

ANEXO:

/*

INSTITUTO POLITÉCNICO NACIONAL ESCUELA SUPERIOR DE FÍSICA Y MATEMÁTICAS

//

Sensor digital de gases, alcohol, con MQ-2 y Display LCD de 16x2

MCU: PIC16F886 Osc: Interno a 4 MHz

Lenguaje de Programación: MikroC (dialecto de C orientado a microcontroladores)

Complejidad computacional del algoritmo: $O = n$

Hardware

- PIC16F886
- LCD 16x2 con controlador Hitachi
- Fuente de CC de 5V
- Capacitor electrolítico de 100uF (filtrado de señales)
- Protoboard
- Sensor LM35
- Potenciómetro de 5KOhms

Software

- MikroC IDE
- PicKit2

Alumno: Flores Rodríguez Jaziel David

*/

// Conexiones de la LCD

sbit LCD_RS at RC4_bit;

sbit LCD_EN at RC5_bit;

sbit LCD_D4 at RC0_bit;

sbit LCD_D5 at RC1_bit;

sbit LCD_D6 at RC2_bit;

sbit LCD_D7 at RC3_bit;

//Direccionamiento de los puertos (Entrada, Salida o Alta Impedancia)

sbit LCD_RS_Direction at TRISC4_bit;

sbit LCD_EN_Direction at TRISC5_bit;

sbit LCD_D4_Direction at TRISC0_bit;

sbit LCD_D5_Direction at TRISC1_bit;

sbit LCD_D6_Direction at TRISC2_bit;

sbit LCD_D7_Direction at TRISC3_bit;

//Declaración de variables globales

float temp = 0; // Valor obtenido del Analog/Digital Converter (ADC)

unsigned int datoADC = 0; // Variable que almacena lo obtenido en el ADC

char tempString[15]; // Almacena temperatura como cadena de caracteres

char txt1[] = "GAS: Alcohol";

char txt2[] = "Sensor de gas";

char txt3[] = "Flores Rodriguez Jaziel David.";

char txt4[] = "Hernani Tiago Yee Madeira";

//Declaración de funciones

void printStrings(); //Función tipo void (no regresa ningún dato)

void getTemp();

void main(){

// Configuración de bits y puertos

ANSEL = 0x01; // Establece bit RA0 como analógico

```

ANSELH = 0x00;      // Configura otros bits analógicos como digitales
TRISA0_bit = 1;     // Establece bit RA0 como entrada
C1ON_bit = 0;       // Deshabilita comparador 1
C2ON_bit = 0;       // Deshabilita comparador 2

//Iniciación de funciones externas
ADC_Init();         // Inicializa el Modulo ADC
Delay_us(100);      // Retardo para estabilización del ADC
Lcd_Init();         // Inicializa el LCD

//Llamado a función printStrings()
printStrings();

while(1){
    getTemp();
    Delay_ms(50);
}

void printStrings(){
    int i;
    Lcd_Cmd(_LCD_CLEAR);    // Comando para limpiar el LCD
    Lcd_Cmd(_LCD_CURSOR_OFF); // Comando para quitar el cursor
    Lcd_Out(1,3,"IPN - ESFM");//Imprime texto personalizado
    delay_ms(700);
    Lcd_Cmd(_LCD_CLEAR);
    Lcd_Out(1,2,"Alumno:");
    Lcd_Out(2,1, txt3);
    delay_ms(1000);
    Lcd_Cmd(_LCD_CLEAR);
    Lcd_Out(1,4,"Profesor:");//Imprime texto personalizado
    Lcd_Out(2,1, txt4);
    delay_ms(1000);
    Lcd_Cmd(_LCD_CLEAR);
    Lcd_Out(1,1,txt2);
    delay_ms(400);
    //Bucle de desplazamiento de texto en la LCD (para más de 16 caracteres)
    for(i = 3; i > 0; i--){
        Lcd_Cmd(_LCD_SHIFT_LEFT);
        delay_ms(500);
    }
    Lcd_Cmd(_LCD_CLEAR);
    Lcd_Out(1,1, txt1);    // Se imprime la cadena text LCD
    Lcd_Chr(2,9,223);      // Imprime el caracter "°" con código ASCII 223
    Lcd_Chr(2,14,"ppm");   // Imprime ppm
    Delay_ms(100);
}

void getTemp(){
    datoADC = ADC_Get_Sample(0);    // Obtenemos lectura del AN0
    delay_ms(600);                  // Periodo de actualización de datos
    temp = (float)(datoADC *(5.0/1023.0));
    FloatToStr(temp, tempString); //Se convierte de float a string para la LCD
    Lcd_Out(2,1, tempString);    //Se imprime el resultado
}

```