Reporte Laboratorio #6 Microcontroladores

Emanuel Lascurain A01552126
Victor Cavazos A01177055

Procedimientos

Primero conectamos nuestra curiosity junto con nuestro PIC18F45K50, a nuestro display LCD según las indicaciones de la práctica, al igual que nuestro 4x4 keypad, según la practica #5
Para completar la práctica, una parte era usar nuestros conocimientos de la práctica 5 con nuestro 4x4 keypad, lo siguiente era saber usar nuestro display LCD dentro de la programación en MPLAB, esto con la ayuda de comandos dados por la práctica.
LCD_init nos sirve para iniciar nuestro LCD con 3 comandos, mover el cursor a la derecha y tenerlo parpadeando, y borrar todo lo anterior.
LCD_rdy nos sirve para checar el status de nuestro LCD ya que este tarda en desplegar y procesar nuevos comandos o instrucciones para seguir desplegando, por lo cual este comando checa si ya está disponible para seguir con la siguiente instrucción.
LCD_cmd esta instrucción sirve para mandar comandos al LCD.
send2LCD sirve para mandar información deseada al LCD y este la despliegue.
Con el uso de los comandos anteriores entonces construimos la lógica para desplegar la información mandada por nuestro teclado y calcular las operaciones matemáticas básicas de una calculadora.

```
//Practica6


//Librerías
#include <xc.h>
#include <math.h>
#include <stdint.h>
#include "device_config.h"


//
#define LCD_DATA_R          PORTD
#define LCD_DATA_W          LATD
#define LCD_DATA_DIR        TRISD

#define LCD_RW              LATCbits.LATC1
#define LCD_RW_DIR          TRISCbits.TRISC1

#define LCD_E               LATCbits.LATC0
#define LCD_E_DIR           TRISCbits.TRISC0

#define LCD_RS              LATCbits.LATC2
#define LCD_RS_DIR          TRISCbits.TRISC2

//Frecuencia y tiempo de espera.
#define SWEEP_FREQ 50
```

```
#define _XTAL_FREQ 2000000


//analógo, digital
enum por_ACDC {digital, analog};          // digital = 0, analog = 1

//Funciones
char is_an_enter(char enter);
char is_an_ac(char ac);
char is_a_number(char number);
char is_a_sign(char sign);
char key_scanner(void);
void LCD_cmd(char cx);
void send2LCD(char xy);
void LCD_rdy(void);
void portsInit(void);
void LCD_init(void);



//main loop
void main(void){


    portsInit();
    LCD_init();

    char entry;
    char op1;
    char op2;
    char sign;
    char result;

    while(1){

        while(1){
            entry = key_scanner();
            if(is_a_number(entry) == 1){
                LCD_cmd(0x07);      //Ingreso de datos
                op1 = entry;
                LCD_cmd(0x8F);     //posicion del primer operando 0F
                send2LCD(op1 + '0');    //poner el primer operando en la LCD
                break;
            }
        }
        while(1){
            entry = key_scanner();
            if(is_a_sign(entry) == 1){
                sign = entry;
                switch (sign) {     //poner signo en la LCD
                    case 10: send2LCD(43) ;
                    break;
                    case 11: send2LCD(45) ;
                    break;
                    case 12: send2LCD(42) ;
                    break;
                    case 13: send2LCD(246) ;
                    break;
```

```
            }
            break;
        }
        if(is_an_ac(entry) == 1){
            LCD_cmd(0x01);//Eliminar datos en pantalla
            LCD_cmd(0x02);//resetear posiciones
            break;
        }
    }
    while(1){
        entry = key_scanner();
        if(is_a_number(entry) == 1){
            op2 = entry;
            send2LCD(op2 + '0');    //poner segundo operando en la LCD
            break;
        }
        if(is_an_ac(entry) == 1){
            LCD_cmd(0x01);//borrar todo en pantalla
            LCD_cmd(0x02);//resetear posiciones
            break;
        }
    }
    while(1){
        entry = key_scanner();
        if(is_an_enter(entry) == 1){
            LCD_cmd(0xD1);
            LCD_cmd(0x06);      //DESHABILITAR EL SHIFTING
            switch(sign){
                case 10: result = op1 + op2 ;
                break;
                case 11: result = op1 - op2 ;
                break;
                case 12: result = op1 * op2 ;
                break;
                case 13: result = op1 / op2 ;
                break;
            }
            if(result > 9 ){
                send2LCD(result/10 + '0');    //realizar el calculo y mostrar en LCD
                send2LCD(result%10 + '0');    //realizar el calculo y mostrar en LCD
            }
            else{
                send2LCD('0');    //realizar el calculo y mostrar en LCD
                send2LCD(result + '0');    //realizar el calculo y mostrar en LCD
            }

            break;
        }
        if(is_an_ac(entry) == 1){
            LCD_cmd(0x01);//borrar todo en pantalla
            LCD_cmd(0x02);//resetear posiciones
            break;
        }
    }
  }
}

//Funciones
```

```
char is_a_number(char number){
   if(number < 10 ){
      return 1;
   }
   else{
      return 0;
   }
}

char is_a_sign(char sign){
   if(sign >=10 && sign <= 13 ){
      return 1;
   }
   else{
      return 0;
   }
}

char is_an_enter(char enter){
   if(enter == 15 ){
      return 1;
   }
   else{
      return 0;
   }
}

char is_an_ac(char ac){
   if(ac == 14 ){
      return 1;
   }
   else{
      return 0;
   }
}

char key_scanner(void){
   LATAbits.LA0 = 0;
   LATAbits.LA1 = 1;
   LATAbits.LA2 = 1;
   LATAbits.LA3 = 1;
   __delay_ms(SWEEP_FREQ);
   if     (PORTAbits.RA4 == 0) {__delay_ms(SWEEP_FREQ); return 1;}
   else if (PORTAbits.RA5 == 0) {__delay_ms(SWEEP_FREQ); return 2;}
   else if (PORTAbits.RA6 == 0) {__delay_ms(SWEEP_FREQ); return 3;}
   else if (PORTAbits.RA7 == 0) {__delay_ms(SWEEP_FREQ); return 10;}
   LATAbits.LA0 = 1;
   LATAbits.LA1 = 0;
   LATAbits.LA2 = 1;
   LATAbits.LA3 = 1;
   __delay_ms(SWEEP_FREQ);
   if     (PORTAbits.RA4 == 0) {__delay_ms(SWEEP_FREQ); return 4;}
   else if (PORTAbits.RA5 == 0) {__delay_ms(SWEEP_FREQ); return 5;}
   else if (PORTAbits.RA6 == 0) {__delay_ms(SWEEP_FREQ); return 6;}
   else if (PORTAbits.RA7 == 0) {__delay_ms(SWEEP_FREQ); return 11;}
   LATAbits.LA0 = 1;
   LATAbits.LA1 = 1;
   LATAbits.LA2 = 0;
```

```
    LATAbits.LA3 = 1;
    __delay_ms(SWEEP_FREQ);
    if      (PORTAbits.RA4 == 0) {__delay_ms(SWEEP_FREQ); return 7;}
    else if (PORTAbits.RA5 == 0) {__delay_ms(SWEEP_FREQ); return 8;}
    else if (PORTAbits.RA6 == 0) {__delay_ms(SWEEP_FREQ); return 9;}
    else if (PORTAbits.RA7 == 0) {__delay_ms(SWEEP_FREQ); return 12;}
    LATAbits.LA0 = 1;
    LATAbits.LA1 = 1;
    LATAbits.LA2 = 1;
    LATAbits.LA3 = 0;
    __delay_ms(SWEEP_FREQ);
    if      (PORTAbits.RA4 == 0) {__delay_ms(SWEEP_FREQ); return 14;}
    else if (PORTAbits.RA5 == 0) {__delay_ms(SWEEP_FREQ); return 0;}
    else if (PORTAbits.RA6 == 0) {__delay_ms(SWEEP_FREQ); return 15;}
    else if (PORTAbits.RA7 == 0) {__delay_ms(SWEEP_FREQ); return 13;}
    else return 'x';
}


void portsInit(void){
    ANSELA = digital;   // Set port A as Digital for  keypad driving
    TRISA  = 0x0F;      // For Port A, set pins 4  to 7 as outputs (rows), and pins 0 to 3 as inputs (cols)
    ANSELC = digital;   // Set port C as Digital for 7 segment cathode selection (only 4 pins used)
    TRISC  = 0x00;      // For Port C, set pins as outputs for cathode selection
    ANSELD = digital;   // Set port D as Digital for 7 segment anodes
    TRISD  = 0x00;      // for Port D, set all pins as outputs for 7 segment anodes
    OSCCON = 0b01110100;// Set the internal oscillator to 8MHz and stable
}

// LCD initialization function
void LCD_init(void){

    LATC = 0;              // Make sure LCD control port is low
    LCD_E_DIR = 0;         // Set Enable as output
    LCD_RS_DIR = 0;         // Set RS as output
    LCD_RW_DIR = 0;         // Set R/W as output
    LCD_cmd(0x38);         // Display to 2x40
    LCD_cmd(0x0F);         // Display on, cursor on and blinking
    LCD_cmd(0x01);         // Clear display and move cursor home
}

// Send command to the LCD
void LCD_cmd(char cx) {
    LCD_rdy();             // wait until LCD is ready
    LCD_RS = 0;             // select IR register
    LCD_RW = 0;             // set WRITE mode
    LCD_E  = 1;            // set to clock data
    Nop();
    LCD_DATA_W = cx;        // send out command
    Nop();                 // No operation (small delay to lengthen E pulse)
    LCD_E = 0;             // complete external write cycle
}

// Function to display data on the LCD
void send2LCD(char xy){
    LCD_RS = 1;
    LCD_RW = 0;
    LCD_E  = 1;
```

```
    LCD_DATA_W = xy;
    Nop();
    Nop();
    LCD_E  = 0;
}

// Function to wait until the LCD is not busy
void LCD_rdy(void){
    char test;
    // configure LCD data bus for input
    LCD_DATA_DIR = 0b00000000;
    test = 0x80;
    while(test){
        LCD_RS = 0;         // select IR register
        LCD_RW = 1;         // set READ mode
        LCD_E  = 1;         // setup to clock data
        test = LCD_DATA_R;
        Nop();
        LCD_E = 0;          // complete the READ cycle
        test &= 0x80;    //0X80   // check BUSY FLAG
    }
    LCD_DATA_DIR = 0b11111111;
}
```

Resultados

Lamentablemente nuestros resultados no fueron los esperados de nuestro programa, ya que al final solo obtuvimos que nuestro LCD desplegará a la izquierda de la pantalla el cursor parpadeando, esto gracias a nuestro comando LCD_init. Dentro de este proceso es probable que cometieramos algún error de conexión de puertos o de declaración de variable ya que después de esto no logramos que nuestro Display hiciera acción alguna.

Conclusiones

Victor: A pesar de que no obtuvieramos los resultados esperados, si aprendi mucho sobre las conexiones en nuestra curiosity ya que pude trabajar con ella en persona, ademas de que al seguir la práctica tengo una buena idea a la hora de usar comandos para circuitos integrados tales como el display LCD, así como factores importantes que no había tomado en cuenta en otras ocasiones, como checar el status del circuito para poder proceder con la siguiente instrucción, o la función Enable.

Emanuel: Personalmente quedo con la inconformidad de no haber podido correr satisfactoriamente la pantalla LCD, estuve tanto tiempo modificando códigos que encontraba por tutoriales, y viendo ejemplos y librerías, que al menos considero que entendí mucho màs acerca de cómo funciona la Curiosity, es la primera vez que trabajo con ella en persona, y pese a que no pudimos acabar, aprendimos otros factores que solo se pueden ver con la práctica y teniendo la tarjeta en fisico, esperamos para la siguiente práctica poder lograr ejecutar bien el programa.