NOMBRE DEL ALUMNO: _____
Apellido Paterno

_____
Apellido Paterno Nombre

N° DE BOLETA: _____ GRUPO: _____

ASIGNATURA: ***Micro Electrónica Programable***

HOJA	DE

FECHA		
DIA	MES	AÑO

EVALUACION

PROFESOR: _____

Práctica 6
LDC y Teclado Matricial

Competencias de La Unidad:

- Emplea el microcontrolador en la comunicación de datos y control de periféricos de forma multiplexada.

Resultado de Aprendizaje Propuesto (RAP):

- Usa diferentes elementos periféricos utilizando los puertos del microcontrolador de forma multiplexada.
- Utiliza los puertos del microcontrolador de forma bidireccional en la solución de un problema.

.

Objetivos de la Práctica:

1. Realizar la simulación de un programa para comprobar su funcionamiento utilizando herramientas computacionales.
2. Realizar el manejo de un Display de Cristal Liquido (LCD) utilizando diversas técnicas de programación.
3. Manejar un teclado de tipo matricial con el fin de visualizar datos en un LDC.
4. Implementar programas en un circuito basado en microcontrolador, para comprobar su funcionamiento

Equipo Necesario	*Material Necesario*
Computadora (con el Software MPLAB IDE, IC-PROG o similar, compilador C, Simulador de circuitos electrónicos "Proteus") Programador tipo JDM o similar.	Instrucciones del PIC 16F887 u otro de gama media o alta Manual de referencia de C para microcontroladores PIC

Introducción Teórica

Display de Cristal Líquido (LDC)

Una **pantalla de cristal líquido** o **LCD** (acrónimo del inglés *Liquid Crystal Display*) es una pantalla delgada y plana formada por un número de píxeles en color o monocromos colocados delante de una fuente de luz o reflectora, por lo general se utiliza para visualizar información en dispositivos de tipo electrónico.

La pantalla de LCD consiste en una matriz de caracteres (por lo regular 5x7 puntos) cuya conexión se puede realizar mediante un cable plano de 14 hilos, en los cuales la mayoría de este tipo de dispositivos cuentan con controlador del tipo HD44780, de la japonesa Hitachi como seria el LM016L, pero no todos están normalizadas, por lo que se recomienda verificar las hojas de especificaciones.

Para poder utilizar el display de cristal liquido con un microcontrolador es necesario enviar los comandos adecuados para visualizar alguna información, en caso de utilizar lenguaje ensamblador se requiere definir el conjunto de instrucciones que permitan realizar dicha función, para simplificar esta tarea el compilador de C contiene un archivo (driver) que contiene las instrucciones para manejarlo, el cual debe llamarse mediante #include denominado LCD.C,[1] y las funciones son las siguientes:

lcd_init ();
Función que inicilializa al LCD y es la primera que se tienes llamar, dicha función realiza lo siguiente:

- Borra el LCD y lo configura en el formato de 4 bits, con dos líneas y con caracteres de 5x8 puntos, en modo encendido, cursor apagado y sin parpadeo.

- Configura el LCD con un autoincremento del puntero de direcciones y sin desplazamiento del display real.

Lcd_gotoxy (byte x , byte y);

Indica la posición del acceso al LCD. Por ejemplo,(1,1) indica la primera posición de la primera línea y (1,2) indica la primera posición de la segunda línea.

[1] Para el compilador CCS se llama LCD.c, para otro compilador es necesario verificar el manual de referencia para conocer como se denomina.

Lcd_getc(byte x , byte y);

Leer el carácter de la posición (x,y)

Lcd_putc (char s);

s es una variable de tipo char. Esta función escribe o imprime en el LCD la variable en la posición correspondiente. Además, se puede utilizar las siguientes variable :

/f **limpia el LCD**
/n **el cursor va a la posición (1,2).**
/b **el cursor retrocede una posición**.

El compilador de C ofrece una función mas versátil para trabajar con el LCD:

Printf (string)
Printf (cstring,values…)
Printf (fname,cstring,values…)

String es una cadena o un array de caracteres, values es una lista de variables separadas por comas y **fname** es una función,

En el formato %nt, donde n es opcional, el parámetro t puede ser ,c (Carácter), s (cadena o carácter), u (entero sin signo) o d (entero con signo), entre otros.

El driver LCD.C está configurado para trabajar con el puerto B o D, y por defecto utiliza el puerto D, pero se puede modificar para utilizar otro puerto.

Para utilizar el puerto B es necesario descomentar la siguiente línea el archivo LCD.C;

#define use_portb_lcd TRUE

Teclado matricial

Un ***teclado matricial*** es un ***arreglo de botones conectados en filas y columnas***, de modo que se pueden leer varios botones con el mínimo número de pines requeridos. Un teclado matricial 4×4 solamente ocupa 4 líneas de un puerto para las filas y otras 4 líneas para las columnas, de este modo se pueden leer 16 teclas utilizando solamente 8 líneas de un microcontrolador. Si asumimos que todas las columnas y filas inicialmente están en alto (1 lógico), la pulsación de un botón se puede detectar al poner cada fila a en bajo (0 lógico) y checar cada columna en busca de un cero, si ninguna columna está en bajo entonces el 0 de las filas se recorre hacia la siguiente y así secuencialmente.

El compilador de C (CCS) contiene al igual que en el LCD un archivo para manejar un teclado matricial de 3x4, denominado KBD.C que puede ser modificado para otro de dispositivos. Las funciones incorporadas son las siguientes:

kdb_init();
Inicializa el sistema, debe ser la primer función en llamar.

kbd_get();
Devuelve el valor de la tecla pulsada en función de la tabla que se tiene programada

El driver KBD.C trabaja con el puerto B o D, por configuración el puerto D funciona por defecto, pero se puede modificar para utilizar otro puerto.

Para utilizar el puerto B es necesario que la siguiente línea se descomente en el archivo;

#define use_portb_kbd TRUE

ACTIVIDADES TEÓRICAS PREVIAS

Investigar los siguientes:

- **Características del modelo LM016L**
- **Características de controlador tipo HD44780**
- **Explique las características y formas de conexión de un teclado matricial.**
- **Investigue el algoritmo para identificar cada una de las teclas en un teclado matricial.**
- **Menciona algunas características Importantes del archivo LCD.C**

ACTIVIDADES PREVIAS

- ***Crear un proyecto de nombre pra6 en la carpeta c:\MEPIC\practica6 en PIC C Compiler. Los programas de cada ejercicio deben ser guardados con el nombre practica6X.c con X= 1, 2, 3…,A.***

ACTIVIDADES PRÁCTICAS

Parte 1: Verificar el funcionamiento de los programas propuestos.

Ejemplo 1: Armar el circuito de la figura 6.1 y probar el siguiente programa, además de simularlo en Proteus.

```
#include <16f887.h>
#fuses XT,NOWDT,NOPUT,NOMCLR,NOPROTECT,NOCPD,NOBROWNOUT,NOIESO,NOFCMEN,NOLVP
#use delay(clock=4000000)
#include <lcd.c>
#use standard_io(a)
void main()
{
signed int8 i; //variable entera con signo la utilizaremos para recorrer texto
set_tris_a(0xFF);
lcd_init(); //inicializa la lcd
printf(lcd_putc,"\fPresiona boton\n"); //la instrucción Printf se utiliza para
//mandar datos utilizando la función
//lcd_putc,
//esto significa que los datos son
//enviados al lcd
```

```
while(TRUE)
{
if(input(PIN_A0)==0)
{
lcd_putc('\f'); //Limpia la LCD
printf(lcd_putc,"Micro-Electronik");
delay_ms(2000);
lcd_putc('\f');
printf(lcd_putc,"\fPresiona boton\n");
}
if(input(PIN_A1)==0)
{
lcd_putc('\f');
for(i=17;i>=-25;i--)
{
lcd_gotoxy(i,1); //indica la posición del primer caracter del
//texto donde i es la variable que decrementamos
printf(lcd_putc,"Recorriendo ala izquierda");
delay_ms(100);
lcd_putc('\f');
}
printf(lcd_putc,"\fPresiona boton\n");
}
if(input(PIN_A2)==0)
{
lcd_putc('\f');
for(i=-25;i<=25;i++)
{
lcd_gotoxy(i,1); //indica la posicion del primer caracter del
//texto donde i es la variable que
//incrementamos
printf(lcd_putc,"Recorriendo ala derecha");
delay_ms(100);
lcd_putc('\f');
}
printf(lcd_putc,"\fPresiona boton\n");
}
}
}
```

- ***Explique que sucede en el LCD al oprimir cada uno de los interruptores.***

- ***Menciona, ¿Cuál es la manera de recorrer un texto hacia la derecha o izquierda?***

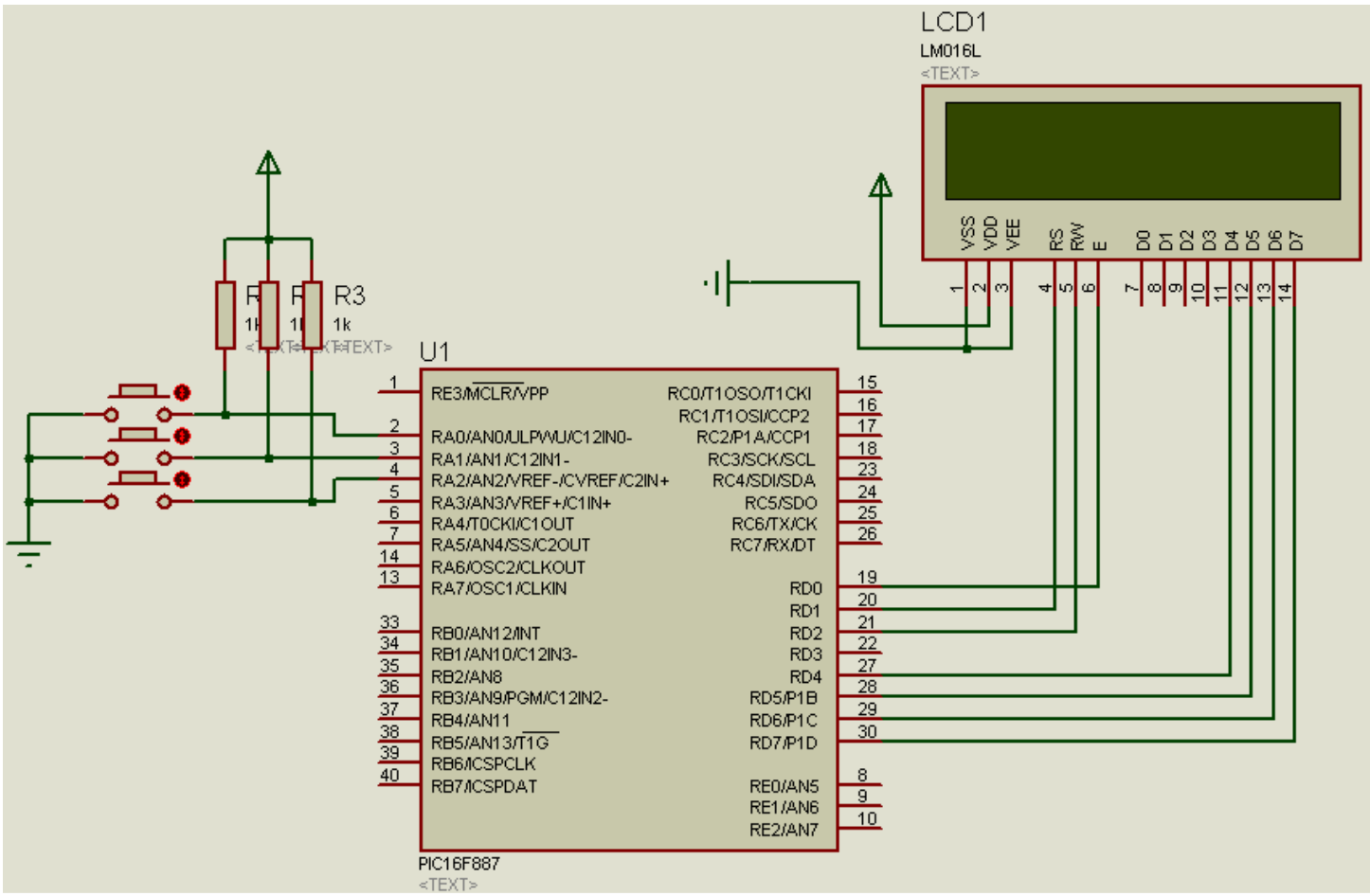

Figura 6.1

Ejemplo 2. Armar el circuito de la figura 6.2 y probar el siguiente programa, además de simularlo en Proteus.

```
#include <16f887.h>
#fuses XT,NOWDT,NOPUT,NOMCLR,NOPROTECT,NOCPD,NOBROWNOUT,NOIESO,NOFCMEN,NOLVP
#use delay(clock=4000000)
#include <kbd.c>
#use standard_io(b)
#use standard_io(c)
#use standard_io(d)
#include <stdlib.h>
#byte WPUB=0X95
#byte IOCB=0X96
#byte OPTION=0X81
#byte ANSEL=0X188
#byte TRISB=0X86
#byte ANSELH=0X189
#rom 0x2100={'*','5','#'}
byte CONST numeros[3] = {0x06,0x5B,0x4F};
int i;
char k;
```

```
char data[3], clave[3]; //Matrices para guardar clave y datos
void main()
{
wpub=0xff;
iocb=0x00;
option=0x00;
ansel=0x00;
anselh=0x00;
kbd_init();
output_c(0);
output_high(pin_c4);
while (TRUE)
{
i=0; //posición de la matriz
while(i<=2){ //Para tres datos
output_d(numeros[i]);
k=kbd_getc(); //Lee el teclado
if (k!=0) //Si se ha pulsado alguna tecla
{data[i]=k; //se guarda en la posición correspondiente
i++; //de la matriz
}
}
for (i=0;i<=2;i++) { //Pasa datos de eeprom a la matriz clave
clave[i]=read_eeprom(i);}
if ((data[0]==clave[0])&&(data[1]==clave[1])&&(data[2]==clave[2]))
{
output_high(pin_c0);
output_d(0x77);
delay_ms(2000);
// output_c(numeros[0]);
}
else
{
output_low(pin_c0);
output_d(0x39);
delay_ms(1000);
}
output_low(pin_c0);
}
}
```

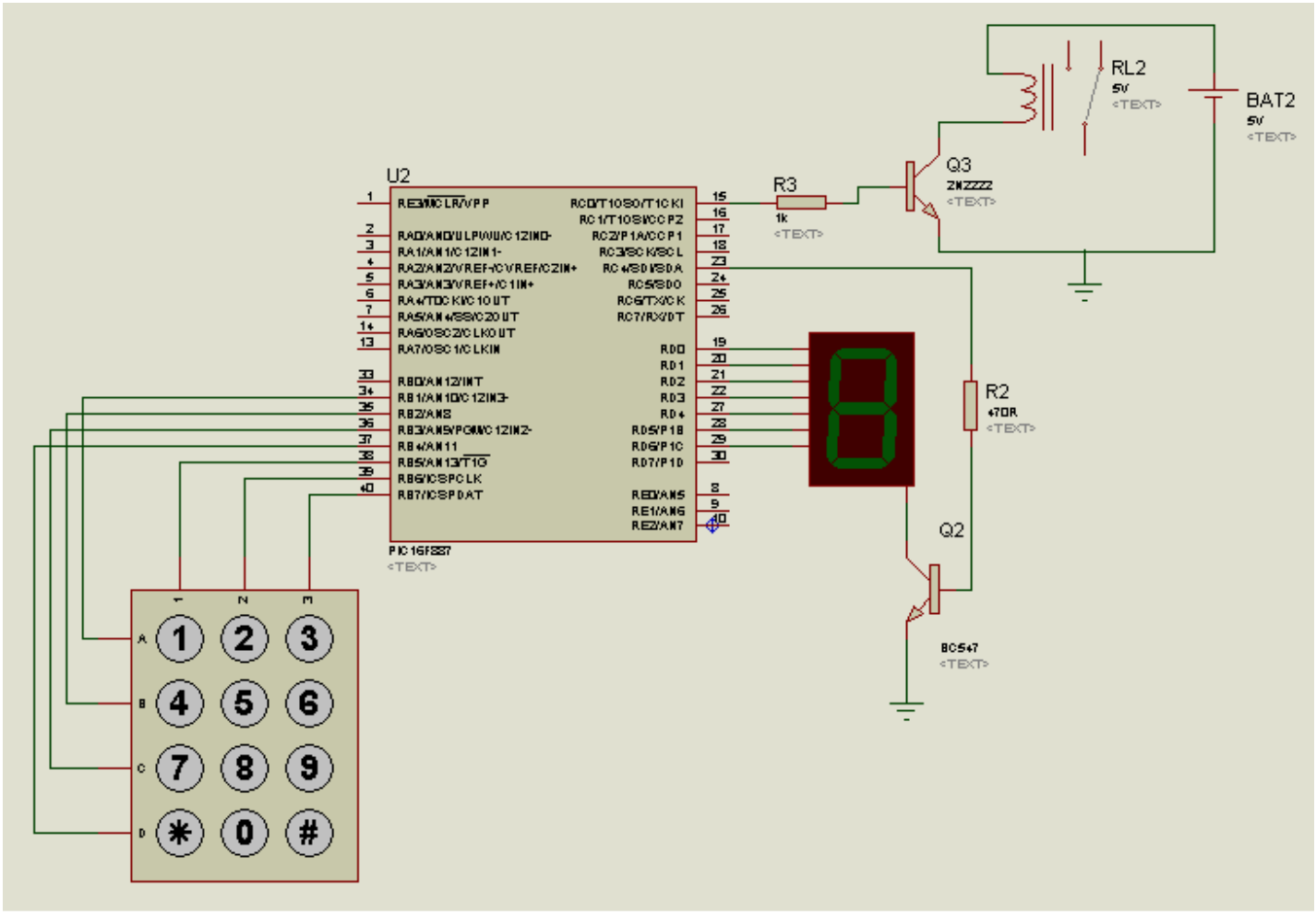

Figura 6.2

Ejemplo 3. Armar el circuito de la figura 6.3 y probar el siguiente programa, además de simularlo en Proteus.

```
#include <16f887.h>
#fuses XT,NOWDT,NOPUT,NOMCLR,NOPROTECT,NOCPD,NOBROWNOUT,NOIESO,NOFCMEN,NOLVP
#use delay(clock=4000000)
#include <lcd.c>
#include <kbd.c>
#byte WPUB=0X95
#byte IOCB=0X96
#byte OPTION=0X81
#byte ANSEL=0X188
#byte TRISB=0X86
#byte ANSELH=0X189
void main()
{
char k;
wpub=0xff;
iocb=0x00;
option=0x00;
ansel=0x00;
anselh=0x00;
```

```
lcd_init();
kbd_init();
printf(lcd_putc,"\fPresiona tecla");
while(TRUE)
{
k=kbd_getc();
if(k!=0)
{
lcd_putc('\n');
printf(lcd_putc,"Tecla= %c",k);
delay_ms(500);
printf(lcd_putc,"\fPresiona tecla");}
}
}
```

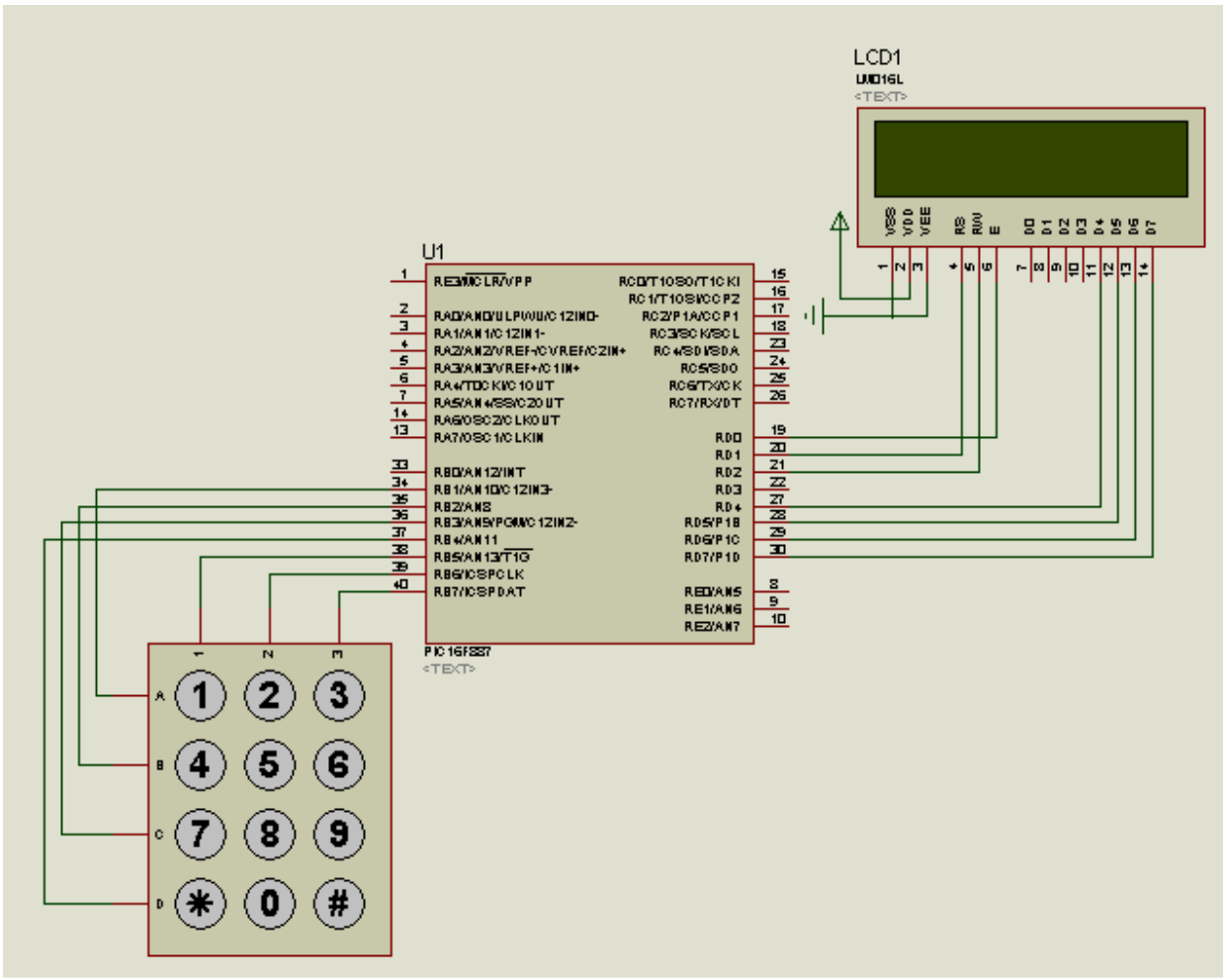

Figura 6.3

Ejemplo 4. Armar el circuito de la figura 6.4 y probar el siguiente programa.

```
#include <16f887.h>
#fuses XT,NOWDT,NOPUT,NOMCLR,NOPROTECT,NOCPD,NOBROWNOUT,NOIESO,NOFCMEN,NOLVP
#use delay(clock=4000000)
#include <lcd.c>
#include <kbd.c>
#byte WPUB=0X95
#byte IOCB=0X96
#byte OPTION=0X81
```

```
#byte ANSEL=0X188
#byte TRISB=0X86
#byte ANSELH=0X189
void main()
{
char k;
wpub=0xff;
iocb=0x00;
option=0x00;
ansel=0x00;
anselh=0x00;
lcd_init();
kbd_init();
printf(lcd_putc,"\fPresiona tecla");
while(TRUE)
{
k=kbd_getc();
if(k!=0)
{lcd_putc('\n');
printf(lcd_putc,"Tecla= %c",k);
delay_ms(500);
if(k=='1')
{lcd_putc('\n');
printf(lcd_putc,"\fAdquiriendo");
delay_ms(1000);}
else if(k=='3')
{lcd_putc('\n');
printf(lcd_putc,"\fCalibrando");
delay_ms(1000);}
else if(k=='8')
{lcd_putc('\n');
printf(lcd_putc,"\fEn espera");
delay_ms(1000);}
else if(k=='*')
{lcd_putc('\n');
printf(lcd_putc,"\fMenu");
delay_ms(1000);}
else if(k=='#')
{lcd_putc('\n');
printf(lcd_putc,"\fTransferir");
delay_ms(1000);}
else {lcd_putc('\n');
printf(lcd_putc,"\fNo valida");
delay_ms(1000);}
printf(lcd_putc,"\fPresiona tecla");}
}
}
```

Parte 2

Utilizando el Circuito de la figura 6.4 y el elemento actuador del circuito 6.3 (relevador), proponga un programa que inicie con el mensaje "Contraseña" y obtenga 3 dígitos de contraseña mediante el teclado, visualizando en un LCD uno a uno de los caracteres, al obtener los tres dígitos visualizar la contraseña y esperar un segundo para mostrar el mensaje "acceso" recorriéndolo a la derecha y activar el relevador durante 2 segundo, el caso de ser incorrecto que aparezca "denegado" recorriendo el texto a la derecha durante 2 segundos, al finalizar que aparezca el texto "Contraseña".

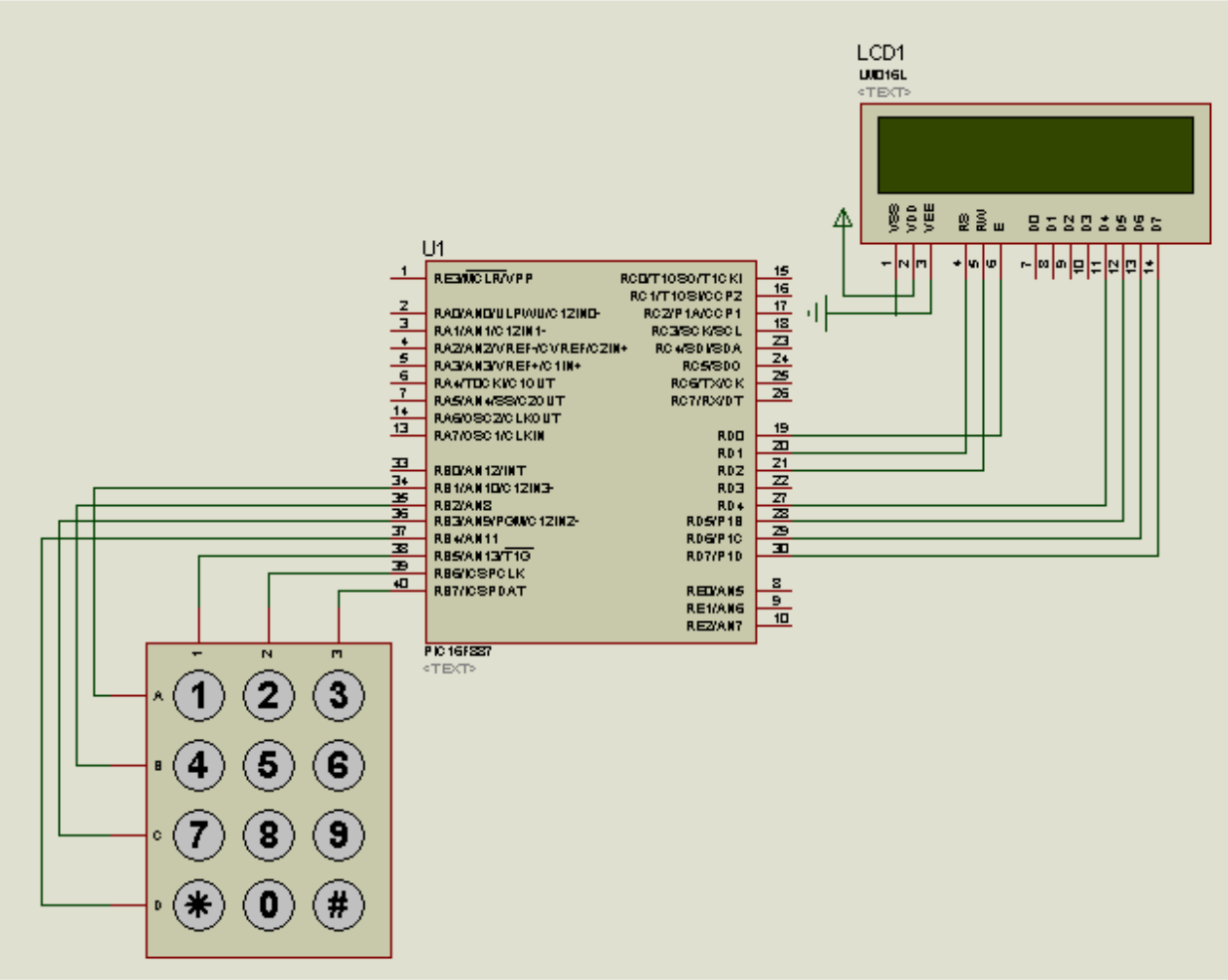

Figura 6.4

2. Conclusiones

A. Realizar conclusiones de manera individual.

3. Cuestionario

a) Menciona los diferentes formatos para expresar un carácter en un LCD con el compilador de C

b) En el caso del archivo de manejo de LCD como de teclado, ¿Que modificaciones se tienen que realizar para utilizar un puerto diferente al B o D?

c) ¿Por qué en la instrucción #rom se tiene el valor 0x2100
d) Que modificación se tendrían que realizar para manejar el teclado en puerto B y LCD en el puerto C
e) Que modificación se tendrían que realizar para manejar el teclado en puerto C y LCD en el puerto D, si se desea utilizar un pic16f887

Comentarios Finales

- **El alumno entrega un reporte de la práctica, como el profesor lo indique.**
- **El reporte debe contener el diagrama de flujo o algoritmo (Seudo código) de cada uno de los programas.**
- **Además, en el reporte deben anexarse las conclusiones y cuestionario contestado.**