

## Tutorial MPLAB v6.X

### PROYECTO

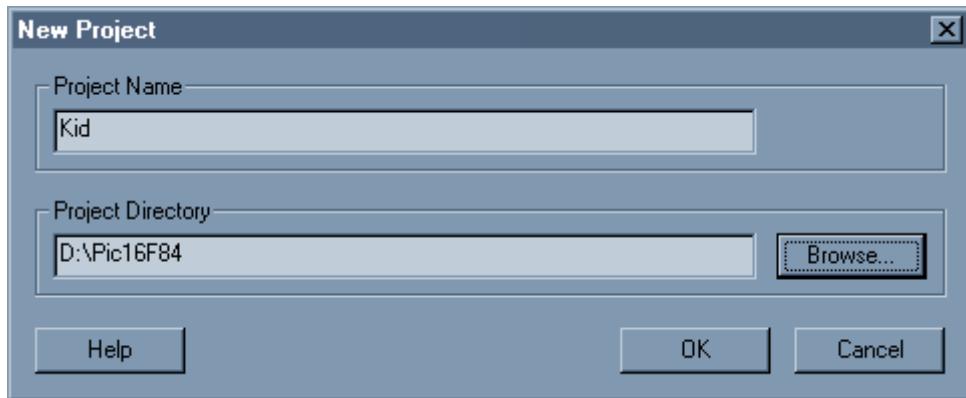
#### Creación de Proyecto

Un proyecto es un conjunto de archivos fuente e instrucciones que permiten construir el objeto y código ejecutable para una aplicación.

Para crear un nuevo proyecto se debe ir a la opción *New* en el campo *Project*.



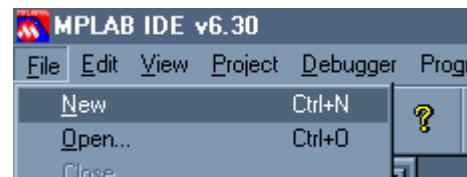
Se abrirá una ventana en la que se introducirá el nombre del proyecto en el campo *Project Name*, y se elegirá la carpeta en el campo *Project Directory* en donde se guardará el proyecto. Una vez elegidos estos, se acepta en el botón *OK*. Los proyectos en la versión MPLAB 6.3 tienen la extensión *\*.mcp*.



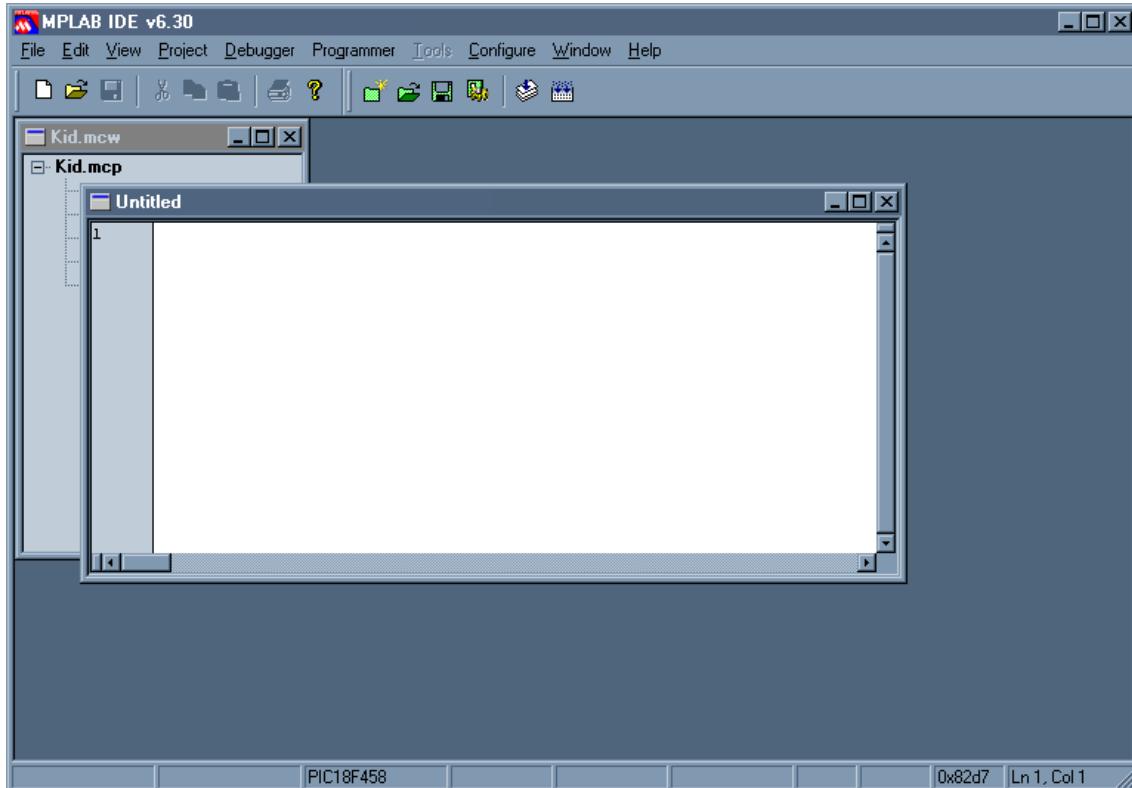
#### Creación de Archivo de Trabajo

Se debe crear un archivo de texto en donde se va a escribir el programa que queremos ejecutar.

Para ello se crea un nuevo archivo en la opción *New* del menú *File*.



Aparece una nueva ventana, como muestra la figura inferior, con la palabra *Untitled*, indicando que no tiene nombre.

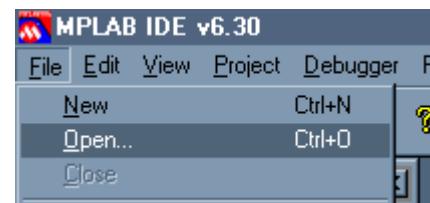


Cuando uno ve la pantalla en blanco no sabe por donde empezar, esto puede ser superado usando una plantilla. Existen plantillas para cada tipo de microcontroladores de la familia de Microchip.

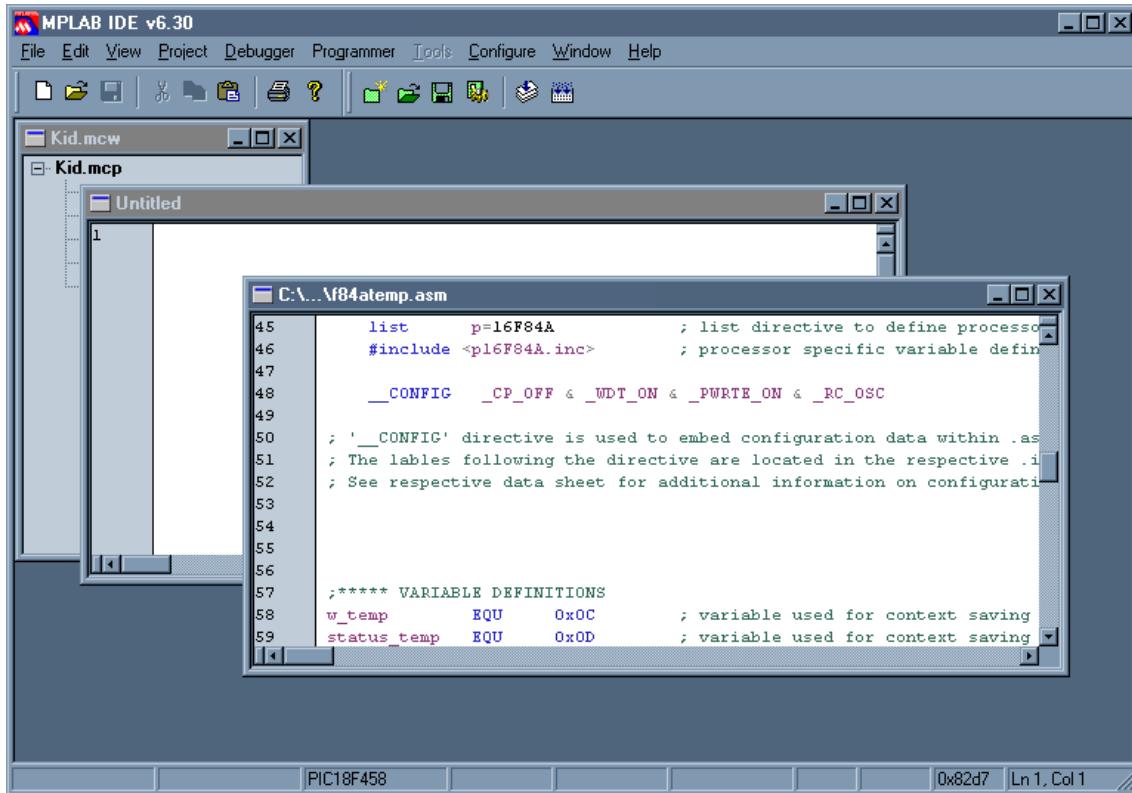
Como en este caso se va a trabajar con el PIC16F84A, se utilizará una plantilla para este integrado. Esta plantilla está identificada como **f84atempl.asm**, y está ubicado junto con las demás en: *C:\Archivos de programa\MPLAB IDE\WCHIP\_Tools\TEMPLATE\Code*

Para abrir el archivo, se elige la opción *Open* del menú *File*.

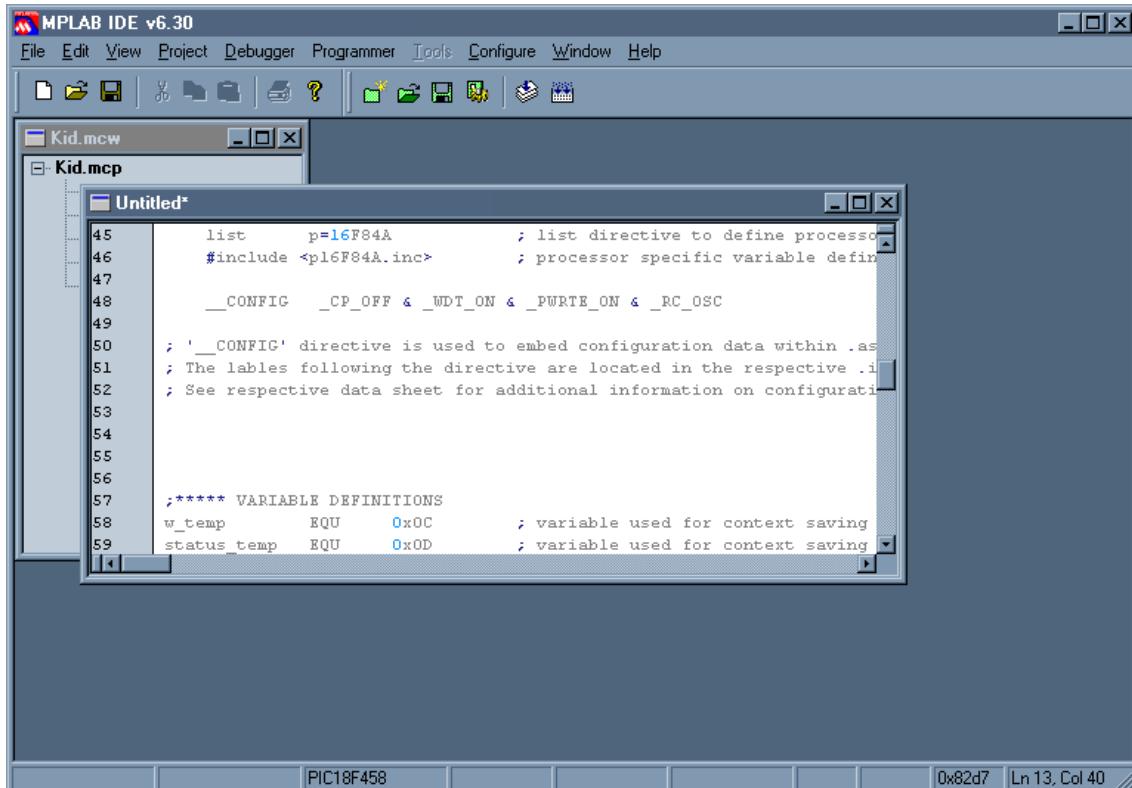
Se dirige a la ubicación ya indicada y se abre la plantilla con la que se quiera trabajar, en este caso es el **f84atempl.asm**



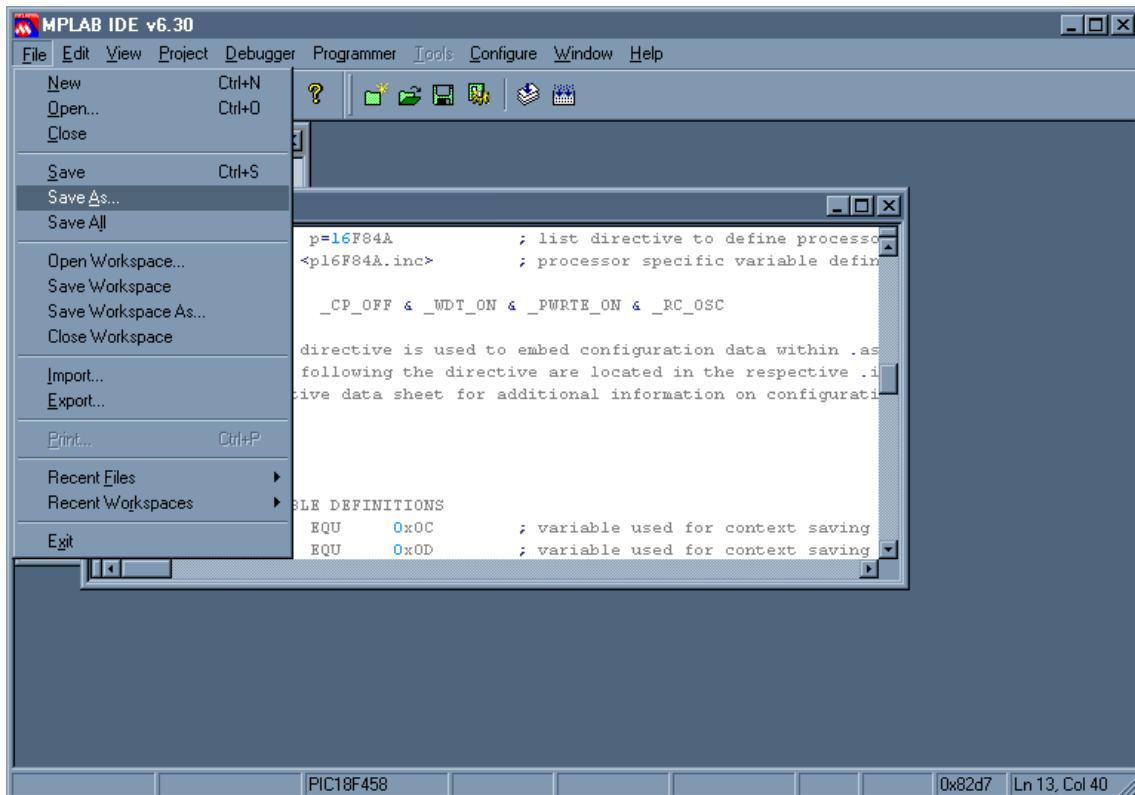
Se selecciona todo el contenido de la plantilla y se copia el contenido de la plantilla al nuevo archivo abierto (el que aparece como Untitled), este proceso se puede hacer con la ayuda de los comandos del menú *Edit*.



Una vez copiado el contenido a la ventana nueva, se cierra la plantilla **f84atemp.asm**.

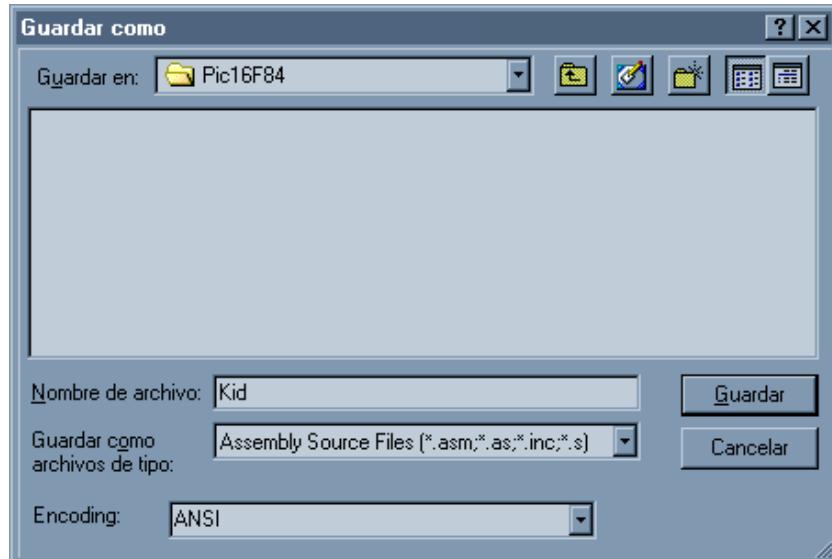


Una vez que ya se copió la estructura del programa desde la plantilla, se guarda el nuevo archivo de texto utilizando la opción *Save As* del menú *File*, como muestra la figura:

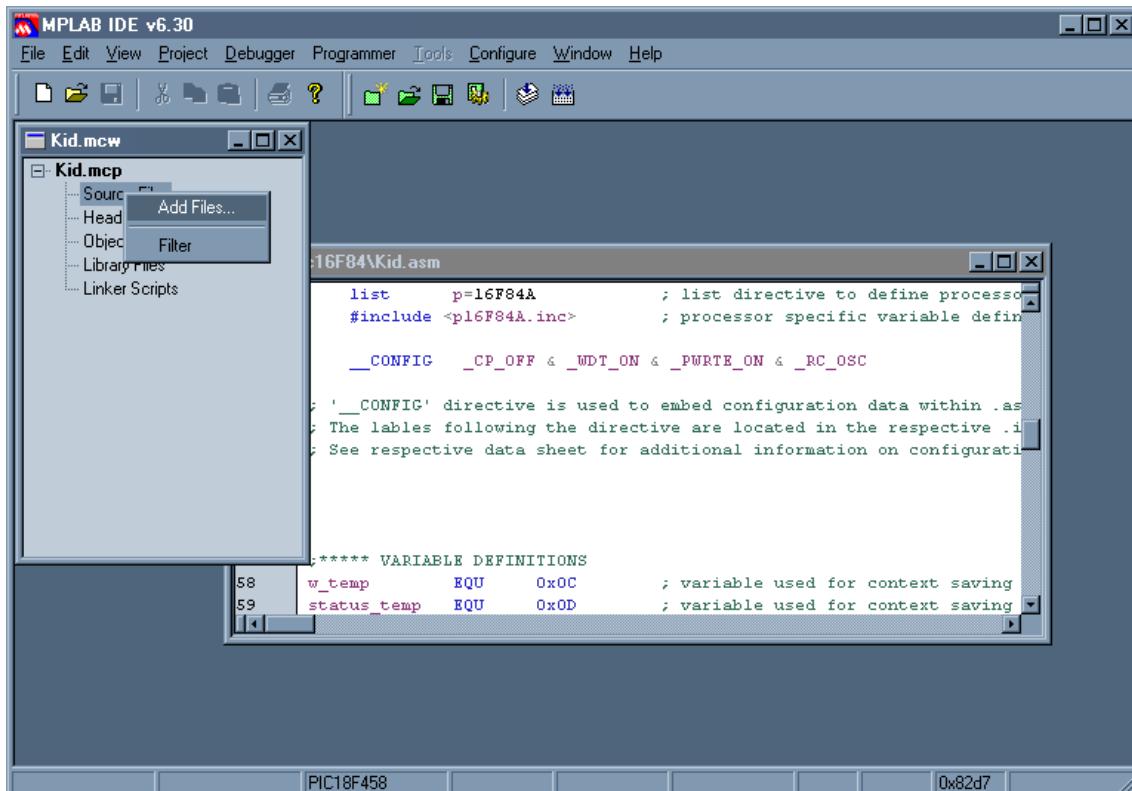


Aparecerá una pantalla en la que se elige la carpeta en donde se guardó el proyecto y se guarda el archivo de texto, preferiblemente con el mismo nombre del proyecto, aunque no es necesario.

Los archivos de texto en donde se programa siempre tienen la extensión *\*.asm*.

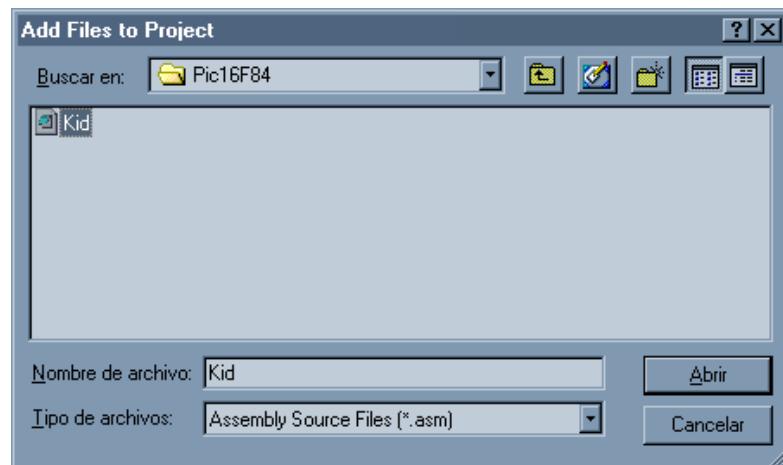


Se debe de agregar (enlazar) este nuevo archivo como parte del proyecto, para ello se hace clic con el botón derecho sobre *Source Files* de la ventana de *Kid.mcw* y eligiendo *Add Files*.



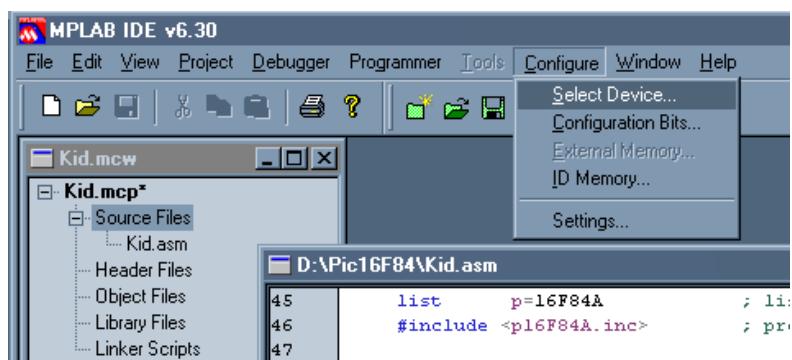
Se elige el archivo de texto que se acaba de crear.

Al hacer clic en el botón *Abrir*, y queda agregado.

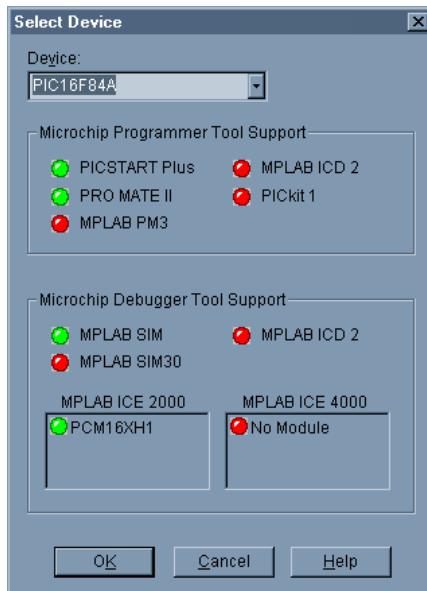


Para finalizar con la etapa de armar el proyecto de nuestro programa, es necesario indicar con qué tipo de dispositivo se va a trabajar.

Para esto, se elige la opción *Select Device* del menú *Configure*.



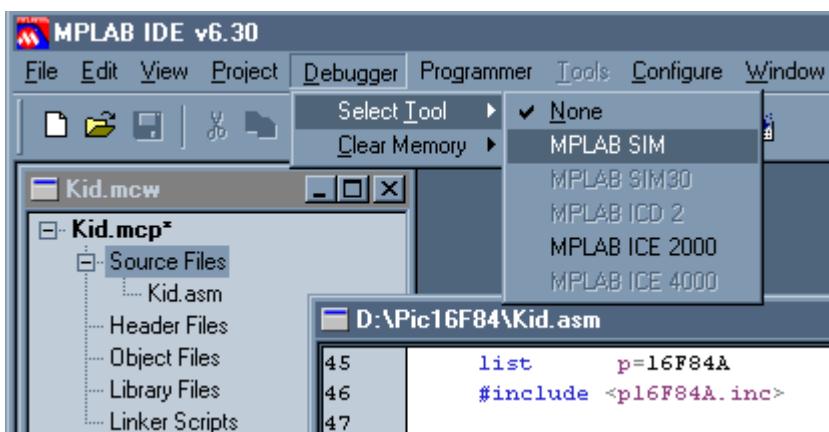
Aparecerá una pantalla como la que se muestra a la derecha, se elige el dispositivo, en este caso es el PIC16F84A, y luego se acepta con el botón OK.



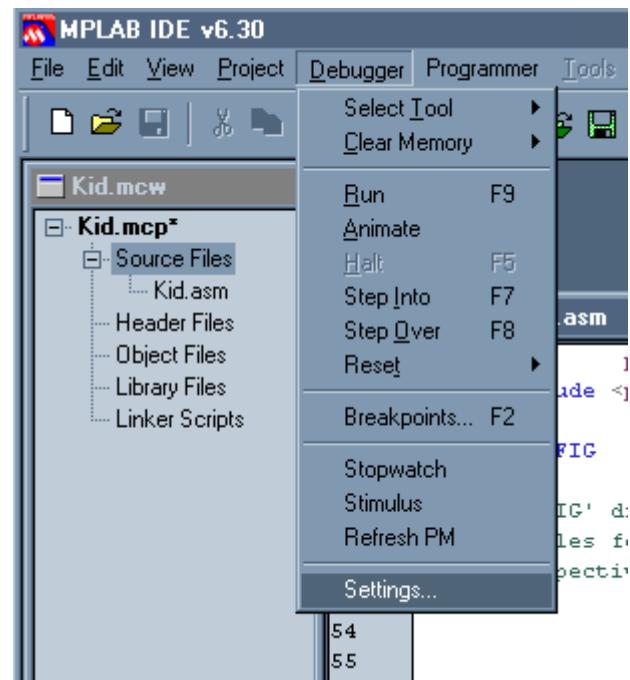
### Configuración del Simulador

Una manera de saber si el programa que se ha realizado funciona como se desea, es simulando su proceso en la computadora. Por esto Microchip facilitó algunos utilitarios como el MPLAB SIM que nos permitirá observar el comportamiento del programa.

Hay que habilitar la opción del simulador seleccionando la opción de MPLAB SIM en *Select Tool* del menú *Debugger*.



Se debe de asegurar que la frecuencia del cristal con la que se va a trabajar en el circuito de prueba sea la misma del simulador; para ello, se debe ir a la opción de *Settings* de *Debugger* y elegir la frecuencia adecuada.



Con esta configuración ya se puede empezar a trabajar en el programa que se quiere implementar. La figura muestra la zona en donde se deben escribir los comandos del programa. Una vez finalizado, se debe ejecutar el programa para verificar que no haya errores, esto se hace presionando la tecla F10. Aparecerá una lista con los errores y la línea dónde se produjo ésta. Si se hace doble clic en la línea donde indica el tipo de error el programa saltará automáticamente a la línea que debe ser corregida. Corregir los errores y volver a presionar F10 para verificar que no haya errores.

Cuando no se generen errores, aparecerá al final de todas la líneas **BUILD SUCCEEDED**, esto indica que no se encontraron errores de compilación y uno puede empezar a simular

```

68: ;*****
69:     ORG 0x000           ; processor reset vector
70:     goto main           ; go to beginning of program
71:
72:
73:     ORG 0x004           ; interrupt vector location
74:     movwf w_temp         ; save off current W register
75:     movf STATUS,w        ; move status register into W reg
76:     movwf status_temp     ; save off contents of STATUS reg
77:
78: ; isr code can go here or be located as a call subroutine elsewhere
79:
80:
81:     movf status_temp,w   ; retrieve copy of STATUS register
82:     movwf STATUS          ; restore pre-isr STATUS register
83:     swapf w_temp,f        ; restore pre-isr W register content
84:     swapf w_temp,w        ; restore pre-isr W register content
85:     retfie                ; return from interrupt
86:
87:
88:
89:
90: main
91:
92: ; remaining code goes here
93:
94:
95:
96:
97:
98:
99:
100:
101:
102:
103: END                   ; directive 'end of program'
104:
105:

```

En el menú de MPLAB se encuentra **Debugger**, esta lista de opciones para el simulador permite realizar lo siguiente:

**Run <F9>**: Ejecuta la simulación y se detiene cuando encuentra un *breakpoint*.

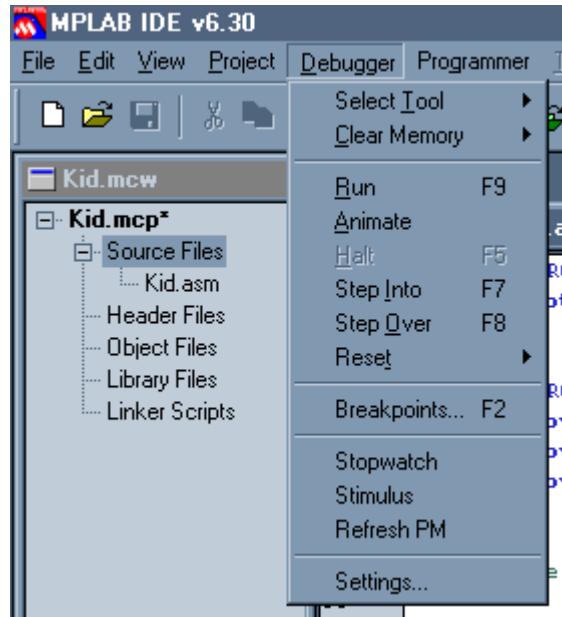
**Step Into <F7>**: Ejecuta línea a línea el programa.

**Step Over <F8>**: Ejecuta línea a línea el programa y ejecuta hasta finalizar la instrucción *call*.

**Breakpoints**: Permite realizar un punto de detención al programa cuando se ejecuta la tecla <F9>.

**Stopwatch**: Permite ver en una ventana el tiempo que se ejecuta cada comando o entre breakpoints.

**Stimulus**: Permite simular entradas digitales, cambios de estado y pulsos de los puertos.



Uno podrá apreciar mejor el comportamiento del dispositivo con la lista de opciones del menú **View**.

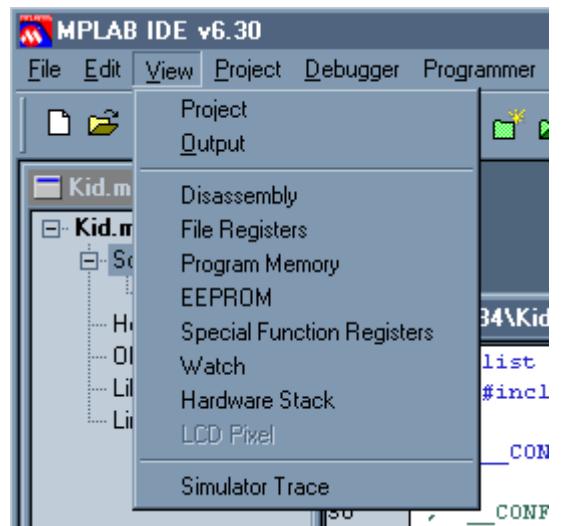
**Program Memory**: Muestra la memoria de programa y las instrucciones en su verdadera posición.

**EEPROM**: Muestra la memoria EEPROM en formato hexadecimal.

**Special Function Registers**: Muestra los registros SFR y su valor actual, los cuales cambian conforme el programa los modifique.

**Watch**: Permite agregar en una ventana los registros SFR o GPR que desean ser observados durante la simulación.

**Hardware Stack**: Permite observar los valores de las posiciones durante las llamadas *call* y las interrupciones



Si uno desea retomar el proyecto que se estaba realizando, debe ir a la opción de *Recent Workspaces* del menú *File* y elegir el proyecto.

