

PICkit™ 2 Programmer/Debugger User's Guide

© 2008 Microchip Technology Inc.

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in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates.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.

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, rfPIC and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, PICkit, PICDEM, PICDEM.net, PICtail, PIC^{32} logo, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, Select Mode, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

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

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

© 2008, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Microchip received ISO/TS-16949:2002 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.

© 2008 Microchip Technology Inc.

Table of Contents

Preface 1

Chapter 1. PICkit 2 Programmer/Debugger Overview

1.1 Introduction 7
1.2 PICkit 2 Development Programmer/Debugger Contents 7
1.3 PICkit 2 Development Programmer/Debugger 7
1.4 PICkit 2 Programmer Application 10

Chapter 2. PICkit 2 Programmer Getting Started

2.1 Introduction 15
2.2 Installing the PICkit 2 Hardware 15
2.3 Installing and Launching the PICkit 2 Programmer Application 16
2.4 Connecting to the Device 16
2.5 Selecting Target Power 18
2.6 Importing a Hex File 19
2.7 Writing the Program to the Device 20
2.8 Verifying the Device 23
2.9 Reading Device Memory 23
2.10 Code Protecting the Device 23
2.11 Erasing and Blank Checking the Device 24
2.12 Automating Write/Read Procedures 24
2.13 PICkit 2 Calibration and Unit ID 25

Chapter 3. Using In-Circuit Serial Programming™ (ICSP™)

3.1 Introduction 27
3.2 Isolate V_{PP}/$\overline{MCLR}$/Port Pin 28
3.3 Isolate ICSPCLK or PGC and ICSPDAT or PGD pins 28
3.4 V_{DD} 29
3.5 V_{SS} 30
3.6 Cable Lengths 30
3.7 Serial EEPROM and KeeLoq HCS Devices 30

Chapter 4. PICkit 2 Debug Express

4.1 Introduction 31
4.2 PICkit 2 Debug Express Kit Contents 31
4.3 Installing the Hardware and Software 32
4.4 Using PICkit 2 Debug Express 32
4.5 Debug Express Tutorial 36

© 2008 Microchip Technology Inc.

Chapter 5. Troubleshooting
5.1 Introduction 49
5.2 Frequently Asked Questions 49
5.3 PICkit 2 Debug Express MPLAB IDE Errors 54
Chapter 6. Updating the PICkit 2 Operating System
6.1 Introduction 61
6.2 Updating the PICkit 2 OS - PICkit 2 Programmer Application 61
6.3 Updating the PICkit 2 OS – MPLAB IDE 62
Chapter 7. PICkit 2 UART Tool
7.1 Introduction 65
7.2 Connecting the PICkit 2 UART Tool 65
7.3 The PICkit 2 UART Tool Window 66
Appendix A. MPLAB IDE Reference
A.1 Introduction 71
A.2 Debugging Functions 71
A.3 Programming Functions 73
A.4 Settings Dialog 75
Appendix B. PICkit 2 Schematics
Index 79
Worldwide Sales and Service 82

© 2008 Microchip Technology Inc.

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 "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® IDE on-line help. Select the Help menu, and then Topics to open a list of available on-line help files.

INTRODUCTION

This chapter contains general information that will be useful to know before using PICkit™ 2 . Items discussed include:

- Document Layout
- Conventions Used in this Guide
- Warranty Registration
- Recommended Reading
- The Microchip Web Site
- Development Systems Customer Change Notification Service
- Customer Support

DOCUMENT LAYOUT

This document describes how to use the PICkit 2 as a development tool to emulate and debug firmware on a target board. The manual layout is as follows:

- **Chapter 1. PICkit 2 Programmer/Debugger Overview** – Provides an overview of the PICkit 2 Programmer/Debugger.
- **Chapter 2. PICkit 2 Programmer Getting Started** – Provides Instructions on how to get started using the PICkit 2 Programmer/Debugger to program Flash-based PIC® Microcontroller Units (MCUs).
- **Chapter 3. Using In-Circuit Serial Programming™ (ICSP™)** – Describes programming with the PICkit 2 Programmer/Debugger using In-Circuit Serial Programming™ (ICSP™).
- **Chapter 4. PICkit 2 Debug Express** – Provides a tutorial on using the PICkit™ 2 Debug Express debugger program.
- **Chapter 5. Troubleshooting** – Provides information on solving common problems and Debug Express errors.
- **Chapter 6. Updating the PICkit 2 Operating System** – Provides instruction on how to update your PICkit 2 Programmer/Debugger's Operating System.
- **Chapter 7. PICkit 2 UART Tool** – Provides information on using the PICkit 2 as a serial UART terminal interface for communicating with a PIC microcontroller
- **Appendix A. MPLAB IDE Reference** – Describes how the PICkit 2 Programmer/Debugger works with MPLAB IDE.
- **Appendix B. PICkit 2 Schematics** – Illustrates the PICkit 2 Programmer/Debugger hardware schematic diagrams.

© 2008 Microchip Technology Inc.

CONVENTIONS USED IN THIS GUIDE

The following conventions may appear in this documentation:

DOCUMENTATION CONVENTIONS

Description	Represents	Examples
Arial font:		
Italic	Referenced books	*MPLAB® IDE User's Guide*
	Emphasized text	...is the *only* compiler...
Initial caps	A window	the Output window
	A dialog	the Settings dialog
	A menu selection	select Enable Programmer
Quotes	A field name in a window or dialog	"Save project before build"
Underlined, italic with right angle bracket	A menu path	*File>Save*
Bold characters	A dialog button	Click **OK**
	A tab	Click the **Power** tab
Text in angle brackets < >	A key on the keyboard	Press <Enter>, <F1>
Courier New font:		
Plain	Sample source code	`#define START`
	Filenames	`autoexec.bat`
	File paths	`c:\mcc18\h`
	Keywords	`_asm, _endasm, static`
	Command-line options	`-Opa+, -Opa-`
	Bit values	`0, 1`
	Constants	`0xFF,'A'`
Italic	A variable argument	*`file`*`.o`, where *`file`* can be any valid filename
Square brackets [ ]	Optional arguments	`mpasmwin [options] file [options]`
Curly brackets and pipe character: { | }	Choice of mutually exclusive arguments; an OR selection	`errorlevel {0|1}`
Ellipses...	Replaces repeated text	`var_name [, var_name...]`
	Represents code supplied by user	`void main (void) { ... }`

WARRANTY REGISTRATION

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

RECOMMENDED READING

This user's guide describes how to use PICkit 2 . Other useful documents are listed below. The following Microchip documents are available and recommended as supplemental reference resources.

44-Pin Demo Board User's Guide (DS41296)

Consult this document for instructions on how to use the 44-Pin demo board as a development tool to emulate and debug firmware on a target board.

Low Pin Count Demo Board User's Guide (DS51556)

Consult this document for instructions on how to use Microchip Technology's low pin count device (8-pin, 14-pin and 20-pin). This document includes a series of tutorials.

MPLAB® IDE Quick Start Guide (DS51281)

Describes how to set up the MPLAB IDE software and use it to create projects and program devices.

MPLAB® IDE User's Guide/Help (DS51519)

Consult this document for more information pertaining to the installation and features of the MPLAB Integrated Development Environment (IDE) software. An on-line Help version is also available.

In-Circuit Serial Programmer™ (ICSP™) Guide (DS30277)

This document contains helpful design guidelines for successful ICSP programming. It includes application notes on hardware designs and the ICSP programming specifications.

MPASM™ Assembler, MPLINK™ Object Linker, MPLIB™ Object Librarian User's Guide (DS33014)

Describes how to use the Microchip PIC® MCU assembler (MPASM assembler), linker (MPLINK linker), and librarian (MPLIB librarian).

README for PICkit™ 2 Debug Express

For the latest information on using the PICkit 2 Debug Express, read the "`Readme for PICkit 2.htm`" file (an HTML file) in the Readmes subdirectory of the MPLAB IDE installation directory. The Readme file contains updated information and known issues that may not be included in this user's guide.

Readme Files

For the latest information on using other tools, read the tool-specific Readme files in the Readmes subdirectory of the MPLAB IDE installation directory. The Readme files contain updated information and known issues that may not be included in this user's guide.

© 2008 Microchip Technology Inc.

THE MICROCHIP WEB SITE

Microchip provides online support via our web site at 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

DEVELOPMENT SYSTEMS CUSTOMER CHANGE NOTIFICATION SERVICE

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

To register, access the Microchip web site at www.microchip.com, click on Customer Change Notification and follow the registration instructions.

The Development Systems product group categories are:

- **Compilers** – The latest information on Microchip C compilers and other language tools. These include the MPLAB C18 and MPLAB C30 C compilers; MPASM™ and MPLAB ASM30 assemblers; MPLINK™ and MPLAB LINK30 object linkers; and MPLIB™ and MPLAB LIB30 object librarians.
- **Emulators** – The latest information on Microchip in-circuit emulators.This includes the MPLAB REAL ICE™ and MPLAB ICE 2000 in-circuit emulators.
- **In-Circuit Debuggers** – The latest information on the Microchip in-circuit debuggers. This includes the MPLAB ICD 2 and PICkit™ 2.
- **MPLAB® IDE** – The latest information on Microchip MPLAB IDE, the Windows® Integrated Development Environment for development systems tools. This list is focused on the MPLAB IDE, MPLAB 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 MPLAB PM3 device programmer and the PICSTART® Plus, PICkit 1 and PICkit 2 development programmers.

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.

© 2008 Microchip Technology Inc.

Chapter 1. PICkit 2 Programmer/Debugger Overview

1.1 INTRODUCTION

This chapter introduces the PICkit 2 Development Programmer/Debugger features and PICkit 2 Programmer application menu functions.

This chapter discusses:

- PICkit 2 Development Programmer/Debugger Contents
- PICkit 2 Development Programmer/Debugger
- PICkit 2 Programmer Application

1.2 PICkit 2 DEVELOPMENT PROGRAMMER/DEBUGGER CONTENTS

The PICkit 2 Development Programmer/Debugger kit contains the following items:

1. The PICkit 2 Development Programmer/Debugger
2. USB cable
3. PICkit Starter Kit and MPLAB IDE CD-ROMs

Additionally, the PICkit 2 Starter Kit and PICkit 2 Debug Express kit both contain a demo board with a PIC microcontroller device.

1.3 PICkit 2 DEVELOPMENT PROGRAMMER/DEBUGGER

The PICkit 2 Development Programmer/Debugger is a low-cost development programmer. It is capable of programming most of Microchip's Flash microcontrollers and serial EEPROM devices. For specific device support, see the `README` file included on the PICkit 2 Starter Kit CD-ROM.

> **Note:** The PICkit 2 is intended for development programming. For production programming, please consider the MPLAB PM3 device programmer or other third party programmers designed for a production environment.

New device support can be added by updating the programming software. The latest software is available on Microchip's web site page for the PICkit 2: www.microchip.com/pickit2.

The PICkit 2 also may be used to debug selected devices. See **Chapter 4. "PICkit 2 Debug Express"** for more details.

The PICkit 2 unit is shown in Figure 1-1.

FIGURE 1-1: PICkit™ 2 MCU PROGRAMMER/DEBUGGER



Legend:

1 – Status LEDs	3 – Lanyard Connection	5 – Pin 1 Marker
2 – Push Button	4 – USB Port Connection	6 – Programming Connector

1.3.1 USB Port Connection

The USB port connection is a USB mini-B connector. Connect the PICkit 2 to the PC using the supplied USB cable.

1.3.2 Status LEDs

The Status LEDs indicate the status of the PICkit 2.

1. **Power** (green) – Power is applied to the PICkit 2 via the USB port.
2. **Target** (yellow) – The PICkit 2 is powering the target device.
3. **Busy** (red) – The PICkit 2 is busy with a function in progress, such as programming.

1.3.3 Push Button

The push button may be used to initiate the Write Device programming function when *Programmer>Write on PICkit Button* is checked on the PICkit 2 Programmer application menu (see item labeled 2 in Figure 1-1.)

The push button may also be used to put the PICkit 2 unit operating system firmware into Bootloader mode. For more information on this feature, see **Chapter 6. "Updating the PICkit 2 Operating System"**.

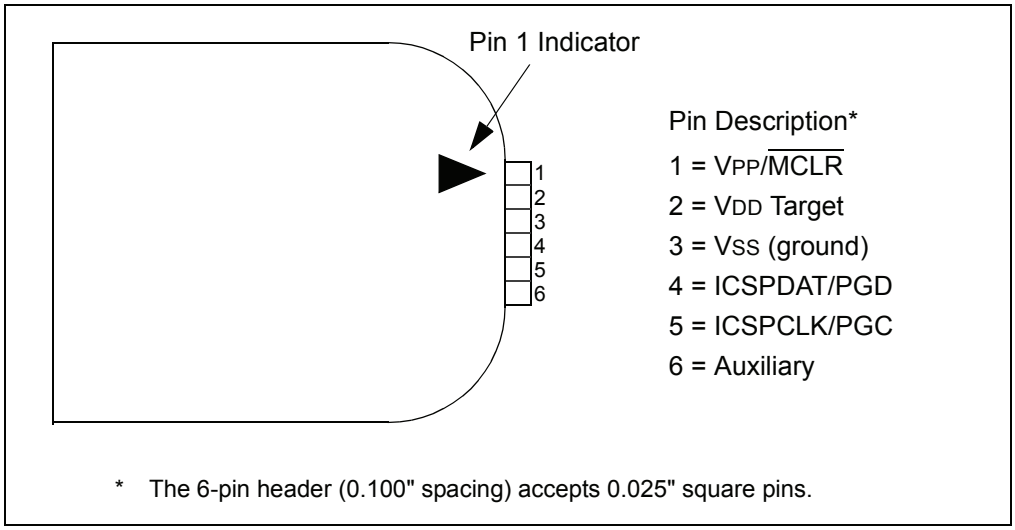
1.3.4 Programming Connector

The programming connector is a 6-pin header (0.100" spacing) that connects to the target device. See the pinout specification in Figure 1-2.

For more information on how to use the PICkit 2 with In-Circuit Serial Programming (ICSP), refer to **Chapter 3. "Using In-Circuit Serial Programming™ (ICSP™)"**.

© 2008 Microchip Technology Inc.

FIGURE 1-2: PICkit™ 2 PROGRAMMER CONNECTOR PINOUT



Note: The programming connector pin functions are different for programming Serial EEPROMS and HCS devices. See the ReadMe file (*Help>Readme)* included with the PICkit 2 programming software for these pinouts.

1.3.5 Lanyard Connection

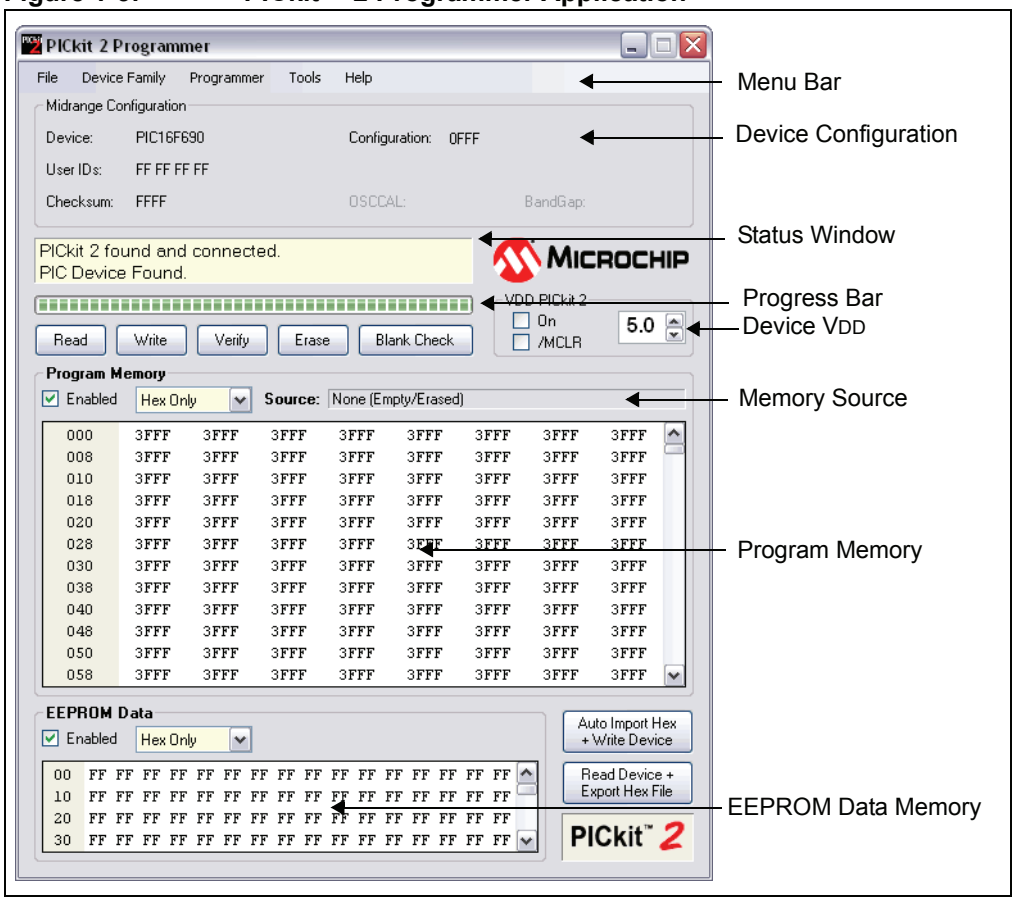
To help prevent possible loss of the PICkit 2, a convenient lanyard connection is available on the programmer.

1.4 PICkit 2 PROGRAMMER APPLICATION

The PICkit 2 Programmer application allows you to program all supported devices listed in the PICkit 2 Readme file. The programming interface appears as shown in Figure 1-3. Its controls are listed in the following sections.

For more information on how to install and use the PICkit 2 Programmer application, see **Chapter 2. "PICkit 2 Programmer Getting Started"**.

Figure 1-3: PICkit™ 2 Programmer Application



© 2008 Microchip Technology Inc.

1.4.1 Menu Bar

The menu bar selects various functions of the PICkit 2 Programmer application. A summary of the functions are:

File

- Import Hex – Import a hex file for programming. The hex file format INHX32 is supported.
- Export Hex – Export a hex file read from a device.The hex file is created in the INHX32 format.
- File History – Up to the last four hex files opened are displayed with their filepath. These recent hex files may be selected to quickly import them. Note that the file history will initially be blank on a new installation until a hex file is imported.
- Exit – Exit the program.

Device Family

Select a device family to search for a connected device in that family. Selecting the device family of the current part will clear all device data.

Some families which cannot be auto-detected (such as Baseline) will bring up a drop down box from which supported devices may be selected.

Programmer

- Read Device – Reads program memory, data EEPROM memory, ID locations and Configuration bits.
- Write Device – Writes program memory, data EEPROM memory, ID locations and Configuration bits.
- Verify – Verifies program memory, data EEPROM memory, ID locations and Configuration bits read from the target MCU against the code stored in the programming application.
- Erase – Performs a Bulk Erase of the target MCU. OSCCAL and band gap values are preserved on parts with these features.
- Blank Check – Performs a Blank Check of program memory, data EEPROM memory, ID locations and Configuration bits.
- Verify on Write – When checked, the device will be immediately verified after programming on a Write (recommended). When unchecked, the device will be programmed but not verified on a Write.
- Hold Device in Reset – When checked, the $\overline{\text{MCLR}}$ (V_{PP}) pin is held low (asserted). When unchecked, the pin is released (tri-stated), allowing an external pull-up to bring the device out of Reset.
- Write on PICkit Button – When checked, a Write operation will be initiated by pressing the PICkit 2 push button.

Tools

- Enable Code Protect – Enables code protection features of the microcontroller on future Write operations.

Note:	To disable code protect, *all* device memory must be erased and rewritten.

- Enable Data Protect – Enables data protection feature of microcontrollers with data EEPROM memory on future Write operations.

Note:	To disable data protect, *all* device memory must be erased and rewritten.

- Set OSCCAL – Allows the OSCCAL value to be changed for devices where it is stored in the last location of Program Memory.
- Target V_{DD} Source
  - Auto-Detect – The PICkit 2 will automatically detect whether the target device has its own power supply or needs to be powered by the programmer on each operation.
  - Force PICkit 2 – The PICkit 2 will always attempt to supply V_{DD} to the target device.
  - Force Target – The PICkit 2 will always assume the target has its own power supply.
- Calibrate V_{DD} & Set Unit ID – Opens a wizard that steps the user through calibrating the PICkit 2 V_{DD} supplied voltage so it is more accurate, and optionally assigning a Unit ID to identify between multiple PICkit 2 devices.
- Use V_{PP} First Program Entry – When checked, it allows the PICkit 2 to connect to and program devices with configurations and code that interferes with the ICSP signal pins, preventing PICkit 2 from detecting them. Using this feature requires that the PICkit 2 supplies V_{DD} to the target.
- Fast Programming – When checked, the PICkit 2 will attempt to program the device as fast as possible. When unchecked, the PICkit 2 will slow down ICSP communication. This may be helpful for targets with loaded ICSP lines.
- Check Communication – Verifies USB communication with the PICkit 2 and ICSP communication with a target device by attempting to identify the connected device by its device ID.
- UART Tool… - Puts the PICkit 2 in UART Mode and opens a terminal-like interface for communicating with a PIC MCU device program through the USART pins. See **Chapter 7. "PICkit 2 UART Tool"** for more information.
- Troubleshoot... – Opens a wizard to help with troubleshooting connectivity from the PICkit 2 to the target device. This is most useful where the programmer is unable to detect the target device at all.
- Download PICkit 2 Programmer Operating System – Performs a download of the PICkit 2 operating system (firmware).

Help

- PICkit 2 User's Guide – Attempts to launch the user's guide PDF (Adobe® Reader must be installed).
- 44-Pin Demo Board Guide – Attempts to launch the 44-Pin Demo Board User's Guide PDF (Adobe Reader must be installed).
- LPC Demo Board Guide – Attempts to launch the Low Pin Count Demo Board User's Guide PDF (Adobe Reader must be installed).
- PICkit 2 Programmer on the web – Opens www.microchip.com/pickit2 in the default web browser.
- Readme – Opens the PICkit 2 `Readme.txt` file.
- About – Opens a dialog with the PICkit 2 Programmer application version, device file version and firmware version.

1.4.2 Device Configuration

The Device Configuration window displays the device, User ID, Configuration Word and Checksum. It also displays OSCCAL and Band Gap for parts with those features.

For baseline (12-bit core) devices, serial EEPROM devices, and KeeLoq® HCS devices, you must select the device from the Device drop-down menu.

© 2008 Microchip Technology Inc.

All other part family devices will be detected by their device ID and the part name will be displayed on the Device line.

1.4.3 Status Window

The status window displays text status of the operations in progress. If an operation is successful, the status window will display a green background. If an operation fails, the status window will display red. If an operation alerts a caution, the status window will display yellow.

1.4.4 Progress Bar

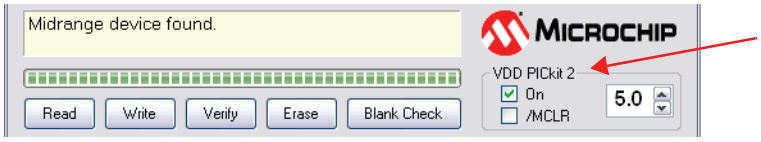
The progress bar displays the progress of an operation.

1.4.5 Device VDD

The PICkit 2 VDD may be turned on and off by clicking the checkbox “On”. The voltage may be set in the box on the right either by typing it directly or using the up/down arrows to adjust it a tenth of a volt at a time. The maximum and minimum allowed voltages will vary depending on the target device.

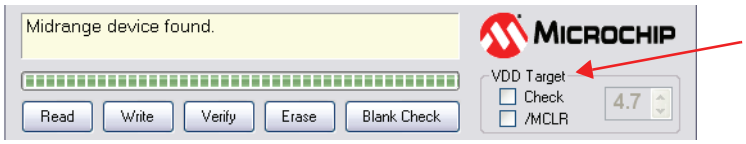
If the “On” checkbox is unchecked, PICkit 2 will automatically turn on the VDD at the set voltage during any requested programming operation.

FIGURE 1-4: PICKIT™ 2 SUPPLIED VDD



If the target device has its own power supply, then the PICkit 2 will display the detected VDD voltage in the box on the right, which will be grayed out to prevent being changed. The checkbox text changes to “check”, and clicking on the checkbox will update the detected VDD voltage value. If *Target VDD>Auto-Detect* is selected, clicking on the checkbox will revert the VDD mode back to PICkit 2 supplied VDD if a target power supply is no longer detected.

FIGURE 1-5: TARGET SUPPLIED VDD



1.4.6 Device $\overline{\text{MCLR}}$ State

The “/MCLR” checkbox shown in Figure 1-4 and Figure 1-5 has the same functionality as the menu selection *Programmer>Hold Device in Reset*. When the box is checked the target device will be held in Reset. When unchecked, the target circuit is allowed to pull $\overline{\text{MCLR}}$ up to VDD to release the device from Reset. This function can be used to prevent a device from executing code before and after programming.

Note: If the target device allows the $\overline{\text{MCLR}}$ pin to be configured as an input port, and it is configured as such, PICkit 2 will not be able to hold the device in Reset.

1.4.7 Memory Source

The Source bar displays the source of the currently loaded device data. If read from a hex file, it will display the hex file name. If read from a device, it will display the part name. `None (Empty/Erased)` indicates the buffers are empty, and it will display `Edited` once Program Memory or Data EEPROM Memory has been edited in the window.

1.4.8 Program Memory

Program code can be loaded into the PICkit 2 Programmer application by selecting *File>Import HEX* to import a hex file or by clicking **Read** to read the device memory. The origin of the code is displayed in the Source block. The Program Memory window displays the program code in hexadecimal. The code may be edited in the window.

The checkbox next to the Program Memory window is only available on devices with EEPROM data memory. If the box is checked, then Program Memory, User IDs, and Configuration Words are written to, read from, and verified on the device. If the box is unchecked, then Program Memory, User IDs, and Configuration Words will not be erased or altered during a Write Device operation, and will not be read or verified. The checkbox does not affect Erase Device or Blank Check operations. Both memory window checkboxes may not be cleared at the same time.

For supported serial EEPROM devices, the device contents are displayed in the Program Memory window instead of the Data EEPROM Memory window for easier viewing in the larger display area.

1.4.9 Data EEPROM Memory

Similar to Program Memory above, data EEPROM code can be loaded into the PICkit 2 Programmer application by selecting *File>Import HEX* to import a hex file or by clicking **Read** to read the device memory. The origin of the code is displayed in the Source block. The Data EEPROM Memory window displays the program code in hexadecimal. The code may be edited in the window.

The check box next to the EEPROM Data window controls whether the EEPROM Data memory is written, read and verified. If the box is checked, then the device EEPROM will be overwritten with the window data. If the box is not checked, then the device EEPROM will not be erased or altered during a Write Device operation. The checkbox does not affect Erase Device or Blank Check operations. Both memory window checkboxes may not be cleared at the same time.

© 2008 Microchip Technology Inc.

Chapter 2. PICkit 2 Programmer Getting Started

2.1 INTRODUCTION

This chapter gives instruction on how to get started using the PICkit 2 Development Programmer/Debugger to program Flash-based PIC microcontroller units.

For information on how to use the PICkit 2 with In-Circuit Serial Programming™ (ICSP™), refer to **Chapter 3. "Using In-Circuit Serial Programming™ (ICSP™)"**.

For information on how to update the PICkit 2 operating system (firmware), refer to **Chapter 6. "Updating the PICkit 2 Operating System"**.

For information on using the PICkit 2 as a debugger in MPLAB IDE, see **Chapter 4. "PICkit 2 Debug Express"**.

- Installing the PICkit 2 Hardware
- Installing and Launching the PICkit 2 Programmer Application
- Connecting to the Device
- Selecting Target Power
- Importing a Hex File
- Writing the Program to the Device
- Verifying the Device
- Reading Device Memory
- Code Protecting the Device
- Erasing and Blank Checking the Device
- Automating Write/Read Procedures

2.2 INSTALLING THE PICkit 2 HARDWARE

To install the PICkit 2 hardware:

- Plug one end of the USB cable into PICkit 2 USB connector. Plug the other end into a USB port on your PC.
- Connect the PICkit 2 to a target board via a 6-pin connector. The target board can be the included demo board or any target equipped with the appropriate 6-pin connector.
- Do not connect the PICkit 2 to a target board that has its own power supply if it is not connected to a powered USB port.
- To connect the PICkit 2 to a target with an MPLAB ICD 2 style RJ-11 connector, the AC164110 RJ-11 to ICSP Adapter kit is required.

When plugging the PICkit 2 into the USB, it is recommended to disconnect it from any target board first. Similarly, when starting up or rebooting the host PC, ensure it is disconnected from a target.

For more information about the PICkit 2 hardware, see **Section 1.3 "PICkit 2 Development Programmer/Debugger"**.

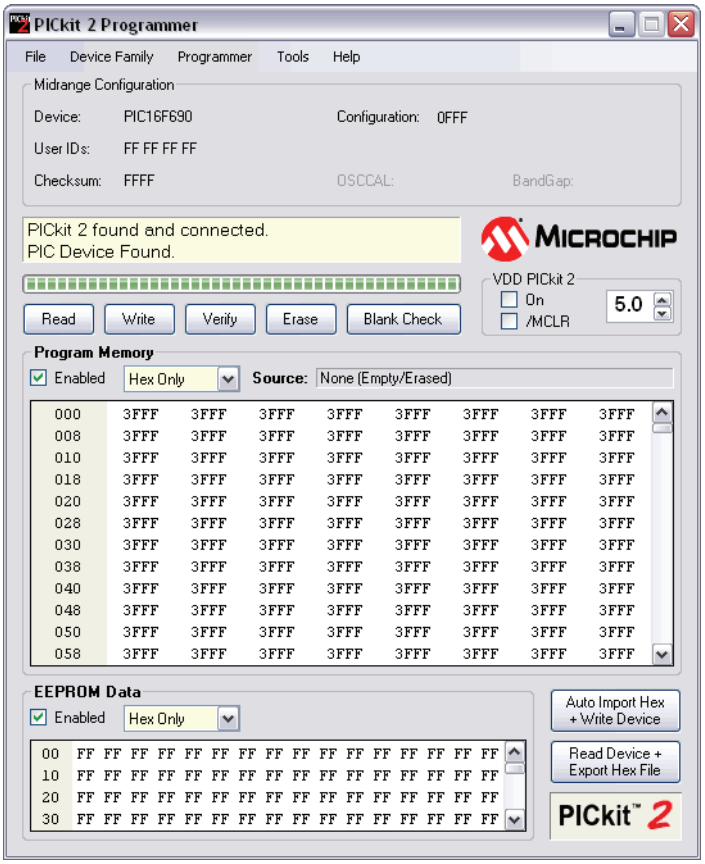
2.3 INSTALLING AND LAUNCHING THE PICkit 2 PROGRAMMER APPLICATION

Insert the PICkit 2 Starter Kit CD-ROM into the CD-ROM drive. In a few moments, the introductory screen should be displayed. Follow the directions on the screen for installing the PICkit 2 Programmer application.

If the introductory screen does not appear, browse to the CD-ROM directory and open the `PICkit_Starter_Kit_Welcome.htm` file.

Once installed, start the PICkit 2 Programmer application by selecting *Start>Programs>Microchip>PICkit 2*. The programming interface appears as shown in Figure 2-1. A listing of its features and functions may be found in **Section 1.4 "PICkit 2 Programmer Application"**.

FIGURE 2-1: PICkit™ 2 PROGRAMMING APPLICATION



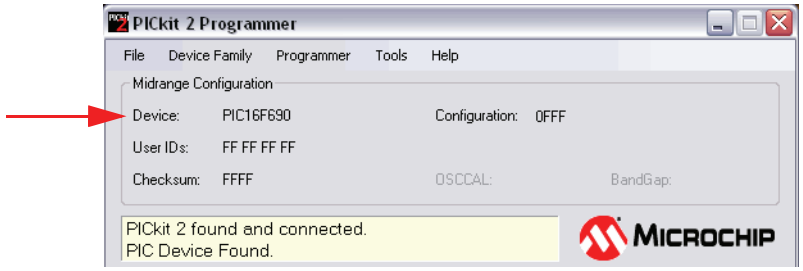
2.4 CONNECTING TO THE DEVICE

The PICkit 2 is capable of programming a variety of Flash-based Microchip PIC micro-controllers and serial EEPROM devices. Supported devices are listed in the PICkit 2 Readme file on the CD-ROM, which can also be viewed by selecting *Help>Readme*.

When the PICkit 2 Programmer application is first opened, it will attempt to identify the connected device by the device ID and display it in the Configuration window as shown in Figure 2-2.

© 2008 Microchip Technology Inc.

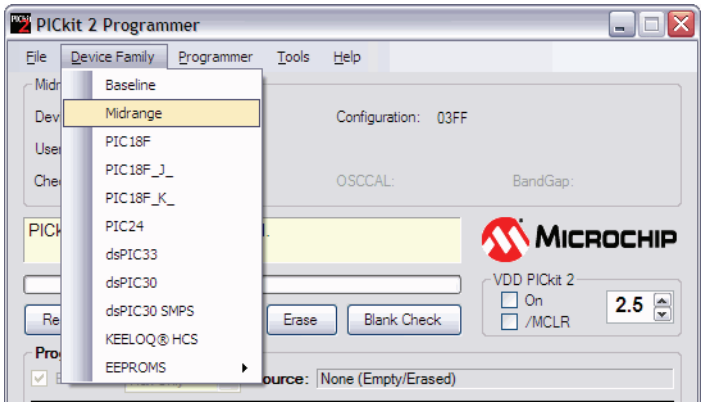
FIGURE 2-2: IDENTIFY DEVICE



If the device on the target is not correctly identified, check the target power (**Section 2.5 "Selecting Target Power"**) and device ICSP connections before attempting to reselect or change the device.

At any time, the device family may be selected to search for connectivity to a device in that family. To connect to a device once the application is already running, select the device family by clicking on the Device Family menu as shown in Figure 2-3.

FIGURE 2-3: SELECT DEVICE FAMILY

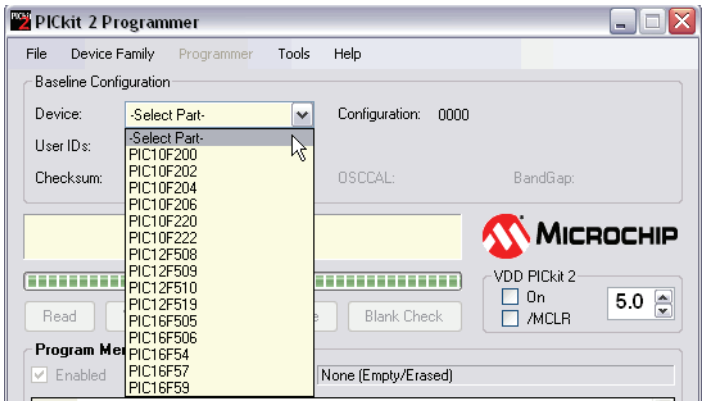


If the Baseline (12-bit core), KEELOQ® HCS or EEPROMs device family is selected, you must select the specific device from the device drop-down box as shown in Figure 2-4. These devices do not have a device ID and do not support automatic detection.

CAUTION

Ensure that the correct Baseline has been selected. These devices do not contain a device ID to confirm device selection. Choosing the wrong Baseline may cause an erasing of the OSCCAL value stored in the last memory location.

FIGURE 2-4: SELECT BASELINE FLASH DEVICE



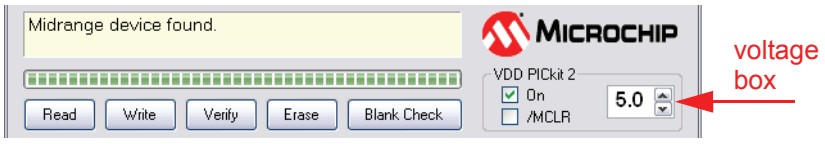
2.5 SELECTING TARGET POWER

The PICkit 2 can supply power to the target or the target may be powered externally.

2.5.1 Target Powered from PICkit 2

If you are going to power the target board from the PICkit 2, do not attach a power supply to the target or the PICkit 2 will sense it and not give you the option to use PICkit 2 power. For a target board not connected to an external power supply, you will see the options displayed in Figure 2-5.

FIGURE 2-5: ENABLE POWER FROM PICkit™ 2



To enable power to the target device, check the VDD PICkit 2 "On" checkbox as shown. The default setting is "Off", i.e., the checkbox is unchecked.

Note: If a target power supply is not detected, the PICkit 2 will always supply power to the target during programming, regardless of the VDD PICkit 2 "On" checkbox state.

The voltage supplied to the target may be adjusted before or after enabling power by adjusting the VDD PICkit 2 voltage box (Figure 2-5).

If a short or heavy current load is detected on the programmer-supplied VDD, then you will receive an error and VDD will be automatically disabled. Refer to Figure 2-6.

CAUTION

The USB port current limit is set to 100 mA. If the target plus the PICkit 2 exceed this current limit, the USB port may turn off. The target may be powered externally if more power is required.

© 2008 Microchip Technology Inc.

FIGURE 2-6: VDD ERROR



To avoid heavy current load errors, it is recommended to keep the target current consumption below 25 mA. Large VDD capacitances should also be avoided as they slow down the VDD risetime. The allowed VDD rise time is 500 µs or less.

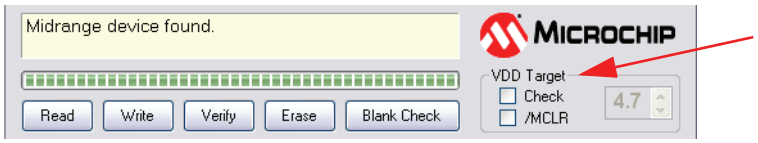
2.5.2 Target Powered from External Supply

The target device may also be powered externally. By default, the PICkit 2 will automatically detect an externally powered board. The heading "VDD PICkit 2" will be changed to "VDD Target", the "On" checkbox will be replaced by a checkbox named "Check", and the detected VDD voltage is displayed in the grayed out voltage box as in Figure 2-7.

Clicking the "Check" checkbox will update the detected VDD voltage displayed in the voltage box. If no VDD voltage is detected when the checkbox is clicked, then PICkit 2 will return to supplying VDD power to the target device.

Note: The maximum external VDD that may be used with the PICkit 2 is 5.0 Volts. The minimum external VDD that may be used with the PICkit 2 is 2.5 Volts.

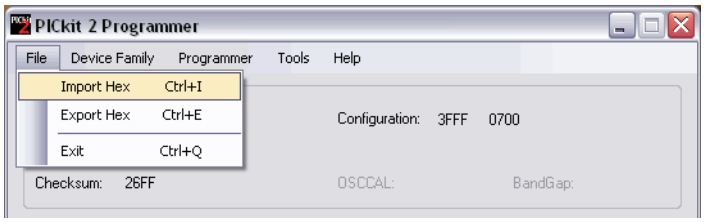
FIGURE 2-7: EXTERNALLY POWERED TARGET



2.6 IMPORTING A HEX FILE

To import a compiled program (hex file) to be programmed into the target device, select *File>Import HEX* as shown in Figure 2-8.

FIGURE 2-8: IMPORT HEX FILE

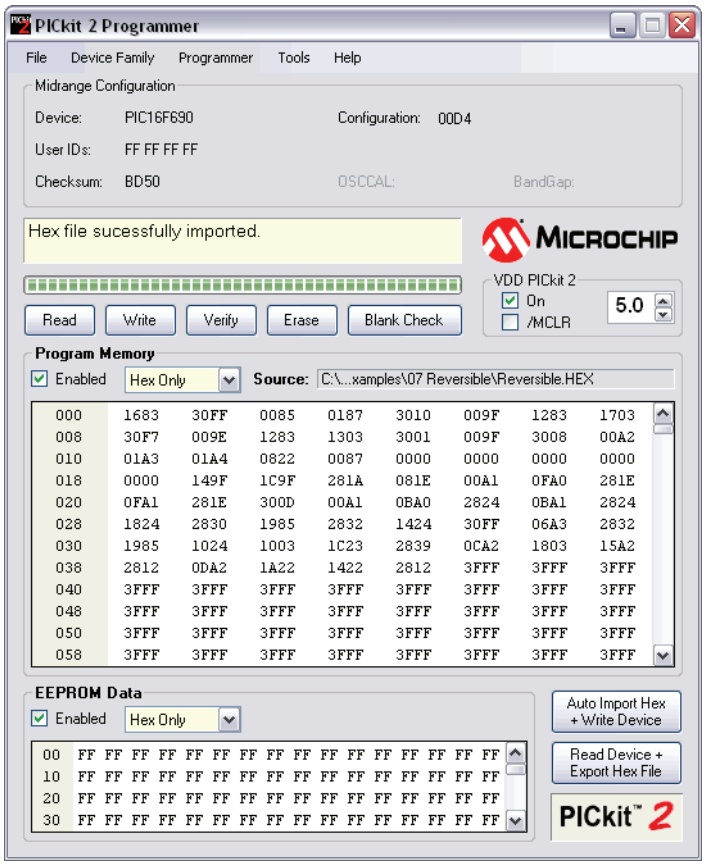


Example source code and hex files may be found under the `Install\Lessons\` directory for the appropriate kit demo board on the PICkit 2 Starter Kit CD-ROM. The hex file `Reversible.hex` from the folder `07 Reversible` will be used.

Browse for the hex file and click **Open**. The code is displayed in the Program Memory and EEPROM Data windows. The name of the hex file is displayed in the Source block under Program Memory.

Note: The example lesson code is meant to be installed to the local hard drive from the CD-ROM using the Setup files in the CD-ROM directory `Install/Lessons/`. The lessons are covered in the user's guide for the included kit Demo Board, and this guide should be read before using the lesson projects and source code.

FIGURE 2-9: EXAMPLE HEX FILE IMPORTED



The PICkit 2 Programmer application will warn you if the hex file does not contain any Configuration Words. You can be sure these are included in the hex file by selecting *File>Export* in MPLAB IDE to save a hex file including configuration memory. (For MPLAB IDE usage, see **Chapter 4. "PICkit 2 Debug Express"**.)

You will also be warned that the hex file is larger than the selected device if the hex file contains memory locations that do not exist in the current device. Any data for non-existent locations will not be imported.

2.7 WRITING THE PROGRAM TO THE DEVICE

After a device family has been selected and a hex file has been imported, the target device can be programmed by clicking **Write** (Figure 2-10). The device will be erased and programmed with the hex code previously imported.

© 2008 Microchip Technology Inc.

When erasing the device during programming, a Bulk Erase method is used. All Baseline, Mid-Range, and many dsPIC30F and PIC18F devices require a minimum VDD for the Bulk Erase. Some of these devices support a low voltage row erase method that can be used at lower voltages, but this method takes longer to erase the device. See the Readme file under *Help>Readme* for devices that support this feature. If a device does not support row erasing, a dialog will pop up to warn you if the device VDD is below the minimum required for a Bulk Erase.

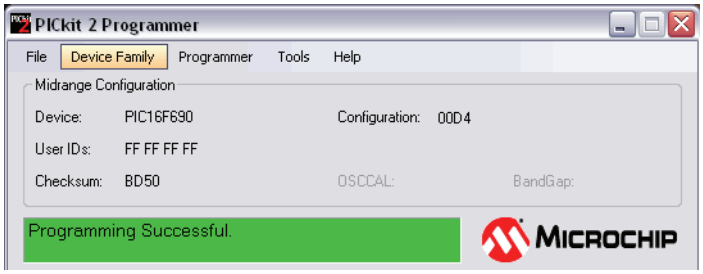
Note: If any Code Protect, Data Protect, Write Protect, or Read Protect configuration bits are currently set in the device, the Bulk Erase method must be used prior to programming. The lower voltage row erase procedure will not succeed.

FIGURE 2-10: BUTTONS – WRITE



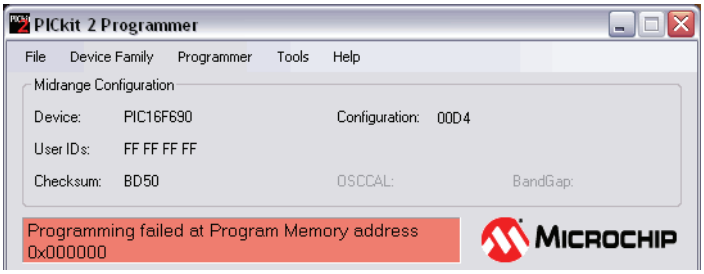
The status of the Write operation is displayed in the status bar located under the Device Configuration window. If the write is successful, the status bar turns green and displays "Programming Successful", as shown in Figure 2-11.

FIGURE 2-11: WRITE SUCCESSFUL STATUS



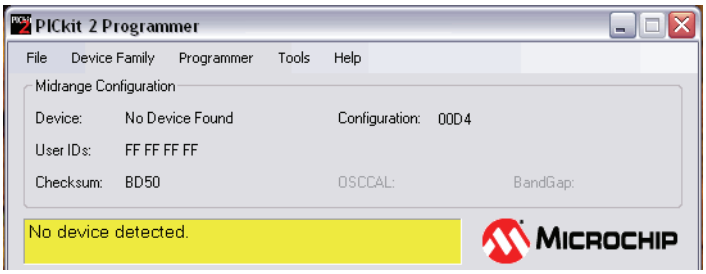
If the write fails, the status bar turns red and displays "Programming Failed", as shown in Figure 2-12. This error indicates that the data was corrupted during the programming sequence. If this error is displayed, try writing the program to the device again. If the error continues, see **Chapter 5. "Troubleshooting"** for assistance.

FIGURE 2-12: WRITE ERROR STATUS



Other write issues may be displayed as warnings and will turn the status bar yellow as in Figure 2-13. In this case, the PICkit 2 and demo board had become disconnected.

FIGURE 2-13: WRITE WARNING STATUS



2.7.1 Writing to Specific Memory Regions

If a device has EEPROM data memory, the "Enabled" checkbox next to Program Memory and EEPROM Data will become available.

The checkboxes select which memory regions' programming operations will be affected. Refer to Table 2-1 for a description of how programming operations are affected by the checkboxes. Erase and Blank Check always operate on all memory regions.

TABLE 2-1: MEMORY REGION SELECTION

Program Memory Enabled	EEPROM Data Enabled	Write/Read/Verify	Erase/Blank Check
Checked	Checked	All Memory Regions	All Memory Regions
Checked	—	Program Memory User IDS Configuration	All Memory Regions
—	Checked	EEPROM only	All Memory Regions
—	—	*Not Allowed*	

During a Write, regions that are unchecked will remain unchanged in the device.

For example, if Program Memory is unchecked while EEPROM Data is checked, then a Write operation will only write EEPROM Data, while Program Memory, User IDs and Configuration Words in the device will remain unchanged.

If Program Memory is checked while EEPROM Data is unchecked, then a Write operation will program Program Memory, User IDs and Configuration Words, while EEPROM Data in the device will remain unchanged.

Due to programming constraints in some devices, the PICkit 2 Programmer application may read and re-write EEPROM data memory during a Write to preserve it.

It is not allowed to have both memory regions unchecked.

2.7.2 Automatic File Reload

Prior to each Write, the imported hex file time stamp is compared to the version on the disk. If the version on the disk is newer, it is reloaded. This occurs only when a hex file has been read from the disk.

This feature ensures that the latest version built will be written to the device. It may be used with the *Tools>Program on PICkit Button* feature to program the latest MPLAB IDE build without switching to the PICkit 2 Programmer software simply by pressing the PICkit 2 unit push button.

© 2008 Microchip Technology Inc.

2.8 VERIFYING THE DEVICE

The Verify function verifies that the program in device memory matches the hex file imported into the PICkit 2 Programmer application. It compares all areas of memory including program memory, data EEPROM memory, ID and Configuration bits.

To verify the code, import the hex file and click **Verify** (Figure 2-14).

Note that a Write operation is automatically verified if *Programmer>Verify on Write* is checked.

FIGURE 2-14: BUTTONS - VERIFY



If the code is the same, the status bar turns green and displays “Device Verified”. If a discrepancy is found, the status bar turns red and displays where the error is located: “Error in Program Memory, Data EEPROM Memory, or Configuration Bits”.

Table 2-1 illustrates how Verify is affected by the memory region checkboxes.

2.9 READING DEVICE MEMORY

To view the code written to the device, click **Read** (Figure 2-15).

FIGURE 2-15: BUTTONS - READ



The code is displayed in the Program Memory and EEPROM Data windows for your review. If all zeros are displayed, it is possible that the device is code-protected (See **Section 2.10 “Code Protecting the Device”**.)

Table 2-1 illustrates how Read is affected by the memory region checkboxes.

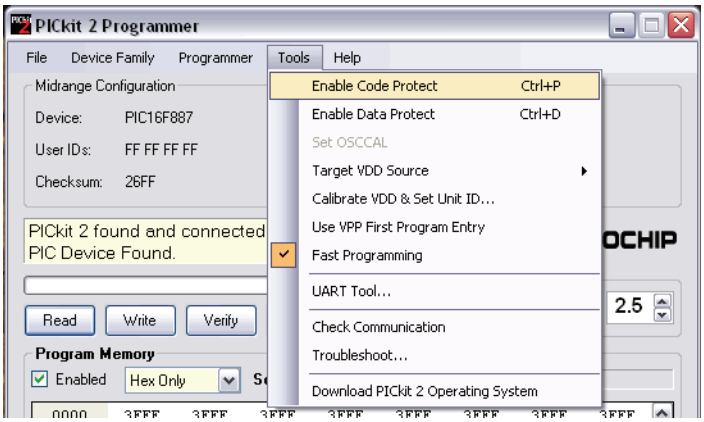
2.10 CODE PROTECTING THE DEVICE

The Code and Data Protect functions enable the read protection features of the device. To protect the program memory code, complete the following steps:

1. Import hex file.
2. Select *Tools>Enable Code Protect* as shown in Figure 2-16.
3. Click **Write**.

Devices that have EEPROM data memory may protect it by selecting *Tools>Enable Data Protect*.

FIGURE 2-16: ENABLE CODE PROTECT



Note: If the device is read after it has been protected, the protected memory regions will display all zeros.
Simply unchecking "Enable Code Protect" will not allow you to read the region. You must erase and reprogram all device memory before you can read that memory region again.

2.11 ERASING AND BLANK CHECKING THE DEVICE

The Erase function erases the program memory, data EEPROM memory, ID and Configuration bits, regardless of the state of the Program Memory and EEPROM Data "Enabled" checkboxes. However, this function is not normally needed since the Write function performs an erase operation prior to programming the device.

To erase the device, click **Erase** (Figure 2-17).

Note: The PICkit 2 Erase function always uses the Bulk Erase method that requires a minimum VDD, even on devices that support row erasing for the Write function. You will be warned if VDD is below the minimum for the connected device.

FIGURE 2-17: BUTTONS – ERASE



The Blank Check function will read the entire device to determine if Program Memory, EEPROM Data memory, User IDs and Configuration bits are erased. All memory regions will be examined, regardless of the state of the Program Memory and EEPROM Data "Enabled" checkboxes.

To Blank Check the device, click **Blank Check** (Figure 2-17).

2.12 AUTOMATING WRITE/READ PROCEDURES

The PICkit 2 Programmer application has two buttons for automating multiple functions.

© 2008 Microchip Technology Inc.

FIGURE 2-18: AUTOMATING BUTTONS



2.12.1 Auto Import Hex + Write Device Button

This features allows the PICkit 2 Programmer application to automatically import a hex file and write it to a connected device when the hex file is updated; for example, on a new firmware build.

To use this feature, click **Auto Import Hex + Write Device**. This will bring up an Import Hex file dialog defaulting to the first hex file in the file history under the File menu. After selecting a file, it will be written to the device. The PICkit 2 Programmer application will now monitor the selected hex file for updates. When the file has been updated (has a newer time stamp), the application will automatically re-import the hex file and write to the target device.

While this feature is enabled, other programming operations are disabled. The **Auto Import Hex + Write Device** button will remain depressed while this feature is active. To stop using this feature, click **Auto Import Hex + Write Device** again.

If an error is encountered during hex file importing or device programming, the application will automatically exit this feature mode.

2.12.2 Read Device + Export Hex File Button

When clicked, this button will read the target device and open an **Export Hex File** dialog.

2.13 PICKIT 2 CALIBRATION AND UNIT ID

The PICkit 2 VDD may be calibrated to account for variations in the unit hardware and the USB voltage of the port it is connected to. A Unit ID string may also be assigned to a PICkit 2 unit to provide unique identification.

2.13.1 Calibrating the PICkit 2 VDD

Calibration allows greater accuracy both in the VDD voltage supplied to the target from PICkit 2, and the voltage detected on a powered target and reported in the software. The calibration is stored in the PICkit 2 unit nonvolatile memory, so the unit will remain calibrated even when used from within MPLAB IDE.

It is important to note, however, that as the calibration is dependent on the USB voltage, moving the PICkit 2 unit to a different USB port, to or from a USB hub or to another computer port may invalidate the calibration.

To calibrate the PICkit 2 unit, a multi-meter or other means of accurately measuring voltages is required. Disconnect the PICkit 2 unit from the target and select *Tools>Calibrate Vdd & Set Unit ID…* to bring up the calibration wizard. Follow the steps in the wizard to complete the calibration process.

Note:	Even when calibrated, the PICkit 2 VDD voltage is limited by the available USB port voltage. PICkit 2 is unable to provide a VDD voltage greater than a Schottky diode drop below the USB port voltage. This voltage can be as low as 4.2 Volts, especially in laptop computer ports.

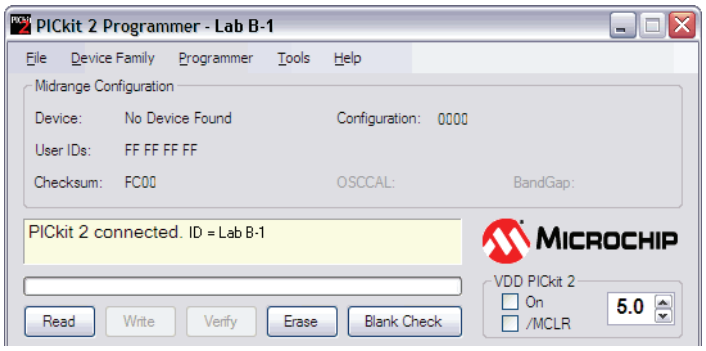
2.13.2 The PICkit 2 Unit ID

During the calibration procedure, the PICkit 2 may optionally be assigned a Unit ID string to identify it uniquely.

Once assigned, the PICkit 2 Unit ID will display in the PICkit 2 Programmer software title bar, and in the Status Window when first connecting to the PICkit 2. An example is shown in Figure 2-19 where the Unit ID is "Lab B-1".

The Unit ID is also displayed in the MPLAB IDE Output window when first selecting PICkit 2 as a Programmer or Debugger.

FIGURE 2-19: UNIT ID



© 2008 Microchip Technology Inc.

Chapter 3. Using In-Circuit Serial Programming™ (ICSP™)

3.1 INTRODUCTION

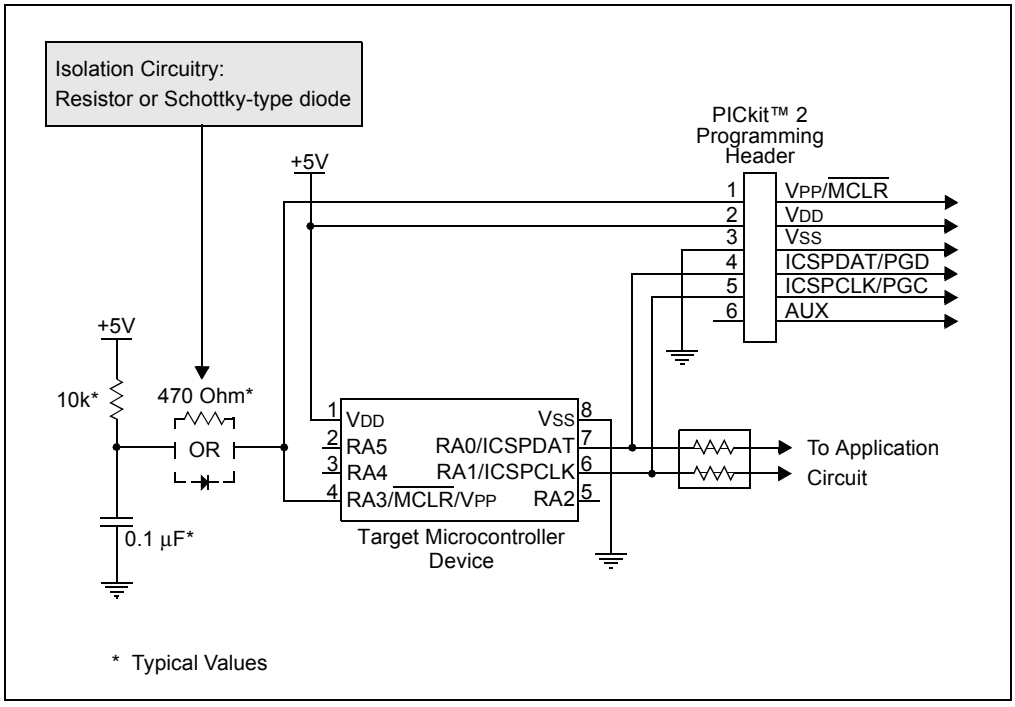
The PICkit 2 Development Programmer/Debugger can program microcontroller devices that are installed in an application circuit using In-Circuit Serial Programming (ICSP). ICSP requires five signals:

- VPP – Programming Voltage; when applied, the device goes into Programming mode.
- ICSPCLK or PGC – Programming Clock; a unidirectional synchronous serial clock line from the programmer to the target.
- ICSPDAT or PGD – Programming Data; a bidirectional synchronous serial data line.
- VDD – Power Supply positive voltage.
- VSS – Power Supply ground reference.

However, the application circuit must be designed to allow all the programming signals to be connected to the device without distorting the programming signals. Figure 3-1 shows a typical circuit as a starting point when designing an application circuit for ICSP. For successful ICSP programming, the precautions in the following sections need to be followed.

Note: For details on how a specific device is programmed, refer to the device programming specification available from the Microchip web site at www.microchip.com.

FIGURE 3-1: TYPICAL ICSP™ APPLICATION CIRCUIT



3.2 ISOLATE VPP/$\overline{\text{MCLR}}$/PORT PIN

When VPP voltage is applied, the application circuit needs to take into consideration that the typical VPP voltage is +12V. This may be an issue in the following situations:

If the VPP pin is used as a $\overline{\text{MCLR}}$ pin

The application circuit is typically connected to a pull up resistor/capacitor circuit, as recommended in the device data sheet. Care must be taken so that the VPP voltage slew rate is not slowed down and exceeds the rise time in the programming specification (typically 1 μs).

If a supervisory circuit or a push button is interfaced to the $\overline{\text{MCLR}}$ pin, it is recommended that they be isolated from the VPP voltage by using a Schottky-type diode or limiting resistor as shown in Figure 3-1. For more information about using supervisory circuits with ICSP, see Application Note AN820 "*System Supervisors in ICSP™ Architectures*" (DS00820).

If the VPP pin is used as an I/O port pin

The application circuit that connects to the I/O pin may not be able to handle the +12V voltage. It is recommended to use a Schottky-type diode or limiting resistor as shown in Figure 3-1 to isolate the circuitry.

3.3 ISOLATE ICSPCLK OR PGC AND ICSPDAT OR PGD PINS

The ICSPCLK or PGC and ICSPDAT or PGD pins need to be isolated from the application circuit to prevent the programming signals from being affected by the application circuitry. ICSPCLK or PGC is a unidirection synchronous serial programming clock line from the programmer to the target. ICSPDAT or PGD is a bidirectional synchronous serial programming data line.

If the design permits, dedicate these pins for ICSP. However, if the application circuit requires that these pins be used in the application circuit, design the circuitry in a manner that does not alter the signal level and slew rates. Isolation circuitry will vary according to the application. Figure 3-1 shows one possibility by using series resistors to isolate the ICSP signals from the application circuit.

© 2008 Microchip Technology Inc.

3.4 VDD

During ICSP programming, the device needs to be powered in accordance with the device specification. Typically, the device supply voltage is connected to the application circuit supply voltage. The application circuit can be powered by the PICkit 2 or externally. There are a few precautions that need to be observed in the situations covered in the following three sections.

3.4.1 The application circuit is powered by the PICkit 2

The PICkit 2 supply voltage may set between the maximum and minimum voltages allowed by the device programming specification, unless the minimum is below +2.5V. Be sure to set the voltage box to the appropriate voltage before programming the device or turning on VDD.

CAUTION

The USB port current limit is set to 100 mA. If the target plus the programmer exceeds this current limit, the USB port may turn off. The target may be powered externally if more power is required.

Note: Current draw should be limited to 25 mA when using the programmer to power the application circuit. Ensure that the application circuit does not slow the VDD rise time to longer than 500 μs.

3.4.2 The application circuit is powered externally

The PICkit 2 may be used with application circuits powered externally between +5.0V and +2.5V.

3.4.3 Bulk Erase is used

Some devices use a Bulk Erase function to erase program memory, data EEPROM memory, ID locations, and Configuration bits. Typically, the Bulk Erase function requires a supply voltage (VDD) of 4.5 to 5.5 Volts (refer to the device programming specification for device specific requirements).

This voltage range can be a problem if the application circuit is designed to operate at a different supply voltage range. In order to Bulk Erase the device, the application circuit needs to take into consideration the Bulk Erase voltage requirement while protecting any voltage sensitive circuitry.

If the application circuit VDD is below the minimum required for the Bulk Erase, a dialog will warn the user before attempting to erase the device.

3.5 Vss

The power supply ground reference, Vss, must be at the same potential as the application circuit.

3.6 CABLE LENGTHS

Minimize the distance the ICSP signals must travel by placing the ICSP connector as close to the application circuit device as possible. Minimize any cable length between the PICkit 2 and application circuit device. The goal is to keep the ICSP signals within the level and slew rate specifications for successful programming.

3.7 SERIAL EEPROM AND KEELOQ HCS DEVICES

The programming signals and connections for these devices are different than those for microcontrollers as described in **Section 3.1 "Introduction"** and Figure 3-1. See the PICkit 2 Programmer Readme file, *Help>Readme*, for programming signal connectivity for these devices.

Additionally, these devices are not intended to be programmed in-circuit. Attempting to program serial EEPROM devices while in-circuit may fail due to conflicts with other devices on the serial bus.

© 2008 Microchip Technology Inc.

Chapter 4. PICkit 2 Debug Express

4.1 INTRODUCTION

In addition to the PICkit 2 Programmer application, the PICkit 2 Development Programmer/Debugger may be used with MPLAB® IDE, the free integrated development environment available on Microchip's website. MPLAB IDE allows the PICkit 2 to be used as an in-circuit debugger as well as a programmer (for selected devices only).

In-circuit debugging allows you to run, examine and modify your program while the device is embedded in your target hardware. This greatly assists you in debugging your firmware and hardware together.

Special PICkit 2 Debug Express software interacts with the MPLAB IDE application to run, stop and single-step through programs. One or more breakpoints can be set and the processor can be reset. Once the processor is stopped, the register's contents can be examined and modified.

For more information on how to use MPLAB IDE, reference the following documentation:

- MPLAB® IDE User's Guide (DS51519)
- MPLAB® IDE Quick Start Guide (DS51281)
- MPLAB® IDE On-line Help

4.2 PICkit 2 DEBUG EXPRESS KIT CONTENTS

The PICkit 2 Debug Express kit (DV164121) contains the following items:

1. The PICkit 2 Development Programmer/Debugger
2. USB cable
3. 44-Pin Demo Board with device*
4. PICkit 2 Starter Kit and MPLAB IDE CD-ROMs

* The following boards may also be used to debug:

- To use the Low Pin Count Demo Board from the Starter Kit, the AC162061 ICD Header and AC164110 adapter will be required.

- To use the Explorer 16 board, be aware that the connector labeled "PICKIT2" incorrectly shows the location of PICkit 2 pin 1 on the silkscreen. That is, pin 1 is actually pin 6.

4.3 INSTALLING THE HARDWARE AND SOFTWARE

Install the PICkit 2 hardware as specified in **Section 2.2 "Installing the PICkit 2 Hardware"**.

Note:	Debug Express requires 4.7k Ohm pull down resistors on ICSPCLK and ICSPDAT. Newer PICkit 2 units have a Red button and have the pull downs internally. Older PICkit 2 units have a Black button and require that the pull downs be added on the target board.

Obtain a copy of the MPLAB IDE software application from the Microchip website or CD-ROM enclosed with the kit and install it as directed.

Note:	Debug Express requires MPLAB IDE version 7.50 or later.

4.4 USING PICKIT 2 DEBUG EXPRESS

4.4.1 Determining Device Support

For a list of current devices supported by PICkit 2 Debug Express, see the "`Readme for PICkit 2.htm`" file in the "Readmes" subdirectory of the MPLAB IDE installation directory.

When selecting a device as discussed in **Section 4.5 "Debug Express Tutorial"**, the "Select Device" dialog shown in Figure 4-11 shows the level of support for the selected device by Debug Express. In the "Debuggers" section of the dialog, the color of the circle next to "PICkit 2" indicates the support level:

Red – The device is not currently supported by PICkit 2 Debug Express

Yellow – The device has beta support in PICkit 2 Debug Express

Green – The device has full support in PICkit 2 Debug Express

Beta support indicates that the device is supported, but has not yet undergone internal certification testing by Microchip.

4.4.2 Reserved Resources

Due to the built-in in-circuit debugging capability of ICD devices and the ICSP function offered by the debugger, the PICkit 2 Debug Express uses some on-chip resources when debugging. It also uses program memory and file register locations in the target device during debugging. These locations are not available for use by user code. In MPLAB IDE, registers marked with an "R" in register displays represent reserved registers.

For information on device resources that are needed for in-circuit debugging, please refer to the MPLAB ICD 2 Help, found in the MPLAB IDE under *Help>Topics*. The device reserved resource information found under "Resources Used By MPLAB ICD 2" is the same for the PICkit 2 Debug Express.

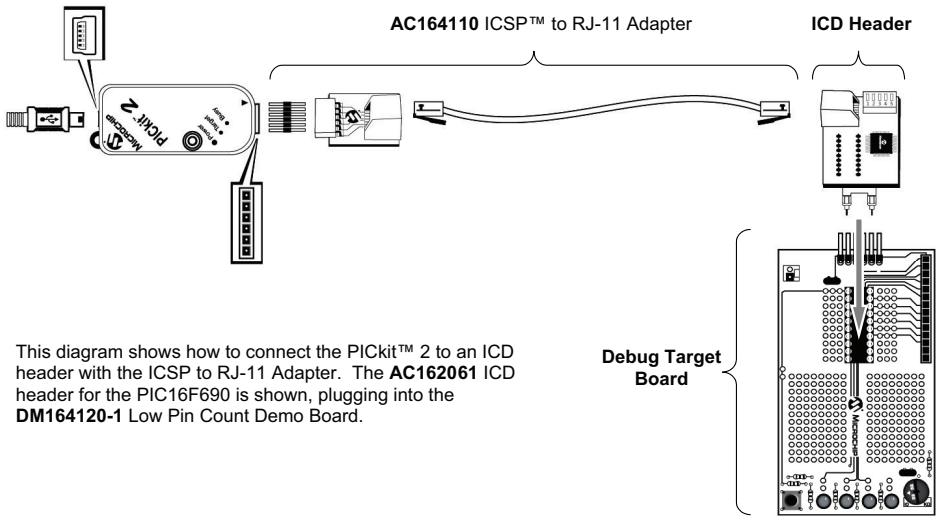
4.4.3 Using an ICD Header

All Baseline and some Mid-Range PIC microcontrollers require a special –ICD device mounted on a debug header circuit board to enable the debugging feature. For a list of these devices and the required ICD header board part number, please see the "*Header Board Specification*" (DS51292). The Header Board Specification is included on the PICkit 2 CD-ROM, and is available online at www.microchip.com.

© 2008 Microchip Technology Inc.

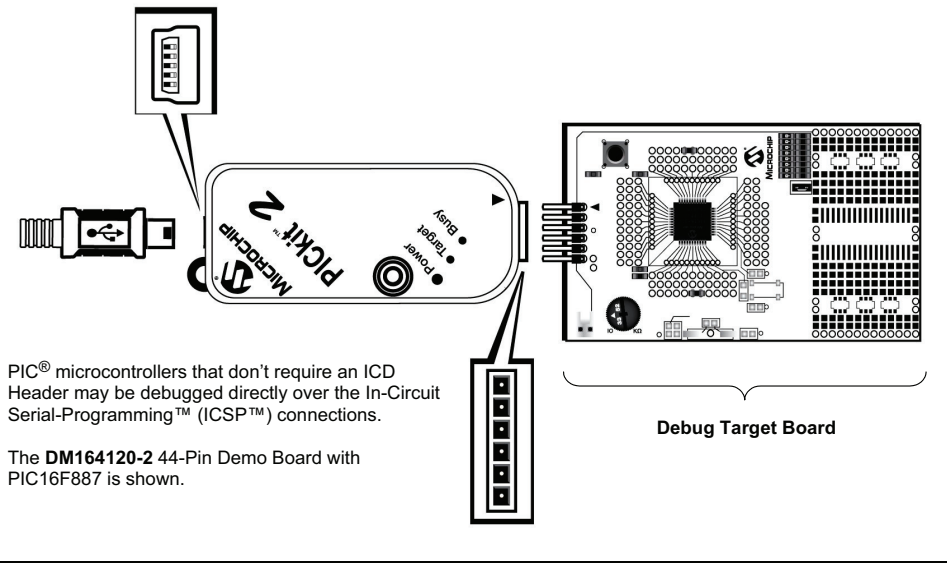
Each ICD header board comes with the necessary –ICD device, and is used on the target board instead of the production microcontroller. However, most header boards have an RJ-11 debug connector which requires the AC164110 RJ-11 to ICSP adapter kit to connect it to PICkit 2. Figure 4-1 illustrates using the AC162061 ICD Header for the PIC16F690 with the AC164110 adapter kit and Low Pin Count Demo Board.

FIGURE 4-1: USING THE PIC16F690 ICD HEADER BOARD



Many Mid-Range PIC microcontrollers and all PIC18 and 16-bit PIC microcontroller devices do not require an ICD header and can be debugged directly through the ICSP programming connections. This is true of the PIC16F887 included on the 44-Pin Demo Board, which can be debugged by simply connecting the demo board to the PICkit 2 as shown in Figure 4-2.

FIGURE 4-2: CONNECTING THE DEMO BOARD TO THE PICKIT™ 2



4.4.4 Configuration Bits and Debug Express

PIC microcontroller devices that do not require an ICD Header and may be debugged directly contain a DEBUG bit in the Configuration Word(s) that enables and disables the debug mode on the PIC microcontroller.

This bit is automatically set appropriately by the MPLAB IDE when using PICkit 2 Debug Express and should not be specified in source code configuration settings.

CAUTION

The DEBUG configuration bit value should not be specified in source code Configuration settings under normal conditions. Doing so may cause the bit to be asserted when programming a device outside the debugger. This will cause the device to function improperly or not all at in the application circuit.

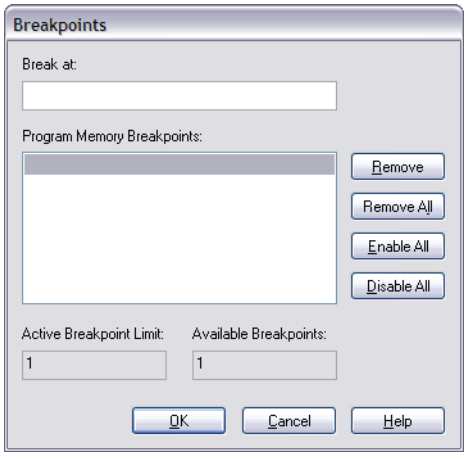
Many 16-bit PIC microcontroller devices such as PIC24 and dsPIC33 families have multiple ICSP programming and debugging port pins labeled PGC1/EMUC1 and PGD1/EMUD1, PGC2/EMUC2 and PGD2/EMUD2, etc. While any ICSP port may be used for programming, only one port is active at a time for debugging. The active EMU port is set in the device Configuration bits. If the active port setting does not match the EMU port to which the PICkit 2 is connected, the device will be unable to enter debug mode. In the MPLAB IDE Configuration Bits dialog, these bits are usually referred to as the "Comm Channel Select" bits.

4.4.5 Debug Express Breakpoints

The number of active breakpoints supported by PICkit 2 Debug Express depends on the target device. Most Baseline and Mid-Range devices support 1 breakpoint, with more breakpoints supported in some PIC18 and 16-bit devices.

The number of active breakpoints available for the current device in MPLAB IDE can be seen by selecting *Debugger>Breakpoints…*. This will open a dialog (Figure 4-3) showing any currently set breakpoints in Program Memory. The "Active Breakpoint Limit:" text box shows how many total breakpoints are available for the current device. The "Available Breakpoints:" text box shows how many breakpoints are currently unused.

FIGURE 4-3: BREAKPOINTS DIALOG FOR PIC16F887



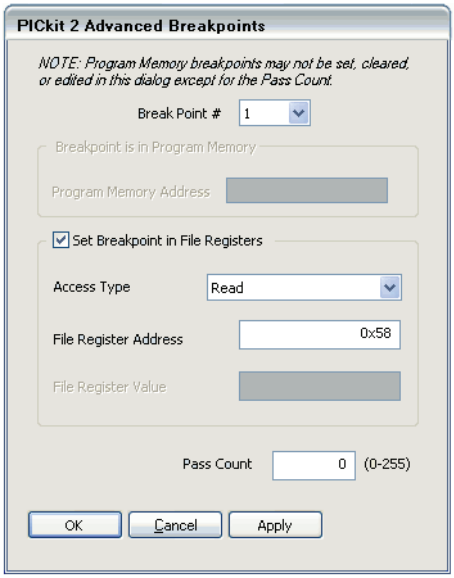
Some PIC18 and 16-bit devices also support advanced breakpoints. Advanced breakpoints allow breakpoints to be set in File Register memory, and will halt execution when a specific File Register is read from or written to. This breakpoint may also be set so it will only halt when a specific value is read from or written to a register. Additionally, a

© 2008 Microchip Technology Inc.

"Pass Count" may be set on any breakpoint type. The pass count is the number of times the breakpoint condition is met before it halts execution. For example, setting a pass count of "2" on a Program Memory breakpoint means that instruction will execute twice without halting execution, and the third time the instruction is executed the breakpoint will halt execution. The default pass count for all breakpoints is "0", which means execution will halt the first time the breakpoint is encountered.

If advanced breakpoints are supported by the current device, the MPLAB IDE menu option *Debugger>Advanced Breakpoints…* will be available to open the advanced breakpoint dialog. If the current device does not support advanced breakpoints, this menu option will be grayed out or absent. Select the breakpoint to edit with the "Break Point #" combo box.

FIGURE 4-4: ADVANCED BREAKPOINTS DIALOG



Note: The Advanced Breakpoint dialog will display any breakpoints already set in Program Memory. However, the dialog may not be used to set or clear breakpoints in Program Memory or to edit the address of an existing Program Memory breakpoint. Only the Pass Count value for Program Memory breakpoints may be edited in the Advanced Breakpoint dialog. To edit, set, or clear Program Memory breakpoints, use the MPLAB IDE editor or the *Debugger>Breakpoints…* menu dialog.

4.4.6 Breakpoint Skidding

The in-circuit debug implementation on PIC microcontrollers will halt execution on the instruction after the breakpoint instruction. This means the breakpoint instruction will have executed when the debugger halts. This is referred to as "breakpoint skidding".

As a result, there are some breakpoint behaviors to be aware of. When a breakpoint is set on a `GOTO`, `CALL`, or `RETURN` instruction, the debugger will halt at the destination instruction, as the program branch instruction with the breakpoint will have executed. Also, when using the debugger Step Over function, a breakpoint will be set on the instruction after the `CALL` instruction that the debugger is "stepping over" if there is an available breakpoint. If the `CALL` instruction is followed immediately by another `CALL` instruction, this will result in the debugger halting at the destination of the second `CALL` instruction. To prevent this, a `NOP` may be placed between the `CALL` instructions. Note that 16-bit devices will halt two instructions after the breakpoint instruction.

4.4.7 Linker Scripts

If your project uses a linker script, special ICD linker script files must be used when debugging that reserve the resources used by the PICkit 2 Debug Express. Each device has a separate debug linker file, which contains an "i" at the end of the device name.

For example:

`16F877i.lkr` – In-Circuit Debug linker file for the PIC16F877 device

`18F4520i.lkr` – In-Circuit Debug linker file for the PIC18F4520 device

When debugging with PICkit 2 Debug Express, the ICD linker file should be used instead of the standard linker file.

4.5 DEBUG EXPRESS TUTORIAL

This tutorial is intended to be used with the PICkit 2 44-Pin Demo Board and PIC16F887 microcontroller. This demo board comes with the PICkit 2 Debug Express kit and is available separately as part number DM164120-2.

If this demo board is not available, it is still recommended to read through this tutorial to get an overview of using the PICkit 2 as a debugger in the MPLAB IDE.

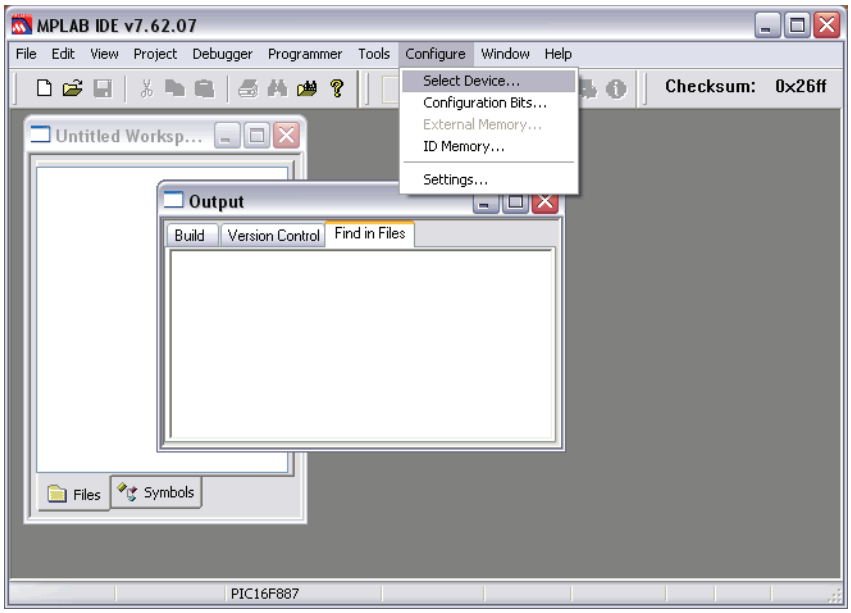
The source file used for the tutorial is installed with the PICkit 2 Programmer software.

4.5.1 Selecting the Device

To select a device in MPLAB IDE:

1. Launch the MPLAB IDE application.
2. From the MPLAB IDE menu bar, select *Configure>Select Device* (Figure 4-5).

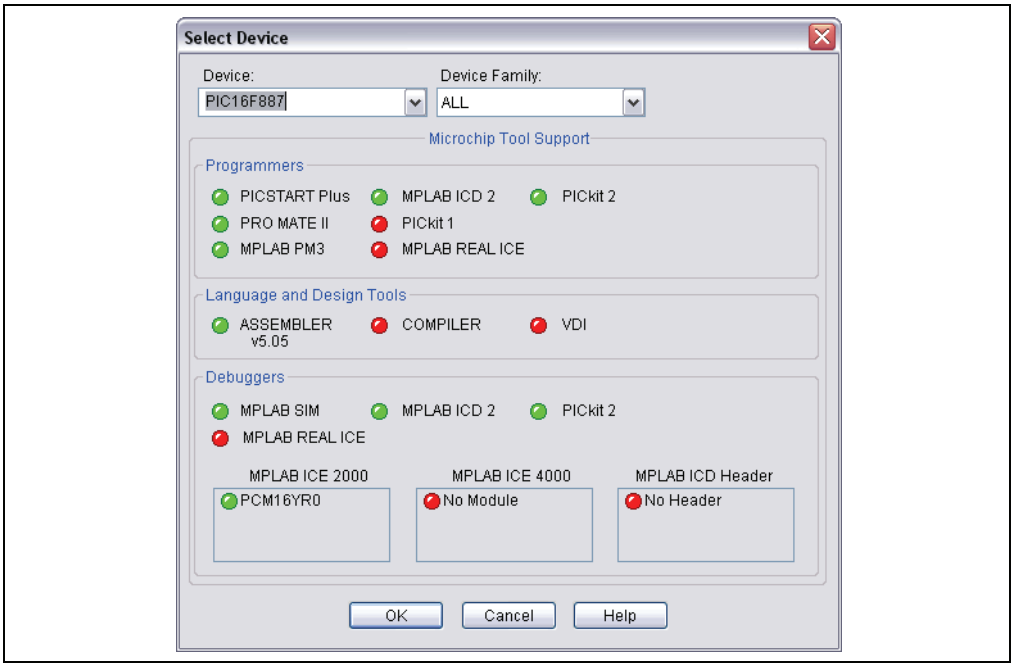
FIGURE 4-5: MPLAB® IDE MENU BAR



© 2008 Microchip Technology Inc.

3. In the Select Device dialog (Figure 4-6), click on the “Device” drop-down list and select your device. In this tutorial, choose the PIC16F887 device.

FIGURE 4-6: SELECT DEVICE DIALOG

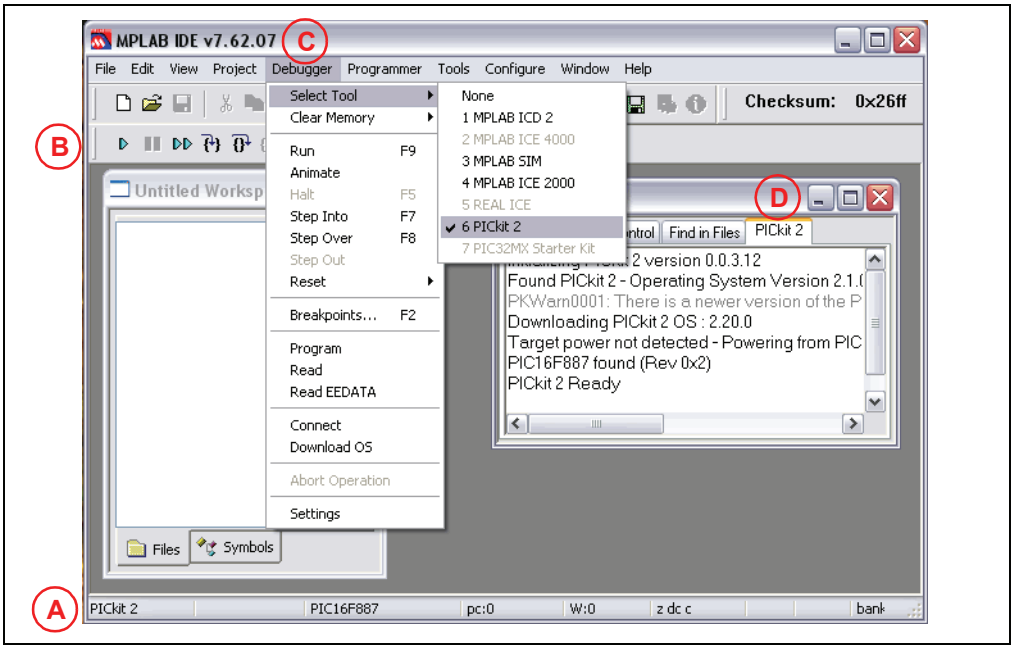


4. No other changes need to be made in this dialog box. Click **OK**.

4.5.2 Selecting PICkit 2 as the Debug Tool

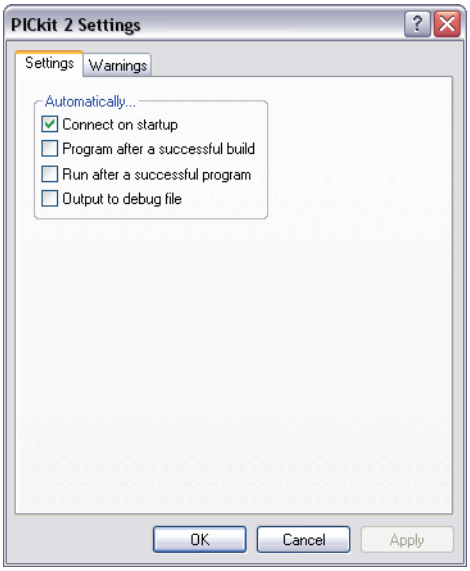
1. Select *Debugger>Select Tool>PICkit 2*. MPLAB IDE will add PICkit 2 debug features (Figure 4-7): (A) the status bar will show PICkit 2 as the debug tool, (B) a PICkit 2 debug toolbar will be added, (C) the Debugger menu will change to add PICkit 2 debug functions and (D) the Output window will display communication status between the PICkit 2 and the target board on the **PICkit 2** tab.

FIGURE 4-7: PICkit™ 2 DEBUG TOOL



2. Select *Debugger>Settings* to set up PICkit 2 operation. Make sure the "Connect on Startup" checkbox is checked to enable the auto-connection feature (Figure 4-8). Then click **OK**.

FIGURE 4-8: PICkit™ 2 SETTINGS DIALOG



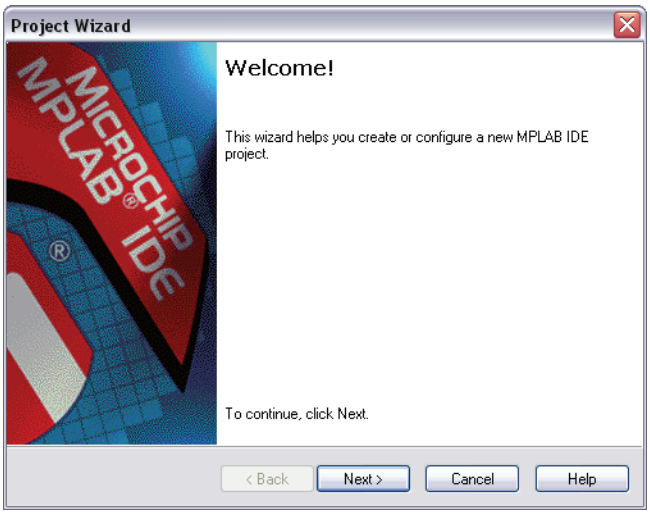
3. If the PICkit 2 did not connect automatically when it was selected as the debug tool, select *Debugger>Connect* to connect now. The connection status will be visible in the Output window. Depending on the version of the MPLAB IDE software or the selected device, a message may appear indicating that the firmware (PICkit 2 operating system) needs to be updated. MPLAB IDE will automatically install new firmware.

4.5.3 Creating an MPLAB IDE Project

An MPLAB IDE project and workspace keep all files and settings for a development project together. The Project Wizard helps you set up a new project.

1. Select *Project>Project Wizard* to set up the project. The Project Wizard Welcome screen will display (Figure 4-9). Click **Next** to continue to Step One.

FIGURE 4-9: PROJECT WIZARD WELCOME



© 2008 Microchip Technology Inc.

2. Select the **PIC16F887** device from the “Device” drop-down box, if it is not already selected (Figure 4-10). Click **Next** to continue to Step Two.

FIGURE 4-10: STEP ONE – SELECT DEVICE



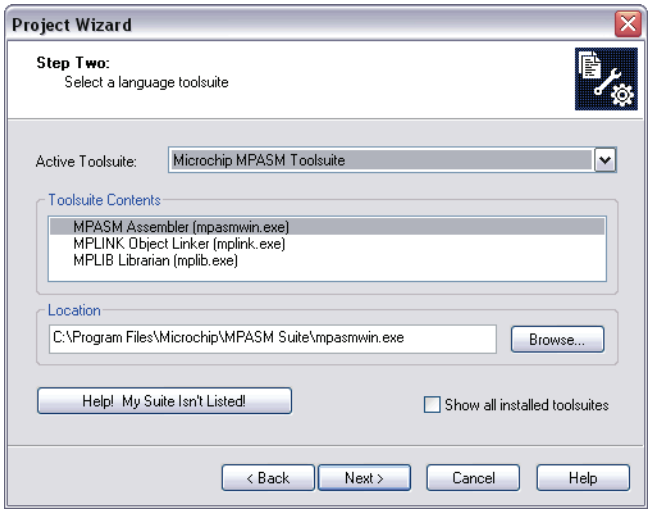
3. For this project, the MPASM™ assembler tool will be used. Select “Microchip MPASM Toolsuite” from the Active Toolsuite drop-down menu (Figure 4-11).

   Make sure the tools are set to the proper executables by default in the `C:\Program Files\Microchip\MPASM Suite` folder as follows:

   - MPASM assembler should be pointing to mpasmwin.exe.
   - MPLINK™ object linker should be pointing to mplink.exe.
   - MPLIB™ object librarian should be pointing to mplib.exe.

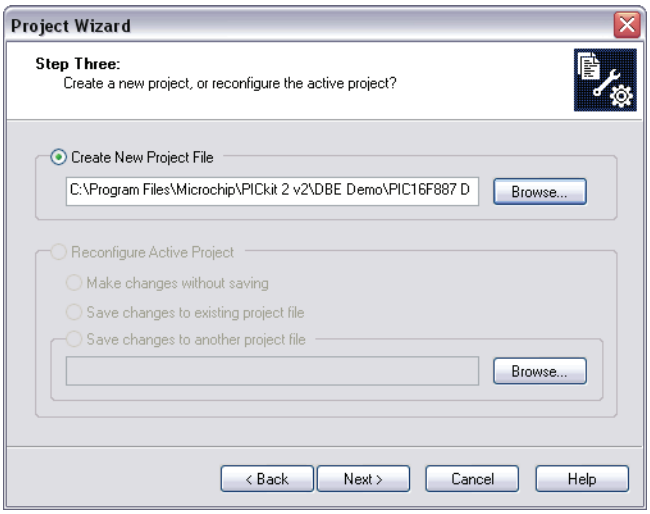
   Click **Next** to continue to Step Three.

FIGURE 4-11: STEP TWO – SELECT LANGUAGE SUITE



4. Click **Browse** (Figure 4-12) to locate or create a new project directory and project name. For this tutorial, select the location `C:\Program Files\Microchip\PICkit 2 v2\DBE Demo` and give the project file a name, such as "`PIC16F887 Debug Demo`".

FIGURE 4-12: STEP THREE – CREATE NEW PROJECT



5. Add the project source file (Figure 4-13). From the left pane window, go to `C:\Program Files\Microchip\PICkit 2 v2\DBE Demo`. Select and highlight the `16F887Demo.asm` file and click **Add**. The file will be placed into the right pane window

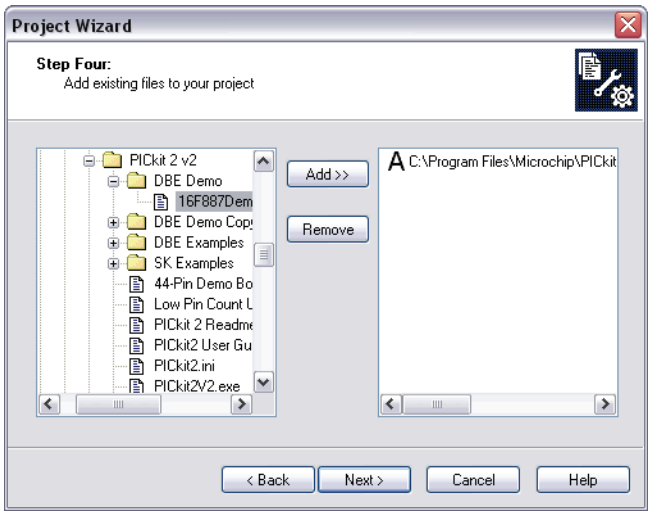
Note:	Other files can be added later.

The "A" allows MPLAB IDE to decide whether the path to the file should be relative or absolute for the project. For a description of and how to change to other possible file-addition modes, see the MPLAB IDE documentation. Do not change the setting for this project. Click **Next** to continue to the Summary window.

Note:	For projects containing more than one assembly file (e.g., multifile projects, C code project) you will also need to add a linker script file. See the language tool documentation for more details.

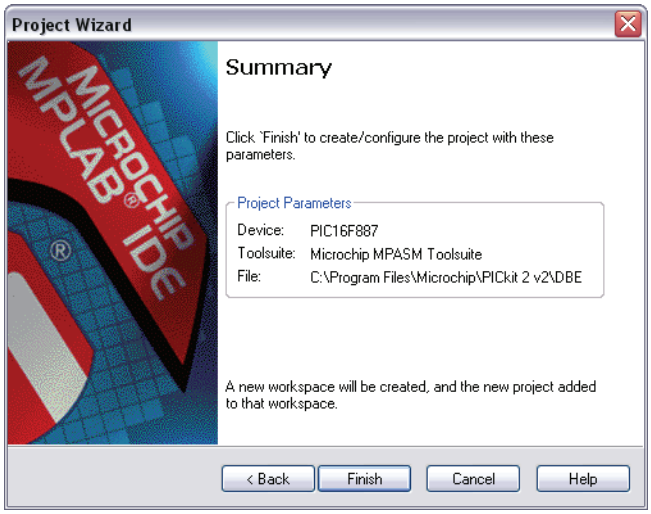
© 2008 Microchip Technology Inc.

FIGURE 4-13: STEP FOUR – ADD FILES



6. If any errors have been made, click **Back** to return to any of the previous steps in the Project Wizard (Figure 4-14). Otherwise, click **Finish**.

FIGURE 4-14: PROJECT WIZARD SUMMARY

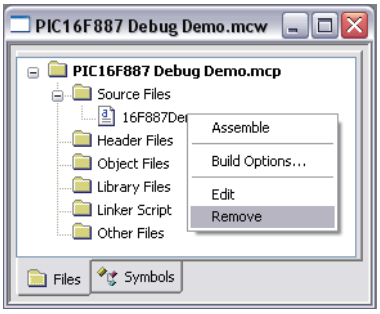


4.5.4 Viewing the Demo Project

After completing the project setup and exiting the Project Wizard, the Project window will display in the MPLAB IDE desktop window (Figure 4-15). If it is not open, select *View>Project* to open it.

If needed, files can be added to or removed from the project using the Project Window. Right click on the file in the Project Window tree to display a pop-up menu with options that include adding or removing files.

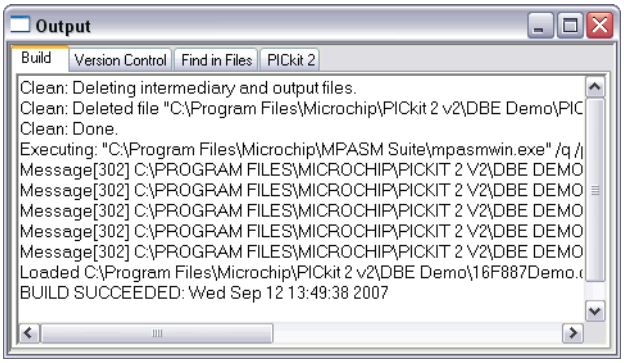
FIGURE 4-15: PROJECT WINDOW FILE MENU



4.5.5 Creating a Hex File

To create a hex file for programming the device, you need to build the project. Select *Project>Build All*, or right click on the project name in the Project Window and select "Build All" from the pop-up menu. The MPASM assembler will create a hex file with the same name as the source `.asm` file. The assembler's progress will be visible in the **Build** tab of the Output window (Figure 4-16).

FIGURE 4-16: OUTPUT WINDOW – BUILD THE PROJECT



4.5.6 Checking Configuration Bit Values

The Configuration bits that will be programmed into the device are set from within the program using `__CONFIG` directives. Once the project is built, the values of these bits can be verified using the Configuration Bits window. Select *Configure>Configuration Bits* to open the window (Figure 4-17.)

The following Configuration bits should now be set for this tutorial:

Config1:

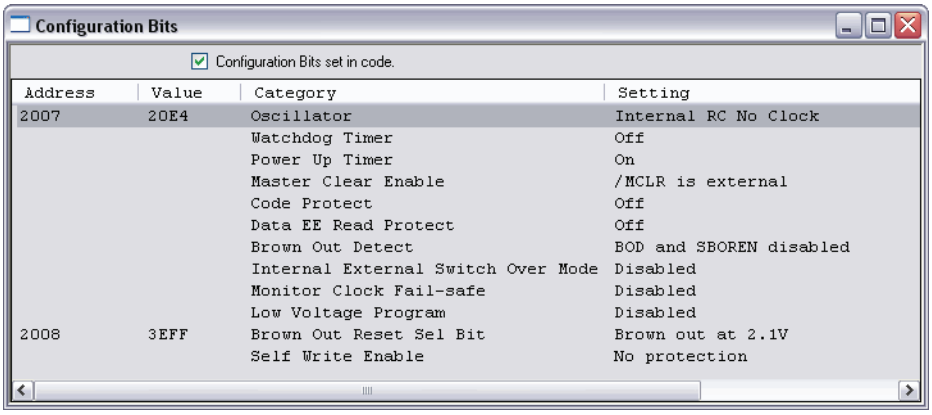
- Oscillator – Internal RC No Clock
- Watchdog Timer – Off
- Power-Up Timer – On
- Master Clear Enable – $\overline{\text{MCLR}}$ is external
- Code-Protect – Off
- Data EE Protect – Off
- Brown-Out Detect – BOD and SBOREN Disabled
- Internal-External Switch Over Mode – Disabled
- Monitor Clock Fail-safe – Disabled
- Low-Voltage Program – Disabled

© 2008 Microchip Technology Inc.

Config 2:

- Self Write Enable – No Protection
- Master Brown-out Reset Sel Bit – Brown-out at 2.1V

FIGURE 4-17: CONFIGURATION BIT SETTINGS



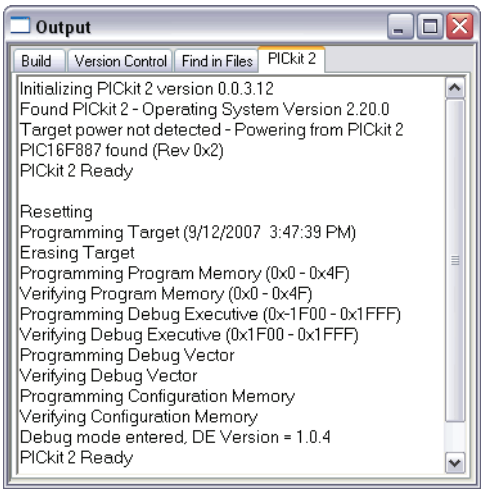
4.5.7 Loading Program Code for Debugging

To program the device, select *Debugger>Program* to program the `16F887Demo.hex` file into the PIC16F887 on the 44-Pin Demo Board.

Programming will only take a few seconds. During programming, the **PICkit 2** tab of the Output dialog window will display the current phase of operation. When programming is complete, the dialog should look similar to Figure 4-18.

Note: Debug executive code is automatically programmed in the upper program memory of the PIC16F887 (target device) when the PICkit 2 is selected as a debugger. Debug code must be programmed into the target device to use the in-circuit debugging capabilities of the PICkit 2.

FIGURE 4-18: OUTPUT WINDOW – PROGRAM DEVICE FOR DEBUG



4.5.8 Running the PIC16F887 Debug Demo

The PICkit 2 Debug Express executes the program code in either real-time (Run) or in steps (Step Into, Step Over, Step Out, Animate.) Real-time execution occurs when you select **Run** in MPLAB IDE. When the program is halted, either by **Halt** or at a breakpoint, you can step through the code.

The following toolbar buttons can be used for quick access to commonly used debug operations:

Debugger Menu	Toolbar Buttons
Run	
Halt	
Animate	
Step Into	
Step Over	
Step Out	
Reset	

To run the demo code:

1. Double click on the `16F887Demo.asm` file from the Project window or select *File>Open* from the toolbar menu. The code will appear in a File window.
2. Select *Debugger>Run* or click **Run**.
3. Turn the potentiometer (RP1), located on the demo board, and observe the LEDs.

If the program were working properly, the LEDs would rotate faster or slower depending on in which direction the potentiometer is turned. However, a bug has been intentionally placed in the code for debugging demonstration purposes. See **Section 4.5.9 "Debugging the PIC16F887 Debug Demo Code"** for debugging instructions.

4. Select *Debugger>Halt* or click **Halt** to stop the program execution. A green solid arrow will mark the line of code in the File window where the program halted.
5. Select *Debugger>Reset>Processor Reset* to reset the program. The arrow will disappear, meaning the device is Reset.

4.5.9 Debugging the PIC16F887 Debug Demo Code

Any of the following issues can prevent the PIC16F887 Debug Demo program from working properly:

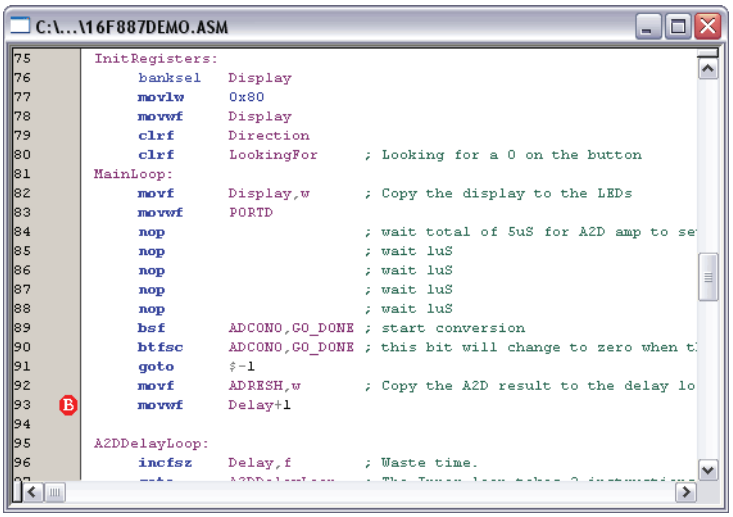
1. The A/D converter value is not being written properly to the Delay routine.
2. The A/D converter is not enabled or has not been set to convert.
3. A typing error in the source code has caused the program to function improperly.

To explore the issue listed first, set a breakpoint at the line of code that writes the value of the A/D result to the high-order Delay byte:

1. Place the cursor on the following line of code in the `16F887Demo.asm` file: `movwf Delay+1`, as shown in Figure 4-19.
   At this breakpoint, the program will stop once the A/D conversion has completed.
2. Either right click on the line to display a drop-down menu and select "Set Breakpoint" or double click on the line. A breakpoint symbol will appear next to the line as the letter B in a solid red octagon, as shown in Figure 4-19.

© 2008 Microchip Technology Inc.

FIGURE 4-19: SET BREAKPOINT

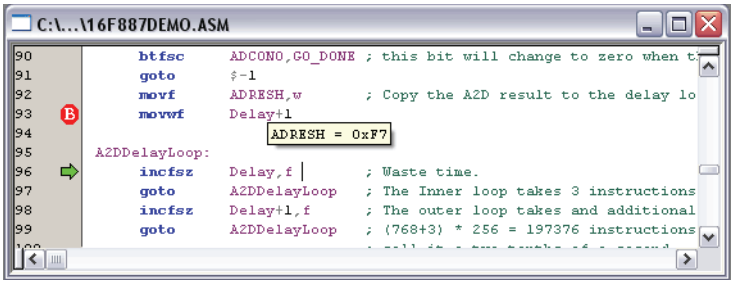


3. Select *Debugger>Run*, or click **Run** to run the program. The breakpoint will stop the program's execution when the program executes the line marked as a breakpoint.

> **Note:** Depending on the device, breakpoint halts may "skid" past the breakpoint location. In this case, the program will halt on the line:
> `incfsz Delay,f`

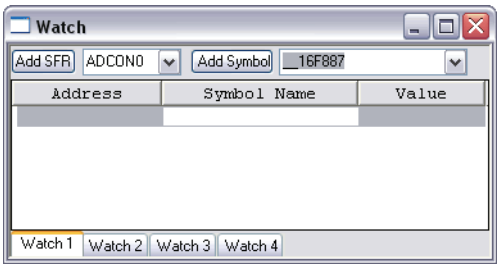
4. Mouse over ADRESH in the line above the breakpoint and it will show the value of the file register (Figure 4-20).

FIGURE 4-20: ADRESH REGISTER VALUE



5. Adjust the potentiometer (RP1) and continue the program execution by selecting *Debug>Run*. The program will run through the loop and halt again.
6. Mouse over ADRESH again to see that the A/D result has not changed. Thus it seems the A/D conversion is not working. The A/D conversion initialization and setup occurs at the beginning of the program.
7. Select *Debugger>Reset* to reset the program.
8. Select *View>Watch* to open a new Watch window. This window will allow you to watch the A/D register value change as the program executes. The Watch dialog opens with the **Watch 1** tab selected, as shown in Figure 4-21.

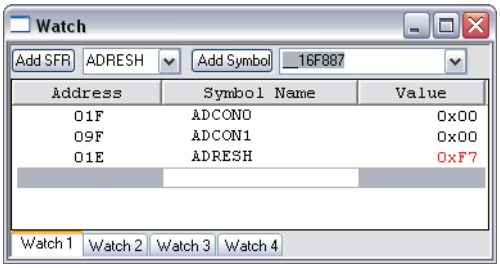
FIGURE 4-21: NEW WATCH WINDOW



Note: It is not recommended to use the *View>File Registers* and *View>Special Function Registers* windows when debugging with PICkit 2 Debug Express. When these windows are open, the entire contents must be written then read back from the target device on every operation. This may slow down debugging operations significantly. Instead, only add the SFR and File Register variables of interest to a Watch window as described in this tutorial.

9. Select ADCON0 from the first drop-down list and then click **Add SFR** to add ADCON0 to the Watch window. Repeat this to add ADCON1 and ADRESH to the Watch window. The selected SFRs should be visible in the Watch window as shown in Figure 4-22.

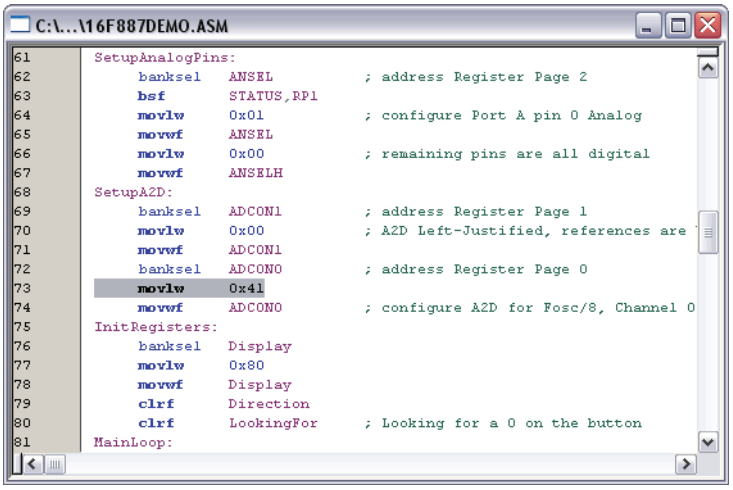
FIGURE 4-22: ADD SFRs TO WATCH



10. Select *Debugger>Run* to the run the program. Again, the program will stop after it reaches the breakpoint.
11. Examine the values of the ADCON0 and ADCON1 registers in the Watch window. The ADCON0 value is '0x40' (`b'01000000'`). This corresponds to the hex value designated in the program. However, this is not correct. A review of the "*PIC16F882/883/884/886/887 Data Sheet*" (DS41291), Analog-to-Digital (A/D) Converter Module section, indicates that the last bit should be a '`1`' (`b'0100000`**`1`**`'`) to turn on the A/D module. To fix this bug, change:
    `movlw 0x40`
    to
    `movlw 0x41`, as shown in Figure 4-23.

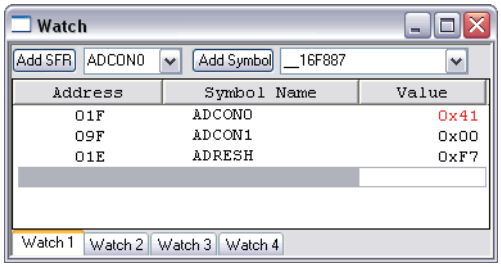
© 2008 Microchip Technology Inc.

FIGURE 4-23: A/D MODULE CODE



12. Select *File>Save* to save your changes.
13. You must rebuild the project and reprogram the device for the changes to take effect. Select *Project>Build All* to rebuild the project and then select *Debugger>Program* to reprogram the device using the PICkit 2.
14. Select *Debugger>Run* to the run the program. Again, the program will stop after it reaches the breakpoint.
15. Examine the values in the Watch window. This time, ADCON0 should be 0x41 (Figure 4-24).

FIGURE 4-24: ADCON CORRECT VALUE



16. Right click on the line of code that marked by the breakpoint and select *Remove>Breakpoint* or double click on the line to remove the breakpoint.
17. Select *Debugger>Run* to run the program real-time. Turn the potentiometer (RP1) to change the value displayed on the LEDs. Now it works!

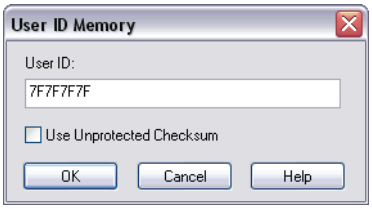
The source code in this tutorial contained only one bug. However, real code may have more. Using MPLAB IDE with PICkit 2 debugging functions, you can successfully find and fix problems in your code.

4.5.10 Programming the Debugged Application

When the program is successfully debugged and running, the next step is to program the device for stand-alone operation in the finished design. When doing this, the resources reserved by the ICD are released for use by the application. To program the application, use the following steps:

1. Disable PICkit 2 as a debug tool by selecting *Debugger>Select Tool>None*.
2. Select PICkit 2 as a programmer in the *Programmer>Select Programmer* menu.
3. Optional: Set up the ID in *Configure>ID Memory* (Figure 4-25).

FIGURE 4-25: USER ID MEMORY



4. Set up the parameters for programming on the *Programmer>Settings*, **Program** tab.
5. Select *Programmer>Program*.

Now the target will run independently.

© 2008 Microchip Technology Inc.

Chapter 5. Troubleshooting

5.1 INTRODUCTION

This chapter describes questions and answers to common problems associated with using the PICkit 2 Development Programmer/Debugger. It also gives information on the PICkit 2 Debug Express MPLAB IDE errors.

- Frequently Asked Questions
- PICkit 2 Debug Express MPLAB IDE Errors

5.2 FREQUENTLY ASKED QUESTIONS

- Device is Not Recognized
- Current Limit Exceeded
- Microsoft Windows Driver
- Verify and Read Return All Zeros
- Microsoft Windows 95/98/NT
- VDD/VPP Errors
- Programming Errors
- Windows Error: Unrecognized USB Device
- PICkit 2 Not Found
- PICkit 2 Programmer Application Locks Up
- Low Pin Count Demo Board Push Button Not Working
- Using the Demo Board Lessons
- Programming Fails on Configuration
- Unable to Program PIC10F Devices
- The PICkit 2 AUX Pin
- PICkit 2 HEX File Format
- Window Display Problems
- Memory Locations are marked 'RR'
- Debug Express is Slow
- Debug Express Overshoots Breakpoint

Device is Not Recognized

Question

Why am I receiving a "*No Device Found*" message?

Answer

Verify that the device is supported and that the target MCU is connected to the PICkit 2 in accordance with **Chapter 3. "Using In-Circuit Serial Programming™ (ICSP™)"**. Verify that PIC18FXXJXX, PIC24X, and dsPIC33F devices have an appropriate capacitance on the VDDCORE/VCAP pin in accordance with the device data sheet.

Verify that the device is a member of the currently active family displayed at the top of the Status window. Select the correct family from the Device Family menu if needed.

See also the Programming Fails on Configuration topic.

Current Limit Exceeded

Question

Why am I receiving the error message "*USB Hub Current Limit Exceeded*" from the Microsoft® Windows® program?

Answer

Verify that the application circuit is not drawing more than 25 mA from the PICkit 2.

Microsoft Windows Driver

Question

After plugging the PICkit 2 unit into the USB port, Windows 98 SE asks for a driver. Where is the driver?

Answer

The PICkit 2 uses the drivers included with Windows 98 SE. When Windows 98 SE prompts for a driver, select "*Search for the best driver for your device*." Then select the checkbox next to "*Microsoft Windows Update*" and click **Next**. Windows OS will automatically install the appropriate driver. **Do not** use Microchip's MPLAB ICD 2 USB device driver.

Verify and Read Return All Zeros

Question

When **Verify** or **Read** are clicked, the Program Memory window comes up with all zeros. What is wrong?

Answer

The device may be code-protected. Ensure code protection has not been selected in the Configuration Word.

Microsoft Windows 95/98/NT

Question

Can I run on Windows 95/98/NT?

Answer

No. These operating systems either do not support USB or have drivers that are not compatible.

VDD/VPP Errors

Question

Why do I keep getting a "VDD Error" or "VPP Error"?

Answer

This error indicates that the PICkit 2 is not able to drive VDD or VPP to the intended voltage. Check the circuit board for shorts, for large current draw and verify that the target device is connected to the PICkit 2 in accordance with **Chapter 3. "Using In-Circuit Serial Programming™ (ICSP™)"**. Make sure that VDD capacitance is not reducing the VDD rise time longer than 500 µs.

© 2008 Microchip Technology Inc.

Programming Errors

Question

Why am I able to program some parts but not others?

Answer

If some parts are configured for Low-Voltage Programming, a floating PGM pin can interfere with programming. Use a resistor to pull this pin low when programming.

Some Mid-Range parts, such as the PIC16F72/73/74/76/77 family and PIC16F737/747/767/777 family require a minimum programming V_{DD} of +4.75V. Depending on the USB voltage, the PICkit 2 may not be able to supply +4.75V on V_{DD}. Program these parts using an external +5.0V power supply.

Some PIC18F parts require significant bypass capacitance on V_{DD}. Try increasing the total bypass capacitance up to 10 uF.

PIC18FXXJXX, PIC24X, and dsPIC30F/33F devices require a 4.7 uF capacitor on the V_{DDCORE}/V_{CAP} pin in order to function properly. If not using a separate regulator to supply V_{DDCORE}, ensure that the ENVREG pin is tied to V_{DD}.

Windows Error: Unrecognized USB Device

Question

Why do I get an “unrecognized device” error when plugging my PICkit 2 into USB?

Answer

This error may occur if PICkit 2 is plugged into USB while connected to a target circuit board. When plugging PICkit 2 in a PC, restarting or booting up a PC, ensure the PICkit 2 is not connected to a target device.

This error may also occur when PICkit 2 is used with some USB hubs. If the PICkit 2 is plugged into a USB hub port, try plugging the PICkit 2 directly into a PC USB port.

PICkit 2 Not Found

Question

I have my PICkit 2 plugged into USB, but the PICkit 2 Programmer application keeps saying “PICkit 2 Not Found?

Answer

Please see the answer for Windows Error: Unrecognized USB Device.

PICkit 2 Programmer Application Locks Up

Question

Why is the PICkit 2 Programmer window locking up?

Answer

Often times, the software hasn’t truly locked up. During a programming operation the PICkit 2 application user interface is inactive. Until the programming operation is complete, the Windows OS may stop updating the PICkit 2 if another application is brought into focus or the PICkit 2 has just been brought back into focus. Once the programming operation is complete, the application window will update. For large memory parts, programming may take several minutes. For 8-bit devices, try waiting at least 2 minutes before concluding the software has truly locked up. For 16-bit devices, try waiting 5 minutes for the operation to complete.

There are a few USB controller chipsets that appear to cause lockup problems with the PICkit 2, which seem to be more common in laptops. This can usually be worked around by connecting a USB hub between the PC and the PICkit 2 unit, or using a USB Cardbus adapter. Using a USB hub with an external power supply is recommended.

Low Pin Count Demo Board Push Button Not Working

Question

When I run the Low Pin Count Demo Board Lessons, why doesn't the push button work?

Answer

When programming the lessons from within MPLAB IDE, the IDE always maintains an output voltage on the PICkit 2 $\overline{\text{MCLR}}$-V_{PP} pin. On the Low Pin Count and 28-pin Demo Boards, the push button signal is connected to the $\overline{\text{MCLR}}$ signal. The PICkit 2 output interferes with the push button signal.

Use the PICkit 2 Programmer software application to program and run the Low Pin Count Demo Board Lessons.

Additionally, future versions of the MPLAB IDE will optionally allow the $\overline{\text{MCLR}}$-V_{PP} pin to be tri-stated when the device is "Released from Reset". Check *Programmer>Settings* for this option. Note that this option would not apply during debugging.

Using the Demo Board Lessons

Question

Where are the included lessons? Is there any documentation on them?

Answer

The Starter Kit and Debug Express kit lessons may be installed from the included PICkit 2 CD-ROM. The lessons will be installed to the local hard drive under `C:\Pk2 Lessons\`.

The documentation for the lesson projects is in the user's guide for the included demo Board. For the Starter Kit, see the *"Low Pin Count Demo Board User's Guide"* (DS51556). For the Debug Express Kit, see the "*44-Pin Demo Board User's Guide"* (DS41296). Both of these documents may be opened from the PICkit 2 Programmer Help menu.

Programming Fails on Configuration

Question

Why does programming a device always fail on the Configuration Word(s), after which PICkit 2 won't recognize the device?

Answer

This may be caused by a Configuration setting or program code that affects the ICSP PGD and PGC pins. This interference can prevent the target PIC MCU from entering programming mode.

Check the menu option *Tools>Use VPP First Program Entry* when attempting a programming operation on these devices. This program mode entry will usually get around the problem, but it requires that the target device be powered from the PICkit 2 unit V_{DD} pin.

© 2008 Microchip Technology Inc.

Unable to Program PIC10F Devices

Question

Why won't my PIC10F parts program in the Low Pin Count Demo Board?

Answer

The Low Pin Count Demo Board supports 8-pin devices. The PIC10F parts are 6-pin devices, even though they can be ordered in an 8-pin DIP. The PIC10F pinout is not compatible with the Low Pin Count Demo Board. Instead, the AC163020 PIC10F2XX Programming Adapter should be used.

The PICkit 2 AUX Pin

Question

What is the AUX pin for? How do I connect it?

Answer

The PICkit 2 AUX pin is not used for programming PIC microcontrollers, and should be left unconnected. The AUX pin is only used when programming some Serial EEPROMS. See the PICkit 2 Readme file (*Help>Readme*) for EEPROM connection information.

PICkit 2 HEX File Format

Question

What HEX file format does the PICkit 2 Programmer application use?

Answer

The PICkit 2 Programmer application uses the Intel Hex 32 Format, often referred to as INHX32. However, PICkit 2 does not support record types 03 and 05. Least Significant Bytes are at lower hex file addresses (little Endian format).

Window Display Problems

Question

Why is the PICkit 2 Programmer application displaying Program Memory and/or EEPROM Data locations as all periods "…"? Why can't I close the *Help>About* dialog?

Answer

These issues are frequently caused by non-standard monitor DPI settings. This can be corrected by setting the DPI to the Normal setting, 96 DPI. To do this

1. Right click on the Windows desktop background and select Properties from the pop-up menu. This will open the Display Properties dialog.
2. In the Display Properties dialog, select the **Settings** tab.
3. On this tab, click **Advanced** to open the Monitor dialog.
4. In the Monitor dialog, select the **General** tab.
5. On this tab, change the "DPI setting:" to "Normal size (96 DPI)
6. Click **Apply**, then the **OK** button to close the dialog.
7. Click **OK** to close the Display Properties dialog.

It may be necessary to restart the PC for the changes to take effect.

Memory Locations are marked 'RR'

Question

Why does the File Register window or Program Memory show some locations marked with 'RR'?

Answer

These are locations that are reserved by ICD debuggers including PICkit 2 Debug Express during debugging for the debug executive. See **Section 4.4.2 "Reserved Resources"**.

Debug Express is Slow

Question

Why is debugging with PICkit 2 so slow?

Answer

Opening the *View>File Register*s or *View>Special Function Registers* windows in the MPLAB IDE can significantly slow down debugging, due to the volume of information being exchanged though the debugger. Use a Watch window instead as discussed in **Section 4.5.9 "Debugging the PIC16F887 Debug Demo Code"**.

The debug interface is also directly affected by the target device oscillator speed. When debugging with an oscillator less than 1 MHz, a slowdown in debugging performance may be noticed.

Debug Express Overshoots Breakpoint

Question

Why isn't PICkit 2 Debug Express stopping on the breakpoint instruction?

Answer

This behavior is referred to as "breakpoint skidding" and is covered in **Section 4.4.6 "Breakpoint Skidding"**.

5.3 PICKIT 2 DEBUG EXPRESS MPLAB IDE ERRORS

PK2Error0002: Allocation failure (Component)
PK2Error0003: Missing component (Component)
PK2Error0005: Failed to acquire component (Component, HRESULT)
PK2Error0006: Failed external call (Component, Method, HRESULT)

Description: An error occurred initializing a PICkit 2 component, during an internal operation, or when communicating with the MPLAB IDE environment. The most probable cause is an incomplete or corrupted installation.

Suggested Actions:

1. Uninstall all versions of MPLAB IDE from the PC.
2. Reinstall the desired MPLAB IDE version.
3. If the problem persists contact Microchip.

PK2Error0008: Read failure (GetLastError**)**
PK2Error0009: Write failure (GetLastError**)**

Description: MPLAB IDE was unable to communicate with the PICkit 2. Possible causes include the PICkit 2 not enumerating properly on USB, a target board causing a USB overcurrent condition or problems with a USB hub.

© 2008 Microchip Technology Inc.

Suggested Actions: Ensure the target board does not pull more than 25 mA of current from the PICkit 2. If it does, use an external power supply with the target board.

A target board may interfere with PICkit 2 enumerating on USB. When connecting PICkit 2 to a USB port, booting or restarting a connected PC, ensure the PICkit 2 is disconnected from the target board.

If the PICkit 2 is connected to a USB hub, try connecting the PICkit 2 directly to a PC USB port.

When using PICkit 2 as a programmer or debugger in the MPLAB IDE, do not run the PICkit 2 Programmer application or PK2CMD at the same time.

If the problem persists, try connecting to PICkit 2 using the PICkit 2 Programmer application.

PK2Error0010: Failed to unload PICkit 2

Description: MPLAB IDE was unable to disconnect from PICkit 2 as a debugger or programmer when switching to the other tool mode.

Suggested Actions: Disconnect the PICkit 2 from USB, reboot the computer, then reconnect PICkit 2 to USB (ensure the PICkit 2 is not connected to a target board.) If the problem persists contact Microchip.

PK2Error0011: PICkit 2 is already busy. Unable to perform requested task.

Description: PICkit 2 Debug Express has been asked to perform an operation while it is busy servicing a prior request. For example, a programming operation may still be process when a debug Step operation is attempted.

Suggested Actions: This situation will not normally happen as the user interface is disabled while PICkit 2 is servicing requests. Wait for the original operation to complete.

PK2Error0012: Unable to open file (Filename)

Description: PICkit 2 Debug Express was unable to open the specified file.

Suggested Actions: Ensure the file exists, and file or directory permissions do not prevent access to the file. This may also be caused by an incomplete or corrupted installation. See PK2Error0002 Suggested Actions.

PK2Error0013: Bad hex line (File - Line)
PK2Error0014: Invalid address in hex file (File - Line - Address)

Description: PICkit 2 Debug Express detected an error in a debug executive hex file.

Suggested Actions: Since all debug executive hex files are provided by Microchip, the issue is most likely a corrupted installation. See PK2Error0002 Suggested Actions.

PK2Error0015: Failed Windows call (WinFunc - GetLastError)

Description: A call to a Windows function within PICkit 2 Debug Express has failed. This internal error is primarily meant for debugging purposes.

Suggested Actions: This may be caused by existed system resources or a corrupted installation.

1. Try rebooting your system to free up memory.
2. Make sure you have a reasonable amount of free space on your hard drive, and that it is not overly fragmented.
3. See PK2Error0002 Suggested Actions for a corrupted installation.

© 2008 Microchip Technology Inc.

PK2Error0016: Failed to find firmware files (MPLABDir)

Description: PICkit 2 Debug Express was unable to find the PICkit 2 Operating System firmware files.

Suggested Actions: This is likely caused by an incomplete or corrupted installation. See PK2Error0002 Suggested Actions.

PK2Error0017: Invalid hex file (Filename - Start - End)

Description: PICkit 2 Debug Express detected an error in a PICkit 2 Operating System hex file.

Suggested Actions: Since all Operating System hex files are provided by Microchip, the issue is most likely a corrupted installation. See PK2Error0002 Suggested Actions.

PK2Error0018: Unable to enter bootloader

Description: PICkit 2 Debug Express was unable to access the PICkit 2 unit bootloader while updating the PICkit 2 Operating System.

Suggested Actions: Disconnect the PICkit 2 unit from USB and the target board. While holding the PICkit 2 unit push button down, plug the PICkit 2 unit back into the USB port. The BUSY LED should flash slowly. If it does not, the bootloader may be corrupted. Contact Microchip about the problem.

PK2Error0019: Failed PICkit OS read ([1] - [2] - [3] - [4])

Description: An error occurred while communicating with the PICkit 2 bootloader.

Suggested Actions: Remove the PICkit 2 unit from USB and the target board, reboot the PC, reconnect the PICkit 2 unit to USB and retry the operation. If this does not work, try updating the firmware using the PICkit 2 Programmer application.

PK2Error0020: Failed PICkit OS verify (Address - Read - Expected)

Description: The download of a PICkit 2 Operating System to the PICkit 2 unit failed.

Suggested Actions: Remove the PICkit 2 unit from USB and the target board, reboot the PC, reconnect the PICkit 2 unit to USB and retry the operation. If this does not work, try updating the firmware using the PICkit 2 Programmer application.

PK2Error0021: Unable to exit bootloader

Description: PICkit 2 Debug Express failed to exit the PICkit 2 bootloader mode after downloading a PICkit 2 Operating System to the unit.

Suggested Actions: Remove the PICkit 2 unit from USB and the target board, reboot the PC, reconnect the PICkit 2 unit to USB and retry the operation. If this does not work, try updating the firmware using the PICkit 2 Programmer application.

PK2Error0022: PICkit 2 not found

Description: MPLAB IDE is not able to detect the PICkit 2 device on any USB port.

Suggested Actions: See PK2Error0008 Suggested Actions.

PK2Error0023: Target V_{DD} measured at X.XV which is outside the operational range of this device (minV - maxV)

Description: The target power supply is outside the voltage range supported by PICkit 2 for the current device. This may be due to device Bulk Erase limitations or PICkit 2 hardware limitations.

© 2008 Microchip Technology Inc.

Suggested Actions: Use a target VDD voltage within the given range. If the target VDD is close to one limit but within the given limits and this error appears, try calibration the PICkit 2 unit with the PICkit 2 Programmer software so the target voltage is read more accurately.

PK2Error0024: PICkit 2 was unable to establish a valid VDD on the target (Attempted V - Read V)

Description: PICkit 2 was unable to provide the intended VDD voltage to the target. This is usually caused by a heavy VDD load.

Suggested Actions: Power the target board externally, and/or try calibrating the PICkit 2 unit with the PICkit 2 Programmer software.

PK2Error0025: Failed to load PICkit 2

Description: PICkit 2 Debug Express encountered a fatal error while attempting to load in the MPLAB IDE.

Suggested Actions: This is likely caused by an incomplete or corrupted installation. See PK2Error0003 Suggested Actions.

PK2Error0027: Failed verify (Address - Expected Value - Value Read)

Description: A verify operation failed. The error message specifies the memory type and address involved, and gives the expected value and the value actually read. A verify operation on a previously programmed and code protected part will fail, as the code protection feature returns all zeroes when the memory is read.

Note:	PICkit 2 Debug Express will terminate the verify operation upon the first incorrect value read from the target. Only this first incorrect value will be reported. There may be others.

Suggested Actions: Ensure the device is not code protected. Check the ICSP connections as described in **Chapter 3. "Using In-Circuit Serial Programming™ (ICSP™)"**. If the target device has AVDD and AVSS pins, check that these are properly connected. Verify that PIC18FXXJXX, PIC24X, and dsPIC33F devices have an appropriate capacitance on the VDDCORE/VCAP pin in accordance with the device data sheet. Some PIC18F devices may require up to 10 uF of capacitance on the VDD pin to program properly.

PK2Error0028: Unable to enter debug mode

Description: PICkit 2 Debug Express is unable to connect to the debug executive in the target device.

Suggested Actions: Common reasons for this error:

1. If the target is a Baseline or Mid-Range device (PIC10F, PIC12F, PIC16F) it may require an ICD header to debug with. See **Section 4.4.3 "Using an ICD Header"**.
2. The device oscillator is not working. Check the Configuration bits and ensure the device has a working oscillator.
3. Make sure to select *Debugger>Program* in MPLAB IDE before trying to run or step your code.
4. If you have a PICkit 2 with a black colored push button, you may need pull-down resistors to debug reliably. See **Section 4.3 "Installing the Hardware and Software"**.
5. The target board is not powered or is not powered properly. Check the power supply.
6. The PICkit 2 VDD output pin has not been connected to the target board. Even if

the target board has its own power supply, this pin must always be connected to the target VDD supply.

7. The PICkit 2 unit has become disconnected from the target board or has an intermittent connection on one or more pins. Check the PICkit 2 connections to the target board.
8. The incorrect Linker Script was used. The "-i" ICD linker script must be used when debugging. (For example, 16F887i.lkr must be used instead of 16F887.lkr).
9. The target application is attempting to use resources reserved for the debug executive, causing the debug executive to become corrupted. For example, the target application is writing to a reserved File Register. See **Section 4.4.2 "Reserved Resources"**.
10. For PIC24 and dsPIC® DSC devices, the incorrect PGXn/EMUXn port is selected in the "Comm Channel Select" Configuration bits. Check that the port the PICkit 2 Debug Express is connected to is selected.
11. A configuration setting may be preventing the target from operating properly and executing code. The target will not enter Debug mode if it is unable to execute code.
12. The device is code-protected. Check Configuration bits for code protect settings.
13. An operation may be attempting in an invalid context. For example, PICkit 2 Debug Express cannot set a breakpoint if the target application is currently running.

PK2Error0029: Failed PICkit 2 operation

Description: PICkit 2 Debug Express failed to complete a debug operation, such as a "Halt" or "Step" operation.

Suggested Actions: See PK2Error0028 Suggested Actions.

PK2Error0030: Failed to read target file registers

Description: PICkit 2 Debug Express was unable to complete a read of one or more target File Registers.

Suggested Actions: See PK2Error0028 Suggested Actions.

PK2Error0031: I/O Operation timeout

Description: A PICkit 2 USB operation timed out.

Suggested Actions: Try the operation again. If it continues to fail, remove the PICkit 2 unit from USB and the target board, reboot the PC, reconnect the PICkit 2 unit to USB and retry the operation. If this does not work, try updating the firmware using the PICkit 2 Programmer application.

PK2Error0032: Failed Blank Check (Address - Expected Value - Value Read)

Description: A Blank Check operation failed. The error message specifies the memory type and address involved, and gives the expected blank value and the value actually read.

> **Note:** PICkit 2 Debug Express will terminate the Blank Check operation upon the first incorrect value read from the target. Only this first incorrect value will be reported. There may be others.

Suggested Actions: Try erasing the device. If the device is not blank after an erase check the ICSP connections as described in **Chapter 3. "Using In-Circuit Serial Programming™ (ICSP™)"**. If the target device has AVDD and AVSS pins, check that these are properly connected. Verify that PIC18FXXJXX, PIC24X, and dsPIC33F devices

© 2008 Microchip Technology Inc.

have an appropriate capacitance on the VDDCORE/VCAP pin in accordance with the device data sheet. Some PIC18F devices may require up to 10 uF of capacitance on the VDD pin to erase properly.

PK2Error0033: User aborted operation

Description: This error is usually given if PICkit 2 Debug Express has popped up a warning and the user opted to cancel the current operation. For example, PICkit 2 Debug Express may warn that a debug target cannot be programmed with the MCLR pin disabled. If the user clicks **Cancel** in the warning dialog, the programmed operation will be aborted with this error.

Suggested Actions: None.

PK2Error0034: Failed to Initialize

Description: A PICkit 2 Debug Express component was unable to complete its initialization sequence.

Suggested Actions: This is likely caused by an incomplete or corrupted installation. See PK2Error0002 Suggested Actions.

PK2Error0035: Failed to retrieve XML Data (node)
PK2Error0036: Failed to open PK2 Script File (filename)
PK2Error0037: PK2 Script File structure size mismatch (struct internal size file size)
PK2Error0038: Unable to fit block in IO buffer (MemType Direction Blocksize)

Description: PICkit 2 Debug Express internal error.

Suggested Actions: This is likely caused by an incomplete or corrupted installation. See PK2Error0002 Suggested Actions.

PK2Error0039: Not halted in debug executive

Description: An operation was attempted that can only be completed when the target is halted while the target is running application code. This may occur if PICkit 2 Debug Express is unable to halt the target.

Suggested Actions: Halt the target before attempting the operation again.

PK2Error0040: Operation not supported for current device

Description: The attempted debug operation is not supported for the current target.

Suggested Actions: None.

PK2Error0041: Failed to create Stream
PK2Error0042: Failed Stream Read
PK2Error0043: Failed IStream Write
PK2Error0044: Failed IStream Open
PK2Error0045: No STATSTG
PK2Error0046: Failed IStream Read
PK2Error0047: Failed Stream Write

Description: PICkit 2 Debug Express internal error.

Suggested Actions: This is likely caused by an incomplete or corrupted installation. See PK2Error0002 Suggested Actions.

NOTES:

© 2008 Microchip Technology Inc.

Chapter 6. Updating the PICkit 2 Operating System

6.1 INTRODUCTION

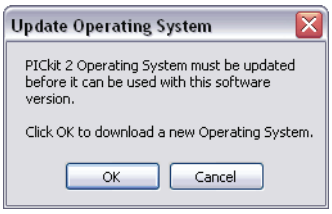
How to update the operating system (firmware) for the PICkit 2 Development Programmer/Debugger is described here.

- Updating the PICkit 2 OS - PICkit 2 Programmer Application
- Updating the PICkit 2 OS – MPLAB IDE

6.2 UPDATING THE PICkit 2 OS - PICkit 2 PROGRAMMER APPLICATION

When the PICkit 2 Programmer application is launched, it will check the firmware version of the PICkit 2 to see if it is the latest version. If not, it will attempt to automatically download the latest version (Figure 6-1).

FIGURE 6-1: UPDATE PICkit™ 2 OS WARNING



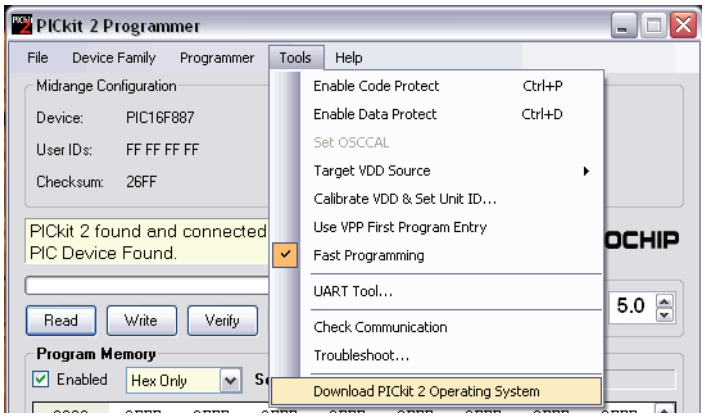
If there is an error in the automatic download, if you need a different operating system (OS), or if you have been instructed to do so, you may manually update the OS.

To update the PICkit 2 firmware/OS manually, complete the following steps:

1. Download the latest PICkit 2 firmware/OS from the Microchip web site at www.microchip.com and place the file in the PICkit 2 installation directory. By default, the location is:
   `C:\Program Files\Microchip\PICkit 2 v2`

2. Select *Tools>Download PICKit 2 Operating System* as shown in Figure 6-2.

FIGURE 6-2: DOWNLOAD PICkit™ 2 OS



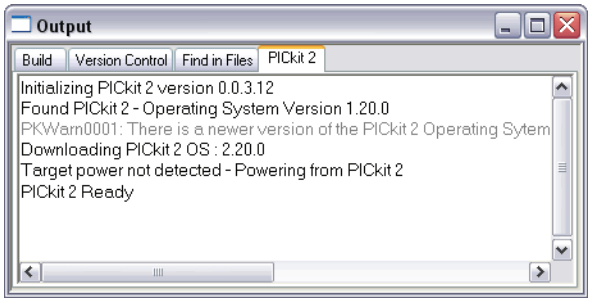
3. Browse to the directory where the latest firmware/OS file was saved.
4. Select the `PK2*.hex` file and click on the **Open** button.

   The progress of the OS update will be displayed in the status bar of the programming software and the Busy LED on the PICkit 2 will flash. When the update completes successfully, the status bar will display "Operating System Verified" and the Busy LED will go out. The firmware/OS update is then complete.

6.3 UPDATING THE PICkit 2 OS – MPLAB IDE

When PICkit 2 is chosen as either the debug tool or programmer in MPLAB IDE, MPLAB IDE will check the firmware version of the PICkit 2 to see if it is the latest version. If not, it will attempt to automatically download the latest version. Its progress is recorded in the Output window on the PICkit 2 tab (Figure 6-3).

FIGURE 6-3: UPDATE PICkit™ 2 OS PROGRESS



If there is an error in the automatic download, if you need a different operating system (OS), or if you have been instructed to do so, you may manually update the OS.

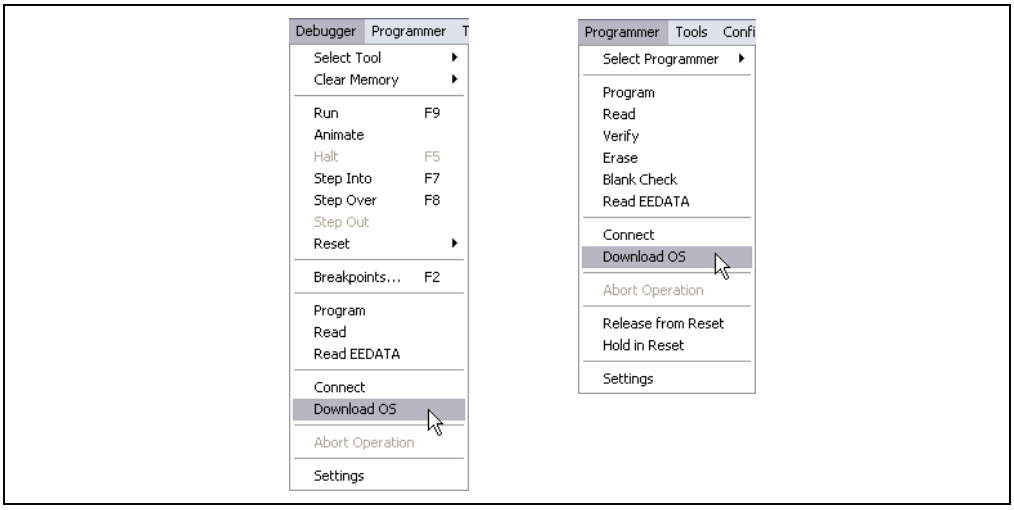
To update the PICkit 2 firmware/OS manually, complete the following steps:

1. Download the latest PICkit 2 firmware/OS from the Microchip web site at www.microchip.com and place the file in the PICkit 2 installation directory. By default, the location is:
   `C:\Program Files\Microchip\MPLAB IDE\PICkit 2`
2. Select PICkit 2 as the debug tool (*Debugger>Select Tool>PICkit 2*) or programmer (*Programmer>Select Programmer>PICkit 2*).

© 2008 Microchip Technology Inc.

3. Select *Debugger>Download OS* or *Programmer>Download OS* as in Figure 6-4.

FIGURE 6-4: DOWNLOAD PICkit™ 2 OS FROM MPLAB® IDE



4. MPLAB IDE will proceed to download the latest firmware/OS file to the PICkit 2. The progress of the OS update will be displayed in the Output window, PICkit 2 tab, and the Busy LED on the PICkit 2 will flash. When the update completes successfully, the text in the **PICkit 2** tab will state that “PICkit 2 Ready” and the Busy LED will go out. The firmware/OS update is then complete.

NOTES:

© 2008 Microchip Technology Inc.

Chapter 7. PICkit 2 UART Tool

7.1 INTRODUCTION

The PICkit 2 Programmer application versions 2.40 and later include the UART Tool feature. This feature allows PICkit 2 to be used as a serial UART terminal interface for communicating with a PIC microcontroller. Potential uses include:

- Displaying debug text output from the microcontroller
- Logging microcontroller data to a text file
- Developing and debugging a microcontroller UART interface
- Interfacing with and sending commands to the microcontroller during development

The tool supports full duplex asynchronous serial communications from 150 to 38400 baud, including custom non-standard baud rates. The PICkit 2 connects via the ICSP connector directly to microcontroller TX and RX pins at logic levels. No transceiver is needed.

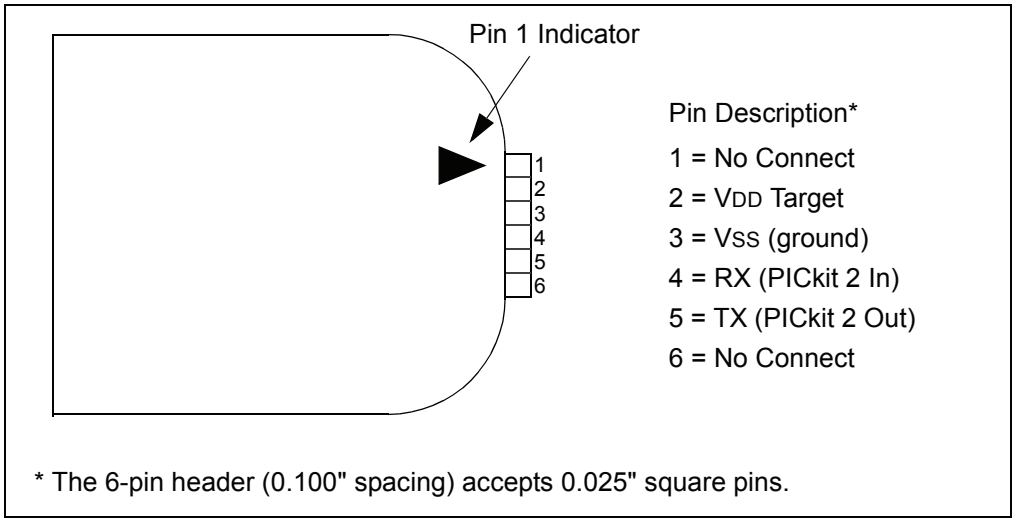
7.2 CONNECTING THE PICkit 2 UART TOOL

The PICkit 2 connects to the target serial port target as shown in Figure 7-1. The PICkit 2 TX signal (out) should connect to the target RX signal (in), and the PICkit 2 RX signal (in) should connect to the target TX signal (out).

When being used with the UART Tool, the PICkit 2 may not be able to supply VDD voltage to the target depending on the application version. See the version Release Notes in the PICkit 2 Readme file. However, even when PICkit 2 is not supplying the target VDD, the PICkit 2 VDD pin must be connected to the target VDD voltage or it will not be able to communicate.

The PICkit 2 UART Tool will work with target VDD voltages from 2.5 Volts to 5.0 Volts.

FIGURE 7-1: PICkit™ 2 UART TOOL



The format of the serial signals on pins 4 and 5 is inverted such that a Start Bit = logic low (VSS), and a Stop Bit = logic high (VDD). The serial data format is always 8 data bits, no parity, with one stop bit.

The BUSY LED on the PICkit 2 unit acts as an activity light for the serial connection.

Note: Even though PICkit 2 may not supply the target VDD, the PICkit 2 VDD pin must be connected to the target VDD voltage or it will not be able to communicate.

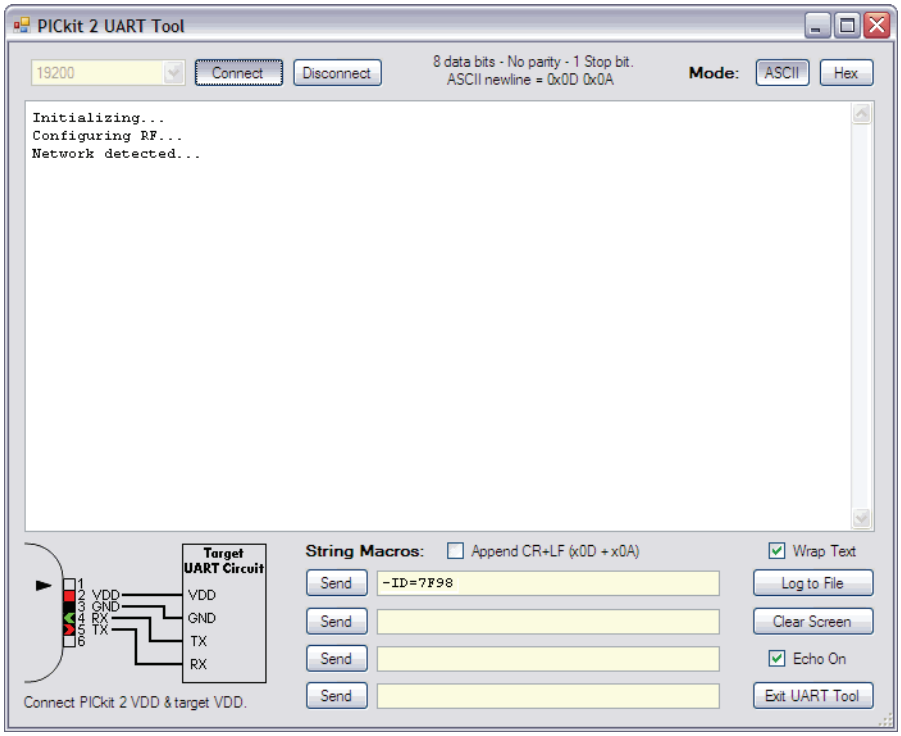
CAUTION

The PICKIT 2 UART Tool is intended for use at logic level signals only. DO NOT connect the PICkit 2 UART Tool to RS-232 +/- 12 Volt signals! Doing so may cause damage to the PICkit 2 unit, your computer USB ports and the target serial port.

7.3 THE PICKIT 2 UART TOOL WINDOW

To open the PICkit 2 UART Tool window, select *Tools>UART Tool…* (Figure 7-2). The PICkit 2 Programmer window will open and the PICkit 2 Programmer application window will close, as the programmer cannot be used while the UART Tool is active, and vice versa.

FIGURE 7-2: PICkit™ 2 UART TOOL WINDOW ASCII MODE



The UART Tool has two modes: ASCII and HEX. The current mode is selected and displayed by the buttons on the upper right hand of the display. The button corresponding to the active mode will be displayed depressed. For example, in Figure 7-2 the ASCII mode is active and so the ASCII button appears pressed.

The mode selection affects how serial data is displayed and entered in the window, and the modes are explained in the next two sections.

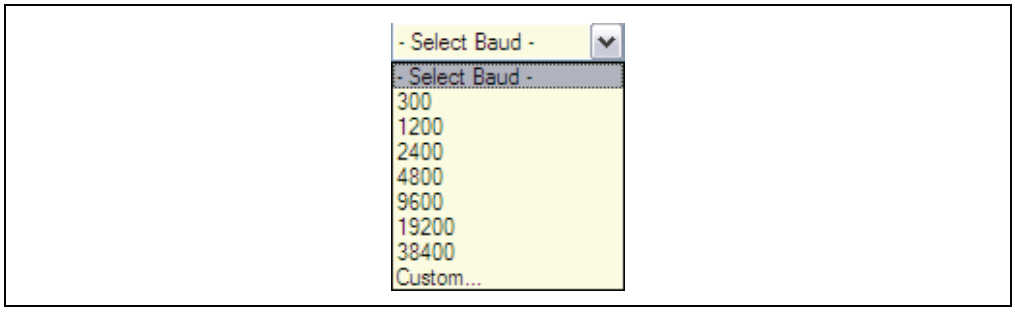
© 2008 Microchip Technology Inc.

The window may be resized by clicking and dragging on the lower right corner of the window.

7.3.1 Setting the Baud Rate and Connecting

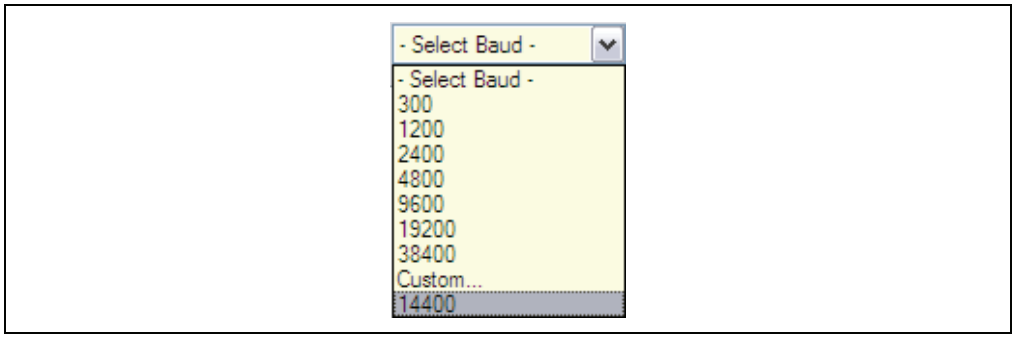
The baud rate may be selected from the combo box in the upper left corner of the UART Tool window. Several common baud rates are listed, and by selecting “Custom…” any baud rate from 150 to 38400 in 1 baud increments may be used (Figure 7-3).

FIGURE 7-3: BAUD RATE SELECTION



Once a custom baud rate has been set, it will be available as the last item in the baud rate selection box (Figure 7-4). Only one custom baud rate may be set at a time. Select “Custom…” again to change the custom baud rate.

FIGURE 7-4: CUSTOM 14400 BAUD RATE



After selecting the baud rate, click **Connect** to enable the UART Tool serial interface at the selected baud. Once the serial interface has been enabled, it may be disabled by clicking **Disconnect**. The baud rate may only be changed while the interface is disconnected.

7.3.2 ASCII Mode

When ASCII mode is selected, serial bytes received from the target’s UART are displayed as ASCII characters in the main window terminal display. All bytes are displayed consecutively. To display a new line, the target UART must transmit the character values 0x0D (carriage return) and 0x0A (line feed) in sequence. If one of these values is sent, or they are sent in reverse order, the character(s) will appear as an unprintable character (box) in the display.

Bytes may be transmitted in three ways:

1. Click on the terminal display to select it. Any characters typed on the PC keyboard will be immediately transmitted out of PICkit 2.
2. Right click on the terminal display and select Paste from the pop-up menu to paste any previous copied or cut text. <Ctrl> + <v> may also be used. Any pasted data will immediately be transmitted out of PICkit 2.
3. Use the “String Macros” at the bottom of the window.

No transmitted characters will be displayed if the "Echo On" checkbox is unchecked. Check this box to display transmitted characters along with received characters in the terminal display.

The "String Macros" at the bottom of the display allow up to four strings of characters to be entered. Each string can be up to 60 characters long. When **Send** next to a string is clicked, the entire string will be transmitted in sequence out of the PICkit 2. The "String Macros" strings are saved even when the UART Tool is not in use.

When checked, the "Append CR+LF" checkbox above the string macro boxes will automatically transmit the carriage return (0x0D) and line feed (0x0A) characters at the end of a string when **Send** is clicked. When unchecked, nothing is transmitted after the string characters.

Text in the terminal display may be selected (highlighted) with the cursor and copied into the clipboard by right clicking and selecting Copy or pressing <Ctrl> + <c>.

7.3.3 Hex Mode

The UART Tool Hex mode displays the hex values of bytes received from the target's UART in the terminal display.

A line of bytes received by the UART Tool is preceded with the text "RX:".

A line of bytes transmitted by the UART Tool is preceded with the text "TX:".

In Figure 7-5, the four hex bytes 0x50, 0x4B, 0x53, and 0x41 were received from the target (0x50 was received first). The three bytes 0x70, 0x6B, and 0x32 were then transmitted, 0x70 first.

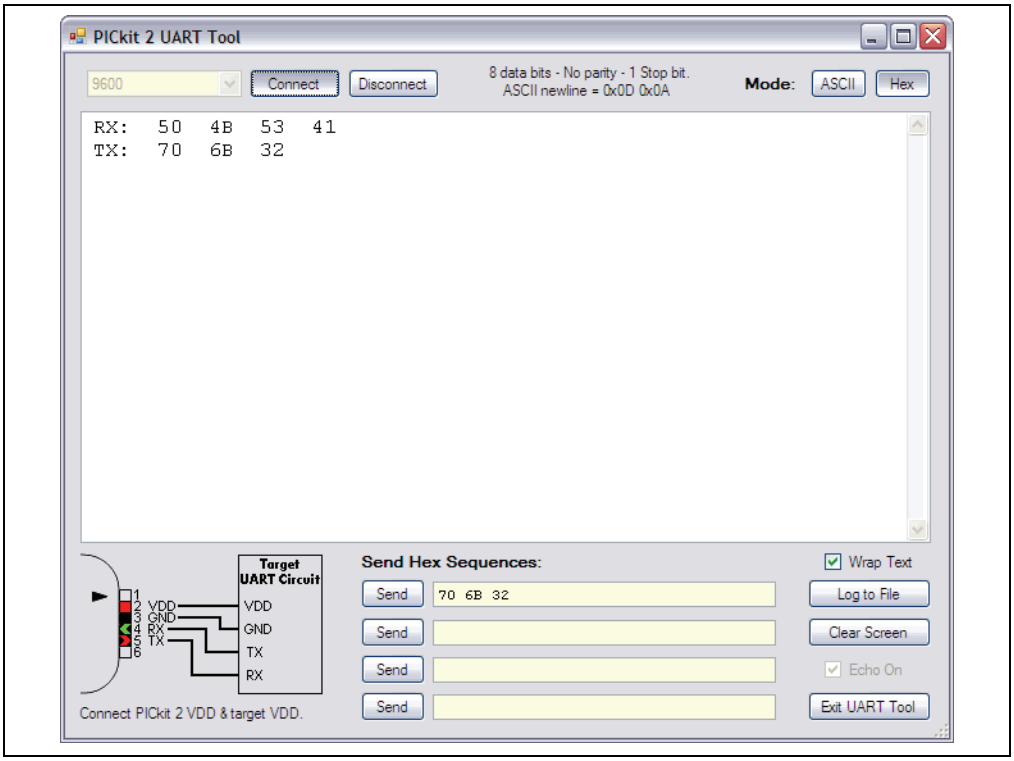
In Hex mode, bytes may only be transmitted one way: by typing a sequence of one or more hex values in one of the four "Send Hex Sequences:" boxes. To send a sequence, click **Send** next to it. A hex sequence may contain from 1 to 48 bytes.

The "Echo On" checkbox has no function in Hex mode.

Data in the terminal display may be selected (highlighted) with the cursor and copied into the clipboard by right clicking and selecting Copy or pressing <Ctrl> + <c>.

© 2008 Microchip Technology Inc.

FIGURE 7-5: PICKIT™ 2 UART TOOL WINDOW HEX MODE



7.3.4 Wrap Text

This checkbox affects both modes, ASCII and HEX. When checked, a displayed line will automatically wrap at the right edge of the terminal display. If unchecked, then a received line will not wrap and a horizontal scroll bar will appear in the terminal display to allow the entire line to be viewed.

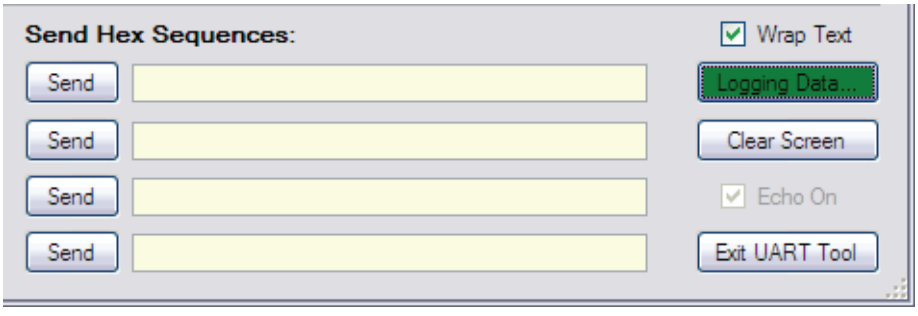
The terminal display will keep the last 200 lines of received data in the buffer. The vertical scroll bar may be used to view previous data or text.

7.3.5 Log to File

The **Log to File** button allows all text and data to be saved to a text file as it appears on the display.

Click **Log to File** to bring up a Save File dialog. Specify the file location and name and click **Save**. While the terminal display is being saved to the log file, the **Log to File** button will turn green and display the text “Logging Data…”. (Figure 7-6)

FIGURE 7-6: LOG TO FILE IS ACTIVE



To stop logging data from the display and close the log file, click **Logging Data…**.

Note:	Only data appearing on the display after the log file is opened will be written to the log file. Existing data is not saved. To save existing data, use Copy and Paste.

7.3.6 Clear Screen

Clear the **Clear Screen** button to clear all text or data from the terminal display window. If data is being logged to a file, clearing the screen will not affect the log file.

7.3.7 Exit UART Tool

Click **Exit** to close the UART Tool window and return to the PICkit 2 Programmer application main window. The UART Tool terminal display data and settings will be preserved during any programming operations until the UART Tool is re-entered.

If the PICkit 2 application is closed, the UART Tool settings, including String Macros and Hex Sequences, will be saved and restored for the next time the UART Tool is used. The terminal display buffer text and data is not saved when the application is closed.

© 2008 Microchip Technology Inc.

Appendix A. MPLAB IDE Reference

A.1 INTRODUCTION

In MPLAB IDE, you first select a PICkit 2 supported device and then "PICkit 2" from either the Debugger or Programmer menu to add debugging or programming functions to MPLAB IDE, respectively. The functions made available are summarized here. For more information about using the PICkit 2 Development Programmer/Debugger with MPLAB IDE, see **Chapter 4. "PICkit 2 Debug Express"**.

- Debugging Functions
- Programming Functions
- Settings Dialog

In addition, "PICkit 2" will be added to the MPLAB IDE status bar, and a **PICkit 2** tab will be added to the Output dialog.

See also **Chapter 5. "Troubleshooting"** for FAQs on using PICkit 2 Debug Express and descriptions of PICkit 2 MPLAB IDE error codes.

A.2 DEBUGGING FUNCTIONS

When you select PICkit 2 from the Debugger menu, debug items will be added to the following MPLAB IDE functions:

- Debugger Menu
- Right Mouse Button Menu
- Toolbars

A.2.1 Debugger Menu

Run F9

Execute program code until a breakpoint is encountered or until Halt is selected.

Execution starts at the current program counter (as displayed in the status bar). The current program counter location is also represented as a pointer in the Program Memory window. While the program is running, several other functions are disabled.

Animate

Animate causes the debugger to actually execute single steps while running, updating the values of the registers as it runs.

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

To Halt Animate, use the menu option *Debugger>Halt*, the toolbar Halt or <F5>.

Halt F5

Halt (stop) the execution of program code. When you click Halt, status information is updated.

Step Into F7

Single step through program code.

For assembly code, this command executes one instruction (single or multiple cycle instructions) and then halts. After execution of one instruction, all the windows are updated.

For C code, this command executes one line of C code, which may mean the execution of one or more assembly instruction, and then halts. After execution, all the windows are updated.

Note:	Do not step into a `SLEEP` instruction.

Step Over F8

In C code, steps over the current line of code.

Step Out

Not available.

Reset F6

Issue a Reset sequence to the target processor. This issues a $\overline{\text{MCLR}}$ to reset the Program Counter to the Reset vector.

Breakpoints

Open the Breakpoint dialog. Set, enable or disable breakpoints in this dialog. For more information on this dialog, see the MPLAB IDE on-line help file.

Note:	You may also right click or double click on a line of code to set a simple breakpoint.

Program

Download your code to the target device.

Read

Read target memory. Information uploaded to MPLAB IDE.

Read EEData

Read target EEData memory. Information uploaded to MPLAB IDE.

Connect

Attempt to establish or re-establish communications between the PC and PICkit 2. The progress of this connection is shown on the **PICkit 2** tab of the Output dialog.

Download OS

Download the latest PICkit 2 operating system.

Settings

Open the PICkit 2 Settings dialog (see **Section A.4 "Settings Dialog"**.) Set up automatic options and enable/disable warnings.

A.2.2 Right Mouse Button Menu

The following will appear on the right mouse menus in code displays, such as program memory and source code files:

Set/Remove Breakpoint

Set or remove a breakpoint at the currently selected line.

© 2008 Microchip Technology Inc.

Enable/Disable Breakpoint

Enable or disable a breakpoint at the currently selected line.

Breakpoints

Remove, enable or disable all breakpoints.

Run To Cursor

Run the program to the current cursor location. Formerly Run to Here.

A.2.3 Toolbars

When PICkit 2 is selected as a debugger, these toolbars are displayed in MPLAB IDE:

- Standard debug toolbar (Run, Halt, Animate, Step Into, Step Over, Step Out, Reset).
- PICkit 2 debug toolbar (Program, Read, Read EEData, Connect).

A.3 PROGRAMMING FUNCTIONS

When you select PICkit 2 from the Programmer menu, program items will be added to the following MPLAB IDE functions:

- Programmer Menu
- Toolbar

A.3.1 Programmer Menu

Program

Program memory areas: program memory, EEDATA memory, Configuration bits and ID locations.

Read

Read memory areas: program memory, EEDATA memory, Configuration bits and ID locations.

Verify

Verify programming of memory areas: program memory, EEDATA memory, Configuration bits and ID locations.

Erase

Erase all data on the device including program memory, EEDATA memory ID and Configuration bits.

Blank Check

Check to see that all device memory is erased/blank.

Read EEDATA

Read EEDATA memory.

Connect

Establish communications between the PICkit 2 and the PC.

Download OS

Download PICkit 2 operating system/firmware.

Release from Reset

Set $\overline{\text{MCLR}}$ to V_{DD}.

Hold in Reset

Set $\overline{\text{MCLR}}$ to ground (zero).

Settings

Open the PICkit 2 Settings dialog (see **Section A.4 "Settings Dialog"**.) Set up automatic options and enable/disable warnings.

A.3.2 Toolbar

Program the target device

Program memory areas: program memory, EEDATA memory, Configuration bits and ID locations.

Read target device memories

Read memory areas: program memory, EEDATA memory, Configuration bits and ID locations.

Read the target EEDATA memory

Read EEDATA memory.

Verify the contents of the target device

Verify programming of memory areas: program memory, EEDATA memory, Configuration bits and ID locations.

Erase the target device memories

Erase all data on the device including program memory, EEDATA memory, ID and Configuration bits.

Verify that target memories are erased

Check to see that all device memory is erased/blank.

Release from Reset

Set the target $\overline{\text{MCLR}}$ pin to V_{DD}.

Hold in Reset

Set the target $\overline{\text{MCLR}}$ pin to V_{SS}.

Connect

Establish communications between the PICkit 2 and the PC.

© 2008 Microchip Technology Inc.

A.4 SETTINGS DIALOG

Select *Debugger>Settings* or *Programmer>Settings* to open the settings dialog and set up PICkit 2.

A.4.1 Settings Tab

This tab of the PICkit 2 Settings dialog allows you to set automatic options

Connect on startup	When PICkit™ 2 is selected as the programmer, immediately attempt to connect.
Program on successful build	When a project builds successfully, immediately attempt to program the device with the updated code.
Run after successful program	When a device program successfully, immediately attempt to run the program.
Output to debug file	Output debug information to a file whenever the program is run.

A.4.2 Warnings Tab

This tab of the PICkit 2 Settings dialog allows you to enable or disable warning messages displayed in the Output window.

- To enable/disable specific messages, click the check box next to the message to check/uncheck it.
- To enable/disable all messages, click **Check All**/**Uncheck All**.

NOTES:

© 2008 Microchip Technology Inc.

Appendix B. PICkit 2 Schematics

PICkit 2 Development Programmer/Debugger schematic diagrams are shown here. Demo board schematics are found in their respective user's guides.

FIGURE B-1: PICkit™ 2 SCHEMATIC DIAGRAM (PAGE 1 OF 2)

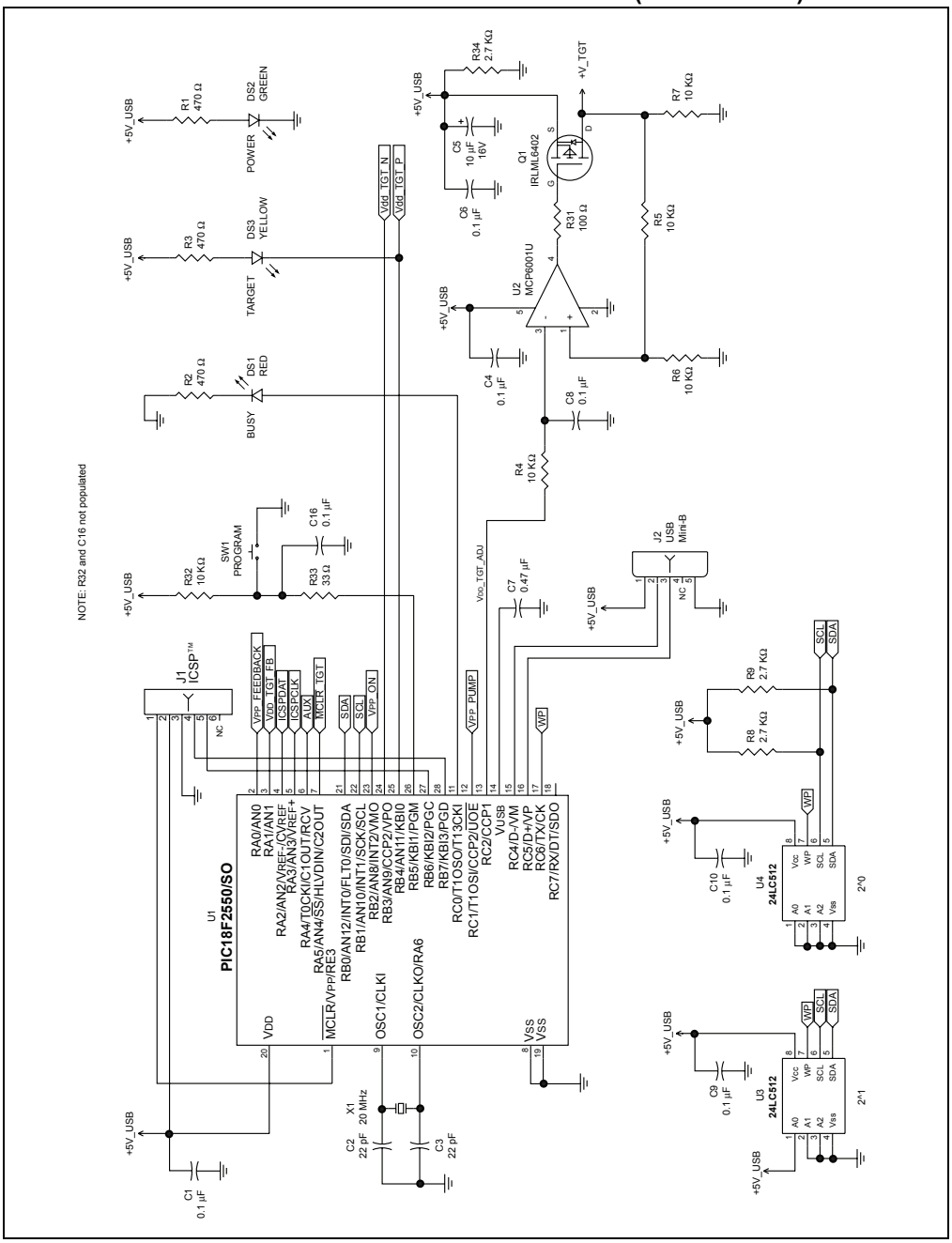
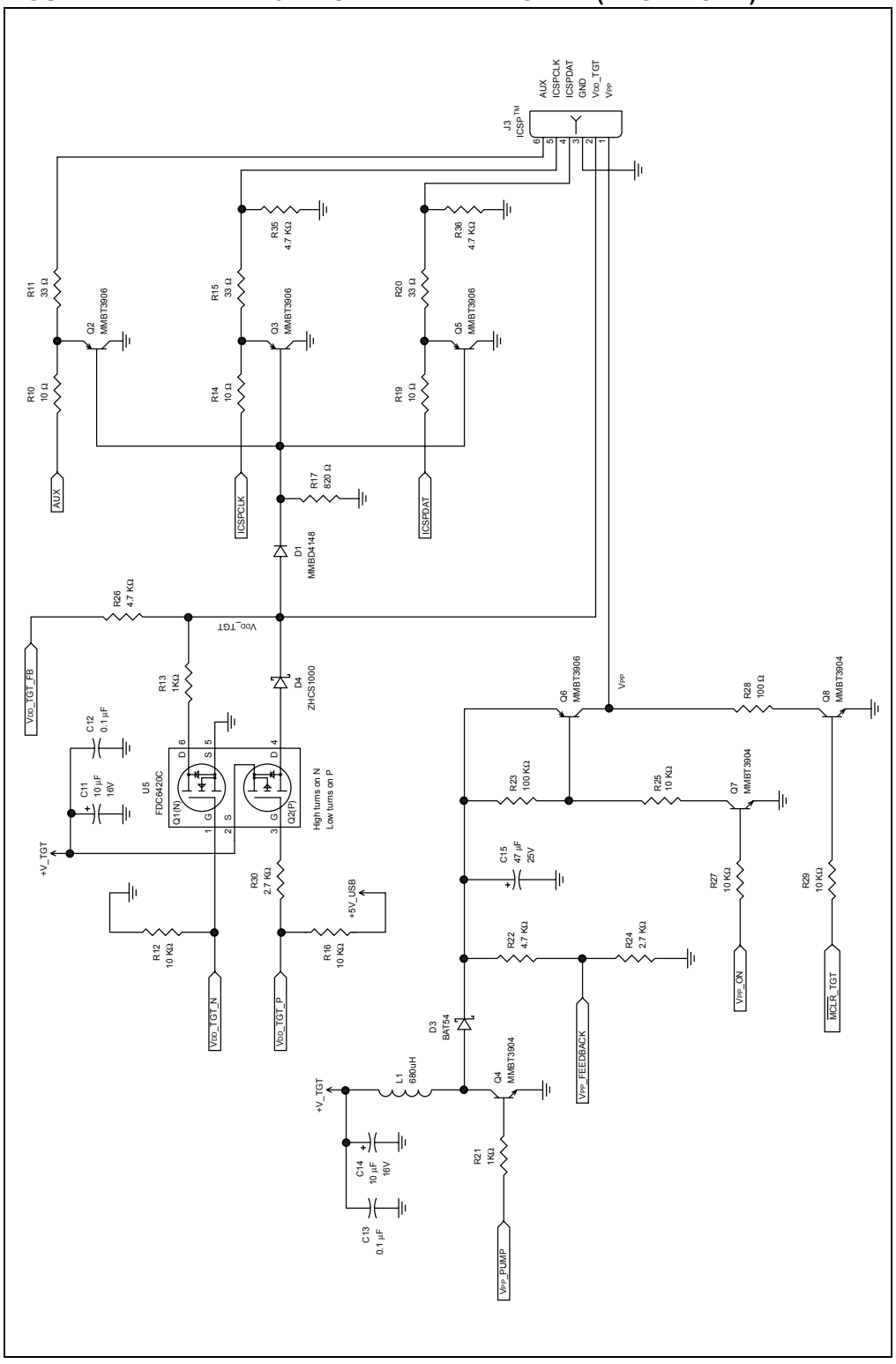


FIGURE B-2: PICkit™ 2 SCHEMATIC DIAGRAM (PAGE 2 OF 2)



© 2008 Microchip Technology Inc.

Index

Numerics

44-Pin Demo Board 31

A

Animate 71
Auto Import Hex + Write Device 25
Auto-Detect 12

B

Blank Check 11, 24, 73, 74
Breakpoints 72, 73

C

Cable Length 30
Calibrate VDD & Set Unit ID 12
Check Communication 12
Code Protect 11, 23
Connect 72
Connecting 73
Connector, 6-Pin 8
Current Limit 18
Customer Notification Service 5
Customer Support 6

D

Data EEPROM Memory 14
Data Protect 11
Debug Express 31
Debugging Functions 71
Device Selection 12, 16
Documentation
  Conventions 3
  Layout 2
Download OS 72, 73
Download PICkit 2 Programmer Operating System 12

E

Enable Code Protect 11
Enable Data Protect 11
Erase 11, 24, 73, 74
Explorer 16 31
Export Hex 11

F

Fast Programming 12
File History 11
Firmware 12
Firmware, Update 61
Force PICkit 2 12
Force Target 12

H

Halt 71
Hex File, Import 19
Hold Device in Reset 11
Hold in Reset 74

I

ICSP 27
ICSPCLK/ICSPDAT 28
Import Hex 11
Import Hex File 19
Install Hardware 15
Install Software 16
Internet Address, Microchip 5

L

LEDs 8
Low Pin Count Demo Board 31

M

$\overline{\text{MCLR}}$ 73, 74
Microsoft Windows 49
MPLAB IDE 71

O

Operating System 12
Operating System, Update 61
OSCCAL 12

P

PGC/PGD 28
Program 72, 73, 74
Program Memory 14
Programmer Application 10
Programming Functions 73
Push Button 8

R

Read 23, 72, 73, 74
Read Device 11
Read Device + Export Hex File 25
Read EEDATA 73, 74
Read EEData 72
Reading, Recommended 4
Readme 4
Reconnect 74
Release from Reset 73
Reset
  Hold in 74
  Processor 72
  Release from 73
Run 71

S

Set OSCCAL 12
Settings Dialog 75
Settings tab 75
Step 72

T

Target Power 18
Target Vdd Source 12
Troubleshooting 49

U

Update Firmware 61
Update Operating System 61
USB Port 8
Use VPP First Program Entry 12

V

Vdd 13, 18, 19, 29, 49
Verify 11, 23, 73, 74
Verify Erased 74
Verify on Write 11
Vpp 28, 49
Vss 30

W

Warning Messages 75
Warnings tab 75
Warranty Registration 4
Web Site, Microchip 5
Write 20
Write Device 11
Write on PICkit Button 11

© 2008 Microchip Technology Inc.

NOTES:

WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://support.microchip.com
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-4182-8400
Fax: 91-80-4182-8422

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-572-9526
Fax: 886-3-572-6459

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

01/02/08

© 2008 Microchip Technology Inc.