

Autor: Fernando Remiro Domínguez
Profesor de Sistemas Electrónicos del
IES Juan de la Cierva
<http://telexline.terra.es/personal/fremiro/>

EL MPLAB

El MPLAB es un software que junto con un emulador y un programador de los múltiples que existen en el mercado, forman un conjunto de herramientas de desarrollo muy completo para el trabajo y/o el diseño con los microcontroladores PIC desarrollados y fabricados por la empresa Arizona Microchip Technology (AMT).

El MPLAB incorpora todas las utilidades necesarias para la realización de cualquier proyecto y, para los que no dispongan de un emulador, el programa permite editar el archivo fuente en lenguaje ensamblador de nuestro proyecto, además de ensamblarlo y simularlo en pantalla, pudiendo ejecutarlo posteriormente en modo paso a paso y ver como evolucionarían de forma real tanto sus registros internos, la memoria RAM y/o EEPROM de usuario como la memoria de programa, según se fueran ejecutando las instrucciones. Además el entorno que se utiliza es el mismo que si se estuviera utilizando un emulador.

En las siguientes líneas se pretende ayudar a todos aquellos que se enfrentan por primera vez a este programa, tanto en su instalación como en la utilización de esta potente herramienta que nos proporciona Arizona Microchip Technology. En el CD-ROM que se adjunta con este curso se encuentran las versiones 4.12.00 y la 4.99.07 del programa, estas versiones y las nuevas que van saliendo cada poco tiempo y que incorporan nuevos tipos de microcontroladores, se pueden obtener de forma gratuita en la página web www.microchip.com, el la cual se encuentra una amplia información sobre todos los dispositivos que fabrica AMT.

De las dos versiones, nosotros vamos a centrarnos en la V.12.00, por

ser esta la que menos recursos de software y hardware necesita para trabajar con ella, además está pensada para trabajar con las herramientas de desarrollo MPLAB-ICD y el PICSTART que se encuentran ya muy difundidas, mientras que la versión V.99.07 está pensada para trabajar con el MPLAB-ICE 2000 soportado en NT, esta versión presenta algunas modificaciones en las ventanas de configuración del programa respecto a la anterior, además para su correcto funcionamiento es necesario disponer de la gama alta de los sistemas operativos que se indican en el siguiente apartado además del hardware más potente.

INSTALACIÓN DEL PROGRAMA

Los requerimientos mínimos para la instalación de los programas son:

- Procesador 386, 486 o Pentium*
- Windows 3.1/ 95/ 98, Windows NT 3.51/4.0, Windows 2000 ,MACOS 7.0, o Unix compatible OS
- 16 MB de memoria RAM para sistema con Windows 95.
- 24 MB de RAM para Windows NT systems.
- 32 MB para sistemas con Windows 2000.
- Unidad de CD-ROM.
- Navegador (3.0 HTML)

Se recomienda por AMT:

- Procesador Pentium
- 32 MB de memoria RAM

El CD-ROM de Microchip requiere para su navegación de un programa *HTML*. Para los equipos con Windows 95/98/NT se recomienda utilizar Internet Explorer versión 5.0 o el Netscape Navigator versión 4.0, además hará falta para la lectura de los numerosos documentos en formato *pdf* el programa Adobe Acrobat Reader versión 3.0 o 4.0.

Estos programas pueden obtenerse gratuitamente en las correspondientes web:

Microsoft Internet Explorer :
www.microsoft.com
Netscape Navigator:
www.netscape.com
Adobe Acrobat Reader:
www.adobe.com
Winzip: <http://www.pkware.com> o en
<http://www.winzip.com>

Al introducir el CD-ROM en la unidad correspondiente en los sistemas que tengan instalado Windows 95/98, Windows 2000 o NT y que tenga habilitada la opción *autorun*, aparece en pantalla el mensaje de la Figura 1, activaremos el botón de *Yes*.



Figura 1.- Mensaje al arrancar el CD-ROM

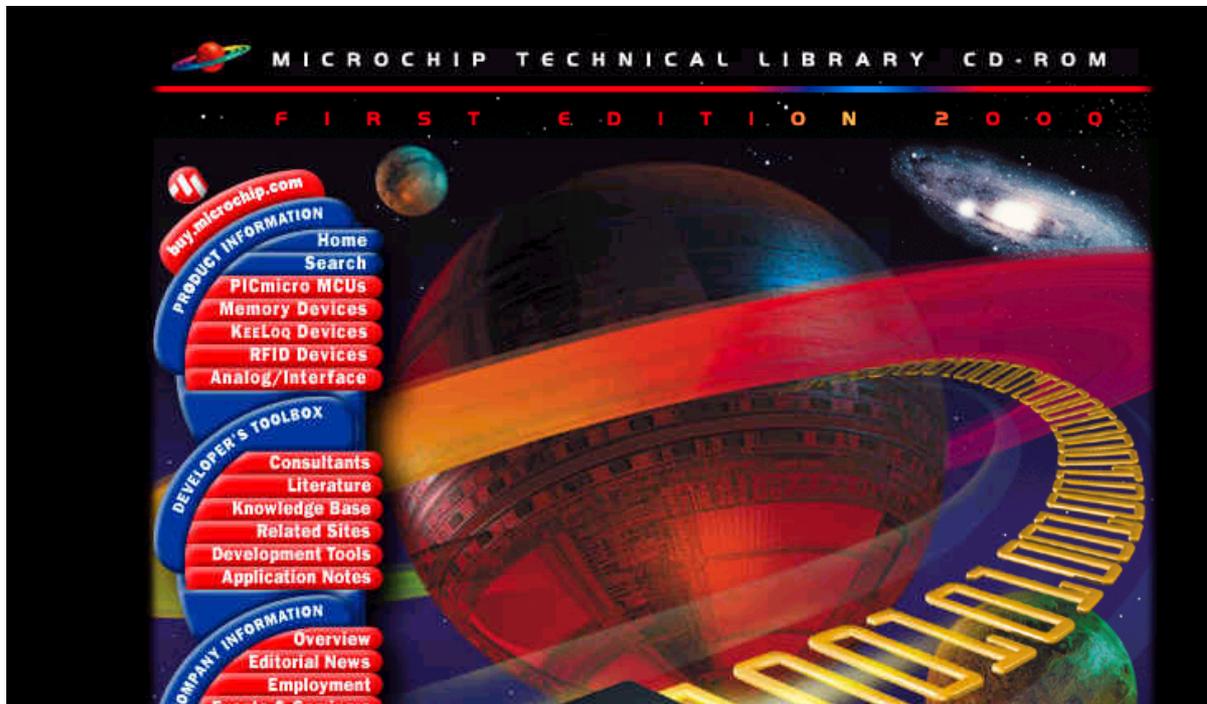


Figura 2.- Pantalla de presentación del CD-ROM y de la página web de Microchip

En caso de que no aparezca este mensaje al arrancar el CD-ROM, buscar con el explorador de Windows el archivo `\indextxt.htm` para el formato de texto o el `\index.htm` para el formato gráfico, al ejecutar este archivo aparecerá una pantalla como la que se muestra en la Figura 2, que es similar a la que tiene Microchip en su página web.

Seguidamente, activamos el botón de *Development Tools*, seleccionamos en la siguiente pantalla la opción *MPLAB-IDE* y al final de la página a la que nos lleva este vínculo, encontramos el enlace *MPLAB Version 4.12.00* que nos lleva a la página de descarga del programa *MPLAB (Disks 1 to 7)* (o los 7 discos individualmente) y que descargaremos sobre un directorio de nuestro disco duro que por ejemplo habremos llamado *C:\KK*. Hemos seguido estos pasos ya que son los mismos que deberemos seguir para descargar las nuevas versiones desde la web de Microchip, pero de una forma más directa, podríamos haber copiado directamente sobre el directorio *C:\KK* los archivos que están en el CD-ROM en el directorio *\download\Tools\PICmicro\DevEnv\MPLABi\Software\v412*. Estos archivos están comprimidos con las utilidades WinZIP. Deberán descomprimirse dichos archivos utilizando las herramientas WinZIP.

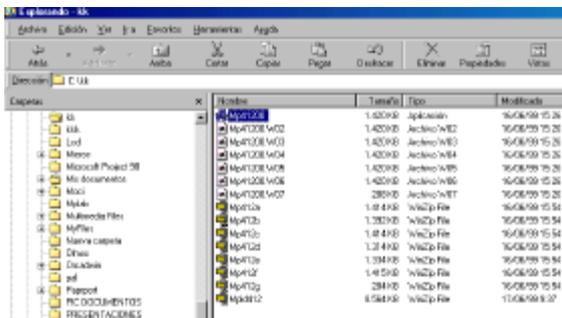


Figura 3.- Archivos en *C:\KK* obtenidos del CD y una vez descomprimidos con las utilidades WinZIP. Ejecutar *MP412000.exe*

Una vez que finalizemos la instalación del programa podremos borrar por completo el subdirectorio *C:\KK* para dejar espacio en el disco duro. Para instalar el programa se ejecuta el archivo *mp41200.exe*, cuando esto ocurre aparece la pantalla de la Figura 4.

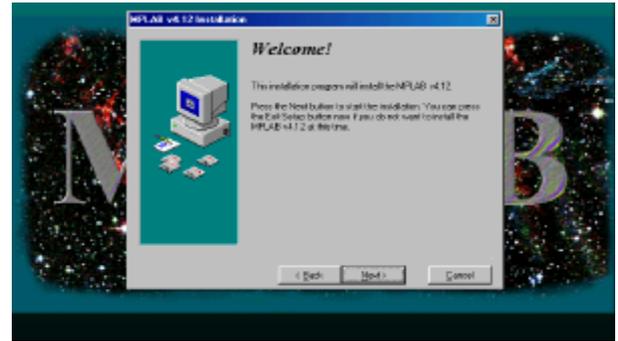


Figura 4.- Pantalla de bienvenida a la instalación del MPLAB.

Activamos el botón de *Next* y el programa responde con una nueva pantalla como la que se muestra en la Figura 5, con el menú de todos los módulos del programa que podemos instalar.



Figura 5.- No es necesario instalar todos los módulos del programa.

En principio, si no se dispone de emulador, podemos ahorrar espacio en el disco duro y no instalar estos módulos, para ello, se desactivan dichos campos tal y como se muestra en la Figura 5, se activa el botón de *Next*, para pasar a la siguiente pantalla. Igualmente pulsamos el botón de *Next* en la pantalla de *Select Language Components* en donde dejaremos activos todos los campos. Aparece la pantalla de la Figura 6 en la que se debe seleccionar el directorio donde queremos que se instale el programa, por defecto lo hace en el directorio *C:\Archivos de programas\MPLAB*.

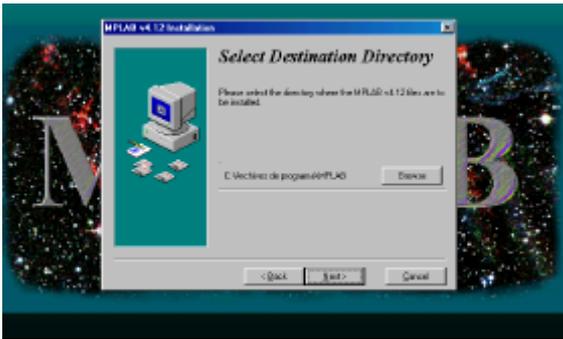


Figura 6.- Selección del subdirectorio donde se instalará el programa MPLAB

Una vez finalizada la instalación que puede tardar unos minutos, se puede pasar a ejecutar el programa MPLAB, es recomendable que si se va a utilizar mucho este programa, lo cual esperamos, se haga un acceso directo a dicho programa con lo que tendremos en el escritorio de Windows un icono como el de la Figura 7.



Figura 7.- Icono de acceso directo al programa MPLAB

Antes de seguir adelante, recomendamos crear una carpeta con el explorador de Windows por ejemplo *C:\Archivos de programas\MPLAB\Trabajo*, dentro de la cual posteriormente iremos creando todos nuestro proyectos.

COMO EMPEZAR.

Cuando se pulsa el icono del MPLAB aparece una pantalla como la que se muestra en la Figura 8.

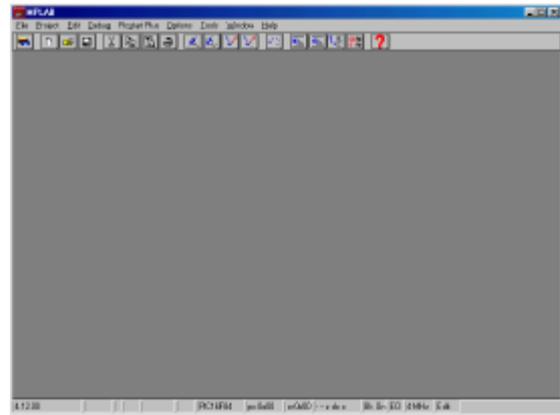


Figura 8.- Escritorio del MPLAB

Lo primero que haremos es seleccionar el modo de trabajo como simulador y el tipo de microcontrolador con el que queremos trabajar. Para ello se selecciona el botón de *Options* de la barra del control que aparece en el escritorio y del menú desplegable la opción *Development Mode*, con lo que aparece la pantalla de la Figura 9 en la que se activa el modo *MPLAB-SIM simulator* y el microcontrolador con el que se desea trabajar, que en nuestro caso será el *PIC16F84*, por último, pulsamos el botón de *Reset* para aceptar los cambios.

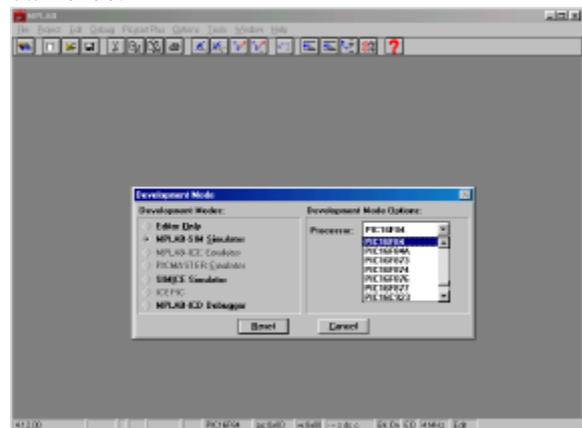


Figura 9.- Selección de la opción de trabajo como simulador y el tipo de microcontrolador

Los iconos que aparecen en la barra de herramientas, son funciones que se encuentran incluidas en el menú de control, pero como en todos los programas de Windows se incluyen para manejar de forma más cómoda el programa. Seguidamente comentaremos

que significa cada uno de los iconos de la barra de herramientas que aparece en esta pantalla, mas adelante veremos que

hay más barras de herramientas que pueden ser conmutadas

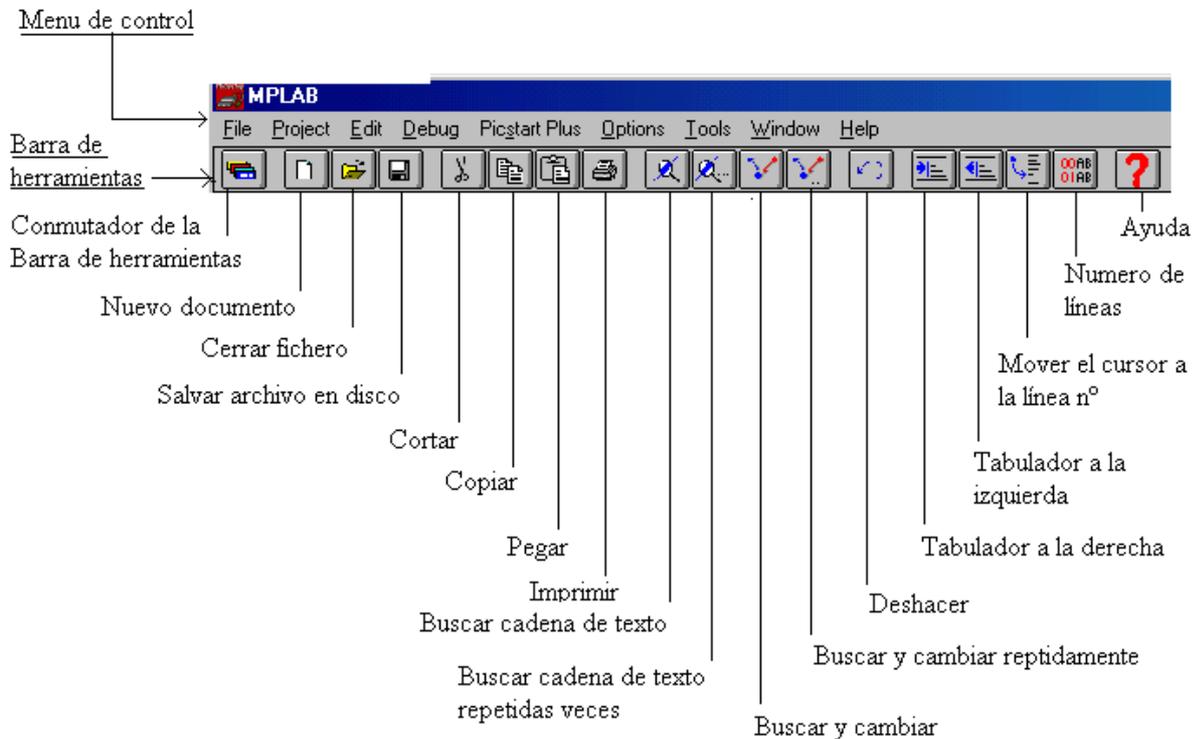


Figura 10.- Barra de herramientas de edición

NUESTRO PRIMER PROYECTO

Bueno, pues ya estamos en condiciones de crear nuestro primer proyecto, para ello comenzamos por activar en el menú de control la opción *File>New* o bien activamos el icono de *crear nuevo documento* en la barra de herramientas. El programa contestará con el cuadro de diálogo de la Figura 11.

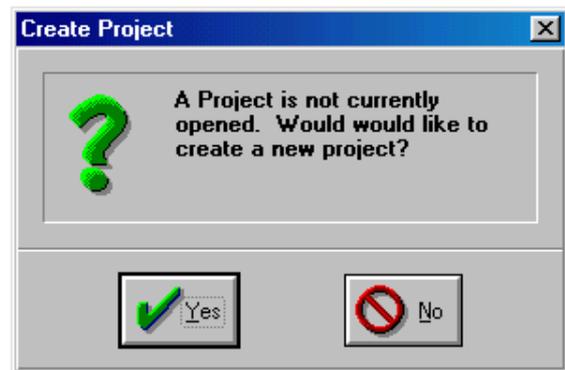


Figura 11.- No hay ningún proyecto abierto ¿Quiere crear un nuevo proyecto?

Activamos el botón de **Yes** y aparece un cuadro de dialogo como el de la Figura 12 en el que se nos pide el nombre del proyecto que tendrá extensión ***.pjt** , como este es nuestro primer proyecto le llamaremos **ejer1.pjt** y lo guardaremos en la carpeta de **trabajo** que habíamos creado anteriormente.

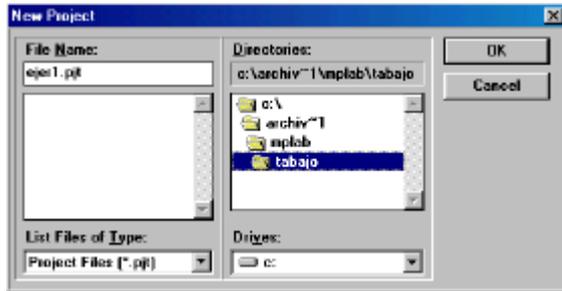


Figura 12.- Creación de un nuevo proyecto

El programa devuelve el cuadro de diálogo de la Figura 13

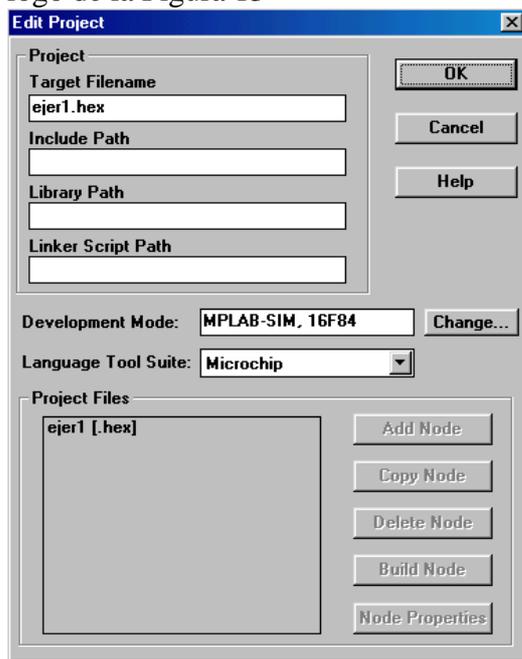


Figura 13.- Propiedades de edición del proyecto

Activamos el botón de **OK** y estamos en condiciones de empezar a escribir nuestro primer proyecto al aparecer una pantalla como la de la Figura 14

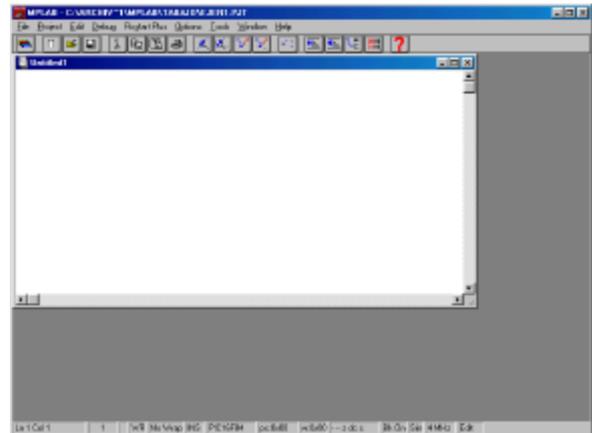


Figura 14.- Apertura del documento para comenzar a escribir nuestro proyecto

EL EDITOR

Comencemos por lo tanto a escribir en lenguaje ensamblador nuestro primer programa que llamaremos **ejer1.asm** y que se muestra en la Figura 15. El programa realiza la suma en binario de dos números ($7+8 = 15$) y para escribirlo usamos el editor de textos. La extensión ***.asm** es la que deben llevar todos los programas escritos en ensamblador.

Deberemos de tener en cuenta que la primera columna del editor está reservada para las **etiquetas** que son expresiones alfanuméricas escogidas por el usuario que definen valores de posiciones de memoria. Estas deben empezar siempre por una letra. Además se debe de tener en cuenta que no pueden usarse expresiones que ya utiliza el ensamblador tales como:

- Instrucciones
- Directivas del propio ensamblador
- Nombres de registros especiales (SFR)
- Nombre de cada uno de los bit de los registros especiales.

En las siguientes columnas, se puede comenzar a escribir el nemónico de la instrucción o las directivas del ensamblador. Por último hay que decir que se pueden y se deben añadir comentarios que son elementos indispensables en muchos casos para seguir el razonamiento de los programas sin perderse,

para ello cuando el MPLAB encuentra un “;”(punto y coma) no se genera código máquina.

En todos estos campos los espacios en blanco no son significativos y las líneas en blanco tampoco.

Para una mejor legibilidad del programa, se recomienda acceder a cada campo utilizando el tabulador.

El uso de mayúsculas y minúsculas en los programas obedece a una serie de reglas o normas de estilo, comunes entre los programadores en ensamblador, que si bien no son obligatorias, facilitan la

lectura del código fuente. Estas reglas son:

- Las directivas del ensamblador se escriben en mayúsculas
- Los nombres de las variables se escriben en mayúsculas.
- Los nemónicos de las instrucciones se escriben en minúsculas
- El programa se escribe utilizando los tabuladores para definir las distintas columnas, tales como etiquetas, comienzo de líneas de programa y columna donde empiezan los comentarios separados por un “;” (punto y coma).

```
*****
; Programa Ejer1.ASM                               Fecha : 1 - Marzo - 2000
; Este programa suma dos valores inmediatos (en este caso 7+8) el resultado se deposita
; en la posición 0x10
; Revisión : 0.0                                     Programa para PIC16C84 y PIC16F84
; Velocidad del Reloj: 4 MHz                         Reloj Instrucción: 1 MHz = 1 uS
; Perro Guardian : deshabilitado                    Tipo de Reloj : XT
; Protección del código : OFF
*****

LIST      p=16F84      ;Tipo de procesador
RESULTADO EQU      0x10      ;Define la posición del resultado
          ORG      0x00      ;Vector de Reset
          goto    INICIO
          ORG      0x05      ;Salva el vector de interrupción
INICIO    movlw   0x07      ;Carga 1er. sumando en W
          addlw  0x08      ;Suma el 2º sumando
          movwf  RESULTADO  ;Almacena el resultado
STOP     sleep          ;Poner ponerse a dormir
          |
          END            ;Fin del programa fuente
```

Figura 15.- Nuestro primer programa ejer1.asm

Cuando terminemos de escribir el programa seleccionamos **File>Save** con lo que aparece el cuadro de diálogo de la Figura 16, donde le damos el nombre a nuestro programa **ejer1.asm**, dentro de nuestra carpeta **Trabajo**.

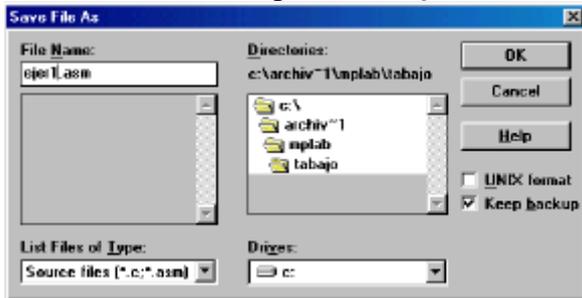


Figura 16.- Salvar el programa ejer1.asm en la carpeta de trabajo

El siguiente paso será volver a editar nuestro proyecto seleccionando en el menú de control **project>edit project**, lo que provoca que aparezca el menú de la Figura 17.

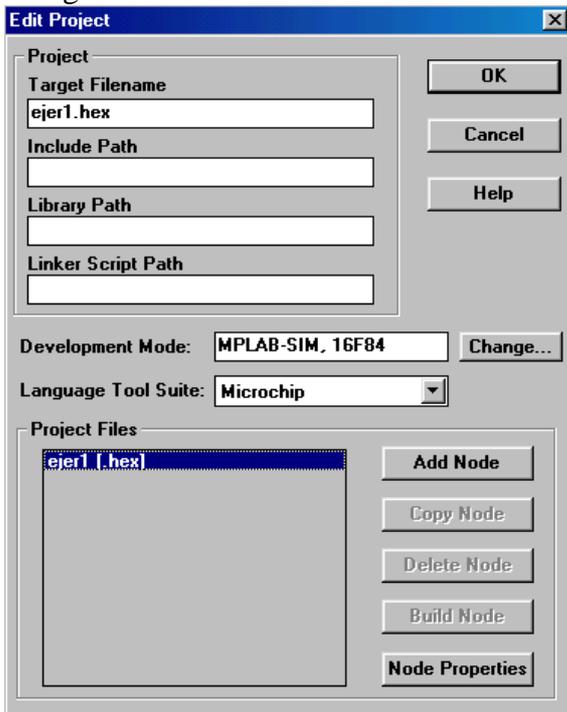


Figura 17.- Pantalla de edición del proyecto

Pulsamos sobre **ejer1[.hex]**, y se activa el botón de **Node Properties**, que hasta el momento aparecía de color gris, si lo activamos aparece el cuadro de diálogo de la Figura 19, donde están reflejadas todas las propiedades del nodo actual. Sin modificar ninguna de estas propiedades se pulsa el botón de **OK** para continuar, lo que nos lleva de nuevo a la pantalla de la Figura 17. Ahora seleccionamos el botón **Add Node** (añadir elementos al nodo), lo que provoca que aparezca un nuevo cuadro de diálogo como el de la Figura 18, en el que seleccionaremos el archivo **ejer1.asm**

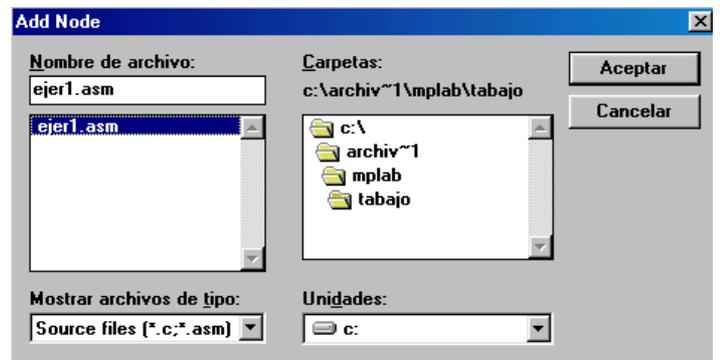


Figura 18.- Nombre del archivo a incluir en el proyecto ejer1.asm

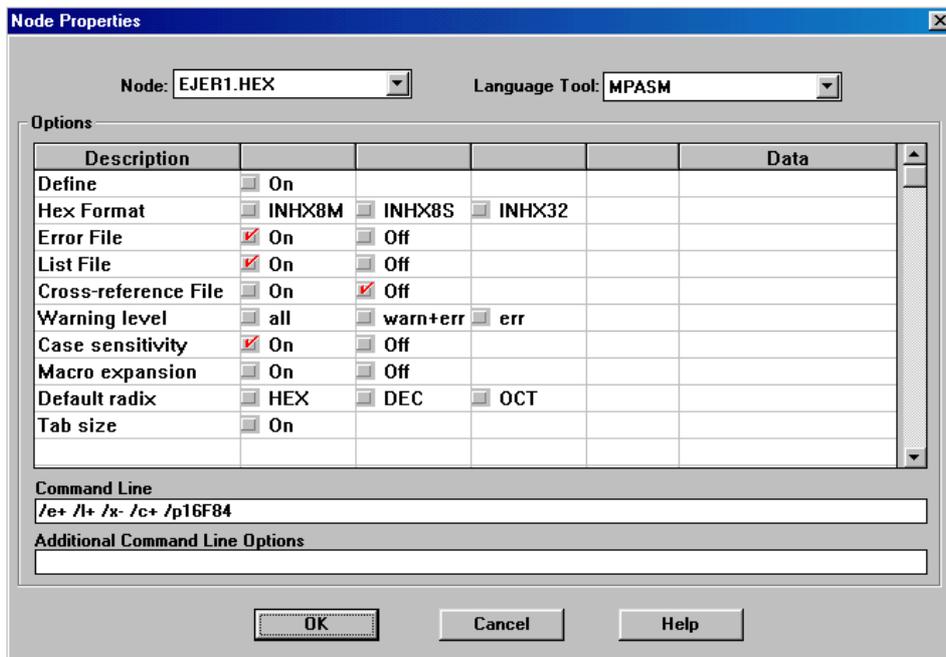


Figura 19.- Propiedades del nodo de nuestro proyecto donde se seleccionan los ficheros y formatos que se obtendrán al ensamblar el programa.

Pulsamos el botón de *Aceptar* y se vuelve a la pantalla de la Figura 17 en la que ha aparecido el fichero *ejer1[.asm]* junto al fichero *ejer1[.hex]* que aparecía antes en el campo de *Project files*. Seguidamente pulsamos el botón de *OK*, lo que nos llevará de vuelta a la pantalla de la Figura 15. Para ensamblar el programa seleccionamos en el menú de control la opción *Project>Build All* (también podríamos haber pulsado el botón correspondiente de la barra de herramientas del simulador



), como luego veremos), y si no se han cometido errores al introducir los códigos, aparece una pantalla como la de la Figura 20, lo que nos indica que el programa se ha ensamblado con éxito y ya estamos en condiciones de iniciar la simulación del programa. Si por el contrario, se han detectado errores, en dicha pantalla será mostrado el error; si se hace doble clic sobre la línea que muestra el error, el cursor saltará directamente a la línea de código donde se encuentra el error. Una vez subsanados los errores habrá que volver a compilar el programa.

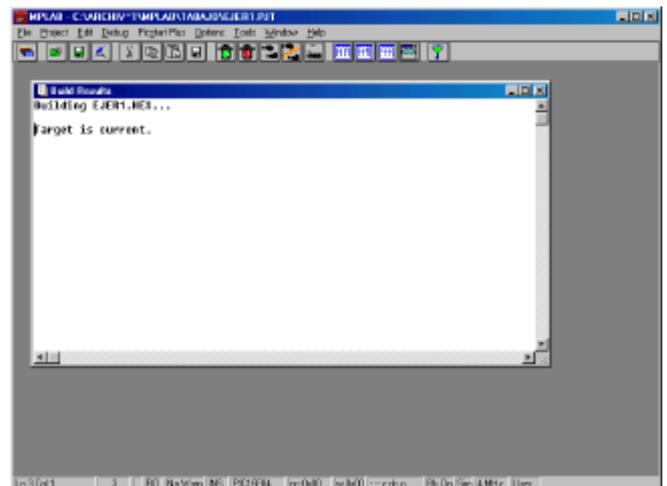


Figura 20.- Pantalla del MPLAB una vez ensamblado correctamente el programa fuente

LA BARRA DE MENÚS

Seguidamente analizaremos las distintas posibilidades de la barra de menú del MPLAB, si bien ya hemos utilizado algunas de las posibles opciones que presenta la barra de herramientas, ahora analizaremos estas una por una.

1.- **Windows:**

Al activar esta opción de la barra de menú, aparece el menú desplegable de la Figura 21.



Figura 21.- Menú desplegado de la opción Windows de la barra de herramientas.

1.1.- Program Memory : Al seleccionar esta opción aparece la pantalla de la Figura 22 en la que se puede apreciar las posiciones de memoria que ocupa cada una de las instrucciones, el código de operación de cada instrucción y la posición de memoria que se le ha dado a cada etiqueta.

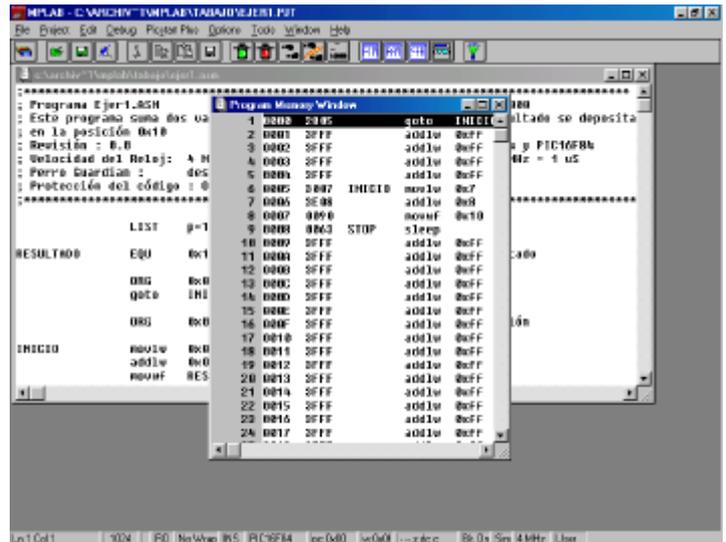


Figura 22.- Pantalla de la ventana de Memoria de programa.

Si hacemos clic sobre la barra de *menú del sistema*, activando el icono que hay en la parte superior izquierda de esta ventana, aparece el menú desplegable de la Figura 23, en el que se puede seleccionar entre tres formas de ver la memoria de programa:

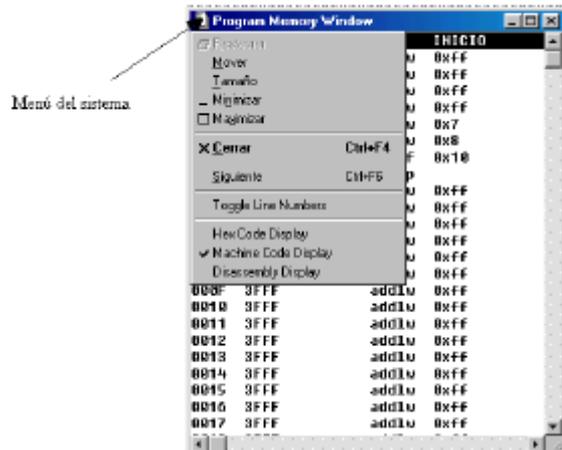


Figura 23.- Despliegue de opciones del menú de sistema.

- **Hex Code Display:** representa la memoria de programa con los datos en hexadecimal. Esta opción es muy útil al usar el programador del dispositivo y comprobar si se grabaron bien los datos. La pantalla que se obtiene es la que se muestra en la Figura 24.

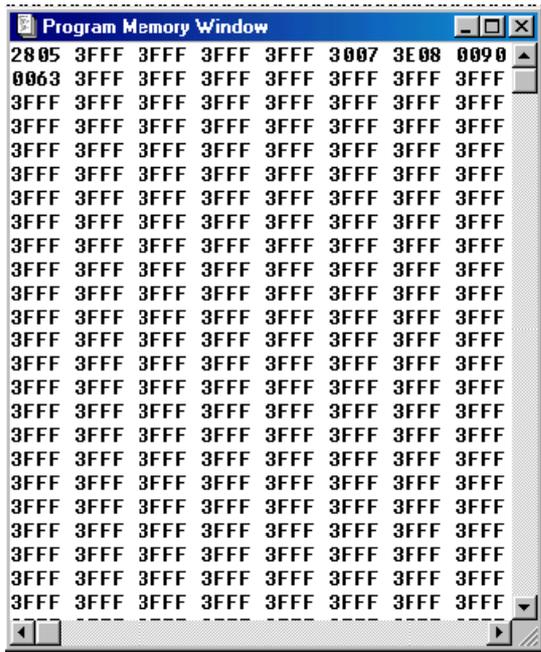


Figura 24.- Memoria de programa en código hexadecimal.

- **Machine Code Display:** esta opción presenta el código máquina ensamblado tal y como se ve en la Figura 22. Con la información de las etiquetas y direcciones de memoria que tienen asignadas.
- **Disassembly Display:** despliega el código hexadecimal desensamblado con los símbolos.

Cuando la ventana está en la opción **Machine Code** o **Disassembly Display**, la instrucción a la que apunta el contador de programa, está resaltada.

1.2.- Trace Memory : La ventana de memoria de traza toma “una instantánea” de la ejecución del programa,

cuando este está corriendo en tiempo real.

Para emuladores que tienen un buffer de traza, que se utiliza cuando el programa corre en tiempo real y este no se puede detener en algunas aplicaciones, nos muestra los puntos por los que pasa el programa. Algunos problemas sólo aparecen cuando la aplicación está corriendo, es decir, estos no dan la cara cuando se ejecuta en modo paso a paso. La memoria de traza es una herramienta de depuración para probar tales aplicaciones. Para más información es recomendable mirar en la guía de usuario del emulador que se esté utilizando. En el simulador, el buffer de traza o memoria de traza, es útil para visualizar un registro a lo largo de la ejecución del programa, de manera, que se puede registrar por donde pasa el programa y después analizarlo. El simulador toma datos de forma un poco distinta que el buffer del emulador.

Antes de activar la **opción Trace Memory**, para poder obtener los datos en la memoria de traza en el simulador, es necesario marcar con el ratón las líneas de código de programa de las cuales queremos obtener los datos al ejecutarse el programa, estas pueden estar en un bloque de instrucciones o bien colocadas en el programa de forma discontinua. Seguidamente, se pulsa el botón de la derecha del ratón, de manera que aparece el menú desplegable de la Figura 25.

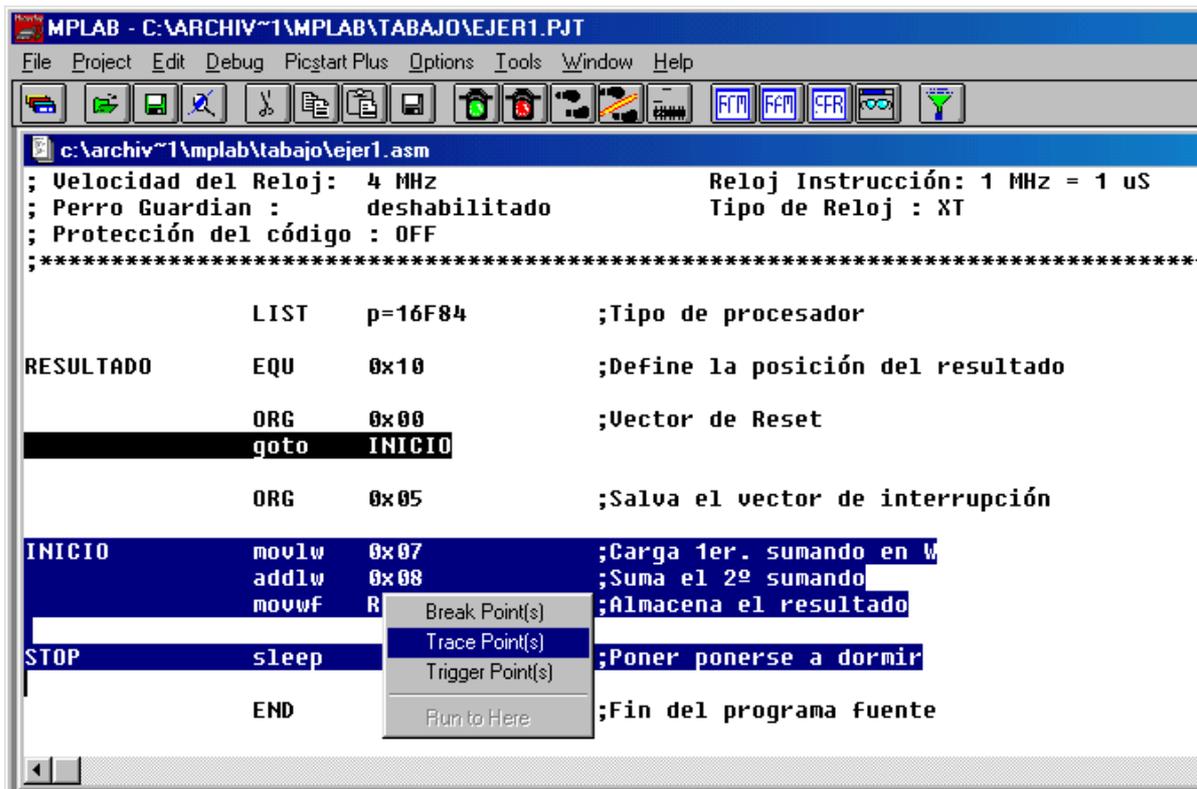


Figura 25.- Selección de las líneas de programa para cargar el buffer de traza.

Al seleccionar la opción *Trace Point(s)* aparecen resaltadas las líneas en color verde. Seguidamente se activa el icono del semáforo verde  (*Run*), lo que hará ejecutar la simulación en “tiempo real” (no olvidemos que en el simulador emula el funcionamiento del microcontrolador y es mucho más lento que este), y después de unos segundos, si activa-

mos el icono del semáforo rojo , se detiene la ejecución del programa. Si ahora se activa dentro de la opción *Window>Trace Memory*, se pueden ver la traza obtenida, que en nuestro caso en la que se muestra en la Figura 26.



Figura 26.- Trazo de memoria obtenida en el programa con los valores marcados en la Figura 25

El simulador muestra en esta ventana el valor del tiempo que tarda en ejecutar cada línea de programa y también cualquier variación sobre los registros al ejecutarse el código de instrucción.

1.3.- EEPROM Memory: Si el dispositivo emulado tiene EEPROM o memoria

Flash, como es el caso del PIC16C84 o el 16F84 respectivamente, el contenido de la memoria EEPROM puede verse seleccionando la opción *Window>EEPROM*.

La memoria de EEPROM no puede modificarse a través de esta ventana. Para ello hay que utilizar el menú de dialogo al que se accede seleccionando

Window>Modify..., que se describe más adelante.

1.4.- Absolute Listing: La Ventana de “Listado de Programa”, realmente nos presenta el archivo de nuestro proyecto con extensión **.lst*, donde se puede ver

el archivo generado por el ensamblador o compilador. El listado muestra el código fuente en modo absoluto con el código objeto generado, tal y como se puede ver en la Figura 27.

```

c:\archiv~1\mplab\tabajo\ejer1.lst
MPASM 02.30 Released          EJER1.ASM    2-20-2000  10:28:18          PAGE 1

LOC  OBJECT CODE      LINE SOURCE TEXT
VALUE

00001 ;*****
00002 ; Programa Ejer1.ASM                      Fecha : 1 - Marzo
00003 ; Este programa suma dos valores inmediatos (en este caso 7+8) el
00004 ; en la posición 0x10
00005 ; Revisión : 0.0                          Programa para PIC1
00006 ; Velocidad del Reloj: 4 MHz              Reloj Instrucción:
00007 ; Perro Guardian :      deshabilitado     Tipo de Reloj : X1
00008 ; Protección del código : OFF
00009 ;*****
00010
00011          LIST      p=16F84          ;Tipo de procesador
00012
000000010 00013 RESULTADO EQU      0x10          ;Define la posición del re
00014
00000 00015          ORG      0x00          ;Vector de Reset
0000 2805 00016          goto    INICIO
00017
0005 00018          ORG      0x05          ;Salva el vector de interr
00019
0005 3007 00020 INICIO    movlw   0x07          ;Carga 1er. sumando en W
00021 0500 00021          -+1... 0x00          ;Carga el 2º sumando
  
```

Figura 27.- Archivo ejer1.lst

Además al final de este archivo aparece la información de las etiquetas utilizadas en el programa, en que línea de programa se encuentran, la memoria utilizada, la memoria libre además de los errores, *warnings* y mensajes reportados por el ensamblador.

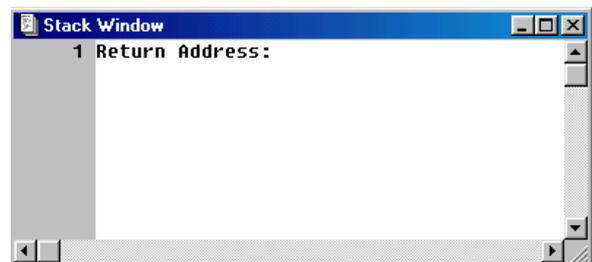


Figura 28.- Ventana de la Pila

1.5.- Stack: El contenido de la pila puede verse al seleccionar la opción **Window>Stack**.

Los contenido de la pila puede mostrarse con o sin número de línea. El formato de presentación se selecciona a través del menú del sistema. Si la Pila se desborda, el MPLAB indica con su rebozamiento con el mensaje *underflow*.

Al menú del sistema se accede pulsando el botón de la esquina de la ventana.

1.6.- File Registers : La lista de registros de propósito general (GPR) del microcontrolador, que son de memoria SRAM, se pueden ver seleccionando la opción **Window>File**. Esta ventana al desplegarse presenta una lista con todos los registros de propósito general del dispositivo emulado, tal y como se muestra en la Figura 29.

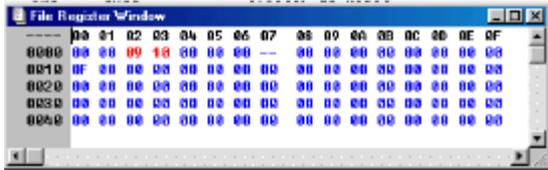


Figura 29.- Listado de registros de propósito general del sistema

Este listado de registros pueden visualizarse de tres maneras. El formato deseado se elige a través del menú del sistema.

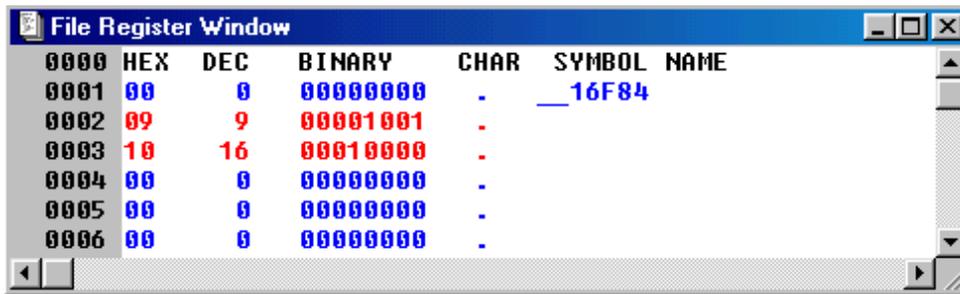


Figura 30.- Formato *Symbolic Display* del listado de registros de propósito general

•*ASCII Display* .– Esta opción presenta un listado de los registros de propósito general con el contenido de los datos en código ASCII.

Se puede modificar el contenido de uno o varios registro en esta ventana. Par ello se pone el puntero del ratón sobre el primer registro que se quiere modificar y pulsando el botón izquierdo se marca el bloque de los registros que se quieren alterar, si sólo se quiere modificar, bastará con ponerse encima de él, seguidamente se pulsa el botón de la derecha y se activa la opción *Fill Register(s)*, lo que hará aparecer la ventana de la Figura 40, en la que se puede ver como aparece la dirección del registro a modificar, una líneas mas adelante se analizan las posibilidades de esta opción.

1.7.- Special Function Registers (SFRs): El contenido de los registros de funciones especiales (FSR) pueden verse seleccionando *Window>Special Function Registers*. El formato proporcionado por esta ventana es más útil para analizar el estado de los FSRs en cada momento, además como puede verse en la Figura 31 se muestra cada uno de los registros

•*Hex Display*.- Esta opción presenta los registros con datos en hexadecimal, tal y como se ve en la Figura 29.

•*Symbolic Display*.- Este formato presenta un archivo con los registros de propósito general con sus etiquetas si las tienen y su contenido en hexadecimal, decimal, binario y formato carácter tal y como se puede ver en la Figura 30.

con el nombre que tiene asignado además de su contenido en distintos códigos: hexadecimal, decimal, binario y formato carácter o ASCII.

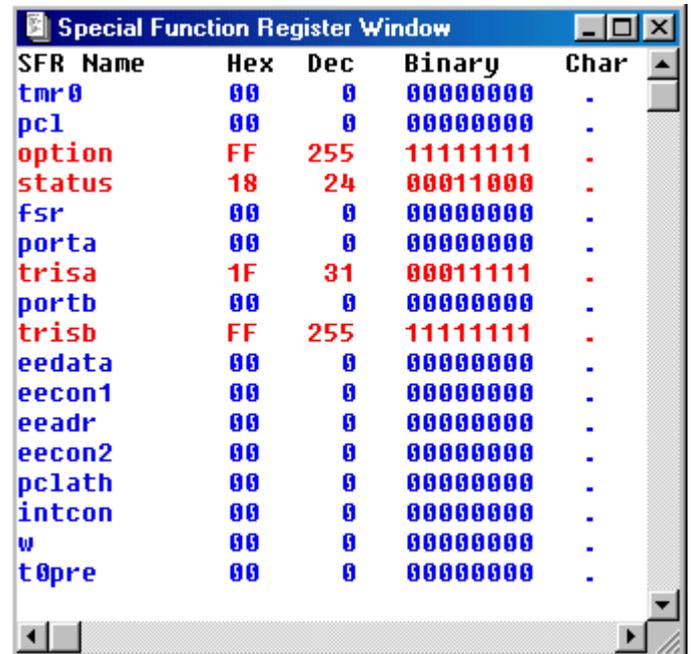


Figura 31.- Ventana de los registros especiales (SFRs).

Para modificar un SFR, hacer doble clic sobre el nombre del registro, esta acción hará aparecer el cuadro de

dialogo de la opción *Modify* (Figura 40) en la que aparecerá ya la dirección del registro seleccionado.

1.8.- Show Symbol List :

Esta ventana muestra un listado de los símbolos, es decir variables y etiquetas, utilizadas en el código fuente del programa.

Estos símbolos están en el archivo *.COD de nuestro proyecto. En la Figura 32 se muestra el listado de símbolos de nuestro programa.

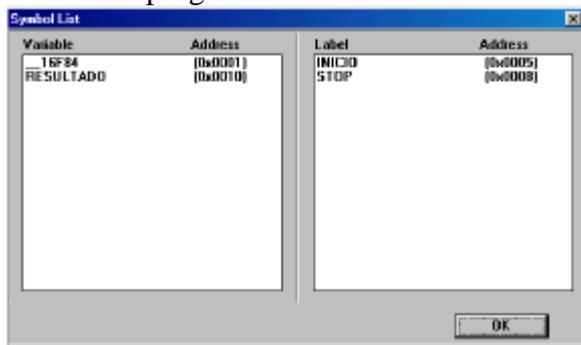


Figura 32.- Ventana del listado de símbolos utilizados en nuestro programa fuente.

1.9.- Stopwatch and Clock Frequency :

Para calcular el tiempo de ejecución de nuestro programa o de una subrutina, podemos contar el número de instrucciones que se realizan y multiplicarlo por 4 veces la frecuencia de la señal de reloj (tiempo de un ciclo máquina) o por 8 en el caso de que las instrucciones sean de salto. Esto en algunas ocasiones es engorroso, pero el MPLAB con esta opción de cronómetro nos permite medir tiempo de ejecución de las instrucciones de nuestro programa sin equivocaciones.

El cronómetro calcula el tiempo basándose en la frecuencia del reloj del microcontrolador PIC que estamos simulando, para ello previamente debemos fijar la frecuencia del oscilador empleado. Esto se realiza haciendo los siguientes pasos: Activamos desde el menú *Options>Processor Setup>Clock frequency* tal y como se muestra en la Figura 33

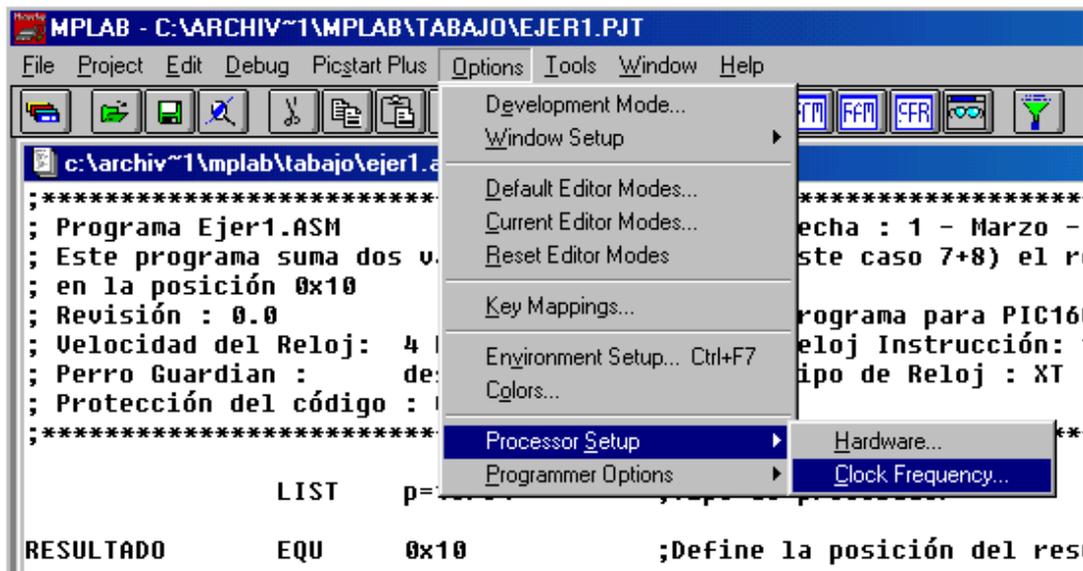


Figura 33.- Camino a seguir para definir la frecuencia del microcontrolador *Options>Processor Setup>Clock frequency*

Inmediatamente se abre un cuadro de dialogo como la de la Figura 34, donde se fija la frecuencia del reloj.

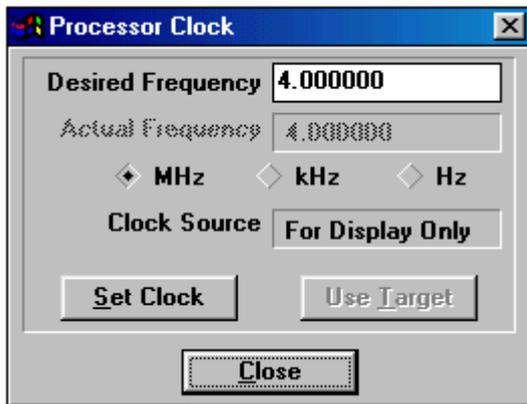


Figura 34.- Definición de la frecuencia de oscilador del microcontrolador.

Después se activa la opción *Windows>Stop Watch*, con esto conseguimos tener siempre abierta la ventana que muestra el tiempo transcurrido y los ciclos máquina empleados en la ejecución de cada instrucción, como puede verse en la Figura 35.

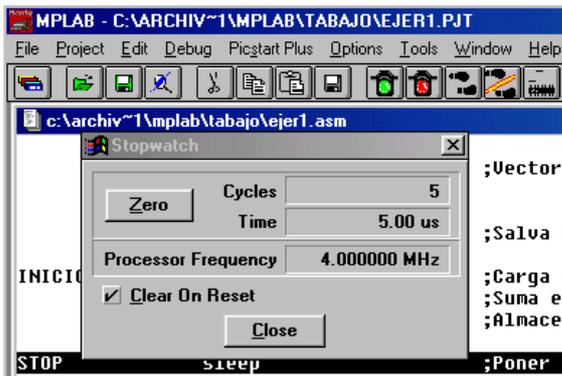


Figura 35.- Cronometro para contar el tiempo que tarda en ejecutarse un programa o parte de él.

1.10.- Project Windows : La Ventana del Proyecto sólo está disponible cuando hay un proyecto abierto. Presenta la lista de archivos que actualmente hay en dicho proyecto. Si el proyecto se ha ensamblado o compilado la ventana del proyecto muestra una lista de todos los archivos incluidos en el proyecto.

Por otra parte, la ventana del Proyecto sólo presenta el archivo del proyecto principal. Un doble clic en cualquier archivo resaltado en la ventana

del proyecto, abrirá dicho archivo para su revisión.

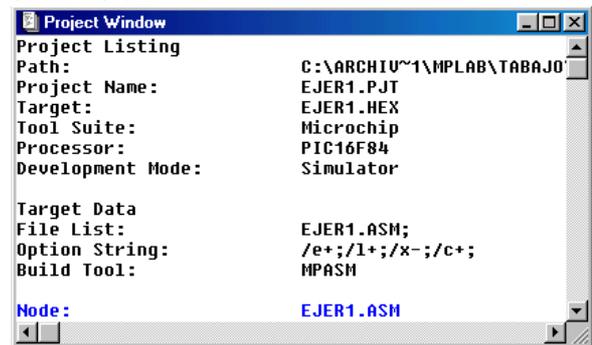


Figura 36.- Ventana de Proyecto

1.11.- Watch Windows :

MPLAB permite supervisar los contenidos de los registros del archivo a través de una ventana temporal. Para abrir una ventana temporal, se selecciona *Window>Watch Windows*. El programa responde con un cuadro de diálogo como el de la Figura.37

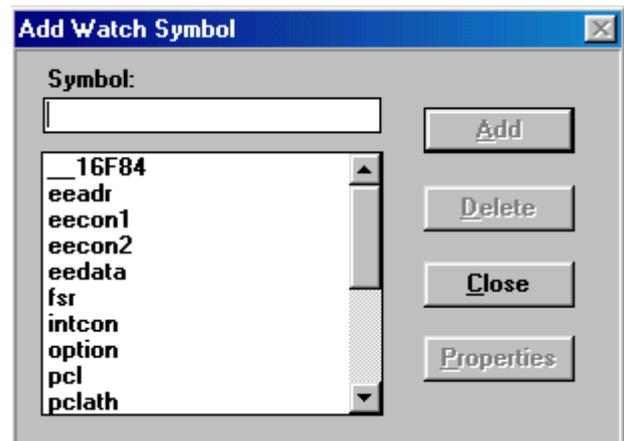


Figura 37.- Cuadro de diálogo de los símbolos de la ventana temporal

Para agregar los registros a visualizar, poner el ratón encima de uno de ellos pulsar el botón de la izquierda, seguidamente activar el botón de *Add*. También se pueden anular los símbolos poniéndose sobre ellos y pulsando el botón izquierdo del ratón y seguidamente el botón de *Delete*. Cuando estén todos los registros seleccionados pulsar el botón de *Close* y aparecerá una ventana, en este primer caso, **Watch_1**, como puede

verse en la Figura 38, en la que se ven los símbolos (etiquetas y variables) seleccionados.

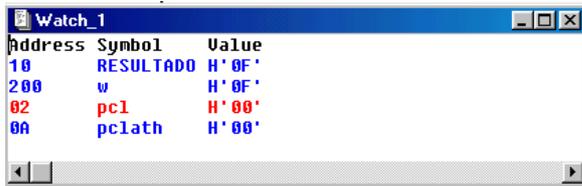


Figura 38.- Ventana Watch_1

Para ver y cambiar las propiedades de un símbolo, hay que pulsar el botón de propiedades que aparece en el cuadro de diálogo de la Figura 37, al hacerlo aparece un nuevo cuadro de diálogo como el de la Figura 39

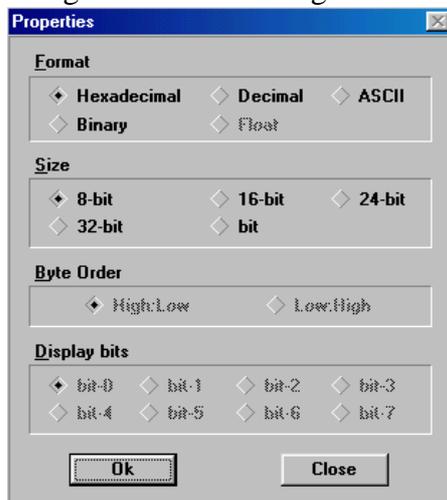


Figura 39.- Cuadro de diálogo de las propiedades de los datos de las ventanas temporales

El contenido de la ventana del reloj puede desplegarse mostrando o sin mostrar los números de la línea. Para elegir el formato deseado se hace a través del menú del sistema (pulsando el icono de la parte superior izquierda de la ventana). El menú del sistema también se usa para revisar la información en la ventana temporal.

La ventana de diálogo permite seleccionar el formato en que se presentan los símbolos:

- Format.-** Determina qué tipo de numeración se desea visualizar.
- Size.-** Determina cuántos bytes serán incluidos en la visualización del número:

hexadecimal, decimal, binario, ASCII o float.

•**Byte Order.-** Determina el orden de visualización de cada byte, disponible sólo para los números de 16 bits.

•**Display Bits.-** Determina en qué momento se visualiza el bit seleccionado al activarse.

1.12.- Modify: Al activar la opción **Window>Modify...** aparece el cuadro de diálogo **Modify** como el que se muestra en la Figura 40. En este cuadro se permite leer y escribir una posición de memoria o el rango de una posición de memoria. **Modify** puede trabajar en las áreas de memoria siguientes:

- Data
- Stack
- Program
- EEPROM (Si tiene)

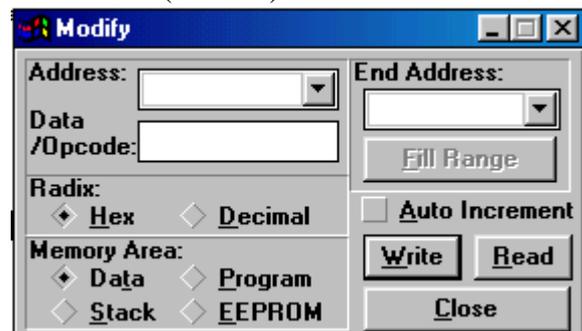


Figura 40.- Cuadro de diálogo de la opción Modify

Como resumen a todo lo que hemos contado hasta el este momento, podemos decir que el comando **Windows**, puede presentar una visión de todos los registros del microcontrolador en cada momento y podemos tener al final una pantalla en la que visualicemos según nos interese las ventanas mas adecuadas para el seguimiento de nuestra aplicación, como puede ser la de la Figura 41.

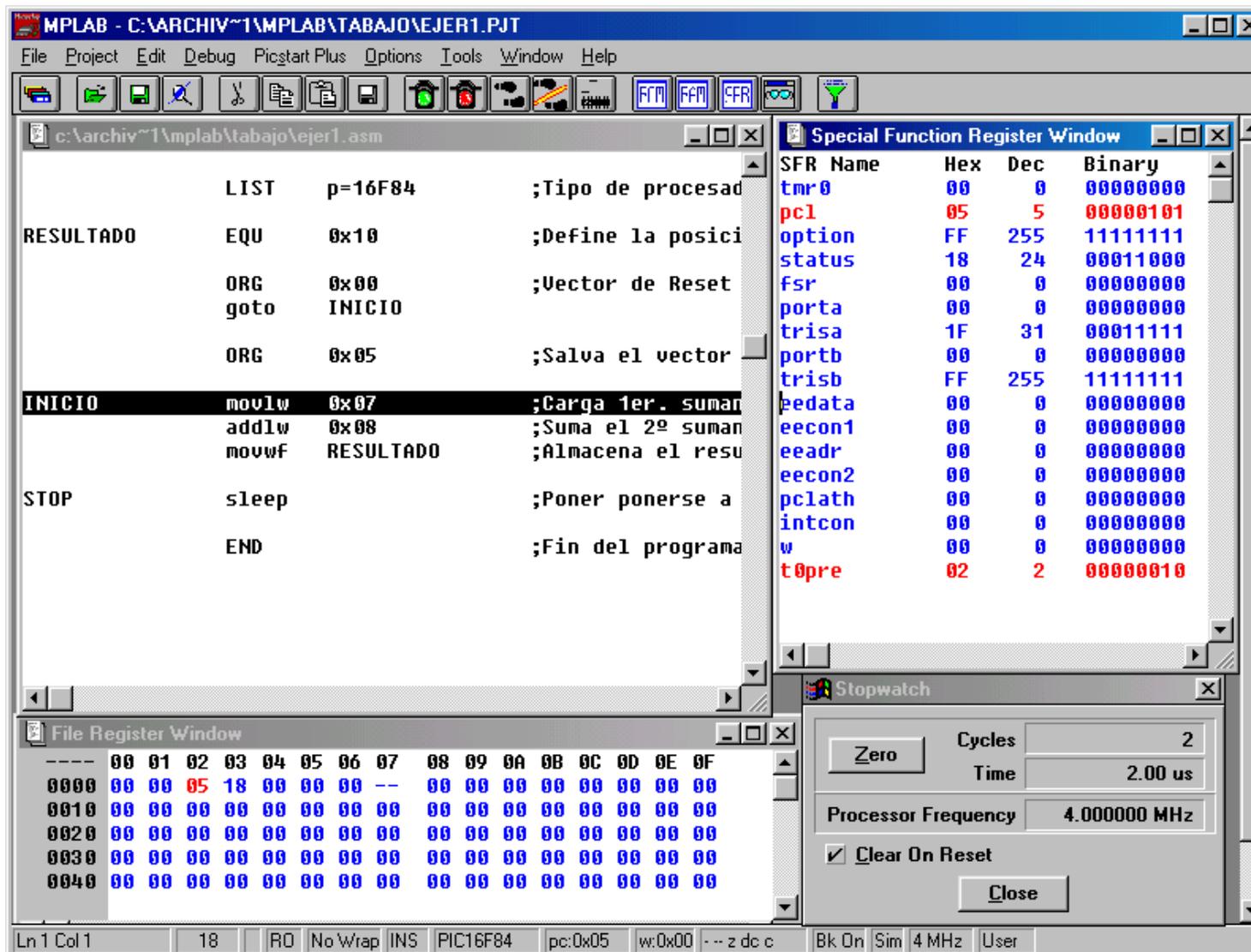


Figura 41.- Presentación de algunas ventanas de forma simultánea en pantalla