

Cómo Escribir Programas y Hacer la Verificación en los PICs

Introducción

MPLAB es un entorno de desarrollo integrado que le permite escribir y codificar los microcontroladores PIC de Microchip para ejecutarlos. El MPLAB incluye un editor de texto, funciones para el manejo de proyectos, un simulador interno y una variedad de herramientas que lo ayudarán a mantener y ejecutar su aplicación. También provee una interfase de usuario para todos los productos con lenguaje Microchip, programadores de dispositivos, sistemas emuladores y herramientas de tercer orden.

El MPLAB está diseñado para ser ejecutado bajo Windows 3.11, y puede operar con Windows 95 y 98. Asume que el usuario ya conoce el entorno de Windows y sabe manejarlo.

La guía que describimos en este capítulo le permitirá realizar las siguientes tareas:

- *Manejar el escritorio MPLAB*
- *Crear un nuevo archivo de código fuente para el ensamble e ingresarlo a un nuevo proyecto para el 16F84*
- *Identificar y corregir los errores simples*
- *Ejecutar el simulador interno*
- *Marcar puntos de interrupción*
- *Crear ventanas de observación*
- *Manejar ventanas para el seguimiento de errores*

Para que Ud. aprenda a programar sus microcontroladores, damos a continuación, paso a paso, las instrucciones de instalación y uso de la aplicación:

1) Instalación

Descargue los archivos del software de instalación y ejecute el archivo MPxxxx.E-
XE. Estos archivos puede obtenerlos por medio de Internet en la dirección:

<http://www.microchip.com/10/Tools>

Estos archivos pueden ser transferidos a disquetes si desea instalar el MPLAB en otra computadora. De acuerdo a la versión que haya descargado, los nombres de los archivos pueden variar levemente. Por ejemplo, la versión 4.00 del MPLAB tendría los siguientes archivos:

MP40000.EXE
MP40000.WO2
MP40000.WO3
MP40000.WO4
MP40000.WO5
MP40000.WO6

Si Ud. lo prefiere, puede venir a nuestras oficinas con este libro y 6 disquetes vírgenes y tendrá la oportunidad de llevarse dicho programa sin cargo.

Copie el contenido de los 6 disquetes en el disco rígido de su PC (en un lugar que pueda identificar).

Cuando ejecute el archivo .EXE, comenzará la instalación del MPLAB en su sistema. Seguidamente deberá elegir los componentes del MPLAB que desea instalar en su sistema. A menos que haya comprado un programador o emulador del dispositivo, sólo debiera instalar las siguientes

herramientas del software:

Archivos MPLAB IDE

Archivos MPASM/MPLINK/MPLIB

Archivos de Protección del Simulador MPLAB-SIM

Archivos de Ayuda (vea la figura 1 que aparecerá cuando ejecute el programa).

Luego aparecerá el menú de la figura 1 que le permitirá seleccionar los componentes de lenguaje Microchip que desee instalar. Usualmente debiera seleccionarlos todos (por defecto). Al hacer "doble click" en MP40000.EXE, el instalador le va diciendo lo que debe hacer.

Luego de instalarlos, ejecute MPLAB.EXE o clique el ícono MPLAB para iniciar el sistema.

Aparecerá el escritorio del MPLAB (figura 2).

2) Configurar el Modo de Desarrollo

El escritorio básico del MPLAB se asemeja al de las aplicaciones de Windows (como pudo ver en la pantalla de la figura 2). Tiene una barra de menú en el margen superior, una barra de herramientas y también una barra de estado en el margen inferior. Podrá advertir que la barra de estado incluye información sobre cómo se ha configurado el sistema.

Nota: El "modo de desarrollo" determina la herramienta, debe elegir alguna, que ejecutará el código. Para esta guía, usaremos el simulador de software *MPLAB-SIM*. Si sabe del tema y tiene un emulador, en este capítulo encontrará



Figura 1

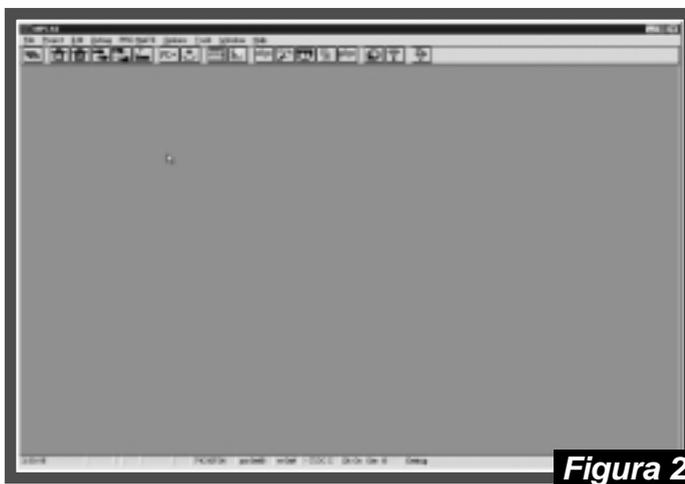


Figura 2

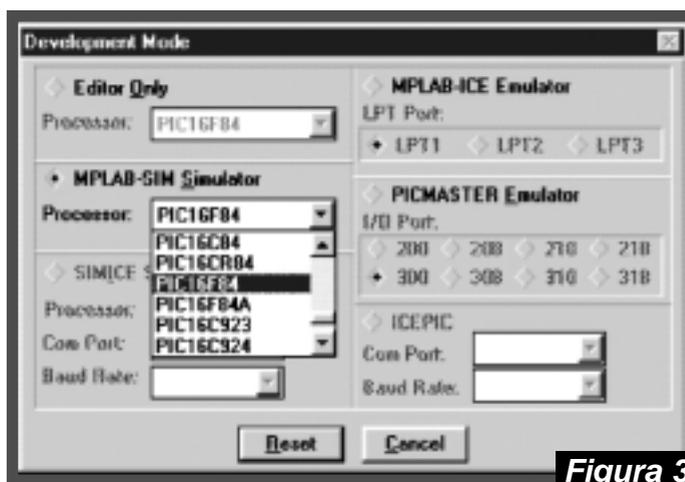


Figura 3

más información para cambiar a una de sus operaciones. Al seleccionar el ítem del menú "Opciones>Modo de Desarrollo",

aparecerá una caja de diálogo semejante a la de la figura 3.

El **MPLAB** es un producto en constante evolución, de modo que pueden aparecer sutiles diferencias entre la pantalla que usted vea y la que mostramos aquí. Seleccione el ícono próximo al Simulador MPLAB-SIM (MPLAB-SIM Simulator) y elija 16F84 (que corresponde a un tipo de PIC) en la lista de procesadores disponibles que pueden ser resistidos por el simulador. Clique 16F84 y luego presione el botón "**Reset**". De este modo se iniciará el simulador, y debería aparecer en la barra de estado "16F84" y "Sim". Se encuentra así en el modo simulador para el 16F84.

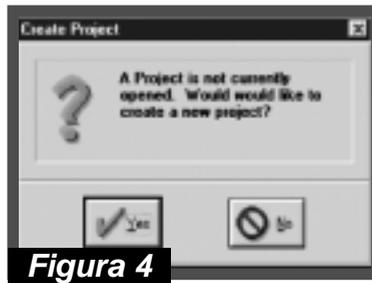


Figura 4

en el menú y aparecerá la caja de diálogo de la figura 4.

Clique en el botón **Sí**, seguidamente aparecerá un diálogo de exploración de Windows estándar. Decida dónde desea crear su proyecto (en qué carpeta o lugar de su disco rígido lo va a guardar) y recuerde dónde lo ubicó. Más tarde necesitará esta información. Esta guía usa un directorio en **c:**

\temp\tutorial y crea el archivo de proyecto llamado **tutor84.pjt** (figura 5). "**PJT**" es

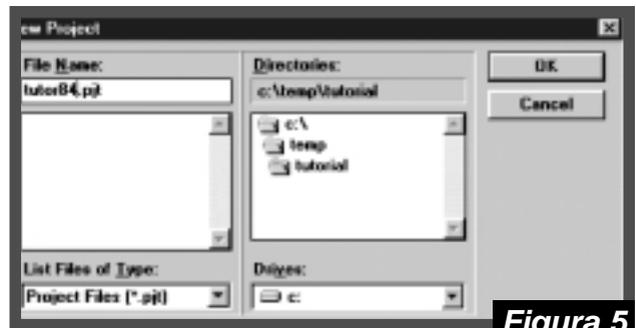


Figura 5

3) Cómo Crear un Proyecto Simple

El simulador se ejecutará desde el mismo archivo, llamado "**archivo hex**", el cual puede ser programado en el micro PIC. Para que se ejecute el simulador, primero deberá crear un archivo de código fuente y realizar el montaje del código fuente.

A continuación explicaremos cómo llevar a cabo este proceso:

Nota: El lenguaje ensamblador produce, entre otros elementos, un archivo **hex**. Este archivo tiene la extensión (**.hex**). A este archivo lo llamaremos:

tutor84.hex.

Más adelante este archivo puede ser cargado directamente en el programador del dispositivo sin usar el ensamblador o un proyecto del MPLAB. Este archivo también puede ser cargado por otros programadores de tercer orden.

Seleccione "**Archivo>Nuevo (File>New)**"

el sufijo estándar para los archivos de proyectos en el MPLAB. El prefijo del nombre de archivo del proyecto, en este caso "tutor84", será el prefijo por defecto de muchos de los archivos que el MPLAB usará o creará para esta guía. Clique "**Aceptar (OK)**" para que aparezca el diálogo Proyecto MPLAB. Este diálogo puede parecer confuso, pero en realidad es muy simple.

Nota: El simulador, los programadores y los sistemas emuladores que operan con el MPLAB usan un archivo **hex** creado por el ensamble, la compilación y/o el "linking" del código fuente. Algunas herramientas diferentes pueden crear archivos hex, al tener en cuenta que estas herramientas formarán parte de cada proyecto.

Los proyectos le dan la flexibilidad para describir cómo se construirá la aplicación y qué herramientas se usarán para crear el archivo **.hex**.

Obviamente, en la guía nos ocupare-



Figura 6

mos de todos estos detalles.

El diálogo "**Editar Proyecto**" será semejante al mostrado en la figura 6.

Advierta que el nombre del archivo de destino ya ha sido completado. Ya conoce el modo de desarrollo que configuramos previamente y asume que usaremos la serie de herramientas de lenguaje Microchip. En la ventana "**Archivos de Proyecto (Files project)**", encontrará **tutor84.[hex]**. Al destacar este nombre, se podrá utilizar el ícono "**Propiedades del Nodo (Node Properties)**". Seguidamente debe indicarle al MPLAB cómo crear el archivo hex. Hágalo clicando el botón "Propiedades del Nodo". Aparecerá el diálogo "Propiedades del Nodo". Este diálogo contiene todas las configuraciones por defecto para una herramienta de lenguaje -en este caso **MPASM**, como podrá ver en el ángulo superior derecho del diálogo. En su forma más simple, el proyecto contiene un archivo hex creado desde un archivo

fuente de ensamblado. Esta será la configuración por defecto cuando aparezca el diálogo "Propiedades del Nodo (Node Properties)", vea la figura 7.

Nota: Como puede ver, hay una cantidad de filas y columnas en este diálogo.

Cada fila usualmente corresponde a un "**cambio**", aquellos elementos que se establecen en la línea de comando cuando se invoca una herramienta. De hecho, la configuración de estos cambios se refleja en la ventana "**Línea de Comando (command line)**", próxima al margen inferior de la pantalla. Esta es la línea de comando que se usará cuando se invoque el MPASM desde el MPLAB. Por el momento, puede usar las configuraciones por defecto, pero cuando ya sepa construir una aplicación, probablemente deseará cambiar algunas.

Al clicar el botón "**Aceptar (OK)**", aplicará estas configuraciones, y retornará al diálogo "**Editar Proyecto (Edit Project)**", con el ícono (botón) "**Agrega Nodo (Add Node)**" disponible.

Presione el botón "**Agregar Nodo**". Aparecerá el diálogo de exploración de Windows estándar, con el mismo directorio usado para el proyecto. Ingrese el nombre de archivo: **tutor84.asm** y presione "**Aceptar**". Retornará al diálogo "**Editar Proyecto**",



Figura 7

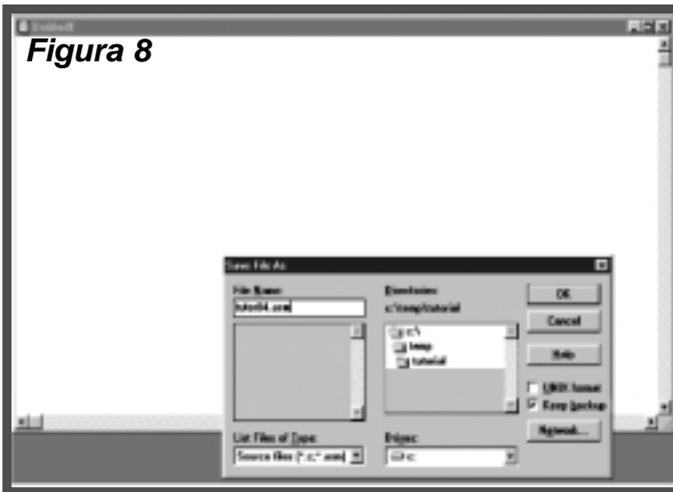


Figura 8

donde podrá ver **"tutor84.asm"** añadido debajo del archivo hex, que indica que es un nodo concurrente. Al presionar **"Aceptar"**, retornará al escritorio MPLAB con un archivo de código fuente abierto y aún sin nombre.

4) Cómo Crear un nuevo Archivo Fuente Simple

Clique dentro del espacio en blanco de la ventana de archivo creada. Seguramente se llamará **"Sin título (Untitled)"**. De este modo accederá al **"foco"** de la ventana. Use la opción de menú **"Archivo->Guardar como..."**, y guarde el archivo vacío como **tutor84.asm**. Cuando abra el diálogo de exploración estándar, encontrará su ubicación en el directorio del proyecto. Ingrese el nombre de archivo y presione **"Aceptar"**. Vea la figura 8.

Ahora estarán disponibles el escritorio MPLAB y la ventana de archivo va-

cio, pero el nombre de la ventana de archivo reflejará su nuevo nombre.

Nota: El nombre del archivo fuente y el nombre del proyecto ("tutor84" en esta guía) deben ser iguales en este tipo de proyectos. Hay otros proyectos de archivo múltiple que usan el "linker" y permiten que el nombre del archivo de salida sea diferente al del archivo de entrada (hay una guía aparte para los proyectos de archivos múltiples que usan el linker).

El **MPASM** siempre creará un archivo **hex** de salida con el mismo nombre que el archivo fuente, y esta configuración no puede modificarse. Si cambia el nombre del archivo fuente, también deberá cambiar el nombre del proyecto.

Ahora ya está listo para escribir el código que almacenará en el PIC para que cumpla una función determinada.

Tabla 1

list	p=16f84	
include	<p16F84.inc>	
c1 equ	h'0c'	; Establece el contador de variable temp c1 en la dirección 0x0c
org	h'00'	; Establece la base de memoria del programa en el vector reset 0x00
reset		
goto	start	; Ir a inicio en el programa principal
org	h'04'	; Establece la base de memoria del programa al comienzo del código del usuario
start		
movlw	h'09'	; Inicializa el contador a un valor arbitrario mayor que cero
movwf	c1	; Guarda el valor en la variable temp definida
loop		
incfsz	c1,F	; Incrementa el contador, ubica los resultados en el registro de archivos
goto	loop	; loop hasta que el contador se completa
goto	bug	; Cuando el contador se completa, va a start para reiniciar
end		

5) Ingresar el Código Fuente

Use el mouse para ubicar el cursor al comienzo de la ventana de archivo vacío **tutor84.asm**, e ingrese el texto de la tabla 1 (vea la página anterior), exactamente como está escrito en cada línea. No debe ingresar los comentarios (los textos que siguen a cada punto y coma).

Este código es un programa muy simple que incrementa un contador y lo "resetea" a un valor predeterminado cuando el contador vuelve a cero.

Nota: Todos los rótulos comienzan en la primera columna, y la última línea tiene una directiva "end". Las páginas de datos del micro PIC contienen información completa sobre instrucciones con ejemplos para su uso.

Guarde el archivo usando la función de menú **"Archivo>Guardar"** (**File>Save**).

6) Ensamble del Archivo Fuente

El ensamble del archivo puede realizarse de varias maneras. Aquí describiremos un método. Use el ítem de menú **"Proyecto>Construir todo (Project>Build All)"**. De este modo ejecutará el lenguaje ensamblador MPASM en el trasfondo usando las configuraciones guardadas con el proyecto anteriormente. Una vez completado el proceso de ensamble, aparecerá la ventana **"Resultados de Construcción (Build Results)"** (figura 9):

Ha ingresado intencionalmente al menos "un error" si ingresó el código tal como lo hemos escrito en el paso anterior. El último "goto" en el programa refiere a un rótulo inexistente llamado

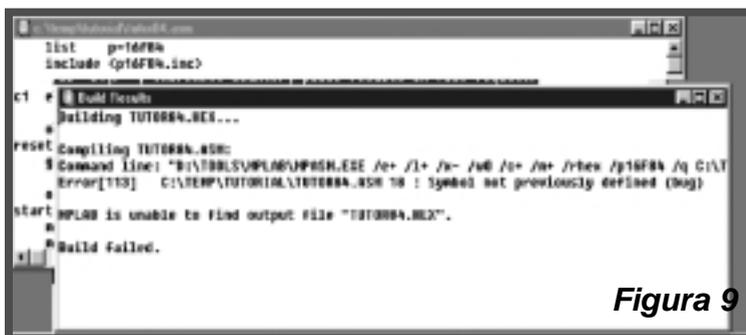
"bug". Dado que este rótulo no ha sido definido previamente, el lenguaje ensamblador informará el error. También podría relevar otros errores.

Haga un doble clic sobre el mensaje de error. De este modo ubicará el cursor en la línea que contiene el error en el código fuente. Cambie "bug" por "start". Use la ventana **"Resultados de construcción (Build Results)"** para hacer una búsqueda de errores, y reparar los que aparecieran en el código fuente. Reensamble el archivo ejecutando la función de menú **"Proyecto>Construir todo"**. Este procedimiento puede demandar un par de repeticiones.

Nota: Cuando reconstruya un proyecto, todos sus archivos fuente serán guardados en el disco.

Luego de reparar todos los problemas en el código fuente, la ventana "Resultados de construcción" mostrará el mensaje **"Construcción completada exitosamente (Build completed successfully)"**. Ya ha completado un proyecto que puede ejecutarse usando el simulador.

Como verá, aún no tiene ni idea que programó, pero ya programó algo que luego introducirá en un PIC. Para tener idea cabal de lo que está haciendo, continúe leyendo esta guía y con una PC al lado, haga Ud. mismo lo que se indica paso por paso.



```
list  pr16f84
include  pr16f84.inc

c1  # Build Results
    Building TUTOR84.HEX...

reset  Compiling TUTOR84.ASM:
       Command line: "C:\TOOLS\MPASM\MPASM.EXE /e+ /l+ /o- /w0 /o+ /m+ /rhex /p16f84 /q C:\T
       Error[113] C:\TEMP\tutorial\tutor84.asm 18 : symbol not previously defined (bug)

start  MPLAB is unable to find output file "TUTOR84.HEX".
Build failed.
```

Figura 9

7) Ejecución de su Programa

Use "**Debug>Ejecutar>Reset (Debug>Run>Reset)**" para iniciar el sistema. El contador del programa se reseteará a cero, que es el vector de reset en el 16F84. La línea del código fuente en esta dirección será destacada con una barra oscura. También advertirá que en la barra de estado, la PC se establecerá en 0x00.

Use el ítem de menú "**Debug>Ejecutar>Paso (Debug>Run>Step)**" (figura 10). Al hacerlo, el contador del programa avanzará hasta la siguiente ubicación de instrucción. La barra oscura seguirá el código fuente y el contador del programa desplegado en la barra de estado avanzará hasta "4".

Cuando ejecute el ítem de menú "Debug>Ejecutar>Paso", advierta la aparición de un texto en el lado derecho del ítem de menú que dirá "F7". El mismo equivale a "tecla de función siete" en su teclado. Muchas funciones del MPLAB se asignan a "teclas-especiales". Estas teclas cumplen la misma función que los ítems de menú a los cuales corresponden. Presione F7 varias veces y verá cómo el contador del programa y la barra avanzan a través del programa.

Ejecute el ítem de menú "**Debug>Ejecutar>Ejecutar (Debug>Run>Run)**" o presione F9 para iniciar la ejecución del programa desde la ubicación actual del contador. Los colores de la barra de estado cambiarán, para indicar que el programa está ejecutando las instrucciones. Ninguno de los campos de la barra de estado se actualizará mientras el programa esté en ejecución.

Detenga el programa ejecutando el ítem de menú "**Debug>Run>Halt (detener)**" o presionando F5. La barra de estado vol-

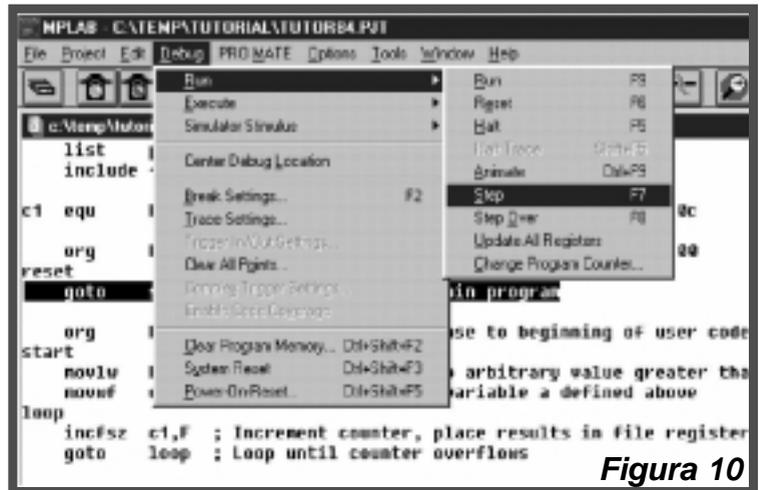


Figura 10

verá a su color original, y el contador del programa y otras informaciones de su estado serán actualizadas.

Nota: Otra manera de ejecutar funciones es usar la barra de herramientas ubicada en el margen superior de la pantalla. Si ubica el cursor sobre los ítems de la barra de herramientas, podrá ver el nombre de su función en la barra de estado. El botón de la izquierda es un botón estándar "**cambiar barra de herramientas (change tool bar)**" que le permite desplegar las barras de herramientas disponibles. Estas pueden ser personalizadas, como podrá advertirlo en la sección "Algunas Sugerencias" al final de esta guía. En la barra de herramientas de Debug, la luz verde es equivalente a F9 (Ejecución) y la luz roja equivale a F5 (Detención).

Hasta aquí, aprendió a instalar el programa y crear un archivo para "**aprender a programar**". Dimos un ejemplo de programación y ya sabemos cómo "**correr el programa en la PC**" para saber si todo está bien, antes de escribir dicho programa en nuestro PIC. Sin embargo, cuando los programas son más largos, es posible cometer errores, para evitarlos, sigamos leyendo este capítulo.

8) Abrir Otras Ventanas Para el Seguimiento de Errores

Hay muchas maneras de visualizar el programa y su ejecución usando el MPLAB. Por ejemplo, este programa está destinado a incrementar un contador temporario pero, ¿cómo puede asegurarse que se está produciendo dicho incremento? Una manera es abrir e inspeccionar una ventana de registro de archivo. Puede hacerlo ejecutando el ítem de menú "**Ventana>registros de archivo (Window>File Register)**". Aparecerá una pequeña ventana con todos los registros de archivo o el RAM del 16F84.

Presione F7 (ejecute instrucción por instrucción, si piensa anularlo) varias veces, y observe la actualización de valores en la ventana de registro de archivo. Hemos colocado la variable del contador en la ubicación de dirección 0x0C. Mientras el contador temporario se incrementa, este incremento se reflejará en la ventana de registro de archivo. Los registros de archivo cambian de color cuando su valor cambia, de modo que los cambios puedan advertirse fácilmente en la inspección. De todos modos, en muchos programas complejos, varios valores pueden cambiar, así resultará más difícil focalizar las variables que le interesan. Usando una ventana de

observación especial, este problema puede solucionarse.

Ahora, ya sabemos mejor qué es lo que estamos haciendo:

Programamos algo que incrementa en "1" cada vez que viene una señal" y podemos verificarlo virtualmente, antes de programar el PIC.

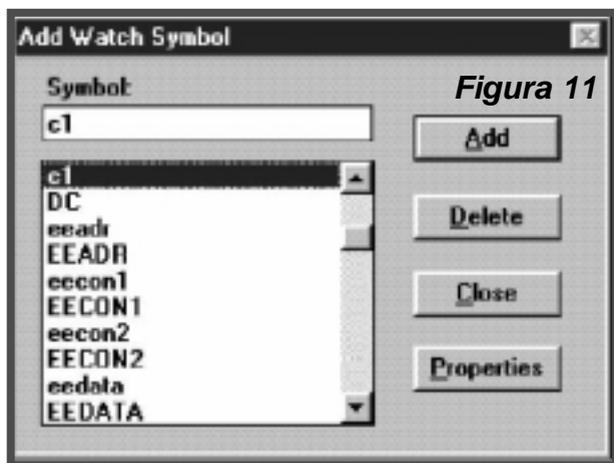
9) Creación de una Ventana de Observación

Ejecute el ítem de menú "**Ventana>Nueva ventana de observación (Window>New Watch Window)**". Aparecerá el diálogo "Agregar Símbolo de observación (Add Watch Symbol)" (figura 11).

Típee "c1" en la caja de nombre de símbolo para que la lista se despliegue hasta el símbolo deseado. Selecciónelo, presione el botón "**Agregar (Add)**", y luego el botón "**Cerrar (Close)**". Aparecerá en su escritorio MPLAB una ventana de observación que mostrará el valor actual del valor "c1" del contador temporario.

Presione F7 varias veces para advertir cómo se actualiza la ventana de observación mientras el valor del contador se incrementa. Si dejó la ventana de registro de archivo abierta, la misma también será actualizada (mueva una de ellas para que pueda ver las dos en la pantalla).

Puede guardar la ventana de observación y sus configuraciones al ejecutar el ítem "**Guardar observación (Save Watch)**" debajo del botón del sistema, ubicado en el ángulo superior izquierdo de la ventana de observación. Al clicar este botón, se desplegará un menú en cascada. Seleccione "Guardar observación" y aparecerá el diálogo de exploración estándar ubicado en el directorio del proyecto. Elija algún nombre arbitrario y presione "Aceptar (OK)".



Si no nombra la ventana de observación, el MPLAB lo hará por usted. La ubicación y el estado en la pantalla de la ventana abierta o cerrada serán guardados con el proyecto, de modo que la próxima vez que abra su proyecto, sus ventanas de observación aparecerán restauradas.

Nota: También puede editar ventanas de observación luego de crearlas. Use el botón del sistema y seleccione "**Agregar Observación (Add Watch)**" para que aparezca un diálogo mediante el cual podrá agregar más ítems. Con la tecla "**Ins**" podrá hacer lo mismo. Si desea borrar un ítem, selecciónelo y presione la tecla Suprimir (Delete); la observación referida desaparecerá de la ventana. Puede seleccionar "**Editar observación (Edit Watch)**" en el menú del sistema para cambiar el modo en el cual se muestra el ítem (en hex, binario, como una variable de 16-bit en vez de 8-bit, etc.).

10) Cómo se Marca un Punto de Interrupción

Presione F5 ("Debug>Ejecutar>Detener") para asegurarse que el procesador del simulador se ha detenido. Clique dentro de la ventana del código fuente la línea siguiente al rótulo "start", que dice "movlw 0'09". Presione el botón derecho del mouse para que aparezca el menú de la figura 12.

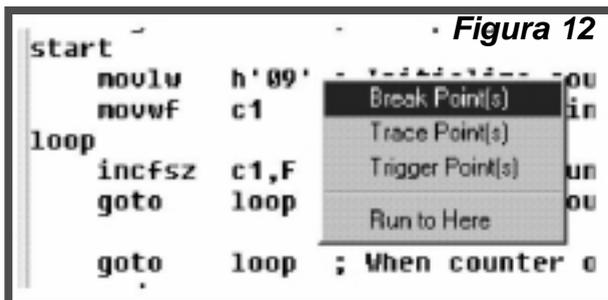


Figura 12

Clique el ítem de menú "**Punto(s) de interrupción (Break Point)**". El menú desaparecerá y la línea donde se ubicó el cursor cambiará de color, para indicar que ha sido establecido un punto de interrupción en dicha ubicación.

Presione F6 o ejecute el ítem de menú "**Debug>Ejecutar>Reset (Debug>Run>Reset)**" para resetear el sistema. Luego, presionando F9, ejecute el sistema. El programa se ejecutará y se detendrá en la instrucción ubicada luego del punto de interrupción. "c1", como aparece en la ventana de observación o en la de registro de archivo, si la tiene aún abierta, reflejará el estado reset de cero, el modo instrucción por instrucción ejecutará la carga y c1 luego reflejará un valor de 0x09. Presione F9 varias veces y advierta que la barra de estado cambia de color mientras el programa se esté ejecutando, y luego retorna a su color original cuando el procesador se detiene.

Resumen

Hasta aquí, en este capítulo Ud. ha aprendido:

- * **configurar un nuevo proyecto.**
- * **crear e ingresar un archivo fuente en un proyecto**
- * **ensamblar un código**
- * **ejecutar su código usando el simulador**
- * **marcar puntos de interrupción y ejecutar su código paso a paso (instrucción por instrucción)**
- * **observar variables en su código**

Algunas Sugerencias:

PUNTOS DE INTERRUPCION - Puede marcar puntos de interrupción en la ventana "Ventanas>Memoria de programa (Window>Program Memory)", en la ventana de ar-

chivo fuente (en este caso tutor84.asm), o en la ventana "Ventanas>Listado Absoluto (Windows>Absolute)".

ARCHIVOS FUENTE - Use "Ventana>Ventana de Proyecto (Window>Project Window)" para que aparezca una lista de sus archivos fuente. Puede hacer un doble clic sobre el nombre de un archivo para trasladarlo al editor.

ERRORES MPASM - Si el MPASM le da un error, haga un doble clic sobre el error en la ventana de error para ir al error en el código fuente. Si tiene múltiples errores, siempre elija el primer error —generalmente un error causa errores subsiguientes y al reparar el primero corregirá los restantes.

CONFIGURACIÓN DE BITS Y EL MODO DEL PROCESADOR - La configuración de bits en el archivo fuente no determinará el modo del procesador para el simulador (o los emuladores). Use "Opciones>Configurar Procesador>Hardware (Options>Processor Setup>Hardware)" para estas configuraciones. Aún cuando puede establecer estos bits en el archivo fuente del MPASM o del MPLAB-C17, el MPLAB no cambia automáticamente los modos. Por ejemplo, la configuración de bit Activar Observación de Dog Timer puede hacerse de tal modo que, cuando programe un dispositivo, el Dog Timer sea activado. En el MPLAB también necesitará acceder al diálogo "Opciones>Configurar Procesador>Hardware" para activar el WDT para el simulador o el emulador. Esto le permitirá hacer un seguimiento de errores con el WDT activado o desactivado sin cambiar su código fuente.

OPCIONES - Use "Opciones>Configurar Entorno (Options>Environment Setup)" para hacer lo siguiente:

- * Establecer teclas de Mapa Europeo

para funciones MPLAB y caracteres ASCII especiales

- * Cambiar la fuente de la pantalla o el tamaño de la fuente
- * Posicionar la barra de herramientas a un lado o al pie de la pantalla
- * Modificar la barra de herramientas
- * Cambiar la cantidad de caracteres desplegados para los rótulos.

ARCHIVOS MAPA - Use el diálogo "Proyecto>Editar Proyecto" y cambie las Propiedades del Nodo del MPASM para producir un archivo MAP llamado tutor84.map. Luego de construir el proyecto, busque el tutor84.map para ver información de la construcción.

MENUS DESACTIVADOS - Si encuentra menús desactivados (opacados), verifique que no haya ingresado el modo "sólo Editor (Editor Only)". Si está seguro de que ha realizado correctamente la configuración, intente salir del MPLAB y reinicie el programa.

Características del MPLAB no Abarcados en este capítulo

Hay muchas características del MPLAB que no se tratan en esta guía. En lugar de describir cada característica, generando una guía extensa y complicada, primero hemos decidido darle el conocimiento básico para operar el MPLAB. Le ofrecemos a continuación un punto de inicio para ayudarlo a explorar otras herramientas del MPLAB.

Otras Ventanas de Datos y Diálogos

Hasta aquí, los diferentes "comandos" u "órdenes" del MPLAB, las hemos dado en

castellano y en inglés, dado que el programa está en inglés, pero para que sepa lo que está haciendo, decidimos colocar su traducción. De aquí en más, sólo colocaremos la traducción, dado que el lector sabrá a qué orden nos referimos.

Esta guía ya le ha presentado la Ventana de Memoria de Programa y las Ventanas de Observación. Aún no ha explorado las otras ventanas. Debiera buscar en "Ventana>Conjunto" y "Ventana>Registros de Función Especial" para ver cómo el MPLAB puede mostrar otras áreas de memoria.

Hay otros diálogos, como el diálogo "Debug>Configuración de puntos de interrupción" para marcar puntos de interrupción.

La ayuda online y la Guía del Usuario de MPLAB le explicarán el uso de estas características.

Rastreo

La ventana de rastreo le brinda una imagen "instantánea" de la ejecución de su programa. Los emuladores que tienen un amortiguador de rastreo, pueden mostrar cómo se ejecuta su programa a altas velocidades.

Nota: Algunas aplicaciones, como los sistemas de control de motor, no pueden ser detenidos. Algunos errores sólo aparecen cuando la aplicación está en ejecución, no ocurren cuando se ejecuta el código instrucción por instrucción. El amortiguador de rastreo le brinda otra herramienta para probar este tipo de aplicaciones.

En el simulador, el amortiguador de rastreo es útil para recoger un extenso registro de la ejecución del programa, de modo que pueda retornar al mismo y analizarlo cuidadosamente. El simulador mos-



Figura 13

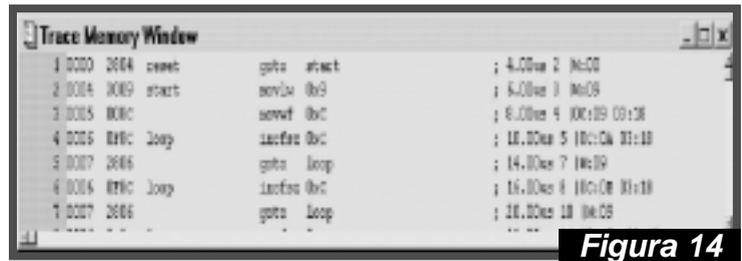


Figura 14

trará una información algo diferente de la registrada por el rastreo del emulador.

Para usar el amortiguador de rastreo del simulador, primero debe seleccionar un código a rastrear. Si clicla y arrastra toda la ventana de memoria del programa, puede seleccionar instrucciones para rastrear. Presione luego el botón derecho del mouse para que aparezca un diálogo en el cual podrá seleccionar "Rastrear Punto(s)".

Ahora resetee y ejecute el código, luego deténgalo después de haberse ejecutado durante algunos segundos. Seleccione "Ventana>Rastreo" para ver los resultados del rastreo (figura 13).

El simulador coloca una marca de tiempo en cada línea y también muestra todos los registros que cambiaron junto a sus valores (figura 14).

El estímulo genera señales para el simulador. Puede establecer pines (pasadores) altos o bajos, e introducir valores directamente en los registros. Hay cuatro modos de estímulo:

- * Estímulo asincrónico - Un diálogo interactivo para controlar señales en los pines de entrada

- * Estímulo de Archivo Pin - Los conteni-

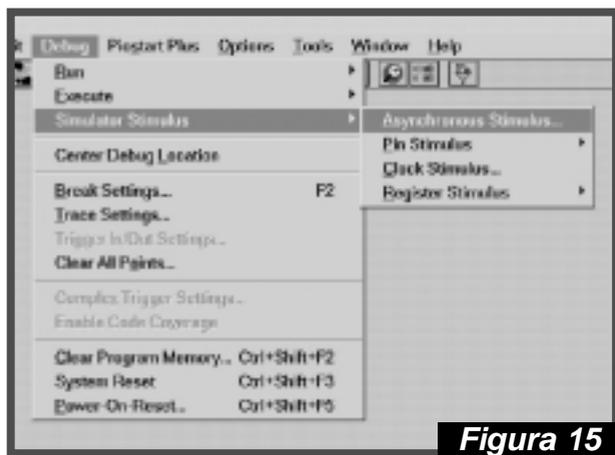


Figura 15

los pines.

Por ejemplo, hemos configurado una señal que cambiará de nivel en un pin I/O sobre el puerto b del 16F84.

Seleccione "Debug>Estímulo de Simulador>Estímulo Asíncronico..."

Vea la figura 15. Se desplegará el diálogo de la figura 16.

Ahora coloque el cursor sobre el botón rotulado "Stim1 (P)" y clique el botón derecho del mouse para que aparezca un diálogo en el cual debe seleccionar "Toggle" (figura 17).

Coloque otra vez el cursor sobre el botón rotulado ahora "Stim1 (T)" (la "P" fue reemplazada por una "T", que significa "Toggle"), presione el botón derecho del mouse, y seleccione "Asignar Pin..." (figura 18).

Aparecerá otro diálogo con una lista de los pines conectados al 16F84 (figura 19).

Coloque el cursor sobre el "RB0" y haga un doble clic.

Debería aparecer el diálogo de la figura 20, de Estímulo Asíncronico.

Advierta que el botón ahora aparece como "RB0 (T)".

Elija "Debug>Ejecutar>Animado" para que el procesador se ejecute en el modo "paso a paso rápido". La barra de estado alternará la ejecución con una detención muy rápida.

Presione el botón "RB0 (T)" en el diálogo Estímulo Asíncronico. Debería ver el cambio de valor del puerto b en la ventana Registro de Función Especial mientras repetidamente clique el botón para simular una señal alta y luego una baja aplicadas al pin 0 del puerto b.

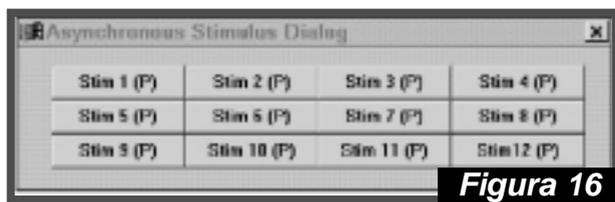


Figura 16

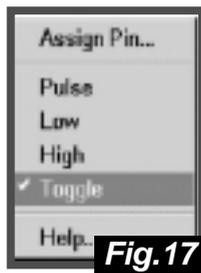


Fig.17

dos de un archivo de texto que describe señales a los pines de entrada

* Estímulo de Archivo de Registro - Los contenidos de un archivo de texto usado para configurar valores de 8-bit directamente en un registro

* Estímulo de reloj - Una fuente de pulsos de estímulo periódica, regular y programable.

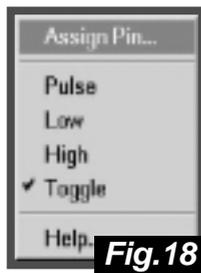


Fig.18

Diálogo de Estímulo Asíncronico

Esta característica de estímulo provee un botón de diálogo para simular volts +5 y 0 aplicados a pines de entrada. Mientras su programa se ejecuta con el simulador, puede presionar los botones de este diálogo para cambiar los niveles de

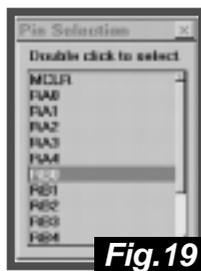


Fig.19



Fig.20

Archivos de "Estímulo Pin"

Un archivo de estímulo pin consiste en columnas de unos y ceros de ingreso que serán aplicadas a los pines cuando el valor "Ciclo" en el Reloj de detención alcance al de la columna CICLO.

Use "Archivo>Nuevo archivo" y tipee el texto de la tabla 2. No debe tipear el texto que sigue a los ";" y "!", que se limita a comentarios.

Use "Archivo>Guardar como..." para guardar como **tu-tor84.sti**.

Nota: Para mantener una compatibilidad retroactiva con versiones anteriores del MPLAB, la primera línea siempre debe comenzar con las palabras "CICLO" o "STEP". Esta columna especifica el CICLO (como lo determina la Ventana del Reloj de Detención del MPLAB) donde los valores de las otras columnas serán aplicados.

Luego de la palabra "CICLO" en la primera línea del archivo se colocan los nombres de los pines del microPIC que recibirán los valores de estímulo altos y bajos. En este ejemplo los pines RB1 y RB0, por dos entradas del port (puerto) B, recibirán entradas de estímulo. En este archivo, la segunda columna contiene valores que serán aplicados a RB1 (puerto b bit 1) y la tercera columna presenta los valores para RB0 (puerto b bit 0). Estos

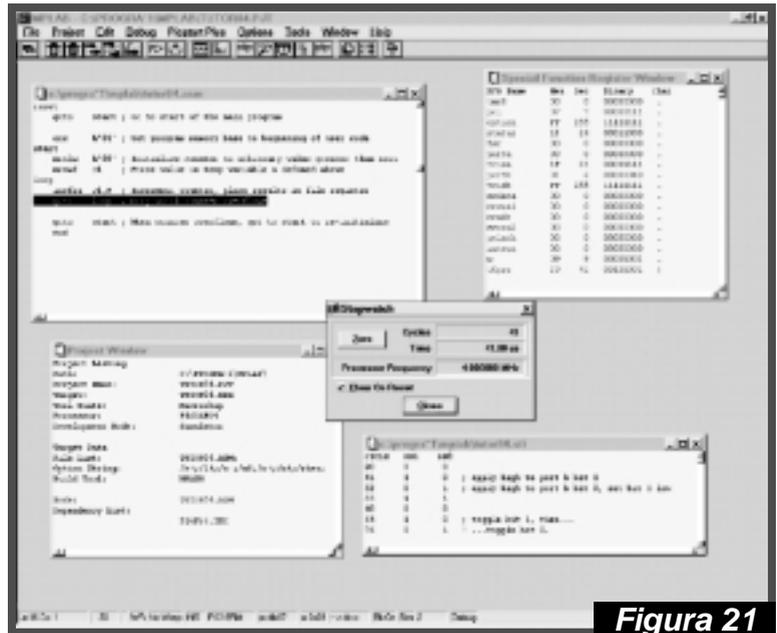


Figura 21

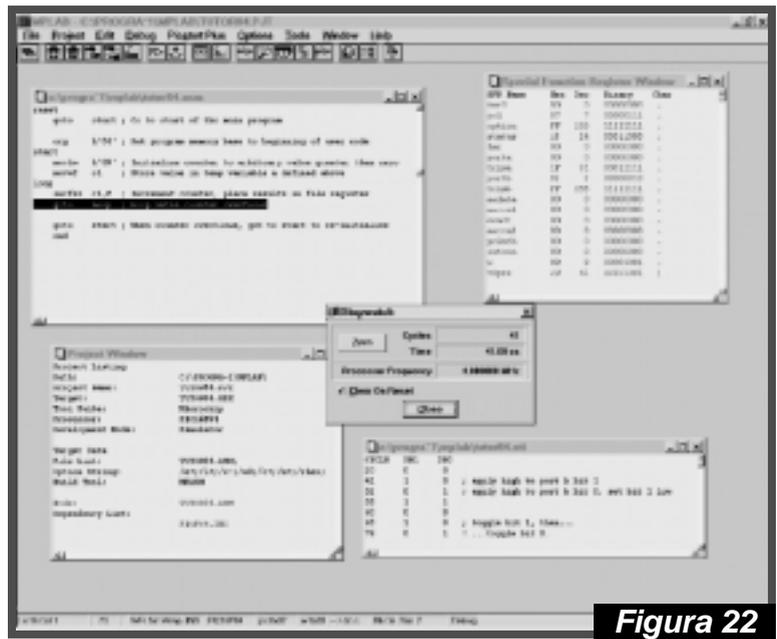


Figura 22

Tabla 2

CICLO	RB1	RB0	
20	0	0	
41	1	0	; aplicar alta a puerto b bit 1
52	0	1	; aplicar alta a puerto b bit 0, establecer bit 1 bajo
55	1	1	
60	0	0	
65	1	0	; toggle bit 1, luego...
76	0	1	! ...toggle bit 0.

nombres deben ser compatibles con los nombres de pin del microPIC de Microchip para el procesador que está siendo simulado.

Nota: Puede ver todos los pines de soporte en la lista de asignación de pines para el Estímulo Asíncrono (clicke el botón derecho del mouse sobre un botón de estímulo y se desplegará la pantalla de la figura 21). Se pueden hacer comentarios en una línea usando los caracteres ";" o "!", precedidos y seguidos, como mínimo, por un espacio. La ventana del Reloj de Detención también mostrará el tiempo transcurrido a cada instrucción, determinado desde el valor CICLO y la frecuencia del reloj. Si el Reloj de Detención se resetea a 0, el archivo de estímulo pin también será efectivamente reseteado.

Abra la ventana del Reloj de Detención y seleccione "Ventana>Reloj de Detención". También seleccione "Ventana>Registros de Función Especial".

Estaremos interesados en observar el "puerto b". También se puede agregar el puerto b a una ventana de observación (figura 22).

Reseteo y ejecute instrucción por instrucción hasta ejecutar 41 ciclos. Luego verá que el "puerto b" cambió su valor al asignado en la segunda línea del archivo de estímulo.

Archivos de Estímulo de Registro

Un archivo de estímulo de registro consiste en una columna de los valores que serán enviados a un registro cuando la dirección de memoria del programa alcance la ubicación esta-

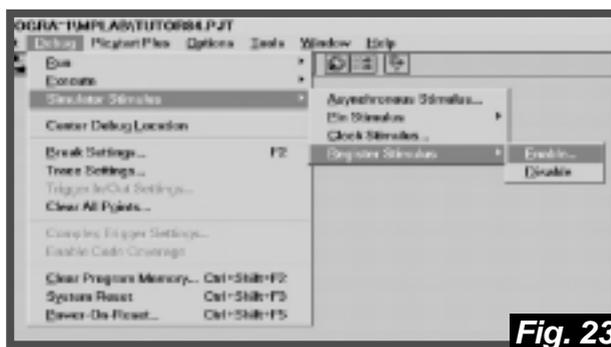


Fig. 23

blecida en el Diálogo Estímulo de Registro. Esto es útil para simular una operación A/D. Abra un nuevo archivo usando "Archivo>Nuevo Archivo" y tipee la siguiente lista de números:

10
2E
38
41
50
7A
99
A0
FD

Guárdela usando "Archivo>Guardar como..." y nómbrala tutor84.reg.

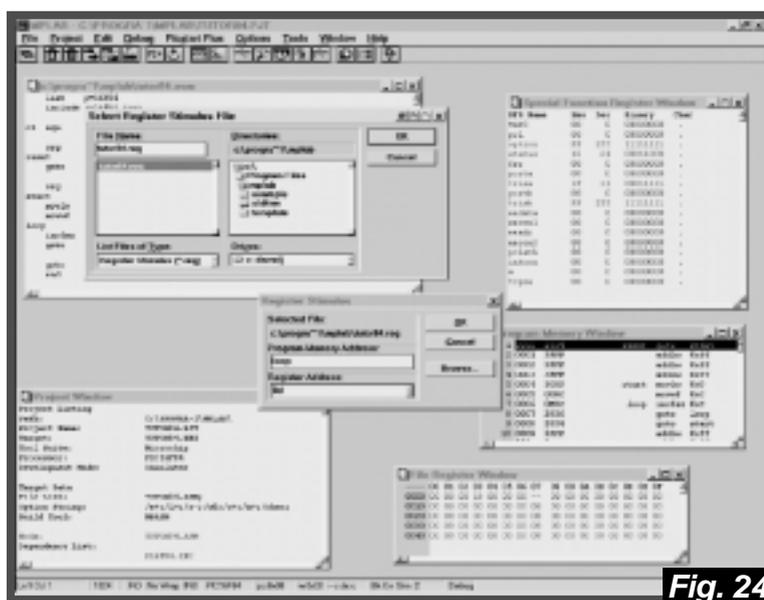


Fig. 24

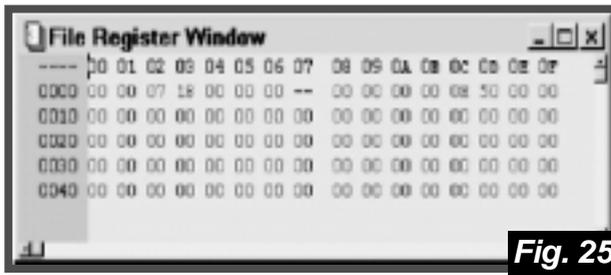


Fig. 25

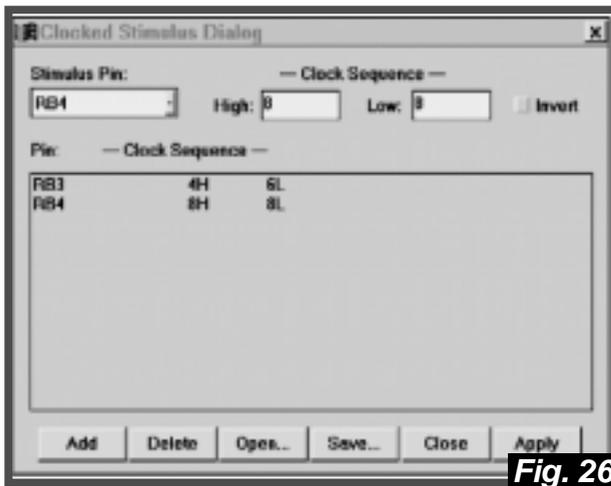


Fig. 26

Este archivo será usado para introducir secuencialmente estos valores en un registro. Seleccione "Estímulo de Simulador>Estímulo de Registro>Activar..." (figura 23)

Luego configure "loop" para que se ubique en el programa cuando se introduzcan los valores, y para propósitos de demostración, introdúzcalos en el registro de archivo en la dirección 0x0d. Luego de configurar "loop" y "0d" en sus cajas correspondientes, presione "Explorar" para que aparezca el diálogo de archivo, en el cual podrá seleccionar tutor84.reg como el archivo de estímulo de registro (figura 24).

Abra la ventana "Ventana>Registros de archivo" para ver el efecto de este estímulo, aparecerá el listado de la figura 25.

Reseteo y luego ejecute instrucción por instrucción el procesador. Cada vez que llegue a 'loop', el valor en el registro de archivo en la dirección 0x0D cambiará. La lista de valores en tutor84.reg será secuencialmente introducida en el registro de ar-

chivo seleccionado (0x0D, con un valor de 0x50 como se muestra en la figura 25). Los valores 0x10, 0x2E, etc. serán introducidos en el registro seleccionado en el diálogo "Debug>Estímulo de Simulador>Registro de Estímulo", cada vez que se ejecute "loop".

Luego de introducido el último valor (0xFD en tutor84.reg), el primer valor será usado nuevamente (0x10). La lista será cíclica mientras se ejecute el MPLAB-SIM.

Estímulo de Reloj

El estímulo de reloj genera una onda regular sobre un pin con un ciclo obligatorio especificado en términos de los ciclos de reloj del procesador (figura 26).

Si abre el diálogo "Debug>Estímulo del Simulador>Estímulo de Reloj...", podrá ingresar varios estímulos de reloj como se muestra en la figura 26. Cuando ejecute el programa usando las configuraciones establecidas en la figura, RB3 será Alta para 4 ciclos de reloj, y luego Baja para 6 ciclos de reloj. RB4 será Alta para 8 ciclos de reloj y luego Baja para 8 ciclos de reloj. Ambos valores se repetirán hasta que salga del MPLAB o los elimine usando este diálogo.

Interrupción Condicional

El diálogo punto de interrupción complejo es útil para configurar un punto de interrupción en una condición compleja y especial.

Cómo Cargar el PIC

Una vez que estamos seguros que nuestro programa funciona de acuerdo con el test realizado mediante el empleo de nuestro programa, estamos en condicio-

nes de "grabar nuestro PIC", para lo cual debemos seguir las instrucciones dadas en el capítulo 2.

Cabe aclarar que el programa MPLAB tiene, en su barra de herramientas, un ícono de ayuda (HELP) que explica paso por paso todo lo que se puede hacer con el programa. Para muchos, el problema reside en que los textos de ayuda están en inglés. Para que tenga una idea de la información que contienen dichos archivos de ayuda, nosotros hemos traducido unas 450 páginas que paulatinamente iremos publicando a modo de guías para diferentes funciones.

Veamos entonces, algunos de los productos que podremos consultar:

Características del Editor

El editor de archivo del programador con el MPLAB tiene un conjunto de características para la escritura y edición del código fuente.

Reloj de Detención

El Reloj de Detención le permite medir el tiempo de ejecución del código. No siempre es preciso en la ejecución instrucción por instrucción. El reloj de detención calcula el tiempo en base a la frecuencia del reloj del microPIC. Configure la frecuencia del reloj en el diálogo "Opciones>Configurar Procesador>Frecuencia de Reloj...".

Proyectos de Archivo Múltiple

Usando el linker, puede tener dos o más archivos en un proyecto. Vea la Guía del Proyecto v3.40 de MPLAB para más información.

MPASM

El MPASM, junto con la guía del Usuario de MPLINK & MPLIB describe la operación

del lenguaje ensamblador, el linker y el bibliotecario, que pueden ser descargados de la página web de Microchip. También está disponible una guía del linker en las direcciones de Internet que fuimos especificando.

PICMASTER, ICEPIC, MPLAB-ICE

Estas son Guías de Usuario separadas para la operación de emuladores que pueden descargarse de la página web de Microchip.

PICSTART Plus y PRO MATE

Estos son programadores de dispositivo de Microchip. Vea la Guía del Usuario MPLAB PRO MATE y la del PICSTART Plus. Ambos manuales pueden descargarse de la página web de Microchip.

MPLAB-C17

El compilador 17cxxx opera con el MPLINK y el Administrador de Proyecto del MPLAB. Vea la Guía del Usuario de MPLAB-C17 y la Guía del Proyecto v3.40 del MPLAB.

Herramientas de Tercer Orden

El PIC C de alta tecnología, el compilador CCS y el PIC BASIC de MicroIngeniería pueden ser usados con el MPLAB. Vea sus documentaciones respectivas para obtener más información sobre su uso con el MPLAB.

Problemas

Vaya a las conferencias sobre el microPIC y Sistemas de Desarrollo en www.microchip.com para hacer preguntas que pueden ser útiles para principiantes o expertos. Los ingenieros de Microchip Application, experimentados usuarios del microPIC, y los ingenieros de Sistemas de Desa-

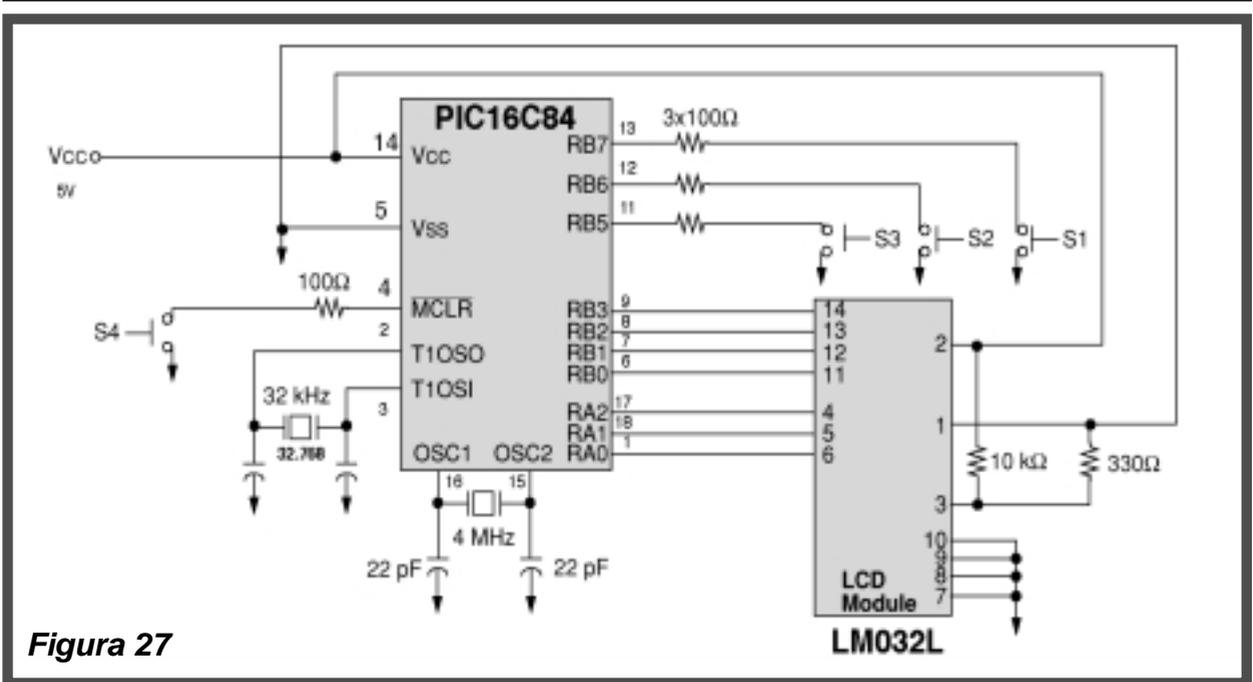


Figura 27

```

nolist
;*****
;
; This is the custom Header File for the real time clock application note
; PROGRAM: CLOCK.H
; Revision:5-10-94
;
;*****
; This is used for the ASSEMBLER to recalculate certain frequency
; dependant variables. The value of Dev_Freq must be changed to
; reflect the frequency that the device actually operates at.
;
Dev_Freq      EQU    D'4000000'      ; Device Frequency is 4 MHz
DB_HI_BYTE    EQU    (HIGH ((( Dev_Freq / 4 ) * 1 / D'1000' ) / 3 )) + 1
LCD_INIT_DELAY EQU    (HIGH ((( Dev_Freq / 4 ) * D'46' / D'10000' ) / 3 )) + 1
INNER_CNTR    EQU    40      RAM Location
OUTER_CNTR    EQU    41      ; RAM Location
;
T1OSO         EQU    0      ; The RC0 / T1OSO / T1CKI
;
RESET_V       EQU    0x0000    ; Address of RESET Vector
ISR_V         EQU    0x0004    ; Address of Interrupt Vector
PMEM_END      EQU    0x07FF    ; Last address in Program Memory
TABLE_ADDR    EQU    0x0400    ; Address where to start Tables
;
HR_MIN_SW     EQU    0x7      ; The switch to select the units
INC_SW        EQU    0x6      ; The switch to increment the selected units
CLR_MIN_SW    EQU    0x5      ; The switch to clear the minutes and seconds
;
FLAG_REG      EQU    0x020    ; Register which contains flag bits
;
; +-----+-----+-----+-----+-----+-----+-----+-----+
; | AM | --- | --- | KEY_INPUT | --- | --- | MIN_UNIT | HR_UNIT |
; +-----+-----+-----+-----+-----+-----+-----+-----+
    
```

Figura 28

Figura 28 (continuación)

```

;
AM          EQU 0x07    ; Flag to specify if AM or PM
;
KEY_INPUT   EQU 0x04    ; Flag to specify if doing key inputs
;
MIN_UNIT    EQU 0x01    ; Flags to specify which units to operate on
HR_UNIT     EQU 0x00    ; (HRS, MIN, or none)
;
HRS         EQU 0x030   ; Holds counter value for HOURS
MIN         EQU 0x031   ; Holds counter value for MINUTES
SECS       EQU 0x032   ; Holds counter value for SECONDS
MSD        EQU 0x033   ; Temporary register, Holds MSD of BIN to BCD conversion
LSD        EQU 0x034   ; Temporary register, Holds LSD of BIN to BCD conversion
TEMP       EQU 0x035   ; Temporary register
CHAR       EQU 0x036   ; Temporary register, Holds value to send to LCD module
;
WAIT_CNTR   EQU 0x040   ; Counter that holds wait time for key inputs
;
;
; LCD Display Commands and Control Signal names.
;
E           EQU 0       ; LCD Enable control line
R_W        EQU 1       ; LCD Read/Write control line
RS         EQU 2       ; LCD Register Select control line
;
;
; LCD Module commands
;
DISP_ON     EQU 0x00C   ; Display on
DISP_ON_C   EQU 0x00E   ; Display on, Cursor on
DISP_ON_B   EQU 0x00F   ; Display on, Cursor on, Blink cursor
DISP_OFF    EQU 0x008   ; Display off
CLR_DISP    EQU 0x001   ; Clear the Display
ENTRY_INC   EQU 0x006   ;
ENTRY_INC_S EQU 0x007   ;
ENTRY_DEC   EQU 0x004   ;
ENTRY_DEC_S EQU 0x005   ;
DD_RAM_ADDR EQU 0x080   ; Least Significant 7-bit are for address
DD_RAM_UL   EQU 0x080   ; Upper Left corner of the Display
;

```

rollo participan en estas conferencias.

Este es el modo más rápido para vencer los obstáculos que puedan surgir, informar sobre problemas o simplemente revisar las preguntas ya registradas de otros usuarios a fin de obtener ayuda.

loj con microcontrolador que emplea un display de cristal líquido de muy bajo consumo. El diagrama posee el valor de cada componente. El programa en Assembler se grafica en la figura 28.

EJEMPLOS DE APLICACIÓN CON PICs

Reloj con Display de Cristal LCD

En la figura 27 se da el circuito de un re-

Lector de Tarjeta Magnética Codificada.

Este circuito es muy fácil de construir y emplea un PIC preprogramado, lo cual no impide que un lector "experto" pueda pro-

gramar su propio componente, dado que brindamos el listado del programa compilado.

El prototipo emplea un lector de tarjeta magnética Canon CCM02 2NO 21 93 20 u otro similar. Precisamente, la dificultad de conseguir este componente en el mercado local es causa de que aún no hayamos

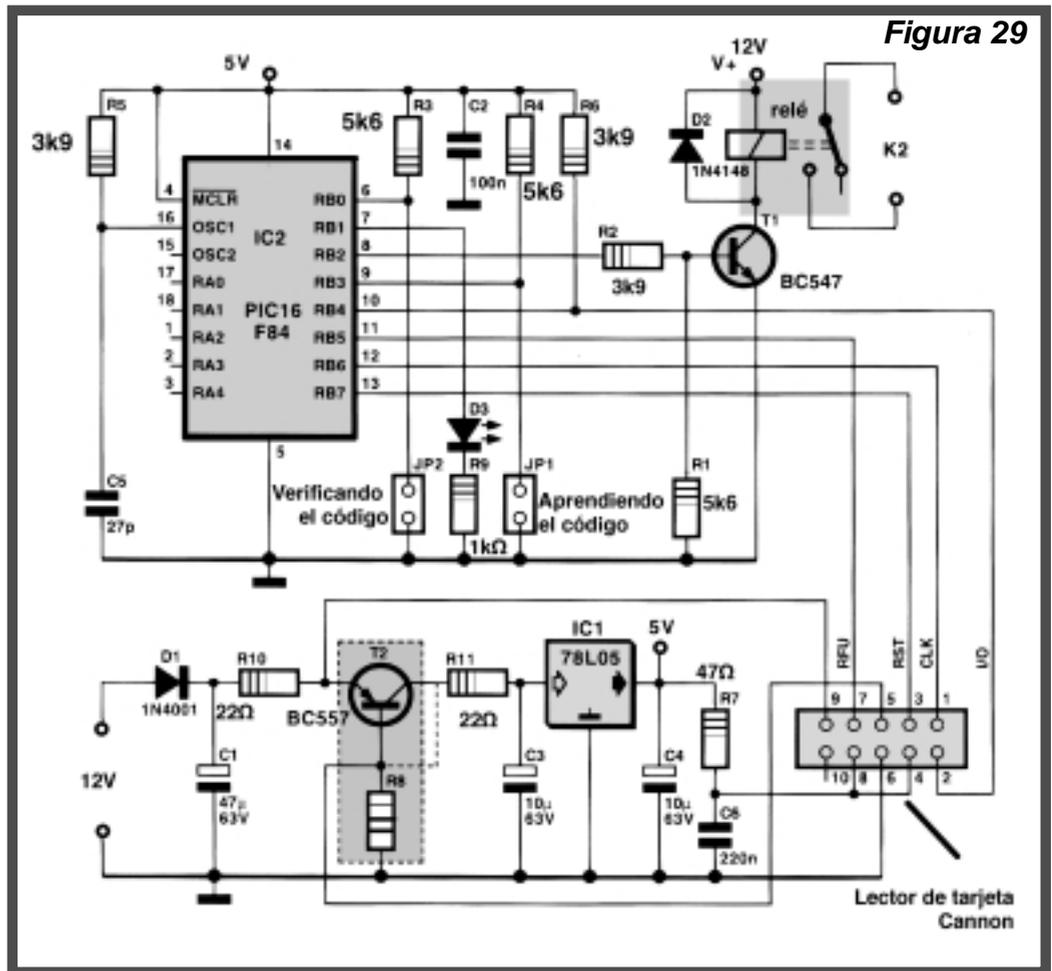


Figura 29

```

symbol f = B0
symbol g = B1
symbol e = B2
symbol r = B3
symbol BLINK = B4
symbol h = B5
DIRS = %11100110
PINS = %00000010
high 7
high 6
low 6
low 7
high 6
low 6
high 5
if PIN3 = 0 then learn
if PIN3 = 1 then check
learn:
for f = 0 to 31
e = 0
for g = 0 to 7
e = e * 2
e = e + PIN4
high 6
low 6

```

```

next g
h = 32 * PIN0
h = h + f
write h,e
next f
PINS = 0
BLINK = 50
goto led
check:
for f = 0 to 31
e = 0
for g = 0 to 7
e = e * 2
e = e + PIN4
high 6
low 6
next g
read f,r
if e <> r then other
goto cont
other:
h = f + 32
read h,r
if e <> r then error
cont:

```

```

next f
PINS = %00000110
goto forever
error:
PINS = 0
BLINK = 255
goto led
led:
for f = 0 to 10
high 1
pause BLINK
low 1
pause BLINK
next f
forever:
goto forever

```

Figura 30

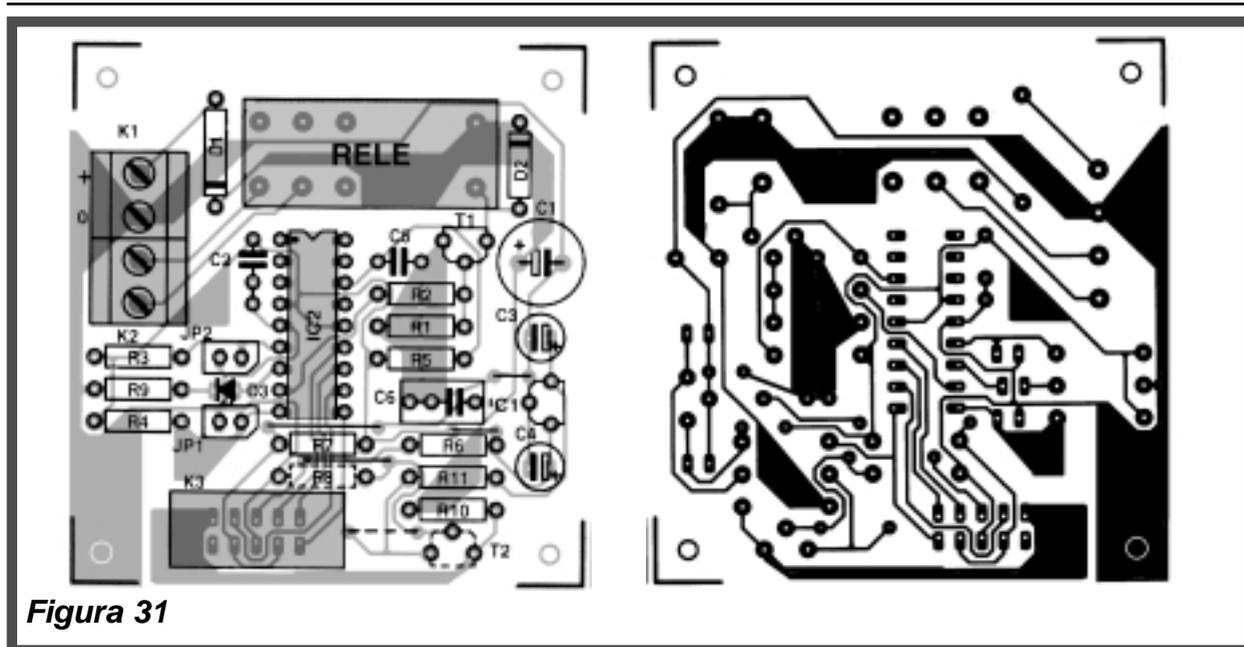


Figura 31

publicado un artículo completo sobre el tema, que tenemos a disposición desde hace varios meses. Sin embargo, confiamos en que los proveedores locales importen a la brevedad estos elementos con el objeto de que podamos publicar la mencionada nota.

En la figura 29 se da el circuito completo del lector de tarjeta (smartcard o tarjeta inteligente) con el valor de sus componentes.

En la figura 30 se da el diseño de la placa de circuito impreso para realizar el montaje y en la figura 31 se da el listado del programa compilado, que deberá ser cargado en el PIC.

Cuando se desea "grabar" una tarjeta con un código almacenado en el PIC, se debe colocar el jumper (puente) JP2, cuando se desea "leer" el código grabado en la tarjeta, se debe hacer un puente en JP1 y quitar el de JP2.