

FACULTAD DE INGENIERÍA, UNAM

LABORATORIO DE MICROCOMPUTADORASS

SEMESTRE 2023-2

GRUPO 11

PREVIO PRÁCTICA 7

PUERTO SERIE SCI(ASÍNCRONO)

NOMBRE DEL ALUMNO:

ARRIAGA MEJÍA JOSÉ CARLOS

PROFESOR

ING. ROMAN V. OSORIO COMPARAN

FECHA DE ENTREGA: 05 DE MAYO DE 2023

CALIFICACION

Objetivo

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

1.- Escribir, comentar y ensamblar el siguiente código

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

ORG 0
GOTO INICIO

ORG 5
INICIO:
BSF STATUS,RP0;Banco 01
BCF STATUS,RP1
BSF TXSTA,BRGH;Configura la velocidad de comunicación BRGH=0
MOVLW D'129'
MOVWF SPBRG;Baund rate de 2400
BCF TXSTA,SYNC ;Desactiva comunicación sincrona
BSF TXSTA,TXEN ;activa transmisión
BCF STATUS,RP0 ;Banco 00
BSF RCSTA,SPEN ;habilita puerto serie
BSF RCSTA,CREN ;activa la recepción continua en modo de comunicación asíncrona
RECIBE:
BTFSS PIR1,RCIF ;Verfifica si la recepción fue completa
GOTO RECIBE
MOVF RCREG,W ;guarda el mesanje
MOVWF TXREG ;manda el mensaje a transmisión
BSF STATUS,RP0 ;cambia a banco 01
TRASMITE:
BTFSS TXSTA,TRMT ;verifica que se haya terminado de transmitir
GOTO TRASMITE
BCF STATUS,RP0 ;cambia a banco 00
GOTO RECIBE
END
```

3.- Realizar un programa que despliegue la siguiente cadena en una terminal. Ejemplo: Hola mundo

```

processor 16f877
include<pl6f877.inc>
    ORG 0
    GOTO INICIO
    ORG 5
INICIO:
    BSF STATUS,RP0 ; Cambiamos al banco 01
    BCF STATUS,RP1 ;
    BSF TXSTA,BRGH ; prende bit BRGH --> alta velocidad
    MOVLW D'129' ; W = 129, como BRGH = 1 => 129->9600 baudios
    MOVWF SPBRG ; se configura a 9600 baudios
    BCF TXSTA,SYNC ; apagamos bit SYNC --> modo asíncrono
    BSF TXSTA,TXEN ; prende bit TXEN --> habilita transmisión
    BCF STATUS,RP0 ; cambiamos al banco 00
    BSF RCSTA,SPEN ; prende bit SPEN --> habilita el puerto serie
TRANS:
    MOVLW 'H' ; carga el valor de H a W
    MOVWF TXREG ; carga el registro de transferencia con W
    BSF STATUS,RP0 ; cambiamos al banco 01
TRANSMITE1:
    BTFSS TXSTA,TRMT ; chequea si se realizó la transmisión
    GOTO TRANSMITE1 ; esperamos bandera transmisión
    BCF STATUS,RP0 ; cambiamos al banco 00
    MOVLW 'O' ;
    MOVWF TXREG ;
    BSF STATUS,RP0 ;
TRANSMITE2:
    BTFSS TXSTA,TRMT ;
    GOTO TRANSMITE2 ;
    BCF STATUS,RP0 ;
    MOVLW 'L' ;
    MOVWF TXREG ;
    BSF STATUS,RP0 ;
TRANSMITE3:
    BTFSS TXSTA,TRMT ;
    GOTO TRANSMITE3 ;
    BCF STATUS,RP0 ;
    MOVLW 'A' ;
    MOVWF TXREG ;
    BSF STATUS,RP0 ;

```

```

TRANSMITE4:
    BTFSS TXSTA,TRMT ;
    GOTO TRANSMITE4 ;
    BCF STATUS,RP0 ;
    MOVLW ' ' ;
    MOVWF TXREG ;
    BSF STATUS,RP0 ;
TRANSMITE5:
    BTFSS TXSTA,TRMT ;
    GOTO TRANSMITE5 ;
    BCF STATUS,RP0 ;
    MOVLW 'm' ;
    MOVWF TXREG ;
    BSF STATUS,RP0 ;
TRANSMITE6:
    BTFSS TXSTA,TRMT ;
    GOTO TRANSMITE6 ;
    BCF STATUS,RP0 ;
    MOVLW 'u' ;
    MOVWF TXREG ;
    BSF STATUS,RP0 ;
TRANSMITE7:
    BTFSS TXSTA,TRMT ;
    GOTO TRANSMITE7 ;
    BCF STATUS,RP0 ;
    MOVLW 'n' ;
    MOVWF TXREG ;
    BSF STATUS,RP0 ;
TRANSMITE8:
    BTFSS TXSTA,TRMT ;
    GOTO TRANSMITE8 ;
    BCF STATUS,RP0 ;
    MOVLW 'd' ;
    MOVWF TXREG ;
    BSF STATUS,RP0 ;
TRANSMITE9:
    BTFSS TXSTA,TRMT ;
    GOTO TRANSMITE9 ;
    BCF STATUS,RP0 ;
    MOVLW 'o' ;
    MOVWF TXREG ;
    BSF STATUS,RP0 ;

TRANSMITE10:
    BTFSS TXSTA,TRMT ;
    GOTO TRANSMITE10 ;
    BCF STATUS,RP0 ;
    MOVLW d'10' ; salto de linea
    MOVWF TXREG ;
    BSF STATUS,RP0 ;
TRANSMITE11:
    BTFSS TXSTA,TRMT ;
    GOTO TRANSMITE11 ;
    BCF STATUS,RP0 ;

FIN:
    GOTO FIN ; se queda en este ciclo para siempre
;GOTO TRANS ; solo transmite una vez
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

Tabla 7.1 Control para activar y desactivar una señal

```

processor 16f877 ; Indica la versión de procesador
include<pl6f877.inc> ; Incluye la librería del procesador
ORG 0
GOTO INICIO
ORG 5
INICIO:
BSF STATUS,RP0 ; Cambiamos al banco 01
BCF STATUS,RP1 ;
MOVLW B'00000000' ; configura al puerto B 8 bits salida
MOVWF TRISE ; carga la configuracion
BSF TXSTA,BRGH ; prende bit BRGH --> alta velocidad
MOVLW D'129' ; W=129, como BRGH=1 => 129->9600 baudios
MOVWF SPBRG ; se configura a 9600 baudios
BCF TXSTA,SYNC ; apagamos bit SYNC --> modo asíncrono
BCF STATUS,RP0 ; cambiamos al banco 00
BSF RCSTA,SPEN ; prende bit SPEN --> habilita el puerto serie
BSF RCSTA,CREN ; prende bit CREN --> haiblita recepcion
RECIBE:
BTFS PIR1,RCIF ; espera en bucle a que termine recepción
GOTO RECIBE
BCF STATUS,Z
MOVLW '0'
SUBWF RCREG,W ; W = RCREG - W = RCREG - 0
BTFS STATUS,Z ;
GOTO APAGA ; apaga salida
BCF STATUS,Z
MOVLW '1'|
SUBWF RCREG,W ; W = RCREG - W = RCREG - 1
BTFS STATUS,Z
GOTO PRENDE ; prende salida
GOTO RECIBE ; se mantiene igual
APAGA:
BCF PORTE,0 ; se limpia bit 0 puerto b
GOTO RECIBE
PRENDE:
BSF PORTE,0 ; se prende bit 0 puerto b
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 ½ 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

Tabla 7.2 Secuencia de control

```

processor 16f877 ; Indica la versión de procesador
include<pl6f877.inc> ; Incluye la librería del procesador
valor1 equ h'21'
valor2 equ h'22'
valor3 equ h'23'
ctel equ 50
cte2 equ .200
cte3 equ .82
ORG 0
GOTO INICIO
ORG 5
INICIO:
    BSF STATUS,RP0 ; Cambiamos al banco 01
    BCF STATUS,RP1 ;
    MOVLW B'00000000' ; configura al puerto B como 8 salidas
    MOVWF TRISE ; carga la configuracion
    BSF TXSTA,BRGH ; prende bit BRGH --> alta velocidad
    MOVLW D'129' ; W=129, como BRGH=1 => 129->9600 baudios
    MOVWF SPBRG ; se configura a 9600 baudios
    BCF TXSTA,SYNC ; apagamos bit SYNC --> modo asíncrono
    BCF STATUS,RP0 ; cambiamos al banco 00
    BSF RCSTA,SPEN ; prende bit SPEN --> habilita el puerto serie
    BSF RCSTA,CREN ; prende bit CREN --> habilita recepción
RECIBE:
    BTFSS PIR1,RCIF ; espera en bucle a que termine recepción
    GOTO RECIBE ; vuelve a esperar
    BCF STATUS,Z ; limpia bandera zero
    MOVLW 'D' ; W <-- 'D'
    SUBWF RCREG,W ; W = RCREG - W = RCREG - D
    BTFSC STATUS,Z ; checa si la resta fue 0 y char fue D
    GOTO DERECHA ; CORRIMIENTO DERECHA
    BCF STATUS,Z ; limpia Z
    MOVLW 'd' ; W <-- 'd'
    SUBWF RCREG,W ; W = RCREG - W = RCREG - d
    BTFSC STATUS,Z ; checa si la resta fue 0 y char fue i
    GOTO DERECHA ; CORRIMIENTO DERECHA
    BCF STATUS,Z
    MOVLW 'I'
    SUBWF RCREG,W ; W = RCREG - W = RCREG - I
    BTFSC STATUS,Z ; checa si la resta fue 0 y char fue I
    GOTO IZQUIERDA ; CORRIMIENTO IZQUIERDA
    BCF STATUS,Z
    MOVLW 'i'
    SUBWF RCREG,W ; W = RCREG - W = RCREG - i
    BTFSC STATUS,Z ; checa si la resta fue 0 y char fue i
    GOTO IZQUIERDA ; CORRIMIENTO IZQUIERDA

```

```

    GOTC RECIBE ; va a recibir siguiente char
    DERECHA:
    BCF STATUS,C ; limpia carry
    MOVLW b'10000000' ;
    MOVWF PORTE ; carga 10000000 en el puerto b
CORRDER:
    CALL RETARDO
    RRF PORTE ; corrimiento derecha
    BTFSS STATUS,C ; checa si se desborda
    GOTC CORRDER ; vuelve a otro corrimiento
    GOTC RECIBE ; va a recibir siguiente char
IZQUIERDA:
    BCF STATUS,C ; limpia carry
    MOVLW b'00000001' ;
    MOVWF PORTE ; carga 00000001 en el puerto b
CORRIZQ:
    CALL RETARDO
    RLF PORTE ; corrimiento izquierda
    BTFSS STATUS,C ; checa si se desborda
    GOTC CORRIZQ ; vuelve a otro corrimiento
    GOTC RECIBE ; va a recibir siguiente char
RETARDO ; subrutina de retardo
    MOVLW cte1
    MOVWF valor1
    tres
    MOVLW cte2
    MOVWF valor2
    dos
    MOVLW cte3
    MOVWF valor3
    uno
    DECFSZ valor3
    GOTC uno
    DECFSZ valor2
    GOTC dos
    DECFSZ valor1
    GOTC tres
    RETURN
END

```

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), par que genere el control indicado en la tabla 7.4; usar la asignación mostrada en la tabla 7.3.

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

Tabla 7.3. Asignación de señales de control de los motores de CD.

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

Tabla 7.4 Control de motores, comunicación serie

Control de motores P76

Proyecto PAPIME PE109719

