

Laboratorio 2: Manejo de Modulo LCD

1. Datos de la Practica

Carrera	INGENIERÍA ELECTRÓNICA		
Semestre		Grupo	
Tipo de Práctica	<input type="checkbox"/> Laboratorio <input type="checkbox"/> Simulación	Fecha	
Asignatura	Curso de Arduino		
Unidad Temática			
Nº Alumnos por práctica	2	Nº Alumnos por reporte	2
Nombre del Profesor			
Nombre(s) de Alumno(s)	1. 2.		
Tiempo estimado		Vo. Bo. Profesor	
Comentarios			

2. Objetivos

- Implementar un medio de visualización a través de un LCD 16x2 para mostrar datos que el Microcontrolador necesite.
- Implementar la librería LiquidCrystal.h para el manejo de datos en la pantalla.

3. Componentes a Utilizar

Por cada práctica y por cada puesto de laboratorio, los materiales a utilizar son:

Cantidad	Descripción
1	Computadora
1	Arduino MEGA
1	Potenciómetro 10K
1	LCD 16x2

4. Introducción

En casi todos los proyectos es necesario leer alguna entrada de tipo digital conectada a pulsadores, interruptores, sensores digitales o similares; 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 imprimir datos 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 del Módulo LCD

El Modulo 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**.

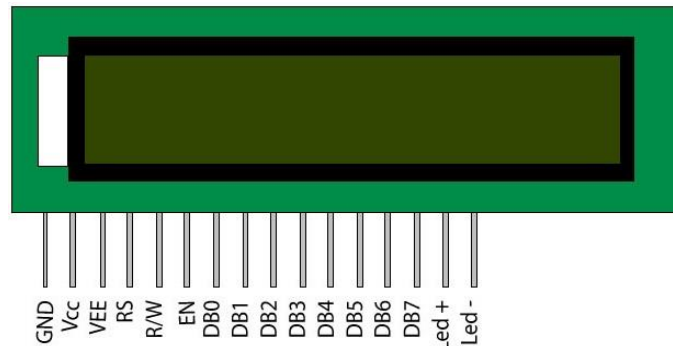


Fig. 1 Modulo LCD 16x2

Pin	Nombre del Pin	Función
1	V _{SS} o GND	Voltaje de Tierra
2	V _{DD}	Voltaje de Alimentación de 5V
3	V _{EE}	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	DO – 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

Programación del Arduino

La librería LiquidCrystal.h de Arduino para la lcd presenta varias funciones de las que se detallan a continuación:

Print: Escribe un texto en el LCD

Sintaxis: lcd.print(textoParaMostrar,BASE);

Parámetros:

textoParaMostrar: Es el texto que queremos que aparezca en la pantalla.

BASE: Esto es OPCIONAL y sirve para definir el formato en el que queremos.

Puede ser: BIN Binario (base 2); DEC Decimal (base 10); OCT Octal (base 8)

HEX Hexadecimal. (base 16).

Ejemplo: `lcd.print('HOLA MUNDO');`

Begin: Esta función sirve para especificar el tamaño de nuestro Display.

Sintaxis: `lcd.begin(columnas,filas)`

Parámetros:

Columnas: Numero de columnas que tiene nuestro LCD

Filas: Numero de filas que tiene nuestro LCD

Ejemplo: `lcd.begin(16,2);`

Clear: Esta función borra todo lo que estuviese escrito en nuestro LCD

Sintaxis: `lcd.clear()`

Home: Coloca el cursor en la esquina superior derecha del display.

Sintaxis: `lcd.home();`

Write: Escribe un carácter en el Display.

Sintaxis: `lcd.write(caracter);`

Parámetros:

carácter: el carácter que queremos mostrar por pantalla.

Ejemplo: `lcd.write('a');`

Blink: Muestra un cursor con forma de bloque.

Sintaxis:

`lcd.blink()`

`lcd.noBlink()`

Cursor: Muestra un cursor con forma de guion bajo.

Sintaxis:

`lcd.Cursor()`

`lcd.noCursor()`

Display: Esta función permite encender y apagar el Display, pero sin perder lo que estaba mostrando. Es decir, que si mostramos por pantalla un texto y acto seguido hacemos un “noDisplay()” dejaremos de ver la información que estábamos mostrando, pero reaparecerá al hacer un “Display()” sin necesidad de volver a mandar el texto.

Esto puede ser interesante por ejemplo si estamos por ejemplo mostrando un menú por un Display y queremos ahorrar energía. podríamos apagarlo tras un periodo de inactividad y volver a encenderlo exactamente en el mismo sitio cuando el usuario vuelva a presionar algún botón.

Sintaxis:

`lcd.display()`

`lcd.noDisplay()`

Text Direction: `rightToLeft()` Cuando se escribe, las letras aparecen a la izquierda del cursor. Es como si en un editor de textos tuviésemos el texto alineado a la derecha.

`leftToRight()` Al contrario que el anterior, las letras aparecen a la derecha del cursor, la sensación que da es la de alineación a la izquierda.

Sintaxis:

`lcd.leftToRight()` `lcd.rightToLeft()`

Autoscroll: Este programa de ejemplo muestra cómo usar las funciones `autoscroll()` y `noAutoscroll()` para mover todo el texto de la pantalla a la izquierda o a la derecha.

`autoscroll()` mueve todo el texto un espacio a la izquierda cada vez que se añade una letra.
`noAutoscroll()` apaga el desplazamiento

Sintaxis:

`lcd.autoscroll()` `lcd.noAutoscroll()`

SetCursor: Esta función sirve para posicionar el cursor del LCD, o lo que es lo mismo, para decir donde queremos escribir cada vez. El primer parámetro del paréntesis es la posición en X (es decir, en horizontal, más a la izquierda o más a la derecha) El segundo es la posición en Y (más arriba o más abajo).

Sintaxis: `lcd.setCursor(fila, columna);`

Ejemplos:

`lcd.setCursor(0, 0); // arriba a la izquierda`
`lcd.setCursor(15, 0); // arriba a la derecha`

Scroll: Desplaza el texto escrito en el Display.

`ScrollDisplayLeft()`: Desplaza el texto hacia la izquierda.

`ScrollDisplayRight()` : Desplaza el texto hacia la derecha.

Sintaxis:

`lcd.ScrollDisplayLeft()` `lcd.ScrollDisplayRight()`

CreateChar: La idea de esta función es la de crear “emoticonos” en el LCD, puedes crear un máximo de 8, que quedan numerados del 0 al 7 y después utilizarlos mediante esta función.

Sintaxis: `lcd.createChar(num, nombre)`

Parámetros:

num: Que carácter de los creados quieres usar.

Nombre: nombre que se le ha puesto al emoticono

Todas estas funciones serán de utilidad para el funcionamiento correcto de la librería `lcd`, además que algunas de ellas se utilizarán en la práctica del laboratorio.

5. Prácticas de Laboratorio

Practica #1 Uso de la librería LiquidCrystal

Código Arduino

En esta primer practica se aprenderá a usar varias funciones de la librería LiquidCrystal.h trabajando primero la simulación y después su implementación con la pantalla lcd en la tarjeta de Arduino. El código para esta práctica es el siguiente:

```
#include <LiquidCrystal.h>
// Inicializa la libreria con los numeros de pines de la lcd
const int rs = 6, en = 7, d4 = 8, d5 = 9, d6 = 10, d7 = 11;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  lcd.begin(16, 2); // Configura el tamaño de la lcd en columnas y filas
}

void loop() { // Corre todas las funciones de demostracion de la lcd
  basicPrintDemo();
  displayOnOffDemo();
  setCursorDemo();
  scrollLeftDemo();
  scrollRightDemo();
  cursorDemo();
  createGlyphDemo();
}

void basicPrintDemo() {
  lcd.clear(); // Limpia Pantalla
  lcd.print("Hola Mundo"); // Imprime el texto
  delay(2000);
}

void displayOnOffDemo() {
  lcd.clear();
  lcd.print("Display On/Off");
  for(int x=0; x < 3; x++) { // Repite el ciclo 3 veces
    lcd.noDisplay(); // Apaga el display
    delay(1000);
    lcd.display(); // Vuelve a encender el display
    delay(1000);
  }
}
```

```
void setCursorDemo() {
  lcd.clear();
  lcd.print("SetCursor Demo");
  delay(1000);
  lcd.clear();
  lcd.setCursor(5,0); // Ubica el Cursor en la columna 5 y fila 0
  lcd.print("5,0");
  delay(2000);
  lcd.setCursor(10,1); //Ubica el Cursor en la columna 10 y fila 1
  lcd.print("10,1");
  delay(2000);
  lcd.setCursor(3,1); // Ubica el Cursor en la columna 3 y fila 1
  lcd.print("3,1");
  delay(2000);
}

void scrollLeftDemo() {
  lcd.clear();
  lcd.print("Scroll Left Demo");
  delay(1000);
  lcd.clear();
  lcd.setCursor(10,0);
  lcd.print("Hola");
  lcd.setCursor(10,1);
  lcd.print("Mundo");
  delay(1000);
  for(int x=0; x<16; x++) {
    lcd.scrollDisplayLeft(); //Desplaza el texto a la izquierda 16 veces
    delay(250);
  }
}

void scrollRightDemo() {
  lcd.clear();
  lcd.print("Scroll Right");
  lcd.setCursor(0,1);
  lcd.print("Demo");
  delay(1000);
  lcd.clear();
  lcd.print("Hola");
  lcd.setCursor(0,1);
  lcd.print("Mundo");
  delay(1000);
  for(int x=0; x<16; x++) {
    lcd.scrollDisplayRight(); //Desplaza el texto a la derecha 16 veces
    delay(250);
  }
}
```

```
void cursorDemo() {
  lcd.clear();
  lcd.cursor(); // Habilita el cursor visible
  lcd.print("Cursor On");
  delay(3000);
  lcd.clear();
  lcd.noCursor(); //Cursor invisible
  lcd.print("Cursor Off");
  delay(3000);
  lcd.clear();
  lcd.cursor();
  lcd.blink(); // Cursor blinking
  lcd.print("Cursor Blink On");
  delay(3000);
  lcd.noCursor(); // Cursor invisible
  lcd.noBlink(); // Blink off
}

void createGlyphDemo() {
  lcd.clear();
  byte heart[8] = {B00000,B01010,B11111,B11111,B11111,B01110,B00100,B00000};
  byte happy[8] = {B00000,B00000,B10001,B00000,B10001,B01110,B00000,B00000};
  byte sad[8] = {B00000,B00000,B10001,B00000,B01110,B10001,B00000,B00000};
  lcd.createChar(0, heart); // Create custom character 0
  lcd.createChar(1, happy); // Create custom character 1
  lcd.createChar(2, sad); // Create custom character 2
  lcd.setCursor(1,0);
  lcd.print("I ");
  lcd.write(byte(0)); // escribe el caracter como un byte
  lcd.print(" Arduino! ");
  lcd.write(byte(1));
  delay(1000);
  lcd.clear();
  lcd.setCursor(1,0);
  lcd.print("Trust Me");
  lcd.setCursor(0,1);
  lcd.print("I'm an Engineer");
  for(int x=0; x<5; x++) { // Loop animation 5 times
    lcd.setCursor(15,1);
    lcd.write(byte(1)); // Write custom char 1
    delay(1000);
    lcd.setCursor(15,1);
    lcd.write(byte(2)); // Write custom char 2
    delay(1000);}
}
```

Simulación en Proteus

La conexión del lcd con la tarjeta Arduino se hará según la **Fig. 2**, teniendo en cuenta que el modelo de lcd a utilizar en Proteus tiene un código LM016L correspondiente a una pantalla de 16x2, es decir 16 columnas y 2 filas.

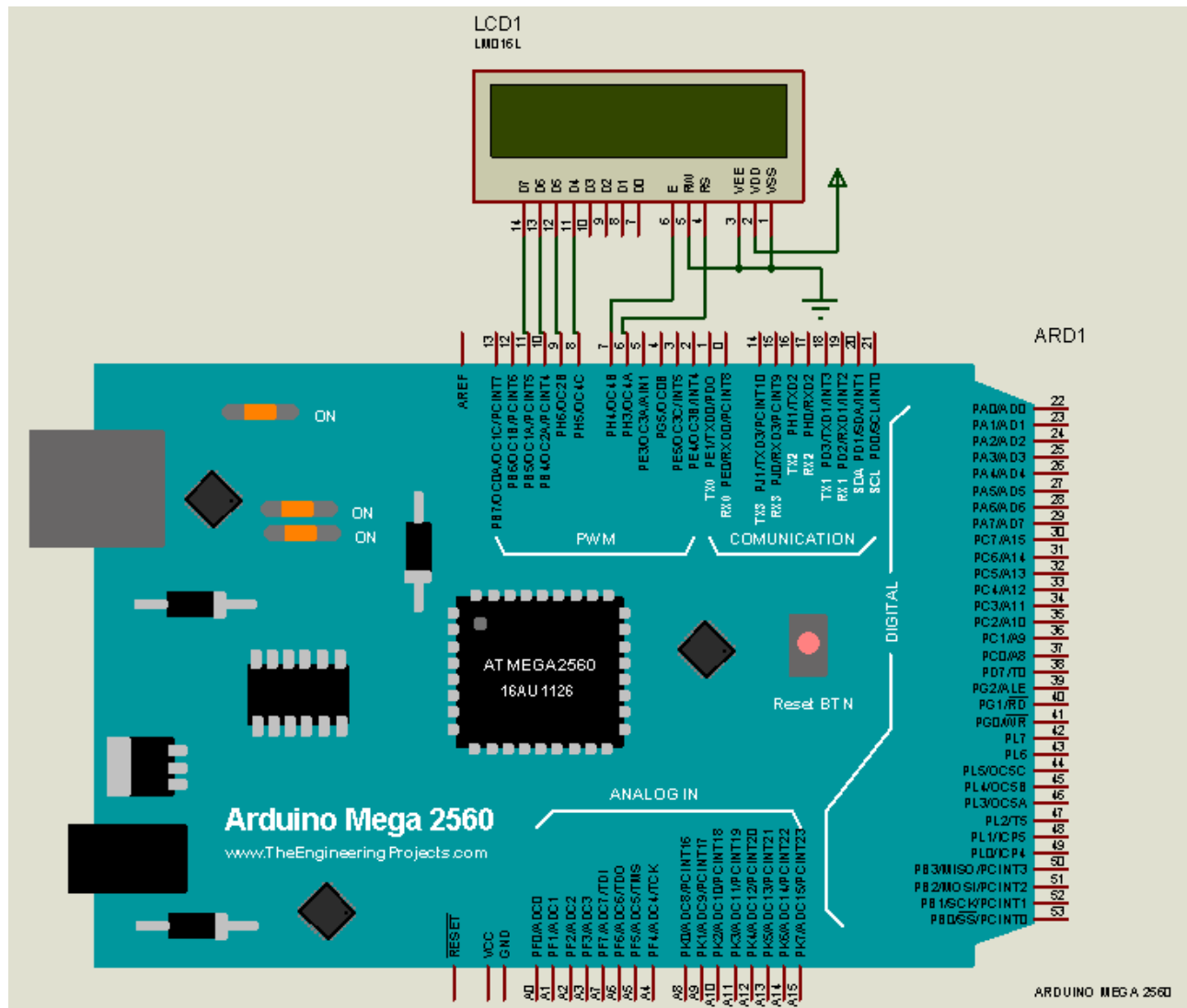


Fig. 2 Simulación de Practica #1

Montaje del Circuito

Para montar la práctica simulada, es necesario tener un componente extra, es decir un potenciómetro, esto es necesario para poder regular el contraste de lo que se está visualizando en la pantalla. Dicho potenciómetro tiene que ser de un valor de 10K y debe tener un pin a VCC y el otro a GND, el pin de en medio (el regulable) debe ir al pin de contraste de la pantalla según **Fig. 3**.

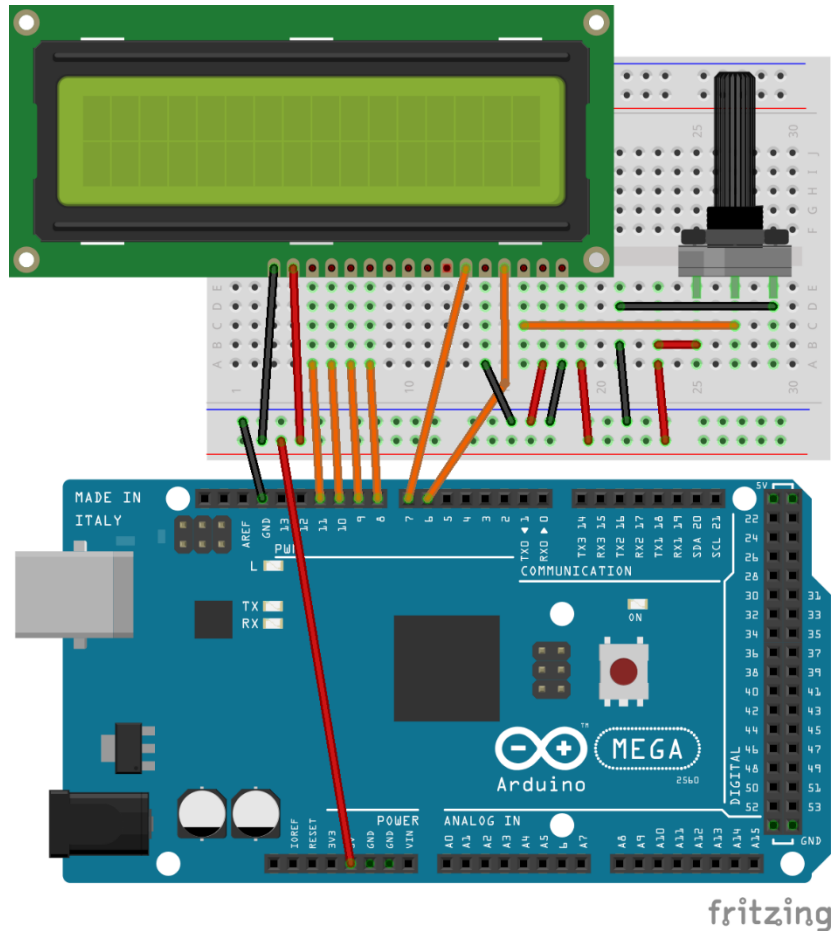


Fig. 3 Diagrama de Montaje de LCD

Actividad

Realice un nuevo código de programación donde se imprima 5 mensajes en la pantalla lcd, ocupando las distintas funciones de la práctica anterior. Dichos mensajes tendrán la siguiente secuencia:

Mensaje #1:	Fila 1 =====➡	Bienvenidos a la	//Tiene que estar centrado
	Fila 2 =====➡	UNI	//Tiene que estar centrado
Mensaje #2:	Fila 1 =====➡	Carreras de la	//Centrado y parpadear 3 veces
	Fila 2 =====➡	FEC	// Centrado y parpadear 3 veces
Mensaje #3:	Fila 1 =====➡	Electrica	//Tiene que desplazarse a la derecha
Mensaje #4:	Fila 1 =====➡	Computación	//Tiene que desplazarse a la izquierda
Mensaje #5:	Fila 1 =====➡	Electronica 🧡	//Tiene que tener el carácter corazón

6. Actividades Propuestas

Realizar un código nuevo en el cual se agregue un pulsador como entrada para cambiar entre los distintos mensajes, por cada pulsación se visualizará un mensaje distinto manteniendo los efectos de los mensajes del ejercicio anterior.