

Nombre de la práctica	Contador Hexadecimal en Cátodo Común de 7 Segmentos			No.	1
Asignatura:	Arquitectura de Computadoras	Carrera:	Ingeniería en Sistemas Computacionales	Duración de la práctica (Hrs)	12 horas

NOMBRE DEL ALUMNO: Tenyo Emanuel Petrearc Pineda
GRUPO: 3012

I. Competencia(s) específica(s):

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

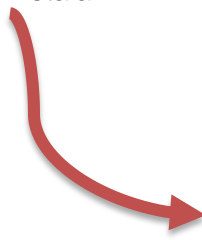
Aula de clases y laboratorio independiente

III. Material empleado:

Tabla Protoboard
PIC16F84A
Cátodo Común de 7 Segmentos
Cristal Oscilador de 4 MHZ
2 capacitores Cerámicos de 22 Picofaradios
Cable
Cargador con cable USB
Resistencias de 120 Ohms

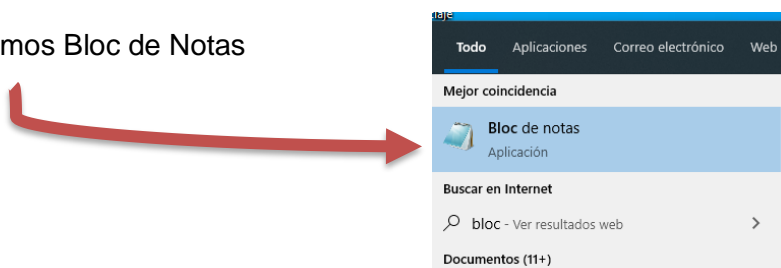
IV. Desarrollo de la práctica:

Descargar e Instalar MPLAB

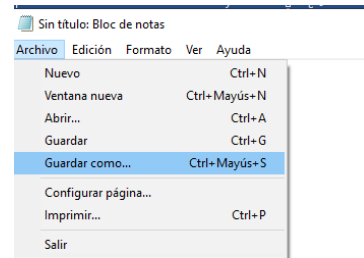


Nombre	Fecha de modificación	Tipo	Tamaño
Data1	13/08/2010 12:37 p. m.	Archivo WinRAR	88,889 KB
ISSetup.dll	13/08/2010 12:18 p. m.	Extensión de la ap...	2,055 KB
MPLAB Tools v8.56	13/08/2010 12:38 p. m.	Paquete de Windo...	9,313 KB
MPLAB_IDE_v8_56	04/12/2014 12:18 p. m.	Archivo WinRAR Z...	100,688 KB
mplabcert	17/07/2009 08:36 p. m.	Archivo BMP	193 KB
setup	13/08/2010 12:38 p. m.	Aplicación	3,783 KB

Abriremos Bloc de Notas



Guardaremos el archivo en la carpeta donde sea de fácil acceso

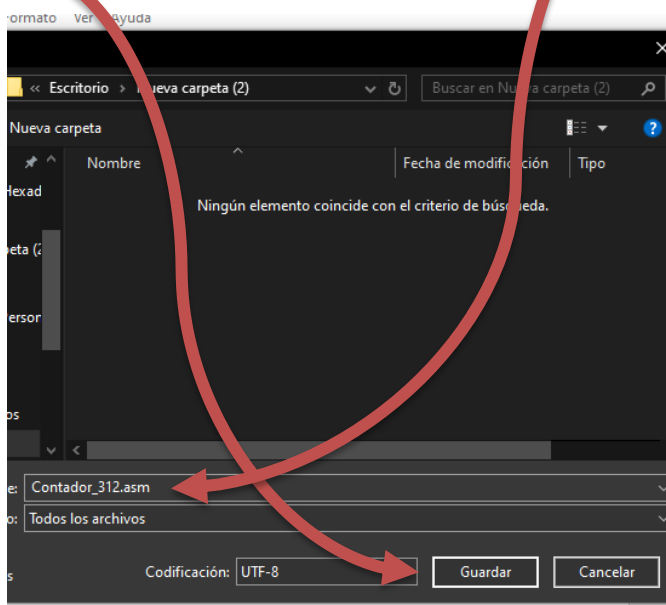


Lo guardaremos con el nombre que quiera, pero con extensión .asm

Ejemplo:

Contador_312.asm

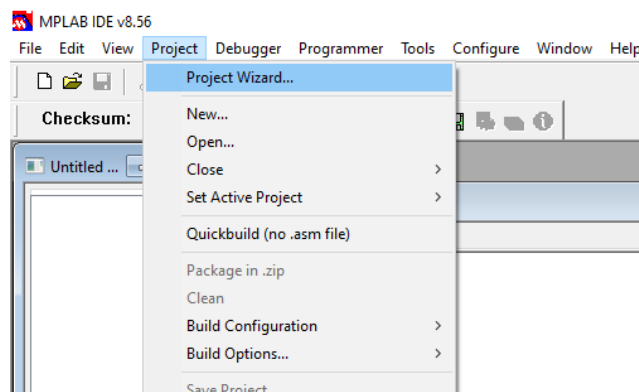
Y daremos Guardar



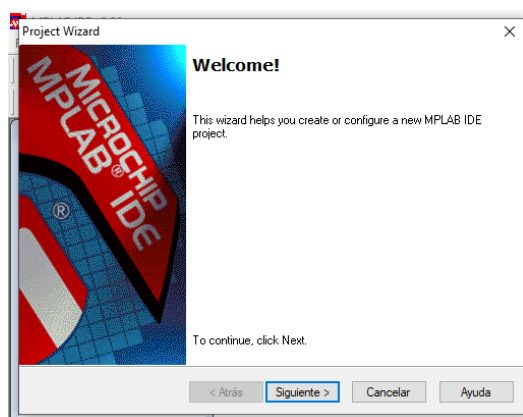
Abrir MPLAB



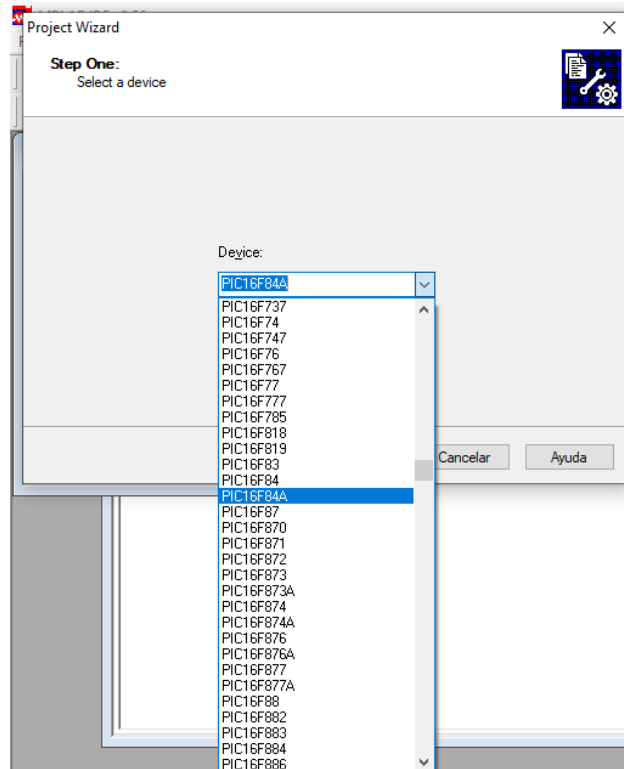
En la pestaña “Project” se deslizará una ventana y seleccionaremos “Project “Wizard”



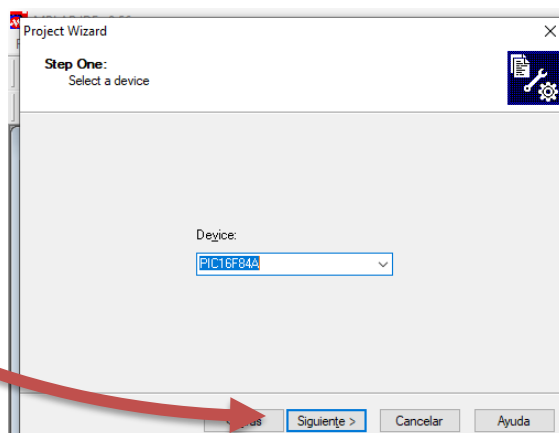
Aparecerá una ventana donde daremos siguiente



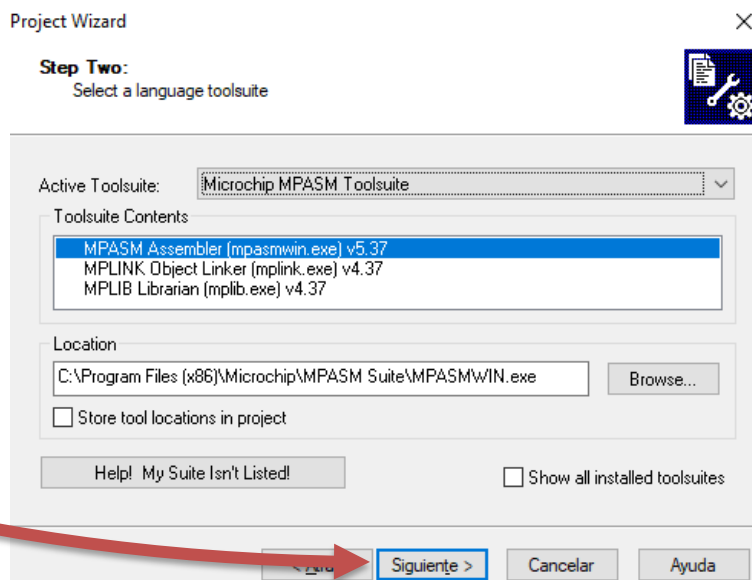
Seleccionaremos nuestro PIC correspondiente, en este caso ocupamos el PIC16F84A



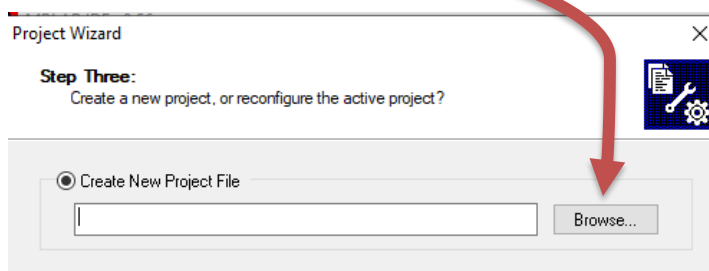
Pulsamos siguiente



En esta ventana daremos siguiente sin cambiar nada



Seleccionaremos donde guardar nuestro proyecto

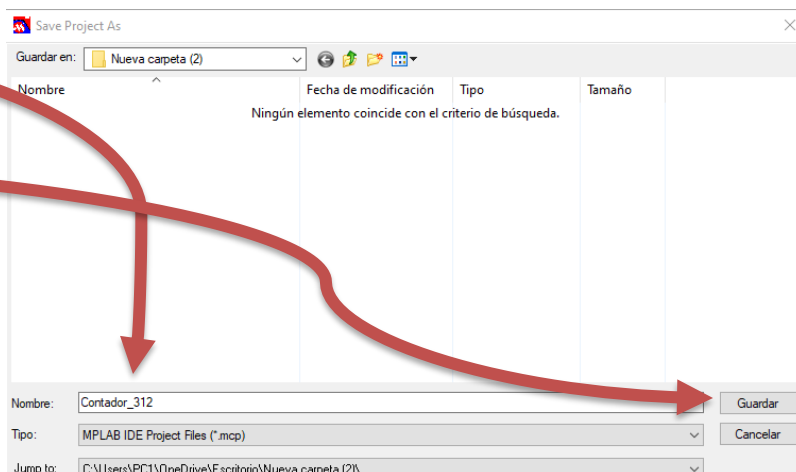


Escribimos el nombre de nuestro proyecto

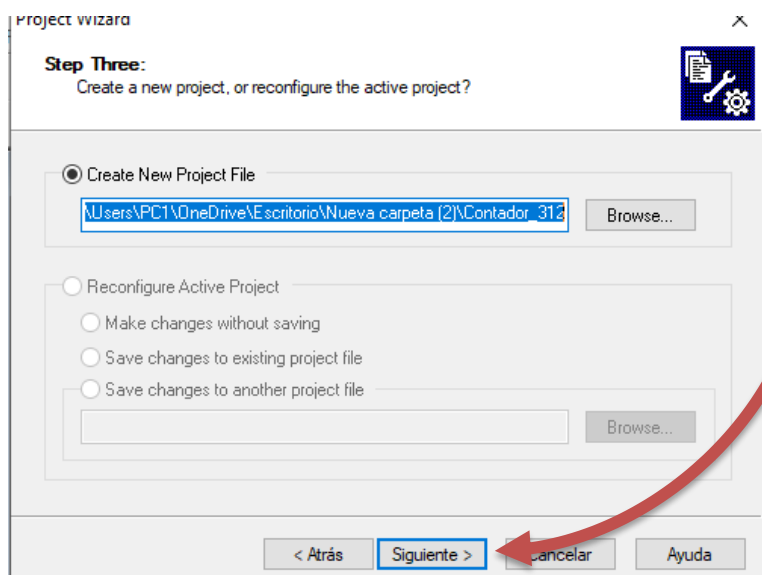
Ejemplo:

Contador_312

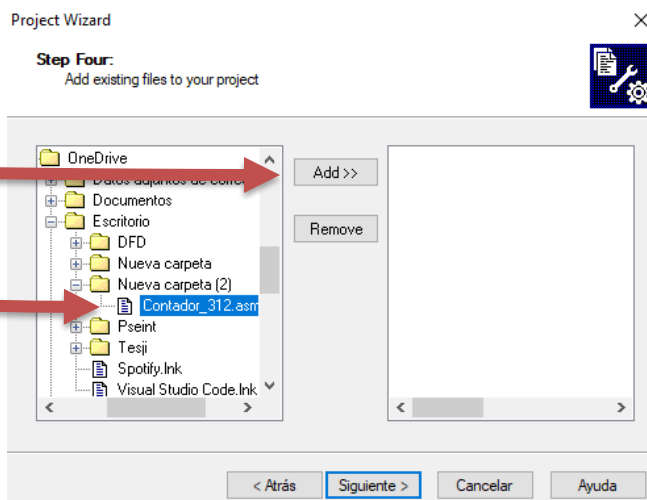
Dar click en guardar



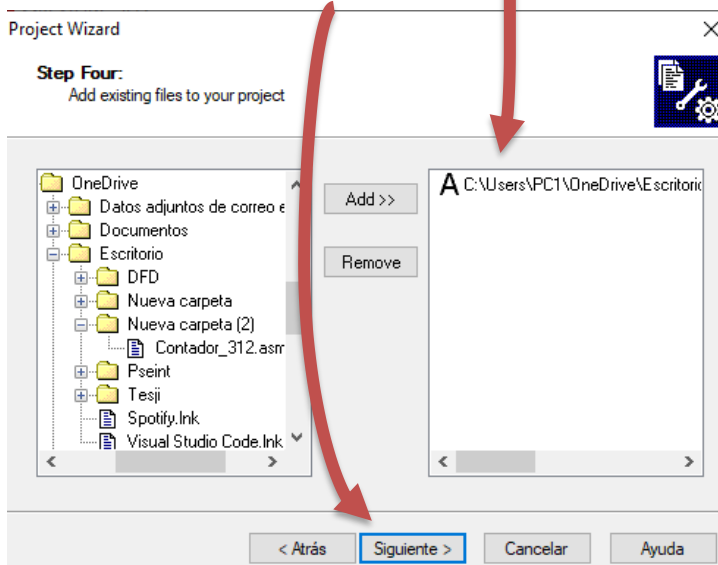
Se guardara la dirección que especificamos y daremos en siguiente



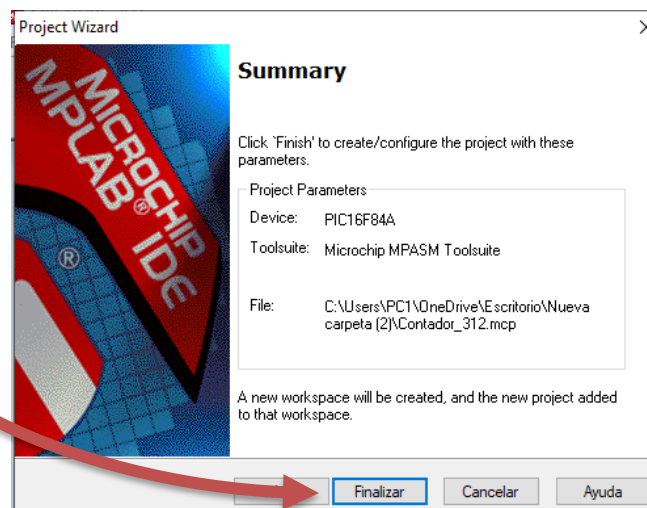
En la siguiente ventana seleccionaremos nuestro archivo con extension .asm y daremos click en "Add"



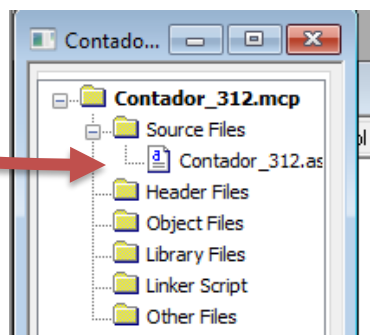
Nos arrastrara el archivo a la derecha
Despues daremos lick en siguiente



Daremos en Finalizar

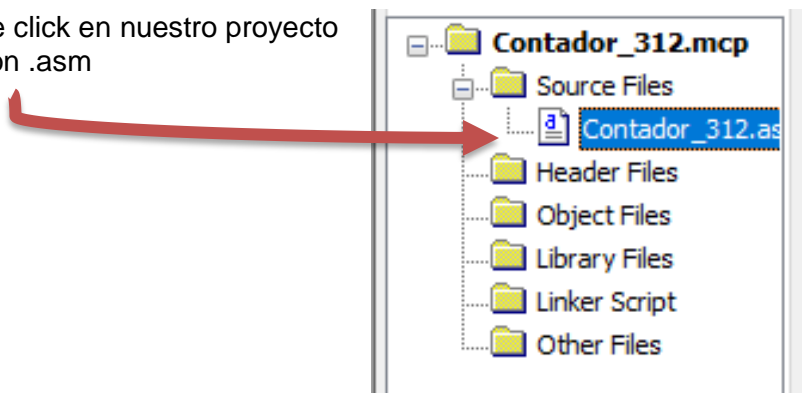


Nuestro proyecto estara creado
Y el archivo estara agregado

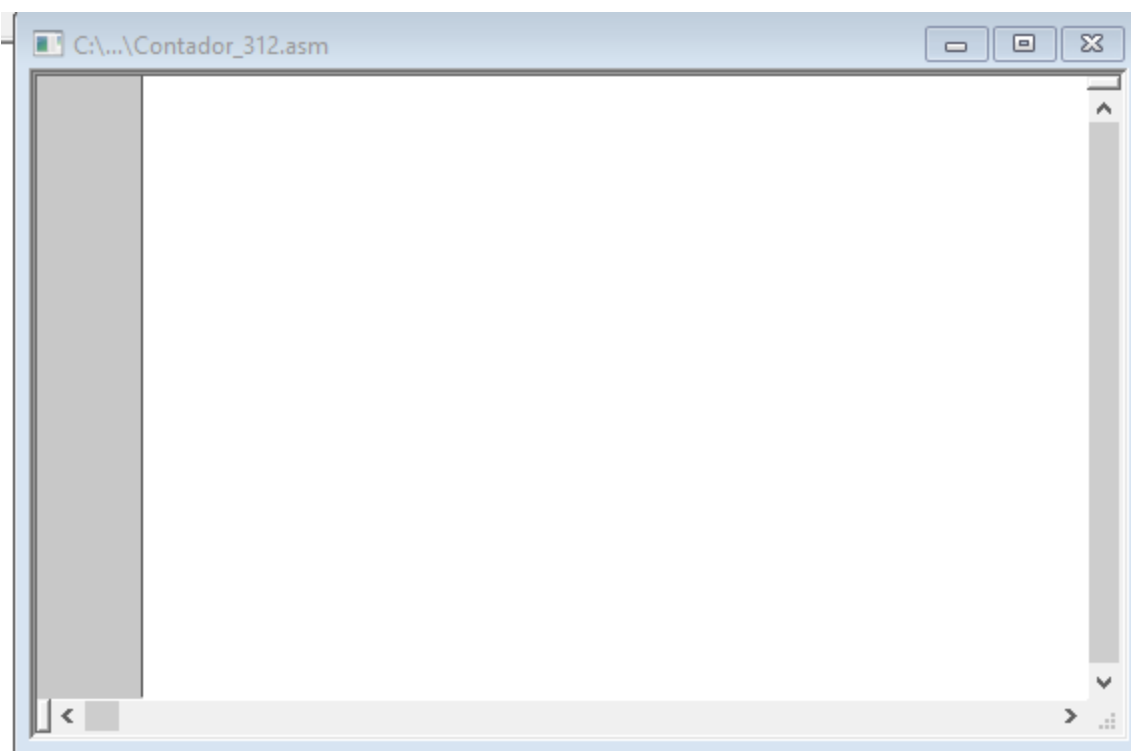




Daremos doble click en nuestro proyecto
Con terminacion .asm



Se nos abra una ventana donde podemos escribir nuestro codigo
En est caso "Ensamblador"



Primero realizamos nuestra zona de datos

Configurar nuestro PIC

```
; ZONA DE DATOS *****  
__CONFIG __CP_OFF & __WDT_OFF & __PWRTE_ON & __XT_OSC
```

- Contador de un Programa. Deshabilitado dentro del PIC
- Poder reprogramar el PIC. Desabilitar la opcion de sellado
- Activar la escritura
- Activar el oscilador de cristal el propio PIC

```
LIST P=PIC16F84A  
INCLUDE <P16F84A.INC>
```

- Listar los PICS que se Programaran
- Incluir las bibliotecas para que pueda operar el PIC dentro del propio PIC

Continuamos con nuestra Zona de Variables

```
;ZONA DE VARIABLES*****  
CBLOCK 0X0C ; SE INICIALIZA LA MEMORIA  
    NUMERO ; VARIABLE QUE LLEVARA EL CONTADOR DE 0-9 Y DE A-F  
    CONTADOR ; LLEVA EL TIEMPO EN CICLOS DE RELOJ  
ENDC  
ORG 0  
COTO START ; CICLO  
ORG 5
```

Las oraciones que siguen despues del punto y coma (;) son mensajes que indican para que sirve cada función



Después continuamos con la Configuración del PIC

```
; CONFIGURACION*****
START      BSF      STATUS,5      ; BANCO 1 ACTIVA EL BIT B DE F
           CLRF     TRISB          ; PORTB ES LA SALIDA
           MOVLW    0X1F          ; MOVER A LA PARTE BAJA DEL REGISTRO DE MEMORIA RA0 RA4 SON LAS ENTRADAS
           MOVWF    TRISA         ; MOVER A F A TRISA
           MOVLW    B'11000111'   ; ASIGNA 256 AL TIMER
           MOVWF    OPTION_REG    ; MOVER F AL REGISTRO OPTION_REG
           BCF      STATUS,5      ; CARGAR EL CONTENIDO DE LA POSICION 5 AL BANCO 0
           CLRW     W             ; DEJA A W=0
           CLRF     NUMERO        ; LIMPIA LA VARIABLE NUMERO
```

Continuamos con el Inicio que realizara el PIC

```
; INICIO*****
MAIN       MOVF     NUMERO,W      ; TOMA EL CONTENIDO DE W Y LO CARGA EN F
           CALL     TABLA         ; LLAMA A LA FUNCION TABLA
           MOVWF    PORTB        ; MUESTRA EL VALOR QUE TOMO LA TABLA
           CALL     PAUSE_1000    ; LLAMA A LA FUNCION PAUSE
           INCF     NUMERO,F      ; REALIZA UN INCREMENTO DE LA VARIABLE 1
           MOVF     NUMERO,W      ; SE CARGA LA VARIABLE EN W
           XORLW    0X10          ; SE COMPARA SI ES QUE LLEGA AL REGISTRO 10
           BTFS    STATUS,Z      ; VERIFICA SI HA LLEGADO
           GOTO     MAIN         ; SE REGRESA A MAIN
           CLRW     W            ; AL LLEGAR A 10 SE RESETEA A F
           CLRF     NUMERO       ; SE LIMPIA A NUMERO
           GOTO     MAIN         ; SE REGRESA A MAIN
; FIN DE INICIO*****
```

La pausa o retardo que tendra entre un digito y el otro

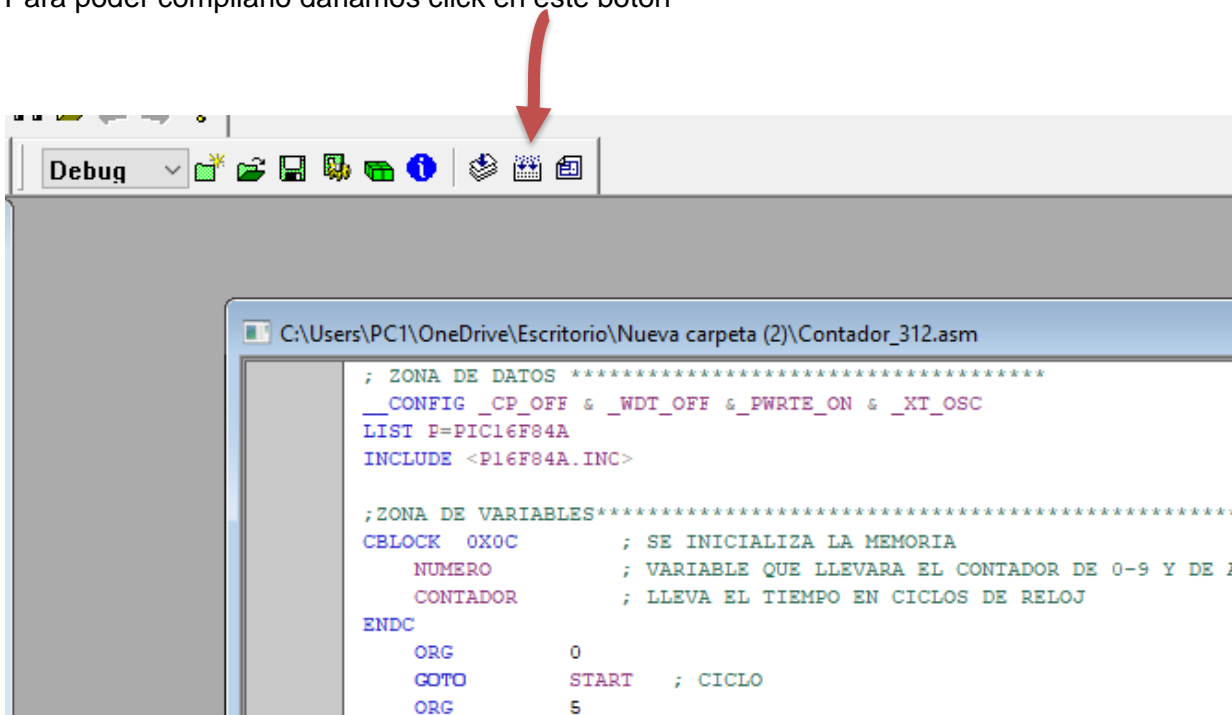
```
; PAUSE_1000 SE REALIZA UN RETARDO DE UN SEGUNDO*****
PAUSE_1000 MOVLW    0X02          ; SE LE ASIGNA 1000 AL CONTADOR
           MOVWF    CONTADOR      ; SE MUEVE EL VALOR DEL CONTADOR DE A a F
DELAY      BCF      INTCON,TOIF   ; SE LIBERA EL BIT DE DESBORDAMIENTO TMR0
           MOVLW    09            ; SE LE CARGA EL 217 A W
           MOVWF    TMR0         ; A TMR0
DELAY2     BTFS    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         ; BUEN EN DELAY
           RETURN                ; REGRESA
```

El proceso que hara en el Catodo Comun de 7 Segmentos

```
; TABLA*****
TABLA      ADDWF    PCL,F          ; SE INICIALIZA LA FUNCION TABLA CON EL CONTENIDO DE F
          RETLW    B'00111111'    ; 0 EN CATADO 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
```

El codigo estaria acabado

Para poder compilarlo dariamos click en este boton





Realizara un proceso donde debe de revisar el codigo, linea por linea, y si es correcto lo compilara sin problemas.

```
Build Version Control Find in Files

Debug build of project 'C:\Users\PC1\OneDrive\Escritorio\Nueva carpeta (2)\Contador_312.mcp' started.
Language tool versions: MPASMWIN.exe v5.37, mplink.exe v4.37, mplib.exe v4.37
Preprocessor symbol '__DEBUG' is defined.
Sat Oct 19 17:17:52 2019

Clean: Deleting intermediary and output files.
Clean: Deleted file "C:\Users\PC1\OneDrive\Escritorio\Nueva carpeta (2)\Contador_312.o".
Clean: Deleted file "C:\Users\PC1\OneDrive\Escritorio\Nueva carpeta (2)\Contador_312.err".
Clean: Deleted file "C:\Users\PC1\OneDrive\Escritorio\Nueva carpeta (2)\Contador_312.lst".
Clean: Deleted file "C:\Users\PC1\OneDrive\Escritorio\Nueva carpeta (2)\Contador_312.cof".
Clean: Deleted file "C:\Users\PC1\OneDrive\Escritorio\Nueva carpeta (2)\Contador_312.hex".
Clean: Done.
Executing: "C:\Program Files (x86)\Microchip\MPASM Suite\MPASMWIN.exe" /q /p16F84A "Contador_312.asm" /l"Contador_312.lst" /e"Contador_312.err" /o"Contador_312.o" /d__DEBUG=1

Message[302] C:\USERS\PC1\ONEDRIVE\ESCRITORIO\NUEVA CARPETA (2)\CONTADOR_312.ASM 20 : Register in operand not in bank 0. Ensure that bank bits are correct.
Warning[205] C:\USERS\PC1\ONEDRIVE\ESCRITORIO\NUEVA CARPETA (2)\CONTADOR_312.ASM 68 : Found directive in column 1. (END)
Executing: "C:\Program Files (x86)\Microchip\MPASM Suite\mplink.exe" /p16F84A "Contador_312.o" /u_DEBUG /z__MPLAB_BUILD=1 /o"Contador_312.cof" /M"Contador_312.map" /A
MPLINK 4.37. Linker
Copyright (c) 1998-2010 Microchip Technology Inc.
Errors : 0


MP2HEX 4.37. COFF to HEX File Converter
Copyright (c) 1998-2010 Microchip Technology Inc.
Errors : 0




Loaded C:\Users\PC1\OneDrive\Escritorio\Nueva carpeta (2)\Contador_312.cof.

Debug build of project 'C:\Users\PC1\OneDrive\Escritorio\Nueva carpeta (2)\Contador_312.mcp' succeeded.
Language tool versions: MPASMWIN.exe v5.37, mplink.exe v4.37, mplib.exe v4.37
Preprocessor symbol '__DEBUG' is defined.
Sat Oct 19 17:17:55 2019

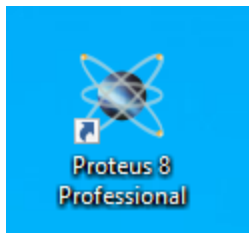
BUILD SUCCEEDED
```

El siguiente paso es instalar y crackear proteus

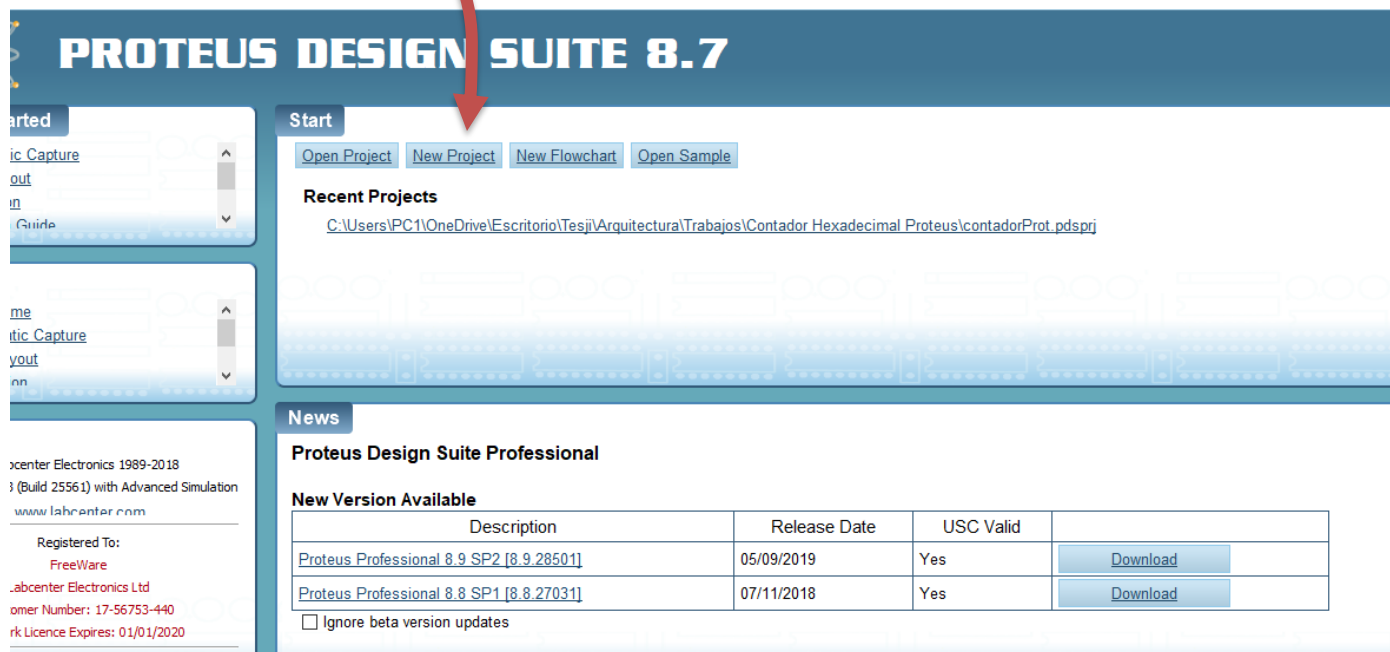


Nombre	Fecha de modificación	Tipo	Tamaño
 Extra proteus 8.7	29/04/2018 01:57 p. m.	Carpeta de archivos	
 Proteus 8.7 Full By B.C	30/09/2019 10:25 p. m.	Archivo WinRAR	375,880 KB
 Proteus 8.7 SP3 PRO	11/04/2018 03:08 a. m.	Aplicación	387,622 KB

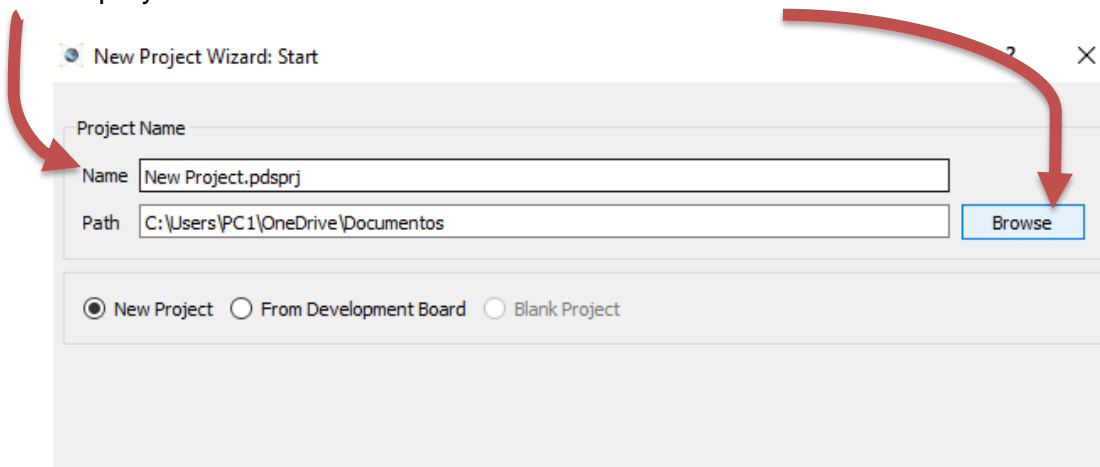
Una vez instalado y crackeado correctamente lo abrimos



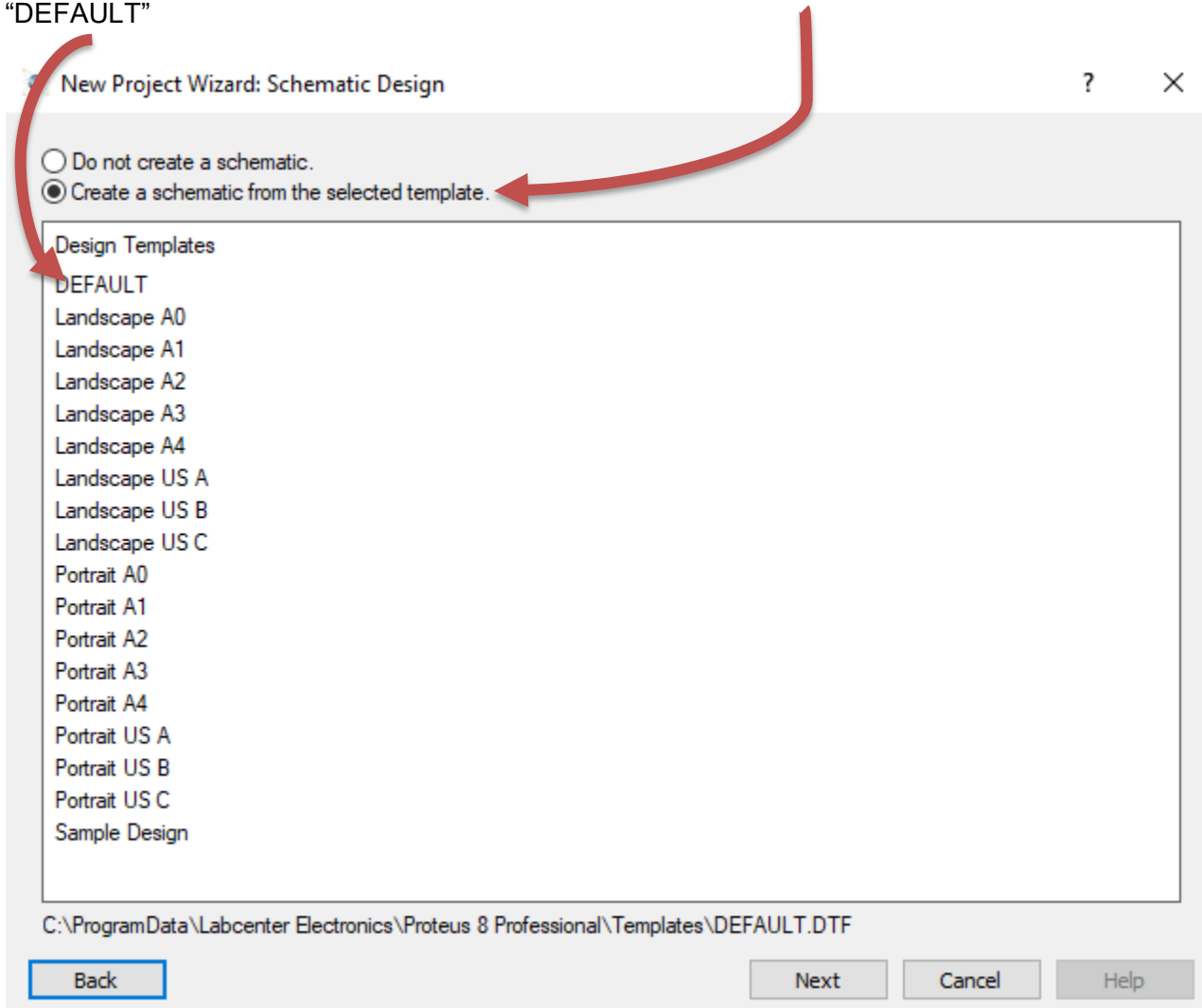
Daremos click en “New Project”



Abrir una ventana donde seleccionaremos una dirección para guardar nuestro proyecto y cambiarle el nombre al proyecto

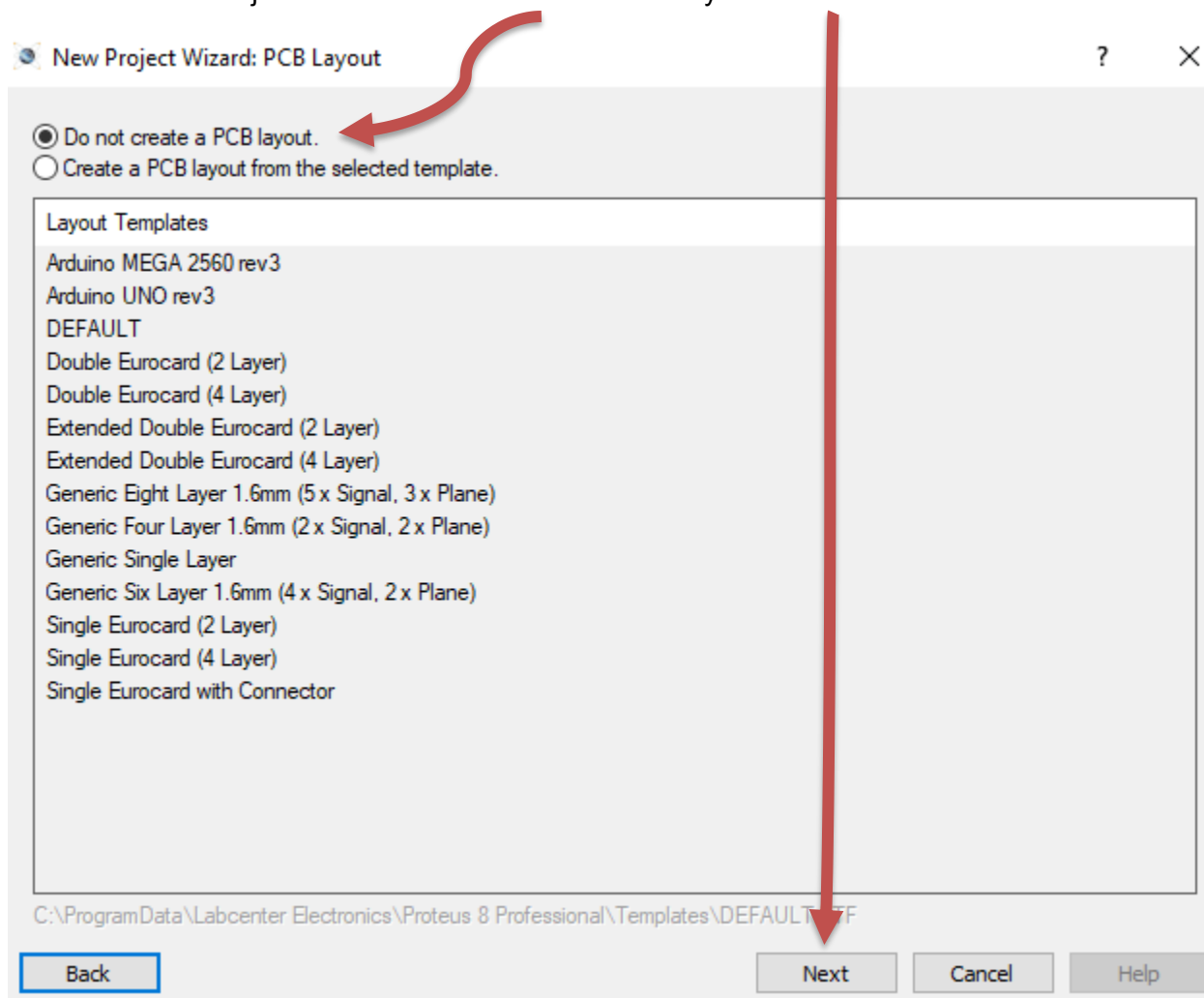


Una vez ubicada, daremos “Next”, marcamos la opción de “Create a schematic...” Y lo dejamos en “DEFAULT”



Daremos "Next"

En esta ventana dejaremos marcada "Do not create..." y damos "Next"



Seleccionaremos "Create Firmware Project"

Seleccionaremos la Familia del PIC que ocuparemos, en este caso fue PIC16

Seleccionaremos el PIC en específico que ocuparemos

New Project Wizard: Firmware

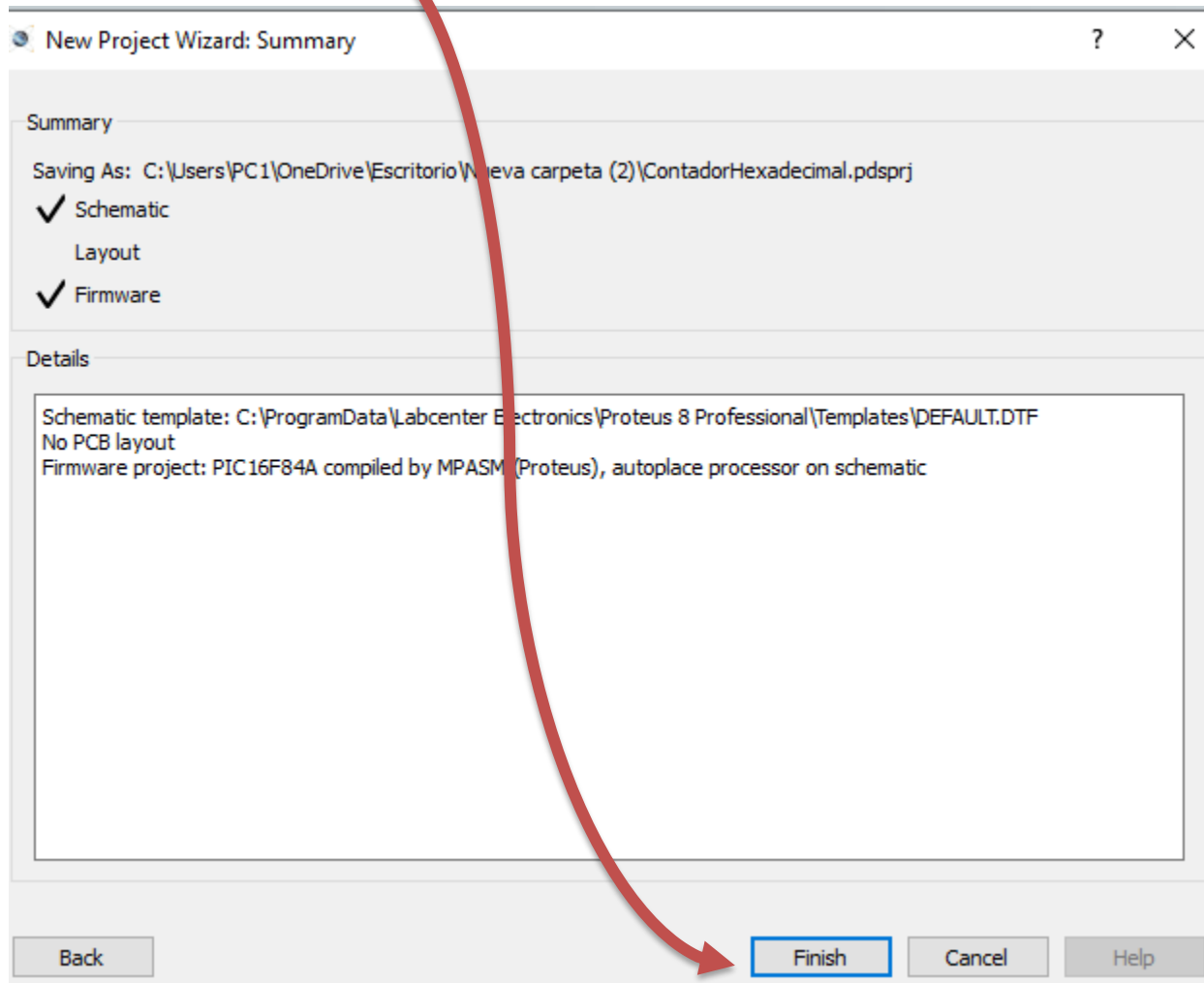
☐ No Firmware Project
☒ Create Firmware Project
☐ Create Flowchart Project

Family: PIC16
Controller: PIC16F84A
Compiler: MPASM (Proteus) [Compilers...]
Create Quick Start Files: ☒
Create Peripherals: ☐

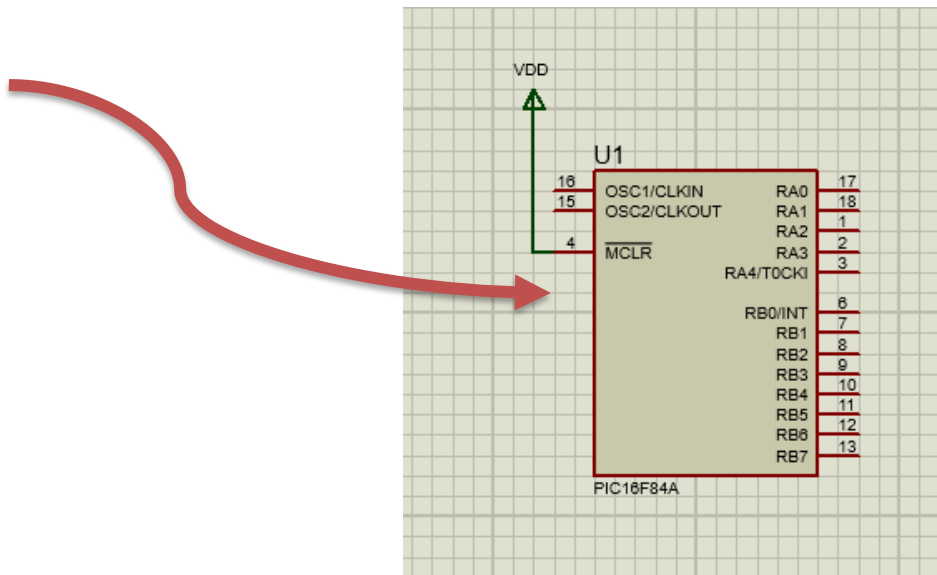
Back Next Cancel Help

Daremos "Next"

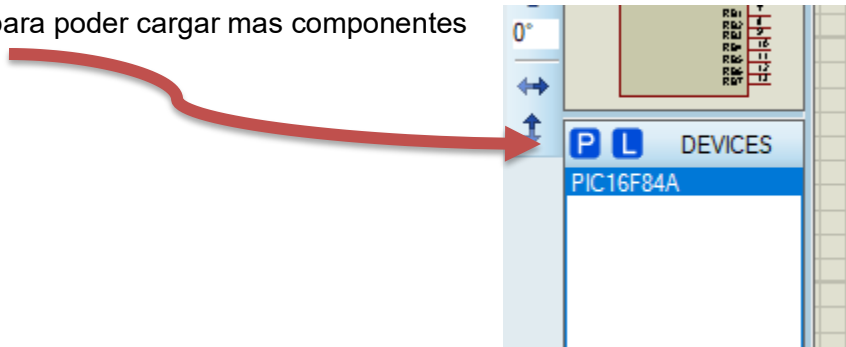
Y por ultimo daremos en "Finish"



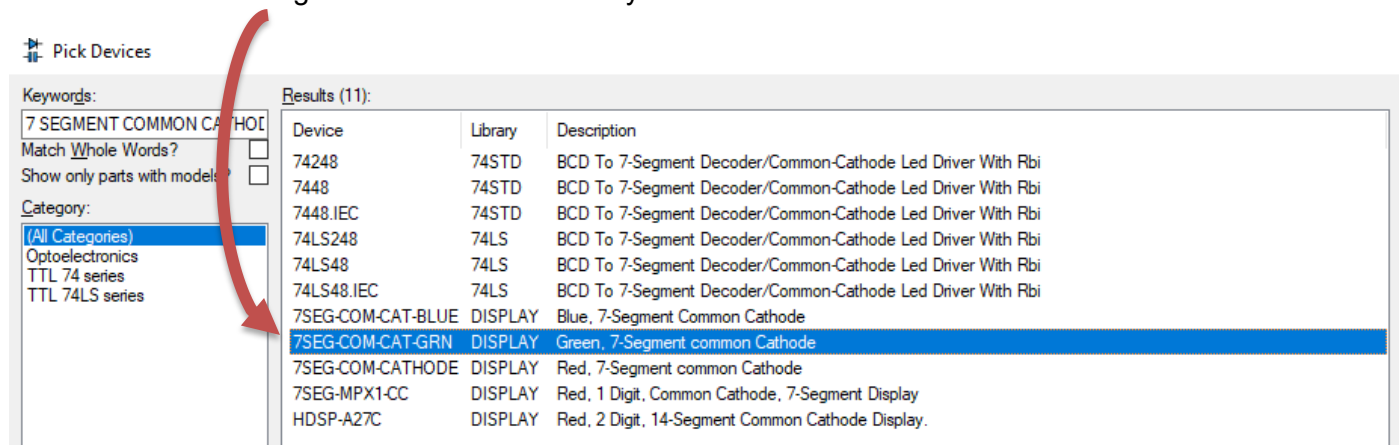
Cargara nuestro PIC



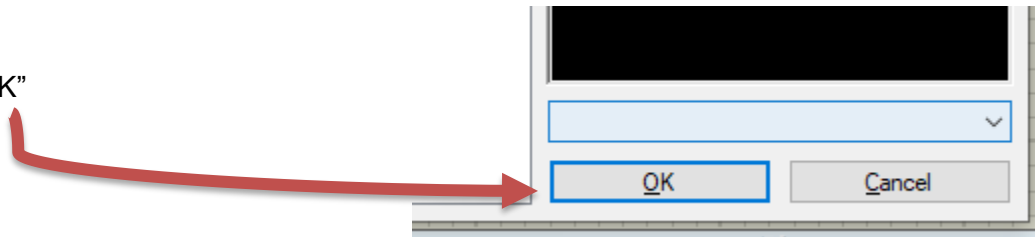
Daremos click en "P" para poder cargar mas componentes



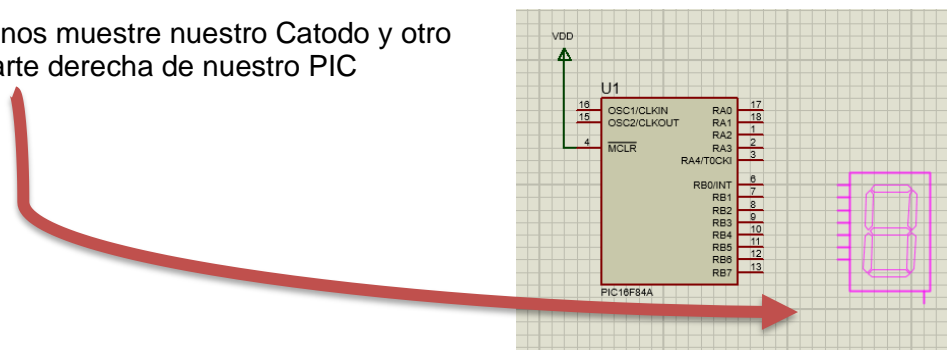
Buscaremos nuestro catodo correspondiente.
En este caso fue "7segment common cathode" y lo seleccionamos



Daremos click en "OK"

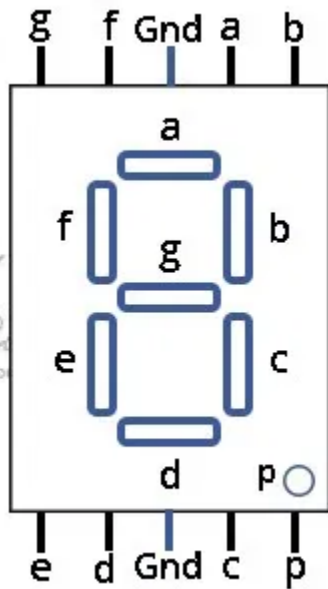


Daremos un click para que nos muestre nuestro Catodo y otro click para colocarlo en la parte derecha de nuestro PIC

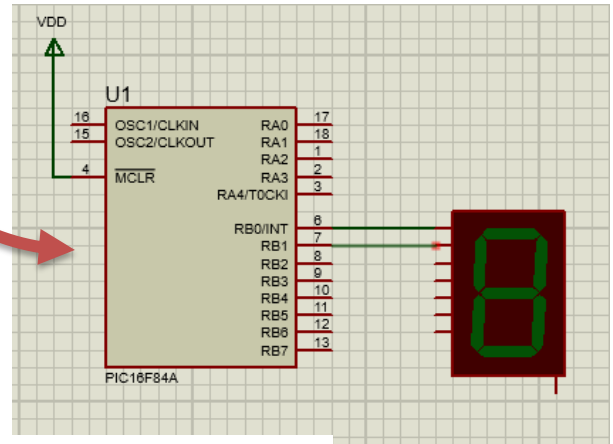
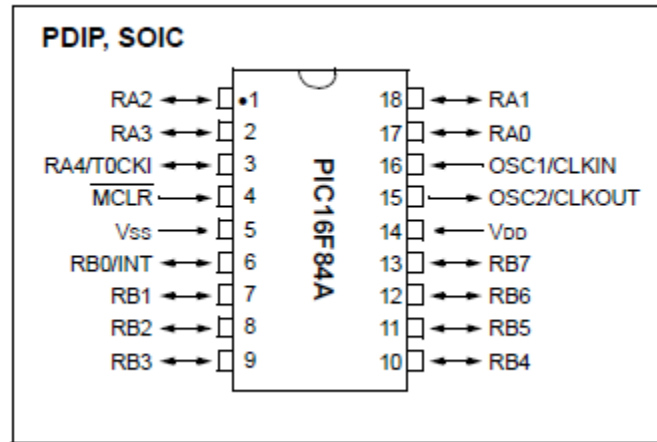




Ahora uniremos las patitas del PIC con las de nuestro Catodo, y esto es de acuerdo al DATASHEET del PIC16F84A y el Display del Catodo Comun de 7 Segmentos



Pin Diagrams



Buscaremos y agregaremos dos Capacitores Ceramicos de 22 picofaradios

Pick Devices

Keywords: ceramic22

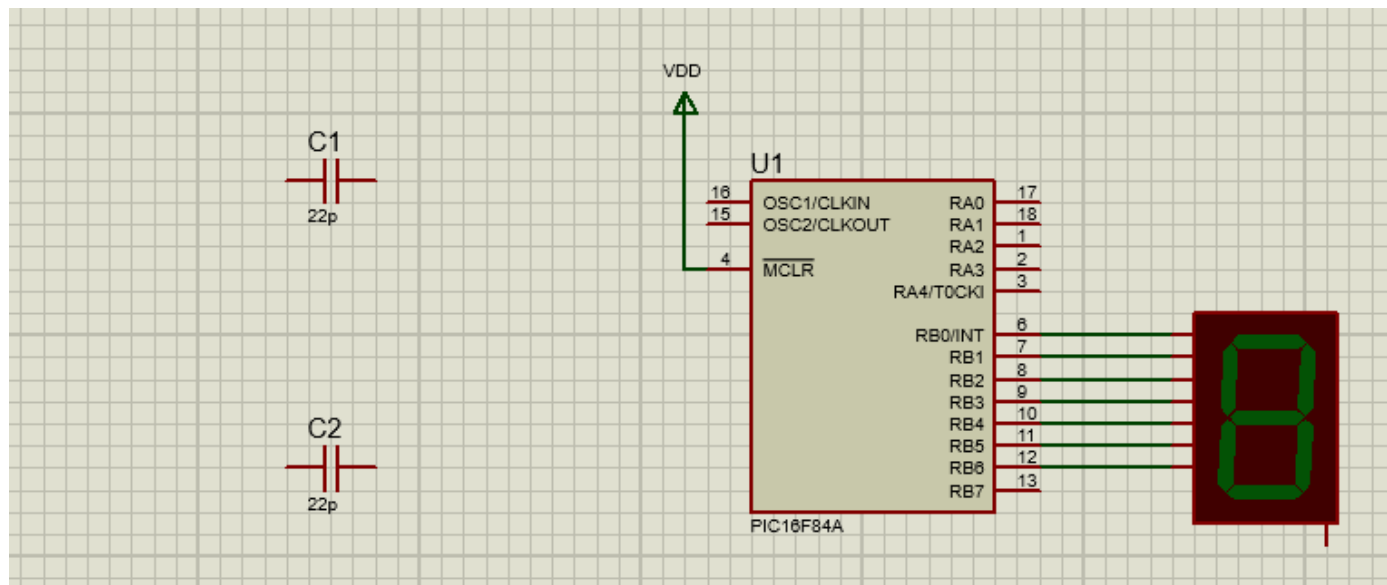
Match Whole Words? ☐

Show only parts with models? ☐

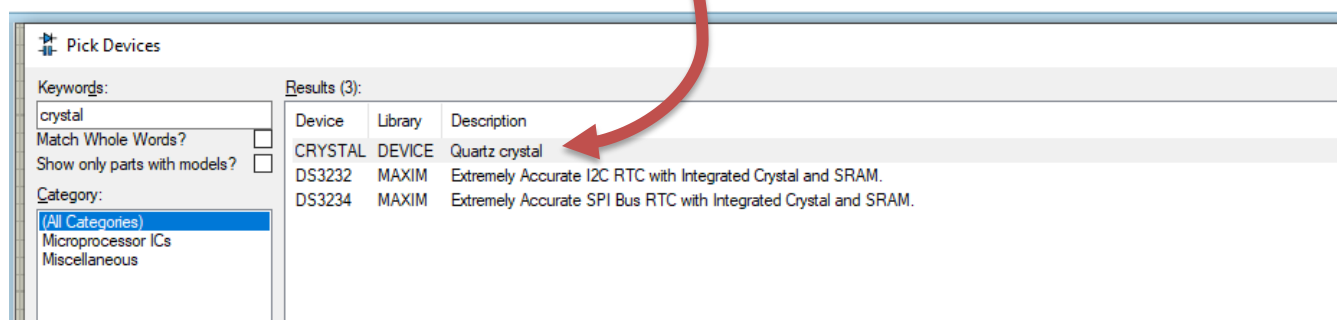
Category: (All Categories) Capacitors

Results (3):

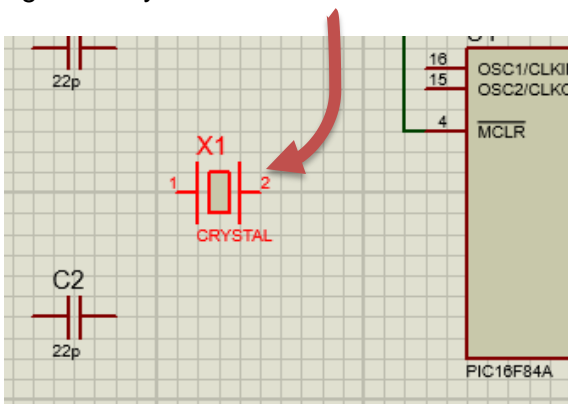
Device	Library	Stock Code	Description
CERAMIC220P	CAPACITORS	Maplin WX60Q	220p Ceramic Capacitor
CERAMIC22N	CAPACITORS	Maplin WX78K	22n Ceramic Capacitor
CERAMIC22P	CAPACITORS	Maplin WX48C	22p Ceramic Capacitor



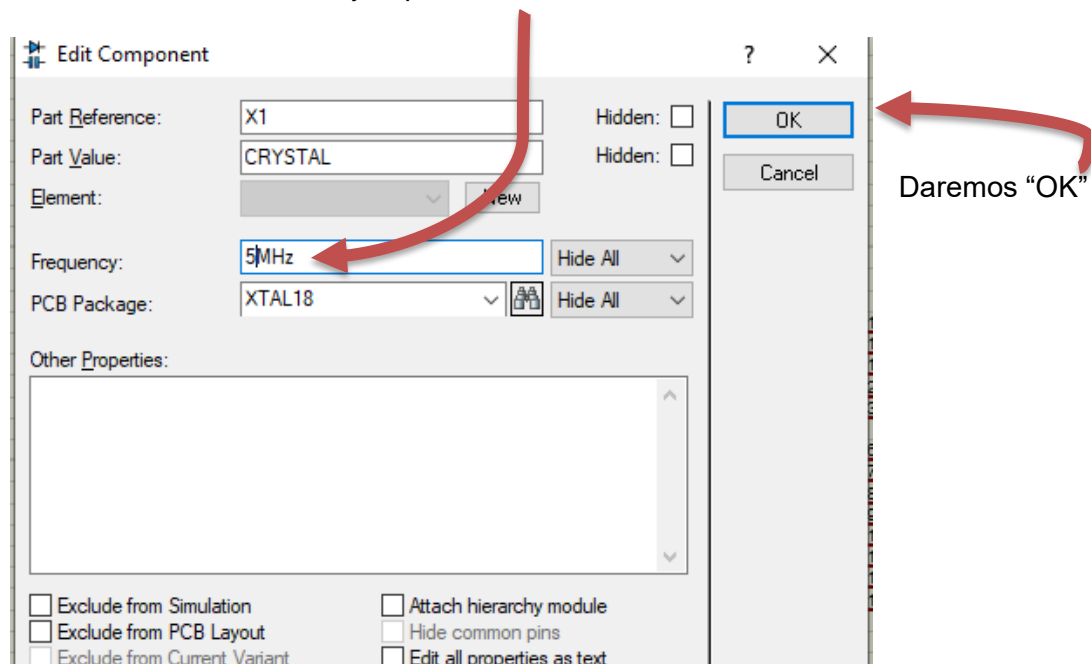
Buscaremos y agregaremos un “Cristal Oscilador”



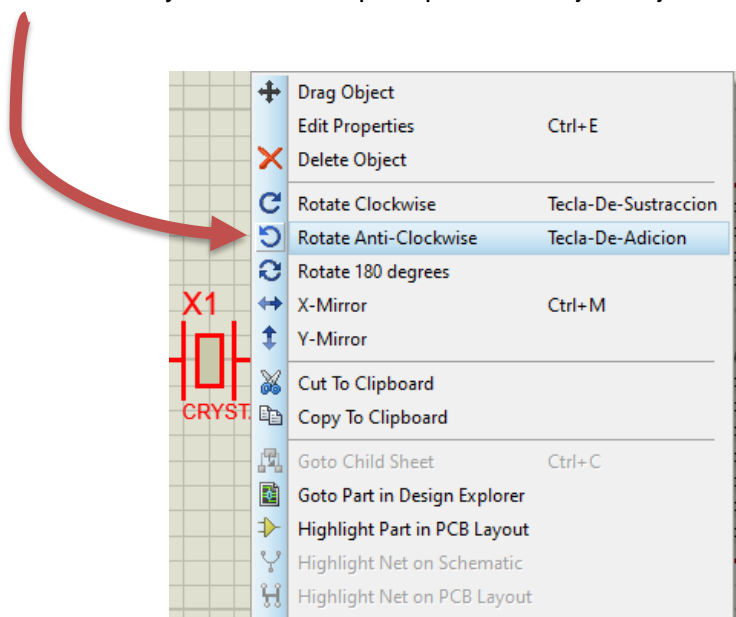
Lo agregaremos y daremos doble click en el



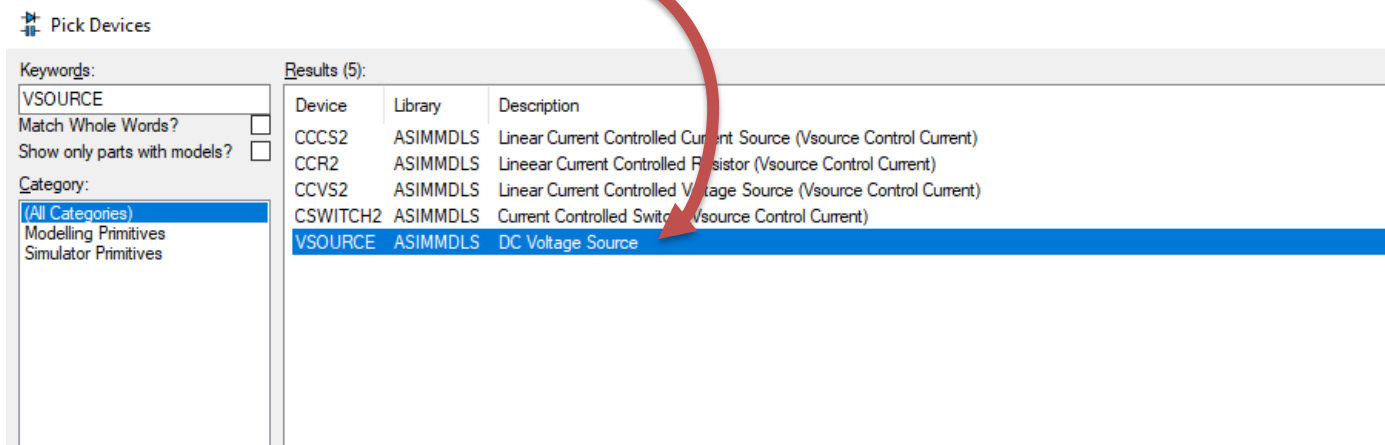
Editaremos la Frecuencia y la pondremos a 5



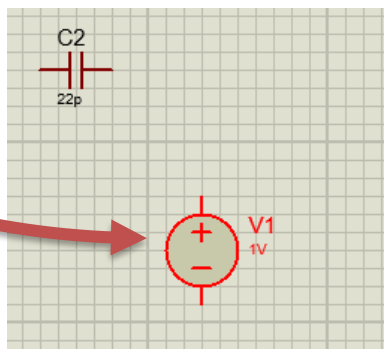
Daremos click derecho y lo rotaremos para poder trabajar mejor



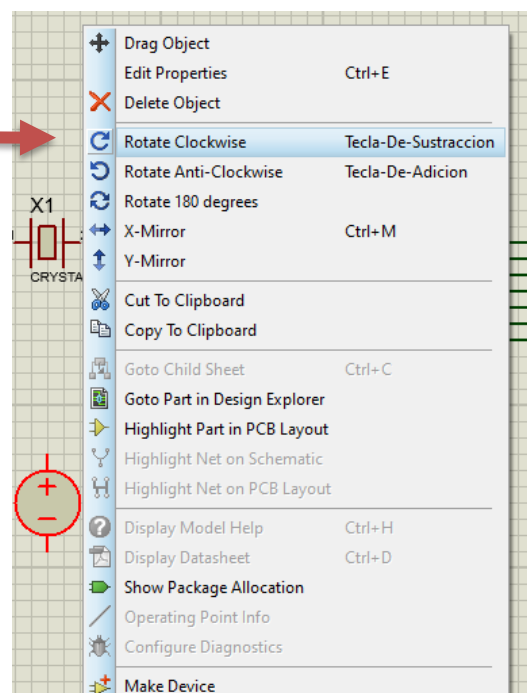
Por ultimo agregaremos una fuente de poder



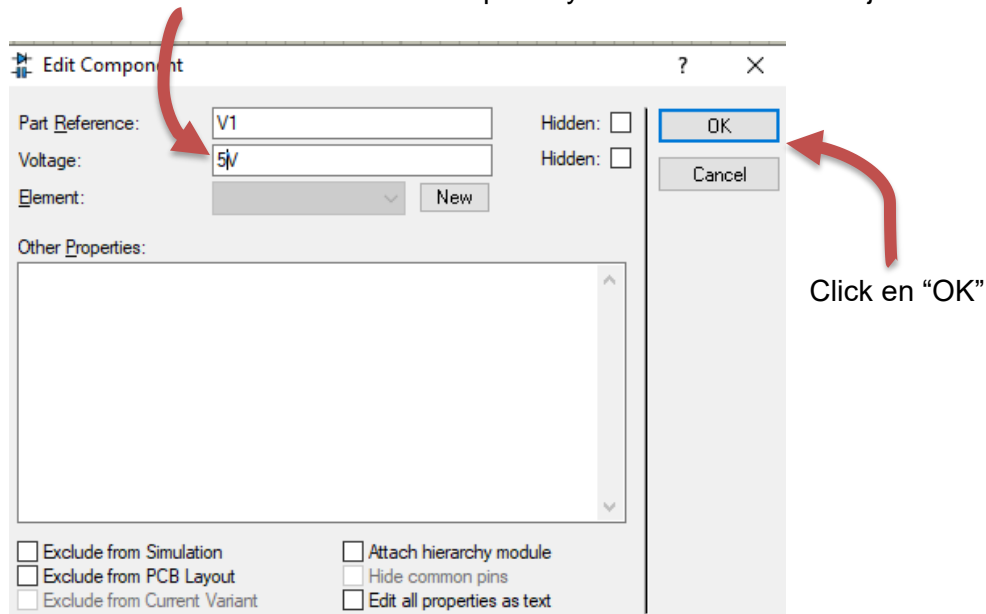
La colocaremos



Daremos click derecho y la rotaremos para poder trabajar mejor

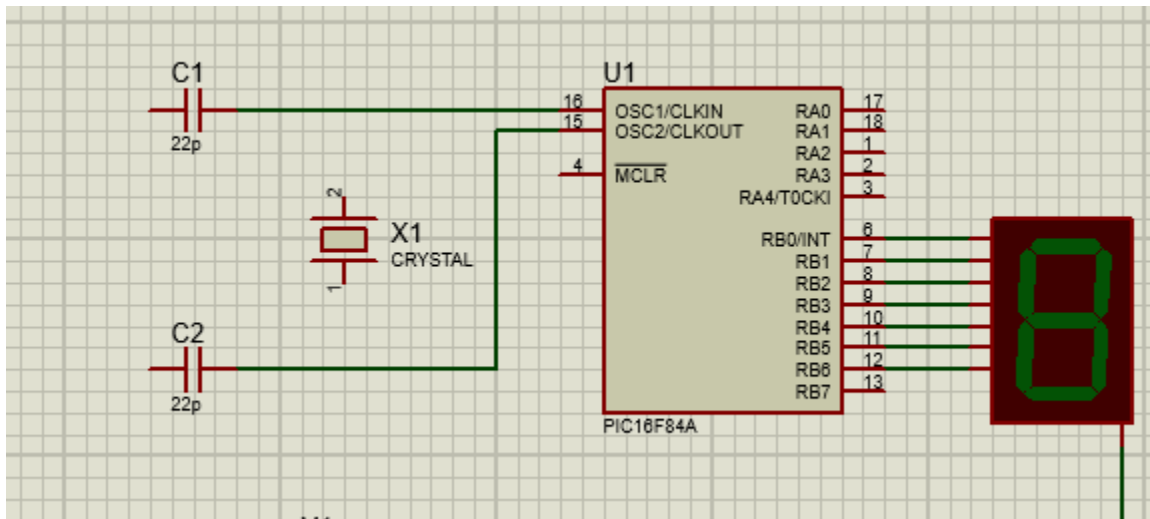


Daremos doble click en la fuente de poder y cambiaremos el voltaje a 5V

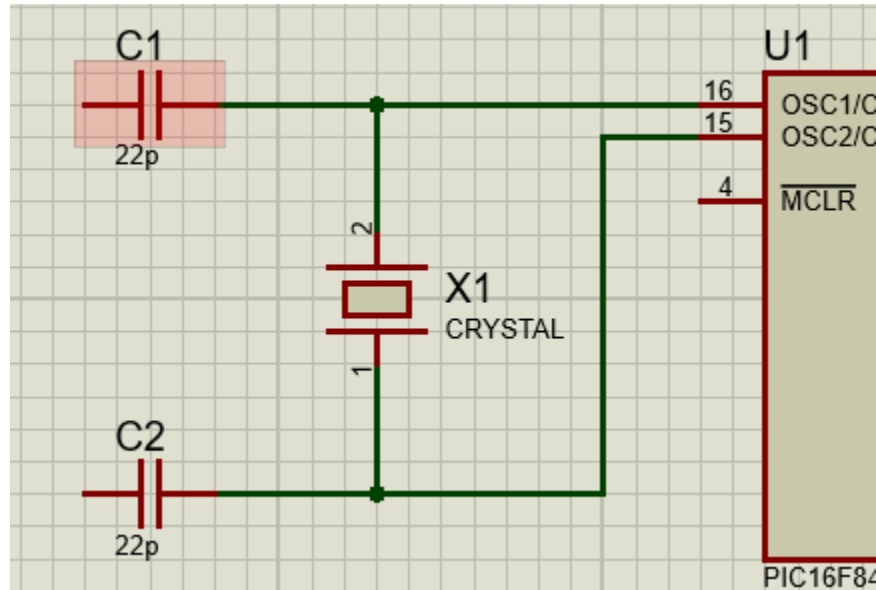


Haremos las siguientes uniones

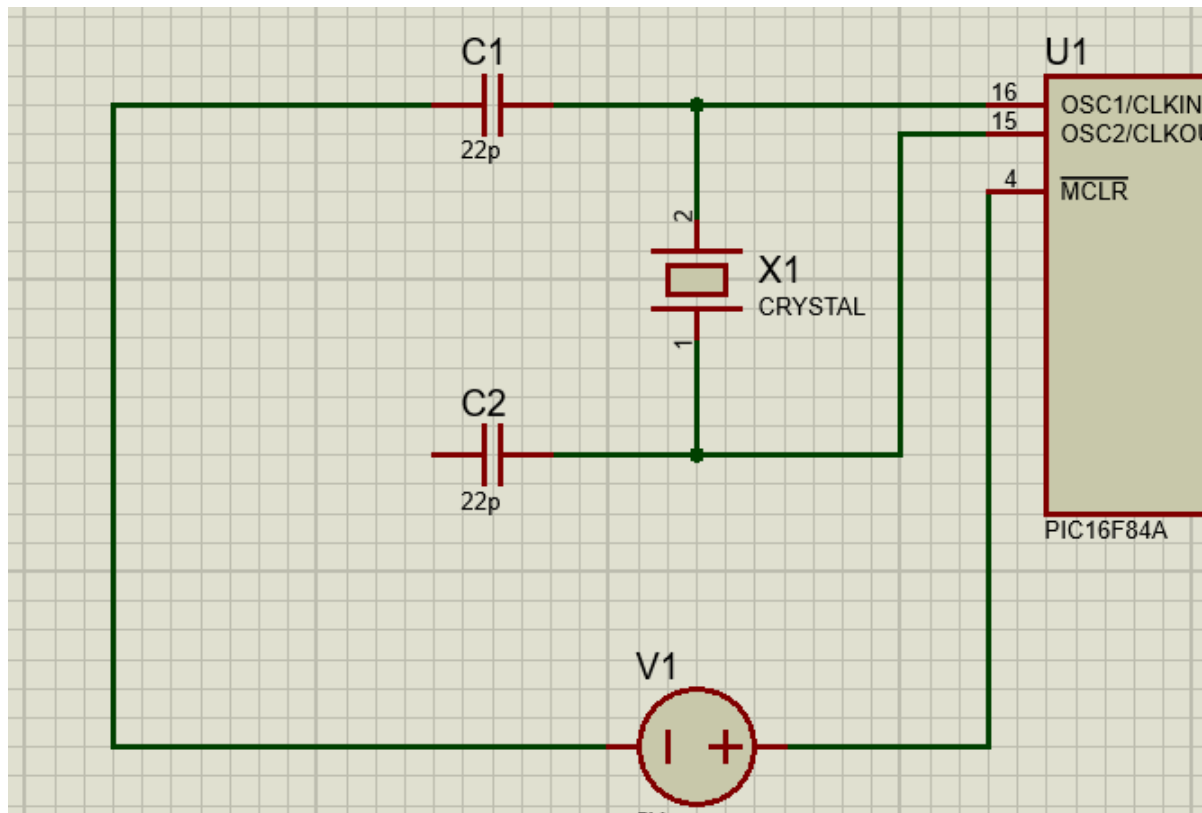
- Capacitor 1 con la patita 16 del PIC
- Capacitor 2 con la patita 15 del PIC



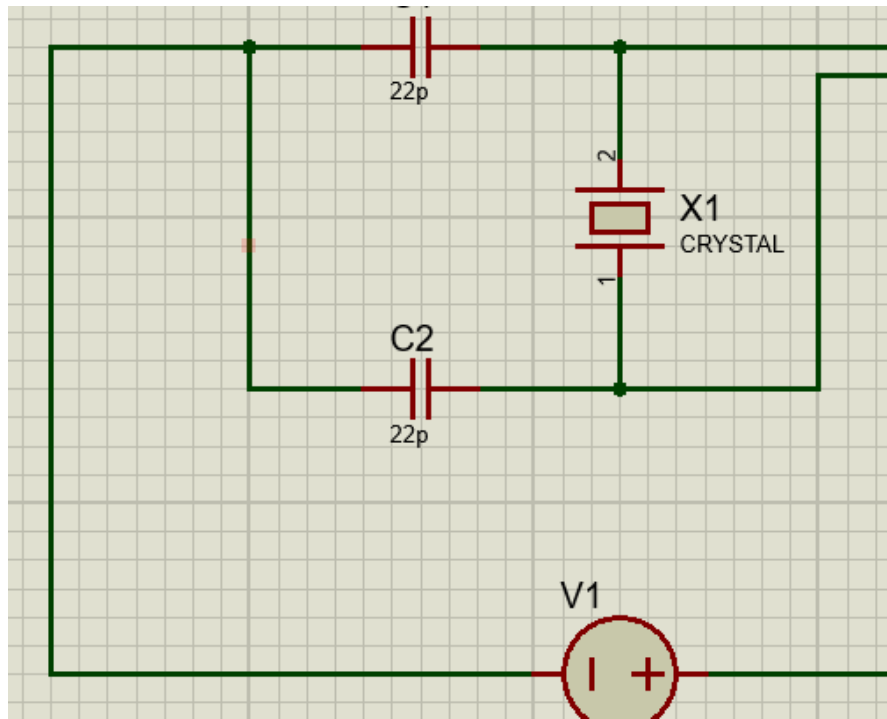
- El Cristal Oscilador lo unimos con cada línea que une entre el capacitor y el PIC



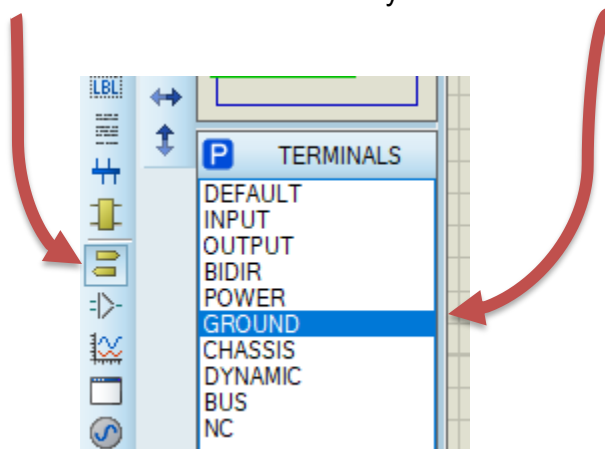
- La fuente de poder del lado positivo con la pata 4 (MCLR), y del lado negativo con el capacitor 1



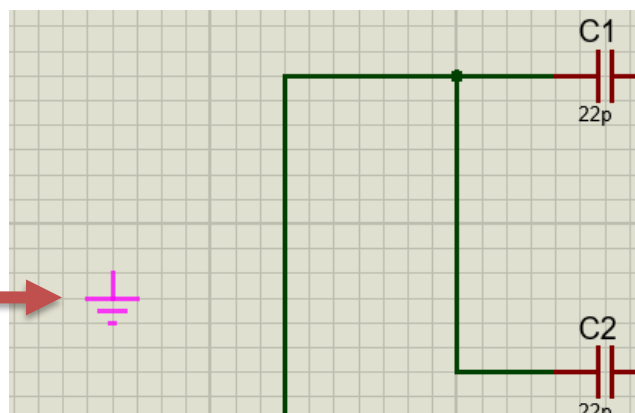
- El capacitor 2 intercepta en la línea que une al capacitor 1 y a la fuente de poder



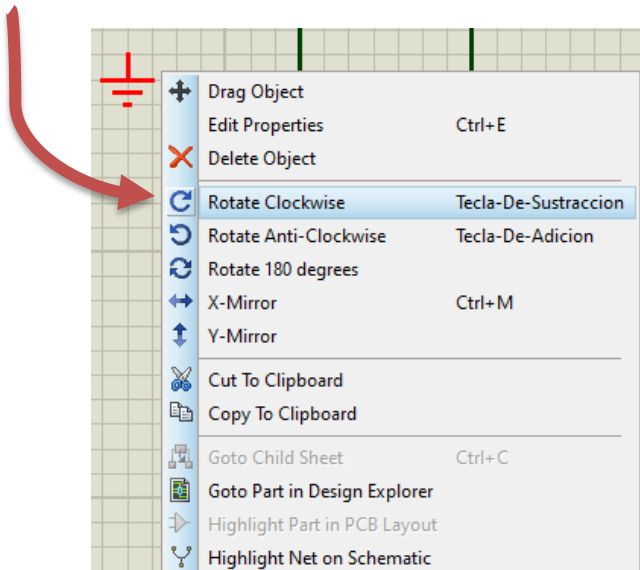
Damos click en las terminales y seleccionaremos "Ground"



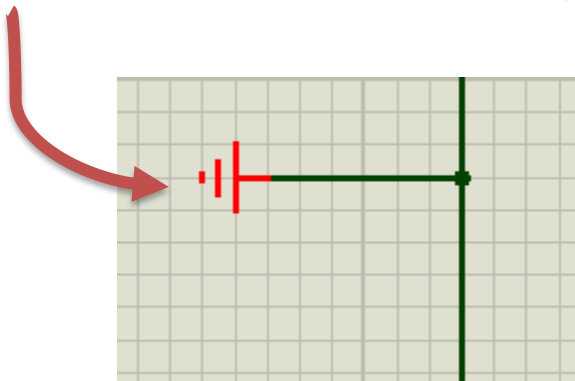
Colcaremos la terminal



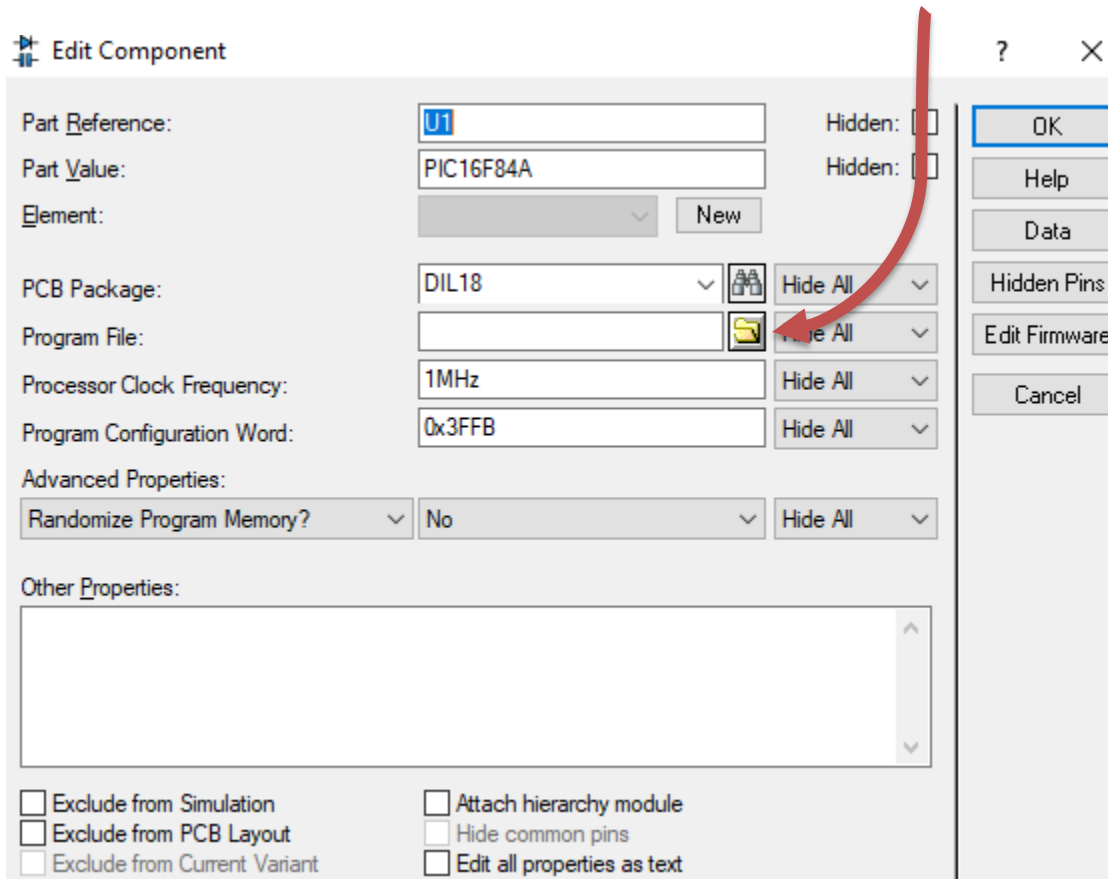
Daremos click izquierdo y la rotaremos



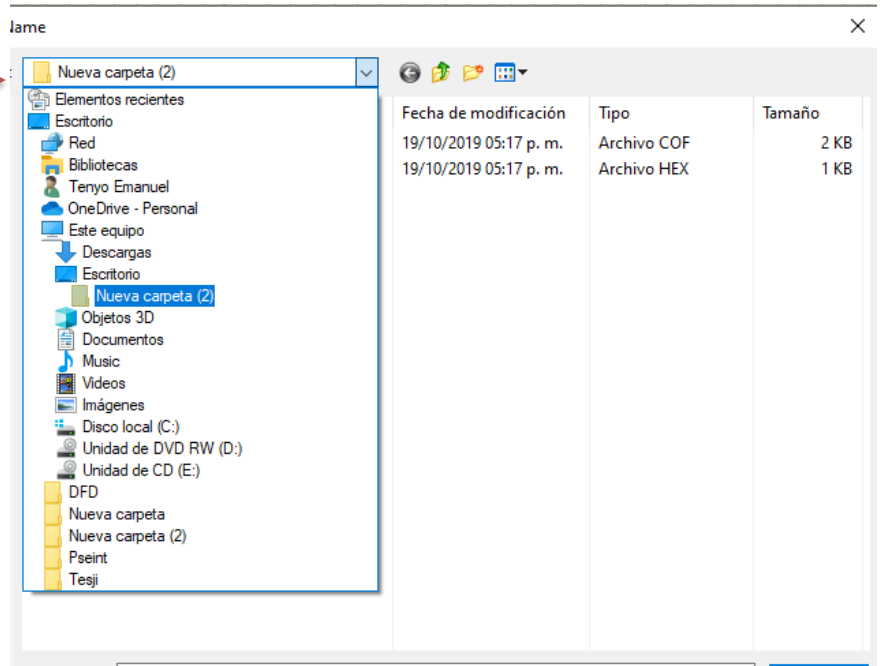
Y la conectamos con la conexión entre la fuente de poder y el capacitor ceramico



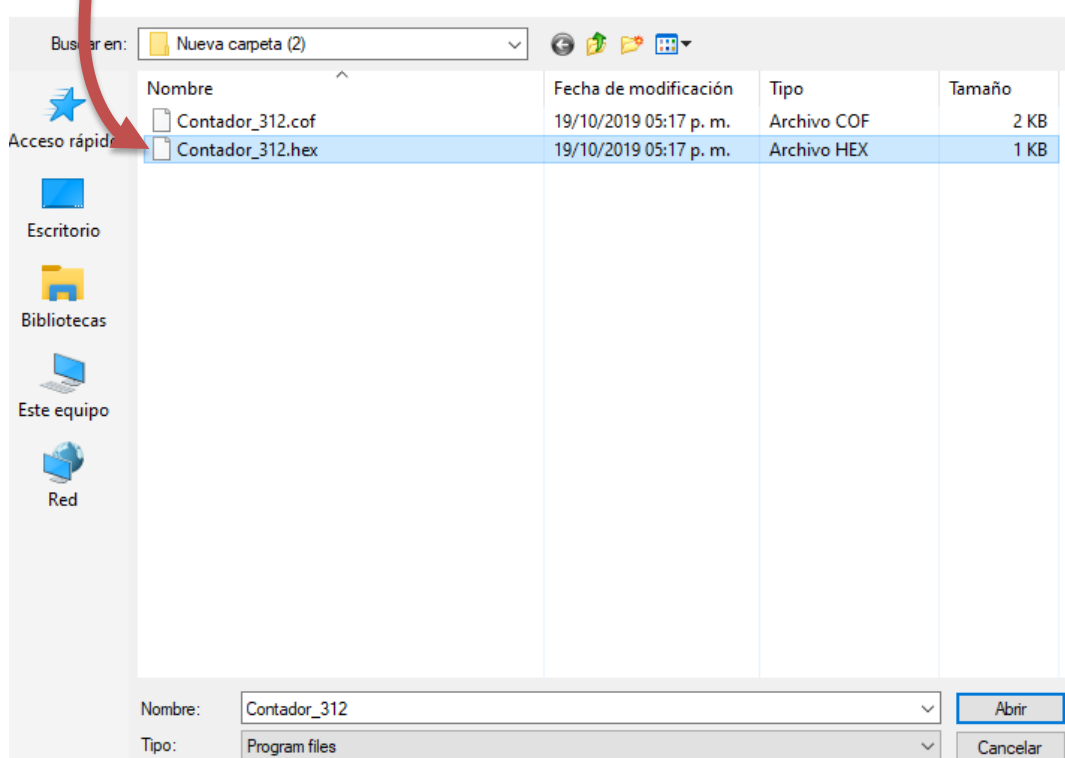
Daremos doble click en uestro PIC y nos aparecera una ventana
Damos click en la carpeta para buscar nuestro archivo compliado anteriormente



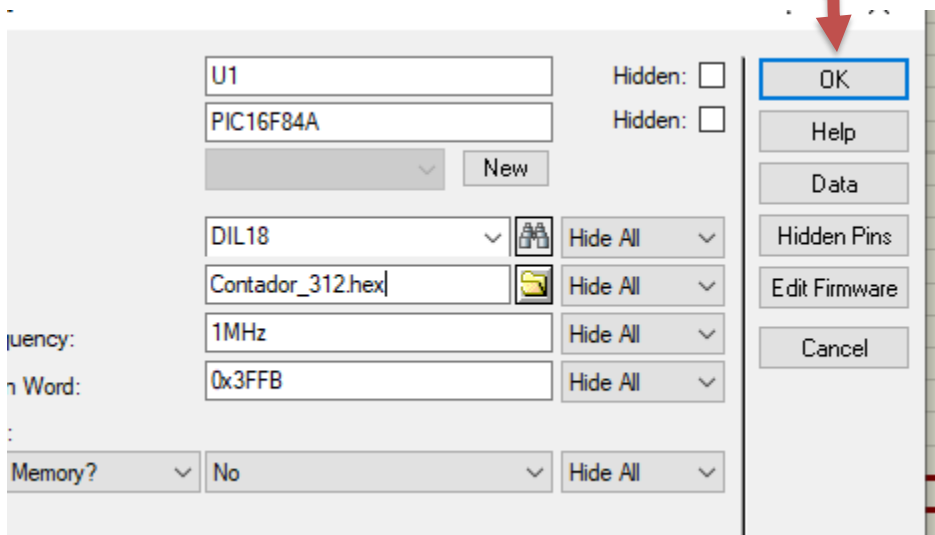
Buscaremos la direccion de nuestro
archivo



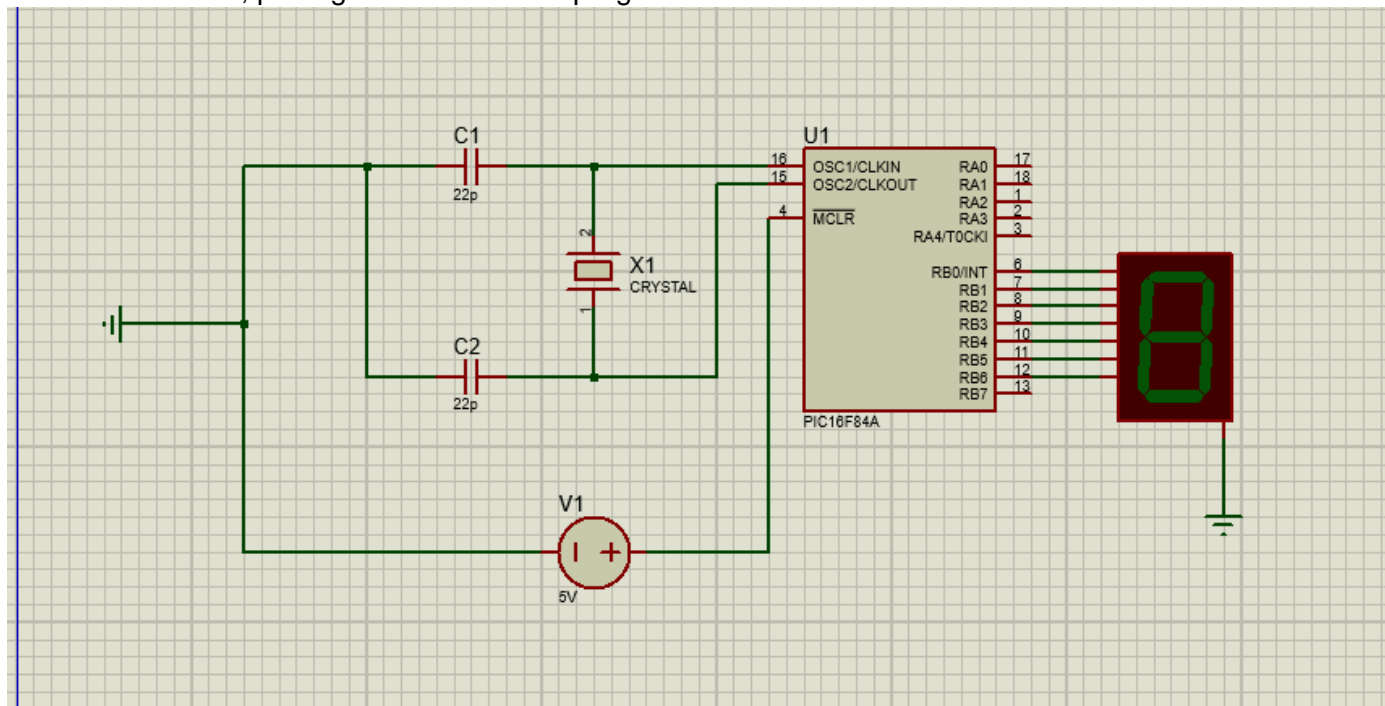
Seleccionaremos el que tenga como extension .hex



Y damos "OK" para confirmar el archivo dentro del PIC



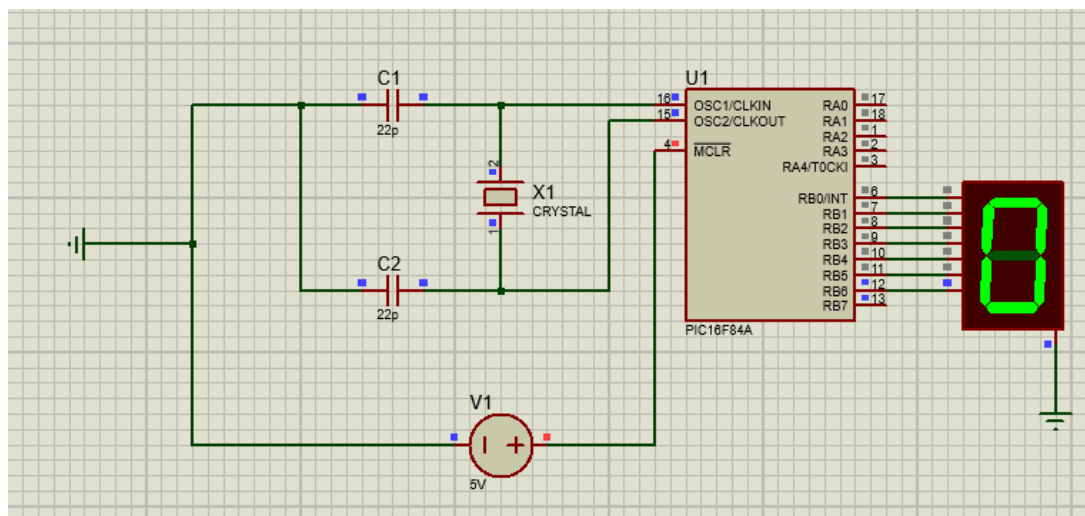
Estando todo listo, proseguimos a correr el programa

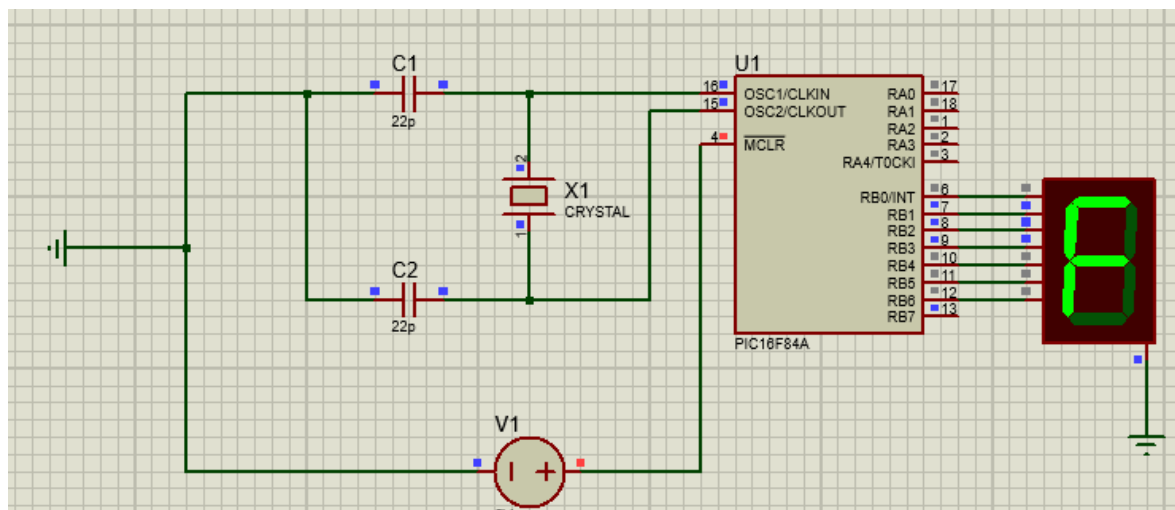
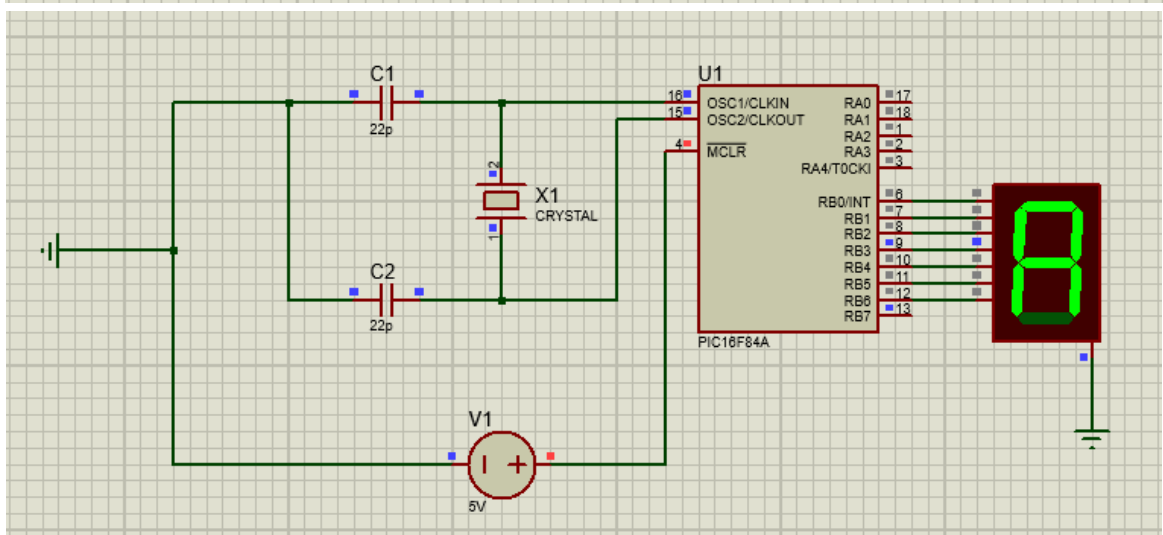
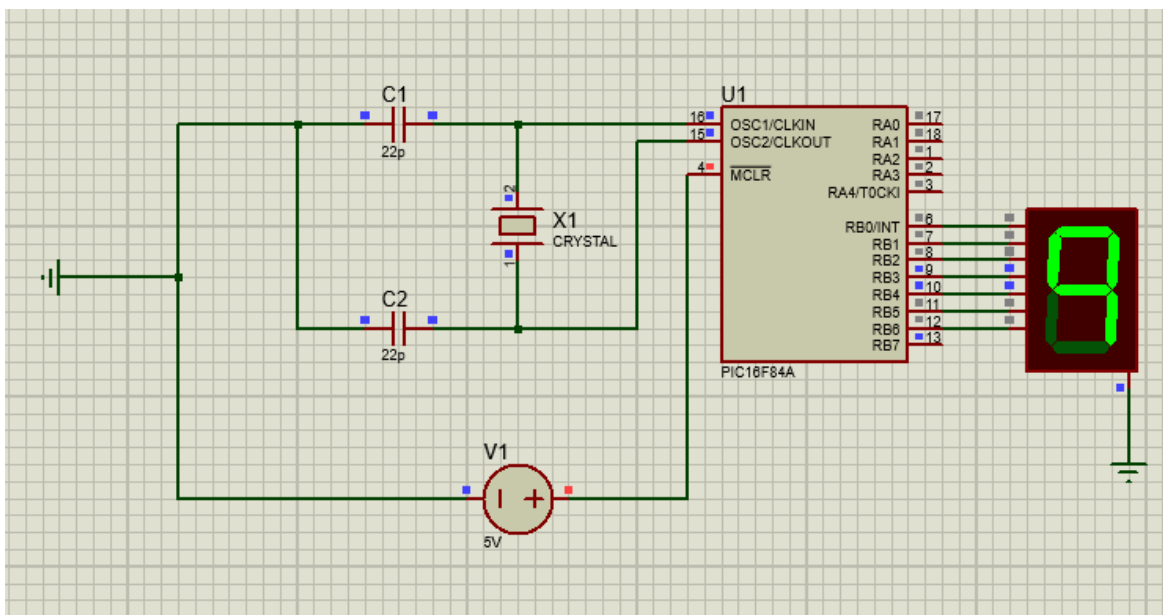


Correr Programa



Observamos que nos muestra del 1-9 y de A-F

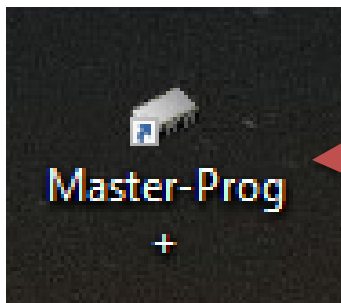




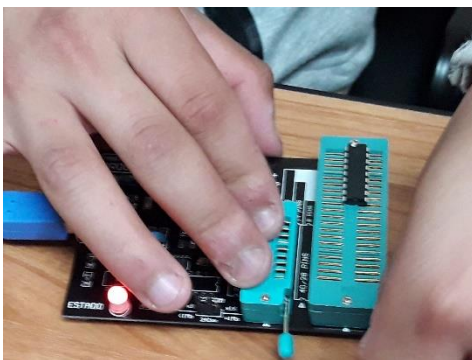
Con esto concluimos que nuestro código está correcto.
Ahora lo pasaremos a un PIC Físico, con ayuda de un Programador para PICs.



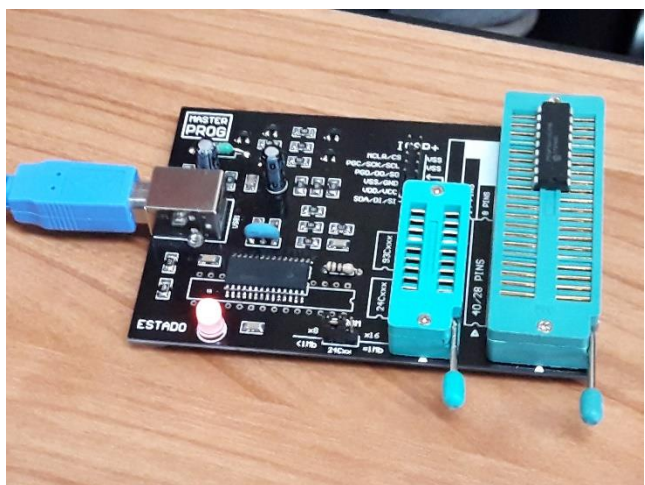
Se instala un software para poder leer el programador y agregar un archivo al PIC



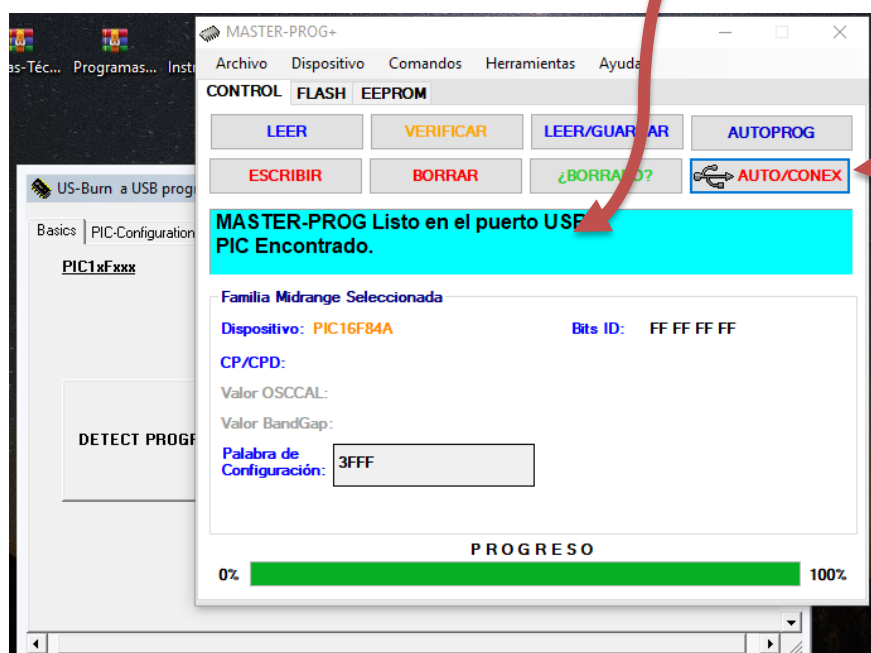
Se coloca el PIC con la media luna hacia adentro y se baja la palanquita



Se conecta con un cable USB a la computadora



Una vez conectado, se abre el programa y detectara el PIC, en caso de que no podemos dar click en "AUTO/CONEX"



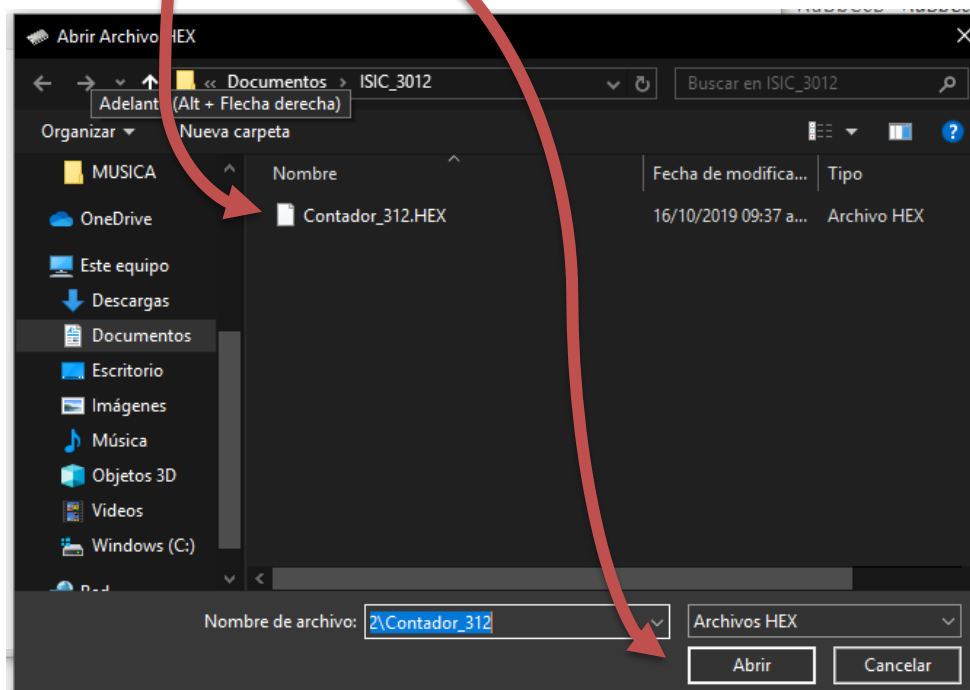
Verificamos para poder saber que trae el PIC



Daremos click en “Escribir” para poder subir nuestro código



Nos abrira una ventana donde debemos buscar nuestro codigo
Lo seleccionamos y lo abrimos



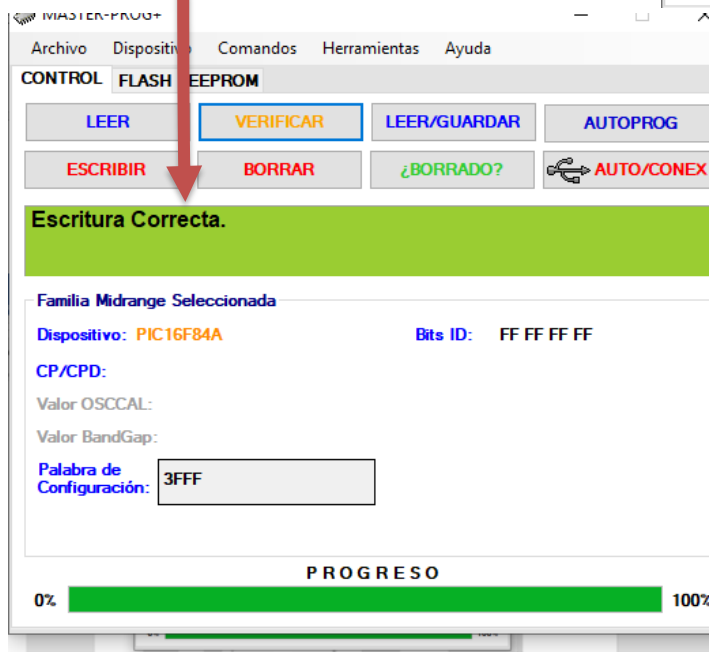
Nos mostrara el progreso de la carga y si logro escribirlo correctamente



Podemos verificar si nuestro PIC fue correctamente programado dando click en “Verificar”



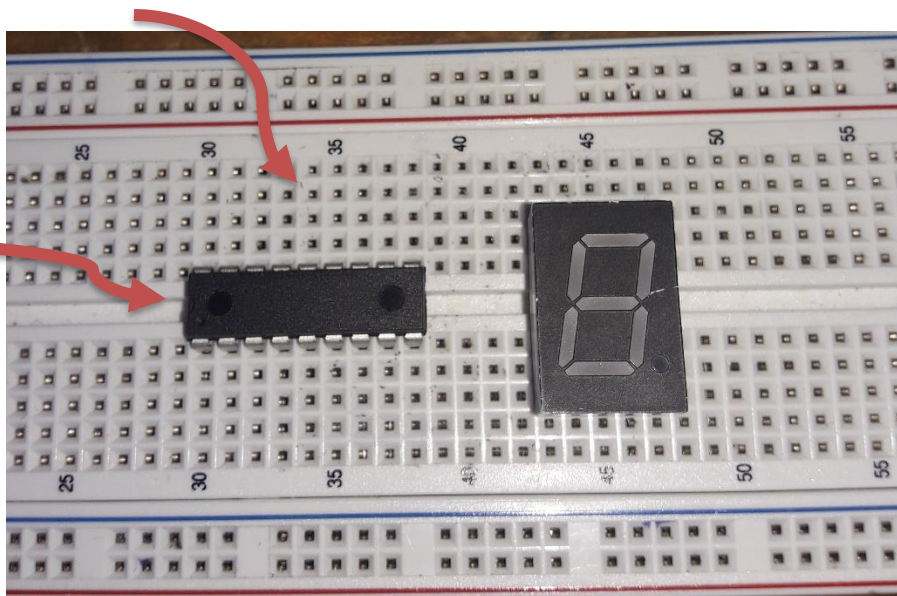
Si fue correctamente saltara el mensaje
“Escritura Correcta”



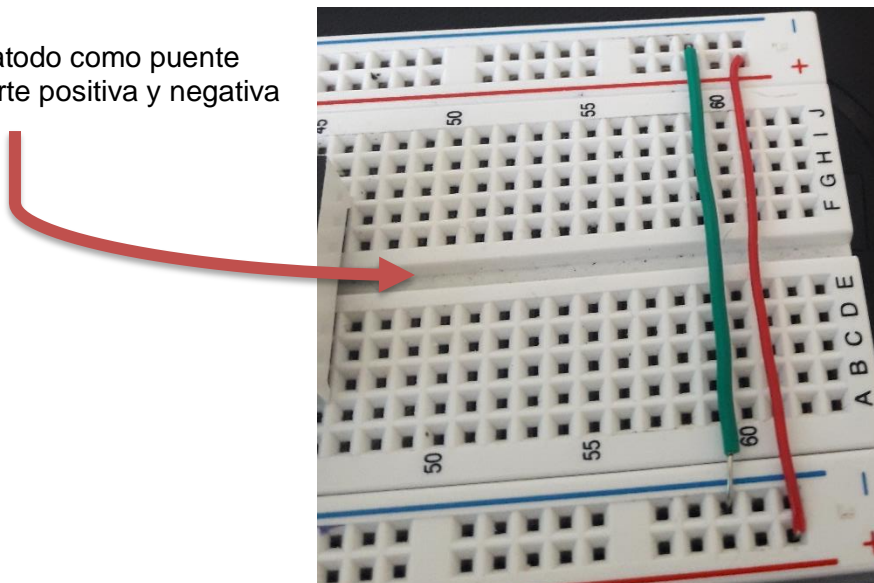
Una vez programado el PIC correctamente lo colocamos en nuestra Protoboard junto con el Catodo comun de 7 Segmento para poder conectarlos con el cable.

Colocamos nuestro PIC y el catodo de esta manera

La media luna del PIC debe estar mirando al lado contrario donde esta el catodo



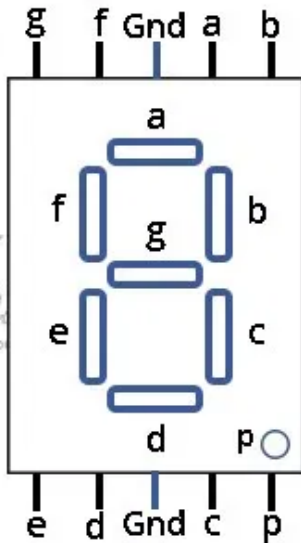
Colocamos dos cables despues del catodo como puente para poder pasar corriente entre la parte positiva y negativa



En el Datasheet del PIC nos muestra una tabla

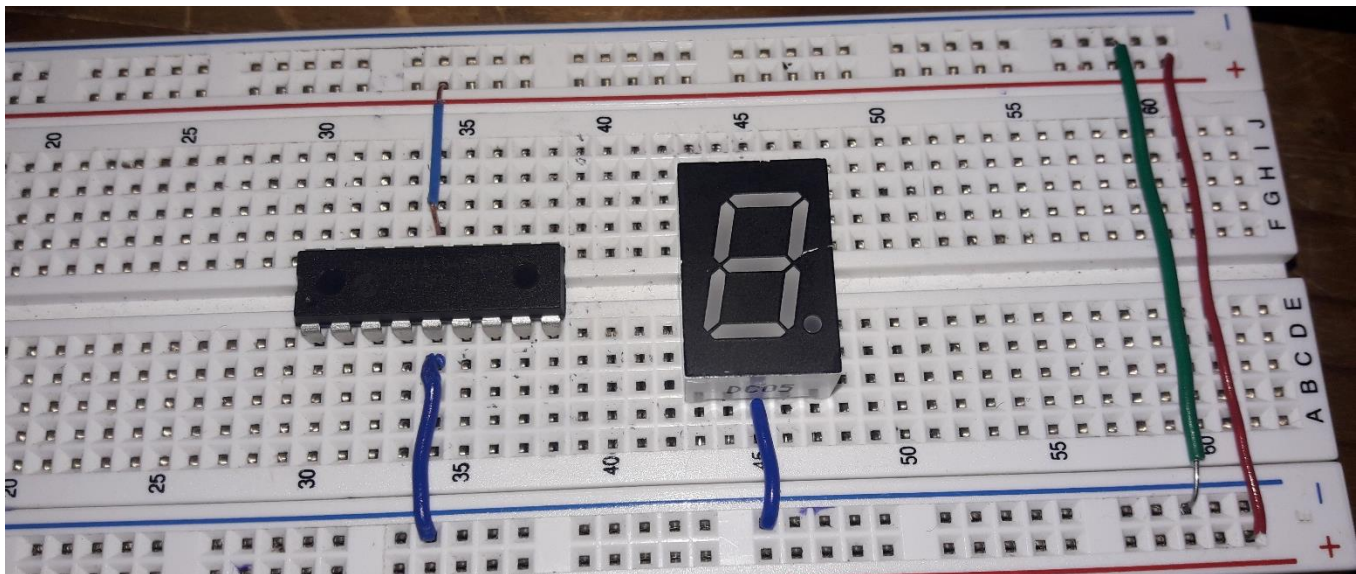
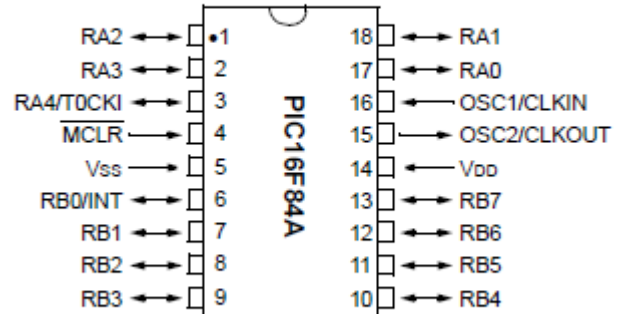
VSS	5	5	5,6	P	—	Ground reference for logic and I/O pins.
VDD	14	14	15,16	P	—	Positive supply for logic and I/O pins.

Y guiandonos en el diagrama que trae, tenemos que conectar un cable a corriente positiva y otro a negativo
Al igual que el catodo



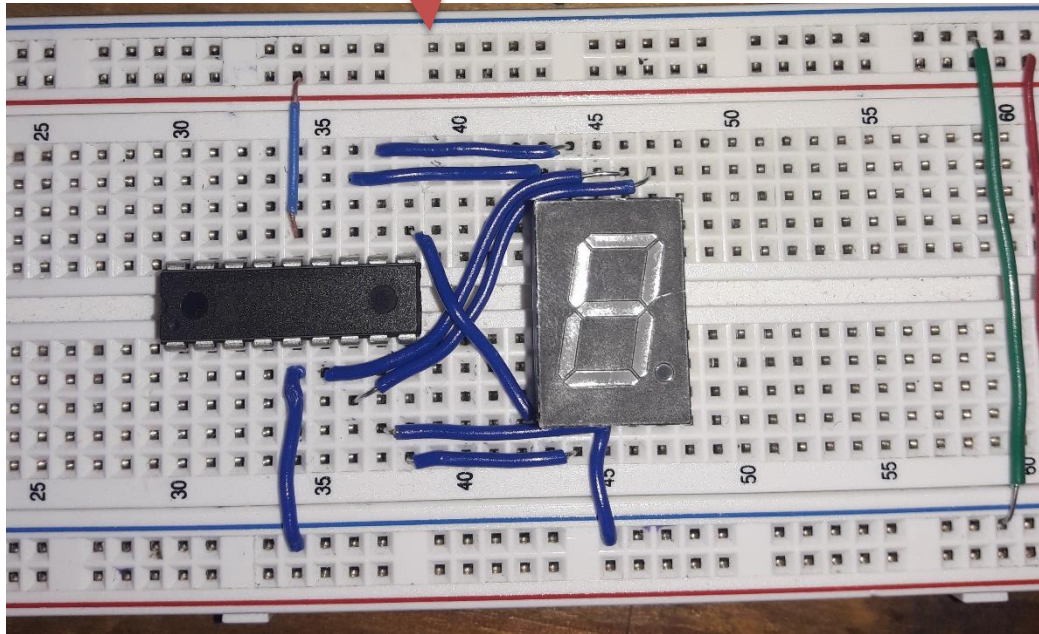
Pin Diagrams

PDIP, SOIC

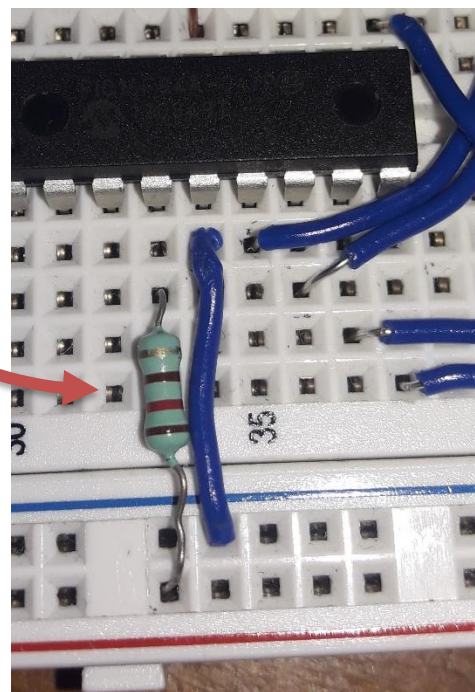


Y con los dos diagramas nos podemos basar para poder completar la conexión entre el PIC y el catodo

Quedandonos de esta manera



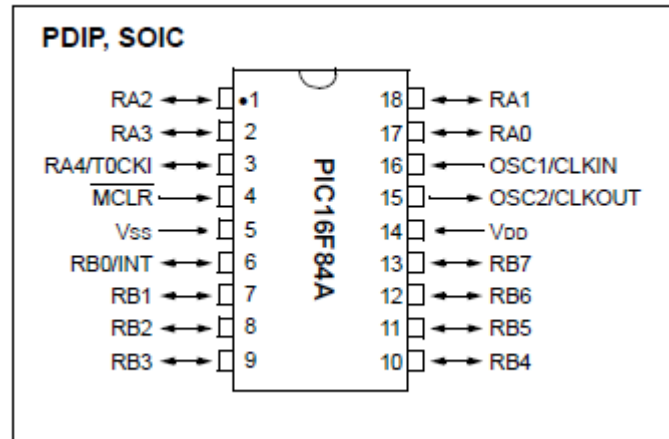
Y le agregamos una resistencia de 120 ohms para que regule energía y evitar que nos queme el PIC



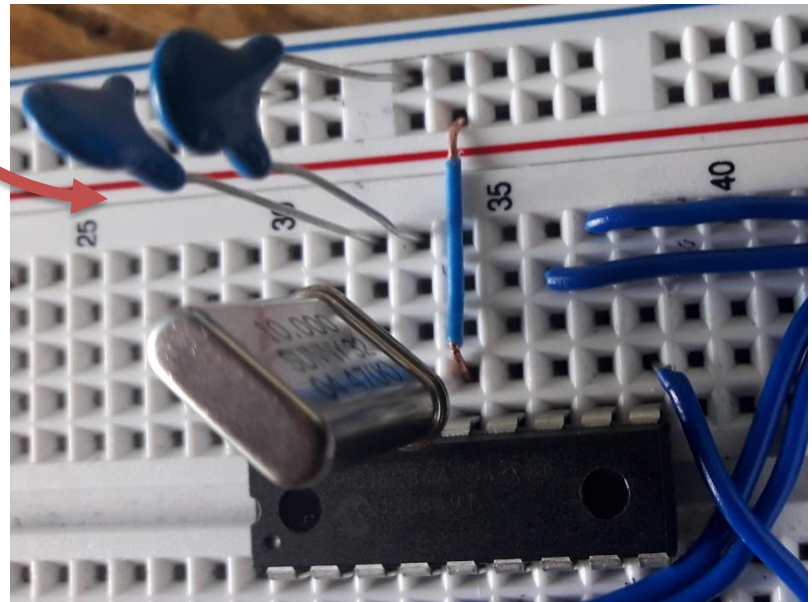
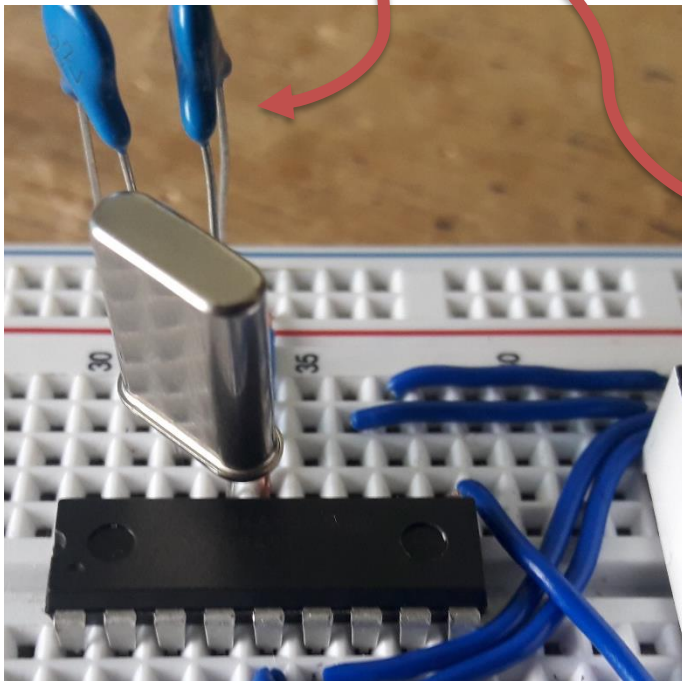
la

En el diagrama tambien nos muestra que necesita un cristal oscilador

Pin Diagrams

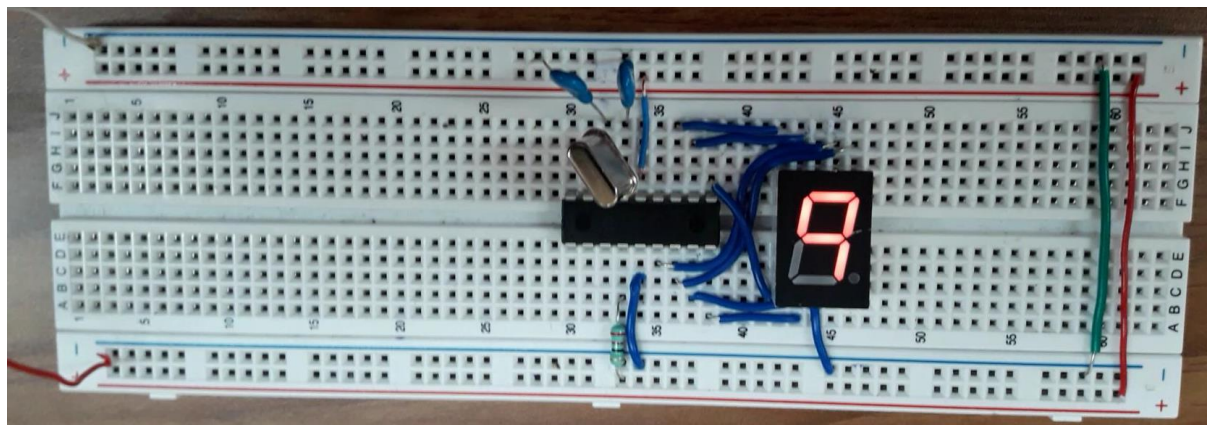
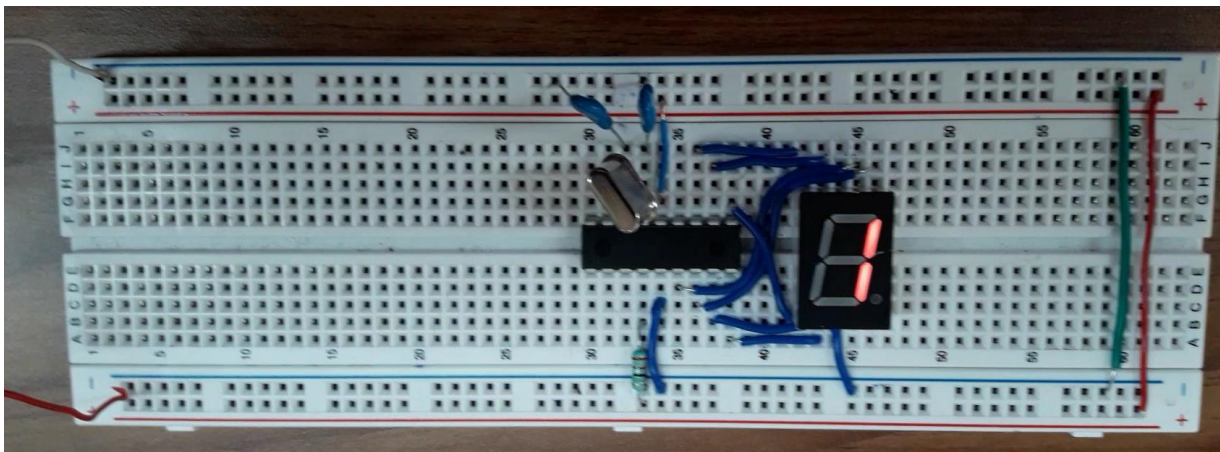
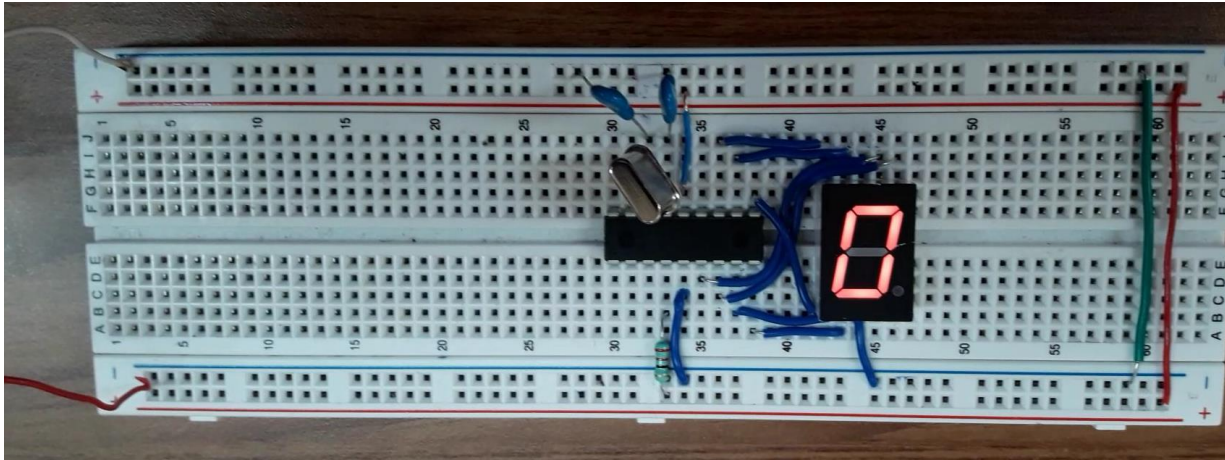


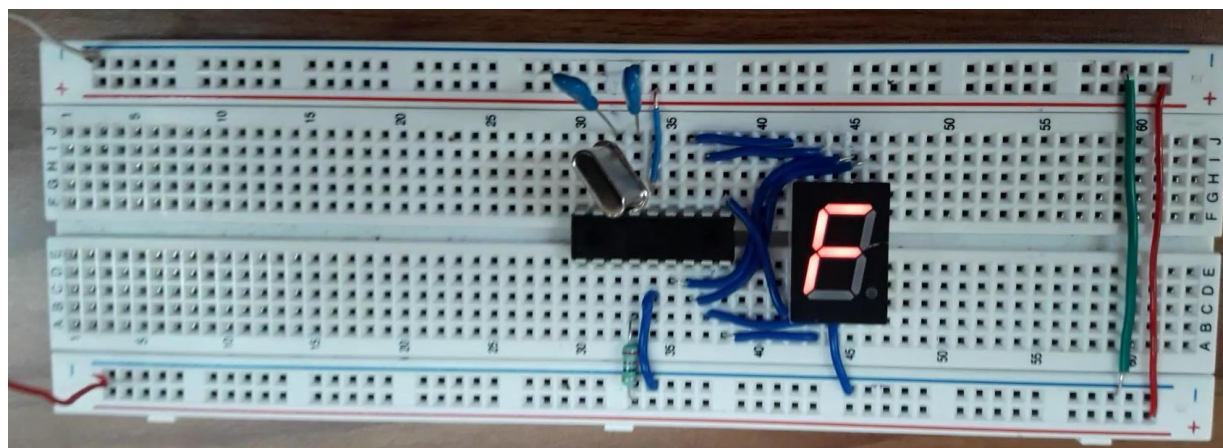
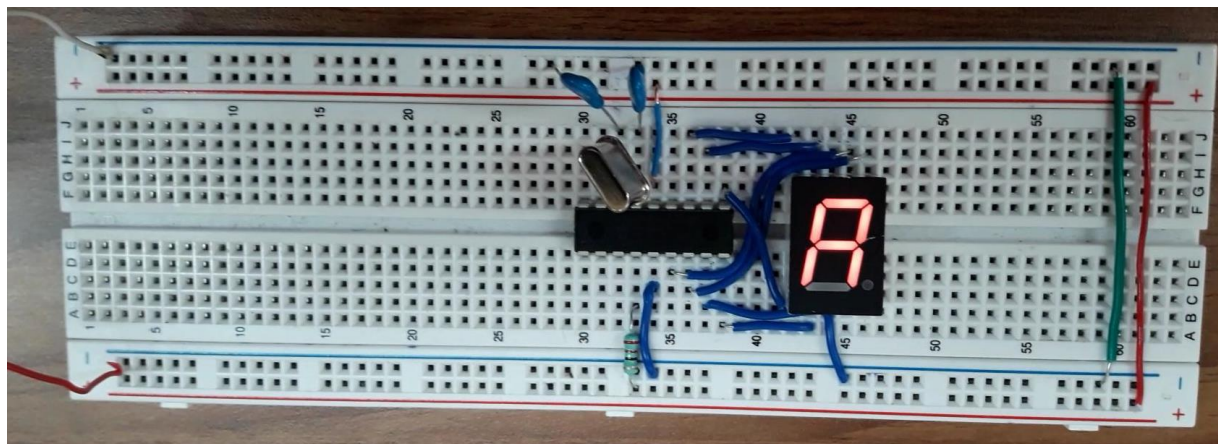
Colocandolos de esta manera



Por ultimo con un cargador de telefono que ya no ocupemos, le quitamos la entrada donde conecta con el telefono y pelamos dos puntas para poder meterlas en la Protoboard

Para comprobarlo lo conectamos a la corriente y a la protoboard y tendrá que correr nuestro Contador Hexadecimal





Con esto terminariamos la practica de nuestro Contador Hexadecimal