

Getting Started with PIC18F Programming in C18 using Laboratory Projects – Lesson 3

Lesson Three – Program Simulation

Objectives

- To work with MPLAB Simulator and observe the program output.

Preparation:

- Review the procedure in Lesson Two
- Review the on-line web seminar on Microchip's website "[Introduction to the MPLAB SIM Software Simulator](#)". (If the link gets updated and does not work, go to www.microchip.com and search for Seminars. Failing that, enter the title in Google and search it). This seminar although, not designed for the beginner still provides a good introduction to MPLAB SIM.

Procedure

In this lab a project will be created and compiled under C18. The program uses the source file developed in Lesson Two.

- Create a new folder on computer drive and name it **lesson 3**
- Open a MPLAB session
- Using the project wizard, create a new C18 project with PIC18F4520 as the device. Name the project '**lesson 3**'. (Remember not to add any extension to project name. MPLAB will automatically add '.mcp' extension.
- The 'lesson 3' project is created and is identified in the status bar.

Adding the Source File

- Go to the Project folder Lesson 2 and open the source file lab2.c.
- When the file is open, resave it in Lesson 3 folder, while renaming it as lab3.c. This ensures that the original project files of Lesson 2 are left intact.
- The project status changes indicating the new file location and the new file name **lab3.c**.
- If the file is not already open, double click the file lab3.c to open it in MPLAB editor. Change the date and file name in the source file (lab3.c) as shown below

```
/*  
    S Gupta  
    May118, 2010  
    Lab3.c  
    Simulating file under MPLAB SIM  
*/
```

Getting Started with PIC18F Programming in C18 using Laboratory Projects – Lesson 3

```
// Files to be included
#include <p18f4520.h>

void main (void)
{
    int x ;           // variable for x coordinate
    int m;           // slope variable
    int y ;          // y variable
    int c ;          // offset variable

    x = 10;          // assign value of 10 to x
    m = 2;           // set slope to 2
    y = 0            // clear the value of y coordinate
    c = 100;         // set offset to 100

    y = m * x + c;   // solve linear equation
                    // y should be equal to 120

    while(1);
}
```

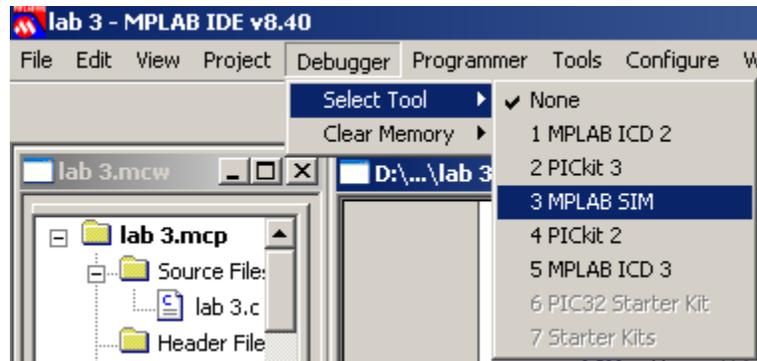
Click on the Build All icon to compile the project. If typed correctly, there will be no error reported by the build process, otherwise fix errors (in typing).

MPLAB SIM

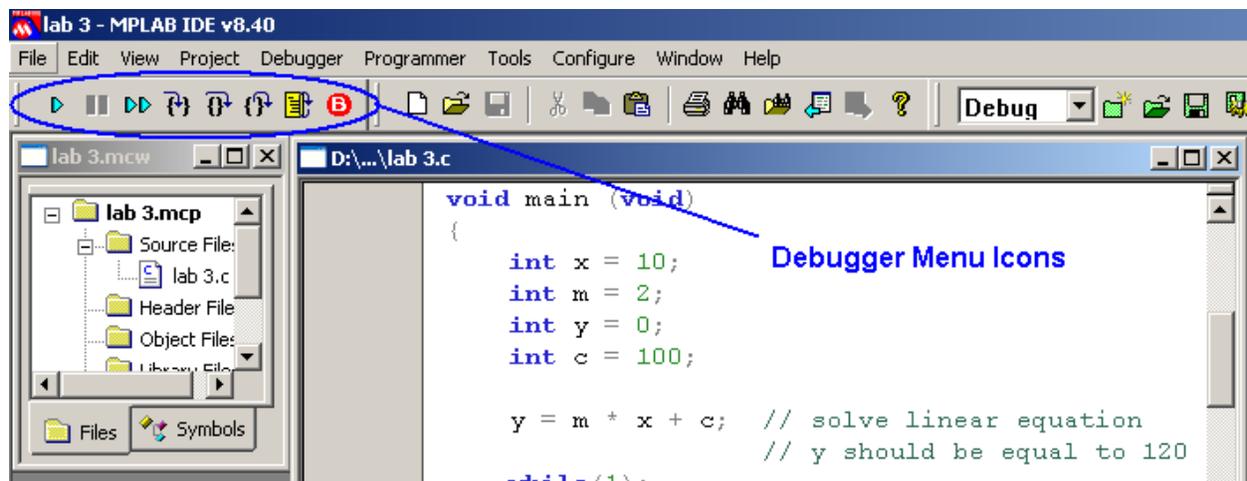
It is going to be assumed that you have browsed through the web seminar on MPLAB SIM (as indicated above). Take this opportunity to view the icons associated with MPLAB SIM on MPLAB IDE window.

- Using Debugger menu select MPLAB SIM (**Debugger**→**Select Tool**→**MPLAB SIM**).

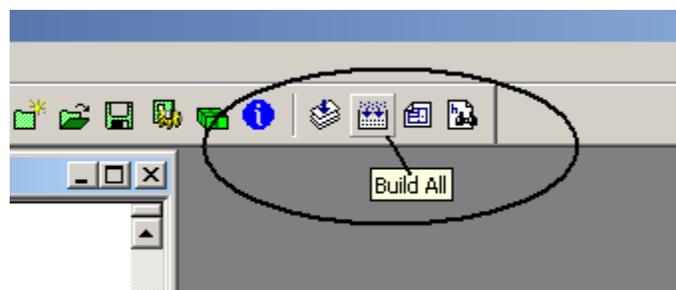
Getting Started with PIC18F Programming in C18 using Laboratory Projects – Lesson 3



- The debugger menu appears in MPLAB IDE



- Hover the mouse (slowly) over each of the icons and become familiar with the name and function of each.
- Using the Build All icon, build the project

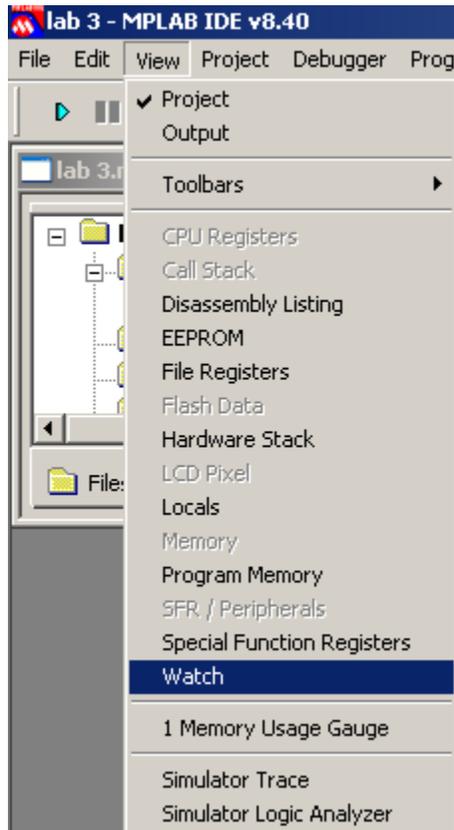


Getting Started with PIC18F Programming in C18 using Laboratory Projects – Lesson 3

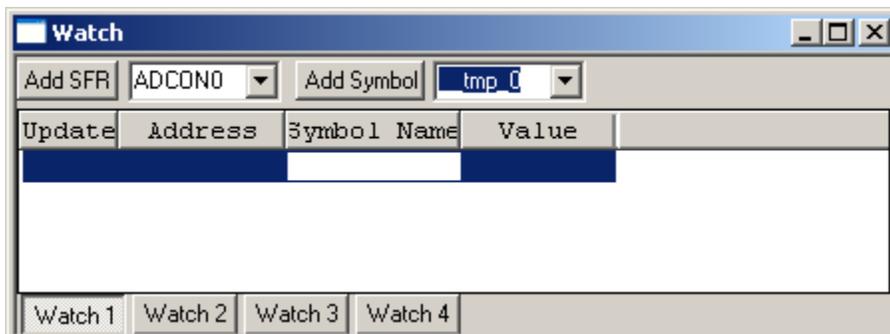
Getting Started with PIC18F Programming in C18 using Laboratory Projects – Lesson 3

Setting Up a Watch Window

Using the view menu select the Watch option (**View**→**Watch**)



A watch panel appears.



The Watch panel allows the viewing of al:

Getting Started with PIC18F Programming in C18 using Laboratory Projects – Lesson 3

- All file registers of the selected CPU (PIC18F4520 in this case)
- All variables declared in the program.
- Additional Watch windows (e.g. Watch 1, Watch 2 etc.) can be set up.

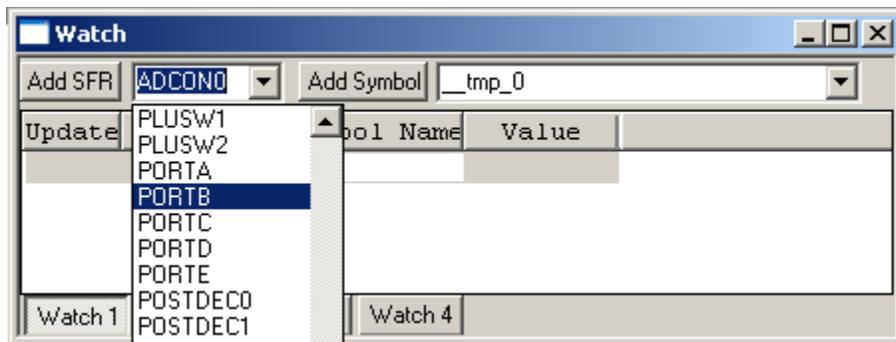
There are two main sections to a watch panel:

Add SFR – It allows the viewing of file registers

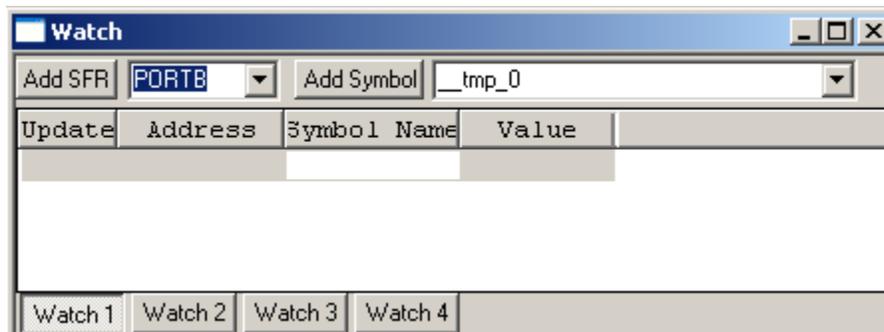
Add Symbol – It allows viewing of variables used in the program

Viewing File Registers

The file register of interest can be selected from the drop down list of Add SFR section and then clicking the Add SFR button. The following shows the selection of PORTB from the drop down list.

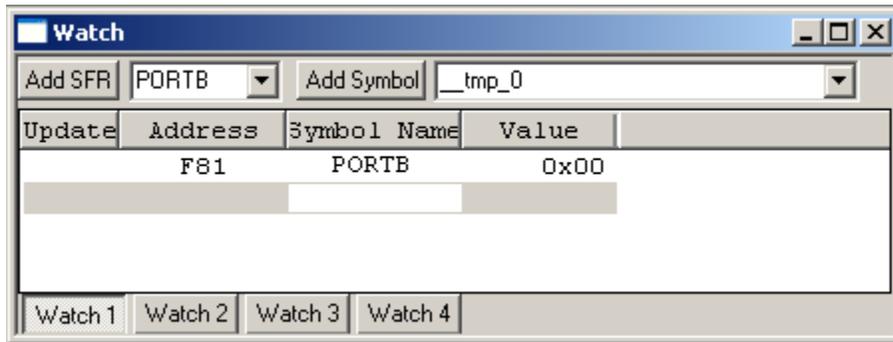


After PORTB has been selected (highlighted), it appears in the Add SFR window. The following shows PORTB in the ADD SFR window.



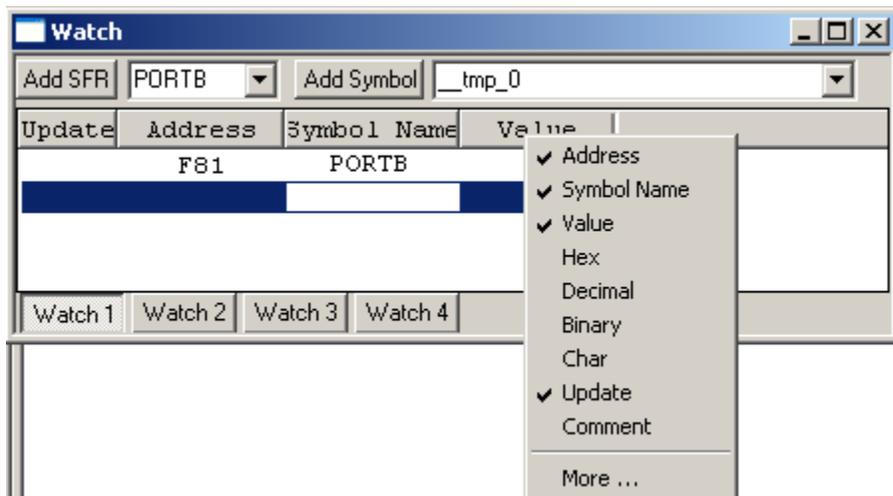
Getting Started with PIC18F Programming in C18 using Laboratory Projects – Lesson 3

PORTB can now be added to the watch window by clicking the Add SFR button.



Note: To delete a register select the register in question and press delete key on the keyboard

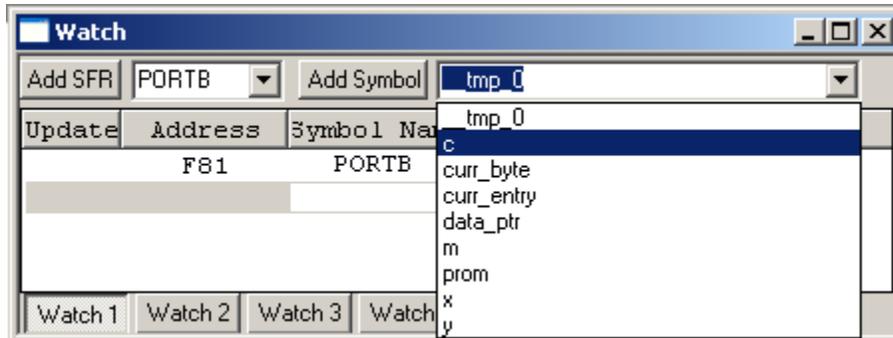
The Watch panel shows the PORTB address (0xF81), its name (PORTB) and its' current value (0x00). The value is displayed by default in Hex, but can be additionally displayed in various other numeric formats. Binary format is quite handy, when viewing registers.



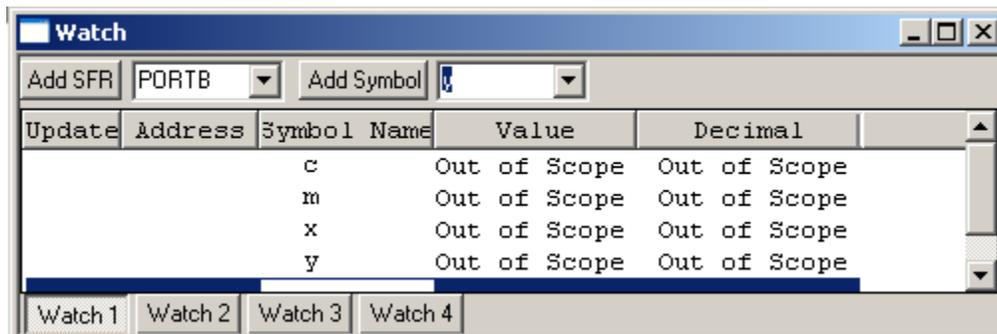
Viewing Variables

Any symbol or variable declared in the program can similarly be viewed through Add Symbol drop down list. On the drop down list every single variable declared in the program is available. The current program uses 4 variables: c, m, x, and y: these are shown in the list in an alphabetical order, along with other variables used by C18 compiler (C Startup file). The following shows the variable c selected. It can then be added by clicking the Add Symbol button.

Getting Started with PIC18F Programming in C18 using Laboratory Projects – Lesson 3



Using the above explanation, add symbols c, m, x, and y to the Watch window. Additionally, add viewing in Hex and decimal format, as shown below.



Out of Scope

The values of these variables are not currently displayed and an 'Out of Scope' statement is placed in the value columns. This simply indicates one of two things:

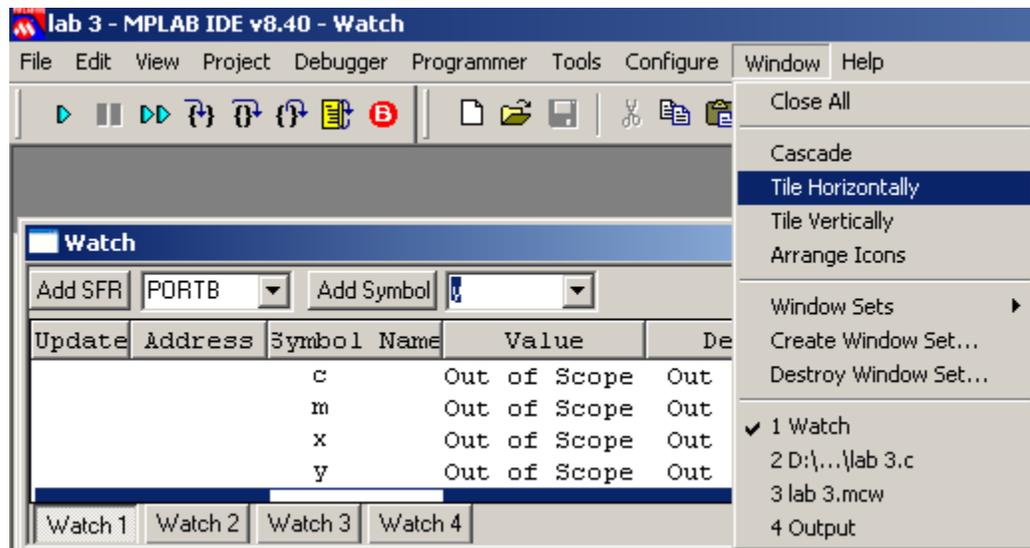
- The program is not yet executing, or
- If the program is executing, at the current step (in the program), the value of variable is not known (more on it later).

Getting Started with PIC18F Programming in C18 using Laboratory Projects – Lesson 3

Tiling of Panels

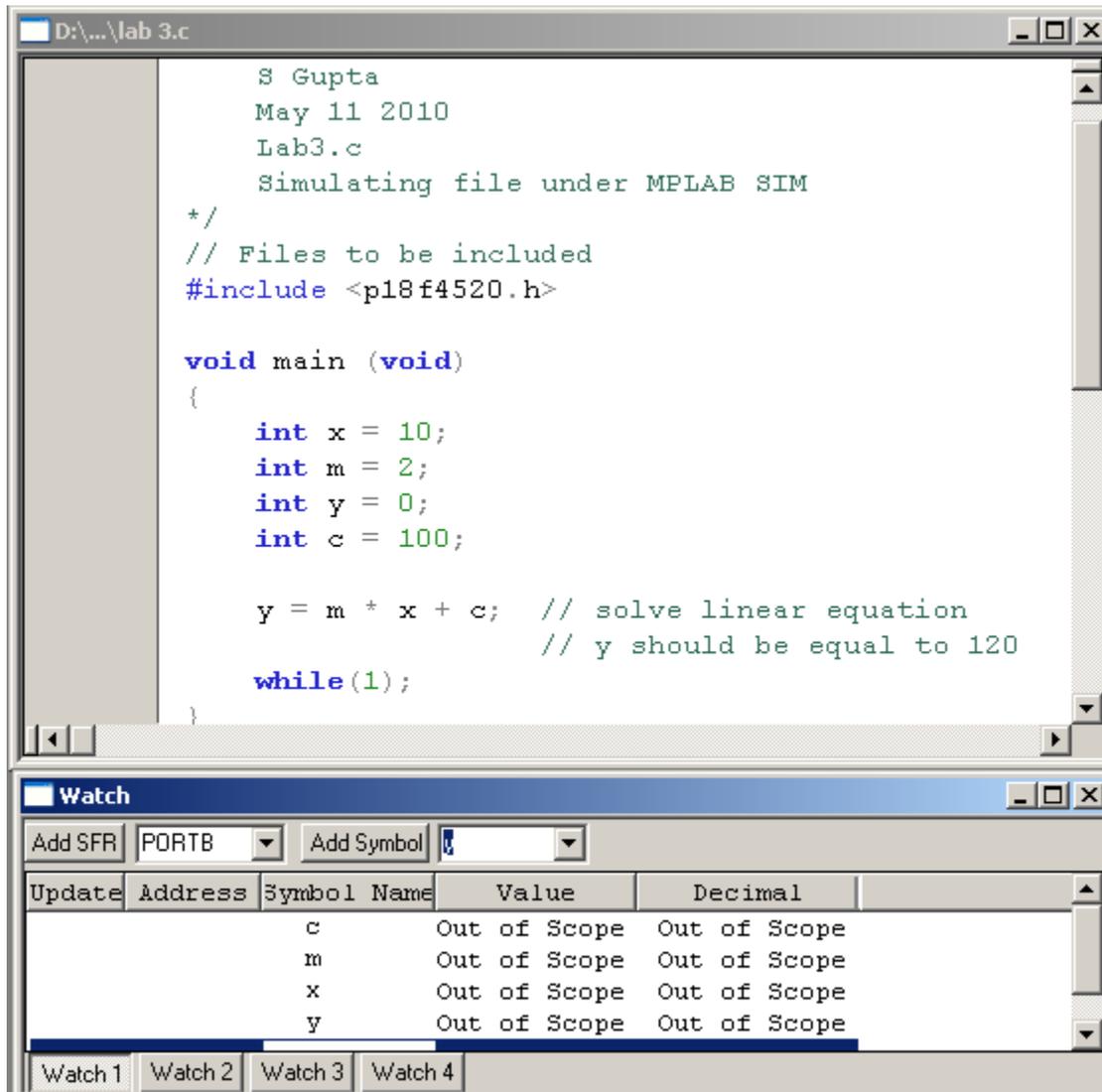
The various panels can now be tiled (horizontally or vertically) so that they can be viewed simultaneously as the program executes. It is achieved through Window menu. Monitor size sometimes will dictate the suitable tiling approach. Ultimately, it is a matter of personal preference.

Once tiled, the panels can be further adjusted and unimportant panels such as Project panel can often be closed to save valuable space on the screen.



Getting Started with PIC18F Programming in C18 using Laboratory Projects – Lesson 3

The following shows horizontally tiled panels, where the panel sizes have been adjusted and other panels removed (closed).



Program Execution and Viewing of Program variables

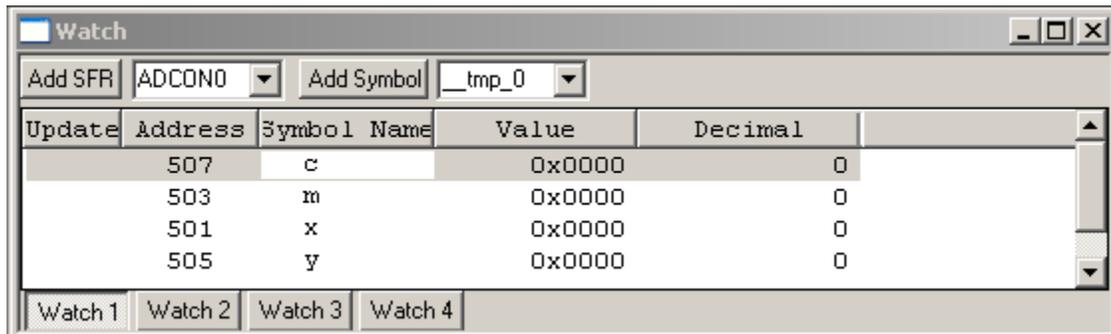
Place the cursor in the source window at the line 'int x = 10;' and click the right mouse button.

From the displayed menu, select 'Run to Cursor'.

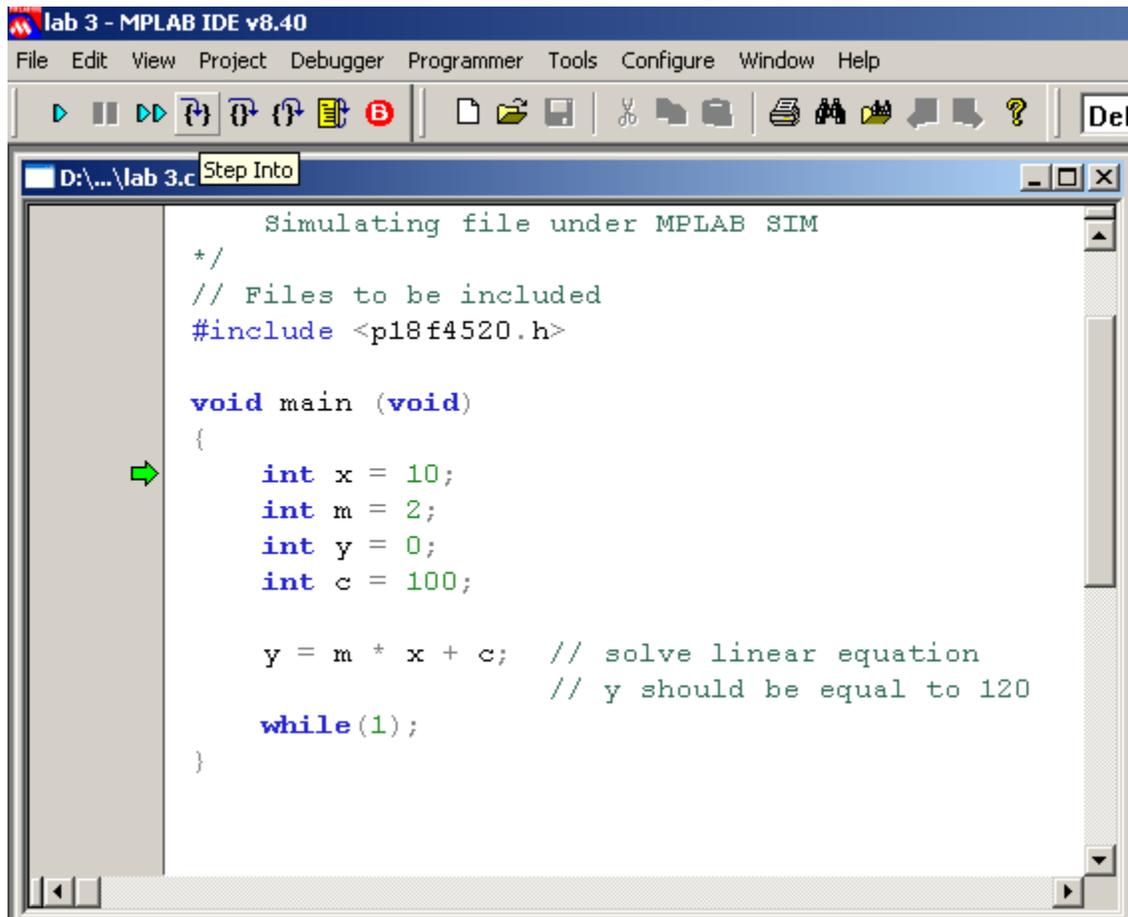
The program execution reaches the start of this line.

Getting Started with PIC18F Programming in C18 using Laboratory Projects – Lesson 3

Check the values of all 4 variables in the Watch window. These all have been initialized to 0 (by the CStartup file).

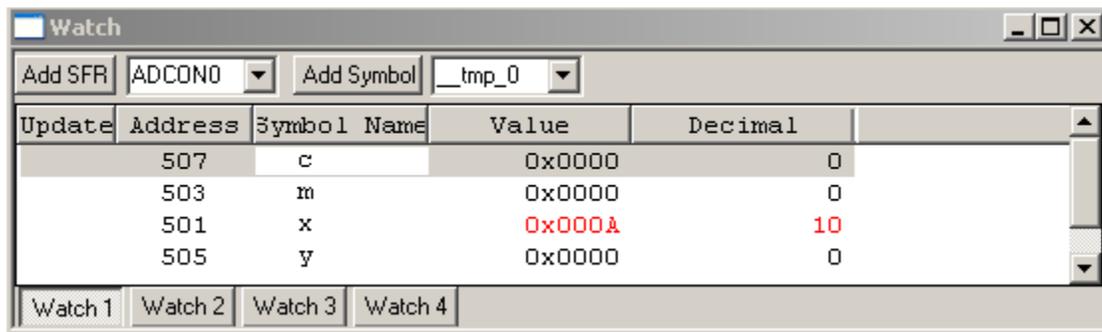


Select the **Step Into** Icon from the Debugger Icon list.



Getting Started with PIC18F Programming in C18 using Laboratory Projects – Lesson 3

The program will execute the current statement, the cursor moves on to the next step and value of variable x is initialized to 10. This can be verified through the Watch window. Since the value of x is changed from 0 to 10, it is highlighted in Red.

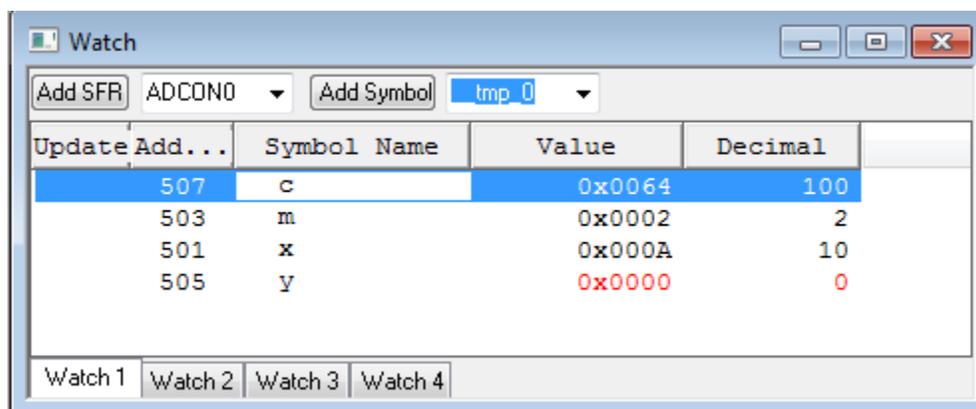


The screenshot shows the Watch window with the following data:

Update	Address	Symbol Name	Value	Decimal
	507	c	0x0000	0
	503	m	0x0000	0
	501	x	0x000A	10
	505	y	0x0000	0

Buttons: Watch 1, Watch 2, Watch 3, Watch 4

Similarly, continue executing the statements one by one till the cursor is positioned to line containing the linear equation. The equation has not yet been executed but all the variables have been initialized and it can be verified through the Watch window.



The screenshot shows the Watch window with the following data:

Update	Add...	Symbol Name	Value	Decimal
	507	c	0x0064	100
	503	m	0x0002	2
	501	x	0x000A	10
	505	y	0x0000	0

Buttons: Watch 1, Watch 2, Watch 3, Watch 4

The statement containing the linear equation will now be executed. Place a break point on the next line (while (1);). The program can now be executed till the end.

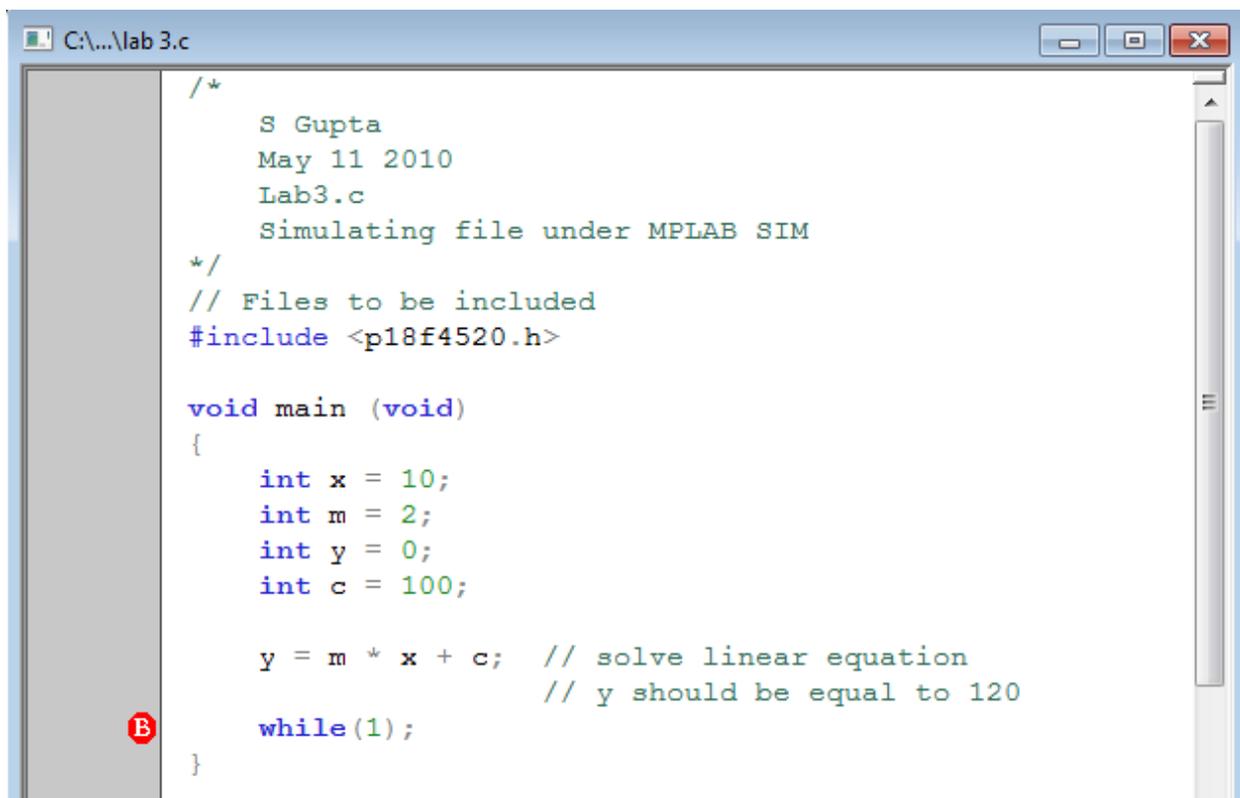
Getting Started with PIC18F Programming in C18 using Laboratory Projects – Lesson 3

Placing a breakpoint

Breakpoints can be placed at different points in the program. The execution stops at the breakpoint that is encountered. The value of various variables can be examined, and also changed, if so desired. The program execution can then be restarted from this point onward. There are some considerations regarding breakpoints.

- A maximum of 3 breakpoints can be placed
- A breakpoint cannot be placed on a non-executable statement

To place a break point, place the cursor to the desired line and double click the left mouse button. A break point is placed at that line and it is indicated by a symbol (similar to a Stop sign) in red containing 'B'.



```
C:\...\lab 3.c

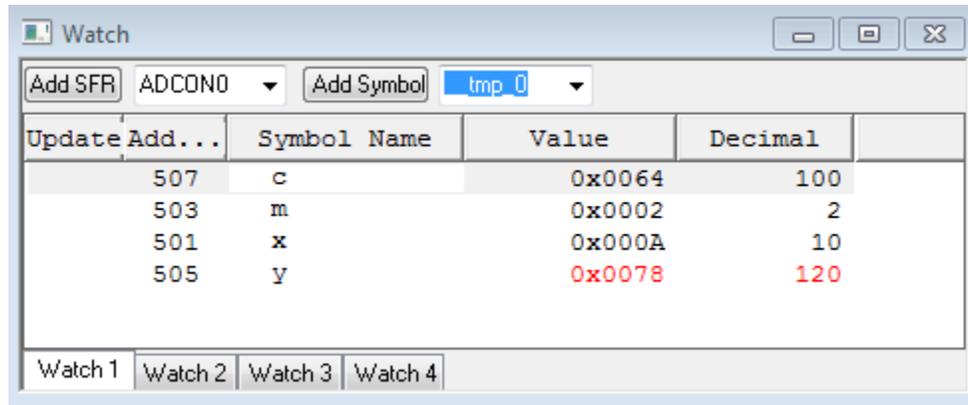
/*
   S Gupta
   May 11 2010
   Lab3.c
   Simulating file under MPLAB SIM
*/
// Files to be included
#include <p18f4520.h>

void main (void)
{
    int x = 10;
    int m = 2;
    int y = 0;
    int c = 100;

    y = m * x + c; // solve linear equation
                  // y should be equal to 120
    while(1);
}
```

Getting Started with PIC18F Programming in C18 using Laboratory Projects – Lesson 3

With cursor currently positioned at 'int c = 100;' statement, press the Run icon or (F9). The execution will complete the next statement and stop at the breakpoint 'while(1);' statement. The watch window reveals that the linear equation has been solved and the value of variable y stands at 120.



The screenshot shows the Watch window in MPLAB IDE. The window title is "Watch". At the top, there are buttons for "Add SFR" and "Add Symbol", and dropdown menus for "ADCON0" and "tmp_0". Below this is a table with columns: "Update", "Add...", "Symbol Name", "Value", and "Decimal". The table contains four rows of data:

Update	Add...	Symbol Name	Value	Decimal
507		c	0x0064	100
503		m	0x0002	2
501		x	0x000A	10
505		y	0x0078	120

At the bottom of the window, there are four tabs labeled "Watch 1", "Watch 2", "Watch 3", and "Watch 4".

This completes the lesson. Close the project and terminate MPLAB session.