MPLAB 8 Appendix

MPLAB® 8 IDE Quick Reference Guide

Table of Contents

MPLAB 8 Appendix

This appendix is intended to be a supplement to lab manuals supplied with a Microchip Technical Training class. Although it may be useful on its own, it is not intended to provide complete instructions for using all aspects of the MPLAB X Integrated Development Environment. For more detailed information on the use of MPLAB X IDE, please consult one of the following Microchip Technical Training classes (for additional details, see **http://www.microchip.com/RTC**):
TLS0101—Getting Started with MPLAB X IDE
TLS0999—Transitioning to MPLAB X IDE for users of MPLAB IDE version 8
Or consult the Microchip Developer's Help Center at **http://microchip.wikidot.com/mplab:_start**

Trademarks

1. Managing Projects

Section 1.1
How to open a project

There are two ways to open a project:
Method 1:
Navigate to the project's directory and open the "Workspace" (*.mcw) file by double clicking on it. This will open the workspace associated with the project.

The Workspace file saves everything the Project file (*.mcp) saves and more. It will save the window position and sizes and watch window contents and settings.

(C:\MTT\classcode\Labx.mcw)

Method 2:

1 Start MPLAB® 8

2 Click on **File ▶ Open Workspace**

3 Navigate to the project's directory and open the "Workspace" (*.mcw) file.

```
MPLAB IDE v8.73
File  Edit  View  Project  Debugger  Programm
  New                          Ctrl+N
  Add New File to Project...
  Open...                      Ctrl+O
  Close                        Ctrl+E
  Save                         Ctrl+S
  Save As...
  Save All                     Ctrl+Shift+S
  Open Workspace...
  Save Workspace
  Save Workspace As...
  Close Workspace
  Import...
  Export...
  Print...                     Ctrl+P
  Recent Files               ▶
  Recent Workspaces          ▶
  Exit
```
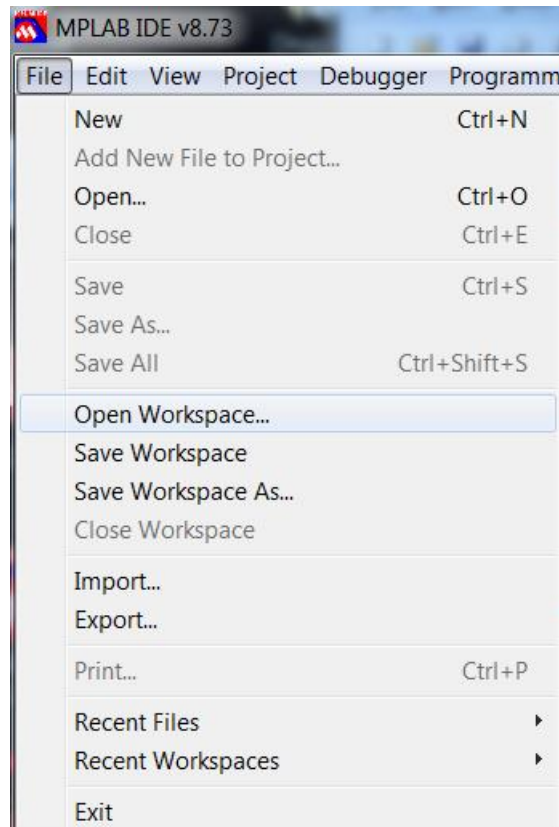
MPLAB 8 Appendix

Section 1.2
How to save a file or project

Method 1:

Click on **File ▶ Save Workspace**

Method 2:

Click on the **Save Workspace** icon
at the top of the MPLAB® IDE
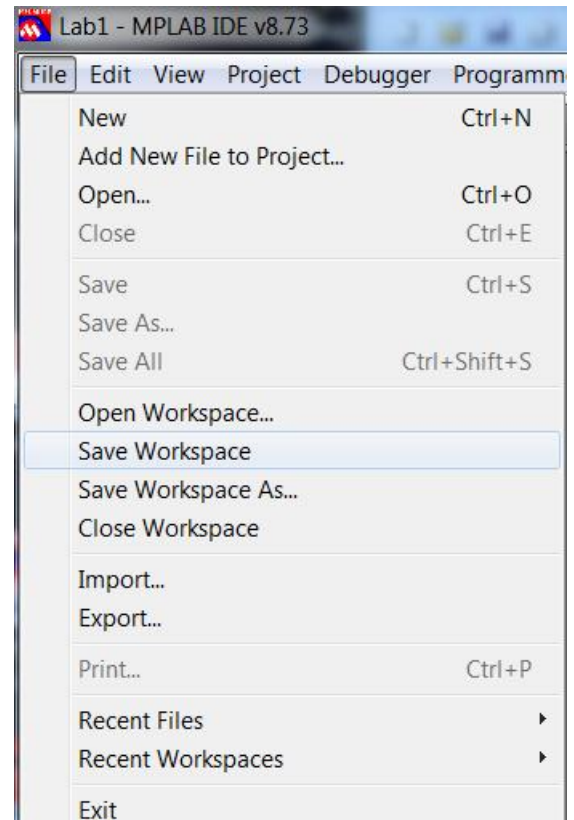
Section 1.3
How to close a project

Click on **File ▶ Close Workspace**
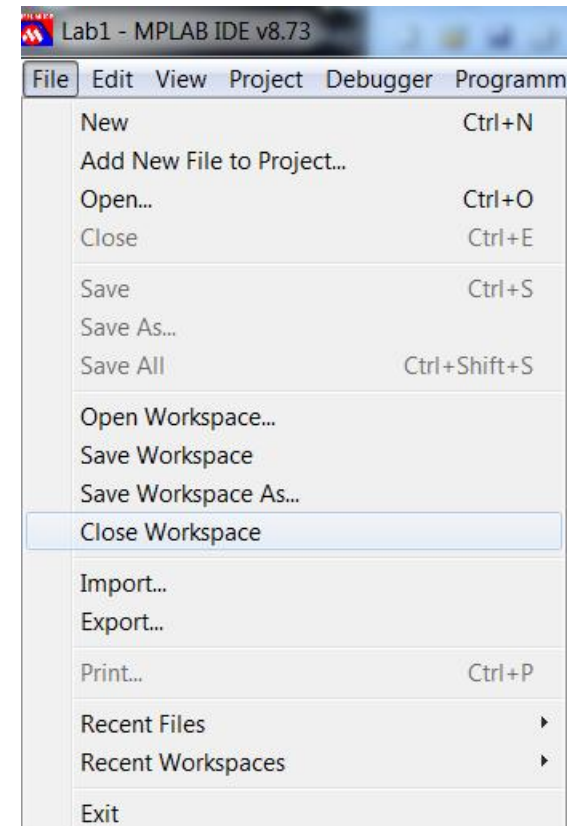
2. Building Projects

Section 2.1
How to build a project for production (stand-alone)

1 If you are building a project to run stand-alone (without a debugger) select the "Release" drop down option at the of the MPLAB® IDE.

Releas ▼

2 Method 1:
If you are building an assembly project or using a Microchip C compiler, click on the "Build All" icon at the top of the MPLAB® IDE

Build All

Method 2:
If you are using Microchip's HI-TECH C compiler, click on the "Build with Compiler" icon at the top of the MPLAB® IDE

Build with Compiler

Method 3:
If you have a large project (many large source files) and made a change in just one source file you can save time by "Making" the project instead of "Building". "Making" will recompile just the source files that have been changed since the last build. Click on one of these icons at the top of the MPLAB® IDE

Make

**Rebuild with Compiler**

Section 2.2
How to build a project for debugging

1 If you are building a project to use with a debugger (i.e. MPLAB® ICD3, MPLAB® SIM Simulator or Proteus Simulator) select the "Debug" drop down option at the of the MPLAB® IDE.

Debug ▼

2 Method 1:
If you are building an assembly project or using a Microchip C compiler, click on the "Build All" icon at the top of the MPLAB® IDE

Build All

Method 2:
If you are using Microchip's HI-TECH C compiler, click on the "Build with Compiler" icon at the top of the MPLAB® IDE

Build with Compiler

Method 3:
If you have a large project (many large source files) and made a change in just one source file you can save time by "Making" the project instead of "Building". "Making" will recompile just the source files that have been changed since the last build. Click on one of these icons at the top of the MPLAB® IDE

Make

Rebuild with Compiler

MPLAB 8 Appendix

3. Debugging Projects

Section 3.1
How to set or change the debugger

Click on **Debugger ▸ Select Tool** and choose your debugger from the list.
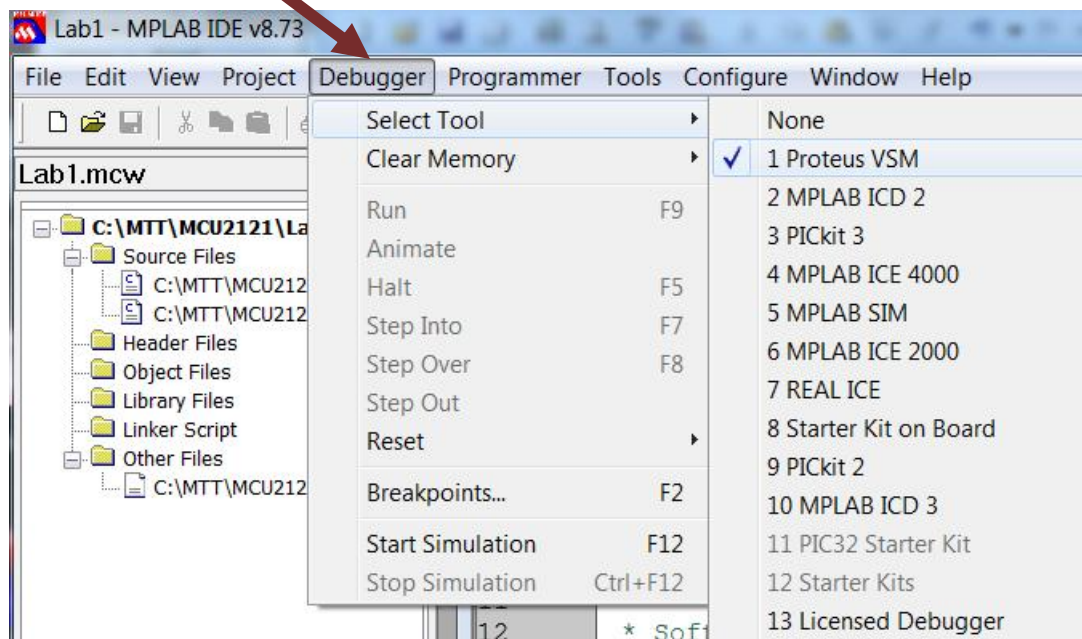
Section 3.2
How to control program execution when debugging

Debug Toolbar Buttons

Run (F9) the program

Pause (F5)

Step into or single step (F7) - Execute the next instruction

Step Over (F8) - Execute each line without stepping into functions (functions are executed without stepping through each line)

Step Out - Execute all instructions in the current function or sub routine then stop program execution

Reset (F6)

> ⓘ **Information**
>
> When you hover the mouse pointer over a toolbar button a tool tip will be displayed explaining the function of the button.

Section 3.3
How to set and clear breakpoints

Standard line breakpoints may be set or cleared by clicking on the line number in the glyph margin.

```
Lab1_Solution.c*
 91
 92    void main(void)
 93    {
 94        char counter = 0;        // Initialize count to zero
 95
 96        LATB = 0x00;             //###  Step 1:  Clear the PORTA
 97        LATA = 0x00;             //###           output latches
 98        TRISB = 0x00;            //###  Step 2:  Configure Port B
 99        TRISAbits.TRISA4 = 1;    //###  Step 3:  Configure Port A
100
101
102    while(1)
103        {
104            LATB = 0x00;                 //### Step 4: Write 0x00
105
106            while(SwitchPressed(&PORTA, 4)) // SwitchPressed deb
107                {
108                    LATB = counter++;   //### Step 5:   Incremen
109                                        //###             the resu
110                    Delay();
111                }
112        }
113    }    // End Main
114
```

More advanced breakpoint features may be accessed by opening the break-points window. From the main menu select **Debugger ▶ Breakpoints**.

Section 3.4
How to display and use Watches

1 To view the contents of a data memory location click on **View ▶ Watch** to open a watch window.

View	Project	Debugger	Programme

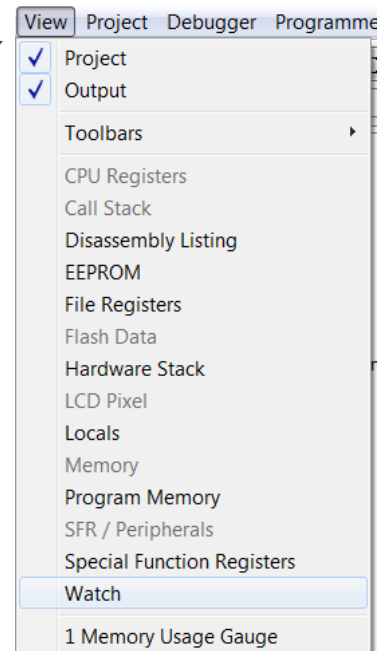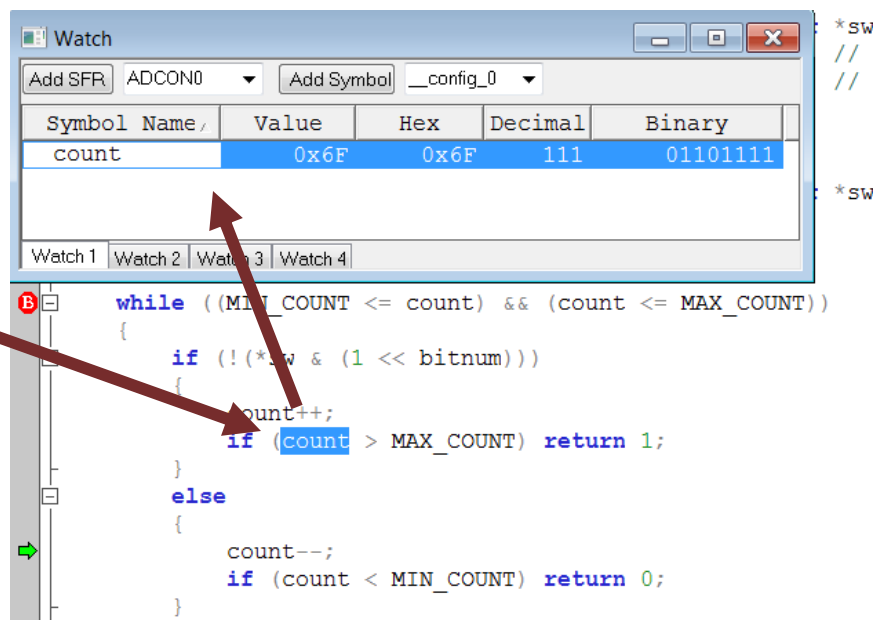|	
✓	Project
✓	Output
	Toolbars ▶
	CPU Registers
	Call Stack
	Disassembly Listing
	EEPROM
	File Registers
	Flash Data
	Hardware Stack
	LCD Pixel
	Locals
	Memory
	Program Memory
	SFR / Peripherals
	Special Function Registers
	Watch
	1 Memory Usage Gauge

Watch

Add SFR ADCON0 ▼ Add Symbol __config_0 ▼

Symbol Name	Value	Hex	Decimal	Binary
count	0x6F	0x6F	111	01101111

Watch 1 | Watch 2 | Watch 3 | Watch 4

2 To add a variable to the watch window highlight the variable you want to watch in the editor

3 Then drag-and-drop it into the watch window

```
B     while ((MIN_COUNT <= count) && (count <= MAX_COUNT))
      {
          if (!(*sw & (1 << bitnum)))
          {
              count++;
          if (count > MAX_COUNT) return 1;
          }
          else
          {
              count--;
              if (count < MIN_COUNT) return 0;
          }
```

This page intentionally blank