



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO



FACULTAD DE INGENIERÍA

LABORATORIO DE MICROCOMPUTADORAS

GRUPO 04

ROJAS MÉNDEZ GABRIEL

314141712

CARREÓN GUZMÁN MARIANA IVETTE

312103914

**PRÁCTICA 9: PROGRAMACIÓN EN C
COMUNICACIÓN SERIE SÍNCRONA, I2C**

SEMESTRE 2022-1

26/11/2021



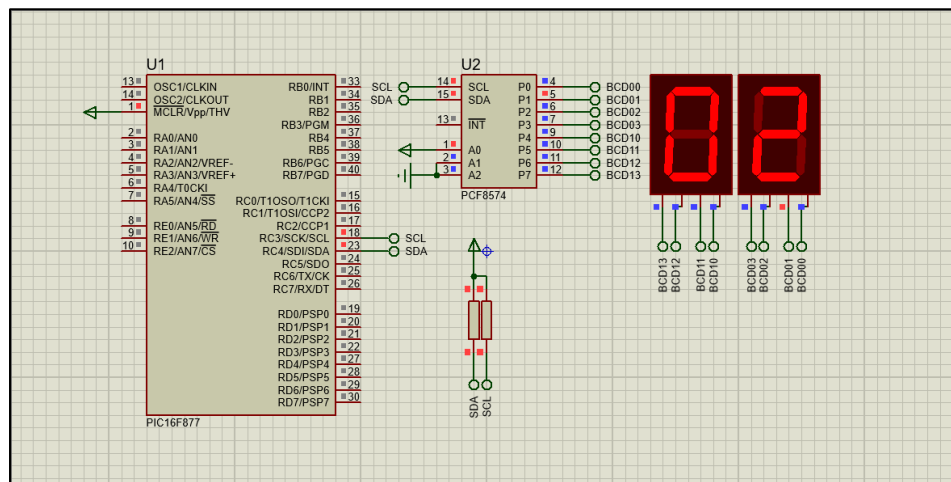
Objetivos:

El alumno experimentará y reforzará sus conocimientos sobre la comunicación serie síncrona en la modalidad de protocolo I2C, usará el circuito PCF8574 como expensor de puertos, conectado como esclavo para controlar diversos dispositivos.

Desarrollo:

1.- El objetivo del siguiente programa será para mayor comprensión de la comunicación I2C y la programación en C, por lo que se pide analizarlo y comentarlo para su reporte; observar en el circuito la conexión de A2, A1 y A0 para generar la dirección del esclavo, así como su uso en el programa.

```
Ejercicio 1.c Ejercicio 2.c Ejercicio 3.c Ejercicio 4.c*
1  #include <16F877.h>           //Librería del microcontrolador.
2  #fuses HS,NOWDT,NOPROTECT
3  #use delay(clock=2000000)     //Frecuencia del reloj.
4  #use i2c(MASTER, SDA=PIN_C4, SCL=PIN_C3,SLOW, NOFORCE_SW) //Configurando modo
5  int contador=0;               //modo esclavo.
6
7  void escribir_i2c(){
8      i2c_start();              //Inicia la comunicación I2C.
9      i2c_write(0x42);          //Modo escritura dirección
10     i2c_write(contador);      //Modo escritura valor
11     i2c_stop();               //Fin de la comunicación.
12 }
13
14 void main(){
15     while(true){              //Ciclo infinito
16         escribir_i2c();        //Llamada a función para escritura de datos
17         delay_ms(500);         //por comunicación I2C.
18         contador++;
19     }
20 }
```



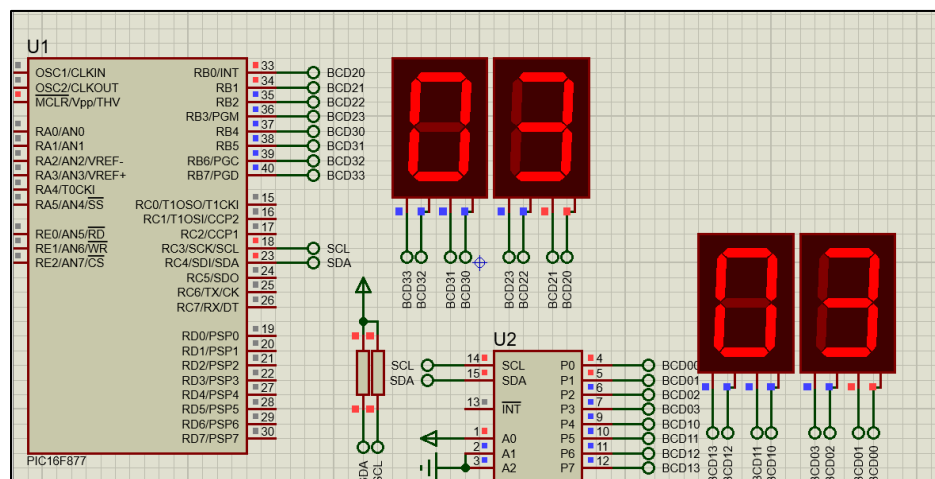


2.- Realizar la modificación al programa para que también muestre el contador en el puerto B.

```

Ejercicio 1.c  Ejercicio 2.c  Ejercicio 3.c  Ejercicio 4.c*
1  #include <16F877.h>           //Librería del microcontrolador.
2  #fuses HS,NOWDT,NOPROTECT
3  #use delay(clock=2000000)     //Frecuencia del reloj.
4  #org 0x1F00, 0x1FFF void loader16F877(void){}
5  #use i2c(MASTER, SDA=PIN_C4, SCL=PIN_C3,SLOW, NOFORCE_SW) //Configurando modo
6  int contador=0;              // esclavo.
7
8  void escribir_i2c(){
9      i2c_start();              //Inicia la comunicación I2C.
10     i2c_write(0x42);          //Modo escritura dirección.
11     i2c_write(contador);      //Modo escritura valor.
12     i2c_stop();               //Fin de la comunicación.
13 }
14
15 void main(){
16     while(true){              //Ciclo infinito
17         escribir_i2c();         //Llamada a función para escritura de datos
18         delay_ms(500);         //por comunicación I2C.
19         contador++;
20         output_b(contador);     //Se muestra el valor del contador en el
21     }                           //puerto B.
22 }

```

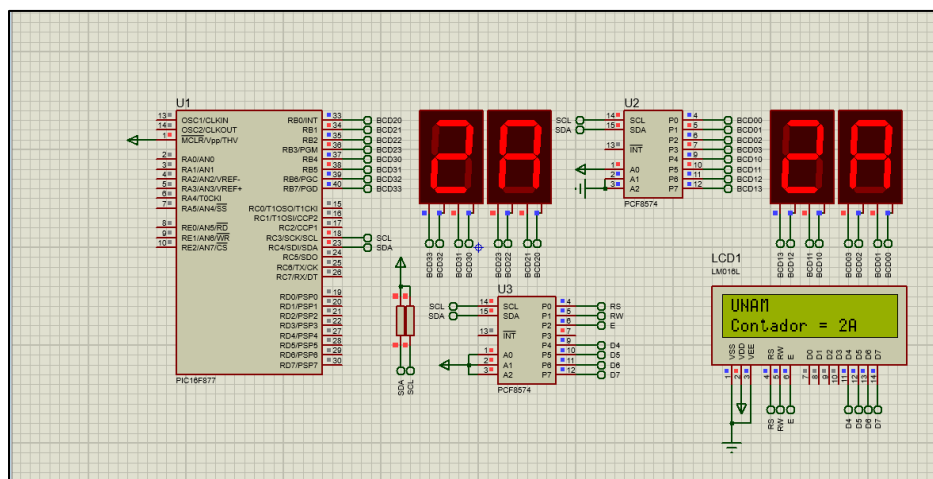


Para la realización de este ejercicio, solo fue necesario emplear la función `output_b()` vista anteriormente, para que de esta manera, el valor del contador también se viera reflejado en el puerto B, de la misma manera que en el esclavo.



3.- Realizar las modificaciones necesarias para que además de lo resuelto en el ejercicio previo, muestre el contador en un display LCD que funciona como esclavo.

```
Ejercicio 1.c Ejercicio 2.c Ejercicio 3.c Ejercicio 4.c
1  #include <16F877.h>           //Librería del microcontrolador.
2  #fuses HS,NOWDT,NOPROTECT,NOWDT,NOLVP
3  #use delay(clock=2000000)     //Frecuencia de reloj.
4  #org 0x1F00, 0x1FFF void loader16f877(void){}
5  #use i2c(MASTER, SDA=PIN_C4, SCL=PIN_C3,SLOW, NOFORCE_SW) //Configurando modo
6  #include <i2c_Flex_LCD.c>     //Librería para el uso del LCD | esclavo
7
8  int contador=0;               //Variable para llevar el conteo
9
10 void escribir_i2c(){
11     i2c_start();               //Inicia la comunicación I2C.
12     i2c_write(0x42);           //Modo escritura dirección.
13     i2c_write(contador);       //Modo escritura valor.
14     i2c_stop();               //Fin de la comunicación.
15 }
16
17 void main(){
18     lcd_init(0x4E,16,2);       //Inicializa el LCD y especifica su dirección
19     lcd_backlight_led(ON);     //Configura la iluminación de fondo.
20     while(true){
21         lcd_gotoxy(1,1);
22         printf(LCD_PUTC, "UNAM\n");
23         lcd_gotoxy(1,2);
24         printf(LCD_PUTC, "Contador = %X\n", contador);
25         escribir_i2c();         //Llamada a función para escritura de datos.
26         delay_ms(500);
27         contador++;
28         output_b(contador);     //Salida del conteo en el puerto B.
29     }
30 }
```

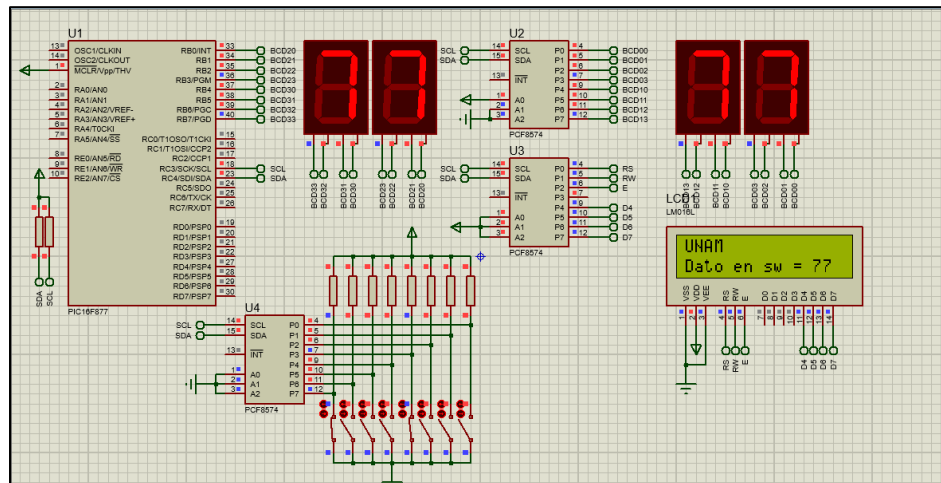
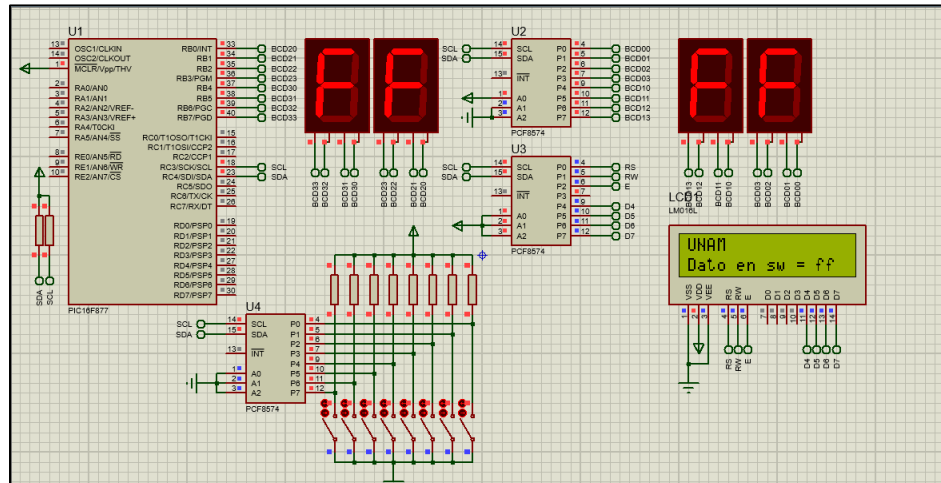


Para la realización de este ejercicio, nuevamente se trabajo sobre la base proporcionada anteriormente, y sólo se agrego la librería y sus funciones del LCD para la comunicación I2C.



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

```
Ejercicio 1.c Ejercicio 2.c Ejercicio 3.c Ejercicio 4.c
1  #include <16F877.h>           //Librería del microcontrolador.
2  #fuses HS,NOWDT,NOPROTECT,NOWDT,NOLVP
3  #use delay(clock=2000000)     //Frecuencia de reloj.
4  #org 0x1F00, 0x1FFF void loader16f877(void){}
5  #use i2c(MASTER, SDA=PIN_C4, SCL=PIN_C3,SLOW, NOFORCE_SW) //Configurando modo
6  #include <i2c_Flex_LCD.c>     //Librería para el uso del LCD |esclavo
7
8  int contador=0;               //Variable para llevar el conteo
9
10 void escribir_i2c(){
11     i2c_start();               //Inicia la comunicación I2C.
12     i2c_write(0x42);           //Modo escritura dirección.
13     i2c_write(contador);       //Modo escritura valor.
14     i2c_stop();               //Fin de la comunicación.
15 }
16
17 void leer_i2c(){               //Función para leer datos del esclavo.
18     i2c_start();               //Inicia la comunicación I2C.
19     i2c_write(0x41);           //Modo escritura dirección.
20     contador=i2c_read(0);       //Lectura de los datos del esclavo.
21     i2c_stop();               //Fin de la comunicación.
22 }
23
24 void main(){
25     lcd_init(0x4E,16,2);        //Inicializa el LCD y especifica su dirección
26     lcd_backlight_led(ON);      //Configura la iluminación de fondo.
27     while(true){
28         lcd_gotoxy(1,1);
29         printf(LCD_PUTC, "UNAM\n");
30         lcd_gotoxy(1,2);
31         printf(LCD_PUTC, "Dato en sw = %x\n", contador);
32         output_b(contador);
33         escribir_i2c();          //Llamada a función para escritura de datos.
34
35         delay_ms(500);           //Salida del conteo en el puerto B.
36         leer_i2c();              //Llamada a función para lectura de datos.
37         delay_ms(50);
38     }
}
```



Para la realización de este ejercicio, solo se implementó una nueva función de la cual se estableció el siguiente algoritmo:

- 1) Generar una función para lectura de datos de un esclavo por comunicación I2C.
- 2) Inicializar la comunicación I2C.
- 3) Establecer el modo de escritura en la dirección del esclavo que proporcionará los datos. B'0100 0001' → H'0x41'
- 4) Capturar los datos proporcionados del esclavo en una variable.
- 5) Finalizar la comunicación I2C.

De esta manera el ejercicio se completó de manera exitosa.



Conclusiones:

Carreón Guzmán Mariana En las actividades realizadas vi la forma en la que se puede controlar un dispositivo externo, en la primera actividad pude ver cómo es que funciona la comunicación I2C y la forma en la que se puede controlar a un esclavo, esto resulta importante ya que puede en las actividades siguientes modificar las actividades realizadas por él. Creo que esta práctica resulta importante porque se puede comprender la manera en la que se comunican el puerto PIC y los dispositivos externos.

Rojas Méndez Gabriel: Con la realización de esta práctica comprendí una manera muy fácil de poder emplear un puerto del PIC para poder comunicarse con distintos dispositivos y esto gracias a la comunicación I2C, que desde mi perspectiva es muy sencilla y muy útil, además, que las funciones establecidas para este tipo de comunicación, son muy intuitivas.