Accessing Serial Flash Memory Using SPI Interface

Libero SoC v11.5 and SoftConsole Flow Tutorial for SmartFusion2 SoC FPGA TU0546

Microsemi®

Table of Contents

Accessing Serial Flash Memory using SPI Interface- Libero SoC v11.5

Introduction

The Libero® System-on-Chip (SoC) software generates firmware projects using SoftConsole, IAR, and Keil tools. This tutorial describes the process to build a SoftConsole application that can be implemented and validated using the SmartFusion®2 SoC field programmable gate array (FPGA) Security Evaluation Kit.

The same firmware project can be built using IAR and Keil tools. Refer to the respective tutorials:

- *Accessing Serial Flash Memory using SPI Interface - Libero SoC and IAR Embedded Workbench Flow Tutorial for SmartFusion2 SoC FPGA*
- *Accessing Serial Flash Memory Using SPI Interface - Libero SoC and Keil uVision Flow Tutorial for SmartFusion2 SoC FPGA*

After completing this tutorial, you will be able to perform the following tasks:

- Create a Libero SoC project using System Builder
- Generate the programming file to program the SmartFusion2 device
- Open the project in SoftConsole from Libero SoC
- Compile application code
- Debug and run code using SoftConsole

Design Requirements

Table 1 • **Design Requirements**

Design Requirements	Description
Hardware Requirements	
SmartFusion2 Security Evaluation Kit • FlashPro4 programmer • USB A to Mini-B cable • 12 V adapter	Rev D or later
Host PC or Laptop	Any 64-bit Windows Operating System
Software Requirements	
Libero SoC	v11.5
SoftConsole	v3.4SP1
FlashPro programming software	v11.5
USB to UART drivers	-
One of the following serial terminal emulation programs: • HyperTerminal • TeraTerm • PuTTY	-

Associated Project Files

Download the associated project files for this tutorial from the Microsemi® website:
http://soc.microsemi.com/download/rsc/?f=m2s_tu0546_liberov11p5_df

The demo design files include:

- LiberoProject
- Programmingfile
- Source Files
- SPI_Flash_Drivers
- Readme file

Refer to the Readme.txt file provided in the design files for the complete directory structure.

Target Board

SmartFusion2 Security Evaluation Kit board (SF2_EVAL_KIT) Rev D (or later).

Design Overview

This design example demonstrates the execution of basic read and write operations on the SPI flash present on the SmartFusion2 Security Evaluation Kit board. This kit has a built-in winbond SPI flash memory W25Q64FVSSIG, which is connected to the SmartFusion2 microcontroller subsystem (MSS) through dedicated MSS SPI_0 interface.

Read and write data information is displayed using HyperTerminal which communicates to the SmartFusion2 MSS using the MMUART_1 interface.

For more information on SPI, refer to the *SmartFusion2 Microcontroller Subsystem User Guide*.

Figure 1 shows interfacing the external SPI flash to MSS SPI_0.

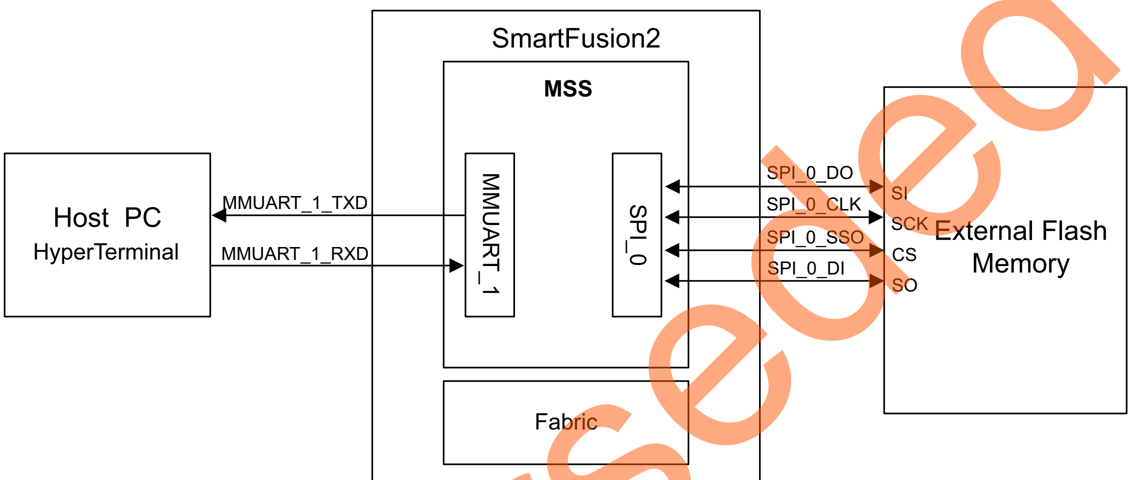


***Figure 1* • SPI Flash Interfacing Block Diagram**

Step 1: Creating a Libero SoC Project

The following steps describe how to create a Libero SoC project:

Launching Libero SoC

1. Click **Start > Programs > Microsemi Libero SoC v11.5 > Libero SoC v11.5**, or click the shortcut on desktop to open the Libero SoC v11.5 Project Manager.
2. Create a new project by selecting **New** on the **Start Page** tab (highlighted in Figure 2), or by clicking **Project > New Project** from the Libero SoC menu.

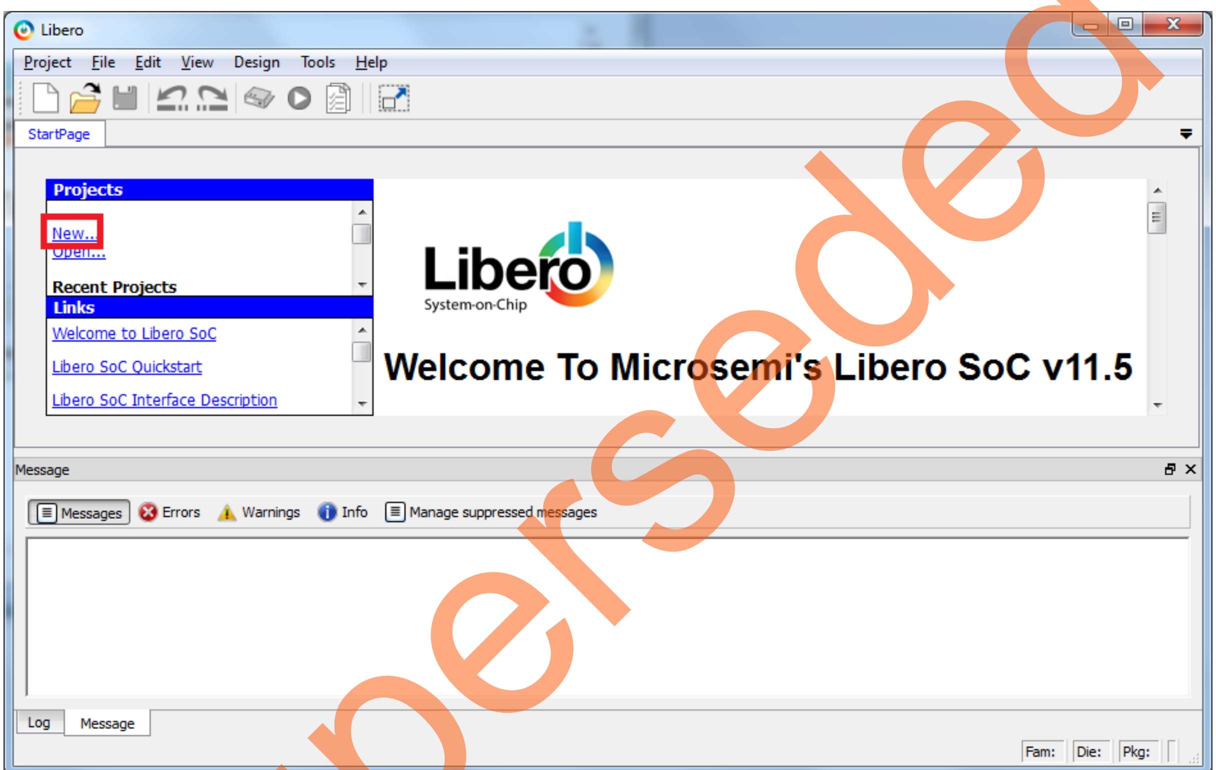


Figure 2 • **Libero SoC Project Manager**

3. In the **Project Details** window, enter the information as displayed in Figure 3.
   – Project Name: SPI_Flash
   – Project Location: Select an appropriate location (for example, D:/Microsemi_prj)
   – Preferred HDL type: Verilog
   – Enable Block Creation: Unchecked

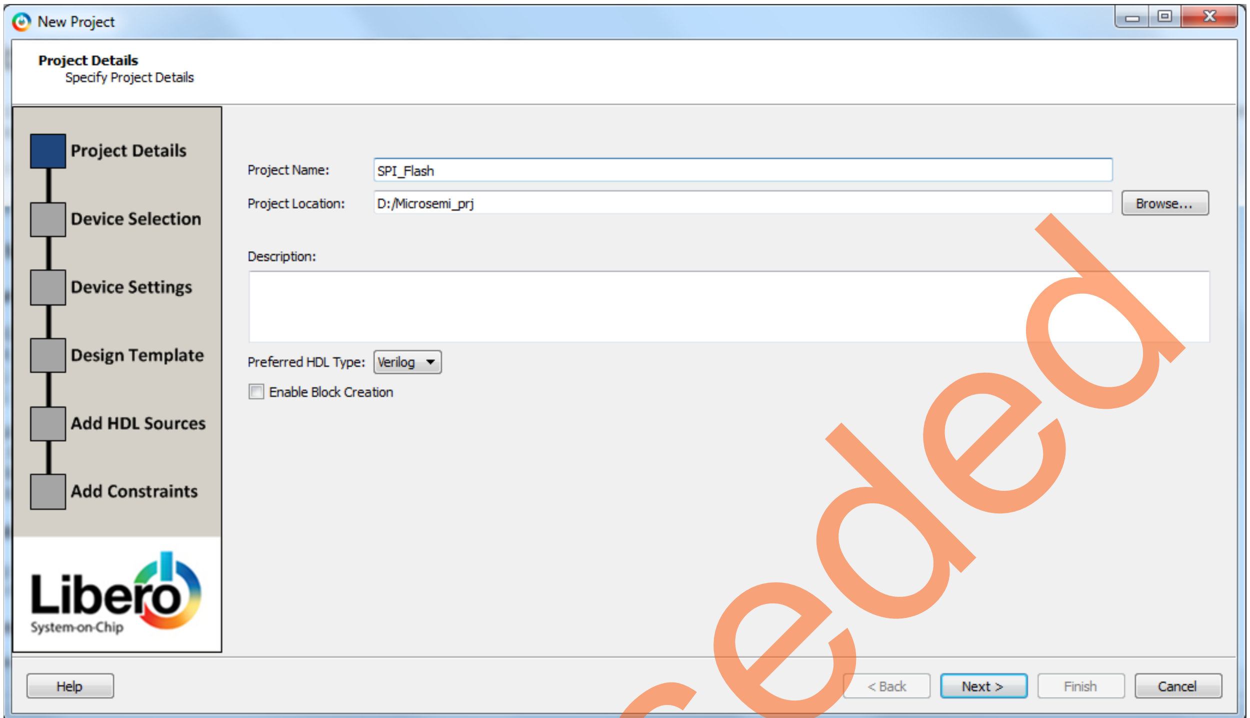


***Figure 3* • Project Details Window**

4. Click **Next**. In the **Device Selection** window, select the information displayed in Figure 4. In the Part Filter (select the following values using the drop-down list)
   – Family: SmartFusion2
   – Die: M2S090TS
   – Package: 484 FBGA
   – Speed: -1
   – Core Voltage: 1.2
   – Range COM

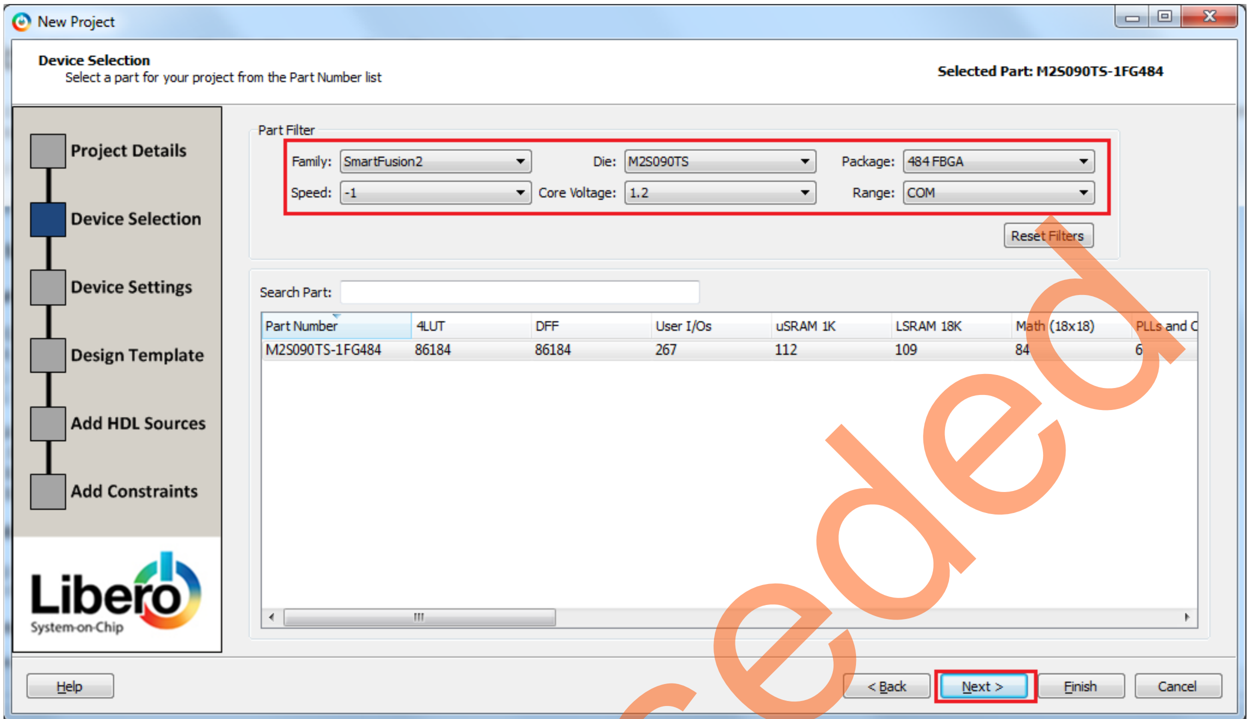


Figure 4 **• Device Selection Window**

5. Click **Next**. The **Device Settings** window is displayed. Retain the default values.
6. Click **Next**. In the **Design Template** page, select the select **Create a System Builder base design** under the **Design Templates and Creators**.

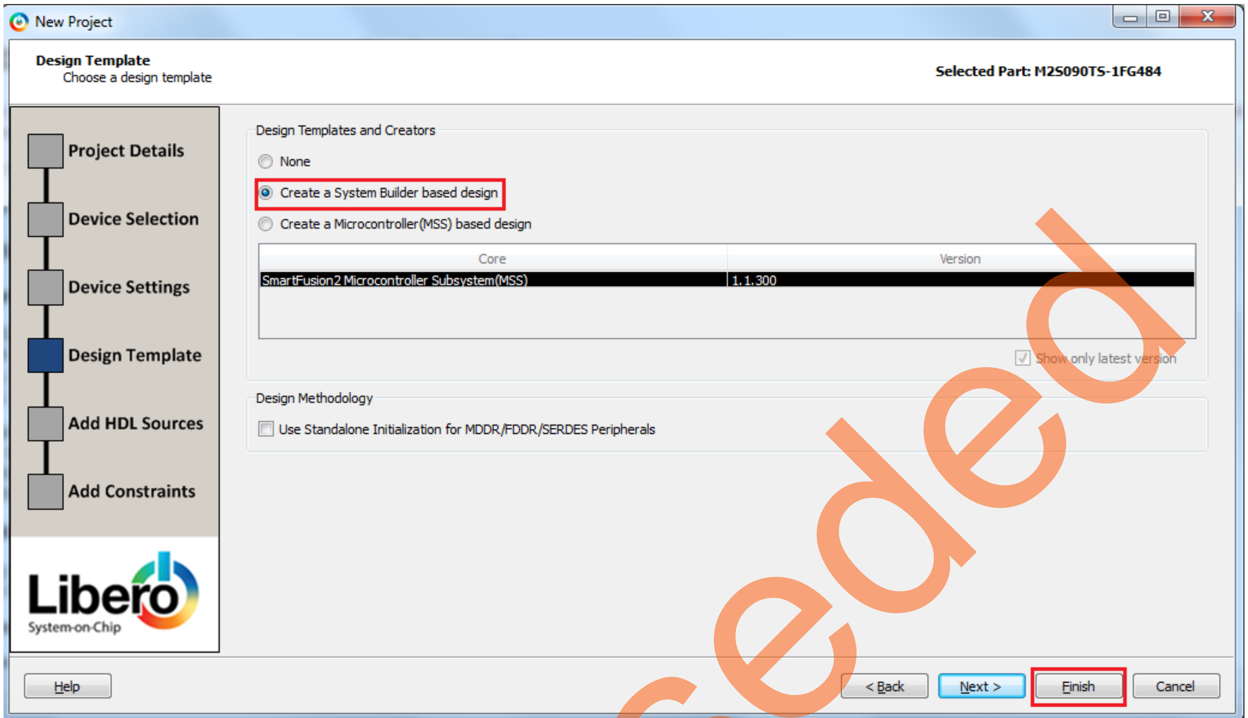


***Figure 5* • Design Template Window**

7. Click **Finish**. A System Builder dialog box is displayed.

Note: System Builder is a graphical design wizard. It creates a design based on high-level design specifications by taking the user through a set of high-level questions that will define the intended system.

8. Enter **SPI_Flash** as the name of the system and click **OK**, as shown in Figure 6.

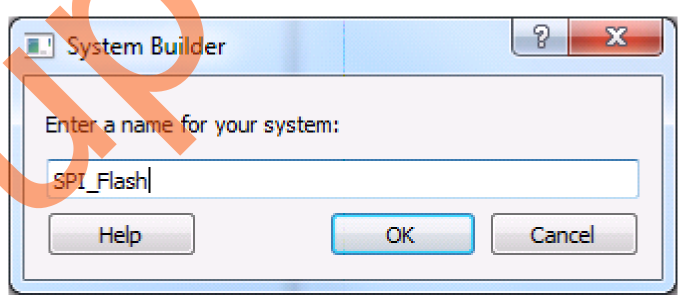


***Figure 6* • System Builder Dialog Box**

9. **System Builder – Device Features** page is displayed, as shown in Figure 7.

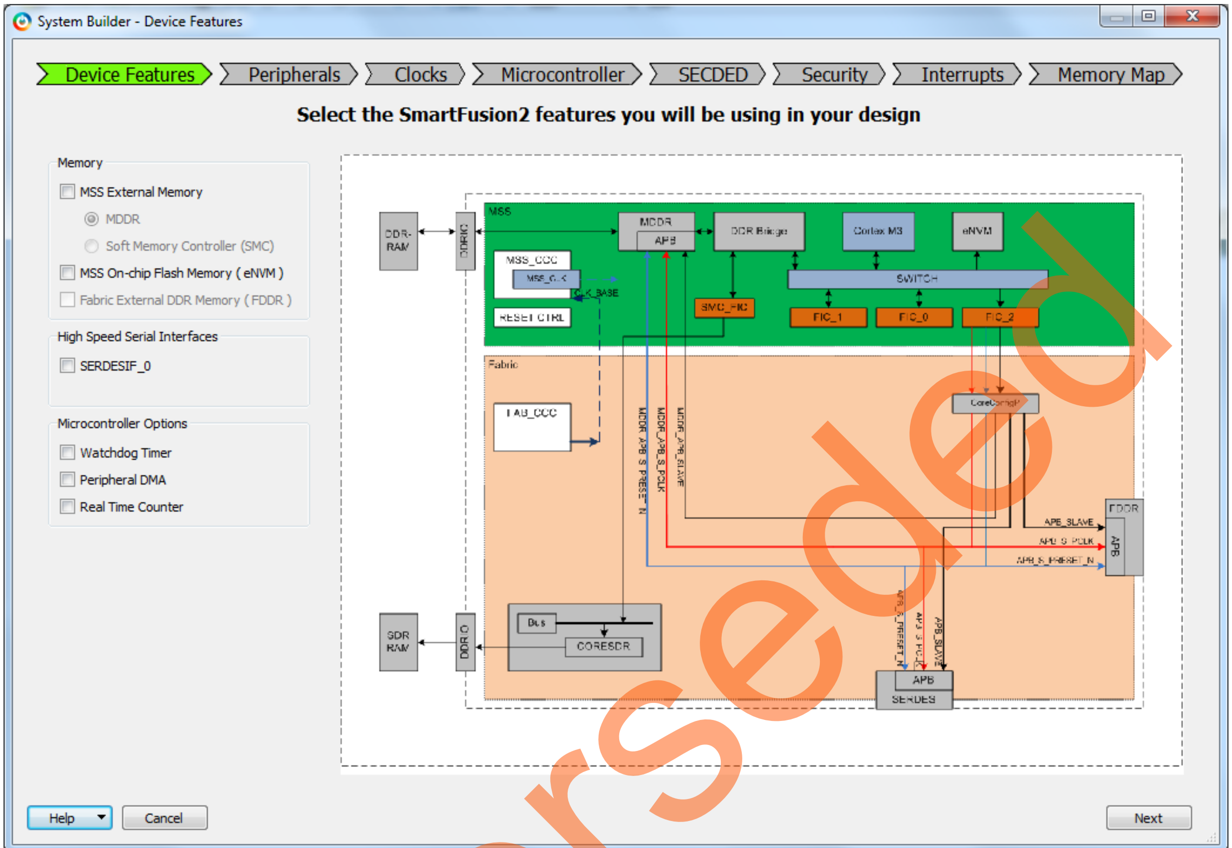


Figure 7 **• System Builder – Device Features Page**

10. Click **Next**, the **System Builder – Peripherals** page is displayed. Under the MSS Peripherals section, Clear all the check boxes except **MM_UART_1** and **MSS_SPI_0**, as shown in Figure 8.

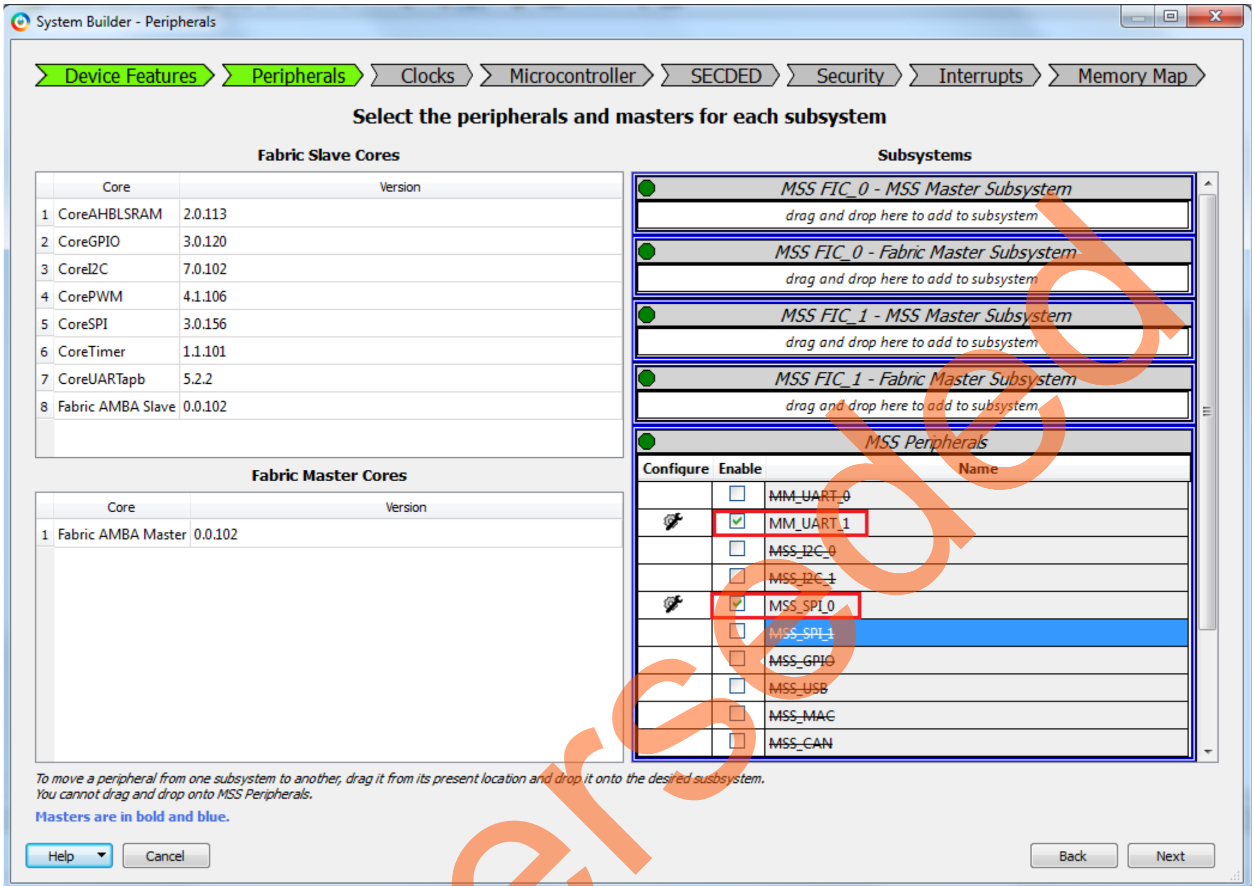


Figure 8 • **System Builder Configurator – Select Peripherals Page**

11. Click **Next**, the **System Builder – ClocksSettings** page is displayed, as shown in Figure 9. Select **System Clock** source as **On-chip 25/50 MHz RC Oscillator**. The M3_CLK is configured to 100 MHz by default.

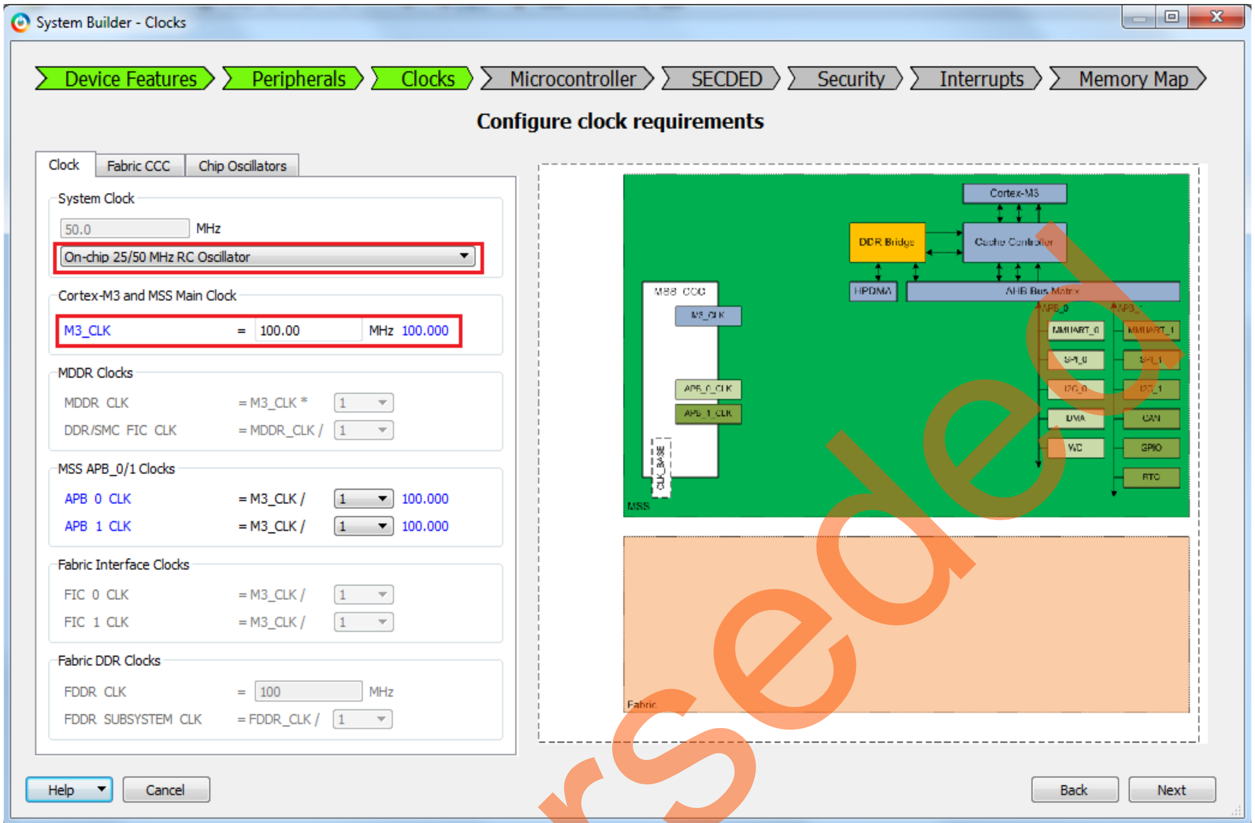


Figure 9 • **System Builder Configurator – Clock Settings Page**

12. Click **Next**, the **System Builder – Microcontroller Options** page is displayed.
    – Retain the default values.
13. Click **Next**, the **System Builder – SECDED Options** page is displayed.
    – Retain the default values.
14. Click **Next**, the **System Builder – Security Options** page is displayed.
    – Retain the default values.
15. Click **Next**, the **System Builder – InterruptsOptions** page is displayed.
    – Retain the default values.
16. Click **Next**, the **System Builder – Memory MapOptions** page is displayed.
    – Retain the default values.
17. Click **Finish**.

    The **System Builder** generates the system based on the selected options.

The System Builder block is created and added to the Libero SoC project, as shown in Figure 10.

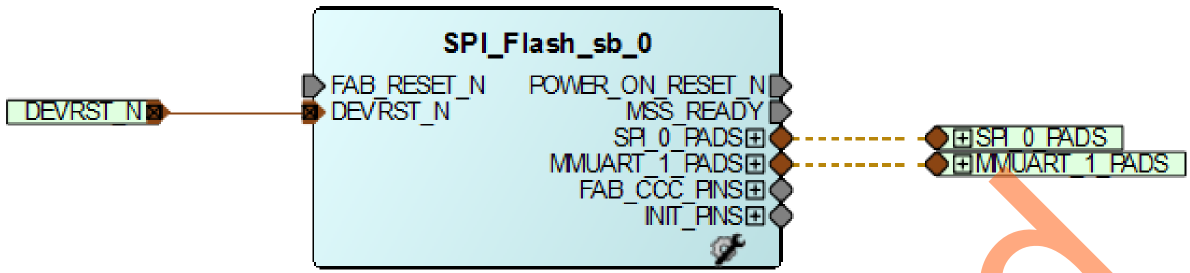


Figure 10 • **System Builder Generated System**

Connecting Components in SPI_Flash SmartDesign

1. Connect the pins as follows:
   – Tie the **FAB_RESET_N** to high by right-clicking and selecting **Tie High**.
   – Mark the output port **POWER_ON_RESET_N** as unused by right-clicking and selecting **Mark Unused**.
   – Mark the output port **MSS_READY** as unused by right-clicking and selecting **Mark Unused**.
   – Expand **INIT_PINS**, right-click **INIT_DONE** and select **Mark Unused**.
   – Expand **FAB_CCC_PINS**, right-click **FAB_CCC_GL0** and select **Mark Unused**.
   – Right-click FAB_CCC_LOCK and select Mark Unused.
2. Click **File** > **Save**. The SPI_Flash design is displayed as shown in Figure 11.

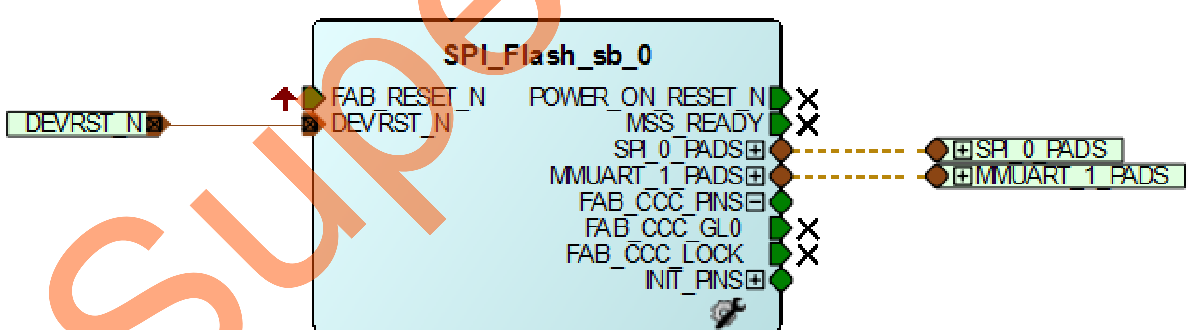


Figure 11 • **SPI_Flash Design**

3. Generate the SPI_Flash Smart Design by clicking SmartDesign > Generate Component or by clicking Generate Component on the SmartDesign toolbar.

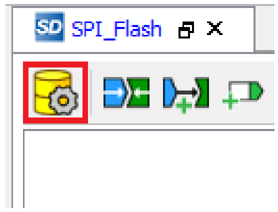


***Figure 12* • Generate Component**

After successful generation of the system, the message **'info: SPI_Flash' was successfully generated** is displayed on the Libero SoC **Log** window as shown in Figure 13.

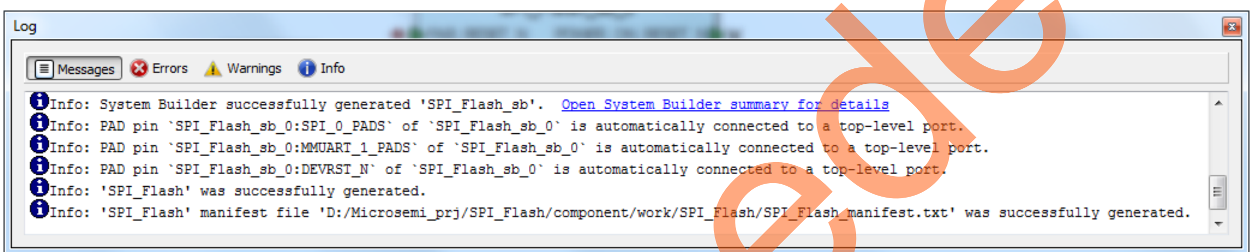


***Figure 13* • Log Window**

Step 2: Generating the Program File

Click **Generate Bitstream** as shown in Figure 14 to generate the programming file.

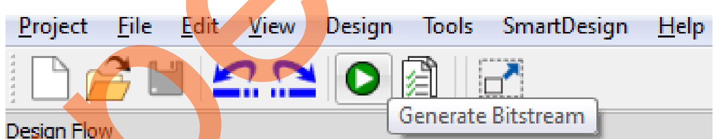


***Figure 14* • Generate Bitstream**

Step 3: Programming the SmartFusion2 Board Using FlashPro

1. Connect the FlashPro4 programmer to the J5 connector of the SmartFusion2 Security Evaluation Kit.
2. Connect the jumpers on the SmartFusion2 Security Evaluation Kit board as listed in Table 2 on page 15. For more information on jumper locations, refer Appendix B - SmartFusion2 Security Evaluation Kit Board Jumper Locations.

CAUTION: While making the jumper connections, the **SW7** power supply switch on the board must be in **OFF** position.

Table 2 • **SmartFusion2** Security **Evaluation Kit Jumper Settings**

Jumper Number	Pin (from)	Pin (to)	Comments
J22, J23, J24,J8, J3	1	2	These are the default jumper settings of the SmartFusion2 Security Evaluation Kit board. Make sure these jumpers are set properly.

3. Connect the power supply to the J6 connector.
4. Switch **ON** the SW7 power supply switch.
   Refer to Appendix A - Board Setup for Running the Tutorial for information on board setup for running the tutorial.
5. To program the SmartFusion2 device, double-click **Run PROGRAM Action** in the **Design Flow** window as shown in Figure 15.

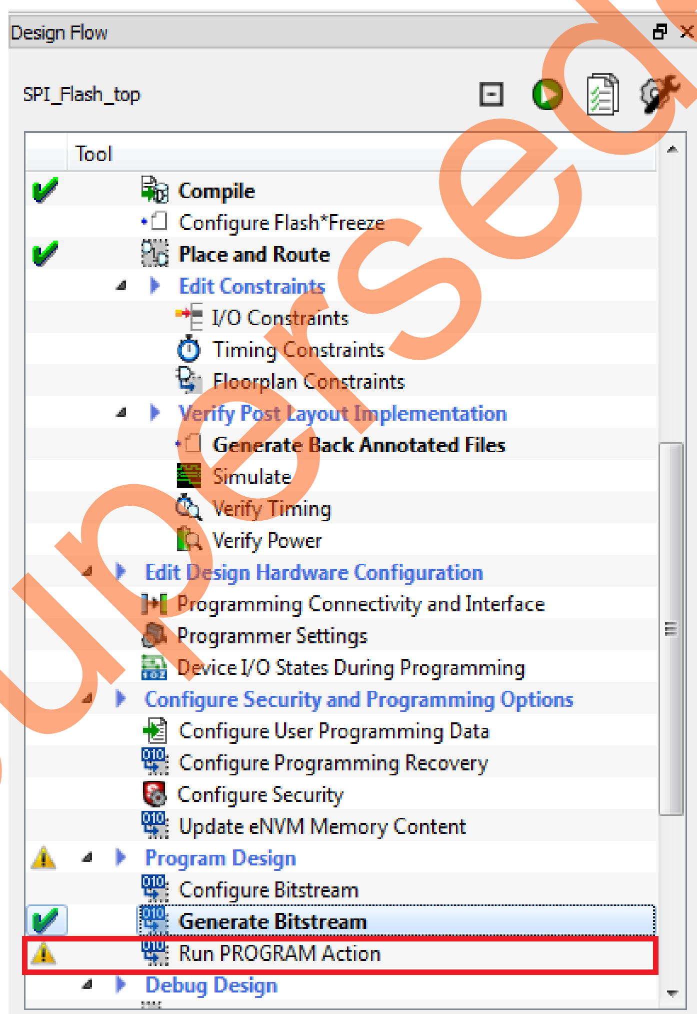


Figure 15 • **Run Programming Action**

Step 4: Configuring and Generating Firmware

The Design Firmware window displays compatible firmware drivers based on peripherals configured in the design. Following drivers are used in this tutorial:

- CMSIS
- MMUART
- SPI

To generate the required drivers,

1. **Double-click Configure Firmware Cores in Handoff design for Firmware Development in design flow window.** Clear all the drivers' check boxes, except CMSIS, MMUART, and SPI as shown in Figure 16.

   Note:Select the latest version of the drivers.

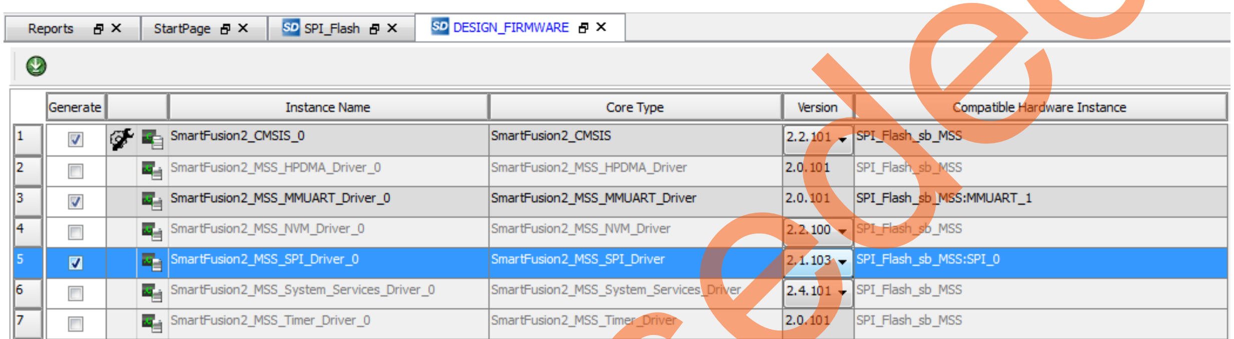


Figure 16 • **Configuring Firmware**

2. Double-click on Export Firmware in Handoff design for Firmware Development in design flow window.
3. Export Firmware dialog box is displayed as shown in Figure 17.

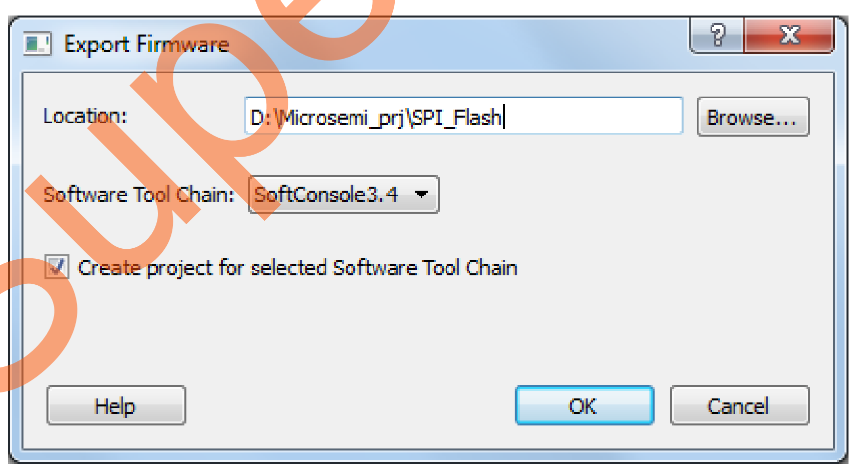


Figure 17 • **Export Firmware Dialog**

4. Select **SoftConsole3.4** from the drop down list.
5. Select **Create project for selected Software Tool Chain**.
6. Click **OK**. An information message like "Firmware project was successfully exported to <drive:\>Microsemi_prj\SPI_Flash" is displayed.

.

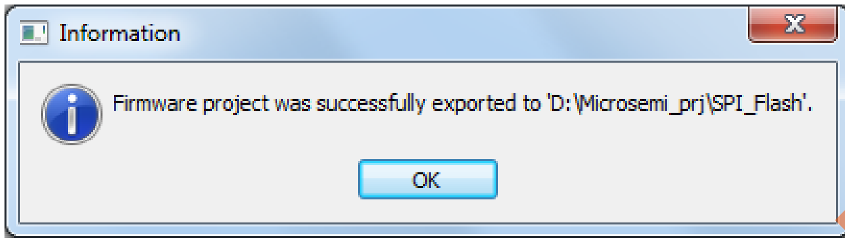


Figure 18 • **Firmware Project Confirmation Dialog**

7. Click **OK**.

Step 5: Building the Software Application using SoftConsole

1. Click **Start > Programs > Microsemi SoftConsole v3.4 > Microsemi SoftConsole v3.4.0.5** to open SoftConsole IDE.
2. SoftConsole **Workspace Launcher** window is displayed. Browse to the SoftConsole Project in the Libero Project folder as shown in Figure 19.

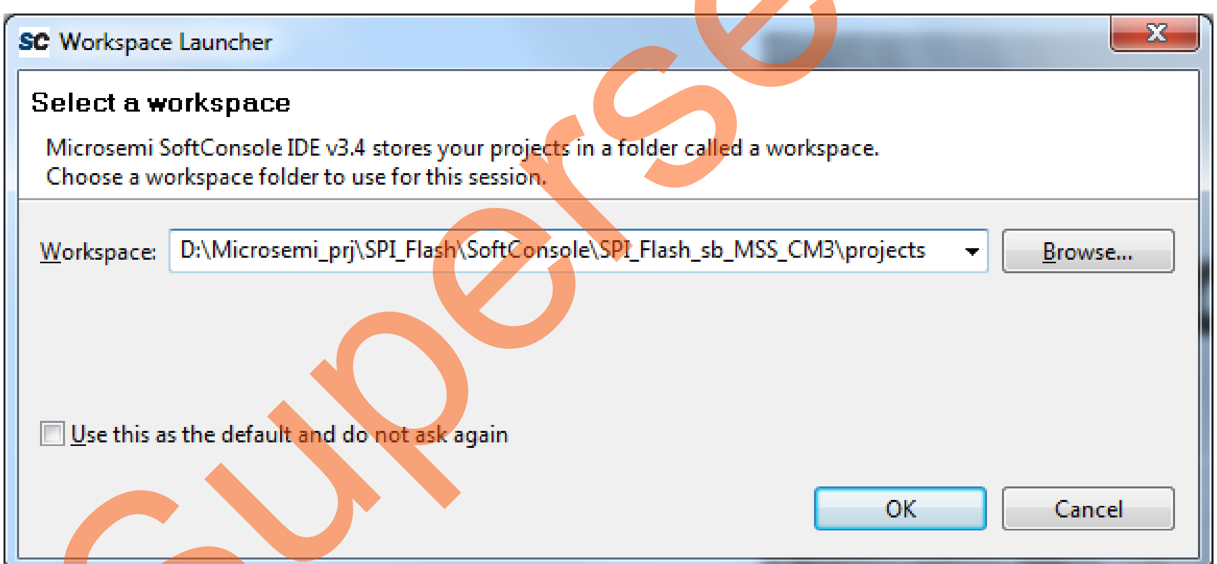


Figure 19 • **Invoking SoftConsole**

The SoftConsole workspace is displayed, as shown in Figure 20.

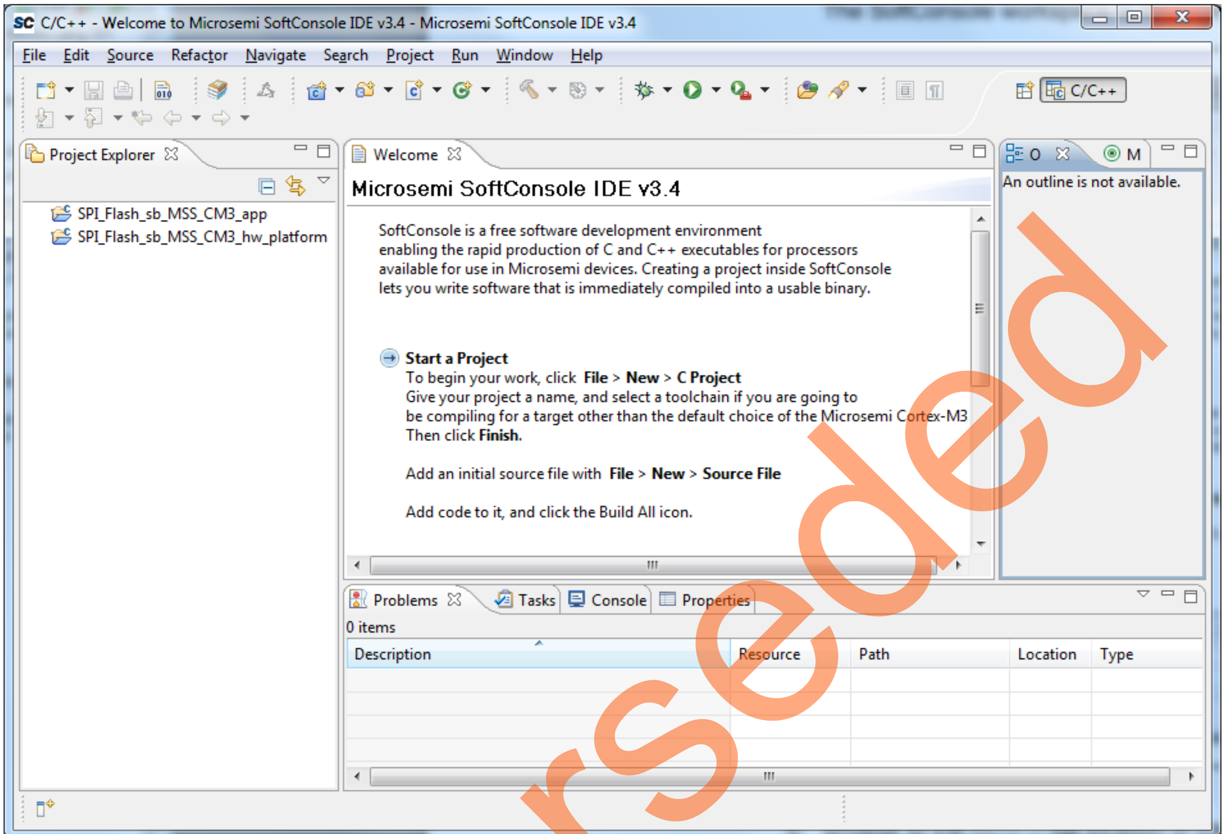


***Figure 20* • SoftConsole Workspace**

3. Browse to the `main.c` file location in the design files folder: *<download_folder>\SF2_SPI_Flash_SC_Tutorial_DF\Source Files*.
4. Copy the `main.c` file and replace the existing `main.c` file under SPI_Flash_sb_MSS_CM3_app project in the SoftConsole workspace.

The SoftConsole window displays the `main.c` file, as shown in Figure 21.

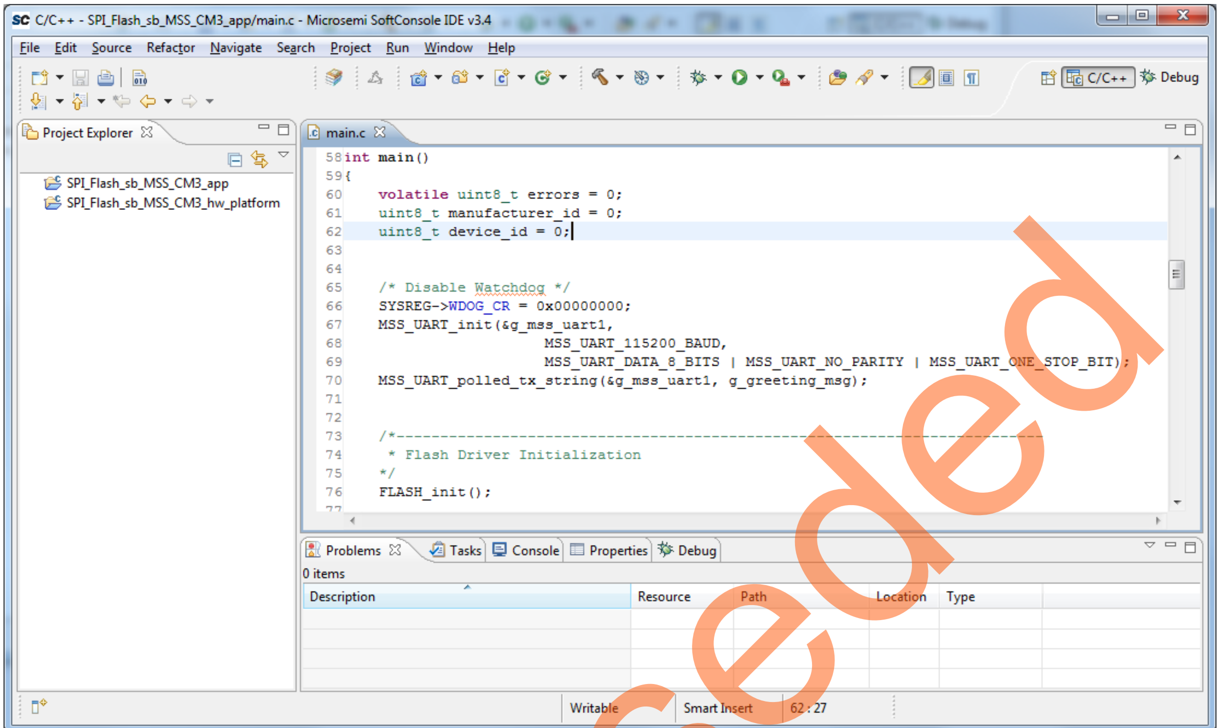


***Figure 21* • SoftConsole Workspace main.c file**

5. winbondflash SPI flash drivers are not included in the Libero generated SoftConsole workspace. To include the drivers in the SoftConsole workspace, browse to the location of the winbondflash drivers in the design files folder: *<download_folder>\SF2_SPI_Flash_SC_Tutorial_DF\SPI_Flash_Drivers*.

6. Copy the **winbondflash** folder to the drivers folder of SPI_Flash_sb_MSS_CM3_hw_platform project in the SoftConsole workspace, as shown in Figure 22.

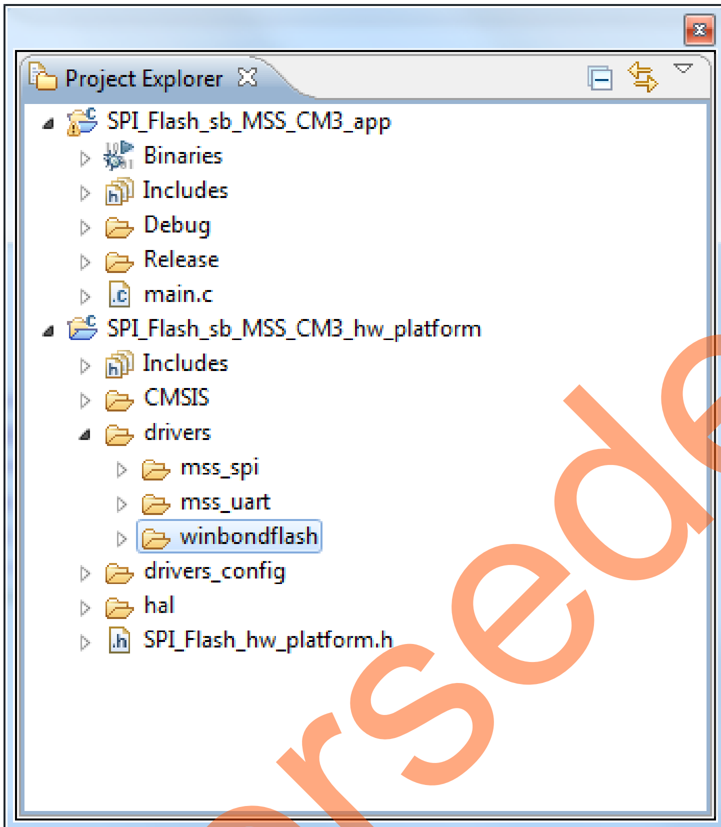


Figure 22 **• Project Explorer window**

This tutorial uses `printf` statements to display memory read data. Redirection of the output of `printf()` to a UART is enabled by adding the **MICROSEMI_STDIO_THRU_UART** symbol.

7. Right-click the **SPI_Flash_sb_MSS_CM3_hw_platform** in Project Explorer window of SoftConsole project and select **Properties** as shown in Figure 23.

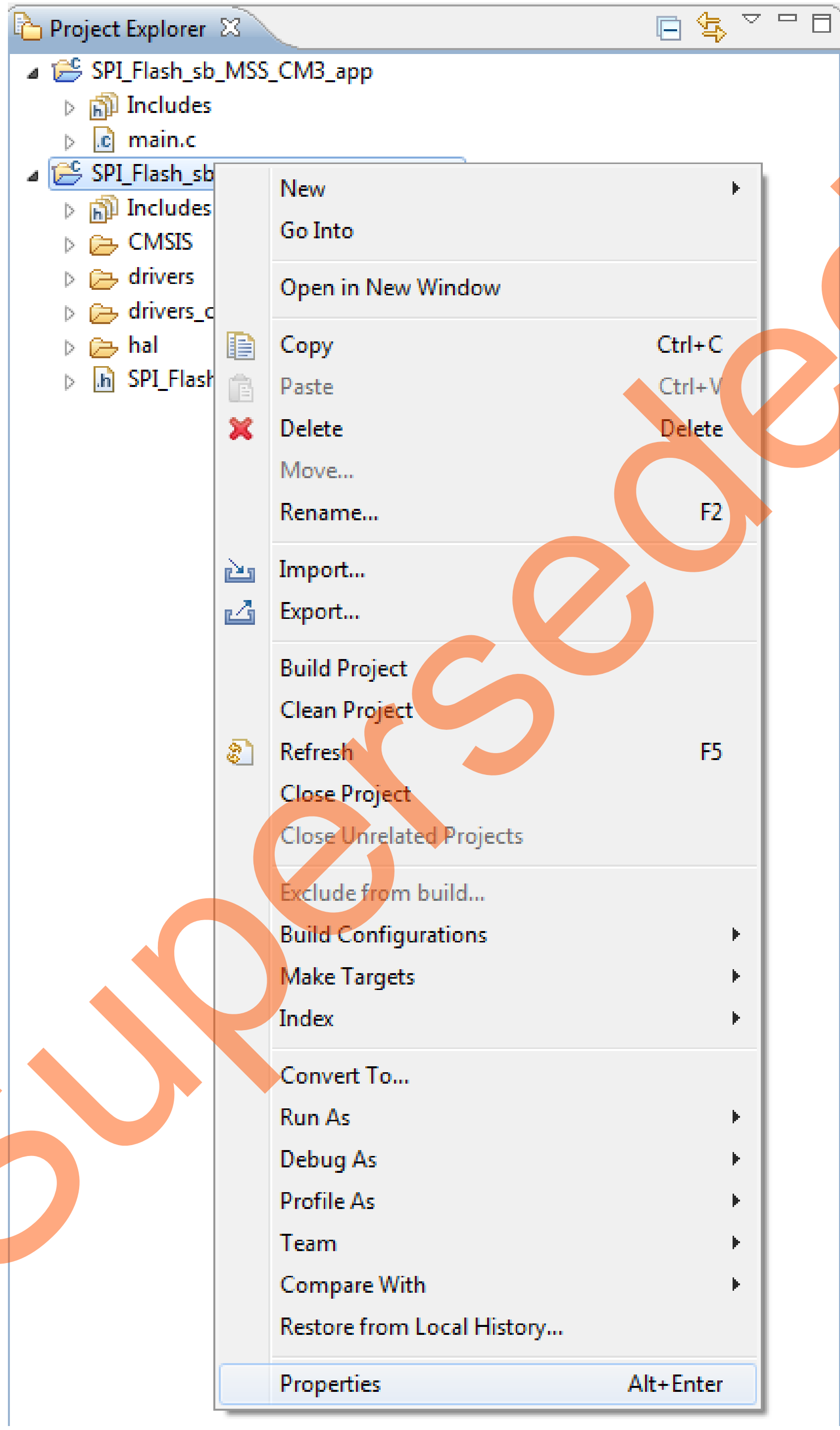


***Figure 23* • Project Explorer window - Properties**

8. In Properties window, select **Settings** under **C/C ++ Build**.
9. Select **Symbols** under **GNU C Compiler**.

10. To add a symbol, click **Add** and enter MICROSEMI_STDIO_THRU_UART in the **Add Symbol** dialog box and click **OK**.

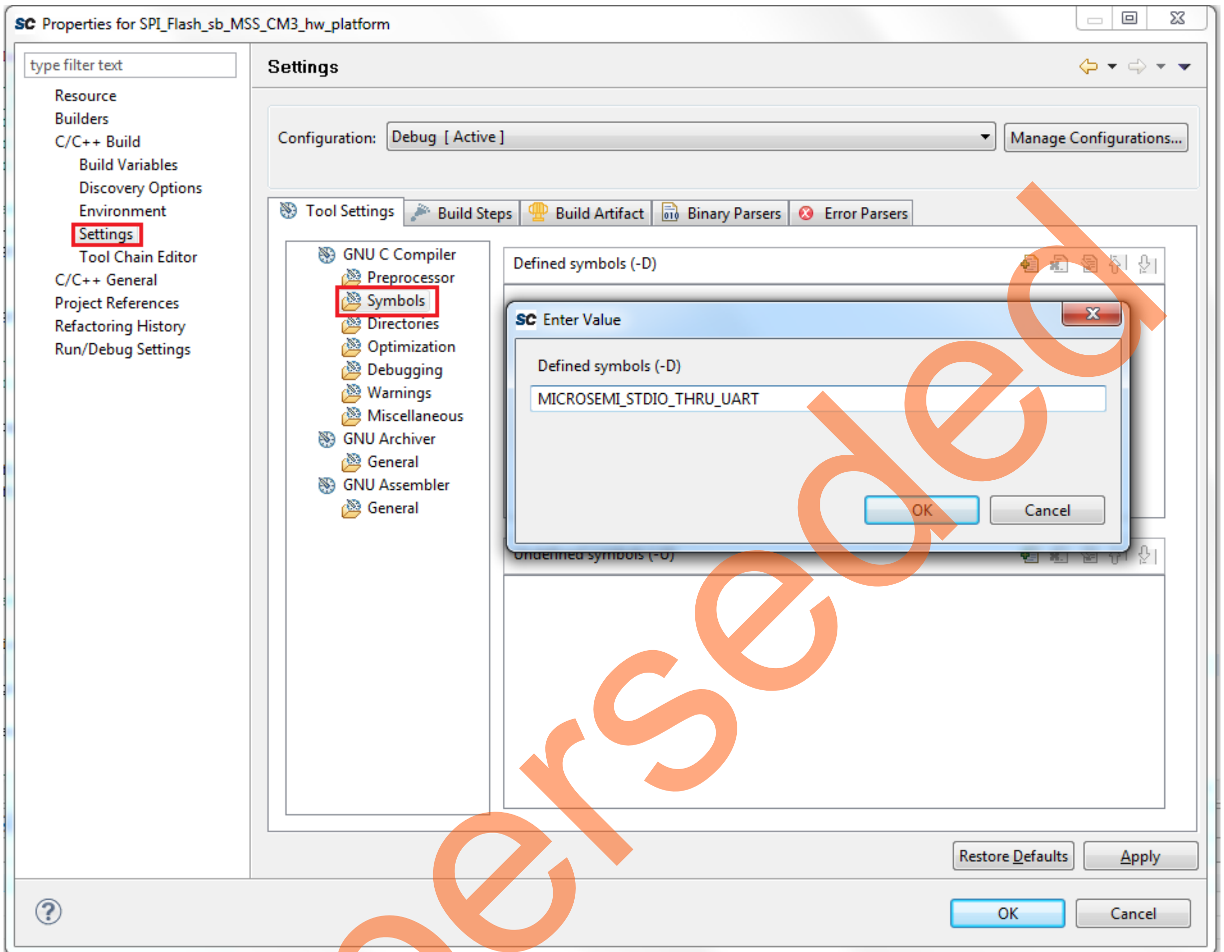


Figure 24 **• SPI_Flash_sb_MSS_CM3_HW_Platform Properties window**

11. Click **Apply** to save the changes made and click **OK** to close the **Properties** window.

12. Perform a build by selecting **Project** > **Clean**. Leave the default settings in the **Clean** dialog box and click **OK**, as shown in Figure 25.

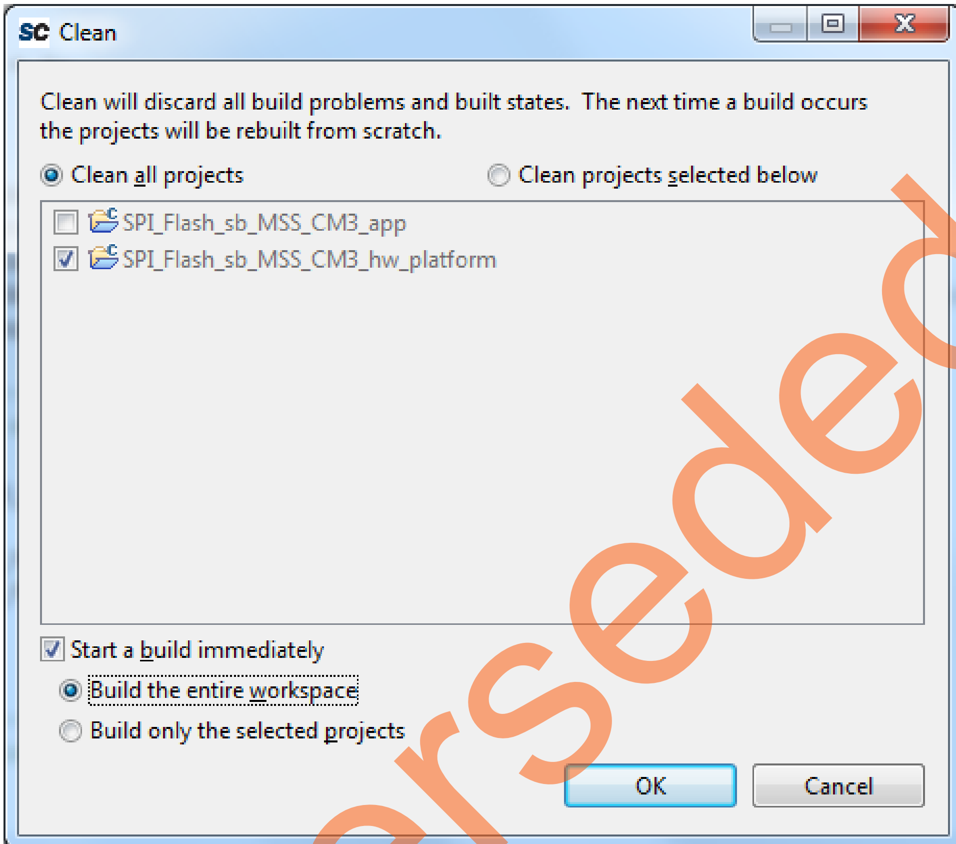


Figure 25 **• Settings for a clean build**

Note: Ensure that there are no errors.

Step 6: Configuring Serial Terminal Emulation Program

1. Install the USB driver. For serial terminal communication through the FTDI mini USB cable, install the FTDI D2XX driver. Download the drivers and the installation guide from: www.microsemi.com/soc/documents/CDM_2.08.24_WHQL_Certified.zip.
2. Connect the host PC to the J18 connector using the USB Mini-B cable. The USB to UART bridge drivers are automatically detected. Of the four COM ports, select the one with Location as **on USB Serial Converter D**. Figure 26 shows an example Device Manager window.

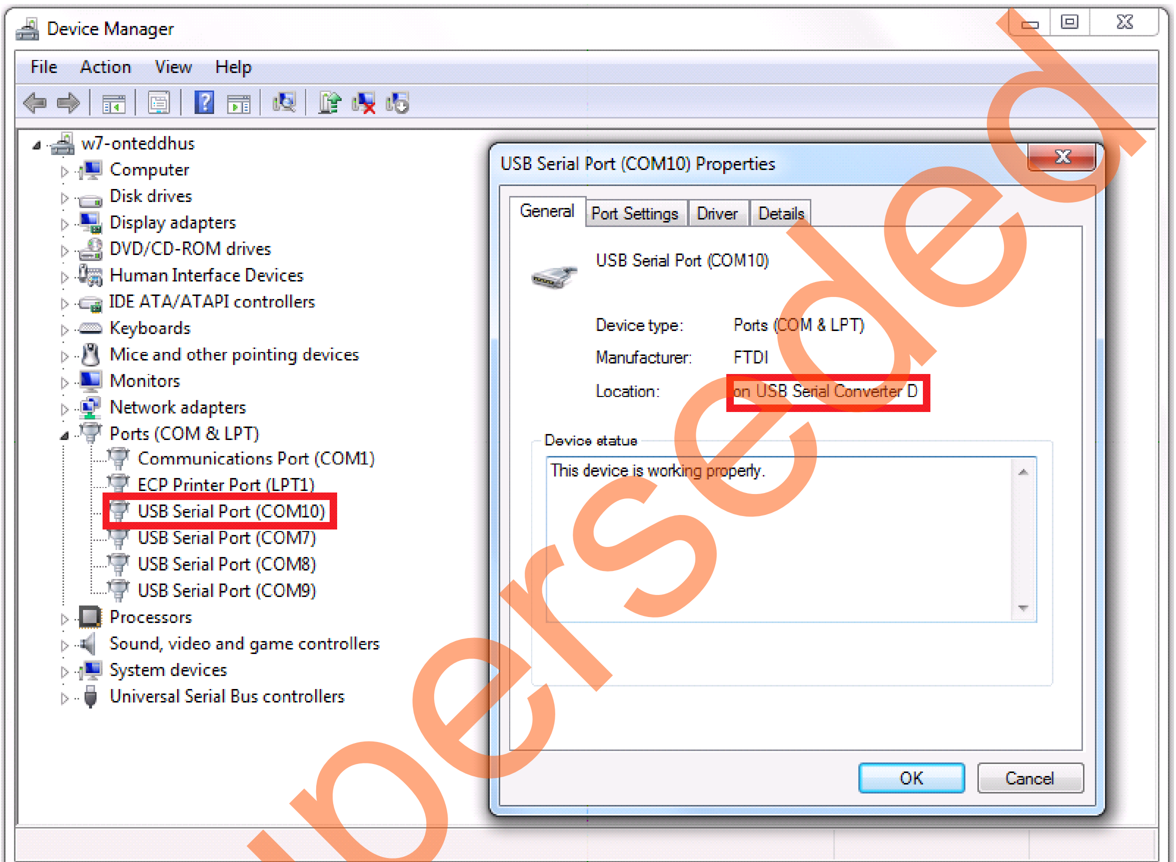


Figure 26 • ***Device Manager Window***

3. Start the HyperTerminal session. If the HyperTerminal program is not available in the computer, any free serial terminal emulation program such as PuTTY or TeraTerm can be used. Refer to the *Configuring Serial Terminal Emulation Programs Tutorial* for configuring the HyperTerminal, TeraTerm, or PuTTY.

   The HyperTerminal settings are as follows:
   – 115200 baud rate
   – 8 data bits
   – 1 stop bit
   – No parity
   – No flow control

Step 7: Debugging the Application Project using SoftConsole

1. Select **Debug Configurations** from the **Run** menu of the SoftConsole. The **Debug Configurations** dialog box is displayed. Double-click **Microsemi Cortex-M3 Target** to view the configurations, as shown in Figure 27.

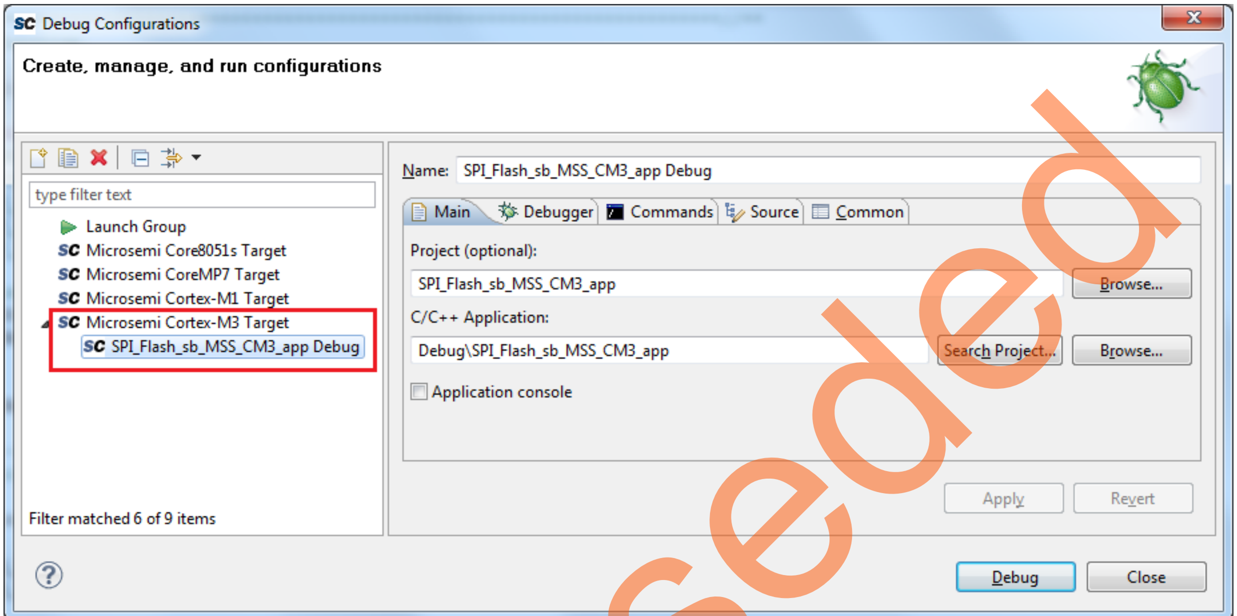


Figure 27 **• Debug Configurations**

2. Ensure that the following values are filled in the corresponding fields:
   – Name: SPI_Flash_sb_MSS_CM3_app Debug
   – Project (optional): SPI_Flash_sb_MSS_CM3_app
   – C/C++ Application: Debug\SPI_Flash_sb_MSS_CM3_app
3. Click **Debug**.
4. On the **Confirm Perspective Switch** window, click **Yes**, as shown in Figure 28.

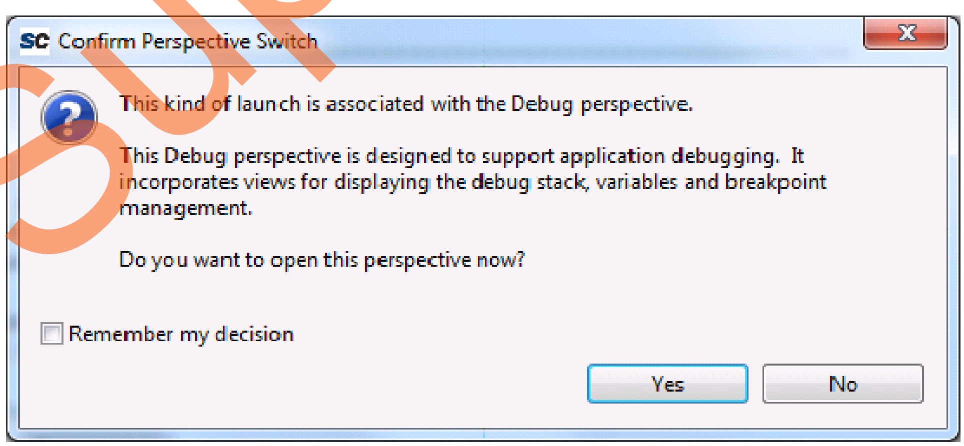


Figure 28 **• Confirm Perspective Switch**

5. The **SoftConsole Debugger Perspective** window is displayed, as shown in Figure 29.

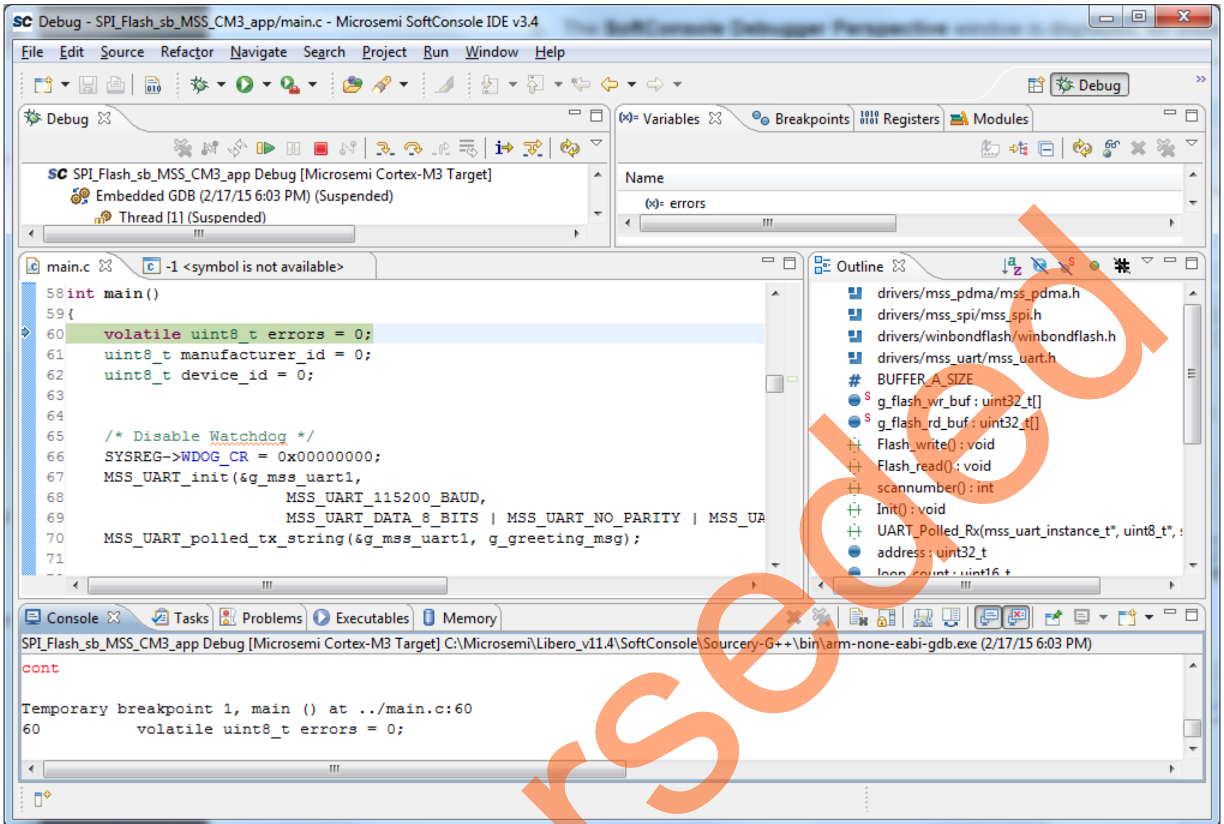


Figure 29 **• SoftConsole Debugger Perspective**

6. Run the application by clicking **Run > Resume**. A greeting message on the HyperTerminal is displayed as shown in Figure 30.

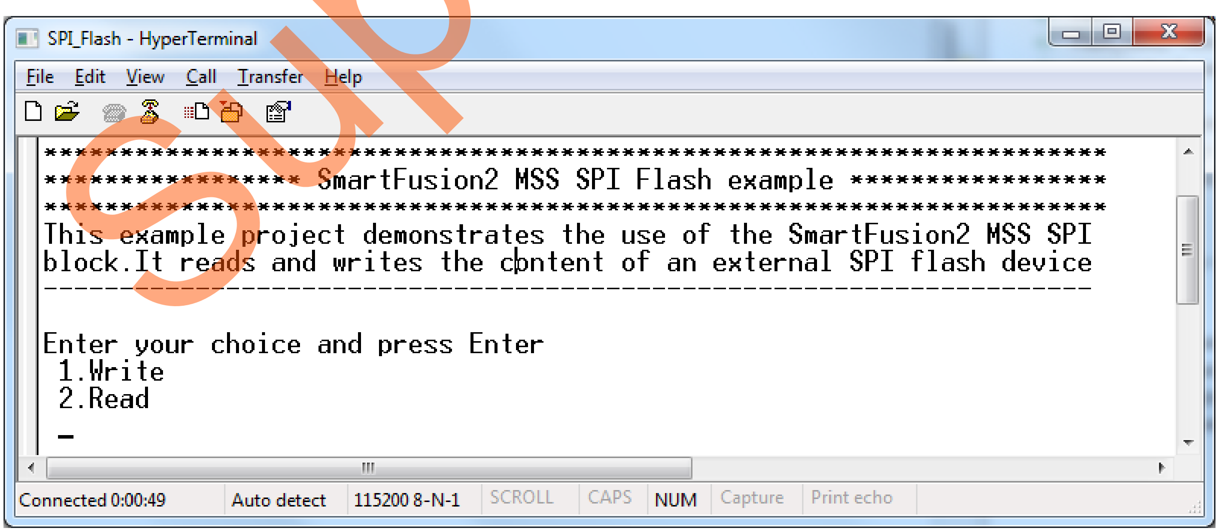


Figure 30 **• HyperTerminal Window**

7. Select option 1 and enter values to write to the SPI flash memory as shown in Figure 31.

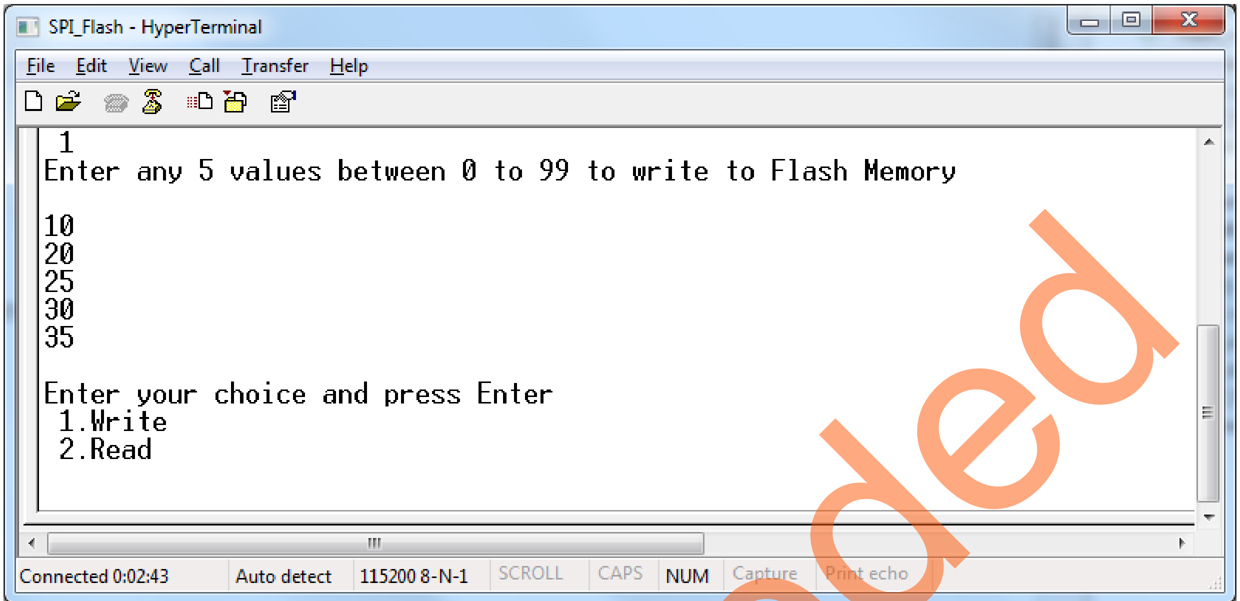


Figure 31 **• HyperTerminal Window - Option 1**

8. Select option 2 to read data from SPI flash memory as shown in Figure 32.

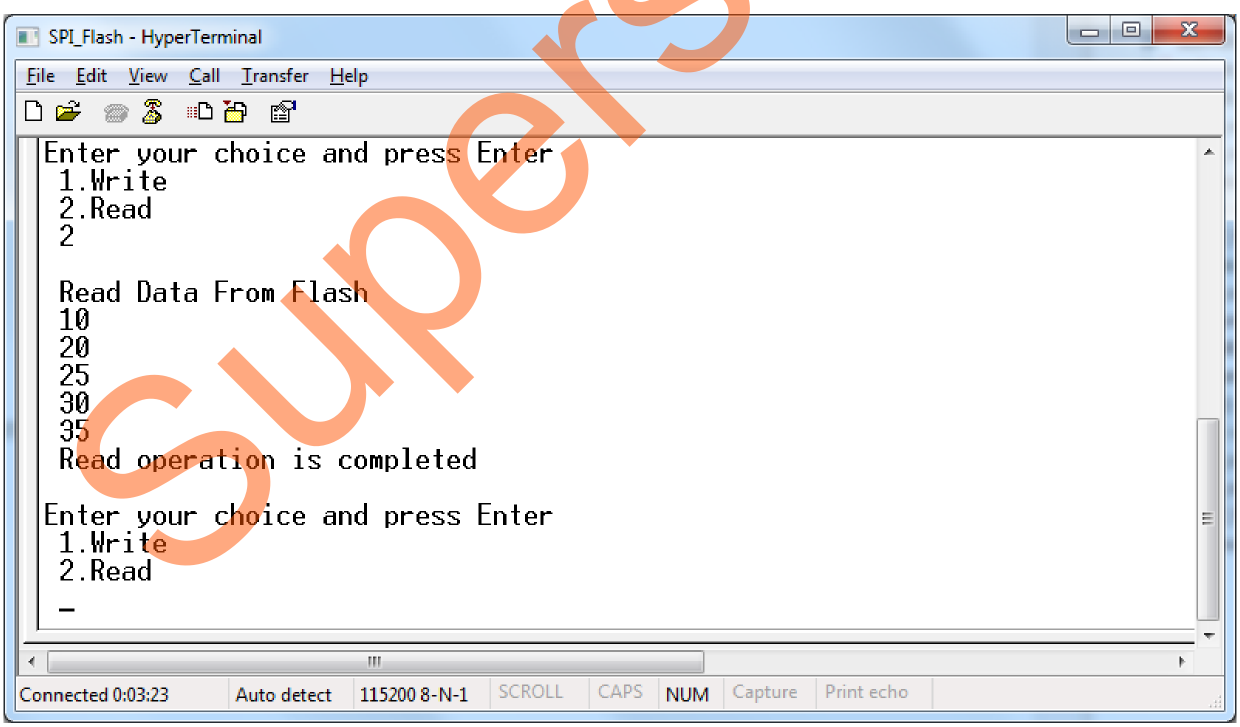


Figure 32 **• HyperTerminal Window - Option 2**

9. Terminate execution of the code by choosing **Run > Terminate**.
10. Launch the debug session:

– By selecting **Debug Configurations** from the **Run** menu of SoftConsole.

or

– By selecting the Debug Configurations using Debug Button as shown in Figure 33.

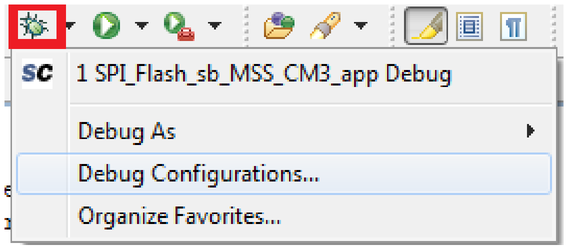


***Figure 33* • Debug Configurations Option**

11. Click the **Registers** tab to view the values of the ARM® Cortex®-M3 processor internal registers, as shown in Figure 34.

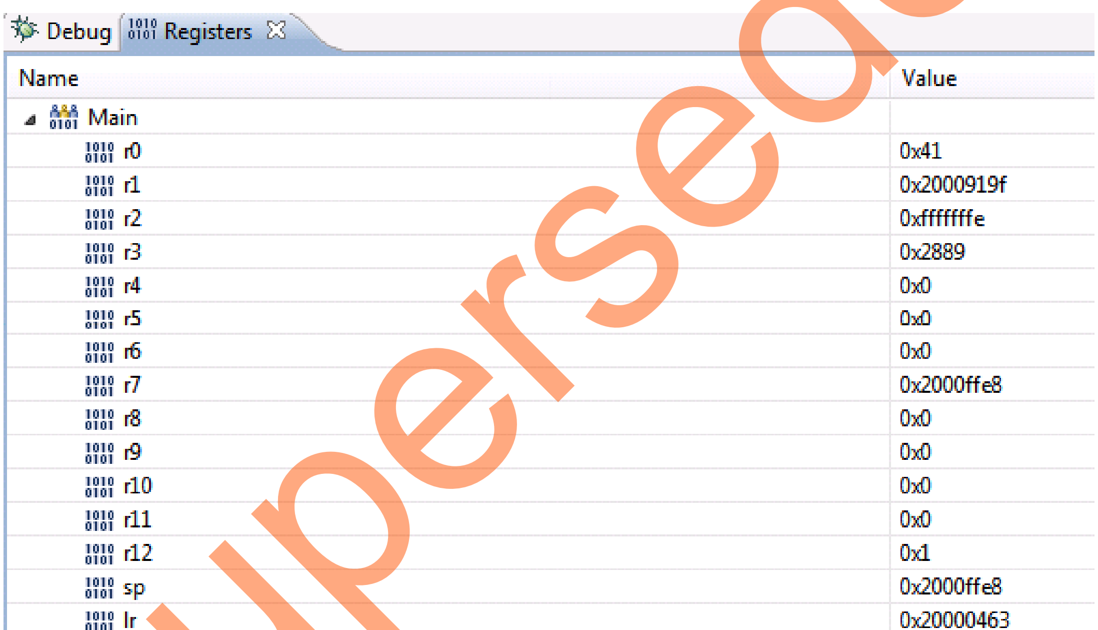
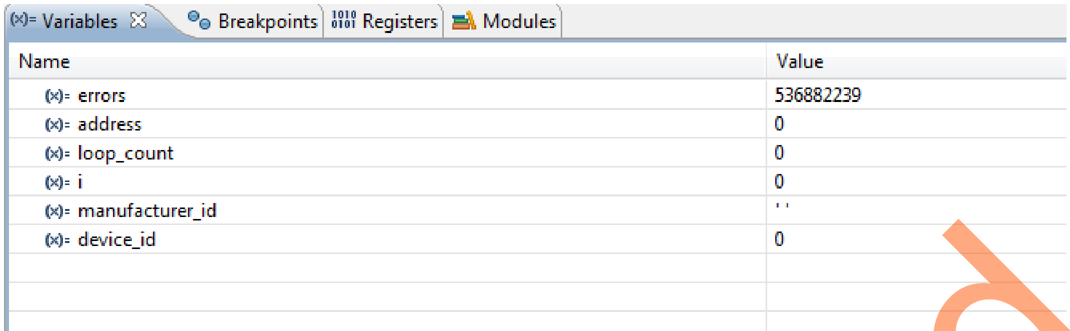


***Figure 34* • Values of Cortex-M3 Internal Registers**

12. Click the **Variables** tab to view the values of variables in the source code, as shown in Figure 35.



Name	Value
(x)= errors	536882239
(x)= address	0
(x)= loop_count	0
(x)= i	0
(x)= manufacturer_id	' '
(x)= device_id	0

Figure 35 • **Values of the Variables in the Source Code**

13. In the **Debug** window, select **Window > Show View > Disassembly** to display the assembly level instructions. The **Disassembly** window with assembly instructions is displayed on the right-side of the Debug perspective, as shown in Figure 36.

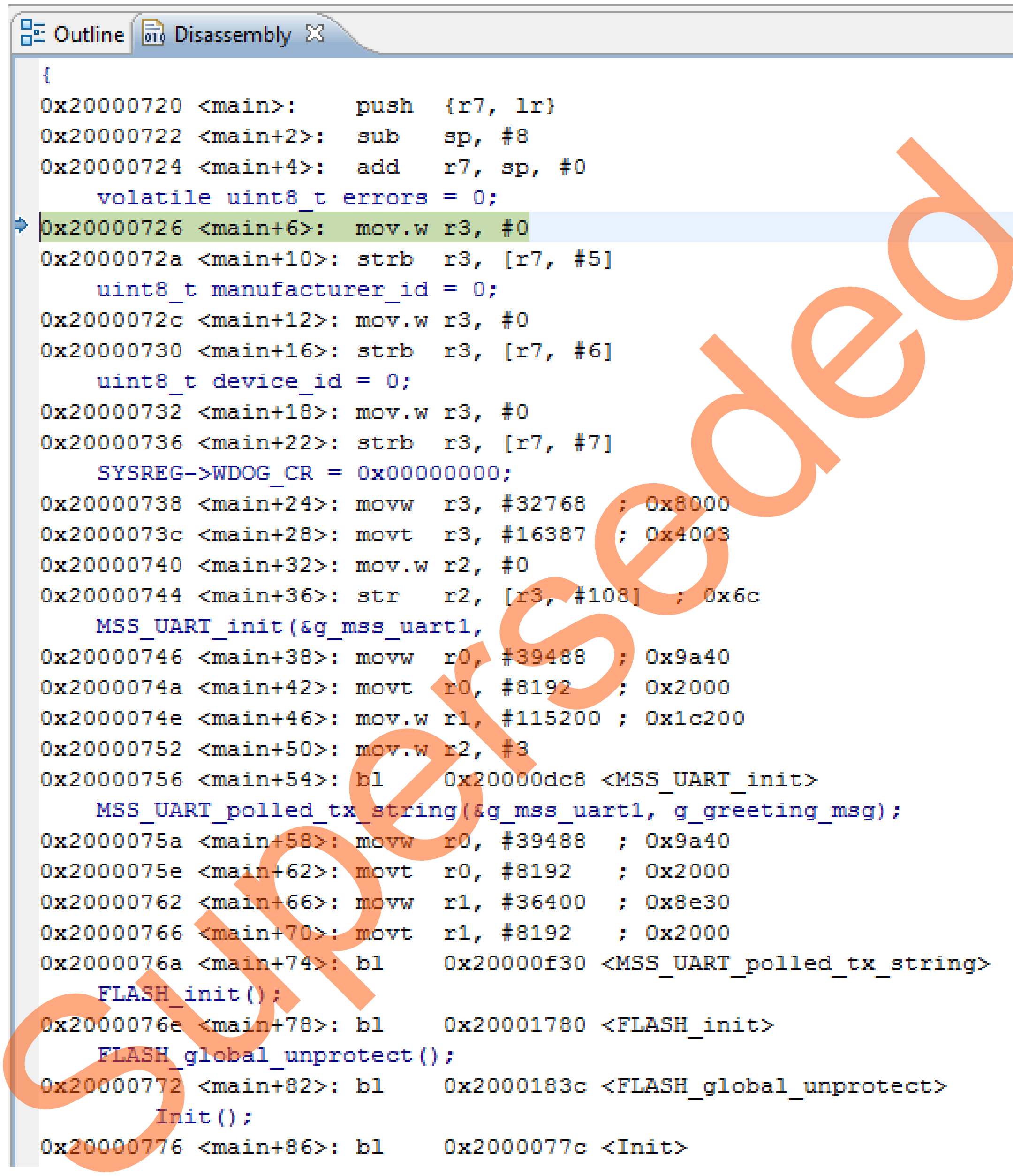


***Figure 36* • Assembly Level Instructions**

14. Source code can be single-stepped by choosing **Run > Step Into or Run > Step Over**. Observe the changes in the source code window and Disassembly view. Performing a Step Over provides an option for stepping over functions. The entire function is run but there is no need to single-step through each instruction contained in the function.
15. Click **Instruction Stepping** (i⇒) and perform **Step Into** operations. Observe that **Step Into** executes a single line of assembly code.

16. Click **Instruction Stepping** to exit the instruction stepping mode. Single-step through the application and observe the instruction sequence in the source code window of the Debug perspective, and the values of the variables and registers.
17. Add breakpoints in the application to force the code to halt, then single-step and observe the instruction sequence.
18. When debug process is finished, terminate execution of the code by choosing **Run > Terminate**.
19. Close Debug Perspective by selecting **Close Perspective** from the Window menu.
20. Close SoftConsole using **File > Exit**.
21. Close the HyperTerminal using **File > Exit**.

Conclusion

This tutorial provides steps to create a Libero SoC design using the System Builder. It describes the procedure to build, debug, and run a SoftConsole application. It also provides a simple design to access SPI flash.

Appendix A - Board Setup for Running the Tutorial

Figure 1 shows the board setup for running the tutorial on the SmartFusion2 Security Evaluation Kit board.

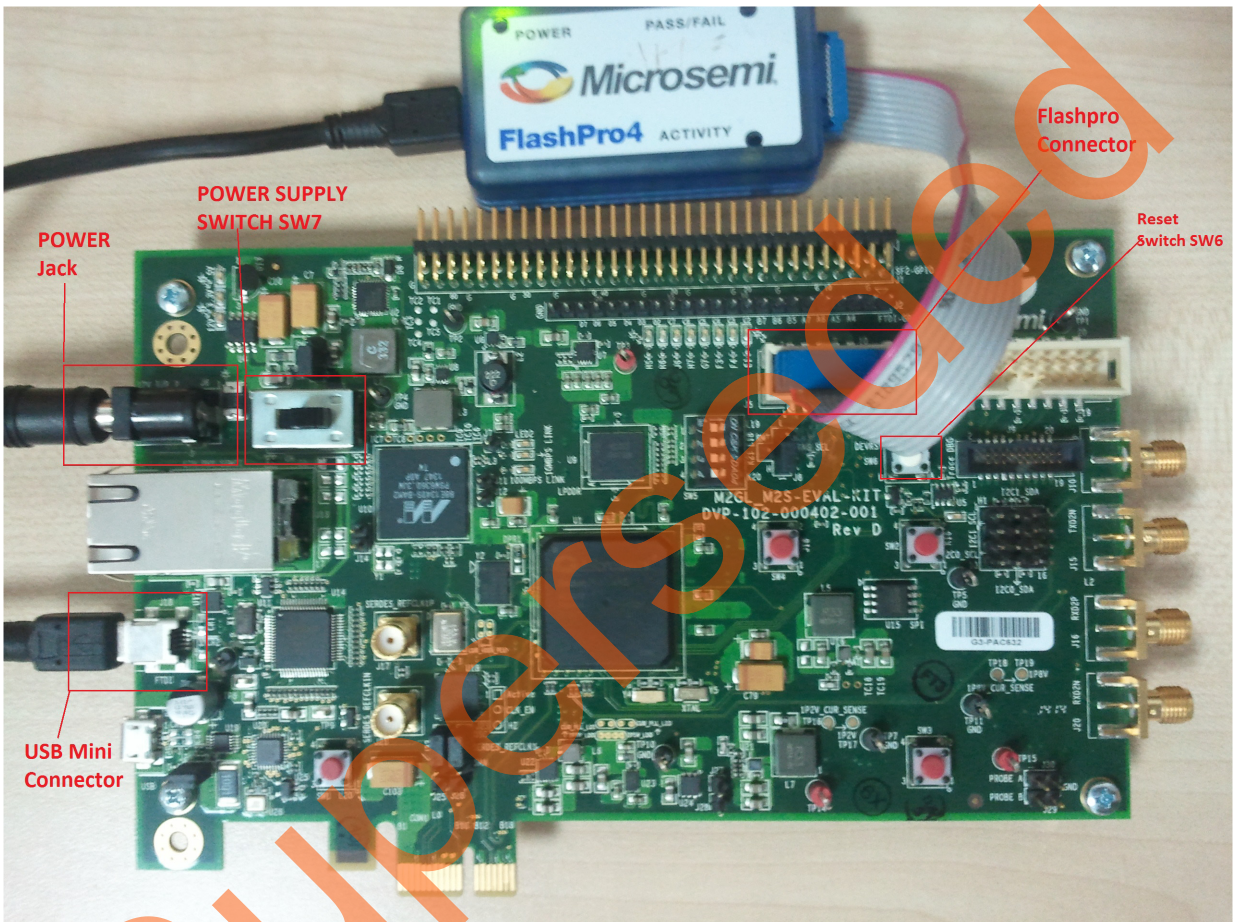


Figure 1 • **SmartFusion2 Security Evaluation Kit Setup**

Appendix B - SmartFusion2 Security Evaluation Kit Board Jumper Locations

Figure 1 shows the jumper locations on the SmartFusion2 Security Evaluation Kit board.

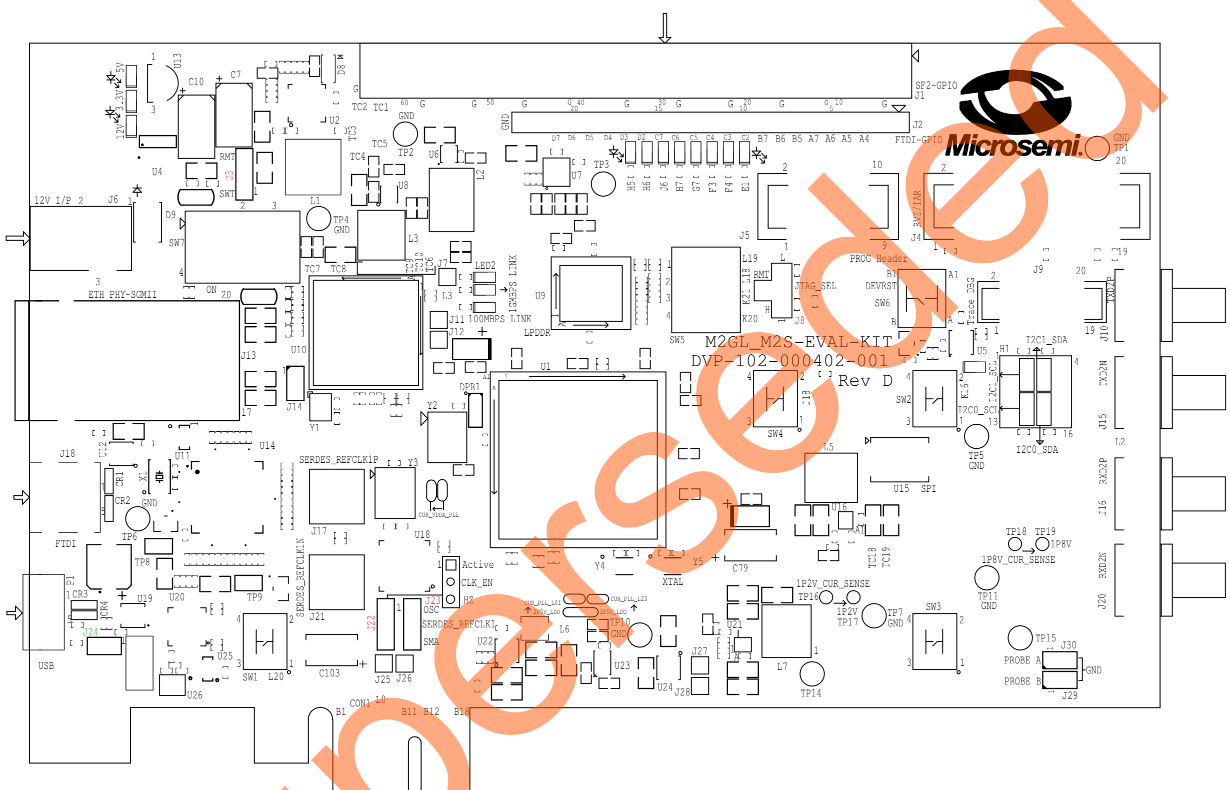


Figure 1 • SmartFusion2 Security Evaluation Kit Board Jumper Locations

Notes:

- Jumpers highlighted in red are set by default.
- Jumpers highlighted in green must be set manually.
- The locations of the jumpers in Figure 1 are searchable.

Microsemi

List of Changes

The following table shows important changes made in this document for each revision.

Revision*	Changes	Page
Revision 3 (March 2015)	Updated the document for Libero SoC v11.5 software release (SAR 64190).	N/A
Revision 2 (October 2014)	Updated the document for Libero SoC v11.4 software release (SAR 61627).	N/A
Revision 1 (April 2014)	Initial release.	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

For Microsemi SoC Products Support, visit
http://www.microsemi.com/products/fpga-soc/designsupport/fpga-soc-support

Website

You can browse a variety of technical and non-technical information on the SoC home page, at 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 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

E-mail: sales.support@microsemi.com

50200546-3/03.15