

## MPLAB

Paso 1: Instalación  
Paso 2: Configurar el Modo de Desarrollo  
Paso 3: Crear un nuevo Proyecto simple  
Paso 4: Crear un nuevo Archivo fuente simple  
Paso 5: Ingresar el Código fuente  
Paso 6: Ensamble del Archivo fuente  
Paso 7: Ejecución de su programa  
Paso 8: Abrir otras ventanas para el seguimiento de errores  
Paso 9: Crear una ventana de observación  
Paso 10: Marcar un punto de interrupción  
Resumen  
Algunas sugerencias  
Características del MPLAB no abarcadas en esta guía  
Otras ventanas de datos y diálogos  
Rastreo  
Estímulo  
Diálogo de Estímulo Asíncrono  
Archivos de Estímulo Pin  
Archivos de Estímulo de Registro  
Estímulo de Reloj  
Interrupción condicional  
Características del Editor  
Reloj de Detención  
Proyectos de archivo múltiple  
MPASM  
PICMASTER, ICEPIC, MPLAB-ICE  
PICSTART Plus y PRO MATE  
MPLAB-C17  
Herramientas de tercer orden  
Problemas  
Glosario

### **Paso uno: Instalación**

Descargue los archivos del software de instalación y ejecute el archivo MPxxxxx.EXE. Estos archivos pueden ser transferidos a disquetes si desea instalar el MPLAB en otra computadora. De acuerdo a la versión que ha 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

Cuando ejecuta 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 debería instalar las siguientes herramientas del software:

archivos MPLAB IDE - archivos MPASM/MPLINK/MPLIB - archivos de Protección del Simulador  
MPLAB-SIM - Archivos de Ayuda

(Pantalla)

Nota: Si tiene un programador PICSTART Plus, también debería seleccionar los "Archivos de Protección del PICSTART Plus". Si tiene otras herramientas (PICMASTER, MPLAB-ICE, ICEPIC, SIMICE, etc.), verifique los ítems asociados en los diálogos para instalar los drivers de software en su máquina. Más adelante podrá reinstalar el MPLAB para agregar componentes adicionales.

El siguiente menú le permitirá seleccionar los componentes de lenguaje Microchip que desee instalar. Usualmente debería seleccionarlos todos (por defecto).

(Pantalla)

Luego de instalarlos, ejecute MPLAB.EXE o cliquee el icono MPLAB para iniciar el sistema. Aparecerá el escritorio del MPLAB.

(Pantalla)

## **Paso 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 previa). 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. Más tarde nos ocuparemos de estas características. Por ahora, veremos cómo se configura el modo de desarrollo.

Nota: El "modo de desarrollo" determina la herramienta, si elige alguna, que ejecutará el código. Para esta guía, usaremos el simulador de software MPLAB-SIM. Si tiene un emulador, más tarde podrá cambiar a una de sus operaciones. La operación será similar. El modo "Sólo Editor" no permite la ejecución del código, y sólo es útil en caso de no haber instalado el simulador, no disponer de un emulador, y que sólo cree el código para programar un micro PIC.

Al seleccionar el ítem del menú "Opciones>Modo de Desarrollo", aparecerá una caja de diálogo semejante a la siguiente:

(Pantalla)

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 botón radio próximo al Simulador MPLAB-SIM y elija 16F84 en la lista de procesadores disponibles que pueden ser resistidos por el simulador. Cliquee 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.

## **Paso 3: Crear un nuevo 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.

Nota: El lenguaje ensamblador produce, entre otros elementos, un archivo hex. Este archivo tiene la extensión.hex. En esta guía, el archivo se llamará *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" en el menú y aparecerá el siguiente diálogo:

(Pantalla)

Cliquee el botón Sí, seguidamente aparecerá un diálogo de exploración de Windows standard. Decida dónde desea crear su proyecto y recuerde dónde lo ubicó. Más tarde necesitará esta información. Esta guía usa un directorio en *c: \temp\tutorial* y creó el archivo de proyecto llamado *tutor84.pjt*.

"PJT" es el sufijo standard 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.

Cliquee "Aceptar" 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 link del código fuente. Algunas herramientas diferentes pueden crear archivos hex, y estas herramientas forman 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. En esta guía no nos ocuparemos de estos detalles, pero cuando necesite estas características, deberá usar "Propiedades del Nodo" para establecerlas. Vea la Guía del Proyecto v3.40 del MPLAB para más información sobre proyectos complejos.

El diálogo "Editar Proyecto" será semejante al siguiente:

(Pantalla)

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", encontrará *tutor84. [hex]*. Al destacar este nombre el botón "Propiedades del Nodo" se tornará utilizable. 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 ensamble. Esta será la configuración por defecto cuando aparezca el diálogo "Propiedades del Nodo".

(Pantalla)

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", 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", aplicará estas configuraciones, y retornará al diálogo "Editar Proyecto", con el botón "Agrega Nodo" disponible.

Presione el botón "Agregar Nodo". Aparecerá el diálogo de exploración de Windows standard, con el mismo directorio usado para el proyecto. Ingrese el nombre de archivo, *tutor84.asm*, y presione "Aceptar". Retornará al diálogo "Editar Proyecto", donde podrá ver "*tutor84.asm*" añadido debajo del archivo hex, indicando que es un nodo concurrente.

Al presionar "Aceptar", retornará al escritorio MPLAB con un archivo de código fuente abierto y aún sin nombre.

#### **Paso cuatro: Crear un nuevo Archivo fuente simple**

Cliquee dentro del espacio en blanco de la ventana de archivo creada. Seguramente se llamará "Sin título". 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 standard, encontrará su ubicación en el directorio del proyecto. Ingrese el nombre de archivo y presione "Aceptar".

Ahora estarán disponibles el escritorio MPLAB y la ventana de archivo vacío, 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 link 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 link). En esta guía, para el tipo de archivo fuente del proyecto, 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.

#### **Paso cinco: Ingresar el Código fuente**

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

```
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
      mov1w   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
```

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". Busque en la Guía del Usuario MPASM con MPLINK y MPLIB más información sobre directivas. 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".

### **Paso seis: 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". 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 siguiente ventana "Resultados de Construcción":

(Pantalla)

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" para hacer una búsqueda de errores, y reparar los que aparecieran en el código fuente. Re-ensamble el archivo ejecutando la función de menú "Proyecto>Construir todo". Este procedimiento puede demandar un par de repeticiones.

Nota: Cuando re-construya 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". Ya ha completado un proyecto que puede ejecutarse usando el simulador.

### **Paso 7: Ejecución de su programa**

Use "Debug>Ejecutar>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". Al hacerlo, el contador del programa avanzará hasta la siguiente ubicación de instrucción. La barra oscura seguirá al código fuente y el contador del programa desplegado en la barra de estado avanzará hasta "4".

(Pantalla)

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 podrá ver al contador del programa y a la barra avanzar a través del programa.

Ejecute el ítem de menú "Debug>Ejecutar>Ejecutar" o presione F9 para iniciar la ejecución del programa desde la ubicación actual del contador. Los colores de la barra de estado cambiarán, indicando 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>Detener" o presionando F5. La barra de estado volverá a su color original, y el contador del programa y otras informaciones de su estado serán actualizados.

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 standard "cambiar barra de herramientas" 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).

### **Paso ocho: 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". Aparecerá una pequeña ventana con todos los registros de archivo o el RAM del 16F84.

Presione F7 (ejecutando 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, su 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, resultando más difícil focalizar las variables que le interesan. Este problema puede solucionarse usando una ventana de observación especial.

### **Paso nueve: Crear una ventana de observación**

Ejecute el ítem de menú "Ventana>Nueva ventana de observación". Aparecerá el diálogo "Agregar Símbolo de observación".

(Pantalla)

Típee "c1" en la caja de nombre de símbolo para que la lista se despliegue hasta el símbolo deseado. Selecciónelo, presione el botón "Agregar", y luego el botón "Cerrar". Aparecerá en su escritorio MPLAB una ventana de observación desplegando 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.

Actividad opcional: Guardar la ventana de observación

Puede guardar la ventana de observación y sus configuraciones ejecutando el ítem "Guardar observación" 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 standard ubicado en el directorio del proyecto. Elija cualquier nombre arbitrario y presione "Aceptar".

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" 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; la observación referida desaparecerá de la ventana. Puede seleccionar "Editar observación" 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.).

### **Paso diez: Marcar un punto de interrupción**

Presione F5 ("Debug>Ejecutar>Detener") para asegurarse que el procesador del simulador se ha detenido. Cliquee 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 siguiente menú:

(Pantalla)

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

Presione F6 o ejecute el ítem de menú "Debug>Ejecutar>Reset" para resetear el sistema. Luego ejecute el sistema presionando F9. 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**

Esta guía le ha enseñado a:

- 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

Una vez que maneje los temas que le enseñamos hasta aquí, puede ver la siguiente sección para obtener más información sobre el MPLAB.

### **Algunas sugerencias:**

**Puntos de interrupción** - Puede marcar puntos de interrupción en la ventana "Ventanas>Memoria de programa", en la ventana de archivo fuente (en este caso *tutor84.asm*), o en la ventana "Ventanas>Listado Absoluto".

**Archivos fuente** - Use "Ventana>Ventana de Proyecto" 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" 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" 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.

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

### **Características del MPLAB no abarcadas en esta guía**

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**

Esta guía ya le ha presentado la Ventana de Memoria de Programa y las Ventanas de Observación. Aún no ha explorado las otras ventanas. Debería 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. Busque en la guía del usuario del simulador más detalles sobre la información recogida por su amortiguador de rastreo de hardware.

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 mostrará una información algo diferente a la registrada por el rastreo del emulador.

Para usar el amortiguador de rastreo del simulador, primero debe seleccionar un código a rastrear. Si clics 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)".

(Pantalla)

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

(Pantalla)

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

## **Estímulo**

El estímulo genera señales para el simulador. Puede establecer pins (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 pins de entrada
- Estímulo de Archivo Pin - Los contenidos de un archivo de texto que describe señales a los pins 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

Diálogo de estímulo asincrónico

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

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 Asincrónico..."

(Pantalla)

Se desplegará el siguiente diálogo

(Pantalla)

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

(Pantalla)

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..."

(Pantalla)

Aparecerá otro diálogo con una lista de los pins conectados al 16F84

(Pantalla)

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

Debería aparecer el siguiente diálogo Estímulo Asincrónico:

(Pantalla)

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 Asincrónico. Debería ver el cambio de valor del puerto b en la ventana Registro de Función Especial mientras repetidamente cliquee el botón para simular una señal alta y luego una baja aplicadas al pin 0 del puerto b.

Archivos de estímulo pin

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

Use "Archivo>Nuevo archivo" y tipee el siguiente texto. No debe tipear el texto que sigue a las ";" y "!", que limitan los comentarios.

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.

Use "Archivo>Guardar como..." para guardar como *tutor84.sti*.

Nota: Para compatibilidad retroactiva con versiones anteriores del MPSIM, 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 pins del microPIC que recibirán los valores de estímulo altos y bajos. En este ejemplo los pins RB1 y RB0, dos entradas del 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 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 pins de soporte en la lista de asignación de pins para el Estímulo Asíncrono (clicquee el botón derecho del mouse sobre un botón de estímulo).

(Pantalla)

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 seleccionando "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.

(Pantalla)

Resetee 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 valores que serán enviados a un registro cuando la dirección de memoria del programa alcance la ubicación establecida 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 nómbrela *tutor84.reg*.

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

(Pantalla)

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:

(Pantalla)

Abra la ventana "Ventana>Registros de archivo" para ver el efecto de este estímulo.

(Pantalla)

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 archivo seleccionado (0x0D, con un valor de 0x50 como se muestra arriba). 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.

(Pantalla)

Si abre el diálogo "Debug>Estímulo del Simulador>Estímulo de Reloj...", podrá ingresar varios estímulos de reloj como se muestra en la figura. 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. Vea la Guía del Usuario de MPLAB.

### **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. Vea la Guía del Usuario de MPLAB para una explicación completa.

### **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 basándose en 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.

### **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 Micro ingeniería 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 micro PIC y Sistemas de Desarrollo en [www.microchip.com](http://www.microchip.com) para hacer preguntas que pueden ser útiles para principiantes o expertos. Los ingenieros de Microchip Aplicación, experimentados usuarios del micro PIC, y los ingenieros de Sistemas de Desarrollo, 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.

La página web de Microchip también tiene vínculos a otros recursos útiles como libros, artículos y grupos de discusión online.

### **Glosario**

*Listado Absoluto* - Archivo generado cuando el proyecto está construido de manera tal que muestra las instrucciones de la fuente mezcladas con el código generado. Es útil si usa Macros en el MPASM o un compilador. Es especialmente útil para hacer seguimientos de errores, ya que puede atravesar las líneas en una ventana, viendo el código generado por el compilador, y ubicando instrucciones sutiles que serían difíciles de apreciar en el nivel de un lenguaje de alto nivel.

*Lenguaje ensamblador* - Programa que convierte instrucciones mnemotécnicas en un código de máquina que pueda ejecutarse en un microcontrolador. Hay una correspondencia uno-a-uno entre la mnemotécnica y el código de máquina.

*Punto de interrupción* - Un punto de detención en el programa para asistir al seguimiento de errores. Cuando se ejecuta el programa, el mismo se detendrá en un punto de interrupción, permitiendo que los registros internos y las variables del usuario sean inspeccionados.

*Proyecto construido* - El acto de ensamblar o compilar todos los componentes en un proyecto.

*Compilador* - Programa que convierte las proposiciones de un lenguaje de alto nivel en un código de máquina que puede ejecutarse en un microcontrolador. Una proposición de un lenguaje de alto nivel usualmente genera una cantidad de instrucciones de código de máquina.

*Modo de Desarrollo* - Establece la herramienta que ejecutará el código (en el caso de que el mismo sea ejecutado por alguna herramienta). Para esta guía hemos usado el MPLAB-SIM, el simulador de software. Puede cambiar a una de las operaciones del emulador si tiene uno instalado, como el MPLAB-ICE. El modo "sólo Editor" no permite la ejecución del código, y es principalmente útil si no tiene instalado el simulador, no tiene un emulador, o si crea un código para programarlo directamente en el micro PIC.

*Programador de dispositivo* - Instrumento de hardware que toma el código de máquina desde un archivo y lo programa en un microcontrolador programable o un chip de memoria programable.

*Emulador* - Instrumento de hardware usado en lugar del microcontrolador en una aplicación. El emulador permite que el código sea rápidamente descargado, ejecutado y probado, tal como será ejecutado en la aplicación final.

*Registros de archivo* - El área RAM interna de micro PIC usada para almacenaje de variables.

*Detención* - Una función debug resultante de un punto de interrupción. El microcontrolador de destino se detiene en una ubicación de memoria del programa para que los registros y variables puedan ser inspeccionados.

*Archivo HEX* - Representación ASCII del código de máquina. Un archivo HEX está compuesto de registros que le especifican al microcontrolador datos o instrucciones que serán ubicados en un dispositivo de memoria programable.

*IDE - Entorno de Desarrollo Integrado* - Aplicación que usa múltiples funciones y características para el desarrollo del código. Las distintas herramientas pueden ser usadas sin cambiar a otros programas.

*Linker* - Programa que convierte el código de un objeto en un código de máquina ejecutable. El linker determina dónde se ubicará el código en la memoria y cómo será usada la RAM por las variables.

*Nodo* - Uno de los archivos de la lista de la ventana Archivos de Proyecto del diálogo Editar Proyecto. Hay un nodo de Proyecto que es la salida, y uno o más nodos de entrada. En esta guía, un archivo fuente MPASM es el único nodo de entrada usado. En otros tipos de proyecto, pueden haber varios nodos de entrada, como archivos de ensamble, archivos fuente c, objetos pre-compilados, bibliotecas y archivos de escritura del linker.

*Código de Objeto* - Código intermedio producido por un lenguaje ensamblador o compilador. Este código no contiene direcciones para la memoria del programa o las variables del RAM incluidas en el mismo, pero contiene demarcadores para las direcciones que el linker debe determinar cuando coloca el código en la memoria del programa.

*Memoria de Programa* - La memoria de un microcontrolador que contiene una serie de instrucciones para ejecutar una aplicación.

*Proyecto* - Grupo de archivos usados para crear una aplicación, junto con instrucciones a lenguajes ensambladores, compiladores y linkers para convertir los archivos en un código de máquina ejecutable. Los archivos pueden ser archivos fuente de ensamble, archivos fuente del compilador, bibliotecas, archivos de objeto pre-compilados, y archivos de instrucciones llamados escrituras del linker.

*Simulador* - Programa de software que imita la ejecución de un microcontrolador en una PC.

*Código fuente* - Archivo de texto que es procesado por un lenguaje ensamblador o un compilador para producir un archivo de objeto intermedio, o código de máquina que pueda ejecutarse en un microcontrolador.

*Registros de Función Especial* - Ubicaciones internas de un micro PIC que pueden ser usadas para afectar la operación interna del controlador o un periférico. Entre los ejemplos se pueden incluir los registros de control de interrupción, los timers y los registros I/O.

*Rastreo* - Ventana que muestra la ejecución de las instrucciones del microcontrolador. Los emuladores tiene un analizador de rastreo de hardware que recoge la información mientras el procesador de destino se ejecuta en tiempo real. Estos datos son cargados a la ventana de rastreo para que puedan ser visualizados en el MPLAB. El simulador también permite operaciones de rastreo para la ejecución del logging.

*Ventana de observación* - Un dispositivo de seguimiento de errores que muestra los contenidos de variables seleccionadas. La muestra puede ser modificada para desplegar los valores en una cantidad de formatos, como el ASCII, hex, binario, o punto flotante.