

## **Práctica 6**

### **Gestión de operaciones con LCD**

- ❑ Introducción: Objetivos
- ❑ Aproximación a LCDs compatibles con el PIC16F877A
  - Características básicas de la familia HD44780
  - Características del dispositivo WH1602B
- ❑ Modelado del LCD en ISIS Proteus
- ❑ Programación en C de CCS de E/S con el LCD
- ❑ Tareas a realizar

# Introducción

## ❑ Objetivos

- Conocer las características de los dispositivos LCD basados en la MCU HD44780.
- Ser capaz de modelar en ISIS Proteus un LCD basado en HD44780 y conectado con el PIC16F877A.
- Ser capaz de programar en C de CCS operaciones entre la MCU y el LCD modelado.



## Aproximación a LCDs compatibles con el PIC16F877A

- ❑ A la hora de conectar el PIC16F877A a un LCD, suelen emplearse dispositivos basados en la MCU de propósito específico HD44780
- ❑ El HD44780 fue desarrollado por Hitachi específicamente para gestionar displays LCD de matriz de puntos, ofreciendo una interfaz a otra MCU de propósito más generalista que controla realmente las operaciones.
- ❑ Existen varios dispositivos LCD, de distintos fabricantes, basados en el HD44780: usaremos el WH1602B de Winstar, que puede conectarse a la placa UNI-DS3 y a través de ella con el PIC16F877A.

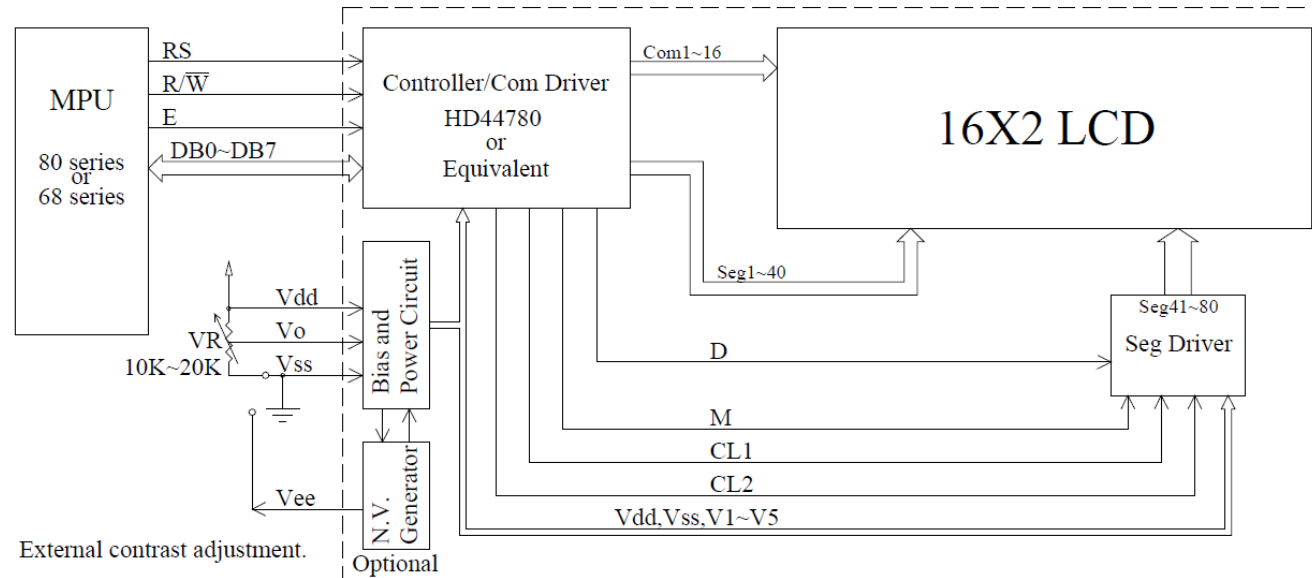


Diagrama de bloques básico del WH1602B (línea discontinua)

## Características básicas de la familia HD44780

- ❑ Los dispositivos basados en el HD44780 se gestionan de forma similar al ser éste el componente principal de todos ellos.
- ❑ Los componentes principales del HD44780 son:
  - Interfaz con otra MCU: líneas Register Select, R/W, Enable, y 8 líneas de datos
  - Interfaz (drivers) con la pantalla LCD
  - RAM de datos de la pantalla
  - ROM y RAM generadoras de caracteres
  - Registros (instrucciones, datos, etc.)

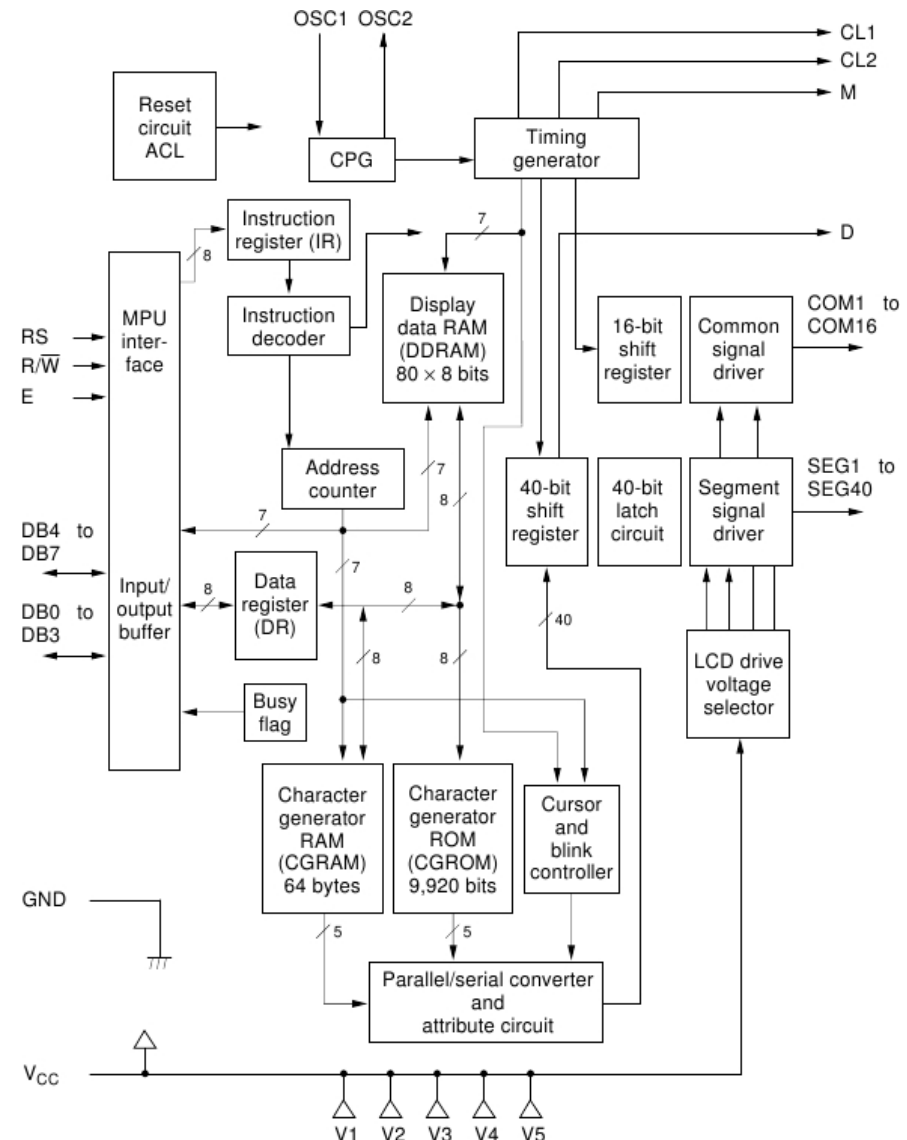


Diagrama de bloques interno del HD44780

## Características básicas de la familia HD44780

- Dependiendo del valor de las líneas RS y R/W se habilitan las siguientes operaciones entre la MCU principal y los registros de instrucciones y datos del HD44780:

RS	R/W	Operation
0	0	IR write as an internal operation (display clear, etc.)
0	1	Read busy flag (DB7) and address counter (DB0 to DB6)
1	0	DR write as an internal operation (DR to DDRAM or CGRAM)
1	1	DR read as an internal operation (DDRAM or CGRAM to DR)

- Las líneas de datos pueden usarse en dos modos: 8 y 4 bits.
  - Si se opera en 8 bits, se utilizan todas las líneas de datos (0 a 7).
  - Si se opera en 4 bits sólo se usan las líneas de datos 4 a 7, y se envían o reciben datos o instrucciones mediante dos transmisiones, cada una de un nibble (4 bits) distinto.

# Características básicas de la familia HD44780

## ❑ Juego de instrucciones del HD44780:

HD44780U based instruction set

Instruction	Code										Description	Execution time (max) (when $f_{cp} = 270 \text{ kHz}$ )
	RS	R/W	B7	B6	B5	B4	B3	B2	B1	B0		
Clear display	0	0	0	0	0	0	0	0	0	1	Clears display and returns cursor to the home position (address 0).	1.52 ms
Cursor home	0	0	0	0	0	0	0	0	1	*	Returns cursor to home position. Also returns display being shifted to the original position. DDRAM content remains unchanged.	1.52 ms
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction (I/D); specifies to shift the display (S). These operations are performed during data read/write.	37 $\mu\text{s}$
Display on/off control	0	0	0	0	0	0	1	D	C	B	Sets on/off of all display (D), cursor on/off (C), and blink of cursor position character (B).	37 $\mu\text{s}$
Cursor/display shift	0	0	0	0	0	1	S/C	R/L	*	*	Sets cursor-move or display-shift (S/C), shift direction (R/L). DDRAM content remains unchanged.	37 $\mu\text{s}$
Function set	0	0	0	0	1	DL	N	F	*	*	Sets interface data length (DL), number of display line (N), and character font (F).	37 $\mu\text{s}$
Set CGRAM address	0	0	0	1	CGRAM address						Sets the CGRAM address. CGRAM data are sent and received after this setting.	37 $\mu\text{s}$
Set DDRAM address	0	0	1	DDRAM address							Sets the DDRAM address. DDRAM data are sent and received after this setting.	37 $\mu\text{s}$
Read busy flag & address counter	0	1	BF	CGRAM/DDRAM address							Reads busy flag (BF) indicating internal operation being performed and reads CGRAM or DDRAM address counter contents (depending on previous instruction).	0 $\mu\text{s}$
Write CGRAM or DDRAM	1	0	Write Data								Write data to CGRAM or DDRAM.	37 $\mu\text{s}$
Read from CG/DDRAM	1	1	Read Data								Read data from CGRAM or DDRAM.	37 $\mu\text{s}$

### Instruction bit names —

I/D - 0 = decrement cursor position, 1 = increment cursor position; S - 0 = no display shift, 1 = display shift; D - 0 = display off, 1 = display on; C - 0 = cursor off, 1 = cursor on; B - 0 = cursor blink off, 1 = cursor blink on; S/C - 0 = move cursor, 1 = shift display; R/L - 0 = shift left, 1 = shift right; DL - 0 = 4-bit interface, 1 = 8-bit interface; N - 0 = 1/8 or 1/11 duty (1 line), 1 = 1/16 duty (2 lines); F - 0 = 5×8 dots, 1 = 5×10 dots; BF - 0 = can accept instruction, 1 = internal operation in progress.

## Características básicas de la familia HD44780

- ❑ Los caracteres en la RAM generadora puede configurarlos el usuario.
- ❑ El juego de caracteres de la ROM generadora viene establecido “de serie”, pero puede variar entre modelos de dispositivo:

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	0	P	`	P					-	9	3	α	p
xxxx0001	(2)			!	1	A	Q	a	q				•	7	4	ä	q
xxxx0010	(3)			"	2	B	R	b	r				「	イ	ツ	×	θ
xxxx0011	(4)			#	3	C	S	c	s				」	ウ	テ	ε	∞
xxxx0100	(5)			\$	4	D	T	d	t				、	エ	ト	μ	Ω
xxxx0101	(6)			%	5	E	U	e	u				•	オ	ナ	1	℃
xxxx0110	(7)			&	6	F	V	f	v				ヲ	カ	ニ	ヨ	ρ
xxxx0111	(8)			'	7	G	W	g	w				フ	キ	ヌ	ラ	π
xxxx1000	(1)			(	8	H	X	h	x				イ	ク	ネ	リ	Σ
xxxx1001	(2)			)	9	I	Y	i	y				ウ	ケ	ル	リ	γ
xxxx1010	(3)			*	:	J	Z	j	z				エ	コ	ン	レ	χ
xxxx1011	(4)			+	;	K	[	k	[				オ	サ	ヒ	ロ	°
xxxx1100	(5)			,	<	L	\	l	\				カ	シ	フ	ワ	⊕
xxxx1101	(6)			-	=	M	]m	]	m				ユ	ズ	ヘ	ン	÷
xxxx1110	(7)			.	>	N	^n	^	n				ヨ	セ	ホ	°	ñ
xxxx1111	(8)			/	?0	_o	o	+	+				ッ	ソ	マ	°	■

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	0	P	`	P	Б	α		°	À	0	à	3	
xxxx0001	(2)			!	1	A	Q	a	q	А	Ј	і	±	А	Ñ	á	ñ
xxxx0010	(3)			"	2	B	R	b	r	В	Г	ѳ	²	À	ò	â	ò
xxxx0011	(4)			#	3	C	S	c	s	З	π	ℓ	³	À	ó	ã	ó
xxxx0100	(5)			\$	4	D	T	d	t	Д	Σ	×	℔	À	ô	ä	ô
xxxx0101	(6)			%	5	E	U	e	u	Ѓ	σ	¥	℥	À	õ	å	õ
xxxx0110	(7)			&	6	F	V	f	v	Ј	Ј	і	q	Æ	Ö	ø	ö
xxxx0111	(8)			'	7	G	W	g	w	П	τ	§	•	Ç	×	÷	
xxxx1000	(1)			(	8	H	X	h	x	У	•	ф	ω	È	†	è	†
xxxx1001	(2)			)	9	I	Y	i	y	Ч	Θ	Θ	¹	É	Ù	é	Ù
xxxx1010	(3)			*	:	J	Z	j	z	Ч	Ω	Ω	²	É	Ú	ê	Ú
xxxx1011	(4)			+	;	K	[	k	[	Ш	δ	«	»	É	Û	ë	Û
xxxx1100	(5)			,	<	L	\	l	\	Щ	≈	℥	¥	İ	Ü	ı	Ü
xxxx1101	(6)			-	=	M	]m	]	m	Ъ	•	Я	℥	İ	Ý	ı	Ý
xxxx1110	(7)			.	>	N	^n	^	n	Ѧ	ε	Q	¼	İ	İ	ı	İ
xxxx1111	(8)			/	?0	_o	o	+	+	Ѧ	Q	Q	Q	İ	İ	ı	İ

ROM CODE A02



## Características básicas de la familia HD44780

- ❑ Las pantallas LCD de matriz de puntos que se usan en dispositivos basados en el HD44780 pueden ser de distinto tamaño: 8X1 (una fila de 8 caracteres), 16x2 (dos filas de 16 caracteres), 20x2, 20x4, etc.



Pantalla 8x1



Pantalla 20x4



Pantalla 16x2



## Características del dispositivo WH1602B

- ❑ El dispositivo WH1602B de Winstar que usaremos en las prácticas con la placa UNI-DS3 ofrece una pantalla 16x2.
- ❑ Este dispositivo tiene 16 pines, organizados del siguiente modo:

Pin No.	Symbol	Level	Description
1	V <sub>SS</sub>	0V	Ground
2	V <sub>DD</sub>	5.0V	Supply Voltage for logic
3	VO	(Variable)	Operating voltage for LCD
4	RS	H/L	H: DATA, L: Instruction code
5	R/W	H/L	H: Read(MPU→ Module) L: Write(MPU→ Module)
6	E	H,H→ L	Chip enable signal
7	DB0	H/L	Data bit 0
8	DB1	H/L	Data bit 1
9	DB2	H/L	Data bit 2
10	DB3	H/L	Data bit 3
11	DB4	H/L	Data bit 4
12	DB5	H/L	Data bit 5
13	DB6	H/L	Data bit 6
14	DB7	H/L	Data bit 7
15	A	-	LED +
16	K	-	LED -

## Modelado del LCD para la placa UNI-DS3 en ISIS Proteus

- ❑ La placa UNI-DS3 tiene un conector de 16 pines para un dispositivo LCD similar al WH1602B. En ISIS Proteus este dispositivo puede modelarse mediante el componente LM016L (Categoría: “Optoelectronics”, Subcategoría: “Alphanumeric LCDs”).
- ❑ Se asume que el dispositivo trabaja en modo 4 bits, lo que condiciona las conexiones entre el dispositivo y la MCU principal: los pines DB0 a DB3 del dispositivo LCD no se conectan a la MCU sino a masa.
- ❑ En la placa UNI-DS3, la línea R/W del LCD también va a masa.
- ❑ Modelaremos el dispositivo LCD conectado al puerto D con arreglo al siguiente esquema:

LCD	PIC16F877A
RS	RD0
Enable	RD1
DB4	RD4
DB5	RD5
DB6	RD6
DB7	RD7

## Programación en C de CCS de E/S con el LCD

- ❑ La librería estándar para gestionar un LCD con el C de CCS es “LCD.C”, que se proporciona con el compilador.
- ❑ Sin embargo, esta librería no es totalmente compatible con el conector de 16 pines para LCD de la placa UNI-DS3, por lo que usaremos la librería modificada “LCD\_digi2.c”, que sí es compatible.
- ❑ Esta librería debe incluirse desde el programa principal en C mediante una directiva `#INCLUDE`.
- ❑ La librería está configurada asumiendo que el LCD se conecta al puerto D de la MCU. Si se desea trabajar con otro puerto, hay que modificarla, comentando o descomentando, o reescribiendo, directivas del tipo:

```
// #define use_portb_lcd TRUE
```

## Funciones de la librería para la gestión del LCD

- ❑ **LCD\_INIT () ;** . Inicializa el dispositivo LCD. Debe usarse antes de cualquier otra operación.
  
- ❑ **LCD\_GOTOXY (BYTE x, BYTE y) ;** . Sitúa el cursor del LCD en el carácter **x** de la fila **y**. El sentido es de izquierda a derecha y de arriba abajo, es decir, la posición (1,1) corresponde al carácter más a la izquierda, en la fila superior. El cursor se desplaza automáticamente al escribir caracteres (este desplazamiento es configurable), por lo que por defecto no es necesario invocar a **LCD\_GOTOXY** tras cada escritura.
  
- ❑ **LCD\_PUTC (CHAR C) ;** . El carácter indicado como parámetro se escribe en la posición del cursor. El LCD responde de forma especial a la escritura de los siguientes caracteres:
  - **\f** : limpia el LCD
  - **\n** : pone el cursor en la posición (1,2)
  - **\b** : el cursor retrocede una posición

## Funciones de la librería para la gestión del LCD

- ❑ **PRINTF(LCD\_PUTC, CSTRING, values) ;** . Escribe en el LCD la cadena indicada por los parámetros. **CSTRING** es una cadena con el formato típico de la instrucción **PRINTF**, donde pueden intercalarse valores de variables indicadas en la lista **values**. Ejemplo:

```
PRINTF(LCD_PUTC, "%d y %d suman %d", var1, var2, var1+var2);
```

- ❑ **LCD\_SEND\_NIBBLE (BYTE X) ;** . El nibble más bajo de **x** se envía por las líneas de datos (modo 4 bits).
- ❑ **LCD\_SEND\_BYTE (BYTE RS, BYTE X) ;** . Los dos nibbles que forman el valor **x** se envían en dos transmisiones consecutivas por las líneas de datos (modo 4 bits). La línea RS toma el valor (0 o 1) indicado en el primer parámetro, por lo que la función puede usarse para enviar al dispositivo LCD bien instrucciones, bien datos a la pantalla.

## Ejemplo

```
#include <16f877a.h>
#fuses HS,NOWDT, NOLVP
#use delay(clock= 8000khz)
#include "lcd_digi2.c"

void main()
{
    int x=0;
    LCD_INIT();                // Inicializa LCD
    LCD_GOTOXY(1,1);           // Pone cursor en esquina superior izquierda
    while (x<16){
        delay_ms(500);
        printf(lcd_putc,"%1x",x);    // Imprime uno a uno digitos hexa de 0 a f
        x++;
    }
    delay_ms(500);
    LCD_PUTC('\f');             // Limpia la pantalla
    while(1);
}
```

## Tareas a realizar

1. Modelar en Isis Proteus la parte del circuito de la placa UNI DS3 correspondiente al LCD, según lo explicado en las transparencias anteriores.
2. Probar en el entorno MPLAB, con el modelo de LCD ya incorporado en el diseño de la placa, el ejemplo de la transparencia 14.
3. Desarrollar un programa en C de CCS que escriba en la línea superior del LCD “INGENIERIA” y en la línea de abajo “INFORMATICA”, teniendo en cuenta que los caracteres deben aparecer de uno en uno y de izquierda a derecha, con un retardo de medio segundo, en la fila superior, y de uno en uno y de derecha a izquierda con un retardo de medio segundo, en la fila inferior. El texto debe aparecer lo más centrado posible.
4. Ampliar el programa anterior para que, antes del escribir el texto indicado, se escriba “ESCUELA” en la línea superior (de una vez), y “SUPERIOR DE” (de una vez) en la línea inferior. Tras un retardo de 3 segundos, el texto desaparecerá durante 1 segundo y volverá a aparecer durante 3 segundos, y así hasta 5 veces para conseguir un efecto de “parpadeo”. Terminado el parpadeo, el programa esperará a que se pulse el interruptor conectado a RB0, y entonces se limpiará la pantalla y comenzará a aparecer el texto de la tarea anterior.

Deben entregarse el diseño de la placa realizado y los programas desarrollados para las tareas 3 y 4.