

Nombre de la práctica	Investigación Discos Duros Mecánicos y de Estado Solido				No.	1	
Asignatura:	Arquitectura Computadoras	de	Carrera:	Ingeniería Sistemas Computacionales	en	Duración de la práctica (Hrs)	--

Alumno: Marco Antonio Rodriguez Garcia Grupo: 3051

I. Competencia(s) específica(s):

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

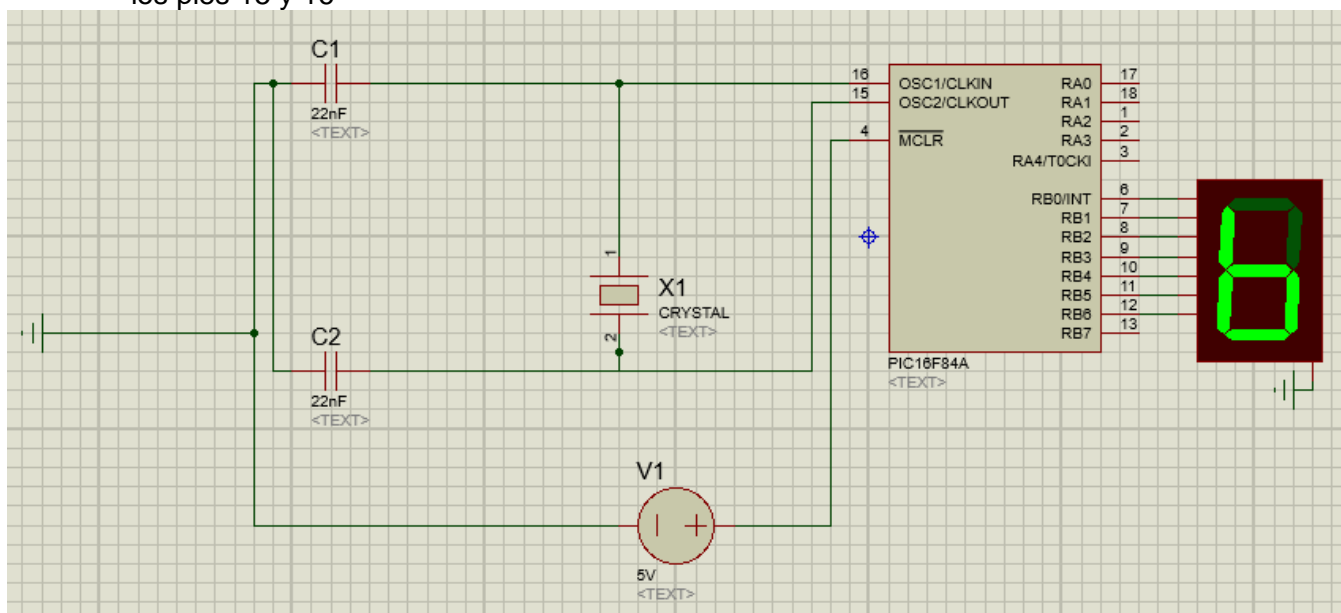
Aula/Casa

III. Material empleado:

- PIC16F84A.
- Capacitadores de 22 Microfaradios.
- Cristal Oscilador a 1 MHz.
- Fuente de energía de 5 v.
- Software Proteus (Con Licencia Legal ☺).
- Software MPLap.

IV. Desarrollo de la práctica:

- Dentro de Proteus:
 - ✓ Los elementos que se agregan son dos capacitadores de 22 nF, un cristal oscilador a 1MHz, una batería o fuente de energía de 5v, un Cátodo común de 7 segmentos verde y por supuesto el PIC16F84A
 - ✓ Lo primero es conectar los capacitadores en paralelo junto con el cristal oscilador, y estos a su vez unidos a la tierra que se conecta a la energía negativa de la batería, para la alimentación del pic según su su Data Sheet se utiliza el pie número 4 y para el cristal y capacitadores se usan los pies 15 y 16





- ✓ Al dar doble clic sobre el pic se abre menu de opciones en el cual, utilizando el icono de folder donde exportaremos el archivo .hex que será el código que llevara el PIC virtual

Edit Component

Part Reference: U1 Hidden: ☒

Part Value: PIC16F84A Hidden: ☐

Element: [Dropdown] New

PCB Package: DIL18 [Dropdown] ? Hide All [Dropdown]

Program File: Contador.hex [Folder Icon] Hide All [Dropdown]

Processor Clock Frequency: 1MHz Hide All [Dropdown]

Program Configuration Word: 0x3FFB Hide All [Dropdown]

Advanced Properties:

Randomize Program Memory? [Dropdown] No [Dropdown] Hide All [Dropdown]

Other Properties:

☐ Exclude from Simulation ☐ Attach hierarchy module

☐ Exclude from PCB Layout ☐ Hide common pins

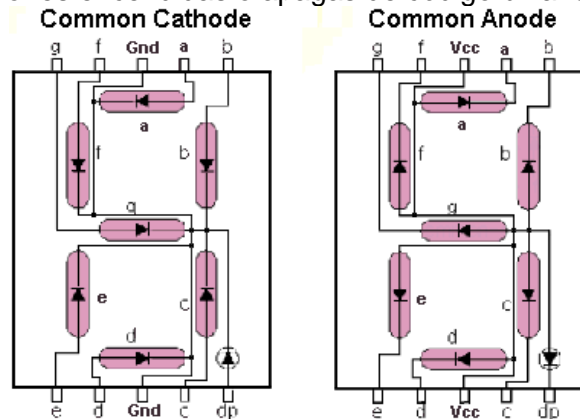
☐ Exclude from Bill of Materials ☐ Edit all properties as text

Buttons: OK, Help, Data, Hidden Pins, Edit Firmware, Cancel



• Dentro de MPLab:

- ✓ Comenzamos con el código utilizamos `__CONFIG` para poder indicar que el pic se podrá volver a programar, dentro de list ingresamos el PIC que vamos a utilizar e incluimos su biblioteca
- ✓ Dentro de la zona de variables inicializamos la memoria C y declaramos las variables número y contador
- ✓ Por otro lado, en el interior de Configuración por medio de Start BSF que activa el Bit del banco en memoria, por medio de CLRF TRISB indicamos que utilizaremos las salidas del puerto B
- ✓ En inicio y retardo hacemos el movimiento de los datos dentro de los bancos de memoria y los puertos que se están empleando.
- ✓ Por último, tenemos tabla, que son los números que nuestro cátodo mostrara, que son representados con posiciones encendidas o apagadas de código binario.



```

; ZONA DE DATOS*****
__CONFIG __CP_OFF & __WDT_OFF & __PWRTE_ON & __XT_OSC
LIST    F=PIC16F84A
INCLUDE <P16F84A.INC> ; CARGAR LAS LIBRERIAS CON EL LENGUAJE C
; FIN DE ZONA DE DATOS*****

; ZONA DE VARIABLES*****
CBLOCK 0x0C ; SE INICIALIZA LA MEMORIA C
    NUMERO ; VA A SER LA VARIABLE QUE LLEVARA EL CONTADOR DE 0-9 Y A a F
    CONTADOR ; LLEVA EL TIEMPO EN CICLOS DE RELOJ
ENDC ; FINALIZA C
    ORG 0 ; INCIO DEL CICLO O BUCLE EN 0
    GOTO START ; CICLO O BUCLE
    ORG 5 ; FIN EN 5
; FIN ZONA DE VARIABLES*****

; CONFIGURACION *****
START BSF STATUS,5 ; BANCO DE MEMORIA 1 ACTIVA EL BIT B EN F
CLRF TRISB ; INDICA QUE PORTB SERA LA SALIDA
MOVLW 0x1E ; 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 EL REGISTRO F EL VALOR DEL TIMER
BCF STATUS,5 ; CARGA EL CONTENIDO DE LA POSICION 5 AL BANCO DE DATOS 0
CLRWF ; DEJA A W EN 0 (CLEAR)
CLRF NUMERO ; LIMPIA LA BARIABLE NUMERO
; FIN CONFIGURACION*****

```



```
; INICIO*****
MAIN    MOVF     NUMERO,W      ; TOMA LO QUE CONTIENE LA VARIABLE NUMERO Y LO PASA A F
        ; W ES EL QUE TRAE EL VALOR DE NUMERO PARA PASARLO A F
        CALL    TABLA        ; LLAMA A LA FUNCION TABLA
        MOVWF   PORTB        ; MUESTRA EL VALOR QUE TOMO DE 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 LA VARIABLE W EN F
        XORLW   0X10          ; SE COMPARA SI ES QUE LLEGA AL REGISTRO 10
        BTFSS   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 QUE TENIA EL CONTADOR
        GOTO    MAIN          ; INDICA UN BUCLE PARA HACERLO OTRA VEZ
; FIN INICIO*****

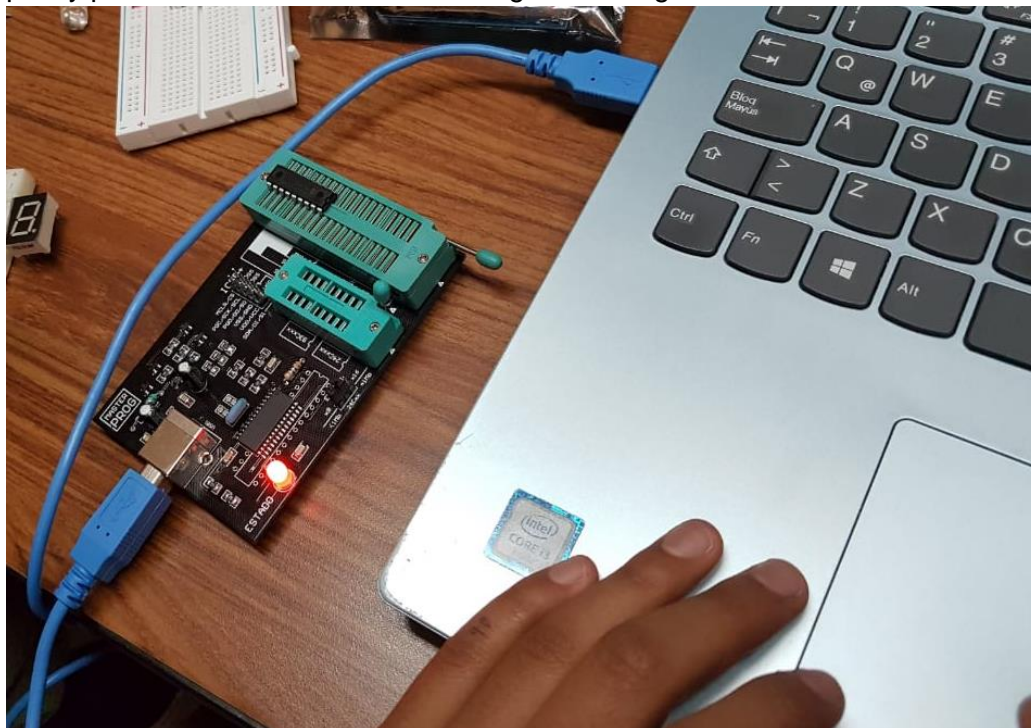
; RETARDO EN UN SEGUNDO*****
PAUSE_1000 MOV LW    0X02      ; SE LE ASIGNA 1000 MILISEGUNDOS AL CONTADOR PARA HACER UN SEGUNDO
           MOVWF   CONTADOR    ; MUEVE LA VARIABLE CONTADOR A F
DELAY      BCF     INTCON, TOIF ; LIBERA EL PIC DE DESBORDAMIENTO EN EL TMR0 (EL DELAY ES EL CATODO)
           MOV LW   09         ; SE CARGA EL 217
           MOVWF   TMR0        ; A TMR0
DELAY2     BTFSS   INTCON, TOIF ; SE LIBERA EL PIC DE DESBORDAMIENTO DEL TMR0
           GOTO    DELAY2      ; BUCLE DEL DALAY2
           DECF    CONTADOR,F  ; DECREMENTA EN UNO EL CONTADOR
           GOTO    DELAY       ; BUCLE EN DELAY
           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' ;0 SE LE ASIGNA AL CATODO COMUN EL VALOR DE 0
          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
; 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

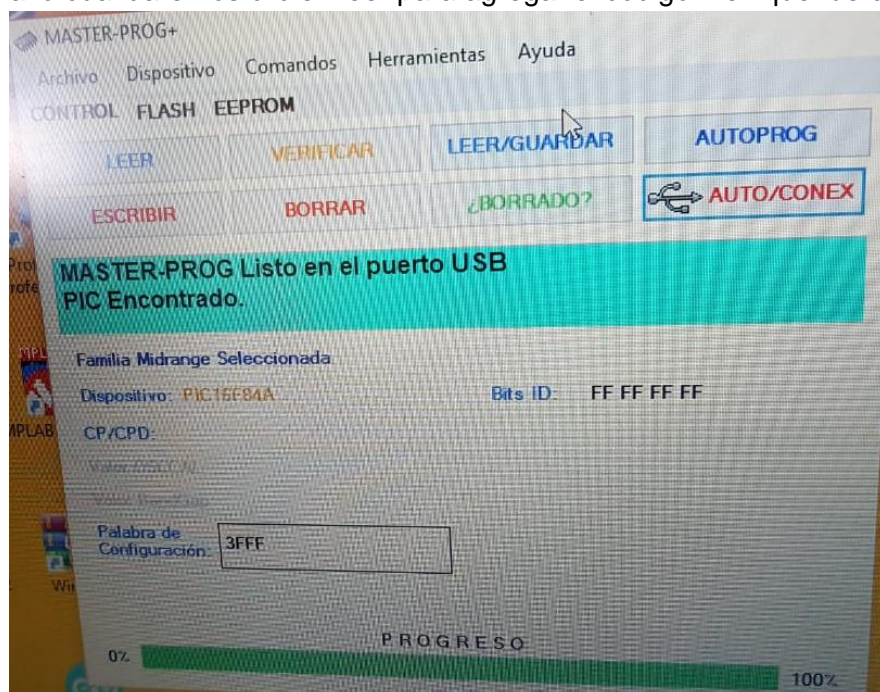
          END
; FIN TABLA*****
```


- **Dentro del Programador de PICs**

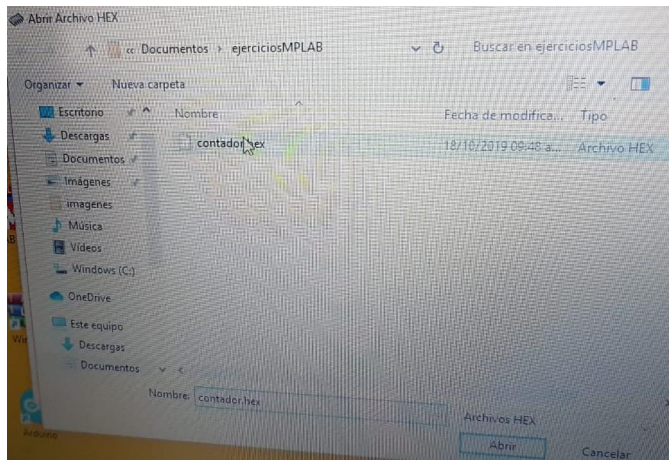
- ✓ El programador de pic consta de una placa con una entrada de energía y transferencia de datos similar al de Arduino, en esta placa podemos colocar nuestro PIC y por medio de una palanca apretar sus pies y por medio del cable de datos cargar un código dentro de la memoria flash del PIC.



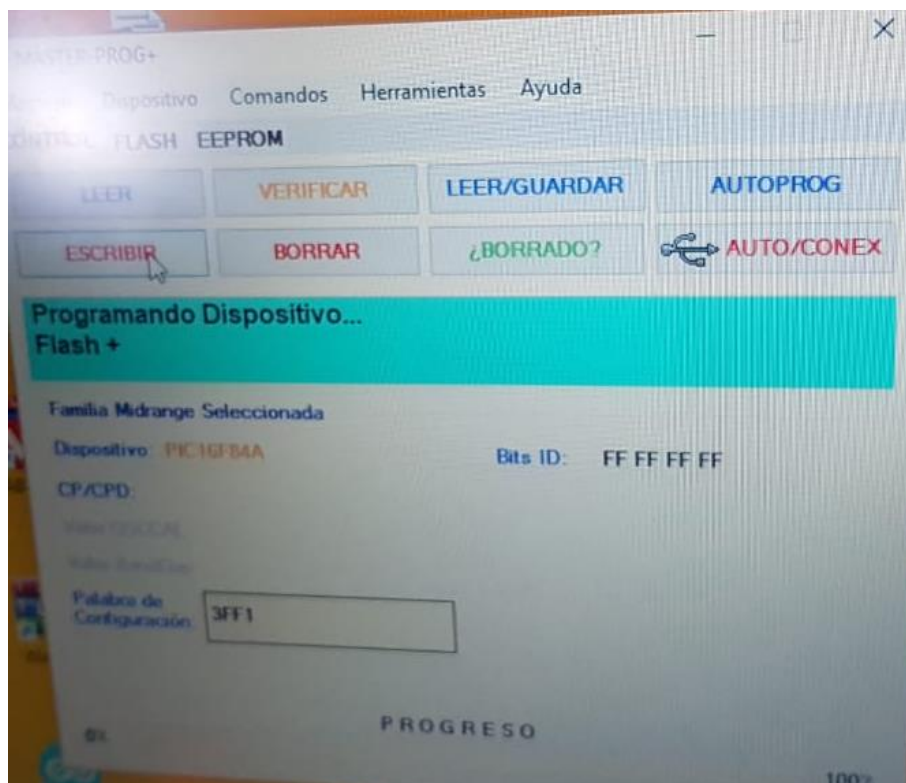
- ✓ Una vez conectado el programador de PIC abriremos el Master Prog+ que controlara al programador a lo cual daremos clic en leer para agregar el código .hex que fue compilado dentro de MPLAB



- ✓ Una vez seleccionado esto buscamos el archivo .hex y lo abriremos



- ✓ Lo siguiente, una vez conectado el PIC al programador y leído el código .hex daremos clic en escribir para comience el proceso de programar el pic

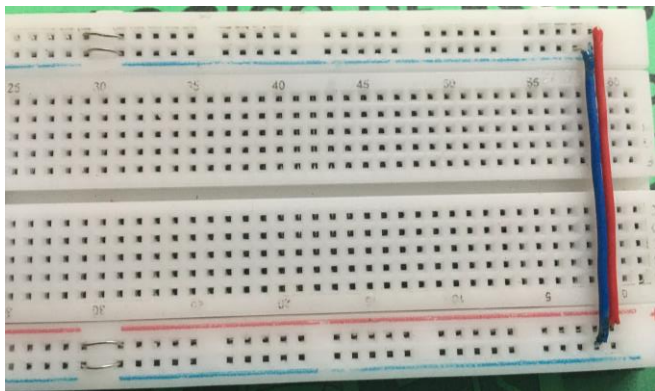


- ✓ Una vez terminado el proceso mostrara el mensaje de Escritura Correcta en la consola a lo cual procederemos a desconectar el PIC del programador y colocarlo dentro de nuestra proto

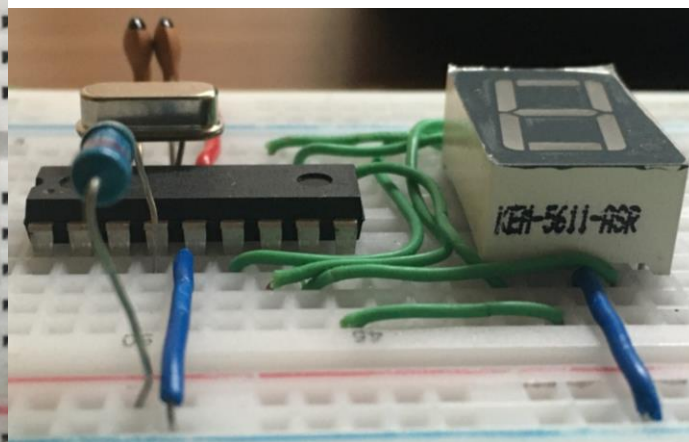
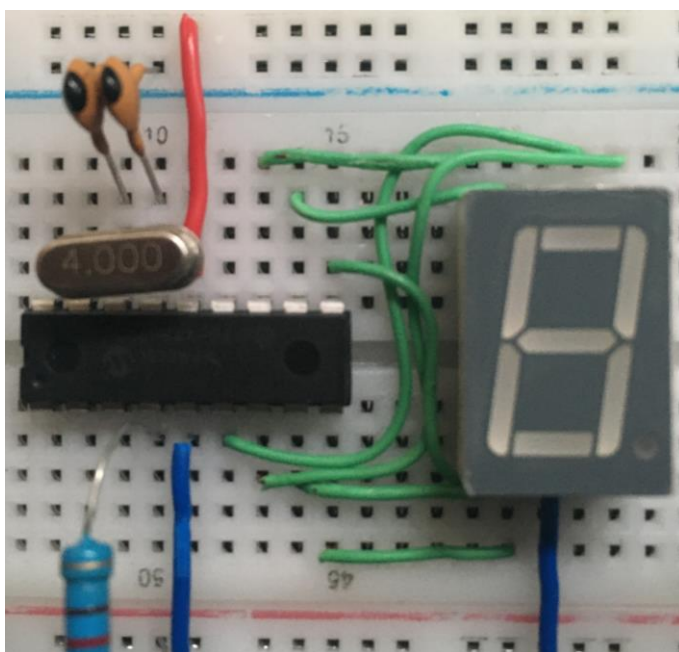


- **Dentro de la Protoboard**

- ✓ Lo primero que se tiene que hacer es realizar los puentes entre los puntos de la fuente de energía para que fluya dentro de toda la protoboard



- ✓ La forma en la que los elementos fueron conectados fue muy similar a la de proteus, en la proto conectamos el PIC en el centro de tal forma que su luna quedase a la izquierda, de entrada, conectamos la energía negativa al pie 5 y la energía positiva en el pie 14, conectamos el cristal oscilador en los pines 15 y 16 del PIC y estos a su vez conectados a la energía negativa por medio de dos capacitadores cerámicos y una resistencia de 330 Ω
- ✓ Para la conexión del cátodo tenemos una entrada de energía negativa en el pin 3, dentro de las conexiones que hicimos desde el PIC al Cátodo tenemos que, el pin 6 del pic conecta con el pin 4 superior del cátodo, de igual manera el pin 7 con el pin 5 superior, 8 con el pin 4 inferior, el 9 con el pin 2 inferior, el pin 10 con el pin 1 inferior, el pin 11 con el pin 2 superior y por ultimo el pin 12 con el pin 1 superior del PIC al CATODO.





V. Conclusiones:

Concluyendo, el uso de softwares de simulación como lo es proteus, son herramientas indispensables si se desea programar para un PIC, pues con este software podemos modelar cual es la reacción de un PIC virtual ante una serie de conexiones o los códigos que les podemos ingresar, por otro lado es una buena ayuda para poder conocer la posición en la que deben estar las conexiones, dentro del producto en físico funciona de la misma manera que el PIC virtual gracias a que este nos sirviera de prueba, por ultimo cabe mencionar que el programador de PICS al ser una placa grande muy parecida a Arduino, tiene una funcionalidad excelente, puesto que, con esta placa podemos insertar nuestros códigos en la memoria Flash que contiene el PIC16F84A