Performance Pak for the
MPLAB® REAL ICE™
In-Circuit Emulator
User’s Guide
Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip’s Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks
The Microchip name and logo, the Microchip logo, AnyRate, dsPIC, FlashFlex, flexPWR, Heldo, JukeBox, KeeLoq, KeeLoq logo, Kleer, LANCheck, LINK MD, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC32 logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, ETHERSYNCH, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and QUIET-WIRE are registered trademarks of Microchip Technology Incorporated in the U.S.A.


SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademarks of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

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

© 2016, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.
ISBN: 978-1-5224-0841-3
Object of Declaration: Performance Pak for MPLAB® REAL ICE™ In-Circuit Emulator

EU Declaration of Conformity

Manufacturer: Microchip Technology Inc.
2355 W. Chandler Blvd.
Chandler, Arizona, 85224-6199
USA

This declaration of conformity is issued by the manufacturer.

The development/evaluation tool is designed to be used for research and development in a laboratory environment. This development/evaluation tool is not a Finished Appliance, nor is it intended for incorporation into Finished Appliances that are made commercially available as single functional units to end users under EU EMC Directive 2004/108/EC and as supported by the European Commission’s Guide for the EMC Directive 2004/108/EC (8th February 2010).

This development/evaluation tool complies with EU RoHS2 Directive 2011/65/EU.

This development/evaluation tool, when incorporating wireless and radio-telecom functionality, is in compliance with the essential requirement and other relevant provisions of the R&TTE Directive 1999/5/EC and the FCC rules as stated in the declaration of conformity provided in the module datasheet and the module product page available at www.microchip.com.

For information regarding the exclusive, limited warranties applicable to Microchip products, please see Microchip’s standard terms and conditions of sale, which are printed on our sales documentation and available at www.microchip.com.

Signed for and on behalf of Microchip Technology Inc. at Chandler, Arizona, USA

Derek Carlson
VP Development Tools

12-Sep-14
Date
# Table of Contents

**Preface** ....................................................................................................................... 7

**Chapter 1. Performance Pak Overview**
- 1.1 Introduction ................................................................................................... 11
- 1.2 Emulator-to-Target Connections ................................................................. 12
- 1.3 Target Communication Connections ........................................................... 13
- 1.4 Isolator Connections ..................................................................................... 13
- 1.5 SPI Trace ..................................................................................................... 14
- 1.6 Target Connection Circuitry .......................................................................... 15
- 1.7 Target Circuit Design Precautions ............................................................... 16

**Chapter 2. Performance Pak Hardware**
- 2.1 Introduction ................................................................................................... 17
- 2.2 High-Speed Driver Board ............................................................................. 18
- 2.3 High-Speed Receiver Board ......................................................................... 19
- 2.4 LVDS Cables and Target Pinout ................................................................. 21

**Appendix A. Revision History** ................................................................................. 23

**Support** ....................................................................................................................... 25

**Glossary** ....................................................................................................................... 29

**Index** ............................................................................................................................. 49

**Worldwide Sales and Service** .................................................................................... 50
Preface

NOTICE TO CUSTOMERS

All documentation becomes dated, and this manual is no exception. Microchip tools and
documentation are constantly evolving to meet customer needs, so some actual dialogs
and/or tool descriptions may differ from those in this document. Please refer to our web site
(www.microchip.com) to obtain the latest documentation available.

Documents are identified with a “DS” number. This number is located on the bottom of each
page, in front of the page number. The numbering convention for the DS number is
“DSXXXXXXXXA”, where “XXXXXXXX” is the document number and “A” is the revision level
of the document.

For the most up-to-date information on development tools, see the MPLAB® X IDE help. Select
the Help menu, and then Topics to open a list of available help files.

INTRODUCTION

This chapter contains general information that will be helpful to know before using the
Performance Pak for the MPLAB® REAL ICE™ In-Circuit Emulator.

Items discussed here include:

• Document Layout
• Conventions Used in this Guide
• Recommended Reading
This document describes how to use the Performance Pak as a development tool to emulate and debug firmware on a target board, as well as how to program devices. The document is organized as follows:

- **Chapter 1: Performance Pak Overview** – information on how to connect the hardware components of the Performance Pak. Also descriptions of hardware configurations and their use.
- **Chapter 2: Performance Pak Hardware** – specifications for the hardware items included in the Performance Pak.

**CONVENTIONS USED IN THIS GUIDE**

The following conventions may appear in this documentation:

<table>
<thead>
<tr>
<th>TABLE 1: DOCUMENTATION CONVENTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
</tr>
<tr>
<td><strong>Arial font:</strong></td>
</tr>
<tr>
<td>Italic</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Initial caps</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Quotes</td>
</tr>
<tr>
<td>Underlined, italic text with right angle bracket</td>
</tr>
<tr>
<td>Bold</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Text in angle brackets &lt; &gt;</td>
</tr>
<tr>
<td><strong>Courier font:</strong></td>
</tr>
<tr>
<td>Plain</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Italic</td>
</tr>
<tr>
<td>Square brackets [ ]</td>
</tr>
<tr>
<td>Curly brackets and pipe character: {</td>
</tr>
<tr>
<td>Ellipses...</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
RECOMMENDED READING

The following Microchip documents are available and recommended as supplemental reference resources.

Multi-Tool Design Advisory (DS51764)
A small document on guidelines and implementation considerations to ensure proper interfacing to the various development tools.

MPLAB® REAL ICE™ In-Circuit Emulator Help
An online version of the comprehensive emulator user's guide in MPLAB X IDE. Usage, troubleshooting, and hardware specifications are included.

Processor Extension Pak (PEP) and Debug Header Specification (DS50001292), Emulation Extension Pak (EEP) and Emulation Header User's Guide (DS50002243)
These booklets describe how to install and use debug and emulation headers. Headers are used to better debug selected devices using special -ME2/-ICE/-ICD device versions, without the loss of pins or resources. Extension Paks contain headers. See also, the related help files.

Transition Socket Specification (DS51194)
Consult this document for information on transition sockets available for use with headers.

MPLAB® REAL ICE™ Isolation Unit Instruction Sheet (DS50001858), Isolator Unit for the MPLAB® REAL ICE™ In-Circuit Emulator Specification (DS50002529)
These documents show you how to hook up the opto-isolation unit hardware for high-power applications.
Chapter 1. Performance Pak Overview

1.1 INTRODUCTION

The Performance Pak (AC244002) for the MPLAB® REAL ICE™ In-Circuit Emulator provides high-speed/LVDS (Low-voltage Differential Signal) communication between the emulator and a target. The emulator system can be configured to use the Performance Pak for communicating debug and programming instructions to the target. Compared to standard communication methods, this form of communication provides the following features.

- Noise cancellation from the LVDS technology, which supplies the following benefits:
  - Communication speeds greater than 15 MIPS for data capture, runtime watches, and Native trace
  - Longer distances between the emulator and the target
  - Operation in noisy environments
- Two additional pins used for SPI trace

Performance Pak connections to enable the above features are shown in the following sections. Performance Pak hardware details are shown in Chapter 2. “Performance Pak Hardware”.

- Emulator-to-Target Connections
- Target Communication Connections
- Isolator Connections
- SPI Trace
- Target Connection Circuitry
- Target Circuit Design Precautions
1.2 EMULATOR-TO-TARGET CONNECTIONS

The high-speed driver board from the Performance Pak is plugged into the emulator pod to configure the system for this type of communication with the target. The modular cables can be inserted into matching connectors on the high-speed receiver board.

The high-speed receiver board is attached via an 8-pin connector on one of the following:

- target board, with an on-board target device as shown in Figure 1-1
- debug or emulation header, which is then plugged into the target board as shown in Figure 1-2

**FIGURE 1-1: HIGH-SPEED/LVDS CONNECTION – DEVICE WITH ON-BOARD ICE CIRCUITRY**

**FIGURE 1-2: HIGH-SPEED/LVDS CONNECTION – DEBUG OR EMULATION HEADER**

* See also Section 1.4 “Isolator Connections”.

* See also Section 1.4 “Isolator Connections”.
1.3 TARGET COMMUNICATION CONNECTIONS

To use the high-speed driver/receiver board combination, the MPLAB REAL ICE in-circuit emulator is connected to the target device with an 8-pin interface. The pin numbering for the connector is shown from the perspective of the top of the target PC board in Figure 1-3.

**Note:** Connections from the emulator to the target are shown in Section 1.2 “Emulator-to-Target Connections”.

---

FIGURE 1-3: HIGH-SPEED CONNECTION AT TARGET

---

1.4 ISOLATOR CONNECTIONS

For high voltage applications, use the high-speed receiver board with an AC isolator unit to isolate the target, e.g., the Isolator Unit (AC244005) for the MPLAB® REAL ICE™ In-Circuit Emulator. See Section 2.3 “High-Speed Receiver Board” for details.

For more on the Isolator Unit, see the Isolator Unit for the MPLAB® REAL ICE™ In-Circuit Emulator Specification (DS50002529).
1.5 SPI TRACE

Serial Port Interface (SPI) Trace, or simply Serial Trace, is a type of instrumented trace. This means that MPLAB X IDE and MPLAB XC C compiler support is required. Also, the device must have an SPI peripheral.

1.5.1 SPI Trace Connections

For Serial Trace, use the device SPI and pins 7 (DAT) and 8 (CLK). The device is connected to the target using high-speed/LVDS communication hardware which provides the extra lines for clock and data. The device does not have to be operating at high speeds to use this feature.

Figure 1-4 shows the proper connections. As with pins 4 (PGD) and 5 (PGC), do not use pull-up or pull-down resistors, capacitors or diodes (Section 1.7 “Target Circuit Design Precautions”).

FIGURE 1-4: SERIAL TRACE CONNECTIONS

The DAT and CLK lines are intended for use with devices that do not have built-in debug logic that allows tracing (Native trace) to use the PGD/PGC/EMUC/EMUD pins. The DAT line connects to either the target device SPI port SDO1 or SDO2. The CLK line connects to SCK1 or SCK2.

1.5.2 SPI Trace Usage

When you dedicate SPI pins to tracing, any multiplexed function on these pins cannot be used by the application.

For devices with remappable peripheral pins, be aware that the SPI trace macro does not touch any PPS register and does not need to know how the peripheral is mapped to a certain pin – it will write to the SPI1 or SPI2 selected in MPLAB X IDE.

SPI trace does require that you enter the clock speed (in the Properties dialog, click the REAL ICE category, and the Clock options category.)

For more on Instrumented Trace in general and SPI Trace in particular, see the MPLAB® REAL ICE™ In-Circuit Emulator User’s Guide (DS50002085).
1.6 TARGET CONNECTION CIRCUITRY

Figure 1-5 shows the interconnections of the MPLAB REAL ICE in-circuit emulator through the target board connector to a device on the target board. The interconnection is very simple. Any problems experienced are often caused by other connections or components on these critical lines that interfere with emulator operation, as discussed in Section 1.7 “Target Circuit Design Precautions”.

FIGURE 1-5: STANDARD CONNECTION TO TARGET CIRCUITRY

TABLE 1-1: TARGET CONNECTOR AND RELATED DEVICE I/O

<table>
<thead>
<tr>
<th>Pin No.</th>
<th>Device I/O</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VPP/MCLR</td>
<td>The emulator requires access to VPP for programming and debugging the device.</td>
</tr>
<tr>
<td>2</td>
<td>VDD</td>
<td>Target VDD is sensed by the emulator to allow level translation for target low-voltage operation and to detect a device. If the emulator does not sense voltage on its VDD line, it will not connect with the device. <strong>Note:</strong> The emulator DOES NOT provide target power.</td>
</tr>
<tr>
<td>3</td>
<td>VSS</td>
<td>Target VSS is sensed by the emulator. <strong>Note:</strong> The emulator DOES NOT provide target VSS or ground.</td>
</tr>
<tr>
<td>4</td>
<td>PGD</td>
<td>The emulator requires access to PGD and PGC for programming and debugging the device.</td>
</tr>
<tr>
<td>5</td>
<td>PGC</td>
<td></td>
</tr>
</tbody>
</table>

TABLE 1-2: CIRCUITRY ON DEVICE I/O

<table>
<thead>
<tr>
<th>Device I/O</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VPP/MCLR and VDD</td>
<td>A pull-up resistor (minimum 50 kΩ) should be connected from the VPP/MCLR line to VDD so that the line may be strobed low to reset the device.</td>
</tr>
<tr>
<td>XTAL</td>
<td>The target device must be running with an oscillator for the emulator to function as a debugger.</td>
</tr>
<tr>
<td>AVDD, AVSS</td>
<td>Not all devices have the AVDD and AVSS lines, but if they are present on the target device, they must be connected to the appropriate levels in order for the emulator to operate. This also applies to voltage regulator pins (e.g., ENVREG/DISVREG on PIC24FJ MCUs).</td>
</tr>
<tr>
<td>VDD, VSS, AVDD, AVSS</td>
<td>In general, it is recommended per device data sheet that all VDD/AVDD and VSS/AVSS lines are connected to the appropriate levels. For devices with a VCAP pin (e.g., PIC18FXXJ devices), the appropriately-valued capacitor should be placed as close to the VCAP pin as possible.</td>
</tr>
</tbody>
</table>
Table 1-6 shows the active emulator lines with some example components that will prevent the MPLAB REAL ICE in-circuit emulator system from functioning.

**FIGURE 1-6: IMPROPER CIRCUIT COMPONENTS**

- **Do not use capacitors on MCLR:** they will prevent fast transitions of VPP.
- **Do not use pull-ups on PGC/PGD:** they will divide the voltage levels since these lines have 4.7 kΩ pull-down resistors in the emulator.
- **Do not use multiplexing on PGC/PGD:** they are dedicated for communications to the emulator.
- **Do not use capacitors on PGC/PGD:** they will prevent fast transitions on data and clock lines during programming and debug communications.
- **Do not use diodes on PGC/PGD:** they will prevent bidirectional communication between the emulator and the target device.
- **Do not exceed recommended cable lengths:** For acceptable cable lengths, see Section 2.4 “LVDS Cables and Target Pinout”.

Additional design information is available in the *Development Tools Design Advisory* (DS51764).
Chapter 2. Performance Pak Hardware

2.1 INTRODUCTION

The Performance Pak for the MPLAB® REAL ICE™ In-Circuit Emulator consists of the following items:

- High-Speed Driver Board
- High-Speed Receiver Board
- LVDS Cables and Target Pinout

To use this type of communication with a debug or emulation header, you will need a device-specific Extension Pak, which includes an 8-pin connector header that contains the desired -ICE/-ICD or -ME2 device and a standard adapter board (8-pin to 6-pin connection.)

**Note:** You will not need the standard adapter board for high-speed communications. Instead, you will plug the 8-pin connector end of the high-speed receiver board directly into the 8-pin connector of the debug header.

For more on available headers, see Recommended Reading in the “Preface”.

© 2016 Microchip Technology Inc.
2.2 HIGH-SPEED DRIVER BOARD

The high-speed driver board consists of two separate multi-point LVDS (Low Voltage Differential Signaling) transmitters and receivers for clock and data. Multi-point LVDS requires 100 ohm terminations at each driver output and receiver input (per the standard). Multi-point configuration Type-2 receivers are used, as these are intended for control signals or where fail-safe provisions are needed. Even though the standard allows for any combination of drivers, receivers and/or transceivers, up to a total of 32 on the line, only two will be used. The driver board has a port expansion which is controlled by an I\(^2\)C interface for sending/receiving status information to/from the emulator. The high-speed driver board assembly is inserted into the emulator pod via the card guide.

**Note:** The driver board can support data rates up to 100 Mbps. However, due to device speeds, the actual rate is up to 20 Mbps.

**FIGURE 2-1: MODULAR CONNECTORS PINOUT OF HIGH-SPEED DRIVER BOARD**

![Front view of Modular Connectors on HS Driver Board](image1)

![Bottom view of Modular Connectors Pinout on HS Driver Board](image2)

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Function</th>
<th>Pin</th>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LVD+</td>
<td>LV Std Data +</td>
<td>5</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>2</td>
<td>LVD-</td>
<td>LV Std Data -</td>
<td>6</td>
<td>LVC-</td>
<td>LV Std Clock -</td>
</tr>
<tr>
<td>3</td>
<td>LVC+</td>
<td>LV Std Clock +</td>
<td>7</td>
<td>VDD_TGT</td>
<td>VDD on target</td>
</tr>
<tr>
<td>4</td>
<td>LV_VDD</td>
<td>Power</td>
<td>8</td>
<td>VPP_TGT</td>
<td>VPP on target</td>
</tr>
</tbody>
</table>

**J3 Pinout**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Function</th>
<th>Pin</th>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DATAEN+</td>
<td>Std Data Enable +</td>
<td>5</td>
<td>USPID-</td>
<td>*Serial Data -</td>
</tr>
<tr>
<td>2</td>
<td>DATAEN-</td>
<td>Std Data Enable -</td>
<td>6</td>
<td>CLKEN-</td>
<td>Std Clock Enable -</td>
</tr>
<tr>
<td>3</td>
<td>CLKEN+</td>
<td>Std Clock Enable +</td>
<td>7</td>
<td>USPIC+</td>
<td>*Serial Clock +</td>
</tr>
<tr>
<td>4</td>
<td>USPID+</td>
<td>*Serial Data +</td>
<td>8</td>
<td>USPIC-</td>
<td>*Serial Clock -</td>
</tr>
</tbody>
</table>

* Optional - see Section 1.5 “SPI Trace”.
2.3 HIGH-SPEED RECEIVER BOARD

A high-speed receiver board assembly is also required when using LVDS connectivity. This board is a counterpart to the high-speed driver board assembly in the REAL ICE In-Circuit Emulator. When the driver is active on the pod, the receiver is active in the receiver board. Alternatively, when the driver is active on the receiver board, the corresponding receiver is active in the driver board, providing transmitting and receiving capability at both ends. The receiver board contains an 8-pin, 0.100 inch centers header, and is used to connect to the target board or a debug header. The receiver board circuitry may be implemented on the target system to avoid using the receiver board.

FIGURE 2-2: MODULAR CONNECTORS PINOUT OF HIGH-SPEED RECEIVER BOARD

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Function</th>
<th>Pin</th>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DATAEN+</td>
<td>Std Data Enable+</td>
<td>5</td>
<td>USPID-</td>
<td>*Serial Data -</td>
</tr>
<tr>
<td>2</td>
<td>DATAEN-</td>
<td>Std Data Enable-</td>
<td>6</td>
<td>CLKEN-</td>
<td>Std Clock Enable-</td>
</tr>
<tr>
<td>3</td>
<td>CLKEN+</td>
<td>Std Clock Enable+</td>
<td>7</td>
<td>USPIC+</td>
<td>*Serial Clock +</td>
</tr>
<tr>
<td>4</td>
<td>USPID+</td>
<td>*Serial Data +</td>
<td>8</td>
<td>USPIC-</td>
<td>*Serial Clock -</td>
</tr>
</tbody>
</table>

* Optional - see Section 1.5 “SPI Trace”.

FIGURE 2-3: 8-PIN HEADER PINOUT OF HIGH-SPEED RECEIVER BOARD

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Function</th>
<th>Pin</th>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VPP</td>
<td>Power</td>
<td>5</td>
<td>ICSPCLK</td>
<td>Standard Com Clock</td>
</tr>
<tr>
<td>2</td>
<td>VDD_TGT</td>
<td>Power on target</td>
<td>6</td>
<td>AUX</td>
<td>Auxiliary</td>
</tr>
<tr>
<td>3</td>
<td>GND</td>
<td>Ground</td>
<td>7</td>
<td>DAT</td>
<td>*Trace Data</td>
</tr>
<tr>
<td>4</td>
<td>ICSPDAT</td>
<td>Standard Com Data</td>
<td>8</td>
<td>CLK</td>
<td>*Trace Clock</td>
</tr>
</tbody>
</table>

* Optional – see Section 1.5 “SPI Trace”.

Front view of Modular Connectors on HS Receiver Board

Bottom view of Modular Connectors Pinout on HS Receiver Board

Top of HS Rcvr Board
FIGURE 2-4: RECEIVER BOARD SCHEMATIC – ICSPDAT

FIGURE 2-5: RECEIVER BOARD SCHEMATIC – ICSPCLK
2.4 LVDS CABLES AND TARGET PINOUT

The emulator-to-target cable length for proper operation has been tested for this driver/receiver board combination and is shipped in the Performance Pak. The recommended length is 3 feet, while the maximum is 10 feet.

The target board should have the following 8-pin connection pinout to plug into the receiver board.

*Optional – see Section 1.5 “SPI Trace”.
Appendix A. Revision History

Revision A (September 2016)

Initial release of this document as a separate document. Previously, this information was located in the MPLAB® REAL ICE™ In-Circuit Emulator User’s Guide (DS50002085), Part 5 - Emulator Accessories, Chapter 19.
INTRODUCTION

Please refer to the items discussed here for support issues.

- Warranty Registration
- myMicrochip Personalized Notification Service
- The Microchip Web Site
- Microchip Forums
- Customer Support
- About Microchip Technology

DEVELOPMENT TOOL REGISTRATION

Registering your development tool entitles you to receive new product updates. Go to the Microchip web site:


Interim software release are available at the Microchip web site.

myMICROCHIP PERSONALIZED NOTIFICATION SERVICE

Microchip’s personal notification service helps keep customers current on their Microchip products of interest. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool.

Please visit http://www.microchip.com/pcn to begin the registration process and select your preferences to receive personalized notifications. A FAQ and registration details are available on the page, which can be opened by selecting the link above.

When you are selecting your preferences, choosing “Development Systems” will populate the list with available development tools. The main categories of tools are listed below:

- **Compilers** – The latest information on Microchip C compilers, assemblers, linkers and other language tools. These include all MPLAB C compilers; all MPLAB assemblers (including MPASM™ assembler); all MPLAB linkers (including MPLINK™ object linker); and all MPLAB librarians (including MPLIB™ object librarian).
- **Emulators** – The latest information on Microchip in-circuit emulators. This includes the MPLAB REAL ICE™ in-circuit emulator.
- **In-Circuit Debuggers** – The latest information on Microchip in-circuit debuggers. These include the PICkit™ 3 and MPLAB ICD 3 in-circuit debuggers.
- **MPLAB® X IDE** – The latest information on Microchip MPLAB X IDE, the Windows® Integrated Development Environment for development systems tools. This list is focused on the MPLAB X IDE, MPLAB X IDE Project Manager, MPLAB Editor and MPLAB SIM simulator, as well as general editing and debugging features.
• **Programmers** – The latest information on Microchip programmers. These include the device (production) programmers: MPLAB REAL ICE In-Circuit Emulator, MPLAB ICD 3 In-Circuit Debugger, MPLAB PM3; and development (non-production) programmers PICkit 3.
• **Starter/Demo Boards** – These include MPLAB Starter Kit boards, PICDEM demo boards, and various other evaluation boards.

**THE MICROCHIP WEB SITE**

Microchip provides online support via our web site at http://www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

• **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user’s guides and hardware support documents, latest software releases and archived software
• **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip consultant program member listing
• **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

**MICROCHIP FORUMS**

Microchip provides additional online support by way of our web forums at http://www.microchip.com/forums. Forums that are currently available include the following subjects:

• Development Tools
• 8-bit PIC MCUs
• 16-bit PIC MCUs
• 32-bit PIC MCUs

**CUSTOMER SUPPORT**

Users of Microchip products can receive assistance through several channels:

• Distributor or Representative
• Local Sales Office
• Field Application Engineer (FAE)
• Technical Support

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document. See our web site for a complete, up-to-date listing of sales offices.

Technical support is available through the web site at http://support.microchip.com.

Documentation errors or comments may be emailed to docerrors@microchip.com.
ABOUT MICROCHIP TECHNOLOGY

Microchip Technology Inc. is a leading provider of microcontroller and analog semiconductors, providing low-risk product development, lower total system cost and faster time to market for thousands of diverse customer applications worldwide. Headquartered in Chandler, Arizona, Microchip offers outstanding technical support along with dependable delivery and quality.

Voice: (480) 792-7200
Fax: (480) 792-7277
myMicrochip: http://www.microchip.com/pcn
Web site: http://www.microchip.com
Forums: http://www.microchip.com/forums
Support: http://support.microchip.com
Glossary

A

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

Absolute Variable/Function
A variable or function placed at an absolute address using the OCG compiler’s `@address` syntax.

Access Memory
PIC18 Only – Special registers on PIC18 devices that allow access regardless of the setting of the Bank Select Register (BSR).

Access Entry Points
Access entry points provide a way to transfer control across segments to a function which may not be defined at link time. They support the separate linking of boot and secure application segments.

Address
Value that identifies a location in memory.

Alphabetic Character
Alphabetic characters are those characters that are letters of the Roman alphabet (a, b, ..., z, A, B, ..., Z).

Alphanumeric
Alphanumeric characters are comprised of alphabetic characters and decimal digits (0, 1, ..., 9).

ANDed Breakpoints
Set up an ANDed condition for breaking, i.e., breakpoint 1 AND breakpoint 2 must occur at the same time before a program halt. This can only be accomplished if a data breakpoint and a program memory breakpoint occur at the same time.

Anonymous Structure
16-bit C Compiler – An unnamed structure.

PIC18 C Compiler – An unnamed structure that is a member of a C union. The members of an anonymous structure may be accessed as if they were members of the enclosing union. For example, in the following code, `hi` and `lo` are members of an anonymous structure inside the union `caster`.

```c
union castaway
{
  int intval;
  struct {
    char lo;  //accessible as caster.lo
    char hi;  //accessible as caster.hi
  };
} caster;
```
ANSI
American National Standards Institute is an organization responsible for formulating and approving standards in the United States.

Application
A set of software and hardware that may be controlled by a PIC® microcontroller.

Archive/Archiver
An archive/library is a collection of relocatable object modules. It is created by assembling multiple source files to object files, and then using the archiver/librarian to combine the object files into one archive/library file. An archive/library can be linked with object modules and other archives/libraries to create executable code.

ASCII
American Standard Code for Information Interchange is a character set encoding that uses 7 binary digits to represent each character. It includes upper and lowercase letters, digits, symbols and control characters.

Assembly/Assembler
Assembly is a programming language that describes binary machine code in a symbolic form. An assembler is a language tool that translates assembly language source code into machine code.

Assigned Section
A GCC compiler section which has been assigned to a target memory block in the linker command file.

Asynchronously
Multiple events that do not occur at the same time. This is generally used to refer to interrupts that may occur at any time during processor execution.

Asynchronous Stimulus
Data generated to simulate external inputs to a simulator device.

Attribute
GCC Characteristics of variables or functions in a C program which are used to describe machine-specific properties.

Attribute, Section
GCC Characteristics of sections, such as “executable”, “readonly”, or “data” that can be specified as flags in the assembler .section directive.

B
Binary
The base two numbering system that uses the digits 0-1. The rightmost digit counts ones, the next counts multiples of 2, then $2^2 = 4$, etc.

Bookmarks
Use bookmarks to easily locate specific lines in a file.
Select Toggle Bookmarks on the Editor toolbar to add/remove bookmarks. Click other icons on this toolbar to move to the next or previous bookmark.

Breakpoint
Hardware Breakpoint: An event whose execution will cause a halt.
Software Breakpoint: An address where execution of the firmware will halt. Usually achieved by a special break instruction.
Build
Compile and link all the source files for an application.

C

C/C++
C is a general purpose programming language which features economy of expression, modern control flow and data structures, and a rich set of operators. C++ is the object-oriented version of C.

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

Central Processing Unit
The part of a device that is responsible for fetching the correct instruction for execution, decoding that instruction, and then executing that instruction. When necessary, it works in conjunction with the arithmetic logic unit (ALU) to complete the execution of the instruction. It controls the program memory address bus, the data memory address bus, and accesses to the stack.

Clean
Clean removes all intermediary project files, such as object, hex and debug files, for the active project. These files are recreated from other files when a project is built.

COFF
Common Object File Format. An object file of this format contains machine code, debugging and other information.

Command Line Interface
A means of communication between a program and its user based solely on textual input and output.

Compiled Stack
A region of memory managed by the compiler in which variables are statically allocated space. It replaces a software or hardware stack when such mechanisms cannot be efficiently implemented on the target device.

Compiler
A program that translates a source file written in a high-level language into machine code.

Conditional Assembly
Assembly language code that is included or omitted based on the assembly-time value of a specified expression.

Conditional Compilation
The act of compiling a program fragment only if a certain constant expression, specified by a preprocessor directive, is true.

Configuration Bits
Special-purpose bits programmed to set PIC MCU and dsPIC DSC modes of operation. A Configuration bit may or may not be preprogrammed.

Control Directives
Directives in assembly language code that cause code to be included or omitted based on the assembly-time value of a specified expression.

CPU
See Central Processing Unit.
Cross Reference File
A file that references a table of symbols and a list of files that references the symbol. If the symbol is defined, the first file listed is the location of the definition. The remaining files contain references to the symbol.

Data Directives
Data directives are those that control the assembler’s allocation of program or data memory and provide a way to refer to data items symbolically; that is, by meaningful names.

Data Memory
On Microchip MCU and DSC devices, data memory (RAM) is comprised of General Purpose Registers (GPRs) and Special Function Registers (SFRs). Some devices also have EEPROM data memory.

Data Monitor and Control Interface (DMCI)
The Data Monitor and Control Interface, or DMCI, is a tool in MPLAB X IDE. The interface provides dynamic input control of application variables in projects. Application-generated data can be viewed graphically using any of four dynamically-assignable graph windows.

Debug/Debugger
See ICE/ICD.

Debugging Information
Compiler and assembler options that, when selected, provide varying degrees of information used to debug application code. See compiler or assembler documentation for details on selecting debug options.

Deprecated Features
Features that are still supported for legacy reasons, but will eventually be phased out and no longer used.

Device Programmer
A tool used to program electrically programmable semiconductor devices such as microcontrollers.

Digital Signal Controller
A digital signal controller (DSC) is a microcontroller device with digital signal processing capability, i.e., Microchip dsPIC DSC devices.

Digital Signal Processing/Digital Signal Processor
Digital signal processing (DSP) is the computer manipulation of digital signals, commonly analog signals (sound or image) which have been converted to digital form (sampled). A digital signal processor is a microprocessor that is designed for use in digital signal processing.

Directives
Statements in source code that provide control of the language tool’s operation.

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

DWARF
Debug With Arbitrary Record Format. DWARF is a debug information format for ELF files.
**Glossary**

**E**

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

**ELF**
Executable and Linking Format. An object file of this format contains machine code. Debugging and other information is specified in with DWARF. ELF/DWARF provide better debugging of optimized code than COFF.

**Emulation/Emulator**
See ICE/ICD.

**Endianness**
The ordering of bytes in a multi-byte object.

**Environment**
MPLAB PM3 – A folder containing files on how to program a device. This folder can be transferred to a SD/MMC card.

**Epilogue**
A portion of compiler-generated code that is responsible for deallocating stack space, restoring registers and performing any other machine-specific requirement specified in the runtime model. This code executes after any user code for a given function, immediately prior to the function return.

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

**Error/Error File**
An error reports a problem that makes it impossible to continue processing your program. When possible, an error identifies the source file name and line number where the problem is apparent. An error file contains error messages and diagnostics generated by a language tool.

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

**Executable Code**
Software that is ready to be loaded for execution.

**Export**
Send data out of the MPLAB IDE/MPLAB X IDE in a standardized format.

**Expressions**
Combinations of constants and/or symbols separated by arithmetic or logical operators.

**Extended Microcontroller Mode**
In Extended Microcontroller mode, on-chip program memory as well as external memory is available. Execution automatically switches to external if the program memory address is greater than the internal memory space of the PIC18 device.
Extended Mode (PIC18 MCUs)
In Extended mode, the compiler will utilize the extended instructions (i.e., ADDFSR, ADDULNK, CALLW, MOVSF, MOVSS, PUSHL, SUBFSR, and SUBULNK) and the indexed with literal offset addressing.

External Label
A label that has external linkage.

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

External Symbol
A symbol for an identifier which has external linkage. This may be a reference or a definition.

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

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

External RAM
Off-chip Read/Write memory.

F
Fatal Error
An error that will halt compilation immediately. No further messages will be produced.

File Registers
On-chip data memory, including General Purpose Registers (GPRs) and Special Function Registers (SFRs).

Filter
Determine by selection what data is included/excluded in a trace display or data file.

Fixup
The process of replacing object file symbolic references with absolute addresses after relocation by the linker.

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

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

Frame Pointer
A pointer that references the location on the stack that separates the stack-based arguments from the stack-based local variables. Provides a convenient base from which to access local variables and other values for the current function.
Free-Standing
An implementation that accepts any strictly conforming program that does not use complex types and in which the use of the features specified in the library clause (ANSI ‘89 standard clause 7) is confined to the contents of the standard headers <float.h>, <iso646.h>, <limits.h>, <stdarg.h>, <stdbool.h>, <stddef.h> and <stdint.h>.

G
GPR
General Purpose Register. The portion of device data memory (RAM) available for general use.

H
Halt
A stop of program execution. Executing Halt is the same as stopping at a breakpoint.

Heap
An area of memory used for dynamic memory allocation where blocks of memory are allocated and freed in an arbitrary order determined at runtime.

Hex Code/Hex File
Hex code is executable instructions stored in a hexadecimal format code. Hex code is contained in a hex file.

Hexadecimal
The base 16 numbering system that uses the digits 0-9 plus the letters A-F (or a-f). The digits A-F represent hexadecimal digits with values of (decimal) 10 to 15. The rightmost digit counts ones, the next counts multiples of 16, then \(16^2 = 256\), etc.

High Level Language
A language for writing programs that is further removed from the processor than assembly.

I
ICE/ICD
In-Circuit Emulator/In-Circuit Debugger: A hardware tool that debugs and programs a target device. An emulator has more features than an debugger, such as trace.

In-Circuit Emulation/In-Circuit Debug: The act of emulating or debugging with an in-circuit emulator or debugger.

-ICE/-ICD: A device (MCU or DSC) with on-board in-circuit emulation or debug circuitry. This device is always mounted on a header board and used to debug with an in-circuit emulator or debugger.

ICSP
In-Circuit Serial Programming. A method of programming Microchip embedded devices using serial communication and a minimum number of device pins.

IDE
Integrated Development Environment, as in MPLAB IDE/MPLAB X IDE.

Identifier
A function or variable name.

IEEE
Institute of Electrical and Electronics Engineers.
Import
Bring data into the MPLAB IDE/MPLAB X IDE from an outside source, such as from a hex file.

Initialized Data
Data which is defined with an initial value. In C,

```c
int myVar=5;
```

defines a variable which will reside in an initialized data section.

Instruction Set
The collection of machine language instructions that a particular processor understands.

Instructions
A sequence of bits that tells a central processing unit to perform a particular operation and can contain data to be used in the operation.

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

International Organization for Standardization
An organization that sets standards in many businesses and technologies, including computing and communications. Also known as ISO.

Interrupt
A signal to the CPU that suspends the execution of a running application and transfers control to an Interrupt Service Routine (ISR) so that the event may be processed. Upon completion of the ISR, normal execution of the application resumes.

Interrupt Handler
A routine that processes special code when an interrupt occurs.

Interrupt Service Request (IRQ)
An event which causes the processor to temporarily suspend normal instruction execution and to start executing an interrupt handler routine. Some processors have several interrupt request events allowing different priority interrupts.

Interrupt Service Routine (ISR)
Language tools – A function that handles an interrupt.
MPLAB IDE/MPLAB X IDE – User-generated code that is entered when an interrupt occurs. The location of the code in program memory will usually depend on the type of interrupt that has occurred.

Interrupt Vector
Address of an Interrupt Service Routine or interrupt handler.

L

L-value
An expression that refers to an object that can be examined and/or modified. An l-value expression is used on the left-hand side of an assignment.

Latency
The time between an event and its response.

Library/Librarian
See Archive/Archiver.
Linker
A language tool that combines object files and libraries to create executable code, resolving references from one module to another.

Linker Script Files
Linker script files are the command files of a linker. They define linker options and describe available memory on the target platform.

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

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

Little Endian
A data ordering scheme for multibyte data whereby the Least Significant Byte is stored at the lower addresses.

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

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

Loopback Test Board
Used to test the functionality of the MPLAB REAL ICE in-circuit emulator.

LVDS
Low Voltage Differential Signaling. A low noise, low-power, low amplitude method for high-speed (gigabits per second) data transmission over copper wire.

With standard I/O signaling, data storage is contingent upon the actual voltage level. Voltage level can be affected by wire length (longer wires increase resistance, which lowers voltage). But with LVDS, data storage is distinguished only by positive and negative voltage values, not the voltage level. Therefore, data can travel over greater lengths of wire while maintaining a clear and consistent data stream.

Source: http://www.webopedia.com/TERM/L/LVDS.html

M

Machine Code
The representation of a computer program that is actually read and interpreted by the processor. A program in binary machine code consists of a sequence of machine instructions (possibly interspersed with data). The collection of all possible instructions for a particular processor is known as its “instruction set”.

Machine Language
A set of instructions for a specific central processing unit, designed to be usable by a processor without being translated.
Macro
Macro instruction. An instruction that represents a sequence of instructions in abbreviated form.

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

Makefile
Export to a file the instructions to Make the project. Use this file to Make your project outside of MPLAB IDE/MPLAB X IDE, i.e., with a `make`.

Make Project
A command that rebuilds an application, recompiling only those source files that have changed since the last complete compilation.

MCU
Microcontroller Unit. An abbreviation for microcontroller. Also uC.

Memory Model
For C compilers, a representation of the memory available to the application. For the PIC18 C compiler, a description that specifies the size of pointers that point to program memory.

Message
Text displayed to alert you to potential problems in language tool operation. A message will not stop operation.

Microcontroller
A highly integrated chip that contains a CPU, RAM, program memory, I/O ports and timers.

Microcontroller Mode
One of the possible program memory configurations of PIC18 microcontrollers. In Microcontroller mode, only internal execution is allowed. Thus, only the on-chip program memory is available in Microcontroller mode.

Microprocessor Mode
One of the possible program memory configurations of PIC18 microcontrollers. In Microprocessor mode, the on-chip program memory is not used. The entire program memory is mapped externally.

Mnemonics
Text instructions that can be translated directly into machine code. Also referred to as opcodes.

Module
The preprocessed output of a source file after preprocessor directives have been executed. Also known as a translation unit.

MPASM™ Assembler
Microchip Technology’s relocatable macro assembler for PIC microcontroller devices, KeeLoq® devices and Microchip memory devices.

MPLAB Language Tool for Device
Microchip’s C compilers, assemblers and linkers for specified devices. Select the type of language tool based on the device you will be using for your application, e.g., if you will be creating C code on a PIC18 MCU, select the MPLAB C Compiler for PIC18 MCUs.
MPLAB ICD
Microchip in-circuit debugger that works with MPLAB IDE/MPLAB X IDE. See ICE/ICD.

MPLAB IDE/MPLAB X IDE
Microchip’s Integrated Development Environment. MPLAB IDE/MPLAB X IDE comes with an editor, project manager and simulator.

MPLAB PM3
A device programmer from Microchip. Programs PIC18 microcontrollers and dsPIC digital signal controllers. Can be used with MPLAB IDE/MPLAB X IDE or stand-alone. Replaces PRO MATE II.

MPLAB REAL ICE™ In-Circuit Emulator
Microchip’s next-generation in-circuit emulator that works with MPLAB IDE/MPLAB X IDE. See ICE/ICD.

MPLAB SIM
Microchip’s simulator that works with MPLAB IDE/MPLAB X IDE in support of PIC MCU and dsPIC DSC devices.

MPLIB™ Object Librarian
Microchip’s librarian that can work with MPLAB IDE/MPLAB X IDE. MPLIB librarian is an object librarian for use with COFF object modules created using either MPASM assembler (mpasm or mpasmwin v2.0) or MPLAB C18 C Compiler.

MPLINK™ Object Linker
MPLINK linker is an object linker for the Microchip MPASM assembler and the Microchip C18 C compiler. MPLINK linker also may be used with the Microchip MPLIB librarian. MPLINK linker is designed to be used with MPLAB IDE/MPLAB X IDE, though it does not have to be.

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

N
Native Data Size
For Native trace, the size of the variable used in a Watches window must be of the same size as the selected device’s data memory: bytes for PIC18 devices and words for 16-bit devices.

Nesting Depth
The maximum level to which macros can include other macros.

Node
MPLAB IDE/MPLAB X IDE project component.

Non-Extended Mode (PIC18 MCUs)
In Non-Extended mode, the compiler will not utilize the extended instructions nor the indexed with literal offset addressing.

Non Real Time
Refers to the processor at a breakpoint or executing single-step instructions or MPLAB IDE/MPLAB X IDE being run in Simulator mode.

Non-Volatile Storage
A storage device whose contents are preserved when its power is off.
NOP
No Operation. An instruction that has no effect when executed except to advance the program counter.

O
Object Code/Object File
Object code is the machine code generated by an assembler or compiler. An object file is a file containing machine code and possibly debug information. It may be immediately executable or it may be relocatable, requiring linking with other object files, e.g., libraries, to produce a complete executable program.

Object File Directives
Directives that are used only when creating an object file.

Octal
The base 8 number system that only uses the digits 0-7. The rightmost digit counts ones, the next digit counts multiples of 8, then $8^2 = 64$, etc.

Off-Chip Memory
Off-chip memory refers to the memory selection option for the PIC18 device where memory may reside on the target board, or where all program memory may be supplied by the emulator. The Memory tab accessed from Options>Development Mode provides the Off-Chip Memory selection dialog box.

Opcodes
Operational Codes. See Mnemonics.

Operators
Symbols, like the plus sign ‘+’ and the minus sign ‘-‘, that are used when forming well-defined expressions. Each operator has an assigned precedence that is used to determine order of evaluation.

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

P
Pass Counter
A counter that decrements each time an event (such as the execution of an instruction at a particular address) occurs. When the pass count value reaches zero, the event is satisfied. You can assign the Pass Counter to break and trace logic, and to any sequential event in the complex trigger dialog.

PC
Personal Computer or Program Counter.

PC Host
Any PC running a supported Windows operating system.

Persistent Data
Data that is never cleared or initialized. Its intended use is so that an application can preserve data across a device Reset.

Phantom Byte
An unimplemented byte in the dsPIC architecture that is used when treating the 24-bit instruction word as if it were a 32-bit instruction word. Phantom bytes appear in dsPIC hex files.
PIC MCUs
PIC microcontrollers (MCUs) refers to all Microchip microcontroller families.

PICkit 2 and 3
Microchip’s developmental device programmers with debug capability through Debug Express. See the Readme files for each tool to see which devices are supported.

Plug-ins
The MPLAB IDE/MPLAB X IDE has both built-in components and plug-in modules to configure the system for a variety of software and hardware tools. Several plug-in tools may be found under the Tools menu.

Pod
The enclosure for an in-circuit emulator or debugger. Other names are “Puck”, if the enclosure is round, and “Probe”, not be confused with logic probes.

Power-on-Reset Emulation
A software randomization process that writes random values in data RAM areas to simulate uninitialized values in RAM upon initial power application.

Pragma
A directive that has meaning to a specific compiler. Often a pragma is used to convey implementation-defined information to the compiler.

Precedence
Rules that define the order of evaluation in expressions.

Production Programmer
A production programmer is a programming tool that has resources designed in to program devices rapidly. It has the capability to program at various voltage levels and completely adheres to the programming specification. Programming a device as fast as possible is of prime importance in a production environment where time is of the essence as the application circuit moves through the assembly line.

Profile
For MPLAB SIM simulator, a summary listing of executed stimulus by register.

Program Counter
The location that contains the address of the instruction that is currently executing.

Program Counter Unit
16-bit assembler – A conceptual representation of the layout of program memory. The program counter increments by 2 for each instruction word. In an executable section, 2 program counter units are equivalent to 3 bytes. In a read-only section, 2 program counter units are equivalent to 2 bytes.

Program Memory
MPLAB IDE/MPLAB X IDE – The memory area in a device where instructions are stored. Also, the memory in the emulator or simulator containing the downloaded target application firmware.

16-bit assembler/compiler – The memory area in a device where instructions are stored.

Project
A project contains the files needed to build an application (source code, linker script files, etc.) along with their associations to various build tools and build options.
Prologue
A portion of compiler-generated code that is responsible for allocating stack space, preserving registers and performing any other machine-specific requirement specified in the runtime model. This code executes before any user code for a given function.

Prototype System
A term referring to a user’s target application, or target board.

Psect
The OCG equivalent of a GCC section, short for program section. A block of code or data which is treated as a whole by the linker.

PWM Signals
Pulse Width Modulation Signals. Certain PIC MCU devices have a PWM peripheral.

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

R
Radix
The number base, hex, or decimal, used in specifying an address.

RAM
Random Access Memory (Data Memory). Memory in which information can be accessed in any order.

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

Read Only Memory
Memory hardware that allows fast access to permanently stored data but prevents addition to or modification of the data.

Real Time
When an in-circuit emulator or debugger is released from the Halt state, the processor runs in Real Time mode and behaves exactly as the normal chip would behave. In Real Time mode, the real time trace buffer of an emulator is enabled and constantly captures all selected cycles, and all break logic is enabled. In an in-circuit emulator or debugger, the processor executes in real time until a valid breakpoint causes a halt, or until the user halts the execution.

In the simulator, real time simply means execution of the microcontroller instructions as fast as they can be simulated by the host CPU.

Recursive Calls
A function that calls itself, either directly or indirectly.

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

Reentrant
A function that may have multiple, simultaneously active instances. This may happen due to either direct or indirect recursion or through execution during interrupt processing.
Relaxation
The process of converting an instruction to an identical, but smaller instruction. This is useful for saving on code size. MPLAB XC16 currently knows how to relax a CALL instruction into an RCALL instruction. This is done when the symbol that is being called is within +/- 32k instruction words from the current instruction.

Relocatable
An object whose address has not been assigned to a fixed location in memory.

Relocatable Section
16-bit assembler – A section whose address is not fixed (absolute). The linker assigns addresses to relocatable sections through a process called relocation.

Relocation
A process performed by the linker in which absolute addresses are assigned to relocatable sections and all symbols in the relocatable sections are updated to their new addresses.

ROM
Read Only Memory (Program Memory). Memory that cannot be modified.

Run
The command that releases the emulator from halt, allowing it to run the application code and change or respond to I/O in real time.

Run-time Model
Describes the use of target architecture resources.

Runtime Watch
A Watch window where the variables change in as the application is run. See individual tool documentation to determine how to set up a runtime watch. Not all tools support runtime watches.

S
Scenario
For MPLAB SIM simulator, a particular setup for stimulus control.

Section
The GCC equivalent of an OCG psect. A block of code or data which is treated as a whole by the linker.

Section Attribute
A GCC characteristic ascribed to a section (e.g., an access section).

Sequenced Breakpoints
Breakpoints that occur in a sequence. Sequence execution of breakpoints is bottom-up; the last breakpoint in the sequence occurs first.

Serialized Quick Turn Programming
Serialization allows you to program a serial number into each microcontroller device that the Device Programmer programs. This number can be used as an entry code, password or ID number.

Shell
The MPASM assembler shell is a prompted input interface to the macro assembler. There are two MPASM assembler shells: one for the DOS version and one for the Windows operating system version.
Simulator
A software program that models the operation of devices.

Single Step
This command steps though code, one instruction at a time. After each instruction, MPLAB IDE/MPLAB X IDE updates register windows, watch variables, and status displays so you can analyze and debug instruction execution. You can also single step C compiler source code, but instead of executing single instructions, MPLAB IDE/MPLAB X IDE will execute all assembly level instructions generated by the line of the high level C statement.

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

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

Source Code
The form in which a computer program is written by the programmer. Source code is written in a formal programming language which can be translated into machine code or executed by an interpreter.

Source File
An ASCII text file containing source code.

Special Function Registers (SFRs)
The portion of data memory (RAM) dedicated to registers that control I/O processor functions, I/O status, timers or other modes or peripherals.

SQTP
See Serialized Quick Turn Programming.

Stack, Hardware
Locations in PIC microcontroller where the return address is stored when a function call is made.

Stack, Software
Memory used by an application for storing return addresses, function parameters, and local variables. This memory is dynamically allocated at runtime by instructions in the program. It allows for reentrant function calls.

Stack, Compiled
A region of memory managed and allocated by the compiler in which variables are statically assigned space. It replaces a software stack when such mechanisms cannot be efficiently implemented on the target device. It precludes reentrancy.

MPLAB Starter Kit for Device
Microchip’s starter kits contains everything needed to begin exploring the specified device. View a working application and then debug and program your own changes.
Static RAM or SRAM
Static Random Access Memory. Program memory you can read/write on the target board that does not need refreshing frequently.

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

Step Into
This command is the same as Single Step. Step Into (as opposed to Step Over) follows a CALL instruction into a subroutine.

Step Over
Step Over allows you to debug code without stepping into subroutines. When stepping over a CALL instruction, the next breakpoint will be set at the instruction after the CALL. If for some reason the subroutine gets into an endless loop or does not return properly, the next breakpoint will never be reached. The Step Over command is the same as Single Step except for its handling of CALL instructions.

Step Out
Step Out allows you to step out of a subroutine which you are currently stepping through. This command executes the rest of the code in the subroutine and then stops execution at the return address to the subroutine.

Stimulus
Input to the simulator, i.e., data generated to exercise the response of simulation to external signals. Often the data is put into the form of a list of actions in a text file. Stimulus may be asynchronous, synchronous (pin), clocked and register.

Stopwatch
A counter for measuring execution cycles.

Storage Class
Determines the lifetime of the memory associated with the identified object.

Storage Qualifier
Indicates special properties of the objects being declared (e.g., const).

Symbol
A symbol is a general purpose mechanism for describing the various pieces which comprise a program. These pieces include function names, variable names, section names, file names, struct/enum/union tag names, etc. Symbols in MPLAB IDE/MPLAB X IDE refer mainly to variable names, function names and assembly labels. The value of a symbol after linking is its value in memory.

Symbol, Absolute
Represents an immediate value such as a definition through the assembly .equ directive.

System Window Control
The system window control is located in the upper left corner of windows and some dialogs. Clicking on this control usually pops up a menu that has the items “Minimize,” “Maximize,” and “Close.”
T
Target
Refers to user hardware.

Target Application
Software residing on the target board.

Target Board
The circuitry and programmable device that makes up the target application.

Target Processor
The microcontroller device on the target application board.

Template
Lines of text that you build for inserting into your files at a later time. The MPLAB Editor stores templates in template files.

Tool Bar
A row or column of icons that you can click on to execute MPLAB IDE/MPLAB X IDE functions.

Trace
An emulator or simulator function that logs program execution. The emulator logs program execution into its trace buffer which is uploaded to the MPLAB IDE/MPLAB X IDE trace window.

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

Trace Macro
A macro that will provide trace information from emulator data. Since this is a software trace, the macro must be added to code, the code must be recompiled or reassembled, and the target device must be programmed with this code before trace will work.

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

Trigraphs
Three-character sequences, all starting with ??, that are defined by ISO C as replacements for single characters.

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

Uninitialized Data
Data which is defined without an initial value. In C,

```
int myVar;
```

defines a variable which will reside in an uninitialized data section.

Upload
The Upload function transfers data from a tool, such as an emulator or programmer, to the host PC or from the target board to the emulator.
USB
Universal Serial Bus. An external peripheral interface standard for communication between a computer and external peripherals over a cable using bi-serial transmission. USB 1.0/1.1 supports data transfer rates of 12 Mbps. Also referred to as high-speed USB, USB 2.0 supports data rates up to 480 Mbps.

V
Vector
The memory locations that an application will jump to when either a Reset or interrupt occurs.

Volatile
A variable qualifier which prevents the compiler applying optimizations that affect how the variable is accessed in memory.

W
Warning
MPLAB IDE/MPLAB X IDE – An alert that is provided to warn you of a situation that would cause physical damage to a device, software file, or equipment.

16-bit assembler/compiler – Warnings report conditions that may indicate a problem, but do not halt processing.

Watch Variable
A variable that you may monitor during a debugging session in a Watches window.

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

Watchdog Timer (WDT)
A timer on a PIC microcontroller that resets the processor after a selectable length of time. The WDT is enabled or disabled and set up using Configuration bits.

Workbook
For MPLAB SIM stimulator, a setup for generation of SCL stimulus.
### Index

<table>
<thead>
<tr>
<th>A</th>
<th>About Microchip Technology .......................................................... 27</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AC244002 .................................................................................. 11</td>
</tr>
<tr>
<td></td>
<td>AC244005 .................................................................................. 13</td>
</tr>
<tr>
<td>C</td>
<td>Cable Lengths ............................................................................. 21</td>
</tr>
<tr>
<td></td>
<td>CLK ............................................................................................ 13, 14</td>
</tr>
<tr>
<td></td>
<td>Clock Speed ................................................................................. 14</td>
</tr>
<tr>
<td></td>
<td>Customer Support .......................................................................... 26</td>
</tr>
<tr>
<td>D</td>
<td>DAT .............................................................................................. 13, 14</td>
</tr>
<tr>
<td></td>
<td>Debug Header Specification .......................................................... 9</td>
</tr>
<tr>
<td></td>
<td>Documentation Conventions .......................................................... 8</td>
</tr>
<tr>
<td></td>
<td>Layout .......................................................................................... 8</td>
</tr>
<tr>
<td></td>
<td>Driver Board, High-Speed ............................................................ 18</td>
</tr>
<tr>
<td>E</td>
<td>EMUC, EMUD .................................................................................. 14</td>
</tr>
<tr>
<td>H</td>
<td>Header Specification ..................................................................... 9</td>
</tr>
<tr>
<td></td>
<td>High-Speed Communication ............................................................ 11</td>
</tr>
<tr>
<td></td>
<td>Driver Board ................................................................................. 18</td>
</tr>
<tr>
<td></td>
<td>Receiver Board .............................................................................. 19</td>
</tr>
<tr>
<td>I</td>
<td>Instrumented Trace - SPI .............................................................. 14, 19, 21</td>
</tr>
<tr>
<td></td>
<td>Internet Address, Microchip ......................................................... 26</td>
</tr>
<tr>
<td>L</td>
<td>LVDS ............................................................................................ 11, 18, 19</td>
</tr>
<tr>
<td>M</td>
<td>myMicrochip Personalized Notification Service ............................. 25</td>
</tr>
<tr>
<td>P</td>
<td>Performance Pak Hardware ............................................................. 17</td>
</tr>
<tr>
<td></td>
<td>Performance Pak Overview ............................................................. 11</td>
</tr>
<tr>
<td></td>
<td>PGC, PGD ...................................................................................... 13, 14, 15</td>
</tr>
<tr>
<td></td>
<td>PIM ............................................................................................... 12</td>
</tr>
<tr>
<td></td>
<td>Pull-up resistor on MCLR .............................................................. 15</td>
</tr>
<tr>
<td>R</td>
<td>Reading, Recommended .................................................................... 9</td>
</tr>
<tr>
<td></td>
<td>Readme .......................................................................................... 9</td>
</tr>
<tr>
<td></td>
<td>Receiver Board, High-Speed ......................................................... 19</td>
</tr>
<tr>
<td></td>
<td>Revision History ........................................................................... 23</td>
</tr>
<tr>
<td>S</td>
<td>Serial Trace .................................................................................. 14</td>
</tr>
<tr>
<td></td>
<td>SPI Trace ..................................................................................... 14, 19, 21</td>
</tr>
<tr>
<td>T</td>
<td>Target Connection Circuitry ........................................................... 15</td>
</tr>
<tr>
<td></td>
<td>SPI ................................................................................................ 14</td>
</tr>
<tr>
<td></td>
<td>Transition Socket Specification ..................................................... 9</td>
</tr>
<tr>
<td>U</td>
<td>USB .............................................................................................. 47</td>
</tr>
<tr>
<td>V</td>
<td>Vdd, Vss ....................................................................................... 13, 15</td>
</tr>
<tr>
<td></td>
<td>Vpp/MCLR .................................................................................... 13, 15</td>
</tr>
<tr>
<td>W</td>
<td>Watchdog Timer ............................................................................ 47</td>
</tr>
<tr>
<td></td>
<td>Web Site, Microchip ....................................................................... 26</td>
</tr>
</tbody>
</table>
### Worldwide Sales and Service

#### AMERICAS

<table>
<thead>
<tr>
<th>City</th>
<th>Address</th>
<th>Telephone</th>
<th>Facsimile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corporate Office</td>
<td>2355 West Chandler Blvd. Chandler, AZ 85224-6199</td>
<td>Tel: 480-792-7200</td>
<td>Fax: 480-792-7277</td>
</tr>
<tr>
<td>Technical Support</td>
<td><a href="http://www.microchip.com/support">http://www.microchip.com/support</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Web Address</td>
<td><a href="http://www.microchip.com">www.microchip.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Atlanta</td>
<td>Duluth, GA</td>
<td>Tel: 678-957-9614</td>
<td>Fax: 678-957-1455</td>
</tr>
<tr>
<td>Austin, TX</td>
<td>Tel: 512-257-3370</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boston</td>
<td>Westborough, MA</td>
<td>Tel: 774-760-0087</td>
<td>Fax: 774-760-0088</td>
</tr>
<tr>
<td>Chicago</td>
<td>Itasca, IL</td>
<td>Tel: 630-285-0071</td>
<td>Fax: 630-285-0075</td>
</tr>
<tr>
<td>Cleveland</td>
<td>Independence, OH</td>
<td>Tel: 216-447-0464</td>
<td>Fax: 216-447-0643</td>
</tr>
<tr>
<td>Dallas</td>
<td>Addison, TX</td>
<td>Tel: 972-818-7423</td>
<td>Fax: 972-818-2924</td>
</tr>
<tr>
<td>Detroit</td>
<td>Novi, MI</td>
<td>Tel: 248-848-4000</td>
<td></td>
</tr>
<tr>
<td>Houston, TX</td>
<td>Tel: 281-894-5983</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Indianapolis</td>
<td>Noblesville, IN</td>
<td>Tel: 317-773-8323</td>
<td>Fax: 317-773-5453</td>
</tr>
<tr>
<td>Los Angeles</td>
<td>Mission Viejo, CA</td>
<td>Tel: 949-462-9523</td>
<td>Fax: 949-462-9608</td>
</tr>
<tr>
<td>New York, NY</td>
<td>Tel: 631-435-6000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>San Jose, CA</td>
<td>Tel: 408-735-9110</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Canada - Toronto</td>
<td>Tel: 905-695-1980</td>
<td>Fax: 905-695-2078</td>
<td></td>
</tr>
</tbody>
</table>

#### ASIA/PACIFIC

<table>
<thead>
<tr>
<th>Region</th>
<th>City</th>
<th>Address</th>
<th>Telephone</th>
<th>Facsimile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hong Kong</td>
<td>Tel: 852-2943-5100</td>
<td>Fax: 852-2401-3431</td>
<td></td>
<td></td>
</tr>
<tr>
<td>China - Beijing</td>
<td>Tel: 86-10-8569-7000</td>
<td>Fax: 86-10-8528-2104</td>
<td></td>
<td></td>
</tr>
<tr>
<td>China - Chengdu</td>
<td>Tel: 86-28-8665-5511</td>
<td>Fax: 86-28-8665-7889</td>
<td></td>
<td></td>
</tr>
<tr>
<td>China - Chongqing</td>
<td>Tel: 86-23-8960-9588</td>
<td>Fax: 86-23-8960-9500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>China - Dongguan</td>
<td>Tel: 86-769-8702-9880</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>China - Guangzhou</td>
<td>Tel: 86-20-8755-8029</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>China - Hangzhou</td>
<td>Tel: 86-571-8792-8115</td>
<td>Fax: 86-571-8792-8116</td>
<td></td>
<td></td>
</tr>
<tr>
<td>China - Hong Kong SAR</td>
<td>Tel: 852-2943-5100</td>
<td>Fax: 852-2401-3431</td>
<td></td>
<td></td>
</tr>
<tr>
<td>China - Nanjing</td>
<td>Tel: 86-25-8473-2460</td>
<td>Fax: 86-25-8473-2470</td>
<td></td>
<td></td>
</tr>
<tr>
<td>China - Qingdao</td>
<td>Tel: 86-532-8502-7355</td>
<td>Fax: 86-532-8502-7205</td>
<td></td>
<td></td>
</tr>
<tr>
<td>China - Shanghai</td>
<td>Tel: 86-21-5407-5533</td>
<td>Fax: 86-21-5407-5066</td>
<td></td>
<td></td>
</tr>
<tr>
<td>China - Shenyang</td>
<td>Tel: 86-24-2334-2829</td>
<td>Fax: 86-24-2334-2393</td>
<td></td>
<td></td>
</tr>
<tr>
<td>China - Shenzhen</td>
<td>Tel: 86-755-8864-2200</td>
<td>Fax: 86-755-8203-1760</td>
<td></td>
<td></td>
</tr>
<tr>
<td>China - Wuhan</td>
<td>Tel: 86-27-5980-5300</td>
<td>Fax: 86-27-5980-5118</td>
<td></td>
<td></td>
</tr>
<tr>
<td>China - Xian</td>
<td>Tel: 86-29-8833-7252</td>
<td>Fax: 86-29-8833-7256</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### EUROPE

<table>
<thead>
<tr>
<th>Country</th>
<th>City</th>
<th>Address</th>
<th>Telephone</th>
<th>Facsimile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Austria</td>
<td>Wels</td>
<td>Tel: 43-7242-2244-39</td>
<td>Fax: 43-7242-2244-39</td>
<td></td>
</tr>
<tr>
<td>Denmark</td>
<td>Copenhagen</td>
<td>Tel: 45-4450-2828</td>
<td>Fax: 45-4485-2829</td>
<td></td>
</tr>
<tr>
<td>France</td>
<td>Paris</td>
<td>Tel: 33-1-69-53-63-20</td>
<td>Fax: 33-1-69-30-90-79</td>
<td></td>
</tr>
<tr>
<td>Germany</td>
<td>Dusseldorf</td>
<td>Tel: 49-2129-3766400</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Germany</td>
<td>Karlsruhe</td>
<td>Tel: 49-721-625370</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Germany</td>
<td>Munich</td>
<td>Tel: 49-89-627-144-0</td>
<td>Fax: 49-89-627-144-44</td>
<td></td>
</tr>
<tr>
<td>Italy</td>
<td>Milan</td>
<td>Tel: 39-0331-742611</td>
<td>Fax: 39-0331-466781</td>
<td></td>
</tr>
<tr>
<td>Italy</td>
<td>Venice</td>
<td>Tel: 39-049-7625286</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Netherlands</td>
<td>Drunen</td>
<td>Tel: 31-416-690399</td>
<td>Fax: 31-416-690340</td>
<td></td>
</tr>
<tr>
<td>Poland</td>
<td>Warsaw</td>
<td>Tel: 48-22-3325737</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spain</td>
<td>Madrid</td>
<td>Tel: 34-91-708-08-90</td>
<td>Fax: 34-91-708-08-91</td>
<td></td>
</tr>
<tr>
<td>Sweden</td>
<td>Stockholm</td>
<td>Tel: 46-8-5090-4654</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UK</td>
<td>Wokingham</td>
<td>Tel: 44-118-921-5800</td>
<td>Fax: 44-118-921-5820</td>
<td></td>
</tr>
</tbody>
</table>

© 2016 Microchip Technology Inc.