

EJEMPLO DE MARQUESINA EN LCD CON VHDL Y LA NEXYS 2

Juan Antonio Jaramillo Gómez, Dr.
jantonioj@yahoo.com.
Mirna Salmerón Guzmán, M. en C.
yupisg2@yahoo.com.mx.
Isis Chetzyl Ballardo Rodríguez, Dra.
chetzyl@hotmail.com

INSTITUTO POLITECNICO NACIONAL
UPIITA

Boletín No. 81
1o. de noviembre de 2020

Resumen

Ahora se presenta en este artículo un ejemplo de uso de un LCD de 16x2 para poner una marquesina que se desplaza a la izquierda, utilizando una estructura de memoria, conectado a la tarjeta de desarrollo Nexys 2 y programada con el lenguaje de descripción de hardware VHDL en el programa ISE de Xilinx. Se proporcionan los códigos VHDL y el archivo de restricciones de usuario para la tarjeta mencionada.

Abstract

Now, this article presents a 16x2 LCD example to put a marquee with shift left, using a memory structure, connected to the Nexys 2 development board and programmed with the VHDL hardware description language in the ISE program from Xilinx. VHDL codes and User Constrain File for the mentioned board are provided.

I. Introducción

Como ya se sabe, un Display de Cristal Líquido o LCD por sus siglas en inglés ("Liquid Crystal Display") es un elemento que sirve para visualizar datos, mensajes o información utilizando los caracteres ASCII grabados en su chip. La información puede ser fija o con movimiento, siendo esta última nombrada marquesina. El potencial de uso está en la imaginación del diseñador, siendo utilizados en máquinas despachadoras (dulces, gasolina, refrescos, etc.), en medidores (multímetros, termómetros, niveladores, etc.), sistemas de control específicos (temporizador y control de temperatura, medidores de nivel con alarmas, sistema de llenado de tinaco, etc.). En un ejemplo anterior, se presentó un control de un LCD con una máquina de estados finitos (FSM), ahora se presenta el control del LCD pero con

una estructura de memoria, es decir, todas las instrucciones de inicialización, datos y control se listan junto con el dato del tiempo para su ejecución al inicio del código, que posteriormente se llama para el manejo del LCD, lo cual al modificar dicho listado se puede cambiar el mensaje de la marquesina. Este es un código más pequeño que el hecho con FSM.

Solo para recordar, los tipos de LCD son:

- Por tamaño: 16x1, 16x2, 16x4, 20x2, etc.
- Por polarización: 3V ó 5V.
- Por la forma de comunicación: serie, paralelo.
- Por iluminación: con luz, sin luz de fondo, RGB, etc.

Aquí se utilizará un LCD hecho en una tarjeta perforada de prototipado rápido, pero también se puede utilizar el PmoCLP de Digilent, que trabaja con un bus paralelo de datos D7-D0 y sus terminales de control (RS, R/W, E), alimentado a 5V, mostrados en la figura 1, 2 y 3. Lo único que se debe de respetar son las conexiones físicas con el archivo de restricciones de usuario.

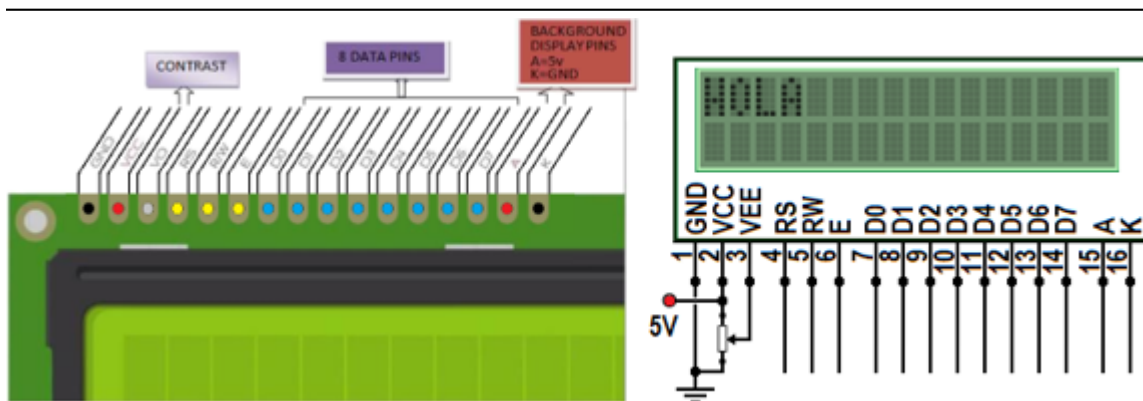


Figura 1. Terminales de un LCD de 16x2 y su conexión de contraste y polarización.

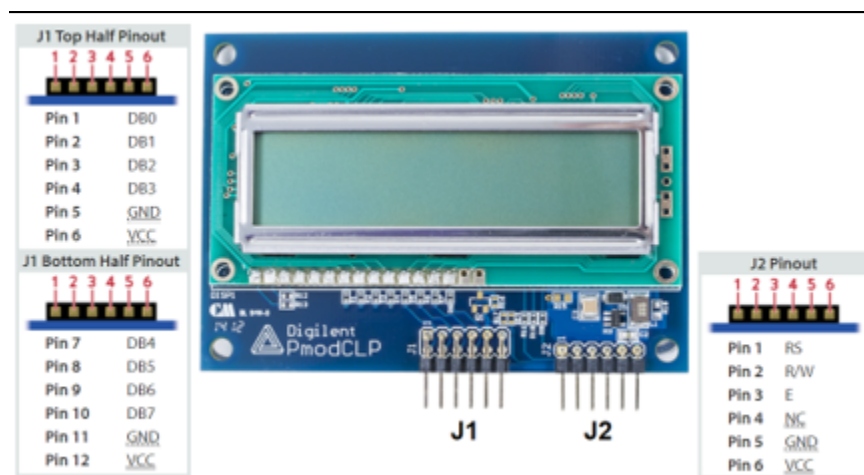


Figura 2. Distribución de terminales del PmodCLP.

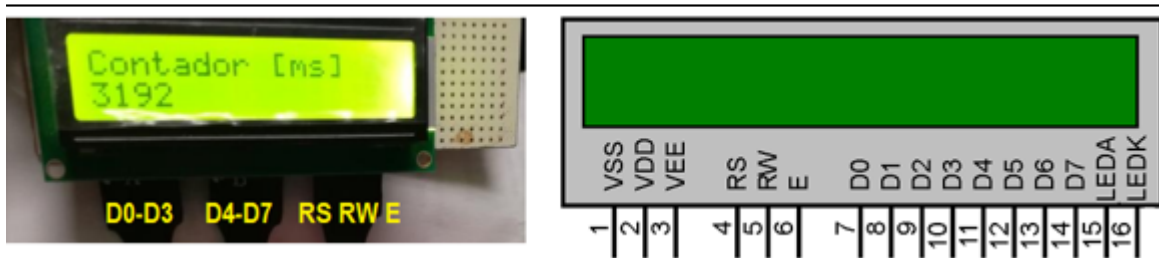


Figura 3. Ejemplo de un LCD hecho en una tarjeta perforada para la Nexys.

Conexiones

El LCD, tanto discreto como en módulo, se debe de conectar a los conectores de expansión (JA, JB, JC o JD) de la Nexys 2 (ver figura 4), recordando que la tarjeta puede entregar 3.3V o 5V y el debe de polarizarse de forma correcta a uno de estos voltajes. Otra consideración sería que si la tarjeta solo entrega 3.3V, y el display es de 5V, hay que poner un convertidor DC/DC, para elevar al voltaje adecuado del LCD. El PmodCLP y el módulo construido pueden trabajar a ambos voltajes, ya que tienen un convertidor DC/DC integrado.

Párrafo

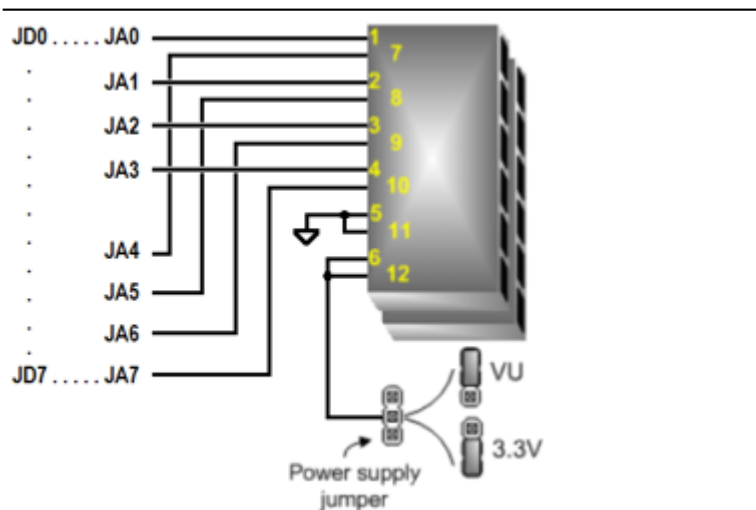


Figura 4. Terminales del conector de expansión de la nexys 2.

Se presenta en la figura 5 dos formas de la conexión de los terminales con el conector de expansión. El bus de datos se conecta al puerto JA o JB y el de control a la parte baja del conector JB (JB4-JB7) o JC (JC4-JC7). La luz de fondo siempre está conectada, aunque con un interruptor o un transistor se puede controlar su encendido.

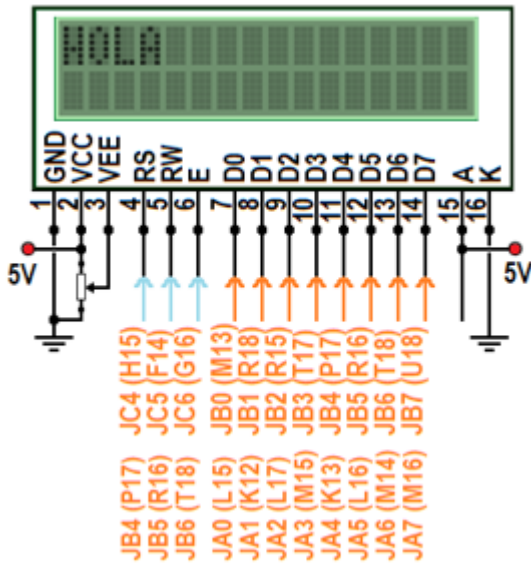


Figura 5. Conexión con los conectores externos JA o JB (datos) y JB o JC (control) de la Nexys 2.

Estructura de Memoria vs Máquina de estados finitos

En la figura 6 se presenta la parte fundamental de la estructura de memoria ROM que contiene todos los Datos/Comandos/Tiempos de control de la LCD, el código completo tiene 116 líneas, ya con comentarios.


```
type ROM_TYPE is array (0 to 35) of std_logic_vector(0 to 35);
signal ROM: ROM_TYPE:= (  --Power On Inicialización
  x"3881312D0", -- Espera 25ms(1312D0) y escribe 0x38
  x"38000000C", -- 12 clks después limpiar LCD_E ='0'
  x"380001388", -- 100us(1388)
  x"38000000C", -- 12 clks (C)
  -- Function Set 0x28
  x"3880007D0", -- 40us(7D0) D=0x38, Y=1000 index 4
  x"38000000C", -- 240ns(C) D=0x38, Y=0000
  -- Entry Mode Set 0x06
  x"0680007D0", -- 40us(7D0) D=0x06, Y=1000 index 6
  x"06000000C", -- 240ns D=0x06, Y=0000
  -- Display On 0x0C
  x"0C80007D0", -- 40us(7D0) D=0x0C, Y=1000 index 8
  x"0C000000C", -- 240ns D=0x0C, Y=0000
  -- Clear Display 0x01
  x"0180007D0", -- 40us(7D0) D=0x01, Y=1000 index 10
  x"01000000C", -- 240ns D=0x01, Y=0000
  -- Selecciona DDRAM (85 ler renglón, columna 6) 1er renglón
  x"858014050", -- 1.64ms D=0x85, Y=1000 index 12
  x"85000000C", -- 240ns D=0x85, Y=0000
  -- Escribiendo datos (55 = U ascii) ----- WRITE
  x"55CFFFFFFF", -- 335ms D=0x55, Y=1100 index 14
  x"55400000C", -- 240ns D=0x55, Y=0100
  -- Escribiendo datos (50 = P ascii)
  x"50CFFFFFFF", -- 335ms D=0x50, Y=1100 index 16
  x"50400000C", -- 240ns D=0x50, Y=0100
  -- Escribiendo datos (49 = I ascii)
  x"49CFFFFFFF", -- 335ms D=0x49, Y=1100 index 18
  x"49400000C", -- 240ns D=0x49, Y=0100
  -- Escribiendo datos (49 = I ascii)
  x"49CFFFFFFF", -- 335ms D=0x49, Y=1100 index 20
  x"49400000C", -- 240ns D=0x49, Y=0100
  -- Escribiendo datos (54 = T ascii)
  x"54CFFFFFFF", -- 335ms D=0x54, Y=1100 index 22
  x"54400000C", -- 240ns D=0x54, Y=0100
  -- Escribiendo datos (41 = A ascii)
  x"41CFFFFFFF", -- 335ms D=0x41, Y=1100 index 24
  x"41400000C", -- 240ns D=0x41, Y=0100
  -- Selecciona DDRAM (C7 2do renglón, columna 14) 2do renglón
  x"CD80007D0", -- 40us(7D0) D=0xCC, Y=1000 index 26
  x"CD000000C", -- 240ns D=0xCC, Y=0000
  -- Escribiendo datos (69 = i ascii)
  x"69CFFFFFFF", -- 335ms D=0x69, Y=1100 index 28
  x"69400000C", -- 240ns D=0x69, Y=0100
  -- Escribiendo datos (70 = p ascii)
  x"70CFFFFFFF", -- 335ms D=0x70, Y=1100 index 30
  x"70400000C", -- 240ns D=0x70, Y=0100
  -- Escribiendo datos (6E = n ascii)
  x"6ECFFFFFFF", -- 335ms D=0x6E, Y=1100 index 32
  x"6E400000C", -- 240ns D=0x6E, Y=0100
  -- Shift left (18)
  x"188FFF000", -- 335ms D=0x8, Y=1000 index 34
  x"18000000C" -- 240ns D=0x8, Y=0000
); --
```

```
type ROM_TYPE is array (0 to 35) of std_logic_vector(0 to 35);
signal ROM: ROM_TYPE:= (  --Power On Inicialización
  x"3881312D0", -- Espera 25ms(1312D0) y escribe 0x38
```

```
type ROM_TYPE is array (0 to 35) of std_logic_vector(0 to 35);
signal ROM: ROM_TYPE:= (  --Power On Inicialización
  x"3881312D0", -- Espera 25ms(1312D0) y escribe 0x38
  x"38000000C", -- 12 clks después limpiar LCD_E ='0'
  x"380001388", -- 100us(1388)
  x"38000000C", -- 12 clks (C)
  -- Function Set 0x28
  x"3880007D0", -- 40us(7D0) D=0x38, Y=1000 index 4
  x"38000000C", -- 240ns(C) D=0x38, Y=0000
  -- Entry Mode Set 0x06
  x"0680007D0", -- 40us(7D0) D=0x06, Y=1000 index 6
  x"06000000C", -- 240ns D=0x06, Y=0000
  -- Display On 0x0C
  x"0C80007D0", -- 40us(7D0) D=0x0C, Y=1000 index 8
  x"0C000000C", -- 240ns D=0x0C, Y=0000
  -- Clear Display 0x01
  x"0180007D0", -- 40us(7D0) D=0x01, Y=1000 index 10
  x"01000000C", -- 240ns D=0x01, Y=0000
  -- Selecciona DDRAM (85 ler renglón, columna 6) 1er renglón
  x"858014050", -- 1.64ms D=0x85, Y=1000 index 12
  x"85000000C", -- 240ns D=0x85, Y=0000
  -- Escribiendo datos (55 = U ascii) ----- WRITE
  x"55CFFFFFFF", -- 335ms D=0x55, Y=1100 index 14
  x"55400000C", -- 240ns D=0x55, Y=0100
  -- Escribiendo datos (50 = P ascii)
  x"50CFFFFFFF", -- 335ms D=0x50, Y=1100 index 16
  x"50400000C", -- 240ns D=0x50, Y=0100
  -- Escribiendo datos (49 = I ascii)
  x"49CFFFFFFF", -- 335ms D=0x49, Y=1100 index 18
  x"49400000C", -- 240ns D=0x49, Y=0100
  -- Escribiendo datos (49 = I ascii)
  x"49CFFFFFFF", -- 335ms D=0x49, Y=1100 index 20
  x"49400000C", -- 240ns D=0x49, Y=0100
  -- Escribiendo datos (54 = T ascii)
  x"54CFFFFFFF", -- 335ms D=0x54, Y=1100 index 22
  x"54400000C", -- 240ns D=0x54, Y=0100
  -- Escribiendo datos (41 = A ascii)
  x"41CFFFFFFF", -- 335ms D=0x41, Y=1100 index 24
  x"41400000C", -- 240ns D=0x41, Y=0100
  -- Selecciona DDRAM (C7 2do renglón, columna 14) 2do renglón
  x"CD80007D0", -- 40us(7D0) D=0xCC, Y=1000 index 26
  x"CD000000C", -- 240ns D=0xCC, Y=0000
  -- Escribiendo datos (69 = i ascii)
  x"69CFFFFFFF", -- 335ms D=0x69, Y=1100 index 28
  x"69400000C", -- 240ns D=0x69, Y=0100
  -- Escribiendo datos (70 = p ascii)
  x"70CFFFFFFF", -- 335ms D=0x70, Y=1100 index 30
  x"70400000C", -- 240ns D=0x70, Y=0100
  -- Escribiendo datos (6E = n ascii)
  x"6ECFFFFFFF", -- 335ms D=0x6E, Y=1100 index 32
  x"6E400000C", -- 240ns D=0x6E, Y=0100
  -- Shift left (18)
  x"188FFF000", -- 335ms D=0x8, Y=1000 index 34
  x"18000000C" -- 240ns D=0x8, Y=0000
); --
```

De los 9 nibbles de la señal, los dos primeros son el Dato (bus de datos de la LCD), el siguiente nibble es el de control de la LCD (LCD_E, LCD_RS, LCD_RW, 0), y los últimos 5 se utilizan para el contador de retardo.

```

-- Escribiendo datos (55 = U ascii) -----
9 nibbles x"55CFFFFFF", -- 335ms      D=0x55, Y=1100 index 14
          |
          | últimos 5 nibbles = contador de retardo  FFFFFFF -- 335ms
          | para CLK de 20ns
          | siguiente nibble = control de la LCD
          | (LCD_E, LCD_RS, LCD_RW, 0) =1100
          | dos primeros = Dato
          | (bus de datos de la LCD) D=0x55
    
```

En la figura 7 se tiene el diagrama de estados que involucra la inicialización (estados azules), ubicación en un renglón y columna determinado (estados verdes) y la escritura de datos en ASCII (estados rojos). Todos los datos encerrados entre paréntesis son datos en ASCII o son comandos de LCD y arroja un código de más de 650 líneas. Todos los caracteres ASCII se obtuvieron de la Tabla 1 y los comandos del datasheet de la LCD.

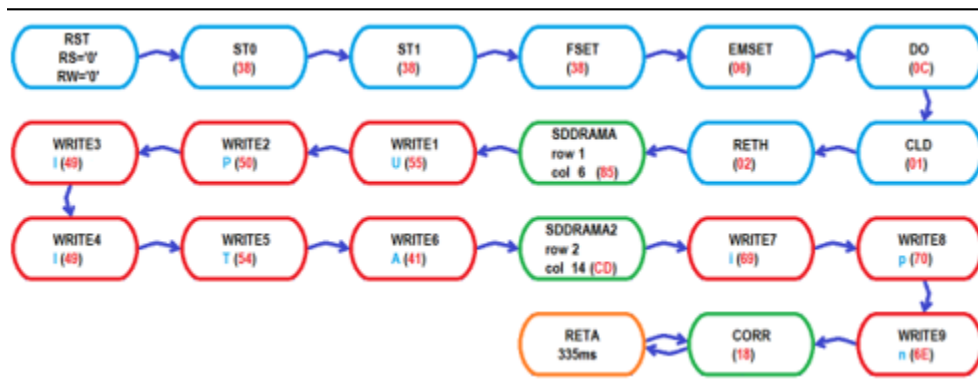


Figura 7. Diagrama de estados de la FSM para codificación ASCII.

Tabla. 1.
Símbolos ASCII.

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	NUL	STX	SOT	ETX	EOT	END	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
10	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
20	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL
80										¡	¢	£	¤	¥	¦	§
90	¨	©	ª	«	¬	­	®	¯	°	±	²	³	´	µ	¶	·
A0	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸
B0	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸
C0	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸
D0	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸
E0	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸
F0	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸	¸

El código escrito y sintetizado se implementó en la tarjeta Nexys 2, cuyos resultados se muestran en la figura 8.



Figura 8. Fotos de las pruebas de la marquesina.

En la siguiente liga, se accesa el código en VHDL.

Dentro de las aplicaciones del LCD, como ya se había comentado anteriormente, se encuentra como interfaz para visualizar instrucciones, datos de medidores (presión, temperatura, báscula, fuerza, etc.), mensajes fijos y/o en movimiento, juegos, animaciones con caracteres especiales, etc.

V. Conclusiones

Se ha presentado la implementación de un código para la inicialización y control de una marquesina, cuya principal ventaja de de la estructura de memoria es la disminución de líneas comparada con una FSM, de más de 600 líneas de código a un poco más de 100 líneas. El código se ha probado en la tarjeta Nexys 2, pero es funcional para otras tarjetas solo respetando los tiempos de retardo. Se entregan los códigos en VHDL y el UCF de la Nexys 2 para que se utilicen, analicen y modifiquen como cada usuario lo requiera. Por último, se les da un panorama de distintos usos del display.

Referencias

1. Digilent (2020). *Pmod CLP: Character LCD with Parallel Interface(LIMITED TIME)*Recuperado el 5 de agosto de 2020, de https://reference.digilentinc.com/reference/pmod/pmodclp/start?_ga=2.147725943.1110548213.1595456878-834976756.1505838925
2. Digilent (2020). *Nexys 2 Reference Manual*.Recuperado el 5 de agosto de 2020, de <https://reference.digilentinc.com/reference/programmable-logic/nexys-2/reference-manual>
3. Xilinx (2020). *Spartan-3E FPGA Starter Kit Board User Guide*Chapter 5, pp. 43-54. Recuperado el 5 de agosto de 2020, de https://www.xilinx.com/support/documentation/boards_and_kits/ug230.pdf
4. Flores Nava L. M., Reyes Barranca M.A., Arellano Cárdenas O. (2013). *Electrónica Digital CINVESTAV*. Recuperado el 5 de agosto de 2020, de http://www.vlsilab.cinvestav.mx/files/Practicas_Spartan_3E.pdf