

Teoría LCD “El Grande”

(Versión 1-10-18)

Ya hemos anteriormente otro LCD pero más pequeño.

Aquí veremos el:



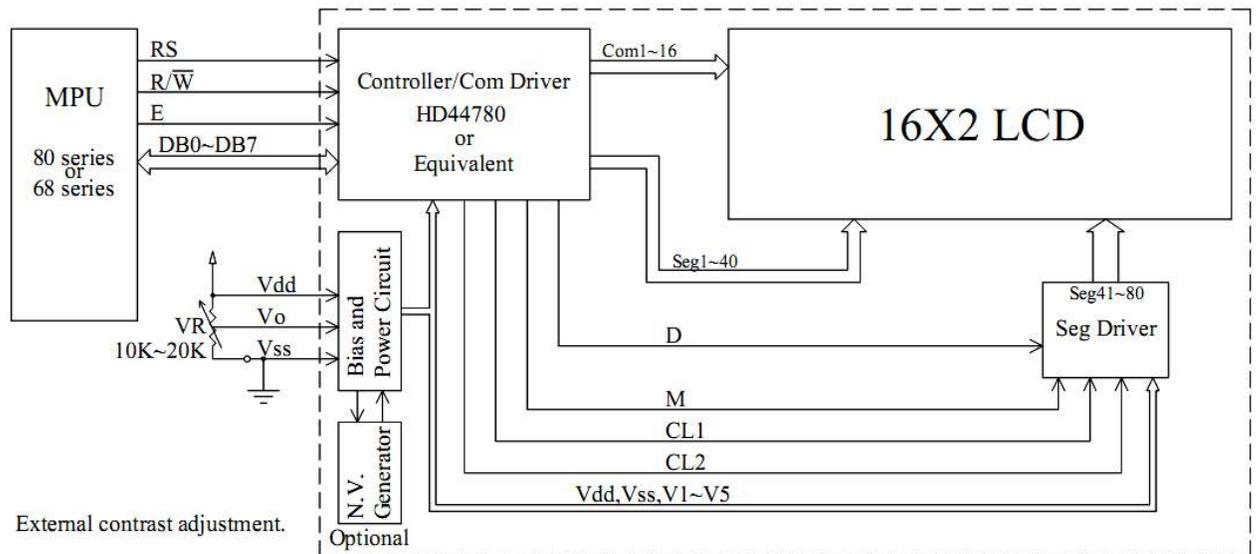
INTRODUCCION:

Cada vez que tengamos como objetivo utilizar un display LCD, lo primero es observar el circuito controlador.



Conociendo el nombre del LCD buscamos la hoja de datos:





En la hoja de datos del LCD o del controlador (**en este caso HD44780**), se describe la forma en que se establece comunicación con el LCD o controlador, de ese modo si la biblioteca que tenemos disponible para Arduino es compatible, podremos utilizarla (esa información estará en la documentación de la biblioteca). Si no sirve tendremos que crear nuestra propia rutina. Se aconseja la lectura informativa del apunte TEORIALCDV1.pdf

<http://www.bolanosdj.com.ar/MOVIL/ARDUINO2/TEORIALCDV1.pdf>

Por suerte la biblioteca de Arduino **LiquidCrystal** es compatible con el LCD actual.

Manejaremos mediante la opción de 4 bit el LCD, siempre es la mas conveniente.

Del apunte antes presentado EjemploLCD.pdf :

<http://www.bolanosdj.com.ar/MOVIL/ARDUINO2/EjemploLCD.pdf>

Rescatamos:

Vamos a conectar los pines de datos y control. Sin entrar en muchos detalles, no vamos a usar todos los pines disponibles, porque no los necesitamos. Solo usaremos dos pines de control, RS y EN y los 4 pines de datos D7, D6, D5, y D4 .

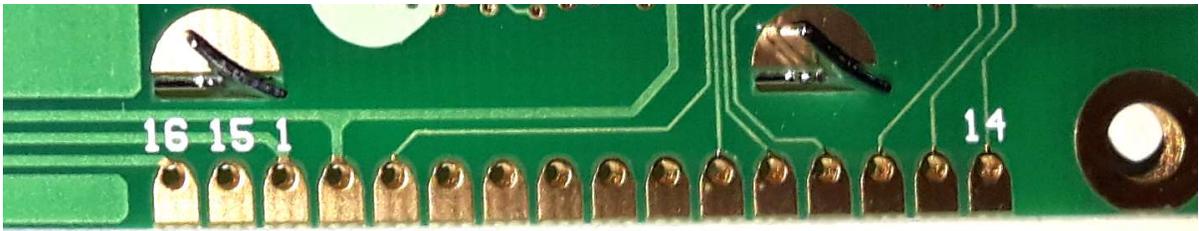
Vamos con las conexiones de control del display:

LCD	ARDUINO
RW, LCD pin 5	GND de Arduino
RS, LCD pin 4	Arduino pin 7
EN, LCD pin 6	Arduino pin 8

Y ya solo nos quedan los 4 cables de datos.

LCD	ARDUINO
DB7, LCD pin 14	Arduino pin 12
DB6, LCD pin 13	Arduino pin 11
DB5, LCD pin 12	Arduino pin 10
DB4, LCD pin 11	Arduino pin 9

Es importante para evitar malas conexiones observar la placa del LCD y la posición de los pines:



16	15	1	2	3	4	5	6	7	8	9	10	11	12	13	14
K	A	Vss	Vdd	Vo	RS	R/W	E	DB0	DB1	DB2	DB3	DB4	DB5	DB6	DB7
Gnd	V+	Gnd	5v	var											

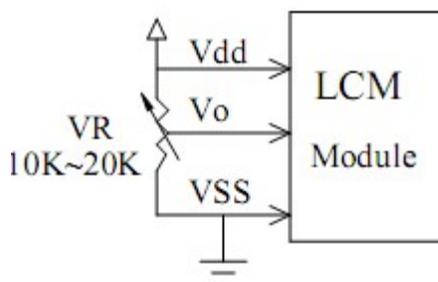
Ahora con la ayuda de la hojas de datos, establecemos la función de los pines:

7.Interface Pin Function

Pin No.	Symbol	Level	Description
1	V _{SS}	0V	Ground
2	V _{DD}	5.0V	Supply Voltage for logic
3	VO	(Variable)	Operating voltage for LCD
4	RS	H/L	H: DATA, L: Instruction code
5	R/W	H/L	H: Read(MPU→Module) L: Write(MPU→Module)
6	E	H,H→L	Chip enable signal
7	DB0	H/L	Data bus line
8	DB1	H/L	Data bus line
9	DB2	H/L	Data bus line
10	DB3	H/L	Data bus line
11	DB4	H/L	Data bus line
12	DB5	H/L	Data bus line
13	DB6	H/L	Data bus line
14	DB7	H/L	Data bus line
15	A	-	LED -
16	K	-	LED +

En el PIN 15 (V+) colocamos un a pequeña resistencia (10 ohms) en serie a 5v

En el PIN 3 (Vo) es el contraste, debemos colocar un potenciómetro o un divisor con 2 resistencias (Ej: 10K- 330 ohms).





Ejemplo de programa:

```
//Biblioteca necesaria para LCDs
#include <LiquidCrystal.h>

//Iniciamos los pines a utilizar
LiquidCrystal lcd(7, 8, 9, 10, 11, 12); //( RS, EN, D4, D5, D6, D7)

void setup() {
  // Aqui el codigo de configuracion, se ejecuta una sola vez:

  lcd.begin(16, 2); //Expresamos el numero de columnas y filas de nuestro LCD
}

void loop() {
  // Aqui el codigo principal que se va a correr repetidamente:

  lcd.setCursor(0, 0); // fijar cursor en columna 0, linea 0 (seria primera fila)
  lcd.print("Hola mundo");

  lcd.setCursor(0, 1); // fijar cursor en columna 0, linea 1 (seria segunda fila)
  lcd.print("Saludos..."); // Enviar el mensaje
}
}
```

Se puede consultar el resumen sobre la biblioteca LiquidCrystal para obtener información sobre las posibles funciones disponibles para comandar el LCD.

BibliotecaLiquidCristal.pdf

A continuación otro ejemplo de programa:

```

//Biblioteca necesaria para LCDs
#include <LiquidCrystal.h>

//Iniciamos los pines a utilizar
LiquidCrystal lcd(7, 8, 9, 10, 11, 12); //( RS, EN, D4, D5, D6, D7)

void setup() {
  // Aqui el codigo de configuracion, se ejecuta una sola vez:

  lcd.begin(16, 2); //Expresamos el numero de columnas y filas de nuestro LCD
}

void loop() {
  // Aqui el codigo principal que se va a correr repetidamente:

  lcd.setCursor(0, 0); // fijar cursor en columna 0, linea 0 (seria primera fila)
  lcd.print("Hola mundo");

  lcd.setCursor(0, 1); // fijar cursor en columna 0, linea 1 (seria segunda fila)
  lcd.print("Saludos..."); // Enviar el mensaje

  delay(5000);

  lcd.clear();// Borra la pantalla LCD y posiciona el cursor en la esquina superior izquierda.

  lcd.setCursor(0, 0); // fijar cursor en columna 0, linea 0 (seria primera fila)
  lcd.print("Mostrando en LCD");
  delay(4000);

  for (int i=0; i<16; i++) // declara i y prueba si es menor
                        //que 16, incrementa i.

  {

  lcd.scrollDisplayRight();// Desplaza el contenido de la pantalla (texto y el cursor) un espacio hacia la derecha
  delay(400);
  }

  lcd.clear();// Borra la pantalla LCD y posiciona el cursor en la esquina superior izquierda.

  lcd.setCursor(0, 0); // fijar cursor en columna 0, linea 0 (seria primera fila)
  lcd.print("Bienvenido al..");

  lcd.setCursor(0, 1); // fijar cursor en columna 0, linea 1 (seria segunda fila)
  lcd.print("mundo ARDUINO"); // Enviar el mensaje

  delay(5000);
  lcd.clear();// Borra la pantalla LCD y posiciona el cursor en la esquina superior izquierda.
}

```

Esquema Fritzing a continuación:

Fritzing es un programa libre de automatización de diseño electrónico que busca ayudar a diseñadores y artistas para que puedan pasar de prototipos (usando, por ejemplo, placas de pruebas) a productos finales.

Fritzing fue creado bajo los principios de Processing y Arduino, y permite a los diseñadores, artistas, investigadores y aficionados documentar sus prototipos basados en Arduino y crear esquemas de circuitos impresos para su posterior fabricación. Además, cuenta con un sitio web complementario que ayuda a compartir y discutir bosquejos y experiencias y a reducir los costos de fabricación. y su diseño de arte de artistas.



