UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

MICROCOMPUTADORAS

GRUPO 01

PROYECTO 4 FINAL: SEMÁFORO INTELIGENTE

ROJAS MÉNDEZ GABRIEL

13/01/2022

SEMESTRE 2022-1

Requerimientos:

Para esta cuarta entrega de un proyecto empleando el microcontrolador PIC16F877 se dejó a elección, en donde se pudiera ver reflejados y aplicados los conocimientos adquiridos a lo largo del curso, además, este proyecto debía estar aplicado a un beneficio para la sociedad, así que se optó por desarrollar un proyecto el cual permitiera un funcionamiento de un semáforo inteligente, el cual permitiera recibir peticiones de los peatones en una intercepción de dos calles, para así poder generar un cambio en el semáforo y así detener el flujo vial, siempre y cuando sea necesario y exista una causa de fuerza mayor, de esta manera se busca como beneficio el aumento en la educación vial, aunque este proyecto puede servir como base para más funciones inteligentes y obtener mayor provecho.

Diseño:

Para lograr el funcionamiento adecuado de este proyecto, se empleó la comunicación I^2C para poder implementar la expansión de puertos, para que así se pudiera implementar dos semáforos para vías de automóviles, cuatro semáforos para uso peatonal y cuatro contadores indicadores de tiempo para que así el peatón sepa cuando hacer alto total debido a un cambio en el paso. Al diseño anterior se agregaron cuatro botones que permitieran capturar las peticiones de los peatones que por alguna emergencia necesiten cruzar el arrollo vehicular, de tal manera, esta función busca que la educación vial sea mayor y evite que los peatones crucen esquivando vehículos o por lugares en donde no se encuentren pasos peatonales, para dicha función de recibir solicitudes del peatón se empleó el tema de interrupciones en el PIC, para ser más específicos, la interrupción de cambios en los puertos RB4 a RB7, de esta manera la solicitud es procesada y al finalizar continua con su funcionamiento base.

Algoritmo empleado:

1) Funcionamiento base de los semáforos.
    a. Si semáforo == 0
        i. Si contador > 0
            1. Llamada a función Contador1_i2c().
            2. Llamada a función Semáforo_verde_i2c().
            3. Llamada a función Contador_i2c().
            4. Llamada a función Semáforo1_rojo_i2c().
            5. Retardo de un 1 segundo.
            6. Decremento de contador1.
        ii. Sino
            1. Contador1 = 0.
            2. Llamada a Contador1_i2c().
            3. Llamada a Semáforo_apagado_i2c().
            4. Retraso de 500 milisegundos.
            5. Llamada a Semáforo_verde_i2c().
            6. Retraso de 500 milisegundos.
            7. Llamada a Semáforo_apagado_i2c().
            8. Retraso de 500 milisegundos.
            9. Llamada a Semáforo_verde_i2c().
            10. Retraso de 500 milisegundos.
            11. Llamada a Semáforo_apagado_i2c().
            12. Retraso de 500 milisegundos.
            13. Llamada a Semáforo_verde_i2c().
            14. Retraso de 500 milisegundos.
            15. Llamada a Semáforo_apagado_i2c().
            16. Retraso de 500 milisegundos.
            17. Llamada a Semáforo_verde_i2c().
            18. Retraso de 500 milisegundos.

19. Llamada a Semáforo_apagado_i2c().
20. Retraso de 500 milisegundos.
21. Llamada a Semáforo_amarillo_i2c().
22. Retraso de 2 segundos.
23. Contador1 = 61.
24. Semáforo = 1.

b. Sino
    i. Si contador > 0
        1. Llamada a función Contador_i2c().
        2. Llamada a función Semáforo1_verde_i2c().
        3. Llamada a función Semáforo_rojo_i2c().
        4. Retardo de un 1 segundo.
        5. Decremento de contador.
    ii. Sino
        1. Contador = 0.
        2. Llamada a Contador_i2c().
        3. Llamada a Semáforo1_apagado_i2c().
        4. Retraso de 500 milisegundos.
        5. Llamada a Semáforo1_verde_i2c().
        6. Retraso de 500 milisegundos.
        7. Llamada a Semáforo1_apagado_i2c().
        8. Retraso de 500 milisegundos.
        9. Llamada a Semáforo1_verde_i2c().
        10. Retraso de 500 milisegundos.
        11. Llamada a Semáforo1_apagado_i2c().
        12. Retraso de 500 milisegundos.
        13. Llamada a Semáforo1_verde_i2c().
        14. Retraso de 500 milisegundos.
        15. Llamada a Semáforo1_apagado_i2c().
        16. Retraso de 500 milisegundos.
        17. Llamada a Semáforo1_verde_i2c().

18. Retraso de 500 milisegundos.
19. Llamada a Semáforo1_apagado_i2c().
20. Retraso de 500 milisegundos.
21. Llamada a Semáforo1_amarillo_i2c().
22. Retraso de 2 segundos.
23. Contador = 61.
24. Semáforo = 0.

2) Funcionamiento para la atención de interrupciones.
   a. Si la entrada en PIN B4 == 1
      i. Contador1 = 0.
      ii. Llamada a Contador1_i2c().
      iii. Llamada a Semáforo_apagado_i2c().
      iv. Retraso de 500 milisegundos.
      v. Llamada a Semáforo_verde_i2c().
      vi. Retraso de 500 milisegundos.
      vii. Llamada a Semáforo_apagado_i2c().
      viii. Retraso de 500 milisegundos.
      ix. Llamada a Semáforo_verde_i2c().
      x. Retraso de 500 milisegundos.
      xi. Llamada a Semáforo_apagado_i2c().
      xii. Retraso de 500 milisegundos.
      xiii. Llamada a Semáforo_verde_i2c().
      xiv. Retraso de 500 milisegundos.
      xv. Llamada a Semáforo_apagado_i2c().
      xvi. Retraso de 500 milisegundos.
      xvii. Llamada a Semáforo_verde_i2c().
      xviii. Retraso de 500 milisegundos.
      xix. Llamada a Semáforo_apagado_i2c().
      xx. Retraso de 500 milisegundos.
      xxi. Llamada a Semáforo_amarillo_i2c().
      xxii. Retraso de 2 segundos.

xxiii. Contador1 = 61.
xxiv. Semáforo = 1.

b. Si la entrada en PIN B5 == 1
   i. Contador1 = 0.
   ii. Llamada a Contador1_i2c().
   iii. Llamada a Semáforo_apagado_i2c().
   iv. Retraso de 500 milisegundos.
   v. Llamada a Semáforo_verde_i2c().
   vi. Retraso de 500 milisegundos.
   vii. Llamada a Semáforo_apagado_i2c().
   viii. Retraso de 500 milisegundos.
   ix. Llamada a Semáforo_verde_i2c().
   x. Retraso de 500 milisegundos.
   xi. Llamada a Semáforo_apagado_i2c().
   xii. Retraso de 500 milisegundos.
   xiii. Llamada a Semáforo_verde_i2c().
   xiv. Retraso de 500 milisegundos.
   xv. Llamada a Semáforo_apagado_i2c().
   xvi. Retraso de 500 milisegundos.
   xvii. Llamada a Semáforo_verde_i2c().
   xviii. Retraso de 500 milisegundos.
   xix. Llamada a Semáforo_apagado_i2c().
   xx. Retraso de 500 milisegundos.
   xxi. Llamada a Semáforo_amarillo_i2c().
   xxii. Retraso de 2 segundos.
   xxiii. Contador1 = 61.
   xxiv. Semáforo = 1.

c. Si la entrada en PIN B6 == 1
   i. Contador = 0.
   ii. Llamada a Contador_i2c().
   iii. Llamada a Semáforo1_apagado_i2c().

iv. Retraso de 500 milisegundos.
v. Llamada a Semáforo1_verde_i2c().
vi. Retraso de 500 milisegundos.
vii. Llamada a Semáforo1_apagado_i2c().
viii. Retraso de 500 milisegundos.
ix. Llamada a Semáforo1_verde_i2c().
x. Retraso de 500 milisegundos.
xi. Llamada a Semáforo1_apagado_i2c().
xii. Retraso de 500 milisegundos.
xiii. Llamada a Semáforo1_verde_i2c().
xiv. Retraso de 500 milisegundos.
xv. Llamada a Semáforo1_apagado_i2c().
xvi. Retraso de 500 milisegundos.
xvii. Llamada a Semáforo1_verde_i2c().
xviii. Retraso de 500 milisegundos.
xix. Llamada a Semáforo1_apagado_i2c().
xx. Retraso de 500 milisegundos.
xxi. Llamada a Semáforo1_amarillo_i2c().
xxii. Retraso de 2 segundos.
xxiii. Contador = 61.
xxiv. Semáforo = 0.

d. Si la entrada en PIN B6 == 1
    i. Contador = 0.
    ii. Llamada a Contador_i2c().
    iii. Llamada a Semáforo1_apagado_i2c().
    iv. Retraso de 500 milisegundos.
    v. Llamada a Semáforo1_verde_i2c().
    vi. Retraso de 500 milisegundos.
    vii. Llamada a Semáforo1_apagado_i2c().
    viii. Retraso de 500 milisegundos.
    ix. Llamada a Semáforo1_verde_i2c().

x. Retraso de 500 milisegundos.
xi. Llamada a Semáforo1_apagado_i2c().
xii. Retraso de 500 milisegundos.
xiii. Llamada a Semáforo1_verde_i2c().
xiv. Retraso de 500 milisegundos.
xv. Llamada a Semáforo1_apagado_i2c().
xvi. Retraso de 500 milisegundos.
xvii. Llamada a Semáforo1_verde_i2c().
xviii. Retraso de 500 milisegundos.
xix. Llamada a Semáforo1_apagado_i2c().
xx. Retraso de 500 milisegundos.
xxi. Llamada a Semáforo1_amarillo_i2c().
xxii. Retraso de 2 segundos.
xxiii. Contador = 61.
xxiv. Semáforo = 0.

3) Funcionamiento para los indicadores de tiempo.
   a. Iniciar comunicación I^2C.
   b. Indicar la dirección del expansor de puertos PCF8574.
   c. Indicar el valor del contador a transmitir al esclavo.
   d. Finalizar la comunicación I^2C.
4) Funcionamiento para los distintos estados del semáforo (verde).
   a. Iniciar comunicación I^2C.
   b. Indicar la dirección del expansor de puertos PCF8574.
   c. Indicar el valor del contador a transmitir al esclavo (67D).
   d. Finalizar la comunicación I^2C.
5) Funcionamiento para los distintos estados del semáforo (amarillo).
   a. Iniciar comunicación I^2C.
   b. Indicar la dirección del expansor de puertos PCF8574.
   c. Indicar el valor del contador a transmitir al esclavo (76D).
   d. Finalizar la comunicación I^2C.
6) Funcionamiento para los distintos estados del semáforo (rojo).

a. Iniciar comunicación I^2C.
b. Indicar la dirección del expansor de puertos PCF8574.
c. Indicar el valor del contador a transmitir al esclavo (176D).
d. Finalizar la comunicación I^2C.

Código:

Proyecto Final.c

```
#include <16F877.h>
#fuses HS, NOPROTECT, NOWDT
#use delay(clock=20000000)
#use i2c(MASTER, SDA=PIN_C4, SCL=PIN_C3, SLOW, NOFORCE_SW)

byte const display[61]={0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17, 0x18, 0x19,
0x20, 0x21, 0x22, 0x23, 0x24, 0x25, 0x26, 0x27, 0x28, 0x29, 0x30, 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37, 0x38, 0x39, 0x40, 0x41, 0x42, 0x43,
0x44, 0x45, 0x46, 0x47, 0x48, 0x49, 0x50, 0x51, 0x52, 0x53, 0x54, 0x55, 0x56, 0x57, 0x58, 0x59, 0x60};

int contador = 61;
int contador1 = 61;
int verde = 67;
int amarillo = 76;
int rojo = 176;
int bandera = 0;

void contador_i2c(){
   i2c_start();
   i2c_write(0x40);
   i2c_write(display[contador]);
   i2c_stop();
}

void contador1_i2c(){
   i2c_start();
   i2c_write(0x42);
   i2c_write(display[contador1]);
   i2c_stop();
}

void semaforo_verde_i2c(){
   i2c_start();
   i2c_write(0x44);
```

Proyecto Final.c

```
   i2c_write(0x44);
   i2c_write(verde);
   i2c_stop();
}

void semaforo_apagado_i2c(){
   i2c_start();
   i2c_write(0x44);
   i2c_write(0x40);
   i2c_stop();
}

void semaforo_amarillo_i2c(){
   i2c_start();
   i2c_write(0x44);
   i2c_write(amarillo);
   i2c_stop();
}

void semaforo_rojo_i2c(){
   i2c_start();
   i2c_write(0x44);
   i2c_write(rojo);
   i2c_stop();
}

void semaforo1_verde_i2c(){
   i2c_start();
   i2c_write(0x46);
   i2c_write(verde);
   i2c_stop();
}
```

Proyecto Final.c

```
66  void semaforo1_apagado_i2c(){
67     i2c_start();
68     i2c_write(0x46);
69     i2c_write(0x40);
70     i2c_stop();
71  }
72
73  void semaforo1_amarillo_i2c(){
74     i2c_start();
75     i2c_write(0x46);
76     i2c_write(amarillo);
77     i2c_stop();
78  }
79
80  void semaforo1_rojo_i2c(){
81     i2c_start();
82     i2c_write(0x46);
83     i2c_write(rojo);
84     i2c_stop();
85  }
86
87  #int_rb
88  void peticion_cambio(){
89     if(input(PIN_B4) == 1){
90           contador1 = 0;
91           contador1_i2c();
92           semaforo_apagado_i2c();
93           delay_ms(500);
94           semaforo_verde_i2c();
95           delay_ms(500);
96           semaforo_apagado_i2c();
97           delay_ms(500);
98           semaforo_verde_i2c();
```

Proyecto Final.c

```
99           delay_ms(500);
100          semaforo_apagado_i2c();
101          delay_ms(500);
102          semaforo_verde_i2c();
103          delay_ms(500);
104          semaforo_apagado_i2c();
105          delay_ms(500);
106          semaforo_verde_i2c();
107          delay_ms(500);
108          semaforo_apagado_i2c();
109          delay_ms(500);
110          semaforo_amarillo_i2c();
111          delay_ms(2000);
112          contador1 = 61;
113          bandera = 1;
114    }
115    if(input(PIN_B5) == 1){
116          contador1 = 0;
117          contador1_i2c();
118          semaforo_apagado_i2c();
119          delay_ms(500);
120          semaforo_verde_i2c();
121          delay_ms(500);
122          semaforo_apagado_i2c();
123          delay_ms(500);
124          semaforo_verde_i2c();
125          delay_ms(500);
126          semaforo_apagado_i2c();
127          delay_ms(500);
128          semaforo_verde_i2c();
129          delay_ms(500);
130          semaforo_apagado_i2c();
131          delay_ms(500);
```

Proyecto Final.c

```
132         semaforo_verde_i2c();
133         delay_ms(500);
134         semaforo_apagado_i2c();
135         delay_ms(500);
136         semaforo_amarillo_i2c();
137         delay_ms(2000);
138         contador1 = 61;
139         bandera = 1;
140     }
141     if(input(PIN_B6) == 1){
142         contador = 0;
143         contador_i2c();
144         semaforo1_apagado_i2c();
145         delay_ms(500);
146         semaforo1_verde_i2c();
147         delay_ms(500);
148         semaforo1_apagado_i2c();
149         delay_ms(500);
150         semaforo1_verde_i2c();
151         delay_ms(500);
152         semaforo1_apagado_i2c();
153         delay_ms(500);
154         semaforo1_verde_i2c();
155         delay_ms(500);
156         semaforo1_apagado_i2c();
157         delay_ms(500);
158         semaforo1_verde_i2c();
159         delay_ms(500);
160         semaforo1_apagado_i2c();
161         delay_ms(500);
162         semaforo1_amarillo_i2c();
163         delay_ms(2000);
164         contador = 61;
```

Proyecto Final.c

```
165         bandera = 0;
166     }
167     if(input(PIN_B7) == 1){
168         contador = 0;
169         contador_i2c();
170         semaforo1_apagado_i2c();
171         delay_ms(500);
172         semaforo1_verde_i2c();
173         delay_ms(500);
174         semaforo1_apagado_i2c();
175         delay_ms(500);
176         semaforo1_verde_i2c();
177         delay_ms(500);
178         semaforo1_apagado_i2c();
179         delay_ms(500);
180         semaforo1_verde_i2c();
181         delay_ms(500);
182         semaforo1_apagado_i2c();
183         delay_ms(500);
184         semaforo1_verde_i2c();
185         delay_ms(500);
186         semaforo1_apagado_i2c();
187         delay_ms(500);
188         semaforo1_amarillo_i2c();
189         delay_ms(2000);
190         contador = 61;
191         bandera = 0;
192     }
193 }
194
195 void main(){
196     enable_interrupts(INT_RB);
197     enable_interrupts(GLOBAL);
```

Proyecto Final.c

```
198    while(TRUE){
199       if(bandera == 0){
200          if(contador1 > 0){
201             contador1_i2c();
202             semaforo_verde_i2c();
203             contador_i2c();
204             semaforo1_rojo_i2c();
205             delay_ms(250);
206             contador1--;
207          }
208          else{
209             contador1 = 0;
210             contador1_i2c();
211             semaforo_apagado_i2c();
212             delay_ms(500);
213             semaforo_verde_i2c();
214             delay_ms(500);
215             semaforo_apagado_i2c();
216             delay_ms(500);
217             semaforo_verde_i2c();
218             delay_ms(500);
219             semaforo_apagado_i2c();
220             delay_ms(500);
221             semaforo_verde_i2c();
222             delay_ms(500);
223             semaforo_apagado_i2c();
224             delay_ms(500);
225             semaforo_verde_i2c();
226             delay_ms(500);
227             semaforo_apagado_i2c();
228             delay_ms(500);
229             semaforo_amarillo_i2c();
230             delay_ms(2000);
```

Proyecto Final.c

```
231             contador1 = 61;
232             bandera = 1;
233          }
234       }
235       else{
236          if(contador > 0){
237             contador_i2c();
238             semaforo1_verde_i2c();
239             semaforo_rojo_i2c();
240             delay_ms(250);
241             contador--;
242          }
243          else{
244             contador = 0;
245             contador_i2c();
246             semaforo1_apagado_i2c();
247             delay_ms(500);
248             semaforo1_verde_i2c();
249             delay_ms(500);
250             semaforo1_apagado_i2c();
251             delay_ms(500);
252             semaforo1_verde_i2c();
253             delay_ms(500);
254             semaforo1_apagado_i2c();
255             delay_ms(500);
256             semaforo1_verde_i2c();
257             delay_ms(500);
258             semaforo1_apagado_i2c();
259             delay_ms(500);
260             semaforo1_verde_i2c();
261             delay_ms(500);
262             semaforo1_apagado_i2c();
263             delay_ms(500);
```

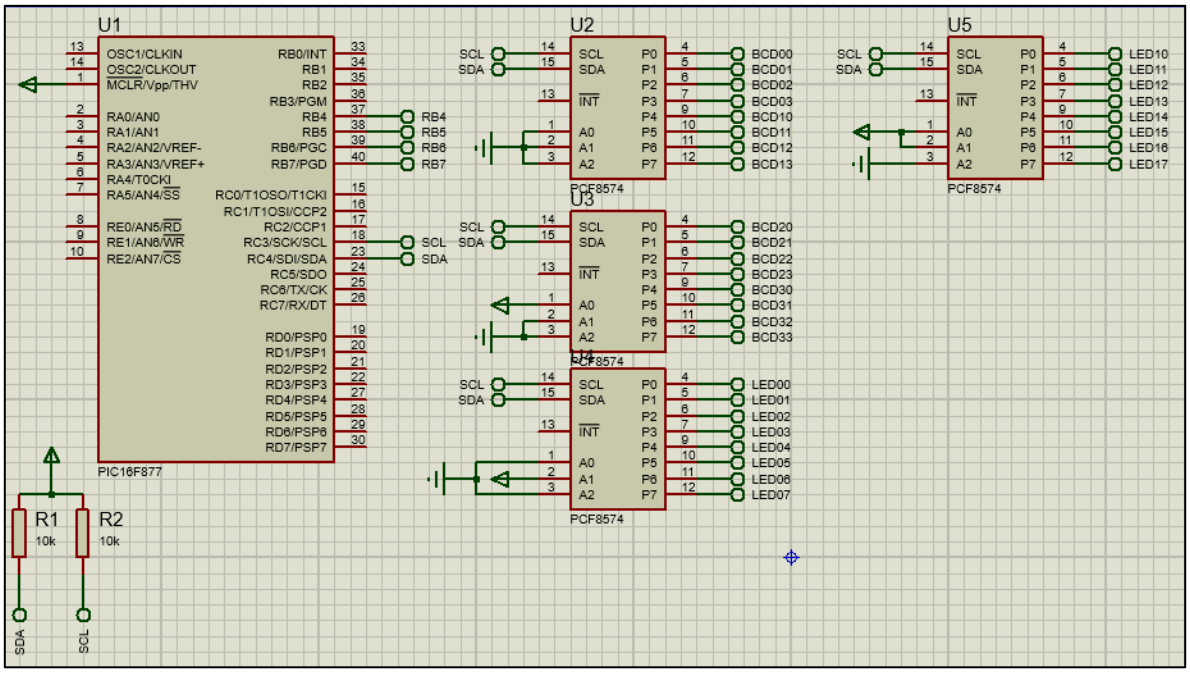
Proyecto Final.c*

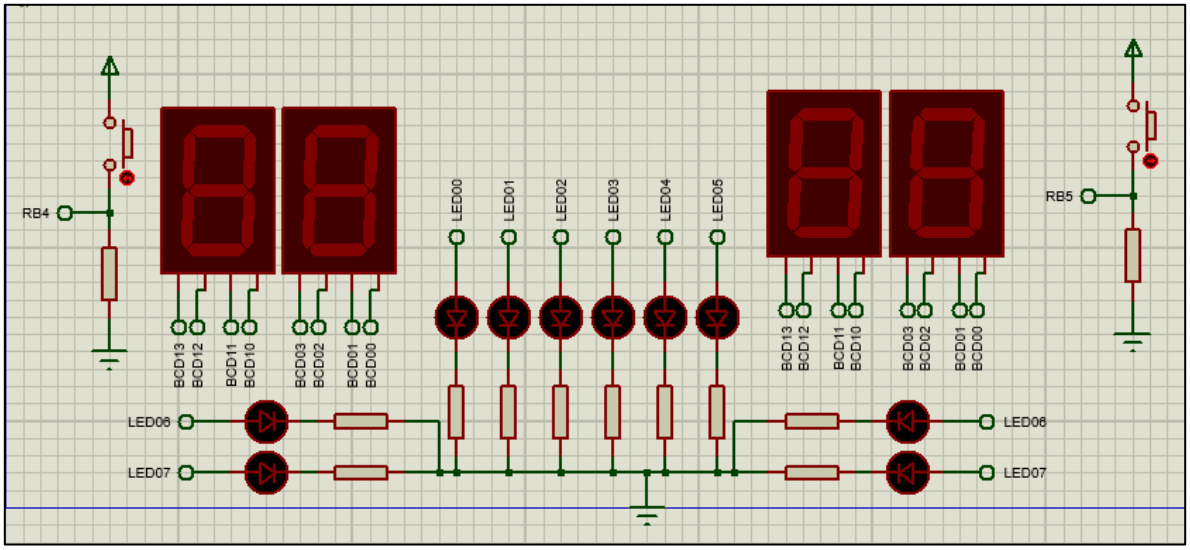
```
264             semaforo1_amarillo_i2c();
265             delay_ms(2000);
266             contador = 61;
267             bandera = 0;
268          }
269       }
270    }
271 }
```

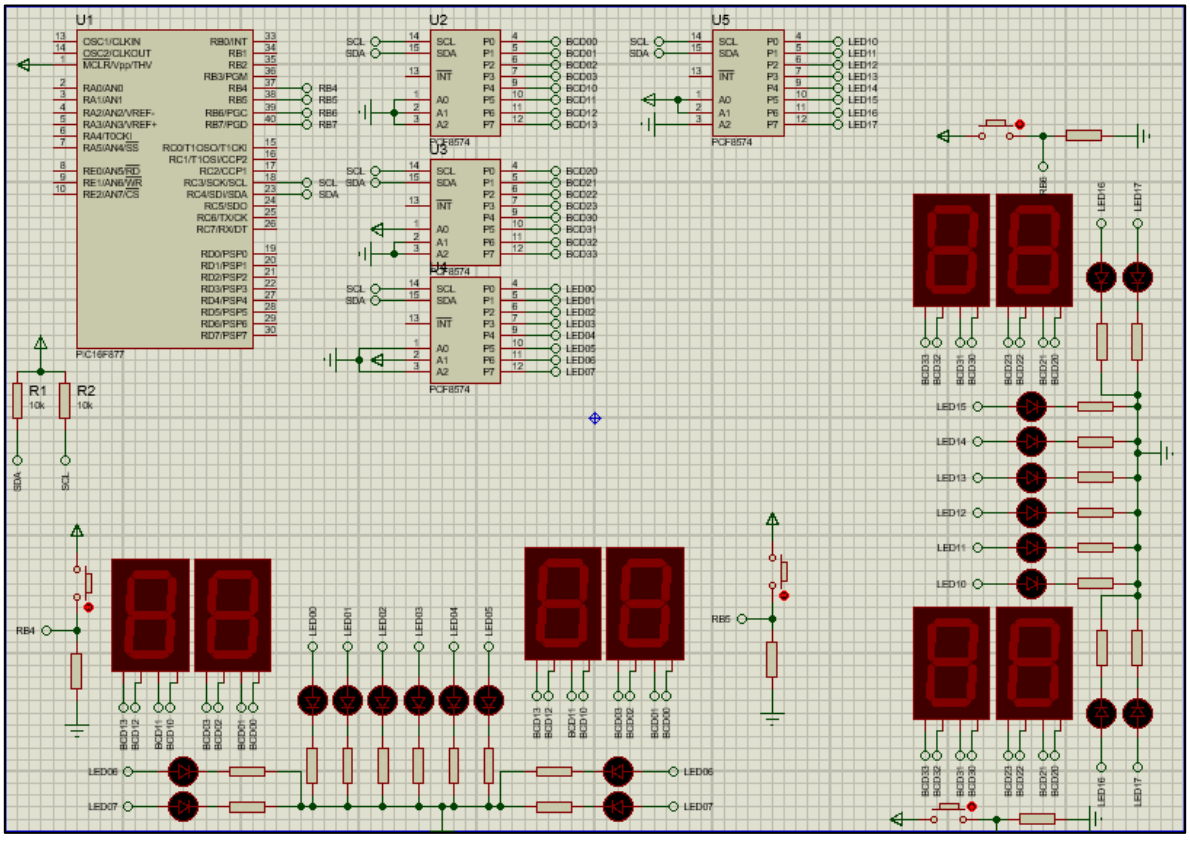
Construcción del circuito.

Los elementos empleados para la construcción de este proyecto fueron:

1) Microcontrolador PIC16F877
2) Expansor de puertos PCF8574
3) LEDs rojos, amarillos y verdes.
4) Display de siete segmentos BCD.
5) Push buttom.







El funcionamiento y explicación se encuentran en el siguiente enlace:

https://youtu.be/i-3A2FfOZrY

Conclusión:

Al momento de desarrollar este proyecto se implementaron los conocimientos adquiridos a lo largo del curso de microcomputadoras, permitiendo así desarrollar algo con una implementación un poco más práctica, esto debido a los requerimientos planteados por el profesor en las especificaciones de desarrollo.

Aunque se empleó el lenguaje C para el desarrollo de este, hubiera sido más interesante desarrollarlo en lenguaje ensamblador debido a que con este uno puede visualizar más detalles, como lo son los registros de configuración y e instrucciones más detalladas que permiten ver paso a paso lo que debe hacer el micro, pero desafortunadamente por cuestiones de tiempo y un temario un poco justo, sólo se pudo ver el tema de las interrupciones en lenguaje C, pero aún así aprender a programar un PIC en un lenguaje de alto nivel como C ya es un conocimiento muy grande el cual va sustentado de todo lo visto en teoría y practica con lenguaje ensamblador.