**FACULTAD DE INGENIERÍA, UNAM**

**LABORATORIO DE MICROCOMPUTADORASS**

**SEMESTRE 2023-2**

**GRUPO 11**

**PREVIO PRÁCTICA 8**

**PUERTOS PARALELOS E/S, PUERTO SERIE**

NOMBRE DEL ALUMNO:

**ARRIAGA MEJÍA JOSÉ CARLOS**

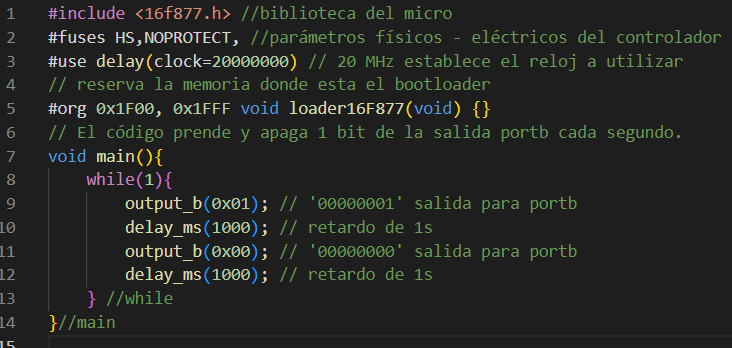
PROFESOR

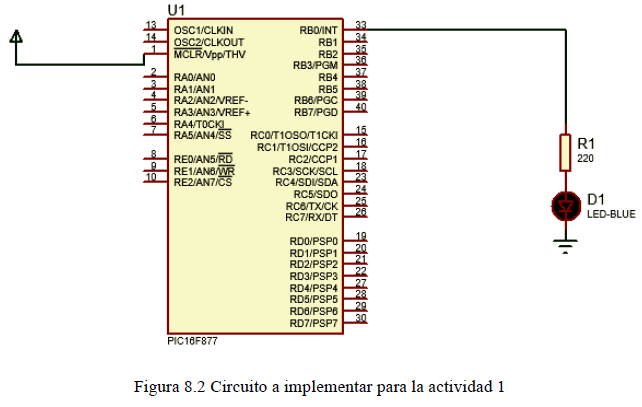
**ING. ROMAN V. OSORIO COMPARAN**

FECHA DE ENTREGA: **12 DE MAYO DE 2023** **CALIFICACION**

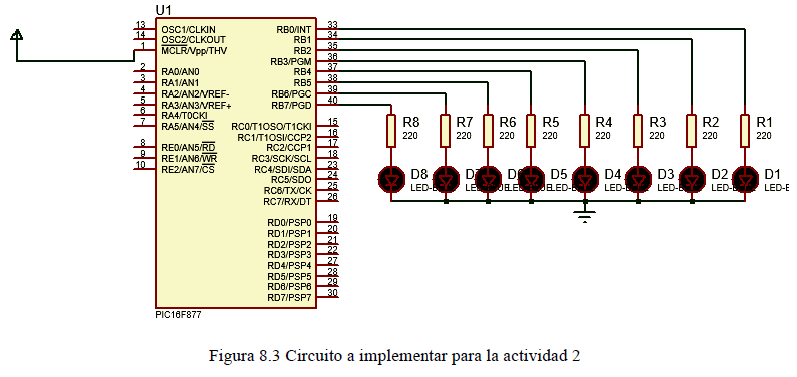
**Objetivo**

Realización de programas a través de programación en C y empleo del puerto serie para visualización y control.





2.- Modificar el programa para que active y desactive todos los bits del puerto B.



#include <16f877.h> //biblioteca del micro

#fuses HS,NOPROTECT, //parámetros físicos - eléctricos del controlador

#use delay(clock=20000000) // 20 MHz establece el reloj a utilizar

// reserva la memoria donde esta el bootloader

#org 0x1F00, 0x1FFF void loader16F877(void) {}

// El código prende y apaga 8 bits de la salida portb cada segundo.

void main(){

while(1){

output\_b(0xff); // '11111111' salida para portb

delay\_ms(1000); // retardo de 1s

output\_b(0x00); // '00000000' salida para portb

delay\_ms(1000); // retardo de 1s

} //while

}//main

3.- Escribir, comentar, compilar el siguiente programa usando el ambiente del PIC C Compiler y comprobar el funcionamiento.

#include <16f877.h> //biblioteca del micro

#fuses HS,NOPROTECT, //parámetros físicos - eléctricos del controlador

#use delay(clock=20000000) // 20 MHz establece el reloj a utilizar

// reserva la memoria donde esta el bootloader

#org 0x1F00, 0x1FFF void loader16F877(void) {}

int var1; // variable global

// el programa asigna la entrada PORTA a la salida PORTB

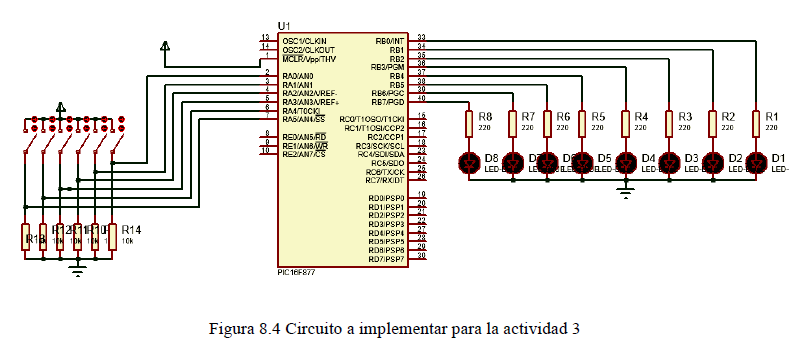
void main(){

while(1){

var1=input\_a(); //lee la entrada del puerto A y almacena en var1

output\_b(var1); //asigna var1 a salida del puerto B

}//while

}//main

4.- Escribir, comentar, compilar, el siguiente programa usando el ambiente del PIC C Compiler y comprobar el funcionamiento.

#include <16f877.h> //biblioteca del micro

#fuses HS,NOPROTECT, //parámetros físicos - eléctricos del controlador

#use delay(clock=20000000) // 20 MHz establece el reloj a utilizar

//utiliza estándar rs232 con la configuración:

//transmite por portC.6 y recibe por portC.7 con 9600 bauds

#use rs232(baud=9600, xmit=PIN\_C6, rcv=PIN\_C7)

// reserva la memoria donde esta el bootloader

#org 0x1F00, 0x1FFF void loader16F877(void) {}

// el programa controla 8 bits de salida e imprime en terminal mensaje

void main(){

while(1){

output\_b(0xff); // salida llena de 8 bits

printf(" Todos los bits encendidos \n\r"); //imprime en consola

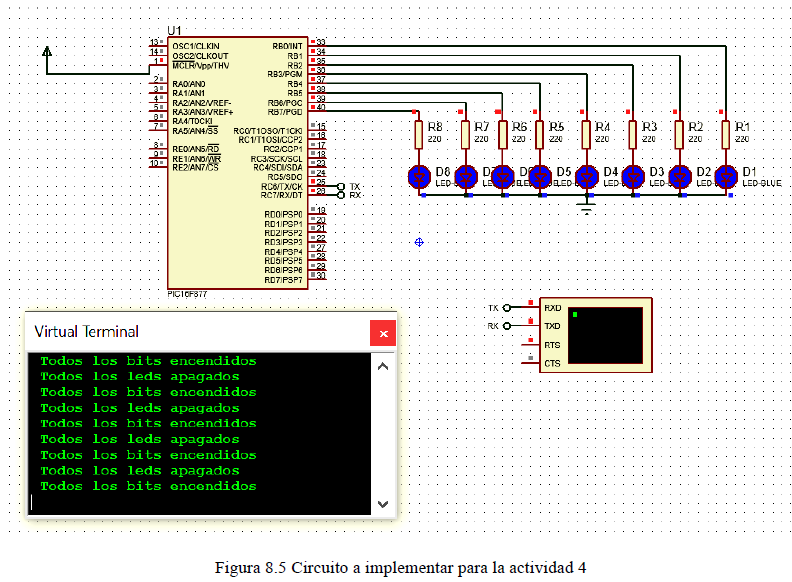
delay\_ms(1000); // retardo 1s

output\_b(0x00); // salida limpia de 8 bits

printf(" Todos los leds apagados \n\r"); //imprime en consola

delay\_ms(1000); // retardo 1s

}//while

}//main

5.- Escribir, comentar, compilar, el siguiente programa usando el ambiente del PIC C Compiler y comprobar el funcionamiento.

**Nota:** La biblioteca lcd.c asigna las terminales para uso del LCD, en la plataforma usada se ha conectado al puerto D; también permite usar el puerto B para las señales de control; en este caso agregar previo a incluir la librería #define use\_portb\_lcd true.

#include <16f877.h> //biblioteca del micro

#fuses HS,NOPROTECT, //parámetros físicos - eléctricos del controlador

#use delay(clock=20000000) // 20 MHz establece el reloj a utilizar

#define use\_portd\_lcd true // establece PORTD como conexión del display LCD

#include <lcd.c> // biblioteca para control de display LCD.

// el programa escribe la frase UNAM FI en display LCD

void main() {

lcd\_init(); //inicialización del display

while( TRUE ) {

lcd\_gotoxy(1,1); //posiciona cursor en (1,1) inicio del primero

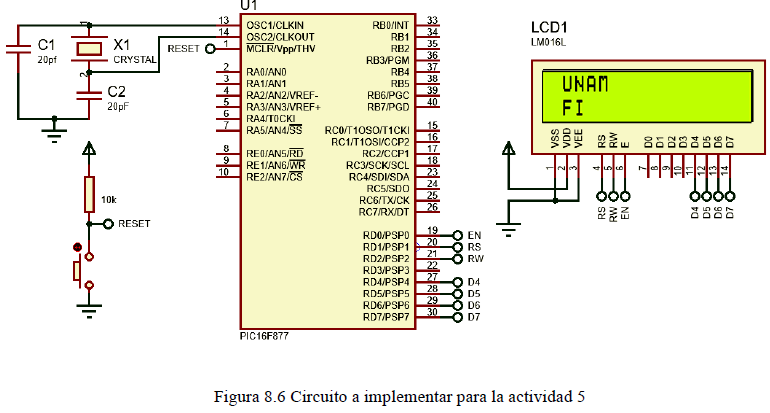
printf(lcd\_putc," UNAM \n "); //escribe los chars en el LCD

lcd\_gotoxy(1,2); //posiciona cursor en (1,2) inicio del segundo

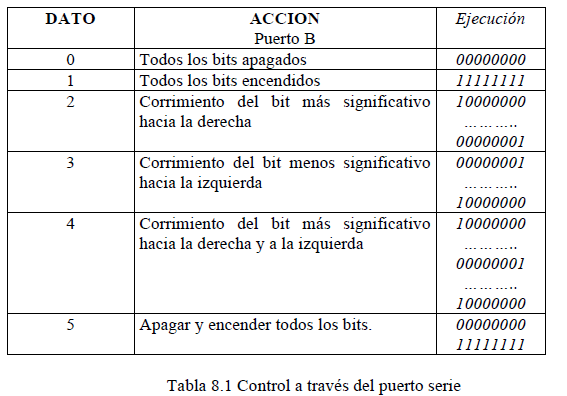
printf(lcd\_putc," FI \n "); //escribe los chars en el LCD

delay\_ms(300); //retardo de 300ms

} //end while

}//end main

6.- Realizar un programa empleando el compilador de C, para ejecutar las acciones mostradas en la siguiente tabla, estas son controladas a través del puerto serie; usar retardos de ½ segundos.



#include <16f877.h> //biblioteca del micro

#fuses HS,NOPROTECT, //parámetros físicos - eléctricos del controlador

#use delay(clock=20000000) // 20 MHz establece el reloj a utilizar

// utiliza estándar rs232 con la configuración:

// transmite por portC.6 y recibe por portC.7 con 9600 bauds

#use rs232(baud=9600, xmit=PIN\_C6, rcv=PIN\_C7)

// reserva la memoria donde esta el bootloader

#org 0x1F00, 0x1FFF void loader16F877(void) {}

char dato; // variable globar tipo caracter 8 bits

void correrDerecha(){ // corrimiento derecha

output\_b(0x80);

delay\_ms(500);

output\_b(0x40);

delay\_ms(500);

output\_b(0x20);

delay\_ms(500);

output\_b(0x10);

delay\_ms(500);

output\_b(0x08);

delay\_ms(500);

output\_b(0x04);

delay\_ms(500);

output\_b(0x02);

delay\_ms(500);

output\_b(0x01);

delay\_ms(500);

}

void correrIzquierda(){ // corrimiento izquierda

output\_b(0x01);

delay\_ms(500);

output\_b(0x02);

delay\_ms(500);

output\_b(0x04);

delay\_ms(500);

output\_b(0x08);

delay\_ms(500);

output\_b(0x10);

delay\_ms(500);

output\_b(0x20);

delay\_ms(500);

output\_b(0x40);

delay\_ms(500);

output\_b(0x80);

delay\_ms(500);

}

void correrMixto(){ // corrimiento mixto

correrDerecha(); // llama corrimiento derecha

correrIzquierda(); // llama corrimiento izquierda

}

// el programa realiza diversas acciones dependiendo de la entrada

void main(){

while(1){

scanf("%c", &dato); // recibe y almacena dato de entrada

switch(dato) { // evalua el caso y realiza la acción

case '0':

output\_b(0x00); // todos los bits apagados

delay\_ms(500);

break;

case '1':

output\_b(0xff); // todos los bits encendidos

delay\_ms(500);

break;

case '2':

correrDerecha(); // Corrimeinto derecha

output\_b(0x00);

delay\_ms(500);

break;

case '3':

correrIzquierda(); // Corrimiento izquierda

output\_b(0x00);

delay\_ms(500);

break;

case '4':

correrMixto(); // ida y vuelta

output\_b(0x00);

delay\_ms(500);

break;

case '5':

output\_b(0xff); // apagar y encender

delay\_ms(500);

output\_b(0x00);

delay\_ms(500);

break;

default:

break;

}// switch

}//while

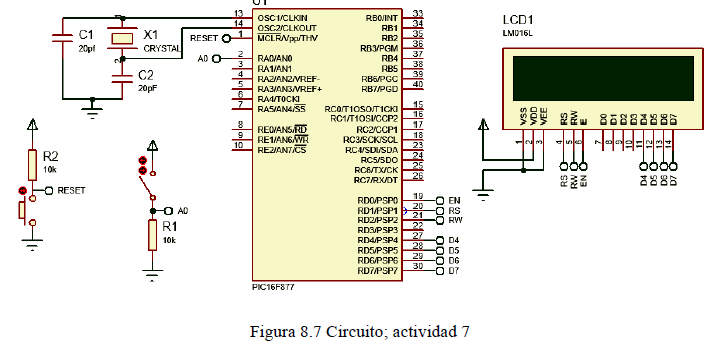
}//main

7.- Realizar un programa que muestre en un Display de Cristal Líquido, la cantidad de veces que se ha presionado un interruptor, el cual esta conectado a la terminal A0.

El despliegue a mostrar es:

a. Primer línea y 5 columna; la cuenta en decimal

b. Segunda línea y 5 columna; la cuenta en hexadecimal



#include <16F877.h> //biblioteca del micro

#fuses HS,NOWDT,NOPROTECT,NOLVP //parametros físicos-eléctricos controlador

#use delay(clock=20000000) // 20 MHz establece el reloj a utilizar

#define use\_portd\_lcd true // establece PORTD como conexión del display LCD

#include <lcd.c> //biblioteca para control de display LCD.

// Variables globales

int conteo; // lleva la cuenta de veces que se activa señal

int dato; // lectura de señal

int ante; // señal anterior (no contar varias veces una misma activación)

// el programa escribe la cuenta de activaciones en decimal y hexadecimal

// en el display LCD.

void main() {

lcd\_init(); //inicialización del display

conteo=0;

while( TRUE ) {

ante=dato;

dato=input\_state(PIN\_A0);

if(dato==1 && dato!=ante){ // detecta solo cambio por subida

conteo++;

}

lcd\_gotoxy(5,1); //posiciona cursor en (5,1) inicio del primero

printf(lcd\_putc," %d\n", conteo); //escribe los chars en el LCD

lcd\_gotoxy(5,2); //posiciona cursor en (5,2) inicio del segundo

printf(lcd\_putc," %x\n", conteo); //escribe los chars en el LCD

delay\_ms(100); //retardo de 300ms

} //end while

}//end mai.}l