

Capítulo 1

Práctica: Display alfanumérico”

*Vale más saber alguna cosa de todo,
que saberlo todo de una sola cosa
~ Blaise Pascal ~*

1.1. Objetivo

¹ Esta, que podría parecer la primera práctica de todas requiere la actuación sobre un hardware algo más complicado que lo visto hasta el momento: un display alfanumérico de 16x2 caracteres, que es necesario estudiar.

Se trata de aprender a actuar sobre un hardware más complejo (displays LCD alfanuméricos compatibles Hitachi 44780) y que nos permitirá la realización de diseños más complejos y espectaculares.

1.2. Especificaciones

Utilizando la información proporcionada se pide que conectemos y actuemos sobre un display alfanumérico. Para ello tendremos que controlar bien el protocolo de comunicaciones con el mismo y la temporización necesaria para dicha comunicación.

Una vez que seamos capaces de hacer lo anterior realizaremos el típico programa **Hola Mundo** que consiste en visualizar en la pantalla el conocido mensaje.

1.3. Para sacar nota

Usa todas las prestaciones del display para hacer que aparezcan mensajes en la segunda línea del display y diseñar efectos de aparición de texto con scroll.

Existe la posibilidad de conexión con un bus de datos de 4 bits en vez de usar un bus de datos de 8 bits. Implementa las rutinas y el hardware necesario.

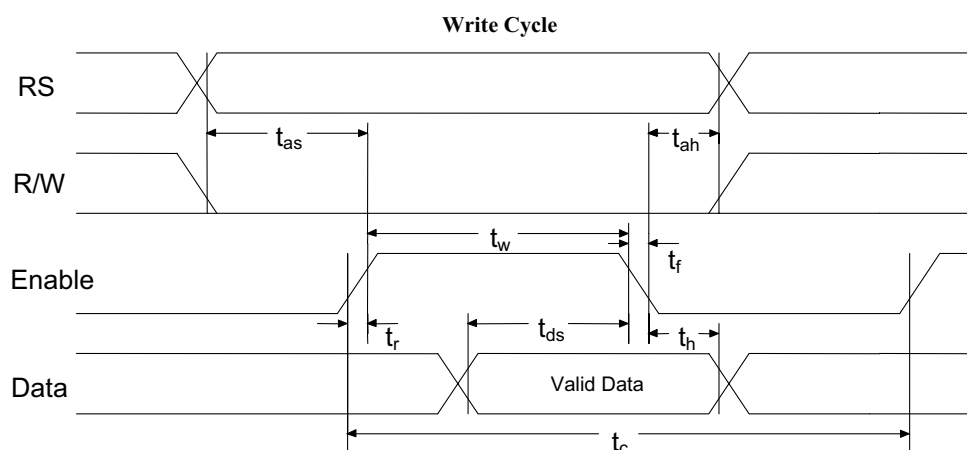
Nota. Los tiempos de espera necesarios para el correcto funcionamiento del display dependen del fabricante (el reloj interno del display suele funcionar a 240 kHz). Para una escritura normal el tiempo necesario es de 40 microsegundos como máximo. Para la iniciación del LCD o para borrar la pantalla, puede ser de hasta 1.62 ms.

¹Versión de 21 de abril de 2008

The Concise LCD Data Sheet.

<http://www.senet.com.au/~cpeacock>

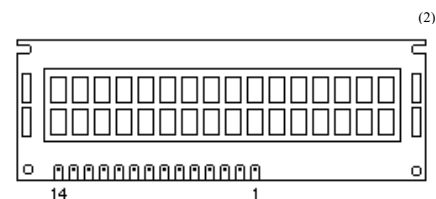
Instruction	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	Description	Clocks
NOP	0	0	0	0	0	0	0	0	0	0	No Operation	0
Clear Display	0	0	0	0	0	0	0	0	0	1	Clears display & sets address counter to zero.	165
Cursor Home	0	0	0	0	0	0	0	0	1	0	Sets address counter to zero, returns shifted display to original position. DDRAM contents remains unchanged.	3
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction, and specifies automatic shift.	3
Display Control	0	0	0	0	0	0	1	D	C	B	Turns display (D), cursor on/off (C) or cursor blinking(B).	3
Cursor/display shift	0	0	0	0	0	1	S/C	R/L	0	0	Moves cursor and shift display. DDRAM contents remains unchanged.	3
Function Set	0	0	0	0	1	DL	N	M	G	0	Sets interface data width(DL), number of display lines (N,M) and voltage generator control (G).	3
Set CGRAM Addr	0	0	0	1	Character Generator RAM						Sets CGRAM Address	3
Set DDRAM Addr	0	0	1	Display Data RAM Address						Sets DDRAM Address	3	
Busy Flag & Addr	0	1	BF	Address Counter						Reads Busy Flag & Address Counter		0
Read Data	1	1	Read Data						Reads data from CGRAM or DDRAM			3
Write Data	1	0	Write Data						Writes data from CGRAM or DDRAM			3



Parameter	Symbol	Min ⁽¹⁾	Typ ⁽¹⁾	Max ⁽¹⁾	Unit
Enable Cycle Time	t_c	500	-	-	ns
Enable Pulse Width (High)	t_w	230	-	-	ns
Enable Rise/Fall Time	t_r, t_f	-	-	20	ns
Address Setup Time	t_{as}	40	-	-	ns
Address Hold Time	t_{ah}	10	-	-	ns
Data Setup Time	t_{ds}	80	-	-	ns
Data Hold Time	t_h	10	-	-	ns

Note ¹ The above specifications are a indication only. Timing will vary from manufacturer to manufacturer.

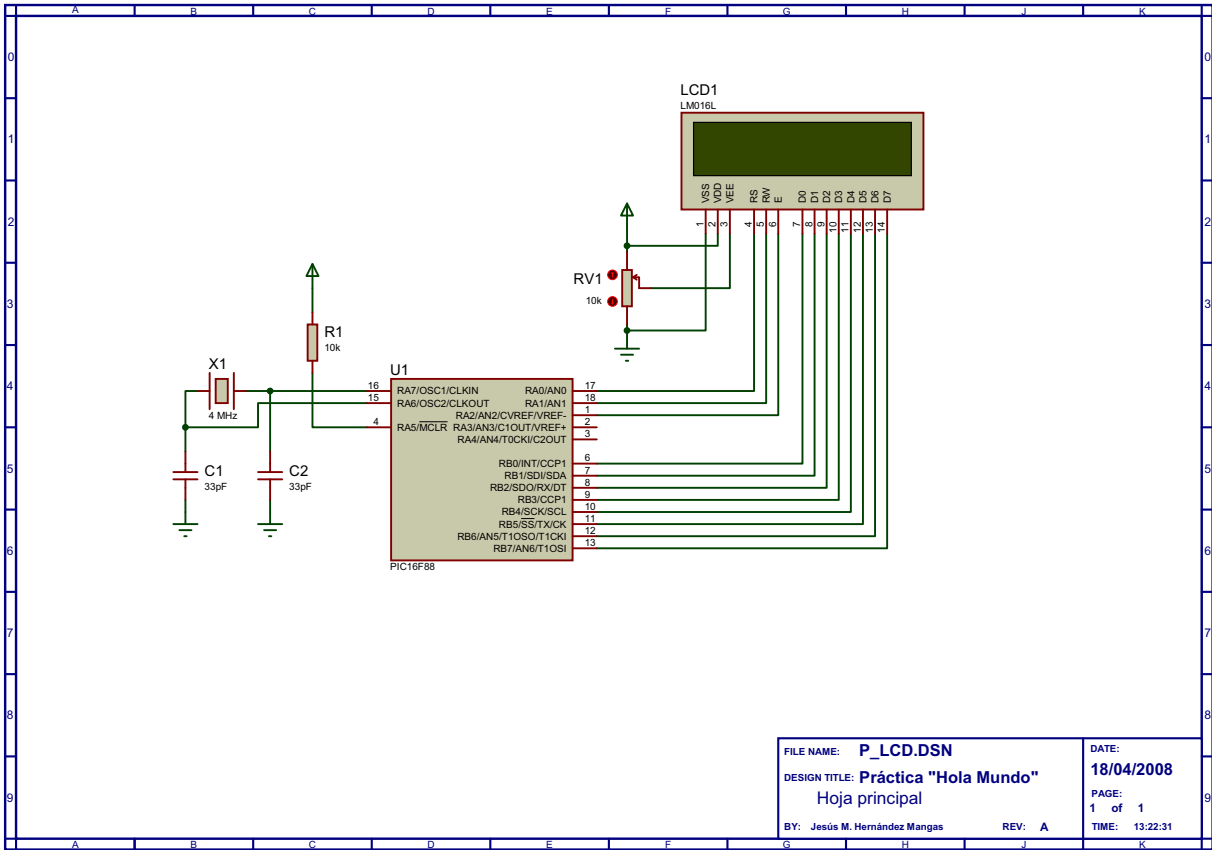
Note ² A 2 line by 16 Character LCD Module is Pictured. Data will work on most 1 line x 16 character, 1 line x 20 character, 2 line x 16 character, 2 line x 20 character, 4 lines x 20 character, 2 lines x 40 character etc. modules compatible with the HD44780 LCD Module.



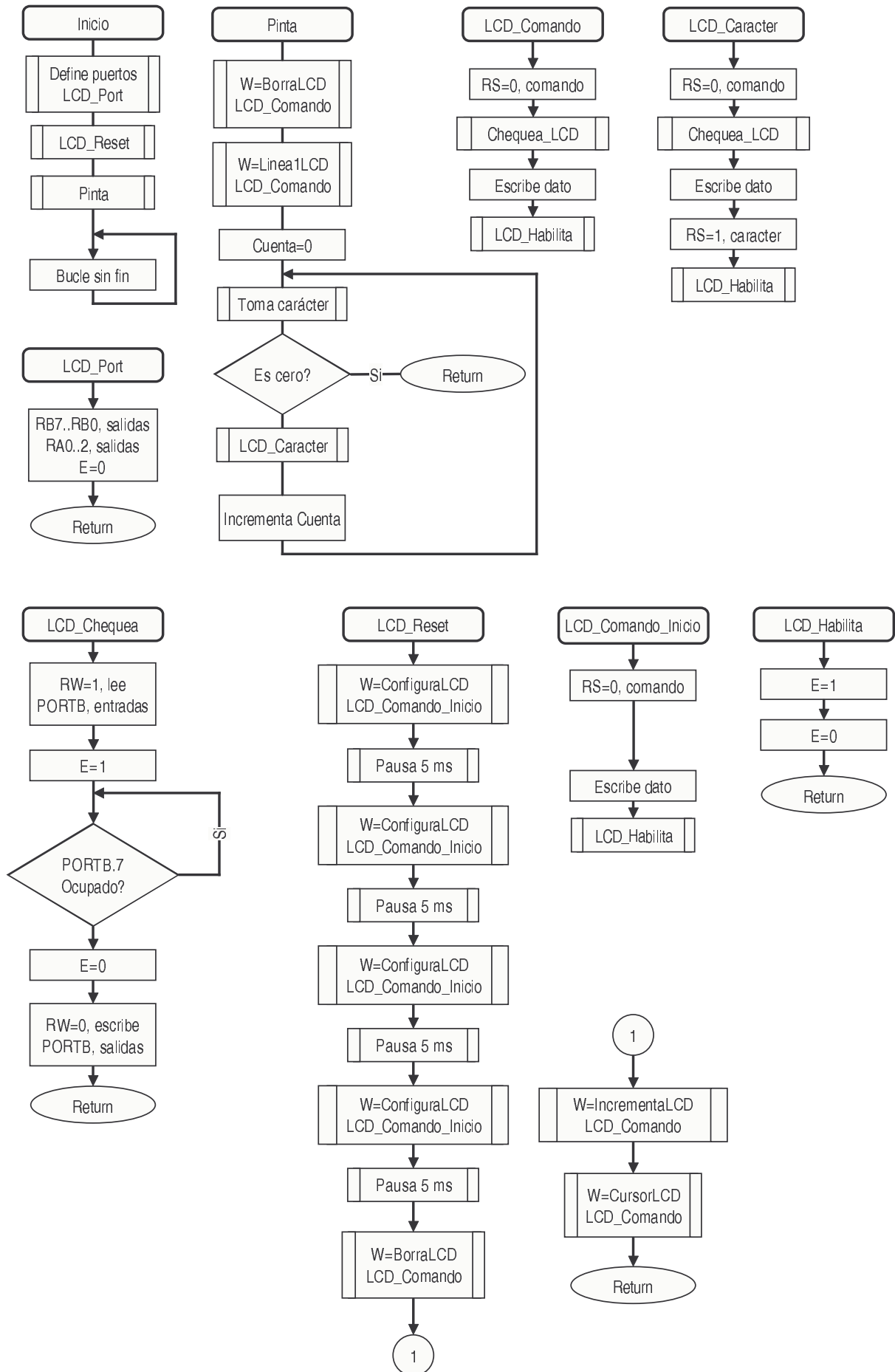
Pin No	Name	I/O	Description
1	Vss	Power	GND
2	Vdd	Power	+5v
3	Vo	Analog	Contrast Control
4	RS	Input	Register Select
5	R/W	Input	Read/Write
6	E	Input	Enable (Strobe)
7	D0	I/O	Data LSB
8	D1	I/O	Data
9	D2	I/O	Data
10	D3	I/O	Data
11	D4	I/O	Data
12	D5	I/O	Data
13	D6	I/O	Data
14	D7	I/O	Data MSB

1.4. Solución

1.4.1. Esquema hardware



1.4.2. Organigrama



1.4.3. Programas

```

1  #include <pic.h>
2  #include "LCD.h"
3
4  // Para el PIC16F88
5  __CONFIG( WDTDIS & XT & UNPROTECT & PWRTDIS & CCPRBO & DEBUGDIS & LVPDIS
6           & BORDIS & MCLREN ); // Palabra de configuración 0x2007
7  __CONFIG( FCMDIS & IESODIS ); // Palabra de configuración 0x2008
8
9  const unsigned char T[]="Hola_Mundo";
10
11 void main()
12 {
13     unsigned char i=0;
14
15     ANSEL = 0;
16     LCD_Port();    // Inicia los puertos
17     LCD_Reset();   // Inicia el LCD
18
19     while( T[i]!=0 ) LCD_Caracter( T[i++] );    // Pinta
20     while(1);    // Bucle infinito
21 }

```

```

1  // Fichero LCD.h
2  // Se debe incluir en el fichero en que se vaya a usar, una sola vez
3  #include <pic.h>
4  #include "binario.h"
5  // Definimos las conexiones
6  #define LCD_Data      PORTB    /* Puerto de datos */
7  #define LCD_RS        RA0      /* Comando o dato */
8  #define LCD_RW        RA1      /* Lectura/Escritura LCD */
9  #define LCD_E         RA2      /* Habilita LCD */
10
11 // COMANDOS para el LCD HD44780
12 #define LCDClear       B00000001 /* Borra y cursor al principio */
13 #define LCDCasa        B00000010 /* Cursor al principio */
14 #define LCDInc         B00000110 /* Selecciona incrementos */
15 /*      **      Bit 1 I/D (=0 dec posición) */
16 /*                      (=1 inc posición)* */
17 /*      Bit 0 S      (=0 no displ. disp.)* */
18 /*                      (=1 desplaza disp.)* */
19 #define LCDDec         B00000100 /* Sel. decrementos, no displ. */
20 #define LCDOn          B00001100 /* Control del display */
21 /*      ***      Bit 2 D      (=0 display apagado) */
22 /*                      (=1 disp.encendido)* */
23 /*      Bit 1 C      (=0 sin cursor)* */
24 /*                      (=1 con cursor) */
25 /*      Bit 0 B      (=0 sin parpadeo)* */
26 /*                      (=1 con parpadeo) */
27 #define LCDOff         B00001000 /* Disp. off, sin cursor, no parp. */
28 #define CursOn         B00001110 /* Disp. on, con cursor, no parp. */
29 #define CursOff        B00001100 /* Disp. on, sin cursor, no parp. */
30 #define CursBlink      B00001111 /* Disp. on, con cursor, con parp. */
31 #define LCDIzquierda   B00011000 /* Desplaz. de cursor o disp. */
32 /*      ***--      Bit 4 S/C (=0) Mueve el cursor */
33 /*                      (=1) Despl. el disp.* */
34 /*      Bit 3 R/L (=0) A la izquierda* */
35 /*                      (=1) A la derecha */
36 #define LCDDerecha     B00011100 /* Desplaza a la derecha el disp. */
37 #define CursIzquierda  B00010000 /* Cursor a la izquierda */
38 #define CursDerecha    B00010100 /* Cursor a la derecha */
39 #define LCDFuncion     B00111000 /* Bit 4 DL (=0 4 bit bus) */
40 /*      ***--      (=1 8 bit bus)* */
41 /*      Bit 3 N      (=0 disp. de 1 linea) */
42 /*                      (=1 disp. de 2 líneas)* */
43 /*      Bit 2 F      (=0 font 5x8)* */

```

```

44                                     /*                      (=1 font 5x10)                      */
45 #define LCDCGRAM      B01000000    /* Pone dirección CGRAM */
46
47 #define LCDLinea1      B10000000    /* Posición 0 (00h). Línea 1 */
48 #define LCDLinea2      B11000000    /* Posición 64 (40h). Línea 2 */
49
50 unsigned char LCD_Counter1, LCD_Counter2;
51
52 // Subrutinas para el manejo del LCD .....
53
54 void LCD_Port()
55 {
56     TRISA = TRISA & 0xF8;          // RA0..RA2 salidas
57     TRISB = 0;                    // PORTB todo salidas
58     LCD_E = 0;
59 }
60
61 void Pausa_5ms()
62 {
63     // Bucle de retardo para el LCD
64     #asm
65         CLRF      _LCD_Counter1      ; Wipe first counter
66         MOVLW     1Ah                 ; Load W with timer Constant
67         MOVWF     _LCD_Counter2      ; for Counter2
68     _LCDLoop
69         DECFSZ    _LCD_Counter1,F    ; Decrement first counter
70         GOTO      _LCDLoop           ; 256 times
71         DECFSZ    _LCD_Counter2,F    ; Decrement, second counter
72         GOTO      _LCDLoop           ; from constant, above
73     #endasm
74 }
75
76 void LCD_Chequea()
77 {
78     /*
79         LCD_RW     = 1;              // Lee
80         TRISB      = 0xFF;          // Todo entradas
81         LCD_E       = 1;              // (Activa LCD)
82
83         while(RB7);                  // Comprueba que no esté ocupado
84
85
86         LCD_E      = 0;              // Desactiva LCD
87         TRISB      = 0x00;          // Todo salidas
88         LCD_RW     = 0;              // Escribe
89     */
90     // Proteus no lo simula bien asi que lo hago asi
91     Pausa_5ms();
92 }
93
94 void LCD_Comando_Inicio(unsigned char c)
95 {
96     LCD_RW = 0;    // Escribe
97     LCD_RS = 0;    // Comando
98     LCD_Data = c;  // Pone comando
99     // Durante el arranque no podemos leer el estado
100     LCD_E = 1;    // Habilita LCD
101     LCD_E = 0;    // Deshabilita LCD
102 }
103
104 void LCD_Comando(unsigned char c)
105 {
106     LCD_RW = 0;    // Escribe
107     LCD_RS = 0;    // Comando
108     LCD_Data = c;  // Pone comando
109     LCD_Chequea(); // Comprueba si no está ocupado
110     LCD_E = 1;    // Habilita LCD

```

```
111     LCD_E      = 0;    // Deshabilita LCD
112 }
113
114 void LCD_Character(unsigned char c)
115 {
116     LCD_RW      = 0;    // Escribe
117     LCD_RS      = 0;    // Comando
118     LCD_Chequea();    // Comprueba si no está ocupado
119     LCD_Data    = c;    // Pone el dato
120     LCD_RS      = 1;    // Dato
121     LCD_E       = 1;    // Habilita LCD
122     LCD_E       = 0;    // Deshabilita LCD
123 }
124
125 void LCD_Reset()
126 {
127     LCD_Comando_Inicio(LCDFuncion); // Define modo de operación
128     Pausa_5ms();
129     LCD_Comando_Inicio(LCDFuncion); //
130     Pausa_5ms();
131     LCD_Comando_Inicio(LCDFuncion); //
132     Pausa_5ms();
133     LCD_Comando_Inicio(LCDFuncion); //
134     Pausa_5ms();
135
136     LCD_Comando(LCDClear );           // Borra pantalla
137     LCD_Comando(LCDInc );             // Autoincremento posición
138     LCD_Comando(CursBlink);           // Cursor parpadeando
139 }
```
