MPLAB® PM3

Device Programmer

User's Guide

For MPLAB X IDE

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in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates.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

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

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, flexPWR, JukeBlox, KEELOQ, KEELOQ logo, Kleer, LANCheck, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC^{32} logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

The Embedded Control Solutions Company and mTouch are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, ECAN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, KleerNet, KleerNet logo, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, RightTouch logo, REAL ICE, SQI, Serial Quad I/O, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, 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.

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

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

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

ISBN: 978-1-63276-335-8

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

Object of Declaration: MPLAB® PM3 Device (Production) Programmer

EU Declaration of Conformity

This declaration of conformity is issued by the manufacturer.

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

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

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

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

Derek Carlson
Derek Carlson
VP Development Tools

16-July-2013
Date

NOTES:

Table of Contents

Chapter 7. Troubleshooting First Steps

Chapter 8. Frequently Asked Questions (FAQs)

Chapter 9. Error Messages

Chapter 10. Engineering Technical Notes (ETNs)

Chapter 11. Stand-Alone Reference

Chapter 12. MPLAB PM3 Card Reference

Appendix A. Hardware Specifications

Appendix B. Revision History

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® X IDE online help. Select the Help menu, and then Topics to open a list of available online help files.

INTRODUCTION

This chapter contains general information that will be useful to know before using the MPLAB® PM3 device programmer. Items discussed in this chapter include:

- Document Layout
- Conventions Used in this Guide
- Recommended Reading

DOCUMENT LAYOUT

This user's guide describes how to use the MPLAB PM3 device programmer as a development and production tool to program devices.

The document is organized as follows:

Part 1 - Getting Started

- **Chapter 1. About the Device Programmer** – describes the MPLAB PM3 and how it works.
- **Chapter 2. Using MPLAB PM3 with the MPLAB X IDE** – describes how to install MPLAB PM3 hardware and MPLAB software; explains how to set up the MPLAB X IDE and MPLAB PM3 to work together, and how to start MPLAB PM3 from MPLAB.
- **Chapter 3. Using MPLAB PM3 with the MPLAB IPE** – describes how to install MPLAB IPE software and use it with MPLAB PM3.
- **Chapter 4. Using MPLAB PM3 in Stand-Alone Mode** – provides instructions for using the MPLAB PM3 in Stand-Alone mode.
- **Chapter 5. Environments and the MPLAB PM3 Card** – provides instructions for using the MPLAB PM3 Card.

Part 2 - Troubleshooting

- **Chapter 6. Troubleshooting** – provides information on solving common problems.
- **Chapter 7. Troubleshooting First Steps** – discusses the first things to try if you are having issues with operation of the programmer.
- **Chapter 8. Frequently Asked Questions (FAQs)** – provides list of frequently asked questions that are useful for troubleshooting.
- **Chapter 9. Error Messages** – provides a list of error messages and suggested resolutions.
- **Chapter 10. Engineering Technical Notes (ETNs)** - provides a list of resolvable hardware issues.

Part 3 - Reference

- **Chapter 11. Stand-Alone Reference** – describes the commands available through the MPLAB PM3 LCD.
- **Chapter 12. MPLAB PM3 Card Reference** – describes the commands available specifically for the MPLAB PM3 Card.
- **Appendix A. Hardware Specifications** – describes how to connect MPLAB PM3 to a communication port and provides instructions on cleaning MPLAB PM3 socket modules.
- **Appendix B. Revision History**

CONVENTIONS USED IN THIS GUIDE

This user’s guide uses the following documentation conventions:

DOCUMENTATION CONVENTIONS

Description	Represents	Examples
Arial font:		
Italic characters	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 text with right angle bracket	A menu path	*File>Save*
Bold characters	A dialog button	Click **OK**
	A tab	Click the **Power** tab
N‘Rnnnn	A number in verilog format, where N is the total number of digits, R is the radix and n is a digit.	4‘b0010, 2‘hF1
Text in angle brackets < >	A key on the keyboard	Press <Enter>, <F1>
Courier New font:		
Plain Courier New	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 Courier New	A variable argument	*`file`*`.o`, where *`file`* can be any valid filename
Square brackets [ ]	Optional arguments	`mcc18 [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)`
		`{ ...`
		`}`

RECOMMENDED READING

This user's guide describes how to use MPLAB PM3 device programmer. Other useful documents are listed below. The following Microchip documents are available and recommended as supplemental reference resources.

Multi-Tool Design Advisory (DS51764)

Please read this first! This document contains important information about operational issues that should be considered when using the MPLAB PM3 device programmer.

Release Notes for MPLAB PM3

For the latest information on using MPLAB PM3 device programmer, read the release notes under "Release Notes and Support Documentation" on the Start Page. The release notes contain update information and known issues that may not be included in this user's guide.

MPLAB® PM3 Device Programmer Online Help File

A comprehensive help file for the programmer is included with MPLAB X IDE. Usage, troubleshooting and hardware specifications are covered. The online help file might be more up-to-date than the user's guide. Also, limitations are listed for various devices.

MPLAB® X IDE User's Guide and Online Help File

A comprehensive guide that describes installation and features of Microchip's MPLAB X Integrated Development Environment (IDE). The online help file provides the most up-to-date information available for the MPLAB X IDE.

MPLAB® IPE Online Help File

The online help file provides the most up-to-date information available for the MPLAB Integrated Programming Environment (IPE), including how to use SQTP programming with the MPLAB PM3.

In-Circuit Serial Programming™ (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.

MPLAB® PM3 ICSP™ Design Guide (DS51474)

This is a quick guide with information on important considerations for implementing ICSP when using the MPLAB PM3 device programmer.

Part 1 – Getting Started

NOTES:

Chapter 1. About the Device Programmer

1.1 INTRODUCTION

This chapter presents an overview of the features and requirements of MPLAB® PM3. The following topics are discussed in this chapter:

- MPLAB PM3 Device Programmer Defined
- How the MPLAB PM3 Device Programmer Helps You
- MPLAB PM3 System Components

1.2 MPLAB PM3 DEVICE PROGRAMMER DEFINED

The Microchip MPLAB PM3 universal device programmer is easy to use, and operates with or without a PC. Using interchangeable programming socket modules, MPLAB PM3 enables you to quickly program devices from the entire line of Microchip microcontrollers. The MPLAB PM3 features a large, bright LCD screen to view menus, programming statistics and status information.

MPLAB PM3 device programmer can be used with MPLAB X IDE (Integrated Development Environment) running under supported operating systems (see “`Readme for MPLAB PM3.htm`” file for support list) to create a project and debug your program (using a debugger, not the MPLAB PM3). The MPLAB PM3 can also be used with the MPLAB IPE (Integrated Programming Environment) to program devices. The MPLAB PM3 can also be used as a stand-alone production programmer.

The MPLAB PM3 has exceptional programming speed to allow high production throughput, especially important for large memory devices, and includes a Secure Digital/Multimedia Card slot for easy and secure data storage and transfer.

The MPLAB PM3 device programmer is designed with 40 programmable socket pins, allowing each socket module to be configured to support many different devices, requiring fewer socket modules to support the entire line of Microchip parts. The programmable socket pin drivers use multi-pin connectors for high reliability and quick interchange. An adapter even allows PROMATE II socket modules to be used on the MPLAB PM3 device programmer.

1.3 HOW THE MPLAB PM3 DEVICE PROGRAMMER HELPS YOU

The MPLAB PM3 device programmer helps you by providing the following benefits for developing a project, and production programming devices.

Development:

- MPLAB PM3 contains field-upgradeable firmware via the latest MPLAB X IDE software package found at the Microchip web site (http://www.microchip.com). This will provide support for future Microchip devices, thus providing the latest programming algorithms to support Microchip PIC® microcontroller (MCU) devices and other Microchip parts.
- PC Host mode for full control - using MPLAB X IDE as the interface, MPLAB PM3 becomes another tool in the MPLAB X IDE, allowing you to quickly compile, test, debug (using a debugger, not the MPLAB PM3), and program your firmware.
- Program devices using ICSP™ (In-Circuit Serial Programming™) on the target board, and user $\overline{\text{GO}}$, PASS and FAIL signals to interface with the MPLAB PM3.
- Read code from an unprotected device into the program memory window of MPLAB X IDE for debugging.
- Verify that the code in the target device matches your firmware.

Production:

- MPLAB PM3 is easy to use and flexible in programming Microchip devices and package types. (Interchangeable socket modules are sold separately.)
- Buzzer notification is helpful in noisy environments.
- Using MPLAB IPE, you can import the hex file created in the MPLAB X IDE and program your devices.
- Stand-Alone mode for programming without a PC (without a PC connection to MPLAB PM3), the unit operates as a stand-alone device programmer. The main programmer features of the MPLAB PM3 are available, including Read, and Program and Verify. However, a PC connection is required to download any operating system updates.
- With an optional MPLAB PM3 Card (Secure Digital [SD], Multimedia Card [MMC], or microSD card with SD Converter) inserted, you can store and transport device settings for programming, as well as for environments.
- Program memory, configuration bits, EEPROM data memory, ID locations and calibration data into devices.
- Verify that devices are blank.
- Program unique, serialized ID numbers into your firmware using Serial Quick Turn Programming (SQTP^{SM}) files while in PC host mode.

1.4 MPLAB PM3 SYSTEM COMPONENTS

The MPLAB PM3 device programmer system consists of the following:

- MPLAB PM3 device programmer (see Figure 1-1 and Figure 1-2)
- Socket connectors

> **Note:** A complete line of socket modules is available. The socket modules can be ordered separately for the devices that will be programmed. Please check the Development Tools Selector (DTS) located on the Microchip web site (www.microchip.com/dts) for socket modules.

- USB interface cable to connect to any standard PC USB port
- ICSP cable
- Power supply and power cables
- MPLAB X IDE software – an Integrated Development Environment including a text editor, project manager and simulator for debugging and programming. MPASM assembler, MPLINK object linker and MPLIB object librarian software is also included.
- MPLAB IPE software – an Integrated Programming Environment for programming devices in a production environment.

FIGURE 1-1: MPLAB PM3 DEVICE PROGRAMMER

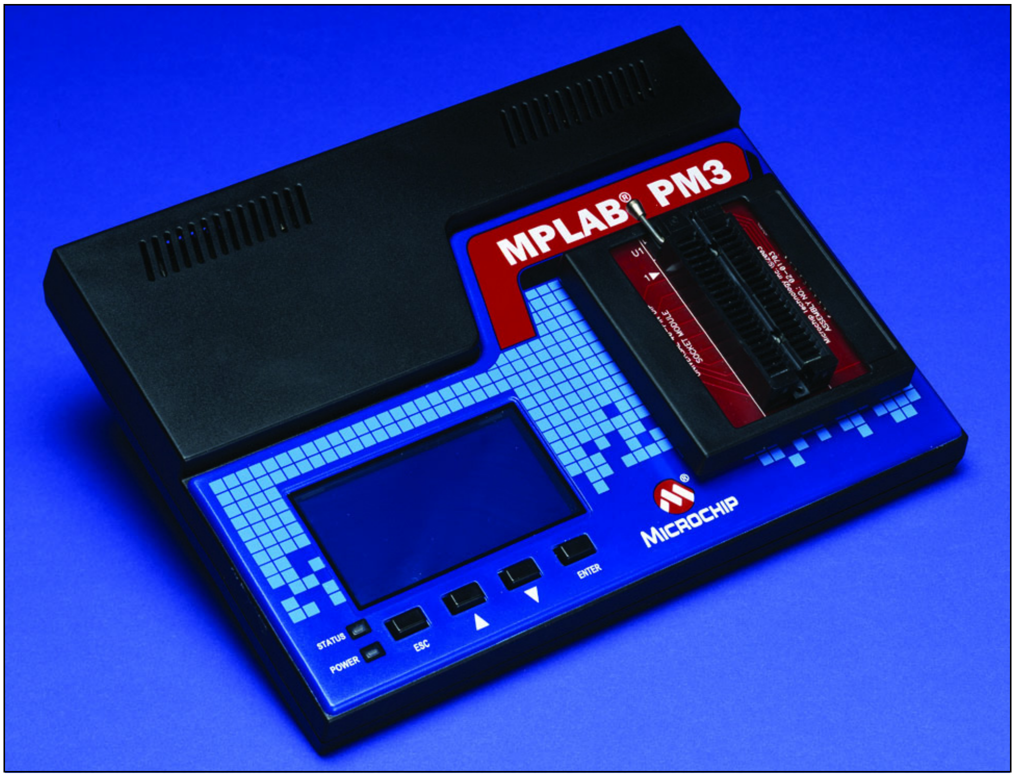
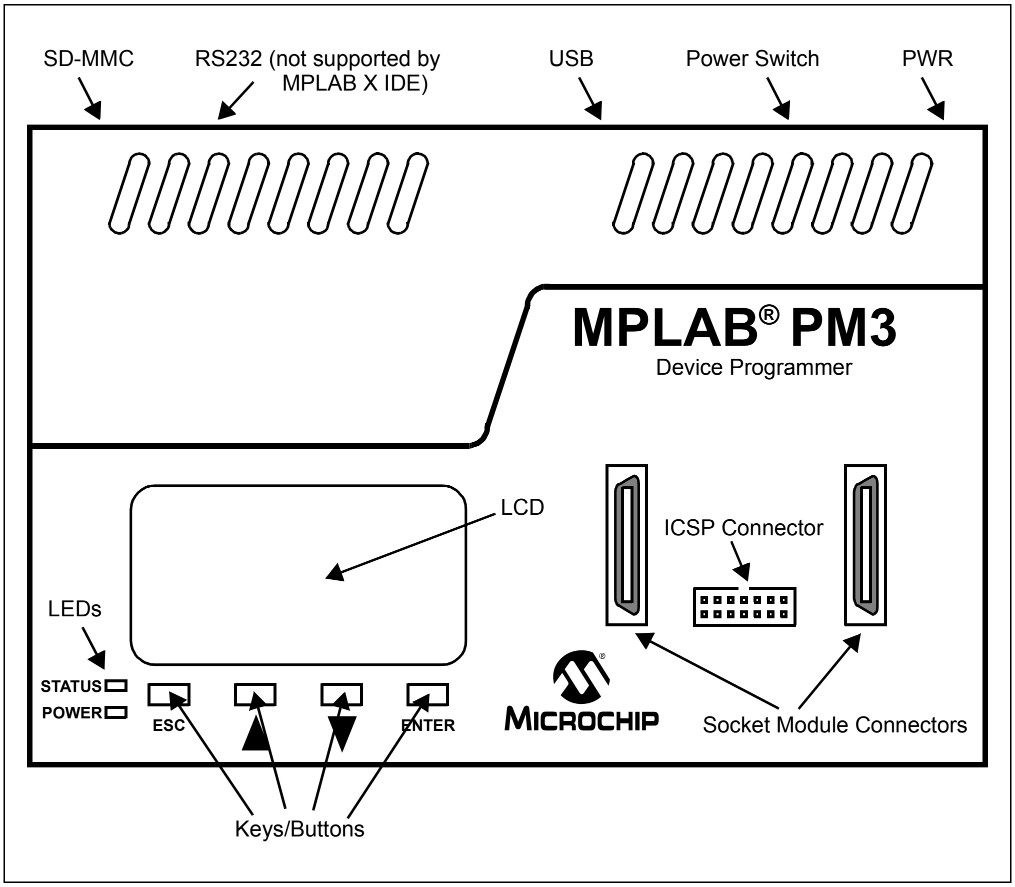


FIGURE 1-2: MPLAB PM3 DETAILED



Chapter 2. Using MPLAB PM3 with the MPLAB X IDE

2.1 INTRODUCTION

Generally, you use the MPLAB X IDE to create a project, debug your code, and create a hex file. Then, you use that file to program devices with the MPLAB PM3.

When you have your hex file and are ready to program, skip to **Chapter 3. "Using MPLAB PM3 with the MPLAB IPE"** or **Chapter 4. "Using MPLAB PM3 in Stand-Alone Mode"**.

This chapter describes how to install MPLAB PM3 hardware and software and set up the programmer for use with the MPLAB X IDE.

The following topics are discussed here:

- Installation and Setup Overview
- Installing MPLAB PM3 Hardware
- Powering Up MPLAB PM3
- Programmer Settings
- Setting Configuration Bits and User ID
- Toolbar Buttons in MPLAB X IDE
- Inserting and Programming a Device
- Special Programming

2.2 INSTALLATION AND SETUP OVERVIEW

Note: Detailed instructions for procedures that appear in the following list with an asterisk "*" are available in the Help file "Getting Started with MPLAB X IDE".

In summary:

1. Install MPLAB X IDE.*
2. Install the USB drivers.*

Return to this page and the following instructions:

3. Install MPLAB PM3 hardware.
4. Power up MPLAB PM3.
5. Install the language toolsuite/compiler for use on your project – refer to compiler documentation for details about the compiler.
6. Launch MPLAB X IDE.
7. Create a hex file by opening an existing project (*File>Open Project*) and building it.
8. Use the project Properties dialog (*File>Project Properties*) to select the MPLAB PM3 programmer as the tool and set up options, such as memory ranges.*
9. Insert and Program a Device

The following items should be noted:

1. Installing USB drivers on Windows OS systems requires that you follow specific instructions. See MPLAB X IDE documentation for details: www.microchip.com/mplabx/
2. Each programmer contains a unique identifier which, when first installed, will be recognized by the OS, regardless of which computer USB port is used.
3. Configuration bits must now be set in code. You can set up Configuration bits in the Configuration window (*Window>PIC Memory Views>Configuration Bits*) and then click "Generate Source Code to Output".
4. MPLAB X IDE operation connects to the hardware tool at runtime (Run or Debug Run). To always be connected to the hardware tool (as in MPLAB IDE v8), click in the *Tools>Options*, **Embedded** button, **Generic Settings** tab, "Maintain active connection to hardware tool" checkbox.

2.3 INSTALLING MPLAB PM3 HARDWARE

IMPORTANT: Do **not** allow the Windows OS to pick a USB driver. Follow the procedure specified in the “Before You Begin” section of the online help for MPLAB X IDE. See the subsection titled “Install the USB Device Drivers (For Hardware Tools)”.

The MPLAB PM3 hardware is simple to set up:

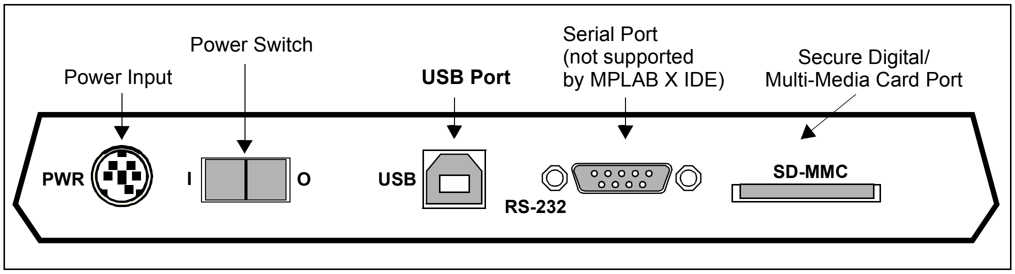
- If you are using MPLAB X IDE:
  - Attach the Communications Cable
  - Connect the Power Supply to the MPLAB PM3
  - Install Socket Module (or Attach the ICSP Cable)
- If you are using the MPLAB PM3 Card in the MPLAB PM3:

  Insert the MPLAB PM3 Card into the SD-MMC port on the back of the programmer.

2.3.1 Attach the Communications Cable

MPLAB PM3 provides communications with the host PC via a USB connector. See Figure 2-1 for the USB port location.

FIGURE 2-1: BACK VIEW OF MPLAB PM3


Power Switch
Power Input
USB Port
Serial Port (not supported by MPLAB X IDE)
Secure Digital/ Multi-Media Card Port
PWR
I
O
USB
RS-232
SD-MMC

- Connect one end of the USB cable to a USB port on your PC.
- Connect the cable from the PC USB port to the corresponding USB connector on the back of the MPLAB PM3.

> **Note:** See IMPORTANT statement above for USB driver.

2.3.2 Connect the Power Supply to the MPLAB PM3

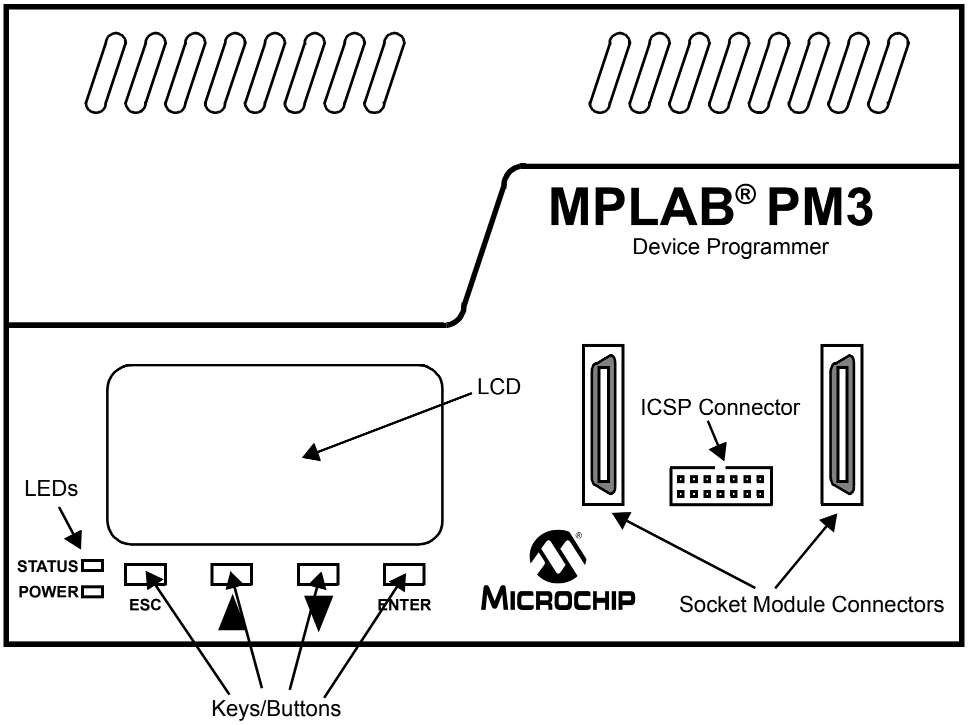
MPLAB PM3 comes with a proprietary external power supply.

1. Make sure that the power switch on the back of the unit is in the “**O**” (off) position (see Figure 2-1).
2. Plug the power supply into a power socket and connect the power supply cable to the unit.

2.3.3 Install Socket Module (or Attach the ICSP Cable)

Socket modules are sold separately. The MPLAB PM3 comes with an 18-inch ICSP cable for ICSP programming. See Figure 2-2 for the location of the socket module connectors and ICSP connector.

FIGURE 2-2: TOP VIEW OF MPLAB PM3



2.3.3.1 INSTALL SOCKET MODULE

Socket modules are available to accommodate each device package.
Refer to Development Tools Selector (DTS) located on the Microchip web site (www.microchip.com/dts) for Microchip devices, tools and socket modules.

Note:	MPLAB PM3 allows hot swapping of socket modules. If the status LED is not lit, sockets can be switched.

2.3.3.1.1 For MPLAB PM3 Socket Modules

1. Align the connectors on the socket module with the connectors on the MPLAB PM3 (Figure 2-2).
2. Push the socket module down, evenly mating the connectors.

Whenever the socket module is changed, it is good practice to insert a known blank device and do a blank check.

2.3.3.1.2 For PRO MATE II Socket Modules

> **Note 1:** To use PRO MATE II socket modules with the MPLAB PM3, you must obtain an AC164350 adapter kit. See the Development Tools Selector (DTS) located on the Microchip web site (www.microchip.com/dts).
>
> **2:** The PRO MATE II ICSP socket module is not supported by the MPLAB PM3. An 18-inch ICSP cable is included with MPLAB PM3 eliminating the need for an ICSP socket.

1. Align the connectors on the adapter with the connectors on the MPLAB PM3.
2. Push the adapter down, evenly mating the connectors.
3. Align the socket module with the adapter on the MPLAB PM3.
4. Tighten the two socket module thumb screws evenly and simultaneously. Avoid over-tightening; they should be finger-tight only.

> **Note:** The gold connector strips on the PRO MATE II socket module are relatively fragile. Avoid touching them with the socket module screws and avoid over-tightening the screws.

It is a good practice to insert a device that you know is blank and do a blank check every time the socket module is changed.

2.3.3.2 INSTALL ICSP CABLE

1. Connect the ICSP cable connector to the ICSP socket on the MPLAB PM3 (Figure 2-2).
2. Do not connect the individual leads; leave them unconnected at this time.

2.3.4 Insert the MPLAB PM3 Card

1. Align the MPLAB PM3 card with the SD-MMC slot on the back of the MPLAB PM3 programmer, and insert the notched corner end of the card into the slot. The card is keyed so that it cannot be inserted incorrectly.
2. Push the card into the slot.
3. To remove the card, push it inwards gently, and then release it (allowing space for it to eject).

2.4 POWERING UP MPLAB PM3

Once you have connected the hardware and installed the software, you are ready to turn on the MPLAB PM3. Toggle the power switch on the side of the MPLAB PM3 to “**I**” (on). The power switch is illustrated in Figure 2-1.

> **Note:** MPLAB PM3 automatically performs a self-test to ensure that the programmer is functional. No calibration is required.

If any portion of the self-test fails, MPLAB PM3 will display the corrective course of action on the LCD panel.

Upon normal start-up, the MPLAB PM3 will beep once. See **Section 9.3 “Error Messages – LCD”** for information on beep codes and LCD error messages. The MPLAB X IDE will provide further information to help you debug the issue.

After a successful power-up, you should see the following types of messages appear on the LCD panel that is on the front of the MPLAB PM3:

- MPLAB PM3 splash screen
- Version numbers and copyright dates
- MPLAB PM3 menu

At this point, you are ready to use the MPLAB PM3. If you are going to use the MPLAB PM3 in Stand-Alone mode, please refer to **Chapter 4. “Using MPLAB PM3 in Stand-Alone Mode”**.

If you are using the MPLAB PM3 Card with either the MPLAB X IDE or in Stand-Alone mode, also refer to **Chapter 5. “Environments and the MPLAB PM3 Card”**.

2.4.1 Indicator Lights and Buzzer

Two indicator lights (LEDs) are located on the front of the programmer. A buzzer, for audio indication, is incorporated into the programmer, as well.

TABLE 2-1: STATUS LED INDICATIONS

LED	Condition
Red	Booting up, Programming failed, Other error
Orange	Working/Busy
Green	Programming passed

TABLE 2-2: POWER LED

LED	Condition
On	Programmer powered
Off	Programmer not powered

2.4.2 Powering Down the MPLAB PM3

Toggle the power switch on the back of the MPLAB PM3 to “**O**” (off).

2.5 PROGRAMMER SETTINGS

Use MPLAB X IDE to set up programming options in the Project Properties window:

1. Right click on the project name in the Projects window and select “Properties” from the menu.
2. Under “Categories”, click on the hardware tool you will use to program your code, e.g., PM3.
3. Review the settings under the “Memories to Program” options category.
   If you wish to use a “Preserve Memory” option, ensure that your code is ***not*** code protected. Code is preserved when the programmer reads the section it needs to save, performs a bulk erase of the device, reprograms the device and then rewrites the area that is preserved with what was saved.
4. Review the settings under the “Program Options” options category.
5. Depending on your hardware tool, there might be other programming option categories. Review each one to ensure the settings are correct for your project.

Available option categories are:

- Memories to Program
- Program Options
- Voltages
- ICSP Options
- Firmware

2.5.1 Memories to Program

Selection	Description
Auto select memories and ranges	**Allow PM3 to Select Memories** - The programmer uses your selected device and default settings to determine what to program. **Manually select memories and ranges** - You select: the type and the range of memory to program (see rows that follow).
Configuration Memory	Check to program configuration memory.
ID	Check to program the User ID.
Program Memory	Select this check box to program Program Memory. The program memory range is determined by the Program memory start and end address fields.
Program Memory Start (hex)	Type the start address for the range of program memory (in hex).
Program Memory End (hex)	Type the end address for the range of program memory (in hex).
Preserve Program Memory	Check to **not** program the target program memory range specified (see rows that follow).
Preserve Program Memory Start (hex)	The starting hex address range in target program memory to be preserved when programming, reading, or verifying. This memory is read from the target and overlayed with existing MPLAB X IDE memory.
Preserve Program Memory End (hex)	The ending hex address range in target program memory to be preserved when programming, reading, or verifying. This memory is read from the target and overlayed with existing MPLAB X IDE memory.
Preserve ID Memory	Check to preserve ID Memory when programming

2.5.2 Program Options

Selection	Description
Erase All Before Program	When this option is enabled (check box is checked), the device will be erased before it is programmed. If it is disabled (unchecked), the device will not be erased before it is programmed. **NOTE:** Selection of the full program memory range will cause the device to be erased even if the "Erase All Before Program" option is disabled. This options is disabled for one-time programmable (OTP) devices, such as the PIC12C509A.
Verify Device Id before Program	When this option is enabled (check box is checked), the device ID will be verified before it is programmed.

2.5.3 Voltages

Set the following with the voltages appropriate for the selected device.

- VDD Min
- VDD Max
- VDD Nom
- VPP
- VDD App

2.5.4 ICSP Options

Check if you want to "Power target circuit from PM3."

2.5.5 Firmware

Selection	Description
Use Latest Firmware	Check to use the latest firmware. Uncheck to select the firmware version below.
Firmware File	Click in the right-hand text box to search for a firmware file (.mjc) to associate with the programmer.

2.6 SETTING CONFIGURATION BITS AND USER ID

Configuration bit and user ID values are set in MPLAB X IDE:

1. Select *Window>PIC Memory Views>* and choose either “Configuration Bits” or “User ID” to set the display.
2. Double click on a value to change it.

2.7 TOOLBAR BUTTONS IN MPLAB X IDE

The MPLAB X IDE Run toolbar consists of icons for the running, debugging and programming functions.

- Set Project Configuration – selects project configuration. Should be “default”.
- Build Project – builds all the project files.
- Clean and Build Project – deletes files from previous builds and then builds the all project files.
- Run Project – builds, programs the target and Runs the selected project.
- Hold in Reset – builds, programs the target and holds in Reset the selected project. (Not applicable for MPLAB PM3; for debuggers only.)
- Debug Project – builds, programs the target and Debug Runs the selected project. (Not applicable for MPLAB PM3; for debuggers only.)

2.8 INSERTING AND PROGRAMMING A DEVICE

To insert a device:

If you are using a socket module, insert the device to be programmed into the MPLAB PM3 socket. Position pin one on the device to match the pin one indicator in the socket module. Secure the device by pushing down the silver lever on the socket or closing the clamshell.

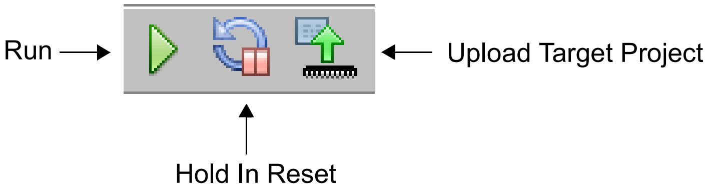
Note: If your device is EPROM, ensure it is blank before programming.

If you are using ISCP, refer to **Section 2.9.2 "ICSP Programming"** and consult **Section A.3 "ICSP Hardware Specifications"**.

To program a device:

MPLAB X IDE is a development, and not a production, environment. As such, the controls to program are based on development tools. The available programming functions are listed below.

FIGURE 2-3: PROGRAM ICONS



- Click **Run**: the project is built (if necessary) and device is programmed. The program will immediately begin execution on completion of programming.
- Click **Upload Target Project**: transfer what is in target memory to MPLAB X IDE.

2.9 SPECIAL PROGRAMMING

2.9.1 SQTP^{SM}

Serial Quick Turn Programming (SQTP) allows you to program a unique serial number into each device. This number can be used as an entry code, password, or ID number.

> **Note:** To use SQTP programming with MPLAB PM3, refer to the MPLAB Integrated Programming Environment (IPE) Online Help. The MPLAB IPE can be downloaded during the MPLAB X IDE installation.

2.9.2 ICSP Programming

The ICSP (In-Circuit Serial Programming) socket is an extension of the MPLAB PM3 device programmer that allows you to program PIC microcontroller devices that are already installed in the target board.

MPLAB PM3 comes equipped with an ICSP (In-Circuit Serial Programming) header and cable. You can locate this connector under where a socket module would be installed. See **Section A.3 "ICSP Hardware Specifications"** for header and cable pin-out information.

To program a device using ICSP:

1. Select *File/Project Properties* and click the PM3 in the Categories pane to display the Options for PM3.
2. From the Option categories drop-down list, select "ICSP Options."
3. Select "Power target circuit from PM3" if you want to power the circuit through the MPLAB PM3 instead of using power from the target board (Figure 4).
4. Connect the ICSP cable connector to the ICSP socket on the MPLAB PM3.
5. Connect the necessary cables to the header on your target board. Refer to Table A-2 in **Appendix A. "Hardware Specifications"** for cable pins, colors and signals.
6. Program the device. See **Section 2.7 "Toolbar Buttons in MPLAB X IDE"**.

> **Note:** For information on how to program a specific device using ICSP, consult the programming specification for that device. See the README for MPLAB PM3 for a list of supported devices. Programming specifications can also be found on the Microchip web site at www.microchip.com.

NOTES:

Chapter 3. Using MPLAB PM3 with the MPLAB IPE

3.1 INTRODUCTION

The MPLAB Integrated Programming Environment (IPE) is a software application that provides a simple interface to quickly program devices. IPE provides a secure programming environment for production programming. The IPE enables importing and exporting a hex file and performing other programming-related functions such as read, verify, erase, etc. The application operates in two modes, a production mode or a feature-rich GUI interface advanced mode.

Refer to the MPLAB Integrated Programming Environment (IPE) online help for detailed information on options and programming. This application enables you to import a hex file, an environment, or an SQTP file; create an environment or modify a file; determine settings for a production environment; and perform programming functions.

This chapter describes how to install MPLAB PM3 hardware and software and set up the programmer for use with the MPLAB IPE.

The following topics are covered here:

- Installation and Setup Overview
- Installing MPLAB PM3 Hardware
- Powering Up MPLAB PM3
- Inserting and Programming a Device
- Special Programming

3.2 INSTALLATION AND SETUP OVERVIEW

For items with an asterisk "*", refer to the Help file "Getting Started with MPLAB X IDE" for details.

In summary:

1. Install MPLAB IPE.
   This application is available during the MPLAB X IDE installation.
2. Install the USB drivers.*

Return to this page and the following instructions:

3. Install MPLAB PM3 Hardware.
4. Power Up MPLAB PM3.
5. Launch MPLAB IPE.
6. Select the device and tool.
7. Connect to the tool (MPLAB PM3).
8. Import a Hex file by selecting *File>Import>Hex*
9. Insert and Program a Device or Save Environment onto an SD card to use in Stand-Alone mode.

3.3 INSTALLING MPLAB PM3 HARDWARE

Note: **IMPORTANT**: Do not allow the Windows OS to pick a USB driver. Follow the procedure specified in the "Before You Begin" section of the online help for MPLAB X IDE. See the subsection titled "Install the USB Device Drivers (For Hardware Tools)".

The MPLAB PM3 hardware is simple to set up:

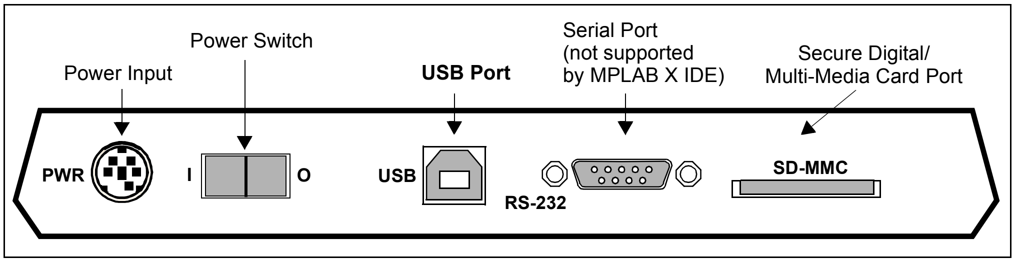
- Attach the Communications Cable
- Connect the Power Supply to the MPLAB PM3
- Install Socket Module (or attach the ICSP Cable)

- If you are using the MPLAB PM3 Card in the MPLAB PM3:
  - Insert the MPLAB PM3 Card into the SD-MMC port on the back of the programmer.

3.3.1 Attach the Communications Cable

MPLAB PM3 provides communications with the host PC via a USB connector. See Figure 3-1 for USB port location.

FIGURE 3-1: BACK VIEW OF MPLAB PM3



- Connect one end of the USB cable to a USB port on your PC.
- Connect the cable from the PC USB port to the corresponding USB connector on the back of the MPLAB PM3.

Note: If you are using USB and a "New Hardware Detected" notice appears on your PC, you must follow the directions on installing the proper driver or your MPLAB PM3 will not work.

3.3.2 Connect the Power Supply to the MPLAB PM3

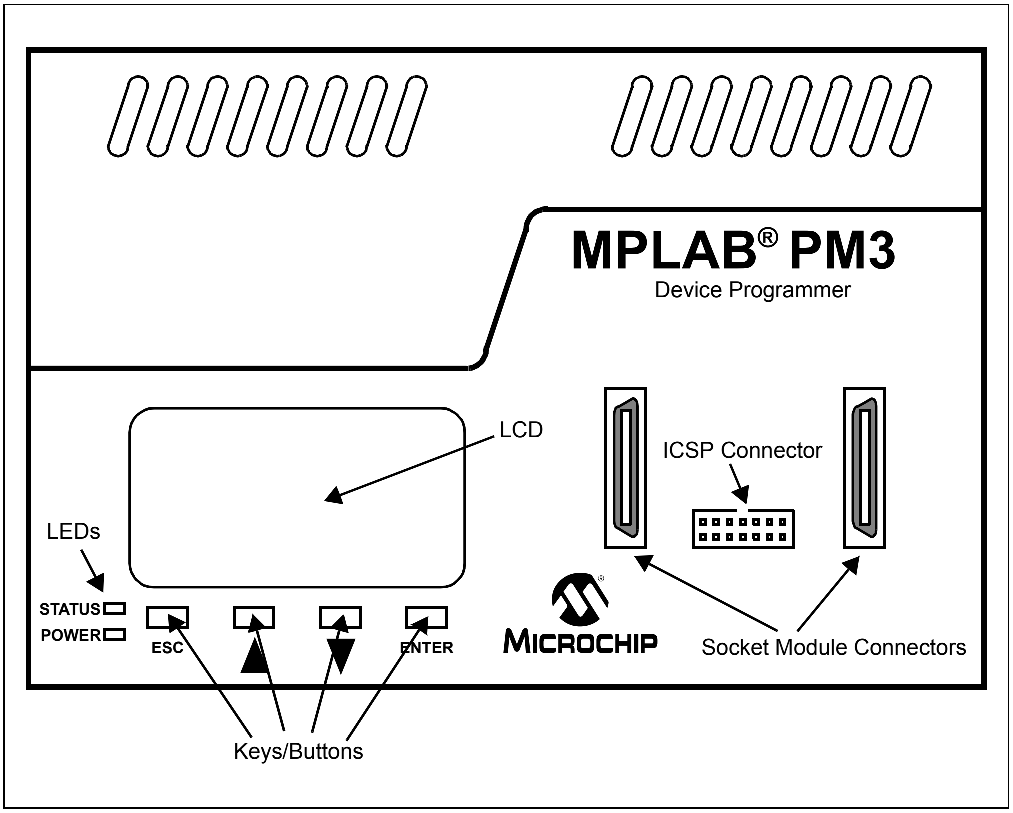
MPLAB PM3 comes with a proprietary external power supply.

1. Make sure that the power switch on the back of the unit is in the "O" (OFF) position (see Figure 3-1).
2. Plug the power supply into a power socket and connect the power supply cable to the unit.

3.3.3 Install Socket Module (or attach the ICSP Cable)

Socket modules are sold separately. The MPLAB PM3 comes with an 18-inch ICSP cable for ICSP programming. See Figure 3-2 for location of socket module connectors and ICSP connector.

FIGURE 3-2: TOP VIEW OF MPLAB PM3



3.3.3.1 INSTALL SOCKET MODULE

Socket modules are available to accommodate each device package.
Refer to Development Tools Selector (DTS) located on the Microchip web site (www.microchip.com/dts) for Microchip's devices, tools and socket modules.

> **Note:** MPLAB PM3 allows hot swapping of socket modules. If the status LED is not lit, sockets can be switched.

3.3.3.1.1 For MPLAB PM3 Socket Modules

1. Align the connectors on the socket module with the connectors on the MPLAB PM3 (Figure 3-2).
2. Push the socket module down evenly, mating the connectors.

Whenever the socket module is changed, it is good practice to insert a known blank device and do a blank check.

3.3.3.1.2 For PRO MATE II Socket Modules

> **Note 1:** In order to use PRO MATE II socket modules with MPLAB PM3, you must obtain an AC164350 adapter. See the Microchip Direct web site (www.microchipdirect.com).
>
> **2:** The PRO MATE II **ICSP** socket module is not supported by the MPLAB PM3. An 18-inch ICSP cable is included with MPLAB PM3 eliminating the need for an ICSP socket.

1. Align the connectors on the adapter with the connectors on the MPLAB PM3.
2. Push the adapter down evenly, mating the connectors.
3. Align the socket module with the adapter on the MPLAB PM3.
4. Tighten the two socket module thumb screws evenly and simultaneously. Avoid over-tightening; they should be finger-tight only.

> **Note:** The gold connector strips on the PRO MATE II socket module are relatively fragile. Avoid touching them with the socket module screws, and avoid over-tightening the screws.

Whenever the socket module is changed, it is good practice to insert a known blank device and do a blank check.

3.3.3.2 INSTALL ICSP CABLE

1. Connect the ICSP cable connector to the ICSP socket on the MPLAB PM3 (Figure 3-2).
2. Leave the individual leads unconnected at this time.

3.3.4 Insert the MPLAB PM3 Card

1. Align the MPLAB PM3 card with the SD-MMC slot on the back of the MPLAB PM3 programmer, and insert the notched corner end of the card into the slot. The card is keyed so that it only goes in one way.
2. Push the card into the slot.
3. To remove the card, push in to eject.

3.4 POWERING UP MPLAB PM3

When you have connected the hardware and installed the software, you are ready to turn on MPLAB PM3. Toggle the power switch on the side of the MPLAB PM3 to “I” (ON), refer back to Figure 3-1.

> **Note:** MPLAB PM3 automatically performs a self-test ensuring the programmer is functional. No calibration is required.

If any portion of the self-test fails, MPLAB PM3 will display the corrective course of action on the LCD panel. On normal start-up, the MPLAB PM3 beeps once. See **Section 9.3 “Error Messages – LCD”** for information on beep codes and LCD error messages. MPLAB X IDE will provide further information to help you debug the issue.

On a successful power-up, you should see the following types of messages appear on the LCD panel on the front of the MPLAB PM3:

- MPLAB PM3 splash screen
- Version numbers and copyright dates
- MPLAB PM3 menu

At this point, you are ready to use MPLAB PM3. If you are going to use MPLAB PM3 in Stand-Alone mode, please refer to **Chapter 4. “Using MPLAB PM3 in Stand-Alone Mode”.** If you are using the MPLAB PM3 Card with either the MPLAB IPE or in Stand-Alone mode, also refer to **Chapter 5. “Environments and the MPLAB PM3 Card”**.

3.4.1 Indicator Lights and Buzzer

Two indicator lights (LEDs) are located on the front of the programmer. A buzzer, for audio indication, is incorporated into the programmer as well.

TABLE 3-1: STATUS LED INDICATIONS

LED	Condition
Red	Booting up, Programming Failed, Other Error
Orange	Working/Busy
Green	Programming Passed

TABLE 3-2: POWER LED

LED	Condition
On	Programmer powered
Off	Programmer not powered

3.4.2 Powering Down the MPLAB PM3

Toggle the power switch on the back of the MPLAB PM3 to “O” (OFF), refer back to Figure 3-1.

3.5 INSERTING AND PROGRAMMING A DEVICE

To insert a device:

If you are using a socket module, insert the device to be programmed into the MPLAB PM3 socket. Position pin one on the device to match the pin-one indicator in the socket module. Secure the device by pushing down the silver lever on the socket or closing the clamshell.

> **Note:** If your device is EPROM, ensure that it is blank before programming.

If you are using ICSP, refer to **Section 3.6.2 “ICSP Programming”** and consult **Section A.3 “ICSP Hardware Specifications”**.

3.6 SPECIAL PROGRAMMING

3.6.1 SQTP

Serial quick turn programming (SQTP) allows you to program a unique serial number into each device. This number can be used as an entry code, password or ID number.

Refer to the MPLAB Integrated Programming Environment (IPE) Online Help for instructions on SQTP programming.

3.6.2 ICSP Programming

The ICSP socket is an extension of the MPLAB PM3 device programmer that allows you to program PIC microcontrollers that are already inserted into the target board.

MPLAB PM3 comes equipped with an ICSP (In-Circuit Serial Programming) header and cable. You can locate this connector under the place where a socket module would be installed. See **Section A.3 “ICSP Hardware Specifications”** for header and cable pin-out information.

Chapter 4. Using MPLAB PM3 in Stand-Alone Mode

4.1 INTRODUCTION

This chapter briefly describes how to use the MPLAB PM3 device programmer in Stand-Alone mode.

Using the MPLAB PM3 programmer in Stand-Alone mode is intended for production programming. The device programmer has a graphical LCD interface that provides complete control over a programming session.

The following topics are discussed here:

- Getting Started in Stand-Alone Mode
- Programming a Device

See **Chapter 11. "Stand-Alone Reference"** for detailed instructions on how to use the MPLAB PM3 in Stand-Alone mode.

4.2 GETTING STARTED IN STAND-ALONE MODE

MPLAB PM3 operating in Stand-Alone mode allows you to read, program and verify a device without using a PC. Stand-Alone mode is useful in situations where a PC may not be available or even required, as in the field or in a lab or production environment.

4.2.1 Installing a Socket Module

See **Section 2.3.3 "Install Socket Module (or Attach the ICSP Cable)"** for instructions.

Note: An 18-inch ICSP cable is included with MPLAB PM3, eliminating the need for an ICSP socket.

The socket module that is installed must be appropriate for the device being programmed. When the MPLAB PM3 device programmer is powered up, the unit automatically detects the type of socket module installed. If the part does not support the installed socket, the MPLAB PM3 will list the appropriate sockets to use.

Please refer to the DTS (www.microchip.com/dts) for selecting the appropriate socket module for your device family and device package.

Note 1: MPLAB PM3 allows hot swapping of socket modules. If the status LED is not lit, the sockets can be switched.

2: To use PRO MATE II socket modules with MPLAB PM3, you must obtain an AC164350 adapter kit.
See the Microchip Direct web site (www.microchipdirect.com).

4.2.2 Downloading a Hex File into MPLAB PM3 Memory

There are 3 ways to obtain the necessary hex file to begin to set up the MPLAB PM3 for Stand-Alone mode.

You can use any of the following methods:

- Use a PC to download the hex file.
- Use a master device to read the hex file into MPLAB PM3 memory.
- Use an SD card (refer to**Section 5.3 "MPLAB PM3 Card"**).

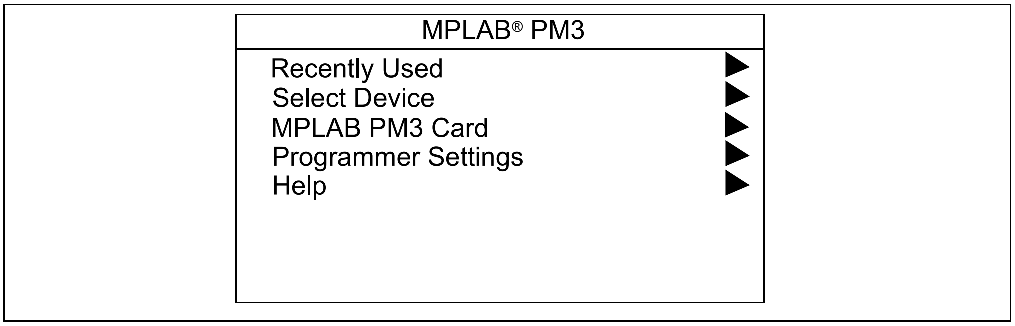
Refer to **Section 2.3 "Installing MPLAB PM3 Hardware"** for instructions about connecting the MPLAB PM3 to a PC.

Refer to MPLAB X IDE online Help for instructions on downloading a hex file.

4.3 PROGRAMMING A DEVICE

After applying power to MPLAB PM3, the unit briefly displays the MPLAB PM3 splash and versions screens. Then, the MPLAB PM3 main menu, shown below, is displayed.

FIGURE 4-1: MPLAB PM3 MAIN MENU



As shown in the figure, the following selections are on the main menu:

- Recently Used
- Select Device
- MPLAB PM3 Card (if inserted)
- (Stand-Alone) Programmer Settings
- Help

4.3.1 Recently Used

Select "Recently Used" to retrieve the last device used in the MPLAB PM3. This option displays the seven devices used most recently and enables quick access to the command menu for the device.

4.3.2 Select Device

Choose “Select Device” to indicate the device family and then, choose the device that is to be programmed. After the device is selected, the MPLAB PM3 displays the command menu with the commands that are applicable to the device type and its features, such as “Program/Verify Device”, “Verify Device” and “All Functions”.

4.3.2.1 PROGRAM/VERIFY DEVICE

Select “Program/ Verify Device” to program the device in the socket module with the hex file that was previously loaded into the internal memory of the device programmer. This is referred to as the image. If the hex file has not been loaded, the system displays a message stating “Valid image not present! Please download an image or use a master device before continuing.” Download the hex file into the MPLAB PM3 to program the device before attempting to continue.

The device programmer checks to see whether the particular device is blank. If the device is not blank, the device programmer asks if you want to continue. Answer “Yes” to continue. Answer “No” to return to the command menu.

If a flash device is being used, the device will be programmed immediately with the image in the MPLAB PM3. If a One-Time Programmable (OTP) device is being used and the “Blank Check Override” is enabled on the MPLAB PM3, two options – “Stop Programming” and “Continue” – will display if the device is not blank. Select “Stop Programming” if you want programming to stop. If not, select “Continue” and the device will be programmed.

The MPLAB PM3 programs the image of the hex file into the microcontroller device connected to the programmer.

Programming is performed at the VDD Applied and the VDD minimum and maximum voltages (for OTP devices) or at the VDD Nominal that has been established as the default per-operating-range for the device. See **Section 11.5.3.3 “Adjust Voltages”** if the verification needs to be performed at different voltage settings. Verification of the device against the MPLAB PM3 image occurs immediately after programming – together with the display of the resulting checksum – if both have been successful.

However, if errors occur during programming or verification, the first address of each memory area that failed is reported on the LCD.

4.3.2.2 VERIFY DEVICE

Select “Verify Device” to compare the contents of the device to that of the MPLAB PM3 memory (image). This comparison is performed at the VDD Applied, VDD minimum and maximum voltage (OPT devices) or VDD Nominal that has been established as the default per the operating range for the device. See **Section 11.5.3.3 “Adjust Voltages”** if the verification needs to be performed at different voltage settings.

If the content of the device matches the MPLAB PM3 image, the message “Passed!” and the resulting checksum are displayed on the LCD. If not, the first address of each memory area that failed is reported on the LCD.

4.3.2.3 READ DEVICE

Select "All Functions", then "Read Device", to retrieve the contents of the device and place it into the MPLAB PM3 memory. This image can then be used to program successive devices with the same information. Or, the information can be code protected and used for programming.

If the device is code protected, a message stating that the device is code protected is displayed on the LCD and the contents are not placed into the MPLAB PM3 memory. However, the resulting checksum is included in the display, if this option has been enabled in the settings of the programmer.

If the device is not code protected, the contents will be placed into the MPLAB PM3 memory. Two options – "No" and "Code Protect All?" – are displayed on the LCD, together with the resulting checksum. Select "No" if code protection of the current or successive devices is not desired. If not, select "Code Protect All?" if you want code protection on the current or successive devices.

After selecting one of the code protection options, the display returns to the "Read Device" option. Press the <ESC> key to return to the command menu for the device. If code protection was selected, "Program/Verify Device" must be used to program the current or successive devices with the code-protected image in the MPLAB PM3 memory.

4.3.2.4 ALL FUNCTIONS

Select the "All Functions" command to display all of the stand-alone commands that are applicable to programming, and reading, verifying, and displaying the status of the device that is currently selected. See **Section 11.5.3 "All Functions"**, for details on each of the options that might be available for a device.

4.3.3 MPLAB PM3 Card

If a memory card is inserted into the MPLAB PM3, the main menu includes the "MPLAB PM3 Card" option. Selecting this option displays the following additional options:

- Load an Environment
- View an Environment
- View Disk Contents
- Open a text File (*.txt)
- Card Properties

Refer to **Section 12.3 "MPLAB PM3 Card in Stand-alone Mode"** for details on these options.

4.3.4 (Stand-Alone) Programmer Settings

On the main menu, the “Programmer Settings” option offers access to the following options:

- Screen Contrast
- Buzzer Volume
- Socket Information
- Checksum Calculation
- Device ID Option
- Blank Check Override
- $\overline{\text{GO}}$ Pin Functionality

Refer to **Section 11.4.3 “(Stand-Alone) Programmer Settings”** for more details.

4.3.5 Help

The “Help” option displays the version number of the MPLAB PM3 OS Suite running on the MPLAB PM3, and the following submenu options:

- ICSP Connector Pinout
- Status Bar Icons
- About

Refer to **Section 11.4.4 “Help”** for details.

NOTES:

Chapter 5. Environments and the MPLAB PM3 Card

5.1 INTRODUCTION

This chapter describes the MPLAB PM3 card functions and introduces the concept of an environment.

The MPLAB PM3 device programmer uses a concept called an MPLAB PM3 environment. An environment is a snapshot of the current programming settings for a specific device. It contains all the necessary information to recreate the current programming state, including a binary image of the device's memory data, links to an optional SQTP file, specific part information, and optional, miscellaneous files.

a) The MPLAB PM3 card is a multimedia card, a stamp-sized flash memory card that weighs approximately 2 grams. It provides the following benefits/advantages:

- portable medium to store and transfer information
- fast copy and/or download
- high-storage capacity
- non-volatile solid-state (data is not lost when the power is turned off)
- write-protect switch on the card casing (for SD cards only)

The MPLAB PM3 card facilitates creative solutions for production programming. For example, you could store information (such as an environment, hex file, etc.) on an MMC card and send it to a manufacturer. At the manufacturer, the card could be placed into an MPLAB PM3 to program Microchip microcontrollers. A PC would not be needed at any point in that process.

The following topics are discussed here:

- MPLAB PM3 Environment
- MPLAB PM3 Card

5.2 MPLAB PM3 ENVIRONMENT

Note: To use environments with MPLAB PM3, refer to the MPLAB Integrated Programming Environment (IPE) Online Help. The MPLAB IPE can be downloaded during the MPLAB X IDE installation.

An environment contains information about the current programming state, memory data, SQTP file information, specific part information and optional miscellaneous files.

When an environment is saved, a directory is created on the removable MPLAB PM3 card (an SD-MMC memory card), on the selected drive of a PC, or on another storage device. The name of the directory created is the same as the name given to the Environment file.

There are additional operations, such as viewing, copying and deleting Environments. Refer to **Chapter 12. "MPLAB PM3 Card Reference"** for specific instructions for each of these operations.

An Environment is the encapsulation of the environment file (*.pm3), the binary bin file (*.bin), optional SQTP num file (*.num) and miscellaneous files.

The PM3 file (*.pm3) is part of the Environment. It is the high-level file that contains the MPLAB PM3 settings, name of the bin file, and name of the SQTP file.

The device's image (binary file) is the data that should be programmed into the device. Once the image is saved to the card, it can be removed from the MPLAB PM3 and sent to another location, inserted into another MPLAB PM3, and a device can be programmed with that information.

You can load an environment from the MPLAB PM3 card directly into the MPLAB PM3 programmer in Stand-Alone mode, or using the MPLAB IPE to transfer the file to the MPLAB PM3 programmer while it is connected to a PC.

5.2.1 Loading an Environment

Ensure that the MPLAB PM3 card is fully inserted into the SD-MMC card slot on the back of the MPLAB PM3 programmer, and that the programmer is powered on. The option for the MPLAB PM3 Card displays **only** when the card is inserted.

Using MPLAB IPE to Load an Environment

Refer to the MPLAB Integrated Programming Environment (IPE) Online Help for instructions.

Using Stand-Alone Mode to Load an Environment

1. From the MPLAB PM3 main menu, select "MPLAB PM3 card" and press <ENTER>.
2. From the MPLAB PM3 card menu, select "Load an Environment" and press <ENTER>.
3. From the PM3CARD:*.pm3 menu, select your environment folder and press <ENTER>.
4. Select your environment name from the your environment folder and press <ENTER>.

5.3 MPLAB PM3 CARD

The MPLAB PM3 card is an MMC card that is formatted in the FAT16 or FAT32-based file system, for cross-readability through an external reader. (Refer to the `Readme for MPLAB PM3` for a list of tested SD/MMC cards.) The MPLAB IPE is able to communicate to the MPLAB PM3 card through a USB interface. The MPLAB PM3 card can also be accessed directly through the MPLAB PM3 programmer in Stand-Alone mode.

5.3.1 MPLAB PM3 Card Through MPLAB IPE

Through the MPLAB IPE Environment menu, you can perform these operations on the MPLAB PM3 card:

- Properties
- Format
- Delete
- Copy
- View
- Save to PM3 SD Card

For detailed instructions, refer to the MPLAB Integrated Programming Environment (IPE) Online Help.

5.3.2 MPLAB PM3 Card Through MPLAB PM3 Programmer

In Stand-Alone mode, you can perform these operations on the MPLAB PM3 card:

- Load an Environment – loads the selected environment (that is stored on the MPLAB PM3 card) into the MPLAB PM3 programmer.
- View an Environment – displays information including the device, memory files and a description of an environment stored on the MPLAB PM3 card.
- View Disk Contents – displays the contents of the MPLAB PM3 card on the MPLAB PM3 LCD screen.
- Open a text File (*.txt) – displays the first 10 lines (approximately) of text from the selected text file on the programmer’s LCD screen.
- Card Properties – displays the properties of the MPLAB PM3 card, including card capacity, bytes that are free and used, and cluster size.

For detailed instructions, see **Section 12.3 “MPLAB PM3 Card in Stand-alone Mode”**.

NOTES:

Part 2 – Troubleshooting

NOTES:

Chapter 6. Troubleshooting

6.1 INTRODUCTION

The troubleshooting information in this chapter can help you resolve typical problems or obstacles in programming microcontroller devices. The following troubleshooting topics are discussed here:

- Troubleshooting Hardware
- Troubleshooting Operational Problems
- Troubleshooting Software
- Common Problems

6.2 TROUBLESHOOTING HARDWARE

Potential problems that could affect the hardware include the following scenarios:

- Communication Failure
- Ensuring Proper Socket Module Contact
- Socket Module Failure

6.2.1 Communication Failure

If communication fails, check your communications port. Refer to **Appendix A. "Hardware Specifications"** for information about connecting the MPLAB PM3 Device Programmer to a communications port.

6.2.2 Ensuring Proper Socket Module Contact

After changing a socket, insert a blank device, and do a blank check to ensure that the socket is making proper contact. A blank device will display as having been erased. An improperly-seated module socket could cause a device to fail the Verify process with errors claiming that the bad data is all zeros (0000).

6.2.3 Socket Module Failure

If you can program a master chip; and if you can read and try to program code-protected chips, but the chips fail the programming attempts; then potential socket pin damage might be the cause of the problem. Contact your Microchip Field Application Engineer (FAE) if your socket module is not operating properly.

6.3 TROUBLESHOOTING OPERATIONAL PROBLEMS

Potential problems that could affect the basic operation of the MPLAB PM3 include the following scenarios:

- Reading a Device Master in Stand-Alone Mode
- Device Pin Damage

6.3.1 Reading a Device Master in Stand-Alone Mode

When reading a device master in Stand-Alone mode, the Device Programmer asks the question, "Code Protect Parts" being programmed. Answer **Yes** to code protect the parts that you will be programming.

6.3.2 Device Pin Damage

On the smaller device packages (SSOP, PQFP and SOIC) the pins can bend easily and cause problems programming devices.

6.4 TROUBLESHOOTING SOFTWARE

Potential problems that could affect the software include the following scenarios:

- Establishing Communication with MPLAB PM3
- Operating System Update Needed

6.4.1 Establishing Communication with MPLAB PM3

MPLAB X IDE attempts to establish communication with the MPLAB PM3 when you select the programmer. If communication cannot be established, no programming can occur. A dialog box appears if the attempt to establish communication fails. If you encounter communications problems, review the following information.

1. Make sure that a USB cable is securely attached between the MPLAB PM3 and the PC.
2. Some system errors are caused by driver and hardware incompatibility.
3. You must use the Microsoft Windows communications driver that is native to the version of Windows that you are using.
4. Make sure you are not using a third party communications driver. Open your `SYSTEM.INI` file and look for the line in the [OPTIONS] section that reads

   ```
   COMM.DRV=COMM.DRV
   ```

   If this line reads differently you are using a different communications driver.

6.4.2 Operating System Update Needed

If the device you selected, when setting up the development mode in MPLAB X IDE, is not supported by the MPLAB PM3, a message box will display when you try to enable the programmer.

Make sure you allow the MPLAB PM3 to upgrade and get the latest versions of MPLAB X IDE software and the MPLAB PM3 operating system.

6.5 COMMON PROBLEMS

Potential problems that could affect the hardware include:

- Failure to Establish Communication with Programmer
- Device Pin Damage (see **Section 6.3.2 “Device Pin Damage”**)
- MPLAB PM3 Card Option Not Available on Programmer Menu

6.5.1 Failure to Establish Communication with Programmer

MPLAB X IDE attempts to establish communication with the programmer upon Run or Debug Run. MPLAB IPE attempts to establish communication with the programmer upon clicking **Connect**.

If communication cannot be established, no programming can occur. If the attempt to establish communication fails, it is reported in the Output window.

- General Communications Troubleshooting
- USB Communications

6.5.1.1 GENERAL COMMUNICATIONS TROUBLESHOOTING

1. Check that the USB cable is connected securely between the MPLAB PM3 and the host computer.
2. Check that the power supply is connected and the power LED on the programmer is lit.
3. Make sure that you are not using a third party communications driver in the PC. Open the SYSTEM.INI file and look for the line in the [OPTIONS] section that reads:

   COMM.DRV=COMM.DRV

   If this line reads differently, you are using a different communications driver.

6.5.1.2 USB COMMUNICATIONS

Make sure you used the MPLAB X IDE supplied USB driver for MPLAB PM3.

CAUTION
If the Windows OS picked a USB driver, MPLAB PM3 will not work and you will not be able to install the proper driver.

6.5.2 MPLAB PM3 Card Option Not Available on Programmer Menu

In order for the MPLAB PM3 Card options to be available on the MPLAB X IDE programmer menu, the MPLAB PM3 Card must be inserted into the MPLAB PM3 Programmer, the programmer must be powered on and the programmer must be enabled through MPLAB X IDE. If the card is not supported, the MPLAB PM3 Card will not appear in MPLAB X IDE. (See the `Readme for MPLAB PM3` for a list of recommended cards.)

NOTES:

Chapter 7. Troubleshooting First Steps

7.1 INTRODUCTION

If you are having problems with MPLAB PM3 device programmer operation, start here.

- The Questions to Answer First
- Top Reasons Why You Can't Program?
- Other Things to Consider?

7.2 THE QUESTIONS TO ANSWER FIRST

1. Which device are you working with? Often, upgrading to a newer version of MPLAB X IDE is required to support newer devices. That is, yellow light = untested support.
2. Are you using a Microchip demo board or one of your own design? Have you followed the guidelines in Chapter 2 for resistors/capacitors for communications connections? See **Chapter 2. "Using MPLAB PM3 with the MPLAB X IDE"** and **Appendix A. "Hardware Specifications"**.
3. Are you using a USB hub in your setup? Is it powered? If you continue to have problems, try using the programmer without the hub (i.e., plugged directly into the PC).

7.3 COMMON PROBLEMS

Some common problems that you might have could include the following:

- Failure to Establish Communication with Programmer
- Device Pin Damage (see **Section 6.3.2 "Device Pin Damage"**)
- No Programmer Options Visible (see **Section 6.4.2 "Operating System Update Needed"**)
- MPLAB PM3 Card Option Not Available on Programmer Menu

7.3.1 Failure to Establish Communication with Programmer

MPLAB X IDE attempts to establish communication with the programmer when "Enable Programmer" is selected. If communication cannot be established, programming cannot be performed. If the attempt to establish communication fails, it is reported in the Output window.

- General Communications Troubleshooting
- USB Communications

7.3.1.1 GENERAL COMMUNICATIONS TROUBLESHOOTING

1. Check that the USB cable is connected securely between the MPLAB PM3 and the host computer.
2. Check that the power supply is connected and the power LED on the programmer is lit.
3. Check the communications driver listed in the [OPTIONS] section of the SYSTEM.INI file. It should read:

   ```
   COMM.DRV=COMM.DRV
   ```

   If the line is different, you are using a an incompatible communications driver.

7.3.1.2 USB COMMUNICATIONS

Make sure you used the MPLAB X IDE supplied USB driver for MPLAB PM3.

CAUTION

If the Windows OS picked a USB driver, MPLAB PM3 will not work and you will not be able to install the proper driver.

7.3.2 MPLAB PM3 Card Option Not Available on Programmer Menu

In order for the MPLAB PM3 Card options to be available on the MPLAB X IDE Programmer menu, the MPLAB PM3 Card must be inserted in the MPLAB PM3 Programmer, the programmer must be powered on and the programmer must be enabled through MPLAB X IDE. If the card is not supported, the MPLAB PM3 Card will not appear on the Programmer menu. (See the `Readme for MPLAB PM3` for a list of recommended cards.)

7.4 TOP REASONS WHY YOU CAN'T PROGRAM?

1. The oscillator is not working. Check your Configuration bits setting for the oscillator. If you are using an external oscillator, try using an internal oscillator. If you are using an internal PLL, make sure your PLL settings are correct.
2. The target board is not powered. Check the power cable connection.
3. The VDD voltage is outside the specifications for this device. See the device programming specification for details.
4. The programmer has become physically disconnected from the PC and/or the target board. Check the connections of the communication cables.
5. The device is code-protected. Check your Configuration bits settings for code protection.
6. Programmer to PC communications have been interrupted. Reconnect to the programmer in MPLAB X IDE.
7. You have not followed the guidelines in **Chapter 2. "Using MPLAB PM3 with the MPLAB X IDE"** for communication connections.

7.5 OTHER THINGS TO CONSIDER?

1. It is possible that the error was a one-time glitch. Try the operation again.
2. It is possible that the target device has been damaged in some way (e.g., over-current.) Development environments are notoriously hostile to components. Consider trying another target device.
3. Microchip Technology Inc. offers demonstration boards to support most of its microcontrollers. Consider using one of the boards that you have used in the past to verify that the MPLAB PM3 device programmer is functioning correctly.
4. Review programming operation to ensure proper application setup (see **Chapter 2. "Using MPLAB PM3 with the MPLAB X IDE"**).
5. If the problem persists, contact Microchip.

NOTES:

Chapter 8. Frequently Asked Questions (FAQs)

8.1 INTRODUCTION

Look here for answers to frequently asked questions about the MPLAB PM3 device programmer.

- How Does It Work
- What's Wrong

8.2 HOW DOES IT WORK

- What's in the silicon that allows it to communicate with the MPLAB PM3 device programmer?

  MPLAB PM3 device programmer can communicate with Flash silicon via the ICSP interface. It uses the program executive downloaded into program or test memory.
- Is the MPLAB PM3 device programmer optoisolated or electrically isolated?

  No. You cannot apply a floating or high voltage (120V) to the current system.
- Will the MPLAB PM3 slow the running of the program?

  No, the device will run at any device speed that is specified in the data sheet.

8.3 WHAT'S WRONG

- How can I manually download the firmware to my MPLAB PM3?

  You can download it manually. Select *File>Project Properties*. Click on the "MPLAB PM3" category and select "Firmware" from the drop-down Option Categories. Uncheck "Use Latest Firmware" and click on "Press to browse for a specific firmware version". Browse for the Firmware File, located in the Directories pane under MPLABX\mplab_ide\mplablibs\modules\ext\MPLABPM3.jar. In the Firmware Files pane, select the `.mjc` file you want and click **OK**. Click **Reset** on the Project Properties dialog.
- I accidentally disconnected my MPLAB PM3 while firmware was downloading. What do I do now?

  Reconnect the MPLAB PM3. It will begin to erase what had been written so it can restart. This erasing will take about 7 seconds. Please be patient. The LEDs are all lit during this process. When it is done, MPLAB X IDE will recognize the device and start the recovery process, i.e., begin the firmware download.
- My memory window does not reflect changes

  In order to see changes in the window, you must do a read of the memory.
- I don't see my problem here. Now what?

  Try the following resources:
  - **Chapter 9. "Error Messages"**
  - **Chapter 10. "Engineering Technical Notes (ETNs)"**

NOTES:

Chapter 9. Error Messages

9.1 INTRODUCTION

- Error Messages – PC
- Error Messages – LCD
- Limitations

9.2 ERROR MESSAGES – PC

The following error messages could be received via the PC, referring to the MPLAB PM3:

- Could Not Open the Communications Port
- USB Communication Could Not Be Established
- Device is Code-Protected
- Parity Error Found in Program Memory
- Program Memory Contains Code-Protected Data
- Invalid Checksum

9.2.1 Could Not Open the Communications Port

If the communications port cannot be opened, it is either because another application has opened the port, or the port does not exist. Make sure the port number is valid and that no other application is using it, e.g., another instance of MPLAB X IDE.

9.2.2 USB Communication Could Not Be Established

If, while MPLAB PM3 is enabled using USB, the MPLAB PM3 unit is powered off and on too quickly, or if the power is temporarily interrupted; try using these steps to reestablish USB communications:

1. Toggle the MPLAB PM3 power switch off, if it is not already.
2. Toggle the MPLAB PM3 power switch on.
3. Wait 10 seconds for Windows to recognize it.
4. If USB communication is not restored, restart the PC.
5. Finally, try connecting through a different USB root hub.

9.2.3 Device is Code-Protected

This warning is received if you attempt to read a code-protected device, Obtain the original hex code from a file or a non-protected device. Similar to the error described in **Section 9.2.5 "Program Memory Contains Code-Protected Data"**.

9.2.4 Parity Error Found in Program Memory

Several devices have a Configuration bit to request the calculation of parity. When the bit is set to "ON," the contents of the program memory are compared and assigned an "even" or "odd" parity. When the bit is set to "OFF," a comparison is not performed.

On a blank device, the parity Configuration bit is set to "ON" by default. If the device is programmed with the parity Configuration bit set to "ON," any subsequent reprogramming of the device could result in a parity error. This occurs because a mismatch was detected between the current parity and the parity last programmed. However, a parity error will not stop the device from being programmed.

Conversely, if the device is programmed with the parity bit set to "OFF," no comparison of the program memory contents for parity will be performed. And, the device can be reprogrammed without receiving a message regarding parity.

None of the OTP devices, including those that have the parity Configuration bit, can reprogram a bit value of '`0`' to a '`1`' successfully. If an OTP device has been programmed with the parity Configuration bit set to "OFF," the setting cannot be changed to "ON". A blank device will have to be used.

9.2.5 Program Memory Contains Code-Protected Data

Program memory contains data that was read from a code-protected device. Most or all of this data is not usable. Obtain the original hex code from a file or a non-protected device. Similar to the error described in **Section 9.2.3 "Device is Code-Protected"**.

9.2.6 Invalid Checksum

This error indicates a hardware problem occurred while trying to read from the programmer. It is not a very common error. It usually indicates some sort of problem with the communication hardware on either the PC or the programmer.

9.3 ERROR MESSAGES – LCD

9.3.1 BIOS Errors

TABLE 9-1: MPLAB PM3 BEEP CODES

Number of Beeps	Meaning	Corrective Action
1	Normal start up	No action required.
2	CPLD Not Configured	Reset* and connect to MPLAB X IDE.
3	Power Monitor Not Found	Reset* and connect to MPLAB X IDE.
4	Display Not Found	Reset* the unit. If the problem persists, contact your field representative to return the unit.
5	Display Read Write Test Failure	Reset* the unit. If the problem persists, contact your field representative to return the unit.
6	FPGA Time-out	Reset* the unit. If the problem persists, contact your field representative to return the unit.

* Reset the MPLAB PM3 by toggling the power switch off, and then on.

TABLE 9-2: BIOS ERROR CODES

Error Code	Definition	Corrective Action
1	CPLD is Not Configured	Reset and connect to MPLAB X IDE.
2	CPLD is Old	Reset and connect to MPLAB X IDE.
3	Flash is Missing	Reset the unit. If the problem persists, contact your field representative to return the unit.
4	Flash is Crossed Incorrectly	Reset the unit. If the problem persists, contact your field representative to return the unit.
5	FPGA Configuration has Time-out	Reset and connect to MPLAB X IDE.
6	FPGA Image is Missing	Reset and connect to MPLAB X IDE.
7	FPGA is Old	Reset and connect to MPLAB X IDE.
8	Power Monitor is Not Found	Reset and connect to MPLAB X IDE.
9	Power Monitor is Old	Reset and connect to MPLAB X IDE.
10	ICSP Monitor is Not Found	Reset and connect to MPLAB X IDE.
11	ICSP Monitor is Old	Reset and connect to MPLAB X IDE.
12	Failure on the Daughter Board	Reset the unit. If the problem persists, contact your field representative to return the unit
13	USB is Not Found	Reset and connect to MPLAB X IDE.
14	USB is Old	Reset and connect to MPLAB X IDE.
15	Display was Not Found	Reset the unit. If the problem persists, contact your field representative to return the unit
16	Memory Failure	Reset the unit. If the problem persists, contact your field representative to return the unit
17	OS is Not Found	Reset and connect to MPLAB X IDE.
18	VDD is Over Current	If programming using a socket module, verify that the proper part is selected, and the part is inserted in the socket properly.
19	VPP is Over Current	If programming using a socket module, verify that the proper part is selected, and the part is inserted in the socket properly.
20	OS is Old	Reset and connect to MPLAB X IDE.
24	Clock is Missing	Reset the unit. If the problem persists, contact your field representative to return the unit
25	Display Read Write Failure	Reset the unit. If the problem persists, contact your field representative to return the unit
26	Power Monitor Cannot Ramp Up Voltages	Ensure the target is disconnected from the MPLAB PM3 and retry. If it passes, check the ICSP wiring. If it fails again, contact your field representative to return the unit.
27	Updating Failed	Turn unit off. Check connections to PC. Make sure appropriate port is selected and the proper drivers are installed. Restart and retry.

9.3.2 Additional Error Messages

- Hardware needs to be reconfigured. Please attach to PC and connect through MPLAB X IDE to update [error number] – BIOS needs to download a newer version of some firmware.
- Fatal Error: Hardware not found. Contact your local field office for assistance [error number] – Cannot read a piece of hardware.
- Fatal Error: Memory Failure. Contact your local field office for assistance. [error number] – Memory test failed.

9.3.3 LCD Version Screen

On system boot-up, the version screen is displayed.

```
Bios Version..................[ver]
OS Version....................[ver]
Supervisor Version............[ver]
Memory Controller Version.....[ver]
HS Shifter Version............[ver]
Daughter Board Version........[ver]
USB EEPROM Version............[ver]
Database Version..............[ver]
    No Calibration Required!
```

9.4 LIMITATIONS

- General Limitations
- Device-Specific Limitations

9.4.1 General Limitations

- View the “Readme for MPLAB PM3.htm” file for a list of currently supported devices.
- Program memory ranges that are not code protected can only be changed from a 1 to a 0. Most devices need to be erased before they can be reprogrammed. DataEE does not need to be erased before it is programmed; however, it is programmed one location at a time. For more information, consult the specific programming specification for your device.
- Some devices program multiple locations at once. The MPLAB PM3, therefore, inhibits programming ranges not aligned to these bounds. See individual programming specifications for more information.

9.4.2 Device-Specific Limitations

In addition to those general limitations, some devices could have device-specific limitations. Refer to the device family limitations listed below.

9.4.2.1 MPLAB PM3 LIMITATIONS FOR PIC18C601/801

Your own memory read/write routines must be used for these ROM-less devices.

9.4.2.2 MPLAB PM3 LIMITATIONS FOR ALL DEVICES

For ICSP you must connect the AVDD and AVSS pins, in addition to the VDD and VSS pins, for the device to program.

Chapter 10. Engineering Technical Notes (ETNs)

The following ETNs are related to the MPLAB PM3 device programmer. Please go to the www.microchip.com site, the MPLAB PM3 Universal Device Programmer page and click on the ETN in the Documentation & Software section for details.

- ETN #26 MPLAB PM3 ICSP MCLR/VPP Slew Rate Modification: Applies to Assembly # 10-00359-R9 or below without ECO 2937.
- ETN #31 MPLAB PM3 AC164339 Socket Module Modification for dsPIC: Applies to the dsPIC33FJ0GS101-SO device with the AC164339 socket module.

NOTES:

Part 3 – Reference

NOTES:

Chapter 11. Stand-Alone Reference

11.1 INTRODUCTION

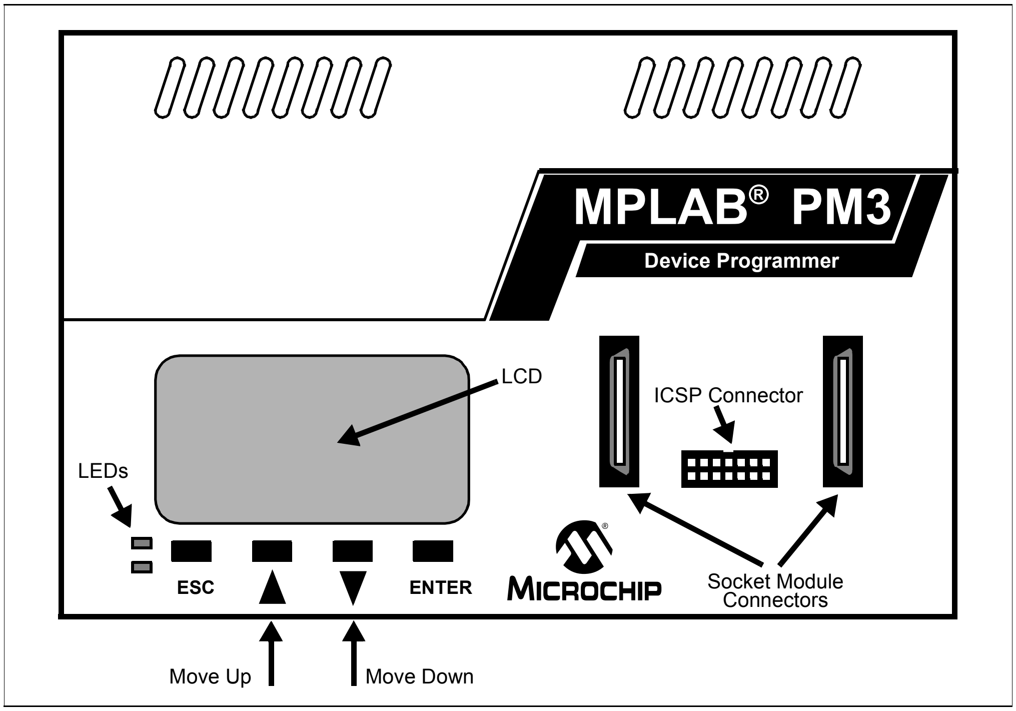
This chapter describes how to use the MPLAB PM3 device programmer in Stand-Alone mode. The device programmer provides an LCD interface that gives you complete control over a programming session. Topics covered are:

- MPLAB PM3 LCD and Keys
- Start-Up Sequence
- Main Menu
- Command Menu

11.2 MPLAB PM3 LCD AND KEYS

The front panel of the programmer looks similar to Figure 11-1.

FIGURE 11-1: MPLAB PM3 FRONT PANEL



11.2.1 LCD

The graphical LCD is a 128 x 64 pixel screen that displays information on a blue background. The selection bar highlights the entire line. When you press <Enter> on a line ending with a right-pointing triangle, a submenu is displayed. When you press <Enter> on a line ending with two greater-than symbols (>>), the action is performed.

11.2.2 Keys

The MPLAB PM3 provides four keys for navigation. The function of each key is:

- <ESC> – return to the previous menu or entry.
- <Up> – moves the selection bar up one line each time it is pressed.
- <Down> – moves the selection bar down one line each time it is pressed.
- <Enter> – performs the selected function.

11.3 START-UP SEQUENCE

When power is applied to the MPLAB PM3 programmer, it initiates a start-up sequence. The system performs a self-check, briefly displays the splash screen, then the versions, followed by the MPLAB PM3 main menu. If you do not get to the Main menu, see **Chapter 6. "Troubleshooting"**.

> **Note:** If you want to update the operating system, you need to be connected to a PC and open MPLAB.

Once the start-up sequence has completed, you will see the Main menu.

11.4 MAIN MENU

The Main menu gives you these options (when an SD/MMC card is not inserted):

- Recently Used
- Select Device
- (Stand-Alone) Programmer Settings
- Help

11.4.1 Recently Used

To display the most recently used devices:

1. Select (highlight) the "Recently Used" option and press <Enter>. Up to seven of the most recently-used devices are listed.
2. To continue using this device, press <Enter>.
3. The system displays the Command menu. See **Section 11.5 "Command Menu"**.

11.4.2 Select Device

To select a specific device, you must:

1. Select (highlight) the "Select Device" option and press <Enter>.
2. Select the device family and press <Enter>.
3. Select the device and press <Enter>.
4. The system displays the Command menu. See **Section 11.5 "Command Menu"**.

11.4.3 (Stand-Alone) Programmer Settings

From the Programmer Settings option on the Main menu, you can access:

- Screen Contrast
- Buzzer Volume
- Socket Information
- Checksum Calculation
- Device ID Option
- Blank Check Override
- $\overline{\text{GO}}$ Pin Functionality

11.4.3.1 SCREEN CONTRAST

This option enables you to select the default, darker or lighter setting to adjust the contrast on your screen. Sample text lines are provided so you can see the result of your selection.

- Normal (default) – to reset the contrast to the default settings, select “Default” and press <Enter> once.
- Darker – to increase the contrast, select “Darker” and press <Enter> until you achieve the desired contrast.
- Lighter – to reduce the contrast, select “Lighter” and press <Enter> until you achieve the desired contrast.

11.4.3.2 BUZZER VOLUME

This options enables you to select the volume of the buzzer tone.

- Loud – to make the buzzer volume loud, select “Loud” and press <Enter>.
- Medium (default) – to return to the default volume, select “Default” and press <Enter>.
- Low – to make the buzzer volume low, select “Low” and press <Enter>.
- Mute – to mute the buzzer volume, select “Mute” and press <Enter>.

11.4.3.3 SOCKET INFORMATION

This option displays the socket part number and the socket ID of the socket module for the socket currently installed on the MPLAB PM3.

11.4.3.4 CHECKSUM CALCULATION

With this option, you can enable or disable the calculation of a checksum for a code-protected device. When you select this menu option from the Main menu, a message appears that asks whether you want to calculate the checksum, states the current setting, and offers two options (see Figure 11-2).

FIGURE 11-2: CHECKSUM CALCULATION SCREEN

```
                 Checksum Calculation
---------------------------------------------------------
Would you like to calculate the Code Protected Checksum
Currently: No


Do Not Calculate                                       >>
Calculate Checksum                                     >>
```

The default setting is “Do Not Calculate”. The setting is saved when the programmer is powered off.

Selecting the “Calculate Checksum” option saves time when programming or verifying larger code-protected devices such as the PIC18F8720 since the entire device does not get re-read.

If you set the checksum calculation option to “Do Not Calculate” and successfully program a code-protected device, the screen will display, “Passed! Protected csum disabled”.

If you set the checksum calculation option to “Do Not Calculate” and the program operation of a code-protected device fails, the screen will display, “Failed! CP sum disabled”.

If you set the checksum calculation option to “Calculate Checksum” for a code-protected device, the checksum will be displayed with the results of a program or verify operation.

11.4.3.5 DEVICE ID OPTION

This option checks that the selected device is inserted into the socket module. When you select this menu option, a message appears asking if you want to check device IDs, states the current setting and offers two options (see Figure 11-3).

FIGURE 11-3: DEVICE ID OPTION SCREEN

```
                 Device Id Option
---------------------------------------------------------
Would you like to check device ids?
Currently: No


Test Device Ids                                        >>
Don’t Test Device Ids                                  >>
```

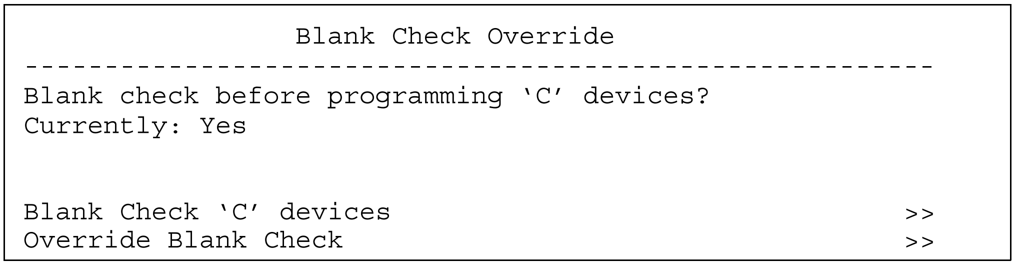
The default setting is “Don’t Test Device Ids” (“Currently: No”). The setting is saved when the programmer is powered off.

Selecting the “Test Device Ids” option verifies that the device selected through the MPLAB PM3 matches the device installed in the socket module. This is helpful in avoiding programming the wrong device. The “Currently:” display changes to **Yes** when the “Test Device Ids” option is set.

11.4.3.6 BLANK CHECK OVERRIDE

Selecting this option disables the blank check from occurring before programming an EEPROM or ‘C’ device. When you select this menu option, a message appears asking if you want to override the blank check, states the current setting and offers two options (see Figure 11-4).

FIGURE 11-4: BLANK CHECK OVERRIDE SCREEN



```
                 Blank Check Override
---------------------------------------------------------
Blank check before programming ‘C’ devices?
Currently: Yes


Blank Check ‘C’ devices                                >>
Override Blank Check                                   >>
```

The default setting is “Blank Check ‘C’ devices” (“Currently: Yes”). The setting is saved when the programmer is powered off.

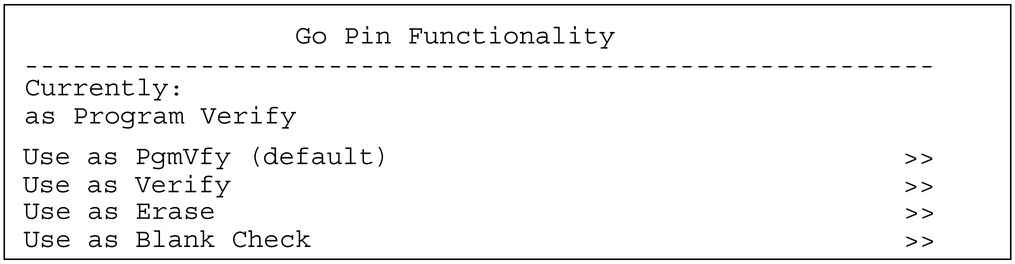
Selecting the “Override Blank Check” enables programming C devices without first checking that they are blank. The “Currently:” display changes to **No** when the “Override Blank Check” option is set.

This setting is used if the C device is preprogrammed, for example, with oscillator values. In that case, if you performed a blank check, the device would fail because the MPLAB PM3 would detect the preprogrammed settings. Using the override setting is helpful in those situations.

11.4.3.7 $\overline{\text{GO}}$ PIN FUNCTIONALITY

This option allows the user to change the functionality of the $\overline{\text{GO}}$ pin. This pin is active low. When you select this menu option, a message appears with the current setting and offers four options (see Figure 11-4).

FIGURE 11-5: GO PIN FUNCTIONALITY SCREEN



```
                 Go Pin Functionality
---------------------------------------------------------
Currently:
as Program Verify

Use as PgmVfy (default)                                >>
Use as Verify                                          >>
Use as Erase                                           >>
Use as Blank Check                                     >>
```

The default setting is “Use as PgmVfy” (“Currently: as Program Verify”). The setting is saved when the programmer is powered off.

This feature is especially useful on the production line in situations where fast device processing is required. It enables repeated operations to be performed on devices as they are automatically loaded/unloaded into the programmer. Each time the device is loaded, the $\overline{\text{GO}}$ pin is activated and the preset operation is performed.

Option	Action
Use as PgmVfy	Programs and verifies device
Use as Verify	Verifies device
Use as Erase	Erases device
Use as Blank Check	Blank checks device

The “Currently:” display changes to reflect the selected option.

11.4.4 Help

This option displays the version number of the MPLAB PM3 OS Suite running on the MPLAB PM3. It also contains these submenu options:

- ICSP Connector Pinout – displays a diagram of the ICSP connector and offers an option to view the pin definitions. Definitions for all the ICSP pins are listed with the number, color, and signal. See **Section A.3.1 "ICSP Header and Cable Pinout"**.
- Status Bar Icons – display these icons and definitions:
  - I – ICSP Mode
  - T – Power is supplied by Target
  - V – LVP Program Entry
  - Q – SQTP file is loaded
  - O – Options Set
  - R – Ranges/Regions Set
- About – displays the name of the product and manufacturer, and has an option to view the versions. The versions for MPLAB PM3-related programs are listed.

11.5 COMMAND MENU

After a device is selected, the device programmer displays the Command menu:

- Program Verify Device
- Verify Device
- All Functions

The functions on the Command menu allow you to perform the basic user tasks for programming a microcontroller device. Pressing <ESC> repeatedly will return you to the Main menu.

11.5.1 Program Verify Device

The “Program Verify Device” option erases (if it is a Flash device), programs, and verifies the device.

11.5.2 Verify Device

The “Verify Device” option verifies that the device was programmed properly.

11.5.3 All Functions

The “All Functions” option displays a submenu with the selections listed in Table 11-1.

TABLE 11-1: ALL FUNCTIONS MENU

Selection	Description
Read Device	reads device content Selective read (e.g., part of program memory, only configuration bits of the device) can be done by setting up Settings (on the last row of this table).
Program Verify Device	programs and verifies the device
Verify Device	verifies that the device was programmed properly
Blank Check Device	checks that the device is completely blank (i.e., all bits are set to a ‘1’) This also checks that all configuration bits are set to a ‘1’ (unprogrammed state).
Erase Device	erases the device in the MPLAB PM3 device programmer This option displays for erasable devices, only.
Device Information	displays the device name, Device ID, Device Revision, Last Checksum and the User IDs
Settings	sets up program/verify options, ICSP settings, adjust voltages, reset voltages, reset statistics count and safe mode

Read Device

If you want to copy firmware from a programmed device into an unprogrammed device, you can read the programmed firmware into MPLAB PM3 memory, then program the new device based on this information.

Select “Read Device” to copy the contents of the device in the socket module into the internal memory of the device programmer.

For PIC16CXX devices, the device programmer will ask the question: “Code Protect Parts?” Answer “Yes” to code protect the parts that you will be programming. Code protection will remain effective until a new device is read. Answer “No” to program devices exactly as read.

After reading a device, the device programmer displays a checksum. If the device is code protected, a code protection message will be displayed prior to the read. Answer “Yes” to continue. Answer “No” to return to the Command menu.

Program Verify Device

To program a device, select “Program Verify Device” and press <Enter>. This command option programs the device (that is in the socket module) with the contents of the internal memory of the device programmer (image).

For non-Flash devices, the device programmer checks to see if the installed microcontroller device is blank. If the device is not blank, the device programmer asks whether you want to continue. Answer “Yes” to continue. Answer “No” to return to the Command menu.

The device programmer programs the contents of its memory into the microcontroller device loaded in the socket module. If a hex file is not loaded into the device programmer, an event message is displayed, “Valid image not present! Please download an image or use a master device before continuing.” You must have an image in the device programmer in order to program a device. To load a hex file into the MPLAB PM3, you must connect the device programmer to a PC or read from a master device. See **Section 4.2.2 “Downloading a Hex File into MPLAB PM3 Memory”**.

After programming a device without errors, the device programmer performs a check to verify that the data programmed into the device and returns the results of the verification. For the installed device, the device programmer performs the verification at the VDD Minimum and VDD Maximum voltages for non-Flash devices or VDD Nominal for Flash devices.

The device programmer reports programming errors and verify errors according to where the error occurred. Errors are reported for program, Configuration bits, EEPROM data memory, and ID locations. Only the first location to fail in each region is displayed. The address, expected value, and read value are displayed.

After programming, the device programmer displays the checksum.

Verify Device

To verify a device, select “Verify Device” and press <Enter>. The device programmer compares the contents of its internal memory to the contents of the programmed microcontroller device loaded in the socket module. If the data and Configuration bit settings are correct, “PASSED!” and the checksum will display on the LCD. The device programmer performs the verification at the VDD Minimum and VDD Maximum voltages for non-Flash devices, and VDD Nominal for Flash devices. The device programmer reports errors according to which part of the device failed. Only the first location to fail in each region is displayed. The address, expected value, and read value are displayed.

Blank Check Device

To perform a blank check, select “Blank Check Device” and press <Enter>. This check will verify that the device is completely blank (all bits are set to a ‘1’) and all Configuration bits are set to a ‘1’ (unprogrammed state). When completed, the programmer LCD displays “Device Blank.” Only the first location to fail in each region is displayed. The address, expected value, and read value are displayed.

An OTP device cannot be erased and reprogrammed.

If an EPROM device is not blank, erase it before programming, or choose another device:

1. Remove any labels covering the device window. If you do not have a windowed device, you cannot reprogram it. A windowed version of any EPROM device can be ordered by requesting the JW package.
2. Place the device in an Ultraviolet (UV) EPROM Eraser. The amount of time required to completely erase a UV erasable device depends on the following cases:
   a) the wavelength
   b) the intensity of the light
   c) the distance from UV source
   d) the process technology of the device (how small the memory cells are)
3. Before attempting to program the device, perform the blank check again to verify that it is blank.

> **Note:** If the device is EEPROM or Flash, you do not have to erase it before reprogramming it. These devices are electrically erased before programming.

Erase Device

To erase a Flash device, select *All Functions>Erase Device* and press <Enter>. Press <Enter> a second time to erase the device. When completed, the LCD displays “Done”.

Device Information

To display device information, select *All Functions>Device Information* and press <Enter>. The system displays the device ID, device revision, last checksum and user ID. It also contains an option for “Check Code Protect”. Select this option to determine whether the device is code protected and press <Enter>.

Settings

To display the “Settings” option, select *All Functions>Settings* and press <Enter>. The “Settings” option contains a submenu with the following options for the selected device:

- Pgm/Vfy Settings
- ICSP Settings
- Adjust Voltages
- Reset Voltages
- Reset Statistics Count
- Safe Mode

11.5.3.1 PGM/VFY SETTINGS

- Memory Ranges – displays the memory ranges selected for the device, as retrieved from an environment.
- Select Regions – toggles the Yes/No settings for programming Program Memory, EEPROM, IDs, and Configuration bits.
- Program Options – toggles the Yes/No settings for the program options to "Erase before Pgm" and "Preserve EEPROM".
- Reset to Defaults – restores the default ranges, regions, and options settings for the selected device.

11.5.3.2 ICSP SETTINGS

These options are available only if the selected device supports ICSP.

- ICSP Power Source – the MPLAB PM3 displays the current ICSP power source and provides two options:
  - Power from MPLAB PM3 – programming the device will be powered through the MPLAB PM3.
  - Target has own supply – programming the device will be powered from the target's power supply.
- ICSP Programming Mode – the MPLAB PM3 displays the current ICSP Program mode voltage criteria and provides two options:
  - $\overline{\text{MCLR}}$ High Voltage – utilizes $\overline{\text{MCLR}}$ high voltage for programming the device.
  - PGM Pin (LVP) – utilizes the low voltage programming pin for programming the device.

11.5.3.3 ADJUST VOLTAGES

This option enables you to adjust the V_{DD} nominal, V_{DD} App, and V_{PP} for a Flash device; and V_{DD} Max, V_{DD} Min, V_{DD} Nominal, and V_{PP} for an EEPROM device.

Each option displays the current setting and enables you to select "Up" or "Down". Each time you press <Enter> the voltage setting is adjusted up or down by 0.125V.

11.5.3.4 RESET VOLTAGES

To reset the voltages, select *All Functions>Settings>Reset Voltages* and press <Enter>. This command resets the voltages to the default voltage settings.

11.5.3.5 RESET STATISTICS COUNT

To reset all of the program statistics, select *All Functions>Settings>Reset Statistics Count* and press <Enter>. This command will reset the counters to zero.

11.5.3.6 SAFE MODE

Safe mode was designed to prevent operational errors during production. When in Safe mode, the only function that is allowed is Program/Verify. If any other button is pressed, a message is displayed stating that the system is in Safe mode and the operation is not allowed.

Note: Requesting other functions in MPLAB X IDE when the MPLAB PM3 is connected to the PC will deactivate Safe mode.

To enter Safe Mode, select *All Functions>Settings>Safe Mode* and press <Enter>. In Safe mode, only Program/Verify options are available on the Command menu. When safe mode is implemented, you will be sent to the Command menu.

Chapter 12. MPLAB PM3 Card Reference

12.1 INTRODUCTION

This chapter explains how to use the MPLAB PM3 card in the MPLAB PM3 device programmer. The MPLAB PM3 card can be accessed through the MPLAB IPE software or directly through the MPLAB PM3 programmer in Stand-Alone mode.

Only those commands specifically for the MPLAB PM3 card are covered here. For other MPLAB IPE commands, refer to IPE online help. For other Stand-Alone mode commands, refer to **Chapter 11. "Stand-Alone Reference"**.

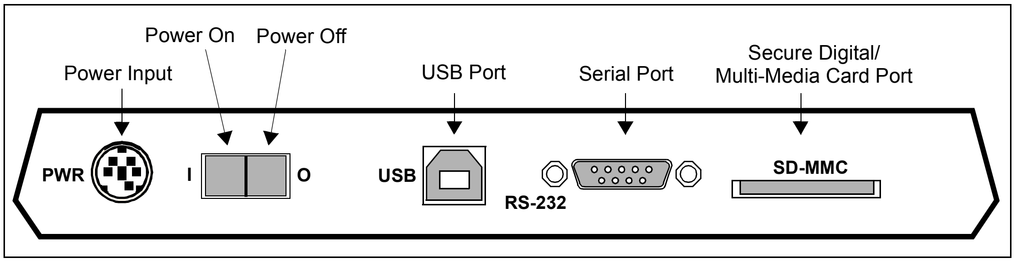
Topics covered in this chapter include:

- Inserting the MPLAB PM3 Card
- MPLAB PM3 Card in Stand-alone Mode
- MPLAB PM3 Card Using MPLAB IPE

12.2 INSERTING THE MPLAB PM3 CARD

The MPLAB PM3 card is a memory device that fits in the SD-MMC port on the back of the MPLAB PM3 programmer (see Figure 12-1). The card is oriented into the slot one particular way. It can be used to store Environments that can be transferred to and from the MPLAB PM3 programmer.

FIGURE 12-1: BACK VIEW OF MPLAB PM3



For a list of Microchip tested and recommended cards, refer to the **"Release Notes for MPLAB PM3"**.

When the card is inserted into the SD-MMC port, MPLAB PM3 card options become available. See the next sections in this chapter for information about using these options in Stand-Alone mode or with the MPLAB IPE.

12.3 MPLAB PM3 CARD IN STAND-ALONE MODE

There are two ways to access the MPLAB PM3 card in the MPLAB PM3 programmer when it is in Stand-Alone mode. Access the card through one of the following menus:

- MPLAB PM3 Card Menu from the main menu
- All Functions Menu after a device is selected

12.3.1 MPLAB PM3 Card Menu

In Stand-Alone mode, with the MPLAB PM3 card inserted, the Main menu has several options:

- Recently Used
  see **Section 11.4.1 "Recently Used"**
- Select Device
  see **Section 11.4.2 "Select Device"**)
- MPLAB PM3 card
- (Stand-Alone) Programmer Settings
  see **Section 11.4.3 "(Stand-Alone) Programmer Settings"**
- Help
  see **Section 11.4.4 "Help"**

After you select the "MPLAB PM3 card" from the Main menu, the device programmer displays the "MPLAB PM3 card" Command menu:

- Load an Environment
- View an Environment
- View Disk Contents
- Open a Text File
- Card Properties

The functions on the Command menu allow you to perform the basic user tasks for programming a microcontroller device. Pressing <ESC> repeatedly will return you to the Main menu.

12.3.1.1 LOAD AN ENVIRONMENT

The "Load an Environment" option loads the environment into the MPLAB PM3 programmer from the MPLAB PM3 card. The environment includes the associated BIN file, the memory, voltage, and ICSP settings and the optional SQTP file.

> **Note:** Program ranges, erase options, and memory region settings are not used in Stand-Alone mode.

12.3.1.2 VIEW AN ENVIRONMENT

The "View an Environment" option displays the contents and settings of an environment. You can only view the environment using this option.

12.3.1.3 VIEW DISK CONTENTS

The "View Disk Contents" option displays the contents of the MPLAB PM3 card inserted in the MPLAB PM3 programmer. Directories (designated with a folder icon) can be expanded further by pressing <ENTER>.

12.3.1.4 OPEN A TEXT FILE

The "Open a Text File" (*.txt) option displays the first couple hundred characters of a text file.

12.3.1.5 CARD PROPERTIES

Displays the card’s byte capacity, bytes free, bytes used, and cluster size.

12.3.2 All Functions Menu

Once a device is selected, the device’s menu is displayed:

- Program Verify Device (see **Section 11.5.1 “Program Verify Device”**)
- Verify Device (see **Section 11.5.2 “Verify Device”**)
- All Functions

When the MPLAB PM3 card is inserted, the “All Functions” option displays a submenu with the selections listed in Table 12-1. Note that there is an additional option for the MPLAB PM3 card at the end of the list. For all other functions listed, refer to **Section 11.5.3 “All Functions”**.

TABLE 12-1: ALL FUNCTIONS MENU

Selection	Description
Read Device	reads the device content A selective read (e.g., part of program memory, only Configuration bits of the device) can be done by setting up Settings (in the last row of this table).
Program Verify Device	programs and verifies the device
Verify Device	verifies that the device was programmed properly
Blank Check Device	checks that the device is completely blank (all bits are set to a ‘1’) This function will also check that all configuration bits are set to a ‘1’ (unprogrammed state).
Erase Device	erases the device in the MPLAB PM3 device programmer This option displays only for erasable devices.
Device Information	displays the device name, Device ID, Device Revision, Last Checksum and the User IDs
MPLAB PM3 Card	displays a submenu of operations that pertain to the MPLAB PM3 card
Settings	sets up program/verify options, ICSP settings, adjust voltages, reset voltages, reset statistics count and safe mode

The MPLAB PM3 card function displays these options:

TABLE 12-2: MPLAB PM3 CARD FUNCTIONS

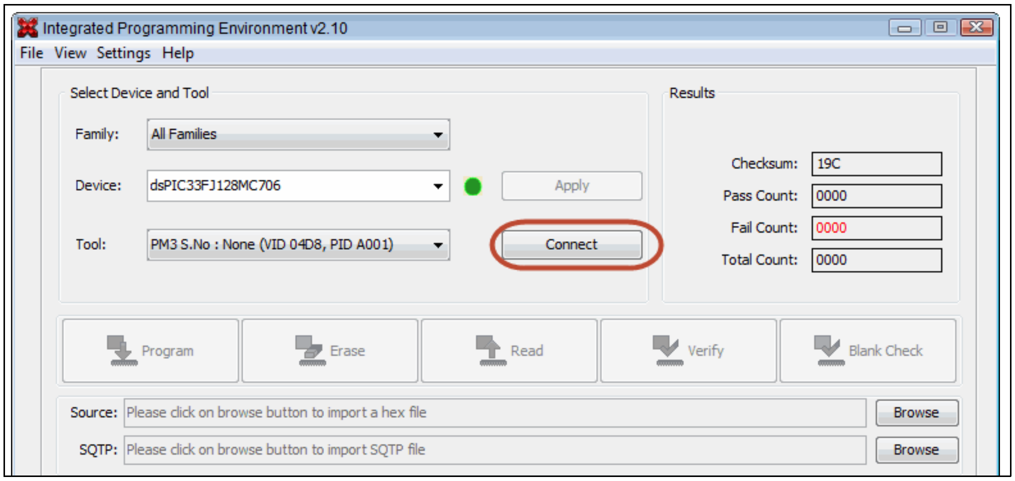
Load an Environment	loads the environment into the MPLAB PM3 programmer from the MPLAB PM3 card
View an Environment	displays the contents and settings of an environment You can only view the Environment using this option.
Card Properties	displays the card’s byte capacity, bytes free, bytes used, and byte cluster.

12.4 MPLAB PM3 CARD USING MPLAB IPE

To use the MPLAB PM3 card with the MPLAB IPE:

1. Launch the MPLAB IPE application.
2. Connect the USB cable between the MPLAB PM3 and the PC.
3. Insert the card into the MPLAB PM3 (if not already done).
4. Click the **Connect** button next to the PM3 tool (see Figure 12-2).

FIGURE 12-2: CONNECT TOOL TO MPLAB IPE



Integrated Programming Environment v2.10

Select Device and Tool

Family: All Families

Device: dsPIC33FJ128MC706

Tool: PM3 S.No : None (VID 04D8, PID A001)

Results

Checksum: 19C

Pass Count: 0000

Fail Count: 0000

Total Count: 0000

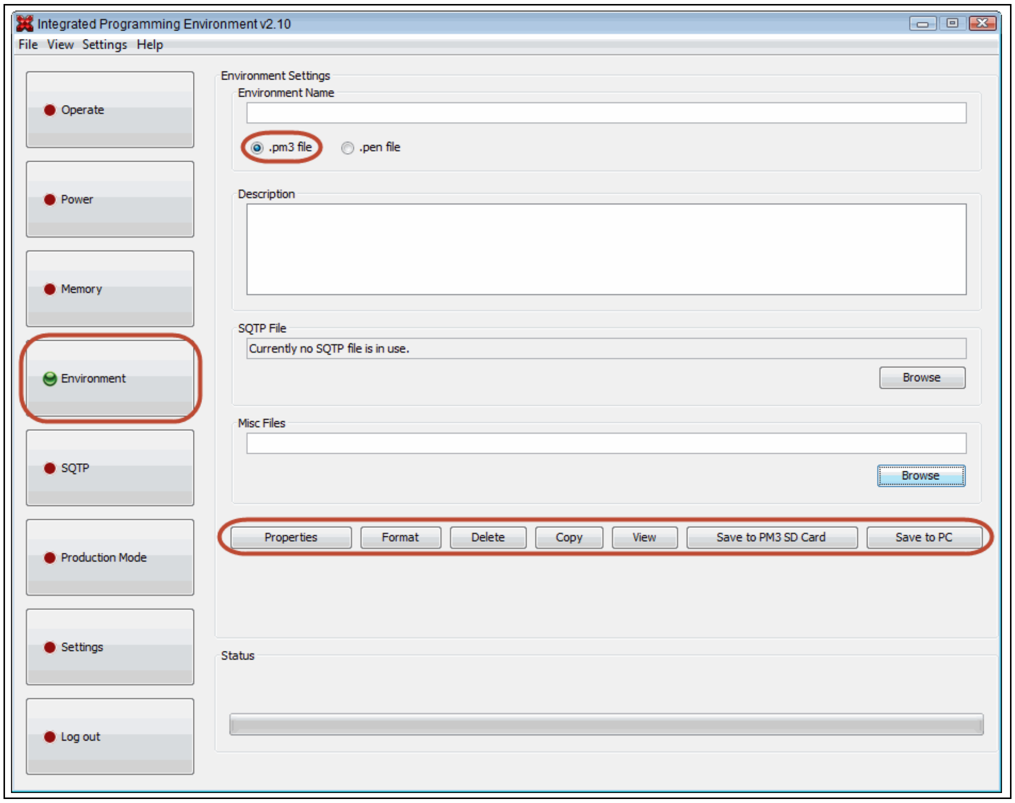
Source: Please click on browse button to import a hex file

SQTP: Please click on browse button to import SQTP file

5. Select *Settings>Advanced Mode* and type the password to log on to Advanced Mode. Refer to the MPLAB IPE online help for more information on Advanced Mode settings.

6. Click the **Environment** button to open the Environment Settings window. Refer to Figure 12-3 and the MPLAB IPE online help for information on all the environment settings.

FIGURE 12-3: ENVIRONMENT WINDOW

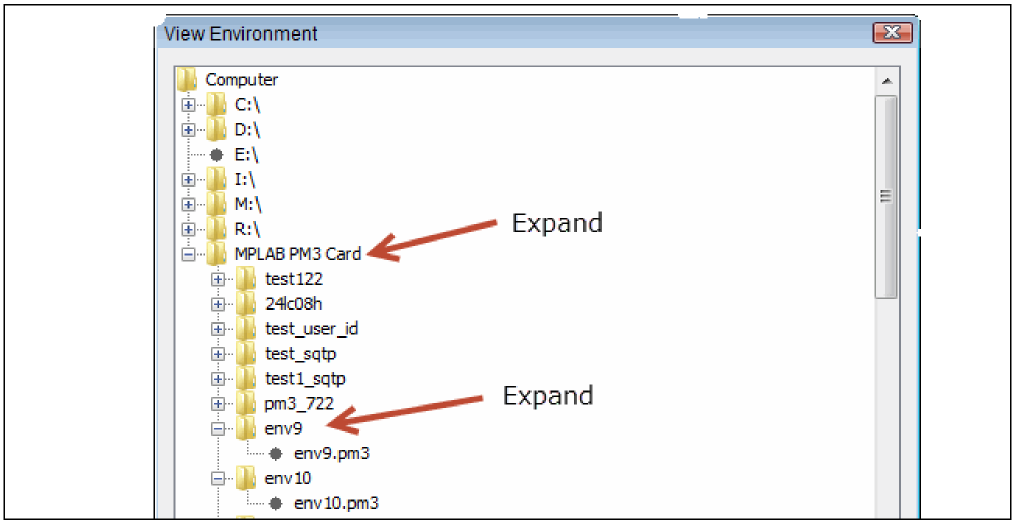


● .pm3 file ○ .pen file

SQTP File: Currently no SQTP file is in use.

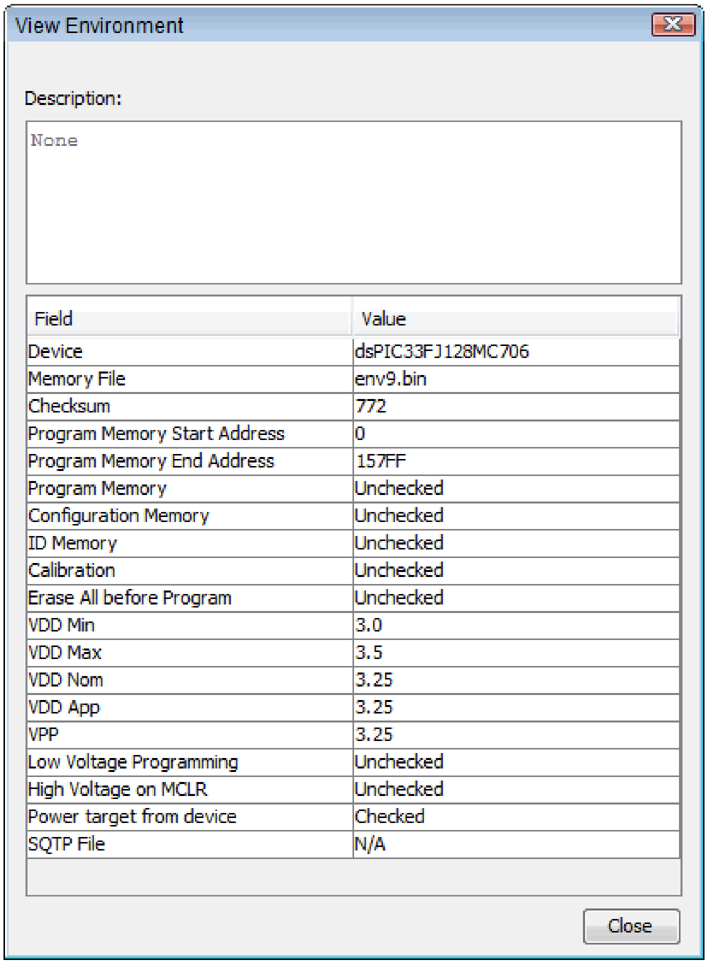
7. To view the files on the card, select **View** to open the View Environment window. Expand the MPLAB PM3 Card to view the files on the card.

FIGURE 12-4: VIEW WINDOW



8. Expand folders to locate the .pm3 files. Select the file you want and click **View** to display the file details. See Figure 12-5 for an example.

FIGURE 12-5: FILE DETAILS



Description:

None

Field	Value
Device	dsPIC33FJ128MC706
Memory File	env9.bin
Checksum	772
Program Memory Start Address	0
Program Memory End Address	157FF
Program Memory	Unchecked
Configuration Memory	Unchecked
ID Memory	Unchecked
Calibration	Unchecked
Erase All before Program	Unchecked
VDD Min	3.0
VDD Max	3.5
VDD Nom	3.25
VDD App	3.25
VPP	3.25
Low Voltage Programming	Unchecked
High Voltage on MCLR	Unchecked
Power target from device	Checked
SQTP File	N/A

Appendix A. Hardware Specifications

A.1 INTRODUCTION

This appendix describes MPLAB PM3 to PC connections, programmer hardware, and socket module hardware. Contents in this appendix include:

- Programmer Specifications
- ICSP Hardware Specifications
- Socket Module Specifications
- ICSP Cable Adapter

A.2 PROGRAMMER SPECIFICATIONS

This section discusses the following topics:

- Physical Dimensions and Layout
- General Specifications

Note:	Values are specified under normal operation and represent recommended operating conditions. Absolute maximum ratings have not been published at this time.

A.2.1 Physical Dimensions and Layout

The physical dimensions of MPLAB PM3 device programmer are the following:

8 3/8″ L x 6 5/16″ W x 13/16″ H. Top and back views of the programmer are shown in Figure A-1 and Figure A-2, respectively.

FIGURE A-1: MPLAB PM3 TOP VIEW

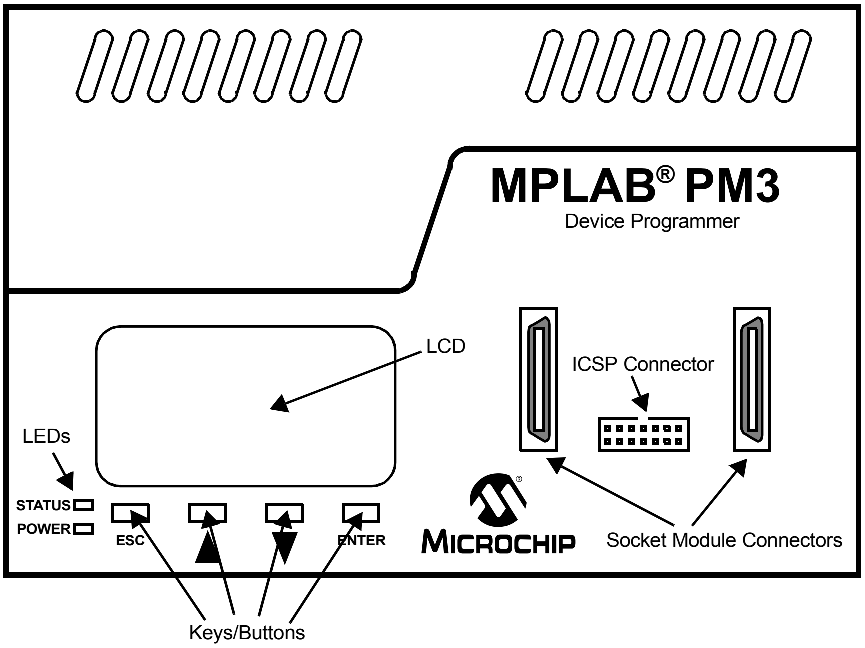
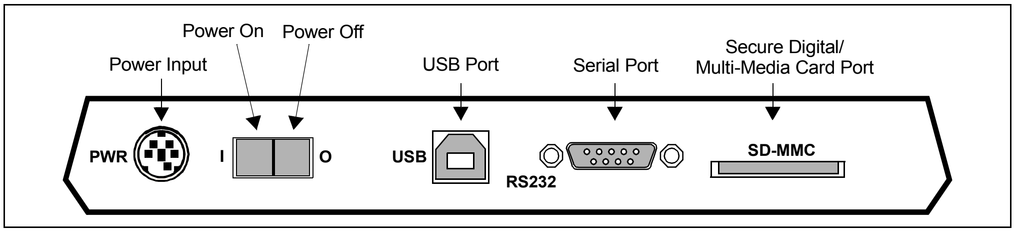


FIGURE A-2: MPLAB PM3 BACK VIEW



A.2.2 General Specifications

TABLE A-1: GENERAL SPECIFICATIONS

Component	Description
Power Supply	20.3W Dual Output +3.3VDC/5A, +5.0VDC/0.75A, w/ 5-pin Mini Din plug
USB Cable	USB A-B M-M cable, 6 feet
Serial Cable	RS-232 DB9 (1 end male, 1 end female)
ICSP Cable	18″ 15-pin, 22 AWG Super Flex Stranded Series 200 wire leads with 2 x 7 keyed housing
Pin Drivers	40 Universal pin drivers capable of supplying VDD, VPP, I/O, Ground
Diagnostics	Power supply, CPUs, FPGA, CPLD

A.3 ICSP HARDWARE SPECIFICATIONS

This section covers:

- ICSP Header and Cable Pinout
- ICSP Pin Capacitance Values
- ICSP VDD/VPP Operating Characteristics
- MPLAB PM3 Handler Operating Characteristics

A.3.1 ICSP Header and Cable Pinout

MPLAB PM3 comes equipped with an ICSP pin header and cable. You can locate this under the place where a socket module would be installed. The header configuration can be found in Figure A-3 (viewed from above on the MPLAB PM3) and the cable color and signal designations are listed in Table A-2.

FIGURE A-3: ICSP HEADER CONFIGURATION

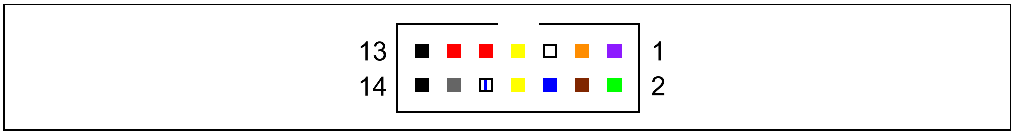


TABLE A-2: ICSP CABLE PINOUT

Pin	Cable Color		Signal
1	Violet	🟪	CLK
2	Green	🟩	PASS
3	Orange	🟧	DATA
4	Brown	🟫	FAIL
5	White	⬜	LVP/Clock
6	Blue	🟦	$\overline{\text{GO}}$
7	Yellow	🟨	$\overline{\text{MCLR}}/V_{PP}$
8	Yellow	🟨	$\overline{\text{MCLR}}/V_{PP}$
9	Red	🟥	V_{DD}
10	White with blue stripe	⬜🟦	Test
11	Red	🟥	V_{DD}
12	Gray	▨	+5V
13	Black	⬛	GND
14	Black	⬛	GND

Information on using ICSP for a specific device is described in the device's programming specification. You can find programming specifications on the Microchip Technology web site, www.microchip.com.

Note: Because the MPLAB PM3 has ICSP built-in, you cannot use the following items:

- PRO MATE II ICSP Socket Module (AC04004)
- MPLAB PM3-to-PRO MATE II device adapter

A.3.2 ICSP VDD/VPP Operating Characteristics

TABLE A-3: POWER VIA PM3

Parameter	Value
VPP Range	9-14V in 125 mV steps
IPP Range	150 mA max
VDD Range	2-5.5V in 125 mV steps
IDD Range	500 mA max
Protection	Over Voltage Shutdown

TABLE A-4: POWER VIA TARGET

Parameter	Value
Target VDD Range	2-5.5V
VPP Range	9-14V in 125 mV steps
IPP	-150 mA max
Target VDD Range	2-5.5V
Protection	IPP Overcurrent Fuse, Over Voltage Shutdown

> **Note:** The use of any cable other than the one that was provided with the MPLAB PM3 unit (other material, length, etc.) could result in unreliable device programming.

ICSP I/O OPERATING CHARACTERISTICS

Parameter	Conditions	VDD	Min	Max
VDD	Supply Voltage Range		2.00	5.50
VIH	Minimum High-level Input Voltage	2.0	1.50	
		3.0	2.10	
		5.5	3.85	
VIL	Maximum Low-level Input Voltage	2.0		0.50
		3.0		0.90
		5.5		1.65
VOH	Minimum High-level Output Voltage @ **IOH** = -50 μA	2.0	1.90	
		3.0	2.90	
		4.5	4.40	
VOL	Maximum Low-level Output Voltage @ **IOL** = 50 μA	2.0		0.10
		3.0		0.10
		4.5		0.10
IOH	High-level Output Current	2.00		-50 μA
		3.3		-4 mA
		5.0		-8 mA
IOL	Low-level Output Current	2.00		-50 μA
		3.3		-4 mA
		5.0		-8 mA
tR, tF	Rise/Fall Times			50 nS
tPD	Propagation Delay Cl = 50 pF		1 nS	5.5 nS

A.3.3 MPLAB PM3 Handler Operating Characteristics

TABLE A-5: PASS/FAIL SIGNALS

Parameter	Conditions	V_{CC}	Min	Max
V_{OH}	Minimum High-level Output Voltage @ I_{OH} = -5.2 mA	5.0	4.50	
V_{OL}	Maximum Low-level Output Voltage @ I_{OL} = 5.2 mA	5.0		0.26
I_{OH}	High-level Output Current	5.0		-20 mA
I_{OL}	Low-level Output Current	5.0		20 mA

TABLE A-6: $\overline{\text{GO}}$ SIGNAL

Parameter	Conditions	V_{CC}	Min	Max
V_{IH}	Minimum High-level Input Voltage	5.0	3.50	
V_{IL}	Maximum Low-level Input Voltage	5.0		1.50

Note: The $\overline{\text{GO}}$ signal has internal 10K pull-up to +5.0V and a 0.1 μF cap to GND.

A.4 SOCKET MODULE SPECIFICATIONS

This section discusses the following topics:

- Socket Specifications
- MPLAB PM3 Socket Module Insertion
- Compatibility with MPLAB PRO MATE II Socket Modules
- Socket Life Expectancy and Cleaning Method

A.4.1 Socket Specifications

TABLE A-7: SOCKET VDD/VPP OPERATING CHARACTERISTICS

Parameter	Value
VPP Range	9-14V in 125 mV steps
IPP Range	150 mA max
VDD Range	2-5.5V in 125 mV steps
IDD Range	500 mA max
Protection	Overcurrent Shutdown

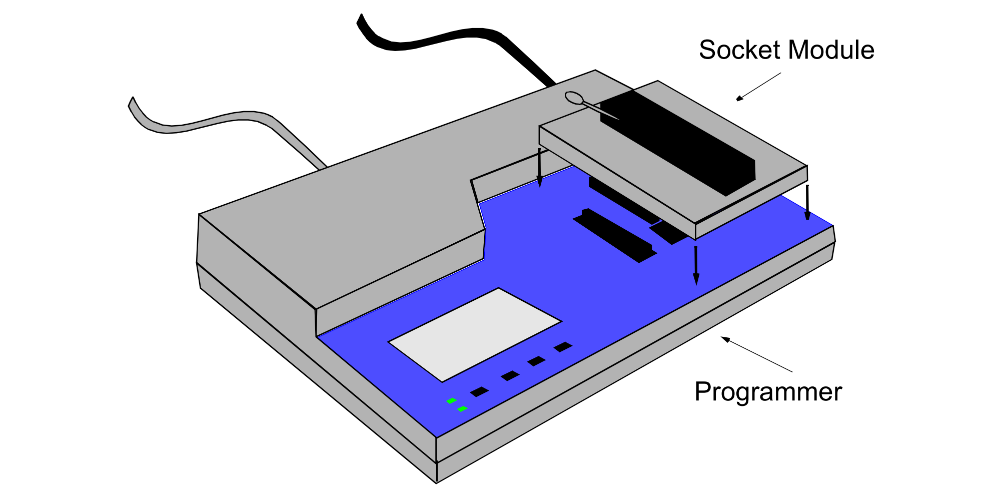
TABLE A-8: SOCKET I/O OPERATING CHARACTERISTICS

Parameter	Conditions	Vcc	Min	Max
VCC	Supply Voltage Range		2.00	5.50
VIH	Minimum High-level Input Voltage	2.0	1.50	
		3.0	2.10	
		4.5	3.15	
		5.5	3.85	
VIL	Maximum Low-level Input Voltage	2.0		0.50
		3.0		0.90
		4.5		1.35
		5.5		1.65
VOH	Minimum High-level Output Voltage @ IOH = -50 µA	2.0	1.90	
		3.0	2.90	
VOL	Maximum Low-level Output Voltage @ IOL = 50 µA	2.0		0.10
		3.0		0.10
		4.5		0.10
IOH	High-level Output Current	1.65		-4 mA
		2.3		-8 mA
		3.0		-24 mA
		4.5		-32 mA
IOL	Low-level Output Current	1.65		4 mA
		2.3		8 mA
		3.0		24 mA
		4.5		32 mA

A.4.2 MPLAB PM3 Socket Module Insertion

The MPLAB PM3 utilizes socket modules to accommodate various Microchip devices. MPLAB PM3 socket modules have two interface connectors on the programmer. (Figure A-4).

FIGURE A-4: SOCKET MODULE ALIGNMENT



A.4.3 Compatibility with MPLAB PRO MATE II Socket Modules

MPLAB PM3 can accept PRO MATE II socket modules with the use of an adapter module.

CAUTION

Do not attempt to connect the PRO MATE II ICSP socket module to the MPLAB PM3.

A.4.4 Socket Life Expectancy and Cleaning Method

Microchip uses socket modules from several manufacturers. Table A-9 gives the expected life (in number of automatic insertions) and cleaning method for each socket module as reported by the manufacturer.

TABLE A-9: SOCKET LIFE EXPECTANCY AND CLEANING METHOD

Manufacturer	Insertions	Cleaning Method
Aries	10,000	None
AMP (18 pin)	25,000	None
3M Textool	10,000	Methyl Alcohol
Yamaichi	25,000	Methyl Alcohol

A.4.4.1 MANUFACTURER

All sockets (except Yamaichi) are labeled with the manufacturer's name. Identify a Yamaichi socket by looking for the letters IC51– (as the prefix to a part number) on the socket.

A.4.4.2 INSERTIONS

The expected life (number of manual insertions) has been found to be shorter than what is reported by the manufacturers. The number of manual insertions depends on the socket condition and how often the socket is cleaned.

Careless insertions or dirty socket conditions can bring the number of insertions down to less than 5,000. Cleanliness and care in inserting devices into a socket are most important with surface mount devices because the socket contacts must remain planar to function properly.

Any bent or nonplanar contacts will result in a failure. Nonplanar socket module contacts occur earlier in the life of a socket module when devices are inserted manually into a socket module. Early contact failure from manual insertions is due to the non-repeatability of the manual insertion method. Therefore, the listed number of insertions might not be reached for sockets if the devices are inserted manually. At present, there are no successful methods to ensure that the contacts on devices are planar when they are manually inserted.

A.4.4.3 CLEANING METHOD – METHYL ALCOHOL

Clean with methyl alcohol, then blow off the contacts with dry compressed air.

DANGER

Methyl alcohol is highly flammable. Use methyl alcohol in a well ventilated area away from sparks, flames or any other source of ignition. Methyl alcohol is poisonous and can cause blindness if taken internally. Avoid inhaling methyl alcohol vapor.

A.5 ICSP CABLE ADAPTER

AC002021 PM3 ICSP Cable

AC164111 PM3 ICSP RJ11 Adapter - This adapter is designed to accommodate the seamless transition from a development environment to a production environment. The adapter plugs into the 14-pin ICSP Insulation Displacement header on the MPLAB PM3 Programmer and provides an interface to which an RJ11 cable assembly (which is common to debugging environments) can be used. The RJ11 interface is the standard on most Microchip demonstration boards. Additionally, the adapter facilitates automation and program sequencing by providing a header footprint with PASS/FAIL and GO signals, and LED indicators that are useful to programming line operators.

NOTES:

Appendix B. Revision History

Revision A (August 2014)

This is the initial release of this document.

NOTES:

Glossary

A

Absolute Section

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

Absolute Variable/Function

A variable or function placed at an absolute address using the OCG compiler's `@ address` syntax.

Access Memory

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

Access Entry Points

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

Address

Value that identifies a location in memory.

Alphabetic Character

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

Alphanumeric

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

ANDed Breakpoints

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

Anonymous Structure

16-bit C Compiler **–** An unnamed structure.

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

```
union castaway
 int intval;
 struct {
  char lo; //accessible as caster.lo
  char hi; //accessible as caster.hi
 };
} caster;
```

ANSI

American National Standards Institute is an organization responsible for formulating and approving standards in the United States.

Application

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

Archive/Archiver

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

ASCII

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

Assembly/Assembler

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

Assigned Section

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

Asynchronously

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

Asynchronous Stimulus

Data generated to simulate external inputs to a simulator device.

Attribute

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

Attribute, Section

GCC Characteristics of sections, such as "executable", "readonly", or "data" that can be specified as flags in the assembler `.section` directive.

B

Binary

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

Bookmarks

Use bookmarks to easily locate specific lines in a file.

Select Toggle Bookmarks on the Editor toolbar to add/remove bookmarks. Click other icons on this toolbar to move to the next or previous bookmark.

Breakpoint

Hardware Breakpoint: An event whose execution will cause a halt.

Software Breakpoint: An address where execution of the firmware will halt. Usually achieved by a special break instruction.

Build

Compile and link all the source files for an application.

C

C\C++

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

Calibration Memory

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

Central Processing Unit

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

Clean

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

COFF

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

Command Line Interface

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

Compiled Stack

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

Compiler

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

Conditional Assembly

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

Conditional Compilation

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

Configuration Bits

Special-purpose bits programmed to set PIC microcontroller modes of operation. A Configuration bit may or may not be preprogrammed.

Control Directives

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

CPU

See Central Processing Unit.

Cross Reference File

A file that references a table of symbols and a list of files that references the symbol. If the symbol is defined, the first file listed is the location of the definition. The remaining files contain references to the symbol.

D

Data Directives

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

Data Memory

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

Data Monitor and Control Interface (DMCI)

The Data Monitor and Control Interface, or DMCI, is a tool in MPLAB X IDE. The interface provides dynamic input control of application variables in projects. Application-generated data can be viewed graphically using any of 4 dynamically-assignable graph windows.

Debug/Debugger

See ICE/ICD.

Debugging Information

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

Deprecated Features

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

Device Programmer

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

Digital Signal Controller

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

Digital Signal Processing\Digital Signal Processor

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

Directives

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

Download

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

DWARF

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

E

EEPROM

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

ELF

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

Emulation/Emulator

See ICE/ICD.

Endianness

The ordering of bytes in a multi-byte object.

Environment

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

Epilogue

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

EPROM

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

Error/Error File

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

Event

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

Executable Code

Software that is ready to be loaded for execution.

Export

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

Expressions

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

Extended Microcontroller Mode

In extended microcontroller mode, on-chip program memory as well as external memory is available. Execution automatically switches to external if the program memory address is greater than the internal memory space of the PIC18 device.

Extended Mode (PIC18 MCUs)

In Extended mode, the compiler will utilize the extended instructions (i.e., `ADDFSR`, `ADDULNK`, `CALLW`, `MOVSF`, `MOVSS`, `PUSHL`, `SUBFSR` and `SUBULNK`) and the indexed with literal offset addressing.

External Label

A label that has external linkage.

External Linkage

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

External Symbol

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

External Symbol Resolution

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

External Input Line

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

External RAM

Off-chip Read/Write memory.

F

Fatal Error

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

File Registers

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

Filter

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

Fixup

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

Flash

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

FNOP

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

Frame Pointer

A pointer that references the location on the stack that separates the stack-based arguments from the stack-based local variables. Provides a convenient base from which to access local variables and other values for the current function.

Free-Standing

An implementation that accepts any strictly conforming program that does not use complex types and in which the use of the features specified in the library clause (ANSI ‘89 standard clause 7) is confined to the contents of the standard headers `<float.h>`, `<iso646.h>`, `<limits.h>`, `<stdarg.h>`, `<stdbool.h>`, `<stddef.h>` and `<stdint.h>`.

G

GPR

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

H

Halt

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

Heap

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

Hex Code\Hex File

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

Hexadecimal

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

High Level Language

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

I

ICE/ICD

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

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

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

ICSP

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

IDE

Integrated Development Environment, as in MPLAB X IDE.

Identifier

A function or variable name.

IEEE

Institute of Electrical and Electronics Engineers.

Import

Bring data into the MPLAB X IDE from an outside source, such as from a hex file.

Initialized Data

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

```
int myVar=5;
```

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

Instruction Set

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

Instructions

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

Internal Linkage

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

International Organization for Standardization

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

Interrupt

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

Interrupt Handler

A routine that processes special code when an interrupt occurs.

Interrupt Service Request (IRQ)

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

Interrupt Service Routine (ISR)

Language tools – A function that handles an interrupt.

MPLAB X IDE – User-generated code that is entered when an interrupt occurs. The location of the code in program memory will usually depend on the type of interrupt that has occurred.

Interrupt Vector

Address of an interrupt service routine or interrupt handler.

L

L-value

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

Latency

The time between an event and its response.

Library/Librarian

See Archive/Archiver.

Linker

A language tool that combines object files and libraries to create executable code, resolving references from one module to another.

Linker Script Files

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

Listing Directives

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

Listing File

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

Little Endian

A data ordering scheme for multibyte data whereby the least significant byte is stored at the lower addresses.

Local Label

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

Logic Probes

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

Loop-Back Test Board

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

LVDS

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

With standard I/0 signaling, data storage is contingent upon the actual voltage level. Voltage level can be affected by wire length (longer wires increase resistance, which lowers voltage). But with LVDS, data storage is distinguished only by positive and negative voltage values, not the voltage level. Therefore, data can travel over greater lengths of wire while maintaining a clear and consistent data stream.

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

M

Machine Code

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

Machine Language

A set of instructions for a specific central processing unit, designed to be usable by a processor without being translated.

Macro

Macro instruction. An instruction that represents a sequence of instructions in abbreviated form.

Macro Directives

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

Makefile

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

Make Project

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

MCU

Microcontroller Unit. An abbreviation for microcontroller. Also uC.

Memory Model

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

Message

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

Microcontroller

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

Microcontroller Mode

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

Microprocessor Mode

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

Mnemonics

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

Module

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

MPASM™ Assembler

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

MPLAB *Language Tool* for *Device*

Microchip’s C compilers, assemblers and linkers for specified devices. Select the type of language tool based on the device you will be using for your application, e.g., if you will be creating C code on a PIC18 MCU, select the MPLAB C Compiler for PIC18 MCUs.

MPLAB ICD

Microchip in-circuit debugger that works with MPLAB X IDE. *See* ICE/ICD.

MPLAB X IDE

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

MPLAB PM3

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

MPLAB REAL ICE™ In-Circuit Emulator

Microchip's next-generation in-circuit emulator that works with MPLAB X IDE. *See* ICE/ICD.

MPLAB SIM

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

MPLIB™ Object Librarian

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

MPLINK™ Object Linker

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

MRU

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

N

Native Data Size

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

Nesting Depth

The maximum level to which macros can include other macros.

Node

MPLAB X IDE project component.

Non-Extended Mode (PIC18 MCUs)

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

Non Real Time

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

Non-Volatile Storage

A storage device whose contents are preserved when its power is off.

NOP

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

O

Object Code/Object File

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

Object File Directives

Directives that are used only when creating an object file.

Octal

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

Off-Chip Memory

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

Opcodes

Operational Codes. *See* Mnemonics.

Operators

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

OTP

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

P

Pass Counter

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

PC

Personal Computer or Program Counter.

PC Host

Any PC running a supported Windows operating system.

Persistent Data

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

Phantom Byte

An unimplemented byte in the dsPIC architecture that is used when treating the 24-bit instruction word as if it were a 32-bit instruction word. Phantom bytes appear in dsPIC hex files.

PIC MCUs

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

PICkit 2 and 3

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

Plug-ins

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

Pod

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

Power-on-Reset Emulation

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

Pragma

A directive that has meaning to a specific compiler. Often a pragma is used to convey implementation-defined information to the compiler. MPLAB C30 uses attributes to convey this information.

Precedence

Rules that define the order of evaluation in expressions.

Production Programmer

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

Profile

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

Program Counter

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

Program Counter Unit

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

Program Memory

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

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

Project

A project contains the files needed to build an application (source code, linker script files, etc.) along with their associations to various build tools and build options.

Prologue

A portion of compiler-generated code that is responsible for allocating stack space, preserving registers and performing any other machine-specific requirement specified in the runtime model. This code executes before any user code for a given function.

Prototype System

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

Psect

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

PWM Signals

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

Q

Qualifier

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

R

Radix

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

RAM

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

Raw Data

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

Read Only Memory

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

Real Time

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

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

Recursive Calls

A function that calls itself, either directly or indirectly.

Recursion

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

Reentrant

A function that may have multiple, simultaneously active instances. This can happen due to either direct or indirect recursion or through execution during interrupt processing.

Relaxation

The process of converting an instruction to an identical, but smaller instruction. This is useful for saving on code size. MPLAB XC16 currently knows how to `relax` a `CALL` instruction into an `RCALL` instruction. This is done when the symbol that is being called is within +/- 32k instruction words from the current instruction.

Relocatable

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

Relocatable Section

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

Relocation

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

ROM

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

Run

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

Run-time Model

Describes the use of target architecture resources.

Runtime Watch

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

S

Scenario

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

Section

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

Section Attribute

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

Sequenced Breakpoints

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

Serialized Quick Turn Programming

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

Shell

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

Simulator

A software program that models the operation of devices.

Single Step

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

Skew

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

Skid

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

Source Code

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

Source File

An ASCII text file containing source code.

Special Function Registers (SFRs)

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

SQTP

See Serialized Quick Turn Programming.

Stack, Hardware

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

Stack, Software

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

Stack, Compiled

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

MPLAB Starter Kit for *Device*

Microchip's starter kits contains everything needed to begin exploring the specified device. View a working application and then debug and program you own changes.

Static RAM or SRAM

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

Status Bar

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

Step Into

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

Step Over

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

Step Out

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

Stimulus

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

Stopwatch

A counter for measuring execution cycles.

Storage Class

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

Storage Qualifier

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

Symbol

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

Symbol, Absolute

Represents an immediate value such as a definition through the assembly `.equ` directive.

System Window Control

The system window control is located in the upper left corner of windows and some dialogs. Clicking on this control usually pops up a menu that has the items “Minimize,” “Maximize,” and “Close.”

T

Target

Refers to user hardware.

Target Application

Software residing on the target board.

Target Board

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

Target Processor

The microcontroller device on the target application board.

Template

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

Tool Bar

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

Trace

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

Trace Memory

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

Trace Macro

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

Trigger Output

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

Trigraphs

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

U

Unassigned Section

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

Uninitialized Data

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

```
int myVar;
```

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

Upload

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

USB

Universal Serial Bus. An external peripheral interface standard for communication between a computer and external peripherals over a cable using bi-serial transmission. USB 1.0/1.1 supports data transfer rates of 12 Mbps. Also referred to as high-speed USB, USB 2.0 supports data rates up to 480 Mbps.

V

Vector

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

Volatile

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

W

Warning

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

16-bit assembler/compiler – Warnings report conditions that could indicate a problem, but do not halt processing. In MPLAB C30, warning messages report the source file name and line number, but include the text ‘`warning:`’ to distinguish them from error messages.

Watch Variable

A variable that you can monitor during a debugging session in a Watch window.

Watch Window

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

Watchdog Timer (WDT)

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

Workbook

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

NOTES:

Index

MICROCHIP

Worldwide Sales and Service

7AMERICAS

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

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

Austin, TX
Tel: 512-257-3370

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

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

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

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

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

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

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110

Canada - Toronto
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: 852-2943-5100
Fax: 852-2401-3431

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

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

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

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
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: 86-755-8864-2200
Fax: 86-755-8203-1760

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

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

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

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

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

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

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

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: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600
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 - Dusseldorf
Tel: 49-2129-3766400

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

Germany - Pforzheim
Tel: 49-7231-424750

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

Italy - Venice
Tel: 39-049-7625286

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

Poland - Warsaw
Tel: 48-22-3325737

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

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

03/25/14