Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

Processor Extension Pak (PEP) and Debug Header Specification

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.

Trademarks

ISBN: 978-1-63277-994-6

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Table of Contents

Chapter 1. PEP and Debug Header Overview

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 "DSXXXXXA", where "XXXXX" 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 online Help (Help menu).

This chapter contains the following:

1.1 WHAT IS A PROCESSOR EXTENSION PAK (PEP)?

A Processor Extension Pak contains a debug header, adapter board, and stand-offs. A PEP is what you purchase when you want a debug header.

1.2 WHAT IS A DEBUG HEADER?

A debug header is a circuit board that allows an emulator or debugger to debug code for a specific device. A special version of the device (-ICE/-ICD) with on-board debug circuitry is located on the header. Connectors on the side of the header allow it to connect directly or through an adapter to the debug tool. Connectors on the bottom of the header allow it to connect directly or through a transition socket to a target board.

1.3 WHY DO I NEED A HEADER TO DEBUG?

Some PIC® microcontrollers (MCUs), particularly low-pin-count devices (with 20 pins or less), generally must use a header for debugging. This is done to free up I/O lines for your application and to make production parts more affordable. Optional headers are also available for high-pin-count devices (with 64 pins or higher).

Debugging requires a two-line connection (plus VDD, VSS and VPP) to communicate with the device. In a high-pin-count device, losing a few I/O lines is generally not a problem for most designs. But in a low-pin-count device, it can be a critical problem. Imagine having to do an 8-pin design where there are only 5 I/Os, having used up 2 I/Os just for debugging!

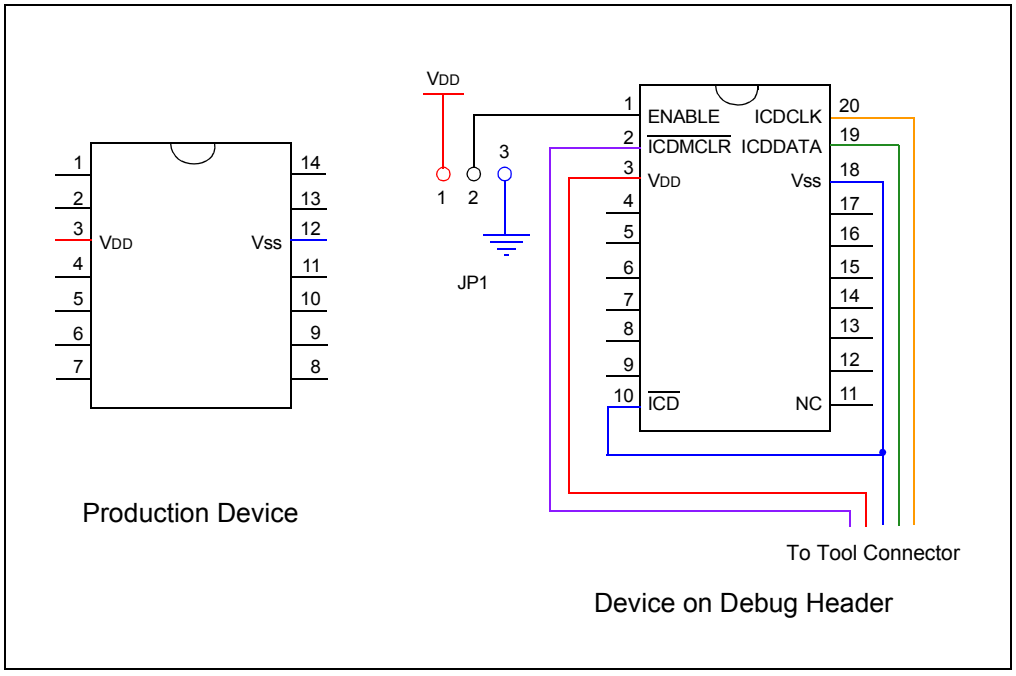
Headers are also used to save you money. In high-pin-count devices, adding debugging to the silicon can generally be done at little or no cost since the silicon is already fairly large. However, low-pin-count devices are low cost specifically because they use very little silicon. So, adding debugging circuitry on-board these parts would add significant cost since it would raise the amount of silicon used by a considerable percentage. The header places the cost for debugging up front and frees your production parts from the extra cost of an unused debug module.

Microchip also makes optional debug modules, usually for high-pin-count devices. The module is optional because you can still do basic debugging without a header, but if you use one, you get back I/O lines, and may also gain additional debugging features. Only certain devices can use an optional header, so see Chapter 3. “Optional Debug Headers”.

Microchip lists what header must be ordered to work with your device, if one is required. Simply consult Chapter 2. “Required Debug Headers”.

Note that in all cases, devices can be programmed “in circuit” (called ICSP™) with very few exceptions. Even devices without an internal debug feature can still be programmed by connecting the programming/debugging tool to the in-circuit programming lines. These devices simply cannot perform debugging without a header.

FIGURE 1-1: PRODUCTION DEVICE VS. HEADER DEVICE

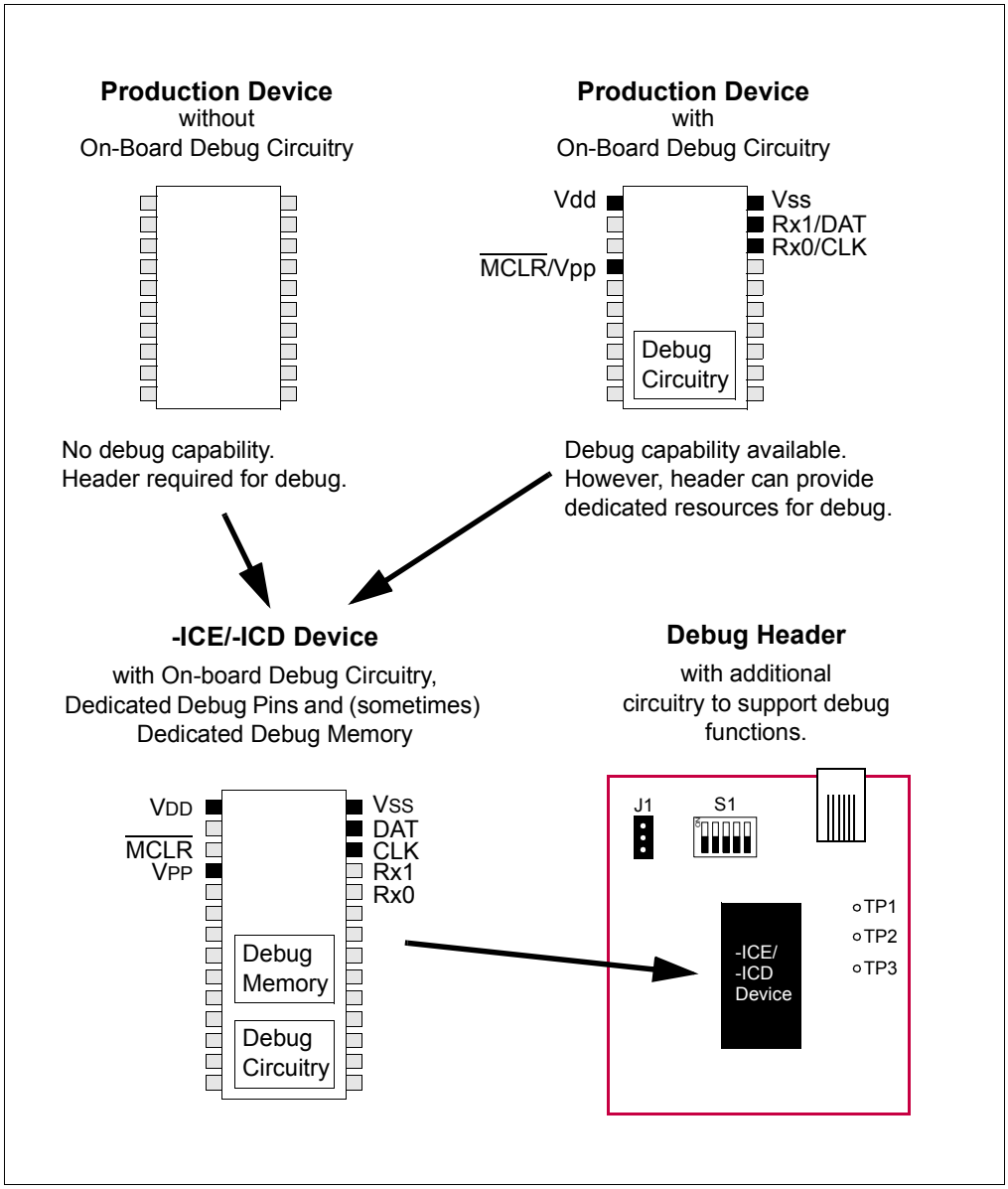


1.4 DEBUG DETAILS

Next generation in-circuit emulators (such as the MPLAB® REAL ICE™ in-circuit emulator) and in-circuit debuggers work with devices that have on-chip debug circuitry. Sometimes the actual production device will have this circuitry and sometimes a special version of this device is required or available for code debugging. This special version of the chip, with the suffix -ICE or -ICD, is mounted on a debug header (Figure 1-2).

> **Note:** -ICE/-ICD devices are only used on the debug header; they are not sold separately.

FIGURE 1-2: DEBUG OPTIONS



To determine which device resources must be dedicated to debugging for either a device with on-board debug capability or the special -ICE/-ICD device, see the “Resources used by ...” section of the in-circuit emulator or in-circuit debugger online Help file.

Then, depending on the debug tool, different features of the special -ICE/-ICD device may be available. A summary is provided below. To determine actual features, see the debug header documentation for a specific device.

TABLE 1-1: DEVICE FEATURES SUMMARY

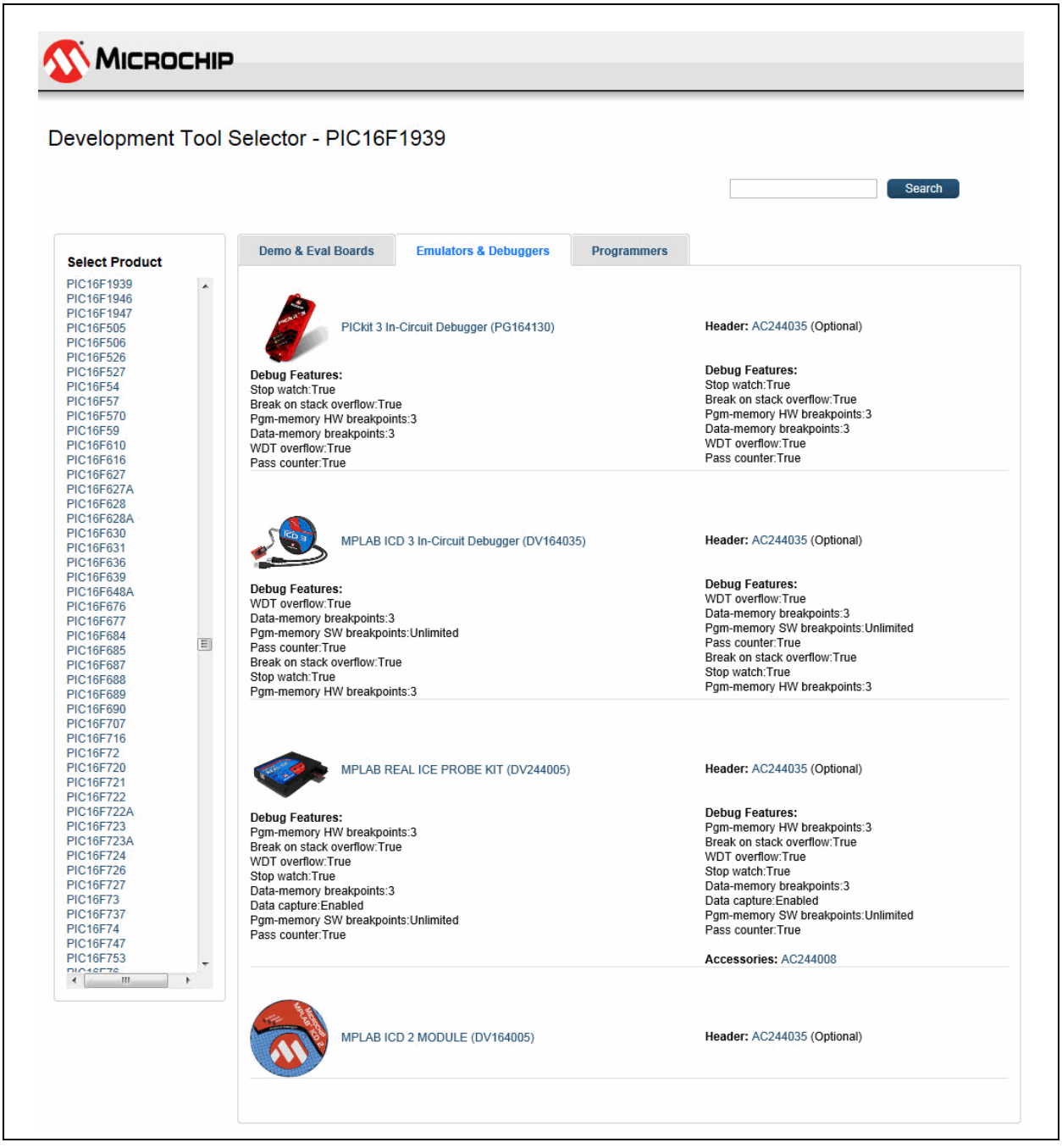
Debug Tool	-ICE	-ICD
In-Circuit Emulators	Basic Emulator Functions	Basic Debug Functions
In-Circuit Debuggers	Basic Debug Functions	

1.5 COMPARE DEBUG HEADER AND DEVICE FEATURES

For some devices that have on-board debug capability, the optional header provides more debug features than the device itself. To determine if your device and optional header differ in their debug features, go to the Development Tool Selector (DTS) to look for your device:

1. In a web browser, go to : http://www.microchip.com/dtsapp/
2. Select your device from the “Select Product” list. Or, type the name of your device into the "Search" box and click **Search**. The name will appear at the top of the “Select Product” list, where you can select it.
3. Click on the tab “Emulators & Debuggers” to see debug features.

FIGURE 1-3: DTS DEVICE INFORMATION



1.6 SUPPORT INFORMATION

Debug headers require specific debug tools to operate with MPLAB X IDE. Acquire these before purchasing a debug header in a Processor Extension Pak (PEP). Available PEPs are listed in Chapter 2. “Required Debug Headers” and Chapter 3. “Optional Debug Headers”.

To continue setting up emulation header hardware, see Section 1.7 “Debug Header Hardware Setup”.

Contact Customer Support for issues with emulation headers.

1.6.1 Tools Support

Debug headers are supported on the following tools:

- PICkit™ 3 in-circuit debugger
- MPLAB® ICD 3 in-circuit debugger
- MPLAB® REAL ICE® in-circuit emulator

1.6.2 Customer Support

Users of Microchip products can receive assistance through several channels:

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

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

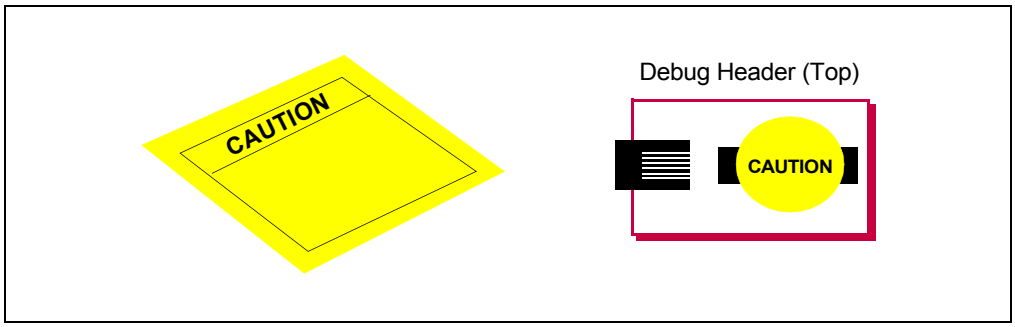
Documentation errors or comments may be sent to: docerrors@microchip.com.

1.7 DEBUG HEADER HARDWARE SETUP

To set up your header, perform the following steps:

1. Check the debug header for any stickers and the header box for any paper inserts that may specify special operating instructions (Figure 1-4). Follow these instructions before doing anything else.

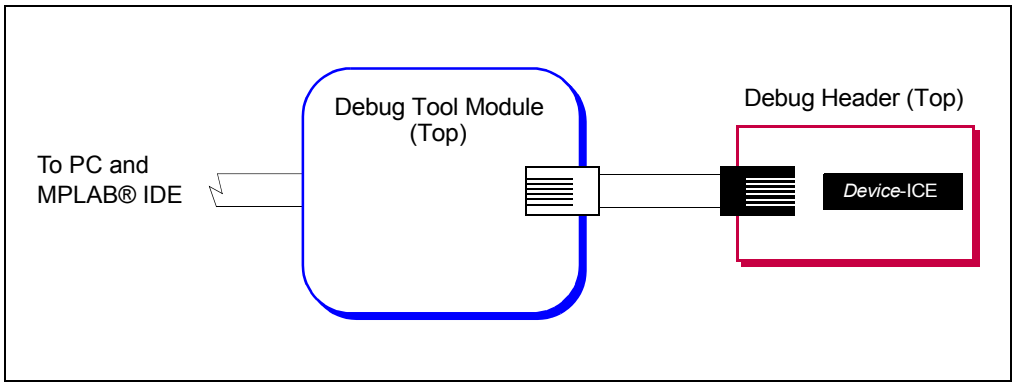
FIGURE 1-4: SPECIAL HEADER INSTRUCTIONS



2. Set any jumpers or switches on the header to determine device functionality or selection, as specified for that header. See the sections "Optional Debug Headers" or "Required Debug Headers" for information on how to set up individual headers.
3. Connect the header to your desired debug tool by consulting the tool documentation for connection options. An example connection is shown in Figure 1-5.

   The special -ICE/-ICD device is mounted on the top of a header and its signals are routed to the emulator or debugger connector. These special device versions are labeled with the appropriate suffix (e.g., *Device*-ICE).

FIGURE 1-5: CONNECT HEADER TO DEBUG TOOL

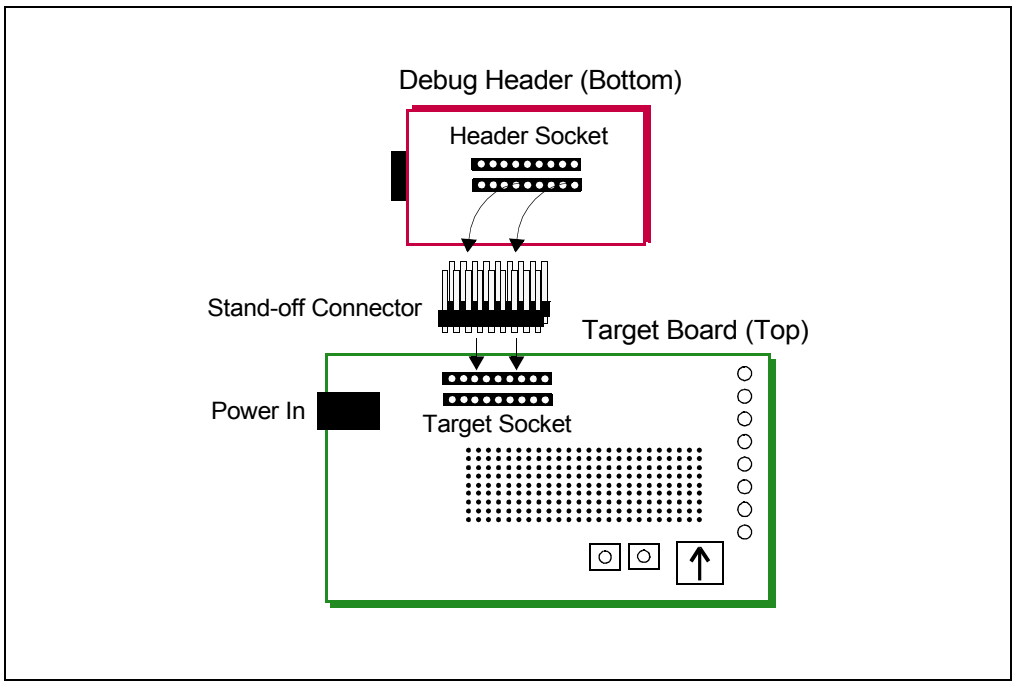


4. Connect the header to the target board. On the bottom of the header is a socket that is used to connect to the target board. The header can be connected to the target board as follows:
   a) PDIP header socket to PDIP target socket with a stand-off (male-to-male) connector
   b) Header socket to plug on the target board
   c) Header socket to target socket with a transition socket (see the *Transition Socket Specification*, DS51194)

   An example connection is shown in Figure 1-6.

The header socket will have the same pin count as your selected device. The -ICE/-ICD device on the top of the header usually has a larger pin count because it has additional pins that are dedicated to debug.

FIGURE 1-6: CONNECT HEADER TO TARGET



5. If using a debug tool that can power the target, power that tool now.
6. Power the target, if needed.

1.8 DEBUG HEADER SETUP FOR MPLAB X IDE

Follow these instructions to use the debug header:

1. Set up the debug header as specified in Section 1.7 “Debug Header Hardware Setup”.
2. Begin creating a project for a device supported by your debug header using the Projects wizard (*File>New Project*). See MPLAB X IDE documentation for more on Projects.
3. In one step of the wizard you will have an opportunity to specify the debug header product number (AC######).
4. In another step you will specify the hardware (debug) tool to which your header is attached.
5. Once the wizard is complete, write code for your project.
6. Select *Debug>Debug Project* to run and debug your code.

Note: A debug header can only be used to debug (Debug menu), not to program (Run menu). See Section 1.9.1 “Programming Details”.

1.9 ADDITIONAL INFORMATION

The following additional information is useful when using a Debug Header from a Processor Extension Pak.

1.9.1 Programming Details

The debug header is designed to be used with the in-circuit emulator or the in-circuit debugger selected as a debugger, not a programmer, in MPLAB X IDE. Any programming of the special -ICE/-ICD device on the header is for debug purposes and includes the debug executive. See your related debug tool documentation for details on using it as a debugger.

To program production (non-special) devices with your debug tool, use the Universal Programming Module (AC162049) or design a modular interface connector on the target. See the appropriate specification for connections. For the most up-to-date device programming specifications, see the Microchip website (www.microchip.com).

Also, production devices may be programmed with the following tools:

- MPLAB PM3 device programmer
- PICkit 3 development programmer
- MPLAB ICD 3 in-circuit debugger (select as a programmer)
- MPLAB REAL ICE in-circuit emulator (select as a programmer)

1.9.2 Calibration Bits

The calibration bits for the band gap and internal oscillator are always preserved to their factory settings.

1.9.3 Performance Issues

PIC MCUs do not support partial program memory erase. Therefore, users may experience slower performance than with other devices.

Also, see either the in-circuit emulator or the in-circuit debugger Help file for information on specific device limitations that may affect performance.

Chapter 2. Required Debug Headers

INTRODUCTION

Some devices have no built-in debug circuitry. Therefore, special -ICE/-ICD versions of these devices are required for debug tool operation.

Currently available debug headers and their associated -ICE/-ICD devices are shown below, sorted by supported device.

TABLE 1: REQUIRED DEBUG HEADERS BY DEVICE

Device Supported by Debug Header	Pin Count	PEP* Part Number	-ICE/-ICD Device on Debug Header	VDD Max
PIC10F200/2/4/6	8/14	AC162059	PIC16F505-ICD	5.5V
PIC10F220/2	8/14	AC162070	PIC16F506-ICD	5.5V
PIC10F320/322	6/8	AC244045	PIC10F320-ICE	5.5V
PIC10LF320/322	6/8	AC244045	PIC10F320-ICE	3.6V
PIC12F508/509	8/14	AC162059	PIC16F505-ICD	5.5V
PIC12F510	8/14	AC162070	PIC16F506-ICD	5.5V
PIC12F519	8/14	AC162096	PIC16F526-ICD	5.5V
PIC12F609/HV609	28	AC162083	PIC16F616-ICD	5.5V
PIC12F615/HV615	28	AC162083	PIC16F616-ICD	5.5V
PIC12F617	28	AC162083	PIC16F616-ICD	5.5V
PIC12F629	8	AC162050	PIC12F675-ICD	5.5V
PIC12F635	14	AC162057	PIC16F636-ICD	5.5V
PIC12F675	8	AC162050	PIC12F675-ICD	5.5V
PIC12F683	8	AC162058	PIC12F683-ICD	5.5V
PIC12F1501(1)	8	AC244051	PIC16F1509-ICE	5.5V
PIC12LF1501(1)	8	AC244052	PIC16LF1509-ICE	3.6V
PIC16F505	8/14	AC162059	PIC16F505-ICD	5.5V
PIC16F506	8/14	AC162070	PIC16F506-ICD	5.5V
PIC16F526	8/14	AC162096	PIC16F526-ICD	5.5V
PIC16F527	20	AC244061	PIC16F527-ICD	5.5V
PIC16F570	28	AC244062	PIC16F570-ICD	5.5V
PIC16F610/HV610	14/16	AC162083	PIC16F616-ICD	5.5V
PIC16F616/HV616	14/16	AC162083	PIC16F616-ICD	5.5V
PIC16F627A/628A	18	AC162053	PIC16F648A-ICD	5.5V
PIC16F630	14	AC162052	PIC16F676-ICD	5.5V
PIC16F631	20	AC162061	PIC16F690-ICD	5.5V
PIC16F636	14	AC162057	PIC16F636-ICD	5.5V
PIC16F639(3)	20	AC162066	PIC16F636-ICD	5.5V

* PEP = Processor Extension Pak.

Note 1: Header optional for other devices, but required for this device.

Note 2: VDDCORE Max

Note 3: Dual die

TABLE 1: REQUIRED DEBUG HEADERS BY DEVICE (CON'T)

Device Supported by Debug Header	Pin Count	PEP* Part Number	-ICE/-ICD Device on Debug Header	VDD Max
PIC16F648A	18	AC162053	PIC16F648A-ICD	5.5V
PIC16F676	14	AC162052	PIC16F676-ICD	5.5V
PIC16F677	20	AC162061	PIC16F690-ICD	5.5V
PIC16F684	14	AC162055	PIC16F684-ICD	5.5V
PIC16F685/687	20	AC162061	PIC16F690-ICD	5.5V
PIC16F688	14	AC162056	PIC16F688-ICD	5.5V
PIC16F689/690	20	AC162061	PIC16F690-ICD	5.5V
PIC16F716	18	AC162054	PIC16F716-ICD	5.5V
PIC16F785/HV785	20	AC162060	PIC16F785-ICD	5.5V
PIC16F1503(1) PIC16F1507(1)	14 20	AC244051	PIC16F1509-ICE	5.5V
PIC16LF1503(1) PIC16LF1507(1)	14 20	AC244052	PIC16LF1509-ICE	3.6V
PIC18F13K50 PIC18F14K50	20	AC244023	PIC18F14K50-ICE	5.5V
PIC18LF13K50 PIC18LF14K50	20	AC244024	PIC18LF14K50-ICE	3.6V, 2.75V(2)
PIC24F04KA200(1) PIC24F04KA201(1)	14 20	AC244028	PIC24F16KA102-ICE	3.6V

* PEP = Processor Extension Pak.

Note 1: Header optional for other devices, but required for this device.

Note 2: VDDCORE Max

Note 3: Dual die

AC162050, AC162058

Header Identification

The AC number is used for ordering the Processor Extension Pak, which contains the debug header. However, this number is not on the header, as the board may be used for multiple headers by inserting various -ICE/-ICD devices. To identify these headers, use the following information.

AC Number	-ICE/-ICD Device	Board Assembly Number
AC162050	PIC12F675-ICD	02-01662
AC162058	PIC12F683-ICD	

Header Setup and Operation

For these headers, device peripherals need to be selected by setting jumper J1 to the appropriate position. For AC162050, this will have the effect of selecting the device.

AC Number	Jumper J1	Function	Device Selected
AC162050	2-3	A/D Disabled	PIC12F629
	1-2	A/D Enabled	PIC12F675
AC162058	1-2	A/D Enabled	PIC12F683

Header Limitations

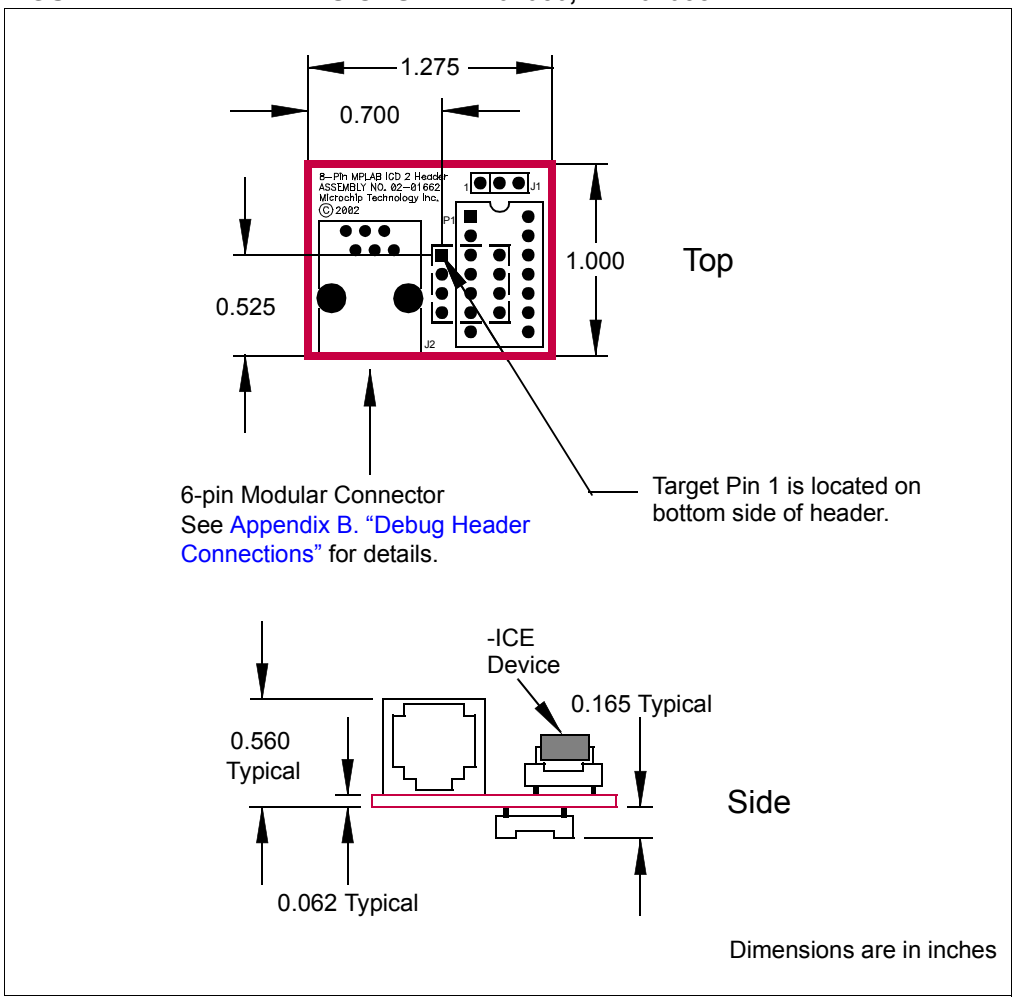
Sometimes a header device (-ICE/-ICD) has operational issues or errata. To determine if a device on a header has limitations, see your hardware tool documentation.

Header Dimensions

The figure below lists the dimensions for the debug header. Dimensions are design values in inches.

If the length and/or width of the debug header is too large a footprint for the target board, consider using stand-offs, transition sockets or other extenders in the header connection socket to raise the header above the target.

FIGURE 2-1: DIMENSIONS – AC162050, AC162058



AC162052, AC162055, AC162056, AC162057

Header Identification

The AC number is used for ordering the Processor Extension Pak, which contains the debug header. However, this number is not on the header, as the board may be used for multiple headers by inserting various -ICE/-ICD devices. To identify these headers, use the following information.

AC Number	-ICE/-ICD Device	Board Assembly Number
AC162052	PIC16F676-ICD	02-01686
AC162055	PIC16F684-ICD	
AC162056	PIC16F688-ICD	
AC162057	PIC16F636-ICD	

Header Setup and Operation

For these headers, device peripherals need to be selected by setting jumper J1 to the appropriate position. For AC162052 and AC162057, this will have the effect of selecting the device.

AC Number	Jumper J1	Function	Device Selected
AC162052	2-3	A/D Disabled	PIC16F630
	1-2	A/D Enabled	PIC16F676
AC162055	Don’t care	N/A	PIC16F684
AC162056	Don’t care	N/A	PIC16F688
AC162057	2-3	PORTC, Comparator 2 Disabled	PIC12F635
	1-2	PORTC, Comparator 2 Enabled	PIC16F636

Header Limitations

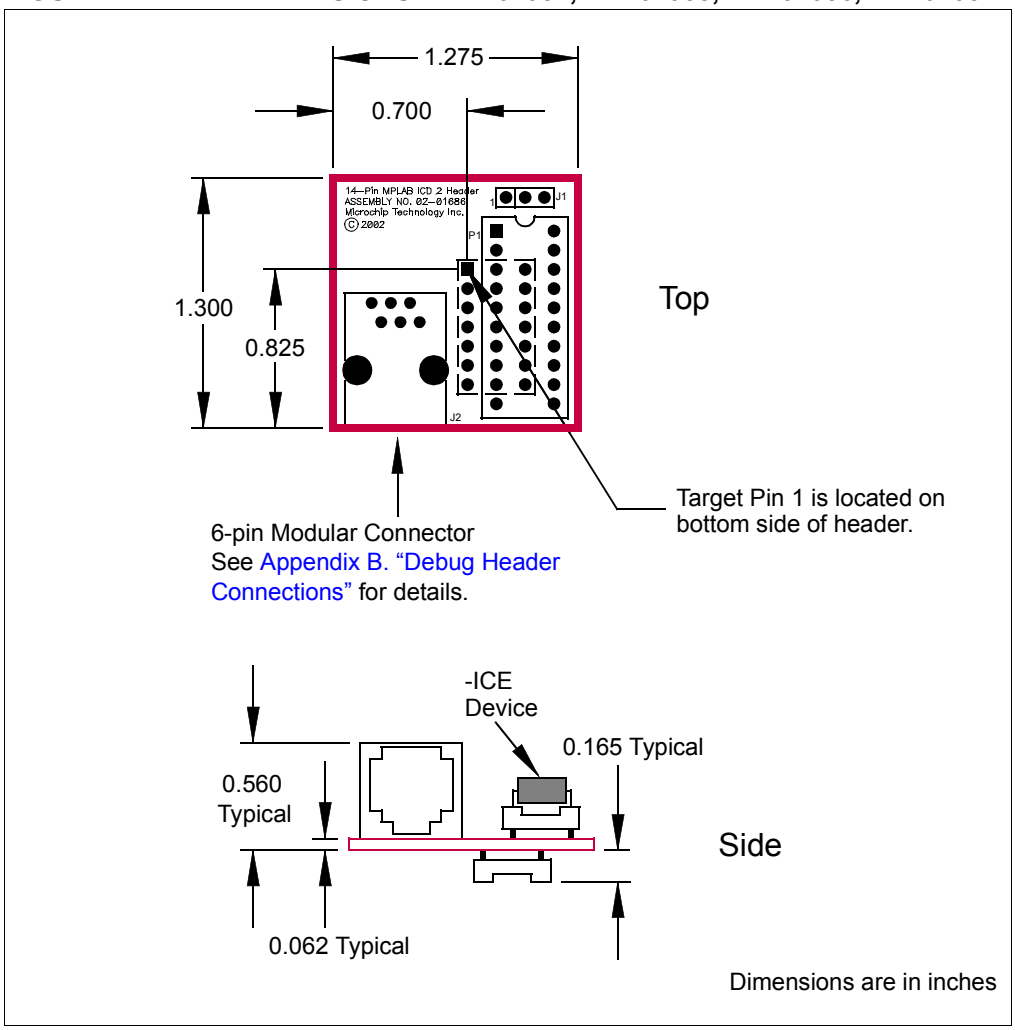
Sometimes a header device (-ICE/-ICD) has operational issues or errata. To determine if a device on a header has limitations, see your hardware tool documentation.

Header Dimensions

The figure below lists the dimensions for the debug header. Dimensions are design values in inches.

If the length and/or width of the debug header is too large a footprint for the target board, consider using stand-offs, transition sockets or other extenders in the header connection socket to raise the header above the target.

FIGURE 2-2: DIMENSIONS – AC162052, AC162055, AC162056, AC162057



AC162053, AC162054

Header Identification

The AC number is used for ordering the Processor Extension Pak, which contains the debug header. However, this number is not on the header, as the board may be used for multiple headers by inserting different -ICE/-ICD devices. To identify these headers, use the following information.

AC Number	-ICE/-ICD Device	Board Assembly Number
AC162053	PIC16F648A-ICD	02-01695
AC162054	PIC16F716-ICD	

Header Setup and Operation – AC162053

For these headers, there are no jumpers/switches. The device with the most program memory is always selected.

If PIC16F627A or PIC16F628A devices are selected for development in MPLAB X IDE, the warning “Invalid target device ID” may be received in the build window. Ignore this warning. The reason for the warning is that the PIC16F648A-ICD device supports PIC16F648A, PIC16F627A and PIC16F628A, but only reports the device ID for the PIC16F648A.

Header Setup and Operation – AC162054

This header supports one device (PIC16F716) so there are no jumpers or switches.

Header Limitations

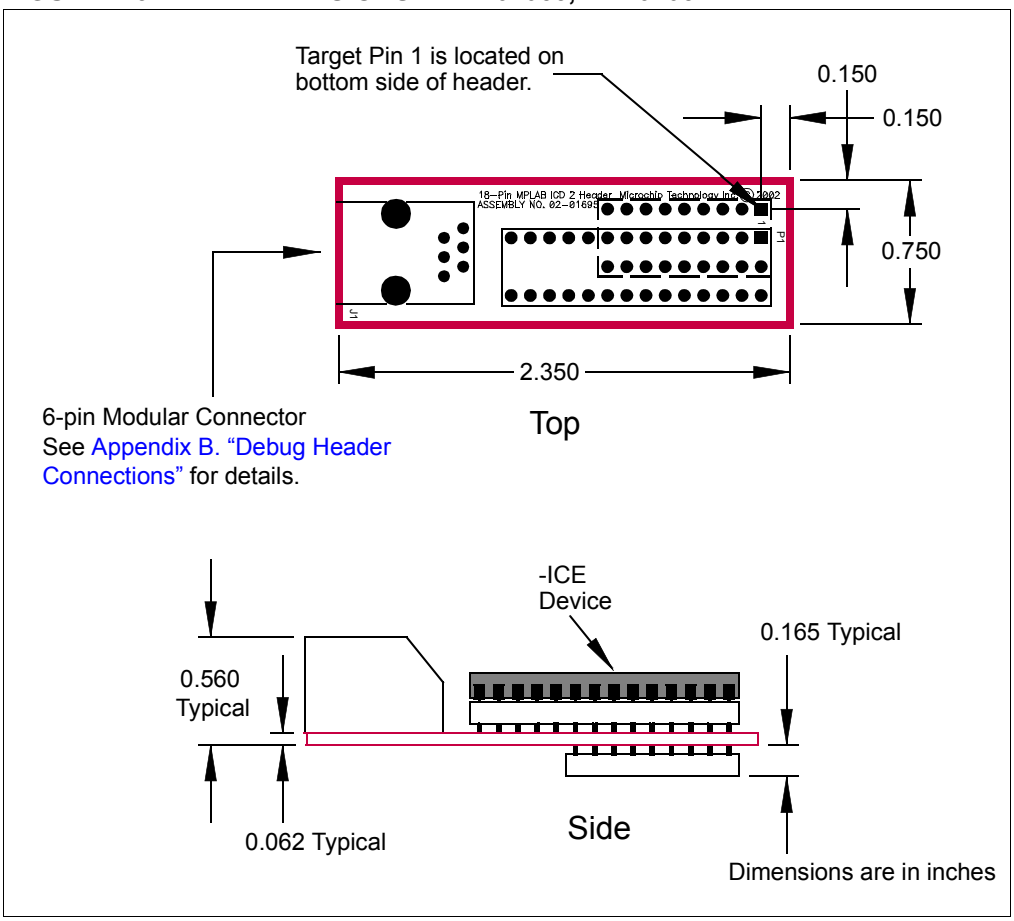
Sometimes a header device (-ICE/-ICD) has operational issues or errata. To determine if a device on a header has limitations, see your hardware tool documentation.

Header Dimensions

The figure below lists the dimensions for the debug header. Dimensions are design values in inches.

If the length and/or width of the debug header is too large a footprint for the target board, consider using stand-offs, transition sockets or other extenders in the header connection socket to raise the header above the target.

FIGURE 2-3: DIMENSIONS – AC162053, AC162054



AC162059, AC162070, AC162096

Header Identification

The AC number is used for ordering the Processor Extension Pak, which contains the debug header. However, this number is not on the header, as the board may be used for multiple headers by inserting different -ICE/-ICD devices. To identify these headers, use the following information.

AC Number	-ICE/-ICD Device	Board Assembly Number
AC162059	PIC16F505-ICD	02-01803
AC162070	PIC16F506-ICD	
AC162096	PIC16F526-ICD	

Header Setup and Operation

The -ICD devices on these headers are specifically designed to select a device without the use of additional jumpers or switches.

These headers support 8- and 14-pin devices (see Figure 2-4.) For the AC162059 and AC162070, there is an 8-pin and a 14-pin connector. For the AC162096, there is only a 14-pin connector. (The 8-pin connector is not populated.) Use the 14-pin connector for 8-pin devices, but make sure device pin 1 is placed at the 14-pin connector pin 1.

Header Limitations

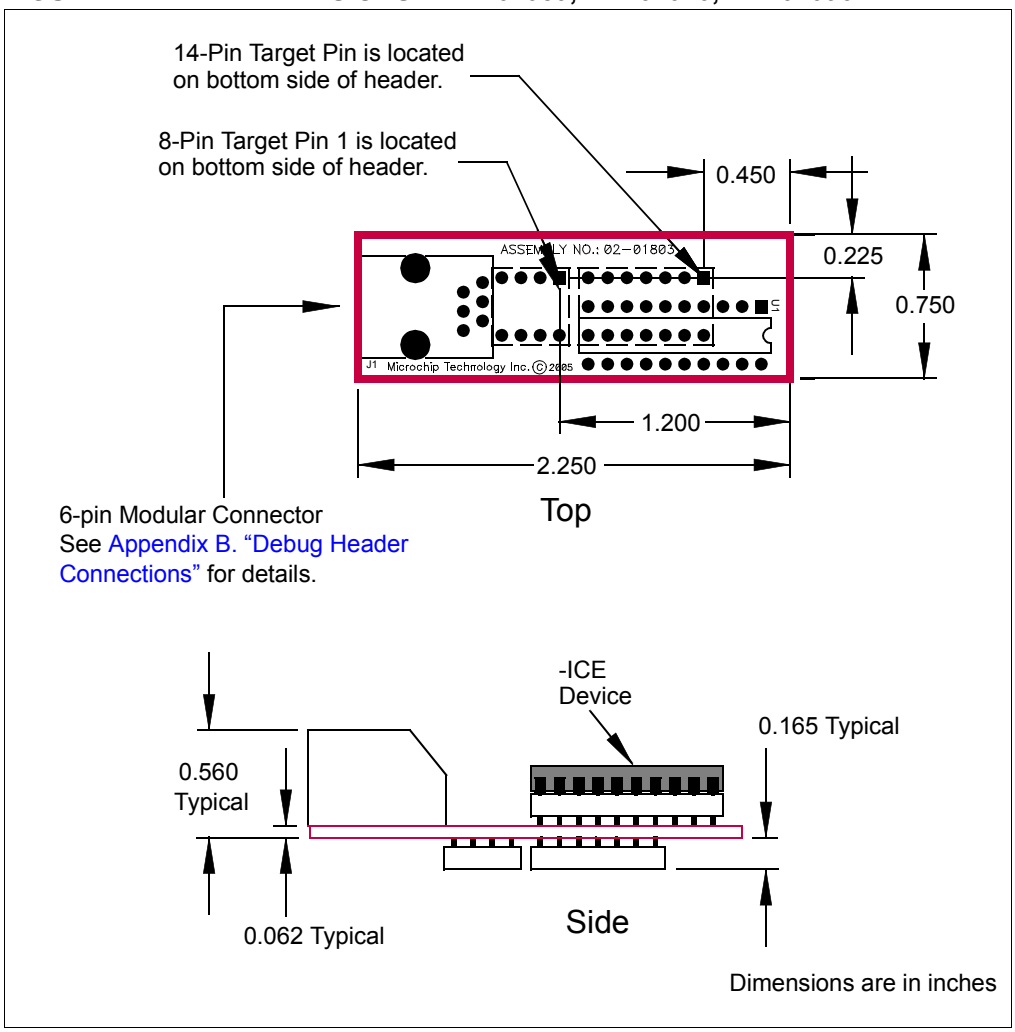
Sometimes a header device (-ICE/-ICD) has operational issues or errata. To determine if a device on a header has limitations, see your hardware tool documentation.

Header Dimensions

The figure below lists the dimensions for the debug header. Dimensions are design values in inches.

If the length and/or width of the debug header is too large a footprint for the target board, consider using stand-offs, transition sockets or other extenders in the header connection socket to raise the header above the target.

FIGURE 2-4: DIMENSIONS – AC162059, AC162070, AC162096



AC162060

Header Identification

The AC number is used for ordering the Processor Extension Pak, which contains the debug header. However, this number is not on the header, as the board may be used for multiple headers by inserting different -ICE/-ICD devices. To identify this header, use the following information.

AC Number	-ICE/-ICD Device	Board Assembly Number
AC162060	PIC16F785-ICD	02-01820

Header Setup and Operation

For the PIC16F785 20-pin header, connect the jumper J2 to enable the shunt regulator.

Device	Device Type	Jumper J2	Function
PIC16F785	F	1-2	Disable shunt regulator
PIC16HV785	HV	2-3	Enable shunt regulator

Header Limitations

Sometimes a header device (-ICE/-ICD) has operational issues or errata. To determine if a device on a header has limitations, see your hardware tool documentation.

Header Dimensions

The figure below lists the dimensions for the debug header. Dimensions are design values in inches.

If the length and/or width of the debug header is too large a footprint for the target board, consider using stand-offs, transition sockets or other extenders in the header connection socket to raise the header above the target.

FIGURE 2-5: DIMENSIONS – AC162060