

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

The Microchip logo, name, PIC, PICMASTER, PICSTART and TrueGauge are registered trademarks of Microchip Technology Inc. in the U.S.A. and other countries. MPLAB, and PRO MATE are trademarks of Microchip in the U.S.A.

All rights reserved. All other trademarks mentioned herein are the property of their respective companies.

© Microchip Technology Incorporated 1996.

*fuzzy*TECH is a registered trademark of Inform Software Corporation.

Intel is a registered trademark of Intel Corporation.

IBM PC/AT is a registered trademark of International Business Machines Corporation.

Netscape is a trademark of Netscape Communications Corporation.

Windows, Excel and Microsoft Explorer, Dynamic Data Exchange are trademarks of Microsoft Corporation.



# **Table of Contents**

## Part 1 – IDE and Simulator

MPLAB Pre	view
	A Quick Look at MPLAB <sup>™</sup>
	MPLAB, An Integrated Development Environment
Chapter 1.	About MPLAB
	Introduction
	Highlights
	MPLAB Integrated Development Environment5
	MPLAB Tools
	MPLAB Hardware and Software Requirements7
	About This Guide7
	Recommended Reading9
	Warranty Registration10
	Customer Support
Chapter 2.	MPLAB Installation
	Introduction
	Highlights
	MPLAB Setup
	Full MPLAB Installation12
	Custom MPLAB Installation12
	Microchip MPLAB Program Group13
	Location of Installed MPLAB Files13
	MPLAB Installation Log
	Running MPLAB
Chapter 3.	MPLAB Tutorial
	Introduction
	Highlights
	Accessing MPLAB Tools and Menu Functions
	Using Projects
	Setting Up and Using the MPLAB Desktop

## Chapter 4. MPLAB Projects

	Introduction	27
	Highlights	27
	New Project	27
	Open Project (Ctrl+F2)	29
	Close Project	29
	Save Project	30
	Edit Project (Ctrl+F3)	30
	Make Project (F10)	31
	Build All (Ctrl+F10)	
	Compile Single File (Alt+F10)	33
	Make Setup	33
	Most Recently Used Projects	
	Drag and Drop Loading	
Chapter 5.	MPLAB Basic Functions	
-	Introduction	35
	Highlights	
	MPLAB Functions	
	Real-Time Program Execution	
	MPLAB-SIM Simulator Environment	
	Simulator Considerations	
	Break, Trace, and Trigger Points	
	Conditional Break Dialog	
	Special Windows	
	Stimulus Functions	
	File Extensions Used by MPLAB	
Chapter 6.	MPLAB Menu Options	
•	Introduction	51
	Highlights	
	MPLAB Functions	
	MPLAB Desktop	
	File Menu	
	Project Menu	
	Debug Menu	
	Options Menu	
	, Tools Menu	
	Window Menu	
	Help Menu	

## Part 2 – Editor

<b>MPLAB Edit</b>	tor Preview
	A Quick Look at MPLAB Editor109
	MPLAB Editor Features
Chapter 1.	About MPLAB Editor
	Introduction
	Highlights
	Product Definition
Chapter 2.	Using the MPLAB Editor
	Introduction
	Highlights113
	File Operations
	Window Modes114
	Templates
	Text Handling
	C Language Awareness120
Chapter 3.	MPLAB Editor Menu Commands
	Introduction
	Highlights
	File Menu
	Editor Menu
	Options Menu
Chapter 4.	MPLAB Editor Default Key Commands
	Introduction
	Highlights
	Function Keys
	Movement Keys
	Control Keys
	Formatting and Editing Keys135
Chapter 5.	MPLAB Editor Error Messages
	Introduction
	Error Messages
Chapter 6.	MPLAB Text Editor Command Line Options
	Introduction
	Command Line Options141

## Part 3 – Appendices

Appendix A.	DDE Linking Support
	Introduction
	DDE Overview
	Typical Uses for DDE
	DDE Conventions
	DDE Warm Link Example147
Appendix B.	MPLAB Key Mapping Functions
	Introduction
	MPLAB Key Mapping Functions149
Appendix C.	On Line Support
	Introduction
	Connecting to the Microchip Internet Web Site
	Connecting to the Microchip BBS
	Using the Bulletin Board
	Software Releases
Annondix D	
Appendix D.	Customizing MPLAB After Installation
	Introduction         161           Options Section         162
Appendix E	•
Appendix E.	PIC16C5X Simulator Issues
	Introduction         165           PIC16C5X Selection Table         165
	I/O Pins
	CPU Mode
	Peripherals
Appendix F.	PIC16CXX Simulator Issues
	Introduction
	Product Selection Tables
	I/O Pins
	Interrupts
	CPU Model
	Special Registers
	Peripherals

## Appendix G. MPLAB-SIM PIC17CXX Simulator Issues

	Introduction	
	PIC17CXX Selection Table	
	I/O Pins	
	Interrupts	
	CPU Model	
	Special Registers	
	Peripherals	
	Memory Modes	
Appendix H.	Upgrading from MPSIM to MPLAB-SIM	
	Introduction	
	Highlights	
	System Navigation	
	Program Memory	
	Breakpoints	
	Tracing	
	Registers and Data Memory	
	Stimulus and Timer	
	Program Memory Patch and Control	
	Symbol Table	190
	View Screen	
	System Setup and Control	191
Glossary		
	Introduction	193
	Highlights	193
	MPLAB Terms	
	PICMASTER Terms No Longer Used	
Index		
	Index	



## Part 1 – IDE and Simulator

Chapter 1.	MPLAB Preview	3
Chapter 2.	About MPLAB	5
Chapter 3.	MPLAB Installation	11
Chapter 4.	MPLAB Tutorial	15
Chapter 5.	MPLAB Projects	27
Chapter 6.	MPLAB Basic Functions	35
Chapter 7.	MPLAB Menu Options	51



### **MPLAB** Preview

### A Quick Look at MPLAB<sup>™</sup>

### What is MPLAB

MPLAB is a Windows<sup>®</sup> based Integrated Development Environment (IDE) for the Microchip Technology Incorporated PIC16/17 microcontroller families. MPLAB allows you to write, debug, and optimize PIC16/17 applications for firmware product designs. MPLAB includes a text editor, simulator, and project manager. MPLAB also supports the PICMASTER<sup>®</sup> emulator and other Microchip development system tools.

### How MPLAB Helps You

The organization of MPLAB tools by function helps make pull-down menus and customizable quick keys easy to find and use. MPLAB tools allow you to:

- Debug your source code
- Automatically locate errors in source files for editing
- Debug with breakpoints based on internal register values
- Watch the program flow with the simulator or watch in real time with the PICMASTER emulator.
- Make timing measurements with a stop watch
- View variables in watch windows
- Find quick answers to questions from the MPLAB on-line Help

### MPLAB, An Integrated Development Environment

The MPLAB desktop provides an environment and development tools for developing and debugging your applications as a project.

### **MPLAB Development Tools**

The MPLAB Integrated Development Environment allows you to quickly move between different development/debugging modes. For example, you can quickly advance from software debugging with the MPLAB-SIM to hardware debugging with the PICMASTER emulator.

The MPLAB Integrated Development Environment includes the following development tools:

#### MPLAB Project Manager

The Project Manager enables you to create a project, and work with specific files related to the project. The MPLAB Project Manager also allows you to build an application and download your code to the emulator or the simulator with a mouse click.

#### MPLAB Editor

The MPLAB Editor allows you to write and edit firmware source files or other text files for PIC16/17 microcontrollers.

#### MPASM Universal PIC16/17 Microcontroller Assembler

The MPASM Universal PIC16/17 Microcontroller Assembler allows you to assemble your code without exiting MPLAB.

#### • MPLAB-SIM Software Simulator

The simulator simulates instruction execution and I/O of PIC16/17 microcontrollers.

#### PICMASTER Emulator

The PICMASTER emulator uses hardware to emulate PIC16/17 microcontrollers in real time. The PICMASTER emulator hardware is sold separately.

### **C** Compiler Support

MPLAB supports C compilers that generate the Byte Craft COD output format.

### **MPLAB Operating Modes**

MPLAB runs in three operating modes:

- MPLAB-SIM Simulator Mode provides a fast tool for debugging firmware designs.
- Editor Only Mode allows you to write, compile/assemble, and remove errors from source code without using the simulator or emulator.
- PICMASTER Emulator Mode allows you to run your target application in real time while performing hardware debugging with the PICMASTER emulator.



### Chapter 1. About MPLAB

### Introduction

This chapter describes MPLAB, lists its primary features, and suggests recommended reading.

## **Highlights**

"About MPLAB" covers the following topics:

- MPLAB Integrated Development Environment
- MPLAB Tools
- About This Guide
- Recommended Reading
- Warranty Registration
- Customer Support

### **MPLAB Integrated Development Environment**

MPLAB is an easy-to-learn, easy-to-use Integrated Development Environment (IDE). The IDE provides firmware development engineers the flexibility to edit, compile, emulate, develop and debug firmware for Microchip's PIC16/17 microcontroller families. The MPLAB IDE runs under Microsoft Windows<sup>®</sup> 3.1 or later.

MPLAB provides functions that allow you to:

- Create and Edit Source Files
- Group Source Files into Projects
- Debug Source Code

### **MPLAB Tools**

MPLAB is a suite of tools for developing and debugging applications in a project. MPLAB includes a text editor, a project manager for keeping your code organized, and an environment for debugging applications for firmware product designs. The environment includes the MPLAB-SIM simulator, and supports other Microchip tools such as the PICMASTER emulator.

#### **Project Manager**

Use a project to do symbolic debugging

The Project Manager is the central metaphor of MPLAB. Without creating a project you cannot do symbolic debugging. Through the Project Manager you do the following operations:

- Create a Project
- Add a Source Code File to a Project
- Assemble or Compile Source Code
- Edit Source Code
- Rebuild All Source Files, or Compile a Single File
- Debug your Source Code

### MPLAB Editor

MPLAB Editor enables programmers to write, and edit source code for PIC16/17 microcontroller families, as well as other text files. See Part 2 of this manual for complete information on the features of MPLAB Editor.

#### **MPASM** Assembler

MPASM Universal PIC16/17 Microcontroller Assembler offers full-featured macro capabilities, conditional assembly and several source and listing formats. MPASM allows you to generate various object code formats to support Microchip development tools as well as third party programmers without exiting MPLAB. See *MPASM User's Guide*, Document Number DS33014, for complete information on the features of MPASM.

### **MPLAB-SIM Simulator**

The MPLAB-SIM simulator allows you to isolate code problems and debug firmware designs on PIC16/17 microcontrollers. It simulates the core functions as well as most of the peripherals of the PIC16/17 microcontroller family.

### **PICMASTER Emulator**

The PICMASTER emulator is a high-performance in-circuit emulator that allows you to run your target application in real time while performing hardware debugging operations.

The PICMASTER emulator has the same user interface as the MPLAB-SIM simulator. Thus, if you need to perform hardware debugging, you can easily change from simulator debugging to emulator debugging. For more information on the PICMASTER emulator refer to the PICMASTER Emulator User's Guide.

#### **Other Tools**

The MPLAB IDE will support Microchip development tools and third party development tools such as programmers, compilers, and other products as they become available.

### **MPLAB Hardware and Software Requirements**

Under the Windows multitudinous operating system you can concurrently run many programs such as EMAIL, spreadsheets, and the MPLAB IDE. Since the MPLAB IDE has multiprocessor emulation capabilities, and supports DDE (Dynamic Data Exchange) with client programs, data collected with the MPLAB IDE can be shared with programs such as Microsoft Excel<sup>™</sup>.

### **Host Computer System Requirements**

To take advantage of the emulator system features, you must install the MPLAB software (MPLAB.EXE) on a host computer having the following minimum configuration:

- Personal Computer with 386 or Higher Processor. Pentium<sup>™</sup> Recommended.
- 4 MB of Memory, 16 MB Recommended
- 8 MB of Hard Disk Space, 20 MB Recommended
- VGA or Super VGA Monitor
- Microsoft Windows 3.1 or Greater

### **About This Guide**

This user guide describes how to use the MPLAB IDE with the MPLAB-SIM simulator. It also covers the MPLAB Editor, most MPLAB functions, and MPLAB Projects.

This manual is divided into three parts as follows:

#### Part 1 – IDE and Simulator

MPLAB Preview – Gives a quick look at MPLAB and how it can help you.

**Chapter 1: About MPLAB** – Itemizes the MPLAB features, describes how this user's guide is organized, and lists recommended reading.

**Chapter 2: MPLAB Installation** – Describes the procedures for installing MPLAB.

**Chapter 3: MPLAB Tutorial** – Takes you through many of the MPLAB functions. After performing each example in this tutorial, you should have a good understanding of how to use MPLAB.

**Chapter 4: MPLAB Projects** – Gives detailed information on using MPLAB Projects to manage the development of PIC16/17 microcontroller applications.

**Chapter 5: MPLAB Basic Functions** – Discusses the basic MPLAB debugging functions and shows how the functions interrelate.

**Chapter 6: MPLAB Menu Options** – Gives detailed information on using the MPLAB menu options. The chapter organization follows the entries on the pull down menus.

#### Part 2 – Editor

**MPLAB Editor Preview** – Gives you a quick look at what the MPLAB Editor is and how it can help you.

Chapter 1: About MPLAB Editor - Itemizes MPLAB Editor's features.

**Chapter 2: Using the MPLAB Editor** – Describes the procedures for using the MPLAB Editor functions.

**Chapter 3: MPLAB Editor Menu Commands** – Gives detailed information on MPLAB Editor Menu Options. The items are organized in the same order as you see the entries on the pull down menus.

**Chapter 4: MPLAB Editor Default Key Commands** – Gives one or two line description of each default key command. The entries are organized by function.

**Chapter 5: MPLAB Editor Error Messages** – Gives you additional information suggesting possible actions you might take should you receive an error message. This chapter also contains non-error informative messages.

**Chapter 6: MPLAB Text Editor Command Line Options** – Describes command line options you can use when starting MPLAB.

#### Part 3 – Appendices

**Appendix A: DDE Linking Support** – Describes how to use the Microsoft<sup>®</sup> Dynamic Data Exchange with MPLAB serving as a client.

**Appendix B: MPLAB Key Mapping Functions** – Lists the available MPLAB key mapping functions.

Appendix C: On Line Support – Information on Microchip's electronic support services.

**Appendix D: Customizing MPLAB after Installation** – Discusses the initialization file used by MPLAB (MPLAB.INI) as well as various user-options available for customizing MPLAB.

**Appendix E: MPLAB-SIM PIC16C5X Simulator Issues** – Discusses I/O pins, interrupts, registers, peripherals, modes, and conditions for using the PIC16C5X family of microcontrollers.

**Appendix F: MPLAB-SIM PIC16CXX Simulator Issues** – Discusses I/O pins, interrupts, registers, peripherals, modes, and conditions for using the PIC16CXX and PIC14000 families of microcontrollers.

**Appendix G: MPLAB-SIM PIC17CXX Simulator Issues** – Discusses I/O pins, interrupts, registers, peripherals, modes, and conditions for using the PIC17CXX family of microcontrollers.

**MPLAB System Glossary** – Defines the terms associated with MPLAB, MPLAB Editor, the PICMASTER emulator, and the MPLAB-SIM simulator.

**Index** – Provides a quick reference to MPLAB functions and features discussed in this manual.

#### **Conventions Used in this Guide**

This manual uses the following documentation conventions:

Table 1.1 Documentation Conventions

Character	Represents
Angle Brackets ( < > )	Delimiters for special keys: <tab>, <esc>, etc.</esc></tab>
Pipe Character (   )	Choice of mutually exclusive arguments; an OR selection
Lower case characters	Type of data
Italic characters	A variable argument; it can be either a type of data (in lower case characters) or a specific example (in uppercase characters)
Courier Font	User entered code or sample code
Underlined, Italics Text with Right Arrow >	Defines a menu selection from the menu bar: <i><u>File &gt; Save</u></i>
0xnnn	0xnnn represents a hexadecimal number where n is a hexadecimal digit
In-text Bold Characters	Designates a button such as <b>OK</b>

### **Recommended Reading**

This manual describes how to use MPLAB. You may also want to read the Data Sheets for the microcontroller devices for which you are developing/ debugging firmware.

Check README.LAB for the latest information.

**<u>README.LAB</u>** For the latest information on using MPLAB, read the README.LAB file (an ASCII text file) on the MPLAB diskette. README.LAB contains update information that may not be included in the *MPLAB User's Guide.* 

**<u>README.XXX</u>** For the latest information on using other tools, an information file about the product may exist that is more current than the printed manual. Check the MPLAB directory for other README files. (In the case of MPASM, for instance, the file is called README.ASM.)

**PIC16/17 Microcontroller Data Book** Contains comprehensive data sheets for Microchip PIC16/17 microcontroller devices available at print time. *Document Number DS00158, Microchip Technology Inc., Chandler, AZ.* 

**Embedded Control Handbook** Contains a wealth of information about microcontroller applications. *Document Number DS00092, Microchip Technology Inc., Chandler, AZ.* The application notes described in this manual are also available from the Microchip BBS and the Microchip Internet Home Page. See Appendix C, On Line Support, for more information about these services.

<u>Microchip ECHB Update I</u> Contains additional application notes released since publication of the standard Embedded Control Handbook.

**MPASM User's Guide** Describes how to use Microchip Universal PIC16/17 Microcontroller Assembler (MPASM). *Document Number DS33014, Microchip Technology Inc., Chandler, AZ.* 

All of the above documents are available from your local sales office or your Microchip Field Application Engineer (FAE).

This manual assumes that you are familiar with both MS-DOS<sup>®</sup> and Microsoft Windows software systems. Many excellent references exist for both of these software programs, and should be consulted for general operation of DOS and Windows.

### **Warranty Registration**

Sending in your Warranty Registration Card ensures that you receive new product updates and notification of interim software releases that may become available.

### **Customer Support**

Microchip endeavors to provide the best service and responsiveness possible to its customers. Technical support questions should first be directed to your distributor and representative, local sales office, Field Application Engineer (FAE), or Corporate Applications Engineer (CAE).

The Microchip Internet Home Page can provide you with technical information, application notes, and promotional news on Microchip products and technology.

The Microchip Web address is http://www.microchip.com

You can also check with the Microchip BBS (Bulletin Board System) for non-urgent support, customer forums, and the latest revisions of Microchip systems development products. Refer to the "BBS Support" appendix for access information.



# Chapter 2. MPLAB Installation

## Introduction

This chapter describes the procedures for installing MPLAB.

# **Highlights**

The items discussed in this chapter include:

- MPLAB Setup
- Full MPLAB Installation
- Custom MPLAB Installation
- Microchip MPLAB Program Group
- Location of Installed MPLAB Files
- MPLAB Installation Log
- Running MPLAB

## **MPLAB Setup**

The MPLAB Setup routine installs the Microchip Integrated Development Environment. Microsoft Windows<sup>®</sup> must be running to execute MPLAB Setup.

We recommend performing a full MPLAB installation and using backups. You can install the MPLAB files in the default directory, C:\MPLAB, or in another directory.

- 1. Insert the MPLAB installation disk 1 in drive A.
- 2. From the Program Manager Run option, type A:Setup

The MPLAB Setup program displays a Welcome! message box with options to continue or exit. Click **OK** to continue.

- 3. Setup next displays a dialog to select the MPLAB directory. Enter the directory name to install the MPLAB files to and click **OK**.
- 4. At this point Setup asks you whether you want to make backups of files that it may overwrite. Click **Yes**. Initial installations do not create backup files.
- 5. Setup displays a dialog to select the backup directory. Enter the directory name for Setup to make backups in and click **OK**.
- 6. Next Setup displays a dialog for choosing a Full or Custom installation. Select **Full** installation and click **OK**. Setup then copies the MPLAB files to the selected directory.

- 7. After copying the MPLAB files, Setup displays a message box asking if you want to add the MPLAB icons to the Program Manager. Click **Yes**.
- 8. Setup displays a dialog that allows you to choose which Program Manager Group you want to install the MPLAB icons into. Setup displays Microchip MPLAB by default. Choose the group to add the icons to and click **OK**.
- 9. When complete Setup displays a message box with the caption "The MPLAB Installation has completed." Click **OK**.

# **Full MPLAB Installation**

In a full installation, MPLAB Setup copies the following applications into the designated directory:

- MPLAB Program Files (Including the Editor)
- MPLAB-SIM Simulator
- Tools (MPASM for Windows)
- Help

Before MPLAB Setup begins copying files, it checks to make sure that the selected drive has available space to install MPLAB.

# **Custom MPLAB Installation**

Custom installation, allows you to install any or all of the following:

- MPLAB Program Files (Including the Editor)
- MPLAB-SIM Simulator
- Tools (MPASM for Windows)
- Help

If, for space reasons, you do not install MPLAB-SIM, Help, or MPASM when you first install MPLAB, you can install this software later.

The Custom MPLAB Installation dialog lists the following information:

- Total Disk Space Required
- Remaining Free Disk Space
- Available Options

# **Chapter 2. MPLAB Installation**

# **Microchip MPLAB Program Group**

When MPLAB Setup is complete, the program group, Microchip MPLAB, contains three icons: MPLAB, MPASM for Windows, and Uninstall MPLAB.

# Location of Installed MPLAB Files

MPLAB Setup copies most of the files into the default directory, C:\MPLAB, or into the directory specified during the installation.

MPLAB Setup copies the following files to the \Windows directory:

MPC.PIF MPLAB.INI DEFAULT.TBR

and the following files to the \Windows\System directory:

OWL252.DLL BIDS45.DLL

If you reinstall MPLAB, the install program saves backup copies of all replaced files. If you want to restore any of the files replaced during the install, copy them from the backup directory specified during the install to the location specified above.

# **MPLAB Installation Log**

The MPLAB Setup program keeps track of the files that it installs in a file called INSTALL.LOG. The log file (INSTALL.LOG) has a record of each file copied during the installation process.

If you wish to remove MPLAB from your system, simply run the included Uninstall program from the Microchip MPLAB program group. The uninstall routine uses the INSTALL.LOG file to determine which files to remove from the MPLAB, Windows, and System directories.

# **Running MPLAB**

MPLAB can be started in a variety of ways:

- From the icon in the Microchip MPLAB program group
- From the Windows *File > Run* command
- From any other application that can launch programs

You can also include on the MPLAB command line a list of text files that MPLAB is to open. MPLAB Editor will open each file that you list on the command line. Filenames given on the command line may include the normal DOS wildcard characters.



## Chapter 3. MPLAB Tutorial

## Introduction

This tutorial takes you through many of the MPLAB functions. After performing this tutorial, you should have a good understanding of how to use MPLAB. The expected time to step through this tutorial is approximately one hour.

The two files used in this tutorial, TUTOR.ASM and SAMPLE.ASM, are located in the MPLAB directory.

If you decide to skip this tutorial, please note that you **MUST CREATE A PROJECT** to do symbolic debugging and make full use of MPLAB's Integrated Development Environment.

# Highlights

The items discussed in this tutorial include:

- Accessing MPLAB Tools and Menu Functions
- Using Projects
- Setting Up and Using the MPLAB Desktop

# **Accessing MPLAB Tools and Menu Functions**

MPLAB is an integrated development environment that integrates the MPLAB Editor, the MPLAB-SIM simulator, and Project Manager with other Microchip tools for configuring, organizing, and controlling the development of firmware product designs.

Familiarize yourself with the MPLAB tools and Menu Functions by clicking on each pull-down menu and looking at the menu functions.

The MPLAB menu functions are listed at the beginning of Chapter 6, MPLAB Menu Options, for your convenience.

# **Using Projects**

This tutorial uses an assembler file and MPASM, but you would use a similar process with a compatible C compiler. Projects allow you to define files related to a project. You can locate a project in any directory. In this tutorial, we use the directory C:\MPLAB\TUTOR to store projects.

**Note:** In this section, references to the assembler can also refer to the compiler.

© 1996 Microchip Technology Inc.

### **Creating a Project**

1. Click <u>Project > New Project</u> to open the New Project dialog.

New Project allows you to specify Drive, Directory, and Project Name. New Project can create new directories to be located anywhere you specify.

2. Create a new project as follows:

In the "Project Path and Name:" field, type

C:\MPLAB\TUTOR\TUTOR.PJT

and click OK.

Project Path and Name: C:\MPLAB\TUTOR\tutor.PJT	ОК
Default Toolbar:	Cancel
Default.tbr	<u>B</u> rowse
mplab.key	
Development Mode:	

Figure 3.1 New Project Dialog

3. Look at the title bar at the top of the desktop. The title bar should now give the name of the project as follows:

MPLAB - C:\MPLAB\TUTOR\TUTOR

4. MPLAB automatically opens *Project > Edit Project*.

 Note:
 You can also verify the project you just created from the File

 Manager. Look at C:\MPLAB\TUTOR. This directory contains two files:
 • TUTOR.CFG

 • TUTOR.CFG
 Project Configuration File

 • TUTOR.PJT
 Project File

### **Assigning Files to a Project**

In the following steps, we add the file TUTOR.ASM to the project, TUTOR.PJT.

- 1. Click <u>Project > Edit Project</u>. This step is not necessary if you are continuing from the previous section "Creating a Project."
- 2. Click Copy File...
- 3. Double click TUTOR.ASM to add this file to your project.

# **Chapter 3. MPLAB Tutorial**

4. Click OK.

MPLAB puts a copy of the file in your project directory.

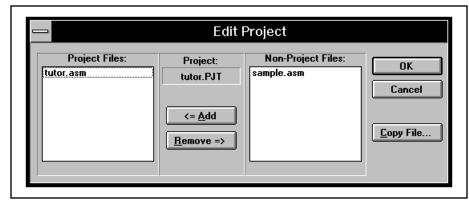
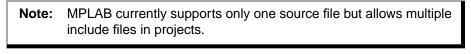


Figure 3.2 Edit Project Dialog



**Note:** Currently, only files with \*.C and \*.ASM extensions are allowed in a project.

## Assembling Source Code

<u>Project > Make Project</u> assembles (or compiles) the source code assigned to a project. Make Project first checks for differences in the time and date of the source file you are assembling against the time and date of the \*.COD file. (The \*.COD file contains object code and symbolic information.) If the source file is newer than the \*.COD file, Make Project rebuilds the project.

If the source is older than the \*.COD file, Make Project checks the time and date of any include files in the project. If any include files are newer than the \*.COD file, then MPLAB will rebuild. If you change an include file, Make Project will catch the change and force an update to the \*.COD file.

If the \*.COD is more recent than any of the source files, no action will be taken.

We will use the MPASM Assembler for the following examples.

### Example 1: Assembling Source Code Containing an Error Edit TUTOR.ASM

- 1. Click *File > Open Source.*
- 2. Select the directory, C:\MPLAB\TUTOR.
- 3. Double click on TUTOR.ASM. An editor window with TUTOR.ASM will open.
- 4. Click the system button in the upper left corner of the TUTOR.ASM window.

5. Click "Toggle Line Numbers."

#### Insert Error in TUTOR.ASM

1. Create and record an obvious error somewhere in the source code such as inserting a spelling error on line 56 by changing the word "Start" to "Star".

#### **Compile the TUTOR Project**

1. Click <u>Project > Make Project</u>.

The compiler opens the MPASM assembler. After the compile completes, the status message reads:

— Compile Status
Status:
There are errors.
Command Line:
mpasmwin C:\MPLAB\TUTOR\tutor.asm /q
ОК



#### Look at Compile Error

1. Close the Compile Status dialog box.

The error file opens.

1000	c3mplabit	utor	tu	itor.err	-
Error[113]	C:\MPLAB\TUTOR\TUTOR.ASM	53	:	Symbol not previously defined	(star
Message[302]	C:\MPLAB\TUTOR\TUTOR.ASM	59	1	Argument out of range. Least	signi
Message[302]	C:\MPLAB\TUTOR\TUTOR.ASM	61	1	Argument out of range. Least	signi
Message[302]	C:\MPLAB\TUTOR\TUTOR.ASM	63	:	Argument out of range. Least	signi
Message[302]	C:\MPLAB\TUTOR\TUTOR.ASM	66	1	Argument out of range. Least	signi
Error[113]	C:\MPLAB\TUTOR\TUTOR.ASM	68	1	Symbol not previously defined	(star
Message[302]	C:\MPLAB\TUTOR\TUTOR.ASM	71	1	Argument out of range. Least	signi
Message[302]	C:\MPLAB\TUTOR\TUTOR.ASM	72	1	Argument out of range. Least	signi
Message[302]	C:\MPLAB\TUTOR\TUTOR.ASM	73	1	Argument out of range. Least	signi
Message[302]	C:\MPLAB\TUTOR\TUTOR.ASM	74	1	Argument out of range. Least	signi
Message[302]	C:\MPLAB\TUTOR\TUTOR.ASM	79	1	Argument out of range. Least	signi
Message[302]	C:\MPLAB\TUTOR\TUTOR.ASM	80	:	Argument out of range. Least	signi
Message[302]	C:\MPLAB\TUTOR\TUTOR.ASM	81	:	Argument out of range. Least	signi
Message[3021	C:\MPLAB\TUTOR\TUTOR.ASM	82	1	Argument out of range. Least	signi

Figure 3.4 Error File Window

Double click on an error to locate the offending source line

# **Chapter 3. MPLAB Tutorial**

#### **Fix Inserted Error**

 Double click on the indicated error in the error file window. Usually it is best to start with the first error.

The MPLAB Editor opens TUTOR.ASM, and locates the current line with the error.

2. Use the MPLAB Editor to fix the error that you just created.

### Example 2: Debugging Source Code

1. Click <u>Project > Make Project.</u>

Make Project runs the MPASM Assembler. After completion, the status message should read:

— Compile Status
Status:
Success, no errors.
Command Line:
mpasmwin C:\MPLAB\TUTOR\tutor.asm /q
ОК

Figure 3.5 Compile Status

### **Building All Source Files**

Build All rebuilds all source files in the selected project window, ignoring time and date.

- 1. From the Windows File Manager, note the time and date of TUTOR.COD.
- 2. Click <u>Project > Build All</u> to build all source files.
- 3. Again note the time and date of TUTOR.COD. The time should be later than the previous time.

## **Compiling a Single File**

Compile Single File attempts to assemble (or compile) the file (\*.ASM or \*.C only) in the currently active window. The file does not have to be assigned to a project. However, you must have a project open to access the Compile Single File function.

- 1. Click *File > Open Source*
- 2. Select SAMPLE.ASM and click Ok.

The SAMPLE.ASM file opens.

3. Click <u>Project > Compile Single File</u>.

 $\mathsf{MPLAB}$  compiles the file, SAMPLE.ASM, and displays "Success, no errors."

Note: When running Make Project or Build All and the project compiles with no errors, MPLAB automatically reloads the binary code from the \*.COD file. When running Compile Single File, MPLAB does not reload the

### **Closing a Project**

1. Click <u>Project > Close Project</u>.

binary code.

2. Answer Yes to save the current project.

MPLAB saves the current project in the location specified in the Title Bar.

## Setting Up and Using the MPLAB Desktop

### Saving Current Desktop Setup

Save Setup is only available when one or more windows are open. You can save your current desktop configuration under a \*.CFG name that you choose, or you can save the configuration as the default.

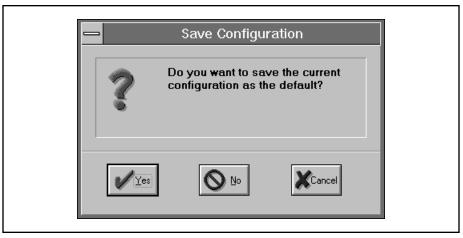


Figure 3.6 Save Configuration Dialog

If you choose **Yes** to save the current window configuration as the default, this configuration will load when you select Load User Configuration in the <u>Options > Environment Setup</u> dialog. The file name of the default configuration is MPLAB.CFG.

In the following example, we will choose  $\ensuremath{\text{No}}$  and save the configuration as SAMPLE.CFG.

- 1. Close all open windows.
- 2. Click *File > Open Source* and open SAMPLE.ASM.
- Click <u>Options > Window Setup > Save Setup</u> to save the current window setup.
- 4. Click **No** to display a save dialog box. Select the name SAMPLE.CFG to save the window configuration.
- 5. Click <u>Options > Window Setup > Load Setup</u> to load a window setup.

### **Exiting MPLAB**

At this point in the tutorial, you will exit, and then reopen MPLAB. After opening MPLAB again, you will load the window configuration that you saved in the steps above.

- Exit by double clicking the System Button in the top left corner of the MPLAB screen, or by choosing one of the following options for exiting MPLAB:
  - <u>File > Exit</u>
  - Alt-F4 (Close)
  - The System Button Close option
- 2. Double click on the MPLAB icon to reopen MPLAB.

## **Customizing Your Desktop**

You can use <u>Options > Window Setup > Load Setup</u> to load a predefined screen configuration that may include windows such as watch windows and source windows.

- 1. Click <u>Options > Window Setup > Load Setup</u> to load the SAMPLE.CFG window configuration that you saved in the previous exercise.
- 2. Select SAMPLE.CFG and click **OK**.

MPLAB loads the file, SAMPLE.ASM.

### **Customizing Your Development Environment**

The following steps guide you through changing the screen font, colors, and the tool bar.

### **Changing the MPLAB Screen Font**

The <u>Options > Environment Setup</u> allows you to change the screen font and point size.

Environme	ent Setup
Screen Font: ANSI OEM System Other Select	Toolbar:       ∠       Enable         ◆       Top         ◆       Bottom         ◇       Left         ◇       Right         ◇       Float
Global Switches: Status Bar Enable Clear Memory on Download Clear Breakpoints on Download Load User Configuration Global Break Enable Track Source Code	Address Labels: width = 9 + chars width = 9 + chars Cancel

Figure 3.7 Environment Setup Dialog – ANSI Screen Font

- 1. Click <u>Options > Environment Setup</u>, or press Ctrl+F7.
- 2. Select the ANSI screen font option and press OK.
- 3. Press Ctrl+F7 and select the Other screen font option. Click on Select.
- 4. Select one of the fonts, select a point size (4 points up to 24 points), and press **OK**.

## **Changing Colors**

The <u>Options > Colors</u> command allows you to change the color of visual cues used by MPLAB.

	Change Special Colors
Colors	Execution Point Background Execution Point Foreground Break Point Text Break + Irace Point Text Break + Trigger Point Text Break + Qualifier Point Text Break + Trace + Trigger Point Text Break + Trace + Qualifier Point Text Break + Trace + Trigger + Qualifier Point Text Trace Point Text Trace + Trigger Point Text Trace + Qualifier Point Text Trace + Qualifier Point Text Trace + Trigger + Qualifier Point Text Trigger Point Text Trigger Point Text Trigger + Qualifier Point Text Qualifier Point Text Status Bar Run Flag Register Window Column Titles
Cancel	Register Window Data Register Window Changed Data

Figure 3.8 Change Special Colors Dialog

- Click <u>Options > Colors</u> to display the Change Special Colors dialog box.
- 2. Click on **Status Bar Run Flag** to change the color of the status bar when the processor is running. The current color will be outlined with a black border.
- 3. Click on a new color to change the color.
- 4. With an executable file loaded in program memory, click the Run icon. The Status bar will change to the new color while the processor is running.

## **Customizing the Toolbar**

The <u>Options > Environment Setup > Toolbar Setup</u> command allows you to reconfigure any of the four toolbars. As an example, we will add a start-of-file command to the User Defined Toolbar.

- Click <u>Options > Environment Setup > Toolbar Setup</u> to display the Toolbar Setup dialog box. Select the User Defined Toolbar by one of the following procedures:
  - Use the Swap Toolbar icon at the extreme left of the toolbar, or
  - Use the Toolbar drop down-list

Toolbar Setup	
N <mark>o</mark> ux Jacke <b>(† 7</b> 24) 11 mm	<b></b>
Toolber User Defined Toolber  Button 1  Operation CursorStartOfFile	Button Size OF Automatic Can Large Medium
Icon BITMAP_25	) Smell

Figure 3.9 Toolbar Setup Dialog

2. Select a toolbar button to edit by clicking on the desired button.

Once selected, the dialog will display the function that is currently attached to the button in the Operation list, and the icon that is attached to the button in the Icon list. Make a note of both the operation and the icon to restore the button to its original state.

- 3. In the Operation list, select CursorStartOfFile, and in the Icon list select Bitmap\_26.
- 4. Click **OK**. The Toolbar setup dialog should close and display the Environment Setup dialog.
- 5. Click **Cancel** on the Environment Setup dialog.
- 6. Test the redefined button. Open a source file and page down a few times. Select the User Defined Toolbar, and press the button you changed. The current window goes immediately to the start of the file.
- 7. Restore the button you redefined to its original state.

### **Recovering a Hidden Window**

As you work with multiple windows, some windows will be hidden by other windows. MPLAB provides four ways to uncover hidden windows:

- Cascade
- Tile
- Currently Open Windows
- Drag Window

### Selecting from the List of Currently Open Windows

- 1. Click *File > Open Source*, select SAMPLE.ASM, and click **OK**.
- 2. Click *File > Open Source*, select TUTOR.ASM, and click **OK**.

3. Click the maximize button in the upper right corner of the TUTOR.ASM window.

The SAMPLE.ASM window is now hidden.

 Click <u>Window > SAMPLE.ASM</u> (from the list of Currently Open Windows) or press Ctrl+F6 to select and put the SAMPLE.ASM window on top.

### **Displaying Multiple Windows for Rapid Editing**

- 1. Click the maximize button in the upper right corner of the TUTOR.ASM window.
- 2. Click <u>Window > Tile Vertical</u>.
- 3. Use the mouse to widen the left and right boundaries of the two windows until you can see the full lines of text. The two windows will overlap as illustrated in Figure 3.10.
- 4. Alternately click in the left and right window.

				MPL	AB			
Elle Proje	ect <u>E</u> ditor	Debug Opti	ons Tools	Window	Help			
a 111		PC= 📩 🗰	A PEZ		- 0			
	ch	nplab/tutor/tu	tor.asm			c:/mplab/tutor	\sample.asm	
processor The Proce PIC16C5X; PIC16C62X PIC16C6X;		605%, PIC160 ted mrs: 6,57,58A 2 ,65,65A	XX, and PIC		*	BANPLE 8 Software Mu is stored in 1 ubrouting " mu 1 " mulplr ",	ltiplier	
The loop The loop After the After the Start ini Start ini Start ini Start ini	W	utes seven t ReduceA and ne time for s zeven tim s the progra A = 25 B = 1, Count = 7, W = 7 reg = 7	ines. DoubleB, each loop « es, the in: h counter t 5, (A decr: (B increa: (Count dec	truction to start. mases to 12 tes to 128) trements to	12 au	<pre># of cycles Scratch RAN inized for con code refer t ; prc16c54 ; 8 bit mul- ; 8 bit mul- ; 8 bit mul-</pre>	: O locations de efficiency ( looped code ) o "multSNSP.asm" ( straight li is the target processor tiplicand tiplier	ne oc
The sampl (Cycle # 7 0 7 2 7 3 7 4 7 5 7 5 7 5 7 5 7 5 7 5 7 5 7 5	<pre># program &lt;=</pre>	lculates val B (B=B+B) 1 2 4 8 16 32 64 128		lownt=Count-	•	; Low byte ; losp coun ; I/O regis ; STATUS reg	ter P6	

Figure 3.10 Using Multiple Windows



## Chapter 4. MPLAB Projects

## Introduction

The MPLAB Project Manager, in combination with other MPLAB tools, allows you to organize and control the development of firmware product designs.

Important: You must use the Project Manager in order to do symbolic debugging with MPLAB.

# Highlights

The MPLAB Project commands include:

- New Project
- Open Project
- Close Project
- Save Project
- Edit Project
- Make Project
- Build All
- Compile Single File
- Make Setup
- Most Recently Used Projects
- Drag and Drop Loading

**Note:** Projects reference fixed paths. Therefore, you cannot use the file manager to move projects. If you move the project source files to another directory, then you must create a new project.

## **New Project**

Select <u>Project > New Project</u> to open the New Project dialog. This allows you to create a project and save it in a directory that you specify. MPLAB assigns the extension .PJT to a project file as a default.

Create a new project as follows:

In the "Project Path and Name:" field, type

C:\MPLAB\TUTOR\TUTOR.PJT

and click OK.

If the directory doesn't exist, you will be asked: "Do you wish to create the Directory?"

If you need to create a directory for a new project, <u>Project > New Project</u> can create a directory for a new project. A new project does not contain any files. Use <u>Project > Edit Project</u> to add files to a project.

The title bar at the top of the desktop will display the path and name of the project that you enter. For example, if you enter

C:\MPLAB\TUTOR\TUTOR.PJT

then the following will display at the top of the desktop:

😑 New Project			
Project Path and Name:			
C:\MPLAB\TUTOR\TUTOR PJT	ОК		
Default Toolbar:	Cancel		
default.tbr	Browse		
Default Key Mapping:	<u>D</u> rowse		
mplab.key			
Development Mode:			
PICMASTER Emulator			

#### MPLAB - C:\MPLAB\TUTOR\TUTOR

#### Figure 4.1 New Project Dialog Box

Project Path and Name	New Project allows you to define files related to a project by entering the Drive, Directory, and Project Name.
Default Toolbar	This text box displays the default tool bar file name for the project. (Currently inactive)
Default Key Mapping	This text box displays the default key mapping file name for the project. (Currently inactive)
Development Mode	Selects the development mode for the project. (Emulator, Simulator, or Editor only) (Currently inactive)
<u>Browse</u>	Click <b>Browse</b> to choose a directory where you want the project to reside.
Cancel	Click <b>Cancel</b> to cancel a selection you make.

# **Chapter 4. MPLAB Projects**

Note:	You can also verify the project you just created from the File
	Manager. Look at C:\MPLAB\TUTOR. This directory contains
	two files:

- TUTOR.CFG Project Configuration File
- TUTOR.PJT Project File

**Note:** The <u>*Project* > *Edit Project*</u> dialog opens after clicking **Ok** in the New Project dialog.

# **Open Project (Ctrl+F2)**

Select *Project > Open Project* to open an existing project.

Upon opening a project, MPLAB will:

- Attempt to restore any previously open windows.
- Restore all break, trace and trigger points stored as part of project.
- Restore the compiler switches through command line options. Select <u>Project > Make Setup</u> to change MPASM or MPLAB-C command line options.
- Load binary code and symbolic information if possible.

Other files associated with project and saved in the project directory are:

- \*.CFG The screen configuration
- \*.COD The file containing symbolic information and object code plus other data.

## **Close Project**

Select *Project > Close Projec*t to close a project.

Upon closing a project, MPLAB will:

- Close all windows
- Unload symbols
- Remove all break, trace, trigger points associated with the project. Any break, trace, or trigger points set with the right mouse button will not be cleared.

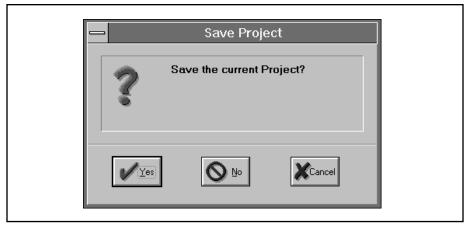


Figure 4.2 Save Project Dialog

Yes Answer Yes to save and close the current project.

<u>No</u> Answer **No** to close the project without saving.

<u>Cancel</u> Stops the close operation.

**Note:** Binary code will still be in the PICMASTER emulator memory after a project is closed.

# **Save Project**

Select <u>*Project > Save Project*</u> to save all information about the project and leave the project open.

Save Project saves the following information about the project:

- Screen Configuration
- Break, Trace and Trigger Points as Set in Respective Dialogs
- Compiler Switches
- Name of Source Files in the Project (not the file contents)

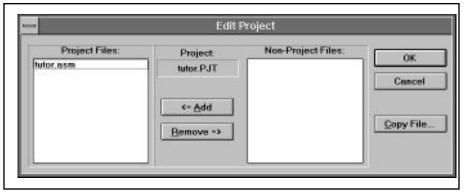
Save Project does not save binary code. You must use <u>*File > Export > Save</u></u><u><i>Hex File*</u> to save binary code if you have made changes directly to program memory.</u>

# Edit Project (Ctrl+F3)

Select <u>Project > Edit Project</u> to open a dialog for adding \*.ASM or \*.C files to a project or removing files from a project. Edit Project does not alter the original copy of the file that you add or remove from a project.

**Note:** MPLAB currently supports only one source file but allows multiple include files in projects by using the #include directive.

# **Chapter 4. MPLAB Projects**



#### Figure 4.3 Edit Project Dialog

<u>Add</u>	MPLAB assigns an *.ASM or *.C file into a project.
<u>Remove</u>	MPLAB removes the file from your project, but does not alter the file in any way.
<u>Copy File</u>	Opens a dialog box for selecting and copying a file to the project directory.
<u>OK</u>	Closes the Edit Project dialog, and saves any changes you have made.
<u>Cancel</u>	Cancels the operation.

# Make Project (F10)

<u>Project > Make Project</u> assembles the source code assigned to a project.

Make Project first checks for differences in the time and date of the source file you are assembling against the time and date of the \*.COD file. (The \*.COD file contains object code and symbolic information.)

**Note:** References to the assembler and compiler are interchangeable determined by whether you are using MPASM or a compatible C Compiler.

- If the Source file is newer than the \*.COD file, Make Project calls the assembler or C compiler to rebuild the object code.
- If the source is older than the \*.COD file, Make Project checks the time and date of any include files in the project.
- If you change an include file, causing the include file to become newer than the \*.COD file, Make Project will catch the change and force an update to the \*.COD file.
- If the \*.COD is more recent than any of the source files, no action will be taken.

### **Compiling Source Code**

When you select <u>*Project > Make Project*</u>, MPLAB calls MPASM or the C Compiler to build the source code.

If the code contains an error, MPLAB displays the status message:

_	Compile Status	
Status:		
There are er	ors.	
Command Lin	e:	
mpasmwin C:	\mplab\tutor\tutor.asm /q	
	ОК	

Figure 4.4 Failed Compilation

### Fixing a Compile Error

Begin correcting errors by fixing the first error first. Double click on the indicated error window to switch to the source code for editing. Use the MPLAB Editor to fix the error in the code. After fixing the error, select <u>*Project > Make Project*</u> again.

If successful, the status message reads:

-	Compile Status	
Status:		
Success, r	o errors.	
Command L	.ine:	
mpasmwin	C:\mplab\tutor\tutor.asm /q	
	ОК	

Figure 4.5 Successful Compilation

# **Chapter 4. MPLAB Projects**

# Build All (Ctrl+F10)

Click <u>Project > Build All</u> to rebuild all source files in the current project. Build All differs from Make Project in that Build All does not check time and date.

# Compile Single File (Alt+F10)

Select <u>Project > Compile Single File</u> to compile the file in the currently active window. The file must be either an \*.ASM or \*.C file. The file does not have to be assigned to a project. However, you must have a project open to access the Compile Single File function.

**Note:** When running Compile Single File, MPLAB does not reload the binary object code.

When running Make Project or Build All and the project compiles with no errors, MPLAB automatically reloads the object code from the \*.COD file.

## Make Setup

Select <u>*Project > Make Setup*</u> to change the switches for the command line options for MPASM and Byte Craft's MPC compiler.

Refer to the MPASM or MPC manuals for details about the command line option switches.

	Mak	e Setup	
Output Filename:		General Options	s:
calram.hex		Hex Output:	Generated Files:
MPASM Options: Default Radix:	Warning Level:	INHX8M	년 Error File 인 List File
i Hex □ Decimal □ Octal	All Messages     Warnings and Errors     Errors Only	MPC Options:	Cross Reference File
Case Sensit	ive 🗹 Macro Expansion	Compile to	Object Only
Extra Options:		Extra Options	:

Figure 4.6 Make Setup Dialog

# **Most Recently Used Projects**

MPLAB lists the Most Recently Used Projects at the bottom of the Project menu. Click on the name of the project that you wish to open.

# **Drag and Drop Loading**

Drag and drop loading provides a quick way to load files in a project. Use the drag and drop loading feature to drag projects (\*.PJT files) or source files (\*.C or \*.ASM files) from the File Manager and drop on the MPLAB icon, or on the MPLAB desktop.



# **MPLAB USER'S GUIDE**

## Chapter 5. MPLAB Basic Functions

## Introduction

This chapter discusses the basic MPLAB debugging functions. You can be in either the MPLAB-SIM simulator or the PICMASTER emulator mode to access debugging functions.

# Highlights

This chapter covers the following information:

- Real-Time Program Execution
- MPLAB-SIM Simulator Environment
- Simulator Considerations
- Break, Trace, and Trigger Points
- Conditional Break Dialog
- Special Windows
- Stimulus Functions
- File Extensions Used by MPLAB

## **MPLAB** Functions

After setting up and compiling projects in MPLAB, you'll want to see how your code runs. If you have a device programmer you can program a microcontroller device and plug the programmed device in your actual application to verify that the application runs as expected. Usually, an application will not run the correctly the first time, and you'll have to debug the code. You can use MPLAB-SIM to simulate your code or you can use the PICMASTER emulator to run your firmware in the actual application while you debug.

Either way, you will use break and trace points as you run your code. Look at register values in the Register window or Special Function Register window to see the processor's state as you run and single-step your code.

The PICMASTER emulator runs code at the actual execution speed (real time) on your target hardware, stopping only at specified breakpoints. MPLAB-SIM simulates the execution of any PIC16/17 and simulates I/O conditions at speeds that are dependent upon the speed of your PC.

The following debug functions work the same with the simulator or the emulator. The main functions are:

- Emulation Memory (Program Memory Window)
- Break, Trace Points

- Single-Stepping
- Register Monitoring (Special Function Register or File Register Windows)

All of these functions use information from an MPLAB project. Line labels in source code, symbolic locations in memory, and function names from code can be used to set break and trace points and to examine and modify registers.

# **Real-Time Program Execution**

In this document the term "real-time" is usually applicable only to the PICMASTER emulator.

### **Execution in MPLAB-SIM Simulator Mode**

When the system is said to be running in real time in the simulator mode, instructions are executing as quickly as possible by the software. This is usually slower than the actual PIC16/17 microcontroller would run at its rated clock speed.

The speed at which the simulator runs is dependent upon the speed of your computer and how many other tasks you have running in the background. The software simulator must update all of the simulated registers and RAM, as well as monitor I/O, set and clear flags, check for break and trace points in software, and simulate the PIC16/17 instruction with instructions being executed on your computer's CPU.

**Note:** Often loops will be used in your code to generate timing delays. When using the simulator, you might wish to decrease these time delays or conditionally remove those sections of your code with "IFDEF" statements to increase simulation speed.

In general when this manual says "real time" and you are in the simulator mode, this means that the software simulation is executing simulated PIC16/17 code as fast as your PC can simulate the instructions.

## Animate Mode

Animate Mode is a method of automatically single-stepping the processor. The simulator actually executes single steps while in Run mode, but it only updates the values of the registers when it is halted. To view the changing registers in the Special Function Register window or the Watch windows, use Animate mode. Animate mode runs slower than the Run function, but allows you to view changing register values.

# **MPLAB-SIM Simulator Environment**

MPLAB-SIM is a discrete-event simulator for the PIC16/17 microcontroller families and is integrated into the MPLAB IDE. The MPLAB-SIM simulator tool is designed to:

- Model operation of Microchip Technology's PIC16C5X, PIC16CXX, and PIC17CXX families of microcontrollers
- Assist users in debugging software that uses Microchip microcontroller devices.

A discrete-event simulator, as opposed to an in-circuit emulator (like the PICMASTER emulator) is designed to debug software. MPLAB-SIM allows you to modify object code and immediately re-execute, inject external stimuli to the simulated processor, and trace the execution of the object code. A simulator differs from an in-circuit emulator in three important areas:

- I/O timing
- Execution speed
- Cost

## I/O Timing

Stimulus is injected to MPLAB-SIM prior to the next instruction cycle. External timing in MPLAB-SIM is processed only once during each instruction cycle. Transient signals, such as a spikes on  $\overline{\text{MCLR}}$  smaller than an instruction cycle will not be simulated but may be caught by an in-circuit emulator. In MPLAB-SIM, external stimulus is injected just before the next instruction cycle.

## **Execution Speed**

The execution speed of a discrete-event software simulator is orders of magnitude less than a hardware oriented solution. Users may view slower execution speed as a handicap or as a tool. MPLAB-SIM attempts to provide the fastest possible simulation cycle, and depending upon the mode of operation, can operate on the order of milliseconds per instruction.

## Cost

Microchip Technology has developed the MPLAB-SIM simulator to be the most cost-effective tool for debugging application firmware. MPLAB-SIM does not require any external hardware to your PC, and operates in most respects exactly the same as the PICMASTER emulator. Unless you need to debug your application in real time on your actual hardware, MPLAB-SIM can usually be used to find and correct most coding errors.

## **Debugging Tool**

MPLAB-SIM is particularly suitable for optimizing algorithms. Unlike an emulator, the simulator makes many internal registers visible and can provide software tools that are difficult or expensive to implement in a hardware incircuit emulator. For the most part, MPLAB-SIM can be used to fully debug your system unless you run into situations where an in-circuit emulator is required.

# **Simulator Considerations**

MPLAB-SIM executes on instruction cycle boundaries, and resolutions shorter than one instruction cycle can not be simulated. MPLAB-SIM is a discrete-event simulator where all stimuli are evaluated and all response generated at instruction boundaries (or Tcy. One Tcy = 4 Tosc, where Tosc is the input clock period). Therefore, there are some physical events that can not be accurately simulated. These fall into the following categories:

- Purely asynchronous events
- Events that have periods shorter than one instruction cycle

In summary, the net result of instruction boundary simulation is that all events get synchronized at instruction boundaries, and events smaller than one instruction cycle are not recognized.

The following list itemizes the functions and peripherals among the entire PIC16/17 family of microcontrollers that are affected by simulation on instruction cycle boundaries:

- Clock pulse inputs smaller than one cycle can not be simulated even though timer prescalers are capable of accepting clock pulse inputs smaller than one cycle.
- PWM output pulse resolution less than one cycle is not supported.
- Compares greater than 8-bits are not supported.
- In unsynchronized counter mode, clock inputs smaller than one cycle can not be used.
- The oscillator waveform on RC0/RC1 pins can not be shown.

# **Break, Trace, and Trigger Points**

The debug functions affect execution of program instructions based upon the following elements:

- Breakpoints
- Trace Points
- Pass Counter Addresses

MPLAB limits the number of named address ranges to a maximum of 16 in each dialog.

All events get synchronized at instruction boundaries. Events smaller than one instruction cycle are not recognized.

# **Chapter 5. MPLAB Basic Functions**

Trace points and breakpoints function totally independent of each other, and you can set them at any program memory location.

The following figures show the dialog boxes for assigning names to address ranges.

Break	Point Settings
Title break1	<u>R</u> emove <u>C</u> lose
End Start	Remove <u>A</u> ll <u>D</u> isable All
	Address is Qualifier Pass Count 0 Set
	Break on Trace Buffer Full
	🗌 <u>E</u> nable External Break
	◆ <u>OR</u> with Internal Break ◇ A <u>N</u> D with Internal Break
	<ul> <li>♦ Rising Edge</li> <li>♦ Ealling Edge</li> </ul>

Figure 5.1 Break Settings Dialog Box

	Trace Point Settings
Title trace1	<u>R</u> emove <u>C</u> lose
Start	Remove <u>A</u> ll <u>D</u> isable All
	Address is Qualifier
	Pass Count 0 Set
	<u> </u>
	◆ <u>T</u> race All
	<ul> <li>Trace <u>F</u>irst Cycle Only</li> <li>Trace Data Cycle <u>O</u>nly</li> </ul>

Figure 5.2 Trace Settings Dialog Box

### **Real-Time Breakpoints**

A breakpoint is a condition in which the processor executes code and halts after a certain condition is met.

MPLAB provides the following ways to set a breakpoint:

- Break on Address Match
- Break on Trace Buffer Full
- Break on Pass Count Reached
- Break on Stack Overflow
- Break on Watch Dog Timer Time Out
- User Halt

The Program Memory Window shown in Figure shows the following information:

- B Breakpoints
- T Trace Points
- Q Pass Counter Addressesc

9		8000	3875	start	novler	0×FF	
10		0009	BADO		BOVWI	0x28	1
11		A080	3001		novler	0×1	
12 13		000B	00A9		BOVWI	0x29	
13		0800	3807		movler	0.x7	
14 15		0000	AABO		BOVWI	0x2A	
15		0808	0000		nop		
16 17 18 19			2013	losp		ReduceA	
17		0818	OBAA		decfaz	0×2A	
18		0011	280P		goto		
		0812	2808		goto	start	
20	B		0829	ReduceA		0x29,₩	
21. 22 23 24 25		0014	02.48		supet		
22		0015	2017		call	DoubleB	
23		0816	3400		retlar	0×0	
24			0829	DoubleB		0 x2 9, 4	
			07A9		addwf		
26			201B		11.00	extend3	
27		UUIA	3401		retly	0.80	

Figure 5.3 Program Memory Window

#### **Break on Address Match**

Break on Address Match allows you to halt the processor when the processor program counter equals a certain value. The processor breaks after the valid instruction is executed. For example, if a breakpoint is set at address 5Ah, then the processor breaks after executing the instruction at address 5Ah.

#### **Break on Trace Buffer Full**

MPLAB can be set to halt the processor after capturing 8K selected cycles (when the trace buffer is full).

# **Chapter 5. MPLAB Basic Functions**

#### **Break on Pass Counter Equal to Predefined Value**

MPLAB has a Pass Counter switch that you can assign to either trace logic or break logic. The pass counter can be used to break or trace after the processor executes an address a predefined number of times.

For example, if the Pass Counter is assigned to break logic, then when the pass counter decrements to zero, the pass counter acts as a breakpoint and halts the processor.

#### Break on Stack Overflow

Break on Stack Overflow causes MPLAB to execute a break when the stack overflows.

#### **Break on Watch Dog Timer**

If enabled, MPLAB executes a break when a Watch Dog Timer time-out generates a device reset.

#### **User Halt**

MPLAB IDE provides three ways to stop at a breakpoint any time the processor is running:

- Click <u>Debug > Run > Halt</u>
- Press F5
- Click the Halt Icon (red stop light)

## **Real-Time Trace Points**

A trace is a function that logs program execution. The MPLAB-SIM simulator and the PICMASTER emulator have an 8K real-time trace buffer that logs addresses and opcodes as they execute. This circular trace buffer continues logging data after the buffer is full, losing the oldest data (unless you have selected Break on Trace Buffer Full in the Breakpoint Settings dialog).

#### **Circular Trace Buffer**

MPLAB continuously captures selected bus cycles into the 8K trace buffer.

The status information captured into the trace buffer is grouped as follows:

- 16 Bits of Address
- 16 Bits of Opcode/Data
- Time Stamp and Changed Registers (MPLAB-SIM Only)
- 8 Bits of External Trace Probe Data (PICMASTER emulator Only)

-				Trace Men	iory Window 💌
1 0000 3 0000 3 0000 4 000A 5 0000 4 000A 5 0000 6 0000 9 0000 9 0000 9 0000 9 0000 9 0000 9 0000 10 0010 10 0000 10 00000 10 00000 10 0000 10 000	2009 3077 0014 3001 0015 3007 0014 0014 0014 0014 0014 0014 0014 0	stert Leep Redecak sutand extends extends extends extends extends	movet movet novet nop onll movt schot onll movt	start Outf Outf Out Out Out Out Out Out Out Out Out Out	1 4.000 1 19100 1 6.000 2 19179 1 0.000 2 120,977 1 0.000 2 120,977 1 0.000 2 120,970 03.10 1 10.000 2 10210 03.10 1 10.000 2 10210 1 10.000 2 1000 03.10 1 0.000 2 1000 1 1000 1 1000 1 1000 1 1 0.000 2 1000 1 1000 1 1000 1 1000 1 1000 1 1 0.000 2 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1 0.000 2 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 1000 1 10000 1 1000 1 10000 1 10000 1 1000 1 1000 1 1000 1
Address	) Opco	Label	Instruc	ction Data	Timing, changed registers and stack

Figure 5.4 Trace Memory Window

#### Halting Trace from the Tool Bar

Halt Trace allows you to take a snapshot of the trace buffer and look at the captured trace without halting the processor. In the Tool Bar, click **Halt Trace** to display a snapshot of the trace buffer without halting the processor. Once the trace buffer is halted, click **Halt Trace** again to take another trace snapshot.



Figure 5.5 Halt Trace Icon

#### **MPLAB-SIM Simulator Trace Display**

The trace window can be used to collect executed instructions from the MPLAB-SIM simulator. In addition to the data shown by the PICMASTER emulator, the simulator will show a time stamp on each line and will echo changes to registers. The time stamp uses the same data as the MPLAB Stop Watch. You can reset the time stamp by resetting the Stop Watch.

**Note:** Saving the trace display to a file in the simulator mode will produce even bigger files than the PICMASTER emulator saved trace files.

# **Chapter 5. MPLAB Basic Functions**

### Assigning a Pass Count to Break or Trace Points

MPLAB has a 16-bit Pass Counter that decrements by one on any address match in program memory.

When the processor is in a Halt state, you can modify the count value in the Break Point Settings or Trace Point settings dialog box. To set up the Pass Counter, first set the desired address ranges and then load the counter with a desired count value (up to 16 bits). When the counter decrements to zero, the emulator will halt.

#### Pass Counter Assigned to Break

If the Pass Counter is assigned to Break, the processor halts upon encountering a breakpoint (either internal or external conditions) or when the Pass Counter reaches zero.

Title 10K_Passes	<u>R</u> emove <u>C</u> lose
Start 500 End 500	Remove <u>A</u> ll <u>D</u> isable All
	Address is Qualifier
☑ q 10K_Passes 500 500	Pass Count 10000 Set
✓ Halt_Range 100 200	Break on Trace Buffer Full     Enable External Break
	<ul> <li>♦ <u>O</u>R with Internal Break</li> <li>◇ AND with Internal Break</li> </ul>
	♦ Rising Edge ♦ Ealling Edge

Figure 5.6 Breakpoint Settings Dialog

**Example:** This example shows breakpoints and pass counter addresses used in the same code. Keep in mind that breakpoints and pass counter addresses are independent of each other.

- 1. Set up a named breakpoint range from address 100 to 200.
- 2. Set a Pass Counter Address at 500.
- 3. Load the Pass Counter with a value of 10000.

The processor halts when executing in real-time if it executes any instructions within address range 100 to 200 or after it halts after executing 10000 instructions at address 500.

#### **Pass Counter Assigned to Trace**

If the Pass Counter is assigned to trace, then the real-time trace buffer does not capture data until the Pass Counter decrements down to zero. When the pass counter decrements to zero, the trace buffer starts capturing data on valid cycles.

#### **Using Pass Counter to Count Events**

The Pass Counter decrements each time an event occurs. You can use this feature to count how many times an event happens.

**Note:** The emulator must be halted before MPLAB will update the pass counter reading, on the status bar.

# **Conditional Break Dialog**

MPLAB halts at a breakpoint when the value of a specified internal register reaches a preset value or condition.

Con	itional Break	
Single Cycles     Multiple Cycles     Update Display Conditions: User Helt     Reg: 02a Value: 01		•
Z Trace Data	Start	Break Settings
Edit	Helt	Save Buffer
	Reset	Close

Figure 5.7 Conditional Break Dialog

#### Conditions

MPLAB will stop at a breakpoint in the Conditional Break dialog based on one of the following conditions:

- <u>User Halt</u> MPLAB executes until you press the Halt button on the Conditional Break display.
- <u>Number of Cycles</u> MPLAB halts after the target processor executes the specified number of cycles.
- Logic Condition satisfied.

#### **Trace Data**

Trace Data allows you to track the value of the registers in the Conditional Break dialog.

### Single Cycle

In the Single Cycle mode, MPLAB single steps the processor until the condition is met.

# **Chapter 5. MPLAB Basic Functions**

### **Multiple Cycles**

- Conditional Break executes instructions in real time (in the emulator), halts at user selected breakpoints, checks the specified condition and continues executing instructions in real time. The emulator or simulator only stop when meeting the specified condition.
- Breakpoints and register conditions are only checked at the breakpoints you specify in the Break Setting dialog.

## **Special Windows**

MPLAB provides windows for viewing various information.

### **Program Memory**

Use the system window control let to change how data is displayed in the window. The Program Memory Window displays the instructions that reside in the emulator memory space or in the simulator memory space.

## **Trace Memory**

The Trace Memory Window displays instructions captured as the target processor runs in real-time.

## **EEPROM Memory**

If the target processor has EEPROM data memory, it is displayed in this window.

## **Absolute Listing**

The Absolute Listing Window shows the list file generated by the assembler or compiler. The Absolute Listing displays source code with the generated object code.

## Stack

The Stack Window displays each address pushed on the target processor stack.

## **File Registers**

The File Registers Window displays a hex dump window of all the File Registers of the particular processor being emulated.

## **Special Function Registers**

The Special Function Registers Window displays the current values of the special function registers for the processor being emulated. Whenever a break occurs, MPLAB updates the contents of the Special Function Registers Window.

### **Show Symbol List**

The Show Symbol List Window displays the symbols from your source code. These symbols are from the \*.COD file in your project.

### Stopwatch

The Stopwatch Window allows you to measure time. Stopwatch runs when the target processor runs. You must manually enter the frequency of the target processor for the stopwatch to be accurate.

### **Custom Watch Windows**

Watch Windows display selected variables and registers. You can store and retrieve watch windows without disturbing the current setup.

## Modify

The Modify dialog lets you read and write a memory location or a range of memory locations. Modify can work on the following memory areas:

- Data
- Stack
- Program
- EEPROM (If available)

## **Stimulus Functions**

MPLAB-SIM simulator functions allow you to set up regular clock stimulus signals and allow the simulator to respond to events from files on your PC. The files can be written with the MPLAB Editor or any other suitable text editor or word processor and should be saved in the same directory as the current project.

## **Asynchronous Stimulus**

An asynchronous stimulus simulates asynchronous events such as signals on input lines and is useful for simulating external interrupts or resets. The inputs are of four types:

- High
- Low
- Toggle
- Pulse

An asynchronous event can only be injected while the simulator is in the "Run" mode. If it is stopped at a breakpoint, you must manually set or clear the pin on the appropriate register.

# **Chapter 5. MPLAB Basic Functions**

Stim 1 (P)         Stim 2 (P)         Stim 3 (P)         Stim 4 (P)           Stim 5 (P)         Stim 6 (P)         Stim 7 (P)         Stim 8 (P)           Stim 9 (P)         Stim 10 (P)         Stim 11 (P)         Stim12 (P)	— Asy	ynchronous	s Stimulus [	Dialog
	Stim 1 (P)	Stim 2 (P)	Stim 3 (P)	Stim 4 (P)
Stim 9 (P) Stim 10 (P) Stim 11 (P) Stim12 (P)	Stim 5 (P)	Stim 6 (P)	Stim 7 (P)	Stim 8 (P)
	Stim 9 (P)	Stim 10 (P)	Stim 11 (P)	Stim12 (P)



### **Pin Stimulus**

DOS MPSIM stimulus files are compatible with MPLAB-SIM.

Specified values are driven into specified pins at a specified simulation step or time. The specified values, pins, and steps are defined in a text file called a stimulus file. Stimulus can be injected either according to step or time. The maximum number of clock stimuli that can be assigned is 12.

The stimulus file forces simulated PIC16/17 MCU pins to values at a specified input step.

Use the <u>Debug>Simulator Stimulus</u> setup menu to open the stimulus file for the MPLAB-SIM simulator. When you run or single step, lines in the file will be executed when the stop watch matches the time or when the step number executes.

## **Clock Stimulus**

Stimulus Pin:		Clock Sequer	_	🗌 Invert
Pin: — Clock S	Sequence —			
Add Del	ete Open	Save	Close	Apply

Figure 5.9 Clock Stimulus

Clock inputs are synchronous events that can be assigned to any valid PIC16/17 MCU input pin. These inputs are described by the duty cycle of the PIC16/17 MCU clock waveform and the polarity (default: initial phase = high). The duty cycle values are in decimal.

### **Register Stimulus**

This function inserts the next value from a DOS text file into a PIC16/17 microcontroller file register when the program counter equals the listed address.

- Register Stimul	lus
Selected File: c:\work\mplab\test.reg Program Memory Address: 0x56 Register Address: adres	<u>OK</u> <u>C</u> ancel <u>B</u> rowse

Figure 5.10 Register Stimulus Dialog

This is useful when simulating devices such as the PIC16C71 and PIC16C74. Both of these devices have A/D converters (among other peripheral modules). The MPLAB-SIM simulator does not perform an A/D conversion, although the interrupt that can be generated upon its completion is supported in the software. Register Stimulus Files allow you to inject values into a register when a certain point in program memory is reached. For example, if the target processor is the PIC16C71, you could set up your source code to branch to the interrupt vector at the end of conversion and inject a value into the ADRES register when the program counter reaches the address of the interrupt service routine.

# **Chapter 5. MPLAB Basic Functions**

## File Extensions Used by MPLAB

The default extensions of files used by the MPLAB are listed below:

- \*.COD Contains symbolic information and object code.
- \*.PJT Contains most of the information related to a project
- \*.C C Source File
- \*.H C Include File
- \*.ASM Assembly Language Source File
- \*.INC Assembly Language Include File
- \*.HEX PIC16/17 Machine Code in Hex Format
- \*.CFG Configuration/Setup Files
- \*.TRC Trace Save Files
- \*.TBR Toolbar File
- \*.LST Absolute listing file generated by assembler/compiler.
- \*.ERR Error File generated by assembler/compiler
- \*.WAT Watch Window File
- \*.TPL Template File
- \*.KEY MPLAB Key Mappings
- \*.TB Conditional Break Trace File
- \*.STI Stimulus Pin File
- \*.REG Stimulus Register File



# **MPLAB USER'S GUIDE**

# Chapter 6. MPLAB Menu Options

## Introduction

This chapter gives detailed information on using the MPLAB menu options. The chapter organization follows the entries on the pull down menus.

# **Highlights**

This chapter will discuss the following:

- MPLAB Functions
- MPLAB Desktop
- File Menu
- Project Menu
- Editor Menu
- Debug Menu
- Options Menu
- Tools Menu
- Window Menu
- Help Menu

# **MPLAB USER'S GUIDE**

## **MPLAB** Functions

#### FILE New Source **Open Source** View Save Save As Save All Close Close All Import Download to Memory Download to Target Memory Copy from Target Memory Export Save Trace Buffer Save Hex File Print Print Setup Exit Most Recently Used Files PROJECT New Project **Open Project Close Project** Save Project Edit Project Make Project **Build All Compile Single File** Make Setup Add/Modify Compiler/Assembler Most Recently Used Projects EDITOR Undo Typing Sequence Cut Сору Paste Select All Select Word **Delete Line** Delete EOL Goto Line Find Replace Repeat Find **Repeat Replace** Match Brace

Template Attach File **Detach File** Create File Save File Insert Edit New Store Store As Delete Insert Mark Find Mark Text Transpose Upper Case Lower Case Indent **Un-Indent** DEBUG Run Run Reset Halt Halt Trace Animate Step Step Over Update All Registers Change Program Counter Execute Execute an Opcode **Conditional Break** Simulator Stimulus Asynchronous Stimulus **Pin Stimulus Clock Stimulus Register Stimulus** Center Debug Location **Break Settings Trace Settings Trigger Out Settings Clear All Points** Clear Program Memory System Reset Power-On-Reset

#### OPTIONS

**Development Mode** Window Setup Save Setup Load Setup **Default Configuration Default Editor Modes** Current Editor Modes **Reset Editor Modes Key Mappings Environment Setup** Colors Processor Setup Hardware Multi-Processor TOOLS DOS Command to Window Repeat DOS Command to Window Verify PICMASTER **Configure Probe** WINDOW Program Memory Trace Memory **EEPROM Memory** Absolute Listing Stack **File Registers Special Function Registers Calibration Data** Show Symbol List Stopwatch Proiect New Watch Window Load Watch Window Modify **Tile Horizontal Tile Vertical** Cascade Iconize All Arrange Icons **Open Windows** HELP **Release Notes Getting Started** MPLAB Help Editor Help MPASM Help About

# **Chapter 6. MPLAB Menu Options**

## MPLAB Desktop

The MPLAB desktop is a resizable window that operates independently of the rest of the menu items. You can maximize the desktop to provide a full screen view by clicking the left mouse button on the up-arrow in the upper right hand corner of the desktop.

The following figure shows a maximized desktop. The up arrow replaces the up-down arrow combination. Click on the up-down arrow combination to reduce the window and size it manually.

MPLAB	• •
Elle Braject Editor Debug Options Tools Window Help	1011-0
の 理由 花園氏 七国国 竹屋 昭和 四面	
10004 ME10214 pe040 -2000 3-0h Dr (4413 Debap	

Figure 6.1 MPLAB Desktop

MPLAB dialog boxes behave as normal Windows applications and allow you to access standard Windows functions (as well as MPLAB-specific functions) through the MS Windows system button in the upper left hand corner. Other standard Windows features include window size buttons, icon buttons, vertical and horizontal scroll bars, and elevator buttons.

All MPLAB functions are accessible through the menu bar located across the top of the desktop. MPLAB menus that pull down from the menu bar allow you to access the emulator functions. Underlined characters on the pull down menus are key accelerators. The key accelerators enable the named function when the menu is pulled down. For example, if the File Menu is pulled down, pressing **O** enables Open Source.

When not using MPLAB, you can iconize the window by clicking the minimize button.

#### **Toolbars**

For your convenience, MPLAB contains four toolbars to provide you with shortcuts for performing routine tasks. The four toolbars are:

- Edit Toolbar
- Debug Toolbar
- Project Toolbar
- User Defined Toolbar

The toolbar icons can be reconfigured by the user to pertain to their specific needs.



Figure 6.2 MPLAB Debug Toolbar

#### **Status Bar**

The table below describes the information presented in the status bar located at the bottom of the screen.

Table 6.3 MPLAB Status Bar

Title	Typical Entry	Description	Result from Double Clicking
Line No., Column – Windows Open	Ln 1 Col 1	Displays current line number and column in file.	Opens Goto Line Dialog
MPLAB Version Number – No Open Windows	2.00.00	Current Version Number	No Action
Lines in File	72	Displays number of lines in current text file.	No Action
File Modified	#	Displays # Symbol if File Has Been Changed Since Opening	No Action
Write/Read Only	WR	Displays Write/Read Only Status.	Toggles between write and read only.
		WR = Editable File	
		RO = Read Only File	
Text Wrap	No Wrap	Displays current wrap mode and wrap column if text wrap is on.	Toggles between wrap and no wrap. No Wrap
		Example 1:NoWrap	Wrap at Column 72
		Example 2:WR 72	
		Useful for text files. Use <u>Options &gt; Current</u> <u>Editor Modes</u> to change wrap column.	

Title	Typical Entry	Description	Result from Double Clicking
Insert/Strikeover	INS	Toggles typing mode between insert and strikeover.	Toggles between INS and OVR.
		INS = Insert Characters	
		OVR = Type over characters	
Current Processor	PIC16C61	Displays the currently selected processor.	No Action
Current Program Counter	pc:0x5f	Displays the current program counter.	Opens Change Program Counter dialog.
Current w Register Value	W:0x00	Displays current w register value.	No Action
Status Bits	ov Z dc c	Upper Case = Set(1)	No Action
		Lower Case = Reset(0)	
Global Break Enable	Bk On	Displays current status of Global Break Enable.	Toggles Global Break Enable On and Off.
Current Development Mode	Em	Displays Current Development Mode:	Displays Development
		Em = Emulator	Mode Dialog.
		Sim = Simulator	
		EO = Editor Only	
Current Pass Count	65533	Displays current pass count.	No Action
		Maximum value = 65534	
Current Tool Bar	Edit	Displays current tool bar	No Action

Table 6.3 MPLAB Status Bar (Continued)

Ln 100 Col 2	2048	RO	No Wrap INS	PIC16C622 pc:0x58	w:0x00 z do o	Bk On Em 0	Debug

Figure 6.4 Status Bar

# File Menu

#### Import

The <u>*File > Import*</u> functions allow you to move data from a PC file to the emulator and into target memory or into the simulator memory. This function also allows you to transfer data from the target into emulator memory for debugging.

#### **Download to Memory**

Select <u>File > Import > Download to Memory</u> to display the dialog box for selecting a file to download. The file you select is downloaded to the PICMASTER Memory or simulator memory. The file must be a valid hex file.

#### PIC16C5X Series and PIC16CXX Series

If the current target processor belongs to the PIC16C5X device family or the PIC16CXX device family, then the file should be in the Intel<sup>®</sup> 8-bit HEX file format (INHX8M). If the object file is successfully downloaded, then the symbols are loaded automatically from the corresponding \*.COD file if it exists. The default extension for the object code is \*.HEX.

#### PIC17CXX Series

If the current target processor belongs to the PIC17CXX device family, then the file should be in the Intel extended HEX file format (INHX32). If the object file is successfully downloaded, then the symbols are loaded automatically from the file \*.COD. The default extension for the object code file is .HEX.

If you do not have a \*.COD file, you may want to turn off source tracking in the <u>Options > Environment Setup</u> dialog.

Note: Use INHX32 if your application addresses memory beyond 32K.

### Export

#### Save Trace Buffer

Select <u>*File > Export > Save Trace Buffer*</u> to display the Save Trace Buffer dialog box. The emulator real-time hardware trace buffer (8K x 40) or simulator trace buffer is saved to the selected file.

**Note:** Saving the complete trace buffer as it appears in the Trace Window (address, data, disassembled code and external logic probe lines) requires about 700K bytes of disk space for the PICMASTER emulator and can exceed 1 M for the MPLAB-SIM simulator!

# **Chapter 6. MPLAB Menu Options**

	Save File As	
File <u>N</u> ame: file0000.trc	Directories: a:\ a:\ a:\ b a:\ b d d d d d d d d d d d d d d d d d d d	OK Cancel
List Files of <u>Type:</u> All Files (*.*)	Drives: a:	
Data Range <ul> <li>All</li> <li>◇ Range</li> <li>Start 0</li> <li>End 8192</li> </ul>	Filter Data	

Figure 6.5 Save Trace Buffer Dialog

<u>Range</u>	Select the range (0 to 8191) of the trace buffer that you want to save. Enter the desired value for the Start line number and for the End line number.
Filter Data: All	Writes the complete trace buffer to the selected file.
Filter Data: Opcode (Data Only)	Saves Opcode / Data only. (For PIC17CXX external read/write cycles)
	Available only with the PICMASTER emulator
Filter Data: Address Only	Saves Address only.

#### Save Hex File

Select <u>File > Export > Save Hex File</u> to display the Save Hex File dialog box.

-	Save Hex File	
File <u>N</u> ame:	<u>D</u> irectories:	ОК
*.hex	a:\	Cancel
÷	🗁 a:\ 🔹	
4		
*		
List Files of <u>T</u> ype:	Dri <u>v</u> es:	
Hex Files (*.hex) 👲	⊑ a:	
♦ All	<ul> <li>Disassembled Code</li> <li>Hex File</li> </ul>	
♦ Range		
Start End	High Byte Only	
	<ul> <li>♦ High and Low Bytes</li> <li>♦ Low Byte Only</li> </ul>	



All Saves the entire file.

<u>Range</u>	Enter a HEX address range to save by typing in the start and end address.
Disassembled Code	Saves the program memory instructions in a DOS text format which can be edited and used for documentation or for an MPASM source file.
HEX File	Saves the current program memory to a disk file in the Intel HEX format.
High Byte Only	Saves all high bytes of the selected range. This option is useful for burning 8-bit EPROMs (for processors with external execution).
High and Low Bytes	Saves the field with a 16-bit opcode (low byte followed by high byte).
Low Byte Only	Saves all low bytes of the selected range. This option is useful for burning 8-bit EPROMs (for processors with external execution).

## Print (Ctrl+P)

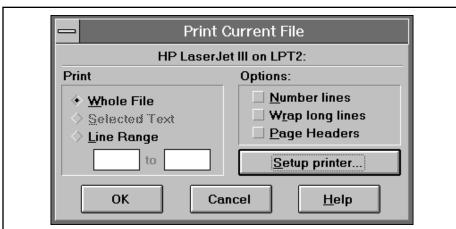
Click  $\underline{File > Print}$  to print some or all of the current file on your currently selected printer.

The Print command starts a dialog to specify the details of how the file is to be printed and on which printer.

By default, MPLAB uses the same printer that you specified the last time you printed a file. The default details of the printing, such as line folding and page headers, are taken from the window modes set on the current window.

The dialog box shows you the name of the printer that MPLAB is currently using—if you haven't specified otherwise, this will be your system default printer.

The file is printed using the same tab width that is being used in the window, and with line numbers if you have line numbering turned on. You can configure both these settings with the <u>Options > Current Editor Modes</u> menu command.



#### **Print Current File Options**

#### Figure 6.7 Print Current File Dialog

Setup Printer	Click the Setup Printer button to change printers, printer font, or page margins.
Whole File	Click Whole File (the default) to print entire file.
Selected Text	Click Selected Text to print only the text that is highlighted. This option will not be available if you don't have any text highlighted.
<u>Line Range</u>	Click Line Range and fill in the start and end line numbers to print a range of lines. You can use the words "start" and "end" to represent the first and last lines of the file.
Number Lines	Select Number Lines to print lines with numbering.
Wrap Long Lines	Select Wrap Long Lines to fold lines too long to fit the page rather than being truncated.
Page Headers	Select Page Headers to start each page with a header giving the file name and other information.

#### **Print Setup**

Click <u>*File > Print Setup*</u> to set up details of the printer that MPLAB will use, run the printer's setup dialog, and select the font. MPLAB records the values you set with the Print Setup option for the selected printer. Thus, you can have different settings for different printers. These values become the defaults.

	Printer Setup	
<u>Available printers:</u> HP DeskJet 560C F HP LaserJet III on L Intel 144FC2-Fax or PostScript (QMS) o SmartLink 19200-Fa	PT2: COM1: n LPT1:	<u>Setup</u> <u>F</u> ont
Print options	Ţop Margin <u>B</u> ottom Margin	OK <u>H</u> elp

#### Figure 6.8 Printer Setup Dialog

*File > Print Setup* lets you specify:

- Which printer to use
- What page margins are to be applied when you print on this device
- What font to use with this printer

<u>*File > Print Setup*</u> also allows you to run the printer's own setup dialog to set device-specific information.

Available Printers	If you want to use a different printer than the one that is highlighted in the Available Printers list, scroll the list and click the left button on the printer name.
Print Options	To change the margins used on each page, check or uncheck the boxes in the Print Options area.
<u>Setup</u>	To run the highlighted printer's own setup dialog, click <b>Setup</b> .
Font	To change the printer font that MPLAB will use for the highlighted printer, click <b>Font</b> .

Because MPLAB is a text editor rather than a word processor, you're restricted to choosing fixed pitch fonts, where the characters are all the same width.

If you run the printer's own setup dialog after selecting a font, you may find that the font is no longer available; some printers offer different fonts in different operating modes.

## Exit (Alt+F4)

Click <u>*File > Exit*</u> to terminate your MPLAB session.

This command will end your MPLAB session. If any of the files you're working with have changed and you haven't saved the changes to disk, MPLAB will prompt you for each one in turn. You can choose to save the changes, discard them, or cancel the operation. You will also be prompted to save the current project.

## **Project Menu**

Chapter 4, "MPLAB Projects," describes the Project Menu items in detail.

# **Debug Menu**

## Run

The Run menu options allow you to control the execution of your firmware in the target processor.

#### Run (F9)

<u>Debug > Run > Run</u> takes the processor out of the halt state and puts the processor into execution until a breakpoint is encountered or until Halt is pressed.</u>

Execution starts at the current program counter (as displayed in the status bar). The current program counter location is also highlighted in the Program Memory window. While the processor is running, the Step and Run buttons are disabled.

#### Reset (F6)

<u>Debug > Run > Reset</u> issues a reset sequence to the target processor. This issues an MCLR to reset the Program Counter to the reset vector. If the processor is running it will continue running from the reset vector address.</u>

#### Halt (F5)

<u>Debug > Run > Halt</u> forces the processor into the halt state. When you click **Halt**, the processor is forced into a Halt state (Program Counter is stopped) and the processor status information is updated.

#### Halt Trace (Shift+F5)

<u>Debug > Run > Halt Trace</u> halts the trace buffer from capturing data but allows the processor to continue running. When the processor runs, Halt Trace is enabled.</u>

Halt Trace is useful in applications where the processor cannot be halted, but where you want to observe the program trace while debugging. A typical use for Halt Trace could be associated with running a brushless DC motor using PWM outputs.

Pressing Halt Trace causes the following events:

- 1. The Trace Buffer stops capturing data. Then the Trace Display window updates to display the contents of the trace buffer up to the time it halted.
- 2. The Processor continues to run.

#### Animate

<u>Debug > Run > Animate</u> causes the simulator to actually execute single steps while in run mode, updating the values of the registers as it runs.

Use Animate mode to view the changing registers in the Special Function Register window or in the Watch windows. Animate mode runs slower than the Run function, but allows you to view changing register values.

#### Step (F7)

<u>Debug > Run > Step</u> single steps the processor. This command executes one processor instruction (single or multiple cycle instructions) and then puts the processor back into halt state. After execution of one instruction, all the windows are updated with the current state of the processor. While the processor runs in real time, MPLAB ignores the Step button.

#### Step Over (F8)

Select <u>Debug > Run > Step Over</u> to execute the instruction at the current program counter location. At a call instruction, Step Over executes the called subroutine, and halts at the address following the call.

#### **Change Program Counter**

<u>Debug > Run > Change Program Counter</u> allows you to change the current program counter.

PC	Change Close			

Figure 6.9 Change Program Counter Dialog

# **Chapter 6. MPLAB Menu Options**

<u>PC</u>	Enter desired Program Counter address.
<u>Change</u>	Click <b>Change</b> to change to the new program counter address. The processor must be halted for the change to take effect.
<u>Close</u>	Exits from the Change Program Counter dialog box.

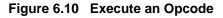
### Execute

The Execute menu options allow you to control the polled execution of your firmware in the target processor.

#### **Execute an Opcode**

Select <u>Debug > Execute > Execute an Opcode</u> to execute a single instruction or a series of instructions without modifying the object code or program memory. After executing the instruction, you may resume executing from the current program memory location.

Opcode	Execute Close			



- <u>Opcode</u> Enter the instruction as an opcode (in hex digits) or enter a symbolic instruction (such as ADDWF 0x19).Click the Opcode list to display the last eight commands. After executing a command, MPLAB highlights the command so you can type in a new instruction. (MPLAB tracks the instructions you enter so you don't get two copies of the same instruction in the opcode list.)
- <u>Execute</u> Click **Execute** to execute an instruction without modifying the current location of the program counter.

#### **Conditional Break**

Select <u>Debug > Execute > Conditional Break</u> to display a dialog box that performs an automated single stepping of the processor. Execution starts upon pressing the **Start** button and continues executing until meeting the condition presented in the dialog or until you click **Halt**.

# **MPLAB USER'S GUIDE**

	Conditional Break		
Single Cycles     Multiple Cycles			
🗹 Update Display			
Conditions: User Halt Reg: 02a Value: 01			
🗹 Trace Data		Start	Break Settings
Edit		Halt	Save Buffer

#### Figure 6.11 Conditional Break Dialog

Single Cycle	Checks condition at every instruction. The Single Cycle option samples conditions after every instruction, allowing you to catch a particular condition.		
<u>Multiple</u> Cycles	Checks condition only at user-defined breakpoints. The Multiple Cycles option runs at real time except for the halt at breakpoints. This option allows interrupts to be serviced.		
<u>Update</u> Display	Executes the conditional break but does not update the disassembled code in the window. MPLAB stores the last 1000 lines.		
Conditions	The condition that you set up is tested on any register location and an 8-bit constant value that you enter. You can test for the following conditions:		
	1.	<u>User Halt</u> When processor is running, click <b>h</b> processor.	Halt to stop the
	2.	<u>Number of Cycles</u> Enter the number of cycle box.	es in the <b>Value</b>
	3.	Register Value Conditions	
		RAM Addr Data Value == Equals	Value Entered
		RAM Addr Data Value <> Not Equals	Value Entered
		RAM Addr Data Value > Greater Than	Value Entered
		RAM Addr Data Value < Less Than	Value Entered
		RAM Addr Data Value >= Greater or Equal	Value Entered
		RAM Addr Data Value <= Less or Equal	Value Entered

If the tested condition is true, execution stops before executing the next instruction. The next instruction in the Program Memory window will be highlighted.

- **Caution:** All register values are treated as 8-bit unsigned values. Therefore, the condition <0 will never be true.
- RegRegister Condition Enter a RAM addresslocation where you want to test against the datavalue at that location. The location that you entermust be a file register location.
- <u>Value</u> Enter an 8-bit value in the Value box that you want to test against.
- Trace DataTrace Data samples specific registers at each time the<br/>processor is halted and displays the register data in the list.EditEdits the list of data variables sampled at each breakpoint.
- <u>Starts</u> Starts execution and continues to execute single steps until the condition is met or until you press the **Halt** button.
- Halt Halts execution of the Conditional Break.
- Resets the processor.
- Break Opens the Breakpoint Settings dialog.

<u>Settings</u> <u>Save Buffer</u> Opens the Save File dialog to save information from the list box in a \*.TB file.

<u>Close</u>

Exits out of the Conditional Break dialog.

-				c:tutor.tb	•
1065	3400		retlw Ox0		
			10 Count rethw OxO	01	1
Areg		to Breg	10 Count decfsz 0x2A	01	
Areg		to Breg	10 Count gets losp	04	
			10 Count call ReduceA	03	
Areg		tO Breg	10 Count awapf 0x29		
Areg BOSC	0129	tO Breg	10 Count swmpf Ox29,W	03	
Areg		to Breg	01 Count swapf 0x29	03	
			01 Count	03	

Figure 6.12 \*.TB File

## **Simulator Stimulus**

MPLAB-SIM simulator functions allow you to set up regular clock stimulus signals and allow the simulator to respond to events from files on your PC. The files can be written with the MPLAB Editor or any other suitable text editor or word processor and should be saved in the same directory as the current project.

#### **Asynchronous Stimulus**

The asynchronous command opens a dialog to simulate asynchronous events through a special window. In this manner you can simulate external interrupts or resets.

An asynchronous stimulus simulates asynchronous events such as signals on input lines and is useful for simulating external interrupts or resets. The inputs are of four types:

- High
- Low
- Toggle
- Pulse

An asynchronous event can only be injected while the simulator is in the "Run" mode. If it is stopped at a breakpoint, you must manually set or clear the pin on the appropriate register.

— Asy	ynchronou	s Stimulus I	Dialog
Stim 1 (P)	Stim 2 (P)	Stim 3 (P)	Stim 4 (P)
Stim 5 (P)	Stim 6 (P)	Stim 7 (P)	Stim 8 (P)
Stim 9 (P)	Stim 10 (P)	Stim 11 (P)	Stim12 (P)



#### **Pin Stimulus**

DOS MPSIM stimulus files are compatible with MPLAB-SIM.

Specified values are driven into specified pins at a specified simulation step or time. The specified values, pins, and steps are defined in a text file called a stimulus file. Stimulus can be injected either according to step or time. The maximum number of clock stimuli that can be assigned is 12.

! Sti	mulus	file	for	SAMPI	E.ASN	4				
STEP	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	!	PortB
Pins										
3	0	0	0	0	1	0	0	1	!	9 x 5
5	0	0	0	0	0	1	0	1		
65	0	0	0	0	1	0	1	0	!	10 x 5
67	0	0	0	0	0	1	0	1		
127	0	0	0	1	1	0	1	1	!	27 x 3
129	0	0	0	0	0	0	1	1		
191	0	0	0	1	0	0	0	1	!	17 x 7
193	0	0	0	0	0	1	1	1		
253	0	1	0	0	0	0	0	0	!	64 x 63
255	0	0	1	1	1	1	1	1		

#### Figure 6.14 File Input for Pin Stimulus

The stimulus file allows you to schedule bit manipulation by forcing simulated PIC16/17 device pins to given values at a specified input step.

Use the <u>Debug > Simulator Stimulus</u> menu to open the stimulus file for the MPLAB-SIM simulator. When you run or single step, lines in the file will be executed when the stop watch matches the time or when the instruction step number executes.

The first line of stimulus file always consists of column headings. It lists first the word "STEP," followed by the pins that are to be manipulated. The data below STEP represents the object file's input request occurrence. The data below each pin name is the input value. You may enter comments at the end of a line by preceding it with an exclamation mark (!).

The default injection point is "step".

#### **Clock Stimulus**

You can assign a repetitive clock to an I/O pin, defining the period of the clock by stating the number of cycles that the pin should be high, and the number of cycles that it should be low.

Pin: — Clock Sequence —	Stimulus Pin: ± High:	- Clock Sequence	Invert
	Pin: — Clock Sequence -		
			9

Figure 6.15 Clock Stimulus

Clock inputs are synchronous events that can be assigned to any valid PIC16/17 MCU input pin. These inputs are described by the duty cycle of the PIC16/17 MCU clock waveform and the polarity (default: initial phase = high). The duty cycle values are in decimal.

You can assign a repetitive clock to an I/O pin, defining the period of the clock by stating the number of cycles that the pin should be high, and the number of cycles that it should be low.

<u>Stimulus Pin</u> :	Assigns a pin for the clock stimulus.
<u>Clock Sequence – High</u> :	The number of cycles in which the clock is high.
<u>Clock Sequence – Low:</u>	The number of cycles in which the clock is low.
Invert:	Start clock with low cycle.
Add:	Add clock to list of active clocks as described in stimulus pin and clock sequence.
<u>Delete</u> :	Delete selected clock from list.
<u>Open</u> :	Open file of previously defined clocks.
<u>Save</u> :	Save current list of clocks to a file.
<u>Close</u> :	Close this dialog.
Apply:	Activate the selected clock.

### **Register Stimulus**

This function inserts the next value from a DOS text file into a PIC16/17 device file register when the program counter equals the listed address.

Register St	imulus
Selected File:	<u>OK</u>
c:\work\mplab\test.reg Program Memory Address:	Cance
0x56	
Register Address:	Browse
portb	1

Figure 6.16 Register Stimulus Dialog

This is useful when simulating devices such as the PIC16C71 and PIC16C74. Both of these devices have A/D converters (among other peripheral modules). The MPLAB-SIM simulator does not perform an A/D conversion, although the interrupt that can be generated upon its completion is supported in the software. Register Stimulus Files allow you to inject values into a register when a certain point in program memory is reached. For example, if the target processor is the PIC16C71, you could set up your source code to branch to the interrupt vector at the end of conversion and inject a value into the ADRES register when the program counter reaches the address of the interrupt service routine.

When the Program Counter equals the interrupt vector (program memory address 0x56), inject the next value in the file (ADVAL.TXT) into the ADRES register (file register address 0x09).

The format of the ADVAL.TXT file is one hex value on each line. For example:

- 0xAA 0X55 0XAA
- 0XAA 0X55

...and so on.

## **Center Debug Location**

Select <u>Debug > Center Debug Location</u> to move the current program counter to the middle of the debugging window.

This function works on the Source Code Window, the Program Memory Window, and the Absolute Listing Window.

## **Break Settings**

Select <u>*Debug > Break Settings*</u> to display the Breakpoint Settings dialog box for defining up to 16 named breakpoint ranges.

After entering a breakpoint title, start address, and end address, click  $\checkmark$  or press **Enter** to accept the breakpoint range definition.

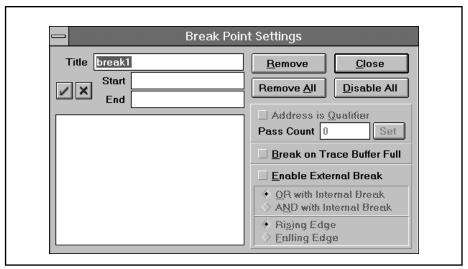


Figure 6.17 Break Settings Dialog

### Saving Breakpoint Settings

Breakpoints are saved as part of the project.

### **Breakpoint Range Definition**

- TitleEnter a unique title (up to 32 characters) for each breakpoint<br/>range. MPLAB accepts the underscore character but does<br/>not allow spaces.MPLAB automatically enters a default, unique title if you<br/>choose to not enter a title. The breakpoint range requires a<br/>title.
- <u>Start, End</u> Enter a Start and End address in hex or as a label for the breakpoint range. The address range is restricted to the valid address range of the target processor. You can just enter the start address and MPLAB will fill in the same end address for a breakpoint on a single location (rather than a range).

You can enter the Start and End values as addresses or labels. If you use labels, MPLAB allows you to modify the labels by using offsets: "MAIN+2", EXECTIMR-10". when you use labels and recompile a project (and the label moves due to the compilation) MPLAB assigns the breakpoints to the new address range.

You can use an existing breakpoint range item as the starting point for entering a new breakpoint range. Click on a desired item in the list box. Type in a new title. Then click  $\checkmark$  or press **Enter** to accept the defined breakpoint range.

#### **Breakpoint Settings**

- ✓ Click the check box to accept the selected range and add the range to the list of ranges. Entering a range enables breakpoints in that range in program memory. Disabled breakpoint ranges will not clear breakpoints included in other ranges.
- <u>X</u> Click X to not add the selected range to the list and to clear the Title, Start, and End fields.
- Breakpoint List The list box allows you to enter up to 16 breakpoint ranges. On selecting a range, the Breakpoint Settings dialog box displays Start, End, and Title to allow you to edit the start and end address. The list box contains the following elements:

v break1	start	start	
✓ break3	Reduce	ADoubleB	
🖌 q end break	65	65	
🗾 middle brea	k start+2	loop-2	

Figure 6.18 Breakpoint List

	<u>Enable</u>	This switch enables the breakpoint range in program memory.			
	đ	Qualifier point designator. When a range item displays the letter q, the address range is a Pass Count address.			
	<u>Title, Start, End</u>	Click an item in the list box to display the title, start, and end data. You can then edit the start and end address or change the title to enter a new breakpoint range.			
Remove	Removes the selected breakpoint range. If an item is not selected, Remove does nothing.				
Remove All	Removes all breakpoint ranges from the list.				
<u>Close</u>	Closes the Break	point Settings dialog box.			
Disable All	Disables all brea	kpoint ranges.			

#### **Pass Counter**

<u>Address is</u> <u>Qualifier</u>	You can assign a Pass Count qualifier address to either break logic or trace logic.
	You must have an element in the list selected. With an element selected, the dialog will enable the Address is Qualifier check box.
	When you check Address is Qualifier, you are assigning the pass counter to all addresses in the selected range. Usually it is useful to set only a single address for the pass counter. After selecting Address is Qualifier, MPLAB enables the Pass Count Edit Box and the Set button. With the Pass Count Edit Box enabled, you can set the pass counter to a desired value (up to 65,534).
Pass Count	Type in a pass count value. The pass count value defines the number of times the program can pass a qualifier address before halting the processor. Each time the program encounters an address that has been set as a qualifier, it decrements the pass count. When the pass count reaches 0, it halts the processor.
<u>Set</u>	Click <b>Set</b> to enter a pass count value. The displayed Pass Count value does not download the value to the emulator's

### **Break on Trace Buffer Full**

Break on Trace Enables break on trace buffer full. Check this box (✔) to halt the processor when the trace buffer is full. The trace buffer is full when it captures 8K instructions/cycles.

pass counter until you click on Set.

## **Trace Settings**

Select <u>*Debug > Trace Settings*</u> to display the Trace Point Settings dialog box for defining up to 16 named trace point ranges.

Title trace1	<u>R</u> emove <u>C</u> lose
Start	Remove <u>A</u> ll <u>D</u> isable All
	Address is <u>Qualifier</u>
	Pass Count 0 Set
	Break on Trace Buffer Full
	♦ <u>T</u> race All
	<ul> <li>Trace First Cycle Only</li> <li>Trace Data Cycle Only</li> </ul>

Figure 6.19 Trace Settings

- **Note:** 1. The pass counter qualifier address can be assigned either to the Trace logic or break logic.
  - 2. The pass counter qualifier address can be set either in this menu or in the Breakpoint Settings dialog.

-		Trace Memory Window	▼ 4
1 D05D	0EA9	swapf 0x29	(0000000)
2 005E	02A8	subwf 0x28	(0000000)
3 005F	2061	call DoubleB	(0000000)
4 0060	3400	(Extra Cycle: Forced NOP)	
5 0060	3400	retlw 0x0	(0000000)
6 0061	0EA9	(Extra Cycle: Forced NOP)	
7 005B	0EA9	ReduceA swapf 0x29	(0000000)
8 005C	0E29	swapf Ox29,W	(0000000)
9 005D	0EA9	swapf 0x29	(0000000)
10 005E	02A8	subwf 0x28	(0000000)
11 005F	2061	call DoubleB	(0000000)
12 0060	3400	(Extra Cycle: Forced NOP)	
13 0060	3400	retlw 0x0	(0000000)
14 0061	0EA9	(Extra Cycle: Forced NOP)	
15 <mark>005</mark> B	0EA9	ReduceA swapf 0x29	(0000000)
16 <mark>005</mark> C	0E29	swapf Ox29,W	(0000000)
17 005D	0EA9	swapf 0x29	(0000000)
18 005E	02A8	subwf 0x28	(0000000)



Trace All

Lets you trace each enabled trace address location. This option displays the trace buffer exactly the way the processor fetches and executes instructions at each clock cycle. Traced locations include extra fetches in multicycle instructions such as GOTO and CALL.

For the PIC17CXX, traced addresses also include data cycles on TABLWT and TABLRD instructions.

To save captured data cycles into a file for plotting or analysis, use <u>*File*</u> > <u>*Export*</u> > <u>*Save Trace Buffer*</u>. In cases where the capture data will be plotted as graphs or histograms, MPLAB provides DDE support for direct transfer of data to other programs like Microsoft Excel.<sup>®</sup> For further information about DDE, refer to the DDE support section of this manual.

#### **Simulator Trace Display**

Simulator Trace Display differs slightly different from the Emulator Trace Display. It does not show the "Extra Cycle: Forced NOP" on two cycle instructions, nor does it show the Logic Probe inputs on the right side of the display. The Simulator Trace Display shows the following additional information:

- A time stamp of when the instruction occurred relative to the last reset.
- The time stamp uses the same data as the MPLAB Stop Watch.
- Registers that changed on the current instruction.
- The return addresses currently on the stack.

·				Trae	e Menory Window
	2909		9121	start	1 4.001s 1 19100
0 0000	3077	stert	mit/Uw	OxFF	: 6.00us 3 1W:FF
3 000 *	00.68		a sveri	(1a28	1 0.00as 3 120;FF 03:10
4 000A	2001		is ave bar	0 mt	; 10.00us 4 (wi01
\$ 000 F	00A9		mary at		: LE,00ag 5 1E9:0L 00:18
# 0002	3007		movier	(3=7	1 18,00as # 18)07
7.0001	00AA		HIVHE	0.82.8	1 16.00%# 7 12A107 03118
0.0008	0000		212		1 18.00 ag 8 1
9 1000F		Lasp		Reduced	1 22.00 as 8 [8:07 [ 10 ]
10 0013	0928	3800088	HIVE	0.023,8	1 24.00ms 10 10101 03119
11 0014	00.48		aupor.		: 24.00as LL 100:FE 03:LB
LT COLL	2017		1.Les	Tenklel	1 30.00as II  Wr01 [ 18 10 ]
13 0017	0929	and or	Have t	0.825,9	1 32.00 mm 13 (W101 03:1m
14 0018	07.85		addat	0.825	1 38.00as L8 109:00 03:18
TR COTA	201.8		0811	retendl	1 30.00as 15 [Will [ 1A 16 13 ]
16 0018	2018	sutsui2	oali	sutsade	: #2.00us 16 (#101 ( 10 1A 16 10 )
17 0013	20LF	entendä	call .	entend5	1 48.00as 17 10101 0 18 10 LA 16 10 1
TH COTA	2021	watend3	us11	tetradi.	[ 50.00as 18 [WiG1 [ 20 18 10 14 16 10 ]
13 0021	2023	satende	oall	watend?	1 58.00 as 18 10101 [ 22 20 18 10 16 10 ]
00.0023	0025	estand7	Coll.	estend5	1 50,00 mm 00 14101 1 08 00 00 18 10 1A 18 10 1

Figure 6.21 Simulator Trace Display

## **Clear All Points**

Select <u>Debug > Clear All Points</u> to clear all break and trace points.

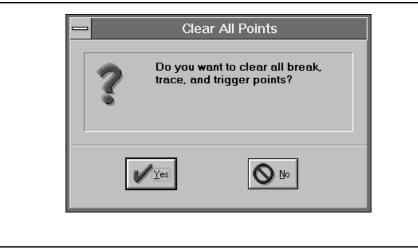


Figure 6.22 Clear All Points Message Box

## Clear Program Memory (Ctrl+Shift+F2)

Select <u>Debug > Clear Program Memory</u> to clear program memory to all 0xFF's. This function sets all program memory bits to ones.

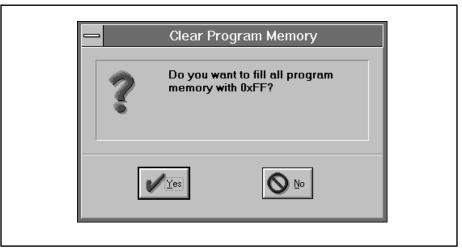


Figure 6.23 Clear Program Memory Message Box

## System Reset (Ctrl+Shift+F3)

Select <u>Debug > System Reset</u> to reset the entire emulator system including the PICMASTER emulator hardware (if connected), software and the target processor. System Reset performs the same initialization that is performed when MPLAB is first entered.

**Note:** To perform a processor reset (MCLR), select <u>Debug > Run ></u> <u>Reset</u>.

**Note:** Always power down the emulator pod when changing probes, and then run a system reset. If you do not run a system reset after changing probes, MPLAB will not be properly configured for the new probe.

## Power-On-Reset (Ctrl+Shift+F5)

Select <u>*Debug > Power-On-Reset*</u> to display the Power-On-Reset dialog box for selecting a POR option.

<ul> <li>Randomize all RAM Locations</li> <li>Fill all RAM Locations with Value</li> <li>Last Values</li> <li>Values from File</li> </ul>	- Power On Reset Emu	Ilation
	Fill all PAM Locations with Value Last Values	Select File

#### Figure 6.24 Power-On-Reset Dialog

Power-On-Reset allows you to emulate the power-on-reset function of the target processor.

Often, uninitialized registers can lead to a program malfunction that is hard to track down. POR will simulate the action of randomizing registers when the application first starts up. If the application misbehaves sometimes, the Power-On-Reset function may help you isolate the problem.

**Note:** Power-On-Reset is implemented on a device by tying the MCLR pin to VDD. The POR signal resets registers as described in the Microchip Databook for the specific device.

The Power-On-Reset dialog can be used for the following functions:

- Randomize registers that have an unknown value at POR.
- Fill registers with a value, or clear registers.
- Set registers to the POR Condition shown in the Microchip Databook for the respective part. The Fill with Value POR function will not affect the registers that have specifically defined reset values.
- Save current POR values to a file.
- Load POR values from a file.

Program memory and breakpoints are undisturbed when using Note: the POR dialog. Randomize Select Randomize to enter random values into registers that have an unknown value at POR. Fill with Value Type a fill value in the Enter Value box that you want to enter into registers at POR. Last Values Select Last Values to enter the last randomized or filled values into the device at POR. Power-On-Reset Click **Power-On-Reset** to reset and set selected register values. Cancel Click Cancel to close the Power-On-Reset dialog box without performing a POR. Save As Opens a dialog box to enter the name of a file (\*.POR) to save Power-On-Reset settings. Values from File To load data values from a file, click Values from File and then click **Select** to open a dialog box to enter the name of an \*.POR file containing values to load at power-onreset.

## **Options Menu**

## **Development Mode**

Select <u>Options > Development Mode</u> to change the current Development Mode setting.

Processor: PIC17C42A
> PICMASTER <u>E</u> mulator /0 Port:
$\diamond$ 200 $$ $\bigstar$ 208 $$ $\diamond$ 210 $$ $\diamond$ 218
$\diamond$ 300 $\diamond$ 308 $\diamond$ 310 $\diamond$ 318

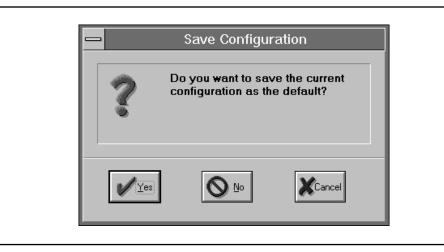
Figure 6.25 Development Mode Dialog

MPLAB-SIM Simulator	Selects the MPLAB-SIM simulator mode. <u>Processor</u> Selects the processor that you are simulating.
PICMASTER Emulator	Selects the PICMASTER Emulator mode (if connected) and displays Em on the Status Bar. <u>I/O Port</u> Selects the I/O address of the PICMASTER interface board in your PC. Your I/O address setting will correspond to your current dip switch settings on your PC Interface Board.
Editor Only	Selects the Editor Only mode and displays EO on the Status Bar. In Editor Only mode all emulator and simulator functions are disabled. You can only edit, compile, and perform project management operations.
<u>Reset</u>	Click <b>Reset</b> to reinitialize the emulator hardware (if connected). If the message "Unable to find Emulator System" displays, then the selected port may not be available. This option is useful mainly after changing the processor probe.
same as the <u>L</u>	ction is a manual reset that you execute and is the Debug > System Reset option. This reset function m Power-On-Reset.
<u>Cancel</u>	Select <b>Cancel</b> to cancel your selection and exit this display.

## Window Setup

### Save Setup

Select <u>Options > Window Setup > Save Setup</u> to save the current configuration to a file. The default extension of the configuration file is \*.CFG.



#### Figure 6.26 Save Setup as Default Message Box

Save Setup only saves the currently open windows—not the break, trace, and trigger points.

- Yes If you answer **Yes** to the question, "Do you want to save the current configuration as the default?" MPLAB will save the current configuration as the default user configuration that MPLAB loads during start-up.
- <u>No</u> If you answer **No**, MPLAB displays the Save Configuration dialog box. At this point, enter the name and path where you want to save the setup.
- <u>Cancel</u> Select **Cancel** to exit Save Setup without saving.

#### Load Setup

Select <u>Options > Window Setup > Load Setup</u> to load a configuration setup from a file previously stored by the <u>Options > Window Setup > Save Setup</u> command. Select the file that you want to read from the dialog box and press **Enter**. The default extension of the setup file is \*.CFG.

### **Default Configuration**

To load the default configuration select one of the following:

Options > Window Setup > Default Configuration, or

<u>Options > Environment Setup</u> and click on Load Default Configuration.

## **Key Mappings**

Select <u>Options > Key Mappings</u> to display the MPLAB key mappings. These key mappings enable you to perform the most common operations quickly. You may use the existing key mappings or modify the key mappings to meet your specific needs for your current project.

- Key Mapping	
Key/Key Sequence:	Change
Prefix key: (none) <b>±</b> C <u>o</u> nfigure	Change
then	<u>D</u> elete
_ <u>A</u> lt + <u>∠</u> <u>C</u> trl + <u>S</u> hift + F2	
Ctrl+F2	Save
Currently mapped to	
ProjectOpenProject	Save As
✓ Show this mapping on menu	Load
Functions:	
±	
Currently mapped from:	Cancel
*	Reset
*	
	Help

Figure 6.27 Key Mapping Dialog

By default, no prefix keys are enabled. This means that only single keys are available for mapping.

MPLAB uses the binary initialization file, MPLAB.KEY, in the Windows directory to record the key mapping values that will be carried from one session to another.

**Functions** 

The Functions pull-down menu displays the MPLAB key mappings. (Refer to the MPLAB Key Mapping Functions appendix for details.)

## **Environment Setup (Ctrl+F7)**

<u>*Options > Environment Setup*</u> opens a dialog box for changing the screen font, toolbar, global switches, and the symbol display width.

Environment Setup	
Screen Font: ANSI OEM System Other Courier New Select	Toolbar:       ✓ Enable         ♦ Top       Bottom         ♦ Bottom       Left         ♦ Right       Float
Global Switches: Status Bar Enable Clear Memory on Download Clear Breakpoints on Download Load User Configuration Global Break Enable Track Source Code OK	Symbol Display Width: Register Variables: width = 8 * chars Address Labels: width = 10 * chars Cancel

Figure 6.28 Environment Setup Dialog

#### **Screen Font**

The screen font option allows you to select a fixed point font for MPLAB screen displays.

### Toolbar

The Toolbar options allow you to select a location for the toolbar on the screen. Available locations are: Top, Bottom, Left, Right, and Float.

<u>Enable</u>

Select **Enable** to display the toolbar.

## **MPLAB USER'S GUIDE**

## **Toolbar Setup**

The toolbar setup dialog box allows you to redefine the operations that the toolbar icons perform.

Toolbar       User Defined Toolbar <ul> <li>Button</li> <li>Button 1</li> <li>Operation</li> <li>CursorStartOfFile</li> <li>✓</li> </ul> Icon         BITMAP_26           Disable Button             Ørenation         Ørenation                     Icon         BITMAP_26                           Ørenation         Ørenation	Button Size OK    Automatic Cancel  Large  Medium  Small

#### Figure 6.29 Toolbar Settings Dialog

<u>Toolbar</u>	Selects which toolbar to edit. There are four possible selections: Edit, Debug, Project, and User Toolbar.
Button	Selects an Icon button location. The toolbar has 16 available locations.
<u>Operation</u>	Selects the operation that MPLAB will perform for the selected button location. Refer to the MPLAB Key Mapping Functions appendix for a list of available operations.
lcon	Selects an icon to display in the selected button location.
Disable Button	Disables the icon button at the selected location on the toolbar.
Gap Before Button	Inserts a small gap before the icon button.
Button Size	Changes the icon button size. Button size options are: Automatic, Large, Medium, and Small.
<u>Ok</u>	Defines the toolbar as currently shown in the Setup dialog box.
<u>Cancel</u>	Returns the toolbar to its previous state.

#### **Global Switches**

The Global Switches allow you to turn these user selections on or off.

Status Bar Enable	Turns the Status Bar on and off.
<u>Clear Memory on</u> Download	When selected, this global switch clears memory before MPLAB downloads to the emulator. This function sets all program memory bits to one.
<u>Clear Breakpoints on</u> Download	When selected, removes all breakpoints, trace points, trigger points, and pass count addresses on download to the emulator.
Load Default Configuration	When selected, MPLAB loads the default user configuration at start-up. To change the default window setup, open the windows you want to load at start-up, click <u>Options &gt; Window Setup &gt;</u> <u>Save Setup</u> and click <b>Yes</b> . The MPLAB window setup may contain any available MPLAB window. MPLAB.CFG is the default user configuration file.
<u>Global Break Enable</u>	When selected, enables all breakpoints. If Global Break Enable is not selected, then all breakpoints are disabled. Global Break Enable is also available from the Status Bar.
Track Source Code	When selected, MPLAB updates the current line in the source code when single stepping. You may wish to turn this feature off if you have a *.HEX file, but no *.COD file.

### Symbol Display Width

The symbol display width options allow you to specify how many character spaces MPLAB allocates when displaying symbolic information.

Register Variables	Allows you to select a width of 6 characters wide to 32 characters.
Address Labels	Allows you to select a width of 6 characters wide to 32 characters.

## Colors

Select  $\underline{Options} > \underline{Colors}$  to change the colors assigned to displayed data. To change the color, select the text that you want to alter by clicking on it. Next, select the new color.

### **Processor Setup**

The Processor Setup commands allow you to configure the emulator environment.

#### Hardware

Select <u>Options > Processor Setup> Hardware</u> to display a dialog box for entering additional hardware setup data for the processor you are emulating.

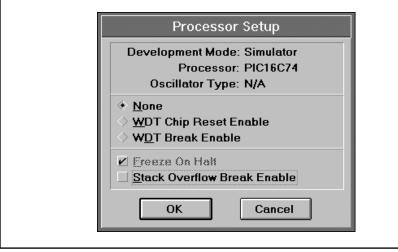
Hardware Selections — PIC16C5X Family

Proce	essor Setup	
Proc	Mode: Simulator essor: PIC16C57 Type: N/A	
◆ <u>N</u> one ◇ <u>W</u> DT Chip Re ◇ W <u>D</u> T Break E		
✓ Freeze On He	ali ow Break Enable	
ОК	Cancel	

Figure 6.30	PIC16C5X Processor Setup Dial	og
-------------	-------------------------------	----

Watch Dog Timer	Enables Watch Dog Timer
Watch Dog Timer Break Enable	Halts the processor when the Watch Dog Timer times out.
Freeze on Halt	The simulator does not support this function.
<u>Stack Overflow Break</u> Enable	Select Stack Break Enable to halt the processor if a stack overflow or underflow occurs.
<u>OK</u>	Accepts entries made.
<u>Cancel</u>	Closes this dialog box without making any changes.

### Hardware Selections — PIC16CXX Family



#### Figure 6.31 PIC16CXX Processor Setup Dialog

Watch Dog Timer	Enables Watch Dog Timer
<u>Watch Dog Timer Break</u> <u>Enable</u>	Halts the processor when the Watch Dog Timer times out.
Freeze on Halt	The simulator does not support this function.
<u>Stack Overflow Break</u> Enable	Select Stack Break Enable to halt the processor if a stack overflow or underflow occurs.
<u>OK</u>	Accepts entries made.
<u>Cancel</u>	Closes this dialog box without making any changes.

Processo	or Setup					
Development Mode: Simulator Processor: PIC17C42A Oscillator Type: N/A	Processor Mode: * <u>Microcontroller</u> Microgrocessor					
<ul> <li>None</li> <li>WDT Chip Reset Enable</li> <li>WDT Break Enable</li> </ul>	Egtended Microcontroller     Off-Chip Memory:     From Target Board					
✓ Freeze On Halt ☐ Stack Overflow Break Enable ☐ Disable Long Writes	Supplied by Emulator     WDT Prescale:     1					
ОК	Cancel					

### Hardware Selections — PIC17CXX Family

Figure 6.32 PIC17C4X Processor Setup Dialog

#### Processor Mode

Selects the microcontroller, microprocessor, or extended microcontroller mode. Refer to the table below and the PIC17CXX data sheet for additional details on modes.

	Mode	Internal Memory	External Memory
Microco	ontroller	0- [max]*	None
Micropr	rocessor	None	0 - 0xFFFF
Extended Microcontroller		0 - [max]*	[max]* - 0xFFFF
[max]	Max refers to the	e maximum ROM memory	value.
*		ochip data sheet to get the controller that you are using	

#### Watch Dog Timer

This group allows you to set up and enable the Watch Dog Timer (WDT).

Watch Dog Timer	Enables Watch Dog Timer.
<u>Watch Dog Timer Break</u> <u>Enable</u>	Halts the processor when the Watch Dog Timer times out.
WDT Prescale	Selects timer cycles required to increment the Watch Dog Timer.
Freeze on Halt	The simulator does not support this function.
<u>Stack Overflow Break</u> <u>Enable</u>	Select Stack Break Enable to halt the processor if a stack overflow or underflow occurs.
<u>OK</u>	Accepts entries made.
Cancel	Closes this dialog box without making any changes.

## **Tools Menu**

## DOS Command to Window (F11)

<u>Tools > DOS Command to Window</u> allows you to run a DOS program such as a compiler, or an internal command such as DIR, and capture the output into an edit window.

The command starts a dialog that prompts you for the command line to be executed. This can be any DOS command, including built-in ones like DIR.

You can specify the working directory that the command is to run in. This affects only the DOS command. MPLAB will still use its previous working directory.

When you start the dialog, MPLAB sets the command string and the proposed working directory name to the values you set the last time you used it.

You may execute only one DOS command at a time. Until the command completes, MPLAB will reject attempts to start another.

— Execute DOS Comma	nd and Capture Output
Current directory: h:\basedata	
<u>C</u> ommand: dir <u>W</u> orking Directory: c:\mplab	<u>±</u>
Record Full Directory Pathnam     Options:	ne Browse OK
☐ Beep when <u>d</u> one <u>M</u> inimize Editor <u>S</u> ave changed file	Cancel
<ul> <li>Reuse output window</li> <li>Show end of output</li> </ul>	<u>H</u> elp

#### Note: You cannot start a Windows application with this dialog.

#### Figure 6.33 DOS Command to Windows

1. Set the various options you want to apply:

<u>Beep When Done</u> causes the Editor to sound the standard system beep when the DOS program finishes.

<u>Minimize Editor</u> causes the Editor to make itself into an icon before running the DOS program

<u>Save Changed Files</u> causes the Editor to see if any of the files you're editing have changed and will give you the opportunity to save the changes before running the DOS program. If you agree to save the changes, all the files you've altered will be written to disk.

<u>Reuse Output Window</u> causes the Editor to place the DOS program's output into the window used the last time you used this dialog. If not, a new output window will be created.

<u>Show End Of Output</u> causes the Editor to automatically scroll the window showing the output to show you the end of it rather than the start.

2. Click **Ok** button to run the DOS program.

The command line, working directory path, and options are recorded and become the default the next time you use this dialog.

## Repeat DOS Command to Window (Ctrl+F11)

<u>Tools > Repeat DOS Command to Window</u> exactly repeats the last DOS command you ran with Execute DOS Command To Window.

This command shows you the command output in a window when it completes. If you have not previously run a command, the Editor will act as if you selected Execute DOS Command To Window and will show you the dialog.

Repeat DOS Command To Window repeats the last command you issued with this dialog.

## Window Menu

All Window options are available in PICMASTER emulator mode and the MPLAB-SIM simulator mode.

In Editor Only mode, Absolute Listing, and Show Symbol List are available. In addition, the window positioning options and the Open Windows selections are also available.

## **Program Memory**

Select <u>Window > Program Memory</u> to display program memory. The program memory window can display locations in the range of program memory for the currently selected processor. You can leave the Program Memory window open at all times and move and resize the window.

The Program Memory window is only available in Emulator and Simulator mode.

-	Program Memory Window									
9		0008	3075	start	novler	0×FF				
10		0009	BADO		BOVWI	0x28				
11		A080	3001		novler	0×1				
2		000B	00A9		novwf	0x29				
13		0800	3007		novlw	0x7				
.4		0000	AA00		BOVWI	0x2A				
15		0808	0000		nop					
16		OUOP	2013	100p	call.	ReduceA				
17		0010	OBAA		decfaz	0x2A				
18		0011	280P		goto	loop				
19		0812	2808		goto	start				
01	B	0013	0829	ReduceA	HOVE	0x29,W				
21		0814	02A8		subwf	0x28				
22		0015	2017		cell	DoubleB				
23		0816	3408		retlw	0×0				
24	.7	0017	0829	Doubles	IROV É	0x29,4				
25	.7	0818	07A9		addwf	0.x2.9				
26	.7	0819	2018		11.00	extend3				
27		001A	3408		retly	0×0				

Figure 6.34 Program Memory Window

### **System Button Options**

Click the system button in the upper left corner of the Program Memory screen to display the following options:

Toggle Line Numbers	Toggles field for displaying line numbers and qualifier points.
Hex Code Display	Displays program memory information as hex data.
Machine Code Display	Displays disassembled hex code with no symbolic information.
Disassembly Display	Displays disassembled hex code with symbols.

### **Program Memory Field Descriptions**

MPLAB displays data in the Program Memory window that it reads directly from emulation memory. Program Memory fields contain the following information:

Field One	Address in hex.
Field Two	Opcode (or data) in hex.
Field Three	Program Label in symbolic format. You can increase the display width of labels from <u>Options &gt; Environment Setup &gt;</u> <u>Address Label</u> .
Field Four	Machine code, disassembled code, or source code.
Highlight Bar	Current location of the program counter.
Program Memor memory location	bry locations display break, trace and trigger out status at each on as follows:

#### **Selecting Points**

Symbol	Point Type	<b>Menu Selection</b> (RMB = Right Mouse Button)
В	Breakpoints	<u>RMB &gt; Break Point(s)</u>
Т	Trace Points	<u>RMB &gt; Trace Point(s)</u>
0	Trigger Output	<u>RMB &gt; Trigger Point(s)</u> (Available only with the PICMASTER emulator)
Q	Pass Count Address	Set from Break or Trace Point Settings Dialog

MPLAB uses a combination of color and notations in the Program Memory window to show break/trace/trigger points. At a particular address, if no points have been set, the text will display normally. If a point is set, the color of the text will change, the width of the line number window will increase to show the active points. Unset points are displayed as periods.

### **Creating a Temporary Real-Time Breakpoint**

To set up a temporary real-time breakpoint from the Program Memory Window, double click the left mouse button anywhere on a valid address line. The processor runs in real time until one of the following occurs:

- The line containing the temporary breakpoint is executed,
- A breakpoint is encountered, or
- You click on Halt.

## **Trace Memory**

Select <u>*Window > Trace Memory*</u> to display the contents of the trace buffer. This window can be left open at all times, moved around and resized.

Field One	Hex Address						
Field Two	Opcode/Data						
Field Three	Symbols for Address Labels						
Field Four	Disassembled Code						
Field Five	Emulator:	Status on External Logic Probe Lines. The status fields are displayed on the extreme right hand side of the window.					
	Simulator:	Time stamp and changed register information.					
To save the c	ontents of th	e trace buffer to a file, execute <i><u>File &gt; Save Trace</u></i>					

To save the contents of the trace buffer to a file, execute <u>File > Save</u> <u>Buffer</u>.

## **EEPROM Memory**

Select <u>Window > EEPROM Memory</u> to display the EEPROM data memory window for a microcontroller device that has EEPROM data memory. The PIC16C84 is an example of a device that supports EEPROM memory.

The EEPROM window can be left open at all times, moved around and resized. This window is for information only and you cannot change values from this window.

	00	01	02	03	04	05	06	07	08	09	ΟA	ОВ	0C	ΟD	ΟE	OF	1
0000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

#### Figure 6.35 EEPROM Memory Window

The EEPROM window displays the following data/opcode hex information of the particular processor being emulated. When an EEPROM register value changes or the processor is halted, the data in the EEPROM window is updated.

#### **System Button Options**

Click the system button in the upper left corner of the EEPROM Memory screen to display the following options:

Toggle Line Numbers	Toggles field for displaying line numbers.
<u>Hex Display</u>	Displays program memory information as Hex data.
ASCII Display	Displays the ASCII character at each memory location.

## **Absolute Listing**

Select <u>*Window > Absolute Listing*</u> to display the \*.LST file and to single step through the \*.LST file generated by MPASM or a compatible C compiler.

Use this function to display both C code and the assembly code that corresponds to the C code. The Absolute Listing gives you a better idea how the C compiler implemented your code. If you are not using C, you will get a similar listing file generated by MPASM.

## Stack

Select <u>Window > Stack</u> to open a window displaying the contents of the stack. The number of available levels depends on the processor type being emulated. The Stack Window can be left open at all times, moved around and resized.

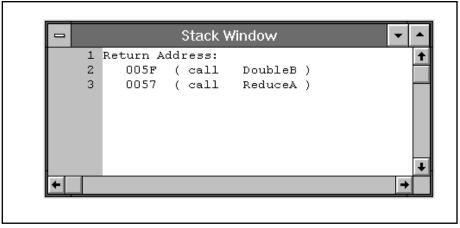


Figure 6.36 Stack Window

#### Hardware Stack Levels

#### PIC16C5X Hardware Stack – 2-Levels Deep

MPLAB displays the contents of the 2-level deep hardware stack.

#### PIC16CXX Hardware Stack – 8-Levels Deep

The PIC16CXX has an 8-level deep hardware stack.

**Note:** If you set Stack Break Enable on the Hardware configuration display, (*Options > Processor Setup > Hardware*) and push or pop the stack beyond its limit, for the PIC16CXX, MPLAB will display an underflow or overflow message.

#### PIC17CXX Hardware Stack – 16-Levels Deep

The PIC17CXX has a 16-level deep hardware stack.

**Note:** If you set Stack Break Enable on the Hardware configuration display, (<u>*Options > Processor Setup > Hardware*</u>) and push the stack beyond its limit, for the PIC17C4X, MPLAB will display an underflow or overflow message.

#### Simulator Stack—PIC16C5X Devices

The MPLAB-SIM simulator presents an accurate simulation of the hardware stack on the PIC16C5X devices and additionally provides warning messages if an underflow or overflow condition occurs. When a CALL instruction is encountered, or when an interrupt has occurred, the value of the PC+ 1 is pushed to the stack, and the stack is popped when a RETLW instruction is executed. If more than two values are pushed to the stack before it is popped, the value will be pushed to the stack, but a warning message will be issued, indicating a stack overflow condition. An error message will also be generated if the user attempts to pop an empty stack. Popping an empty stack will cause the last value popped to be put in the PC.

### Simulator Stack—PIC16CXX Devices

The MPLAB-SIM simulator presents an accurate simulation of the hardware stack on the PIC16CXX devices, and additionally provides warning messages if an underflow or overflow condition occurs. When a CALL instruction is encountered, or when an interrupt has occurred, the value of the PC+ 1 is pushed to the stack, and the stack is popped when a RETLW, RETURN, or RETFIE instruction is executed. If more than eight values are pushed to the stack before it is popped, the value will be pushed to the stack, but a warning message will be issued, indicating a stack overflow condition. An error message will also be generated if the user attempts to pop an empty stack. Popping an empty stack will cause the stack pointer to point to the top of a full stack, and will not generate an error message if another pop is initiated.

### Simulator Stack—PIC17CXX Devices

The MPLAB-SIM simulator presents an accurate simulation of the hardware stack on the PIC17CXX, and additionally provides warning messages if an underflow or overflow condition occurs. When a CALL or LCALL instruction is encountered or when an interrupt has occurred, the value of the PC+ 1 is pushed to the stack. The stack is popped when a RETLW, RETURN, or RETFIE instruction is executed. If more than sixteen values are pushed to the stack before it is popped, the value will be pushed to the stack, a warning message will be issued indicating a stack overflow condition, and the STAKAVL bit will be cleared until a reset condition occurs.

## **File Registers**

Select <u>Window > File Registers</u> to display a window of all the File Registers of the particular processor being emulated. When a file register value changes, or the processor is interrogated, the data in the File Register window is updated. The File Register window can be left opened at all times, moved around and resized.

	00	01	02	03	04	05	06	07	08	09	ΟA	Οв	0C	ΟD	ΟE	OF	
0000	44	FF	00	1A	$7  \mathrm{F}$	00	03				00	00	00				ľ
0010																00	H
0020	FF	00	FF	00	FF	00	FF	00	FF	04	FF	00	$7  \mathrm{F}$	00	FF	00	1
0030	FF	02	FF	00	FF	04	FF	41	FD	02	FF	40	EF	00	FF	00	1
0040	FF	00	FF	00	FF	00	FΕ	CO	$\mathbf{DF}$	08	FF	20	FF	08	FF	00	
0050	FF	02	$\mathbf{F}\mathbf{F}$	00	FF	02	$\mathbf{F}\mathbf{F}$	80	$\mathbf{DF}$	40	$\mathbf{F}\mathbf{F}$	00	$\mathbf{F}\mathbf{F}$	00	FF	04	1
0060	FF	00	$\mathbf{F}\mathbf{F}$	00	FF	10	$\mathbf{DF}$	00	FF	00	FF	00	FΕ	00	FF	80	1
0070	FF	08	FF	00	FF	02	FF	00	FF	04	FF	00	FF	00	FF	44	1
0080	00	$\mathbf{F}\mathbf{F}$	00	1A	$7  \mathrm{F}$	1F	$\mathbf{F}\mathbf{F}$				00	00	00		00		1
0090																00	1
00A0	FF	00	$\mathbf{F}\mathbf{F}$	00	FF	00	FF	CO	EF	08	FF	80	$\mathbf{F}\mathbf{F}$	00	FF	00	1
00в0	FF	00	FF	12	FF	04	FF	11	FF	20	FF	40	FΕ	00	FF	08	- 8

Figure 6.37 File Registers Window

From the File Register Window, you can open the Modify dialog box to fill a range of registers with a constant value. Either select a desired range by clicking the left mouse button and dragging over the values that you want to change, or simply place the cursor on the register you want to change and click the right mouse button to display the Fill Register popup menu. Select the Fill Register menu option to display the Modify dialog box with the address range already entered.

### **System Button Options**

Click the system button in the upper left corner of the File Register Memory screen to display the following options:

Toggle Line Numbers	Toggles field for displaying line numbers.
<u>Hex Display</u>	Displays file registers information as Hex data.
<u>Symbolic Display</u>	Displays data at each memory location in the following formats: Hex, Decimal, Binary, ASCII Character, Symbol, and Name.
ASCII Display	Displays the ASCII character at each memory location.

## **Special Function Registers**

Select <u>Window > Special Function Registers</u> to display the contents of the Special Function Registers (SFR) for the processor being emulated. Whenever a break occurs, the contents of the Special Function Registers are updated.

This window can be left opened at all times, moved around, and resized. A sample of this window is shown below.

SFR Name	Hex	Dec	Binary	Char	t
tmrO	FF	255	11111111		
option	FF	255	11111111		_
pcl	00	0	00000000		
status	1A	26	00011010		
fsr	7 F	127	01111111		
porta	00	0	00000000		
trisa	1F	31	00011111		
portb	03	3	00000011		
trisb	FF	255	11111111		
pclath	00	0	00000000		
intcon	00	0	00000000		
pir1	00	0	00000000		
pie1	00	0	00000000		
pcon	00	0	00000000		
cmcon	00	0	00000000		
vrcon	00	0	00000000		
w	00	0	00000000		Ŧ

Figure 6.38 Special Function Registers

Field One	Name of Special Function Register
Field Two	Data as a hexadecimal value
Field Three	Data as a decimal value
Field Four	Data as a binary number
Field Five	Data as ASCII characters
To modify the	contants of a particular Special Functi

To modify the contents of a particular Special Function Register:

- 1. Double click on a register in this window to invoke the Modify dialog box with the symbol/address and data fields already filled in, or
- 2. Use the execute <u>Window > Modify</u> menu.

Note: The SFR names and addresses are different for every device.

## **Calibration Dialog**

The Calibration Data dialog is for use with the PIC140XX device family. It displays the floating point data in the Emulator Probe for alteration by the user.

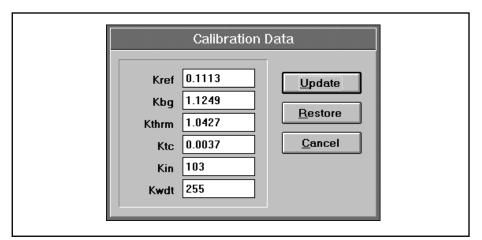


Figure 6.39 Calibration Data Dialog

- Update The Update button takes the information from calibration data dialog, converts it from the IEEE 754 format to the Microchip version of IEEE 754, and stores it in the memory area for use in the embedded code. Only the first four numbers Kref, Kbg, Kthrm, and Ktc are in floating point format. The last two, Kin and Kwdt, are 8 bit unsigned numbers with values from 0 to 255.
- Restore The Restore button takes the original calibration data uploaded from the probe during system reset and resets the values in the memory area for use in the embedded code.
- Cancel The Cancel button closes the dialog and does not modify the values in the memory area.

## Show Symbol List (Ctrl+F8)

Click <u>Window > Show Symbol List</u> to display all symbols known to MPLAB. Symbols include constants and labels. Show Symbol List is an information only dialog box. To display Show Symbol List, a project **MUST BE OPEN AND BUILT.** 

The symbols displayed in this dialog box represent the symbols imported from your source code after compiling or assembling.

	5	ymbol List	
ariable	Address	Label	Address
16C74 reg reg ount CLATH	(0x0001) (0x0028) (0x0029) (0x0029) (0x0029) (0x0001) (0x0000) (0x0000)	Double8 loop ReduceA start	(0x0061) (0x0057) (0x0056) (0x0050)
		J [ [	OK

Figure 6.40 Show Symbol List

<u>Variable, Address</u>	Displays variables from the File Register memory and the address of each variable.
Label, Address	Displays labels from program memory and the address of each label.
<u>Constants</u>	Constants defined in the source code can be used in executing opcode and as operands for instructions in Modify.

## Stopwatch

Select <u>Window > Stopwatch</u> to display the current value of the Cycle counter. The system Stopwatch is a 48-bit timer that counts the number of clock cycles that the processor executes. The counting occurs with real-time execution and with polled execution. The timer triggers on every cycle of an instruction.

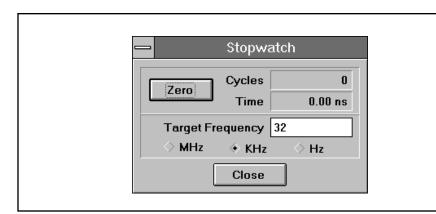


Figure 6.41 Stopwatch Dialog Box

<u>Timer Value</u>	The 48-bit Timer counts and displays the number of cycles that the processor executes. The stopwatch value displays in seconds and is computed from the number of cycles executed and from the operating frequency that you entered.
	Executing the processor in real time at 25 MHz Input Clock requires approximately 499 Days to overflow the Cycle counter.
<u>Zero</u>	Click <b>Zero</b> to reset the 48 bit cycle counter to zero. You can reset the timer at any time when the processor is halted.
Example:	You can use this timer for precise timing measurements. If you need to measure the exact time a subroutine takes to execute, then simply reset the timer before entering the subroutine and put a breakpoint at the end of the subroutine. The timer displays the total number of cycles executed in the subroutine and also displays the execution time.
<u>Target Frequency</u>	Allows you to enter a user-defined Input Clock frequency to the target processor (CLKIN frequency). MPLAB uses this data for making real-time computations only. This entry does not actually set the input clock frequency to the processor. The Timer Value/Cycle Counter window uses this frequency to translate the Cycle count to Time count.
	To enter the desired frequency, first select the desired range (either MHz, kHz or Hz) by clicking the appropriate button. Then type in the desired frequency. For example, to enter 32 kHz as input clock frequency, enter data as shown in the above the Dialog Box.

### **Project Window**

The Project Window is only available when a project is open. It displays the list of files currently in the project. If the project has been compiled, the project window displays a list of all included files in the project. Otherwise, the Project Window it only displays the main project file. Double click on any file displayed in the Project Window to open that file for editing.

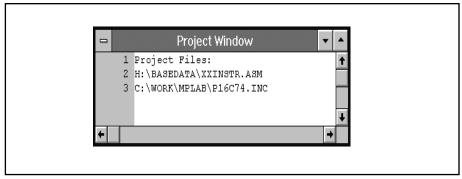


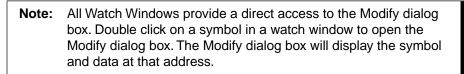
Figure 6.42 Project Window

## **New Watch Window**

Select <u>Window > New Watch Window</u> to display a new watch window and its Edit Watch dialog box. You may also access the Edit Watch dialog by either selecting the Edit Watch option from the system menu or by pressing the <Ins> key.

Use the Edit Watch dialog box to add symbol and address data to the new watch window.

To remove a watch variable, place the cursor on the line containing the variable that you want to delete and press the <delete> key.



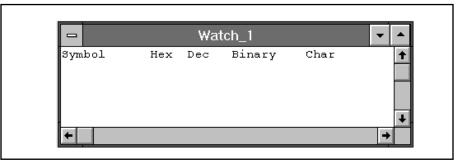


Figure 6.43 Watch Window

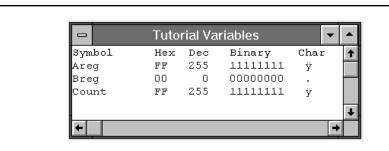


Figure 6.44 Watch Window

### **System Button Options**

<u>R</u> estore		7	Char	Ť
<u>M</u> ove			Ϋ́	
<u>S</u> ize		)00 L11	· ÿ	
Mi <u>n</u> imize		[	3	
Maximize			-	+
<u>C</u> lose	Ctrl+F4			
Next	Ctrl+F6			
Toggle Line Numbers				
Edit Watch				
Save Watch				

Figure 6.45 Watch Window System Button Options

#### **Edit Watch**

From the Edit Watch dialog box you can assign a title to a new watch window and add symbol and address data to the new watch window.

To access the Edit Watch dialog box at a later time, click the system button in the upper left corner of the watch window and select Edit Watch. You may also access the Edit Watch dialog box by pressing the <Ins> key (Insert) when a watch window is selected.

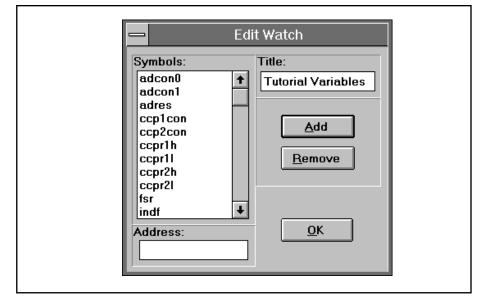


Figure 6.46 Edit Watch Dialog Box

<u>Title</u>	Enter up to a 39-character title for the new Watch Window. MPLAB accepts all characters for the title.
<u>Symbols</u>	You may use any valid symbolic names for Register Names.
Address	Enter addresses as hexadecimal values. An address beginning with A through F must have a preceding 0 (zero).
<u>Add</u>	Click the <b>Add</b> button to insert a symbol or address in the watch window. Add data to a watch window in the sequence you want to display the data. You can double click on a symbol in the list to automatically add it to the watch window.
<u>Remove</u>	Select a symbol or type in an address. Then click the <b>Remove</b> button to delete an entry from a watch window. You may also press <b>DEL</b> when the watch window is selected to remove a variable from the watch window. The Watch Window removes the variable where the cursor is.
<u>Ok</u>	Click <b>Ok</b> to exit the Edit Watch dialog box. After exiting Edit Watch, MPLAB will display the new watch window but the new watch window is not saved yet. You must use Save Watch to save the watch window.
Save Wa	atch

You must save the watch window to disk if you want to retrieve it later. MPLAB saves the watch window with the title you enter in the same watch dialog with the extension, \*.WAT.

**Note:** A Watch Window created in one project can overwrite a watch window having the same name from another project in the same directory since the Watch Windows are saved as files.

To access Save Watch, click the system button in the upper left corner of the watch window.

### Load Watch Window

Select <u>Window > Load Watch Window</u> to load a watch window that you previously saved to disk. Select a watch window file to load and click Ok, or double click the desired file.

## Modify

Select <u>*Window > Modify*</u> to display and/or modify the contents of Data Memory, Program Memory, the Stack, or EEPROM memory.

Modify allows you to Read/Write to a specific address, Read/Write while incrementing to the next address, or fill an address block. MPLAB allows you to leave the Modify window open at all times and move it around.

	Modi	ify
Address: Data /Opcode: Radix:	♦ <u>D</u> ecimal ea: ♦ <u>P</u> rogram ♦ EEPROM	End Address: <u>Fill Range</u> <u>Auto Incremen</u> <u>Write</u> <u>Read</u> <u>Close</u>

Figure 6.47 Modify Dialog Box

MPLAB provides four ways to open the Modify dialog box:

- Select <u>Window > Modify</u>.
- Double click on a item in the Special Functions Register window
- Double click an item in a Watch Window.
- Select an address or range in the File Register Window and click the right mouse button to display the Fill Register button. Click the Fill Register button to display the Modify dialog box.

<u>Address</u>	Enter the Address at which data is to be read or modified. You can enter a numerical address or a symbol. (Label)
Data/Opcode	Click <b>Read</b> to display data value/ Opcode at a selected address and memory area. Click <b>Write</b> to write data value/ Opcode to the selected address and memory area.
Radix	Hex, Decimal, or Mnemonics
<u>Memory Area</u>	Select the Memory Area that you want to modify:

	Data Memory: Program Memor Stack: EEPROM:	RAM Memory y: ROM Memory in the emulator Stack Memory on the Device EE Data Memory
<u>End Adr</u>	The ending address	for Fill Range.
Fill Range	Fills the range defined by the two addresses with the value entered in Data/Opcode.	
Auto Increment	Select Auto Increment to increment to the next address after each Read/Write.	
	displays th at the ado read a ran area minu	ment increments to the next address, ne next address, and reads the contents ress. If you are using Auto Increment to age, enter the address of the memory s one because the first read will the address.
<u>Write</u>	Enter new data in the Data/Opcode field, and click Write to modify the data at the specified address. (You can enter data in symbolic format.) When data is modified, all the appropriate windows are updated with the new information.	
<u>Read</u>	Click <b>Read</b> to read the data at a specified address.	
<u>Close</u>	Click <b>Close</b> to exit from Modify.	
Caution: Use care when modifying the stack or File register 2 (PCL). The effect of modifying these registers is not seen until the processor is taken out of halt.		

## **Tile Horizontal**

The <u>Window > Tile Horizontal</u> command sizes open windows in a horizontal format making each window as wide as possible to allow you to see as much of each line in as many windows as possible. The command arranges all open windows in a tile pattern, placing the windows above one another. Excess windows are tiled in a horizontal pattern in the lower part of the screen.

Windows containing the output from commands run by the <u>*Tools > DOS</u></u> <u><i>Command To Window*</u> command are arranged preferentially at the top of the screen.</u>

## **Tile Vertical**

<u>Window > Tile Vertical</u> command sizes open windows vertically in columns to allow you to see as many lines as possible in each window.

The Tile Vertical command arranges all open windows in a tile pattern, placing them side by side so that each window is as deep as possible.

# **Chapter 6. MPLAB Menu Options**

### Cascade

The <u>*Window > Cascade*</u> command arranges all open windows in a cascade pattern.

## **Iconize All**

<u>Window > Iconize All</u> makes all windows into icons.

## Arrange Icons

<u>Window > Arrange lcons</u> arranges all iconized windows so that their icons are visible in rows at the bottom of the desktop. Open windows are not affected by this command.

## **Open Windows**

<u>Window > Open Windows</u> lists the open windows at the end of the Window commands.

Whenever you open a window, MPLAB records the name in the list, ordering it so that the windows you have used most recently always appear at the top.

More WindowsWhen the list contains more files than can be<br/>displayed on the Window menu, an extra menu<br/>command More Windows is automatically added.<br/>The command starts a dialog that shows you the<br/>entire contents of the list and lets you pick the<br/>window you want to open.To open a window from the list, either double-click<br/>the window name in the list with the left mouse

button, or select the name and press the **Open** button. The More Windows dialog box also provides

additional options. The settings of the additional options are remembered and become the default for the next time you use this dialog.

	Select Window
c:\win\mplab.ini c:\work\mplab\editor\src\c File Register Window h:\basedata\16xx.asm h:\basedata\led621.c Program Memory Window r:\home\pbrink\basedata\1 Special Function Register Stack Window Trace Memory Window	165x.asm
Windows showing	Changed <u>w</u> indows only
<ul> <li>☑ Unnamed files</li> <li>☑ Templates</li> <li>☑ Command output</li> </ul>	OK Cancel <u>H</u> elp

Figure 6.48 More Windows Dialog

<u>Changed Windows</u> <u>Only</u>	Displays only windows that have been changed since opening.
Named Files	Lists all windows having an associated file name.
Unnamed Files	Lists all windows that do not have an associated file name.
<u>Templates</u>	Lists all template windows.
Command Output	Lists all windows containing command output obtained by executing <u>Tools &gt; DOS Command to</u> <u>Window</u> .

## **Help Menu**

### Release Notes (Shift+F1)

<u>Help > Release Notes</u> opens and displays the recent change history of MPLAB software. The Release Notes are contained in the file, README.1ST.

### **Getting Started**

<u>Help > Getting Started</u> contains help on installing your PICMASTER emulator and on getting it running with the MPLAB IDE.

### **MPLAB Help**

<u>Help > MPLAB Help</u> contains help on using MPLAB.

# **Chapter 6. MPLAB Menu Options**

## **Editor Help**

<u>Help > Editor Help</u> contains help on using the MPLAB Editor.

## **MPASM Help**

<u>Help > MPASM Help</u> opens the entire MPASM User's Guide. Click on the highlighted items to get more information on a particular item.

MPASM Help also contains a Quick Reference Guide with the following information:

- Compile Directives
- PIC16C5X Instruction Set
- PIC16CXX Instruction Set
- PIC17C4X Instruction Set

#### About

<u>*Help > About*</u> displays the MPLAB version number as well as the address of Microchip Technology and the following versions:

- IDE Version
- Processor Version
- Disassembler Version
- MPLAB-SIM Version

In addition, MPLAB displays the following information:

- Free System Resources (%)
- Free Space in USER Heap (%)
- Free space in GDI Heap (%)
- Copyright Notice



MPLAB Edito	or Preview	109
Chapter 1.	About MPLAB Editor	111
Chapter 2.	Using the MPLAB Editor	113
Chapter 3.	MPLAB Editor Menu Commands	121
Chapter 4.	MPLAB Editor Default Key Commands	131
Chapter 5.	MPLAB Editor Error Messages	137
Chapter 6.	MPLAB Text Editor Command Line Options	139

Part 2 – Editor



## Chapter 1. About MPLAB Editor

## Introduction

This chapter itemizes the MPLAB Editor features, describes how this user's guide is organized, and lists recommended reading.

# Highlights

Chapter 1 covers the following topics:

- Product Definition
- About This Guide
- Recommended Reading
- Customer Support

## **Product Definition**

### What is MPLAB Editor

MPLAB Editor is a basic part of the MPLAB Integrated Development Environment (IDE). The MPLAB IDE and Editor are designed to allow PIC16/17 Microcontroller developers an easy and quick method to develop and debug firmware for Microchip Technology Incorporated's PIC16/17 microcontroller product families.

The MPLAB IDE runs under Microsoft Windows<sup>®</sup> 3.1 or later.

### **MPLAB Editor Features**

The MPLAB Editor provides functions that allow you to perform the following operations:

#### **File Commands**

- Create A New File
- Open an Existing File
- Change a File Name
- Save A File To Disk
- Save Changes to a File or Template
- Backup an Existing File
- Close a File

#### **Template Operations**

- Create a Template File
- Attach a Template File
- Automatically Attach Template Files
- Create a Template
- Insert a Template
- Edit a Template
- Delete a Template
- Save a Changed Template File
- Detach a Template File

#### **Text Handling**

- Type Text
- Select Text
- Delete, Cut, Copy, and Paste Text
- Find Text
- Replace Text
- Text Processing
  - Automatic Text Wrapping
  - Indenting and Unindenting Text
  - Changing Case
  - Handling Braces
  - Undoing Edit Actions

#### **C** Language Awareness

- Position the "#" Character in Column One
- Position Closing Brace "}" in Same Column as Opening Brace



## Chapter 2. Using the MPLAB Editor

## Introduction

This chapter contains overviews and descriptive information that explains the features of MPLAB Editor, and gives step-by-step instructions for using the MPLAB Editor.

## Highlights

This chapter discusses the following topics:

- File Operations
- Window Modes
- Templates
- Text Handling
- C Language Awareness

## **File Operations**

This section describes how to create, open, change, insert, save, backup, and close files with the MPLAB Editor.

### **Create A New File**

The *<u>File > New Source</u>* command opens a new, empty window in which you can type. The window does not initially have a file name. In order to save what it contains use *<u>File > Save As</u>*.

When MPLAB Editor creates the window, it applies the set of modes that are defined for "new files".

## **Open Existing Files From MPLAB Editor**

To edit one or more existing files, use the <u>*File > Open Source*</u> command. It opens a standard dialog to select the files to edit. If a selected file is already open, MPLAB Editor activates the window that is currently showing the file.

## Save A File To Disk

To save a file to disk you have a choice of two commands.

- Use <u>File > Save</u> to save the contents of the current window to disk. The MPLAB Editor replaces the file with the contents of the current window.
- Use <u>File > Save As</u> to supply the name of the disk file to write to. The MPLAB Editor confirms overwrites of existing files.

### **Close A File**

To close a file use the  $\underline{File > Close}$  command. MPLAB Editor prompts to either write the file to disk or to abandon the operation.

Use the *File > Close All* command to close all the open files.

## Window Modes

MPLAB Editor associates a set of Window Modes with every edit window. The modes can be set with the *Options > Default Editor Modes* commands.

The possible window modes are:

- **Auto Indenting** If this mode is on, MPLAB Editor automatically indents new lines to match the indentation of the previous line.
- Language This mode is a string which defines the language information MPLAB Editor should use for the window. Currently, you can set this only to C.
- **Overwrite** If this mode is on, MPLAB Editor replaces the characters under the cursor as you type.
- **Page Headers** If this mode is on, MPLAB Editor begins each printed page with a title showing the name of the file, the date and time, and the page number.
- **Soft Tabs** If this is mode on, MPLAB Editor inserts a suitable number of spaces to bring the cursor to the next tab stop, as defined by the Tab Size mode. Otherwise, MPLAB Editor inserts a single tab character into the file when you press the TAB key.
- **Show Line Numbers** If this mode is on, MPLAB Editor displays the number of each line in the window.
- **Strip Trailing Spaces** If this mode is on, MPLAB Editor removes any trailing space or tab characters from a line when you press **Enter**.
- **Tab Size** This mode is a numeric value that defines the width of a tab stop on the screen.
- Wrap Column This mode is a numeric value that defines the column for automatic text wrapping.
- **Wrap Enabled** If this mode is on, MPLAB Editor automatically wraps the current line at the Wrap Column selection.
- Wrap Long Lines If this mode is on, MPLAB Editor wraps lines that do not fit on the page when printing. Otherwise, MPLAB truncates long lines.

## **Templates**

Templates are pre-built sections of text that you can include in your files. They can help you avoid repetitive typing of the same information.

For example:

If documented C procedures begin by preceding each procedure with an explanatory section like this:

Rather than type the lines of asterisks every time, create a template containing the text:

The "<???>" characters are a template mark that shows where to type in extra items. The template marks can be searched for automatically, as will be explained shortly.

Templates are held in libraries called template files. MPLAB Editor can have multiple template files ready for use.

**Note:** Template files are not ordinary text files and cannot be edited directly.

You can attach a template file manually at any time, or you can have MPLAB Editor attach a set of templates automatically each time you invoke it.

## **Create a Template File**

A template file must be created before storing a template. Use the <u>Editor ></u> <u>Template > Create File</u> command to have MPLAB Editor create the file. Templates are auto-attached when created.

## Attach a Template File

A template file must be attached before accessing any templates in the template file. When you attach a template file, MPLAB Editor loads it into memory and constructs an index of the templates it contains.

Use the *Editor > Template > Attach File* command to attach a template file.

### **Automatically Attach Template Files**

MPLAB Editor can attach a template file automatically, rather than manually, with the <u>Editor > Template > Attach File</u> command. MPLAB Editor scans the current directory for a valid template file called AUTO.TPL and attaches it.

## **Create a Template**

Use the <u>Editor > Template > New</u> command to create a template. MPLAB Editor opens an edit window whose caption displays that it is a template.

Edit the window normally, using the <u>*Editor > Template > Insert Mark*</u> command to insert template marks.

The template must be stored in a template file to use it. However, MPLAB Editor does not save the template to disk until it receives <u>Editor > Template</u> <u>Save File</u> command.

### **Insert a Template**

MPLAB Editor can insert a template from any attached template file at the position of the cursor.

The <u>*Editor > Template > Insert*</u> command gives a list of all the attached template files. Select one of the template files to see a list of the templates it contains.

## **Using Templates**

Once inserted, use the <u>*Editor* > *Template* > *Find Mark*</u> command to search for template marks. Enter the desired text at the mark.

## **Edit a Template**

To change a template in a template file, attach the template file and use the *<u>Template > Edit</u>* command.

When you select a template, MPLAB Editor places it in an edit window. The window is the same as for a normal text file, but the caption shows that it is actually a template.

## **Delete a Template**

To delete a template from a template file, attach the template file and use the <u>*Editor > Template > Delete*</u> command.

## Save a Changed Template File

When you edit, create, or delete a template, MPLAB Editor changes a copy of the template in memory. MPLAB Editor does not change the template file on disk until you use the *Editor > Template > Save File* command

## **Detach a Template File**

Use the *Editor > Template > Detach File* command to remove a template file from memory.

## **Text Handling**

## **Type Text**

MPLAB Editor inserts text in either insert or strikeover mode. MPLAB Editor shows the mode as either "INS" or "OVR" on the status bar.

## Select Text

MPLAB Editor allows the user to select an area of text and operate on it. The selected area appears highlighted. Select text by using either a mouse, or with the keyboard in the standard Windows fashion.

## **Find Text**

To search for text in a window, place the cursor at the start point of the search, and select *Editor > Find*.

To repeat a search without using the Find Dialog, select *Editor > Repeat Find*.

## **Find Special Characters**

In the <u>Editor > Find</u> dialog you can search for special characters by using the escape sequence for the character. You can use "\n" for carriage returns, "\t" for tabs, and "\\" for backslashes.

## **Replace Text**

To search for text in a window and replace it, place the cursor at the start point, and use the *Editor > Replace* command.

## **Replace Special Characters**

In the <u>*Editor* > *Replace*</u> dialog you can replace special characters by using the escape sequence for the character. You can use "\n" for carriage returns, "\t" for tabs, and "\\" for backslashes.

### **Text Processing**

Although MPLAB Editor is intended to be used as a program text editor, it has several features that make it useful in general text editing.

#### **Automatic Text Wrapping**

When typing ordinary text, it may be convenient to have the program fit the text into the available line width. This would typically not be the case when editing a program source code file.

To change the text wrapping mode, double-click the left mouse button in the wrap area of the status bar. This area shows the text "No Wrap" when wrapping is not active. The double click action turns wrapping on. The status bar shows "Wr 72" when wrapping is enabled and set at column 72.

The points at which MPLAB Editor wraps a line vary with the language type defined for the window.

#### Language type "(none)" or "C"

MPLAB Editor breaks the line at the closest white space character or hyphen to the defined wrap column.

#### Language type "TeXt"

MPLAB Editor breaks the line at the closest white space character to the defined wrap column.

**Note:** MPLAB Editor wraps the line being typed only when the cursor is at the end of the line. If you move the cursor to somewhere within the line and enter text, MPLAB Editor does not wrap the line even if it extends past the wrap column.

### **Indenting And Unindenting Text**

When editing program source, indenting and unindenting source is very common. MPLAB Editor provides a facility to change the indentation level of either a single line, or of a larger block of text.

To indent a single line, place the cursor anywhere within it and use the <u>Editor</u> <u>> Text Indent</u> command. MPLAB Editor moves the entire text of the line to the right by one tab stop.

To un-indent a single line, place the cursor within it and use the <u>Editor > Text</u> <u>Un-Indent</u> command. MPLAB Editor moves the text left by one tab stop. MPLAB Editor does not alter a line that does not start with either a tab or a space.

#### **Changing Case**

Use the <u>Editor > Text > Uppercase</u> or the <u>Editor > Text > Lowercase</u> commands to change the case of a selected block of text.

#### **Handling Braces**

MPLAB Editor allows the user to manipulate brace characters such as brackets and parentheses, which often delimit sections of text or program sources.

The definition of a brace character varies, depending on the language type set in a window's window modes. For C, braces are defined as the characters that have syntactic meaning – opening braces are  $\{ \ [ or (, and closing braces are \} ] or ).$ 

For the language type "none," brace characters are defined as those commonly used in text or many other languages – opening braces are { [ ( or <, and closing braces are } ] ) or >.

To locate matching braces, place the cursor on one of the braces and use <u>*Editor > Match Brace*</u>. The cursor moves to the matching brace, respecting the level of nesting in the code.

#### **Undoing Edit Actions**

MPLAB Editor records edit actions and can reverse them with <u>Editor > Undo</u>. The undo depth is configurable by the user. For additional details, see the "*Customizing MPLAB after Installation*" appendix.

## C Language Awareness

When editing files that have language type set to "C", MPLAB Editor provides these facilities:

- MPLAB Editor always moves a "#" character typed in an otherwise empty line to column 1.
- MPLAB Editor moves a closing "}" brace typed in an otherwise empty line to the same column as the matching preceding opening brace "{" if the opening brace is the only character in its line.

For example:

```
11
           *****
******
// EXAMPLE.C
11
#include <PIC16C84.H>
void delay(void);
void main(void)
{
   unsigned int i,j;
   TRISB = 0xff;
   PORTB = 0;
   i = 0x1;
   while(1)
   {
      PORTB = i;
      if (i == 0x80)
          i = 0x1;
      else
          i <<= 1;
      TRISB = 0;
      delay();
      TRISB = 0xff;
      delay();
   }
}
void delay(void)
{
   int x, y;
   x = 0x3f;
   y = 0xff;
   while(x--)
   {
      while(y--)
         NOP();
   }
}
```



## Chapter 3. MPLAB Editor Menu Commands

## Introduction

The MPLAB Editor commands are supported by the following menu options:

File Menu	Commands for opening, viewing, saving, and closing files.	
Editor Menu	Text-manipulating commands.	
<b>Options Menu</b>	Commands for setting editor modes.	

# Highlights

This chapter discusses the following topics:

- File Menu
- Editor Menu
- Options Menu

## **File Menu**

Only the File Menu options printed in black are described in this section.

File		
New Source	<ctrl+n></ctrl+n>	Import
Open Source	<ctrl+0></ctrl+0>	Download to Memory
View		Download to Target Memory
Save	<ctrl+s></ctrl+s>	Copy from Target Memory
Save As		Load Simulator Stimulus
Save All		Export
Close		Save Trace Buffer
Close All	<shift+f9></shift+f9>	Save Hex File
		Print
		Print Setup
		Exit
		Most Recently Used Files

### **File Menu Commands**

#### New Source - Ctrl+N

This menu item creates a new, empty edit window. The window has no file name associated with it. The tab size and other settings are the values you set up for "new files" with the <u>Options > Default Editor Modes</u> command.

#### **Open Source – Ctrl+O**

Opens one or more existing files for editing. *File > Open Source* opens a dialog that allows you to open one or more existing files.

#### Steps

- 1. Use the Drive and Directory list boxes to select the disk drive and the directory.
- 2. Select the files you want to open in the File Name list box.

To add single files from the list, hold down the **Ctrl** key and click on the desired files.

To add a list of files, either hold the **Shift** key and click on the first and last file in the desired range, or click on the first file and drag down to the last file.

- or -

Type the name of the files to open in the File Name field.

- 3. If you want to open the files in read only mode, check the **Read Only** box. This affects all the files you open in this operation.
- 4. Click the **OK** box to open the files.

The "List Files of Type" list at lower left allows you to restrict the files shown in the list to those matching specific filename patterns. for example, "\*.asm" will list all files with the suffix ".asm".

Whether you close the dialog with the **OK** or the **Cancel** buttons, MPLAB Editor makes its current working directory the one in the dialog.

If a file is selected that is already being edited, MPLAB Editor activates the window showing the file.

#### View

Opens one or more existing files in read only mode. You can examine their contents, but not alter them. The <u>*File > View*</u> action is exactly as if you used the <u>*File > Open Source*</u> command and checked the Read Only box.

#### Save – Ctrl+S

Saves the contents of a file being edited to disk. If the current file is unnamed, MPLAB Editor prompts for a new filename. If a file of the same name exists, MPLAB Editor makes that the backup copy and saves the current file.

# **Chapter 3. MPLAB Editor Menu Commands**

#### Save As

Saves the contents of a file to disk, allowing you to specify the file name.

#### Steps

- 1. Use the Drive and Directory list boxes to select the disk drive and the directory where you want to save the file.
- 2. Either type the name of the file into in the File Name edit control, or select the name of an existing file you want to overwrite from the list box.
- 3. Click the **OK** box to save the data to the file. If you specify the name of a file that already exists, MPLAB Editor confirms the overwrite.

As previously noted, the List Files of Type list at the lower left allows you to restrict the files shown in the list to those matching specific filename patterns.

#### Save All

Saves all altered files, stores all altered templates into template files, and saves all altered template files in a single operation.

Clicking **Cancel** in any of the dialogs that occur in this process cancels the entire operation.

#### Close

Closes the file being shown in the current window. If you have changed the file and haven't saved the changes to disk, MPLAB Editor prompts you to save the changes, discard them, or cancel the close.

#### Close All – Shift+F9

Closes all the files that you're working on. If you have changed any of the files and haven't saved the changes to disk, MPLAB Editor prompts you to save the changes, discard them, or cancel the entire operation.

#### **Most-Recently-Used Files**

The MPLAB Editor adds the Most-Recently-Used Files list to the end of the File Menu. Whenever you open a file, MPLAB Editor records the file name in the list, ordering it so that the files that have been most recently used appear at the top. Any file in the list can be reopened simply by clicking on the menu item. This is a user configurable option. See "Customizing MPLAB After Installation" in the accompanying *MPLAB PICMASTER User's Guide*.

#### **More Files**

This menu item is the list of files that you have opened recently. You can configure the number of files that MPLAB Editor records. When the Most-Recently-Used Files list contains more files than can be displayed on the File Menu, MPLAB Editor adds an extra menu command, **More Files**.

The More Files dialog shows you the entire file list, and lets you pick the file you want to open. To open a file from the list, either double-click its name in the list with the left mouse button, or select the name and press the **Open** button.

## **Editor Menu**

The Editor Menu contains options that allow you to change text. If you have no files open, the menu options display in grey characters.

Editor	
Undo	Template
Cut	Attach File
Сору	Detach File
Paste	Create File
Select All	Save File
Select Word	Insert
Delete Line	Edit
Delete EOL	New
Goto Line	Store
Find	Store As
Replace	Delete
Repeat Find	Insert Mark
Repeat Replace	Find Mark
Match Brace	Text
	Transpose
	Upper Case
	Lower Case
	Indent
	Un-Indent

## **Editor Options**

#### Undo – Ctrl+Z

Undo the last edit action. When there is no edit action to undo, the menu command shows Can't Undo, and you cannot select the command.

The undo depth is configurable by the user. See "Customizing MPLAB After Installation" in the accompanying *MPLAB PICMASTER User's Guide*.

#### Cut – Ctrl+X

Deletes the highlighted text in the current window, placing it on the clipboard. After this operation, you can paste the deleted text into another MPLAB Editor window, or into another Window's application.

### Copy – Ctrl+C

Copies the highlighted text in the current window onto the clipboard. After this operation, you can paste the copied text into another MPLAB Editor window, or into another Window's application.

#### Paste – Ctrl+V

Pastes the contents of the clipboard into the current window at the position of the cursor. You can only perform this operation if the clipboard contains data in text format. MPLAB Editor does not support pasting of bitmaps or other clipboard formats.

### Select All

Highlights the entire contents of the current window.

#### Select Word – Left Mouse Button Double-Click

Highlights the word containing the cursor.

#### Delete Line – Ctrl+Shift+K

Deletes the entire line containing the cursor and moves the cursor to the start of the next line.

### Delete EOL – Ctrl+K

Deletes the text from the position of the cursor to the end of the line. If the cursor is at the start of a line that is completely empty, the next line is moved up to close the gap.

#### Goto Line – Ctrl+G

Moves the cursor to the start of a specific line.

This menu command allows you to specify either an absolute or a relative line number.

#### Find – F3

Searches the current window for a text string.

This command lets you specify search parameters. If there is highlighted text in the current window, the Find operation uses it as the search string.

#### Replace – F4

Searches the current window for a text string, and optionally replaces occurrences with another string.

This command lets you specify search parameters. If there is highlighted text in the current window, the Replace operation uses it as the search string.

#### **Repeat Find – Shift+F3**

Repeats the last search operation, without prompting for details.

#### Repeat Replace – Shift+F4

Repeats the last replace operation, without prompting for details.

#### Match Brace – Ctrl+B

This command locates braces that match each other. Position the cursor on either an opening or a closing brace.

### **Template Options**

This menu item contains options that relate to handling templates.

#### **Attach Template File**

This menu item attaches a template file, loading it into memory so the templates within it can be used. The command starts a dialog that allows you to browse the files on your disk for template files, \*.TPL.

#### **Detach Template File**

This command allows you to specify which attached template files you wish to detach. Once a template file is detached, you cannot use the templates it contains.

You cannot detach a template file if you're currently editing any of the templates it contains. You must save any changes you want to keep with the <u>Editor > Template > Store</u> or <u>Editor > Template > Store As</u> commands, and then close the windows.

#### **Create Template File**

This command creates a file of type \*.TPL. Template files contain binary information; this is the only method by which you can create them.

#### Save Template File

Saves a template file to disk after you have edited its contents.

#### **Insert Template**

Inserts a template from an attached template file into a window. The template is inserted into the current window at the current cursor position.

#### **Edit Template**

Edits a template contained in an attached template file.

MPLAB Editor creates a new window with the text of the template in it. After editing the template use the <u>Editor > Template > Store</u> command to update the template.

#### **New Template**

Creates a new, empty edit window for the new template.

#### Store

Stores a template in the template file. This command overwrites the previous version of the template in the template file.

#### **Store Template As**

Stores a template in a template file, allowing you to specify both the template file and the template.

If you select the name of an existing template, this command overwrites the previous template's contents. MPLAB Editor confirms before executing this command.

The template file on disk is not altered by this command. To save your template changes permanently, use <u>*Editor > Template > Save File.*</u>

### **Delete Template**

Deletes a template from an attached template file.

The template file on disk is not altered by this command. To save your template changes permanently, use *Editor > Template > Save File*.

#### **Insert Template Mark**

Inserts a template mark into the current template.

A template mark is a sequence of characters ("<??>") that marks a position in a template to put variable details.

#### **Find Template Mark**

Moves the cursor to the next template mark and highlights it.

## **Text Options**

This menu shows a list of text-related commands.

#### Transpose - Ctrl+T

Transposes the character to the right of the cursor and the character to the left of the cursor. This command has no effect if the cursor is positioned at the start or end of a line.

#### Uppercase

Changes all lowercase characters in the currently highlighted text to uppercase characters.

#### Lowercase

Changes all uppercase characters in the currently highlighted text to lowercase characters

#### Indent

Moves the one or more lines of text to the right by one tab stop.

MPLAB Editor indents all highlighted lines. If no lines are highlighted, MPLAB Editor indents only the line that the cursor is in.

#### Unindent

Moves one or more lines of text to the left by one tab stop. Lines that do not start with white space are not affected.

## **Options Menu**

This item on the menu bar contains various system configuration options. The options printed in black are described in this section.

Options	
Development Mode	Key Mappings
Window Setup	Environment Setup
Save Setup	Colors
Load Setup	Processor Setup
Default Configuration	Hardware
Default Editor Modes	PICMASTER I/O Port
Current Editor Modes	Multi-Processor
Reset Editor Modes	Simulator I/O Setup

### **Default Editor Modes**

This command sets the modes that MPLAB Editor uses for existing and new files.

#### Steps

1. Determine file type.

To set the default editor modes for files created with the  $\underline{File > New}$ <u>Source</u> command, click the **New Files** button.

To set the default editor modes for a given type of files click the **File Type** button, and either select the file type from the adjoining list, or

type a new one (with a leading ".") into the edit control. This mode is applied when you use the <u>*File > Open Source*</u>, <u>*File > View*</u>, and <u>*File > Save As*</u> commands,

To set default modes for file types not in the list set above, click the **Other Types** button.

- 2. Check the mode boxes in the dialog to specify display, formatting, printing, and file save modes.
- 3. Click the **Apply** button. You can set modes for other file types by returning to Step 1.
- 4. To record the mode settings permanently, click the **Save** button.

The modes you can set in the dialog are:

Display/Input Modes		
These settings control how text is displayed in the window and any special actions MPLAB Editor takes as you type.		
Auto Indent	Indents new lines to the same level as the preceding line.	
Strip Trailing Spaces	Removes white space from the end of a line whenever you press <b>Enter</b> .	
Show Line Numbers	Displays line numbers in the window.	
Overwrite	Overwrites the characters under the cursor with the characters you type. Otherwise MPLAB Editor inserts characters at the cursor.	
Language	Select from this list to associate a language with a window.	

Screen Formatting		
These settings affect tabs and text wrapping.		
Tab Size	Defines the width of a tab character.	
Soft Tabs	Inserts spaces instead of tabs when Soft Tabs is checked.	
Wrap Enabled	Automatically breaks the text at the column specified in Wrap Column.	

#### Printing

These settings affect how MPLAB Editor prints the text in the current window. These settings become the defaults for the *File > Print* command, but can be overridden at the time of printing.

Page Headers	Prints each page headed with the file name, the page number and the date.
Wrap Long Lines	Wraps lines that are too wide to fit the page to the next line during printing.

File Modes	
These settings are the file modes that apply to file.	
Read Only	With this mode set, you cannot change the file in any window showing it.
Backup When Saving	Makes a backup copy of any existing file of the same name when you write the file to disk.
Strip Ctrl+Z On Load	Automatically removes any <b>Ctrl+Z</b> character at the end of the file when it is loaded.
Add Ctrl+Z On Save	Automatically adds a <b>Ctrl+Z</b> character to the end of the file on disk when it saves the file.
No EOLN after last line	Writes an end-of-line terminator ( <b>CR-LF</b> or <b>LF</b> ) after the last character of the last line when it saves the file.

### **Current Editor Modes**

This command allows you to configure the modes that apply to the current window. For a full description of the sections below please see the Default Editor Modes section.

#### **Display/Input Modes**

These settings control how text is displayed in the window and any special actions MPLAB Editor takes as you type.

#### **Screen Formatting**

These settings affect tabs and text wrapping.

#### Printing

These settings affect how MPLAB Editor prints the text in the current window. These settings become the defaults for the <u>*File > Print*</u> command, but can be overridden at the time of printing.

#### **Current File**

These settings are file modes that apply to the current file.

### **Reset Editor Modes**

Resets the modes that apply to the current window to the default values.



## Chapter 4. MPLAB Editor Default Key Commands

## Introduction

This appendix describes the default key commands specific to the MPLAB Editor and lists the equivalent menu command (if any).

The key commands perform the most common operations quickly. These key commands can be modified to suit individual preferences. By default, no prefix keys are enabled, so that only single keys are available for mapping.

For a table that lists the keys with the names of the functions used in the key mapping dialog, see Appendix B: MPLAB Key Mapping Functions.

# Highlights

The categories for the default key commands are:

- Function Keys
- Movement Keys
- Control Keys
- Formatting and Editing Keys

## **Function Keys**

	•
F1	Opens the MPLAB Editor Help File at the Contents topic.
F2	Executes the <u>Debug &gt; Break Settings</u> command to open the Break Settings dialog.
F3	Executes the <i>Edit &gt; Find</i> command to find strings.
F4	Executes the <i>Edit &gt; Replace</i> command to replace strings.
F5	Executes the <u>Debug &gt; Halt</u> command to halt debugging.
F6	Executes the <u>Debug &gt; Reset</u> command to issue a reset to the emulated or simulated processor.
F7	Executes the <u>Debug &gt; Step</u> command to execute a single opcode from program memory.
F8	Executes the <u>Debug &gt; Step Over</u> command to step over a call instruction in program memory.
F9	Executes the <u>Debug &gt; Run</u> command to issue a run to the emulated/simulated processor.

F11	Executes the <u>Execute &gt; DOS Command To Window</u> command to run a DOS command and capture its output in a window.
Shift+F3	Executes the <i>Edit &gt; Repeat Last Find</i> command to repeat the last find operation.
Shift+F4	Executes the <i>Edit &gt; Repeat Last Replace</i> command to repeat the last replace operation.
Shift+F5	Executes the <u>Debug &gt; Halt Trace</u> command to halt the execution trace for the simulated processor.
Shift+F7	Executes the <u>Debug &gt; Step Trace</u> command to step the execution trace for the simulated processor.
Shift+F9	Executes the <i>File &gt; Close All</i> command to close all files and windows.
Ctrl+F2	Executes the <u>Project &gt; Open Project</u> command to open the Open Project Dialog.
Ctrl+F3	Executes the <i><u>Project &gt; Edit Project</u></i> command to open the Edit Project Dialog.
Ctrl+F4	Executes the <i>Project &gt; Make Project</i> command to initiate a "make" for the current project.
Ctrl+F5	Executes the <i>Project &gt; Build All</i> command to build all the source files for the current project.
Ctrl+F6	Executes the <u>Debug &gt; Development Mode</u> command to open the Development Mode dialog.
Ctrl+F7	Executes the <i>Options &gt; Environment Setup</i> to open the Environment Setup dialog.
Ctrl+F8	Executes the <u>Windows &gt; Show Symbol List</u> command to open the Symbol List dialog.
Alt+F4	Executes the <i>File &gt; Exit</i> command to end your MPLAB Editor session.
Shift+Ctrl+F2	Executes the <u>Debug &gt; Clear Program Memory</u> command to clear all of program memory to an "erased" state.
Shift+Ctrl+F3	Executes the <u>Debug &gt; System Reset</u> command to reset the entire emulator system.
Shift+Ctrl+F5	Executes the <u>Debug &gt; POR Reset Emulation</u> command to open the POR Reset dialog.

## **Movement Keys**

For these keys, adding Shift to the combination causes a selection to be extended.

Note that where the **Alt** key is combined with either an arrow key, or one of **Home**, **End**, **PgDn**, **PgUp**, Ins or **Del**, you must use the keys in the extended key areas, and not those in the numeric keypad.

hift+Up Mo	oves the cursor up by one line oves the cursor up by one line, extending the ghlighting
- hię	ghlighting
own M	
	oves the cursor down by one line
	oves the cursor down by one line, extending the lection
eft Mo	oves the cursor left by one character
	oves the cursor left by one character, extending e selection
Strl+Left Mo	oves the cursor left by one word
	oves the cursor left by one word, extending the lection
tight Mo	oves the cursor right by one character
-	oves the cursor right by one character, extending e selection
Strl+Right Mo	oves the cursor right by one word
-	oves the cursor right by one word, extending the lection
gDn Mo	oves the cursor down by one page
	oves the cursor down by one page, extending the lection
- J	oves the cursor to the start of the last line in the ndow
<b>U</b>	oves the cursor to the start of the last line in the ndow, extending the selection
gUp Mo	oves the cursor up by one page
<b>-</b> .	oves the cursor up by one page, extending the lection
01	oves the cursor to the start of the first line of the ndow
• •	oves the cursor to the start of the first line of the ndow, extending the selection
lome Mo	oves the cursor to the start of the line

Shift+Home	Moves the cursor to the start of the line, extending the selection
Ctrl+Home	Moves the cursor to the start of the file
Ctrl+Shift+Home	Moves the cursor to the start of the file, extending the selection
Alt+Home	Moves the cursor to the first non-white-space character in the current line
Alt+Shift+Home	Moves the cursor to the first non-white-space character in the current line, extending the selection
End	Moves the cursor to the end of the line
Shift+End	Moves the cursor to the end of the line, extending the selection
Ctrl+End	Moves the cursor to the end of the file
Ctrl+Shift+End	Moves the cursor to the end of the file, extending the selection

# **Control Keys**

move the cursor to the brace character matching the one currently under itCtrl+Shift+BExecutes the <i>Edit &gt; Text Match Brace Select</i> command to move the cursor to the brace character matching the one currently under it and highlight all the text between and including the brace charactersCtrl+CExecutes the <i>Edit &gt; Copy</i> text to the clipboardCtrl+GExecutes the <i>Edit &gt; Goto Line</i> cursor to a specific lineCtrl+HDeletes the character to the left of the cursorCtrl+IInserts a TAB character, or the required number of spaces to bring the cursor to the next TAB stop, depending on whether the current window's Window Mode is set for hard or soft tabs.Ctrl+KExecutes the <i>Edit &gt; Delete To End Of Line</i> command to delete the entire line that the cursor is inCtrl+NExecutes the <i>Edit &gt; Delete Line</i> command to delete the entire line that the cursor is inCtrl+NExecutes the <i>File &gt; New</i> command to create a new, empty edit windowCtrl+OExecutes the <i>File &gt; Open</i> command to open an		
Command to move the cursor to the brace character matching the one currently under it and highlight all the text between and including the brace charactersCtrl+CExecutes the <i>Edit &gt; Copy</i> command to copy selected text to the clipboardCtrl+GExecutes the <i>Edit &gt; Goto Line</i> cursor to a specific lineCtrl+HDeletes the character to the left of the cursorCtrl+IInserts a TAB character, or the required number of spaces to bring the cursor to the next TAB stop, depending on whether the current window's Window Mode is set for hard or soft tabs.Ctrl+KExecutes the <i>Edit &gt; Delete To End Of Line</i> command to delete everything from the cursor position to the end of the lineCtrl+NExecutes the <i>Edit &gt; Delete Line</i> command to delete the entire line that the cursor is inCtrl+NExecutes the <i>File &gt; New</i> command to copen an	Ctrl+B	move the cursor to the brace character matching the
Ctrl+GExecutes the Edit > Goto Line Cursor to a specific lineCtrl+HDeletes the character to the left of the cursorCtrl+IInserts a TAB character, or the required number of spaces to bring the cursor to the next TAB stop, depending on whether the current window's Window Mode is set for hard or soft tabs.Ctrl+KExecutes the Edit > Delete To End Of Line of the lineCtrl+Shift+KExecutes the Edit > Delete Line the entire line that the cursor is inCtrl+NExecutes the File > New empty edit windowCtrl+OExecutes the File > Open command to open an	Ctrl+Shift+B	command to move the cursor to the brace character matching the one currently under it and highlight all
cursor to a specific lineCtrl+HDeletes the character to the left of the cursorCtrl+IInserts a TAB character, or the required number of spaces to bring the cursor to the next TAB stop, depending on whether the current window's Window Mode is set for hard or soft tabs.Ctrl+KExecutes the <i>Edit &gt; Delete To End Of Line</i> command to delete everything from the cursor position to the end of the lineCtrl+Shift+KExecutes the <i>Edit &gt; Delete Line</i> command to delete the entire line that the cursor is inCtrl+NExecutes the <i>File &gt; New</i> command to create a new, empty edit windowCtrl+OExecutes the <i>File &gt; Open</i> command to open an	Ctrl+C	Executes the <u>Edit &gt; Copy</u> command to copy selected text to the clipboard
Ctrl+IInserts a TAB character, or the required number of spaces to bring the cursor to the next TAB stop, depending on whether the current window's Window Mode is set for hard or soft tabs.Ctrl+KExecutes the <i>Edit &gt; Delete To End Of Line</i> command to delete everything from the cursor position to the end of the lineCtrl+Shift+KExecutes the <i>Edit &gt; Delete Line</i> command to delete the entire line that the cursor is inCtrl+NExecutes the <i>Edit &gt; Delete Line</i> command to delete the entire line that the cursor is inCtrl+OExecutes the <i>File &gt; New</i> command to open an	Ctrl+G	Executes the <i>Edit &gt; Goto Line</i> command to move the cursor to a specific line
Spaces to bring the cursor to the next TAB stop, depending on whether the current window's Window Mode is set for hard or soft tabs.Ctrl+KExecutes the <u>Edit &gt; Delete To End Of Line</u> command to delete everything from the cursor position to the end of the lineCtrl+Shift+KExecutes the <u>Edit &gt; Delete Line</u> command to delete the entire line that the cursor is inCtrl+NExecutes the <u>File &gt; New</u> command to create a new, empty edit windowCtrl+OExecutes the <u>File &gt; Open</u> command to open an	Ctrl+H	Deletes the character to the left of the cursor
Ctrl+Shift+KExecutes the <u>Edit &gt; Delete Line</u> command to delete the entire line that the cursor is inCtrl+NExecutes the <u>File &gt; New</u> command to create a new, empty edit windowCtrl+OExecutes the <u>File &gt; Open</u> command to open an	Ctrl+I	spaces to bring the cursor to the next TAB stop, depending on whether the current window's Window
the entire line that the cursor is in         Ctrl+N       Executes the <i>File &gt; New</i> command to create a new, empty edit window         Ctrl+O       Executes the <i>File &gt; Open</i> command to open an	Ctrl+K	
Ctrl+OExecutes the <i>File &gt; Open</i> command to open an	Ctrl+Shift+K	
	Ctrl+N	
existing file	Ctrl+O	Executes the <i>File &gt; Open</i> command to open an existing file

# **Chapter 4. MPLAB Editor Default Key Commands**

Ctrl+Shift+O	Splits the current line at the position of the cursor, leaving the cursor unmoved
Ctrl+P	Executes the <i>File &gt; Print</i> command to print the file showing in the current window
Ctrl+Q	Executes the <u>Edit &gt; Text Insert ASCII Code</u> command to insert a character specified by its ASCII code number
Ctrl+S	Executes the <i>File &gt; Save</i> command to save the current file to disk
Ctrl+T	Executes the <u>Edit &gt; Text Transpose Characters</u> command to transpose the character under the cursor with the one to its left
Ctrl+V	Executes the <i>Edit &gt; Paste</i> command to paste data from the clipboard into the current window
Ctrl+W	Executes the <u>Window &gt; Select</u> command to let you choose between many open windows
Ctrl+Shift+W	Executes the <u>Edit &gt; Text Widen Brace Select</u> command to highlight all the text between and including the closest pair of brace characters that encompasses either the currently-highlighted text, or the character under the cursor if no text is selected
Ctrl+X	Executes the <i>Edit &gt; Cut</i> command to cut selected text to the clipboard
Ctrl+Z	Executes the <u>Edit &gt; Undo</u> command to undo the last edit action

# Formatting and Editing Keys

Enter	Inserts a new line
BackSpace	Deletes the character to the left of the cursor
Del	Deletes the character to the right of the cursor
Shift+Del	Executes the $Edit > Cut$ command to cut selected text to the clipboard
Ins	Toggles the current window between Insert and Overwrite modes
Shift+Ins	Executes the <u>Edit &gt; Paste</u> command to paste data from the clipboard into the current window
Ctrl+Ins	Executes the <u>Edit &gt; Copy</u> command to copy selected text to the clipboard
Keypad 5	Cancels the current selection, removing the highlight

Tab	Inserts a tab character, or the required number of
	spaces to bring the cursor to the next tab stop,
	depending on whether the current window's Window
	Mode is set for hard or soft tabs.



### Chapter 5. MPLAB Editor Error Messages

### Introduction

This appendix lists the most common error messages you might receive while using the MPLAB Editor.

### **Error Messages**

#### **Cannot Detach Template File**

A template file cannot be detached if one or more of the templates it contains is being edited.

#### Steps

- For each of the windows containing such templates, use the <u>Editor > Template > Store</u> or <u>Editor > Template > Store As</u> commands to save changes you want to keep. Then close the window with the <u>File > Close</u> command.
- Preserve the changes to the template file itself with the <u>Editor ></u> <u>Template > Save File</u> command.

The template file can then be detached.

#### **Create A Further Backup Level**

MPLAB Editor has detected that the file you are about to overwrite in this operation is already in a directory that contains backups of overwritten files. You may not wish to make a further backup of the file.

If you click the **Yes** button, MPLAB Editor continues as normal and makes a backup copy of the file before saving the new data to it. This involves creating a new sub-directory (called by default "\$MPLABBK") to contain it; depending on how deep your directory structure is at this point, the full path name of the directory, or of the backed up file within it, may exceed the limits imposed by the operating system.

If you click the **No** button, MPLAB Editor does not make a backup copy of the existing file, but simply overwrites it with the new data. The current contents of the disk file are irrevocably lost.

If you click the **Cancel** button, MPLAB Editor abandons the save operation and does not change the disk file.

#### **Default Printer Not Configured**

Whenever you exit from MPLAB Editor, the name of the printer you last used is recorded in the MPLAB Initialization file, MPLAB.INI. This device is used as the default printer for the MPLAB Editor the next time you run MPLAB.

While starting this session, MPLAB Editor has detected that the printer/port combination you used last time is no longer configured in your system. MPLAB Editor now uses the Windows default printer unless you specify otherwise. You can check that this is what you want, and correct it if not, by using the *File > Print Setup* command.

#### Invalid ASCII Code

You have entered an invalid ASCII code number in the dialog. Make sure that your entry is a decimal number, in the range 1 to 255.

#### **Invalid Line Number**

You have entered an invalid line number in the Line To Go To edit control. Check that what you have typed is either a numeric string, or is the word end.

If you have typed a number preceded by either + or - to specify a line number relative to the line that the cursor is now in, verify that the resulting line number is not less than 1, or greater than the number of the last line in the file.

#### Save Changes To File?

You have changed this file since the last time you saved it to disk. Before MPLAB Editor closes it, you have a chance to update the disk copy.

- Yes Click Yes to save the contents of the window to disk. If the file hasn't yet got a name, you'll see a dialog asking you to specify the file to save to.
- No Click No to irretrievably discard the changes you've made since the last save.
- **Cancel** Click **Cancel** to abandon the operation completely and return to editing the window.

#### Save Changes To Template?

You have changed this template since the last time you stored it in the inmemory copy of a template file. Before MPLAB Editor closes it, you have a chance to update the in-memory template file.

Yes Click Yes to save the contents of the window to an in-memory copy of the template file. If the template has not yet been named, MPLAB Editor will prompt you to specify a name and template file location.

- No Click No to discard the changes you've made since the last save.
- **Cancel** Click **Cancel** to abandon the operation completely and return to editing the window.



## **Chapter 6. MPLAB Text Editor Command Line Options**

## Introduction

This appendix describes command line options that you can use when starting MPLAB.

## **Command Line Options**

When you start MPLAB, you can give several options on the command line to control its actions. These options must follow the name of the MPLAB Text Editor executable itself, and precede any file names.

### **Options for Opening Files**

MPLAB	Starts MPLAB. The example below opens the files data.txt and results.txt in the current directory.
	Example: MPLAB data.txt results.txt
/v	The /v option opens files for editing in read-only mode.
	Example: MPLAB /v data.txt results.txt

### **Configuration Options**

**/k** The configuration options affect how MPLAB configures itself at start-up.

The /k option starts a new instance of MPLAB and loads the key mappings contained in the file mapping.key in the current directory.

By default, MPLAB will look for a file called MPLAB.KEY in the Windows directory at start-up, and will load the key mappings contained in MPLAB.KEY.

Example: MPLAB /k mapping.key

The /k option specifies a key map file to be loaded on start-up instead of the MPLAB.KEY file in the Windows directory.

Pathname can specify any valid key map file;

Enter the single character "-" (minus) to disable MPLAB from loading a key mapping, but use built-in default key map values.

Using this option will disable the default MPLAB.key file. The Save button in the key mapping dialog will not be available when you change the mapping. To save to this file, use the Save As button. To load mappings from a different file on start-up, or to force MPLAB to use the built-in defaults described in this help file, you can use the "/k" command line option to name any key map file. For example, starting MPLAB with a command line:

MPLAB/k c:\kmaps\uemap.key

would load the key mappings in the file c:\key\maps\uemap.key. If you have a key map file MPLAB.KEY, but want to use the built-in key mappings described in this help file instead, use the "/k" option and give the filename as the single character "-" (minus).

**Note:** When you use the "/k" option, the dialog started by the <u>Options ></u> <u>Key Mappings</u> command will not save changed mappings to the file you specify, or to the standard key mapping file, by default. When saving, you will need to give an explicit file name.



# Part 3 – Appendices

Appendix A.	DDE Linking Support	145
Appendix B.	MPLAB Key Mapping Functions	149
Appendix C.	On Line Support	157
Appendix D.	Customizing MPLAB After Installation	161
Appendix E.	MPLAB-SIM PIC16C5X Simulator Issues	165
Appendix F.	MPLAB-SIM PIC16CXX Simulator Issues	169
Appendix G.	MPLAB-SIM PIC17CXX Simulator Issues	179
Appendix H.	Upgrading from MPSIM to MPLAB-SIM	185
Glossary		193
Index		203



## **Appendix A. DDE Linking Support**

## Introduction

DDE Linking Support describes how to use the Microsoft Dynamic Data Exchange with MPLAB serving only as a client. As a client, MPLAB can transmit data to other programs, but cannot receive data. MPLAB supports warm DDE links.

# Highlights

DDE Linking Support covers the following topics:

- DDE Overview
- Typical Uses for DDE
- DDE Conventions
- DDE Warm Link Example

## **DDE Overview**

MPLAB supports Dynamic Data Exchange (DDE) with other Windows programs such as Microsoft Excel.<sup>®</sup> Currently MPLAB can serve only as a client. As a client, MPLAB can transmit data to other programs that ask for data, but MPLAB cannot receive data from other applications. MPLAB supports warm DDE links. Once you establish a permanent DDE link, MPLAB transmits data to the client whenever the processor encounters a breakpoint.

# **Typical Uses for DDE**

With the DDE support that MPLAB provides, you can use the emulator to transmit data from the following MPLAB sources:

- Trace Buffer Address or Data Fields
- File Registers
- Program Memory

You can plot the data transferred from the emulator to the client (Microsoft Excel could be a client). Plotting a histogram of an address field from the trace buffer represents profiling information of the executing program.

You could also get digitized data from an A/D converter on the target board and transfer this data into Data RAM of the processor. Once in Data RAM, you could establish a DDE link to transfer that data to a client such as Excel and plot the data in real time. When transferring data from Data RAM, the DDE item name would be "ram." See item 3, "ram" under Item Name below.

## **DDE Conventions**

Every Windows program that supports DDE can receive data from MPLAB. All Windows programs that support DDE need an application name, a topic, and an item name. To use the emulator to transmit data to client applications, you must use the following conventions:

Application Name: MPLAB

Topic Name:	File Name. Include the full path for the file name. The file name should be typed exactly as it appears in the Caption Bar of the Main Window.
Item Name:	Caption Bar of the Main Window. Currently MPLAB supports the following five Items:
	1. data <start> <end></end></start>
	The topic name "data" specifies the Data/Opcode field of the Real Time Trace Buffer. The <start> and <end> fields represent the start and end address of the trace buffer (0 to 8K). Example: data 100 200</end></start>
	2. address <start> <end></end></start>
	The topic name "address" specifies the address field of the Real Time Trace Buffer. The <start> and <end> fields represent the start and end address of the trace buffer (0 to 8K).</end></start>
	Example: address 100 200
	3. ram <start> <end></end></start>
	The topic name "ram" specifies the Data Ram (File Registers) of the processor. The <start> and <end> fields represent the start and end address of the Data Ram (in case of PIC17CXX series this would be 0 to 255).</end></start>
	Example: ram 50 100
	4. prog <start> <end></end></start>
	The topic name "prog" specifies the emulation program memory of the MPLAB. The <start> and <end> fields represent the start and end address of the Program Memory (in the case of the PIC16C54 series this would be 0 to 512). Example: prog 50 100</end></start>
	5. ext <start> <end></end></start>
	The topic, ext, specifies the External Trace Input logic probes (Tr0 - Tr7) on the emulator. The <start> and <end> fields represent Tr0 through Tr7.</end></start>

Example: ext 0 7

# **DDE Warm Link Example**

Establishing a warm link with Excel as the client is very simple. In the following example, assume that you want to establish a warm link for Data RAM (File Registers) locations 100 to 200.

- 1. In Excel, select a column of 100 cells.
- 2. With MPLAB running, type in the following formula:

= MPLAB | 'c:\MPLAB\DDEDEMO.HEX'!'RAM 100 200'

Where 'c:\MPLAB\DDEDEMO.HEX' is the name of a valid MPLAB \*.HEX file that can be loaded into MPLAB. It should be the \*.HEX file of the currently loaded project.

3. Press CNTRL+SHIFT+ENTER

The 100 cells selected in Excel get data from File Registers 100 to 200.

This procedure has established a permanent warm (hot) link for RAM. From now on, whenever the processor is launched and then halts (as a result of a breakpoint), values of Data RAM locations (from address 100 to 200) will automatically update in Excel.



# **Appendix B. MPLAB Key Mapping Functions**

# Introduction

This appendix lists the available MPLAB key mapping functions.

# **MPLAB Key Mapping Functions**

Key Map Function	Definition	Default Key Assignment
CursorBottomOfWindow	"Move Cursor to Bottom of Window"	Ctrl+PgDn
CursorBottomOfWindowSelect	"Move Cursor to Bottom of Window Selecting"	Ctrl+Shift+PgDn
CursorDown	"Move Cursor Down"	Down
CursorDownSelect	"Move Cursor Down Selecting"	Shift+Down
CursorEndOfFile	"Goto End of File"	Ctrl+End
CursorEndOfFileSelect	"Goto End of File Selecting"	Ctrl+Shift+End
CursorEndOfLine	"Move Cursor to End of Line"	End
CursorEndOfLineSelect	"Move Cursor to End of Line Selecting"	Shift+End
CursorLeft	"Move Cursor Left"	Left
CursorLeftSelect	"Move Cursor Left Selecting"	Shift+Left
CursorLeftWord	"Move Cursor Left by Word"	Ctrl+Left
CursorLeftWordSelect	"Move Cursor Left by Word Selecting"	Ctrl+Shift+Left
CursorPageDown	"Page Down"	PgDn
CursorPageDownSelect	"Page Down Selecting"	Shift+PgDn
CursorPageUp	"Page Up"	PgUp
CursorPageUpSelect	"Page Up Selecting"	Shift+PgUp
CursorRight	"Cursor Right"	Right
CursorRightSelect	"Cursor Right Selecting"	Shift+Right
CursorRightWord	"Cursor Right by Word"	Ctrl+Right
CursorRightWordSelect	"Cursor Right by Word Selecting"	Ctrl+Shift+Right
CursorStartOfFile	"Cursor Start of File"	Ctrl+Home
CursorStartOfFileSelect	"Cursor Start of File Selecting"	Ctrl+Shift+Home
CursorStartOfLine	"Cursor Start of Line"	Home
CursorStartOfLineSelect	"Cursor Start of Line Selecting"	Shift+Home
CursorStartOfText	"Cursor Start of Text"Alt+Home	Alt+Home
CursorStartOfTextSelect	"Cursor Start of Text Selecting"	Alt+Shift+Home
CursorTopOfWindow	"Cursor Top of Window"	Ctrl+PgUp

Key Map Function	Definition	Default Key Assignment
CursorTopOfWindowSelect	"Cursor Top of Window Selecting"	Ctrl+Shift+PgUp
CursorUp	"Cursor Up"	Up
CursorUpSelect	"Cursor Up Selecting"	Shift+Up
DebugAnimate	"Animate"	
DebugBreak	"Set Break Settings"	F2
DebugCenterDebug	"Center Debug Location"	
DebugChangePC	"Change Program Counter"	
DebugClearAll	"Clear All Qualifiers"	
DebugClearMemory	"Clear Program Memory"	Ctrl+Shift+F2
DebugConditionalBreak	"Conditional Break"	
DebugExecuteOpcode	"Execute an Opcode"	
DebugHalt	"Halt the Processor"	F5
DebugHaltTrace	"Halt the Trace"	Shift+F5
DebugPerformanceAnalysis	"Performance Analysis"	
DebugPORReset	"Power-On-Reset"	Ctrl+Shift+F5
DebugStep	"Step"	F6
DebugReset	"Reset Processor"	F9
DebugRun	"Run"	F7
DebugStepOver	"Step Over"	F8
DebugStepTrace	"Step The Trace Window"	Shift+F7
DebugAsyncStim	"Asynchronous Stimulus	
DebugClockStim	"Clock Stimulus"	
DebugPinStim	"Pin Stimulus"	
DebugRegStim	"Register Stimulus"	
DebugSystemReset	"System Reset"	Ctrl+Shift+F3
DebugTrace	"Trace Settings"	
DebugTrigger	"Trigger Settings"	
DebugUpdateRegisters	"Update Registers"	
EditCancelSelection		keypad5
EditClearUndo	"Forgets details of all stored undo actions"	
EditCopy	"Copies highlighted text to the clipboard"	Ctrl+C
		Ctrl+Ins
EditCut	"Cuts highlighted text to the clipboard"	Ctrl+X
		Shift+Del

# **Appendix B. MPLAB Key Mapping Functions**

Key Map Function	Definition	Default Key Assignment
EditDeleteBackwards	"Delete Character Backwards"	Backspace
		Ctrl+H
EditDeleteForwards	"Delete Forwards"	Del
EditDeleteLine	"Deletes the entire line containing the cursor"	Ctrl+Shift+K
EditDeleteSelection	"Delete Selection"	
EditDeleteToEndOfLine	"Deletes from the cursor to end of line"	Ctrl+K
EditFind	"Searches for a text string"	F3
EditGotoLine	"Moves the cursor to a specific line"	Ctrl+G
EditInsertHardTab	"Insert Hard Tab"	
EditInsertSoftTab	"Insert Soft Tab"	
EditInsertTab	"Insert Tab"	Tab
		Ctrl+I
EditMarkUnchanged	"Mark File As Unchanged"	
EditNewLine	"Insert New Line"	Enter Ctrl+M
EditPaste	"Pastes the clipboard at the cursor position"	Ctrl+V Shift+Ins
EditRepeatLastFind	"Repeats the last search action exactly"	Shift+F3
EditRepeatLastReplace	"Repeats the last replace action exactly"	Shift+F4
EditReplace	"Replaces a text string"	F4
EditSelectAll	"Highlights all the text in the current window"	
EditSelectLine	"Select Line"	
EditSelectWord	"Highlights the word containing the cursor"	
EditShowCursor	"Scrolls the window to bring the cursor into view"	
EditShowNextLine	"Show Next Line"	
EditShowNextPage	"Show Next Page"	
EditShowPreviousLine	"Show Previous Line"	
EditShowPreviousPage	"Show Previous Page"	
EditSplitLine	"Split Line"	Ctrl+Shift+O
EditTextIndent	"Moves text right by one tab stop"	
EditTextInsertASCIICode	"Inserts an arbitrary character code"	Ctrl+Q
EditTextLowercaseSelection	"Converts the highlighted text to lower case"	
EditTextMatchBrace	"Moves to a matching brace character"	Ctrl+B
EditTextMatchBraceSelect	"Moves to a matching brace character and highlights"	Shift+Ctrl+B

Key Map Function	Definition	Default Key Assignment
EditTextTransposeCharacters	"Swaps the characters to left and right of the cursor"	Ctrl+T
EditTextUnIndent	"Moves text left by one tab stop"	
EditTextUppercaseSelection	"Converts the highlighted text to upper case"	
EditTextWidenBraceSelect	"Highlights the next largest braced area of text"	Shift+Ctrl+W
EditUndo	"Undoes the last edit action"	Ctrl+Z
ExecDosCommand	"Runs a DOS command and captures output"	F11
ExecRepeatDosCommand	"Repeats the last DOS command-with-capture"	Ctrl+F11
FileAbandon	"Abandon File"	
FileClose	"Closes the file in the current window"	
FileCloseAll	"Closes all open files"	Shift+F9
FileExit	"Ends your MPLAB session"	Alt+F4
FileImportDownloadToMemory	"Download A Hex file to the Engine"	
FileImportDownloadToTarget	"Download A Hex file to the Target"	
FileImportReadTarget	"Copy Engine Memory to Target"	
FileInsert	"Inserts a file at the position of the cursor"	
FileName	"Changes the file name for the current window"	
FileNew	"Creates a new, empty edit window"	Ctrl+N
FileOpen	"Opens an existing file"	Ctrl+O
FilePrint	"Prints the current file"	Ctrl+P
FilePrintSetup	"Changes details of the current printer"	
FileSave	"Saves the current file to disk"	Ctrl+S
FileSaveAll	"Saves all open files to disk"	
FileSaveAs	"Saves the current file to disk"	
FileSaveHex	"Save Hex File"	
FileSaveTrace	"Save Trace File"	
FileSimulatorStimulus	"Load Simulator Stimulus File"	
FileView	"Opens an existing file in read-only mode"	
FileWrite	"Writes the current file to disk"	
HelpAbout	"Gives information about this MPLAB version"	
HelpBugs	"MPLAB Bug List"	
HelpCommands	"Gives help on MPLAB commands"	
HelpContents	"Enters the MPLAB help file at the Contents screen"	F1
HelpEditor	"Editor Help"	
HelpMpasm	"MPASM Help"	

# **Appendix B. MPLAB Key Mapping Functions**

Key Map Function	Definition	Default Key Assignment
НеІрМрс	"MPLAB-C Help"	
HelpOnHelp	"Gives help on using the help system"	
HelpPIC16/17	"PIC16/17 Users Guide"	
HelpReleaseNotes	"Release Notes"	Shift+F1
OptionsColors	"Edit Color Options"	
OptionsCommunicationsPort	"Set Communications Address"	
OptionsCurrent	"Sets modes for the current window and file"	
OptionsDefault	"Sets default modes for files"	
OptionsDevelopmentMode	"Select Development Mode"	Ctrl+F6
OptionsEnvironmentSetup	"Setup Environment"	Ctrl+F7
OptionsHardware	"Select Hardware Options"	
OptionsKeyMapping	"Changes the mapping of keys to commands"	
OptionsLoadSetup	"Load Setup File"	
OptionsMultiProcessor	"Setup Multi-Processor"	
OptionsPreferences	"Sets MPLAB configuration options"	
OptionsResetModes	"Sets the current file/window modes to default values"	
OptionsSaveSetup	"Save Setup File"	
OptionsScreenFontANSI	"Sets the screen font to the standard ANSI fixed- pitch font"	
OptionsScreenFontOEM	"Sets the screen font to the standard OEM font"	
OptionsScreenFontOther	"Selects the screen font from all available fixed- pitch fonts"	
OptionsScreenFontSystem	"Sets the screen font to the standard system font"	
OptionsSimulatorIOSetup	"Setup Simulator I/O"	
OptionsToggleInsertMode	"Toggle Insert Mode"	Ins
OptionsToggleLineNumbers	"Toggle Line Numbers"	
OptionsToggleStatusBar	"Hides or shows the status bar"	
OptionsToolbarBottom	"Moves the tool bar to the bottom of the window"	
OptionsToolbarFloat	"Makes the tool bar a floating window"	
OptionsToolbarHide	"Makes the tool bar invisible"	
OptionsToolbarLeft	"Moves the tool bar to the left of the window"	
OptionsToolbarRight	"Moves the tool bar to the right of the window"	
OptionsToolbarShow	"Makes the tool bar visible"	
OptionsToolbarTop	"Moves the tool bar to the top of the window"	
ProjectBuildAll	"Build Full Project"	Ctrl+F10

Key Map Function	Definition	Default Key Assignment
ProjectCloseProject	"Close Project"	
ProjectCompileSingle	"Compile Single File"	Alt+F10
ProjectEditProject	"Edit the Project Definition"	Ctrl+F3
ProjectMakeProject	"Make the Current Project"	F10
ProjectMakeSetup	"Setup the Project Make"	
ProjectNewProject	"Create a New Project"	
ProjectOpenProject	"Open a Project"	Ctrl+F2
ProjectSaveProject	"Save the Current Project"	
SwapToolbar	"Swap Toolbar"	
SysSetMenuMode	"Set Menu Mode"	
TemplateDelete	"Deletes a template from a template file"	
TemplateEdit	"Edits a template from a template file"	
TemplateFileAttach	"Loads a template file for use"	
TemplateFileCreate	"Creates an empty template file"	
TemplateFileDetach	"Releases an attached template file"	
TemplateFileSave	"Saves an altered template file to disk"	
TemplateFindMark	"Searches for a template marker in the current window"	
TemplateInsert	"Inserts a template at the position of the cursor"	
TemplateInsertMark	"Inserts a template marker at the position of the cursor"	
TemplateNew	"Creates a new window for a template"	
TemplateStore	"Saves a template into a template file"	
TemplateStoreAs	"Saves a template into a template file"	
ToolsEmulatorConfiguration	"Setup Emulator Configuration"	
ToolsProgramHeader	"Program Emulator Header"	
ToolsProgramPod	"Program Emulator Pod"	
ToolsVerifyEmulator	"Verify Emulator Components"	
WindowAbsoluteListing	"Absolute Listing"	
WindowArrangelcons	"Arranges all iconic windows neatly"	
WindowCascade	"Arranges windows in a cascade pattern"	
WindowClose	"Closes the current window"	Ctrl+F4
WindowDuplicate	"Makes a duplicate of the current window"	
WindowEeprom	"EEPROM Memory Window"	
WindowFileRegisters	"File Register Memory"	
WindowIconize	"Iconize Window"	

# Appendix B. MPLAB Key Mapping Functions

Key Map Function	Definition	Default Key Assignment
WindowIconizeAll	"Makes all windows into icons"	
WindowLoadWatch	"Load a Watch Window"	
WindowMaximize	"Maximize Window"	
WindowModify	"Modify Window"	
WindowNewWatch	"Create New Watch Window"	
WindowNext	"Activates the next non-iconic window"	
WindowProgramMemory	"Program Memory Window"	
WindowRestore	"Restore Window"	
WindowSelect	"Chooses a window to activate from a list"	Ctrl+W
WindowSpecialFunctionRegisters	"Special Functions Register Window"	
WindowStack	"Stack Window"	
WindowStopwatch	"Stopwatch Window"	
WindowSymbolList	"Symbol List Window"	Ctrl+F8
WindowTileHorizontal	"Tiles windows to maximize height"	
WindowTileVertical	"Tiles windows to maximize width"	
WindowTrace	"Trace Window"	
WindowWiden	"Maximizes the width of the current window"	



### Appendix C. On Line Support

#### Introduction

Microchip provides two methods of on-line support. These are the Microchip BBS and the Microchip World Wide Web (WWW) site.

Use Microchip's Bulletin Board Service (BBS) to get current information and help about Microchip products. Microchip provides the BBS communication channel for you to use in extending your technical staff with microcontroller and memory experts.

To provide you with the most responsive service possible, the Microchip systems team monitors the BBS, posts the latest component data and software tool updates, provides technical help and embedded systems insights, and discusses how Microchip products provide project solutions.

The web site, like the BBS, is used by Microchip as a means to make files and information easily available to customers. To view the site, the user must have access to the Internet and a web browser, such as Netscape or Microsoft Explorer. Files are also available for FTP download from our FTP site.

### **Connecting to the Microchip Internet Web Site**

The Microchip web site is available by using your favorite Internet browser to attach to:

#### www.microchip.com

The file transfer site is available by using an FTP service to connect to:

#### ftp.mchip.com/biz/mchip

The web site and file transfer site provide a variety of services. Users may download files for the latest Development Tools, Datasheets, Application Notes, User's Guides, Articles and Sample Programs.

A variety of Microchip specific business information is also available, including listings of Microchip sales offices, distributors and factory representatives. Other data available for consideration is:

- Latest Microchip Press Releases
- Technical Support Section with Frequently Asked Questions
- Design Tips
- Device Errata
- Job Postings
- Microchip Consultant Program Member Listing
- · Links to other useful web sites related to Microchip Products

#### **Connecting to the Microchip BBS**

Connect worldwide to the Microchip BBS using either the Internet or the CompuServe<sup>®</sup> communications network.

Internet: You can telnet or ftp to the Microchip BBS at the address mchipbbs.microchip.com

<u>CompuServe Communications Network</u>: In most cases, a local call is your only expense. The Microchip BBS connection does not use CompuServe membership services, therefore

#### You do not need CompuServe membership to join Microchip's BBS.

There is **no charge** for connecting to the BBS, except for a toll charge to the CompuServe access number, where applicable. You do not need to be a CompuServe member to take advantage of this connection (you never actually log in to CompuServe).

The procedure to connect will vary slightly from country to country. Please check with your local CompuServe agent for details if you have a problem. CompuServe service allow multiple users at baud rates up to 14400 bps.

The following connect procedure applies in most locations.

- 1. Set your modem to 8-bit, No parity, and One stop (8N1). This is not the normal CompuServe setting which is 7E1.
- 2. Dial your local CompuServe access number.
- 3. Depress **<Enter**→ and a garbage string will appear because CompuServe is expecting a 7E1 setting.
- 4. Type +, depress **<Enter**, **>** and Host Name: will appear.
- 5. Type **MCHIPBBS**, depress **<Enter**→**>** and you will be connected to the Microchip BBS.

In the United States, to find the CompuServe phone number closest to you, set your modem to 7E1 and dial (800) 848-4480 for 300-2400 baud or (800) 331-7166 for 9600-14400 baud connection. After the system responds with Host Name:, type **NETWORK**, depress **<Enter**, > and follow CompuServe's directions.

For voice information (or calling from overseas), you may call (614) 723-1550 for your local CompuServe number.

#### Using the Bulletin Board

The bulletin board is a multifaceted tool. It can provide you with information on a number of different topics.

- Special Interest Groups
- Files
- Mail
- Bug Lists

#### **Special Interest Groups**

Special Interest Groups, or SIGs as they are commonly referred to, provide you with the opportunity to discuss issues and topics of interest with others that share your interest or questions. SIGs may provide you with information not available by any other method because of the broad background of the PIC16/17 user community.

There are SIGs for most Microchip systems, including:

- MPASM
- TrueGauge<sup>®</sup>
- PRO MATE<sup>™</sup>
- *fuzzy*TECH<sup>®</sup>-MP
  ASSP
- PICSTART<sup>®</sup>
- UtilitiesBugs
- MTE1122
   MPLAB

These groups are monitored by the Microchip staff.

#### Files

Microchip regularly uses the Microchip BBS to distribute technical information, application notes, source code, errata sheets, bug reports, and interim patches for Microchip systems software products. Users can contribute files for distribution on the BBS. For each SIG, a moderator monitors, scans, and approves or disapproves files submitted to the SIG. No executable files are accepted from the user community in general to limit the spread of computer viruses.

#### Mail

The BBS can be used to distribute mail to other users of the service. This is one way to get answers to your questions and problems from the Microchip staff, as well as keeping in touch with fellow Microchip users worldwide.

Consider mailing the moderator of your SIG, or the SYSOP, if you have ideas or questions about Microchip products, or the operation of the BBS.

#### **Software Releases**

Software products released by Microchip are referred to by version numbers. Version numbers use the form:

xx.yy.zz

Where  $\mathbf{x}\mathbf{x}$  is the major release number,  $\mathbf{y}\mathbf{y}$  is the minor number, and  $\mathbf{z}\mathbf{z}$  is the intermediate number.

#### **Intermediate Release**

Intermediate released software represents changes to a released software system and is designated as such by adding an intermediate number to the version number. Intermediate changes are represented by:

- Bug Fixes
- Special Releases
- Feature Experiments

Intermediate released software does not represent our most tested and stable software. Typically, it will not have been subject to a thorough and rigorous test suite, unlike production released versions. Therefore, users should use these versions with care, and only in cases where the features provided by an intermediate release are required.

Intermediate releases are primarily available through the BBS.

#### **Production Release**

Production released software is software shipped with tool products. Example products are PRO MATE, PICSTART, and PICMASTER. The Major number is advanced when significant feature enhancements are made to the product. The minor version number is advanced for maintenance fixes and minor enhancements. Production released software represents Microchip's most stable and thoroughly tested software.

There will always be a period of time when the Production Released software is not reflected by products being shipped until stocks are rotated. You should always check the BBS or the WWW for the current production release.

### Systems Information and Upgrade Hot Line

The Systems Information and Upgrade Line provides system users a listing of the latest versions of all of Microchip's development systems software products. Plus, this line provides information on how customers can receive any currently available upgrade kits. The Hot Line Numbers are: 1-800-755-2345 for U.S. and most of Canada, and 1-602-786-7302 for the rest of the world.

These phone numbers are also listed on the "Important Information" sheet that is shipped with all development systems. The hot line message is updated whenever a new software version is added to the Microchip BBS, or when a new upgrade kit becomes available.



## Appendix D. Customizing MPLAB After Installation

## Introduction

MPLAB uses the initialization file, MPLAB.INI, to record the values that carry over from one session to another. You can also enter various options in MPLAB.INI to customize how MPLAB operates.

MPLAB.INI is held in your Windows directory, and can be edited with MPLAB or any other editor. However, MPLAB will change many parts of the file when it exits. You should not edit sections of MPLAB.INI with the MPLAB Editor.

#### **Initialization File Format**

The information contained in MPLAB.INI is in the standard format used by all Windows applications. MPLAB.INI is divided into sections with section titles enclosed by square brackets. The editable section of MPLAB.INI is in the options section.

All lines following a section name are treated as a group. Each section contains lines of the general form:

key string=argument,argument,....

The key string is text (which may not include spaces) that MPLAB uses to locate a particular setting. The comma-separated arguments that follow the "=" sign provide the data relevant to the key.

**Note:** Both the section (options) and the key string are case sensitive. If you place a key string in the list, make sure to use all lower case letters, otherwise MPLAB will not find the change.

[options]

mru-list-size=5

mru-files-shown=5

open-maximized=0

allow-save-always=0

backup-mode=5



# **Options Section**

This section contains various items that allow you to customize MPLAB to suit your own preferences. MPLAB never alters the contents of this section itself.

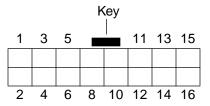
You can safely use MPLAB to alter the contents of this section. Any changes you make will not take effect until you terminate MPLAB and restart.

Key String	Possible Values	Default
allow-save-always	0 - <i><u>File &gt; Save</u></i> enabled only after a file is changed. 1 - <u>File &gt; Save</u> always enabled.	0
auto-file-action	<ul> <li>0 - Perform no action when MPLAB opened.</li> <li>1 - Perform a <i>File &gt; New</i> when MPLAB opened.</li> <li>2 - Perform a <i>File &gt; Open</i> when MPLAB opened.</li> </ul>	0
backup-directory	Eight character name using valid DOS characters (see backup- mode).	\$MPLABBK
backup-mode	<ul> <li>0 - Backups are in same directory as source, but with extension "*.\$\$\$".</li> <li>1 - Backups are in subdirectory of source file using name from backup-directory option</li> <li>5 - Create no backups.</li> </ul>	0
deselect-on-copy	<ul><li>0 - Leave selection highlighted after copy.</li><li>1 - Remove highlight after copy.</li></ul>	0
dragdrop-flip	<ul> <li>0 - Drag-and-drop performs a move operation by default and copy if Ctrl key is pressed.</li> <li>1 - Drag-and-drop copies by default, and moves with Ctrl key.</li> </ul>	0
max-undo-actions	Number of undo actions, 8 to 128	32
max-vertical-tile	<ul><li>0 - Leave room for one row of icons at bottom of screen</li><li>1 - Cover entire screen</li></ul>	0
minimize-on-empty	<ul><li>0 - Do nothing when all child windows are closed</li><li>1 - Minimize MPLAB when all child windows are closed</li></ul>	0
mru-files-shown	Number of files shown on File menu, 0 to 8.	5
mru-list-size	Number of files MPLAB will record for File menu, 0 to 64	5
open-maximized	<ul> <li>0 - Open window in a default state.</li> <li>1 - Open window maximized always.</li> <li>2 - Open window maximized if current window is maximized.</li> <li>11 - Open window maximized if MPLAB is maximized.</li> <li>12 - Open window maximized if MPLAB and current window are both maximized.</li> </ul>	0

# Appendix D. Customizing MPLAB After Installation

save-clears-undo	0 - Do not clear undo buffer on save. 1 - Clear undo buffer on save	1
save-find-strings	<ul> <li>0 - Do not record find strings in MPLAB.INI file.</li> <li>1 - Record last eight string used in Find (or Replace) dialog in MPLAB.INI file.</li> </ul>	0
select-search-match	<ul><li>0 - Do not highlight matching string when found.</li><li>1 - Highlight matching string when found.</li><li>Note: You can override this value in the Find Dialog.</li></ul>	1
sound-beep	<ul> <li>0 - Do not generate beeps for error messages.</li> <li>1 - Generate beeps for error messages.</li> <li>Note: MPLAB always generates beeps for conditions where there is no other error indication, ie. attempting to insert something into a read-only window.</li> </ul>	1
start-maximized	<ul> <li>0 - Start MPLAB in a default state.</li> <li>1 - Start MPLAB in a maximized state.</li> <li>Note: If this options does not exist in MPLAB.INI, MPLAB starts in the state it was at the end of the last session.</li> </ul>	0
track-vertical- thumbtack	<ul><li>0 - Update text when vertical scroll bar is released.</li><li>1 - Scroll text to follow vertical scroll bar.</li></ul>	1
use-dragdrop	0 - drag-and-drop not available. 1 - drag-and-drop available.	1

## Logic Probe Connector Pin Outs





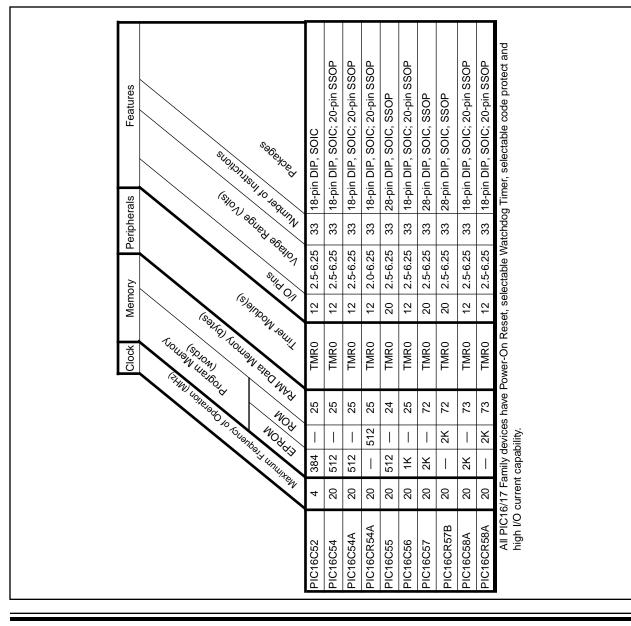
# Appendix E. PIC16C5X Simulator Issues

## Introduction

This appendix discusses I/O pins, interrupts, registers, peripherals, modes, and conditions for using the PIC16C5X family of microcontrollers.

# **PIC16C5X Selection Table**

At the time of printing, the following selection table contains the current list of processors available in the PIC16C5X family.



# I/O Pins

When modifying pins either manually or via the stimulus file, use the following pin names only. These are the only ones that the MPLAB-SIM simulator recognizes as valid I/O pins. Because the pinout is device-specific, some pins (for example RC0 on a PIC16C54) will not be available on all parts in this family.

- MCLR
- T0CKI
- RA0-RA3
- RB0-RB7
- RC0-RC7

These pin names can be used in the Modify window (*Window > Modify*) and in stimulus files.

# **CPU Mode**

#### **Reset and Sleep Conditions**

All reset conditions are supported by the MPLAB-SIM simulator.

An  $\overline{\text{MCLR}}$  reset during normal operation or during  $\overline{\text{SLEEP}}$  can easily be simulated by driving the  $\overline{\text{MCLR}}$  pin low (and then high) via the stimulus file or by using  $\underline{\text{Debug} > \text{Run} > \text{Reset}}$ .

A <u>WDT time-out reset</u> is simulated when WDT is enabled and proper prescaler is set (by initializing OPTION register appropriately) and WDT actually overflows. WDT time-out period (with prescale = 1) is approximated at 18 ms (to closest instruction cycle multiple).

The Time-out  $(\overline{TO})$  and Power-down  $(\overline{PD})$  bits in the Status register reflect appropriate reset condition. This feature is useful for simulating various power-up and time out forks in the user code.

### Watch Dog Timer

The Watchdog timer is fully simulated in the MPLAB-SIM simulator. Because it is fuse-selectable on the device, it must be enabled by <u>Options > Processor</u> <u>Setup > Hardware</u> in the MPLAB-SIM simulator. The period of the WDT is determined by the prescaler settings in the OPTION register. The basic period (with prescaler = 1) is approximated at 18 ms (to closest instruction cycle multiple).

## **Peripherals**

### **Peripherals supported**

Along with providing core support, the TIMER0 timer/counter module is fully supported. In both internal and external clock modes. The prescaler is made readable and writable as 'T0PRE' symbol.

It is important to remember that because the MPLAB-SIM simulator executes on instruction cycle boundaries, resolutions below 1 Tcy cannot be simulated.



## Appendix F. PIC16CXX Simulator Issues

## Introduction

This appendix discusses I/O pins, interrupts, registers, peripherals, modes, and conditions for using the PIC16CXX (PIC16C6X, PIC16C62X, PIC16C7X and PIC6C8X) and PIC14000 families of microcontrollers.

## **Product Selection Tables**

At the end of this Appendix there are selection tables which contain the current list of microcontrollers available in the PIC16CXXX and PIC14000 families.

# I/O Pins

The PIC16CXXX family of devices has I/O pins multiplexed with other peripherals (and therefore referred by more than one name). When modifying pins either manually or via the stimulus file, use the following pin names only. These pin names are the only ones that the MPLAB-SIM simulator recognizes as valid I/O pins. (Pins are only available as described in the data sheet of the specific device.)

- MCLR
- RA0-RA5
- RB0-RB7
- RC0-RC7
- RD0-RD7
- RE0-RE7

These pin names can be used in the Modify window (*Window > Modify*) and in stimulus files.

## Interrupts

The MPLAB-SIM simulator supports all interrupts on the PIC16CXXX family. (Peripherals are only available as described in the data sheet of the particular device.)

- Timer0 overflow
- Timer1 overflow
- Timer2
- CCP1
- CCP2
- SSP (in SPI mode ONLY)

- Change on Port RB <7:4>
- External interrupt from RB0/INT pin
- Parallel Slave Port
- Comparators
- USART
- A/D complete
- EEPROM write complete

# **CPU Model**

## **Reset Conditions**

All reset conditions are supported by the MPLAB-SIM simulator.

An <u>MCLR reset during normal operation or during SLEEP</u> can easily be simulated by driving the <u>MCLR</u> pin low (and then high) via the stimulus file or by using MPLAB <u>Debug > Run > Reset</u>.

The Time-out  $(\overline{TO})$  and Power-down  $(\overline{PD})$  bits in the Status register reflect appropriate reset condition. This feature is useful for simulating various power-up and time out forks in the user code.

### Sleep

The MPLAB-SIM simulator simulates the SLEEP instruction, and will appear "asleep" until a wake-up from sleep condition occurs. For example, if the Watchdog timer has been enabled, it will wake the processor up from sleep when it times out (depending upon the prescaler setting in the OPTION register).

Another example of a wake-up-from-sleep condition, would be Timer1 wake-up from sleep. In this case, when the processor is asleep, Timer1 would continue to increment until it overflows, and if the interrupt is enabled, will wake the processor on overflow and branch to the interrupt vector.

## Watch Dog Timer

The Watchdog timer is fully simulated in the MPLAB-SIM simulator. Because it is fuse-selectable on the device, it must be enabled with <u>Options ></u> <u>Processor Setup > Hardware</u>. The period of the WDT is determined by the prescaler settings in the OPTION register. The basic period (with prescaler = 1) is approximated at 18 ms (to closest instruction cycle multiple).

# **Special Registers**

To aid in debugging this device, certain items that are normally not observable have been declared as "special" registers. Prescalers and postscalers cannot be declared in your code as "registers", so there are special labels that show up in the Special Function Registers window.

The following are special symbols that are available for the processors in the PIC16CXX family. (Consult the data sheet for the particular device you are using for information on which symbols are implemented.)

T0PRE - Prescaler for timer0 T1PRE - Prescaler for timer1 T2PRE - Prescaler for timer2 T2POS - Postscaler for timer2 CCP1PRE - Prescaler for CCP1 SPIPRE - Prescaler for SPI SSPSR - SSP Shift register

## **Peripherals**

### **Peripherals supported**

Along with providing core support, the following peripheral modules (in addition to general-purpose I/O) are supported. (Consult the data sheet for the particular device you are using for information on which symbols are implemented.)

- Timer0
- Timer1
- Timer2
- CCP1
- CCP2
- Parallel Slave Port
- SSP (in SPI Mode only)
- Comparators
- A/D (Limited)
- USART (Limited)

#### TIMER0

Timer0 (and the interrupt it can generate on overflow) is fully supported by the MPLAB-SIM simulator, and will increment by the internal or external clock. Clock input must have a minimum high time of 1 Tcy and a minimum low time of 1 Tcy due to stimulus requirements. The prescaler for Timer0 is made accessible as T0PRE.

## TIMER1

Timer1 in its various modes is supported by the MPLAB-SIM simulator, except when running in counter mode by an external crystal. The MPLAB-SIM simulator supports timer1 interrupts generated on overflow, and interrupts generated by wake-up from sleep. The prescaler for Timer1 is viewable as T1PRE in the Special Function Registers window. The external oscillator on RC0/RC1 is not simulated, but a clock stimulus can be assigned to those pins.

## TIMER2

Timer2 and the interrupt that can be generated on overflow are fully supported by the MPLAB-SIM simulator, and both the prescaler and postscaler for Timer2 are viewable as T2PRE and T2POS.

### CCP1 and CCP2

#### CAPTURE

The MPLAB-SIM simulator fully supports capture and the interrupt generated. The prescaler for the CCP module is viewable CCP1PRE.

#### COMPARE

Compare mode, its interrupt, and the special event trigger (resetting Timer1 by CCP1) are supported in this version of the MPLAB-SIM simulator.

#### PWM

PWM output (resolution greater than 1 Tcy only) are not supported in this version of the MPLAB-SIM simulator.

#### SSP

The Synchronous Serial Port is supported in SPI mode only. the shift register (SSPSR) can be added to the view screen, observed and modified. The MPLAB-SIM simulator currently does not support the  $l^2C^{TM}$  mode.

### USART

Timing and interrupt generation is supported. Baud rate generator is supported. Reading writing of the registers are supported but actual receive or transmit operation is not simulated.

#### A/D Converter

All the registers, timing function and interrupt generation are implemented. The simulator, however, does not load any meaningful value into A/D result register (ADRES) at the end of a conversion.

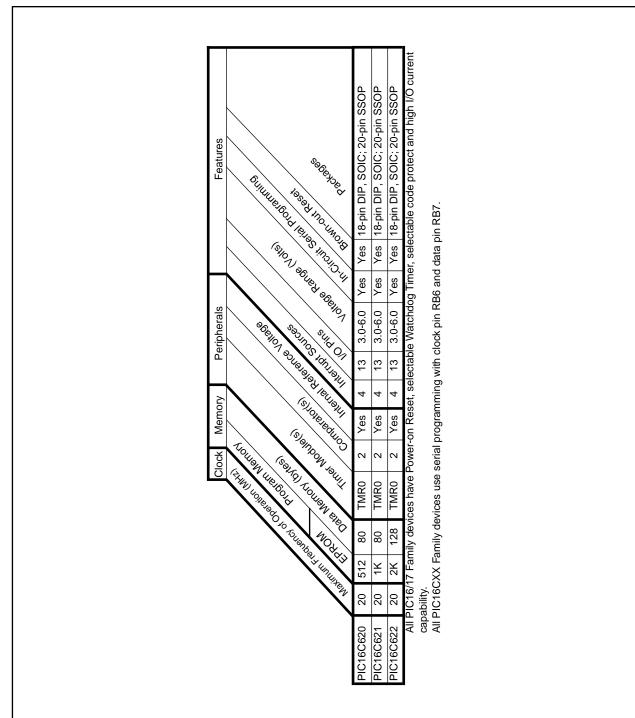
## **EEPROM Data Memory**

The EEPROM data memory (for PIC16C8X devices) is fully simulated. The registers and the read/write cycles are fully implemented. the write cycle time is approximated to 10 ms (to nearest instruction cycle multiple).

The simulator simulates the functions of WRERR and WREN control bits in the EECON1 register.

Features	Culture Secretore Control Cont	- 28-pin SDIP, SOIC, SSOP	Yes 28-pin SDIP, SOIC, SSOP	Yes 28-pin SDIP, SOIC, SSOP	Yes 28-pin SDIP, SOIC	Yes 28-pin SDIP, SOIC	<ul> <li>40-pin DIP;</li> <li>44-pin PLCC, MQFP</li> </ul>	Yes 40-pin DIP; 44-pin PLCC, MQFP, TQFP	Yes 40-pin DIP; 44-pin PLCC, MQFP	<ul> <li>40-pin DIP;</li> <li>44-pin PLCC, MQFP</li> </ul>	Yes 40-pin DIP; 44-pin PLCC, MQFP, TQFP	Yes 40-pin DIP; 44-pin PLCC, MQFP, TQFP	All PIC16/17 family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect, and high I/O current capability. All PIC16C6X family devices use serial programming with clock pin RB6 and data pin RB7. Please contact your local sales office for availability of these devices.
	OT OTO II	, Yes	Yes Ye	Yes Ye	Yes Ye	Yes Ye	Yes -	Yes Ye	Yes Ye	Yes	Yes Ye	Yes Ye	rotect, a
rals	500 11 0 10 10 10 10 10 10 10 10 10 10 10	3.0-6.0	3.0-6.0 Y	3.0-6.0 Y	3.0-6.0 Y	3.0-6.0 Y	3.0-6.0 Y	3.0-6.0 Y	3.0-6.0 Y	3.0-6.0 Y	3.0-6.0 Y	3.0-6.0 Y	table code p in RB7.
Peripherals	LI A DI LI A D	52	22	22	22	22	33	33	33	33	33	33	ar, selec data pi
	NO CHILLS SI IN	2	7	7	10	10	ω	ω	8	11	11	11	og Time 36 and
	Alered of the	1		I	I	I	Yes	Yes	Yes	Yes	Yes	Yes	<i>l</i> atchdc < pin RE vices.
lory	Coloris Contract (10, 138, 140, 150, 150, 150, 150, 150, 150, 150, 15	SPI/I <sup>2</sup> C	SPI/I <sup>2</sup> C	SPI/I <sup>2</sup> C	SPI/I <sup>2</sup> C, USART	SPI/I <sup>2</sup> C, USART	SPI/I <sup>2</sup> C	SPI/I <sup>2</sup> C	SPI/I <sup>2</sup> C	SPI/I <sup>2</sup> C, USART	SPI/I <sup>2</sup> C, USART	SPI/I <sup>2</sup> C, USART	ectable W with clock f these de
Memory	into and interest of the second secon	-	~	~	2	2	~	~		2	, 2	2	et, se iming ility of
Clock	Local and the second se	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	All PIC16/17 family devices have Power-on Reset, selectable Watchdog Timer, selectable c All PIC16C6X family devices use serial programming with clock pin RB6 and data pin RB7. Please contact your local sales office for availability of these devices.
	10318 - 80 - MO	128	128	128	192	192	128	128	128	192	192	192	s have es use ales of
	Totol WORK		1	2K	I	<del>4</del>	I	I	2K	I		4K	device: / device · local s
		X	ZK		<del>4</del>		2K	2K	Ι	4 7	4K		family X family tct your
		20	20	20	20	20	20	20	20	20	20	20	C16/17 C16C6) e conta
		PIC16C62	PIC16C62A <sup>(1)</sup>	PIC16CR62 <sup>(1)</sup>	PIC16C63 <sup>(1)</sup>	PIC16CR63 <sup>(1)</sup>	PIC16C64	PIC16C64A <sup>(1)</sup>	PIC16CR64 <sup>(1)</sup>	PIC16C65	PIC16C65A <sup>(1)</sup>	PIC16CR65 <sup>(1)</sup>	All PIC All PIC Note 1: Pleas

Table F.1 PIC16C6X Family of Microcontrollers



#### Table F.2 PIC16C62X Family of Microcontrollers

**Appendix F. PIC16CXX Simulator Issues** 

						,uometu			Derio	Darioharale				Eastings
				APACING CANADA	≥			, allo			Sieles	$\mathbf{N}$		reautes
			To Lang	Sandon Contant Contant	1 Pho	Coules Structure State	THE SHO		100 - 100 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 - 140 -	449.05 H	MILON DAY CONNINCTION OF A CONNINC A	Solution of the second	SHOT ST	1001 of 170
		Inuit	NO4 A JI	N IOUIL	Ċ)	ALLAS TRA	Je d	alle	3	NO I	Sentor VIII	N. U.	Jid No.	and the second
PIC16C70 <sup>(1)</sup>	20	512	36	TMR0			Ι	4	4	13	3.0-6.0	Yes	Yes	18-pin DIP, SOIC; 20-pin SSOP
PIC16C71	20	¥	36	TMR0		1	1	4	4	13	3.0-6.0	Yes	1	18-pin DIP, SOIC
PIC16C71A <sup>(1)</sup>	20	<del>ź</del>	68	TMR0		1	1	4	4	13	3.0-6.0	Yes	Yes	18-pin DIP, SOIC; 20-pin SSOP
PIC16C72 <sup>(1)</sup>	20	2K	128	TMR0, TMR1, TMR2	- -	SPI/I <sup>2</sup> C		S	8	22	3.0-6.0	Yes	Yes	28-pin SDIP, SOIC, SSOP
PIC16C73	20	4K	192	TMR0, TMR1, TMR2	2 5	SPI/I <sup>2</sup> C, USART		5	11	22	3.0-6.0	Yes	I	28-pin SDIP, SOIC
PIC16C73A <sup>(1)</sup>	20	<del>}</del>	192	TMR0, TMR1, TMR2	5	SPI/I <sup>2</sup> C, USART		ъ	11	22	3.0-6.0	Yes	Yes	28-pin SDIP, SOIC
PIC16C74	20	4K	192	TMR0, TMR1, TMR2	2 5	SPI/I <sup>2</sup> C, USART	Yes	8	12	33	3.0-6.0	Yes	I	40-pin DIP; 44-pin PLCC, MQFP
PIC16C74A <sup>(1)</sup>	20	4 K	192	TMR0, TMR1, TMR2	2 5	SPI/I <sup>2</sup> C, USART	Yes	8	12	33	3.0-6.0	Yes	Yes	40-pin DIP; 44-pin PLCC, MQFP, TQFP
	C16/17 C16C7	7 Fami 'X Farr	ly devic vily dev	All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer, selectable or All PIC16C7X Family devices use serial programming with clock pin RB6 and data pin RB7.	on F rogr	teset, sele ramming v	ectable vith clo	e Watch ock pin	dog Tii RB6 a⊧	mer, si nd dat	electable a pin RB7	code p.	protect	All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability. All PIC16C7X Family devices use serial programming with clock pin RB6 and data pin RB7.
Note 1: Pleas	se cont	act yoı	ur local	sales office for a	avai	lability of	these (	devices						

Table F.3 PIC16C7X Family of Devices

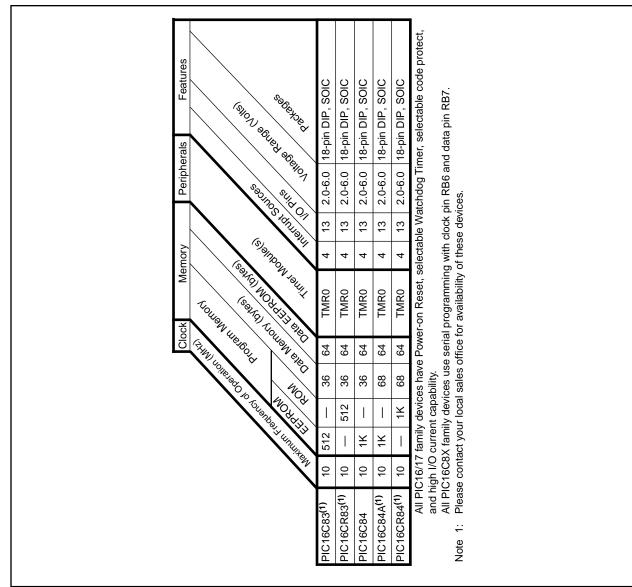


Table F.4 PIC16C8X Family of Devices

**Appendix F. PIC16CXX Simulator Issues** 

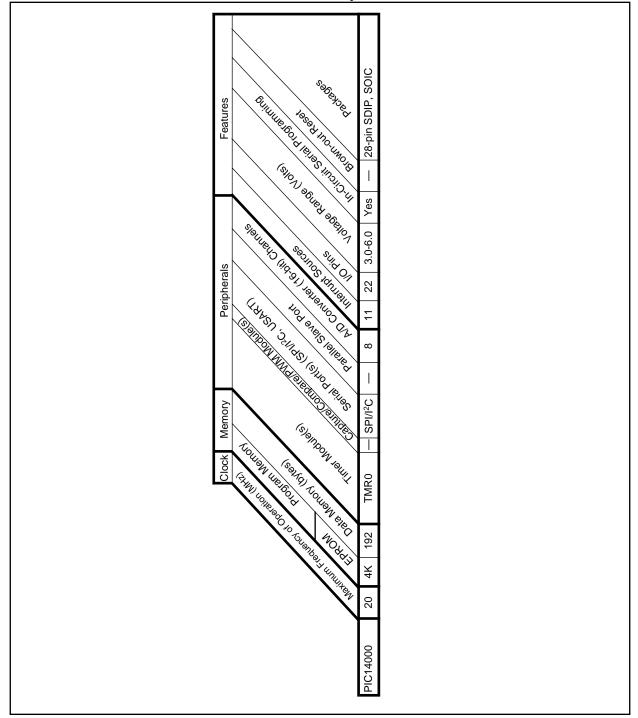


 Table F.5
 PIC14XXX Family of Devices



### Appendix G. MPLAB-SIM PIC17CXX Simulator Issues

### Introduction

This appendix discusses I/O pins, interrupts, registers, peripherals, modes, and conditions for using the PIC17CXX family of microcontrollers.

## **PIC17CXX Selection Table**

At the end of this Appendix there is a selection table which contains the current list of microcontrollers available in the PIC17CXX family.

### I/O Pins

The PIC17CXX family of devices has I/O pins multiplexed with other peripherals (and therefore referred by more than one name). When modifying pins either manually or via the stimulus file, use the following pin names only. These are the only ones that the MPLAB-SIM simulator recognizes as valid I/O pins:

- MCLR
- RA0-RA5
- RB0-RB7
- RC0-RC7
- RD0-RD7
- RE0-RE2

These pin names can be used in the Modify window (*Window > Modify*) and in stimulus files.

### Interrupts

MPLAB-SIM supports all interrupts on the PIC17CXX:

- External interrupt on INT pin
- TMR0 overflow interrupt
- External interrupt on RA0 pin
- Port B input change interrupt
- Timer/Counter1 interrupt
- Timer/Counter2 interrupt
- Timer/Counter3 interrupt
- Capture1 interrupt

- Capture2 Interrupt
- Serial port transmit interrupt\*
- Serial port receive interrupt\*

\*Serial port timing only

# **CPU Model**

### **Reset Conditions**

All reset conditions are supported by the MPLAB-SIM simulator.

An <u>MCLR reset during normal operation or during SLEEP</u> can easily be simulated by driving the <u>MCLR</u> pin low (and then high) via the stimulus file, by clicking on the Reset button on the toolbar or selecting <u>Debug>Run>Reset</u>.

A <u>WDT time-out reset</u> is simulated when the WDT is enabled (see <u>*Options>ProcessorSetup*</u> dialog), the proper prescaler is set, and the WDT actually overflows. WDT time-out period is approximated at 12 ms (to closest instruction cycle multiple) but can be changed by using the dialog.

The Time out  $(\overline{TO})$  and Power-Down  $(\overline{PD})$  bits in the ALUSTA register reflect appropriate reset condition. This feature is useful for simulating various power-up and time-out forks in the user code.

### Sleep

The MPLAB-SIM simulator simulates the SLEEP instruction and will appear "asleep" until a wake-up from sleep condition occurs. For example, if the Watchdog timer has been enabled, it will wake the processor up from sleep when it times out. Another example of a wake-up-from-sleep condition, would be an input change on Port B. If the interrupt is enabled and the GLINTD bit is set, the processor will wake-up and will resume executing from the instruction following the SLEEP command. If the GLINTD = 0, the normal interrupt response will take place.

### Watch Dog Timer

The Watchdog Timer is fully simulated in the MPLAB-SIM simulator. Because it is fuse-selectable and fuse-configurable on the device, it must be enabled and configured by the <u>Options>Processor Setup</u> dialog in the MPLAB-SIM simulator. The basic period of the WDT (with prescaler = 1) is approximated at 12ms (to closest instruction cycle multiple).

# **Special Registers**

To aid in debugging this device, certain items that are normally not observable have been declared as "special" registers. Prescalers cannot be declared in user code as "registers", so the following special symbols are available in the Special Function Registers window:

**T0PRE (Prescaler for Timer 0)** 

WDTPRE (Prescaler for WDT)

### Peripherals

### **Peripherals supported**

Along with providing core support, the following peripheral modules (in addition to general-purpose I/O) are supported:

- Timer 0 in both internal and external clock modes
- Timer1 and Timer2 (and their respective period registers)
- Timer3
- Two Capture Modules
- Two PWM Modules
- USART (limited)

### TIMER0

Timer0 (and the interrupt it can generate on overflow) is fully supported by the MPLAB-SIM simulator, and will increment by the internal or external clock. Delay from external clock edge to timer increment has also been simulated, as well as the interrupt latency period. Clock input must have a minimum high time of 1Tcy and a minimum low time of 1Tcy due to the stimulus file requirements. The prescaler for Timer0 is made accessible as T0PRE. It can be watched and modified.

### TIMER1 and TIMER2

Timer1 and Timer2 in its various modes is fully supported by the MPLAB-SIM simulator. Delays from clock edge to increment (when configured to increment from rising or falling edge of external clock) is simulated as well as the interrupt latency periods. Clock input must have a minimum high time of 1Tcy and a minimum low time of 1Tcy due to the stimulus file requirements.

### **TIMER3 and Capture**

The MPLAB-SIM simulator fully supports Timer3 and the Capture module in all of its modes. Delays from clock edge to increment (when configured in external mode), delay for capture and interrupt latency periods are fully supported. Clock input must have a minimum high time of 1Tcy and a minimum low time of 1Tcy due to the stimulus file requirements.

### PWM

Both PWM outputs are supported (resolution greater than 1Tcy only) are supported in this version of the MPLAB-SIM simulator.

### USART

Timing and interrupt generation is supported. Baud rate generator is supported. Reading and writing of the registers are supported but actual receive or transmit operation is not simulated.

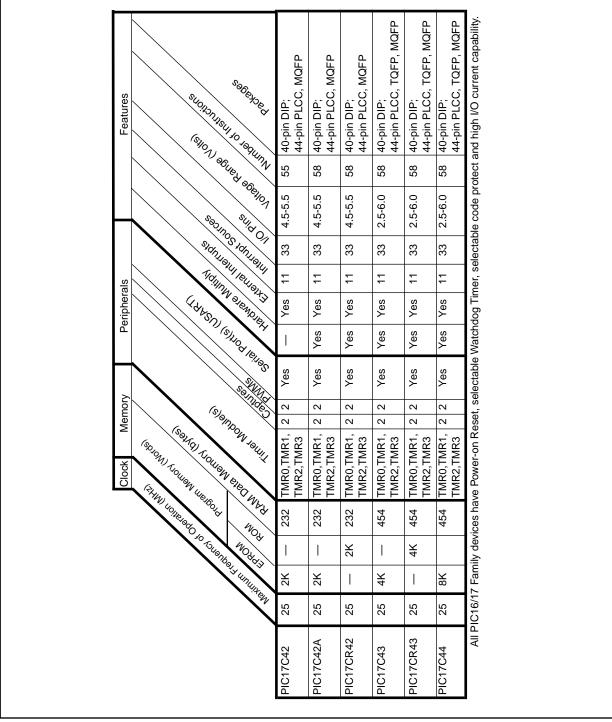
### **Memory Modes**

The following memory modes are supported by the MPLAB-SIM simulator:

- Microcontroller Mode
- Extended Microcontroller Mode
- Microprocessor Mode

The default is Microcontroller mode. If you would like to use any of the other modes, you must use the <u>Options > Processor Setup > Hardware</u> dialog.

# **Appendix G. MPLAB-SIM PIC17CXX Simulator Issues**



#### Table G.1 PIC17CXX Family of Microcontrollers



# Appendix H. Upgrading from MPSIM to MPLAB-SIM

### Introduction

If you have used the DOS based version of MPSIM, the MPLAB-SIM simulator may seem completely different. The core simulator engine is unchanged, but some of the command features of the DOS based program show up in MPLAB-SIM as portions of the MPLAB debugging environment. This list covers the DOS commands and explains how the equivalent functions are accessed in the MPLAB-SIM simulator.

# Highlights

Upgrading from MPSIM to the MPLAB-SIM simulator covers the following topics:

- System Navigation
- Program Memory
- Breakpoints
- Tracing
- Registers and Data Memory
- Stimulus and Timer
- Program Memory Patch and Control
- Symbol Table
- View Screen
- System Setup and Control

### **System Navigation**

DOS CMD	Equivalent MPLAB-SIM Function	MPLAB Action
AB	Exit MPLAB.	File > Exit
Н	Use the on-line Help system.	Help
Q	Exit MPLAB.	<u>File &gt; Exit</u>

# **Program Memory**

DOS CMD	Equivalent MPLAB-SIM Function	MPLAB Action
DE	Fill memory with 0xFFFF (all 1's).	<u>Window &gt; Modify</u> , or <u>Debug &gt; Clear</u> <u>Program Memory</u>
DI	Show Program Memory.	Window > Program Memory
DM	Use Program Memory system button to change display format.	<u>Window &gt; Program Memory &gt; System</u> <u>Button</u>
FM	Fill range of memory.	<u>Window &gt; Modify</u>
IA	Display memory.	Window > Program Memory
IN	Modify memory.	Window > Modify
LO	Load Hex object file.	File > Import > Download to Memory
М	Modify Program Memory.	<u>Window &gt; Modify</u>
0	Save Hex File.	File > Export > Save Hex File
SF	Search memory.	Make Program Memory window active. Then select <u>Editor &gt; Find</u> .
SI	Search memory.	Make Program Memory window active. Then select <u>Editor &gt; Find</u> .
SM	Search memory.	Make Program Memory window active. Then select <u>Editor &gt; Find</u> .
ZM	Fill memory with 0's.	Select <u>Window &gt; Modify</u> to display memory window. Fill range with 0.

# **Breakpoints**

DOS CMD	Equivalent MPLAB-SIM Function	MPLAB Action
В	Set breakpoint.	View break settings in <u>Window &gt; Program</u> <u>Memory</u> , or in <u>Debug &gt; Break Settings</u> menu.
BC	Clear breakpoint.	Use <u>Debug &gt; Break Settings</u> dialog or drag range with left mouse button. Select break with right mouse button.
DB	Display breakpoints.	View breakpoints in Program Memory window ( <i>Window &gt; Program Memory</i> ) or Break Settings dialog ( <i>Debug &gt; Break</i> <u>Settings</u> ).
С	Run	Click on Run icon.
E	Run from new PC.	Set new program counter on Debug menu with <u>Debug &gt; Run &gt; Change Program</u> <u>Counter</u> . Click on Run icon.
GO	Reset, Run	Click on Reset icon, then click Run icon.
RS	Reset Target.	Click on Reset icon.
SS	Single Step.	Click on single step icon.

# Tracing

DOS CMD	Equivalent MPLAB-SIM Function	MPLAB Action
DX	Display Trace Settings.	View trace settings in <u>Window &gt; Program</u> <u>Memory</u> , or in <u>Debug &gt; Trace Settings</u> menu.
ТА	Set Trace Points.	Use <u>Debug &gt; Trace Settings</u> dialog or drag range with left mouse button. Select trace with right mouse button.
ТС	Trace Instructions.	Use <u>Debug &gt; Trace Settings</u> dialog or drag range with left mouse button. Select trace with right mouse button.
TF	Save Trace Buffer.	File > Export > Save Trace Buffer.
TR	Trace data with Conditional Break.	Debug > Execute Conditional Break.

# **Registers and Data Memory**

DOS CMD	Equivalent MPLAB-SIM Function	MPLAB Action
DR	Display register values.	Open <u>Window &gt; Special Function</u> <u>Registers</u> window, add registers to Watch window ( <u>Window &gt; New Watch Window</u> ), or open <u>Window &gt; File Registers</u> .
EE	Modify EEPROM data memory.	<u>Window &gt; Modify</u>
F	Change File Register value.	Double click on register value in Special Function Register window or Watch window, or select register range in File Register window and use right mouse button.
IR	Initialize POR with random values.	Use <u>Debug &gt; Power-On-Reset</u> selection.
LR	Load registers from file.	Use <u>Debug &gt; Power-On-Reset</u> selection.
SE	Modify I/O pin.	Use <u>Window &gt; Modify</u> to write pin. Use <u>Window &gt; Special Function Registers</u> window to view.
UR	Save registers to file.	Use <u>Debug &gt; Power-On-Rese</u> t selection.
W	Modify W register.	Double click on W register in <u>Window &gt;</u> <u>Special Function Registers</u> window.

# **Stimulus and Timer**

DOS CMD	Equivalent MPLAB-SIM Function	MPLAB Action
СК	Assign clock synchronous stimulus.	Select <u>Debug &gt; Simulator Stimulus &gt;</u> <u>Clock Stimulus</u> dialog.
DK	Assign key asynchronous stimulus.	Select <u>Debug &gt; Simulator Stimulus &gt;</u> <u>Asynchronous Stimulus</u> dialog.
DW	Enable/Disable Watch Dog Timer.	Select <u>Options &gt; Processor Setup &gt;</u> <u>Hardware</u> dialog.
FI	Setup File Register Stimulus.	Select <i>Debug &gt; Simulator Stimulus &gt;</i> <i>Register Stimulus</i> dialog.
IP	Assign Pin File Stimulus.	Select <u>Debug &gt; Simulator Stimulus &gt; Pin</u> <u>Stimulus</u> dialog.
RE	Reset Stopwatch.	Select Window > Stopwatch. Click on Zero.
SC	Show Stopwatch.	Select <u>Window &gt; Stopwatch</u> .
ST	Assign Pin file Stimulus.	Select <u>Debug &gt; Simulator Stimulus &gt; Pin</u> <u>Stimulus</u> .
WP	Set WDT Period.	Select <u>Options &gt; Processor Setup &gt;</u> <u>Hardware</u> .
ZT	Reset Stopwatch.	Select <u><i>Window &gt; Stopwatch</i></u> window. Click on Zero.

# **Program Memory Patch and Control**

DOS CMD	Equivalent MPLAB-SIM Function	MPLAB Action
DP	No equivalent MPLAB-SIM function.	
RA	No equivalent MPLAB-SIM function.	
RP	No equivalent MPLAB-SIM function.	
ZP	No equivalent MPLAB-SIM function.	

Part 3 – Appendices

# Symbol Table

DOS CMD	Equivalent MPLAB-SIM Function	MPLAB Action
DL	No equivalent MPLAB-SIM function.	
DS	Show symbols.	Select <u>Window &gt; Show Symbol List</u> .
GS	No equivalent MPLAB-SIM function.	
LS	No equivalent MPLAB-SIM function. Project automatically loads symbol tables.	

# **View Screen**

DOS CMD	Equivalent MPLAB-SIM Function	MPLAB Action
AD	Add item to Watch window.	Open Watch Window ( <u>Window &gt; Load</u> <u>Watch Window</u> ) and press <b><ins></ins></b> or select Edit Watch from the System menu.
DV	Delete item from Watch window.	Select <u>Window &gt; Load Watch Window</u> . Press system menu (upper left), and click <b>Edit Watch</b> , or select variable and press <b><del></del></b> .
NV	Close Watch window.	Press close button on Watch Window.
V	Open Watch window.	Select <u>Window &gt; Load Watch Window</u> or <u>Window &gt; New Watch Window</u> .

# **System Setup and Control**

DOS CMD	Equivalent MPLAB-SIM Function	MPLAB Action
EL	No equivalent MPLAB-SIM function.	
CW or FW	Set configuration bits.	Select <u>Options &gt; Processor Setup &gt;</u> <u>Hardware</u> dialog.
GE	No equivalent MPLAB-SIM function.	
LJ	No equivalent MPLAB- SIM function.	
Р	Select target processor.	Select <u>Options &gt; Development Mode</u> dialog.
SR	Set display or entry radix.	Select Radix on <u><i>Window &gt; Modify</i></u> window.



### Glossary

### Introduction

To provide a common frame of reference, this Glossary defines the terms that follow.

## **Highlights**

This glossary contains definitions for the following systems:

- MPLAB Terms
- PICMASTER Terms No Longer Used

### **MPLAB** Terms

#### Application

A set of software and hardware developed by the user, usually designed to be a product controlled by a PIC16/17 microcontroller.

#### Assembler Source Code

A text file that is processed by an assembler to produce a one-to-one correspondence between assembler instructions and PIC16/17 machine code.

#### **Asynchronous Stimulus**

Data generated to simulate external inputs to the simulator.

#### Breakpoint

An address where execution of the firmware loaded in program memory will halt.

#### Build

A function that recompiles all the source files for an application.

#### C Code

A program written in the high level language called "C," and which will be converted into PIC16/17 machine code.

#### Caution

An alert that is provided to warn you of a situation that would cause physical damage to a device, software file, or equipment.

#### Compile

To translate a user's "C" source text code into machine code.

#### **Configuration Bits**

Unique bits programmed to set modes of operation. A configuration bit may or may not be preprogrammed.

#### Data RAM

General purpose file registers from RAM on the PIC16/17 device being emulated. The File Register window displays data RAM.

#### DDE

See Dynamic Data Exchange.

#### Download

Download is the process of sending data from the host PC to the emulator or to the target board.

#### **Dynamic Data Exchange**

The message protocol in Microsoft Windows that transfers information in realtime between windows applications. MPLAB acts as a DDE client that can transmit data to requesting applications, but cannot receive data from other applications.

#### EEPROM

Electrically Erasable Programmable Read Only Memory.

#### Emulation

The process of executing software loaded in the PICMASTER program memory on the emulator probe as if the firmware resided on the microcontroller device under development.

#### **Emulation Memory**

Program memory contained within the emulator.

#### Emulator

Hardware that performs emulation. See Emulation.

#### **Emulator System**

The Microchip Emulator System includes the PICMASTER Pod, the PC Interface Card, a Probe Kit, and the MPLAB Software.

#### Export

Send data out of MPLAB in a standardized format.

#### Extended Microcontroller Mode (PIC17CXX Devices Only)

In extended microcontroller mode, on-chip program memory as well as external memory is available. Execution automatically switches to external if the program memory address is greater than the internal memory space of the PIC17CXX device. Inaccessible memory in extended microcontroller mode includes fuses, test memory, and boot memory.

#### **External Break Line**

An External Break Input Signal logic probe line (BRK) for hardware breakpoints. The PICMASTER emulator can be programmed to halt the target processor when the external break line senses an edge (rising or falling) from any compatible external connection.

#### **External RAM**

Off-chip Read/Write memory.

#### **File Registers**

On-chip general purpose and special function registers.

#### Halt

The command that stops the emulator. Executing Halt is the same as stopping at a breakpoint. The program counter stops, and the user can inspect and change register values, and single step through code.

#### Hex Code

A file of executable instructions assembled or compiled from source code into standard hex format code. Hex code can be directly converted to object code.

#### **High Level Language**

A language for writing programs that is of a higher level of abstraction from the processor than assembler code. High level languages employ a compiler to translate statements into machine instructions that the target processor can execute.

#### IDE

Integrated Development Environment. An application that has multiple functions for software development. The MPLAB IDE integrates a compiler, an assembler, a project manager, an editor, a debugger, simulator, and an assortment of other tools within one Windows application. A user developing an application should be able to write code, compile, debug and test an application without leaving the MPLAB desktop.

#### Import

Bring data into the MPLAB Integrated Development Environment (IDE) from an outside source

#### **Logic Probes**

Up to fourteen logic probes connect to the Emulator. The logic probes provide external trace inputs, external trace halt signal, trigger output signal, external break input signal, trace buffer clock, +5V, and a common ground.

#### **Make Project**

A command that rebuilds an application, re-compiling only those source files that have changed since the last complete compilation.

#### **Microcontroller Mode**

One of the possible program memory configurations of the PIC17CXX family of microcontrollers. In microcontroller mode, only internal execution is allowed. Thus, only the on-chip program memory is available in microcontroller mode. Accessible memory includes: program memory, configuration bits, test memory, and boot memory (FE00h to FFFFh).

#### **Microprocessor Mode**

One of the possible program memory configurations of the PIC17CXX family of microcontrollers. In microprocessor mode, the on-chip program memory is not used. The entire 64K program memory is mapped externally. Inaccessible memory in microprocessor mode includes configuration bits, test memory, and boot memory.

#### **MPLAB Software**

The name of the main executable program that supports the IDE with an Editor, Project Manager, and Emulator/Simulator Debugger. The MPLAB Software resides on the PC host. The executable file name is MPLAB.EXE. MPLAB.EXE calls many other files.

#### MRU

Most <u>Recently Used</u>. Refers to files and windows available to be selected from MPLAB main pull down menus.

#### **Multiprocessor Emulation**

Emulation of one to four emulators from a single PC host. Multiprocessor emulation requires MPLAB software and one to four PICMASTER emulators.

#### Non Real-Time

Refers to the processor executing single step instructions, executing a specified number of cycles, or executing until a specified condition is met.

#### **Object code**

The machine code that is produced from the source code after it is processed by an assembler or compiler. This code will be-the memory-resident code that will run on the PIC16/17 in the user's application.

#### **Off-Chip Memory**

Off-chip memory refers to the memory selection option for the PIC17CXX device where memory may reside on the target board, or where all program memory may be supplied by the Emulator. <u>Options > Processor Setup ></u> <u>Hardware</u> provides the Off-Chip Memory selection dialog box.

#### **Pass Counter**

A counter that decrements each time an event (such as the execution of an instruction at a particular address) occurs. When the pass count value reaches zero, MPLAB halts the processor. You can assign the Pass Counter to either break logic or to trace logic.

#### PC

Any IBM<sup>®</sup> or compatible Personal Computer. MPLAB needs a 386X or better machine.

#### PC Host

The computer running Windows 3.x.

#### **PC Interface Card**

A Microchip proprietary parallel interface card allowing the MPLAB software residing on the PC to communicate with the PICMASTER emulator.

#### **Performance Analysis**

A histogram of the measured execution of some characteristic of the application software, often measuring the execution time of one or more routines.

#### PIC16/17

PIC16/17 refers to the PIC16C5X, PIC16CXX, and PIC17CXX Microchip microcontroller families.

#### PICMASTER

The hardware unit that provides tools for emulating and debugging firmware applications and is also referred to as the emulator. This unit contains emulation memory, breakpoint logic, counters, timers, and a trace analyzer among some of its tools.

#### Pod

The external emulator box that contains emulation memory, trace memory, event and cycle timers, and trace/breakpoint logic. Occasionally used as an abbreviated name for the PICMASTER Universal In-Circuit Emulator.

#### **Power on Reset Emulation**

A software randomization process that writes random values in data RAM areas to simulate uninitialized values in RAM upon initial power application.

#### Probe

A device specific interface between the simulator and the target application. The probe connects to the emulator via a ribbon cable, and the target application board connects to the probe via a ribbon cable. The probe is sometimes called the header board.

#### **Program Counter**

A register that specifies the current execution address.

#### **Program Memory**

Memory in the emulator or simulator containing the downloaded target application firmware.

#### Project

A set of source files to build the object code for an application.

#### **Prototype System**

A term referring to a user's target application, or target board.

#### **PWM Signals**

Pulse Width Modulation Signals.

#### Qualifier

An address or an address range used by the Pass Counter.

#### Radix

The number base, hex, or decimal, used in selecting an address and for entering data in the *Window > Modify* command.

#### **Real-time**

When released from the halt state, the processor runs in real-time mode and behaves exactly as the normal chip would behave. In real-time mode, the realtime trace buffer is enabled and constantly captures all selected cycles, and all break logic is enabled. In the PICMASTER emulator the processor executes in real-time until a valid breakpoint causes a halt, or until the user halts the emulator.

In the simulator real-time simply means execution of the microcontroller instructions as fast as they can be simulated by the host CPU.

#### Run

The command that releases the emulator from halt, allowing it to run the application code and change or respond to I/O in real time.

#### SFR

<u>Special Function Registers.</u>

#### Simulator

A software program that models the operation of the PIC16/17 microprocessor.

#### **Simulator Stimulus**

Data generated to exercise the response of simulation to external signals. Often the data is put into the form of a list of actions in a text file.

#### Single Step

This command steps though code, one instruction at a time. After each instruction, MPLAB updates register windows, watch variables, and status displays so you can analyze and debug instruction execution.

You can also single step C compiler source code, but instead of executing single instructions, MPLAB will execute all assembly level instructions generated by the line of the high level C statement.

#### Source

Source code, usually a text file of assembly instructions or C code.

#### **Special Function Registers**

Registers that control I/O processor functions, I/O status, timers, or other modes or peripherals.

#### Stack

"Push-Down" list of calling routines. Each time a PIC16/17 microcontroller executes a call, the software pushes the return address to the stack. A RET or RETLW command pops the address from the stack.

#### Static RAM, or SRAM

Static Random Access Memory. Program memory you can Read/Write on the target board that does not need refreshing frequently.

#### Step-Into

This command is the same as Single Step. Step-Into (as opposed to Step-Over) follows a CALL instruction into a subroutine.

#### Step-Over

Step-Over allows you to debug code without stepping into subroutines. When stepping over a CALL instruction, the next breakpoint will be set at the instruction after the CALL. If for some reason, the subroutine gets into an endless loop or does not return properly, the next breakpoint will never be reached.

The Step-Over command is similar to Single Step except for its handling of CALL instructions.

#### Stopwatch

A 48-bit counter for measuring execution cycles.

#### Symbol

An label usually produced by an assembler or compiler that refers to machine locations by function names, variable locations, constant declarations, source line-number, or other reference to user source code.

#### **System Button**

The System Button is located in the upper left corner of Windows and some dialogs. This button usually has "Minimize", "Maximize", and "Close." In some MPLAB windows, additional modes or functions can be found under the System Button.

#### Target

Refers to user hardware.

#### **Target Application**

Firmware residing on the target board.

#### **Target Board**

The circuitry and programmable device that makes up the target application.

#### **Target Processor**

The microcontroller device on the target application board that is being emulated.

#### Template

Lines of text that you build for inserting into your files at a later time. The MPLAB Editor stores templates in template files.

#### **Tool Bar**

A row or column of icons that you can click on to execute MPLAB functions.

#### Trace

An emulator or simulator function that logs program execution. The emulator logs program execution in

#### **Trace Memory**

Trace memory contained within the Emulator. Trace Memory is sometimes called the Trace Buffer.

#### **Trigger Output**

Trigger output refers to a PICMASTER emulator output signal that can be generated at any address or address range, and is independent of the trace, arm, and breakpoint settings. Any number of trigger output points can be set. The trigger output signal is available on logic probe pin 13 (white probe tip).

#### Upload

The Upload function transfers data from the emulator to the host PC or from the target board to the emulator.

#### Watch Dog Timer

A timer on a PIC16/17 microcontroller that resets the processor after a selectable length of time.

#### Watch Variable

A variable that you may monitor during a debugging session. Watch windows contain a list of watch variables that are updated at each breakpoint.

# **PICMASTER Terms No Longer Used**

Arm

Replaced by Pass Counter

#### **Arm Points**

Now referred to as qualifier addresses for the Pass Counter.

#### **Conditional Trace**

Replaced by Conditional Break

#### Qualifiers

Now referred to as breakpoints, trace points, and trigger points.

#### Software Trace

Replaced by Conditional Break

#### Timer

Replaced by Stopwatch



# Index

#### Α

About MPLAB	5
Absolute Listing 45, 6	69, 88, 91
Address Labels	83, 90
Arrange Icons	103
ASCII Display	91, 94
Assembling Source Code	17

#### В

BBS10
application notes159
bug reports159
Connecting to158
errata sheets159
Software Releases159
source code159
Special Interest Groups159
Systems Information and Up-
grade Hot Line160
Using the158
Break on Stack Overflow41
Break on Trace Buffer Full40
Break on Watch Dog Timer41
Break Settings
Break on Address Match40
Break on Trace Buffer Full40
Breakpoint Range Definition 71
Pass Counter41
Saving Breakpoint Settings .70
Break, Trace, and Trigger Points 38
Breakpoint
Conditional Break
Real-Time40
Build All19, 33
_

### С

C Compiler
Make Setup31
Cascade103
Center Debug Location69
Change Program Counter62
Clear All Points75
Clear Breakpoints on Download .83
Clear Memory on Download83
Clear Program Memory75
Clock frequency97
Close Project29
Colors
Changing23

Compile
Compile Single File
Conditional Break 44, 63
Customer Support10
Customizing
Desktop21
Development Environment 22
Cycle count97

#### D

D
DDE145
client145
Conventions 146
item name145
MPLAB can transmit data to
other programs 145
Overview
permanent link145
Plotting a histogram of
an address field 145
Topic Name146
Transmit Data from
Data Fields
File Registers
Program Memory 145
Trace Buffer 145
Typical Uses for
Warm Link Example
•
Debug61 Break Settings
Center Debug Location 69
Clear All Points
Execute
Conditional Break
Execute an Opcode 63
Powe-On-Reset
Run
Change Program
counter
Halt
Halt Trace62
Reset61
Run61
Step62
Step Over62
System Reset76
Default Configuration 70

Default Configuration ......79 Desktop Customizing ......21 Window Setup ......21 **Development Environment** 

	40
Environment Setup	80
Global Switches	83
Screen Font	81
Toolbar	81
Toolbar Setup	82
Execute	63
Conditional Break	63
Execute an Opcode	63
Execution	
Non Real-Time	36
Exit21,	61

### F

File	
Exit61	
Export	
Save Hex File 57	7
Save Trace Buffer 56	5
Print 58	3
File Registers45, 93	3
ASCII Display94	ł
Hex Display94	ł
Symbolic Display 94	ł
System Button Options 94	
File Types	
Location of13	3

### G

Global Break Enable
Download
Global Break Enable
Load User Configuration 83 Status Bar Enable
Track Source Code83

### Η

Halt	61
Halt Trace	42
Hardware	84
PIC16CXX Family	. 84, 85
PIC17CXX Family	86
Help	104
About	105
MPLAB Help	104
Release Notes	104
Hex Display	91

### I

Iconize All103IDE3, 5MPLAB5Import55Download to Memory55Initialization File61Format161MPLAB.INI161Options Section163Initialization File Options162allow-save-always162auto-file-action162Backup-directory162backup-mode162dragdrop-flip162max-undo-actions162max-vertical-tile162mru-files-shown162mru-list-size162open-maximized163select-search-match163start-maximized163start-maximized163track-vertical-thumbtack163
start-maximized163
use-dragdrop163
Installation

### Κ

key mapping fu	nctions	 80
Key Mappings		 80

### L

Line Numbers	
Toggle 89	
Load Setup79	
Load User Configuration	
Load Watch Window 101	

#### Μ

Make Project
Menu Functions 15
Mode, Operating
Editor Only 4
MPLAB SIM 4
PICMASTER Emulator 4
Modify 101
Address 101
Auto Increment 102
Data/Opcode101
End All 102
Fill Range102
Memory Area 101
Radix101
Read 102
MPASM
Assembler
Universal PIC16/17 Microcon-
troller Assembler 4
MPASM Help 105 MPLAB
A Quick Look At 3
A QUICK LOOK AT
Accessing Tools
Changing Colors
Changing Screen Font 22
Customizing 161
Development Tools
Editor 4
Full Installation12
How to use with PICMASTER 6
IDE3
Initialization File
MPLAB.INI
Opening text files from com-
mand line13
Operating Modes 4

Program Group11, 13 Project Manager3 Running13
Setup 11
tools
debugging environment . 5
tutorial15
Windows 3.1 based IDE 3
MPLAB Editor 6
MPLAB Functions5
Create and Edit Source
Files5
Debug Firmware Source
Code 5
Group Source Files into
Projects5
MPLAB/PICMASTER Help 104
MPSIM
Simulator 4
Multiple Cycles 45

### Ν

7
5
6
8

### 0

Open Project	29
Load binary code	29
Open Windows	. 103
Options	78
Colors	83
Development Mode	78
Environment Setup	80
Key Mappings	80
Processor Setup	
Hardware	84
Window Setup	79
Default Configuration	79
Load Setup	
Save Setup	79
Options Section	
Initialization File	. 161
P	

Pass Counter 43	3
Assigned to Break 43	
Assigned to Trace	
Points	
set up 43	
Using to Count Events 44	ł

Peripherals
PIC16C5X167
Supported181
PIC16C5X Family
2-Levels Deep Hardware
Stack
Support Issues165
PIC16CXX Family
8-Levels Deep Hardware
Stack
I/O Pins169
Interrupts169
Peripherals171
Stack Overflow Break
Enable 84, 85, 86
Support Issues169
Watch Dog Timer
Watch Dog Timer Break
Enable84, 85
PIC17CXX Family86
Interrupts179
Watch Dog Timer86
WDT Prescale86
PICMASTER6
Emulator4
how to use
Power-On-Reset
Print
Print Options
Processor Mode
Hardware84
Microcontroller
Microprocessor
Processor Setup83
Hardware
Program Memory
Creating Temporary
Breakpoint
Field Descriptions
System Button Options 89, 99
Disassembly Display89
Hex Code Display
Machine Code Display 89
Toggle Line Numbers 89
Project6
Assigning Files to16
Build All33
Close29
Compile a Single File33
Creating a16
Edit30
Make31
New27

Open	
Save	
successful compile	19
Using	15
Project Manager	6
Projects	
Most Recently Used	34

### R

Recommended Reading	9
Register Variables	83
Release Notes	
Repeat DOS Command to	
Window	88
Reset	61
Conditions17	0, 180
Conditions supported	166
Run	61

### S

Save Hex File
Save Trace Buffer
Save Watch100
Screen Font81
Show Symbol List96
Simulator Stimulus
Single Cycle
Software Releases159
Intermediate Release
Production Release
Source Code
Compiling
Debugging19
Rebuilding in Project
Window
Special Registers167
Stack
PIC16C5X92, 93
Status Bar54
Current Development Mode .55
Current Pass Count55
Current Processor55
File Modified54
Insert/Strikeover55
Line No54
Lines in File54
MPLAB Version54
Write/Read Only54
Status Bar Enable83

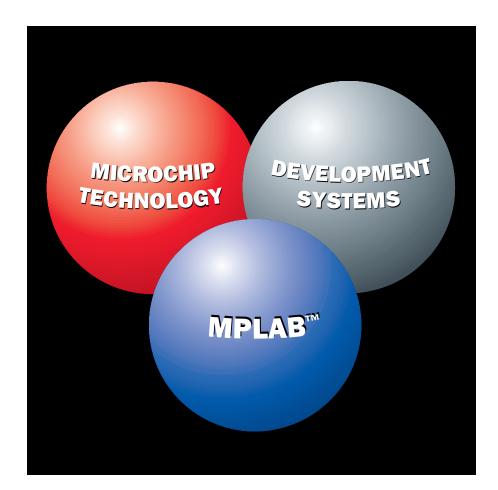
Step	62
Step Over	62
Stopwatch	96
Target Frequency	97
Time Count	97
Support	
Customer	10
Symbol Display Width	83
Address Labels	83
Register Variables	83
Symbolic Display	94
System Button Options	94
System Reset	76

### Т

-
Tile Horizonatal102
Tile Vertical102
Toggle Line Numbers89
Toolbar54
Customizing23
Debug54
Edit
Project54
User Defined54
Tools
DOS Command to
Window87
Repeat DOS Command to
Window88
tools
debugging environment5
project manager6
text editor5
Trace
Circular Trace Buffer41
Halting from the Toolbar42
Trace Data44
Trace Memory42
Trace Points
Real-Time41
Trace Settings73
Track Source Code83
Tutorial
MPLAB Functions15
SAMPLE.ASM15
TUTOR.ASM15
Using Projects15
W
Warranty Registration 10

Warranty Registration	10
Watch	86
Watch Dog Timer	
Watch Dog Timer Break	
Enable	

# **MPLAB** IDE, Simulator, Editor



# **USER'S GUIDE**



# INDEX

Watch Window	
Edit Watch	99
New Watch Window	98
Save Watch1	00
WDT See Watch Dog Timer	
Web Site	
connecting to1	57
file transfer1	57
Window	
Absolute Listing	91
Arrange Icons1	03
Cascade1	
Displaying Multiple	
EEPROM Memory	
File Registers	
Iconize All1	
Load Watch Window1	
Modify1	
MPASM Help1	05
Open Windows1	03
Program Memory	
Recovering Hidden	
Show Symbol List	
Special Function Registers	
Stack	-
Stopwatch	
Tile Horizontal1	
Tile Vertical1	
Trace Memory	
Window Setup	
Default Configuration	
Load Setup	
Save Setup	79

.