



UNIVERSIDAD
NACIONAL AUTÓNOMA DE
MÉXICO



FACULTAD DE INGENIERÍA

Laboratorio de microcomputadoras

PRACTICA #7

“PUERTO SERIE SCI (ASÍNCRONO)”

EQUIPO:

- BUSTOS RAMÍREZ LUIS ENRIQUE
- EGUIARTE MORETT LUIS ANDRÉS

SEMESTRE: 2017-2

Objetivo.

Familiarizar al alumno en el uso de una Interfaz de Comunicación Serie Asíncrona de un microcontrolador.

Desarrollo.

Realizar los siguientes apartados:

1.- Utilizando el programa resuelto en la práctica No. 3, ejercicio 2 (Control de acciones), realizar las modificaciones necesarias para que ahora se controle por medio del teclado de su PC, el cuál transmitirá el comando de la acción a ejecutar.

```
processor 16f877
include<pl16f877.inc>

valor1 equ h'21'
valor2 equ h'22'
valor3 equ h'23'
v0 equ h'24'
v1 equ h'25'
v2 equ h'26'
v3 equ h'27'
v4 equ h'29'
v5 equ h'30'

org 0h
goto INICIO
org 05h

INICIO:
    clrf PORTB          ;Limpia el puerto B
    bsf STATUS,RP0
    bcf STATUS,RP1      ;Cambio al Banco 1
    clrf TRISB          ;Configura el puerto B como salida.

    ;Configuración del registro transmisor
    bsf TXSTA, BRGH      ;Bit de selección de velocidad alta,BRGH=1
    movlw d'129'         ;Velocidad 9600 baud
    movwf SPBRG          ;Carga la velocidad de comunicación
    bcf TXSTA, SYNC      ;Comunicación asincrona SYNC=0
    bsf TXSTA, TXEN      ;Activa la transmisión
    bcf STATUS, RP0      ;Cambio al Banco 0

    ;Configuración del registro receptor
    bsf RCSTA, SPEN      ;Habilita puerto serie
    bsf RCSTA, CREN      ;Configura recepción continua en modo
                        ;asíncrono

RECIBE:
    ;Registros Banderas
    btfss PIR1, RCIF     ;Verifica si la recepción esta completa
    goto RECIBE          ;No, sigue poleando(Revisando)
```

```

        movf RCREG,w          ;Si,mueve lo que recibe registro de
                                ;recepción RCREG a W
CICLO:
        clrf PORTB

        ;Verifica si la entrada por el puerto B es b"0000"
        movlw '0'             ;Mueve '0' a W
        movwf v0               ;Mueve el contenido de W a v0, v0=0
        movfw RCREG            ;Mueve el contenido de RCREG a W
        xorwf v0,w             ;Realiza v0 xor W
        btfsc STATUS,Z         ;Verifica si Z=0
        goto APG               ;NO, son iguales
                                ;SI, verifica otra opción

        ;Verifica si la entrada por el puerto B es b"0001"
        movlw '1'
        movwf v1
        movfw RCREG
        xorwf v1,w
        btfsc STATUS,Z
        goto UNOS

        ;Verifica si la entrada por el puerto B es b"0010"
        movlw '2'
        movwf v2
        movfw RCREG
        xorwf v2,w
        btfsc STATUS,Z
        goto DER

        ;Verifica si la entrada por el puerto B es b"0011"
        movlw '3'
        movwf v3
        movfw RCREG
        xorwf v3,w
        btfsc STATUS,Z
        goto IZQ

        ;Verifica si la entrada por el puerto B es "0100"
        movlw '4'
        movwf v4
        movfw RCREG
        xorwf v4,w
        btfsc STATUS,Z
        goto DERIZQ

        ;Verifica si la entrada por el puerto B es "0101"
        movlw '5'
        movwf v5
        movfw RCREG
        xorwf v5,w
        btfsc STATUS,Z
        goto ENCAPG
        goto CICLO

        ;Todos los bits del puerto B apagados

```

APG:

```

        movlw h'00'
        movwf PORTB
        goto CICLO

        ;Todos los bits del puerto B encendidos
UNOS:
        movlw h'FF'
        movwf PORTB
        goto CICLO

        ;Corrimiento al la Derecha
DER:
        movlw h'80'
        movwf PORTB
        call retardo

DER1:
        rrf PORTB,1
        call retardo
        btfss PORTB,0
        goto DER1
        goto CICLO

        ;Corrimiento a la Izquierda
IZQ:
        movlw h'01'
        movwf PORTB
        call retardo

IZQ1:
        rlf PORTB,1
        call retardo
        btfss PORTB,7
        goto IZQ1
        goto CICLO

        ;Corrimiento Derecha-Izquierda
DERIZQ:
        movlw h'80'
        movwf PORTB
        call retardo

DER2:
        rrf PORTB,1
        call retardo
        btfss PORTB,0
        goto DER2

        movlw h'01'
        movwf PORTB
        call retardo

IZQ2:
        rlf PORTB,1
        call retardo
        btfss PORTB,7
        goto IZQ2
        goto CICLO

```

```

        ;Enciende y apaga todos los bits del puerto B
ENCAPG:
        movlw h'00'
        movwf PORTB
        call retardo
        movlw h'FF'
        movwf PORTB
        call retardo
        goto CICLO

        ;Rutina que genra un Retardo
retardo:
        movlw 15h
        movwf valor1
tres movlw 60h
        movwf valor2
dos  movlw 50h
        movwf valor3
uno  decfsz valor3
        goto uno
        decfsz valor2
        goto dos
        decfsz valor1
        goto tres
        return
end

```

2.- Realizar un programa que muestre las vocales (mayúsculas y minúsculas en un display de 7 segmentos, las cuales serán enviadas vía serie a través del teclado de la PC.

```

processor 16f877
include<p16f877.inc>

;Variables para el DELAY

valor1 equ h'21'
valor2 equ h'22'
valor3 equ h'23'

;Definición de segmentos de DISPLAY

;Vocales mayúsculas
AM EQU 5FH ; ,7,,5,4,3,2,1: 01011111B
EM EQU 37H ; ,,6,5,,3,2,1: 00110111B
IM EQU 48H ; ,7,,,4,,,,: 01001000B
OM EQU 7EH ; ,7,6,5,4,3,2,,: 01111110B
UM EQU 7AH ; ,7,6,5,4,,2,,: 01111010B

;Vocales minúsculas
Am EQU 7DH ; ,7,6,5,4,3,,1: 01111101B
Em EQU 6FH ; ,7,6,,4,3,2,1: 01101111B
Im EQU 40H ; ,7,,,,,,,: 01000000B
Om EQU 71H ; ,7,6,5,,,,,1: 01110001B
Um EQU 70H ; ,7,6,5,,,,,: 01110000B

```

```
;Variable para cargar valores
var equ h'24'
```

```
org 0h
goto INICIO
org 05h
```

```
INICIO:
```

```
    clrf PORTB
    bsf STATUS,RP0
    bcf STATUS,RP1    ;Cambia al Banco 1
    clrf TRISB        ;Configura el puerto B como salida.

    ;Configuración del registro transmisor
    bsf TXSTA, BRGH    ;Bit de selección de velocidad alta,BRGH=1
    movlw d'129'       ;Velocidad 9600 baud
    movwf SPBRG        ;Carga la velocidad de comunicación
    bcf TXSTA, SYNC    ;Comunicación asincrona SYNC=0
    bsf TXSTA, TXEN    ;Activa la transmisión
    bcf STATUS, RP0    ;Cambio al Banco 0

    ;Configuración del registro receptor
    bsf RCSTA, SPEN    ;Habilita puerto serie
    bsf RCSTA, CREN    ;Configura recepción continua en modo
                        ;asíncrono
```

```
RECIBE:
```

```
    ;Registros Banderas
    btfss PIR1, RCIF   ;Verifica si la recepción esta completa
    goto RECIBE        ;No, sigue poleando(Revisando)
    movf RCREG,w       ;Si,mueve lo que recibe registro de
                        ;recepción RCREG a W
```

```
CICLO:
```

```
    clrf PORTB
```

```
    ;Verifica si es "a"
```

```
    movlw 'a'          ;Mueve 'a' a W
    movwf var           ;Mueve el contenido de W a var, var=a
    movfw RCREG         ;Mueve el contenido de RCREG a W
    xorwf var,w         ;Realiza var xor W
    btfsc STATUS,Z      ;Verifica si Z=0
    goto a              ;NO, son iguales
                        ;SI, verifica la siguiente opción
```

```
    ;Verifica si es "e"
```

```
    movlw 'e'
    movwf var
    movfw RCREG
    xorwf var,w
    btfsc STATUS,Z
    goto e
```

```
    ;Verifica si es "i"
```

```
    movlw 'i'
    movwf var
```

```

        movfw RCREG
        xorwf var,w
        btfsc STATUS,Z
        goto i

;Verifica si es "o"
        movlw 'o'
        movwf var
        movfw RCREG
        xorwf var,w
        btfsc STATUS,Z
        goto o

;Verifica si es "u"
        movlw 'u'
        movwf var
        movfw RCREG
        xorwf var,w
        btfsc STATUS,Z
        goto u

;Verifica si es "A"
        movlw 'A'
        movwf var
        movfw RCREG
        xorwf var,w
        btfsc STATUS,Z
        goto A

;Verifica si es "E"
        movlw 'E'
        movwf var
        movfw RCREG
        xorwf var,w
        btfsc STATUS,Z
        goto E

;Verifica si es "I"
        movlw 'I'
        movwf var
        movfw RCREG
        xorwf var,w
        btfsc STATUS,Z
        goto I

;Verifica si es "O"
        movlw 'O'
        movwf var
        movfw RCREG
        xorwf var,w
        btfsc STATUS,Z
        goto O

;Verifica si es "U"
        movlw 'U'
        movwf var
        movfw RCREG

```

```

        xorwf var,w
        btfsc STATUS,Z
        goto U

;Muestra vocales minúsculas en el DISPLAY
a:

        movlw Am      ;Mueve el valor de Am a W
        movwf PORTB   ;Mueve el valor de W a PORTB
        call retardo  ;Tiempo de Retardo
        goto CICLO

e:

        movlw Em
        movwf PORTB
        call retardo
        goto CICLO

i:

        movlw Im
        movwf PORTB
        call retardo
        goto CICLO

o:

        movlw Om
        movwf PORTB
        call retardo
        goto CICLO

u:

        movlw Um
        movwf PORTB
        call retardo
        goto CICLO

;Muestra vocales mayúsculas en el DISPLAY
A:

        movlw AM
        movwf PORTB
        call retardo
        goto CICLO

E:

        movlw EM
        movwf PORTB
        call retardo
        goto CICLO

I:

        movlw IM
        movwf PORTB
        call retardo
        goto CICLO

O:

        movlw OM
        movwf PORTB
        call retardo
        goto CICLO

U:

        movlw UM
        movwf PORTB
        call retardo

```



```

        goto CICLO

retardo:
        movlw h'10'    ;Rutina que genera un retardo
        movwf valor1   ; (10;4Mhz) (60;20Mhz)
tres    movlw h'50'
        movwf valor2
dos     movlw h'60'
        movwf valor3
uno     decfsz valor3
        goto uno
        decfsz valor2
        goto dos
        decfsz valor1
        goto tres
        return
end

```

3.- Empleando el programa No. 3 de la práctica 6 (convertidor analógico digital), realizar las modificaciones necesarias para desplegar el número de canal, de valor mayor a las otras entradas, en el monitor de la PC.

```

processor 16f877
include<pl16f877.inc>

; Variable para el DELAY
val equ h'20'

;Variable para los Canales
call equ h'21'
cal2 equ h'22'

org 0h
goto INICIO
org 05h

INICIO:
        clrf PORTA
        clrf PORTB
        bsf STATUS,RP0    ;Cambia la banco 1
        bcf STATUS,RP1
        movlw 00h         ;Configura puertos A y E como analógicos
        movwf ADCON1
        movlw 3fh         ;Configura el puerto A como entrada
        movwf TRISA
        movlw h'0'
        movwf TRISB       ;Configura puerto B como salida
        bcf STATUS,RP0    ;regresa al banco 0

        ;Configuración del registro transmisor
        bsf TXSTA, BRGH   ;Bit de selección de velocidad alta,BRGH=1
        movlw d'129'     ;Velocidad 9600 baud
        movwf SPBRG       ;Se carga la velocidad de comunicación
        bcf TXSTA, SYNC   ;Comunicación asincrona SYNC=0
        bsf TXSTA, TXEN   ;Activa la transmisión

```

```

    bcf STATUS, RP0    ;Cambio al banco 0

    ;Configuración del registro receptor
    bsf RCSTA, SPEN    ;Habilita puerto serie
    bsf RCSTA, CREN    ;Configura recepción continua en modo
                        ;asíncrono

CICLO
    movlw b'11000001' ;Configuración ADCON0 (Canal 0)
    movwf ADCON0
    bsf ADCON0,2       ;Conversión en progreso GO=1
    call RETARDO       ;Espera que termine la conversión

ESPERA
    btfsc ADCON0,2     ;Pregunta por DONE(Terminó conversión?)
    goto ESPERA
    movf ADRESH,0      ;Mueve el resultado de la conv. a W
    movwf call         ;Guardamos el valor en call
    movlw b'11001001' ;Configuración ADCON0 canal 1
    movwf ADCON0
    bsf ADCON0,2       ;Conversión en progreso GO=1
    call RETARDO       ;Espera que termine la conversión

ESPERA1
    btfsc ADCON0,2     ;Pregunta por DONE(Terminó conversión?)
    goto ESPERA1
    movf ADRESH,0      ;Mueve el resultado de la conv. a W
    movwf cal2         ;Guardamos el valor en cal2

    ;Compara los canales 1 y 2
    movf call,w        ;Mueve call a w
    subwf cal2         ;Realiza canal2 - canal1(w)
    btfss STATUS,0     ;Verifica si W es mayor
    goto CANAL2        ;No, entonces Ve2>Ve1
    goto CANAL1        ;Si, entonces Ve1>Ve2

CANAL1
    movlw '1'          ;Valor ASCII
    movwf TXREG         ;Mueve W al registro de transmisión TXREG
    bsf STATUS, RP0    ;Cambio al banco 1
    call TRANSMITE      ;Transmite el numero de canal
    goto CICLO

CANAL2
    movlw '2'          ;Valor ASCII
    movwf TXREG         ;Mueve W al registro de transmisión TXREG
    bsf STATUS, RP0    ;Cambio al banco 1
    call TRANSMITE      ;Transmite el numero de canal
    goto CICLO

RETARDO
    ;DELAY de 2micoSeg aprox.
    movlw h'30'
    movwf val
loop   decfsz val
    goto loop
    return

```

```
TRANSMITE                ;Rutina de TRANSMISION
VERIFICA:                ;LOOP de Verificación
    btfss TXSTA,TRMT     ;Transmisión de datos completa?
    goto VERIFICA        ;No, sigue verificando
    bcf STATUS,RP0       ;Si, cambia al banco 0
    return
end
```

Conclusiones.

Con esta práctica nos fue posible familiarizarnos con la interfaz de comunicación serie asíncrona del microcontrolador PIC16F877, con el uso de esta interfaz nos fue posible aprender a usarla para efectuar diversas aplicaciones que ya habían sido realizadas con anterioridad en prácticas anteriores, solo que para este caso hicimos uso de esta interfaz para su realización.