ICEPIC™ IN-CIRCUIT EMULATOR USER'S GUIDE

Table of Contents

Table of Contents

NOTES:

General Information

Introduction

This chapter contains general information about this manual and contacting customer support.

Highlights

Topics covered in this chapter:

- About this Guide
- Recommended Reading
- Warranty Registration
- Troubleshooting
- The Microchip Internet Web Site
- Development Systems Customer Notification Service
- Customer Support

About This Guide

Document Layout

This document describes how to use the ICEPIC In-Circuit Emulator development tool to emulate and debug firmware for the PIC16C5X and PIC16CXXX family of microcontrollers. The manual layout is as follows:

- **Chapter 1: Overview and Installation** – Describes what the ICEPIC In-Circuit Emulator is, how it works, how to install the ICEPIC hardware and MPLAB ICEPIC software, and establish communication.
- **Chapter 2: Tutorial – PIC16CXXX** – Shows you how to develop and debug an application using MPLAB IDE Projects and the ICEPIC In-Circuit Emulator.
- **Chapter 3: General Setup** – Describes how to get the hardware and software for the ICEPIC In-CIrcuit Emulator up and running.
- **Chapter 4: Basic Functions** – Describes the basic functions of the ICEPIC In-Circuit Emulator.
- **Chapter 5: Troubleshooting** – Provides information on solving common problems.
- **Appendix A: Specifications** – Describes the serial port specifications for the ICEPIC In-Circuit Emulator, as well as instructions for achieving high baud rates and determining your UART settings.

- **Appendix B: Daughter Board Specifications** – Describes various specifications and jumper link configuration settings for those daughter boards that were available at the time this document was published.
- **Glossary** – A glossary of terms used in this guide.
- **Index** – Cross-reference listing of terms, features and sections of this document.
- **Worldwide Sales and Service** – A listing of Microchip sales offices, service locations and telephone numbers worldwide.

Conventions Used in this Guide

This manual uses the following documentation conventions:

Table 1: Documentation Conventions

<table>
<tr><th>Description</th><th>Represents</th><th>Examples</th></tr>
<tr><td colspan="3">Code (Courier font):</td></tr>
<tr><td>Plain characters</td><td>Sample code
Filenames and paths</td><td>#define START
c:\autoexec.bat</td></tr>
<tr><td>Angle brackets: < ></td><td>Variables</td><td><label>, <exp></td></tr>
<tr><td>Square brackets [ ]</td><td>Optional arguments</td><td>MPASMWIN [main.asm]</td></tr>
<tr><td>Curly brackets and pipe character: { | }</td><td>Choice of mutually exclusive arguments
An OR selection</td><td>errorlevel {0|1}</td></tr>
<tr><td>Lower case characters in quotes</td><td>Type of data</td><td>"filename"</td></tr>
<tr><td>Ellipses...</td><td>Used to imply (but not show) additional text that is not relevant to the example</td><td>list ["list_option..., "list_option"]</td></tr>
<tr><td>0xnnn</td><td>A hexadecimal number where n is a hexadecimal digit</td><td>0xFFFF, 0x007A</td></tr>
<tr><td>Italic characters</td><td>A variable argument; it can be either a type of data (in lower case characters) or a specific example (in uppercase characters).</td><td>char isascii (char, ch);</td></tr>
<tr><td colspan="3">Interface (Helvetica font):</td></tr>
<tr><td>Underlined, italic text with right arrow</td><td>A menu selection from the menu bar</td><td><u>File > Save</u></td></tr>
<tr><td>Bold characters</td><td>A window or dialog button to click</td><td>OK, Cancel</td></tr>
</table>

Table 1: Documentation Conventions (Continued)

Description	Represents	Examples
Characters in angle brackets < >	A key on the keyboard	<Tab>, <Ctrl-C>
Documents (Helvetica font):		
Italic characters	Referenced books	*MPLAB IDE User's Guide*

Documentation Updates

All documentation becomes dated and this user's guide is no exception. Since the MPLAB IDE, ICEPIC In-Circuit Emulator and other Microchip tools are constantly evolving to meet customer needs, some MPLAB IDE dialogs and/ or tool descriptions may differ from those in this document. Please refer to our web site at www.microchip.com to obtain the latest documentation available.

Documentation Numbering Conventions

Documents are numbered with a "DS" number. The number is located on the bottom of each page, in front of the page number. The numbering convention for the DS Number is: DSXXXXXA,

where:

XXXXX	=	The document number.
A	=	The revision level of the document.

Warranty Registration

Please complete the enclosed Warranty Registration Card and mail it promptly. Sending in your Warranty Registration Card entitles you to receive new product updates. Interim software releases are available at the Microchip web site.

Recommended Reading

This user's guide describes how to use the ICEPIC In-Circuit Emulator. The user may also find the data sheets for specific microcontroller devices informative in developing firmware.

README.TXT

The README.TXT file contains updated information that may not be included in this document, as well as a listing of the PICmicro® MCU devices that the version of the MPLAB ICEPIC software you are using supports.

LINK.TXT

This file contains the description of the jumper links for the various daughter boards currently available for the ICEPIC In-Circuit Emulator.

MPLAB IDE User's Guide (DS51025)

Comprehensive guide that describes installation and features of Microchip's MPLAB Integrated Development Environment (IDE), as well as the editor and simulator functions in the MPLAB IDE environment.

MPASM User's Guide with MPLINK & MPLIB (DS33014)

Describes how to use Microchip Universal PICmicro Microcontroller Assembler (MPASM), Linker (MPLINK) and Librarian (MPLIB).

Technical Library CD-ROM (DS00161)

This CD-ROM contains comprehensive data sheets for Microchip PICmicro MCU devices available at the time of print. To obtain this disk, contact the nearest Microchip Sales and Service location (see back page) or download individual data sheet files from the Microchip web site (http://www.microchip.com).

Embedded Control Handbook Vol. 1 & 2 and the Embedded Control Update 2000 (DS00092, DS00167 and DS00711)

These handbooks consist of several documents that contain a wealth of information about microcontroller applications. To obtain these documents, contact the nearest Microchip Sales and Service location (see back page).

The application notes described in these manuals are also obtainable from Microchip Sales and Service locations or from the Microchip web site (http://www.microchip.com).

PICmicro Mid-Range MCU Family Reference Manual (DS33023)

This manual explains the general details and operation of the MCU family architecture and peripheral modules. It is designed to complement the device data sheets.

Microsoft® Windows® Manuals

This manual assumes that users are familiar with the Microsoft® Windows® operating system. Many excellent references exist for this software program and should be consulted for general operation of Windows.

The Microchip Internet Web Site

Microchip provides online support on the Microchip World Wide Web (WWW) site.

The web site 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® Communicator or Microsoft® Internet 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:

http://www.microchip.com

The file transfer site is available by using an FTP program/client to connect to:

ftp://ftp.microchip.com

The web site and file transfer site provide a variety of services. Users may download files for the latest Development Tools, Data Sheets, 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
- Conferences for products, development systems, technical information
- Listing of seminars and events

Development Systems Customer Notification Service

Microchip started the customer notification service to help our customers keep current on Microchip products with the least amount of effort. Once you subscribe to one of our list servers, you will receive e-mail notification whenever we change, update, revise or have errata related to that product family or development tool. See the Microchip web page at http://www.microchip.com for other Microchip list servers.

The Development Systems list names are:

- Compilers
- Emulators
- Programmers
- MPLAB
- Otools (other tools)

Once you have determined the names of the lists that you are interested in, you can subscribe by sending a message to:

`listserv@mail.microchip.com`

with the following as the body:

`subscribe <listname> yourname`

Here is an example:

`subscribe programmers John Doe`

To UNSUBSCRIBE from these lists, send a message to:

`listserv@mail.microchip.com`

with the following as the body:

`unsubscribe <listname> yourname`

Here is an example:

`unsubscribe programmers John Doe`

The following sections provide descriptions of the available Development Systems lists.

Compilers

The latest information on Microchip C compilers, Linkers and Assemblers. These include MPLAB-C17, MPLAB-C18, MPLINK, MPASM as well as the Librarian, MPLIB for MPLINK.

To SUBSCRIBE to this list, send a message to:

`listserv@mail.microchip.com`

with the following as the body:

`subscribe compilers yourname`

Emulators

The latest information on Microchip In-Circuit Emulators. These include MPLAB-ICE and PICMASTER.

To SUBSCRIBE to this list, send a message to:

`listserv@mail.microchip.com`

with the following as the body:

`subscribe emulators yourname`

Programmers

The latest information on Microchip PICmicro device programmers. These include PRO MATE® II and PICSTART® Plus.

To SUBSCRIBE to this list, send a message to:

`listserv@mail.microchip.com`

with the following as the body:

`subscribe programmers yourname`

MPLAB

The latest information on Microchip MPLAB IDE, the Windows Integrated Development Environment for development systems tools. This list is focused on MPLAB IDE, MPLAB-SIM, MPLAB Project Manager and general editing and debugging features. For specific information on MPLAB compilers, linkers and assemblers, subscribe to the COMPILERS list. For specific information on MPLAB emulators, subscribe to the EMULATORS list. For specific information on MPLAB device programmers, please subscribe to the PROGRAMMERS list.

To SUBSCRIBE to this list, send a message to:

`listserv@mail.microchip.com`

with the following as the body:

`subscribe mplab yourname`

Otools

The latest information on other development system tools provided by Microchip. For specific information on MPLAB and its integrated tools refer to the other mail lists.

To SUBSCRIBE to this list, send a message to:

`listserv@mail.microchip.com`

with the following as the body:

`subscribe otools yourname`

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Corporate Applications Engineer (CAE)
- Hotline

Customers should call their distributor, representative, or field application engineer (FAE) for support. Local sales offices are also available to help customers. See the back cover for a listing of sales offices and locations.

Corporate Applications Engineers (CAEs) may be contacted at (480) 786-7627.

In addition, there is a Systems Information and Upgrade Line. This line provides system user’s 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 Hotline Numbers are:

1-800-755-2345 for U.S. and most of Canada.

1-480-786-7302 for the rest of the world.

Chapter 1. Overview and Installation

1.1 Introduction

This chapter provides an overview of the ICEPIC In-Circuit Emulator, as well as the instructions for installing and configuring the ICEPIC In-Circuit Emulator for use with the MPLAB IDE.

1.2 Highlights

Topics covered in this chapter:

- What the ICEPIC In-Circuit Emulator Is
- System Requirements
- ICEPIC Kit Components
- ICEPIC System Components
- Installing the ICEPIC In-Circuit Emulator Hardware
- Setting up the ICEPIC In-Circuit Emulator
- Using the ICEPIC In-Circuit Emulator

1.3 What the ICEPIC In-Circuit Emulator Is

The ICEPIC In-Circuit Emulator is a low-cost in-circuit emulation solution for the PIC16C5X and PIC16CXXX family of PICmicro microcontrollers (MCUs). The modular system design can support different subsets of the PIC16C5X and PIC16CXXX family of PICmicro MCU products through the use of interchangeable daughter boards.

Working with the MPLAB IDE system software, the ICEPIC In-Circuit Emulator offers the following features:

- Real-time, nonintrusive emulation of the PIC16C5X and PIC16CXXX family of PICmicro MCUs
- 8K words of emulation memory
- Full-speed, real-time emulation
- Compatible with all Microsoft® Windows® operating systems
- 8K hardware breakpoints
- Source-level debugging (both Assembly and C)
- Custom watch points
- Symbolic debug capability
- RS-232 interface

With the ICEPIC In-Circuit Emulator, you can:

- Debug your source code on your own hardware
- Debug your firmware in real time
- Debug with hardware breakpoints
- Watch/Modify internal registers and processor status

1.4 System Requirements

To take advantage of the ICEPIC In-Circuit Emulator’s features, you must install the MPLAB IDE software (`mplab.exe`) on a host computer. See the *MPLAB IDE User’s Guide* (DS51025) for system requirements for the MPLAB IDE. ICEPIC will require:

- One free serial port

1.5 ICEPIC Kit Components

There are two parts to the ICEPIC Kit: the ICEPIC Base Unit and the device-specific daughter board. Each kit is sold separately.

1.5.1 ICEPIC Base Unit

The Base Unit contains the:

1. ICEPIC pod, which contains the motherboard
2. RS-232 cable
3. 9V power supply
4. MPLAB ICEPIC software
5. *ICEPIC In-Circuit Emulator User's Guide* (DS51103)
6. Warranty/Registration card (not shown)

Figure 1.1: ICEPIC In-Circuit Emulator System

1.5.2 ICEPIC Daughter Boards

Each device-specific daughter board contains:

1. Daughter board
2. Extender cable
3. Warranty/Registration card (not shown)

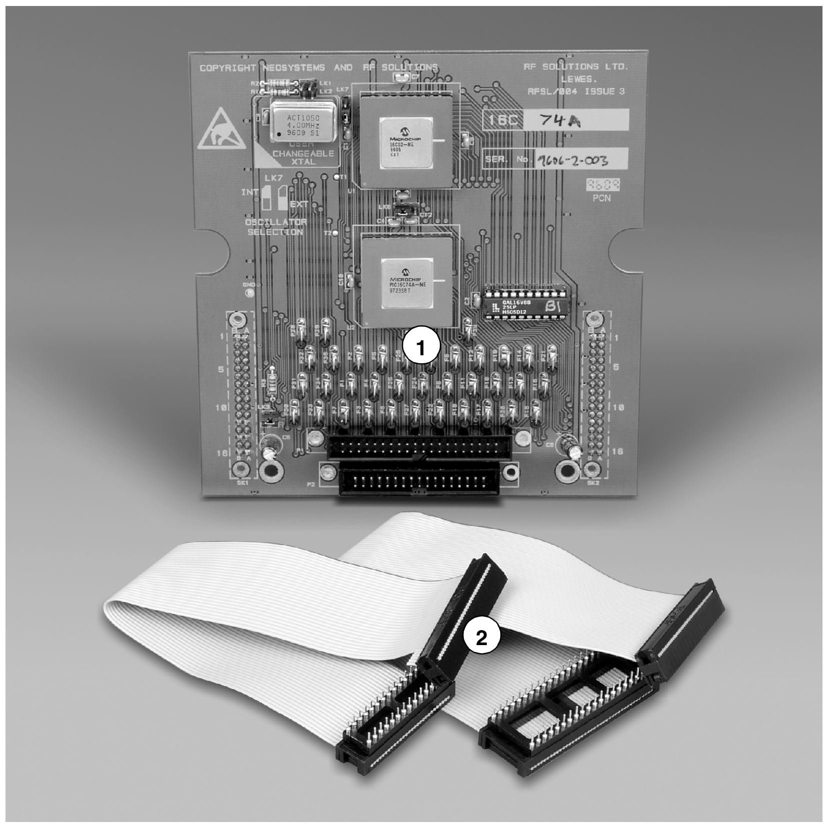


Figure 1.2: Sample Daughter Board

1.6 ICEPIC System Components

The ICEPIC In-Circuit Emulator system consists of two main parts:

1. Motherboard
2. Daughter Board (see Appendix B and the `readme.txt` file for a listing of available daughter boards.)

1.6.1 Motherboard

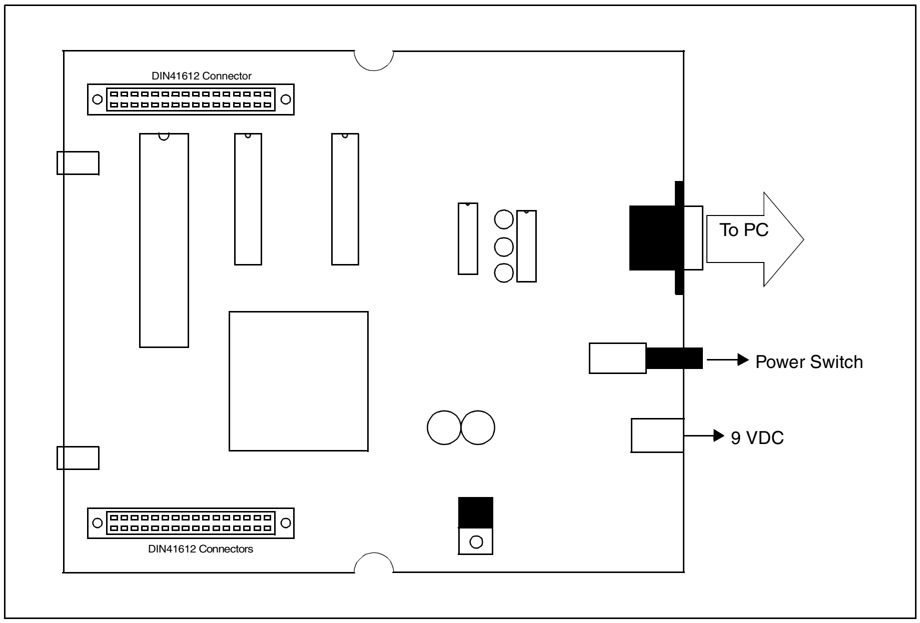


Figure 1.3: Motherboard

The motherboard contains all of the emulation and control logic for all of the supported PICmicro MCU devices. The emulator contains emulation memory, breakpoint logic and control logic. The emulator controls the interfaces to an interchangeable daughter board via two DIN41612 stacking connectors.

1.6.2 Daughter Boards

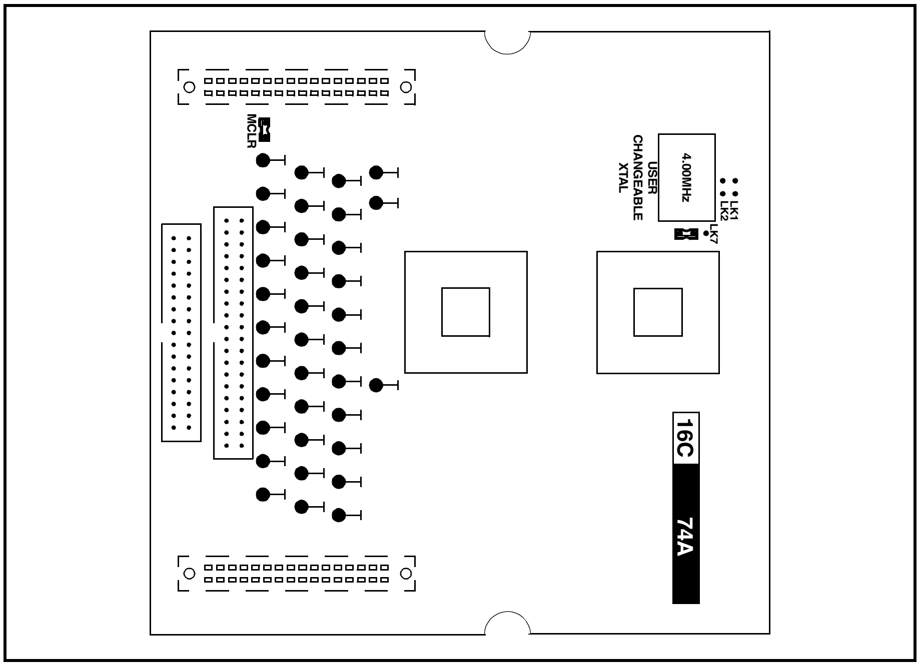


Figure 1.4: Daughter Board (PIC16C74A)

The emulator requires an interface connection to the target microcontroller device that you are emulating. A daughter board specific to the microcontroller family that you are emulating provides this interface connection. The installed daughter board configures the emulator for emulating a target microcontroller and snaps onto the motherboard via two DIN41612 stacking connectors.

Interchangeable daughter boards allow development engineers to easily reconfigure the emulator system for emulating different target processors. The daughter board’s operating frequencies may range up to 20 MHz, depending on the daughter board.

ICEPIC daughter boards for future microcontroller devices will be made available as Microchip releases future devices.

1.7 Installing the ICEPIC In-Circuit Emulator Hardware

This section discusses how to install the daughter board to the motherboard and connect the ICEPIC In-Circuit Emulator to your PC.

Note: Be sure to disconnect the power supply from the pod prior to performing these procedures.

1.7.1 Tools Required

The tools required for this procedure consist of:

- Phillips head screwdriver

WARNING

Care must be taken to avoid exposing the target probe cable to any excessive static or reverse voltage. Therefore, it is strongly recommended you ground yourself prior to handling the ICEPIC motherboard or the daughter boards.

WARNING

For users *OUTSIDE* the UK only!
If the power supply shipped with this unit contains a 3-pin I.E.C. mains input, the earth pin is connected directly to the system 0 volts. This is linked to the target header probe and the RS-232 port.

1.7.2 Removing the Pod Cover

1. Because you will need to connect a daughter board to the ICEPIC motherboard, the ICEPIC pod is not secured together when shipped. The screws that hold the pod together are taped to the inside of the front cover.
2. Carefully remove the top of the pod, followed by the front plate (labeled ICEPIC-16CXXX).

It is not necessary to remove the motherboard from the pod to connect the daughter board.

1.7.3 Inserting the Daughter Board

1. Connect the extender cable to the daughter board.
2. The daughter board will have several factory-preset jumper links. Ensure that these links are set correctly for your application. Refer to Appendix B or the `links.txt` file for more information.
3. Connect the daughter board to the motherboard via the two DIN41612 connectors.

Note:	The connectors are keyed. In other words, they will only fit together in one direction. This helps to ensure correct polarity of the two boards.

4. Ensure that the emulator crystal (XTAL) is securely inserted.
5. Slide the front panel back into position on the pod, ensuring that the extender cable passes through the slot and that the POWER and RUN LEDs are visible.
6. Replace the top of the pod and secure it together with the two pod screws.
7. Connect the RS-232 serial cable from the back of the pod to an available COM port on your PC.
8. Connect the 9 VDC power supply.

Note:	Do not exceed the voltage specifications for the emulator pod or any probe device as described in the respective data sheets.

9. Connect the extender cable into the target application.

Note:	Pin 1 of the extender cable is colored "red." Pin 1 of the DIP header connector is adjacent to the "red" side of the cable.

1.7.4 Removing the Daughter Board

1. Disconnect the power supply from the pod.
2. Disconnect the RS-232 cable from the pod.
3. Remove the two screws that hold the pod together.
4. Remove the top of the pod, followed by the front plate (labeled ICEPIC-16CXXX).
5. Carefully ease the daughter board (the top PCB) from the two DIN41612 connectors.

1.7.5 Applying Power to System Components

The ICEPIC In-Circuit Emulator must be run using external (target board) power.

To prevent damage to any part of the system, power up the system components in the following sequence:

1. Turn on the power to the host computer.
2. Turn on the power to the ICEPIC pod.
3. Turn on the power to the target board.

> **Note:** It is recommended that the $\overline{MCLR}$ pin on the target board is not held low.

The emulator hardware and target application are now ready for emulation.

Turn the system components off in reverse order.

1.7.6 Software Installation

The ICEPIC software is an add-on tool for the MPLAB IDE and works under any Microsoft Windows operating system.

You must install the MPLAB IDE software prior to installing the ICEPIC software. Refer to the *MPLAB IDE User's Guide* (DS51025) for more information. Enter Microsoft Windows.

4. Insert the MPLAB ICEPIC installation disk into drive A.
5. Execute the installation program:

   Windows 3.1: From the File Manage, or from the *Program Manager > Run* option, run `a:\ipmlab.exe`.

   Windows 95/98 or greater: Click the **Start** button and select **Run**. Enter `a:\ipmplab.exe` and click **OK**.

> **Note:** Windows NT® and Windows® 2000 users must have administrative privileges to install the ICEPIC software.

6. Follow the on-line instructions to install the software.

1.8 Setting Up the ICEPIC In-Circuit Emulator

1.8.1 Starting MPLAB IDE

After installing the MPLAB IDE and ICEPIC In-Circuit Emulator software, invoke the MPLAB IDE by executing the file `mplab.exe`. For more information on using the MPLAB IDE software, refer to the *MPLAB IDE User's Guide* (DS51205) and the included `readme.lab` file in the MPLAB directory.

1.8.2 Setting Up the Development Mode

Use the Development Mode dialog to select and setup the ICEPIC In-Circuit Emulator for use with the MPLAB IDE software.

1. Select *Options > Development Mode*.
2. At the Tools tab, select ICEPIC and the processor you are going to use.
3. Click on the **Ports** tab. Select the desired COM port (COM1, COM2, COM3 or COM4) and baud rate settings for ICEPIC Communication.

   No hardware jumpers or switch modifications are necessary. The system is auto-baud capable. In most cases, a baud rate of 38.4K is reliable. If serial I/O errors are reported, try reducing the baud rate to 19.2.
4. Select the **Configuration** tab to enable or disable the watchdog timer.

   Select the **Break Options** tab to choose only the breakpoint options you wish to use while working with the selected processor.
5. Click **OK**.

   If the MPLAB IDE cannot find the ICEPIC In-Circuit Emulator, check your hardware connections (Section 1.7) and then go to step 1. If you still cannot connect, refer to Chapter 5: Troubleshooting.

1.8.3 Setting Up a Project

Create a new project by selecting *Project > New Project*. Find or create a directory for the new project and then name the project (e.g., `newproj.pjt`). For more information on creating and using projects, refer to the *MPLAB IDE User's Guide* (DS51025).

1.8.4 Set Up Problems

If you have difficulty starting the MPLAB IDE or setting up a new project, refer to the *MPLAB IDE User's Guide* (DS51025). If you have difficulty setting up the development mode, refer to Chapter 5: Troubleshooting.

1.9 Using the ICEPIC In-Circuit Emulator

Debugging with the ICEPIC In-Circuit Emulator consists of a three-step process:

1. Program the application code into the ICEPIC In-Circuit Emulator.
2. Debug the application.
3. Modify the source code, rebuild the project and repeat.

1.9.1 Programming Your Application Code

Developing and debugging code in the MPLAB IDE environment is based on projects. Although emulation can be performed without having a project open, projects have the following advantages:

- Single or multiple source files can be easily built and maintained
- Symbolic debugging is available
- The debugging environment can be saved for later use

Some of the information that is retained with a project is:

- Development mode and processor
- Clock source and frequency
- Source files associated with the project
- Name of the final PICmicro executable file
- Open windows and their sizes and positions
- Named break settings
- Configuration bit settings

For more information on creating and using projects, refer to the *MPLAB IDE User's Guide* (DS51025).

1.9.2 Debugging Your Application

You can use the Run, Halt, Step and Animate items on the Debug menu to execute your code. Other Debug menu items reset the processor and change the program counter.

Real-time execution occurs when the processor is put in the MPLAB IDE's Run mode. To begin in real-time mode, open your source file for viewing (*Open > File*). Click the Run tabular button or issue the *Debug > Run > Run* command. The processor will run until a breakpoint is reached or until you halt the processor by selecting *Debug > Run > Halt* from the menu.

Step mode execution can be accessed after the processor is halted. To step through a single statement of code at a time, select *Debug > Run > Step*. The processor halts after each step.

You can also set a breakpoint. Select *Debug > Break Settings* and specify the location at which you want to break. Or, right-click the mouse on the line of source code where you want to break and select Break Point(s).

By default, the Status bar at the bottom of the MPLAB IDE desktop is yellow while the program runs. To change the colors that signify a program run, select *Options > Environment Setup* and click the **Colors** tab.

1.9.3 Modifying Debug Code and Rebuilding Your Hex File

To modify the code being debugged, use MPLAB Projects. Select *File > Open* to edit your source file, then make the changes you feel are necessary for debugging. Next, select *Project > Build All* to recompile and rebuild the hex file. Click **Program** to program the debug device with the updated hex file.

Chapter 2. Tutorial – PIC16CXXX

2.1 Introduction

The ICEPIC In-Circuit Emulator is designed to work with the PIC16C5X and PIC16CXXX family of PICmicro MCUs. This tutorial will help you start using the ICEPIC In-Circuit Emulator hardware and MPLAB IDE.

2.2 Highlights

Topics covered in this chapter:

- Running MPLAB IDE
- Setting Up the Development Mode
- Creating a Project
- Building the Project
- Using Breakpoints
- Using Named Breakpoints

2.3 Running MPLAB IDE

After installing the MPLAB IDE software, invoke it by executing the file `mplab.exe`.

For more information on using the MPLAB IDE, refer to the *MPLAB IDE User's Guide* (DS51025) and the included file `readme.lab`.

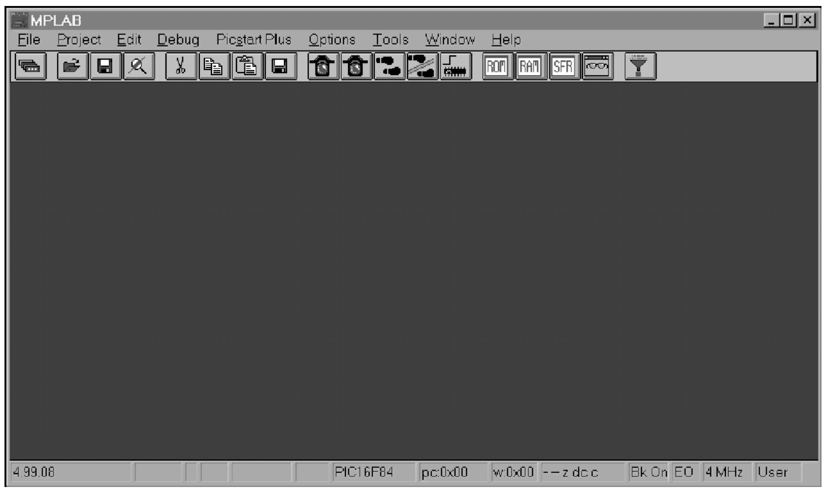


Figure 2.1: MPLAB IDE

2.4 Setting Up the Development Mode

Open the Development Mode dialog (*Options > Development Mode*) to set up the ICEPIC In-Circuit Emulator for use with the MPLAB IDE software. Set up the development mode by clicking on each tab of the dialog and setting the options as specified below.

2.4.1 Tools Tab

The **Tools** tab displays development mode and device information.

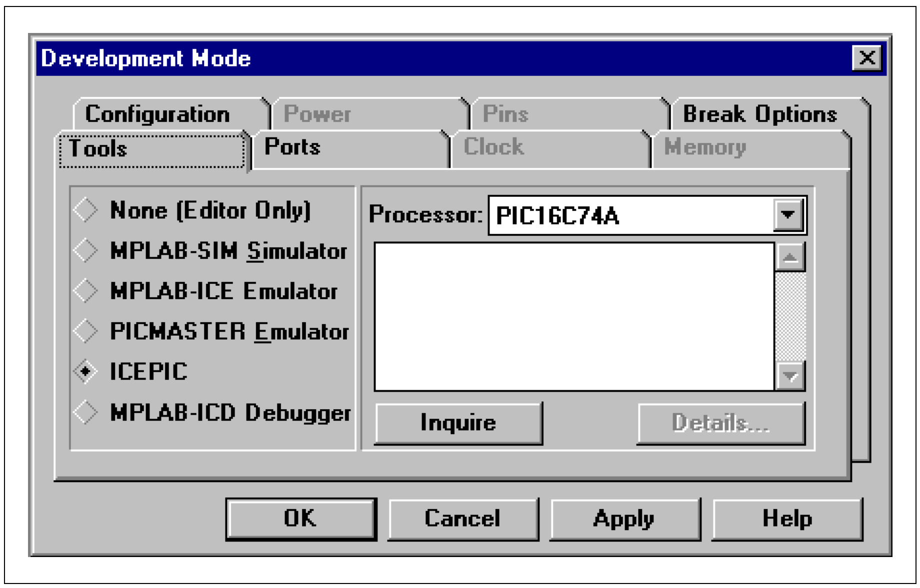


Figure 2.2: Development Mode Dialog – Tools Tab

Select ICEPIC for the development mode.

Choose the PIC16C74A processor to emulate from the drop-down list.

Click **Apply** to accept the setting of this tab.

2.4.2 Ports Tab

The **Ports** tab displays information about the available COM ports on the host PC.

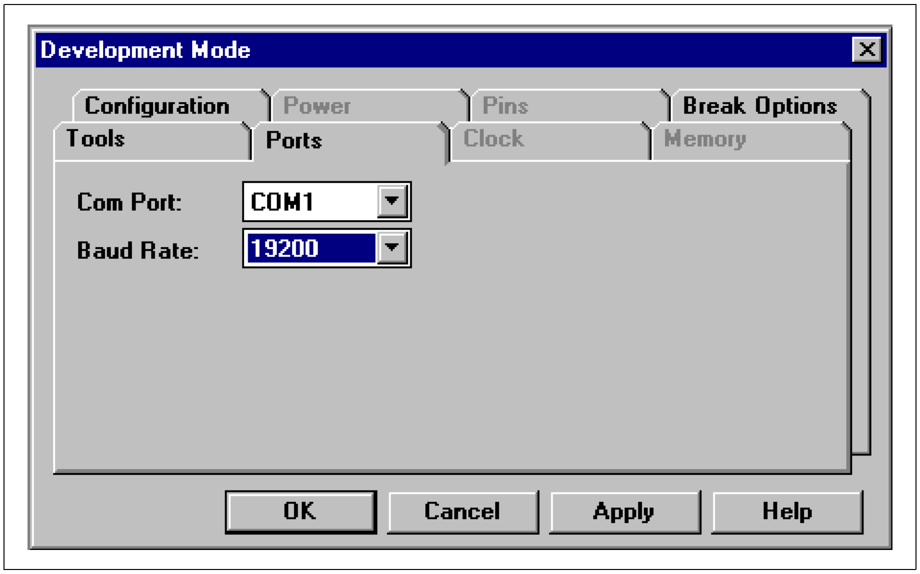


Figure 2.3: Development Mode Dialog – Ports Tab

From the COM Port drop-down list, set the COM port that the ICEPIC In-Circuit Emulator is connected to and the desired baud rate at which the data will travel over the COM port.

Click **Apply** to accept the setting of this tab.

If you have any problems configuring the COM port, refer to Chapter 5: Troubleshooting.

2.4.3 Configuration Tab

The **Configuration** tab controls the settings for the watchdog timer.

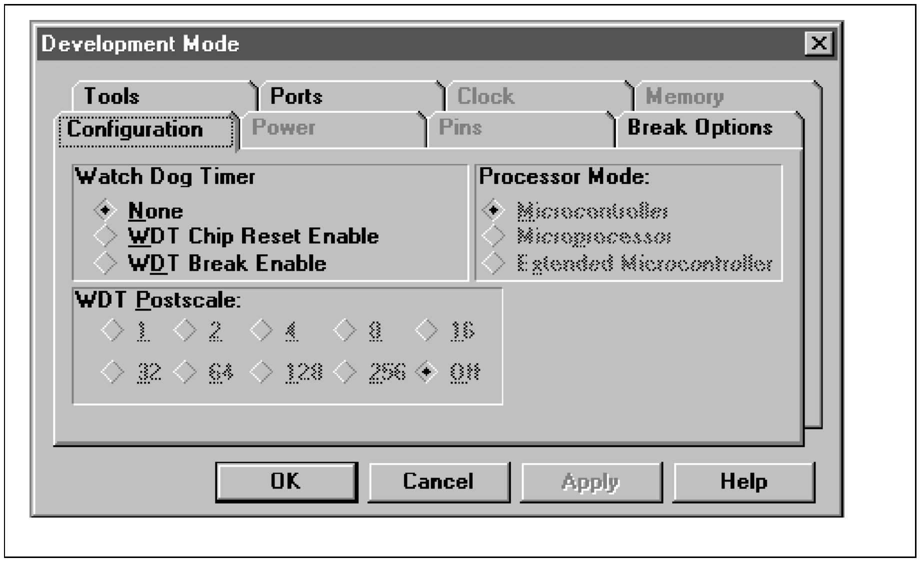


Figure 2.4: Development Mode Dialog – Configuration Tab

The only options available for use with the ICEPIC In-Circuit Emulator are to either enable or disable the watchdog timer to reset the processor when the watchdog timer times out. And, if using a PIC16CXXX device, to execute a break when the watchdog timer time-out generates a reset.

Click **Apply** to accept the setting of this tab.

2.4.4 Break Options Tab

The **Break Options** tab is used to change the global break environment options.

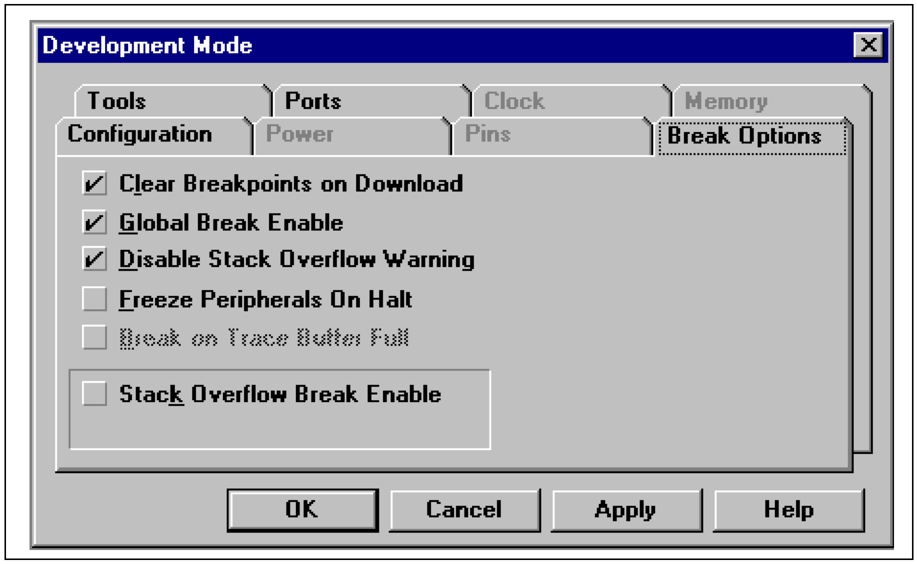


Figure 2.5: Development Mode Dialog – Break Options Tab

The default values are Clear Breakpoints On Download, Global Break Enable and Disable Stack Overflow Warning. Freeze Peripherals On Halt and Stack Overflow Break Enable are also available from this tab. For this tutorial, leave Freeze Peripherals On Halt and Stack Overflow Break Enable unchecked.

Note:	The Break on Trace Buffer Full setting is not available for use with the ICEPIC In-Circuit Emulator.

Click **Apply** to accept the setting of this tab.

2.5 Creating a Project

The best way to develop an application using the MPLAB IDE software is to create a project. In this tutorial, you are going to create a project named `icetut16.pjt`. Before doing this, however, you should create a folder called `icetut16` in which to place the new project. Also, you should copy the file `icetut16.asm` from the MPLAB install directory to this project directory to keep all project files in one place.

Create a new project by selecting *Project > New Project*. The New Project dialog will appear.

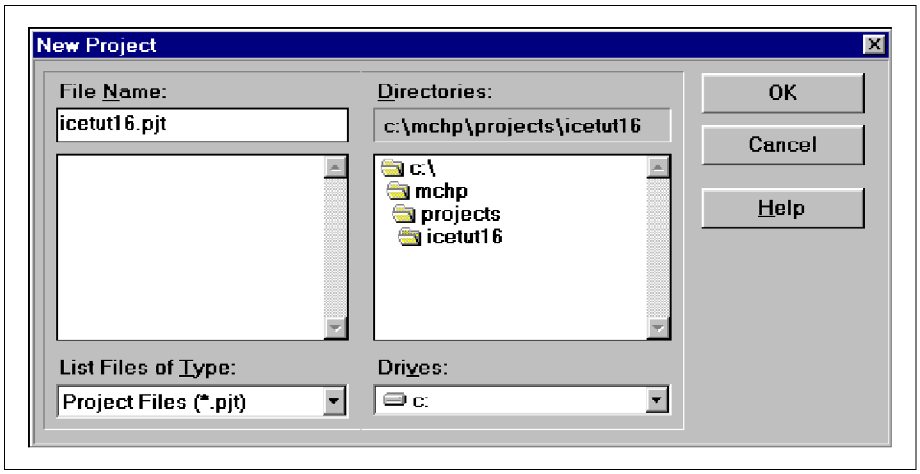


Figure 2.6: New Project Dialog

Find the directory you created for the project. Name the project `icetut16.pjt`. Click **OK** to close this dialog and open the Edit Project dialog.

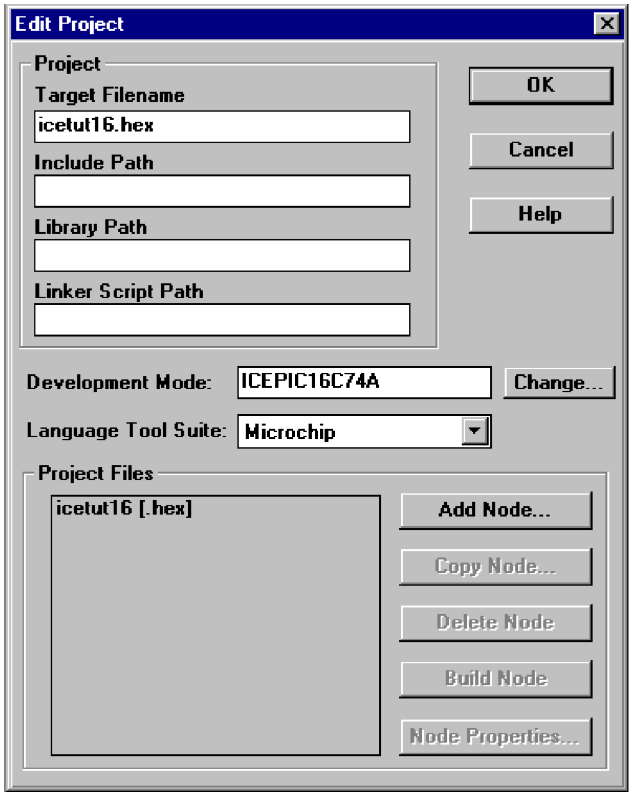


Figure 2.7: Edit Project Dialog – Hex File Only

The only file listed in the Project Files section of this dialog is `icetut16 [.hex]`, which will become the output file.

> **Note:** If you are using a version of MPLAB IDE that has not been recently installed, (i.e., the default configurations may have been changed), you will need to:
>
> - Click on the Hex file to activate the Node Properties button.
> - Click on this button to open the Node Properties dialog.
> - Select MPASM as the Language Tool in this dialog.
> - Click **OK** to close this dialog and return to the Edit Project dialog.

Next to the Project Files section are a group of buttons, one of which, the **Add Node** Button, is active. Click on it now to open the Add Node dialog.

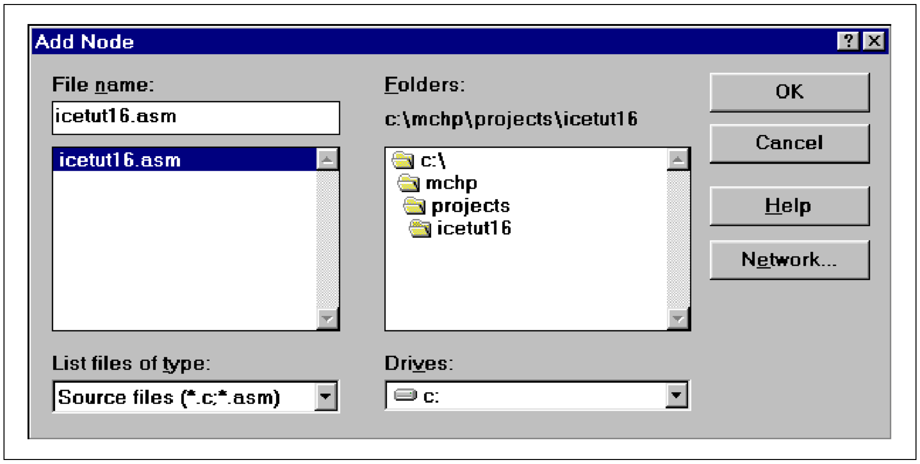


Figure 2.8: Add Node Dialog

Find the `.asm` file you copied into the project directory. Select the file by clicking on its name. If you could not find this file, or if it is missing, type `icetut16.asm` in the File name box of the dialog.

Click **OK** to close this dialog and return to the Edit Project dialog.

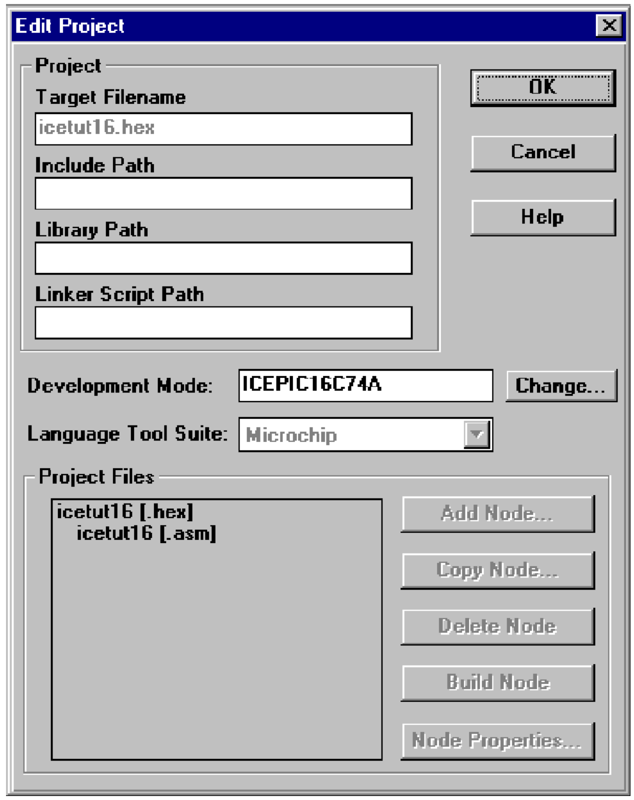


Figure 2.9: Edit Project Dialog – Hex and ASM Files

Now the `.asm` file `icetut16 [.asm]` should be listed below the Hex file `icetut16 [.hex]`. Click **OK** to close this dialog.

If you were able to find and copy the file `icetut16.asm` to the project directory, you should now be in the main MPLAB IDE window (Figure 2.1) with no other windows open. Proceed to Section 2.6.

If you could not find the file `icetut16.asm` to copy to the project directory, you should now be in the main MPLAB IDE window with one empty open file window called Untitiled.

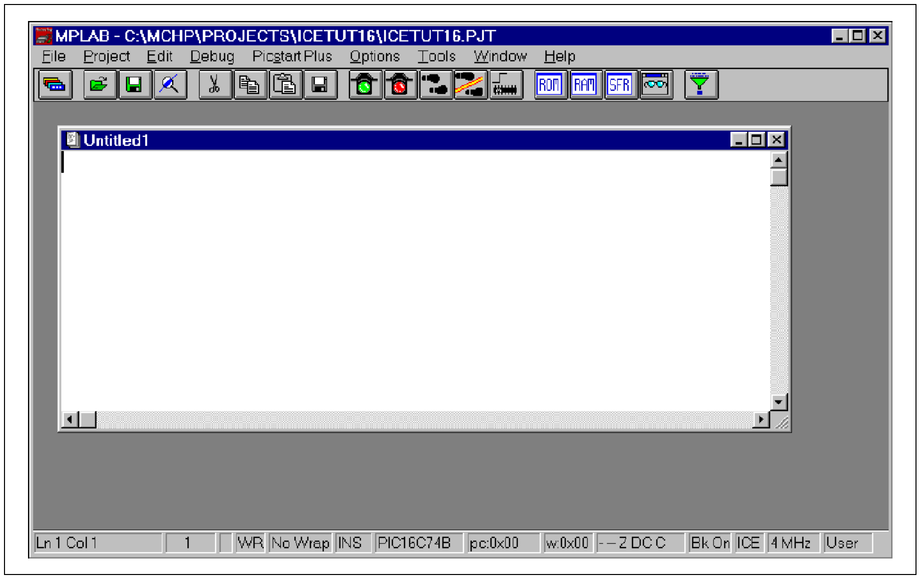


Figure 2.10: MPLAB IDE with Untitled File Window

Give this file window a name by using *File > Save As*. When the Save File As dialog opens, select the project directory `icetut16` and enter `icetut16.asm` as the File Name. Click **OK**. You have now named the file and are ready to begin entering the source code into the empty window (see Section 2.5.1).

2.5.1 icetut16.asm Source Code

```
       list p=16c74a
       title "PIC16C74A Tutorial - Flash PORTB LEDs"

       #include <p16c74a.inc>

TEMP1  equu     0x20              ;Delay loop
TEMP2  equ     0x21              ;temp. variables
TEMP3  equ     0x22

       org     0x00              ;Reset Vector
       goto    START

       org     0x05              ;Start Program
START
       clrf    PORTB             ;Clear PortB (LED's off)

       bsf     STATUS, RP0       ;Select Bank 1
       clrf    TRISB             ;Set PortB as output
       bcf     STATUS, RP0       ;Select Bank 0


LOOP
       movlw   0xFF
       movwf   PORTB             ;Set PortB
       call    DELAY             ;Wait

       clrf    PORTB             ;Clear PortB
       call    DELAY             ;Wait

       goto    LOOP              ;Repeat

;****************************************
;*  This routine is a software delay.  *
;*  Fosc = 1/Tosc; Tcycle = 4 x Tosc   *
;*  Delay = TEMP1xTEMP2xTEMP3xTcycle   *
;****************************************

DELAY

       movlw   0xFF
       movwf   TEMP1             ;TEMP1 = 255
       movwf   TEMP2             ;TEMP2 = 255
       movlw   0x07
       movwf   TEMP3             ;TEMP3 = 7

DLOOP
       decfsz  TEMP1, F
```

```
goto    DLOOP

decfsz  TEMP2, F
goto    DLOOP

decfsz  TEMP3, F
goto    DLOOP

return

END
```

2.6 Building the Project

For this tutorial, building the project is the same as assembling the source code, as there is only one `.asm` file in the project. Select *Project > Build All* to build the project. When complete, the Build Results window will appear.

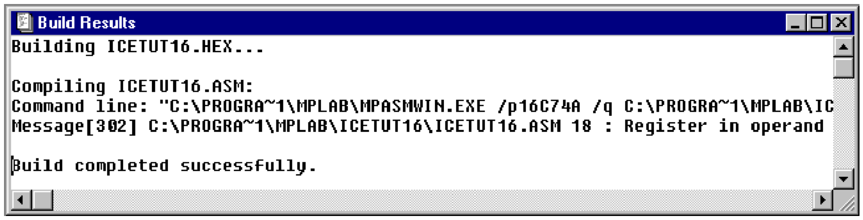


Figure 2.11: Build Results Window

If you entered the source code yourself and the build failed, check your typing and try the build again.

For more information on creating and building projects, refer to the *MPLAB IDE User's Guide* (DS51025).

2.7 Using Breakpoints

Now that your project has been built and the source code successfully assembled into an executable (`.hex`) program, you are ready to run this program on the ICEPIC In-Circuit Emulator using the MPLAB IDE.

Open the source code file in a window by first selecting *File > Open* to open the Open Existing File dialog. Find the project directory in the directory list (Drives, Folders) and select `icetut16.asm` as the File Name. Click **OK**.

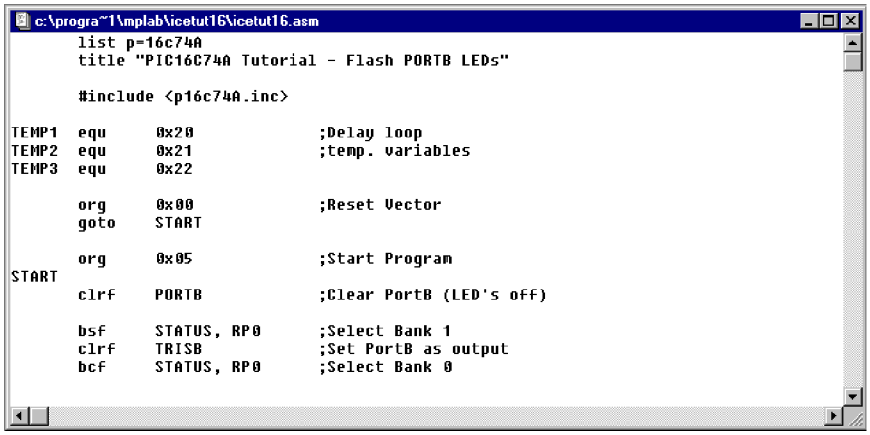


Figure 2.12: Source Code File Window

Next, open a new watch window by selecting *Window > Watch Windows > New Watch Window*. Scroll down in the scroll list of the Add Watch Window dialog until you find PORTB. Click on it to select the PORTB symbol. Click **Add**. The PORTB symbol should now appear in the watch window Watch_1 (Figure 2.13). Click **Close** to close the Add Watch Window dialog.

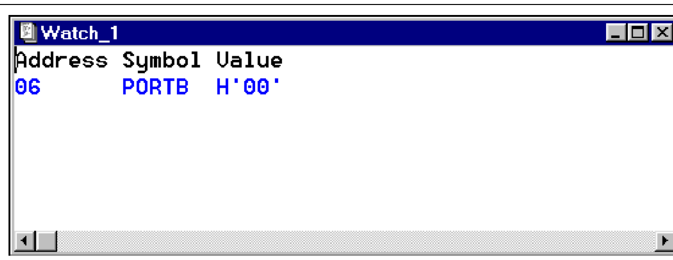


Figure 2.13: Watch Window

Now you will set a breakpoint in the source code file window. Go to the main loop of the program and click the right mouse button on the line `movwf PORTB`. A menu will appear. Select Break Point(s). The line of code will change color. If you don't like this color, you may change it by selecting *Options > Environment Setup*, clicking the **Color** tab and selecting a different Color for Break Point Text.

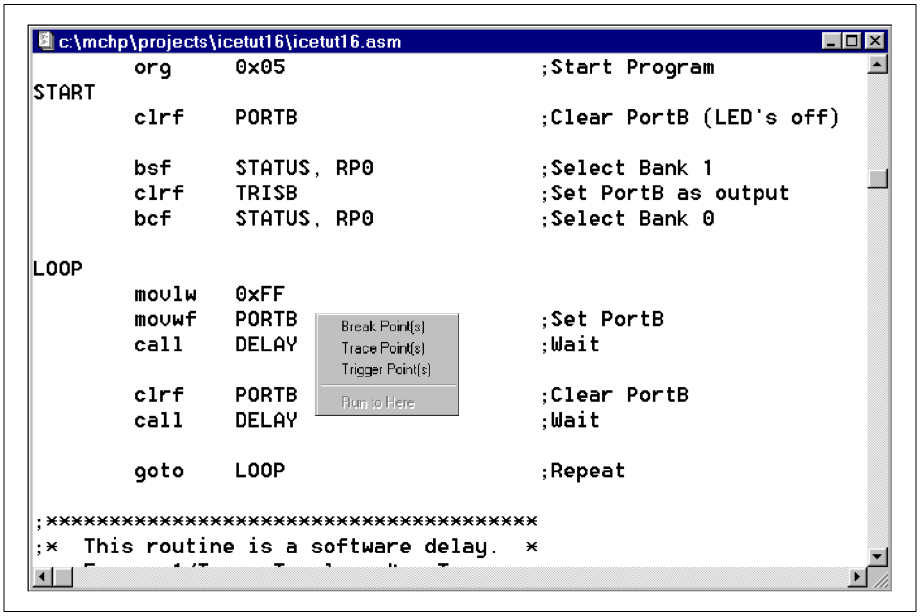


Figure 2.14: Set Breakpoint

Run the program by selecting *Debug > Run > Run* or by clicking the toolbar green stoplight icon. The status bar on the bottom of the MPLAB IDE window will change color, indicating that the program is running. The program will halt when it reaches the breakpoint.

Single step by selecting *Debug > Run > Step*, by pressing <F7> or by clicking the toolbar single step icon. You should now see that the value of PORTB in the Watch_1 watch window has changed from `H'00'` to `H'FF'` and should be a different color.

This demonstrates how breakpoints work with the ICEPIC In-Circuit Emulator. The breakpoint halts an emulation run after the line of code it is associated with is executed.

2.8 Using Named Breakpoints

MPLAB IDE allows up to 16 named breakpoints. These breakpoints can be selectively enabled and disabled. Breakpoints set in this manner are retained with the project. Right mouse button menu breakpoints are not.

Reset the program by selecting *Debug > System Reset* or by clicking on the toolbar reset processor icon. This will clear all breakpoints.

Open the Break Point Settings dialog by selecting *Debug > Break Settings*. Enter `0x000A` into the Start box of the dialog to place the **break1** breakpoint at the instruction `movwf PORTB`. Click **Add**.

Note:	To determine the address of an instruction, use the Program Memory Window (*Window > Program Memory*).

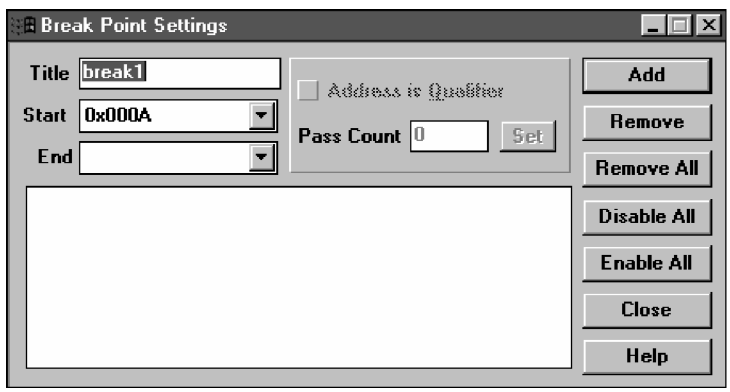


Figure 2.15: Break Point Settings Dialog

You should see **break1** enabled at location `0x000A` in the Break Point Settings dialog and you should see the corresponding instruction line change color in the source code file window. Click **Close** on the Break Point Settings dialog.

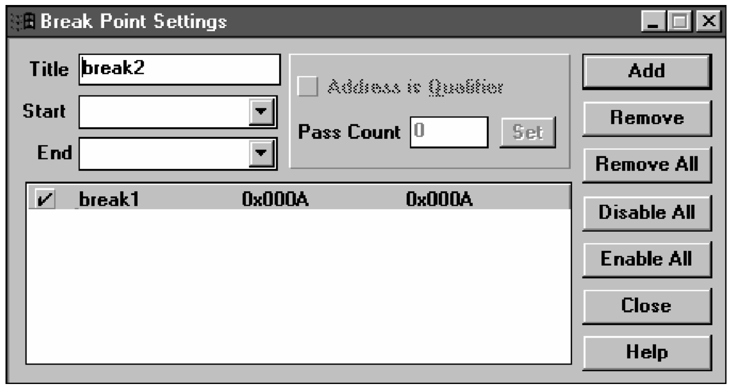


Figure 2.16: Break Point break1 Set

Run the program by selecting *Debug > Run > Run* or by clicking the toolbar green stoplight icon. The status bar on the bottom of the MPLAB IDE window will change color, indicating that the program is running. The program will halt when it reaches the breakpoint.

You may single step to see the value of PORTB in the Watch_1 watch window change from `H'00'` to `H'FF'`.

Chapter 3. General Setup

3.1 Introduction

This chapter describes how to get the hardware and software for the ICEPIC In-Circuit Emulator up and working.

3.2 Highlights

Topics covered in this chapter:

- Running the ICEPIC In-Circuit Emulator
- Setting Up the Development Mode
- Using MPLAB Projects

3.3 Running the ICEPIC In-Circuit Emulator

After installing the MPLAB IDE, invoke it by executing the file `mplab.exe`.

For more information on installing and using MPLAB IDE, refer to the *MPLAB IDE User's Guide* (DS51025) and the included file `readme.lab` in the MPLAB install directory.

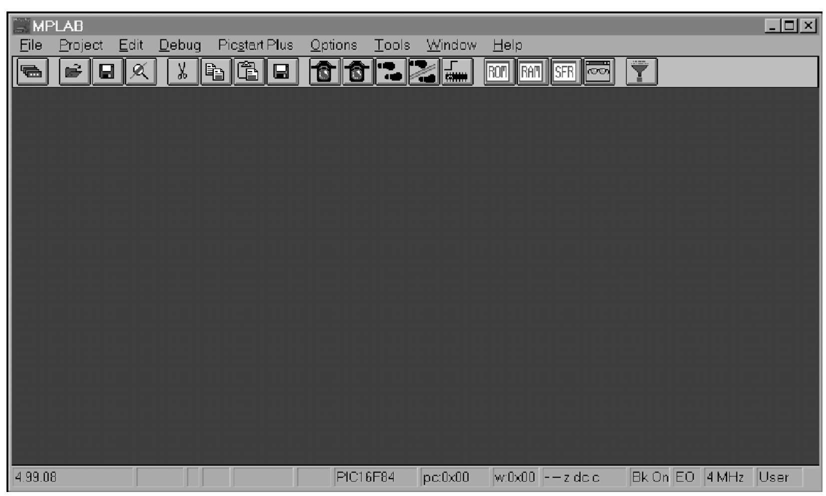


Figure 3.1: MPLAB IDE Desktop

3.4 Setting Up the Development Mode

Use the Development Mode dialog to select the ICEPIC In-Circuit Emulator for use with the MPLAB IDE software.

1. Select *Options > Development Mode* to display the Development Mode dialog.

.

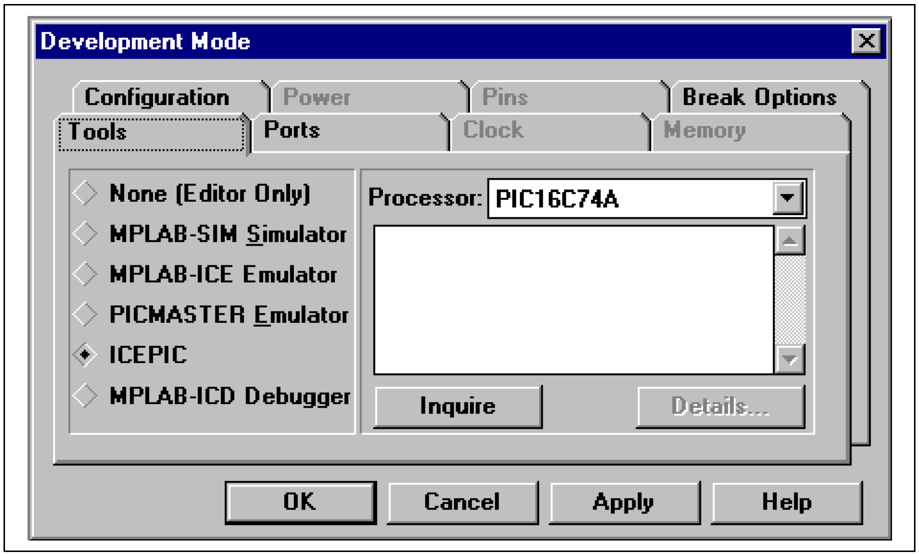


Figure 3.2: Development Mode Dialog

2. Under the **Tools** tab, select the ICEPIC development mode and the processor you are going to use.

Note:	The MPLAB IDE will only display the processors that are compatible with the daughter board installed in the ICEPIC In-Circuit Emulator.

You may also access the Development Mode dialog from the Edit Project dialog by clicking Change next to the Development Mode item. See Chapter 2 for more information on setting up projects.

If the MPLAB IDE cannot find the ICEPIC In-Circuit Emulator, check your hardware connections (Section 1.7) and then go to Step 1. If you still cannot connect, see Chapter 5: Troubleshooting.

3. Select the **Ports** tab to set the COM port that the ICEPIC In-Circuit Emulator is connected to and the desired baud rate at which the data will travel over the COM port.
4. Select the **Configuration** tab to enable or disable the watchdog timer to reset the processor when the watchdog timer times out.
5. Select the **Break Options** tab to change the global break options.
6. When you are finished setting up the ICEPIC In-CIrcuit Emulator, click **OK**.

3.5 Using MPLAB Projects

The MPLAB IDE is the host software for the ICEPIC In-Circuit Emulator. It functions as a sophisticated debugging tool, providing access to RAM, ROM, EEPROM and a variety of other debug functions.

3.5.1 MPLAB IDE Project Features

Developing and debugging code in the MPLAB IDE is based on projects. Although emulation can be performed without having a project open, projects have the following advantages:

> **Note:** If you do not put your source files into a project, the MPLAB IDE cannot debug properly.

- Single or multiple source files can be easily built and maintained.
- Symbolic debugging is available.
- The debugging environment can be saved for later use.

Some of the information that is retained with a project is:

- Development mode and processor
- Source files associated with the project
- Name of the final PICmicro MCU executable file
- Open windows and their sizes and positions
- Named break settings
- Configuration bit setting

> **Note:** The CONFIG bit set in your source code is not applied to the system under emulation. Use the Development Mode dialog (*Options > Development Mode > Configuration Tab*) to set them.

3.5.2 Creating a Project

Select *Project > New Project* to create a project and open the Edit Project dialog. The project will contain information about your source, object and other files, as well as a variety of important project settings.

3.5.3 Saving a Project

To save your current project, to retain the values for later use, or to use as a backup or default as you continue with your debugging, click **OK** in the Edit Project dialog. Then, select *Project > Save Project* to save your project.

DEVELOPMENT MODE

The selected development mode is retained with the project information. To change the development mode for a project, follow these steps:

1. Open the project. The previously used development mode will be selected.
2. Change the development mode by selecting *Options > Development Mode* to access the Development Mode dialog. Select the development mode and click **OK**.
3. Select *Project > Save Project* to save the project.

For more information on creating and using projects, refer to the *MPLAB IDE User's Guide* (DS51025).

Chapter 4. Basic Functions

4.1 Introduction

This chapter briefly discusses the basic MPLAB IDE debugging functions of the ICEPIC In-Circuit Emulator. For more information on general debugging features, refer to the *MPLAB IDE User’s Guide* (DS51025).

4.2 Highlights

Topics covered in this chapter:

- Program Execution
- Breakpoints
- Conditional Break
- Verify Emulator – Diagnostics Program
- Program Memory in the ICEPIC In-Circuit Emulator

4.3 Program Execution

The ICEPIC In-Circuit Emulator executes in real-time mode or in polled mode.

- Real-time execution occurs when the processor is put in the MPLAB IDE's Run mode.
- Polled execution occurs when you single step the processor, modify values at a breakpoint or execute an opcode.

4.3.1 Real-Time Execution

When the ICEPIC In-Circuit Emulator is run in real time, instructions execute just as the processor would without the emulator. The processor executes in real time until a breakpoint halts the emulator or until the HALT function is manually executed. To execute in real time, click **Run** from the Debug toolbar button or issue the *Debug > Run > Run* command. The Debug toolbar provides Run, Halt and Step buttons for controlling the emulator. While in Run mode, register displays on the screen will not update.

> **Note:** Double clicking on an instruction in the Program Memory window runs the emulator until the program counter reaches that same instruction again or until the program hits a valid breakpoint.

4.3.2 Polled Execution

Polled execution provides the capability to:

- Step through code, one instruction at a time, watch the program flow and see all register contents.
- Force the emulator to execute any single opcode.
- Break on a register value or condition.

In MPLAB IDE, the Execute menu options allow you to control the polled execution of your firmware in the target processor.

To implement an opcode polled execution, select Execute an Opcode from the Debug toolbar or issue the *Debug > Execute > Execute an Opcode* command. To execute a conditional break polled execution, select Conditional Break from the Debug toolbar or issue the *Debug > Execute > Conditional Break* command.

4.3.3 Animate Mode

Animate Mode is a method of automatically single-stepping the processor. 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. To implement Animate mode, select *Debug > Run > Animate*.

4.4 Breakpoints

A breakpoint is a state where the processor halts after a certain condition is met. The breakpoints are software breakpoints and can be set at any program memory address location.

The ICEPIC In-Circuit Emulator provides the following ways to set a breakpoint:

- Break on Address Match
- Break on Stack Overflow
- Break on Watchdog Timer
- Break on User Halt

> **Note:** MPLAB IDE limits the number of named events to a maximum of 16 in the breakpoint dialog.

4.4.1 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 before 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.

> **Note:** On multicycle instructions, the ICEPIC In-Circuit Emulator ignores extra (noninstruction) fetches and breaks only on a valid instruction

4.4.2 Break on Stack Overflow

Break on Stack Overflow causes the ICEPIC In-Circuit Emulator to execute a break when the stack overflows.

> **Note:** The ICEPIC In-Circuit Emulator only supports Break on Stack Overflow for the PIC16CXXX family.

4.4.3 Break on Watchdog Timer

If enabled, the ICEPIC In-Circuit Emulator executes a break when a watchdog timer time-out generates a reset.

> **Note:** The ICEPIC In-Circuit Emulator only supports Break on Watchdog Timer for the PIC16CXXX family.

4.4.4 Break on User Halt

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

- Click *Debug > Run > Halt*
- Press F5
- Click the HALT icon (red stop light)

4.5 Conditional Break

When a conditional break is set, MPLAB IDE halts when the value of a specified internal register reaches a preset value or condition.

You can set the conditional break by selecting the *Debug > Execute > Conditional Break* command.

4.5.1 Single Cycle

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

4.5.2 Multiple Cycles

In Multiple Cycles mode, the MPLAB IDE checks the conditions at breakpoints you define.

4.5.3 Conditions

The ICEPIC In-Circuit Emulator will stop at a breakpoint in the Conditional Break dialog based on one of the following conditions:

- **User Halt** – The MPLAB IDE executes until you press the **Halt** button on the Conditional Break dialog.
- **Number of Cycles** – The MPLAB IDE halts after the target processor executes the specified number of cycles.
- **Register Value Conditions** – The MPLAB IDE halts only when the register value hits a specific condition.

4.6 Verify Emulator – Diagnostics Program

To verify the ICEPIC In-Circuit Emulator, you will need to run the Diagnostics program. This program is part of the initial ICEPIC installation. This is a separate executable program (`icediags.exe`) and is installed by the `ipmplab.exe` program onto your hard disk. Before running the diagnostic program, you must first:

- Disconnect the ICEPIC system from the target application board.
- Connect the $\overline{\text{MCLR}}$ pull-up jumper (Link3, LK3).
- Make sure that the clock and power are present.

To access the Diagnostics program, select the ICEDIAGS icon in the folder where you installed the MPLAB ICEPIC software to display the ICEPIC Test Menu.

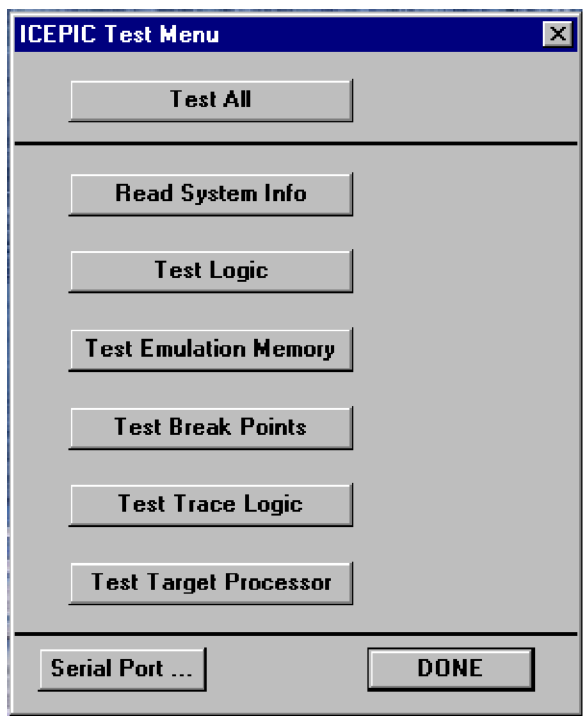


Figure 4.1: ICEPIC Test Menu Dialog

Click the **Test All** button to test all functions. Each block may be tested individually by clicking the desired test button. If a function block does not pass the test, failure is displayed by ‘Failed’ instead of ‘Pass’.

Note:	If any of the tests fail, note down the error and contact RF Solutions for assistance. See the General Information chapter for more information.

4.6.1 ICEPIC Test Menu Items

The information below describes the functions for each test button.

4.6.1.1 Read System Info

This button checks the system information, as described in Table 4.1.

Table 4.1: System Info

System Info	Description
Name	"ICEPIC" is displayed.
Series	A, B, etc., depending on your system.
Hardware Version	Displays the hardware (board) revision number.
Firmware Version	Displays the controller Firmware Version number.
Trace Support	Displays YES or NO depending on whether your system supports trace.
Target Board	Displays all the processors your daughter board supports.
Target ID	An ID number (HEX) is displayed. This information is for factory testing only.
ID String	An ID string of your system. This information is for factory testing only.

4.6.1.2 Test Logic

Tests various functions of the system logic.

4.6.1.3 Test Emulation Memory

Tests the emulation RAMs in your ICEPIC system. Every location of the memory is tested. If an error is encountered, the address, expected data and actual data are reported.

4.6.1.4 Test Breakpoints

Tests the breakpoint logic. Both the system logic related functions and the emulator bond-out chips are tested.

4.6.1.5 Test Trace Logic

This button is enabled only if your system has hardware trace support.

4.6.1.6 Test Target Processor

Tests the target processor for various functions. Before testing, make sure the system is not connected to the target application, since this test modifies various I/O pins and special function registers.

4.6.1.7 Command Line Parameters

The installation program sets the command line parameters for the diagnostics program during installation. However, if the ICEPIC In-Circuit Emulator is reconnected to a different COM port, the following parameters must be set:

-C# where # is the COM port number (1, 2, 3 or 4)

-B# where # is the desired baud rate (2400, 4800, 9600, 19200, 38400 or 57600)

For example: If the ICEPIC In-Circuit Emulator is connected to COM2 and you wish to test it at 9600 baud, then the correct command line is:

```
icediags.exe -c2 -b9600
```

4.7 Program Memory in the ICEPIC In-Circuit Emulator

The ICEPIC In-Circuit Emulator emulates program memory as described below.

4.7.1 PIC16C5X and PIC16CXXX Series

The PIC16C5X and PIC16CXXX series have no external memory capability. (All the program memory resides on the chip.) For these families, during development with the ICEPIC In-Circuit Emulator, all program code resides in emulation program memory.

NOTES:

Chapter 5. Troubleshooting

5.1 Introduction

This chapter describes some common problems associated with running the ICEPIC In-Circuit Emulator and the steps to follow to resolve those problems.

5.2 Highlights

Topics covered in this chapter:

- Common Problems

5.3 Common Problems

Communications cannot be established with the ICEPIC In-Circuit Emulator.

SOLUTIONS:

If you cannot establish communications with the ICEPIC In-Circuit Emulator, follow these steps:

1. Make sure there is power to the pod. The ICEPIC In-Circuit Emulator is powered by a 9 VDC power supply.
2. The ICEPIC system communicates with the host via a standard RS-232 port. This requires that you have at least one free serial port on your system. The desired COM port must be supported by the Windows operating system.

   If the MPLAB IDE is unable to detect the ICEPIC system on the selected COM port, try another COM port, especially if the COM ports are COM3 or COM4. Try switching to COM1 or COM2.
3. If occasional serial communication errors are reported, reduce the baud rate and perform a system reset.
4. Make sure you are using the most current version of the MPLAB ICEPIC (`ipmplab.exe`) software. You can obtain the most current version from Microchip's web site (www.microchip.com) or from RF Solutions (www.rfsolutions.co.uk).

The Emulator Clock/Target System Clock is not present.

SOLUTION:

Ensure you have a clock present at all times. If a clock is not present, configure the target system clock via a jumper on the daughter board to select the internal emulator clock or the user target system clock.

- If the emulator is to provide the system clock from the on-board canned oscillator, set the internal clock jumper (INTCLK).
- If the clock on the target application is to provide the system clock, set the external clock jumper (EXTCLK).

> **Note:** CMOS clock levels are required and a crystal CANNOT be used.

Breakpoints disappear on download.

SOLUTIONS:

If breakpoints disappear when you download a file, the global switch, Clear Memory on Download, may be selected in the *Options > Environment Setup* dialog box.

With the Clear Memory on Download option selected, the MPLAB IDE software will clear emulation memory before downloading a file.

Double click the breakpoints toggle in the status bar to turn breakpoints on and off.

Stack overflow or underflow keeps occurring.

Stack overflow or underflow may occur when emulating some PICmicro MCU devices. The following details the ICEPIC In-Circuit Emulator hardware stack levels by family:

- 2-Levels Deep Hardware Stack (PIC16C5X)

  Some data may be left in the stack as the PIC16C5X device has no way to clear the stack. No overflow or underflow is supported for the PIC16C5X.

- 8-Levels Deep Hardware Stack (PIC16CXXX)

  The Host Software displays an underflow or overflow message when you set Stack Break Enable on the Hardware configuration display and push or pop the stack beyond its limit. Select *Options > Processor Setup > Hardware* to display the Hardware configuration dialog box.

Appendix A. Specifications

A.1 Introduction

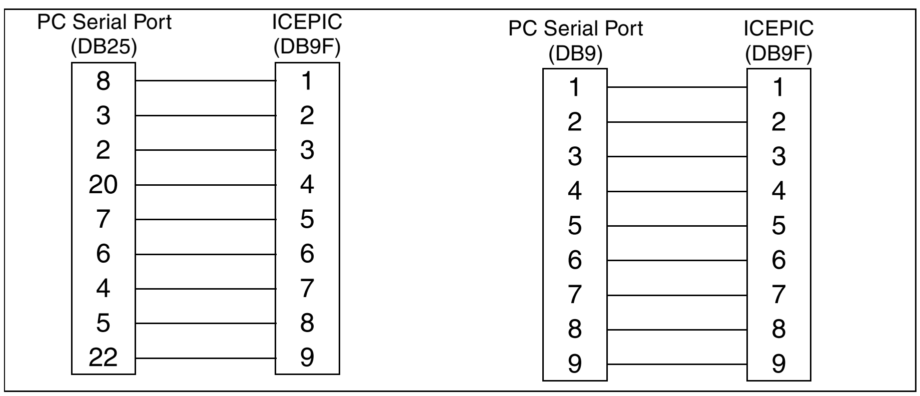
This appendix discusses various specifications for the ICEPIC In-Circuit Emulator.

Topics covered in this appendix:

- Serial Port Specifications
- Achieving High Baud Rates
- Determining the UART

A.2 Serial Port Specifications

The ICEPIC In-Circuit Emulator interfaces to the host computer via an RS-232 connection. A standard DB9 connector used.



The ICEPIC serial communications protocol is listed below.

Protocol	Description
COM Port	Any port supported by the Microsoft Windows operating system. The port is selectable from the software.
Baud Rate	2400 BPS to 57600 BPS. Selected from software. The emulator system automatically detects the baud rate.
Parity	None. Fixed, cannot be selected by the user.
Data Bits	Fixed to 8 bits. Cannot be set by the user.
Stop Bits	One Stop Bit. Cannot be selected by the user.
Software Receive Buffer	Fixed to 2048 bytes.
Software Transmit Buffer	Fixed to 2048 bytes.
Firmware Buffer	Fixed to 64 bytes.

A.3 Achieving High Baud Rates

The following list provides ideas on how to achieve high baud rates with your PC.

- Use an I/O Card with a 16550 processor.
- Older IBM® PC compatibles were equipped standard with 8250 UART. Newer computers come with 16550 processors (especially notebook computers). Unlike the 8250, the 16550 UARTs have a 16 byte FIFO which improves the speed of communications. Microsoft Windows will have enough time before servicing an interrupt.
- Upgrade your Windows 3.1 to one of the new Windows operating systems. Even if your computer is not networked, the new Windows operating system's serial port driver are significantly better.
- Upgrade you computer to at least 64 MB RAM.
- Remove any screen savers. These slow down your Windows operation.

There are various third party COM Drivers on the market, which are replacements for the standard driver that comes with the Microsoft Windows operating system. These drivers are supposedly faster, but ICEPIC software has not been tested with these drivers.

A.4 Determining the Baud Rates

To determine the baud rate setting for your COM port, go to your DOS prompt and enter `mode`.

If you have a newer system, the baud rate should be set to at least 1200.

To change your baud rate, see your Microsoft Windows documentation.

Appendix B. Daughter Board Specifications

B.1 Introduction

This appendix discusses the various specifications and jumper link configuration settings for those daughter boards that were available at the time this document was published.

For more current information, refer to the `links.txt` file in the MPLAB ICEPIC disk.

Topics covered in this appendix:

- OSC Module Configuration
- MCLR
- OSC Input – Internal or External Sources
- Daughter Board Links

B.2 OSC Module Configuration

These links configure the system emulator chip. For normal operation with the OSC Module supplied, these links should be left open.

For all PICmicro MCU pins, OSC1 and OSC2 can be input or output. The only mode where OSC2 in a output is RC mode. Then, the output from OSC2 is 1/4 of the frequency at OCS1.

The function of the PIC16C01 or PIC16C02 is the same as for any PICmicro MCU with the mode of operation being set by the two links A and B on the personal daughter board, just above the OSC module.

Link A	Link B	Mode
Open	Open	RC
Open	Closed	HC
Closed	Open	XT
Closed	Closed	LP

The ICEPIC Daughter Board is driven from an EXT source (CMOS/TTL output Oscillator Module) which feeds into OSC1. Therefore, if you connect the Links to RC mode, you should see the 1/4 frequency on OSC2.

B.3 MCLR

This link connects a weak pull-up resistor (27 kΩ) to the emulator MCLR input. This link must be connected to enable all functions such as a single step, animate or when the emulator is halted. It may be removed when running in real time, thereby truly emulating a reset condition.

B.4 OSC Input – Internal or External Sources

This link enables the user to select the clock from either the on-board user-changeable OSC module (INT position) or from the clock on OSC1 pin 1 of the user's target board (EXT position). Due to the inherent nature of emulators and cables, use of an XTAL on a your target board cannot be guaranteed. For reliable operation, use the INT option.

B.5 Daughter Board Links

Daughter Board	OSC Mod Link A and B	MCLR	OSC/IP	Link 6	Comments
DB12C67	LK1 and LK2	LK3		Connect between positions 2 and 3.	
DB5X	LK1 and LK2	LK3	N/A	Connect between positions 2 and 3.	
DB55X	LK1 and LK2	LK3	N/A	Connect between positions 2 and 3.	
DB62X	LK1 and LK2	LK3	LK7	Connect between positions 2 and 3.	
DB64X/66X	LK1 and LK2	LK3	LK7	N/A	
DB711	LK1 and LK2	LK3	LK7	Must be connected between 2 and 3. In case of PIC16C61 emulation (or to disable the A/D Converter, LK6 must be connected between position 1 and 2.	
DB715	LK1 and LK2	LK3		Connect between positions 2 and 3.	
DB74A	LK1 and LK2	LK3	LK7	For PIC16C7X devices connect between 1 and 2. For other processors, connect between 2 and 3.	1 and 2
DB77	LK1 and LK2	LK3	LK7	N/A	
DB84A	LK1 and LK2	LK3	LK7	Connect between positions 2 and 3.	
DB92X	LK1 and LK2	LK3	J1	Connect between positions 2 and 3.	3 and 4

Note 1: When using the A/D converter, the system +5V supply is connected to the Analog V_{DD} (AV_{DD}). If the user wishes to supply a higher precision, low-noise voltage to the AV_{DD} pin, then the link may be removed and the external Analog Power Supply fed in directly to pin 2. Care must be taken not to exceed the normal power ratings of the AV_{DD} pin.

2: Issue 4 and later of this daughter board does not have Link 6, as it is controlled automatically by the ICEPIC software.

3: Jumper Link J7 is currently Not Used.

4: J8 provides a link to enable the OSC2/CLKOUT (Center pin 2) signal of the target probe cable to be connected to either the OSC2/CLKOUT (pin 1) of the PIC16C02 or the OSC module output (pin 1). This provides a direct output of the OSC module signal to the target cable.

Glossary

Introduction

To provide a common frame of reference, this glossary defines the terms for several Microchip tools.

Highlights

This glossary contains terms and definitions for the following tools:

- MPLAB IDE, MPLAB-SIM, MPLAB Editor
- MPASM, MPLINK, MPLIB
- MPLAB-CXX
- MPLAB-ICE and PICMASTER emulators
- MPLAB-ICD
- PICSTART Plus, PRO MATE programmer

Terms

Absolute Section

A section with a fixed (absolute) address that cannot be changed by the linker.

Access RAM (PIC18CXXX Devices Only)

Special general purpose registers on PIC18CXXX devices that allow access regardless of the setting of the bank select bit (BSR).

Alpha Character

Alpha characters are those characters, regardless of case, that are letters of the alphabet: (a, b, …, z, A, B, …, Z).

Alphanumeric

Alphanumeric characters include alpha characters and numbers: (0,1, …, 9).

Application

A set of software and hardware developed by the user, usually designed to be a product controlled by a PICmicro microcontroller.

Assemble

The process of translating assembly source code into an executable program. This operation is performed by an assembler.

Assembler

A language tool that translates a user's assembly source code (`.asm`) into machine code. MPASM is Microchip's assembler.

Assembly

A programming language that is once removed from machine language. Machine languages consist entirely of numbers and are almost impossible for humans to read and write. Assembly languages have the same structure and set of commands as machine languages, but they enable a programmer to use names (mnemonics) instead of numbers.

Assigned Section

A section which has been assigned to a target memory block in the linker command file. The linker allocates an assigned section into its specified target memory block.

Breakpoint – Hardware

An event whose execution will cause a halt.

Breakpoint – Software

An address where execution of the firmware will halt. Usually achieved by a special break opcode.

Build

A function that recompiles all the source files for an application.

C

A high level programming language that may be used to generate code for PICmicro MCUs, especially high-end device families.

Calibration Memory

A special function register or registers used to hold values for calibration of a PICmicro microcontroller on-board RC oscillator or other device peripherals.

COFF

Common Object File Format. An intermediate file format generated by MPLINK that contains machine code and debugging information.

Command Line Interface

Command line interface refers to executing a program on the DOS command line with options. Executing MPASM with any command line options or just the file name will invoke the assembler. In the absence of any command line options, a prompted input interface (shell) will be executed.

Compile

The process of translating C source code into machine code. This operation is performed by a compiler.

Compiler

A language tool that translates a user's C source code into machine code. MPLAB-C17 and MPLAB-C18 are Microchip's C compilers for PIC17CXXX and PIC18CXXX devices, respectively.

Configuration Bits

Unique bits programmed to set PICmicro microcontroller modes of operation. A configuration bit may or may not be preprogrammed. These bits are set in the *Options > Development Mode* dialog for simulators or emulators and in the __CONFIG MPASM directive for programmers.

Control Directives

Control directives in MPASM permit code to be conditionally assembled.

Data Directives

Data directives are those that control MPASM's allocation of memory and provide a way to refer to data items symbolically; that is, by meaningful names.

Data Memory

General purpose file registers (GPRs) from RAM on the PICmicro device being emulated. The File Register window displays data memory.

Daughter Board

The ICEPIC In-Circuit Emulator (Third Party product) uses interchangeable daughter boards to support the different subsets of the PIC16C5X and PIC16CXXX family of PICmicro microcontrollers. The daughter board is a circuit board that connects to the motherboard.

Directives

Directives provide control of the assembler's operation by telling MPASM how to treat mnemonics, define data and format the listing file. Directives make coding easier and provide custom output according to specific needs.

Download

Download is the process of sending data from the PC host to another device, such as an emulator, programmer or target board.

EPROM

Erasable Programmable Read Only Memory. A programmable read-only memory that can be erased usually by exposure to ultraviolet radiation.

EEPROM

Electrically Erasable Programmable Read Only Memory. A special type of PROM that can be erased electrically. Data is written or erased one byte at a time. EEPROM retains its contents even when power is turned off.

Emulation

The process of executing software loaded into emulation memory as if the firmware resided on the microcontroller device under development.

Emulation Memory

Program memory contained within the emulator.

Emulator

Hardware that performs emulation.

Emulator System

The MPLAB-ICE emulator system includes the pod, processor module, device adapter, cables and MPLAB Software. The PICMASTER emulator system includes the pod, device-specific probe, cables and MPLAB Software. The ICEPIC In-Circuit Emulator (Third Party product) is another emulator system.

Event

A description of a bus cycle which may include address, data, pass count, external input, cycle type (fetch, R/W) and time stamp. Events are used to describe triggers and breakpoints.

Executable Code

See Hex Code.

Export

Send data out of the MPLAB IDE in a standardized format.

Expressions

Expressions are used in the operand field of MPASM's source line and may contain constants, symbols, or any combination of constants and symbols separated by arithmetic operators. Each constant or symbol may be preceded by a plus or minus to indicate a positive or negative expression.

> **Note:** MPASM expressions are evaluated in 32-bit integer math. (Floating point is not currently supported.)

Extended Microcontroller Mode (PIC17CXXX and PIC18CXXX 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 PIC17CXXX or PIC18CXXX device.

External Input Line (MPLAB-ICE only)

An external input signal logic probe line (TRIGIN) for setting an event based upon external signals.

External Linkage

A function or variable has external linkage if it can be accessed from outside the module in which it is defined.

External RAM (PIC17CXXX and PIC18CXXX Devices Only)

Off-chip Read/Write memory.

External Symbol

A symbol for an identifier which has external linkage.

External Symbol Definition

A symbol for a function or variable defined in the current module.

External Symbol Reference

A symbol which references a function or variable defined outside the current module.

External Symbol Resolution

A process performed by the linker in which external symbol definitions from all input modules are collected in an attempt to update all external symbol references. Any external symbol references which do not have a corresponding definition cause a linker error to be reported.

File Registers

On-chip general purpose and special function registers.

Flash

A type of EEPROM where data is written or erased in blocks instead of bytes.

FNOP

Forced No Operation. A forced `NOP` cycle is the second cycle of a two-cycle instruction. Since the PICmicro architecture is pipelined, it prefetches the next instruction in the physical address space while it is executing the current instruction. However, if the current instruction changes the program counter, this prefetched instruction is explicitly ignored, causing a forced `NOP` cycle.

GPR

General Purpose Register.

Halt

A function 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

Executable instructions assembled or compiled from source code into standard hexadecimal format code. Also called executable or machine code. Hex code is contained in a hex file.

Hex File

An ASCII file containing hexadecimal addresses and values (hex code) suitable for programming a device. This format is readable by a device programmer.

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 (such as C) employ a compiler to translate statements into machine instructions that the target processor can execute.

ICD

In-Circuit Debugger. MPLAB-ICD is Microchip's in-circuit debugger for PIC16F87X devices. MPLAB-ICD works with MPLAB IDE.

ICE

In-Circuit Emulator. MPLAB-ICE is Microchip's in-circuit emulator that works with MPLAB IDE. PICMASTER and ICEPIC (Third Party product) are other ICE devices.

IDE

Integrated Development Environment. An application that has multiple functions for firmware development. The MPLAB IDE integrates a compiler, an assembler, a project manager, an editor, a debugger, a simulator and an assortment of other tools within one Windows application. A user developing an application can write code, compile, debug and test an application without leaving the MPLAB IDE desktop.

Identifier

A function or variable name.

Import

Bring data into the MPLAB Integrated Development Environment (IDE) from an outside source, such as from a hex file.

Initialized Data

Data which is defined with an initial value. In C, `int myVar=5;` defines a variable which will reside in an initialized data section.

Internal Linkage

A function or variable has internal linkage if it can not be accessed from outside the module in which it is defined.

Librarian

A language tool that creates and manipulates libraries. MPLIB is Microchip's librarian.

Library

A library is a collection of relocatable object modules. It is created by assembling multiple source files to object files and then using the librarian to combine the object files into one library file. A library can be linked with object modules and other libraries to create executable code.

Link

The process of combining object files and libraries to create executable code. This operation is performed by a linker.

Linker

A language tool that combines object files and libraries to create executable code. Linking is performed by Microchip's linker, MPLINK.

Linker Script Files

Linker script files are the command files of MPLINK (.LKR). They define linker options and describe available memory on the target platform.

Listing Directives

Listing directives are those directives that control the MPASM listing file format. They allow the specification of titles, pagination and other listing control.

Listing File

A listing file is an ASCII text file that shows the machine code generated for each C source statement, assembly instruction, MPASM directive or macro encountered in a source file.

Local Label

A local label is one that is defined inside a macro with the `LOCAL` directive. These labels are particular to a given instance of a macro's instantiation. In other words, the symbols and labels that are declared as local are no longer accessible after the `ENDM` macro is encountered.

Logic Probes

Up to 14 logic probes can be connected to some Microchip emulators. The logic probes provide external trace inputs, trigger output signal, +5V and a common ground.

Machine Code

Either object or executable code.

Macro

A collection of assembler instructions that are included in the assembly code when the macro name is encountered in the source code. Macros must be defined before they are used; forward references to macros are not allowed.

All statements following a `MACRO` directive and prior to an `ENDM` directive are part of the macro definition. Labels used within the macro must be local to the macro so the macro can be called repetitively.

Macro Directives

Directives that control the execution and data allocation within macro body definitions.

Make Project

A command that rebuilds an application, re-compiling only those source files that have changed since the last complete compilation.

MCU

Microcontroller Unit. An abbreviation for microcontroller. Also µC.

Memory Models

Versions of libraries and/or precompiled object files based on a device’s memory (RAM/ROM) size and structure.

Microcontroller

A highly integrated chip that contains all the components comprising a controller. Typically this includes a CPU, RAM, some form of ROM, I/O ports and timers. Unlike a general-purpose computer, which also includes all of these components, a microcontroller is designed for a very specific task – to control a particular system. As a result, the parts can be simplified and reduced, which cuts down on production costs.

Microcontroller Mode (PIC17CXXX and PIC18CXXX Devices Only)

One of the possible program memory configurations of the PIC17CXXX and PIC18CXXX families of microcontrollers. In microcontroller mode, only internal execution is allowed. Thus, only the on-chip program memory is available in microcontroller mode.

Microprocessor Mode (PIC17CXXX and PIC18CXXX Devices Only)

One of the possible program memory configurations of the PIC17CXXX and PIC18CXXX families of microcontrollers. In microprocessor mode, the on-chip program memory is not used. The entire program memory is mapped externally.

Mnemonics

Instructions that are translated directly into machine code. Mnemonics are used to perform arithmetic and logical operations on data residing in program or data memory of a microcontroller. They can also move data in and out of registers and memory as well as change the flow of program execution. Also referred to as Opcodes.

MPASM

Microchip Technology’s relocatable macro assembler. MPASM is a DOS or Windows-based PC application that provides a platform for developing assembly language code for Microchip’s PICmicro microcontroller families. Generically, MPASM will refer to the entire development platform including the macro assembler and utility functions.

MPASM will translate source code into either object or executable code. The object code created by MPASM may be turned into executable code through the use of the MPLINK linker.

MPLAB-CXX

Refers to MPLAB-C17 and MPLAB-C18 C compilers.

MPLAB-ICD

Microchip's in-circuit debugger for PIC16F87X devices. MPLAB-ICD works with MPLAB IDE. The MPLAB-ICD system consists of a module, header, demo board (optional), cables and MPLAB Software.

MPLAB-ICE

Microchip's in-circuit emulator that works with MPLAB IDE.

MPLAB IDE

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.

MPLAB-SIM

Microchip's simulator that works with MPLAB IDE.

MPLIB

MPLIB is a librarian for use with COFF object modules (`filename.o`) created using either MPASM v2.0, MPASMWIN v2.0, or MPLAB-C v2.0 or later.

MPLIB will combine multiple object files into one library file. Then MPLIB can be used to manipulate the object files within the created library.

MPLINK

MPLINK is a linker for the Microchip relocatable assembler, MPASM and the Microchip C compilers, MPLAB-C17 or MPLAB-C18. MPLINK also may be used with the Microchip librarian, MPLIB. MPLINK is designed to be used with MPLAB IDE, though it does not have to be.

MPLINK will combine object files and libraries to create a single executable file.

MPSIM

The DOS version of Microchip's simulator. MPLAB-SIM is the newest simulator from Microchip.

MRU

Most Recently Used. Refers to files and windows available to be selected from MPLAB IDE main pull down menus.

Nesting Depth

The maximum level to which macros can include other macros.

Non Real-Time

Refers to the processor at a breakpoint or executing single step instructions or MPLAB IDE being run in simulator mode.

Node

MPLAB IDE project component.

`NOP`

No Operation. An instruction that has no effect when executed except to advance the program counter.

Object Code

The intermediate code that is produced from the source code after it is processed by an assembler or compiler. Relocatable code is code produced by MPASM or MPLAB-C17/C18 that can be run through MPLINK to create executable code. Object code is contained in an object file.

Object File

A module which may contain relocatable code or data and references to external code or data. Typically, multiple object modules are linked to form a single executable output. Special directives are required in the source code when generating an object file. The object file contains object code.

Object File Directives

Directives that are used only when creating an object file.

Off-Chip Memory (PIC17CXXX and PIC18CXXX Devices Only)

Off-chip memory refers to the memory selection option for the PIC17CXXX or PIC18CXXX device where memory may reside on the target board, or where all program memory may be supplied by the Emulator. The Memory tab accessed from *Options > Development Mode* provides the Off-Chip Memory selection dialog box.

Opcodes

Operational Codes. See Mnemonics.

Operators

Arithmetic symbols, like the plus sign '+' and the minus sign '-', that are used when forming well-defined expressions. Each operator has an assigned precedence.

OTP

One Time Programmable. EPROM devices that are not in windowed packages. Since EPROM needs ultraviolet light to erase its memory, only windowed devices are erasable.

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, the event is satisfied. You can assign the Pass Counter to break and trace logic, and to any sequential event in the complex trigger dialog.

PC

Personal Computer or Program Counter.

PC Host

Any IBM® or compatible Personal Computer running Windows 3.1x or Windows 95/98, Windows NT, or Windows 2000.

PICmicro MCUs

PICmicro microcontrollers (MCUs) refers to all Microchip microcontroller families.

PICSTART Plus

A device programmer from Microchip. Programs 8-, 14-, 28- and 40-pin PICmicro microcontrollers. Must be used with MPLAB Software.

Pod

The external emulator box that contains emulation memory, trace memory, event and cycle timers, and trace/breakpoint logic.

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.

Precedence

The concept that some elements of an expression are evaluated before others, (i.e., * and / before + and -). In MPASM, operators of the same precedence are evaluated from left to right. Use parentheses to alter the order of evaluation.

Program Counter

A register that specifies the current execution address.

Program Memory

The memory area in a PICmicro microcontroller where instructions are stored. Memory in the emulator or simulator containing the downloaded target application firmware.

Programmer

A device used to program electrically programmable semiconductor devices such as microcontrollers.

Project

A set of source files and instructions to build the object and executable code for an application.

PRO MATE

A device programmer from Microchip. Programs all PICmicro microcontrollers and most memory and KEELOQ® devices. Can be used with MPLAB IDE or stand-alone.

Prototype System

A term referring to a user's target application or target board.

PWM Signals

Pulse Width Modulation Signals. Certain PICmicro devices have a PWM peripheral.

Qualifier

An address or an address range used by the Pass Counter or as an event before another operation in a complex trigger.

Radix

The number base, hex, or decimal, used in specifying an address and for entering data in the *Window > Modify* command.

RAM

Random Access Memory (Data Memory).

Raw Data

The binary representation of code or data associated with a section.

Real-Time

When released from the halt state in the emulator or MPLAB-ICD mode, the processor runs in real-time mode and behaves exactly as the normal chip would behave. In real-time mode, the real-time trace buffer of MPLAB-ICE is enabled and constantly captures all selected cycles and all break logic is enabled. In the emulator or MPLAB-ICD, 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.

Recursion

The concept that a function or macro, having been defined, can call itself. Great care should be taken when writing recursive macros; it is easy to get caught in an infinite loop where there will be no exit from the recursion.

Relocatable Section

A section whose address is not fixed (absolute). The linker assigns addresses to relocatable sections through a process called relocation.

Relocation

A process performed by the linker in which absolute addresses are assigned to relocatable sections and all identifier symbol definitions within the relocatable sections are updated to their new addresses.

ROM

Read Only Memory (Program Memory).

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.

Section

An portion of code or data which has a name, size and address.

SFR

Special Function Registers of a PICmicro device.

Shared Section

A section which resides in a shared (non-banked) region of data RAM.

Shell

The MPASM shell is a prompted input interface to the macro assembler. There are two MPASM shells: one for the DOS version and one for the Windows version.

Simulator

A software program that models the operation of the PICmicro microprocessor.

Single Step

This command steps through code, one instruction at a time. After each instruction, MPLAB IDE 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 IDE will execute all assembly level instructions generated by the line of the high level C statement.

Skew

The information associated with the execution of an instruction appears on the processor bus at different times. For example, the executed Opcodes appears on the bus as a fetch during the execution of the previous instruction, the source data address and value and the destination data address appear when the Opcodes is actually executed and the destination data value appears when the next instruction is executed. The trace buffer captures the information that is on the bus at one instance. Therefore, one trace buffer entry will contain execution information for three instructions. The number of captured cycles from one piece of information to another for a single instruction execution is referred to as the skew.

Skid

When a hardware breakpoint is used to halt the processor, one or more additional instructions may be executed before the processor halts. The number of extra instructions executed after the intended breakpoint is referred to as the skid.

Source Code - Assembly

Source code consists of PICmicro instructions and MPASM directives and macros that will be translated into machine code by an assembler.

Source Code - C

A program written in the high level language called "C" which will be converted into PICmicro machine code by a compiler. Machine code is suitable for use by a PICmicro MCU or Microchip development system product like MPLAB IDE.

Source File - Assembly

The ASCII text file of PICmicro instructions and MPASM directives and macros (source code) that will be translated into machine code by an assembler. It is an ASCII file that can be created using any ASCII text editor.

Source File - C

The ASCII text file containing C source code that will be translated into machine code by a compiler. It is an ASCII file that can be created using any ASCII text editor.

Special Function Registers

Registers that control I/O processor functions, I/O status, timers, or other modes or peripherals.

Stack - Hardware

An area in PICmicro MCU memory where function arguments, return values, local variables and return addresses are stored, (i.e., a "Push-Down" list of calling routines). Each time a PICmicro MCU executes a `CALL` or responds to an interrupt, the software pushes the return address to the stack. A return command pops the address from the stack and puts it in the program counter.

The PIC18CXXX family also has a hardware stack to store register values for "fast" interrupts.

Stack - Software

The compiler uses a software stack for storing local variables and for passing arguments to and returning values from functions.

Static RAM or SRAM

Static Random Access Memory. Program memory you can Read/Write on the target board that does not need refreshing frequently.

Status Bar

The Status Bar is located on the bottom of the MPLAB IDE window and indicates such current information as cursor position, development mode and device, and active tool bar.

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 the same as Single Step except for its handling of CALL instructions.

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. Stimulus may be asynchronous, synchronous (pin), clocked and register.

Stopwatch

A counter for measuring execution cycles.

Symbol

A symbol is a general purpose mechanism for describing the various pieces which comprise a program. These pieces include function names, variable names, section names, file names, struct/enum/union tag names, etc.

Symbols in MPLAB IDE refer mainly to variable names, function names and assembly labels.

System Button

The system button is another name for the system window control. Clicking on the system button pops up the system menu.

System Window Control

The system window control is located in the upper left corner of windows and some dialogs. Clicking on this control usually pops up a menu that has the items “Minimize,” “Maximize,” and “Close.” In some MPLAB IDE windows, additional modes or functions can be found.

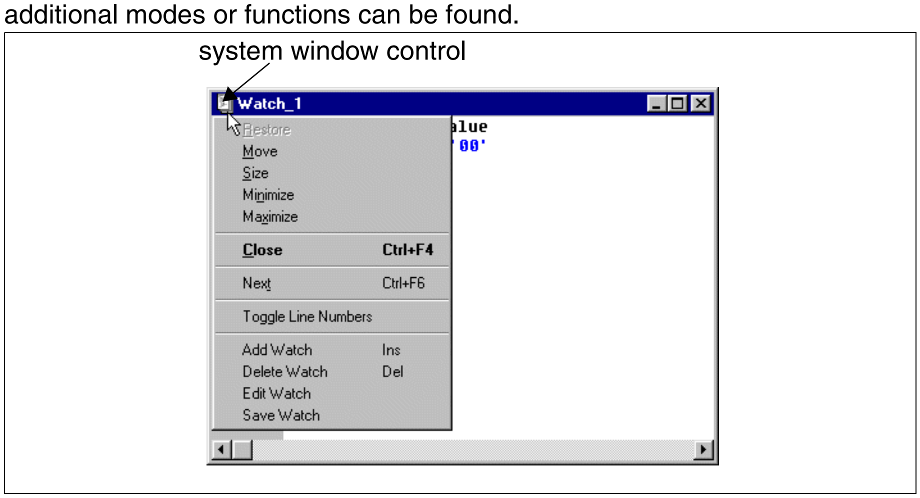


Figure G1: System Window Control Menu - Watch Window

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 IDE functions.

Trace

An emulator or simulator function that logs program execution. The emulator logs program execution into its trace buffer which is uploaded to MPLAB IDE's trace window.

Trace Memory

Trace memory contained within the emulator. Trace memory is sometimes called the trace buffer.

Trigger Output

Trigger output refers to an emulator output signal that can be generated at any address or address range and is independent of the trace and breakpoint settings. Any number of trigger output points can be set.

Unassigned Section

A section which has not been assigned to a specific target memory block in the linker command file. The linker must find a target memory block in which to allocate an unassigned section.

Uninitialized Data

Data which is defined without an initial value. In C, `int myVar;` defines a variable which will reside in an uninitialized data section.

Upload

The Upload function transfers data from a tool, such as an emulator or programmer, to the host PC or from the target board to the emulator.

Warning

An alert that is provided to warn you of a situation that would cause physical damage to a device, software file, or equipment.

Watchdog Timer (WDT)

A timer on a PICmicro microcontroller that resets the processor after a selectable length of time. The WDT is enabled or disabled and set up using configuration bits.

Watch Variable

A variable that you may monitor during a debugging session in a watch window.

Watch Window

Watch windows contain a list of watch variables that are updated at each breakpoint.

NOTES:

Index

NOTES:

MICROCHIP

WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-786-7200 Fax: 480-786-7277
Technical Support: 480-786-7627
Web Address: http://www.microchip.com

Rocky Mountain
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-786-7966 Fax: 480-786-7456

Atlanta
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 770-640-0034 Fax: 770-640-0307

Boston
2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3838 Fax: 978-692-3821

Chicago
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071 Fax: 630-285-0075

Dallas
4570 Westgrove Drive, Suite 160
Addison, TX 75001
Tel: 972-818-7423 Fax: 972-818-2924

Dayton
Two Prestige Place, Suite 130
Miamisburg, OH 45342
Tel: 937-291-1654 Fax: 937-291-9175

Detroit
Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250 Fax: 248-538-2260

Los Angeles
18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888 Fax: 949-263-1338

New York
150 Motor Parkway, Suite 202
Hauppauge, NY 11788
Tel: 631-273-5305 Fax: 631-273-5335

San Jose
Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408-436-7950 Fax: 408-436-7955

Toronto
6285 Northam Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel: 905-673-0699 Fax: 905-673-6509

ASIA/PACIFIC

China - Beijing
Microchip Technology Beijing Office
Unit 915
New China Hong Kong Manhattan Bldg.
No. 6 Chaoyangmen Beidajie
Beijing, 100027, No. China
Tel: 86-10-85282100 Fax: 86-10-85282104

China - Shanghai
Microchip Technology Shanghai Office
Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xia Road
Shanghai, 200051
Tel: 86-21-6275-5700 Fax: 86-21-6275-5060

Hong Kong
Microchip Asia Pacific
RM 2101, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200 Fax: 852-2401-3431

India
Microchip Technology Inc.
India Liaison Office
Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O'Shaugnessey Road
Bangalore, 560 027, India
Tel: 91-80-207-2165 Fax: 91-80-207-2171

Japan
Microchip Technology Intl. Inc.
Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa, 222-0033, Japan
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

Korea
Microchip Technology Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea
Tel: 82-2-554-7200 Fax: 82-2-558-5934

ASIA/PACIFIC (continued)

Singapore
Microchip Technology Singapore Pte Ltd.
200 Middle Road
#07-02 Prime Centre
Singapore, 188980
Tel: 65-334-8870 Fax: 65-334-8850

Taiwan
Microchip Technology Taiwan
11F-3, No. 207
Tung Hua North Road
Taipei, 105, Taiwan
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

EUROPE

Denmark
Microchip Technology Denmark ApS
Regus Business Centre
Lautrup hoj 1-3
Ballerup DK-2750 Denmark
Tel: 45 4420 9895 Fax: 45 4420 9910

France
Arizona Microchip Technology SARL
Parc d'Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - ler Etage
91300 Massy, France
Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

Germany
Arizona Microchip Technology GmbH
Gustav-Heinemann Ring 125
D-81739 Munich, Germany
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

Italy
Arizona Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Taurus 1 V. Le Colleoni 1
20041 Agrate Brianza
Milan, Italy
Tel: 39-039-65791-1 Fax: 39-039-6899883

United Kingdom
Arizona Microchip Technology Ltd.
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44 118 921 5869 Fax: 44-118 921-5820

9/01/00

DNV Certification, Inc.
USA

DNV MSC
The Netherlands
Accredited by the RvA

ANSI*RAB QMS ACCREDITED

DNV

ISO 9001 / QS-9000
REGISTERED FIRM

Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs and microperipheral products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.