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

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

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

Contact us

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

MPLAB® Starter Kit for PIC24H Microcontrollers 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.

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.

MPLAB® STARTER KIT FOR PIC24H MICROCONTROLLERS USER'S GUIDE

Table of Contents

NOTES:

MPLAB® STARTER KIT FOR PIC24H MICROCONTROLLERS USER'S GUIDE

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 PIC24H Microcontrollers (MCUs). 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 PIC24H MCU device's processing capabilities. The manual layout is as follows:

- **Chapter 1. Introduction –** This chapter introduces the starter kit and provides an overview of its features.
- **Chapter 2. Starter Kit Demo** – This chapter describes how to use the starter kit demo software.
- **Chapter 3. Develop 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. Schematics** – This appendix provides detailed schematic diagrams of the starter kit.

CONVENTIONS USED IN THIS GUIDE

This manual 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) { ... }`

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 PIC24H MCUs. 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 Microchip tools, read the associated Readme files (HTML files) included with the software.

PIC24H MCU Documentation

For the most up-to-date information on PIC24H MCU devices (data sheets, errata, family reference manual, etc.), please see the Microchip web site at: http\\www.microchip.com.

MPLAB® C Compiler for PIC24 MCUs and dsPIC® DSCs User's Guide (DS51284)

This document helps you use Microchip's 16-bit C compilers to develop your application. The compilers are the MPLAB C Compiler for dsPIC DSCs and PIC24 MCUs, the MPLAB C Compiler for dsPIC DSCs (subset of the first), and the MPLAB C Compiler for PIC24 MCUs (subset of the first). These compilers are GNU-based language tools, based on source code from the Free Software Foundation (FSF). For more information about FSF, see www.fsf.org.

MPLAB® Assembler, Linker and Utilities for PIC24 MCUs and dsPIC® DSCs User's Guide (DS51317)

This document helps you use Microchip Technology's 16-bit language tools based on GNU technology. The language tools discussed are the MPLAB Assembler for dsPIC DSCs and PIC24 MCUs, MPLAB Object Linker for dsPIC DSCs and PIC24 MCUs, MPLAB Archiver/Librarian for dsPIC DSCs and PIC24 MCUs and other 16-bit device utilities.

MPLAB® IDE User's Guide (DS51519)

This document describes how to use the MPLAB IDE integrated development environment, as well as the MPLAB Project manager, MPLAB 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, listings of Microchip sales offices, distributors and factory representatives

DEVELOPMENT SYSTEMS CUSTOMER CHANGE NOTIFICATION SERVICE

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

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

The Development Systems product group categories are:

- **Compilers** – The latest information on Microchip C compilers and other language tools. These include 16-bit language tools:
  - MPLAB C Compiler for dsPIC DSCs and PIC24 MCUs (and subsets)
  - MPLAB Assembler for dsPIC DSCs and PIC24 MCUs
  - MPLAB Object Linker for dsPIC DSCs and PIC24 MCUs
  - MPLAB Archiver/Librarian for dsPIC DSCs and PIC24 MCUs

  and 8-bit language tools:
  - MPLAB C Compiler for PIC18 MCUs
  - MPASM™ Assembler
  - MPLINK™ Object Linker
  - MPLIB™ Object Librarian
- **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, MPLAB ICD 3, and PICkit™ 2.
- **MPLAB IDE** – The latest information on Microchip MPLAB IDE, the Windows® Integrated Development Environment for development systems tools. This list is focused on the MPLAB IDE, MPLAB IDE Project Manager, MPLAB Editor and MPLAB SIM simulator, as well as general editing and debugging features.
- **Programmers** – The latest information on Microchip programmers. These include the MPLAB PM3 device programmer and the PICSTART® Plus 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 (November 2008)

This is the initial release of this document.

NOTES:

Chapter 1. Introduction

Thank you for purchasing Microchip Technology's MPLAB® Starter Kit for PIC24H Microcontrollers (MCUs). This kit is intended to introduce and demonstrate the features of the starter kit and the strong processing capabilities of PIC24H MCU devices.

The starter kit demonstrates a low-cost hardware and software solution for processing sensor signals and interfacing audio and visual displays. The board also includes signal conditioning circuitry, which helps users to perform a quick evaluation of the power of PIC24H MCUs by processing signals coming from an external sensor. In addition, the starter kit has on-board debug circuitry you can use to develop and debug your own applications without using other debug tools (i.e., in-circuit emulator or debugger).

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

- Overview
- Operational Requirements
- Board Setup

1.1 OVERVIEW

The MPLAB Starter Kit for PIC24H MCUs connects directly to the USB port on a personal computer (PC). The PC USB connection supplies communications and power to the board.

The starter kit includes debug and programmer circuitry that allows applications to be programmed onto the board's PIC24H MCU device and then debugged, all using MPLAB IDE.

The sensor signals from an on-board triaxial analog accelerometer are routed to the fast on-chip ADC module in the PIC24H MCU for software processing. This feature allows the Starter Kit to be maneuvered by tilting. Based on the sensed acceleration due to the tilting of the board, a visual output is generated on the on-board OLED display controlled by the Parallel Master Port (PMP) module. Additionally, the speech segments are audibly produced through the on-board speaker via the output compare module as a Pulse-Width Modulated (PWM) digital waveform. This output is converted to an analog speech signal by a low-pass filter on the starter kit board. Alternatively, applications can use the conditioning circuitry to plug-in a wide range of analog sensors, grab the sensor signals through the ADC and process them.

In addition to the Recommended Reading listed in the Preface, the following manufacturers' data sheets are also recommended as reference sources:

- Bosch Sensortec Data Sheet, *BMA140 Triaxial Analog Acceleration Sensor*
- National Semiconductor Corporation Data Sheet, *LM4853 Boomer® Audio Power Amplifier Series Mono 1.5W/ Stereo 300mW Power Amplifier* (DS200334)

1.2 OPERATIONAL REQUIREMENTS

To communicate with and program the MPLAB Starter Kit for PIC24H MCUs, the following hardware and software requirements must be met:

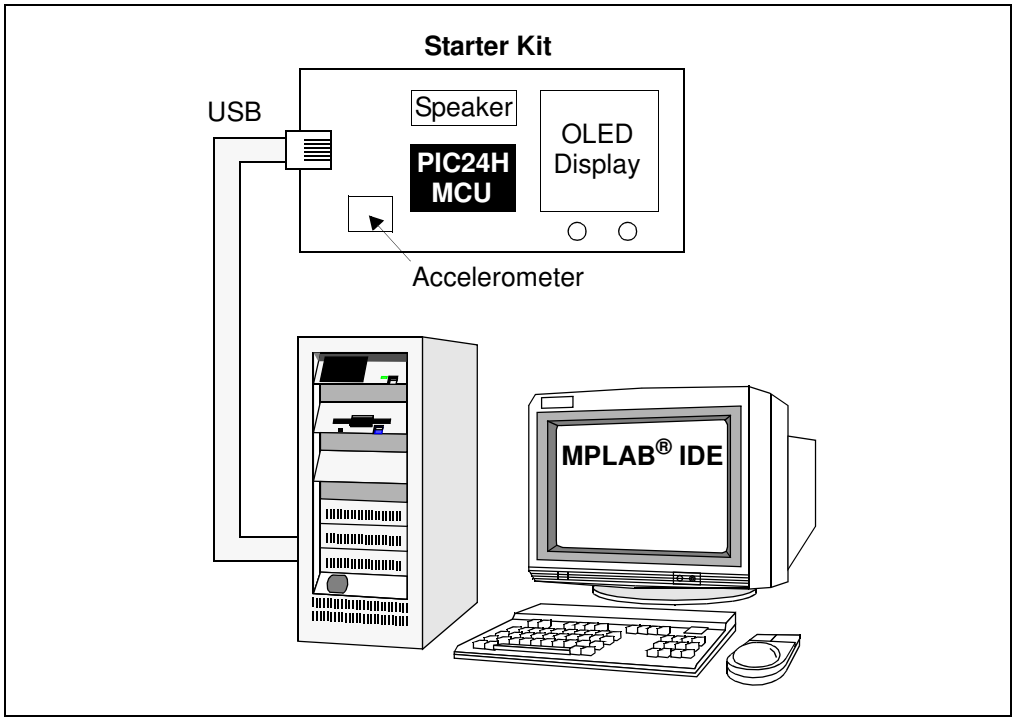
- PC-compatible system
- An available USB port on the PC or a powered USB hub
- CD-ROM drive
- Windows® 2000 SP4, Windows XP SP2, or Windows Vista (32-bit)(1) Operating Systems.

Note 1: Only initial testing has been performed on the 32-bit Windows Vista Operating System for this release. The 64-bit Windows Vista Operating System is not supported at this time.

1.3 BOARD SETUP

Figure 1-1 shows the setup for the MPLAB Starter Kit for PIC24H MCUs. The USB connection provides communication and power to the board. As soon as the starter kit is powered through the USB cable, the preloaded demonstration starts.

FIGURE 1-1: MPLAB STARTER KIT FOR PIC24H MCUs SETUP



Chapter 2. Starter Kit Demo

This chapter describes the MPLAB® Starter Kit for PIC24H Microcontrollers demonstration that is preloaded on the PIC24H MCU device, which showcases the multitasking of accelerometer sensing, the OLED display, speech playback, and the switch press monitor. This software application demonstrates how to use the MPLAB Starter Kit for PIC24H MCUs for signal capture and processing of the sensor signal, speech decoding and playback, and controlling the OLED display.

A detailed explanation of the starter kit hardware is provided in **Chapter 4. "Hardware"**.

Topics covered include:

- Running the Demo
- Understanding the Demo
- Other Demo Code Examples

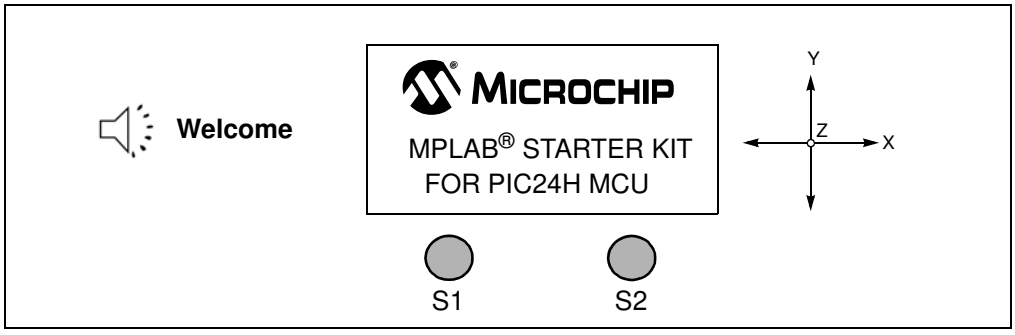
2.1 RUNNING THE DEMO

To run the demo, follow these steps:

1. Power-up the starter kit by connecting the board to the USB port of a computer. You should briefly see a pop-up message in the system tray that states (1) new hardware has been found, (2) drivers are being installed, and (3) new hardware is ready for use. If you do not see these messages and the starter kit does not work, try reconnecting the USB cable. If reconnecting the USB cable does not work, refer to **Section 3.8 "Troubleshooting"**.

   When powered up, an audible welcome message is played while simultaneously displaying the Start-up screen on the OLED display (see Figure 2-1).

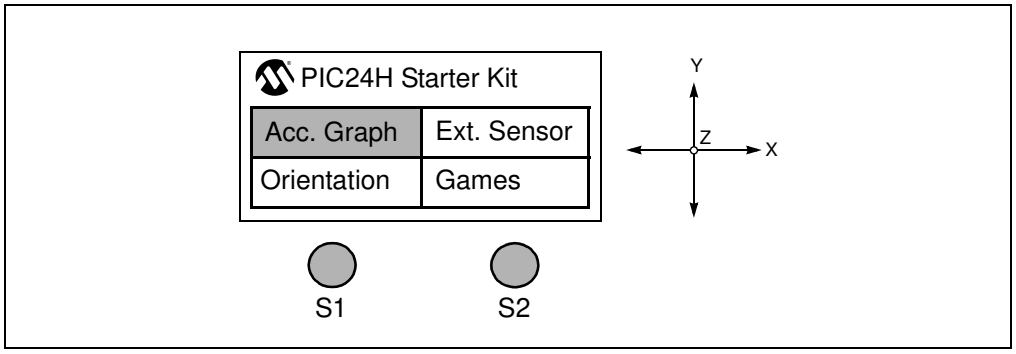
FIGURE 2-1: START-UP SCREEN



After the start-up exercise, the Home screen appears on the OLED display, as shown in Figure 2-2.

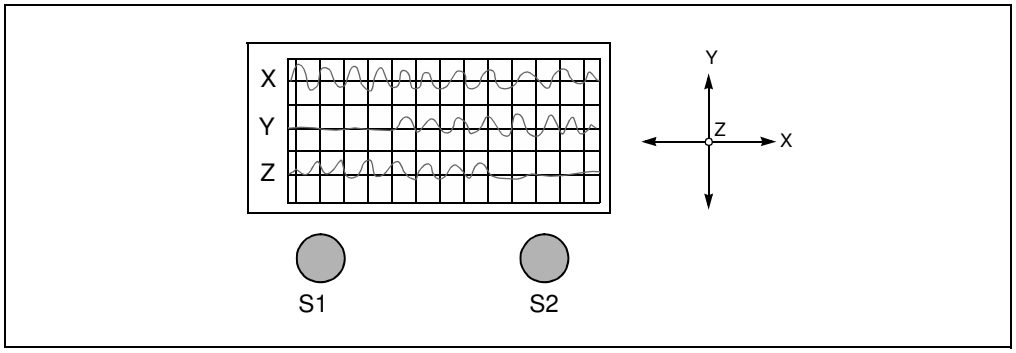
The Home screen has four cells from which to choose: Accelerometer (Acc.) Graph, External (Ext.) Sensor, Orientation and Games. The starter kit can be maneuvered by tilting it about the X and Y axes. Based on the direction of tilt, one of the four cells is highlighted for selection. Switch S1 or S2 can be pressed to select the highlighted cell.

FIGURE 2-2: STARTER KIT HOME SCREEN



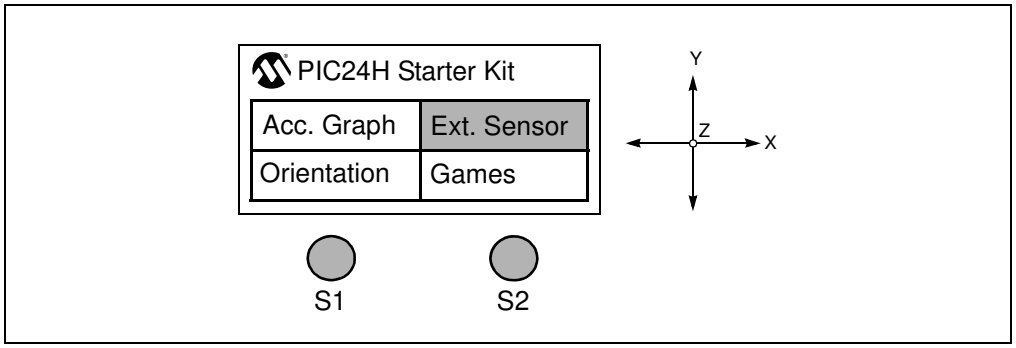
2. Selecting the **Acc. Graph** cell starts the application, which captures the triaxial outputs of the accelerometer and displays them on the OLED display as a graph, as shown in Figure 2-3. Switch S1 or S2 can be pressed at any time to return to the Home screen.

FIGURE 2-3: ACCELEROMETER GRAPH



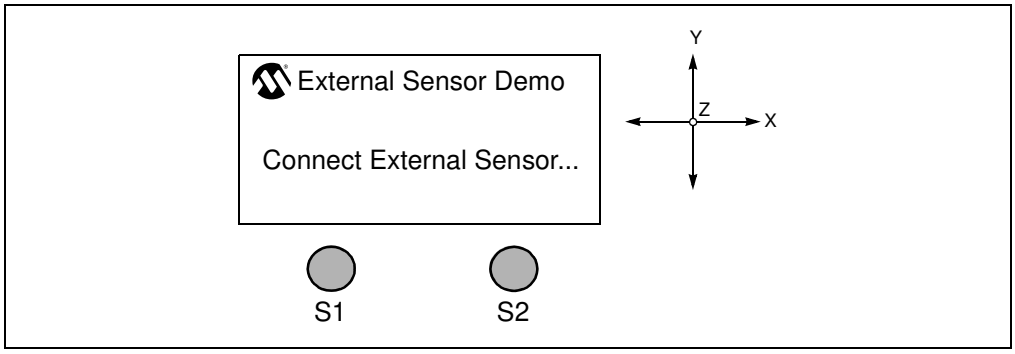
3. Selecting the **Ext. Sensor** cell (see Figure 2-4) starts the application, which captures the output of the external sensor that can be plugged-in at points TP12 and TP11.

FIGURE 2-4: SELECTING THE EXTERNAL SENSOR OPTION



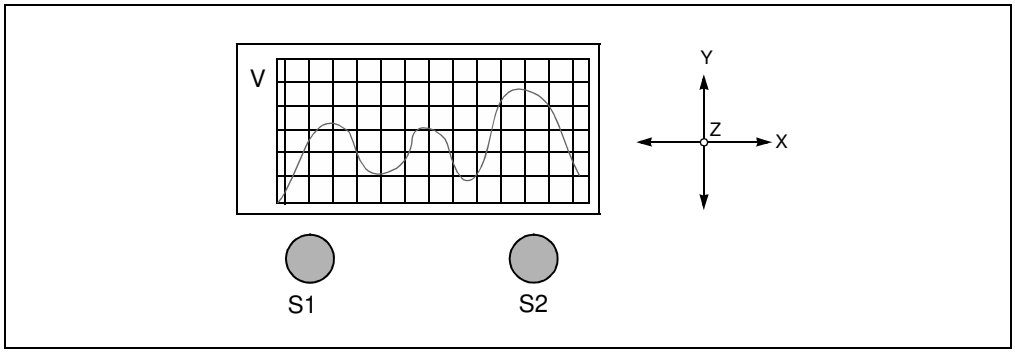
Before displaying the external sensor signal an information screen is displayed, which prompts the user to plug-in an external sensor, as shown in Figure 2-5.

FIGURE 2-5: EXTERNAL SENSOR PROMPT



Switch S1 can be pressed to slow down or speed up the display (see Figure 2-6) in case the displayed signal is too fast or too slow.

FIGURE 2-6: EXTERNAL SENSOR SIGNAL



Switch S2 can be pressed at any time to return to the Home screen.

4. Selecting the **Orientation** cell (see Figure 2-7) starts the application, which indicates the orientation of the starter kit.

   Patterns are displayed on the OLED display and messages are played out on the speaker indicating *Portrait*, *Landscape* and *Plane* orientations (see Figure 2-8). The acceleration in each of the axes is displayed on the left side of the screen as a fraction of gravitational acceleration constant on earth, 1g = 9.8 m/s^2.

FIGURE 2-7: SELECTING THE ORIENTATION OPTION

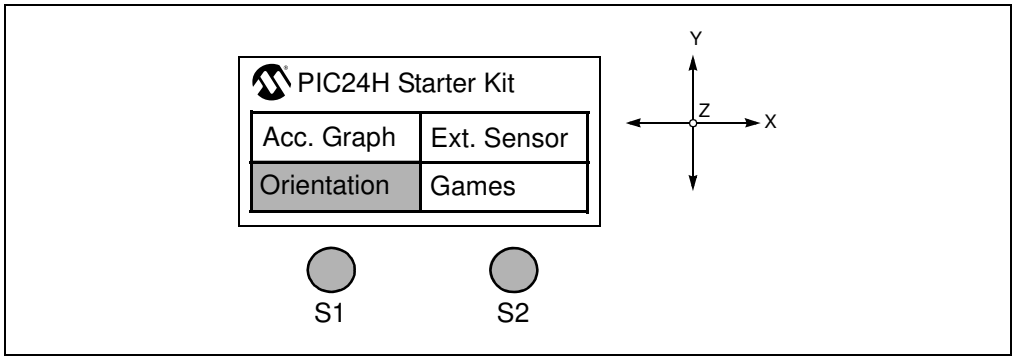
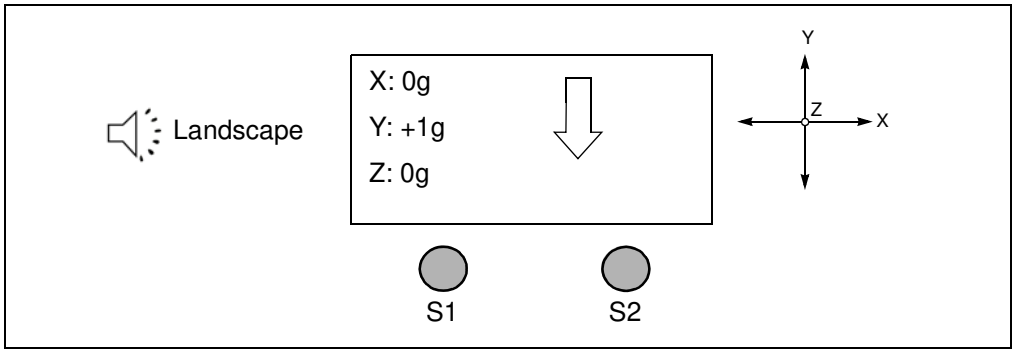
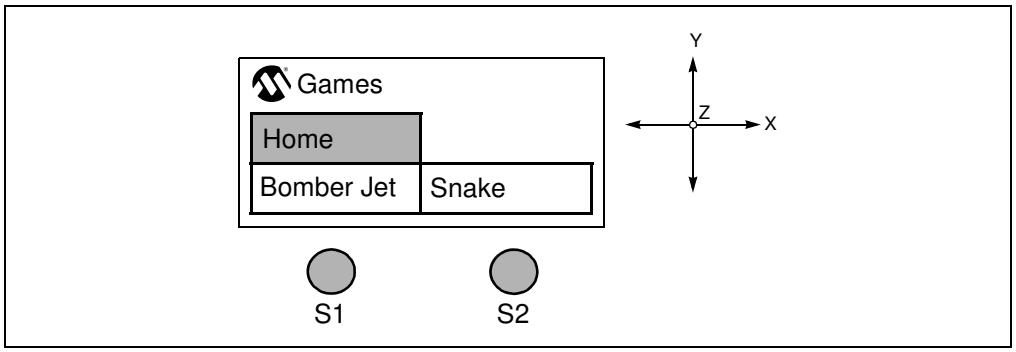


FIGURE 2-8: ORIENTATION PATTERNS



5. Selecting the **Games** cell displays a new screen with three cells: Home, Bomber Jet and Snake, as shown in Figure 2-9.

FIGURE 2-9: GAMES SCREEN

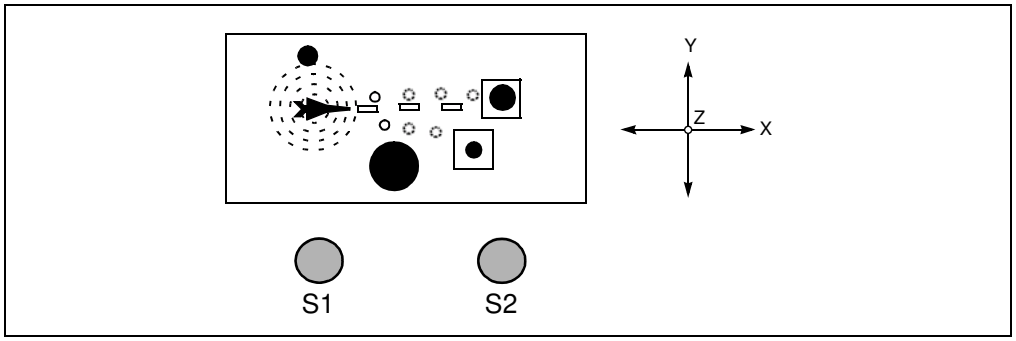


Selecting the Home cell will return you to the Home screen.

6. Selecting the **Bomber Jet** cell starts the Bomber Jet game.

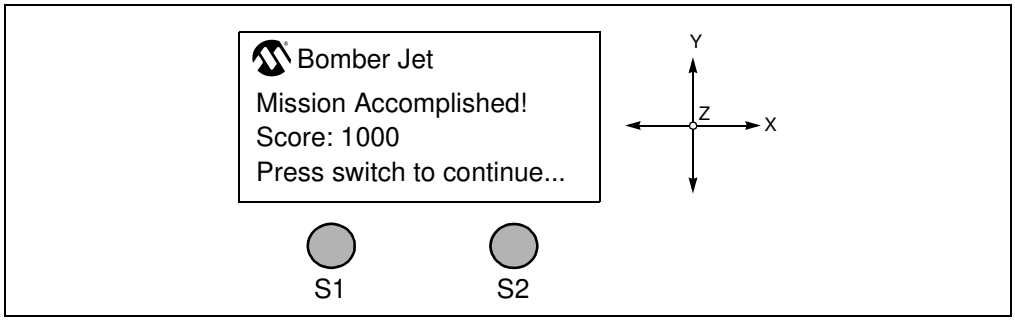
   The Jet can be maneuvered on the X and Y planes of the display by tilting the starter kit about X and Y axes. Asteroids and alien ships are encountered in the game. The jet should be maneuvered such that it doesn't collide with the asteroid, alien ship, and isn't hit by a missile from the alien ship. Pressing switch **S1** turns on a protective shield, momentarily. The shield will destroy any asteroid or alien ship missile in its path. Pressing **S2** releases missiles from the Bomber Jet. The alien ships and the asteroids are destroyed when hit by the missiles from the Bomber Jet. See Figure 2-10 for an example of the display during game play.

FIGURE 2-10: BOMBER JET GAME



The score increases whenever a missile fired by the Bomber Jet missile strikes an alien ship or an asteroid. Using the protective shield decreases the score. The game automatically exits to a Score screen after achieving a score of 1000 or when hit by an alien missile, or colliding with an alien ship or asteroid (see Figure 2-11).

FIGURE 2-11: BOMBER JET GAME SCORE

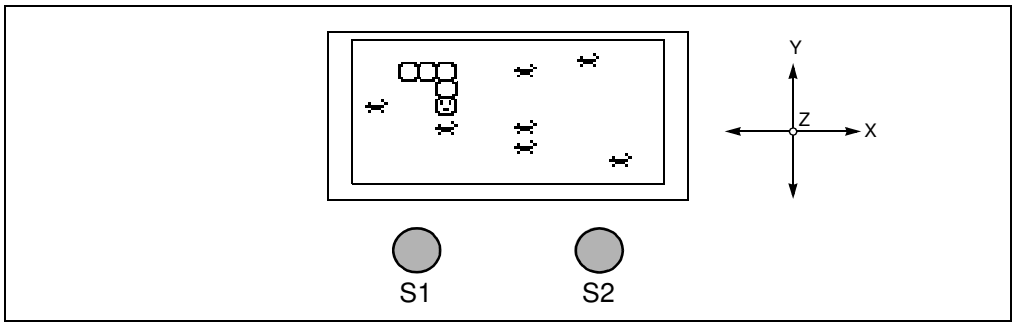


After the Score screen appears, Switch **S1** or **S2** can be pressed at any time to return to the Games screen.

7. Selecting the **Snake** cell starts the Snake game.

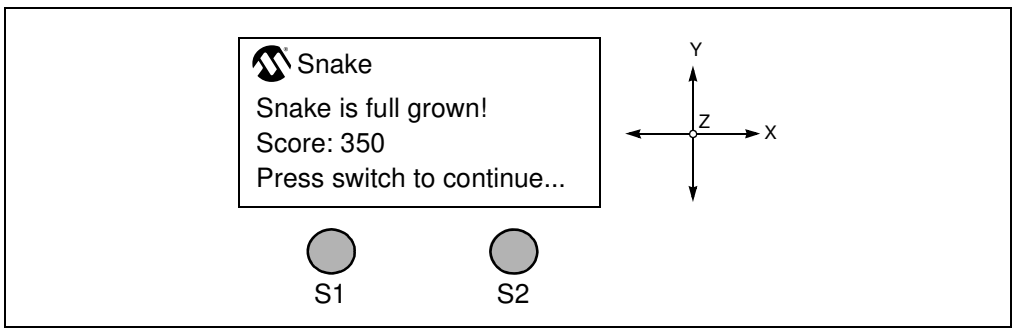
   The snake can be maneuvered on the X and Y plane of the display by tilting the starter kit about X and Y axes. Multiple snake food appears, which the snake must eat. The snake grows in size when it eats. The snake should be maneuvered to eat the food such that it doesn't collide with any of the four walls. See Figure 2-12 for an example of the display during game play.

FIGURE 2-12: SNAKE GAME



The score increases if the snake eats food. The game automatically exits to a Score screen after the snake grows a tail length of 25 rings or the snake hits any of the four walls (see Figure 2-13).

FIGURE 2-13: SNAKE GAME SCORE



After the Score screen appears, switch **S1** or **S2** can be pressed at any time to return to the Games screen.

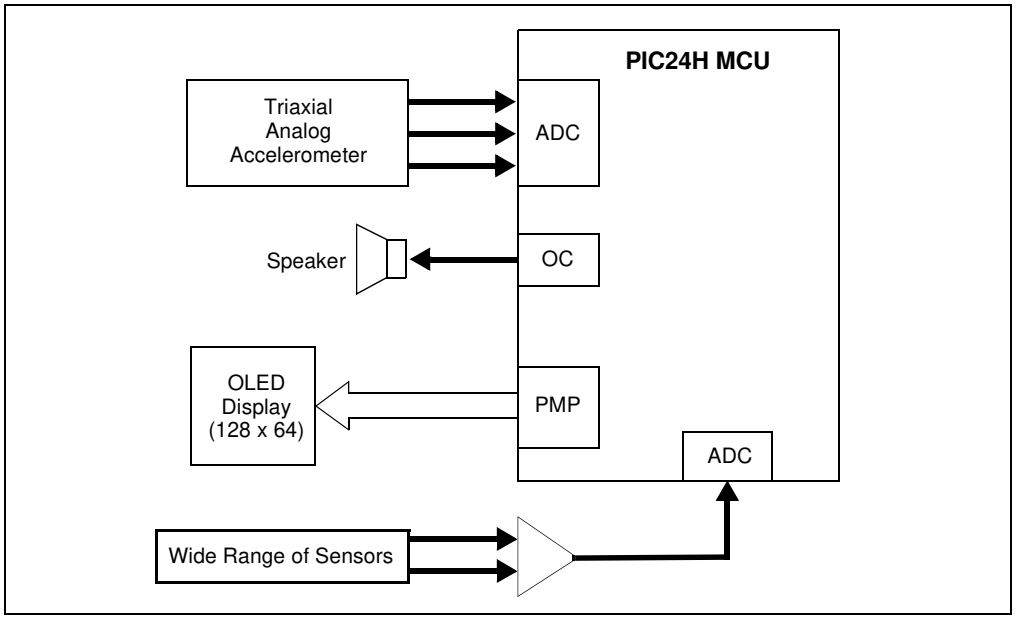
2.2 UNDERSTANDING THE DEMO

The PIC24H MCU device on the starter kit is pre-programmed with the main application demonstration. The CD that accompanies the starter kit also contains the main application code. As shown in Figure 2-14, this sample application uses the board to capture the triaxial acceleration signals from the accelerometer, and plays speech messages through the speaker with a visual display on the OLED display.

The board also features conditioning circuitry that provides the user the flexibility of plugging in a wide range of sensors and performing signal processing on the captured sensor signals.

Detailed descriptions of the basic modules are provided in **Chapter 4. "Hardware"**. The following sections give a brief functional description of these modules.

FIGURE 2-14: STARTER KIT BOARD



2.2.1 Accelerometer Interface

The starter kit has a triaxial analog accelerometer. The three (X, Y and Z) acceleration outputs are captured using the ADC module on the PIC24H MCU device. The ADC-captured accelerometer samples are processed differently in the PIC24H MCU based on the game chosen.

2.2.2 OLED Display Interface

The starter kit has a 128 x 64 OLED display. Different screens are displayed for the user to choose from. Depending on the stimulus obtained from the accelerometer and the switches, different screens are displayed, signals are graphed and games are controlled.

2.2.3 Speaker Interface

The speaker plays messages. The speech messages are compressed using the G.711 A-law and are stored in the program memory of the PIC24H. The PIC24H decodes the compressed data and generates PWM signals which are demodulated before being output through the speaker.

2.2.4 External Sensor Interface

The starter kit features an analog conditioning circuit which can be used by the user to plug in a wide range of external sensors. The differential output of the sensor flows through a differential amplifier and an anti-aliasing filter before being sampled by the on-chip ADC.

2.3 OTHER DEMO CODE EXAMPLES

The starter kit software CD includes another demo code example, External_Sensor_Demo.

The External_Sensor_Demo code example demonstrates the low-cost sensor signal capture and processing. When a sensor is plugged into the analog conditioning circuitry the code example captures the sensor signal through the ADC channel. The captured discrete time sensor signal can be processed inside the code example. Instructions are provided in the source file as to where user desired signal processing routines need to be added to process the sensor signal captured by the ADC.

The characteristics of the external analog sensor that can be used are provided in **Section 4.1.4 “Analog Conditioning Circuitry”**.

NOTES:

Chapter 3. Develop an Application

The MPLAB Starter Kit for PIC24 MCUs 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:

- Installing the Hardware and Software
- Setting Up an Example Application for Debug
- Running the Example Application
- Debugging the Example Application
- Programming the Debugged Application
- Creating Other PIC24H MCU Applications
- Determining Device Support and Reserved Resources
- Troubleshooting
- Settings Dialog, Info Tab

3.1 INSTALLING THE HARDWARE AND SOFTWARE

To install the hardware:

If you have not already set up the hardware to run the demo, follow these steps:

1. Power-up the starter kit by connecting the board to the USB port of a computer.

   You should briefly see a pop-up balloon in the system tray that states (1) new hardware has been found, (2) drivers are being installed, and (3) new hardware is ready for use. If you do not see these messages and the starter kit does not work, try reconnecting the USB cable. If reconnecting the USB cables does not work, see **Section 3.8 "Troubleshooting"**.

2. When powered up, the application starts and runs as described in Section 2.1.

To install the software:

Run the CD-ROM enclosed with the starter kit and install the software as directed.

3.2 SETTING UP AN EXAMPLE APPLICATION FOR DEBUG

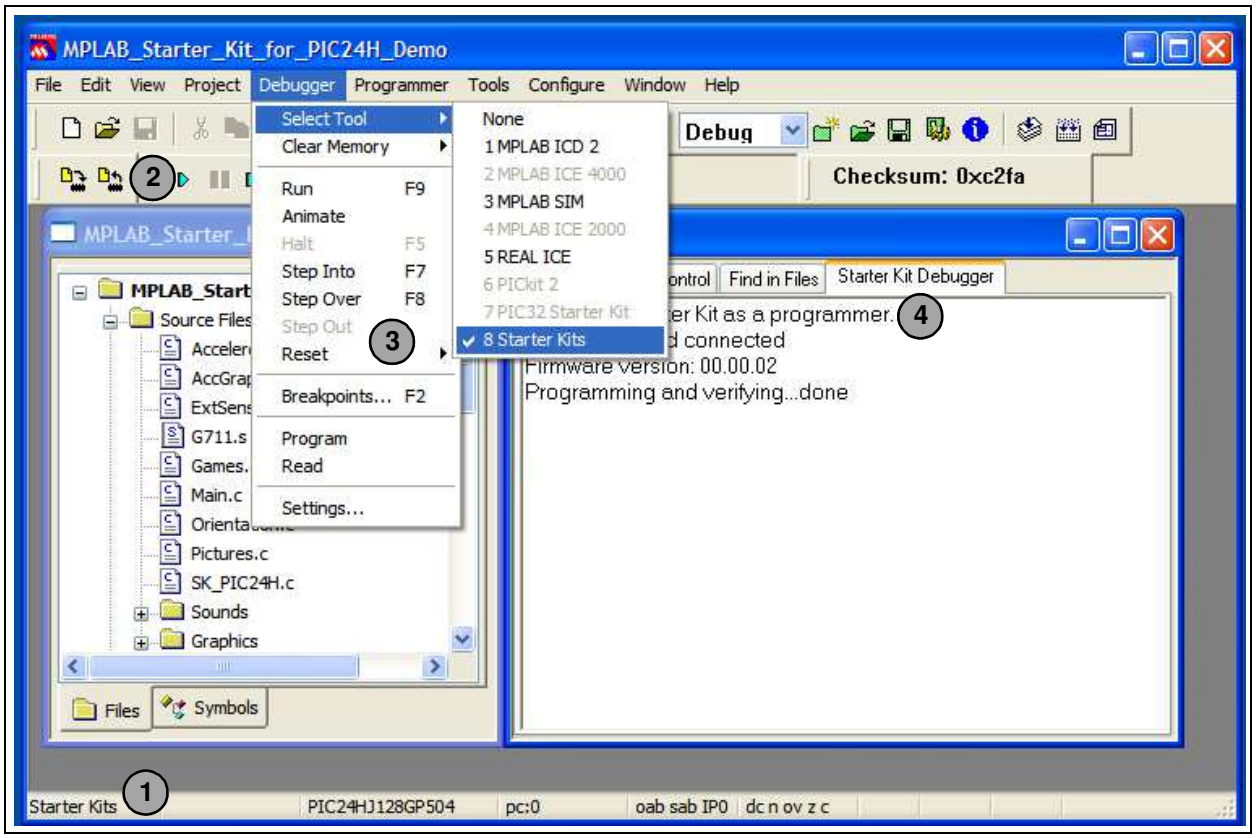
The MPLAB IDE software that is installed on your PC by the starter kit CD-ROM automatically opens an example application that you may use to examine debug features of the starter kit.

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 will change to add starter kit debug features (Figure 3-1): (1) the status bar will show Starter Kits as the debug tool, (2) a Starter Kit debug toolbar will be added, (3) the Debugger menu will change to add Starter Kit debug functions and (4) the Output window will display communication status between MPLAB IDE and the stater kit on the **Starter Kit Debugger** tab.

   Also, several device resources are used for debug. For details, see **Section 3.7 "Determining Device Support and Reserved Resources"**.

FIGURE 3-1: STARTER KIT AS DEBUG TOOL



4. Select *Debugger>Program* to program the application code into the PIC24H MCU 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.

3.3 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 **Halt** or a breakpoint, you can step.

The toolbar buttons shown in Table 3-1 can be used for quick access to commonly used debug operations:

TABLE 3-1: TOOLBAR BUTTONS

Debugger Menu	Debug Toolbar	Debugger Menu	DSK Toolbar
Run		Program	
Halt		Read	
Animate			
Step Into			
Step Over			
Reset			
Breakpoints			

To see how these options function, do the following:

1. Select *Debugger>Reset>Processor Reset* or click **Reset** to reset the program.
2. Select *Debugger>Run* or click **Run**. Observe how the application operates.
3. Select *Debugger>Halt* or click **Halt** to stop the program execution. A green solid arrow will mark the line of code in the File window where the program halted.
4. Select *Debugger>Step Into* or click **Step Into** 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* click **Reset** to reset the program again. The arrow will disappear, meaning the device is reset.

3.4 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 may 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:

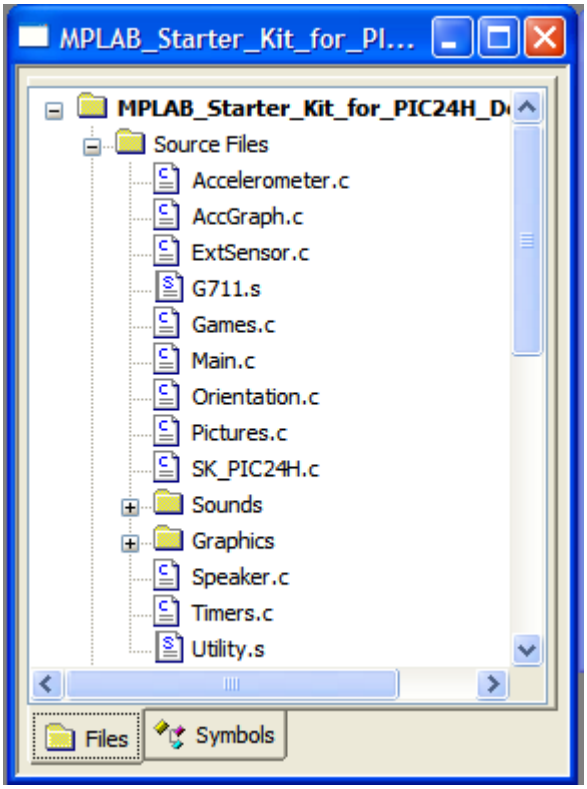
- Editing Application Code
- Using Breakpoints and Mouseovers
- Using Watch Windows

3.4.1 Editing Application Code

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

1. Select *Edit>New* to create new code **or** *Edit>Open* to search for and open an existing code file.
2. Double click a file in the Project window to open an existing code file. See an example Project window in Figure 3-2.

FIGURE 3-2: EXAMPLE PROJECT



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

3.4.2 Using Breakpoints and Mouseovers

To set a breakpoint in code:

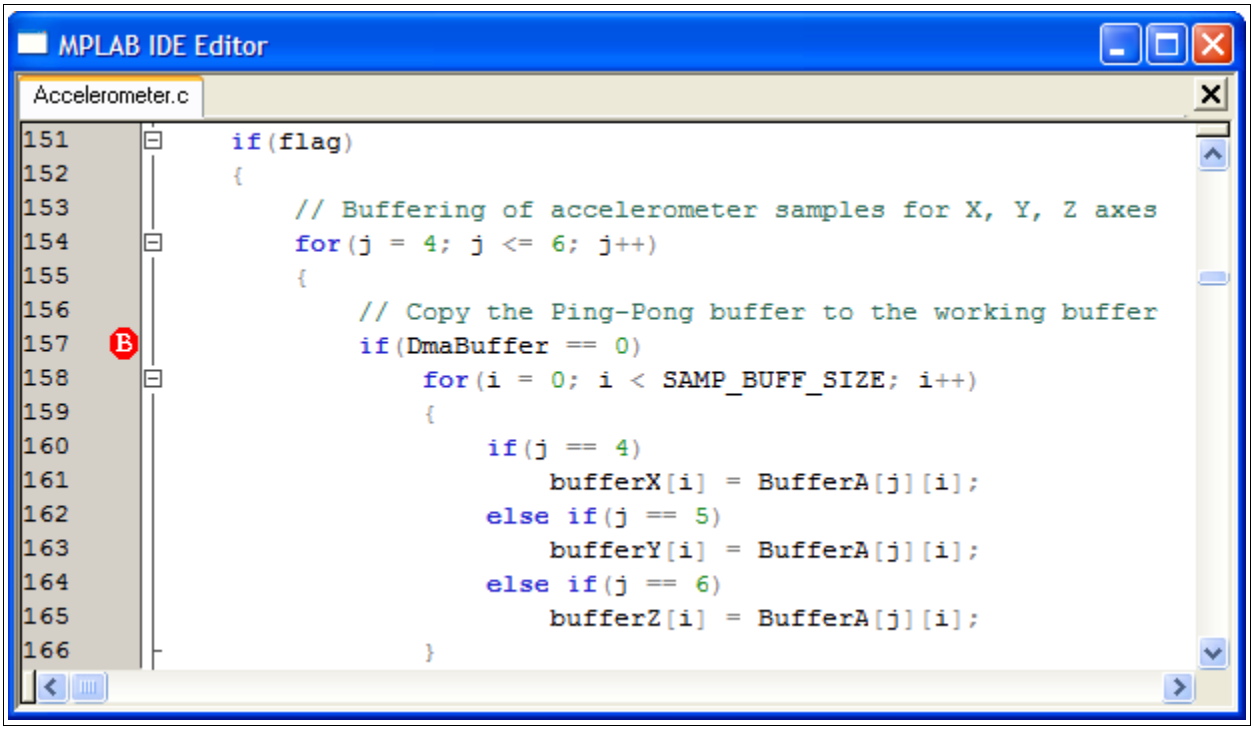
1. **Double Click the Gutter** – Double click in the window gutter next to the line of code where you want the breakpoint. Double click again to remove the breakpoint.

   Note: Double click must be set up for breakpoints. See the *Edit>Properties*, ***ASM/C/BAS* File Type** tab, and the check box for "Double-click Toggles Breakpoint".

2. **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.
3. **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 will appear as a red hexagon with a “B” as shown in Figure 3-3.

FIGURE 3-3: EXAMPLE BREAKPOINT



Once code is halted, hovering over variables pops up the current value of those variables (see Figure 3-3.)

Note:	This feature must be set up. See the *Edit>Properties*, **Tooltips** tab, and check the “Enable Variable Mouseover Values” check box.

3.4.3 Using Watch Windows

To use a Watch window:

1. 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).
2. Select an SFR or Symbol from the list and click the related **Add** button to add it to the Watch window. Or click the “Address” column and enter an absolute address.

A Watch window populated with SFRs and Symbols will look like Figure 3-4. For more information on using Watch windows, see MPLAB IDE Help.

FIGURE 3-4: EXAMPLE WATCH