



MPLAB Starter Kit for PIC18F User's Guide

Note the following details of the code protection feature on Microchip devices:

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

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

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in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates.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

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, rfPIC and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.


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

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, nanoWatt XLP, Omniscent Code Generation, PICC, PICC-18, PICKit, PICDEM, PICDEM.net, PICTail, PIC³² logo, REAL ICE, rLAB, Select Mode, Total Endurance, TSHARC, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

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

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

© 2009, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.

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

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Table of Contents

Preface	1
Chapter 1. Introduction to the Starter Kit	
1.1 Overview	7
1.2 Operational Requirements	7
1.3 Initial Board Setup	7
Chapter 2. The Demonstration Application	
2.1 Bootloader From the MicroSD™ Memory Card	9
2.2 USB HID Mouse Demo	11
2.3 USB HID Joystick Demo	12
2.4 USB HID Custom Device Demo	13
2.5 USB CDC Demo	14
2.6 USB MSD Card Reader Demo	14
Chapter 3. Developing an Application	
3.1 Setting Up an Example Application for Debug	16
3.2 Running the Example Application	18
3.3 Debugging the Example Application	18
3.4 Programming the Debugged Application	21
3.5 Creating Other Applications	21
3.6 Determining Device Support	21
3.7 Reserved Resources	21
3.8 Debug Version Information	21
3.9 Troubleshooting	22
Chapter 4. Hardware	
4.1 Application Functional Overview	23
4.2 Programmer/Debugger Functional Overview	24
4.3 Board Components	25
Appendix A. Starter Kit Schematics	29
Appendix B. USB Current Draw Requirements	33
B.1 USB Current Specifications	33
B.2 PIC18F Soft Start Circuitry	34
B.3 Soft Start Functions and Usage	35
B.4 USB Suspend Current Limitations for the Starter Kit	37
Index	39
Worldwide Sales and Service	42

MPLAB Starter Kit for PIC18F User's Guide

NOTES:

Preface

NOTICE TO CUSTOMERS

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

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

For the most up-to-date information on development tools, see the MPLAB® IDE 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 you use the MPLAB Starter Kit for PIC18F. Items discussed in this chapter include:

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

DOCUMENT LAYOUT

This document describes how to use the starter kit as a development and demonstrative tool for PIC18F J-series device capabilities and features. The manual layout is as follows:

- **Chapter 1. Introduction to the Starter Kit** – This chapter introduces the starter kit and provides an overview of its features.
- **Chapter 2. The Demonstration Application** – This chapter describes the preprogrammed PIC18F demo application.
- **Chapter 3. Developing an Application** – This chapter describes how to debug application software on the starter kit using MPLAB® IDE.
- **Chapter 4. Hardware** – This chapter provides a functional overview of the starter kit and identifies the major hardware components.
- **Appendix A. Starter Kit Schematics** – This appendix provides detailed schematic diagrams of the starter kit.
- **Appendix B. USB Current Draw Requirements** – This appendix briefly discusses USB current consumption specifications and the methods used by the starter kit to meet those specifications.

MPLAB Starter Kit for PIC18F User's Guide

CONVENTIONS USED IN THIS GUIDE

This manual uses the following documentation conventions:

DOCUMENTATION CONVENTIONS

Description	Represents	Examples
Arial font:		
Italic characters	Referenced books	<i>MPLAB[®] IDE User's Guide</i>
	Emphasized text	...is the <i>only</i> 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	<u><i>File>Save</i></u>
Bold characters	A dialog button	Click OK
	A tab	Click the Power tab
Text in angle brackets < >	A key on the keyboard	Press <Enter>, <F1>
Courier New font:		
Plain 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	<i>file.o</i> , where <i>file</i> can be any valid filename
Square brackets []	Optional arguments	mcc18 [options] <i>file</i> [options]
Curly brackets and pipe character: { }	Choice of mutually exclusive arguments; an OR selection	errorlevel {0 1}
Ellipses...	Replaces repeated text	var_name [, var_name...]
	Represents code supplied by user	void main (void) { ... }

WARRANTY REGISTRATION

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

RECOMMENDED READING

This user's guide describes how to use the MPLAB Starter Kit for PIC18F. Other useful documents are listed below. The following Microchip documents are available and recommended as supplemental reference resources.

Readme Files

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

PIC18FJ Device Data Sheets and Flash Programming Specifications

Refer to the appropriate device Data Sheet for device-specific information and specifications. Also, refer to the appropriate device Flash Programming Specification for information on instruction sets and firmware development. These files may be found on the Microchip web site or from your local sales office.

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

This document discusses the MPLAB ASM Assembler, MPLAB LINK Linker, MPLIB Archiver/Librarian and other 8-bit device utilities, and helps you use these tools in developing applications.

MPLAB® C18 C Compiler User's Guide (DS51288)

This document helps you use Microchip's MPLAB C18 C compiler to develop your application.

MPLAB® IDE User's Guide (DS51519)

This document describes how to use the MPLAB IDE (Integrated Development Environment), as well as the MPLAB IDE Project Manager, MPLAB IDE Editor and MPLAB SIM Simulator. Use these development tools to help you develop and debug application code.

THE MICROCHIP WEB SITE

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

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

DEVELOPMENT SYSTEMS CUSTOMER CHANGE NOTIFICATION SERVICE

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

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

The Development Systems product group categories are:

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

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

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

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

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

DOCUMENT REVISION HISTORY

Revision A (June 2009)

- Initial Release of this Document

NOTES:

Chapter 1. Introduction to the Starter Kit

Thank you for purchasing Microchip Technology's MPLAB Starter Kit for PIC18F. This board is intended to introduce and demonstrate the capabilities and features of the PIC18F J-series of Flash microcontrollers. In addition, the starter kit has on-board, in-circuit debug circuitry so that you may develop and debug your own applications.

This chapter introduces the starter kit and provides an overview of its features. Topics covered include:

- Overview
- Operational Requirements
- Initial Board Setup

1.1 OVERVIEW

The MPLAB Starter Kit for PIC18F provides an all-in-one solution for debugging and programming applications using Microchip's own MPLAB Integrated Development Environment (IDE). A USB connection to a host computer supplies communications and power to the board; no additional external power supply is needed.

The starter kit includes integrated debug and programmer circuitry that allows applications to be programmed onto the application side of the PIC18F MCU and then debugged, all using MPLAB IDE. The need for an additional programmer or hardware interface has been completely eliminated.

The application side of the starter kit contains a range of hardware components to demonstrate the utility and processing power of Microchip's PIC18F46J50 family USB microcontrollers.

1.2 OPERATIONAL REQUIREMENTS

To communicate with and program the starter kit, the following hardware and software requirements must be met:

- PC compatible system with CD-ROM drive
- One available USB port on the PC or a powered USB hub
- MicroSD flash memory card (a card preloaded with demo applications is provided)
- Microsoft® Windows® 2000 SP4, Windows XP SP2 or Windows Vista® (32-bit)

<p>Note: Only initial testing has been performed on 32-bit Windows Vista for this release of the demo applications. The 64-bit version is not supported at this time.</p>
--

1.3 INITIAL BOARD SETUP

With its pre-installed demo application, the MPLAB Starter Kit for PIC18F is designed to be used straight out of the box. Except for a single connection to a computer, no additional hardware or configuration is necessary.

1.3.1 Installing the Software

Before connecting the starter kit to any computer for the first time, it is important to install the accompanying software on the MPLAB Starter Kit for PIC18F CD first. This ensures that both the MPLAB development environment and C compiler, as well as the relevant tutorial and help files for the starter kit, are ready to use when the board is connected.

To install the software, insert the starter kit CD into the CD-ROM drive. The installation process starts automatically. The process pauses for user responses to accept the Microchip software licenses and to confirm the installation directories; respond appropriately.

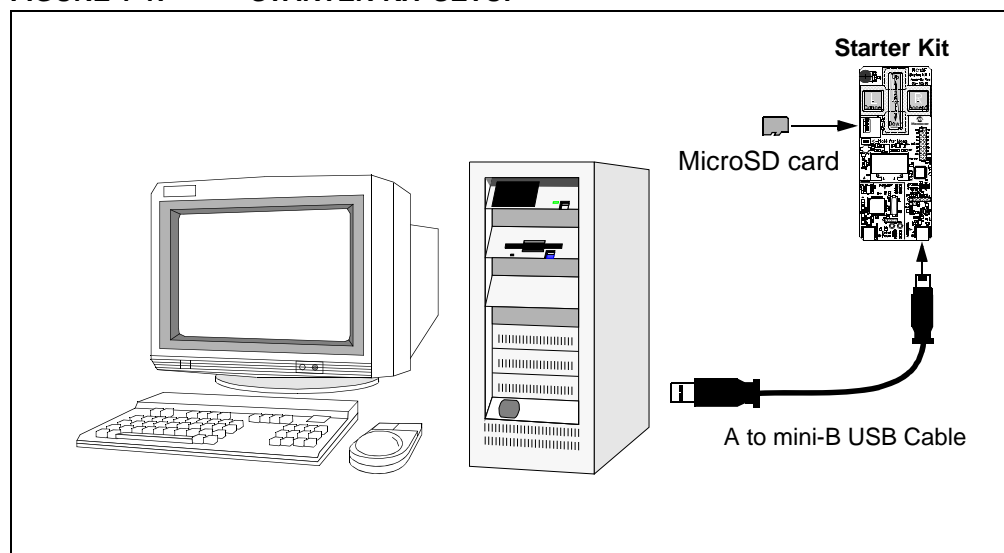
1.3.2 Connecting the Hardware

Once the starter kit software is installed, connect the provided USB cable (A to mini-B) to any available USB port on the PC or powered hub, then to the starter kit at the mini-B receptacle on the application side of the board (on the right side of the board's lower edge, as shown in Figure 1-1). The PC USB connection provides communication and power to the board. The MicroSD™ Flash card, used to store the demo applications, may be inserted into the MicroSD slot at any time.

If the cable is connected correctly, the green power LED is lit. The OLED display will display the "Microchip PIC18F Starter Kit" main menu. At this point, the application waits for the bootloader Reset switch to be pressed. When it is pressed, the bootloader application is loaded from the MicroSD card and executed. At this point, you can use the "Up" and "Down" touch pads to scroll through the available menus, and the "Cancel" and "Accept" pads to select menu items. Refer to **Chapter 2. "The Demonstration Application"** for a complete description of how to use the bootloader application.

When one of the precompiled applications is executed, a sequence of pop-up balloons in the system tray (lower right of desktop) should appear, stating that (1) new hardware has been found, (2) drivers are being installed and (3) the new hardware is ready for use. If you do not see these messages and the starter kit does not work, try unplugging and reconnecting the USB. If this does not work, refer to **Section Chapter 3. "Developing an Application"**.

FIGURE 1-1: STARTER KIT SETUP



Chapter 2. The Demonstration Application

This chapter describes the precompiled demonstration applications that are provided with the MPLAB Starter Kit for PIC18F, and how various features highlight the device's processing power and hardware functionality. These include:

- Bootloader from the MicroSD Memory Card
- Interactive Display System (OLED)
- Touch Interface (CTMU)
- USB Device Capabilities

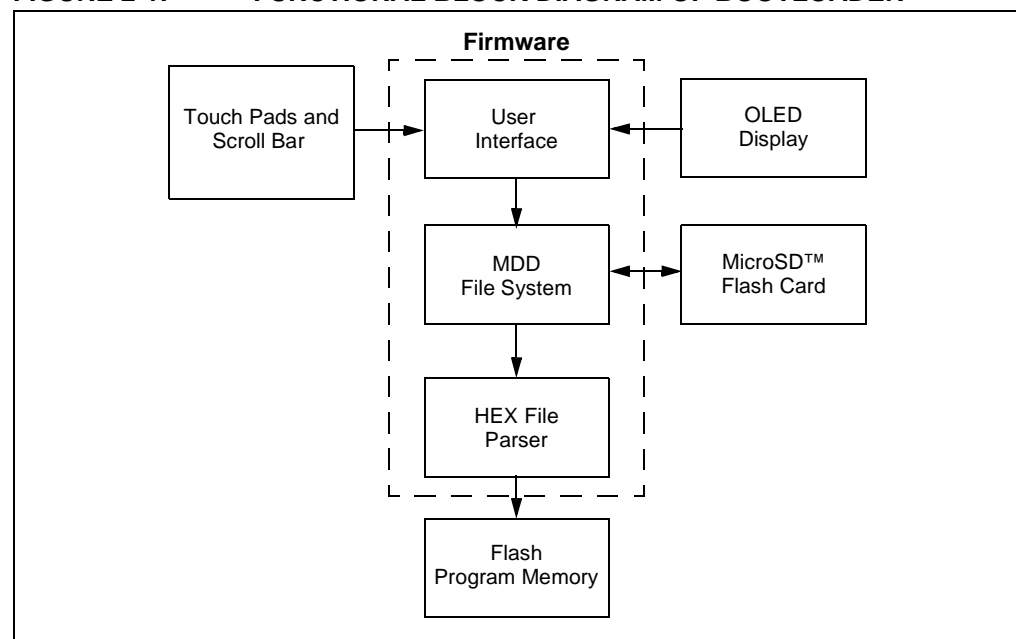
2.1 BOOTLOADER FROM THE MicroSD™ MEMORY CARD

The Flash program memory of Microchip's PIC18F46J50 device is readable, writable and erasable during normal operation over the entire VDD range. This feature gives users the ability to perform bootloader operations. To demonstrate this feature, the MPLAB Starter Kit for PIC18F is preprogrammed with a MicroSD bootloader that gives the ability to load, program and run PIC18 applications from a MicroSD memory card. The bootloader application extensively uses, and serves to demonstrate, the Microchip Memory Disk Drive (MDD) file system.

The MicroSD bootloader accepts user inputs from the touch pads and scroll bar, uses the OLED display for user outputs, and uses substantial parts of the Microchip Memory Disk Drive file system to read files and folders from a MicroSD memory card.

Figure 2-1 summarizes the essential firmware design of the bootloader. At system Reset, the MicroSD bootloader checks for a user request to run the bootloader (by pressing the S1 switch within a 10-second window). If this is not the case, the bootloader returns control to the application currently programmed in the microcontroller's memory.

FIGURE 2-1: FUNCTIONAL BLOCK DIAGRAM OF BOOTLOADER



MPLAB Starter Kit for PIC18F User's Guide

When the MicroSD bootloader starts, it initializes the capacitive touch routines and the OLED display. It then monitors for a MicroSD memory card insertion. When an insert event is detected by the bootloader, the MDD file system is initialized and the file structure is displayed on the OLED. Note that only folders and HEX files will be displayed.

The user can press the **Up** and **Down** buttons to select a specific file or folder. Pressing the **Accept** button when a folder is selected causes the bootloader to display the content of that specific folder on the OLED. To move one level up in the directory hierarchy, either press the **Cancel (Left)** button, or press the **Accept** button when the “.” folder is selected.

When a HEX file is selected and the **Accept (Right)** button is pressed, the MicroSD bootloader will prompt the user for a confirmation to load the file from the MicroSD memory card and program it to the microcontroller's Flash memory. After this operation is finished, the user may choose to execute that application or to remain in the bootloader.

If the **Cancel** button is pressed in the root directory, the MicroSD bootloader terminates. The last application programmed into the microcontroller is then executed.

Figure 2-2 shows a typical sequence of OLED messages from the bootloader initialization through the loading of an application (read from top left to bottom right).

FIGURE 2-2: TYPICAL SEQUENCE FOR BOOTLOADER MESSAGES



2.1.1 Memory Organization

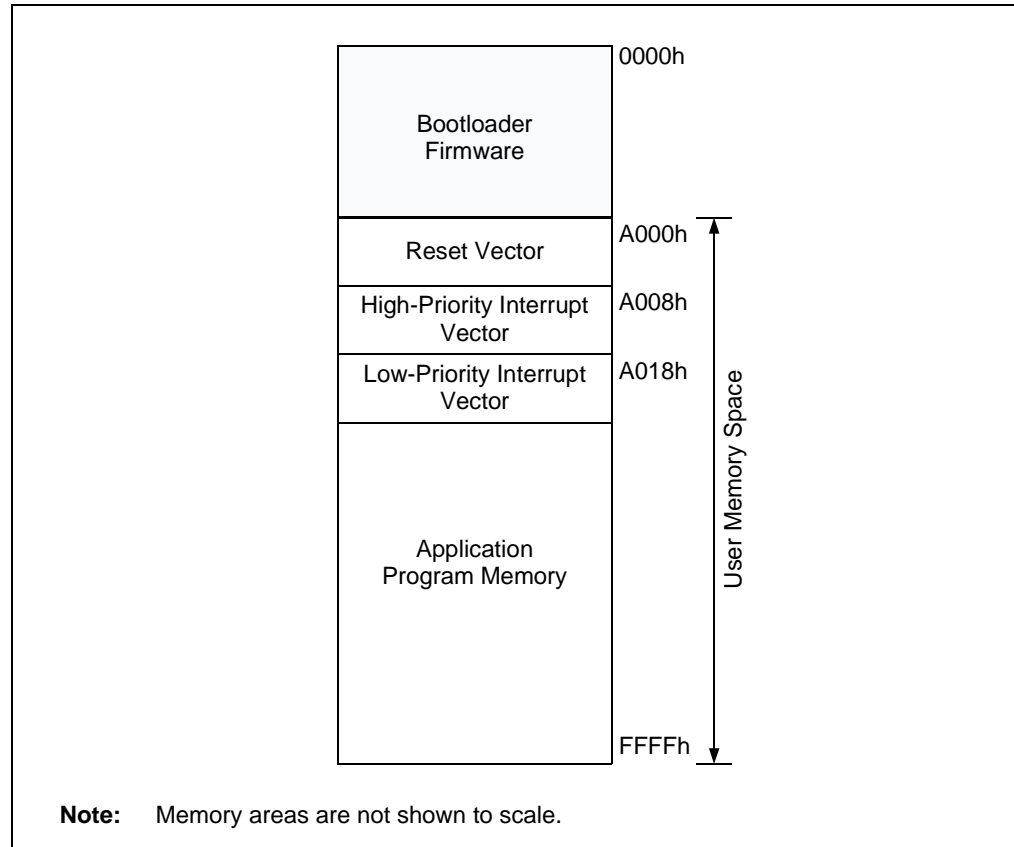
The first 40 Kbytes of Flash program memory are used by the MicroSD bootloader. Figure 2-3 shows the memory map of the PIC18F46J50. The application code must ensure that the boot area is not overwritten.

2.1.2 Remapped Vectors

Since the hardware Reset and interrupt vectors lie within the boot area, they are remapped through software to the beginning of the user memory space. Remapping is simply a branch for interrupts; users may observe an additional latency of two instruction cycles to handle interrupts.

The Demonstration Application

FIGURE 2-3: PROGRAM MEMORY MAP OF THE PIC18F46J50 DEMO



2.1.3 Writing Application Code

The bootloader operates as a separate entity, which means that an application can be developed with very little concern about what the bootloader is doing.

When using the MPLAB® C18 C compiler to develop PIC18F firmware for an application, the linker file must be changed to protect the program memory used by the bootloader and the interrupt handler must be updated with the new vectors. An example linker file (18f46j50_g.lkr) is provided on the starter kit CD-ROM.

If the need exists to re-enter the bootloader from the application (and it usually does), it will be necessary to monitor the state of the S1 switch periodically. Since the bootloader assumes Reset conditions, a RESET instruction should be initiated if the S1 is pressed. An example stub application (USB Device-HID-Mouse\Firmware\mouse.c) on the starter kit CD demonstrates how to re-enter the bootloader from the main application.

2.2 USB HID MOUSE DEMO

This demo uses the starter kit as a USB mouse. When the application is run, the starter kit functions like any other mouse under the Windows operating system.

The movement of the mouse cursor on the X and Y axes is controlled using the on-board accelerometer. The acceleration on all three axes is used to compute the board tilt, which is then converted to movement of the mouse cursor on the X and Y axis. This conversion is not linear, allowing for slow movement of the cursor when the board is tilted a small amount, and fast movement for a steeper tilt. Keeping the board level (with respect to the ground) stops all cursor motion.

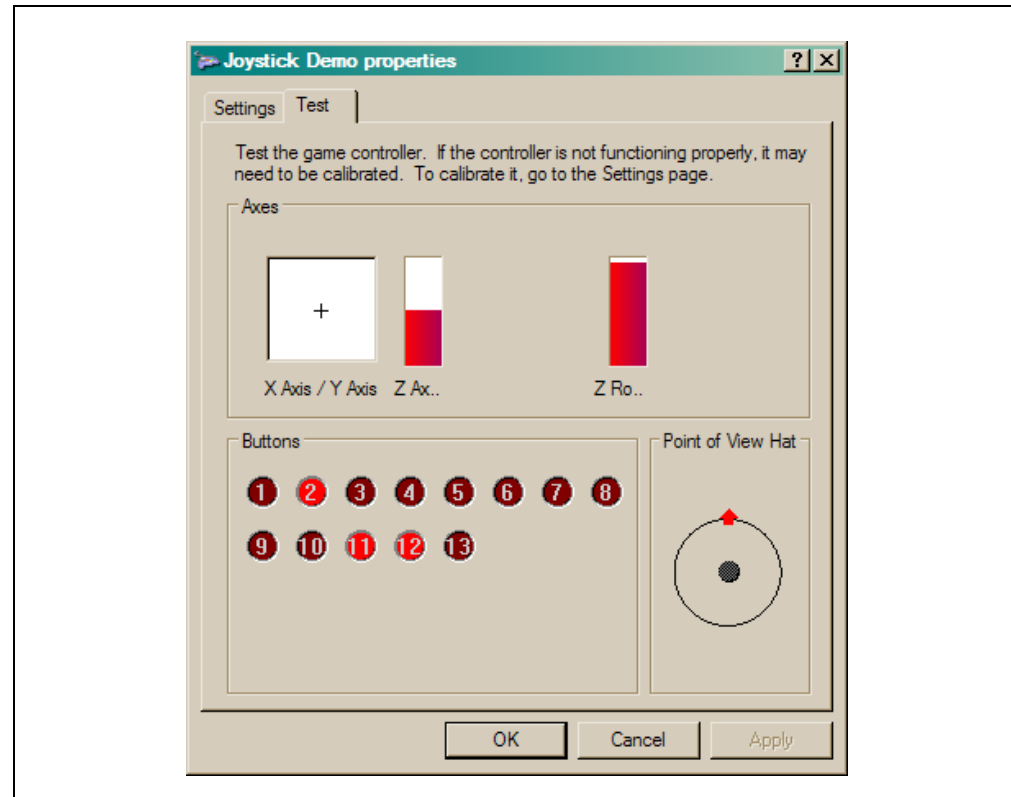
The touch pads are used for the **Left** and **Right** mouse buttons, while the scroll bar is used for the mouse scroll wheel.

2.3 USB HID JOYSTICK DEMO

This demo uses the starter kit as a USB joystick.

To test the joystick feature, open the Game Controller applet in the Control Panel. In the following dialog, select “Joystick Demo” under “Installed Game Controllers”, then click **Properties**. At the “Joystick Demo properties” dialog, select the **Test** tab to show the interactive test display (Figure 2-4).

FIGURE 2-4: JOYSTICK DEMO PROPERTIES DIALOG



The following controls are implemented in the Joystick Demo:

1. **X-Axis/Y-Axis:** The movement of the cursor around the white area is controlled using the on-board accelerometer. Like the Mouse Demo, acceleration on all three axes is used to compute board tilt, which is then converted to movement on the X and Y axis. While holding the starter kit board so that the USB connector faces towards you, tilt the board to the right and left to change the X-axis position. Tilt the board up and down to change the Y-axis position.
2. **Z-Axis:** The Z-axis is controlled using the scroll bar. Touch the bar and slide your finger towards the bottom or top to change the Z-axis position.
3. **Z-Rotation:** Rotation around the Z-axis is controlled with the potentiometer. Rotate the potentiometer to change the rotation.
4. **Buttons 1 and 2:** Of the 13 buttons displayed, buttons 1 and 2 are controlled using the L and R touch pads, respectively. Press and hold the pad to turn on (light up) the corresponding joystick indicator.

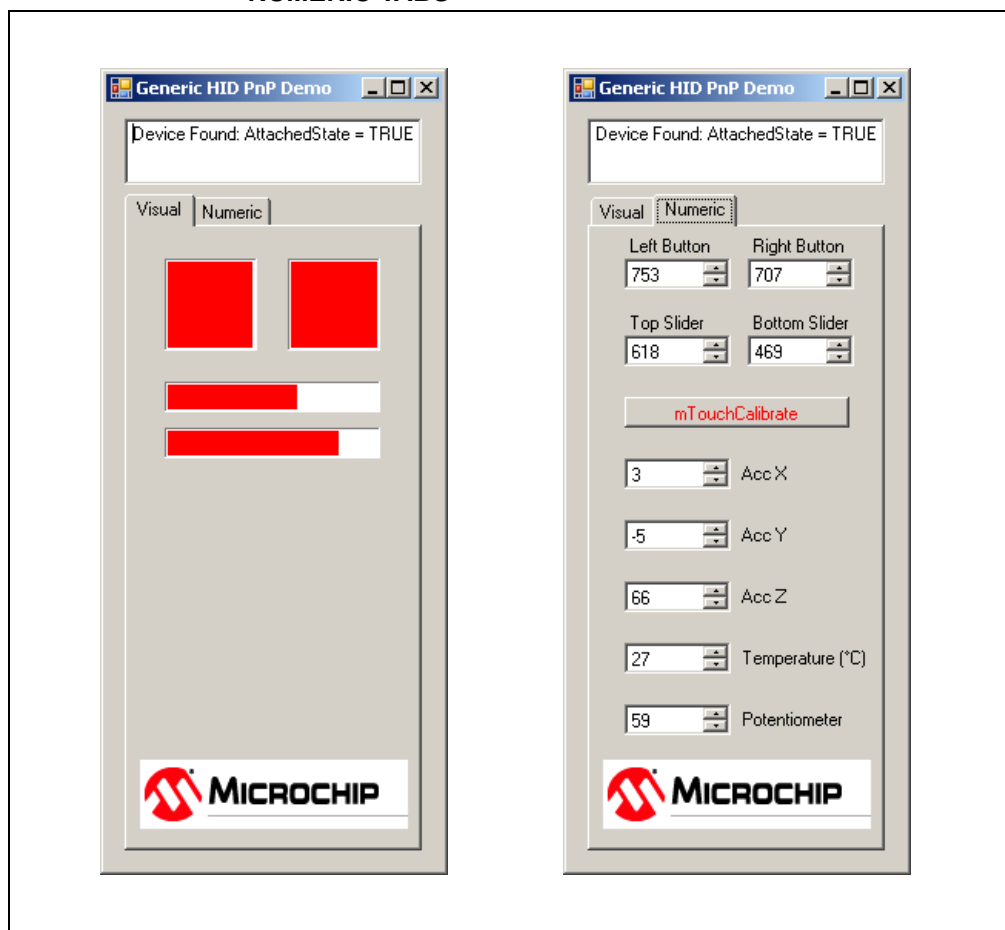
2.4 USB HID CUSTOM DEVICE DEMO

The HID Custom Device demo enables generic USB data transfer to and from an attached device. Although the HID class normally is used for interface products, such as keyboards and mice, the HID interface can also be used for other data transfer purposes. The precompiled application provided with the starter kit is meant to be used with the PC application project, `HID_PnP_Demo.exe`, provided on the CD-ROM disc that comes with the kit.

If the USB device has been plugged in and has successfully enumerated, the device should be detected automatically by the demo PC application. Figure 2-5 shows how the PC application appears in Windows XP.

After the device enumerates, the PC application reads the raw data from the touch pads, the accelerometer (three axes acceleration and temperature) and the potentiometer. The state of the touch pads (pressed or not) is displayed in the **Visual** tab; all of the raw data is displayed in the **Numeric** tab.

FIGURE 2-5: HID DEMO APPLICATION, VISUAL (LEFT) AND NUMERIC TABS



Note: The demo PC application is intended to be run under Windows and was written in Microsoft® Visual C++®. If the executable fails to launch correctly, verify that the .NET Framework Version 2.0 (or later) redistributable package has been installed. To verify this, check the list of installed programs in the Control Panel (from the **Start** button in Windows XP, *Settings>Control Panel>Add or Remove Programs*).

If it is not installed, the .NET redistributable package is included on the CD that comes with the starter kit. It is also available for download from the Microsoft web site.

2.4.1 mTouch™ Calibration

In addition to displaying information from the accelerometer and touch pads, the **Numeric** tab also has an interactive **mTouch calibration** button. Clicking on the button automatically reruns the same touch pad calibration routine that is performed whenever the starter kit is initialized. No additional input from the user is required.

Note that manual calibrations are not stored for future use, but are replaced with the results of the automatic calibration the next time the starter kit is powered up.

2.5 USB CDC DEMO

This demo allows the device to appear as a serial (COM) port to the host computer.

If the starter kit is running this demo, and it is connected for the first time to a host computer, the operating system may request a driver to install (i.e., an INF file). Select the "Install from a list or specific location (Advanced)" option and point to the INF file provided on the CD-ROM disc that comes with the board.

Once the device is successfully installed:

1. Launch a terminal emulator program, such as HyperTerminal.
2. Select the appropriate COM port. On most machines, this will be COM5 or higher.

Once the starter kit is connected, it will send data from the acceleration sensor in raw numeric format.

2.6 USB MSD CARD READER DEMO

This demo uses the starter kit as a USB MicroSD card reader. The MSD (Mass Storage Device) interface displays a new drive volume in "My Computer", similar to a USB Flash memory device ("thumb drive"). If a compatible card is inserted in the card slot, files can be read, deleted and manipulated like any other drive on the computer, following successful enumeration.

Common operating systems typically assign USB mass storage volumes (such as Flash memory devices) to the next available drive letter. If the automatically assigned drive letter has already been assigned to some other device (such as a mapped network drive), a conflict can occur with some Windows operating systems. For more information, refer to the Microsoft Knowledge Base article 297694 at the following link:

<http://support.microsoft.com/kb/297694>

To resolve this issue, do one of the following:

- Obtain the hot fix available from Microsoft
- Manually reassign the conflicting drive volume (such as a mapped network drive) with a drive letter that is later in the alphabet (such as "Z")

Chapter 3. Developing an Application

The MPLAB Starter Kit for PIC18F may be used with MPLAB® IDE, the free integrated development environment available on Microchip's web site. MPLAB IDE allows the starter kit to be used as an in-circuit debugger, as well as a programmer, for the featured device.

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

Special starter kit software interacts with the MPLAB IDE application to run, stop and single step through programs. Breakpoints can be set and the processor can be reset. Once the processor is stopped, the register's contents can be examined and modified. For more information on how to use MPLAB IDE, reference the following documentation:

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

This chapter includes the following:

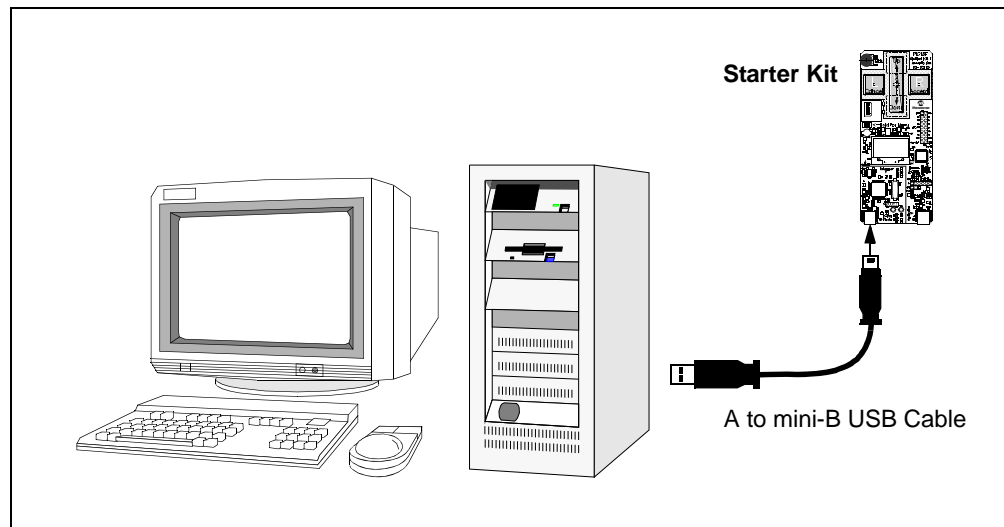
- Setting Up an Example Application for Debug
- Running the Example Application
- Debugging the Example Application
- Programming the Debugged Application
- Creating Other Applications
- Determining Device Support
- Reserved Resources
- Debug Version Information
- Troubleshooting

MPLAB Starter Kit for PIC18F User's Guide

3.1 SETTING UP AN EXAMPLE APPLICATION FOR DEBUG

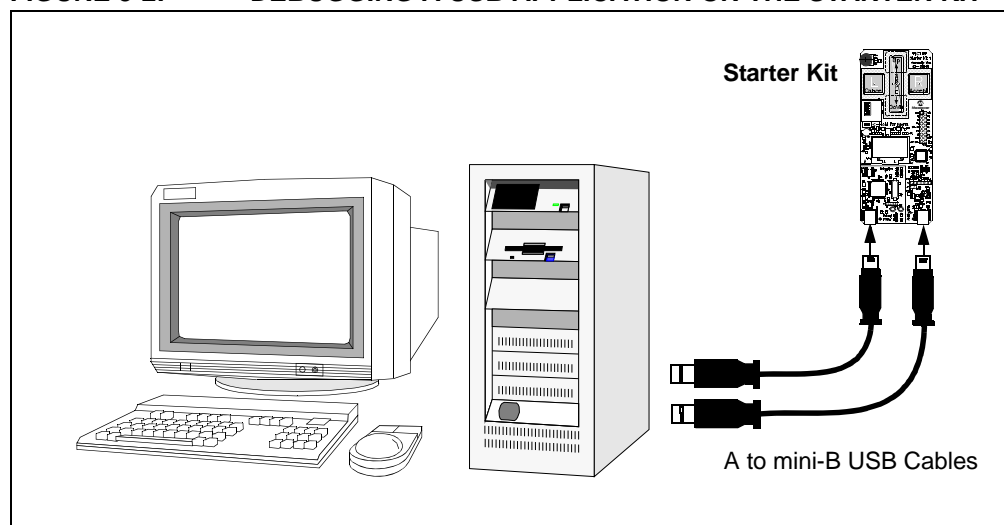
Once the starter kit software is installed, the on-board debugger must be connected to any available USB port on the PC or powered hub, then to the starter kit at the mini-B receptacle, J2, on the debugger/programmer side of the board (Figure 3-1). At the same time, a sequence of pop-up balloons in the system tray (lower right of desktop) will appear, stating that (1) new hardware has been found, (2) drivers are being installed and (3) the new hardware is ready for use. If you do not see these messages and the starter kit does not work, try unplugging and reconnecting the USB. If this does not work, refer to **Section 3.9 “Troubleshooting”**.

FIGURE 3-1: CONNECTING THE STARTER KIT DEBUGGER/PROGRAMMER



If you debug a USB device application, a second USB cable must be connected to another USB port on the PC or power hub, then to the starter kit at the mini-B receptacle, J3, on the application side of the board (Figure 3-2).

FIGURE 3-2: DEBUGGING A USB APPLICATION ON THE STARTER KIT



Developing an Application

The MPLAB IDE software that is installed on your PC by the starter kit CD-ROM automatically opens an example application that you may use to examine debug features of the starter kit. To prepare the application for debug:

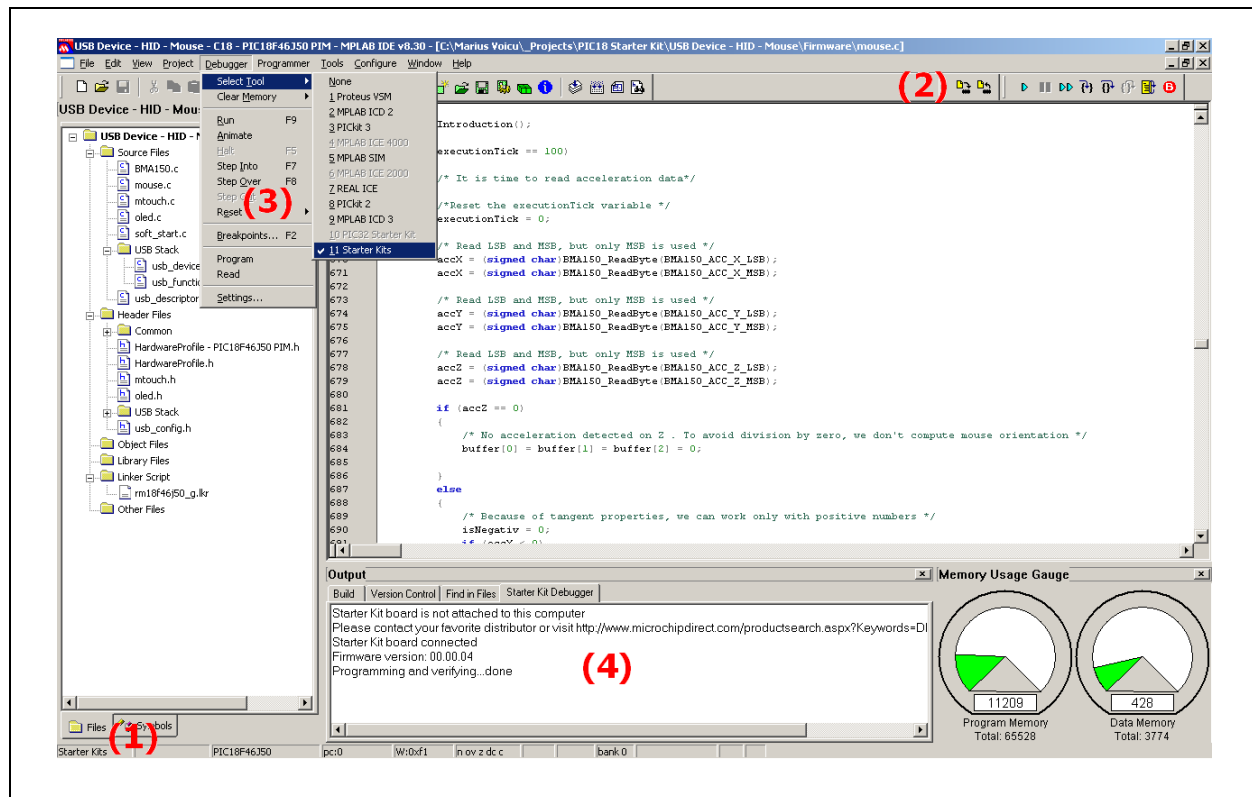
1. Launch MPLAB IDE. The example application project and related workspace will open. For information on projects and workspaces, see the MPLAB IDE documentation mentioned at the beginning of this chapter.
2. Select **Project>Build All** to build the application code. The build's progress will be visible in the **Build** tab of the Output window.
3. Select **Debugger>Select Tool>Starter Kits**. MPLAB IDE changes to add the starter kit debug features (Figure 3-3):
 - 1) the status bar shows "Starter Kits" as the debug tool
 - 2) a "Starter Kit" debug toolbar is added
 - 3) the Debugger menu changes to add starter kit debug functions
 - 4) the Output window displays the communication status between MPLAB IDE and the starter kit on the **Starter Kit Debugger** tab.

Several device resources are also used for debug. For details, see **Section 3.6 "Determining Device Support"**.

4. Select **Debugger>Program** to program the application code into the PIC18F46J50 device on the starter kit. The debug programming progress will be visible in the **Starter Kit** tab of the Output window.

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

FIGURE 3-3: STARTER KIT AS DEBUG TOOL IN MPLAB® IDE









MPLAB Starter Kit for PIC18F User's Guide

3.2 RUNNING THE EXAMPLE APPLICATION

The starter kit executes in either real time (Run) or steps (Step Into, Step Over, Animate). Real-time execution occurs when you select Run in MPLAB IDE. Once the device code is halted, either by a Halt or a breakpoint, you can step.

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

TABLE 3-1: PIC18F STARTER KIT DEBUGGER TOOLBAR

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

To see how these options function, do the following:

1. Select Debugger>Reset>Processor Reset or click the **Reset** button to reset the program.
2. Select Debugger>Run or click the **Run** button. Observe how the application operates.
3. Select Debugger>Halt or click the **Halt** button to stop the program execution. A green solid arrow marks the line of code in the File window where the program Halted.
4. Select Debugger>Step Into or click the **Step Into** button to step the program execution once. The green solid arrow will move down one line of code in the File window. Click the button several times to step through some code.
5. Select Debugger>Reset>Processor Reset and click the **Reset** button to reset the program again. The arrow disappears, meaning the device is Reset.

3.3 DEBUGGING THE EXAMPLE APPLICATION

For the example code given, everything works fine. However, when you are developing code, it will likely not work the first time and need to be debugged. MPLAB IDE provides an editor and several debug features, such as breakpoints and Watch windows, to aid in application code debugging.

This section includes:

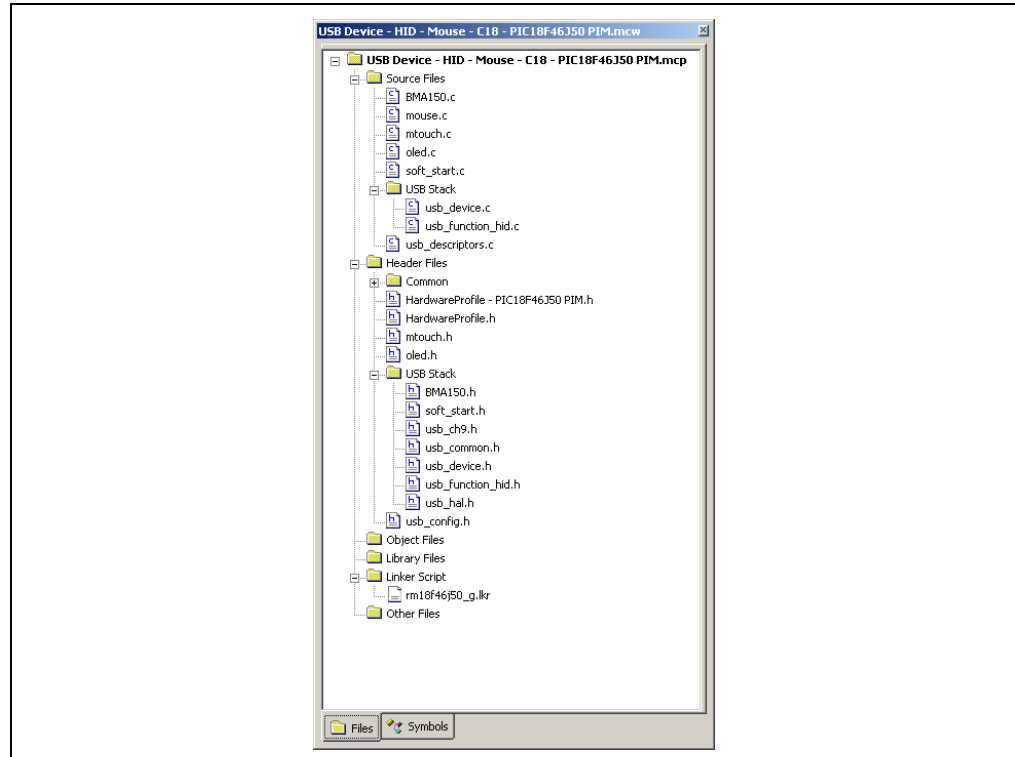
- Editing Application Code
- Using Breakpoints and Mouseovers
- Using Watch Windows

3.3.1 Editing Application Code

To view application code so that it may be edited, do one of the following:

- Select Edit>New to create new code or Edit>Open to search for and open an existing code file.
- Double-click on a file in the Project window to open an existing code file. See an example Project window in Figure 3-4.

FIGURE 3-4: EXAMPLE PROJECT



For more information on using the editor to create and edit code, see MPLAB IDE Editor Help.

3.3.2 Using Breakpoints and Mouseovers

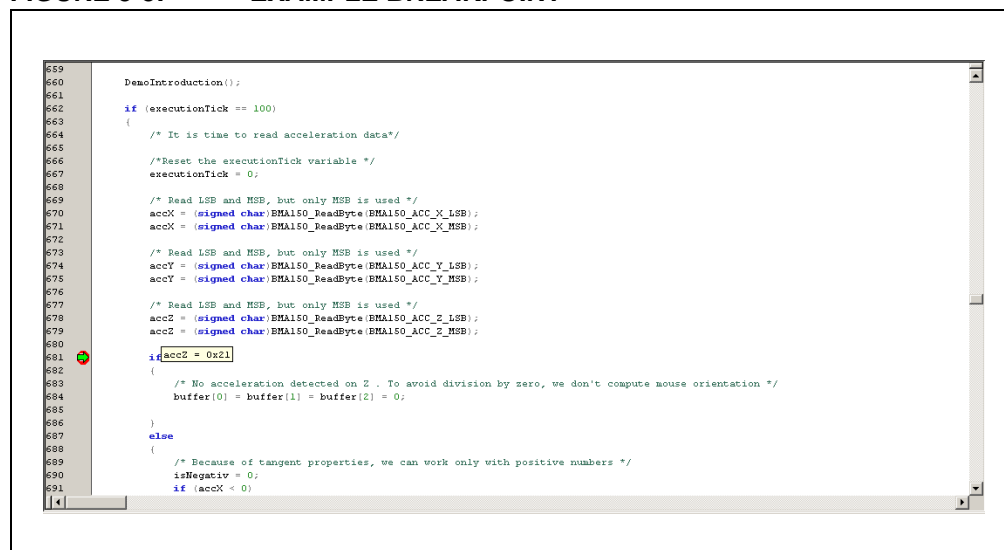
To set a breakpoint in code, use one of the following methods:

- **Double-Click in Gutter:** Double-click in the window gutter next to the line of code where you want the breakpoint. Double-click again to remove the breakpoint.
- **Pop-up Menu:** Place the cursor over the line of code where you want the breakpoint. Then, right click to pop up a menu and select "Set Breakpoint". Once a breakpoint is set, "Set Breakpoint" will become "Remove Breakpoint" and "Disable Breakpoint". Other options on the pop-up menu under Breakpoints are for deleting, enabling or disabling all breakpoints.
- **Breakpoint Dialog:** Open the Breakpoint dialog (*Debugger>Breakpoints*) to set, delete, enable or disable breakpoints. See MPLAB IDE Help for more information on this dialog.

A breakpoint set in code appears as a red "stop sign" with a "B". Once code is halted, hovering over variables pops up the current value of those variables (Figure 3-5).

Note: This feature must be set up. From the menu bar, select *Edit>Properties*; from the Properties dialog, select the **Tooltips** tab and then click the checkbox for "Enable Variable Mouseover Values".

FIGURE 3-5: EXAMPLE BREAKPOINT



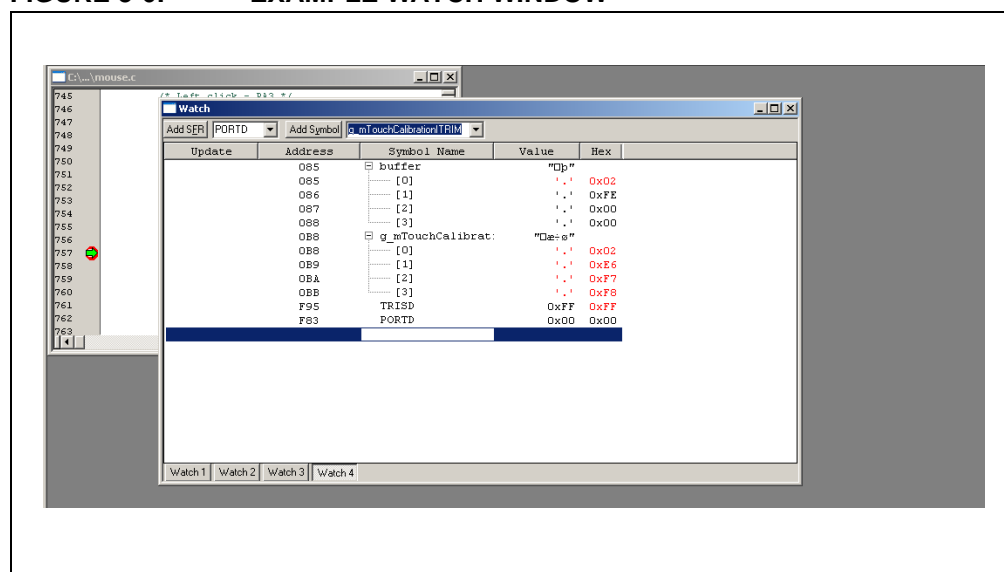
3.3.3 Using Watch Windows

To use a Watch window:

- The Watch window is made visible on the desktop by selecting **View>Watch**. It contains four selectable watch views (via tabs) in which to view variables (SFRs, symbols and absolute addresses).
- Select an SFR or symbol from the list and click the related **Add** button to add it to the Watch window, or click in the “Address” column and enter an absolute address.

A Watch window populated with the SFRs and Symbols will look like Figure 3-6. For more on using Watch windows, see MPLAB IDE Help.

FIGURE 3-6: EXAMPLE WATCH WINDOW



3.4 PROGRAMMING THE DEBUGGED APPLICATION

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

1. Disable the starter kit as a debug tool by selecting *Debugger>Select Tool>None*.
2. Select the starter kit as the programmer in the *Programmer>Select Programmer* menu.
3. Select *Programmer>Program*.

Now, the starter kit will run independently.

3.5 CREATING OTHER APPLICATIONS

This starter kit is just one way to use Microchip PIC18 microcontrollers in an application. Other tools and resources exist to support these devices:

- PIC18 Demo Boards, such as the Low Pin Count USB Development Kit, PICDEM FS USB, PIC18F87J50 FS USB Demonstration Plug-In Module, PIC18F46J50 FS USB Demonstration Plug-In Module
- MPLAB C Compiler for PIC18 MCUs, Full Version: More optimization options than the student version for full-scale development.
- Application Notes: Libraries and example applications with source code for using the PIC18 family of microcontrollers.

For more information on demo boards on application notes, see the design center at the Microchip web site (www.microchip.com). For additional information on the full version of the MPLAB C Compiler for PIC18 MCUs, go to <http://www.microchip.com/c18>. For additional information on Microchip USB software and tools, go to <http://www.microchip.com/usb>.

3.6 DETERMINING DEVICE SUPPORT

Due to the built-in in-circuit debugging capability of ICD devices and the In-Circuit Serial Programming™ (ICSP™) function offered by the debugger, the starter kit uses some on-chip resources when debugging. It also uses program memory and file register locations in the target device during debugging. These locations are not available for use by user code. In the MPLAB IDE; registers marked with an “R” in the register displays represent reserved registers.

3.7 RESERVED RESOURCES

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

3.8 DEBUG VERSION INFORMATION

Selecting either *Debugger>Settings* or *Programmer Settings* from the menu bar opens the Starter Kit Settings dialog. Currently, there is only one (**Info**) tab on this dialog, displaying the following information:

- Firmware Version: The version of firmware on the starter kit board.
- Debug Exec Version: The version of the debug executive that is loaded into the PIC18 device program memory to enable debug operation.

3.9 TROUBLESHOOTING

3.9.1 Debug Connection Problems

While using the starter kit as a debugger, you may get the error, “Unable to Enter Debug Mode”, when programming the device. This can result from communication being lost between the starter kit and MPLAB IDE. To resolve this:

1. Unplug the USB cable from the starter kit.
2. Plug the USB cable back into the starter kit.

MPLAB IDE should automatically reconnect to the starter kit. If this does not work, do the following:

1. Check the USB connection between the PC and starter kit at both ends.
2. If using a USB hub, make sure it is powered.
3. Make sure the USB port is not in use by another device.

3.9.2 Programming Problems

If during the course of developing your own application, you can no longer program the device on the starter kit, you may have set device Configuration bits to code-protect or some other state that prevents programming. To view the settings of the Configuration bits, select Configure>Configuration Bits.

Chapter 4. Hardware

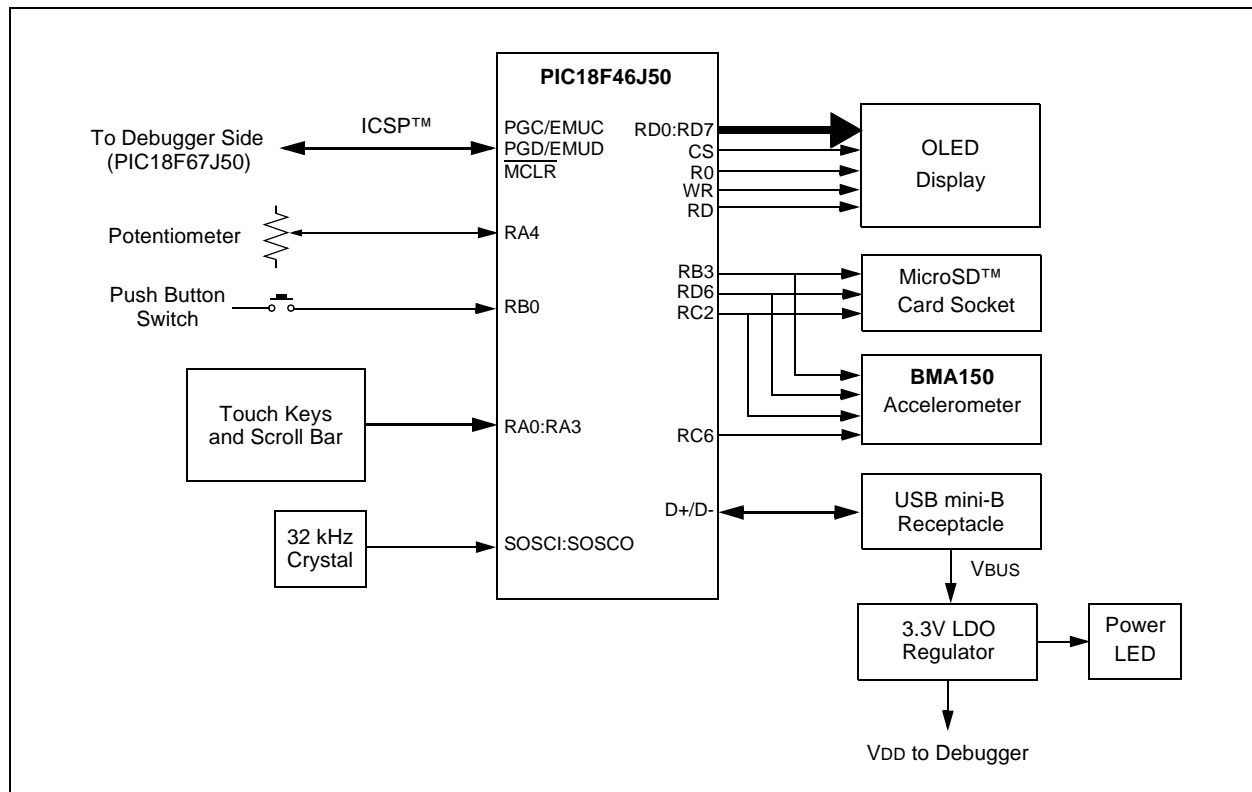
This chapter provides a functional overview of the MPLAB Starter Kit for PIC18F, and identifies the major hardware components. Topics covered include:

- Application Functional Overview
- Programmer/Debugger Functional Overview
- Board Components

4.1 APPLICATION FUNCTIONAL OVERVIEW

Figure 4-1 illustrates the main functions of the starter kit:

FIGURE 4-1: APPLICATION SIDE BLOCK DIAGRAM



The application side of the starter kit is centered on the PIC18F46J50 microcontroller, which requires very little additional hardware to perform its tasks. The MPLAB Starter Kit for PIC18F comes preprogrammed with a MicroSD bootloader that has the ability to load, program and run precompiled applications from a MicroSD memory card to the device's Flash program memory. The MicroSD bootloader uses substantial parts of the Microchip Memory Disk Drive file system. The precompiled demos use substantial parts of the Microchip USB Stack Library.

The applications accept user inputs from three sources: the capacitive touch pad, S1 (A5), the acceleration sensor, U6 (A4), and the potentiometer, R24 (A7).

The microcontroller uses one of its A/D Converter channels to sample and convert the potentiometer's value to a digital value. Four additional A/D channels are used to monitor the directional touch pads of S1. The values from these channels are analyzed with the CTMU to determine when a touch-and-release event occurs on any of the pads. The application firmware determines which action to take based on the application's current context. For example, the capacitive buttons are used as **Left** and **Right** buttons for the USB HID Mouse Demo, while the capacitive slider is used to emulate the scroll wheel. The capacitive slider is also used by the USB HID Joystick Demo to generate the Z axes coordinates.

The microcontroller uses the SPI interface to configure the BMA150 acceleration sensor, then to read 3-axis acceleration and temperature data in digital format. The acceleration data is used to compute the board's tilt; this is converted to cursor movements for the USB HID Mouse Demo and X-Y coordinates for the USB HID Joystick Demo.

The microcontroller uses the same SPI interface to communicate with the MicroSD memory card. This allows the USB MSD Card Reader Demo to read and write files to any MicroSD memory card.

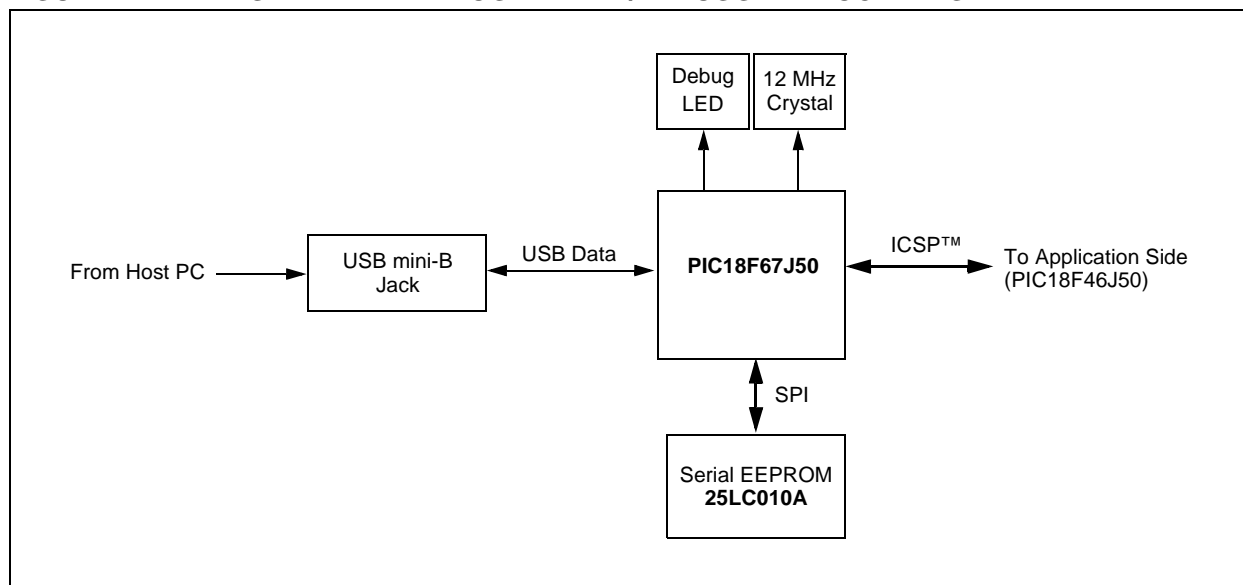
The microcontroller directly drives the OLED display (LED1) using the microcontroller's Parallel Master Port (PMP). Data is driven on I/O ports, RD<7:0>. Port control signals, A0, RD, WR and CS, are provided on ports RB5 and RE<2:0>, respectively. A DC boost circuit, comprised of power MOSFET Q4 along with D5 and L3, provides the operating voltage for the OLED.

The microcontroller uses its on-chip USB engine and transceiver to communicate with the mini-B receptacles on the application side.

4.2 PROGRAMMER/DEBUGGER FUNCTIONAL OVERVIEW

Figure 4-2 illustrates the debugging/programming operation of the starter kit.

FIGURE 4-2: STARTER KIT PROGRAMMER/DEBUGGER BLOCK DIAGRAM



In its default configuration, the starter kit functions as a USB bus-powered device. Power is provided via the USB cable; the nominal 5 volt unregulated supply is regulated by a Microchip MC1727 3.3 volt, Low Dropout (LDO) linear regulator. Proper main system power is indicated by the green LED (D2).

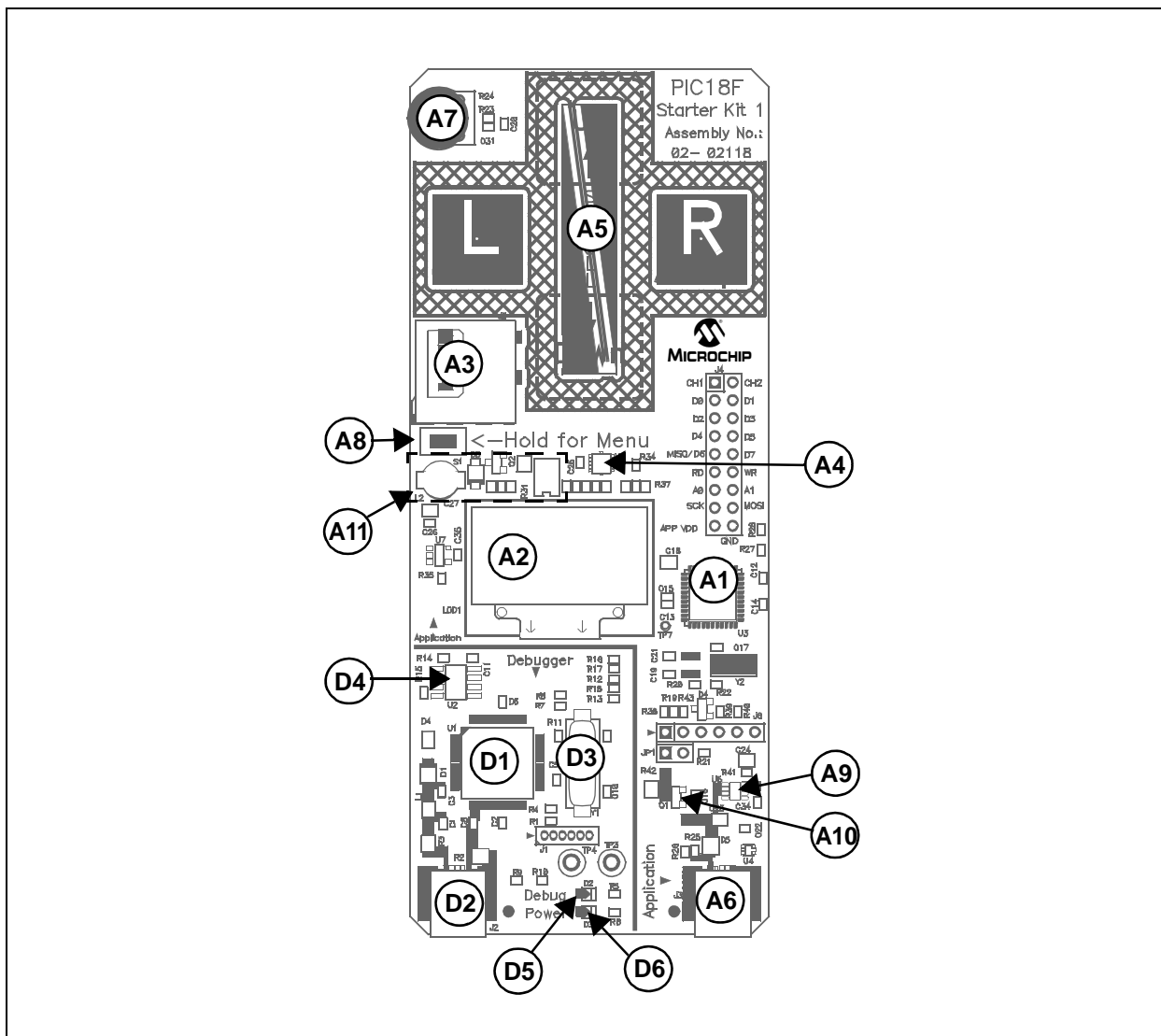
The debugging and programming side of the starter kit is controlled by a PIC18F67J50 microcontroller running at 48 MHz. The PIC18F67J50's built-in USB engine provides the communications interface between the starter kit and the host PC. The microcontroller manages debugging or programming of the target PIC18F46J50 by controlling the target's MCLR, PGC1/EMUC1 and PGD1/EMUD1 signals. Target clocking is provided by the PIC18F67J50.

A Microchip 25LC010A serial EEPROM is used to store the starter kit's serial number and debug control information.

4.3 BOARD COMPONENTS

Figure 4-3 identifies the key hardware components for the starter kit.

FIGURE 4-3: PIC18F STARTER KIT COMPONENT LAYOUT



MPLAB Starter Kit for PIC18F User's Guide

TABLE 4-1: PIC18F STARTER KIT COMPONENT DESCRIPTIONS

Ref	Debug/Programmer Component	Ref	Application Component
D1	PIC18F67J50 Microcontroller (U1)	A1	PIC18F46J50 Microcontroller (U3)
D2	mini-B USB Receptacle (J2)	A2	OLED Display (LED1)
D3	Debugger/Programmer Clock Crystal (Y1)	A3	MicroSD™ Card Socket (J5)
D4	25LC010A Serial EEPROM (U2)	A4	BMA150 Acceleration Sensor (U6)
D5	Debug LED (D2)	A5	Capacitive Touch Pad (S1)
D6	Power LED (D3)	A6	mini-B USB (Peripheral) Receptacle (J3)
		A7	Potentiometer (R24)
		A8	Bootloader Reset Push Button Switch (S1)
		A9	MCP1727 Voltage Regulator (U5)
		A10	Soft Start Circuit (Q1)
		A11	OLED Voltage Boost Circuitry

4.3.1 Programmer/Debugger Components

The components listed here (in order of their reference tags in Figure 4-3) are the key components of the programmer/debugger side of the starter kit:

- D1. **PIC18F67J50 Microcontroller (U1):** Controls the programming/debugging operations of the target PIC18F46J50 microcontroller. It also provides the 12 MHz clock for the application side of the microcontroller.
- D2. **mini-B USB Receptacle (J2):** Provides system power and bidirectional communication between the host PC and starter kit.
- D3. **Debugger/Programmer Clock Crystal (Y1):** Provides an accurate 12 MHz frequency reference for the PIC18F67J50 microcontroller for stable USB operations in Programming and Debugging modes. The PIC18F67J50 also uses this to generate a second 12 MHz clock for use by the PIC18F46J50 microcontroller.
- D4. **25LC010A Serial EEPROM (U2):** Store the starter kit's serial number and debug control information.
- D5. **Target Power LED (D2):** When lit, indicates that power is being supplied to the starter kit from VBUS, either from the Application or Programmer/Debugger side.
- D6. **Debug LED (D3):** When lit, indicates that the starter kit is operating in Programming or Debug mode.

4.3.2 Application Components

The components listed here (in order of their reference tags in Figure 4-3) are the key components of the application side of the starter kit:

- A1. **PIC18F46J50 Microcontroller (U3):** This provides the processing power for the demo applications and application development on the starter kit. The microcontroller features 64 Kbytes of Flash program memory and 3.7 Kbytes RAM. The demo application uses an external 12 MHz signal from the programmer side as clock source. Custom applications that do not use the USB module may also use the microcontroller's on-chip FRC oscillator as a clock source. (USB applications must use the 12 MHz programmer clock source, as the tolerance of the FRC oscillator exceeds USB specifications.)
- A2. **OLED Display (LED1):** A 128 x 64 pixel, monochrome organic LED array provides a wide range of graphics and alphanumeric display options.
- A3. **MicroSD Card Socket (J5):** A standard socket that is read and write compatible with all MicroSD cards. The socket and the application side of the microcontroller communicate over an SPI interface.

- A4. **BMA150 Acceleration Sensor (U6):** This device implements a complete 3-axis accelerometer and temperature sensor solution in an 8-lead QFN package. The accelerometer is a semiconductor-based strain gauge with no moving parts and never requires calibration. All acceleration and temperature data is provided in real time in digital format, compatible with standard SPI interfaces.
- A5. **Capacitive Touch Pads (S1):** This 5-element keypad is an integral part of the board's top layer. The application responds to capacitive changes induced by direct contact with the user by monitoring each of the touch pad's elements with the microcontroller's integrated CTMU module.
- A6. **mini-B USB Receptacle (J5):** Physically identical to J1, this provides USB connectivity for the application side when it is functioning as a peripheral device.
- A7. **Potentiometer (R24):** Provides a variable analog input to the microcontroller for certain demo applications.
- A8. **Bootloader Reset Push Button (S1):** The switch provides a hardware signal interpreted by the main application that is used to reset the application and invoke the MicroSD bootloader.
- A9. **OLED Voltage Boost Circuitry:** L3, D5 and Q4 (along with other passive components) are connected in a voltage boost topology. This provides the OLED display with its nominal operating voltage of 13 Vdc.
- A10. **Soft Start Circuit:** Transistor Q1 and its associated passive components form a PWM controlled regulator. This is used to keep the non-microcontroller portions of the application within the Inrush and Suspend Current limitations spelled out in the USB specification. See **Appendix B. "USB Current Draw Requirements"** for more information.
- A11. **MCP1727 Voltage Regulator (U5):** The 3.3V linear regulator regulates the USB unregulated voltage to 3.3 volts (with respect to Vss) and supplies the starter kit with system power.

NOTES:

Appendix A. Starter Kit Schematics

The following schematic diagrams are included in this appendix:

Programmer/Debugger:

- Figure A-1: Programmer/Debugger System

Application:

- Figure A-2: Application Microcontroller, Touch Switches and Associated Components
- Figure A-3: OLED Display, LED and Application Side USB Connectors

FIGURE A-1: STARTER KIT SCHEMATIC, SHEET 1: PROGRAMMER/DEBUGGER SYSTEM

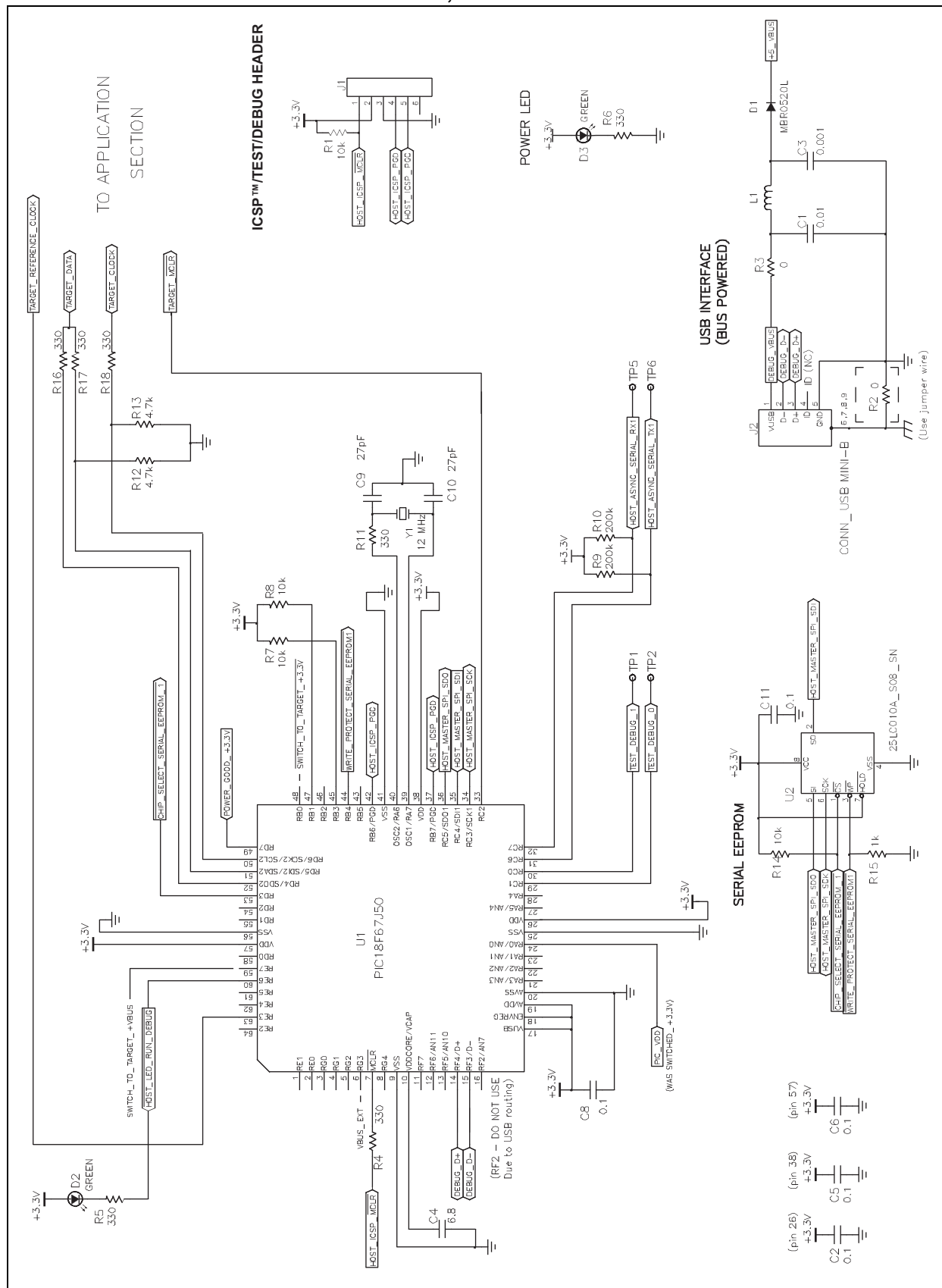


FIGURE A-2: STARTER KIT SCHEMATIC, SHEET 2: APPLICATION MICROCONTROLLER, SOFT START POWER CIRCUIT, ACCELEROMETER AND SWITCHES

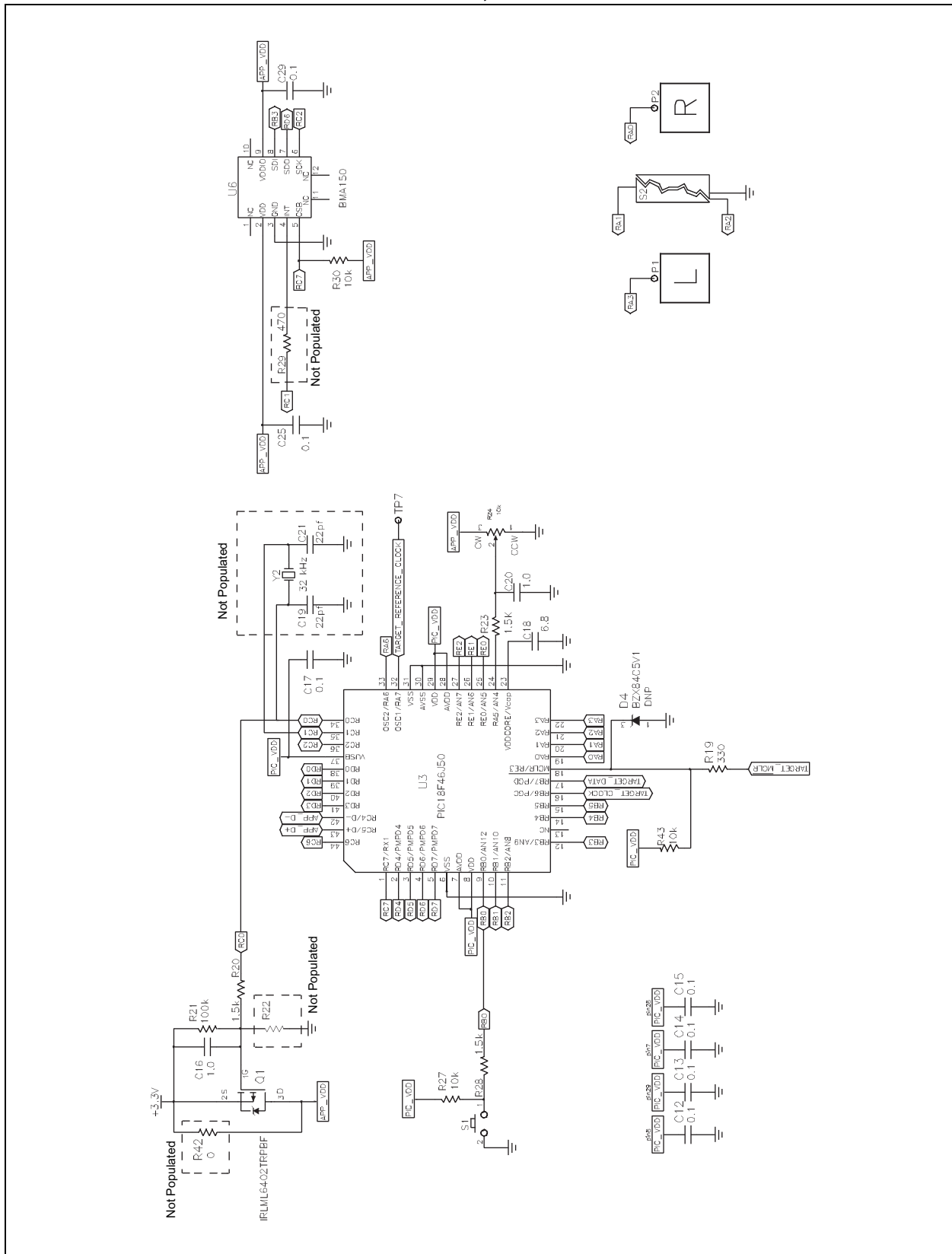
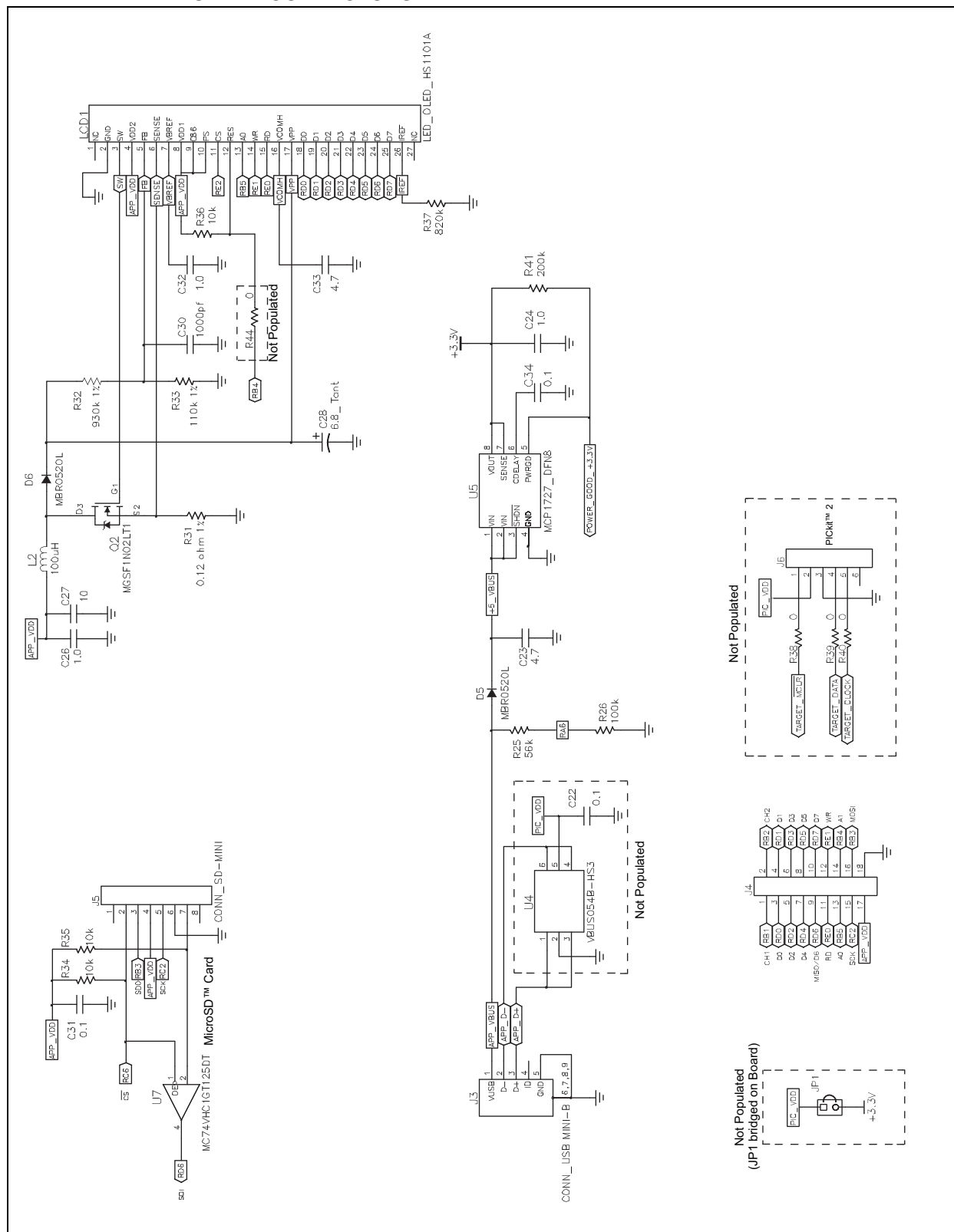


FIGURE A-3: STARTER KIT SCHEMATIC, SHEET 3: OLED DISPLAY, SD INTERFACE, APPLICATION SIDE USB CONNECTOR AND POWER SUPERVISOR, AND OTHER CONNECTORS



Appendix B. USB Current Draw Requirements

This chapter describes the key USB specification current limits and provides a functional overview of the soft start circuitry used by the MPLAB Starter Kit for PIC18F to meet them. The appendix includes the following sections:

- USB Current Specifications
- PIC18F Starter Kit Soft Start Circuitry
- Soft Start Functions and Usage
- Suspend Current Limitations with the Starter Kit

B.1 USB CURRENT SPECIFICATIONS

There are three major USB current specifications that the soft start circuitry addresses: inrush current during initial plug in, total application current draw, and suspend current limitations.

B.1.1 Inrush Current

All USB devices are required to limit their inrush draw from the bus such that there is a capacitance change of no more than 10 μ F for the device. (Since the USB is a 5V bus, this equates to a charge inrush of 50 μ C.) Some applications, like the OLED on the starter kit, may cause a capacitance change of greater than 10 μ F. To meet the specification, the application can use current management techniques to limit the inrush current and spread the charge over a longer period of time.

B.1.2 Total Application Current Draw

USB devices are limited to 100 mA of total current draw during the enumeration phase. The enumeration phase is the time during which the device is first attached to the bus and when the host sends a `SET_CONFIGURATION` command to the device. Devices may request for additional current by changing the `bMaxPower` parameter of the configuration descriptor. The configuration descriptor can be found in the file, `usb_descriptors.c`, in each of the example demos. More information about the `bMaxPower` parameter and the configuration descriptor, in general, can be found in Table 9-10 of the USB specification.

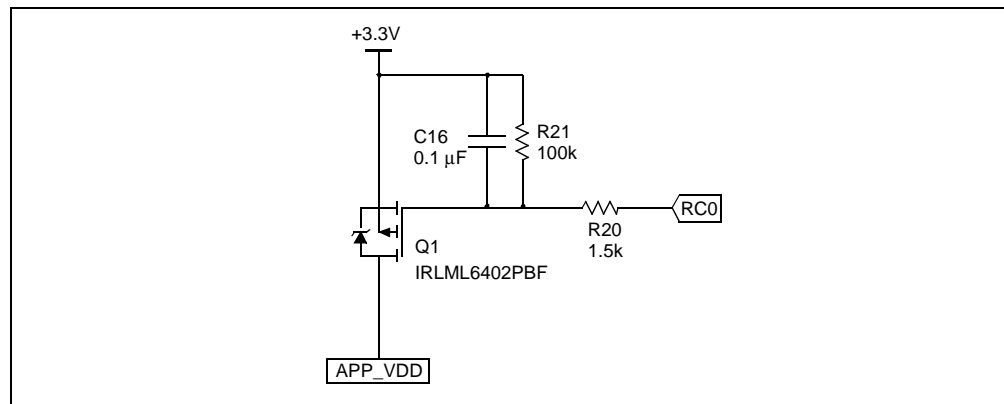
B.1.3 Suspend Current Limitations

When the USB bus is in Suspend mode, all attached USB devices are required to use less than 2.5 mA of current draw from the bus. For applications that have features that require more current than is allowed, there are several options available. The first is to be a self-powered device and only draw current from the power supply. The second is to implement a software controlled method for reducing the current to below the 2.5 mA limit.

B.2 PIC18F SOFT START CIRCUITRY

One strategy to meet the key USB current specifications is to include a software controllable soft start mechanism. The starter kit implements the simple soft start circuit shown in Figure B-1.

FIGURE B-1: PIC18F STARTER KIT SOFT START CIRCUIT



Transistor, Q1, a power MOSFET, blocks the current from the 3.3V power rail to the non-microcontroller portions of the application circuit (e.g., SD card, OLED, etc). R21 ensures that on initial power-up, Q1 is blocking the current flow. R20 and C16 form an RC filter on the digital output from the microcontroller (RC0 port). This RC filter creates an analog voltage on the gate of Q1 from the PWM signal supplied by the microcontroller. The values of C16 and R20 are chosen to produce a large RC time constant compared to the PWM frequency, and such that, the current drawn from the pin is within the port pin's specification.

The PWM frequency is based on the RC network and is chosen to produce a fairly constant voltage on the gate of Q1. The PWM duty cycle is determined by the gate threshold of Q1. Once the frequency is selected, changing the PWM duty cycle will result in a change of the gate voltage. The duty cycle is set so that the starting voltage on the gate of Q1 is above the threshold voltage across all device variations, so that the device is in an OFF state. The duty cycle is varied at a rate in which the gate voltage on Q1 changes slowly enough to ensure that the transistor stays in its linear region during the full charging of Q1's output (Figure B-2). The result is that voltage is supplied to the application at a slower rate, thus lowering the inrush current.

The duty cycle should stop sweeping at a point that is lower than any possible value for Q1's threshold voltage; this ensures that the entire linear region is used for as long as possible. Once this point is reached, the software can disable the PWM and set the output of RC0 low; this puts Q1 in Full Saturation mode.

Figure B-2 and Figure B-3 show how the change in gate voltage (yellow trace) relates to the change in the circuit's output (blue trace). Figure B-2 captures a close look at the linear region, while Figure B-3 captures the entire PWM cycle from no output to full output. Note that the gate voltage decreases slowly and constantly over the entire potential threshold voltage range of Q1.

USB Current Draw Requirements

FIGURE B-2: VARYING PWM OUTPUT TO SLOWLY CROSS THE LINEAR REGION OF Q1'S OUTPUT

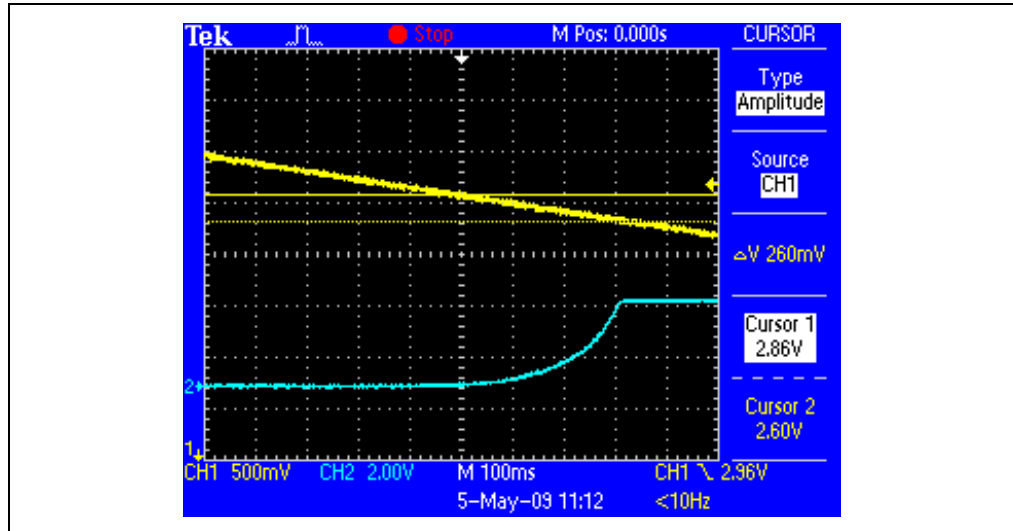
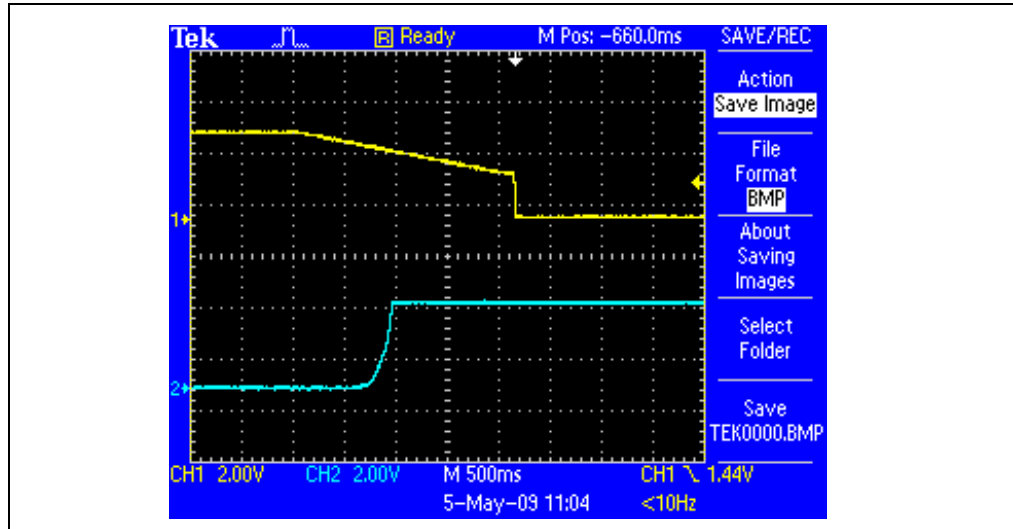


FIGURE B-3: THE ENTIRE SOFT START PROCESS



B.3 SOFT START FUNCTIONS AND USAGE

A software component on the microcontroller side is also required to manage the PWM output and control the external circuit. The demo applications supplied with the starter kit use three different software function calls to manage USB-related power consumption. This section describes the functions provided and how they are used to meet the USB requirements.

B.3.1 AppPowerEnable()

The `AppPowerEnable()` function starts the soft start sequence. The function initializes the port pins, timers and PWM modules to get ready for the soft start. It is optional if `APP_POWER_AUTO_ENABLE` is defined.

In light of the Total Application Current Draw specification of 100 mA, this function should not be called until after the device is enumerated. See the following section for more details.

B.3.2 AppPowerReady()

The AppPowerReady() function returns a BOOL data type that will return TRUE if the soft start is complete or FALSE if it is not. This function should be called periodically during the soft start to insure that the soft start continues ramping properly.

If APP_POWER_AUTO_ENABLE is defined, AppPowerReady() will automatically call AppPowerEnable() if it is not already started. In light of the Total Application Current Draw specification of 100 mA, this function should not be called until after the device is enumerated. The application code that relies on components that are gated by the soft start should wait until AppPowerReady() returns TRUE before running.

EXAMPLE B-1: USING AppPowerReady() TO CONTROL APPLICATION START-UP

```
// User Application USB tasks
if((USBDeviceState < CONFIGURED_STATE) || (USBSuspendControl==1))
    return;

// Soft Start the APP_VDD
if(AppPowerReady() == FALSE) return;

// The rest of the application code down here
// (once power is enabled).
```

B.3.3 AppPowerDisable()

The AppPowerDisable() function turns the power off to the application. This function should be used when the device is going to go into Suspend mode in order to meet the USB suspend current specification.

EXAMPLE B-2: USING AppPowerDisable()

```
void USBCBSuspend(void)
{
    //...
    AppPowerDisable();
    //...
    Sleep();
    //...
}
```

After the device resumes from suspend, the AppPowerEnable() and AppPowerReady() functions are used again to restart the application power. In this demo, that is done through the same calls in the ProcessIO() function that initially powered the device on start-up.

B.4 USB SUSPEND CURRENT LIMITATIONS FOR THE STARTER KIT

The USB specification limits devices to 2.5 mA of current draw from the bus while a device is in a suspended state. This limit is tested during compliance testing for all USB certified devices.

The design of the PIC18F Starter Kit utilizes an on-board debugger. To optimize the starter kit for cost and available features, the debugger section provides the system clock for both the debugger and the application portions of the circuit. As a result, the debugger is always running, even when the debugger USB cable is not attached, resulting in a current draw higher than allowed during USB suspend. In contrast, typical applications will have their own clock sources and be able to better control their own current consumption to meet the USB suspend specification. This can be achieved by switching to a lower clock frequency or placing the microcontroller into Sleep mode, among other possible methods.

MPLAB Starter Kit for PIC18F User's Guide

NOTES:

Index

A

Acceleration Sensor	11, 12, 24
Application Components	
Soft Start Circuit	27
Application Components	
Acceleration Sensor	27
Bootloader Reset Push Button	27
Capacitive Touch Pad	27
MicroSD Card Socket	26
mini-B USB Receptacle	27
OLED Display	26
OLED Voltage Boost Circuitry	27
PIC18F46J50 Microcontroller	26
Potentiometer	27
Voltage Regulator	27

B

Block Diagrams	
Application Side	23
Programmer/Debugger Side	24
Soft Start Circuit	34
Board Components	
Debug LED	26
Debugger/Programmer Clock Crystal	26
mini-B USB Connector	26
PIC18F67J50 Microcontroller	26
Serial EEPROM	26
Target Power LED	26
Bootloader	
Display Messages	10
Memory Map	11
Writing Applications	11

C

Charge Time Measurement Unit (CTMU)	24
Code Examples	
Using AppPowerDisable()	36
Using AppPowerReady() to	
Control Application Start-up	36
Customer Change Notification Service	4
Customer Support	5

D

Debug	7
Executive	17
Reserved Resources	21
Setup	16

Debugger Toolbar

Animate	18
Halt	18
Reset	18
Run	18
Step Into	18
Step Over	18

Demonstration Applications

Bootloader from MicroSD Card	9
HID Custom Device	13
HID Joystick	12
HID Mouse	11
USB CDC COM Port	14
USB MSD Card Reader	14

Documentation

Conventions	2
Layout	1

H

Host Computer Requirements	7
----------------------------------	---

I

Initial Board Setup	7
Connecting Hardware	8
Installing Software	8
Internet Address	4

M

Microchip Internet Web Site	4
MPLAB IDE	7, 15
mTouch Calibration (manual)	14

P

Programmer	21
------------------	----

R

Reading, Recommended	3
Reserved Resources	21
Revision History	5

S

Schematic Diagrams	29
Soft Start	
Circuitry	34
Functions, Usage	35
Starter Kit	
Board Components	25

T

Troubleshooting 22

U

Unable to Enter Debug Mode 22

USB

Current Specifications..... 33

Suspend Current Limitations 37

W

Warranty Registration 2

Watch Window 18

WWW Address..... 4

NOTES:



WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office

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

Atlanta

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

Boston

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

Chicago

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

Dallas

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

Detroit

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

Kokomo

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

Los Angeles

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

Santa Clara

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

Toronto

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

ASIA/PACIFIC

Asia Pacific Office

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

Australia - Sydney

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

China - Beijing

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

China - Chengdu

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

China - Hong Kong SAR

Tel: 852-2401-1200
Fax: 852-2401-3431

China - Nanjing

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

China - Qingdao

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

China - Shanghai

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

China - Shenyang

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

China - Shenzhen

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

China - Wuhan

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

China - Xiamen

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

China - Xian

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

China - Zhuhai

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

ASIA/PACIFIC

India - Bangalore

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

India - New Delhi

Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune

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

Japan - Yokohama

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

Korea - Daegu

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

Korea - Seoul

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

Malaysia - Kuala Lumpur

Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang

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

Philippines - Manila

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

Singapore

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

Taiwan - Hsin Chu

Tel: 886-3-572-9526
Fax: 886-3-572-6459

Taiwan - Kaohsiung

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

Taiwan - Taipei

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

Thailand - Bangkok

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

EUROPE

Austria - Wels

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

Denmark - Copenhagen

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

France - Paris

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

Germany - Munich

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

Italy - Milan

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

Netherlands - Drunen

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

Spain - Madrid

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

UK - Wokingham

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