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

Libero SoC and Keil uVision Flow Tutorial for SmartFusion2 SoC FPGA

Microsemi®

Table of Contents

Accessing Serial Flash Memory using SPI Interface - Libero SoC and Keil uVision Flow Tutorial for SmartFusion2 SoC FPGA

Introduction

The Libero® System-on-Chip (SoC) software generates firmware projects using Keil, SoftConsole, and IAR tools. This tutorial describes the process to build a Keil uVision application that can be implemented and validated using the SmartFusion®2 system-on-chip (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 SoftConsole Flow Tutorial for SmartFusion2 SoC FPGA
- Accessing Serial Flash Memory using SPI Interface - Libero SoC and IAR Embedded Workbench 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 Keil uVision from Libero SoC
- Compile application code
- Debug and run code using Keil uVision

Tutorial 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
Keil debugger	-
Host PC or Laptop	Any 64-bit Windows Operating System
Software Requirements	
Libero SoC • Keil uVision 5 • FlashPro programming software v11.4	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_keil_liberov11p4_tu_df

The demo design files include:

- LiberoProject
- Programmingfile
- Source Files
- SPI_Flash_Drivers
- Readme

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.

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.

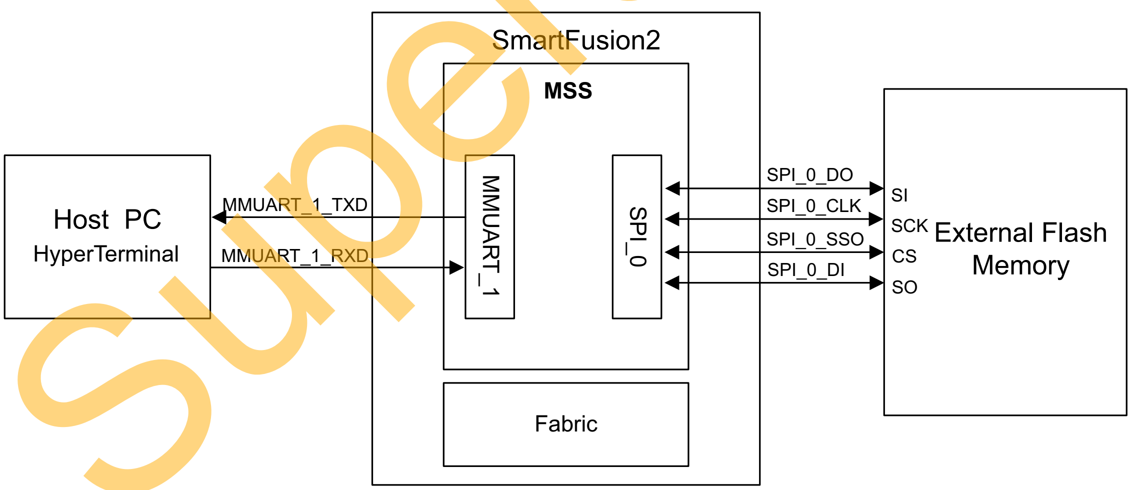

Figure 1 • **SPI Flash Interfacing Block Diagram**

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.

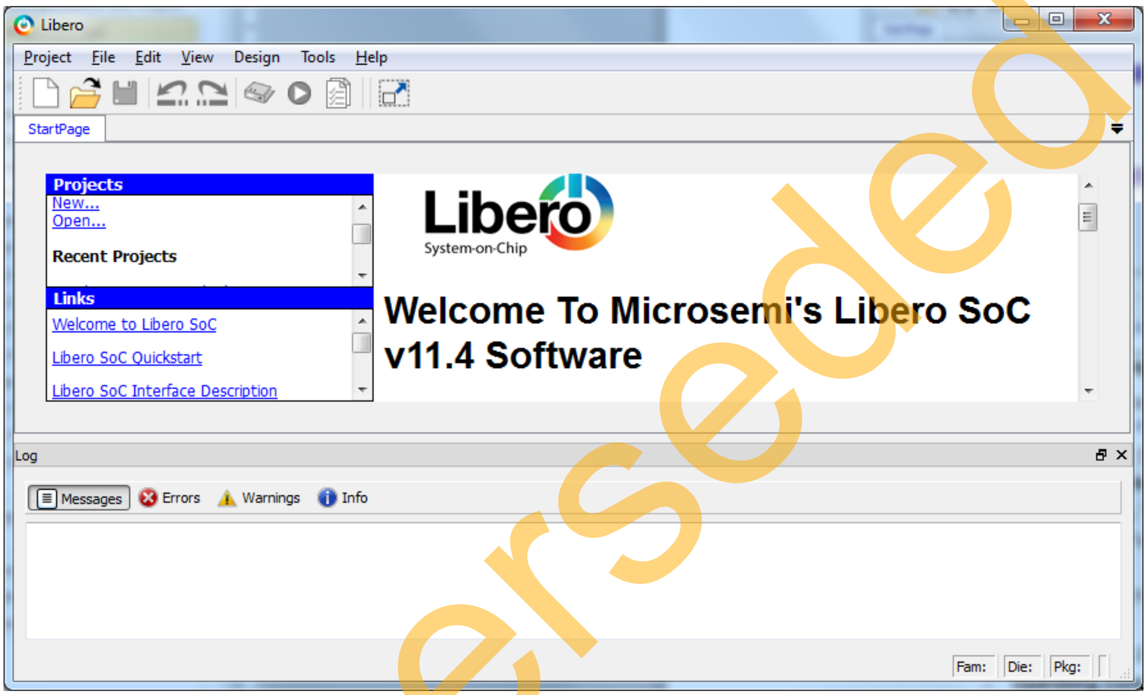

***Figure 2* • Libero SoC Project Manager**

3. Enter the information as required for the new project and the device in the **New Project** dialog box as shown in Figure 3 on page 6.
   – 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.

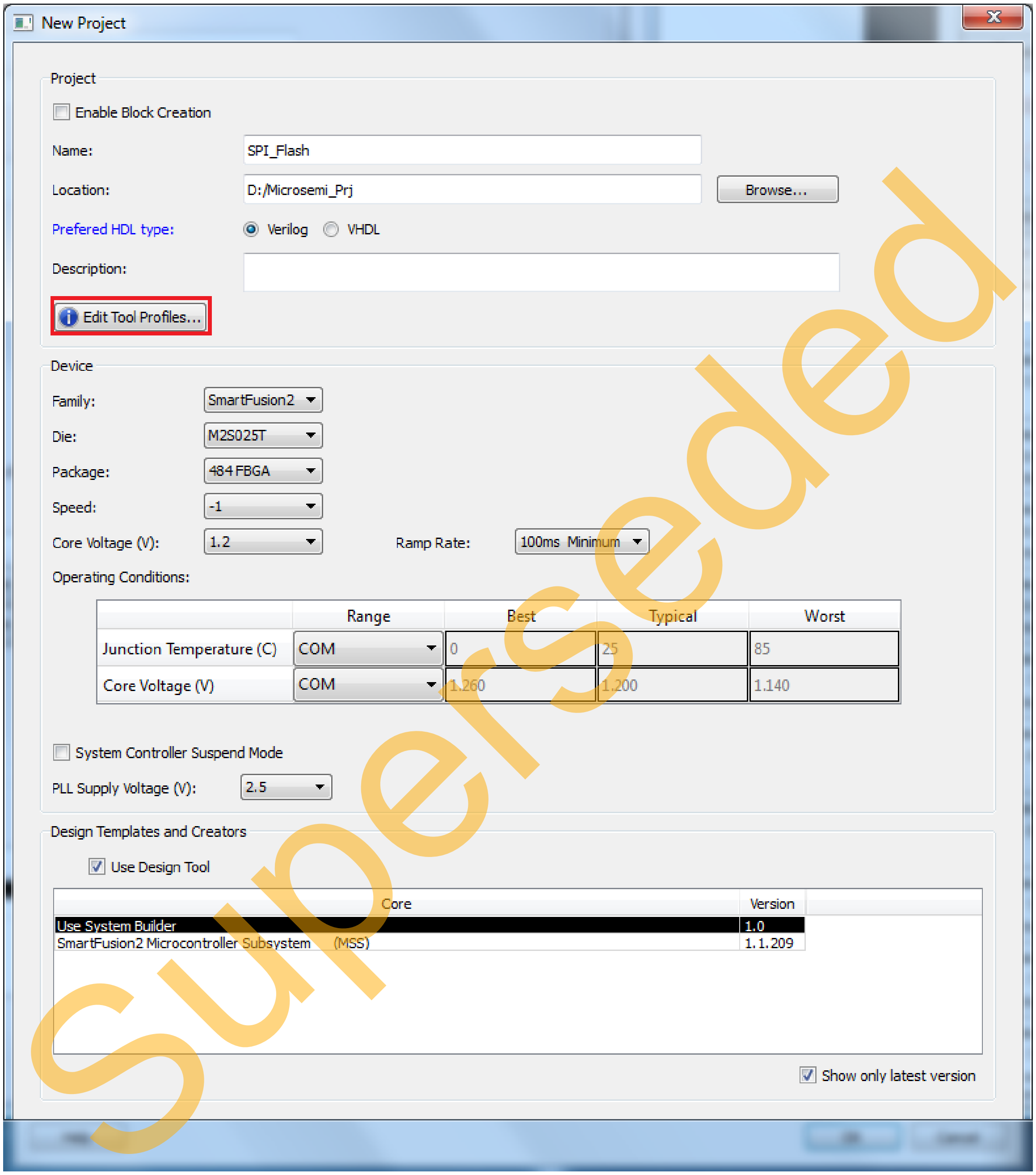

***Figure 3* • New Project Dialog Box**

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 6) displays the **Tool Profiles** window as shown in Figure 4. Check the following tool settings:
   – Software IDE: Keil
   – Synthesis: Synplify Pro ME I-2013.09M-SP1
   – Simulation: ModelSim ME 10.2c
   – Programming: FlashPro 11.4

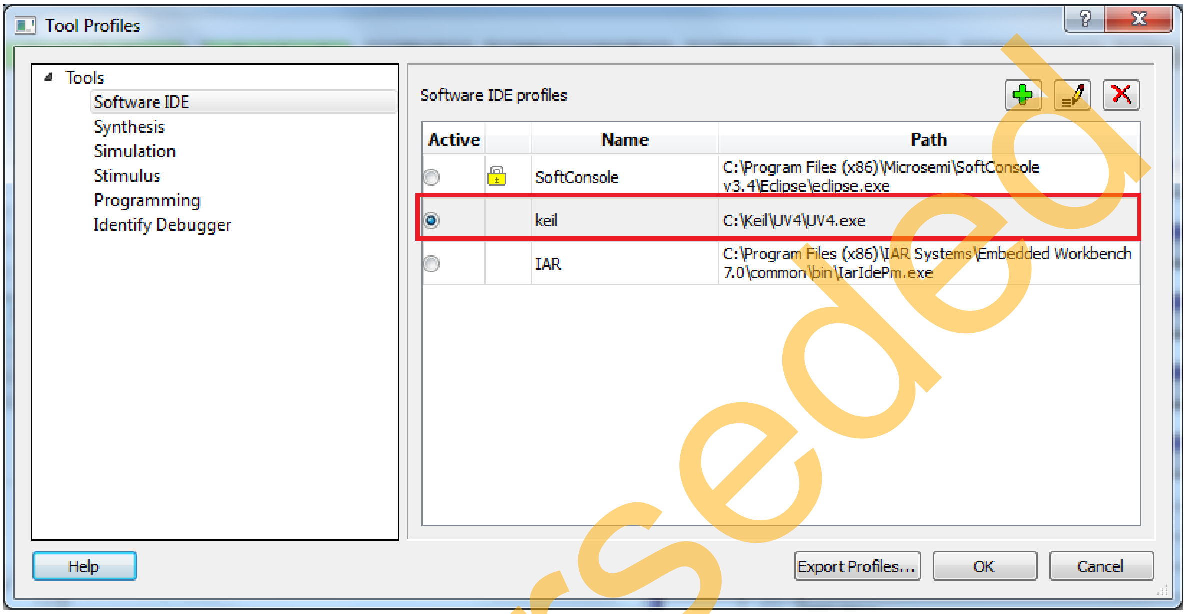

***Figure 4* • Tool Profiles**

6. Click **OK** on the **Tool Profiles** window.
7. Click **OK** on the **New Project** window. The **System Builder** dialog box is displayed.
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.

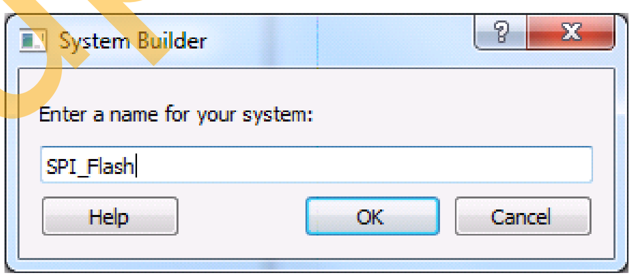

***Figure 5* • Create New System Builder Dialog Box**

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

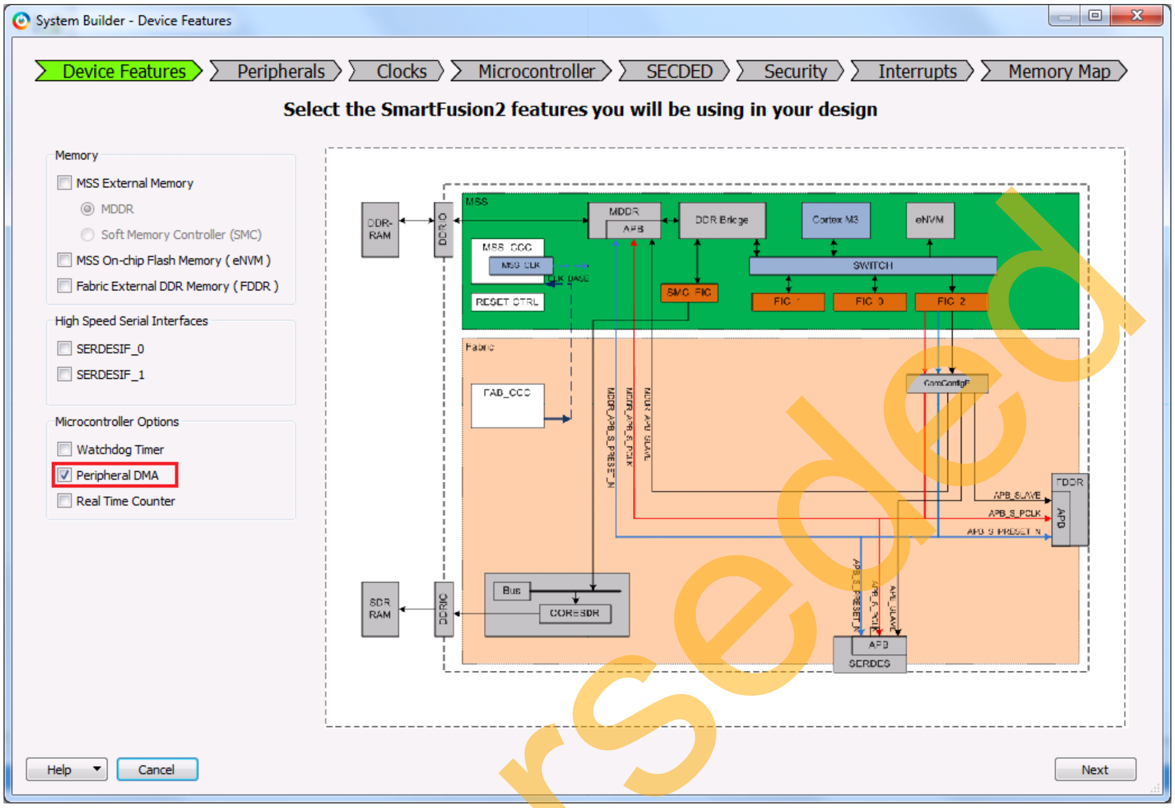

***Figure 6* • System Builder – Device Features Page**

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.

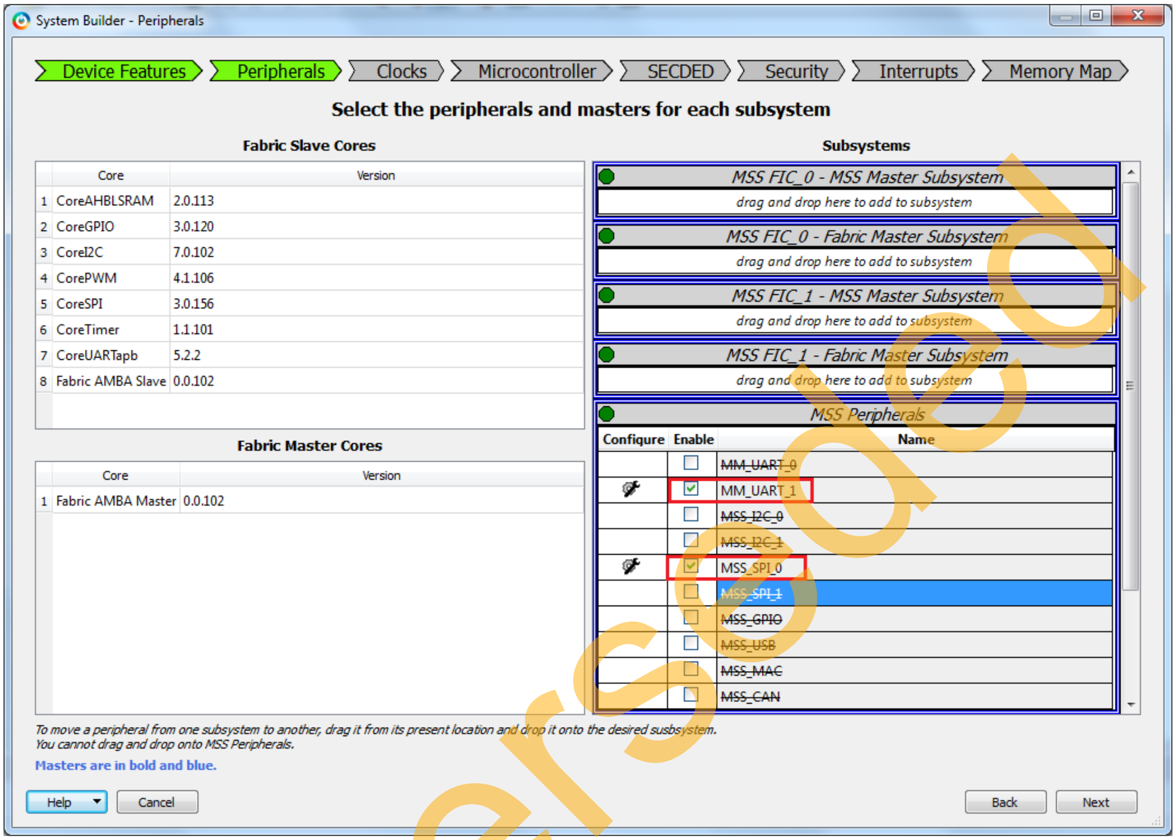

***Figure 7* • System Builder Configurator – Peripherals Page**

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

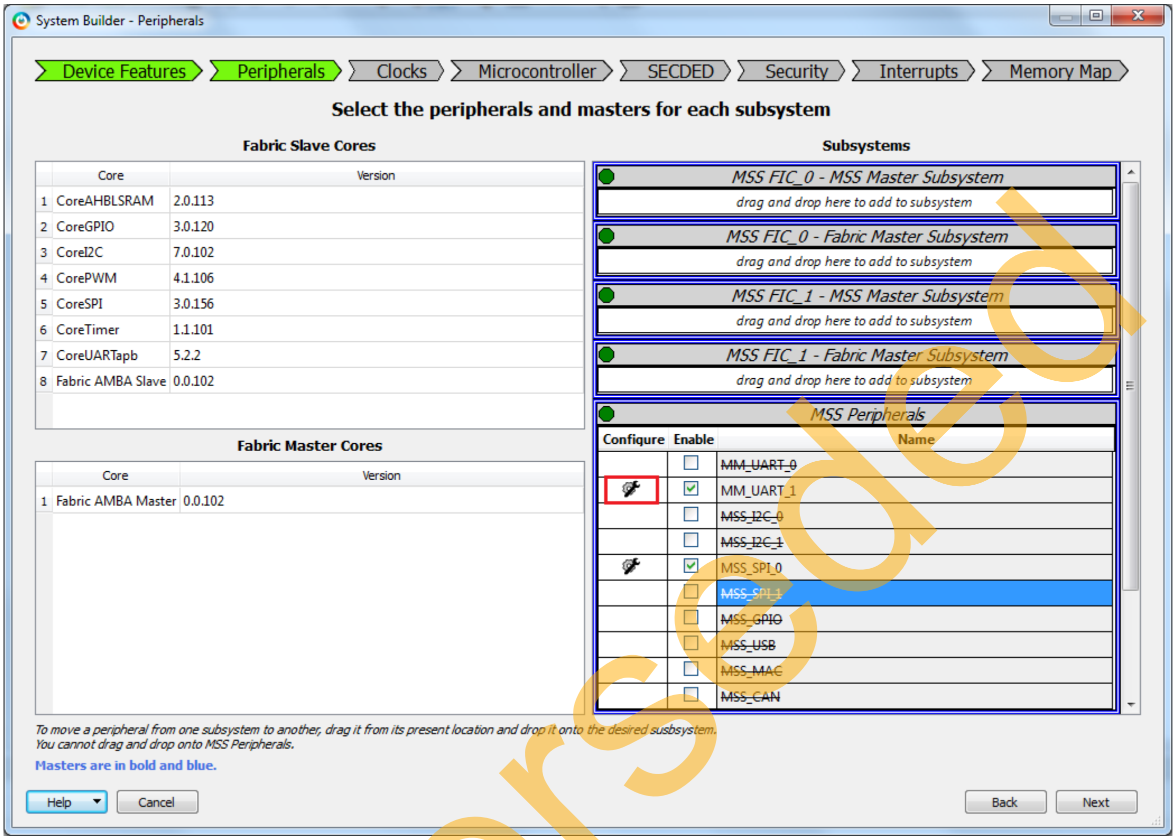

***Figure 8* • System Builder – Peripherals Page**

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

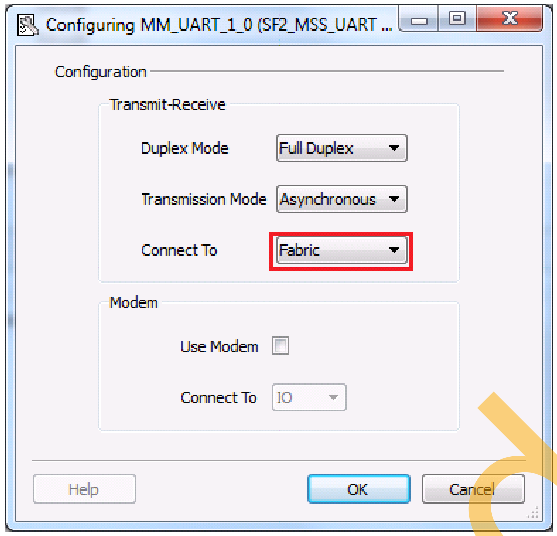

***Figure 9* • Configuring MM_UART_1**

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.

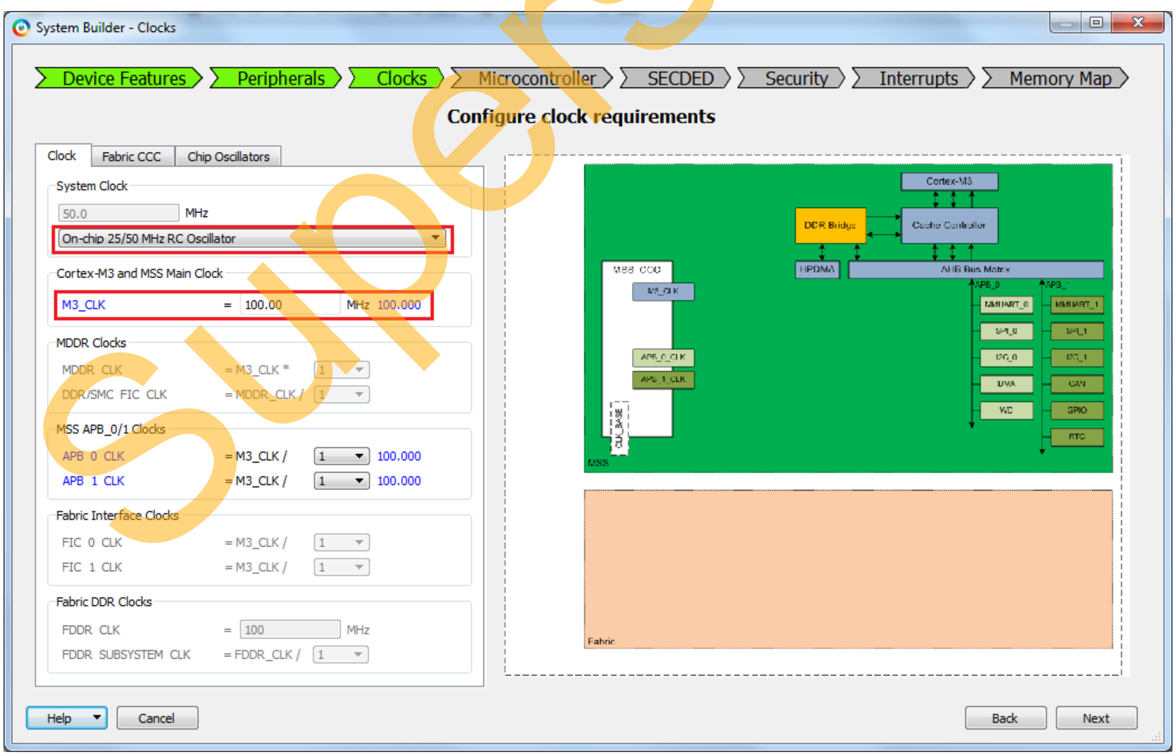

***Figure 10* • System Builder – Clocks Page**

14. Click **Next**. The **System Builder – Microcontroller** page is displayed. Leave the default settings.
15. Click **Next**. The **System Builder – SECDED** page is displayed. Leave the default settings.
16. Click **Next**. The **System Builder – Security** page is displayed. Leave the default settings.
17. Click **Next**. The **System Builder – Interrupts** page is displayed. Leave the default settings.
18. Click **Next**. The **System Builder – Memory Map** page is displayed. Leave the default settings.
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.

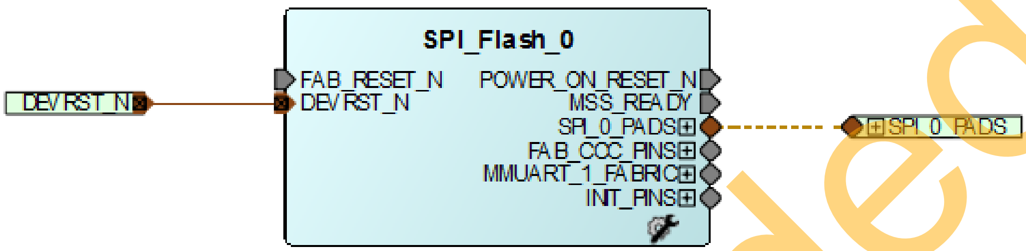

Figure 11 • **System Builder Generated System**

Connecting Components in SPI_Flash_0 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_0 is displayed as shown in Figure 12.

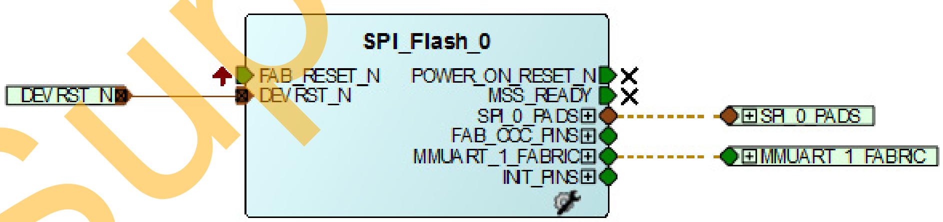

Figure 12 • **SPI_Flash_0 Design**

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 clear all the drivers, except CMSIS, MMUART, PDMA, and SPI as shown in Figure 13.

Note: Select the latest version of the drivers.

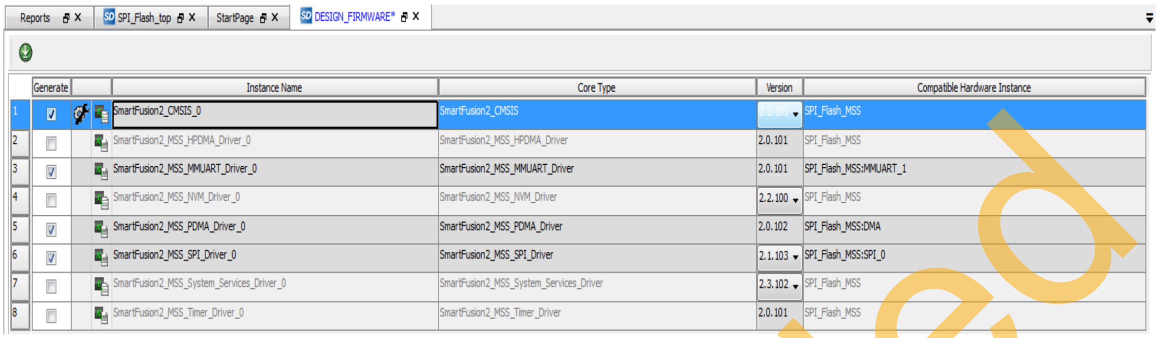

Figure 13 • **Configuring Firmware**

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

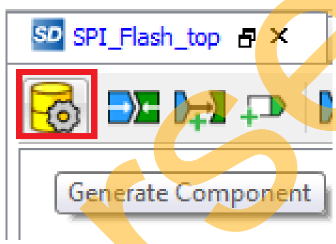

Figure 14 • **Generate Component**

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.

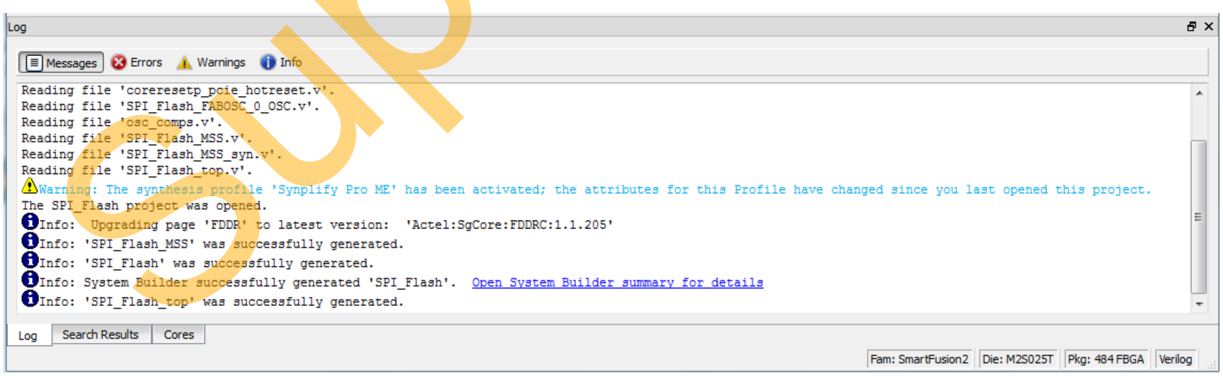

Figure 15 • **Log Window**

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**.

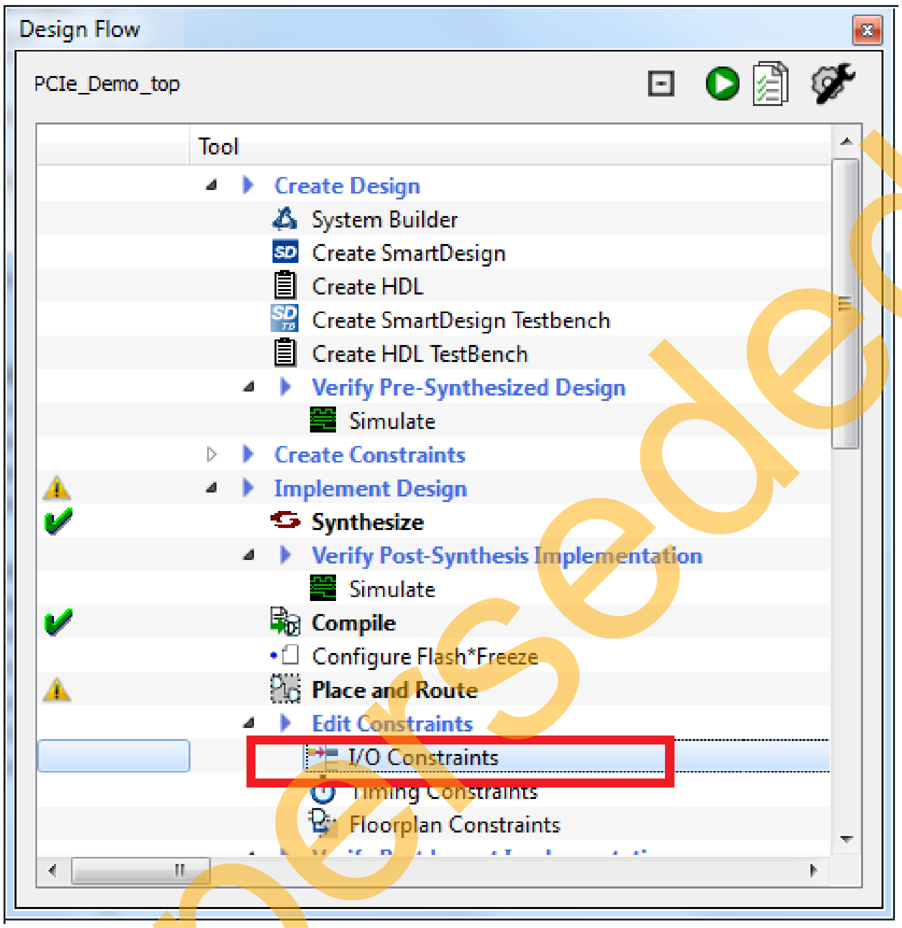

***Figure 16* • I/O Constraints**

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.

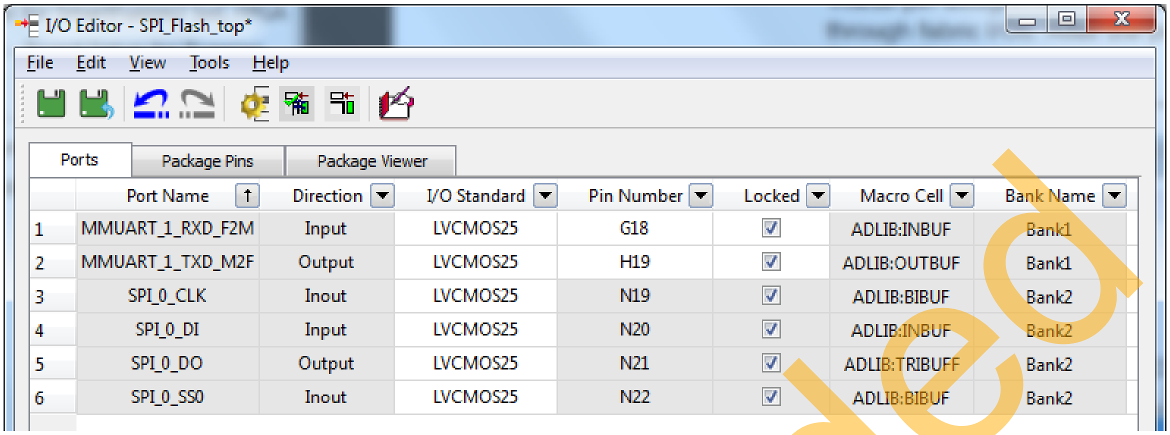

Figure 17 **• I/O Editor**

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. For more information on jumper locations, refer to the "Appendix C- SmartFusion2 Evaluation Kit Board Jumper Locations" on page 45.

   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, 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 B - Board Setup for Programming the Tutorial" on page 44 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.

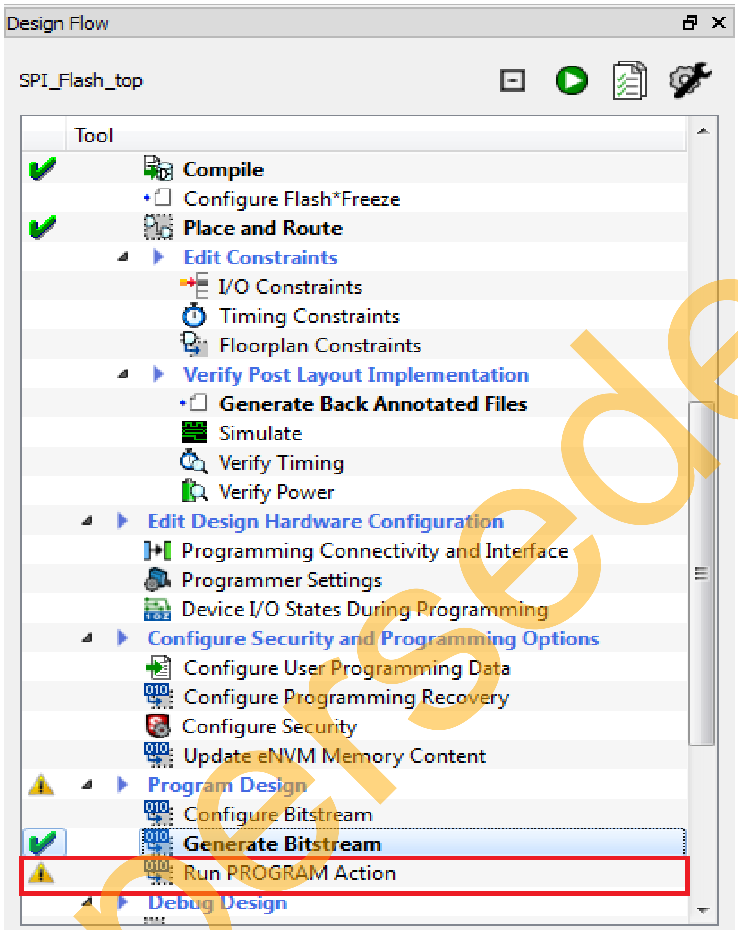

***Figure 19* • Run Program Action**

After successful programming, the SmartFusion2 Evaluation Kit is ready for running and debugging the Keil application through ULINK-ME Debugger.

Step 4: Building the Software Application Using Keil uVision 5 IDE

1. Launch the Keil IDE, as shown in Figure 20.

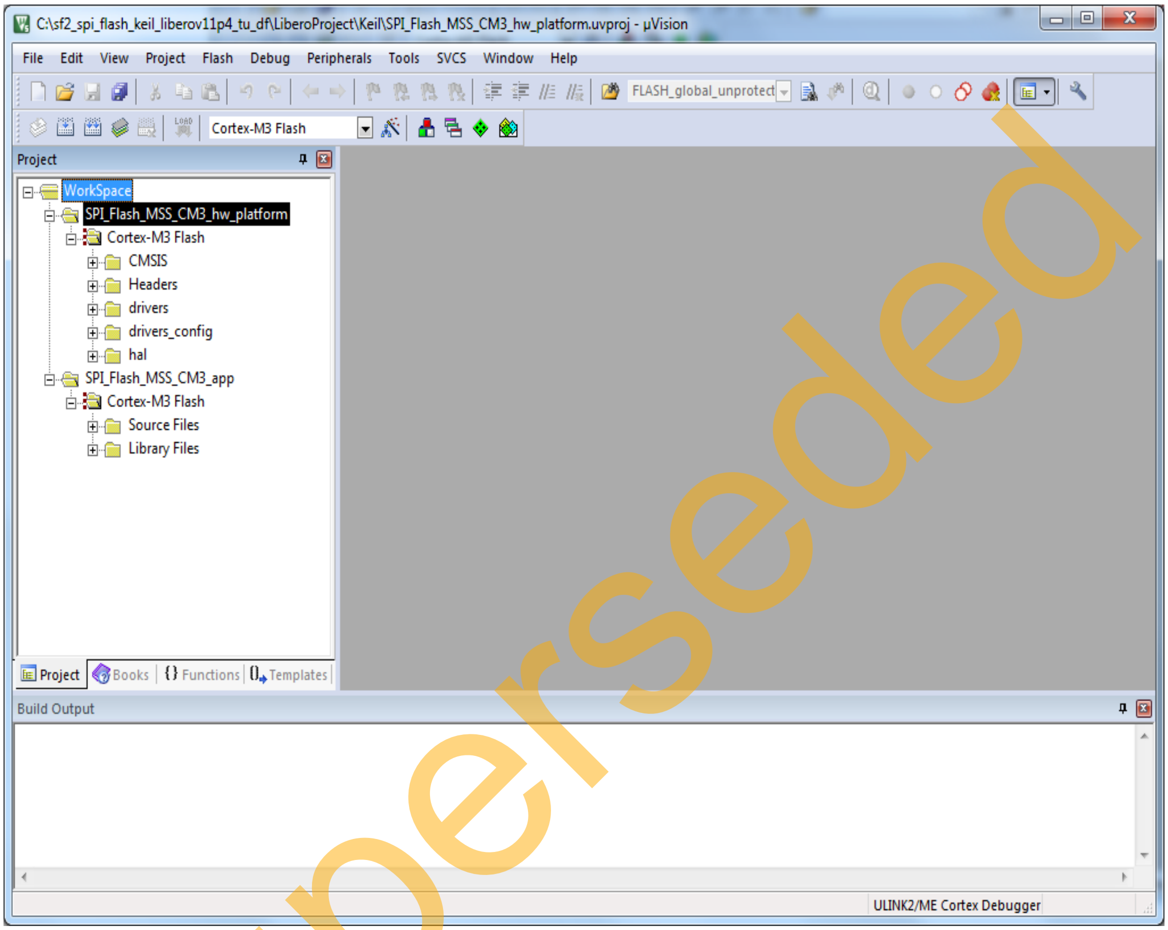

Figure 20 **• uVision Workspace**

2. Browse to the `main.c` file location in the design files folder: *<download_folder>/sf2_spi_flash_keil_liberov11p4_tu_df\SourceFiles*.
3. Copy the `main.c` file and replace the existing `main.c` file under SPI_Flash_MSS_CM3_0_app project in the uVision workspace.

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

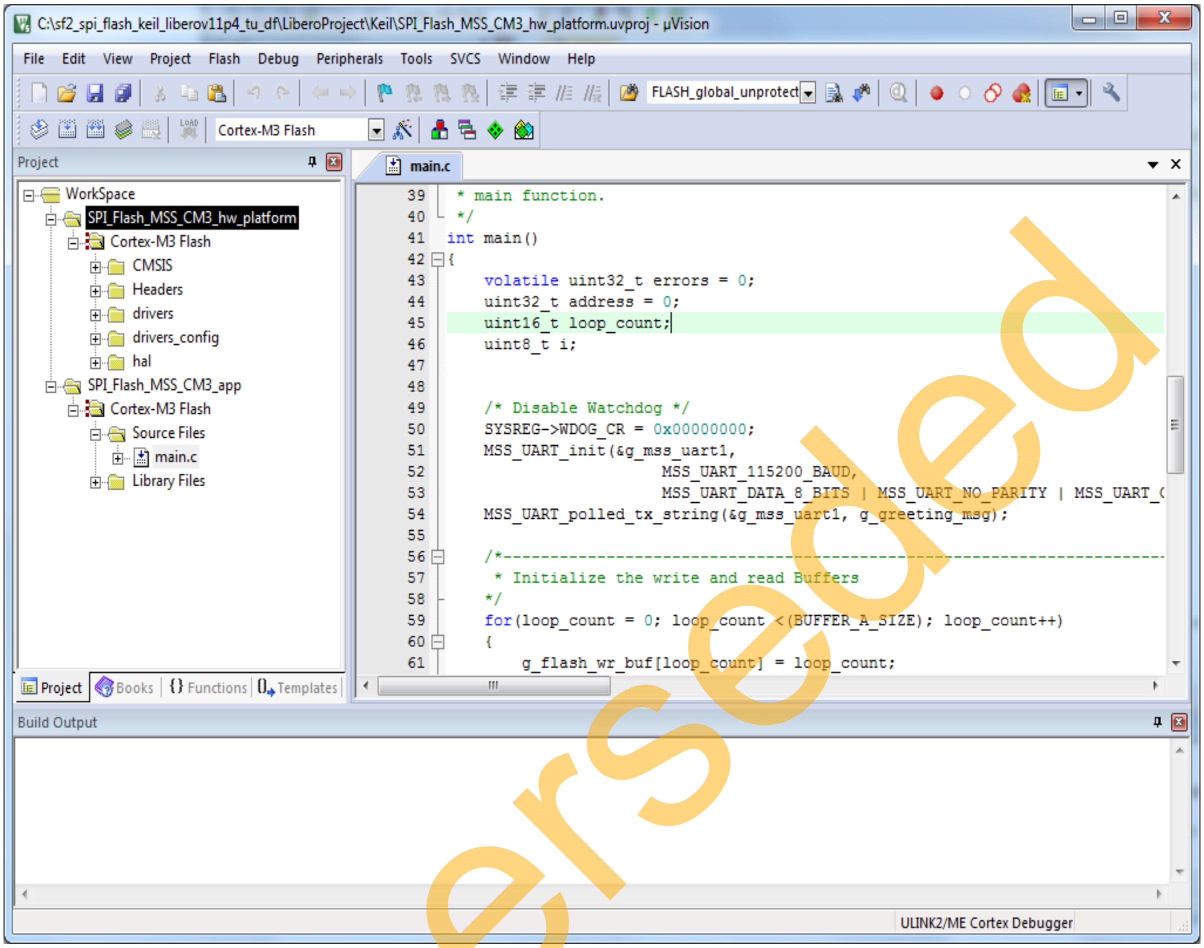

***Figure 21* • uVision Workspace main.c file**

4. winbondflash drivers are not included in the Libero generated uVision workspace. To include the drivers in the uVision workspace, browse to the location of the winbond flash drivers in the design files folder:
   <download_folder>\sf2_spi_flash_keil_liberov11p4_tu_df\SPI_Flash_Drivers folder.
5. Copy the **winbond flash** folder to the drivers folder of SPI_Flash_MSS_CM3_hw_platform project in the uVision workspace.

6. Right-click **drivers** and add the driver file (winbondflash.c) to SPI_Flash_MSS_CM3_hw_platform project in the Keil uVision workspace as shown in Figure 22.

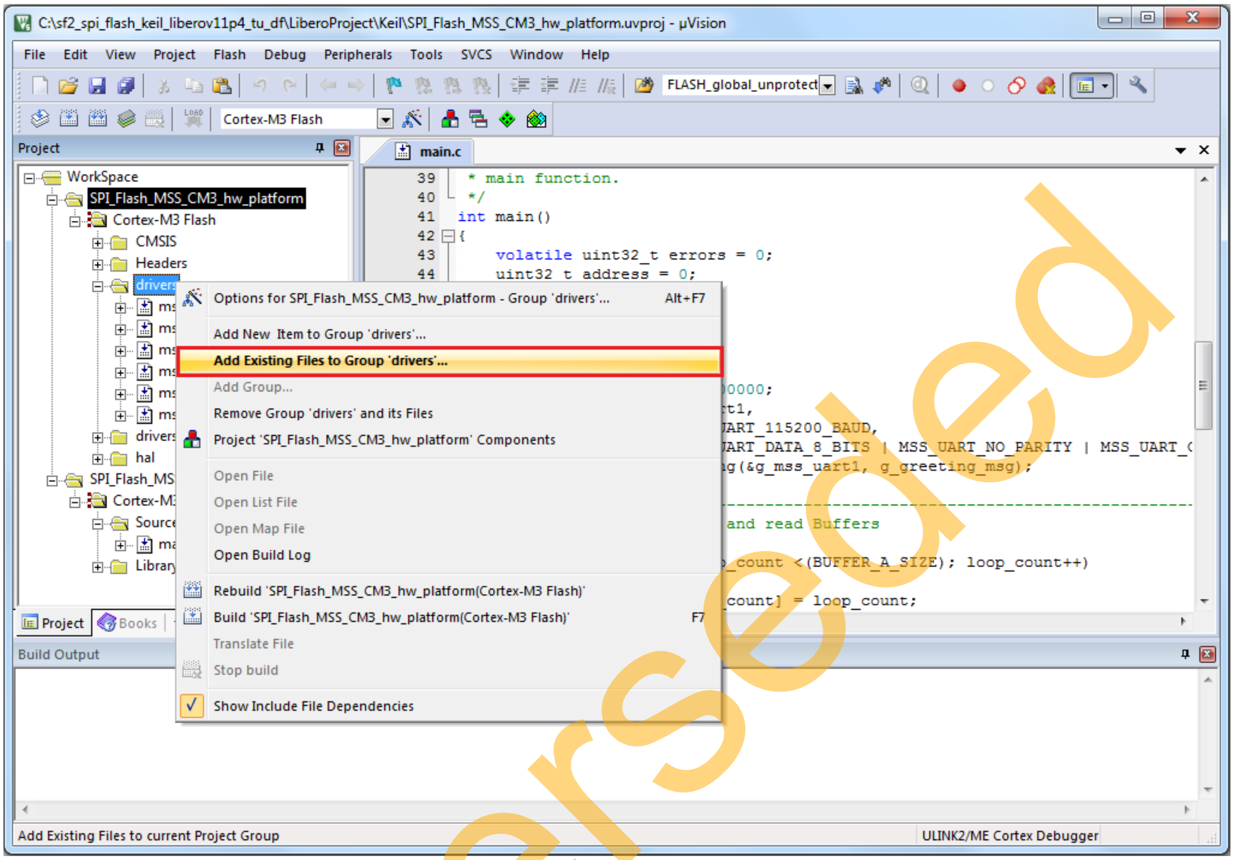

***Figure 22* • Project Explorer Window**

7. Change **SPI_Flash_MSS_Cm3_hw_platform** debug mode to **Cortex-M3_SRAM** by selecting **Cortex-M3_SRAM** from the drop-down list, as shown in Figure 23.

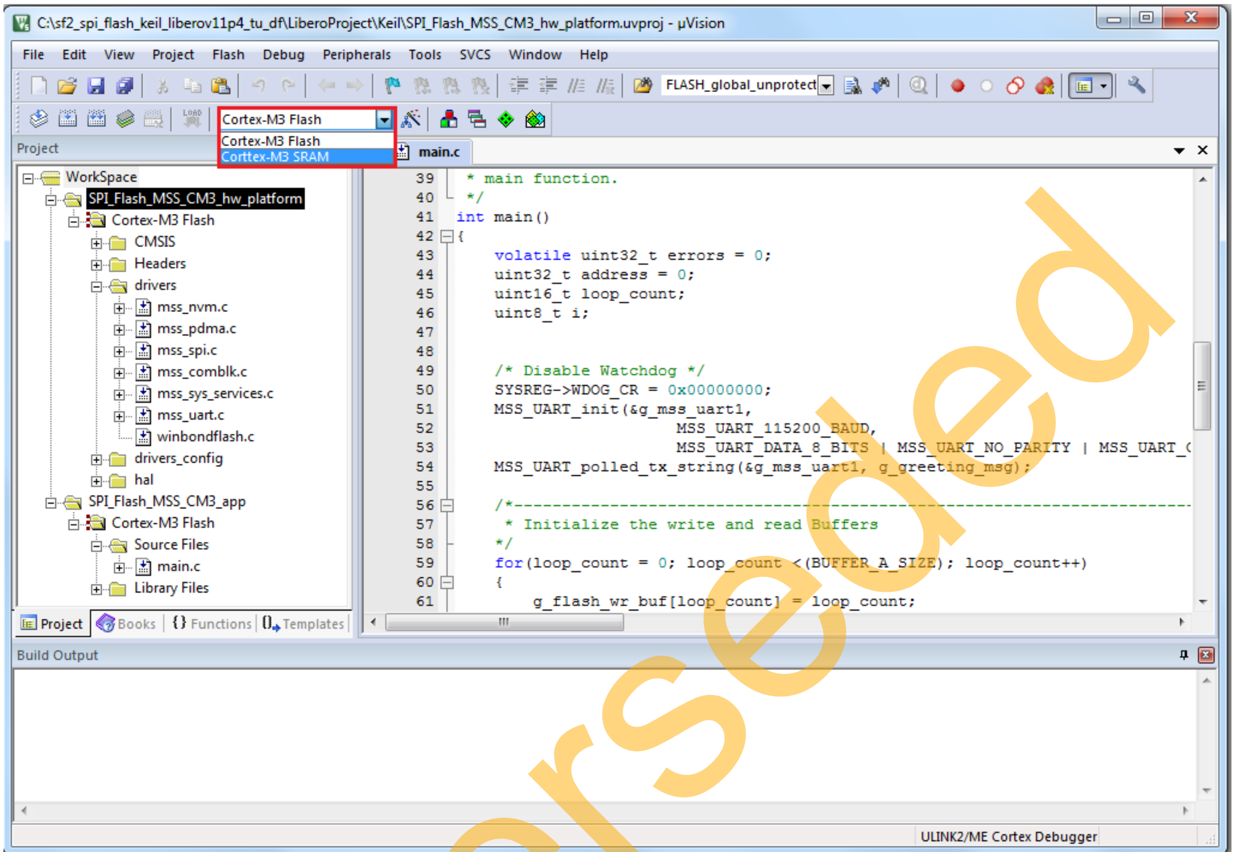

***Figure 23* • Cortex-M3_SRAM Settings**

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.

Follow the steps given below to add MICROSEMI_STDIO_THRU_UART symbol:

a. Right-click **Cortex - M3 SRAM** under **SPI_Flash_MSS_CM3_hw_platform** and click **Options for SPI_Flash_MSS_Cm3_hw_platform - Target Cortex - M3 SRAM**.

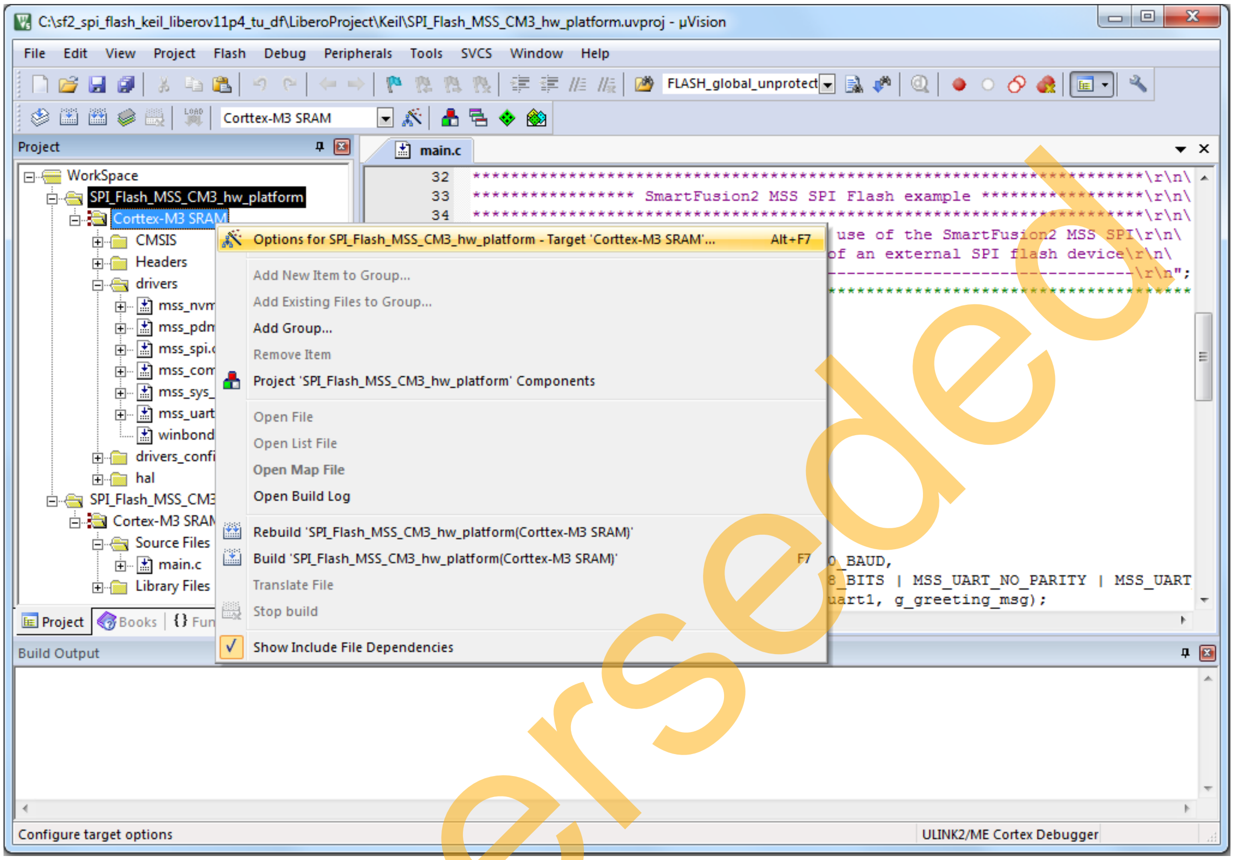

Figure 24 **• Target Options**

b. Go to **C/C++** tab and enter **MICROSEMI_STDIO_THRU_UART** at **Define** under Preprocessor Symbols as shown in Figure 25 on page 22.

c. Click **OK**.

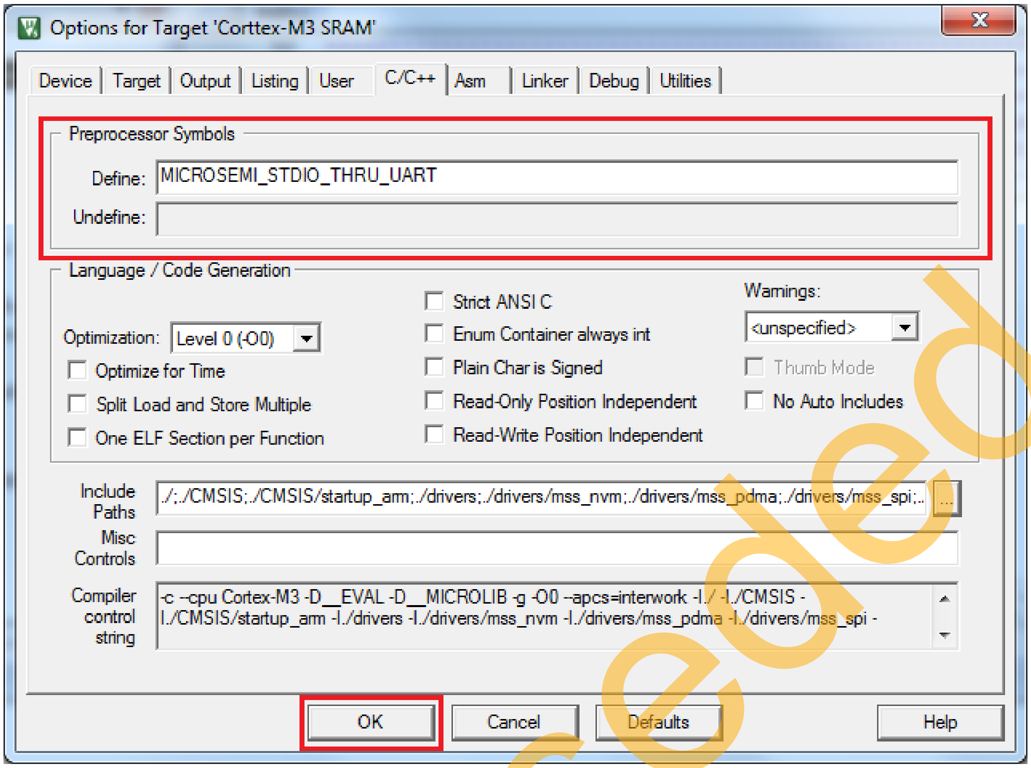

***Figure 25* • Target Options-Add Symbols**

8. Right-click **Cortex-CM3_SRAM** under SPI_Flash_MSS_Cm3_hw_platform and select **Build SPI_Flash_MSS_Cm3_hw_platform (Cortex-CM3 SRAM)** as shown in Figure 26.

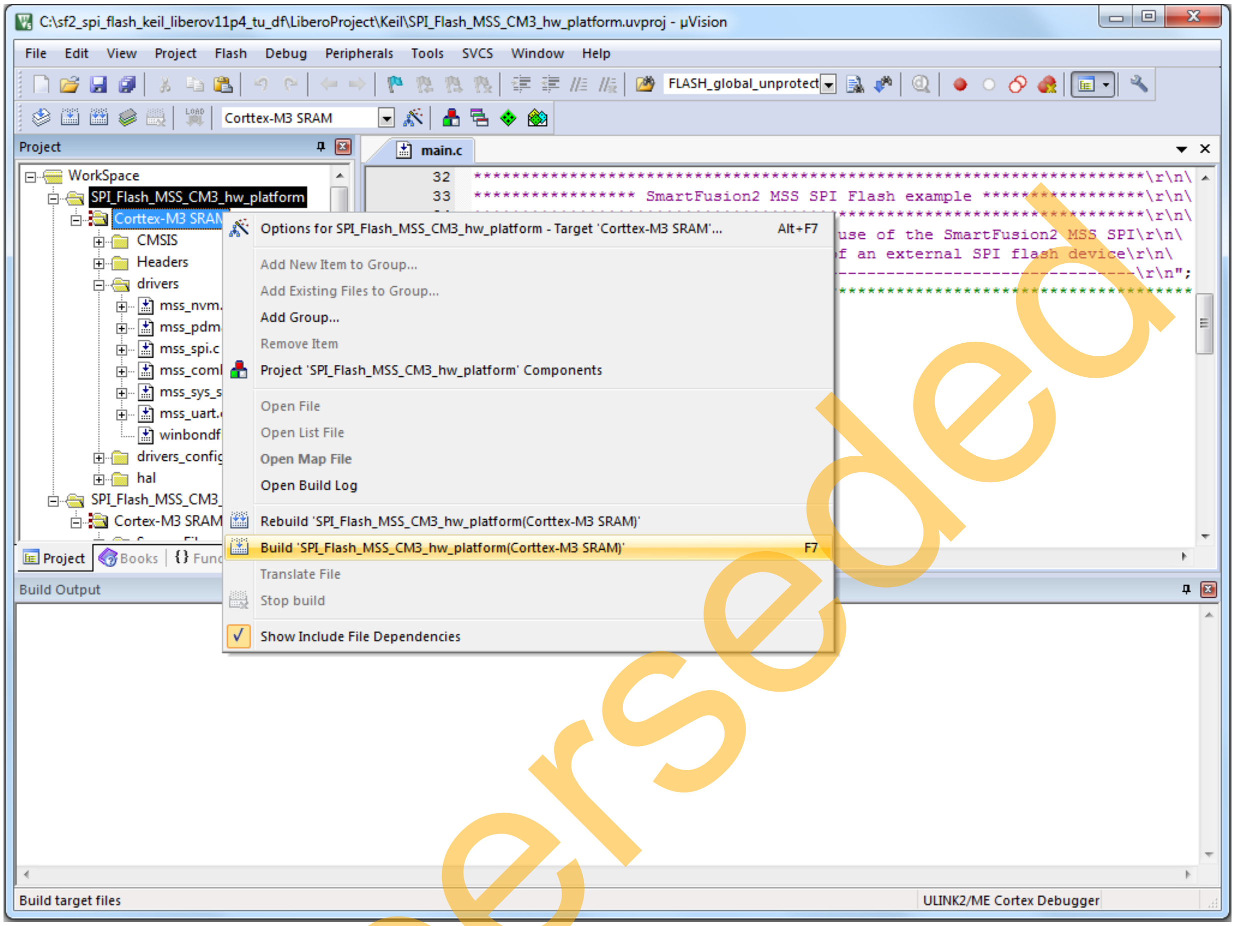

Figure 26 • **Build HW Platform Window**

9. Right-click **SPI_Flash_MSS_CM3_app** and select **Set as Active Project**.

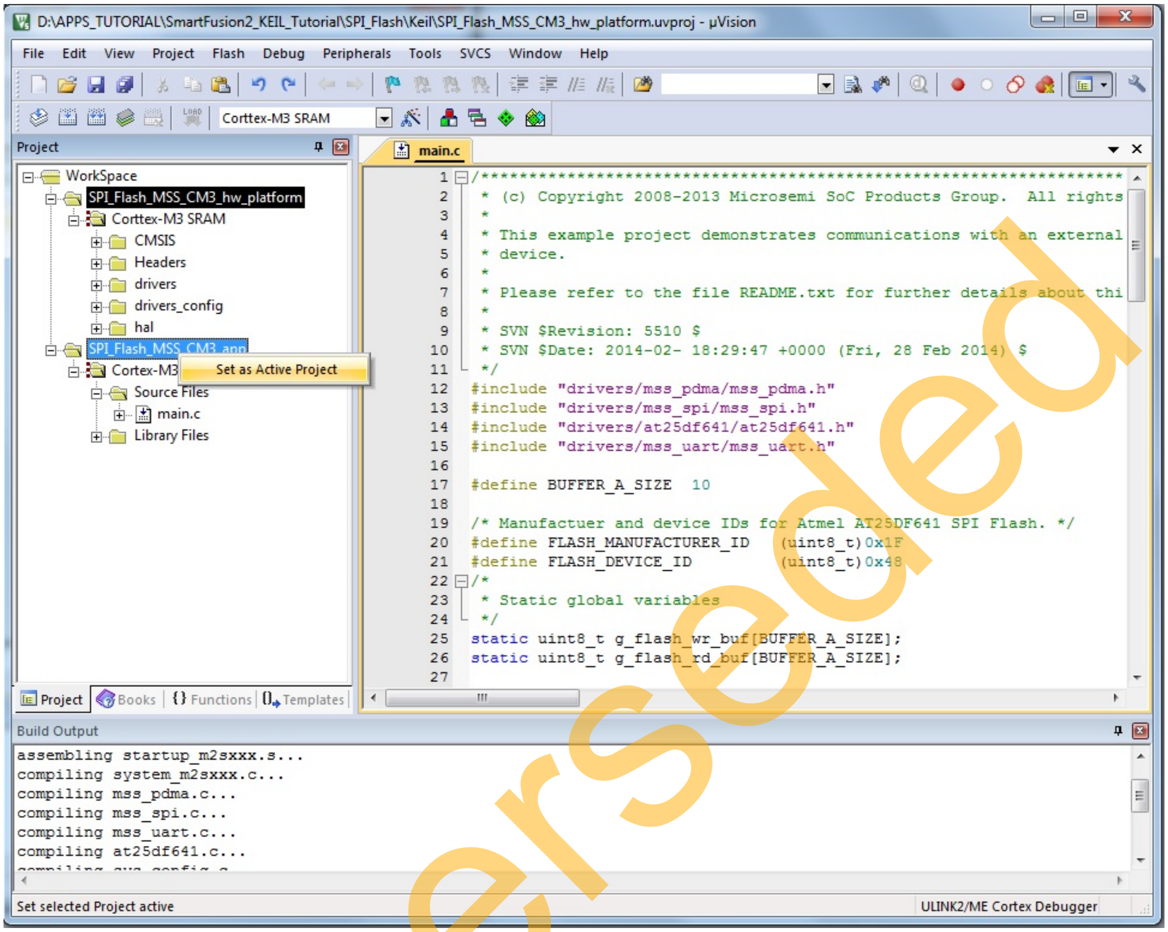

***Figure 27 •* Set as Active Project**

10. Change **SPI_Flash_MSS_CM3_app** debug mode to **Cortex-M3_SRAM** by selecting **Cortex-M3_SRAM** from the drop-down menu as shown in Figure 28.

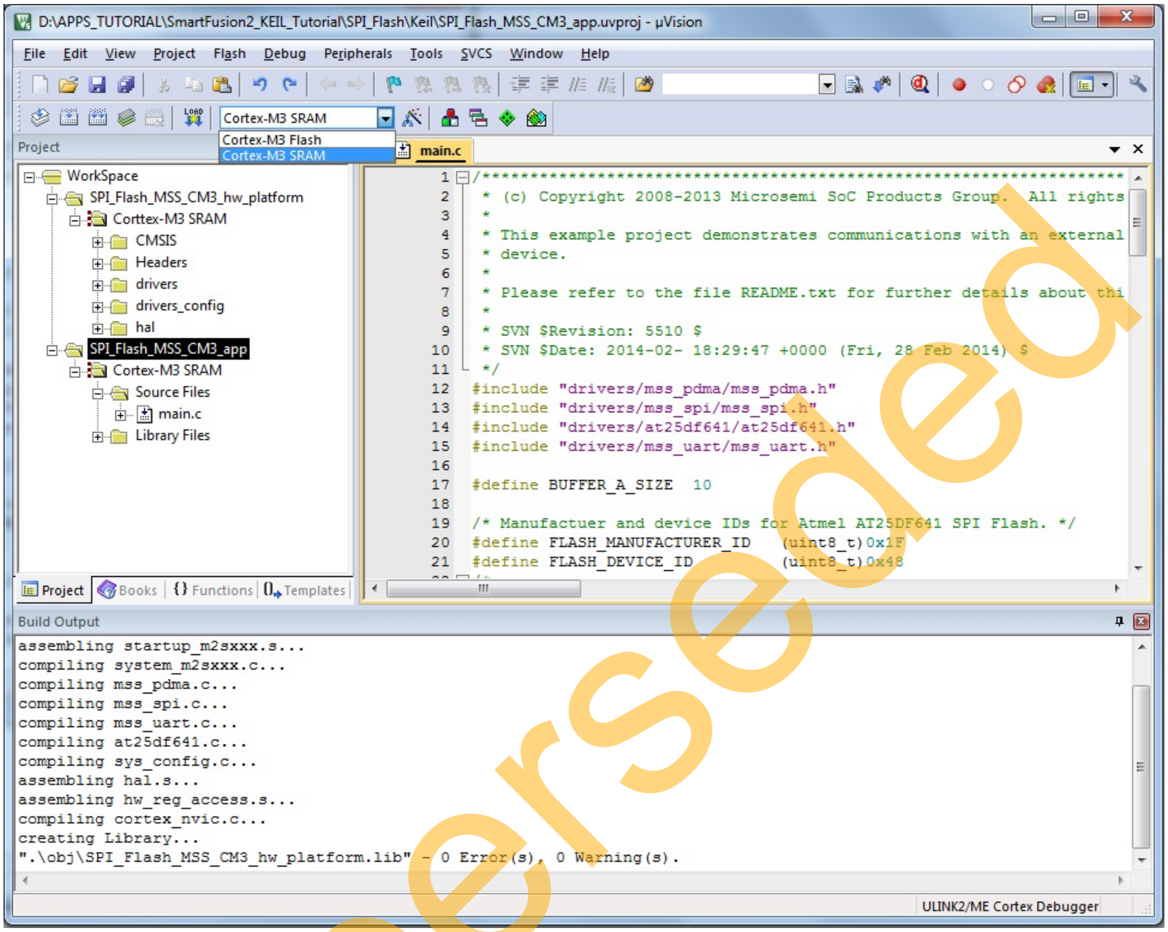

***Figure 28* • Cortex-M3_SRAM Settings**

11. Right-click **Cortex-M3 SRAM** under SPI_Flash_MSS_CM3_app and click **Options for project**.

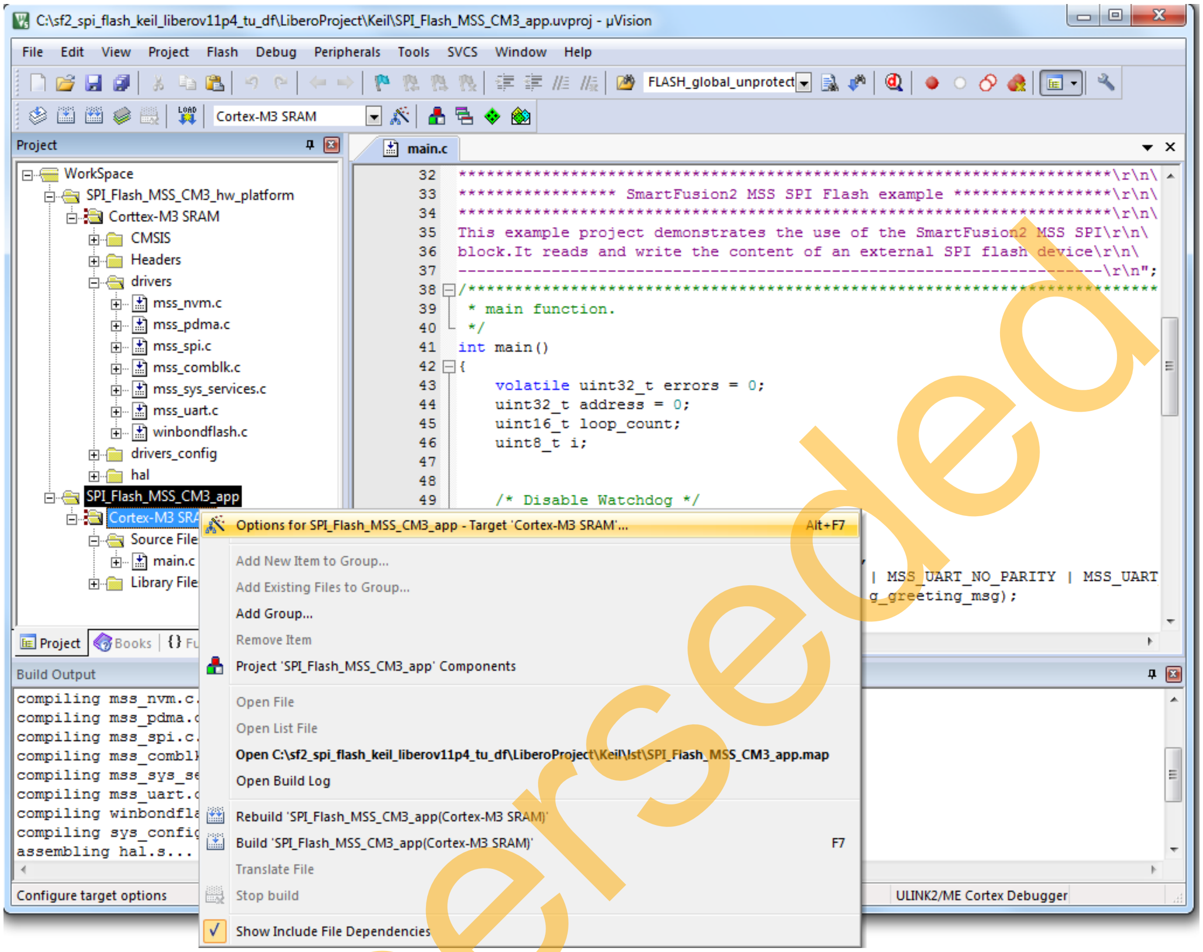

***Figure 29* • Target Options**

12. Click the **Linker** tab and navigate to the sf2_spi_flash_keil_liberov11p4_tu_df\LiberoProject\Keil\CMSIS\startup_arm folder to select the **Scatter File** as `smartfusion2_esram_debug.sct`, as shown in Figure 30.

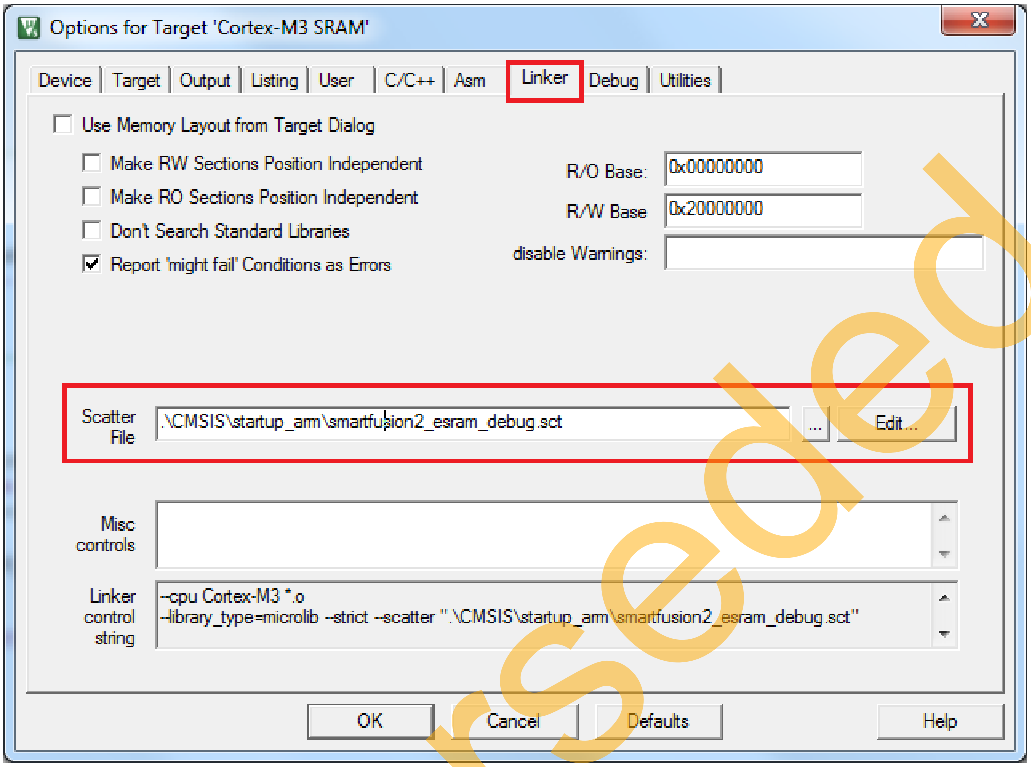

***Figure 30* • Target Options - Scatter File**

13. Click the **Debug** tab and browse the //sf2_spi_flash_keil_liberov11p4_tu_df/SourceFiles/ folder for initialization file provided as shown in Figure 31.

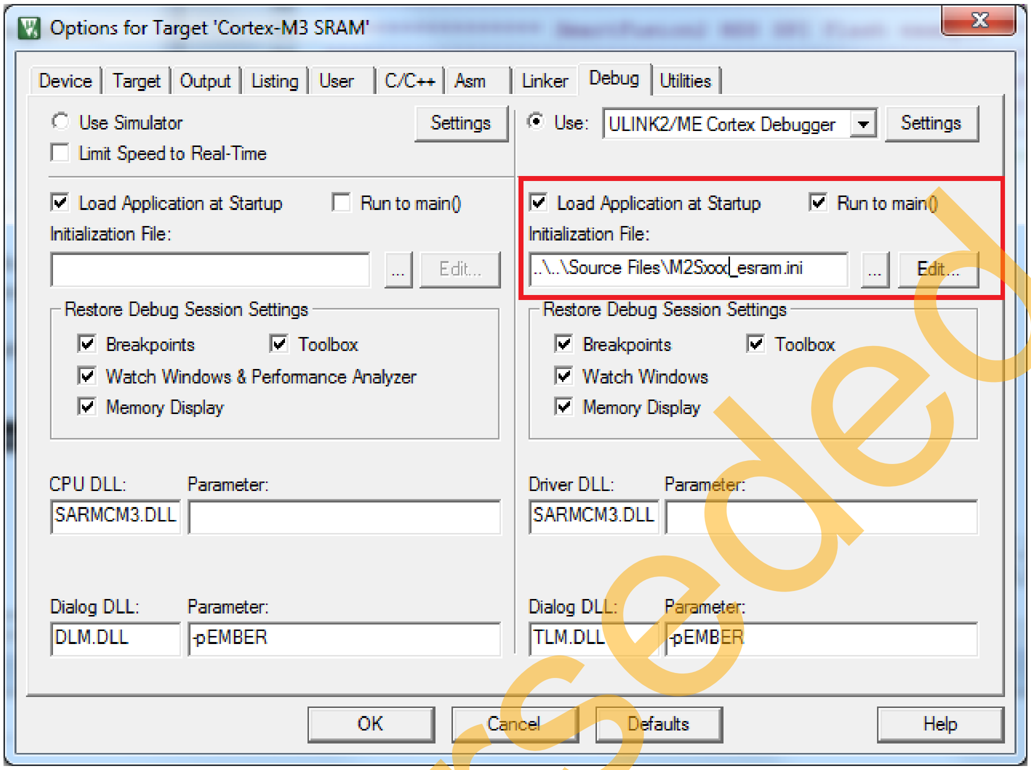

***Figure 31* • Target Options - Initialization File**

14. Click the **Utilities** tab and uncheck **Use Debug Driver** and **Update Target before Debugging** check boxes.

15. Select **ULINK2/ME Cortex Debugger** from the drop-down list and click **OK** as shown in Figure 32.

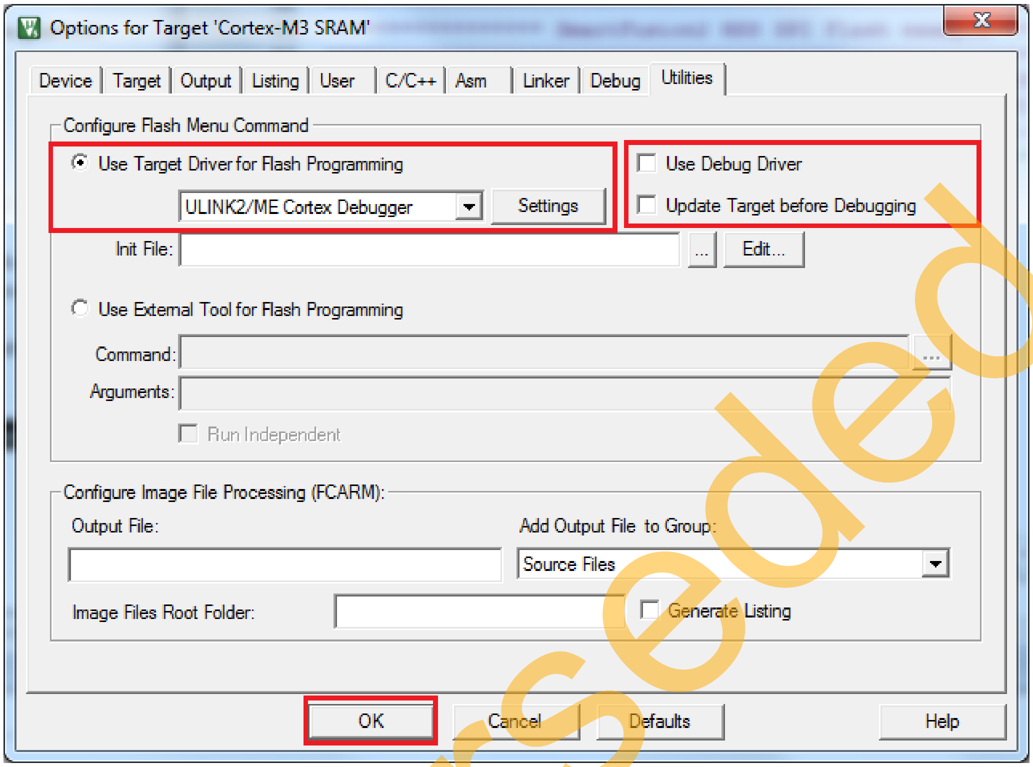

***Figure 32* • Target Options - Utilities Settings**

16. Right-click **Cortex-M3 SRAM** under **SPI_Flash_MSS_CM3_app** and select **Build SPI_Flash_MSS_CM3_app (Cortex-M3 SRAM)** as shown in Figure 33. It compiles all of the source files and links the object files into an AXF file to debug. Make sure that there are no errors. Correct any syntax errors and rebuild if necessary.

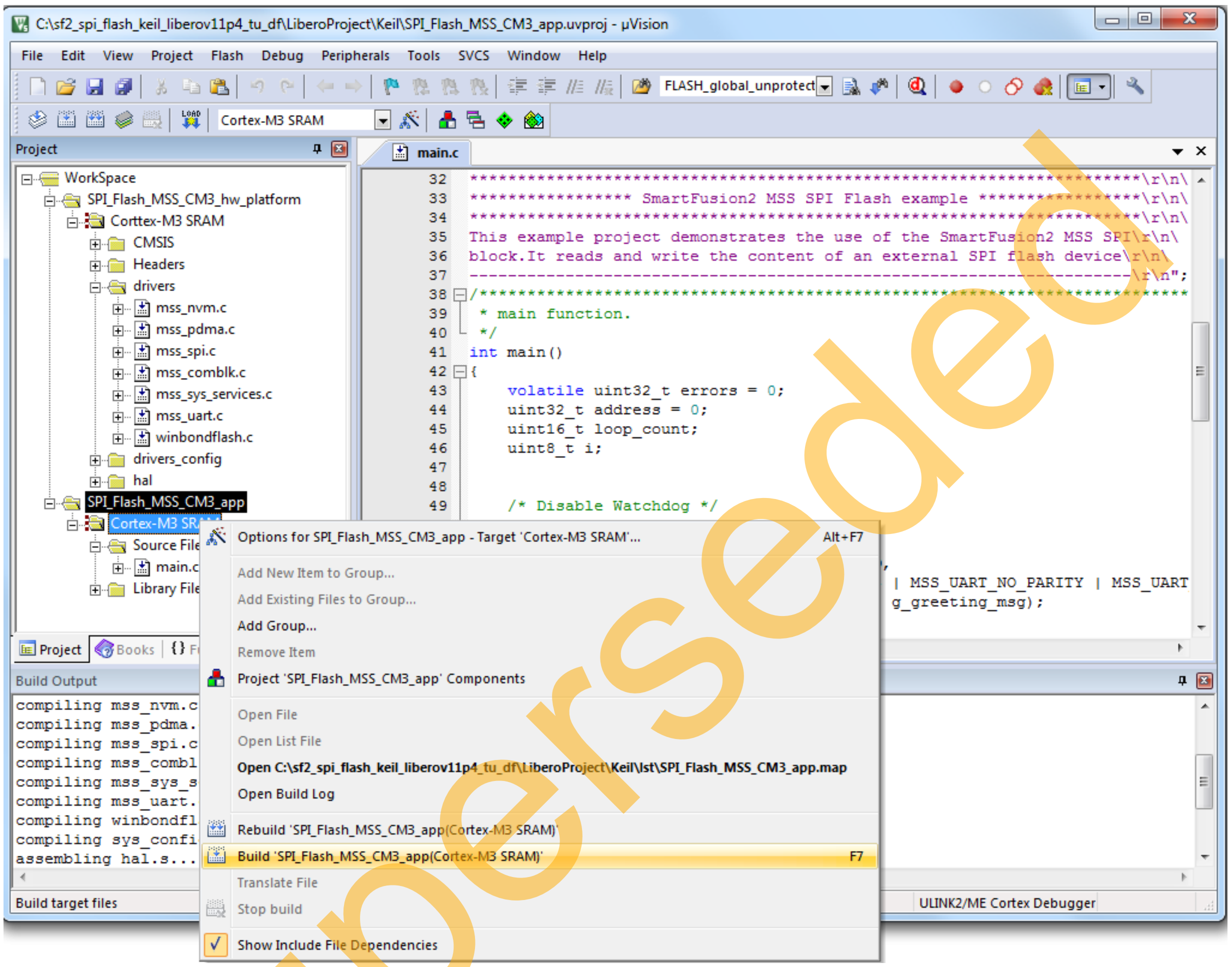

***Figure 33* • Build Application Window**

Figure 34 shows the messages that are displayed in the console after the build.

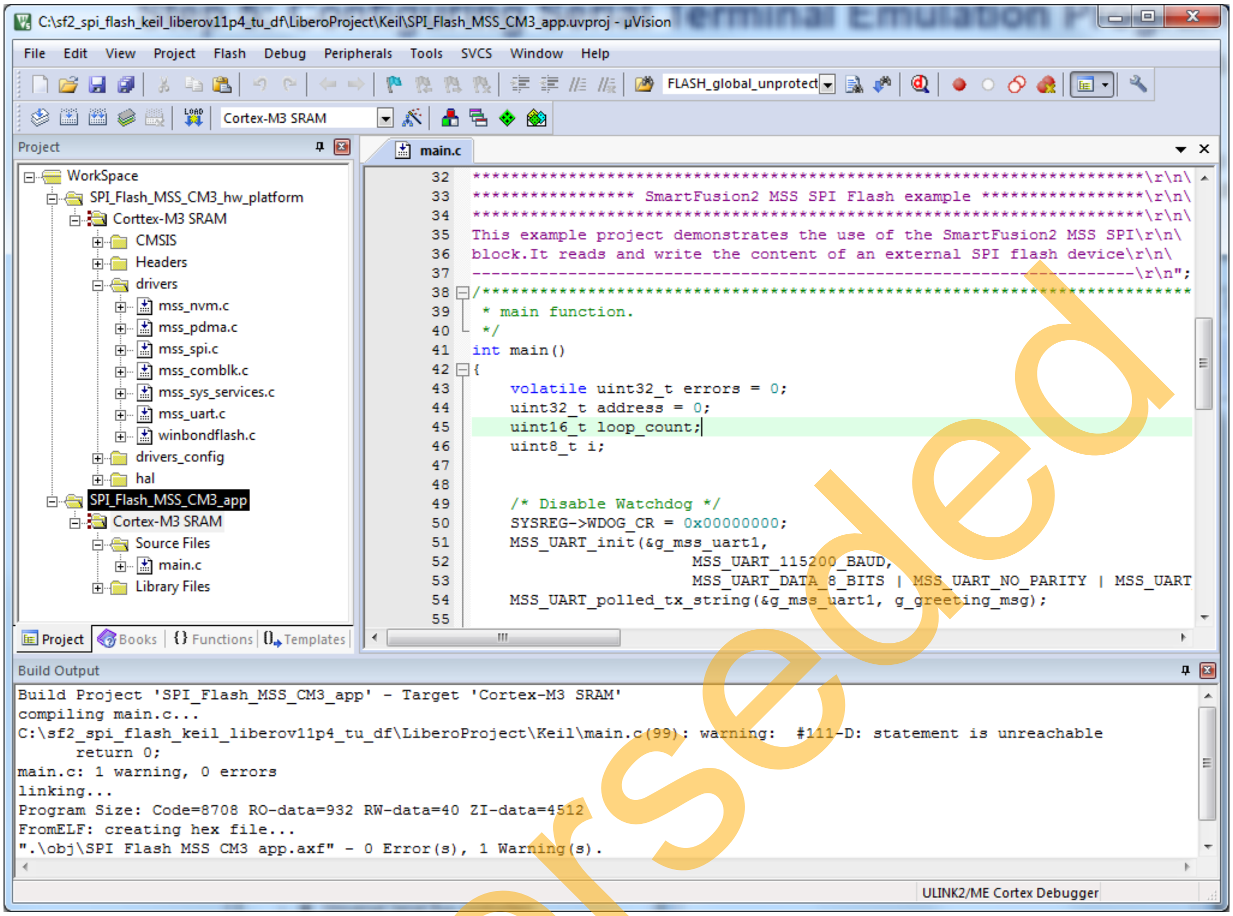

***Figure 34* • Build Output**

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 35 shows an example Device Manager window.

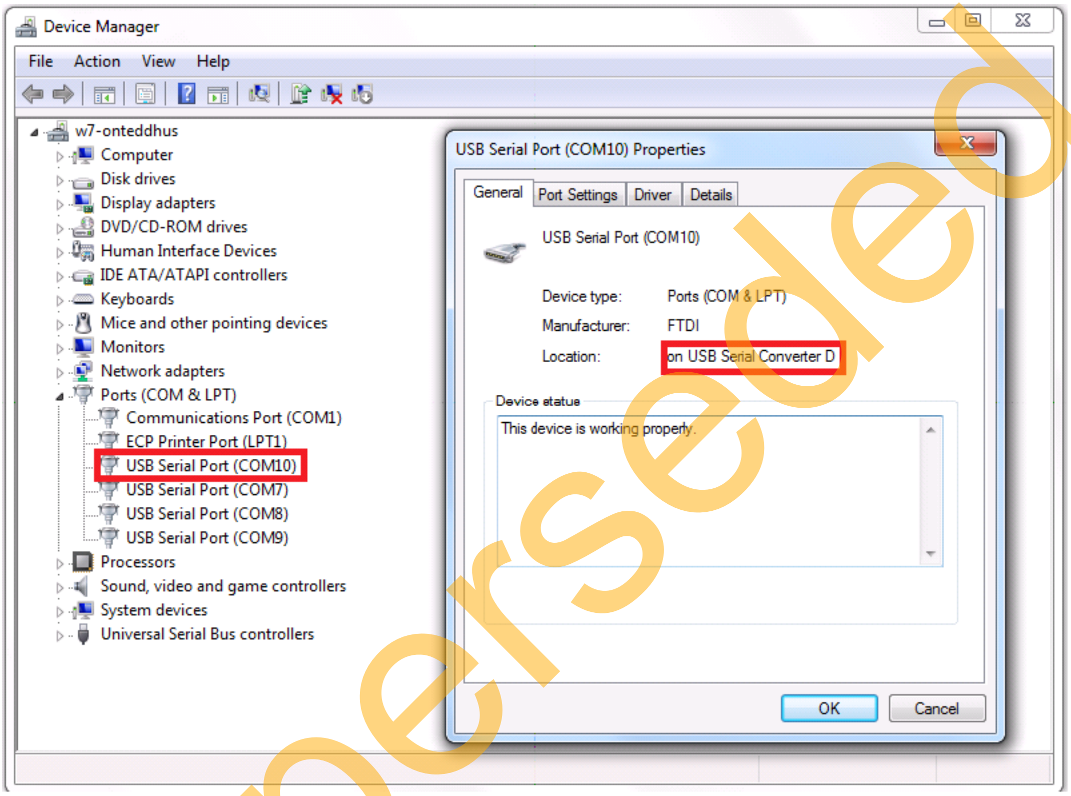

***Figure 35* • Device Manager Window**

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: Connecting the ULINK-ME to the Board and PC

This section describes the connection between the SmartFusion2 Evaluation Kit board, ULINK-ME, and host PC. Use the appropriate settings for the board that is in use.

1. Connect Pin 2 and Pin 3 on the jumper J8 on the SmartFusion2 Evaluation Kit board.
2. Connect the USB A-Mini B cable between the host PC and the SmartFusion2 Evaluation Kit board. This is used to display the HyperTerminal communications.
3. Verify that the ULINK-ME debugger is connected to the SmartFusion2 Evaluation Kit board RVI Header as shown in Figure 36 and also to the host PC through a USB A-Mini B cable. The ULINK-ME adapter has one LED that indicates connection status in the following ways:
   – Blinking slowly indicates that ULINK-ME is ready to communicate with the debugger.
   – Blinking speedily indicates that the target board is executing the program under debugger control.
   – Remaining **ON** during debugging indicates that the debugger has halted the target board.
   – Remaining **ON** during download indicates that target download and verification is in progress.
4. Switch **ON** the SW7 power supply switch.

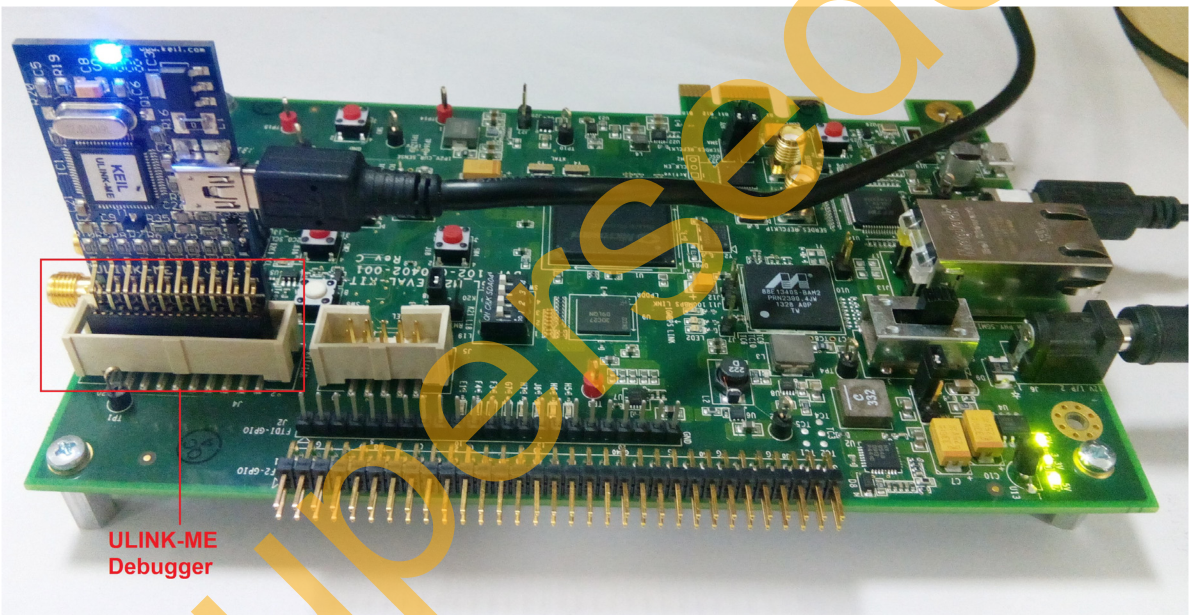

***Figure 36* • ULINK-ME Connections**

Refer to "Appendix A - Board Setup for Debugging from Keil uVision" on page 43 for information on the board setup for running the tutorial.

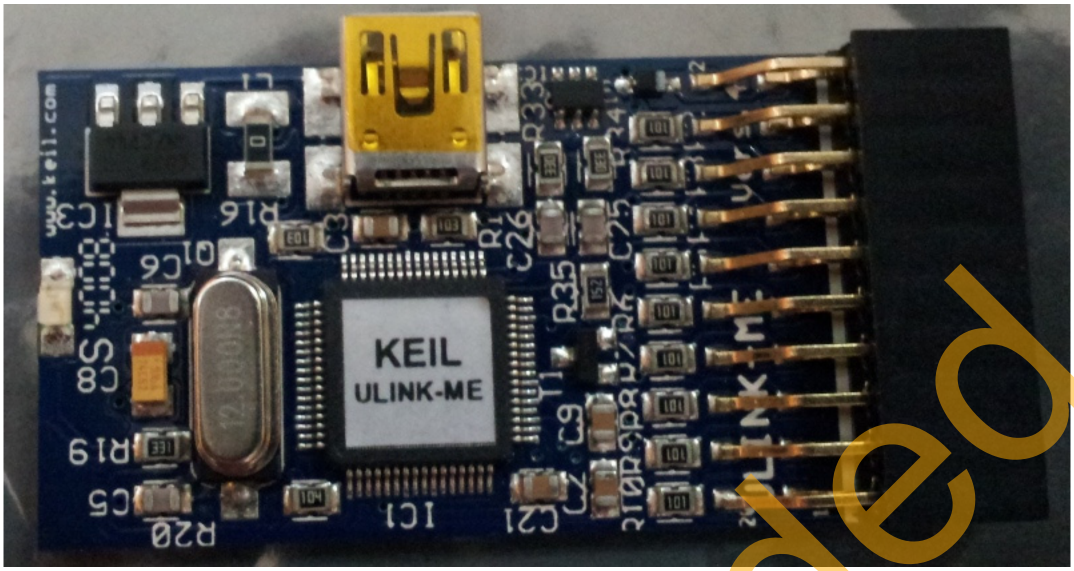

Figure 37 • **ULINK-ME Debugger**

Step 7: Debugging the Application Project using Keil uVision 5

1. Select **Start/Stop Debug Session** from the **Debug** menu in the uVision window to run it through the debug hardware as shown in Figure 38. The processor code will be downloaded to the SmartFusion2 eSRAM.

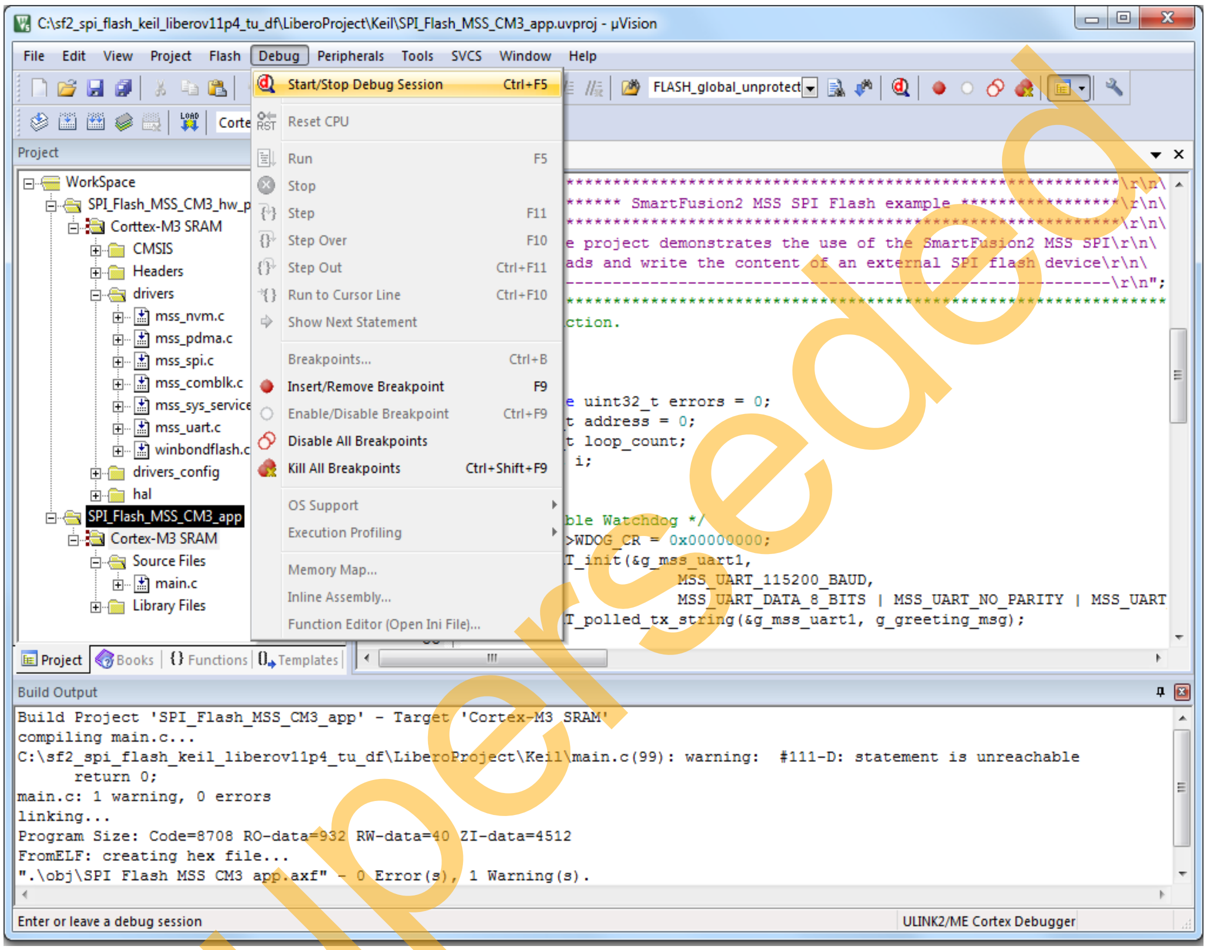

***Figure 38* • Selecting Start/Stop Debug Session**

The code will automatically 'run to main' and then stop as shown in Figure 39.

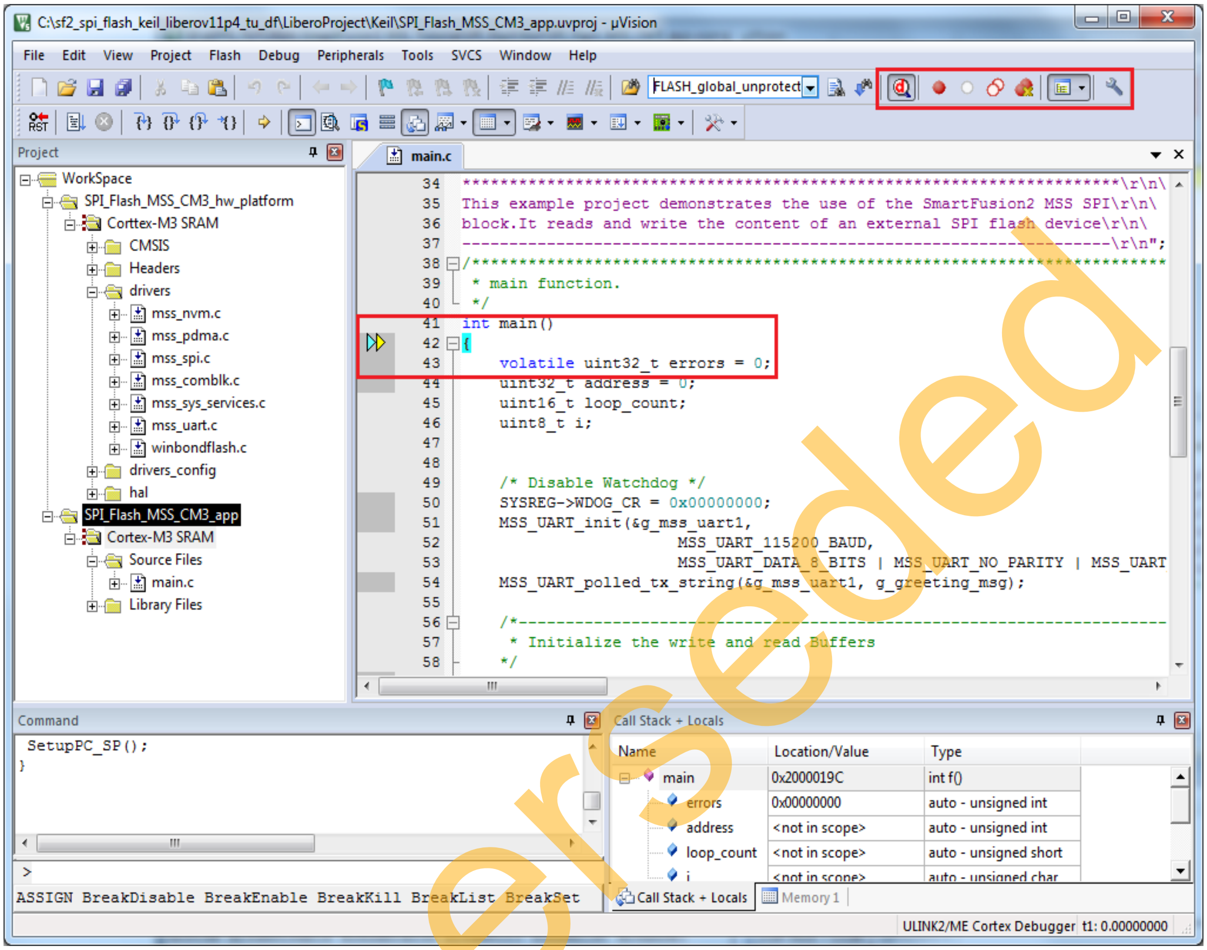

***Figure 39* • Debug Menu**

2. Click **Run** from the **Debug** menu as shown in Figure 40.

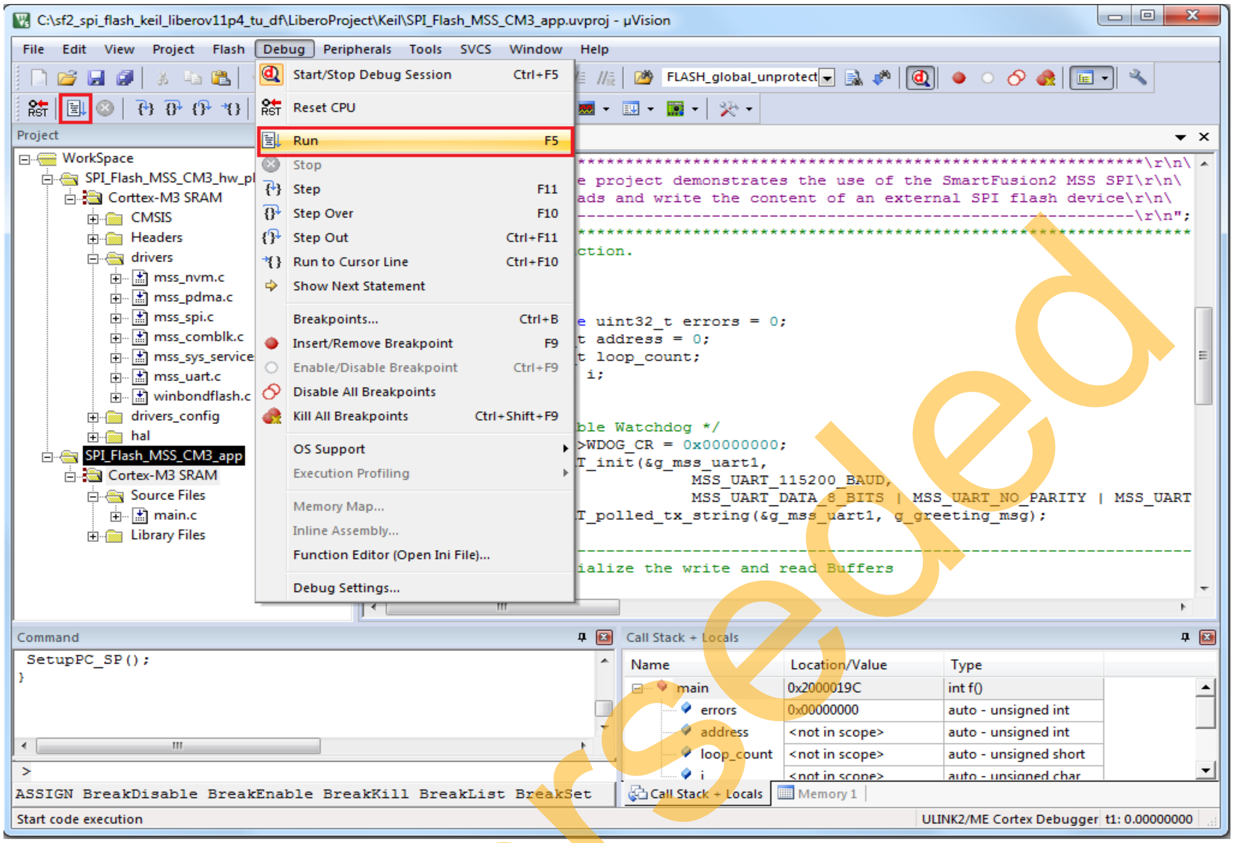

***Figure 40* • Selecting Run from the Debug Menu**

On successful operation, the HyperTerminal window displays a message as **Read Data From Flash** as shown in Figure 41.

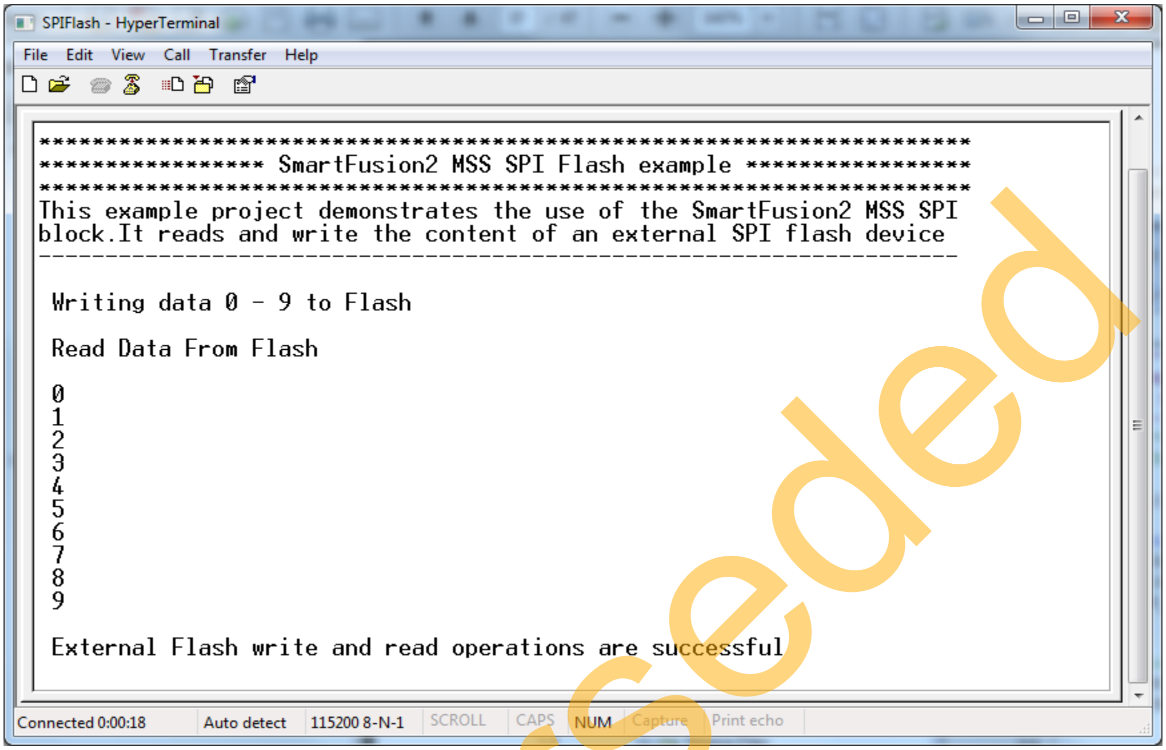

***Figure 41* • HyperTerminal Window**

The **Disassembly** window is displayed in the middle of the **Debug** section as shown in Figure 42. If not, click the **Disassembly** icon to display the **Disassembly** section.

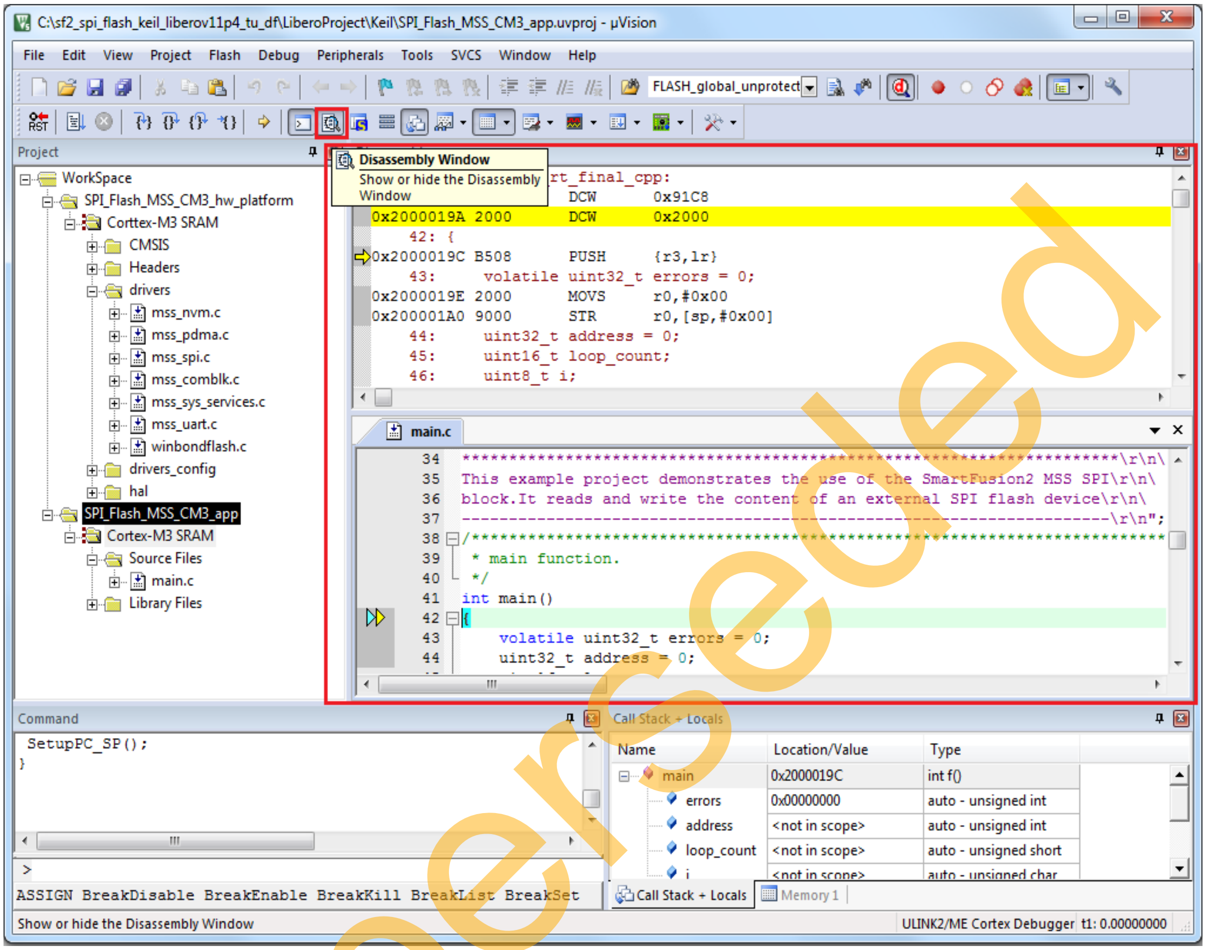

Figure 42 **• Disassembly Window**

3. Check the **Registers** section to view the values of the ARM® Cortex™-M3 processor internal registers.

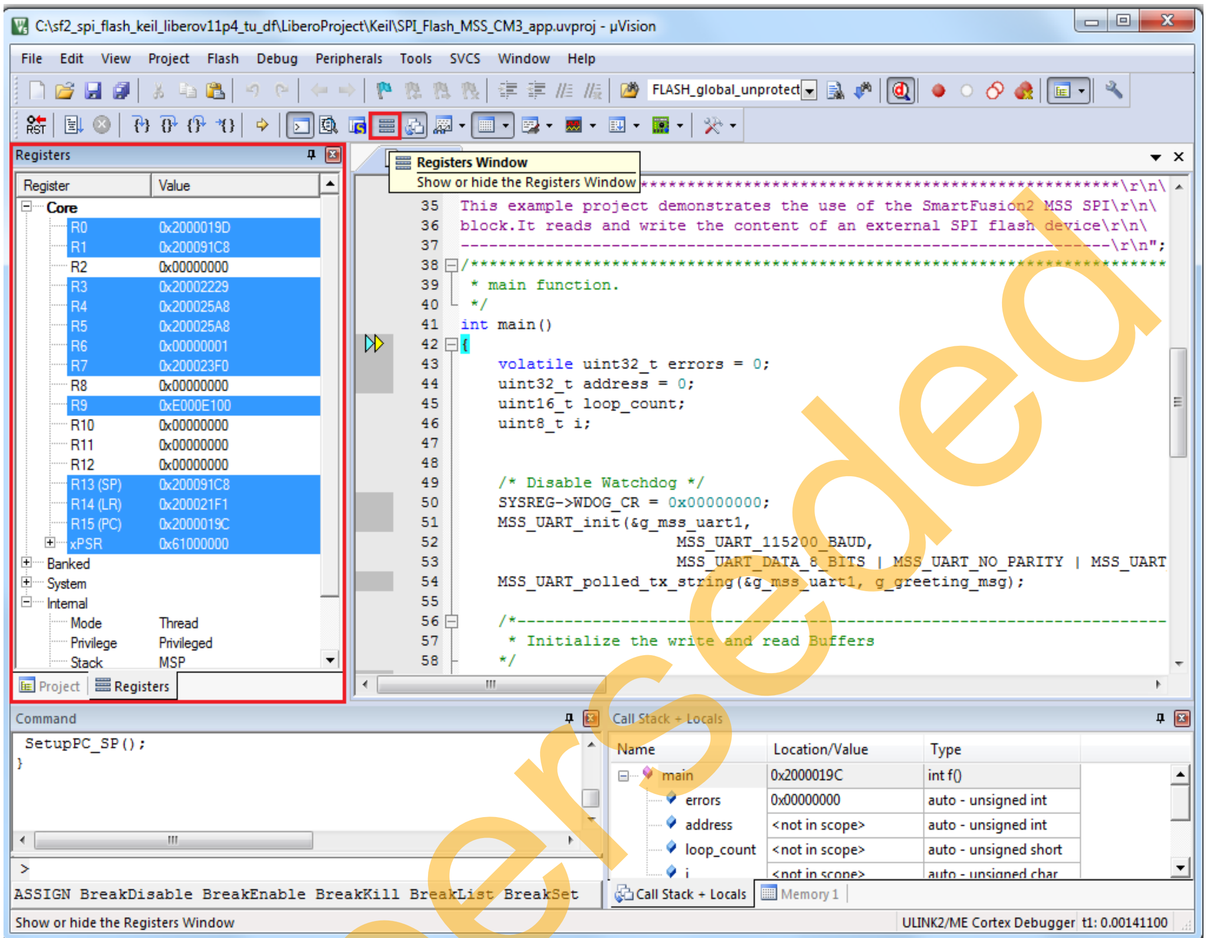

Figure 43 **• Values of the Cortex-M3 Internal Registers**

4. When debug process is finished, terminate execution of the code by choosing **Debug > Start/Stop Debug Session** as shown in Figure 44.

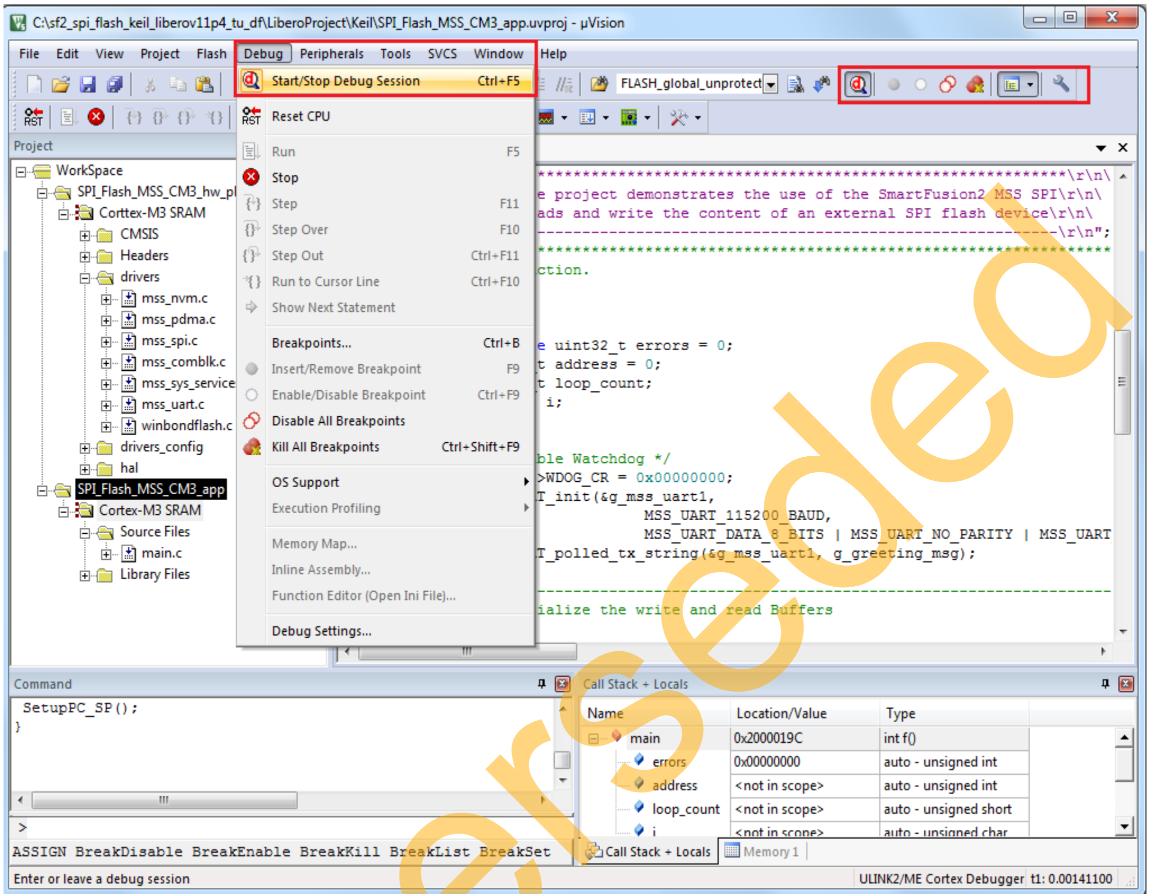

***Figure 44* • Keil uVision Workbench - Stop Debug Option**

5. The Step Level Debugging can be performed before running the application using **Run**. These can be accessed from the Debug menu or on the Keil uVision workbench as shown in Figure 45:

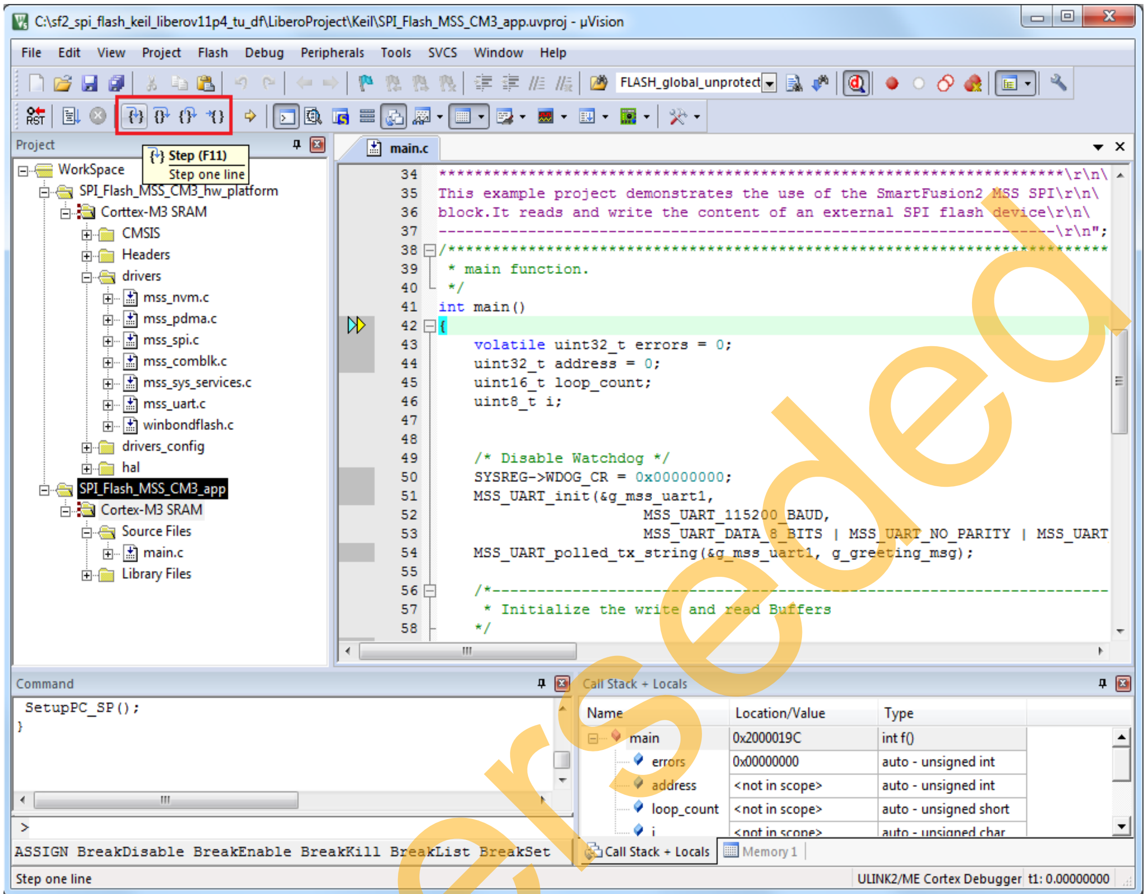

***Figure 45* • Keil uVision Workbench - Step Level Debugging**

– Source code can be single-stepped by selecting from the Debug menu **Debug > Step**, **Debug > Step Over**, **Debug > Step Out** or by selecting the respective options from the Keil uVision workbench as shown in Figure 45. Observe the changes in the source code window and Disassembly section. 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.

– Select **Debug > Step Out** to exit the instruction in stepping mode.

6. Add breakpoints from the **Debug** menu in workbench to force the code to halt, start Debug session, and then single-step and observe the instruction sequence.
7. Close uVision using **File > Exit**.
8. Close the HyperTerminal using **File > Exit**.

Conclusion

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

Appendix A - Board Setup for Debugging from Keil uVision

Figure 1 shows the board setup for debugging the Keil uVision on the SmartFusion2 Evaluation Kit board.

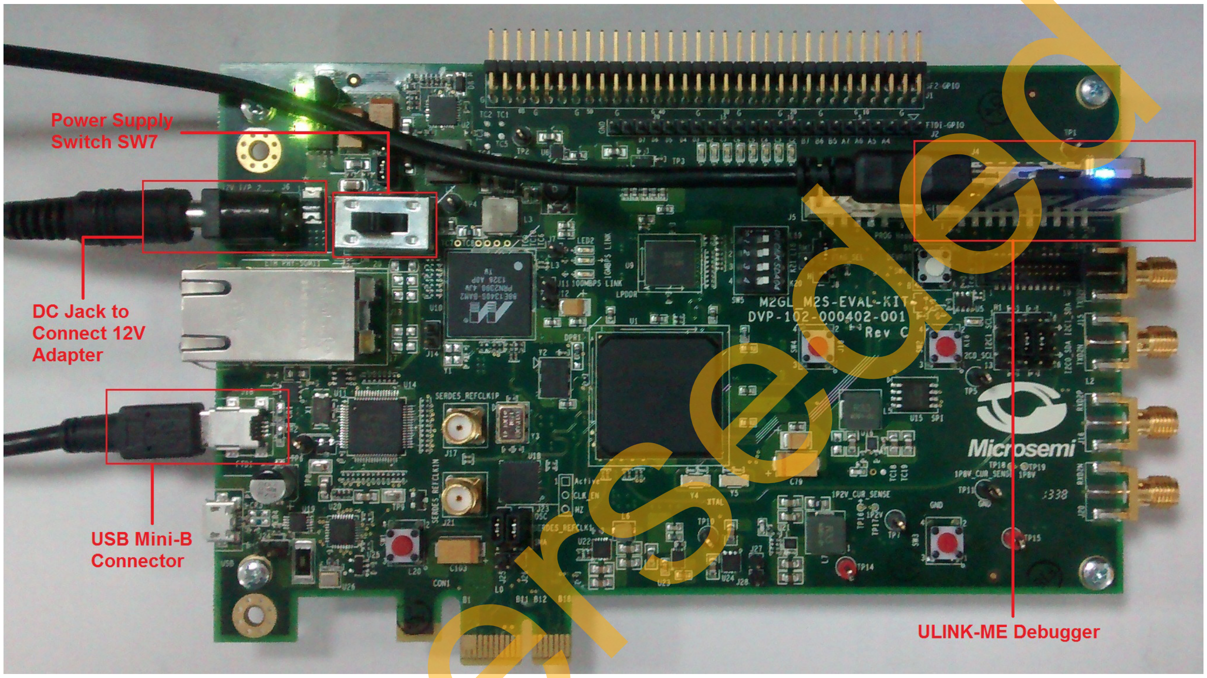

Figure 1 • **SmartFusion2 Evaluation Kit in Debug Mode using Keil uVision**

Appendix B - Board Setup for Programming the Tutorial

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

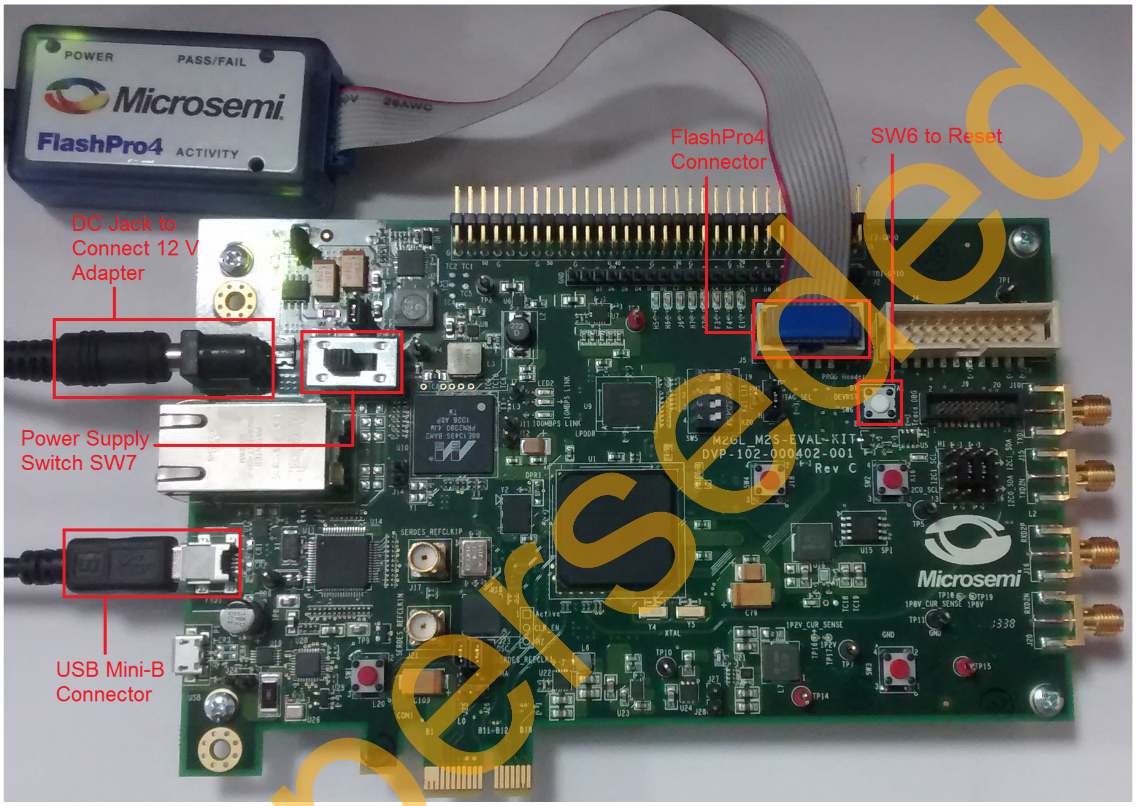

Figure 1 • **SmartFusion2 Evaluation Kit in Programming Mode**

Appendix C- SmartFusion2 Evaluation Kit Board Jumper Locations

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

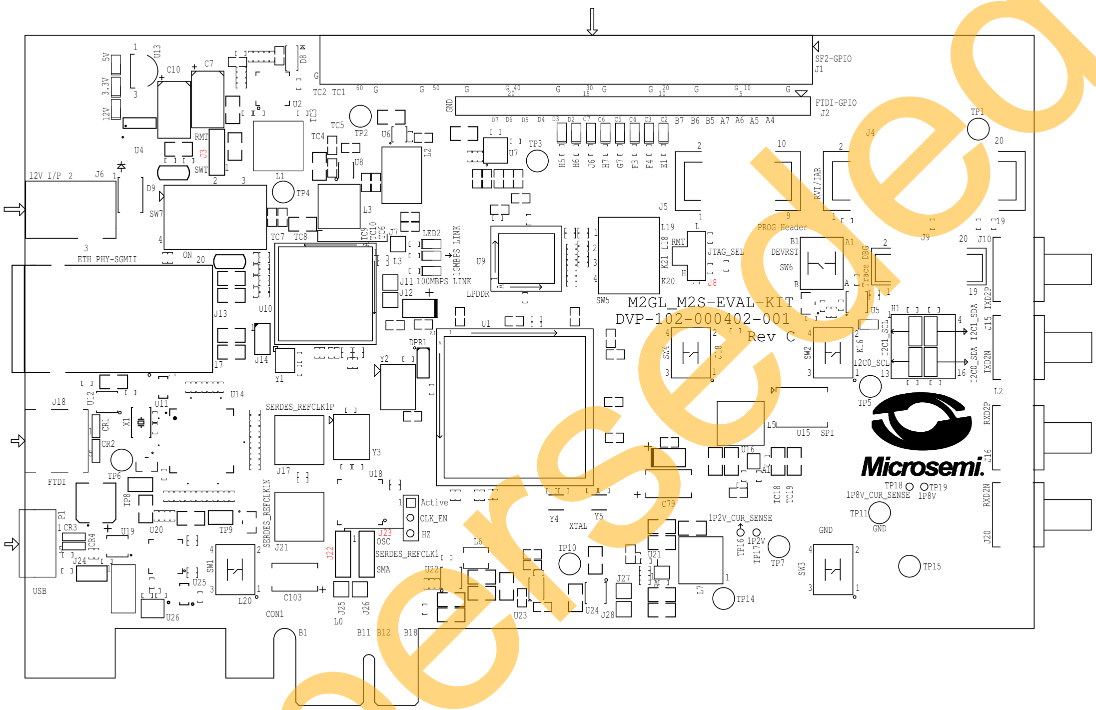

Figure 1 **• SmartFusion2 Evaluation Kit Board Jumper Locations**

Note:

- Jumpers highlighted in red (J22, J23, J3, J8) are set by default.
- The location of the jumpers in Figure 1 are searchable.

List of Changes

The following table lists the critical changes that were made in each revision of the chapter in the tutorial.

Date	Version	Changes
November 2014	2	Updated the document for Libero v11.4 software release changes (SAR 61938).
April 2014	1	Initial release.

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

50200548-2/11.14