



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO



FACULTAD DE INGENIERÍA

MATERIA

- Laboratorio de Microcomputadoras
 - Grupo:04

PROYECTO FINAL

PROFESOR

- M.I. Ruben Anaya García

ALUMNOS

- Carreón Guzmán Mariana Ivette
 - Núm. Cta.: 312103914
 - Gpo. Teoría: 04
 - Rojas Méndez Gabriel
 - Núm. Cta.: 314141712

SEMESTRE 2022-1

Objetivo

El desarrollo de este proyecto final tiene como primicia la recopilación de todos los conceptos desarrollados a lo largo del curso de Laboratorio de Microcomputadoras, generando así una aplicación que sirva como repaso y visualización general de conocimientos obtenidos en la materia.

Introducción

A lo largo de este semestre pudimos desarrollar diferentes prácticas, en las cuales se nos mostró los componentes de un microcontrolador, la forma en la que funcionan sus puertos, así como sus ventajas y desventajas.

En el caso de nuestro proyecto optamos por hacer uso del concepto de esclavo y maestro y juntar nuestro conversor, usar el display para mostrar nuestros nombres y las conversiones realizadas junto con la comunicación I2C.

Todo esto resultó sumamente interesante, ya que tuvimos que revisar los conceptos aprendidos para realizar correctamente la reproducción de nuestro programa, se tuvo algunos problemas al momento de la implementación del esclavo y la comunicación I2C que al final se pudo solucionar. El resultado final del proyecto se muestra a continuación

Desarrollo

Se optó por el uso de el lenguaje C ya que nos encontramos más familiarizados con este lenguaje y sin duda fue más efectivo.

```
Proyecto Final.c
1  #include <16F877.h> //Librería de microcontrolador
2  #fuses HS,NOPROTECT,NOBROWNOUT
3  #device ADC = 10
4  #use delay(clock=2000000) //Frec. de Osc. 20Mhz
5  #use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7) //Configuración Puerto SERIAL
6  #org 0x1F00, 0x1FFF void loader16f877(void) {}
7  #use i2c(MASTER, SDA=PIN_C4, SCL=PIN_C3,SLOW, NOFORCE_SW)
8  #include <i2c_Flex_LCD.c>
9
10 int16 conversion = 0;
11 int contador = 0, carga = 0;
12 char dato;
13
14 void escribir_i2c(){
15     i2c_start();
16     i2c_write(0x42);
17     i2c_write(contador);
18     i2c_stop();
19 }
20
21 void leer_i2c(){ //Función para leer datos del esclavo.
22     i2c_start(); //Inicia la comunicación I2C.
23     i2c_write(0x41); //Modo escritura dirección.
24     carga = i2c_read(0); //Lectura de los datos del esclavo.
25     i2c_stop(); //Fin de la comunicación.
26 }
27
28 #int_rda
29 void recepcion_serie(){
30     dato = getchar();
31     printf("Se recibio el dato: %c\n\r", dato);
32 }
33
```

Como se puede observar en la captura anterior, se realizaron 2 funciones una que lee y otra que escribe lo cual permite la implementación de la comunicación I2C.

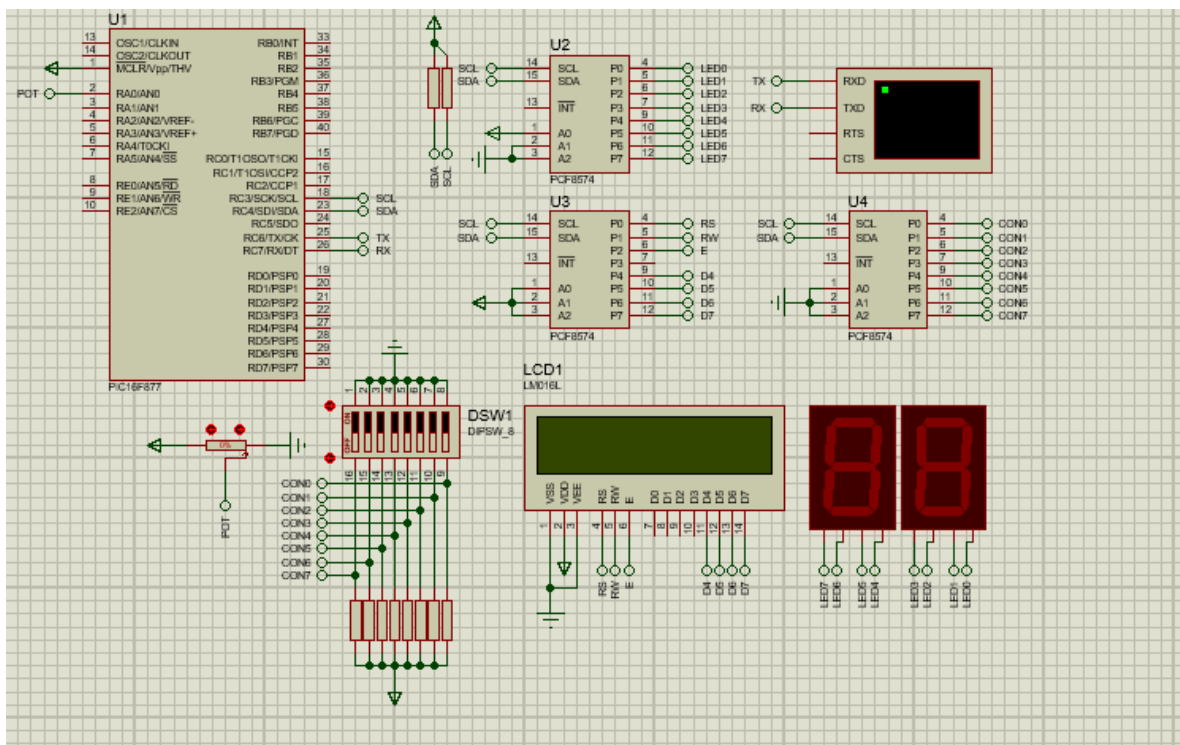
```
Proyecto Final.c
34 void main(){
35     enable_interrupts(INT_RDA);
36     enable_interrupts(GLOBAL);
37     lcd_init(0x4E,16,2);
38     lcd_backlight_led(ON);
39     setup_port_a(ALL_ANALOG);
40     setup_adc(ADC_CLOCK_INTERNAL);
41     set_adc_channel(0);
42     delay_us(20);
43     while(TRUE){
44         switch(dato)
45         {
46             case '0':
47                 lcd_gotoxy(1,1);
48                 printf(LCD_PUTC, "Gabriel  ");
49                 lcd_gotoxy(1,2);
50                 printf(LCD_PUTC, "Mariana  ");
51                 break;
52             case '1':
53                 escribir_i2c();
54                 contador++;
55                 delay_ms(200);
56                 break;
57             case '2':
58                 conversion = read_adc();
59                 lcd_gotoxy(1,1);
60                 printf(LCD_PUTC, "Voltaje");
61                 lcd_gotoxy(1,2);
62                 printf(LCD_PUTC, "Vin %2.2f V", conversion/204.60);
63                 break;
64             case '3':
65                 break;
66         }
67     }
68 }
```

```

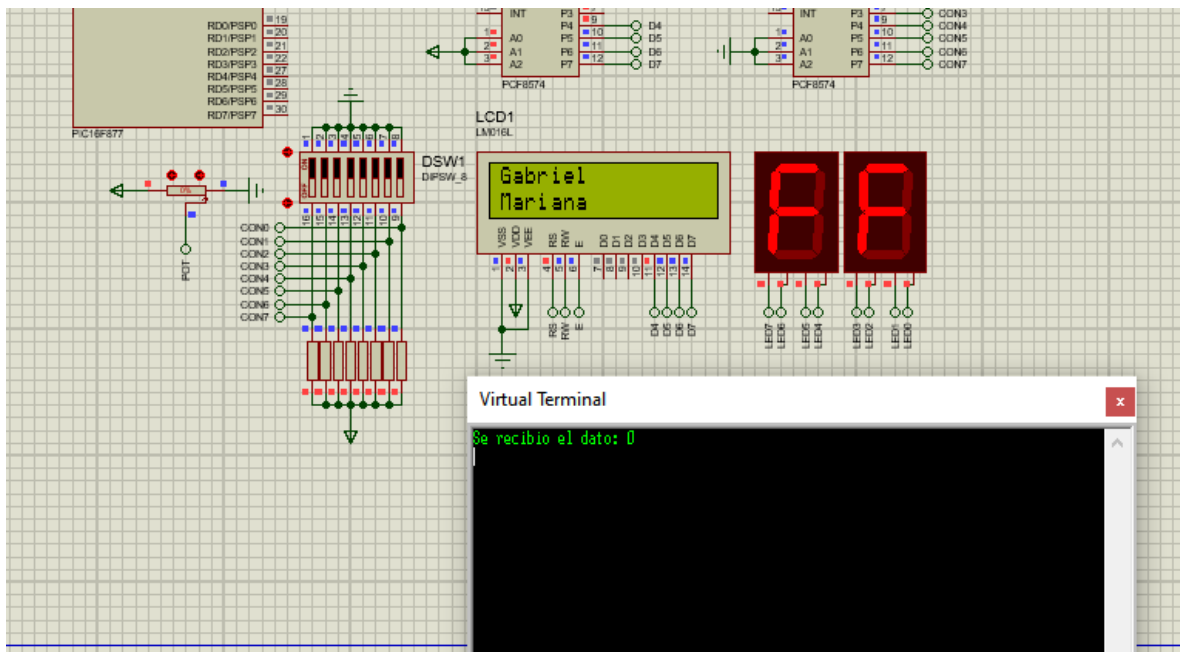
66
67
68     case '3':
69         leer_i2c();
70         lcd_gotoxy(1,1);
71         printf(LCD_PUTC, "Dec      ");
72         lcd_gotoxy(1,2);
73         printf(LCD_PUTC, "%u      ", carga);
74         break;
75
76     case '4':
77         leer_i2c();
78         lcd_gotoxy(1,1);
79         printf(LCD_PUTC, "Hex      ");
80         lcd_gotoxy(1,2);
81         printf(LCD_PUTC, "%x      ", carga);
82         break;
83
84     default:
85         lcd_gotoxy(1,1);
86         printf(LCD_PUTC, "Dato");
87         lcd_gotoxy(1,2);
88         printf(LCD_PUTC, "invalido");
89
90 }
91

```

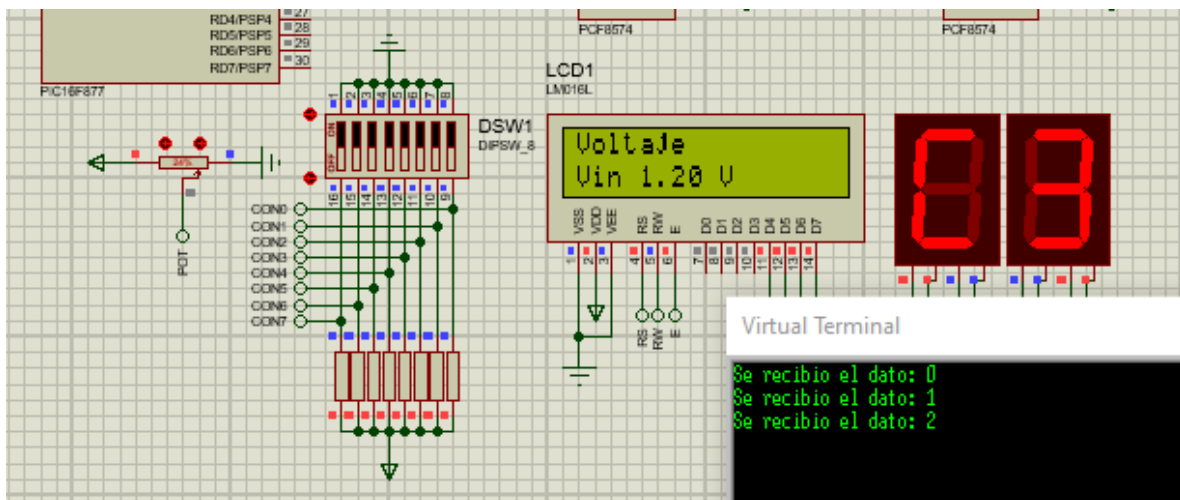
Ejecución del programa



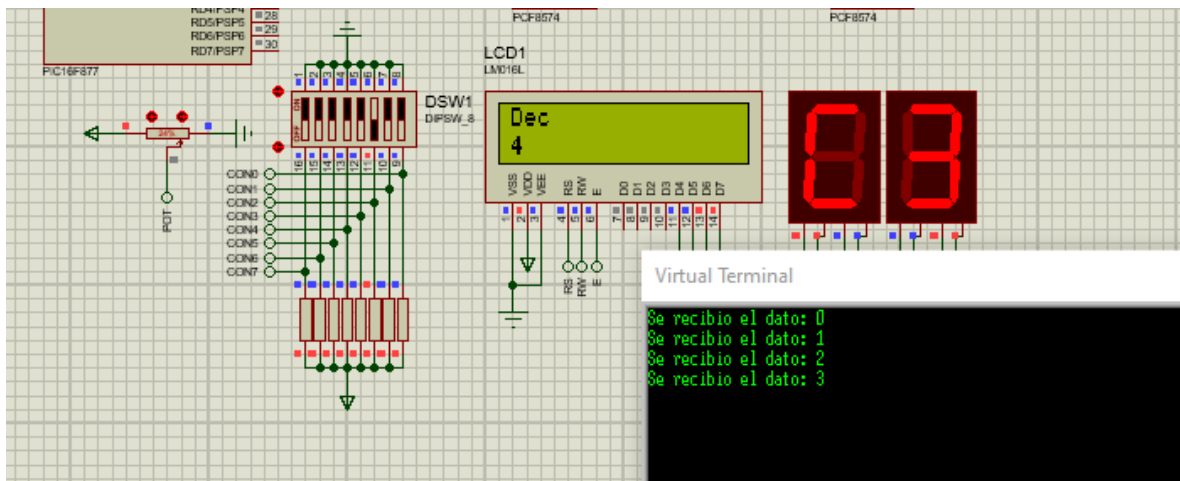
Case "0"



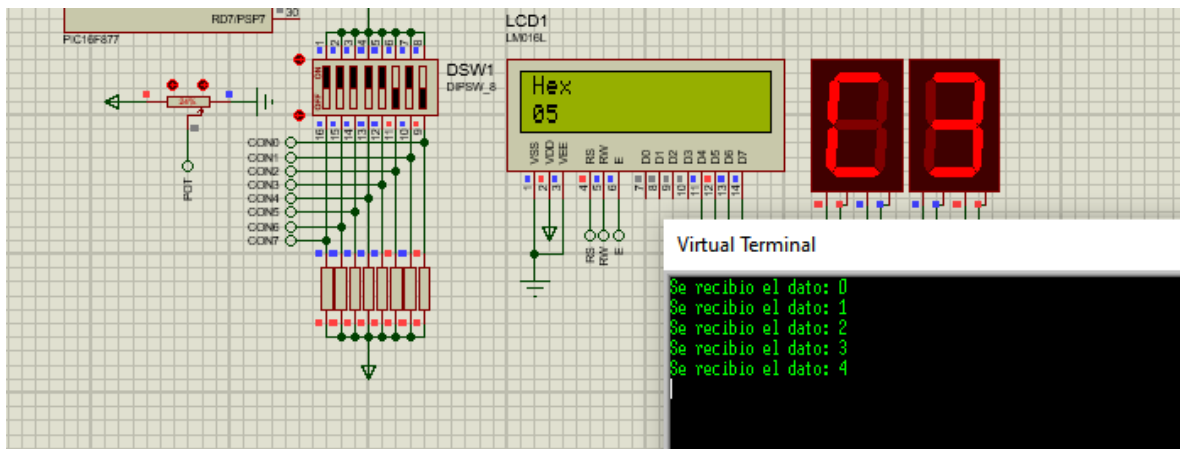
Case "3"



Case "3"



Case "4"



Conclusiones

A lo largo del curso se fueron desarrollando diversos temas, los cuales resultan de suma importancia a la hora de poder desarrollar aplicaciones para un microcontrolador, hoy en día en la vida cotidiana podemos ver diversas aplicaciones que son realizadas mediante estos dispositivos, es por ello que resulta importante saber las bases que manejan para poder desarrollar nuevas y mejores aplicaciones. En este caso optamos por diseñar un conversor de unidades, pero podría también desarrollarse una aplicación que controle los grados de movimientos, ya que sabemos cómo controlar un servomotor, o bien podemos controlar la pluma de un estacionamiento y desplegar en pantalla un mensaje que diga los pasos a seguir para pagar un boleto de estacionamiento.

El objetivo principal es poder desarrollar nuestras habilidades con los microcontroladores y creemos que con la realización de este proyecto se cumplieron todas las expectativas tanto del curso, como del trabajo.