

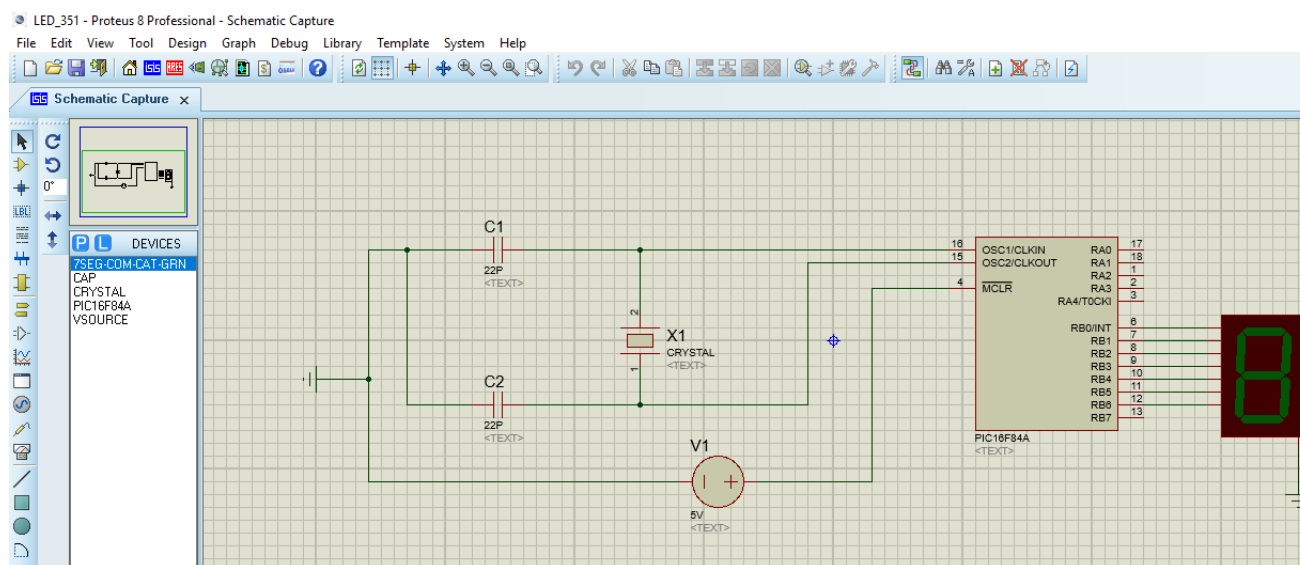


Nombre de la práctica	Reporte de contador			No.	
Asignatura:	Arquitectura de Computadoras	de Carrera:	Ingeniería en Sistemas Computacionales	Duración de la práctica (Hrs)	

Carlos Daniel Pérez Soto

Desarrollo de la práctica:

En esta practica lo primero que hicimos fue crear un esquema para ver cómo quedaría nuestro proyecto, el cual lo hicimos en el IDE Proteus.



Después de haber creado el diagrama hicimos el código para que funcionara el cátodo, el cual lo hicimos con la ayuda del IDE MPLAB el cual es programado en lenguaje ensamblador.

```

;ZONA DE DATOS*****
__CONFIG __CP_OFF & __WDT_OFF & __PWRTE_ON & __XT_OSC
LIST P=PIC16F84A
INCLUDE <P16F84A.INC>
;FIN DE ZONA DE DATOS*****
;ZONA DE VARIABLES*****
CBLOCK 0X0C ;SE INICIALIZA LA MEMORIA 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 EN 0 BUCLE EN 0
GOTO 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

```



```

        CLRWF TRISB                ;INDICA QUE PORTB SERA LA SALIDA
        MOVLW 0X1F                ;MUEVE LA PARTE BAJA DEL REGISTRO.
RA0 RA4 SERAN LAS ENTRADAS
        MOVWF TRISA                ;MUEVA EL CONTENIDO DE F A TRISA
        MOVLW B'11000111'        ;ASIGNA 256 AL TIMER
        MOVWF OPTION_REG          ;MUEVE AL REGISTRO F EL VALOR DEL TIMER
        BCF STATUS,5              ;CARGA EL CONTENIDO DE LA POSICION 5
AL BANCO 0
        CLRW                      ;DEJA A W EN 0
        CLRWF NUMERO              ;LIMPIA LA VARIABLE NUMERO
;FIN CONFIGURACION*****

;INICIO*****
MAIN MOVF NUMERO,W                ;TOMA LO QUE CONTIENE LA VARIABLE NUMERO
        CALL TABLA                ;LLAMA A LA FUNCION TABLA
        MOVWF PORTB                ;MUESTRA EL VALOR QUE TOMO DE LA
TABLA
        CALL PAUSE_1000            ;LLAMA A LA FUNCION PAUSE_1000
        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 REGITRO 10
        BTFSSSTATUS,Z              ;VERIFICA Y VALIDA SI HA LLEGADO
        GOTO MAIN                  ;REALIZA UN BUCLE A MAIN
        CLRW                      ;SE REINICIA EL CICLO AL LLEGAR A 10
        CLRWF NUMERO              ;SE LIMPIA LA VARIABLE NUMERO
        GOTO MAIN                  ;INDICA UN BUCLE
;FIN INICIO*****

;RETARDO EN SEGUNDO*****
PAUSE_1000 MOVLW 0X02              ;SE LE ASIGNA 1000 AL CONTADOR
        MOVWF CONTADOR            ;MUEVE LA VARIABLE CONTADOR A F
DELAY BCF INTCON,T0IF              ;LIBERA EL BIT DE DESBORDAMIENTO EN
EL TMR0
        MOVLW 09                  ;SE CARGA EL 217 (1)
        MOVWF TMR0                ;A TMR0 (1)
DELAY2 BTFSSINTCON,T0IF            ;SE LIBERA EL BIT DE DESBORDAMIENTO DEL
TMR0
        GOTO DELAY2                ;BUCLE DEL DELAY2
        DECFSZ CONTADOR,F          ;DECREMENTA EN 1 EL CONTADOR
        RETURN                    ;REGRESA
;FIN RETARDO EN UN SEGUNDO*****
;TABLA*****
TABLA ADDWF PCL,F                ;SE INICIALIZA LA FUNCION TABLA CON EL CONTENIDO
DE F
        RETLW B'00111111'          ;SE LE ASIGNA AL CATODO CUMUN 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

```



```

6      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

7      RETLW      B'00000111' ;SE LE ASIGNA AL CATODO COMUN EL VALOR DE

8      RETLW      B'01111111' ;SE LE ASIGNA AL CATODO COMUN EL VALOR DE

9      RETLW      B'01100111' ;SE LE ASIGNA AL CATODO COMUN EL VALOR DE
;HEXADECIMAL

      RETLW      B'01111011' ;SE LE ASIGNA AL CATODO COMUN LA LATRA A
      RETLW      B'01111100' ;SE LE ASIGNA AL CATODO COMUN LA LATRA b
      RETLW      B'00111001' ;SE LE ASIGNA AL CATODO COMUN LA LATRA C
      RETLW      B'01011110' ;SE LE ASIGNA AL CATODO COMUN LA LATRA d
      RETLW      B'01011110' ;SE LE ASIGNA AL CATODO COMUN LA LATRA E
      RETLW      B'01110001' ;SE LE ASIGNA AL CATODO COMUN LA LATRA F

      END
;FIN TABLA*****

```

```

;ZONA DE DATOS*****
__CONFIG __CP_OFF & __WDT_OFF & __PWRTE_ON & __XT_OSC
LIST      P=PIC16F84A
INCLUDE <P16F84A.INC>
;FIN DE ZONA DE DATOS*****
;ZONA DE VARIABLES*****
CBLOCK 0X0C ;SE INICIALIZA LA MEMORIA 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 EN 0 BUCLE EN 0
    GOTO START ;CICLO O BUCLE
    ORG 5 ;FIN EN 5

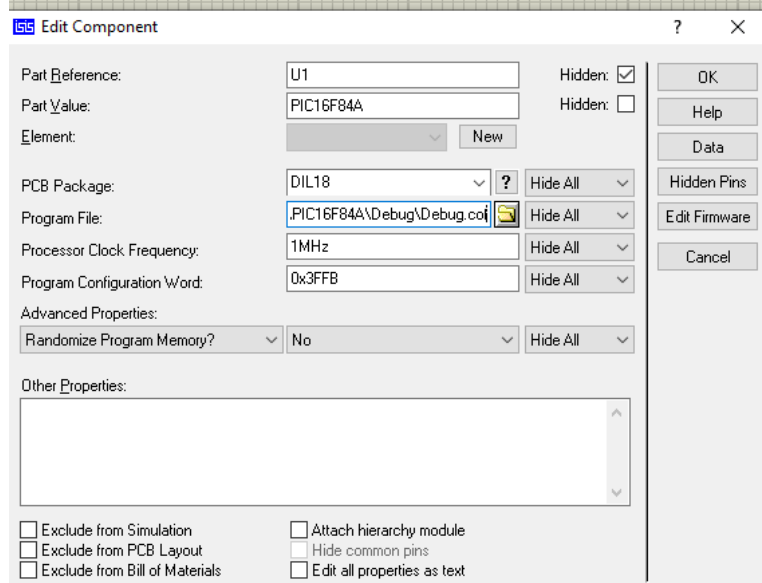
;FIN ZONA DE VARIABLES*****
;CONFIGURACION*****
START BSF STATUS,5 ;BANCO 1 ACTIVA EL BIT B EN F
      CLRF TRISB ;INDICA QUE PORTB SERA LA SALIDA
      MOVLW 0X1F ;MUEVE LA PARTE BAJA DEL REGISTRO. RA0 RA4 SERAN LAS ENTRADAS
      MOVWF TRISA ;MUEVE EL CONTENIDO DE F A TRISA
      MOVLW B'11000111' ;ASIGNA 256 AL TIMER
      MOVWF OPTION_REG ;MUEVE AL REGISTRO F EL 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
     CALL TABLA ;LLAMA A LA FUNCION TABLA
     MOVWF PORTB ;MUESTRA EL VALOR QUE TOMO DE LA TABLA
     CALL PAUSE_1000 ;LLAMA A LA FUNCION PAUSE_1000
     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 REGITRO 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
;FTN FIN*****

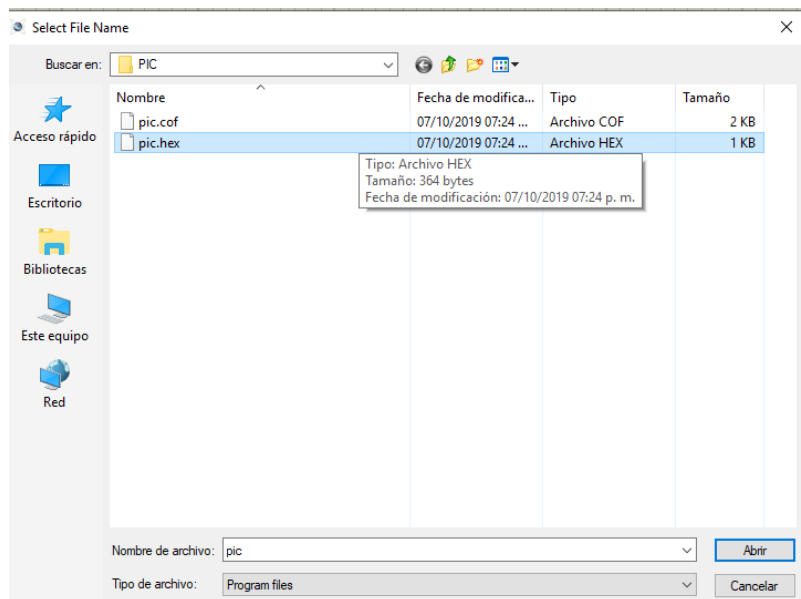
```

Después de haber creado el código lo guardamos con la extensión .asm, después nos dirigimos nuevamente a Proteus para cargar el código en el PIC16F84A

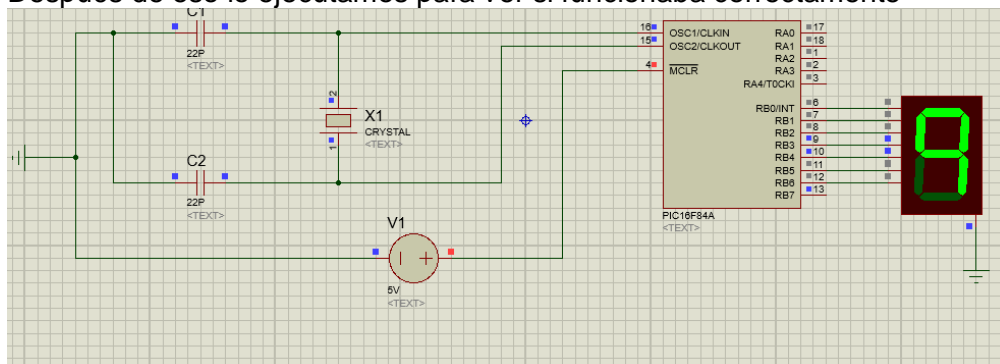
Le dimos doble clic al PIC16F84A y dimos clic a la carpeta en la opción program file.



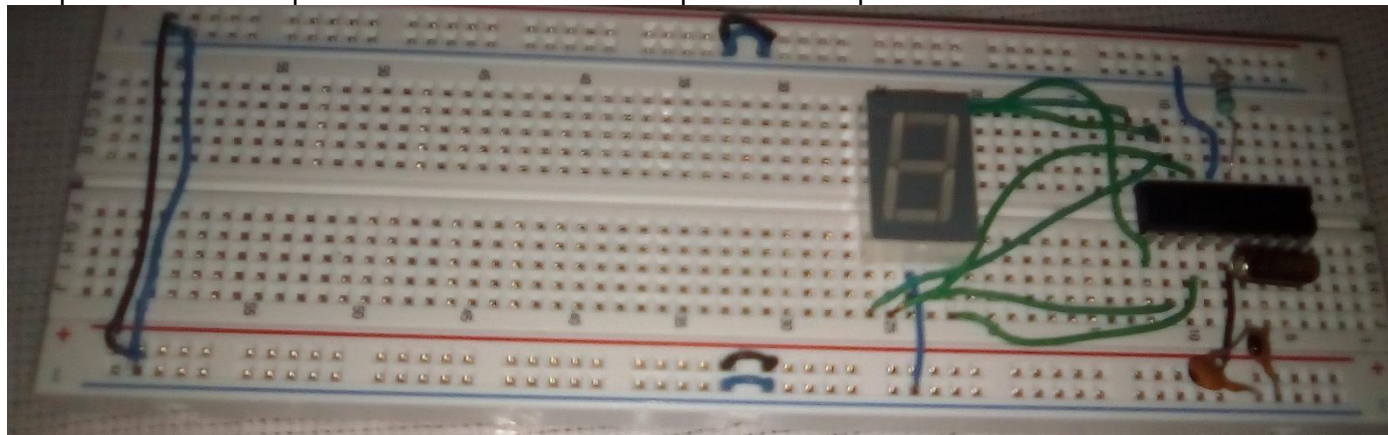
Después nos fuimos a la dirección en la cual guardamos el código le dimos clic en abrir y posteriormente dimos clic en OK



Después de eso lo ejecutamos para ver si funcionaba correctamente

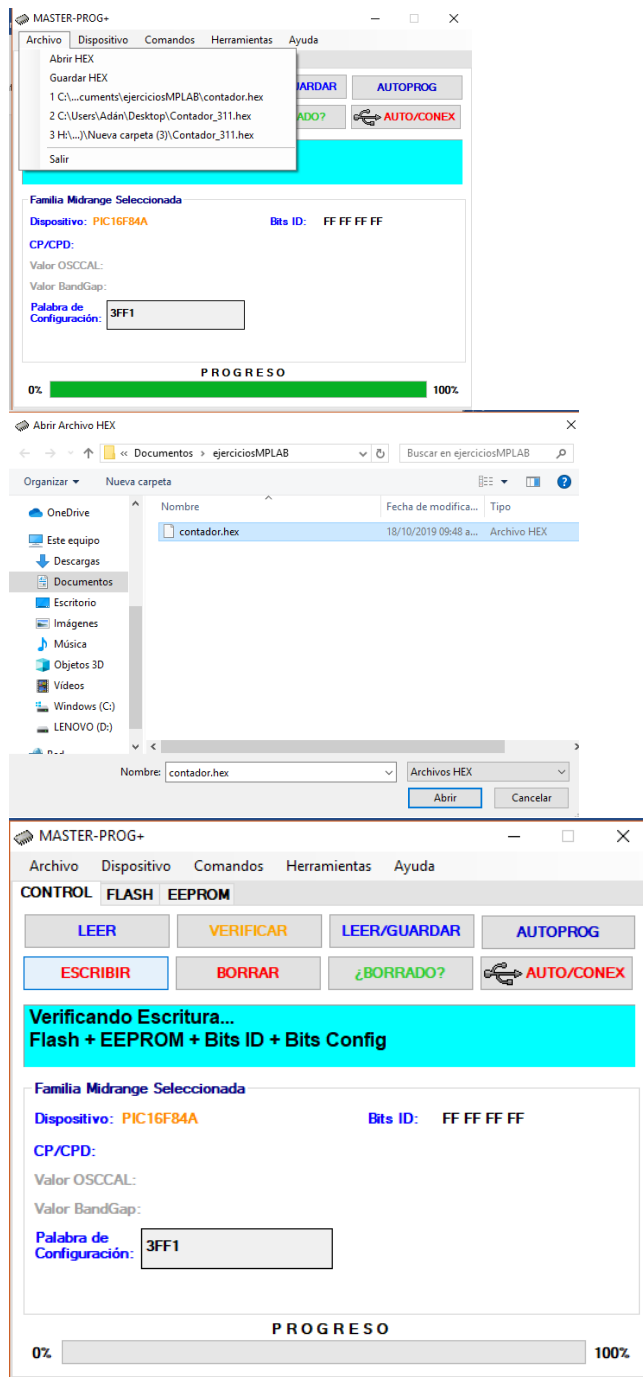


Después de verificar que funcionara correctamente pasamos a representarlo de manera física.

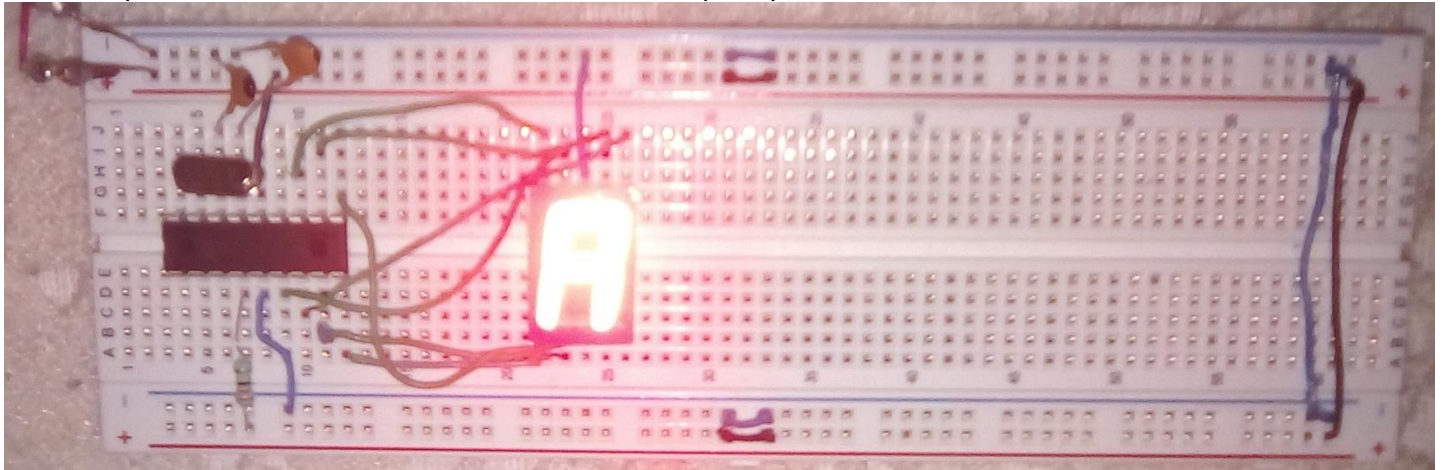


Después de haber hecho físicamente el circuito tuvimos que programar el PIC16F84A





Después de haber programado correctamente el PIC16F84A lo conectamos correctamente en la tabla PROTO, posteriormente lo conectamos a la corriente para probar el funcionamiento.



Conclusiones:

En esta practica pude aprender un poco sobre Datasheet en este caso lo hicimos con un ánodo, también aprendí a como conectar dichos