Universidad Nacional Autónoma de México

Facultad de Ingeniería

Semestre 2023-2

Laboratorio de Microcomputadoras

Práctica 9:

Programación en C

Comunicación serie síncrona, I2C

Profesor:

M.I. Ruben Anaya García

Alumnos:	Número de cuenta:
Barreiro Valdez Alejandro	317520888
Zepeda Baeza Jessica	317520747

Grupo Laboratorio: 4

Grupo Teoría: 1

Fecha de realización: 24 de mayo de 2023

Fecha de entrega: 30 de mayo de 2023

Desarrollo

En esta práctica se utilizará la comunicación serie síncrona utilizando el protocolo I2C utilizando el circuito PCF8574 como un expansor de puertos. El bus I2C se utilizó para conectar un microcontrolador a diversos periféricos. Actualmente es utilizado como protocolo para comunicar un microcontrolador y dispositivos externos. En este protocolo se utilizan dos líneas para transferencia y recepción: SDA (datos bidireccional) y SCL (sincronización). En esta práctica se utilizará el modelo del maestro esclavo para generar la transferencia de datos y la lectura de datos entre un maestro y un esclavo. Se revisaron las diferentes banderas que se utilizan con este fin. Además, se revisaron las diferentes funciones que se incluyen en la directiva I2C para la programación en C. Se puede configurar el programa para estar en modo maestro o esclavo, para configurar los pines de SDA y SCL, y elegir la velocidad. Por último, se analizaron las direcciones de los dispositivos que se configuran para su uso. Con toda esta información se realizaron los diferentes programas que utilizan la comunicación serie síncrona.

Los programas desarrollados fueron:

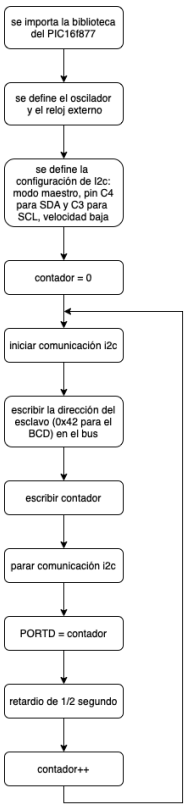
1. Un contador en BCD de 7 segmentos utilizando el protocolo I2C.
2. Un contador donde se muestran dos contadores en dos BCD de 7 segmentos.
3. Agregar al programa anterior la escritura en LCD utilizando I2C.
4. Agregar una última funcionalidad al programa anterior donde se agrega un esclavo que sirve como entrada para que las demás salidas muestren ese número.

Algoritmos

Ejercicio 2

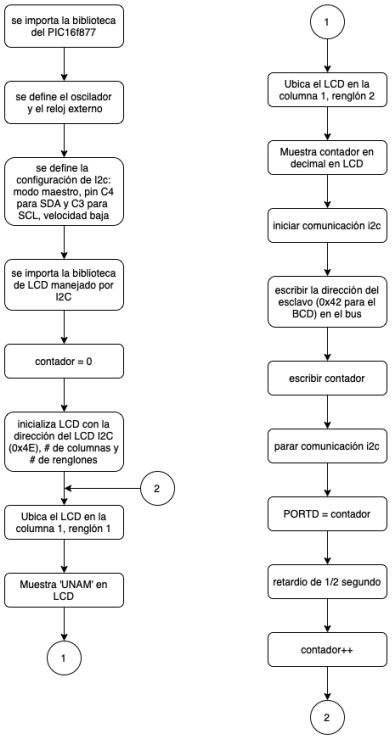
Para el ejercicio 2 se tomó como base el código proporcionado por el ejercicio 1. Lo único que se modificó fue agregar como salida en el Puerto D el valor de CONTADOR de manera que se pueda visualizar en los otros dos decodificadores BCD-7SEG.

Por lo tanto, también se incluye la biblioteca del PIC16F877, se configura el reloj externo y el oscilador así como los puertos para el I2C y su modo maestro. Se inicializa el contador en 0 y se hacen los pasos para la escritura de un dato por medio de I2C. Esto incluye iniciar la comunicación, definir la dirección del bus de donde se escribirá el dato, escribir el dato (en este caso CONTADOR) y parar la comunicación. Una vez, que se escribió se manda el mismo dato al Puerto D, se realiza un retardo de medio segundo y se incrementa CONTADOR. Después el programa vuelve a ejecutar desde la escritura mediante I2C, de manera que se muestra el nuevo valor de CONTADOR de manera infinita. La carta ASM realizada para este ejercicio es la siguiente:



Ejercicio 3

Para el ejercicio 3 se realizó la siguiente carta ASM:



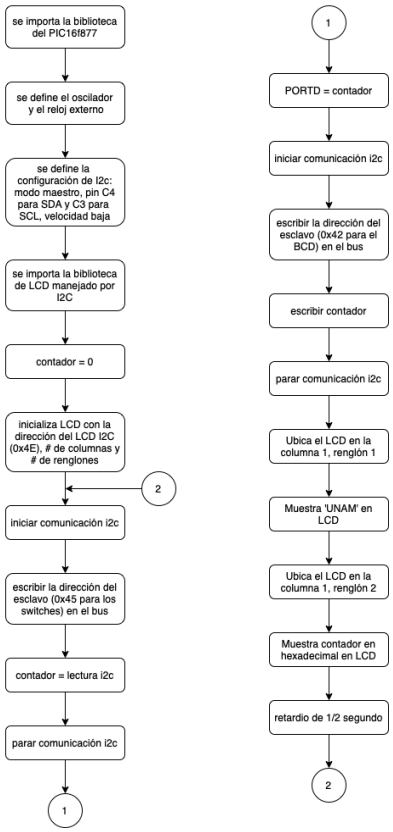
De igual forma, se toma como base el ejercicio anterior pero se agregan instrucciones para manejar un LCD mediante comunicación I2C. Lo primero que se agrega después de las

configuraciones del microcontrolador, es la librería i2c_LCD.c que permite implementar el protocolo de comunicación I2C para el LCD. Lo siguiente, después de inicializar el contador, es inicializar el LCD incluyendo tanto su dirección de esclavo como el número de columnas y de renglones. Y por último, se ubica el LCD en el renglón y columna donde se quiere escribir para escribir el dato. En este caso, se mando “UNAM” en el primer renglón y en el segundo “Contador = [valor del contador en decimal]”.

Después de escribir en el LCD se hace lo mismo que en el código anterior donde se muestra CONTADOR en los dos decodificadores BCD usando tanto el Puerto D como otro esclavo de I2C. Una vez que termina de mostrar CONTADOR, éste se incrementa y regresa a ubicar el LCD en el renglón 1, columna 1 para repetir el proceso de mostrarlo en distintos dispositivos.

Ejercicio 4

Para el último ejercicio se realizó la siguiente carta ASM:



Otra vez, se tomó como base el ejercicio anterior pero se agregaron instrucciones para que CONTADOR sea definido como una lectura de un dispositivo esclavo de I2C. LAs configuraciones se quedaron igual. Se agregaron instrucciones para hacer la lectura del dato antes de mostrarlo. Primero se inicia la comunicación I2C, luego se indica la dirección del esclavo que se leerá. Después se lee el dato y se guarda CONTADOR. Por último, se para la comunicación I2C. Lo demás es mandar el dato CONTADOR al Puerto D, al BCD

esclavo de I2C y al LCD también esclavo de I2C en formato hexadecimal, como en los demás ejercicios. Se llama el retardo de medio segundo al final y el flujo regresa a las instrucciones que leen el dato para poder obtener un nuevo valor en CONTADOR y mostrarlo.

Programas comentados

Ejercicio 1

```
#include <16F877.h>
#fuses HS,NOWDT,NOPROTECT
#use delay(clock=20000000)
#use i2c(MASTER, SDA=PIN_C4, SCL=PIN_C3,SLOW, NOFORCE_SW)
/*
Directiva para protocolo I2C, se configura:
-Modo maestro
-SDA en Pin C4
-CSL en Pin C3
-Velocidad baja
*/

int contador=0;   |             //Variable de contador

void escribir_i2c(){
   i2c_start();                 //Inicializa comunicación
   i2c_write(0x42);             //Localiza el dispositivo
   i2c_write(contador);         //Escribe el contador
   i2c_stop();                  //Termina I2C
    }

void main()
{
    while(true)
   {
    escribir_i2c();             //Se escribe el contador
    delay_ms(500);              //Retardo de 500 ms
    contador++;                 //Suma uno al contador
   }
}
```

Ejercicio 2

```
#include <16F877.h>
#fuses HS,NOWDT,NOPROTECT
#use delay(clock=20000000)
#use i2c(MASTER, SDA=PIN_C4, SCL=PIN_C3,SLOW, NOFORCE_SW)
/*
Directiva para protocolo I2C, se configura:
-Modo maestro
-SDA en Pin C4
-CSL en Pin C3
-Velocidad baja
*/

int contador=0;             //Variable de contador

void escribir_i2c(){
   i2c_start();             //Inicia comunicación
   i2c_write(0x42);         //Localiza el dispositivo
   i2c_write(contador);     //Escribe el contador
   i2c_stop();              //Termina I2C
    }

void main()
{
    while(true)
   {
    escribir_i2c();         //Se escribe el contador
    output_d(contador);     //Puerto D = contador
    delay_ms(500);          //Retardo de 500ms
    contador++;             //Suma uno al contador
   }
}
```

Ejercicio 3

```c
#include <16F877.h>
#fuses HS,NOWDT,NOPROTECT
#use delay(clock=20000000)
#use i2c(MASTER, SDA=PIN_C4, SCL=PIN_C3,SLOW, NOFORCE_SW)
/*
Directiva para protocolo I2C, se configura:
-Modo maestro
-SDA en Pin C4
-CSL en Pin C3
-Velocidad baja
*/
#include <i2c_LCD.c> //Utilizado para el protocolo entre LCD e I2C

int contador=0;                                     //Variable de contador

void escribir_i2c(){
   i2c_start();                                     //Inicia comunicación
   i2c_write(0x42);                                 //Localiza el dispositivo
   i2c_write(contador);                             //Escribe el contador
   i2c_stop();                                      //Termina I2C
   }

void main()
{
   lcd_init(0x4E, 16, 2);                           //Inicialización LCD con dirección
   while(true)
   {
    lcd_gotoxy(1,1);                                //Colocarse en renglón 1, columna 1
    printf(lcd_putc, "UNAM");                       //Imprimir "UNAM"
    lcd_gotoxy(1,2);                                //Colocarse en renglón 2, columna 1
    printf(lcd_putc, "Contador=%d", contador);      //Imprimir "Contador = numero"
    escribir_i2c();                                 //Se escribe el contador
    output_d(contador);                             //Puerto D = contador
    delay_ms(500);                                  //Retardo de 500ms
    contador++;                                     //Suma uno al contador
   }
}
```

Ejercicio 4

```c
#include <16F877.h>
#fuses HS,NOWDT,NOPROTECT
#use delay(clock=20000000)
#use i2c(MASTER, SDA=PIN_C4, SCL=PIN_C3,SLOW, NOFORCE_SW)
/*
Directiva para protocolo I2C, se configura:
-Modo maestro
-SDA en Pin C4
-CSL en Pin C3
-Velocidad baja
*/
#include <i2c_LCD.c> //Utilizado para el protocolo entre LCD e I2C

int contador=0;                                     //Variable de contador

void escribir_i2c(){
   i2c_start();                                     //Inicia comunicación
   i2c_write(0x42);                                 //Dirección dispositivo escritura
   i2c_write(contador);                             //Escribir el contador
   i2c_stop();                                      //Finalizar escritura
   }

void leer_i2c(){
   i2c_start();                                     //Inicia comunicación
   i2c_write(0x45);                                 //Dirección dispositivo lectura
   contador = i2c_read(0);                          //Lectura del contador
   i2c_stop();                                      //Finalizar lectura
   }

void main()
{
   lcd_init(0x4E, 16, 2);                           //Inicializar el LCD en dirección 4E
   while(true)
   {
    leer_i2c();                                     //Lectura de contador
    output_d(contador);                             //Puerto D = contador
    escribir_i2c();                                 //Escritura de contador
    lcd_gotoxy(1,1);                                //Posicionarse en columna 1, renglón 1
    printf(lcd_putc, "UNAM");                       //Escritura de UNAM
    lcd_gotoxy(1,2);                                //Posicionarse en columna 1, renglón 2
    printf(lcd_putc, "Contador=%X", contador);      //Escritura de "Contador = HEX"
    delay_ms(500);                                  //Retardo de 500ms
   }
}
```

Conclusiones y/o comentarios

Zepeda Baeza Jessica:

Esta práctica nos permitió observar el funcionamiento del protocolo I2C para la lectura y escritura de datos en distintos dispositivos esclavos. La ventaja de esto fue que anteriormente los dispositivos utilizados, como Switches, decodificadores BCD-7SEG y LCDs, fueron conectados directamente a los puertos paralelos y esta práctica nos permitió entender cómo es posible manejar todos en conjunto mediante un expansor de puertos y el protocolo I2C. Esto último es mucho más eficiente en dispositivos como el LCD donde de pasar a conectar 14 terminales ahora sólo es necesario conectar 6. Cada ejercicio de la práctica tomó como base el ejercicio anterior y fue agregando pequeñas funcionalidades del I2C para crear al final un sistema mucho más complejo. Además se aprendieron los pasos necesarios tanto para la lectura como escritura desde un maestro hacia un esclavo.

Barreiro Valdez Alejandro:

En esta práctica se pudo utilizar el protocolo I2C para la comunicación en serie síncrona para la ampliación de los dispositivos periféricos que puede tener el microcontrolador PIC16F877A. En esta práctica se configuró este protocolo para la escritura y la lectura de diferentes dispositivos esclavos a partir de uno maestro. De esta manera se aprendió a realizar esta tarea utilizando una biblioteca en C y los diferentes algoritmos que se pueden seguir para la lectura y la escritura de datos por este protocolo. Lo primero que se realizó en la práctica fue la escritura donde se aprendió a iniciar la comunicación, a especificar la dirección de los dispositivos y a indicar la escritura. Posteriormente, se agregaron otros periféricos de salida como otros BCD de siete segmentos y el display de cristal líquido. Estos elementos se configuran utilizando C y se alambraron utilizando el circuito PCF8574 que permite la entrada y salida de datos a través de SDA y SCL. Por último, se configuró una entrada para que los dispositivos la utilizaran para reflejar su salida a partir de esa entrada. Con todos estos dispositivos se pudo entender cómo funciona el protocolo I2C y cómo puede ser muy útil para la ampliación de puertos en un microcontrolador.