



**Universidad Nacional Autónoma de
México
Facultad de Ingeniería**



Semestre 2023-2

Laboratorio de Microcomputadoras

Práctica 3:

Sistema mínimo de microcontrolador PIC16F877

Profesor:

M.I. Ruben Anaya García

Alumnos:

Barreiro Valdez Alejandro

Zepeda Baeza Jessica

Número de cuenta:

317520888

317520747

Grupo Laboratorio: 4

Grupo Teoría: 1

Fecha de realización: 14 de marzo de 2023

Fecha de entrega: 21 de marzo de 2023

Desarrollo

En esta práctica se entenderá cómo se utiliza el sistema mínimo del microcontrolador PIC16F877. Se podrá comprender cuáles son las partes de este sistema gracias al esquema donde se muestran sus partes. Además, se realizarán aplicaciones donde se tengan salidas que sean provenientes de un puerto paralelo. Para el caso del laboratorio, se utilizó un sistema mínimo que ya se encontraba armado y se probaron distintos programas en él. El flujo de trabajo para esta práctica consiste en generar un programa, ensamblarlo y cargarlo al sistema mínimo para probar su funcionamiento. Las actividades que se desarrollaron en esta práctica fueron:

1. Probar el programa que se tiene en el manual de prácticas. Su funcionamiento consistió en generar una señal de reloj en el bit menos significativo del Puerto B. El periodo de esta señal fue definido por tres ciclos anidados que decrementan un cierto valor. Es decir, cada vez que se decremente en uno el valor del ciclo más lento significa que el valor del ciclo intermedio ya decrementó hasta cero y de igual forma, para que decremente en uno el valor intermedio, el valor del ciclo rápido debe llegar a cero.

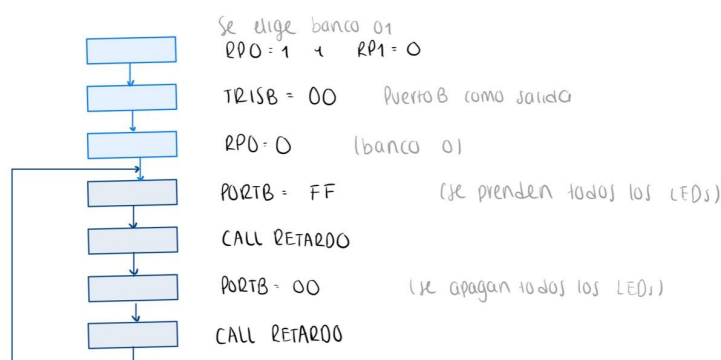
Por esta misma razón, el valor del ciclo lento es el que más influye en el tiempo de espera del código. Al hacerlo un número pequeño como 8h se tiene un periodo de señal muy pequeño, casi imperceptible. Al convertirlo en 80h, se tiene una espera de aproximadamente medio segundo.

2. Con base en el programa del manual, generar un programa donde se prendan todos los LEDs y se apaguen con un retraso de un segundo. (Ejercicio 1)
3. Un programa que realice una secuencia de las potencias de dos en el puerto de B con retardo de medio segundo entre cada una. (Ejercicio 2)
4. Un programa que simule el funcionamiento de un semáforo con los estados que se piden en el manual de prácticas por dos segundos cada uno. (Ejercicio 3)

Algoritmos

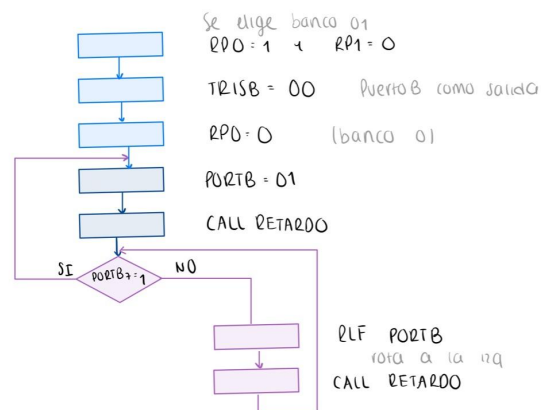
Ejercicio 1

Para el Ejercicio 1 se realizó la siguiente carta ASM:



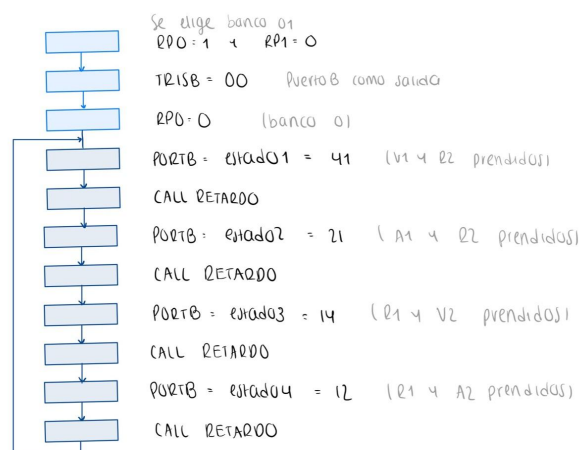
El ciclo comienza prendiendo todos los bits del Puerto B al tomar el valor FF. Después se llama la subrutina Retardo para esperar un segundo. Al finalizar la subrutina, se apagan todos los bits del Puerto B al tomar el valor 00 y se vuelve a llamar Retardo para esperar otro segundo. Se repite el ciclo.

Para el Ejercicio 2 se realizó la siguiente carta ASM:



El ciclo comienza asignándole el valor 01 al Puerto B (se prende el primer LED) y se espera medio segundo. Después verifica si el bit más significativo del Puerto B está prendido. En caso de que lo esté, significa que la rotación debe volver a comenzar desde el primer LED por lo que se regresa al inicio del ciclo. En caso de que aún no se prenda el último bit o LED, se hace la rotación a la izquierda y se espera medio segundo.

Para el Ejercicio 3 se realizó la siguiente carta ASM:



De igual forma, se coloca en el banco 01 para cambiar el valor de TRISB a 0 y definir el Puerto B como salida. Luego se regresa al banco 00.

En este algoritmo el Puerto B toma el valor de cada estado y llama a la subrutina Retardo para esperar dos segundos entre ellos. De manera que una vez que termina el cuarto estado se regresa al primero.

Programas comentados

Ejercicio 1

```
processor pl6f877
include <pl6f877.inc>

valor1 equ h'21'      ;Valor 1 para el retardo
valor2 equ h'22'      ;Valor 2 para el retardo
valor3 equ h'23'      ;Valor 3 para el retardo
ctel equ 80h          ;constante 1 de lo que se tarda el retardo
cte2 equ 50h          ;constante 2 de lo que se tarda el retardo
cte3 equ 60h          ;constante 3 de lo que se tarda el retardo

ORG 0
GOTO INICIO
ORG 5
INICIO:
    BSF STATUS,RP0
    BCF STATUS,RP1    ;Se selecciona el banco 01
    MOVLW H'0'
    MOVWF TRISB        ;portb = salida
    BCF STATUS,RP0    ;Se selecciona el banco 00
    CLRF PORTB        ;El valor de portb es cero
loop2:
    MOVLW 0xFF
    MOVWF PORTB        ;portb = FF = 11111111
    CALL retardo        ;Se prenden los LEDs y se llama al retardo
    CLRF PORTB        ;Se apagan los LEDs
    CALL retardo        ;Se llama al retardo
    GOTO loop2        ;Se repite el ciclo

retardo:
    ;Se inicia el proceso de retardo
    MOVLW cte1
    MOVWF valor1        ;valor1 = cte1
tres:
    MOVLW cte2
    MOVWF valor2        ;valor2 = cte2
dos:
    MOVLW cte3
    MOVWF valor3        ;valor3 = cte3
uno:
    DECFSZ valor3        ;Decrementa valor3 hasta que sea 0
    GOTO uno
    DECFSZ valor2        ;Decrementa valor2 hasta que sea 0
    GOTO dos
    DECFSZ valor1        ;Decrementa valor1 hasta que sea 0
    GOTO tres
    RETURN                ;Se acaba el retardo
END
```

Ejercicio 2

```

processor p16f877
include <p16f877.inc>

valor1 equ h'21'      ;Valor 1 para el retardo
valor2 equ h'22'      ;Valor 2 para el retardo
valor3 equ h'23'      ;Valor 3 para el retardo
cte1 equ 80h          ;constante 1 para lo que se tarda el retardo
cte2 equ 50h          ;constante 2 para lo que se tarda el retardo
cte3 equ 60h          ;constante 3 para lo que se tarda el retardo

ORG 0
GOTO INICIO
ORG 5
INICIO:
    BSF STATUS,RP0
    BCF STATUS,RP1    ;Se selecciona el banco 01
    MOVLW H'0'
    MOVWF TRISE        ;portb = salida
    BCF STATUS,RP0    ;Se selecciona el banco 00
    CLRF PORTB        ;El valor de portb es cero
loop2:
    MOVLW 0x01        ;Se coloca 01 en W
    MOVWF PORTE        ;Se mueve el valor de W a PORTE
    CALL retardo       ;Se llama el retardo
ROTA:
    BTFSS PORTB,7      ;Se revisa el valor del bit 7 de M, si es 1 hay un salto
    GOTO SIGUE         ;Si el bit 7 de M es 0, salta a la etiqueta sigue
    GOTO loop2         ;Si el bit 7 de M es 1, salta al inicio del programa
SIGUE:
    RLF PORTB         ;Rotan los bits en M hacia la izquierda usando el carry
    CALL retardo       ;Se llama el retardo
    GOTO ROTA         ;Salta a la etiqueta rota

retardo:              ;Instrucciones para generar el retardo
    MOVLW cte1
    MOVWF valor1      ;valor1 = cte1
tres:
    MOVLW cte2
    MOVWF valor2      ;valor2 = cte2
dos:
    MOVLW cte3
    MOVWF valor3      ;valor3 = cte3
uno:
    DECFSZ valor3      ;Decrementa hasta que valor 3 sea cero
    GOTO uno
    DECFSZ valor2      ;Decrementa hasta que valor 2 sea cero
    GOTO dos
    DECFSZ valor1      ;Decrementa hasta que valor 1 sea cero
    GOTO tres
    RETURN            ;Se acaba el retardo
END

```

Ejercicio 3

```

processor pl6f877
include <pl6f877.inc>

valor1 equ h'21'      ;Valor 1 para el retardo
valor2 equ h'22'      ;Valor 2 para el retardo
valor3 equ h'23'      ;Valor 3 para el retardo
cte1 equ 0xFF         ;constante 1 para tiempo del retardo
cte2 equ 0x90         ;constante 2 para tiempo del retardo
cte3 equ 60h         ;constante 3 para tiempo del retardo
estado1 equ 41h       ;Valor para prender Verdel y Rojo2
estado2 equ 21h       ;Valor para prender Amarillo1 y Rojo2
estado3 equ 14h       ;Valor para prender Rojo1 y Verded2
estado4 equ 12h       ;Valor para prender Rojo1 y Amarillo2

ORG 0
GOTO INICIO
ORG 5
INICIO:
    BSF STATUS,RP0
    BCF STATUS,RP1    ;Se selecciona el banco 01
    MOVLW H'0'
    MOVWF TRISB       ;portb = salida
    BCF STATUS,RP0    ;Se selecciona el banco 00
    CLRF PORTB        ;portb = 00
loop2:
    MOVLW estado1
    MOVWF PORTB        ;portb = estado1 = V1 y R2 prendidos
    CALL retardo       ;Se llama al retardo
    MOVLW estado2
    MOVWF PORTB        ;portb = estado2 = A1 y R2 prendidos
    CALL retardo       ;Se llama al retardo
    MOVLW estado3
    MOVWF PORTB        ;portb = estado3 = R1 y V2
    CALL retardo       ;Se llama al retardo
    MOVLW estado4
    MOVWF PORTB        ;portb = estado4 = R1 y A2
    CALL retardo       ;Se llama el retardo
    GOTO loop2         ;Regreso al primer estado
retardo:
    MOVLW cte1
    MOVWF valor1       ;valor1 = cte1
tres:
    MOVLW cte2
    MOVWF valor2       ;valor2 = cte2
dos:
    MOVLW cte3
    MOVWF valor3       ;valor3 = cte3
uno:
    DECFSZ valor3      ;Decrementa hasta que valor3 = 0
    GOTO uno
    DECFSZ valor2      ;Decrementa hasta que valor2 = 0
    GOTO dos
    DECFSZ valor1      ;Decrementa hasta que valor1 = 0
    GOTO tres
    RETURN             ;Regresar
END

```

Conclusiones y/o comentarios

Zepeda Baeza Jessica:

Considero que el objetivo de esta práctica fue enfocarse en el funcionamiento y composición del sistema mínimo del microcontrolador PIC16F877. En un principio se observó tanto en diagramas como físicamente, gracias a los microcontroladores ya ensamblados, la conexión del reloj, el reset y el voltaje al sistema mínimo así como también

las conexiones con el bootloader que hacen posible tener un programador en circuito. Por otra parte, se observaron los distintos puertos que componen al microcontrolador así como sus tamaños y en este caso, sus conexiones a LEDs o Switches. Los programas realizados fueron sencillos debido a que consistían principalmente en asignar valores al Puerto B y observar las salidas, esto porque considero que su objetivo realmente era trabajar con los puertos y con su configuración para tratarlos como entradas o salidas. Por último, se analizó en el código proporcionado la forma de hacer esperar cierto tiempo a un programa mediante contadores dentro de ciclos anidados y utilizar este bloque como una subrutina que se puede llamar varias veces a lo largo del programa.

Barreiro Valdez Alejandro:

En esta práctica se pudo utilizar el sistema mínimo construido a partir del microcontrolador PIC16F877A para resolver problemas en el ensamblador y observarlos en el sistema. Se codificó la resolución a los problemas, se compiló el proyecto y se cargó en la tarjeta. Uno de los mayores aprendizajes de esta práctica es la utilización de puertos como entrada o salida dentro del sistema. Para los ejercicios de esta práctica se utilizó el puerto B como salida de los programas. En el sistema mínimo se observaban los LEDs prendiendo dependiendo del valor que se indicara en el programa. Esto no hubiera sido perceptible sin instrucciones para un retardo que permitiera generar un retraso de medio segundo o un segundo entre cada instrucción. Se programaron soluciones a problemas que se habían tenido en prácticas pasadas como la rotación de bits y otros que sirvieron como introducción para el sistema mínimo. Utilizando TRISB se pudo definir que este puerto se utilizaría solo para salidas, pero en prácticas posteriores se podrá observar cómo los puertos también podrán ser utilizados como entradas.