

Laboratorio de Microcomputadoras
Práctica No. 9
Programación en C
Comunicación serie síncrona, I2C

Objetivo: 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.

Introducción

El bus I2C o I²C (Inter-Integrated Circuit) fué desarrollado por Philips, su propósito original fue conectar un microcontrolador de un televisor a diversos periféricos. Se ha adoptado como protocolo de comunicación serial que permite transferencia de datos entre un microcontrolador y dispositivos externos. La tasa de transferencia estándar es de 100Kbps y la más alta puede alcanzar los 3.4 Mbps.

Utiliza dos líneas para la transferencia y recepción de datos:

- a. **SDA** (Serial **D**Ata Line); línea de datos bidireccional.
- b. **SCL** (Serial **C**Lock Line); línea de sincronización ó señal de reloj.

Los dispositivos conectados a estas líneas son de tipo “colector abierto”, por lo tanto debe de conectar resistencias de *pull-up* de 10 K Ω , así como tener tierras comunes para establecer las mismas referencias entre todos los dispositivos.

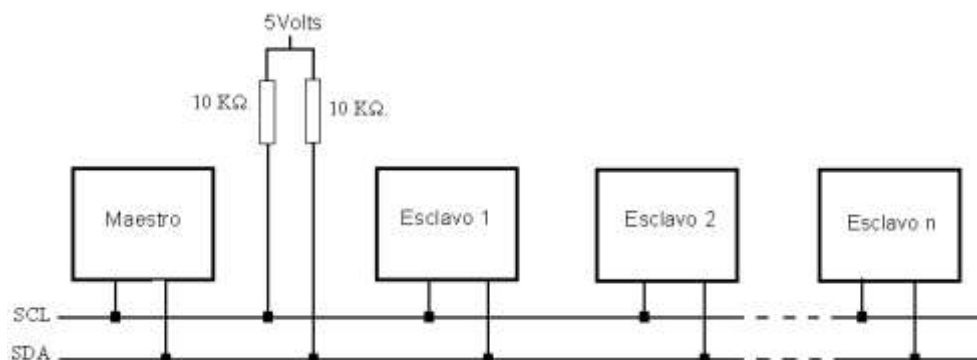


Figura 8.1 Comunicación I2C

Se puede establecer comunicación entre un maestro y uno o varios esclavos, así como en modo multi-maestro. Cada uno de los dispositivos tiene una dirección única, en general el protocolo consta de las siguientes etapas:

- a. Señal de inicio **START**
- b. Selección del dispositivo esclavo **ADDRESS**
- c. Indica acción a realizar lectura o escritura **R/W**
- d. Respuesta del esclavo **ACK**
- e. Envío/recepción de información **DATA**
- f. Señal de paro **STOP**

S	ADDRESS	R/W	ACK	DATA	A	P
	Maestro		Esclavo	Maestro	Esclavo	Maestro

Si R/W= '0' el esclavo escribirá; por lo tanto, el maestro envía información.

Si R/W= '1' el esclavo será leído; por lo tanto, el maestro solicita información.

Transferencia de datos del maestro al esclavo

El maestro escribirá información al esclavo, la transferencia de datos se describe a continuación:

Transferencia de un dato:

S	ADDRESS	0	ACK	DATA	A	P
	Maestro		Esclavo	Maestro	Esclavo	Maestro

El algoritmo a programar será:

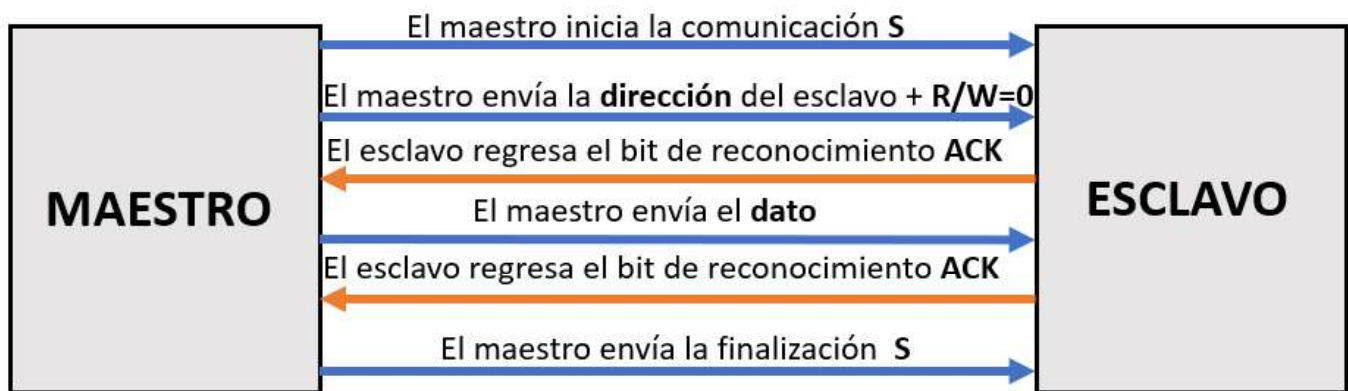


Figura 8.2 Algoritmo I2C

El maestro podrá continuar transferir información, previo al envía del bit de paro S.

Transferencia múltiple:

S	DIR_ESCLAVO	0	A	Dato	A	Dato	A		Dato	A	P
	Maestro		E	M	E	M	E		M	E	M

Lectura de datos del esclavo

En este caso, el maestro solicitará información al esclavo; el maestro responderá con el bit de reconocimiento cada que reciba información del esclavo a excepción del último dato, en el cuál responderá con un NACK.

Lectura de un dato:

S	ADDRESS	1	ACK	DATA	N	P
Maestro			Esclavo		Maestro	

Lectura múltiple de datos:

S	ADDRESS	1	A	Dato	A	Dato	A	Dato	N	P
Maestro				E	M	E	M	E		M

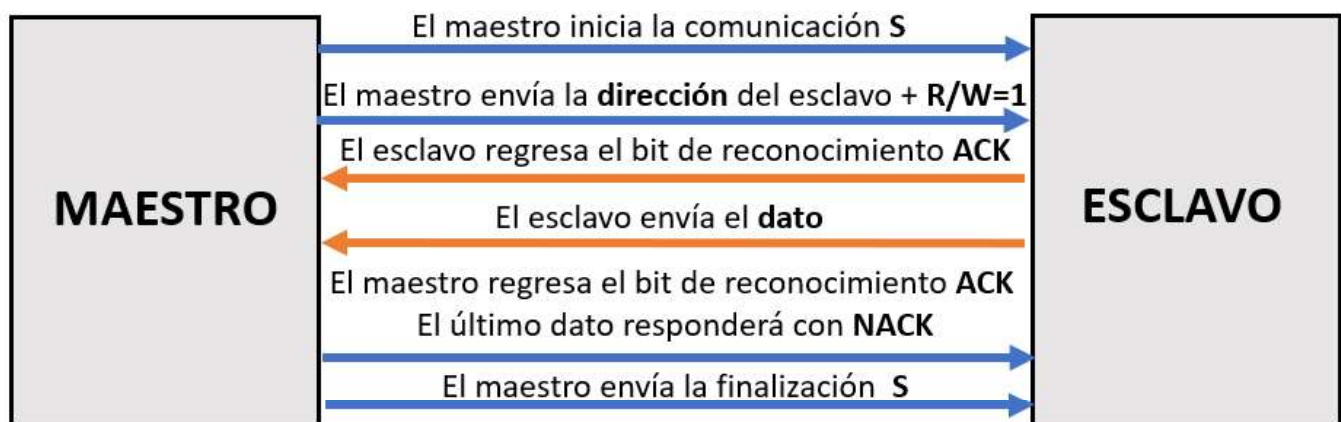


Figura 8.3 Algoritmo para lectura de datos del maestro

La dirección del esclavo podrá ser de 7 0 10 bits, es importante consultar la hoja técnica del dispositivo con el que se desee establecer la comunicación; en la práctica se usará el formato descrito previamente (dirección de 7 bits).

Comunicación I2C en el PIC

El módulo MSSP del PIC16F877(A) tiene seis registros para la operación del I2C. Asigna en las terminales: **RC3/SCK/SCL** para la señal de reloj y **RC4/SDI/SDA** para los datos; estos son:

- Registro de control SSPCON
- Registro de control 2 SSPCON2
- Registro de estado SSPSTAT
- Registro de transmisión / recepción SSPBUF
- Registro de corrimiento SSPSR
- Registro de dirección SSPADD

Se describen brevemente, consultar la hoja técnica para mayor información.

Registro de estado 94h

SSPSTAT	SMP	CKE	D/A	P	S	R/W	UA	BF
---------	-----	-----	-----	---	---	-----	----	----

Define el tipo de velocidad en la comunicación:

- Estándar
- Alta
- Contiene banderas del estado que guarda la comunicación serie síncrona

Registro de control 14h

SPBRG	WCOL	SSPOV	SSPEN	CLKP	SSPM3	SSPM2	SSPM1	SSPM0
-------	------	-------	-------	------	-------	-------	-------	-------

Registro de configuración:

- Habilita la comunicación serie asíncrona
- Define la configuración I2C
- Configura la función del microcontrolador, ya sea maestro o esclavo
- Define el formato de direccionamiento (7 o 10 bits)

Registro de control 2 91h

SPBRG	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
-------	------	---------	-------	-------	------	-----	------	-----

Registro de configuración:

- Contiene las banderas que seleccionan la acción que será realizada
 - Habilita la condición START, STOP, START repetido
 - Habilita la recepción de datos en modo maestro
 - Habilita los bits de reconocimiento ACK

Registro Transmisión/Recepción de datos 13h

SSPBUF	7						0
--------	---	--	--	--	--	--	---

Registro en el cual se envía o recibe el dato, de acuerdo a la acción realizada

Registro configuración de dirección/velocidad 93h

SSPADD	7						0
--------	---	--	--	--	--	--	---

- a. En modo esclavo
 - a. Almacena la dirección del esclavo
- b. En modo maestro
 - a. Define la velocidad de transmisión que será usada en SCL

Cálculo de la velocidad:

$$f_{SCL} = f_{osc} / 4 \times (SSPADD + 1)$$

f_{osc} : Frecuencia del microcontrolador

SSPADD: 7 bits menos significativos del registro

Funciones y recursos del compilador de C para empleo del protocolo I2C.

Directiva

#use I2C (configuraciones)

Las configuraciones más empleadas son:

MASTER	Configura modo Maestro
SLAVE	Configura modo Esclavo
SCL=PIN	Define el pin a usar para SCL
SDA=PIN	Define el pin a usar para SDA
ADDRESS=dirección	Especifica la dirección del esclavo
FAST	Configura velocidad alta
SLOW	Define velocidad baja
FORCE_HW	Emplea funciones por hardware

Ejemplos:

- a. Configura modo maestro, asigna C4 para SDA y C3 para SCL, así como velocidad baja.
 - a. #use I2C(MASTER,SDA=PIN_C4, SCL=PIN_C3, SLOW)

- b. Configura modo esclavo, asigna C4 para SDA y C3 para SCL, velocidad baja, con dirección 0xa0.
 - a. `#USE I2C(SLAVE, SDA=PIN_C4, SCL=PIN_SDA, SLOW, ADDRESS=0X0A)`

Funciones disponibles:

- a. Inicia la comunicación I2C, enviando el bit S
 - a. `I2C_START();`
- b. Escribe un dato o la dirección del esclavo en el bus; cuando se use para direccionar al esclavo, los 7 bits más significativos indicarán la dirección establecida y el ultimo bit indicará la modalidad de escritura o lectura: A6,A5,A4,A3,A2,A1,A0,**R/W**. Por ejemplo; para un dispositivo cuya dirección esté especificada en su hoja técnica como: 1010000**R/W**:
 - a. `I2C_WRITE(ADDRESS);`
 - i. Cuando sigue a una función `I2C_START()`
 - 1. Modo escritura
 - a. `I2C_WRITE(0XA0);`
 - 2. Modo lectura
 - a. `I2C_WRITE(0XA1);`
 - ii. Cuando sigue a una función `I2C_WRITE`
 - 1. Modo escritura
 - a. `I2C_WRITE(DATO);`
- c. Finalizará la comunicación; enviará el bit de paro
 - a. `I2C_STOP();`
- d. El maestro realiza una petición al esclavo (modo lectura)
 - a. Lectura de datos:
 - i. Cuando reciba un dato, regresa un ACK
 - 1. `dato=I2C_READ();`
 - ii. Cuando reciba el último dato, regresa un NACK
 - 1. `dato=I2C_READ(0);`
- e. El esclavo recibe peticiones del maestro (modo escritura)
 - a. Regresa un TRUE en caso de haber recibido un dato
 - i. `I2C_POLL();`
 - b. Lectura del dato recibido
 - i. `dato=I2C_READ();`

Expansor de puertos PCF8574

Descripción

Este dispositivo es un expansor de puertos que permite comunicarse vía I2C con un microcontrolador maestro.

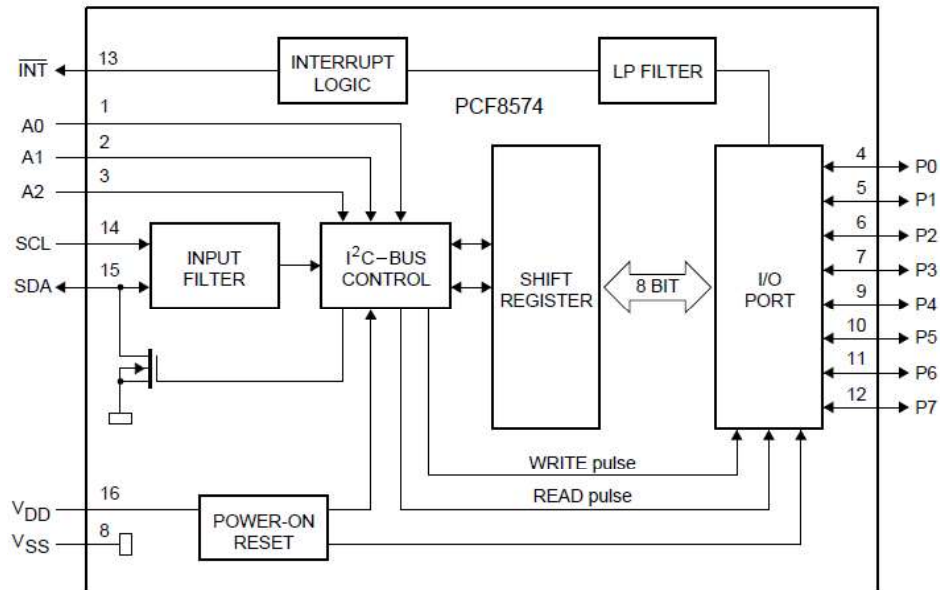
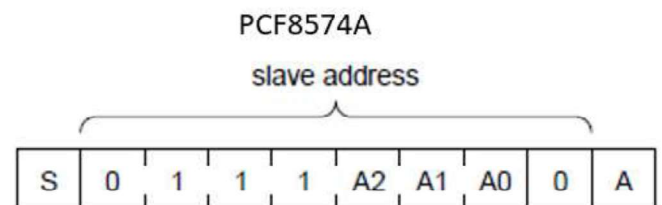


Figura 8.4 Diagrama general del PCF8574

Este circuito tiene las siguientes terminales:

- Terminal de datos SDA
- Terminal de reloj SCL
- Terminal de interrupciones INT, para generar una petición de interrupción cuando suceda una transición en el expansor de puertos
- VDD y VSS para alimentación de circuito 5V y 0V respectivamente
- Tres terminales para configurar la parte baja de la dirección que se asignará A2, A1 y A0.
- Ocho terminales bidireccionables P7...P0

La dirección de memoria es configurada colocando los niveles lógicos A0, A1 y A2 para conformar la dirección de 7 bits; existen dos modelos de este circuito: PCF8574 y PCF8574A, cuya única diferencia radica en los valores fijos de los bits más significativos.



En la práctica es empleado el modelo PCF8574 y la dirección configurada se indica en la actividad correspondiente.

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.

```
#include <16F877.h>
#fuses HS,NOWDT,NOPROTECT
#use delay(clock=20000000)
#use i2c(MASTER, SDA=PIN_C4, SCL=PIN_C3,SLOW, NOFORCE_SW)
int contador=0;

void escribir_i2c(){
    i2c_start();
    i2c_write(0x42);
    i2c_write(contador);
    i2c_stop();
}

void main()
{
    while(true)
    {
        escribir_i2c();
        delay_ms(500);
        contador++;
    }
}
```

El circuito se describe a continuación:

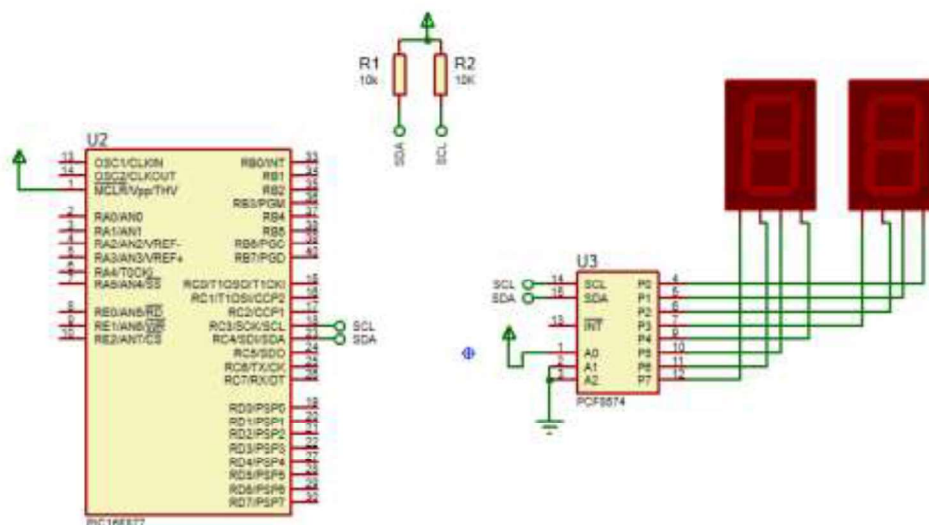


Figura 8.5 Circuito actividad 1; un maestro y un esclavo



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 fábrica 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 práctica se usará LCD de 16x2

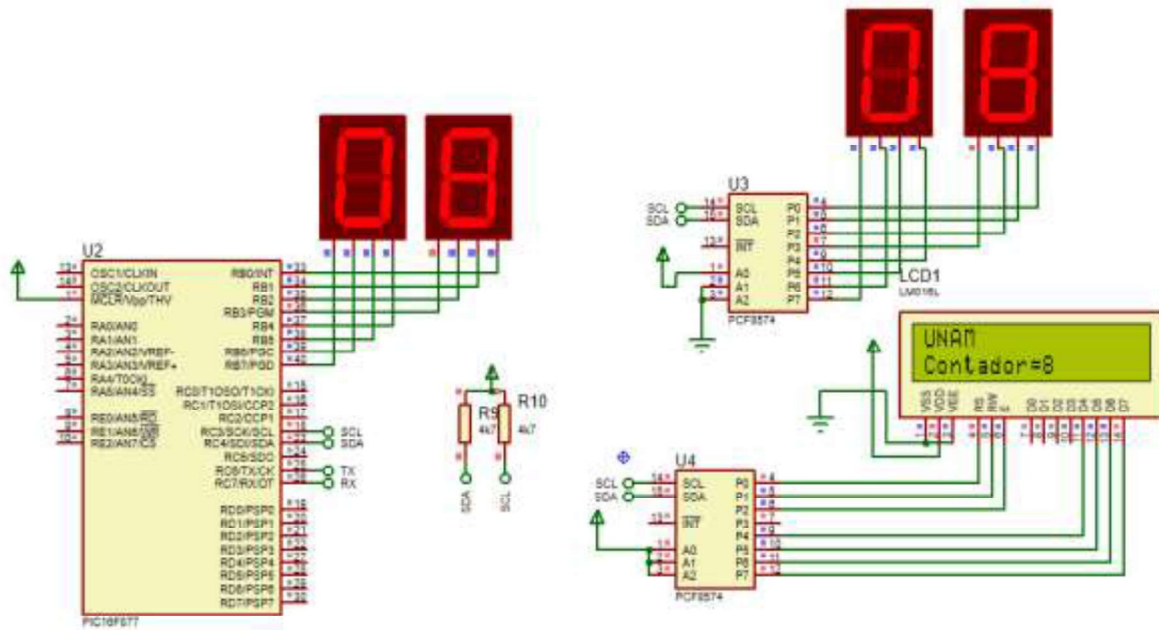


Figura 8.7 Circuito actividad 3; un maestro y dos esclavos

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.

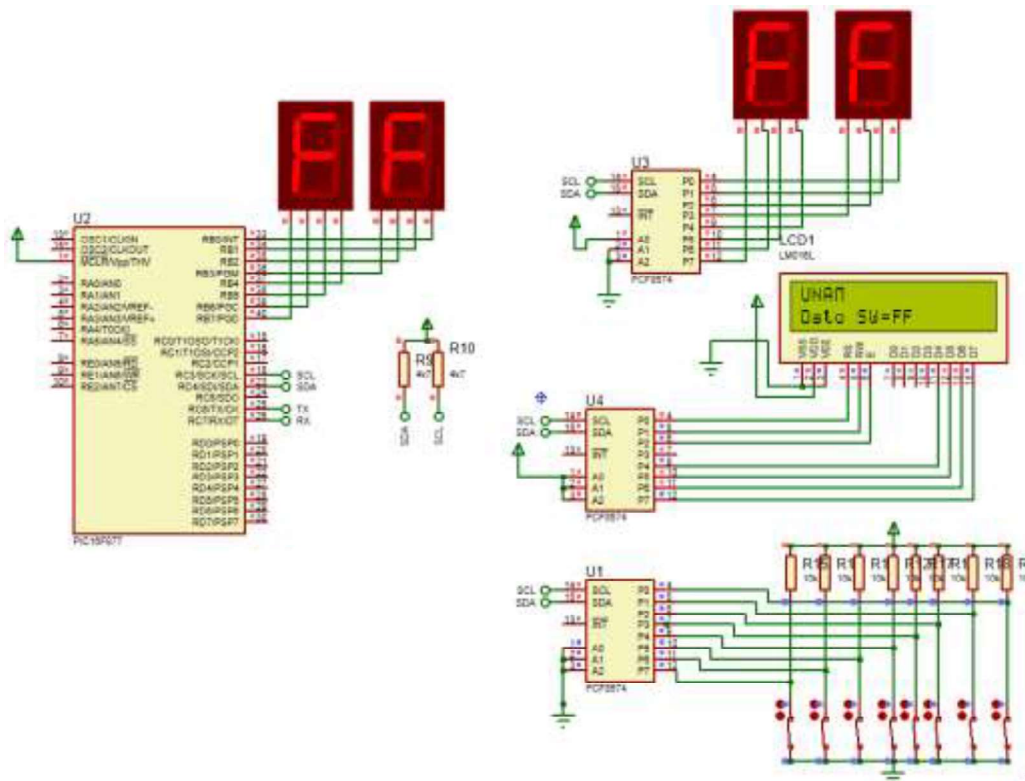


Figura 8.8 Circuito actividad 4; un maestro y tres esclavos