

DISPLAY LCD

Un display LCD (Liquid Cristal Display) es una pequeña pantalla para mostrar caracteres a través de una matriz de puntos luminosos. Estos displays tienen unos pines para conectarlos a un microcontrolador, como Arduino. Nosotros usaremos uno de 16 columnas y 2 filas.

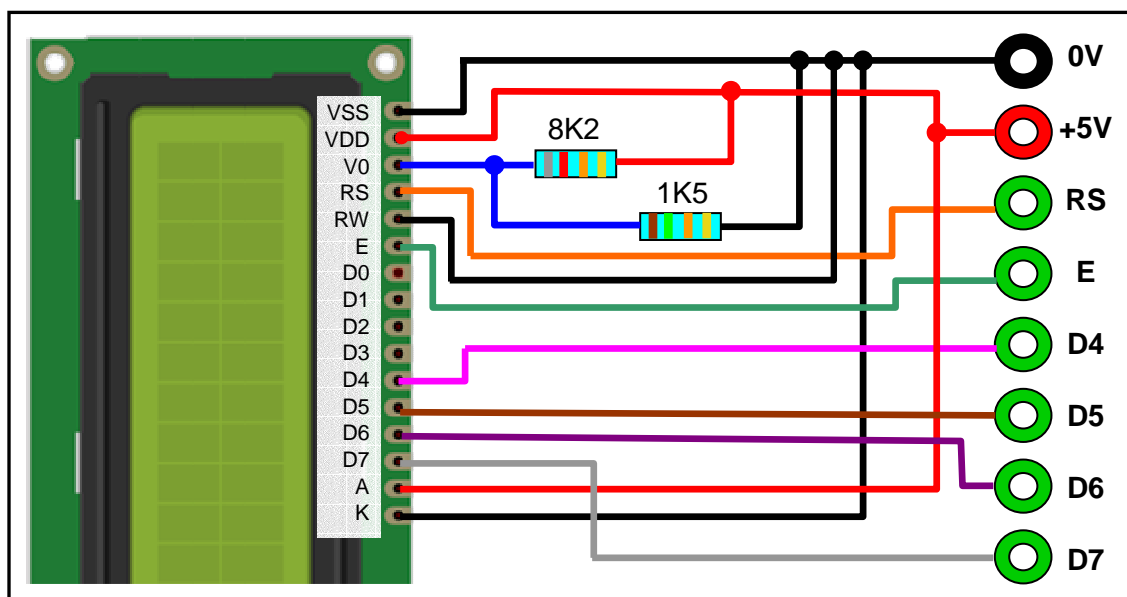
Normalmente, los displays tienen 16 pines. Nosotros no los vamos a usar todos.



Pin	Símbolo	Nivel
1	VSS	0V
2	VDD	+5V
3	V0	-
4	RS	HIGH/LOW
5	R/W	HIGH/LOW
6	E	HIGH/LOW
7	D0	HIGH/LOW
8	D1	HIGH/LOW
9	D2	HIGH/LOW
10	D3	HIGH/LOW
11	D4	HIGH/LOW
12	D5	HIGH/LOW
13	D6	HIGH/LOW
14	D7	HIGH/LOW
15	BLA	+5V
16	BLK	0V

Pines	Función
1-2	Alimentación de los circuitos electrónicos
3	Controla contraste del display. Tensión entre 0 y 5 V
4	Selecciona entre datos o instrucciones
5	Selecciona entre lectura (HIGH) o escritura (LOW)
6	Habilita o deshabilita la escritura en el display
7 al 14	Bus de datos de 8 bits. Solo usaremos 4.
15-16	Alimentación de la retroalimentación del display

Para nuestras prácticas usaremos el display dentro de una cajita con bornas enchufables a los entrenadores Arduino mediante cables de bananas. Algunas de las conexiones se realizan dentro de la caja y el nivel de contraste del display lo dejaremos fijo para simplificar.



LA LIBRERÍA LIQUIDCRYSTAL

Para usar el display con la tarjeta Arduino utilizaremos la librería LiquidCrystal que ya viene incluida en el entorno integrado de Arduino y que dispone de varias funciones.

Ejemplo 1: Vamos a hacer que aparezca en la segunda fila del display la frase “EL CAJON DEL ARDU”.



```
// Incluimos la librería LiquidCrystal
#include <LiquidCrystal.h>

// Inicializamos la librería con los números
// de los pines de Arduino que usaremos.
// LiquidCrystal (RS, E, D4, D5, D6, D7)
LiquidCrystal lcd(7,6,5,4,3,2);

void setup(){
  // Indicamos el número de columnas y filas del display
  lcd.begin(16,2);
  // Limpiamos la pantalla
  lcd.clear();
  // Situamos el cursor en la primera columna de la segunda fila
  lcd.setCursor(0,1);
  lcd.print("EL CAJON DE ARDU");
}
void loop(){}
```

Principales funciones de la librería LiquidCrystal

La librería Servo dispone de varias funciones. Las que más usaremos son:

▪ LiquidCrystal

Crea un objeto tipo LiquidCrystal para controlar el display LCD.

La sintaxis es: **LiquidCrystal** lcd ($n_1, n_2, n_3, n_4, n_5, n_6$)

donde *lcd* es el nombre que le damos al objeto tipo LiquidCrystal creado y n_1 a n_6 son los números de pines de Arduino donde irán conectados los pines RS, E, D4, D5, D6 y D7 del display LCD.

▪ .begin()

Especifica el número de columnas y de filas del display.

La sintaxis es: lcd.**begin** (n_C, n_F)

donde *lcd* es un objeto tipo LiquidCrystal, n_C es el número de columnas y n_F el número de filas del display. En nuestro caso siempre serán 16 y 2 respectivamente.

- **.setCursor()**

Ubica el cursor en la columna y fila especificadas como parámetros. La primera columna y la primera fila se indexan como 0. Por tanto, la posición situada arriba a la izquierda es la (0,0).

La sintaxis es: `lcd.setCursor (C,F)`

donde *lcd* es un objeto tipo `LiquidCrystal`, *C* es el número de la columna y *F* el número de la fila donde se posiciona el cursor. *C* será un número entre 0 y 15 y *F* un número entre 0 y 1.

- **.print()**

Escribe en el display el carácter, la cadena de caracteres, el número o el valor de la variable que se indica como parámetro.

La sintaxis es: `lcd.print (contenido)`

donde *lcd* es un objeto tipo `LiquidCrystal` y *contenido* es lo que se pretende escribir en el display.

Ejemplos:

```
lcd.print ("Hola, mundo");
lcd.print ('a');
lcd.print (variable);
lcd.print (3.14);
```

Redondeo de decimales: Cuando se quiere imprimir una variable float o un número con decimales, podemos añadir un segundo parámetro que indica el número de decimales con que se redondeará el dato al imprimirlo. Si no se indica nada, por defecto es 2. Por ejemplo, si el contenido de *variable* es 27.863, cualquiera de las instrucciones:

```
lcd.print (variable,1);
lcd.print (27.863,1);
```

haría que se imprimiera en el display 27.9

- **.clear()**

Borra la pantalla del display.

La sintaxis es: `lcd.clear ()`

donde *lcd* es un objeto tipo `LiquidCrystal`.

- **.autoscroll() y .noAutoscroll()**

Activa y desactiva, respectivamente, el desplazamiento automático del texto de derecha a izquierda. La sintaxis es: `lcd.autoscroll ()`

`lcd.noAutoscroll ()`

donde *lcd* es un objeto tipo `LiquidCrystal`.

Ejemplo 2: Escribe un texto en pantalla y posteriormente lo desplaza de forma indefinida.

```
#include <LiquidCrystal.h>

// LiquidCrystal (RS, E, D4, D5,D6, D7)
LiquidCrystal lcd(7,6,5,4,3,2);
| void setup(){

    lcd.begin(16,2);
    lcd.clear();
    // Situamos el cursor en la primera columna de la segunda fila
    lcd.setCursor(0,0);
    lcd.print("No hay plaza en este momento");
}
void loop(){
    lcd.setCursor(16,1); // situamos el cursor fuera del display
    lcd.autoscroll();
    lcd.print(" "); // imprimimos un espacio
    delay(500);
}
```

- **.noDisplay()**

Apaga la pantalla del display.

La sintaxis es: `lcd.noDisplay ()`

donde *lcd* es un objeto tipo LiquidCrystal.

- **.display()**

Enciende la pantalla del display que previamente ha sido apagada con noDisplay().

La sintaxis es: `lcd.display ()`

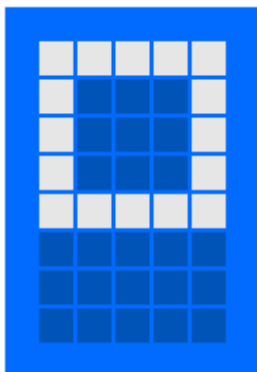
donde *lcd* es un objeto tipo LiquidCrystal.

Ejemplo 3: Escribe un texto en pantalla y posteriormente lo hace parpadear cada 0,5 segundos.

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(7,6,5,4,3,2);
void setup(){
  lcd.begin(16,2);
  lcd.print("Hola mundo");
}
void loop(){
  lcd.noDisplay();//apago el display
  delay(500);
  lcd.display();//enciendo el display
  delay(500);
}
```

Creación de caracteres con LiquidCrystal

A veces necesitamos utilizar caracteres que no están definidos en la librería. Podemos crearlos y utilizarlos fácilmente. Cada carácter del display está formado por una matriz de 40 puntos distribuidos en 8 filas y 5 columnas. Podemos definir, por ejemplo, el carácter de la figura con:



O sea, se coloca un 1 en los puntos que se tienen que iluminar y un 0 en los que tienen que estar apagados.

```
byte micaracter1[8]={
  B11111,
  B10001,
  B10001,
  B10001,
  B11111,
  B00000,
  B00000,
  B00000
};
```

A continuación hay que registrar el carácter creado y asociarlo a un número de 0 a 7, por ejemplo a un 1, con la función `.createChar()`

Ejemplo: `lcd.createChar(1, micaracter1);`

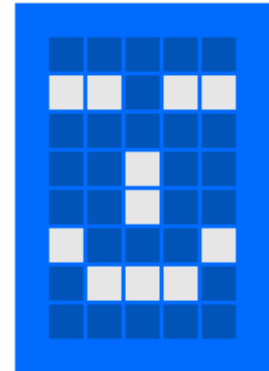
Por último, para imprimir el carácter creado, se usa la función `.write()`.

Esta función escribe un carácter en el display (no cadenas de texto, como `.print()`).

Ejemplo: `lcd.write(1);`

Ejemplo 4: Este programa, crea el carácter personalizado de la figura y luego lo imprime

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
byte customChar[8] = { // definicion del nuevo caracter
    B00000,
    B11011,
    B00000,
    B00100,
    B00100,
    B10001,
    B01110,
    B00000
};
void setup() {
    lcd.begin(16, 2);
    lcd.print("Nuevo char: ");
    lcd.createChar(1, customChar); // registro el nuevo caracter
    lcd.write(1); // escribo el nuevo caracter en pantalla
}
void loop() {}
```



Nota: Si asociamos el carácter creado al número 0, el compilador suele dar error y hay que definirlo como un byte. Es decir, habría que poner:

```
lcd.createChar(0, customChar); // registro el nuevo caracter
lcd.write(byte(0)); // escribo el nuevo caracter en pantalla
```

Impresión de variables con LiquidCrystal

Cuando queremos imprimir el valor de variables se utiliza la función .print(), igual que con las cadenas de texto. Por ejemplo, si tenemos una variable que se llama temperatura, tan solo hay que ejecutar:

```
lcd.print(temperatura);
```

Si queremos imprimir cadenas de texto y variables mezcladas, podemos hacerlo utilizando variables de tipo String.

Ejemplo 5: Programa que lee un nivel de luminosidad de una LDR e imprime su valor en el display con intervalos de 1 segundo.

```
#include <LiquidCrystal.h>
String cadena1 = "Luminosidad ";
int lumin=0;
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
void setup() {
    lcd.begin(16, 2);
}
void loop() {
    lumin=analogRead(A0); //LDR en pin A0
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(cadena1+lumin);
    delay(1000);
}
```