

Accessing Serial Flash Memory Using SPI Interface - Libero SoC v11.7 and IAR Embedded Workbench Flow Tutorial for SmartFusion2

TU0547 Tutorial



Power Matters.™

Contents

1	Preface	6
1.1	Purpose	6
1.2	Intended Audience	6
2	Accessing Serial Flash Memory using SPI Interface - Libero SoC v11.7 and IAR Embedded Workbench Flow Tutorial for SmartFusion2 7	
2.1	Introduction	7
2.2	Design Requirements	8
2.2.1	Project Files	8
2.3	Design Overview	8
2.4	Step 1: Creating a Libero SoC Project	10
2.4.1	Launching Libero SoC	10
2.4.2	Connecting Components in SPI_Flash SmartDesign	18
2.5	Step 2: Generating the Program File	19
2.6	Step 3: Programming SmartFusion2 Security Evaluation Board Using FlashPro	20
2.7	Step 4: Configuring and Generating Firmware	22
2.8	Step 5: Building Software Application using IAR Embedded Workbench	24
2.9	Step 6: Configuring Serial Terminal Emulation Program	38
2.10	Step 7: Debugging the Application Project using IAR Workbench	39
2.11	Conclusion	45
3	Appendix: Board Setup for Programming the Tutorial	46
4	Appendix: Board Setup for Running the IAR Tutorial	47
5	Appendix: SmartFusion2 Security Evaluation Kit Board Jumper Locations	48
6	Revision History	49
7	Product Support	50
7.1	Customer Service	50
7.2	Customer Technical Support Center	50
7.3	Technical Support	50
7.4	Website	50
7.5	Contacting the Customer Technical Support Center	50
7.5.1	Email	50
7.5.2	My Cases	50
7.5.3	Outside the U.S.	51
7.6	ITAR Technical Support	51

Figures

Figure 1.	SPI Flash Interfacing Block Diagram	9
Figure 2.	Libero SoC Project Manager	10
Figure 3.	Project Details Page	11
Figure 4.	Device Selection Page	12
Figure 5.	Device Settings	13
Figure 6.	Device Template Page	13
Figure 7.	New Project Information Window	14
Figure 8.	System Builder Window	14
Figure 9.	System Builder – Device Features Page	15
Figure 10.	System Builder – Peripherals Page	16
Figure 11.	System Builder – Clock Page	17
Figure 12.	SPI_Flash Smart Design	18
Figure 13.	SPI_Flash Smart Design	18
Figure 14.	Generate Component	19
Figure 15.	Log Window	19
Figure 16.	Manage Constraints	19
Figure 17.	Timing - Derive Constraints Tab	20
Figure 18.	Associate the Derive Constraint SDC file - Message	20
Figure 19.	Generate Bitstream	20
Figure 20.	Run Program Action	21
Figure 21.	Configuring Firmware	22
Figure 22.	Export Firmware Dialog Box	22
Figure 23.	Firmware Successfully Exported Message	23
Figure 24.	Invoking IAR Embedded Workbench from Libero SoC Software	24
Figure 25.	IAR Workspace	25
Figure 26.	IAR Workspace main.c file	25
Figure 27.	IAR Workspace Window - Add winbondflash SPI Driver Files	26
Figure 28.	IAR Workspace Window - Display winbondflash SPI Driver Files	27
Figure 29.	IAR Workspace Window - Choose Options	28
Figure 30.	IAR Workspace Window - Adding Symbol	29
Figure 31.	IAR Workspace Window - Choose Options	30
Figure 32.	IAR Node Options	31
Figure 33.	IAR Debugger Options - Selecting Driver	32
Figure 34.	IAR Debugger Options - Download	33
Figure 35.	IAR Workspace - Hardware Platform Code Compilation using Make	34
Figure 36.	IAR Workspace - Successful Hardware Platform Code Compilation using Make	34
Figure 37.	IAR Workspace - SPI_Flash_sb_MSS_CM3_app Set as Active	35
Figure 38.	IAR Workspace - Execute Clean on SPI_Flash_sb_MSS_CM3_app Project	36
Figure 39.	IAR Workspace - Deleted Files	37
Figure 40.	IAR Workspace - Select Rebuild All	37
Figure 41.	IAR Workspace - Rebuild All	38
Figure 42.	Device Manager Window	38
Figure 43.	Debug Window	39
Figure 44.	IAR Workbench - Download and Debug Option	39
Figure 45.	IAR Workbench - Debugger Perspective	40
Figure 46.	IAR Workbench - Go Option	40
Figure 47.	HyperTerminal Window	41
Figure 48.	HyperTerminal Window - Option 1	41
Figure 49.	HyperTerminal Window - Option 2	42
Figure 50.	Values of Cortex-M3 Internal Registers	43
Figure 51.	Values of Source Code Variables	43
Figure 52.	Assembly Level Instructions	44

Figure 53.	IAR Workbench - Stop Debugging Option	44
Figure 54.	IAR Workbench - Step Level Debugging	44
Figure 55.	SmartFusion2 Security Evaluation Kit Setup	46
Figure 56.	SmartFusion2 Security Evaluation Kit J-Link Programmer Connection	47
Figure 57.	SmartFusion2 Security Evaluation Kit Board Jumper Locations	48

Tables

Table 1.	Design Requirements	8
Table 2.	SmartFusion2 Security Evaluation Kit Jumper Settings	20

1 Preface

1.1 Purpose

This document contains the following sections:

- [Accessing Serial Flash Memory using SPI Interface - Libero SoC v11.7 and IAR Embedded Workbench Flow Tutorial for SmartFusion2](#)

1.2 Intended Audience

Triton devices are used by:

- FPGA designers
- Embedded designers
- System-level designers

2 Accessing Serial Flash Memory using SPI Interface - Libero SoC v11.7 and IAR Embedded Workbench Flow Tutorial for SmartFusion2

2.1 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: SoftConsole v4.0 and Libero SoC v11.7](#)
- [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

2.2 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	v11.7
FlashPro programming software	v11.7
IAR Embedded Workbench for ARM	v6.4
Host PC Drivers	USB to UART drivers
SoftConsole	v3.4 SP1* <i>Note: *For this application note, SoftConsole v3.4 SP1 is used. For using SoftConsole v4.0, see the TU0546: SoftConsole v4.0 and Libero SoC v11.7 Tutorial.</i>
One of the following serial terminal emulation programs: - HyperTerminal - TeraTerm - PuTTY	—

2.2.1 Project Files

The design files for this tutorial can be downloaded from the Microsemi® website:

http://soc.microsemi.com/download/rsc/?f=m2s_tu0547_liberov11p7_df

The design files include:

- Libero project
- Programming files
- Source files
- SPI_Flash_Drivers
- Readme files

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

2.3 Design Overview

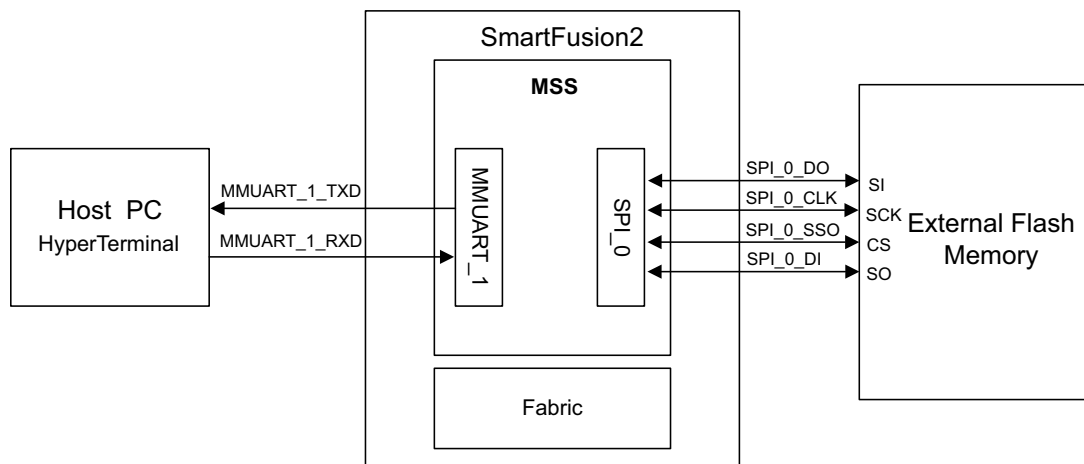
This design example demonstrates the execution of basic read and write operations on the SPI flash present on the SmartFusion2 Security Evaluation Kit board. This kit has a built-in winbond SPI flash memory, named W25Q64FVSSIG, which is connected to the SmartFusion2 microcontroller subsystem (MSS) through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



2.4 Step 1: Creating a Libero SoC Project

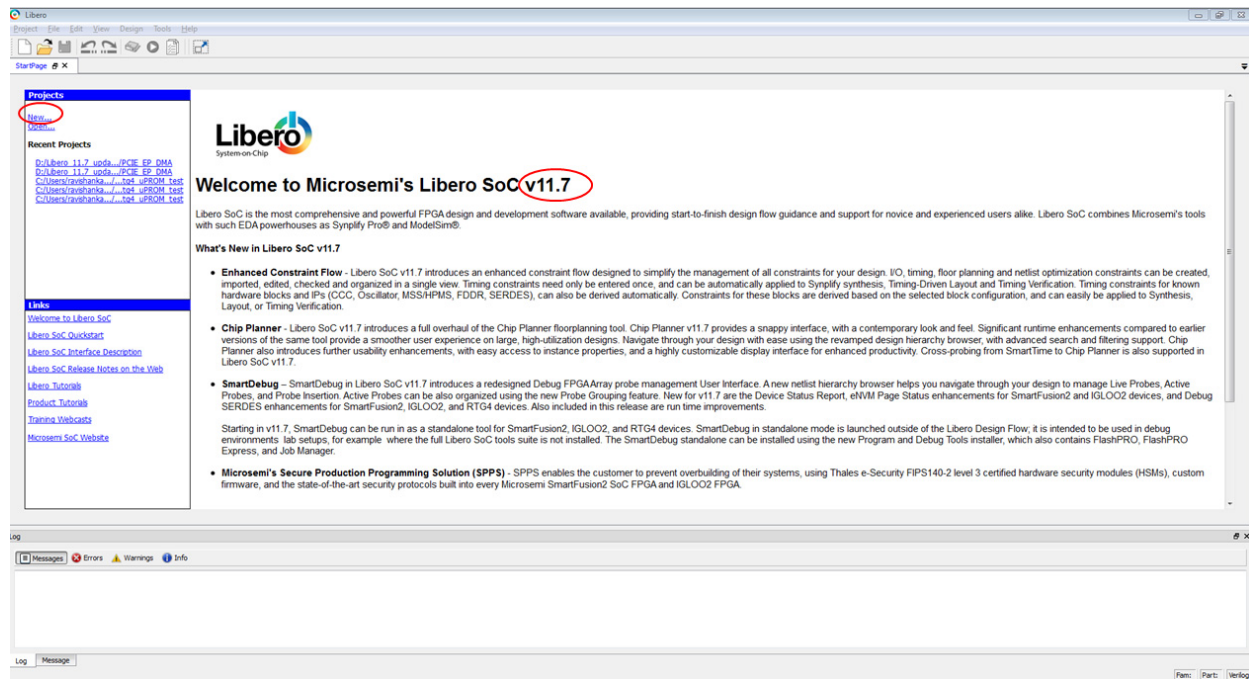
The following steps describe how to create a Libero SoC project:

2.4.1 Launching Libero SoC

The following steps describe how to launch Libero SoC:

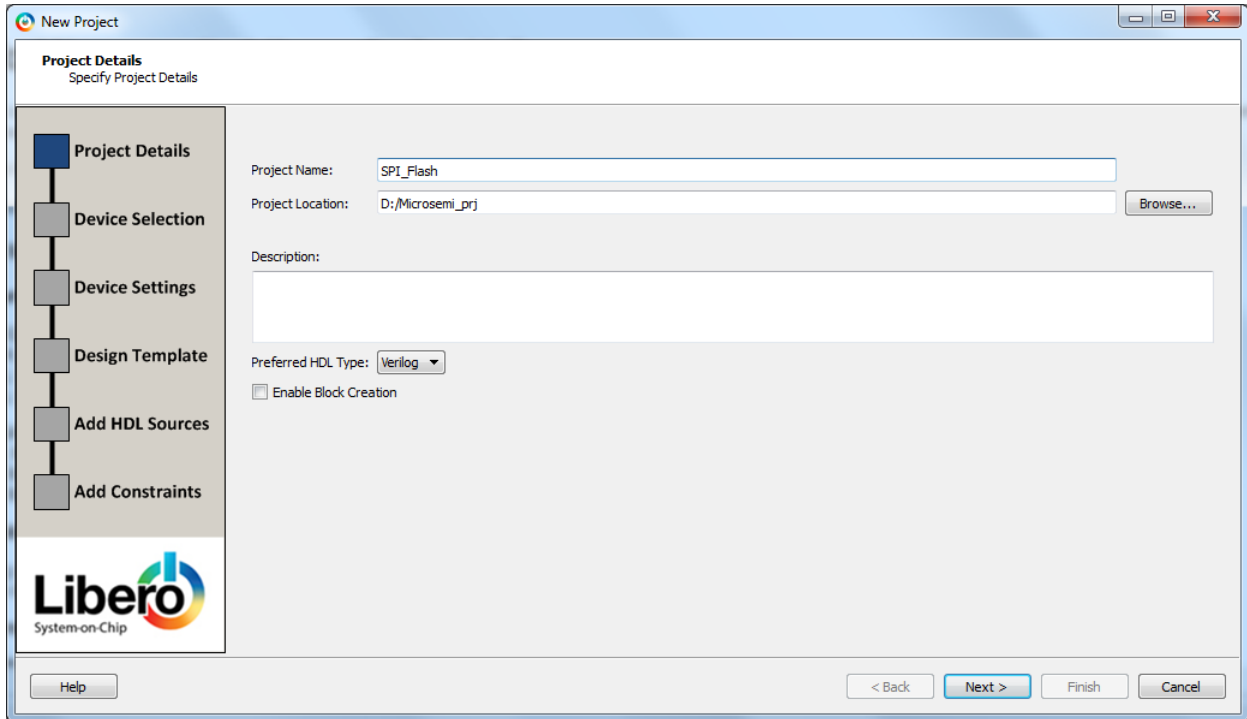
1. Click **Start > Programs > Microsemi Libero SoC v11.7 > Libero SoC v11.7**, or double-click the shortcut on desktop to open the Libero SoC v11.7 Project Manager.
2. Create a new project using one of the following options:
 - Select **New** on the **Start Page** tab as shown in [Figure 2](#).
 - Click **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

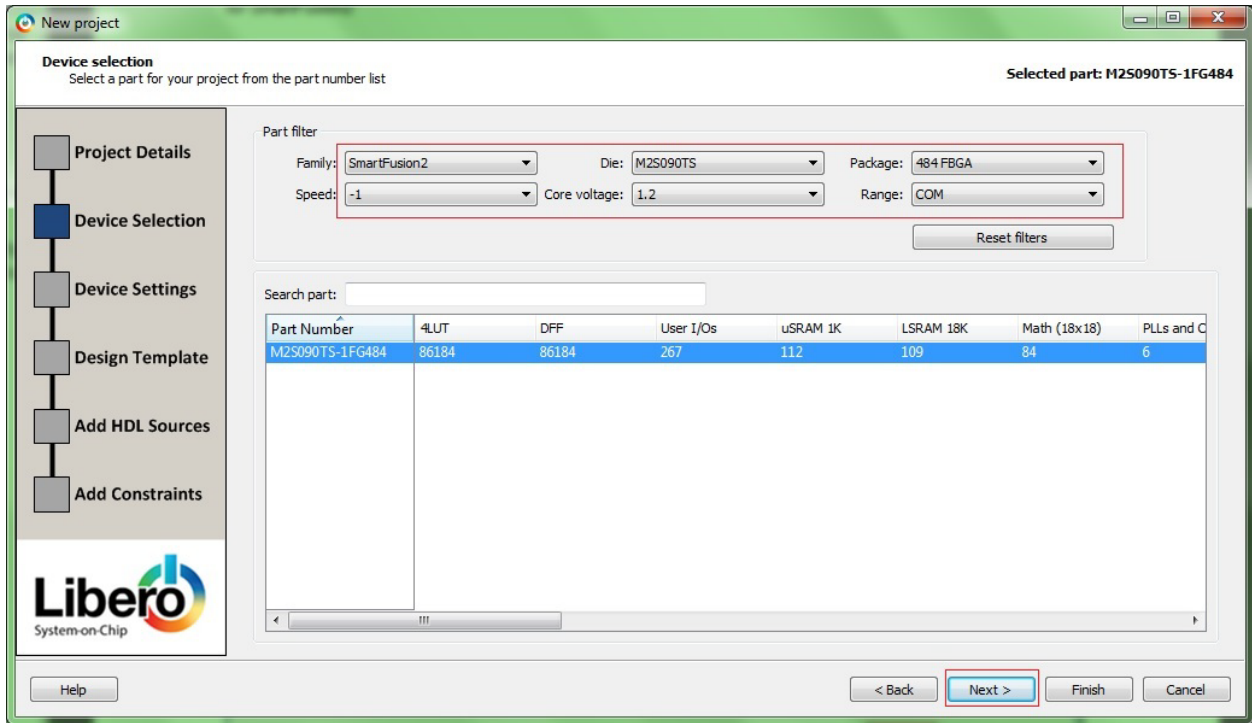
Figure 3 • Project Details Page



The screenshot shows the 'New Project' dialog box with the 'Project Details' tab selected. The 'Project Name' field contains 'SPI_Flash'. The 'Project Location' field contains 'D:/Microsemi_prj' and has a 'Browse...' button next to it. The 'Description' field is empty. The 'Preferred HDL Type' is set to 'Verilog' with a dropdown arrow. The 'Enable Block Creation' checkbox is unchecked. The Libero System-on-Chip logo is visible in the bottom left. At the bottom right, there are buttons for '< Back', 'Next >', 'Finish', and 'Cancel'.

4. Click **Next**. This opens **Device Selection** page as shown in [Figure 4](#).
5. 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



New project

Device selection
Select a part for your project from the part number list

Selected part: M2S090TS-1FG484

Part filter

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

Reset filters

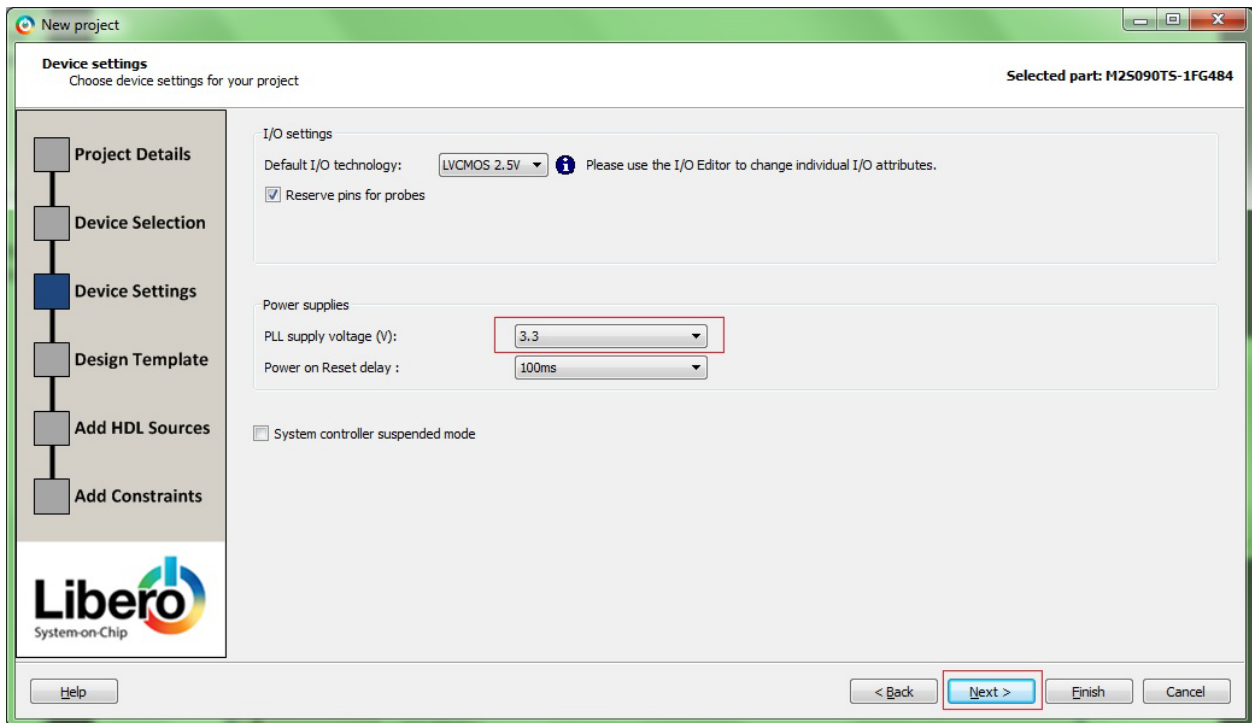
Search part:

Part Number	4LUT	DFF	User I/Os	uSRAM 1K	LSRAM 18K	Math (18x18)	PLLs and C
M2S090TS-1FG484	86184	86184	267	112	109	84	6

Help < Back **Next >** Finish Cancel

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

Figure 5 • Device Settings



New project

Device settings
Choose device settings for your project

Selected part: M25090TS-1FG484

Project Details

Device Selection

Device Settings

Design Template

Add HDL Sources

Add Constraints

Libero
System-on-Chip

I/O settings

Default I/O technology: LVC MOS 2.5V  Please use the I/O Editor to change individual I/O attributes.

☒ Reserve pins for probes

Power supplies

PLL supply voltage (V): 3.3

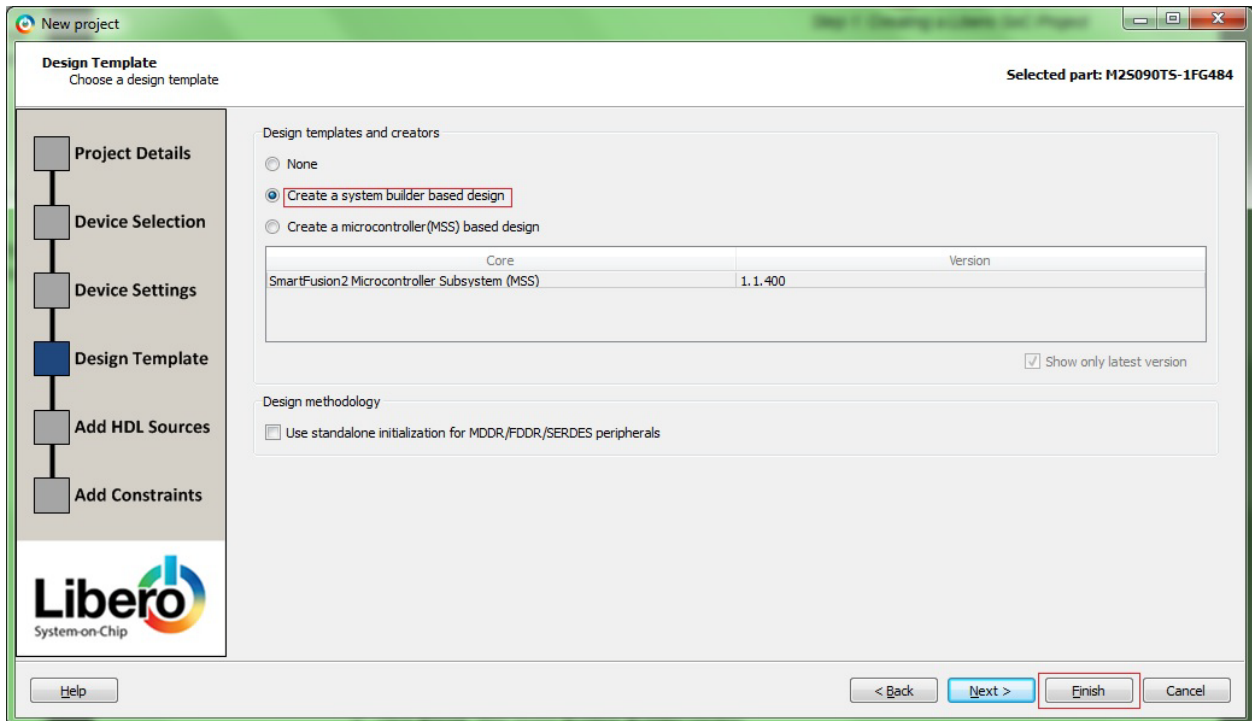
Power on Reset delay : 100ms

☐ System controller suspended mode

Help < Back **Next >** Finish Cancel

- 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



New project

Design Template
Choose a design template

Selected part: M25090TS-1FG484

Project Details

Device Selection

Device Settings

Design Template

Add HDL Sources

Add Constraints

Libero
System-on-Chip

Design templates and creators

☐ None

☒ Create a system builder based design

☐ Create a microcontroller(MSS) based design

Core	Version
SmartFusion2 Microcontroller Subsystem (MSS)	1.1.400

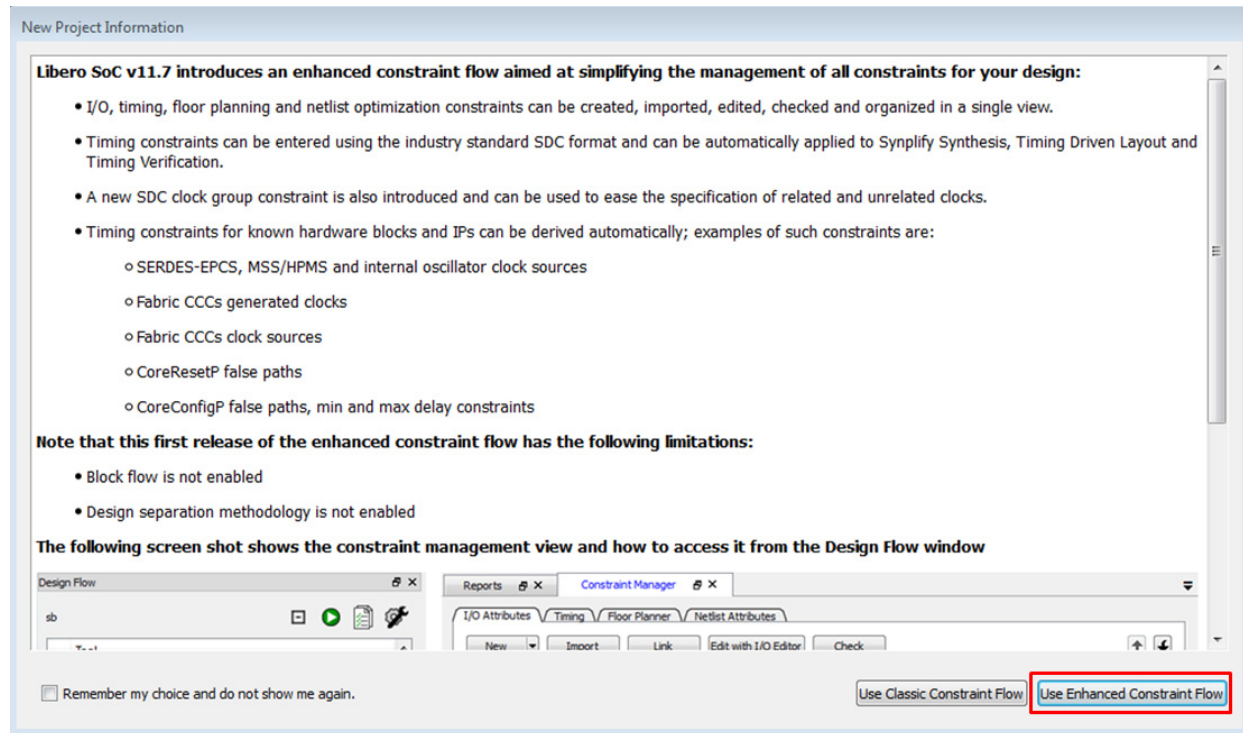
☒ Show only latest version

Design methodology

☐ Use standalone initialization for MDDR/FDDR/SERDES peripherals

Help < Back **Next >** **Finish** Cancel

Figure 7 • New Project Information Window



8. Click **Finish**. **New Project Information** window is displayed. Select **Use Enhanced Constraint Flow** option to use the new constraint flow as shown in [Figure 7](#).
 9. **System Builder** window is displayed. Enter the name of the system as **SPI_Flash** and click **OK**, as shown in [Figure 8](#).
- 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.

Figure 8 • System Builder Window

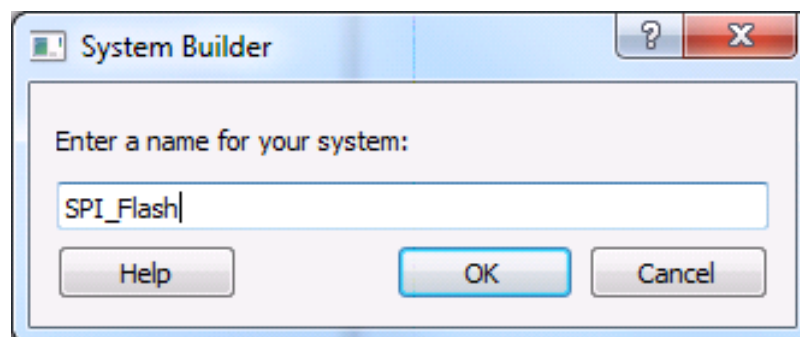
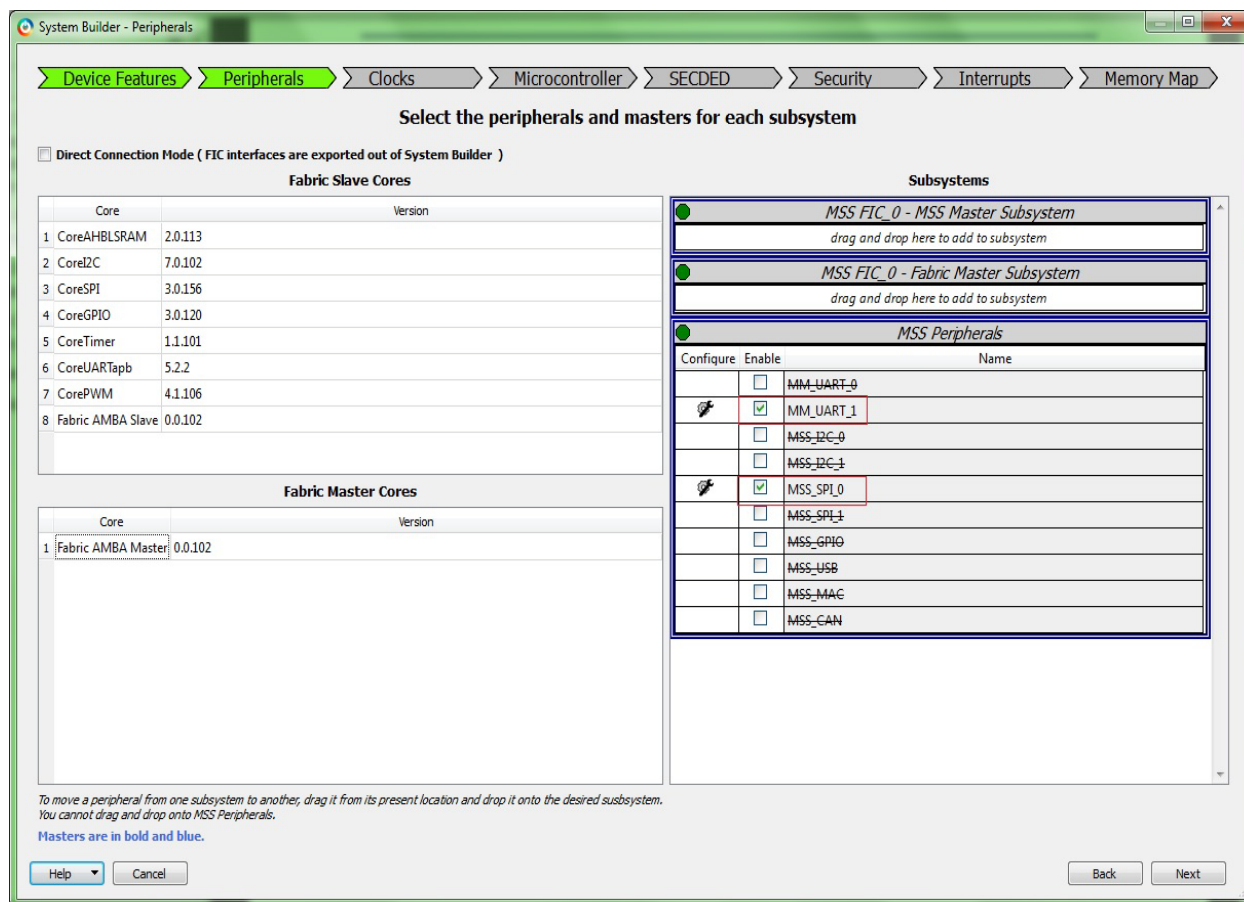


Figure 9 • System Builder – Device Features Page



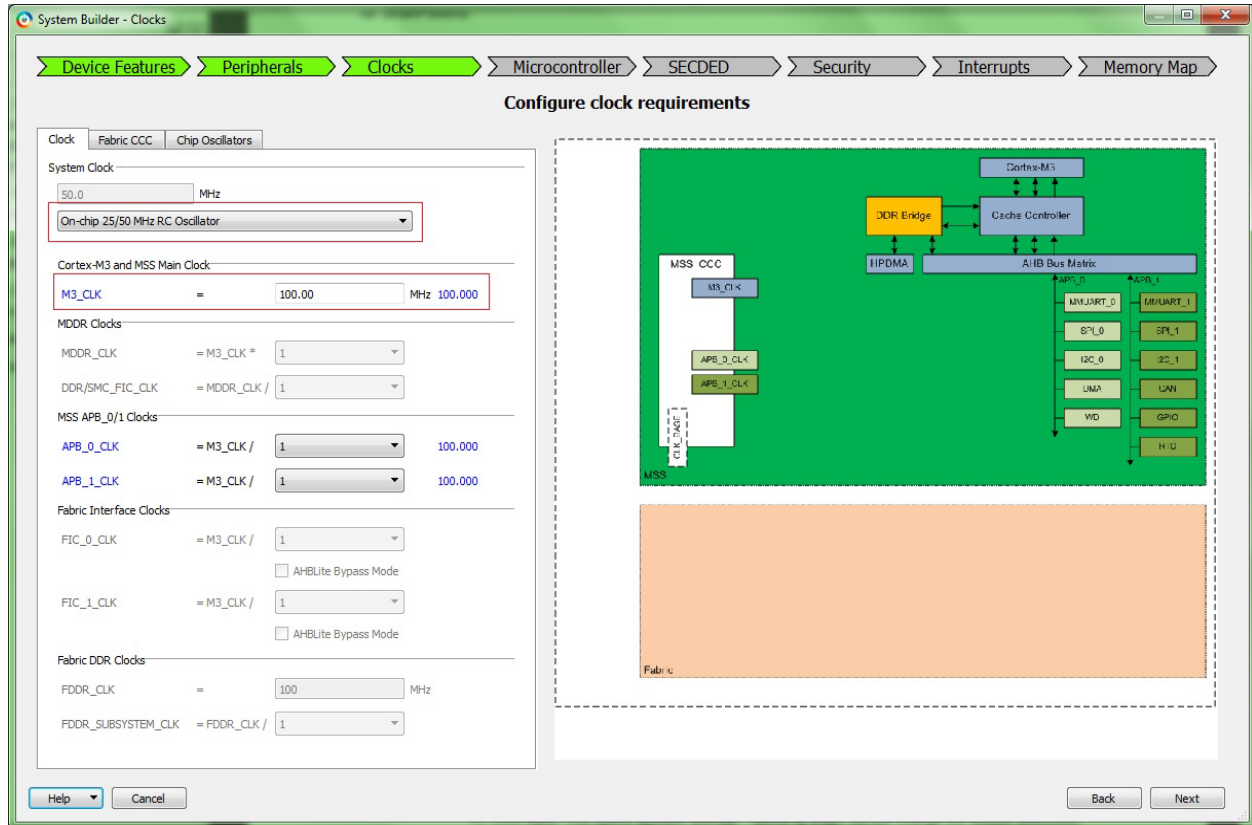
10. Click **Next**. This opens **System Builder - Peripherals** page as shown in Figure 10.
11. Under the MSS Peripherals section, clear all the check boxes except **MM_UART_1** and **MSS_SPI_0**, as shown in Figure 10.

Figure 10 • System Builder – Peripherals Page



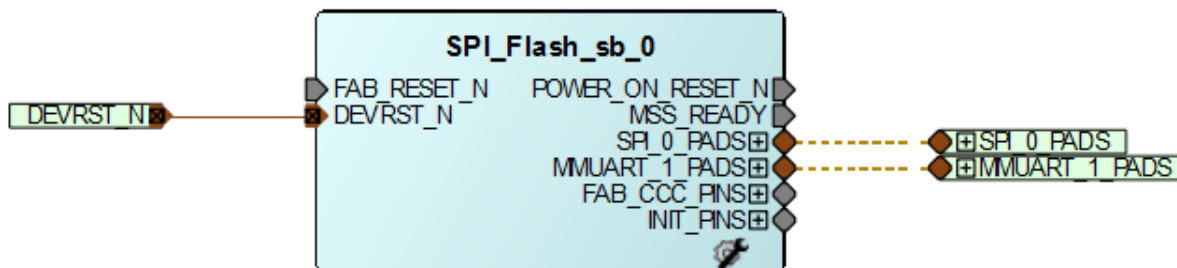
12. Click **Next**. This opens **System Builder - Clocks** page as shown in Figure 11.
13. In the **System Builder - Clocks** page (Refer to Figure 11):
 - Select **System Clock** frequency as **50 MHz** and clock source as **On-chip 25/50 MHz RC Oscillator**
 - Select **M3_CLK** as **100 MHz**
 - Select **APB_0_CLK** and **APB_1_CLK** frequency as **M3_CLK/1**
 - Do not change the default settings of remaining parameters.

Figure 11 • System Builder – Clock Page



14. Click **Next**. This opens **System Builder - Microcontroller** page. Do not change the default selections.
15. Click **Next**. This opens **System Builder - SECEDED** page. Do not change the default selections.
16. Click **Next**. This opens **System Builder - Security** page. Do not change the default selections.
17. Click **Next**. This opens **System Builder - Interrupts** page. Do not change the default selections.
18. Click **Next**. This opens **System Builder - Memory Map** page. 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 12](#).

Figure 12 • SPI_Flash Smart Design



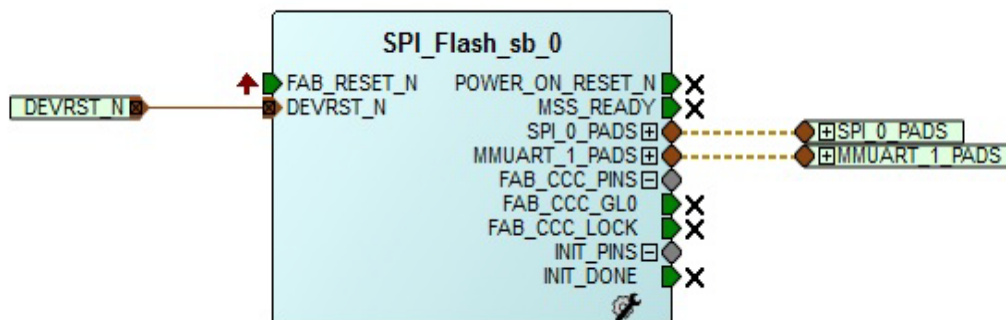
2.4.2 Connecting Components in SPI_Flash SmartDesign

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_GLO** 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**.

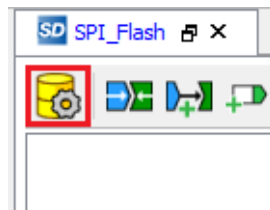
The **SPI_Flash** design is displayed, as shown in [Figure 13](#).

Figure 13 • 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 14.

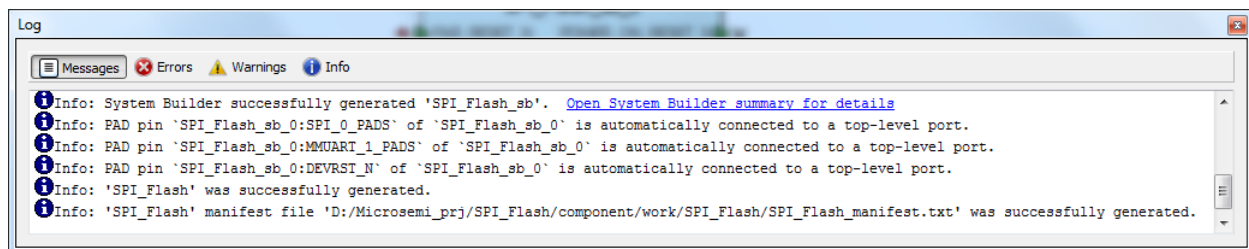
Figure 14 • Generate Component



After successful generation of all the components, the following message is displayed on the log window, as shown in Figure 15.

Info: 'SPI_Flash' was successfully generated.

Figure 15 • Log Window



2.5 Step 2: Generating the Program File

The following step describes how to generate the program file:

1. Double-click **Manage Constraints** and click **Derive Constraints** option under **Timing** tab to generate SDC file for root module.
2. Click **Yes** to associate SDC file to Synthesis, Place and Route and Timing Verification stages, as shown in Figure 16, Figure 17 on page 20, and Figure 18 on page 20.

Figure 16 • Manage Constraints

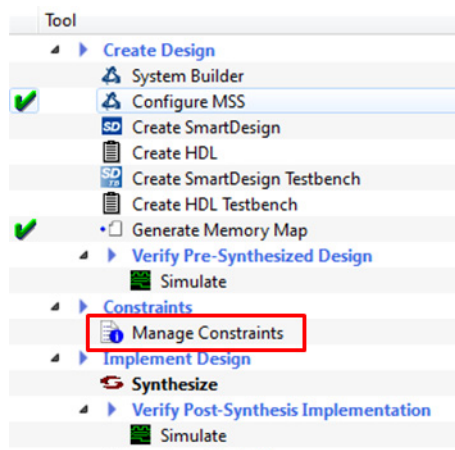


Figure 17 • Timing - Derive Constraints Tab

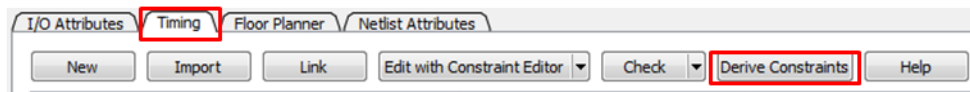
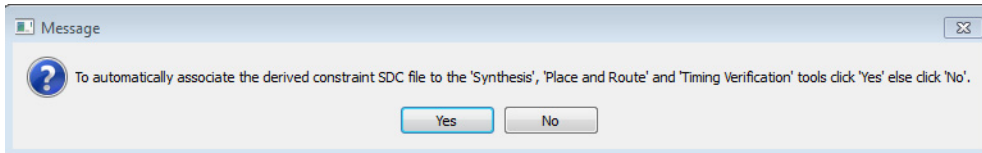


Figure 18 • Associate the Derive Constraint SDC file - Message



3. Click **Generate Bitstream** as shown in Figure 19 to complete place and route, and generate the programming file.

Figure 19 • Generate Bitstream



2.6 Step 3: Programming SmartFusion2 Security Evaluation Board Using FlashPro

The following steps describe how to program the SmartFusion2 Security Evaluation board using FlashPro:

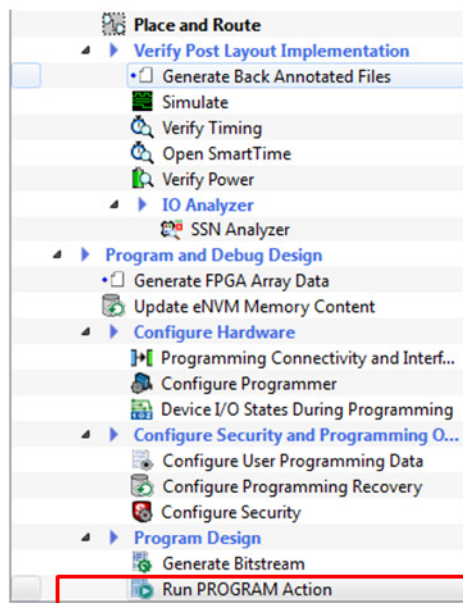
1. Connect the FlashPro4 programmer to the **J5** connector of the SmartFusion2 Security Evaluation Kit.
2. Connect the jumpers on the SmartFusion2 Security Evaluation Kit board as per Table 2.
For more information on jumper locations, refer to [Appendix: SmartFusion2 Security Evaluation Kit Board Jumper 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: 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 20.

Figure 20 • Run Program Action



2.7 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:

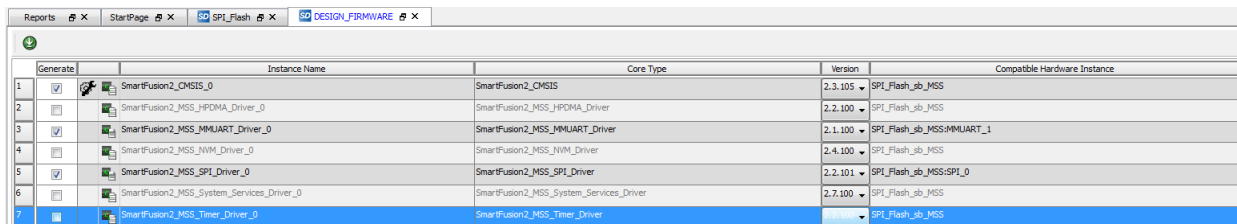
- CMSIS
- MMUART
- SPI

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 21.

Note: Select the latest version of the drivers.

Figure 21 • Configuring Firmware

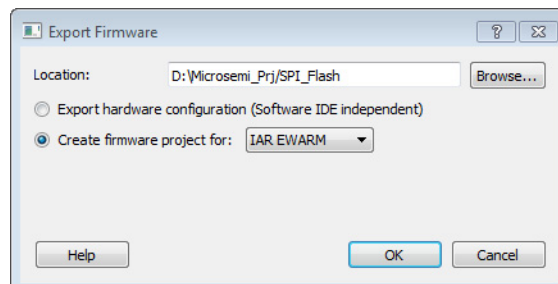


	Generate	Instance Name	Core Type	Version	Compatible Hardware Instance
1	☒	SmartFusion2_CMSIS_0	SmartFusion2_CMSIS	2.3.105	SPI_Flash_ab_MSS
2	☒	SmartFusion2_MSS_HPDMA_Driver_0	SmartFusion2_MSS_HPDMA_Driver	2.2.100	SPI_Flash_ab_MSS
3	☒	SmartFusion2_MSS_MMUART_Driver_0	SmartFusion2_MSS_MMUART_Driver	2.1.100	SPI_Flash_ab_MSS:MMUART_1
4	☒	SmartFusion2_MSS_NVM_Driver_0	SmartFusion2_MSS_NVM_Driver	2.4.100	SPI_Flash_ab_MSS
5	☒	SmartFusion2_MSS_SPI_Driver_0	SmartFusion2_MSS_SPI_Driver	2.2.101	SPI_Flash_ab_MSS:SPI_0
6	☐	SmartFusion2_MSS_System_Services_Driver_0	SmartFusion2_MSS_System_Services_Driver	2.7.100	SPI_Flash_ab_MSS
7	☐	SmartFusion2_MSS_Timer_Driver_0	SmartFusion2_MSS_Timer_Driver	2.7.100	SPI_Flash_ab_MSS

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

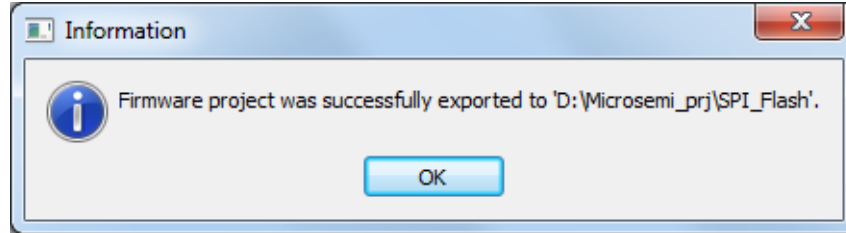
The **Export Firmware** dialog box is displayed, as shown in Figure 22.

Figure 22 • Export Firmware Dialog Box



4. In the **Export Firmware** dialog box:
 - Select **Create firmware project for**.
 - Select **IAR EWARM** from the drop down list.
5. Click **OK**. The successful firmware generation window is displayed as shown in [Figure 23](#).

Figure 23 • Firmware Successfully Exported Message



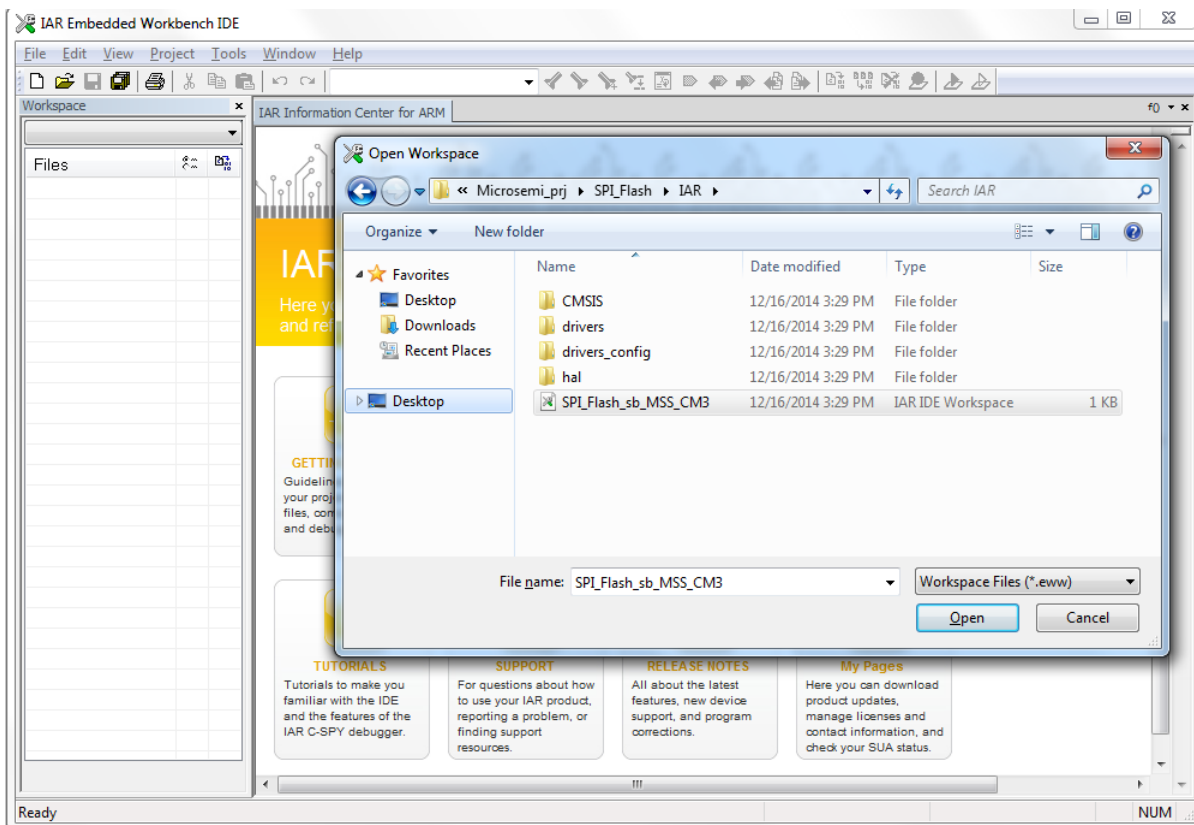
6. Click **OK**.
- The SmartFusion2 Security Evaluation Kit is ready for running and debugging the IAR Embedded Workbench application through J-Link Debugger.

2.8 Step 5: Building Software Application using IAR Embedded Workbench

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

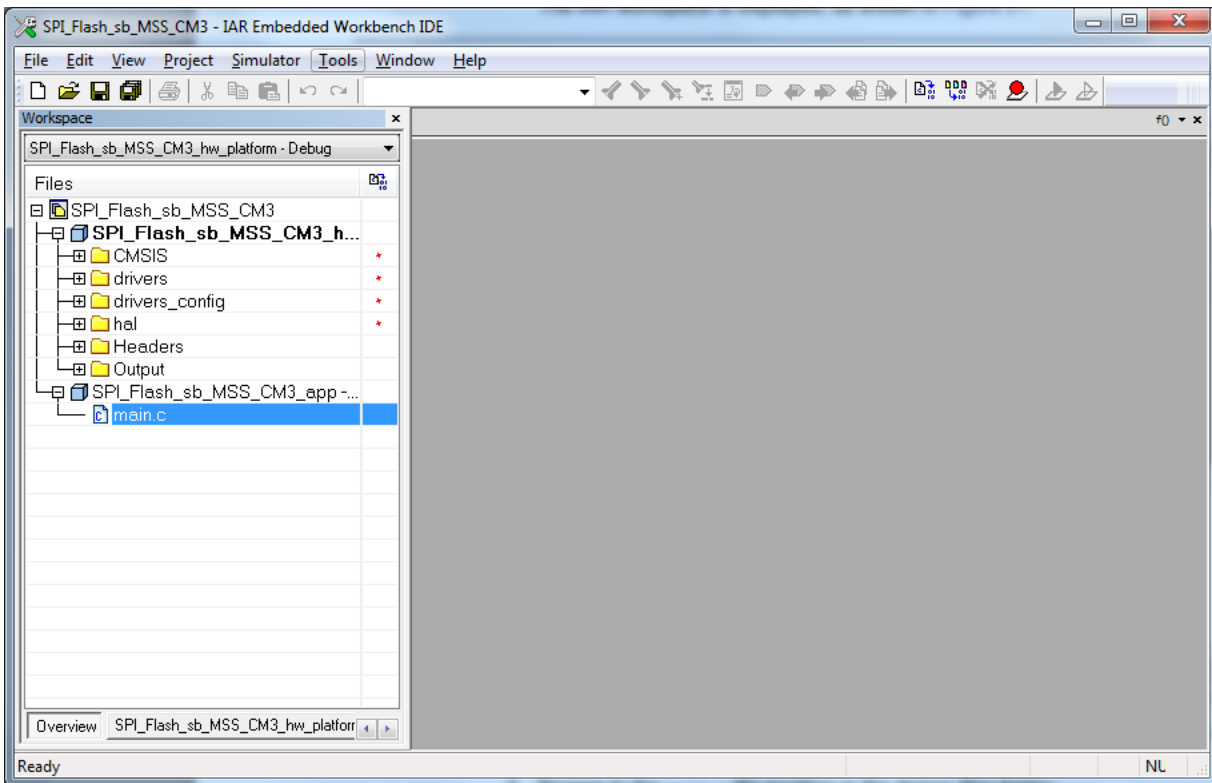
1. Connect the J-Link programmer to **J4 connector** of SmartFusion2 Security Evaluation Kit. Refer to "[Appendix: SmartFusion2 Security Evaluation Kit Board Jumper Locations](#)" on page 48 for information on the board setup for running and debugging the IAR software application.
Ensure that the SmartFusion2 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 24.

Figure 24 • Invoking IAR Embedded Workbench from Libero SoC Software



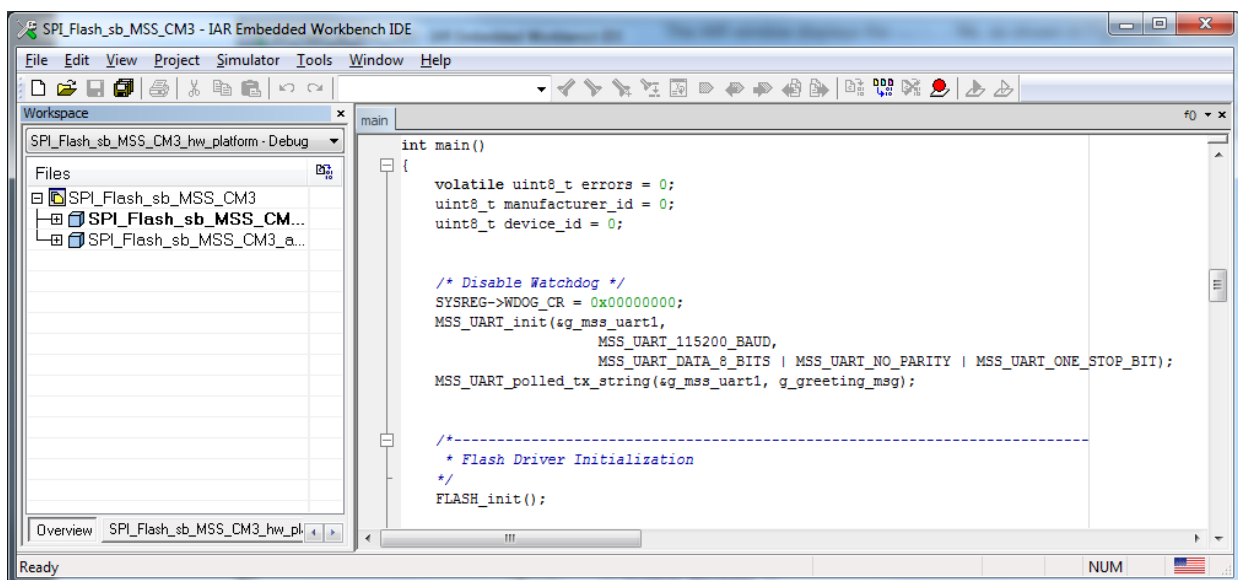
The IAR workspace is displayed, as shown in Figure 25.

Figure 25 • 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.
5. The IAR window displays the `main.c` file, as shown in Figure 26.

Figure 26 • IAR Workspace main.c file



6. 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`.
7. Copy the **winbondflash** folder to the drivers folder of SPI_Flash_sb_MSS_CM3_hw_platform project in the IAR workspace: `projectdirectory\IAR\drivers`.
8. 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 27.

Figure 27 • IAR Workspace Window - Add winbondflash SPI Driver Files

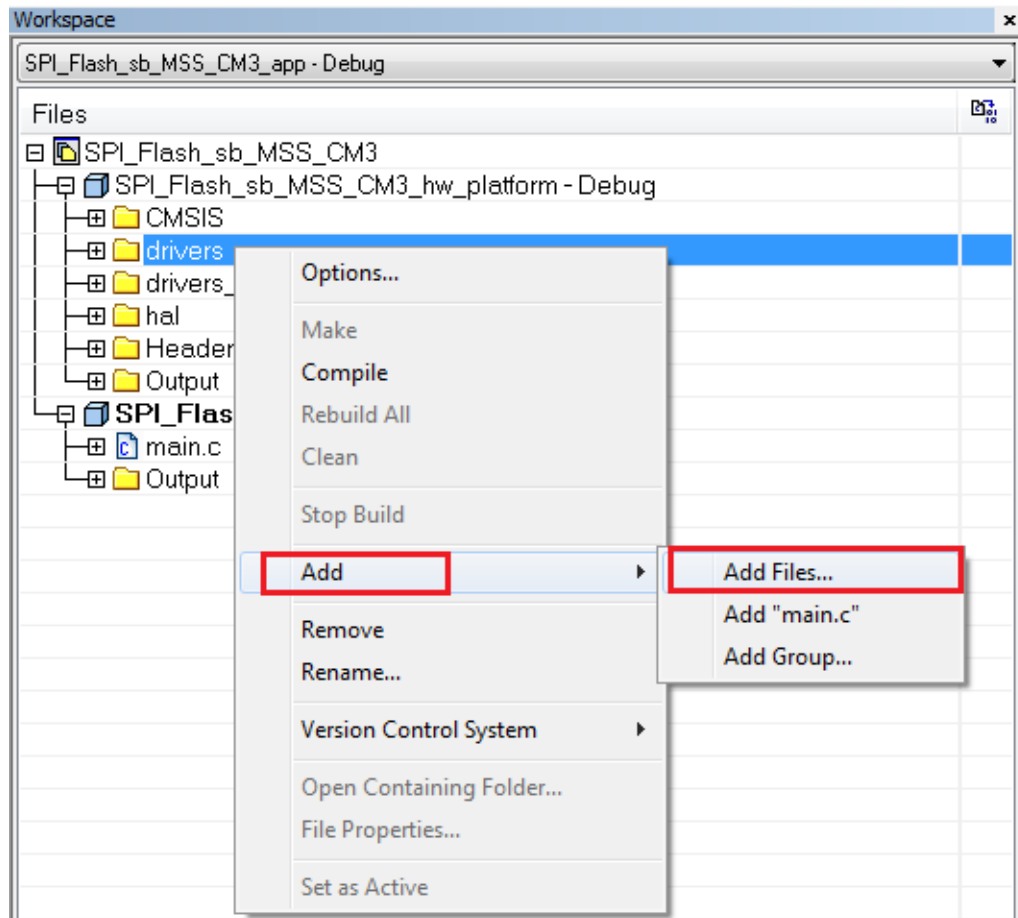
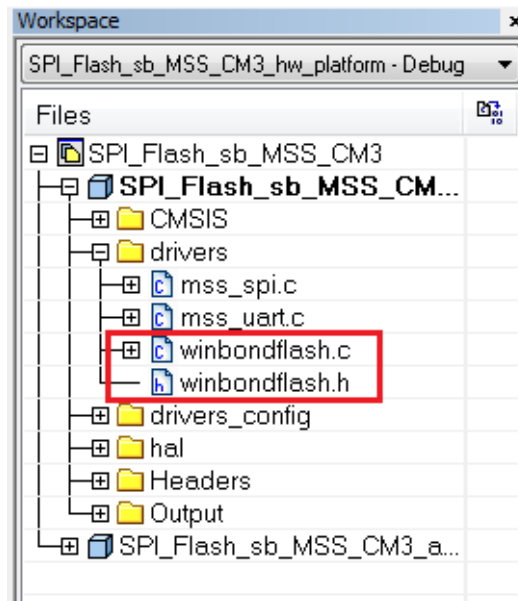


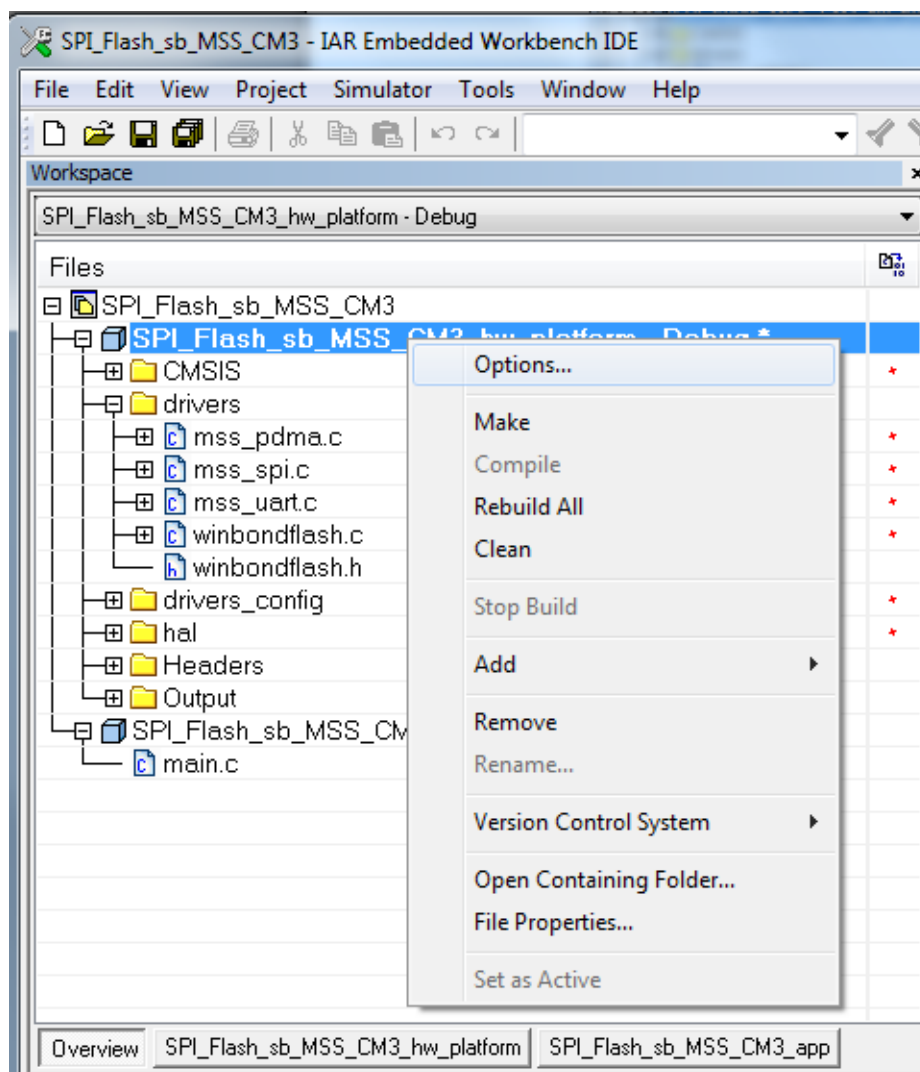
Figure 28 shows the IAR workspace window displaying winbondflash SPI Driver Files.

Figure 28 • IAR Workspace Window - Display winbondflash SPI Driver Files



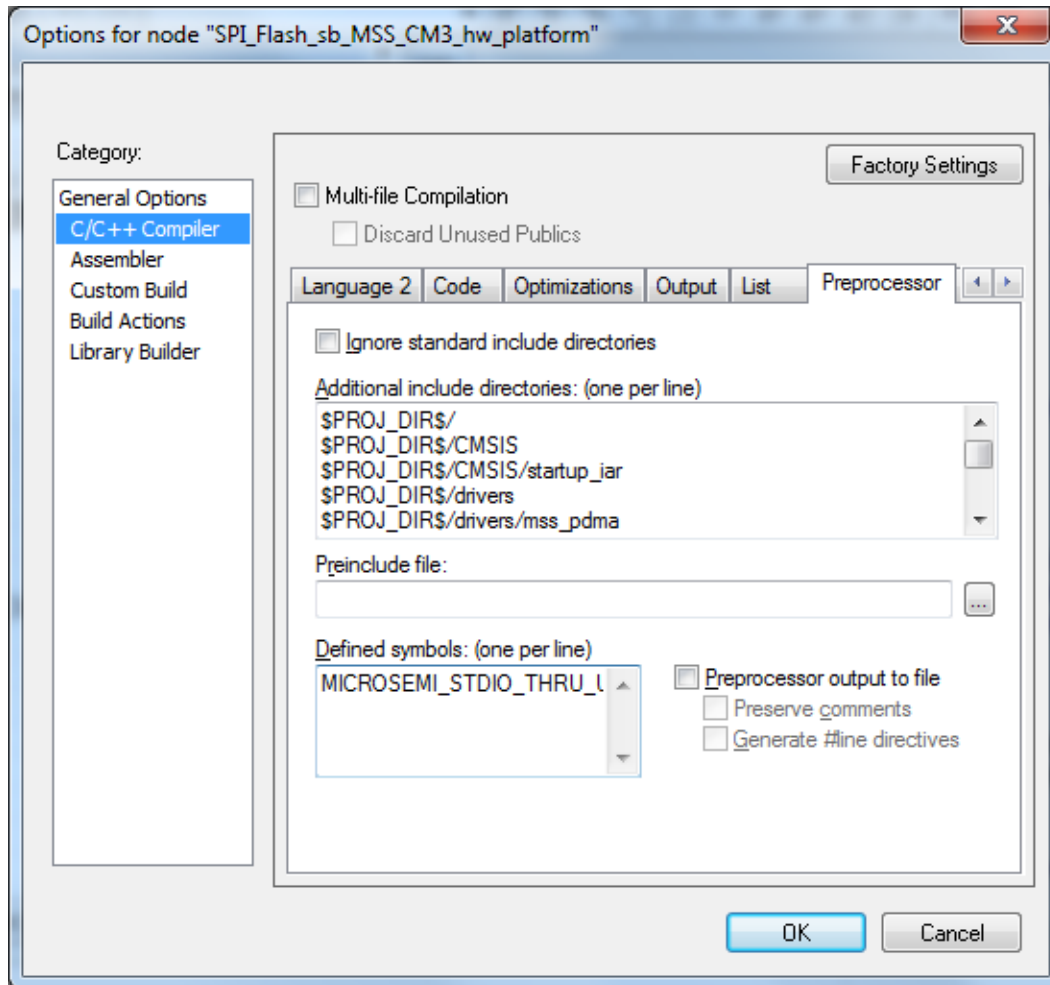
9. To configure the project, right-click the project name (SPI_Flash_sb_MSS_CM3_hw_platform) and click **Options**, as shown in Figure 29.

Figure 29 • IAR Workspace Window - Choose Options



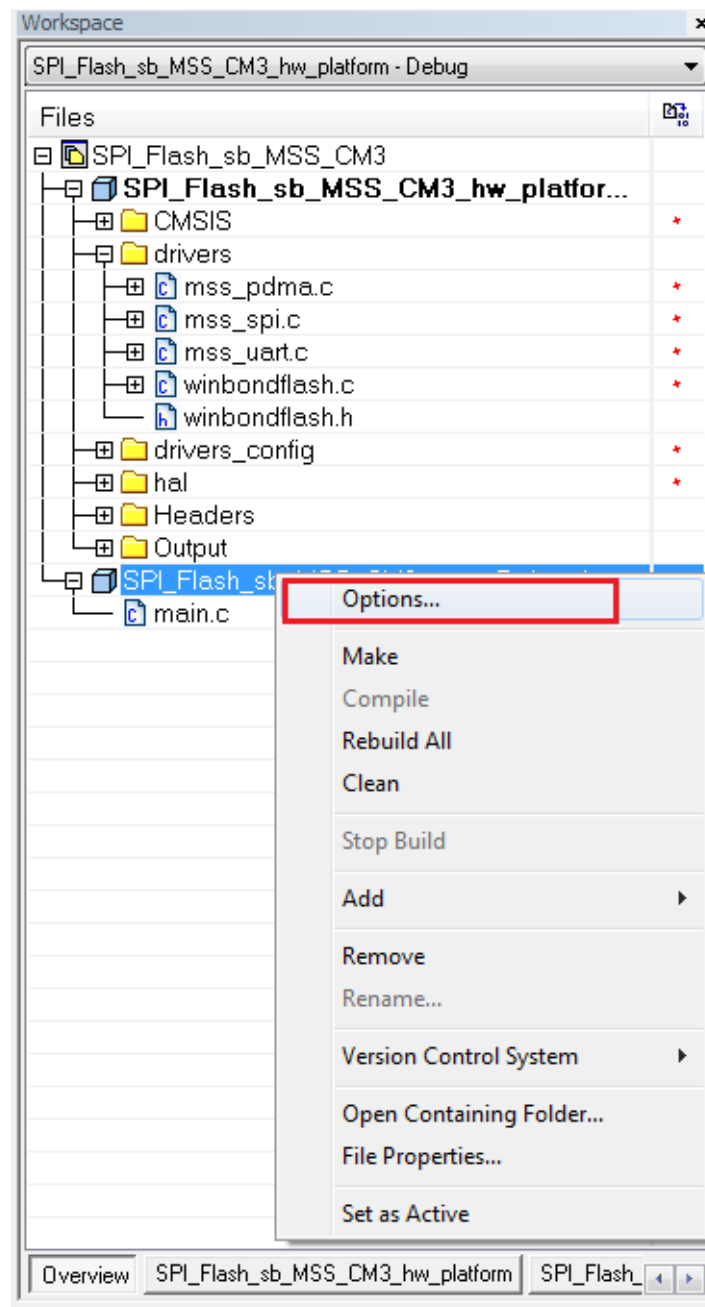
10. This tutorial uses `printf` statements to display memory read data. Redirection of the output of `printf()` to a UART is enabled by adding the **MICROSEMI_STDIO_THRU_UART** symbol.
11. In **Options** window, click **C/C++ Compiler**.
12. Click **Preprocessor** tab.
13. Under **Defined symbols** enter **MICROSEMI_STDIO_THRU_UART** and click **OK**, as shown in Figure 30.

Figure 30 • IAR Workspace Window - Adding Symbol



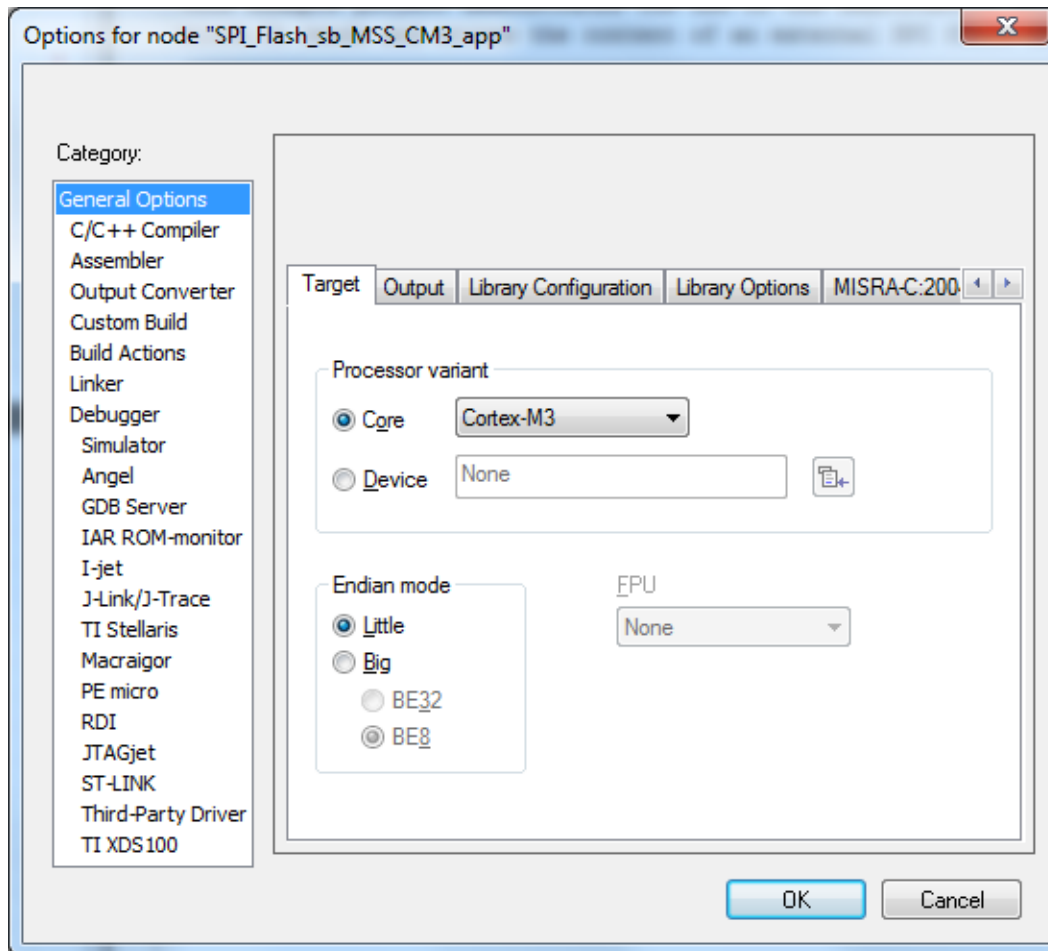
14. To configure the project, right-click the project name (SPI_Flash_sb_MSS_CM3_app), and click **Options**, as shown in Figure 31.

Figure 31 • IAR Workspace Window - Choose Options



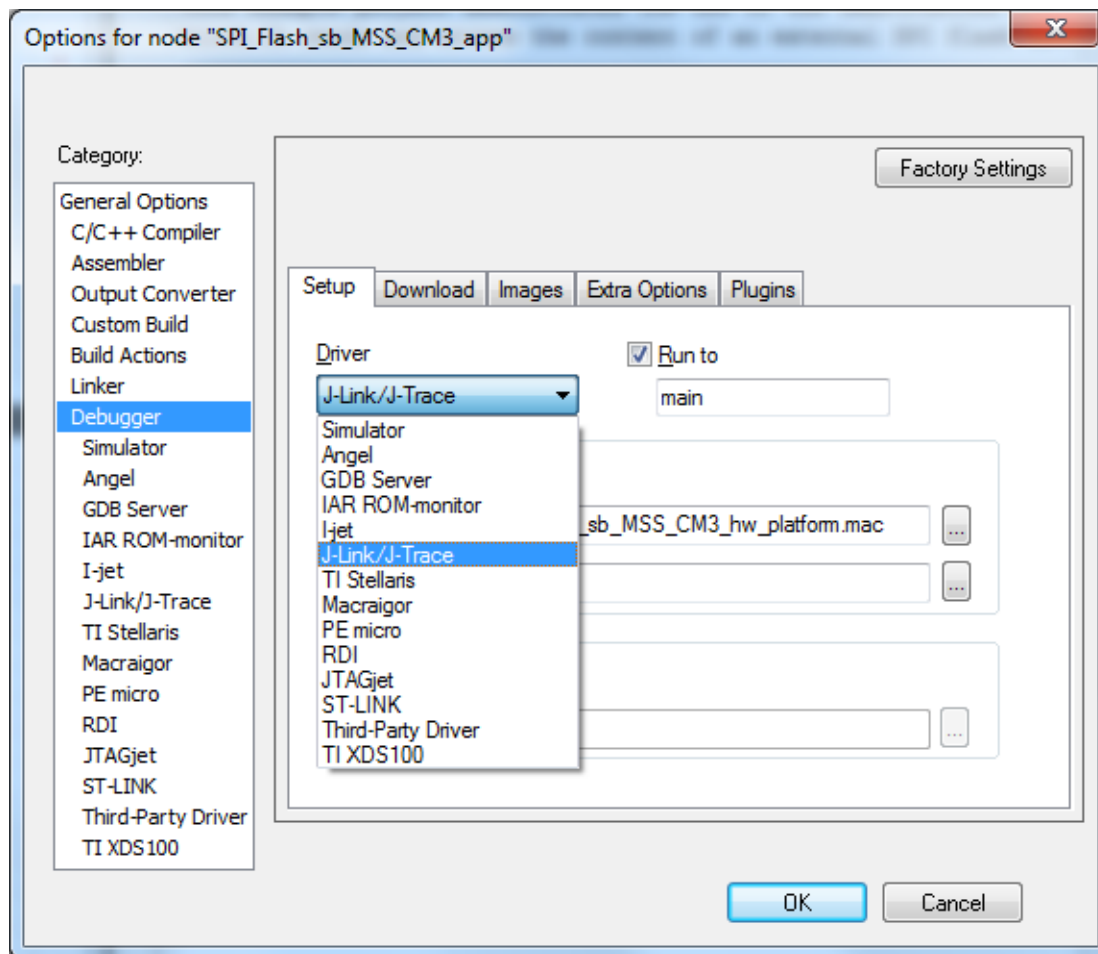
The Options for node SPI_Flash_sb_MSS_CM3_app window is displayed, as shown in Figure 32.

Figure 32 • IAR Node Options



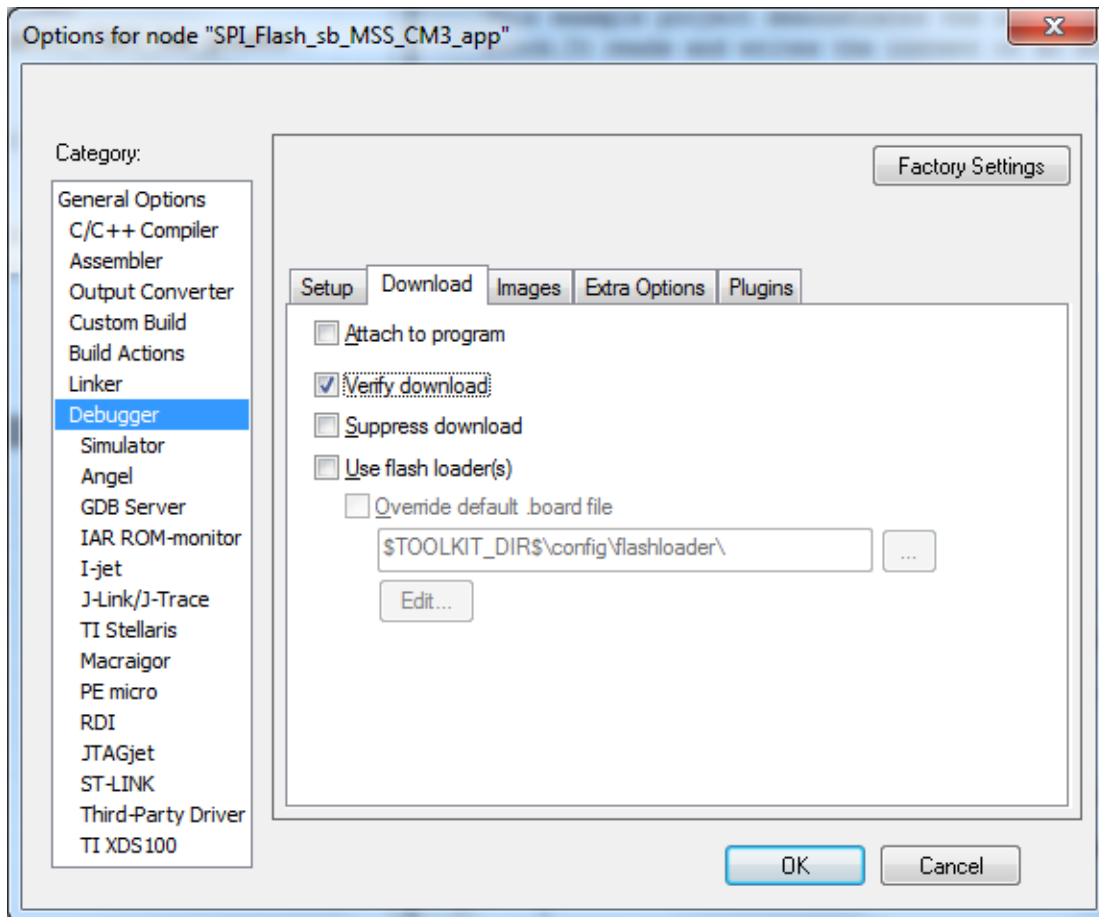
15. Click **Debugger**. Under the **Setup** tab, select **J-Link/J-Trace** from the driver the drop-down list, refer to Figure 33.

Figure 33 • IAR Debugger Options - Selecting Driver



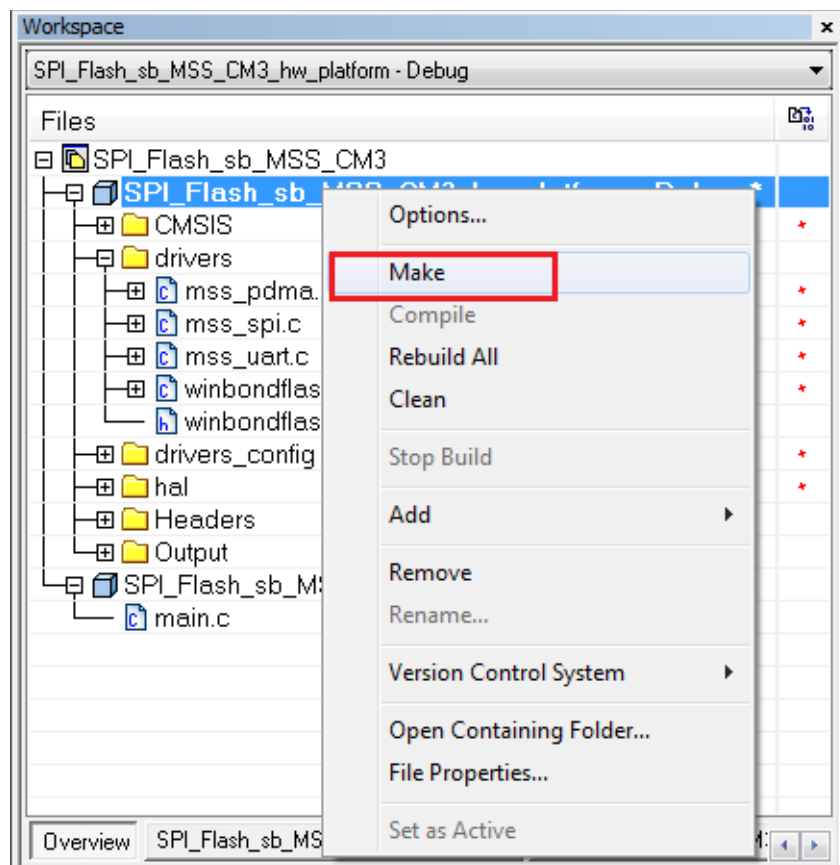
16. Click **Download** tab and select the **Verify download** check box, as shown in Figure 34.

Figure 34 • IAR Debugger Options - Download



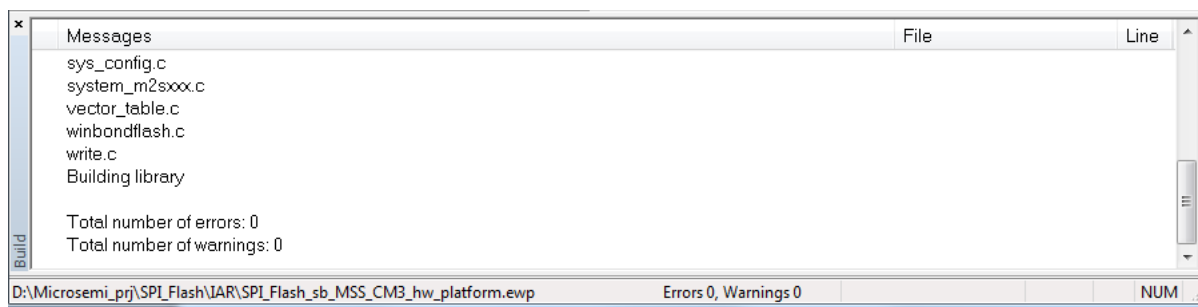
17. Click **OK** to close the **Options** window and build the project.
18. Right-click **SPI_Flash_sb_MSS_CM3_hw_platform - Debug** and select **Make**, as shown in [Figure 35](#) and [Figure 36](#).

Figure 35 • IAR Workspace - Hardware Platform Code Compilation using Make



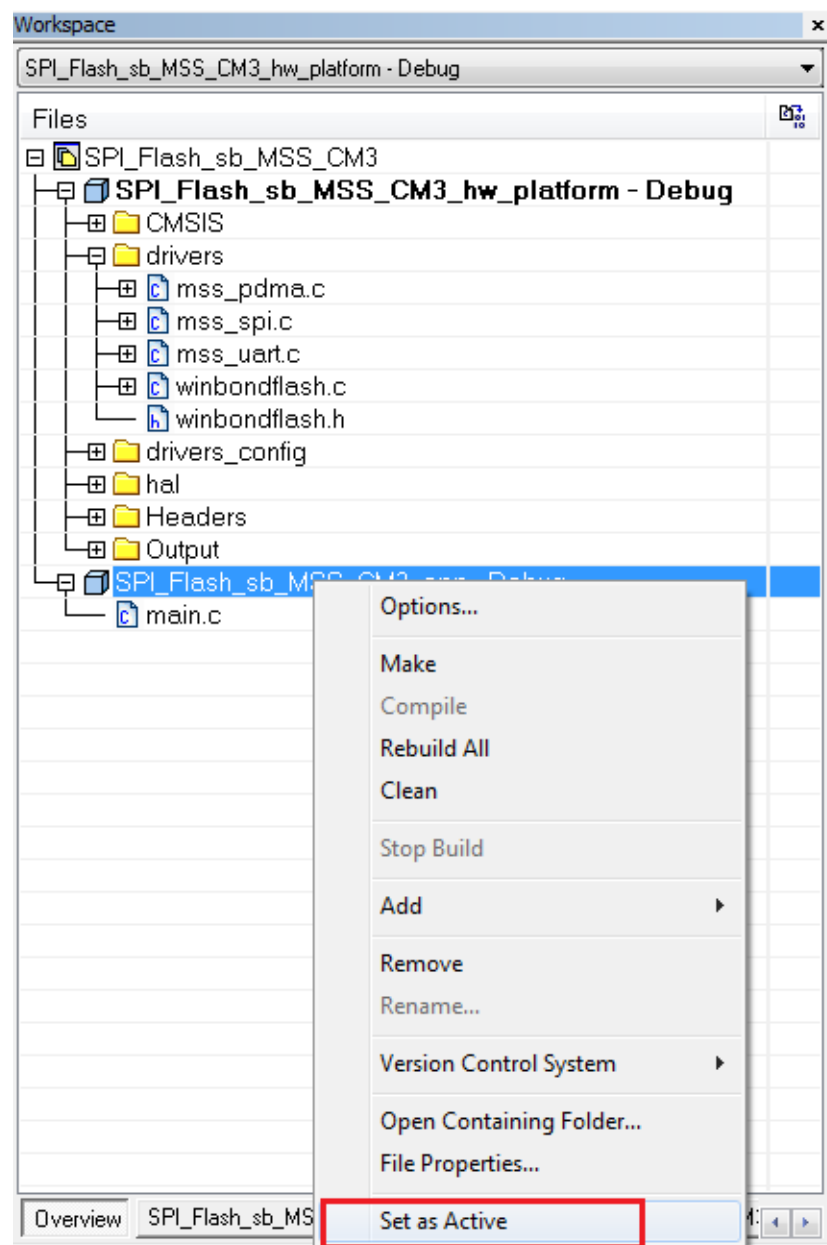
Successful Hardware Platform Code Compilation page is displayed as shown in [Figure 36](#).

Figure 36 • IAR Workspace - Successful Hardware Platform Code Compilation using Make



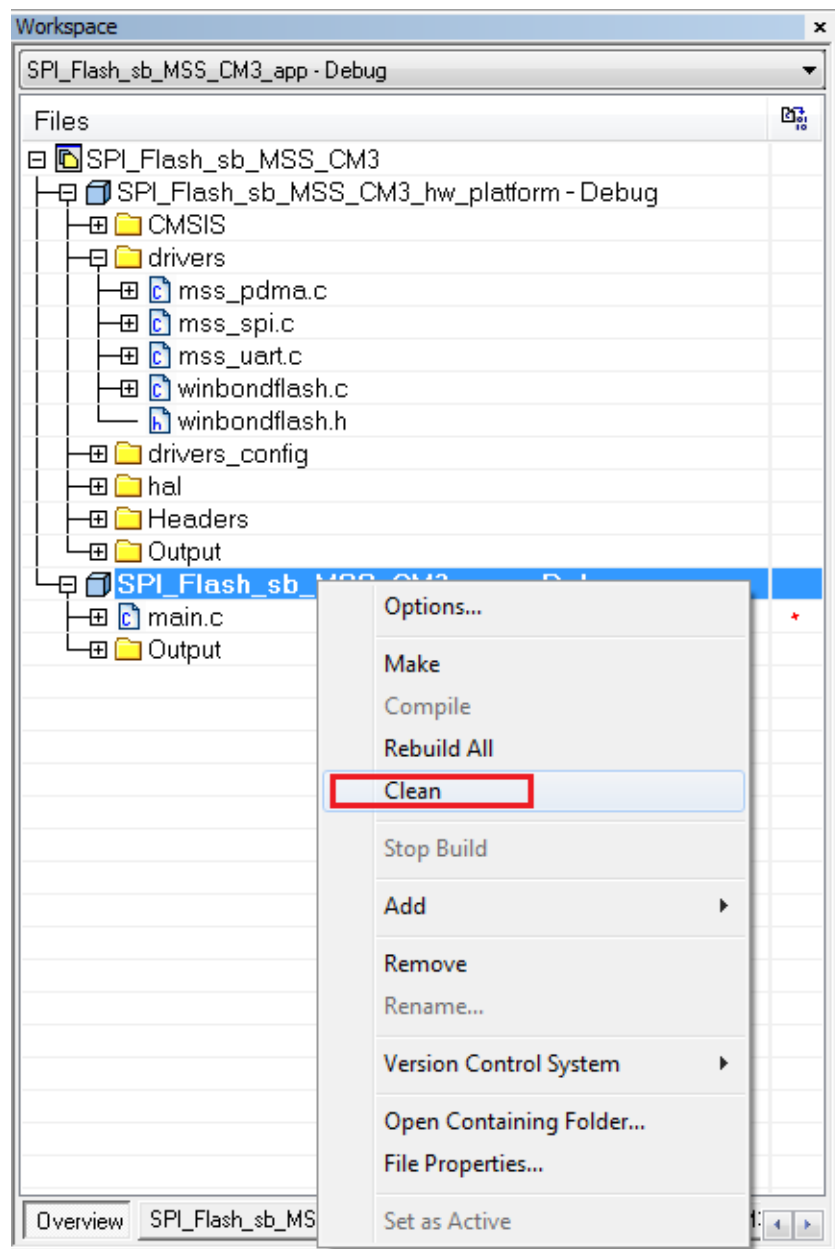
19. Right-click **SPI_Flash_sb_MSS_CM3_app - Debug** project name and select **Set as Active**, as shown in Figure 37.

Figure 37 • IAR Workspace - SPI_Flash_sb_MSS_CM3_app Set as Active



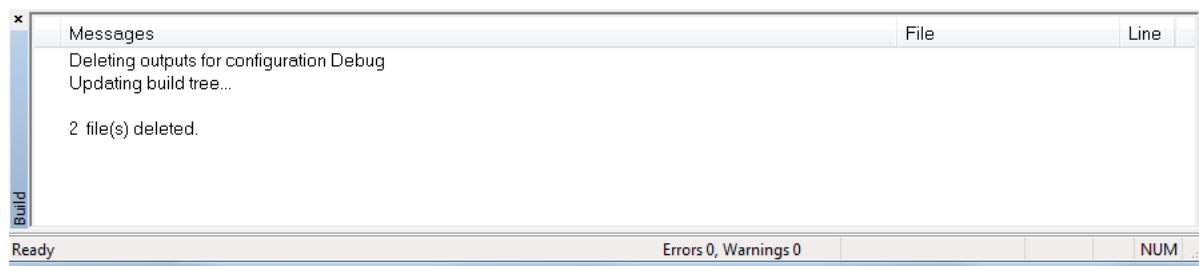
20. Right-click **SPI_Flash_sb_MSS_CM3_app - Debug** project name and select **Clean**, as shown in Figure 38.

Figure 38 • IAR Workspace - Execute Clean on SPI_Flash_sb_MSS_CM3_app Project



21. After cleaning the project, the **Messages** log section shows that some files are deleted, as shown in [Figure 39](#).

Figure 39 • IAR Workspace - Deleted Files



22. Right-click **SPI_Flash_sb_MSS_CM3_app - Debug** project name and click **Rebuild All**, as shown in [Figure 40](#).

Figure 40 • IAR Workspace - Select Rebuild All

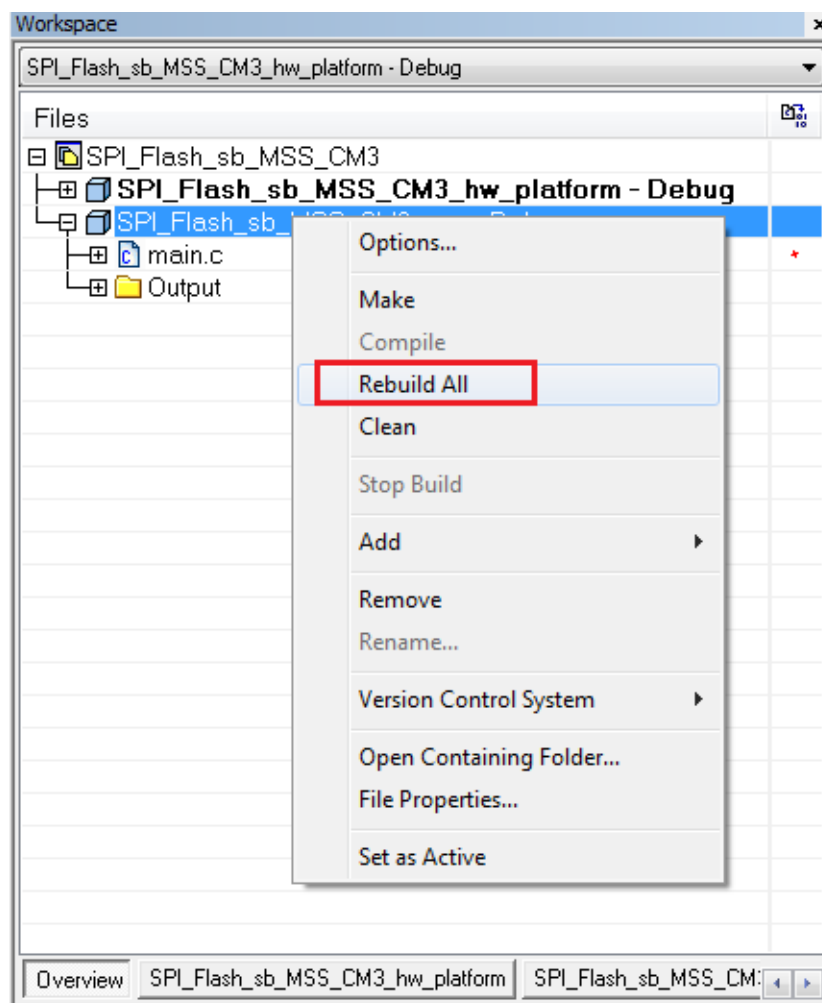
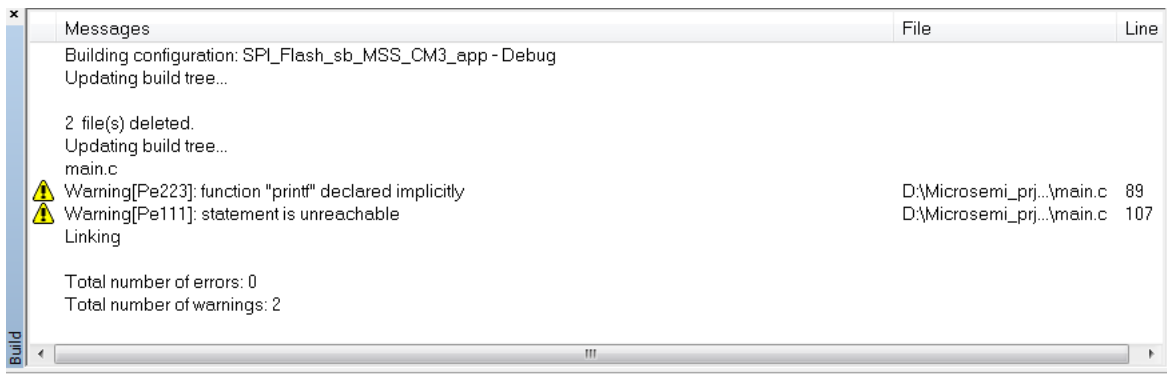


Figure 41 • IAR Workspace - Rebuild All

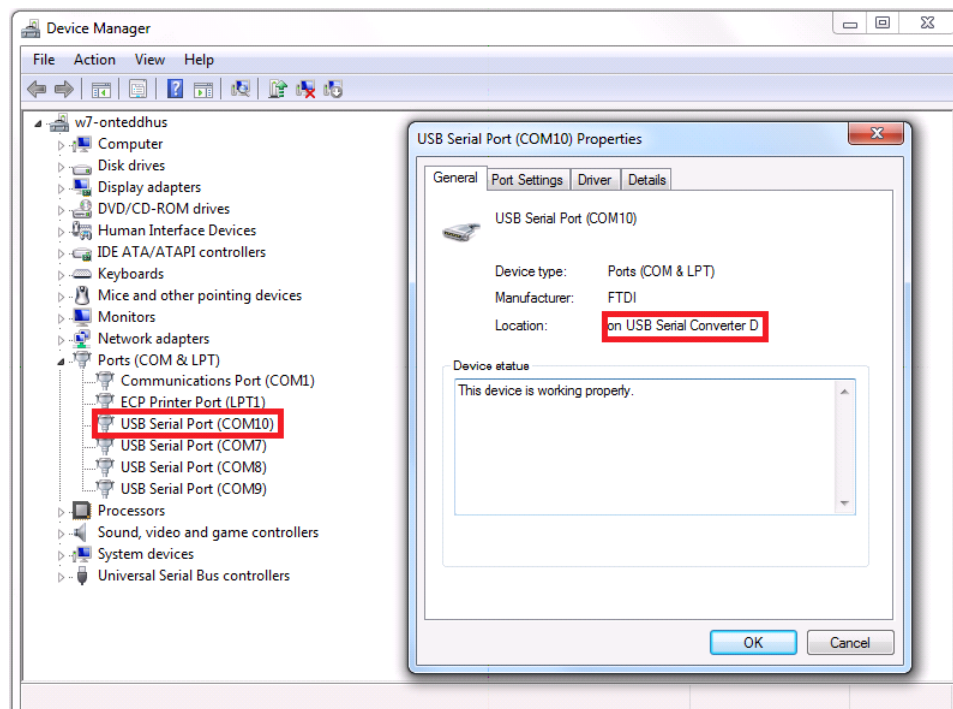


2.9 Step 6: Configuring Serial Terminal Emulation Program

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 42 shows an example **Device Manager** window.

Figure 42 • 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:

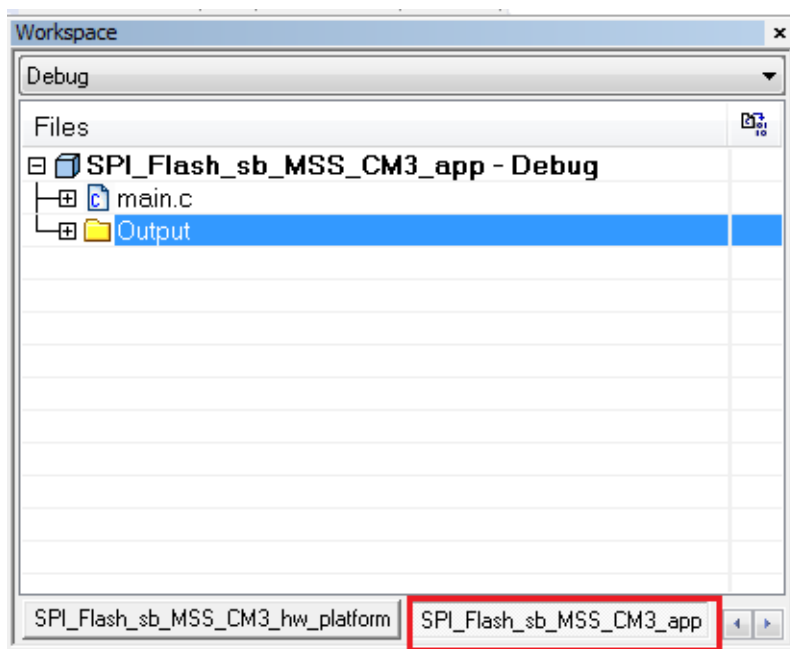
- 115200 baud rate
- 8 data bits
- 1 stop bit
- No parity
- No flow control

2.10 Step 7: Debugging the Application Project using IAR Workbench

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

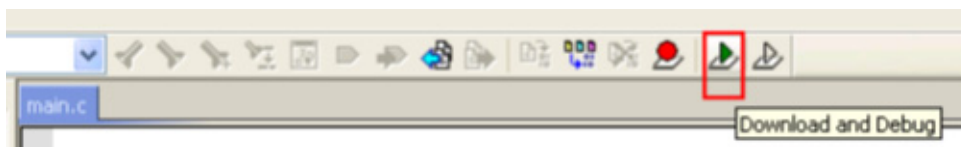
1. Switch to **SPI_Flash_sb_MSS_CM3_app - Debug** tab from Overview tab as shown in [Figure 43](#).

Figure 43 • Debug Window



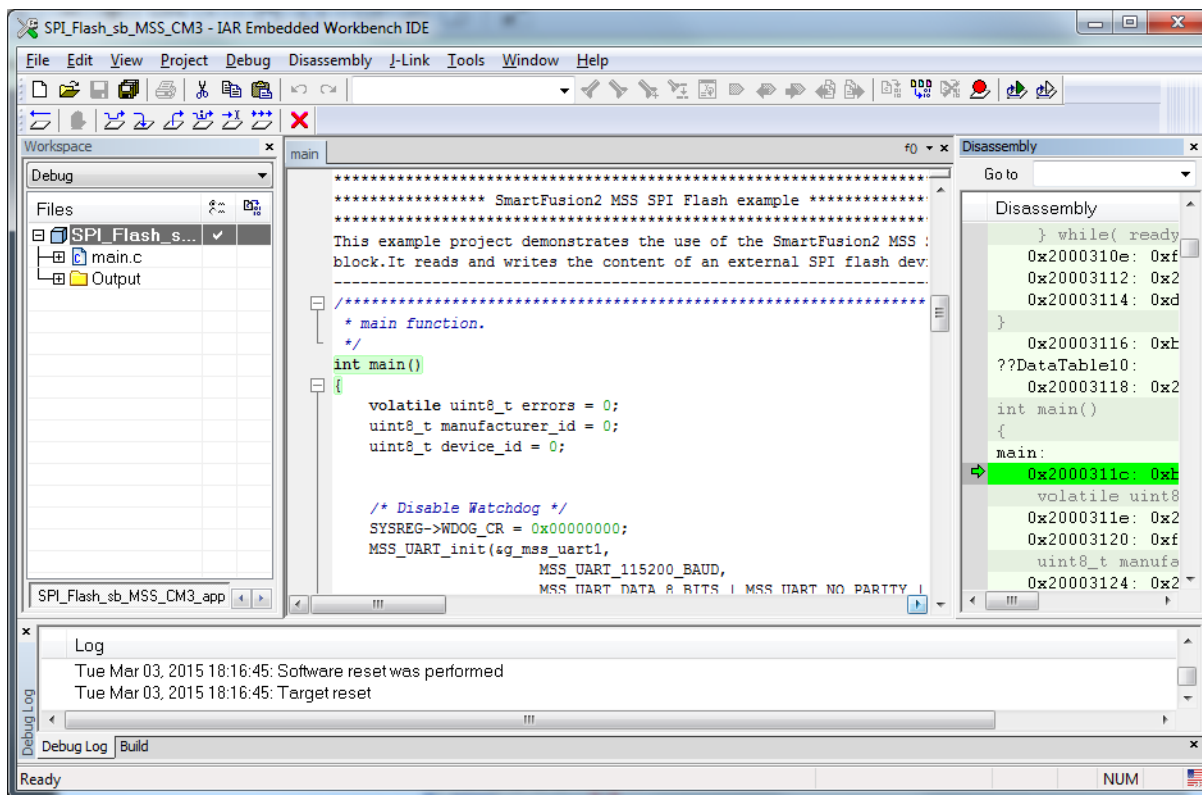
2. In the IAR Workbench, click **Download and Debug** as shown in [Figure 44](#).

Figure 44 • IAR Workbench - Download and Debug Option



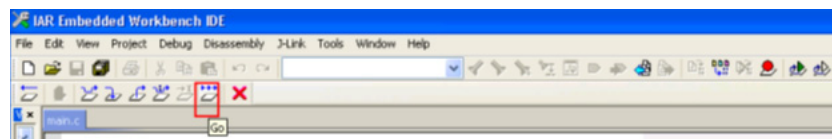
IAR Debugger Perspective window is opened, as shown in Figure 45.

Figure 45 • IAR Workbench - Debugger Perspective



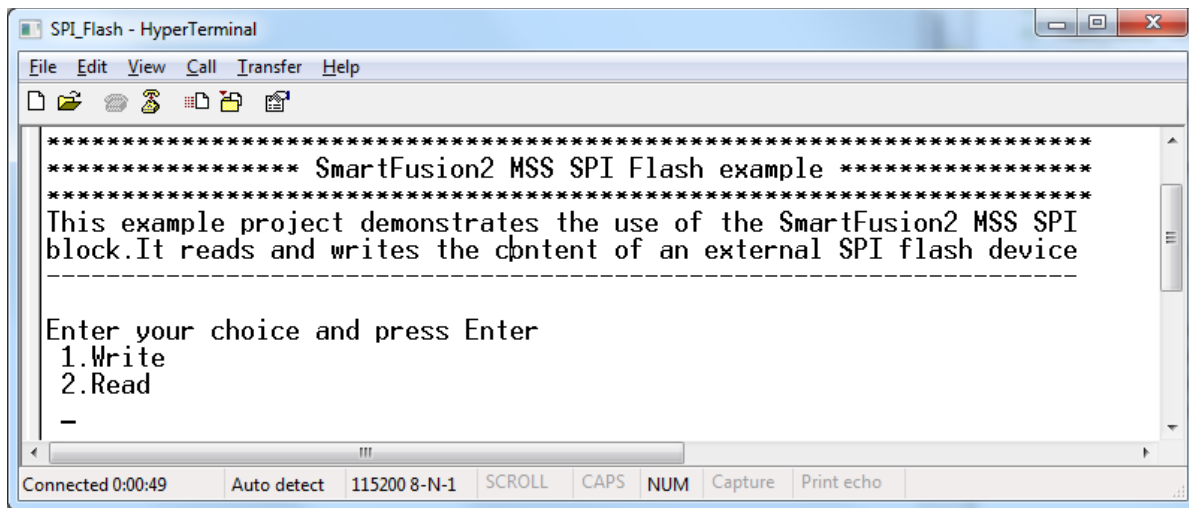
3. Click **Go** on IAR workbench to run the application, as shown in Figure 46.

Figure 46 • IAR Workbench - Go Option



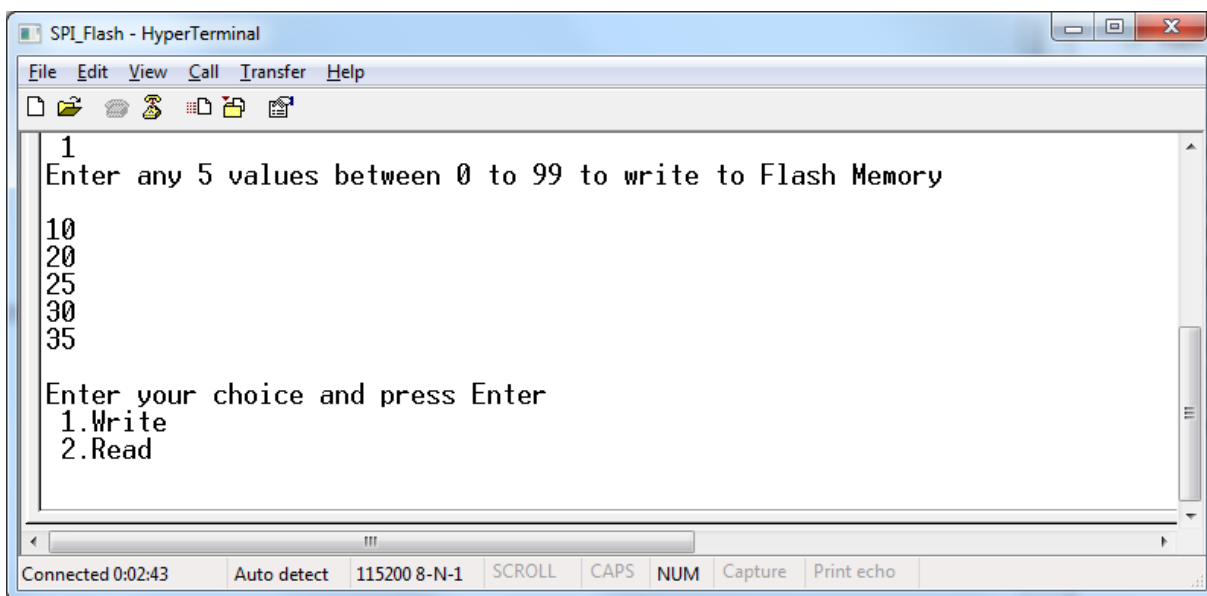
4. On successful operation, the HyperTerminal window displays a message, as shown in Figure 47.

Figure 47 • HyperTerminal Window



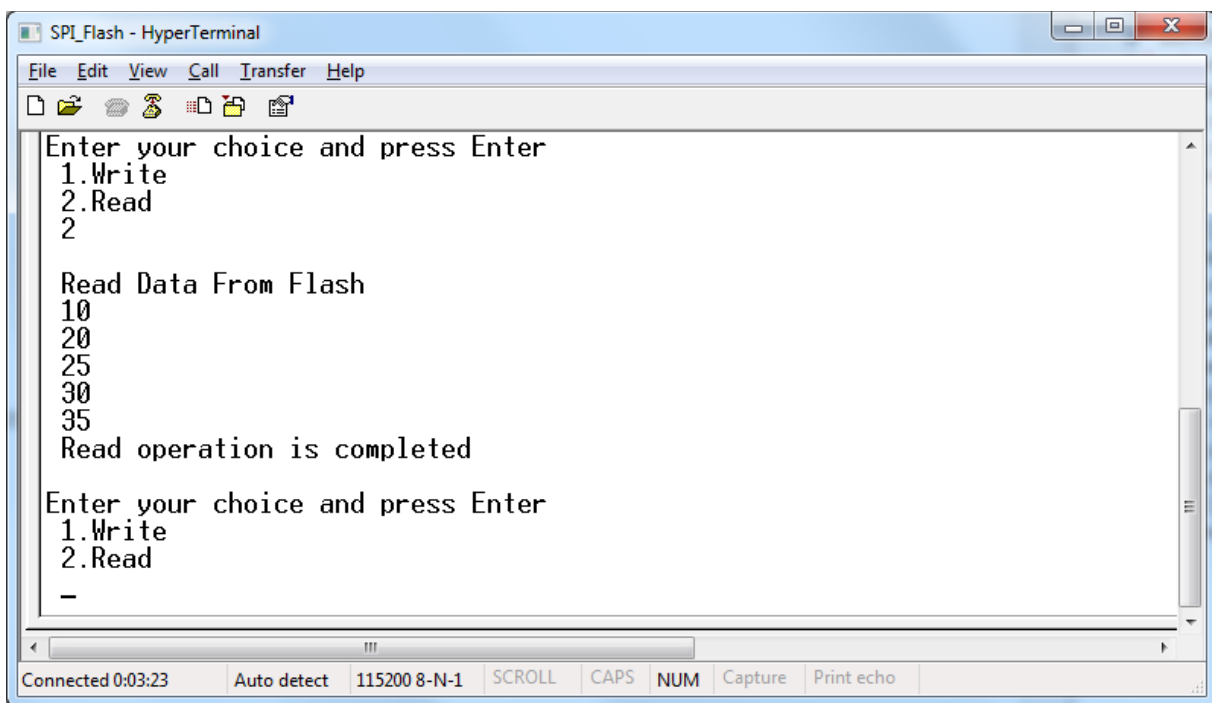
5. Select option 1 and enter values to write to the SPI Flash Memory, as shown in Figure 48.

Figure 48 • HyperTerminal Window - Option 1



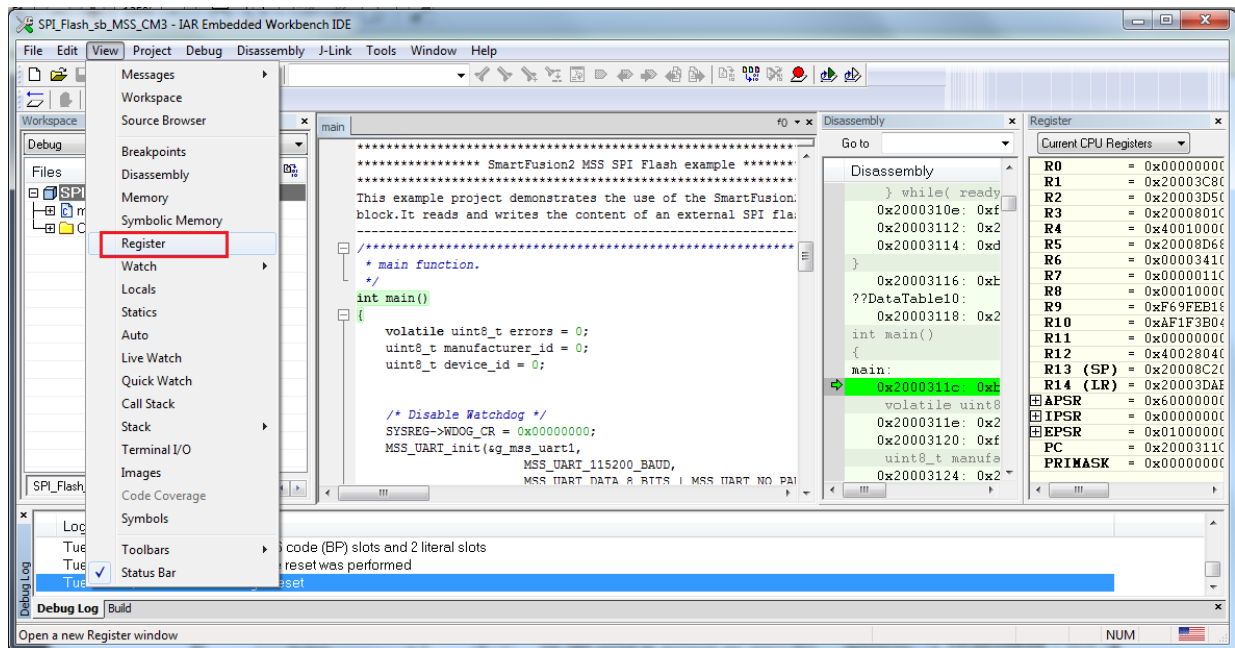
6. Select option 2 to read data from SPI Flash Memory, as shown in Figure 49.

Figure 49 • HyperTerminal Window - Option 2



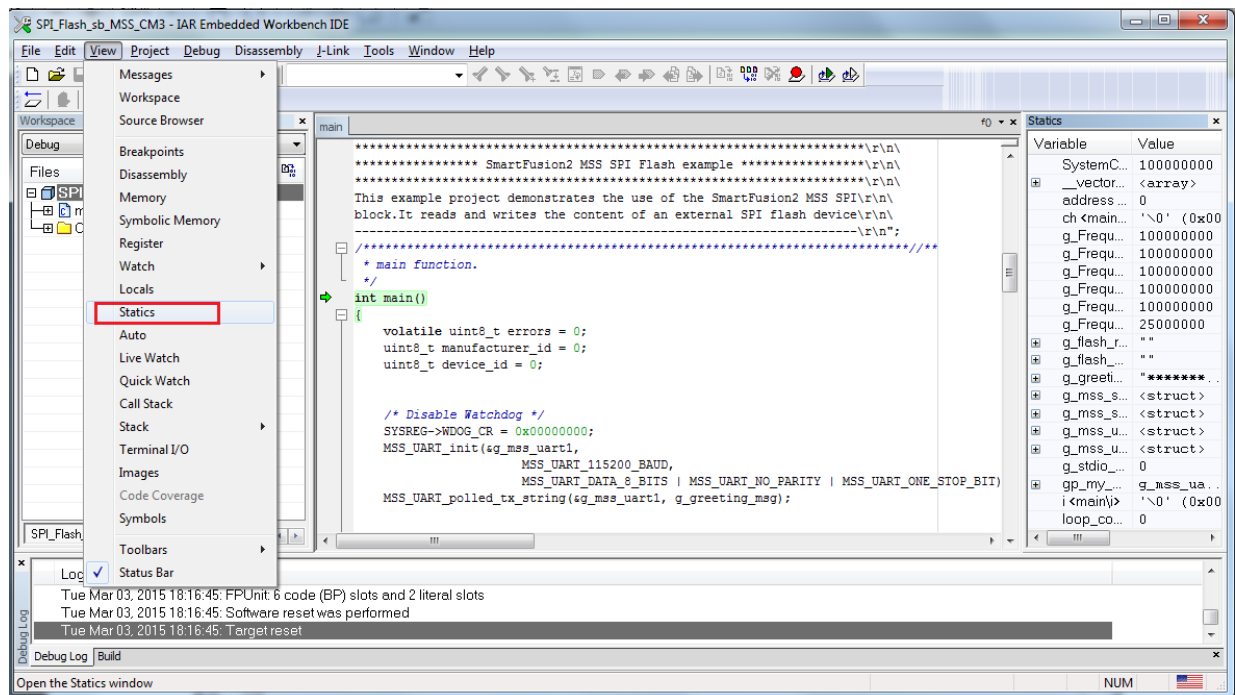
7. Click **View > Register** to view the values of the ARM® Cortex®-M3 processor internal registers, as shown in Figure 50.

Figure 50 • Values of Cortex-M3 Internal Registers



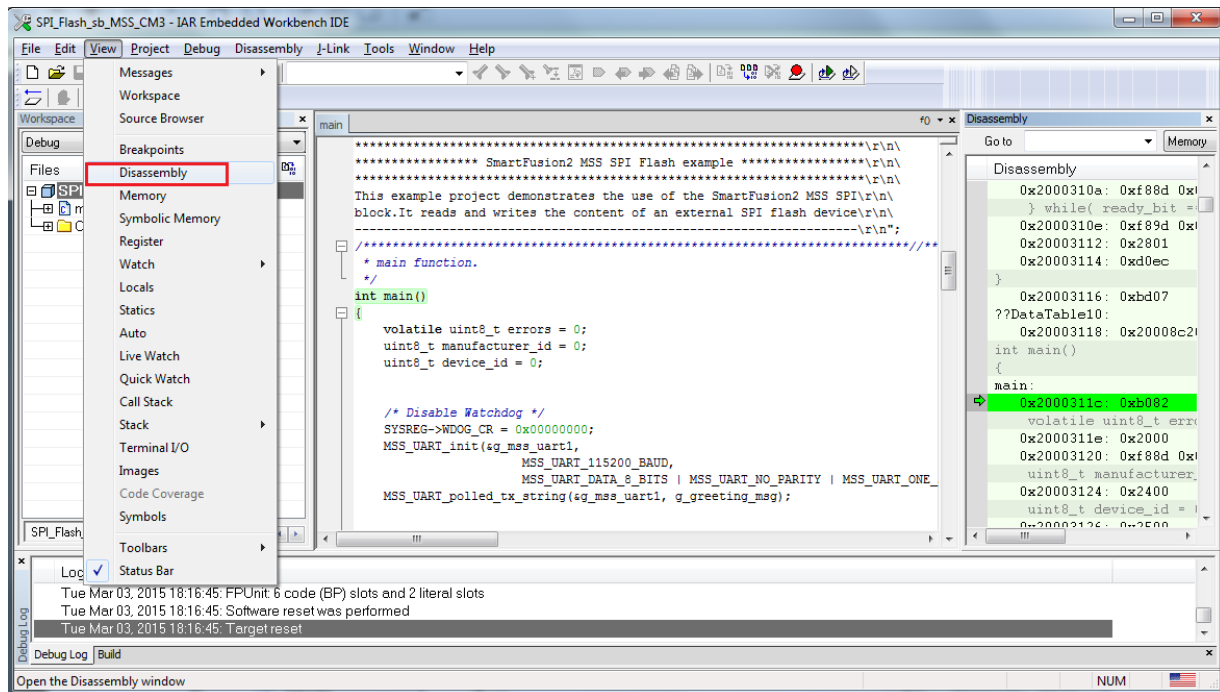
8. Click **View > Statics** to view the values of variables in the source code, as shown in Figure 51.

Figure 51 • Values of Source Code Variables



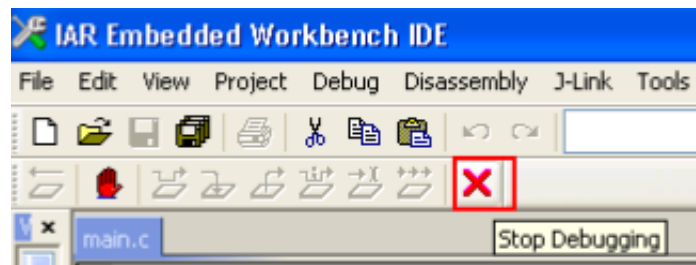
9. Click **View > Disassembly** to view the values of variables in the source code, as shown in Figure 52.

Figure 52 • Assembly Level Instructions



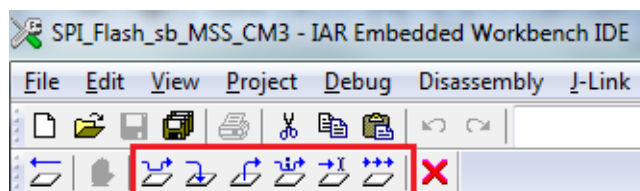
10. When debug process is finished, terminate execution of the code by choosing **Debug > Stop Debugging**, as shown in Figure 53.

Figure 53 • 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 54.

Figure 54 • 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 54. 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**.

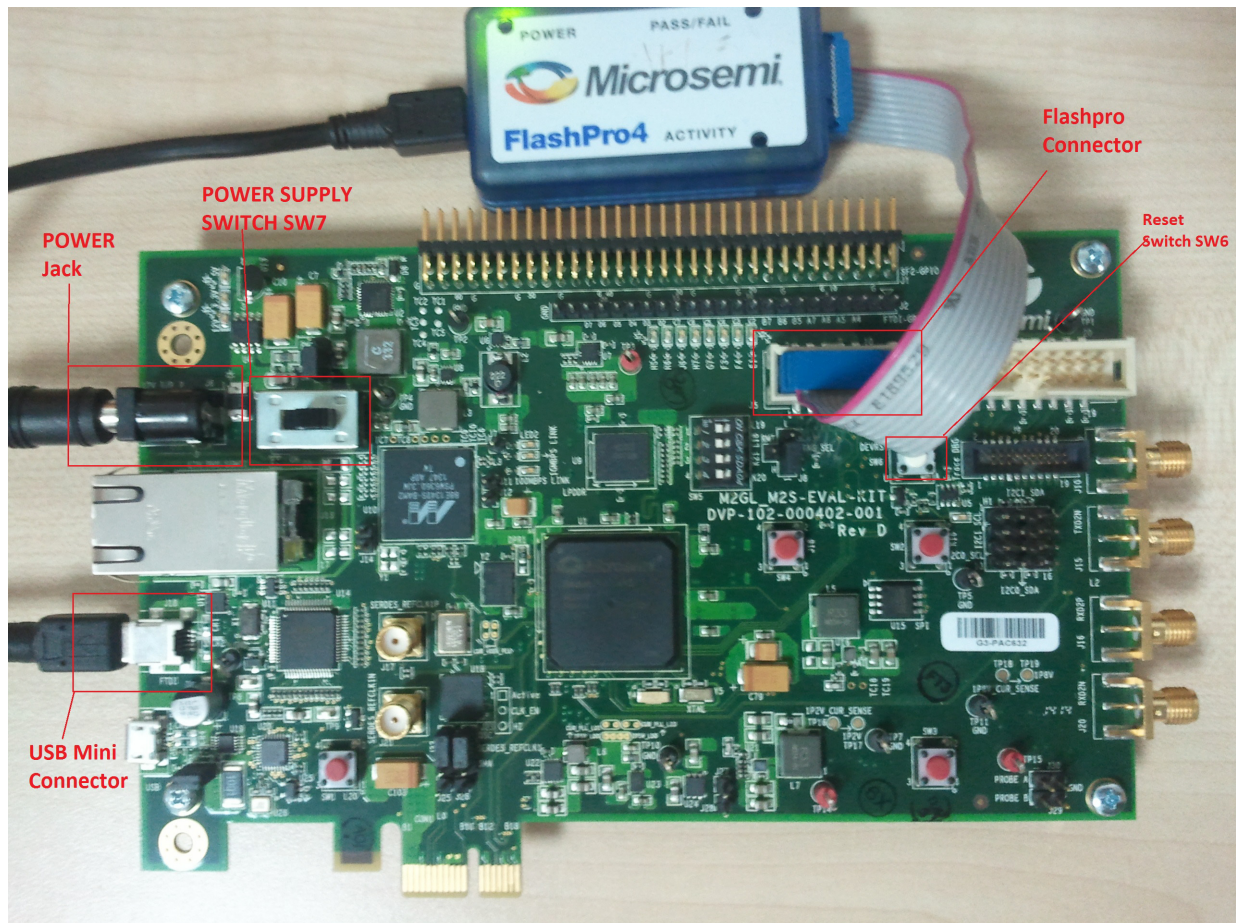
2.11 Conclusion

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.

3 Appendix: Board Setup for Programming the Tutorial

Figure 55 on page 46 shows the board setup for programming the tutorial on the SmartFusion2 Security Evaluation Kit board.

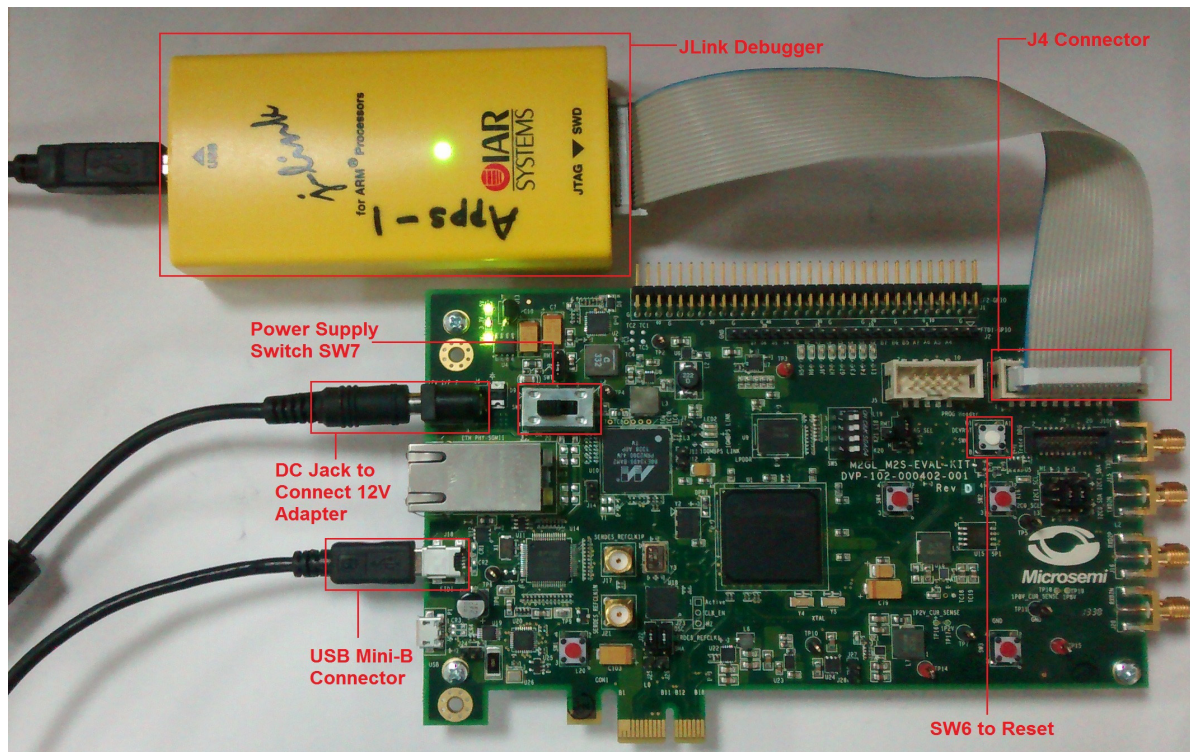
Figure 55 • SmartFusion2 Security Evaluation Kit Setup



4 Appendix: Board Setup for Running the IAR Tutorial

Figure 56 on page 47 shows the board setup for running and debugging the tutorial on the SmartFusion2 Security Evaluation Kit board.

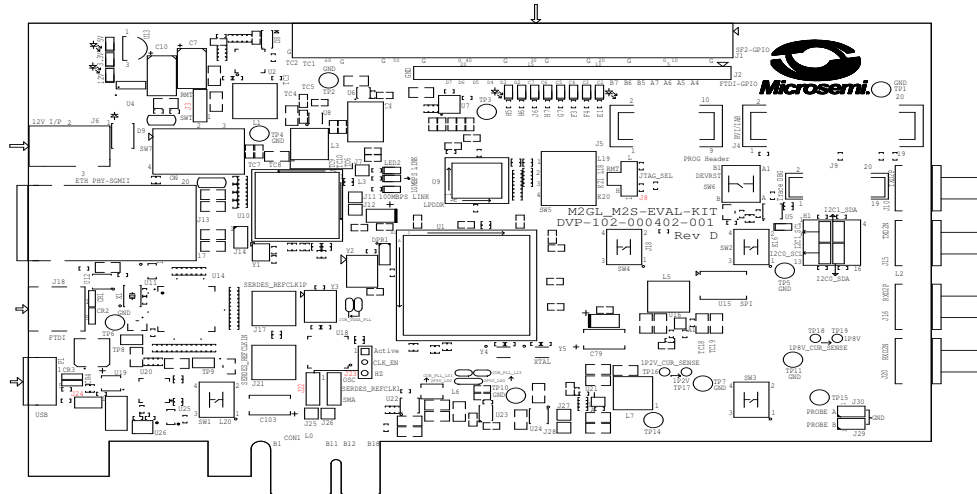
Figure 56 • SmartFusion2 Security Evaluation Kit J-Link Programmer Connection



5 Appendix: SmartFusion2 Security Evaluation Kit Board Jumper Locations

Figure 57 on page 48 shows the jumper locations on the SmartFusion2 Security Evaluation Kit board.

Figure 57 • SmartFusion2 Security Evaluation Kit Board Jumper Locations



Note:

- Jumpers highlighted in red (J22, J23, J24, J8, and J3) are set by default.
- The location of the jumpers in [Figure 57 on page 48](#) are searchable.

6 Revision History

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

Revision	Changes
Revision 5 (April 2016)	Updated the document for Libero SoC v11.7 software release (SAR 77812)
Revision 4 (October 2015)	Updated the document for Libero SoC v11.6 software release (SAR 72826)
Revision 3 (March 2015)	Updated the document for Libero SoC v11.5 software release (SAR 64188).
Revision 2 (November 2014)	Updated the document for Libero SoC v11.4 software release (SAR 61628).
Revision 1 (April 2014)	Initial release.

7 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.

7.1 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

7.2 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.

7.3 Technical Support

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

7.4 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>.

7.5 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.

7.5.1 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.

7.5.2 My Cases

Microsemi SoC Products Group customers may submit and track technical cases online by going to [My Cases](#).

7.5.3 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](#).

7.6 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

© 2016 Microsemi Corporation. All rights reserved. Microsemi and the Microsemi logo are trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.

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; Enterprise Storage and Communication solutions, 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 4,800 employees globally. Learn more at www.microsemi.com.

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.