



CodeWarrior™ Development Studio for Freescale™ S12(X) Microcontrollers Quick Start

SYSTEM REQUIREMENTS

Hardware	PC with 1 GHz Intel® Pentium®-compatible processor 512 MB of RAM (1 GB recommended) CD-ROM drive Depending on host-target connection: Parallel Port, 9-pin Serial Port, or USB Port
Operating System	Microsoft® Windows® 2000, Microsoft Windows XP, or Microsoft Windows Vista™ Operating Systems 32 bit (Home Premium Edition and Business Edition)
Disk Space	2 GB total 400MB on Windows system disk

This Quick Start explains how to install the CodeWarrior Development Studio for S12(X) V5.0 software and then create, build, and debug a project using the IDE.

NOTE In this document, numbered steps are for advanced users. Lettered steps are the expanded descriptions.

Section A: Installing Software

1. Install the CodeWarrior software.
 - a. Insert the **CodeWarrior Development Studio** installation CD into the CD-ROM drive — Auto Install begins.

NOTE If Auto Install does not start, run `Setup.exe`, located in the root directory of the CD.

-
- ... The CodeWarrior software may be part of a DVD included with your kit. In this case, click **Install CodeWarrior Development Studio for S12(X) V5.0**, follow the on-screen instructions, and skip to step "Check for updates".
-

- b. Click **Launch the installer** — the **Install** wizard appears.
 - c. Click **Next** — the **License Agreement** page appears.
 - d. Select the **I accept the terms in the license agreement** option button.
 - e. Continue clicking **Next** to step through wizard pages, accepting default settings — the **Ready to Install the Program** page appears.
 - f. Click **Install** — At the end of installation, a page appears announcing installation is complete.
 - g. Select the **Yes, check for program updates (Recommended) after setup complete** checkbox to check for updates.
2. Check for updates.
-

NOTE If the updater already has internet connection settings, you may proceed directly to sub-step f.

- a. Click **Settings** in the **CodeWarrior Updater** dialog box — the **CodeWarrior Updater Settings** dialog box appears.
 - b. Click **Settings** — the **Connections** page of **Internet Properties** dialog box appears.
 - c. Modify settings, as appropriate, to successfully connect to internet.
 - d. Click **OK** — the **Internet Properties** dialog box closes.
 - e. Select an item in the **Update Check Scheduling** list box and click **OK**; or click **Cancel**.
 - f. Click **Next**.
 - g. If necessary, enter the username and password.
 - h. If updates are available, follow the on-screen instructions to download the updates to your computer.
-

NOTE If no updates are found, the software application will display an appropriate message.

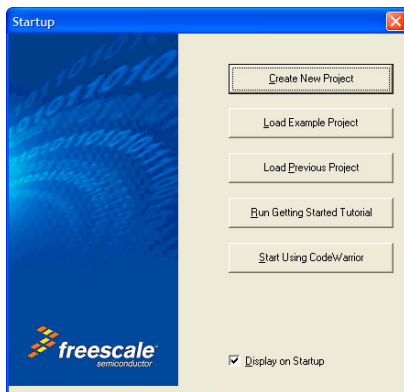
- i. Click **Finish** — the installation completes and the **CodeWarrior Updater** dialog box closes.

For licensing and activation of your CodeWarrior Development Studio for Freescale S12(X) Microcontrollers, refer to the *CodeWarrior Development Suite Quick Start*. Save the license file, `license.dat` to the installation root folder, the default is `C:\Program Files\Freescale\CodeWarrior for S12(X) V5.0`.

Section B: Creating and Building an S12(X) Project

1. Create a new project.
 - a. Select **Start > Programs > Freescale CodeWarrior > CodeWarrior Development Studio for S12(X) V5.0 > CodeWarrior IDE** — the IDE starts and the **Startup** dialog box appears.

Startup Dialog Box

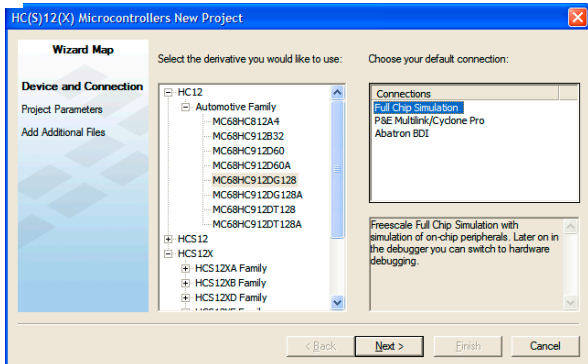


- b. Click **Create New Project** — the **Device and Connection** page appears.

NOTE This section of the quick start demonstrates using the New Project Wizard. We use an MC68HC912DG128 target as an example.

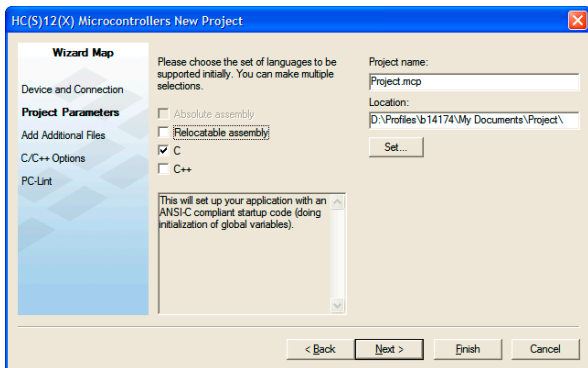
- c. Expand **HC12** and **Automotive Family** and select the **MC68HC912DG128** derivative.

Device and Connection Page



- d. Select **Full Chip Simulation** as your default connection.
- e. Click **Next** — the **Project Parameters** page appears.

Project Parameters Page



- f. In the **Project name** text box, type a project name of your choice.

NOTE The IDE automatically creates a folder with the same name in specified location. The IDE automatically adds .mcp extension when it creates project.

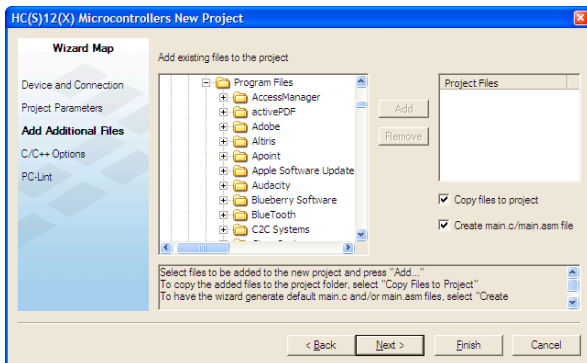
- g. In the **Location** text box enter location to store the project.
- h. Select the **C** checkbox as language to be supported by the project.

You can select **Finish** to accept defaults for remaining options.

- i. Click **Next** — the **Add Additional Files** page appears.

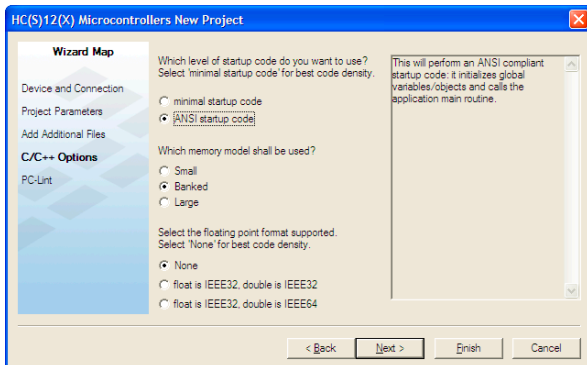
This page allows you to browse folders and add or remove files to or from the project.

Add Additional Files Page



- j. Click **Next** — the **C/C++ Options** page appears.

C/C++ Options Page



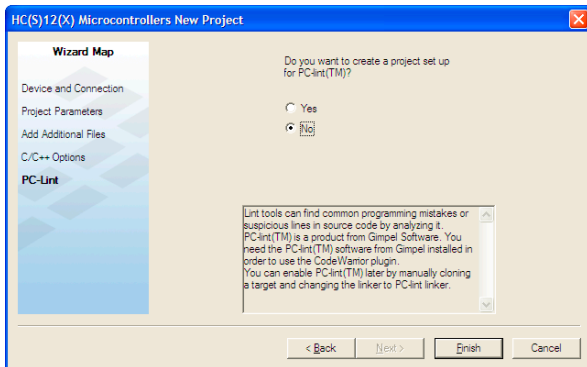
- k. Select the **ANSI startup code** option button as level of startup code to use.
- l. Select the **Small as memory model** option button as memory model to use.



Click the **None** option button for floating point format to support.

- n. Click **Next** — the **PC-Lint** page appears.

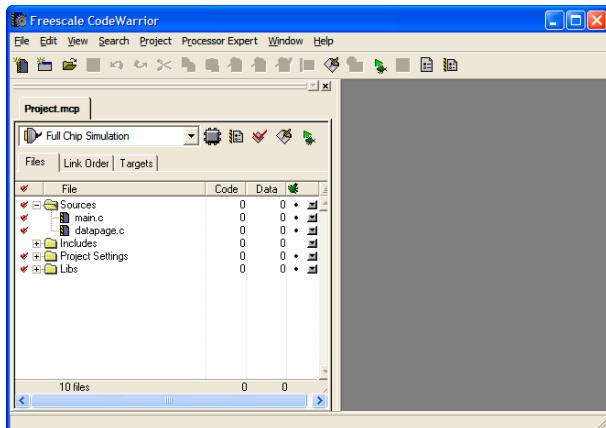
PC-Lint Page



- o. Select the **No** option button.
- p. Click **Finish** — the IDE creates a project according to your specifications; the Project window appears, docked at left side of main window.


NOTE To undock project window, double-click the double gray lines.
To re-dock window, right click in title tab and select **Docked**.

Project Window



connection.

For this example, we specified Full Chip Simulation (FCS).

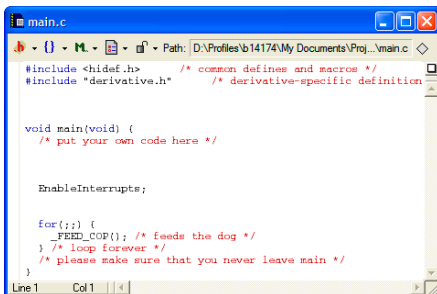
a. To change the derivative and connection, click the **Change MCU/Connection**  icon.

b. Select **Full Chip Simulation** from the drop-down list.

3. Edit the source code

a. Double click `main.c` in the Sources folder – the **Editor** window opens displaying contents of file.

main.c in Editor Window



```

main.c
D:\Profiles\b14174\My Documents\Proj...main.c
#include <hidef.h> /* common defines and macros */
#include "derivative.h" /* derivative-specific definition

void main(void) {
    /* put your own code here */

    EnableInterrupts;

    for(;;) {
        _FEED_COP(); /* feeds the dog */
    } /* loop forever */
    /* please make sure that you never leave main */
}
Line 1 Col 1

```

b. Make changes to the contents of `main.c` file, if desired.

c. If you make changes to the `main.c` file, from IDE main menu bar, select **File > Save** – IDE saves the changes.

4. Add files, if required.

a. In the project window, select a folder.

b. From the IDE main menu bar, select **Project**.

c. Select **Add Files** — the **Select files to add** dialog box appears.

d. Navigate to the directory that contains file you want to add.

e. Select filename of the file you want to add to the project.

f. Click **Open** — Project messages appear indicating access path has been added to target, if path is new to the project.

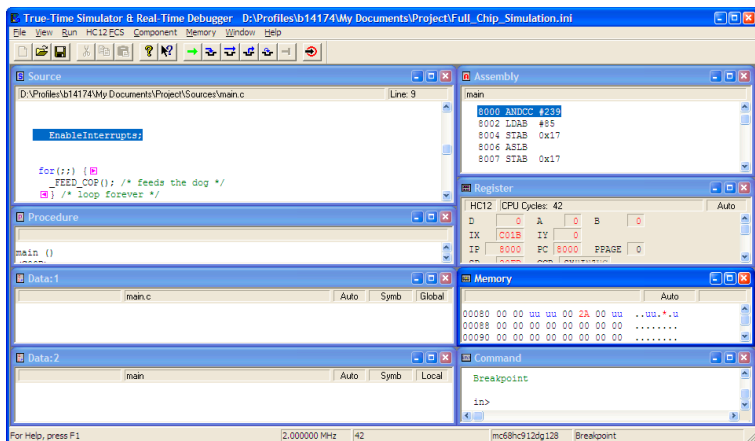
g. In the project window, filename of the added file appears under the selected folder.

- From the IDE main menu bar, select **Project**.
- Select **Make** — the IDE builds (assembles, compiles, and links) project; the **Error & Warnings** window opens showing any error messages and warning messages.

Section C: Debugging an S12(X) Project

- Start the debugger.
 - Click on the project window title bar to activate the project window.
 - From the main menu bar, select **Project > Debug** — the **True-Time Simulator & Real-Time Debugger** window appears.

True-Time Simulator & Real-Time Debugger Window




NOTE The **Source** and **Assembly** panes display `main.c` program and code.

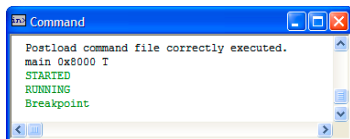
2. Set breakpoints.



- Point at a C statement in the **Source** window and right-click — **Source** context menu appears.
- Select **Set Breakpoint** — Permanent breakpoint mark is set.

project.

- a. From the main menu of the **True-Time Simulator & Real-Time Debugger** window, select **Run** — Run menu appears.
- b. Select **Start/Continue** or click on **Start/Continue** icon  — the program executes till the first breakpoint and the **Command** pane displays the program status.

Command Pane

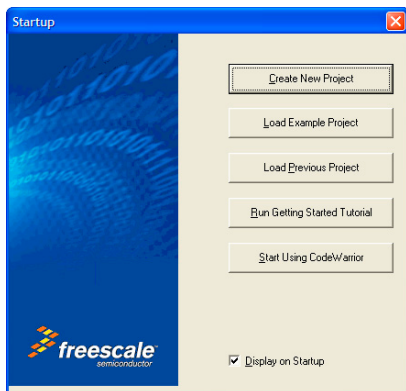


4. Click the **Start/Continue** icon  — the simulator resumes execution.
5. Click the **Halt** icon  — the simulator stops program execution.
6. From the **Debugger Simulator** window toolbar, select **File > Exit** to exit the debugger.
7. From the IDE main window toolbar, select **File > Exit** to exit the CodeWarrior IDE.

Section D: Creating and Building an S12(X) Project with XGATE Support

1. Create a new project.
 - a. Select **Start > Programs > Freescale CodeWarrior > CodeWarrior Development Studio for S12(X) V5.0 > CodeWarrior IDE** — the IDE starts and the **Startup** dialog box appears.

Startup Dialog Box

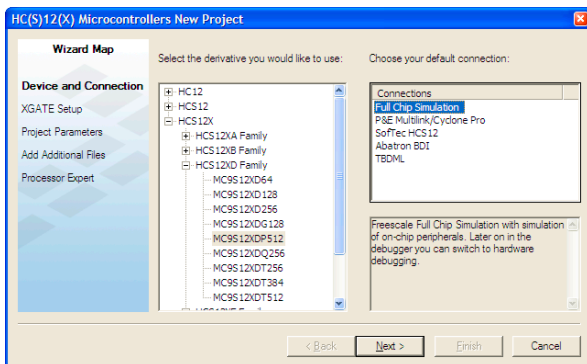


- b. Click **Create New Project** — the **Device and Connection** page appears.

NOTE This section of the quick start demonstrates using the New Project Wizard. We use an MC9S12XDP512 target as an example.

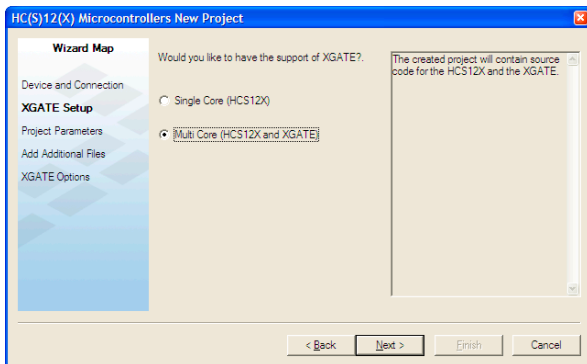
- c. Expand **HCS12X** and **HCS12XD Family** and select the **MC9S12XDP512** derivative.

Device and Connection Page



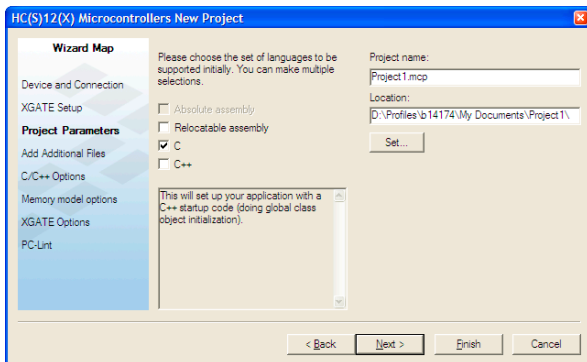
- d. Click **Next** — the **XGATE Setup** page appears.

Multi Core (HCS12X and XGATE) option button.XGATE Setup Page



- e. Click **Next** — the **Project Parameters** page appears.

Project Parameters Page



NOTE The IDE automatically creates a folder with the same name in specified location. The IDE automatically adds `.mcp` extension when it creates project.

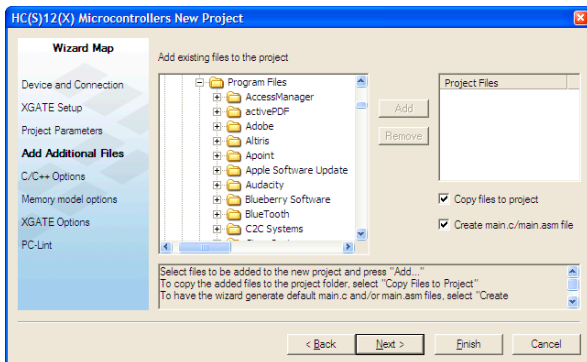
- f. In the **Location** text box enter location to store the project.
 g. Select the **C** checkbox as language to be supported by the project.

... You can select **Finish** button to accept defaults for remaining options.

- h. Click **Next** — the **Add Additional Files** page appears.

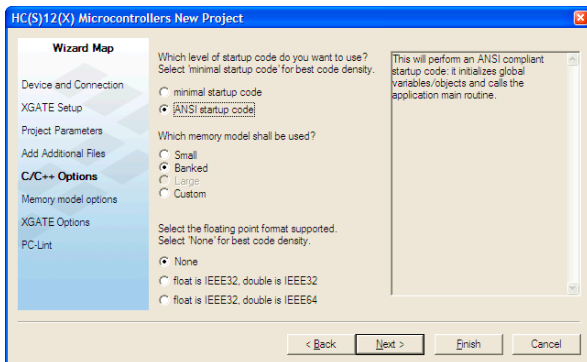
This page allows you to browse folders and add or remove files to or from the project.

Add Additional Files Page



- i. Click **Next** — the **C/C++ Options** page appears.

C/C++ Options Page

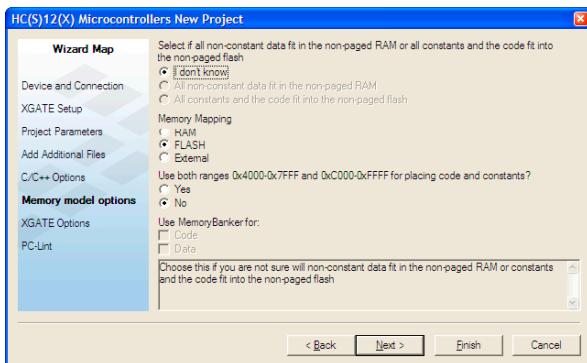


- j. Select the **ANSI startup code** option button as level of startup code to use.
- k. Select the **Banked** option button as memory model to use.

Click the **None** option button for floating point format to support.

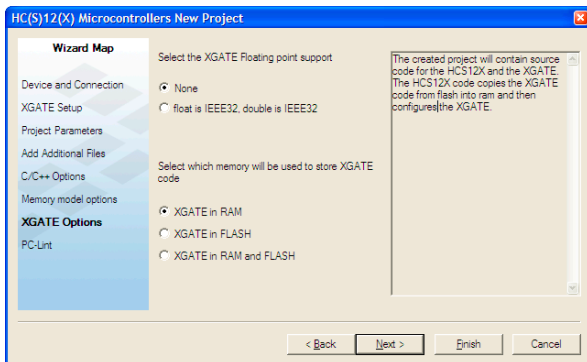
- m. Click **Next** — the **Memory model options** page appears.
- n. Select **FLASH** from the **Memory Mapping** option buttons.

Memory Model Options Page



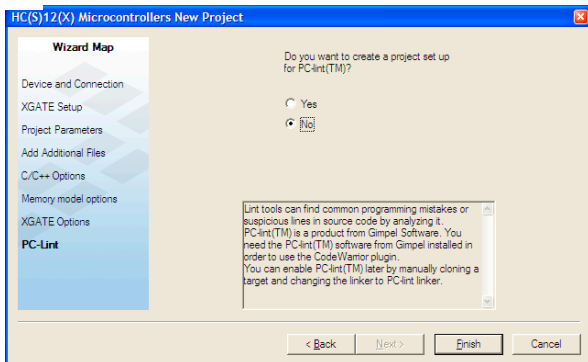
- o. Click **Next** — the **XGATE Options** page appears.
- p. Select **None** from the **Select the XGATE Floating point support** option buttons.
- q. Select **XGATE in RAM** from the **Select which memory will be used to store XGATE code** option buttons.

XGATE Options Page



- r. Click **Next** — the **PC-Lint** page appears.

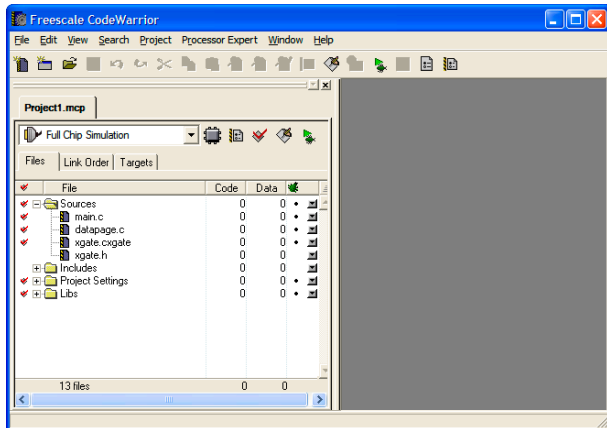
PC-Lint Page



- s. Select **No** from the **Do you want to create a project set up for PC-lint(TM)** option buttons.
- t. Click **Finish** — the IDE creates a project according to your specifications; the project window appears, docked at left side of main window.

NOTE To undock project window, double-click the double gray lines. To re-dock window, right click in title tab and select **Docked**.


Project Window



1. Undock Project window, double-click docking handle (double gray lines at top of window). To re-dock window, right click in title bar of Project Window, and select **Docked**.

2. Select a connection.

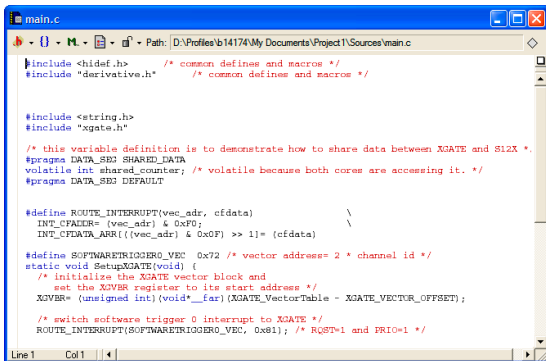
For this example, we specified Full Chip Simulation (FCS).

- a. To change the derivative and connection, click **Change MCU/Connection** icon .
- b. Select **Full Chip Simulation** from the drop-down list.

3. Edit the source code.

- a. Double click `main.c` in the **Sources** folder – the **Editor** window opens displaying contents of file.
- b. Make changes to the contents of `main.c` file, if desired
- c. If you make changes to the file, from IDE main menu bar, select **File > Save** – IDE saves the changes.

main.c in Editor Window



```

#include <hidef.h> /* common defines and macros */
#include "derivative.h" /* common defines and macros */

#include <string.h>
#include "xgate.h"

/* this variable definition is to demonstrate how to share data between XGATE and S12X */
#pragma DATA_SECTION(shared_counter, "XGATE_SHARED_COUNTER")
volatile int shared_counter; /* volatile because both cores are accessing it. */
#pragma DATA_SECTION(shared_counter, "XGATE_SHARED_COUNTER")

#define ROUTE_INTERRUPT(vec_adr, cfddata) \
    INT_CFAADDR = (vec_adr) & 0xF0; \
    INT_CFDATA_ADR[(vec_adr) & 0x0F] >> 1) = (cfddata)

#define SOFTWARE_TRIGGER0_VEC 0x72 /* vector address= 2 * channel id */
static void SetupXGATE(void) {
    /* initialize the XGATE vector block and
    set the XGATE register to its start address */
    XGATE = (unsigned int)(void__far)(XGATE_VectorTable - XGATE_VECTOR_OFFSET);

    /* switch software trigger 0 interrupt to XGATE */
    ROUTE_INTERRUPT(SOFTWARE_TRIGGER0_VEC, 0x81); /* RQST=1 and PRIO=1 */
}

```

4. Add files, if required.

- a. In the project window, select a folder.
- b. From the IDE main menu bar, select **Project**.
- c. Select **Add Files** — the **Select files to add** dialog box appears.
- d. Navigate to the directory that contains file you want to add.
- e. Select filename of the file you want to add to project.



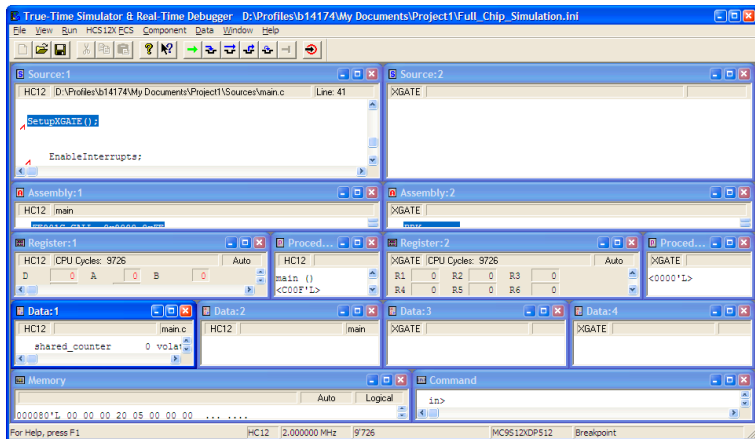
Open — Project messages appear indicating access path has been added to target, if path is new to the project.

- g. In the project window, filename of the added file appears under the selected folder.
5. Build the project.
- a. From the IDE main menu bar, select **Project**.
 - b. Select **Make** — the IDE builds (assembles, compiles, and links) project; the **Error & Warnings** window opens showing any error messages and warning messages.


Section E: Debugging an S12(X) Project with XGATE Support

1. Start the debugger.
 - a. Click on the project window title bar to activate the project window.
 - b. From the main menu bar, select **Project > Debug** — the **True-Time Simulator & Real-Time Debugger** window appears.

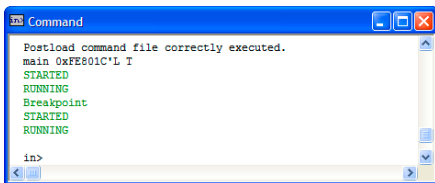
True-Time Simulator & Real-Time Debugger Window





NOTE The **Source** and **Assembly** panes display `main.c` program and code. The left set of windows displays the state of HCS12X core and the right set displays the state of XGATE core.

- a. From the main menu of the **True-Time Simulator & Real-Time Debugger** window, select **Run** — Run menu appears.
- b. Select **Start/Continue** or click on **Start/Continue** icon  — the program executes till the first breakpoint and the **Command** pane displays the program status.

Command Pane



3. Click the **Start/Continue** icon  — the simulator resumes execution.
4. Click the **Halt** icon  — the simulator stops program execution.
5. From the **Debugger Simulator** window toolbar, select **File > Exit** to exit the debugger.
6. From the IDE main window toolbar, select **File > Exit** to exit the CodeWarrior IDE.

Congratulations!

You have successfully created, built, and debugged an S12(X) project and an S12(X) project with XGATE Support using the CodeWarrior Development Studio for S12(X) V5.0 software!



and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

Copyright © 2006-2009 Freescale Semiconductor, Inc. All rights reserved.

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

How to Contact Us

Corporate Headquarters	Freescale Semiconductor, Inc. 6501 William Cannon Drive West Austin, Texas 78735 U.S.A.
World Wide Web	http://www.freescale.com/codewarrior
Technical Support	http://www.freescale.com/support

Revised: 27 February 2009