Facultad de ingeniería

Materia: Laboratorio de Microcomputadoras

REPORTE DE LA PRÁCTICA 3

Título: Sistema mínimo microcontrolador PIC16F877

Integrantes:

- Martínez Pérez Brian Erik - 319049792
- Nuñez Rodas Abraham Enrique - 114003546
- Vicenteño Maldonado Jennifer Michel - 317207251

Profesor: Moises Melendez Reyes

Grupo: 1

Fecha de Entrega: 16 de marzo de 2025

Semestre: 2025-2

Objetivo: Desarrollar la habilidad de interpretación de esquemáticos. Conocer el diagrama del sistema mínimo del microcontrolador, el software de comunicación. Realizar aplicaciones con puertos paralelos en la modalidad de salida; ejecución de un programa en tiempo real.

Ejercicios

Ejercicio 1: Revisar a detalle y en concordancia con el circuito 3.2, identificar las conexiones faltantes, discutir con sus compañeros y con su profesor(a) el impacto y función de los mismos.

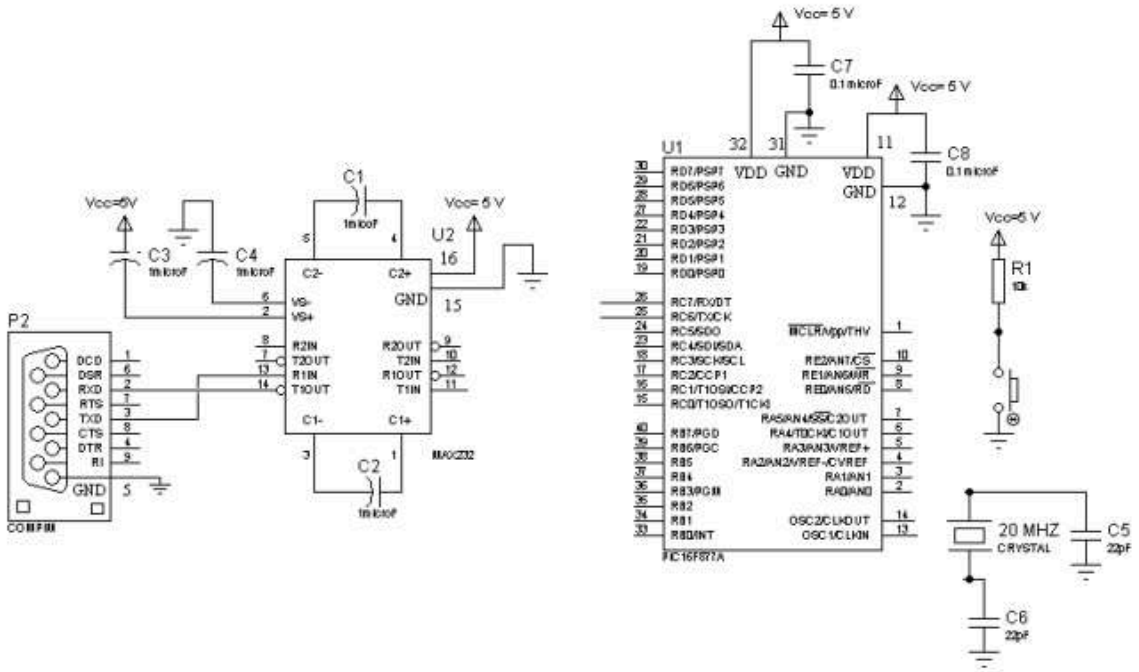
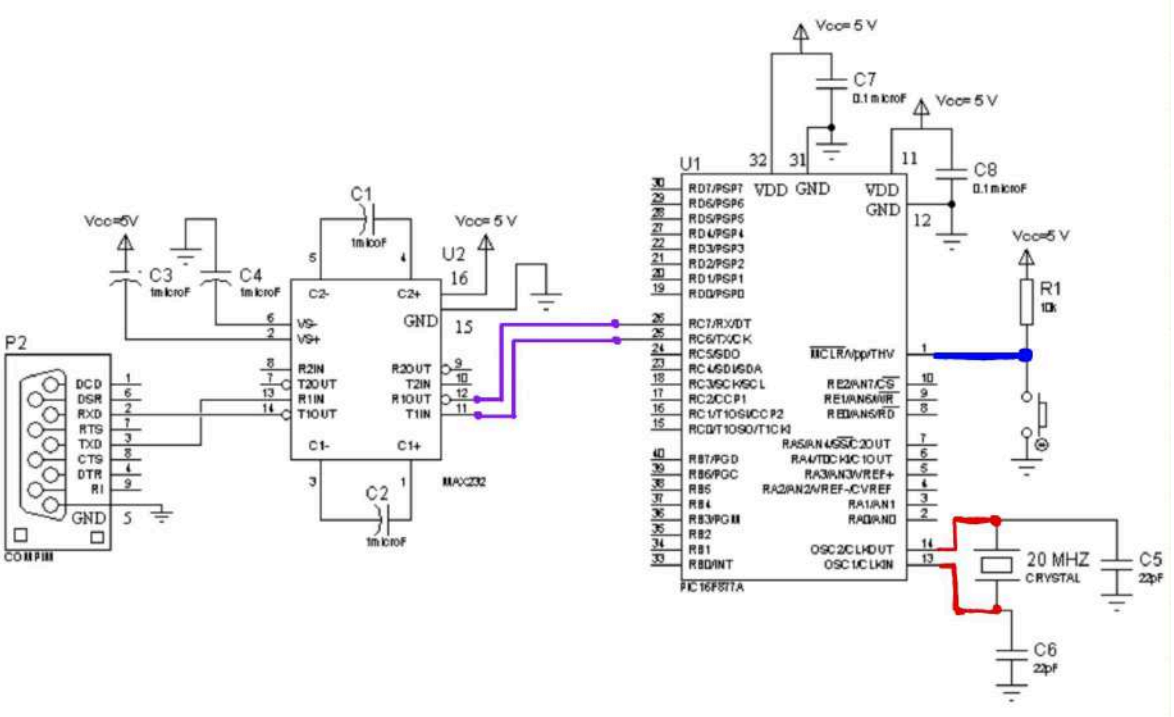


Figura 3.5 Circuito de prueba

ROJO: Cristal de cuarzo: El propósito es la generación de la frecuencia de operación externa.
AZUL: Botón reset: El propósito es la generación del pulso en bajo para producir un reset en el sistema.
MORADO: Convertidor USB a TTL, diseñado para convertir señales entre un puerto USB de una computadora y niveles de voltaje TTL.

Ejercicio 2: Completar las conexiones faltantes, utilizando jumpers; cerciorar el alambrado correcto.



Ejercicio 3: Una vez resueltas las actividades anteriores, identificar la terminal PB0 del puerto B, realizar la conexión con la salida de una resistencia y un led.

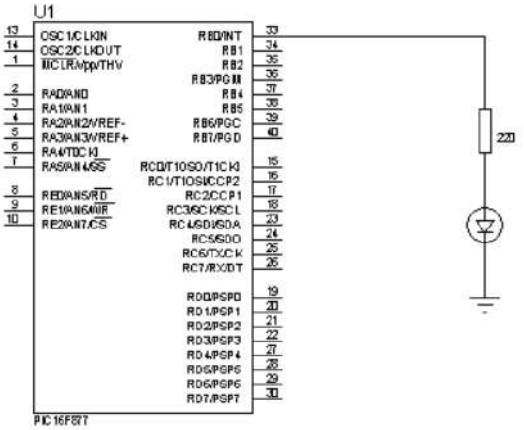
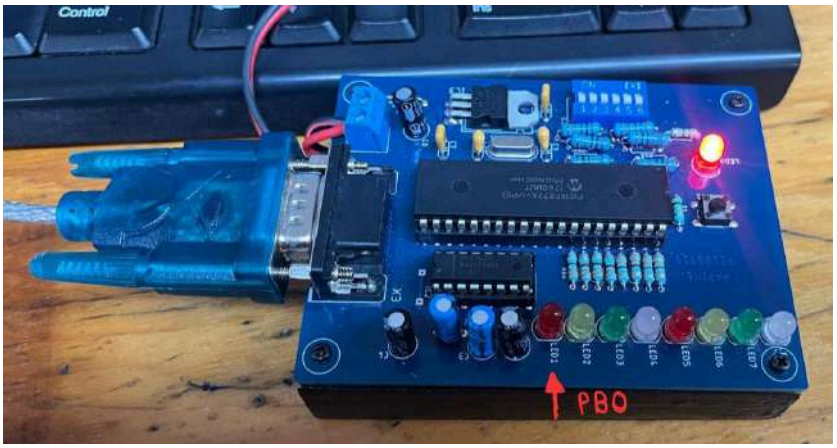


Figura 3.6 Circuito PB0



Ejercicio 4: Escribir, comentar e indicar que hace el siguiente programa.

```
processor 16f877
include <p16f877.inc>

valor1 equ h'21'
valor2 equ h'22'
valor3 equ h'23'
cte1 equ 20H
cte2 equ 50h
cte3 equ 60h

ORG 0
GOTO INICIO
ORG 5

INICIO:BSF STATUS,RP0
       BCF STATUS,RP1
       BCF STATUS, C
       MOVLW H'0'
       MOVWF TRISB
       BCF STATUS,RP0
       CLRF PORTB

loop2 MOVLW B'11111111'
       MOVWF PORTB
CALL retardo
       MOVLW B'00000000'
       MOVWF PORTB
CALL retardo
GOTO loop2

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
END
```

```
processor 16f877
include <p16f877.inc>

valor1 equ h'21'  ; Variable para retardo, almacenada en dirección 0x21
valor2 equ h'22'
valor3 equ h'23'


cte1 equ 20h  ; Constante para el retardo
cte2 equ 50h  ; Constante para el retardo
cte3 equ 60h  ; Constante para el retardo

ORG 0  ; Dirección de inicio del programa
GOTO INICIO
```

```
ORG 5  ; Ubicación del código principal
INICIO:
    BSF STATUS,RP0  ; Selecciona el banco de memoria 1
    BCF STATUS,RP1  ; Asegura que estamos en el banco correcto
    MOVLW H'0'  ; Carga el valor 0 en el registro W
    MOVWF TRISB     ; configuro los pines del puerto b en salidas
    BCF STATUS,RP0      ; regresa al banco 0 de ram
    CLRF PORTB      ; limpia el puerto b

loop2           ;loop principal, prohgrama ciclico
    BSF PORTB,0     ;PORTB.0 <-- 0, como tenemos un led, este programa se encendera
    CALL retardo  ; llama a una subrutina de retardo
    BCF PORTB,0 ; PORTB.0 <-- 0, apaga el led conectado al PORTB.0
    CALL retardo    ; llama a retardo
    GOTO loop2      ;regresa a loop2, loop infinito

retardo: ; subrutina para generar los retardos y visualizar el encendio del led
    MOVLW cte1
    MOVWF valor1

tres:
    MOVLW cte2  ; Carga la constante cte2 en W
    MOVWF valor2  ; Almacena W en valor2

dos:
    MOVLW cte3  ; Carga la constante cte3 en W
    MOVWF valor3  ; Almacena W en valor3

uno:
    DECFSZ valor3,F  ; Decrementa valor3 y salta si es cero
    GOTO uno  ; Si valor3 no es cero, repite
    DECFSZ valor2,F  ; Decrementa valor2 y salta si es cero
    GOTO dos  ; Si valor2 no es cero, repite
    DECFSZ valor1,F  ; Decrementa valor1 y salta si es cero
    GOTO tres  ; Si valor1 no es cero, repite
    RETURN  ; Regresa de la subrutina

END
```

Ejercicio 5: Ensamblar y cargar el programa anterior en el microcontrolador; que es lo que puede visualizar.

El código configura el puerto B del PIC16F877 como salida y hace parpadear el LED conectado al pin RB0. además utiliza una subrutina de retardo basada en decrementos de registros para controlar el tiempo entre encendidos y apagados del LED.

Ejercicio 6: En el programa, modifique el valor de cte1 a 8h, ensamblar y programar; que sucede y porqué?

Cuando cte1 es más grande, el retardo es mayor; cuando es más pequeño, el retardo disminuye. El efecto visible es que el LED parpadea más rápido porque los ciclos de encendido y apagado son más cortos.

Ejercicio 7: Modifique cte1 a 80h; ensamblar y programar, ¿existe algún cambió?

El efecto visible es que el LED parpadea más lento porque los ciclos de encendido y apagado son más grandes.

Ejercicio 8: - Modificar el programa anterior, para que ahora se actualice el contenido de todos los bits del puerto B y se genere una rutina de retardo de un segundo.
Este programa requiere de 8 salidas conectadas al puerto B, tal como se muestra en la figura.

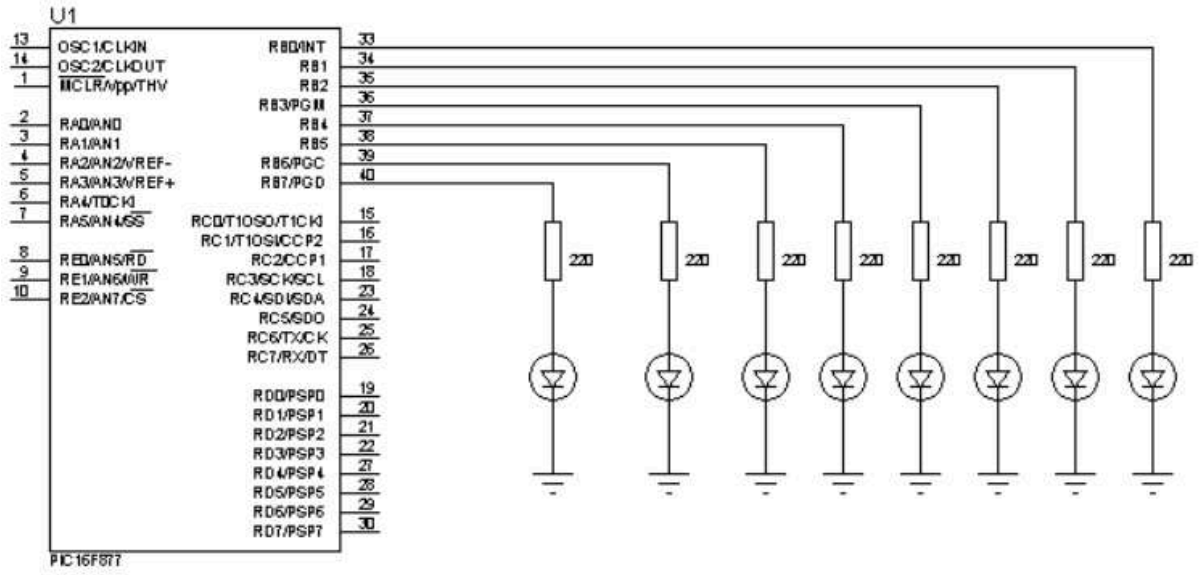
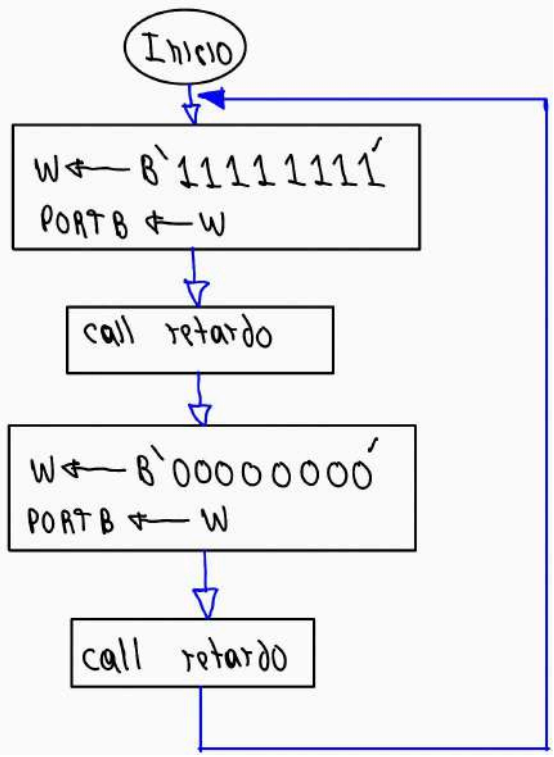


Figura 3.7 Conexión del sistema mínimo al módulo de 8 leds

Diagrama de flujo:



Funcionamiento de la solución:

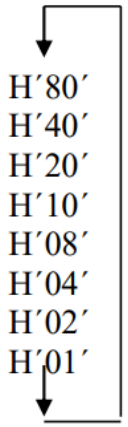
Código:

```
processor 16f877
include <p16f877.inc>

valor1 equ h'21'
valor2 equ h'22'
valor3 equ h'23'
cte1 equ 80H
cte2 equ 50h
cte3 equ 60h

ORG 0
GOTO INICIO
ORG 5

INICIO:BSF STATUS,RP0
```

```
        BCF STATUS,RP1
        BCF STATUS, C
        MOVLW H'0'
        MOVWF TRISB
        BCF STATUS,RP0
        CLRF PORTB

loop2 MOVLW B'11111111'
        MOVWF PORTB
CALL retardo
        MOVLW B'00000000'
        MOVWF PORTB
CALL retardo
GOTO loop2

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

END
```

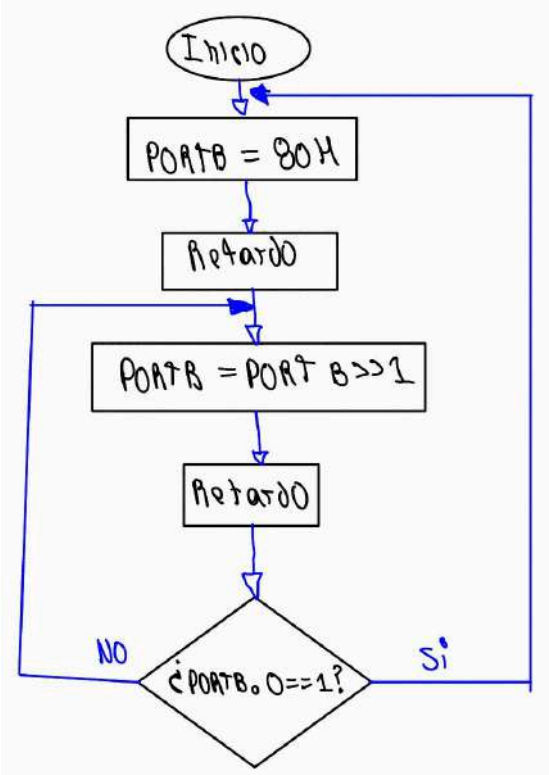
Análisis: *El objetivo del programa es hacer parpadear todos los pines del puerto B del microcontrolador PIC16F877 con un retardo controlado. La configuración del registro TRISB como salida y la manipulación del puerto B mediante la carga de valores en PORTB funcionan correctamente. El principal periférico utilizado es el puerto B (PORTB), que se configura como salida digital para el control de LEDs. Se observa que la modificación de cte1 afecta directamente la duración del retardo, lo que demuestra el correcto funcionamiento de la subrutina de retardo.*

Ejercicio 9: Realizar un programa que muestre la siguiente secuencia en el puerto B con retardos de ½ segundo. secuencia:

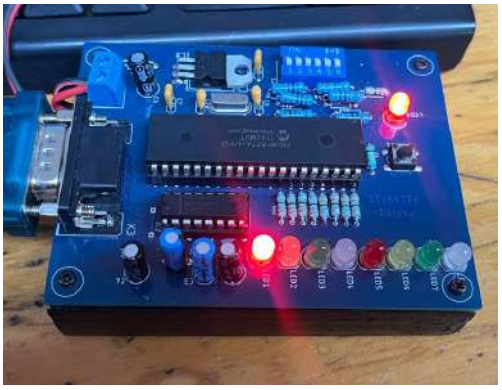


El circuito empleado es el mismo que en el ejercicio anterior.

Diagrama de flujo:



Funcionamiento de la solución:



Código:

```
processor 16f877
include <p16f877.inc>

; Definición de variables en memoria
valor1 equ h'21'
valor2 equ h'22'
valor3 equ h'23'
cte1 equ 80h  ; Constante para el retardo
cte2 equ 50h
cte3 equ 60h

ORG 0
GOTO INICIO  ; Ir al inicio del programa
ORG 5

INICIO:
   BSF STATUS,RP0  ; Cambiar al banco 1
   BCF STATUS,RP1
   BCF STATUS, C   ; Limpiar el bit de acarreo
   MOVLW H'0'
   MOVWF TRISB     ; Configurar PORTB como salida
   BCF STATUS,RP0  ; Volver al banco 0
   CLRF PORTB      ; Limpiar PORTB
   MOVLW 0x80
   MOVWF PORTB     ; Encender el bit más alto de PORTB
   CALL retardo    ; Llamar al retardo

loop2:
   RRF PORTB, 1    ; Rotar el valor de PORTB a la derecha
   CALL retardo    ; Llamar al retardo
   BTFSC STATUS, C ; Si el bit de acarreo está en 1, reiniciar
   GOTO INICIO
   CALL retardo    ; Otro retardo
   GOTO loop2      ; Repetir el ciclo

; Subrutina de retardo
retardo:
   MOVLW cte1
   MOVWF valor1    ; Cargar valor de retardo

tres:
   MOVLW cte2
   MOVWF valor2

dos:
```

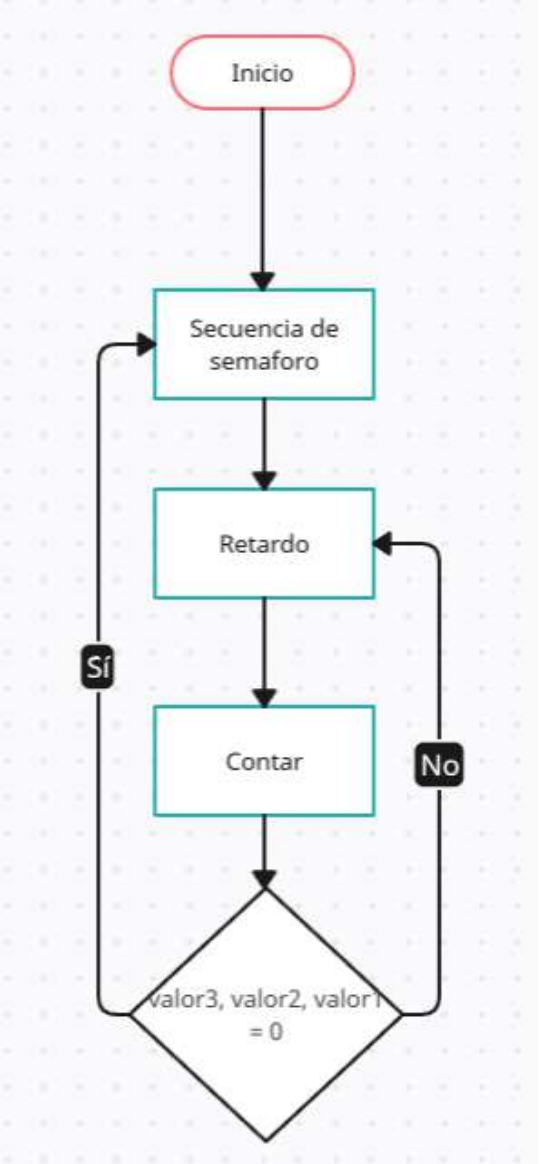
```
    MOVLW cte3
    MOVWF valor3

uno:
    DECFSZ valor3   ; Disminuir contador y verificar si es cero
    GOTO uno
    DECFSZ valor2
    GOTO dos
    DECFSZ valor1
    GOTO tres
    RETURN          ; Regresar de la subrutina

END
```

Análisis:
Primero asignamos al puerto b como salida, usamos RRF para rotar el puerto hacia la derecha e ir moviendo el led que va a ir encendiendo (del más significativo al menos), utiliza un retardo para que este cambio sea notorio, al llegar al último bit se activa la bandera C y se reinicia el programa.

Ejercicio 10: Realizar un programa que controle el funcionamiento de dos semáforos; cada estado tendrá una duración de 2 segundos.

Estado	Salida
1	V1, R2
2	A1, R2
3	R1, V2
4	R1, A2

Diagrama de flujo:



Funcionamiento de la solución:

Código:

```
processor 16f877
include <p16f877.inc>

; Definición de variables en memoria
valor1 equ h'21'
valor2 equ h'22'
valor3 equ h'23'
cte1 equ 0xFF  ; Constante para el retardo
cte2 equ 50h
cte3 equ 60h

ORG 0
GOTO INICIO   ; Ir al inicio del programa
```

```
ORG 5

INICIO:
   BSF STATUS,RP0   ; Cambiar al banco 1
   BCF STATUS,RP1
   BCF STATUS, C    ; Limpiar el bit de acarreo
   MOVLW H'0'
   MOVWF TRISB      ; Configurar PORTB como salida
   BCF STATUS,RP0   ; Volver al banco 0
   CLRF PORTB       ; Apagar todos los pines de PORTB

loop2:
   ; Encender los pines RB2 y RB4
   BSF PORTB, 2
   BSF PORTB, 4
   CALL retardo  ; Esperar un tiempo
   ; Apagar los pines RB2 y RB4
   BCF PORTB, 2
   BCF PORTB, 4

   ; Encender los pines RB1 y RB4
   BSF PORTB, 1
   BSF PORTB, 4
   CALL retardo
   ; Apagar los pines RB1 y RB4
   BCF PORTB, 1
   BCF PORTB, 4

   ; Encender los pines RB0 y RB6
   BSF PORTB, 0
   BSF PORTB, 6
   CALL retardo
   ; Apagar los pines RB0 y RB6
   BCF PORTB, 0
   BCF PORTB, 6

   ; Encender los pines RB0 y RB5
   BSF PORTB, 0
   BSF PORTB, 5
   CALL retardo
   ; Apagar los pines RB0 y RB5
   BCF PORTB, 0
   BCF PORTB, 5

   GOTO loop2  ; Repetir el ciclo

; Subrutina de retardo
```

```
retardo:
   MOVLW cte1
   MOVWF valor1   ; Cargar valor de retardo

tres:
   MOVLW cte2
   MOVWF valor2

dos:
   MOVLW cte3
   MOVWF valor3

uno:
   DECFSZ valor3   ; Disminuir contador y verificar si es cero
   GOTO uno
   DECFSZ valor2
   GOTO dos
   DECFSZ valor1
   GOTO tres
   RETURN          ; Regresar de la subrutina

END
```

Análisis:
Primero configuramos a Port B como salida, después vamos asignando la secuencia de apagado y encendido con BSF y BCF (instrucciones bit a bit, activar y borrar) de la siguiente forma:
rb2 y rb4 encendidos, retardo y apaga
rb1 y rb4, retardo y apaga
rb0 y rb6, retardo y apaga
rb0 y rb5, retado y apaga
Sigue repitiendo la secuencia cada que se activa c

Conclusiones:

Martínez Pérez Brian Erik

Para esta práctica interpretamos los elementos principales que componen el sistema mínimo del microcontrolador PIC16F877, como lo son el botón de reset, el cuarzo que funciona como reloj de frecuencia externa y el módulo para comunicación entre nuestra computadora y el PIC . Además en el laboratorio configuramos el puerto B para utilizarlo como salida, en este caso creamos códigos los cuales encendían y apagaban los led de acuerdo a lo solicitado en los problemas de la práctica.

Nuñez Rodas Abraham Enrique

Durante esta práctica, profundicé en la configuración y programación del microcontrolador PIC16F877, enfocándome en el manejo del puerto B como salida para controlar LEDs. La experiencia me permitió reforzar habilidades en la interpretación de esquemáticos y en la implementación de retardos mediante código ensamblador. Además, comprendí la importancia de ajustar constantes para modificar tiempos de retardo y cómo estas modificaciones afectan el comportamiento del hardware conectado. Este ejercicio práctico consolidó mi entendimiento sobre la interacción entre software y hardware en sistemas embebidos, destacando la relevancia de una programación precisa para el correcto funcionamiento de dispositivos electrónicos.

Vicenteño Maldonado Jennifer Michel
Durante está práctica utilizamos los retardos vistos en prácticas anteriores, para hacer más notorias la secuencias solicitadas, también modificamos estos para que cumplieran un tiempo específico. También utilizamos instrucciones como RRF para modificar todo el bloque de bits del puerto b y otras cómo BSF y BCF para modificar bits específicos del puerto b y asignarles la secuencia.
Aprendimos cuales son los elementos básicos para realizar un programa sencillo en el microcontrolador, como configurar las salidas, como conectarlo a la computadora y con el programa .

Referencias:

del PIC, 2. 1-La Familia. (s/f). 2.- Descripción General del PIC16F877. Edu.ar., de https://exa.unne.edu.ar/ingenieria/sysistemas/public_html/Archi_pdf/HojaDatos/Microcontrolador es/PIC16F877.pdf

(S/f). Newark.com. Recuperado de https://mexico.newark.com/microchip/pic16f877a-i-p/microcontroller-mcu-8-bitpic16/dp/69K7640?srsltid=AfmBOorWLTceQMTppGk0OMGmmjB6UpIiiw55U28F2qBZH2pUYET 3Elnu