Universidad del Valle de Guatemala

Colegio Universitario

**Lab 3 LCD**

Juan González

Carne:17594

Seudocódigo

inicio

Verificar que display este encendido

Verificar bandera de TX , para recibir datos, y enviar datos.

Si display esta encendido , mandar valor de potenciómetro

Verificar valor correcto de los potenciómetros , verificar cual potenciomtro envia valores

Si el valor es recibido enviar un dato adicional proveniente de la compu.

Verificar valores en LCD

**Link de git hub**

<https://github.com/vaelan98/Digital-2/tree/7fbb2ddb9f5a2a6a48e123fcf70fed54c696956d>

/\*

\* File: LCD\_Instructions.h

\* Author:Juan Gonzalez

\*carne:17594

\* Ejemplo de librerias sacados de ejemplos de Pablo Mazariegos

\* Inicializacion de LCD con ayuda de Rodrigo Figueroa

\* //pasar enteros a Decimales sacados de GEEKS

\* Created on 10 de febrero de 2020, 10:03 PM

#pragma config FOSC = INTRC\_NOCLKOUT// Oscillator Selection bits (INTOSCIO oscillator: I/O function on RA6/OSC2/CLKOUT pin, I/O function on 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)

// #pragma config statements should precede project file includes.

// Use project enums instead of #define for ON and OFF.

#define \_XTAL\_FREQ 4000000

#include <xc.h>

#include <stdint.h>

#include "LCD.h"

void Recieve(void){

PIR1bits.RCIF=1;

if(PIR1bits.RCIF== 1){

cur(2,13);

ADCread3();

cur(2,14);

ADCread4();

PIR1bits.RCIF=0;

}

if(PIR1bits.TXIF==1){

ADCON0bits.CHS = 0b1011;

ADCON0bits.GO\_DONE=1;

while(ADCON0bits.GO\_DONE);

TXREG= ADRESH;

PIR1bits.TXIF=0;

}

}

void main(void) {

TRISA= 0;

TRISD = 0;

//TRISCbits.TRISC7 = 1;

PORTA = 0;

PORTD = 0;

PORTC=0;

TRISC=0;

/////Com Serial

TXSTAbits.SYNC = 0;

TXSTAbits.BRGH =1;

BAUDCTLbits.BRG16 = 1;

SPBRG = 25;

SPBRGH = 0;

RCSTAbits.SPEN = 1;

RCSTAbits.RX9 = 0;

RCSTAbits.CREN = 1;

TXSTAbits.TXEN = 1;

PORTD = 0;

///

ADCinit();

lcd\_init();

lcd\_clear();

\_\_delay\_ms(250);

while(1){

cur(1,1);

word("P1");

cur(1,7);

word("P2");

cur(1,13);

word("P3");

cur(2,5);

word("v");

cur(2,11);

word("v");

ADCON0bits.CHS = 0b1101;

cur(2,2);

lee1();

cur(2,1);

lee2();

\_\_delay\_ms(10);

ADCON0bits.CHS = 0b1011;

cur(2,8);

lee1();

cur(2,7);

lee2();

\_\_delay\_ms(10);

Recieve();

\_\_delay\_ms(50);

\_\_delay\_ms(250);

}

}

/\*

\* File: LCD\_Instructions.h

\* Author:Juan Gonzalez

\*carne:17594

\* Ejemplo de librerias sacados de ejemplos de Pablo Mazariegos

\* Inicializacion de LCD con ayuda de Rodrigo Figueroa

\* //pasar enteros a Decimales sacados de GEEKS

\* Created on 10 de febrero de 2020, 10:03 PM

\*/

#ifndef \_\_LCD\_H\_

#define \_\_LCD\_H\_

#define RS PORTDbits.RD0

#define EN PORTDbits.RD1

#include <xc.h> // include processor files - each processor file is guarded.

#include <stdint.h>

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

unsigned char \*guardar[10];

uint8\_t i;

uint8\_t Voldc;

uint8\_t enviar;

double voltaje;

void caracter(uint8\_t info);

void ftoa2(float m, char\* Q, int puntor);

void ADCinit(void){

ADCON1bits.ADFM =0;

ADCON1bits.VCFG0 =0;

ADCON1bits.VCFG1 =0;

ADCON0bits.ADCS0 =1;

ADCON0bits.ADCS1 =0;

ADCON0bits.ADON = 1;

}

void port(char E){

PORTA = E;

}

void com(char E){

RS = 0;

port(E);

EN = 1;

\_\_delay\_us(5);

EN = 0;

}

void lcd\_clear(void){

com(0);

com(1);

}

void cur(char a, char b){

char tiempo;

if(a == 1)

{

tiempo = 0x80 + b - 1;

com(tiempo);

}

else if(a == 2)

{

tiempo = 0xC0 + b - 1;

com(tiempo);

}

\_\_delay\_ms(4);

}

void lcd\_init()

{

\_\_delay\_ms(20);

com(0x30);

\_\_delay\_ms(10);

com(0x30);

\_\_delay\_us(110);

com(0x30);

\_\_delay\_ms(11);

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

com(0x38);

\_\_delay\_ms(4);

com(0x08);

\_\_delay\_ms(4);

com(0x01);

\_\_delay\_ms(4);

com(0x06);

\_\_delay\_ms(4);

com(0x0C);

\_\_delay\_ms(4);

}

void letra(char E)

{

RS = 1; // => RS = 1

port(E); //Data transfer

EN = 1;

\_\_delay\_us(40);

EN = 0;

}

void word(char \*a)

{

int i;

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

letra(a[i]);

}

void Lcd\_Shift\_Right()

{

com(0x01);

com(0x0C);

}

void Lcd\_Shift\_Left()

{

com(0x01);

com(0x08);

}

void lee1(){

\_\_delay\_ms(10);

ADCON0bits.GO\_DONE=1;

while(ADCON0bits.GO\_DONE);

Voldc= ADRESH;

voltaje=Voldc\*5.0/255.0;

ftoa2(voltaje,guardar,2);

word(guardar);

\_\_delay\_ms(50);

}

void lee2(void){

\_\_delay\_ms(10);

ADCON0bits.GO\_DONE=1;

while(ADCON0bits.GO\_DONE);

Voldc= ADRESH;

voltaje=Voldc\*5.0/255.0;

itoa(guardar,voltaje,10);

word(guardar);

\_\_delay\_ms(50);

}

void ADCread3(void){

enviar= RCREG;

voltaje=enviar\*5.0/255.0;

itoa(guardar,voltaje,10);

word(guardar);

}

void ADCread4(){

enviar= RCREG;

voltaje=enviar\*5.0/255.0;

// itoa(buffer,voltaje,10);

ftoa2(voltaje,guardar,2);

word(guardar);

}

void caracter (uint8\_t info){

EN =0;

RS =1;

PORTAbits.RA7 = (info & 0b10000000)>>7;

PORTAbits.RA6 = (info & 0b01000000)>>6;

PORTAbits.RA5 = (info & 0b00100000)>>5;

PORTAbits.RA4 = (info & 0b00010000)>>4;

PORTAbits.RA3 = (info & 0b00001000)>>3;

PORTAbits.RA2 = (info & 0b00000100)>>2;

PORTAbits.RA1 = (info & 0b00000010)>>1;

PORTAbits.RA0 = (info & 0b00000001);

\_\_delay\_ms(10);

EN = 1;

\_\_delay\_ms(5);

EN = 0;

\_\_delay\_ms(5);

\_\_delay\_ms(30);

}

////Tomado de Geeks

void reverse(char\* zpr, int len){

int i = 0, j= len - 1, tiempo;

while (i<j){

tiempo = zpr[ i ];

zpr[i] = zpr[j];

zpr[j] = tiempo ;

i++;

j--;

}

}

int Dtr (int y, char zpr[], int P){

int i = 0;

while(y){

zpr[i++] = (y% 10) + '0';

y =y/10;

}

while(i<P){

zpr[i++] = '0';

reverse(zpr,i);

zpr[i] = '\0';

return i;

}

}

void ftoa2(float m, char\* Q, int puntor){

int coso = (int)m;

float cosof = m - (float)coso;

int i = Dtr(coso, Q, 0);

if(puntor != 0){

Q[i] = '.';

cosof= cosof\* pow(10,puntor);

Dtr((int)cosof, Q +1,puntor);

}

}

#endif /\* LCD\_INSTRUCTIONS\_H \*/