

Universidad Nacional Autónoma de México

Facultad de Ingeniería

Laboratorio de Microcomputadoras

Práctica No. 4: Puertos Paralelos E/S

Profesor: Rubén Anaya García

Alumnos:

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

Grupo: 4

Semestre: 2021-2

Práctica No. 4 Puertos Paralelos E/S

Objetivos

Emplear los puertos paralelos que contiene un microcontrolador para realizar funciones de control, configurando estos como entrada y salida.

Desarrollo

Para cada uno de los siguientes apartados, realizar los programas solicitados y comprobar el funcionamiento de ellos.

1.- Empleando dos puertos paralelos del microcontrolador PIC, uno de ellos configurado como entrada y el otro como salida; realizar un programa que de acuerdo al valor del bit menos significativo del puerto A, se genere la acción indicada en el puerto B.

Valor PA0	Acción puerto B
0	00000000
1	11111111

Código del ejercicio:

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

valor1 EQU h'21' ;valor1=h'21'
valor2 EQU h'22' ;valor2=h'22'
valor3 EQU h'23' ;valor3=h'23'
cte1 EQU 30h
cte2 EQU 50h
cte3 EQU 60h

ORG 0 ;Carga al vector de RESET la dirección de inicio
GOTO INICIO
ORG 05 ;Dirección de inicio del programa del usuario
INICIO:
    CLRF PORTA
    BSF STATUS,RP0 ;Cambia la banco 1
    BCF STATUS,RP1
    MOVLW 07H ;Configura puertos A y E como digitales
    MOVWF ADCON1
    MOVLW 3FH ;Configura el puerto A como entrada
    MOVWF TRISA

    MOVLW h'0'
    MOVWF TRISE ;Configura el puerto B como salida
    BCF STATUS,5 ;Cambio BANCO 0
    CLRF PORTE ;Limpia los bits del PUERTO B

LOOP2:
    BTFSC PORTA,0 ;Revisa si el bit 0 del puerto a esta en 0
    GOTO ENCENDER ;Si no es asi, llama a subrutina de encendido
    GOTO APAGAR ;Si es asi, llama a la subrutina de apagado

ENCENDER:
    MOVLW H'FF' ;Cargamos a w el valor de FFh=11111111b
    MOVWF PORTE ;Esto nos permitira encender todos los bits
                  ;del puerto B
    GOTO LOOP2
APAGAR:
    CLRF PORTE ;Limpia el puerto B
    GOTO LOOP2

end
```

Código 1: Ejercicio 1

Para el funcionamiento se necesitó el cambio de banco y la configuración de los puertos A y E para poder ser utilizados como entradas. También se configuraron las salidas con el puerto B del PICF877.

La función principal es revisar si el bit 0 del puerto a está levantado o apagado. Si el bit vale 0 se limpiará la información del puerto B, si el bit es 1 se asignará el valor FF en el puerto B para encender todas las salidas del puerto B.

Diagrama de flujo

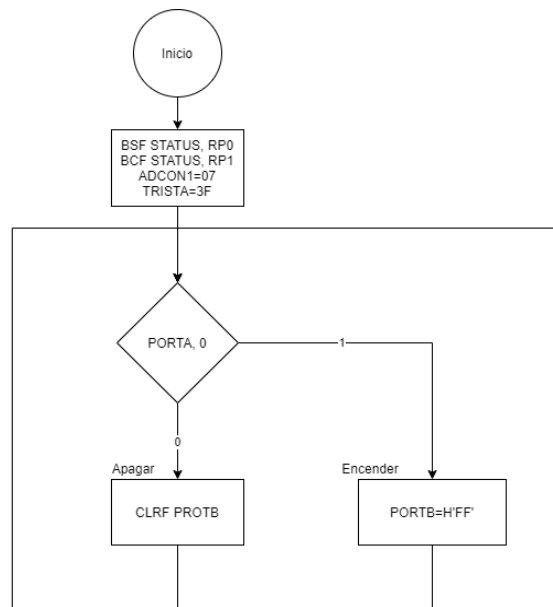
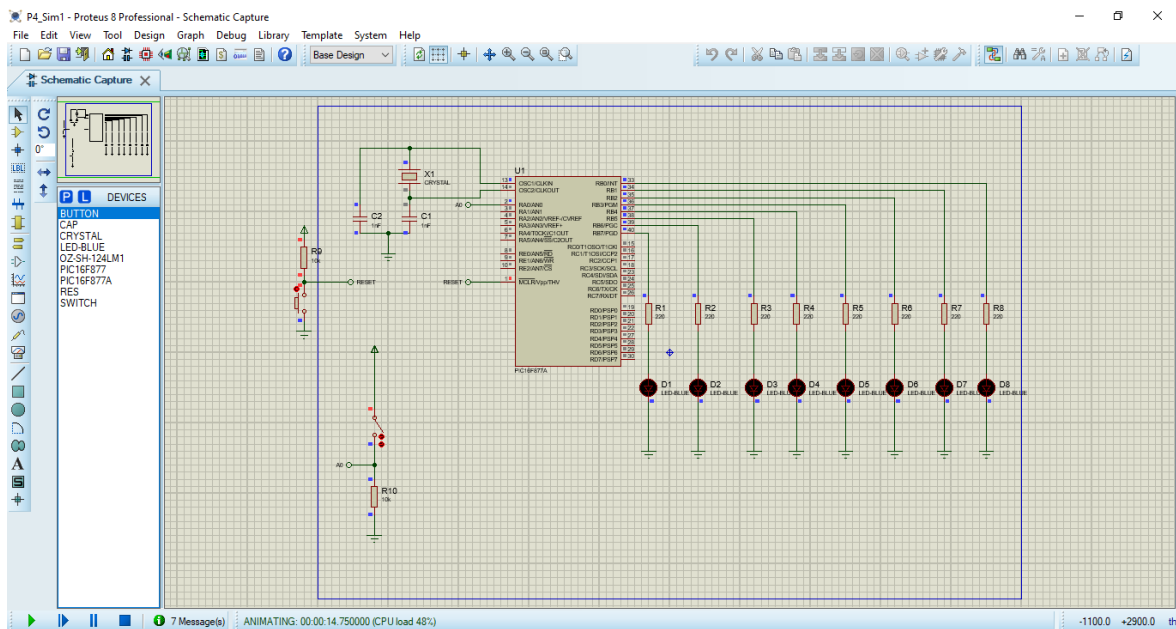
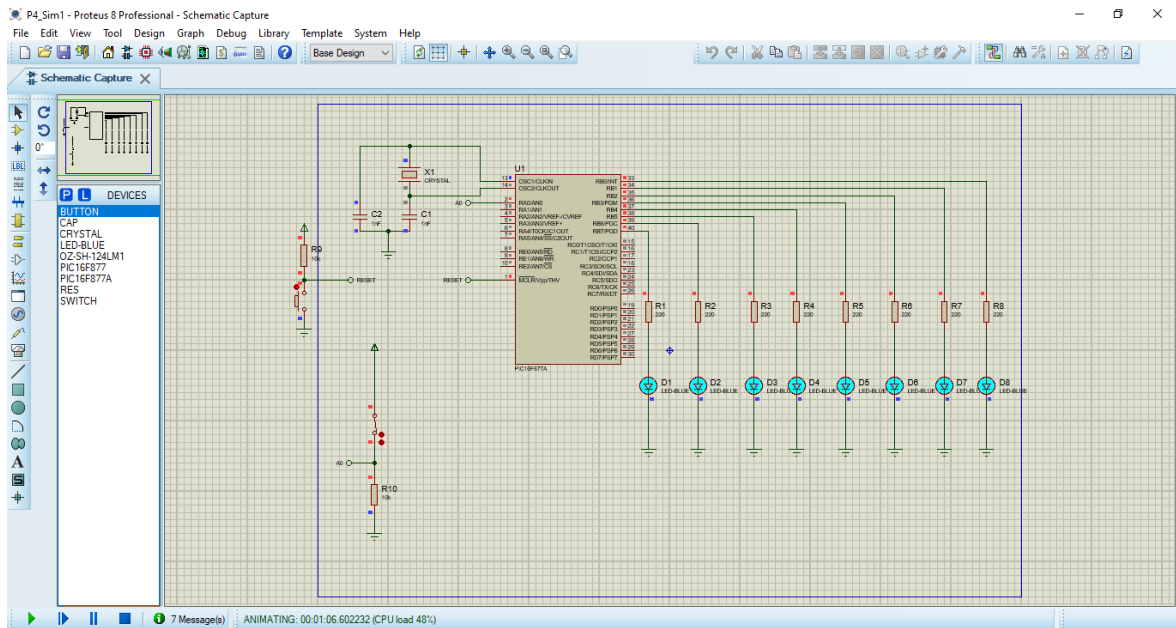


Diagrama 1: Ejercicio 1

Simulación



Simulación 1: Simulación del ejercicio 1, estado 0 (Apagado)



Simulación 2: Simulación del ejercicio 1, estado 1 (Encendido)

2.- Realizar un programa, el cual realice las siguientes acciones de control, para lo cual requiere trabajar un puerto de entrada y otro puerto de salida, usar los sugeridos en el ejercicio anterior; generar retardos de $\frac{1}{2}$ seg., en las secuencias que lo requieran.

DATO	ACCION	Ejecución
\$00	Todos los leds apagados	00000000
\$01	Todos los leds encendidos	11111111
\$02	Corrimiento del bit más significativo hacia la derecha	10000000 01000000 00100000 00000001
\$03	Corrimiento del bit menos significativo hacia la izquierda	00000001 00000010 00000100 10000000
\$04	Corrimiento del bit más significativo hacia la derecha y a la izquierda	10000000 01000000 00000001 00000010 10000000
\$05	Apagar y encender todos los bits.	00000000 11111111

Código del ejercicio:

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

;Variables para el DELAY
valor1 EQU h'21'
valor2 EQU h'22'
valor3 EQU h'23'
cte1 EQU 10h
cte2 EQU 50h
cte3 EQU 60h

v0 EQU h'24'
v1 EQU h'25'
v2 EQU h'26'
v3 EQU h'27'
v4 EQU h'28'
v5 EQU h'30'

c0 EQU 0h
c1 EQU 1h
c2 EQU 2h
c3 EQU 3h
c4 EQU 4h
c5 EQU 5h

org 0h
goto INICIO
org 05h

INICIO:
    clrf PORTA
    bcf STATUS,RP0 ;Cambio al Banco 1
    bcf STATUS,RP1
    movlw h'0'
    movwf TRISEB ;Configura Puerto B como salida
    clrf PORTEB ;Limpia los bits de Puerto 1

    movlw 06h ;Configura puertos A y E como digitales
    movwf ADCON1
    movlw 3fh ;Configura el Puerto A como entrada
    movwf TRISA
    bcf STATUS,RP0 ;Regresa al Banco 0

CICLO:
    movlw c0
    movwf v0
    movfw PORTA ;Mueve lo que hay en PORTA a W
    xorwf v0,w ;Verifica si la entrada es $00
    btfsc STATUS,Z ;z=0?
    goto APG ;NO, entonces v0=W
    ;SI, entonces v0!=W

    movlw c1
    movwf v1
    movfw PORTA
    xorwf v1,w ;Verifica si la entrada es $01
    btfsc STATUS,Z
    goto UNOS

    movlw c2
    movwf v2
    movfw PORTA
    xorwf v2,w ;Verifica si la entrada es $02
    btfsc STATUS,Z
    goto DER
```

```

movwf v3
movwf PORTA
xorwf v3,w      ;Verifica si la entrada es $03
btfsc STATUS,Z
goto IZQ

movlw c4
movwf v4
movwf PORTA
xorwf v4,w      ;Verifica si la entrada es $04
btfsc STATUS,Z
goto DERIZQ

ENCAPG:
movlw c5
movwf v5
movwf PORTA
xorwf v5,w      ;Verifica si la entrada es $04
btfsc STATUS,Z
goto APG
movlw h'00'
movwf PORTE
call retardo
movlw h'FF'
movwf PORTE
call retardo
goto CICLO

APG:      ;Apaga los bits del puerto B
movlw h'00'
movwf PORTE
goto CICLO

UNOS:    ;Enciende los bits del puerto B
movlw h'FF'
movwf PORTE
goto CICLO

DER:      ;Realiza corrimiento a la derecha
movlw h'80'
movwf PORTE
call retardo

DER1:
rrf PORTE,1
call retardo
btfss PORTE,0
goto DER1
goto CICLO

IZQ:      ;Realiza corrimiento a la izquierda
movlw h'01'
movwf PORTE
call retardo

IZQ1:
rlf PORTE,1
call retardo
btfss PORTE,7
goto IZQ1
goto CICLO

DERIZQ:   ;Realiza corrimiento a la derecha y
          ;luego a la izquierda
movlw h'80'
movwf PORTE
call retardo

DER2:
rrf PORTE,1
call retardo
btfss PORTE,0
goto DER2

movlw h'01'
movwf PORTE
call retardo

IZQ2:
rlf PORTE,1
call retardo
btfss PORTE,7
goto IZQ2
goto CICLO

```

```

retardo
    movlw ctel      ;Rutina que genera un DELAY
    movwf valor1
tres movwf cte2
    movwf valor2
dos  movlw cte3
    movwf valor3
uno  decfsz valor3
    goto uno
    decfsz valor2
    goto dos
    decfsz valor1
    goto tres
    return
end

```

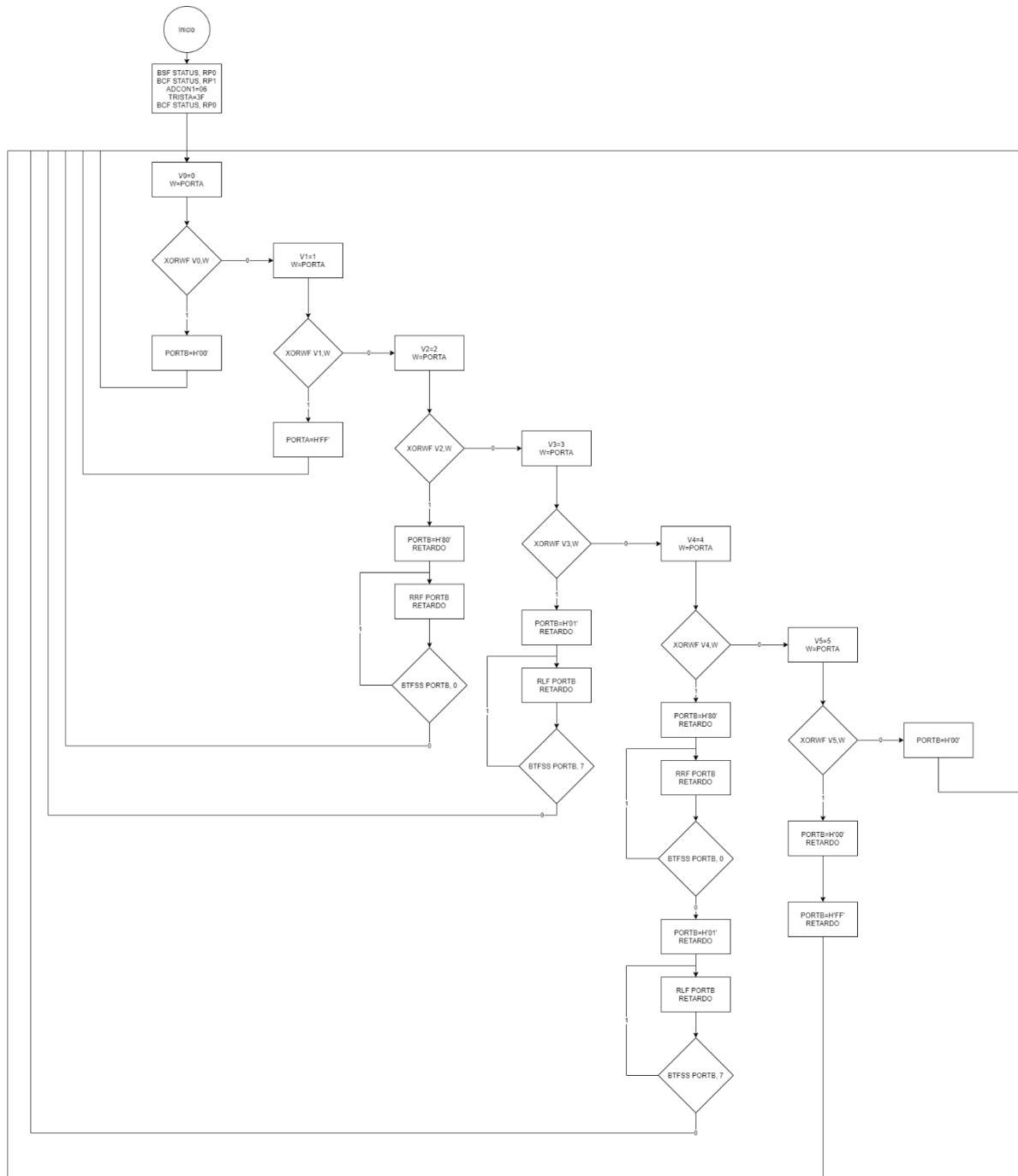
Código 2: Ejercicio 2

En el segundo ejercicio también se configuraron los puertos A y E para las entradas y el puerto b para la salida.

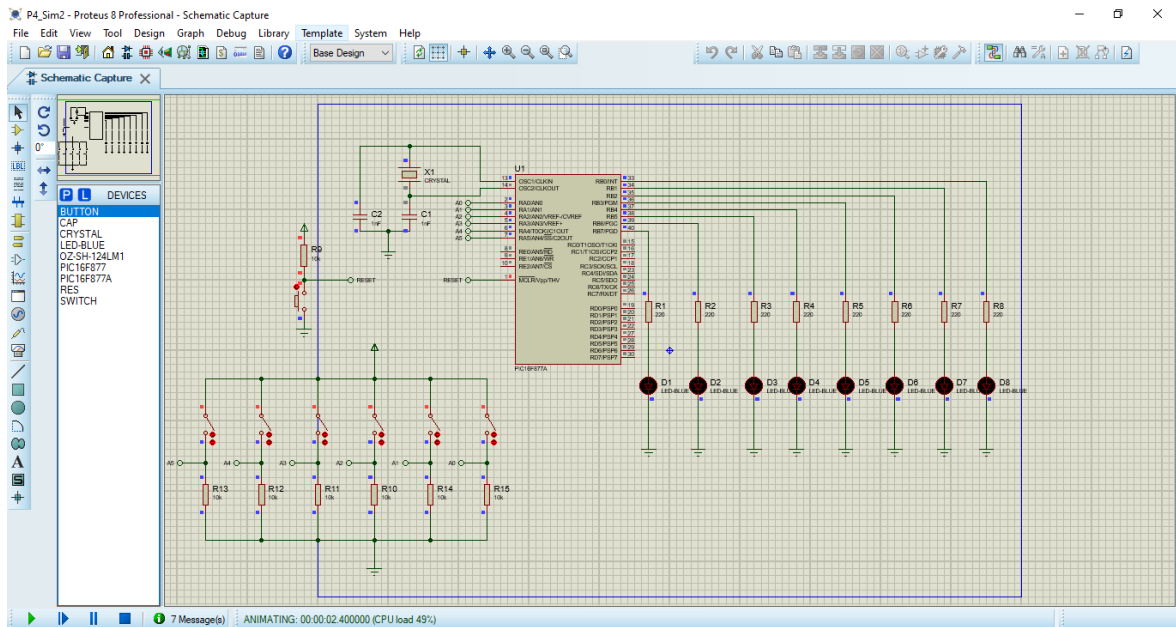
El ejercicio revisará el estado de la entrada para poder revisar que acción se realizará.

- Estado 00: Realiza el apagado de los Led's, esto a partir de la asignación de los valores h'00' al puerto B
- Estado 01: Realiza el encendido del led a partir de la asignación de los valores h'ff' al puerto B.
- Estado 02: A partir de la asignación del valor de h'80' se realizará un corrimiento a la derecha, así el recorrimiento se reflejará en el puerto de salida. Al llegar al último bit se reiniciará la función de ciclo.
- Estado 03: A partir de la asignación del valor de h'01' se realizará un corrimiento a la izquierda, así el recorrimiento se reflejará en el puerto de salida. Al llegar al bit 7 se reiniciará la función de ciclo.
- Estado 04: Para el estado siguiente se realizará la asignación del valor h'80'. Se realizará el recorrimiento a la derecha, se realizará hasta llegar al bit 0. Al terminar se realizará la asignación del valor h'01' y se realizará el recorrimiento a la izquierda hasta llegar al bit 7.
- Estado 05: Para el estado de encendido y apagado se realizará la asignación de h'00' y se realizará la llamada a la subrutina de retardo para que mantenga el apagado por una cantidad de tiempo, al terminar el llamado a la subrutina se realizará la asignación de h'ff' al puerto B e igual se utilizará la subrutina de retardo.

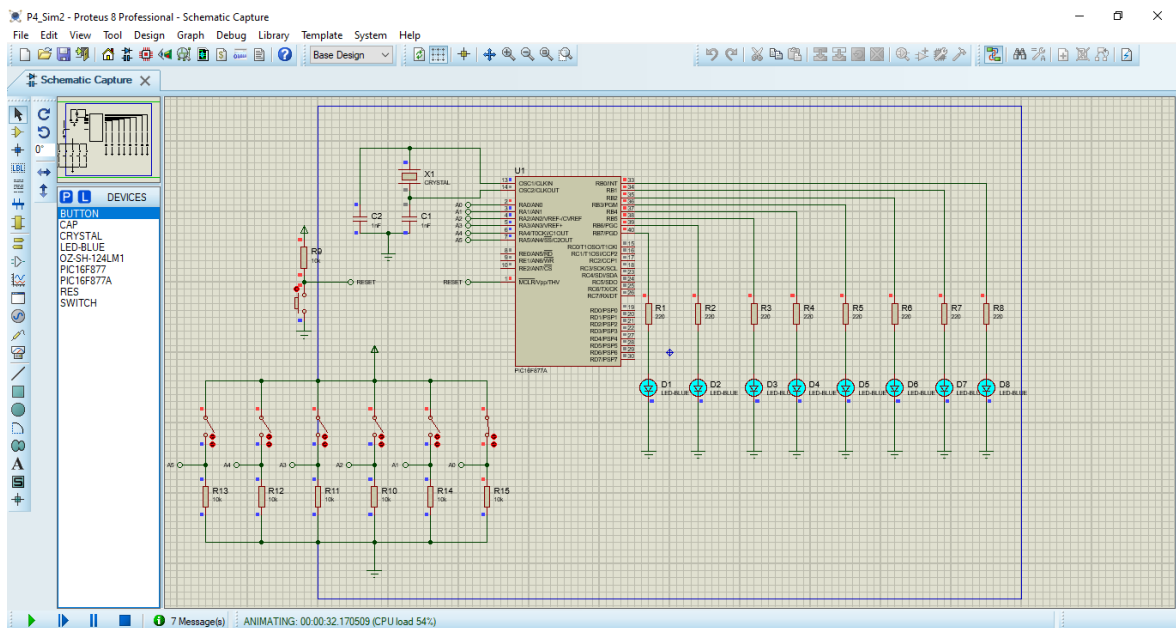
Diagrama de flujo



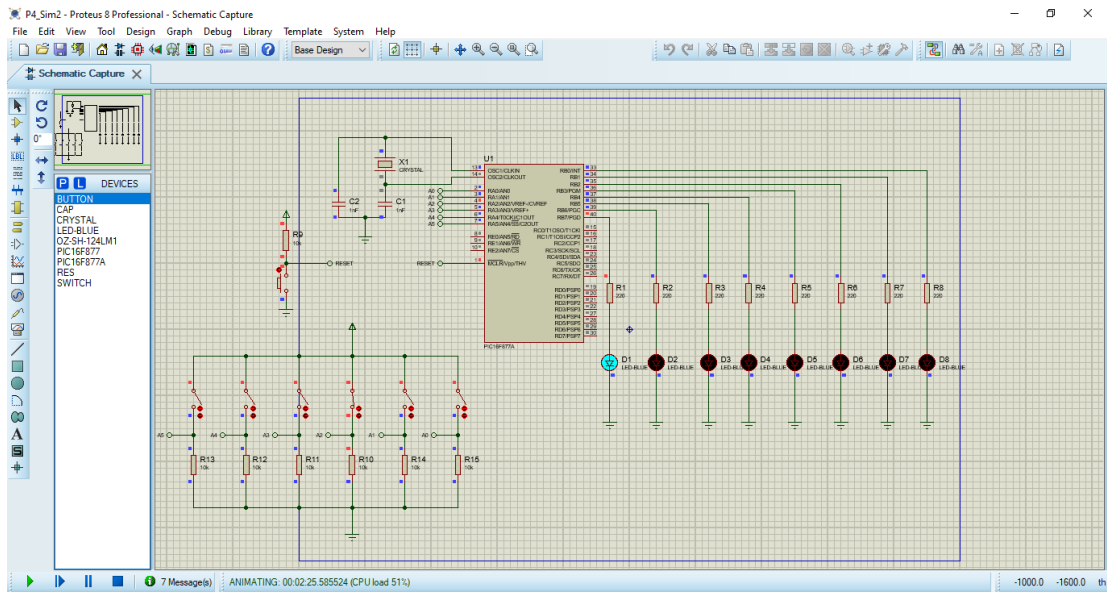
Simulación



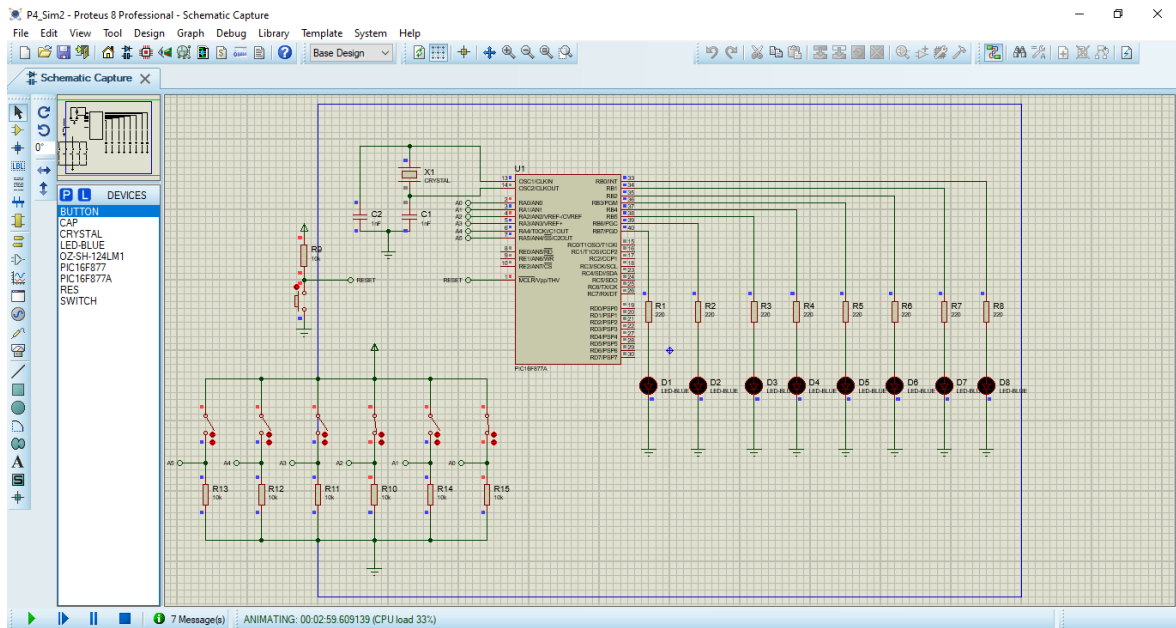
Simulación 3: Estado 00 (Apagado)

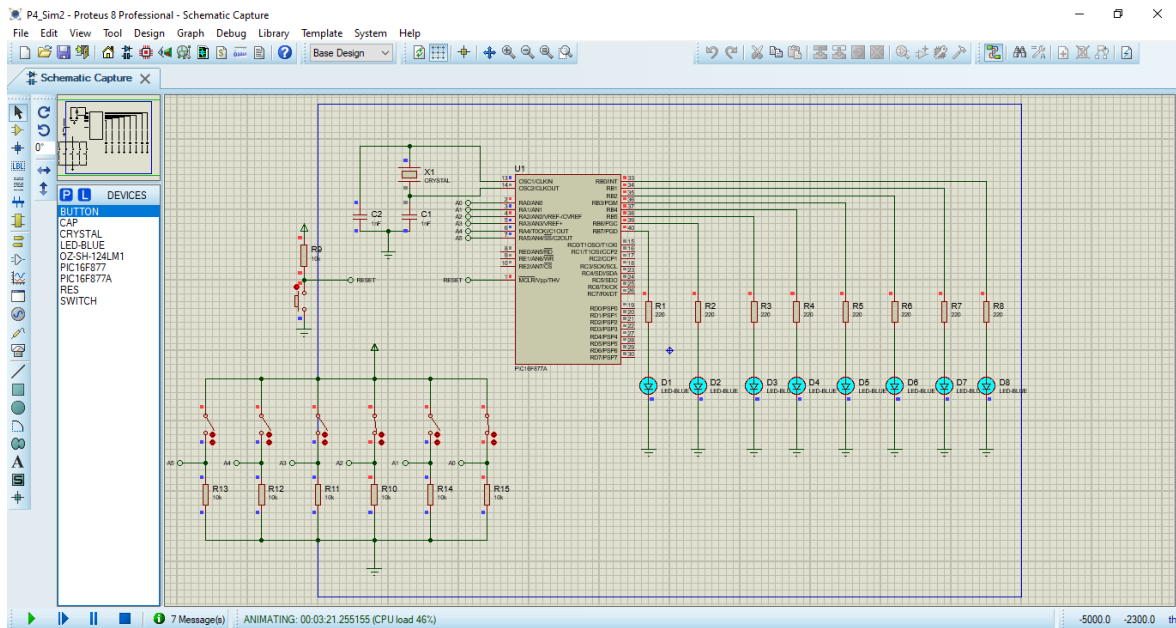


Simulación 4: Estado 01 (Encendido)



Simulación 7: Estado 04 (Recorrimiento de derecha a izquierda)





Simulación 8: Estado 05 (Apagado y encendido)

Conclusiones:

Alfonso Murrieta Villegas

En la presente práctica aprendimos a emplear los puertos de salidas de nuestro microcontrolador con el objetivo de ingresar y egresar datos, así generando diferentes estados por medio de la programación.

Por último, el saber controlar tanto las entradas como salidas de un sistema nos da la ventaja de poder determinar la función a la que tendrá que estar destinado un sistema.

Sergio Gabriel Reza Chavarría

Con la realización de la práctica se pudo llevar a cabo la implementación de nuestros programas en un ambiente de desarrollo práctico. El simulador de Proteus nos permitió realizar el manejo de puertos de entrada y salida y poder con ellos ejercer diferentes acciones ya programadas, ensambladas y cargadas al microcontrolador.

Esto nos proporcionará la manera de interacción entre el programa realizado y las diferentes acciones que se puedan realizar con diferentes dispositivos adicionales

Joaquín Valdespino Mendieta

En la realización de la práctica pudimos implementar y comprender las funciones para el control del microcontrolador, dando puertos de entrada y salida para el flujo de datos que nos permiten tener diversas aplicaciones, como se vio en la implementación de los programas en el microcontrolador.