

## Introducción

Los display de cristal líquido de cuarzo (LCD) son pantallas que muestran caracteres alfanuméricos y símbolos, se puede configurar mediante el control de un microprocesador de 4 a 8 bits, funciones como RAM, generador de caracteres, entre otras se encuentran internas en un chip, se puede interconectar un sistema mínimo con este controlador, la composición de estos displays están formados por 6 niveles: film de filtro vertical, sustrato de vidrio con electrodos de óxido de indio, cristal líquido, sustrato de vidrio con film electrodo común, film de filtro horizontal y una superficie reflectante.

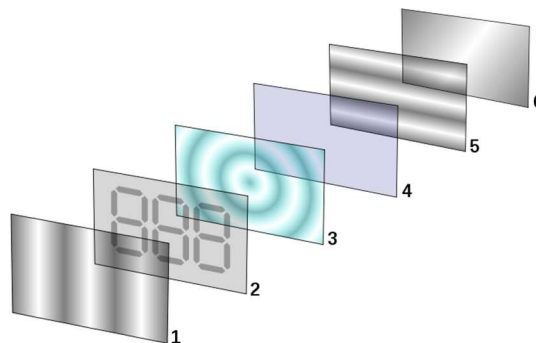


Figura 1. Partes del display 16x2

Como podemos observar en la imagen 2 tenemos tres tipos de pines de alimentación, control los cuales se clasifican en selección de registro (RS) donde el control o instrucción se ve con "0" y la selección con "1", de misma manera tenemos (RW) que permite escribir "1" o leer un dato "0" dentro de esta sección de pines tenemos enable el cual si encontramos un "1" significa que podemos leer o escribir, finalmente dentro de la estructura tenemos los pines de datos.

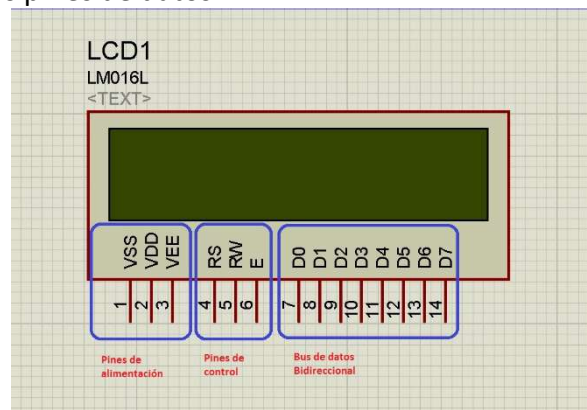


Figura 2. Estructura de un LCD

Para el uso de este dispositivo es fundamental conocer algunos aspectos del controlador LCD HD44780 tiene dos registros de 8 bits, un registro de instrucciones, donde almacena código de instrucciones e información de direcciones para la visualización RAM de datos (DDRAM) y generador de caracteres (CGRAM) y solo se puede escribir desde el MPU, por otro lado tenemos un registro de datos el cual almacena temporalmente datos para ser escritos en el RAM de datos o CGRAM esta acción es automática mediante una operación interna. La transferencia de datos se completa cuando la MPU lee el registro de datos, DR. Después de la lectura, los datos en DDRAM o CGRAM en la siguiente dirección se envían al DR para la siguiente lectura desde el MPU mediante una señal del selector de registro estos dos registros se pueden seleccionar.

El RAM de datos (DDRAM) almacena datos representados en códigos de caracteres de 8 bits y su capacidad máxima es de 80 caracteres, el área no utilizada puede usarse como RAM de datos generales al igual que ajusta la dirección y posteriormente es recibida después de este ajuste la dirección es enviada y recibida. Los caracteres de la memoria se presentarán en la pantalla

Generador de caracteres (CGRAM) es usado para reescribir patrones de caracteres por programa, en esta configuración los códigos de caracteres en las direcciones muestran los caracteres almacenados en CGRAM. Almacena los caracteres propios creados por cada usuario.

Generador de caracteres ROM (CGROM) es una memoria interna donde se almacena una tabla con los caracteres que podemos visualizar en el LCD.

Existen diferentes maneras de comunicación de este dispositivo como la comunicación en paralelo, serial a paralelo o la comunicación serial UART utilizando el puerto TX y RX de un microcontrolador. Esta puede ser utilizada con un baud entre 2400 y 38400 bps.

### **Análisis y descripción del problema**

A través de una máquina de estados del controlador se enviarán comandos para especificar la configuración que deseamos como, borrar, posicionamiento de una localidad específica, escribir algún carácter entre otras.

Para la modificación de patrones de caracteres se puede seguir la siguiente secuencia de pasos:

1. Determinar y relacionar los códigos y patrones de caracteres
2. Crear una lista entre las direcciones y los datos del EPROM.
3. Programar los patrones de caracteres en el EPROM
4. Enviar el EPROM al display
5. El procesamiento informático en el EPROM se realiza en LCD para crear una lista de patrones de caracteres, que se envía al usuario.
6. En caso de no presentar problemas con el listado se crea un LSI (Low speed internal, un reloj de baja velocidad interno en el microprocesador) y muestras se envían al usuario para su evaluación. Cuando el usuario confirma que los patrones de caracteres están correctamente escritos, la producción en masa de la LSI procede en display.

### Desarrollo de la propuesta de solución.

Para poder controlar la LCD de 16x2 se hará uso de la tarjeta DE10-lite y se programará en VHDL con el uso de la herramienta Quartus. Para comenzar con el desarrollo del controlador el cual tendrá como objetivo el desplegar 4 distintos mensajes a partir del habilitado de 4 diferentes switches. Para esto se quiere implementar estados para poder configurar la LCD y posterior a esto poder escribir en la misma.

Se tiene que tener conocimiento del cómo funciona la LCD, el funcionamiento de esta se da a partir del controlador HD44780U el cual es ampliamente utilizado en las LCD más comerciales que existen. Analizando la hoja de datos del controlador HD44780U podemos encontrar que es un controlador alfanumérico el cual consta de dos tipos de memorias la DDRAM en la cual se almacenaran todos los caracteres que se pueden visualizar en la pantalla y la CGROM encargada de generar dichos caracteres. En la hoja de datos podemos encontrar información de cómo podemos controlar la pantalla dicha información se nos presenta en forma de tabla con las diferentes configuraciones para el control de la LCD.

Table 6 Instructions

Instruction	Code										Description	Execution Time (max) (when $f_{cp}$ or $f_{osc}$ is 270 kHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear display	0	0	0	0	0	0	0	0	0	1	Clears entire display and sets DDRAM address 0 in address counter.	
Return home	0	0	0	0	0	0	0	0	1	—	Sets DDRAM address 0 in address counter. Also returns display from being shifted to original position. DDRAM contents remain unchanged.	1.52 ms
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction and specifies display shift. These operations are performed during data write and read.	37 $\mu$ s
Display on/off control	0	0	0	0	0	0	1	D	C	B	Sets entire display (D) on/off, cursor on/off (C), and blinking of cursor position character (B).	37 $\mu$ s
Cursor or display shift	0	0	0	0	0	1	S/C	R/L	—	—	Moves cursor and shifts display without changing DDRAM contents.	37 $\mu$ s
Function set	0	0	0	0	1	DL	N	F	—	—	Sets interface data length (DL), number of display lines (N), and character font (F).	37 $\mu$ s

Figura 3. Algunas instrucciones de control para LCD

En la tabla anterior podemos observar que para el control de la LCD requerimos 8 bits de control (DB0-DB7) y adicionalmente 3 bits (RS, RW, E) más para poder comunicar a la LCD que tipo de instrucción queremos mandar. Teniendo este conocimiento podemos pasar con la implementación de los estados para poder pasar a la elaboración del código en Quartus.

Se pretende utilizar ocho diferentes estados los cuales son **power\_up**, **initialize**, **WRT**, **SEND1**, **SEND2**, **SEND3**, **SEND4**, **CLR**. Para mandar los cuatro diferentes mensajes y dado que cada dígito se debe de enviar de forma individual por medio 8 bits se decidió formar 4 estados los cuales contendrán los mensajes a mandar a la LCD. Los estados restantes tendrán la función de:

**power\_up** : Asegurar el energizado correcto de la LCD, así como también un tiempo de espera para su correcta energización al final de esto manda a '0' RS y RW para la configuración en el siguiente estado.

**Initialize** : En este estado se configurara la LCD con el funcionamiento buscado por nosotros, como lo puede ser el número de renglones de la pantalla a utilizar, el sentido de escritura que la LCD adoptara (Izquierda o derecha) la limpieza de la LCD, encendido del cursor y el funcionamiento de este durante todo el proceso.

**WRT** : Este estado se encarga de la lectura de los cuatro switches a utilizar, cabe mencionar que en este estado y dependiendo del switch activado nos enviara a cual mensaje será desplegado en la LCD (SEND1 a SEND4).

**CLR** : En este último estado será el encargado de limpiar nuestra LCD para que el siguiente mensaje que se quiera desplegar se pueda escribir de manera correcta en la pantalla.

El funcionamiento del código hace uso de funciones IF y ELSIF, teniendo en cuenta que para la aplicación de las distintas configuraciones en la LCD se tiene que hacer uso de retardos para el buen envío de datos al controlador. Para poder implementar dichas interrupciones se habilita una entrada de CLK la cual estará internamente conectada al pin P11 de nuestra tarjeta el cual nos entrega un pulso de reloj de 50Mz. Tomando como ejemplo una pequeña parte de nuestro código tenemos que:

```

WHEN power_up =>
  IF(clk_count < (50000 * freq)) THEN --wait 50 ms
    clk_count := clk_count + 1;
    state <= power_up;
  ELSE --power-up complete
    clk_count := 0;
    rs <= '0';
    rw <= '0';
    state <= initialize;
  END IF;

```

En el estado power\_up hacemos uso de un IF para hacer un retardo de 50ms, se puede notar se nuestro IF se cumple siempre y cuando nuestro contador de ciclos de reloj que en este caso fue una variable previamente definida sea menor a la frecuencia la cual es una constante igualmente definida multiplicada por 50000, si lo vemos en término del periodo observamos que son 50ms, cuando el IF no se cumpla este pasara al ELSE y pondrá nuestro RS y RW a 0 para poder configurar nuestra LCD pasando al estado de “initialize”.

Esta misma estructura de retardos es utilizada para la escritura en la LCD, solo que en este caso RS se envía a ‘1’ y RW a ‘0’, estas condiciones son necesarias para poder escribir y lo podemos observar en la hoja de datos.

Instruction	Code										Description	Execution Time (max) (when $f_{cp}$ or $f_{osc}$ is 270 kHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Write data to CG or DDRAM	1	0	Write data								Writes data into DDRAM or CGRAM.	37 $\mu$ s $t_{ADD} = 4 \mu$ s*

Figura 4. Configuración para escribir en LCD.

En la escritura de datos, así como también en la configuración se debe de hacer uso del pin de “E” ya que este será el encargado de decirle a la LCD que lea la instrucción proporcionada por medio de los 8 bits y de RS y RW. Por lo cual cada vez que mandemos un dato al final de este tendremos que mandar un pulso al “E”.

Para poder desplegar los distintos mensajes como ya se menciona anteriormente se utilizo un estado en el cual solo se monitoreará el estado de los switches los cuales son los encargados de dirigirnos al estado de despliegue de cada mensaje, el estado es el siguiente.

```
WHEN WRT=>
  IF (S1='1') THEN
    state <= SEND1;
  ELSIF(S2='1') THEN
    state <= SEND2;
  ELSIF (S3='1') THEN
    state <= SEND3;
  ELSIF (S4='1') THEN
    state <= SEND4;
  ELSE
    state <= WRT;
  END IF;
```

Para poder llevar a cabo la práctica de manera física se realizaron las siguientes conexiones para el funcionamiento de la LCD:

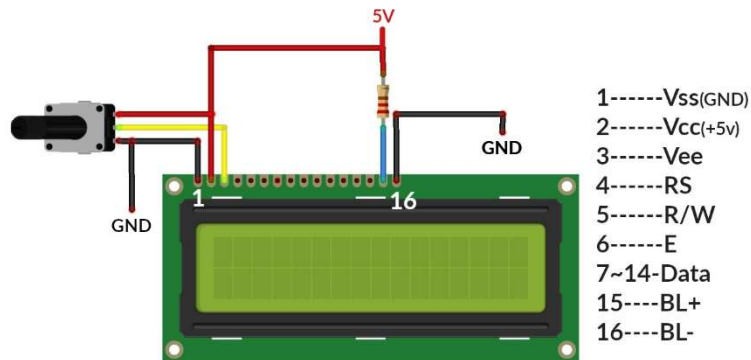


Figura 5. Conexiones para la LDC

Para los bits de datos se utilizará el puerto GPIO de nuestra tarjeta con las siguientes conexiones.

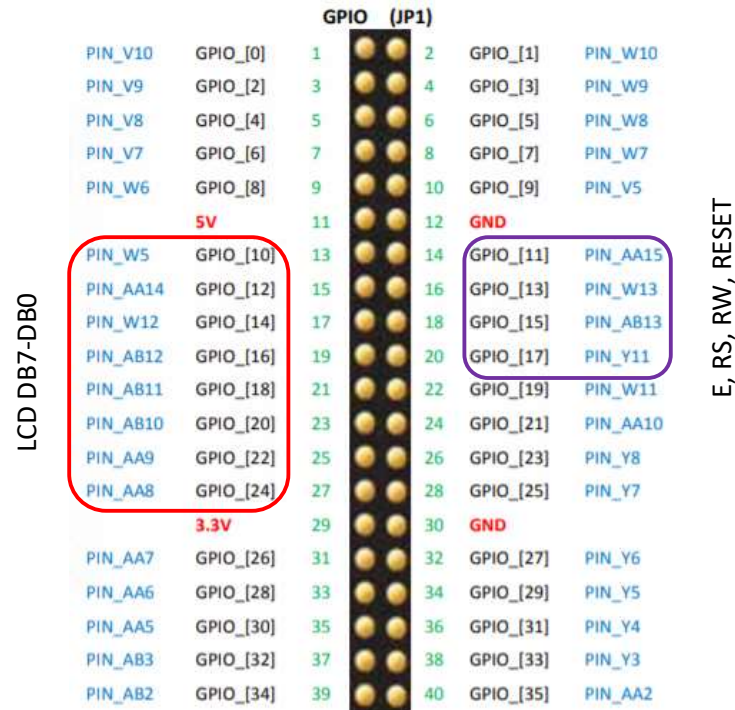


Figura 6, Conexiones para datos.

in	clk	Input	PIN_P11
out	e	Output	PIN_AA15
out	lcd_data[7]	Output	PIN_W5
out	lcd_data[6]	Output	PIN_AA14
out	lcd_data[5]	Output	PIN_W12
out	lcd_data[4]	Output	PIN_AB12
out	lcd_data[3]	Output	PIN_AB11
out	lcd_data[2]	Output	PIN_AB10
out	lcd_data[1]	Output	PIN_AA9
out	lcd_data[0]	Output	PIN_AA8
in	reset_n	Input	PIN_Y11
out	rs	Output	PIN_W13
out	rw	Output	PIN_AB13
in	S1	Input	PIN_C10
in	S2	Input	PIN_C11
in	S3	Input	PIN_D12
in	S4	Input	PIN_C12

Figura 7. Configuración de pines en Quartus.

Podemos observar el funcionamiento de nuestra implementación en el siguiente video:

<https://youtu.be/sqkXn3X56nw>

Y el código utilizado se puede encontrar en la siguiente liga:

[https://github.com/lapr98/Evidencia\\_2.git](https://github.com/lapr98/Evidencia_2.git)

## Conclusiones

El control de la LCD 16x2 es un buen ejemplo del manejo de memorias por medio de la programación de un FPGA en este caso con la tarjeta DE10-Lite. El conocimiento de la arquitectura de una maquina de estados a partir de un código de VHDL fue de mucha utilidad, ya que con la implementación de esta para el proceso de inicialización y escritura en la LCD fue de mayor facilidad, así como también proporciono el orden necesario para el buen funcionamiento de la LCD. El conocimiento de como crear retardos en el lenguaje de VHDL fue un reto en esta evidencia, ya que se tuvo que adquirir información sobre contadores implementados con flip-flops así como también de como poder desarrollar un divisor de frecuencia, ambas cosas necesarias para poder mandar los datos de manera correcta. Se obtuvo un mejor entendimiento y conocimiento referente al manejo de nuestra tarjeta DE10-Lite como por ejemplo en comportamiento de los puertos, así como también las funciones que pueden llegar a realizar como por ejemplo el pin P11 en cual es capaz de entregarnos un puseo de reloj de 50MHz y puede ser conectado de manera interna a nuestra entrada configurada en la entidad del código.

## Bibliografía consultada

- Introducción a las LCD. (s. f.). Recuperado de [https://www.5hertz.com/index.php?route=tutoriales/tutorial&tutorial\\_id=9](https://www.5hertz.com/index.php?route=tutoriales/tutorial&tutorial_id=9)
- Display de cristal líquido (LCD), Definición, Características y Conexión. (2017, febrero 14). IngenieríaElectrónica. <https://ingenieriaelectronica.org/display-de-cristal-liquido-lcd-definicion-caracteristicas-y-conexion/>
- HD44780U pdf, HD44780U description, HD44780U datasheets, HD44780U view: ALLDATASHEET ::: (s. f.). Recuperado de <https://pdf1.alldatasheet.es/datasheet-pdf/view/63663/HITACHI/HD44780U.html>