

## MANEJO DE DISPLAY LCD (Versión 1.1)

Si bien muchas aplicaciones donde debemos mostrar información podemos resolverlas con display de 7 segmentos, estos presentan algunas limitaciones importantes, por ejemplo: no muestran caracteres alfanuméricos ni ASCII, y tienen un elevado consumo de corriente (recordemos que estamos encendiendo LEDs).

Los módulos LCD (display de cristal líquido) solucionan estos inconvenientes y tienen algunas ventajas adicionales como la facilidad que con que se pueden conectar a microprocesadores y microcontroladores, sumado a la óptima presentación de la información.

### CARACTERISTICAS PRINCIPALES DE LOS MODULOS LCD

Los módulos LCD se encuentran en diferentes presentaciones, por ejemplo 2x16 (2 líneas de 16 caracteres), 2x20, 4x20, 4x40, etc. Es conveniente buscar en catálogos para encontrar aquel que cumpla con las necesidades de nuestro proyecto.

La forma de utilizarlos y sus interfases (como se conectan) son similares. En la hoja de datos del display viene la distribución de pines, alimentación y el controlador (CI interno del modulo), etc. Es fundamental también buscar la hoja de datos de CI interno o controlador de LCD, allí encontraremos la información para operar con el mismo.

Describiremos a continuación como trabajar con el display 2x16 (FDCC1602E con controlador interno SPLC780) y se podrá considerar esta información genérica para displays similares.

Se muestra en la figura un display de este tipo.



Los pines de conexión de estos módulos incluyen un bus de datos de 8 bits, un pin de habilitación (E), un pin de selección que indica si lo que se esta enviando por el bus es un dato o una instrucción (RS) y un pin que indica si se va a leer o escribir en el módulo (R/W). La tabla a continuación describe en detalle los pines mencionados.

PIN	SÍMBOLO	Nombre y función
1	VSS	GND (Tierra 0V)
2	VDD	Alimentación +5V
3	VO	Ajuste del contraste
4	RS#	Selección DATO/CONTROL
5	RW#	Lectura o escritura en LCD
6	E	Habilitación
7	D0	D0 bit menos significativo
8	D1	D1
9	D2	D2
10	D3	D3
11	D4	D4
12	D5	D5
13	D6	D6
14	D7	D7 bit más significativo
15	LED+	Ánodo de LED backlight
16	LED-	Cátodo de LED backlight

**Nota: # significa negado.**

Según la operación que se desee realizar en el módulo LCD, los pines de control E, RS#, RW# deben tener un estado determinado. Además debe tener en el bus de datos un código que indique un carácter para mostrar en la pantalla o una instrucción de control para el display.

Los módulos LCD responden a un conjunto especial de instrucciones, estas deben ser enviadas por el microcontrolador o sistema de control al display, según la operación que se requiera. Se muestran a continuación el conjunto de instrucciones del modulo LCD.

CONTROL Y DATO	SEÑAL DE CONTROL		DATO / DIRECCIÓN								DESCRIPCIÓN	TIEMPO DE EJEC	
	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0			
Borrar pantalla	0	0	0	0	0	0	0	0	0	0	1	Limpia todo el display y retorna el cursor a la posición de inicio, dirección 0	
Cursor a casa	0	0	0	0	0	0	0	0	0	1	-	Retorna el cursor a la posición inicio (dirección 0). También retorna el display, desplazando a la posición original. Los contenidos de la DDRAM permanecen sin cambios.	
Seleccionar modo	0	0	0	0	0	0	0	0	1	I/D	S	Configura la dirección de movimiento y si se desplaza o no el display. Esta operación es realizada durante operaciones de lectura escritura.	
Encender/apagar pantalla	0	0	0	0	0	0	0	1	D	C	B	Configura el estado ON/OFF de todo el display (D), el cursor (C) y el parpadeo del caracter en la posición del cursor.	
Desplazar Cursor / Pantalla	0	0	0	0	0	0	1	S/C	R/L	-	-	Mueve el cursor y desplaza el display sin cambiar los contenidos de la DDRAM.	
Activar función	0	0	0	0	0	1	D/L	N	F	-	-	Configura el tamaño de la interfase (DL), el número de líneas del display (N) y la fuente del carácter (F) . N=0 es 1 línea. N=1 es 2 líneas.	
CG RAM	0	0	0	0	1	Dirección generador de RAM					Ajusta la dirección del generador de caracteres. El dato CG RAM es enviado y recibido después de este ajuste.		
DD RAM	0	0	0	1	Dirección de datos RAM					Ajusta la dirección de la DDRAM. La dirección es enviado y recibido después de este ajuste.			
Bandera de ocupado	0	0	BF	AC					Lectura de la bandera Busy Flag. Indicando que operaciones internas son realizadas y lectura de los contenidos del contador de direcciones.				
Escritura CG RAM/DD RAM	1	0	Escritura de dato								Escribe datos en la DDRAM o en la CGRAM		
Lectura CGRAM/DDRAM	1	1	Lectura de dato								Lectura de datos desde la DDRAM o la CGRAM		

NOTA: nótese que el pin RS# debe tomar el valor 0(cero) cuando lo que se va a enviar es una instrucción de control y debe tomar el valor 1(un) cuando lo que se va a enviar es un dato.

<b>SIGNIFICADO DE LAS ABREVIATURAS</b>	
I/D	= 1 incremental = 0 decrementa
S	= 1 desplaza el mensaje en la pantalla = 0 mensaje fijo en la pantalla
D	= 1 encender (activar) la pantalla = 0 apagar la pantalla (desactivar)
C	= 1 activar cursor = 0 desactivar cursor
B	= 1 parpadea caracter señalado por el cursor = 0 no parpadea el caracter
S/C	= 1 desplaza pantalla = 0 mueve el cursor
RL	= 1 desplazamiento a la derecha = 0 desplazamiento a la izquierda
DL	= 1 datos de 8 bits = 0 datos de 4 bits
BF	= 1 durante operación interna del módulo = 0 finaliza la operación interna

Nota: Cuando se va a cargar la dirección de la DDRAM donde se va a escribir el próximo caracter, nótese que el D7 siempre es 1. Por lo tanto cuando apunto a una dirección de memoria en el display hay que considerar este 1 adicional. Se verá con más detalle al describir el mapa de memoria del módulo LCD.

La interfase entre el microcontrolador y el LCD se puede hacer con el bus de datos del pic trabajando a 4 u 8 bits. Las señales de control trabajan de la misma forma en cualquiera de los dos casos, la diferencia se establece en el momento de iniciar el display, ya que existe una instrucción que permite establecer dicha configuración. O sea tenemos que avisarle al LCD que vamos a operar en 8 o a 4 bits.

Los caracteres que se envían al display se almacenan en la memoria RAM del módulo. Existen posiciones de memoria RAM, cuyos datos son visibles en la pantalla y otras que no son visibles, estas últimas se pueden utilizar para guardar caracteres que luego se desplazan a la zona visible.

Es importante anotar que solo se pueden mostrar caracteres ASCII de 7bits, por lo tanto algunos caracteres especiales no se pueden ver (es aconsejable tener a mano una tabla de caracteres ASCII para conocer los datos que son prohibidos). También se tiene la opción de mostrar caracteres especiales creados por el programador y almacenarlos en la memoria RAM que posee el módulo.

#### INTERFAZ CON MICROCONTROLADOR A 8 BITS

Esta forma de manejo es la mas sencilla de programar, pero tiene la desventaja de utilizar 8 pines del microcontrolador solo para el envío de datos y otros 2 pines para las señales de control.

En principio en la mayor parte de las aplicaciones se va requerir escribir en el LCD y rara vez leer en el mismo, por lo tanto en este apunte nos dedicaremos exclusivamente a escribir en el LCD. Esto

implica que el pin de selección de lectura/escritura (R/W) lo conectaremos siempre a tierra GND.

Utilizaremos el puerto B como bus de datos (manejará los pines D0 a D7 del LCD) y el puerto A se encarga de manejar las señales de control (manejará los pines E y RS del LCD).

Deberemos programar 2 subrutinas, una que llamaremos INSTRUC que será invocada cuando se quiera enviar una instrucción al módulo LCD, por ejemplo limpiar pantalla, indicar una posición de memoria, indicar si utilizaremos interfaz de 8 o 4 bits etc.

Y otra subrutina que llamaremos ESCRIB que será invocada cuando se quiera escribir un dato para ser visualizado en el módulo LCD.

Configuraremos entonces todo el puerto B como salida al igual que los pines RA0 y RA1 del puerto A, estando asignado cada pin del puerto como se indica a continuación:

Pin puertoB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
<b>Pin LCD</b>	<b>D7</b>	<b>D6</b>	<b>D5</b>	<b>D4</b>	<b>D3</b>	<b>D2</b>	<b>D1</b>	<b>D0</b>

Pin puertoA	RA4	RA3	RA2	RA1	RA0
<b>Pin LCD</b>				<b>E</b>	<b>RS</b>

RS: '1' DATO (o sea vamos a escribir) '0' CONTROL (o sea se va a enviar un instrucción al modulo LCD).

**RECORDAR:** Como solo vamos a escribir en pantalla, el pin R/W vale siempre 0 (cero), así que lo conectamos directamente a masa.

Evidentemente se debe configurar el puertoB del PIC como salida, y en el puertoA los pines RA0 y RA1 deben ser configurados como salidas. El resto de los pines del puertoA se puede configurar como se desee. Recuerde que estamos basando esta explicación para interface de 8 bits.

Lo primero que hay que hacer es mandar una serie de instrucciones al modulo LCD que constituyen la configuración del mismo. Luego recién podremos escribir el mensaje a presentar en pantalla.

**Las siguientes son una serie de instrucciones muy importantes (son independiente a que usemos interface de 4 u 8 bits).**

**ACTIVAR FUNCION (Instrucción)**

Se informa el tipo de interfaz que se va a usar, 4 o 8 bits. La cantidad de líneas (1 o 2). La fuente de caracter (5x8 dots o 5x10 dots).

<b>RS</b>	<b>RW#</b>	D7	D6	D5	D4	D3	D2	D1	D0
<b>0</b>	<b>0</b>	0	0	0	DL	N	F	-	-

DL= 0 interfaz de 4 bits.

DL= 1 interfaz de 8 bits.

N= 0 se va a usar 1 sola línea del display.  
N= 1 se van a usar 2 líneas del display.

F= 0 fuente de caracter 5x8 dots.  
F= 1 fuente de caracter 5x10 dots.

Entonces si enviamos al puertoB 30h

D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	1	0	0	0	1

DL= 1 interfaz de 8 bits.

N= 0 se va a usar 1 sola línea del display.

F= 0 fuente de caracter 5x8 dots.

Ahora se debe llamar a la subrutina INSTRU (esta subrutina la debemos crear nosotros, le podemos dar el nombre que queramos), la cual va a enviar 30h al puertoB, hacer RS=0 o sea le dice al LCD que lo que va a recibir es una instrucción (RA0= 0 del puertoA) y E=1 o sea habilita al LCD (RB1= 0 del puertoA), luego vuelve a hacer E=0 deshabilita.

#### **ACTIVAR DISPLAY (Instrucción)**

Configura el estado ON/OFF de todo el display, el estado del cursor y el parpadeo del caracter en la posición del cursor.

RS	RW#	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	1	D	C	B

RS se maneja con porta                      RW# se conecta a masa

Donde:

D = 1 enciende pantalla (activar)  
D = 0 apaga pantalla (desactivar)

C = 1 activar cursor  
C = 0 desactivar cursor

B = 1 parpadea caracter señalado por le cursor  
B = 0 no parpadea caracter.

Ej; si envió 0Ch al puertoB

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	1	1	0	0

Le estoy diciendo:  
D=1 encienda la pantalla  
C=0 desactivar cursor  
B=0 no parpadeo del caracter

Ahora se debe llamar nuevamente a la subrutina INSTRU (esta subrutina la debemos crear nosotros, le podemos dar el nombre que queramos), la cual va a enviar 0Ch al puertoB, hacer RS=0 o sea le dice al LCD que lo que va a recibir es una instrucción (RA0= 0 del puertoA) y E=1 o sea habilita al LCD (RB1= 0 del puertoA), luego vuelve a hacer E=0 deshabilita.

#### **BORRAR PANTALLA (Instrucción)**

Limpia pantalla y retorna el cursor al inicio (dirección 00h del display).

RS	RW#	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	1

Se invoca la subrutina INSTRUC.

#### SELECCIONAR MODO (Instrucción)

Selecciona el modo de desplazamiento del display y se desplaza o no el cursor.

RS	RW#	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	1	ID	S

S= 0 display no se desplaza, dato fijo en pantalla.

S= 1 display se desplaza.

ID= 1 incremento.

ID= 0 decremento.

En nuestro ejemplo enviaremos al puertoB 06h.

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	1	1	0

S= 0 dato fijo en pantalla

NOTAR que los unos en enviados a D2 y D1 carecen de sentido porque no hay desplazamiento.

Se invoca la subrutina INSTRUC.

Hasta aquí se han enviado las instrucciones previas para inicializar el módulo LCD y dejarlo listo para la escritura de datos en pantalla.

El envío de los datos se realiza por medio de los pines D0 a D7 del LCD, que se encuentran conectados a los pines RB0 a RB7 del puertoB del PIC. Por lo tanto los datos a escribir deben ser enviados a través del puertoB.

Un dato se envía por medio de su código ASCII, pero previamente se debe indicar al módulo la posición de memoria en que deseo escribir el dato que enviaré a continuación. Esto se hace por medio de la instrucción DDRAM.

#### DDRAM (Instrucción)

Le informa al módulo LCD la dirección de memoria en la cual se va almacenar el código del dato que se le enviará a continuación.

RS	RW#	D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	Posición de memoria a escribir dato						

Luego se invoca a subrutina INSTRUC, ya que es una instrucción para el LCD.

**NOTA: el 1 en D7 debe ir siempre.**

Aclaremos algo para no confundirnos. Supongamos que deseamos escribir 'A' en la posición de memoria 00h del LCD.

Como tenemos que asegurar el 1 en el D7 deberemos enviar al puertoB (recordar que D0 ... D7 del LCD esta conectado a RB0 ... RB7 del puertoB) en lugar de 00h el 80h, esto es para asegurar el 1 en el D7 (el len D7 hace 00000000 se convierta en 10000000 o sea 80h).

Así quedará en este caso:

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	0	0

Ahora que le informamos donde vamos a escribir el dato, deberemos enviarlo y aclararle que se esta enviando un dato, para lo cual necesitaremos una subrutina que justamente hará esto y que la llamaremos ESCRIB.

La secuencia de instrucciones sería así:

**MOVLW 80h;** carga en w dirección de memoria del LCD donde se va almacenar el dato

**CALL INSTRUC;** subrutina que gestiona la instrucción con el LCD- en este caso que tome lo que se esta enviando como una dirección de memoria

**MOVLW 'A';** este es el dato a presentar en pantalla

**CALL ESCRIB;** subrutina que gestiona el ingreso del dato al LCD a la posición de memoria antes enviada.



NOTA: En el apéndice se transcribe la solución completa al ejemplo de escribir en el display el mensaje 'HOLA MUNDO'. Se recomienda observar con detalle las subrutinas INSTRUC y ESCRIB que operan con interfaz de 4 bit.

Se utiliza PIC16F84.

### MAPA DE LA MEMORIA DEL MODULO LCD

#### MAPA DE LA MEMORIA DEL MODULO LCD (2 líneas x 16 caracteres)

00h	01h	02h	03h	04h	05h	06h	07h	08h	09h	0Ah	0Bh	0Ch	0Dh	0Eh	0Fh	10h	---	1Fh
40h	41h	42h	43h	44h	45h	46h	47h	48h	49h	4Ah	4Bh	4Ch	4Dh	4Eh	4Fh	50h	---	5Fh
<b>ÁREA VISIBLE</b>																<b>Area no visible - 16 posiciones</b>		

Como habíamos mencionado anteriormente, al enviar una dirección hay que asegurar el 1 en D7

Ej.

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	0	0

De este modo podríamos pensar en un mapa de memoria equivalente del módulo LCD como muestra la siguiente figura, en el caso de solo utilizar la zona visible:

80h	81h	82h	83h	84h	85h	86h	87h	88h	89h	8Ah	8Bh	8Ch	8Dh	8Eh	8Fh
C0h	C1h	C2h	C3h	C4h	C5h	C6h	C7h	C8h	C9h	CAh	CBh	CCh	CDh	CEh	CFh

## MANEJO DEL LCD CON INTERFAZ DE 4 BITS

Es el caso mas utilizado para aprovechar los puertos del PIC. Dejando disponibles pines de los mismos para otras funciones.

Por ejemplo si usamos el puertoB:

**RB4 RB5 RB6 RB7** respectivamente conectados a **D4 D5 D6 D7**

**RB0 RB1** a **RS** y **E** respectivamente

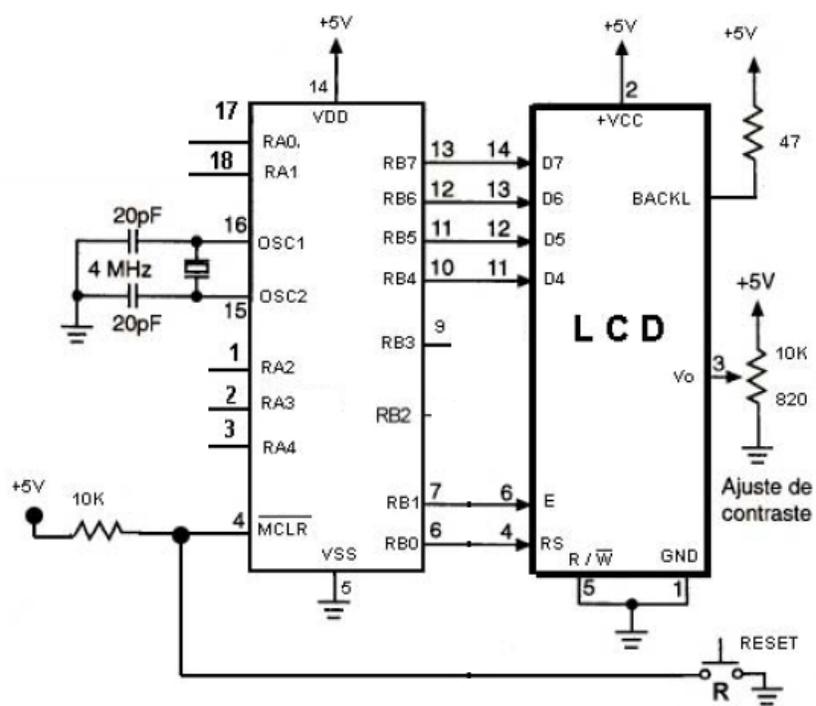
**RW** va a masa.

Las subrutinas INSTRUC y ESCRIB serán diferentes que para el caso de interfaz de 8 bits. Estas enviaran primero los 4bit MSB y luego los 4 bits LSB.

A continuación se muestra el conexionado para manejar un display LCD con interfaz de 4 bit y se transcribe un programa que presenta mensajes en el LCD con interfaz de 4 bits.

Archivo:

**programalcd3.txt**



**MANEJO DE LCD CON PIC 16F84**

## LCD MODELO 8X2 (Dos líneas de 8 caracteres)

El display de LCD de 8x2 modelo WH0802A se maneja de manera similar al LCD 16x2 modelo FDCC1602E presentado en la explicación anterior, si bien presentan un controlador interno diferente (KS066).



Podemos utilizar el mismo programa sugerido (**programalcd3.txt**), con la salvedad de tener en cuenta que el mapa de la RAM es menor:

### *Display Character Address Code :*

Display position	1	2	3	4	5	6	7	8
DD RAM Address	00	01						07
DD RAM Address	40	41						47

De este modo podríamos pensar en un mapa de memoria equivalente del módulo LCD como muestra la siguiente figura, en el caso de solo utilizar la zona visible:

80h	81h	82h	83h	84h	85h	86h	87h
C0h	C1h	C2h	C3h	C4h	C5h	C6h	C7h

Se puede obtener las hojas de datos de los display mencionados en:

[www.djbolanos.com.ar](http://www.djbolanos.com.ar)