Accessing Serial Flash Memory Using SPI Interface - Libero SoC v11.4

Libero SoC and SoftConsole Flow Tutorial for SmartFusion2 SoC FPGA

Microsemi

Table of Contents

Accessing Serial Flash Memory using SPI Interface- Libero SoC v11.4

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) 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 Evaluation Kit • FlashPro4 programmer • USB A to Mini-B cable • 12 V adapter	Rev C or later
Host PC or Laptop	Any 64-bit Windows Operating System
Software Requirements	
Libero SoC	v11.4
SoftConsole	v3.4SP1
FlashPro programming software	v11.4
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=sf2_spi_flash_sc_liberov11p4_tu_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 Evaluation Kit board (SF2_EVAL_KIT) Rev C (or later).

Design Overview

This design example demonstrates the execution of basic read and write operations on the SPI flash present on the SmartFusion2 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. The SPI flash memory transfers are performed using the peripheral direct memory access (PDMA).

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.

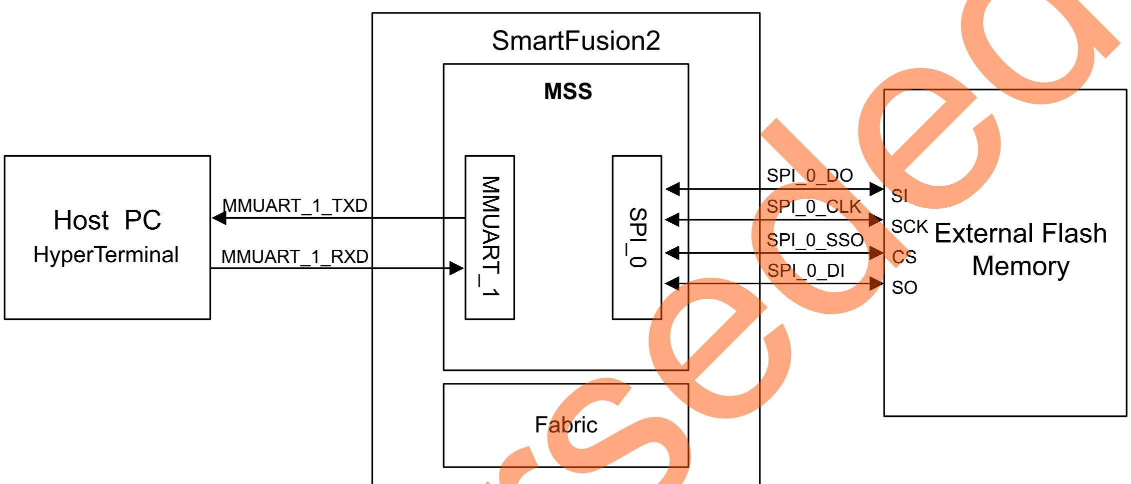


***Figure 1* • SPI Flash Interfacing Block Diagram**

Step 1: Creating a Libero SoC Project

Launching Libero SoC

1. Click **Start > Programs > Microsemi Libero SoC v11.4 > Libero SoC v11.4**, or click the shortcut on desktop to open the Libero SoC v11.4 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.

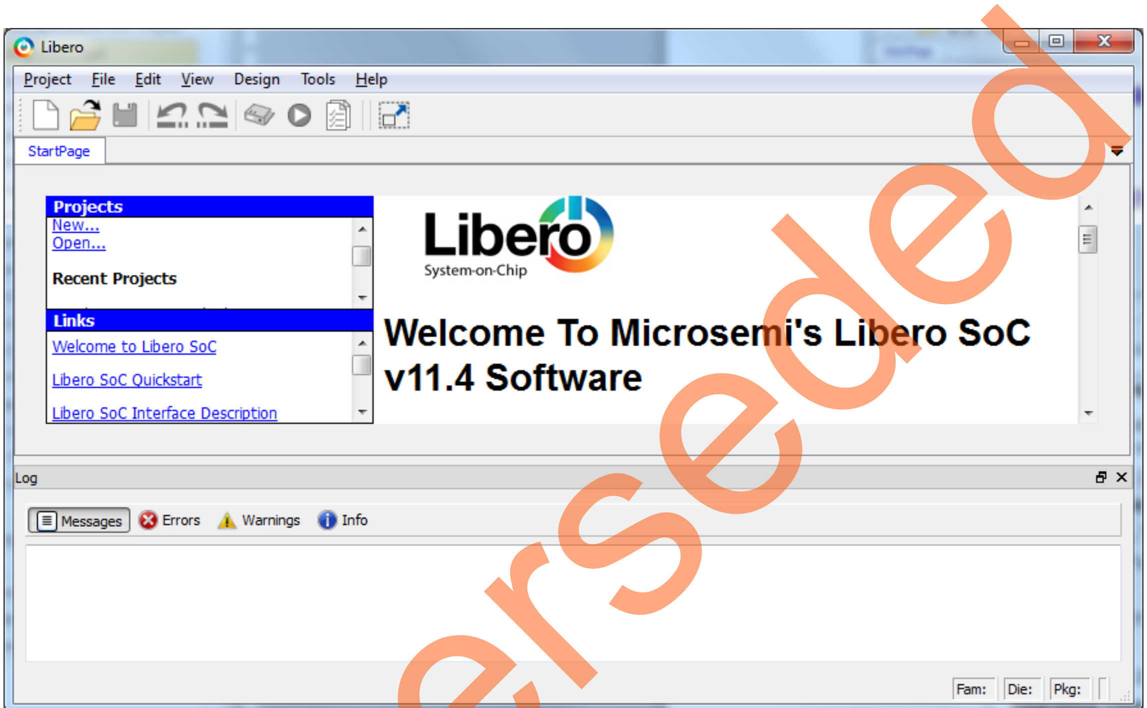


Figure 2 • **Libero SoC Project Manager**

3. Enter the information as required for the new project and the device in the **New Project** dialog box as shown in Figure 3.
   – Project
   - Name: SPI_Flash
   - Location: Select an appropriate location (for example, D:/Microsemi_prj)
   - Preferred HDL type: Verilog
   – Device (select the following values using the drop-down list provided):
   - Family: SmartFusion2
   - Die: M2S025T
   - Package: 484 FBGA
   - Speed: STD
   - Core Voltage: 1.2
   – Operating conditions: COM

4. Check the **Use Design Tool** check box and select **Use System Builder** in the **Design Templates and Creators** section of the **New Project** window as shown in Figure 3.

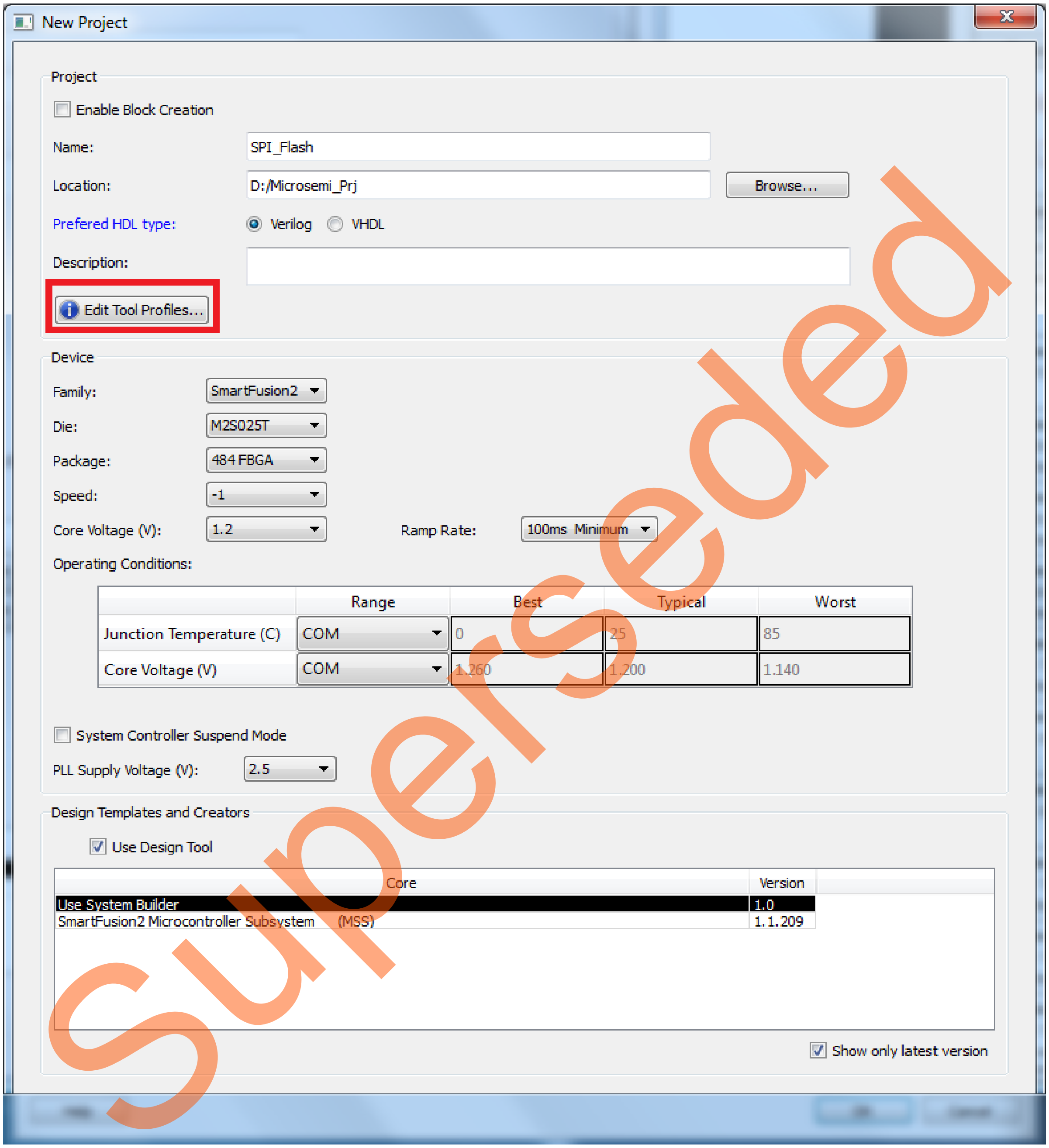


Figure 3 **• New Project Dialog Box**

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.

5. Clicking **Edit Tool Profiles** (highlighted in Figure 3 on page 7) displays the **Tool Profiles** window as shown in Figure 4. Check the following tool settings:
   - Software IDE: SoftConsole

- Synthesis: Synplify Pro ME I-2013.09M-SP1
- Simulation: ModelSim ME 10.2c
- Programming: FlashPro 11.4

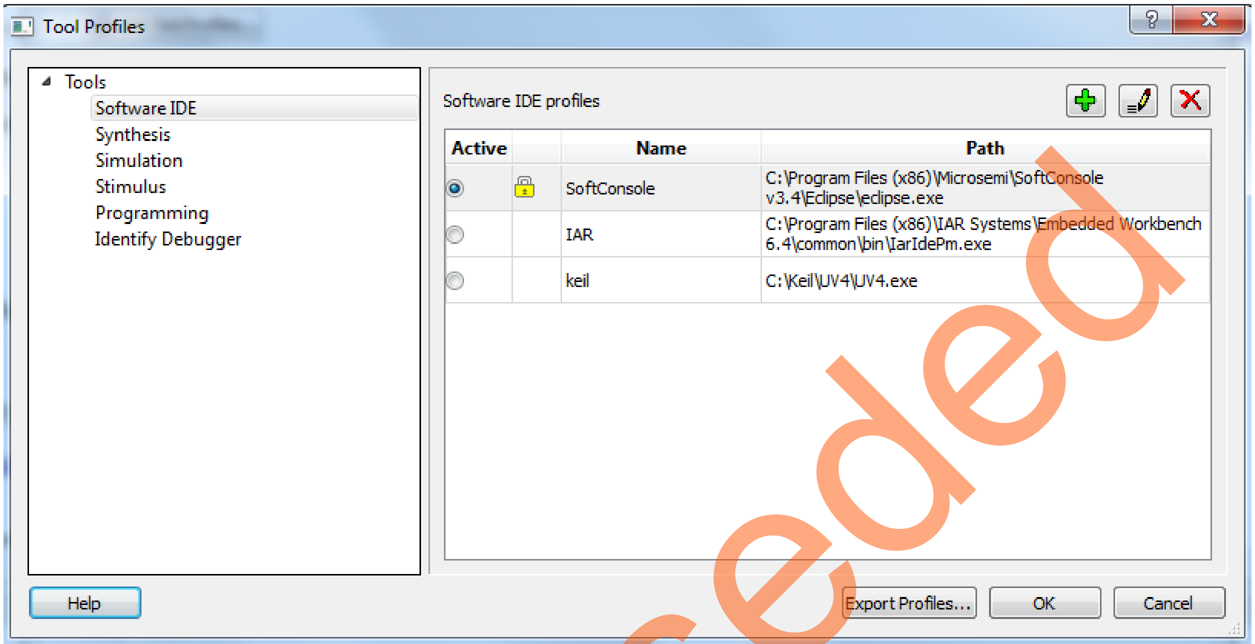


***Figure 4* • Tool Profiles**

6. Click **OK** on the **Tool Profiles** window.
7. Click **OK** on the **New Project** window. This displays the **System Builder** dialog box.
8. **Enter a name for your system**, enter **SPI_Flash** as the name of the system and click **OK**. The **System Builder** dialog box is displayed with the **Device Features** page open by default, as shown in Figure 5.

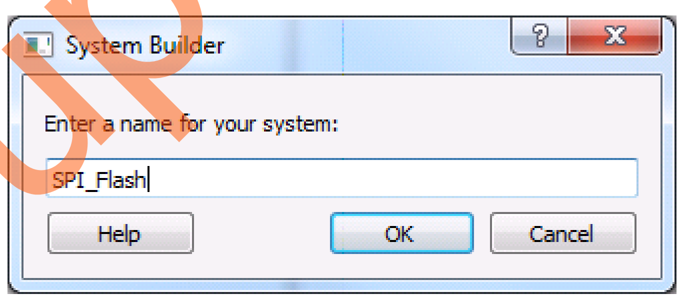


***Figure 5* • Create New System Builder Dialog Box**

9. In the **System Builder – Device Features** page, check the **Peripheral DMA** check box under **Microcontroller Options** as shown in Figure 6.

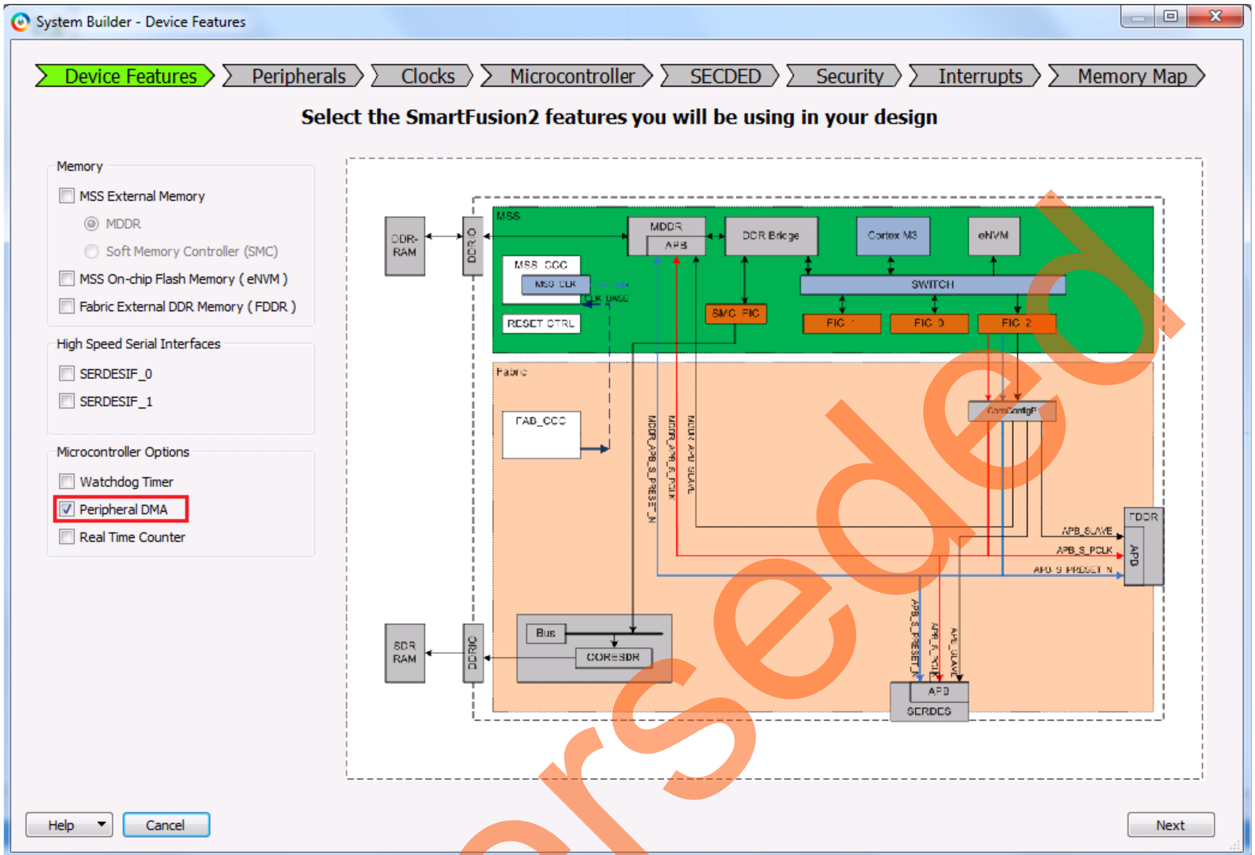


Figure 6 • System Builder – Device Features Page

10. Click **Next**. The **System Builder – Peripherals** page is displayed. Under the MSS Peripherals section, uncheck all the check boxes except **MM_UART_1** and **MSS_SPI_0**, as shown in Figure 7.

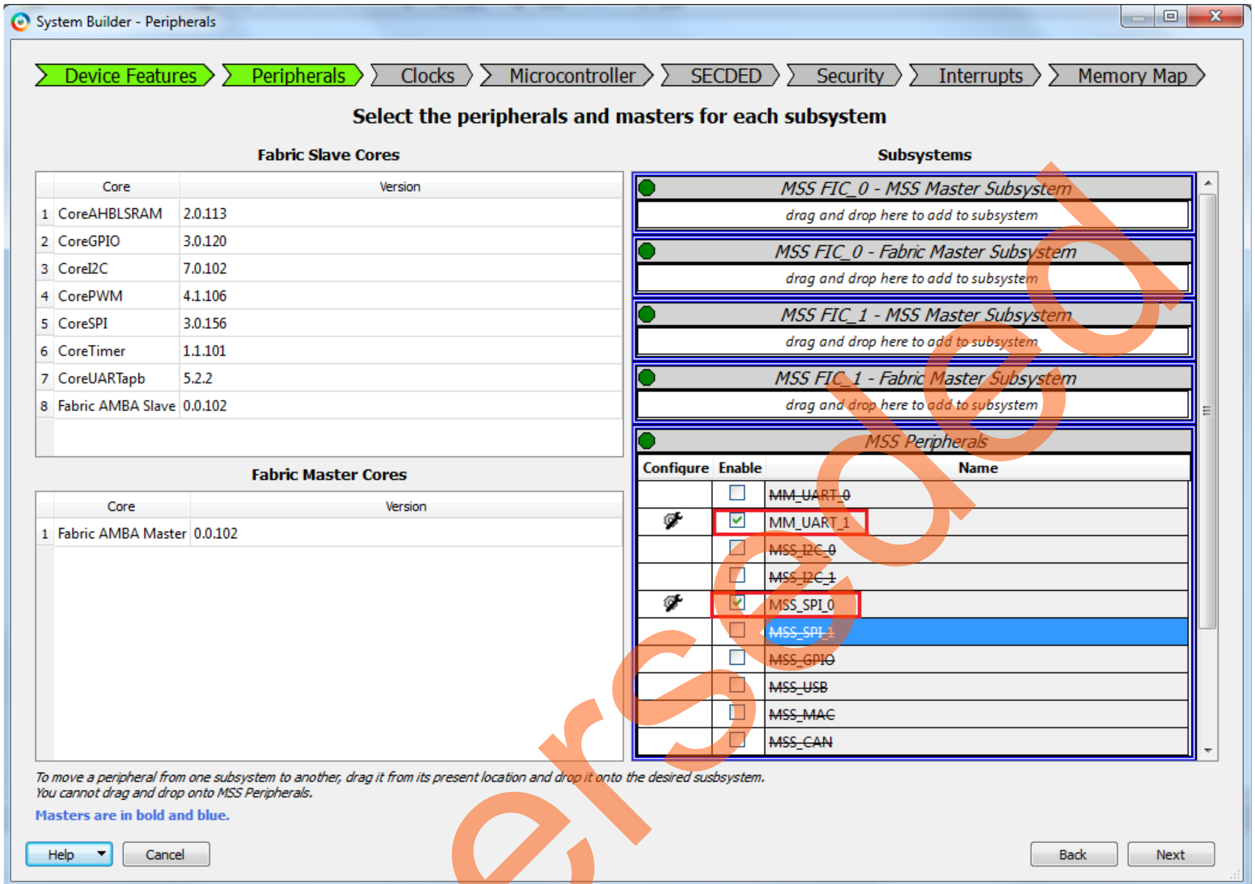


Figure 7 **• System Builder Configurator – Peripherals Page**

11. Configure **MMUART_1** for Fabric by clicking on the **MM_UART_1** configurator highlighted as shown in Figure 8.

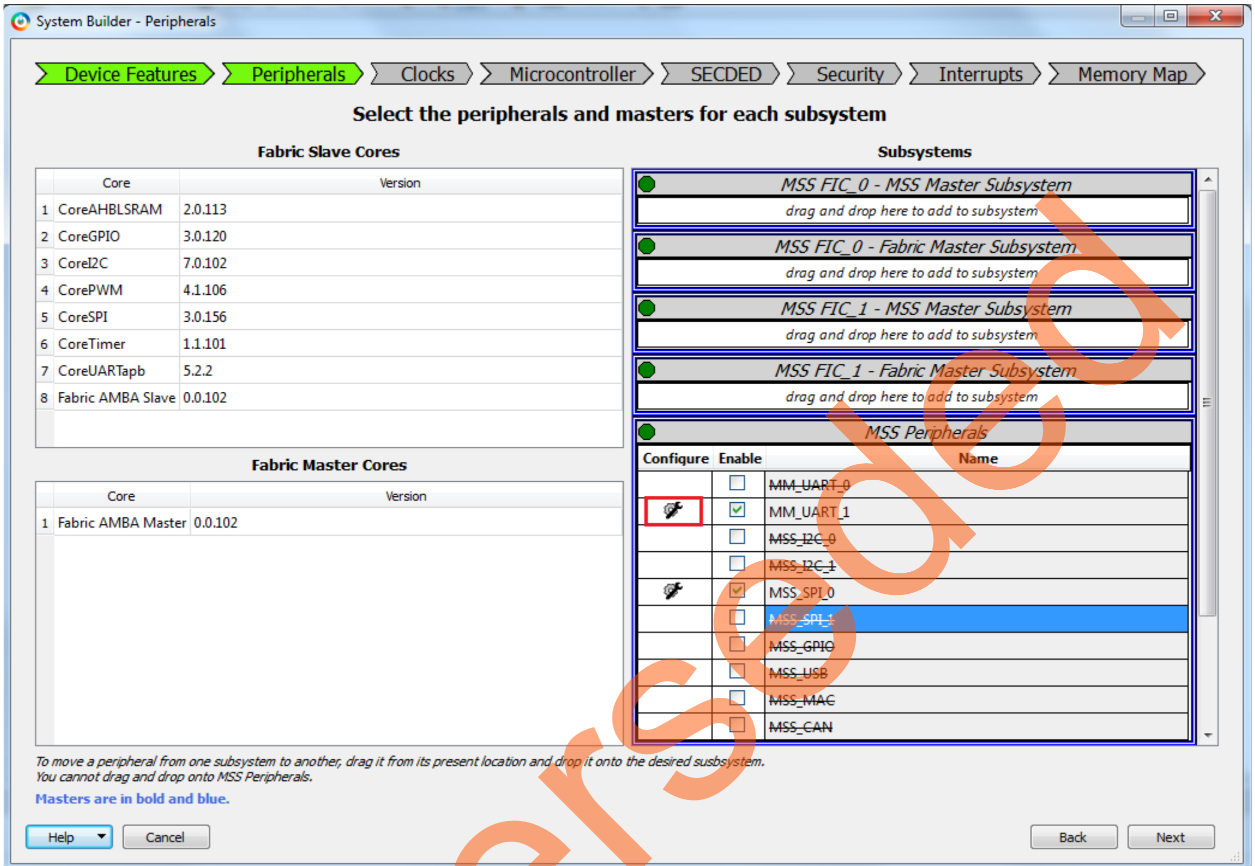


***Figure 8* • System Builder – Peripherals Page**

12. In the MM_UART_1 configurator window, select **Fabric** from the **Connect To** drop-down list, as shown in Figure 9.

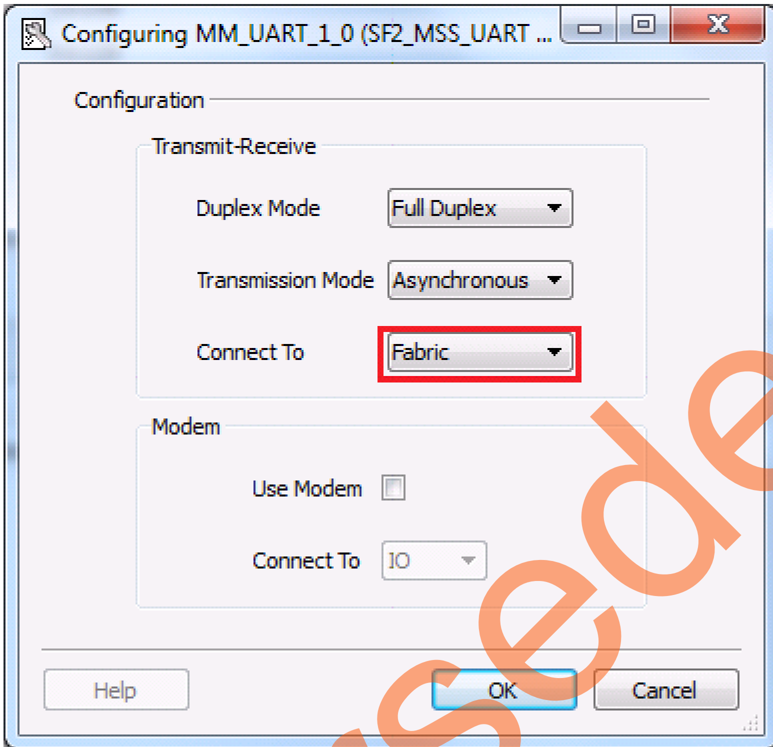


***Figure 9* • Configuring MM_UART_1**

13. Click **Next**. The **System Builder – Clocks** page is displayed, as shown in Figure 10. Select **System Clock** source as **On-chip 25/50 MHz RC Oscillator**. The M3_CLK is configured to 100 MHz by default.

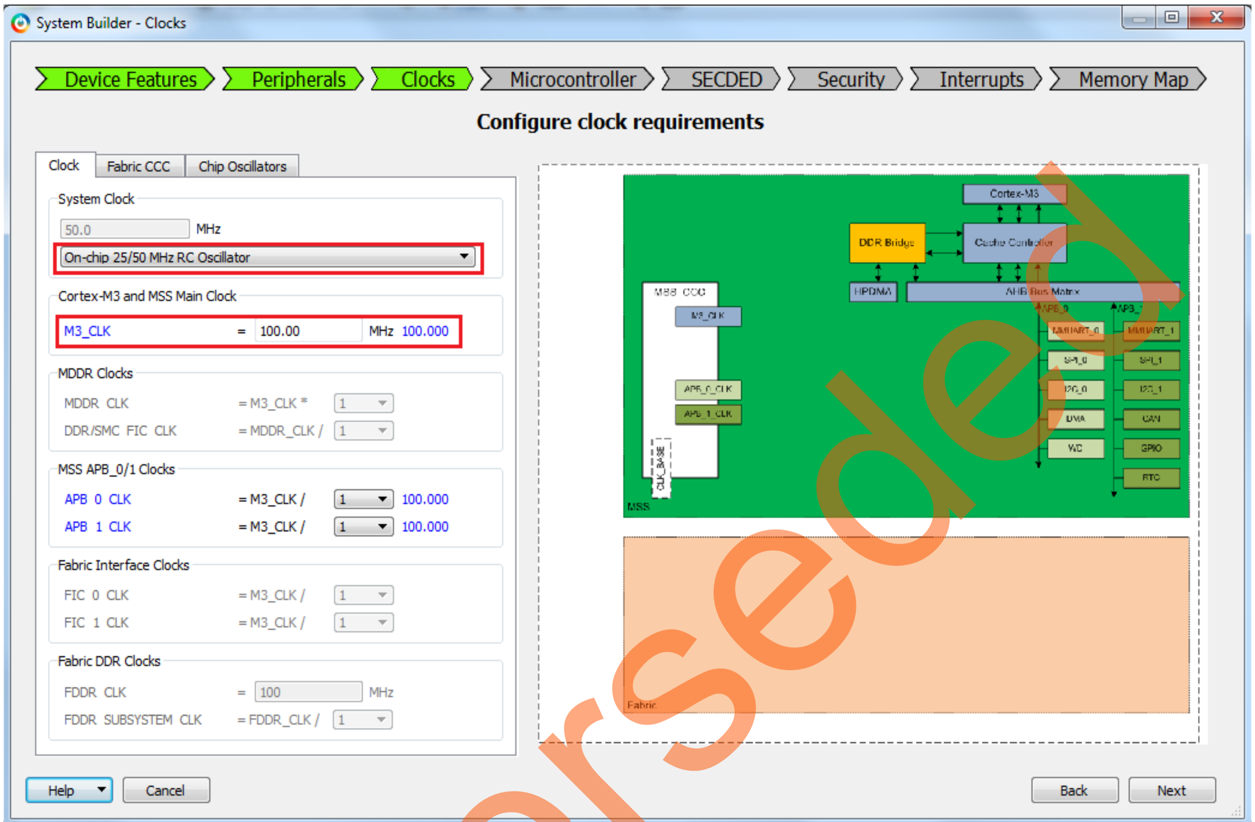


Figure 10 **• System Builder – Clocks Page**

14. Click **Next**. The **System Builder – Microcontroller** page is displayed. Leave all the default selections.
15. Click **Next**. The **System Builder – SECDED** page is displayed. Leave all the default selections.
16. Click **Next**. The **System Builder – Security** page is displayed. Leave all the default selections.
17. Click **Next**. The **System Builder – Interrupts** page is displayed. Leave all the default selections.
18. Click **Next**. The **System Builder – Memory Map** page is displayed. Leave all the default selections.
19. 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 automatically, as shown in Figure 11.

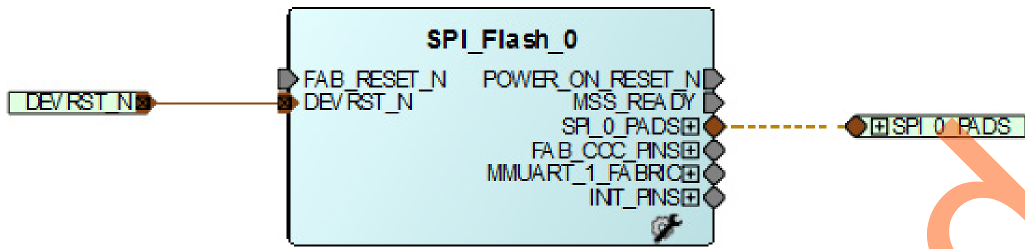


Figure 11 • **System Builder Generated System**

Connecting Components in SPI_Flash_top SmartDesign

Perform the following steps to connect the SmartDesign components:

1. Right-click **FAB_RESET_N** and select **Tie High**.
2. Right-click **POWER_ON_RESET_N** and select **Mark Unused**.
3. Right-click **MSS_READY** and select **Mark Unused**.
4. Right-click **MMUART_1_FABRIC** and select **Promote to Top Level**.
5. Expand **INIT_PINS**, right-click **INIT_DONE** and select **Mark Unused**.
6. Expand **FAB_CCC_PINS**, right-click **FAB_CCC_GL0** and select **Mark Unused**.
7. Click **File** > **Save**. The SPI_Flash_top design is displayed as shown in Figure 12.

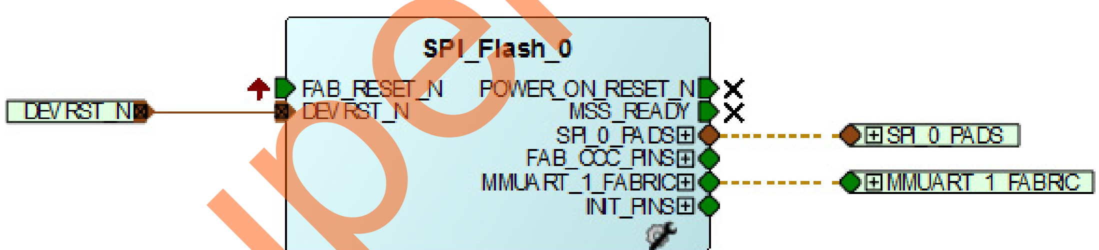


Figure 12 • **SPI_Flash_top Design**

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
- PDMA
- SPI

To generate the required drivers,

1. Click **Design > Configure Firmware** and select all the drivers' check boxes, except CMSIS, MMUART, PDMA, and SPI as shown in Figure 13.

   Note:Select the latest version of the drivers.

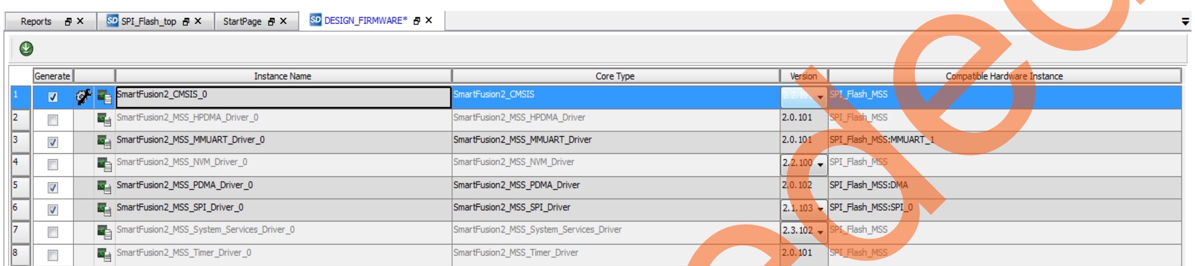


Figure 13 • **Configuring Firmware**

2. From the **SPI_Flash_top** tab, click **Generate Component**, as shown in Figure 14.

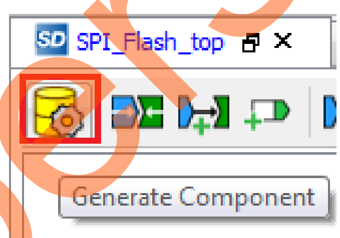


Figure 14 • **Generate Component**

If the design is generated without any errors, a message **'SPI_Flash_top' was generated** is displayed on the Libero SoC **Log** window as shown in Figure 15.

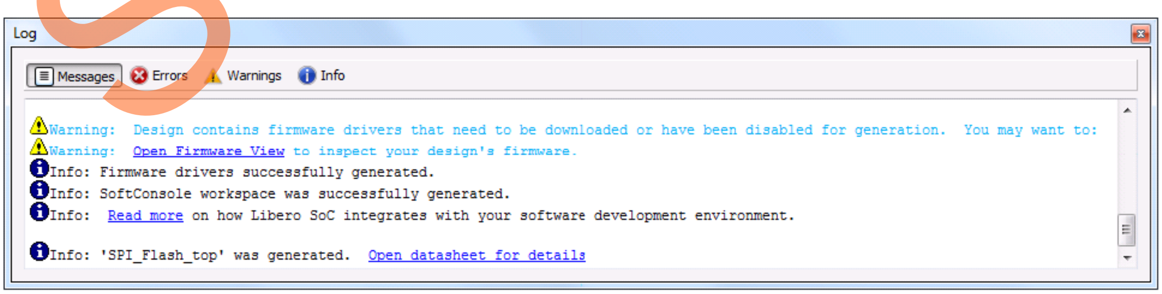


Figure 15 • **Log Window**

Step 2: Generating the Program File

1. Double-click **I/O Constraints** in the **Design Flow** window as shown in Figure 16. The **I/O Editor** window is displayed after completing **Synthesize and Compile**.

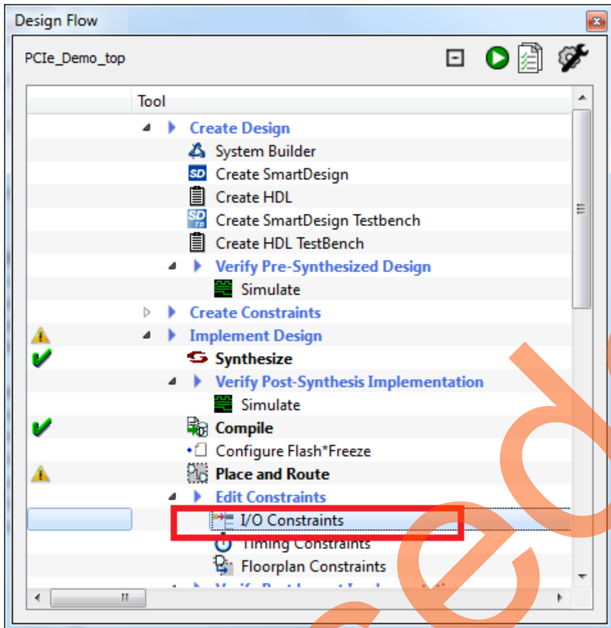


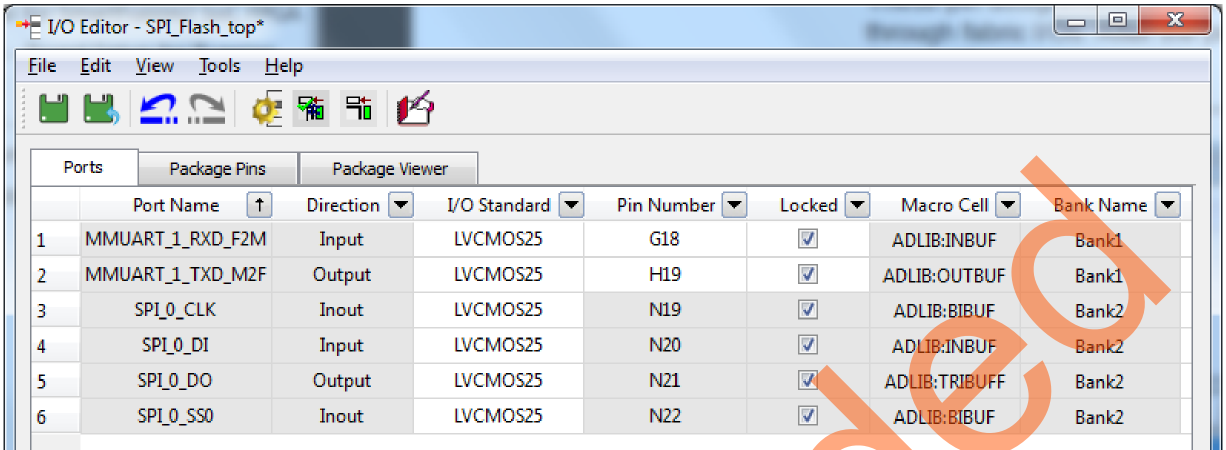
Figure 16 **• I/O Constraints**

2. In the **I/O Editor** window, make the pin assignments as shown in Table 2.

Table 2 • **Port to Pin Mapping**

Port Name	Pin Number
MMUART_1_RXD_F2M	G18
MMUART_1_TXD_M2F	H19

These pin assignments are for connecting MMUART_1 ports TX and RX to the mini-B USB through fabric I/Os. After the pins are assigned, the **I/O Editor** window is displayed as shown in Figure 17.



	Port Name	Direction	I/O Standard	Pin Number	Locked	Macro Cell	Bank Name
1	MMUART_1_RXD_F2M	Input	LVCMOS25	G18	☑	ADLIB:INBUF	Bank1
2	MMUART_1_TXD_M2F	Output	LVCMOS25	H19	☑	ADLIB:OUTBUF	Bank1
3	SPI_0_CLK	Inout	LVCMOS25	N19	☑	ADLIB:BIBUF	Bank2
4	SPI_0_DI	Input	LVCMOS25	N20	☑	ADLIB:INBUF	Bank2
5	SPI_0_DO	Output	LVCMOS25	N21	☑	ADLIB:TRIBUFF	Bank2
6	SPI_0_SS0	Inout	LVCMOS25	N22	☑	ADLIB:BIBUF	Bank2

***Figure 17* • I/O Editor**

3. After updating the I/O Editor, click **Commit and Check**.
4. Close the **I/O Editor** window.
5. Click **Generate Bitstream** as shown in Figure 18 to complete place-and-route and generate the programming file.

***Figure 18* • Generate Bitstream**

Step 3: Programming the SmartFusion2 Board Using FlashPro

1. Connect the FlashPro4 programmer to the J5 connector of the SmartFusion2 Evaluation Kit.
2. Connect the jumpers on the SmartFusion2 Evaluation Kit board as listed in Table 3 on page 18. For more information on jumper locations, refer Appendix B - SmartFusion2 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 3 • **SmartFusion2 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 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 19.

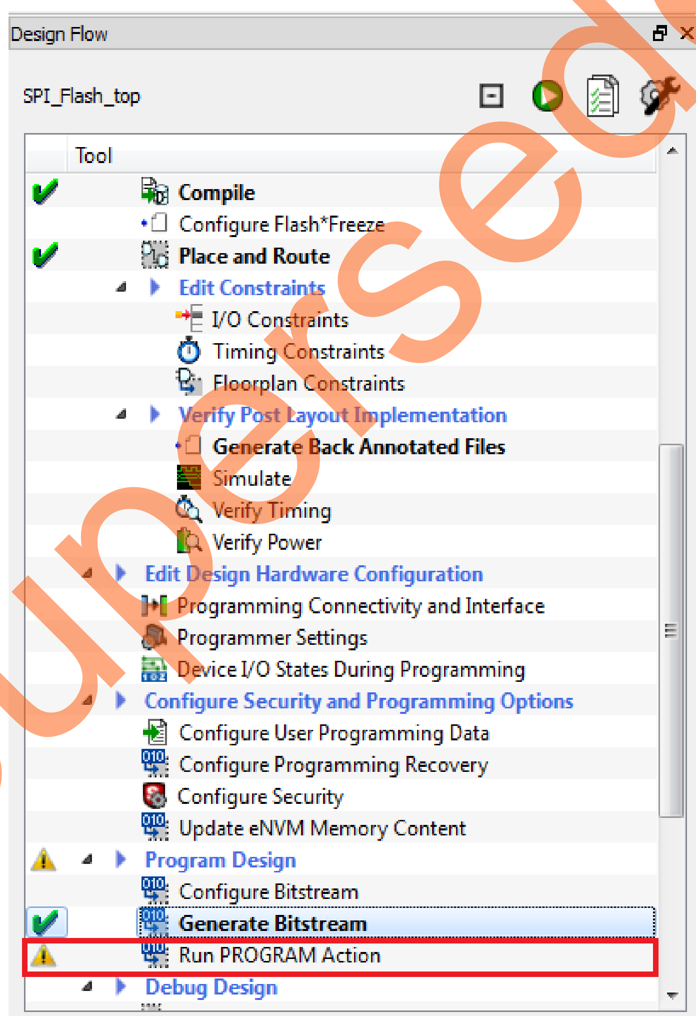


Figure 19 **• Run Programming Action**

Step 4: Building the Software Application using SoftConsole

1. Launch the SoftConsole IDE as shown in Figure 20.

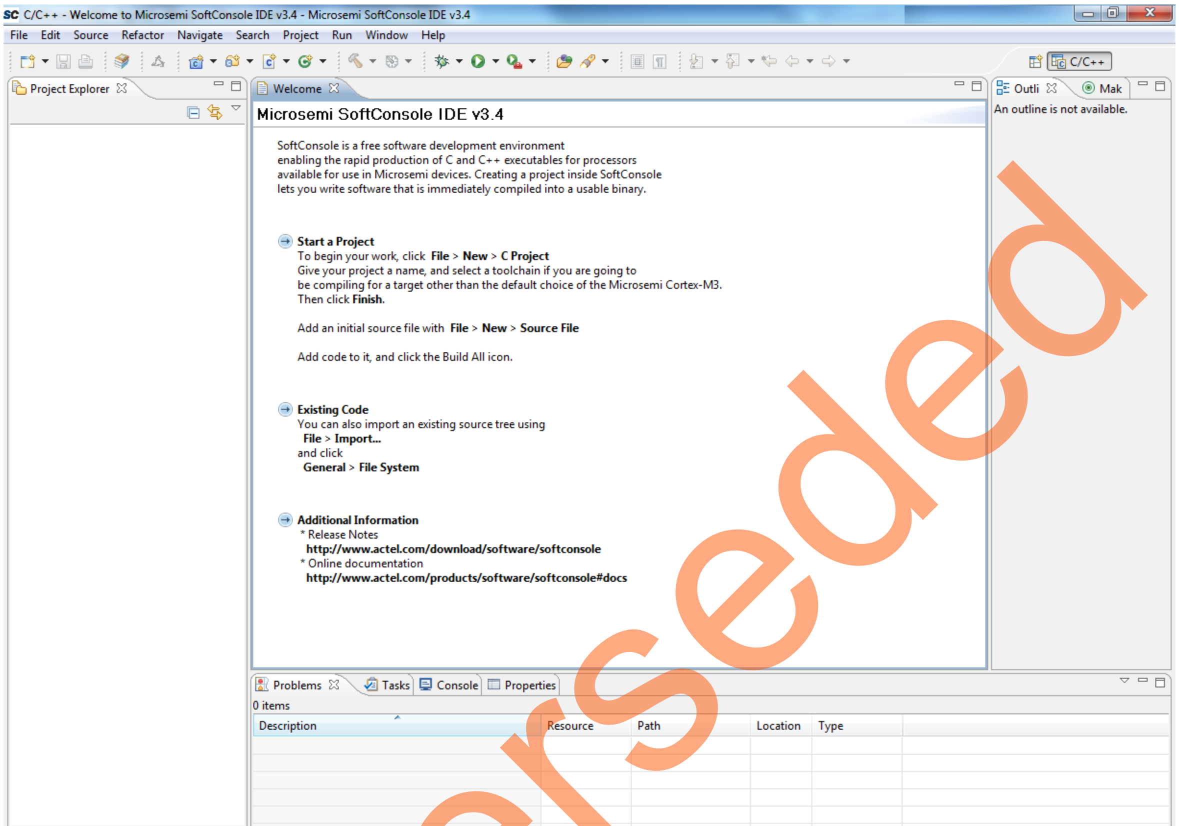


***Figure 20* • Invoking SoftConsole**

2. Right-click **Project Explorer** on the left pane and select **Import** as shown in Figure 21.

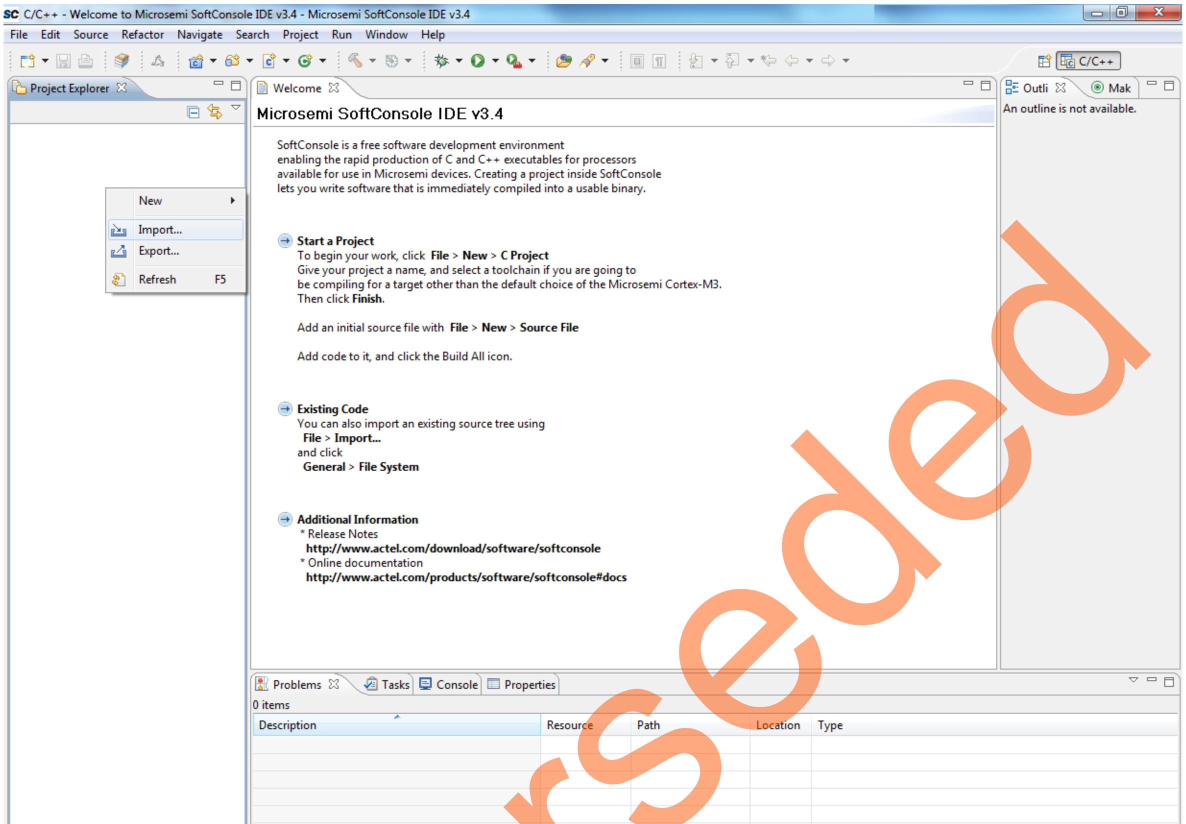


Figure 21 **• SoftConsole IDE - Import Projects**

3. Navigate to **General** > **Select Existing Projects into Workspace** as shown in Figure 22.

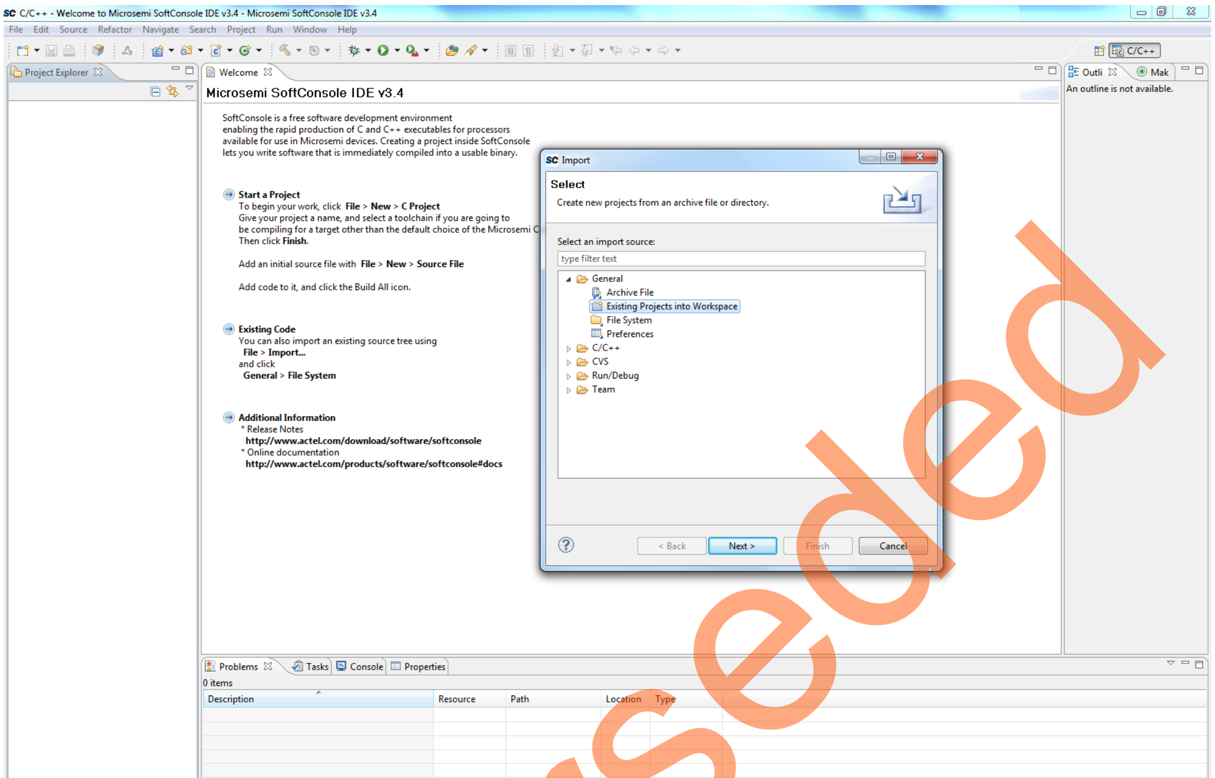


***Figure 22* • SoftConsole IDE- Existing Projects Into Workspace**

The **Import** dialog box is displayed as shown in Figure 23.

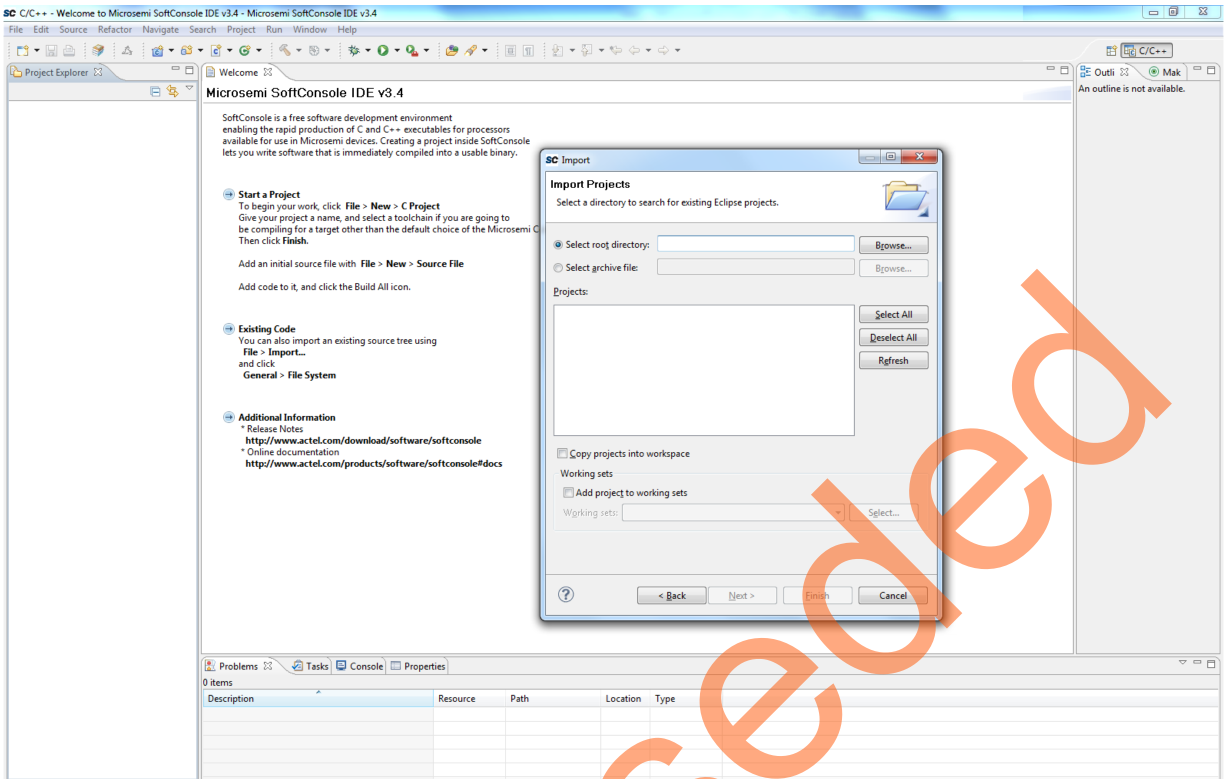


Figure 23 • SoftConsole IDE - Selecting a Directory

4. Click **Browse** to navigate to the **Select root directory** as shown in Figure 24.

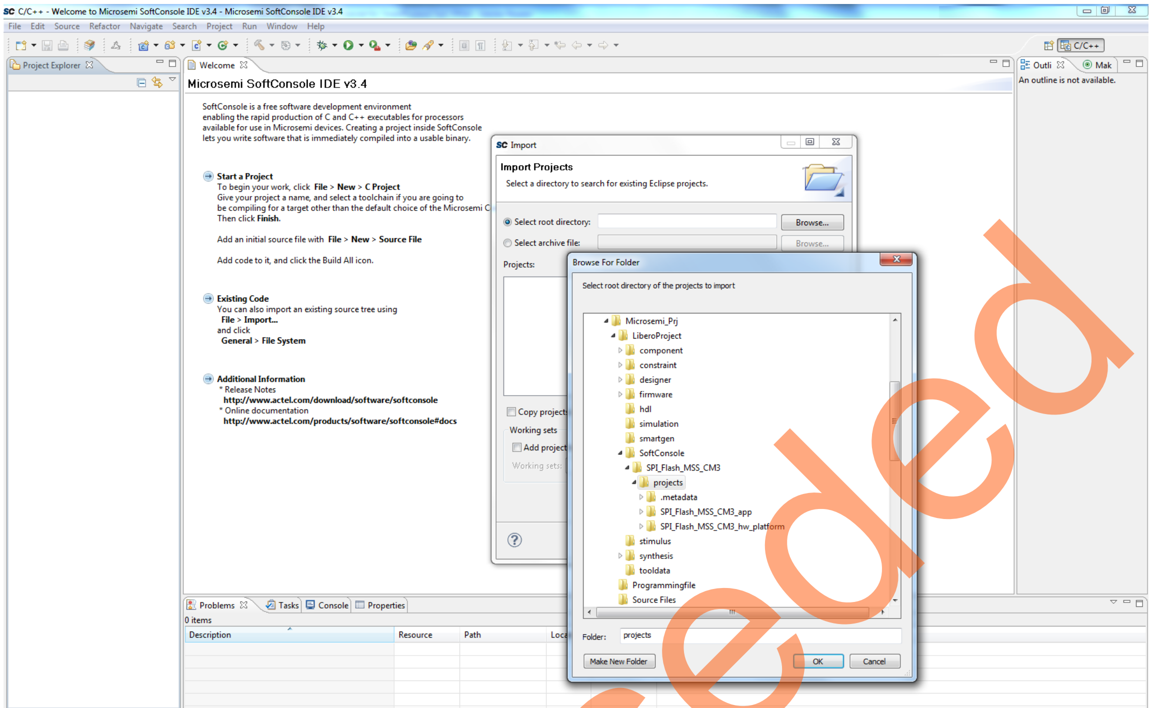


Figure 24 • **SoftConsole IDE - Browse for Folder**

5. Click **Finish**.

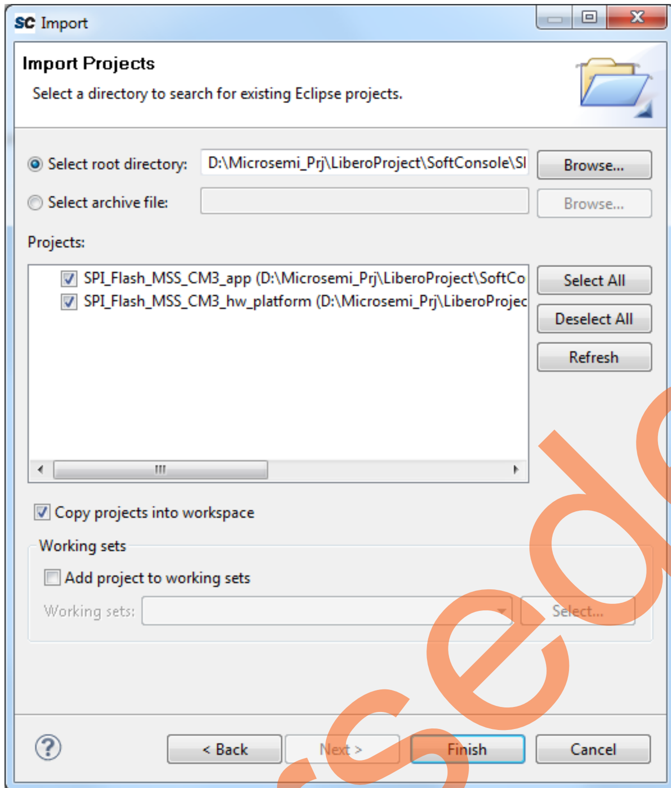


***Figure 25* • Import Project Window**

The SoftConsole workspace is displayed, as shown in Figure 26.

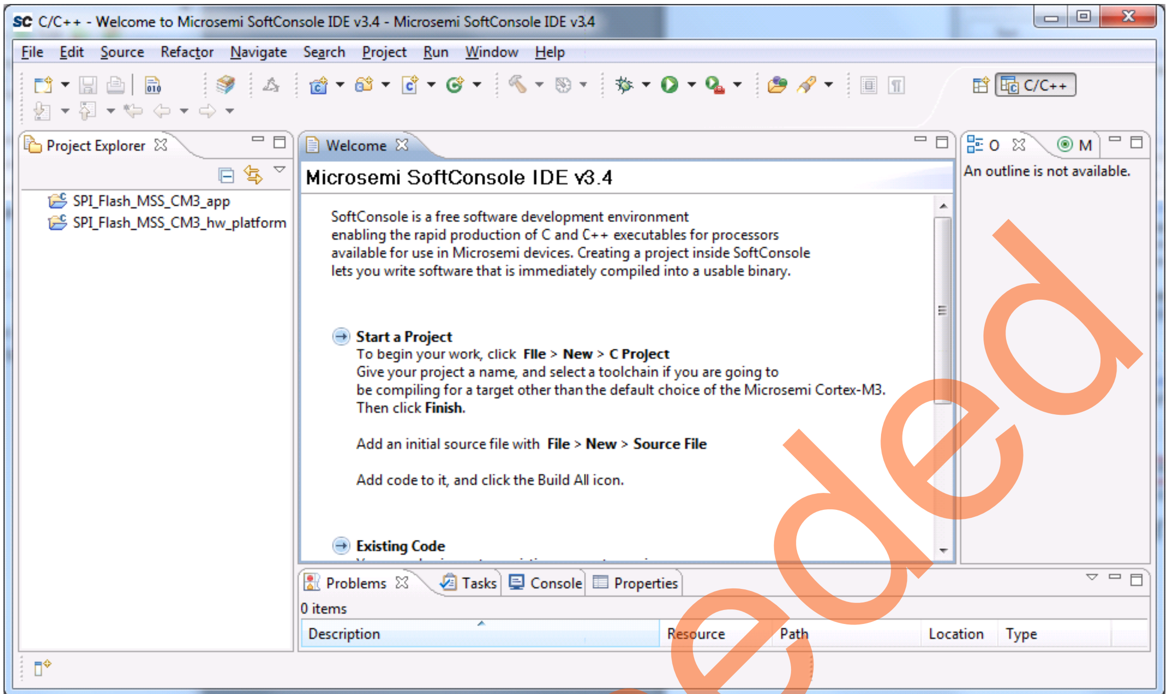


***Figure 26* • SoftConsole Workspace**

6. Browse to the `main.c` file location in the design files folder: *<download_folder>\sf2_spi_flash_sc_liberov11p4_tu_df\Source Files*.
7. Copy the `main.c` file and replace the existing `main.c` file under SPI_Flash_MSS_CM3_app project in the SoftConsole workspace.

The SoftConsole window displays the `main.c` file, as shown in Figure 27.

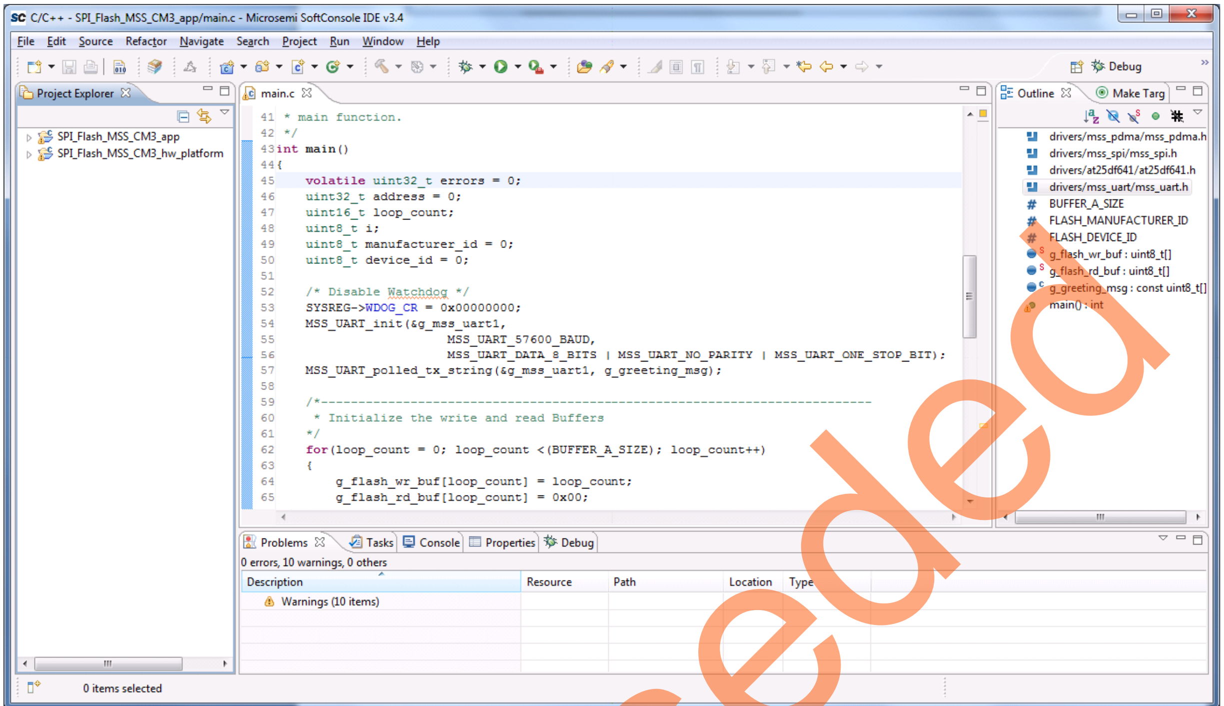


Figure 27 • SoftConsole Workspace main.c file

8. 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_liberov11p4_tu_df\SPI_Flash_Drivers*.
9. Copy the **winbondflash** folder to the drivers folder of SPI_Flash_MSS_CM3_hw_platform project in the SoftConsole workspace, as shown in Figure 28.

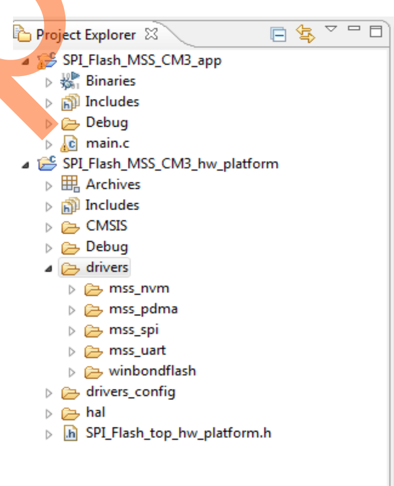


Figure 28 • Project Explorer window

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.

10. Right-click the **SPI_Flash_MSS_CM3_hw_platform** in Project Explorer window of SoftConsole project and select **Properties** as shown in Figure 29.

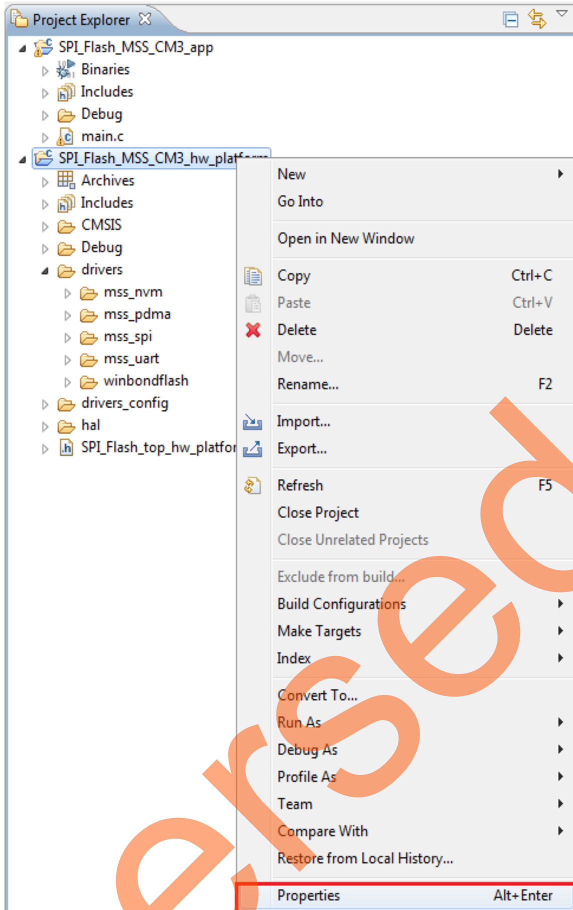


***Figure 29* • Project Explorer window - Properties**

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

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

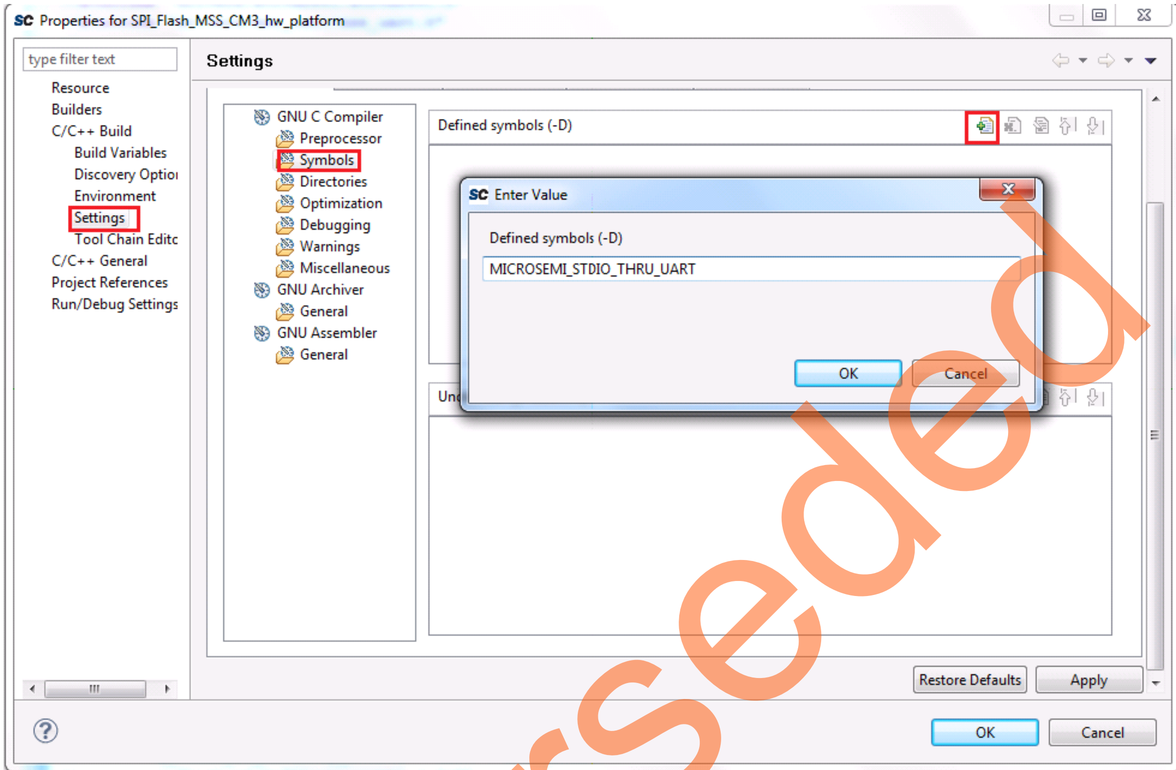


Figure 30 **• SPI_Flash_MSS_CM3_HW_Platform Properties window**

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

15. Perform a build by selecting **Project** > **Clean**. Leave the default settings in the **Clean** dialog box and click **OK**, as shown in Figure 31.

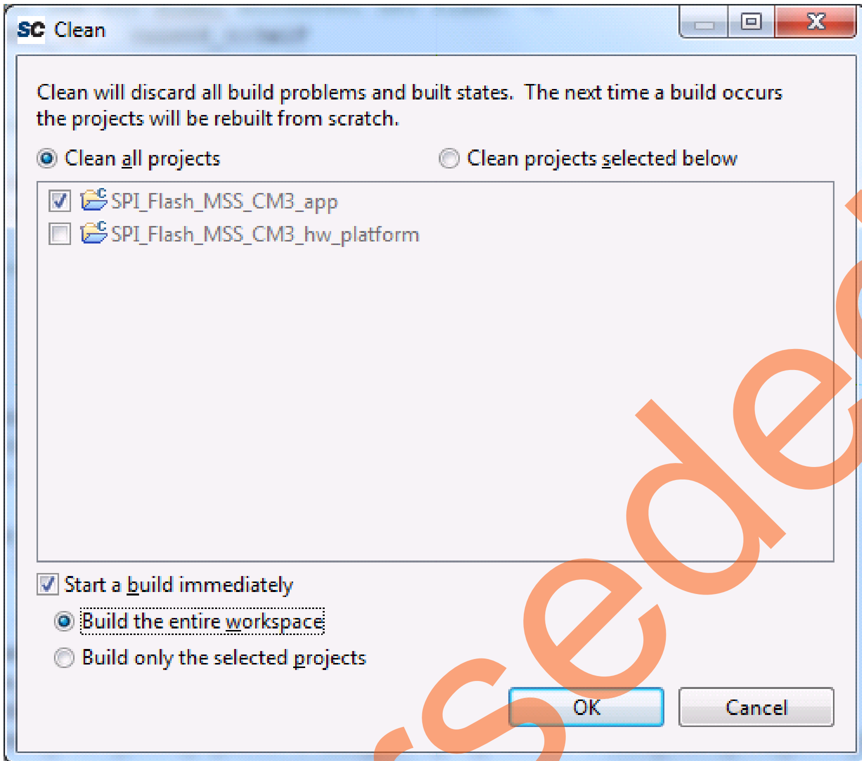


***Figure 31* • Settings for a clean build**

Note: Ensure that there are no errors.

Step 5: 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 32 shows an example Device Manager window.

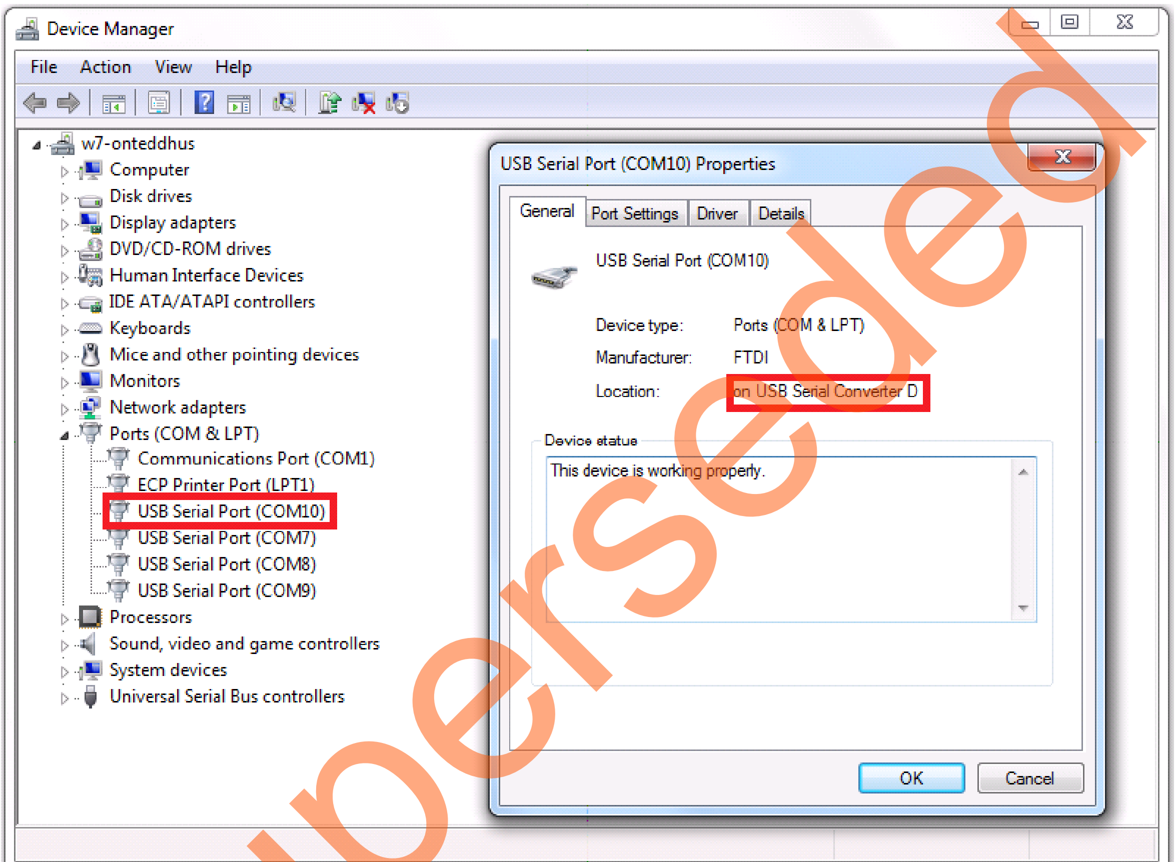


Figure 32 • ***Device Manager Window***

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 6: 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 33.

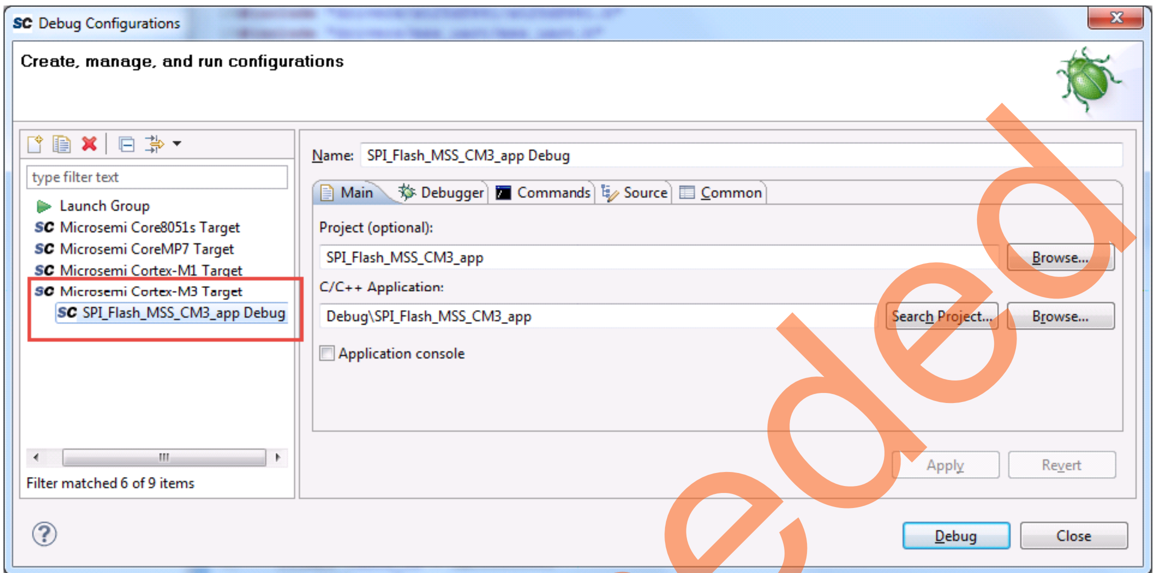


Figure 33 **• Debug Configurations**

2. Ensure that the following values are filled in the corresponding fields:
   – Name: SPI_Flash_MSS_CM3_app Debug
   – Project (optional): SPI_Flash_MSS_CM3_app
   – C/C++ Application: Debug\ SPI_Flash_MSS_CM3_app
3. Click **Debug**.
4. On the **Confirm Perspective Switch** window, click **Yes**, as shown in Figure 34.

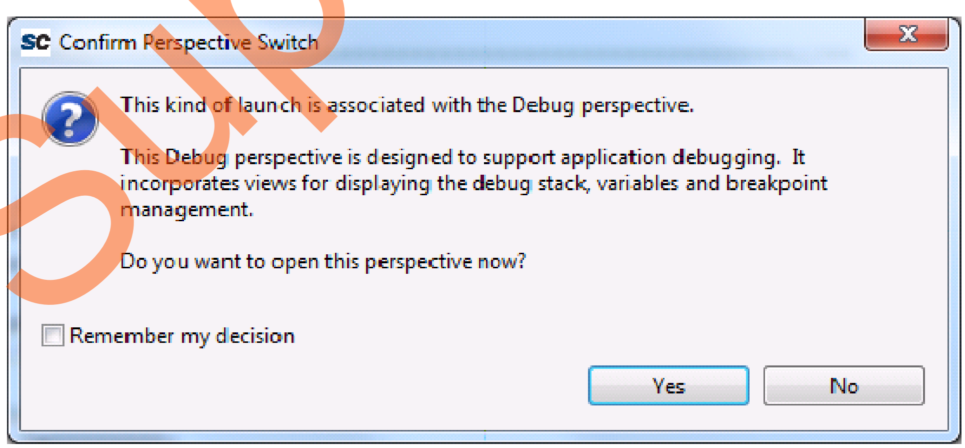


Figure 34 **• Confirm Perspective Switch**

5. The **SoftConsole Debugger Perspective** window is displayed, as shown in Figure 35.

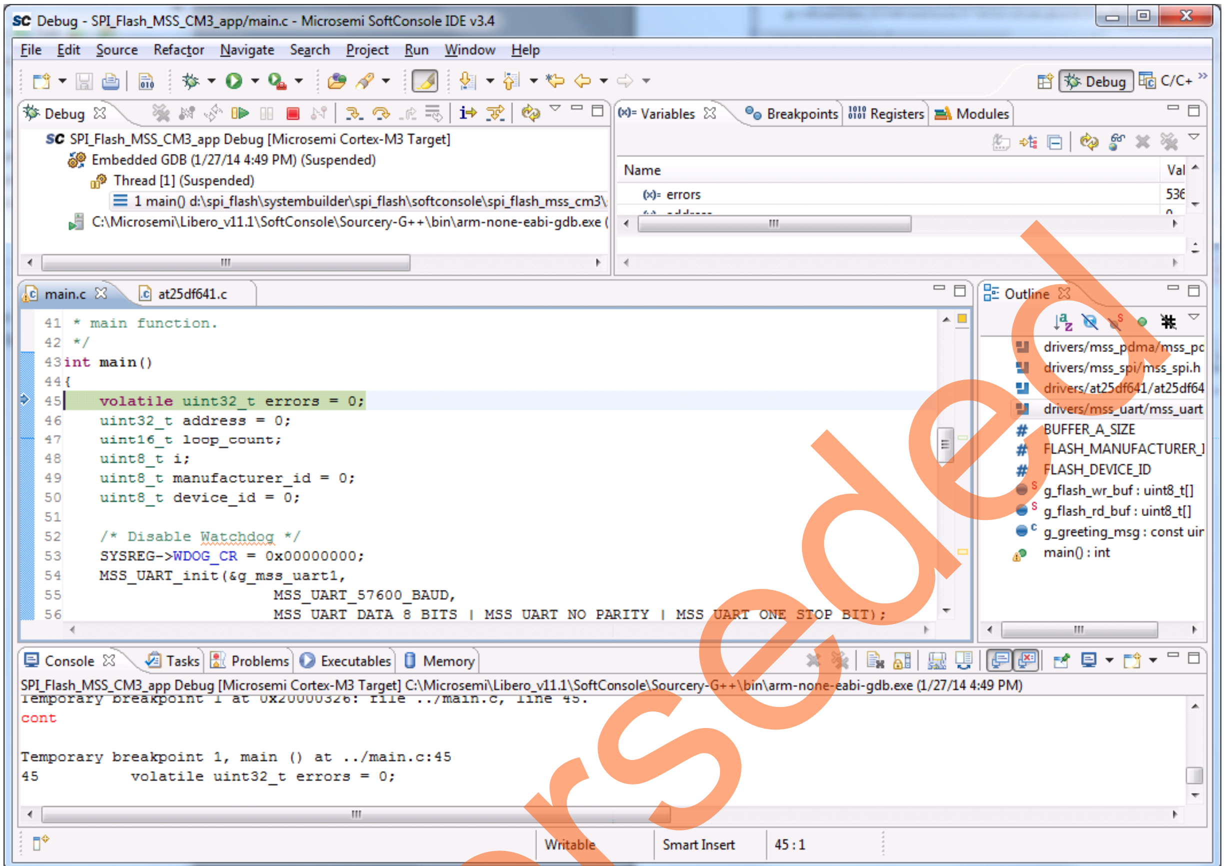


***Figure 35* • SoftConsole Debugger Perspective**

6. Run the application by clicking **Run > Resume**. Read data from SPI Flash is displayed along with a greeting message on the HyperTerminal, as shown in Figure 36.

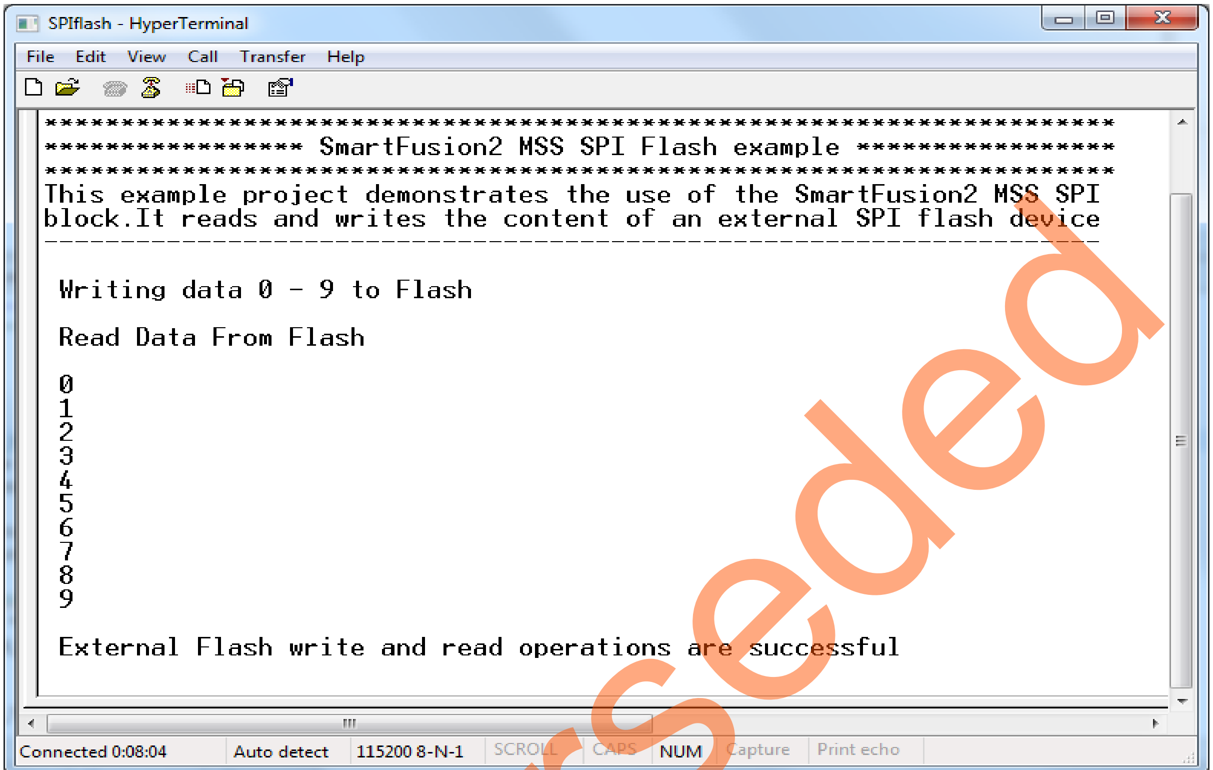


***Figure 36* • HyperTerminal Window**

7. Terminate execution of the code by choosing **Run > Terminate**.
8. Launch the debug session by selecting **Debug Configurations** from the **Run** menu of SoftConsole.

9. Click the **Registers** tab to view the values of the ARM® Cortex™-M3 processor internal registers, as shown in Figure 37.

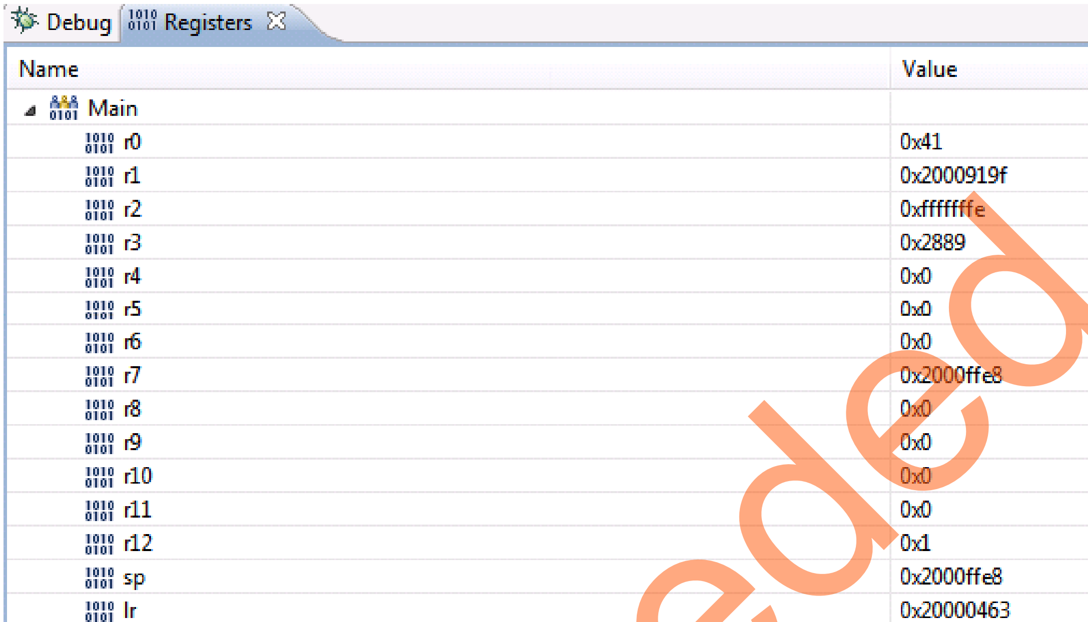


Figure 37 **• Values of Cortex-M3 Internal Registers**

10. Click the **Variables** tab to view the values of variables in the source code, as shown in Figure 38.

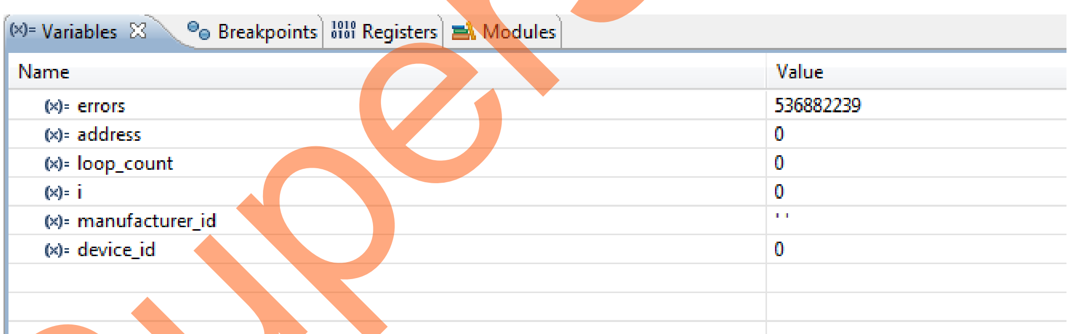
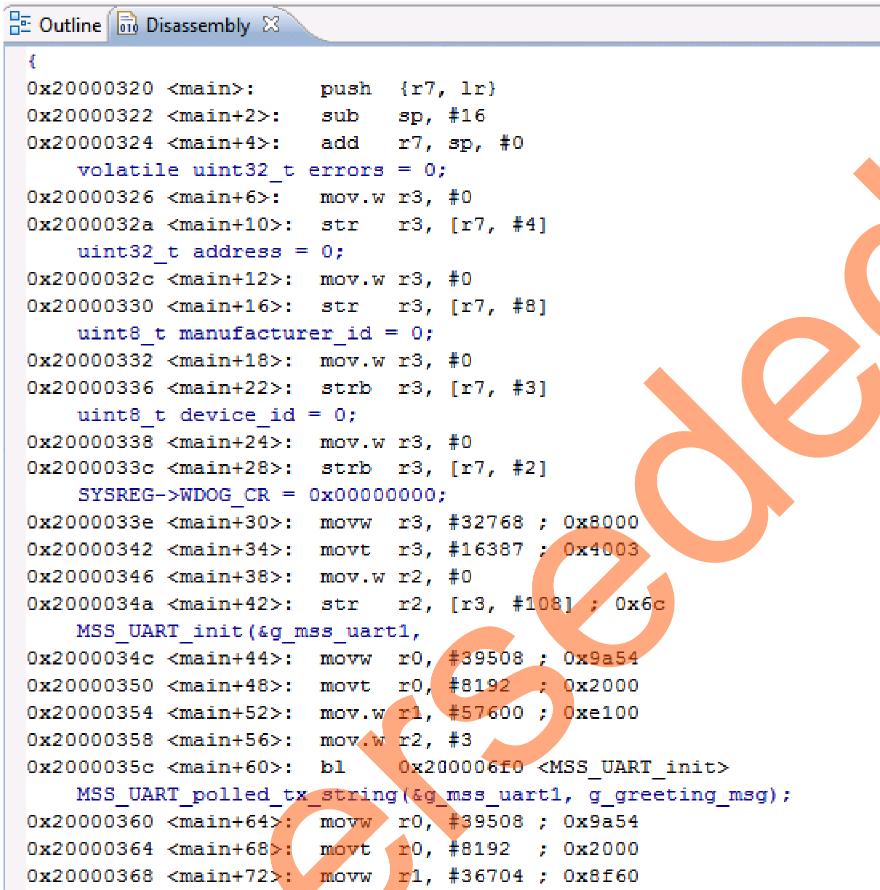


Figure 38 **• Values of the Variables in the Source Code**

11. 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 39.




```
{
0x20000320 <main>:      push  {r7, lr}
0x20000322 <main+2>:    sub   sp, #16
0x20000324 <main+4>:    add   r7, sp, #0
    volatile uint32_t errors = 0;
0x20000326 <main+6>:    mov.w r3, #0
0x2000032a <main+10>:   str   r3, [r7, #4]
    uint32_t address = 0;
0x2000032c <main+12>:   mov.w r3, #0
0x20000330 <main+16>:   str   r3, [r7, #8]
    uint8_t manufacturer_id = 0;
0x20000332 <main+18>:   mov.w r3, #0
0x20000336 <main+22>:   strb  r3, [r7, #3]
    uint8_t device_id = 0;
0x20000338 <main+24>:   mov.w r3, #0
0x2000033c <main+28>:   strb  r3, [r7, #2]
    SYSREG->WDOG_CR = 0x00000000;
0x2000033e <main+30>:   movw  r3, #32768 ; 0x8000
0x20000342 <main+34>:   movt  r3, #16387 ; 0x4003
0x20000346 <main+38>:   mov.w r2, #0
0x2000034a <main+42>:   str   r2, [r3, #108] ; 0x6c
    MSS_UART_init(&g_mss_uart1,
0x2000034c <main+44>:   movw  r0, #39508 ; 0x9a54
0x20000350 <main+48>:   movt  r0, #8192  ; 0x2000
0x20000354 <main+52>:   mov.w r1, #57600 ; 0xe100
0x20000358 <main+56>:   mov.w r2, #3
0x2000035c <main+60>:   bl    0x200006f0 <MSS_UART_init>
    MSS_UART_polled_tx_string(&g_mss_uart1, g_greeting_msg);
0x20000360 <main+64>:   movw  r0, #39508 ; 0x9a54
0x20000364 <main+68>:   movt  r0, #8192  ; 0x2000
0x20000368 <main+72>:   movw  r1, #36704 ; 0x8f60
```

***Figure 39* • Assembly Level Instructions**

12. 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.
13. Click **Instruction Stepping** ( ) and perform **Step Into** operations. Observe that **Step Into** executes a single line of assembly code.
14. 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.
15. Add breakpoints in the application to force the code to halt, then single-step and observe the instruction sequence.
16. When debug process is finished, terminate execution of the code by choosing **Run > Terminate**.
17. Close Debug Perspective by selecting **Close Perspective** from the Window menu.
18. Close SoftConsole using **File > Exit**.
19. 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 Evaluation Kit board.

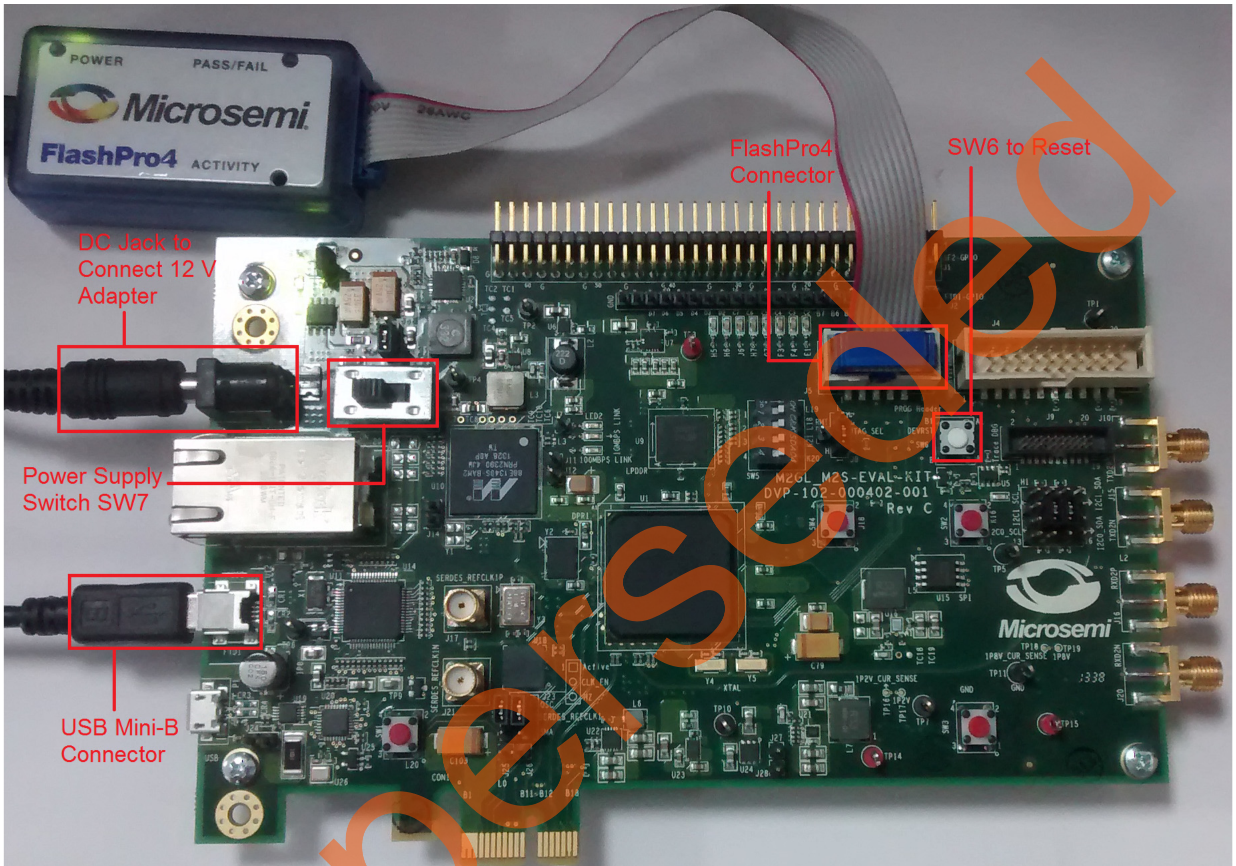


Figure 1 • **SmartFusion2 Evaluation Kit Setup**

Appendix B - SmartFusion2 Evaluation Kit Board Jumper Locations

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

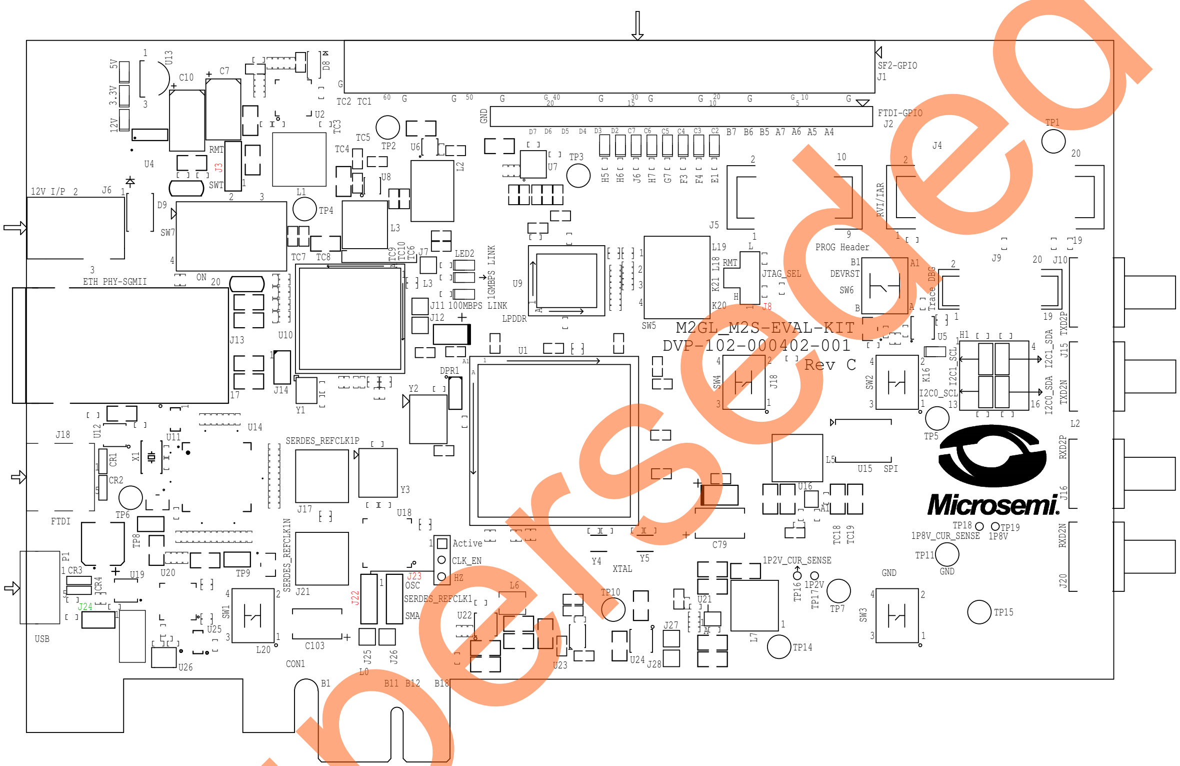


Figure 1 • **SmartFusion2 Evaluation Kit Board Jumper Locations**

Note:

- 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.

List of Changes

The following table lists the critical changes that were made in each revision

Date	Changes	Page
Revision 2 (October 2014)	Updated the document for Libero v11.4 software release (SAR 61627).	NA
Revision 1 (April 2014)	First Release.	NA
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 Customer Support website (www.microsemi.com/soc/support/search/default.aspx) for more information and support. Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on the website.

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

50200546-2/10.14