

Laboratorio de Microcomputadoras
Práctica No. 8
Programación en C
Puertos Paralelos E/S, Puerto Serie

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

Actividades previas

Instalar el compilador de C para el microcontrolador PIC (PIC C Compiler)

Introducción

El IDE (Entorno de desarrollo Integrado) del compilador contiene los comandos necesarios para crear un proyecto y compilarlo para posteriormente comprobar el funcionamiento en un sistema real o empleando el simulador Proteus.

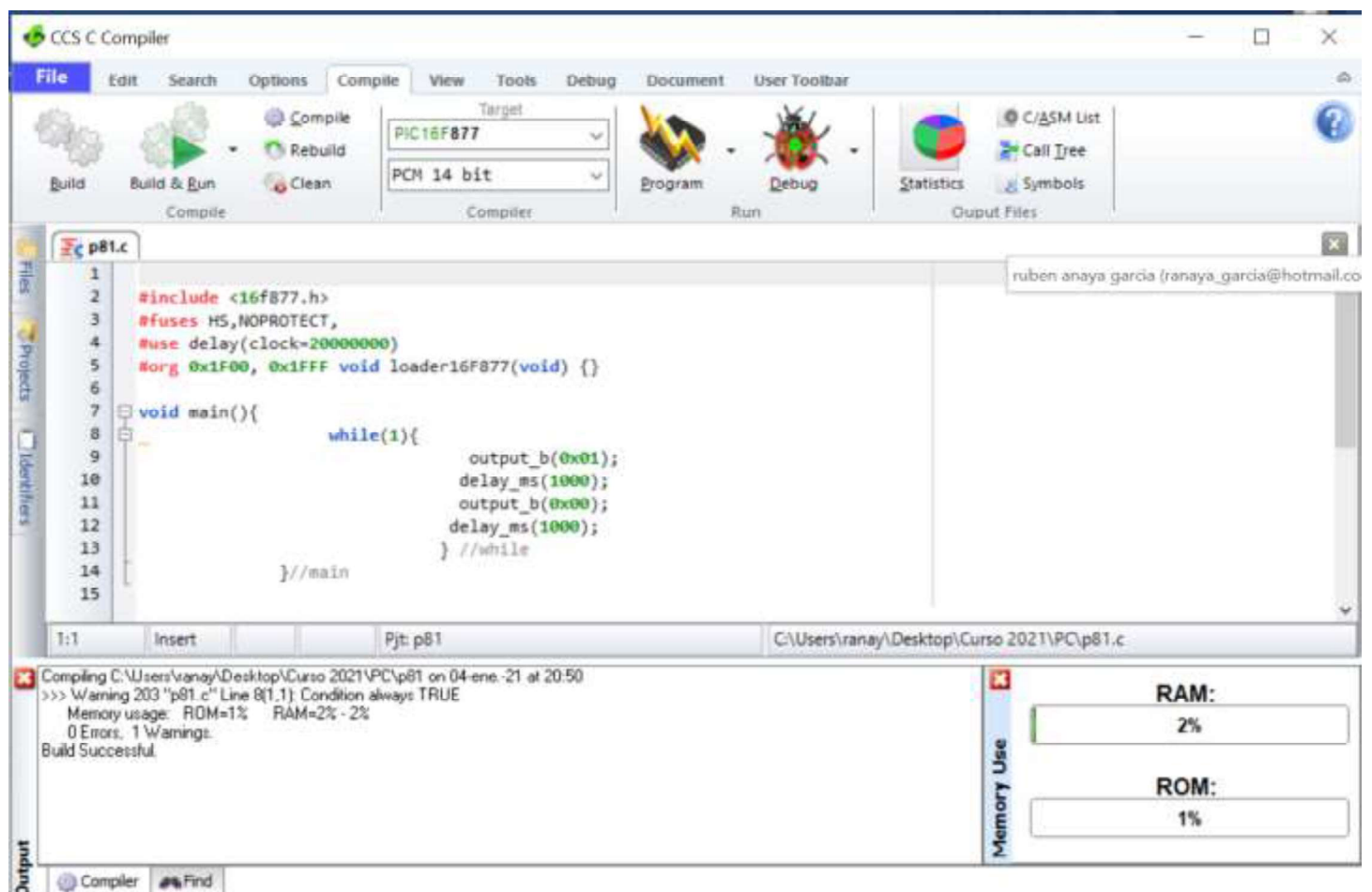


Figura 8.1 IDE del PIC C Compiler

El compilador dispone de una gran variedad de funciones para cada uno de los periféricos del microcontrolador; se recomienda consultar la ayuda del compilador para obtener detalles del uso de las mismas.

Para acceder a la ayuda, seleccionar el icono ?, ubicado en la parte superior derecha (seleccionar **Built in functions**, para acceder a la ayuda de las funciones)

- a. Funciones disponibles para manejo de terminales digitales:

DISCRETE I/O	get tris x()	output X()	output drive()
	input()	output bit()	output low()
	input state()	input change x()	output toggle()
	set tris x()	output float()	
	input x()	output high()	
	port x pullups()		

- b. Funciones para manejo del puerto serie asíncrono:

RS232 I/O	assert()	getch()	putc()
	fgetc()	getchar()	putchar()
	fgets()	gets()	puts()
	fprintf()	kbhit()	setup_uart()
	ftc()	perror()	set_uart_speed()
	fputs()	getc()	printf()
	putc()	scanf()	-

Desarrollo. Realizar los siguientes actividades.

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

```
#include <16f877.h>
#fuses HS,NOPROTECT,
#use delay(clock=20000000)
#org 0x1F00, 0x1FFF void loader16F877(void) {}

void main(){
    while(1){
        output_b(0x01);
        delay_ms(1000);
        output_b(0x00);
        delay_ms(1000);
    } //while
} //main
```

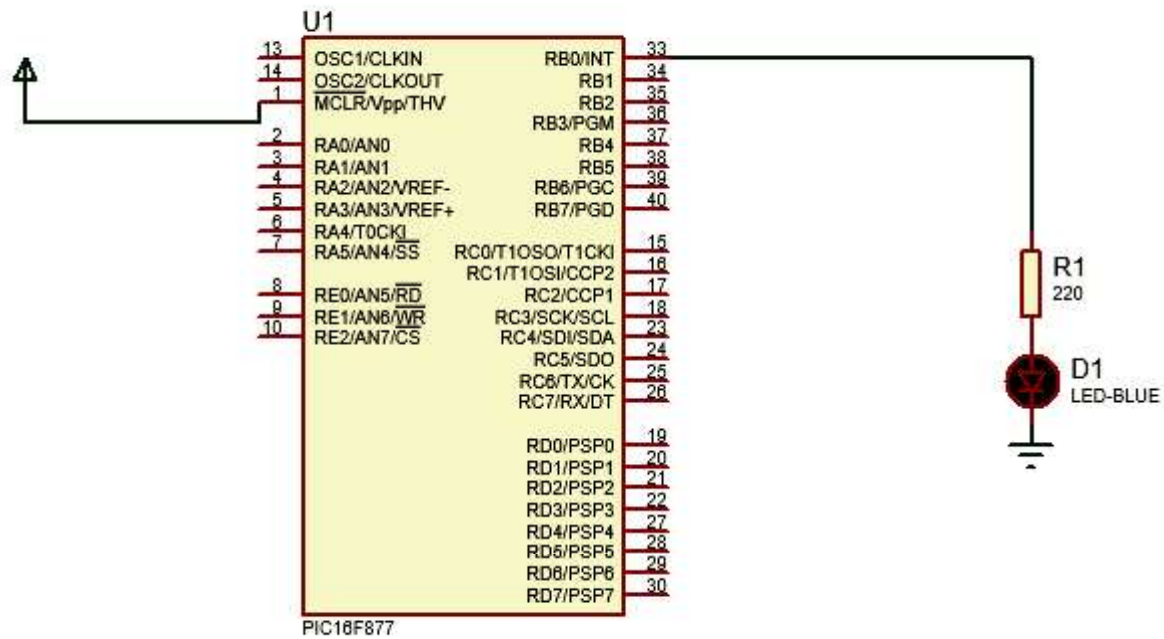


Figura 8.2 Circuito a implementar para la actividad 1

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

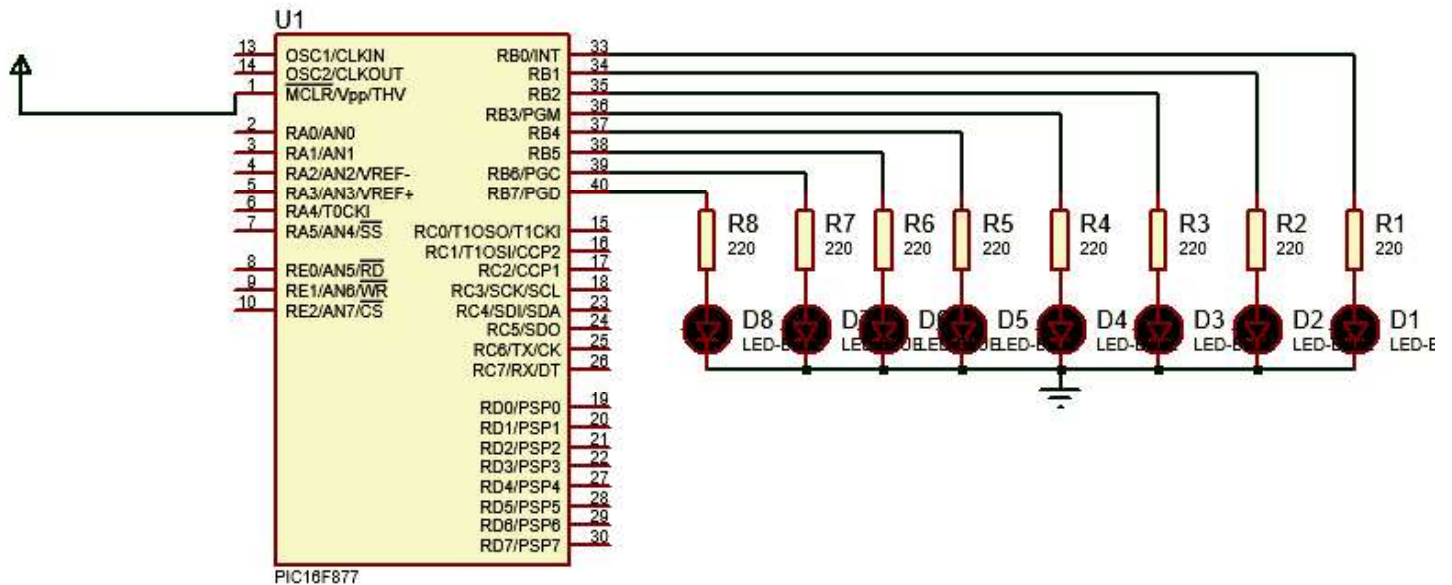


Figura 8.3 Circuito a implementar para la actividad 2

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

```
#include <16f877.h>
#fuses HS,NOPROTECT,
#use delay(clock=20000000)
#org 0x1F00, 0x1FFF void loader16F877(void) {}
```

```
int var1;
```

```
void main(){
    while(1){
        var1=input_a();
        output_b(var1);
    } //while
} //main
```

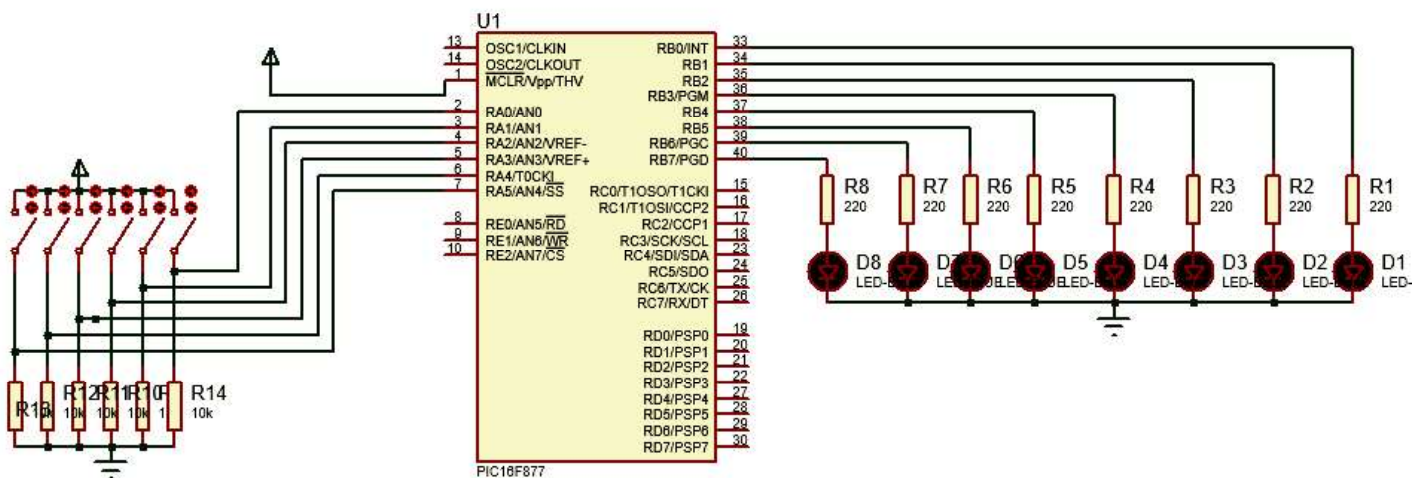


Figura 8.4 Circuito a implementar para la actividad 3

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

```
#include <16f877.h>
#fuses HS,NOPROTECT,
#use delay(clock=20000000)
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)
#org 0x1F00, 0x1FFF void loader16F877(void) {}

void main(){
    while(1){
        output_b(0xff); //
        printf(" Todos los bits encendidos \n\r");
        delay_ms(1000);
        output_b(0x00);
        printf(" Todos los leds apagados \n\r");
        delay_ms(1000);
    } //while
} //main
```

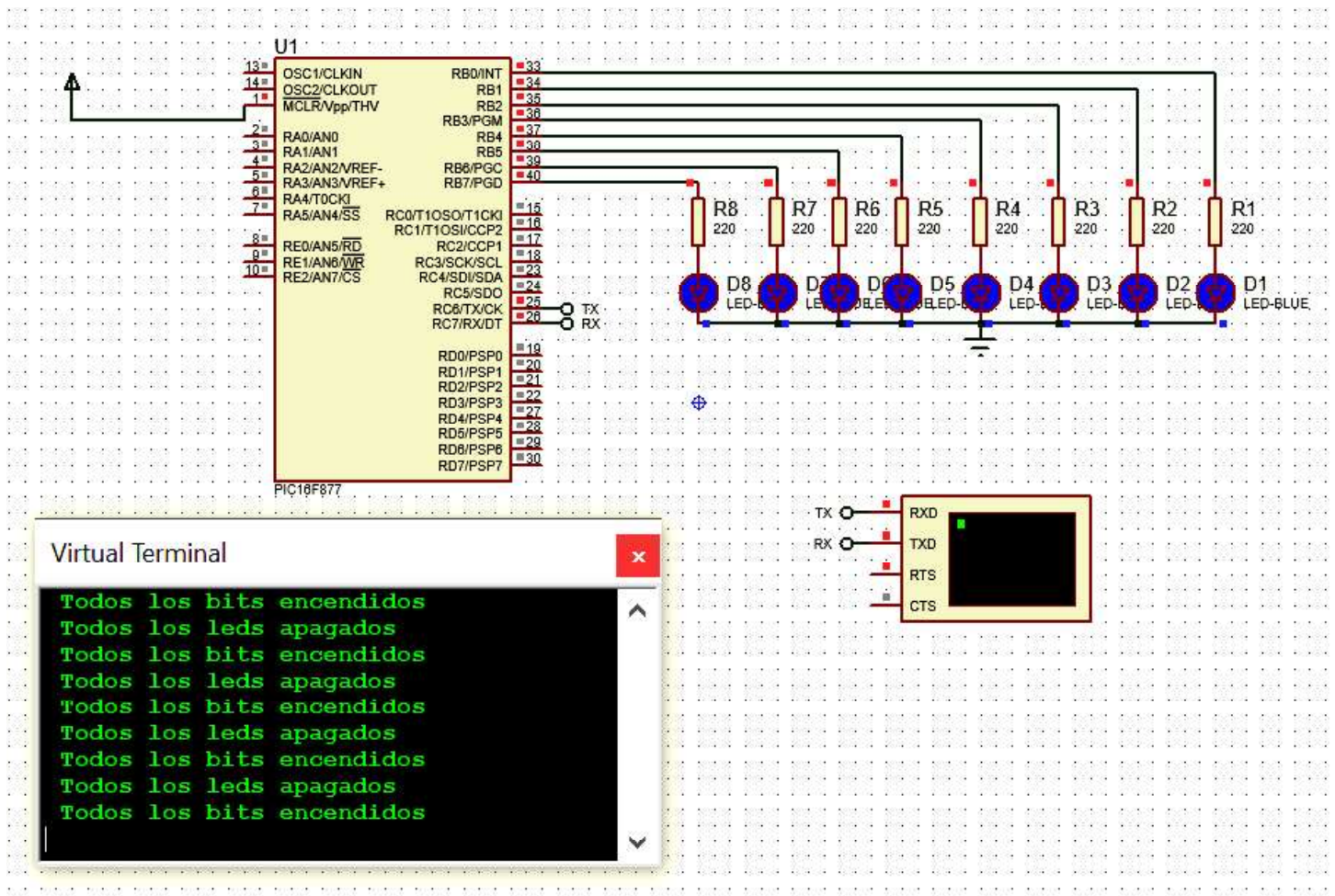


Figura 8.5 Circuito a implementar para la actividad 4

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

```
#include <16F877.h>
#fuses HS,NOWDT,NOPROTECT,NOLVP
#use delay(clock=20000000)
#define use_portb_lcd true
#include <lcd.c>

void main() {

    lcd_init();

    while( TRUE ) {
        lcd_gotoxy(1,1);
        printf(lcd_putc," UNAM \n ");
        lcd_gotoxy(1,2);
        printf(lcd_putc," FI \n ");
        delay_ms(300);
    }
}
```

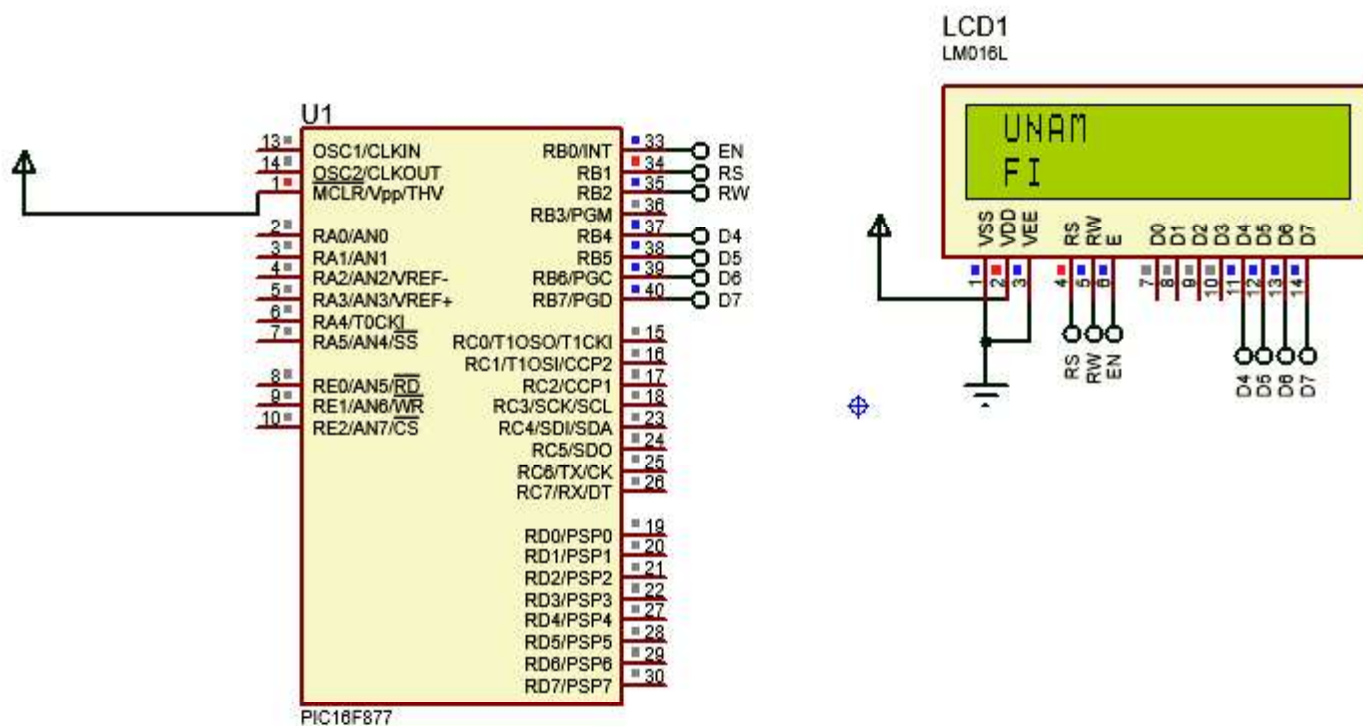


Figura 8.6 Circuito a implementar para la actividad 5

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.

DATO	ACCION Puerto B	Ejecución
0	Todos los bits apagados	00000000
1	Todos los bits encendidos	11111111
2	Corrimiento del bit más significativo hacia la derecha	10000000 00000001
3	Corrimiento del bit menos significativo hacia la izquierda	00000001 10000000
4	Corrimiento del bit más significativo hacia la derecha y a la izquierda	10000000 00000001 10000000
5	Apagar y encender todos los bits.	00000000 11111111

Tabla 8.1 Control a través del puerto serie

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:

- Primer línea y 5 columna; la cuenta en decimal
- Segunda línea y 5 columna; la cuenta en hexadecimal

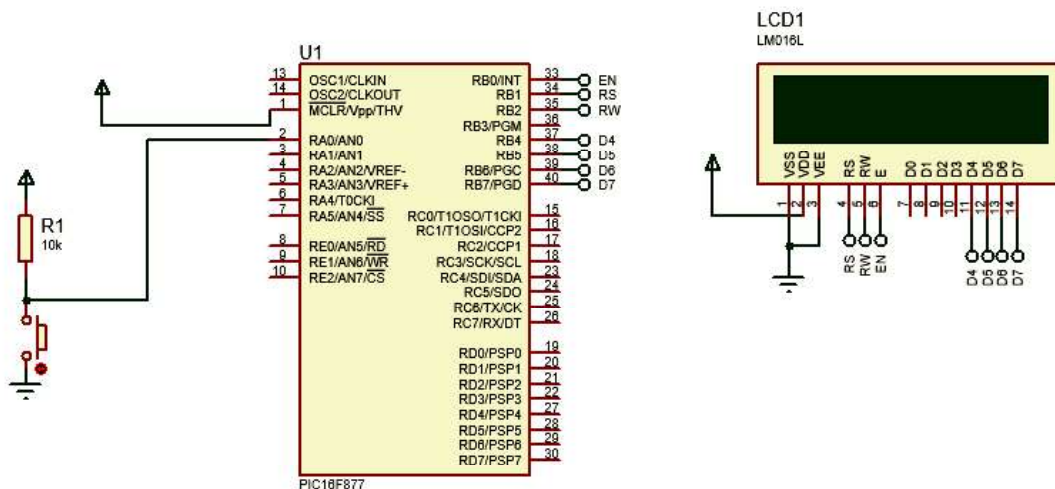


Figura 8.7 Circuito sugerido; actividad 7