FACULTAD DE INGENIERÍA, UNAM

LABORATORIO DE MICROCOMPUTADORASS

SEMESTRE 2023-2

GRUPO 11

PREVIO PRÁCTICA 9

COMUNICACIÓN SERIE SÍNCRONA, I2C

NOMBRE DEL ALUMNO:

ARRIAGA MEJÍA JOSÉ CARLOS

PROFESOR

ING. ROMAN V. OSORIO COMPARAN

FECHA DE ENTREGA: **19 DE MAYO DE 2023** **CALIFICACION**

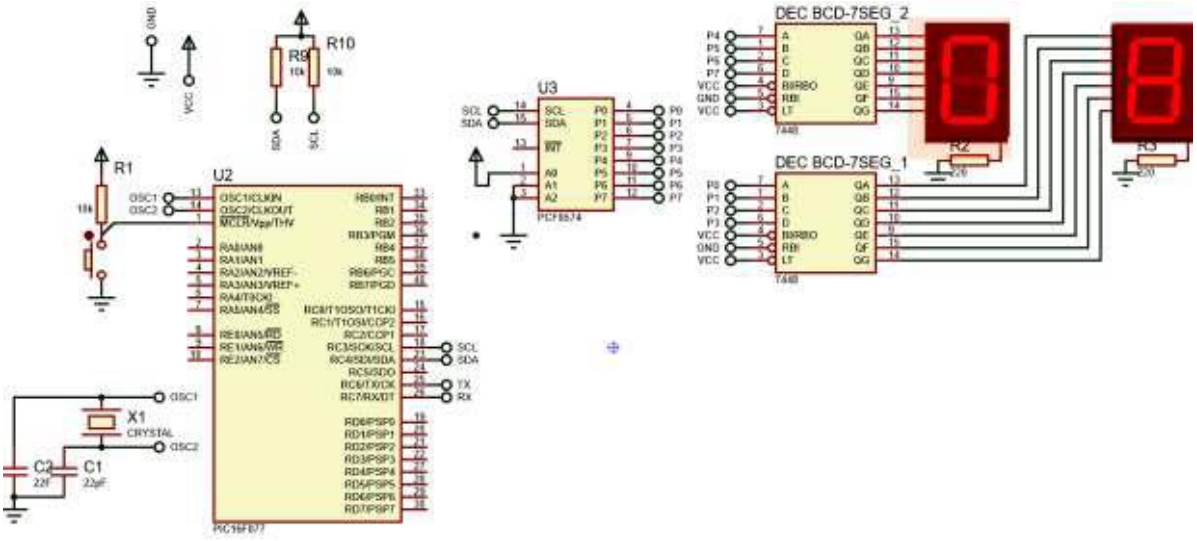
Objetivo

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

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.

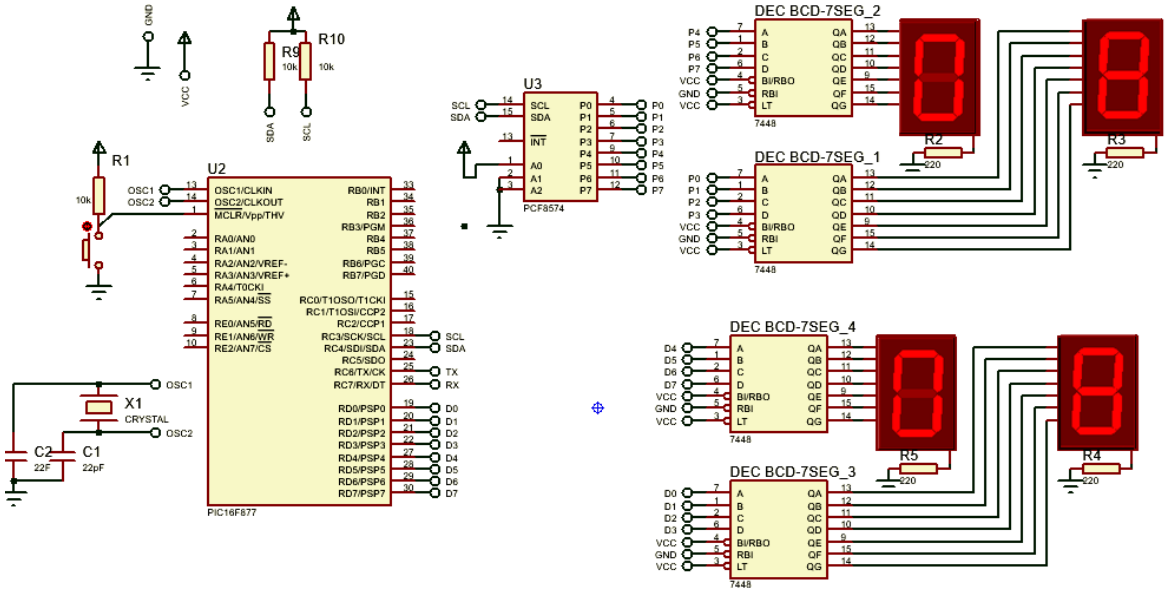
```
#include <16F877.h> // biblioteca del microprocesador
#fuses HS,NOWDT,NOPROTECT // conexiones eléctricas
#use delay(clock=20000000) // 20 MHz
#use i2c(MASTER, SDA=PIN_C4, SCL=PIN_C3,SLOW, NOFORCE_SW) // configuración
comunicación síncrona I2C
int contador=0; // variable contadora para aumenta
// función para escribir por medio de i2c
void escribir_i2c() {
      i2c_start(); // inicia comunicación i2c
      i2c_write(0x42); // escribe 0x42 = 0100001 0 --> 0100001 = 33 dir, 0 =
      Escritura
      i2c_write(contador); // escribe la variable global contador
      i2c_stop(); // finaliza comunicación i2c
}
//el programa es un cronómetro de medio segundo
void main() {
      while(true) {
            escribir_i2c(); // escribe en consola
            delay_ms(500); // retardo de medio segundo
            contador++; // aumenta valor de contador
      }
}
```

El circuito se describe a continuación: Considerar que se usa un decodificador BCD-7SEG.



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

Nota: Considerar que se usar un decodificador BCD-7SEG; por lo que deberá modificar el programa para ver la cuenta continua.




```
#include <16F877.h> // biblioteca del microprocesador
#fuses HS,NOWDT,NOPROTECT // conexiones eléctricas
#use delay(clock=20000000) // 20 MHz
#use i2c(MASTER, SDA=PIN_C4, SCL=PIN_C3,SLOW, NOFORCE_SW) // configuración
comunicación síncrona I2C
int contador=0; // variable contadora para aumenta
// función para escribir por medio de i2c
void escribir_i2c() {
      i2c_start(); // inicia comunicación i2c
      i2c_write(0x42); //escribe 0x42=0100001 0 --> 0100001=33 dir,
      0=Escritura
      i2c_write(contador); // escribe la variable global contador
      i2c_stop(); // finaliza comunicación i2c
}
// función para escribir a través de puerto paralelo
void escribir_puerto() {
      output_d(contador);
}
//el programa es un cronometro de medio segundo
void main() {
      while(true) {
            escribir_i2c(); // escribe en consola
            escribir_puerto(); // escribe por puerto b
            delay_ms(500); // retardo de medio segundo
            contador++; // aumenta valor de contador
      }
}
```

3.- Realizar las modificaciones necesarias para que además de lo resuelto en el ejercicio previo, muestre el contador en un display LCD que funcionará como esclavo I2C.

Consideraciones:

a. Debe incluir la librería i2c_LCD.c a su programa; esta librería contiene el protocolo de comunicación I2C para uso del Display de Cristal Líquido LCD.

b. La biblioteca I2C_LCD permite emplear con los mismos nombres las funciones empleadas para el LCD en formato paralelo.

c. Es necesario incluir la función que configura la forma de comunicación I2C del LCD.

a. lcd_init(DIRECCION_ESCLAVO,COLUMNAS,RENGLONES);

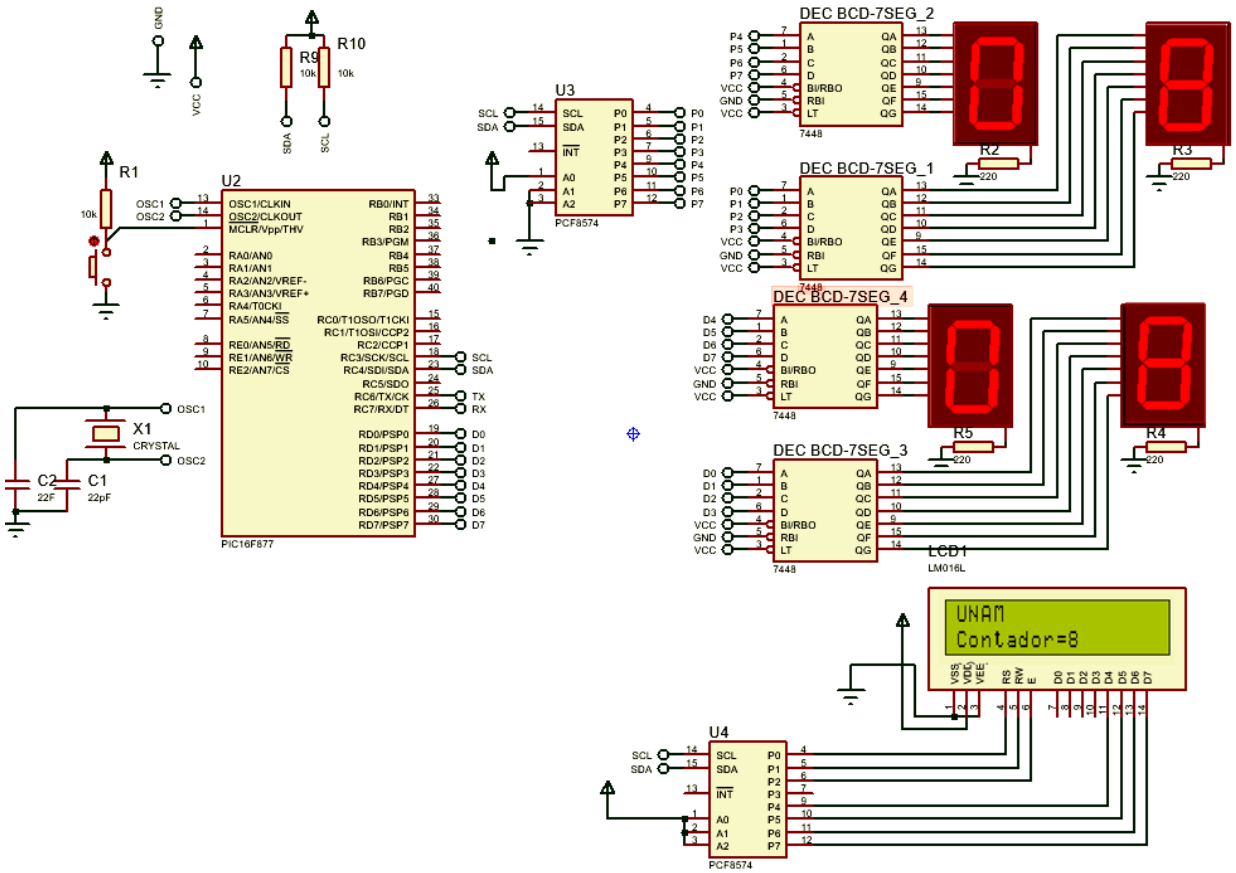
i. DIRECCION_ESCLAVO: es la dirección configurada por los valores fijos de fabrica y los definidos por hardware del módulo PCF8574 que controla al LCD; ubicar en el esquemático la configuración, para obtener la dirección.

ii. COLUMNAS: es la cantidad de columnas de LCD

iii. RENGLONES: es la cantidad de filas disponibles en el LCD

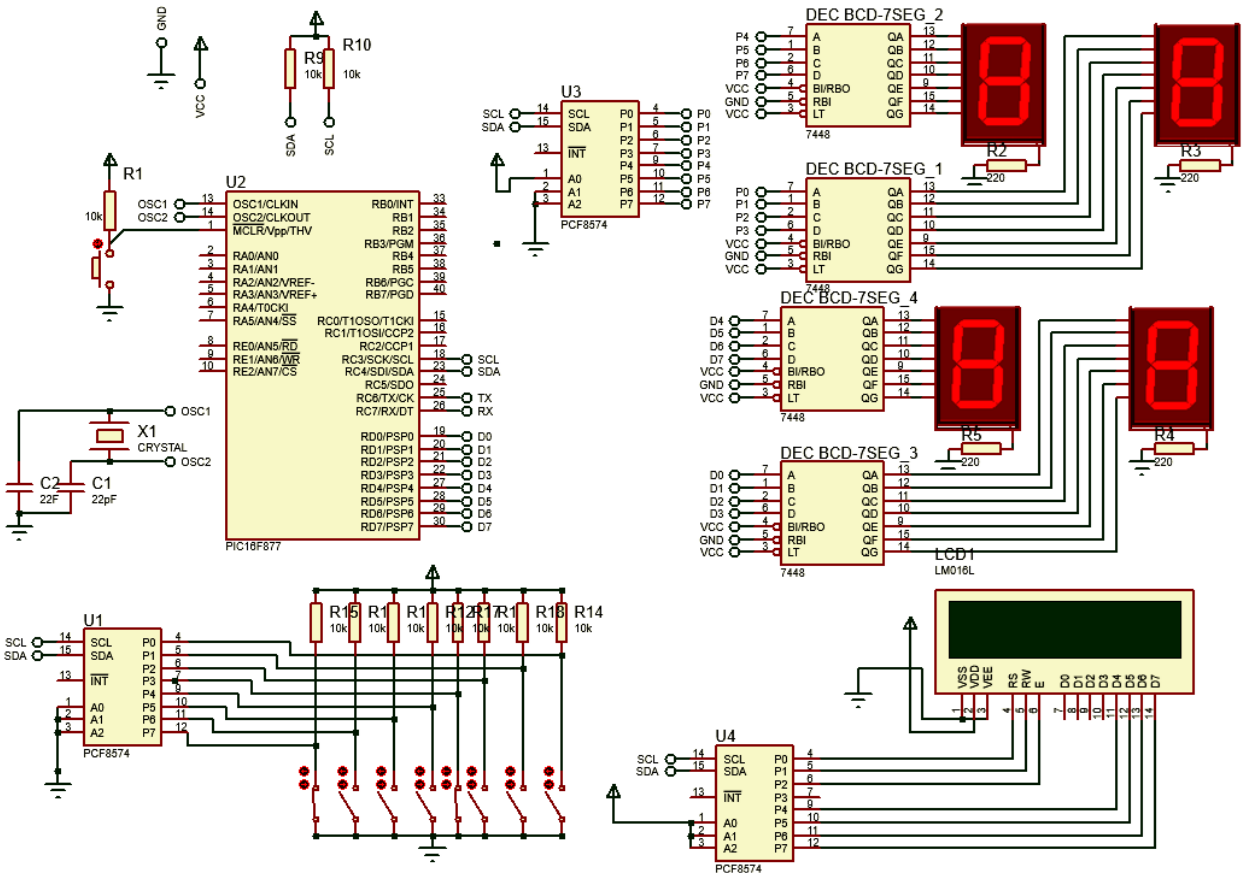
iv. En la practica se usará LCD de 16x2

d. Uso DEC BCD-7SEG



```
#include <16F877.h> // biblioteca del microprocesador
#fuses HS,NOWDT,NOPROTECT // conexiones eléctricas
#use delay(clock=20000000) // 20 MHz
#use i2c(MASTER, SDA=PIN_C4, SCL=PIN_C3,SLOW, NOFORCE_SW) // configuración
comunicación síncrona I2C
#include <i2c_LCD.c> // biblioteca para control de display LCD.
int contador=0; // variable contadora para aumenta
// función para escribir por medio de i2c
void escribir_i2c() {
      i2c_start(); // inicia comunicación i2c
      i2c_write(0x42); //escribe 0x42=0100001 0 --> 0100001=33 dir,
      0=Escritura
      i2c_write(contador); // escribe la variable global contador
      i2c_stop(); // finaliza comunicación i2c
}
// función para escribir a través de puerto paralelo
void escribir_puerto() {
      output_d(contador);
}
// función para escribir en el display lcd
void escribir_lcd() {
      lcd_gotoxy(11,2); //posiciona cursor en (11,2)
      printf(lcd_putc, "%d", contador);
}
//el programa es un cronometro de medio segundo
void main() {
      lcd_init(78,16,2); // inicialización display lcd --> dir = 0100111 = 39
      lcd_gotoxy(1,1); // posiciona cursor en 1,1
      printf(lcd_putc,"UNAM\n"); //escribe en display
      lcd_gotoxy(1,2); // posiciona cursor en 1,2
      printf(lcd_putc,"Contador="); // escribe en display
      while(true) {
            escribir_i2c(); // escribe en consola
            escribir_puerto(); // escribe por puerto d
            escribir_lcd(); // escribe en el lcd
            delay_ms(500); // retardo de medio segundo
            contador++; // aumenta valor de contador
      }
}
```

4.- Realizar un programa de tal forma que obtenga la lectura de la entrada generada por otro dispositivo esclavo y los muestre en los tres periféricos usados en la actividad 3.




```
#include <16F877.h> // biblioteca del microprocesador
#fuses HS,NOWDT,NOPROTECT // conexiones eléctricas
#use delay(clock=20000000) // 20 MHz
#use i2c(MASTER, SDA=PIN_C4, SCL=PIN_C3,SLOW, NOFORCE_SW) // configuración
comunicación síncrona I2C
#include <i2c_LCD.c> // biblioteca para control de display LCD.
int contador=0; // variable contadora para aumenta
// funcion para escribir por medio de i2c
void escribir_i2c() {
      i2c_start(); // inicia comunicación i2c
      i2c_write(0x42); //escribe 0x42=01000010 --> 0100001=33 dir, 0 =
      Escritura
      i2c_write(contador); // escribe la variable global contador
      i2c_stop(); // finaliza comunicación i2c
}
// funcion para escribir a través de puerto paralelo
void escribir_puerto() {
      output_d(contador);
}
// función para escribir en el display lcd
void escribir_lcd() {
      lcd_gotoxy(11,2); //posiciona cursor en (11,2)
      printf(lcd_putc, "%d", contador);
}
```

```
// función para leer de dispositivo i2c
void leer_i2c() {
      i2c_start(); // inicia comunicación i2c
      i2c_write(0x41); // excribe 0x41 = 0100000 1 --> 0100000 = 32, 1 =
      Lectura
      contador = i2c_read(0);
      i2c_stop(); // finaliza comunicación I2C
}
//el programa es un cronometro de medio segundo
      void main() {
      lcd_init(78,16,2); // inicialización display lcd --> dir = 0100111 = 39
      lcd_gotoxy(1,1); // posiciona cursor en 1,1
      printf(lcd_putc,"UNAM\n"); //escribe en display
      lcd_gotoxy(1,2); // posiciona cursor en 1,2
      printf(lcd_putc,"Contador= "); // escribe en display
      while(true) {
            leer_i2c(); // lee valor
            escribir_i2c(); // escribe en consola
            escribir_puerto(); // escribe por puerto b
            escribir_lcd();
            delay_ms(100); // retardo de medio segundo
      }
}
```