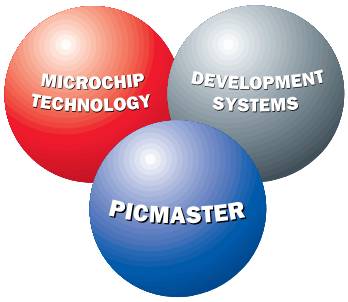
# PICMASTER® EMULATOR

## User's Guide



# PICMASTER EMULATOR USER'S GUIDE

# PICMASTER Emulator Preview

# A Quick Look at the PICMASTER® Emulator

## What is the PICMASTER Emulator

The PICMASTER Emulator is a hardware debugging tool that operates with the MPLAB™ Integrated Development Environment. MPLAB has software tools to debug source code and MPLAB-SIM to simulate PIC16/17 instructions and simple I/O. The PICMASTER emulator is one of the MPLAB tools. MPLAB with the PICMASTER emulator can debug application hardware as it executes in real-time.

An emulator, also called an in-circuit emulator or ICE, differs from a simulator in these areas:

- I/O — The emulator has I/O and on-chip devices that function exactly like the target processor. In comparison, a simulator provides limited I/O and is a limited model of the special functions of a microcontroller,
- Execution Speed — The emulator runs code and responds to external and internal events at the same speed the application runs.
- Real Time Analyzer — The emulator watches code execution and monitors external signals as the application runs. It also triggers other electronic devices such as oscilloscopes and high speed timing analyzers to help pinpoint timing problems.

The PICMASTER emulator connects into application hardware and runs like the PIC16/17 microcontroller in your design. The emulator probe plugs into the PIC16/17 socket on your application, and runs your code from PICMASTER emulation memory.

## How the PICMASTER Emulator Helps You

The PICMASTER emulator allows you to:

- Debug your source code on your own hardware.
- Debug your firmware in real time.
- Debug with hardware breakpoints and breakpoints based on internal register values.
- Watch the program flow in real time.
- Make timing measurements.
- Set breakpoints based upon external signals.
- Capture and trigger external events as your firmware executes at full speed.

# MPLAB Integrated Development Environment

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

This document covers the basic set up and operation of the PICMASTER emulator, but it does not cover all functions of the MPLAB Integrated Development Environment (IDE). Read the MPLAB User's Guide to get a full understanding of the features and debug capabilities of the MPLAB Integrated Development Environment.

## MPLAB Development Tools

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

The MPLAB Integrated Development Environment includes the following development tools:

- **MPLAB Project Manager**

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

- **MPLAB Editor**

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

- **MPLAB-SIM Simulator**

  The simulator models instruction execution of PIC16/17 microcontrollers and PIC16/17 I/O at the speed of your CPU.

- **PICMASTER Emulator**

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

- **MPASM Universal PIC16/17 Microcontroller Assembler**

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

# Chapter 1. About PICMASTER Emulator

## Introduction

This chapter describes the PICMASTER emulator, lists its primary features, and suggests recommended reading.

## Highlights

This chapter covers the following topics:

- **Emulator Components, Hardware, and Architecture**
- **Hardware and Software Requirements**
- **About this Guide**
- **Recommended Reading**
- **Warranty Registration**
- **Customer Support**

## Emulator Components, Hardware, and Architecture

The PICMASTER emulator provides product development engineers with a complete microcontroller design tool set for all microcontrollers in the PIC16/17 microcontroller families.

### Emulator System Components

The four major components of the emulator System are:

- Emulator Pod
- PC Interface Card
- Probe
- MPLAB Integrated Development Environment (IDE)

#### Emulator Pod

The emulator contains all emulation and control logic common to all microcontroller devices. The emulator contains emulation memory, trace memory, event and cycle timers, and trace/breakpoint logic. The emulator controls and interfaces to an interchangeable probe via a 14 inch ribbon cable.

### PC Interface Card

The PC interface card connects the emulator system to a compatible PC host. This high-speed parallel interface requires a single half-size standard AT®/ISA slot in the PC host. A 37-conductor cable connects the PC interface card to the emulator pod.

### Probe

The emulator requires an interface connection to the target microcontroller device that you are emulating. A probe specific to the microcontroller family that you are emulating provides this interface connection. The installed probe configures the emulator system for emulating a target microcontroller, and connects to the ribbon cable coming from the emulator.

Interchangeable probes allow development engineers to easily reconfigure the emulator system for emulating different target processors. The probe operating frequencies may range up to 25 MHz, depending on the probe and emulator pod.

Supporting probes for future microcontroller devices will be made available as Microchip releases future devices.

### MPLAB Software

The MPLAB software runs in the Windows® 3.1 or greater environment, and provides full display, modification, and control of the target application under emulation. MPLAB software supports the PIC16C/17 microcontroller families. The MPLAB software automatically identifies the probe type, and configures the emulator system for emulating the target processor.

With MPLAB and MS Windows you can concurrently run many programs such as inter-office mail, spreadsheets, and PICMASTER. Since MPLAB supports DDE (Dynamic Data Exchange) with client programs, emulation data collected with the MPLAB system can be shared with programs such as Microsoft Excel.

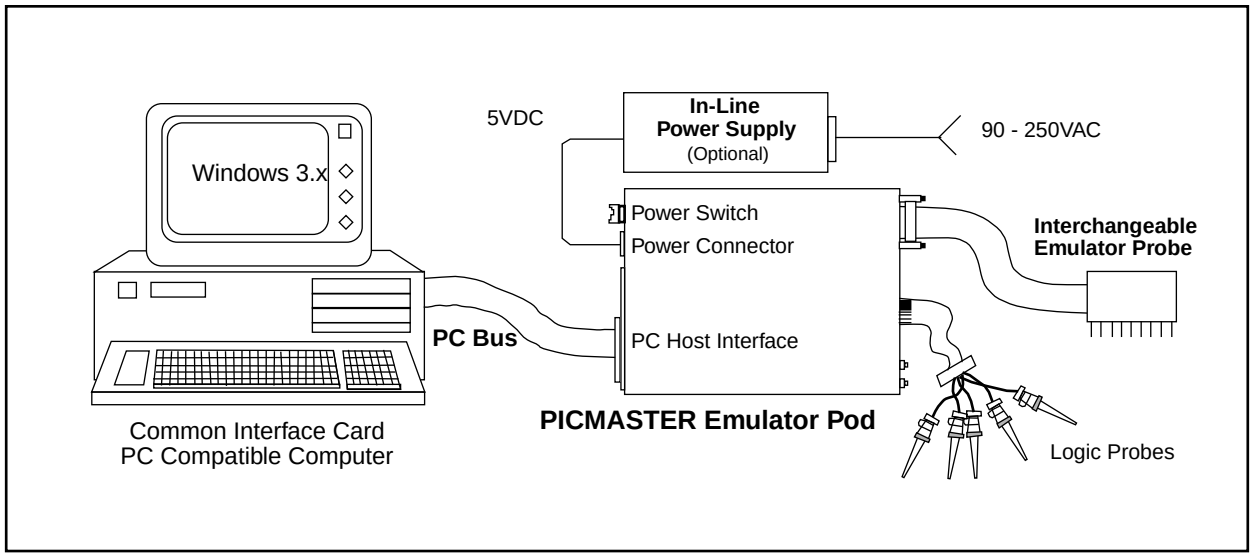



**Figure 1.1 Block Diagram of Emulator System**

## Emulator Architecture

The universal architecture of the emulator allows expansion to support all new PIC16/17 microcontrollers. Data and program memory paths extend to 16 bits.

Under MPLAB, up to four emulators can run simultaneously under the same PC host, allowing engineers to develop multi-microcontroller systems. For example, a target application might contain a PIC16CXX processor, and a PIC17CXX processor. MPLAB software allows users to synchronize breakpoints and instruction stepping between multiple emulators.

# Hardware and Software Requirements

The emulator system is designed to operate on most PC compatible 386 or better class systems running MS Windows 3.1 or Windows 95. This real-time emulation system provides advanced features generally found on more expensive development tools.

## Host Computer System Requirements

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

- PC compatible machine: 386 or higher.
- VGA required. Super VGA recommended.
- MS-DOS/PC-DOS version 5.0 or greater.
- Microsoft Windows version 3.1 or greater operating in 386 enhanced mode.

- 4 MB of Memory, 16 MB Recommended
- 8 MB of Hard Disk Space, 20 MB Recommended
- One 8-bit PC/AT® (ISA) I/O expansion slot (half size)
- Mouse or other pointing device

### Emulator Electrical Specifications

For the complete electrical specifications for the PICMASTER Pod, refer to the PICMASTER Pod Electrical Specifications (Document Number DS51021).

For information on a specific probe, refer to the probe specification for the applicable probe.

# About this Guide

This document describes how to use the emulator as a development tool to emulate and debug firmware on a target board. (For a detailed discussion about basic Microchip microcontroller devices, refer to the data sheets for the specific microcontrollers).

### Documentation Layout

The manual layout is as follows:

**PICMASTER Emulator Preview** - This preview describes how the PICMASTER emulator works with MPLAB, and describes how the emulator helps debug applications.

**Chapter 1: Introduction -** This chapter describes the PICMASTER emulator, lists its primary features, and suggests recommended reading.

**Chapter 2: PICMASTER Emulator Installation -** This chapter describes the procedures for installing the PICMASTER Emulator.

**Chapter 3: PICMASTER Emulator Basic Functions -** This chapter briefly discusses the basic MPLAB debugging functions of the PICMASTER emulator.

**Chapter 4: PICMASTER Emulator Menu Options -** This chapter gives detailed information on using the MPLAB menu options directly applicable to the PICMASTER emulator.

**Chapter 5: Migrating from PICMASTER 1.xx Software -** This chapter describes the procedures for migrating from PICMASTER 1.xx software.

**Appendix A: Troubleshooting Guide -** The troubleshooting information in this appendix can help you with typical problems you may have in using the emulator, and describes how to resolve these problems.

**Appendix B: On Line Support -** Information on Microchip's electronic support services.

**Appendix C: PICMASTER Logic Probes -** This appendix gives the pin data for the external logic probes that connect to the PICMASTER emulator.

**Index -** The Index provides a quick reference to PICMASTER emulator functions and features discussed in this manual.

**Worldwide Sales and Service -** This reference gives the address, telephone and fax number for Microchip Technology Inc. sales and service locations throughout the world.

## Conventions Used in this Guide

This manual uses the following documentation conventions:

**Table 1.2 Documentation Conventions**

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

# Recommended Reading

**README.EM** For the latest information on using PICMASTER, read the README.EM file (an ASCII text file) on the PICMASTER diskette. README.EM contains update information that may not be included in the *PICMASTER User's Guide*.

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

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

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

**Microchip ECHB Update I** Contains additional application notes released since publication of the standard Embedded Control Handbook.

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

**PICMASTER Pod Electrical Specifications** Contains PICMASTER Pod electrical specifications. *Document Number DS51021, Microchip Technology, Chandler, AZ.*

**Probe Specifications** The specifications for each Microchip PICMASTER emulator probe are available from *Microchip Technology, Chandler, AZ.*

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

This manual assumes that you are familiar with both MS-DOS® and Microsoft Windows 3.x software systems. Many excellent references exist for both of these software programs, and should be consulted for general operation of DOS and Windows.

# Warranty Registration

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

# Customer Support

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

The Microchip Internet Home Page can provide you with technical information, application notes, and promotional news on Microchip products and technology. The Microchip Web address is http://www.microchip.com

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

# Chapter 2. PICMASTER Emulator Installation

## Introduction

This chapter describes the procedures for installing the PICMASTER emulator.

## Highlights

The items discussed in this chapter include:

- **Unpacking the System**
- **Installing PICMASTER Emulator Hardware**
- **Installing Software**
- **Getting the Most From MPLAB – Using Projects**
- **Where to go from Here**

## Unpacking the System

### Packing List

The PICMASTER emulator system includes several items:

- An emulator pod and a parallel cable to connect it to the PC interface card
- A PC interface card
- A power supply
- A probe and cable assembly to connect to the emulator
- Disks with complete system software and documentation
- A warranty registration card
- Manuals - PICMASTER Emulator User's Guide (this document), MPLAB User's Guide, and MPASM User's Guide.

# Installing PICMASTER Emulator Hardware

The following steps should be followed to install your PICMASTER Emulator hardware. **If you already have installed the PICMASTER Emulator system then skip to the next section, "Installing Software" (page 14).**

Steps for Installing Hardware

**1. Select I/O Address**

Locate the half-height PC Interface Card shown in Figure 2.1 and make your I/O address selection by setting the four DIP switches on SW1. The I/O address configuration DIP switch (SW1) is located on the top left corner of the PC Interface Card. The factory default setting is 300H. You may use another address setting if it does not conflict with other device cards in your system.

**Table 2.1 Base Address Corresponding to DIP Switch Settings**

| Base Address | DIP Switch Setting | | | |
|---|---|---|---|---|
| | 1 | 2 | 3 | 4 |
| 200 | ON | ON | ON | ON |
| 208 | ON | OFF | ON | ON |
| 210 | ON | ON | OFF | ON |
| 218 | ON | OFF | OFF | ON |
| 300 | OFF | ON | ON | OFF |
| 308 | OFF | OFF | ON | OFF |
| 310 | OFF | ON | OFF | OFF |
| 318 | OFF | OFF | OFF | OFF |

**Caution:** Disconnect power from host computer before proceeding.

**2. Install PC Interface Card**

Ground yourself to either the PC's power supply or the case (be sure you have disconnected the power from host computer). Open the computer enclosure and install the PC Interface card in any available slot. Secure the card edge connector to the chassis.

**Note:** The interface cable is oversized and may dislodge the card during use if the card is not secured.

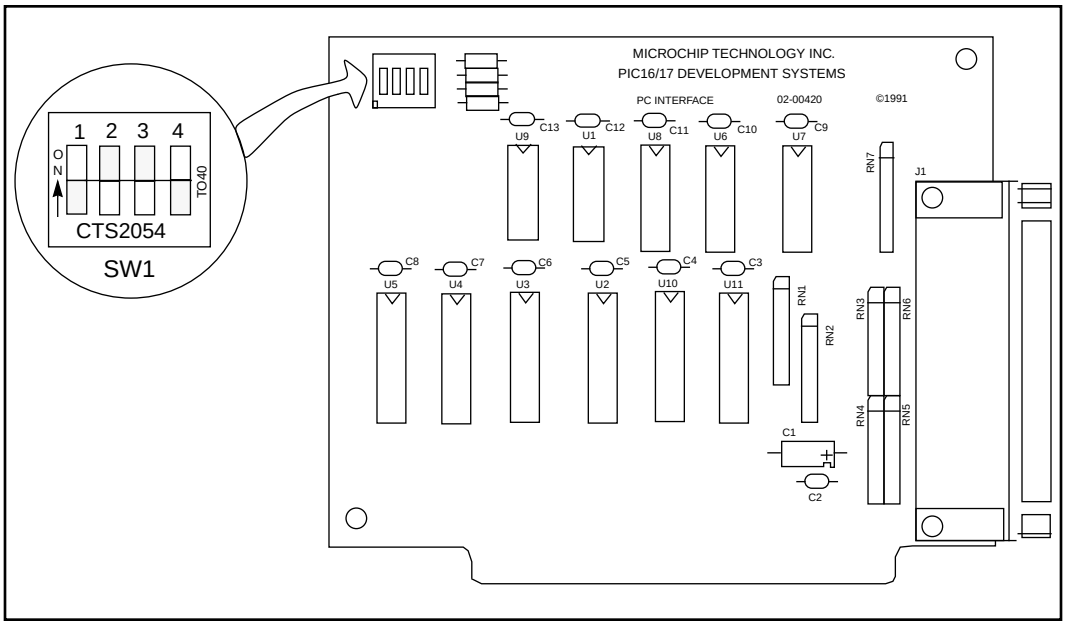



**Figure 2.1 PC Interface Card**

**3. Connect 37-Pin Cable (PARALLEL BUS)**

Connect one end of the 37-pin cable to the connector on the PC Interface Card. Secure the connector with mounting screws. Connect the other end of the cable to the 37-pin female PARALLEL BUS connector on the emulator pod and secure the connector with cable screws.

**Note:** The RS-232 port is not supported. This port is adjacent to the PARALLEL BUS connector on the pod.

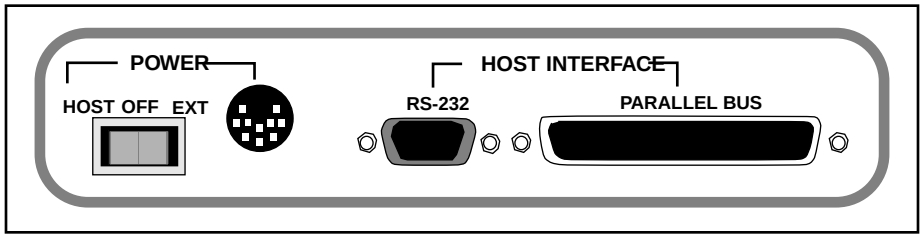



**Figure 2.2 PICMASTER Pod Rear Panel**

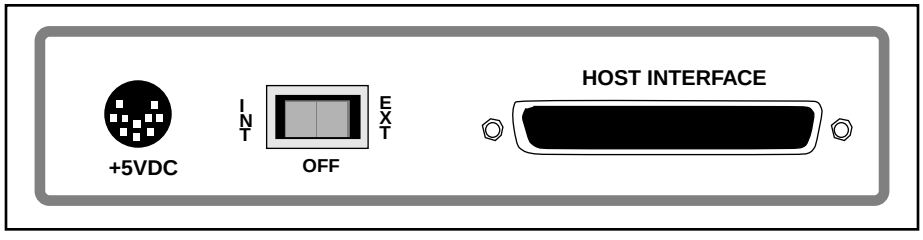



**Figure 2.3 PICMASTER CE Pod Rear Panel**

**4. Assemble the Probe**

Carefully attach a daughter board to the probe according to the probe specifications. See the *PICMASTER Probe Specifications* guide included with each probe kit for specifications.

**5. Configure the Target System Clock**

Configure the jumpers on the probe for the internal or external clock depending on your application. Refer to the probe data sheet for more information.

**6. Select INT/EXT Power to Probe**

You can configure power to the probe to come from the emulator, or from the target system.

Configure the jumpers on the probe for internal or external power depending on your application. Refer to the probe data sheet for more information.

> **Note:** When using the external clock option (where the target application board provides clock the signal), the external clock must always be present for the emulator system to function properly.

**7. Connect Ribbon Cable**

Continue assembling your system by connecting one end of the 14" flat cable to the emulator at the connector marked EMULATOR PROBE. Connect the other end of this cable to the Emulator Probe Board assembled in Step 4 above.

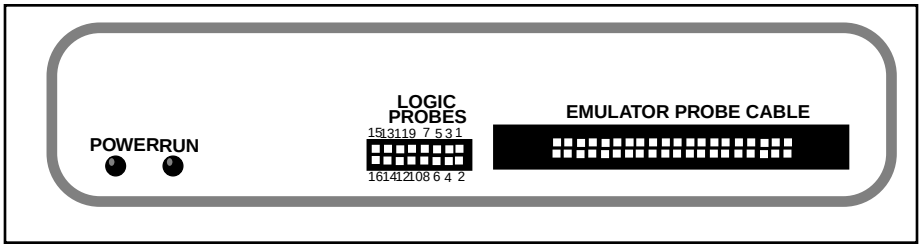



**Figure 2.4 PICMASTER Pod Front Panel**

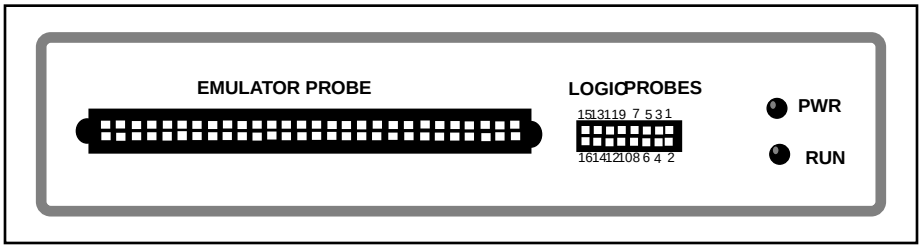



**Figure 2.5 PICMASTER CE Pod Front Panel**

8. **Connect Logic Probes**

   If you will use the external logic probes, insert them into the pod in the connector marked LOGIC PROBES (See PICMASTER Pod Electrical Specifications, Appendix D, for additional electrical specifications).

9. **Connect Probe Board to Target Board**

   Plug the probe board into the microcontroller socket on the target system with the provided extender socket or extender cable.

The emulator system is now assembled and ready for power up.

## Applying Power to System Components

Select a power source for the emulator. The PC interface board can power the emulator system directly. If your PC power supply is heavily loaded, the green power light on the PICMASTER emulator will not come on. Should this happen, use the external power supply provided. If you use the external power supply, insert the circular DIN plug into the receptacle marked POWER. Ensure that the three-position rocker power switch is in the center OFF position.

Power up the system components in the following sequence to prevent damage to any of the sub-system parts or user target application parts:

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

1. **Apply power to the PC host.**
2. **Apply power to the emulator assembly.**

   If power is to come from the PC, press the power rocker switch to the HOST position (left). If power is to come from the external power supply, press the rocker switch to the EXT position (right).

3. **Apply power to the target application circuit.**

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

## Power Down System Components

1. **Remove power from user target system**
2. **Remove power from emulator system.**
3. **Remove power from Host PC.**

# Installing Software

The MPLAB Setup routine installs the Microchip MPLAB Integrated Development Environment.

To install MPLAB, follow these steps:

1. Enter Microsoft Windows and insert the MPLAB installation disk 1 into drive A.
2. From the File Manager, or the *Program Manager > Run* option, run **a:setup**.

   The MPLAB Setup screen opens with the options **OK**, or **Cancel**.
3. Click **OK** to Install the PICMASTER emulator DLL's, or
   Click **Cancel** to abort installing the emulator DLL's.

## Communicating with the PICMASTER Hardware

If you are installing a new PICMASTER system and have not changed the I/O Port setting on the PC Interface card, MPLAB should connect automatically to the emulator hardware. However, if you are using an existing system or you have changed the I/O Port setting, use the following steps to set up the MPLAB software for the PICMASTER Hardware:

1. Make sure the emulator is connected to the PC with the parallel cable. Make sure there is a probe connected to the emulator and turn on emulator power.
2. From the Microchip MPLAB program group, run MPLAB.
3. MPLAB will either find the emulator and start, or it will display a message that it cannot find the emulator. If it finds the emulator you may skip to Chapter 5, "Migrating from PICMASTER 1.xx Software". If the emulator is not found, continue with step 4.
4. Select the **Editor Only** option from the dialog. MPLAB will start without attempting to connect to the emulator.
5. Select the *Options > Development Mode* menu item to open the Development Mode dialog.
6. Select PICMASTER Emulator.

   When you select an address, the software will indicate if an I/O card exists at the selected address. If an I/O card does not exist at an address, select a different I/O port.
7. Click *Run > Reset*.

# Getting The Most From MPLAB – Using Projects

MPLAB is the host software for the PICMASTER emulator and the MPLAB-SIM simulator. It functions as a sophisticated debugging tool, providing access to RAM, ROM, EEPROM, and a variety of other debug functions. To realize the true capabilities of MPLAB, you must use it as a project manager as well.

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

Perform the following steps to gain some familiarity with the MPLAB projects:

1. *Select Project > New Project* to create a new project. The project manager will create a project file of the same name as the file you choose, except it will have the extension *.PJT.
2. Add a file to the project. When you close the New Project dialog the Edit Project dialog opens automatically. Add the files you want to this project and close the dialog. Since the linker is not currently available, MPLAB only supports one source file per project. However, that file can include other files.
3. Select *Project > Build All*. MPLAB rebuilds all of the files in the project. If you select *Project > Make Project*, MPLAB only rebuilds the files that have changed. If there are any errors in the source code, MPLAB displays the error file and allows you to go immediately to the file and line where the error occurred by double-clicking on the error line.
4. Open the source code using *File > Open Source*. Select the *Window > Program Memory*, *Window > File Registers*, and *Window > Special Function Registers* to open the corresponding windows.
5. Select *Project > Close Project*. MPLAB records the open windows and their positions when you select Yes from the close project message box.
6. Select *Project > Open Project*. MPLAB reopens all of the project windows, reloads the binary code into the emulator, and loads the symbol information for debugging.
7. Using Run, Halt, Step, and Reset from the toolbar or the *Debug > Run* menu, you can debug your application from your source code. If you discover a bug in your source code, you can correct it immediately, recompile and reload the code into the emulator with a single keystroke (*Project > Make Project* or **F10**).

# Where To Go From Here

For all users, we recommend you work through the tutorial on projects in the MPLAB User's Guide. This tutorial gives you an overview of the debugging capabilities of the MPLAB system.

For users of PICMASTER 1.xx software or the DOS based MPSIM, we recommend you create a project based upon your latest embedded software project, and explore the capabilities of MPLAB.

# Chapter 3. PICMASTER Emulator Basic Functions

## Introduction

This chapter briefly discusses the basic MPLAB debugging functions of the PICMASTER emulator. For more information on general debugging features, refer to the MPLAB User's Guide.

## Highlights

This chapter covers the following information:

- **Program Execution**
- **Break, Trace, and Trigger Points**
- **Hardware Dialog**
- **Conditional Break Dialog**
- **Program Memory in the PICMASTER Emulator**
- **Multi-Processor Support**

## Program Execution

The emulator executes in real-time mode or in polled mode.

- Real time execution occurs when you run the processor without interruptions until encountering a valid breakpoint, or until encountering a user halt.
- Polled execution occurs when you single step the processor, modify values at a breakpoint, execute an opcode from the *Debug > Execute > Execute an Opcode* dialog, or use the *Debug > Execute > Conditional Break* dialog.

### Real Time Execution

When you run the emulator in real time, instructions execute just as the processor would without emulation. The emulator executes in real time until a valid breakpoint halts the emulator or until you force the emulator to halt.

To execute the emulator in real time, click the **Run** icon on the Tool Bar, or double click on an instruction in the Program Memory window.

| | |
|---|---|
| **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. |

When running in real time:

- The emulator behaves exactly as the normal chip would behave in real time.
- If break logic is enabled the emulator or simulator will halt upon encountering a valid breakpoint.
- All of the trace and trigger logic is enabled.

The Debug Tool Bar provides Run, Halt, and Step buttons for controlling the emulator or simulator. For example note Figure 3.1.

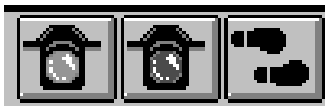


**Figure 3.1 Tool Bar Icons: Run, Halt, Step**

## Polled Execution

- Polled execution occurs when you single step the processor, modify values at a breakpoint, execute an opcode from the *Debug > Execute > Execute an Opcode* dialog, or use the *Debug > Execute > Conditional Break* dialog.

Polled execution provides the capability to:

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

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

## Animate Mode

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

# Break, Trace, and Trigger Points

PICMASTER allows the user to define events based upon the following elements:

- Breakpoints
- Trace Points

- Trigger Output Points
- Pass Counter Addresses

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

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

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

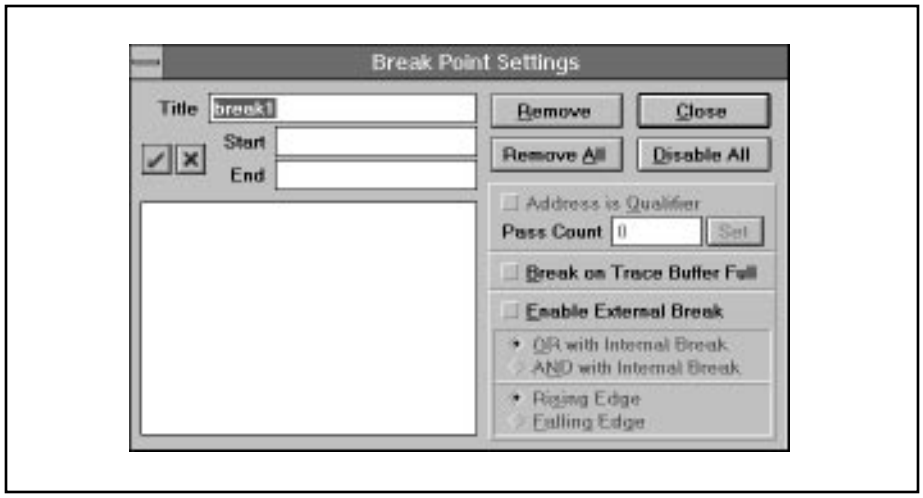



**Figure 3.2 Break Settings Dialog Box**

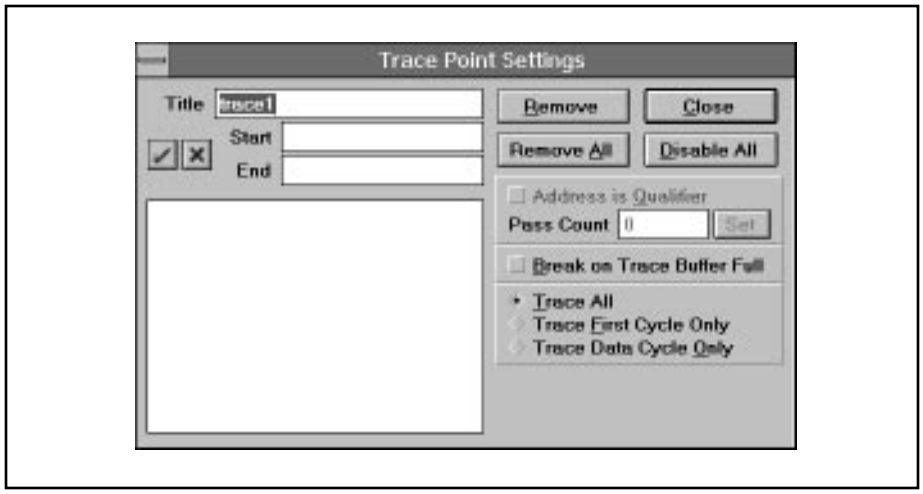



**Figure 3.3 Trace Settings Dialog Box**

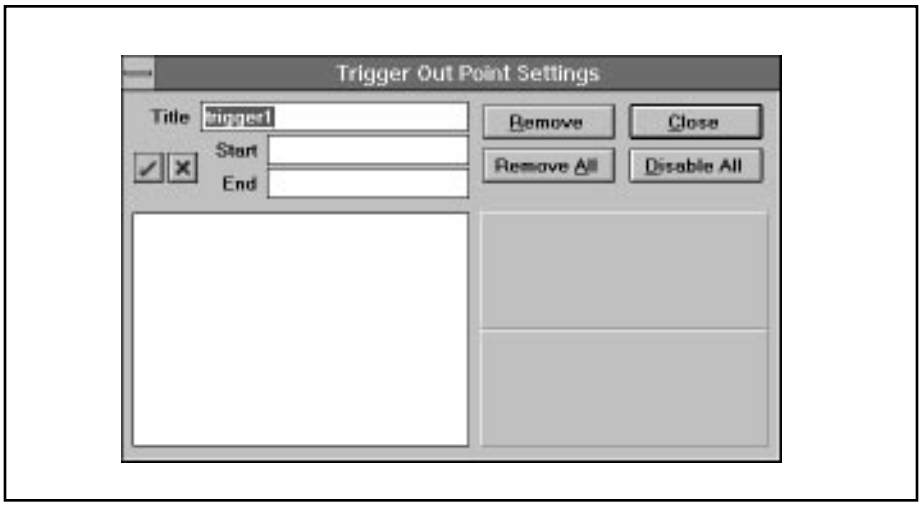



**Figure 3.4 Trigger Out Settings Dialog Box**

## Events in Program Memory Window

MPLAB displays the break, trace, and trigger out events and pass counter addresses as shown in Figure 3.5.

B Breakpoints

T Trace Points

O Trigger Output Points

Q Pass Counter Addresses

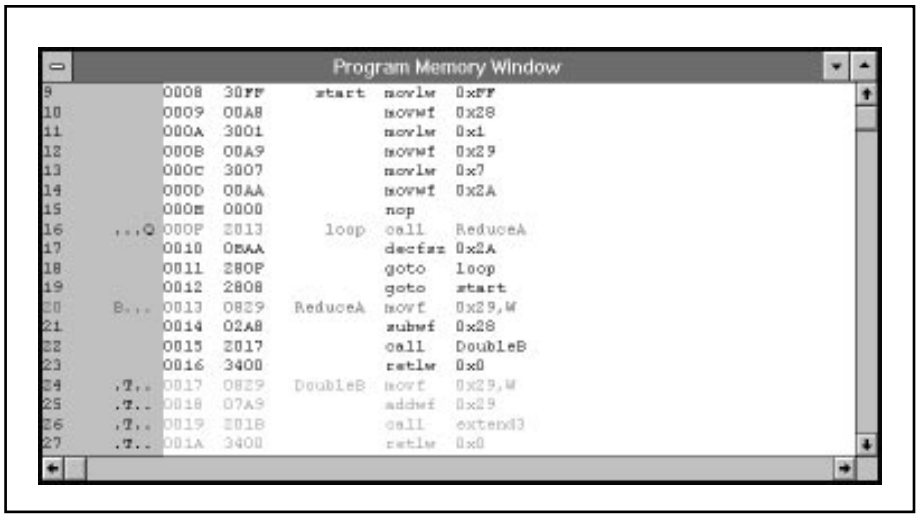



**Figure 3.5 Program Memory Window**

# Chapter 3. PICMASTER Emulator Basic Functions

## Real Time Breakpoints

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

The PICMASTER emulator provides the following ways to set a breakpoint:

- Break on Address Match
- Break on Trace Buffer Full
- Break on Pass Count Reached
- Break when External Condition is Met
- Break on Stack Overflow (Not all PICMASTER probes support this feature)
- Break on Watch Dog Timer Time Out (Not all PICMASTER probes support this feature)
- Break on User Halt

### Break on Address Match

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

**Note:** On multicycle instructions, the PICMASTER emulator ignores extra (non-instruction) fetches and breaks only on a valid instruction fetch.

### Break on Trace Buffer Full

You can program the PICMASTER emulator to halt the processor after capturing 8K selected cycles (when the trace buffer is full).

### Break on Pass Counter Equal to Predefined Value

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

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

**Note:** While the Pass Counter is running, if the emulator encounters any other breakpoint, the PICMASTER emulator will halt immediately.

### Break on Stack Overflow

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

| | |
|---|---|
| **Note:** | The PICMASTER emulator only supports Break on Stack Overflow for the PIC16CXX family and PIC17C4X family. It also supports break on stack underflow for the PIC16CXX family. |

### Break on Watch Dog Timer

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

### Break on User Halt

MPLAB executes until you press the **Halt** button on the Conditional Break display (*Debug > Execute > Conditional Break*).

## Real Time Trace Points

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

### Circular Trace Buffer

The PICMASTER emulator continuously captures selected bus cycles into the 8K x 40 bit trace buffer.

The 40 bits of status information captured into the trace buffer are grouped as follows:

- 16 Bits of Address
- 16 Bits of Opcode/Data
- 8 Bits of External Trace Probe Data

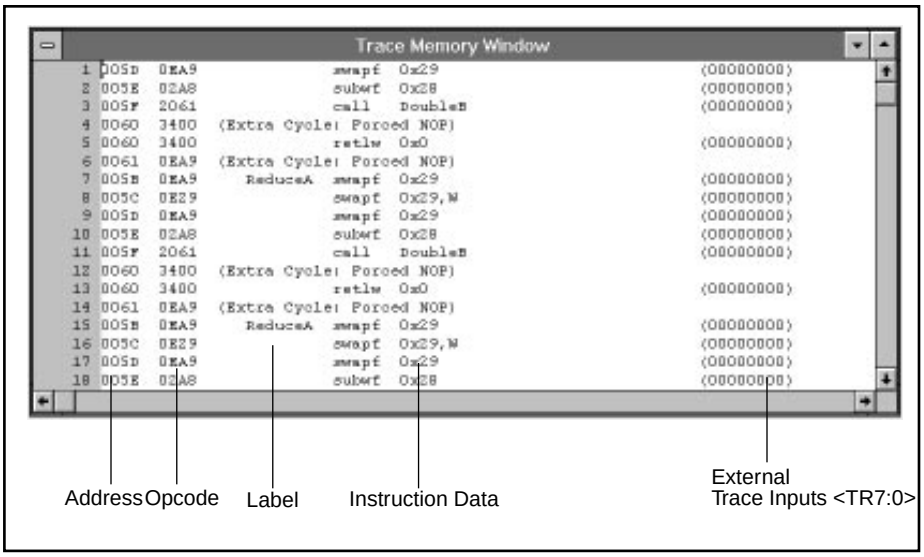



**Figure 3.6 Trace Memory Window**

## Halting Trace from the Tool Bar

Halt Trace allows you to take a snapshot of the trace buffer and look at the captured trace without halting the processor.

Select Halt Trace from the Tool Bar or select *Debug > Run > Halt Trace* from the menu to halt trace collection and display a snapshot of the trace buffer. Once the trace buffer halts, click Halt Trace from the Tool Bar again, or select *Debug > Run > Start Trace* from the menu to restart the trace collection. You must restart trace collection before another snapshot can be taken of the trace buffer.

## Halting Trace at a Precise Time

To halt the trace at a precise time, use the External Trace Halt probe signal (logic probe 11, BRKT). A *rising* edge on the External Trace Halt signal halts the trace buffer and displays the trace data.

**Example:** The External Trace Halt probe signal can be very useful in applications where you want to see program execution without halting the processor. For example, imagine the PIC17C42 controlling a high powered turbine (or a motor) using PWM signals. It is undesirable to halt the processor while the turbine is running. But to locate a bug, say on a current limiting signal being turned on, you could connect this external signal to the External Trace Halt signal.

## Halting the Trace at a Selected Address

When using PICMASTER logic probes you can stop the trace analyzer with an external signal and view the instructions in the trace buffer leading up to an event.

To halt a trace at an address, connect the External Trace Halt Signal to the Trigger Output Signal on the PICMASTER logic probes, and set the Trigger Out point to the address to stop the analyzer. The trace software will collect and display information leading up to the Trigger Out address in the analyzer without halting the processor.

## Trace Capture Cycles

### Capture Cycles

Trace provides three methods for capturing cycles into the trace buffer:

| | |
|---|---|
| Trace All | Valid for all PIC16/17 devices |
| Trace First Cycle Only | Valid for all PIC16/17 devices |
| Trace Data Cycle Only | Valid for PIC17CXX devices only |

Each capture cycle method also provides a trace of the state (0 or 1) on logic probes Tr0 through Tr7 in the Trace Memory window. The Tr7 bit is the left most bit in the External Trace Input field. The following figure shows the Trace Memory window.

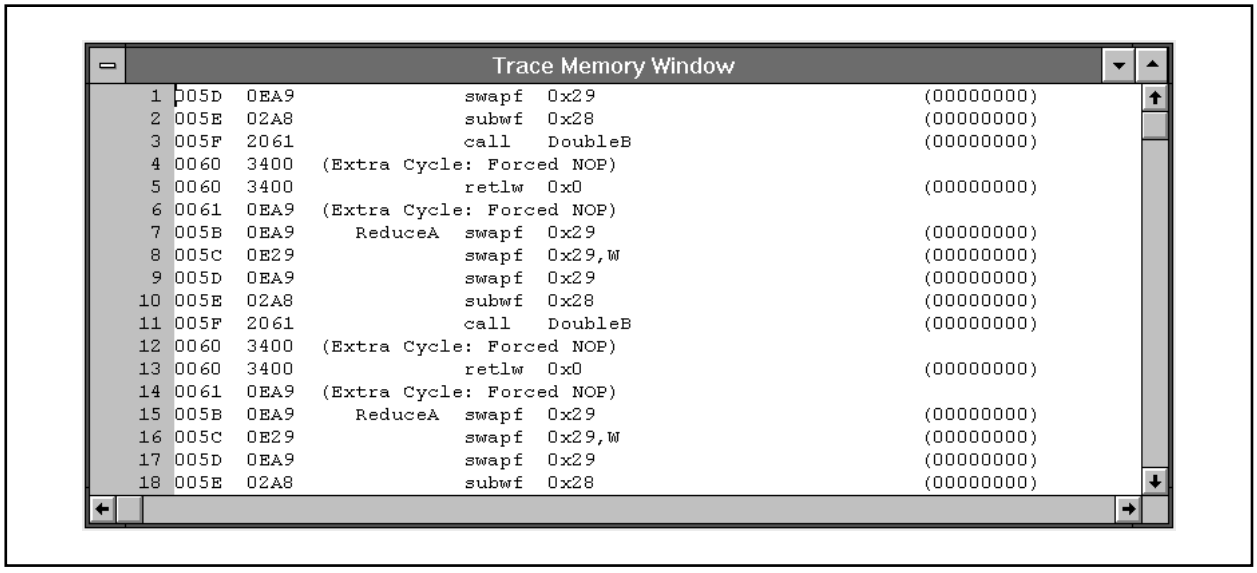



**Figure 3.7 Trace Memory Window**

Trace First Cycle Only — Eliminates (filters) cycles from the trace buffer of all extra forced NOP cycles (or data cycles) of two cycle instructions. This allows the trace buffer to display only the exact instruction flow sequence. Eliminating extra cycles also conserves valuable trace buffer space and provides more room to capture meaningful data.

Trace Data Cycle Only — Captures only the data cycle (second cycle) of the two-cycle table write (TABLWT) and table read (TABLRD) instructions. This option applies only to the PIC17CXX series. You can use this feature to

- Read an array of tables
- Capture data memory values, in real-time, from a desired RAM location and table write the values to an unused program memory location for debugging purposes

## External Break and Trigger Out Points

### External Break

Logic probe 14 (BRK), the External Break Input Signal, halts the processor either on a rising or falling edge. With the External Break Input Signal probe, you can program the emulator to halt the target processor when the External Break Input Signal detects an edge. Through the breakpoint settings dialog, you can select the edge to be either rising or falling. (See Figure 3.X and Figure 3.X.)In addition, you can program this external break line to either logically OR or logically AND with other internal hardware breakpoints.

When selected to OR with internal breaks, the target processor halts when it encounters the first valid break (either internal or external). When selected to AND with other internal hardware breakpoints, the processor halts only if the external break condition is true (either High or Low depending on your selection) and if any of the internal break conditions are valid.

**Note:** When an external break condition halts the processor, the processor actually halts after execution of the already fetched instruction.

The External Break Input Signal is extremely useful when emulating multiple processors in a system. For example, you can connect the Trigger Output Signal (logic probe 13, TRIGO) of the first PICMASTER emulator to the External Break Input (logic probe 14, BRK) of the second PICMASTER emulator. This connection allows you to halt both emulators at the same time.

**Example:** For an application of the External Break Input Signal, consider a logic signal on a target board indicating the target system is malfunctioning. If you connect the signal from the target board to the External Break Input line (logic probe 14), then as soon as a system error occurs, the processor halts.

### Trigger Output

The Trigger Output signal is available on logic probe 13, TRIGO. The trigger output signal is normally in a logic low state and goes high when a selected address is executed.

You can use the trigger output to:

- Trigger an oscilloscope precisely when the processor executes an instruction at a particular address.
- Trigger other instruments such as logic analyzers.
- Provide cross triggering between emulators to halt multiple PICMASTER emulators at the same time.
- Display the current trace information in the PICMASTER trace buffer without halting the processor.

To set up for cross triggering, connect the Trigger Output logic probe from one emulator to the External Break Input signal of an another PICMASTER emulator.

**Example:** A very simple use for trigger output is to measure the time for a subroutine to execute. Select all of the subroutine as the trigger range. Then measure the pulse width on an oscilloscope. The result is the same as looking at the cycle counter between breakpoints set at the start and end of the subroutine, except it will not halt the processor at a breakpoint.

## Assigning a Pass Count to Break or Trace Points

The PICMASTER emulator has a 16-bit Pass Counter which decrements by one on any address match in program memory.

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

### Pass Counter Assigned to Break

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

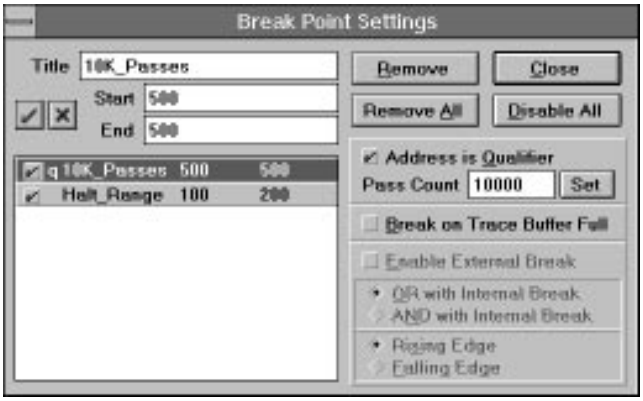



**Figure 3.8 Breakpoint dialog**

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

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

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

### Pass Counter Assigned to Trace

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

### Using Pass Counter to Count Events

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

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

## Hardware Dialog

The MPLAB menu selection, *Options > Processor Setup > Hardware* provides dialogs for setting up the PICMASTER emulator.

Refer to Chapter 4 for dialogs for the different PIC16/17 microcontroller families.

# Conditional Break Dialog

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

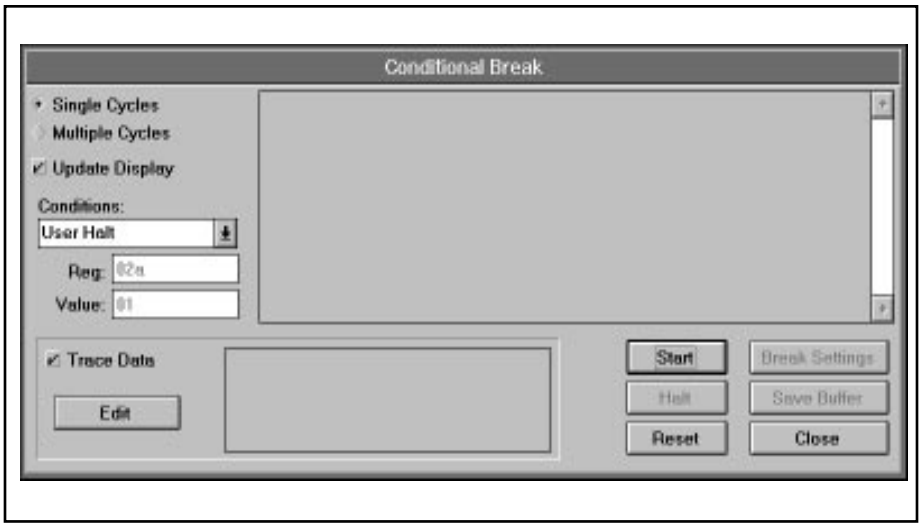



**Figure 3.9 Conditional Break Dialog**

## Conditions

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

- User Halt – The PICMASTER emulator executes until you press the **Halt** button on the Conditional Break display.
- Number of Cycles – The PICMASTER emulator halts after the target processor executes the specified number of cycles.
- Logic Condition satisfied.

## Trace Data

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

## Single Cycle

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

### Multiple Cycles

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

# Program Memory in the PICMASTER Emulator

The PICMASTER emulator emulates program memory as described below.

## PIC16C5X and PIC16CXX Series

The PIC16C5X and PIC16CXX series have no external memory capability. (All the program memory resides on the chip.) For these families, during development with the PICMASTER emulator, all program code resides in emulation program memory.

## PIC17CXX Series

The PIC17CXX requires program memory configuration in one of the following modes:

### Microcontroller Mode

In Microcontroller Mode, all memory resides on-chip, and during development the emulator provides the complete memory.

### Microprocessor Mode

In Microprocessor Mode, the PIC17CXX has a total of 64K words of address space. All instructions are fetched from external memory. On-chip memory is not accessible. When emulating this mode, you can configure the PICMASTER emulator to fetch all the instructions from either:

1. The Target Application Board.

   Set *Options > Processor Setup > Hardware* to supply Off Chip Memory 'From the Target Board'. All instructions from 0 to FFFF are fetched from the target application board. The trace buffer can capture code execution from either the PICMASTER emulator memory or target memory, but you cannot make changes directly to this memory, and the Program Memory window will not show you the contents of external memory.

   You can read this information into the PICMASTER emulator emulation memory for debugging. Use the *File > Import > Copy From Target Memory* to read the program from external memory into the PICMASTER emulator. Set *Options > Processor Setup > Hardware* to supply Off Chip Memory 'From the Emulator'.

2. The Emulator.

   The PICMASTER emulator provides the complete 64K memory.

### Extended Microcontroller Mode

In the PIC17CXX Extended Microcontroller Mode, the lower portion of program memory resides on chip and the remainder is fetched from external off-chip memory.

When emulating this mode, all program memory in the lower portion is always supplied by the PICMASTER emulator emulation memory. Program space in the upper portion can be programmed to be supplied either by the emulator or by the target application board.


**Figure 3.10 Hardware Dialog Box**

Use the *Options > Processor Setup > Hardware* dialog box to configure memory for the PIC17CXX device family.

**Note:** For specific information about these various memory configurations, refer to the data sheet for the device you are emulating.

## Download to Target Memory

Select *File > Import > Download to Target Memory* to download to target memory. This command is available only for the PIC17CXX operating in either Microprocessor or Extended Microcontroller modes.

In Microprocessor Mode or Extended Microcontroller Mode, the PIC17CXX is capable of executing instructions from external memory; if you have RAM on your target board, then the PICMASTER emulator can download the object code (*.HEX files) to this external static RAM and execute code from this RAM. To set up RAM, select *Options > Processor Setup > Hardware: Off Chip Memory*.

### Copy from Target Memory

Select *File > Import > Copy from Target Memory* to copy the contents of external memory on your target board to the emulator memory. After copying external memory from the target board to the emulator memory, you can save the contents of emulator memory to a file using *File > Export > Save Hex File*.

This command applies only to the PIC17CXX operating in either Microprocessor or Extended Microcontroller Modes.

## Multi-Processor Support

Select *Options > Processor Setup > Multi-Processor* to display the Multiprocessor Emulation dialog box. Multi-Processor emulation supports one master and up to three slave emulators.

Multi-Processor emulation allows you to emulate multiple processors using the same host computer. This function is only applicable when more than one instance of MPLAB is running under Microsoft Windows under the same host computer.

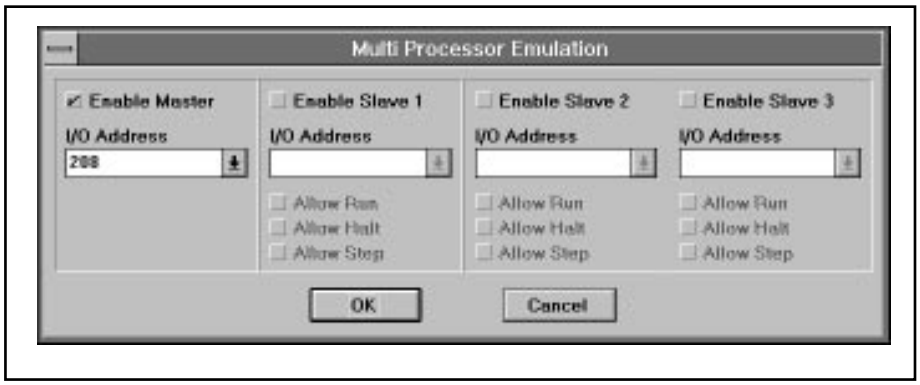



**Figure 3.11 Multi-Processor Dialog**

| | |
|---|---|
| Enable Master | Enables the master PICMASTER emulator. |
| I/O Address | Selects the PICMASTER I/O Port. |
| Enable Slave 1,2,3 | Enables slaves 1, 2, and 3. |
| I/O Address | Selects address of multi processors. I/O addresses can only be used once. |
| Allow Halt | Check this option to halt this instance of MPLAB whenever a Halt occurs on a different instance of MPLAB. |
| Allow Step | Check this option to allow this instance of MPLAB to perform Single Stepping whenever a Single Step command is issued on a different MPLAB. |

| | |
|---|---|
| Allow Run | Check this option to allow this instance of MPLAB to execute a Run command whenever a Run command is issued on a different instance of MPLAB. |
| Ok | Accepts multi processor selections, and launches the slave applications. |
| Cancel | Closes the Multi Processor dialog box without making any changes. |

**Note:** These commands are software triggered commands and are not synchronized through hardware. In cases where absolute hardware synchronization is necessary, use the External Break Input (Logic Probe 14) and Trigger Output Signals to cross couple each other.

# Chapter 4. PICMASTER Emulator Menu Options

## Introduction

This chapter gives detailed information on using the MPLAB menu options directly applicable to the PICMASTER emulator.

## Highlights

This chapter discusses portions of the following MPLAB pull-down menus:

- **File Menu**
- **Debug Menu**
- **Options Menu**
- **Tools Menu**

## File Menu

### Download to Target Memory

Select *File > Import > Download to Target Memory* to download to target memory. This command is available only for the PIC17CXX operating in either Microprocessor or Extended Microcontroller modes.

In Microprocessor Mode or Extended Microcontroller Mode, the PIC17CXX is capable of executing instructions from external memory; if you have RAM on your target board, then the PICMASTER emulator can download the object code (*.HEX files) to this external static RAM and execute code from this RAM. To set up RAM, select *Options > Processor Setup > Hardware: Off Chip Memory*.

### Copy from Target Memory

Select *File > Import > Copy from Target Memory* to copy the contents of external memory on your target board to the emulator memory. After copying external memory from the target board to the emulator memory, you can save the contents of emulator memory to a file using *File > Export > Save Hex File*.

This command applies only to the PIC17CXX operating in either Microprocessor or Extended Microcontroller Modes.

# Debug Menu

## Run

### Animate Mode

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

### Update All Registers

Select *Run > Update All Registers* to update the displayed values for all registers to the current state of the registers – or to the condition in which the executing program left the registers.

A typical use for this command would be to check an external stimulus on the target board without executing an instruction. For example, if you change a value on an input port on the target board, you could use Update All Registers to verify the new input value.

## Break Settings

Select *Debug > Break Settings* to display the Break Point Settings dialog box for defining up to 16 named breakpoint ranges.

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

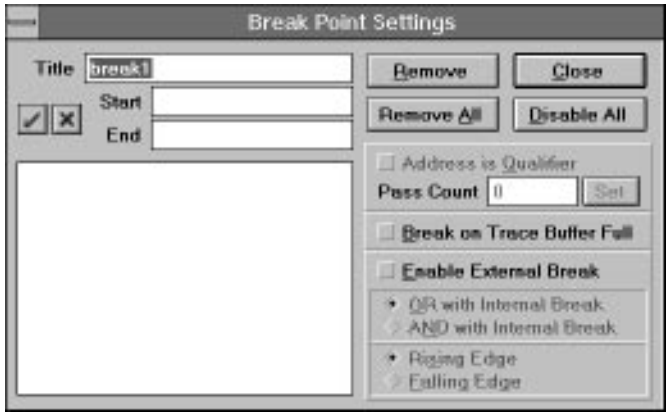



**Figure 4.1 Break Settings**

### External Break

Enable External Break

Defines the following additional break conditions for external break input signal:

**OR** with Internal Break Allows you to halt from the external input signal (logic probe 14, BRK), or with an internal breakpoint setting. If you select the OR condition the processor halts at the first valid occurrence of a break condition (internal or external).

**AND** with Internal Break Requires that the external break input signal (Logic probe 14, BRK) occurs at the same time as with other internal breakpoint settings. If you select the **AND** condition, the processor halts only when the External break occurs at a breakpoint address.

Rising Edge/Falling Edge

Selects the polarity of the external break input signal (Logic probe 14, BRK).

Select **Rising Edge** or **Falling Edge** button to program the external break input signal (Logic probe 14, BRK) to latch on either a rising or falling edge. When the external break input signal latches, the processor halts after one or two instructions.

## Trace Settings

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

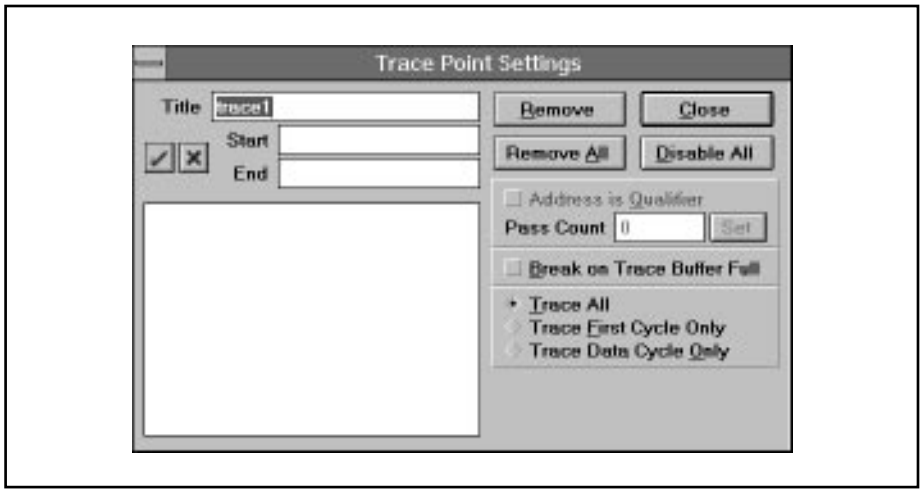



**Figure 4.2 Trace Settings**

**Note:** 1. The pass counter qualifier address can be assigned either to the Trace logic or break logic.

2. The pass counter qualifier address can be set either in this menu or in the Break Point Settings dialog.

## Capture Cycles

Trace provides three methods for capturing cycles into the trace buffer:

| | |
|---|---|
| Trace All | Valid for all PIC16/17 devices |
| Trace First Cycle Only | Valid for all PIC16/17 devices |
| Trace Data Cycle Only | Valid for PIC17CXX devices only |

Each capture cycle method also provides a trace of the state (0 or 1) on logic probes Tr0 through Tr7 in the Trace Memory window. The Tr7 bit is the left most bit in the External Trace Input field. The following figure shows the Trace Memory window.

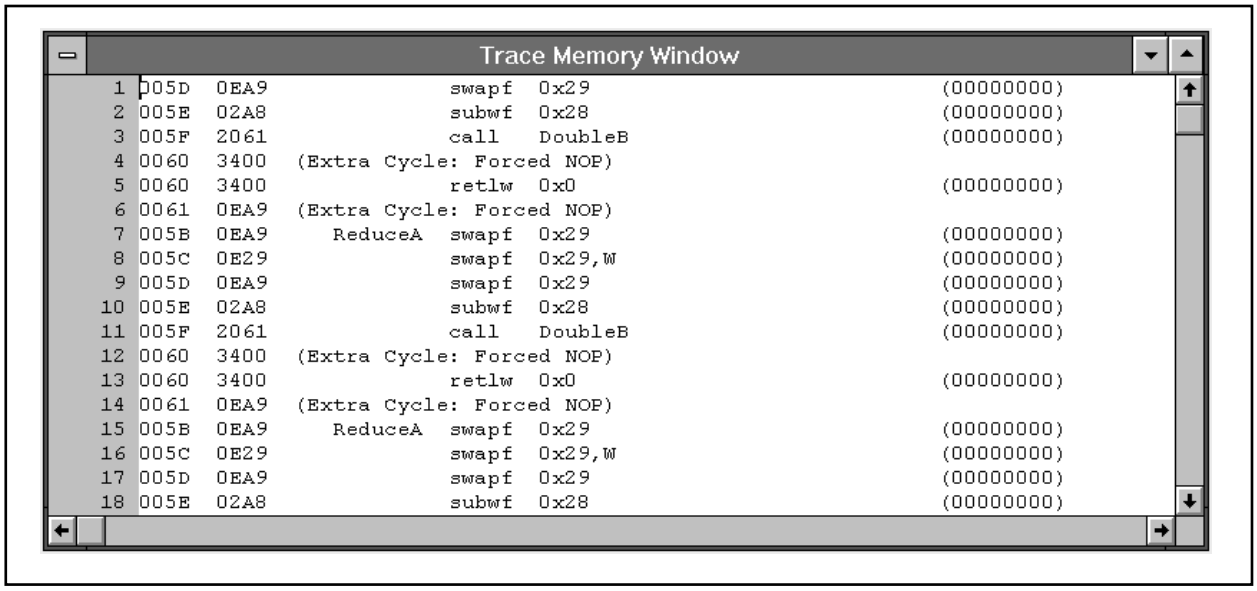



**Figure 4.3 Trace Memory Window**

Trace First Cycle Only — Eliminates (filters) cycles from the trace buffer of all extra forced NOP cycles (or data cycles) of two cycle instructions. This allows the trace buffer to display only the exact instruction flow sequence. Eliminating extra cycles also conserves valuable trace buffer space and provides more room to capture meaningful data.

Trace Data Cycle Only — Captures only the data cycle (second cycle) of the two-cycle table write (TABLWT) and table read (TABLRD) instructions. This option applies only to the PIC17CXX series. You can use this feature to

- Read an array of tables
- Capture data memory values, in real-time, from a desired RAM location and table write the values to an unused program memory location for debugging purposes

## Trigger Out Settings

Select *Debug > Trigger Out Settings* to display a dialog box for entering named trigger output address ranges.

When an instruction is selected for trigger output, the trigger output line on the external logic probe 13, TRIGO, goes to a high state. If the instruction is not selected, logic probe 13, TRIGO, goes to a low state. (Refer to the Logic Probes table in the Appendices.)

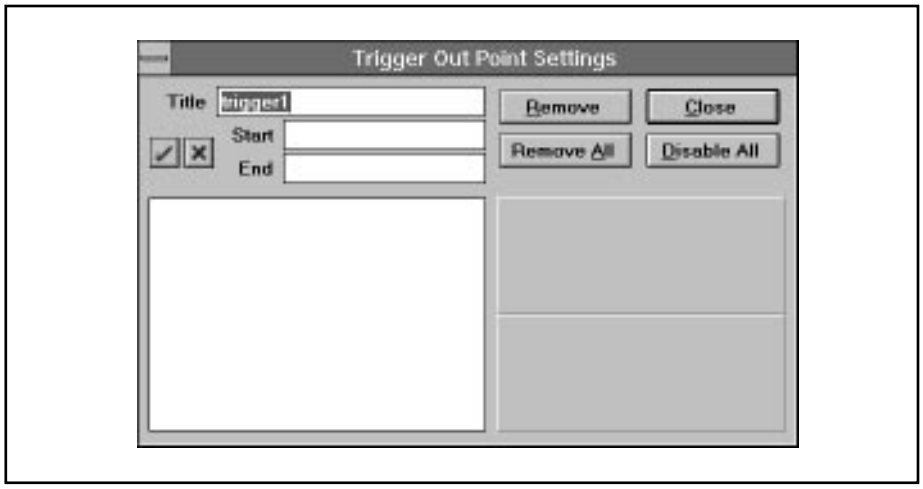



**Figure 4.4 Trigger Out Point Settings**

Applications for trigger output include:

- Trigger an Oscilloscope or other instrument precisely when the processor executes an instruction at a particular address.
- Cross trigger between emulators to halt multiple emulators at the same time by connecting the Trigger Output signal to the External Break Input signal of another emulator.

### System Reset (Ctrl+Shift+F3)

Select *Debug > System Reset* to reset the entire emulator system including the hardware, software and the target processor. System Reset performs the same initialization that is performed when MPLAB is first entered.

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

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

## Options Menu

### Development Mode

Select *Options > Development Mode* to change the current Development Mode setting.

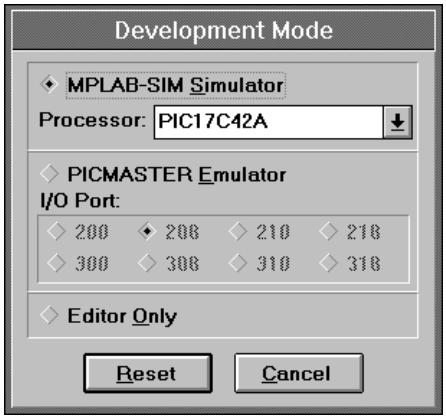



**Figure 4.5 Development Mode Dialog**

MPLAB-SIM Simulator — Selects MPLAB-SIM Simulator mode and displays "Sim on the Status Bar.

Processor — Selects the simulated processor.

PICMASTER Emulator — Selects the PICMASTER Emulator mode and displays Em on the Status Bar.

I/O Port — Click an I/O address to select an **I/O Port Address** for the PICMASTER interface card on your PC.

I/O Port Selects the I/O address of the PICMASTER interface card in your PC. Your I/O address setting will correspond to your current dip switch settings on your PC interface card.

Select a new address button to change the I/O Port selections.

Before selecting an I/O address with this command, set the dip switch settings on the PC interface card to the desired I/O address. The factory default I/O address setting is 0x300. Normally you will not need to change the factory default I/O address setting. However, if you have an address conflict with another card (Ethernet interface cards are often located at 0x300), then you must first change the conflicting switch setting on your PC interface card and then make a corresponding I/O address change with the MPLAB Software. The following table gives the dip switch setting corresponding to each I/O address.

whenever a particular I/O port is selected, the IDE determines if there is an interface card at that address. The IDE displays a warning message if there is no I/O card at the selected address.

**Table 4.6 I/O Address Corresponding to DIP Switch Settings**

| Base Address | DIP Switch Setting | | | |
|---|---|---|---|---|
| | 1 | 2 | 3 | 4 |
| 200 | ON | ON | ON | ON |
| 208 | ON | OFF | ON | ON |
| 210 | ON | ON | OFF | ON |
| 218 | ON | OFF | OFF | ON |
| 300 | OFF | ON | ON | OFF |
| 308 | OFF | OFF | ON | OFF |
| 310 | OFF | ON | OFF | OFF |
| 318 | OFF | OFF | OFF | OFF |

**Caution:** If you start with an invalid communication port selected MPLAB will report that PC-interface card is not found, define a valid communication port, and press **Reset**.

| | |
|---|---|
| Editor Only | Selects Editor Only mode. |
| Reset | Click **Reset** to reinitialize the emulator hardware. This reset function is a manual reset that you execute and is the same as the *Debug > System Reset* option. |
| Cancel | Select **Cancel** to cancel your selection and exit this display. |

## Processor Setup

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

### Hardware

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

### Hardware Selections — PIC16C5X Family

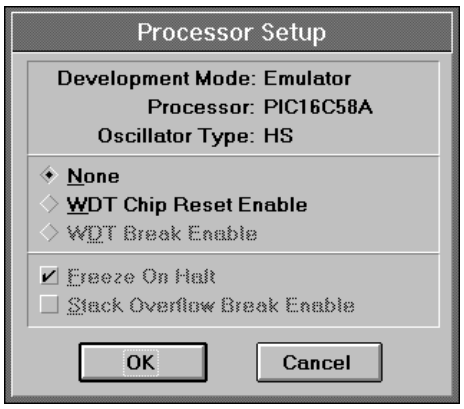



**Figure 4.7 PIC16C5X Processor Setup Dialog**

| | |
|---|---|
| Oscillator Type | The Oscillator Type field provides information only and cannot be modified with MPLAB.<br><br>Select the Oscillator Type by setting a dip switch on the probe. MPLAB reads the setting on the probe and displays the switch setting. See the appropriate probe specification for the oscillator switch settings. |
| None | Disables the Watch Dog Timer. |

| | |
|---|---|
| WDT Chip Reset Enable | Enables the Watch Dog Timer. |
| OK | Accepts entries made. |
| Cancel | Closes this dialog box without making any changes. |

### Hardware Selections — PIC16CXX Family

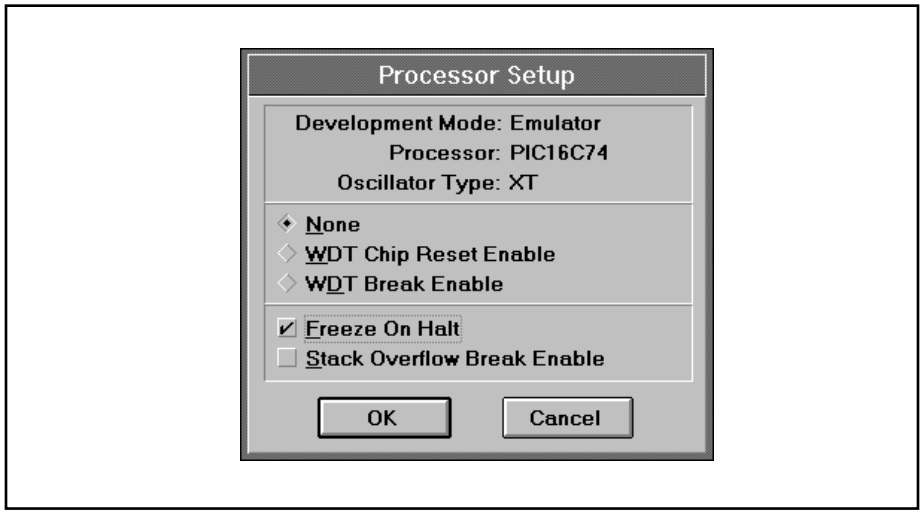



**Figure 4.8 PIC16CXX Processor Setup Dialog**

| | |
|---|---|
| Oscillator Type | The Oscillator Type field provides information only and cannot be modified with MPLAB.<br><br>Select the Oscillator Type by setting a dip switch on the probe. MPLAB reads the setting on the probe and displays the switch setting. See the appropriate probe specification for the oscillator switch settings. |
| None | Disables the Watch Dog Timer |
| WDT Chip Reset Enable | Enables the Watch Dog Timer. |
| WDT Break Enable | Halts the processor when the Watch Dog Timer times out. |
| Freeze on Halt | The emulator halts all internal processor events and ignores (does not record) external events such as an event on the probe interrupt pin. . |
| Stack Overflow Break Enable | Select Stack Break Enable to halt the processor if a stack overflow or underflow occurs. |
| OK | Accepts entries made. |
| Cancel | Closes this dialog box without making any changes. |

### Hardware Selections — PIC17CXX Family

The Processor Setup dialog for the PIC17CXX family allows you to set configuration bits and the emulator configuration.

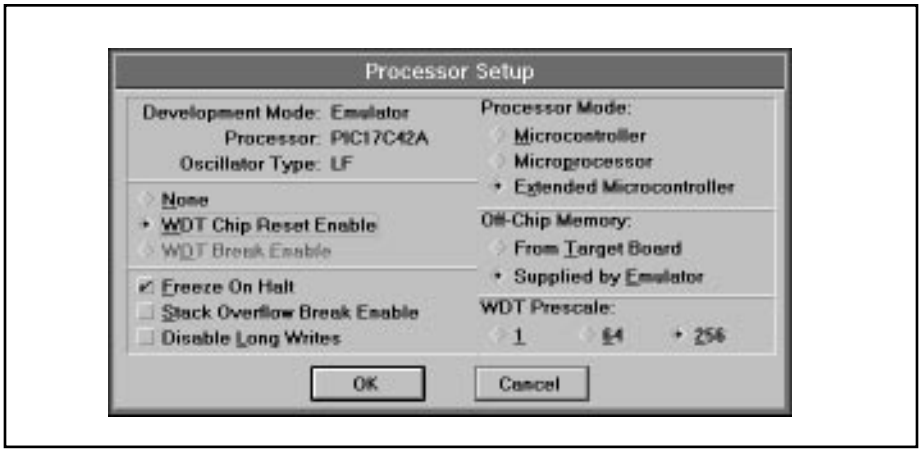



**Figure 4.9 PIC17C4X Processor Setup Dialog**

Oscillator Type — The Oscillator Type field provides information only and cannot be modified with MPLAB.

Select the Oscillator Type by setting a dip switch on the probe. MPLAB reads the setting on the probe and displays the switch setting. See the appropriate probe specification for the oscillator switch settings.

None — Disables the Watch Dog Timer.

WDT Chip Reset Enable — Enables the Watch Dog Timer.

Freeze on Halt — Select **Freeze on Halt** to "Freeze" peripherals when the target processor is halted at a breakpoint. In a normal debug session, Freeze on Halt should be ON to freeze all peripherals when a breakpoint occurs to permit a true snapshot of the chip status.

With Freeze on Halt selected, the emulator halts all internal processor events and ignores (does not record) external events such as an event on the probe interrupt pin. Consult the probe specification for details on frozen peripherals. Frozen peripherals are specific to the emulator chip and may vary from processor to processor.

**Note:** In some applications, it could be necessary to allow the peripherals to continue running even though the chip is halted. For example, imagine the PIC17C42 controlling a motor using the PWM outputs of the PIC17C42. If the chip is halted and the peripherals are frozen, then all the PWM lines are either high or low which could put the motor in an uncontrolled situation. However, if the peripherals are not frozen, then the PWMs continue to run even when the chip is halted.

Stack Overflow Break Enable — Halts the processor when a stack overflows.

Disable Long Writes — Disables any attempt to try to program EPROM.

**Processor Mode**

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

| Mode | Internal Memory | External Memory |
|---|---|---|
| Microcontroller | 0- [max]* | None |
| Microprocessor | None | 0 - 0xFFFF |
| Extended Microcontroller | 0 - [max]* | [max]* - 0xFFFF |
| [max] Max refers to the maximum ROM memory value.<br>* Refer to the Microchip data sheet to get the memory size for the PIC17CXX microcontroller that you are using. | | |

**Off Chip Memory**

Allows you to select where program memory resides.

| | |
|---|---|
| From Target Board | Selects the target board as the source for program memory. |
| Supplied by Emulator | Selects the emulator as the source for program memory. (The emulator will supply program memory instead of the target board). |

**WDT Prescale**

The WDT Prescale value selects the number of timer cycles required increment the Watch Dog Timer.

| | |
|---|---|
| OK | Accepts entries made. |
| Cancel | Closes this dialog box without making any changes. |

## Multi-Processor

Select *Options > Processor Setup > Multi-Processor* to display the Multiprocessor Emulation dialog box. Multi-Processor emulation supports one master and up to three slave emulators.

Multi-Processor emulation allows you to emulate multiple processors using the same host computer. This function is only applicable when more than one instance of MPLAB is running under Microsoft Windows under the same host computer.

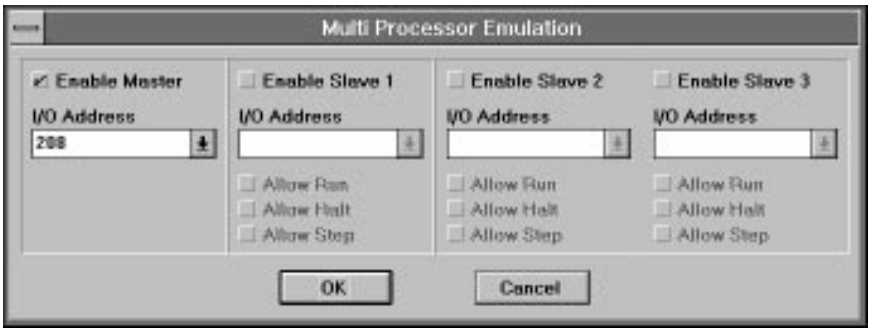



**Figure 4.10 Multi-Processor Dialog**

| | |
|---|---|
| Enable Master | Enables the master PICMASTER emulator. |
| I/O Address | Selects the PICMASTER I/O Port. |
| Enable Slave 1,2,3 | Enables slaves 1, 2, and 3. |
| I/O Address | Selects address of multi processors. I/O addresses can only be used once. |
| Allow Halt | Check this option to halt this instance of MPLAB whenever a Halt occurs on a different instance of MPLAB. |
| Allow Step | Check this option to allow this instance of MPLAB to perform Single Stepping whenever a Single Step command is issued on a different MPLAB. |
| Allow Run | Check this option to allow this instance of MPLAB to execute a Run command whenever a Run command is issued on a different instance of MPLAB. |

**Note:** These commands are software triggered commands and are not synchronized through hardware. In cases where absolute hardware synchronization is necessary, use the External Break Input (Logic Probe 14) and Trigger Output Signals to cross couple each other.

| | |
|---|---|
| Ok | Accepts multiprocessor selections, and launches the slave applications. |
| Cancel | Closes the Multi Processor dialog box without making any changes. |

**Note:** When using a multiprocessor emulation environment, you will need to change the I/O address for at least one PICMASTER emulator. For example, two PICMASTER emulators can run at the same time on the same host computer. In this case, the two PC interface cards must be set at different I/O addresses.

# Tools Menu

## Verify PICMASTER

The *Tools > Verify PICMASTER* function is an element of the MPLAB system that gives you the ability to determine if your PICMASTER emulator is operating correctly.

The Verify function purpose is to give you some indication whether or not your PICMASTER emulator is working correctly.

The test encompasses the following functional blocks:

- Write/Read Emulation Memory
- Write/Read Break, Trace, Trigger Out Memory
- Write/Read Trace Memory
- Emulation Memory
- Breakpoint Logic
- Trace Logic
- Processor RAM
- Pass Counter
- Table Write (PIC17CXX Family Only)

### Using Verify PICMASTER

The Verify PICMASTER function gives you an assurance that your hardware is working correctly. Verify leads you through several simple steps to prepare the PICMASTER emulator for the Verify process. It uses a series of message boxes as follows:

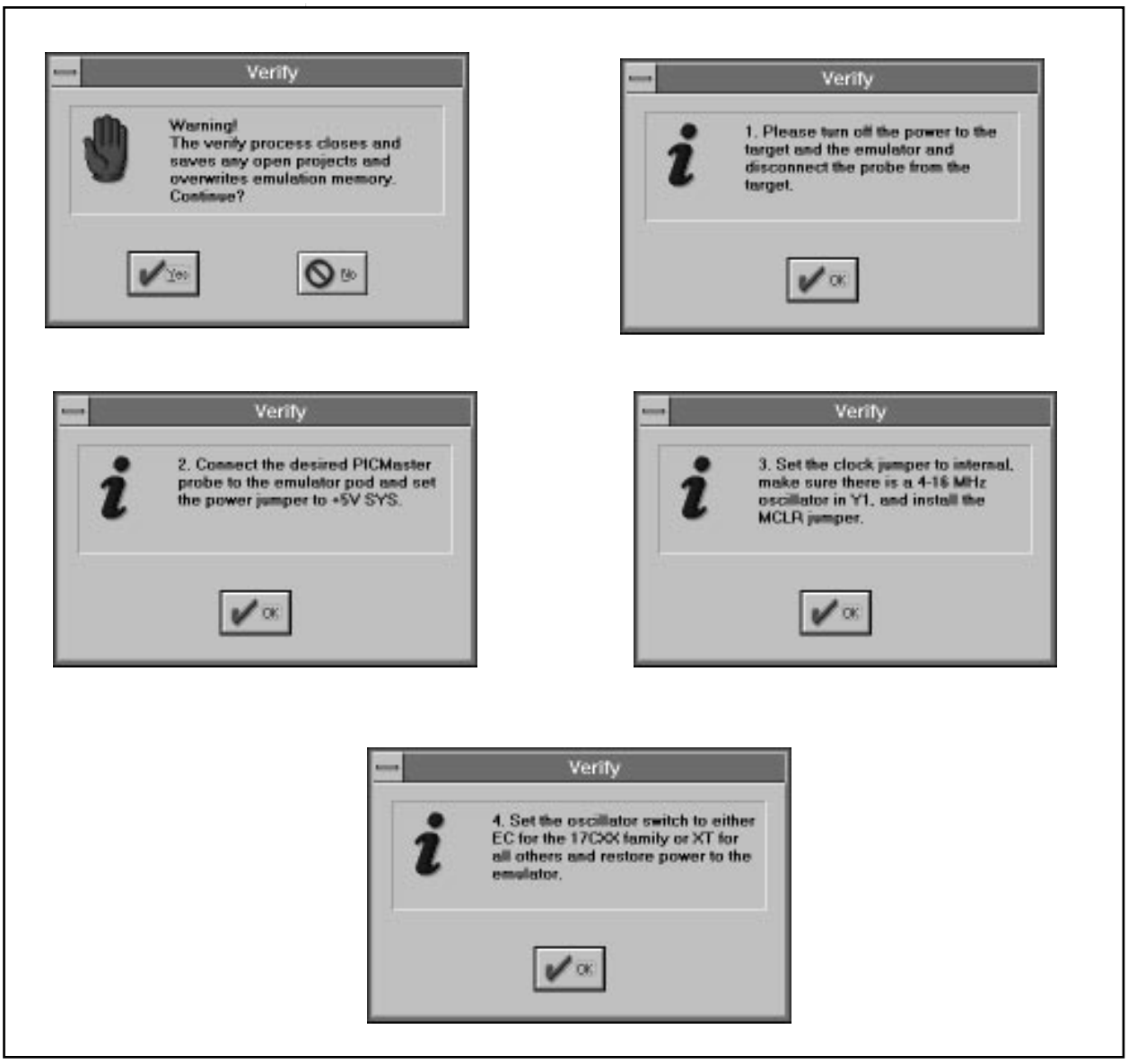



**Figure 4.11 Verify Setup Dialog Boxes**

Verify next displays the Verify PICMASTER dialog, as seen below. When you press either "Single Test" or "Start Loop," Verify indicates that it is testing the PICMASTER hardware. When done, it either indicates a pass or a failure in the Status field of the dialog. If there is a failure, Verify indicates the name of the test that failed.

The Non-Critical errors are auxiliary functions such as the 16-bit Pass Counter; the PICMASTER hardware can still function as an emulator.

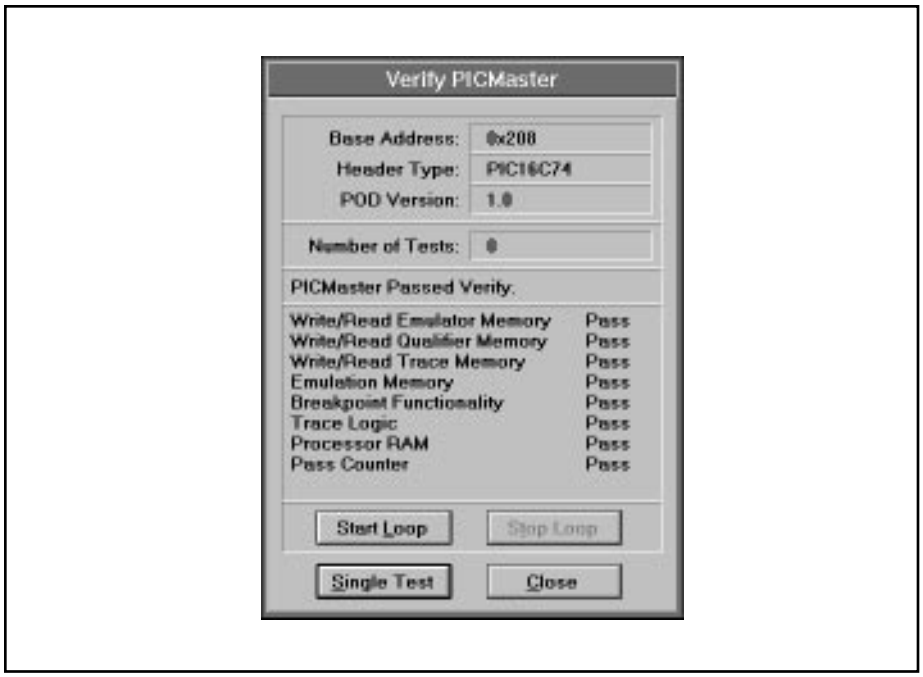



**Figure 4.12 Verify PICMASTER**

## Configure Probe

Select *Tools > Configure Probe* to display the Program Header EEPROM dialog box. This dialog box allows you to select the target processor that you will be emulating.

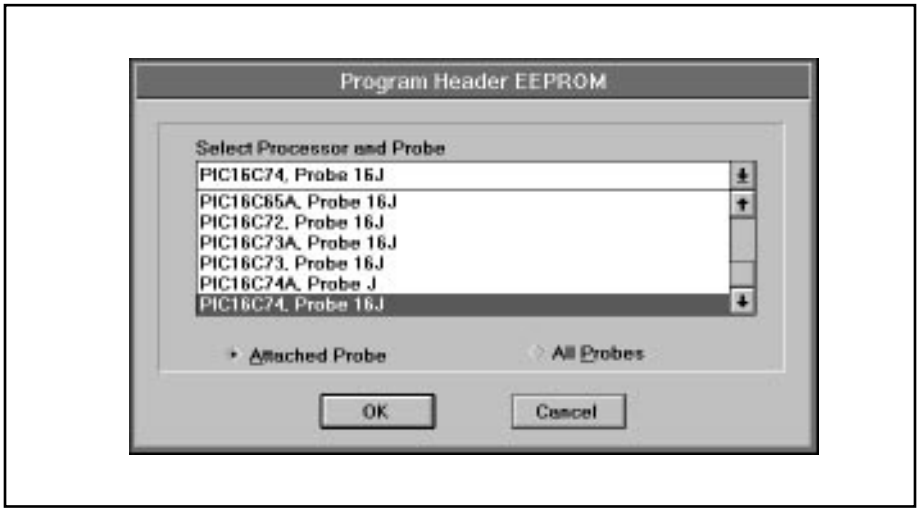



**Figure 4.13 Configure Probe Dialog**

| | |
|---|---|
| Select Processor and Probe | Displays all available processors and the supporting probe for each processor. |
| Attached Probe | Selects the currently attached probe. |
| All Probes | Allows you to select from all available probes. |
| OK | Accepts the selections that you make and configures the probe as the selected device. |
| Cancel | Closes this dialog box without making any changes. |

# Chapter 5. Migrating From PICMASTER 1.xx Software

## Introduction

This chapter describes the procedures for migrating from PICMASTER 1.xx software.

## Highlights

The items discussed in this chapter include:

- **Migrating from PICMASTER 1.xx Software**

## Migrating From PICMASTER 1.xx Software

If you are a previous user of the PICMASTER emulator, you will notice some differences between the PICMASTER software and MPLAB. MPLAB provides project management and enhanced editing capabilities that the PICMASTER 1.xx software did not have. These enhancements reduce the debug cycle time for an embedded project by having the editor, compiler, assembler, and debugger available in one package on your desktop.

Since MPLAB is project based, you no longer are required to load a *.HEX file into program memory. The project manager automatically loads code into program memory and loads symbols for debugging.

While PICMASTER emulator functionality is still available, some functions have moved. The following table shows some of these changes.

| PICMASTER Menu | MPLAB Menu | Example |
|---|---|---|
| Configure | Options/Tools | *Options > Processor Setup > Hardware* |
| Setup | Debug | *Debug > Break Settings* |
| Watch | Window | *Window > Program Memory* |
| Run | *Debug > Run* | *Debug > Run > Reset* |
| Utility | Tools/Project | *Project > Make Setup* |

For a complete list of the menu items available in MPLAB please refer to the MPLAB User's Guide, Chapter 6: MPLAB Menu Options.

# Appendix A. Troubleshooting Guide

## Introduction

The troubleshooting information in this appendix can help you with typical problems you may have in using the emulator, and describes how to resolve these problems.

## Highlights

The troubleshooting information in this chapter includes the following:

- **Hardware Related Problems**
  - Assembling the Probe
  - Emulator Power Source
  - PC Interface Card Connection
  - Emulator Clock/Target System Clock
  - Running a Second PICMASTER Emulator
- **Software Related Problems**
  - PC Interface Base Address
  - External Program Memory with the PIC17CXX
  - Breakpoints/Trace Points Disappear on Download
  - Enabling Breakpoints
  - Stack Overflow, Underflow
  - Resetting the Emulator System

## Hardware Related Problems

### Assembling the Probe

If the two probe boards are not making good contact, the message “Can't Identify Probe” may be displayed. If you receive this message, ensure that the boards are squeezed tightly together.

To assemble a probe from scratch, carefully attach a daughter board to the probe according to the probe specifications. Secure the daughter board to the probe board with the four standoff connectors. The daughter board provides device identification, and pinouts for connecting to the target board.

## Emulator Power Source

The PC Interface Card can power the Emulator System directly. However, sometimes a PC power supply may not be able to provide enough power to the PICMASTER emulator.

If you power on the PICMASTER emulator but the green power LED light on the PICMASTER emulator is off, your PC power supply may be heavily loaded. In this event, use the external power supply provided.

1. Ensure that the three position power rocker switch on the PICMASTER emulator is in the center OFF position.
2. Insert the circular DIN plug into the receptacle marked POWER
3. Connect the power supply to the wall outlet
4. Power on the PICMASTER emulator

### Selecting Power Source on the Probe

You can configure power to the probe to come from the Emulator, or to come from the target system by selecting INT (Internal) or EXT (External) power on the probe board.

Internal Power: Jumper INT if the Emulator is to supply +5VDC.

External Power: Jumper EXT if the target prototype system is to supply power.

## PC Interface Card Connection

The PC interface card connects the Emulator System to a compatible PC host. (This high-speed parallel interface requires a single half-size standard AT/ISA slot in the PC host.)

**PC to Pod Connection:** The Emulator system will not operate properly if the D-Sub ribbon cable connecting the PC interface card to the Emulator is not fully inserted. Verify that this cable is fully inserted if the Emulator system fails to operate properly.

**Probe Cable Connection:** Connect one end of the 14" ribbon cable to the Emulator at the connector marked EMULATOR PROBE. Connect the other end of this cable to the assembled Emulator Probe Board.

Ensure cable is fully inserted and the ejectors are locked or the system may not operate properly.

### Emulator Clock/Target System Clock

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 probe to select the internal Emulator Clock or the user target system clock.

Internal Clock: Jumper INTCLK if the Emulator is to provide the system clock from the crystal on the probe board.

External Clock: Jumper EXTCLK if the clock on the target application is to provide the system clock.

### Running a Second PICMASTER Emulator

If you get an “Unable to Find Emulator System” error when running a second PICMASTER emulator, select the correct base address from *Options > Development Mode*.

The *Options > Processor Setup > Multi-Processor* option allows your target application to use multiple processors running under the same PC Host. Select this command to run multiple instances of MPLAB.EXE under the same PC host.

## Software Related Problems

### PC Interface Base Address

Base address conflicts can occur with Ethernet cards or other I/O cards that will cause undesired operation with the PICMASTER emulator. If a base address conflict should occur, change the dip switch setting on the PC Interface Card to an unused address.

Locate the PC Interface Card and make your I/O address selection by setting the four DIP switches on SW1. The I/O address configuration DIP switch (SW1) is located on the top left corner of the PC Interface Card. The factory default setting is 300H. You may use another address setting if it does not conflict with other device cards in your system.

The eight selectable PC interface card address options are given in the following table:

**Table A.1 Base Address Corresponding to DIP Switch Settings**

| Base Address | DIP Switch Setting | | | |
|---|---|---|---|---|
| | 1 | 2 | 3 | 4 |
| 200 | ON | ON | ON | ON |
| 208 | ON | OFF | ON | ON |
| 210 | ON | ON | OFF | ON |
| 218 | ON | OFF | OFF | ON |
| 300 | OFF | ON | ON | OFF |
| 308 | OFF | OFF | ON | OFF |
| 310 | OFF | ON | OFF | OFF |
| 318 | OFF | OFF | OFF | OFF |

**Caution:** If you should start the PICMASTER emulator with an invalid communication port selected, define a valid communication port from *Options > Development Mode*.

**Caution:** Pressing OK will cause the Emulator to RESET. This will erase all information presently in the Emulator system.

**Note:** If you are using a multiprocessor emulation environment, change the base address for at least one PICMASTER emulator. For example, two PICMASTER emulators can run at the same time on the same host computer. In this case, the two PC Interface Boards must be set at different base addresses.

## External Program Memory with the PIC17CXX

If the external portions of your target board do not work properly when using a PIC17CXX device in the Extended Microcontroller mode, check the settings in the *Options > Processor Setup > Hardware* dialog box.

### Processor Mode

The PIC17CXX requires program memory configuration in one of the following modes:

- Microcontroller Mode
- Microprocessor Mode
- Extended Microcontroller Mode

In the PIC17CXX extended microcontroller mode, the lower portion of program memory resides on chip and the remainder is fetched from external off chip memory.

When emulating the Extended Microcontroller mode, all program memory in the lower portion is always supplied by the PICMASTER emulation memory. Program space in the upper portion can be programmed to be supplied either by the emulator or by the target application board.

If you have memory or I/O devices on your target board and these are memory mapped, select “Off Chip Memory: From Target Board” as shown below.

**Dialog Selections**

| | |
|---|---|
| Oscillator Type | The Oscillator Type field provides information only and cannot be modified with MPLAB.<br><br>Select the Oscillator Type by setting a dip switch on the probe. MPLAB reads the setting on the probe and displays the switch setting. See the appropriate probe specification for the oscillator switch settings. |
| None | Disables the Watch Dog Timer. |
| WDT Chip Reset Enable | Enables the Watch Dog Timer. |
| Freeze on Halt | Select **Freeze on Halt** to “Freeze” peripherals when the target processor is halted at a breakpoint. In a normal debug session, Freeze on Halt should be ON to freeze all peripherals when a breakpoint occurs to permit a true snapshot of the chip status.<br><br>With Freeze on Halt selected, the emulator halts all internal processor events and ignores (does not record) external events such as an event on the probe interrupt pin. Consult the probe specification for details on frozen peripherals. Frozen peripherals are specific to the emulator chip and may vary from processor to processor. |

**Note:** In some applications, it could be necessary to allow the peripherals to continue running even though the chip is halted. For example, imagine the PIC17C42 controlling a motor using the PWM outputs of the PIC17C42. If the chip is halted and the peripherals are frozen, then all the PWM lines are either high or low which could put the motor in an uncontrolled situation. However, if the peripherals are not frozen, then the PWMs continue to run even when the chip is halted.

| | |
|---|---|
| Stack Overflow Break Enable | Halts the processor when a stack overflows. |
| Disable Long Writes | Disables any attempt to try to program EPROM. |

Processor Mode

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

| Mode | Internal Memory | External Memory |
|---|---|---|
| Microcontroller | 0- [max]* | None |
| Microprocessor | None | 0 - 0xFFFF |
| Extended Microcontroller | 0 - [max]* | [max]* - 0xFFFF |
| [max] Max refers to the maximum ROM memory value.<br>* Refer to the Microchip data sheet to get the memory size for the PIC17CXX microcontroller that you are using. | | |

Off Chip Memory

Allows you to select where program memory resides.

From Target Board — Selects the target board as the source for program memory.

Supplied by Emulator — Selects the emulator as the source for program memory. (The emulator will supply program memory instead of the target board).

WDT Prescale

The WDT Prescale value selects the number of timer cycles required increment the Watch Dog Timer.

OK — Accepts entries made.

Cancel — Closes this dialog box without making any changes.

## Breakpoints/Trace Points Disappear on Download

If breakpoints and trace points 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 software will clear emulation memory before downloading a file.

## Enabling Breakpoints

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

# Appendix A. Troubleshooting Guide

## Stack Overflow, Underflow

Stack overflow or underflow may occur for some PIC16/17 emulated devices. The following paragraphs detail the PICMASTER 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—PIC16CXX

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.

### 16-Levels Deep Hardware Stack—PIC17CXX

No underflow is supported for the PIC17CXX device family.

## Resetting the Emulator System

Often, resetting the Emulator will clear an existing problem and allow you to continue using the PICMASTER emulator without further troubleshooting.

Select *Debug > System Reset* to reset the entire Emulator system (including the Hardware, Software and the target processor). System Reset performs the same initialization that is performed when the PICMASTER emulator is first loaded.

# Appendix B. On Line Support

## Introduction

Microchip provides two methods of on-line support. These are the Microchip BBS and the Microchip World Wide Web (WWW) site.

Use Microchip's Bulletin Board Service (BBS) to get current information and help about Microchip products. Microchip provides the BBS communication channel for you to use in extending your technical staff with microcontroller and memory experts.

To provide you with the most responsive service possible, the Microchip systems team monitors the BBS, posts the latest component data and software tool updates, provides technical help and embedded systems insights, and discusses how Microchip products provide project solutions.

The web site, like the BBS, is used by Microchip as a means to make files and information easily available to customers. To view the site, the user must have access to the Internet and a web browser, such as Netscape or Microsoft Explorer. Files are also available for FTP download from our FTP site.

## Connecting to the Microchip Internet Web Site

The Microchip web site is available by using your favorite Internet browser to attach to:

**www.microchip.com**

The file transfer site is available by using an FTP service to connect to:

**ftp.mchip.com/biz/mchip**

The web site and file transfer site provide a variety of services. Users may download files for the latest Development Tools, Datasheets, Application Notes, User's Guides, Articles and Sample Programs.

A variety of Microchip specific business information is also available, including listings of Microchip sales offices, distributors and factory representatives. Other data available for consideration is:

- Latest Microchip Press Releases
- Technical Support Section with Frequently Asked Questions
- Design Tips
- Device Errata
- Job Postings
- Microchip Consultant Program Member Listing
- Links to other useful web sites related to Microchip Products

# Connecting to the Microchip BBS

Connect worldwide to the Microchip BBS using either the Internet or the CompuServe® communications network.

Internet: You can telnet or ftp to the Microchip BBS at the address **mchipbbs.microchip.com**

CompuServe Communications Network: In most cases, a local call is your only expense. The Microchip BBS connection does not use CompuServe membership services, therefore

**You do not need CompuServe membership to join Microchip's BBS.**

There is **no charge** for connecting to the BBS, except for a toll charge to the CompuServe access number, where applicable. You do not need to be a CompuServe member to take advantage of this connection (you never actually log in to CompuServe).

The procedure to connect will vary slightly from country to country. Please check with your local CompuServe agent for details if you have a problem. CompuServe service allow multiple users at baud rates up to 14400 bps.

The following connect procedure applies in most locations.

1. Set your modem to 8-bit, No parity, and One stop (8N1). This is not the normal CompuServe setting which is 7E1.
2. Dial your local CompuServe access number.
3. Depress **<Enter↵>** and a garbage string will appear because CompuServe is expecting a 7E1 setting.
4. Type +, depress **<Enter↵>** and Host Name: will appear.
5. Type **MCHIPBBS,** depress **<Enter↵>** and you will be connected to the Microchip BBS.

In the United States, to find the CompuServe phone number closest to you, set your modem to 7E1 and dial (800) 848-4480 for 300-2400 baud or (800) 331-7166 for 9600-14400 baud connection. After the system responds with `Host Name:`, type **NETWORK,** depress **<Enter↵>** and follow CompuServe's directions.

For voice information (or calling from overseas), you may call (614) 723-1550 for your local CompuServe number.

# Using the Bulletin Board

The bulletin board is a multifaceted tool. It can provide you with information on a number of different topics.

- Special Interest Groups
- Files
- Mail
- Bug Lists

## Special Interest Groups

Special Interest Groups, or SIGs as they are commonly referred to, provide you with the opportunity to discuss issues and topics of interest with others that share your interest or questions. SIGs may provide you with information not available by any other method because of the broad background of the PIC16/17 user community.

There are SIGs for most Microchip systems, including:

- MPASM
- PRO MATE™
- PICSTART®
- Utilities
- Bugs
- TrueGauge®
- *fuzzy*TECH®-MP
- ASSP
- MTE1122
- MPLAB

These groups are monitored by the Microchip staff.

## Files

Microchip regularly uses the Microchip BBS to distribute technical information, application notes, source code, errata sheets, bug reports, and interim patches for Microchip systems software products. Users can contribute files for distribution on the BBS. For each SIG, a moderator monitors, scans, and approves or disapproves files submitted to the SIG. No executable files are accepted from the user community in general to limit the spread of computer viruses.

## Mail

The BBS can be used to distribute mail to other users of the service. This is one way to get answers to your questions and problems from the Microchip staff, as well as keeping in touch with fellow Microchip users worldwide.

Consider mailing the moderator of your SIG, or the SYSOP, if you have ideas or questions about Microchip products, or the operation of the BBS.

# Software Releases

Software products released by Microchip are referred to by version numbers. Version numbers use the form:

```
xx.yy.zz
```

Where `xx` is the major release number, `yy` is the minor number, and `zz` is the intermediate number.

### Intermediate Release

Intermediate released software represents changes to a released software system and is designated as such by adding an intermediate number to the version number. Intermediate changes are represented by:

- Bug Fixes
- Special Releases
- Feature Experiments

Intermediate released software does not represent our most tested and stable software. Typically, it will not have been subject to a thorough and rigorous test suite, unlike production released versions. Therefore, users should use these versions with care, and only in cases where the features provided by an intermediate release are required.

Intermediate releases are primarily available through the BBS.

### Production Release

Production released software is software shipped with tool products. Example products are PRO MATE, PICSTART, and PICMASTER. The Major number is advanced when significant feature enhancements are made to the product. The minor version number is advanced for maintenance fixes and minor enhancements. Production released software represents Microchip's most stable and thoroughly tested software.

There will always be a period of time when the Production Released software is not reflected by products being shipped until stocks are rotated. You should always check the BBS or the WWW for the current production release.

## Systems Information and Upgrade Hot Line

The Systems Information and Upgrade Line provides system users a listing of the latest versions of all of Microchip's development systems software products. Plus, this line provides information on how customers can receive any currently available upgrade kits. The Hot Line Numbers are: 1-800-755-2345 for U.S. and most of Canada, and 1-602-786-7302 for the rest of the world.

These phone numbers are also listed on the "Important Information" sheet that is shipped with all development systems. The hot line message is updated whenever a new software version is added to the Microchip BBS, or when a new upgrade kit becomes available.

# Appendix C. Pod Electrical Specifications

## Introduction

This document describes the hardware electrical specification of the PICMASTER and PICMASTER CE in-circuit emulator pods.

Refer to "PICMASTER Universal In-Circuit Emulator User's Manual" (Document number DS30106) for full functional description of the system. Also refer to "Probe Specification" for information on any specific probe.

## Power

Power to the PICMASTER pod may be supplied by the host PC or by external power supply included in your system. PICMASTER uses a single +5V supply.

### Power supply requirements

+5V, ±5%, 2A

#### **Power Host/Off/Ext Switch** (In the rear panel)

Three position switch selects between the power options.

Host: Host PC supplies power to the pod.

Ext: External power supply powers the pod.

Off: Disconnect power.

#### **Power** (In the front panel)

Is lit when +5V supply is good. It is off if the supply goes below acceptable level.

If using host power supply and if there is concern about the power rating of the host PC, check supply voltage level at $V_{DD}$ logic probe (pin 1) to be greater than 4.75V.

### Applying Power to System Components

Power up the system components in the following sequence to prevent damage to any of the sub-system parts or user target application parts:

1. Apply power to the PC host.
2. Apply power to the Emulator assembly.

   If power is to come from the PC, press the power rocker switch to the HOST position (left). If power is to come from the external power supply, press the rocker switch to the EXT position (right).
3. Apply power to the target application circuit.

## Power Down System Components

Reverse the above steps to properly power down the system.

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

### Parallel Bus Port

(Rear panel) The host PC connects to the emulator via this port.

### RS-232 Port (PICMASTER Only)

Reserved for factory use. It has no user function.

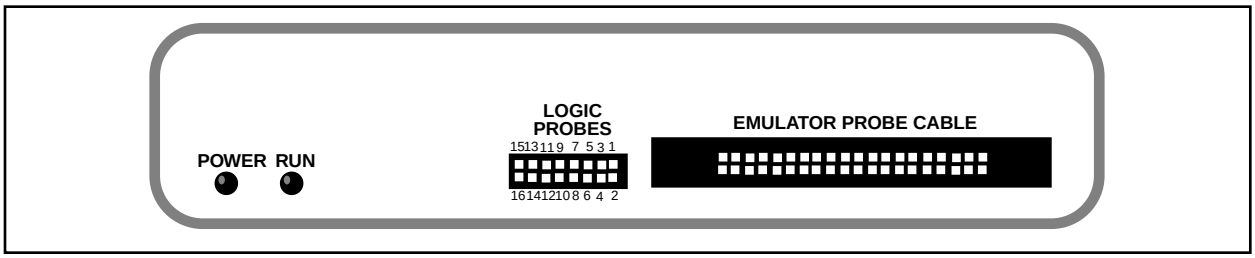



**Figure C.1 PICMASTER POD FRONT PANEL**

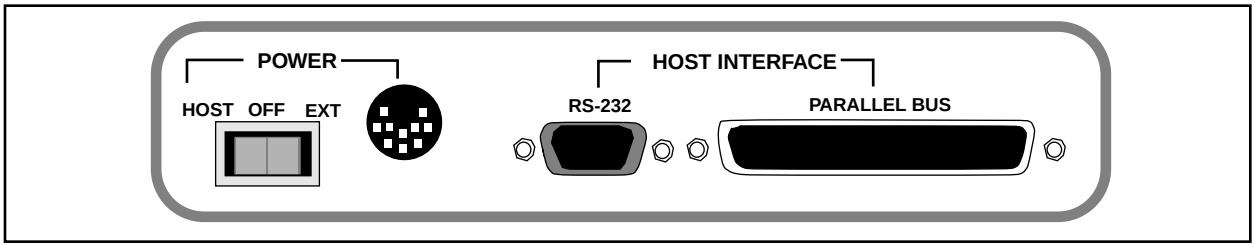



**Figure C.2 PICMASTER POD REAR PANEL**

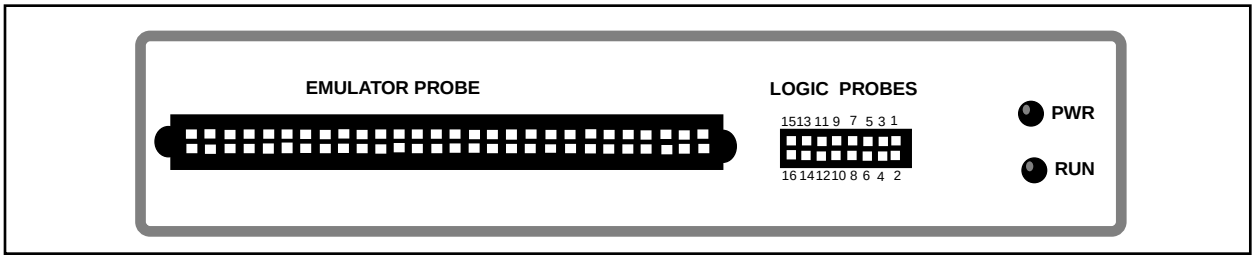



**Figure C.3 PICMASTER CE POD FRONT PANEL**

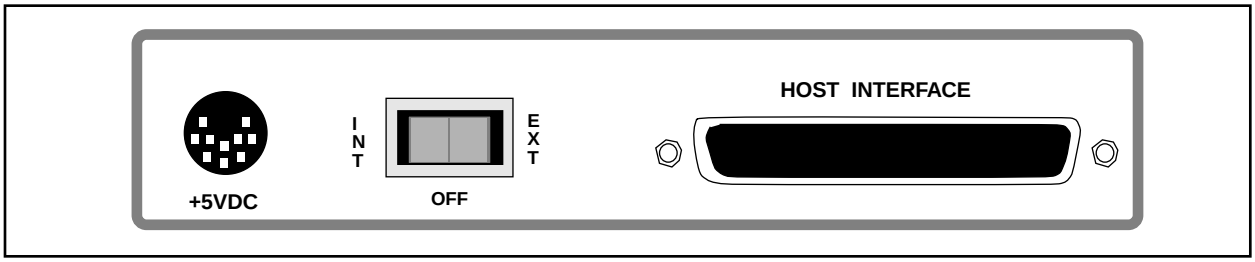



**Figure C.4 PICMASTER CE POD REAR PANEL**

# Logic Probes

This 16 pin connector at the front provides 8 trace inputs, one external break input, one trigger output, power, ground and several no-connects.

## Logic Probe Specifications

| | | | |
|---|---|---|---|
| Logic inputs: | | | |
| (TR [0:7]) | $V_{IH}$ | = | 3.2V min |
| | $V_{IL}$ | = | 1.8V max |
| Trigger outputs: | | | |
| (TRIGO, TRIGT) | $V_{OH}$ | = | 2.4V min |
| | $V_{OL}$ | = | 0.4V max |
| Break inputs: | | | |
| (BRKT, BRK) | $V_{IH}$ | = | 3.2V min |
| | $V_{IL}$ | = | 1.8V max |

**Table C.1 LOGIC PROBE PINOUT DESCRIPTION**

| Pin # | Name | Description |
|---|---|---|
| 1 | $V_{DD}$ | +5V (output). This is a direct power output from the pod.<br>It doesn't have over-current protection.<br>The user should be cautious about using this supply.<br>Do not exceed 250mA if powered from the Host Computer or 500mA if powered from external supply. |
| 2 | N/C | No Connect |
| 3 | Tr0 (I) | Bit 0 of the 8 external trace inputs |
| 4 | N/C | No Connect |
| 5 | Tr2 (I) | Bit 2 of the 8 external trace inputs |
| 6 | Tr1 (I) | Bit 1 of the 8 external trace inputs |
| 7 | Tr4 (I) | Bit 4 of the 8 external trace inputs |
| 8 | Tr3 (I) | Bit 3 of the 8 external trace inputs |
| 9 | Tr6 (I) | Bit 6 of the 8 external trace inputs |
| 10 | Tr5 (I) | Bit 5 of the 8 external trace inputs |
| 11 | BRKT (I) | External Trace Halt Signal: A rising edge on this line will halt trace buffer without halting the processor. For more information refer to Trace section. |
| 12 | Tr7 (I) | Bit 7 of the 8 external trace inputs |
| 13 | TRIGO (O) | Trigger Output Signal. Can be used for triggering Oscilloscopes, accurate time measurement, etc. For more details refer to Trigger Output section. |
| 14 | BRK (I) | External Break Input Signal. Halts the processor either on a rising or falling edge. The edge is programmable in software. |
| 15 | TRIGT (O) | This is the same clock applied to the trace buffer to incrementing trace address. This clock can be used to trigger instruments like Logic Analyzers |
| 16 | GND | Common Ground |

# INTERFACE CARD

The half size PC interface card (shown in Figure C.5) has DIP switches for base address selection. This selection is made at the time of installation. Base addressed available use as shown in Table C.2.

**Note:** Some older PC interface cards have incorrect silk screen notation. Please follow this document.

**Table C.2 Base Address Selection**

| Base Address | DIP Switch Setting | | | |
|---|---|---|---|---|
| | 1 | 2 | 3 | 4 |
| 200 | ON | ON | ON | ON |
| 208 | ON | OFF | ON | ON |
| 210 | ON | ON | OFF | ON |
| 218 | ON | OFF | OFF | ON |
| 300* | OFF | ON | ON | OFF |
| 308 | OFF | OFF | ON | OFF |
| 310 | OFF | ON | OFF | OFF |
| 318 | OFF | OFF | OFF | OFF |

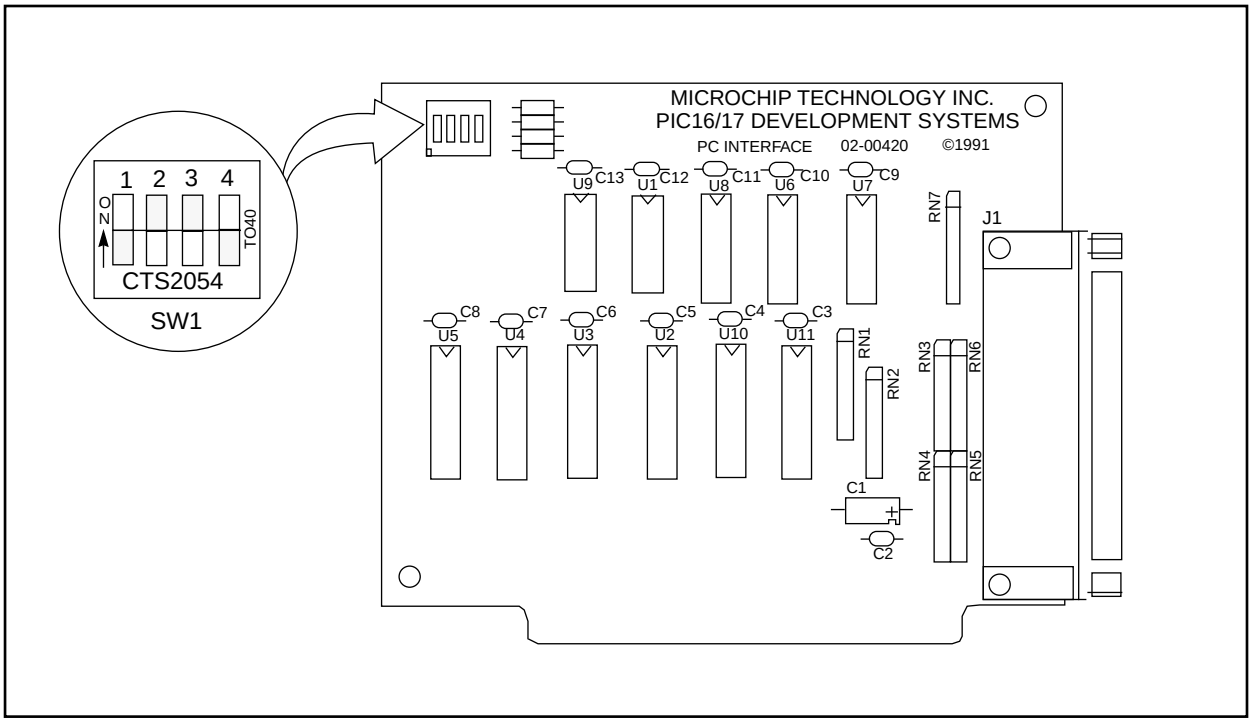



**Figure C.5 PC Interface Card**

# Index



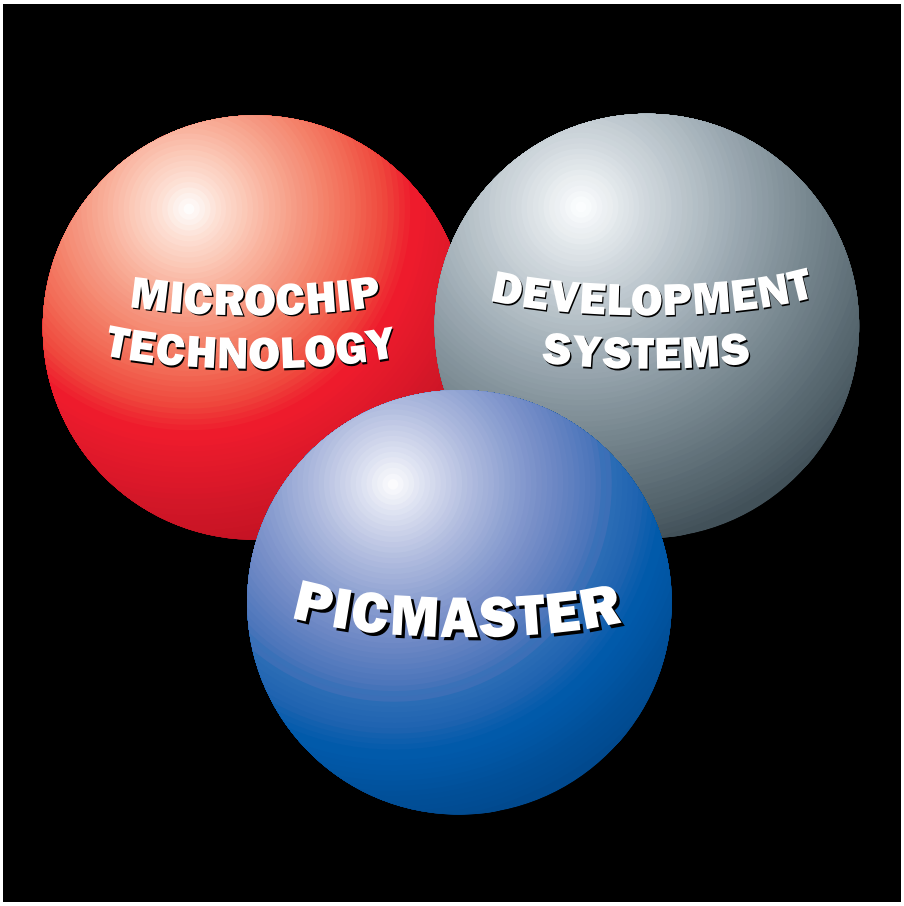
 

# INDEX

# INDEX

# INDEX

# PICMASTER®
## IN-CIRCUIT EMULATOR




# USER'S GUIDE

MICROCHIP

# WORLDWIDE SALES & SERVICE

## AMERICAS

**Corporate Office**
Microchip Technology Inc.
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 602 786-7200 Fax: 602 786-7277
*Technical Support: 602 786-7627*
*Web:* http://www.microchip.com

**Atlanta**
Microchip Technology Inc.
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 770 640-0034 Fax: 770 640-0307

**Boston**
Microchip Technology Inc.
5 Mount Royal Avenue
Marlborough, MA 01752
Tel: 508 480-9990 Fax: 508 480-8575

**Chicago**
Microchip Technology Inc.
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 708 285-0071 Fax: 708 285-0075

**Dallas**
Microchip Technology Inc.
14651 Dallas Parkway, Suite 816
Dallas, TX 75240-8809
Tel: 214 991-7177 Fax: 214 991-8588

**Dayton**
Microchip Technology Inc.
Suite 150
Two Prestige Place
Miamisburg, OH 45342
Tel: 513 291-1654 Fax: 513 291-9175

**Los Angeles**
Microchip Technology Inc.
18201 Von Karman, Suite 1090
Irvine, CA 92715
Tel: 714 263-1888 Fax: 714 263-1338

## AMERICAS *(continued)*

**New York**
Microchip Technology Inc.
150 Motor Parkway, Suite 416
Hauppauge, NY 11788
Tel: 516 273-5305 Fax: 516 273-5335

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

## CANADA

Microchip Technology Inc.
5925 Airport Road, Suite 200
Mississauga, Ontario L4V 1W1, Canada
Tel: 905 405-6279 Fax: 905 405-6253

## ASIA/PACIFIC

**Hong Kong**
Microchip Technology
Unit No. 3002-3004, Tower 1
Metroplaza
223 Hing Fong Road
Kwai Fong, N.T. Hong Kong
Tel: 852 2 401 1200 Fax: 852 2 401 3431

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

**Singapore**
Microchip Technology
200 Middle Road
#10-03 Prime Centre
Singapore 188980
Tel: 65 334 8870 Fax: 65 334 8850

**Taiwan**
Microchip Technology
10F-1C 207
Tung Hua North Road
Taipei, Taiwan, ROC
Tel: 886 2 717 7175 Fax: 886 2 545 0139

## EUROPE

**United Kingdom**
Arizona Microchip Technology Ltd.
Unit 6, The Courtyard
Meadow Bank, Furlong Road
Bourne End, Buckinghamshire SL8 5AJ
Tel: 44 1 628 850303 Fax: 44 1 628 850178

**France**
Arizona Microchip Technology SARL
Zone Industrielle de la Bonde
2 Rue du Buisson aux Fraises
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 Muenchen, 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 39 689 9939 Fax: 39 39 689 9883

## JAPAN

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

4/16/96

MICROCHIP