



<b>Nombre de la práctica</b>	<b>Contador con ensamblador</b>			<b>No.</b>	<b>1</b>
<b>Asignatura:</b>	Arquitectura de Computadoras	<b>Carrera:</b>	Sistemas Computacionales	<b>Duración de la práctica (Hrs)</b>	<b>4</b>

### I. Competencia(s) específica(s):

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

Aula n3

### III. Material empleado:

- MPLAP ID v8.56
- Proteus8 Profesional
- Tarjeta Protoboar
- Cátodo Común de 7 segmentos
- Crystal Oscilador
- Cable para proto
- PIC16F84A
- Capacitor Cerámico
- Resistencias

### IV. Desarrollo de la práctica:

Se utilizo el software de Proteus8, para el diseño del circuito del contador ascendente, digamos que fue para ver como quedaba como una prueba antes de pasarlo a fisico.



Para insertar lo que vamos a ir utilizando en el Proteus se le clic en el cuadrado azul con la P, se abrirá una ventana la cual vamos a escribir por ejemplo cátodo común de 7 segmentos.



**Pick Devices**

Keywords: 7SEG-COM

Match Whole Words? ☐

Show only parts with models? ☐

Category: (All Categories)

Optoelectronics

Results (6):

Device	Library	Description
7SEG-COM-AN-BLUE	DISPLAY	Blue, 7-Segment Common Anode
7SEG-COM-AN-GRN	DISPLAY	Green, 7-Segment Common Anode
7SEG-COM-ANODE	DISPLAY	Red, 7-Segment Common Anode
7SEG-COM-CAT-BLUE	DISPLAY	Blue, 7-Segment Common Cathode
<b>7SEG-COM-CAT-GRN</b>	<b>DISPLAY</b>	<b>Green, 7-Segment common Cathode</b>
7SEG-COM-CATHODE	DISPLAY	Red, 7-Segment common Cathode

7SEG-COM-CAT-GRN Preview:

Schematic Model [7SEGCOMK]

Para el capacitor, se hace lo mismo y se muestra así:

**cap**

Match Whole Words? ☐

Show only parts with models? ☐

Category: (All Categories)

Analog ICs

Capacitors

Diodes

Microprocessor ICs

Modelling Primitives

Operational Amplifiers

Switches & Relays

Results (1):

Device	Library	Description
<b>CAP</b>	<b>DEVICE</b>	<b>Generic non-electrolytic capacitor</b>

Analogue Primitive [CAPACITOR]

Para el crystal:

Device Library Description

<b>CRYSTAL</b>	<b>DEVICE</b>	<b>Quartz crystal</b>
DS3232	MAXIM	Extremely Accurate I2C RTC with Integrated Crystal and SRAM.
DS3234	MAXIM	Extremely Accurate SPI Bus RTC with Integrated Crystal and SRAM.

Schematic Model [CRYSTAL.MDF]

Para el PIC16F84A:

Device Library Description

<b>PIC16F84A</b>	<b>PICMICRO</b>	<b>PIC16 Microcontroller (1024B code, 68B data, 64B EPROM, Ports A-B, 1xTimers)</b>
------------------	-----------------	---

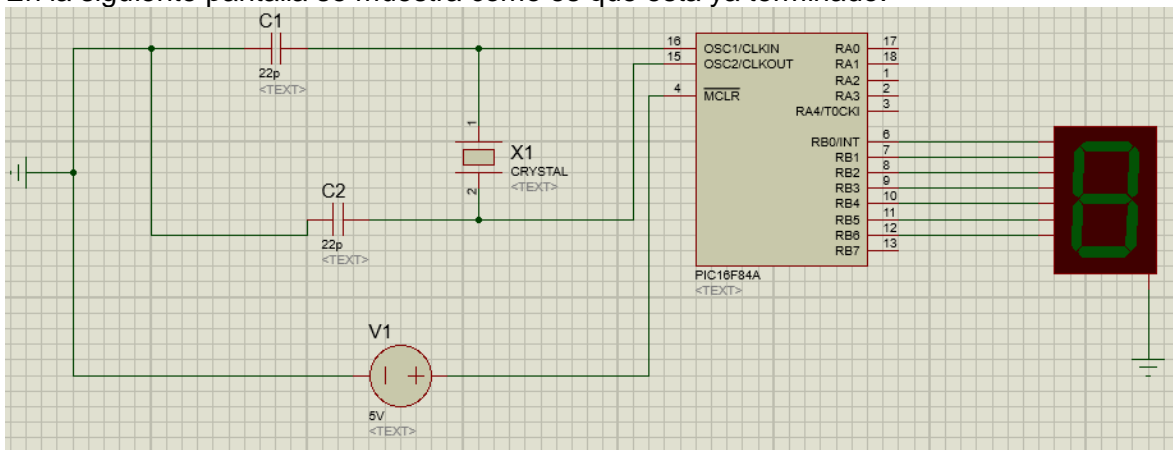
VSM DLL Model [PIC16]

Para el VSOURCE que es la fuente de poder:



Device	Library	Description	Analogue Primitive [VSOURCE]
CCCS2	ASIMMDLS	Linear Current Controlled Current Source (Vsource Control Current)	
CCR2	ASIMMDLS	Linear Current Controlled Resistor (Vsource Control Current)	
CCVS2	ASIMMDLS	Linear Current Controlled Voltage Source (Vsource Control Current)	
CSWITCH2	ASIMMDLS	Current Controlled Switch (Vsource Control Current)	
VSOURCE	ASIMMDLS	DC Voltage Source	

En la siguiente pantalla se muestra como es que esta ya terminado:



Después se utilizó el software MPLAB ID, para crear y compilar en ensamblador:



En esta parte se muestra todo el código que se utilizó para que el circuito funcione:



```

;ZONA DE DATOS*****
__CONFIG _CP_OFF & _WDT_OFF & _PWRTE_ON & _XT_OSC
LIST P=PIC16F84A
INCLUDE <P16F84A.INC>
;FIN DE LA ZONA DE DATOS*****

;ZONA DE VARIABLES*****
CBLOCK 0X0C                ;SE INICIALIZA LA MEMORIA EN C
    NUMERO                ;VARIABLE QUE LLEVARA EL CONTADOR DE 0-9 Y A-F
    CONTADOR              ;LLEVA EL TIEMPO EN CICLOS DE RELOJ
ENDC                      ;FINALIZA C
    ORG 0                 ;INICIO DEL CICLO O BUCLE EN 0
    GOTC START            ;CICLO O BUCLE
    ORG 5                 ;FIN EN 5
;FIN ZONA DE VARIABLES *****

;CONFIGURACION*****
START    BSF    STATUS,5    ;BANCO 1 ACTIVA EL BIT B EN F
        CLRF    TRISB       ;INDICA QUE PORTB SERA LA SALIDA
        MOVLW   0X1F        ;MUEVE LA PARTE BAJA DEL REGISTRO. RA0 RA4 SERAN LAS ENTRADAS
        MOVWF   TRISA       ;MUEVE EL CONTENIDO DE F A TRISA
        MOVLW   B'11000111' ;ASIGNA 256 AL TIMER
        MOVWF   OPTION_REG  ;MUEVE AL REGISTRO F AL VALOR DEL TIMER
        BCF     STATUS,5    ;CARGA EL CONTENIDO DE LA POSICION 5 AL BANCO 0
        CLRW    ;DEJA A W EN 0
        CLRF    NUMERO      ;LIMPIA LA VARIABLE NUMERO
;FIN CONFIGURACION*****
;INICIO*****
MAIN     MOVF    NUMERO,W    ;TOMA LO QUE CONTIENE LA VARIABLE NUMERO Y LO PASA A 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 EN 1
        MOVF    NUMERO,W    ;SE CARGA EL CONTENIDO DE W EN F
        XORLW   0X10        ;SE COMPARA SI ES QUE LLEGA AL REGISTRO 10
        BTFS    STATUS,Z    ;VERIFICA Y VALIDA SI HA LLEGADO
        GOTO    MAIN        ;REALIZA UN BUCLE A MAIN
        CLRW    ;SE REINICIA EL CICLO AL LLEGAR A 10
        CLRF    NUMERO      ;SE LIMPIA LA VARIABLE NUMERO
        GOTO    MAIN        ;INDICA UN BUCLE
;FIN INICIO*****

;RETARDO EN UN SEGUNDO*****
PAUSE_1000 MOVLW   0X02      ;SE LE ASIGNA 1000 AL CONTADOR
        MOVWF   CONTADOR    ;NUEVE LA VARIABLE CONTADOR A F
DELAY     BCF    INTCON,TOIF ;LIBERA EL BIT DE DESBORDAMIENTO EN EL TMRO
        MOVLW   09          ;SE CARGA EL 217
        MOVWF   TMR0        ;A TMR0
DELAY2     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
;FIN RETARDO EN UN SEGUNDO*****

```



```

;TABLA*****
TABLA  ADDWF  PCL,F      ;SE INICIALIZA LA FUNCION CON EL CONTENIDO DE F
      RETLW  B'00111111' ;SE LE ASIGNA AL CATODO COMUN EL VALOR DE 0
      RETLW  B'00000110' ;SE LE ASIGNA AL CATODO COMUN EL VALOR DE 1
      RETLW  B'01011011' ;SE LE ASIGNA AL CATODO COMUN EL VALOR DE 2
      RETLW  B'01001111' ;SE LE ASIGNA AL CATODO COMUN EL VALOR DE 3
      RETLW  B'01100110' ;SE LE ASIGNA AL CATODO COMUN EL VALOR DE 4
      RETLW  B'01101101' ;SE LE ASIGNA AL CATODO COMUN EL VALOR DE 5
      RETLW  B'01111101' ;SE LE ASIGNA AL CATODO COMUN EL VALOR DE 6
      RETLW  B'01000111' ;SE LE ASIGNA AL CATODO COMUN EL VALOR DE 7
      RETLW  B'01111111' ;SE LE ASIGNA AL CATODO COMUN EL VALOR DE 8
      RETLW  B'01100111' ;SE LE ASIGNA AL CATODO COMUN EL VALOR DE 9

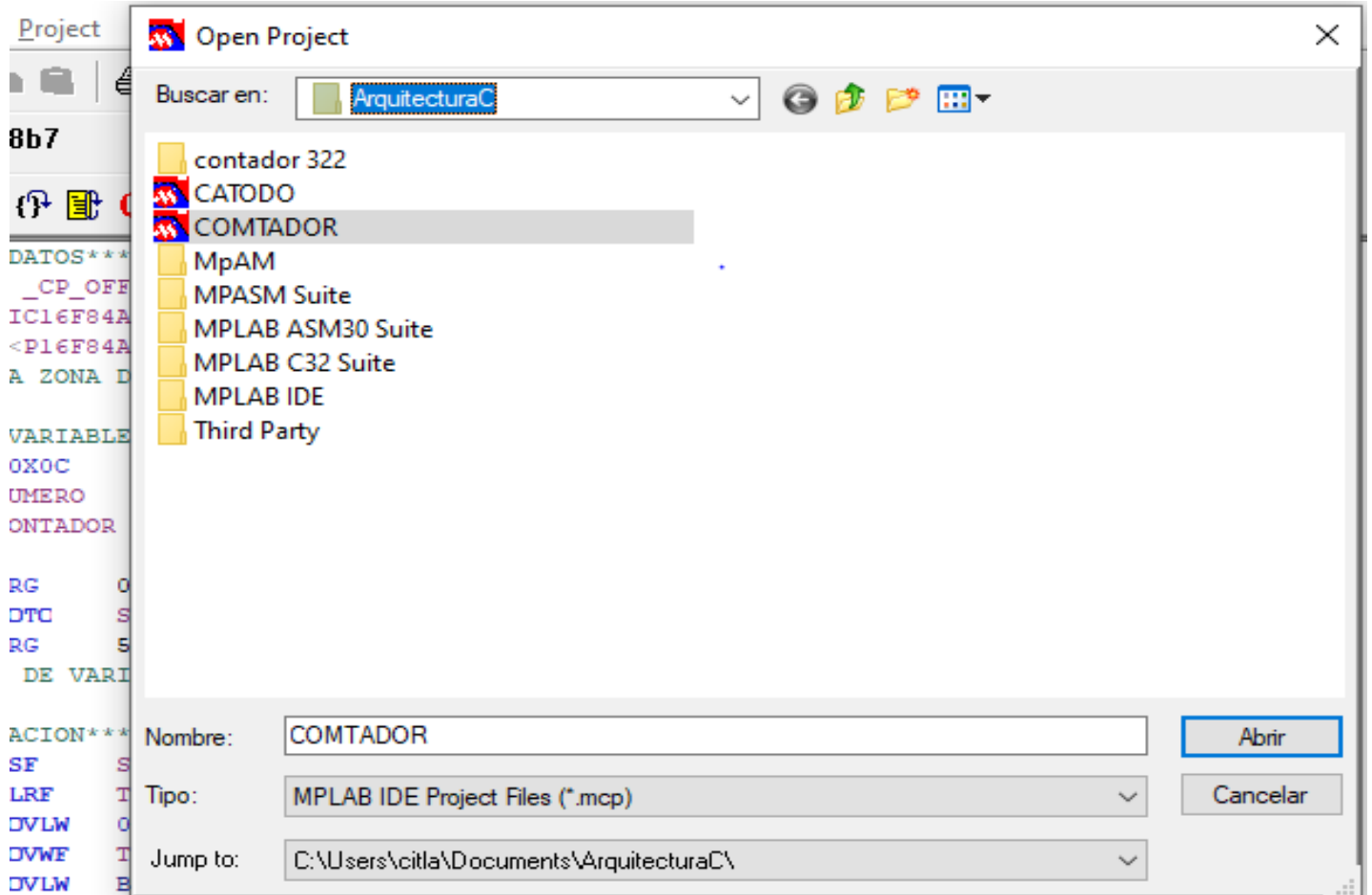
      RETLW  B'01110111' ;SE LE ASIGNA AL CATQDO COMUN EL VALOR DE A
      RETLW  B'01111100' ;SE LE ASIGNA AL CATODO COMUN EL VALOR DE b
      RETLW  B'00111001' ;SE LE ASIGNA AL CATODO COMUN EL VALOR DE C
      RETLW  B'01011110' ;SE LE ASIGNA AL CATODO COMUN EL VALOR DE d
      RETLW  B'01111001' ;SE LE ASIGNA AL CATODO COMUN EL VALOR DE E
      RETLW  B'01110001' ;SE LE ASIGNA AL CATODO COMUN EL VALOR DE F

      END

;CHECAR *****
;FIN TABLA*****

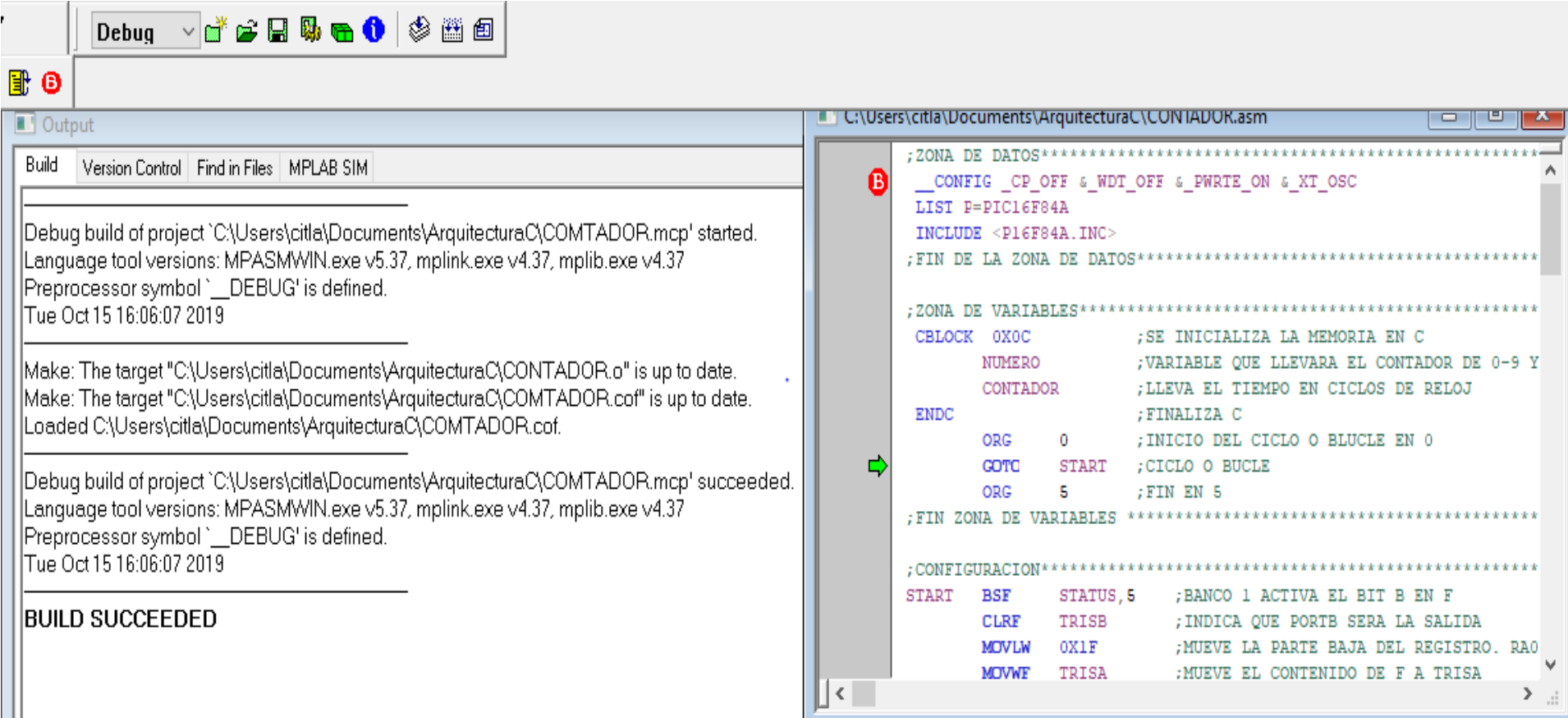
```

Una vez hecho el código, se compilo y se crea un .hex el cual nos sirve para poder correrlo en el Proteus. Pero para poder abrirlo, tenemos que dar click en Project, en Open, se abra la ventana donde seleccionamos el contador, en este caso se llama Comtador, y selecciona Abrir.





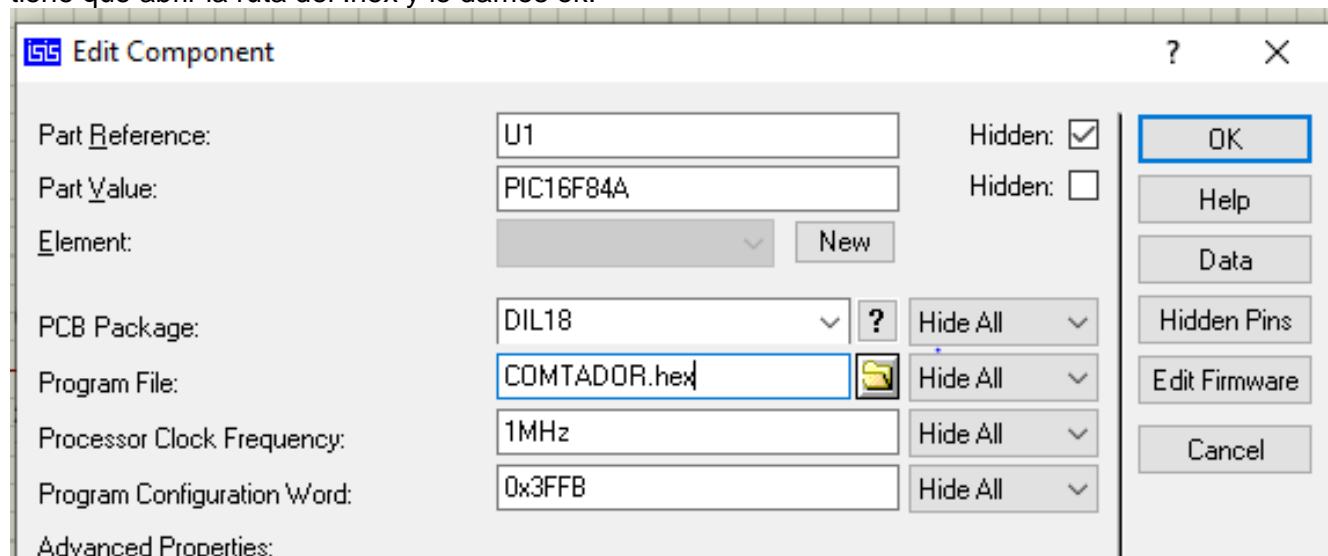
Posteriormente, en la pestaña de Make le damos clic, y mostrará si el código es correcto o si tiene fallas, si ya no tiene fallas entonces ya está listo.



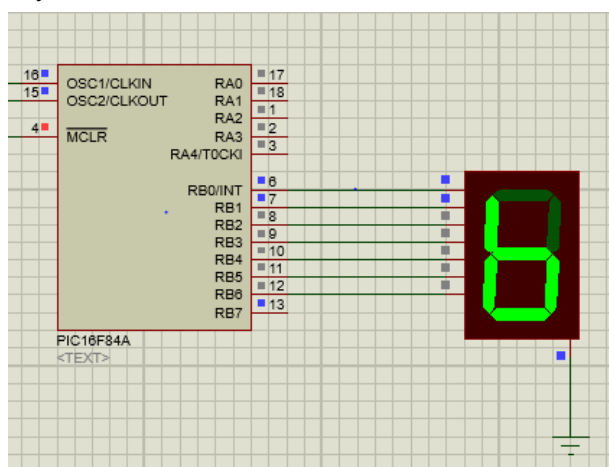
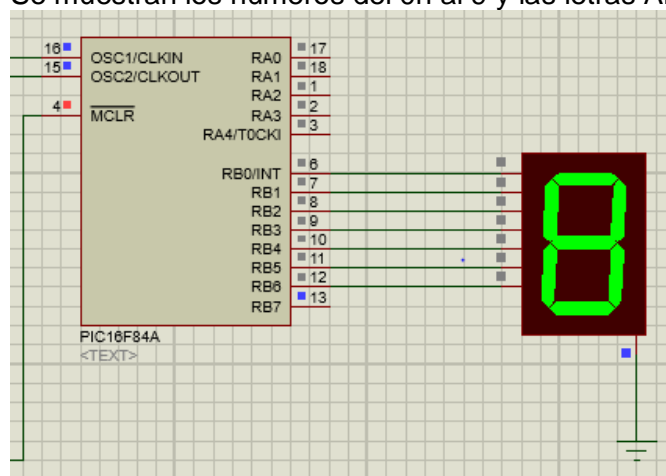
The screenshot displays the MPLAB IDE interface. The top toolbar includes a 'Debug' dropdown menu and various icons for file operations and simulation. The 'Output' window on the left shows the build process for the project 'C:\Users\citla\Documents\Arquitectura\COMTADOR.mcp'. It reports that the debug build started successfully, listing the language tool versions (MPASMWIN.exe v5.37, mplink.exe v4.37, mplib.exe v4.37) and the preprocessor symbol '\_\_\_DEBUG' defined. The build process for the target 'C:\Users\citla\Documents\Arquitectura\CONTADOR.o' and 'C:\Users\citla\Documents\Arquitectura\COMTADOR.cof' is shown as successful. The 'BUILD SUCCEEDED' message is prominently displayed at the bottom of the output window.

The source code editor on the right shows the assembly file 'C:\Users\citla\Documents\Arquitectura\CONTADOR.asm'. The code is organized into sections: 'ZONA DE DATOS', 'ZONA DE VARIABLES', and 'CONFIGURACION'. The 'ZONA DE DATOS' section includes configuration settings for the PIC16F84A, such as '\_\_\_CONFIG \_\_CP\_OFF & \_\_WDI\_OFF & \_\_PWRTE\_ON & \_\_XT\_OSC'. The 'ZONA DE VARIABLES' section defines variables like 'CBLOCK 0x0C' and 'NUMERO', and includes comments for memory initialization and variable usage. The 'CONFIGURACION' section contains instructions for setting up the hardware, such as 'START BSF STATUS,5' and 'CLRF TRISB'. A green arrow points to the 'ENDC' instruction in the 'ZONA DE VARIABLES' section.

Después en el Proteus, en el circuito se le da click en el PIC16F84A, se abre una pestaña el la cual se tiene que abrir la ruta del .hex y le damos ok.

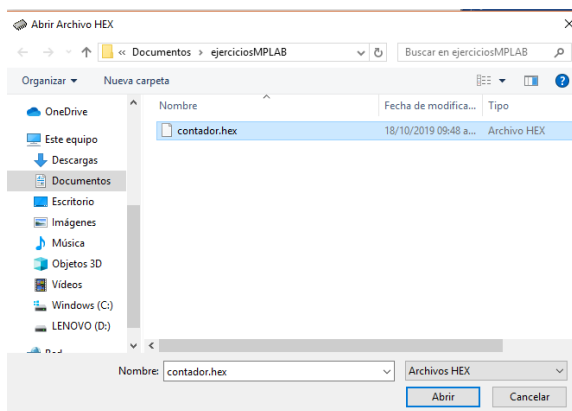
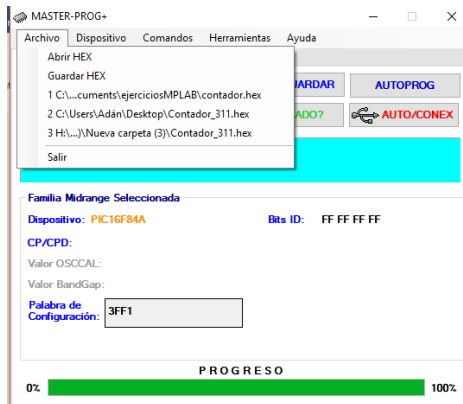


Para poder correr el circuito le damos click en play  
Se muestran los números del 0n al 9 y las letras AbC y d



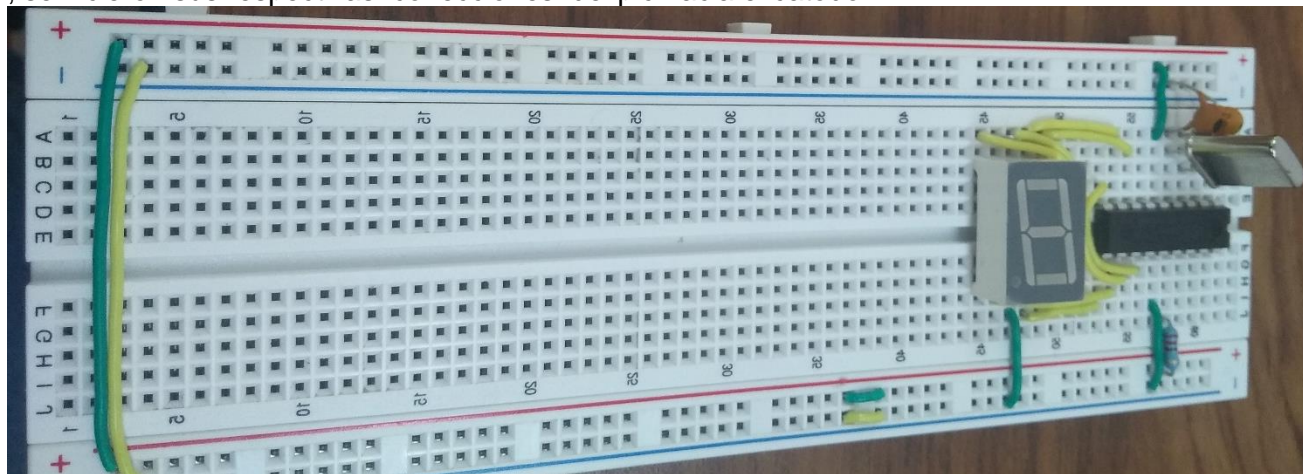


Posteriormente se programo el PIC16F84A, con un programador de PIC:

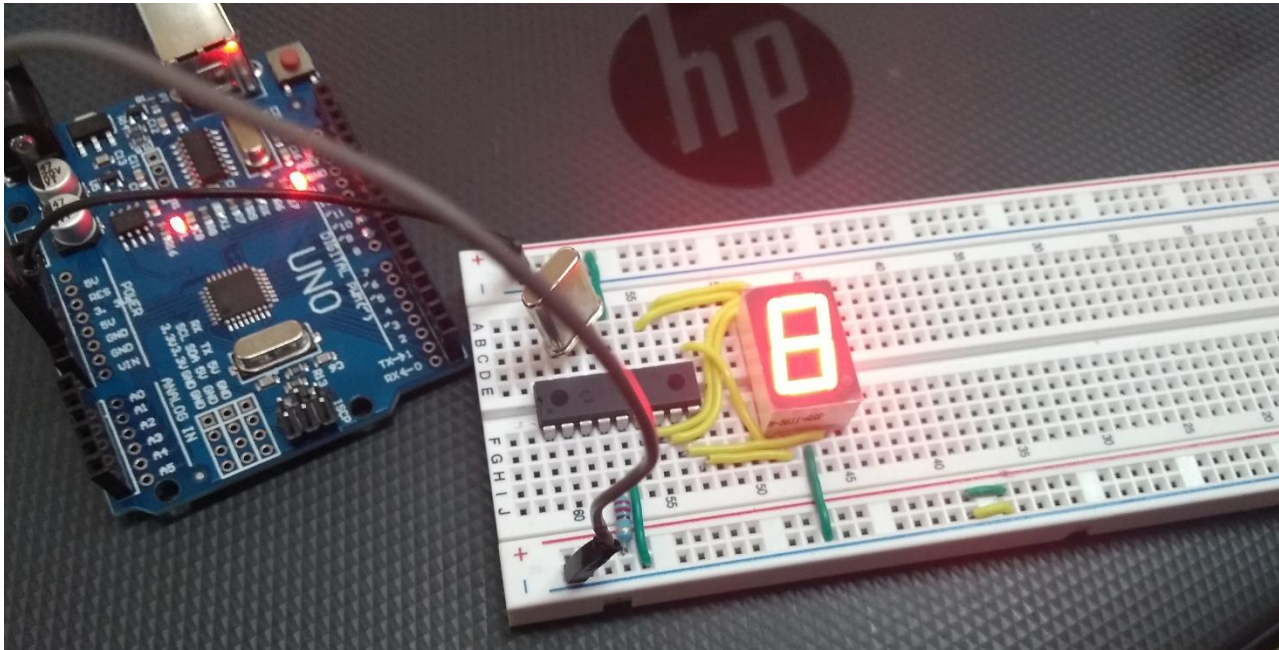




Para finalizar lo que se hizo fue pasar el circuito o simulacion creado en proteus a una tarjeta protoboard, se hicieron sus respectivas conexiones del pic hacia el catodo.:



Funcionamiento físico de simulación.



## V. Conclusiones:

En conclusion se aprendio lo basico del lenguaje ensamblador asi como a utilizar softwe de simulacion de circuitos ai mismo a crear fisicamente el modelo simulado a uno fisico .