



*For Whirlpool Corporation
Application Note*

Metrowerks CodeWarrior IDE 5.5 Tutorial

Executive Summary

This application note will introduce new users to the Metrowerks CodeWarrior integrated development environment for the HCS12 microprocessor. It will begin by walking the user through creating a new project in CodeWarrior. The Metrowerks simulator will then be utilized to run a small C++ program and view actual changes to data, memory and the registers on the microprocessor. The program will load two variables containing the values 2 and 3. After the simulator runs, the sum will be stored in another variable and the result of 5 should be verifiable in the microprocessor's data and register values.

Introduction

Metrowerks CodeWarrior is an integrated development environment that provides visual simulations and eases the design of complex embedded applications. It is fully featured to compile drivers for the HCS12 microprocessor in assembly code, C and C++. It allows the users to quickly and easily create an initial project, source files, build, debug and run the project over its entire scope. The CodeWarrior tool-suite includes many pre-configured source files, library files and build targets, but it also allows for software developers to be fully flexible by creating custom projects from scratch. It also includes a real-time simulator that displays current information on data, processes, register values and memory.

This application note will use Metrowerks CodeWarrior IDE version 5.5.1272 and begin with creating a new project in CodeWarrior.

Creating a Project

Launch Metrowerks CodeWarrior. The IDE should pop up as shown below.

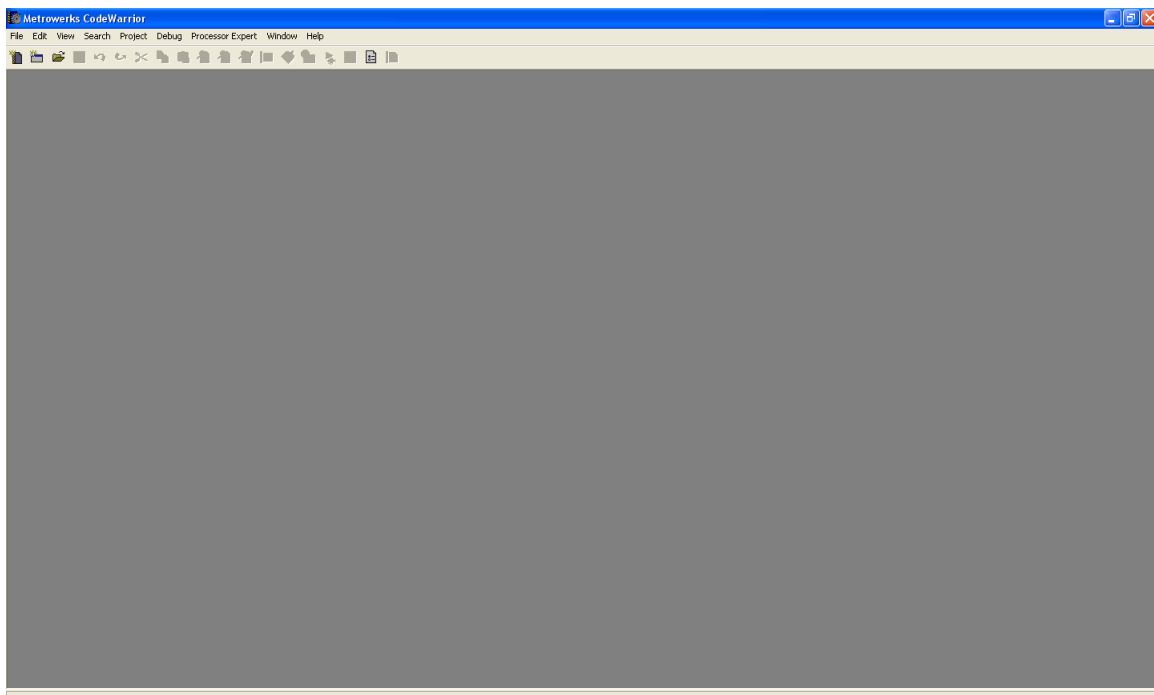


Figure 1 - Metrowerks CodeWarrior IDE

Click **File > New...** or click  on the toolbar.

A dialog box should appear that includes three project options. Select **HC(S)12 New Project Wizard**. This will allow us to choose all the necessary options in building the project. You can also build a project using an existing makefile. Refer to the CodeWarrior manual for this under 'Help'.

Click **Set...** This will display a new dialog box titled 'Create New Project...' Choose a location to save the project and create a name for the project in the 'File name:' box. Click **Save**.

Click **OK** and the dialog box should now display a large list of derivatives you can choose from. This option allows you to select the processor you wish to work with. Choose the appropriate derivative. For our purposes, choose MC68HC812A4 and click **Next**.

Now you may choose the set of languages you wish to work with. You may choose multiple languages, but in our case only check C++. Click **Next**.

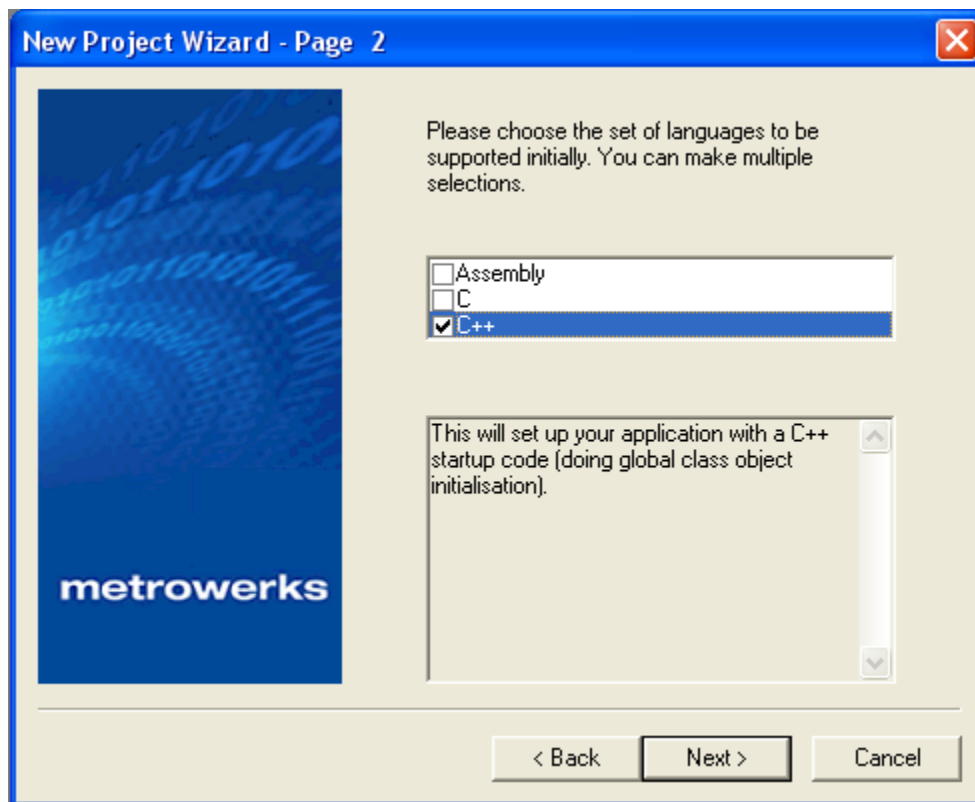


Figure 2 - New Project Wizard

The next dialog box (New Project Wizard - Page 3), allows you the option to set up for PC-lint. If you have the PC-lint software, choose yes. For our case, choose No and click **Next**.

If your processor supports floating point format, choose your format. For our case, choose None and click **Next**.

Choose your need of memory usage between small, banked and large. For our needs, select Small and click **Next**.

Choose the connections you need. For our case, choose Metrowerks Full Chip Simulator and click **Finish**.

The project setup should now be complete. The project will pop up with every item organized under tabs listed as: Files, Link Order or Targets.

Editing Source Code

Under the 'Files' tab, expand the 'Sources' folder by clicking on the plus sign to the left of it. Double-click on main.cpp.

Edit the main.cpp to look like this:

```
#include <hidef.h>    /* common defines and macros */
#include <mc68hc812a4.h> /* derivative information */

#pragma LINK_INFO DERIVATIVE "hc12a4"

void main(void) {

    unsigned x = 2, y = 3, z;

    EnableInterrupts;

    z = x + y;

    for(;;) {} /* wait forever */

}
```

This program takes the sum of the values of two variables (x = 2 and y = 3) and stores the result in another variable (z = 5).

Start debugging your program, by clicking on the **Debug** button  or by pressing **F5**.

Running a Simulation

The True-Time Simulator & Real-Time Debugger window should pop up:

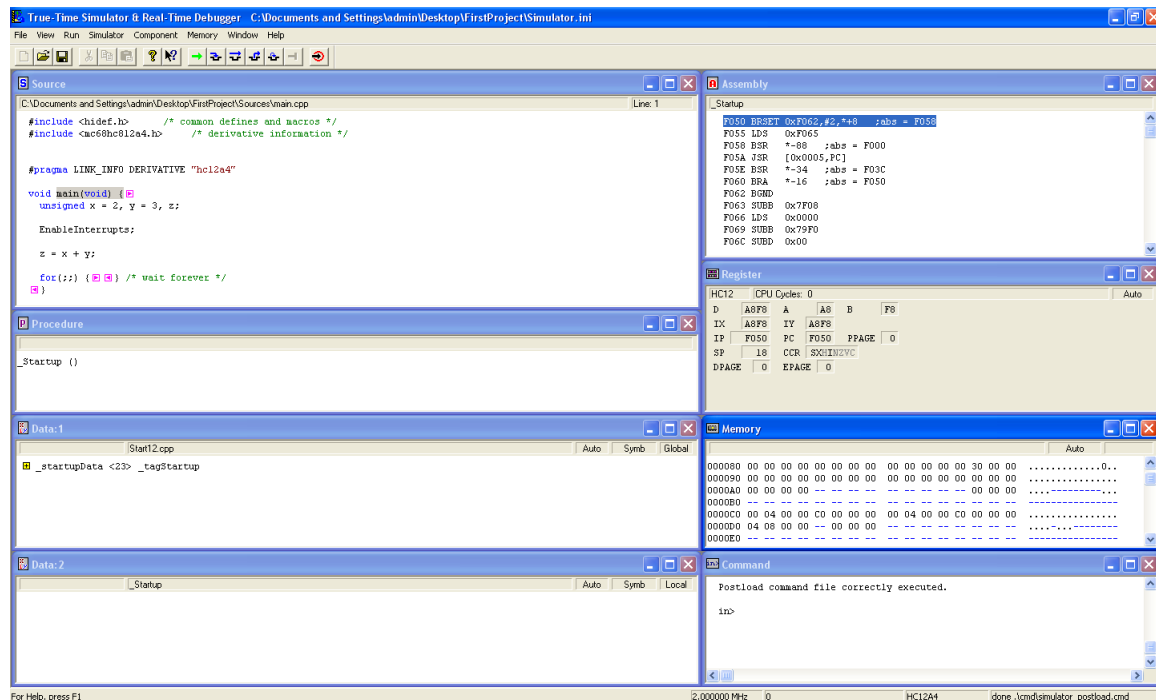


Figure 3 - True-Time Simulator & Real-Time Debugger

The available windows are as follows:

Source – This window displays your source code. You can edit and view your code in here before you run your program.

Procedure – This window displays the list of function calls that have been made so far.

Data 1 – This window displays the data and variable types of all variables that are used outside of your main source code.

Data 2 – This window displays the data and variable types of all variables that are used in your main source code.

Assembly – This window shows the translated assembly code derived from your main source code.

Register – This window shows all of the current values inside the register of your chosen processor.

Memory – This window shows the data stored in memory as a result of what has been run so far.

Command – This window shows the commands that have been executed by the processor and simulator.

After the simulator has opened and you have become comfortable with its environment, click on the **Start/Continue** button  or press **F5**. The simulator will continue to run until you press the **Halt** button  or press **F6**.

You should now see the results of your program.

The Data 2 window should show all values that were set for the variables in your program. If your simulation was successful, you should see values of 2, 3 and 5 for variables x, y and z, respectively.

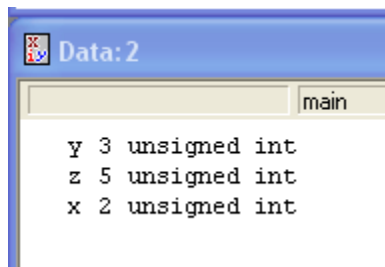


Figure 4 - Data 2 Window

The Register window should show the new values stored in the registers of your processor. If your simulation was successful, you should see values of 5 in the D register and 5 in the B register.

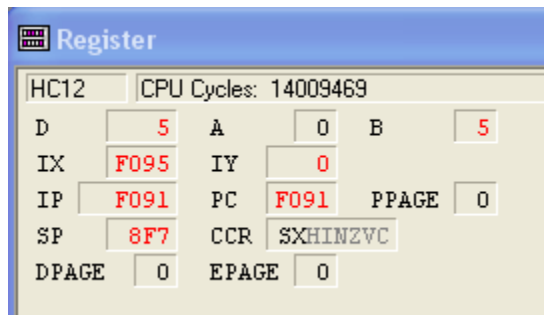


Figure 5 - Register Window

Conclusion

This procedure is complete with creating a project, adding source code and running a real-time simulation. It shows how Metrowerks CodeWarrior can be used to run a simple program on the HCS12 microprocessor and introduced a familiarity with the integrated development environment.

References

CodeWarrior development tools can be found at:

<http://www.freescale.com/webapp/sps/site/homepage.jsp?nodeId=012726>

Information about Metrowerks CodeWarrior found in this document plus additional information can be found in the manual under 'Help' in the CodeWarrior IDE.