

## SISTEMAS ELECTRÓNICOS Y AUTOMÁTICOS PRACTICAS DE MICROCONTROLADORES PIC

### PRÁCTICA 7:

### *El módulo LCD*

- 
- *El módulo LCD*
  - *Ejemplos de funcionamiento*
-

## 1. Objetivos:

- Conocer el funcionamiento y programación del módulo LCD del entorno de desarrollo EasyPIC
- Grabar programas-ejemplo sobre el dispositivo EasyPIC y comprobar su funcionamiento.

## 2. El módulo LCD

Este apartado está destinado a una breve descripción del funcionamiento del módulo LCD incluido en el sistema de evaluación EasyPIC.

### 2.1. Visualizador LCD

Las pantallas de cristal líquido LCD o display LCD para mensajes (*Liquid Cristal Display*) tienen la capacidad de mostrar cualquier carácter alfanumérico, permitiendo representar la información que genera cualquier equipo electrónico de una forma fácil y económica.

La pantalla consta de una matriz de caracteres (normalmente de 5x7 o 5x8 puntos) distribuidos en una, dos, tres o cuatro líneas de 16 hasta 40 caracteres cada línea.

El proceso de visualización es gobernado por un microcontrolador incorporado a la pantalla, siendo el Hitachi 44780 el modelo de controlador más utilizado.



Figura 1. LCD 2x16: está compuesto por 2 líneas de 16 caracteres

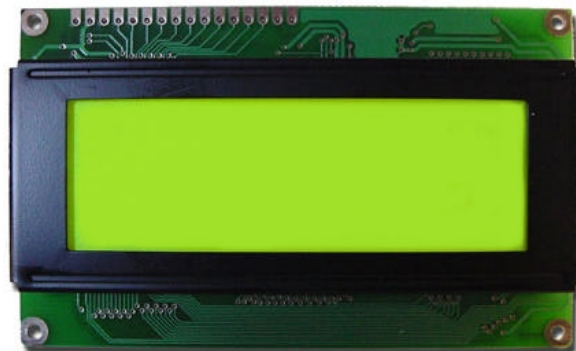


Figura 2. LCD 4x40: está compuesto por 4 líneas de 40 caracteres

La tarjeta EasyPIC dispone de un módulo LCD tipo **16x2**.

Las características generales de un módulo **LCD 16x2** son las siguientes:

- Consumo muy reducido, del orden de 7.5mW
- Pantalla de caracteres ASCII, además de los caracteres japoneses Kanji, caracteres griegos y símbolos matemáticos.
- Desplazamiento de los caracteres hacia la izquierda o a la derecha
- Memoria de 40 caracteres por línea de pantalla, visualizándose 16 caracteres por línea
- Movimiento del cursor y cambio de su aspecto
- Permite que el usuario pueda programar 8 caracteres
- Pueden ser gobernados de 2 formas principales:
  - Conexión con bus de 4 bits
  - Conexión con bus de 8 bits

## 2.2. Patillaje

A continuación se presenta la descripción de señales empleadas por el módulo LCD así como el número de patilla a la que corresponden.

PIN N°	SÍMBOLO	DESCRIPCIÓN
1	V <sub>SS</sub>	Patilla de tierra de alimentación
2	V <sub>DD</sub>	Patilla de alimentación de 5 V
3	V <sub>O</sub>	Patilla de contraste del cristal líquido. Normalmente se conecta a un potenciómetro a través del cual se aplica una tensión variable entre 0 y +5V que permite regular el contraste del cristal líquido.
4	RS	Selección del registro de control/registro de datos: RS=0 Selección del registro de control RS=1 Selección del registro de datos
5	R/W	Señal de lectura/escritura R/W=0 El módulo LCD es escrito R/W=1 El módulo LCD es leído
6	E	Señal de activación del módulo LCD: E=0 Módulo desconectado E=1 Módulo conectado
7-14	D0-D7	Bus de datos bi-direccional. A través de estas líneas se realiza la transferencia de información entre el módulo LCD y el sistema informático que lo gestiona

## 2.3. DDRAM

El módulo LCD posee una zona de memoria RAM llamada DDRAM (Data Display RAM) donde se almacenan los caracteres que se van a mostrar en la pantalla.

Tiene una capacidad de 80 bytes, 40 por cada línea, de los cuales sólo 32 se pueden visualizar a la vez (16 bytes por línea).

De las 80 posibles, las dos direcciones más importantes de la DDRAM son:

- Dirección 00h, que es el comienzo de la primera línea
- Dirección 40h, que es el comienzo de la segunda línea

## 2.4. CARACTERES DEFINIDOS EN LA CGROM

El LCD dispone de una zona de memoria interna no volátil llamada CGROM donde se almacena una tabla con los 192 caracteres que pueden ser visualizados.

Cada uno de los caracteres tiene su representación binaria de 8 bits. Para visualizar un carácter debe recibir por el bus de datos el código correspondiente.

Por ejemplo:

“A” → b'01000001'

Table 4 Correspondence between Character Codes and Character Patterns (ROM Code: A00)

Lower 4 Bits \ Upper 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
xxxx0000	CG RAM (1)			0	1	A	Q	a	q				-	9	3	α	ρ
xxxx0001	(2)		!	1	A	Q	a	q				.	7	4	ä	q	
xxxx0010	(3)		"	2	B	R	b	r				「	イ	ツ	×	β	θ
xxxx0011	(4)		#	3	C	S	c	s				」	ウ	テ	ε	ω	
xxxx0100	(5)		\$	4	D	T	d	t				、	エ	ト	†	μ	Ω
xxxx0101	(6)		%	5	E	U	e	u				・	オ	ナ	1	σ	Ü
xxxx0110	(7)		&	6	F	V	f	v				ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)		'	7	G	W	g	w				フ	キ	ヌ	ラ	g	π
xxxx1000	(1)		<	8	H	X	h	x				イ	ク	ネ	リ	J	Σ
xxxx1001	(2)		)	9	I	Y	i	y				ウ	ケ	ル	ル	”	y
xxxx1010	(3)		*	:	J	Z	j	z				エ	コ	ン	レ	j	〒
xxxx1011	(4)		+	;	K	L	k	l				オ	サ	ヒ	ロ	*	π
xxxx1100	(5)		,	<	L	¥	l	l				ハ	シ	フ	ワ	φ	π
xxxx1101	(6)		-	=	M	]	m	}				ユ	ズ	ヘ	ン	も	÷
xxxx1110	(7)		.	>	N	^	n	→				ヨ	セ	ホ	”	ñ	
xxxx1111	(8)		/	?	O	_	o	←				ツ	ソ	マ	”	ö	■

Note: The user can specify any pattern for character-generator RAM.

Figura 3. Juego de caracteres del módulo LCD

También permite definir 8 nuevos caracteres de usuario que se guardan en una zona de RAM denominada CGRAM (Character Generator RAM)

## 2.5. Modos de funcionamiento

El LCD tiene 3 modos de funcionamiento principales:

- **Modo Comando**
- **Modo Carácter o Dato**
- **Modo de lectura del Busy Flag o LCD Ocupada**

## 2.6. Comandos de Control

Consisten en diferentes códigos que se introducen a través del bus de datos del módulo LCD:

### Clear display

Borra el módulo LCD y coloca el cursor en la primera posición (dirección 0).  
Pone el bit I/D a " 1 " por defecto.

**Código:**

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

**Tiempo de ejecución:** 1.64  $\mu$ s

### Home

Coloca el cursor en la posición de inicio (dirección 0) y hace que el display comience a desplazarse desde la posición original. El contenido de la memoria RAM de datos de visualización (DD RAM) permanecen invariables. La dirección de la memoria RAM de datos para la visualización (DD RAM) es puesta a 0.

**Código:**

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	X

**Tiempo de ejecución:** 1.64  $\mu$ s

### Entry mode set

Establece la dirección de movimiento del cursor y especifica si la visualización se va desplazando a la siguiente posición de la pantalla o no. Estas

operaciones se ejecutan durante la lectura o escritura de la DD RAM o CG RAM. Para visualizar normalmente poner el bit S a '0'.

**Código:**

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	1/D	S

**Tiempo de ejecución:** 40  $\mu$ s

**Display on/off control**

Activa o desactiva poniendo en ON/OFF tanto al display (D) como al cursor (C) y se establece si este último debe o no parpadear (B).

**Código:**

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

**Tiempo de ejecución:** 40  $\mu$ s

**Cursor or display shift**

Mueve el cursor y desplaza el display sin cambiar el contenido de la memoria de datos de visualización DD RAM.

**Código:**

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	X	X

**Tiempo de ejecución:** 40  $\mu$ s

**Function set**

Establece el tamaño de interface con el bus de datos (DL), número de líneas del display (N) y tipo de carácter (F).

**Código:**

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	X	X

**Tiempo de ejecución:** 40  $\mu$ s

**Set the cg ram address**

El módulo LCD además de tener definidos todo el conjunto de caracteres ASCII, permite al usuario definir 4 u 8 caracteres gráficos. La composición de estos caracteres se va guardando en una memoria llamada CG RAM con capacidad para 64 bytes. Cada carácter gráfico definido por el usuario se compone de 16 u 8 bytes que se almacenan en sucesivas posiciones de la CG RAM.

Mediante esta instrucción se establece la dirección de la memoria CG RAM a partir de la cual se irán almacenando los bytes que definen un carácter gráfico. Ejecutado este comando todos los datos que se escriban o se lean posteriormente, lo hacen desde esta memoria CG RAM.

**Código:**

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	Dirección de la CG RAM					

**Tiempo de ejecución:** 40  $\mu$ s

**Set the dd address**

Los caracteres o datos que se van visualizando, se van almacenando previamente en una memoria llamada DD RAM para de aquí pasar a la pantalla.

Mediante esta instrucción se establece la dirección de memoria DD RAM a partir de la cual se irán almacenando los datos a visualizar. Ejecutado este comando, todos los datos que se escriban o lean posteriormente los hacen desde esta memoria DD RAM. Las direcciones de la 80h a la 8Fh corresponden con los 16 caracteres del primer renglón y de la C0h a la CFh con los 16 caracteres del segundo renglón, para este modelo.

**Código:**

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	Dirección de la DD RAM					

**Tiempo de ejecución:** 40  $\mu$ s

**Read busy flag et address**

Cuando el módulo LCD está ejecutando cualquiera de estas instrucciones, tarda un cierto tiempo de ejecución en el que no se le debe mandar ninguna

otra instrucción. Para ello dispone de un flag llamado BUSY (BF) que indica que se está ejecutando una instrucción previa.

Esta instrucción de lectura informa del estado de dicho flag además de proporcionar el valor del contador de direcciones de la CG RAM o de la DD RAM según la última que se haya empleado.

### Código:

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	BF	Dirección de la CG RAM o de la DD RAM						

**Tiempo de ejecución:** 40  $\mu$ s

### Write data to cg or dd ram

Mediante este comando se escribe en la memoria DD RAM los datos que se quieren presentar en pantalla y que serán los diferentes códigos ASCII de los caracteres a visualizar.

Igualmente se escribe en la memoria CG RAM los diferentes bytes que permiten confeccionar caracteres gráficos a gusto del usuario.

El escribir en uno u otro tipo de memoria depende de si se ha empleado previamente la instrucción de direccionamiento DD RAM o la de direccionamiento CG RAM.

### Código:

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	Código ASCII o byte del carácter gráfico							

**Tiempo de ejecución:** 40  $\mu$ s

### Read data from cg or dd ram

Mediante este comando se lee de la memoria DD RAM los datos que haya almacenados y que serán los códigos ASCII de los caracteres visualizados.

Igualmente se lee de la memoria CG RAM los diferentes bytes con los que se ha confeccionado un determinado carácter gráfico.



El leer de uno u otro tipo de memoria depende de si se ha empleado previamente la instrucción de direccionamiento de la DD RAM o la de direccionamiento CG RAM.

### Código:

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	Código ASCII o byte del carácter gráfico							

**Tiempo de ejecución:** 40  $\mu$ s

### Abreviaturas

Se listan a continuación las abreviaturas empleadas en los códigos anteriores y su significado:

S = 1	Desplaza la visualización cada vez que se escribe un dato
S = 0	Modo normal
I/D = 1	Incremento del cursor
I/D = 0	Decremento del cursor
S/C = 1	Desplaza el display
S/C = 0	Mueve el cursor
R/L = 1	Desplazamiento a la derecha
R/L = 0	Desplazamiento a la izquierda
BF = 1	Módulo ocupado
BF = 0	Módulo disponible
DL = 1	Bus de datos de 8 bits
DL = 0	Bus de datos de 4 bits
N = 1	LCD de 2 líneas
N = 0	LCD de 1 línea
F = 1	Carácter de 5x10 puntos
F = 0	Carácter de 5x7 puntos
B = 1	Parapdeo de cursor ON
C = 1	Cursor ON
D = 1	Display ON
X =	Indeterminado

## 2.7. Secuencia de inicialización

El módulo LCD ejecuta automáticamente una secuencia de inicio interna en el instante de aplicarle la tensión de alimentación si se cumplen los requisitos de alimentación expuestos en su manual.

Dichos requisitos consisten en que el tiempo que tarde en estabilizarse la tensión desde 0.2 V hasta los 4.5V mínimos necesario sea entre 0.1 ms y 10 ms. Igualmente el tiempo de desconexión debe ser como mínimo de 1 ms antes de volver a conectar.

La secuencia de inicio ejecutada es la siguiente:

- 1.- CLEAR DISPLAY.  
El flag Busy se mantiene a "1" (ocupado) durante 15 mS hasta que finaliza la inicialización.
- 2.- FUNCTION SET  
Se elige por defecto el tamaño del bus de datos a 8 bits (DL=1) y el número de renglones del display en 1 (N=0).
- 3.- DISPLAY ON/OFF CONTROL  
Se elige por defecto display en OFF (D=0), cursor en OFF (C=0) y parpadeo del cursor en OFF (B=0)
- 4.- ENTRY MODE SET  
Se elige por defecto incremento del cursor (I/D=1) y modo normal, no desplazamiento, del display (S=0).
- 5.- Se selecciona la primera posición de la DD RAM.

Si no se satisfacen las condiciones de alimentación, la secuencia de inicialización habría que realizarla por software, donde las instrucciones que aplica el usuario podrían ser las expuestas anteriormente o cualquier otra según sus propias necesidades.

Es importante que la primera instrucción que se envíe realice una espera de unos 15 ms o mayor para la completa reinicialización interna del módulo LCD.

## 2.8. Conexión de LCD mediante 4 bits

En la tarjeta EasyPIC, el visualizador LCD está conectado al Puerto D del PIC 16F877A mediante un bus de 4 bits.

Las conexiones son:

- Las 4 líneas superiores del módulo LCD, pines <DB7:DB4> se conectan a las 4 líneas superiores del Puerto D del PIC, pines <RD7:RD4>.
- Pin RS del LCD a la línea RD2 del PIC.
- Pin R/W del LCD a la línea RD1 del PIC
- Pin Enable del LCD a la línea RD3 del PIC.

La figura 4 muestra un esquema de la conexión del módulo LCD con el microcontrolador en la EasyPIC.

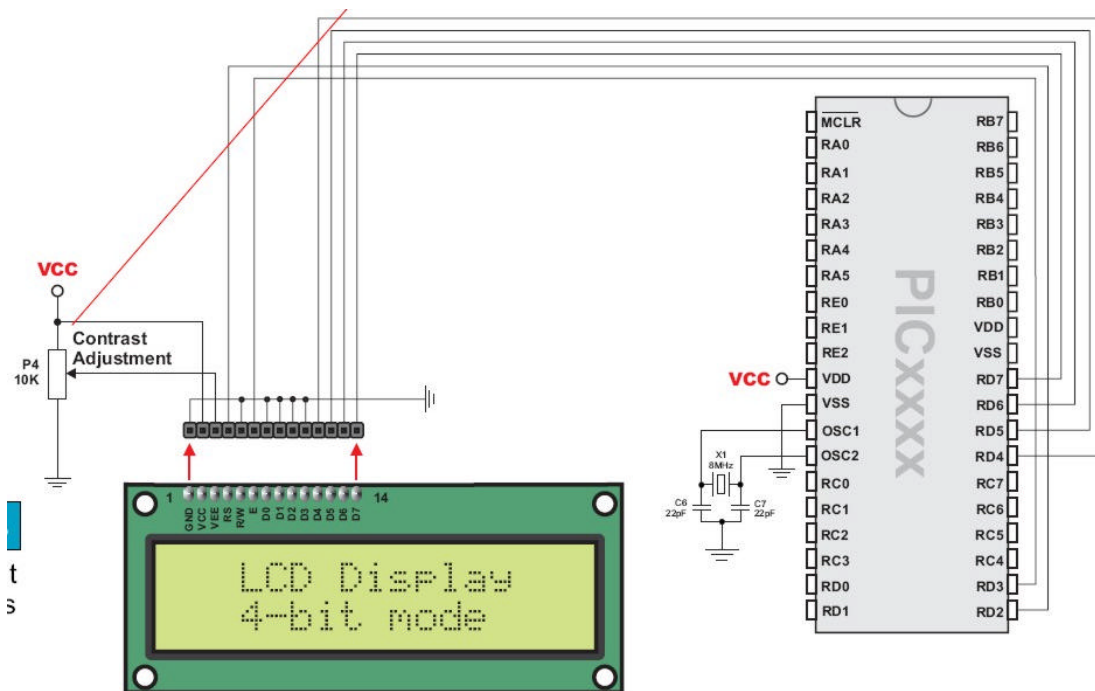


Figura 4. Conexión del módulo LCD con el microcontrolador en la EasyPIC

### 3. LIBRERÍA DE SUBRUTINAS

La librería LCD\_EASY.INC contiene las subrutinas de control que permiten realizar las tareas básicas de control de un módulo LCD conectado según el circuito de la figura 4.

La librería LCD\_EASY.INC es una versión de la librería LCD\_4BIT.INC (descrita en el libro) modificada según la conexión del módulo LCD con el microcontrolador PIC16F877A en la EasyPIC (figura 4).

Aunque esta librería está suficientemente documentada, se destacan algunas de sus subrutinas principales:

#### *LCD\_Inicializa*

*Inicialización del módulo LCD: Configura funciones del LCD, produce reset por software, borra memoria y enciende pantalla. Es necesario ejecutar esta subrutina siempre al principio de los programas que vayan a utilizar la visualización mediante LCD.*

#### *LCD\_EnviaComando.*

*Escribe un comando en el registro del módulo LCD. La palabra de comando ha sido entregada a través del registro W. Trabaja en Modo Comando.*

#### *LCD\_Character.*

*Escribe en la memoria DDRAM del LCD el carácter ASCII introducido a a través del registro W. Trabaja en Modo Dato.*

#### *LCD\_Borra*

*Borra toda la pantalla*

#### *LCD\_Linea1*

*Cursor al principio de la Línea 1.*

#### *LCD\_Linea2*

*Cursor al principio de la Línea 2.*

#### *LCD\_PosicionLinea1*

*Cursor a posición de la Línea 1*

#### *LCD\_PosicionLinea2*

*Cursor a posición de la Línea 2*

#### *LCD\_OFF*

*Pantalla apagada.*

*LCD\_CursorON*  
*Pantalla encendida y cursor*

*LCD\_CursorOFF*  
*Pantalla encendida y cursor apagado.*

La librería LCD\_EASY.INC hace uso de las subrutinas de retardo incluidas en la librería RETARDOS.INC.

Recordad que las subrutinas de retardo de la librería RETARDOS.INC están diseñadas suponiendo que la frecuencia del oscilador es de 4 MHz.

Como en la tarjeta EasyPIC, el reloj es de 8 MHz, habrá que duplicar las llamadas a las subrutinas de temporización para lograr los tiempos deseados. Así, por ejemplo, si se desea una espera de 500ms, el código a utilizar será:

```
call Retardo_500ms  
call Retardo_500ms
```

## 4. PROGRAMAS EJEMPLO y EJERCICIOS

### EJEMPLO 1:

- a) El programa **Lcd\_Test.HEX**, para el PIC16F877A, produce que en la pantalla del módulo LCD se visualice un mensaje.
- b) Comprueba el funcionamiento del programa en la tarjeta EasyPIC.

### EJEMPLO 2:

- a) El programa **LCD\_01.asm**, para el PIC16F877A, produce que en la pantalla del módulo LCD se visualiza el mensaje "Hola". Al terminar de escribir la frase el PIC entrará en modo de bajo consumo.
- b) Comprueba el funcionamiento del programa en la tarjeta EasyPIC.
- c) Modifica el programa de modo que en la pantalla del módulo LCD se visualice tu nombre.

**EJEMPLO 3:**

- a) El programa **LCD\_02.asm**, para el PIC16F877A, produce que en la pantalla del módulo LCD se visualiza el mensaje "Hola". La escritura de cada carácter se irá realizando cada 500ms. Después se borrará y comenzará de nuevo.
- b) Comprueba el funcionamiento del programa en la tarjeta EasyPIC.
- c) Modifica el programa de modo que en la pantalla del módulo LCD se visualice tu nombre.

**EJEMPLO 4:**

- a) El programa **LCD\_04.asm**, para el PIC16F877A, produce que en la pantalla del módulo LCD se visualiza un contador descendente de 59 hasta 0 y vuelve a repetir la cuenta ininterrumpidamente. En cada valor estará unos 500 ms.
- b) Comprueba el funcionamiento del programa en la tarjeta EasyPIC.
- c) Modifica el programa de modo que el contador cuente desde 30 hasta 0.
- d) Modifica el programa de modo que el contador cuente en modo ascendente desde 0 hasta 30.

**EJEMPLO 5:**

- a) En el programa **LCD\_05.asm**, cada vez que se presiona el pulsador conectado al pin RA4 se incrementa un contador que se visualiza en el centro de la primera línea de la pantalla.
- b) Comprueba el funcionamiento del programa en la tarjeta EasyPIC.

**EJEMPLO 6:**

- a) Escribe un programa que realice la siguiente tarea: cada vez que se presione el pulsador conectado al pin **RA4** se visualice el mensaje "**HOLA**" en el centro de la primera línea de la pantalla, y cuando se presione el pin **RA5** se visualice el mensaje "**ADIOS**", también en el centro de la primera línea de la pantalla.
- b) Comprueba el funcionamiento del programa en la tarjeta EasyPIC.