Introduction to MPLAB® IDE and HI-TECH C® PRO for the PIC10/12/16 MCU Family – Lite Mode Compiler Tutorial

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.

INTRODUCTION TO MPLAB® IDE AND HI-TECH C® PRO COMPILER

Table of Contents

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.

INTRODUCTION

This chapter contains general information that will be useful to know before using the Introduction to MPLAB® IDE and HI-TECH C® PRO Compiler. Items discussed in this chapter include:

- Document Layout
- The Microchip Web Site
- Customer Support
- Document Revision History

DOCUMENT LAYOUT

This document provides an introduction to the MPLAB® IDE and HI-TECH C® PRO Compiler – Lite Mode.

- **Chapter 1. "Introduction and Project Setup"**
- **Chapter 2. "Testing Code with the MPLAB® SIM Simulator"**
- **Chapter 3. "Checking Program Memory Use"**

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

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support
- Development Systems Information Line

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 (August 2007)

- Initial Release of this Document.

Revision B (September 2008)

- Updated tutorial to be consistent with new compiler release.

Chapter 1. Introduction and Project Setup

1.1 INTRODUCTION

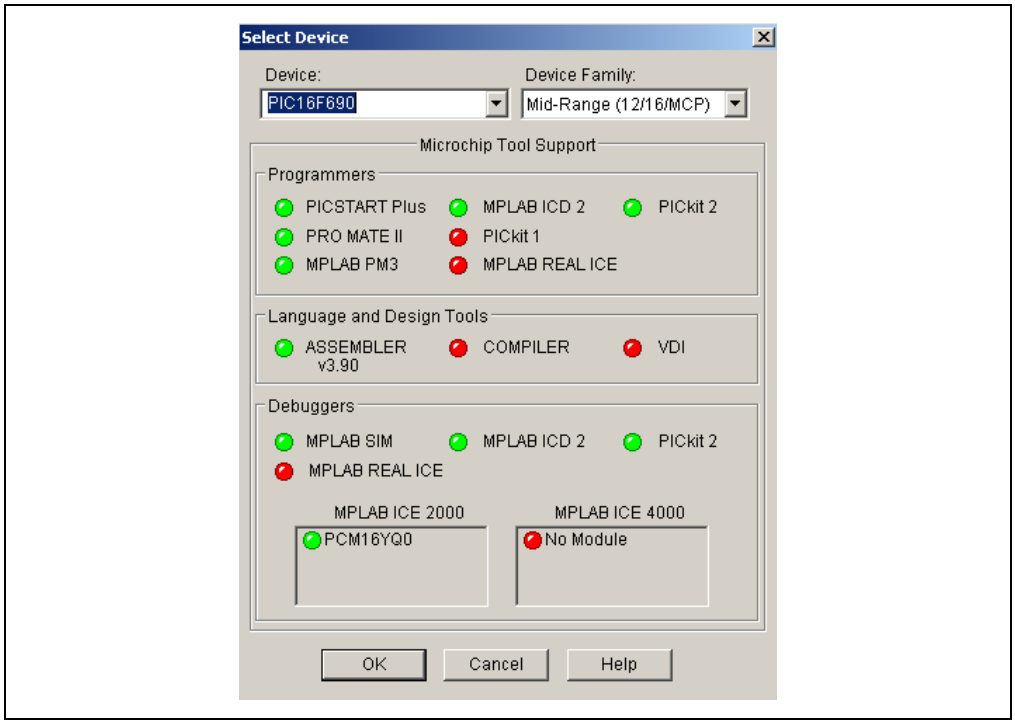
The following document is provided as an introduction to the MPLAB® Integrated Design Environment (IDE) using the free student version HI-TECH C® PRO compiler. The free MPLAB IDE is available for download at www.microchip.com. Download the “Full Zipped” current version of the software to a folder on your computer and install from that directory. The current version comes with the free C PRO compiler. The compiler is also available for download from the HI-TECH web site at http://www.htsoft.com/downloads/demos.php. Locate the current version of C PRO on the page and follow the instructions. You may be required to register with HI-TECH before you can begin the download process. Once all software has been installed, open the MPLAB IDE and follow the instructions below.

1.2 SELECTING THE DEVICE

To show menu selections in this document, the menu item from the top row in the MPLAB IDE will be shown after the menu name like this: *MenuName>MenuItem*. To choose the *Select Device* entry in the *Configure* menu, it would be written as *Configure>Select Device*.

Choose *Configure>Select Device*. Click **OK**. In the Device dialog, select “PIC16F690” from the list if it’s not already selected.

FIGURE 1-1: SELECT DEVICE DIALOG



The “lights” indicate which MPLAB IDE components support this device (see Figure 1-1).

Note: The lights for the assembler and compiler are for Microchip’s tools, not third party tools. These lights will also show what the earliest version of the assembler/compiler these features can be used with. (Hence v3.90 for assembler.)

- A green light indicates full support.
- A yellow light indicates preliminary support for an upcoming part by the particular MPLAB IDE tool component. Components with a yellow light instead of a green light are often intended for early adopters of new parts who need quick support and understand that some operations or functions may not be available.
- A red light indicates no support for this device. Support may be forthcoming or inappropriate for the tool (e.g., dsPIC® DSC devices cannot be supported on MPLAB® ICE 2000).

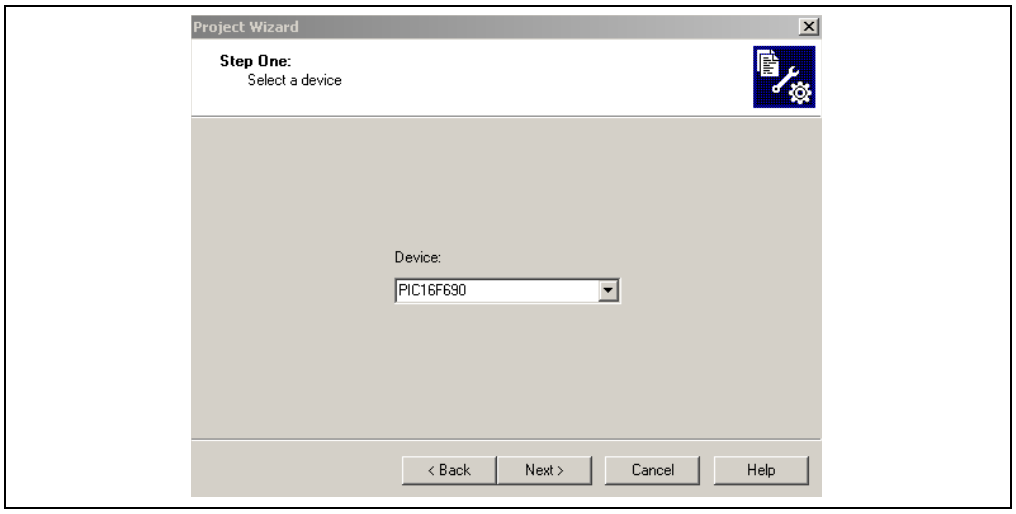
1.3 CREATING THE PROJECT

The next step is to create a project using the Project Wizard. A project is the way the files are organized to be compiled and assembled. We will use a single C code file. Choose *Project>Project Wizard*.

From the Welcome dialog, click on **Next>** to advance.

The next dialog (Step One) allows you to select the device, which we’ve already done. Make sure that it says PIC16F690. If it does not, select “PIC16F690” from the drop-down menu. Click **Next>** (see Figure 1-2).

FIGURE 1-2: PROJECT WIZARD – STEP ONE



1.4 SETTING UP LANGUAGE TOOLS

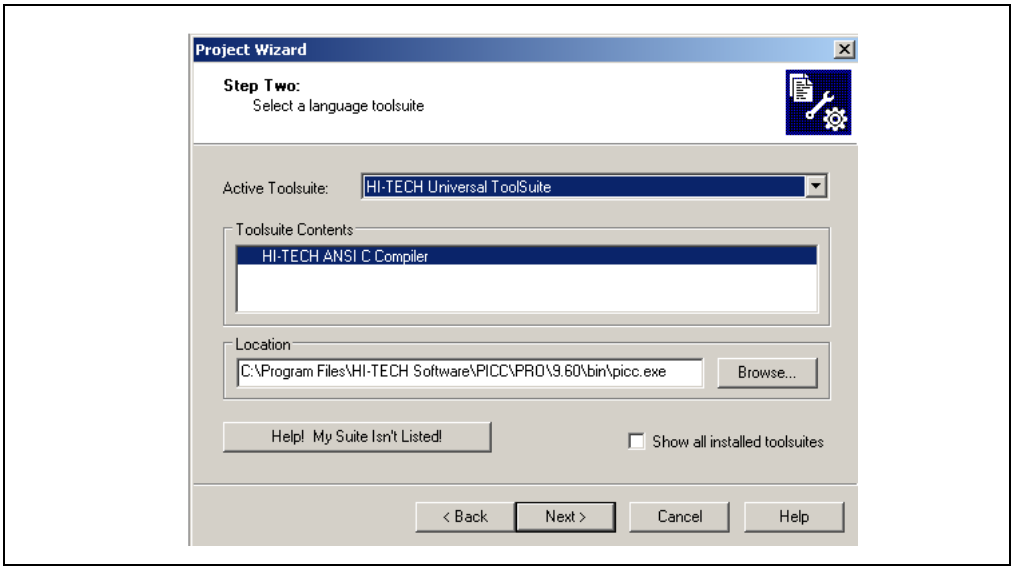
Step two of the Project Wizard sets up the language tools that are used with this project. Select “HI-TECH Universal ToolSuite” in the Active Toolsuite list box (see Figure 1-3).

Then “HI-TECH ANSI C Compiler” should be visible in the Toolsuite Contents box. If MPLAB IDE and the HI-TECH C PRO Compiler were both installed into their default directories, the compiler’s executable will be located in:

`C:\Program Files\HI-TECH Software\PICC\PRO\`version# `\bin\picc.exe`

If this does not show up correctly, use the browse button to set it to the proper file in the MPLAB IDE subfolders.

FIGURE 1-3: PROJECT WIZARD – STEP TWO



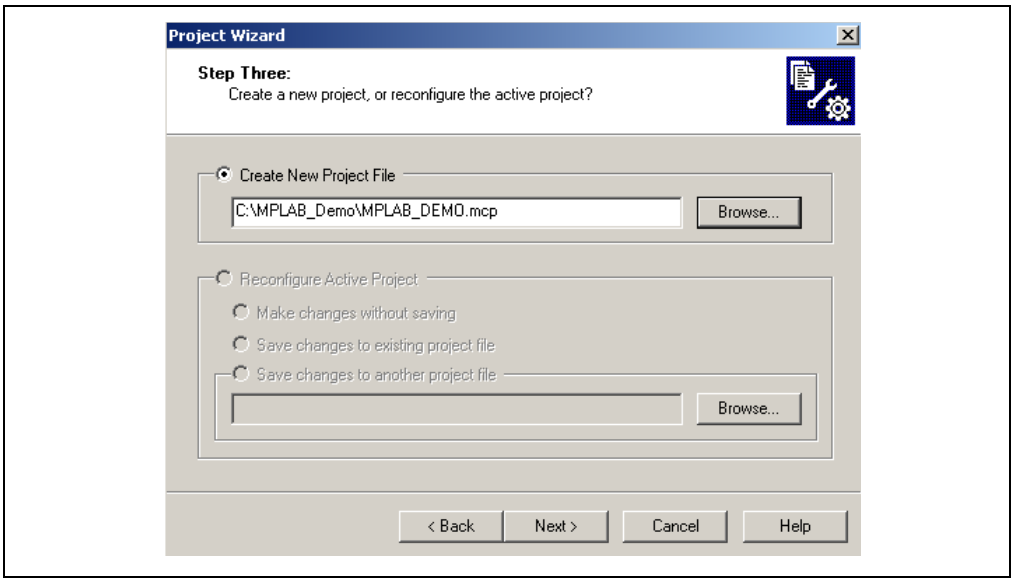
When you are finished, click **Next>**.

1.5 CREATE A NEW PROJECT

Step Three of the Project Wizard allows you to name the project and put it into a folder (see Figure 1-4).

This sample project will be called `MPLAB_DEMO`. Using the Browse button, create a folder for this project (we've used `C:\MPLAB_DEMO`) and place the project in this folder. Click **Next>**.

FIGURE 1-4: PROJECT WIZARD – STEP THREE

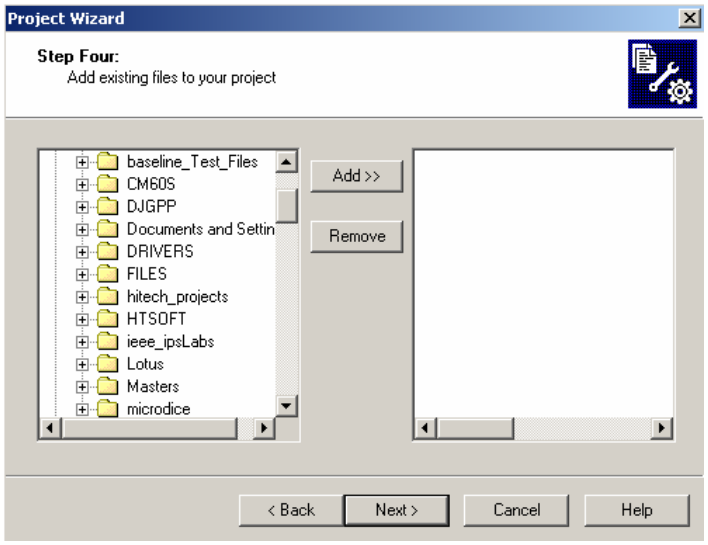


It is important to keep your path small as there are a limited number of characters allowed in the name. Therefore, it is recommended that the `C:\` drive is used for your projects.

1.6 ADDING FILE TO THE PROJECT

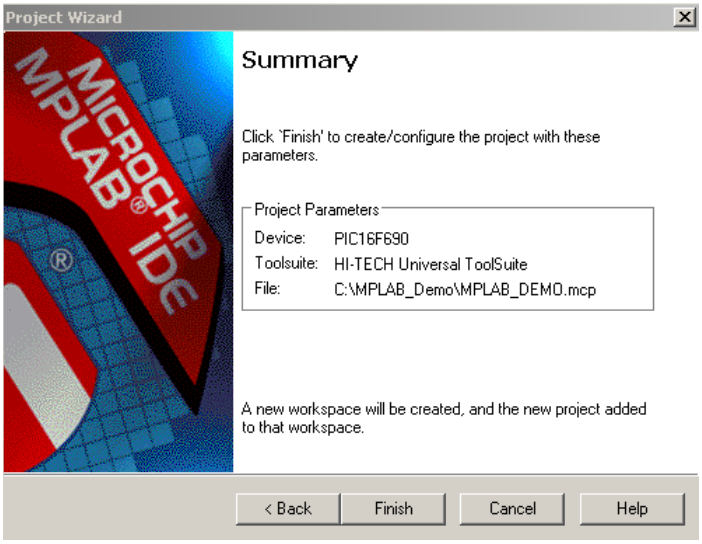
Step Four of the Project Wizard allows file selection for the project. This tutorial will not contain any existing files so simply select **Next** (see Figure 1-5).

FIGURE 1-5: PROJECT WIZARD – STEP FOUR



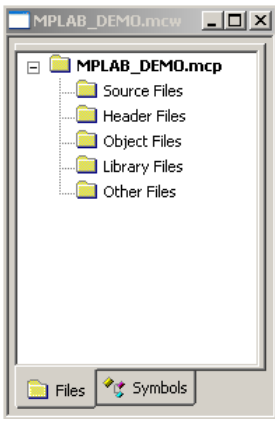
The final screen of the Project Wizard is a summary showing the selected device, the tool suite and the new project file name (see Figure 1-6).

FIGURE 1-6: PROJECT WIZARD – SUMMARY



After pressing the **Finish** button, review the Project window in the MPLAB IDE workspace. It should look like Figure 1-7. If the Project window is not open, select *View>Project*.

FIGURE 1-7: PROJECT DIALOG



Note: Files can be added and projects saved by using the right mouse button in the Project window. In case of error, files can be manually deleted by selecting them and using the right mouse click menu.

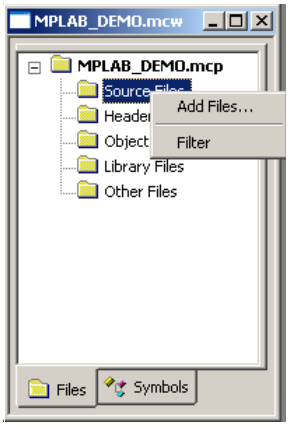
1.7 CREATING A C SOURCE FILE

A C source file will need to be created in order to develop the firmware for an application. Select *File>New*. A new window will now appear on the workspace called Untitled (see Figure 1-8). Save this file as `MPLAB_DEMO.c` in the Project folder created in the Project Wizard (be sure to include the *.c* extension):

`C:\MPLAB_DEMO\MPLAB_DEMO.c`

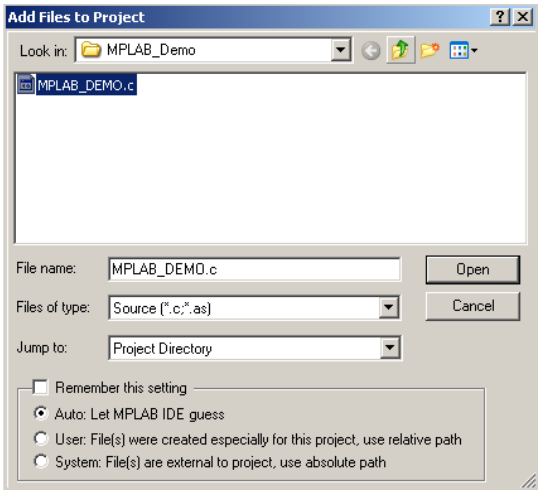
This new C file will need to be added to the project. In the Project window, right click on **Source Files** and select **Add Files** (see Figure 1-8).

FIGURE 1-8: ADD FILES



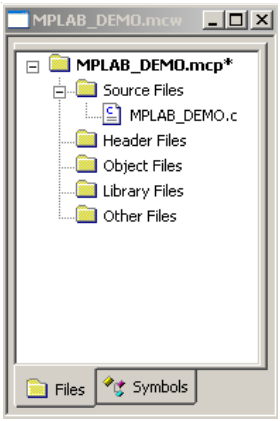
Browse to the `C:\MPLAB_DEMO` directory and add the file that was created (see Figure 1-9).

FIGURE 1-9: ADD FILES TO PROJECT DIALOG



The Project window should now contain the C source file as shown in Figure 1-10.

FIGURE 1-10: FILES



Open the MPLAB_DEMO.c source file by double clicking it within the Project window. Once opened, copy and paste the lines of code in Example 1-1 and save:

EXAMPLE 1-1: CODE EXAMPLE 1

```
#include   <pic.h>
/*HI-TECH include file to determine appropriate
    device register definition header file.
    Include this file whenever using the PICC-LITE
    or other HI-TECH PICC compilers.
*/


//----------------DEVICE CONFIGURATION
/***************Configure device as follows:

    Use the internal 8MHz oscillator (INTIO)
    Disable the Watchdog Timer (WDTDIS)
    Disable the Power-up Timer (PWRTDIS)
    Disable the Master Clear Reset (MCLRDIS)
    Disable Code Protect (UNPROTECT)
    Disable Brown-out Reset (BORDIS)
    Disable Internal External Switch Over Mode (IESODIS)
    Disable Fail-Safe Clock Monitor (FCMDIS)

    ***For a detailed description of each device configuration
        above refer to the pic16f690 datasheet (DS41262D)
        "Special Features of the CPU" section.

        available for download at www.microchip.com

*/

__CONFIG(INTIO & WDTDIS & PWRTDIS & MCLRDIS & UNPROTECT & BORDIS &
IESODIS & FCMDIS);

unsigned char Count = 0; //create an 8-bit variable in Data
                         //Memory called Count

//------------------PROGRAM MEMORY

void main(void)
{}
```

1.8 BUILDING THE PROJECT

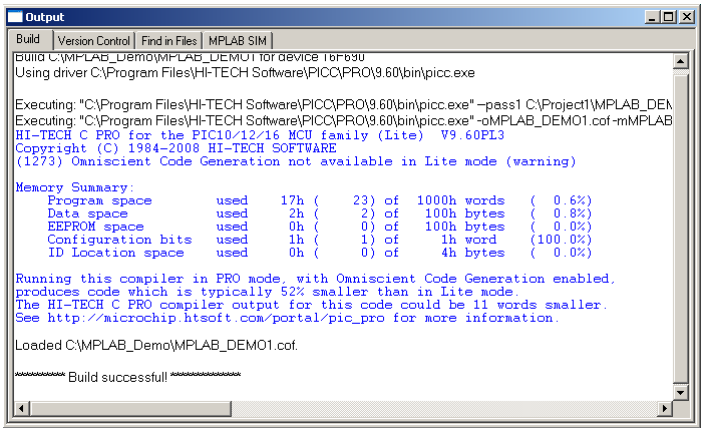
From the Project menu, we can compile the current files.

To build the project, select either:

- *Project>Build*
- Right click on the project name in the Project window and select “Build All”.
- Click the **Build Project** icon  on the Project toolbar. Hover the mouse over the icons to see the pop-up text of what they represent.

The Output window shows the result of the build process as seen in Figure 1-11. There should be no errors on any step and the warnings can be ignored.

FIGURE 1-11: OUTPUT WINDOW



1.9 CREATING CODE

Return to the `MPLAB_DEMO.c` source file.

TIP: Line numbers are shown here. Line numbers may be toggled on/off by right clicking in the Editor window, selecting “Properties”, and then checking/unchecking “Line Numbers” on the **Editor** tab of the Editor Options dialog.

The device configuration comments and code (`__CONFIG`) in the first part of the file is for more advanced functions such as setting up Configuration bits in a final application. These details can be ignored at this point with focus on writing main function code. The new code will be placed within the `main()` function braces.

When any source file is opened, you are automatically in the Editor. Modify the `main()` as follows in Example 1-2.

EXAMPLE 1-2: CODE EXAMPLE 2

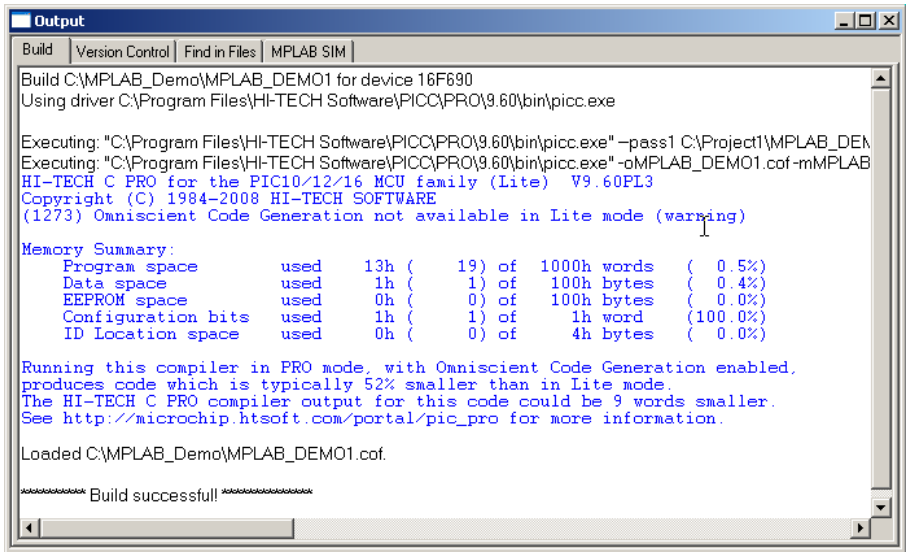
```
void main(void)
{
    ADCON0 = 0;   //Initialize a register called ADCON0 to zero
                  //This is a special function register (SFR) located
                  //at a defined Data Memory address

    while(1)      //Create an infinite loop to hold the repetitive
                  //section of this function
    {
    Count = Count + 1;  //Increment the Count variable by 1
    ADCON0 = Count;     //Assign the Count variable value
                        //to the ADCON0 SFR

    }

}
```

Note: The ADCON0 register is normally used to configure the Analog-to-Digital Converter (ADC) peripheral. In this tutorial we can ignore the ADCON0 function and just consider it a register in Data Memory.

1.10 BUILDING THE PROJECT AGAIN

Select *Project>Build* to compile the code. The Output window (shown in Figure 1-12) will show errors, if any, in your code as well as information concerning memory usage and device configuration. To view the specific line of code that generated the errors/message, double click on the specific line in the Output window.

FIGURE 1-12: OUTPUT WINDOW



Upon a successful build, the output file generated by the language tool will be loaded. The `MPLAB_DEMO.cof` file can be found in the project directory. This file contains the object code that can be programmed into a PIC® MCU and debugging information so that source code can be debugged and source variables can be viewed symbolically in Watch windows as shown later in this tutorial.

NOTES:

Chapter 2. Testing Code with the MPLAB® SIM Simulator

2.1 TESTING CODE WITH THE SIMULATOR

In order to test the code, software or hardware is needed that will execute the PIC® MCU instructions. A debug execution tool is a hardware or software tool that is used to inspect code as it executes a program. Hardware tools such as MPLAB ICE, MPLAB ICD 2 or PICkit™ 2 Debug Express can execute code in real devices. If hardware is not available, the MPLAB simulator can be used to test the code. For this tutorial use the MPLAB SIM simulator.

The simulator is a software program that runs on the PC to *simulate* the instructions of the PIC® MCU. It does not run in "real time", since the simulator program is dependent upon the speed of the PC, the complexity of the code, overhead from the operating system and how many other tasks are running. However, the simulator accurately *measures* the time it would take to execute the code if it were operating in real time in an application.

Note: Other debug execution tools include MPLAB ICE 2000, MPLAB ICE 4000 and MPLAB ICD 2. These are optional hardware tools to test code on the application PC board. Most of the MPLAB IDE debugging operations are the same as the simulator, but unlike the simulator, these tools allow the target PIC MCU to run at full speed in the actual target application.

Select the simulator as the debug execution tool. This is done from the *Debugger>Select Tool* pull-down menu. After selecting "MPLAB SIM", the following changes should be seen (refer to Figure 2-1):

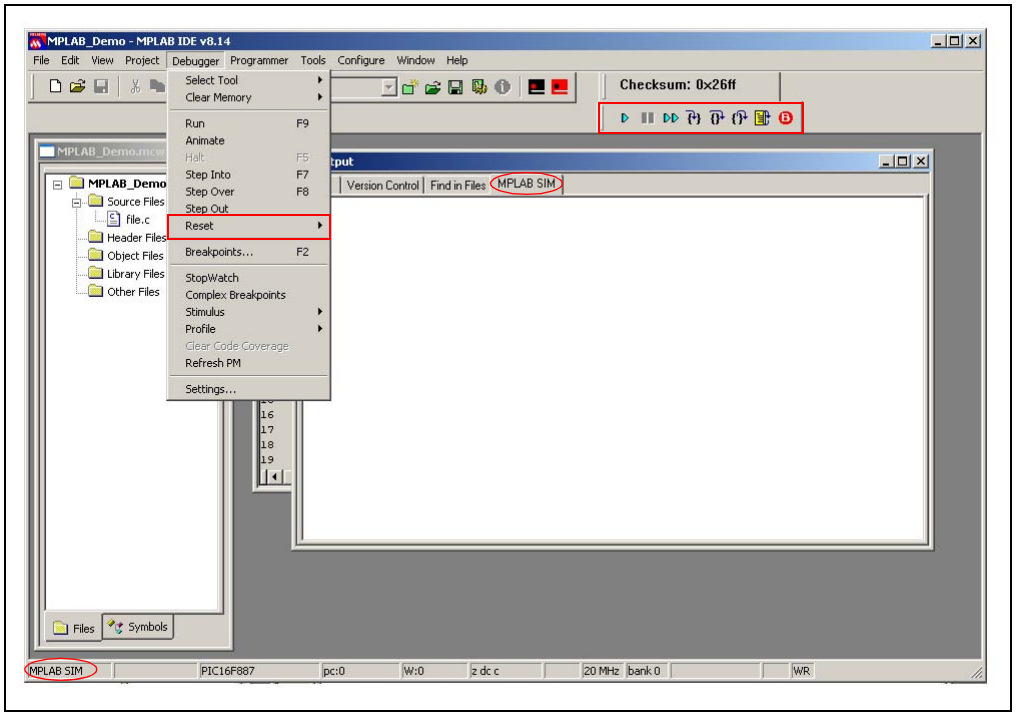
1. The status bar on the bottom of the MPLAB IDE window should change to "MPLAB SIM".
2. Additional menu items should now appear in the Debugger menu.
3. Additional toolbar icons should appear in the Debug Tool Bar.

TIP: Position the mouse cursor over a toolbar button to see a brief description of the button's function.

4. An MPLAB SIM tab is added to the Output window.

Note: To display line numbers to the left side of the code select *Edit>Properties*. Click on the 'C' File Types tab and select the Line Numbers checkbox.

FIGURE 2-1: MPLAB® SIMULATOR



Next, select *Debugger>Reset*. The following reset options are available:

- $\overline{\text{MCLR}}$ Reset
- Watchdog Timer Reset
- Brown-out Reset
- Processor Reset

Next, select Debugger>Reset>Processor Reset. The other options will become clear with time.

To single step through the application program, select *Debugger>Step Into*. This will execute the currently indicated line of code, and move the arrow to the next line of code to be executed.

There are shortcuts for these commonly used functions in the Debug Tool Bar.

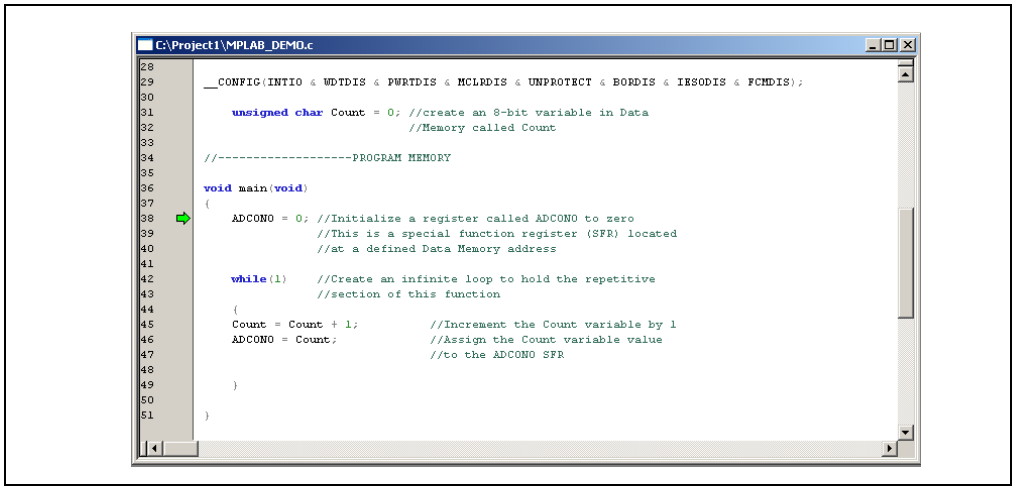
TIP:	See Table 2-1 for commonly used functions and their shortcuts. Click on the appropriate icon on the toolbar or use the hot key shown next to the menu item. This is usually the best method for repeated stepping.

TABLE 2-1: DEBUGGER MENU ITEMS

Debugger Menu	Toolbar Buttons	Function	Hot Key
Run		Run Program	F9
Halt		Halt Program Execution	F5
Animate		Continually step into instruction. To Halt, use *Debugger>Halt* or the Halt icon	
Step Into		Step into the next instruction	F7
Step Over		Step over the next instruction	F8
Step Out Of		Step out of the subroutine	
Reset		Perform a $\overline{\text{MCLR}}$ Reset	F6

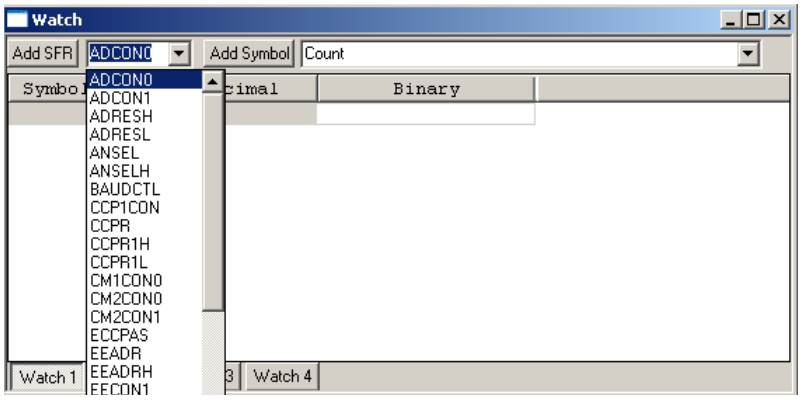
Next, press the **Step Into** icon or select *Debugger>Step Into* until the green program counter arrow points to the `unsigned char Count = 0;` line in the `main( )`.

FIGURE 2-2: MPLAB® IDE EDITOR



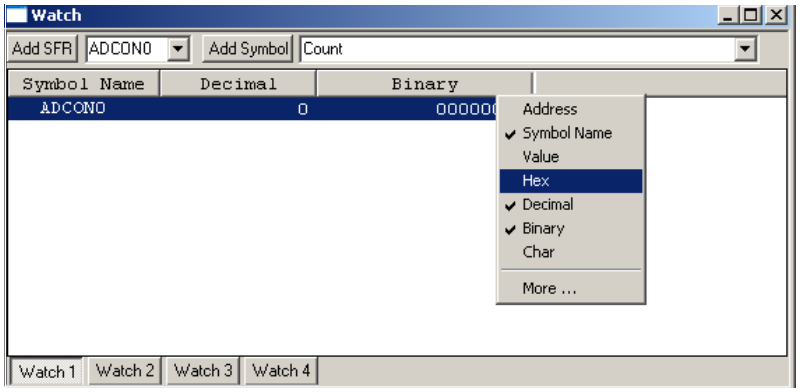
Select *View>Watch* to bring up an empty Watch window. There are two pull downs on the top of the Watch window. The one on the left labeled “Add SFR” can be used to add the Special Function Register, ADCON0, into the Watch. Select **ADCON0** from the list and then click **Add SFR** to add it to the window (see Figure 2-3).

FIGURE 2-3: WATCH WINDOW



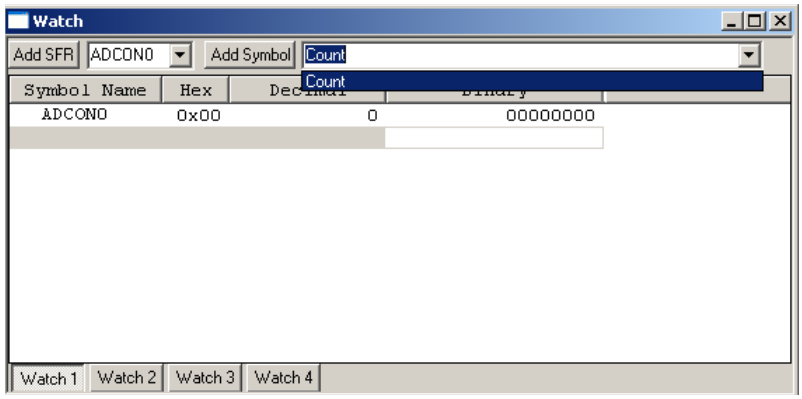
The Watch window allows you to view the contents of registers in Data Memory in the number system of your choice. Ensure that Hex, Decimal and Binary representations are visible. If not, right click on the Symbol Name or anywhere else on the tab bar and select the numbering systems not shown (see Figure 2-4).

FIGURE 2-4: WATCH WINDOW OPTIONS



The pull down on the right of the **Add Symbol** button, allows symbols (variables from your code) to be added from the program. Use this pull down to add the `Count` variable into the Watch window. Select Count from the list and then click **Add Symbol** to add it to the window.

FIGURE 2-5: SELECTING A SYMBOL



Step through the code using the **Step Into** icon on the toolbar . `Count` and `ADCON0` values should change as each line of code is executed.

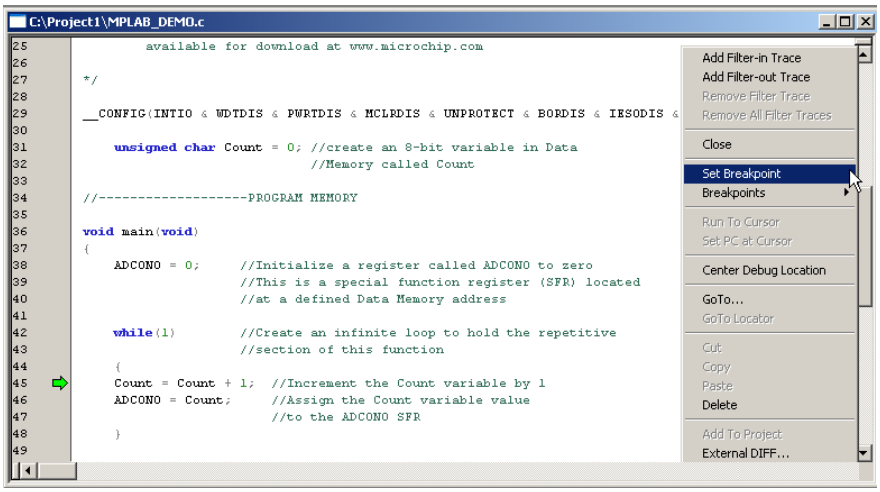
Note: Anytime a register value does change, the value font color turns red.

You could continue single stepping through the code, but instead, set a breakpoint just before the first value is moved into `ADCON0`. To set a breakpoint, put the cursor on the following line:

```
ADCON0 = Count;
```

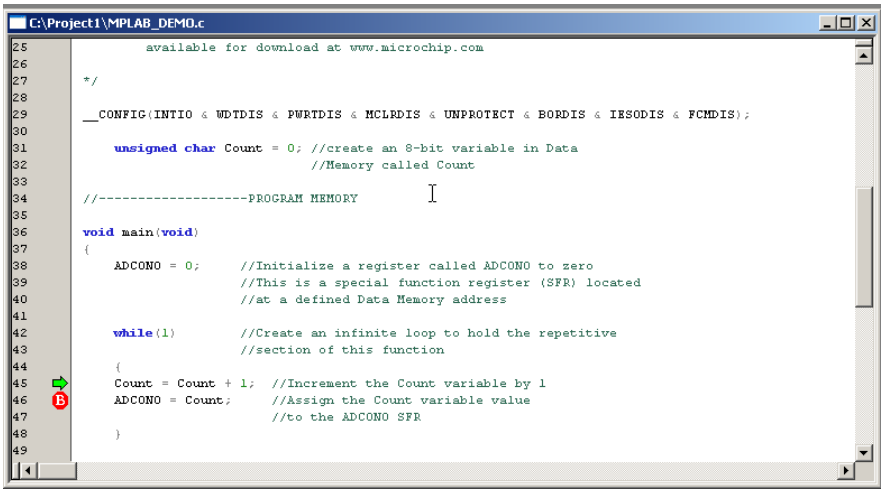
Right click the mouse and select **Set Breakpoint** from the context menu (see Figure 2-6).

FIGURE 2-6: MPLAB® IDE EDITOR



A red “B” will show on the line (see Figure 2-7). (You can also double-click on a line to add a breakpoint.)

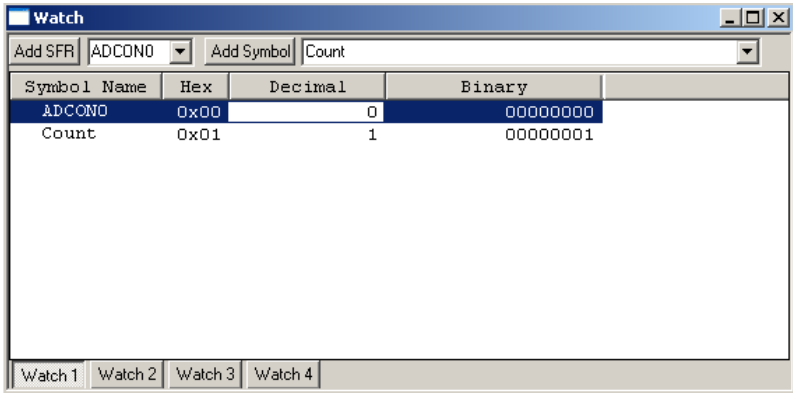
FIGURE 2-7: BREAKPOINT EXAMPLE



Select *Debugger> Processor Reset*. Selecting Reset will reset the processor, the registers, and simulator time. Select *Debugger>Run* to run the application. A text message "Running…" will briefly appear on the status bar at the bottom of the MPLAB IDE window before the application halts at this first breakpoint.

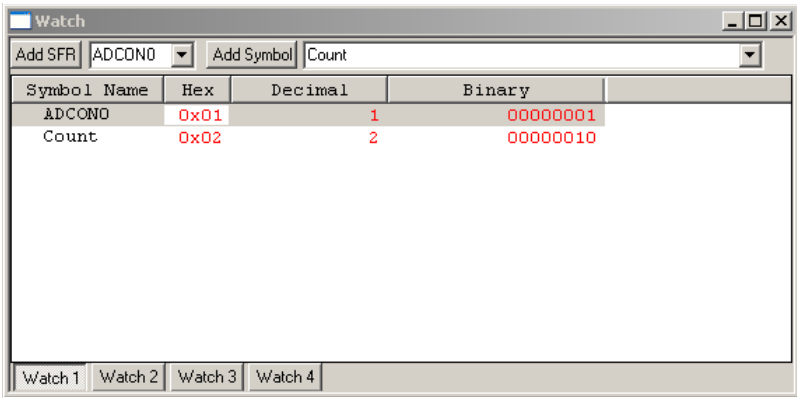
The Watch window should now show that the variable count was incremented by one, but since the breakpoint is at the line before the `ADCON0=Count`; executes, `ADCON0` still has a value of zero (see Figure 2-8).

FIGURE 2-8: WATCH WINDOW



Press the **Run** icon to execute the code until it hits this point again. The Watch window should now show both `ADCON0` and `Count` values increased by one.

FIGURE 2-9: WATCH WINDOW



This would seem to indicate that the program is working as designed. You can single step through the code, or run the code more times to verify that it is executing properly. You could also select the **animate** icon. Animate causes the debugger to actually execute steps while running, updating the values of the registers as it runs. Animate runs slower than the **Run** function.

NOTES:

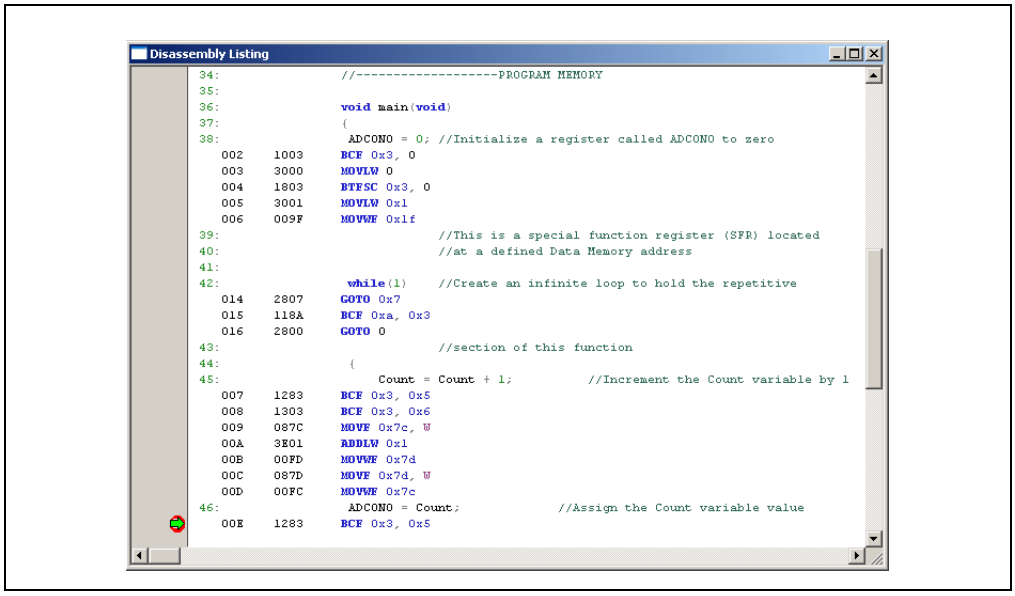
Chapter 3. Checking Program Memory Use

3.1 CHECKING PROGRAM MEMORY USE

A program written in C will generate some overhead in Program Memory. Each address in Program Memory will essentially hold one line of Assembly code. When the C project is compiled, a list file (`.lst`) is generated showing Assembly and Machine Language translations of the C code. The list file can be used to assess how much Program Memory is used.

To view the list file, select *View>Disassembly Listing*. The list file for the code created in this tutorial is shown in Figure 3-1. Note that the breakpoint also appears in the disassembly listing. As you may have guessed, this file can be used to step through code with the debugger enabled.

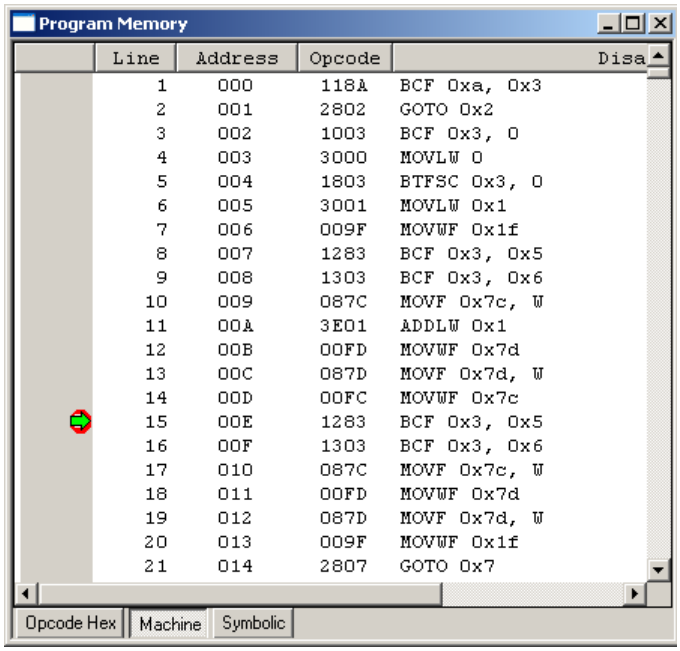
FIGURE 3-1: DISASSEMBLY LISTING



The list file shows the assembly overhead generated for each line of C code. The address in Program Memory occupied by each assembly instruction including its hexadecimal equivalent (Machine Code) is also shown. These hexadecimal values will make up a `.hex` file that will be downloaded onto the PIC Microcontroller using the PICkit 2 programmer.

Program Memory usage can also be viewed by selecting *View>Program Memory* as shown in Figure 3-2.

FIGURE 3-2: PROGRAM MEMORY WINDOW



Once again, both Machine and Assembly languages are shown including the address in Program Memory occupied.

3.2 CONCLUSION

The preceding tutorial provided an introduction to the MPLAB IDE using the HI-TECH C PRO compiler. For more information please refer to www.microchip.com for application notes and software examples.

NOTES:

WORLDWIDE SALES AND SERVICE

AMERICAS

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

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

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

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

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

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

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

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

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

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

ASIA/PACIFIC

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

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

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

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

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

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

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

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

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

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

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

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

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

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

ASIA/PACIFIC

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

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

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

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

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

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

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

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

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

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

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

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

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

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

EUROPE

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

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

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

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

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

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

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

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

01/02/08