**Practica #2: Uso de Teclado Matricial y Pantalla de Caracteres LCD**

1. **Objetivos**
   * Implementar un medio de visualización a través de un LCD 16x2 para mostrar datos que se ingrese por un teclado en el Microcontrolador.
   * Entender el funcionamiento de las librerías del teclado y la pantalla lcd.
2. **Materiales**

Computadora con Software MPLAB X IDE y Proteus

1. **Introducción**

En casi todos los proyectos es necesario leer alguna entrada de tipo digital conectada a pulsadores, interruptores, teclados o sensores digitales; también es necesario escribir datos por medio de una salida de tipo digital conectada a LED, pantallas LCD, display de siete segmentos o similares. En este laboratorio aprenderemos a leer datos adquiridos por un teclado matricial e imprimirlos mediante una pantalla lcd, la cual está formada por filas y columnas que manejaremos según las coordenadas y las instrucciones que se presentan en su librería.

**Conexión de los Periféricos**

**Pantalla LCD**

El Módulo LCD a implementar en esta guía de laboratorio es una LCD de 16x2, la cual posee 16 pines de conexión que se muestran en la Fig. 1 y que cada pin realiza una función según la Tabla 1. En Proteus aparece con el nombre de LM016L.

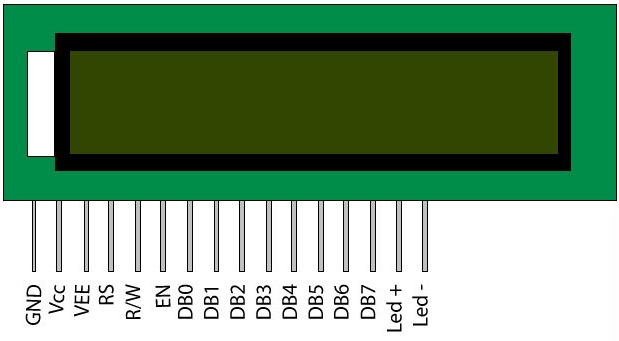


Fig. 1 Modulo LCD 16x2

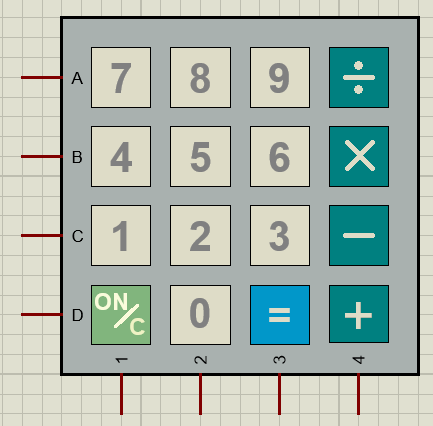
|  |  |  |
| --- | --- | --- |
| Pin | Nombre del Pin | Función |
| 1 | VSS o GND | Voltaje de Tierra |
| 2 | VDD | Voltaje de Alimentación de 5V |
| 3 | VEE | Voltaje de Regulación de Contraste. Normalmente se conecta a un Potenciómetro para regular su contraste. |
| 4 | RS | Selección del Registro de Control/Registro de Datos  RS=0 Selección de Registro de Control.  RS=1 Selección de Registro de Datos. |
| 5 | R/W | Señal de Lectura/Escritura  R/W=0 El Modulo LCD es escrito.  R/W=1 El Modulo LCD es leído. |
| 6 | E | Señal de Activación del Módulo LCD:  E=0 Modulo Desconectado.  E=1 Modulo Conectado |
| 7 – 14 | D0 – D7 | Bus de Datos bi-direccional. A través de estas líneas se realiza la transferencia entre el módulo LCD y el sistema que lo gestiona. |
| 15 | LED + | Ánodo del Led Interno (Backlight) |
| 16 | LED - | Cátodo del Led Interno (Backlight) |

Tabla 1 Función de los Pines del Módulo LCD 16x2

Teclado Matricial

El teclado matricial a implementar en Proteus recibe el nombre de Keypad-Smallcalc representando a un teclado 4x4 (Ver Figura). Las conexiones del teclado matricial se harán siguiendo las especificaciones de la librería quedando a como se muestra en la tabla.

|  |  |
| --- | --- |
| **Nombre de Pin de Teclado** | **Pin de Conexión** |
| Fila A | RB4 |
| Fila B | RB5 |
| Fila C | RB6 |
| Fila D | RB7 |
| Columna 1 | RB0 |
| Columna 2 | RB1 |
| Columna 3 | RB2 |
| Columna 4 | RB3 |



1. **Procedimiento de la Practica.**
2. Abrir el MPLAB X IDE y cree un nuevo proyecto con el nombre de **Keypad\_LCD**.
3. Cree un archivo fuente con el nombre de **Menu** y copie el siguiente código.

#include <xc.h>

#include "16x2\_LCD\_4bit\_File.h"

#include "keypad.h"

#include "ConfigurationBits.h"

#define \_XTAL\_FREQ 4000000

#include <PIC18F4550.h>

int x = 0;

void initMain(void){

TRISC = 0xF8; //Configura los pines LSB (B0, B1 y B2) como salidas

INTCON2bits.RBPU=0;

LCD\_Init();

LCD\_String\_xy(1,0,"MENU");

LCD\_String\_xy(2,0,"1->R 2->A 3->V");}

void main(void) {

initMain();

char k = '\0';

while(1){

k = kbd\_getc();//obtiene el valor del teclado

if(k != 0){

x = k - 48;

LCD\_String\_xy(1,6,"");

LCD\_Char(k);

}

switch(x){

case 0: //Apagar todos los leds

LATC = 0xF8;

break;

case 1: // Encender Led 1

LATC = 0xF9;

break;

case 2: // Encender Led 2

LATC = 0xFA;

break;

case 3: // Encender Led 3

LATC = 0xFC;

break;

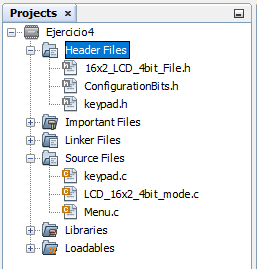
default: // Encender todos los leds

LATC = 0xFF;

break;

} }return;}

1. En la estructura de directorios del Proyecto en MPLAB X IDE se tendrá que agregar nuevos archivos correspondiente a las librerías de la pantalla LCD y el teclado matricial. Estos archivos deben agregarse con la opción ***Add Existing Item*** y deben de quedar según el orden de la Figura.



1. Compile el proyecto completo y simule en Proteus a como se muestra en la Figura.
   1. **Parte Final: Cerradura Electrónica.**
2. Ahora que manejamos como usar los dispositivos del teclado matricial y la pantalla LCD, haremos una cerradura electrónica, por lo que se tendrá que crear un nuevo proyecto llamado **Cerradura Electronica**. Este proyecto hará uso de las mismas librerías del ejercicio principal.
3. Crear un nuevo archivo fuente llamado igual **Cerradura Electronica**, el que tendrá el código principal que controlará la cerradura.

#include <xc.h>

#include "16x2\_LCD\_4bit\_File.h"

#include "keypad.h"

#include "ConfigurationBits.h"

#define \_XTAL\_FEQ 8000000

#include <PIC18F4550.h>

char buffer[4] = {'\*', '\*', '\*', '\*'}; //inicializar vector

const char pswd[4] = {'1', '2', '3', '4'}; //vector de contraseña almacenada

void initMain(void) {

TRISC = 0xF8; //3 LSB's como salidas

INTCON2bits.RBPU = 0; //Habilita las pullups para el teclado

LCD\_Init();

LCD\_String\_xy(1, 1, "Pswd:");

LCD\_String\_xy(2, 1, "Cerrado...");

}

void main(void) {

char k = '\0';

char x = '\0';

int i = 0;

initMain();

while (1) {

k = kbd\_getc();

if ((k != 0)&&(k!='\*')&&(k!='#')) {

buffer[i] = k;

LCD\_String\_xy(1,6+i,"");

LCD\_Char('\*');

i++;

}

if (k == '\*') {

LCD\_String\_xy(1, 6 + i, "");

LCD\_Char('\0');

i--;

buffer[i + 1] = '\0';

}

if ((i == 4)&&(k == '#')) {//si ya se han ingresado 4 caracteres comprueba

if ((pswd[0] == buffer[0])&&(pswd[1] == buffer[1])&&(pswd[2] == buffer[2])&&(pswd[3] == buffer[3])) {

LCD\_String\_xy(2, 1, "ABIERTO");

LATC = 0xF4;

\_\_delay\_ms(3000);

buffer[0] = '\0';

buffer[1] = '\0';

buffer[2] = '\0';

buffer[3] = '\0';

LCD\_Clear();

LCD\_String\_xy(1, 1, "Pswd:");

LCD\_String\_xy(2, 1, "Cerrado...");

LATC = 0xF9;

i = 0;

} else {

LATCbits.LATC0 = 0;

\_\_delay\_ms(100);

LATCbits.LATC0 = 1;

\_\_delay\_ms(100);

LATCbits.LATC0 = 0;

\_\_delay\_ms(100);

LATCbits.LATC0 = 1;

\_\_delay\_ms(100);

LCD\_Clear();

LCD\_String\_xy(1, 1, "Password");

LCD\_String\_xy(2, 1, "Incorrecta");

\_\_delay\_ms(1000);

LCD\_Clear();

LCD\_String\_xy(1, 1, "Intente");

LCD\_String\_xy(2, 1, "Nuevamente");

LATC = 0x02;

\_\_delay\_ms(1000);

buffer[0] = '\0';

buffer[1] = '\0';

buffer[2] = '\0';

buffer[3] = '\0';

LCD\_Clear();

LCD\_String\_xy(1, 1, "Pswd:");

LCD\_String\_xy(2, 1, "Cerrado...");

LATC = 0xF9;

i = 0;

for (int j = 0; j < 4; j++) //OJO reinicia el buffer a valores por defecto

buffer[j] = '\*';

}

}

}

return;

}

1. Una vez creado los archivos, compilar el proyecto para generar el archivo .hex y utilizaremos el diagrama electrónico creado en el ejercicio anterior.