

Contexto teórico

Periféricos internos

PIO

- Interfaz de E/S con dos puertos paralelo de un byte (A y B)
- Sus bits son programables como *entrada (1)* o *salida (0)*
- Tiene 4 registros internos tipo byte a partir de la *dirección 30H*
 - **PA y PB:** almacenan el dato presente en cada puerto
 - **CA y CB:** determinan el comportamiento en cada puerto.

HAND

- Interfaz de periferia que cumple la temporización especificada en el interfaz Centronics, *no admitiendo* modo de programación alguno. Se comunica con la CPU por consulta de estado o a través de int, en función del BIT 7 de su registro de estado.
- Tiene 2 registros internos tipo byte a partir de la *dirección 40H*
 - **DATO:** hacer OUT permite sacar algún bit; IN devuelve el último sacado.
 - **EST:** estado. Bit 0: BUSY (R). Bit 1: STROBE (R). Bit 7: INT (W/R)

Otros

- PIC: Controlador de interrupciones.
- TIMER: Contador de eventos.
- CDMA: Controlador de Acceso directo a memoria

Periféricos externos

- Pantalla y teclado
- Unidades de disco
- Barra de LEDs
- Barra de Microconmutadores (micro switches)
- Impresora

Ejemplo 01 - Configuración P1 C0

Teclas 0-7 permiten cambiar el estado de cada micro switch

PA	EQU 30H	; Dirección del contenido del Puerto A (PIO)
PB	EQU 31H	; Dirección del contenido del Puerto B (PIO)
CA	EQU 32H	; Dirección de control para Puerto A (PIO)
CB	EQU 33H	; Dirección de control para Puerto B (PIO)
	ORG 2000H	
	MOV AL, 0FFH	; PA entradas (Microconmutadores)
	OUT CA, AL	; Ahora todos los bits de Puerto A son entradas
	MOV AL, 0	; PB salidas (Luces)
	OUT CB, AL	; Ahora todos los bits de Puerto B son salidas
POLL:	IN AL, PA	; Recibo bits de PA (estado de micro switches)
	OUT PB, AL	; Modifico bits de PB (enciendo o apago luces)
	JMP POLL	
	END	

Ejemplo 02 - Configuración P1 C0

PIC EQU 20H ; Dirección del controlador de interrupciones
 TIMER EQU 10H ; Dirección del reloj interno
 PIO EQU 30H ; Dirección de la interfaz externa
 N_CLK EQU 10 ; Número de interrupción del reloj en vector

ORG 40

IP_CLK DW RCLK ; Redirección a la rutina de manejo de int

ORG 1000H

INICIO DB 0 ; Dato en memoria. Podría ir a registro

ORG 2000H

CLI ; Ahora no puedo atender interrupciones
 MOV AL, 0FDH ; 1111 1101: sólo escucharé al reloj
 OUT PIC+1, AL ; Escritura en el registro IMR (21H)
 MOV AL, N_CLK
 OUT PIC+5, AL ; Escritura en el registro INT1 (25H)
 MOV AL, 1 ; El reloj interrumpirá cada 1 segundo
 OUT TIMER+1, AL ; Escritura en el registro COMP (11H)
 MOV AL, 0 ; Dato en común
 OUT PIO+3, AL ; Registro CB (33H): todo salida
 OUT PIO+1, AL ; Registro PB (31H): luces apagadas
 OUT TIMER, AL ; Registro CONT (10H): reloj en cero
 STI ; Listo para escuchar interrupciones
 LAZO: JMP LAZO

ORG 3000H

RCLK: INC INICIO ; En este caso, no es necesario salvar AX
 CMP INICIO, 0FFH ; Incrementar variable en memoria
 JNZ LUCES ; Variable en memoria ya es 255?
 MOV INICIO, 0 ; Caso afirmativo: reiniciar en 0
 LUCES: MOV AL, INICIO
 OUT PIO+1, AL ; Registro PB (31H): actualizar luces
 MOV AL, 0
 OUT TIMER, AL ; Registro CONT (10H): reiniciar reloj
 MOV AL, 20H
 OUT PIC, AL ; Registro EOI (20H): ya terminé!
 IRET ; Volver al funcionamiento normal
 END

Ejemplo 04 - Configuración P1 C1

```

PIO    EQU 30H                ; Dirección de la interfaz externa

      ORG 1000H
MSJ    DB "CONCEPTOS DE "
      DB "ARQUITECTURA DE "
      DB "COMPUTADORAS"
FIN    DB ?

      ORG 2000H
MOV AL, 0FDH                ; INICIALIZACION PIO PARA IMPRESORA
OUT PIO+2, AL                ; Puerto A (32H): todo entrada, menos bit 1 (salida)
MOV AL, 0
OUT PIO+3, AL                ; Puerto B (33H): todo salida
IN AL, PIO                    ; Recibo bits de Puerto A (30H)
AND AL, 0FDH                ; Máscara: ==== ==0= (STROBE = 0)
OUT PIO, AL                  ; FIN INICIALIZACION
MOV BX, OFFSET MSJ           ; Inicio de Mensaje
MOV CL, OFFSET FIN-OFFSET MSJ ; Longitud de Mensaje
POLL: IN AL, PIO              ; Recibo bits de Puerto A (30H)
      AND AL, 1                ; Extraigo el bit 0 (BUSY)
      JNZ POLL                ; Esperar mientras siga ocupada
      MOV AL, [BX]             ; Traigo carácter de memoria
      OUT PIO+1, AL            ; Lo mando al Puerto B (31H)
      IN AL, PIO               ; PULSO 'STROBE'
      OR AL, 02H               ; Máscara: ==== ==1= (STROBE = 1)
      OUT PIO, AL              ; Lo mando al Puerto A (30H)
      IN AL, PIO               ; Recibo bits de Puerto A (30H)
      AND AL, 0FDH            ; Máscara: ==== ==0= (STROBE = 0)
      OUT PIO, AL              ; FIN PULSO
      INC BX                   ; Siguiendo dirección
      DEC CL                   ; Caracteres restantes
      JNZ POLL                 ; Repetir hasta CL = 0
      INT 0                    ; Detener la ejecución

      END

```

Ejemplo 05 - Configuración P1 C1

```

PIO    EQU 30H                                ; Dirección de la interfaz externa

      ORG 1000H
NUM_CAR DB 5                                ; Dato en memoria, irá a un registro
CAR     DB ?                                ; Obligatorio en memoria por INT 6

      ORG 3000H ; SUBROUTINA DE INICIALIZACION PIO PARA IMPRESORA
INI_IMP: MOV AL, 0FDH
          OUT PIO+2, AL                      ; CA = todo entrada, excepto bit = 1 (salida)
          MOV AL, 0
          OUT PIO+3, AL                      ; CB = todo salida
          IN AL, PIO
          AND AL, 0FDH
          OUT PIO, AL                        ; PA: STROBE = 0
          RET

      ORG 4000H ; SUBROUTINA DE GENERACIÓN DE PULSO 'STROBE'
PULSO: IN AL, PIO
          OR AL, 02H
          OUT PIO, AL                        ; PA: STROBE = 1
          IN AL, PIO
          AND AL, 0FDH
          OUT PIO, AL                        ; PA: STROBE = 0
          RET

      ORG 2000H ; PROGRAMA PRINCIPAL
PUSH AX                                ; Salvamos AX
CALL INI_IMP                           ; Rutina 3000H
POP AX                                 ; Recuperamos AX
MOV BX, OFFSET CAR                      ; Dirección de carácter
MOV CL, NUM_CAR                          ; Cantidad de lecturas
LAZO: INT 6                              ; Solicitar ingreso
POLL: IN AL, PIO
          AND AL, 1                          ; Extraigo bit BUSY
          JNZ POLL                           ; Esperar mientras esté ocupada
          MOV AL, [BX]                       ; Traigo carácter de memoria
          OUT PIO+1, AL                      ; Lo mando al Puerto B (31H)
          PUSH AX                             ; Salvamos AX
          CALL PULSO                          ; Rutina 4000H
          POP AX                             ; Recuperamos AX
          DEC CL                              ; Lecturas restantes
          JNZ LAZO                           ; Repetir hasta CL = 0
          INT 0                               ; Detener la ejecución

      END

```

Ejemplo 07 - Configuración P1 C2

```

HAND EQU 40H                                ; Dirección del periférico externo

ORG 1000H
MSJ DB "INGENIERIA E "
    DB "INFORMATICA"
FIN DB ?

ORG 2000H
IN AL, HAND+1                                ; Recibo bits de EST (41H)
AND AL, 7FH
OUT HAND+1, AL                               ; EST (41H): 0=== ==== (INT = 0)
MOV BX, OFFSET MSJ                           ; Inicio de cadena
MOV CL, OFFSET FIN-OFFSET MSJ                ; Longitud de cadena
POLL: IN AL, HAND+1
      AND AL, 1
      JNZ POLL                               ; Chequeamos bit 0 (BUSY)
      MOV AL, [BX]                           ; Traigo carácter de memoria
      OUT HAND, AL                           ; Lo mando a DATO (40H)
      INC BX                                 ; Siguiendo dirección
      DEC CL                                 ; Caracteres restantes
      JNZ POLL                               ; Repetir hasta CL = 0
      INT 0

END

```

Ejemplo 08 - Configuración P1 C2

```

PIC    EQU 20H                ; Dirección del controlador de int
HAND  EQU 40H                ; Dirección del periférico externo
N_HND EQU 10

    ORG 40
IP_HND DW RHND                ; Redirección a rutina de manejo int

    ORG 1000H
MSJ    DB "UNIVERSIDAD "
        DB "NACIONAL DE LA PLATA"
FIN    DB ?

    ORG 3000H                ; Precondición: BX = OFFSET MSJ
RHND:  PUSH AX                ; Salvamos AX
        MOV AL, [BX]          ; Traigo carácter de memoria
        OUT HAND, AL          ; Lo mando a DATO (40H)
        INC BX                ; Siguiente dirección
        DEC CL                ; Caracteres restantes
        MOV AL, 20H           ; EOI (20H): ya terminé!
        OUT PIC, AL           ; Recuperamos AX
        POP AX                ; Vuelta al funcionamiento normal
        IRET

    ORG 2000H
MOV BX, OFFSET MSJ            ; Inicio de cadena
MOV CL, OFFSET FIN-OFFSET MSJ ; Longitud de cadena
CLI                            ; Ahora no me interrumpen
MOV AL, 0FBH
OUT PIC+1, AL                 ; IMR (21H): 1111 1011 (bit 2 = 0)
MOV AL, N_HND                 ; Número de int en vector
OUT PIC+6, AL                 ; INT2 (26H): escribo el número
MOV AL, 80H
OUT HAND+1, AL                ; EST (41H): 1000 0000 (INT = 1)
STI                            ; Listo para escuchar int
LAZO:  CMP CL, 0
        JNZ LAZO              ; Si no hay cadena, no hacer nada
        IN AL, HAND+1          ; Recibo bits de EST (41H)
        AND AL, 7FH
        OUT HAND+1, AL        ; EST (41H): 0=== ==== (INT = 0)
        INT 0

    END

```