

Facultad de ingeniería

Materia: Laboratorio de Microcomputadoras

PROYECTO FINAL

Título: Multímetro con interfaz LCD y botones

Integrantes:

- Martínez Pérez Brian Erik - 319049792 - gpo:01
- Nuñez Rodas Abraham Enrique - 114003546 - gpo: 02
- Vicenteño Maldonado Jennifer Michel - 317207251 - gpo:05

Profesor: Moises Melendez Reyes

Grupo: 1

Fecha de Entrega: 15 de mayo de 2025

Semestre: 2025-2



Objetivo: Que el alumno aplique los conocimientos de la teoría y el funcionamiento de los microcontroladores para diseñar e implementar un sistema de despliegue de datos, utilizando lo aprendido en el curso de laboratorio de microcomputadoras, además de realizar investigación; lo que le dará la visión para utilizar cualquier componente en el mercado con un sistema microcontrolador.

Ejercicio

Proyecto: Deberán diseñar, programar y ensamblar un sistema con microcontrolador, para el despliegue de datos mediante un display LCD, se deberá contar con un menú de 4 botones para la selección de la función a realizar, la que desplegará en el display los siguientes datos:

- La temperatura leída mediante un sensor LM35 en grados Celsius, [°C].
- El voltaje medido en un rango desde 0 hasta 20 volts de corriente directa, en volts, [V].
- Un amperímetro que muestre la corriente directa en miliamperes, [mA].

Al iniciar la aplicación mostrará el número de equipo y nombres de cada uno de los integrantes, luego un menú con tres opciones, las cuales se seleccionarán mediante los botones o (se sabe cuál opción se selecciona anteponiendo un * a la opción)

*Temperatura Voltaje

Corriente

Desarrollo:

Diagrama de flujo:

Diagrama de Flujo General

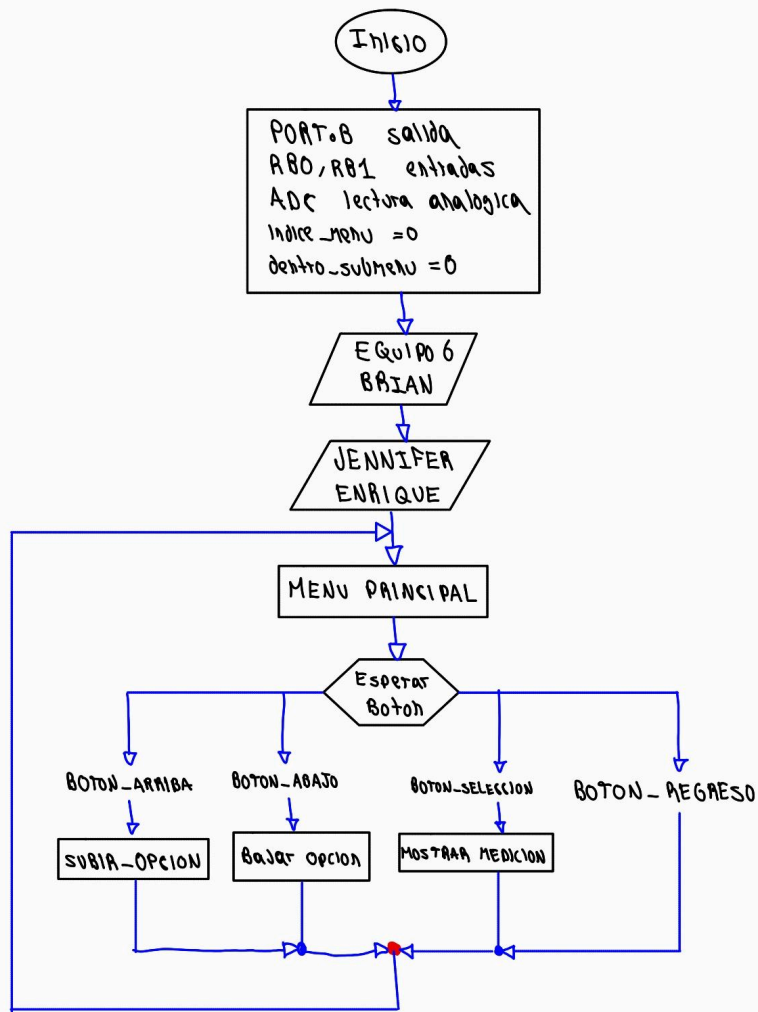
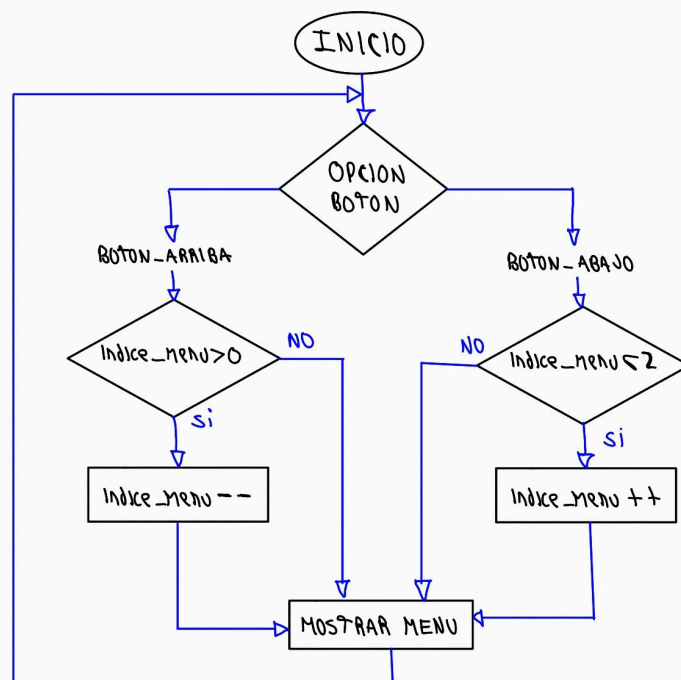
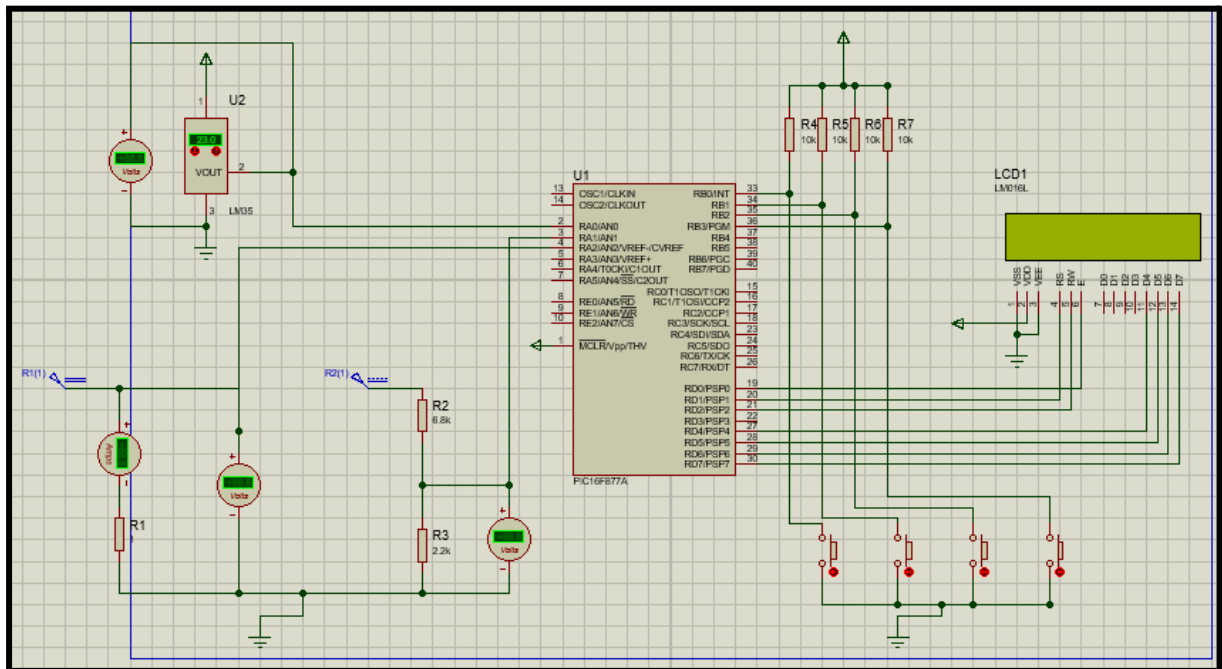


Diagrama Navegación en Menú

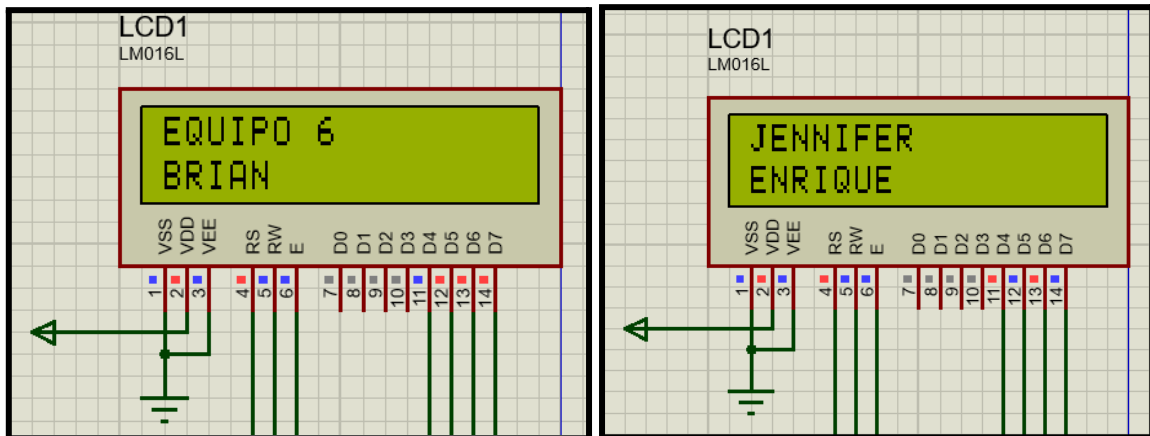


imágenes:

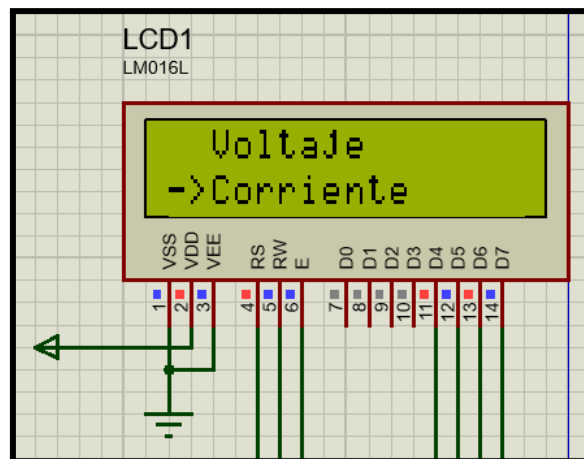
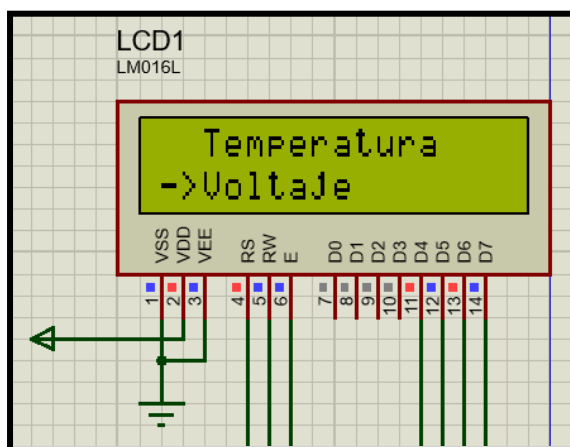
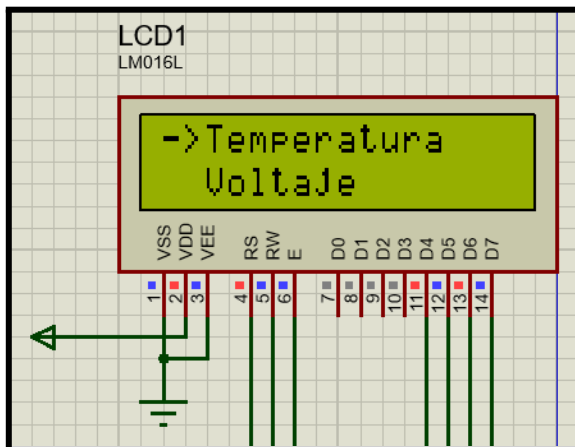
Diagrama electrónico de Proteus:



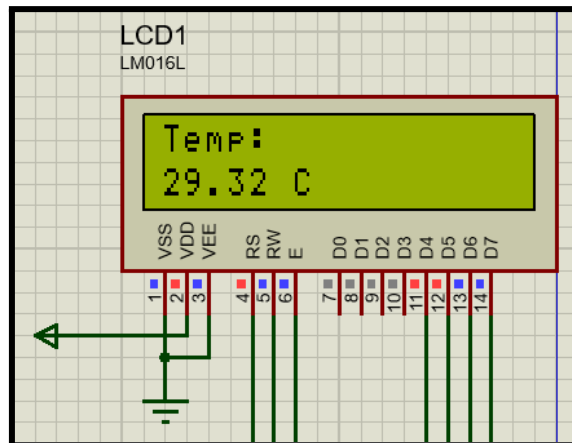
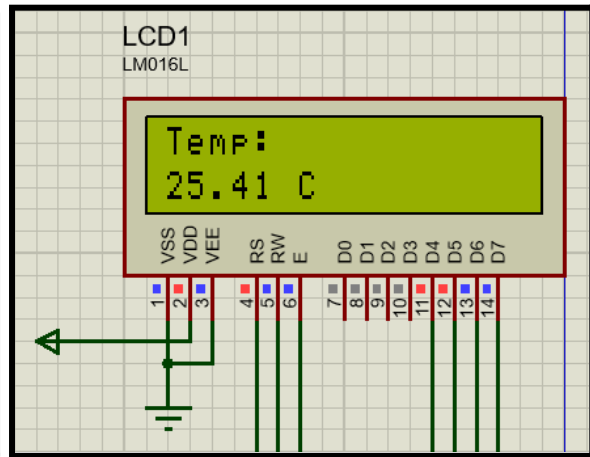
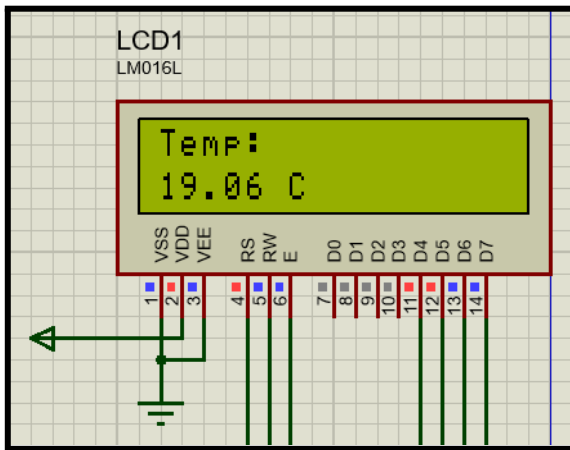
Nombres integrantes:



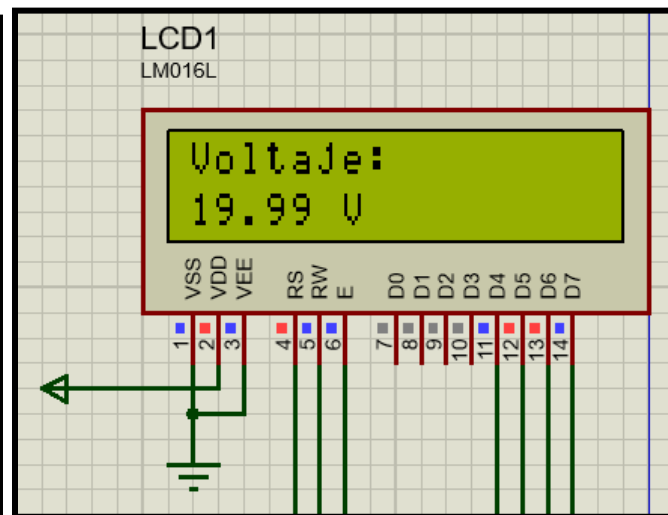
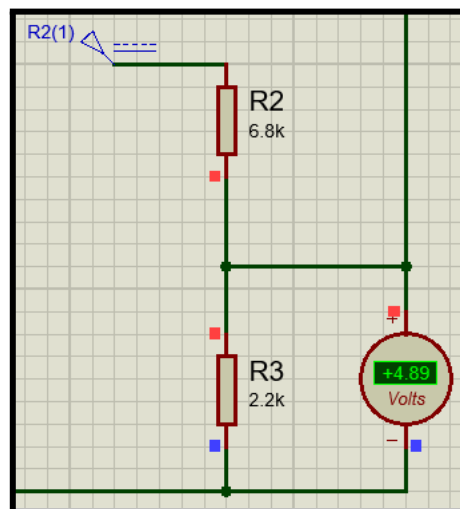
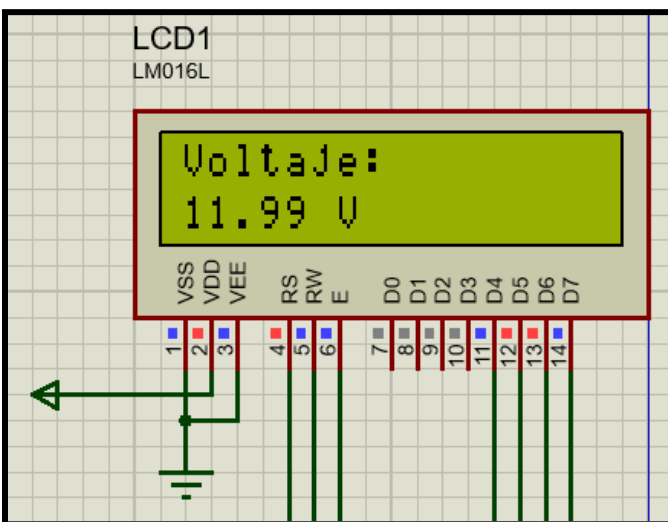
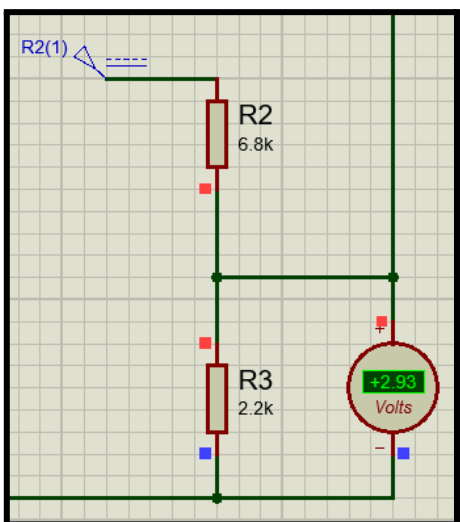
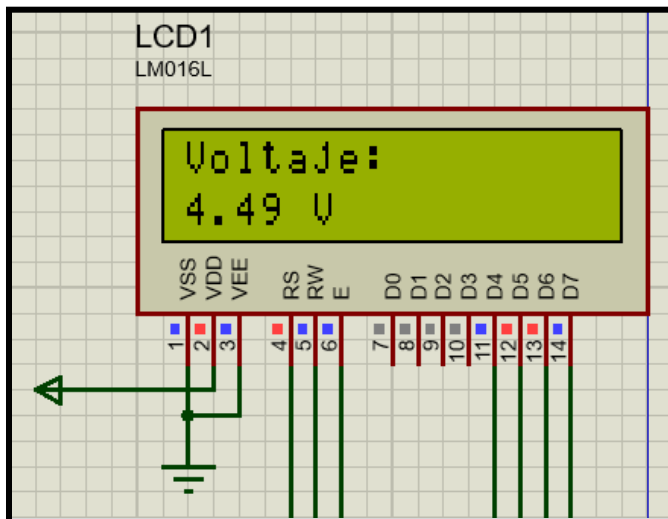
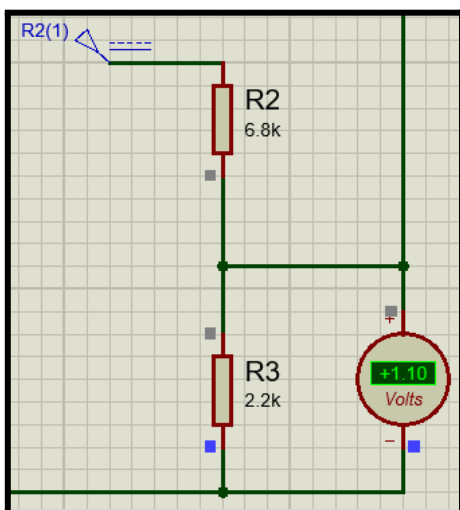
Opciones de menú:



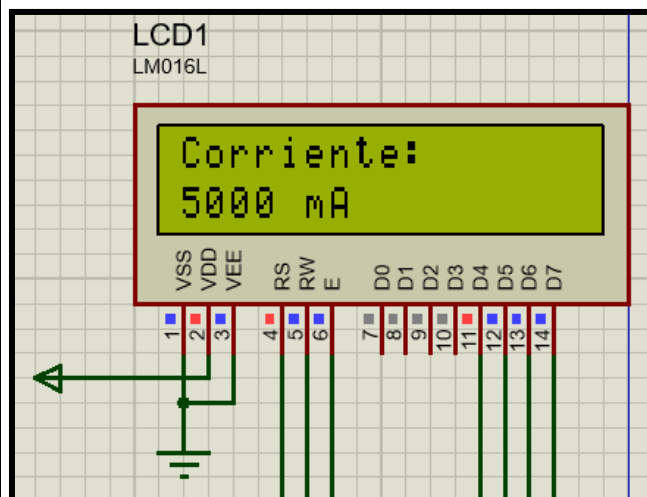
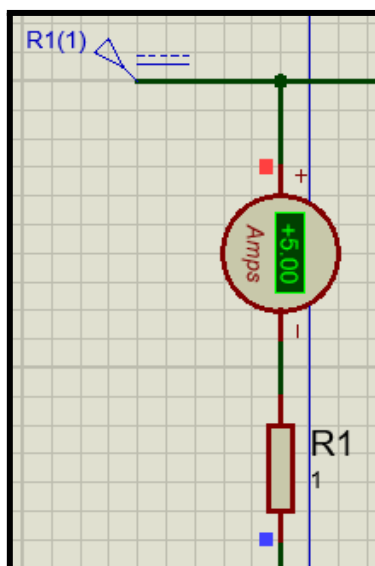
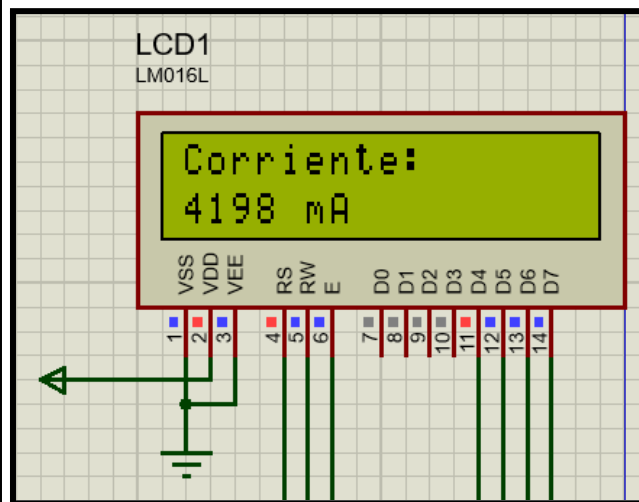
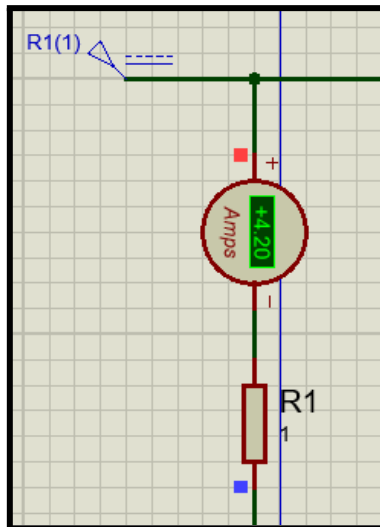
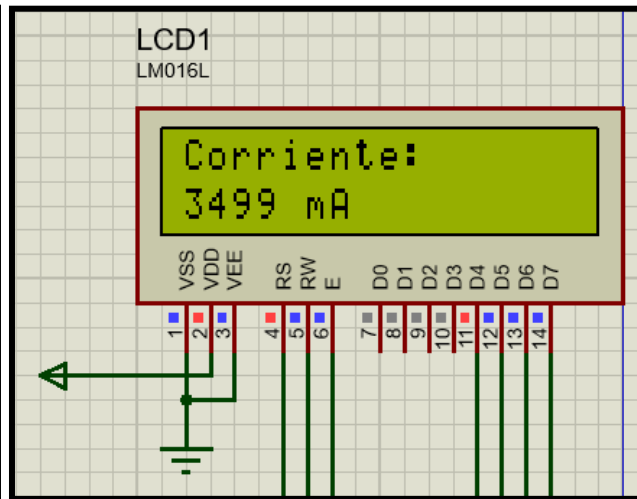
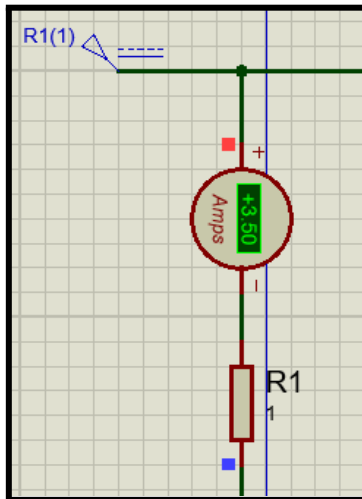
Medición de Temperatura:



Medición de voltaje:



Medición de corriente:



Código:

```
#include <16F877A.h>
#device ADC=10 // Conversión A/D
de 10 bits
#fuses
XT,NOWDT,NOPUT,NOLVP,NOPROTECT, HS
#use delay(clock=20000000) //
Frecuencia de 20 MHz
#use fast_io(B)
#include <lcd.c> // Librería para
control de pantalla LCD
```

```
// Definiciones de pines para botones
del menú
#define BOTON_ARRIBA PIN_B0
// Subir opción
#define BOTON_ABAJO PIN_B1 //
Bajar opción
#define BOTON_SELECCION
PIN_B2 // Entrar/submenu
#define BOTON_REGRESO PIN_B3
// Regresar al menú
```

```
// Variables para el estado del menú
int indice_menu = 0; // Índice de
opción actual del menú
int dentro_submenu = 0; // Indica si
se está visualizando un dato (fuera del
menú principal)
```

```
// Muestra el menú principal con
opciones navegables
void mostrar_menu_principal() {
    lcd_putc("\f"); // Limpia pantalla
completa
    if(indice_menu == 0) {
        lcd_gotoxy(1,1);
        lcd_putc("->Temperatura");
        lcd_gotoxy(1,2);
        lcd_putc(" Voltaje");
    } else if(indice_menu == 1) {
        lcd_gotoxy(1,1);
```

```
        lcd_putc(" Temperatura");
        lcd_gotoxy(1,2);
        lcd_putc("->Voltaje");
    } else if(indice_menu == 2) {
        lcd_gotoxy(1,1);
        lcd_putc(" Voltaje");
        lcd_gotoxy(1,2);
        lcd_putc("->Corriente");
    }
}
```

```
// Lee temperatura del sensor LM35 y
la devuelve en grados Celsius
float obtener_temperatura() {
    int16 valor_adc = read_adc();
    return (valor_adc * 5.0 / 1023.0) *
100.0;
}
```

```
// Lee voltaje usando un divisor
resistivo y lo devuelve en volts
float obtener_voltaje() {
    int16 valor_adc = read_adc();
    return (valor_adc * 5.0 / 1023.0) /
0.2444; // Conversión para rango
hasta 20V
}
```

```
// Lee corriente usando una resistencia
shunt de 1 ohm y devuelve en mA
float obtener_corriente() {
    int16 valor_adc = read_adc();
    return ((valor_adc * 5.0 / 1023.0) / 1)
* 1000.0; // Conversión directa de
voltaje a corriente
}
```

```
// Muestra en pantalla el valor del
sensor correspondiente a la opción
seleccionada
void mostrar_valor_sensor() {
    lcd_putc("\f");
```

```

switch(indice_menu) {
    case 0:
        set_adc_channel(0);
delay_ms(10);

printf(lcd_putc,"Temperatura:\n%.2f C",
obtener_temperatura());
        break;
    case 1:
        set_adc_channel(1);
delay_ms(10);
        printf(lcd_putc,"Voltaje:\n%.2f
[V]", obtener_voltaje());
        break;
    case 2:
        set_adc_channel(2);
delay_ms(20);
        printf(lcd_putc,"Corriente:\n%.0f
[mA]", obtener_corriente());
        break;
}
}

```

```

void main() {
    set_tris_b(0x0F);          // Pines
RB0–RB3 como entradas (botones)
    port_b_pullups(TRUE);      // Activa
resistencias pull-up internas en
PORTB

    lcd_init();                // Inicializa
pantalla LCD

    setup_adc(ADC_CLOCK_INTERNAL);
    setup_adc_ports(ALL_ANALOG);
    lcd_putc("\f");

    // Pantalla de bienvenida con
nombres del equipo
    lcd_gotoxy(1,1);
    lcd_putc("EQUIPO 6");
    lcd_gotoxy(1,2);
    lcd_putc("BRIAN");
    delay_ms(2000);
    lcd_putc("\f");

```

```

    lcd_gotoxy(1,1);
    lcd_putc("JENNIFER");
    lcd_gotoxy(1,2);
    lcd_putc("ENRIQUE");
    delay_ms(2000);
    lcd_putc("\f");

    mostrar_menu_principal(); //
Muestra el menú por primera vez

    while(1) {
        // Navegación hacia arriba en el
menú
        if(!input(BOTON_ARRIBA) &&
!dentro_submenu) {
            if(indice_menu > 0)
                indice_menu--;
            mostrar_menu_principal();
            while(!input(BOTON_ARRIBA));
            // Espera a que se suelte el botón
            delay_ms(100);          //
Anti-rebote
        }

        // Navegación hacia abajo en el
menú
        if(!input(BOTON_ABAJO) &&
!dentro_submenu) {
            if(indice_menu < 2)
                indice_menu++;
            mostrar_menu_principal();
            while(!input(BOTON_ABAJO));
            delay_ms(100);
        }

        // Seleccionar opción para mostrar
sensor
        if(!input(BOTON_SELECCION) &&
!dentro_submenu) {
            dentro_submenu = 1;
            mostrar_valor_sensor();
            delay_ms(300);
        }
    }

```

```

// Volver al menú desde la
visualización del sensor
if(!input(BOTON_REGRESO) &&
dentro_submenu) {
    dentro_submenu = 0;
    lcd_putc("\f");
}
mostrar_menu_principal();
delay_ms(300);
}
}

```

Análisis:

¿Se puede comprobar que la solución propuesta funciona de forma adecuada?

Se puede comprobar que la solución funciona correctamente. El código implementa un sistema de navegación por menú utilizando botones físicos conectados al puerto B del microcontrolador PIC16F877A. Cada opción del menú permite visualizar un valor leído desde un canal analógico del ADC. La lectura del sensor, su conversión y visualización en la pantalla LCD se realizan sin errores, y el sistema responde correctamente a la interacción del usuario mediante los botones.

¿Se alcanzaron los objetivos?

Sí, se alcanzaron los objetivos del desarrollo. El sistema logra mostrar una interfaz básica en una pantalla LCD con un menú de tres opciones (Temperatura, Voltaje, Corriente). Cada opción activa el canal ADC correspondiente, realiza la conversión de la señal analógica a digital y transforma ese valor en una magnitud física entendible, como °C, volts o miliamperios. También se implementa una pantalla de bienvenida y una navegación intuitiva por menú.

¿Cuál es el flujo interno de los datos en el microcontrolador?

El flujo interno comienza con la detección del estado de los botones de entrada (RB0–RB3). Según el botón presionado, se actualiza una variable de estado (indice_menu) que determina qué opción está activa. Si se selecciona una opción, el sistema activa el canal correspondiente del ADC (por ejemplo, set_adc_channel(0) para temperatura), espera un tiempo de muestreo (delay_ms(10)), y luego realiza la lectura con read_adc(). Este valor digital (de 10 bits, entre 0 y 1023) se convierte a una magnitud física usando una fórmula en punto flotante. Finalmente, el resultado se muestra en el LCD usando funciones como lcd_putc y printf.

¿Cuáles fueron los modos de direccionamiento utilizados?

Modo directo: se accede directamente a registros o puertos como set_tris_b(0x0F) y input(BOTON_ARRIBA).

Modo inmediato: se usan constantes en instrucciones, como `delay_ms(300)`, `0x0F`, `1023.0`, `1000.0` y demás valores literales dentro del código. Además, aunque no se usa explícitamente, en operaciones como `indice_menu++` o `indice_menu > 0`, se manejan registros a través de variables en memoria RAM interna.

¿Qué periféricos se usaron?

Conversor Analógico-Digital (ADC): utilizado para leer los valores analógicos provenientes de sensores en los canales AN0, AN1 y AN2.

Puerto B (RB0–RB3): configurado como entrada digital para leer los botones del menú.

Pantalla LCD: utilizada para mostrar tanto el menú como los valores medidos por los sensores.

Resistencias pull-up internas: activadas en `port_b_pullups(TRUE)` para estabilizar la lectura de los botones en las entradas digitales

Conclusiones:

Martínez Pérez Brian Erik

En este proyecto fue posible aplicar de manera efectiva los conocimientos adquiridos en el curso de laboratorio de microcomputadoras, integrando teoría y práctica en el diseño de un sistema que contenga la interacción con periféricos como el ADC, la pantalla LCD y entradas digitales, consolidando habilidades clave como la lectura de sensores, el procesamiento de señales analógicas y la creación de interfaces básicas de usuario. Además, el proceso implicó la búsqueda e interpretación de información técnica externa, lo cual fortaleció la capacidad de investigación y resolución autónoma de problemas.

Núñez Rodas Abraham Enrique

En este proyecto logré integrar de forma práctica los conceptos teóricos de microcontroladores y electrónica analógica para diseñar un multímetro digital completamente funcional. Al programar en CCS-C aprendí a configurar correctamente el ADC de 10 bits, a aplicar fórmulas de conversión para sensores (LM35, divisor resistivo y shunt de $1\ \Omega$) y a gestionar menús por medio de botones con anti-rebote. La implementación del bucle de submenú “en vivo” me permitió profundizar en técnicas de muestreo periódico y refresco dinámico de la pantalla LCD. Además, la simulación en Proteus me enseñó a validar conexiones, ajustar parámetros de componentes y depurar fallas de forma eficiente. Finalmente, documentar cada etapa —desde el diagrama de flujo hasta el análisis de

resultados— reforzó mi capacidad de investigación y comunicación técnica, dejándome con la confianza de poder adaptar este sistema a otros sensores o microcontroladores en futuros desarrollos.

Vicenteño Maldonado Jennifer Michel

Con ayuda del proyecto pude observar una aplicación práctica más completa en comparación a las vistas durante las prácticas, realizarlo con el simulador representó un reto debido a la compatibilidad de las versiones, afortunadamente programar en ensamblador algunas de las funciones que ocupamos para este proyecto como el manejo del LCD, sensor de temperatura e incluso la creación de diagramas de flujo nos permitió entender el funcionamiento del programa y a desarrollar más fácilmente la programación en CCS-C.

El proyecto permite realizar la navegación a través del uso de botones conectados al puerto B, en un menú principal y uno secundario, cada selecciona activa un canal ADC, que convierte las señales análogas leídas en digitales, al actualizar la variable de lectura constantemente logramos que pudiera medir la temperatura en tiempo real.

Referencias:

- del PIC, 2. 1-La Familia. (s/f). 2.- Descripción General del PIC16F877. Edu.ar., https://exa.unne.edu.ar/ingenieria/sistemas/public_html/Archi_pdf/HojaDatos/Microcontroladores/PIC16F877.pdf
- (S/f). Newark.com. Recuperado de <https://mexico.newark.com/microchip/pic16f877a-i-p/microcontroller-mcu-8-bit-pic16/dp/69K7640?srsId=AfmBOorWLTceQMTppGk0OMGmmjB6Upliw55U28F2qBZH2pUYET3EInu>
- *Descripción General del PIC16F877 1 2.-Descripción General del PIC16F877 2.1.-La Familia del PIC16F877.* (n.d.). https://exa.unne.edu.ar/ingenieria/sistemas/public_html/Archi_pdf/HojaDatos/Microcontroladores/PIC16F877.pdf
- *Manual de usuario Microchip PIC16F877A (280 páginas).* (2025). Manual.cr. https://www.manual.cr/microchip/pic16f877a/manual?utm_
- *PIC16F87XA Devices Included in this Data Sheet: High-Performance RISC CPU.* (n.d.). <https://ww1.microchip.com/downloads/en/DeviceDoc/39582C.pdf>

- *puertos-de-entradasalida* - *MIKROE*. (2024). MIKROE.
<https://www.mikroe.com/ebooks/microcontroladores-pic-programacion-en-c-con-ejemplos/puertos-de-entradasalida>