

# PRACTICA 3- SOLUCIONES

## Entrada/Salida

### Ejercicio 1a

```
ORG 1000H    ; Memoria de datos
patron db 0C3h    ;1100 0011b
```

```
CB EQU 33h
PB EQU 31h
```

```
ORG 2000H    ;Prog principal
mov al, 0
out CB, al
mov al, patron
out PB, al
HLT
END
```

### Ejercicio 1b

```
ORG 1000H    ; Memoria de datos
prendida db "Llave prendida"
apagada db "Llave apagada"
fin_apagada db ?
```

```
CA EQU 32h
PA EQU 30h
```

```
ORG 2000H    ; Prog principal
mov al, 0ffh
out CA, al

    in al, PA
; poner en 0 todos los bits menos el más sig
    and al, 80h ; 1000 0000
; si es 0
    cmp al, 0
    jz esta_apagada
; esta prendida
    mov bx, offset prendida
    mov al, OFFSET apagada - OFFSET prendida
    jmp fin
esta_apagada: mov bx, offset apagada
    mov al, OFFSET fin_apagada - OFFSET apagada

fin: int 7 ; imprimir
HLT
END
```

### Ejercicio 1c

```
PA EQU 30H
PB EQU 31H
CA EQU 32H
CB EQU 33H
```

```
ORG 2000H
MOV AL, OFFH    ; PA entradas (Micro-conmutadores)
OUT CA, AL
MOV AL, 0        ; PB salidas (Luces)
```

```

        OUT CB, AL
POLL:   IN  AL, PA
        OUT PB, AL
        JMP POLL
        END

```

### Ejercicio 1d

```

PIC      EQU 20H
TIMER    EQU 10H
PIO      EQU 30H
N_CLK    EQU 10

```

```

        ORG 40
IP_CLK   DW  RUT_CLK

```

```

        ORG 1000H
PATRON   DB  0
FINAL    DB  0

```

```

        ORG 2000H
        CLI
        MOV AL, 0FDH
        OUT PIC+1, AL

        MOV AL, N_CLK

        OUT PIC+5, AL
        MOV AL, 1
        OUT TIMER+1, AL
        MOV AL, 0
        OUT PIO+3, AL
        OUT PIO+1, AL
        OUT TIMER, AL
        STI
LAZO:    CMP FINAL, 1
        JNZ LAZO
        HLT

```

```

        ORG 3000H
RUT_CLK: INC PATRON
        CMP PATRON, 0FFh
        JNZ LUCES
        MOV FINAL, 1
        MOV AL, 0FFh
        OUT PIC+1, AL
        JMP FIN
LUCES:   MOV AL, PATRON
        OUT PIO+1, AL
        MOV AL, 0
        OUT TIMER, AL
FIN:     MOV AL, 20H
        OUT PIC, AL
        IRET
        END

```

### Ejercicio 2a

; Ejecutar en configuración 1

```

        ORG 1000H; Memoria de datos
char    db "A"

```

```

PA      EQU 30h
PB      EQU 31h
CA      EQU 32h
CB      EQU 33h

```

```

        ORG 2000H    ; Prog principal
        mov al, 01h ; strobe salida (0), busy entrada (1)
        out CA, al
        mov al, 0    ; puerto de datos todo salida
        out CB, al

```

```

; inicializo strobe en 0
        in  al, PA
        and al, 11111101b
        out PA, al

```

```

; espero que busy=0
poll:   in  al, PB
        and al, 01h ; 1000 0000
        jnz poll

```

```

; se que busy es 0, mandar caracer
    mov al, char
    out PB, al
; mandar flanco ascendente de strobe
    in al, PA
    or al, 00000010b
    out PA, al

    nop    ; esperamos un poco que imprima
    nop    ; esperamos un poco que imprima
    nop    ; esperamos un poco que imprima
    nop    ; esperamos un poco que imprima
    nop    ; esperamos un poco que imprima
    nop    ; esperamos un poco que imprima

    HLT
    END

```

### Ejercicio 2b

```

PIO    EQU    30H

        ORG 1000H

MSJ     DB    "ORGANIZACIÓN Y      "
        DB    "ARQUITECTURA DE   "
        DB    "COMPUTADORAS"
FIN     DB    ?

        ORG 2000H
; INICIALIZACION PIO PARA IMPRESORA
; CA
        MOV AL, 0FDH
        OUT PIO+2, AL
; CB
        MOV AL, 0
        OUT PIO+3, AL
; Strobe
        IN  AL, PIO
        AND AL, 0FDH
        OUT PIO, AL
; FIN INICIALIZACION

        MOV BX, OFFSET MSJ
        MOV CL, OFFSET FIN - OFFSET MSJ
POLL:   IN  AL, PIO
        AND AL, 1
        JNZ POLL
; Enviar carácter
        MOV AL, [BX]
        OUT PIO+1, AL
; Pulso STROBE
        IN  AL, PIO
        OR  AL, 02H
        OUT PIO, AL
; Reiniciar STROBE
        IN  AL, PIO
        AND AL, 0FDH
        OUT PIO, AL
        INC BX      ; Mover el puntero de la cadena
        DEC CL
        JNZ POLL    ; Verificar fin de la cadena
        INT 0
        END

```

**Ejercicio 2c**

<pre> PIO      EQU 30H            ORG 1000H NUM_CAR  DB  5 CAR       DB  ?  ; SUBROUTINA DE INICIALIZACION ; PIO PARA IMPRESORA           ORG 3000H INI_IMP: MOV AL, 0FDH           OUT PIO+2, AL           MOV AL, 0           OUT PIO+3, AL           IN  AL, PIO           AND AL, 0FDH           OUT PIO, AL           RET  ; PROGRAMA PRINCIPAL           ORG 2000H           PUSH AX           CALL INI_IMP           POP  AX           MOV BX, OFFSET CAR           MOV CL, NUM_CAR LAZO:     INT  6 POLL:     IN  AL, PIO           AND AL, 1           JNZ POLL           MOV AL, [BX]           OUT PIO+1, AL           PUSH AX           CALL PULSO           POP  AX           DEC  CL           JNZ LAZO           INT  0           END </pre>	<pre> ; SUBROUTINA DE GENERACIÓN ; DE PULSO 'STROBE'           ORG 4000H PULSO:    IN  AL, PIO           OR  AL, 02H           OUT PIO, AL           IN  AL, PIO           AND AL, 0FDH           OUT PIO, AL           RET </pre>
---	--

**Ejercicio 2d**

```

EOI      EQU 20h
IMR      EQU 21h
INT0     EQU 24h

IDINT0   EQU 10

PA       EQU 30h
PB       EQU 31h
CA       EQU 32h
CB       EQU 33h

          ORG 1000H
flag     db 0
longitud db 0
cadena   db ?

          org 40
dir_rut  dw rut_f10

          org 3000h
; cancelar interrupciones futuras
rut_f10: mov al, 0FFH
          out IMR, al
; indicamos al programa que no lea más

```

```

    mov flag,1

    mov al, 20h
    out EOI, al
    iret

    ORG 2000H
    cli
; INICIALIZACION PIO PARA IMPRESORA
    MOV AL, 0FDH
    OUT CA, AL
    MOV AL, 0
    OUT CB, AL
    IN AL, PA
    AND AL, 0FDH
    OUT PA, AL
; Inicialización del PIC
    mov al, 0FEh      ; FE = 1111 1110
    out IMR, al
    mov al, IDINT0
    out INT0, al
    sti

; Lectura de cadena
    MOV BX, OFFSET cadena
loop: int 6           ; leer char
    inc bx
    inc longitud
    cmp flag, 0      ; verifico si presionaron f10
    jz loop

; Impresión de los caracteres leídos
    MOV BX, OFFSET cadena ; reiniciar puntero al comienzo
POLL: nop
    IN AL, PA
    AND AL, 1
    JNZ POLL
; Enviar carácter
    MOV AL, [BX]
    OUT PB, AL
; Pulso STROBE
    IN AL, PA
    OR AL, 02H
    OUT PA, AL
; Reiniciar STROBE
    IN AL, PA
    AND AL, 0FDH
    OUT PA, AL
; pasar al siguiente char
    INC BX
    DEC longitud
    JNZ POLL
    INT 0
    END

```

### Ejercicio 3a

```

HAND    EQU 40H
        ORG 1000H
MSJ     DB "INGENIERIA E      "
        DB "INFORMATICA"
FIN     DB ?

        ORG 2000H
        IN AL, HAND+1

```

```

        AND AL, 7FH
        OUT HAND+1, AL
        MOV BX, OFFSET MSJ
        MOV CL, OFFSET FIN-OFFSET MSJ
POLL:   IN  AL, HAND+1
        AND AL, 1
        JNZ POLL
        MOV AL, [BX]
        OUT HAND, AL
        INC BX
        DEC CL
        JNZ POLL
        INT 0
        END

```

### Ejercicio 3d

PIC	EQU 20H		
HAND	EQU 40H		
N_HND	EQU 10		
IP_HND	ORG 40 DW RUT_HND	MSJ	ORG 1000H DB "UNIVERSIDAD" DB "NACIONAL DE LA PLATA" DB ?
RUT_HND:	ORG 3000H PUSH AX MOV AL, [BX] OUT HAND, AL INC BX DEC CL JNZ FINAL MOV AL, 0FFH OUT PIC+1, AL		ORG 2000H MOV BX, OFFSET MSJ MOV CL, OFFSET FIN-OFFSET MSJ CLI MOV AL, 0FBH OUT PIC+1, AL MOV AL, N_HND OUT PIC+6, AL MOV AL, 80H OUT HAND+1, AL
FINAL:	MOV AL, 20H OUT PIC, AL POP AX IRET	LAZO:	STI CMP CL, 0 JNZ LAZO IN AL, HAND+1 AND AL, 7FH OUT HAND+1, AL INT 0 END

### Ejercicio 4a

```

DIN EQU 60h
DOUT EQU 61h
CTRL EQU 62H

ORG 1000H
char DB "A"

; programa principal
ORG 2000H
; programo la USART
; Bits de CTRL:
; Sync | ER | RTS | DTR | RxEN | TxEN | Vb | Sy/As
; Para comunicación asíncrona (Sy/As = 1)
; Velocidad 6 baudios (VB=0)
; Comunicación por DTR (DTR=1)
; Reiniciando flags de errores (ER =1)
; El resto no importa (x)
MOV AL, 51H ; binario=01010001 o x1x1xx01

```

```

        OUT CTRL, AL
POLL: IN  AL, CTRL
        AND AL, 81H
; verifico que el bit 0 y el 7
; estén ambos en 1
        CMP AL, 81H
        JNZ POLL
        MOV AL, char
        OUT DOUT, AL
        INT 0
        END

```

**Ejercicio 4b**

```

        DIN EQU 60h
        DOUT EQU 61h
        CTRL EQU 62H

        ORG 1000H
cadena DB "USART DTR POLLING"
fin DB ?

; programa principal
        ORG 2000H
        MOV BX, OFFSET cadena
        MOV CX, OFFSET fin - OFFSET tabla
; programo la USART
        MOV AL, 51H           ; binario=01010001
        OUT CTRL, AL

POLL: IN  AL, CTRL
        AND AL, 81H
; verifico que el bit 0 y el 7
; estén ambos en 1
        CMP AL, 81H
        JNZ POLL
; Envío el caracter
        MOV AL, [BX]
        OUT DOUT, AL
        INC BX
        DEC CX
        JNZ POLL
        INT 0
        END

```

**Ejercicio 4c**

```

        USART EQU 60H
        XON EQU 11H
        XOFF EQU 13H

; definición de datos
                ORG 1000H
caracteres DW 0
TABLA DB "XON/XOFF Polling"
FIN DB ?

; PROGRAMA PRINCIPAL
                ORG 2000H
INICIO: MOV BX, OFFSET TABLA ; puntero a Tabla
; programo la USART
        MOV AL, 51H           ; binario= 01010001
        OUT USART+2, AL
TEST: IN AL, USART+2          ; espero a que se
        AND AL, 01H           ; envíe el carácter
        CMP AL, 01H           ; a la impresora.

```

```

JNZ TEST
MOV AL, [BX]
OUT USART+1, AL
INC BX
INC caracteres
CMP caracteres, (OFFSET FIN) - (OFFSET TABLA)
JZ FINAL
IN AL, USART+2          ; Consulto si RxDY
AND AL, 02H             ; se activó. De ser
CMP AL, 02H             ; así, la impresora
JZ RXON                 ; transmite un XON ó
JMP TEST                ; un XOFF al CPU.

; espera recibir XON
RECIBIR: IN AL, USART+2
AND AL, 02H
CMP AL, 02H
JNZ RECIBIR
RXON: IN AL, USART
MOV AH, AL
CMP AL, XON             ; si es XON sigo
JZ TEST                 ; la impresión.
CMP AH, XOFF            ; si es XOFF espero
JZ RECIBIR              ; que libere el buffer
FINAL: INT 0
END

```

## Anexo DMA

El formato del registro control es el siguiente

TC				MT	ST	TT	STOP
----	--	--	--	----	----	----	------

Donde:

TC: Terminal Count

MT: Modo de transferencia

ST: Sentido de transferencia

TT: Tipo de transferencia

STOP: habilitar o detener transferencia

### Ejercicio 2

- Para que el al HAND-SHAKE emita una interrupción, la línea busy del procesador debe estar en 0
- El al HAND-SHAKE utiliza la línea DREC del CMDA para indicarle que debe iniciar la transferencia. Se comunican a través de la línea DREC y la línea DACK
- EL DMAC lee desde memoria un byte, en la dirección especificada en el registro RF (compuesto por RFL y RFH). Luego envía ese byte al HAND-SHAKE cuando este le indica mediante DREQ que puede recibir datos. Finalmente, el HAND-SHAKE envía el carácter a la impresora.
- El DMAC genera una interrupción cuando finaliza de enviar los caracteres a la impresora
- Cuando todos los caracteres han sido enviados a la impresora, detectado mediante la variable FLAG cuyo valor se cambia desde la subrutina que maneja las interrupciones del CMDA (RUT\_DMA)

### Ejercicio 3a

Al ser memoria memoria, el bit TT=1. Al ser por robo de ciclo MT=0. Como queremos que se realice, STOP=0. Entonces el byte de configuración debe ser **XXXX0X10**

El carácter X indica que el valor no importa. El bit ST no importa porque es transferencia memoria memoria.

### Ejercicio 3b

Al ser entre un Periférico y Memoria, el bit TT=0. Al ser Periférico → Memoria, el bit ST=0 Al ser por ráfagas, MT=1. Como queremos que se realice, STOP=0. Entonces el byte de configuración debe ser **XXXX1000**

El carácter X indica que el valor no importa.

### Ejercicio 3c

Al ser entre un Periférico y Memoria, el bit TT=0. Al ser Memoria → Periférico, el bit ST=1 Al ser por robo de ciclo, MT=0. Como queremos que se realice, STOP=0. Entonces el byte de configuración debe ser **XXXX0100**

El carácter X indica que el valor no importa.