



Nombre de la práctica	Contador Ascendente de 7 Segmentos			No.	1
Asignatura:	Arquitectura de Computadoras	de	Carrera:	ISIC	Duración de la práctica (Hrs) 6 hrs

Nombre: Jocelinne Arce Martínez

Grupo: 3012

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

En el aula

## III. Material empleado:

- 1 Tabla Protoboard
- 1 Cátodo de 7 segmentos
- 1 PIC16F84A
- 1 Cristal Oscilador de 5 MHz
- 1 Resistencia de 220 OMS
- 2 Capacitores cerámicos de 22p
- 1 cargador
- 1m de cable delgado

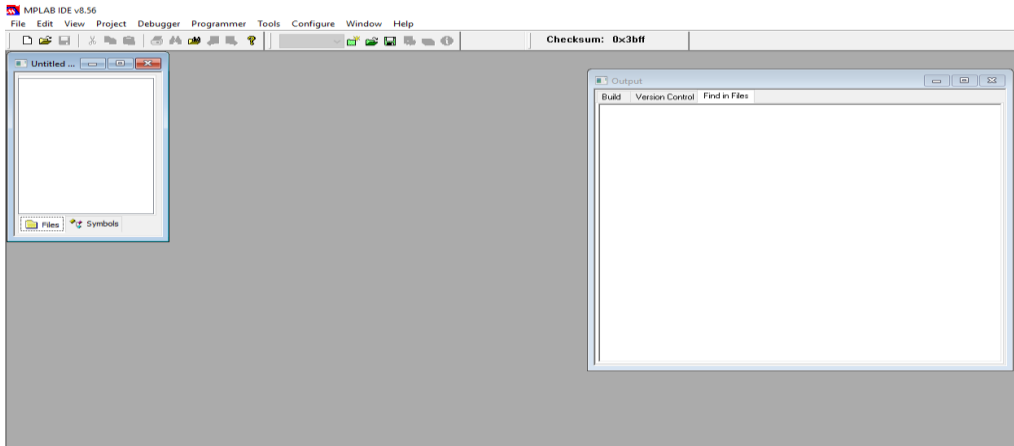
## IV. Desarrollo de la práctica:

Para la creación del contador se siguieron los siguientes pasos que a continuación se presentan:

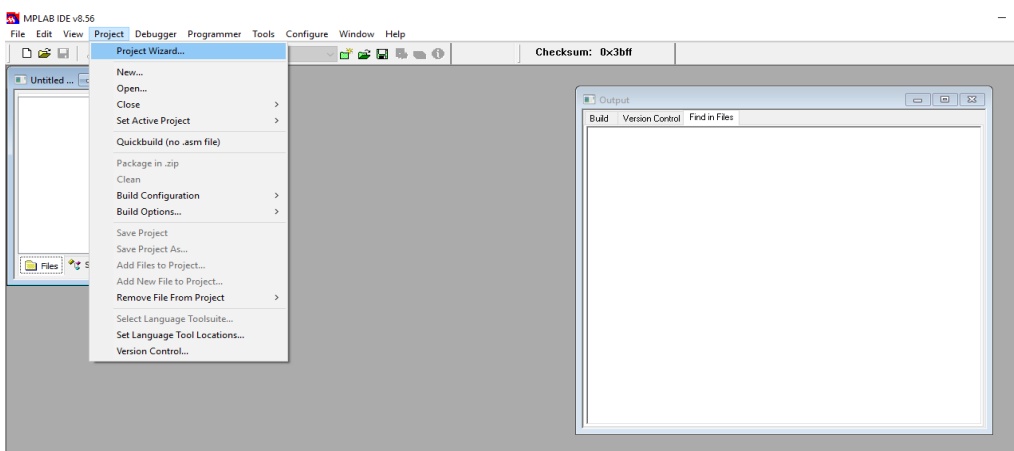
- Instalar en la computadora: MPLAB IDE y Proteus.
- Escribir el código en MPLAB IDEA
- Codificar el código en Proteus
- Programar el PIC16F84A
- Representación en la Tabla Protoboard

- Para la creación del código

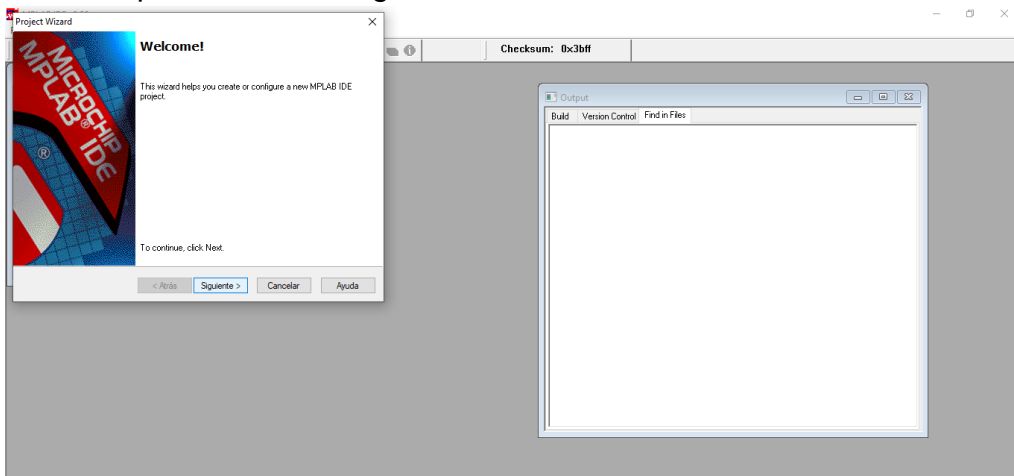
## 1. Abrir MPLAB IDEA



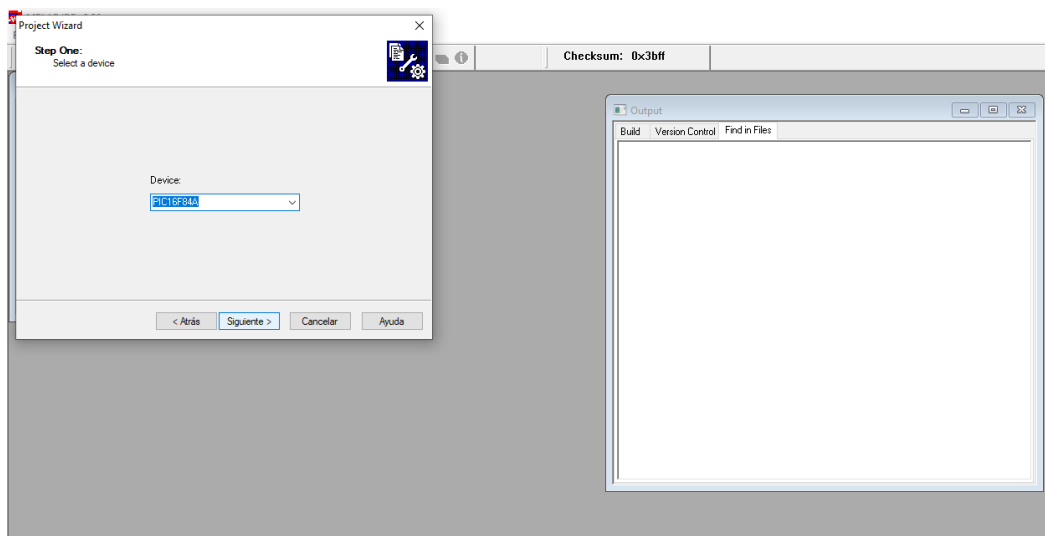
## 2. Seleccionar **Project** (proyecto), ahí seleccionar la opción **Project Wizard**



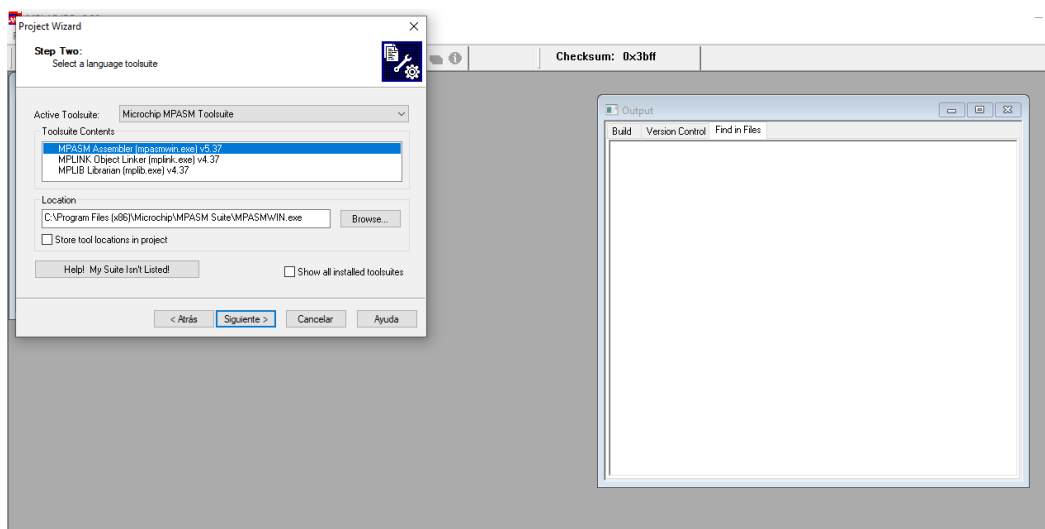
## 3. Nos aparecerá la siguiente ventana, en la cual debemos dar clic en **Siguiente**



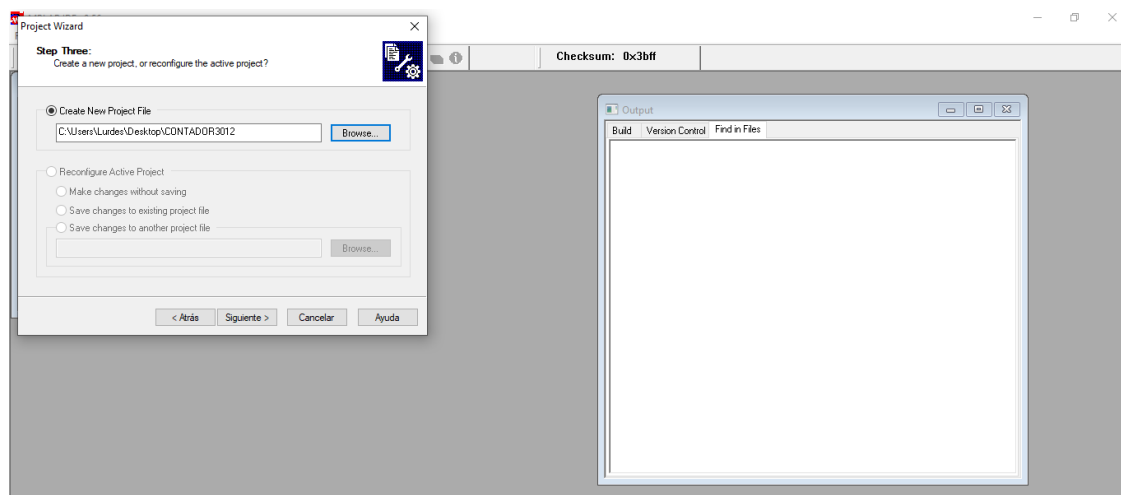
4. Se abrirá lo siguiente, y en ella debemos buscar el pic que vamos a utilizar, el cual es **PIC16F84A** y dar clic en **Siguiente**



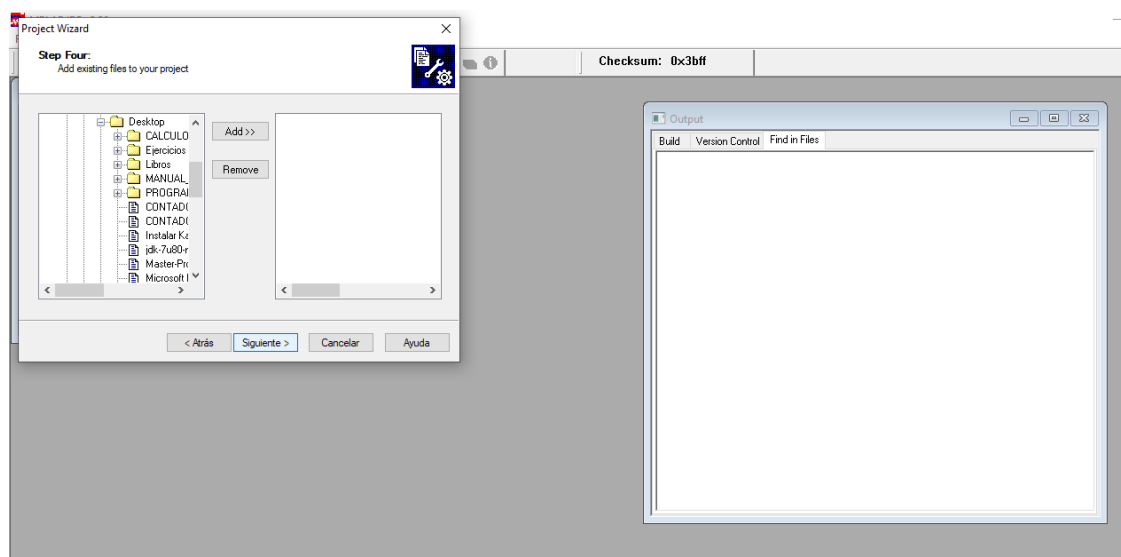
5. Aparecerá la siguiente ventana, ahí debemos seleccionar **MPASM Assembler (mpasmwin.exe) v5.37** y dar clic en **siguiente**



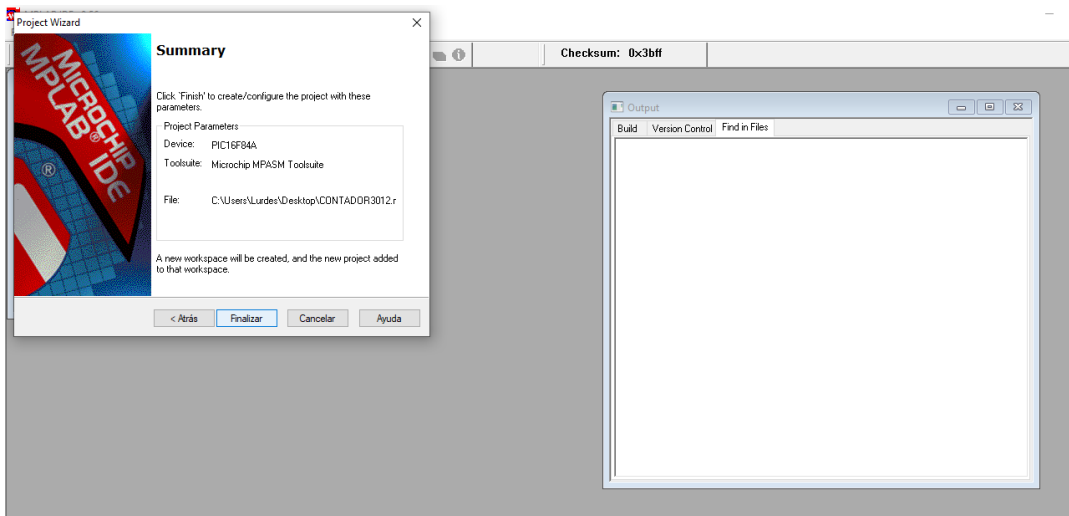
6. En la siguiente ventana debemos guardar el Project que estamos haciendo



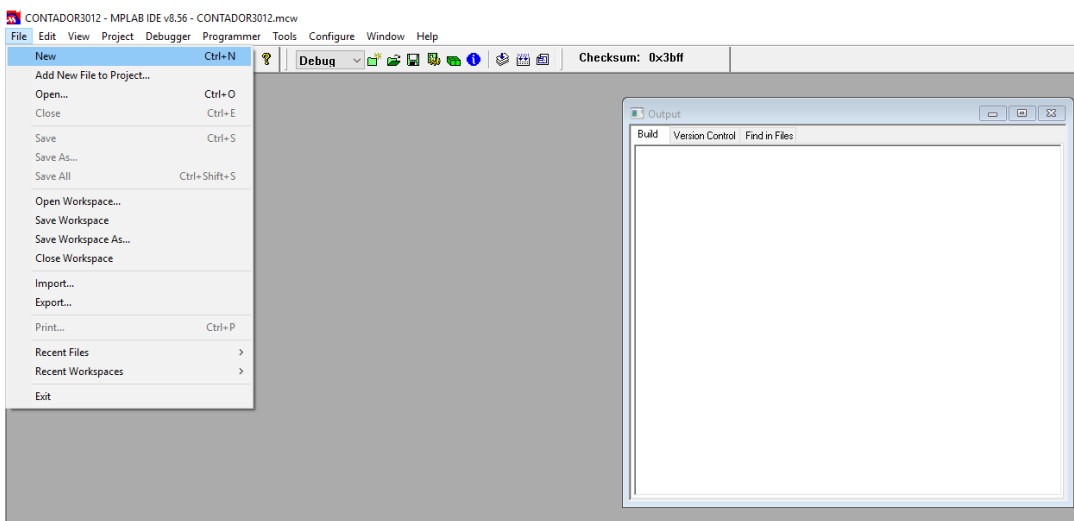
7. En esta ventana debemos seleccionar el archivo que ya se guardó anteriormente, para agregarlo, y después dar click en **siguiente**



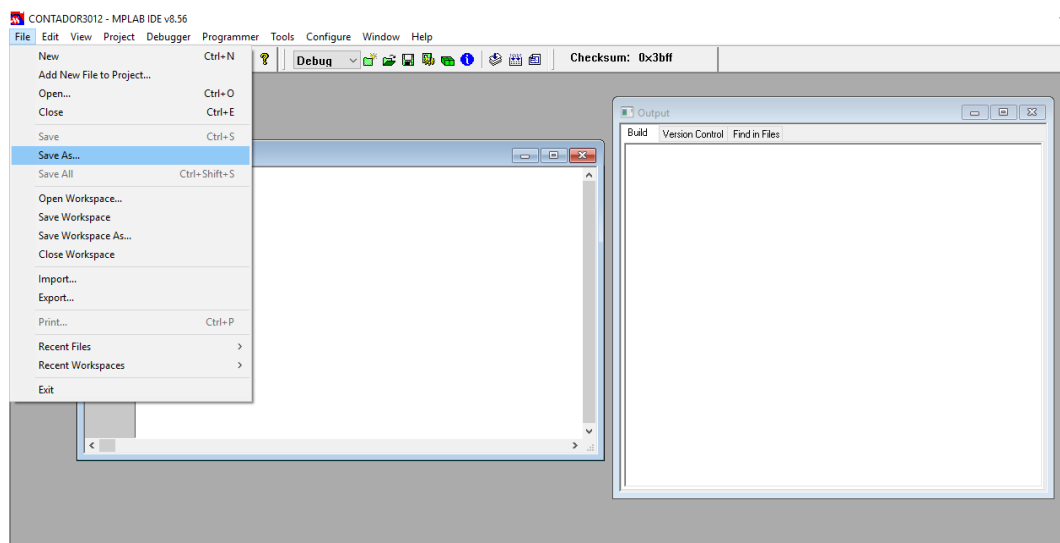
8. Nos aparecerá la siguiente ventana en la cual debemos dar clic en **finalizar**, significa que el archivo se ha guardado correctamente.



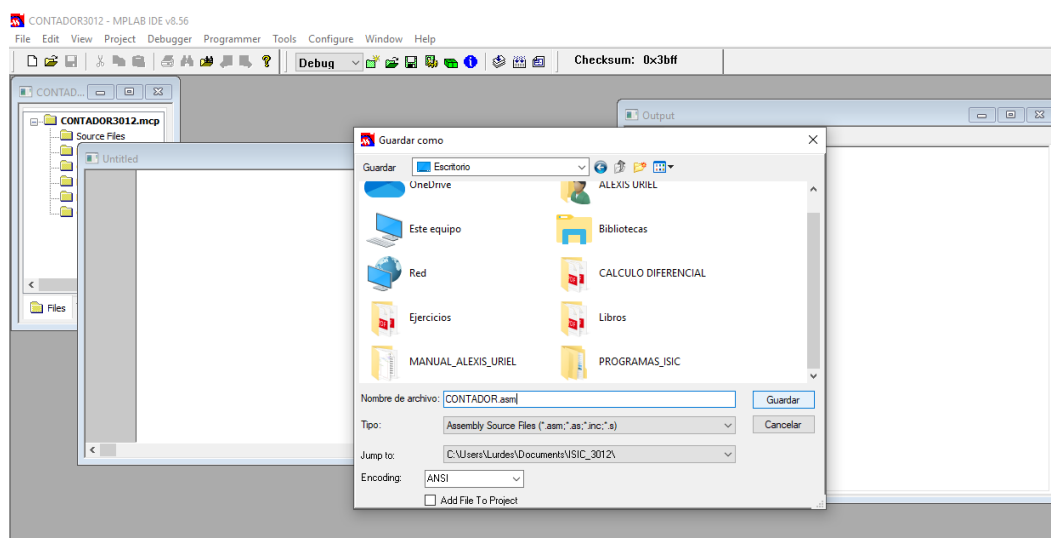
9. Nos aparecerá la siguiente ventana, ahí debemos seleccionar **File**, y dar clic en **New**



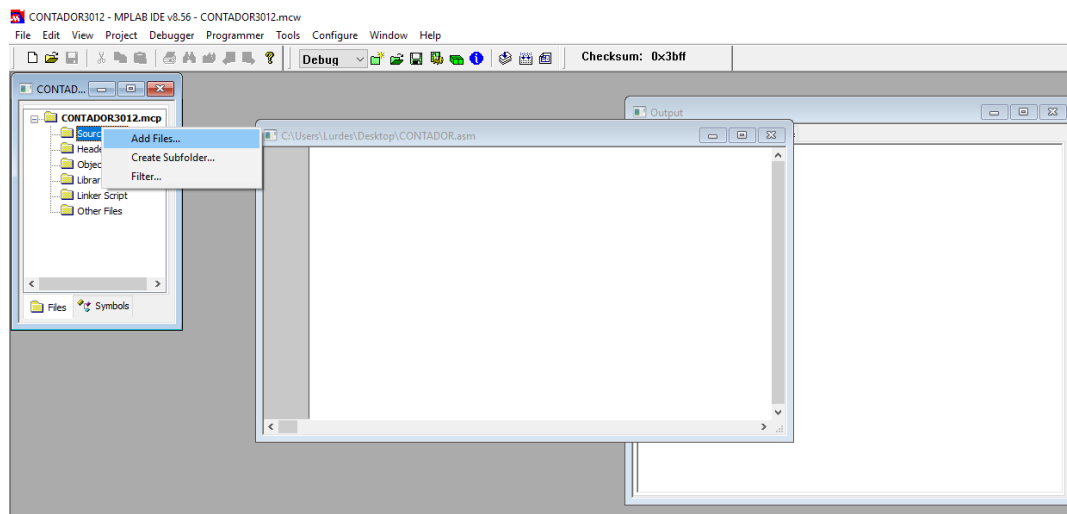
10. Nos aparecerá un nuevo cuadro en blanco, debemos nuevamente seleccionar **File** y dar clic en **Save As**



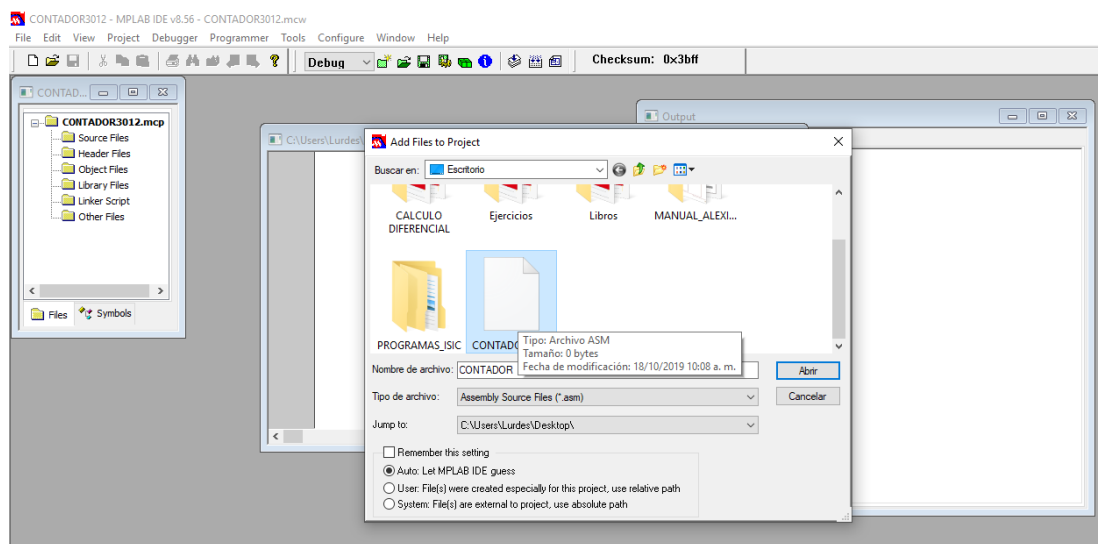
## 11. En esta ventana debemos guardar el archivo



## 12. Debemos dar clic en **Add Files** para agregar archivo guardado anteriormente



13. Nos parecerá la siguiente ventana, en la cual debemos **buscar el archivo** que habíamos guardado anteriormente, ya seleccionado dar clic en **abrir**





14. Por ultimo en el cuadro en blanco que nos aparece escribiremos el siguiente código

Se utilizó código binario para poder hacer que en el cátodo pudieran aparecer los numero del 0-9 y las letras A, b, C, d, E, F.

```

; ZONA DE DATOS *****
__CONFIG __CP_OFF & __WDI_OFF & __PWRTE_ON & __XT_OSC
LIST P=PIC16F84A
INCLUDE <P16F84A.INC>

; ZONA DE VARIABLES *****
CBLOCK 0x0C                ; SE INICIALIZA LA MEMORIA
    NUMERO                 ; VARIABLES QUE LLEVARA EL CONTADOR DE 0-9 Y A-F
    CONTADOR              ; LLEVA EL TIEMPO EN CICLOS DE RELOJ
ENDC
ORG 0                      ; INICIO EN 0
GOTO START                ; CICLO
ORG 5                      ; FIN EN 5

; CONFIGURACION *****
START    BSF    STATUS,5    ; BANCO 1 ACTIVA EL BIT B DE F
        CLRF   TRISB        ; PORT ES LA SALIDA
        MOVWF  0X1F         ; MOVER A LA PARTE BAJA DEL REGISTRO DE MEMORIA RA0 RA4 SON LAS ENTRADAS
        MOVWF  TRISA        ; MOVER A F A TRISA
        MOVWF  B'11000111'  ; ASIGNAR 256 AL TIMER
        MOVWF  OPTION_REG   ; MUEVE AL REGISTRO EL VALOR DEL TIMER
        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

; 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 ES VALOR QUE TOMO DE LA TABLA
        CALL   PAUSE_1000   ; LLAMA A LA FUNCION PAUSE
        INCF   NUMERO,F     ; REALIZA UN INCREMENTO DE LA VARIABLE EN 1
        MOVF   NUMERO,W     ; SE CARGA LA VARIABLE EN W
        XORLW  0X10         ; SE COMPARA SI ES QUE LLEGA AL REGISTRO DOS
        BTFS   STATUS,Z     ; VERIFICA SI HA LLEGADO
        GOTO   MAIN        ; SE REGRESA AL 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 *****
; PAUSE_1000 SE REALIZA DE 1 SEGUNDO *****
PAUSE_1000 MOVWF 0X02        ; SE LE ASIGNA 1000 AL CONTADOR
        MOVWF  CONTADOR     ; SE MUEVE EL VALOR DEL CONTADOR DE A A F
        MOVWF  CONTADOR     ; SE MUEVE EL VALOR DEL CONTADOR DE A A F
        BCF    INTCON,TOIF   ; SE LIBERA EL BIT DE DESBORDAMIENTO TMR0
        MOVWF  09           ; SE CARGA EL 217 A W
        MOVWF  TMR0         ; A TMR0
        BTFS   INTCON,TOIF   ; SE LIBERA EL BIT DE DESBORDAMIENTO DEL TMR0
        GOTO   DELAY2       ; BUCLE DEL DELAY2
        DECF   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

```



Si el código es correcto aparece lo siguiente:

```
MPLAB IDE v8.56 - [Output]
File Edit View Project Debugger Programmer Tools Configure Window Help
Build Version Control Find in Files
Checksum: 0xd0b5

Release build of project 'C:\Users\Lurdes\Documents\SIC_3012\Contador_312 disposable_mcp' started.
Language tool versions: MPASMWIN.exe v5.37, mplink.exe v4.37, mplib.exe v4.37
Fri Oct 18 10:13:11 2019

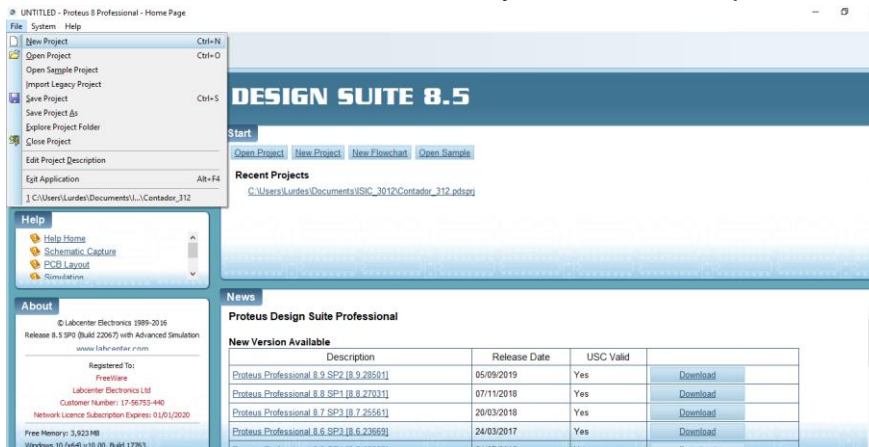
Clean: Deleting intermediary and output files.
Clean: Done.
Executing 'C:\Program Files (x86)\Microchip\MPASM Suite\MPASMWIN.exe' /q /p16F84A "Contador_312.asm" /f "Contador_312.lst" /e "Contador_312.err"
Warning[205] C:\Users\Lurdes\Documents\SIC_3012\CONTADOR_312.ASM 2: Found directive in column 1: ( _CONFIG)
Warning[205] C:\Users\Lurdes\Documents\SIC_3012\CONTADOR_312.ASM 3: Found directive in column 1: (JUMP)
Warning[205] C:\Users\Lurdes\Documents\SIC_3012\CONTADOR_312.ASM 4: Found directive in column 1: (INCLUDE)
Warning[205] C:\Users\Lurdes\Documents\SIC_3012\CONTADOR_312.ASM 7: Found directive in column 1: (GLOCK)
Warning[205] C:\Users\Lurdes\Documents\SIC_3012\CONTADOR_312.ASM 10: Found directive in column 1: (ENDC)
Message[302] C:\Users\Lurdes\Documents\SIC_3012\CONTADOR_312.ASM 16: Register in operand not in bank 0. Ensure that bank bits are correct.
Message[302] C:\Users\Lurdes\Documents\SIC_3012\CONTADOR_312.ASM 18: Register in operand not in bank 0. Ensure that bank bits are correct.
Message[302] C:\Users\Lurdes\Documents\SIC_3012\CONTADOR_312.ASM 20: Register in operand not in bank 0. Ensure that bank bits are correct.
Warning[205] C:\Users\Lurdes\Documents\SIC_3012\CONTADOR_312.ASM 67: Found directive in column 1: (ENDC)
Executing 'C:\Program Files (x86)\Microchip\MPASM Suite\mplink.exe' /p16F84A "Contador_312.o" /z __MPLAB_BUILD=1 /o "Contador_312.col" /M "Contador_312.map" /W /x
MPLINK 4.37: Linker
Copyright (c) 1998-2010 Microchip Technology Inc.
Errors:
Loaded C:\Users\Lurdes\Documents\SIC_3012\Contador_312.cof

Release build of project 'C:\Users\Lurdes\Documents\SIC_3012\Contador_312 disposable_mcp' succeeded.
Language tool versions: MPASMWIN.exe v5.37, mplink.exe v4.37, mplib.exe v4.37
Fri Oct 18 10:13:13 2019

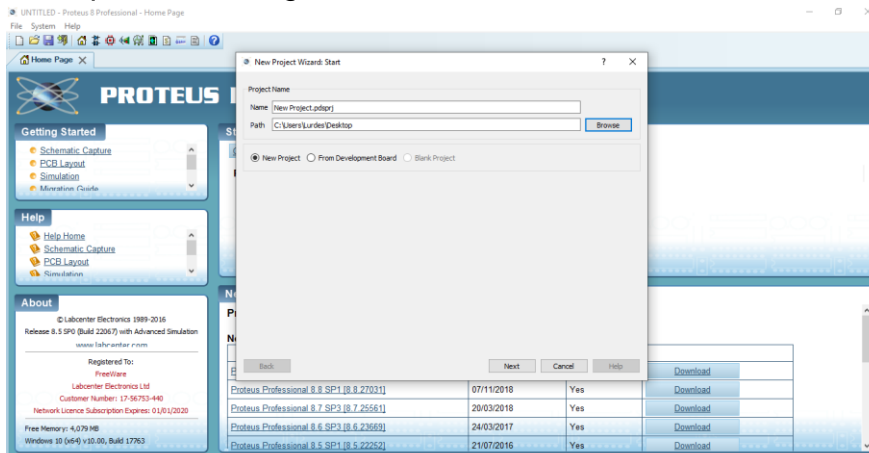
BUILD SUCCEEDED
```

- Copiar el código. Utilizamos **Proteus**

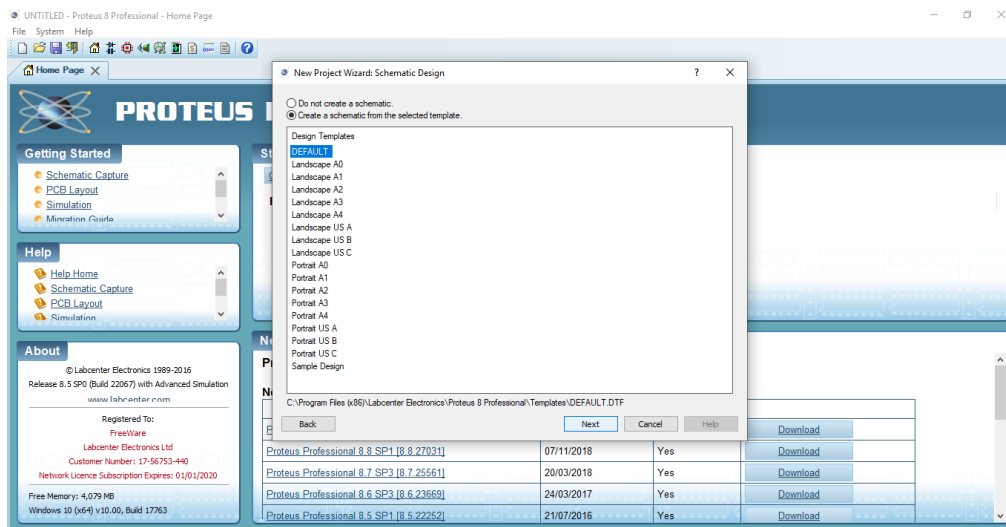
1. Abrimos Proteus, Seleccionar **File** y dar clic en la opción **New Project**



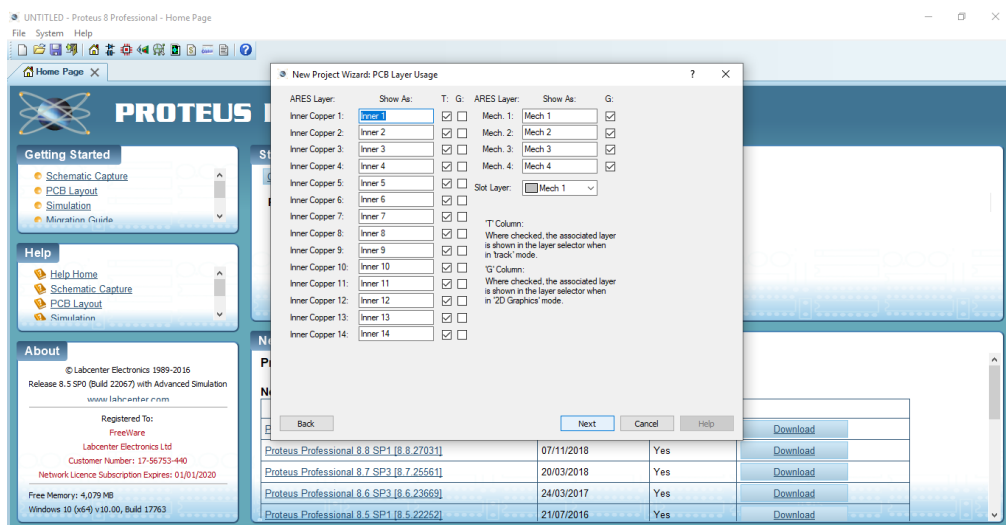
2. Nos aparecerá la siguiente ventana, ahí debemos dar clic en **Next**



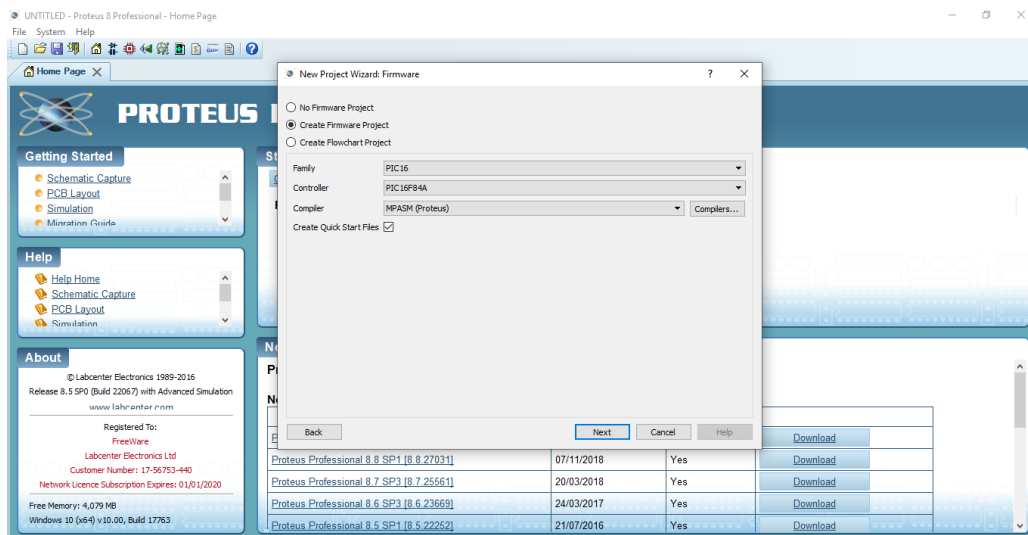
### 3. En la siguiente ventana debemos seleccionar **DEFAULT** y dar clic en **Next**



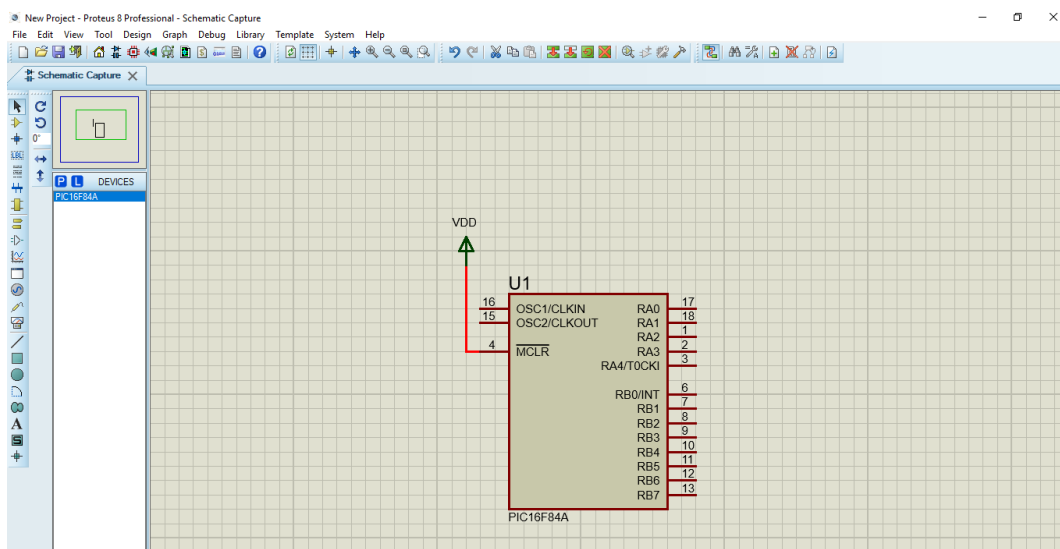
### 4. Nos aparecerá la siguiente ventana, debemos seleccionar **Inner 1** y dar clic en **Next**



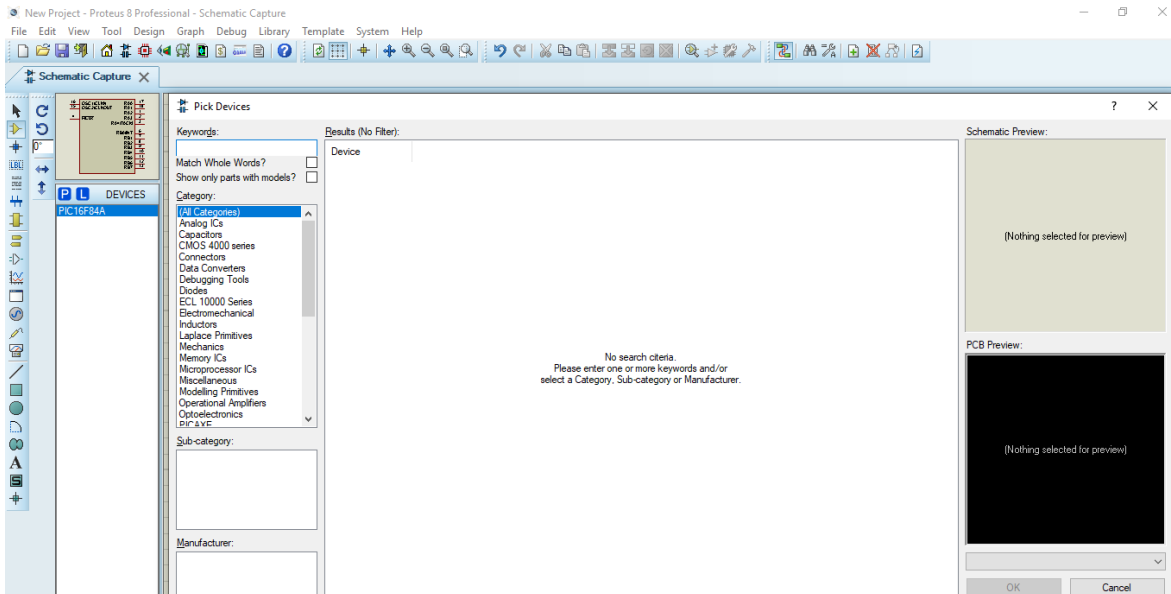
### 5. En esta ventana debemos seleccionar **Create Firmware Project** y dar clic en **Next**



6. Nos aparecerá esto que representa el PIC16F84A, debemos eliminar la flecha

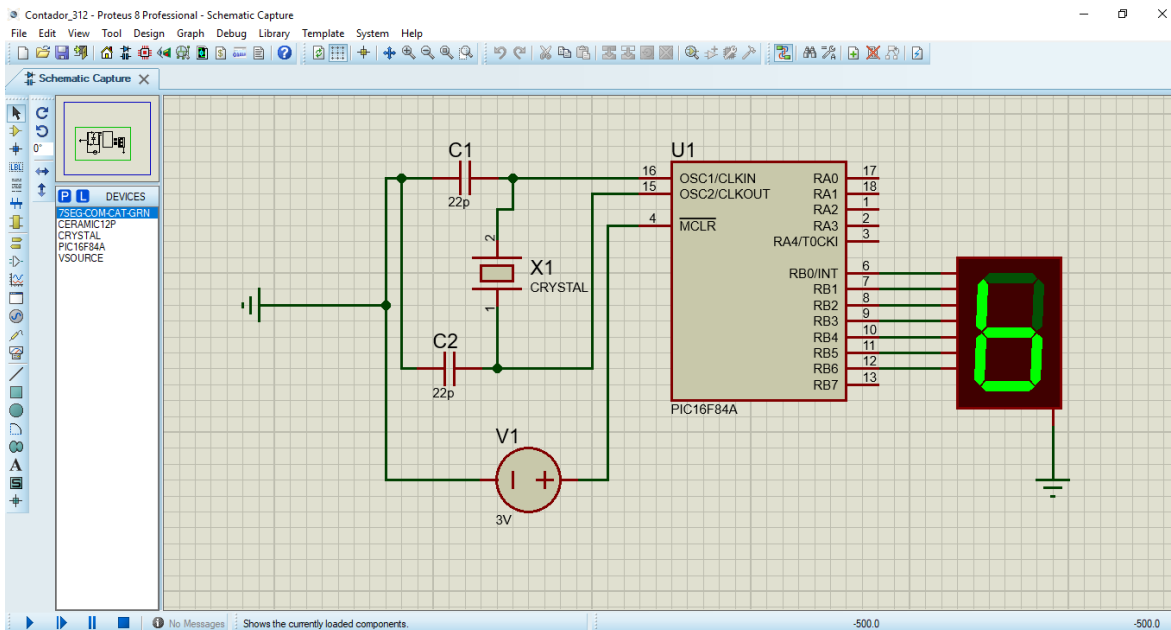


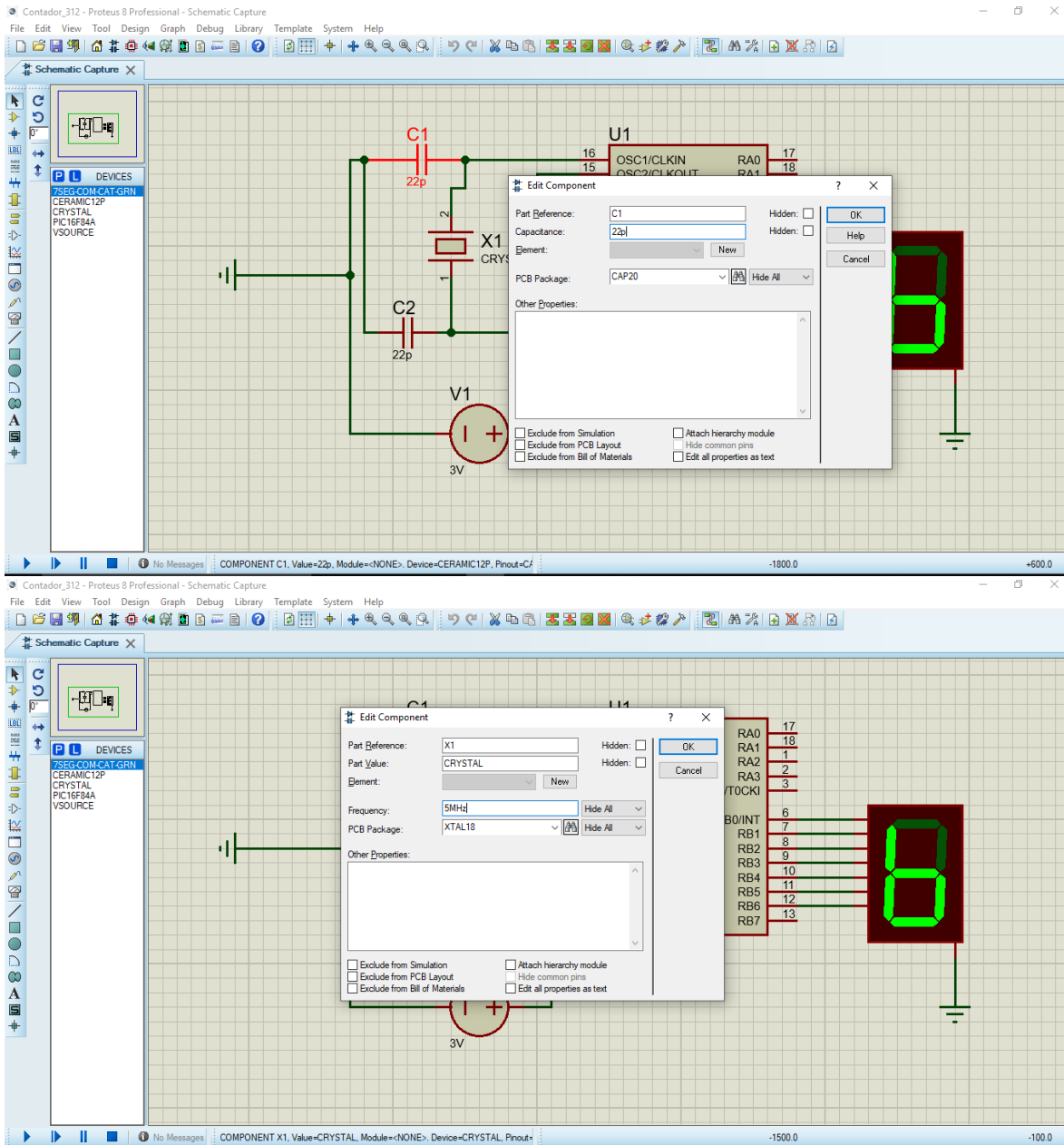
7. Debemos buscar los materiales para la creación del circuito en la tabla proto



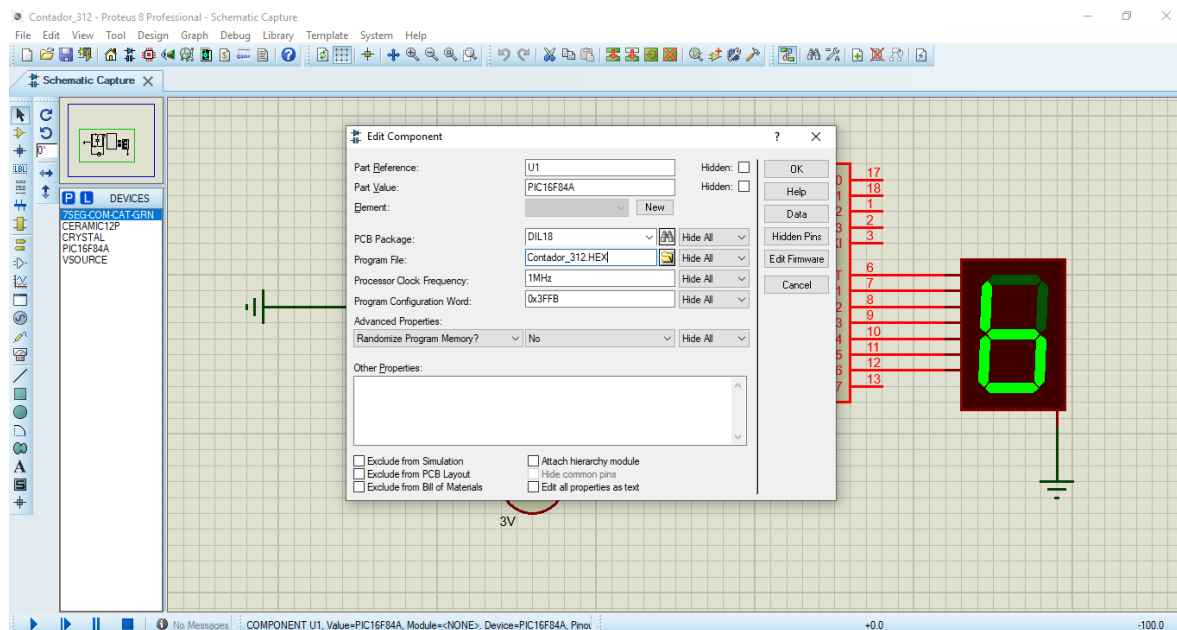
8. Ya habiendo terminado nuestro circuito hay que cambiar la intensidad de:

- Capacitores a 22p
- Crystal 5 Hz
- Pila a 3V

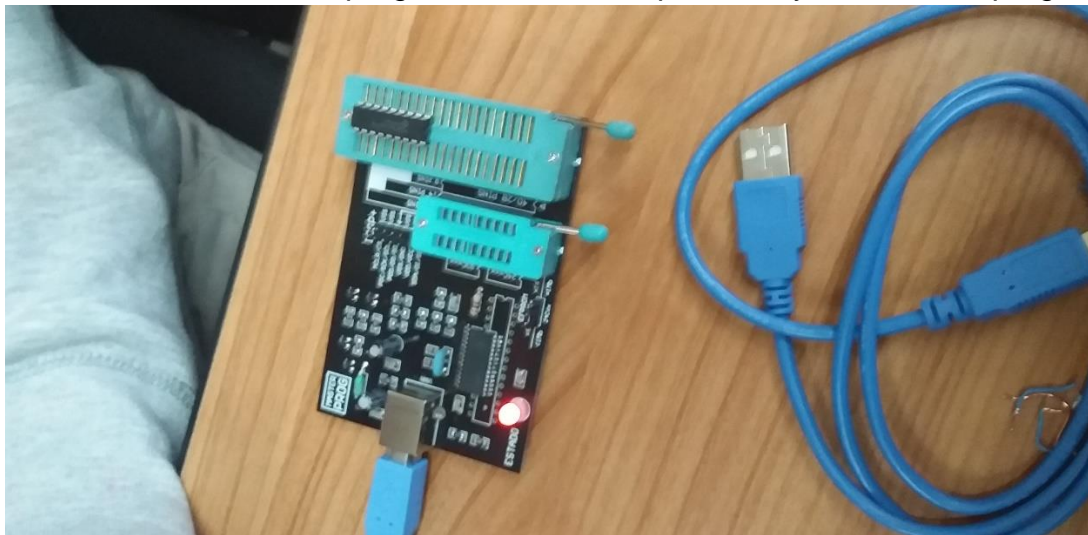




9. Debemos dar doble clic al microcontrolador nos aparecerá la siguiente ventana, ahí debemos de buscar el nombre de nuestro archivo hecho anteriormente en MPLAB IDE con extensión HEX
10. Por último, lo corremos y comprobaremos que simulará lo que sucederá a continuación en nuestra tabla Proto, pues en el cátodo de 7 segmentos correrán los numero del 0-9 y las letras A, b, C, d, F.



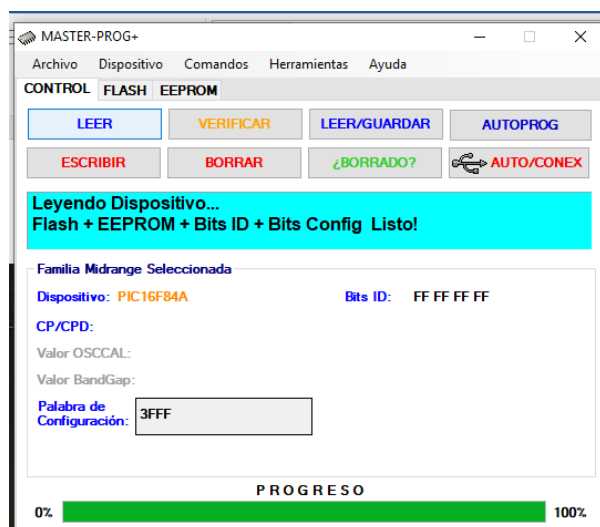
- Programación del PIC16F84A
1. Se conecta el cable del programador a la computadora y el PIC en el programador



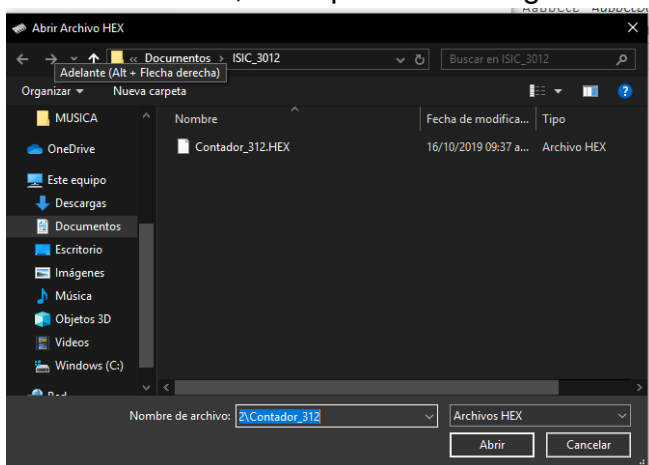
2. Abrimos en la computadora el Programa MASTER-PROG+



### 3. Dar clic en **Auto/Conex**



### 4. Dar clic en **Leer**, nos aparecerá la siguiente ventana ahí debemos buscar el archivo





## 5. Debemos esperar a que cargue



## 6. Debemos dar clic en **ESCRIBIR**, esperar para que la programación sea correcta.



## 7. Como último paso para la programación de nuestro Pic debemos dar clic en **VERIFICAR**, este como el nombre indica verificara si la escritura es correcta.



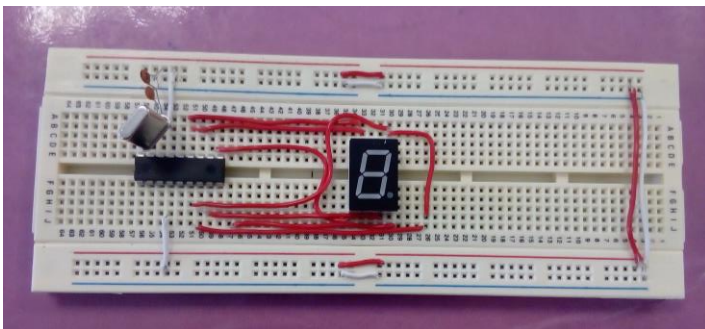


- En la tabla Protoboard

Con base a todo lo que hicimos anteriormente continuamos con la representación en físico

1. El esquema echo en Proteus se pasó a la tabla Protoboard, utilizamos cable para la conexión de los materiales
2. Checamos si funcionada, conectando energía eléctrica a la tabla con un cargador con cables pelados, con esto observamos que en el cátodo aparecieron de manera continua los números del 0-9 y las letras A, b, C, d y F.

Este es el resultado:





## V. Conclusiones:

Puedo concluir que si es posible simular algo en digital y pasarlo a un hardware. Todo ello requirió conocimiento sobre hardware, software y código ensamblador, además la utilización de programas que nos ayudaron a la simulación de condiciones.

Con ayuda de MPLA IDE, Proteus y el Programador pudimos hacer software y meterlo en un hardware. Lo que se observó al hacer esto es lo que ya se había observado en algún momento en Proteus pues el Cátodo de 7 segmentos represento todo lo programado.