/\*

\*File: LCD.c

\* Se tomo y se adaptaron las librerias de Ligo George

\* de la pagina www.electrosome.com

\* Pagina: https://electrosome.com/lcd-pic-mplab-xc8/

\*/

#include "LCD.h"

void Lcd\_Port(char a) {

if (a & 1)

D0 = 1;

else

D0 = 0;

if (a & 2)

D1 = 1;

else

D1 = 0;

if (a & 4)

D2 = 1;

else

D2 = 0;

if (a & 8)

D3 = 1;

else

D3 = 0;

if (a & 16)

D4 = 1;

else

D4 = 0;

if (a & 32)

D5 = 1;

else

D5 = 0;

if (a & 64)

D6 = 1;

else

D6 = 0;

if (a & 128)

D7 = 1;

else

D7 = 0;

}

void Lcd\_Cmd(char a) {

RS = 0;

Lcd\_Port (a);

EN = 1;

\_\_delay\_ms(5);

EN = 0;

}

void Lcd\_Clear(void) {

Lcd\_Cmd(0);

Lcd\_Cmd(1);

}

void Lcd\_Init(void) {

Lcd\_Port(0x00);

\_\_delay\_ms(30);

Lcd\_Cmd (0x30);

\_\_delay\_ms(6);

Lcd\_Cmd (0x30);

\_\_delay\_ms(15);

Lcd\_Cmd (0x30);

//////////////////////////////////////////////////////////////////////////

Lcd\_Cmd (0x08);

Lcd\_Cmd (0x01);

Lcd\_Cmd (0x08);

Lcd\_Cmd (0x06);

}

void Lcd\_Set\_Cursor(char a, char b) {

char temp, z, y;

if (a == 1) {

temp = 0x80 + b - 1;

z = temp >> 4;

y = temp & 0x0F;

Lcd\_Cmd(z);

Lcd\_Cmd (y);

}else if (a == 2) {

temp = 0xC0 + b - 1;

z = temp >> 4;

y = temp & 0x0F;

Lcd\_Cmd(z);

Lcd\_Cmd (y);

}

}

void Lcd\_Write\_Char(char a){

char temp;

RS = 1;

Lcd\_Port(temp);

EN = 1;

\_\_delay\_us(40);

EN = 0;

}

void Lcd\_Write\_String(char \*a){

int i;

for (i = 0; a[i] != '\0'; i++)

Lcd\_Write\_Char(a[i]);

}

/\*

\*File: LCD.h

\* Se tomo y se adaptaron las librerias de Ligo George

\* de la pagina www.electrosome.com

\* Pagina: https://electrosome.com/lcd-pic-mplab-xc8/

\*/

#ifndef LCD\_H

#define LCD\_H

#endif

#ifndef \_XTAL\_FREQ

#define \_XTAL\_FREQ 8000000

#endif

#ifndef RS

#define RS PORTAbits.RA0

#endif

#ifndef RW

#define RW PORTAbits.RA1

#endif

#ifndef EN

#define EN PORTAbits.RA2

#endif

#ifndef D0

#define D0 PORTDbits.RD0

#endif

#ifndef D1

#define D1 PORTDbits.RD1

#endif

#ifndef D2

#define D2 PORTDbits.RD2

#endif

#ifndef D3

#define D3 PORTDbits.RD3

#endif

#ifndef D4

#define D4 PORTDbits.RD4

#endif

#ifndef D5

#define D5 PORTDbits.RD5

#endif

#ifndef D6

#define D6 PORTDbits.RD6

#endif

#ifndef D7

#define D7 PORTDbits.RD7

#endif

#include <xc.h>

void Lcd\_Port (char a);

void Lcd\_Cmd (char a);

void Lcd\_Clear (void);

void Lcd\_Set\_Cursor (char a, char b);

void Lcd\_Init (void);

void Lcd\_Write\_Char (char a);

void Lcd\_Write\_String (char \*a);

void Lcd\_Shift\_Right (void);

void Lcd\_Shift\_Left (void);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* File: mainTemplate.c

\* Author: Mariandree Rivera

\* Carnet: 18178

\* Archivo template

\*

\* Created on February 08, 2021,

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Importacion de librerias

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <xc.h>

#include <stdint.h>

#include "LCD.h"

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Palabra de configuracion

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// CONFIG1

#pragma config FOSC = XT // Oscillator Selection bits (XT oscillator: Crystal/resonator on RA6/OSC2/CLKOUT and RA7/OSC1/CLKIN)

#pragma config WDTE = OFF // Watchdog Timer Enable bit (WDT disabled and can be enabled by SWDTEN bit of the WDTCON register)

#pragma config PWRTE = OFF // Power-up Timer Enable bit (PWRT disabled)

#pragma config MCLRE = OFF // RE3/MCLR pin function select bit (RE3/MCLR pin function is digital input, MCLR internally tied to VDD)

#pragma config CP = OFF // Code Protection bit (Program memory code protection is disabled)

#pragma config CPD = OFF // Data Code Protection bit (Data memory code protection is disabled)

#pragma config BOREN = OFF // Brown Out Reset Selection bits (BOR disabled)

#pragma config IESO = OFF // Internal External Switchover bit (Internal/External Switchover mode is disabled)

#pragma config FCMEN = OFF // Fail-Safe Clock Monitor Enabled bit (Fail-Safe Clock Monitor is disabled)

#pragma config LVP = OFF // Low Voltage Programming Enable bit (RB3 pin has digital I/O, HV on MCLR must be used for programming)

// CONFIG2

#pragma config BOR4V = BOR40V // Brown-out Reset Selection bit (Brown-out Reset set to 4.0V)

#pragma config WRT = OFF // Flash Program Memory Self Write Enable bits (Write protection off)

#define \_XTAL\_FREQ 8000000

#define RS PORTAbits.RA0

#define RW PORTAbits.RA1

#define EN PORTAbits.RA2

#define D0 PORTDbits.RD0

#define D1 PORTDbits.RD1

#define D2 PORTDbits.RD2

#define D3 PORTDbits.RD3

#define D4 PORTDbits.RD4

#define D5 PORTDbits.RD5

#define D6 PORTDbits.RD6

#define D7 PORTDbits.RD7

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Variables

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

char counter = 0;

//\*\*\*\*\*\*\*\*\*\*

// Interrupciones

//\*\*\*\*\*\*\*\*\*\*

void \_\_interrupt() ISR()

{

//esto se activara si la interrupcion viene del receptor en el UART

if (PIR1bits.RCIF == 1)

{

//esto nos limpiara la interrupcion

PIR1bits.RCIF = 0;

TXREG = (RCREG + 1);

while (TXSTAbits.TRMT == 0);

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Prototipos de funciones

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void setup(void);

void UART\_Init(void);

void \_\_interrupt() ISR();

void Lcd\_Init(void);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Ciclo principal

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void main(void) {

unsigned int a;

setup();

void Lcd\_Init(void);

UART\_Init();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Loop principal

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

while (1) {

Lcd\_Clear();

// if (PORTCbits.RC7 == 0) {

Lcd\_Set\_Cursor(1, 1);

Lcd\_Write\_String("Hola Mundo");

// }

// if (PORTCbits.RC7 == 0) {

Lcd\_Set\_Cursor(2, 1);

Lcd\_Write\_String("Adios Mundo");

\_\_delay\_ms(2000);

Lcd\_Clear();

// }

// if (PORTCbits.RC7 == 0) {

Lcd\_Set\_Cursor(1, 1);

Lcd\_Write\_String("Developed By");

Lcd\_Set\_Cursor(2, 1);

Lcd\_Write\_String("electroSome");

\_\_delay\_ms(2000);

Lcd\_Clear();

// }

// Lcd\_Set\_Cursor(1, 1);

// Lcd\_Write\_String("www.electroSome.com");

for (a = 0; a < 15; a++) {

\_\_delay\_ms(300);

Lcd\_Shift\_Left();

}

for (a = 0; a < 15; a++) {

\_\_delay\_ms(300);

Lcd\_Shift\_Right();

}

Lcd\_Clear();

Lcd\_Set\_Cursor(2, 1);

Lcd\_Write\_Char('M');

Lcd\_Write\_Char('S');

\_\_delay\_ms(2000);

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Configuracion

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void setup(void) {

ANSEL = 0;

ANSELH = 0b00000001;

TRISA = 0;

PORTA = 0;

TRISB = 0b00000111;

PORTB = 0;

TRISD = 0;

PORTD = 0;

TRISE = 0;

PORTE = 0;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Funciones

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void UART\_Init()

{

//Seleccionara los 8bits para la transmisicion de los datos.

TXSTAbits.TX9 = 0;

//habilitar la transmision

TXSTAbits.TXEN = 1;

//habilitacion del modo asincrono

TXSTAbits.SYNC = 0;

//operacion en velocidad lenta

TXSTAbits.BRGH = 0;

// habilita el puerto serial

RCSTAbits.SPEN = 1;

// habilita que constantemente se reciban datos

RCSTAbits.CREN = 1;

//Baudrate 10417

SPBRG = 11;

//activacion de las interrupciones

INTCONbits.GIE = 1;

// habilitacion de las interrupciones perifericas, ver diagrama.

INTCONbits.PEIE = 1;

//habilita las interrupciones del receptor.

PIE1bits.RCIE = 1;

// limpia el flag de la interrupcion

PIR1bits.RCIF = 0;

}

