

Nombre de la práctica	Cátodo común de 7 segmentos			No.	2
Asignatura	Arquitectura de computadoras	de	Carrera:	Isic	Duración de la práctica (Hrs)
					2hrs

IVETH MARTINEZ BECERRA

III. Material empleado:

Pic16f84a

Cátodo común de 7 segmentos

Proto

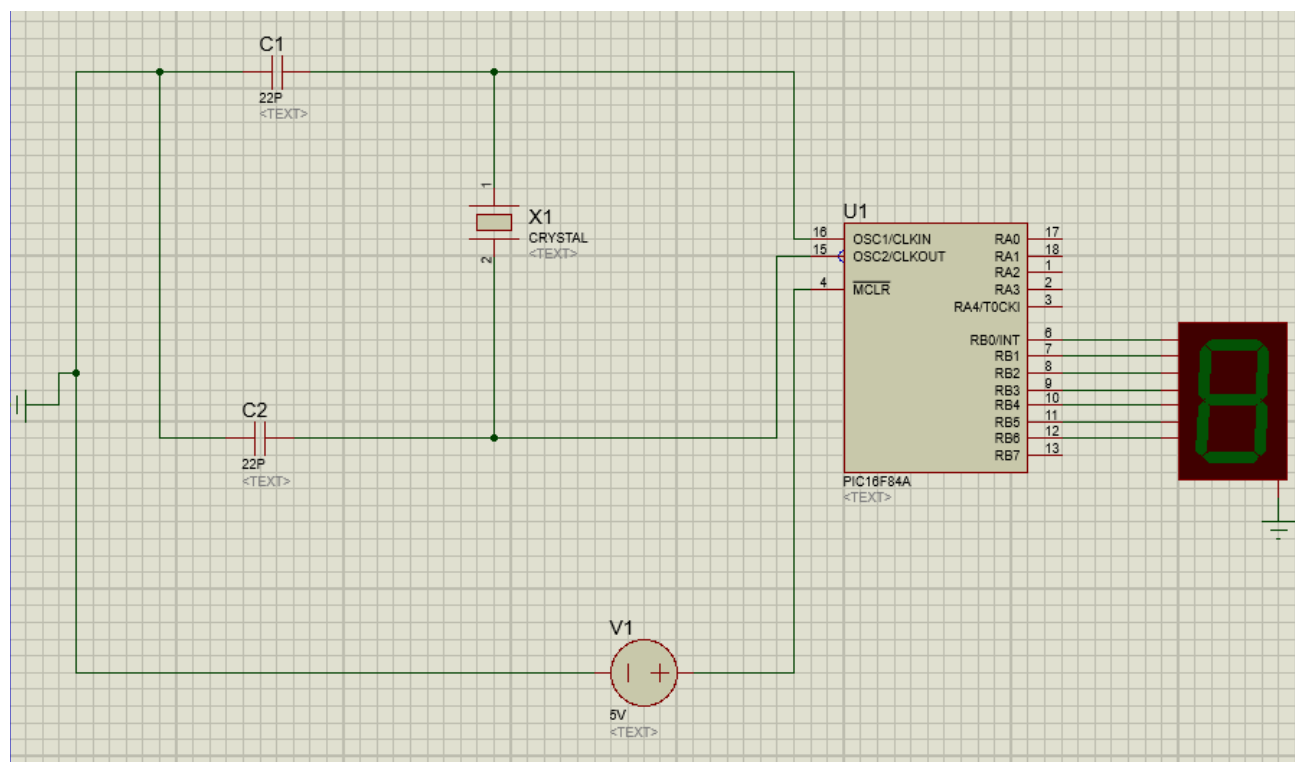
Cable para proto

Cristal oscilador

Cerámicos de 22 picofaradios

IV. Desarrollo de la práctica:

como primer punto fue realizar una simulacion de como iba a funcionar esto lo realizamos es proteus donde creamos un proyecto al cual llamamos contador y seleccionamos los componentes los cuales íbamos a ocupar en este caso fue un cátodo común de 7 segmentos, un cristal, dos cerámicos de 22 pico faradios, y un pic16f84a y quedo de la siguiente manera



pasando a la parte de la codificación nos apoyamos de mlab para programar el pic16f84 en lenguaje ensamblador , creamos un nuevo proyecto al cual llamamos contador, y nos quedo de siguiente manera



```

;zona de datos***
;wdt permite escribir otra vez
__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
    NUMERO                ; VARIABLE QUE LLEVARA EL CONTADOR DE 0-9 Y DE A-F
    CONTADOR              ; LLEVA EL TIEMPO EN CICLOS DE RELOG
ENDC                ; FINALIZA C
    ORG      0                ; INICIO DEL CICLO O BUCLE EN 0
    GOTC     START            ; CICLO O BUCLE
    ORG      5                ; FIN EN 5
;FIN DE VARIABLES*****
;configuracion*****
START BSF     STATUS,5        ;banco 1 activa el bit B en F
    CLRF     TRISE            ;indica que portb sera la salida
    MOVLW    0X1F            ;mueve la parte baja del registro, ra0 ra4 seran las entradas
    MOVWF    TRISA           ;Muve 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 d ela posicion 5 al banco 0
    CLRW                    ;deja a w en 0
    CLRF     NUMERO          ;limpia la variable numero
;fin configuracion*****

;INICIO*****
MAIN    MOVE     NUMERO,W      ; TOMA LO QUE CONTIENE LA VARIABLE NUMERO
    CALL      TABLA           ; LLAMA A LA FUNCION TABLA
    MOVWF     PORTE           ; MUESTRA EL VALOR DQUE TOMO DE LA TABLA
    CALL      PAUSE_10000     ; LLMA A LA FUNCION PAUSE_1000
    INCF      NUMERO,F        ;REALIZA UN INCREMENTO EN LA VARIABLE EN 1
    MOVE      NUMERO,W       ;SE CARGA EL CONTENIDO EN W EN F
    XORLW     0X10            ;SE CONPARA SI LLEGA AL REGISTRO 10
    BTFSS     STATUS,Z        ;VERIFICA SI HA LLEGADO
    GOTC      MAIN           ;QUE LO REPITA, REALIZA UN BUCLE A MAIN
    CLRW                    ;SE REINICIA EL CODIGO AL REGRESAR A 10
    CLRF      NUMERO          ;SE LIMPIA LA VARIABLE NUMERO
    GOTC      MAIN           ;INDICA UN BUCLE

;FTN INICIO*****

```



```

GOTC    MAIN          ;QUE LO REPITA, REALIZA UN BUCLE A MAIN
CLRWF   ;SE REINICIA EL CODIGO AL REGRESAR A 10
CLRF    NUMERO        ;SE LIMPIA LA VARIABLE NUMERO
GOTC    MAIN          ;INDICA UN BUCLE
;FIN INICIO*****

;RETARDO EN UN SEGUNDO*****
PAUSE_10000 MOVLW    0X02      ;SE LE AGIGNA 1000 AL CONTADOR
MOVWF    CONTADOR      ;MUEVE 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 DESBORDAMINETO DEL TMR0
GOTC     DELAY2        ;BUCLE DEL DELAY2
DECFSZ   CONTADOR,F    ; DECREMENTA EN 1 EL CONTADOR
GOTC     DELAY         ;BUCLE EN DELAY
RETURN   ; REGRESA

;FIN RETARDO EN UN SEGUNDO*****

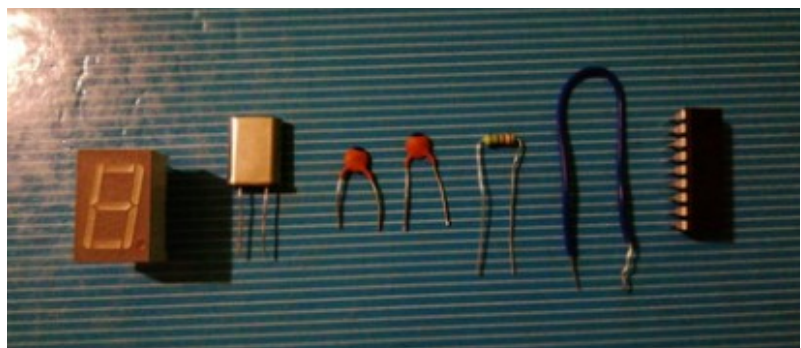
;TABLA*****
TABLA    ADDWF        PCL,F      ;SE INICIALIZA LA FUNCIO TABLA CON EL CONTENIDO DE F
RETLW    B'00111111'          ;SE LE ASIGNA AL CATODO COMUN EL VALOR DE 0
RETLW    B'00000110'          ;SE LE ASIGNA AL CATODO COMUN EL VALOR DE 1
RETLW    B'01011011'          ;2
RETLW    B'01001111' 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

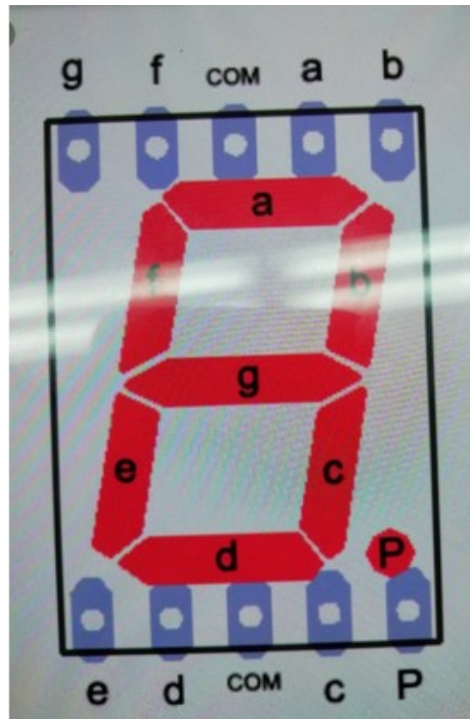
```

pasando a la parte del hardware y para ya armarlo con material tangible ocupamos lo siguiente:

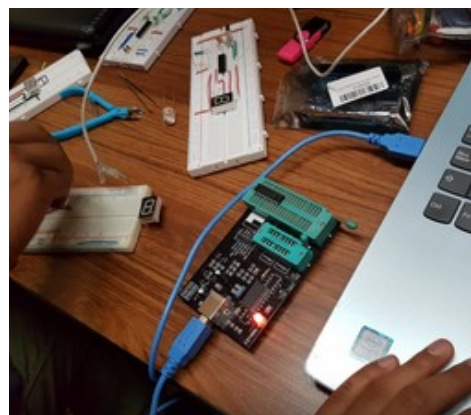
- catodo comun de 7 segmentos
- un cristal oscilador
- 2 ceramicos de 22 picofaradios
- una resistencia
- cable para proto
- pic16f84a



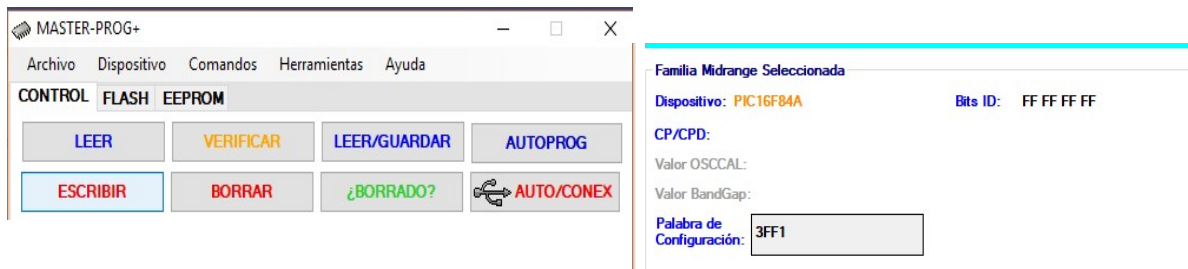
lo siguiente fue apoyarme del diagrama que ya habia realizado en proteus y de otro para observar en que posicion tenia que ir colocados los cables para que prendieran en el orden asignado en el codigo



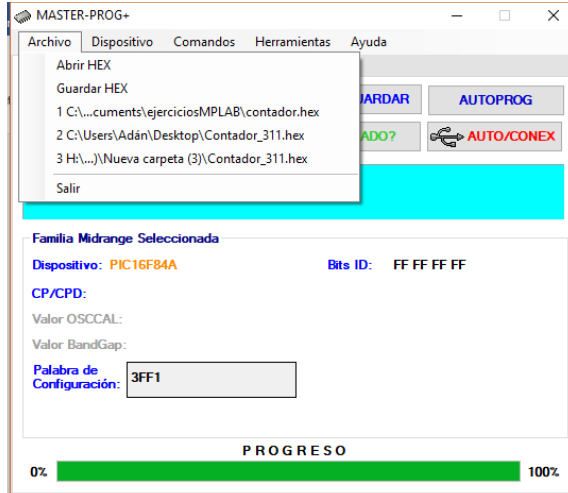
para programar el pic realizamos los siguientes pasos



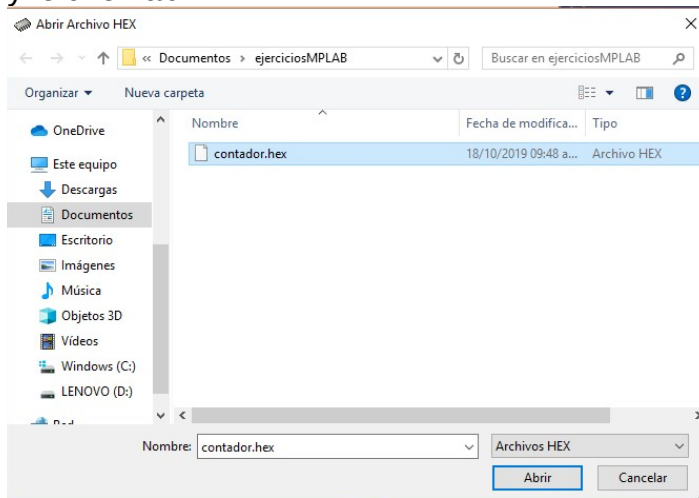
colocamos el pic y le pusimos el seguro para que fuera reconocido en el programa
posteriormente realizamos la coneccion a la maquina para pasarle al pic el codigo ya antes
hecho en mplab
0.-pulse en AUTO/CONEX para que reconociera el pic



1.- di click en abrir HEX como se muestra en la siguiente imagen



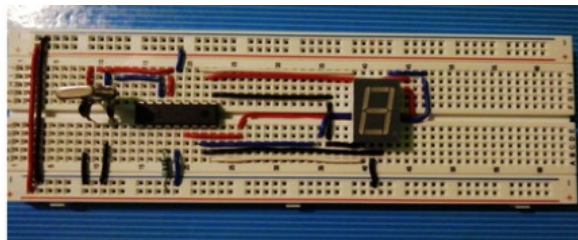
2.- seleccionamos el archivo de mplab que en mi caso se llama contador con extencion .hex y le di en abrir



3.-una vez seleccionado el archivo con extencion .hex pulse en escribir y verifique la accion



y listo el pic a quedado grabado de acuerdo a lo que ya tenia de proteus y del anterior diagrama pude realizar toda la coneccion y me quedo de la siguiente manera:



una vez programado el pic lo conecte a la corriente para verificar que funcionara y los resultados fueron satisfactorios

