

Uso del LCD externo tipo Hitachi HD44780 conectado a PIC 18 con el compilador XC8:

Adaptado de MPLAB® C18 C COMPILER LIBRARIES (DS51297F)

por: Rommel Contreras - rommeljose@gmail.com

Descripciones de funciones

BusyXLCD

Función:	¿Está ocupado el controlador LCD?
Incluir:	<code>xlcd.h</code>
Prototipo:	<code>unsigned char BusyXLCD(void);</code>
Observaciones:	Esta función devuelve el estado de la bandera de ocupado del controlador LCD Hitachi HD44780.
Valor de retorno:	1 si el controlador está ocupado 0 en caso contrario.

Nombre del archivo: `busyxlcd.c`

Ejemplo de código: `while(BusyXLCD());`

OpenXLCD

Función:	Configurar los pines de E/S del PIC® e inicializar el controlador del LCD.
Incluir:	<code>xlcd.h</code>
Prototipo:	<code>void OpenXLCD(unsigned char <i>lcdtype</i>);</code>
Argumentos:	<p><i>lcdtype</i></p> <p>Una máscara de bits que se crea realizando una operación AND a nivel de bits ('&') con un valor de cada una de las categorías enumeradas a continuación. Estos valores se definen en el archivo <code>xlcd.h</code>.</p> <p>Interfaz de datos:</p> <p><code>FOUR_BIT</code> Modo de interfaz de datos de 4 bits</p> <p><code>EIGHT_BIT</code> Modo de interfaz de datos de 8 bits</p> <p>Configuración del LCD:</p> <p><code>LÍNEA_5X7</code> 5x7 caracteres, visualización de una sola línea</p> <p><code>LÍNEA_5X10</code> Visualización de 5x10 caracteres</p> <p><code>LINEAS_5X7</code> 5x7 caracteres, visualización de varias líneas</p>
Observaciones:	Esta función configura los pines de E/S del PIC18 utilizados para controlar el controlador de LCD Hitachi HD44780. También inicializa este controlador.

Nombre del archivo: `openxlcd.c`

Ejemplo de código: `OpenXLCD(EIGHT_BIT & LINES_5X7);`

putcXLCD

Ver `WriteDataXLCD`.

putsXLCD putsXLCD

Función:	Escribir una cadena en el controlador LCD Hitachi HD44780.
Incluir:	xlcd.h
Prototipo:	<pre>void putsXLCD(char *buffer); void putsXLCD(const rom char *buffer);</pre>
Argumentos:	<i>buffer</i> Puntero a los caracteres que deben escribirse en el controlador de la pantalla LCD.
Observaciones:	<p>Esta función escribe una cadena de caracteres situada en el <i>búfer</i> en el controlador LCD Hitachi HD44780. Detiene la transmisión cuando se encuentra un carácter nulo. El carácter nulo no se transmite.</p> <p>Las cadenas situadas en la memoria de datos deben utilizarse con las versiones "puts" de estas funciones.</p> <p>Las cadenas situadas en la memoria del programa, incluidas las literales de cadena, deben utilizarse con las versiones "puts" de estas funciones.</p>
Nombre del archivo:	putsxlcd.c putrxlcd.c
Ejemplo de código:	<pre>char mybuff [20]; putsXLCD("Hola Mundo"); putsXLCD(mybuff);</pre>

ReadAddrXLCD

Función:	Lea el byte de dirección del controlador LCD Hitachi HD44780.
Incluir:	xlcd.h
Prototipo:	<pre>unsigned char ReadAddrXLCD(void);</pre>
Observaciones:	<p>Esta función lee el byte de dirección del controlador LCD Hitachi HD44780. El controlador del LCD no debe estar ocupado cuando se realiza esta operación - esto puede ser verificado usando la función <code>BusyXLCD</code>.</p> <p>La dirección leída desde el controlador es para la RAM del generador de caracteres o la RAM de datos de la pantalla, ¿dependiendo de la función anterior <code>Set? RamAddr</code> que fue llamada.</p>
Valor de retorno:	Esta función devuelve una cantidad de 8 bits. La dirección está contenida en los 7 bits de orden inferior y la bandera de estado BUSY en el bit más significativo.
Nombre del archivo:	readaddr.c
Ejemplo de código:	<pre>char addr; while (BusyXLCD()); addr = ReadAddrXLCD();</pre>

ReadDataXLCD

Función:	Leer un byte de datos del controlador LCD Hitachi HD44780.
Incluir:	xlcd.h
Prototipo:	<pre>char ReadDataXLCD(void);</pre>

Observaciones: Esta función lee un byte de datos del controlador LCD Hitachi HD44780. El controlador del LCD no debe estar ocupado cuando se realiza esta operación - esto puede verificarse utilizando la función `BusyXLCD`.
Los datos leídos desde el controlador son para la RAM del generador de caracteres o la RAM de datos de la pantalla, ¿dependiendo de la función anterior `SetRamAddr` que fue llamada.

Valor de retorno: Esta función devuelve el valor de los datos de 8 bits.

Nombre del archivo: `readdata.c`

Ejemplo de código: `datos de char;
while (BusyXLCD()); data = ReadAddrXLCD();`

SetCGRamAddr

Función: Establece la dirección del generador de caracteres.

Incluir: `xlcd.h`

Prototipo: `void SetCGRamAddr(unsigned char addr);`

Argumentos: *dirección*
Dirección del generador de caracteres.

Observaciones: Esta función establece la dirección del generador de caracteres del controlador LCD Hitachi HD44780. El controlador de LCD no debe estar ocupado cuando se realiza esta operación - esto puede verificarse utilizando la función `BusyXLCD`.

Nombre del archivo: `setcgram.c`

Ejemplo de código: `char cgaddr = 0x1F;
while(BusyXLCD()); SetCGRamAddr(cgaddr);`

SetDDRamAddr

Función: Establece la dirección de los datos de la pantalla.

Incluir: `xlcd.h`

Prototipo: `void SetDDRamAddr(unsigned char addr);`

Argumentos: *dirección*
Mostrar la dirección de los datos.

Observaciones: Esta función establece la dirección de datos de la pantalla del controlador LCD Hitachi HD44780. El controlador de LCD no debe estar ocupado cuando se realiza esta operación - esto se puede verificar utilizando la función `BusyXLCD`.

Nombre del archivo: `setddram.c`

Ejemplo de código: `char ddaddr = 0x10;
while(BusyXLCD()); SetDDRamAddr(ddaddr);`

WriteCmdXLCD

Función: Escriba un comando en el controlador LCD Hitachi HD44780.

Incluir: `xlcd.h`

Prototipo: `void WriteCmdXLCD(unsigned char cmd);`

Argumentos:***cmd***

Especifica el comando a realizar. El comando puede ser uno de los siguientes valores definidos en `xlcd.h`:

<code>DOFF</code>	Apagar la pantalla
<code>CURSOR_OFF</code>	Habilitar la visualización sin cursor
<code>BLINK_ON</code>	Habilitar la visualización con el cursor parpadeante
<code>BLINK_OFF</code>	Activar la visualización con el cursor sin parpadear
<code>SHIFT_CUR_LEFT</code>	El cursor se desplaza hacia la izquierda
<code>SHIFT_CUR_RIGHT</code>	El cursor se desplaza a la derecha
<code>SHIFT_DISP_LEFT</code>	La pantalla se desplaza hacia la izquierda
<code>SHIFT_DISP_RIGHT</code>	La pantalla se desplaza hacia la derecha

Alternativamente, el comando puede ser una máscara de bits que se crea realizando una operación AND a nivel de bits ('& ') con un valor de cada una de las categorías enumeradas a continuación. Estos valores se definen en el archivo `xlcd.h`.

Modo de transferencia de datos:

<code>FOUR_BIT</code>	Modo de interfaz de datos de 4 bits
<code>EIGHT_BIT</code>	Modo de interfaz de datos de 8 bits

Tipo de pantalla:

<code>LÍNEA_5X7</code>	5x7 caracteres, una sola línea
<code>LÍNEA_5X10</code>	Visualización de 5x10 caracteres
<code>LÍNEAS_5X7</code>	5x7 caracteres, varias líneas

Observaciones:

Esta función escribe el byte de comando en el controlador LCD Hitachi HD44780. El controlador del LCD no debe estar ocupado cuando se realiza esta operación - esto puede verificarse utilizando la función `BusyXLCD`.

Nombre del archivo: `wcmdxlcd.c`**Ejemplo de código:**

```
while( BusyXLCD() );
WriteCmdXLCD( EIGHT_BIT & LINES_5X7 );
WriteCmdXLCD( BLINK_ON ); WriteCmdXLCD( SHIFT_DISP_LEFT );
```

putcXLCD WriteDataXLCD

Función:

Escribe un byte en el controlador LCD Hitachi HD44780.

Incluir:

`xlcd.h`

Prototipo:

```
void WriteDataXLCD( char data );
```

Argumentos:***datos***

El valor de los *datos* puede ser cualquier valor de 8 bits, pero debe corresponder a la tabla de RAM de caracteres del controlador LCD HD44780.

Observaciones:

Esta función escribe un byte de datos en el controlador LCD Hitachi HD44780. El controlador del LCD no debe estar ocupado cuando se realiza esta operación - esto puede verificarse utilizando la función `BusyXLCD`.

Los datos leídos desde el controlador son para la RAM del generador de caracteres o la RAM de datos de la pantalla, ¿dependiendo de la función anterior Set? `RamAddr` que fue llamada.

Ejemplo de USO:

```
#include "../mcc_generated_files/mcc.h"
#include "../Lib_mias_all/microchip_LCD/xlcd.h"
#include "../Lib_mias_all/micelaneos.h"

/*          Programa Principal      */

void main(void)
{
    // Initialize the device
    SYSTEM_Initialize();

    LCD_inicio();

    while (1)
    {
        // Add your application code

        while (BusyXLCD()); //wait untill LCD controller is busy
        SetDDRamAddr(0x80); // Principio area visible LCD

        while( BusyXLCD() );
        putsXLCD("  CARIPITO ");
        __delay_ms(2000);

        while (BusyXLCD());
        SetDDRamAddr(0xC0); // Principio area visible LCD

        while( BusyXLCD() );
        putsXLCD(" MONAGAS, Venez. ");
        __delay_ms(2000);

        while (BusyXLCD()); // Wait if LCD busy
        WriteCmdXLCD(0x01); // Clear display

    }
}
/**
  End of File
*/
```

Referencias:

- **MPLAB® C18 C COMPILER LIBRARIES (DS51297F)**
El propósito de este documento es proporcionar información detallada sobre las bibliotecas y los archivos de objetos precompilados que se pueden utilizar con MPLAB de Microchip® C18 C Compiler.
- **MPLAB® IDE Quick Start Guide (DS51281)**
Describe cómo configurar el software IDE de MPLAB y usarlo para crear proyectos y dispositivos de programa.