



Nombre de la práctica	Contador 1-9 y A-F		No.	2
Asignatura:	Arquitectura de Computadoras	Carrera:	Ing. Sistemas Computacionales	Duración de la práctica (Hrs)

## I. Competencia(s) específica(s):

Crear un contador programado en ensamblador

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

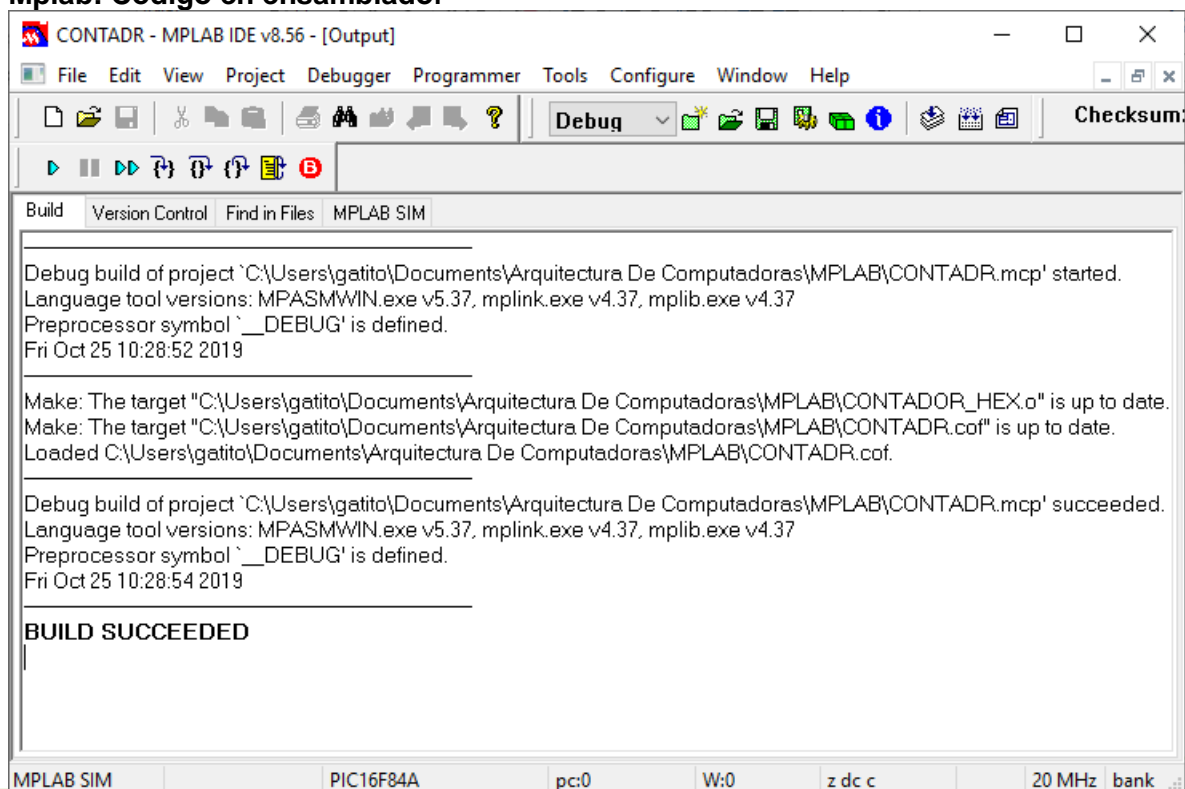
Salón de Clases

## III. Material empleado:

1 Proto  
1 cristal oscilador  
1 resistencia de 330  $\Omega$   
1 pic 16F84A  
1 cátodo común de 7 segmentos  
2 cerámicos de 22 picofaradio  
Cable para proto  
Software proteus  
Software Mplab  
Programador de PIC'S

## IV. Desarrollo de la práctica:

### ▪ Mplab: Código en ensamblador





```

CONTADR - MPLAB IDE v8.56 - [C:\Users\gatito\Documents\Arquitectura De Computadoras\MPLAB\CONTADOR_HEX.asm*]
File Edit View Project Debugger Programmer Tools Configure Window Help
Checksum: 0xd8b7

;ZONA DE DATOS*****
__CONFIG _CP_OFF & _WDT_OFF & _PWRTE_ON & _XT_OSC ;OBLIGATORIO COLOCARLO PARA NO CERRAR LA ESCRITURA
LIST F=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
    BTFSS STATUS,2 ;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 TMR0
    MOVLW 09 ;SE CARGA EL 217
    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
;FIN RETARDO EN UN SEGUNDO*****

```



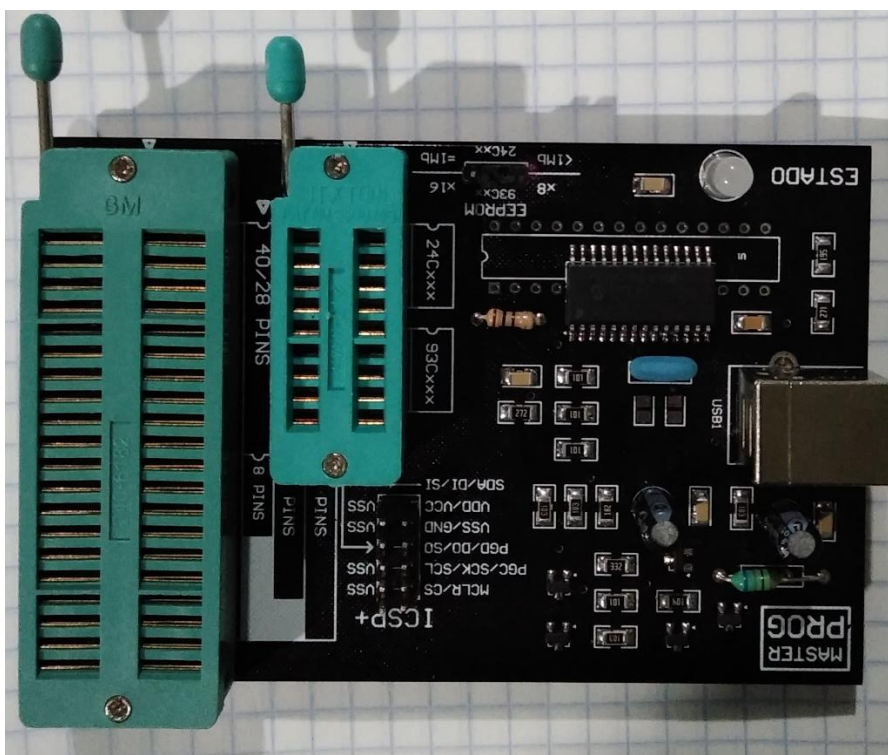
```
;TABLA*****
TABLA  ADDWF  PCL,F      ;SE INICIALIZA LA FUNCION CON EL CONTENIDO DE F
RETLW  B'00111111' ;SE LE ASIGNA AL CATODO COMUN EL VALOR DE 0
RETLW  B'00000110' ;SE LE ASIGNA AL CATODO COMUN EL VALOR DE 1
RETLW  B'01011011' ;SE LE ASIGNA AL CATODO COMUN EL VALOR DE 2
RETLW  B'01001111' ;SE 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 CATODO 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
;FIN TABLA*****

MPLAB SIM      PIC16F84A      pc:0      W:0      z dc c      20 MHz      bank 0      Ln 1, Col 1
```

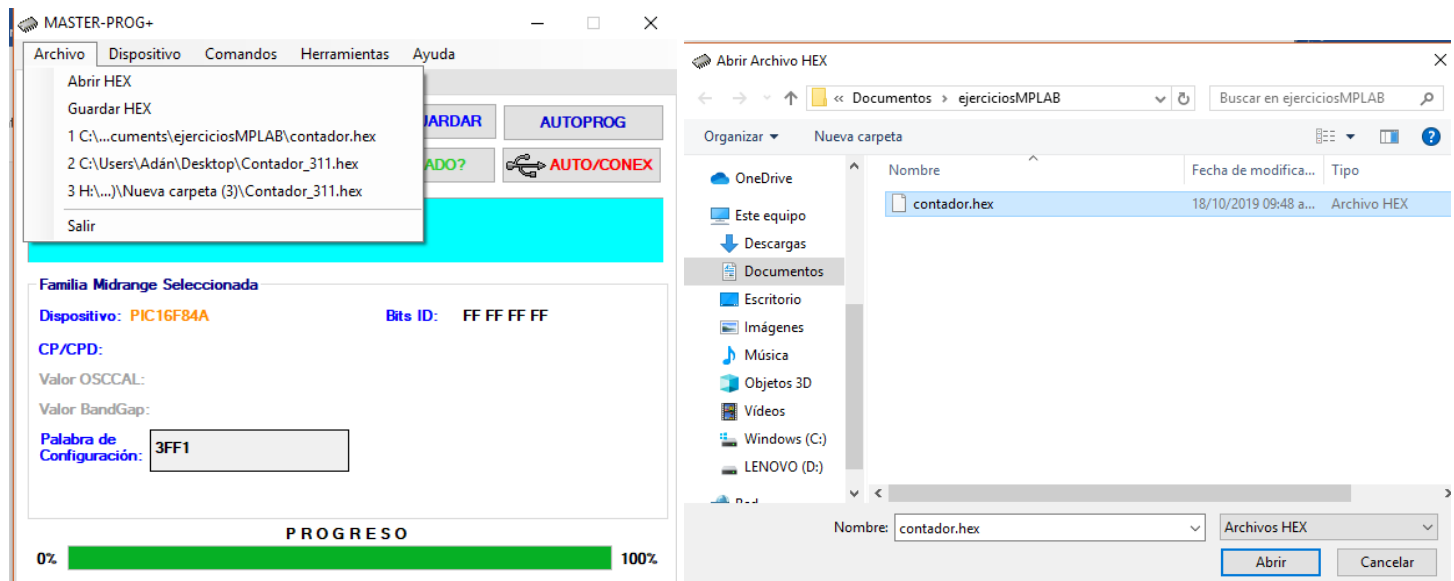
- **Programación de PIC16F84A**  
Conectar programador de pic



Verificar que reconoce el pic16F84A



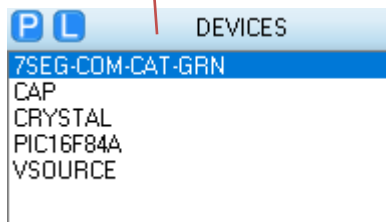
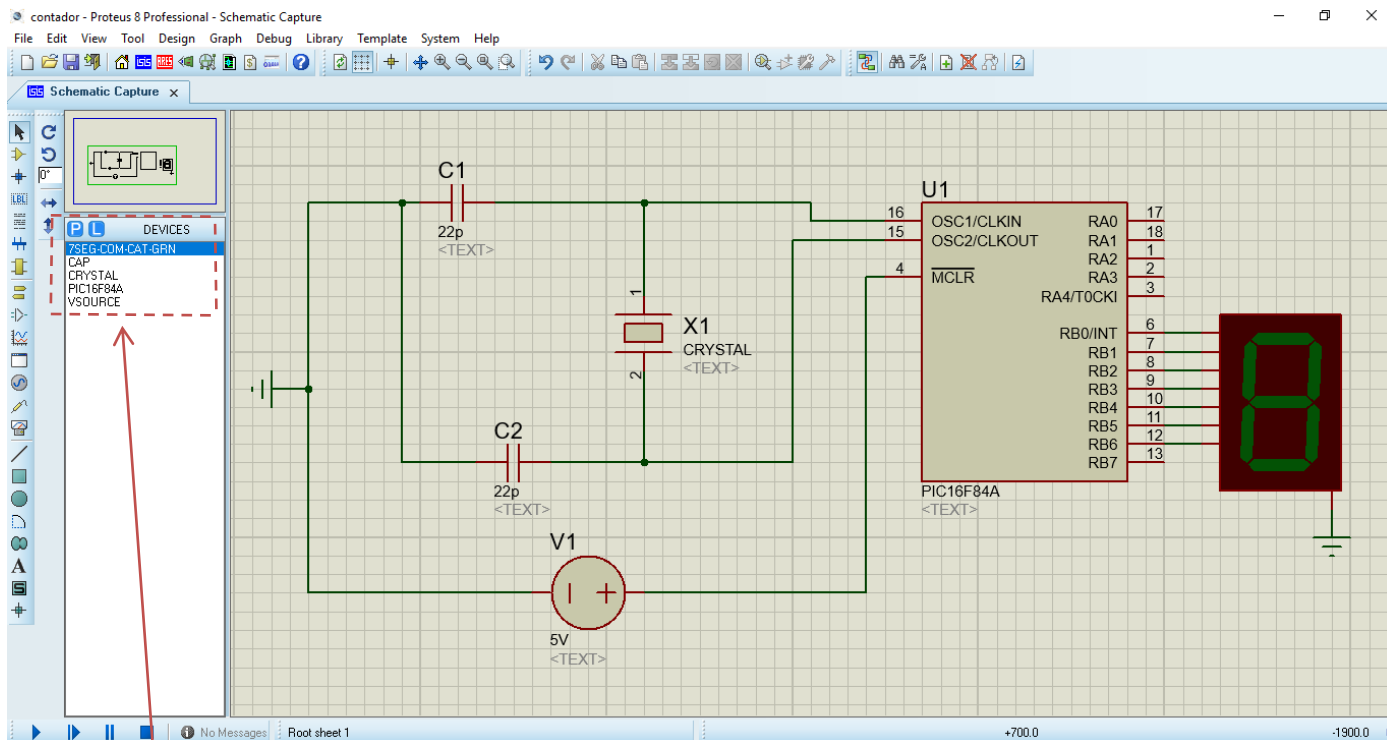
Dar clic en Archivo y abrir HEX, seleccionar el archivo .hex que genera Mplab



Dar clic en escribir



## Proteus: Material



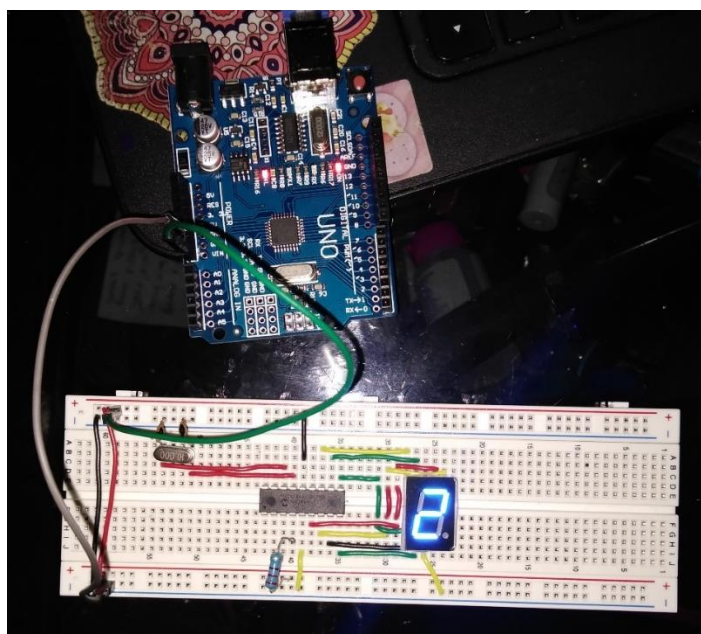
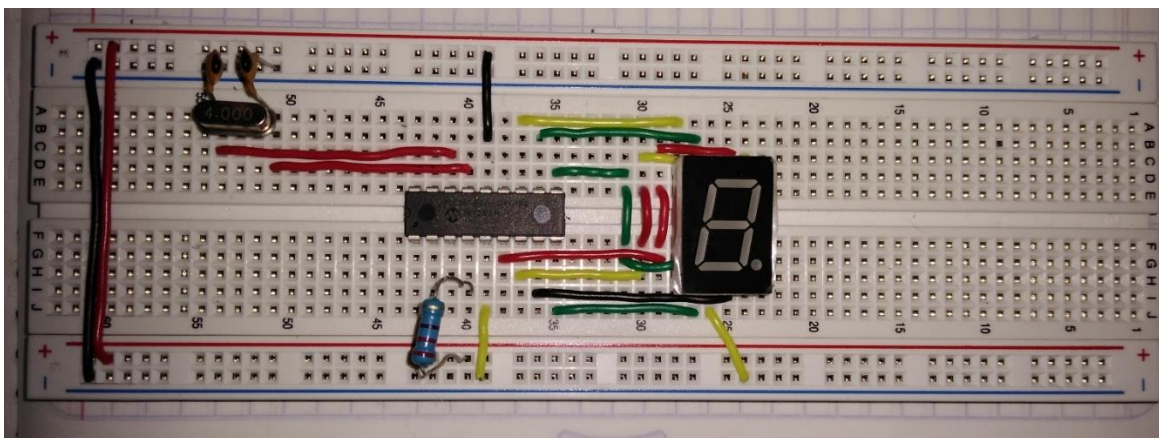
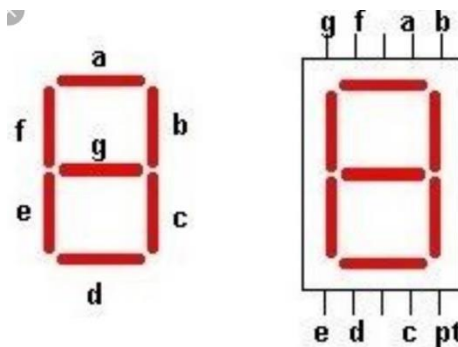
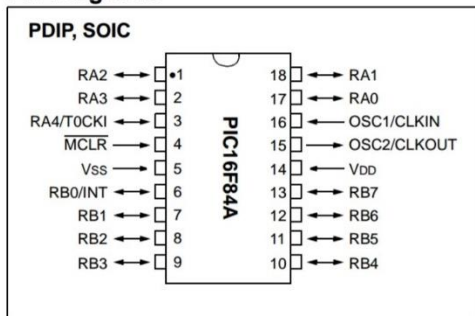
Para hacer la simulación del circuito usamos el software Proteus, usamos los siguientes componentes: 7SEG-COM-CAT-GRN (cátodo común de 7 segmentos color verde), CAP (capacitores/ cerámicos), CRYSTAL (cristal oscilador), PIC16F84A, VSOURCE (corriente).



Colocamos el Pic en la parte derecha se encuentra RB0/INT hasta RB7 donde cada patita del pic se conecta con el cátodo de 7 segmentos, el cátodo va conectado a tierra. El cerámico 1 va conectado a OSC1 al igual que el cerámico 2 este conecta a OSC2 dentro de la conexión de los dos cerámicos se conecta un cristal. De la salida MCLR conecta la corriente y conectado a tierra.

## Proto física

### Pin Diagrams





Para armar el circuito, nos apoyamos con el data sheet del pic16F84A y el del cátodo de 7 segmentos, además de la simulación que realizamos en proteux.

Conectar el cátodo con el pic, es decir en de la patita 6 -12 corresponden a la conexión de cátodo

## V. Conclusiones:

Los softwares que utilizamos fueron de gran ayuda, ya que proteux permite simular el circuito, tiene elementos suficientes para diseñarla, el código es basado en ensamblador el cual da una señal de prendido o apagado para cada salida.

Llevarlo a la práctica primero se tuvo que programar el pic, con el código de ensamblador especialmente con el archivo .hex que genera mplab.