



Universidad Nacional Autónoma de México

Facultad de Ingeniería

Laboratorio de Microcomputadoras

Nombre del Profesor:

Amaranto de Jesús Dávila Jauregui

Semestre 2023-2

Grupo 3

Práctica 3b.

Nombre del alumno:

- ❖ Castelan Ramos Carlos
- ❖ Corona Nava Pedro Jair

Fecha de entrega: 23/05/2023

Práctica 3b.

Introducción:

El PIC16F877A es un microcontrolador de la familia PIC (Peripheral Interface Controller) fabricado por Microchip Technology. Es ampliamente utilizado en aplicaciones que requieren control y procesamiento de datos en tiempo real, como sistemas embebidos, automatización industrial y dispositivos electrónicos de consumo.

El PIC16F877A cuenta con una arquitectura de conjunto de instrucciones reducido (RISC) y se basa en una estructura de registros, lo que lo hace eficiente y fácil de programar. Tiene una amplia gama de periféricos integrados, como puertos GPIO, convertidores analógico a digital (ADC), temporizadores, módulos de comunicación serie, entre otros, que permiten una mayor flexibilidad en el diseño de sistemas.

Una de las instrucciones importantes en el PIC16F877A es la instrucción FSR (File Select Register) que se utiliza para acceder a la memoria de datos del microcontrolador. La instrucción FSR permite seleccionar una dirección de memoria específica para realizar operaciones de lectura o escritura. Al manipular el contenido del registro FSR, se puede acceder a diferentes ubicaciones de memoria, lo que proporciona una forma eficiente de gestionar datos en el programa.

La instrucción INCF (Increment F) es otra instrucción útil en el PIC16F877A. Esta instrucción se utiliza para incrementar el contenido de un registro especificado por el FSR. Al aplicar la instrucción INCF al FSR, se incrementa el valor almacenado en la ubicación de memoria seleccionada por el FSR. Esto es especialmente útil en casos donde es necesario realizar un recorrido o manipulación de datos en una región de memoria de manera secuencial.

En resumen, el PIC16F877A es un microcontrolador versátil que ofrece un conjunto de instrucciones potentes y periféricos integrados. La instrucción FSR permite acceder a la memoria de datos de forma flexible, mientras que la instrucción INCF facilita la manipulación de datos en una secuencia determinada. Estas características hacen del PIC16F877A una opción popular para una amplia variedad de aplicaciones embebidas.

Instrucciones:

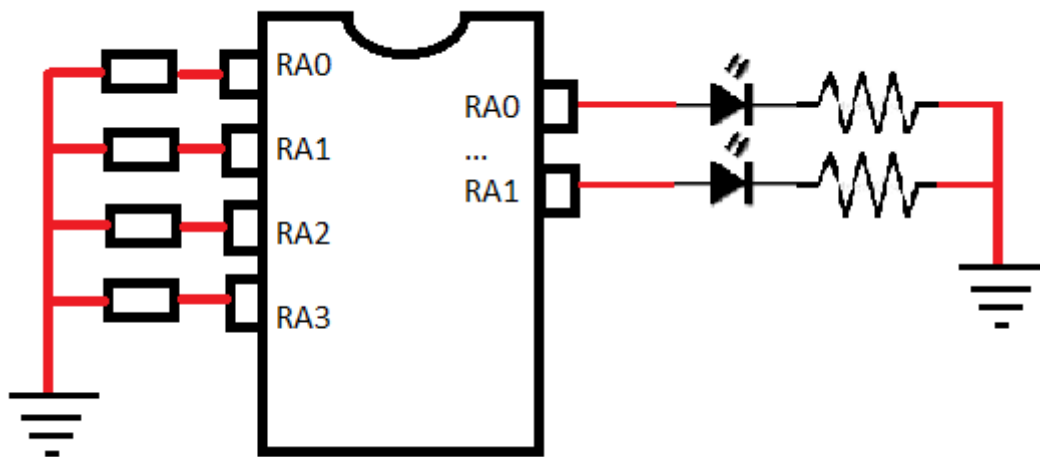
Diseñar un sistema que grabe una secuencia de máximo 16 comandos en la memoria de datos y los reproduzca.

El sistema cuenta con:

- 8 leds en donde se muestra la acción correspondiente al comando grabado.
- 2 dip switch (RA0:RA1) con los que se especifica el comando.
- 1 dip switch (RA2) para grabar comandos.
- 1 dip switch (RA3) para reproducir comandos grabados.

Comandos:

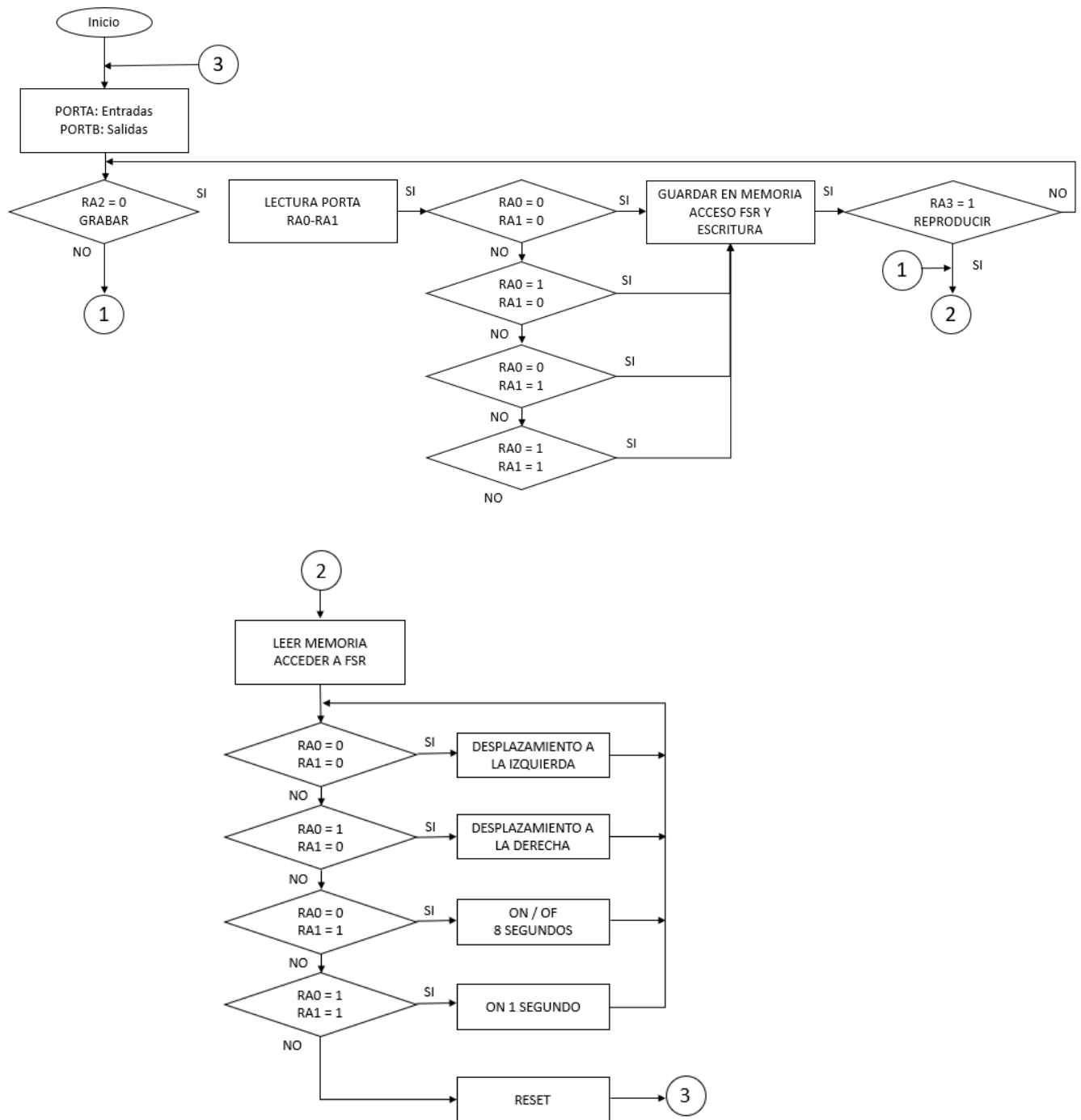
RA1	RA0	Funciones
0	0	Desplazamiento a la izquierda (150 ms)
0	1	Desplazamiento a la derecha (150 ms)
1	0	ON / OFF 8 veces
1	1	ON 1 segundos



Propuesta de solución:

Realización de un programa que mediante la definición de puertos como entrada y salida a través del TRIS logremos interpretar las entradas como instrucciones para grabado haciendo uso de de las localidades de memoria a partir de los comando FSR e INCF para seleccionar la localidad de memoria a para guardar un dato y así ir incrementando la posición de las direcciones. Enseguida que se desee grabar seleccionaremos la función a realizar para así guardarla y luego preguntar si se quiere reproducir para que finalmente se pueda acceder a las localidades FSR y se pueda reproducir las funciones.

Diagrama de flujo:



Código:

Para este código, primeramente se realizaron el cambio de banco de memoria para configurar al PORTA como entrada y al PORTB como salida, enseguida, mediante el uso de condicionales con BTFSS a partir de los los valores de entrada de PORTA, para el grabado de acciones definimos al RA2 y hacemos uso de la instrucción FSR para el grabado de combinaciones definidas con la funcionalidad de cada combinación para RA0-RA1, así se guardan las combinaciones y el registro FSR como apuntador se incrementa con la instrucción INCF y se continúa preguntando si se quiere seguir grabando. Finalmente se pregunta si se quiere reproducir con el puerto RA3 y así vuelve a acceder a la memoria para que con el FSR e INCF recorra la lista de funciones guardadas y reproduzca las acciones guardadas.

```
PROCESSOR 16F877
INCLUDE <P16F877.INC>
cblock
    d1
    d2
    d3
    d4
    d5
    d6
endc
cblock 0x37
    count
    times
endc

ORG 0
GOTO INICIO
ORG 5
; DELAYS
; DELAY150
movlw    0x86
movwf    d1
movlw    0xA3
movwf    d2
movlw    0x02
movwf    d3

; DELAY4S
```

```

        movlw    0xB6
        movwf    d4
        movlw    0x99
        movwf    d5
        movlw    0x2C
        movwf    d6

;CARLOS CASTELAN RAMOS - PEDRO JAIR CORONA NAVA

INICIO:    CLRF PORTA ;Limpia PORTA
           CLRF PORTB ;Limpia PORTB

           ;CAMBIA A BANCO 1
           BSF STATUS, RP0
           BCF STATUS, RP1

           ;PORT A - ENTRADA
           MOVLW H'FF'
           MOVWF TRISA

           ;PORTB - SALIDA
           MOVLW H'0'
           MOVWF TRISB

           ;DEFINIR INICIO DE ESCRITURA DE MEMORIA FSR
           MOVLW 0X20
           MOVWF FSR ;FSR=20

           ;CAMBIA A BANCO 0
           BCF STATUS,5

;PREGUNTA SI QUIERE GRABAR
GRABAR:    CALL DEBOUNE1
           BTFSS PORTA, 2
           GOTO REPRODUCIR ;Reproduce lo que tenga guardado
           GOTO SELEC_G ;SI - LOOP DE SELECCION A GRABAR

SELEC_G:   MOVF PORTA,W      ;W<-- (PORTE)
           ANDLW 3           ;W <-- W&00000011
           ADDWF PCL,F       ;(PCL)<-- (PCL)+W
           GOTO G_DESP_IZQ ;PC+0    -> Switches: 00
           GOTO G_DESP_DER ;PC+1    -> Switches: 01

```

```

        GOTO G_ON_OFF ;PC+2 -> Switches: 10
        GOTO G_ON4   ;PC+3   -> Switches: 11

G_DESP_IZQ: MOVLW H'0' ;Carga 00 a W
            MOVWF INDF ;Mueve el valor de W a contenido de FSR
            INCF FSR   ;Incrementa posici n FSR
            GOTO REPRODUCIR

G_DESP_DER: MOVLW H'1' ;Carga 01 a W
            MOVWF INDF ;Mueve el valor de W a contenido de FSR
            INCF FSR   ;Incrementa posici n FSR
            GOTO REPRODUCIR

G_ON_OFF:   MOVLW H'2' ;Carga 10 a W
            MOVWF INDF ;Mueve el valor de W a contenido de FSR
            INCF FSR   ;Incrementa posici n FSR
            GOTO REPRODUCIR

G_ON4:      MOVLW H'3' ;Carga 11 a W
            MOVWF INDF ;Mueve el valor de W a contenido de FSR
            INCF FSR   ;Incrementa posici n FSR
            GOTO REPRODUCIR

;PREGUNTA SI QUIERE REPRODUCIR
REPRODUCIR: CALL DEBOUNE2
            BTFSS PORTA, 3
                GOTO GRABAR ; NO - Pregunta si quiere grabar de nuevo
                GOTO LEER_MEM; SI - Lee memoria de instrucciones
guardadas

;LECTURA DE MEMORIA
LEER_MEM:   MOVLW 0X20 ;Parte de la posici n 20
            GOTO LECTURA

LECTURA:   MOVF FSR, W
            ANDLW 3      ;W <-- W&00000011
            ADDWF PCL,F   ;(PCL)<-- (PCL)+W
            GOTO DESP_IZQ ;PC+0   -> Switches: 00
            GOTO DESP_DER ;PC+1   -> Switches: 01
            GOTO ON_OFF   ;PC+2   -> Switches: 10
            GOTO ON4      ;PC+3   -> Switches: 11

;DESPLAZAMIENTO A LA IZQUIERDA
DESP_IZQ:   BCF PORTB, 0 ; Apagamos el LED
            CALL DELAY150 ; Retardo de 150 ms
            RLF PORTB, F ; Desplaza a la izquierda
            BSF PORTB, 7 ; Enciende el LED

```

```

        CALL DELAY150 ; Retardo de 150 ms
        INCF FSR      ;Incrementa posición FSR
        GOTO LECTURA

;DESPLAZAMIENTO A LA DERECHA
DESP_DER:   BCF PORTB, 0 ; Apagamos el LED
            CALL DELAY150 ; Retardo de 150 ms
            RRF PORTB, F ; Desplazamiento a la derecha del valor en el
puerto B

            BSF PORTB, 7 ; Enciende el LED
            CALL DELAY150 ; Retardo de 150 ms
            INCF FSR      ;Incrementa posición FSR
            GOTO LECTURA

;ENCENDER Y APAGAR 8 VECES
ON_OFF:     CLRF PORTB ;Apaga todos los leds
            MOVLW .8      ; Establecer el número de iteraciones
            MOVWF times

LOOP:       MOVLW 0xFF    ; Encender todos los LEDs
            MOVWF PORTB
            CALL DELAY150 ; Esperar un momento
            ; Apagar todos los LEDs
            CLRF PORTB
            CALL DELAY150 ; Esperar un momento

            ; Decrementar el contador de iteraciones
            DECFSZ times,f
            GOTO LOOP    ; Volver al inicio del loop si no ha terminado

            ; Detener el programa
            GOTO $        ; Bucle infinito
            INCF FSR      ;Incrementa posición FSR
            GOTO LECTURA

;ENCENDER 4 SEGUNDOS
ON4:        CLRF PORTB; APAGAMOS LOS LEDS
            MOVLW H'FF' ;Pasamos puros 1 a w
            MOVWF PORTB; Encendemos los leds
            CALL DELAY4S
            INCF FSR      ;Incrementa posición FSR
            GOTO LECTURA

;DELAY150MS

```



```

DELAY150:    decfsz    d1, f
             goto      $+2
             decfsz    d2, f
             goto      $+2
             decfsz    d3, f
             goto      DELAY150

             ;1 cycle
             nop

;DELAY 4 SEG
DELAY4S:     decfsz    d3, f
             goto      $+2
             decfsz    d4, f
             goto      $+2
             decfsz    d5, f
             goto      DELAY4S

             ;1 cycle
             nop

;DEBOUNE PARA OSICLACION DE GRABAR
DEBOUNE1:    BTFSC     PORTA, 2
             GOTO      $-1
             CALL      DELAY150
             BTFSS     PORTA, 2
             GOTO      $-2
             CALL      DELAY150

;DEBOUNE PARA OSICLACION DE REPRODUCIR
DEBOUNE2:    BTFSC     PORTA, 3
             GOTO      $-1
             CALL      DELAY150
             BTFSS     PORTA, 3
             GOTO      $-2
             CALL      DELAY150

END

```

Conclusiones:

- Carlos Castelan Ramos: Para esta práctica, se logró comprender el uso y funcionamiento del set de instrucciones del PIC para el acceso a memoria y guardado de información haciendo uso de registros con la instrucción FSR y el incremento de la misma con apuntador con INCF así con diversas instrucciones condicionales y llamados a subrutinas se logró realizar el código con éxito.
- Corona Nava Pedro Jair: Logramos implementar las funcionalidades de guardado de registros con el uso de un apuntador para la aplicación de grabado y reproducción de funciones que anteriormente ya habíamos realizado.