(0)(0)(0)(0)(~C3)(C2)(~C1)(~C0)
           ;~ :significa negado

mov dx,37ah
mov al,bl
out dx,al
call INGRES_INI
ret
actualizando endp
end
```