

Getting Started with PIC18F Programming in C18 using Laboratory Projects

Lesson 2 - Creating a C18 Source File

Objectives

- To expand on the C18 project created in First Lesson.
- To create a C18 source file and understand the significance of some of the source file statements.

Preparation:

- Review the procedure in First Lesson on creating a MPLAB C18 project.
- Review the datasheet for PORTD (Section:10 input/output ports)

Required Resources:

- Computer with MPLAB and C18
- PIC18F4520 datasheet

Procedure

In this lab a new project will be created that is compiled under C18. The program uses the source file developed in First Lesson and adds additional 'C18' statements

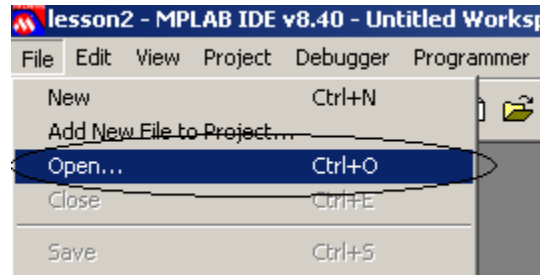
Note: As advised in previous lesson, it is highly recommended that the project be saved under a separate directory.

- Create a new folder on computer drive and name it **lesson2**
- Open a MPLAB session
- Using the project wizard (**Project→Project Wizard**), create a new C18 project with PIC18F4520 as the device. Verify that each of the 4 component (files) associated with C18 project are properly located. Name the project '**lesson2**'. (Remember not to add any extension to project name. MPLAB will automatically add '.mcp' extension.
- The 'lesson2' project is created and is identified in the status bar.

Source File

Getting Started with PIC18F Programming in C18 using Laboratory Projects

Currently, the **Source Files** folder in project panel (accessible through View→Project, if closed) is empty, as it contains no source file. The source file created in 'First Lesson' will now be added to this project. It can be done in a couple of different ways. Here a safer approach will be taken.



- Using **File→Open**, point to the location, where 'Lesson1' project has been saved and open the 'first.c' source file.

Resaving the Source File under new name

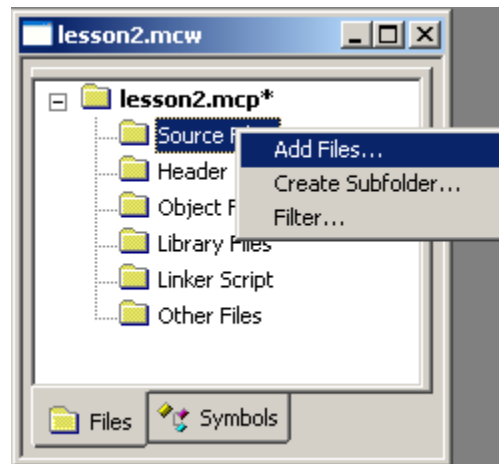
- This file (which belongs to an earlier project) needs to be saved under a new name and in the current project folder.
- With mouse pointer in the newly open file (first.c), using the **File→Save As** save the file in 'Lesson2' project folder and under a new name '**lab2.c**'.
- Project menu still shows that no source file has been attached to the project.
- The status changes indicating the new file location and the new file name **lab2.c**.

Getting Started with PIC18F Programming in C18 using Laboratory Projects

Attaching the Source File to the project

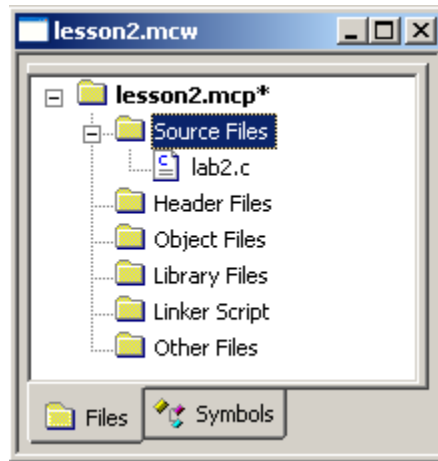
Although, the file is saved in the project directory, it is still not attached to the project.

- Highlight the **Source Files** on the project menu and using the right mouse click, select attach from the displayed menu select **Add Files**.

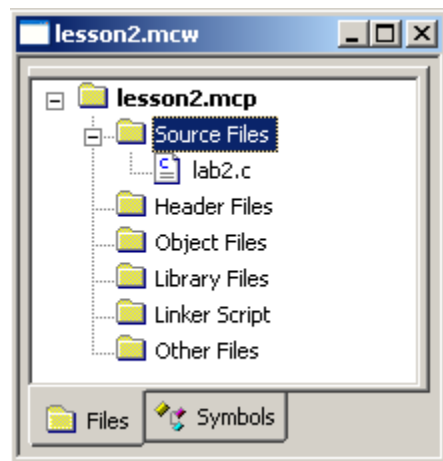


Getting Started with PIC18F Programming in C18 using Laboratory Projects

- From the displayed dialog box select the newly saved **lab2.c** file and click the Open button. This file gets added to the project as shown below.



Note: Sometimes an asterisk (*) appears next to file names as in the above (**lesson2.mcp2***). This simply indicates that the project and/or file have not been saved after a change. Save the file (**Project→Save Project** in this case) and the asterisk will disappear, as shown below.



Getting Started with PIC18F Programming in C18 using Laboratory Projects

Double click the lab2.c to open it in MPLAB editor. And modify the code as shown below.

```
/*
    S Gupta
    May 8, 2010
    Lab2.c
    Starting with C18 Programming
*/
// 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.

Getting Started with PIC18F Programming in C18 using Laboratory Projects

Explanation of the Source File Statements

*/**

It indicates the start of a comment field.

**/*

It indicates the end of the comment field. The comment field started by the */** continues till the matching **/* is not found. Anything in between these two fields is treated as comments and is completely ignored by the compiler.

This style of commenting is generally reserved for sections, where multiple lines of comments are needed. For example, at the start of a program multiple lines of comments are used to indicate the programmer's name, date, source file name, and a brief purpose of the program. In situations, where a single line of comment will suffice, the following commenting style is preferred.

//

It indicates that from this point onwards to the end of current lines only consists of comments (and not the code). This style lends well to adding short comments after a C statement.

Comments:

Enough can never be said about the significance of proper comments.

- **Use comments liberally**
- **Use comments to indicate your thought process**
- **The comments indicate to another user, or even the programmer (after a lapse), the reasons for the lines of code.**

#include <p18f4520.h>

- *#include* indicates that the filename that follows should be included in the source file.
- *<>* indicates that the file is located in the compiler directory. The name of the file P18F4520.h is included between these symbols.
- The p18f4520.h file contains the names and addresses of various registers and register bits in PIC18F4520 microcontroller.

void main (void)

Every C program has a main function. Functions will be discussed in more details later on. Here it is sufficient to understand that it represents the Main or Starting point of the program. In

Getting Started with PIC18F Programming in C18 using Laboratory Projects

general a function may be supplied with none (void) or some values and it may return a value or none (void).

void

The word 'main' is preceded by the word 'void' indicates that on termination, the main function does not return anything (so the qualifier void)

(void)

The word '(void)' following the word 'main' indicates that the main function is not supplied any arguments (values)

{ }

The body (definition) of every function is enclosed within curly brackets ({}).

The definition of main function is encompassed by curly brackets.

Variable declaration

The first 4 lines declare four variables

```
int x ;           // variable for x coordinate
int m;           // slope variable
int y ;          // y variable
int c ;
```

```
int x ;           // variable for x coordinate
```

- It declares x as an integer type variable
(An integer variable has a range of -32768 to +32767)
- The entire line consists of 3 parts:
 - int** - Indicates that the name that follows will be an integer variable
 - x - It is the name of the variable
 - ;- It is the termination character. Every single C statement must end with a semi-colon

After the C statement comment follows starting with '//'

Variable assignment

The next 4 lines assign values to these variables

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

Getting Started with PIC18F Programming in C18 using Laboratory Projects

```
c = 100;      // set offset to 100
```

```
x = 10 ;      // assign value of 10 to x
```

- It assigns a value of 10 to x
- The statement ends with a semi-colon, as every C statement should.

Calculation

The program calculates the y-axis coordinate, given the x-axis coordinate, the slope, and the offset value. It is done by the following statement. The multiplication of slope (m) and x coordinate is indicated by the asterisk (*) symbol and not a X as is done in algebra.

```
y = m * x + c;
```

while(1):

This statement holds the microcontroller at this point and prevents it from executing any additional statements

It is a good idea to place one such statement at the end of every main function (ONLY the main function and not elsewhere).

Getting Started with PIC18F Programming in C18 using Laboratory Projects

Combination of Variable Declaration and Initialization

The program can be shortened by, combining the variable declaration and assignment (initialization). The following shows a modified program.

```
/*
    S Gupta
    May 8, 2010
    Lab2b.c
    Starting with C18 Programming
*/
// Files to be included
#include <p18f4520.h>

void main (void)
{
    // declare and initialize variables
    int x = 10 ;    // variable for x coordinate
    int m = 2 ;    // slope variable initialize to 2
    int y = 0 ;    // y variable
    int c = 100 ;  // set offset variable to 100

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

    while(1);
}
// -----
```

This completes the lesson. Although, the program is functional, the result of calculation has not been revealed anywhere. This will be the topic of next lesson.

Close the project and terminate MPLAB session.