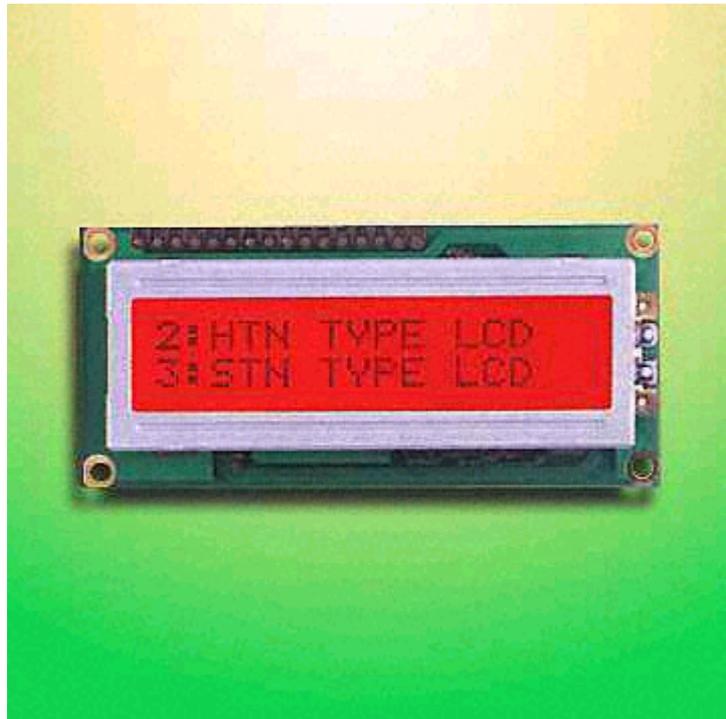




MODULO LCD





MODULO LCD

INTRODUCCIÓN

Las siguientes páginas intentan dar una breve descripción del funcionamiento del módulo LCD incluido en el sistema de evaluación μ PIC Trainer el WM-C1602M.

Se trata de un módulo microcontrolado capaz de representar 2 líneas de 16 caracteres cada una. A través de 8 líneas de datos se le envía el carácter ASCII que se desea visualizar así como ciertos códigos de control que permiten realizar diferentes efectos de visualización. Igualmente mediante estas líneas de datos el módulo devuelve información de su estado interno.

Con otras tres señales adicionales se controla el flujo de información entre el módulo LCD y el equipo informática que lo gestiona.

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

Pin nº	Símbolo	Conexión	Descripción
1	V_{ss}	V_{ss}	Patilla de tierra de alimentación
2	V_{dd}	V_{dd}	Patilla de alimentación de +5V
3	V_o	V_o	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	RA0	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	RA1	Señal de lectura/escritura R/W=0 El Módulo LCD es escrito R/W=1 El Módulo LCD es leído
6	E	RA2	Señal de activación del módulo LCD: E=0 Módulo desconectado E=1 Modulo conectado
7-14	D0-D7	RB0-RB7	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



Patilla	Nombre	Nivel	Descripción
1	Vss	0 V.	Masa
2	Vcc	5 V.	Alimentación
3	Vee	NOTA	Polarización del cristal
4	RS	lógico	Selección: 1=Datos; 0=Instrucción
5	R/W	"	Read/Write: 1=Lectura; 0=Escritura
6	E	"	Habilitación
7-14	DB0-DB7	"	Bus de datos (DB0, LSB)

INTEREASE CON-EL μ PIC TRAINER

El internase entre este módulo LCD y el sistema μ PIC TRAINER se realiza, de acuerdo con el esquema del equipo de siguiente manera:

- **RBO-RB7** : Están conectadas a las líneas de datos DO-D7 del módulo. A través por tanto, de la puerta B, se envían códigos ASCII o de control al módulo o, se recibe por parte de este, el estado interno del mismo.

EL Port B habrá que programarlo como salida cuando se vayan a enviar códigos ASCII o de control y como entrada cuando se desee conocer el estado interno del módulo.

- **RA0**: Se conecta con la señal de control **R/S**. Sacando un nivel lógico "0" por esta línea se selecciona el registro de control del módulo. Sacando un nivel lógico " 1 " se selecciona el registro de datos. *Esta línea debe programarse como salida.*
- **RA1**: Se conecta con la señal **R/W**. Sacando un nivel lógico "0" por ella, el módulo es escrito con la información presente en ese momento en el Port B que deberá actuar como salida. Sacando un " 1 " se lee el estado interno del módulo LCD. Dicho estado se recibe a través del Port B que deberá estar programado como entrada. *La línea RA1 debe programarse como salida.*
- **RA2**: Se conecta con la señal **E**. Cuando se aplica un nivel "1" el módulo queda habilitado y es posible por tanto la transferencia de información entre el Port B y las líneas de datos DO-D7. Aplicando un "0" el módulo queda desconectado y sus líneas de datos DO-D7 en alta impedancia. *RA2 debe programarse también como salida.*

**JUEGO DE INSTRUCCIONES:**

Seguidamente se presenta una serie de comando o instrucciones que permiten configurar diferentes opciones de trabajo del módulo LCD y conseguir con ello distintos efectos de visualización

El juego de instrucciones consiste en diferentes códigos que se introducen a través del Bus de datos del módulo LCD conectado al Port B de μ PIC Trainer

➤ 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:

RA0	RA1	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

Tiempo de ejecución:1.64 mS

➤ 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:

RA0	RA1	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	X

Tiempo de ejecución:1.64 mS

**➤ 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:

RA0	RA1	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	S

Tiempo de ejecución:40 μS

➤ 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:

RA0	RA1	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

Tiempo de ejecución:40 μS

➤ 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:

RA0	RA1	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	X	X

Tiempo de ejecución:40 μS

**➤ FUNCTION SET**

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

Código:

RA0	RA1	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	X	X

Tiempo de ejecución:40 μS

➤ SET THE CGRAM 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 6 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:

RA0	RA1	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	Dirección de la CG RAM					

Tiempo de ejecución:40 μs

**➤ SET THE DD RAM 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 almacenado 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:

RA0	RA1	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	Dirección de la RAM						

Tiempo de ejecución:40 µs

➤ READ BUSY FLAG & 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:

RA0	RA1	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	BF	Dirección de la CG RAM o de la DDRAM						

Tiempo de ejecución:1 µs

**➤ WRITE DATA TO CG OR DDRAM**

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:

RA0	RA1	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	Código ASCII o byte del carácter gráfico							

Tiempo de ejecución:40 μs

➤ READ DATA FRW 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:

RA0	RA1	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	Código ASCII o byte del carácter gráfico							

Tiempo de ejecución:40 μs

**ABREVIATURAS**

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

Abreviatura	Descripción
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 dos líneas
N=0	LCD de una línea
F=1	Carácter de 5 x 10 puntos
F=0	Carácter de 5 x 7 puntos
B=1	Parpadeo de cursor ON
C=1	Cursor ON
D=1	Display ON
X=1	Interlineado



JUEGO DE CARACTERES

Es el mostrado en la siguiente Figura. Las posiciones marcadas como CG RAM (n) corresponden a uno de los 8 posibles caracteres gráficos definidos por el usuario.



Higher -Order Lower- Order Bits 4 bit	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)		0	1	P	\	P		-	9	E	α	ρ
xxxx0001	(2)	!	1	A	Q	a	9	u	7	#	Δ	ä	q
xxxx0010	(3)	"	2	B	R	b	r	Γ	ι	ψ	×	ρ	θ
xxxx0011	(4)	#	3	C	S	c	s	∫	∅	τ	ε	ε	ω
xxxx0100	(5)	\$	4	D	T	d	t	∪	∩	∫	∫	μ	α
xxxx0101	(6)	%	5	E	U	e	u	∩	∩	∩	∩	∩	∩
xxxx0110	(7)	&	6	F	V	f	v	∩	∩	∩	∩	∩	∩
xxxx0111	(8)	'	7	G	W	g	w	∩	∩	∩	∩	∩	∩
xxxx1000	(1)	(8	H	X	h	x	∩	∩	∩	∩	∩	∩
xxxx1001	(2))	9	I	Y	i	y	∩	∩	∩	∩	∩	∩
xxxx1010	(3)	*	#	J	Z	j	z	∩	∩	∩	∩	∩	∩
xxxx1011	(4)	+	;	K	L	k	l	∩	∩	∩	∩	∩	∩
xxxx1100	(5)	,	<	L	*	l	*	∩	∩	∩	∩	∩	∩
xxxx1101	(6)	-	=	M	I	m	i	∩	∩	∩	∩	∩	∩
xxxx1110	(7)	.	>	N	^	n	→	∩	∩	∩	∩	∩	∩
xxxx1111	(8)	/	?	O	_	o	←	∩	∩	∩	∩	∩	∩



CARACTERES GRAFICOS

El usuario puede definir hasta 8 caracteres de 5 x 7 puntos o 4 de 5 x 10. Se seleccionan y visualizan aplicando a la DD RAM cualquier valor entre 00 y 07 ó 08 y 0Fh como si de un código ASCII se tratara.

Se definen introduciendo en sucesivas direcciones de la CG RAM unos bytes cuyos patrones binarios definen el carácter tal y como se muestra en la siguiente Tabla.

Un carácter de 5 x 7 necesita de 8 octetos en la CG RAM para ser definido, uno de 5 x 10 necesita de 16. La CG RAM es una memoria de 64 posiciones en total.

En el ejemplo de la Tabla, para definir la R en 5 x 7 se introducen 8 octetos en las 8 primeras posiciones (0 a la 7) de la CG RAM. Cada bit de cada uno de esos octetos que valga nivel "1" implica que su correspondiente pixel en el LCD se active.

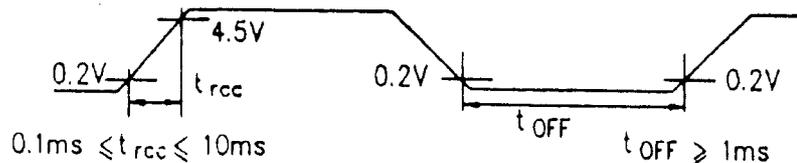
Como es el primer conjunto de 8 bytes es decir, el primer carácter de la CG RAM, este se selecciona aplicando el código 00 en la DD RAM como si fuera cualquier otro código ASCII.

Character codes (DD RAM data)	CG RAM address	Character patterns (CG RAM data)	
7 6 5 4 3 2 1 0 Higher-order bits ← Lower-order bits	5 4 3 2 1 0 Higher-order bits ← Lower-order bits	7 6 5 4 3 2 1 0 Higher-order bits ← Lower-order bits	
0 0 0 0 * 0 0 0	0 0 0	* * * 1 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 1 1 1 0 1 0 1 0 0 1 0 0 1 0 1 1 1 0 0 1 1 1	Character Pattern#1
0 0 0 * 0 0 1	0 0 1	* * * 1 0 0 0 0 0 1 0 1 0 1 1 1 1 1 0 0 1 0 0 1 1 1 1 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0	Cursor Position Character Pattern#2
0 0 0 0 * 1 1 1	1 1 1	* * * 1 0 0 1 0 1 1 1 0 1 1 1	



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 el siguiente cronograma.



Dichos requisitos consisten en que el tiempo que tarde en estabilizarse la tensión desde 0.2 V hasta los 4.5V mínimos necesarios 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. Se ejecuta el comando **CLEAR DISPLAY** borrando la pantalla. El flag **BUSY** se mantiene a "1" (ocupado) durante 15 mS hasta que finaliza la inicialización.
2. Se ejecuta el comando **FUNCTION SET**, que establece el interfaz con el Bus de datos. 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. Se ejecuta el comando **DISPLAY ON/OFF CONTROL**, que hace que el display que en OFF (D=0); también cursor en OFF (C=0) y sin parpadeo del cursor en (B=0)
4. Se ejecuta el comando **ENTRY MODE SET**, que establece la dirección de movimiento del cursor con autoincremento del cursor (I/D=1) y modo normal, no desplazamiento, del display (S=0).

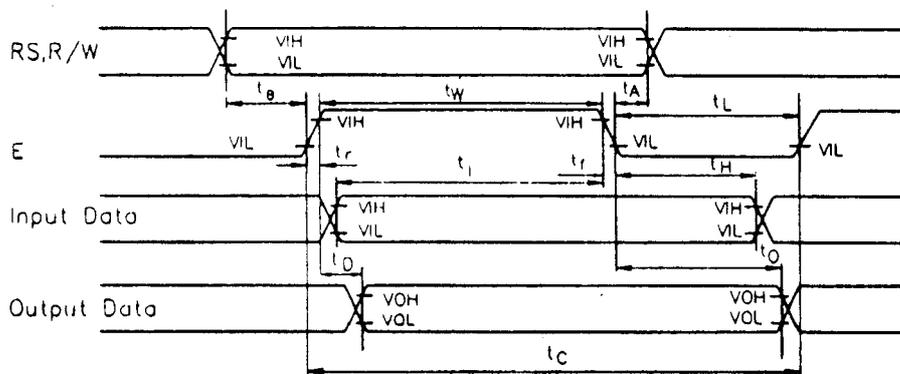
Si la conexión de la alimentación no reúne las condiciones que exige el módulo LCD, habría que realizar la secuencia de inicialización por software. En cualquier caso, es importante enviar al LCD la primera instrucción de trabajo



después de que hayan transcurrido 15 ms, para completar dicha secuencia de inicialización.

DIAGRAMA DE TIEMPOS

Es el mostrado en la figura siguiente. Junto con la tabla que acompaña se conocen los tiempos de las señales para diferentes chips controladores de módulos



LCD.

CHIP DE CONTROL		SAMSUNG	HITACHI	SANYO	EPSON	OKI	RECOMMENDED	UNIT
PARAMETRO		KS0066	HD44780	LC7985 NA	SED1278	MSM6222	TIMING	
Enable Cycle Time	IC (min)	1000	1000	1000	500	667	1000	nS
Enable Pulse Width	High Level	t_W (min)	450	450	450	220	450	nS
	Low Level	t_L (min)	450	450	450	220	450	nS
E Raise Time	t_r (max)	25	25	25	25	25	25	nS
E Fall Time	t_f (max)	25	25	25	25	25	25	nS
Set-up Time	t_B (min)	140	140	140	40	140	140	nS
Data Set-up Time	t_I (min)	195	195	195	60	180	195	nS
Data Delay Time	t_D (max)	320	320	320	120	220	320	nS
Address Hold Time	t_A (max)	10	10	10	10	10	10	nS
Hold Time	t_H (min)	10	10	10	10	10	10	nS
Input Data	t_O (min)	20	20	20	20	20	20	nS
Output Data	t_C (min)	20	20	20	20	20	20	nS



RUTINAS DE CONTROL

En este apartado se presentan una serie de rutinas escritas en el ensamblador MPASM de MICROCHIP con objeto de proporcionar al usuario una idea de cómo gestionar las diferentes actividades a realizar con el módulo LCD. Están realizadas basándose en un PIC 16X84 a 4 MHz de velocidad.

LCD_E

Genera un pulso por la patilla RA2 (señal E) para activar al módulo LCD. El pulso es de 1µS de duración trabajando a 4 MHz. En caso de frecuencias mayores de trabajo habrá que aumentar este tiempo para no rebasar la duración mínima impuesta por el fabricante del LCD.

```
LCD_E    bsf    RA,2        ;Activa señal E
          nop              ;Espera de 1µS
          bcf    RA,2        ;Desactiva señal E
          retum
```

LCD-BUSY

Chequea el estado del flag Busy del LCD y espera a que finalice cualquier instrucción previa antes de retomar.

```
LCD-BUSY bsf    RA,1        ;Pone el LCD en modo lectura
          bsf    STATUS,S    ;Selecciona el banco 1
          movlw 0xff
          movwf TRISB       ;Port B activo de entrada
          bcf    STATUS,S    ;Selecciona el banco 0
          bsf    RA,2        ;Activa el LCD (señal E)
          nop
L_BUSY   btfsc  RB,7        ;Chequea el bit busy
          goto  L_BUSY      ;Está "1 " (ocupado)
          bcf    RA,2        ;Desactiva el LCD (señal E)
          bsf    STATUS,5    ;Selecciona el banco 1
          clrf  TRISB       ;Port B activo de salida
          bcf    STATUS,5    ;Selecciona el banco 0
          bcf    RA,1        ;Pone el LCD en modo escrita
          retum
```



LCD_REG

Deposita el código de instrucción presente en el registro W del PIC sobre el PortB. Espera que el LCD ejecute la última operación y genera el pulso de activación en la señal E.

```
LCD_REG    bcf          RA,0          ;Desactiva RS (modo instrucción)
            movwf       RB           ;Saca el código de instrucción
            call        LCD_BUSY     ;Espera que se libere el LCD
            goto       LCD_E        ;Genera pulso en la señal E
```

LCD_DATOS

Deposita el código ASCII del carácter a visualizar presente en el registro W, sobre el Port B. Espera a que el LCD ejecute la última operación y genera el pulso de activación en la señal E.

```
LCD_DATOS  bcf          RA,0          ;Desactiva RS (modo instrucción)
            movwf       RB           ;Valor ASCU a sacar por RB
            call        LCD_BUS      ;Espera que se libere el LCD
            bsf         RA,0          ;Activa RS (modo dato)
            goto       LCD_E        ;Genera pulso en la señal E
```

LCD_INI

Realiza la inicialización del módulo LCD según los tiempos marcados por el fabricante (15 mS). En este ejemplo se inicia con un internase de 8 bits de bus de datos, 2 líneas de visualización y caracteres de 5 x 7 puntos.

```
LCD-INI    movlw       b'001110001
            call        LCD_REG       ;Código de instrucción
            call        DELAY_5MS     ;Temporiza 5 mS
            movlw       b'001110001
            call        LCD_REG       ;Código de instrucción
            call        DELAY_5MS     ;Temporiza 5 mS
            movlw       b'00111000'
            call        LCD-REG       ;Código de instrucción
            call        DELAY-5MS     ;Temporiza 5 mS
            return
```

**DELAY_5MS**

Genera una temporización de 5 mS. Se emplean dos variables llamadas DATO_A y DATO_B que se decrementan hasta completar la temporización.

```
DELAY_5MS    movlw    0x1A
              movwf   DATO-B    ;Carga la variable DATO -B
              clrf    DATO_A    ;Carga la variable DATO-A
DELAY-1      decfsz   DATO_A,1   ;Decrementa la variable DATO-A
              goto    DELAY-1
              decfsz   DATO_B,1   ;Decrementa la variable DATO-B
              goto    DELAY_1
              return
```

DELAY 10 ms

Genera, con ayuda del TMRO y el preescaler de 256, una temporización de unos 10 mS que se repite tantas veces como indique la variable TEMPO1.

```
DELAYI0      bcf      INTCON,2    ;flag de estado del TMRO
              movlw   0xD8
              movwf   TMERO        ;Carga TMRO para que cuente 39
DELAY10_1    btfss   INTCON,2    ;Espera overflow del TMRO
              goto    DELAY10_1
              decfsz  TEMPO1,1    ;Repite TEMPO1 veces
              goto    DELAY10
              return
```

DELAY 1 s

En base a la rutina anterior se realiza una temporización de 1s cargando en la variable TEMPO1 el valor 100 (64h).

```
DELAYIS      movlw   0x64
              movwf  TEMPO1    ;Caiga la variable TEMIP01
              call   DELAYLS
              return
```



INSTRUCCIONES CON EL LCD

Instrucción	Código										Descripción	Tiempo max de ejecución
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Borra el display y coloca el cursor en la primera posición 0 DDRAM	82µs~1.64ms
Return home	0	0	0	0	0	0	0	0	0	1	* Coloca el cursor en la posición de inicio y hace que el display comience a desplazarse desde la posición original. El contenido de la DDRAM no varía	40µs~1.64ms
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	Establece el sentido de desplazamiento de la información en el display. Esta operación se realiza durante la lectura o escritura de la DDRAM	40µs
Display ON/OFF control	0	0	0	0	0	0	1	D	C	B	Activa o desactiva poniendo en ON/OFF tanto el display D=0 (off) o D=1(on),como el cursor C=0(off) o C=1(on) y establece si este debe parpadear o no B=0(off) o B=1(on)	40µs
Cursor or display shift	0	0	0	0	0	1	S/C	R/L	*	*	Mueve el cursor y desplaza el display sin cambiar el contenido de la DDRAM	40µs
Funtion set	0	0	0	0	1	DL	N	F	*	*	Establece el tamaño de interfase con el bus de datos(DL),el número de líneas del display(N) y la font de los caracteres	40µs
CG RAM address set	0	0	0	1	Dirección CGRAM					Establece la dirección de CGRAM a partir de la cual se almacenan los caracteres de usuario	40µs	
DD RAM address set	0	0	1	Dirección de la DDRAM					Estable la dirección DDRAM a partir de la cual se almacenan los datos a visualizar	40µs		
Read Busy Flag and Address	0	1	BF	Dirección de DDRAM o CGRAM					Lectura del flag de Busy e indica de la dirección de la CGRAM o DDRAM última empleada.	1µs		
Write data into the CG RAM or the DDRAM	1	0	Dato a escribir					Escribe en DDRAM o CGRAM los datos que se quieren presentar en el LCD	40µs			
Read data from the CG RAM or the DDRAM	1	1	Dato a leer					Lee de la DDRAM o CGRAM los datos que se direccionen	40µs			
	I/D = 1 Incrementa la dirección del cursor I/D = 0 Decrementa la dirección del cursor S = 1 Desplaza la visualización cada vez que se escribe un dato S/C = 1 Se desplaza la visualización S/C = 0 Se desplaza el cursor R/L = 1 Desplazamiento a la derecha R/L = 0 Desplazamiento a la izquierda DL = 1 Trabaja con bus de datos de 8 bits DL = 0 Trabaja con bus de datos de 4 bits N = 1 La presentación se hace en 2 líneas N = 0 La presentación se hace en 1 línea F = 1 Caracteres de 5x10 dots F = 0 Caracteres de 5x7 dots BF = 1 LCD ocupado en una operación interna BF = 0 LCD disponible para aceptar instrucciones D = 1 Pantalla activa (ON) C = 1 Cursor activo (ON) * Indeterminado										DDRAM: Dispaly Data RAM CGRAM: Character Generator RAM ACC : Address of CGRAM ADD : Address od DDRAM AC: Address counter used for botch DDRAM and CGRAM	