Using UART with a SmartFusion cSoC

Libero SoC and SoftConsole Flow Tutorial

Microsemi

Table of Contents

Introduction 3
Tutorial Requirements 4
Objective 4

Working with Libero SoC and SoftConsole 5
Step 1 – Creating a Libero SoC Project 5
Step 2 – Configuring MSS Peripherals 6
Step 3 – Generating the MSS Component 9
Step 4 – Generating the Program File 11
Step 5 – Programming SmartFusion Board Using FlashPro 12
Step 6 – Building the Software Application Through SoftConsole 13
Step 7 – Configuring the Serial Terminal Emulation Program 16
Step 8 – Installing Drivers for the USB to RS232 Bridge 18
Step 9 – Debugging the Application Project Using SoftConsole 18
Step 10 – Debugging the Application Using the printf Statement in SoftConsole 22
Step 11 – Building Executable Image in Release Mode 27

Appendix A – Libero SoC Catalog Settings 29

Appendix B – Firmware Catalog Settings 31

List of Changes 33

Product Support 35
Customer Service 35
Customer Technical Support Center 35
Technical Support 35
Website 35
Contacting the Customer Technical Support Center 35
ITAR Technical Support 36

Introduction

This tutorial describes how to use Microsemi tools to develop an application that can be implemented on SmartFusion® customizable system-on-chip (cSoC) devices.

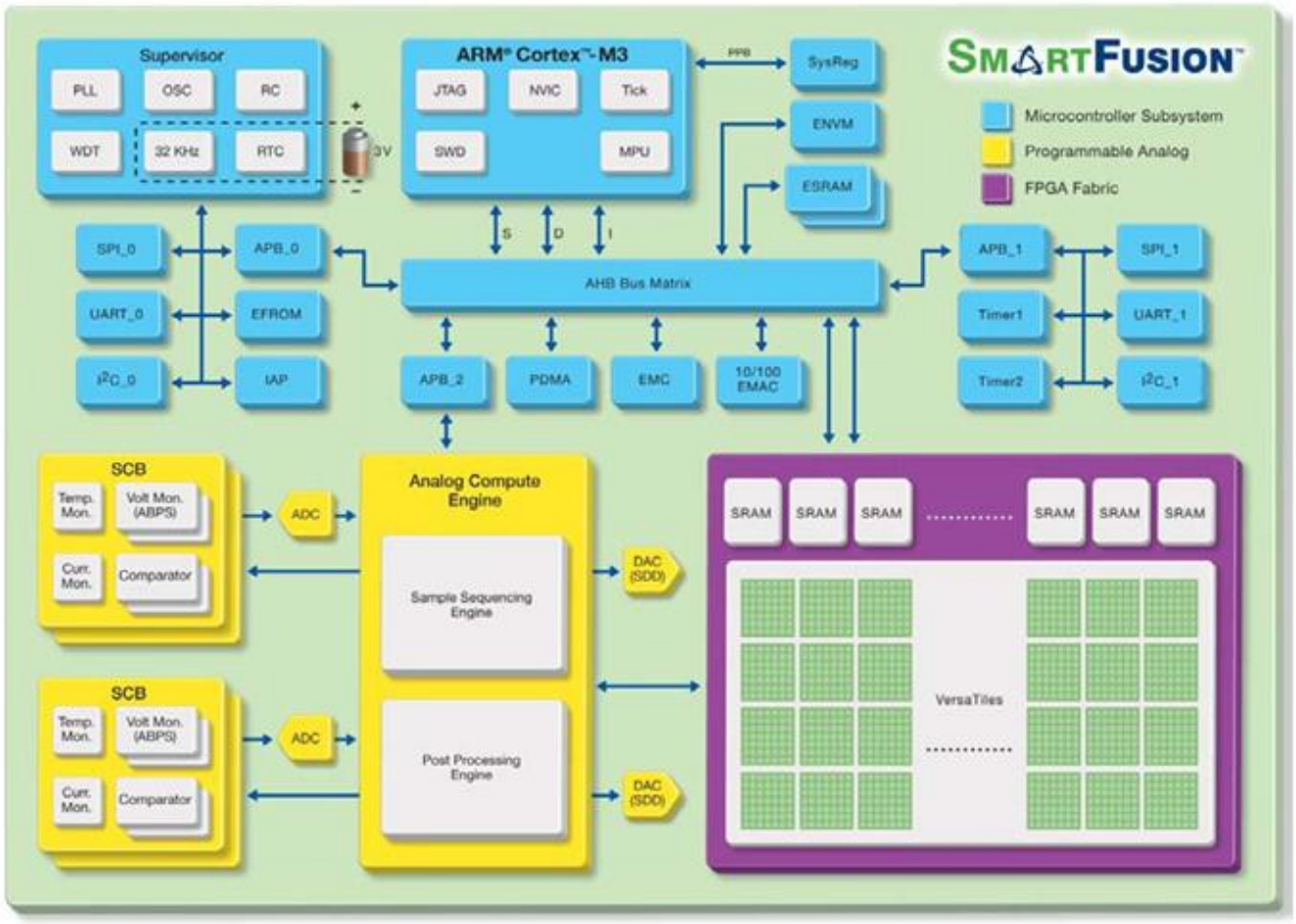

Figure 1 · SmartFusion cSoC Block Diagram

After completing this tutorial, you will be familiar with the following:

- Creating and setting up a Libero® system-on-chip (SoC) project for a SmartFusion cSoC device
- Using SmartDesign microcontroller subsystem (MSS) configurator to configure the peripherals and generate an embedded memory configuration file (EFC)
- Generating the programming file to program the SmartFusion cSoC device
- Opening the project in SoftConsole IDE from Libero SoC and writing application code
- Compiling application code
- Creating and launching a debug session

Tutorial Requirements

Software Requirements

This tutorial requires the following software installed on your PC:

- Libero SoC v10.0 or later, which can be downloaded from www.microsemi.com/soc/download/software/libero/default.aspx.
- Microsemi SoftConsole v3.3 or later, which is installed as a part of Libero SoC v10.0 installation or can be downloaded from www.microsemi.com/soc/download/software/softconsole/default.aspx.

Hardware Requirements

This tutorial requires the following hardware.

- SmartFusion Evaluation Kit Board or SmartFusion Development Kit Board.
- Two USB cables (programming and communication)—one for connecting the programmer to your PC and the other to connect the universal asynchronous receiver/transmitter (UART) interface on the board to the PC.

Associated Project Files

You can download the associated solution project files for this tutorial from the Microsemi website: www.microsemi.com/soc/download/rsc/?f=SmartFusion_UART_HW_flow_LiberoSoftConsole_tutorial_DF.

You can download the programming file (*.stp) for this tutorial from the Microsemi website: www.microsemi.com/soc/download/rsc/?f=SmartFusion_UART_HW_flow_LiberoSoftConsole_tutorial_PF.

MSS Components Used

- ARM® Cortex™-M3 processor
- Communications matrix
- Clock conditioning circuit (CCC)
- UART0

Target Board

SmartFusion Evaluation Kit Board (A2F-EVAL-KIT) or SmartFusion Development Kit Board (A2F-DEV-KIT).

Objective

The objective of this tutorial is to instruct how to configure the SmartFusion UART hard (on-chip) peripherals and access them through the ARM Cortex-M3 processor.

Design Steps

Following are the major steps to be executed for this tutorial:

- Create a Libero SoC project for a SmartFusion cSoC.
- Configure the SmartFusion cSoC MSS peripherals.
- Generate the MSS component.
- Perform synthesis and layout, and generate a programming file to program the SmartFusion cSoC device.
- Program the SmartFusion A2F200M3F or A2F500M3F cSoC device.
- Open the software project in SoftConsole from Libero SoC and write the application code.
- Run the application to communicate with HyperTerminal through UART0 on the SmartFusion Evaluation Kit Board or the SmartFusion Development Kit Board.

Working with Libero SoC and SoftConsole

This section describes how to create a new Libero SoC project, configure the microcontroller subsystem (MSS), program the design on the SmartFusion cSoC board, and run an application program in the SoftConsole IDE.

Step 1 – Creating a Libero SoC Project

1. Launch Libero SoC v10.0 or later.
2. From the **Project** menu, select **New Project**. Enter the information shown below in the Libero **New Project** dialog box.
   - Name: SmartFusion_UART
   - Location: <..> (For example, C:\Microsemiprj\UART_Libero_SoftConsole)
   - Family: SmartFusion
   - Die: If you are using SmartFusion Evaluation Kit Board, enter A2F200M3F; if you are using SmartFusion Development Kit Board, enter A2F500M3F.
   - Package: 484 FBGA
   - Speed: STD

   Leave others as default.

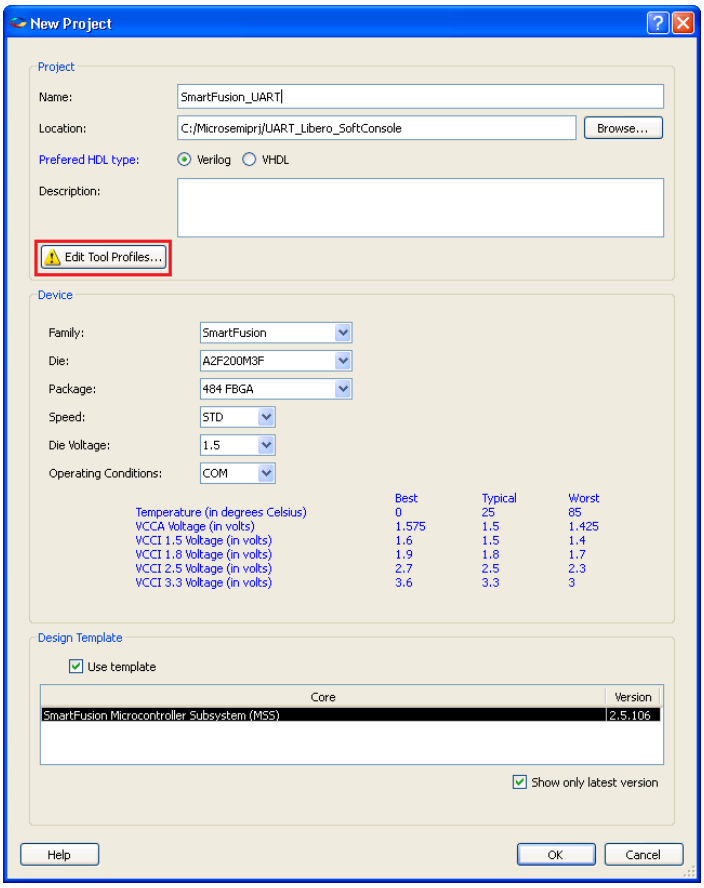

Figure 2 · New Project Dialog Box

3. Click **Edit Tool Profiles** and add SoftConsole by clicking **Software IDE**, as shown in Figure 3 · .

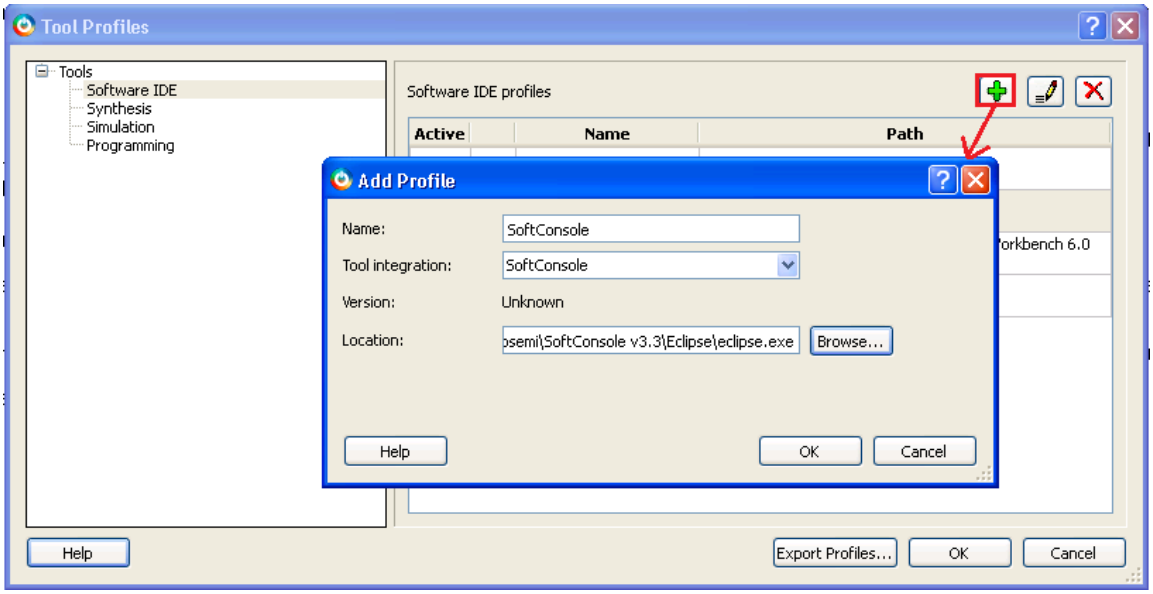

Figure 3 · Selecting SoftConsole as Software IDE

4. After adding the Profile, click **OK** to close the **Add Profile** dialog window.

   Repeat steps above for Synthesis, Simulation, and Programming and then click **OK** to close the Tool Profiles dialog window.

5. Select the MSS core in the **New Project** dialog box and click **OK**.

6. The project is created and the Libero SoC window appears, as shown in Figure 4 · . The SmartDesign "SmartFusion_UART" is created with instantiation of the MSS component.

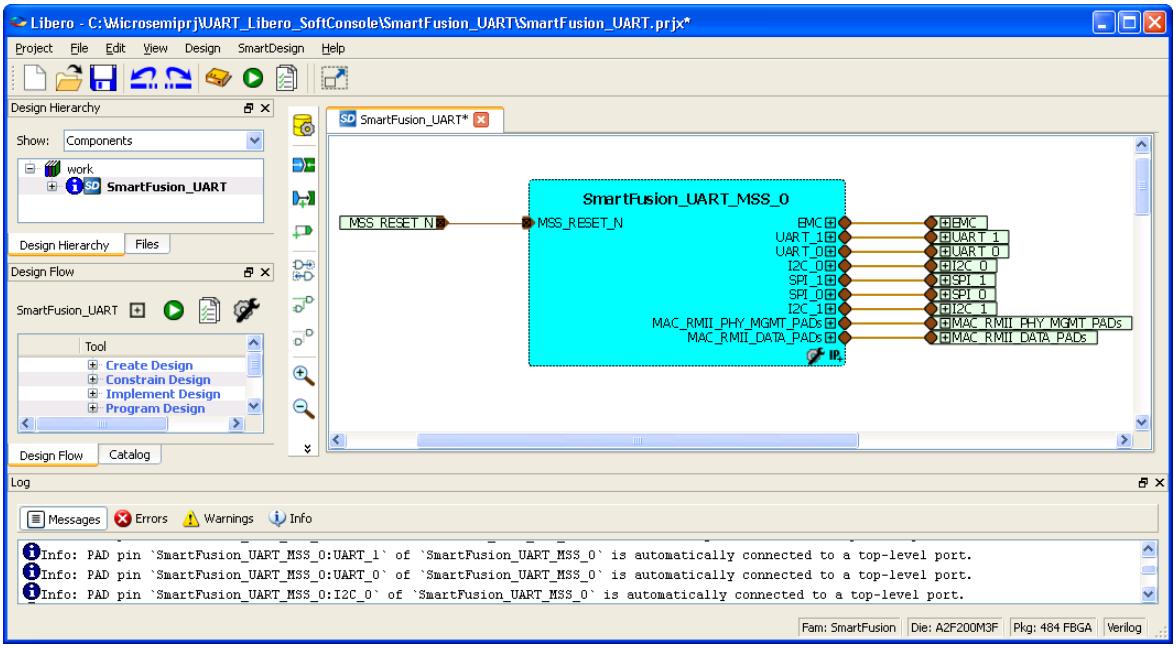

Figure 4 · The Libero SoC Window After Completing New Project Wizard

Step 2 – Configuring MSS Peripherals

1. Double-click the **SmartFusion_UART_MSS_0** component to configure the MSS. The MSS is displayed in the SmartDesign Canvas in a new tab, as shown in Figure 5 · .

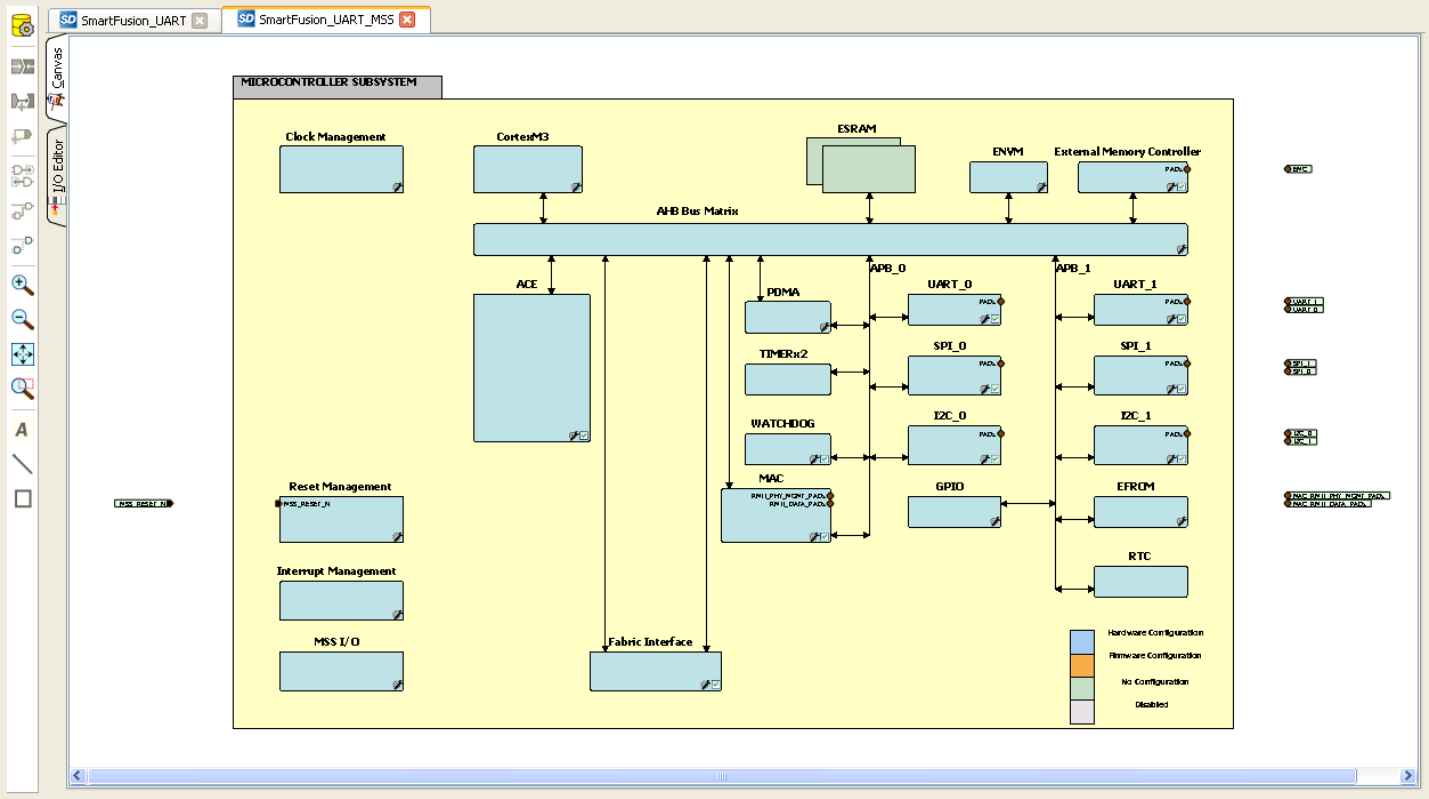

Figure 5 · MSS in the SmartDesign Canvas

The enabled MSS peripherals are highlighted in blue, and can be configured in the hardware. The disabled MSS peripherals are shown in gray.

To disable a peripheral that is not required, select the peripheral, right-click, and clear the **Enabled** check box, or clear the check box in the lower right corner of the **peripheral** box.

An enabled peripheral is shown in Figure 6 · .

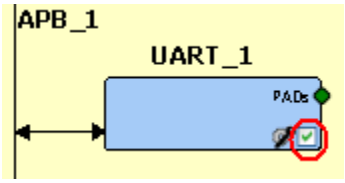

Figure 6 · Enabling a Peripheral

2. The following peripherals must be disabled because they are not used in this tutorial: analog compute engine (ACE), MAC, WATCHDOG, Fabric Interface, SPI0, SPI1, 2C0, I2C1, UART1, and EMC.

3. After disabling the peripherals, the SmartDesign canvas looks similar to Figure 7 · .

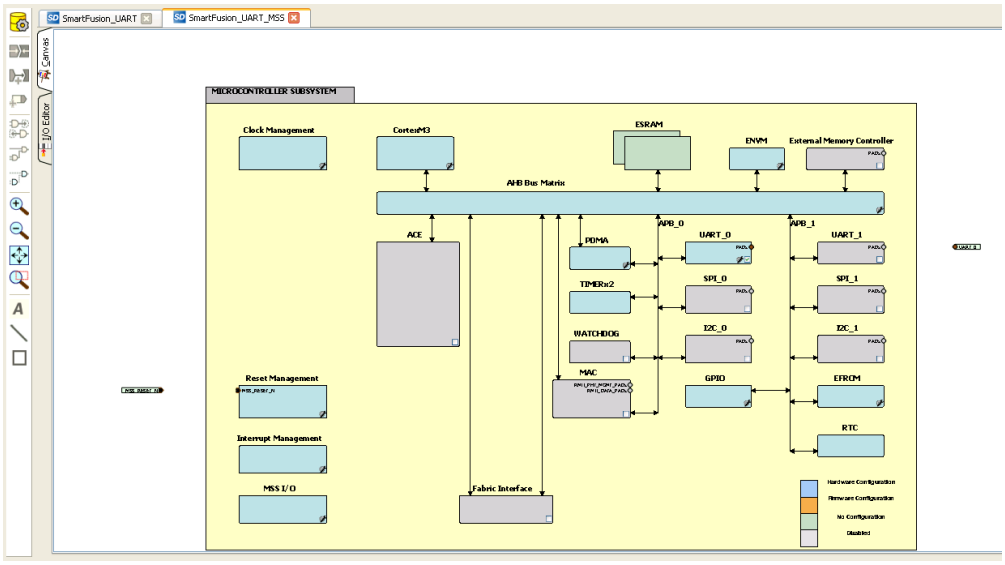

Figure 7 · Used MSS Peripherals

4. Double-click the **Clock Management** block and configure as shown below:
   - CLKA: On-Chip RC Oscillator
   - MSS clock source: PLL output
   - MSS clock frequency: 80 MHz

   Use default settings for all other fields.

   After completing the configuration, click **OK**.

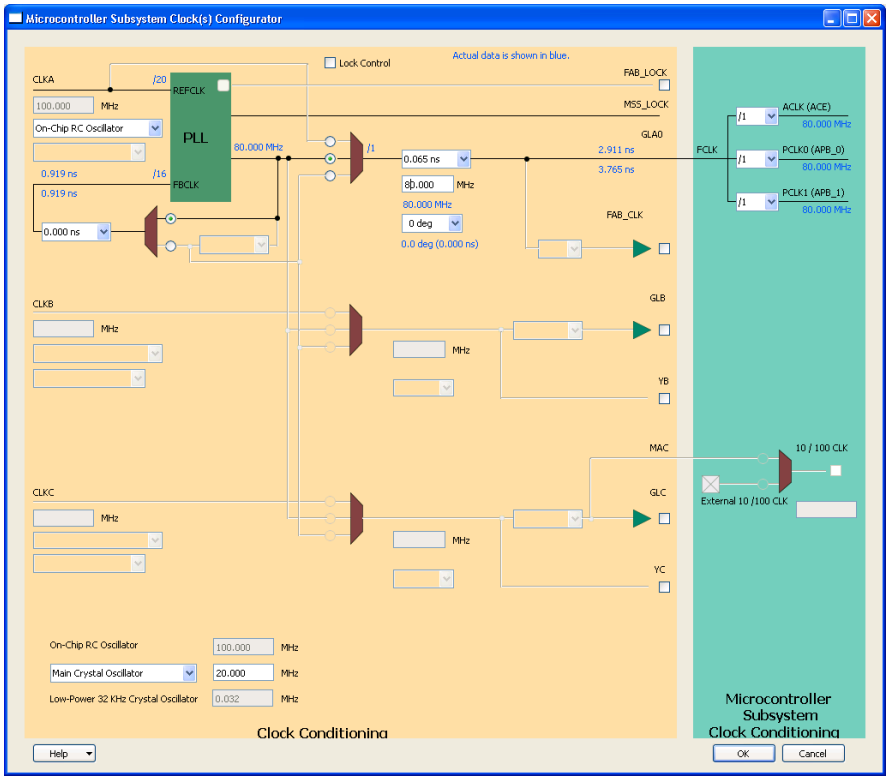

Figure 8 · MSS Clock Configuration

5. Click **File > Save** to save SmartFusion_UART_MSS.

Step 3 – Generating the MSS Component

1. Right-click the **SmartFusion_UART_MSS_0** component in the **SmartFusion_UART_MSS** tab and select **Update Instance(s) with Latest Component…**, as shown in Figure 9 · .

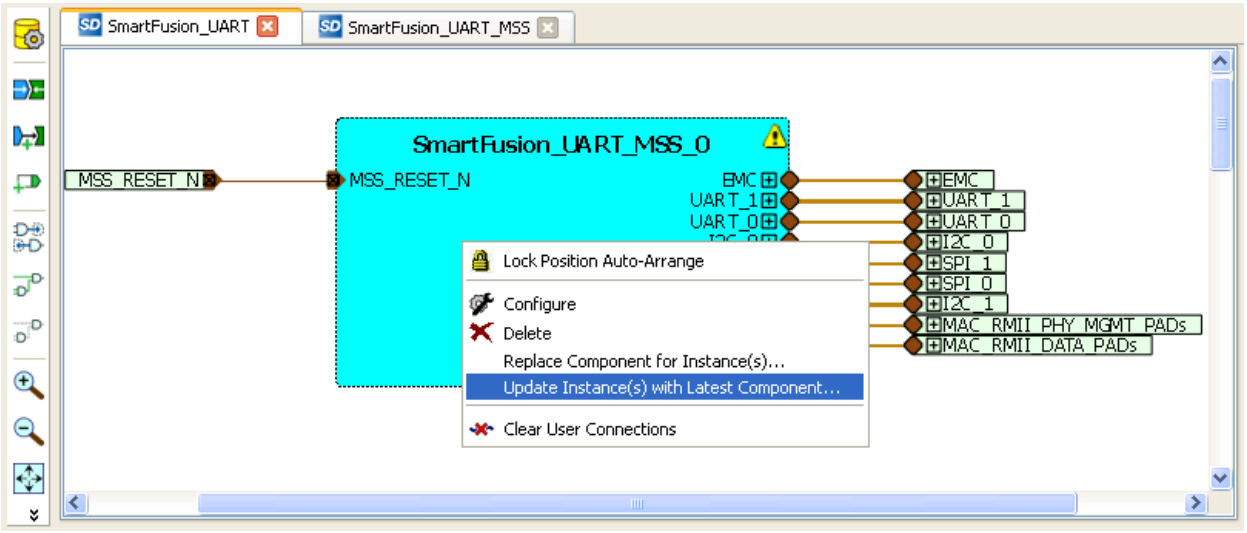

Figure 9 · Updating the MSS

2. Select **Design > Configure Firmware**, as shown in Figure 10 · .

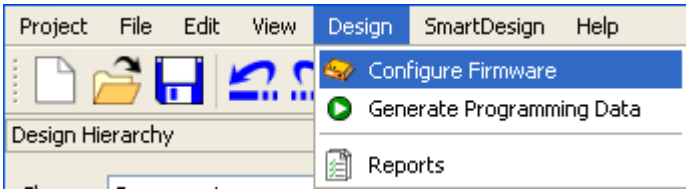

Figure 10 · Opening Design_Firmware

3. On the **DESIGN_FIRMWARE** tab, clear the **Generate** check boxes for all the peripherals for which you do not need to generate the firmware. Click **Configuration** on the SmartFusion_CMSIS_PAL_0 instance and select **SoftConsole** as the configuration.

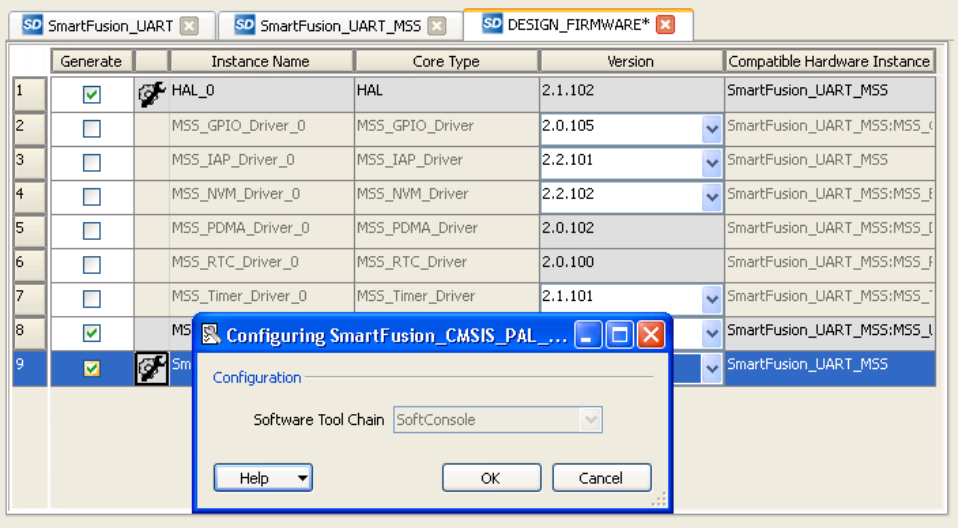

Figure 11 · Configuring SmartFusion_CMSIS_PAL_0

Note: Check whether or not you are able to see the latest version of the drivers without any warning or error indicating that firmware is missing from the Vault. If missing, refer to Appendix B – Firmware Catalog Settings.

4. Click **File > Save** to save the **DESIGN_FIRMWARE**.
5. **Save** the design and generate the component by clicking **Generate Component** or click **SmartDesign > Generate Component**.

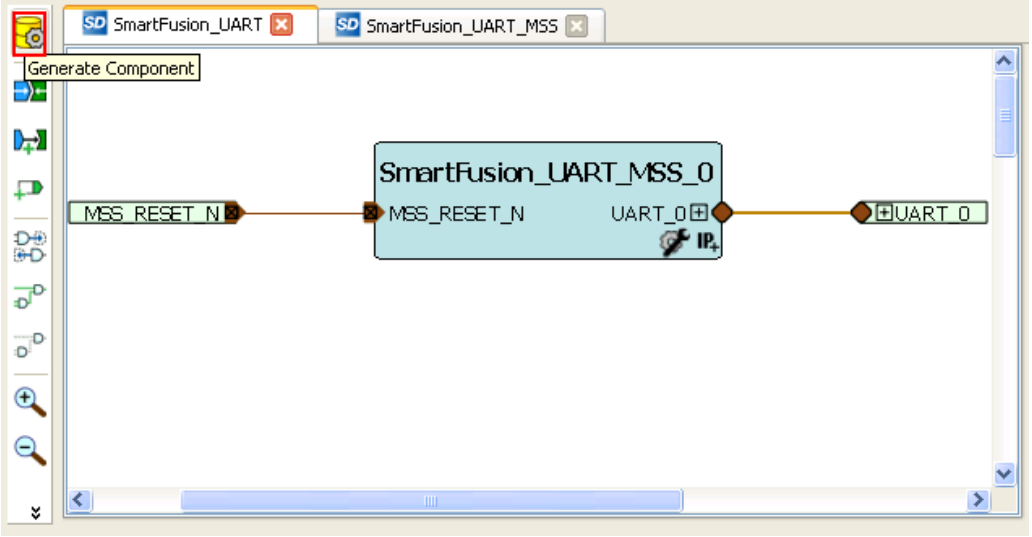

Figure 12 · Generating the Component

6. After successful generation of the MSS component, the log window displays the message "**Info: 'SmartFusion_UART' was successfully generated. Open datasheet for details**". The datasheet contains project information such as generated files, used I/Os, and memory map.
7. Ensure that the SoftConsole folder is created with the folders and files, as shown in Figure 13 · .

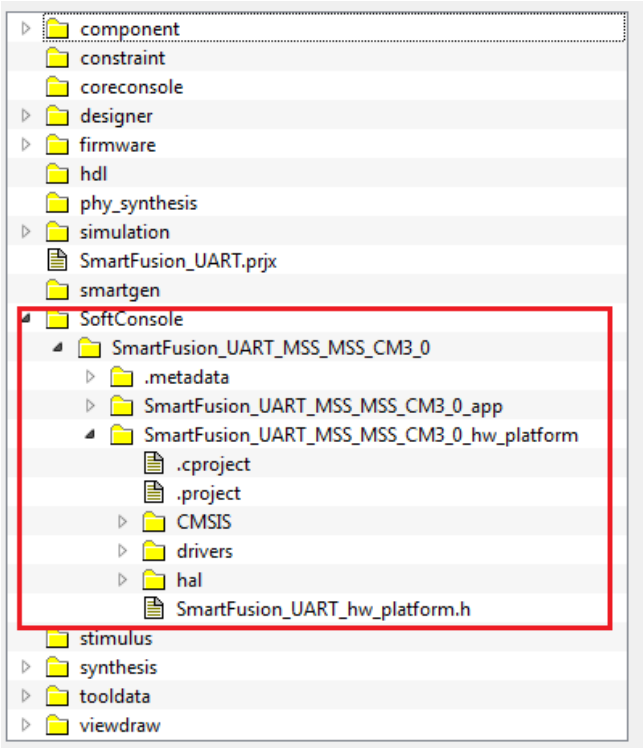

Figure 13 · Files Window After Generating the Project

Step 4 – Generating the Program File

Libero SoC provides the push-button flow for generating the programming data of a project in a single step. By clicking **Generate Programming Data**, you can complete the synthesis, place-and-route, verify the timing, and generate the programming file. You can also complete the flow by running the synthesis and place-and-route tools in interactive mode step-by-step. For more information, refer to the *Libero SoC Quick Start Guide*.

Push- Button Design Flow

Click **Generate Programming Data**, as shown in Figure 14 · to complete the place-and-route, verify timing, and generate the programming file. This completes the *.fdb file generation.

Figure 14 · Generating Programming Data

The **Design Flow** window is displayed as shown in Figure 15 · .

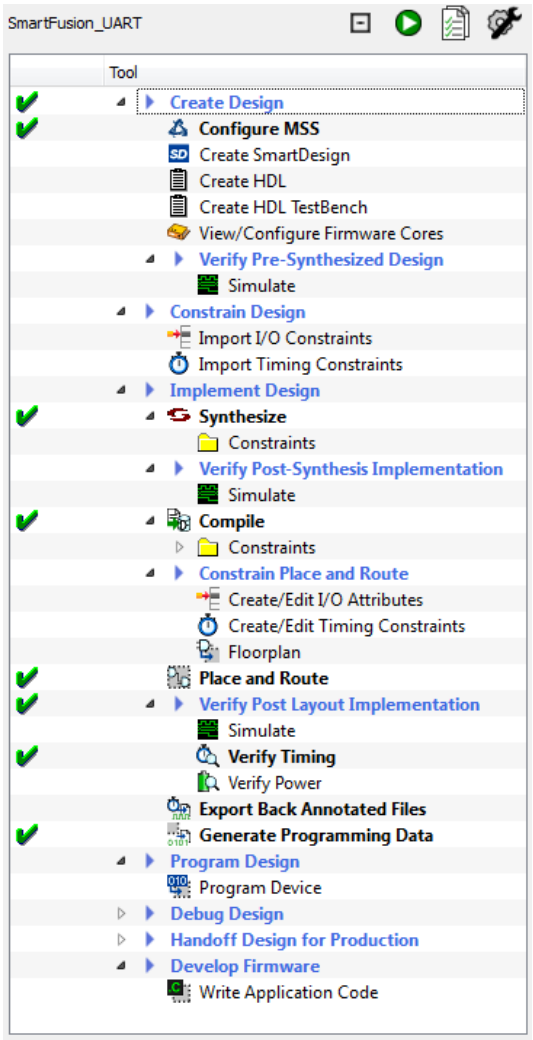

Figure 15 · Design Flow Window After Building the Project

Step 5 – Programming SmartFusion Board Using FlashPro

Before you proceed with programming the device, ensure that the low cost programming stick (LCPS) or FlashPro4 is properly connected to the board. Use the following details to ensure the correct jumper settings. Refer to the *SmartFusion Evaluation Kit User's Guide* and the *SmartFusion Development Kit User's Guide* for additional information.

Jumper Settings for SmartFusion Evaluation Kit Board

- JP10: Connect pin 1 and 2.
- JP7: Connect pin 1 and 2 for LCPS programming mode.
- J6: Connect pins 1 and 2 with the jumper.
- JP6: Connect pins 2 and 3 with the jumper.
- J13: Connect the USB cable to the J13 connector. When the cable is connected, the FlashPro4 or FlashPro drivers may be installed if they are not already installed.
- J14: Connect one end of USB mini B cable to J14.

Jumper Settings for SmartFusion Development Kit Board

SW9 must be off (JTAGSEL = H) in order to program the SmartFusion device. SW9 remains in the off position for Libero SoC and SoftConsole programming. Make the jumper settings as shown in the following table:

Table 1 · Jumper Settings for Development Kit Board

Factory Default	Factory Default	Factory Default
JP1: 1–2	JP12: 1–2	JP21: 1–2
JP2: 1–2	JP13: 1–2	JP22: 2–3
JP4: 1–3; 7–9	JP14: 1–2	JP23: 1–2
JP5: 1–2; 3–4	JP15: 1–2	JP24: 1–2
JP6: 2–3	JP16: 2–3	JP27: 1–2
J7: 2–3; 6–7; 10–11; 14–15	JP17: 2–3	JP28: 1-2
JP7: 1–2	JP18: 1–2	J32: 1–2; 3–4; 5–6
JP8: 3-4; 7-8; 11-12; 15-16	JP19: 2–3	–
JP11: 1-2	JP20: 1–2	–

Programming the Device

1. Double-click **Program Device** under **Program Design** in the **Design Flow** window to program the SmartFusion cSoC device as shown in Figure 16 · .
2. Click **Yes** to the prompt that the timing constraints are not yet set.

   Note: Do not interrupt the programming sequence; it may damage the device or the programmer. If you face any problems, contact Microsemi SoC Products Group Technical Support at soc_tech@microsemi.com.

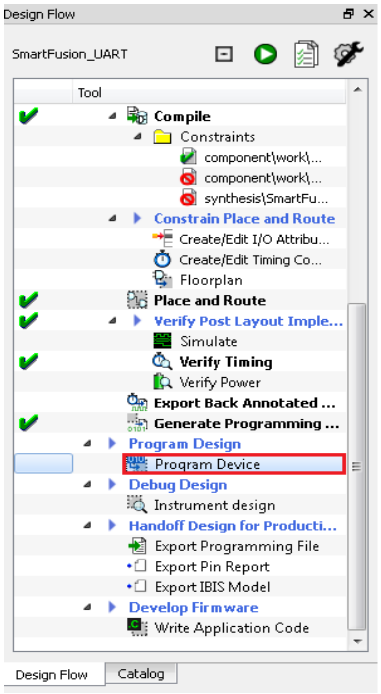

Figure 16 · Design Flow Window

You can use FlashPro interactively by right-clicking **Program Device** in the **Design Flow** window and selecting **Open Interactively**. For more information on FlashPro, refer to the *FlashPro User's Guide*.

Step 6 – Building the Software Application Through SoftConsole

1. From **Libero SoC**, open the SoftConsole project by double-clicking **Write Application Code** under **Develop Firmware** in the **Design Flow** window.

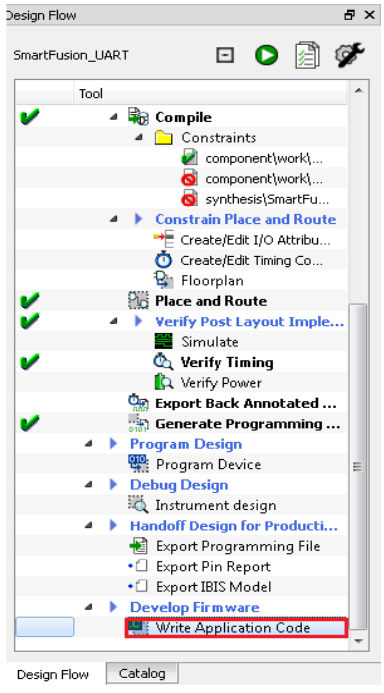

Figure 17 · Invoking SoftConsole from Libero SoC

The SoftConsole perspective looks similar to Figure 18 · .

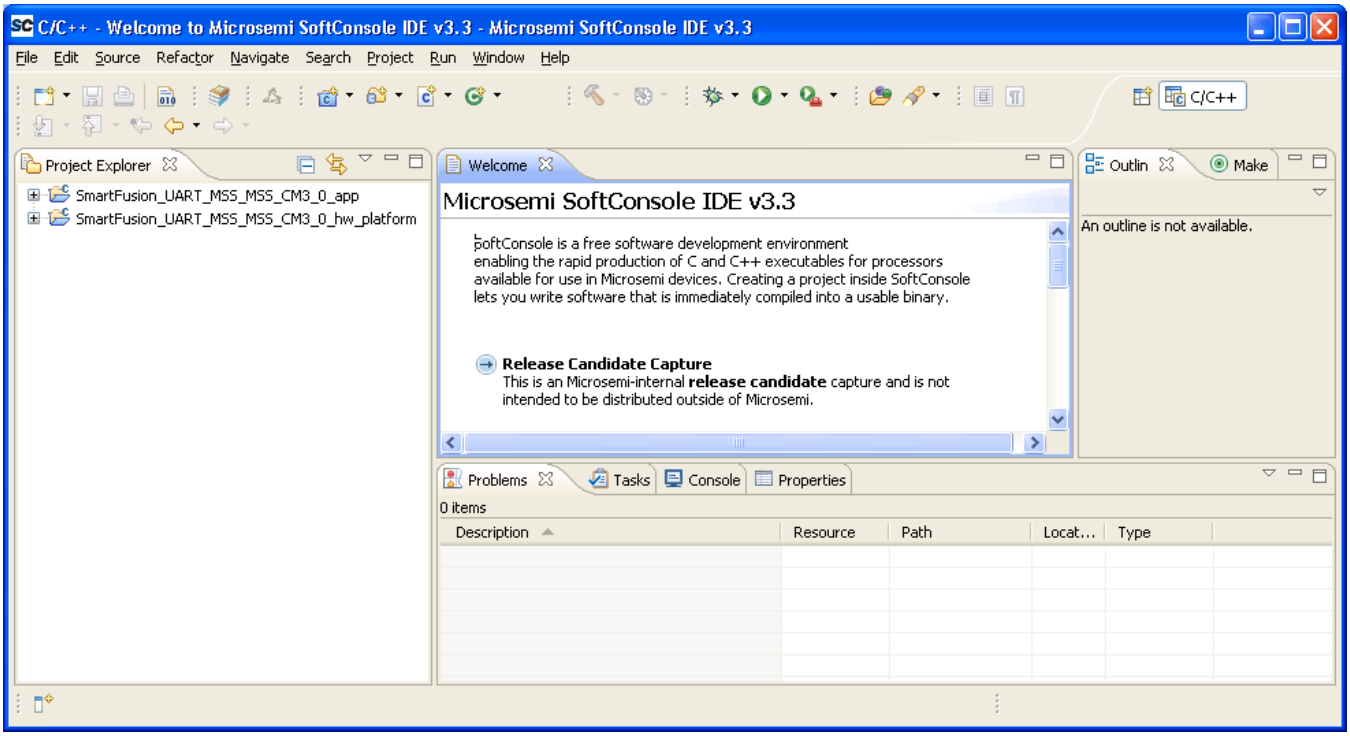

Figure 18 · SoftConsole Workspace

2. Copy the code provided below and paste it in **main.c** file under the SmartFusion_UART_MSS_MSS_CM3_0_app project in the SoftConsole editor, and then delete the existing code.

```
#include "mss_uart.h"
#define  Microsemi_logo \
"\n\r \
**      **  *******   ******  *****    ****    *****  ******  **      **  ******* \n\r \
* *    * *     *     *        *    *  *    *  *       *       * *    * *     *    \n\r \
*  * *  *     *     *        *****   *    *   ****   ******  *  * *  *     *    \n\r \
*   *   *     *     *        *    *  *    *       *  *       *   *   *     *    \n\r \
*       *  *******   ******  *    *   ****   *****   ******  *       *  ******* "

/* Main function */
int main()
{
        const uint8_t greeting[] = "\n\rWelcome to SmartFusion "
                        "- customizable system-on-chip (cSoC)";

    /* Initialize and configure UART0. */
    MSS_UART_init
    (
              &g_mss_uart0,
           MSS_UART_57600_BAUD,
           MSS_UART_DATA_8_BITS | MSS_UART_NO_PARITY | MSS_UART_ONE_STOP_BIT
     );

    /* Send the Microsemi Logo over the UART_0 */
```

```
    MSS_UART_polled_tx_string( &g_mss_uart0, (const uint8_t *)Microsemi_logo);

    /* Send greeting message over the UART_0 */
    MSS_UART_polled_tx( &g_mss_uart0, greeting, sizeof(greeting) );

    while(1)
        {
        }
    return 0;
}
```

3. The SoftConsole window is displayed, as shown in Figure 19 · .

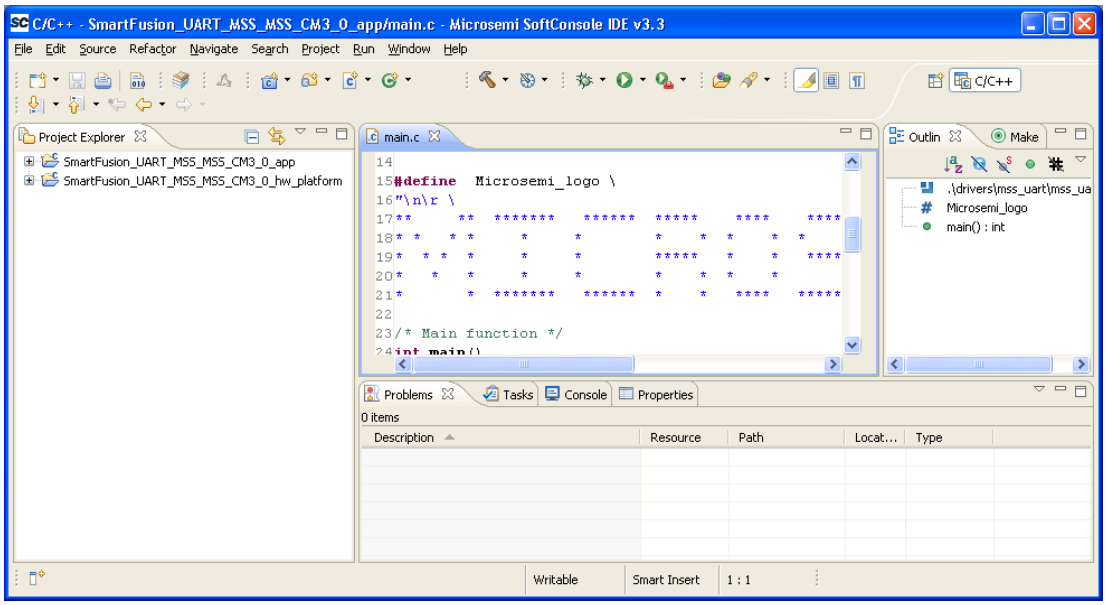

Figure 19 · SoftConsole Workspace

4. Perform a clean build by selecting **Project > Clean**. Accept the default settings in the **Clean** dialog box and click **OK**.

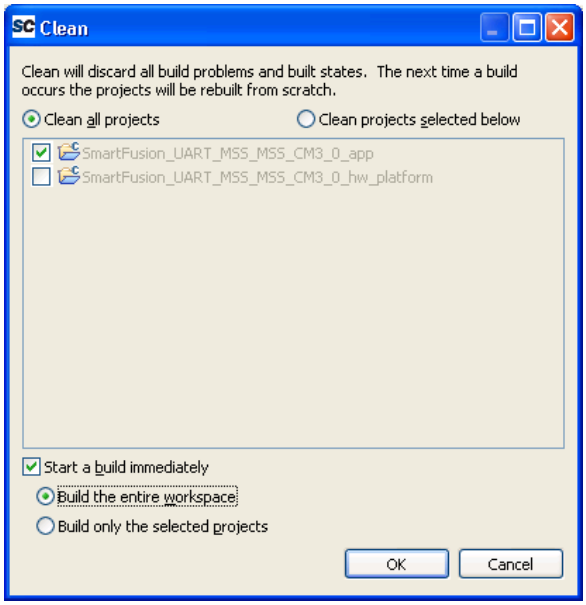

Figure 20 · Settings for a Clean Build

5. Ensure that there are no errors displayed in the **Messages** log. Now follow the next steps to configure HyperTerminal.

Step 7 – Configuring the Serial Terminal Emulation Program

Prior to running the application program, you need to configure the terminal emulator program (HyperTerminal, included with Windows®) on your PC. Follow the steps given below to use the SmartFusion Evaluation Kit Board or the SmartFusion Development Kit Board:

1. Connect a second mini USB cable between the USB connector on the SmartFusion Evaluation Kit Board (or the SmartFusion Development Kit Board) and a USB port of your PC. If Windows prompts you to connect to Windows Update, select **No, not at this time** and click **Next**.
2. If the SFE USB to RS232 Controller drivers are automatically detected (this can be verified in Device Manager), as shown in Figure 21 · , go to the next step. If not, follow Step 8 – Installing Drivers for the USB to RS232 Bridge to install drivers for the USB to RS232 Bridge.

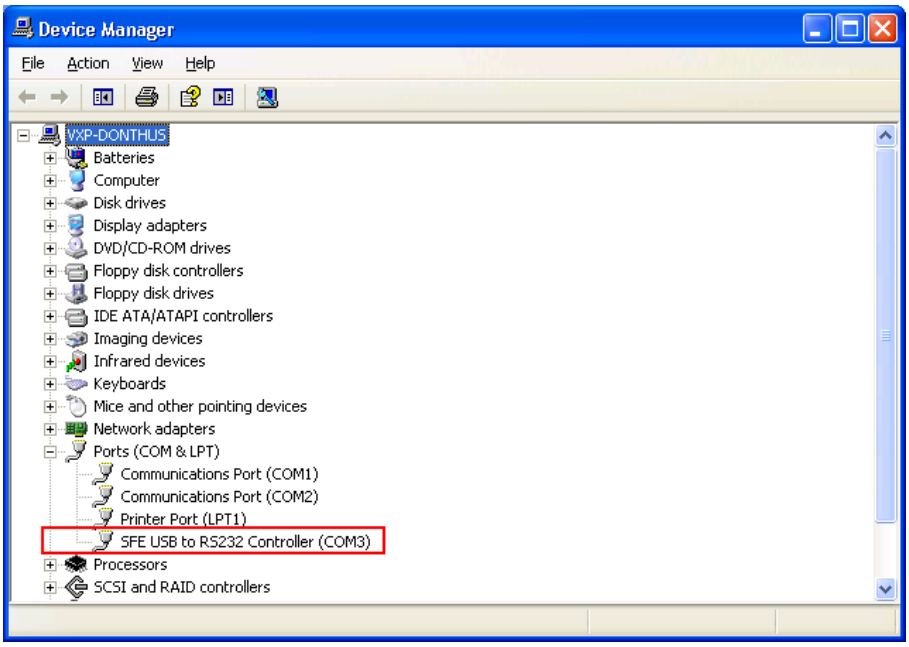

Figure 21 · Device Manager Listing SFE USB to RS232 Controller Drivers

3. From the Windows **Start** menu, select **Programs > Accessories > Communications > HyperTerminal**. This opens HyperTerminal. If your PC does not have HyperTerminal, use any free serial terminal emulation program such as PuTTY or Tera Term. Refer to the *Configuring Serial Terminal Emulation Programs Tutorial* for configuring HyperTerminal, Tera Term, and PuTTY.
4. Enter **Hyperterminal** in the **Name** field in the **Connection Description** dialog box and click **OK**.

Figure 22 · New Connection

5. Select the appropriate COM port (to which USB-RS232 drivers are pointed) from the **Connect using** drop-down list and click **OK**.

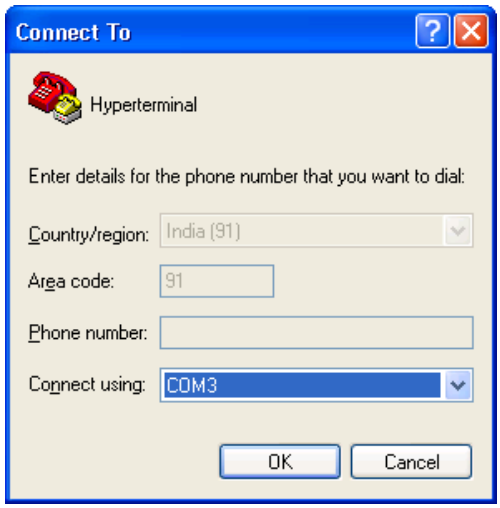

Figure 23 · Selecting the COM Port

6. Set the following in the **COM Properties** window and click **OK**:
   - Bits per second: 57600
   - Data bits: 8
   - Parity: None
   - Stop Bits: 1
   - Flow control: None

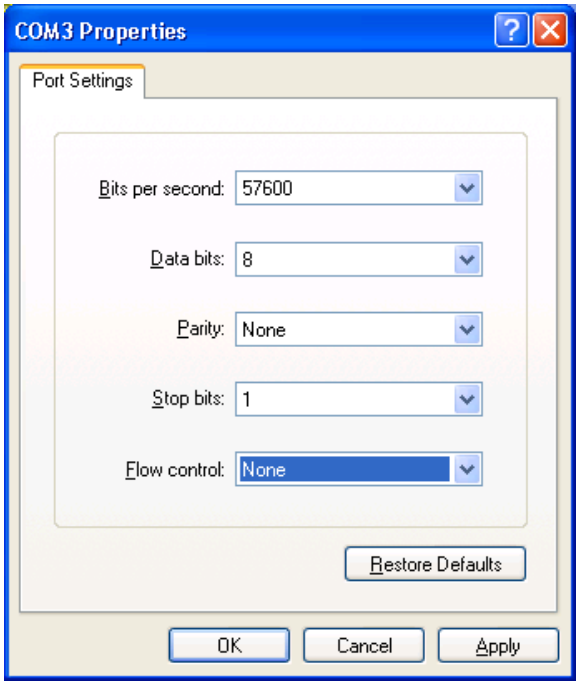

Figure 24 · Setting the COM Properties

7. Click **OK** to close the HyperTerminal Properties dialog box.

You can directly open HyperTerminal next time (without configuring) by selecting **Programs > Accessories > Communications > HyperTerminal > HyperTerminal**.

Step 8 – Installing Drivers for the USB to RS232 Bridge

Note: To install the USB-RS232 drivers, you should have administrative privileges for your PC.

Use the following steps to install drivers for the USB to RS232 Bridge:

1. Unzip the **USB_Drivers.zip** file and browse to the USB Drivers folder. Refer to the instructions in the readme.txt file contained in the zip folder.
2. Double-click (run) the **PreInstaller.exe**.
3. Accept the default installation location and click **Install**.
4. Click **Continue Anyway** if prompted.
5. When the installation is complete, click **OK**. The Ports (COM & LPT) section of the Device Manager lists SFE USB to RS232 Controller under the Ports section of Device Manager.

Step 9 – Debugging the Application Project Using SoftConsole

Follow the steps given below to debug the application project using SoftConsole:

1. Select **Open Debug Dialog** or **Debug Configurations** from the **Run** menu of the SoftConsole. The Debug dialog is displayed.
2. Double-click on **Microsemi Cortex-M3 RAM target**. The **Debug Configurations** window is displayed, as shown in Figure 25 · .

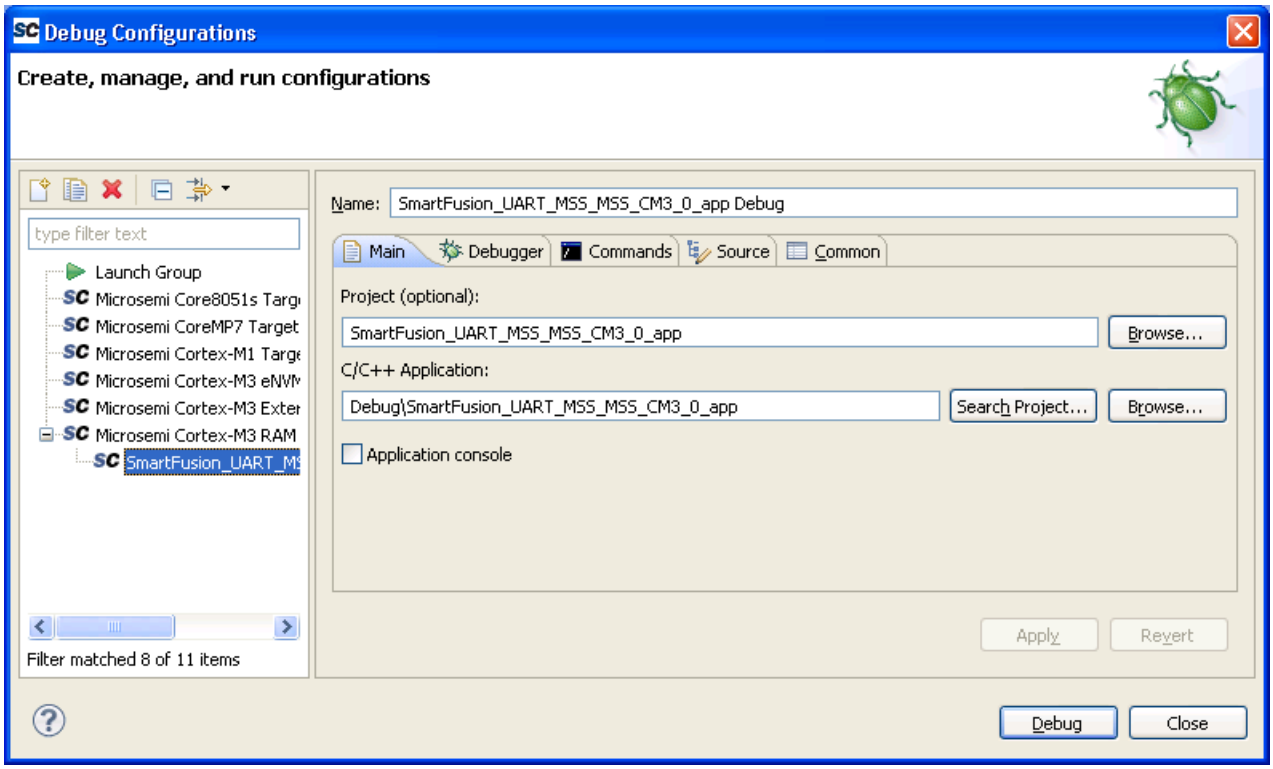

Figure 25 · Debug Configurations Window

3. If the **C/C++ Application** project is not displayed (as seen under **C/C++ Application**), click **Search Project** to select the program to run, as shown in Figure 26 · .

   Note: If you select the project name listed in project explorer and then invoke the debug window, it displays all the fields automatically.

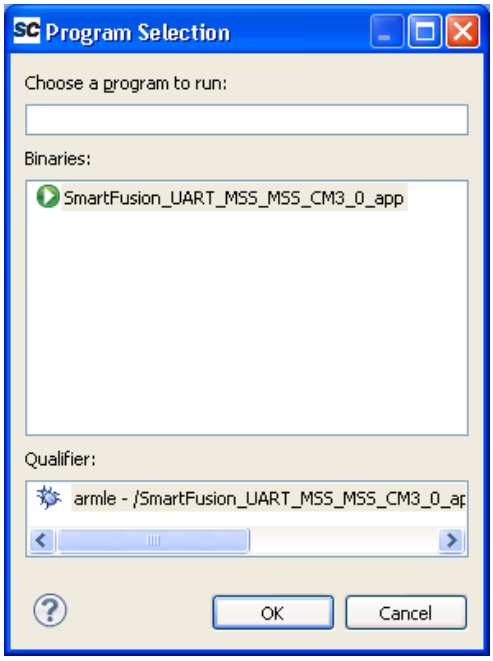

Figure 26 · Select Program

4. On the Debug window, click **Apply** and **Debug**.
5. Click **Yes** when prompted for **Confirm Perspective Switch**. This displays the debug view mode.

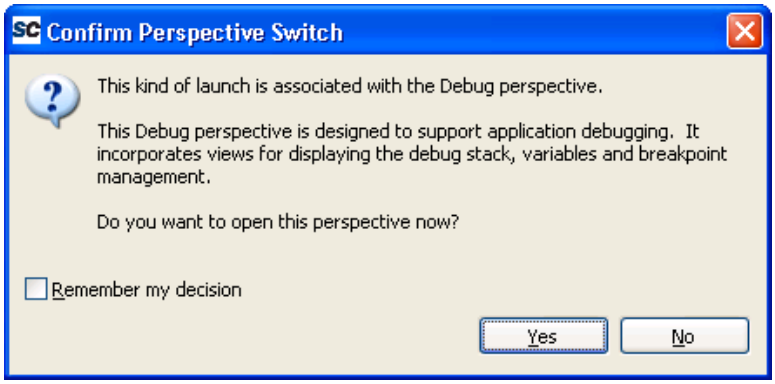

Figure 27 · Confirm Perspective Switch

6. The SoftConsole Debug Perspective window is displayed, as shown in Figure 28 · .

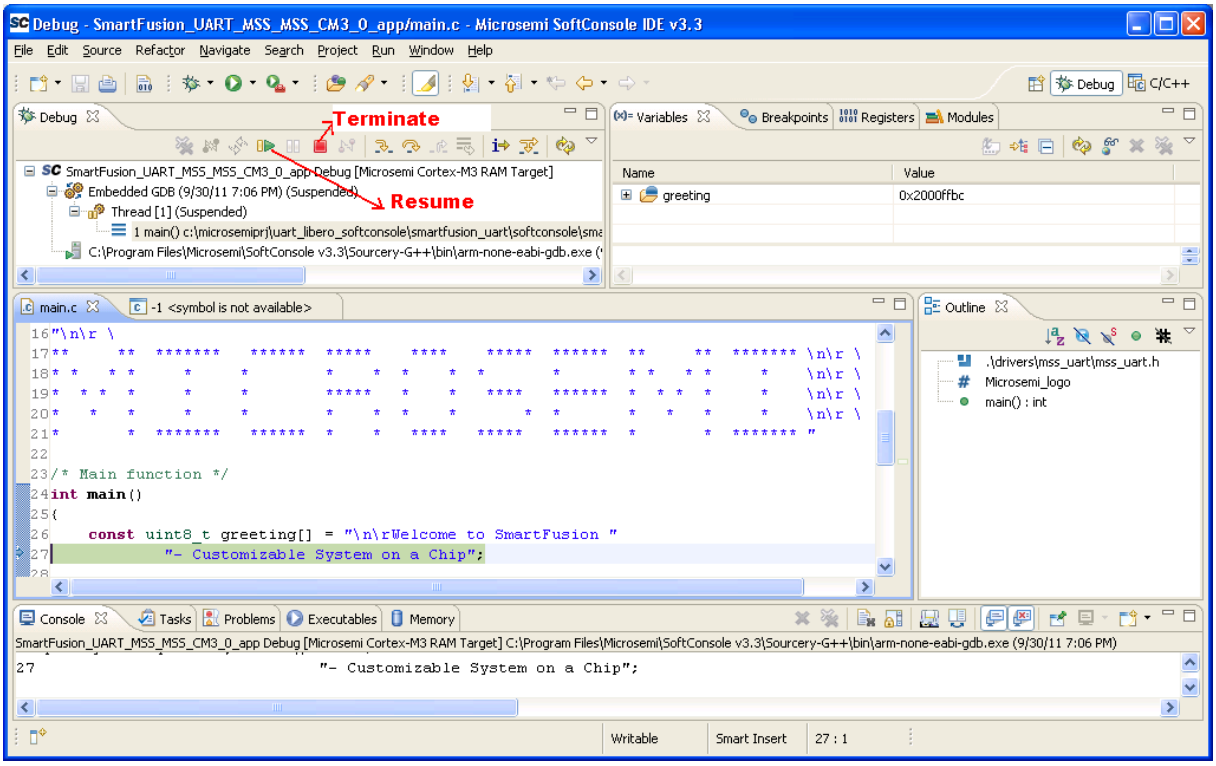

Figure 28 · The SoftConsole Debug Perspective

7. Click **Resume** or press the **F8** key to resume debugging.
8. Observe the HyperTerminal window. It should display the greeting message with the Microsemi name, as shown in Figure 29 · .

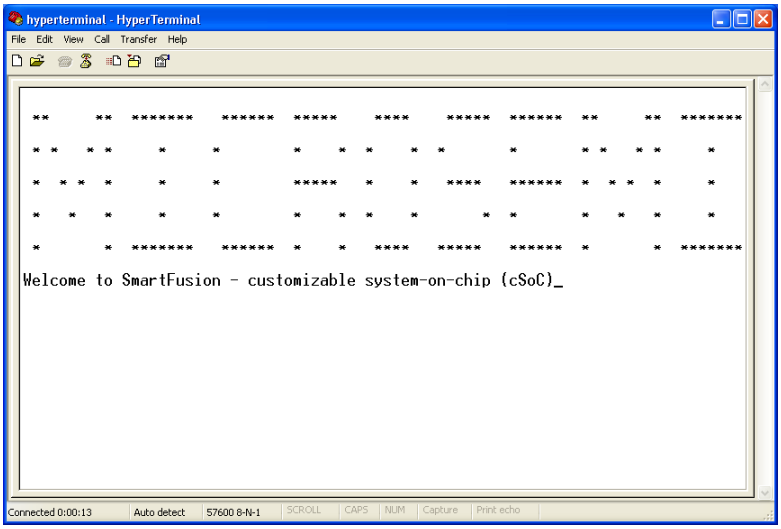

Figure 29 · HyperTerminal with Microsemi Name and Greeting Message

9. Suspend the software application by selecting **Run > Suspend** from the SoftConsole menu.

10. Click the **Registers** tab to view the values of the Cortex-M3 processor internal registers.

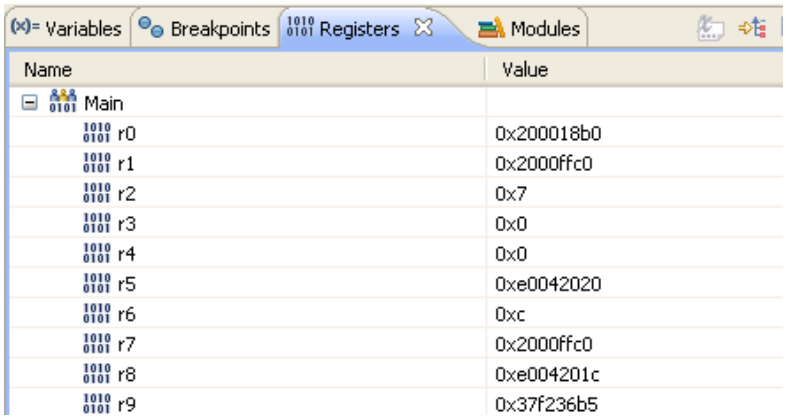

Figure 30 · Values of the Cortex-M3 Internal Registers

11. Click the **Variables** tab to view the values of variables in the source code.

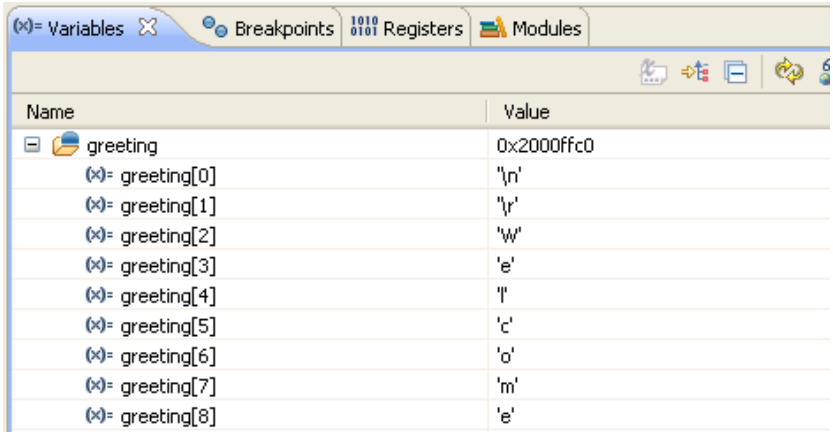

Figure 31 · Values of Variables in the Source Code

12. In the **Debug** window, select **Window > Show View > Disassembly** to display the assembly level instructions. The **Assembly** window is displayed on the right-side of the Debug perspective.

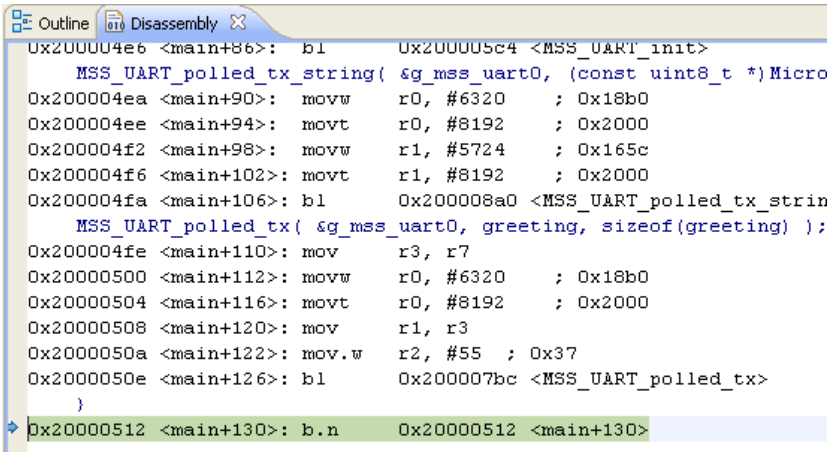

Figure 32 · Assembly Level Instructions

13. You can single-step through the source code by choosing **Run > Step Into** or **Run > Step Over** or by clicking **Step Into** or **Step Over** . Observe the changes in the source code window and **Disassembly** view. Performing a **Step Over** allows you an option for stepping over functions. The entire function is executed but there is no need to single-step through each instruction contained in the function.

14. Click **Instruction Stepping** and perform **Step Into** operations. Observe that **Step Into** executes a single line of assembly code.
15. Click **Instruction Stepping** to exit the instruction stepping mode. Single-step through the application and observe the instruction sequence in the source code window in the middle of the Debug perspective, and the values of the variables and registers.
16. Resume execution of the code by choosing **Run > Resume** or by clicking **Resume**.
17. Add breakpoints in the application. Enter commands in the HyperTerminal window to force the code to halt, then single-step and observe the instruction sequence.
18. Select **Embedded GDB** in the **Debug** tab and right-click and choose **Terminate and Remove** to stop the debugger when the process is finished.
19. Close HyperTerminal and SoftConsole using **File > Exit**.

This concludes the debugging process of this tutorial.

Step 10 – Debugging the Application Using the printf Statement in SoftConsole

Minor modifications need to be made in the above project (SmartFusion_UART) to debug by using the printf function. Refer to the steps below to use printf. You will create a new project to implement this function.

1. Follow the steps given below to create an application using the printf statement. Ensure that you close the previous SoftConsole project and also ensure that the name of the new project is different from the one used earlier (for example, SF_UART).
2. Invoke the SoftConsole project from Libero SoC.
3. Right-click the project **<project_name>_hw_platform**, and select **Properties**.
4. Click **Settings** under **C/C++ Build**.
5. Select **Symbols** under GNU Compiler. To add a symbol, click **Add** and enter **ACTEL_STDIO_THRU_UART** in the **Add Symbol** dialog box. Click **OK**.

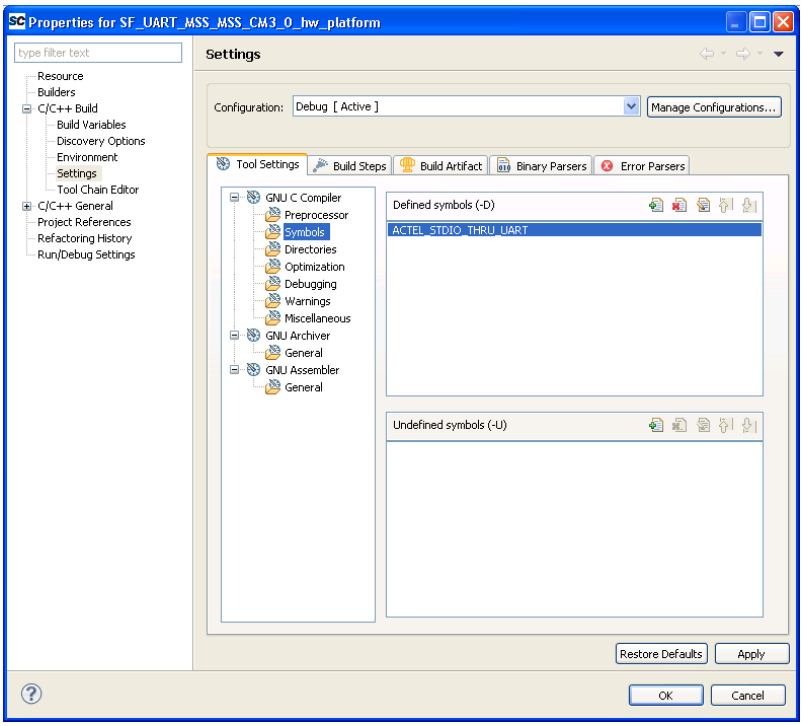

Figure 33 · Adding Symbol in GNU C Compiler

6. Click **Apply** to apply the changes that are made, and then click **OK** to close the **Properties** window.

7. Right-click the project **<project_name>_app** and select **Properties**. Select **Miscellaneous** under the **GNU C Linker** and enter the following in the **Linker flags** field:
-T*../../<project_name>_MSS_MSS_CM30_hw_platform/CMSIS/startup_gcc/debug-in-actel-smartfusion-envm.ld.*

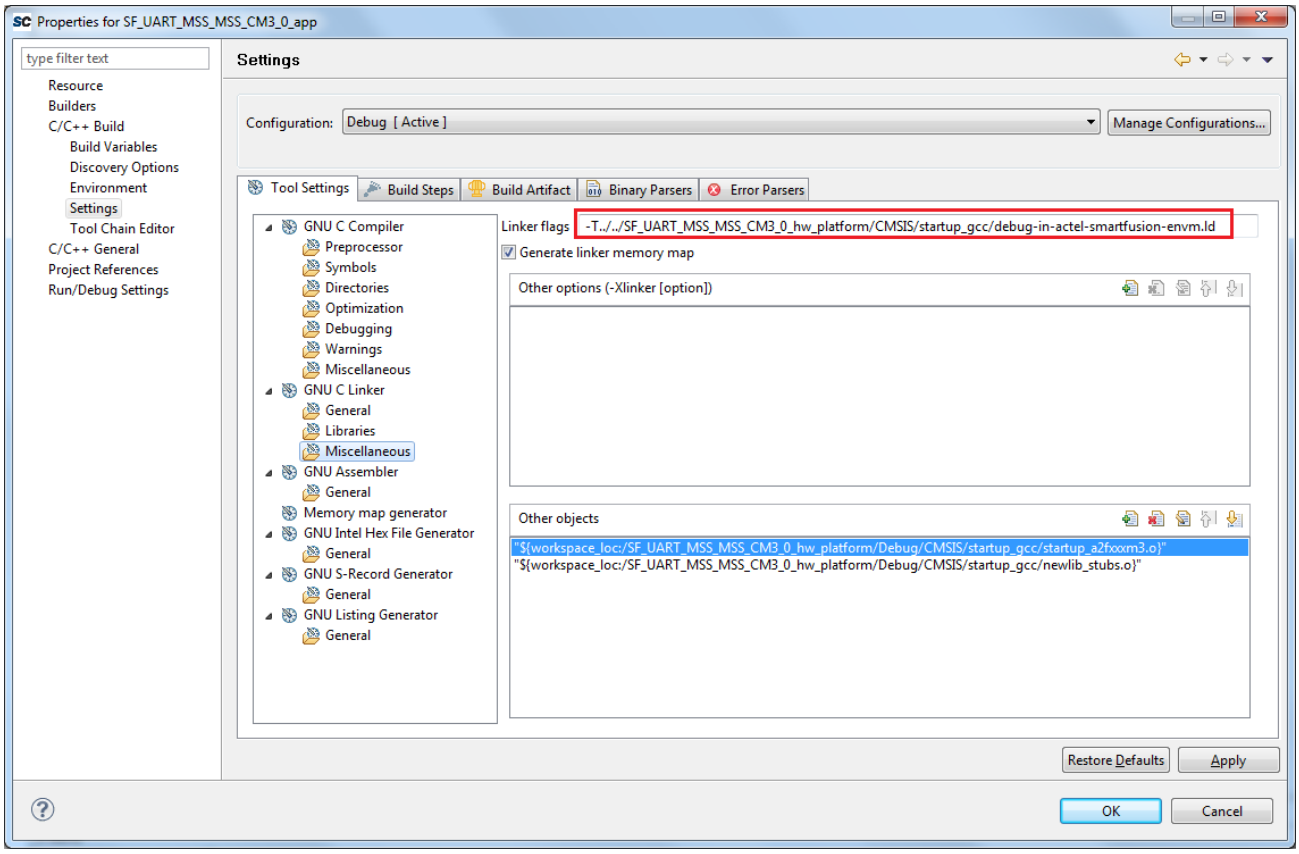

Figure 34 · Specifying Values for the Linker Flags Field

8. Click **Apply** to apply the changes that are made, and then click **OK** to close the **Properties** window.
9. Copy the code provided below and paste in the **main.c editor**.

```
#include "mss_uart.h"
#include <stdio.h>

/* Main function */
int main()
{
        int i = 0, j = 8;

        /* printing Microsemi logo on the serial terminal using "printf" */
        printf ("\n\r \
**      **  *******   ******  *****     ****     *****  ******  **      **  ******* \n\r \
* *    * *     *      *       *    *   *    *   *       *       * *    * *     *    \n\r \
*  * *  *     *      *       *****    *    *    ****   ******  *  * *  *     *    \n\r \
*    *   *     *      *       *    *   *    *        *  *       *    *   *     *    \n\r \
*        *  *******   ******  *    *    ****    *****   ******  *        *  ******* \n\r"

        );
```

```
        /* printing greeting message on the serial terminal using "printf" */
        printf ("Welcome to SmartFusion-customizable system-on-chip (cSoC)\n\r");

        /* printing 8 Table on UART */
        printf("Times 8 Table: \n\r");
        for(i = 0; i <= 10; i++)
        {
                printf("   %d x %d = %d\n\r", i, j, j*i);
        }

    while(1)
        {
        }
    return 0;
}
```

10. To build the project, select the project name in the **Project Explorer** tab and click **Project > Clean Project**. Enter the settings as shown in Figure 35 · and click **OK**.

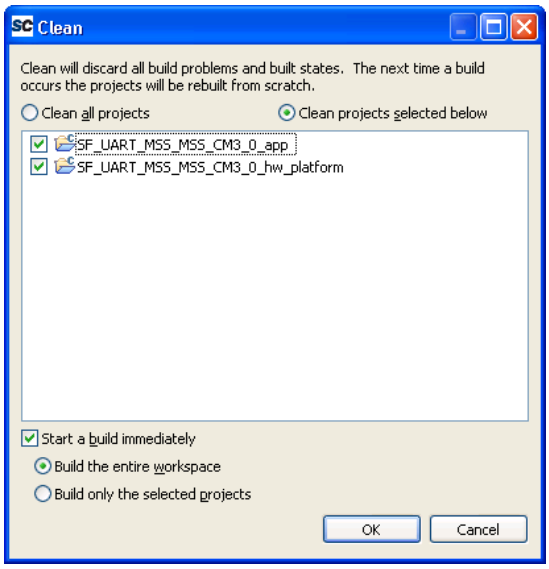

Figure 35 · Building a Clean Project

11. Ensure that there are no errors and warnings.
12. Select **Open Debug Dialog** or **Debug Configurations** from the **Run** menu. The Debug dialog box is displayed. Double-click **Microsemi Cortex-M3 eNVM target**. The window is displayed, as shown in Figure 36 · .

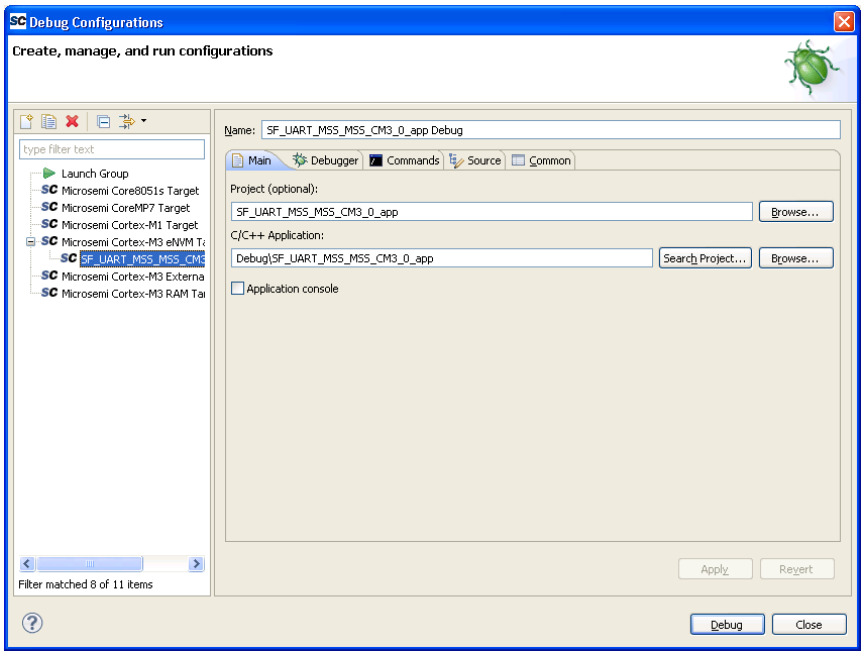

Figure 36 · Microsemi Cortex-M3 eNVM Target

13. Click **Debug** to debug the application.
14. Click **Yes** when prompted to **Confirm Perspective Switch**. Ensure that the USB cable is connected to the programmer and the device and also ensure that HyperTerminal is open and connected.
15. The SoftConsole Debug Perspective screen is displayed, as shown in Figure 37 · .

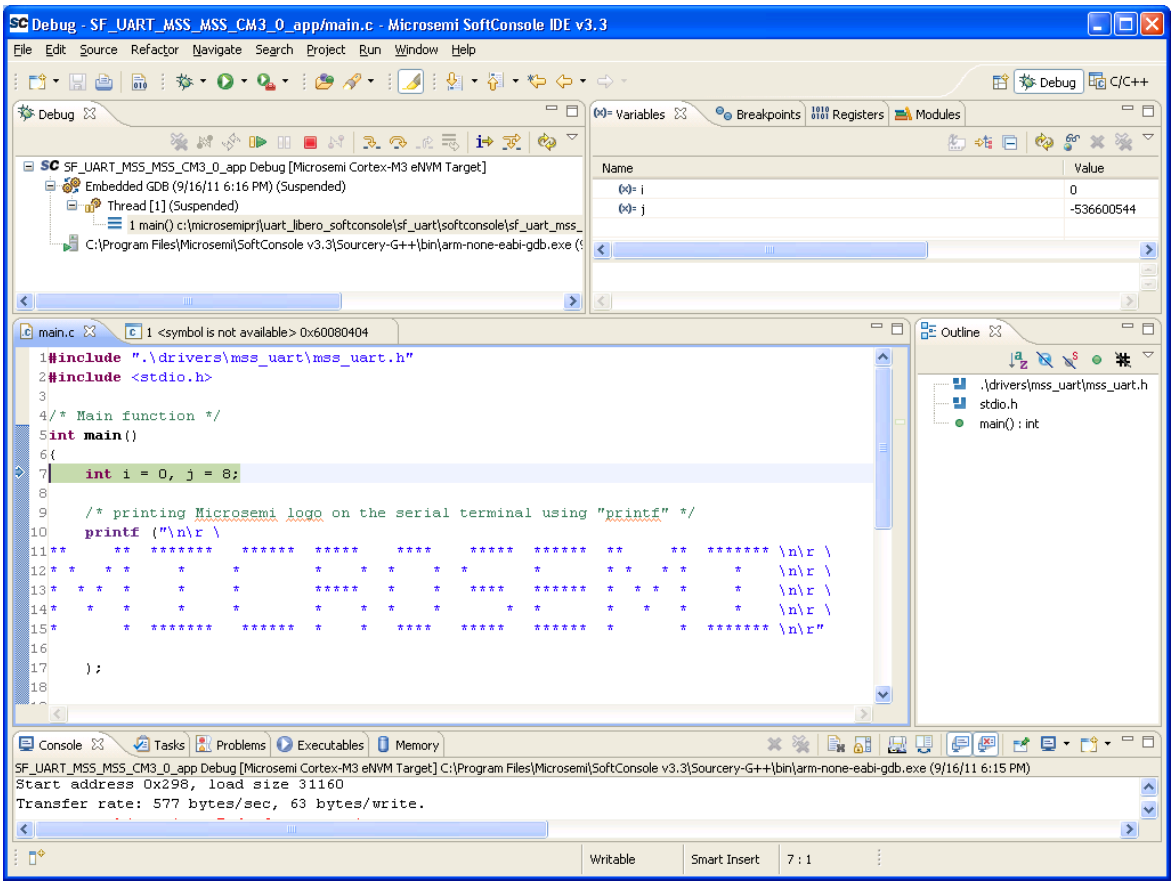

Figure 37 · The SoftConsole Debug Perspective

16. Click **Resume** or press **F8** key to resume debugging.

    Observe the HyperTerminal window; it should display a greeting message along with the multiplier table of 8.

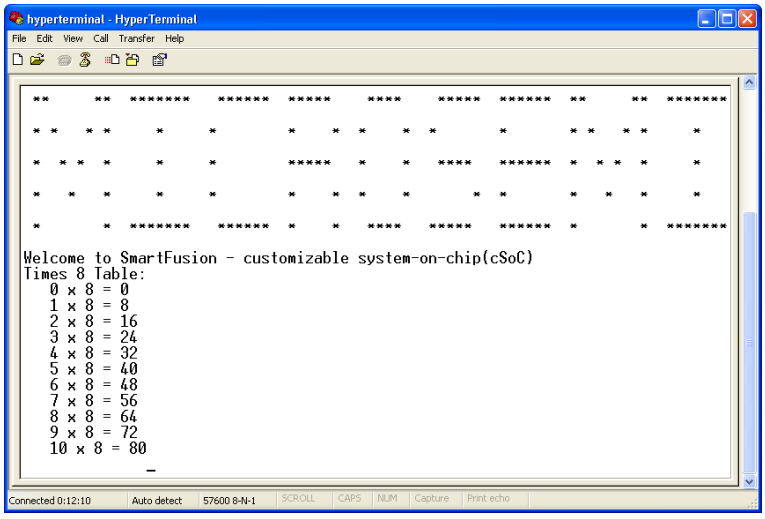

Figure 38 · The HyperTerminal Screen

Refer to Step 9 – Debugging the Application Project Using SoftConsole, for more information on debugging the application.

Step 11 – Building Executable Image in Release Mode

You can build an application executable image in release mode and load it into eNVM for executing code in the eNVM of the SmartFusion cSoC. You can load the application executable image into eNVM with the help of the eNVM data storage client from SmartDesign MSS Configurator and IAP or FlashPro programming software. In release mode, you cannot use SoftConsole debugger to load the executable image into eNVM. For more information on how to build an executable image for the application, refer to the *SmartFusion: Building Executable Image in Release Mode and Loading into eNVM Tutorial*.

This concludes the tutorial.

Appendix A – Libero SoC Catalog Settings

Listed below are the steps for configuring the vault location and setting up the repositories in Libero SoC.

1. On the **Catalog** window, click **Options**.

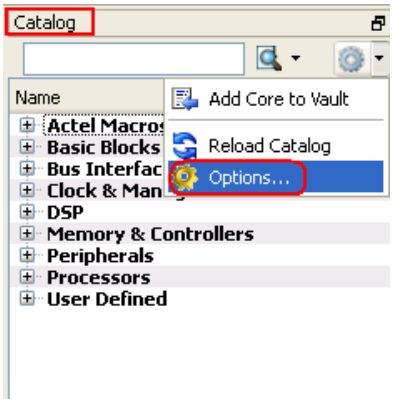

Figure 39 · Catalog – Options

2. The **Options** window is displayed. Click **Repositories** under **Vault/Repositories Settings** and add the following details in the address field.
   - www.actel-ip.com/repositories/SgCore
   - www.actel-ip.com/repositories/DirectCore
   - www.actel-ip.com/repositories/Firmware

Note: Click **Add** after entering each path.

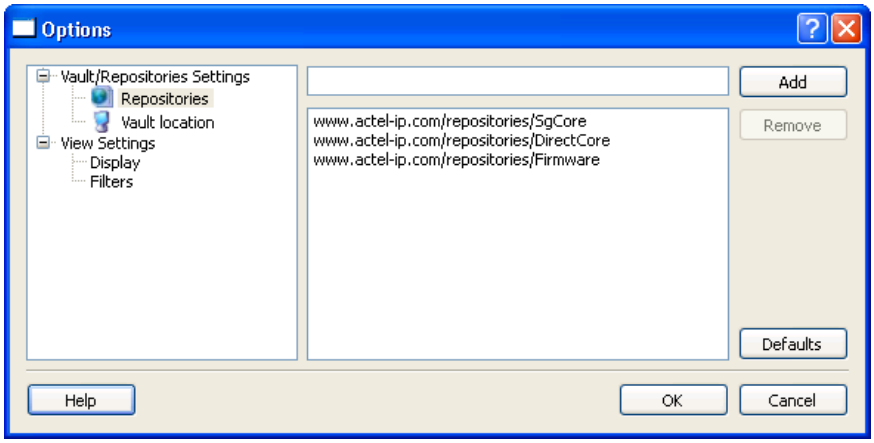

Figure 40 · Setting Repositories

3. Click **Vault location** under **Vault/Repositories Settings** in the **Options** window. Browse to a location on your PC to set the vault location where the IPs can be downloaded from the repositories.

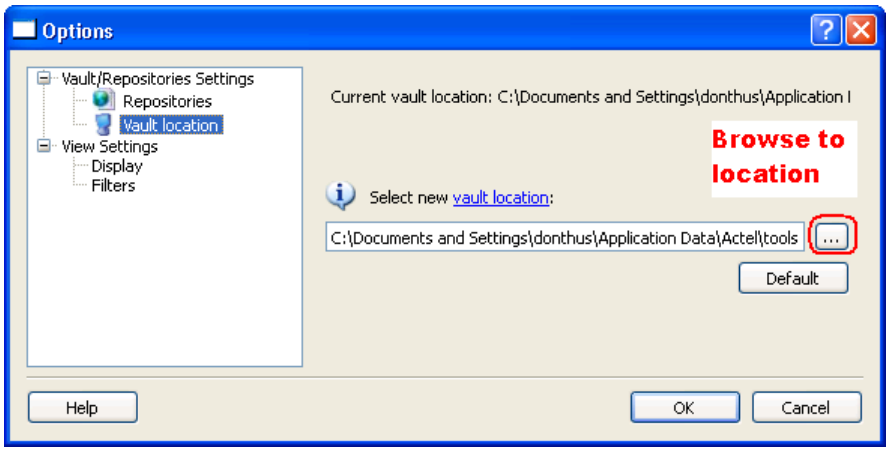

Figure 41 · Setting the Vault Location

4. Click **OK**.

Appendix B – Firmware Catalog Settings

1. Open the **<Libero Installation directory>\Designer\bin\catalog.exe**.
2. Select **Tools > Vault/Repositories Settings**, from the **Firmware Catalog** window.

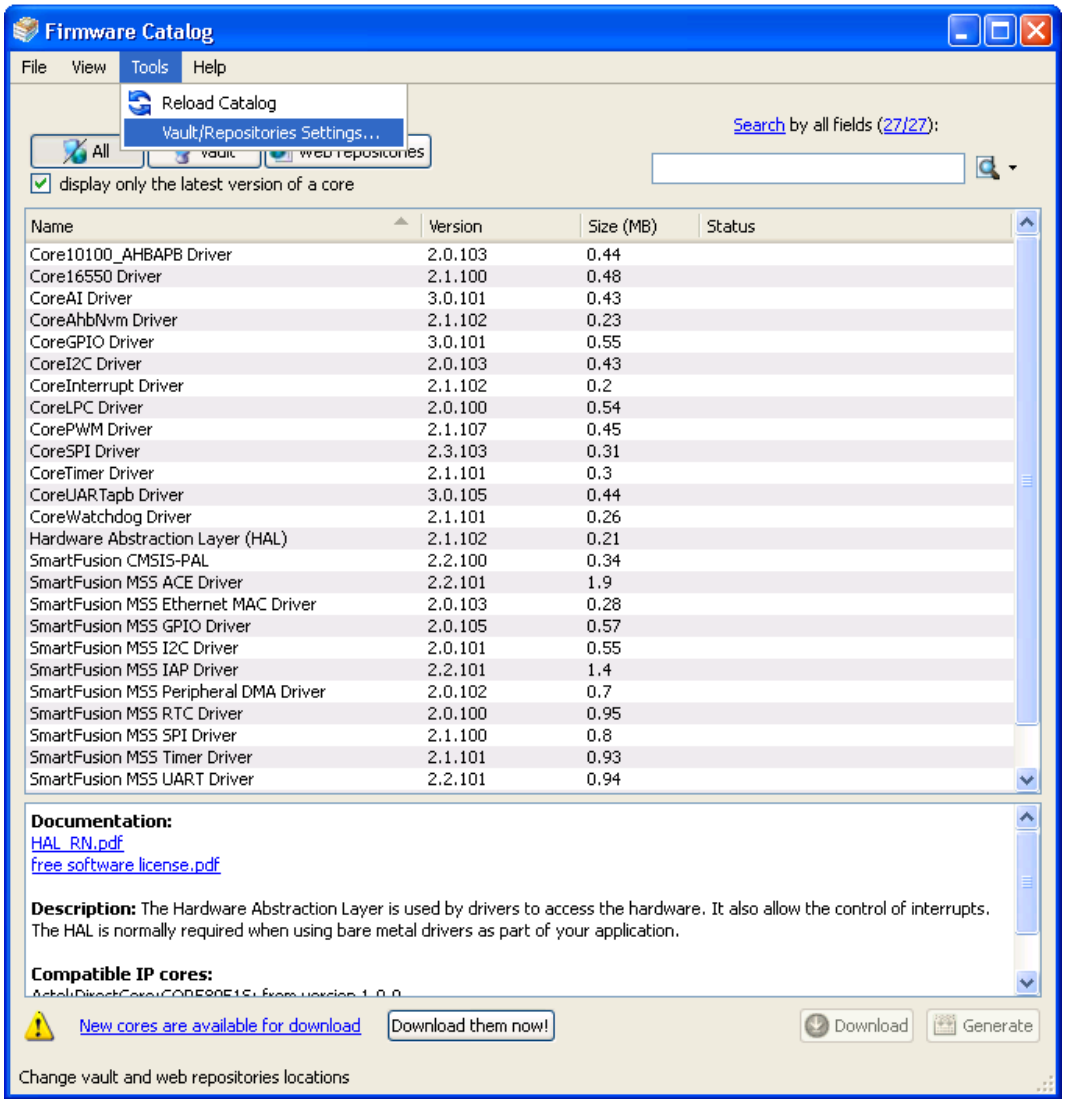

Figure 42 · Firmware Catalog Settings

3. Select **Repositories** under **Vault/Repositories Settings** in the **Options** dialog box.
4. Confirm that the following repositories are displayed (add them, if required):
   - www.actel-ip.com/repositories/SgCore
   - www.actel-ip.com/repositories/DirectCore
   - www.actel-ip.com/repositories/Firmware
5. Add the above paths in the address field, if required, by selecting the repository and clicking **Add**.

If new cores are available for download, click **Download them now!** to download the new cores to the vault.

List of Changes

Revision	Changes	Page
Revision 5 (April 2012)	Replaced Figure 13 · (SAR 38281)	10
	Modified Step 10 – Debugging the Application Using the printf Statement in SoftConsole (SAR 38281)	22
	Replaced Figure 34 · (SAR 38281)	23
	Modified Associated Project Files under Tutorial Requirements (SAR 38281)	4
	Modified note under Step 1 – Creating a Libero SoC Project (SAR 38281)	5
	Replaced Figure 15 · (SAR 38281)	11
	Replaced Figure 16 · and Figure 17 · (SAR 38281)	13
	Modified note under Step 6 – Building the Software Application Through SoftConsole (SAR 38281)	13
Revision 4 (January 2012)	Modified Step 2 – Configuring MSS Peripherals section. (SAR 36491)	7
	Removed Point 5 in Step 10 – Debugging the Application Using the printf Statement in SoftConsole section. (SAR 36491)	22
Revision 3 (November 2011)	Updated the document for Libero SoC v10.0 (SAR 35048).	N/A

Note: The revision number is located in the part number after the hyphen. The part number is displayed at the bottom of the last page of the document. The digits following the slash indicate the month and year of publication.

Product Support

Microsemi SoC Products Group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, electronic mail, and worldwide sales offices. This appendix contains information about contacting Microsemi SoC Products Group and using these support services.

Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From North America, call **800.262.1060**
From the rest of the world, call **650.318.4460**
Fax, from anywhere in the world **408.643.6913**

Customer Technical Support Center

Microsemi SoC Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions about Microsemi SoC Products. The Customer Technical Support Center spends a great deal of time creating application notes, answers to common design cycle questions, documentation of known issues and various FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Technical Support

Visit the Microsemi SoC Products Group Customer Support website for more information and support (http://www.microsemi.com/soc/support/search/default.aspx). Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on website.

Website

You can browse a variety of technical and non-technical information on the Microsemi SoC Products Group home page, at http://www.microsemi.com/soc/.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center. The Technical Support Center can be contacted by email or through the Microsemi SoC Products Group website.

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is soc_tech@microsemi.com.

My Cases

Microsemi SoC Products Group customers may submit and track technical cases online by going to My Cases.

Outside the U.S.

Customers needing assistance outside the US time zones can either contact technical support via email (soc_tech@microsemi.com) or contact a local sales office. Sales office listings can be found at www.microsemi.com/soc/company/contact/default.aspx.

ITAR Technical Support

For technical support on RH and RT FPGAs that are regulated by International Traffic in Arms Regulations (ITAR), contact us via soc_tech_itar@microsemi.com. Alternatively, within My Cases, select **Yes** in the ITAR drop-down list. For a complete list of ITAR-regulated Microsemi FPGAs, visit the ITAR web page.

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo CA 92656 USA
Within the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

Microsemi Corporation (NASDAQ: MSCC) offers a comprehensive portfolio of semiconductor solutions for: aerospace, defense and security; enterprise and communications; and industrial and alternative energy markets. Products include high-performance, high-reliability analog and RF devices, mixed signal and RF integrated circuits, customizable SoCs, FPGAs, and complete subsystems. Microsemi is headquartered in Aliso Viejo, Calif. Learn more at **www.microsemi.com**.

© 2012 Microsemi Corporation. All rights reserved. Microsemi and the Microsemi logo are trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.

50200216-5/04.12