Microchip Technology Incorporated

An Introduction to the MPLAB®
Integrated Development Environment

This seminar is an introduction to the MPLAB Integrated Development Environment.

My name is Darrel Johansen and I'm a manager for the Development Systems Group here at Microchip Technology.

What Is MPLAB® IDE?

MPLAB IDE is a software program that runs on your PC to provide a development environment for your embedded system design.

www.microchip.com/mplab

What is MPLAB IDE?

MPLAB IDE is a software program that runs on your development environment for your embedded microcontroller design.

You can download MPLAB IDE for free at microchip.com/mplab.

Design Cycle

1. High Level Design
2. Software Coding
3. Generate Executable
4. Test
5. "Burn" into Device

The design cycle for developing an embedded controller application is:

1) Do the high level design – From the features and performance desired, decide which PICmicro or dsPIC device you need, then design the associated hardware circuitry.

2) Knowing which peripherals and pins control your hardware, write the software. Use either assembly language, which is directly translatable into machine code, or using a compiler that allows a more natural language for creating programs. With these Language Tools you can write and edit code that is more or less understandable, with constructs that help you organize your code.

3) Compile or assemble the software using a Language Tool to convert your code into machine code for the PICmicro device. This machine code will eventually becomes firmware, the code programmed into the microcontroller.

4) Test your code. Usually a complex program does not work exactly the way you might have imagined, and "bugs" need to be removed from your design to get it to act properly.

5) "Burn" your code into a microcontroller and verify that it executes correctly in your finished application.

This seminar will show you how MPLAB can do steps 1 through 4.

Step 5 will require some hardware and will be covered in another seminar.

First Project

● Create Project

● Build Project
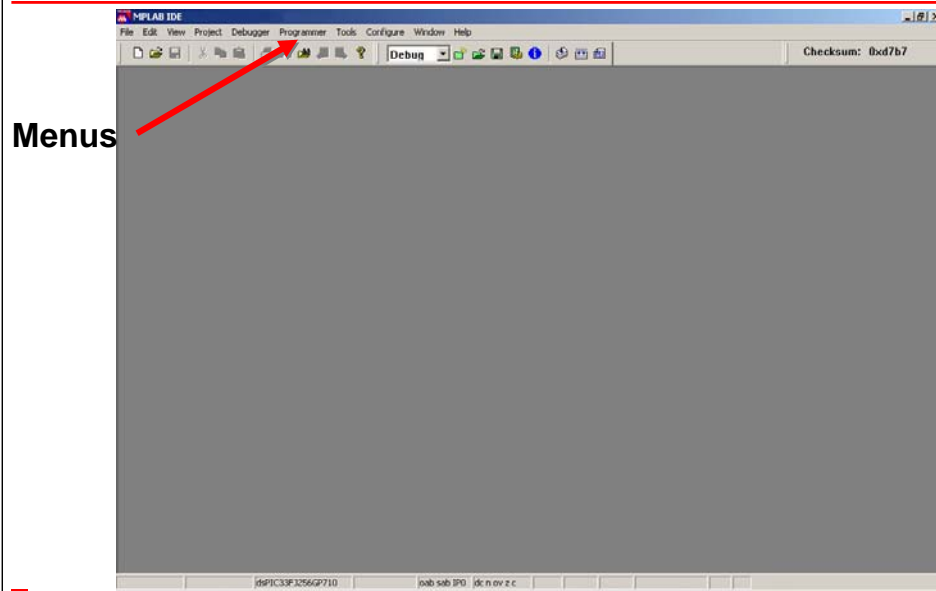
● Test Code with Simulator

This exercise will show you how to put together a project framework, how to build an application, and
how to test it with the simulator to verify that it works as expected.
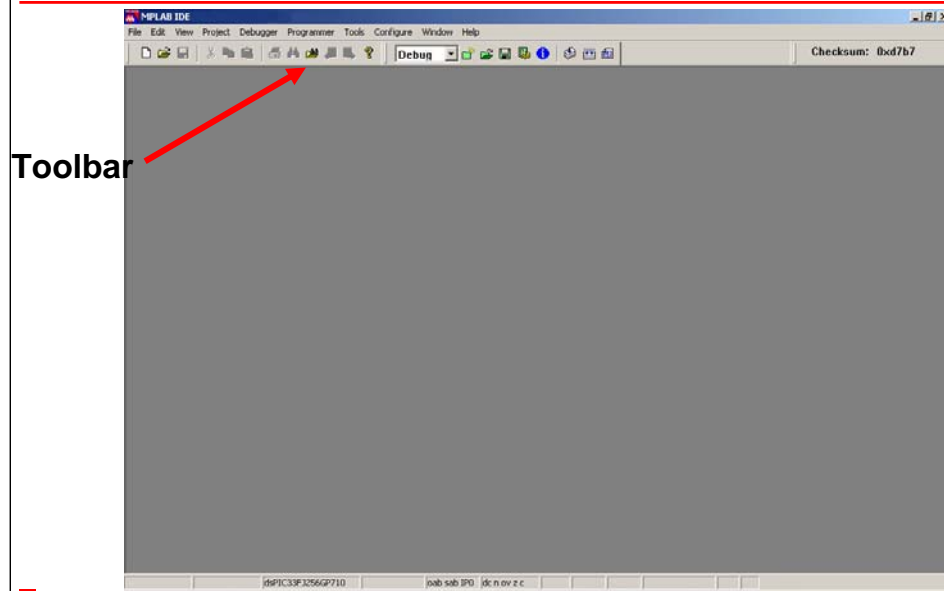
MPLAB Desktop
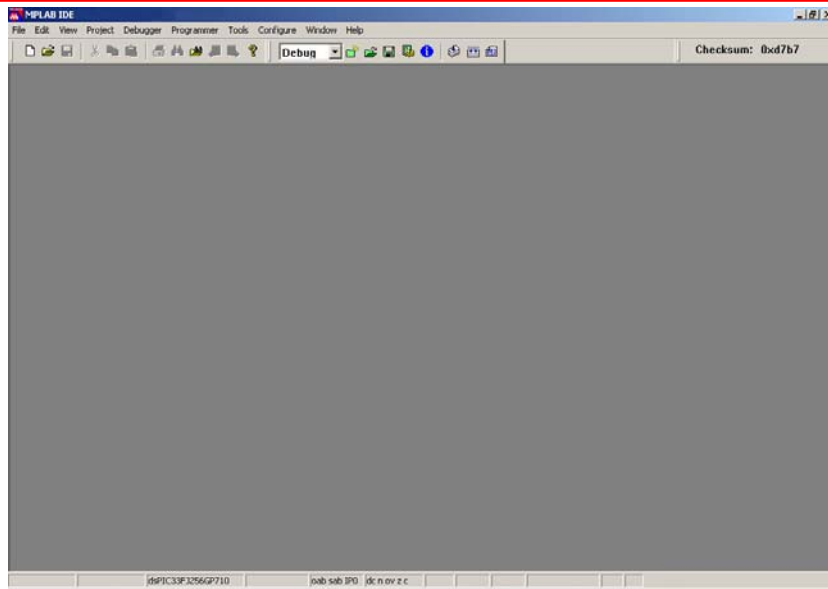
After you have downloaded and installed MPLAB IDE, you can start it up, and you'll see a standard windows graphical user interface…
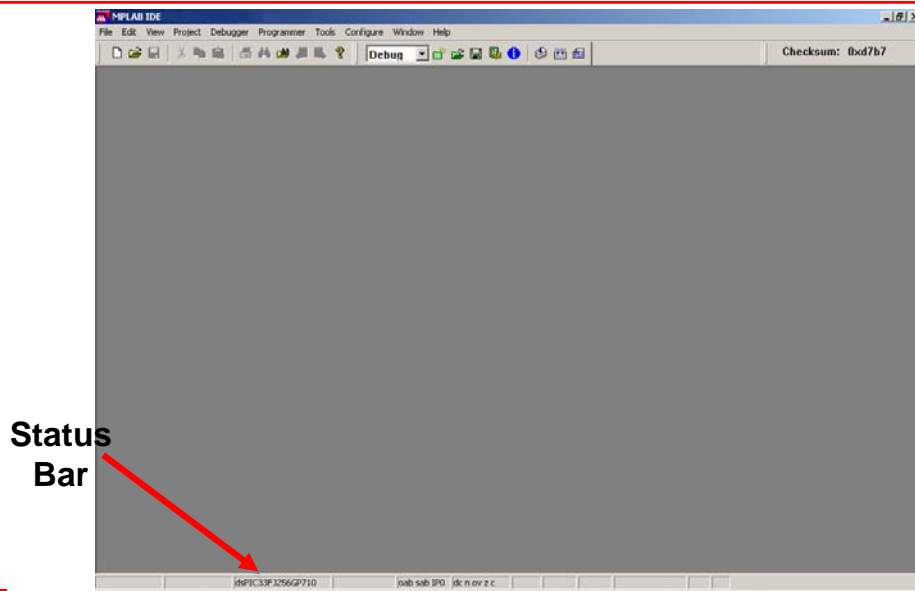
MPLAB Desktop

Menus

with menus…

MPLAB Desktop

Toolbar
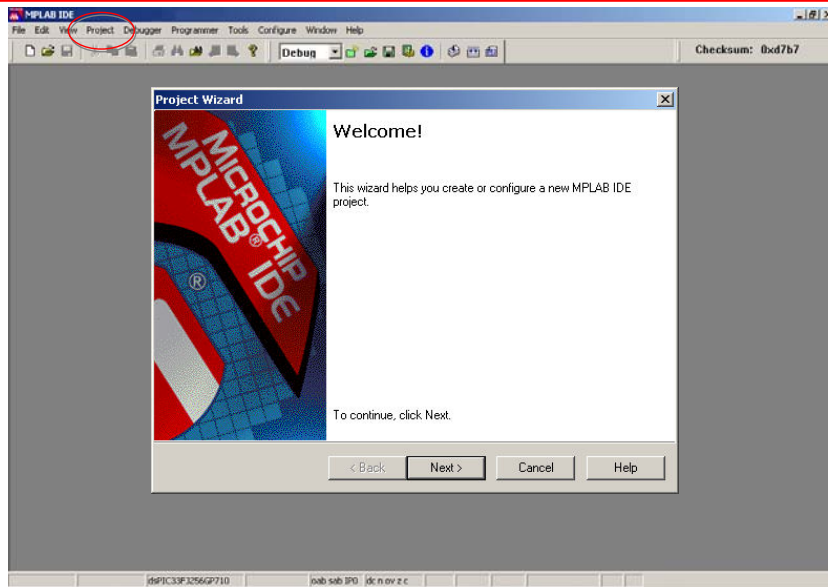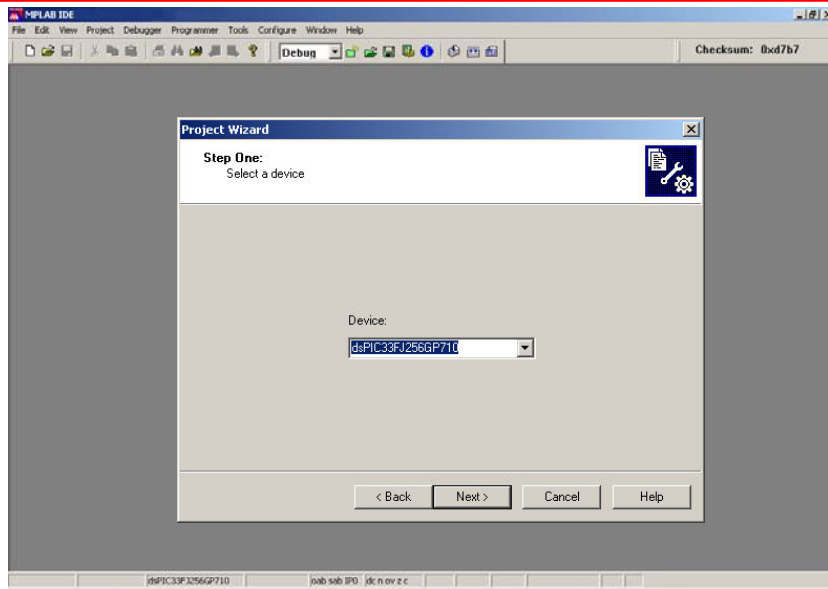
toolbar icons…

… and a status bar.

MPLAB Desktop

To get started, we'll use the Project Wizard, which is selected from the Project menu.

Many of our development boards and starter kits come with finished projects that you can load and examine.
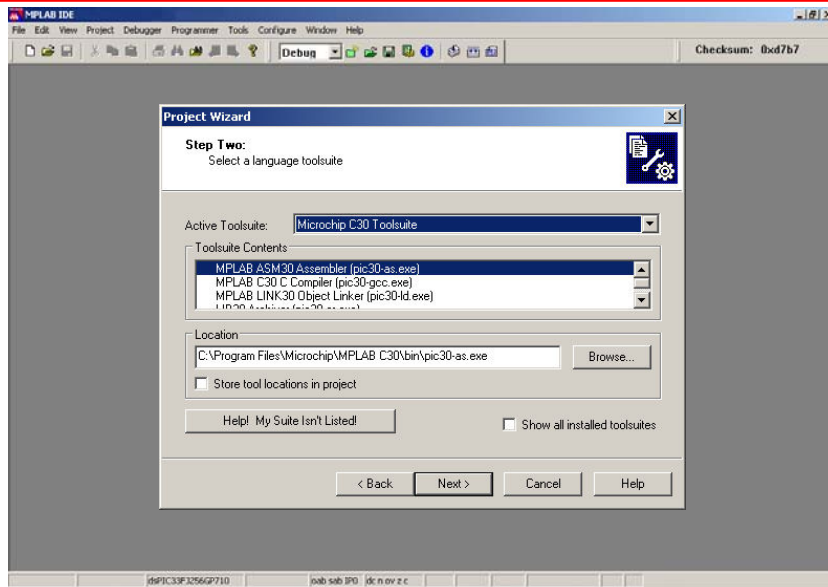
The Project Wizard makes it very easy to set up a new project.

Create A Project

An Introduction to MPLAB IDE   10

The first thing the wizard asks for is to define which processor you are using. We'll use one of the 16-bit dsPIC Digital Signal Controllers for this project.
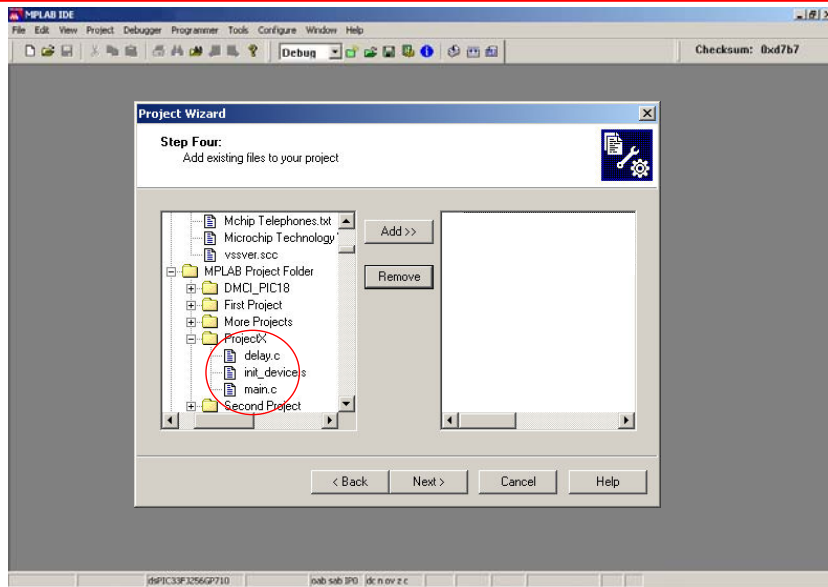
Project Wizard

If you have installed any C compilers, they will show up in this list. If not, you may only see the free assemblers that come with MPLAB IDE. Note that all of Microchip's C Compilers can be downloaded for free evaluation.

Project Wizard

An Introduction to MPLAB IDE   12
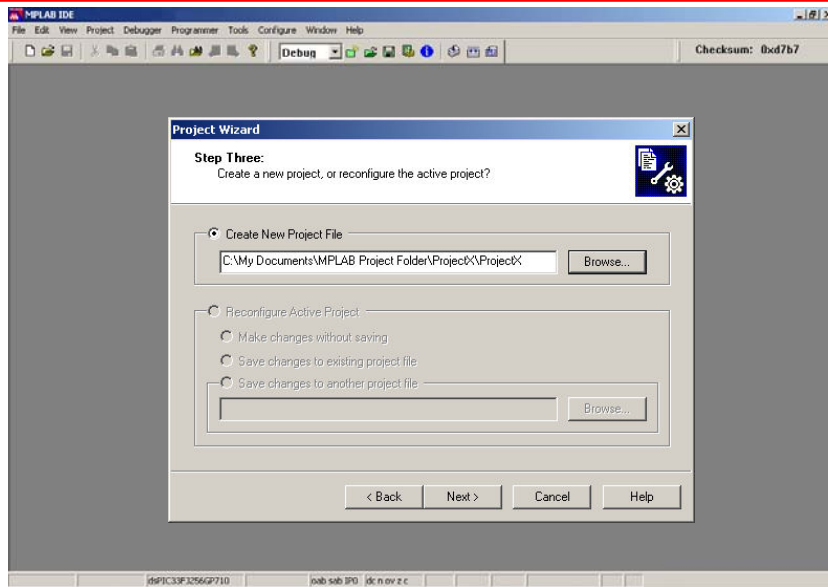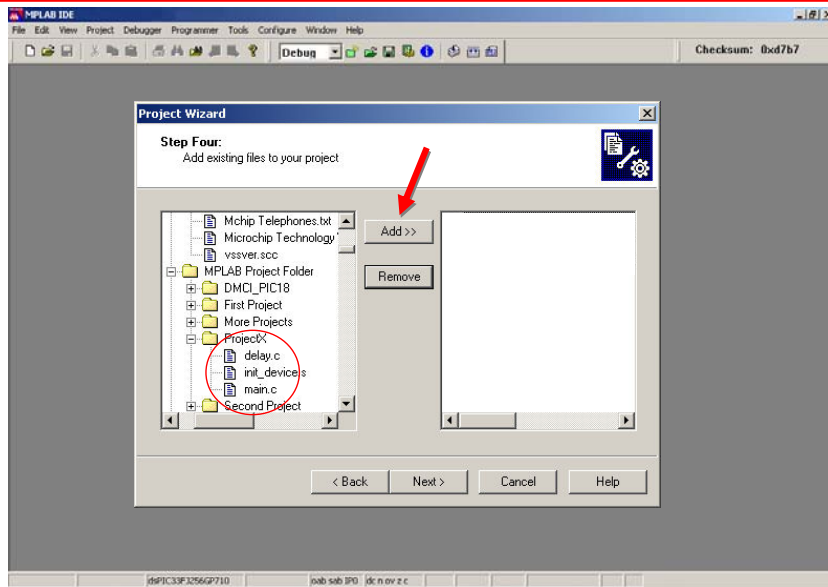
Next we'll browse to the folder to store our project and name it "ProjectX."
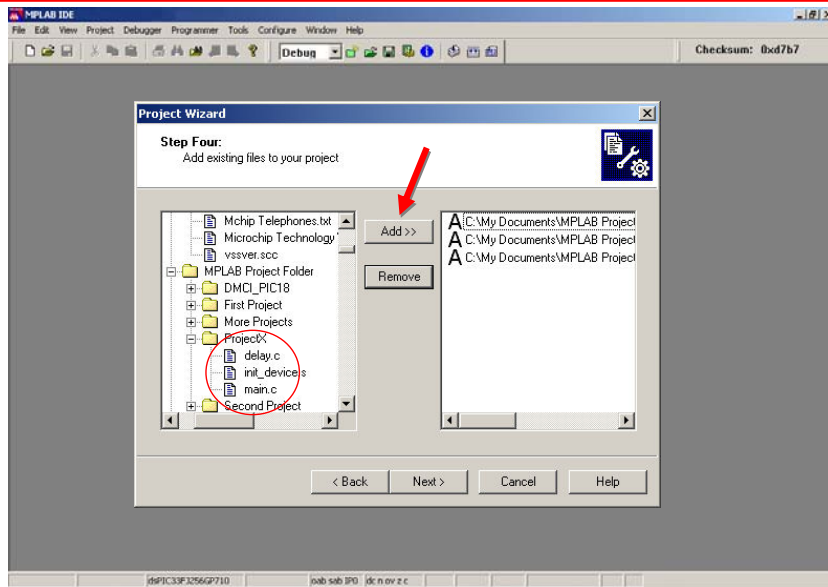
Project Wizard

We have already written the code for this project, so we simply add them to our project, selecting them from the left file listing…

Project Wizard

…and pressing the "Add>>" button…

Project Wizard

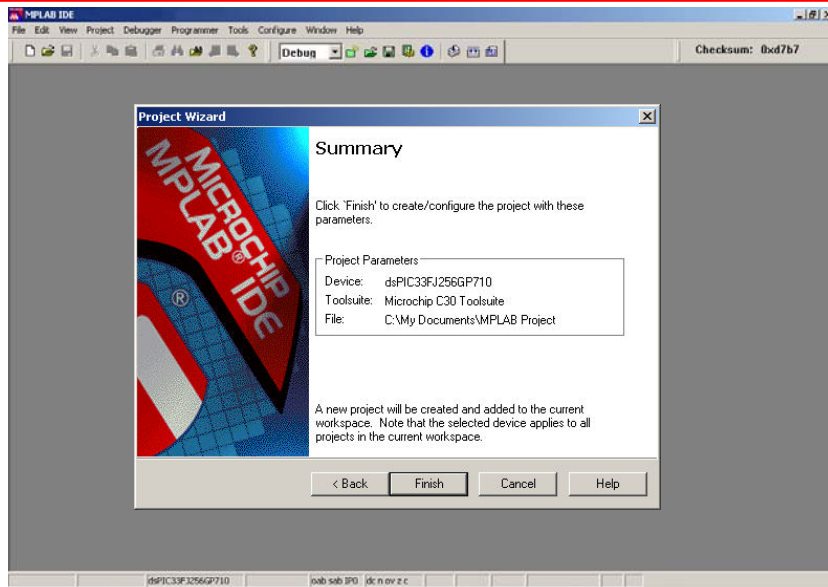An Introduction to MPLAB IDE   15
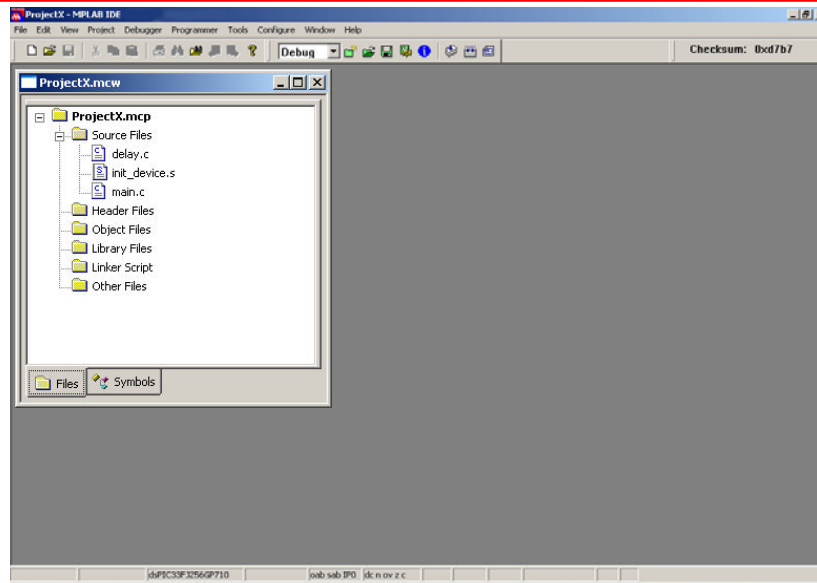
… to make them show up in the right pane..

If you had not written any code yet, you easily can add files to the project later.
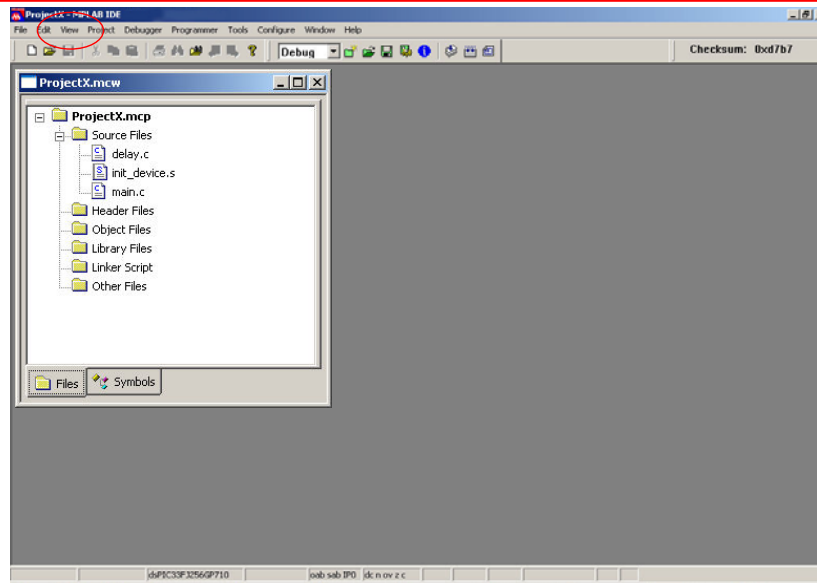
Project Wizard

And that's all there is to creating the project. The summary screen tells us that we're using a dsPIC33 processor with MPLAB C for the PIC30 (also called C30), and that the project is in a folder on drive C:.
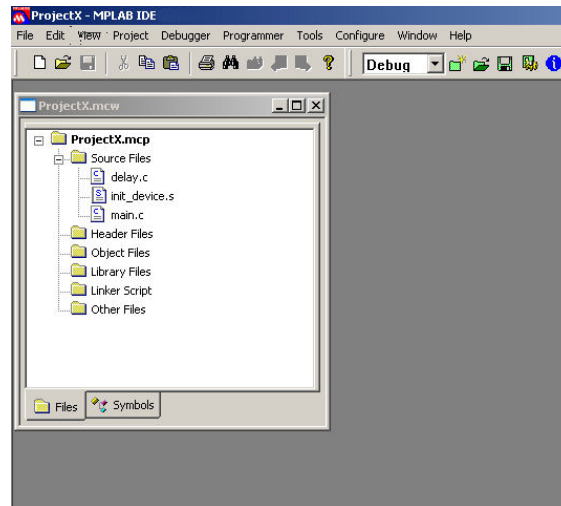
Wizard

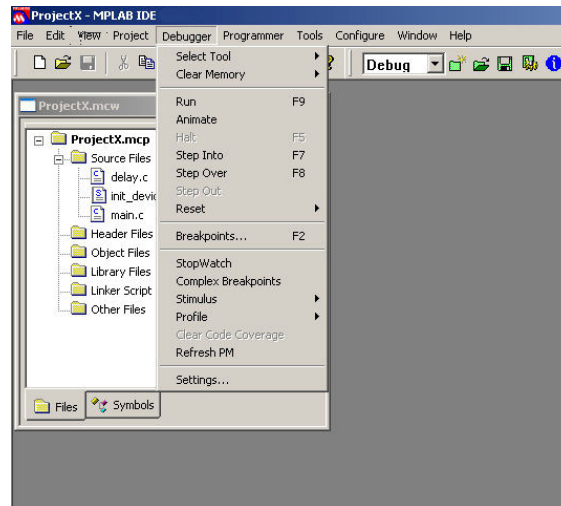When we exit the project wizard, the project window will look like this.

Wizard

If the project window is not shown, pull down the View menu and select the project window.
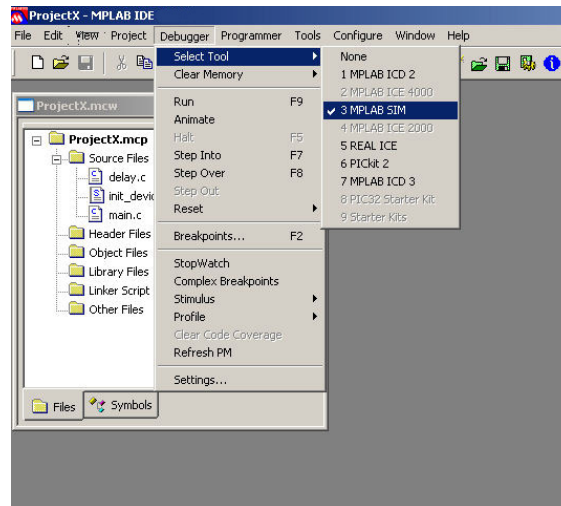
In order to test the code, we need a debug tool. MPLAB SIM is a debug tool. It is software simulator that can be used to test code on the PC.
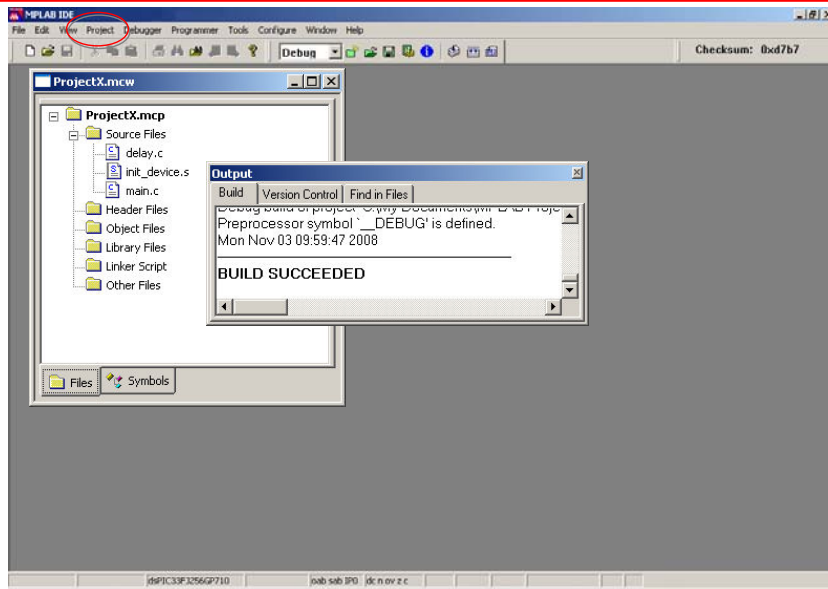
Select Simulator

An Introduction to MPLAB IDE

Select the Debugger Tool pull down menu…

Select Simulator

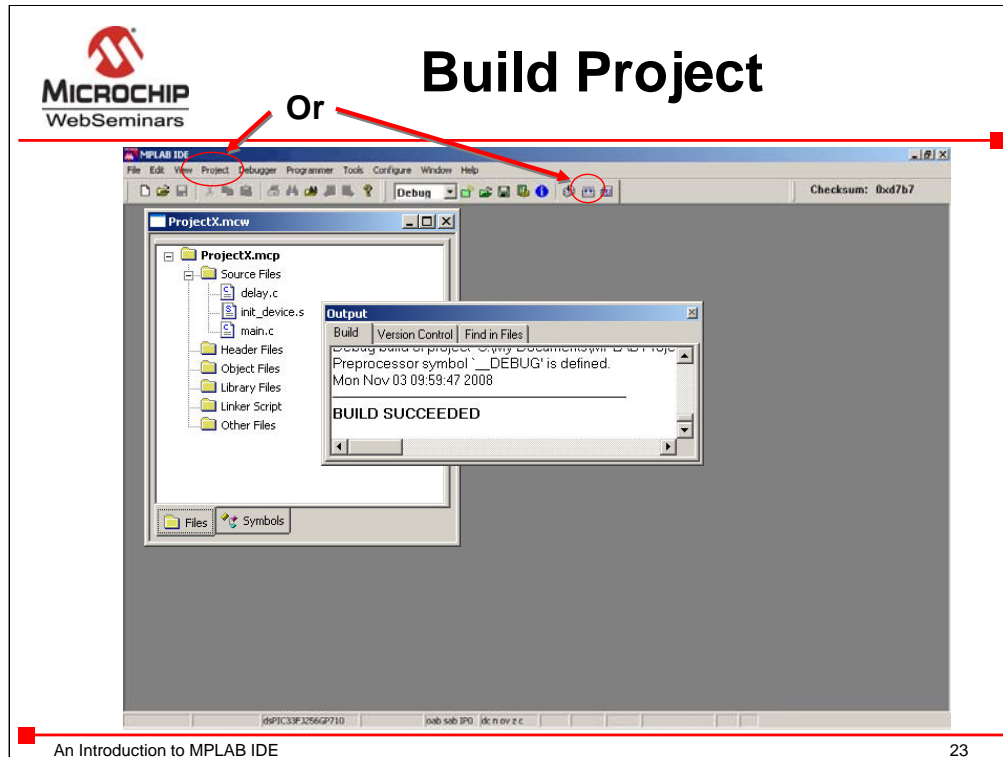An Introduction to MPLAB IDE   21

..scroll over to "Select Tool" and the submenu will pop up so we can select "MPLAB SIM."
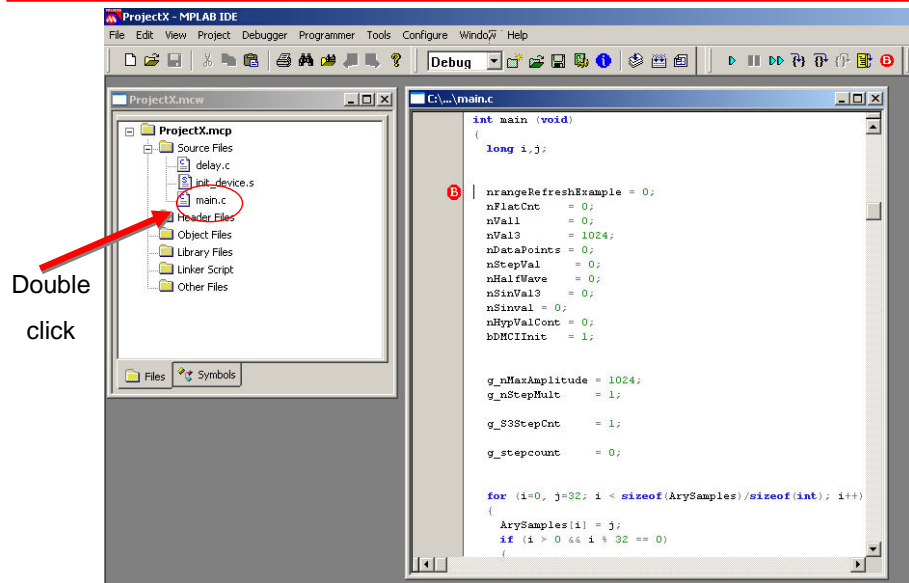
Build Project

Now we can build the project by selecting "Build Project" under the project menu pull down…

… or use the Build All icon from the tool bar. The output window will pop up and tell us if the project built correctly.
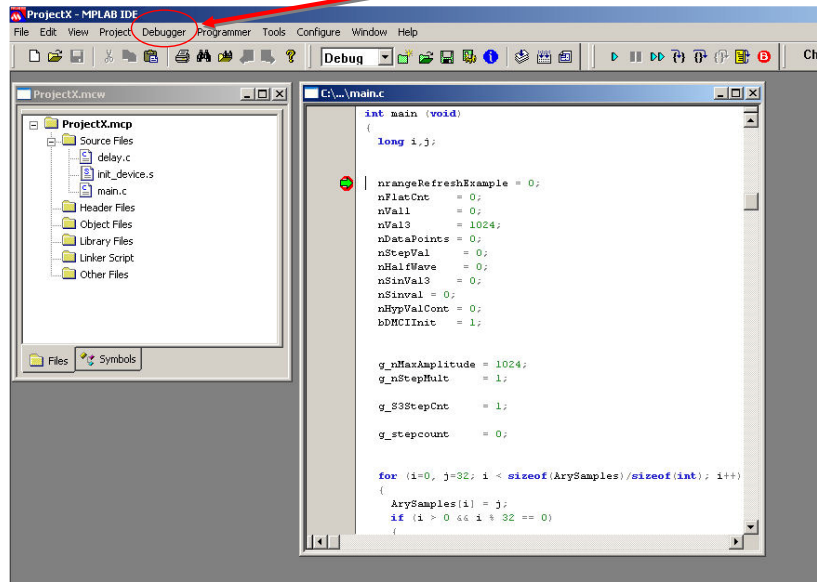
Set Initial Breakpoint

If we double click on the file named "main.c" in the project window, it will open in the programmer's text editor window. Here, we scrolled down to the function, "main."

Using the right mouse button, we set a breakpoint two lines into the function, as indicated by the red symbol with a "B."

Run To Breakpoint

MICROCHIP WebSeminars
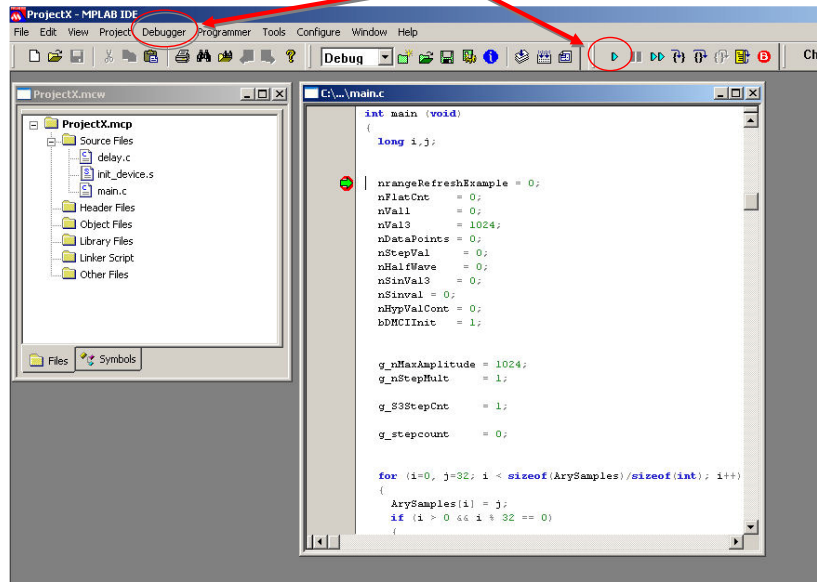
An Introduction to MPLAB IDE   25

Next we select "Run" from the Debugger menu…
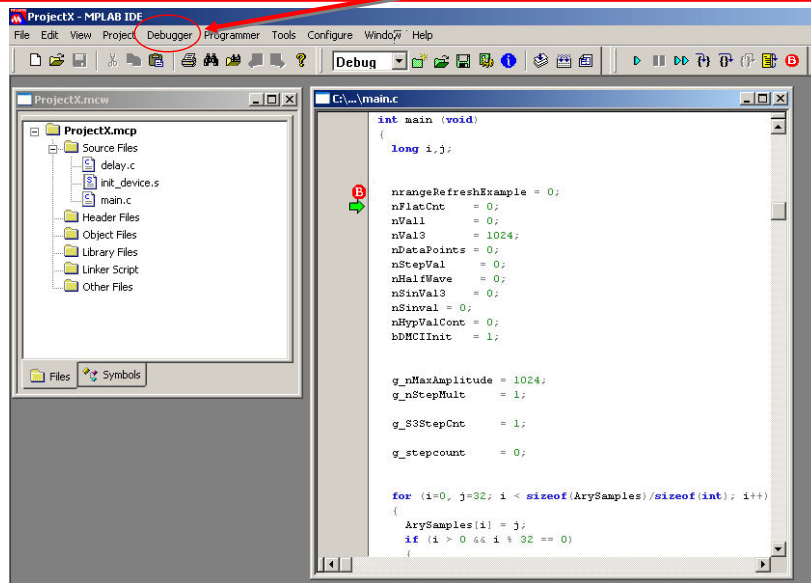
Run To Breakpoint

Or

An Introduction to MPLAB IDE 26

… or the green arrow on the tool bar.

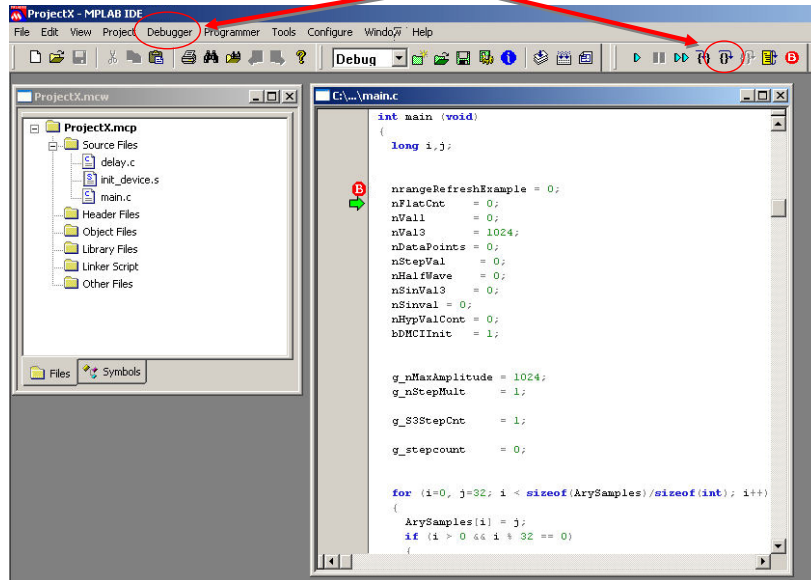# Set Initial Breakpoint

Now we can single step through the code from our breakpoint. Use the Debugger menu pulldown …

Set Initial Breakpoint
Or

…or the toolbar icon to step.

Set Initial Breakpoint

An Introduction to MPLAB IDE   29
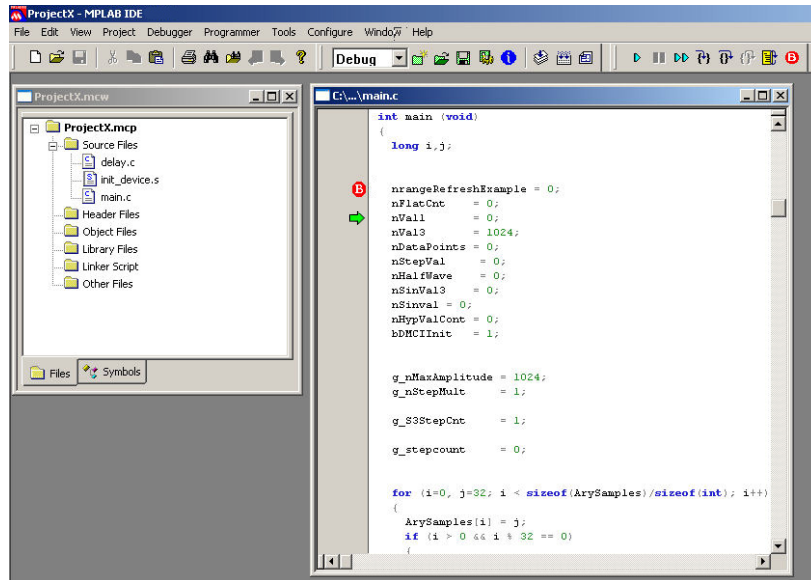
We can continue stepping through the code…

Set Initial Breakpoint

Page 30

Set Initial Breakpoint

Page 31

Set Initial Breakpoint

Page 32
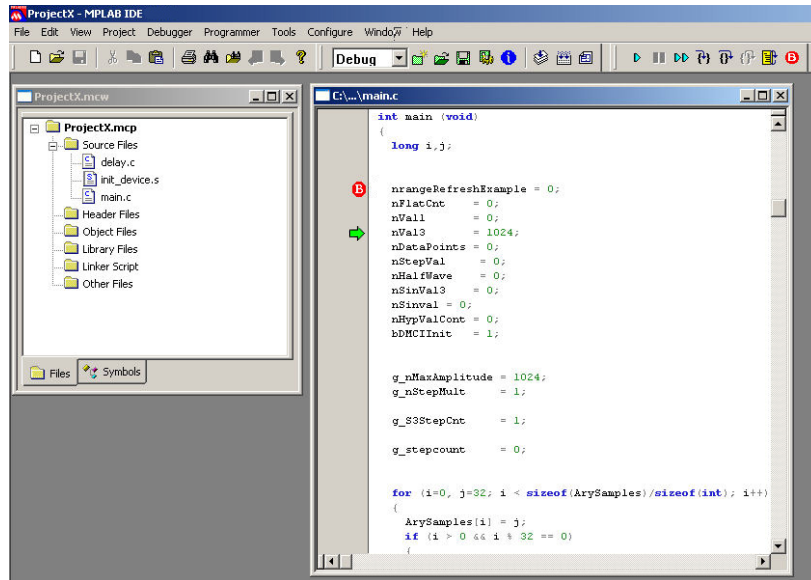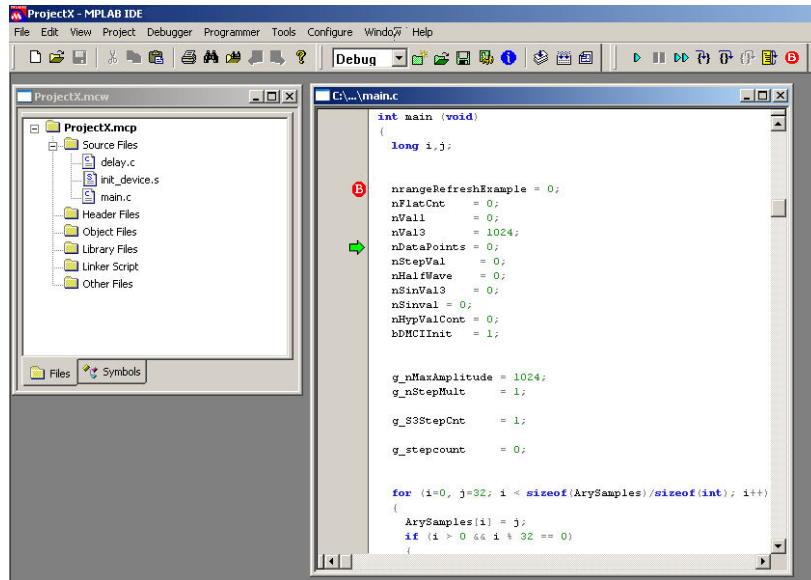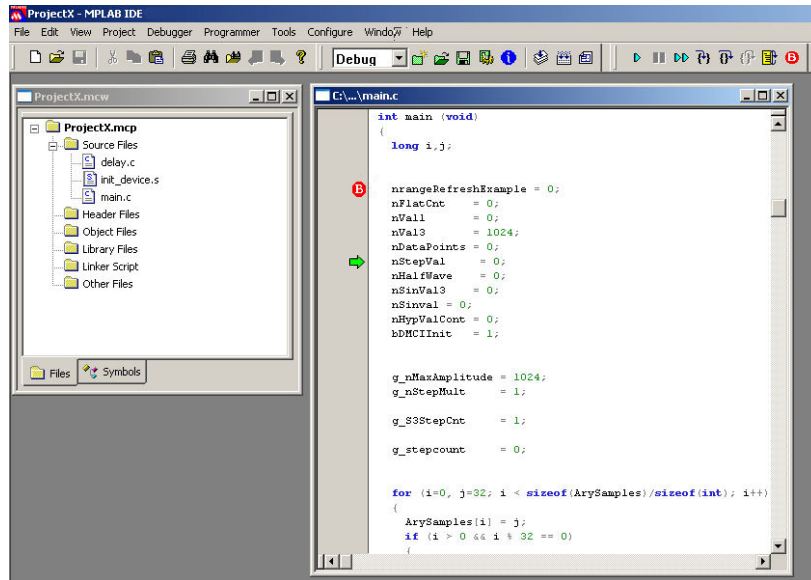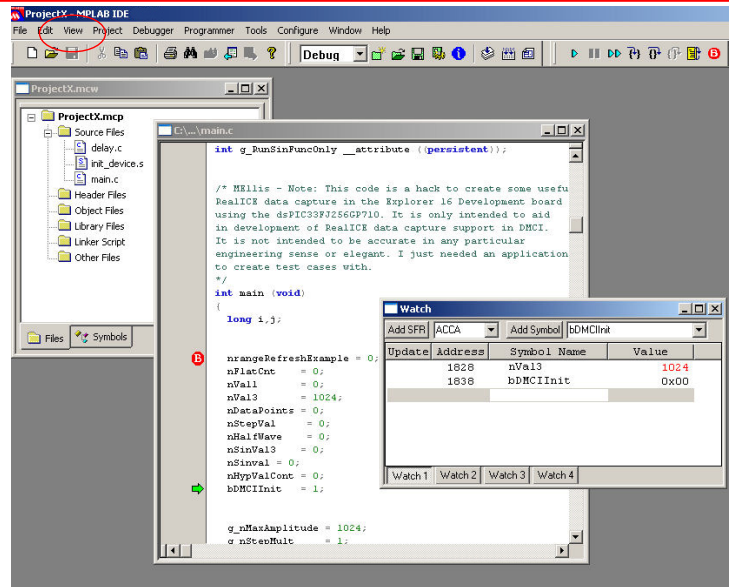
Watch Window

And that's how to get started. If you want to look at variables and their values, you can open the Watch window, which is on the View pulldown, and add variables to be viewed.

Page 33

Summary: The Next Step

- **Other main debugging features:**
  - Step-Into/Step-Over

Once you are familiar with these basic steps, you are ready to go on to explore the power of the MPLAB tool.

You might wish to explore the difference between Step-Into and Step-Over.

Summary: The Next Step

MICROCHIP
WebSeminars

- **Other main debugging features:**
  - Step-Into/Step-Over
  - Trace

As an alternative to breakpoints, you can choose to trace your code with a record of instruction execution

Summary: The Next Step

- **Other main debugging features:**
  - Step-Into/Step-Over
  - Trace
  - Change variable in Watch window

You can change variable values in the Watch window to see the effects on your application

Summary: The Next Step

MICROCHIP
WebSeminars

- ● **Other main debugging features:**
  - ● Step-Into/Step-Over
  - ● Trace
  - ● Change variable in Watch window
  - ● Complex breakpoints

And you can set complex breakpoints on particular variable values or sequences of events.

Summary: The Next Step

- **Other main debugging features:**
  - Step-Into/Step-Over
  - Trace
  - Change variable in Watch window
  - Complex breakpoints
  - Many, many other features:

•Tabbed editor	•Free C compilers	•HW Debuggers
•Color highlighted text	•Powerful plug-ins	•Programmers
•Version control support	•Mouse over variable	•Third party tools
•Macros	•Stimulus generator	•Data Monitor and Control Interface
•Custom hot keys	•Dockable windows	•Stopwatch
		•Code coverage

There is a lot more to MPLAB than we've covered here. Some features are listed here. All are covered in the on-line documentation that comes with MPLAB.

Once you have the basics mastered, it's easy to explore and find valuable tools to assist in your development efforts.

Download MPLAB and Try It!

WWW.MICROCHIP.COM/MPLAB

If you haven't already done it, now is the time to get started with MPLAB IDE.

For an overview of more of Microchip's development tools, you might wish to view the web seminar titled "An Introduction to Microchip's Development Tools."

To get MPLAB for free, simply go to our web site at microchip.com/mplab.

This is the end of our presentation.

Thank you for your time.