



Nombre de la práctica	Realizar un contador tanto en un simulador como en físico			No.	1
Asignatura:	Arquitectura de computadoras	Carrera:	Ingeniería en Sistemas Computacionales	Duración de la práctica (Hrs)	6 horas

NOMBRE DEL ALUMNO: Alexis Uriel Martínez Cruz

GRUPO: 3012

I. Competencia(s) específica(s):

II. Lugar de realización de la práctica (laboratorio, taller, aula u otro):

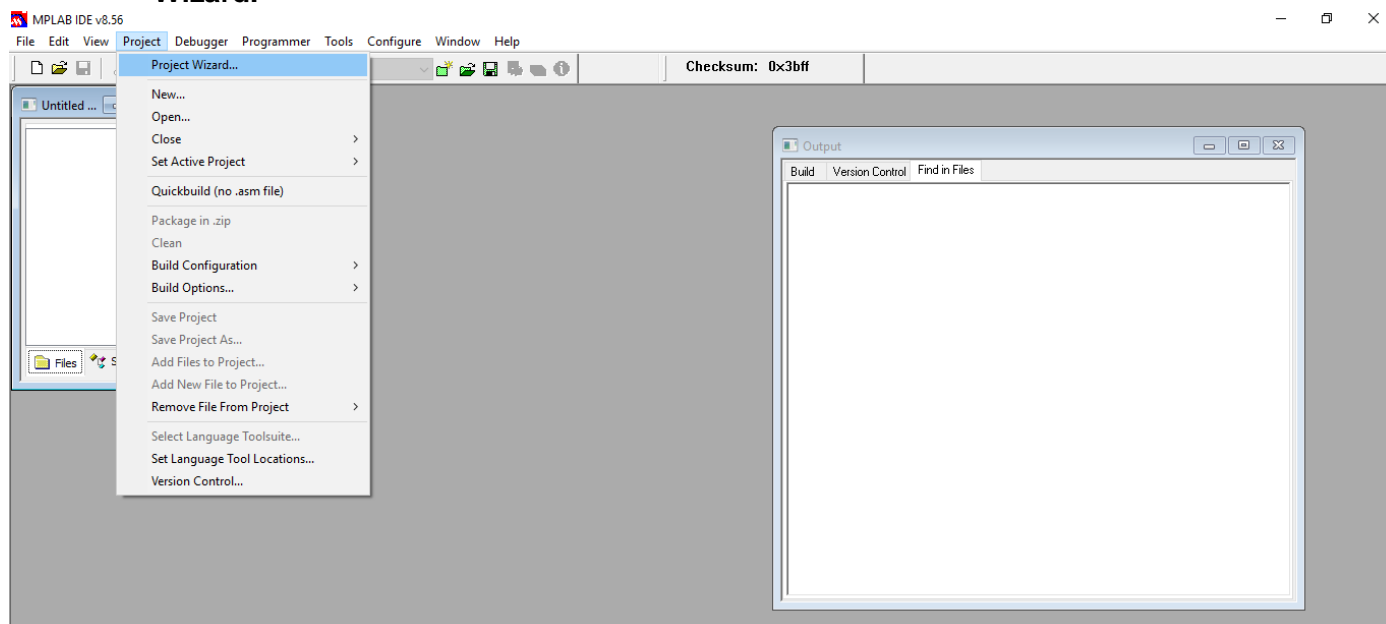
Aula de clases

III. Material empleado:

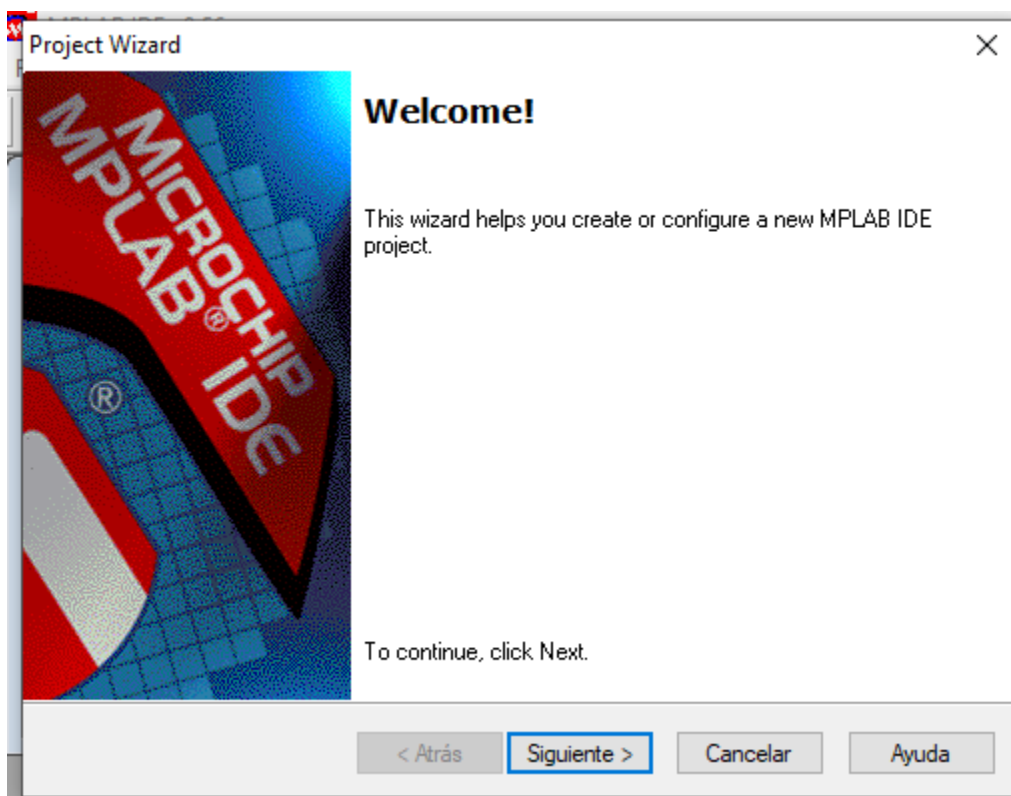
1. Tabla protoboard
2. PIC 16F84A
3. Cable para protoboard
4. 2 capacitores cerámicos de 22 picofaradios
5. 1 resistencia
6. 1 cátodo de 7 segmentos
7. 1 cristal oscilador de 4 MHz
8. Programador de PIC'S, software del mismo y computadora
9. Un cargador

IV. Desarrollo de la práctica:

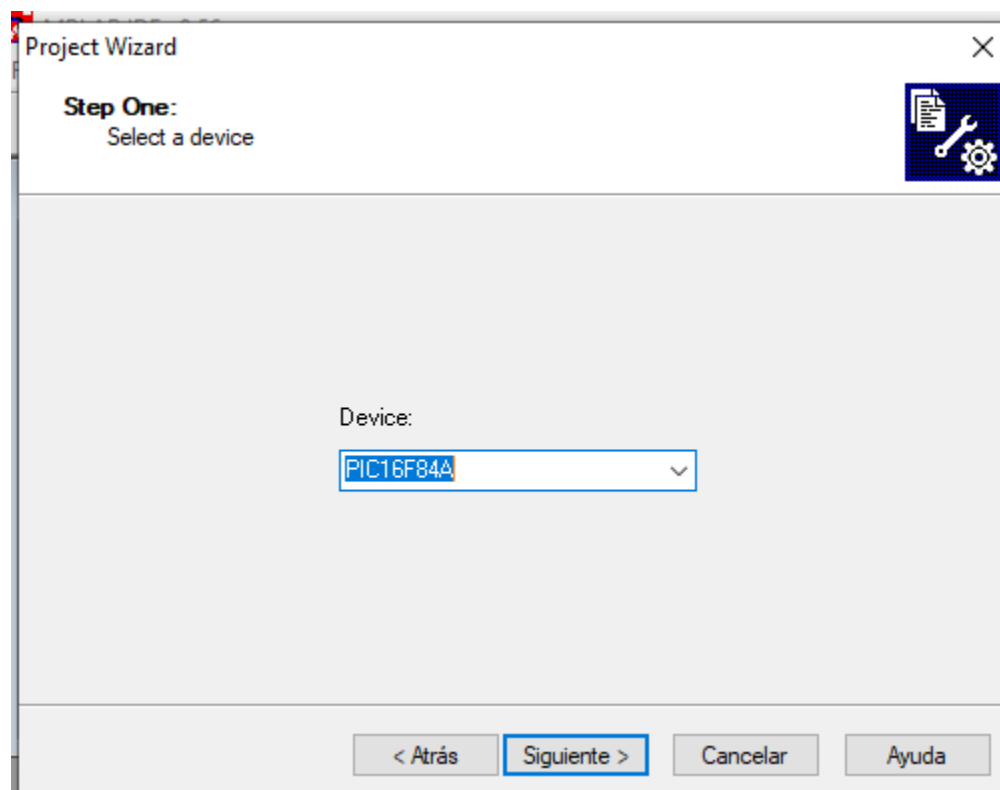
1. Crear el código que requerimos para desarrollar el contador para esto usamos la aplicación de MPLAB.
2. Ingresamos a la aplicación, buscamos en la barra de tareas **Project** y seleccionamos **Project Wizard**.



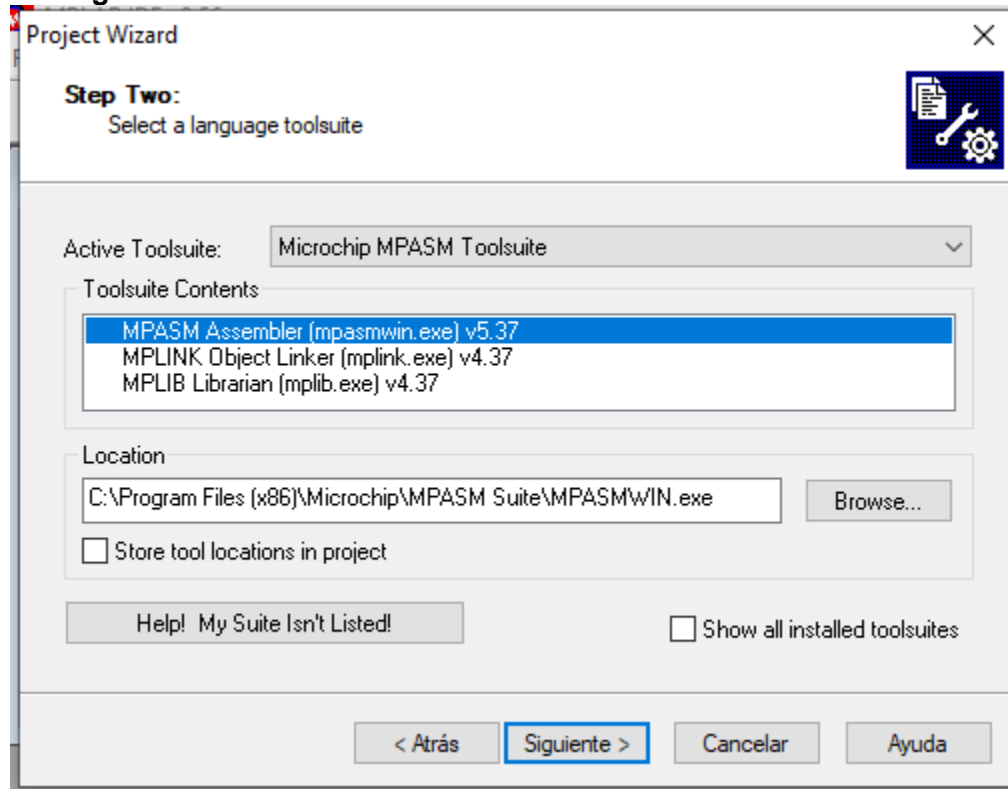
3. Saldrá un cuadro de dialogo y damos **siguiente**



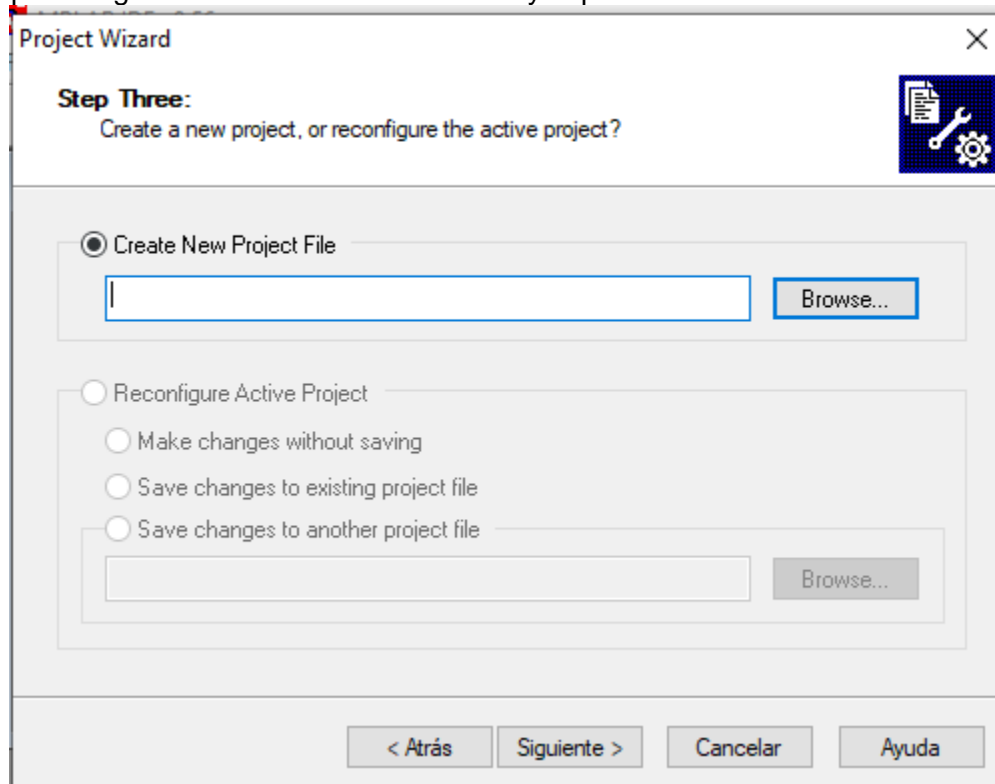
4. Seleccionamos el pic que vamos a utilizar en este caso fue el **PIC16F84A**



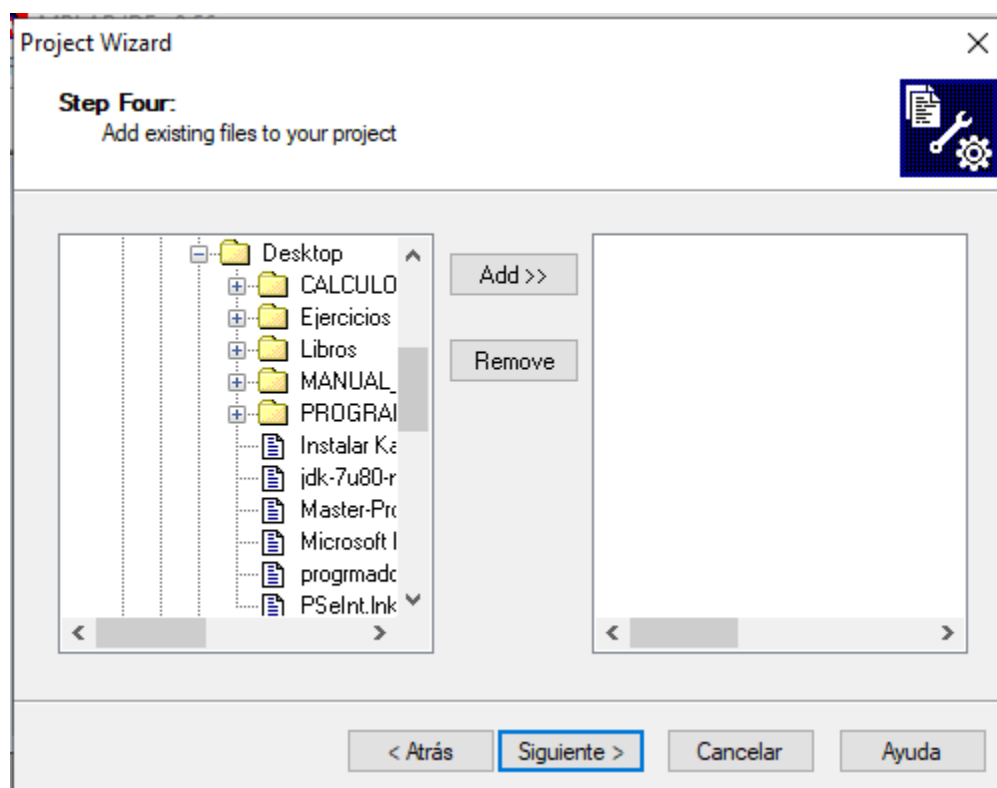
5. Damos siguiente



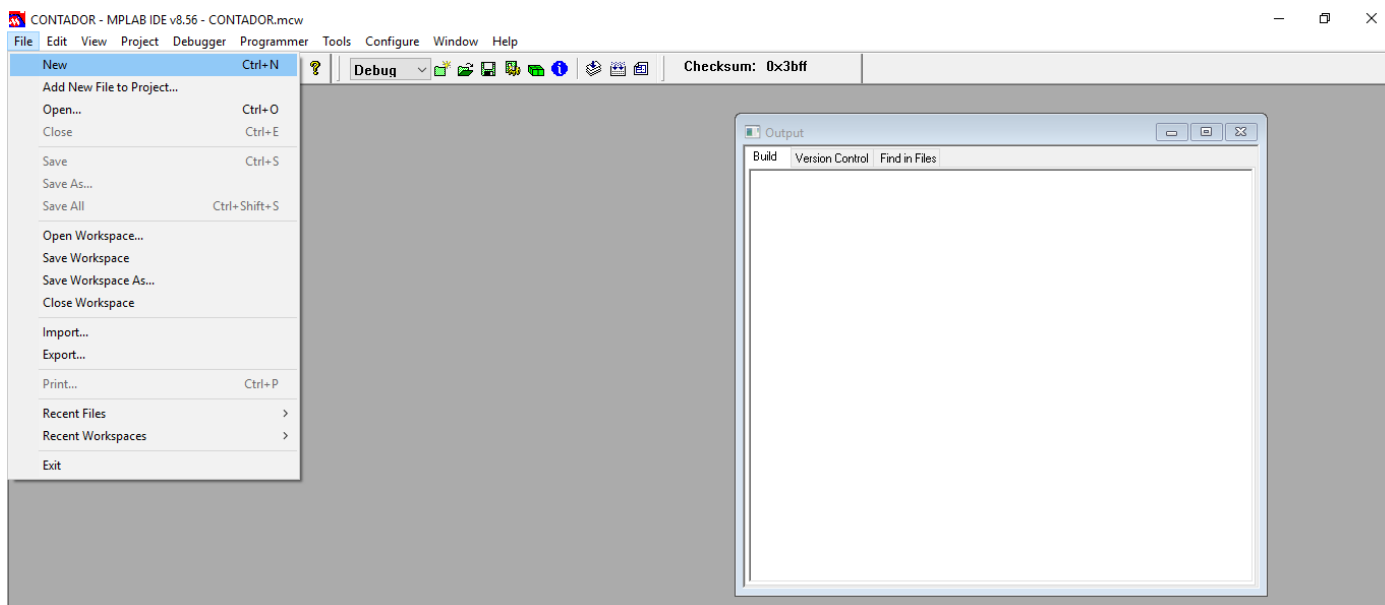
6. Después lo guardaremos donde deseemos y le pondremos un nombre



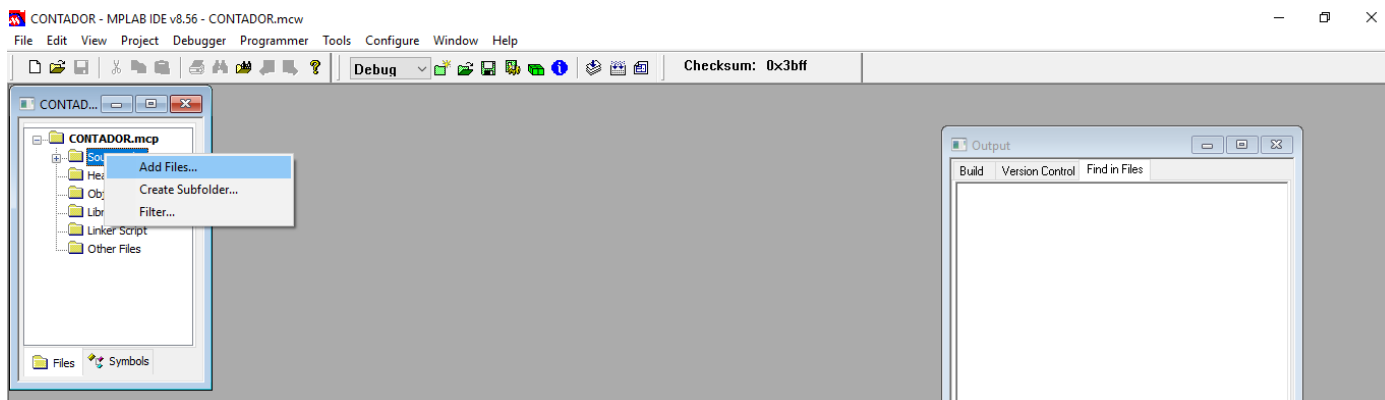
7. Dar clic en siguiente y por último en finalizar



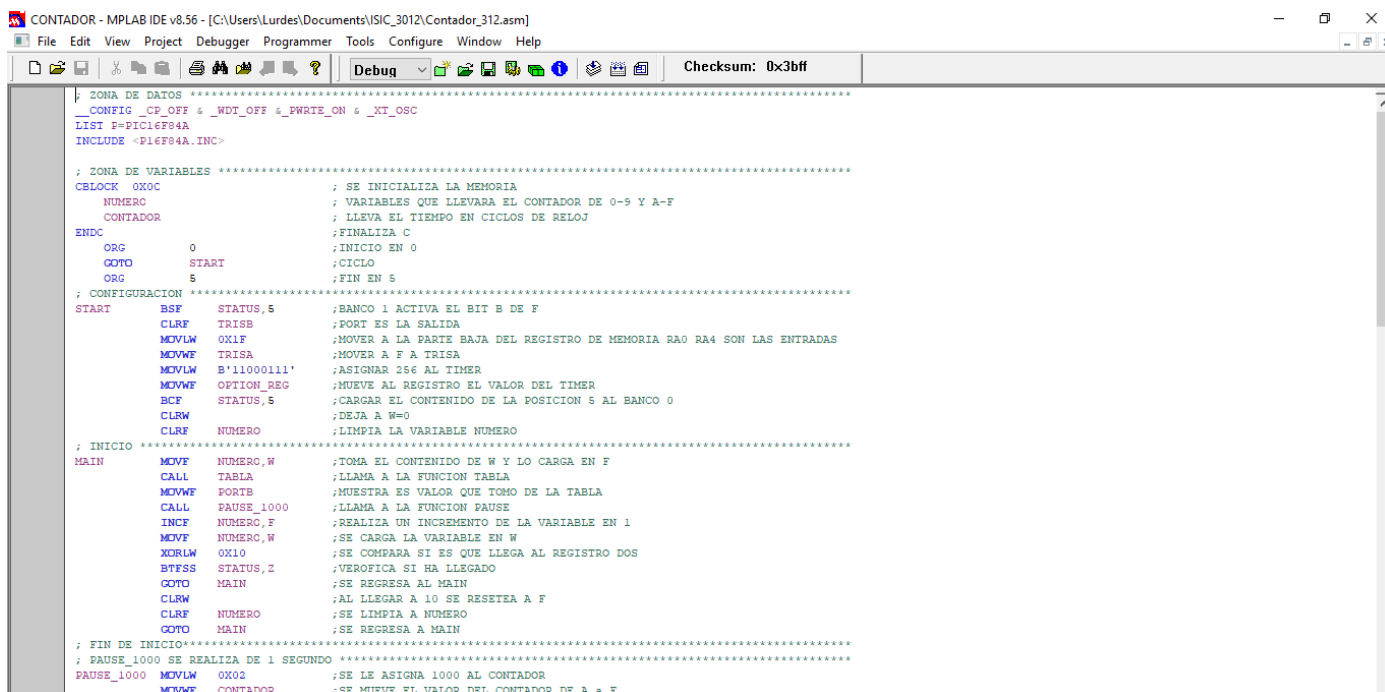
8. Después en la barra buscamos **File y New**



9. Después guardamos con el nombre que desees y con la terminación **.asm** de ensamblador y lo añadimos al proyecto que estamos realizando.



10. Después solo queda escribir el código que es el siguiente, cada línea esta especificada de cual es su función.





```

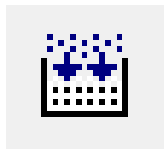
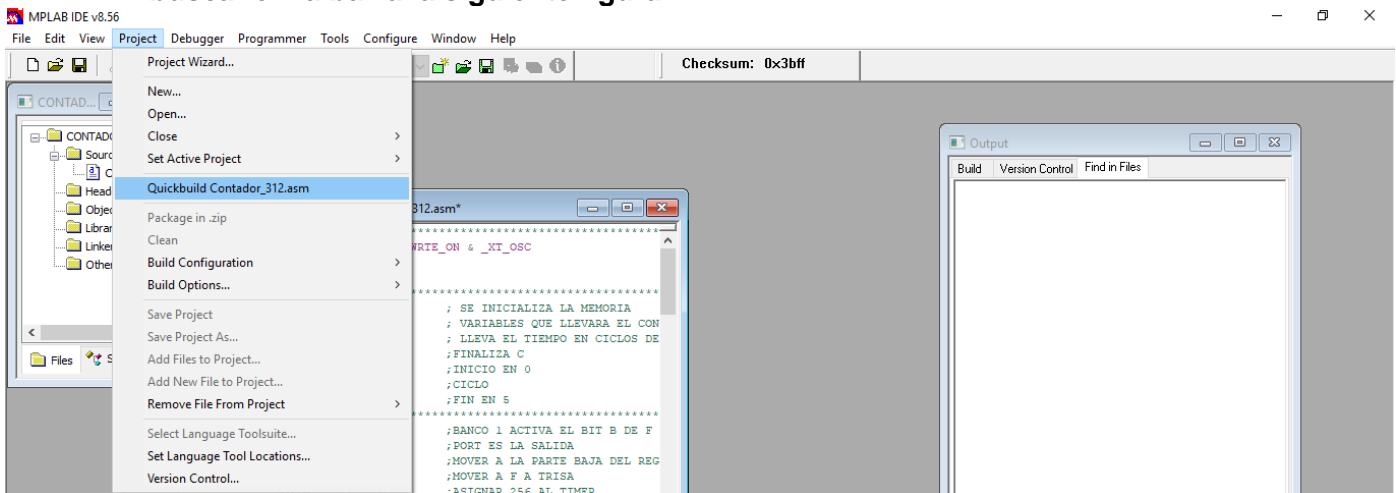
DELAY      BCF      INTCON,TOIF      ;SE LIBERA EL BIT DE DESBORDAMIENTO TMR0
           MOVLW    09                ;SE CARGA EL 217 A W
           MOVWF    TMR0              ;A TMR0
DELAY2     BTFSS    INTCON,TOIF      ;SE LIBERA EL BIT DE DESBORDAMIENTO DEL TMR0
           GOTO     DELAY2            ;BUCLE DEL DELAY2
           DECFSZ    CONTADOR,F        ;DECREMENTA EN 1 EL CONTADOR
           GOTO     DELAY              ;BUCLE EN DELAY
           RETURN                      ;REGRESA

; TABLA *****
TABLA      ADDWF    PCL,F              ;SE INICIALIZA LA FUNCION TABLA CON EL CONTENIDO DE F
           RETLW    B'00111111'        ;0 EN EL CATODO DE 7 SEGMENTOS
           RETLW    B'00000110'        ; 1
           RETLW    B'01011011'        ; 2
           RETLW    B'01001111'        ; 3
           RETLW    B'01100110'        ; 4
           RETLW    B'01101101'        ; 5
           RETLW    B'01111101'        ; 6
           RETLW    B'01000111'        ; 7
           RETLW    B'01111111'        ; 8
           RETLW    B'01100111'        ; 9
           RETLW    B'01110111'        ; A
           RETLW    B'01111100'        ; b
           RETLW    B'00111001'        ; C
           RETLW    B'01011110'        ; d
           RETLW    B'01111001'        ; E
           RETLW    B'01110001'        ; F

END

```

11. Posteriormente procederemos a copiarlo para ver si fue escrito correctamente, para esto hay dos formas, la primera es ir a **Project** y en **quickbuild + nombre del código** o la segunda es **buscar en la barra la siguiente figura**:



Y tendrá que aparecer de la siguiente manera sino hay que corregir cada error que tengamos

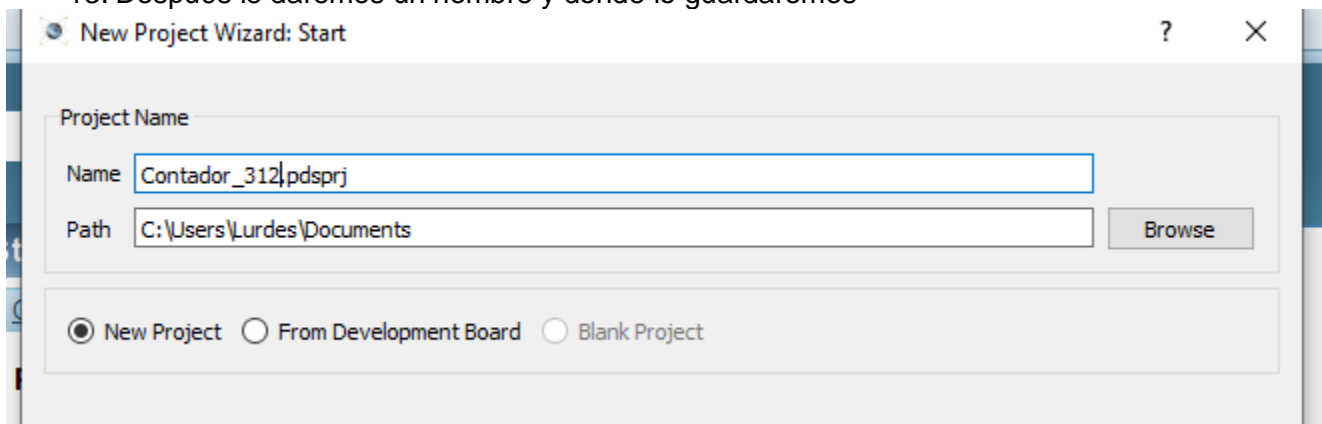
BUILD SUCCEEDED

Aquí terminarías de usar MPLAB y continuamos a usar Proteus

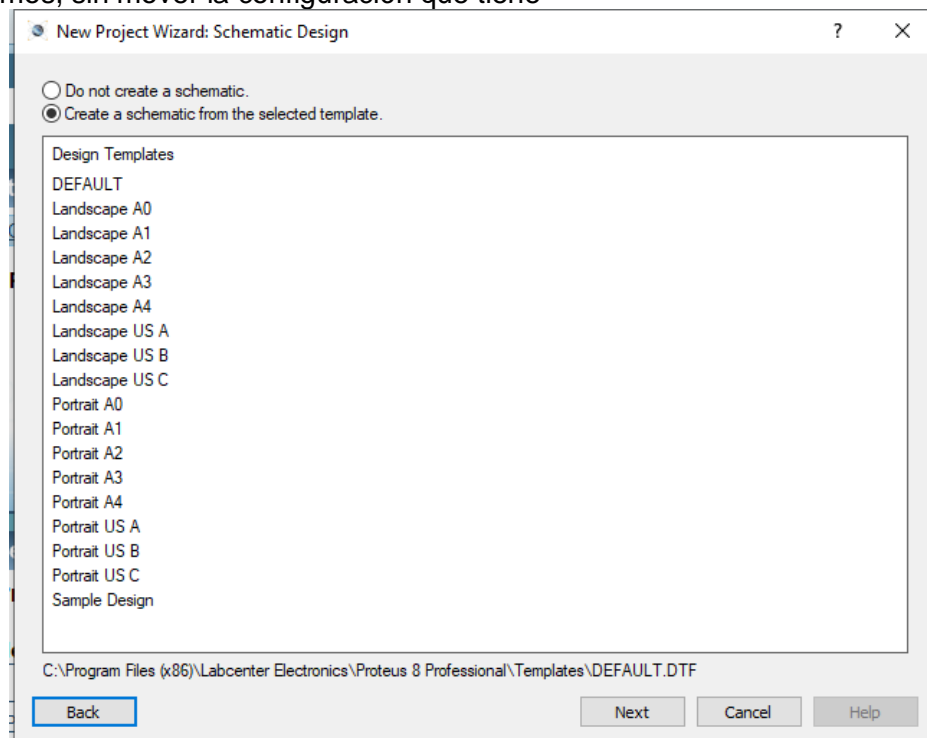
12. Primero en la barra buscar file y new Project



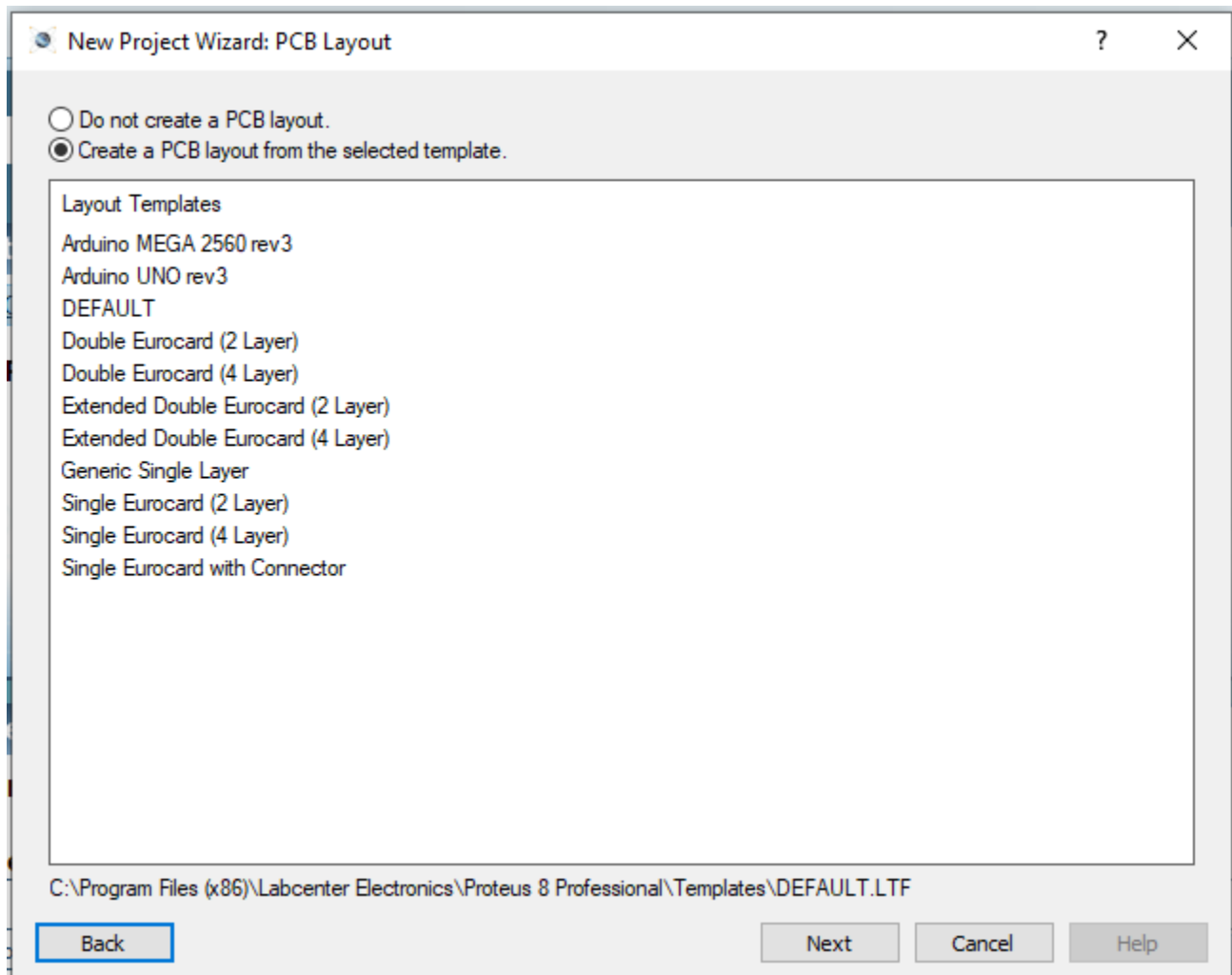
13. Después le daremos un nombre y donde lo guardaremos



14. Seguimos, sin mover la configuración que tiene



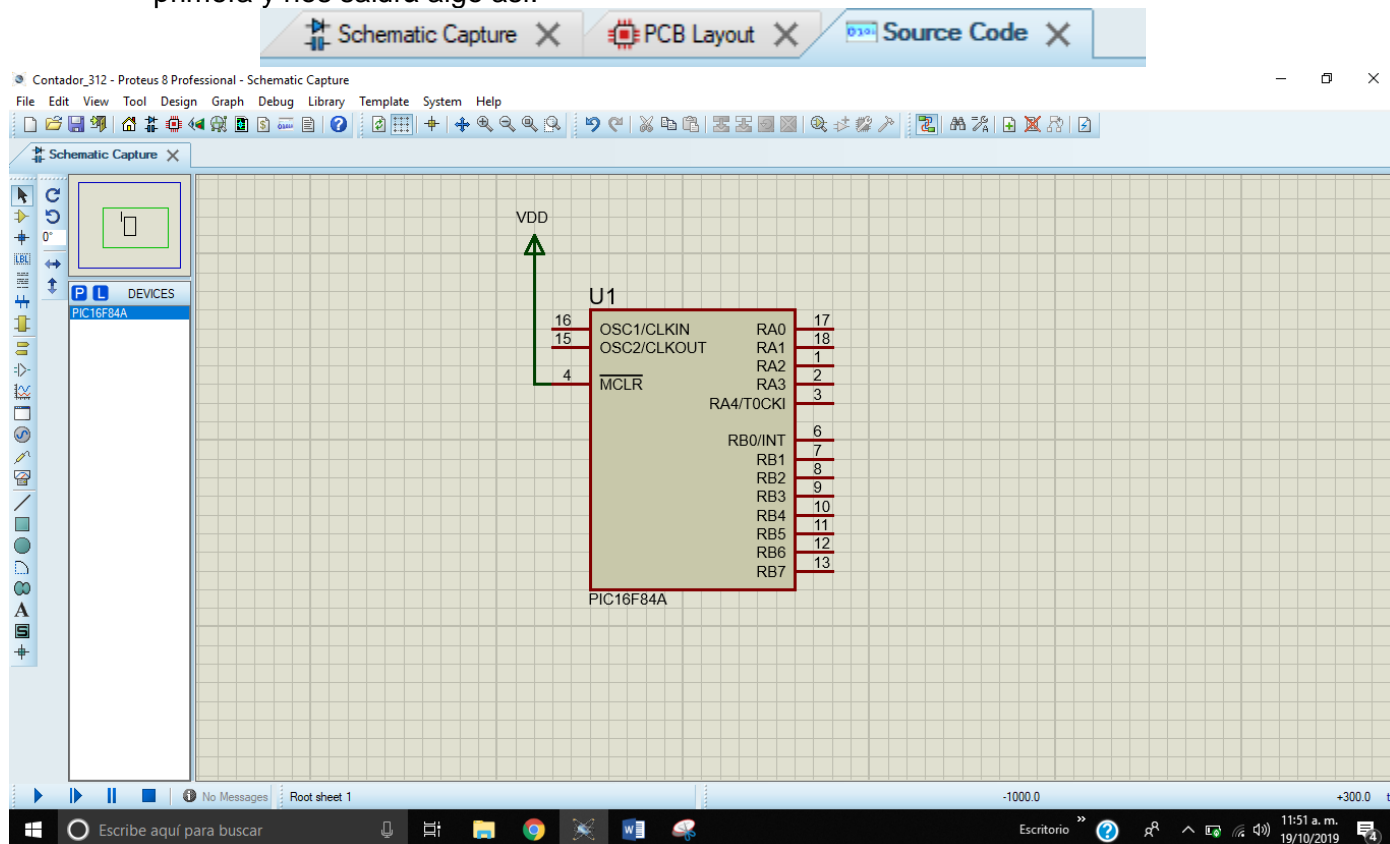
15. Cambiamos a la segunda opción de crear y por ultimo continuar



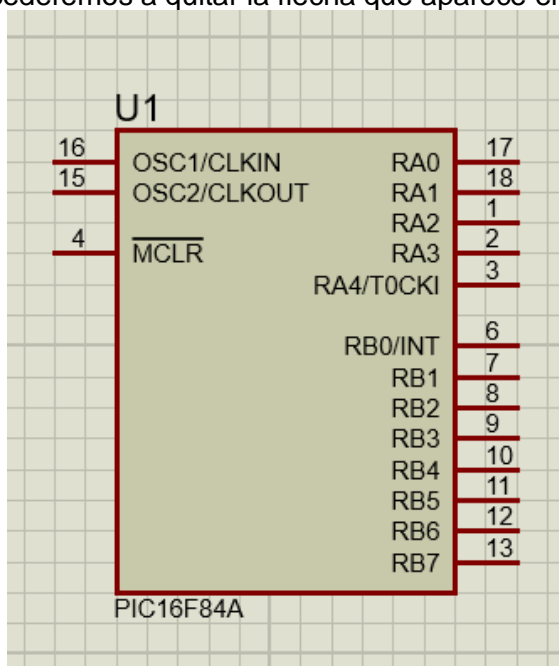
16. Después seleccionamos la familia del pic y el controlador que vamos a usar en este caso es la familia **PIC16** y el controlador es **PIC16F84A**.



17. Nos aparecerán 3 pestañas las cuales quietaremos las dos últimas y nos quedaremos con la primera y nos saldrá algo así.

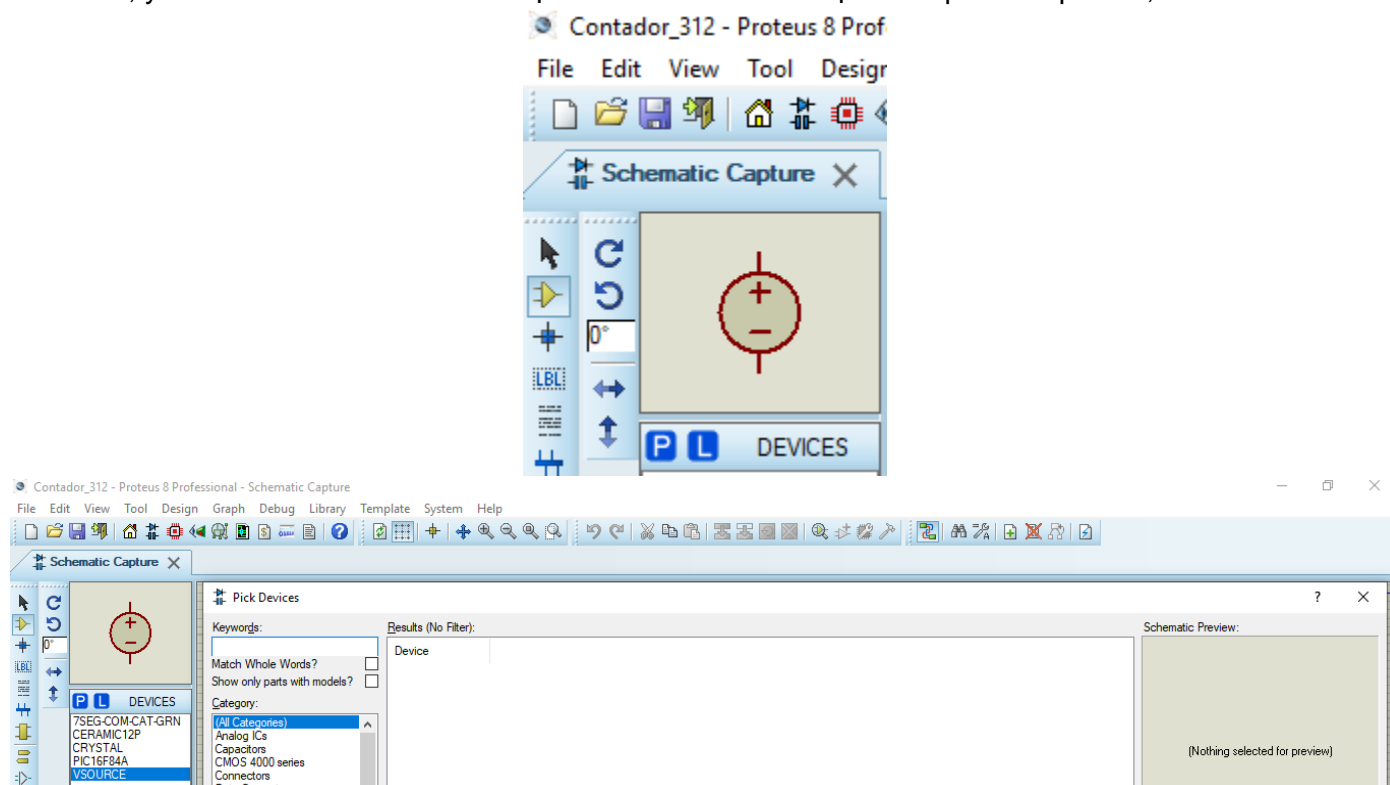


18. A continuación, procederemos a quitar la flecha que aparece en el microcontrolador



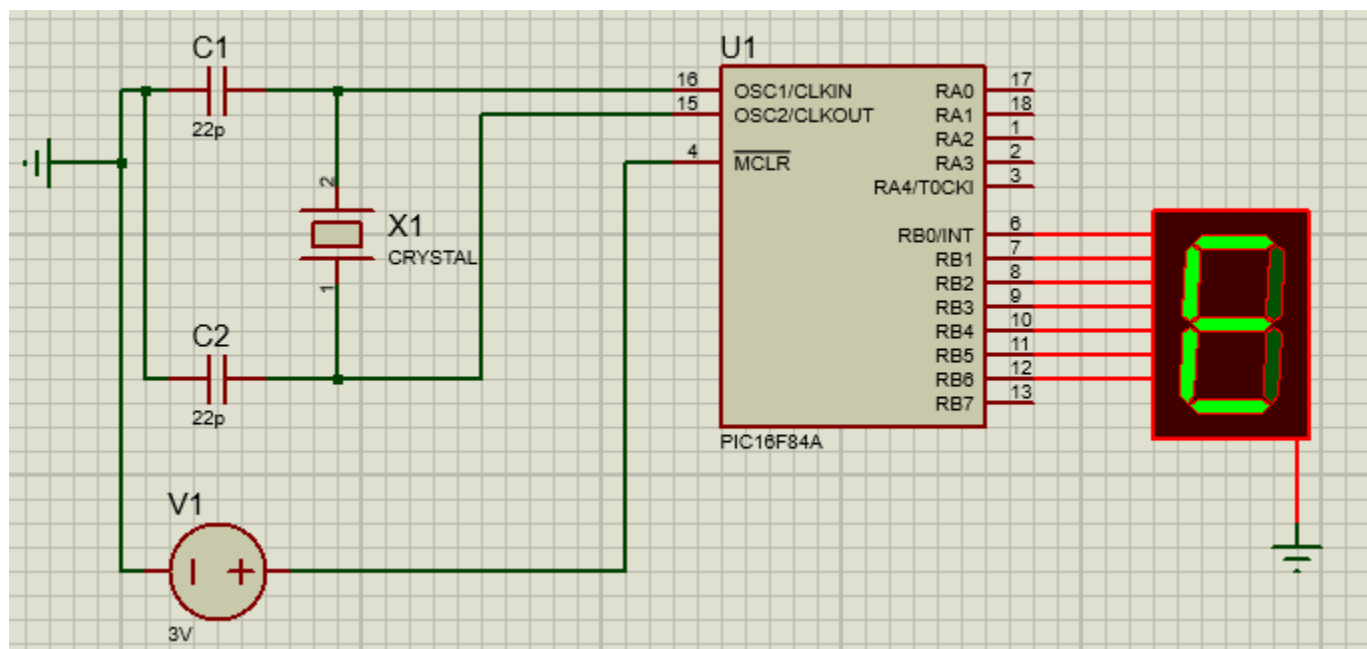
19. Después le tenemos que agregar los siguientes componentes:

Dos tierras, una pila o una fuente, un cátodo de 7 segmentos, dos capacitores cerámicos y un cristal oscilador, y conectarlos al PIC estos los podrás encontrar en la parte superior izquierda, donde sale una **P**



En esta parte es donde buscaremos nuestros componentes

20. Tendremos un esquema conectado de la siguiente manera



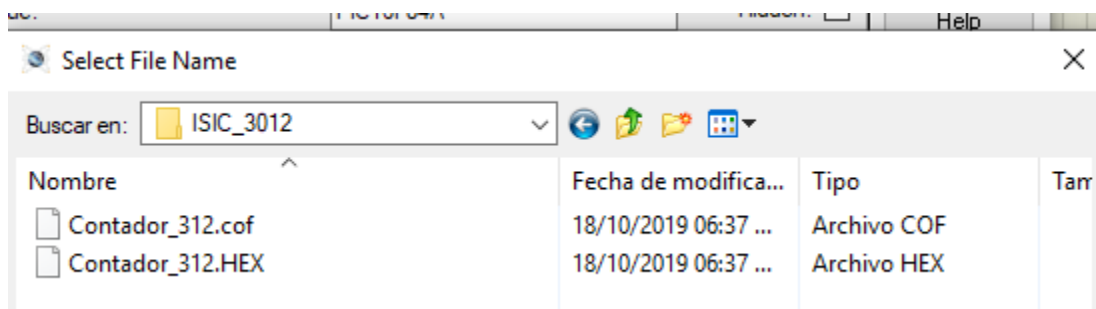
21. Tendremos que cambiar las propiedades del cristal, de los capacitores y de la pila
A los capacitores le cambiaremos que sean de 22 picofaradios:

Al crystal le cambiaremos que sea de 5 MHz

Y a la pila la cambiaremos a 3 V

Para cambiar esto solo damos doble clic en el componente

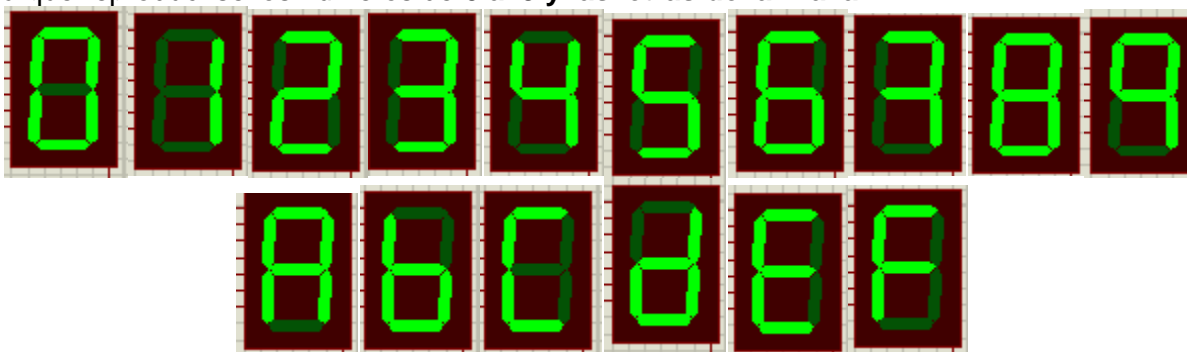
22. Una vez hecho esto daremos doble clic en el PIC y en **program file**, damos clic en la carpeta **y buscamos el archivo con extensión. HEX** el cual se creó cuando copilamos el código en **MPLAB**.



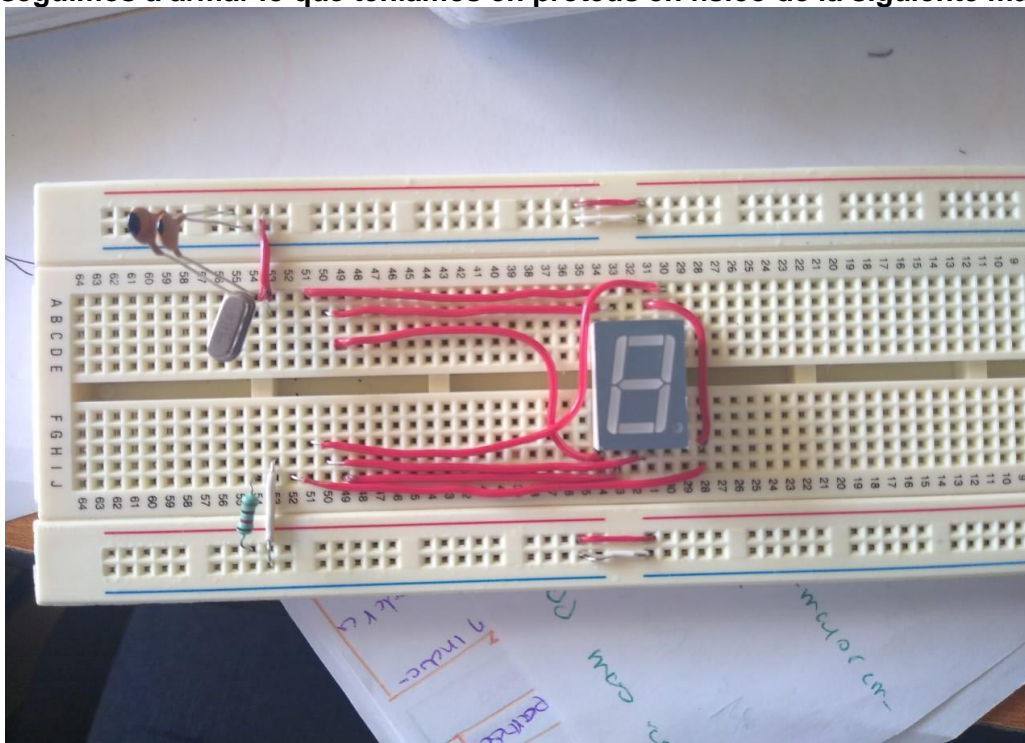
Una vez cargado en la parte inferior izquierda aparece un símbolo de reproducción daremos clic para verificar si funciona nuestro circuito que hemos creado.

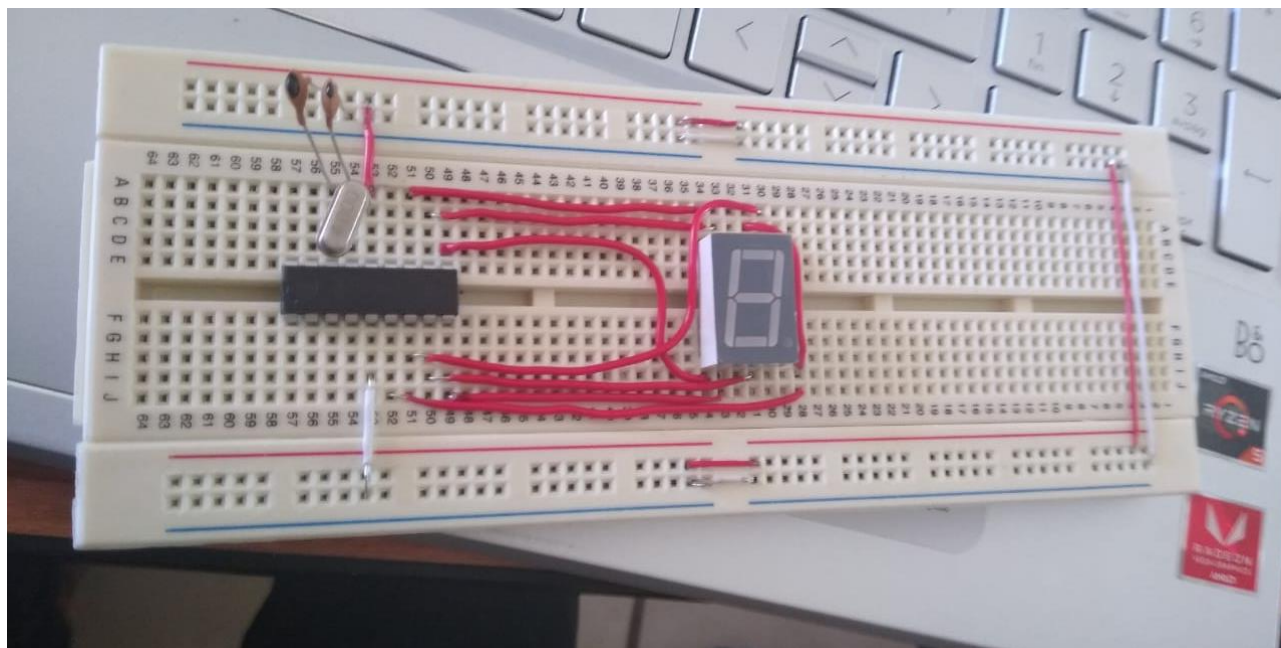


Y tendrá que reproducirse los números de 0 al 9 y las letras de la A a la F

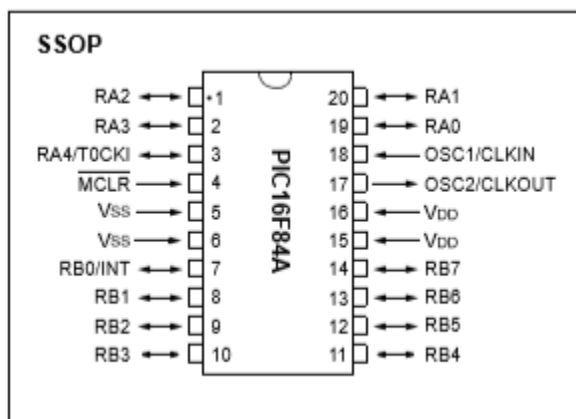
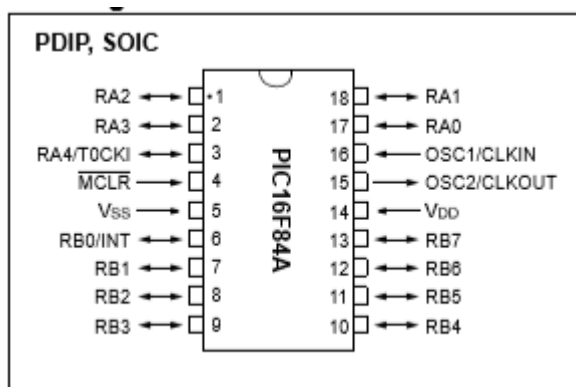


Proseguimos a armar lo que teníamos en proteus en físico de la siguiente manera

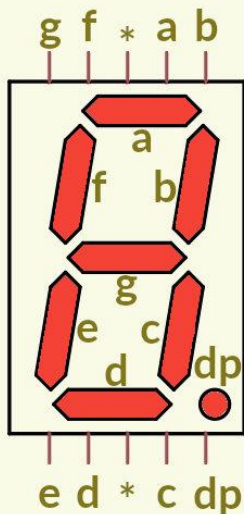




Para hacer esto ya saber dónde conectar cada cable requerimos de dos cosas muy importantes el data sheet del controlador y del display o catodo



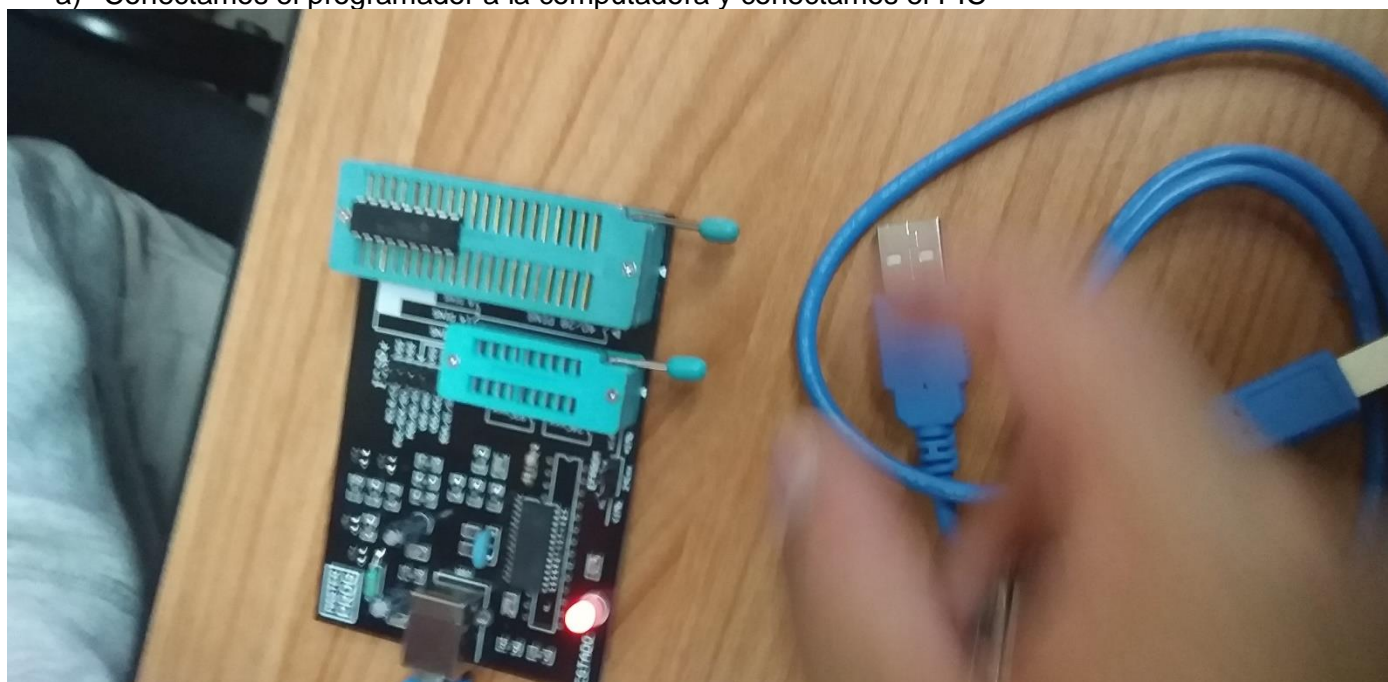
Display 7 segmentos cátodo común



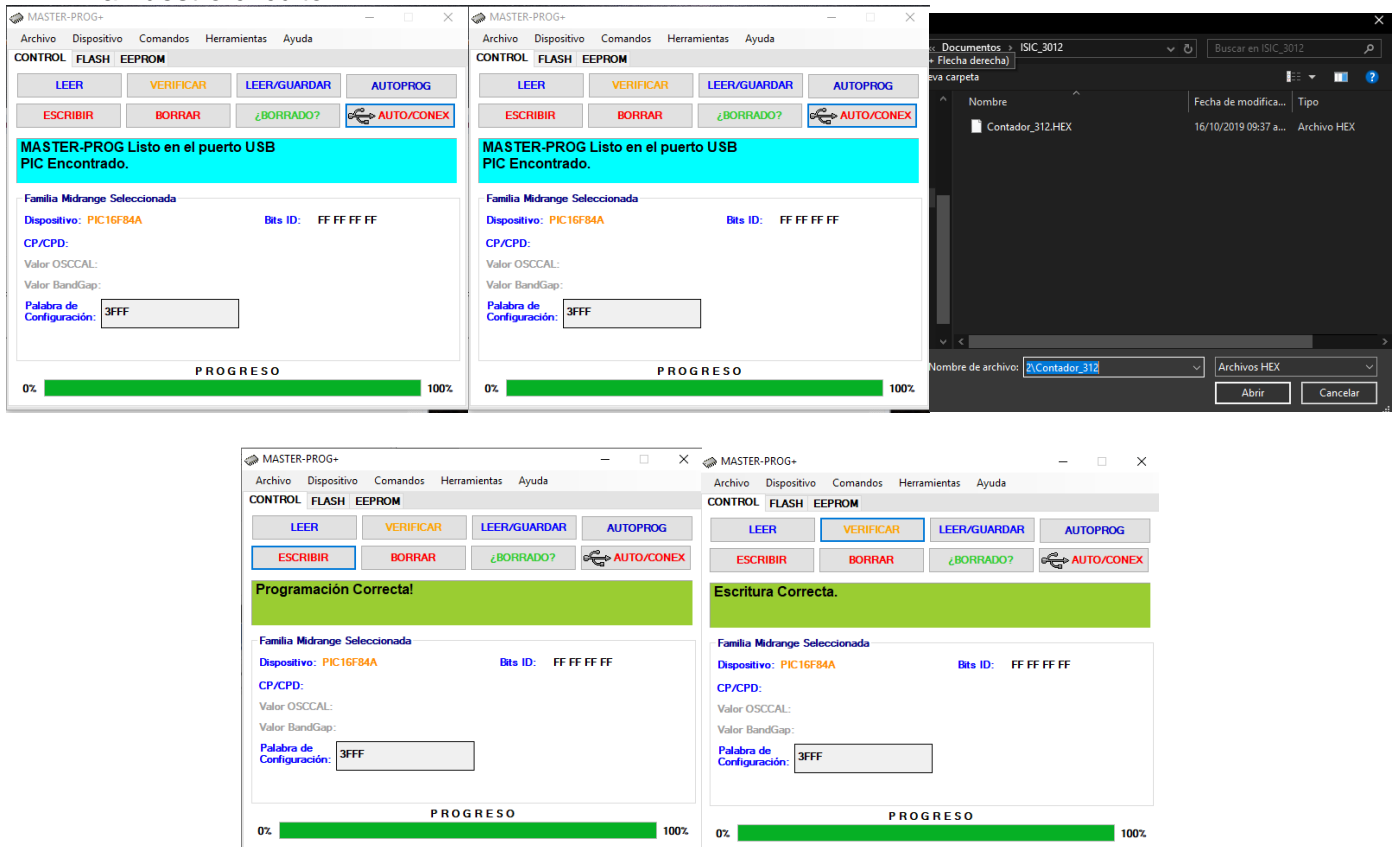
		Cátodo Cómun						
Común*	Número	g	f	e	d	c	b	a
GND	0	0	1	1	1	1	1	1
GND	1	0	0	0	0	1	1	0
GND	2	1	0	1	1	0	1	1
GND	3	1	0	0	1	1	1	1
GND	4	1	1	0	0	1	1	0
GND	5	1	1	0	1	1	0	1
GND	6	1	1	1	1	1	0	1
GND	7	0	0	0	0	1	1	1
GND	8	1	1	1	1	1	1	1
GND	9	1	1	0	1	1	1	1

Una vez armado nuestro circuito pasaremos a programar nuestro microcontrolador
Para esto utilizamos un programador de PIC'S Y SU RESPECTIVO SOFTWARE:

- Conectamos el programador a la computadora y conectamos el PIC



- b) Abrimos el software y damos clic en **AUTO/CONEX**, una vez conectado y reconocido el pic, seguiremos a dar clic en **LEER**, después cargaremos el archivo con la extensión **.HEX** después damos clic en **ESCRIBIR** y por último en **VERIFICAR** y listo quitamos el pic y podemos conectarlo a nuestro circuito.



V. Conclusiones:

Esta práctica para mi tuvo un grado de complejidad ya que jamás había trabajado con una tabla protoboard ni tampoco con los componentes que usamos, tuve que investigar mucho acerca de todos los componentes, en primer lugar, como funcionaba la tabla, una vez hecho eso buscar el data sheet del PIC y del cátodo para saber que, hacia cada una de las patitas de ambos, pero al final salió el resultado que esperábamos.