

Universidad Nacional Autónoma de México

Facultad de Ingeniería

Laboratorio de Microcomputadoras

Práctica No. 3: Sistema mínimo microcontrolador PIC16F877

Profesor: Rubén Anaya García

Alumnos:

- Murrieta Villegas Alfonso
- Reza Chavarria, Sergio Gabriel
- Valdespino Mendieta Joaquín

Grupo: 4

Semestre: 2021-2

- 2. Completar las conexiones faltantes, utilizando jumpers; cerciorar el alambrado correcto.**

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

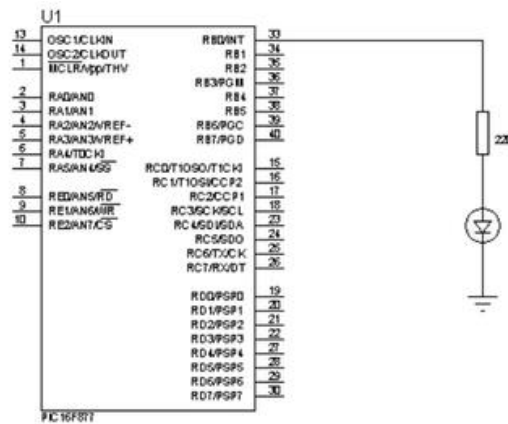


Figura 3.6 Circuito PB0

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

A continuación, se anexarán los comentarios del código.

```
processor 16f877
include <pl6f877.inc>

valor1 EQU h'21' ;valor1=h'21'
valor2 EQU h'22' ;valor2=h'22'
valor3 EQU h'23' ;valor3=h'23'
cte1 EQU 20h ;cte1=20h
cte2 EQU 50h ;cte2=50h
cte3 EQU 60h ;cte3=60h

ORG 0
GOTO INICIO

INICIO:
BSF STATUS,RP0 ;RP0='1'
BCF STATUS,RP1 ;RP1='0'
MOVLW H'0' ;W=00H
MOVWF TRISE ;TRISE=00H
BCF STATUS,RP0 ;RP0='0'
CLRF PORTE ;Limpia el Port B

LOOP2:
BSF PORTE,0 ;Asigna 0 puerto B
CALL RETARDO ;Llama a la subrutina de retardo
BCF PORTE,0 ;Limpia el puerto B
CALL RETARDO ;Llama a función retardo
GOTO LOOP2 ;Llama a subrutina LOOP2
```

```

RETARDO:
    MOVLW    cte1    ;Asigna W=20H
    MOVWF    valor1  ;Asigna valor1=W

TRES:
    MOVLW    cte2    ;Asigna W=50H
    MOVWF    valor2  ;Asigna valor2=W

DOS:
    MOVLW    cte3    ;Asigna W=60H
    MOVWF    valor3  ;Asigna valor3=W

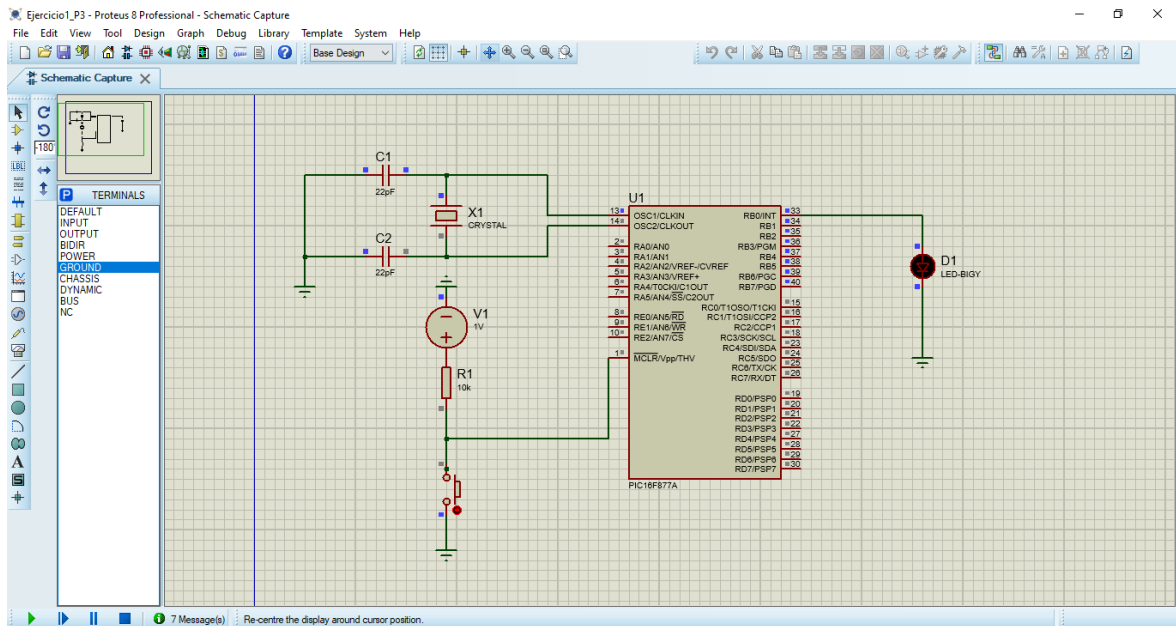
UNO:
    DECFSZ   valor3   ;Decementa valor3 -1
    GOTO     UNO      ;Si el resultado es diferente de 0 ir a uno
    DECFSZ   valor2   ;Decremenar valor2 -1
    GOTO     DOS      ;Si el resultado es diferente de 0 ir a dos
    DECFSZ   valor1   ;Decremenar valor1 -1
    GOTO     TRES     ;Si el resultado es diferente de 0 ir a tres
    RETURN
END

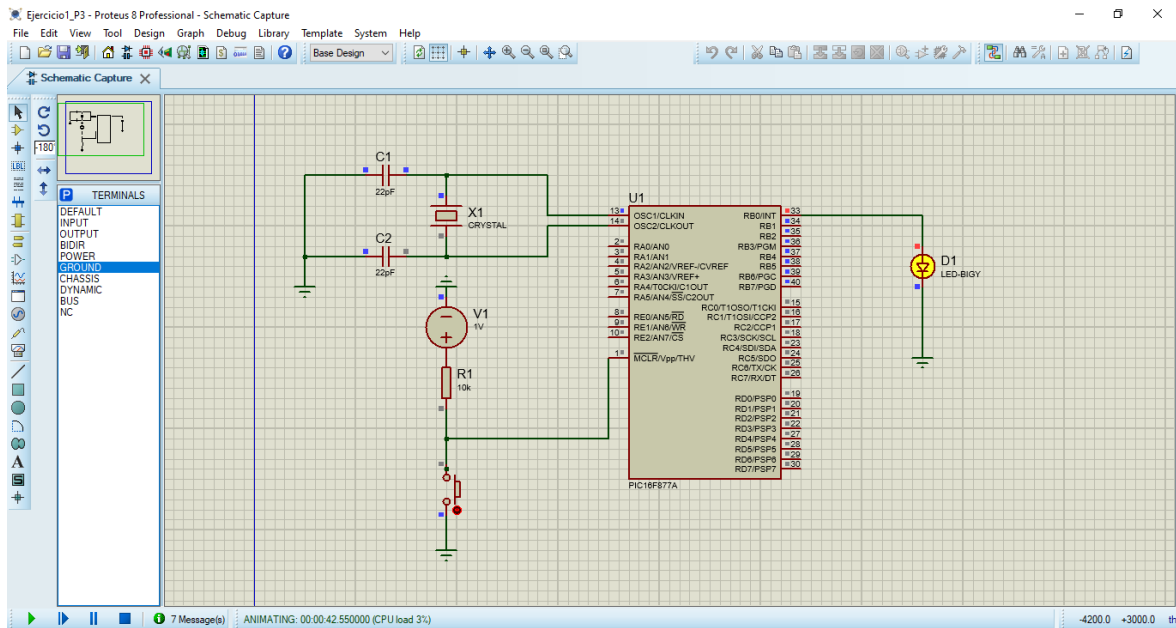
```

Código 1: Ejercicio 1

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

Al cargar el archivo ensamblado y prender el simulador de Proteus genera el encendido y apagado del led a partir de la salida del puerto B. Por cada encendido y apagado existe un tiempo de retardo.





Simulación 1: Simulación del ejercicio 1 en Proteus

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

Al cambiar el valor de la constante 1 genera que el tiempo de retardo que se genera en el encendido y apagado del LED es más rápido en comparación al archivo original. Esto se debe a la disminución del tiempo de la subrutina, con respecto a la cantidad de ciclos a realizar.

7. Modifique cte1 a 80h; ¿ensamblar y programar, existe algún cambio?

Con el aumento de la constante de 80H genera un aumento considerable en el retardo del encendido y apagado del LED. Existe un aumento en la cantidad de ciclos realizados.

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.

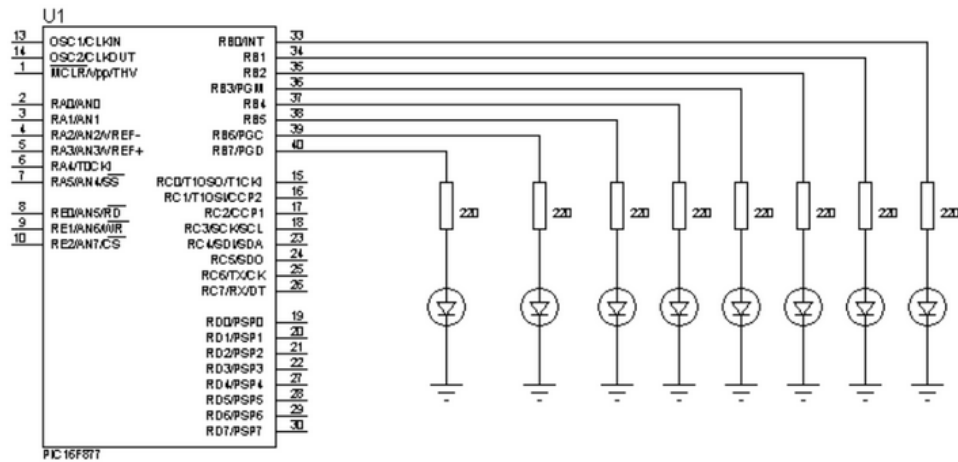


Figura 3.7 Conexión del sistema mínimo al módulo de 8 leds

Código:

```
processor 16f877
include <pl6f877.inc>
valor1 equ h'21'      ;valor1=h'21'
valor2 equ h'22'      ;valor2=h'22'
valor3 equ h'23'      ;valor3=h'23'
ctel equ 10h          ;ctel=20h
cte2 equ 50h          ;cte2=50h
cte3 equ 60h          ;cte3=60h

ORG 0
GOTO INICIO
ORG 5
INICIO
    BSF STATUS,RP0      ;RP0='1'
    BCF STATUS,RP1      ;RP1='0'
    MOVLW H'0'          ;W=00H
    MOVWF TRISB          ;TRISB=00H
    BCF STATUS, RP0      ;RP0='0'
    CLRF PORTB          ;Limpia el port B

loop2
    MOVLW 0xFF          ;Asignar FF al puerto B
    MOVWF PORTB          ;W=PORTB
    CALL retardo         ;Llama a la subrutina de retardo
    CLRF PORTB          ;Limpia el puerto B
    CALL retardo         ;Llama a la subrutina de retardo
    GOTO loop2          ;llama a Loop2

retardo
    MOVLW ctel          ;W=20H
    MOVWF valor1        ;valor1=20H
```

```

tres
    MOVLW cte2      ;W=50H
    MOVWF valor2    ;valor2=50H

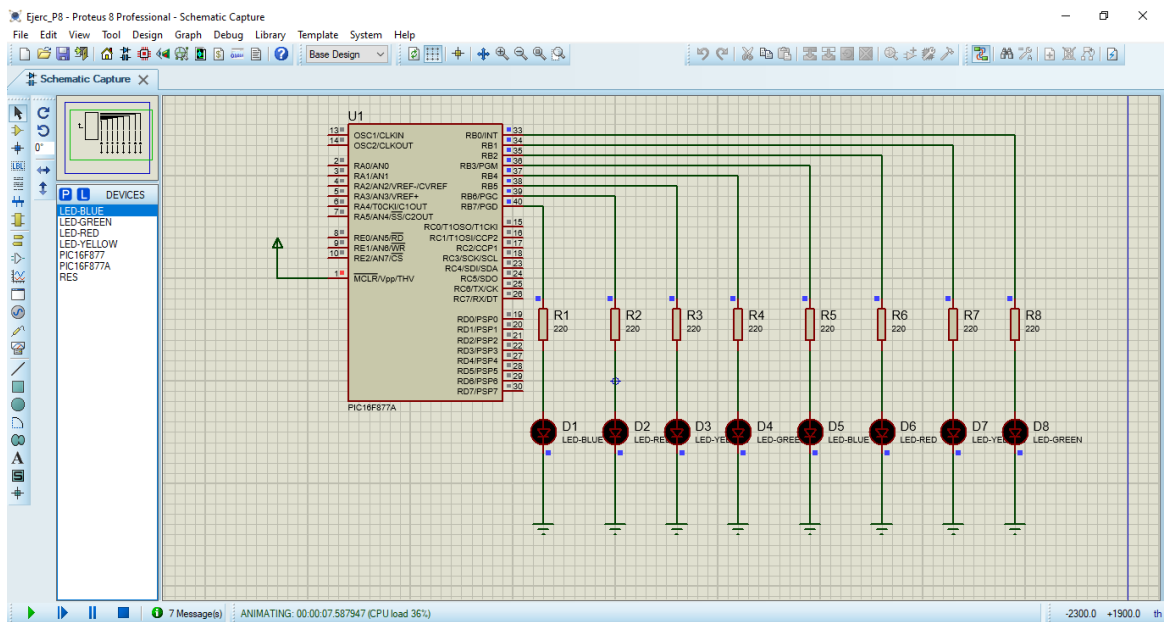
dos
    MOVLW cte3      ;W=60h
    MOVWF valor3    ;valor3=60H

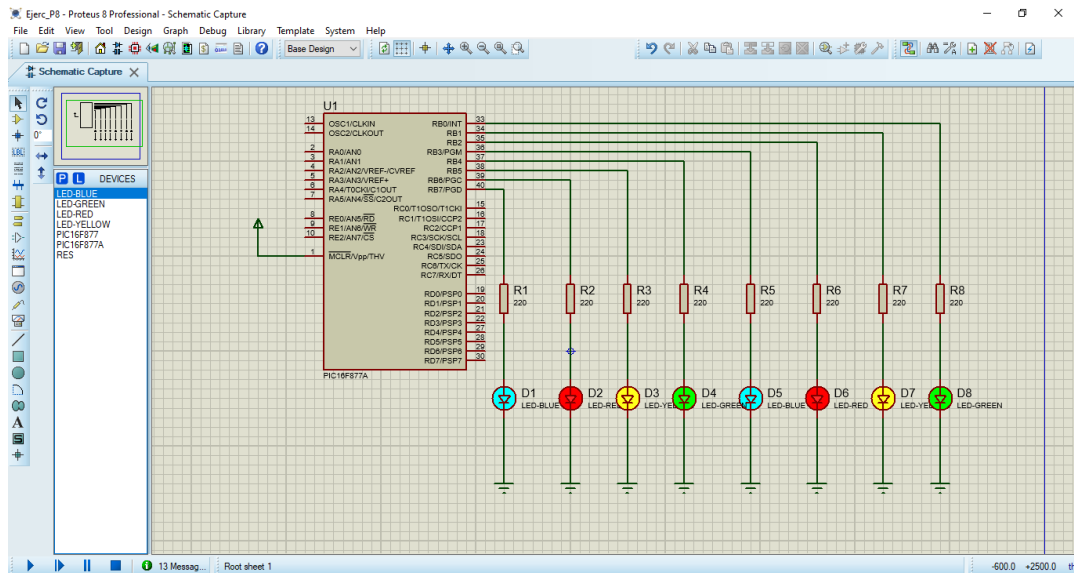
uno
    DECFSZ valor3    ;Decementa valor3 -1
    GOTO uno        ;Si el resultado es diferente de 0 ir a uno
    DECFSZ valor2    ;Decementa valor2 -1
    GOTO dos        ;Si el resultado es diferente de 0 ir a dos
    DECFSZ valor1    ;Decementa valor1 -1
    GOTO tres       ;Si el resultado es diferente de 0 ir a tres
RETURN
END

```

Código 2: Código de activación de todo el puerto B

Simulación Proteus:

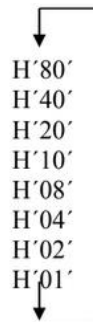




Simulación 2: Simulación de encendido y apagado de todo el puerto B

- Realizar un programa que muestre la siguiente secuencia en el puerto B con retardos de 1/2 segundo.

Secuencia:



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

Código:

Para el manejo del problema se utilizó la función de recorrimiento para los valores del puerto b. Se revisa si se llega a recorrer por completo hasta que se active el bit de acarreo. Si este llega a activarse, se reinicia los valores del puerto B.


```

processor 16f877
include <pl6f877.inc>
;AUX      EQU h'20'          ;auxiliar para el recorrimiento
valor1 EQU h'21'          ;valor1=h'21'
valor2 EQU h'22'          ;valor2=h'22'
valor3 EQU h'23'          ;valor3=h'23'
cte1 EQU 20h              ;cte1=20h
cte2 EQU 50h              ;cte2=50h
cte3 EQU 60h              ;cte3=60h

ORG 0                    ;Vector de RESET, origen de programa
GOTO inicio
ORG 5

inicio
BSF STATUS, RP0
bcf STATUS, RP1
movlw h'0'
movwf TRISE             ;Configurar puerto B como salida
bcf STATUS, RP0         ;Cambio BANCO 0
clrf PORTE              ;Limpia los bits del PUERTO B

loop2
MOVLW H'80' ; W <= 80H
BCF STATUS,0 ; CARRY <= '0'b
MOVWF PORTE ; PORTE <= 80H

CICLO:
CALL retardo
RRF PORTE,F ; Rota a la derecha PORTE
BTFS STATUS,0 ; skip si CARRY = '1'b
GOTO CICLO
GOTO loop2

retardo
movlw cte1              ;W=20H
movwf valor1            ;valor1=20H

tres
movwf cte2              ;W=50H
movwf valor2            ;valor2=50H

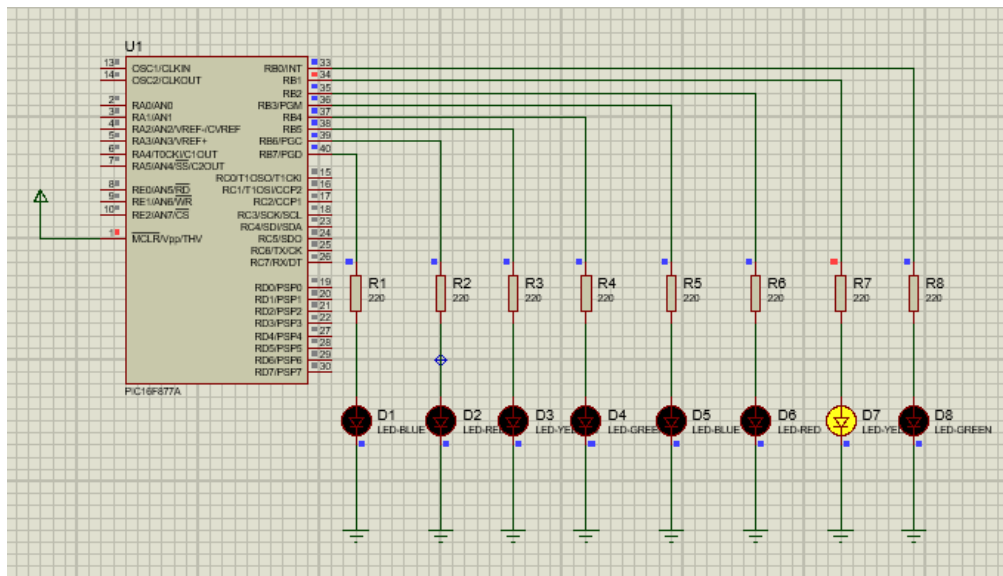
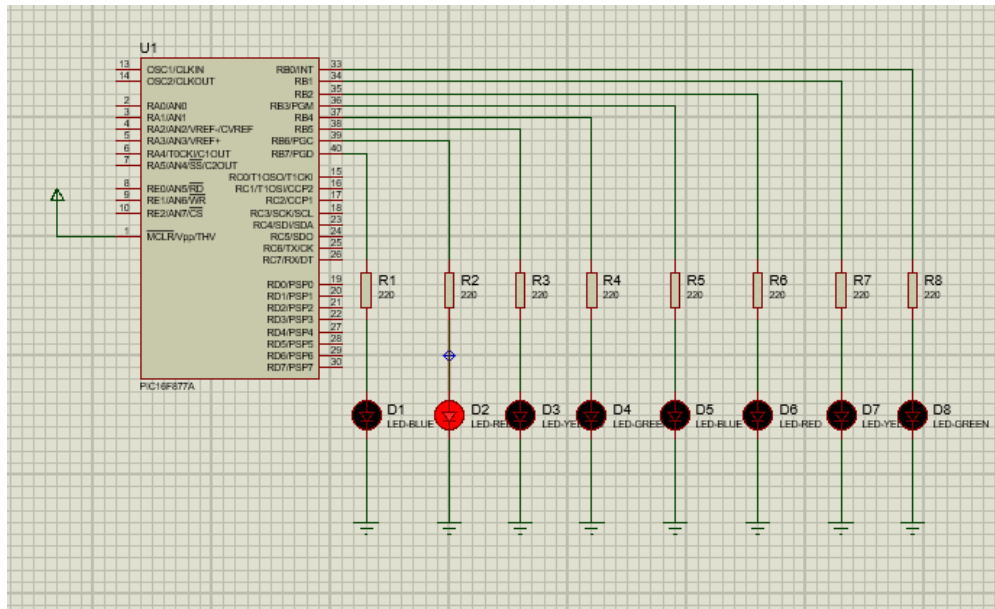
dos
movlw cte3              ;W=60h
movwf valor3            ;valor3=60H

uno
decfsz valor3           ;Decementa valor3 -1
goto uno                ;Si el resultado es diferente de 0 ir a uno
decfsz valor2           ;Decementa valor2 -1
goto dos                ;Si el resultado es diferente de 0 ir a DOS
decfsz valor1           ;Decementa valor1 -1
goto tres               ;Si el resultado es diferente de 0 ir a tres
return
end

```

Código 3: Código de Recorrimiento

Simulación de Proteus:



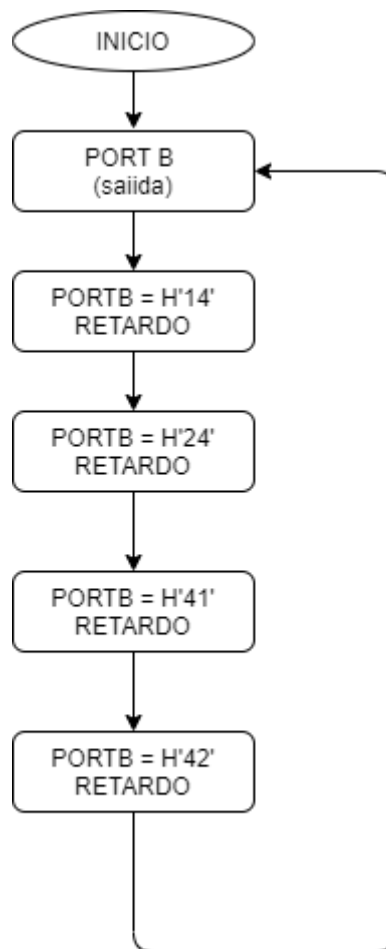
10. Realizar un programa que controle el funcionamiento de dos semáforos; cada estado tendrá una duración de 2 segundos.

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

NOTA: El estado 3 en la segunda posición no debería ser V1 sino V2 debido a que el semáforo 1 no puede tener dos condiciones al mismo tiempo



Diagrama de flujo del programa:



Código del ejercicio:

```

processor 16f877      ;
include <pl6f877.inc>
valor1 equ h'21'
valor2 equ h'22'
valor3 equ h'23'
cte1 equ 90h
cte2 equ 90h
cte3 equ 0A0h

        org 0
        goto inicio
        org 5

inicio bsf STATUS, RP0
        bcf STATUS, RP1 ;Cambio a Banco
        movlw h'0'
        movwf TRISE
        bcf STATUS, RP0
        clrf PORTE

loop2
        movlw h'14' ;Estado 1 Verde 1 Rojo 2
        MOVWF PORTE
        call retardo

        movlw h'24' ;Estado 2 Amarillo 1 Rojo 2
        MOVWF PORTE
        call retardo

        movlw h'41' ;Estado 3 Rojo 1 Verde2
        MOVWF PORTE
        call retardo

        movlw h'42' ; Estado 4 Rojo 1 Amarillo 2
        MOVWF PORTE
        call retardo

        goto loop2

retardo
        movlw cte1          ;W=20H
        movwf valor1        ;valor1=20H

tres
        movwf cte2          ;W=50H
        movwf valor2        ;valor2=50H

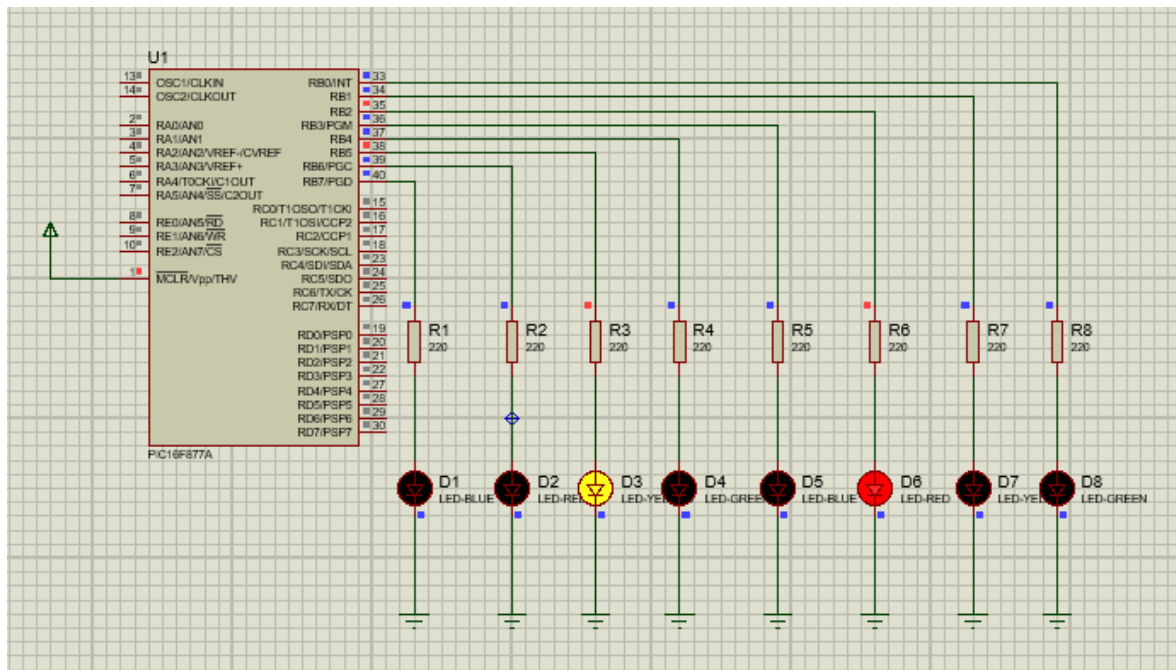
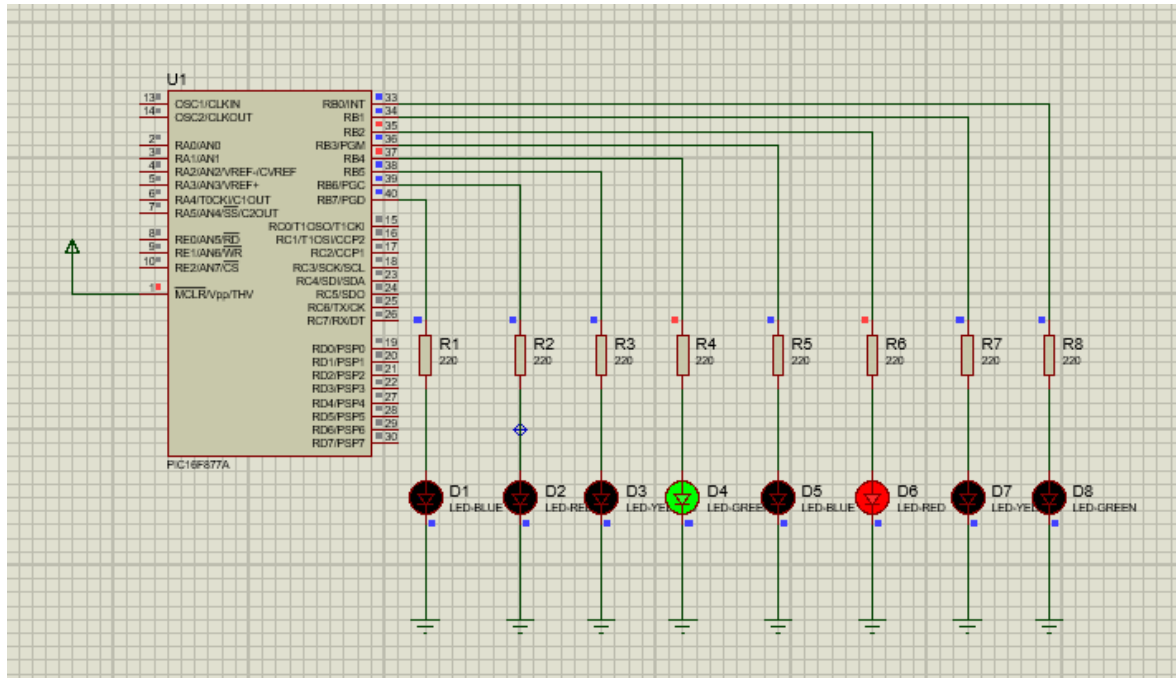
dos
        movlw cte3          ;W=60h
        movwf valor3        ;valor3=60H

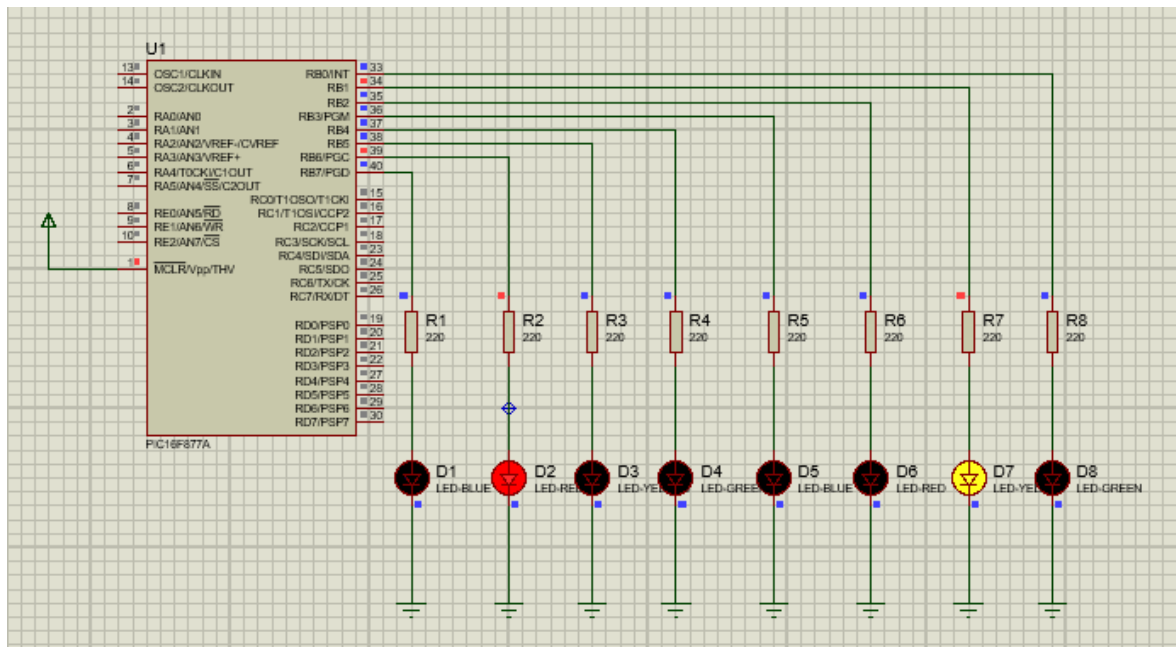
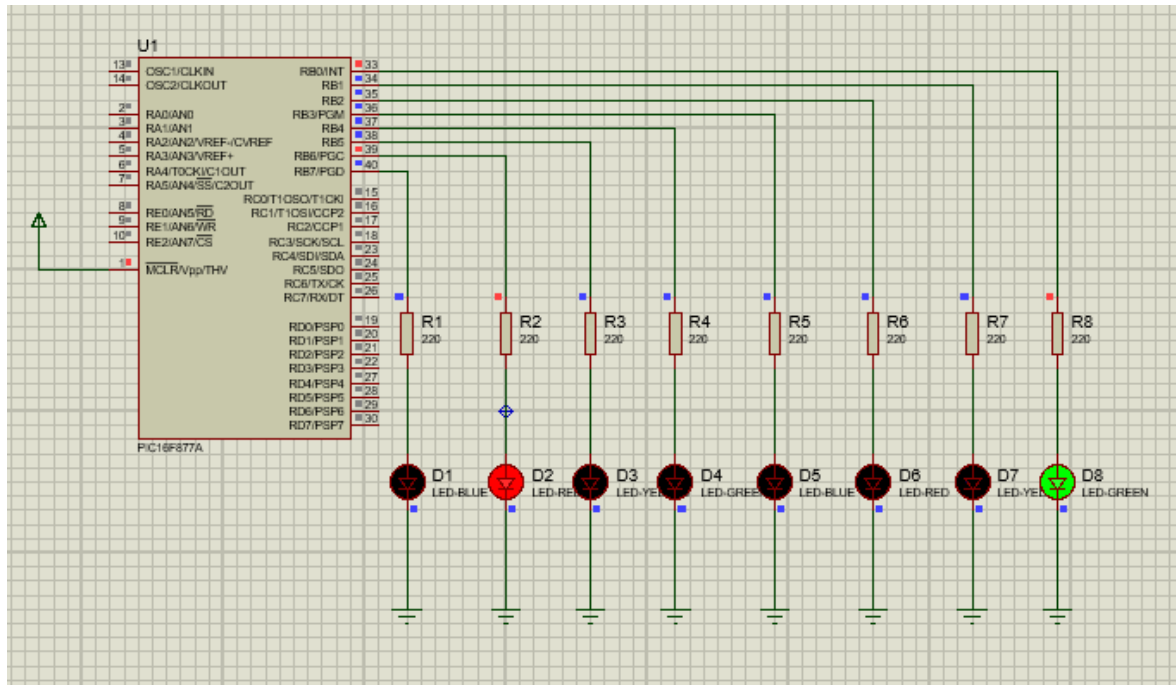
uno
        decfsz valor3        ;Decementa valor3 -1
        goto uno             ;Si el resultado es diferente de 0 ir a uno
        decfsz valor2        ;Decementa valor2 -1
        goto dos             ;Si el resultado es diferente de 0 ir a DOS
        decfsz valor1        ;Decementa valor1 -1
        goto tres            ;Si el resultado es diferente de 0 ir a tres
        return
end

```

Código 4: Ejercicio del semáforo

Evidencia empleando Proteus:





Simulación 3: Estados del Semáforo

Conclusiones

Alfonso Murrieta Villegas

En la presente práctica aprendimos a relacionar esquemas o diagramas de sistemas mínimos de microcontroladores a través de un software de comunicación, además y como parte práctica realizamos diversos programas con la finalidad de utilizar los puertos de salida del microcontrolador en una modalidad de salida.

Por último, realizamos simulaciones mediante Proteus con la finalidad de representar nuestros programas mediante hardware que nos hiciera explícitos nuestros resultados además de ejecutar programas en tiempo real

Joaquín Valdespino Mendieta

En esta práctica pudimos interpretar los diagramas o esquemáticos de los sistemas de microcontrolador, así como el software dedicado, por otro lado, se logró observar e implementar la lógica detrás de los programas que utilizan componentes o puertos de salida todo esto mediante simulaciones de Proteus, que nos dan un paso para implementarlo en físico, retomando así la programación a bajo nivel de estos dispositivos.

Sergio Gabriel Reza Chavarría

En la práctica se pudo aterrizar los conceptos vistos en la parte de programación en el simulador de Proteus. Esto permite a los alumnos en la creación de componentes ya programados para, en un futuro, poder implementarlos en físico.

Además, se pudieron revisar los esquemáticos de los componentes para darnos una idea de cómo se manejarían en formato físico.