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

PICkit™ 1 Flash Starter Kit 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 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona and Mountain View, California in October 2003. The Company's quality system processes and procedures are for its PICmicro® 8-bit MCUs, 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.

PICKIT™ 1 FLASH STARTER KIT USER'S GUIDE

Table of Contents

Preface

INTRODUCTION

This chapter contains general information about this user's guide and customer support that will be useful prior to using the PICkit™ 1 Flash Starter Kit.

HIGHLIGHTS

Items discussed in this chapter are:

- About this Guide
- Warranty Registration
- Recommended Reading
- Troubleshooting
- Microchip On-Line Support
- Customer Change Notification Service
- Customer Support

ABOUT THIS GUIDE

This document describes how to use the PICkit™ 1 Flash Starter Kit. The manual layout is as follows:

- **Chapter 1: PICkit™ 1 Flash Starter Kit** – An overview of the PICkit™ 1 Flash Starter Kit and instructions on how to use it.
- **Chapter 2: PICkit™ 1 Classic Software** – This chapter provides instructions on how to use the PICkit 1 Classic programming software to program 8 and 14-pin Mid-range Flash Microcontroller (MCU) Family devices.
- **Chapter 3: PICkit™ 1 Baseline Flash Software** – This chapter provides instructions on how to use the PICkit 1 Baseline Flash programming software to program 8 and 14-pin Baseline (12-bit Core) Flash Microcontroller (MCU) Family devices.
- **Chapter 4: PICkit™ 1 and MPLAB® IDE** – A quick overview on how to load a project in MPLAB IDE and program a device with MPLAB IDE.
- **Chapter 5: Troubleshooting** – This chapter describes common problems and possible solutions for solving problems with the PICkit™ 1 Flash Starter Kit.
- **Appendix A: PICkit™ 1 Hardware Description** – This appendix contains the PICkit 1 hardware description and schematic diagrams.
- **Appendix B: Baseline Flash Microcontroller Programmer Hardware Description** – This appendix contains the Baseline Flash Microcontroller Programmer hardware description and schematic diagram.
- **Appendix C: PIC12F675 Programming Projects** – This appendix contains introductory lessons for persons new to PICmicro® microcontroller units (MCU).
- **Worldwide Sales and Service** – A listing of Microchip sales and service locations and telephone numbers worldwide.

Conventions Used in This Guide

This manual uses the following documentation conventions:

TABLE 1-1: DOCUMENTATION CONVENTIONS

Description	Represents	Examples
Code (Courier font):		
Plain characters	Sample code Filenames and paths	`#define START` `c:\autoexec.bat`
Angle brackets: < >	Variables	`<label>, <exp>`
Square brackets [ ]	Optional arguments	`MPASMWIN [main.asm]`
Curly brackets and pipe character: { | }	Choice of mutually exclusive arguments; An OR selection	`errorlevel {0|1}`
Lower case characters in quotes	Type of data	`"filename"`
Ellipses...	Used to imply (but not show) additional text that is not relevant to the example	`list ["list_option..., "list_option"]`
0xnnn	A hexadecimal number where n is a hexadecimal digit	`0xFFFF, 0x007A`
Italic characters	A variable argument; it can be either a type of data (in lower case characters) or a specific example (in upper case characters).	`char isascii (char, ch);`
Interface (Arial font):		
Underlined, italic text with right arrow	A menu selection from the menu bar	*File > Save*
Bold characters	A window or dialog button to click	**OK**, **Cancel**
Characters in angle brackets < >	A key on the keyboard	<Tab>, <Ctrl-C>
Documents (Arial font):		
Italic characters	Referenced books	*MPLAB IDE User's Guide*

Documentation Updates

All documentation becomes dated, and this user's guide is no exception. Since the PICkit™ 1 Flash Starter Kit *User's Guide* and other Microchip tools are constantly evolving to meet customer needs, some PICkit™ 1 Flash Starter Kit actual dialogs and/or tool descriptions may differ from those in this document. Please refer to our web site to obtain the latest documentation available.

Documentation Numbering Conventions

Documents are numbered with a "DS" number. The 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 = The document number.

A = The revision level of the document.

WARRANTY REGISTRATION

Please complete the enclosed Warranty Registration Card and mail it promptly. Sending in your 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 PICkit™ 1 Flash Starter Kit. Other useful documents are listed below:

PIC12F629/675 Data Sheet (DS41190)

Consult this document for information regarding the PIC12F629/675 8-pin Flash-based 8-bit CMOS microcontroller device specifications.

PIC16F630/676 Data Sheet (DS40039)

Consult this document for information regarding the PIC16F630/676 14-pin Flash-based 8-bit CMOS microcontroller device specifications.

MPLAB® IDE, Simulator, Editor User's Guide (DS51025)

Consult this document for more information pertaining to the installation and features of the MPLAB Integrated Development Environment (IDE) Software.

To obtain these documents, contact the nearest Microchip sales location (see back page). These documents are also available on the Microchip web site at: www.microchip.com.

Microsoft® Windows® Manuals

This manual assumes that users are familiar with the Microsoft Windows operating system. Many excellent references exist for this software program, and should be consulted for general operation of Windows.

TROUBLESHOOTING

See the Troubleshooting section for information on common problems.

MICROCHIP ON-LINE SUPPORT

Microchip provides on-line support on the Microchip web site at:

http://www.microchip.com

A file transfer site is also available by using an FTP service connecting to:

ftp://ftp.microchip.com

The web site and file transfer site provide a variety of services. Users may download files for the latest development tools, data sheets, application notes, user guides, articles and sample programs. A variety of Microchip specific business information is also available, including listings of Microchip sales offices and distributors. Other information available on the web site includes:

- Latest Microchip press releases
- Technical support section with FAQs
- Design tips
- Device errata
- Job postings
- Microchip consultant program member listing
- Links to other useful web sites related to Microchip products
- Conferences for products, development systems, technical information and more
- Listing of seminars and events

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip started the customer notification service to help customers stay current on Microchip products with the least amount of effort. Once you subscribe, you will receive email notifications whenever we change, update, revise or have errata related to your specified product family or development tool.

Go to the Microchip web site (www.microchip.com) and click on Customer Change Notification. Follow the instructions to register.

The Development Systems product group categories are:

- Compilers
- Emulators
- In-Circuit Debuggers
- MPLAB IDE
- Programmers

Here is a description of these categories:

Compilers – The latest information on Microchip C compilers and other language tools. These include the MPLAB C17, MPLAB C18 and MPLAB C30 C Compilers; MPASM™ and MPLAB ASM30 assemblers; MPLINK™ and MPLAB LINK30 linkers; and MPLIB™ and MPLAB LIB30 librarians.

Emulators – The latest information on Microchip in-circuit emulators. This includes the MPLAB ICE 2000 and MPLAB ICE 4000.

In-Circuit Debuggers – The latest information on Microchip in-circuit debuggers. These include the MPLAB ICD and MPLAB ICD 2.

MPLAB – 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 SIM simulator, MPLAB IDE Project Manager and general editing and debugging features.

Programmers – The latest information on Microchip device programmers. These include the PRO MATE® II device programmer and PICSTART® Plus development programmer.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributors
- Local Sales Office
- Field Application Engineers (FAEs)
- Corporate Applications Engineers (CAEs)
- Systems Information and Upgrade Hot Line

Customers should call their distributor or field application engineer (FAE) for support. Local sales offices are also available to help customers. See the last page of this document for a listing of sales offices and locations.

Corporate applications engineers (CAEs) may be contacted at (480) 792-7627.

Systems Information and Upgrade Line

The Systems Information and Upgrade Information Line provides system users with a listing of the latest versions of all of Microchip's development systems software products. Plus, this line provides information on how customers can receive the most current upgrade kits. The Information Line Numbers are:

1-800-755-2345 for U.S. and most of Canada.
1-480-792-7302 for the rest of the world.

Chapter 1. PICkit™ 1 Flash Starter Kit

1.1 INTRODUCTION

The PICkit 1 Flash Starter Kit is a low-cost introductory programmer and starter kit with an easy to use PC host program user interface for programming Microchip's 8/14-pin Flash family of microcontrollers. The starter kit is designed to help the user get up to speed quickly using PIC® microcontrollers.

Starting with Version 2 PC software and PIC16C745 firmware, new Baseline (12-bit Core) Flash PICmicros® can be programmed by the PICkit 1 Flash Starter Kit.

1.2 HIGHLIGHTS

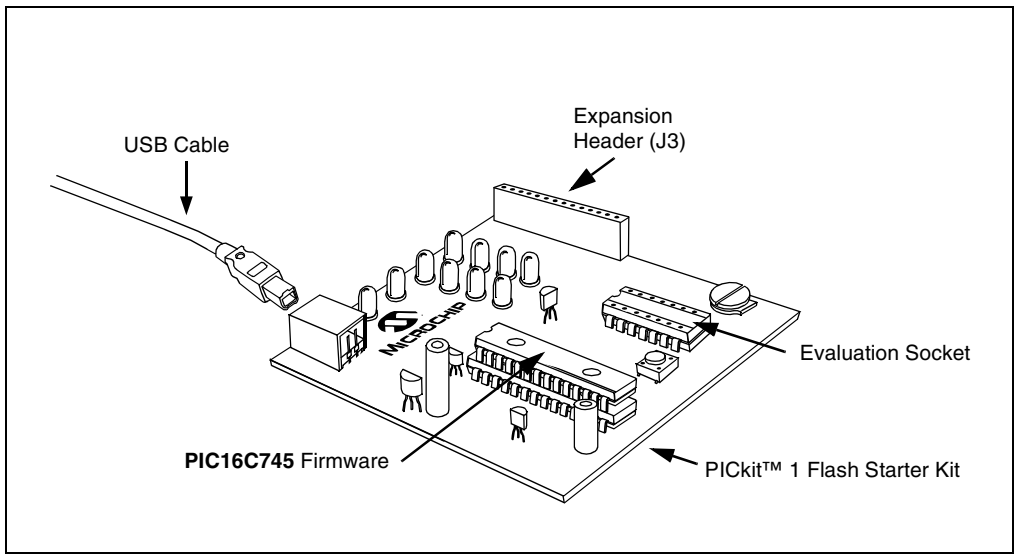
This chapter discusses:

- Running the PICkit 1 Flash Starter Kit Default Demonstration
- PICkit 1 Classic and Baseline Flash Programming Software
- New to Programming PICmicro MCUs

1.3 RUNNING THE PICkit 1 FLASH STARTER KIT DEFAULT DEMONSTRATION

The PICkit 1 Flash Starter Kit comes preprogrammed with a demonstration program. Connect the PICkit 1 Flash Starter Kit to the PC's USB port using the USB cable as shown in Figure 1-1. The demo program will blink the eight red lights in succession. Press the **Push Button Switch**, labeled **SW1**, on the board and the sequence of the lights will change. Rotate the **Variable Resistor**, labeled **RP1**, and the light sequence will blink at a different rate.

FIGURE 1-1: PICkit 1 FLASH STARTER KIT



1.4 PICkit 1 CLASSIC AND BASELINE FLASH PROGRAMMING SOFTWARE

There are two PC host programs for the PICkit 1 Flash Starter Kit:

1.4.1 PICkit 1 Classic

The PICkit 1 Classic programming software programs 8 and 14-pin Mid-range (14-bit Core) Flash Microcontroller (MCU) Family devices. Instructions on how to use the PICkit 1 Classic program is given in Chapter 2.

1.4.2 PICkit 1 Baseline Flash

The PICkit 1 Baseline Flash programming software programs 6, 8 and 14-pin Baseline (12-bit Core) Flash Microcontroller (MCU) Family devices. Instructions on how to use the PICkit 1 Baseline Flash program is given in Chapter 3.

1.5 NEW TO PROGRAMMING PICmicro® MCUs

Appendix C. "PIC12F675 Programming Projects" contains introductory lessons for persons new to PICmicro® microcontroller units (MCU). The programming projects in the appendix describe different concepts in controlling the PIC12F675 PICmicro® microcontroller unit (MCU). Each lesson includes instructions for running a program demo that illustrates basic concepts. It is best to follow the lessons in sequential order, for each lesson builds upon the previous one. (The hex files and source code for the lessons can be found on the PICkit 1 Flash Starter Kit CDROM.)

Chapter 2. PICkit™ 1 Classic Software

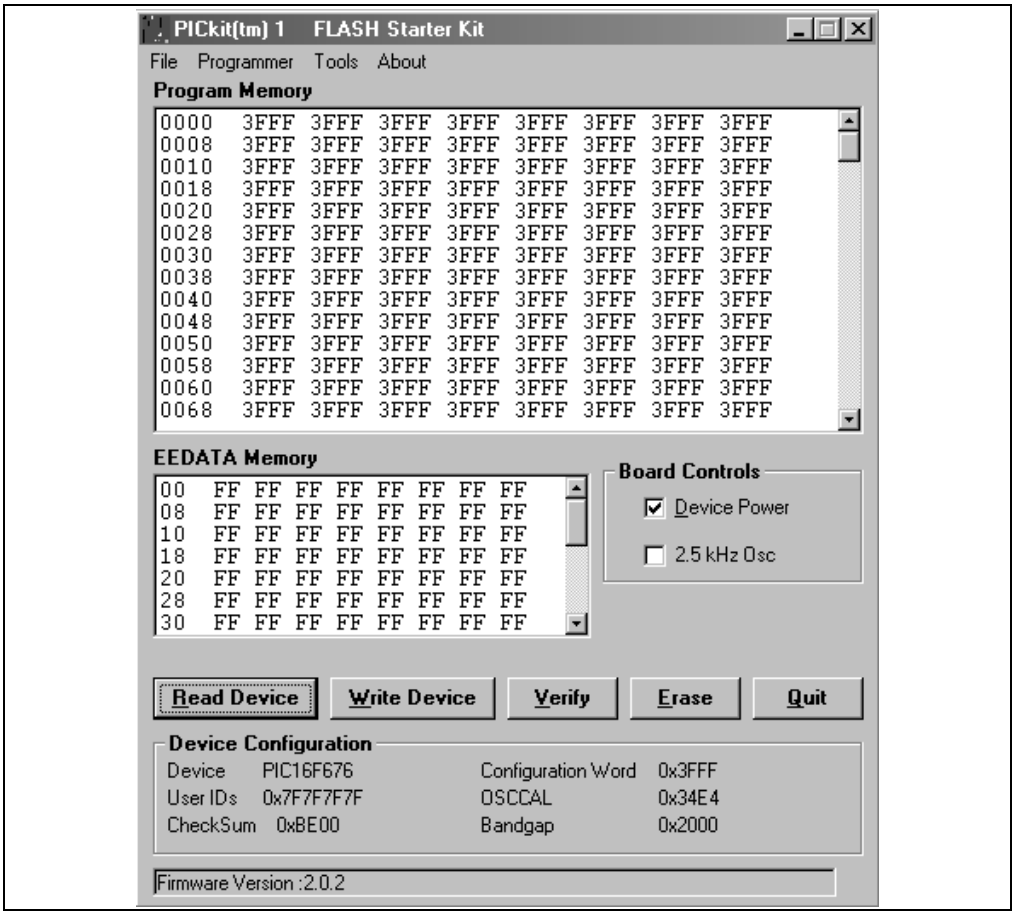
2.1 INTRODUCTION

This chapter provides instructions on how to use the PICkit 1 Classic programming software to program 8 and 14-pin Mid-range (14-bit Core) Flash Microcontroller (MCU) Family devices.

2.2 USING PICkit 1 CLASSIC PROGRAMMING SOFTWARE

Start the PICkit 1 Classic program by selecting *Start > Programs > PICkit 1 Flash Starter Kit* > *PICkit 1 Classic* . The programming interface appears as shown in Figure 2-1.

FIGURE 2-1: PICkit 1 CLASSIC PROGRAMMING INTERFACE



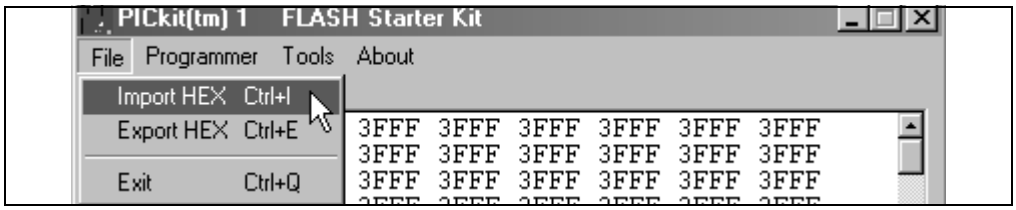
Notice that the Device Power check box is selected. This is a default function indicating the device power is turned on.

Note:	To turn the device power off, deselect the check box.

2.2.1 Import HEX File

To import a compiled program (hex file), select *File > Import HEX*, as shown in Figure 2-2. Browse for the hex file and click **Open**.

FIGURE 2-2: IMPORT HEX FILE

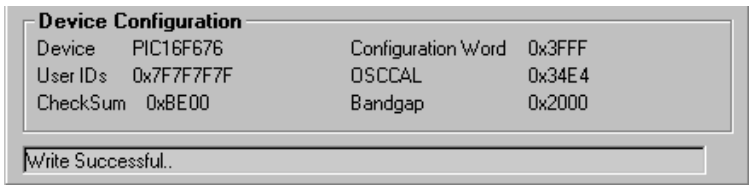


The code is displayed in the Program Memory and EEDATA Memory windows. (For more information on Program and EEDATA memory, see Lesson 5 and Lesson 6 in **Appendix C. "PIC12F675 Programming Projects"**.)

After the hex file is imported, write the program to the device by clicking the **Write Device** button. The existing program will be erased and replaced with the new one. The status of the program write is displayed in the status bar located at the bottom of the interface window.

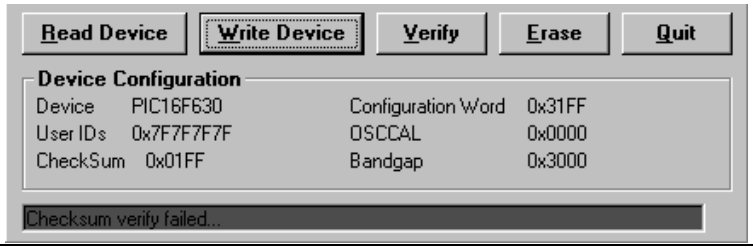
If the write is successful, the status bar turns green and displays "Write Successful", as shown in Figure 2-3.

FIGURE 2-3: WRITE SUCCESSFUL STATUS BAR



If the write fails, the status bar turns red and displays "Checksum verify failed", as shown in Figure 2-4. This error indicates the data was corrupted during the programming sequence. If this error is displayed, try writing the program to the device again. If this error continues, see **Chapter 5. "Troubleshooting"** for assistance.

FIGURE 2-4: WRITE ERROR STATUS BAR



2.2.2 Automatic File Reload

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

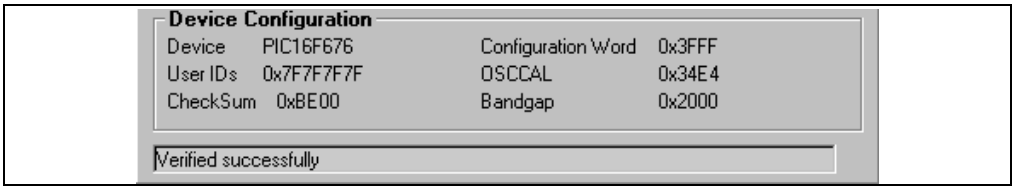
In the normal mode of operation, the hex file is updated with every build in MPLAB IDE. This ensures that the latest version built by MPLAB IDE will be written to the device.

2.2.3 Verify Program Code

This function verifies the program written to the device against a hex file. It compares all areas of memory including Program, EEDATA and Configuration.

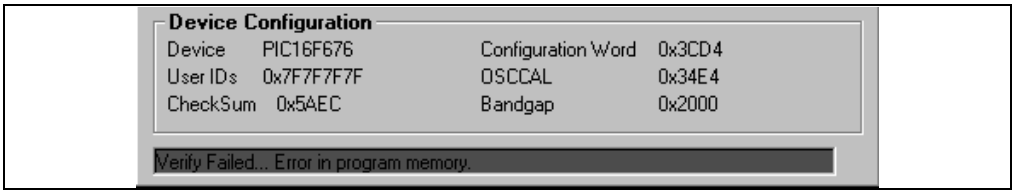
To verify the code, import the hex file and click **Verify**. If the code is the same, the status bar turns green and displays “Verified successfully”, as shown in Figure 2-5.

FIGURE 2-5: VERIFY PROGRAM CODE STATUS BAR



If a discrepancy is found, the status bar turns red and displays the error “Verify Failed... Error in program memory.”, as shown in Figure 2-6. If the Verify fails, it is possible that the device is code protected.

FIGURE 2-6: CODE ERROR STATUS BAR



2.2.4 Read Device

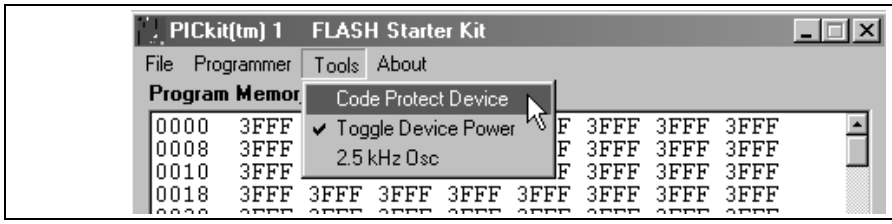
To view the code written to the device, click **Read Device**. The code is displayed in the Program and EEDATA Memory windows for your review. If all zeros are displayed in the Program Memory window, it is possible that the device is code protected.

2.2.5 Code Protect Device

This function enables the code protection features of the device. To protect the code, complete the following steps:

1. Import the hex file.
2. Select *Tools > Code Protect Device*, as shown in Figure 2-7.
3. Click **Write Device**.

FIGURE 2-7: CODE PROTECT



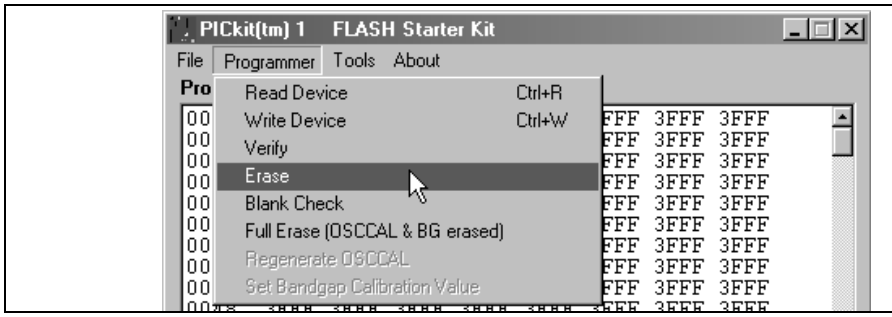
Note: If the device is read after it has been code protected, the Program and EEDATA Memory windows will display all zeros.

2.2.6 Erase

The Erase function erases code from the device. However, this function is not normally needed since the Write Device function performs an erase prior to writing code to the device.

To erase the device using the Erase function, click the **Erase** button or select *Programmer > Erase* from the toolbar menu, as shown in Figure 2-8.

FIGURE 2-8: DEVICE ERASE



2.2.7 Full Erase

Unlike the Erase function, the Full Erase allows the user to erase not only the device, but the OSCCAL and bandgap calibration as well. Performing a full erase is not recommended or needed in the normal course of events. Only use this function if the OSCCAL or bandgap data has been corrupted.

To perform a full erase, select *Programmer > Full Erase (OSCCAL & BG erased)* from the toolbar menu, as shown in Figure 2-8.

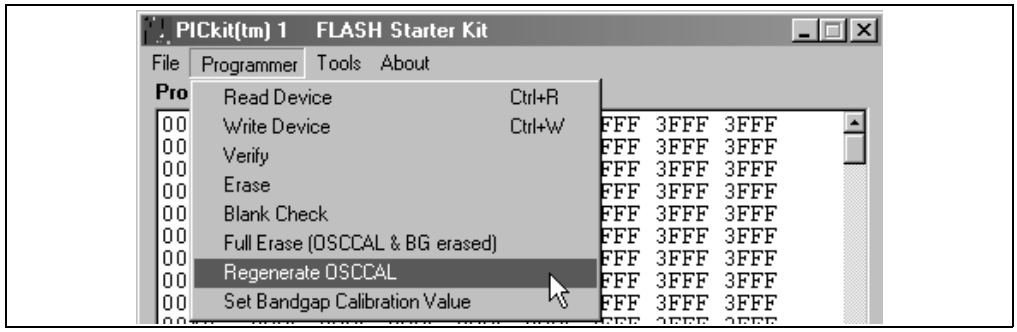
Once the full erase is complete, regenerate the OSCCAL and reset the bandgap calibration value bits.

Note: The regenerated OSCCAL function runs a program on the device to recalculate the oscillator calibration value. See Lesson 7 in **Appendix C. "PIC12F675 Programming Projects"** for more details.

2.2.8 Regenerate OSCCAL

To regenerate the OSCCAL, select *Programmer > Regenerate OSCCAL* from the toolbar menu, as shown in Figure 2-9.

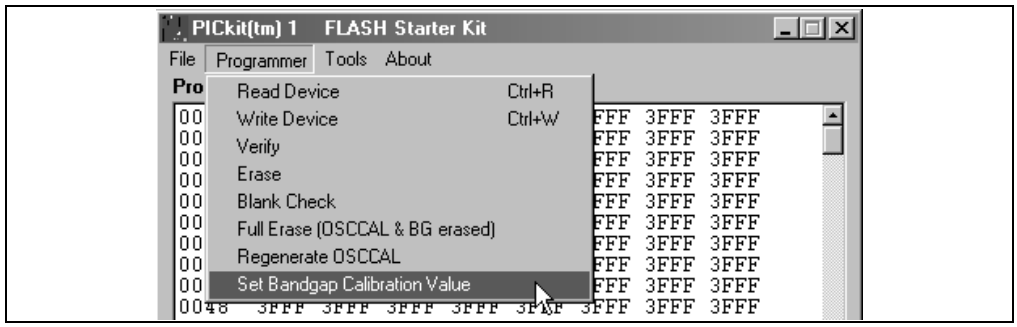
FIGURE 2-9: REGENERATE OSCCAL



2.2.9 Set Bandgap Calibration Value

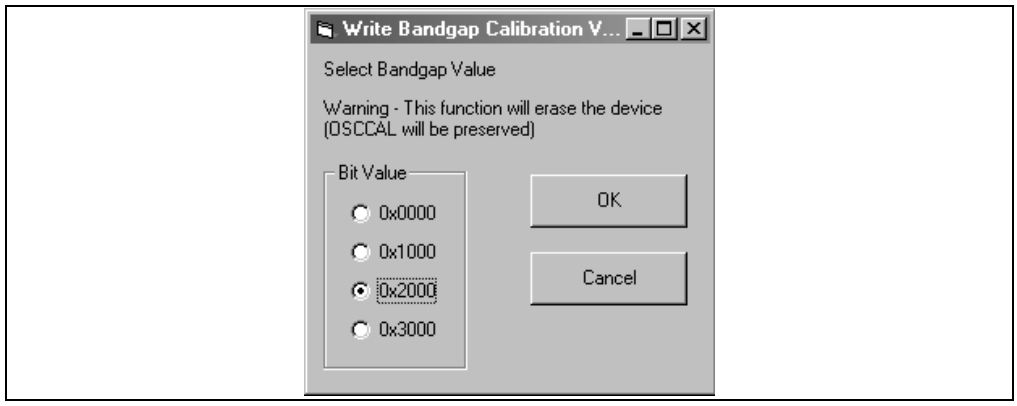
To set the bandgap calibration value, select *Programmer > Set Bandgap Calibration Value* from the toolbar menu, as shown in Figure 2-10.

FIGURE 2-10: REGENERATE BANDGAP CALIBRATION VALUE



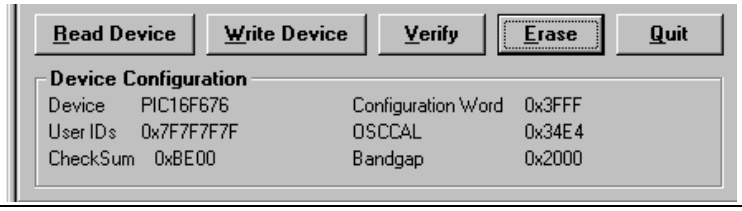
The Write Bandgap Calibration Value window will appear as shown in Figure 2-11. Select a bit value from the list and click OK.

FIGURE 2-11: SELECT BANDGAP BIT VALUE



The bandgap bit value will appear in the Device Configuration box, as shown in Figure 2-12.

FIGURE 2-12: DEVICE CONFIGURATION



For more information on the OSCCAL and bandgap calibration, see the device data sheets located on the CDROM or the Microchip web site (www.microchip.com).

Note: The "Regenerated OSCCAL" and "Set Bandgap Calibration Value" functions are only available on the PIC12F629/675 and PIC16F630/676 devices.

2.2.10 2.5 kHz OSC

The 2.5 kHz OSC is a square wave signal that, when selected, is input to pin 3 of the evaluation socket. See the schematic Figure A-1 in Appendix A.

This function is used by Lesson 7 in **Appendix C. "PIC12F675 Programming Projects"**, and for OSCCAL regeneration.

FIGURE 2-13: BOARD CONTROLS

Chapter 3. PICkit™ 1 Baseline Flash Software

3.1 INTRODUCTION

This chapter provides instructions on how to use the PICkit 1 Baseline Flash programming software to program 6, 8 and 14-pin Baseline (12-bit Core) Flash Microcontroller (MCU) Family devices.

3.2 USING PICkit 1 BASELINE FLASH PROGRAMMING SOFTWARE

Start the PICkit 1 Baseline Flash program by selecting *Start > Programs > PICkit(tm) 1 Flash Starter Kit* > *PICkit 1 Baseline Flash* . The programming interface appears, as shown in Figure 3-1.

FIGURE 3-1: PICkit 1 BASELINE FLASH PROGRAMMING INTERFACE

Notice that the Device Power check box is selected. This is a default function indicating the device power is turned on.

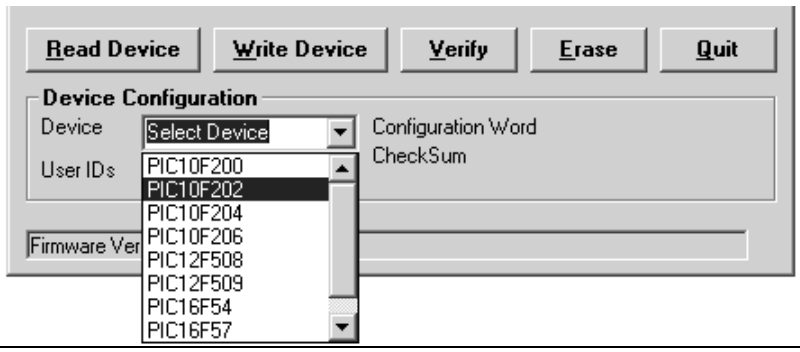
Note:	To turn the device power off, deselect the check box.

3.2.1 Select Device

The user must first select the device intended for programming. Select the device by clicking on the **Device** drop-down menu, as shown in Figure 3-2. The drop-down menu lists the available devices that can be programmed by the PICkit 1 Baseline Flash programming software.

Note: Baseline (12-bit core) devices do not contain a Device ID. As a result, the PICkit 1 programming software cannot determine which device is in the evalution socket of the PICkit 1 Flash Starter Kit. It is important that the user verify that the target device and the selected device match. Otherwise unintended results will happen.

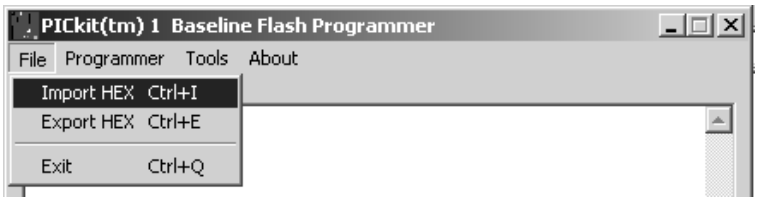
FIGURE 3-2: SELECT DEVICE



3.2.2 Import HEX File

To import a compiled program (hex file), select *File > Import HEX*, as shown in Figure 3-3. Browse for the hex file and click **Open**.

FIGURE 3-3: IMPORT HEX FILE

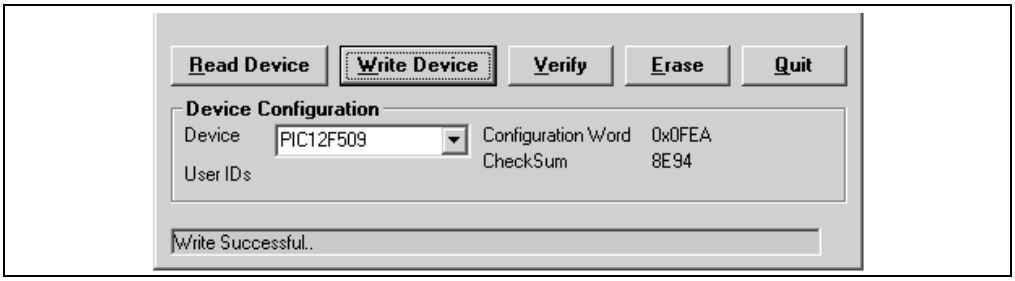


The code is displayed in the Program Memory window.

After the hex file is imported, write the program to the device by clicking the **Write Device** button. The existing program will be erased and replaced with the new one. The status of the program write is displayed in the status bar located at the bottom of the Interface window.

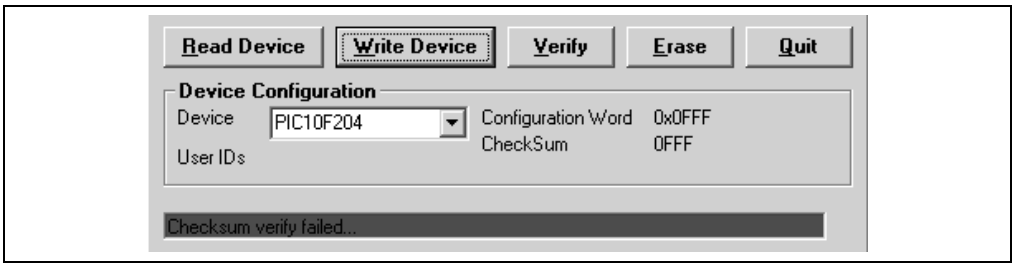
If the write is successful, the status bar turns green and displays "Write Successful", as shown in Figure 3-4.

FIGURE 3-4: WRITE SUCCESSFUL STATUS BAR



If the write fails, the status bar turns red and displays “Checksum verify failed”, as shown in Figure 3-5. This error indicates the data was corrupted during the programming sequence. If this error is displayed, write the program to the device again. If this error continues, see **Chapter 5. “Troubleshooting”** for assistance.

FIGURE 3-5: WRITE ERROR STATUS BAR



3.2.3 Automatic File Reload

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

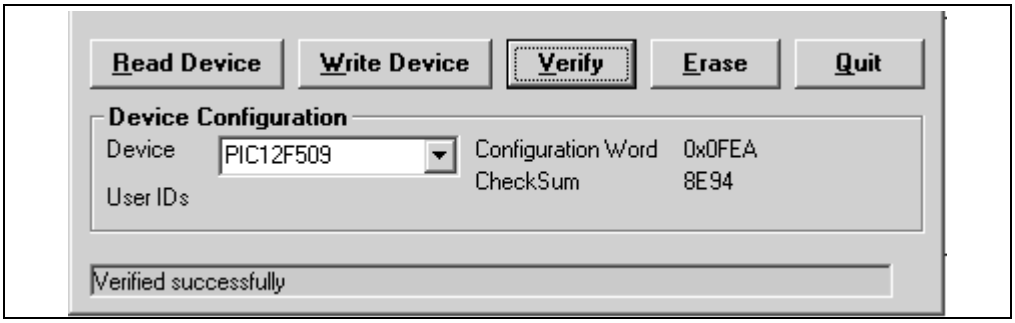
In the normal mode of operation, the hex file is updated with every build in MPLAB IDE. This ensures that the latest version built by MPLAB IDE will be written to the device.

3.2.4 Verify Program Code

This function verifies the program written to the device against a hex file. It compares all areas of memory including Program and Configuration.

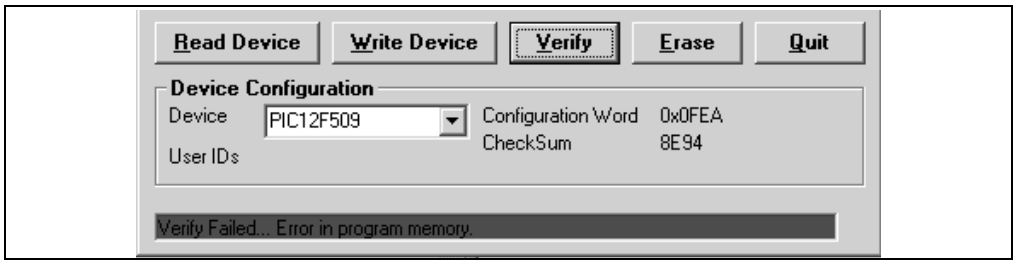
To verify the code, import the hex file and click **Verify**. If the code is the same, the status bar turns green and displays “Verified successfully”, as shown in Figure 3-6.

FIGURE 3-6: VERIFY PROGRAM CODE STATUS BAR



If a discrepancy is found, the status bar turns red and displays the error “Verify Failed Error in program memory”, as shown in Figure 3-7.

FIGURE 3-7: VERIFY PROGRAM CODE ERROR STATUS BAR



3.2.5 Read Device

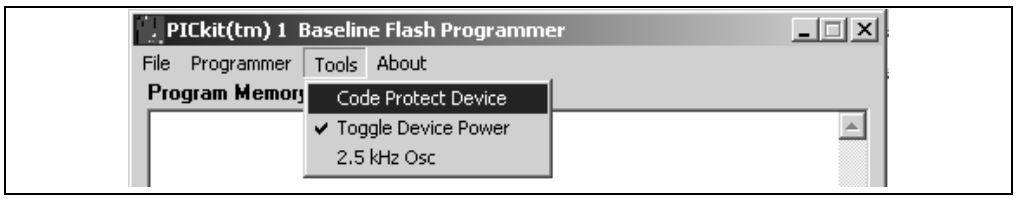
To view the code written to the device, click **Read Device**. The code is displayed in the Program Memory windows for your review.

3.2.6 Code Protect Device

This function enables the code protection features of the device. To protect the code, complete the following steps:

1. Import the hex file.
2. Select *Tools > Code Protect Device*, as shown in Figure 3-8.
3. Click **Write Device**.

FIGURE 3-8: CODE PROTECT DEVICE



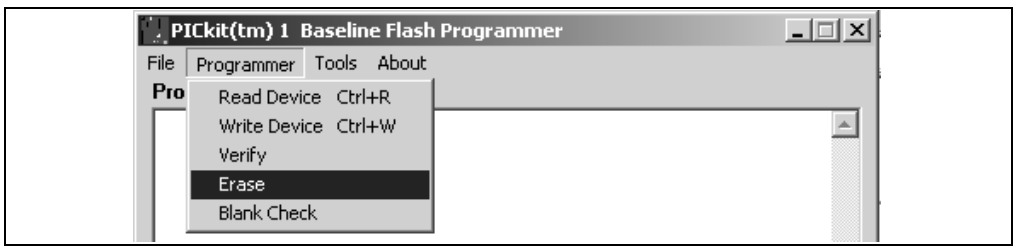
Note: If the device is read after it has been code protected, the Program Memory window displays all zeros.

3.2.7 Erase

The Erase function erases code from the device. However, this function is not normally needed since the Write Device function performs an erase prior to writing code to the device.

To erase the device using the Erase function, click the **Erase** button, or select *Programmer > Erase* from the toolbar menu, as shown in Figure 3-9.

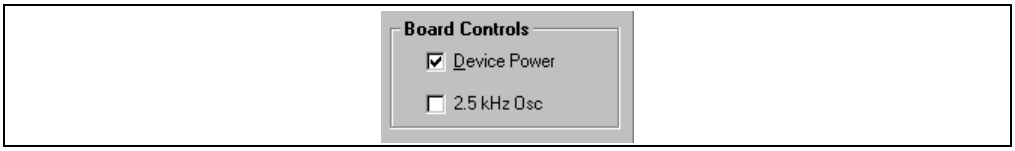
FIGURE 3-9: ERASE



3.2.8 2.5 kHz OSC

The 2.5 kHz OSC is a square wave signal that, when selected, is input to pin 3 of the evaluation socket. See the schematic Figure A-1 in the Appendix A.

FIGURE 3-10: BOARD CONTROLS



NOTES:

Chapter 4. PICkit™ 1 and MPLAB® IDE

4.1 INTRODUCTION

This chapter provides a quick overview on how to create and load a project in MPLAB IDE and program a device with MPLAB IDE. The PIC12F675 programming projects described in **Appendix C. "PIC12F675 Programming Projects"** are used as examples in this chapter.

Detail information on MPLAB IDE can be found in the "*MPLAB® IDE v6.xx Quick Start Guide",* (DS51025) available on the MPLAB IDE CD-ROM included in the kit or from the Microchip Technology Inc. website (http://www.microchip.com).

MPLAB IDE is the integrated development environment supplied by Microchip Technology Inc. for developing software for PIC® microcontrollers. MPLAB IDE is used to:

- Create source code using the built-in editor.
- Assemble, compile and link source code using various language tools. An assembler, linker and librarian come with MPLAB IDE. Supported C compilers are available from Microchip. Third party compilers may be supported also. Check the release notes or readme files for details.
- Debug the executable logic by watching program flow with the built-in simulator, or in real time with the MPLAB ICE 2000 emulator or MPLAB ICD 2 in-circuit debugger. Third party emulators may also be supported. Check the release notes or readme files for details.
- Make timing measurements with the simulator or emulator.
- View variables in watch windows.

4.2 HIGHLIGHTS

- Using the PIC12F675 Programming Project Source Code
- Project Management in MPLAB IDE

4.3 USING THE PIC12F675 PROGRAMMING PROJECT SOURCE CODE

If the PIC12F675 Programming Project source code was installed from the PICkit 1 CDROM, and "Lessons" was selected from the options list, the source and hex files will be installed on the PC harddrive at:

```
C:\PICkit 1\Classic\Lesson n
```

Where n indicates the lesson number.

In order to use the lesson source code you will have to:

1. Create a new or open an existing MPLAB IDE project.
2. Configure the project for the PIC12F675.
3. Add the code to the project.
4. Compile the software.

Details for each of these steps can be found on the "*MPLAB® IDE v6.xx Quick Start Guide*" available on the MPLAB IDE CD-ROM included with the PICkit 1 Flash Starter Kit.

4.4 PROJECT MANAGEMENT IN MPLAB IDE

Work in MPLAB IDE is done by creating or loading a project. A project contains all the information needed to write, compile and program a PICmicro MCU device.

4.4.1 Creating a Project

1. Select *Project > Project Wizard*, as shown in Figure 4-2.
2. Follow the steps in the wizard to create a project.

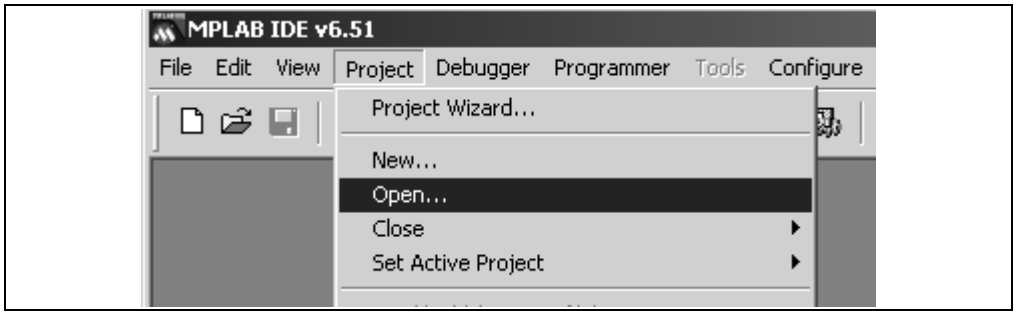
FIGURE 4-1: CREATING A PROJECT

4.4.2 Opening a Project

Opening a project in MPLAB IDE is accomplished with the following steps:

1. Select *Project > Open*, as shown in Figure 4-2.
2. Browse and locate the project.
3. Click **OPEN**.

FIGURE 4-2: OPENING A PROJECT

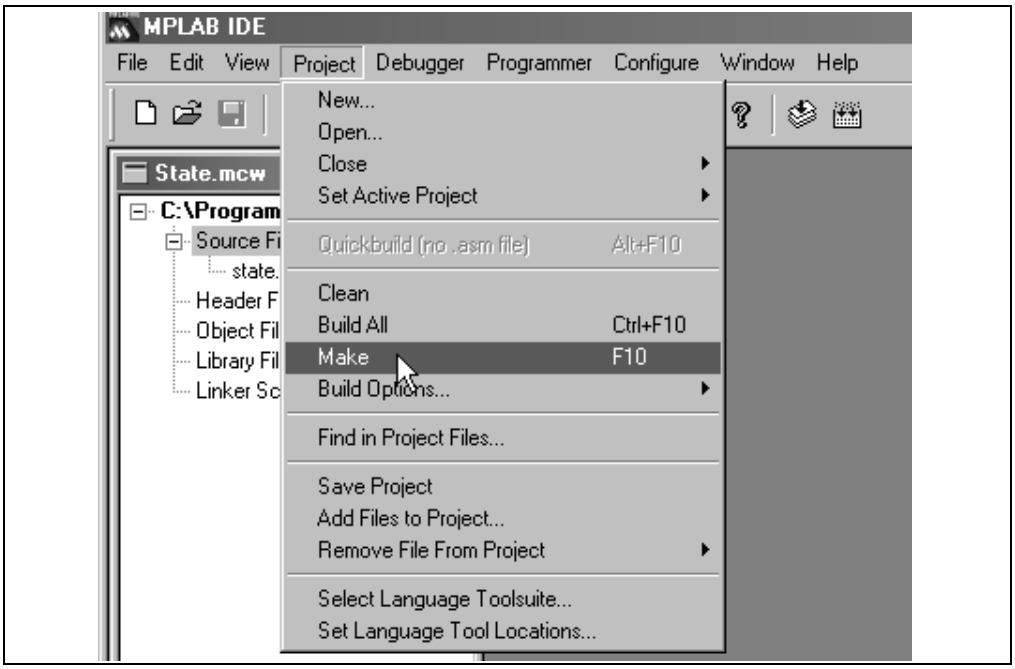


4.4.3 Compiling the Project

After creating or opening a project, it is necessary to compile it. Select *Project > Make*, as shown in Figure 4-3.

A window will appear and show the progress. If there are no errors, the program can be simulated or programmed into the PICkit 1 Flash Starter Kit software. (See **Section 4.4.5 "Programming the Device from MPLAB IDE"**.

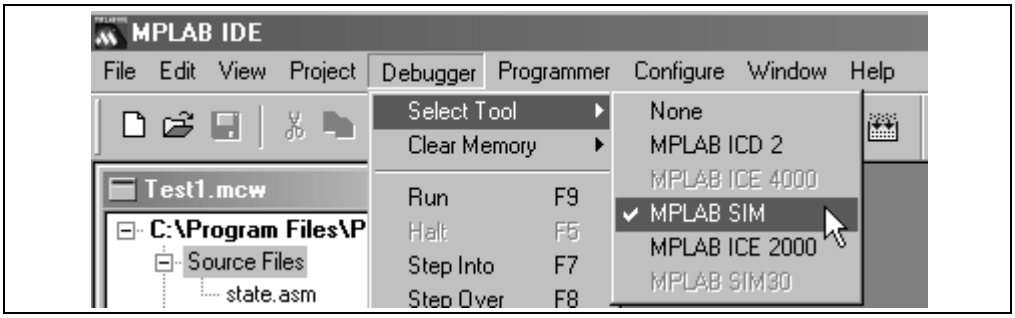
FIGURE 4-3: COMPILE PROJECT



4.4.4 Simulating the Project

In order to simulate the program to see what the PIC microcontroller is actually doing with each instruction, select the MPLAB IDE simulator debugger. Select *Debugger > Select Tool > MPLAB SIM*, as shown in Figure 4-4.

FIGURE 4-4: MPLAB SIMULATION



Once MPLAB SIM is selected, five buttons will appear on the right end of the menu toolbar, as shown in Figure 4-5.

FIGURE 4-5: SIMULATION TOOLBAR

These buttons allow the user to:

1. Run code:
2. Pause code:
3. Single Step code, step into functions:
4. Single Step code, step over functions:
5. Reset code: