Accessing Serial Flash Memory Using SPI Interface - Libero SoC v11.6 and IAR Embedded Workbench Flow Tutorial for SmartFusion2

TU0547 Tutorial

Microsemi®

Table of Contents

Accessing Serial Flash Memory using SPI Interface - Libero SoC v11.6 and IAR Embedded Workbench Flow Tutorial for SmartFusion2

Introduction

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

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

- *TU0546: Accessing Serial Flash Memory using SPI Interface - Libero SoC and SoftConsole Flow Tutorial for SmartFusion2 SoC FPGA*
- *TU0548: 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:

- Creating a Libero SoC project using System Builder
- Generating the programming file to program the SmartFusion2 device
- Opening the project in IAR Embedded Workbench from Libero SoC
- Compiling application code
- Debugging and run code using IAR Embedded Workbench

Design Requirements

Table 1 • Design Requirements

Design Requirements	Description
Hardware Requirements	
SmartFusion2 Security Evaluation Kit • FlashPro4 programmer • J-Link programmer • USB A to Mini-B cable • 12 V Adapter	Rev D or later
Host PC or Laptop	Any 64-bit Windows Operating System
Software Requirements	
Libero SoC	v11.6
FlashPro programming software	v11.6
IAR Embedded Workbench for ARM	v6.4
Host PC Drivers	USB to UART drivers
• HyperTerminal • TeraTerm • PuTTY	–

Project Files

The design files for this tutorial can be downloaded from the Microsemi® website:
http://soc.microsemi.com/download/rsc/?f=m2s_tu0547_liberov11p6_df

The design files include:

- Liberoproject
- Programmingfiles
- Source files
- SPI_Flash_Drivers
- Readme files

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, named 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.

For more information on SPI, refer to the *UG0331: SmartFusion2 Microcontroller Subsystem User Guide*.

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

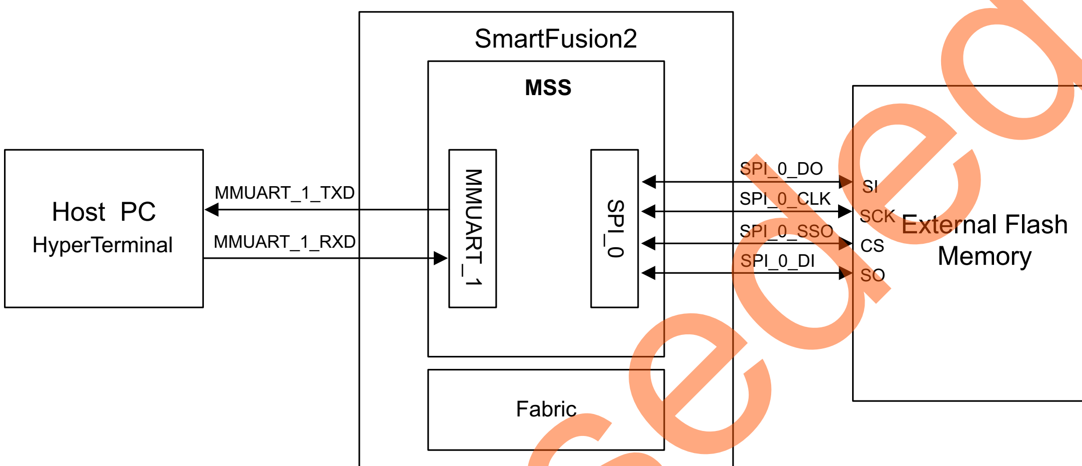

Figure 1 **• SPI Flash Interfacing Block Diagram**

Step 1: Creating a Libero SoC Project

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

Launching Libero SoC

The following steps describe how to launch Libero SoC:

1. Click **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.

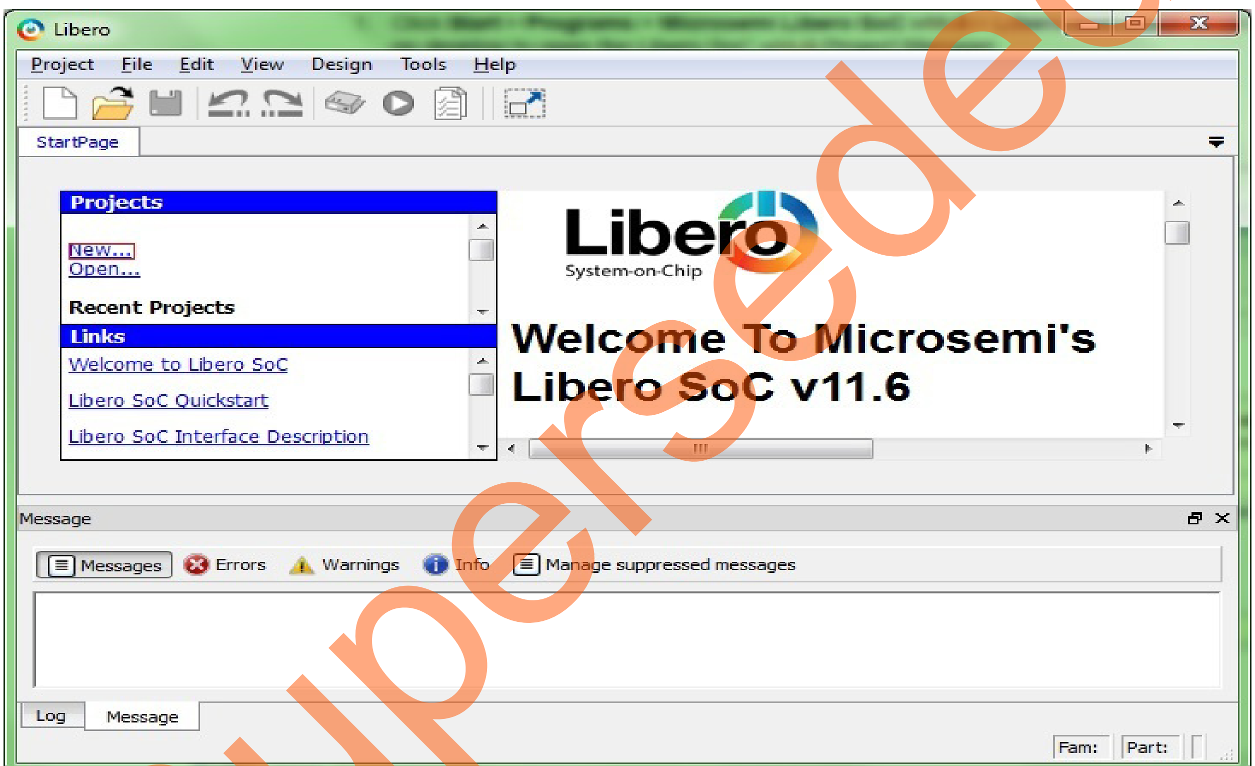

Figure 2 • **Libero SoC Project Manager**

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

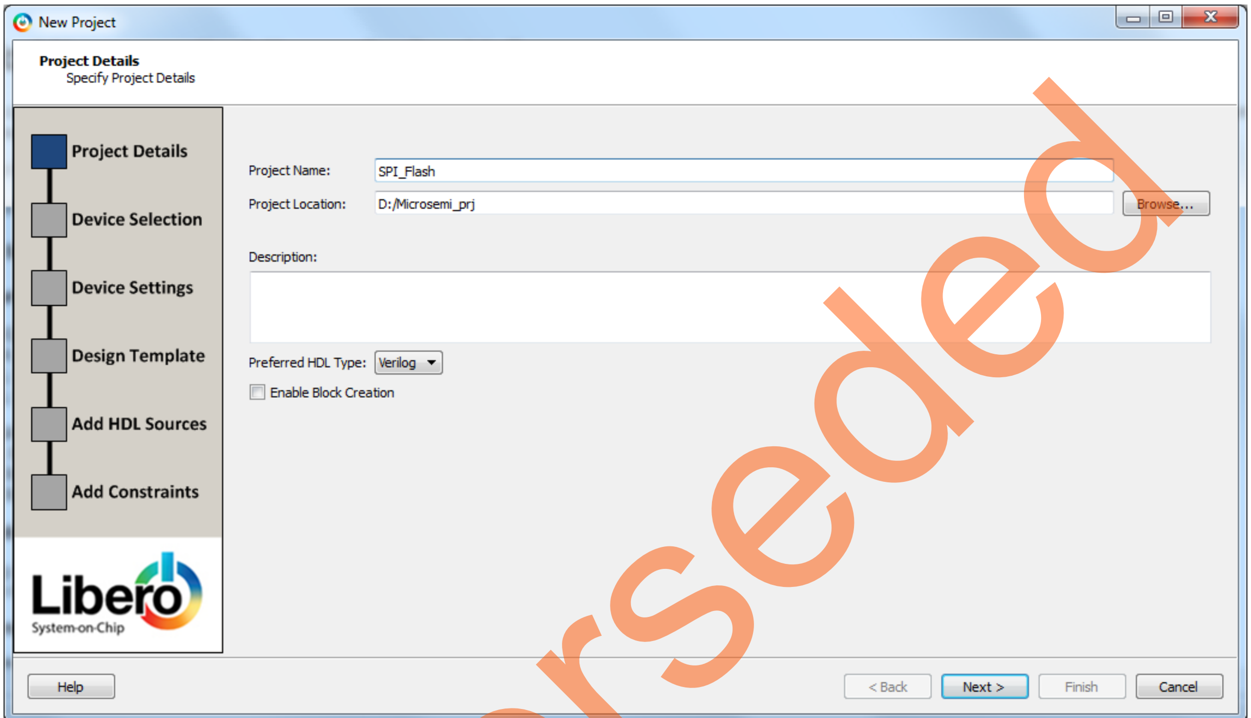

***Figure 3* • Project Details Page**

4. Click **Next**. This opens **Device Selection** page as shown in Figure 4.

   Select the following values from the drop down lists:

   – **Family**: SmartFusion2
   – **Die**: M2S090TS
   – **Package**: 484 FBGA
   – **Speed**: -1
   – **Core voltage**: 1.2
   – **Range**: COM

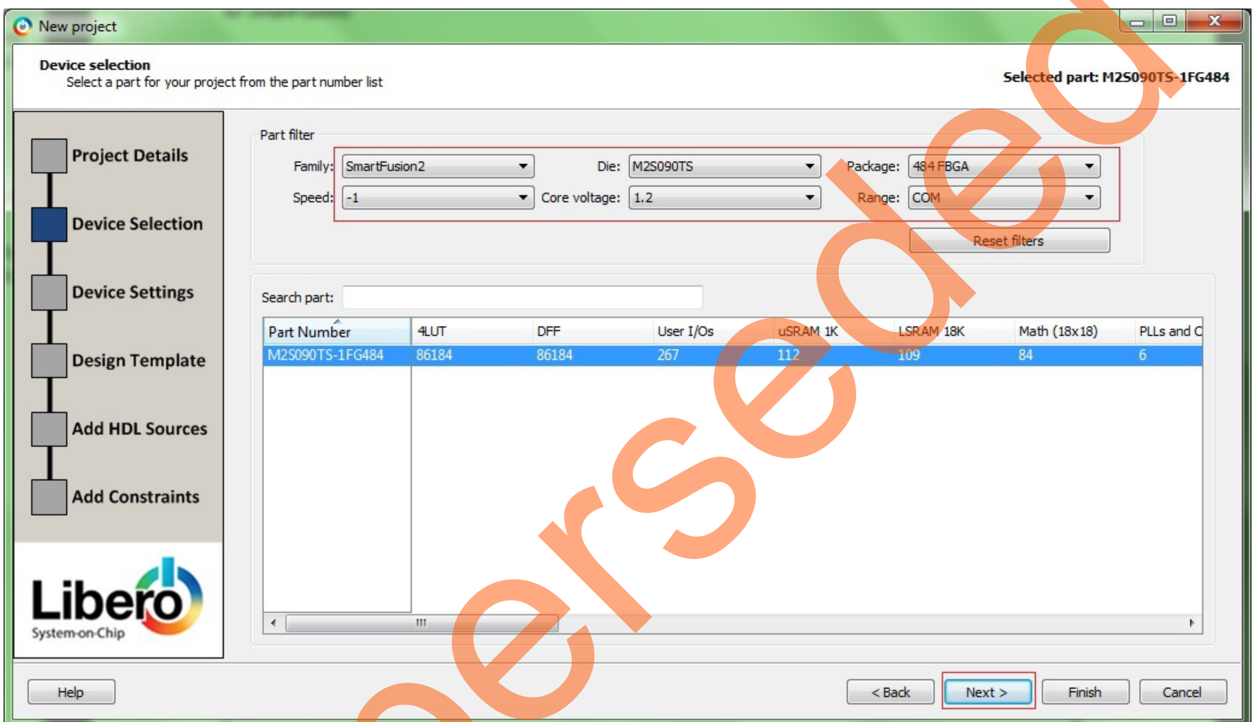

Figure 4 **• Device Selection Page**

5. Click **Next**. This opens **Device Settings** page as shown in Figure 5. Select PLL supply voltage as 3.3.

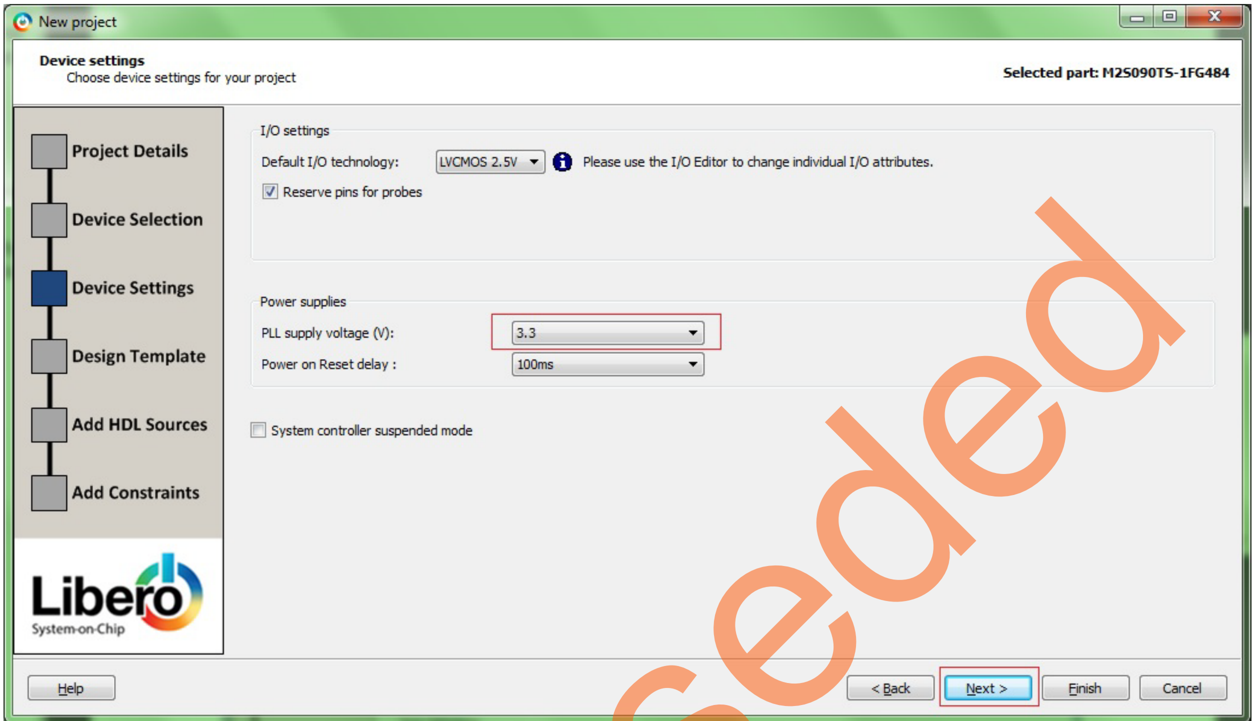

***Figure 5* • Device Settings**

6. Click **Next**. This opens **Design Template** page as shown in Figure 6, Under Design Templates and Creators, select **Create a system builder based design**.

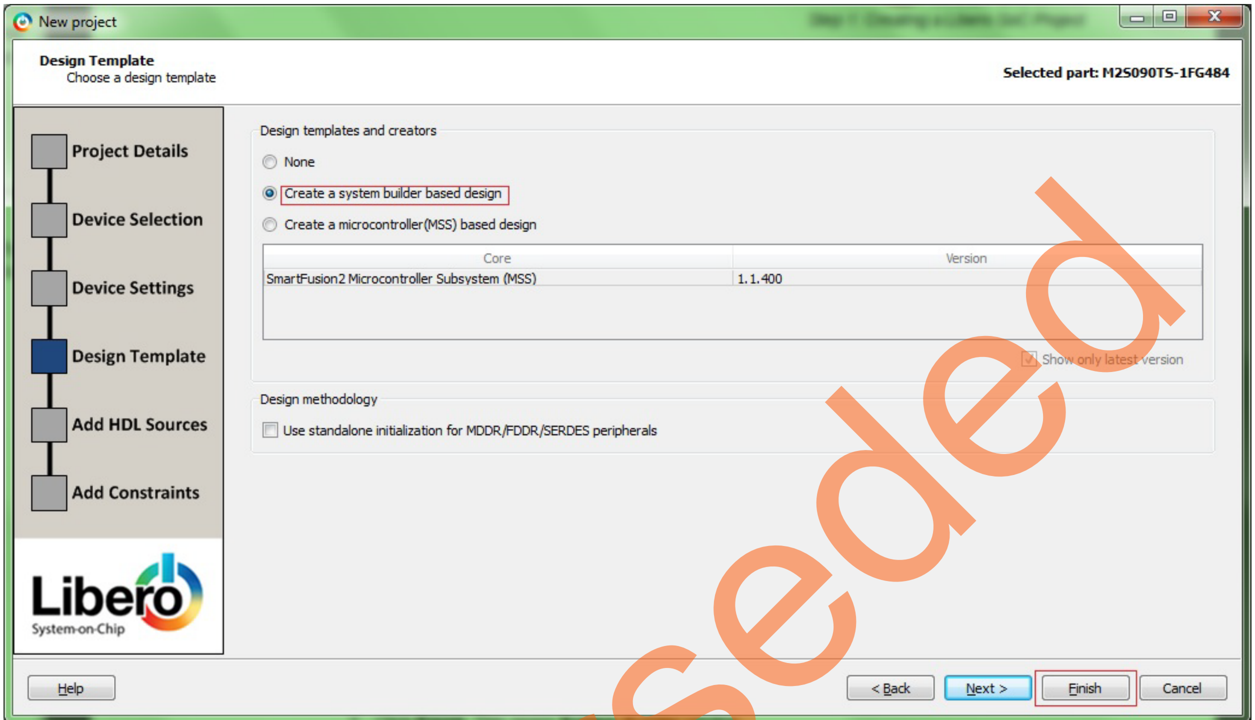

***Figure 6* • Device Template Page**

7. Click **Finish.** This opens **System Builder** window.

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 define the intended system.

8. Enter the name of the system as **SPI_Flash** and click **OK**, as shown in Figure 7.

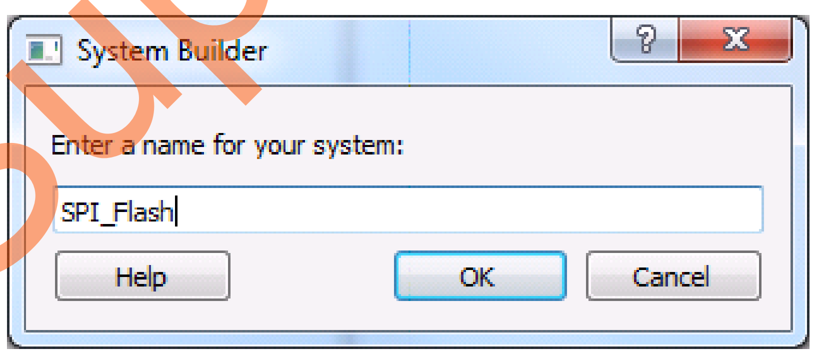

***Figure 7* • System Builder Window**

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

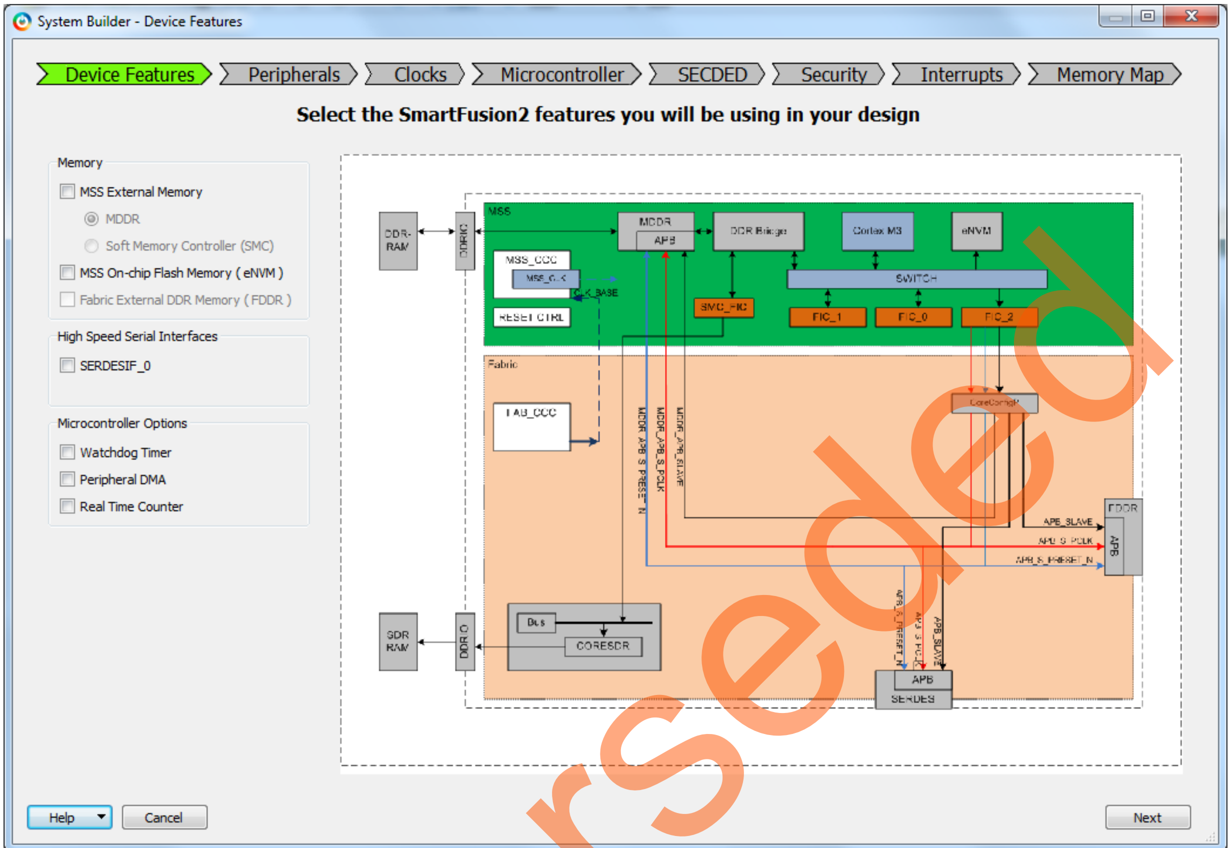

Figure 8 • **System Builder – Device Features Page**

9. Click **Next**. This opens **System Builder - Peripherals** page as shown in Figure 9.
10. Under the MSS Peripherals section, clear all the check boxes except **MM_UART_1** and **MSS_SPI_0**, as shown in Figure 9.

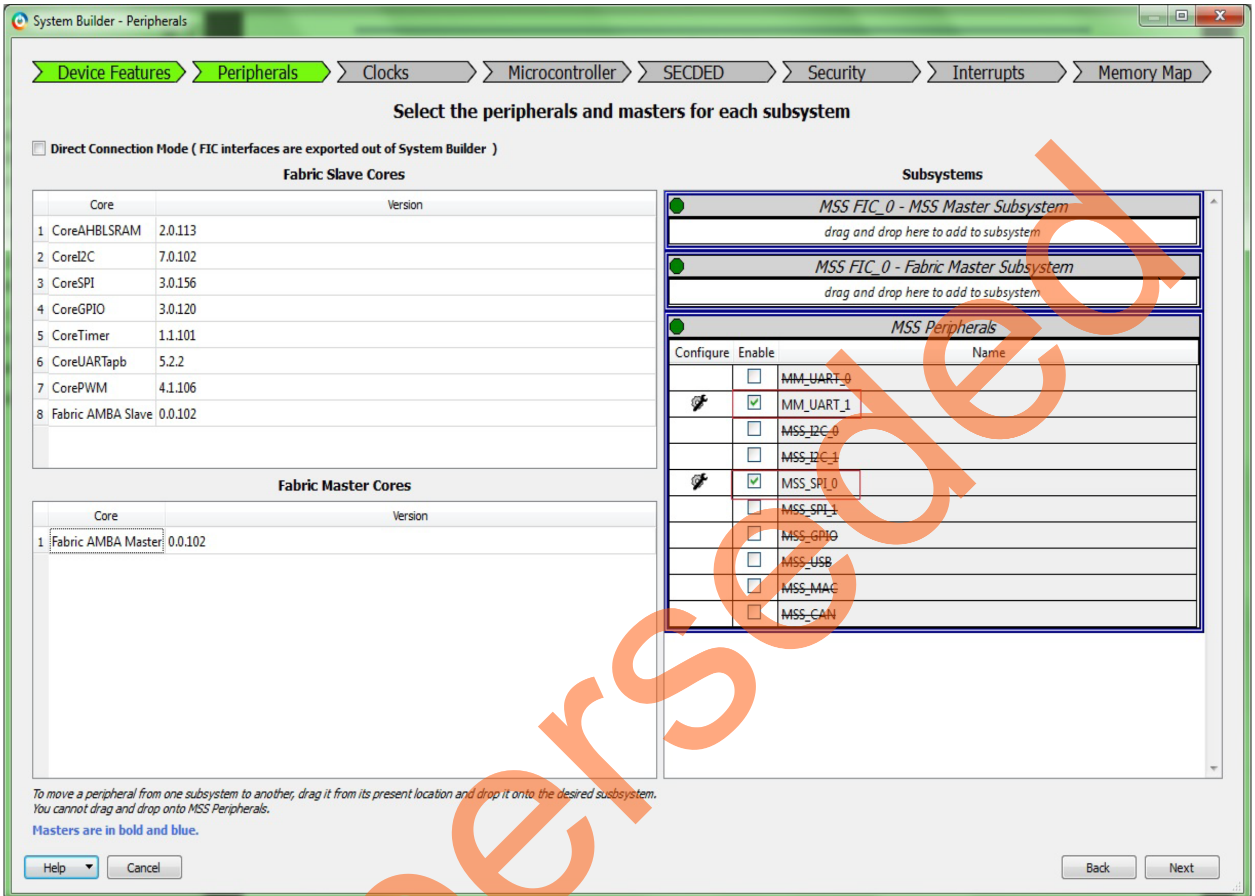

Figure 9 **• System Builder – Peripherals Page**

11. Click **Next**. This opens **System Builder - Clocks** page as shown in Figure 10.
12. In the **System Builder - Clocks** page (Refer to Figure 10):
    - 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**
    - Do not change the default settings of remaining parameters.

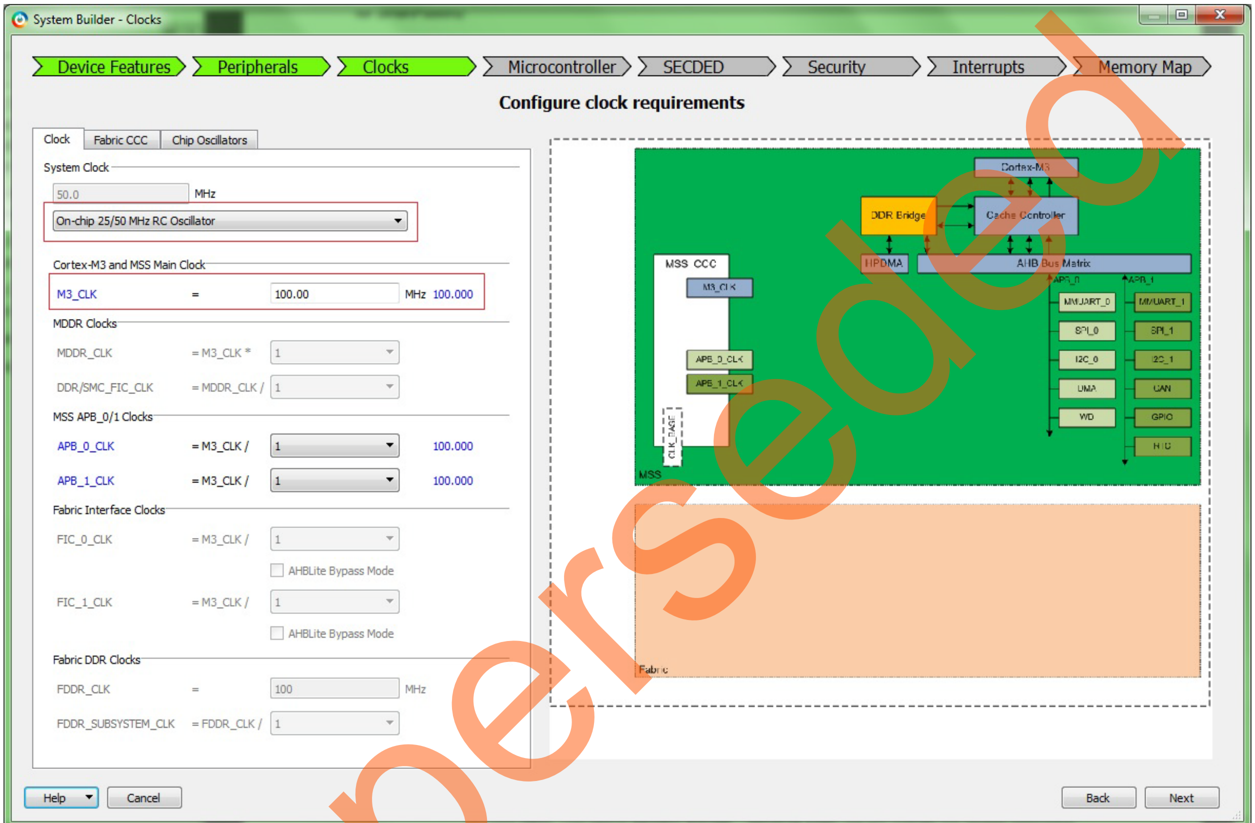

***Figure 10* • System Builder – Clock Page**

13. Click **Next**. This opens **System Builder - Microcontroller** page. Do not change the default selections.
14. Click **Next**. This opens **System Builder - SECDED** page. Do not change the default selections.
15. Click **Next**. This opens **System Builder - Security** page. Do not change the default selections.
16. Click **Next**. This opens **System Builder - Interrupts** page. Do not change the default selections.
17. Click **Next**. This opens **System Builder - Memory Map** page. Do not change the default selections.
18. Click **Finish**.

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

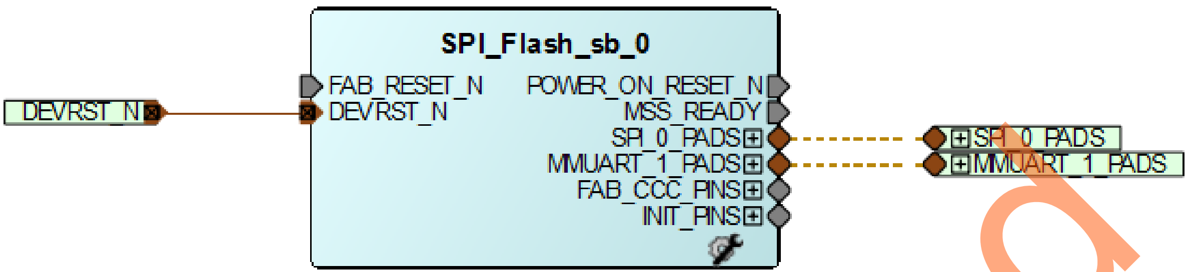

Figure 11 **• SPI_Flash Smart Design**

Connecting Components in SPI_Flash 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.

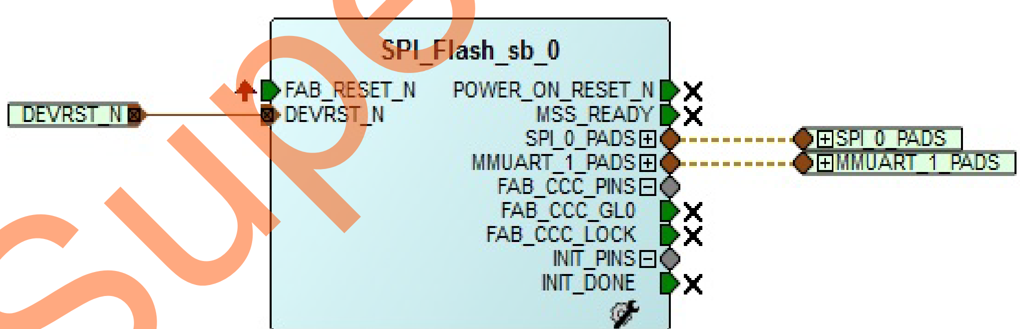

Figure 12 **• SPI_Flash Smart Design**

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

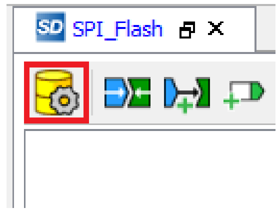

***Figure 13* • Generate Component**

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.
```

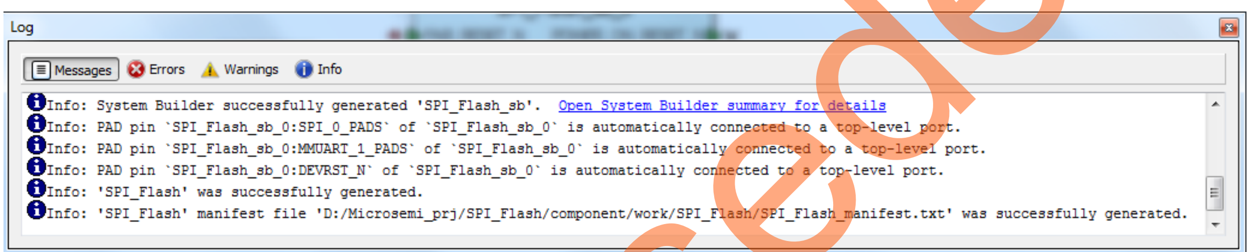

***Figure 14* • Log Window**

Step 2: Generating the Program File

The following step describes how to generate the program file:

Click **Generate Bitstream** as shown in Figure 15 to complete place and route, and generate the programming file.

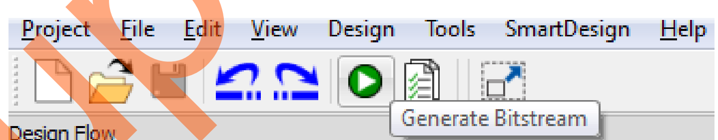

***Figure 15* • Generate Bitstream**

Step 3: Programming SmartFusion2 Security Evaluation Board Using FlashPro

The following steps describe how to program the SmartFusion2 Security Evaluation 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 per Table 2.
   For more information on jumper locations, refer to Appendix C: SmartFusion2 Security Evaluation Kit Board Jumper Locations.

   CAUTION: Ensure that the power supply switch, **SW7** is switched OFF while connecting the jumpers on the SmartFusion 2 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.
4. Switch ON the power supply switch, **SW7**. Refer to Appendix A: Board Setup for Programming the Tutorial for information on the board setup for running the tutorial.
5. To program the SmartFusion2 device, double-click **Run PROGRAM Action** in the **Design Flow** tab as shown in Figure 16.

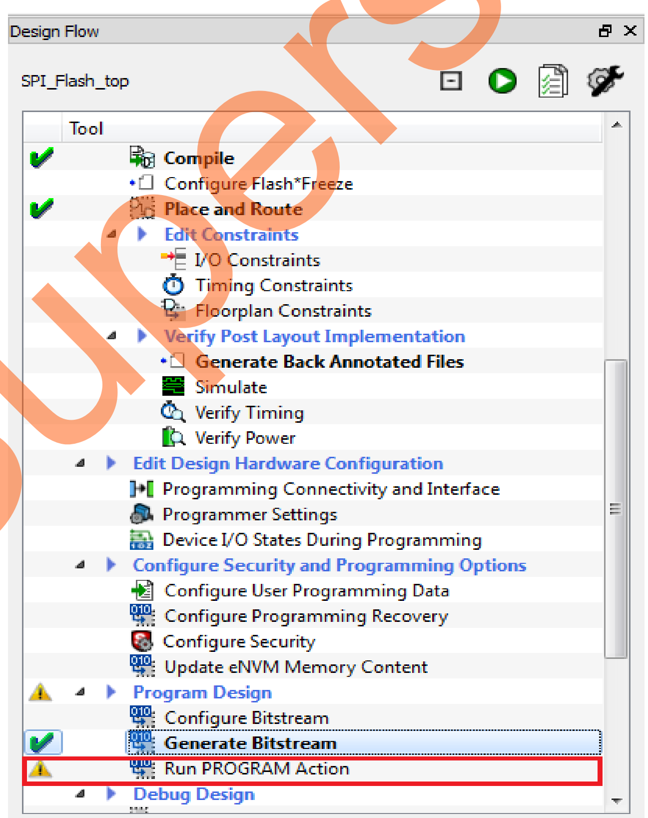

Figure 16 • **Run Program Action**

Step 4: Configuring and Generating Firmware

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

- CMSIS
- MMUART
- SPI

To generate the required drivers:

1. Double-click **Configure Firmware Cores** in **Handoff design for Firmware Development** in **Design Flow** window.
2. Clear all the drivers check boxes, except **CMSIS**, **MMUART**, and **SPI**, as shown in Figure 17.

Note: Select the latest version of the drivers.

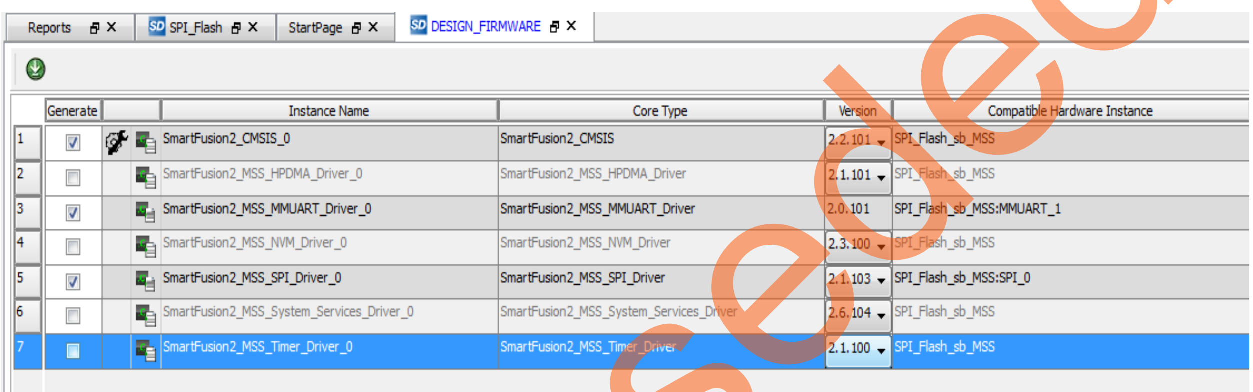

Figure 17 • **Configuring Firmware**

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

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

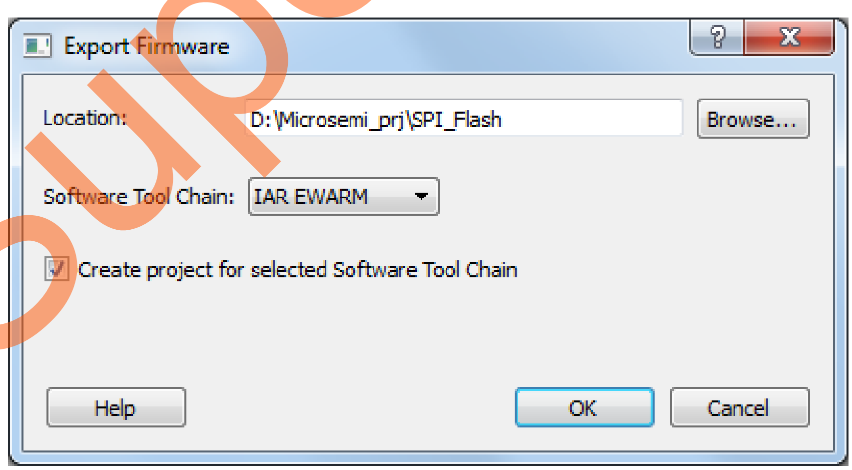

Figure 18 • **Export Firmware Dialog Box**

4. In the **Export Firmware** dialog box:
   - Select **Create project for selected Software Tool Chain check box**.
   - Select **IAR EWARM** from the drop down list.

5. Click **OK**. The successful firmware generation window is displayed as shown in Figure 19.

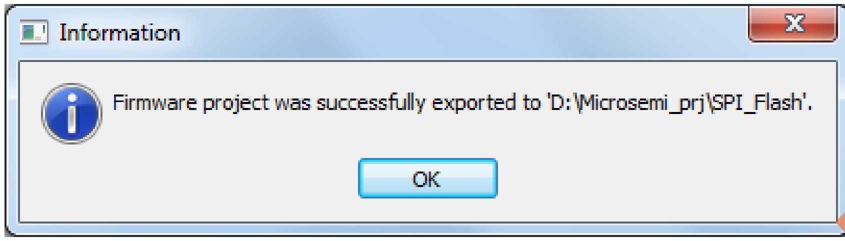

Figure 19 **• Firmware Successfully Exported Message**

6. Click **OK**.

The SmartFusion2 Security Evaluation Kit is ready for running and debugging the IAR Embedded Workbench application through J-Link Debugger.

Step 5: Building Software Application using IAR Embedded Workbench

The following steps describe how to build a software application using IAR embedded workbench:

1. Connect the J-Link programmer to **J4 connector** of SmartFusion2 Security Evaluation Kit.

   Refer to "Appendix B: Board Setup for Running the IAR Tutorial" on page 44 for information on the board setup for running and debugging the IAR software application.

   Ensure that the SmartFusion2 Security Evaluation Kit Jumper **J8** is in **2-3 closed** position for IAR Embedded Workbench and J-Link communication.

2. Open the IAR project by double-clicking **SPI_Flash_sb_MSS_CM3 IAR** project as shown in Figure 20.

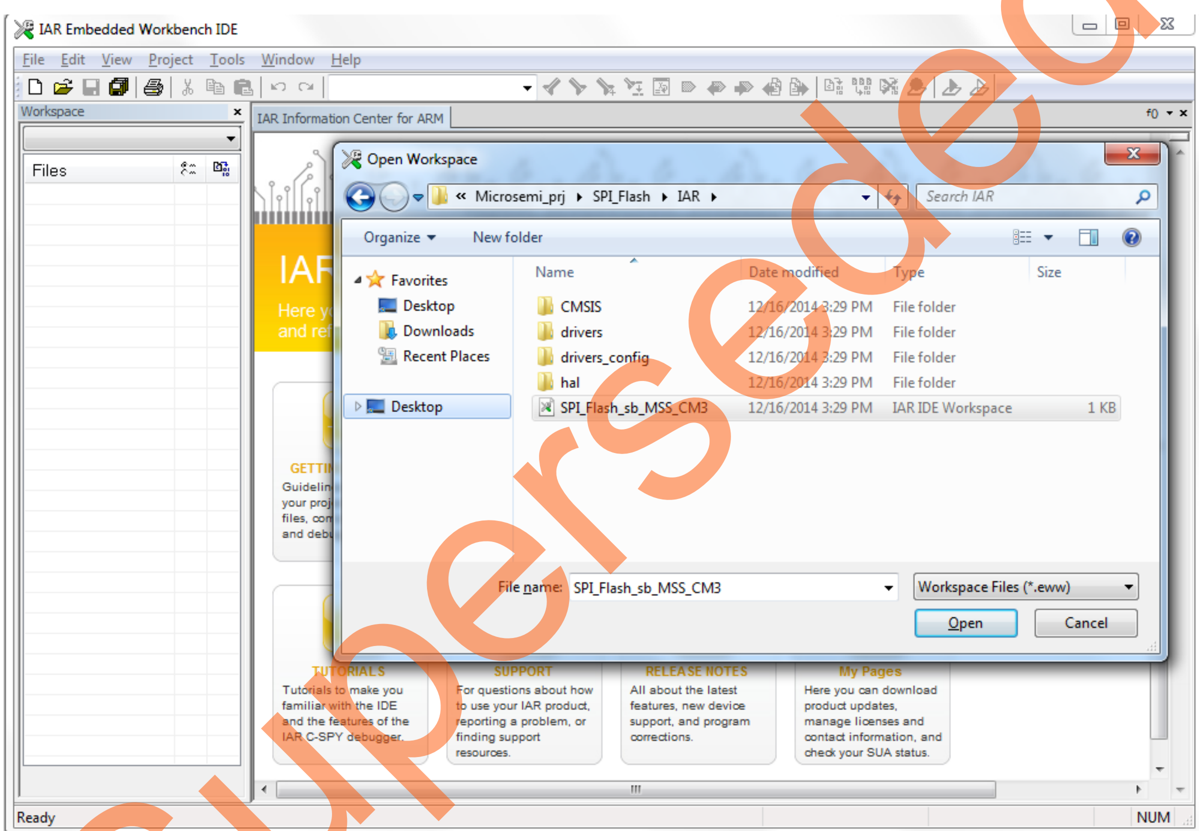

Figure 20 • Invoking IAR Embedded Workbench from Libero SoC Software

The IAR workspace is displayed, as shown in Figure 21.

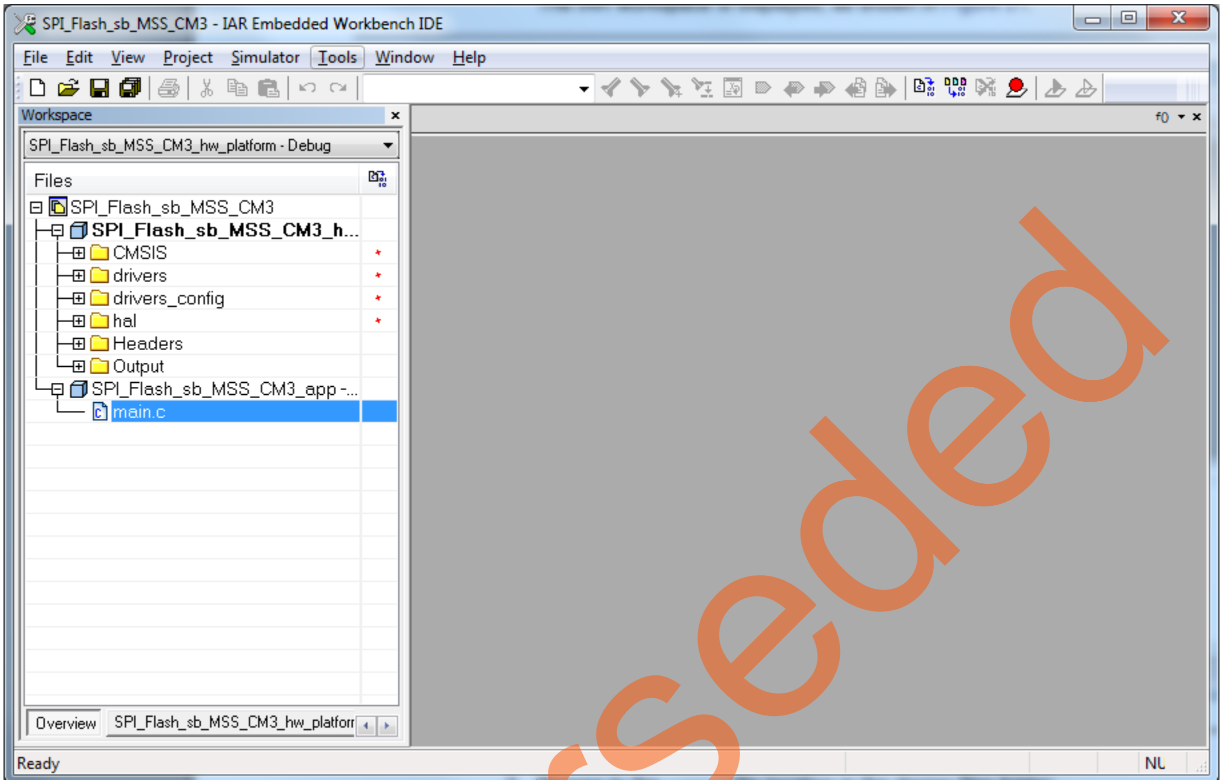

Figure 21 **• IAR Workspace**

3. Browse to the `main.c` file location in the design files folder: *<download folder>\SF2_SPI_Flash_IAR_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 IAR workspace.

The IAR window displays the `main.c` file, as shown in Figure 22.

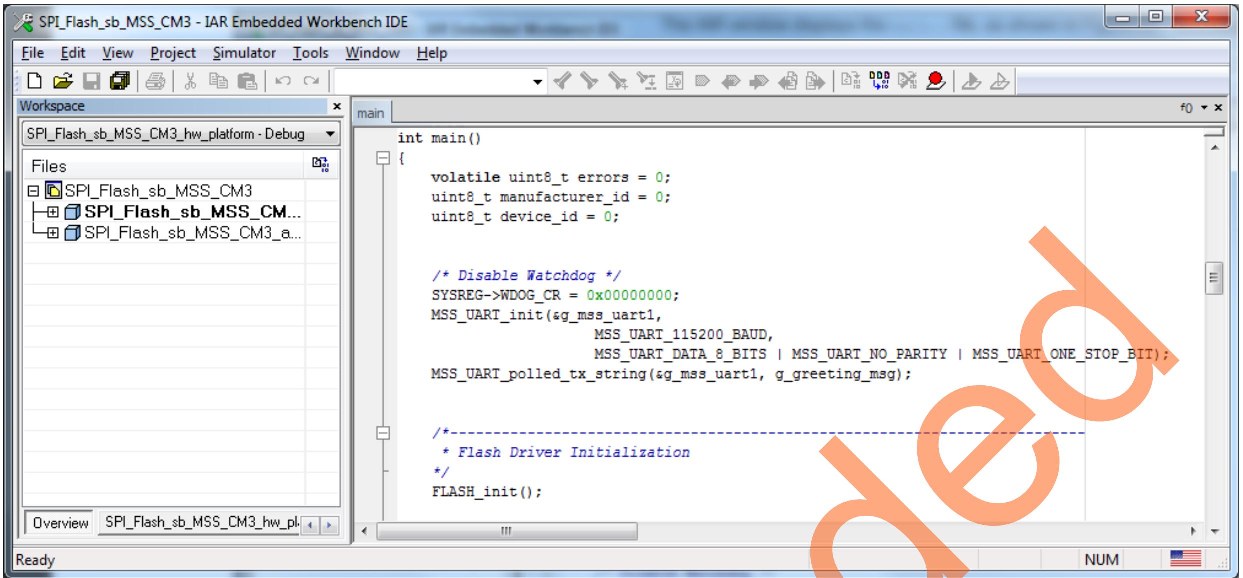

***Figure 22* • IAR Workspace main.c file**

5. The winbondflash SPI flash drivers are not included in the Libero generated IAR workspace. To include the drivers in the IAR workspace, browse to the location of the winbondflash drivers in the design files folder: *<download_folder>\SF2_SPI_Flash_IAR_Tutorial_DF\SPI_Flash_Drivers*.
6. Copy the **winbondflash** folder to the drivers folder of SPI_Flash_sb_MSS_CM3_hw_platform project in the IAR workspace: *projectdirectory\IAR\drivers*.

7. Right-click and add the driver files (winbondflash.c & winbondflash.h) to the SPI_Flash_sb_MSS_CM3_hw_platform project in the IAR workspace, as shown in Figure 23.

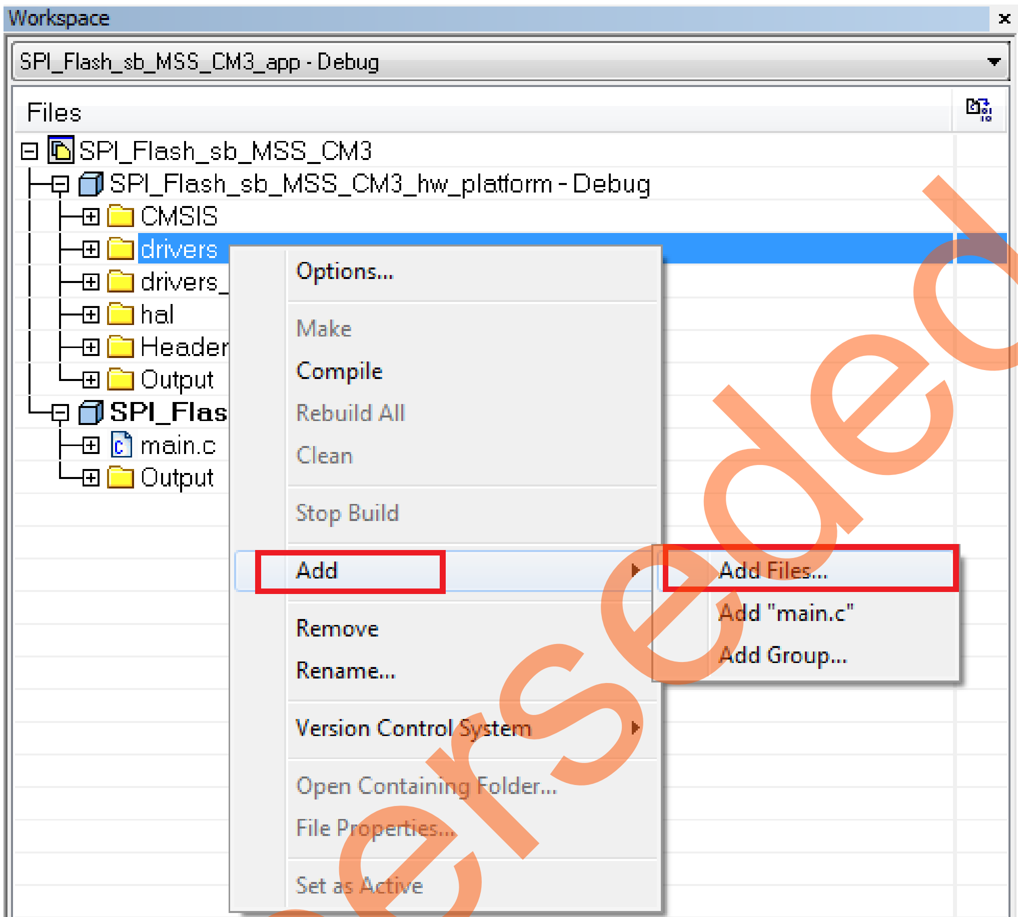

Figure 23 **• IAR Workspace Window - Add winbondflash SPI Driver Files**

Figure 24 shows the IAR workspace window displaying winbondflash SPI Driver Files.

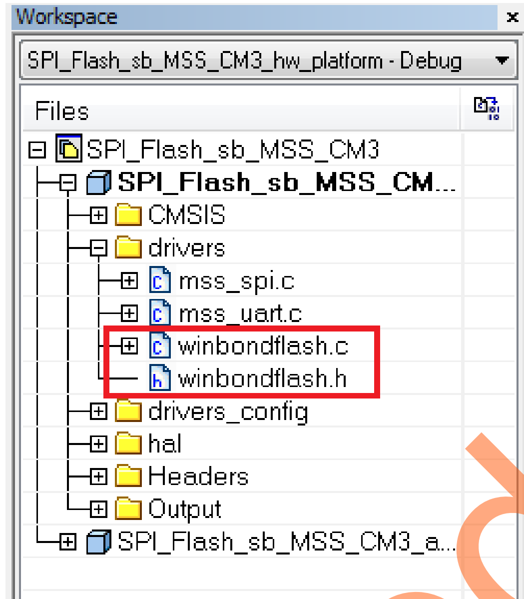

Figure 24 **• IAR Workspace Window - Display winbondflash SPI Driver Files**

8. To configure the project, right-click the project name (SPI_Flash_sb_MSS_CM3_hw_platform) and click **Options**, as shown in Figure 25.

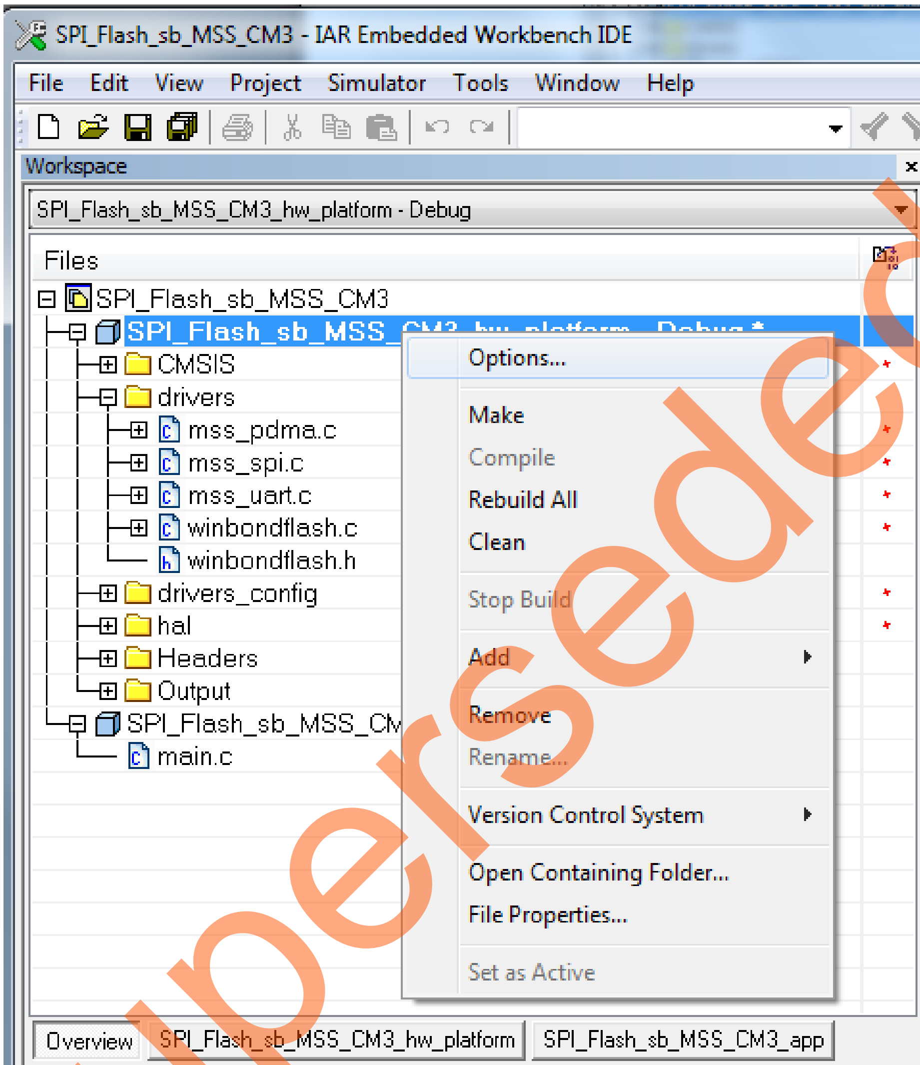

***Figure 25 •* IAR Workspace Window - Choose Options**

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.

9. In **Options** window, click **C/C ++ Compiler**.
10. Click **Preprocessor** tab.

11. Under **Defined symbols,** enter **MICROSEMI_STDIO_THRU_UART** and click **OK**, as shown in Figure 26.

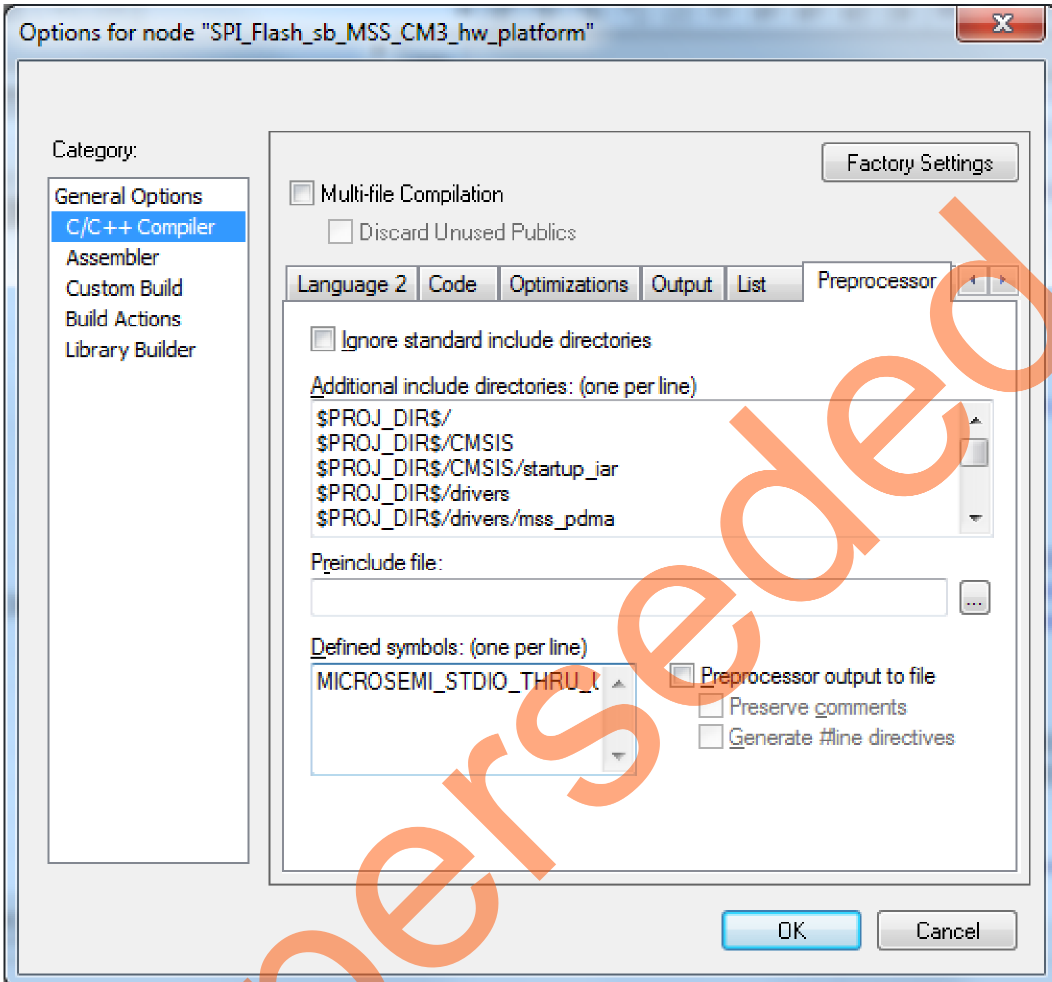

Figure 26 **• IAR Workspace Window - Adding Symbol**

12. To configure the project, right-click the project name (SPI_Flash_sb_MSS_CM3_app), and click **Options**, as shown in Figure 27.

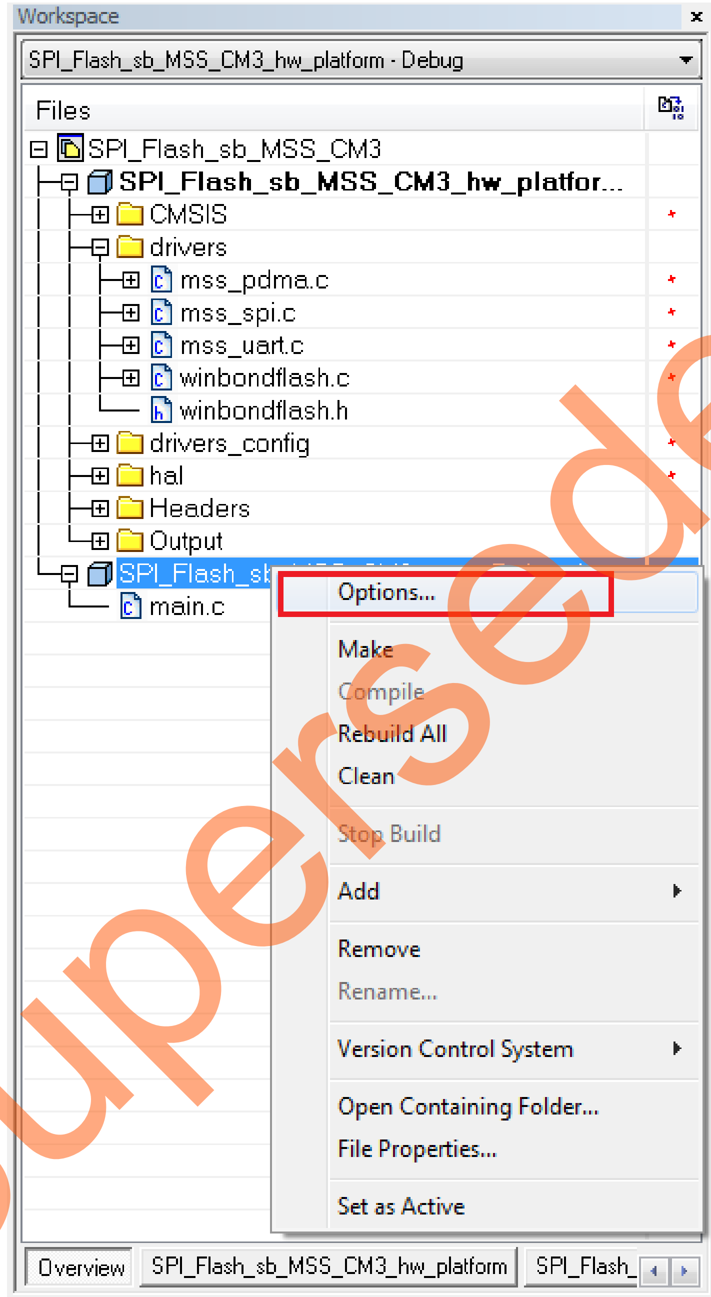

Figure 27 • **IAR Workspace Window - Choose Options**

The **Options for node SPI_Flash_sb_MSS_CM3_app** window is displayed, as shown in Figure 28.

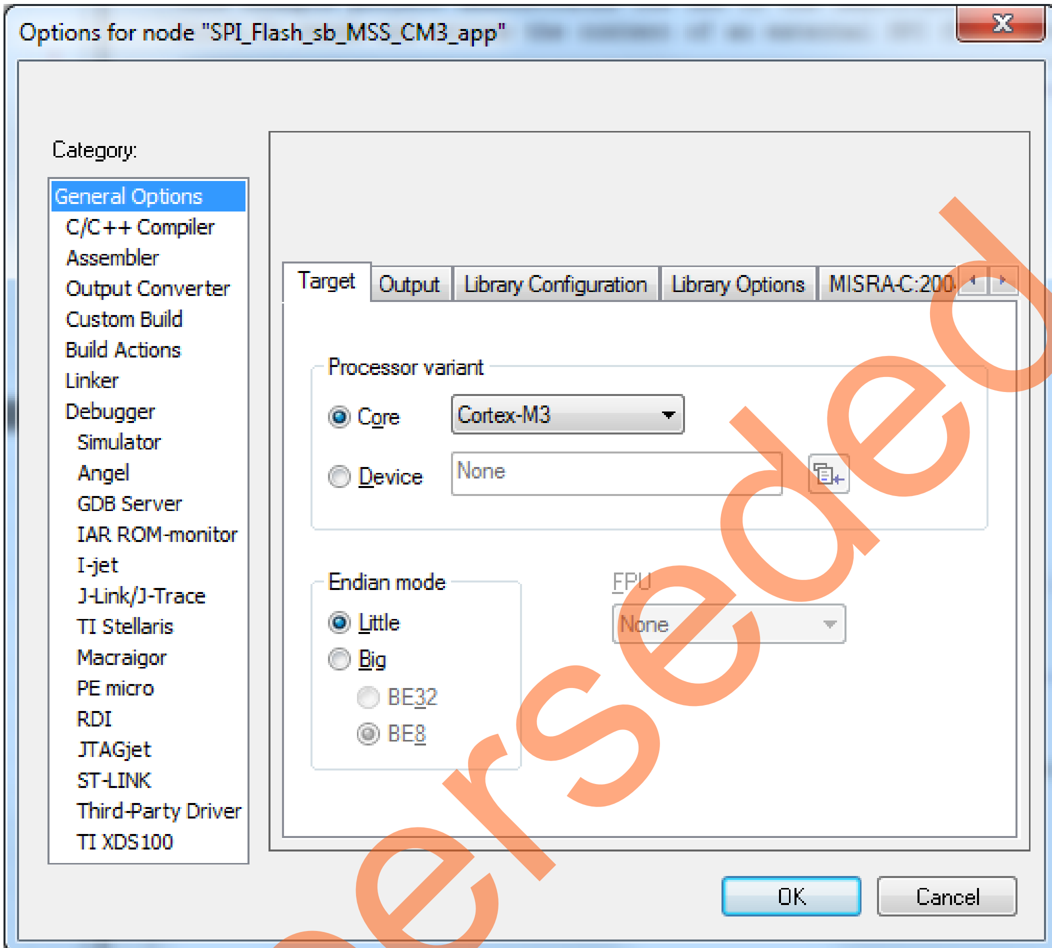

Figure 28 **• IAR Node Options**

Click **Debugger**. Under the **Setup** tab, select **J-Link/J-Trace** from the driver the drop-down list, refer to Figure 29.

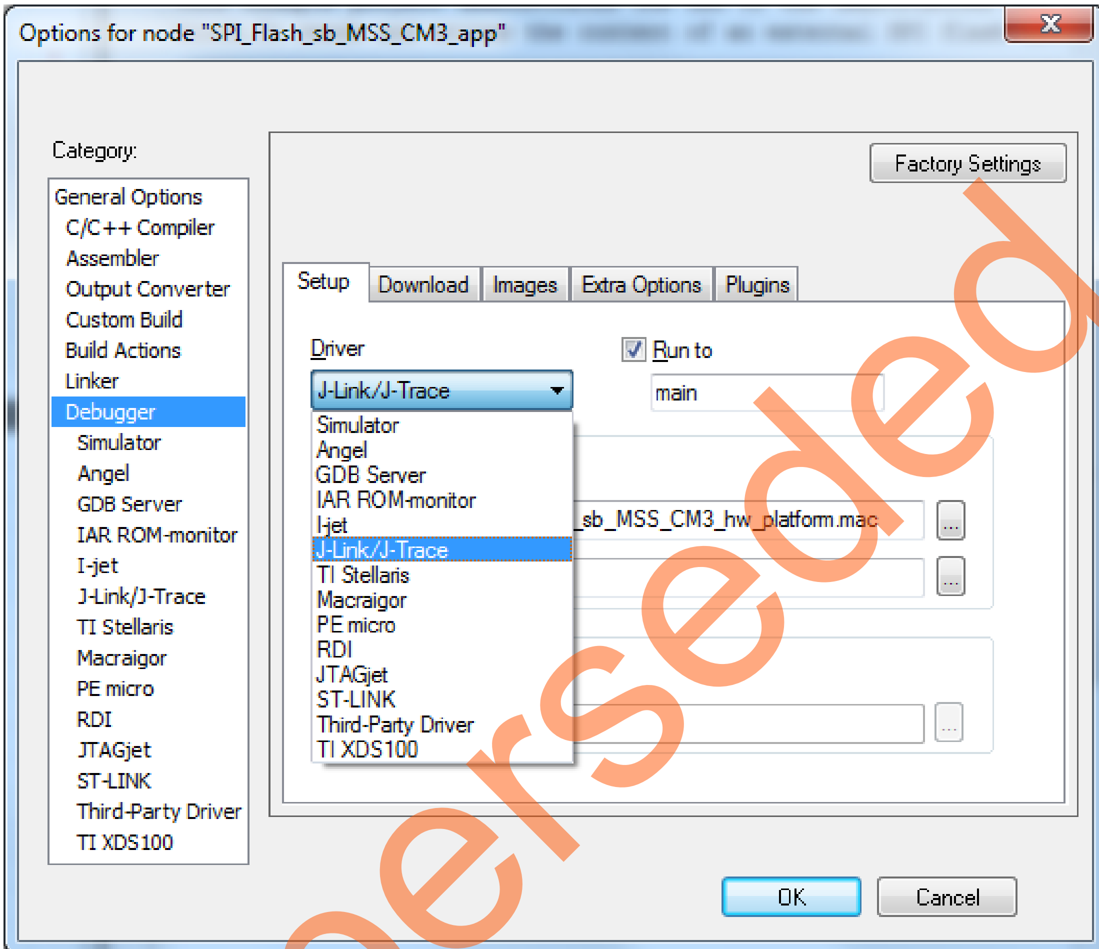

Figure 29 **• IAR Debugger Options - Selecting Driver**

13. Click **Download** tab and select the **Verify download** check box, as shown in Figure 30.

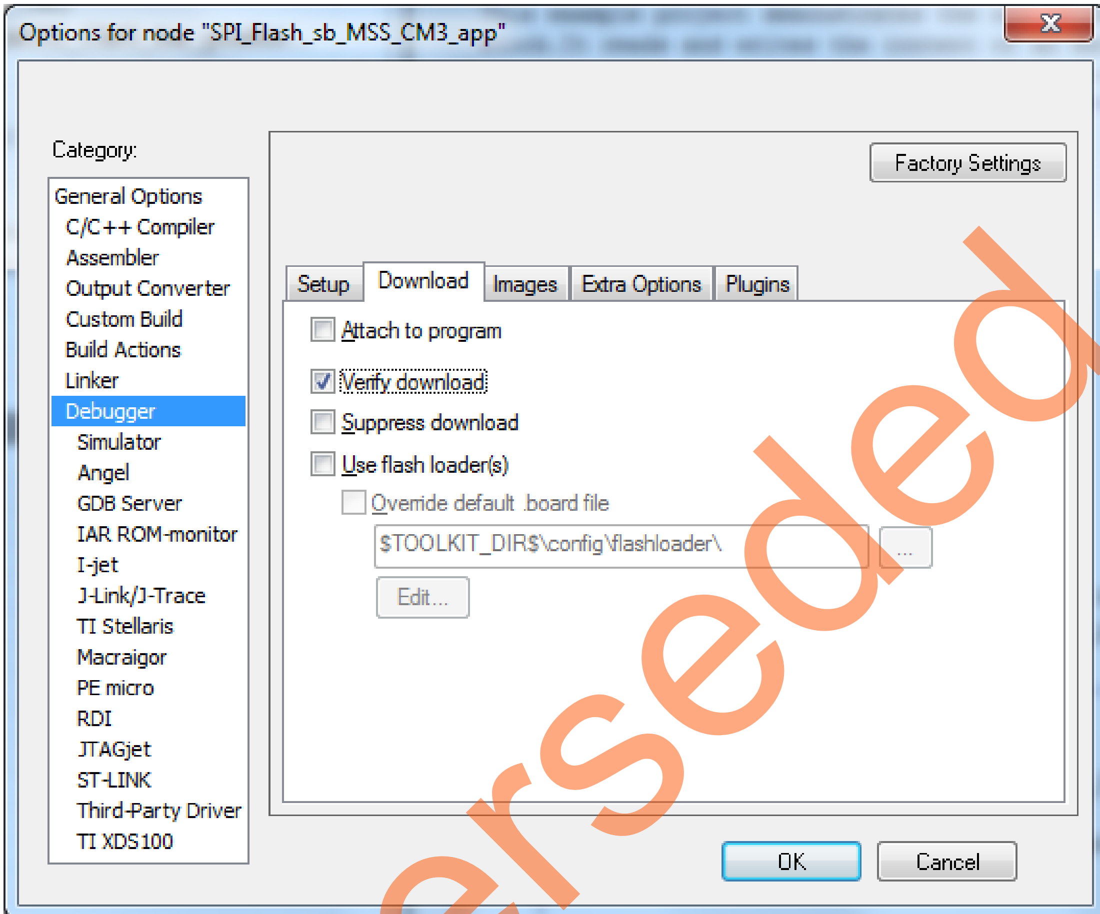

***Figure 30* • IAR Debugger Options - Download**

14. Click **OK** to close the **Options** window and build the project.

15. Right-click **SPI_Flash_sb_MSS_CM3_hw_platform - Debug** and select **Make**, as shown in Figure 31 and Figure 32.

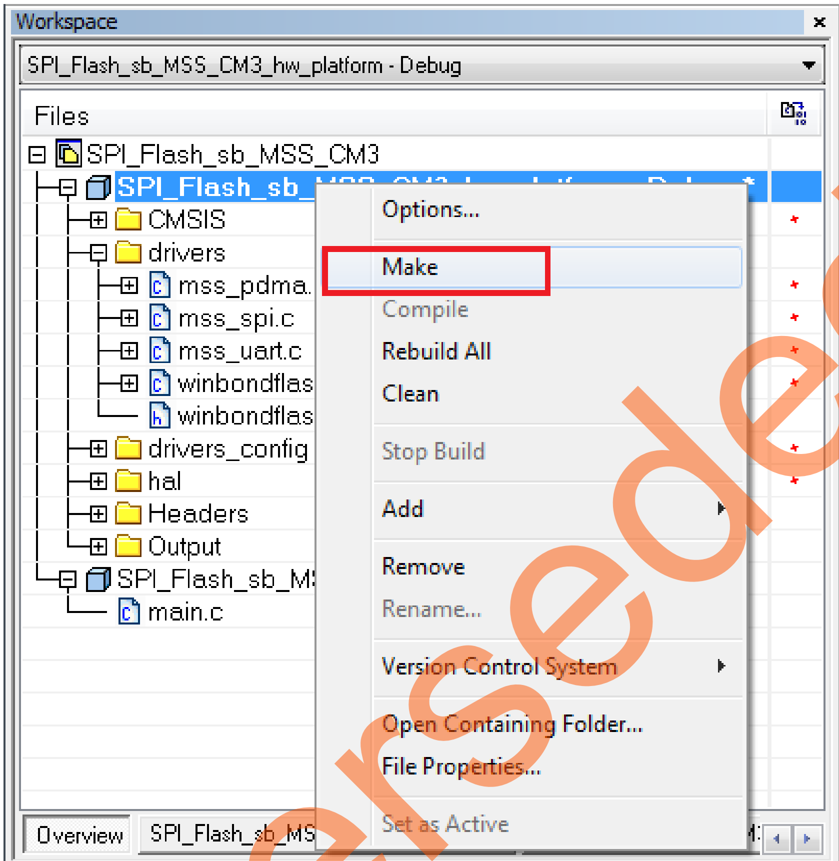

Figure 31 **• IAR Workspace - Hardware Platform Code Compilation using Make**

Successful Hardware Platform Code Compilation page is displayed as shown in Figure 32.

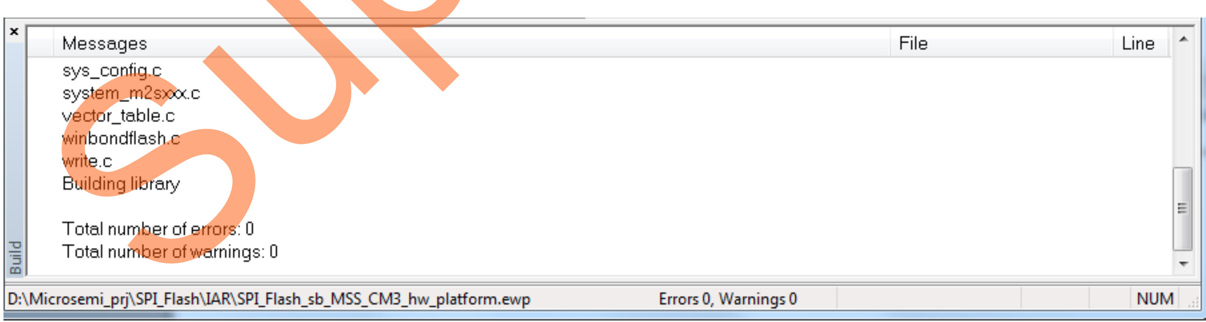

Figure 32 **• IAR Workspace - Successful Hardware Platform Code Compilation using Make**

16. Right-click **SPI_Flash_sb_MSS_CM3_app - Debug** project name and select **Set as Active**, as shown in Figure 33.

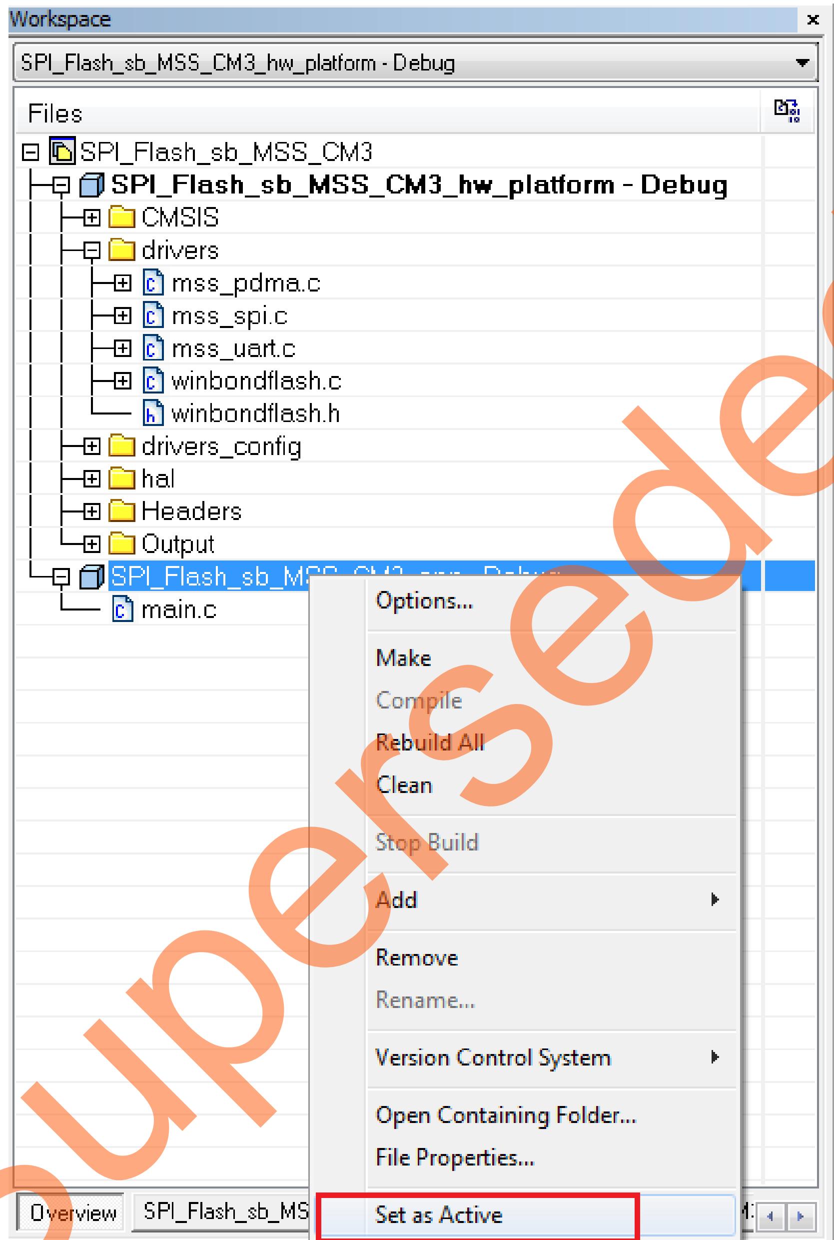

***Figure 33* • IAR Workspace - SPI_Flash_sb_MSS_CM3_app Set as Active**

17. Right-click **SPI_Flash_sb_MSS_CM3_app - Debug** project name and select **Clean**, as shown in Figure 34.

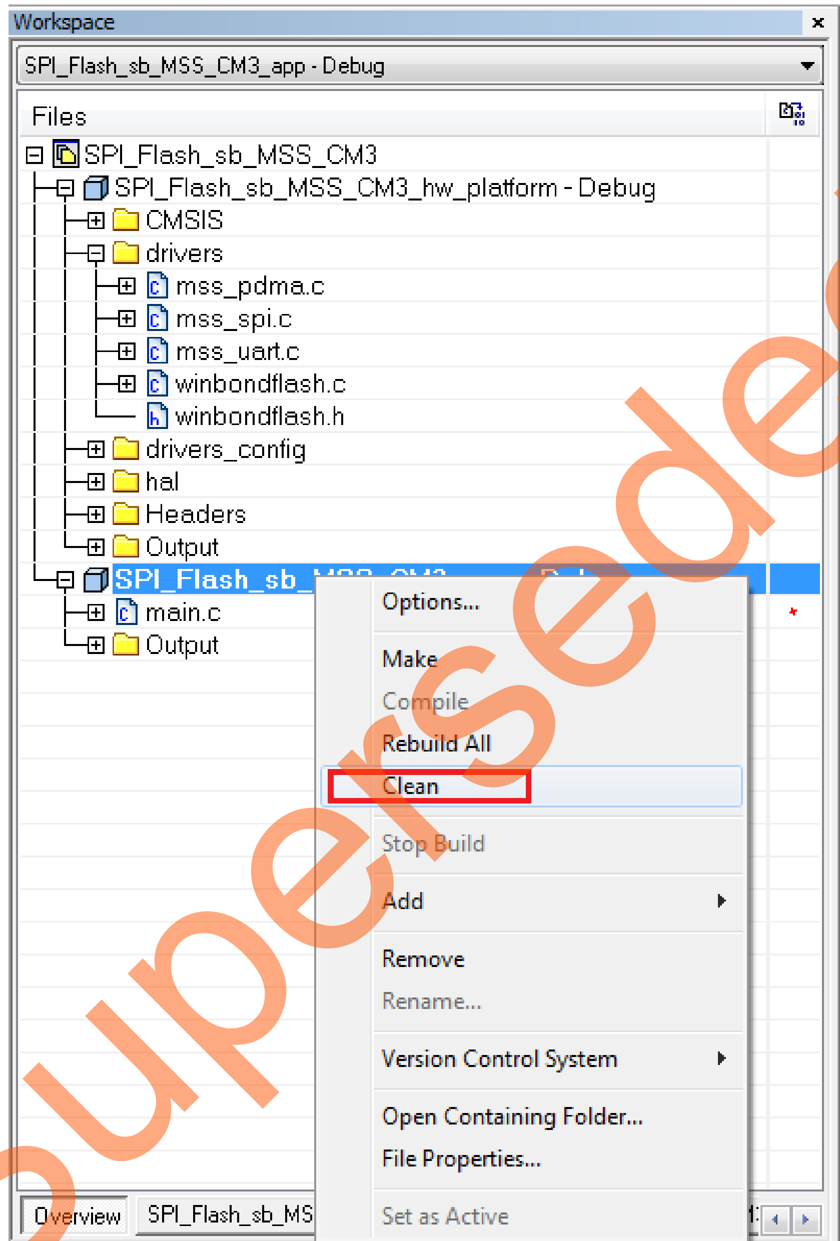

Figure 34 **• IAR Workspace - Execute Clean on SPI_Flash_sb_MSS_CM3_app Project**

18. After cleaning the project, the **Messages** log section shows that some files are deleted, as shown in Figure 35.

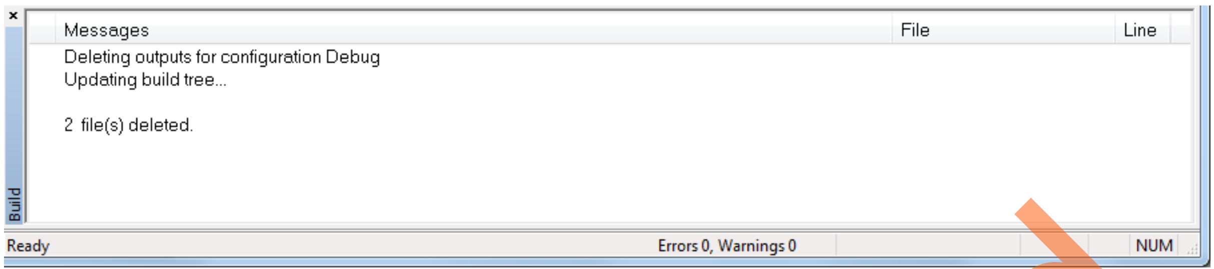

***Figure 35* • IAR Workspace - Deleted Files**

19. Right-click **SPI_Flash_sb_MSS_CM3_app - Debug** project name and click **Rebuild All**, as shown in Figure 36.

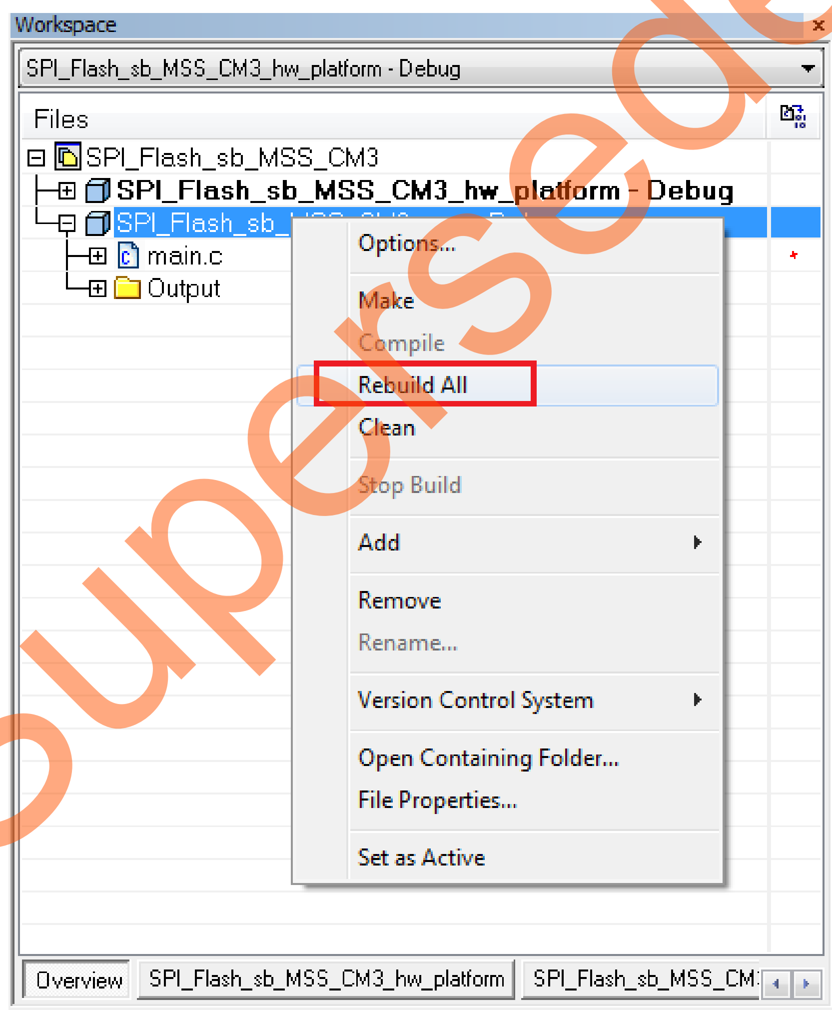

***Figure 36* • IAR Workspace - Select Rebuild All**

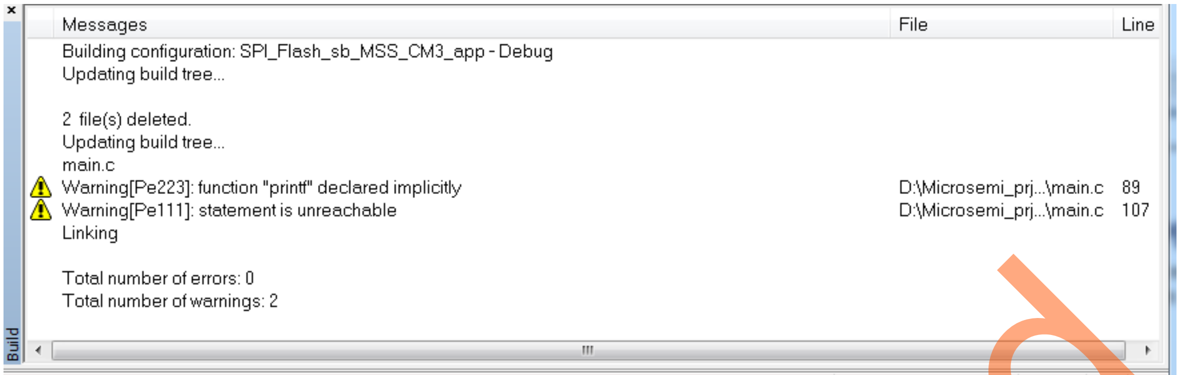

***Figure 37* • IAR Workspace - Rebuild All**

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. From the four COM ports, select the one with Location as **on USB Serial Converter D**. Figure 38 shows an example **Device Manager** window.

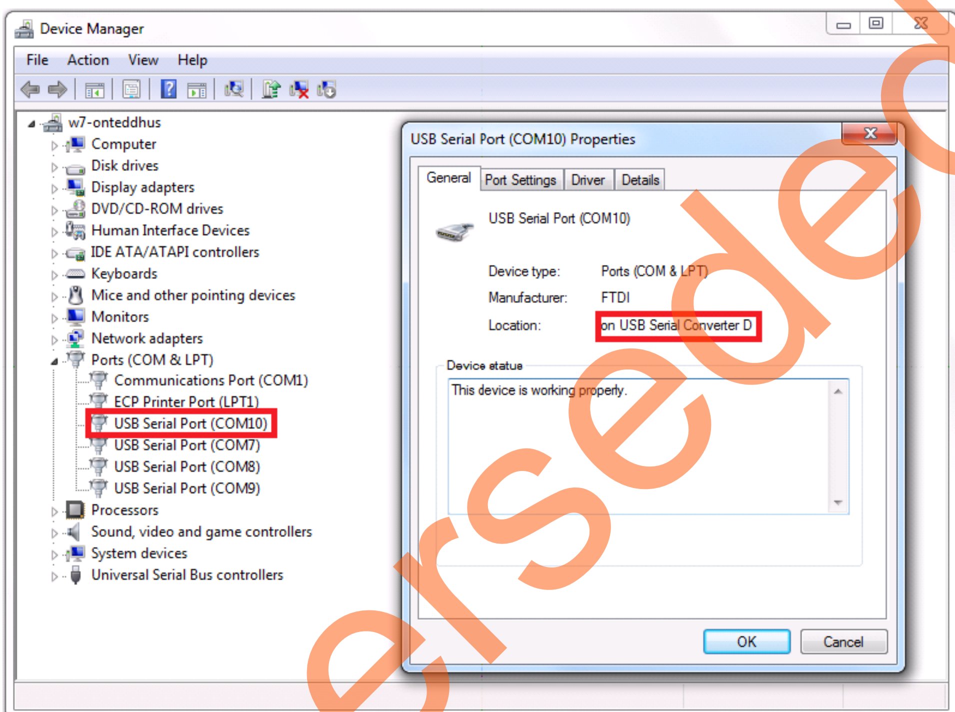

***Figure 38* • Device Manager Window**

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

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

Step 7: Debugging the Application Project using IAR Workbench

The following steps describe how to debug the application project using IAR Workbench:

1. Switch to **SPI_Flash_sb_MSS_CM3_app - Debug** tab from Overview tab as shown in Figure 39.

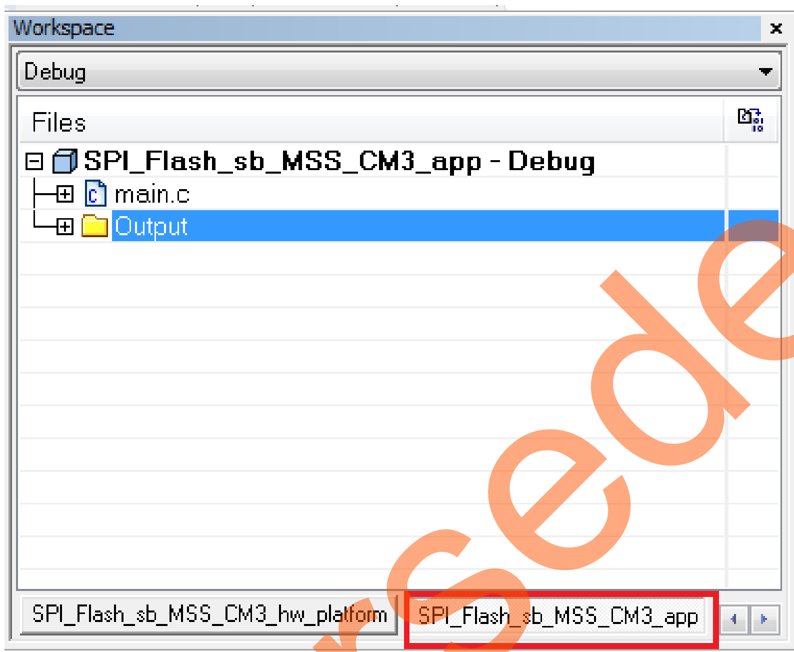

Figure 39 **• Debug Window**

2. In the IAR Workbench, click **Download and Debug** as shown in Figure 40.

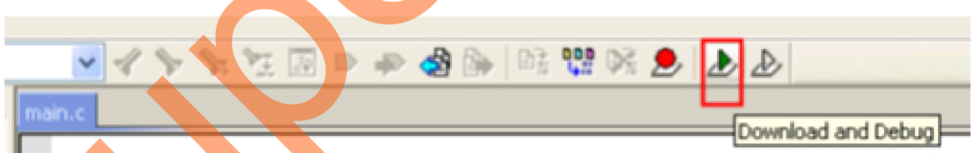

Figure 40 **• IAR Workbench - Download and Debug Option**

IAR Debugger Perspective window is opened, as shown in Figure 41.

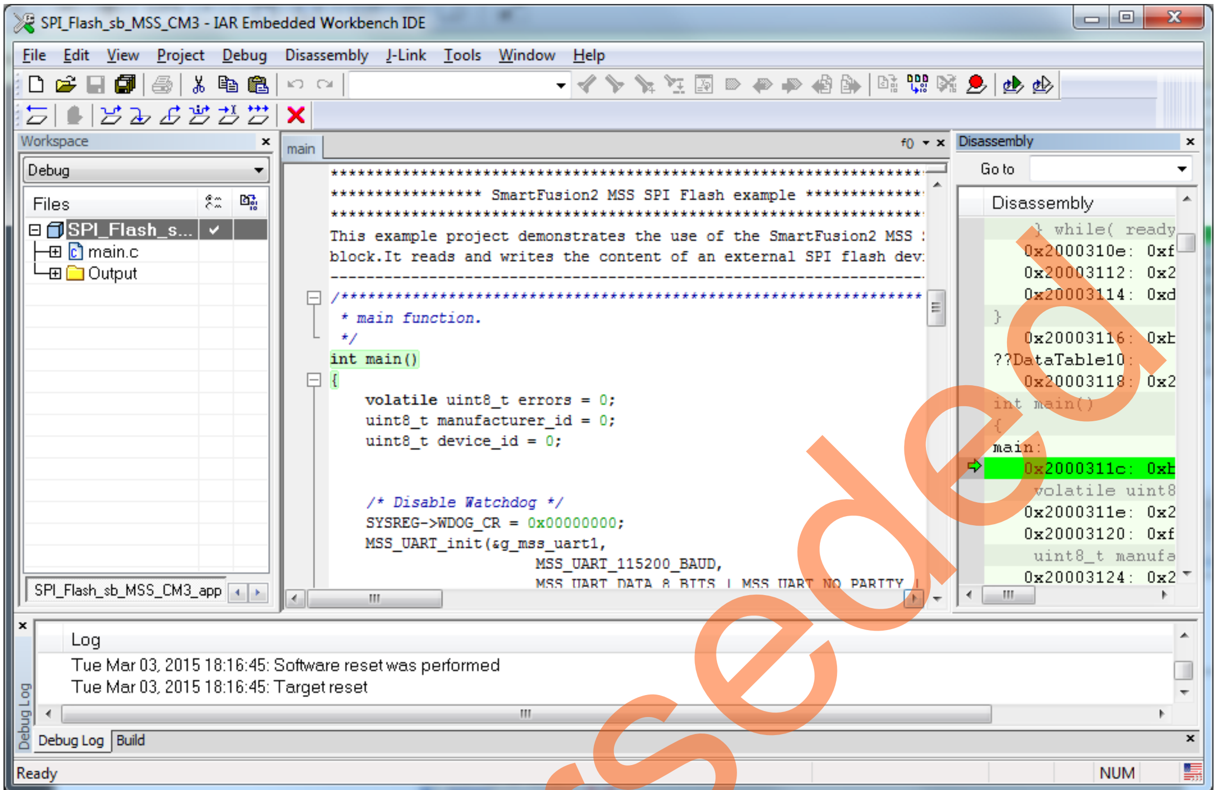

Figure 41 • **IAR Workbench - Debugger Perspective**

3. Click **Go** on IAR workbench to run the application, as shown in Figure 42.

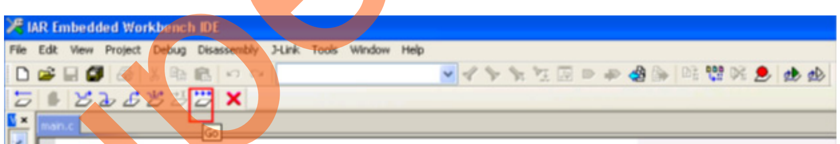

Figure 42 • **IAR Workbench - Go Option**

4. On successful operation, the HyperTerminal window displays a message, as shown in Figure 43.

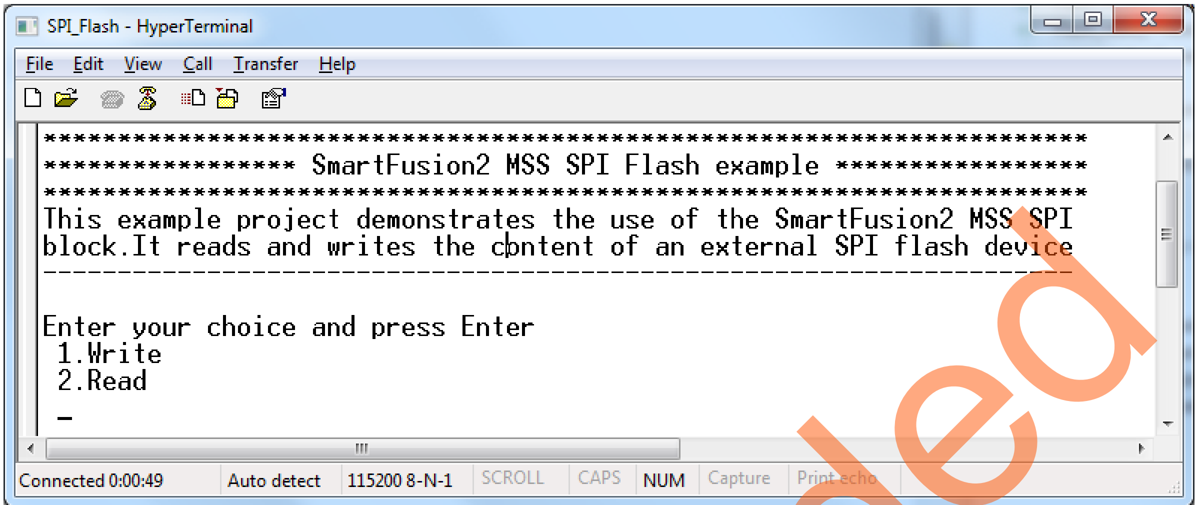

***Figure 43* • HyperTerminal Window**

5. Select option **1** and enter values to write to the SPI Flash Memory, as shown in Figure 44.

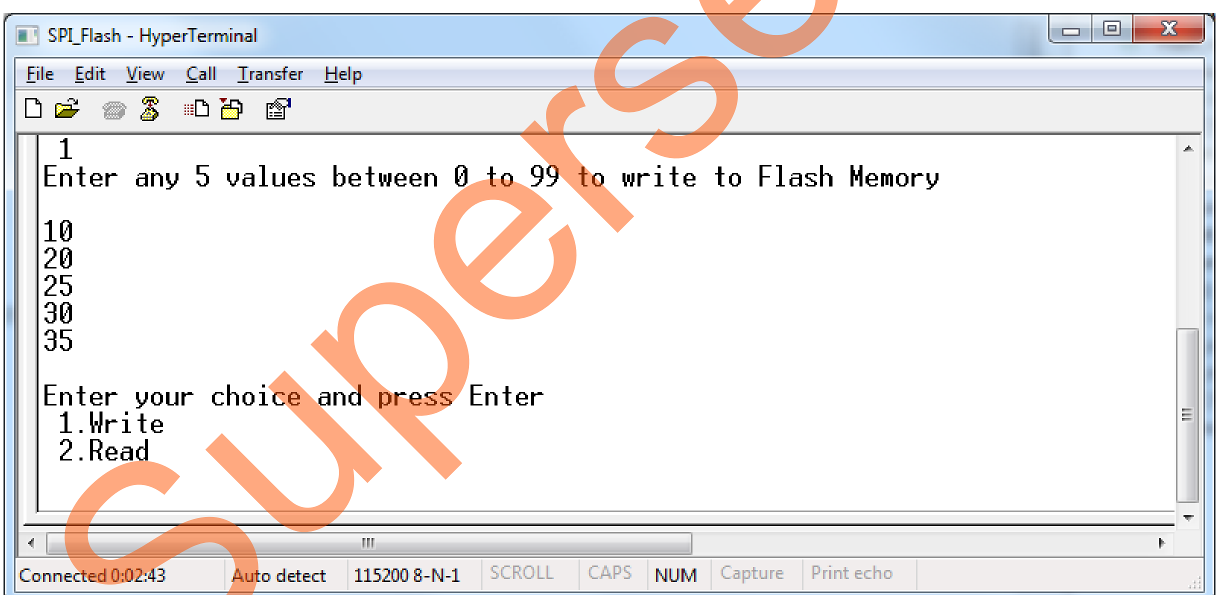

***Figure 44* • HyperTerminal Window - Option 1**

6. Select option **2** to read data from SPI Flash Memory, as shown in Figure 45.

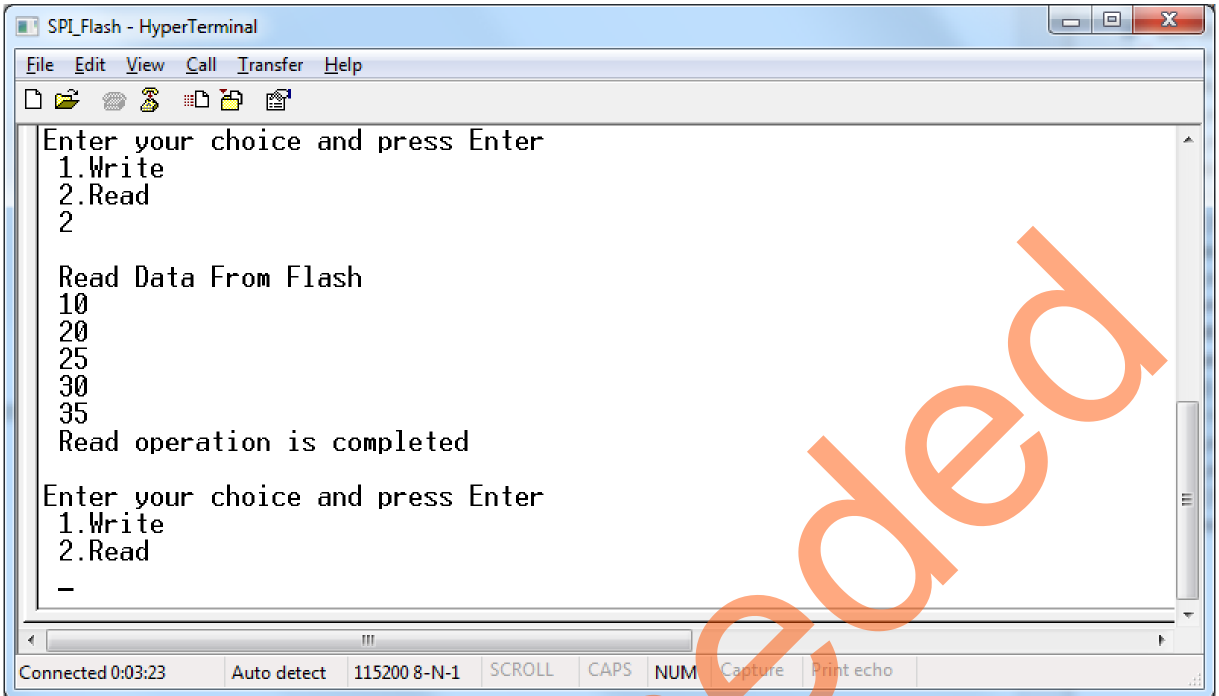

Figure 45 **• HyperTerminal Window - Option 2**

7. Click **View > Register** to view the values of the ARM® Cortex®-M3 processor internal registers, as shown in Figure 46.

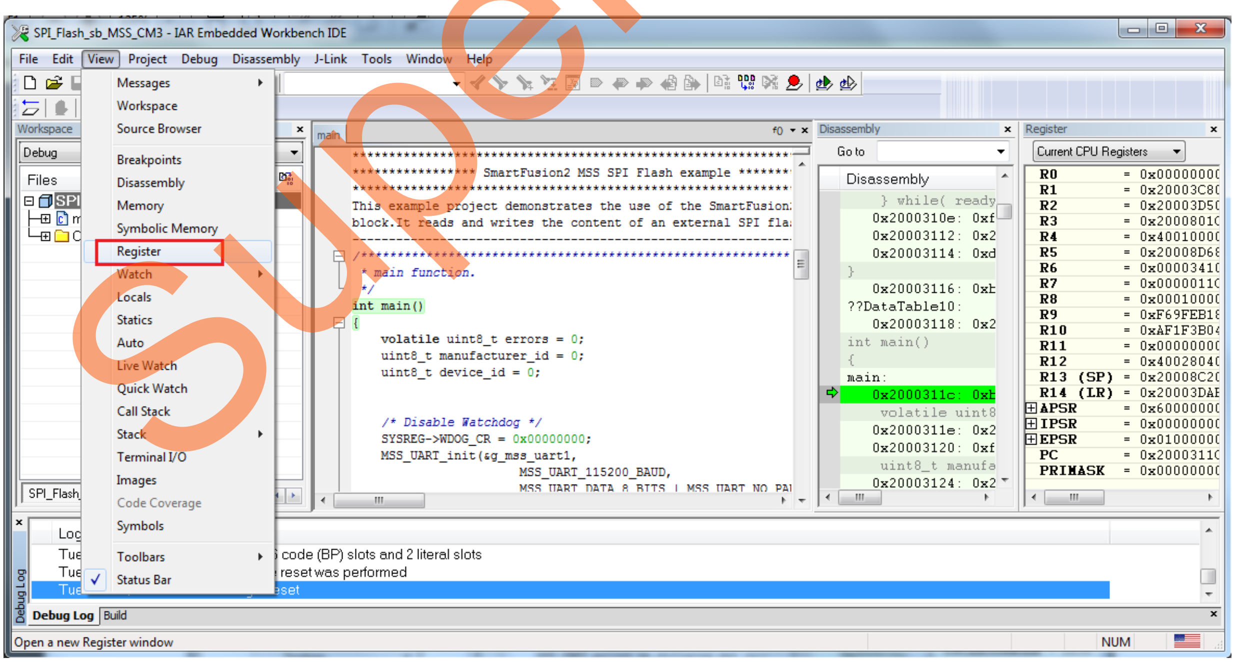

Figure 46 **• Values of Cortex-M3 Internal Registers**

8. Click **View > Statics** to view the values of variables in the source code, as shown in Figure 47.

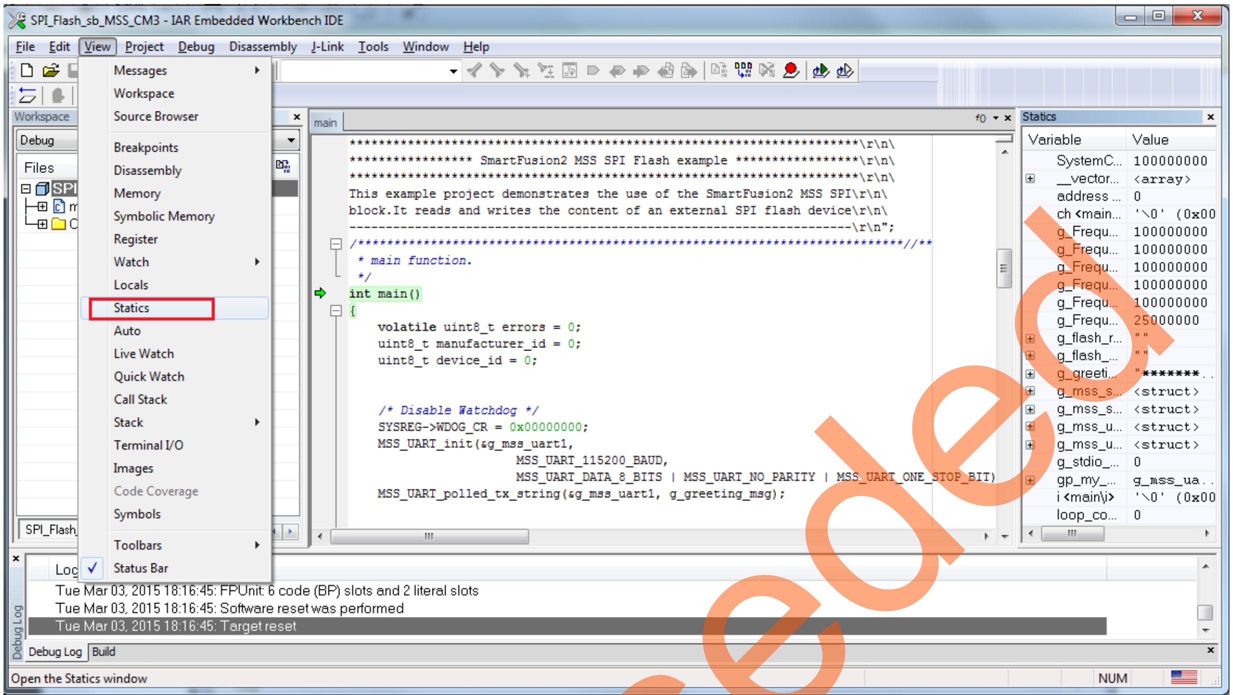

***Figure 47* • Values of Source Code Variables**

9. Click **View > Disassembly** to view the values of variables in the source code, as shown in Figure 48.

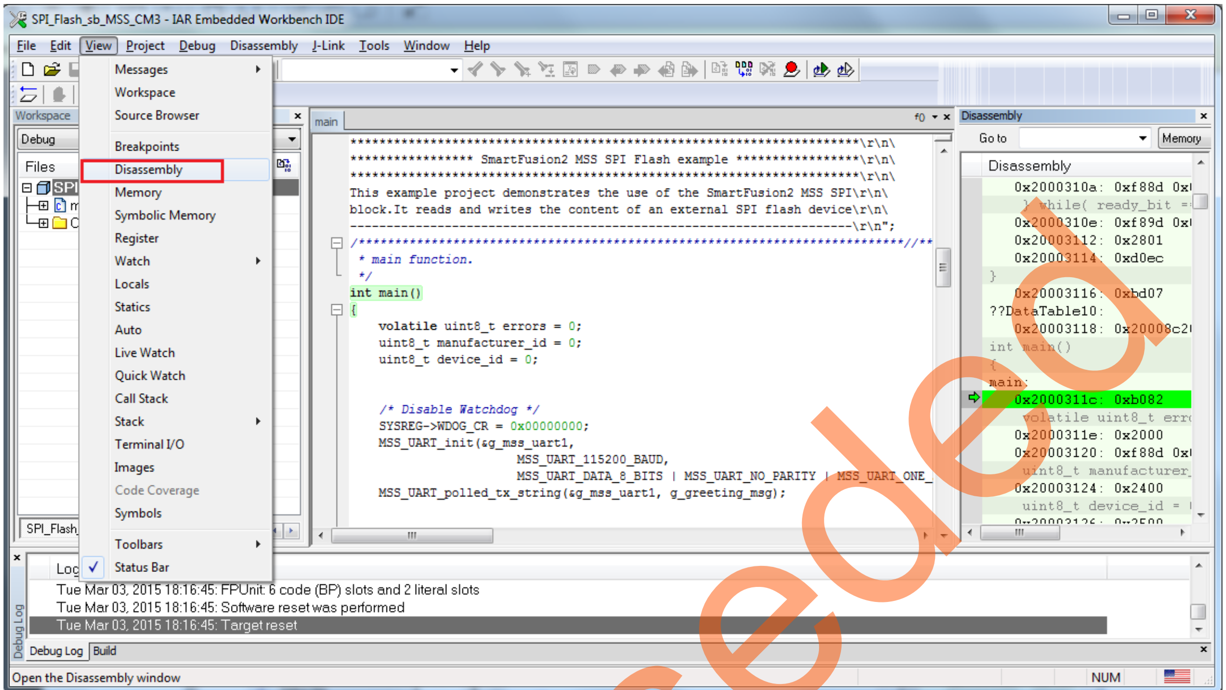

***Figure 48* • Assembly Level Instructions**

10. When debug process is finished, terminate execution of the code by choosing **Debug > Stop Debugging**, as shown in Figure 49.

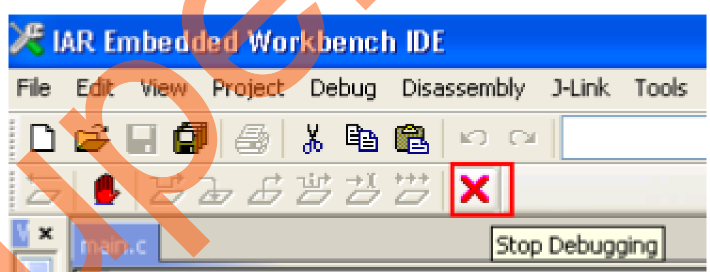

***Figure 49* • IAR Workbench - Stop Debugging Option**

11. The Step Level Debugging can be performed before running the application using Go. This can be accessed from the Debug menu or on the IAR workbench, as shown in Figure 50.

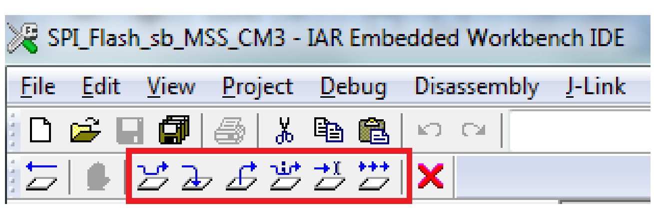

***Figure 50* • IAR Workbench - Step Level Debugging**

– Source code can be single-stepped by selecting from the Debug menu **Debug > Step Into**, **Debug > Step Out**, **Debug > Step Over** or selecting the respective options from the IAR workbench as shown in Figure 50. 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.

12. Close **Debug Perspective** by selecting **Close Perspective** from the Window menu.
13. Close IAR Embedded Workbench using **File > Exit**.
14. Close the HyperTerminal using **File > Exit**.

Conclusion

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

Appendix A: Board Setup for Programming the Tutorial

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

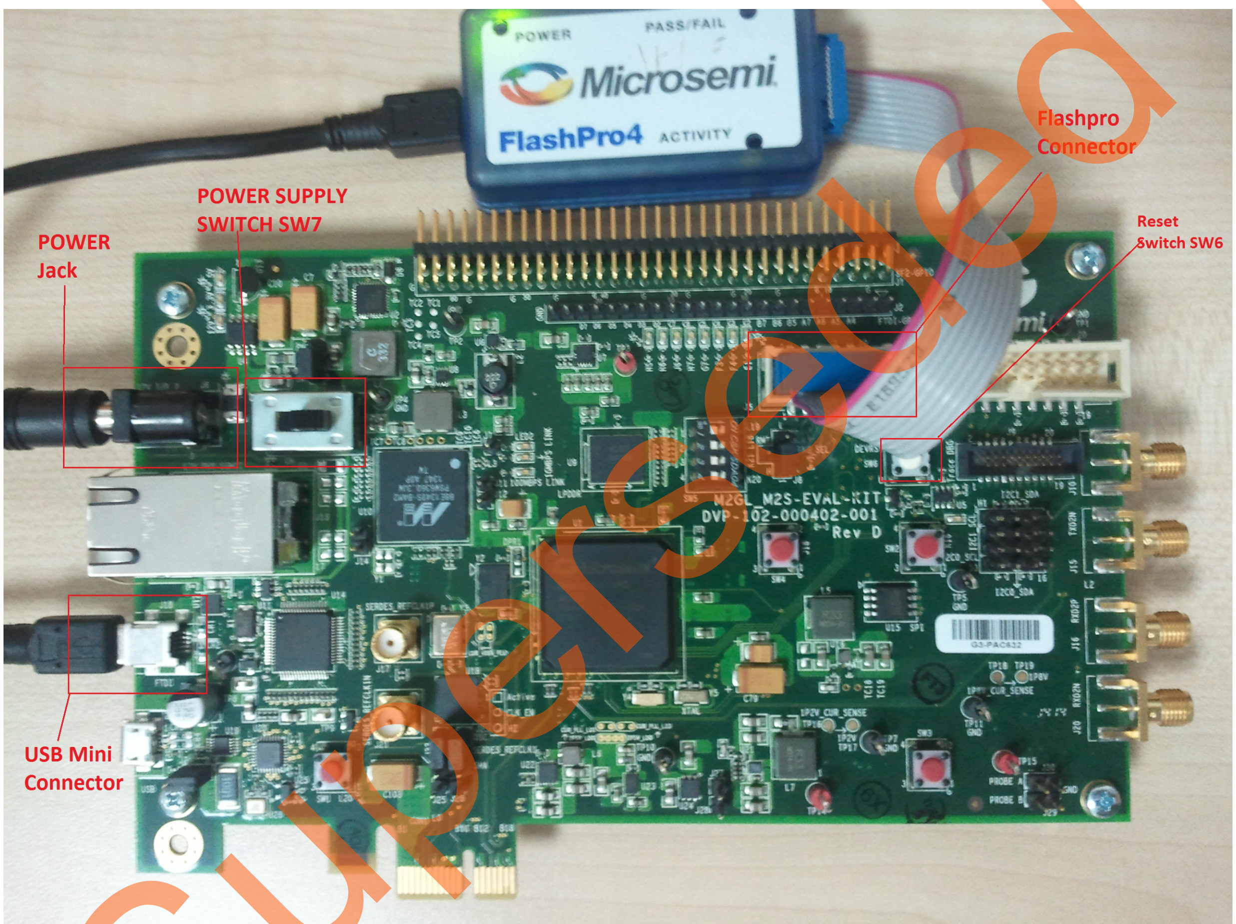

Figure 1 **• SmartFusion2 Security Evaluation Kit Setup**

Appendix B: Board Setup for Running the IAR Tutorial

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

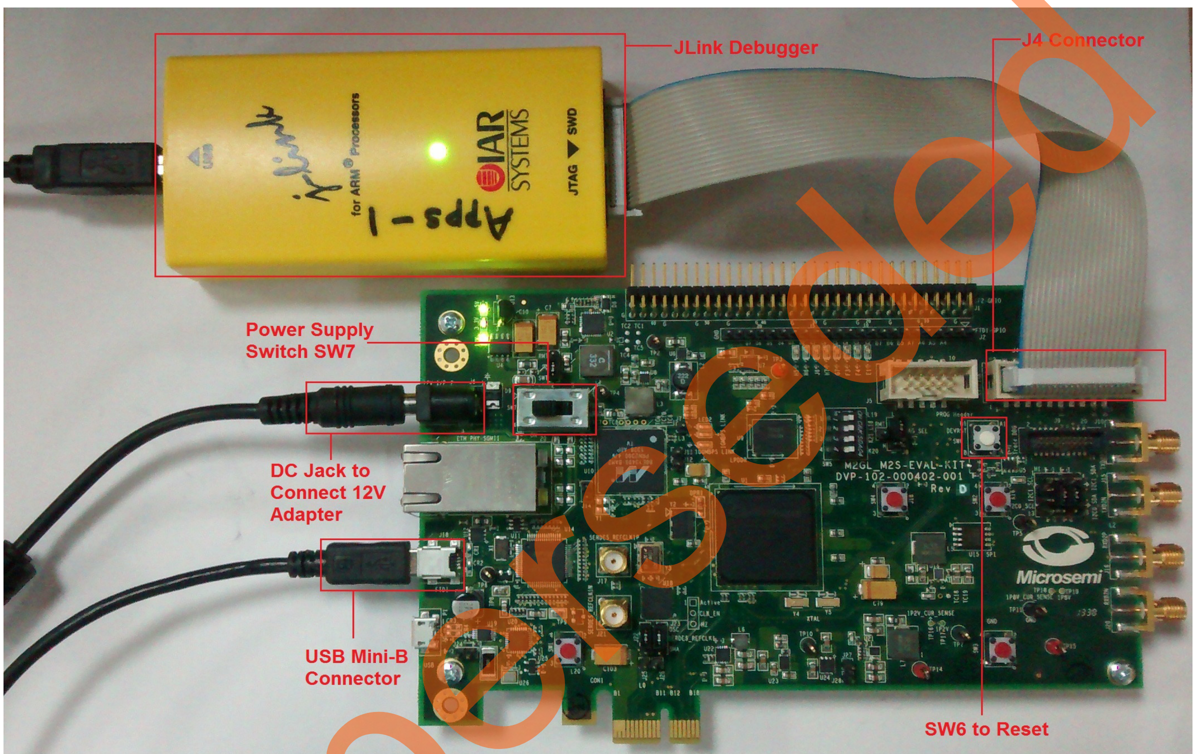

Figure 1 • **SmartFusion2 Security Evaluation Kit J-Link Programmer Connection**

Appendix C: SmartFusion2 Security Evaluation Kit Board Jumper Locations

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

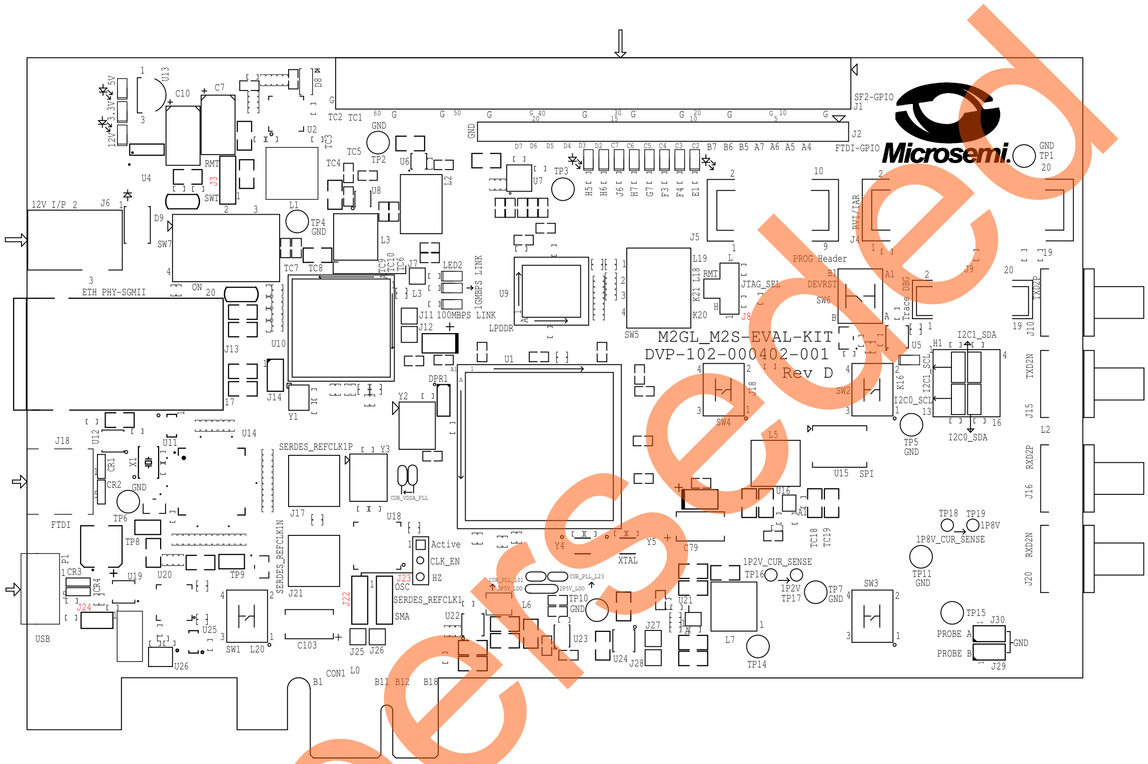

Figure 1 • **SmartFusion2 Security Evaluation Kit Board Jumper Locations**

Notes:

- Jumpers highlighted in red (J22, J23, J24, J8, and J3) 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 (SAR 72826)	N/A
Revision 3 (March 2015)	Updated the document for Libero SoC v11.5 software release (SAR 64188).	N/A
Revision 2 (November 2014)	Updated the document for Libero SoC v11.4 software release (SAR 61628).	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 SoC 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**.

50200547-4/10.15