Accessing Serial Flash Memory Using SPI Interface - Libero SoC v11.6 and Keil uVision Flow for SmartFusion2

TU0548 Tutorial

Microsemi

Table of Contents

Accessing Serial Flash Memory using SPI Interface - Libero SoC v11.6 and Keil uVision Flow for SmartFusion2

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) Security Evaluation Kit.

The same firmware project can be built using the IAR and Keil tools as well. Following are the respective references:

- *TU0546: Accessing Serial Flash Memory Using SPI Interface - Libero SoC and SoftConsole Flow Tutorial for SmartFusion2 SoC FPGA*
- *TU0547: Accessing Serial Flash Memory Using SPI Interface - Libero SoC and IAR Embedded Workbench Flow Tutorial for SmartFusion2*

This tutorial describes the following:

Step 1: Creating a Libero SoC Project
Step 2: Generating the Program File
Step 3: Programming the SmartFusion2 Security Evaluation Board Using FlashPro
Step 4: Configuring and Generating Firmware
Step 5: Building the Software Application Using Keil uVision 5 IDE
Step 6: Configuring Serial Terminal Emulation Program
Step 7: Connecting the ULINK-ME to the Board and PC
Step 8: Debugging the Application Project using Keil uVision 5

Design Requirements

Table 1 lists the design requirements of Keil uVision flow.

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
Keil debugger	–
Host PC or Laptop	Any 64-bit Windows Operating System
Software Requirements	
Libero SoC	v11.6
Keil uVision	v5
FlashPro programming software	v11.6
Host PC Drivers	USB to UART drivers

Project Files

The design files for this tutorial can be downloaded from the Microsemi® website:
http://soc.microsemi.com/download/rsc/?f=m2s_tu0548_liberov11p6_df

The 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.

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 the 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.

Refer to the *UG0331: SmartFusion2 Microcontroller Subsystem User Guide* for more information on SPI.

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

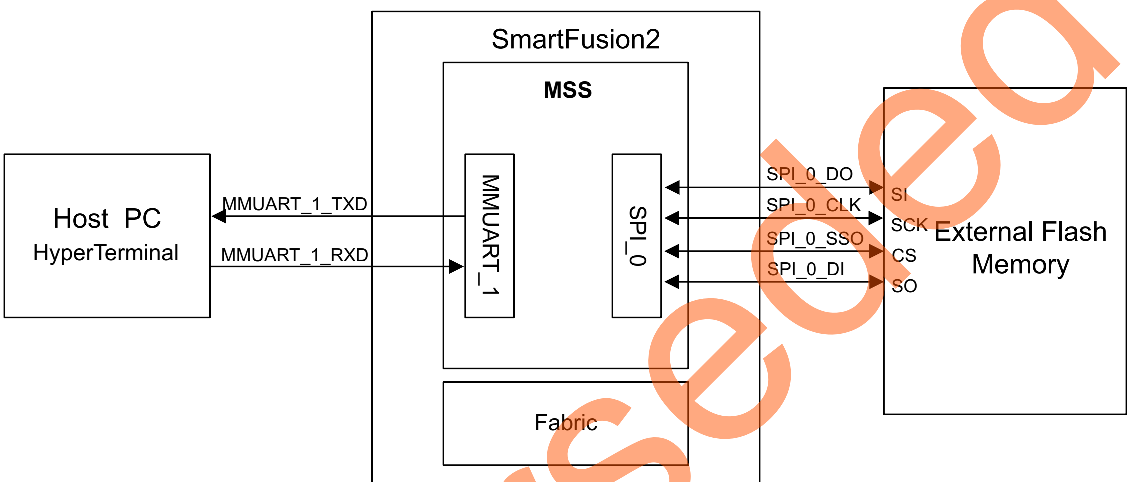

***Figure 1* • SPI Flash Interfacing Block Diagram**

Step 1: Creating a Libero SoC Project

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

Launching Libero SoC

The following steps describe how to launch Libero SoC:

1. Choose **Start > Programs > Microsemi Libero SoC v11.6 > Libero SoC** v11.6, or double-click the shortcut on desktop to open the Libero SoC v11.6 Project Manager.
2. Create a new project using one of the following options:
   – Select **New** on the **Start Page** tab, as shown in Figure 2.
   – Click **Project > New Project** from the Libero SoC menu.

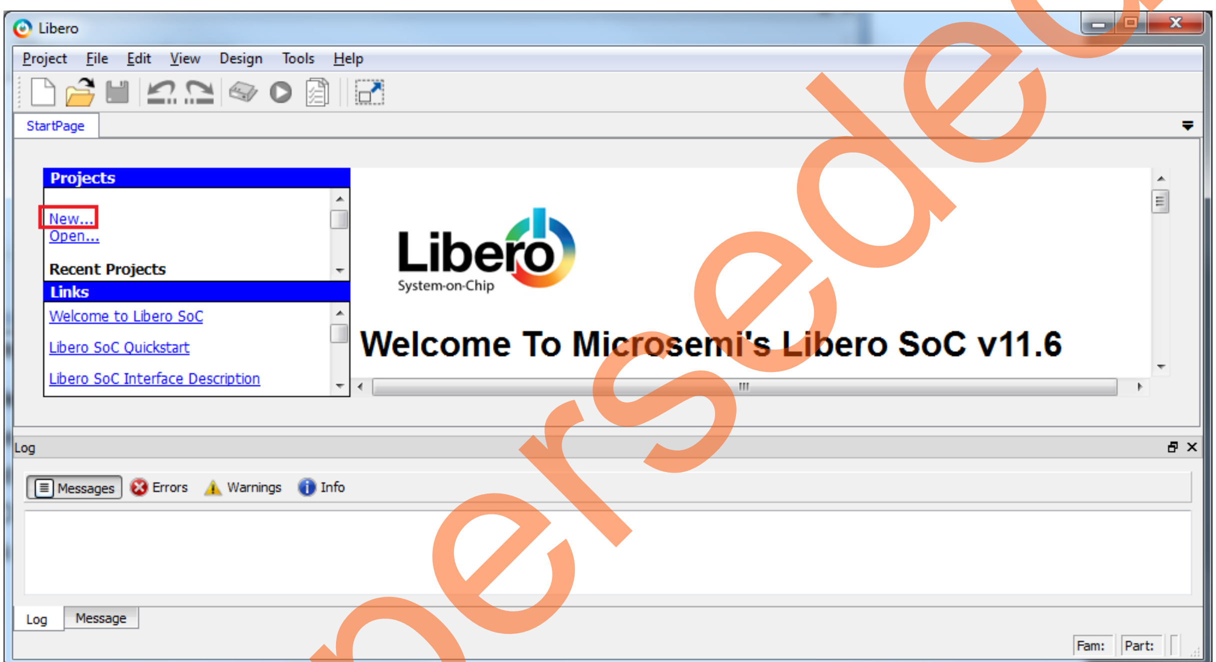

Figure 2 **• Libero SoC Project Manager**

3. Enter the following information in the **Project Details** page, as shown 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

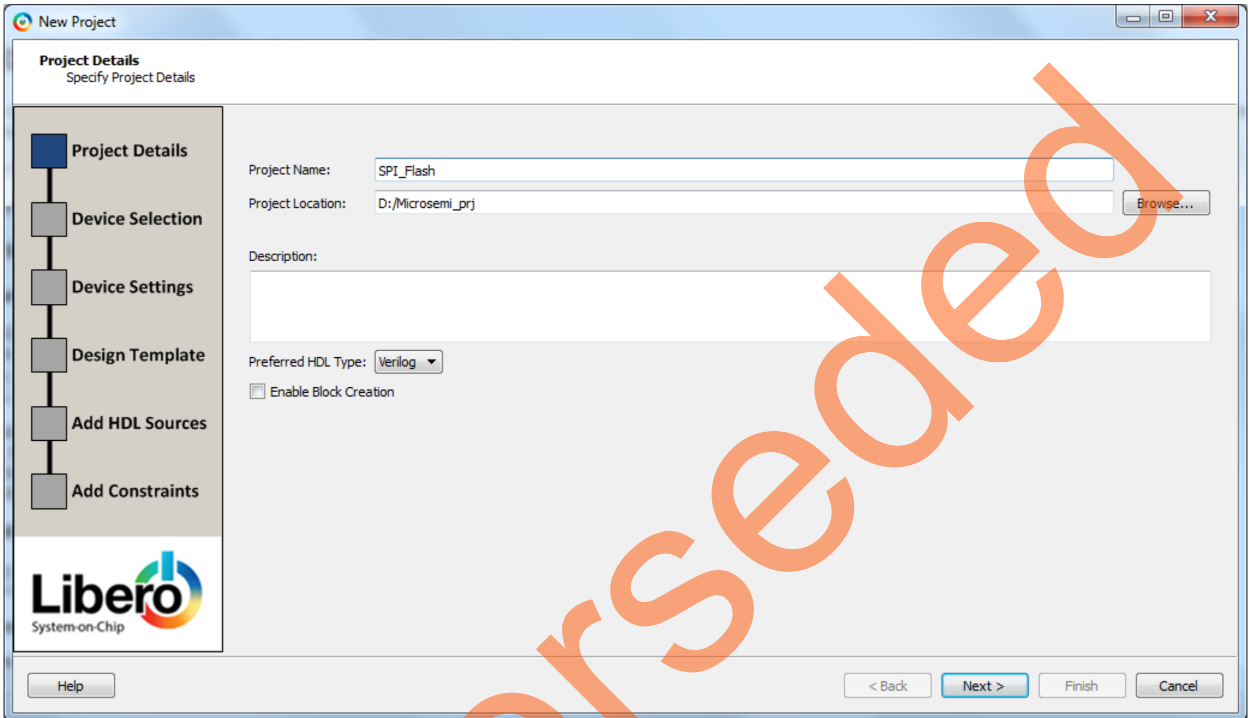

***Figure 3* • Project Details Page**

4. Click **Next**. The **Device Selection** page is displayed, as shown in Figure 4 on page 8. Select the following values from the drop-down list:
   – **Family**: SmartFusion2
   – **Die**: M2S090TS
   – **Package**: 484FBGA
   – **Speed**: -1
   – **Core Voltage**: 1.2
   – **Range**: COM

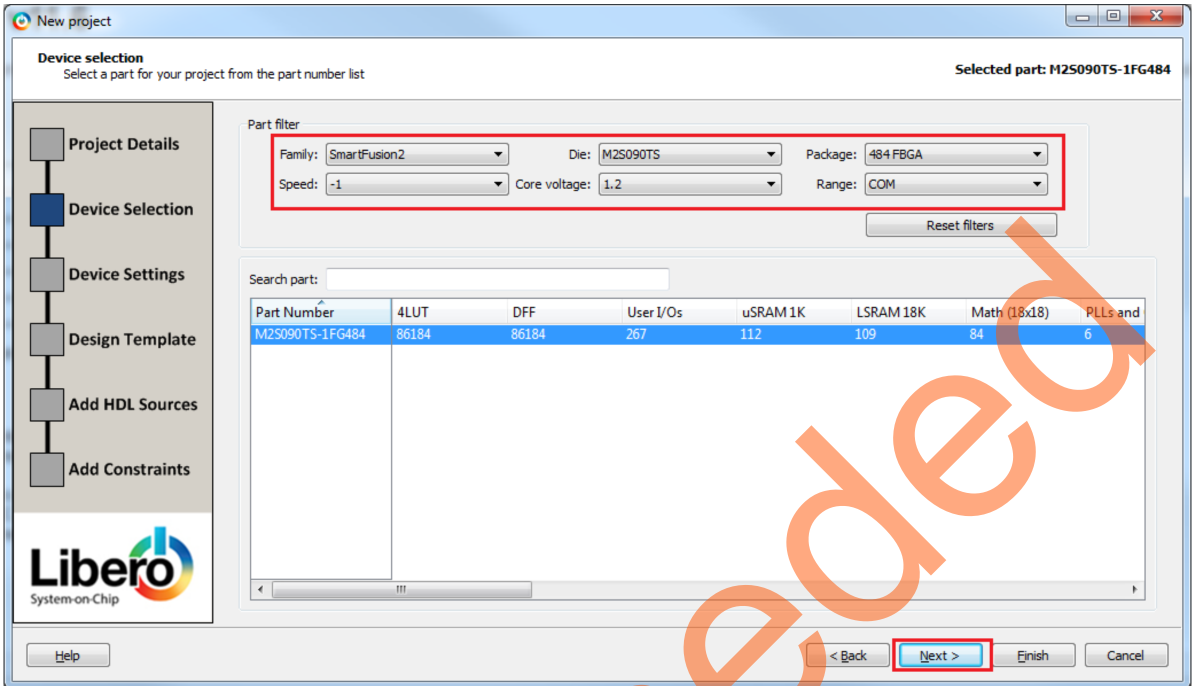

Figure 4 **• Device Selection Page**

5. Click **Next**. The **Device Settings** page is displayed.
6. Select **PLL supply voltage (V)** as **3.3** as shown in Figure 5 and click **Next**.

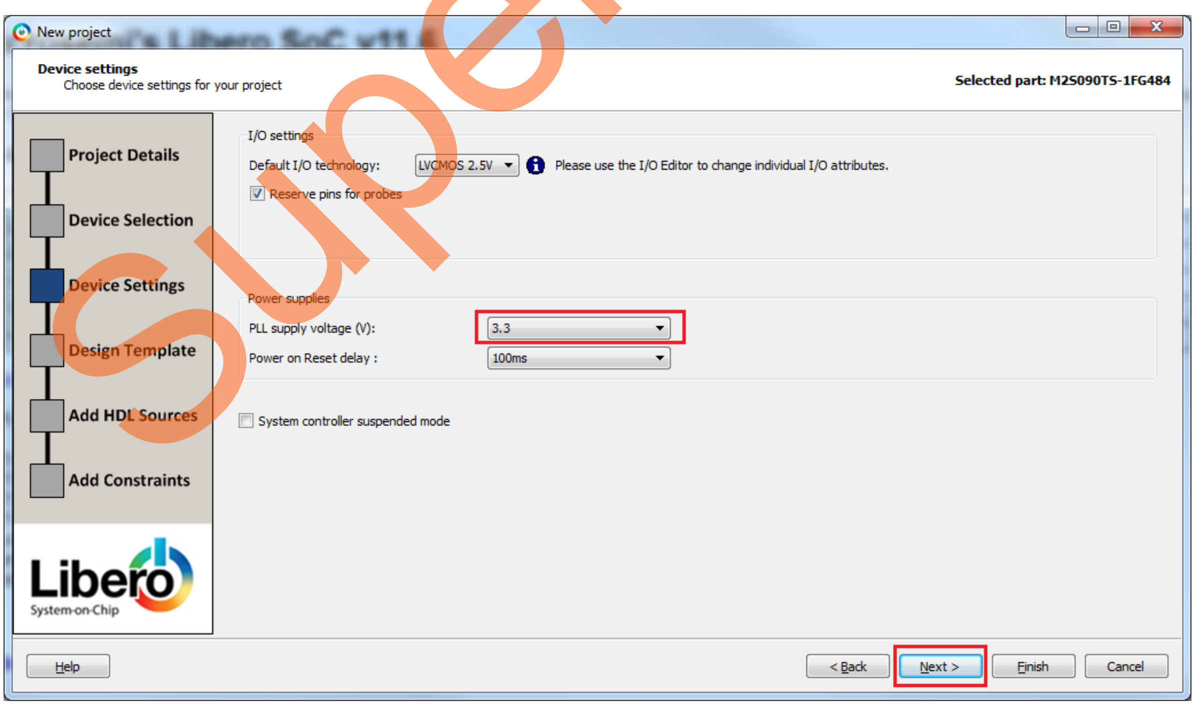

Figure 5 **• Device Settings Page**

7. Click **Next**. The **Design Template** page is displayed, as shown in Figure 6. Under Design Templates and Creators, click **Create a system builder based design**.

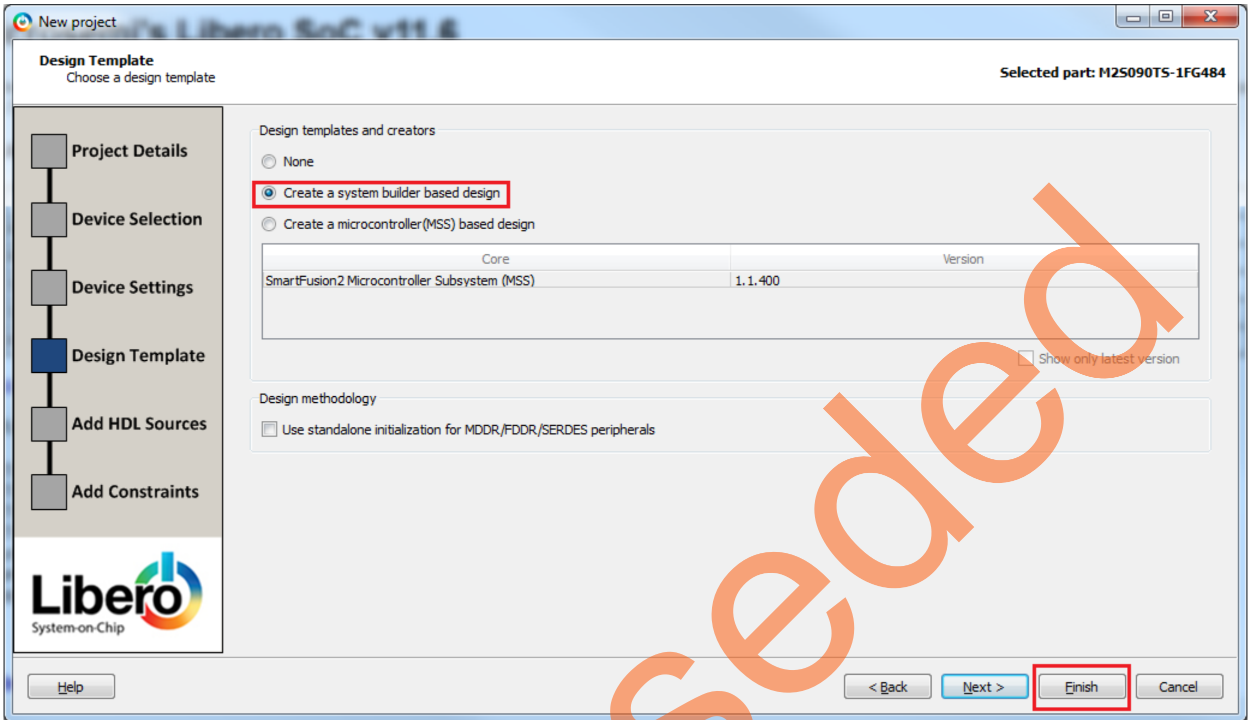

***Figure 6* • Design Template Window**

8. Click **Finish.** The **System Builder** window 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 to create the intended system.

9. **Enter a name for your system** as **SPI_Flash** and click **OK.**, as shown in Figure 7.

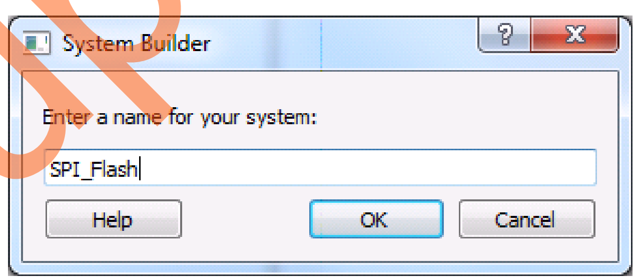

***Figure 7* • System Builder Window**

Figure 8 shows the **System Builder – Device Features** page.

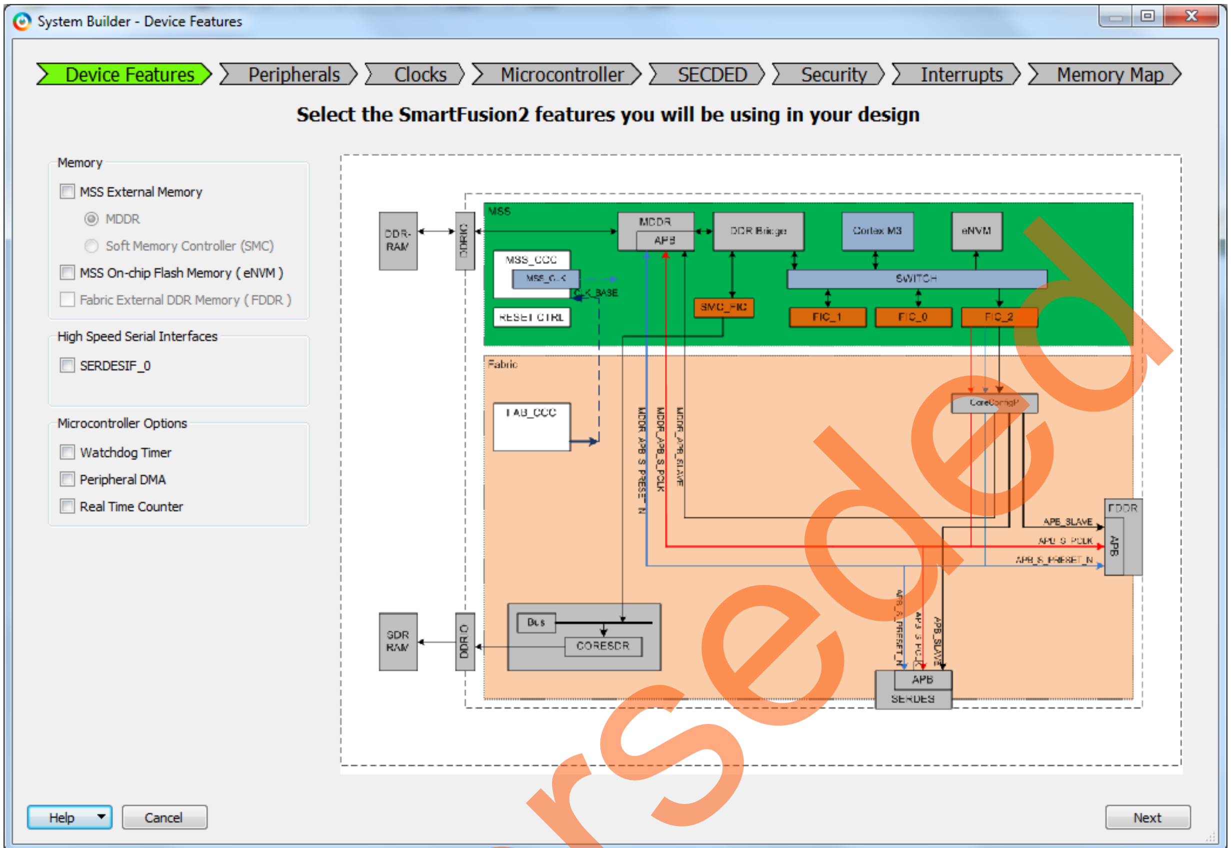

***Figure 8* • System Builder – Device Features Page**

10. Click **Next**. The **System Builder - Peripherals** page is displayed, as shown in Figure 9 on page 11.

11. Under the MSS Peripherals section, clear all the check boxes except **MM_UART_1** and **MSS_SPI_0**, as shown in Figure 9.

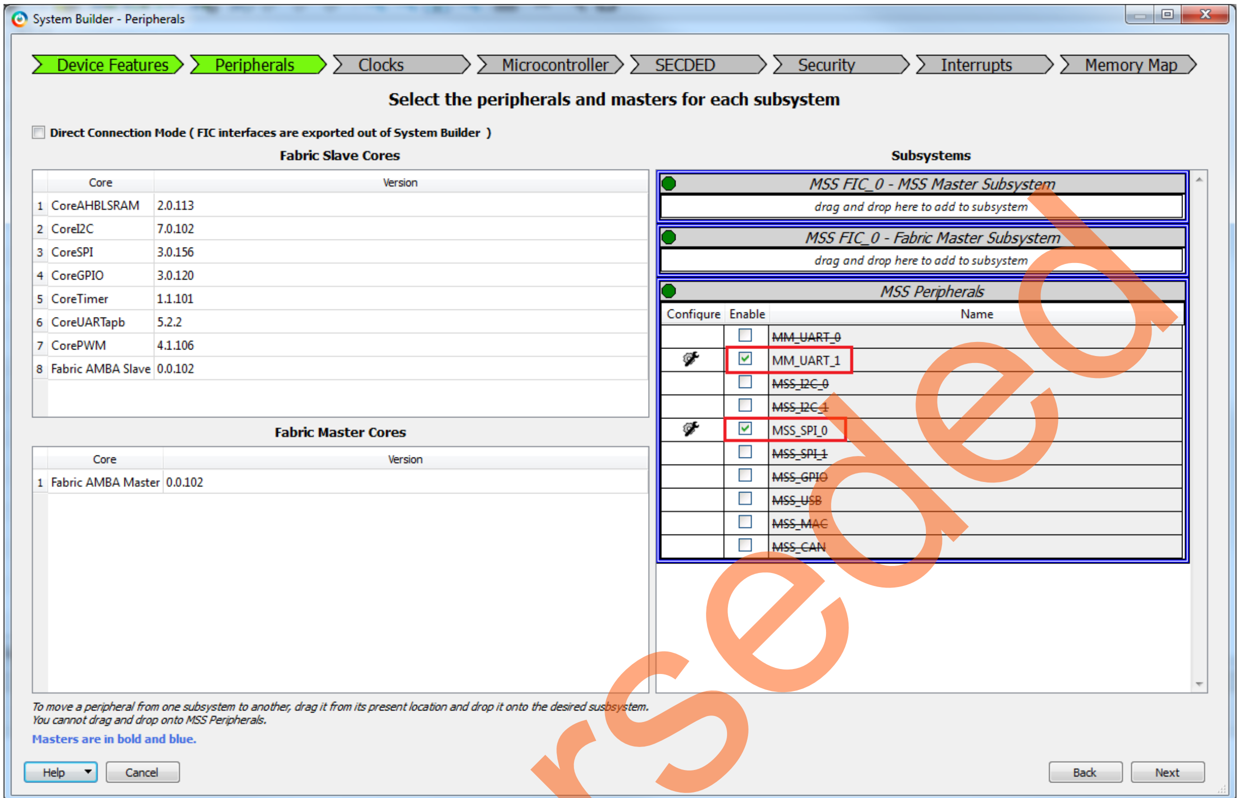

***Figure 9* • System Builder – Peripherals Page**

12. Click **Next**. The **System Builder - Clocks** page is displayed, as shown in Figure 10 on page 12.
13. In the **System Builder - Clocks** page (see Figure 10 on page 12):
    - Select **System Clock** frequency as **50 MHz** and clock source as **On-chip 25/50 MHz RC Oscillator**
    - Select **M3_CLK** as **100 MHz**
    - Select **APB_0_CLK** and **APB_1_CLK** frequency as **M3_CLK/1**

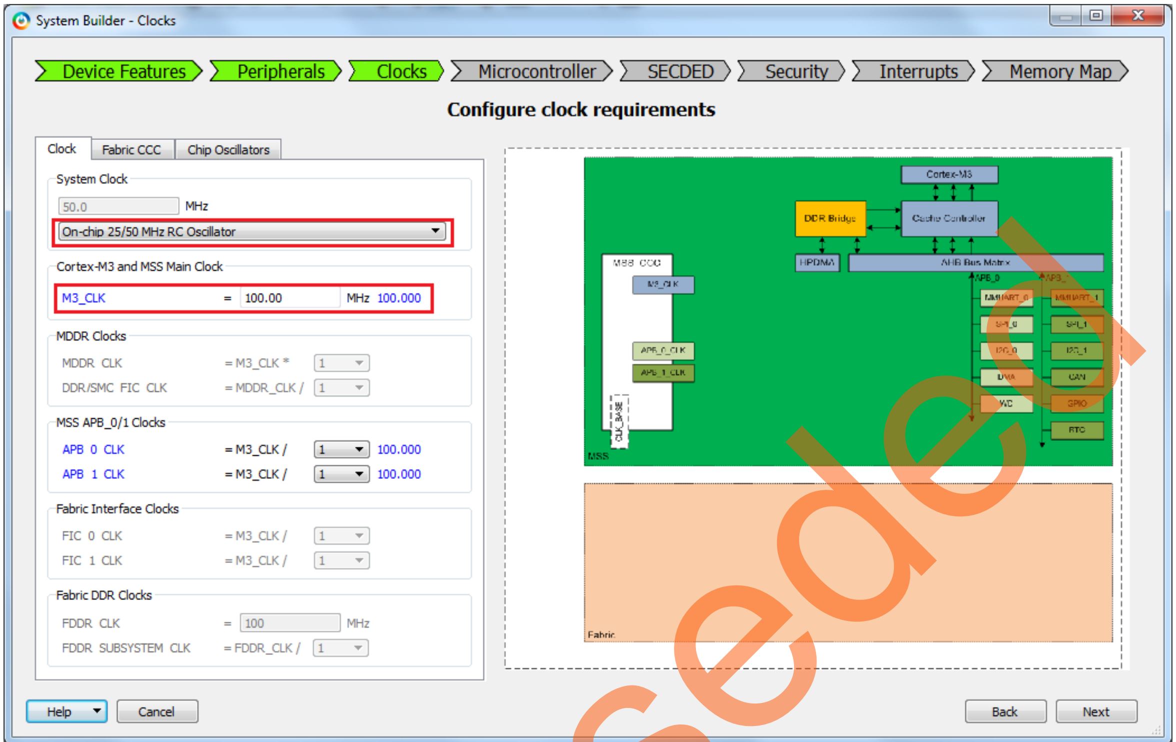

Figure 10 **• System Builder – Clocks Page**

14. Click **Next**. The **System Builder - Microcontroller** page is displayed. Do not change the default selections.
15. Click **Next**. The **System Builder - SECDED** page is displayed. Do not change the default selections.
16. Click **Next**. The **System Builder - Security** page is displayed. Do not change the default selections.
17. Click **Next**. The **System Builder - Interrupts** page is displayed. Do not change the default selections.
18. Click **Next**. The **System Builder - Memory Map** page is displayed. Do not change the default selections.
19. Click **Finish**.

20. Select **File > Save** to save **SPI_Flash**. Select the **SPI_Flash** tab on the Smart Design canvas, as shown in Figure 11.

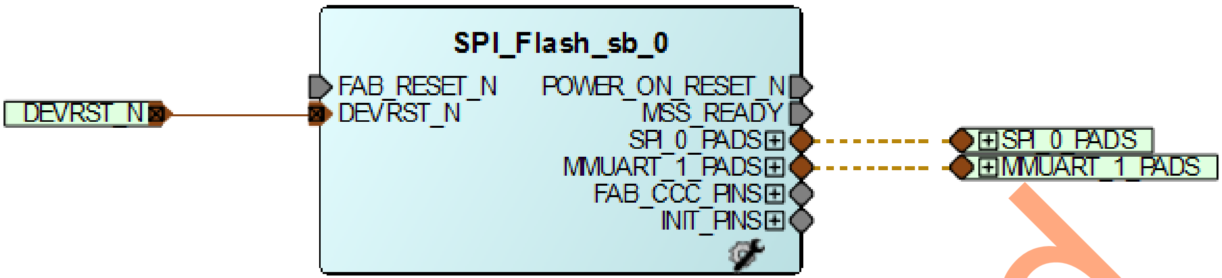

Figure 11 • **SPI_Flash SmartDesign**

Connecting Components in SPI_Flash_0 SmartDesign

The following steps describe how to connect the components in the **SPI_Flash** SmartDesign:

1. Right-click **POWER_ON_RESET_N** and select **Mark Unused**.
2. Right-click **MSS_READY** and select **Mark Unused**.
3. Expand **INIT_PINS**, right-click **INIT_DONE** and select **Mark Unused**.
4. Expand **FAB_CCC_PINS**, right-click **FAB_CCC_GL0** and select **Mark Unused**.
5. Right-click **FAB_CCC_LOCK** and select **Mark Unused**.
6. Right-click **FAB_RESET_N** and select **Tie High**.
7. Click **File > Save**.

   The SPI_Flash design is displayed, as shown in Figure 12.

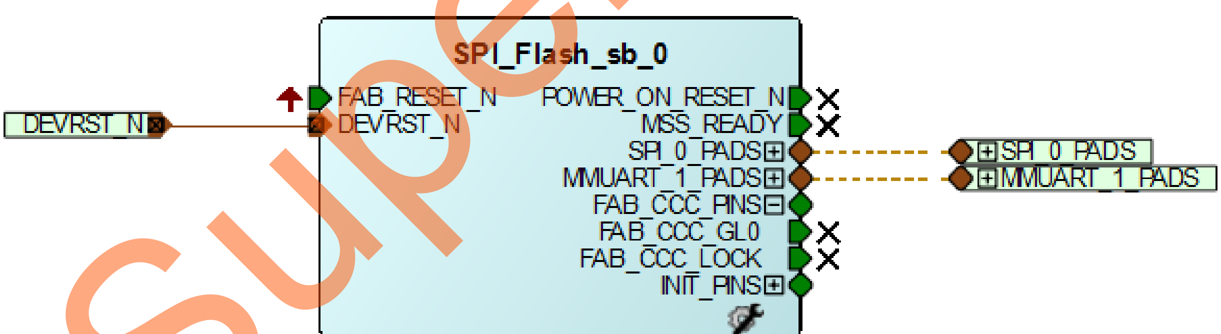

Figure 12 • **SPI_Flash SmartDesign**

8. Generate the SPI_Flash SmartDesign by clicking **SmartDesign > Generate Component** or by clicking **Generate Component** on the SmartDesign toolbar, as shown in Figure 13.

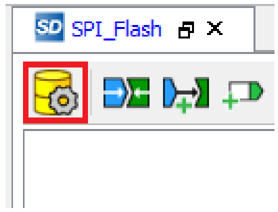

***Figure 13* • Generate Component**

After successful generation of all the components, the following message is displayed on the log window, as shown in Figure 14.

```
Info: 'SPI_Flash' was successfully generated.
```

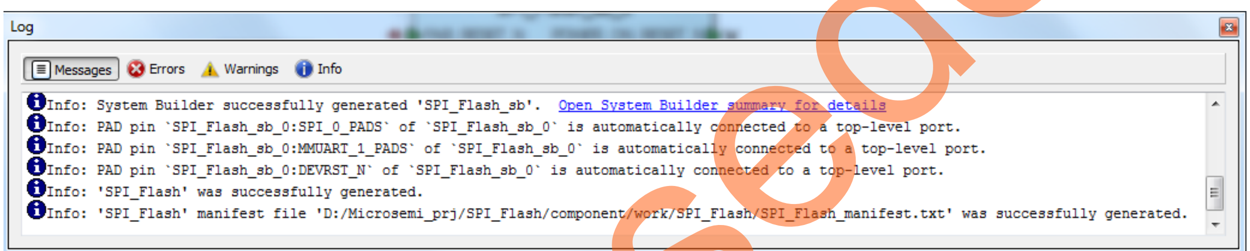

***Figure 14* • Log Window**

Step 2: Generating the Program File

The following step describe how to generate the program file:

Click **Generate Bitstream** as shown in Figure 15 to generate the programming file.

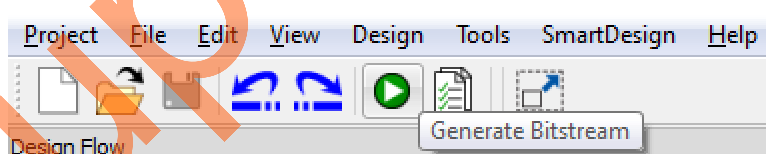

***Figure 15* • Generate Bitstream**

Step 3: Programming the SmartFusion2 Security Evaluation Board Using FlashPro

The following steps describe how to program the SmartFusion2 Security Evaluation Kit 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. For more information on jumper locations, refer to the "Appendix C: SmartFusion2 Security Evaluation Kit Board Jumper Locations" on page 44.

   CAUTION: Ensure that the power supply switch, **SW7** is switched OFF while connecting the jumpers on the SmartFusion2 Security Evaluation Kit.

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. Ensure that these jumpers are set accordingly.

3. Connect the power supply to the J6 connector.

   Switch **ON** the power supply switch, SW7. Refer to "Appendix B: Board Setup for Programming the Tutorial" on page 43 for information on the board setup for running the tutorial.
4. To program the SmartFusion2 device, double-click **Run PROGRAM Action** in the **Design Flow** tab, as shown in Figure 16.

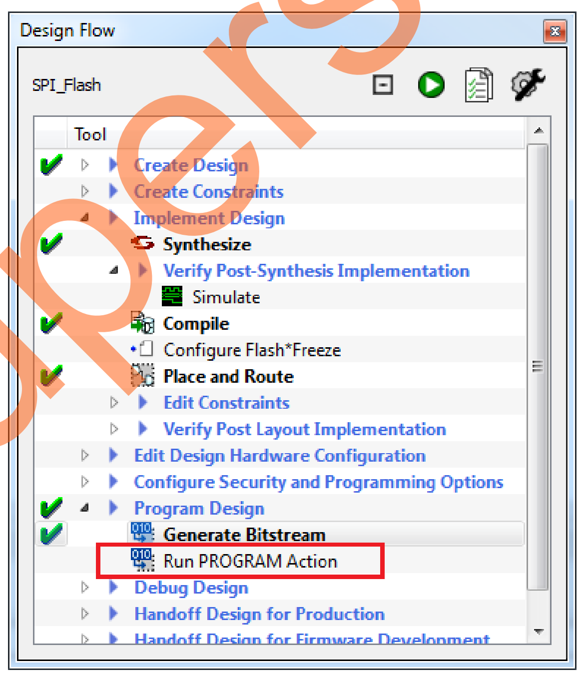

Figure 16 • **Run Program Action**

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 on **Configure Firmware Cores** in **Handoff design for Firmware Development** in **Design Flow** window.
2. Clear all the drivers check boxes, except **SmartFusion2_CMSIS_0**, **SmartFusion2_MSS_MMUART_Driver_0**, and **SmartFusion2_MSS_SPI_Driver_0**, as shown in Figure 17.

Note: Select the latest version of the drivers.

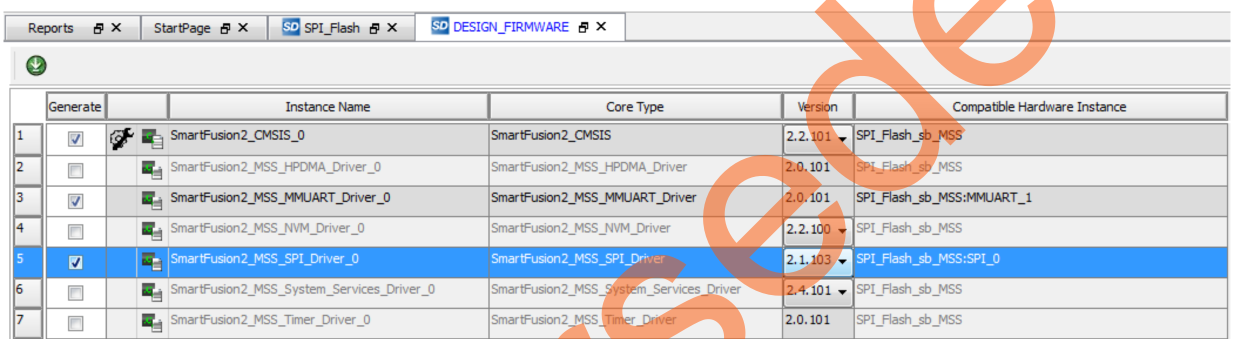

Figure 17 • **Configuring Firmware**

3. Double-click on **Export Firmware** in **Handoff design for Firmware Development** in **Design Flow** window.

   Export Firmware dialog box is displayed as shown in Figure 18.

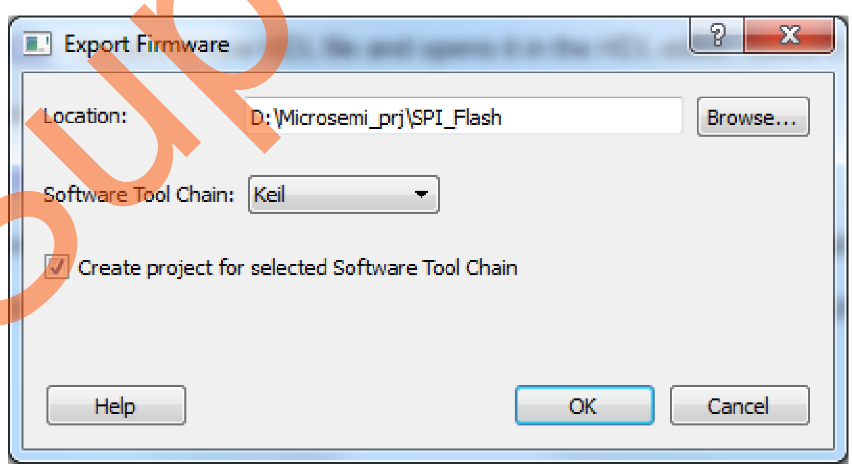

Figure 18 • **Export Firmware Dialog Box**

4. In the **Export Firmware** dialog box:
   – Select **Create project for selected Software Tool Chain**.
   – Select **Keil** from the drop-down list.
5. Click **OK**. The successful firmware generation window is displayed.

The SmartFusion2 Security Evaluation Kit is ready for running and debugging the Keil application through ULINK-ME Debugger.

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

The following steps describe how to build a software application using Keil uVision 5 IDE:

1. Launch the Keil IDE. Open the Keil project by double-clicking SPI_Flash_sb_MSS_CM3 Keil project, as shown in Figure 19.

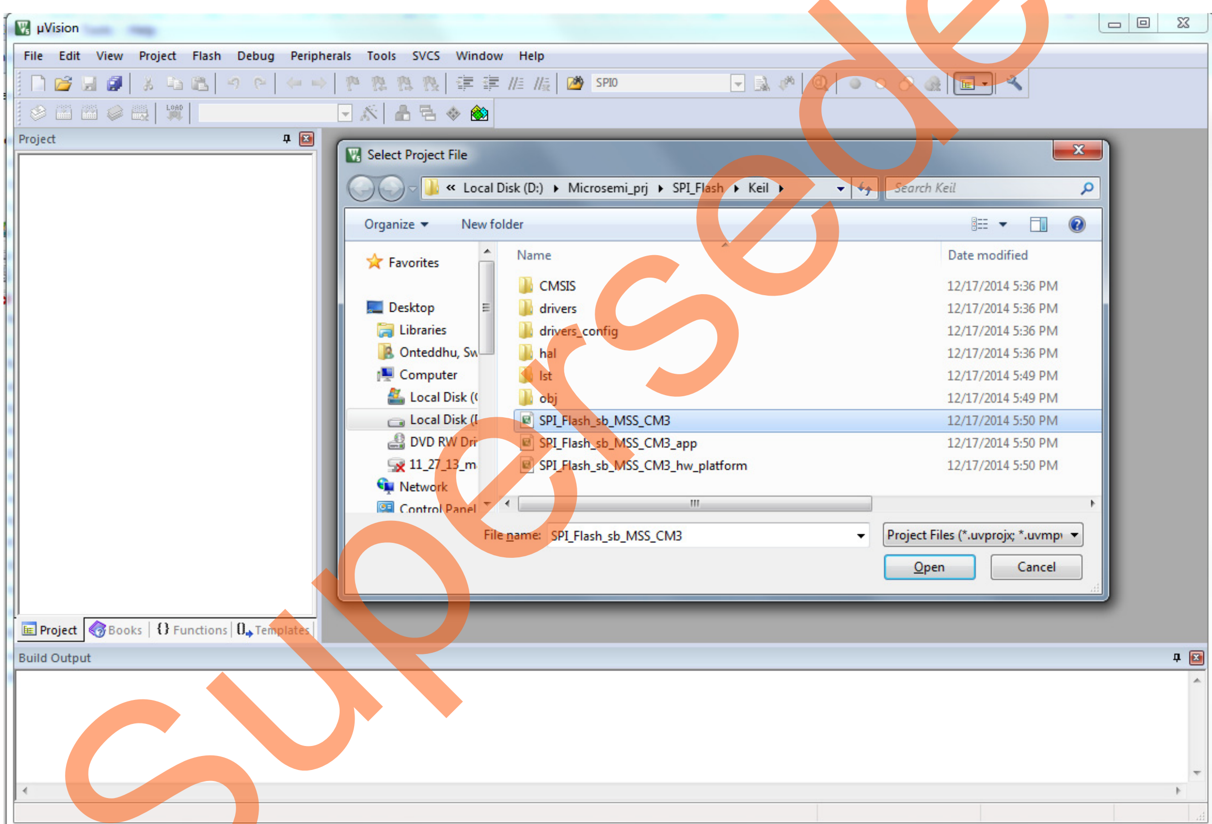

***Figure 19* • Keil Homepage**

2. The Keil workspace is displayed, as shown in Figure 20.

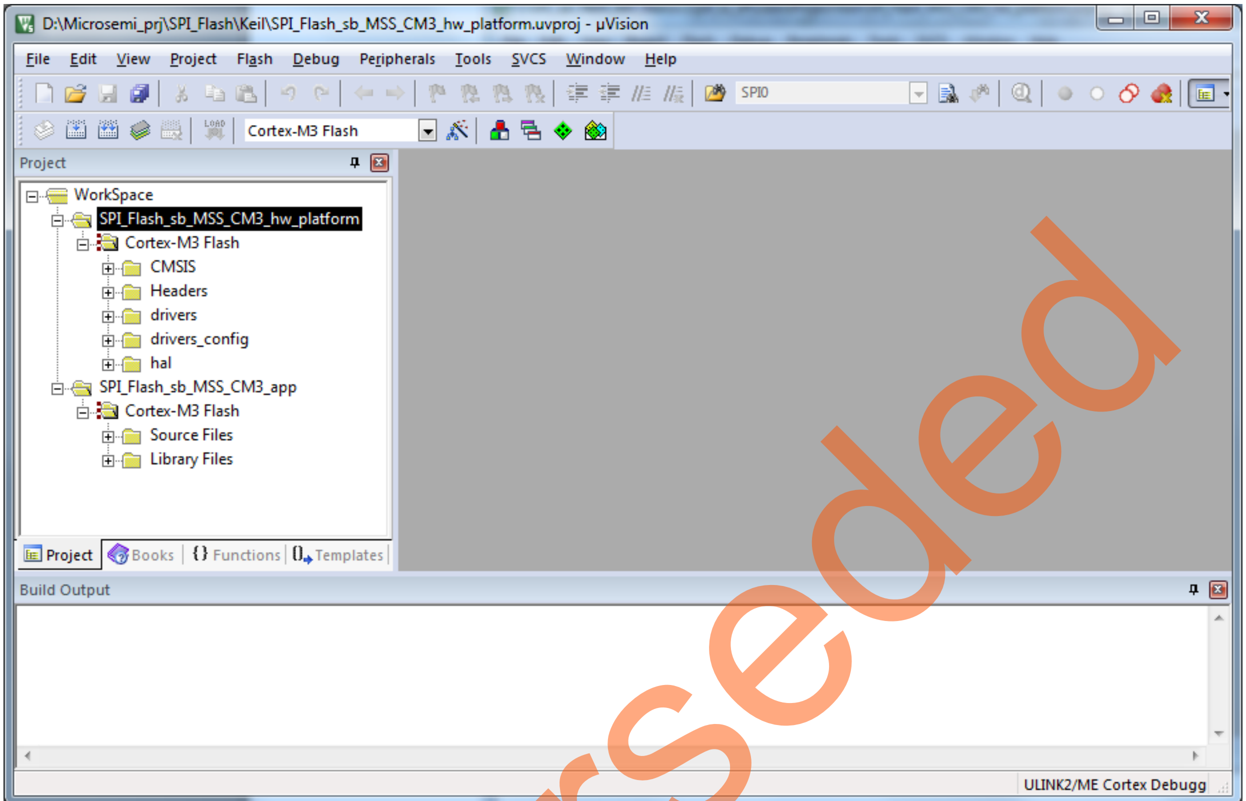

Figure 20 **• uVision Workspace**

3. Browse to the `main.c` file location in the design files folder: *<download_folder>/SF2_SPI_Flash_Keil_Tutorial_DF\SourceFiles*.
4. Copy the `main.c` file and replace the existing `main.c` file under SPI_Flash_sb_MSS_CM3_app project in the uVision workspace.

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

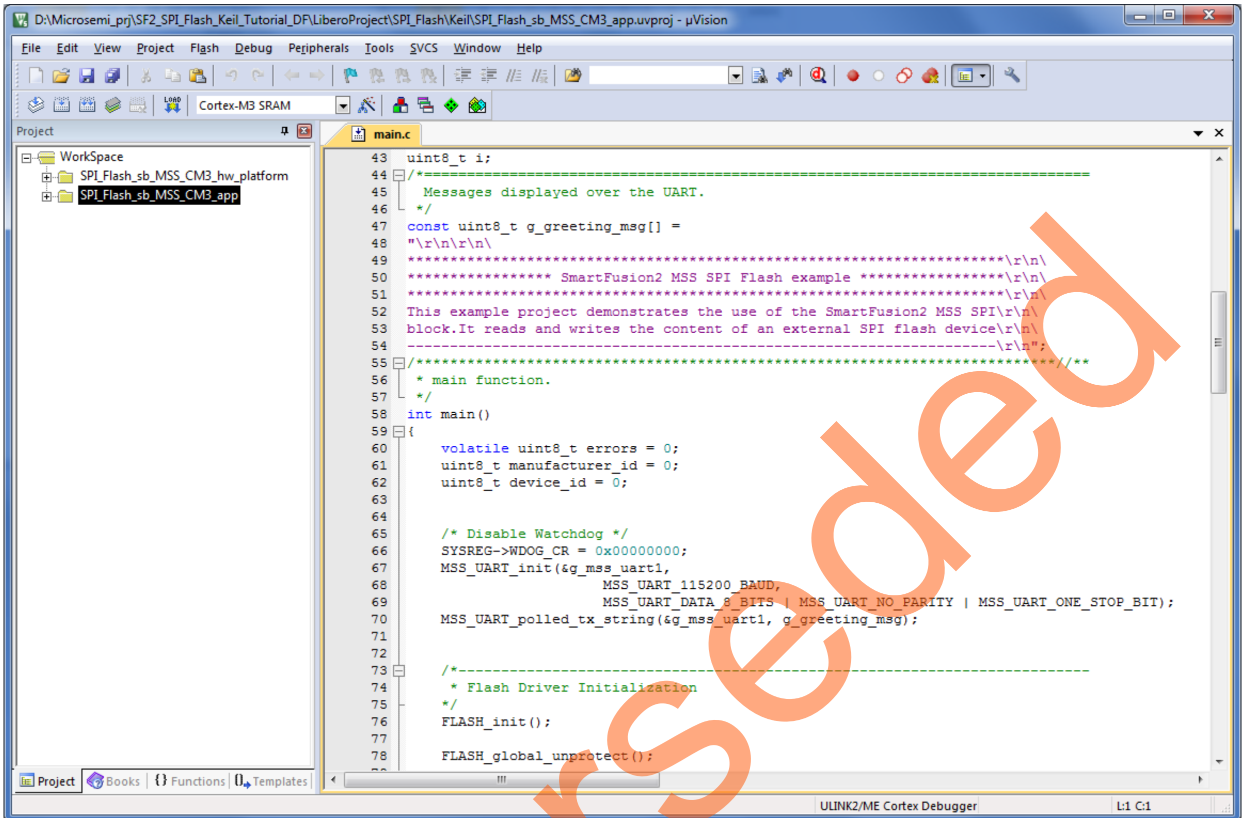

***Figure 21* • uVision Workspace main.c file**

5. 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 winbondflash drivers in the design files folder:
   <download_folder>\SF2_SPI_Flash_Keil_Tutorial_DF\SPI_Flash_Drivers.
6. Copy the **winbond flash** folder to the drivers folder of SPI_Flash_sb_MSS_CM3_hw_platform project in the uVision workspace.

7. Right-click and add the driver file (`winbondflash.c`) to **SPI_Flash_sb_MSS_CM3_hw_platform** project in the Keil uVision workspace, as shown in Figure 22.

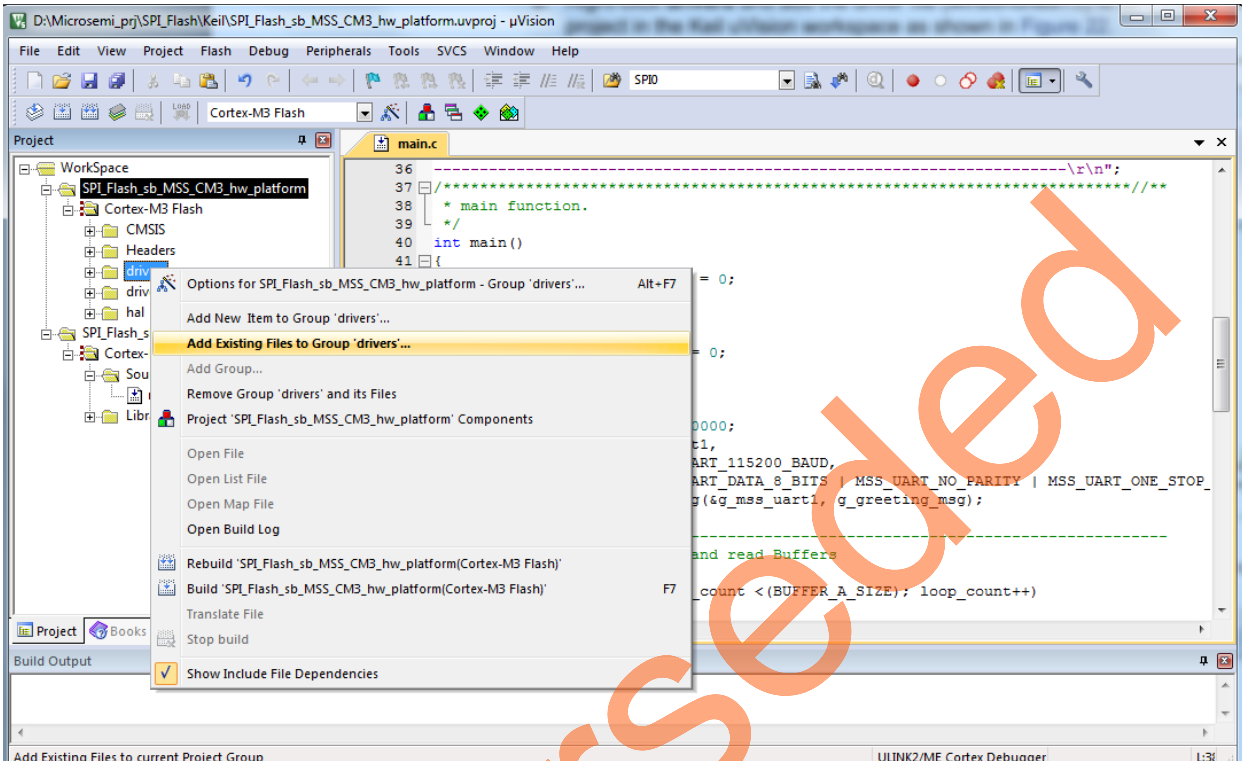

Figure 22 **• uVision Workspace Window - Add winbondflash SPI Driver Files**

8. Change **SPI_Flash_sb_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.

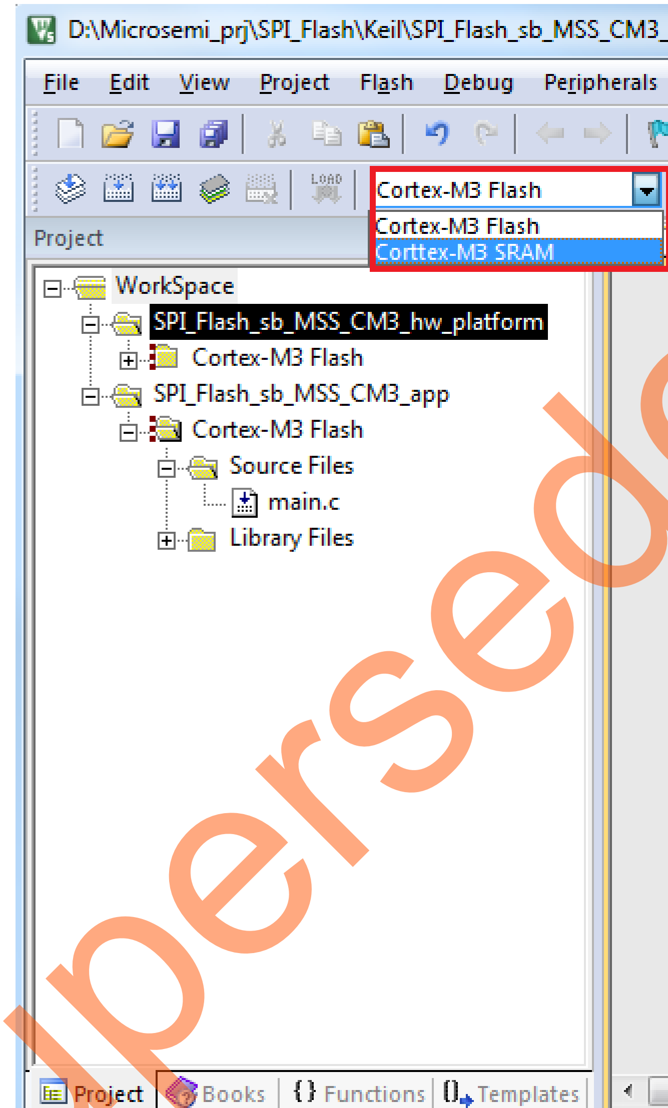

***Figure 23* • Cortex-M3_SRAM Settings**

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 to add MICROSEMI_STDIO_THRU_UART symbol:

a. Right-click **Cortex - M3 SRAM** under **SPI_Flash_sb_MSS_CM3_hw_platform** and click **Options for SPI_Flash_sb_MSS_CM3_hw_platform - Target Cortex - M3 SRAM**.

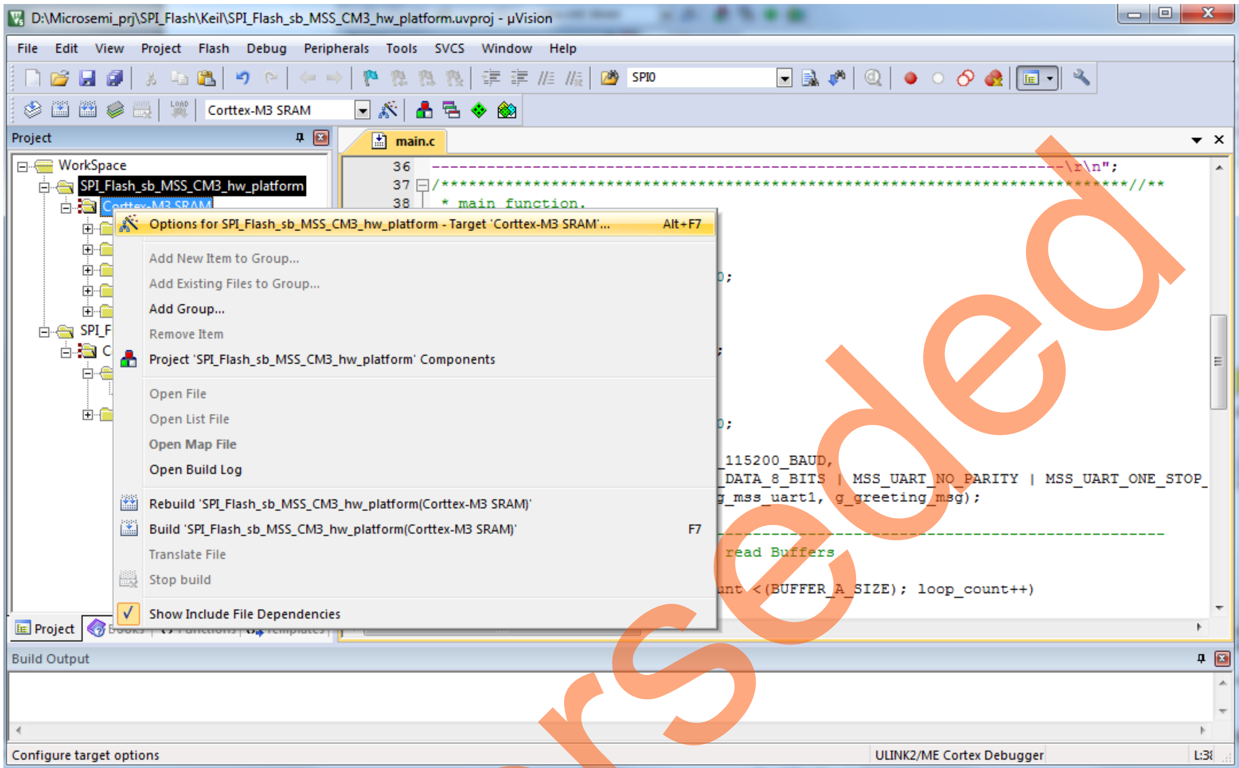

***Figure 24* • Target Options**

b. Go to **C/C++** tab and enter **MICROSEMI_STDIO_THRU_UART** at **Define** under Preprocessor Symbols, as shown in Figure 25 on page 23.

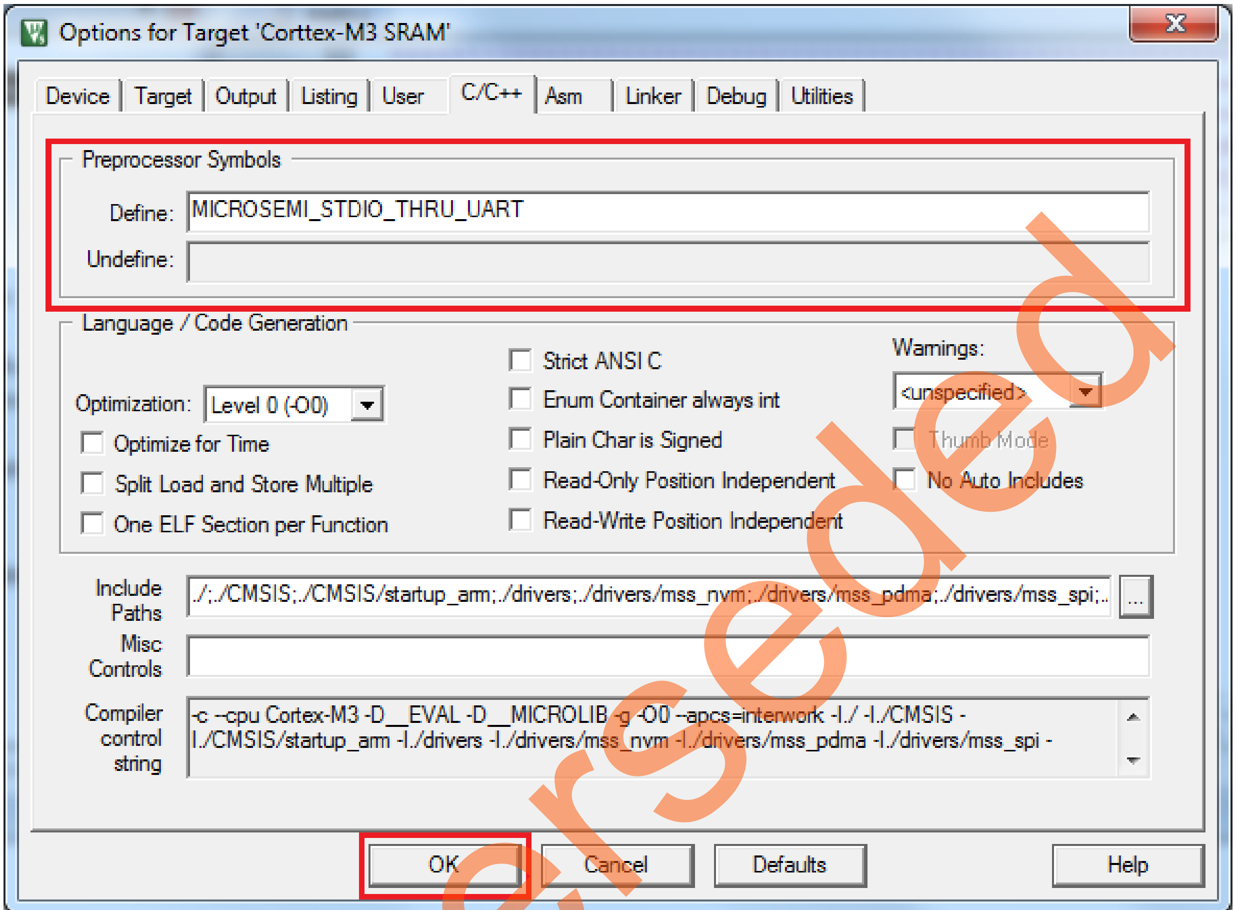

***Figure 25* • Target Options-Adding Symbols**

c. Click **OK**.

9. Right-click **Cortex-CM3_SRAM** under **SPI_Flash_sb_MSS_CM3_hw_platform** and select **Build SPI_Flash_sb_MSS_CM3_hw_platform (Cortex-CM3 SRAM)**, as shown in Figure 26.

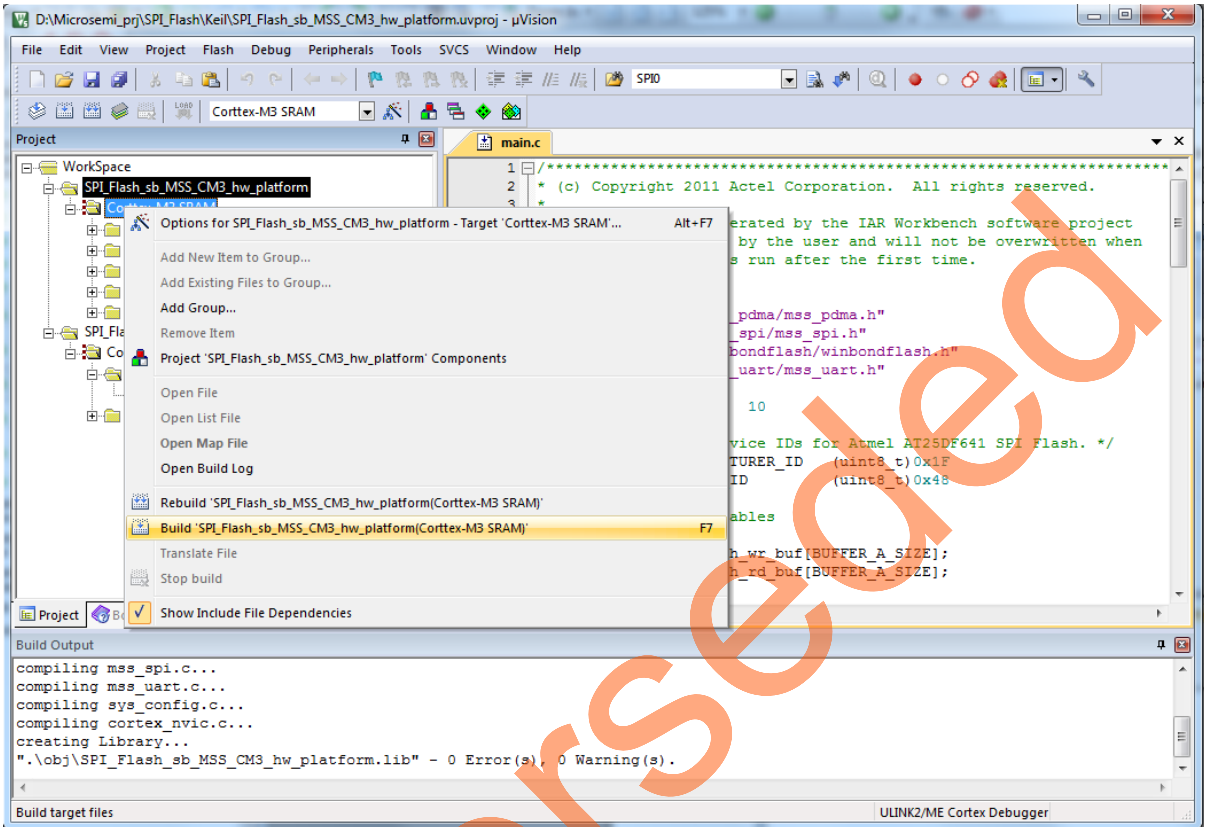

***Figure 26* • Build HW Platform Window**

10. Right-click **SPI_Flash_sb_MSS_CM3_app** and select **Set as Active Project**.

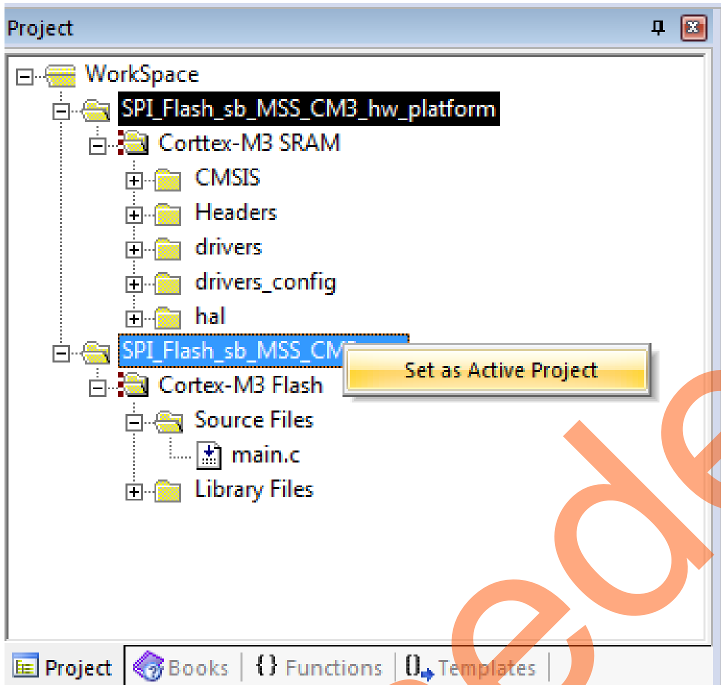

***Figure 27* • Set as Active Project**

11. Change **SPI_Flash_sb_MSS_CM3_app** debug mode to **Cortex-M3_SRAM** by selecting **Cortex-M3_SRAM** from the drop-down list, as shown in Figure 28.

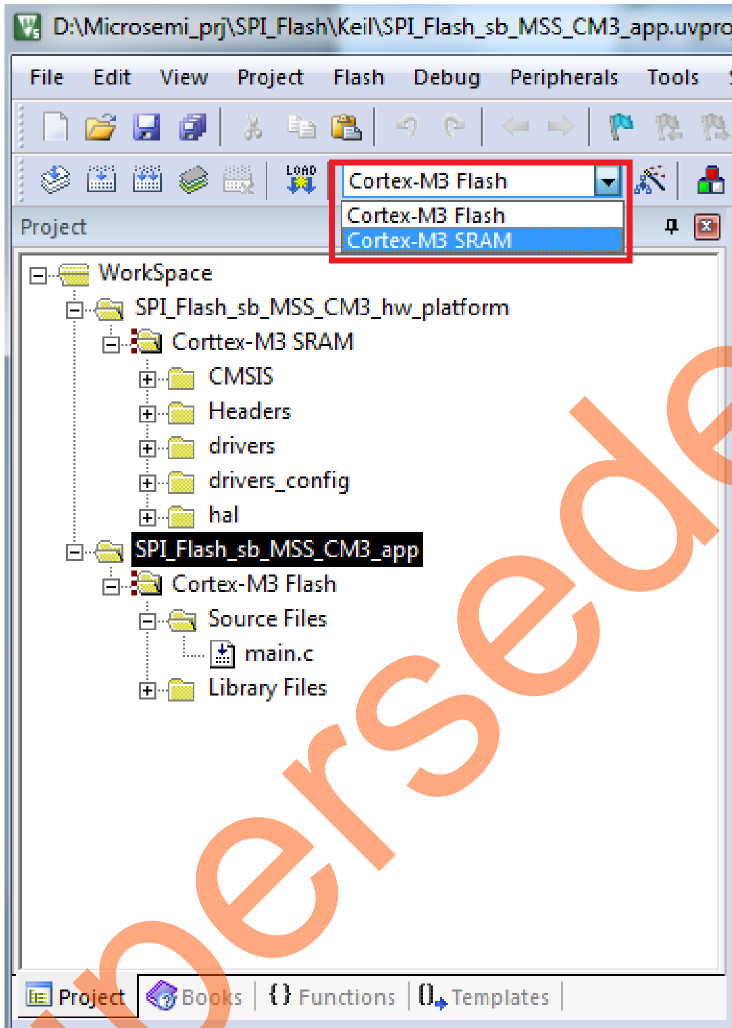

Figure 28 • **Cortex-M3_SRAM Settings**

12. Right-click **Cortex-M3 SRAM** under **SPI_Flash_sb_MSS_CM3_app** and click **Options for project**.

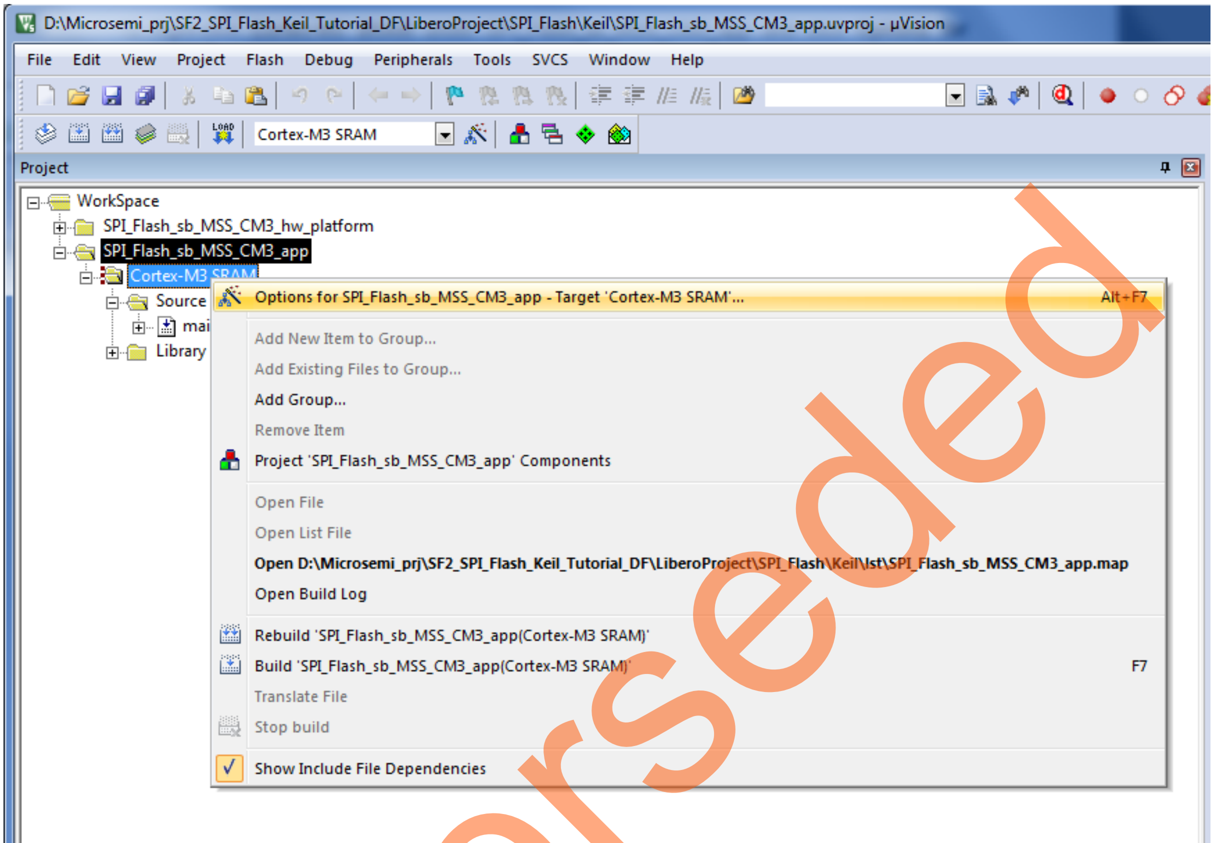

***Figure 29* • Target Options**

13. Click the **Linker** tab and navigate to the *SF2_SPI_Flash_Keil_Tutorial_DF\LiberoProject\Keil\CMSIS\startup_arm* folder to select the **Scatter File** as `smartfusion2_esram_debug.sct`, as shown in Figure 30.

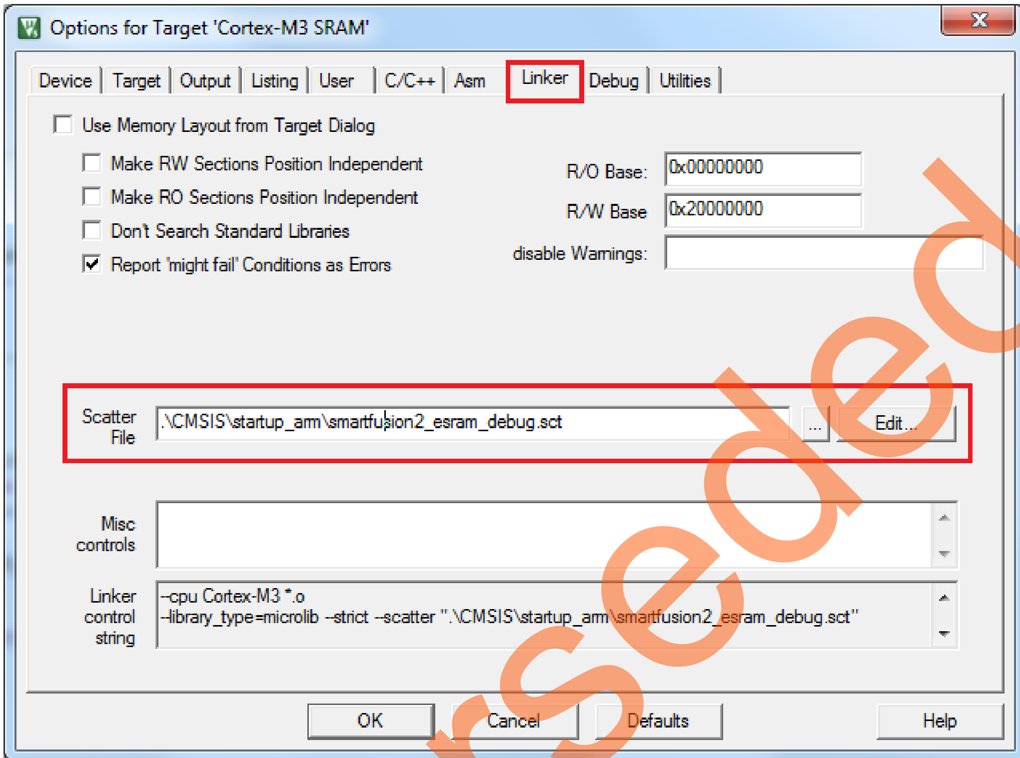

***Figure 30* • Target Options - Scatter File**

14. Click the **Utilities** tab and clear **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 31.

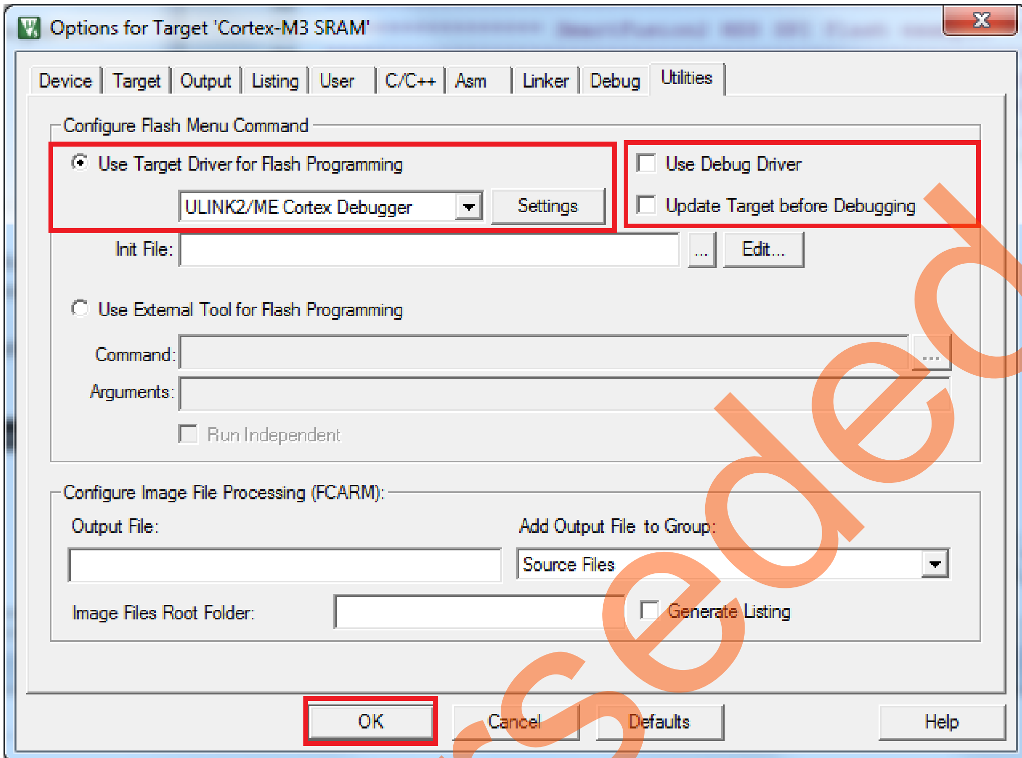

***Figure 31* • Target Options - Utilities Settings**

16. Right-click **Cortex-M3 SRAM** under **SPI_Flash_sb_MSS_CM3_app** and select **Build SPI_Flash_sb_MSS_CM3_app (Cortex-M3 SRAM)**, as shown in Figure 32 on page 30. It compiles all of the source files and links the object files into an AXF file to debug. Ensure that there are no errors. Correct syntax errors, if any and rebuild if necessary.

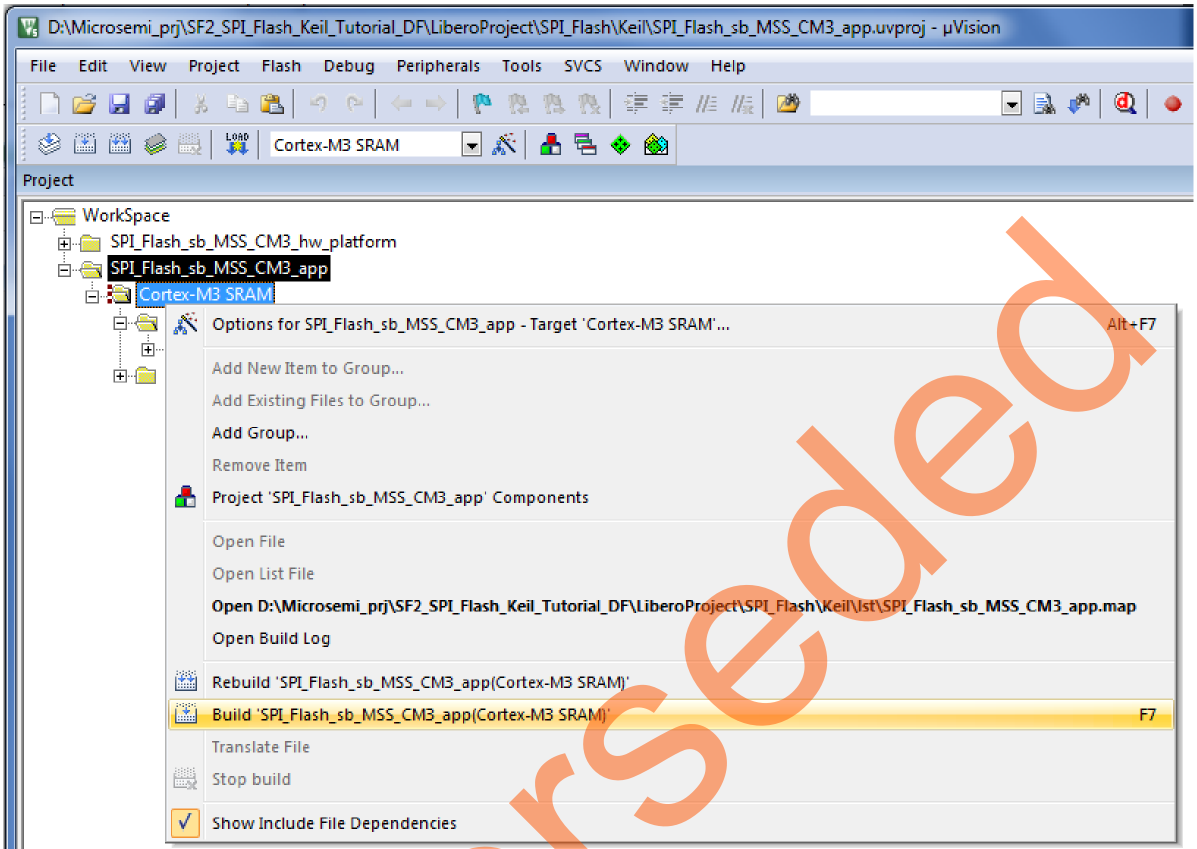

***Figure 32* • Build Application Window**

Figure 33 displays the messages in the console after the build.

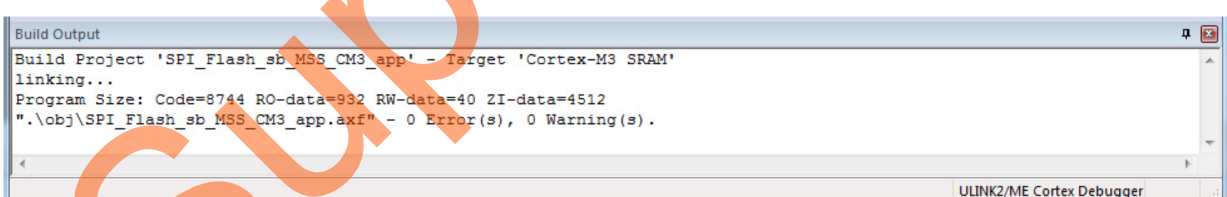

***Figure 33* • Build Output**

Step 6: Configuring Serial Terminal Emulation Program

The following steps describe how to configure 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 34 shows an example **Device Manager** window.

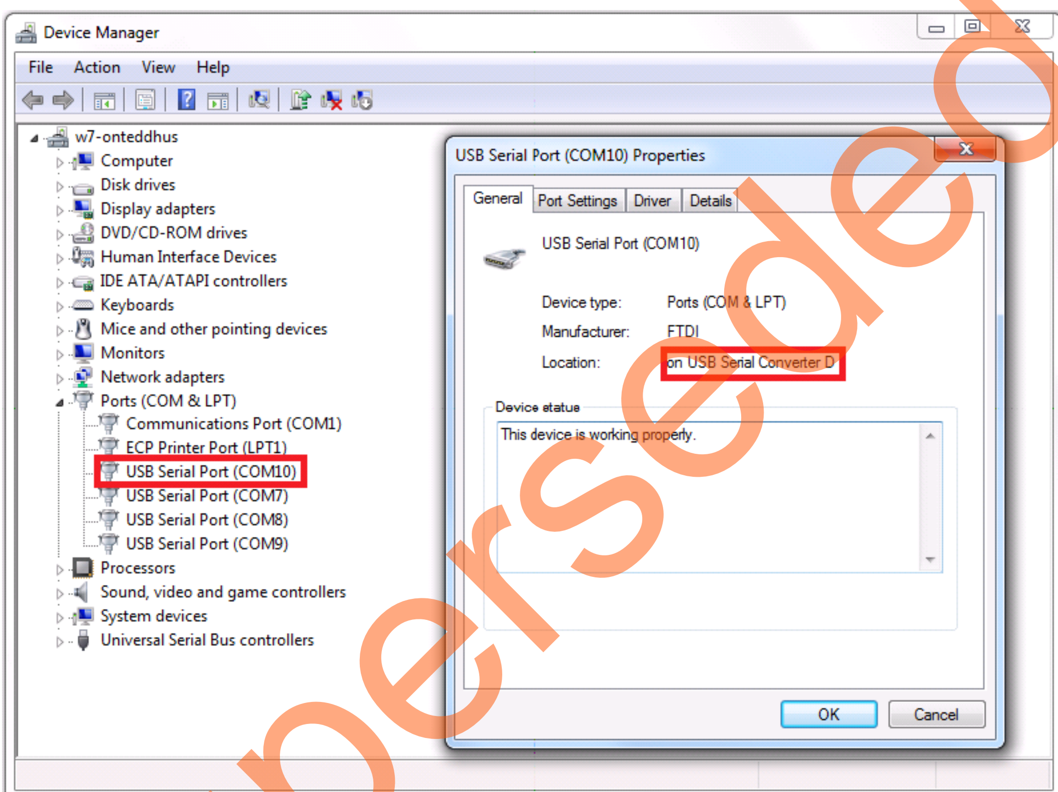

***Figure 34* • Device Manager Window**

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: Connecting the ULINK-ME to the Board and PC

The following steps describe the connection between the SmartFusion2 Security 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 Security Evaluation Kit board.
2. Connect the USB A-Mini B cable between the host PC and the SmartFusion2 Security Evaluation Kit board. This is used to display the HyperTerminal communications.
3. Verify that the ULINK-ME debugger is connected to the SmartFusion2 Security Evaluation Kit board RVI Header as shown in Figure 35 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.

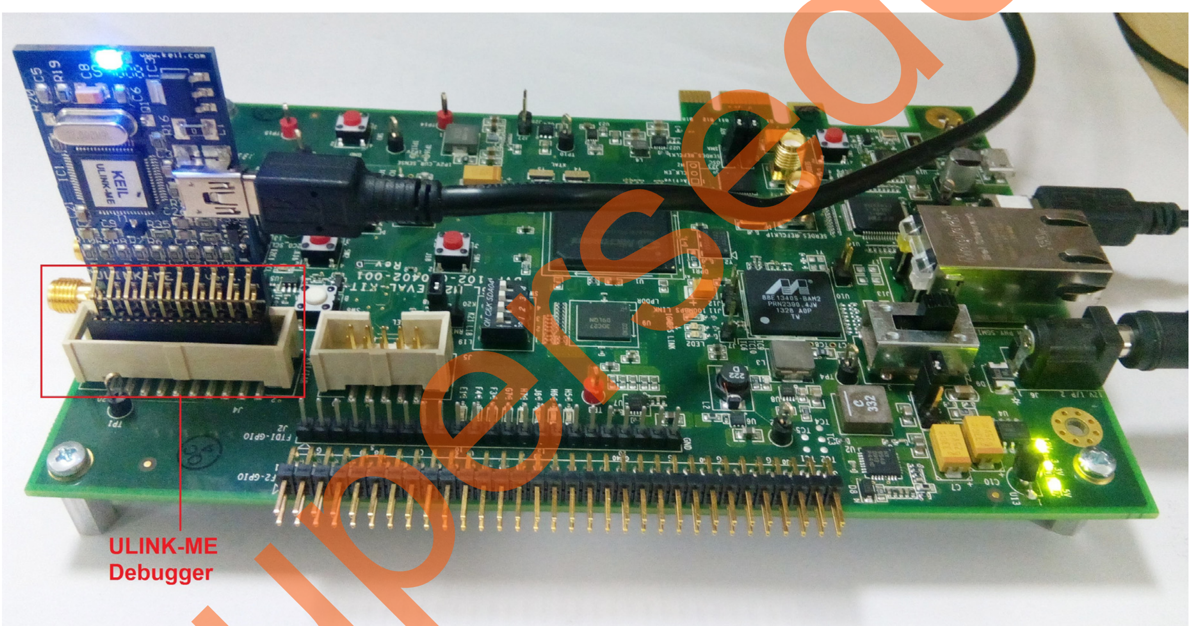

***Figure 35* • ULINK-ME Connections**

Refer to "Appendix A: Board Setup for Debugging from Keil uVision" on page 42 for information on the board setup for running the tutorial.

Figure 36 • **ULINK-ME Debugger**

Step 8: Debugging the Application Project using Keil uVision 5

The following steps describe how to debug the application project using Keil uVision:

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 37. The processor code is downloaded to the SmartFusion2 eSRAM.

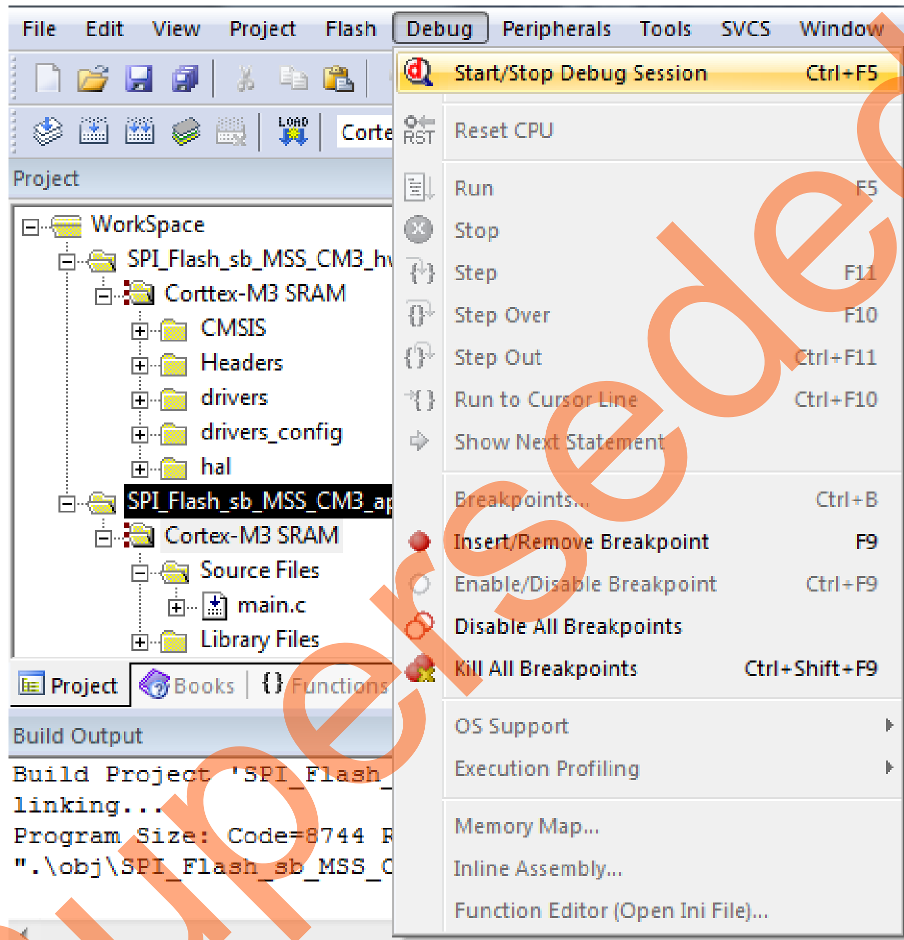

***Figure 37* • Selecting Start/Stop Debug Session**

The code automatically runs in the `main.c` file, as shown in Figure 38.

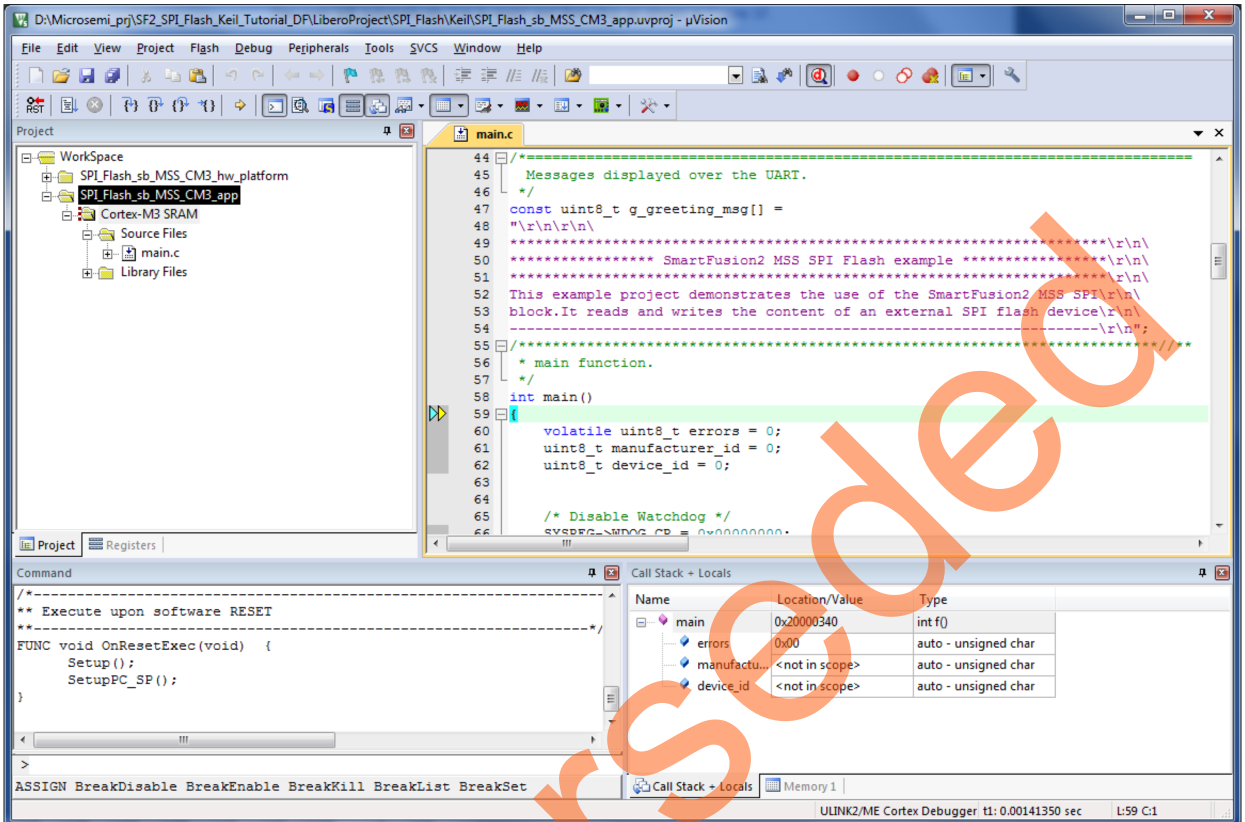

***Figure 38* • Debug Menu**

2. Click **Run** from the **Debug** menu, as shown in Figure 39.

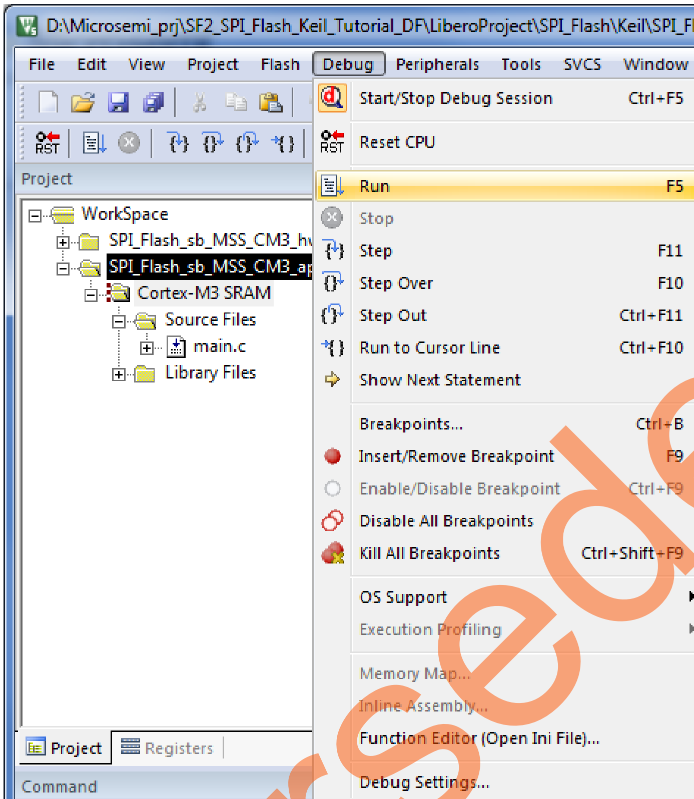

***Figure 39* • Selecting Run from the Debug Menu**

3. On successful operation, the HyperTerminal window displays a message as shown in Figure 40.

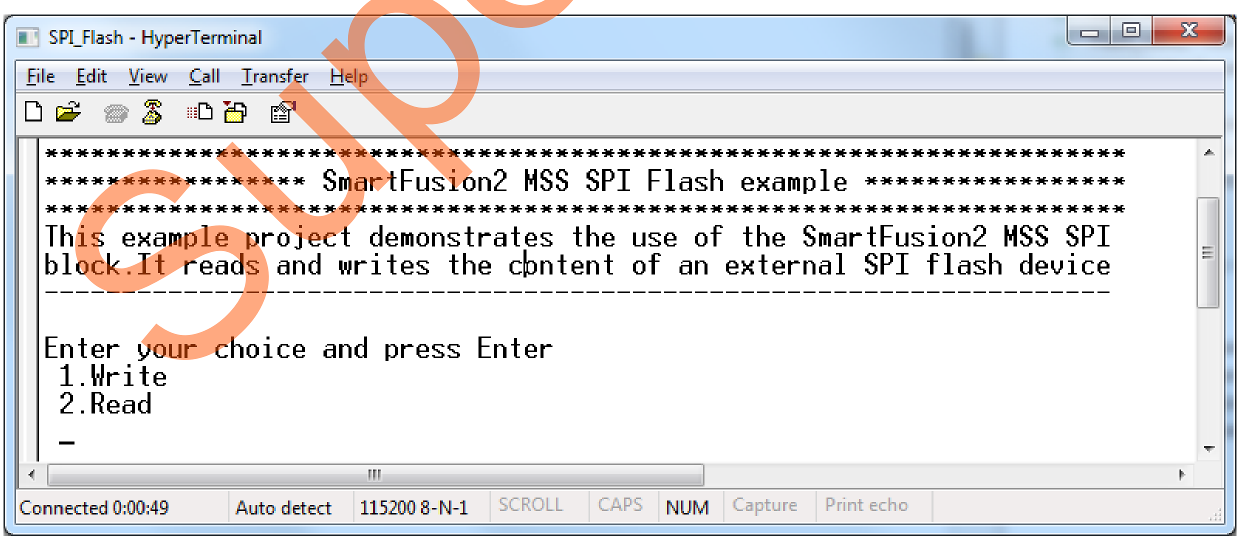

***Figure 40* • HyperTerminal Window**

4. Select option 1 and enter values to write to the SPI Flash Memory, as shown in Figure 41.

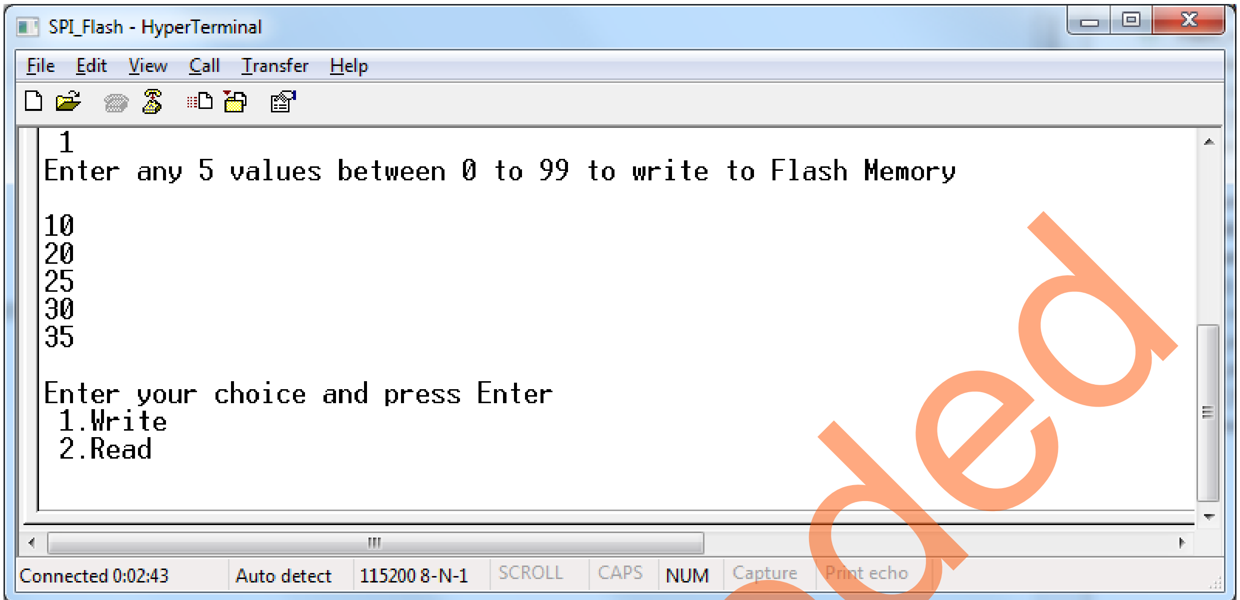

Figure 41 **• HyperTerminal Window - Option 1**

5. Select option 2 to read data from SPI Flash Memory, as shown in Figure 42.

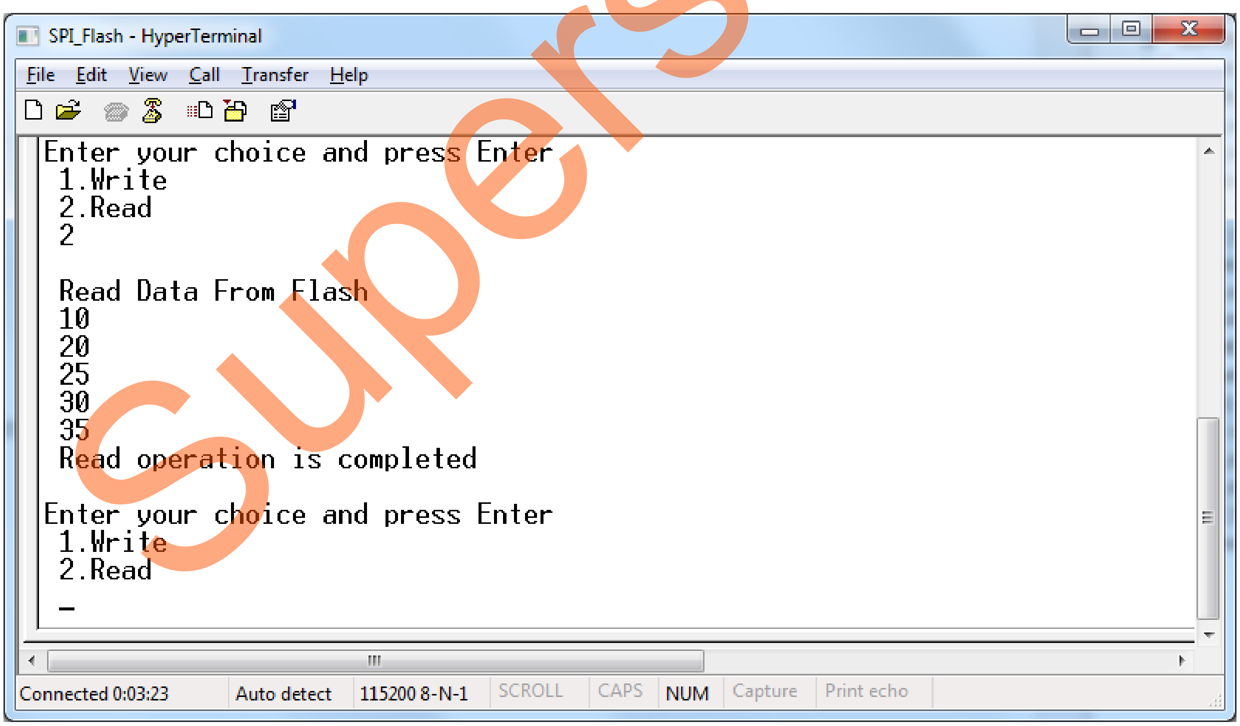

Figure 42 **• HyperTerminal Window - Option 2**

The **Disassembly** window is displayed in the middle of the **Debug** section as shown in Figure 43. If not, click the **Disassembly** icon to display the **Disassembly** section.

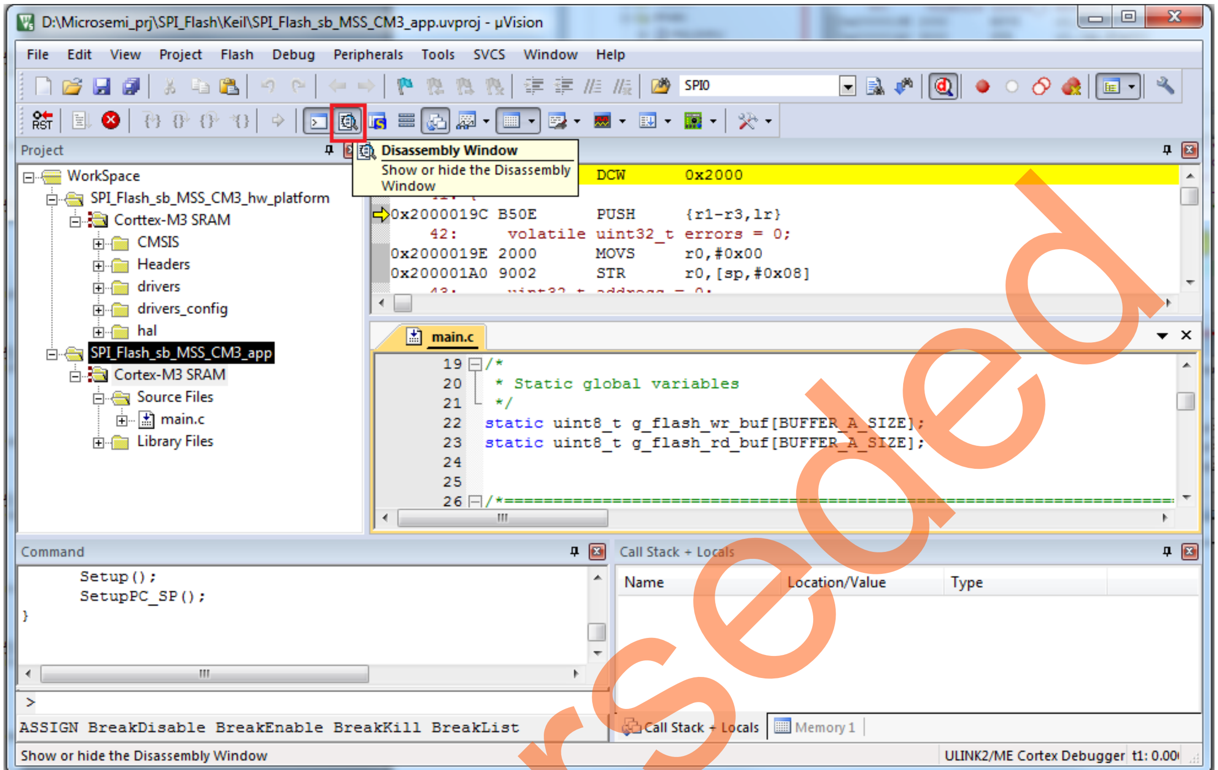

***Figure 43* • Disassembly Window**

6. Click **Registers Window** to view the values of the ARM® Cortex®-M3 processor internal registers, as shown in Figure 44.

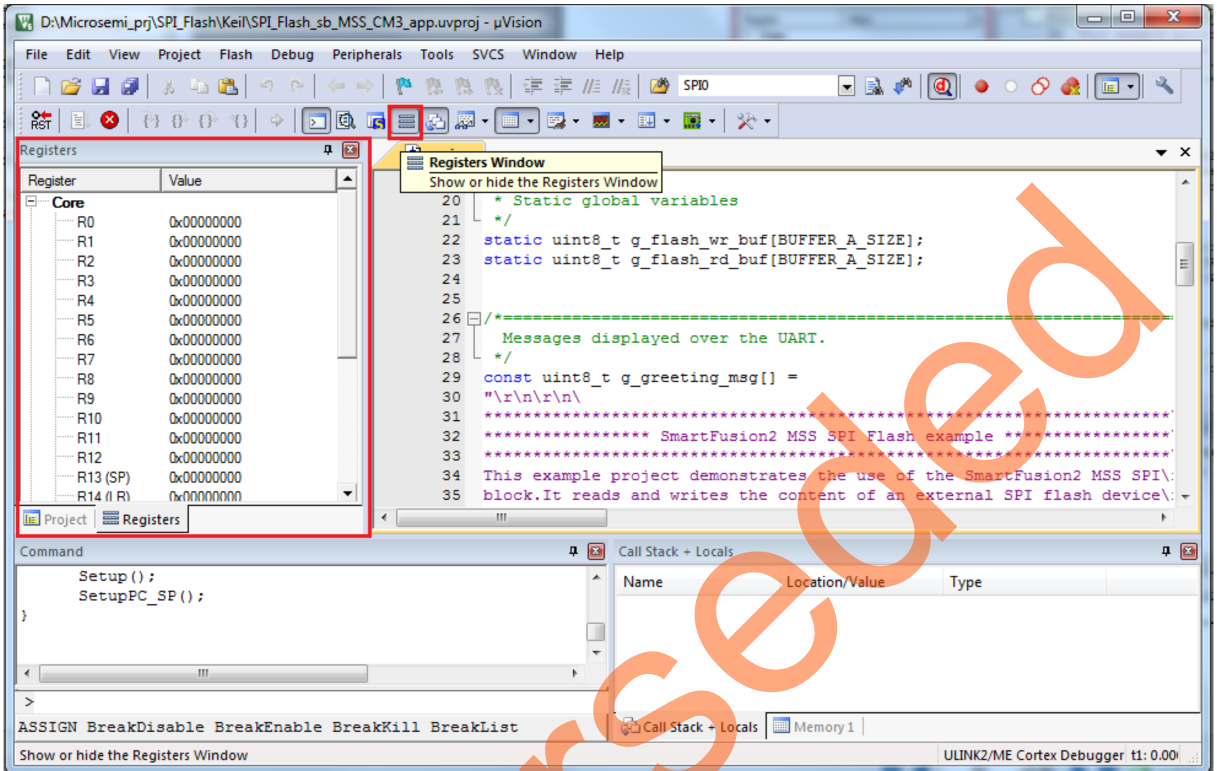

Figure 44 **• Values of the Cortex-M3 Internal Registers**

7. When the debug process is finished, terminate execution of the code by choosing **Debug > Start/Stop Debug Session**, as shown in Figure 45.

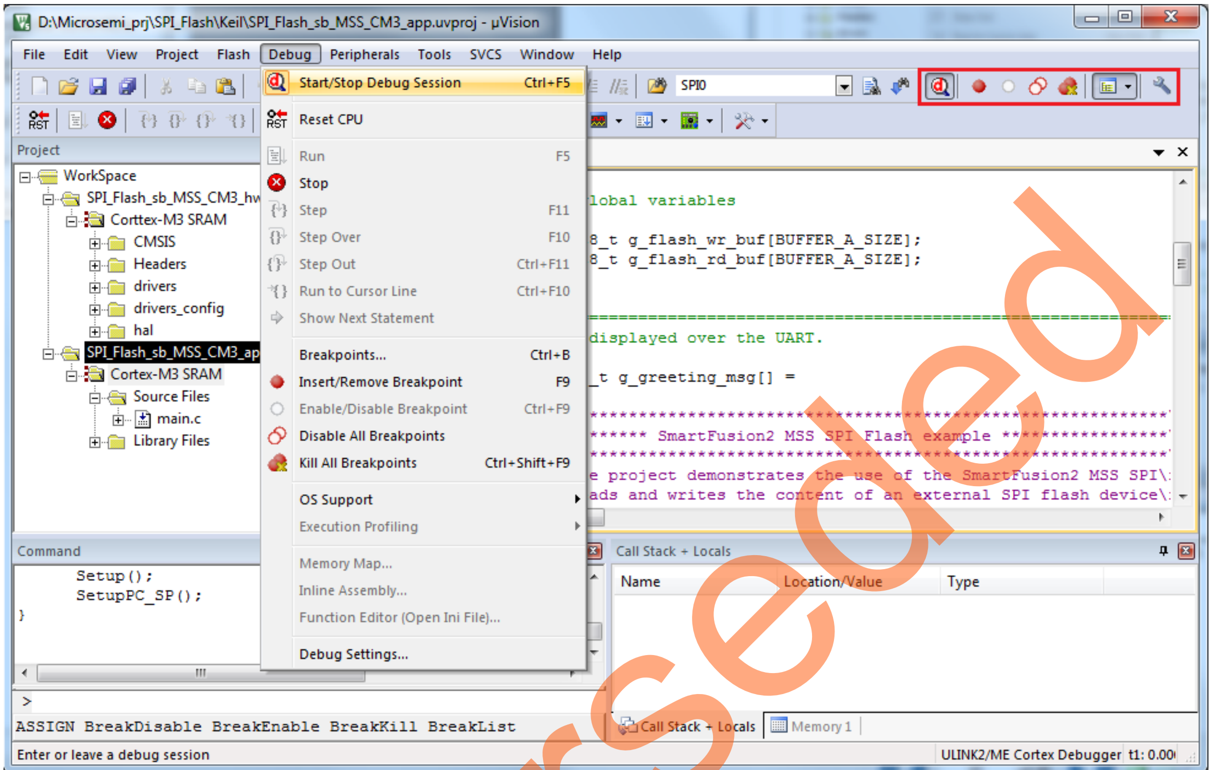

***Figure 45* • Keil uVision Workbench - Stop Debug Option**

8. 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 46:

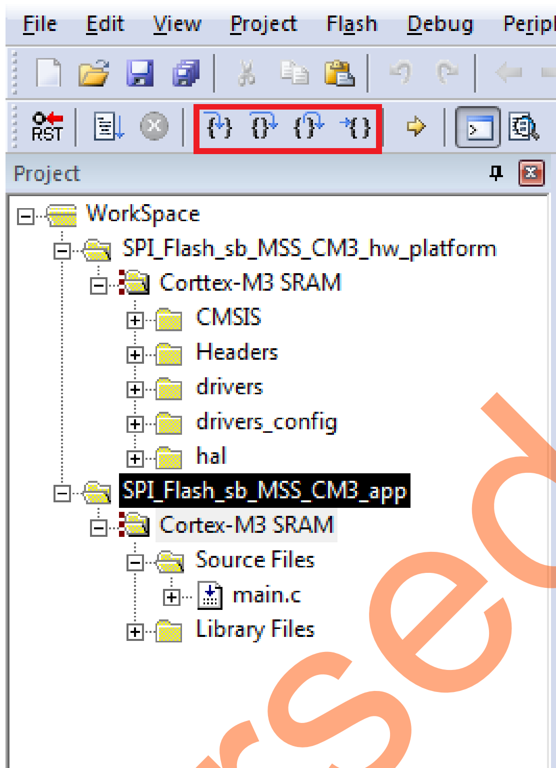

Figure 46 **• Keil uVision Workbench - Step Level Debugging**

– 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 46. 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.

9. 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.

10. Close uVision using **File > Exit**.

11. 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 Security Evaluation Kit board.

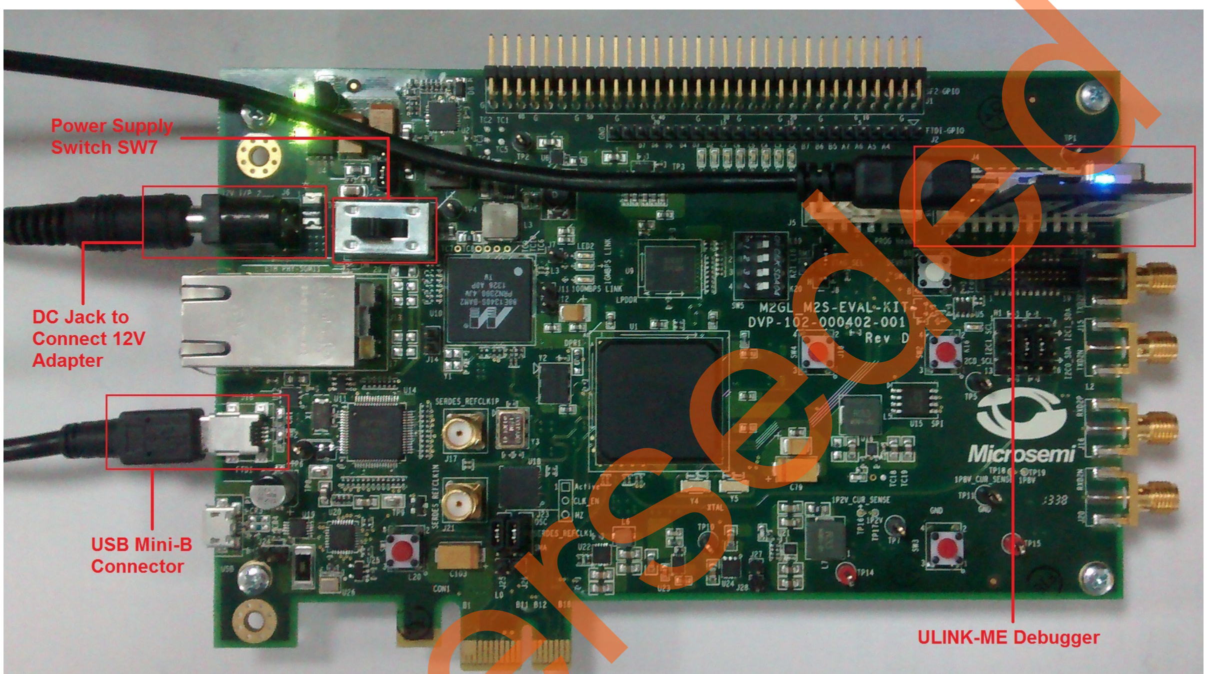

***Figure 1* • SmartFusion2 Security Evaluation Kit in Debug Mode using Keil uVision**

Appendix B: Board Setup for Programming the Tutorial

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

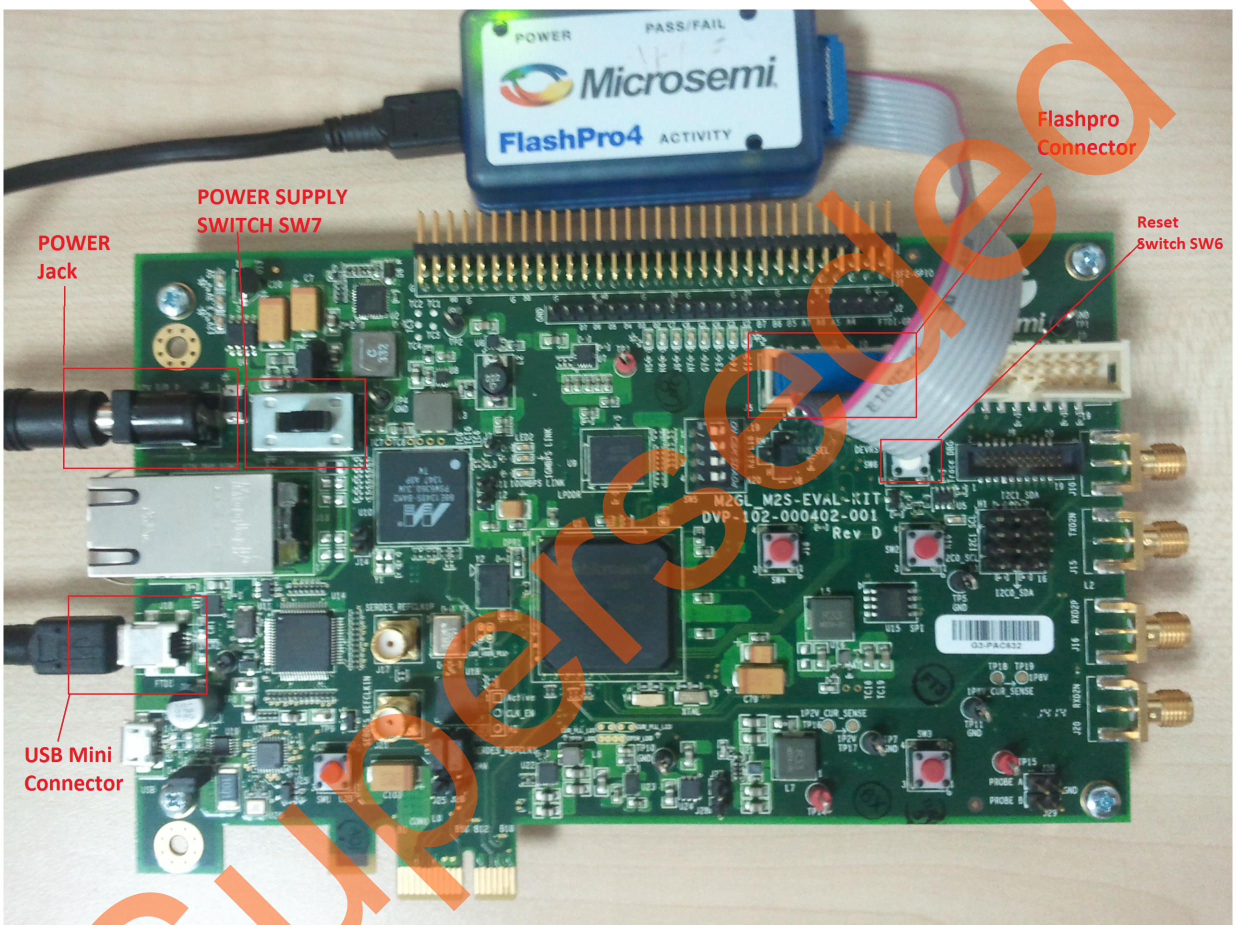

Figure 1 **• SmartFusion2 Security Evaluation Kit in Programming Mode**

Appendix C: SmartFusion2 Security Evaluation Kit Board Jumper Locations

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

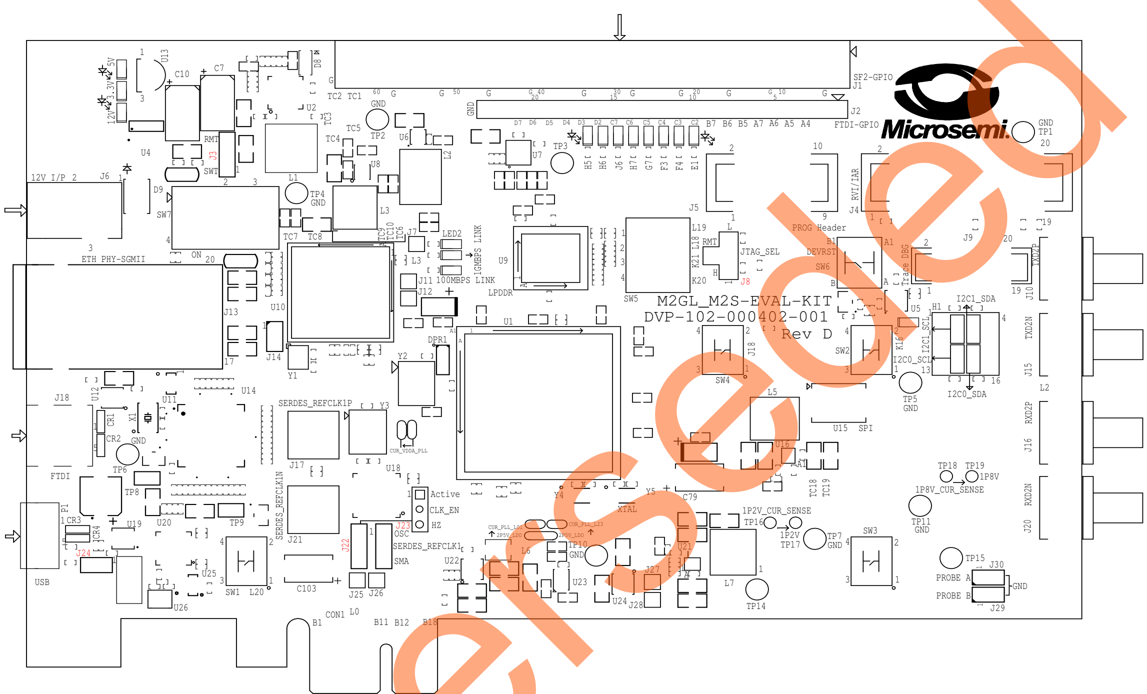

Figure 1 **• SmartFusion2 Security Evaluation Kit Board Jumper Locations**

Notes:

- Jumpers highlighted in red (J22, J23, J24, J3, J8) are set by default.
- The location of the jumpers in Figure 1 are searchable.

List of Changes

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

Revision*	**Changes**	**Page**
Revision 4 (October 2015)	Updated the document for Libero SoC v11.6 software release changes (SAR 72567).	NA
Revision 3 (March 2015)	Updated the document for Libero SoC v11.5 software release (SAR 64189).	N/A
Revision 2 (November 2014)	Updated the document for Libero SoC v11.4 software release (SAR 61938).	N/A
Revision 1 (April 2014)	Initial release.	N/A

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/design-support/fpga-soc-support.

Website

You can browse a variety of technical and non-technical information on the Microsemi SoC Products Group home page, at http://www.microsemi.com/products/fpga-soc/fpga-and-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. Visit About Us for sales office listings and corporate contacts.

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@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

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense & security, aerospace and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; security technologies and scalable anti-tamper products; Ethernet Solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif., and has approximately 3,600 employees globally. Learn more at **www.microsemi.com**.

50200548-4/10.15