



Instituto de Estudios Universitarios

Ricardo García Reyes

Matrícula: 108580

Grupo: DS05

Ingeniería en Desarrollo de Software (En línea)

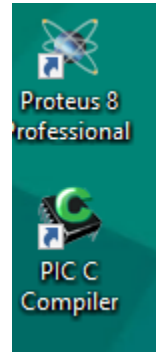
Materia: Programación y animación 3D

Mtro. Juan Manuel Amezcua Ortega

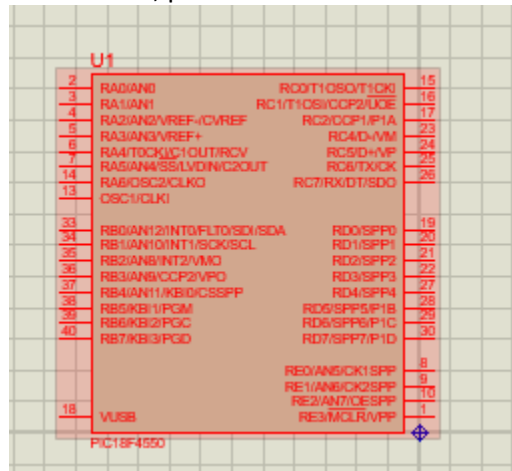
Actividad 4: Programación de un contador decimal

Saltillo, Coahuila 18 de Diciembre 2020

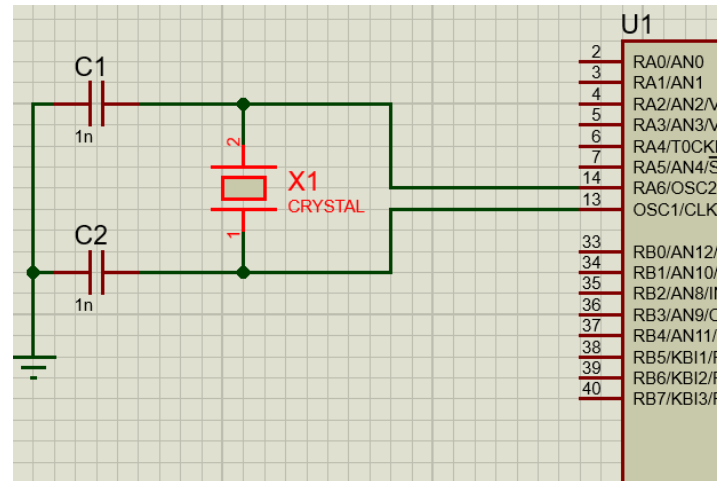
Para esta práctica, se pidió utilizar un pic de la familia 18f. Se decidió utilizar el software de simulación de proteus 8 y el software compilador de un archivo en lenguaje c llamado pic c compiler.



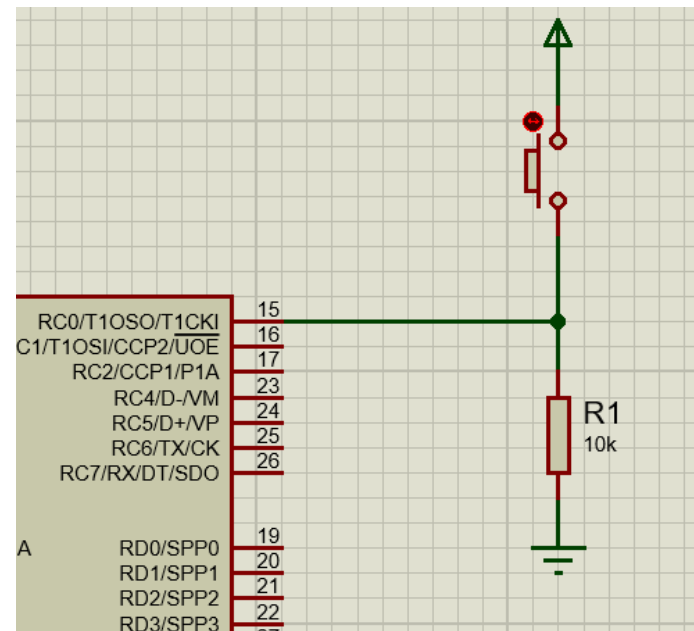
La primera acción realizada es abrir proteus e importar un PIC 18F4550, pues uno de la familia Pic 18f era requerido en la práctica.



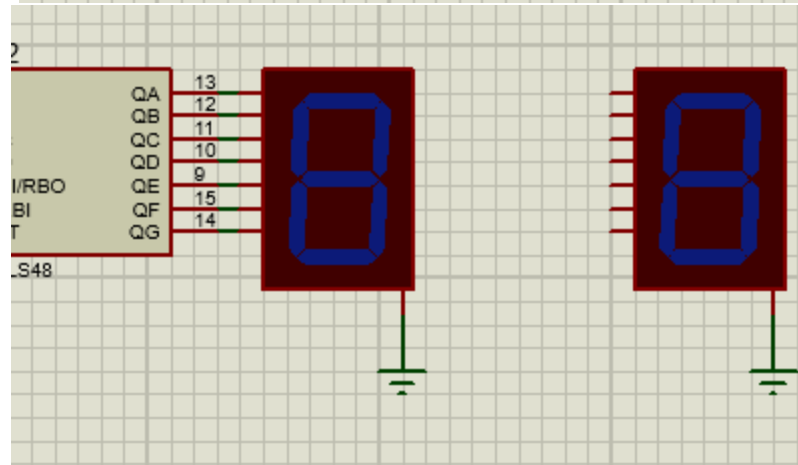
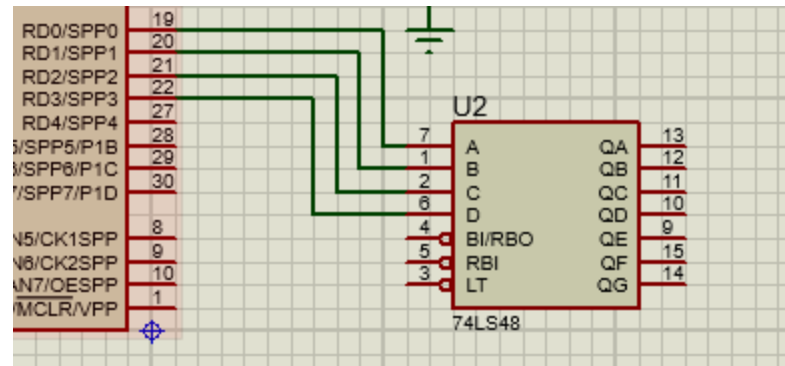
Se conectan a continuación el cristal y dos capacitores requeridos para el circuito.



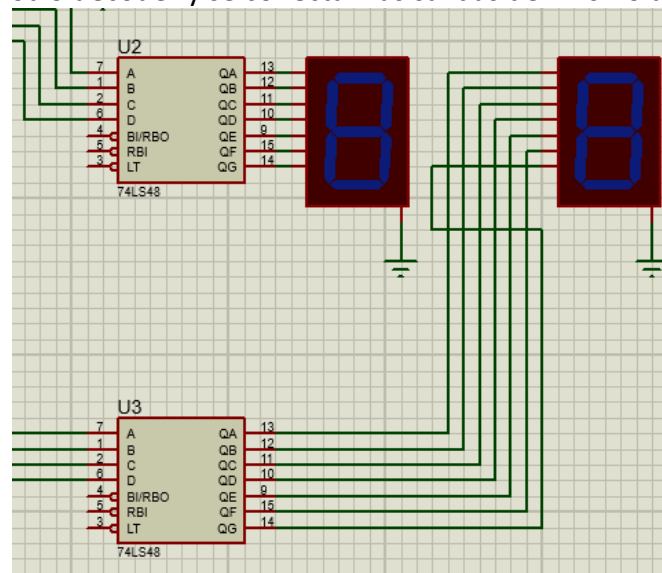
Después se conecta el push button que será requerido para activar el circuito del contador, pues la entrada de la señal será RC0.

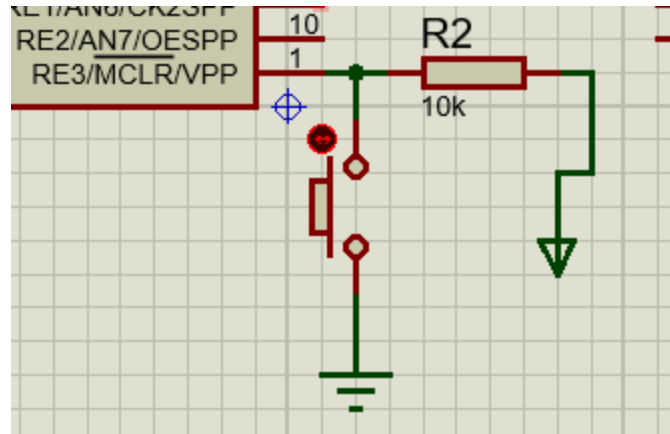


Se conectan los decoders correspondientes en la salida del pic para posteriormente conectarlos a la entrada de los displays de 7 segmentos.

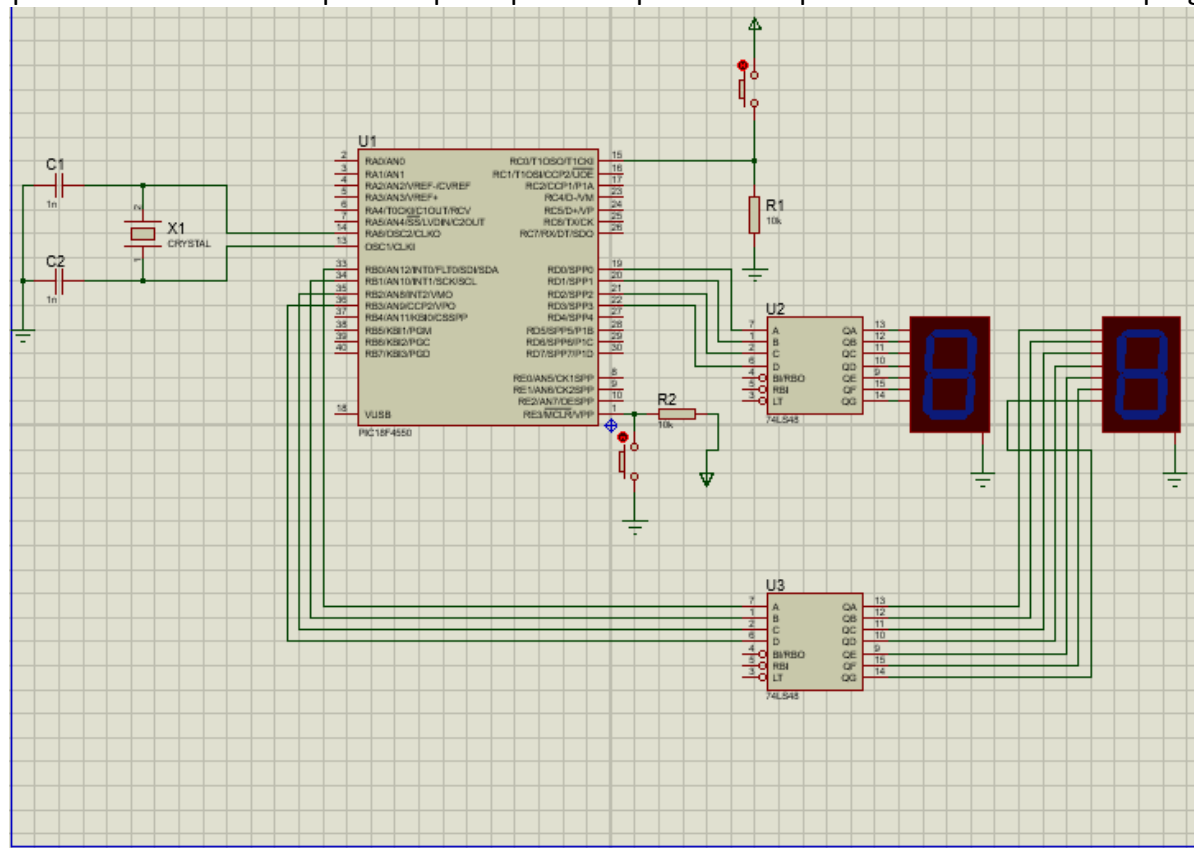


Se conecta otro decoder y se conectan las salidas del mismo a los displays.





Se conecta un botón que no se utiliza en esta práctica pero que hace que el botón pueda funcionar en modo programación o modo RE3.



Al final, el circuito queda de la siguiente manera.

programa_ricky.c

```

1  #include<18F4550.h>
2  #fuses NOMCLR INTRC_IO NOMCLR
3  #use delay(clock=4000000)
4  void main() {
5      int8 decenas,unidades;
6      set_tris_C(0x0F);
7      set_tris_D(0x00);
8      set_tris_B(0x00);
9      setup_oscillator(OSC_4MHZ); // oscilador interno
10     decenas = 0;
11     unidades =0;
12     output_d(0);
13     output_b(0);
14     while(TRUE) {
15         if(input(PIN_C0)==1) {
16             output_b(unidades);
17             output_d(decenas);
18             unidades=unidades+1;
19             if(unidades==10) {
20                 unidades=0;

```

programa_ricky.c










```

13     output_b(0);
14     while(TRUE) {
15         if(input(PIN_C0)==1) {
16             output_b(unidades);
17             output_d(decenas);
18             unidades=unidades+1;
19             if(unidades==10) {
20                 unidades=0;
21                 decenas=decenas+1;
22                 if(decenas==10) {
23                     decenas=0;
24                     }//decenas
25                 }//unidades
26             }//PIN
27             delay_ms(200);
28         }//while
29     }//main

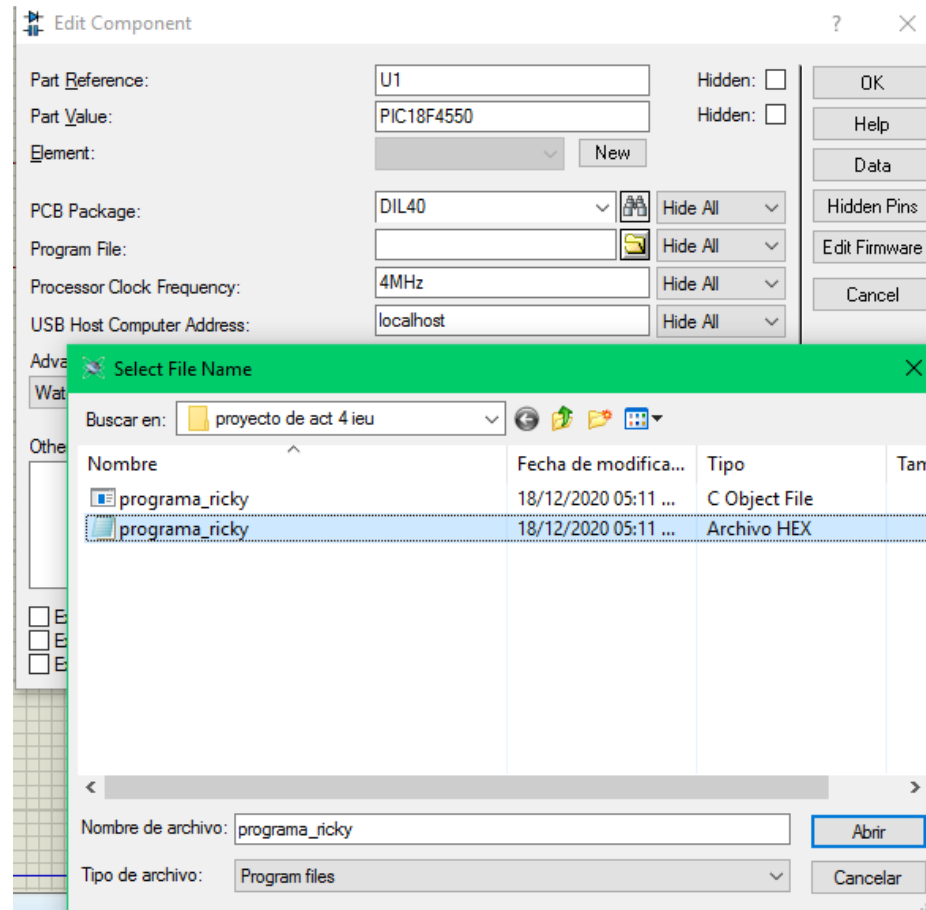
```

Ahora tenemos un programa en lenguaje c. En el cual se declaran los puertos a usar y se entra en una lógica en la cual cuando el pin_c0 esté activo, se corre un ciclo en el que van aumentando las unidades. Cuando estas llegan a 10, entonces se aumentan las decenas hasta que las decenas sean 10, entonces ambos contadores se resetean.

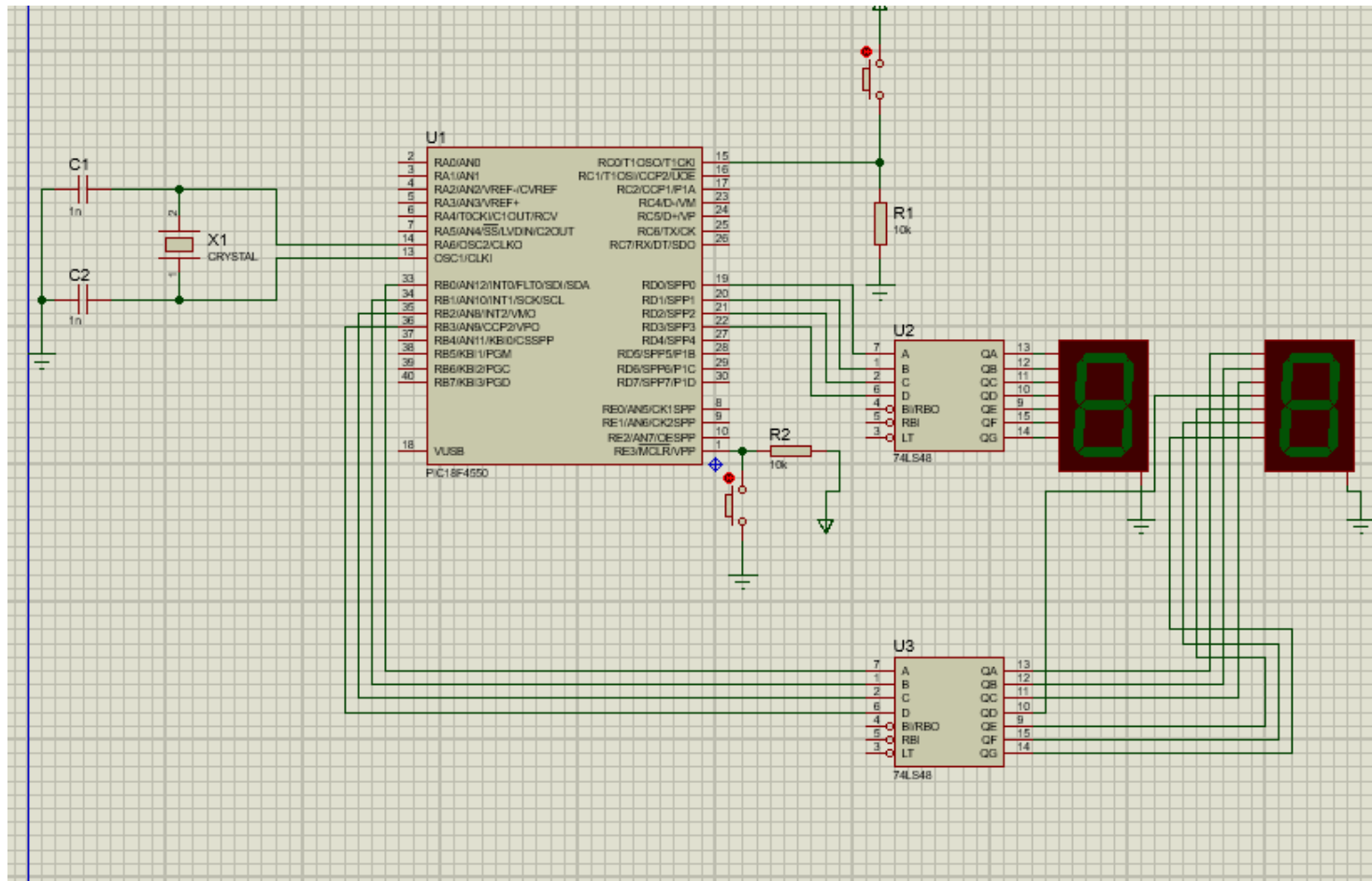
Una vez hecho el programa en c, utilizaremos el css compiler para transformar ese archivo en c a .hex, el cual luego grabaremos en la pic.

Nombre	Fecha de modifica...	Tipo	Tamaño
 programa_ricky	18/12/2020 05:10 ...	C Source File	1 KB
 programa_ricky	18/12/2020 05:11 ...	C Object File	16 KB
 programa_ricky.err	18/12/2020 05:11 ...	Archivo ERR	1 KB
 programa_ricky.esym	18/12/2020 05:11 ...	Archivo ESYM	14 KB
 programa_ricky	18/12/2020 05:11 ...	Archivo HEX	1 KB
 programa_ricky	18/12/2020 05:11 ...	C Output File	4 KB
 programa_ricky	18/12/2020 05:11 ...	C Output File	2 KB
 programa_ricky	18/12/2020 05:11 ...	C Output File	2 KB
 programa_ricky	18/12/2020 05:11 ...	C Output File	1 KB

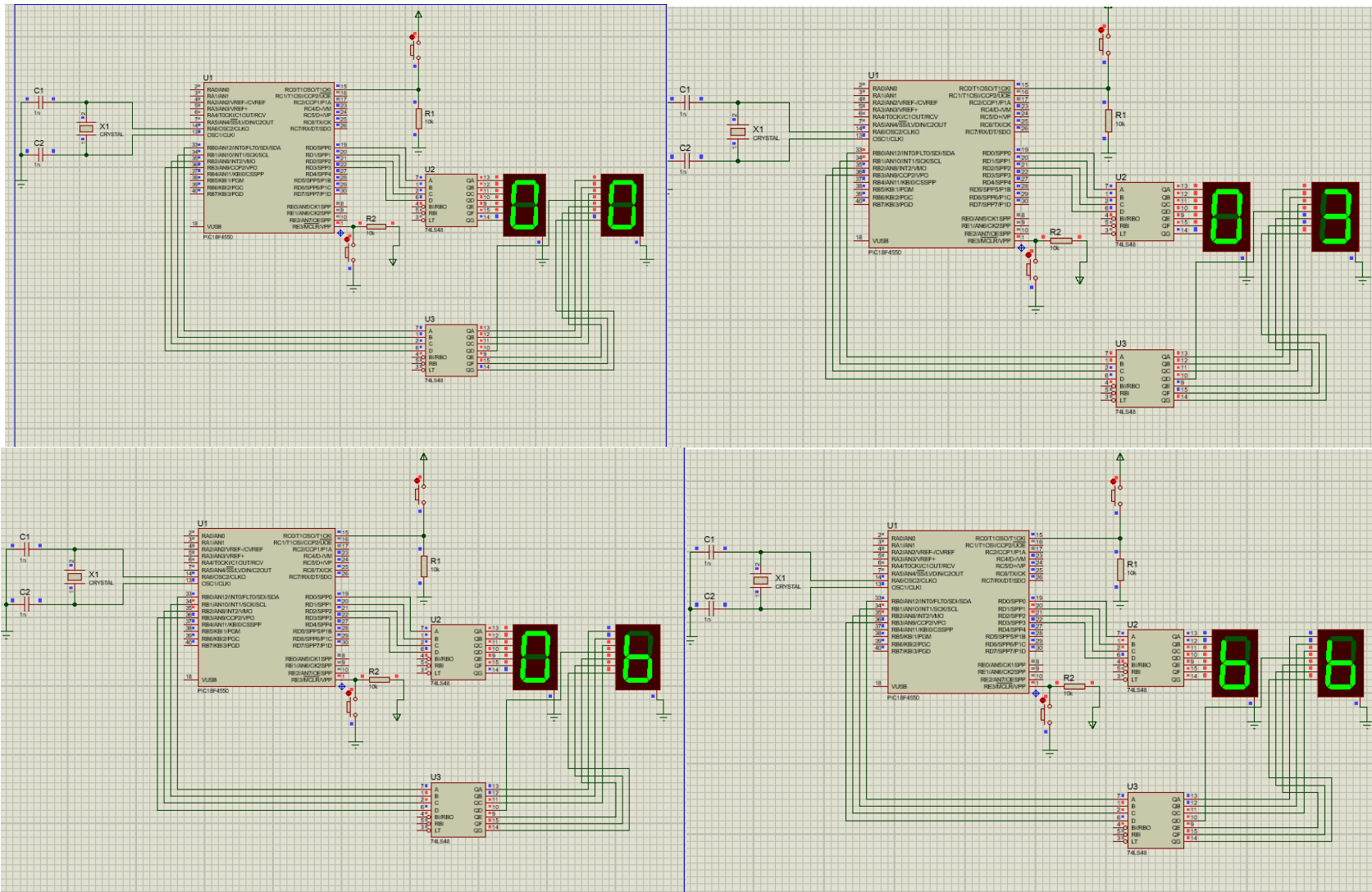
Cuando el circuito está listo, hacemos click derecho sobre la pic y cambiamos el archivo del programa.



Finalmente, al correr la simulación, me di cuenta que los displays azules no marcaban tanto el número y decidí cambiarlos a verde.



A continuación se observan los números cambiando cada que el botón se pulsa.



Conclusión: Como conclusión, se puede decir que una pic es un controlador fácil de manipular, además de que se puede cargar código de diferentes lenguajes como c y ensamblador. Es necesario tener un buen software simulador y familiarizarse con él porque son muchas cosas las que pueden fallar en el proceso de la práctica.

Referencias:

- Ing. Mecánica Eléctrica y Sistemas Computacionales (youtube). (2016, 30 septiembre). Contador 0 a 99 PIC 18F4550 CCS C COMPILER Isis Proteus Display 7 Segmentos. Recuperado 18 de diciembre de 2020, de <https://www.youtube.com/watch?v=aI7OmlN-doM&t=45s>