UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

MATERIA

- Laboratorio de Microcomputadoras
- Grupo:04

PRÁCTICA 03
Sistema mínimo microcontrolador PIC16F877

PROFESOR

- M.I. Ruben Anaya García

ALUMNOS

- Carreón Guzmán Mariana Ivette
  - Núm. Cta.: 312103914
    - Gpo. Teoría: 04
  - Rojas Méndez Gabriel
  - Núm. Cta.: 314141712

SEMESTRE 2022-1

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.

Introducción

El sistema requiere de 3 modulos que son imprescindibles estos son:

- Reloj; formado por un cristal de cuarzo de 20 MHz y dos capacitores de 22 ρF, cuyo objetivo es la generación de la frecuencia de operación externa.
- Circuito de reset; formado por una resistencia y un push buttom; cuya finalidad es la generación del pulso en bajo para producir un reset en el sistema.
- La alimentación al sistema; Vdd= 5 V y GND= 0V

El programa debe ser descargado al dispositivo empleando un programador externo. Con la finalidad no depender de la existencia del programador externo y tener la ventaja de tener un programador en circuito de debe agregar:

a. Circuito que permita la comunicación serie asíncrona.
b. Programar con antelación el bootloader al procesador

Desarrollo

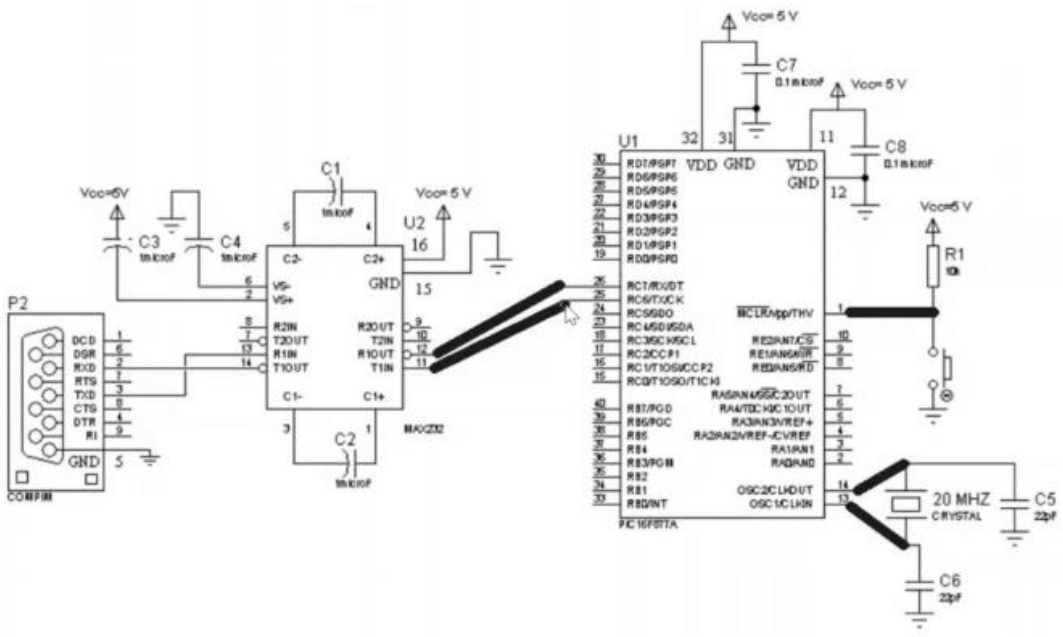
Ejercicio 1

Revisar a detalle y en concordancia con el circuito 3.2, Identificar las conexiones faltantes, discutir con sus compañeros y con su profesor el impacto y funcionamiento de los mismos.

Se debe de tener conectado el reloj para poder contar con la frecuencia de oscilación, en este caso podemos observar que el reset no se encuentra conectado y al no contar con la conexión con el MAX 232 no es posible descargar los programas.

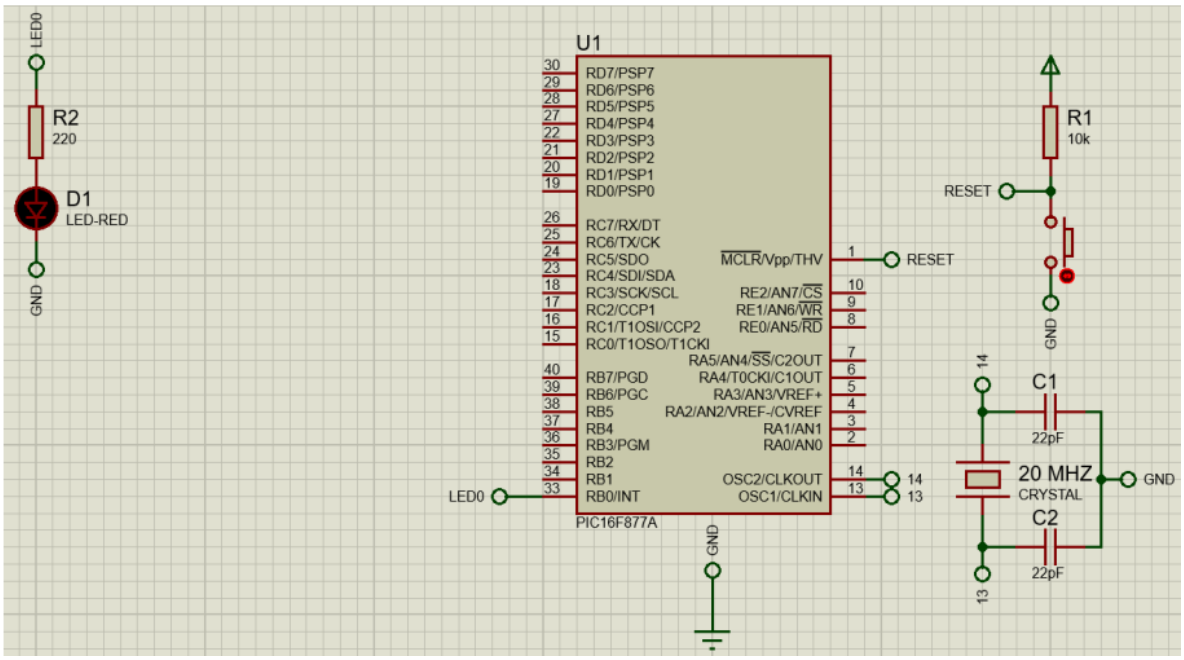
Ejercicio 2

Completar las conexiones faltantes, utilizando jumpers



Ejercicio 3

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



Ejercicio 4

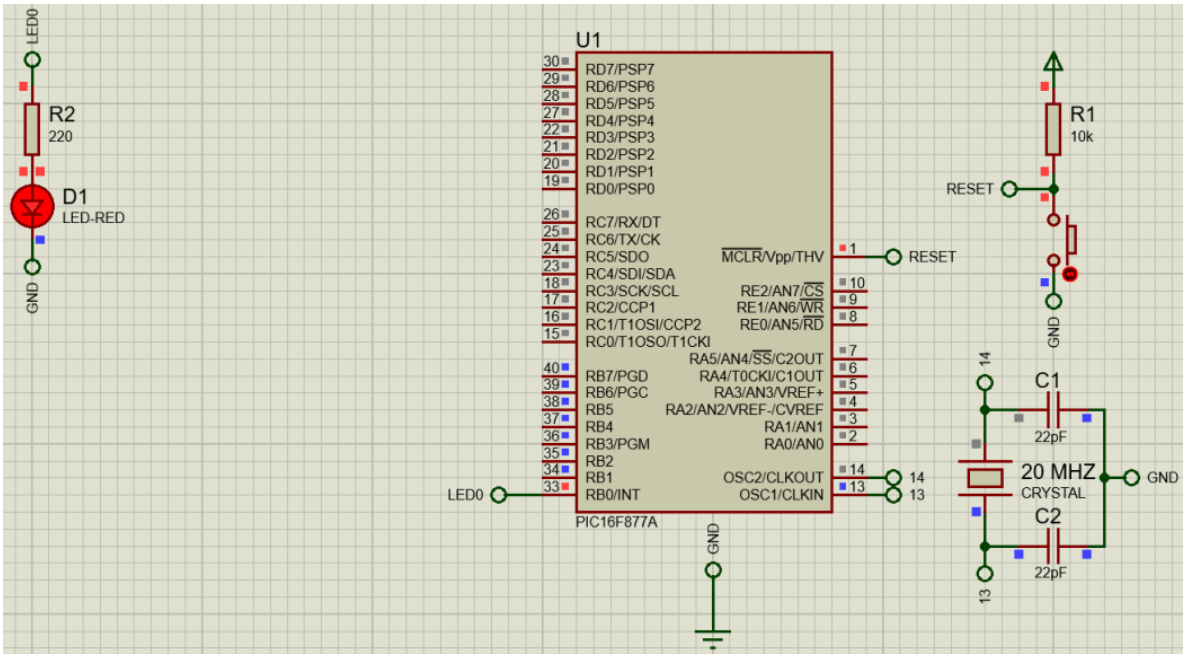
Escribir, comentar e indicar qué hace el siguiente programa

```
        processor 16f877       ;Indica la versión de procesador
        include <p16f877.inc>  ;Incluye la librería de la versión del procesador

valor1  equ h'21'          ;Asigna el valor de 21 a valor1
valor2  equ h'22'          ;Asigna el valor de 22 a valor2
valor3  equ h'23'          ;Asigna el valor de 23 a valor3
cte1    equ 20h            ;asigna el valor de 20 a cte1
cte2    equ 50h            ;asigna el valor de 50 a cte2
cte3    equ 60h            ;asigna el valor de 60 a cte3

        ORG 0              ;Especifica un origen (vector de reset)
        GOTO INICIO        ;Código del programa

        ORG 5              ;Indica origen para inicio del programa
INICIO: BSF STATUS, RP0    ;Cambia al banco 0
        BCF STATUS, RP1    ;Regresa al banco 1
        MOVLW H'0'         ;Guarda en el registro W el valor 0
        MOVWF TRISB        ;Mueve el contenido del registro W al registro TRISB
        BCF STATUS, RP0    ;Pone en 0 el bit de RP0 del registro status
        CLRF PORTB         ;Borra el contenido del registro PORTB

LOOP2:  BSF PORTB, 0       ;Pone en 1 el bit 0 del registro PORTB
        CALL RETARDO       ;Llama a la subrutina RETARDO
        BCF PORTB, 0       ;Pone en 0 el bit 0 del registo PORTB
        CALL RETARDO       ;Llama a la subrutina RETARDO
        GOTO LOOP2         ;Salta a LOOP2

RETARDO: MOVLW cte1        ;Guarda en el registo W el valor de cte1
        MOVWF valor1       ;Mueve el contenido del registro W al registro valor1

TRES:   MOVLW cte2         ;Guarda en el registo W el valor de cte2
        MOVWF valor2       ;Mueve el contenido del registro W al registro valor2

DOS:    MOVLW cte3         ;Guarda en el registo W el valor de cte3
        MOVWF valor3       ;Mueve el contenido del registro W al registro valor3

UNO:    DECFSZ valor3      ;Decrementa el registro valor3 hasta cero
        GOTO UNO           ;Salta a UNO
        DECFSZ valor2      ;Decrementa el registro valor2 hasta cero
        GOTO DOS           ;Salta a DOS
        DECFSZ valor1      ;Decrementa el registro valor1 hasta cero
        GOTO TRES          ;Salta a TRES
        RETURN             ;Regresa de la subrutina
        END                ;Directiva de fin de programa
```

Ejercicio 5

Ensamblar y cargar el programa anterior en el microcontrolador, qué es lo que puede visualizar.

Al momento de cargar el código se pudo observar que el LED se encendía cada 120 [ms]

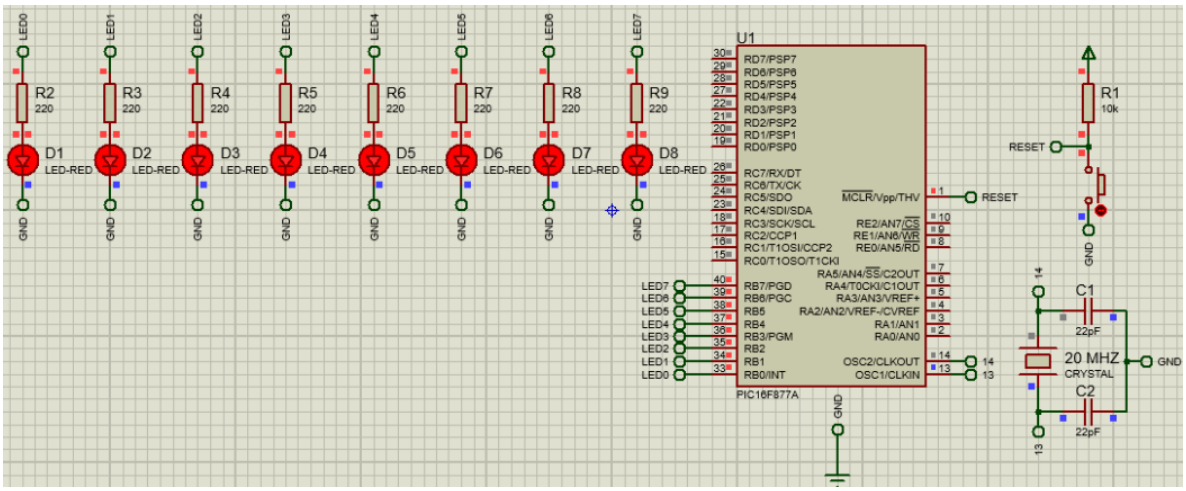


Ejercicio 6

En el programa, modifique el valor de cte1 a 8h, ensamblar y programar; ¿Qué sucede y por qué?

```
        processor 16f877       ;Indica la versión de procesador
        include <p16f877.inc>  ;Incluye la librería de la versión del procesador

valor1  equ h'21'          ;Asigna el valor de 21 a valor1
valor2  equ h'22'          ;Asigna el valor de 22 a valor2
valor3  equ h'23'          ;Asigna el valor de 23 a valor3
ctel    equ 8h         ;asigna el valor de 20 a ctel
cte2    equ 50h            ;asigna el valor de 50 a cte2
cte3    equ 60h            ;asigna el valor de 60 a cte3

        ORG 0              ;Especifica un origen (vector de reset)
        GOTC INICIO        ;Código del programa

        ORG 5              ;Indica origen para inicio del programa
INICIO: BSF STATUS, RP0    ;Cambia al banco 0
        BCF STATUS, RP1    ;Regresa al banco 1
        MOVLW H'0'         ;Guarda en el registro W el valor 0
        MOVWF TRISB        ;Mueve el contenido del registro W al registro TRISB
        BCF STATUS, RP0    ;Pone en 0 el bit de RP0 del registro status
        CLRF PORTB         ;Borra el contenido del registro PORTB

LOOP2:  BSF PORTB, 0       ;Pone en 1 el bit 0 del registro PORTB
        BSF PORTB, 1       ;Pone en 1 el bit 1 del registro PORTB
        BSF PORTB, 2       ;Pone en 1 el bit 2 del registro PORTB
        BSF PORTB, 3       ;Pone en 1 el bit 3 del registro PORTB
        BSF PORTB, 4       ;Pone en 1 el bit 4 del registro PORTB
        BSF PORTB, 5       ;Pone en 1 el bit 5 del registro PORTB
        BSF PORTB, 6       ;Pone en 1 el bit 6 del registro PORTB
        BSF PORTB, 7       ;Pone en 1 el bit 7 del registro PORTB
        CALL RETARDO       ;Llama a la subrutina RETARDO
        BCF PORTB, 0       ;Pone en 0 el bit 0 del registo PORTB
        BCF PORTB, 1       ;Pone en 0 el bit 1 del registo PORTB
        BCF PORTB, 2       ;Pone en 0 el bit 2 del registo PORTB
        BCF PORTB, 3       ;Pone en 0 el bit 3 del registo PORTB
        BCF PORTB, 4       ;Pone en 0 el bit 4 del registo PORTB
        BCF PORTB, 5       ;Pone en 0 el bit 5 del registo PORTB
        BCF PORTB, 6       ;Pone en 0 el bit 6 del registo PORTB
        BCF PORTB, 7       ;Pone en 0 el bit 7 del registo PORTB
        CALL RETARDO       ;Llama a la subrutina RETARDO
        GOTC LOOP2         ;Salta a LOOP2

RETARDO: MOVLW ctel          ;Guarda en el registo W el valor de ctel
         MOVWF valor1        ;Mueve el contenido del registro W al registro valor1

TRES:    MOVLW cte2          ;Guarda en el registo W el valor de cte2
         MOVWF valor2        ;Mueve el contenido del registro W al registro valor2

DOS:     MOVLW cte3          ;Guarda en el registo W el valor de cte3
         MOVWF valor3        ;Mueve el contenido del registro W al registro valor3

UNO:     DECFSZ valor3       ;Decrementa el registro valor3 hasta cero
         GOTC UNO            ;Salta a UNO
         DECFSZ valor2       ;Decrementa el registro valor2 hasta cero
         GOTC DOS            ;Salta a DOS
         DECFSZ valor1       ;Decrementa el registro valor1 hasta cero
         GOTC TRES           ;Salta a TRES
         RETURN              ;Regresa de la subrutina
         END                 ;Directiva de fin de programa
```

En este caso tenemos más LEDs y la frecuencia en la que se encienden y apagan es más rápida esto es debido a que el tiempo se ve reducido.

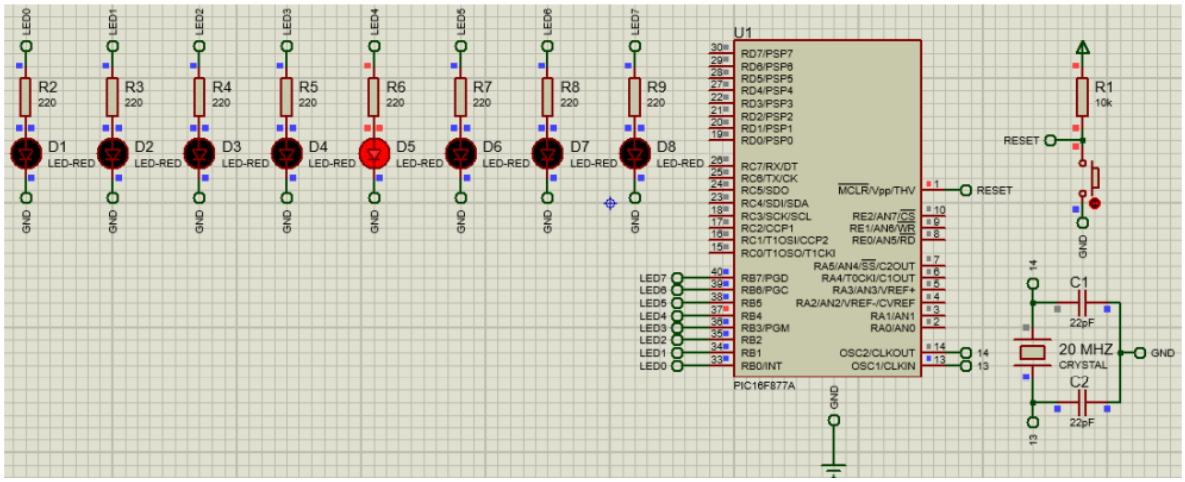


Ejercicio 7

Modifique cte1 a 80h; ensamblar y programar, ¿Existe algún cambio?
Podemos observar que existe un cambio en el tiempo de encendido y apagado, al hacer el cambio del valo cte1 este se vuelve mucho más lento.

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.



```
        processor 16f877        ;Indica la versión de procesador
        include <p16f877.inc>   ;Incluye la librería de la versión del procesador

valor1  equ h'21'           ;Asigna el valor de 21 a valor1
valor2  equ h'22'           ;Asigna el valor de 22 a valor2
valor3  equ h'23'           ;Asigna el valor de 23 a valor3
cte1    equ 10h             ;Asigna el valor de 20 a cte1
cte2    equ 50h             ;Asigna el valor de 40 a cte2
cte3    equ 60h             ;Asigna el valor de 20 a cte3

        ORG 0               ;Especifica un origen (vector de reset)
        GOTO INICIO         ;Código del programa

        ORG 5               ;Indica origen para inicio del programa
INICIO: BSF STATUS, RP0     ;Cambia al banco 0
        BCF STATUS, RP1     ;Regresa al banco 1
        MOVLW H'0'          ;Guarda en el registro W el valor 0
        MOVWF TRISB         ;Mueve el contenido del registro W al registro TRISB
        BCF STATUS, RP0     ;Pone en 0 el bit de RP0 del registro status
        CLRF PORTB          ;Borra el contenido del registro PORTB

LOOP2:  MOVLW H'80'         ;Guarda en el registro W el valor 80
        MOVWF H'20'         ;Mueve el valor de W a la localidad 20

 LOOP3:   MOVFW H'20'       ;Mueve el valor de W a la localidad
          MOVWF PORTB       ;Mueve el contenido de W a PORTB
          RRF H'20',1       ;Rotación a la derecha de la localidad H'20'
          CALL RETARDO      ;Llamada a subrutina retardo
          BTFSS H'20',0     ;Comprueba si lo que hay en H'20' es igual a 0H
          GOTO LOOP3        ;Salta a LOOP3
          GOTO LOOP2        ;Salta a LOOP2
```

Ejercicio 9

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

```
        processor 16f877       ;Indica la versión de procesador
        include <p16f877.inc>  ;Incluye la librería de la versión del procesador

valor1  equ h'21'          ;Asigna el valor de 21 a valor1
valor2  equ h'22'          ;Asigna el valor de 22 a valor2
valor3  equ h'23'          ;Asigna el valor de 23 a valor3
cte1    equ 20h            ;asigna el valor de 20 a cte1
cte2    equ 50h            ;asigna el valor de 50 a cte2
cte3    equ 60h            ;asigna el valor de 60 a cte3

        ORG 0              ;Especifica un origen (vector de reset)
        GOTO INICIO        ;Código del programa

        ORG 5              ;Indica origen para inicio del programa
INICIO: BSF STATUS, RP0    ;Cambia al banco 0
        BCF STATUS, RP1    ;Regresa al banco 1
        MOVLW H'0'         ;Guarda en el registro W el valor 0
        MOVWF TRISB        ;Mueve el contenido del registro W al registro TRISB
        BCF STATUS, RP0    ;Pone en 0 el bit de RP0 del registro status
        CLRF PORTB         ;Borra el contenido del registro PORTB

LOOP2:  BSF PORTB, 0       ;Pone en 1 el bit 0 del registro PORTB
        CALL RETARDO       ;Llama a la subrutina RETARDO
        BCF PORTB, 0       ;Pone en 0 el bit 0 del registo PORTB
        CALL RETARDO       ;Llama a la subrutina RETARDO
        GOTO LOOP2         ;Salta a LOOP2

RETARDO: MOVLW cte1        ;Guarda en el registo W el valor de cte1
        MOVWF valor1       ;Mueve el contenido del registro W al registro valor1

TRES:   MOVLW cte2         ;Guarda en el registo W el valor de cte2
        MOVWF valor2       ;Mueve el contenido del registro W al registro valor2

DOS:    MOVLW cte3         ;Guarda en el registo W el valor de cte3
        MOVWF valor3       ;Mueve el contenido del registro W al registro valor3

UNO:    DECFSZ valor3      ;Decrementa el registro valor3 hasta cero
        GOTO UNO           ;Salta a UNO
        DECFSZ valor2      ;Decrementa el registro valor2 hasta cero
        GOTO DOS           ;Salta a DOS
        DECFSZ valor1      ;Decrementa el registro valor1 hasta cero
        GOTO TRES          ;Salta a TRES
        RETURN             ;Regresa de la subrutina
        END                ;Directiva de fin de programa
```

En este caso usamos un retardo llamando a cte, con ella fuimos manejando el tiempo.

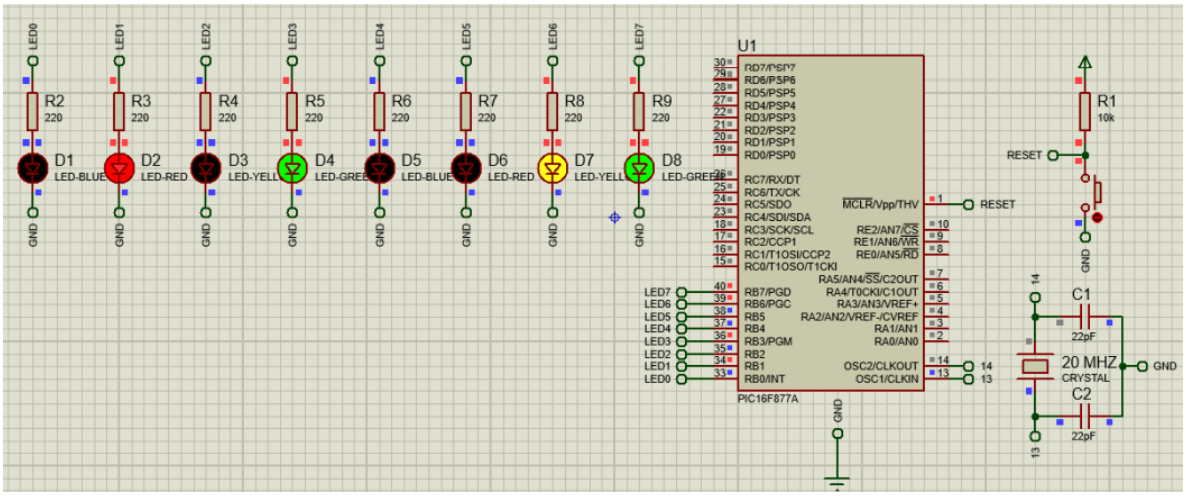
Ejercicio 10

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

En este caso se nos pidió generar 2 semáforos, como sabemos debe de haber un cambio de un estado a otro, esto es logrado gracias a los retardos, como vimos en lo ejercicios previos se puede controlar con ayuda de la cte1. El código generado queda de la siguiente manera:

```
        PROCESSOR 16F877
        INCLUDE <P16F877.INC>

        VALOR1 EQU H'21'
        VALOR2 EQU H'22'
        VALOR3 EQU H'23'
        CTE1   EQU H'50'
        CTE2   EQU H'50'
        CTE3   EQU H'60'
        STATE1 EQU H'40'
        STATE2 EQU H'41'

        ORG 0
        GOTO INICIO

        ORG 5
INICIO: BSF STATUS, RP0
        BCF STATUS, RP1
        MOVLW H'00'
        MOVWF TRISB
        BCF STATUS, RP0
        CLRF PORTB
        MOVLW H'9C'
        MOVWF STATE1
        MOVLW H'AC'
        MOVWF STATE2

LOOP:   MOVF STATE1, 0
        MOVWF PORTB
        CALL RETARDO
        MOVF STATE2, 0
        MOVWF PORTB
        CALL RETARDO
        GOTO SWAP
```

```
SWAP:    SWAPF STATE1
         SWAPF STATE2
         GOTO LOOP

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;
```



Conclusiones

Carreón Guzmán Mariana Ivette

En esta práctica pudimos seguir aprendiendo respecto a la forma de trabajar con Proteus y con un microcontrolador, aprendimos a interpreta los esquemáticos y de esta forma poder analizar si en

dado caso llegara a haber una conexión faltante. En esta práctica pude ver que ya tengo un mayo entendimiento de el lenguaje ensamblador ya que resultó más sencillo la elaboración del código

Rojas Méndez Gabriel

En la práctica aprendimos un poco más respecto al lenguaje ensamblador, así como el correcto análisis que se debe de llevar a cabo para poder realizar circuitos. Al momento de ver el microcontrolador en Proteus resulta de suma importancia el saber ubicar los diferentes puertos y sus funciones para de esta forma lograr realizar un buen diseño y por lo tanto no dañar el circuito.