



Proyecto:

Interfaz de teclado con PIC

Directores del proyecto:

Ingeniero de Diseño: M.C. Julio Cesar Silva Briano

Ingeniero de Firmware: M.C. Julio Cesar Silva Briano

Versiones e información del Documento:

Versión: 1.0

Archivo origen: Keyboard Documento detallado V10.doc

INDICE

1	Historia del Documento	1
2	Introducción	2
2.1	Propósito	2
2.2	A quien va dirigido el documento.....	2
2.3	Ámbito	2
3	Descripción General.....	3
3.1	Perspectiva del producto	3
3.2	Restricciones	3
4	Requerimientos del sistema	4
4.1	Requerimientos hardware	4
4.1.1	Modelo general a bloques	4
5	Diseño detallado.....	5
5.1.1	Pantalla de LCD.....	5
5.2	µC.....	6
5.2.1	Diagramas eléctricos.....	6
5.2.2	Programador para el microcontrolador PIC16F84.....	7
6	Desarrollo del Firmware.	9
7	Referencias.....	10
8	Apéndice A código del firmware.....	I
8.1	Keyblcd.c Creado por el titular del documento.	I
8.2	keyblcd.h Creado por el titular del documento.....	V
8.3	LcdTest.c Creado por J.Winpenny. Con modificaciones J.Silva.....	VI
8.4	LcdTest.h Creado por J.Winpenny. Con modificaciones J.Silva	IX
8.5	scancod.h Creado por el titular del documento.....	X
9	Creditos	XIV

FIGURAS

Figura 1	<i>Diagrama a bloques del proyecto</i>	4
Figura 2	<i>Pantalla LCD</i>	5
Figura 3	<i>Diagrama de pines</i>	6
Figura 4	<i>Conexiones de pines para el PIC16F84</i>	7

INDICE

Figura 5 <i>Diagrama eléctrico global del sistem</i>	7
Figura 6 <i>Programador para PIC16F84</i>	8

TABLAS

Tabla 1 <i>Historia del documento</i>	1
Tabla 2 <i>Definición de pines</i>	6

1 Historia del Documento

Versión	Fecha	Cambio hecho
1.0	25-07-2002	Primer documento escrito y propuesto.

Tabla 1 *Historia del documento.*

2 Introducción

El proyecto *Interfaz de teclado con PIC* es un proyecto para el laboratorio de electrónica. Es parte importante del proyecto una sencilla explicación de funcionamiento y diseño, es la intención de este documento ser la guía de diseño e implementación de un sistema como este.

2.1 Propósito

- Este documento servirá para el análisis y diseño del hardware, Firmware y Software del *Interfaz de teclado con PIC*.
- Desarrollo de una herramienta clásica.

2.2 A quien va dirigido el documento.

Este documento va dirigido a la comunidad de analistas y diseñadores de firmware para microcontroladores.

2.3 Ámbito

Las funciones del sistema son:

- Lectura del PS/2 usando un servicio de interrupción.
- Despliegue de la tecla obtenida.

Los objetivos son:

- Servir como práctica.

3 Descripción General

3.1 Perspectiva del producto

Es importante que la comunicada estudiantil o el pequeño laboratorio puedan construir algunas de las herramientas de trabajo, y también pueda ser una introducción de lenguaje C aplicada en Firmware.

3.2 Restricciones

- Solo lee datos, no se tiene comunicación del PIC al PS/2, pues la comunicación es solo en un sentido.

4 Requerimientos del sistema

4.1 Requerimientos hardware

La lectura de una señal asíncrona puede ser detectada con un microcontrolador. Se opta por el uso de un microcontrolador de la familia MICROCHIP PIC16F84. Como no es una señal analógica si no una señal TTL basta con el uso de un microcontrolador y no de un DSP.

4.1.1 Modelo general a bloques

El siguiente diagrama muestra en general los elementos principales del proyecto.

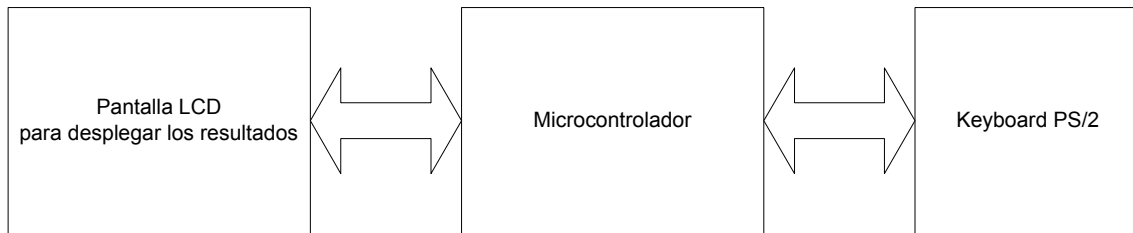


Figura 1 Diagrama a bloques del proyecto

Los bloques son:

- **Pantalla LCD:** elemento externo que sirve de interfaz, el microcontrolador se comunica con este para informar de los resultados.
- **μC :** Microcontrolador que hace el procesamiento de la información obtenida de la interfaz con la señal a medir.
- **PS/2:** Teclado estandar 101/102 teclas.

5 Diseño detallado

5.1.1 Pantalla de LCD

La interfaz LCD es un medio visual en el cual se pueden escribir datos ASCII para mayor detalle, monitoreo o depuración de un sistema basado en un microcontrolador, bastante extendido en los sistemas mínimos. Se puede encontrar código fuente para una gran variedad de arquitecturas de microcontroladores.

Notas:

- El código fue creado para manejo de un LCD 16x2 trabajando en modo 4 bits.
- LCD tipo HD44780

La siguiente figura muestra una pantalla de LCD y sus diferentes pines.

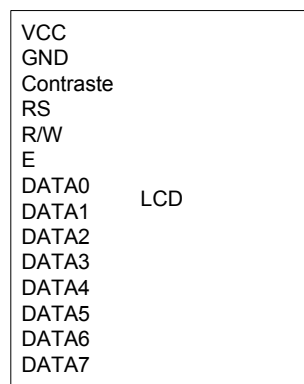


Figura 2 *Pantalla LCD*

Ahora se explica cada una de sus líneas

I/O	Nombre	Descripción
Salida	DATA0	Bus de datos para transferencia de código ASCII.
Salida	DATA1	Bus de datos para transferencia de código ASCII.
Salida	DATA2	Bus de datos para transferencia de código ASCII.
Salida	DATA3	Bus de datos para transferencia de código ASCII.
Salida	DATA4	Bus de datos para transferencia de código ASCII.
Salida	DATA5	Bus de datos para transferencia de código ASCII.
Salida	DATA6	Bus de datos para transferencia de código ASCII.
Salida	DATA7	Bus de datos para transferencia de código ASCII.
Salida	RS	Diferencia si es datos o instrucción.
Salida	R/W	Control de escritura o lectura.

Descripción General

Salida	E	Habilitador de acceso.
Entrada	VCC	Línea de +5V.
Entrada	GND	Línea de tierra.
Entrada	Contraste	Línea para cambiar la intensidad del LCD, use resistencia variable, para ajuste.

Tabla 2 Definición de pines

5.2 μ C

El microcontrolador elegido que reúne los requerimientos de desempeño fue PIC16F84, a continuación se muestra un resumen de las características del microcontrolador.

- Arquitectura RISC.
- Conjunto de 35 instrucciones.
- ISP.
- Timer 0, 8 bits
- Comunicaciones I2C, USART.
- 1K EEPROM.
- 68 Bytes de Memoria de datos.
- 13 Pines de I/O.
- 12 fuentes de interrupción.

A continuación se muestra el diagrama de Pins del 16F84

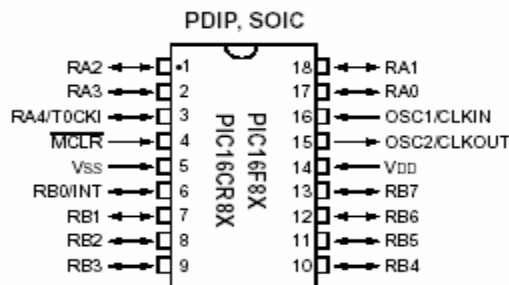


Figura 3 Diagrama de pines

5.2.1 Diagramas eléctricos.

La conexión del PIC se realiza de la siguiente manera:

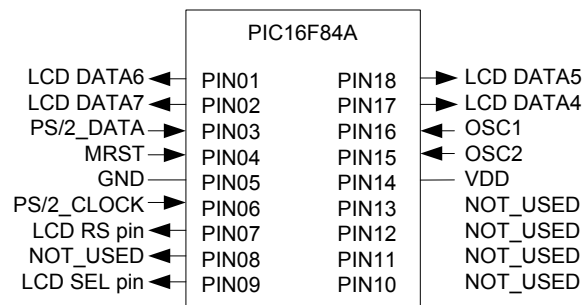


Figura 4 Conexiones de pines para el PIC16F84

Y el diagrama global del sistema es:

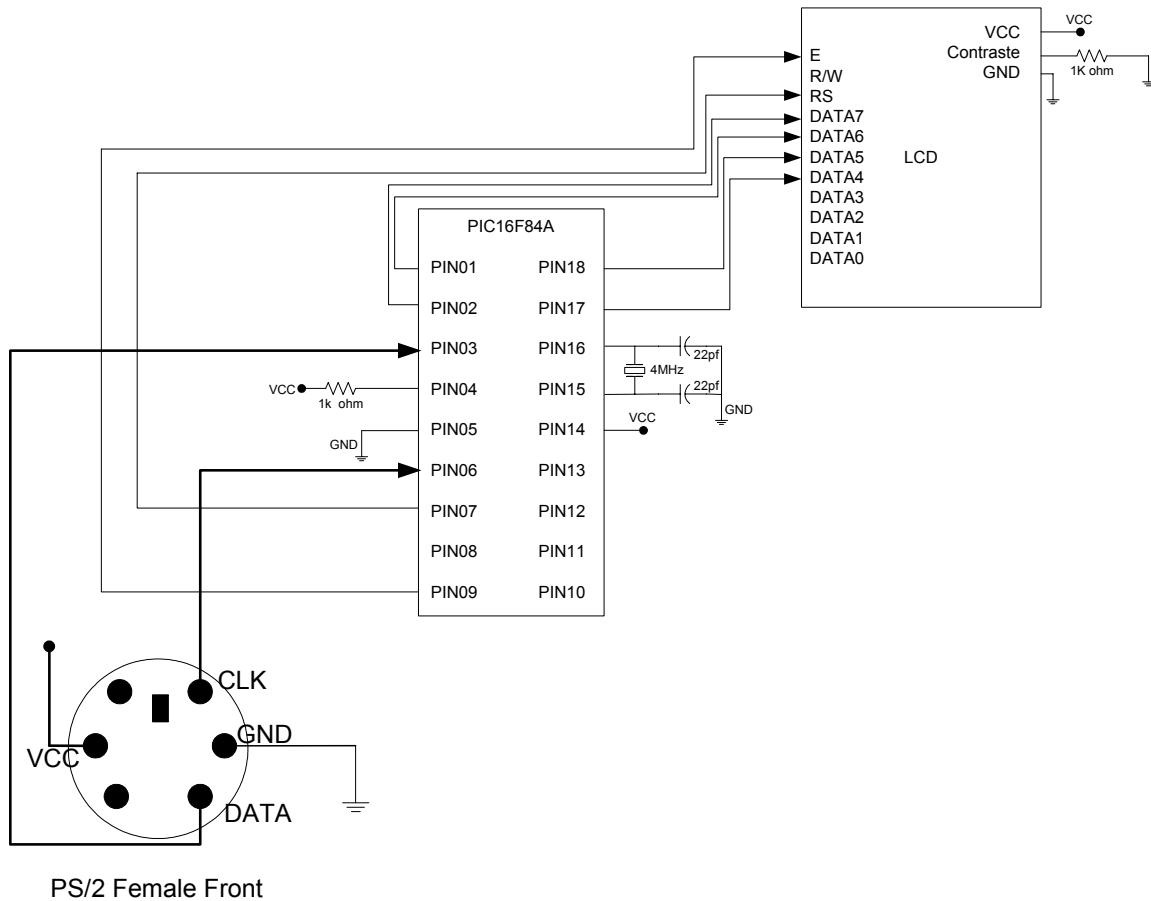


Figura 5 Diagrama eléctrico global del sistem

Las especificaciones son:

- El cristal de **4Mhz**
- Los capacitores que se colocan en el cristal son de **22pF**
- Conector Hembra minidin PS/2
- Pantalla LCD 16x2, común.

5.2.2 Programador para el microcontrolador PIC16F84

Es necesario un sistema económico para programar PICs a continuación se muestra el diagrama usado.

Descripción General

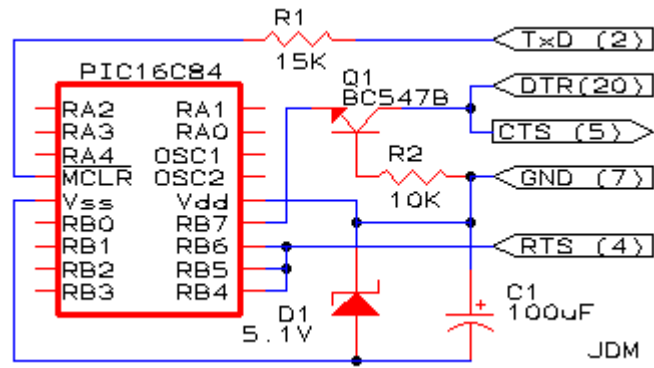


Figura 6 Programador para PIC16F84

El Software con el cual es compatible este bloque es el ICPROG, puede ser bajado de la página en Internet. www.ic-prog.com

6 Desarrollo del Firmware.

El Firmware fue desarrollado en C, una versión restringida del ANSI para microcontroladores PIC, este código es compilado con la herramienta C2C compilador de C, versión 4.00.7e. Se puede bajar el compilador de la página Web <http://www.picant.com/c2c/c.html>, se requiere junto con este, una versión de MPLAB. Se recomienda la versión 5.70.00 algunas anteriores pueden funcionar.

Los archivos que incluyen todo el proyecto son:

- keyblcd.c Creado por el titular del documento.
- keyblcd.h Creado por el titular del documento.
- LcdTest.c Creado por J. Winpenny.
- LcdTest.h Creado por J. Winpenny.
- scancod.h Creado por el titular del documento.

Estadísticas de código:

MEMORY USAGE MAP ('X' = Used, '-' = Unused)

```

0000 : X--XXXXXXXXXXXX XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX
0040 : XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX
0080 : XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX
00C0 : XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX
0100 : XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX
0140 : XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX
0180 : XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX
01C0 : XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX
0200 : XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX
0240 : XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX
0280 : XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX XXXXXXXXXX-----
2000 : -----X----- -----

```

All other memory blocks unused.

Program Memory Words Used: 695

Program Memory Words Free: 329

Errors : 0

Warnings : 0 reported, 0 suppressed

Messages : 0 reported, 3 suppressed

7 Referencias

Manual de C: <http://www.picant.com/c2c/doc/c2c.html>

Data sheet PIC16F84A: <http://www.microchip.com/download/lit/pline/picmicro/families/16f8x/35007b.pdf>

Detallado del LCD 16x2 1622A: <http://www.ee.vt.edu/cel/datasheets/hd44780.pdf>

8 Apéndice A código del firmware

8.1 Keyblcd.c Creado por el titular del documento.

```
//-----
//
// Main Project      : Keyboard and LCD
// C2C Project Name  : Keyblcd
// File Name         : keyblcd.c
// Version           : 1.0
// PIC Used          : PIC16F84A
// Compiler version   : C2C plus C-Compiler Version 4.00.7 English
// Assembler         : mpasmwin From MPLAB 5.70.00, Microchip Technology, Inc.
// Initial Date      : Decembar 15 2002
// Last Update date   : February 16 2002
// Firmware Eng.     : Julio Cesar Silva Briano
// Desing Eng.       : Julio Cesar Silva Briano
//
//-----
//
// Based Document:
//
// 1) File Name      :
// 2) File Name      : Data sheet : F84A Data sheet - 35007b.pdf
// 3) File Name      : C2C help index.pdf
//
// Note :
//
//-----
//
// Description of this C code:
//
// This routine continually executes this main routines
//
// 1) Control LCD 16x2
// 2) Keyboard decoder
//
//-----
//
// Hardware Description:
//
//
//
// PIN01-| PA.2      U      PA.1  |-PIN18
// PIN02-| PA.3      U      PA.0  |-PIN17
// PIN03-| CLKin     1      OSC1  |-PIN16
// PIN04-| RST       6      OSC2  |-PIN15
// PIN05-| GND       F      VCC   |-PIN14
// PIN06-| PB.0      8      PA.7   |-PIN13
// PIN07-| PB.1      4      PA.6   |-PIN12
// PIN08-| PB.2      A      PA.5   |-PIN11
// PIN09-| PB.3      U      PA.4   |-PIN10
//
//-----
//
// Used pines:
//
// PIN01 = LCD_DATA_6
// PIN02 = LCD_DATA_7
// PIN03 = DATA
// PIN04 = Not Used
// PIN05 = Ground
// PIN06 = CLOCK
// PIN07 = LCD_RS
// PIN08 = Not Used
// PIN09 = LCD_SEL
// PIN10 = Not Used
// PIN11 = Not Used
```

Documento Público creado por M.C. Julio Cesar Silva Briano

Apéndice A Código del Firmware

```
//      PIN12 = Not Used
//      PIN13 = Not Used
//      PIN14 = VCC
//      PIN15 = OSC2
//      PIN16 = OSC1
//      PIN17 = LCD_DATA_4
//      PIN18 = LCD_DATA_5
//
//-----
//
// Hardware NOTES:
//      Defalut OSC 4MHZ
//
//-----

asm __config 0x3FF9 ;//CODE FOR 16F84A ONLY!!

//-----
// INCLUDE SECTION
//-----
#include "LCDTest.c"
#include "keyblcd.h"
#include "scancod.h"

//-----
// Function Name      : void main(void)
// Description        : Main block
// Local Variables    : NONE
// Return Value       : NONE
//-----
void main(void)
{
    //-----
    // Main block
    //-----

    set_bit( STATUS, RP0 );
    OPTION_REG = CFG_OPTION;      //Configure the OPTION register

    set_tris_a( PortAConfig);      // PORT A Setup
    set_tris_b( PortBConfig );     // PORT B Setup

    clear_bit( STATUS, RP0 );
    output_low_port_a( LCD_SEL ); // Disable LCD

    //Configure the INTCON register
    INTCON = CFG_INTCON;
    // Setup the LCD device
    LCD_Setup();
    //disable TMR0
    clear_bit(INTCON,5);
    //disable INT
    clear_bit(INTCON,4);

    //-----
    // Initialization
    //-----
    intc=0;
    indexBuffer=0;
    newByte=0;
    counter3=0;

    LCD_FunctionMode();
    LCD_Write_4_Bit(set_dd_line1);
    LCD_DataMode();
    WriteLCDString( " keyboard      " );

    texto[0] = ' ';
    texto[1] = ' ';
    texto[2] = ' ';
    texto[3] = ' ';
```

Documento Público creado por M.C. Julio Cesar Silva Briano


```

texto[4] = ' ';
texto[5] = ' ';
texto[6] = ' ';
texto[7] = ' ';
texto[8] = ' ';
texto[9] = ' ';
texto[10] = ' ';
texto[11] = ' ';
texto[12] = ' ';
texto[13] = ' ';
texto[14] = ' ';
texto[15] = ' ';

//enable INT
set_bit(INTCON,4);

//-----
//          Main Loop
//-----
for(;;)
{
    //Wait new byte
    if (newByte)
    {
        newByte=0;
        //decode the key
        decode();
        //the second code is the real key
        if (charCounter==2)
        {
            charCounter=0;
            indexBuffer++;
            texto[indexBuffer] = realKey;
            if (indexBuffer==15)
                indexBuffer=0;
        }
    }
    //display chacarter
    LCD_FunctionMode();
    LCD_Write_4_Bit(set_dd_line2); // Goto line 2
    LCD_DataMode();
    WriteVariableString();
}

//-----
// Function Name   : void inteHandler( void )
// Description     : Detect the clock pulse for new bit
// Local Variables : NONE
// Return Value    : NONE
//-----
void inteHandler( void )
{
    pinIn=input_pin_port_a(4);

    intc++;
    switch (intc)
    {
        case START:
            if (pinIn || input_pin_port_b(0))
            {
                intc=0;
            }
            keyIn=0;
            statusParity=0;
            break;
        case PARITY:
    }
}

```

Apéndice A Código del Firmware

```
        statusParity = pinIn;
        break;
    case STOP:
        newByte = 1;
        intc=0;
        break;
    default:
        keyIn = keyIn >> 1;
        if (pinIn)
        {
            keyIn = keyIn | 0x80;
        }
        break;
    }
}

//-----
// Function Name      : void interrupt( void )
// Description         : general interrupt handler
// Local Variables    : NONE
// Return Value       : NONE
//-----
void interrupt( void )
{
    //RB0/INT Interrupt processing
    if( INTCON & 2 )
    {
        clear_bit( INTCON, 1 );
        inteHandler();
    }
}

//-----
// Function Name      : void decode(void)
// Description         : decode stroke key
// Local Variables    : NONE
// Return Value       : NONE
//-----
void decode(void)
{
    char i,indice;

    charCounter++;

    switch (keyIn) //scan code
    {
        case 0xF0: //The up-key identifier
            isUp=1;
            shift=0;
            break;
        case 0x12: //Left SHIFT
            charCounter=1;
            shift=1;
            break;
        case 0x59: //Righ SHIFT
            charCounter=1;
            shift=1;
            break;
        case 0x05: // F1
            shift=0;
            break;
        default :
            if (!shift) //If shift not pressed
            {
                for (i=0;i<=67;i++)
                {
                    if (index[i]==keyIn)
                    {
                        realKey=unshifted[i];
                    }
                }
            }
    }
}
```

Documento Público creado por M.C. Julio Cesar Silva Briano

```

    }
    else
    {
        for (i=0;i<=67;i++)
        {
            if (index[i]==keyIn)
            {
                realKey=shifted[i];
            }
        }
        shift=0;
        break;
    }
}
}

```

8.2 keybled.h Creado por el titular del documento.

```

/*****
/* PORT configuration */
/*****
/*
#define PortAConfig 00010000b // l=input
#define PortBConfig 00000001b // l=input

/*****
/* OPTION REG and INTCON configuration */
/*****
/*
#define CFG_OPTION 10000000b
#define CFG_INTCON 10010000b

//*****
// General
//*****
#define FALSE 0
#define TRUE 1
#define START 1
#define PARITY 10
#define STOP 11

//*****
// Global Variables
//*****
char intc;
char keyIn;
char pinIn;
char statusParity;
char newByte;
char indexBuffer;
char realKey;

char counter3;
char charCounter=0;

//--- For Decode Key
char isUp=0;
char shift=0;
char mode=0;
//-----

//*****
// Function prototypes
//*****
void tmrHandler( void );
void inteHandler( void );
void interrupt( void );
void decode(void);

```

Documento Público creado por M.C. Julio Cesar Silva Briano

8.3 LcdTest.c Creado por J.Winpenny. Con modificaciones J.Silva

```

/*****
/*
/*  LcdTest.c
/*          LCD interface implementation
/*
/*
/*
/*  Version      Date          Comments
/*-----
/*  1.3          19/07/1999    Original release.
/*  1.4          11/12/1999    Added WriteLCDString(const char *)
/*
/*
/*-----
/*
/*
/*          By: J. Winpenny
/*
/*
/*
/*
/*-----
/*          Mode : HD44780 type LCD
/*
/*  Interface :      SEL = Port B bit 3
/*                  WR = Port B bit 2
/*                  RS = Port B bit 1
/*
/*                  Data_7 = Port A bit 3
/*                  Data_6 = Port A bit 2
/*                  Data_5 = Port A bit 1
/*                  Data_4 = Port A bit 0
/*
/*
/*          This version is for 4 bit mode
/*
/*
/*  Add notes: PIC16F84A, 4Mhz, LCD 16x2
/*
/*-----
*****/

#include "LCDTest.h"

void delayM1(void)
{
    delay_ms(1);
}
void delayMS5(void)
{
    delay_ms(5);
}
void delayMS2(void)
{
    delay_ms(2);
}

/*****
/* Setup the lcd device
*****/
void LCD_Setup(void)
{
    /* Reset the LCD */

    /* Power up delay */
    delay_ms(30);
    LCD_FunctionMode();

```

Documento Público creado por M.C. Julio Cesar Silva Briano

```

/* This sequence resets the LCD */
LCD_Write_8_Bit( system_set_4_bit );
delayMS5();

LCD_Write_8_Bit( system_set_4_bit );
delayMS5();

LCD_Write_8_Bit( system_set_4_bit );
delayMS5();

LCD_Write_4_Bit( system_set_4_bit );
delayMS2();

LCD_Write_4_Bit( display_off );
delayMS2();

LCD_Write_4_Bit( entry_mode );
delayMS2();

LCD_Write_4_Bit( display_on );
delayMS2();

LCD_Write_4_Bit( set_dd_ram );
delayMS2();

LCD_DataMode();
}

/*****
/* Put LCD in Function Mode */
*****/
void LCD_FunctionMode(void)
{
    output_low_port_b( LCD_RS );
    delayM1();
}

/*****
/* Put LCD in Data Mode */
*****/
void LCD_DataMode(void)
{
    output_high_port_b( LCD_RS );
    delayM1();
}

/*****
/* Write a single byte to the LCD */
/* 8 Bit Mode */
*****/
void LCD_Write_8_Bit(char d )
{
    /* Write Mode */
    output_low_port_b( LCD_WR );

    delayM1();

    /* Setup data */

    if ( d & 0x80 )
        output_high_port_a( LCD_DATA_7 );
    else output_low_port_a( LCD_DATA_7 );

    if ( d & 0x40 )
        output_high_port_a( LCD_DATA_6 );
    else output_low_port_a( LCD_DATA_6 );
}

```

Apéndice A Código del Firmware

```
if ( d & 0x20 )
    output_high_port_a( LCD_DATA_5 );
else output_low_port_a( LCD_DATA_5 );

if ( d & 0x10 )
    output_high_port_a( LCD_DATA_4 );
else output_low_port_a( LCD_DATA_4 );

delayM1();

/* Select LCD */
output_high_port_b( LCD_SEL );

delayM1();

/* De-select LCD */
output_low_port_b( LCD_SEL );
}

/*****
/* Write a single byte to the LCD */
/* 4 Bit Mode */
*****/
void LCD_Write_4_Bit(char d )
{
    /* Write Mode */
    output_low_port_b( LCD_WR );

    delayM1();

    /* Output Higher 4 bits */
    if ( d & 0x80 )
        output_high_port_a( LCD_DATA_7 );
    else output_low_port_a( LCD_DATA_7 );

    if ( d & 0x40 )
        output_high_port_a( LCD_DATA_6 );
    else output_low_port_a( LCD_DATA_6 );

    if ( d & 0x20 )
        output_high_port_a( LCD_DATA_5 );
    else output_low_port_a( LCD_DATA_5 );

    if ( d & 0x10 )
        output_high_port_a( LCD_DATA_4 );
    else output_low_port_a( LCD_DATA_4 );

    delayM1();

    output_high_port_b( LCD_SEL );
    delayM1();

    /* Clock in the data */
    output_low_port_b( LCD_SEL );
    delayM1();

    /* Output Lower 4 bits */
    d <<= 4;

    if ( d & 0x80 )
        output_high_port_a( LCD_DATA_7 );
    else output_low_port_a( LCD_DATA_7 );

    if ( d & 0x40 )
        output_high_port_a( LCD_DATA_6 );
    else output_low_port_a( LCD_DATA_6 );

    if ( d & 0x20 )
        output_high_port_a( LCD_DATA_5 );
```

Documento Público creado por M.C. Julio Cesar Silva Briano

```

else output_low_port_a( LCD_DATA_5 );

if ( d & 0x10 )
    output_high_port_a( LCD_DATA_4 );
else output_low_port_a( LCD_DATA_4 );

delayM1();

output_high_port_b( LCD_SEL );

/* Clock in the data */
delayM1();

output_low_port_b( LCD_SEL );
}

/*****
/* Write a const string to the LCD */
*****/
void WriteLCDString( const char *lcdptr )
{
    char pi;

    pi = 0;
    // Check for end of string
    while( lcdptr[pi] != 0 )
    {
        // Display on LCD
        LCD_Write_4_Bit( lcdptr[pi++] );
    }
}

/*****
/* Write a const string to the LCD */
*****/
void WriteVariableString( void )
{
    char i;

    for (i=0;i<=15;i++)
    {
        LCD_Write_4_Bit( texto[i] );
    }
}

```

8.4 LcdTest.h Creado por J.Winpenny. Con modificaciones J.Silva

```

/* General definitions */
/*****
/* Definitions for the LCD interface */
*****/
#define LCD_SEL      3    /* Port B bit 3 ( Enables LCD ) */
#define LCD_WR       2    /* Port B bit 2 ( Logic 0 = Write ) */
#define LCD_RS       1    /* Port B bit 1 ( Register select ) */
#define LCD_DATA_4    0    /* LCD BIT 4   Port A bit 0 */
#define LCD_DATA_5    1    /* LCD BIT 5   Port A bit 1 */
#define LCD_DATA_6    2    /* LCD BIT 6   Port A bit 2 */
#define LCD_DATA_7    3    /* LCD BIT 7   Port A bit 3 */

/*****
/* LCD Commands ( Refer to LCD Data Sheet ) */
/* Standard command should work with most common devices */
*****/
/*
#define clear_lcd      0x01 /* Clear Display */
#define return_home    0x02 /* Cursor to Home position */
#define entry_mode     0x06 /* Normal entry mode */

```

Documento Público creado por M.C. Julio Cesar Silva Briano

Apéndice A Código del Firmware

```
#define entry_mode_shift 0x07 /* - with shift */
#define system_set_8_bit 0x38 /* 8 bit data mode 2 line ( 5x7 font ) */
#define system_set_4_bit 0x28 /* 4 bit data mode 2 line ( 5x7 font ) */
#define display_on 0x0c /* Switch ON Display */
#define display_off 0x08 /* Cursor plus blink */
#define set_dd_line1 0x80 /* Line 1 position 1 */
#define set_dd_line2 0xC0 /* Line 2 position 1 */
#define set_dd_ram 0x80 /* Line 1 position 1 */
#define write_data 0x00 /* With RS = 1 */
#define cursor_on 0x0E /* Switch Cursor ON */
#define cursor_off 0x0C /* Switch Cursor OFF */

/*****
/* Global variables */
*****/
char texto[16];

/*****
/* LCD Commands ( Refer to LCD Data Sheet )
/* Standard command should work with most common devices
*****/
void LCD_Setup(void);
void LCD_FunctionMode(void);
void LCD_DataMode(void);
void LCD_Write_8_Bit(char);
void LCD_Write_4_Bit(char);
void WriteLCDString( const char *lcdptr );
```

8.5 scancod.h Creado por el titular del documento.

```
const char index[68]={
0x0D,
0x0E,
0x15,
0x16,
0x1A,
0x1B,
0x1C,
0x1D,
0x1E,
0x21,
0x22,
0x23,
0x24,
0x25,
0x26,
0x29,
0x2A,
0x2B,
0x2C,
0x2D,
0x2E,
0x31,
0x32,
0x33,
0x34,
0x35,
0x36,
0x39,
0x3A,
0x3B,
0x3C,
0x3D,
0x3E,
0x41,
0x42,
0x43,
```

Documento Público creado por M.C. Julio Cesar Silva Briano


```
0x44,  
0x45,  
0x46,  
0x49,  
0x4A,  
0x4B,  
0x4C,  
0x4D,  
0x4E,  
0x52,  
0x54,  
0x55,  
0x5A,  
0x5B,  
0x5D,  
0x61,  
0x66,  
0x69,  
0x6B,  
0x6C,  
0x70,  
0x71,  
0x72,  
0x73,  
0x74,  
0x75,  
0x79,  
0x7A,  
0x7B,  
0x7C,  
0x7D,  
0  
};
```

```
const char unshifted[68]={  
9,  
'|',  
'q',  
'l',  
'z',  
's',  
'a',  
'w',  
'2',  
'c',  
'x',  
'd',  
'e',  
'4',  
'3',  
' ',  
'v',  
'f',  
't',  
'r',  
'5',  
'n',  
'b',  
'h',  
'g',  
'y',  
'6',  
' ',  
'm',  
'j',  
'u',  
'7',
```

Apéndice A Código del Firmware

```
'8',  
'',  
'k',  
'i',  
'o',  
'0',  
'9',  
'.',  
'-',  
'1',  
'0',  
'p',  
'+',  
'',  
'',  
'\\',  
13,  
'..',  
'\\',  
'<',  
8,  
'1',  
'4',  
'7',  
'0',  
'',  
'2',  
'5',  
'6',  
'8',  
'+',  
'3',  
'-',  
'*',  
'9',  
0  
};
```

```
const char shifted[68]={  
9,  
'',  
'Q',  
'!',  
'Z',  
'S',  
'A',  
'W',  
'"',  
'C',  
'X',  
'D',  
'E',  
'',  
'#',  
'',  
'V',  
'F',  
'T',  
'R',  
'%',  
'N',  
'B',  
'H',  
'G',  
'Y',  
'&',  
'L',  
'M',  
'J',  
'U',
```

```
'/',  
'(',  
';',  
'K',  
'I',  
'O',  
'=',  
' )',  
':',  
'',  
'L',  
'0',  
'P',  
'?',  
' ',  
' ',  
' ',  
' ',  
13,  
'^',  
'*',  
'>',  
8,  
'1',  
'4',  
'7',  
'0',  
' ',  
'2',  
'5',  
'6',  
'8',  
'+',  
'3',  
'-',  
'*',  
'9',  
0  
};
```

9 Creditos

Documentación : M.C. Julio Cesar Silva Briano.

País de procedencia: México.

Otras contribuciones: Manejo de LCD en modo 4 bits, gracias al código en C público de J. Wimpenny.

Contactenos : juliosb73@yahoo.com