



UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO

FACULTAD DE INGENIERÍA

Práctica 7: Puerto Serie SCI (Asíncrono)

INTEGRANTES DE EQUIPO Y GRUPO DE TEORÍA

López Becerra Ricardo - 420053710 - GRUPO 3

Navarrete Zamora Aldo Yael - 317242409 - GRUPO 4

ASIGNATURA

Laboratorio de Microcomputadoras

GRUPO DE LABORATORIO

8

FECHA DE REALIZACIÓN

11 de mayo del 2023

FECHA DE ENTREGA

25 de mayo de 2023

1. Desarrollo

1. Escribir, comentar y ensamblar el siguiente código.

	processor 16f877 include<p16f877.inc>	RECIBE:	BTFSS PIR1,RCIF GOTO RECIBE
	ORG 0 GOTO inicio		MOVF RCREG,W MOVWF TXREG
INICIO:	ORG 5 BSF STATUS,RP0 BCF STATUS,RP1 BSF TXSTA,BRGH MOVLW D'129' MOVWF SPBRG BCF TXSTA,SYNC BSF TXSTA,TXEN	TRASMITE:	BSF STATUS,RP0 BTFSS TXSTA,TRMT GOTO TRASMITE BCF STATUS,RP0 GOTO RECIBE
	BCF STATUS,RP0		END
	BSF RCSTA,SPEN BSF RCSTA,CREN		

2. Abrir la terminal de su elección, configurar esta a 9600 Bauds y un protocolo 8 bits de inicio, 8 bits de datos y un 1 de paro; comprobar su funcionamiento.

Comentarios

En estos dos primeros ejercicios se da el código base para poder utilizar el puerto serial del microcontrolador. Para realizar esto, se debe configurar el tipo de tasa, se debe indicar si se desea comunicación síncrona o asíncrona, se debe habilitar el transmisor, el puerto serie y se le debe indicar si se quiere recibir información de manera continua. De entre todos los dispositivos vistos hasta ahora, este es el que más configuraciones necesita.

Al ensamblar el código y ejecutarlo como indica el manual, se puede ver que lo programado es una función ".eco", por lo que el microcontrolador devuelve los datos que se le envían.

Código comentado

```
processor 16f877
include<p16f877.inc>
ORG 0
GOTO INICIO
ORG 5
INICIO:
;CAMBIA DE BANCO
BSF STATUS,RP0
BCF STATUS,RP1
;TASA ALTA
BSF TXSTA,BRGH
;TASA DE TRANSFERENCIA
MOVLW D'129'
MOVWF SPBRG
; coMUNICACIoN ASINCRoNA
```

```
        BCF TXSTA,SYNC
;HABILITA EL TRANSMISOR
        BSF TXSTA,TXEN
        BCF STATUS,RP0
;HABILITA EL PUERTO SERIE
        BSF RCSTA,SPEN
;HABILITA LA RECEPCION CONTINUA
        BSF RCSTA,CREN
RECIBE:
        ;ESPERA A QUE LLEGE UN DATO
        BTFSS PIR1,RCIF
        GOTO RECIBE
        MOVF RCREG,W
        ;ENVIA LOS DATOS RECIBIDOS PARA TRANSMITIRLO
        MOVWF TXREG
        ;CAMBIA AL BANCO 1
        BSF STATUS,RP0
TRASMITE:
        ;UN LOOP ESPERANDO A QUE SE ENVIEN LOS DATOS
        BTFSS TXSTA,TRMT
        GOTO TRASMITE
        ;REGRESA AL BANCO 0
        BCF STATUS,RP0
        GOTO RECIBE
END
```

3. Realizar un programa que despliegue la siguiente cadena en una terminal. "HOLA UNAM".

Algoritmo

El algoritmo inicia configurando los registros BRGH TXSTA y el SPBRG, una vez que los tenemos configurados para la correcta comunicación asíncrona con el seteo de SYNC, podemos escuchar a que el usuario ponga una entrada en la terminal, cuando recibamos el dato, lo que pasará es que el programa ejecutará una secuencia, en la que mandará letra por letra la cadena solicitada, primero enviará la 'H', después la 'O', y así sucesivamente usando retardos intermedios.

Código comentado

```
        processor 16f877
        include <p16f877.inc>
        cteA equ 0x20
        ORG 0
        GOTO INICIO
        ORG 5
INICIO:
        ;CAMBIA DE BANCO
        BSF STATUS,RP0
        BCF STATUS,RP1
        CLRF TRISB
        ;TASA ALTA
```

```
        BSF TXSTA,BRGH
;TASA DE TRANSFERENCIA
        MOVLW D'129'
        MOVWF SPBRG
; COMUNICACION ASINCRONA
        BCF TXSTA,SYNC
;HABILITA EL TRANSMISOR
        BSF TXSTA,TXEN
        BCF STATUS,RP0
;HABILITA EL PUERTO SERIE
        BSF RCSTA,SPEN
;HABILITA LA RECEPCION CONTINUA
        BSF RCSTA,CREN
        CLRF PORTB

RECIBE:                                ; Etiqueta "RECIBE"
        ; ESPERA A QUE LLEGUE UN DATO
        BTFSS PIR1,RCIF                ; Verificar RCIF (indicando recibido un dato)
        GOTO RECIBE                    ; Saltar "RECIBE" si RCIF esta en '0'
        MOVLW h'48'                    ; Cargar el valor hexadecimal '48' en W

W
        MOVWF TXREG                    ; Mover el valor de W a TXREG
        call retardo                    ; Llamar a la subrutina "retardo"

        ; CAMBIA AL BANCO 1
        BSF STATUS,RP0

TRASMITE_H:                            ; Etiqueta "TRASMITE_H"
        BTFSS TXSTA,TRMT                ; Verificar TRMT en TXSTA
        GOTO TRASMITE_H                ; Saltar "TRASMITE_H" si TRMT esta en '0'

        ; ... (continua de manera similar para las
        siguientes secciones de transmision)

        ; REGRESA AL BANCO 0
        BCF STATUS,RP0                ; Limpiar el bit RP0 del registro STATUS
        GOTO RECIBE                    ; Saltar a la etiqueta "RECIBE"

retardo:                                ;
        MOVLW 0X30                    ;
        MOVWF cteA                    ;

LOOPR:
        DECFSZ cteA                    ;
        GOTO LOOPR                    ;
        RETURN                        ; CODIGO RETARDO

END
```

4. Realizar un programa que ejecute el control indicado; el dato proviene a través del puerto serie:

DATO	ACCION
Puerto Serie	Terminal 0 del puerto B (PB0)
'0'	0
'1'	1

Algoritmo

Para este ejercicio simplemente se debía controlar el bit 0 del puerto B con los caracteres de la terminal. Para esto, el código recibe los datos del puerto serie y mueve los datos al registro Option. A través de restas, se determina si es 1 o 0 la opción elegida y si no es ninguna de las dos opciones regresa a la etapa de lectura.

Una vez que se determinó que la opción es 0 o 1, en el registro W se carga el valor correspondiente para el PORTB que inmediatamente después es limpiado y se carga el valor de W. Finalmente, transmitimos el valor que cargamos en el PORTB por el puerto serie.

Código comentado

```
processor 16f877
include<p16f877.inc>
valor1 equ h'31'
valor2 equ h'32'
valor3 equ h'33'
cte1 equ 20h
cte2 equ 50h
cte3 equ 60h
Opcion equ H'34'
ORG 0
GOTO INICIO

ORG 5
INICIO
;Cambiamos al banco 1
BSF STATUS,RP0
BCF STATUS,RP1
MOVLW H'0'
MOVWF TRISB
;Activamos el BRGH
BSF TXSTA,BRGH
MOVLW D'129'
;Configuramos el baud rate a 9600
MOVWF SPBRG
;Configuramos la asincrona
BCF TXSTA,SYNC
;Activamos el transmisor
BSF TXSTA,TXEN
;Volvemos al banco 0
BCF STATUS,RP0
```

```
;Habilitamos la recepcion
BSF RCSTA,SPEN
;Habilitamos la recepcion continua
BSF RCSTA,CREN
RECIBE:
;Verificamos la bandera recepcion
BTFSS PIR1,RCIF
GOTO RECIBE
;Copiamos lo recibido a W
MOVF RCREG,W
MOVWF Opcion
;Comparamos con 0
MOVLW .0
SUBWF Opcion, W
BTFSC STATUS,Z
GOTO CERO
;Comparamos con 1
MOVLW .1
SUBWF Opcion, W
BTFSC STATUS,Z
GOTO UNO
GOTO RECIBE

;Dependiendo de si se
; recibio 0 o 1 se carga ese valor
; en el registro W
UNO:
MOVLW H'01 '
GOTO CONTINUA
CERO:
MOVLW H'00 '
GOTO CONTINUA

CONTINUA:
;Lo transmitimos
MOVWF TXREG
CLRF PORTB
MOVWF PORTB
;Vamos al banco 1
BSF STATUS,RP0
TRANSMITE:
;Comprobamos si ya termino de transmitir
BTFSS TXSTA,TRMT
GOTO TRANSMITE
;Volvemos al banco 0
BCF STATUS,RP0
GOTO RECIBE
END
```

5. Realizar un programa que ejecute el control indicado; la secuencia será reconocida cada

que sea recibido el comando, usar retardos de 1/2 segundo entre cada estado generado:

DATO Puerto Serie	ACCION Salida Puerto B
'D' ó 'd'	10000000
	01000000
	00100000
	00010000
	00001000
	00000100
	00000010
	00000001
'I' ó 'i'	00000001
	00000010
	00000100
	00001000
	00010000
	00100000
	01000000
	10000000

Algoritmo

Este caso fue similar al anterior, solo que en este caso en dos casos se podía ir a cada opción porque se podía entrar a ambos tipos de corrimiento con la letra mayúscula y minúscula indicada. Una vez seleccionada la opción correcta, se llaman a las subrutinas hechas en prácticas anteriores para corrimientos, con la diferencia de que estas no se pueden detener a mitad de su ejecución debido a que recibir datos por el puerto serial es bloqueante.

Código comentado

```
processor 16f877
include<p16f877.inc>
valor1 equ h'31'
valor2 equ h'32'
valor3 equ h'33'
cte1 equ H'FF'
cte2 equ 50h
cte3 equ 60h
Opcion equ H'34'
ORG 0
GOTO INICIO
```

```
ORG 5
INICIO
;Cambiamos al banco 1
BSF STATUS,RP0
```

```
BCF STATUS,RP1
MOVLW H'0 '
MOVWF TRISB
;Activamos el BRGH
BSF TXSTA,BRGH
MOVLW D'129 '
;Configuramos el baud rate a 9600
MOVWF SPBRG
;Configuramos la asincrona
BCF TXSTA,SYNC
;Activamos el transmisor
BSF TXSTA,TXEN
;Volvemos al banco 0
BCF STATUS,RP0
;Habilitamos la recepcion
BSF RCSTA,SPEN
;Habilitamos la recepcion continua
BSF RCSTA,CREN
```

```
loop1:
;LLamamos a la subrutina que recibe
; los datos
call RECIBE
;Dependiendo del valor ASCII del
;caracter recibido elegimos la
; opcion correcta
MOVLW H'44 '
SUBWF Opcion , W
BTFSC STATUS,Z
GOTO RIGHT_SHIFT
```

```
MOVLW H'64 '
SUBWF Opcion , W
BTFSC STATUS,Z
GOTO RIGHT_SHIFT
```

```
MOVLW H'49 '
SUBWF Opcion , W
BTFSC STATUS,Z
GOTO LEFT_SHIFT
```

```
MOVLW H'69 '
SUBWF Opcion , W
BTFSC STATUS,Z
GOTO LEFT_SHIFT
```

```
;Subrutina que abstrae el proceso de
;recepcion de datos
```



```
RECIBE
; Verificamos la bandera recepcion
BTFSS PIR1,RCIF
GOTO RECIBE
; Copiamos lo recibido a W
MOVF RCREG,W
MOVWF Opcion
return
```

6. Descargar la aplicación practica7.apk e instalar en su dispositivo móvil (Android), realizar un programa para el microcontrolador, de manera que reciba el comando a través del puerto serie, con conexión inalámbrica (bluetooth), para que genere el control indicado en la tabla ; usar la asignación mostrada.

Notas importantes:

- El dato que recibe es el código ASCII del carácter transmitido.
- Para vincularse con el dispositivo Bluetooth deberá comprobar su identificador.
- Considerar la ubicación de las señales de control y los valores encontrados en la práctica 5.

MOTOR2		
PC2	PB3	PB2
ENABLE M2	DIR1 M2	DIR2 M2

MOTOR1		
PC1	PB1	PB0
ENABLE M1	DIR1 M1	DIR2 M1

Comando Puerto serie	ACCION	
	MOTOR M1	MOTOR M2
'S'	PARO	PARO
'A'	DERECHA	DERECHA
'T'	IZQUIERDA	IZQUIERDA
'D'	DERECHA	IZQUIERDA
'I'	IZQUIERDA	DERECHA

Algoritmo

Primero, configuramos los registros de recepción y transmisión, para posteriormente realizar la configuración asíncrona y activar el transmisor. Posteriormente, en la función RECIBE, esperamos a que el dato sea recibido mediante la comunicación serial, una vez que haya sido recibido el dato, comparamos con sus caracteres ASCII si se trata de alguna de las letras mencionadas en la parte de arriba. Dependiendo de en qué letra haya coincidido nos vamos a la etiqueta que le corresponde, ya sea un giro horario o un giro antihorario, en este caso es importante mencionar que los motores de esta tarjeta estaban conectados al revés, entonces simplemente tuvimos que cambiar algunos sentidos dentro del código.

Código comentado

```
processor 16f877          ; Definición del procesador utilizado
include<p16f877.inc>      ; Incluir el archivo de inclusión del procesador
```



```
valor1 equ h'31'      ; Definicion de una constante "valor1"
valor2 equ h'32'      ; Definicion de una constante "valor2"
valor3 equ h'33'      ; Definicion de una constante "valor3"
cte1 equ H'FF'        ; Definicion de una constante "cte1"
cte2 equ 50h          ; Definicion de una constante "cte2"
cte3 equ 60h          ; Definicion de una constante "cte3"
Opcion equ H'34'      ; Definicion de una constante "Opcion"

ORG 0                 ; Ubicacion inicial del programa
GOTO INICIO           ; Salto a la etiqueta "INICIO"

ORG 5                 ; Ubicacion especifica del programa
INICIO:              ; Etiqueta "INICIO"
BSF STATUS,RP0        ; Cambiar al banco 1
BCF STATUS,RP1        ; Limpiar RP1 para asegurar el banco 1
MOVLW H'0'            ; Cargar el valor hexadecimal '0' en el registro W
MOVWF TRISB           ; Mover el contenido de W al registro TRISB
BSF TXSTA,BRGH        ; Activar el bit BRGH en el registro TXSTA
MOVLW D'129'          ; Cargar el valor decimal '129' en el registro W
MOVWF SPBRG           ;
BCF TXSTA,SYNC        ; Limpiar el bit SYNC en el registro TXSTA
BSF TXSTA,TXEN        ; Activar el bit TXEN en el registro TXSTA
BCF STATUS,RP0        ; Volver al banco 0
BSF RCSTA,SPEN        ; Activar el bit SPEN en el registro RCSTA
BSF RCSTA,CREN        ; Activar el bit CREN en el registro RCSTA

loop1:               ; Etiqueta "loop1"
call RECIBE           ; Llamar a la subrutina "RECIBE"
CLRF PORTC            ; Limpiar el contenido del puerto C
CLRF PORTB            ; Limpiar el contenido del puerto B
MOVLW H'41'           ; Cargar el valor hexadecimal '41' en W
SUBWF Opcion, W       ; Restar el contenido de "Opcion" a W
BTFSC STATUS,Z        ;
GOTO H_H              ;

MOVLW H'44'           ; Cargar el valor hexadecimal '44' en W
SUBWF Opcion, W       ; Restar el contenido de "Opcion" a W
BTFSC STATUS,Z        ; Saltar si la banderaZ = 1
GOTO H_AH              ;

MOVLW H'49'           ; Cargar el valor hexadecimal '49' en W
SUBWF Opcion, W       ; Restar el contenido de "Opcion" a W
BTFSC STATUS,Z        ; Saltar si la banderaZ = 1
GOTO AH_H              ;

MOVLW H'54'           ; Cargar el valor hexadecimal '54' en W
SUBWF Opcion, W       ; Restar el contenido de "Opcion" a W
BTFSC STATUS,Z        ; Saltar si la banderaZ = 1
GOTO AH_AH            ;
```



```
MOVLW H'53'           ; Cargar el valor hexadecimal '53' en W
SUBWF Opcion, W        ; Restar el contenido de "Opcion" a W
BTFSC STATUS,Z         ; Saltar si Z = 1
GOTO PARO_PARO         ;

PARO_PARO:             ; Etiqueta "PARO_PARO"
    BCF PORTC, 2        ; Limpiar el bit 2 del puerto C
    BCF PORTC, 1        ; Limpiar el bit 1 del puerto C
    GOTO loop1          ; Volver a la etiqueta "loop1"

PARO_H:                ; Etiqueta "PARO_H"
    BSF PORTC, 2        ; Activar el bit 2 del puerto C
    BSF PORTC, 1        ; Activar el bit 1 del puerto C
    BSF PORTB, 0        ; Activar el bit 0 del puerto B
    BCF PORTB, 1        ; Limpiar el bit 1 del puerto B
    GOTO loop1          ; Volver a la etiqueta "loop1"

H_PARO:                ; Etiqueta "H_PARO"
    BCF PORTC, 1        ; Limpiar el bit 1 del puerto C
    BSF PORTC, 2        ; Activar el bit 2 del puerto C
    BSF PORTB, 2        ; Activar el bit 2 del puerto B
    BCF PORTB, 3        ; Limpiar el bit 3 del puerto B
    GOTO loop1          ; Volver a la etiqueta "loop1"

AH_AH:                ; Etiqueta "AH_AH"
    BSF PORTC, 1        ; Activar el bit 1 del puerto C
    BSF PORTC, 2        ; Activar el bit 2 del puerto C
    BSF PORTB, 3        ; Activar el bit 3 del puerto B
    BCF PORTB, 2        ; Limpiar el bit 2 del puerto B
    BSF PORTB, 0        ; Activar el bit 0 del puerto B
    BCF PORTB, 1        ; Limpiar el bit 1 del puerto B
    GOTO loop1          ; Volver a la etiqueta "loop1"

H_H:                  ; Etiqueta "H_H"
    BSF PORTC, 1        ; Activar el bit 1 del puerto C
    BSF PORTC, 2        ; Activar el bit 2 del puerto C
    BSF PORTB, 2        ; Activar el bit 2 del puerto B
    BCF PORTB, 3        ; Limpiar el bit 3 del puerto B
    BSF PORTB, 1        ; Activar el bit 1 del puerto B
    BCF PORTB, 0        ; Limpiar el bit 0 del puerto B
    GOTO loop1          ; Volver a la etiqueta "loop1"

H_AH:                 ; Etiqueta "H_AH"
    BSF PORTC, 1        ; Activar el bit 1 del puerto C
    BSF PORTC, 2        ; Activar el bit 2 del puerto C
    BSF PORTB, 3        ; Activar el bit 3 del puerto B
    BCF PORTB, 2        ; Limpiar el bit 2 del puerto B
    BSF PORTB, 1        ; Activar el bit 1 del puerto B
    BCF PORTB, 0        ; Limpiar el bit 0 del puerto B
```

```
GOTO loop1                ; Volver a la etiqueta "loop1"

AH_H:                      ; Etiqueta "AH_H"
    BSF PORTC, 1            ; Activar el bit 1 del puerto C
    BSF PORTC, 2            ; Activar el bit 2 del puerto C
    BSF PORTB, 2            ; Activar el bit 2 del puerto B
    BCF PORTB, 3            ; Limpiar el bit 3 del puerto B
    BSF PORTB, 0            ; Activar el bit 0 del puerto B
    BCF PORTB, 1            ; Limpiar el bit 1 del puerto B
    GOTO loop1              ; Volver a la etiqueta "loop1"

RECIBE:                     ; Etiqueta "RECIBE"
    BTFSS PIR1,RCIF         ; Verificar la bandera RCIF en el registro PIR1 (E
    GOTO RECIBE             ; Saltar a la etiqueta RECIBE si la bandera RCIF e
    MOVF RCREG,W            ; Mover el contenido del registro RCREG (Receive I
    MOVWF Opcion            ; Mover el contenido de W a la variable "Opcion"
    return                  ; Retornar de la subrutina "RECIBE"

retardo                      ;
    MOVLW cte1              ;
    MOVWF valor1            ;
tres                          ;
    MOVLW cte2              ;
    MOVWF valor2            ;
dos                           ;
    MOVLW cte3              ;
    MOVWF valor3            ;
uno                           ;
    DECFSZ valor3           ;
    GOTO uno                ;
    DECFSZ valor2           ;
    GOTO dos                ;
    DECFSZ valor1           ;
    GOTO tres               ;
    RETURN                  ; CODIGO DEL RETARDO
END
```

2. Conclusiones

- López Becerra Ricardo: Este periférico es uno de los más importante del microcontrolador debido a que permite que nos comuniquemos con otra computadora de manera sencilla. Además, tiene mucho potencial esta opción dado que nos permite enviar datos de manera legible para un usuario en una terminal.
- Navarrete Zamora Aldo Yael: Esta práctica nos muestra como podemos hacer uso del puerto serial para interactuar o poder comunicar la tarjeta PIC con nuestra comptutadora, podemos instalar Putty o mediante algún bootloader poder visualizar la salida en terminal del PIC. Considero que el objetivo de realizar la comunicación en serie de forma asíncrona se ha logrado de manera exitosa.