UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

División de Ingeniería Eléctrica

LABORATORIO DE MICROCOMPUTADORAS

Reporte Proyecto Final

Multímetro con interfaz LCD y botones

Profesor: Ing. Moisés Meléndez Reyes

Grupo: 1

Equipo: 5

Alumnos (Gpo. teoría):

- Cruz Hernández, Alejandro Marcelo (2)
- García Sánchez, Emilio (2)
- Ordiales Caballero, Iñaky (3)
- Rojas Eng, Aurelio (2)

Semestre: 2023 - 1

Fecha de entrega: 5 de enero de 2023.

1. 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.

2. Desarrollo del proyecto final

Link para la simulación y video:

https://drive.google.com/drive/folders/1Qnfxk0nN9gUOS7ZhX10Om2S9nUHF3BXy?usp=sharing

Para el proyecto final del laboratorio de microcomputadoras se nos asignó la tarea de crear un lector de temperatura, voltaje y corriente. Para esto se deberían usar los convertidores analógico-digital que ofrece el PIC 16, además de los periféricos para la interacción con el usuario. El usuario podrá seleccionar a través de un menú las diferentes opciones, recorriendo de arriba a abajo con dos botones y seleccionándolo o regresando con otros dos. Además se debe de mostrar en todo momento ya sea el menú o las lecturas realizadas en un display LCD de 16x2.

Para poder realizar esto se nos dieron los diagramas de conexiones y las especificaciones de los componentes, por lo que nuestro trabajo fue enfocarnos en cómo programarlos y utilizarlos adecuadamente. Un requerimiento específico era que los botones deberían tener un efecto instantáneo (Interrupciones) y que se debe utilizar el sensor de temperatura LM35 para tomar las medidas, por lo que se tuvo que revisar su datasheet. Basándonos en el siguiente esquema, este documento presenta la solución propuesta y simulada para el proyecto final.

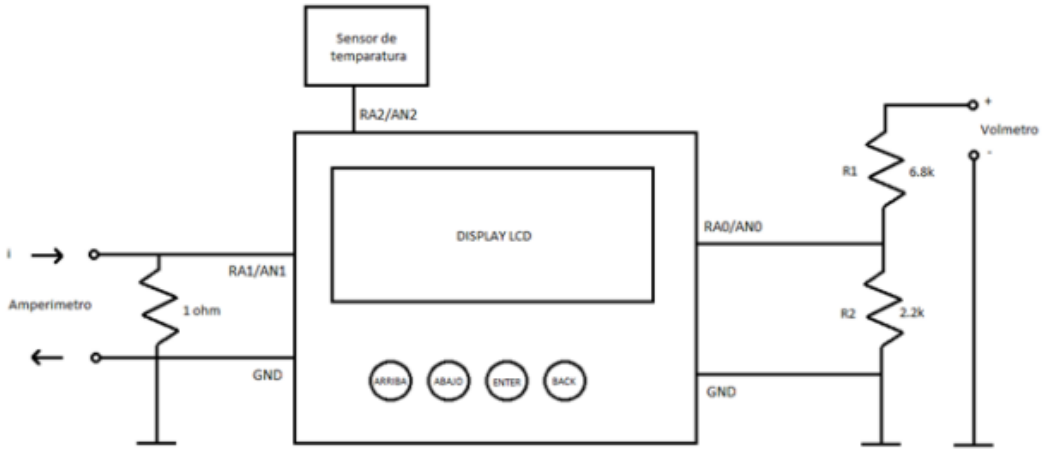
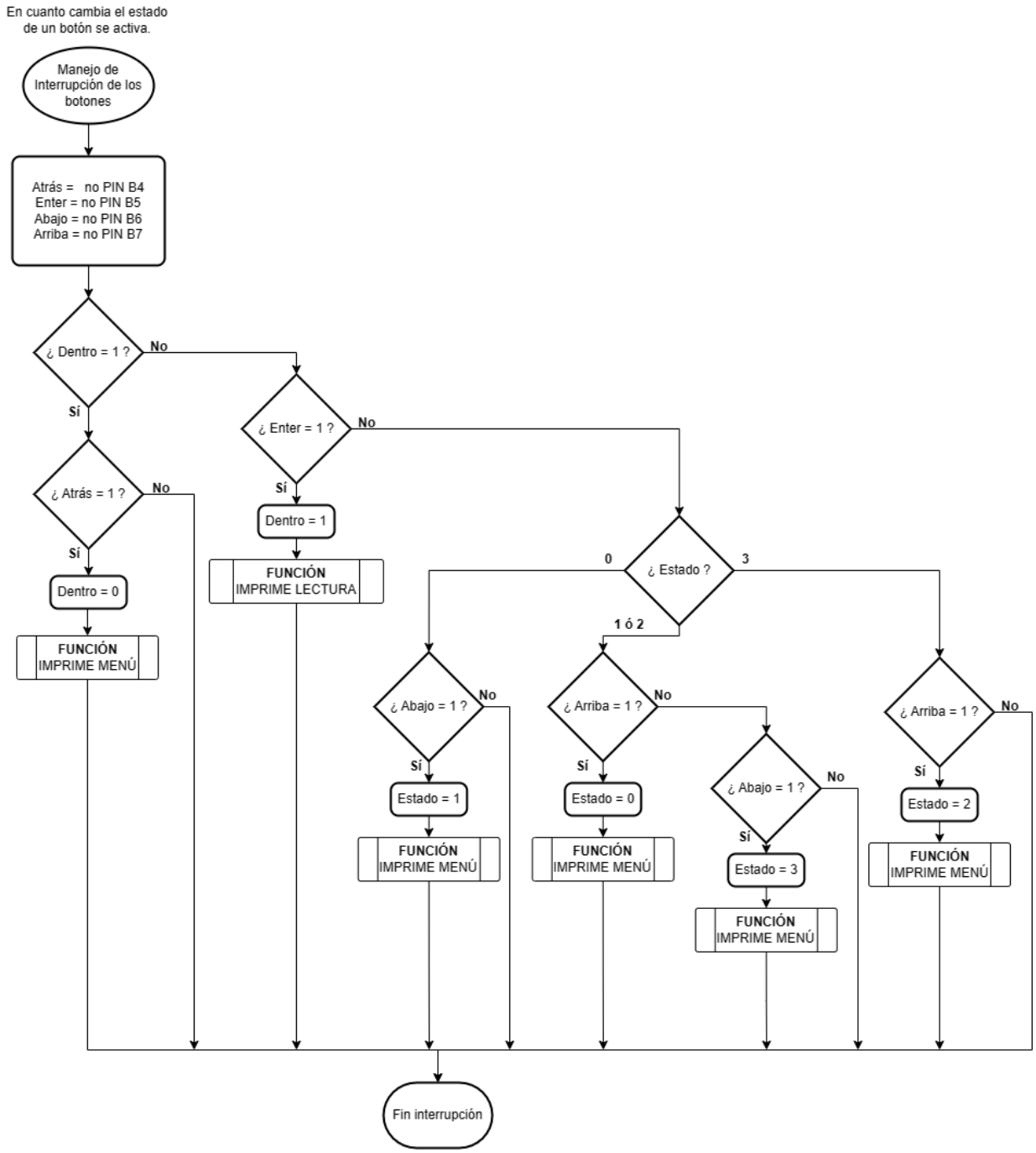
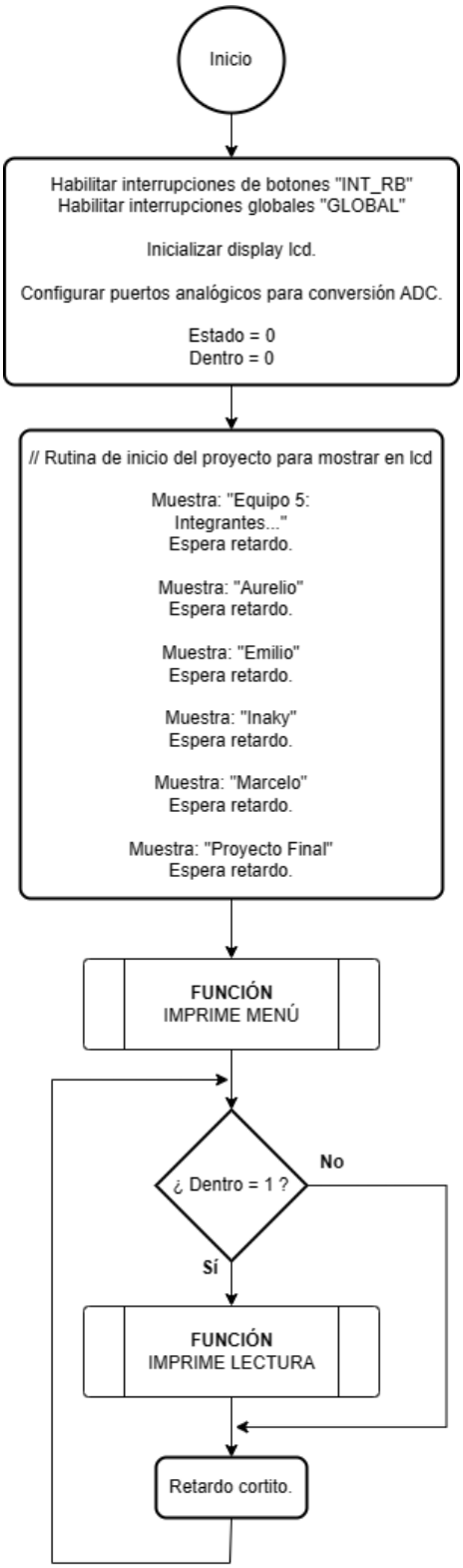


Diagrama de flujo:

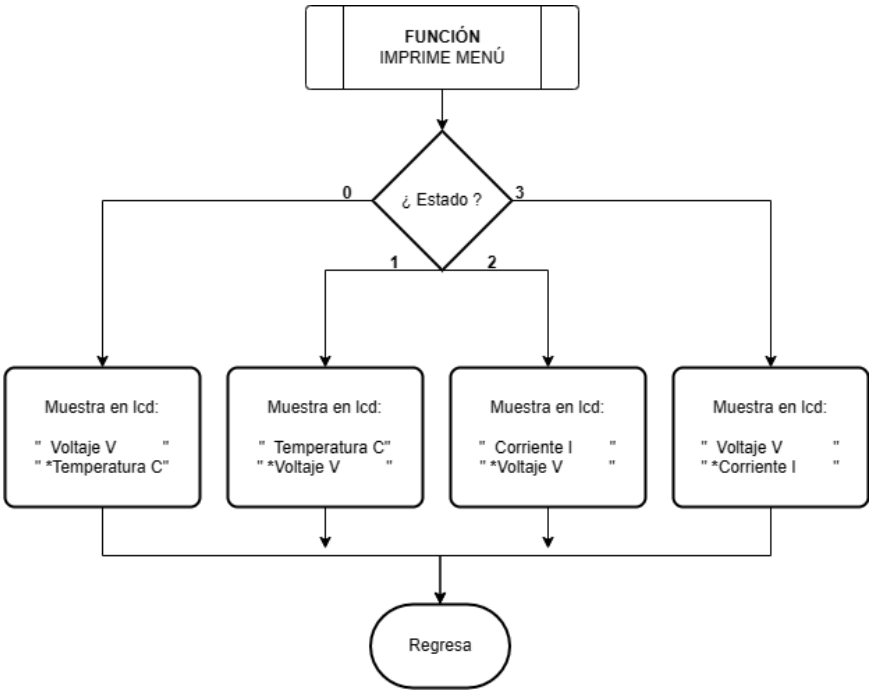
Manejo de interrupciones.



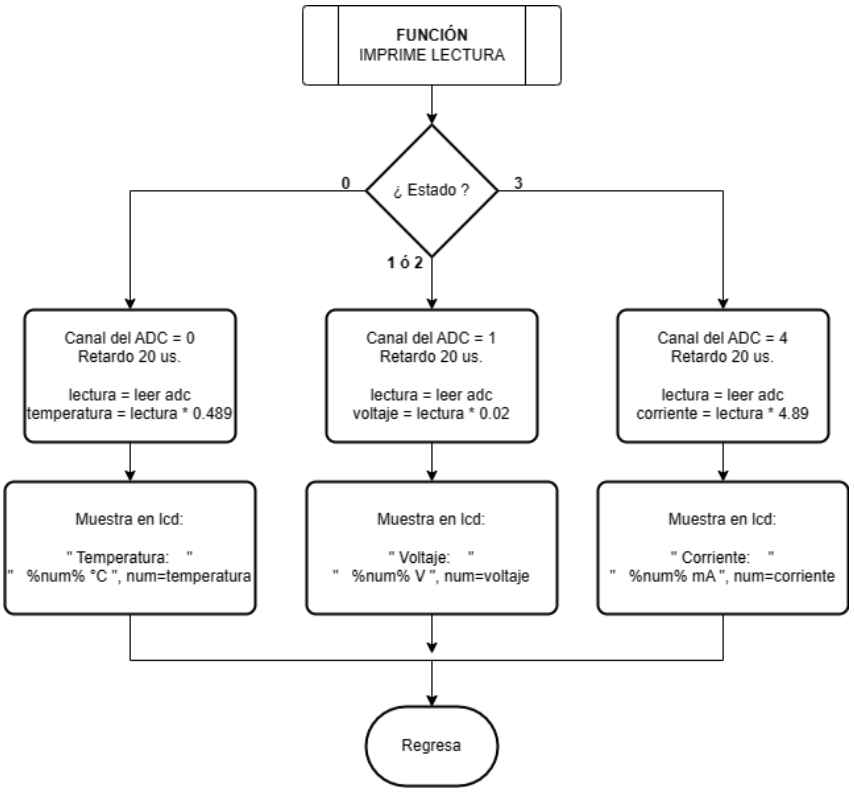
Main.


Inicio
Habilitar interrupciones de botones "INT_RB"
Habilitar interrupciones globales "GLOBAL"
Inicializar display lcd.
Configurar puertos analógicos para conversión ADC.
Estado = 0
Dentro = 0
// Rutina de inicio del proyecto para mostrar en lcd
Muestra: "Equipo 5:
Integrantes..."
Espera retardo.
Muestra: "Aurelio"
Espera retardo.
Muestra: "Emilio"
Espera retardo.
Muestra: "Inaky"
Espera retardo.
Muestra: "Marcelo"
Espera retardo.
Muestra: "Proyecto Final"
Espera retardo.
FUNCIÓN
IMPRIME MENÚ
¿ Dentro = 1 ?
No
Sí
FUNCIÓN
IMPRIME LECTURA
Retardo cortito.

Función menú.



Función lectura.



Código en C: (los retardos son cortos porque la simulación va muy lenta).

```
#include <16F877A.h>            // biblioteca del micro
#device ADC=10                  // establece el convertidor A/D en 10 bits resolución
#fuses HS,NOWDT,NOPROTECT,NOLVP // parametros físicos - eléctricos del controlador
#use delay(clock=20000000)      // 20 MHz establece el reloj a utilizar
// utiliza estándar rs232 con la configuración:
//   transmite por portC.6 y recibe por portC.7 con 9600 bauds
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)
#define use_portd_lcd true      // establece el PORTD como conexión del display LCD
#include <lcd.c>                //biblioteca para control de display LCD.
#org 0x1F00, 0x1FFF void loader16F877(void) {} // reserva la memoria del bootloader


float temperatura;   // variable para almacenar la lectura de temperatura
float voltaje;       // variable para almacenar la lectura de voltaje
float corriente;     // variable para almacenar la lectura de corriente
INT16 lectura;       // variable para guardar temporalmente lectura digital
int dentro;          // indica si se esta en el menú o dentro de medición
int estado;          // estado en el que está el menú
/*
Edo 0 --> *Temperatura
           Voltaje

Edo 1 -->  Temperatura
          *Voltaje

Edo 2 --> *Voltaje
           Corriente

Edo 3 -->  Voltaje
          *Corriente
*/

// Función para iniciar el LCD y presentar el proyecto
void inicioLCD() {
   lcd_gotoxy(2,1);  // posiciona cursor en (2,1)
   printf(lcd_putc," Equipo 5:\n"); //escribe los chars en el LCD
   lcd_gotoxy(2,2);
   printf(lcd_putc," Integrantes...");
   delay_ms(50);     // espera 50 ms
   lcd_gotoxy(1,2);
   printf(lcd_putc,"   Aurelio      ");
   delay_ms(50);
   lcd_gotoxy(1,2);
   printf(lcd_putc,"   Emilio       ");
   delay_ms(50);
   lcd_gotoxy(1,2);
   printf(lcd_putc,"   Inaky        ");
   delay_ms(50);
   lcd_gotoxy(1,2);
   printf(lcd_putc,"   Marcelo      ");
   delay_ms(50);
   lcd_gotoxy(1,1);
   printf(lcd_putc,"Proyecto final  ");
```

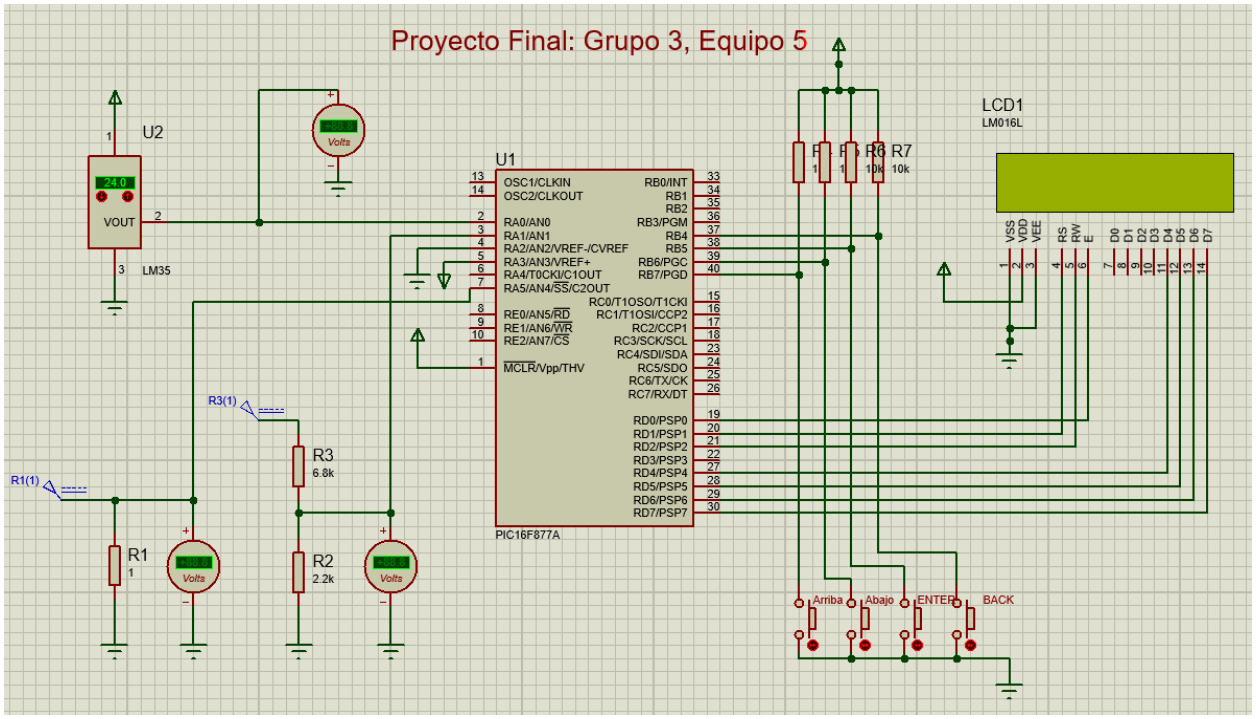
```
   lcd_gotoxy(1,2);
   printf(lcd_putc,"  :)            ");
   delay_ms(50);
}


// función para imprimir el menú según el estado actual
void imprimeMenu() {
   switch (estado){
      case 0:
         lcd_gotoxy(1,2);
         printf(lcd_putc,"   Voltaje V    ");
         lcd_gotoxy(1,1);
         printf(lcd_putc," * Temperatura C");
         break;
      case 1:
         lcd_gotoxy(1,1);
         printf(lcd_putc,"   Temperatura C");
         lcd_gotoxy(1,2);
         printf(lcd_putc," * Voltaje V    ");
         break;
      case 2:
         lcd_gotoxy(1,2);
         printf(lcd_putc,"   Corriente I  ");
         lcd_gotoxy(1,1);
         printf(lcd_putc," * Voltaje V    ");
         break;
      case 3:
         lcd_gotoxy(1,1);
         printf(lcd_putc,"   Voltaje V    ");
         lcd_gotoxy(1,2);
         printf(lcd_putc," * Corriente I  ");
         break;
      default:
         break;
   }
}


// función para imprimir la lectura según el estado actual.
void imprimeLectura() {
   lcd_gotoxy(1,1);
   switch (estado){
      case 0:
         set_adc_channel(0);     // Configura el canal 0 para usar
         delay_us(20);           // Retardo de 20 us
         lectura = read_adc();   // obtiene el resultado de A/D
         temperatura = lectura * 0.489; // ecuación lineal temper. °C
         printf(lcd_putc," Temperatura:   ");
         lcd_gotoxy(1,2);
         printf(lcd_putc,"    %04.1f  C     ", temperatura);
         break;
      case 1:
      case 2:
         set_adc_channel(1);     // Configura el canal 1 para usar
```

```
         delay_us(20);           // Retardo de 20 us
         lectura = read_adc();   // obtiene el resultado de A/D
         voltaje = lectura * 0.02;  // ecuación lineal voltaje V
         printf(lcd_putc," Voltaje:       ");
         lcd_gotoxy(1,2);
         printf(lcd_putc,"    %04.1f V      ", voltaje);
         break;
      case 3:
         set_adc_channel(4);     // Configura el canal 2 para usar
         delay_us(20);           // Retardo de 20 us
         lectura = read_adc();   // obtiene el resultado de A/D
         corriente = lectura * 4.89;   // ecuación lineal corriente mA
         printf(lcd_putc," Corriente:     ");
         lcd_gotoxy(1,2);
         printf(lcd_putc,"    %05.1f mA    ", corriente);
         break;
      default:
         break;
   }
}


// Control de interrupciones de los botones
#INT_RB
void port_rb(){
   // código de rutina de interrupción.

   int back  = !input_state(PIN_B4);   // estado botón back
   int enter = !input_state(PIN_B5);   // estado botón enter
   int down  = !input_state(PIN_B6);   // estado botón abajo
   int up    = !input_state(PIN_B7);   // estado botón arriba

   if (dentro) {  // checa si se está en lectura y sale a menú
      if (back) {
         dentro = 0;
         imprimeMenu();
      }
      return;
   }

   if (enter) {   // checa si está en menú y entra a lectura
      dentro = 1;
      imprimeLectura();
      return;
   }

   // cambia el estado según el botón (arriba o abajo) y imprime
   // el nuevo menú sólo cuando es necesario. Se usó una tabla
   // de transiciones para saber cuando se cambiaba de edo.
   switch (estado) {
      case 0:
         if (down) {
            estado = 1;
            imprimeMenu();
         }
```

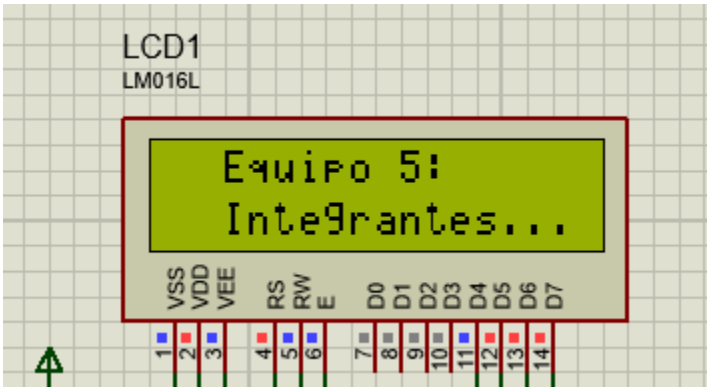
```
         break;
      case 1:
      case 2:
         if (up) {
            estado = 0;
            imprimeMenu();
         }else if (down) {
            estado = 3;
            imprimeMenu();
         }
         break;
      case 3:
         if (up) {
            estado = 2;
            imprimeMenu();
         }
         break;
      default:
         break;
   } // end switch
   return;
}


// Función principal para el flujo del programa
int main() {

   //inicialización componentes
   enable_interrupts(INT_RB);    // habilita interrupciones de botones
   enable_interrupts(GLOBAL);    // habilita interrupciones globales
   lcd_init();                   // inicialización del display
   setup_port_a(ALL_ANALOG);     // Define el puerto A como analógico
   setup_adc(ADC_CLOCK_INTERNAL);// Define frecuencia de muestreo A/D
   // inicialización de variables
   estado = 0;
   dentro = 0;
   lectura = 0;
   temperatura = 0;
   voltaje = 0;
   corriente = 0;

   inicioLCD();   // inicia el LCD y presenta proyecto
   imprimeMenu(); // imprime el menú en su estado inicial

   // entra a un ciclo infinito donde si está dentro(lectura)
   // actualiza el valor que se muestra en el display cada 10ms
   while(true) {

      if (dentro){
         imprimeLectura();
      }
      delay_ms(10);
   } // end while
   return 0;
} //end main
```

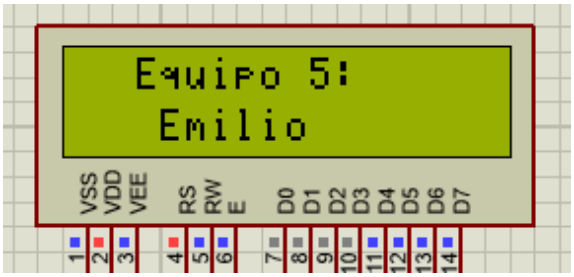
Evidencias:

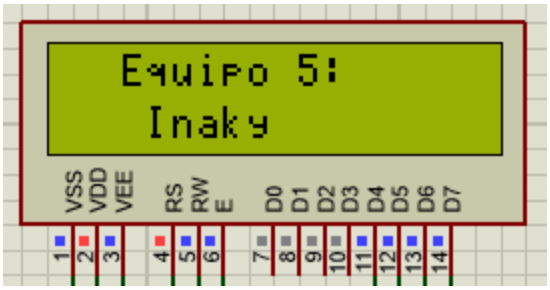
Circuito conectado.

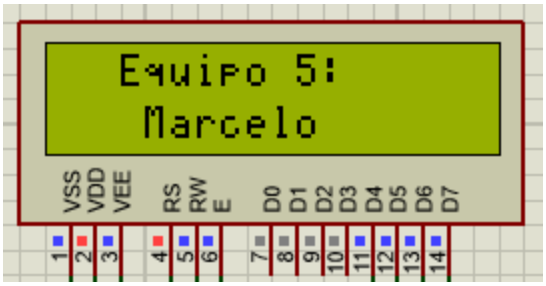


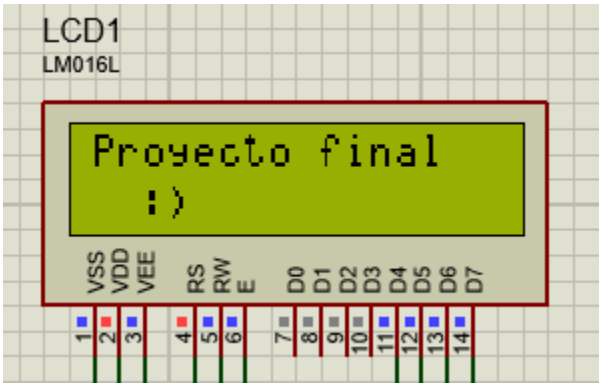
Mensajes iniciales lcd.



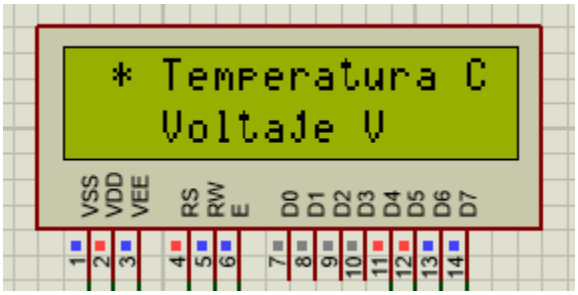




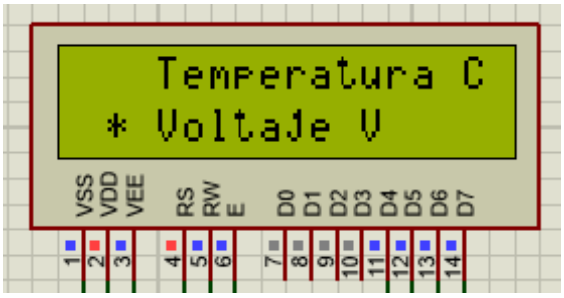




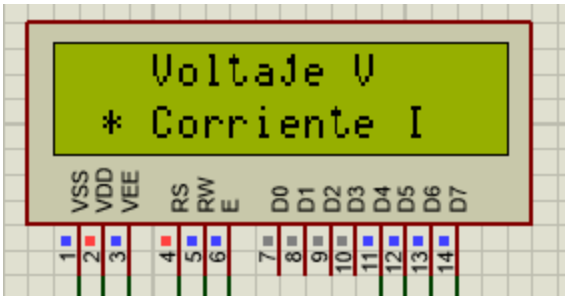
Diferentes vistas del menú.



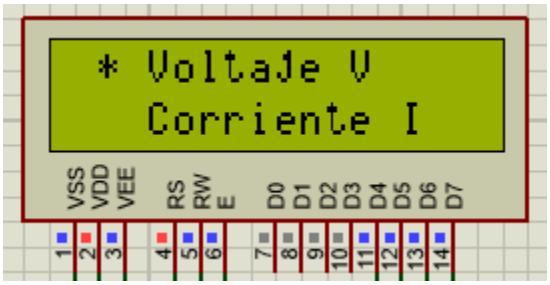
***Le picamos el botón abajo.**



***Le picamos el botón abajo.**

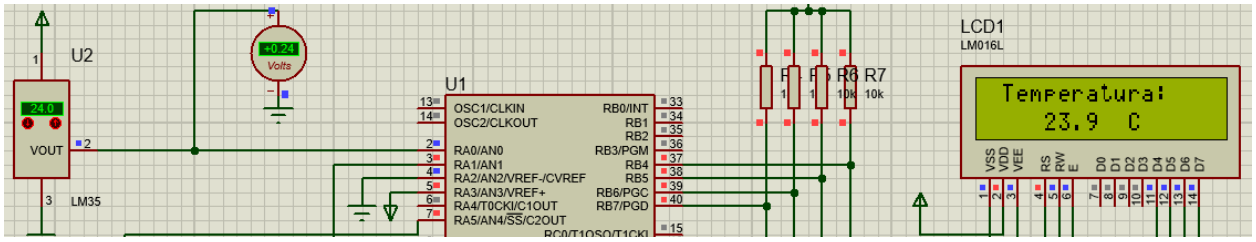


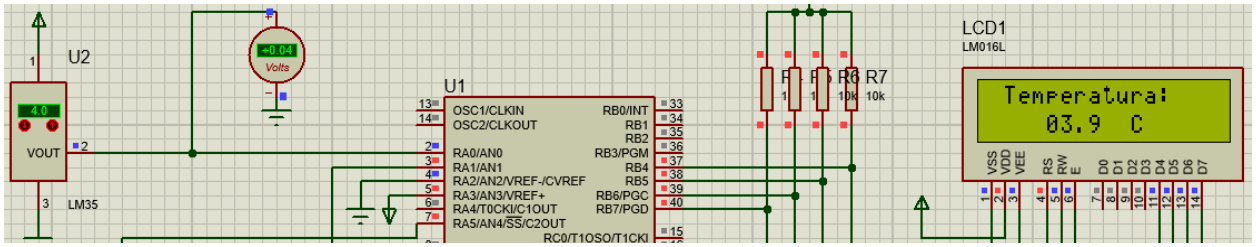
**** Le picamos el botón arriba.***

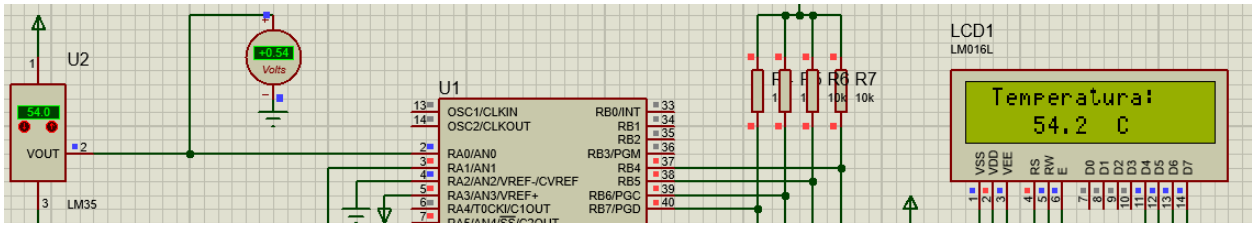


Lectura de temperatura:

**** Le picamos Enter en temperatura.***



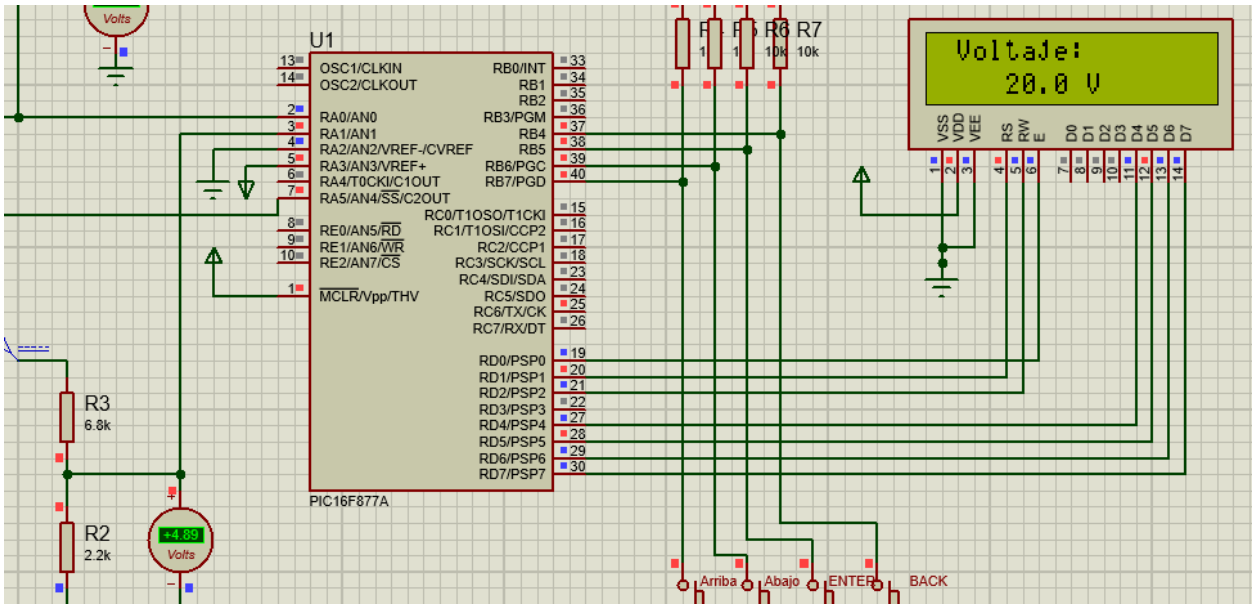




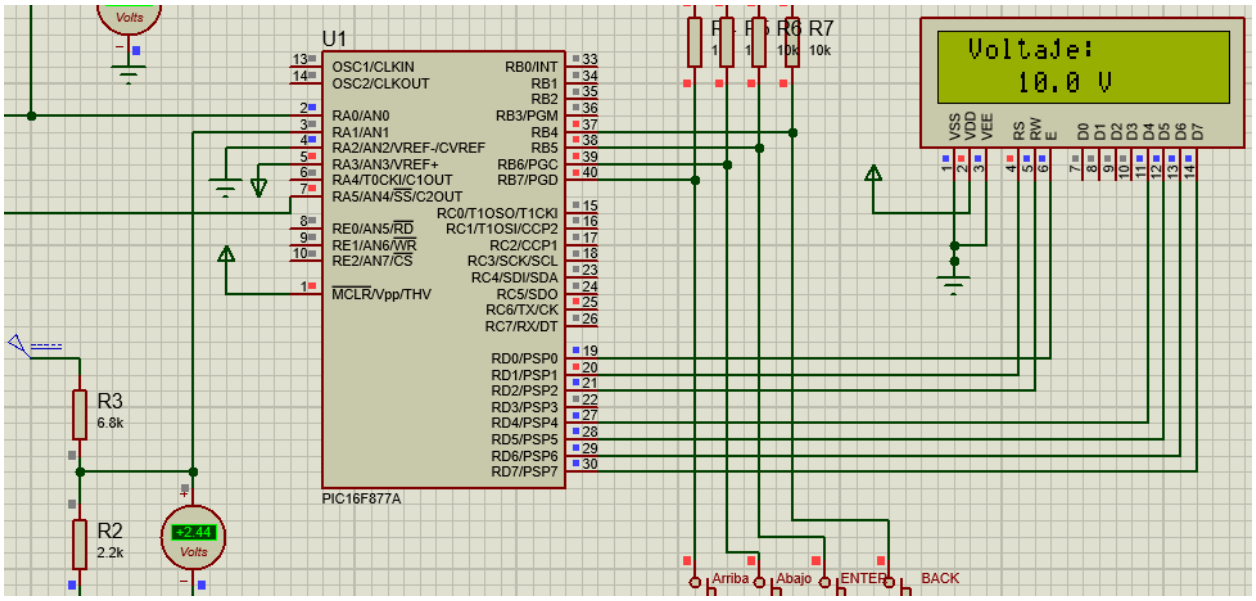
Lectura de voltaje:

**** Le picamos Enter en voltaje.***

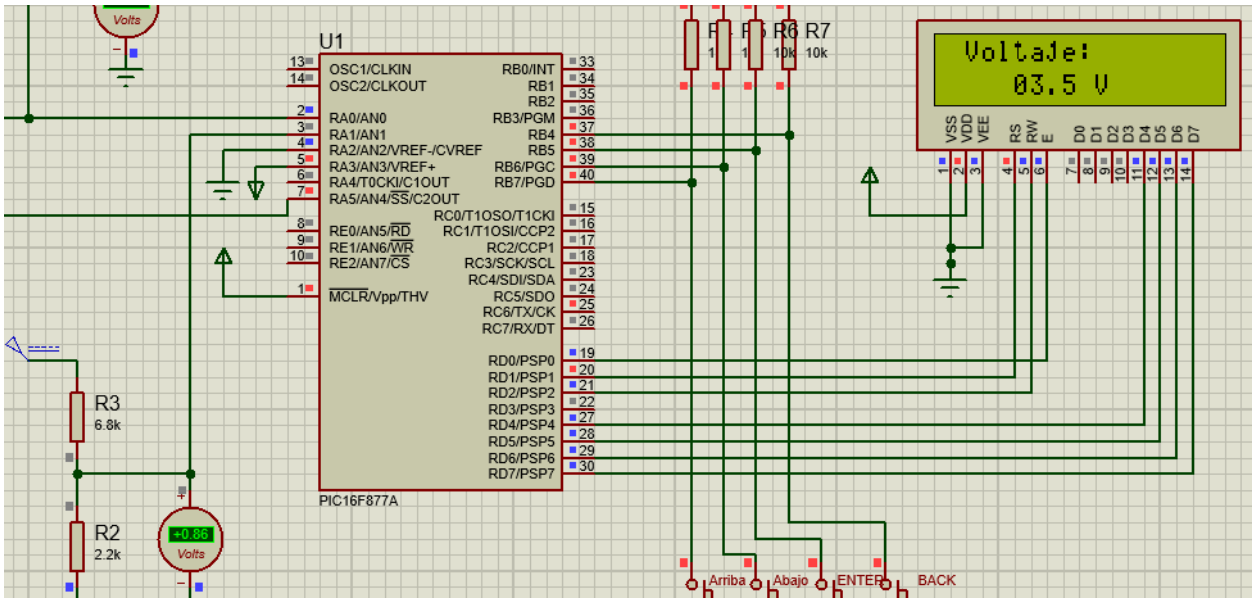
** Con una entrada de 20 V.



** Con una entrada de 10 V.



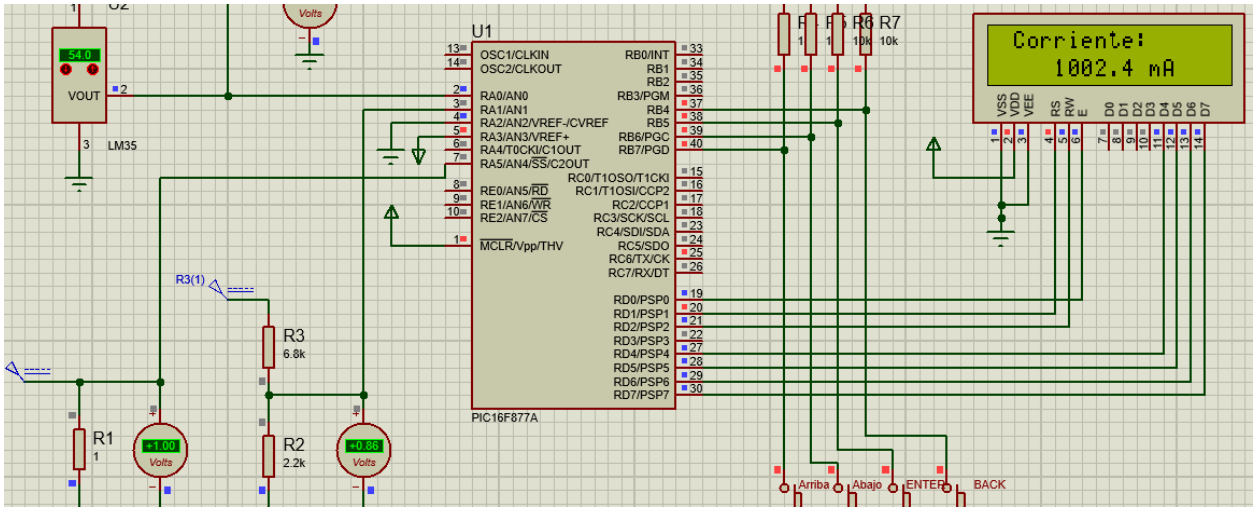
** Con una entrada de 3.5 V.



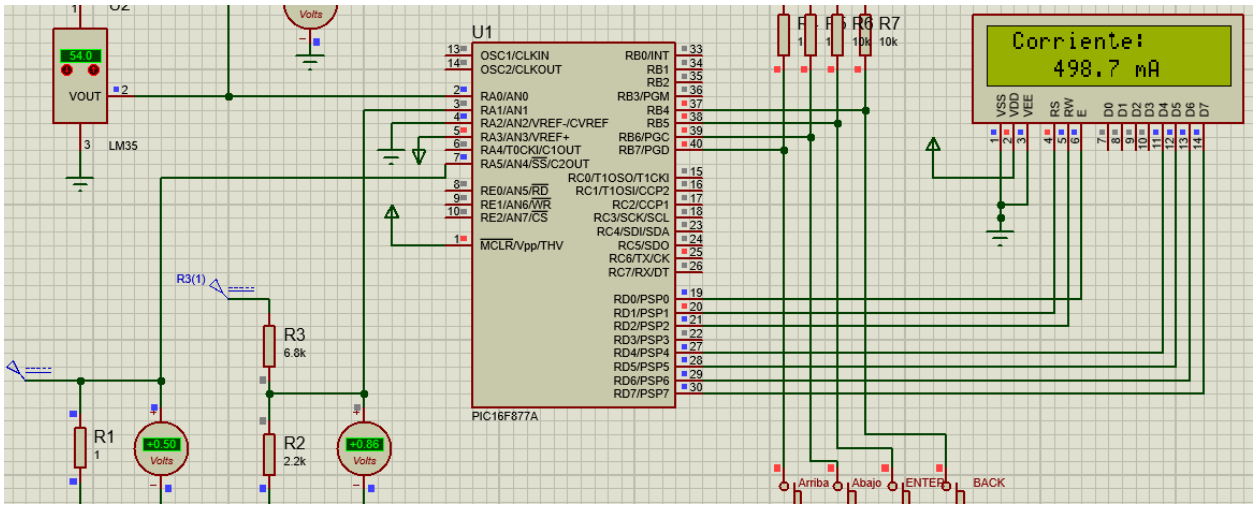
Lectura de corriente:

**** Le picamos Enter en corriente.***

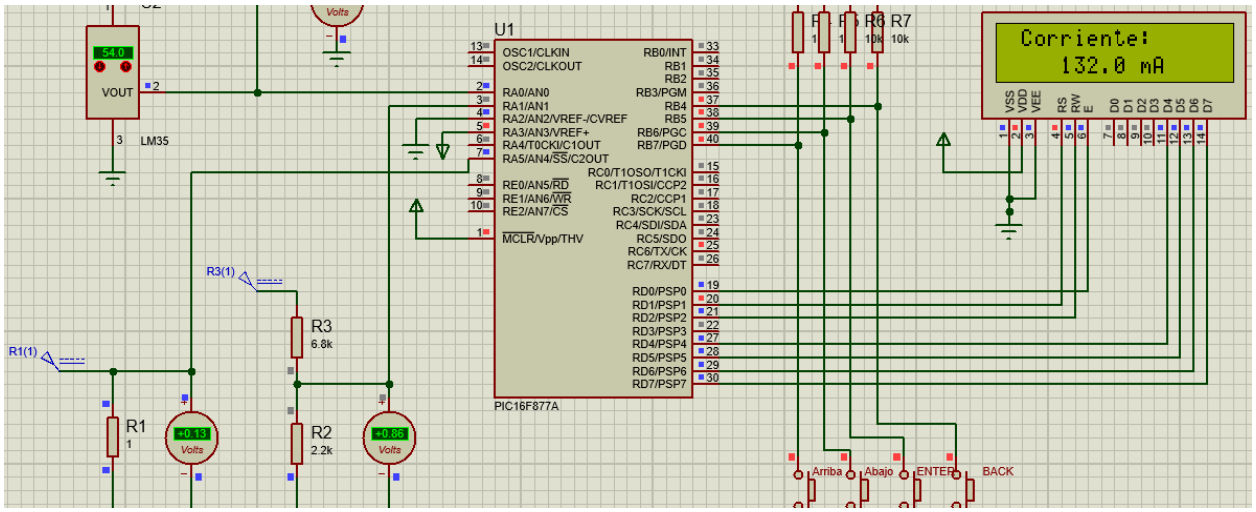
** Con 1 Volt de entrada.



** Con 0.5 Volts de entrada.



** Con 0.13 Volts de entrada.



***** DENTRO DE LA ENTREGA SE AÑADE LA SIMULACIÓN Y UN VIDEO PARA PODER VER EL FUNCIONAMIENTO *****

3. Análisis de resultados general

Nuestra propuesta de solución realmente es limpia y sencilla, por lo que resulta fácil de comprender. De forma resumida el código lo que hace es inicializar todos los componentes, mostrar en el display lcd la información correspondiente de los integrantes e imprimir el menú en su estado inicial. Después checa si se seleccionó una opción y con un pequeño retardo realiza las lecturas y las muestra en el display. Los 4 botones con los que se controla todo el sistema se manejan utilizando interrupciones (como se explica en la práctica 10) por lo que en el momento en el que cambian su estado se llama a la función de manejo de interrupciones de los botones Rb. Dentro de esta se checa si se está en el menú o ya dentro de una opción seleccionada y dependiendo de esto se puede mover el menú, entrar a una opción de lectura o salir de ésta. En la simulación se comprobó con diversas pruebas que el programa funciona tal y como se espera y cumple con lo requerido en el proyecto.

El sistema está constituido del PIC16 y los siguientes periféricos: 4 botones conectados a tierra para la interacción con el usuario (con sus resistencias pull up respectivas), un display lcd para mostrarle al usuario información mediante comunicación paralela, el sensor de temperatura LM35 para medir la temperatura en °C, fuentes de voltaje con sus resistencias (de 1k y un divisor de voltaje) para la lectura del voltaje y corriente. Todo el sistema está alimentado a través de 5 voltios conectados directamente al PIC16. Gracias al software Proteus se pudo realizar la simulación completa del sistema y comprobar que la solución cumple adecuadamente con su propósito. No muestra ningún error durante su funcionamiento y logra los objetivos planteados en un inicio satisfactoriamente al demostrar una buena aplicación de los conocimientos teóricos y prácticos revisados durante el curso y el haber aprendido a través de su hoja de información sobre el funcionamiento del sensor LM35. Además de que efectivamente es el primer sistema completo y práctico que se realizó en el laboratorio lo que nos permite aterrizar de mejor manera la utilidad de la asignatura y amplía nuestro horizonte de las microcomputadoras y controladores.

Como se explicó en un inicio del análisis el flujo de datos se realiza mayormente gracias a las interrupciones, las cuales se encargan de actualizar las variables de estados y muestran en el display la información correspondiente al llamar a las funciones adecuadas según el caso (imprimeMenu o imprimeLectura). Realmente el ciclo del hilo de ejecución sólo espera hasta que se seleccione una opción y entonces imprime cada x milisegundos la lectura registrada y convertida a las unidades necesarias. La comunicación para escribir en el display se vuelve muy sencilla utilizando la biblioteca ***lcd.c*** y conectándolo en el puerto paralelo D. El hilo principal sólo cede el control a las interrupciones que lo regresan al menú y mueven el mismo. Fuera de eso el ciclo continúa sin otra tarea. Finalmente como se ha mencionado anteriormente al programar en alto nivel se pierde el control de la memoria, pero se puede esperar que al utilizar variables por su nombre y ningún apuntador o arreglo, el direccionamiento utilizado a lo largo de toda la solución sería directo. Se confía en que el compilador sea eficiente con el uso de la memoria y la maneje de la mejor forma posible.

4. Conclusiones y comentarios (individuales)

Cruz Hernández, Alejandro Marcelo

Con la elaboración del proyecto final juntamos todo el conocimiento obtenido tanto en las clases de teoría como en la práctica dentro del laboratorio, donde estudiamos el funcionamiento de microcontroladores, también se tuvo que realizar algunas investigaciones adicionales para cumplir con algunos puntos del proyecto como el uso de interrupciones para hacer que al tocar un botón se realice la acción enseguida dentro del lcd. Usamos el microcontrolador para realizar el diseño y así llegar a implementar un sistema que nos despliega datos; todo esto nos dio una idea más amplia para utilizar los componentes que hay dentro del mercado de los sistemas de microcontroladores. Al realizar las simulaciones se cumplió con los objetivos de manera correcta desde nuestro puntos de vista de manera satisfactoria obteniendo lo esperado dentro del proyecto final.

García Sánchez, Emilio

Considero este proyecto final bastante completo, ya que nos ayuda a saber si realmente aprendimos lo necesario, tanto en el laboratorio como en la teoría. Ya que varios de los requerimientos del proyecto fueron similares a ejercicios realizados en el laboratorio y en teoría. Desafortunadamente no pudimos realizar la práctica 10 debido a la falta de tiempo del semestre, ya que esto nos hubiera ayudado a entender sin tanta dificultad el cómo implementar las interrupciones para nuestro display lcd a la hora de utilizar el menú de navegación con botones, sin embargo al final se pudo implementar con ayuda de un poco de investigación. Siento que este proyecto engloba la mayoría de lo visto en el curso del laboratorio ya que utilizamos datos análogos (como puede ser la temperatura y la corriente) y poder convertir los datos leídos para que puedan ser interpretados por nuestro microcontrolador.

Por lo que considero que se cumple en su totalidad el objetivo del proyecto final, ya que se implementó todo lo aprendido tanto en el ámbito teórico como en el laboratorio, recurriendo también a la investigación para llegar a los resultados deseados. De este proyecto aprendo de una mejor manera como funcionan los microcontroladores y la manera en que pueden implementarse en la vida cotidiana.

Ordiales Caballero, Iñaky

En la elaboración de este proyecto final de la asignatura pudimos poner a prueba gran parte de nuestro aprendizaje del laboratorio de microcomputadoras. Creo que sería muy difícil encontrar un sistema práctico y aplicado en el que se utilice todo lo revisado en el curso, pero con este pudimos cubrir buena parte. Inclusive se tuvo que revisar por nuestra cuenta la práctica 10 para comprender el uso de interrupciones que nos ayudaran a que los botones afectaran instantaneamente al display lcd. A mí la solución propuesta me pareció limpia y ordenada, realiza sin complicarse lo solicitado en el proyecto y en las simulación funciona muy bien. Además el hecho de poder conectar sensores analógicos y convertir sus mediciones a las unidades adecuadas pienso que es algo muy usado en la industria de los microcontroladores, por lo que siempre motiva ver los conocimientos teóricos-prácticos de una asignatura como futuras herramientas de nuestra carrera profesional. Viendo el objetivo me parece que se cumple sencillamente al hacer cualquier sistema que se aplique realmente como es el caso. Me gustó realizar esto como proyecto final del laboratorio, pero me quedo con las ganas de probarlo en físico y no sólo en la simulación.

Rojas Eng, Aurelio

La elaboración de este proyecto puede considerarse un gran resumen de los amplios conocimientos que nos dejó el laboratorio de microcomputadoras. Si bien no estamos viendo como tal cada una de las prácticas ni estamos codificando en un lenguaje ensamblador, todo lo visto sirvió como base para poder hacer el desarrollo correcto del proyecto.
Lo que bien se podría considerar a futuro es una implementación de este mismo sistema que construimos con la ayuda de Proteus, pero en físico con componentes que ya no sean prestados de laboratorio, lo que nos dejaría un mayor reto y conocimiento de las conexiones del sistema, que creo que es un punto a favor sobre lo que podríamos aprender, sin embargo, creo que los conocimientos físicos fueron solamente explicados y no se hizo realmente conexiones tan complejas como para poder hacer la implementación de este mismo sistema de una forma sencilla.
Para terminar quisiera sobre los diferentes tipos de microcomputadoras que existen en el mercado y cómo realmente podemos implementar esta misma solución en distintos chips, esto se me hace una idea muy interesante sobre el manejo de múltiples microcomputadoras y no solamente del que vimos a lo largo del curso, la PIC 16F877A, la cuál nos dejó un amplio conocimiento para poder indagar más y resolver problemas sin importar al microcontrolador que utilicemos. Creo entonces que cumplimos correctamente con los objetivos planteados.

5. Bibliografía:

Microchip Technology Inc. (2003). *PIC16F87XA Data Sheet 28/40/44-Pin Enhanced Flash Microcontrollers* (DS39582B). MICROCHIP.

CCS C COMPILER HELP (2021). From CCS INFO: ccs_c_manual.pdf https://www.ccsinfo.com/downloads/ccs_c_manual.pdf(ccsinfo.com)

Texas Instruments Incorporated. (2017, diciembre). *LM35 Precision Centigrade Temperature Sensors*. Texas Instruments. Recuperado 2 de enero de 2023, de https://www.ti.com/lit/ds/symlink/lm35.pdf

Castaño, S. (2022, 3 enero). *LCD con PIC*. Control Automático Educación. https://controlautomaticoeducacion.com/microcontroladores-pic/lcd-pic/

Labcenter Electronics, *Software Proteus 8.0 para la simulación de circuitos*