SmartFusion2 ARM Cortex-M3 System Builder Lab Guide

Microsemi

Microsemi SoC Products Group, San Jose, CA 95134

Printed in the United States of America

Release: June, 2015

Table of Contents

Introduction

This tutorial demonstrates how to implement a basic SmartFusion2 Microcontroller Subsystem (MSS) configuration that includes the GPIO block, MMUART_0, MMUART_1 and the RTC using System Builder.

The SmartFusion2 GPIO[8] will be configured as an input and GPIO[1:0] will be configured as outputs. The outputs will drive LEDs on the SmartFusion2 target board; GPIO[8] will be connected to one of the switches on the target board.

After completing this tutorial you will be familiar with the following:

- Creating a Libero SoC project
- Using System Builder to configure the SmartFusion2 MSS and generate the design
- Generating sample firmware projects from the Firmware catalog
- Synthesizing the SmartFusion2 design with Synplify Pro ME
- Importing a PDC constraint file to pin assignments and running layout on the design
- Generating a programming file and programming the SmartFusion2 device on the Starter Kit board
- Using SoftConsole to download Cortex-M3 applications to the SmartFusion2 target board and debug the applications

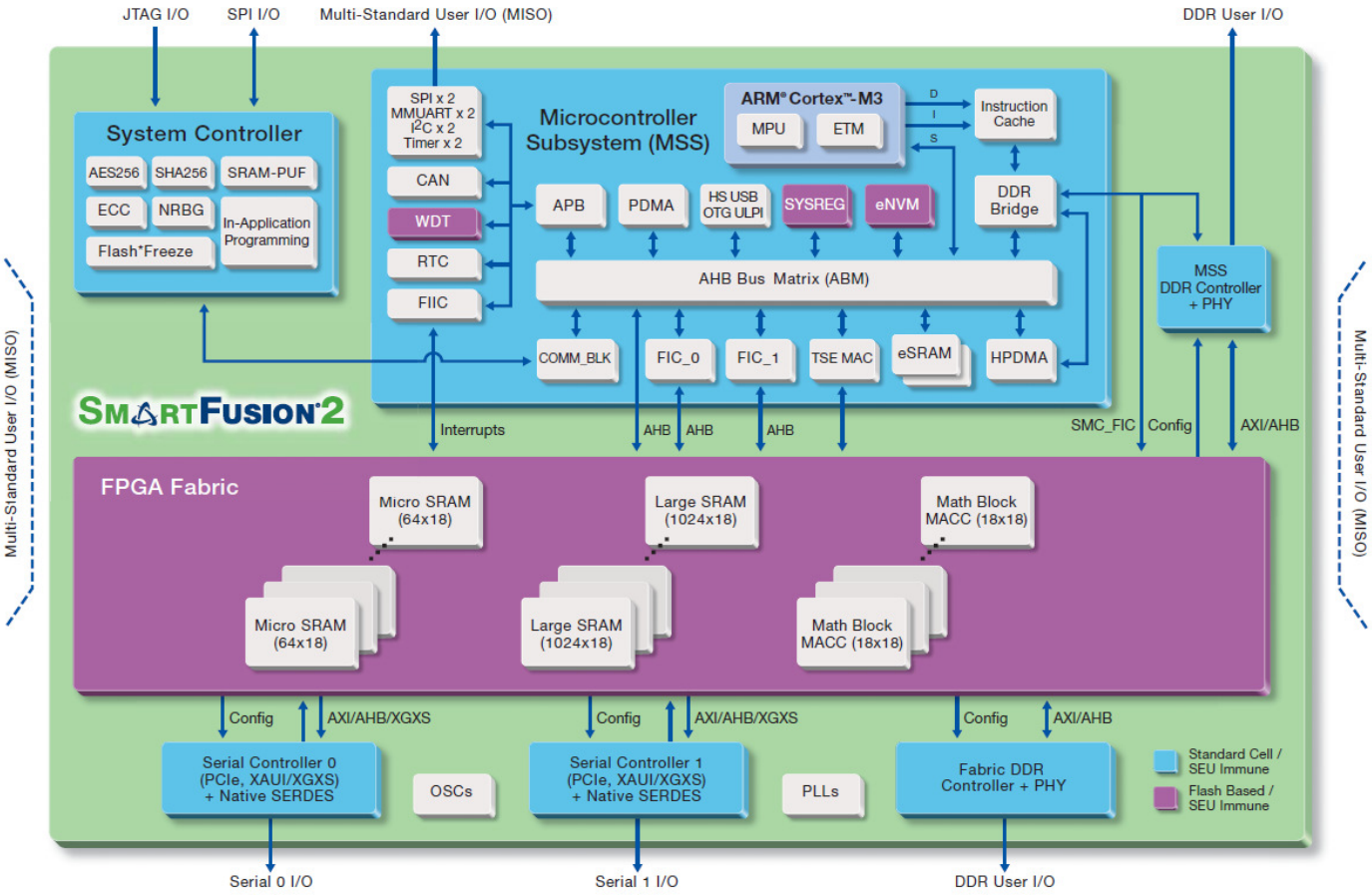


Figure 1 - SmartFusion2 Block Diagram

Components of SmartFusion2 Device Used

This tutorial uses the SmartFusion2 Cortex-M3, the GPIO block, MMUART_1 and MMUART_0.

Tutorial Requirements

Software Requirements

- Microsemi Libero SoC v11.5
- ModelSim 10.3c
- Synplify Pro ME I-2014.03M-SP1
- FlashPro v11.5
- SoftConsole v3.4 SP1 (Build id: 3.4.0.5-M20130329-1700)
- HyperTerminal or similar software, normally under **Start > Programs > Accessories > Communications > HyperTerminal**.

Note: HyperTerminal is no longer standard on newer windows installations. For alternatives to HyperTerminal, see http://helpdeskgeek.com/windows-7/windows-7-hyperterminal/

USB Drivers for the FT232R USB to UART IC:

http://www.microsemi.com/document-portal/doc_download/131593-usb-uart-driver-files

SmartFusion2 MSS Component versions

The table below lists the version of the SmartFusion2 MSS which must be used.

Libero SoC Version	SmartFusion2 MSS version	System Builder
11.4	1.1.209	1.0
11.5	1.1.300	1.0

Table 1 – SmartFusion2 MSS component versions

Hardware Requirements

This tutorial can be used on the following Microsemi SmartFusion2 boards:

- SmartFusion2 Starter kit with M2S050_ES-FGG896 (SF2-STARTER-KIT-ES-2[1])
- SmartFusion2 Starter kit with M2S050-FGG484 (SF2-STARTER-KIT)
- SmartFusion2 Starter Kit with M2S010-FGG484 (SF2-484-STARTER-KIT)
- SmartFusion2 Evaluation Kit with M2S025T-1FGG484 (M2S-EVAL-KIT)
- SmartFusion2 Security Evaluation Kit with M2S090TS-1FGG484 (M2S090S-EVAL-KIT[4])
- SmartFusion2 Development Kit with M2S050T-1FGG896 (SF2-DEV-KIT-PP[2], SF2-DEV-KIT-PP-1[2] or SF2-DEV-KIT[3])
- SmartFusion2 Advanced Development Kit with M2S150T-1FCG1152ES (M2S150-ADV-DEV-KIT-ES[4])

Notes:

1. SF2-STARTER-KIT-ES-2 has been discontinued. Please refer to PDN1403 for more details.
2. The SF2-DEV-KIT-PP and SF2-DEV-KIT-PP-1 part numbers have been discontinued per PDN1303 and PDN1307.
3. The SF2-DEV-KIT has been discontinued per PDN1406.
4. The SmartFusion2 Security Evaluation Kit and the Advanced Development Kit require a Libero SoC Platinum license. The Libero SoC Gold or Platinum license can be used for the other kits.

Extracting the source files

Extract *SF2_Cortex-M3_SB_tutorial.zip* to extract the required lab files to the <*C: or D:*>*Microsemiprj* folder on the HDD of your PC. Confirm that a folder named *SF2_Cortex-M3_tutorial* containing a sub-folder named *Source_files* was extracted.

Step 1 – Download Firmware Drivers

In order to use the Firmware drivers and sample projects for SmartFusion2 it is necessary to download them from the online Repositories into the IP vault on your machine.

1. Open the Microsemi SoC Firmware catalog (**Start > Programs > Microsemi Libero SoC v11.5 > Firmware Catalog v11.5 > Firmware Catalog**.
2. Look for the message New cores are available for download at the bottom of the Firmware Catalog.

Figure 2 - New cores message in the Firmware catalog

3. Click on **Download them now** to download all the most recent drivers for peripherals including Sample code and CMSIS compliant HAL etc. The resulting window should look like the figure below.

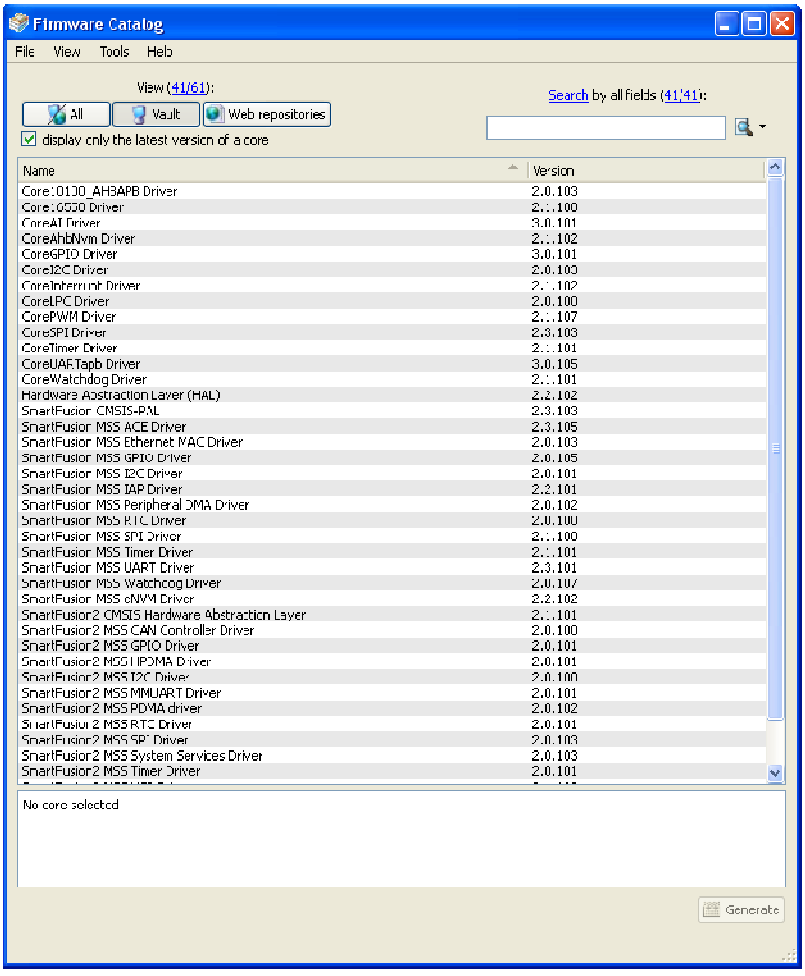


Figure 3 – Firmware Catalog fully Populated

4. Close the Firmware Catalog (**File > Exit**).

Step 2 – Creating a Libero SoC Project and configuring the SmartFusion2 MSS

In this step you will create a Libero SoC v11.5 project and configure the SmartFusion2 MSS.

Launching Libero SoC

1. Click **Start > Programs > Microsemi Libero SoC v11.5 > Libero SoC v11.5**, or click the shortcut on your desktop. The Libero SoC Project Manager will open.

Figure 4 - Libero SoC Project Manager

2. Click **Project > Tool Profiles** from the Libero SoC menu to open the Tool Profiles dialog box. Verify that the following tool profiles exist:
   - Synthesis: Synplify Pro ME I-2014.03M-SP1 contained in <Libero_v11.5_installation>\Synopsys\synplify_I201403MSP1\bin\synplify_pro.exe
   - Simulation: ModelSim ME v10.3c contained in < Libero_v11.5_installation>\Model\win32acoem\modelsim.exe
   - Programming: FlashPro v11.5 contained in <Libero_v11.5_installation>\designer\bin\flashpro.exe
3. Click **OK** to close the Tool Profiles dialog box.

4. Create a new project by selecting **New** on the Start Page tab (circled in the figure above), or by clicking **Project > New Project** from the Libero SoC menu. The New Project wizard will open.

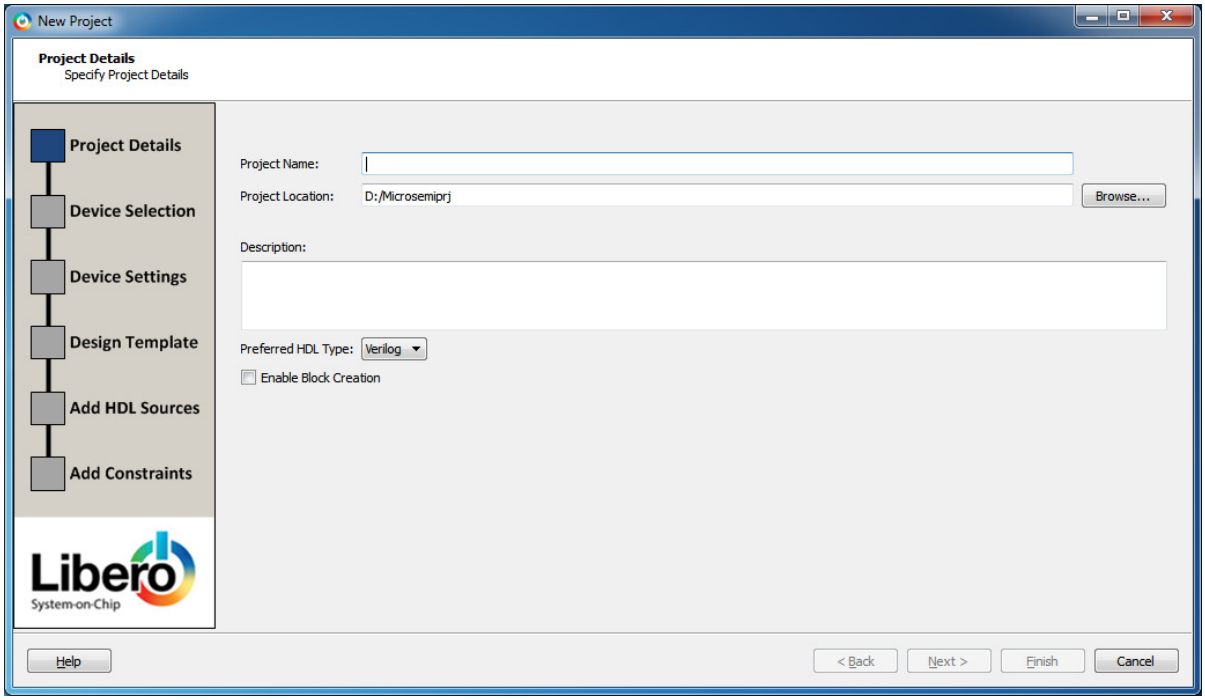


Figure 5 - Libero SoC New Project dialog box

5. Enter the information shown below in the Project Details page of the New Project dialog box then click **Next**:
   - Project Name: Simple_MSS_SB
   - Project Location: <C: or D:>\Microsemiprj\SF2_Cortex-M3_SB_tutorial (depending on where you extracted the source files)
   - Preferred HDL type: Verilog or VHDL
   - Enable Block Creation: un-checked

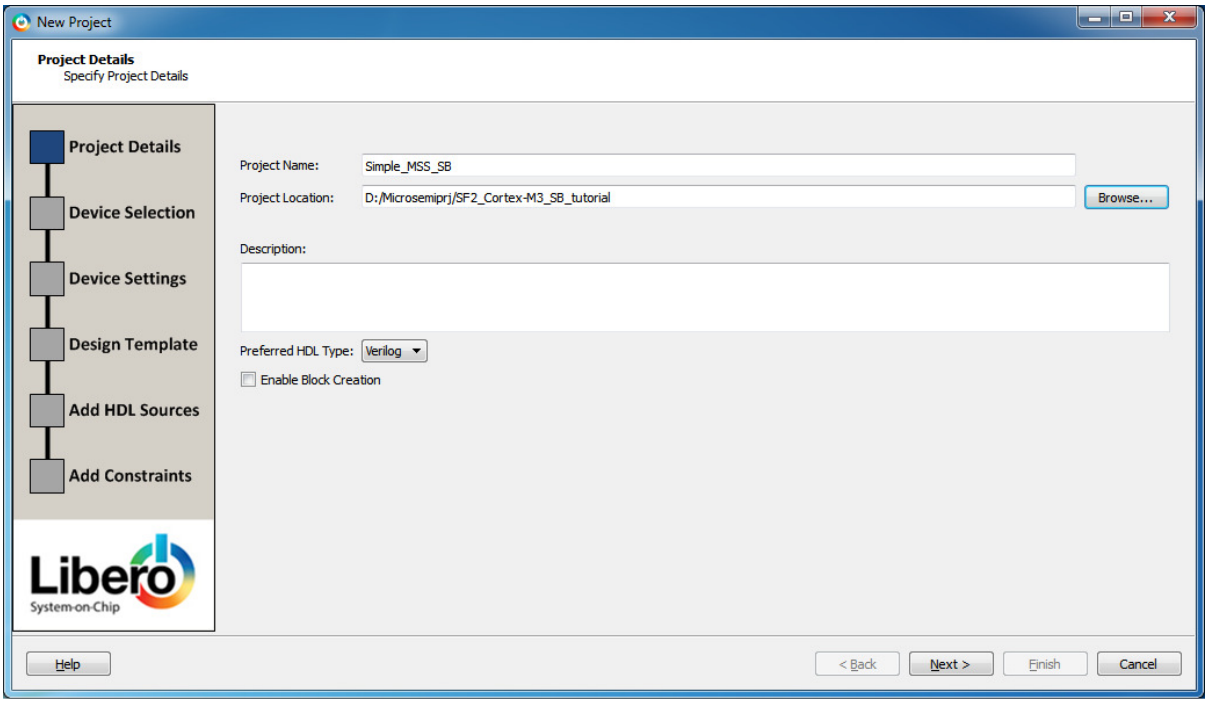


Figure 6 - Project Details

6. Enter the following in the Device Selection page of the New Project dialog box then click **Next**:
   - Family: SmartFusion2
   - Die: Refer to Table 2 below for your specific board
   - Package: Refer to Table 2 below for your specific board
   - Speed: Refer to Table 2 below for your specific board
   - Core Voltage(V): 1.2
   - Range: COM

Board	Die	Package	Speed	PLL Supply Voltage
SF2-STARTER-KIT-ES-2	M2S050T_ES	896 FBGA	STD	3.3
SF2-STARTER-KIT	M2S050	484 FBGA	STD	2.5
SF2-484-STARTER-KIT	M2S010	484 FBGA	STD	2.5
M2S-EVAL-KIT	M2S025T	484 FBGA	-1	3.3
M2S090S-EVAL-KIT	M2S090TS	484 FBGA	-1	3.3
SF2-DEV-KIT-PP	M2S050T	896 FBGA	-1	3.3
SF2-DEV-KIT-PP-1	M2S050T	896 FPGA	-1	3.3
SF2-DEV-KIT	M2S050T	896 FBGA	-1	3.3
M2S150-ADV-DEV-KIT-ES	M2S150T	1152 FC	-1	3.3

Table 2 – Die, Package, Speed Grade and PLL settings for the supported target boards

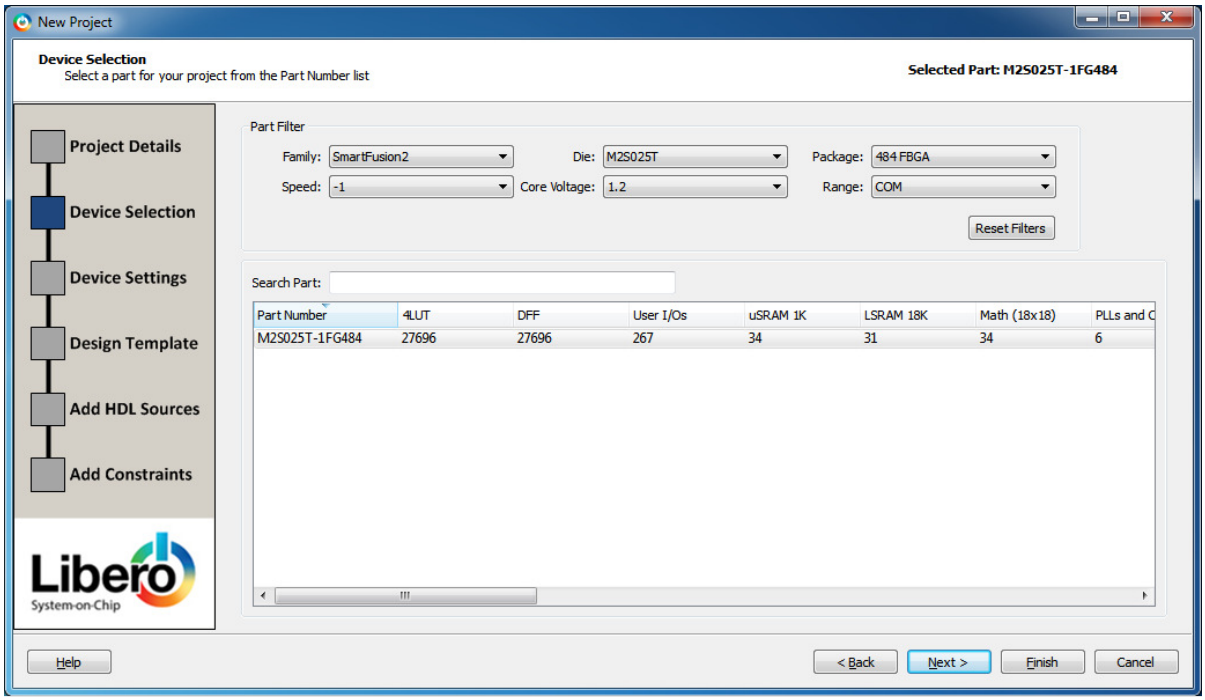


Figure 7 - Device selection (settings for M2S-EVAL-KIT board shown)

7. Enter the following in the Device Settings page of the New Project dialog box then click **Next**:
   - Default I/O Technology: LVCMOS 2.5V (default)
   - Reserve Pins for Probes: checked (default)
   - Power Supplies
     - PLL Supply Voltage (V): Refer to the table on the previous page for your specific board
     - Ramp rate: 100ms Minimum (default)
   - System Controller Suspend Mode: un-checked (default)

   The PLL Supply voltage can be either 2.5V or 3.3V. The voltage setting in the New Project dialog box must match the PLL Analog Supply voltage on the board to ensure the PLL works correctly. The PLL Analog Supply voltage is tied to 3.3V on some of the target boards, so the setting must be changed. Refer to Table 2 for the PLL Supply voltage for your target board. Change the supply voltage using the pull-down menu if needed.

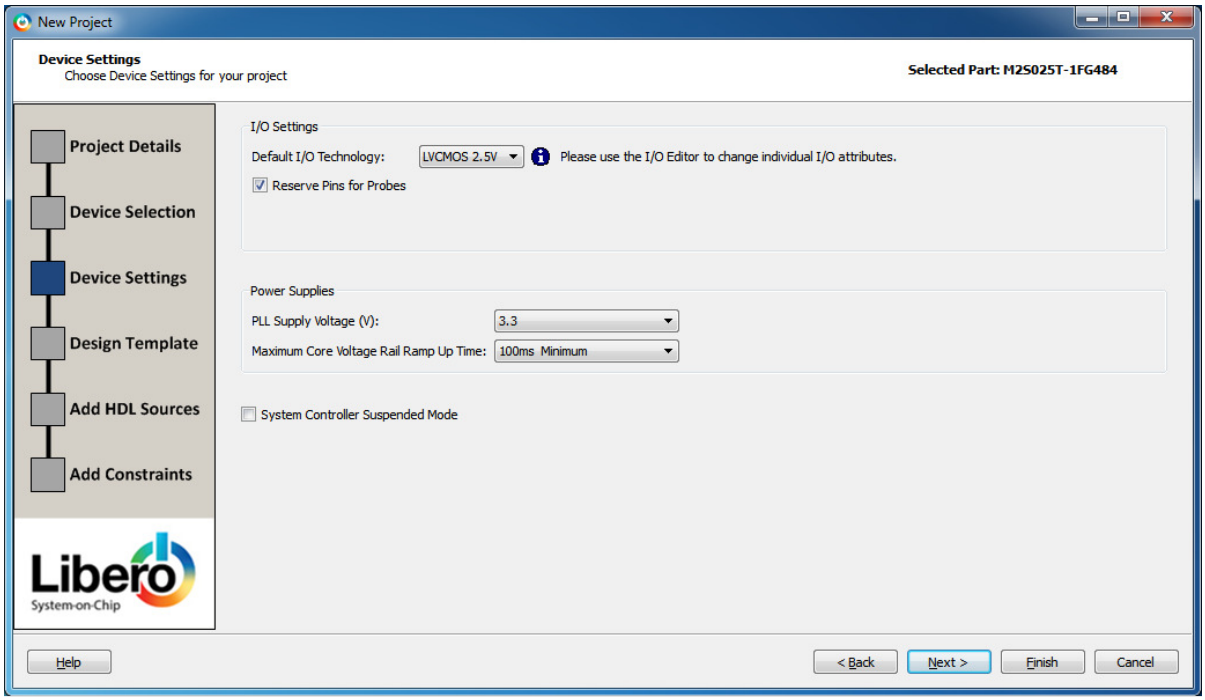


Figure 8 - Device Settings (settings for M2S-EVAL-KIT board shown)

8. Enter the following in the Design Template page of the New Project dialog box then click **Next**:
   - Design Templates and Creators: Create a System Builder based design
   - Design Methodology:
     - Use Standalone Initialization for MDDR/FDDR/SERDES peripherals: un-checked (default)

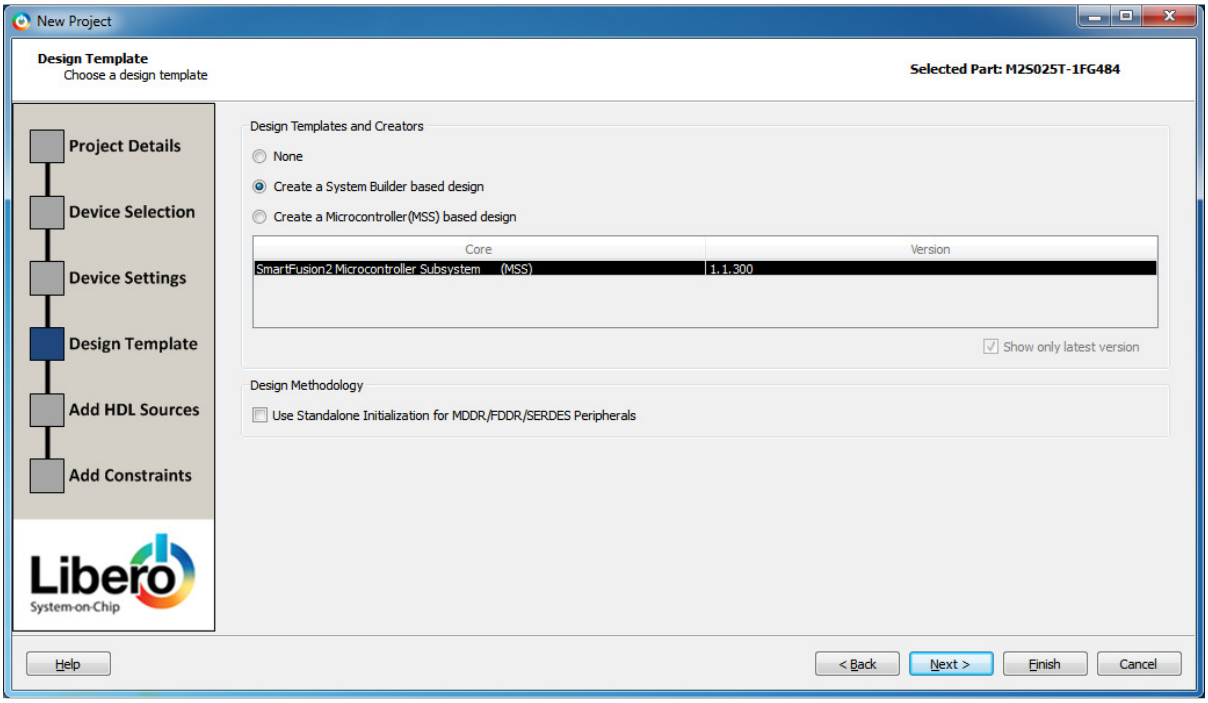


Figure 9 - Design Template settings

9. The Add HDL Source Files page will be visible. There are no HDL source files for this lab. Click **Next**.
10. The Add Constraints page will be visible. An I/O constraint file has been provided. Import the file into the project by clicking **Import File**.

11. Enter the following in the Import Files dialog box then click **Open**:
    - Location: <C: or D:>\Microsemiprj\SF2_Cortex-M3_SB_tutorial\Source_files
    - File name: SF2_Simple_MSS_IO.pdc
    - Files of type: I/O Constraint Files (*.pdc)

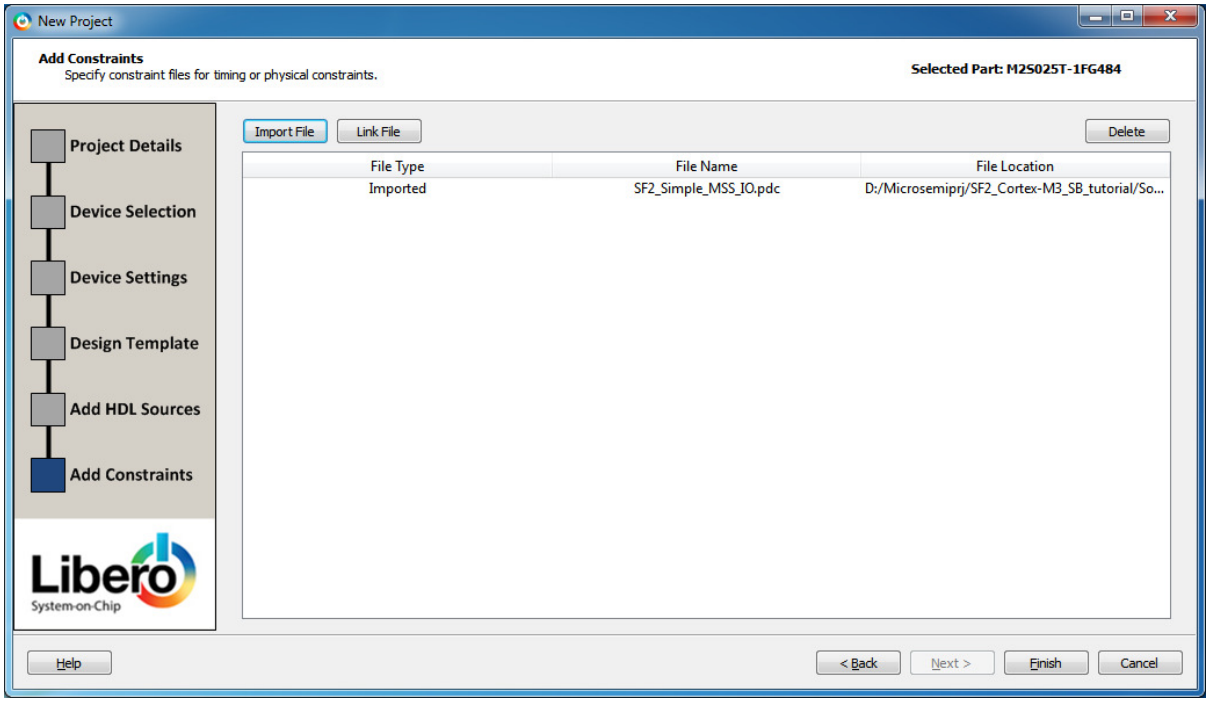


Figure 10 - Imported I/O Constraint file

12. Click **Finish** in the New Project dialog box.
13. Enter *Simple_MSS_system* in the "Enter a name for your system" dialog box and click **OK**.

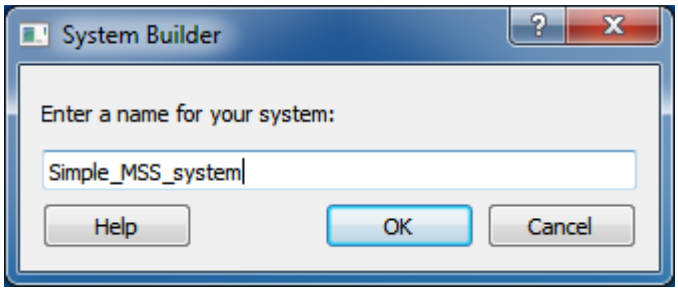


Figure 11 - Entering a name for the system

Building the System with System Builder

14. System Builder will open with the "System Builder - Device Features" page visible as shown below.

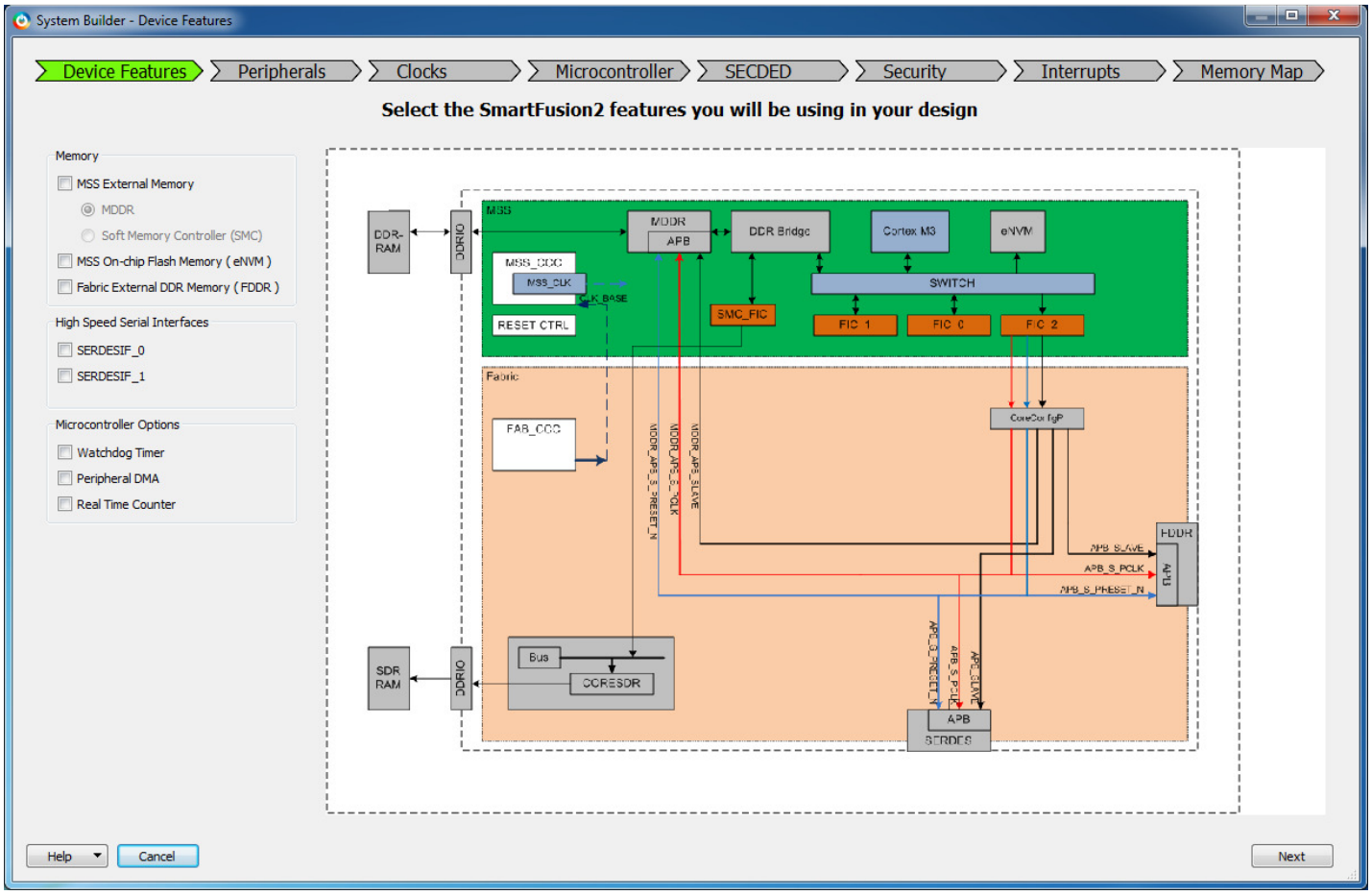


Figure 12 – SmartFusion2 System Builder

15. Enter the following on the "System Builder - Device Features" page:

- Memory
  - MSS External Memory — un-checked (default)
  - MSS On-chip Flash Memory (eNVM) — checked
  - Fabric External DDR Memory* (FDDR) — un-checked (default)
- High Speed Serial Interfaces
  - SERDESIF_0 — un-checked (default)
  - SERDESIF_1* — un-checked (default)
- Microcontroller Options
  - Watchdog Timer — un-checked (default)
  - Peripheral DMA — un-checked (default)
  - Real Time Counter — checked

* Not available for all target boards

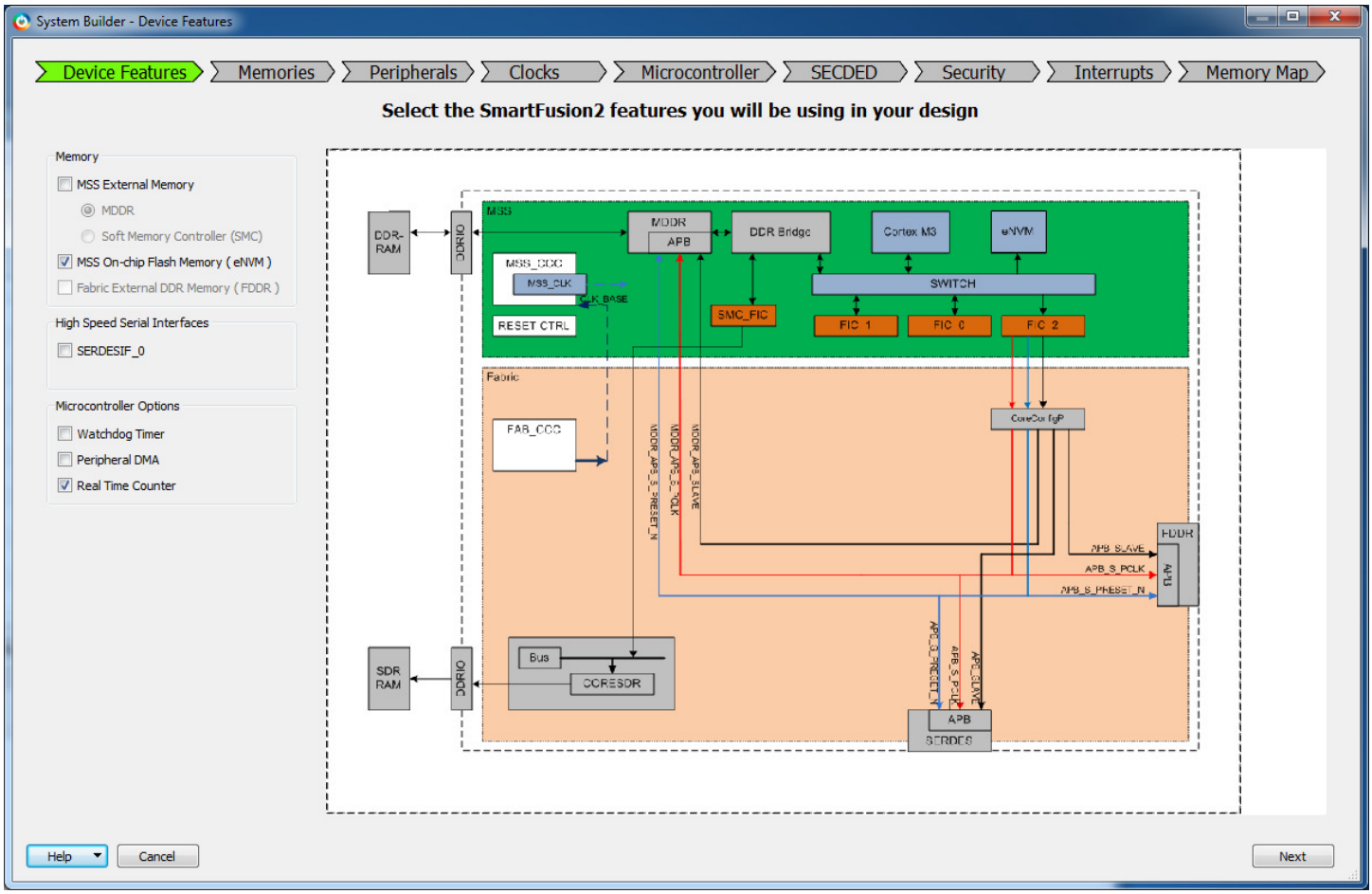


Figure 13 - System Builder Device Features after making selections (settings for M2S-EVAL-KIT board shown)

16. Click **Next**. The "System Builder – Memory" page will open.

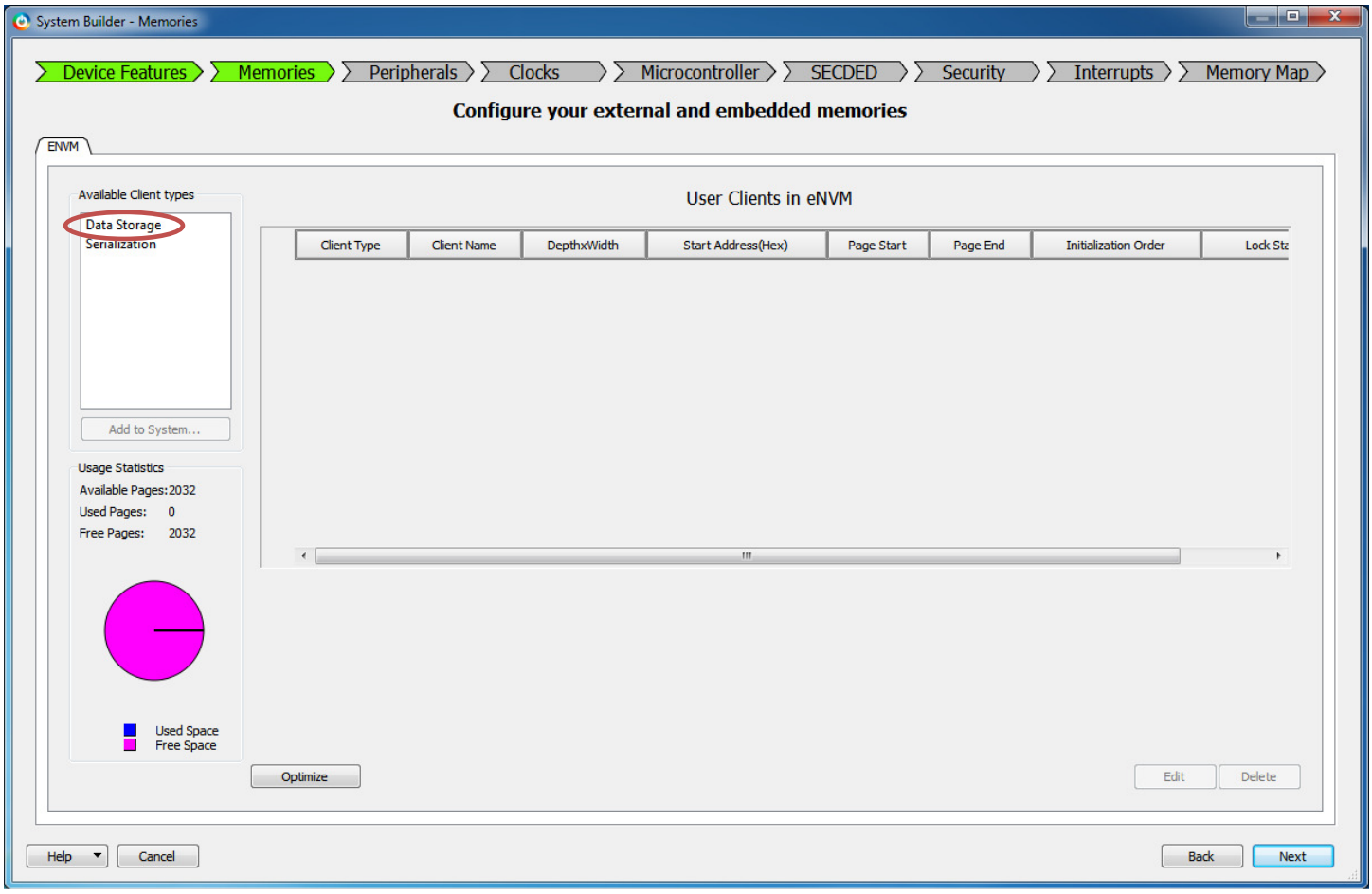


Figure 14 - System Builder Memories page

17. Click Data Storage under Available client types then click **Add to System**. Enter the following in the Add Data Storage Client dialog box:
    - Client name: PGM_store
    - Content: No content (Client is a placeholder)
    - Start address: 0 (default)
    - Size of word: 32 bits
    - Number of words: 2048
    - Use as ROM: un-checked (default)
    - Use Content for Simulation: un-checked (default)

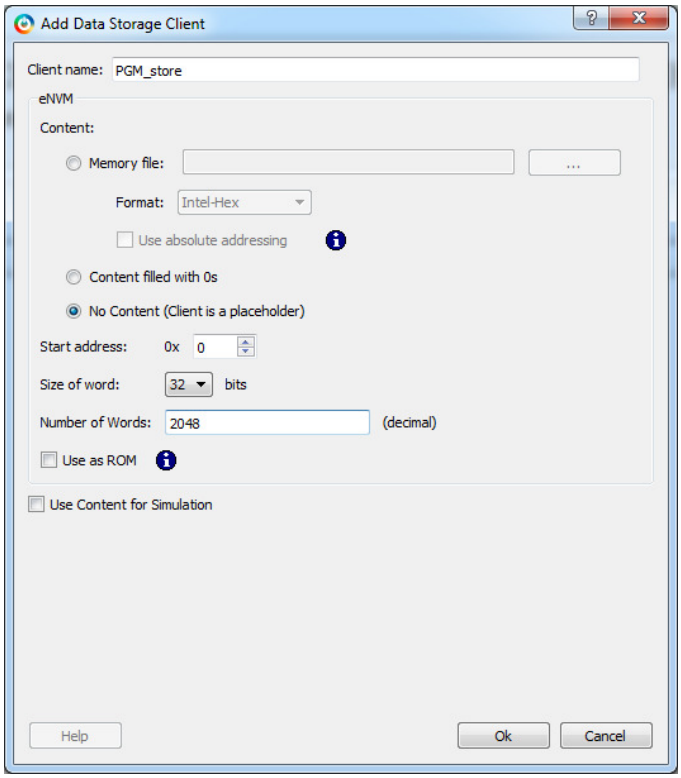


Figure 15 - Configuring a data storage client

18. Click **Ok** to close the Add Data Storage Client dialog box.
19. The client will be visible in the System Builder Memories page.

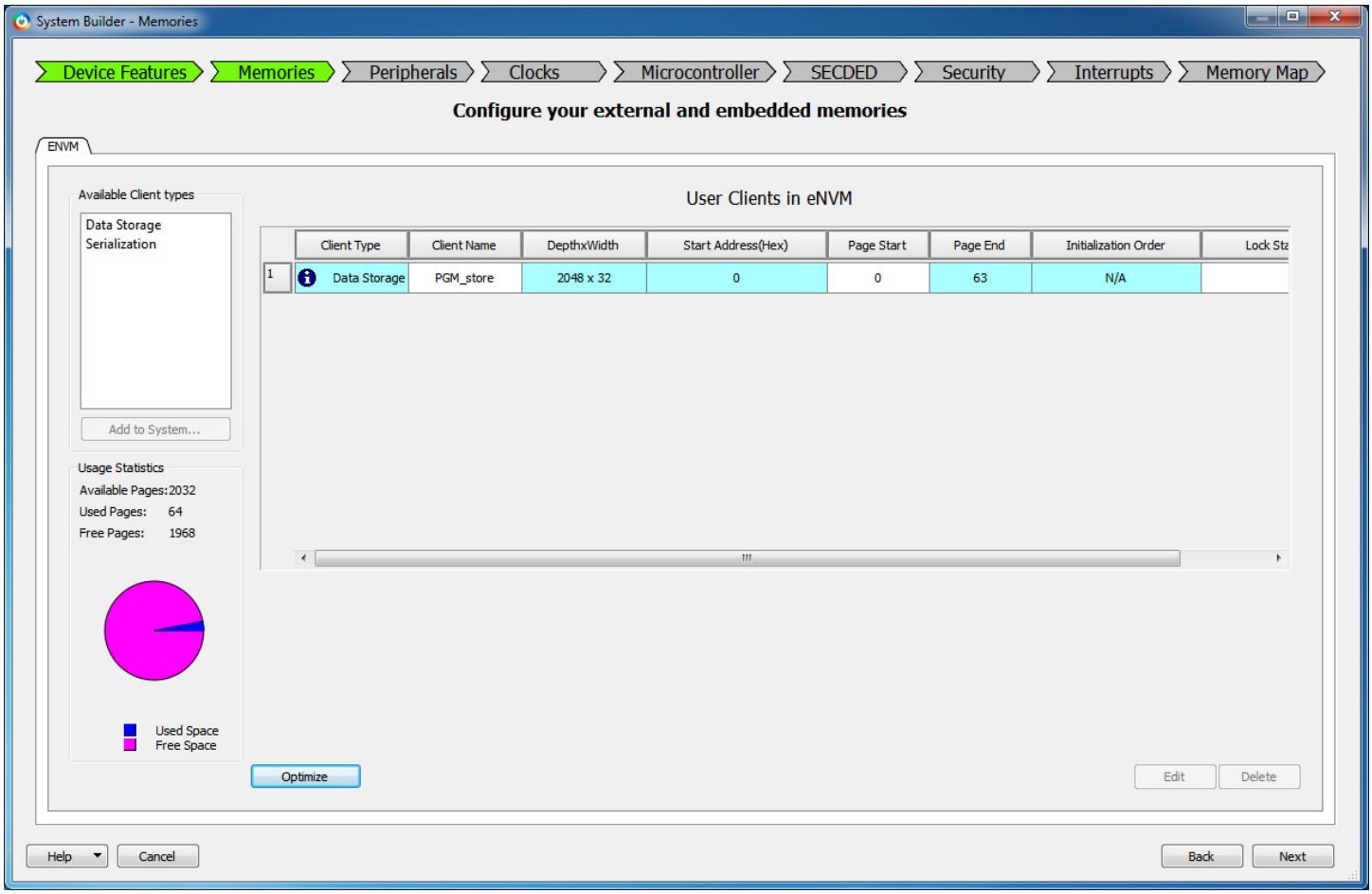


Figure 16 - System Builder Memories page after adding the data storage client

20. Click **Next**. The "Sytem Builder – Peripherals" page will open. Here you can enable or disable MSS peripherals and add fabric peripherals. Peripherals can be configured by clicking the wrench symbol ( ). The number of peripherals used can be changed by editing the number in the Quantity column.

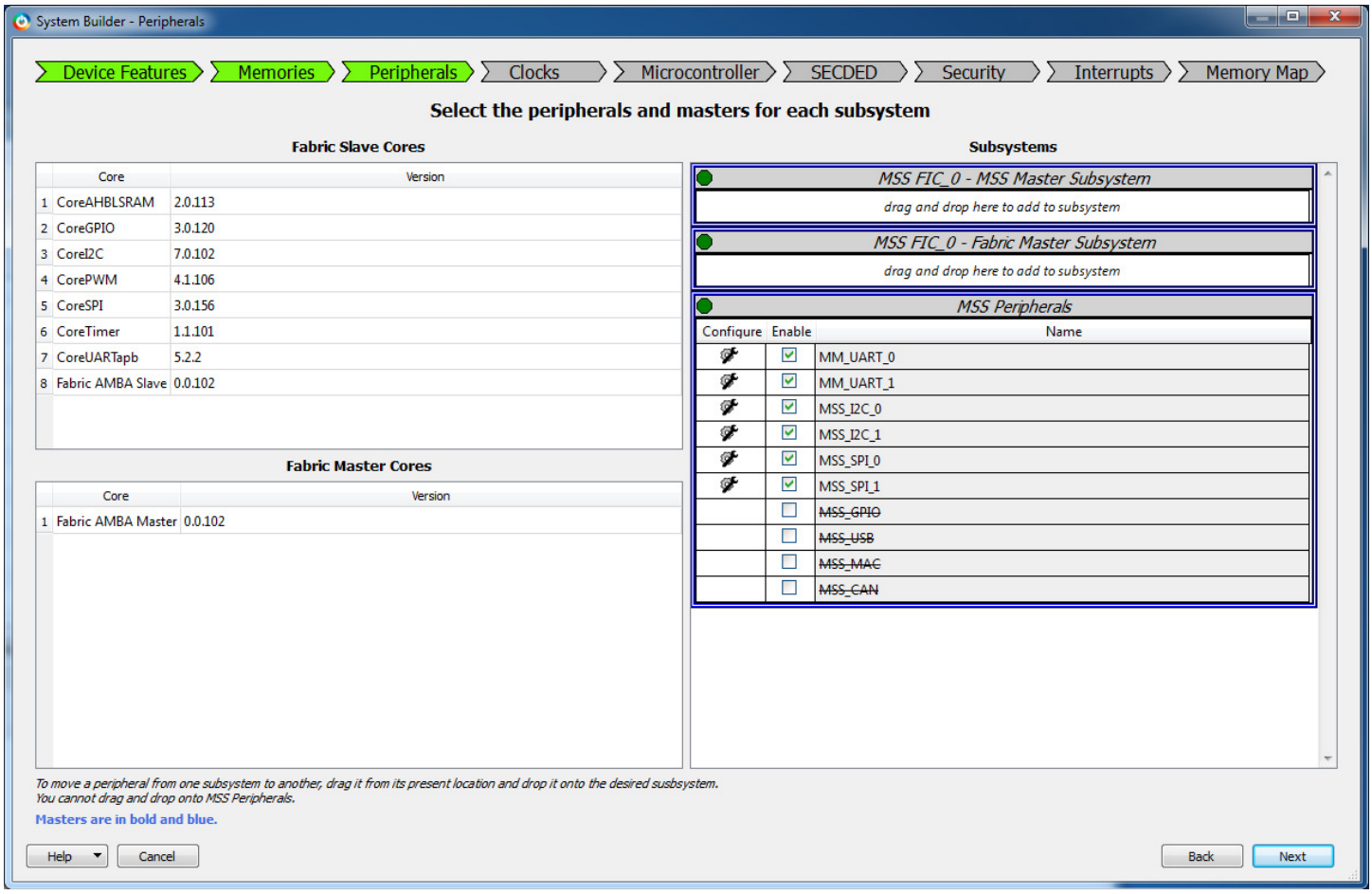


Figure 17 - System Builder Peripherals page (M2S-EVAL-KIT board shown)

21. Disable the MSS_I2C_0, MSS_I2C_1, MSS_SPI_0 and MSS_SPI_1 peripherals by clicking the box under Enable.

If the target board is the SmartFusion2 Development kit (SF2-DEV-KIT-PP, SF2-DEV-KIT-PP-1 or SF2-DEV-KIT), complete the step 22 below. Skip step 22 if you are using any other board.

22. Double-click the wrench symbol ( ) next to MM_UART_0 to open the configurator. Enter the following in the Configuring MM_UART_0_0 dialog box:
    - Duplex Mode: Full Duplex (default)
    - Transmission Mode Asynchronous (default)
    - Connect To: Fabric
    - Use Modem: un-checked

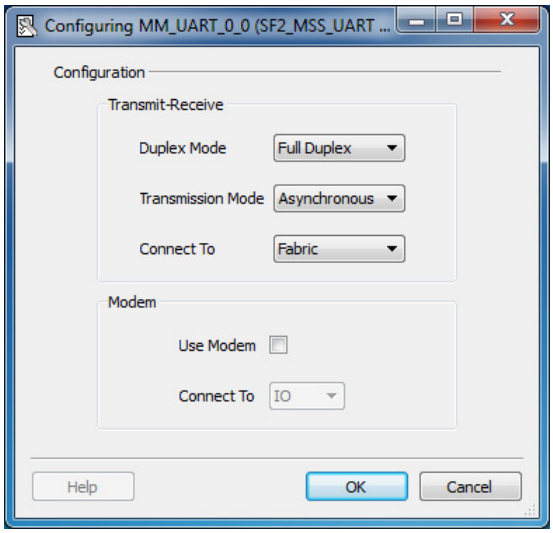


Figure 18 - Configuring MM_UART_0 (SmartFusion2 Development kit only)

23. Enable the MSS_GPIO peripheral by clicking the box under the Enable column.
24. Double-click the wrench symbol for the MSS_GPIO peripheral to open the MSS GPIO Configurator. Configure the GPIOs as shown below.
    - Set/Reset Definition: accept default settings
    - Configure GPIO[1:0] and GPIO_[8] per the table below:

GPIO ID	Direction	Package Pin	Connectivity
GPIO_0	Output	NA	FABRIC_A
GPIO_1	Output	NA	FABRIC_A
GPIO_8	Input	NA	FABRIC_A
GPIO_[2:7] and GPIO_[9:31]	Not Used	NA	NA

Table 3 – SmartFusion2 GPIO configuration

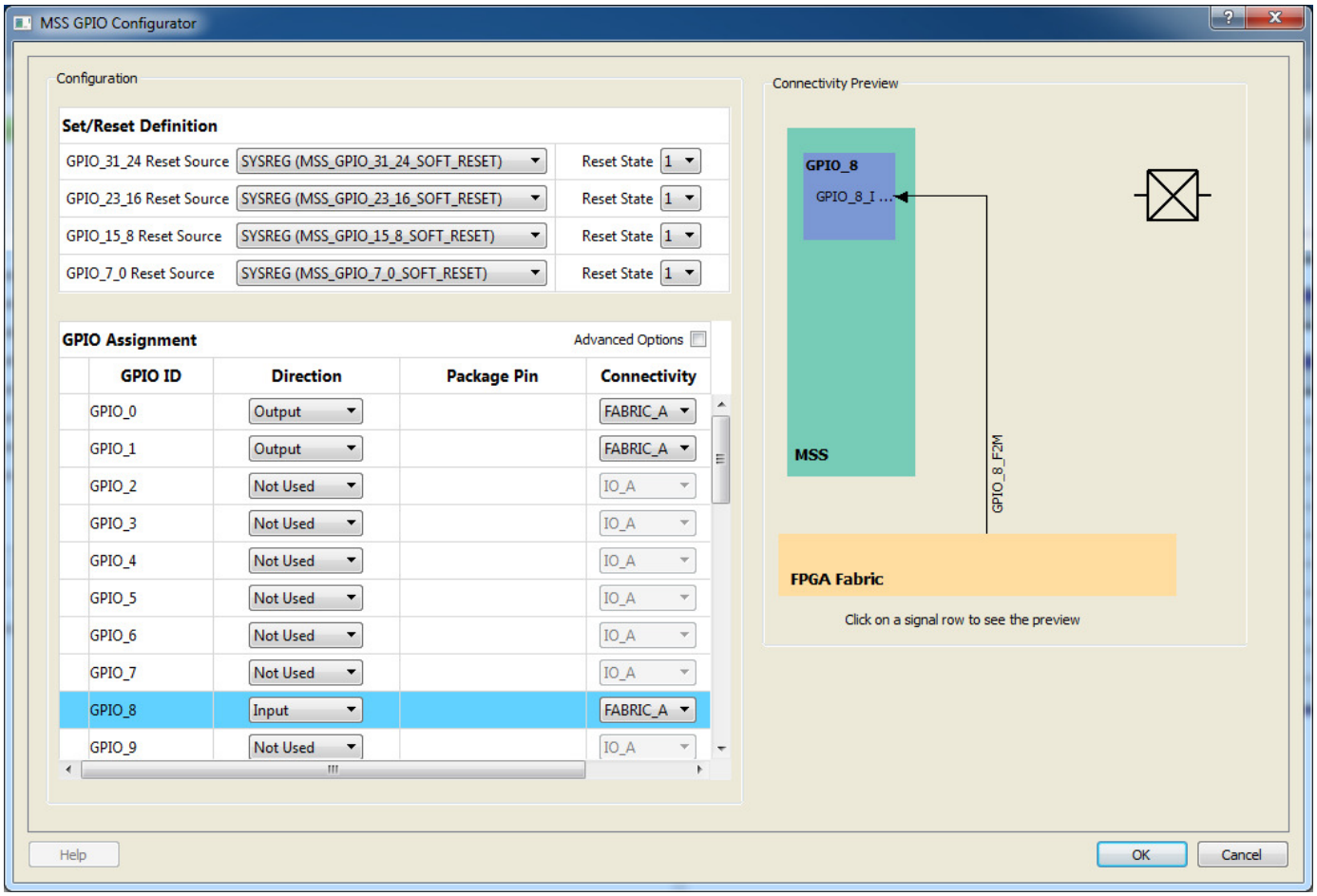


Figure 19 - SmartFusion2 GPIO Configuration

25. Click **OK** to close the MSS GPIO configurator.
26. The Peripherals page will look like the figure below.

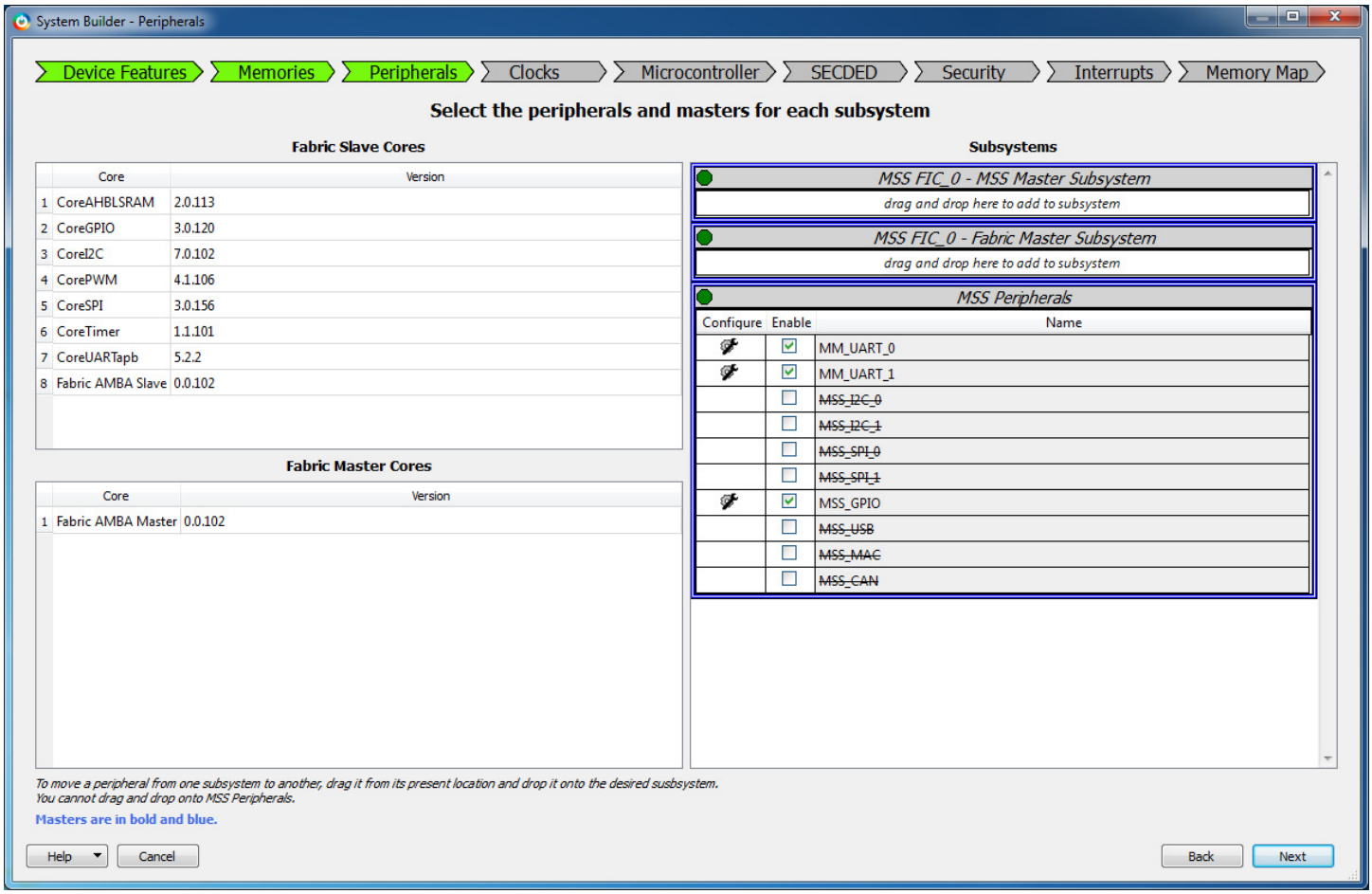


Figure 20 - System Builder Peripherals page after configuring peripherals (M2S-EVAL-KIT board shown)

27. Click **Next**. The "System Builder – Clocks" page will open. Enter the following:

- System Clock: Select On-chip 25/50 MHz RC Oscillator from the pull-down menu
- M3_CLK: 100 MHz (default)
- APB_0 CLK: M3_CLK/4 (25 MHz)
- APB_1 CLK: M3_CLK/1 (100 MHz) (default)
- FIC_0_CLK: M3_CLK/1 (100 MHz) (default)

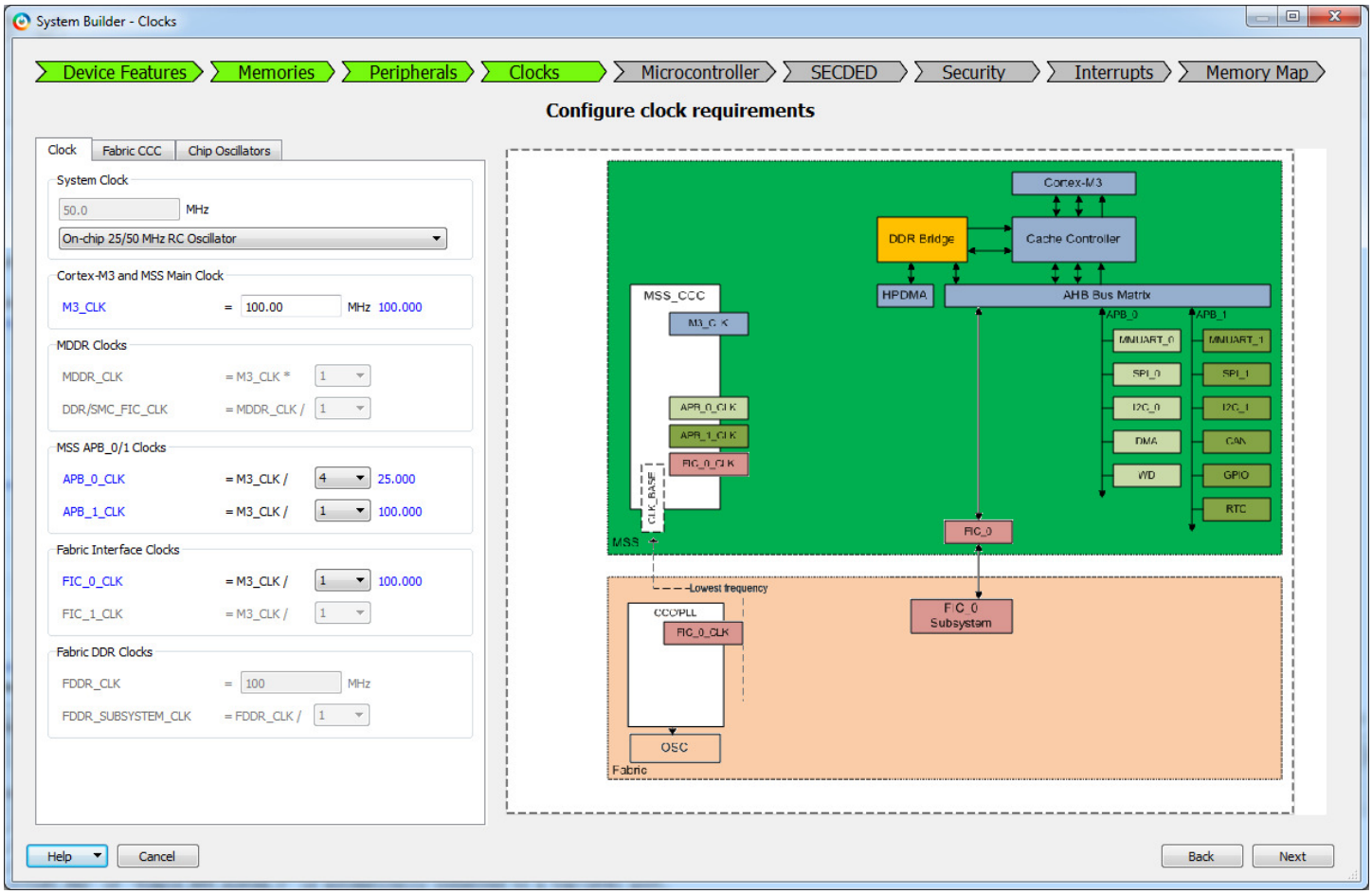


Figure 21 - Configuring the MSS clocks (settings for M2S-EVAL-KIT board shown)

28. Click **Next**. The "System Builder - Microcontroller" page will open. This page has multiple tabs which allow configuring the Cortex-M3 microcontroller, the Cache Controller and the AHB Bus Matrix.

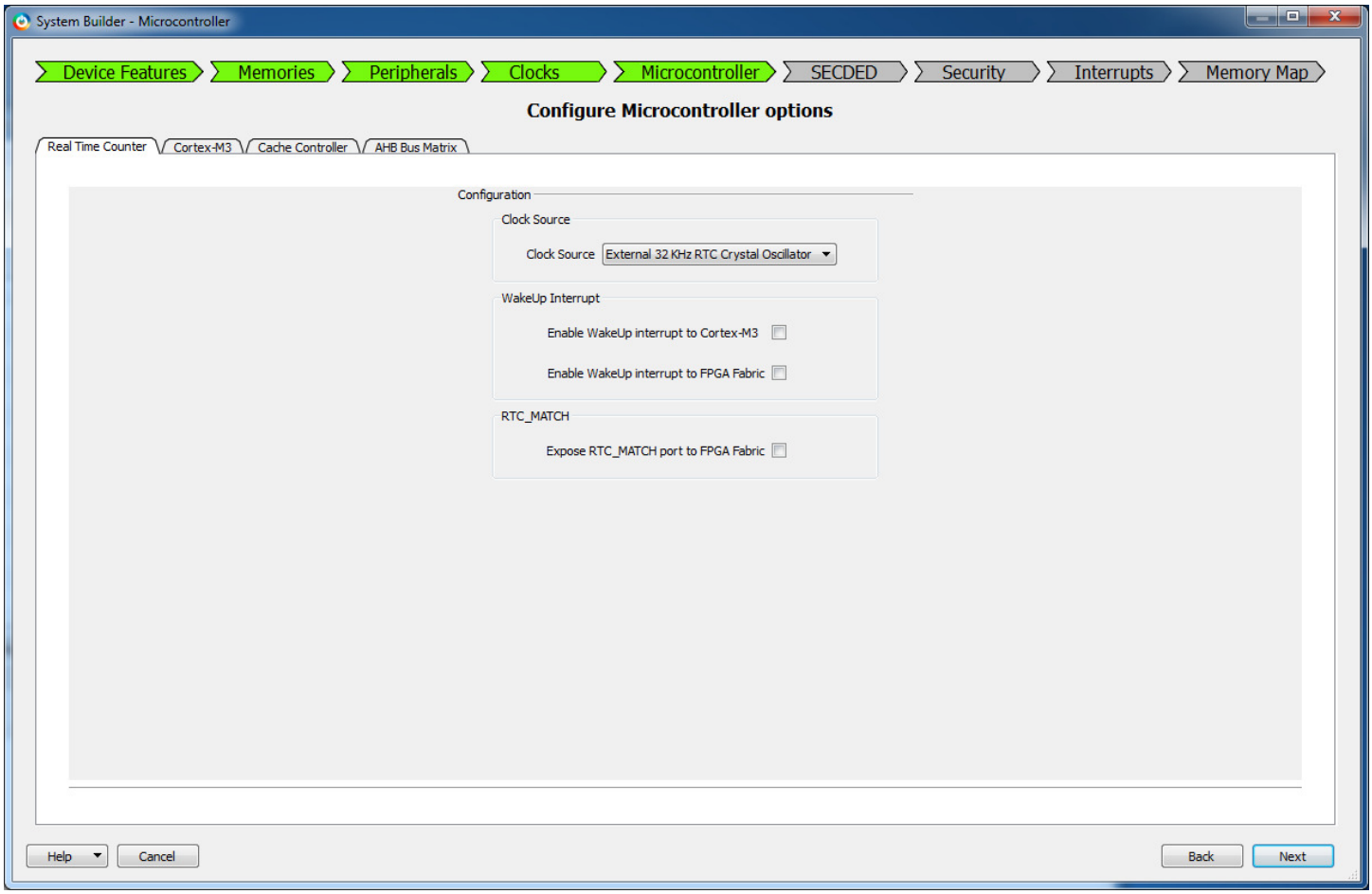


Figure 22 - System Builder Microcontroller options page

29. Select the On-chip 1 MHz RC Oscillator as the RTC clock source from the pull-down menu in the Clock Source field on the Real Time Counter tab.

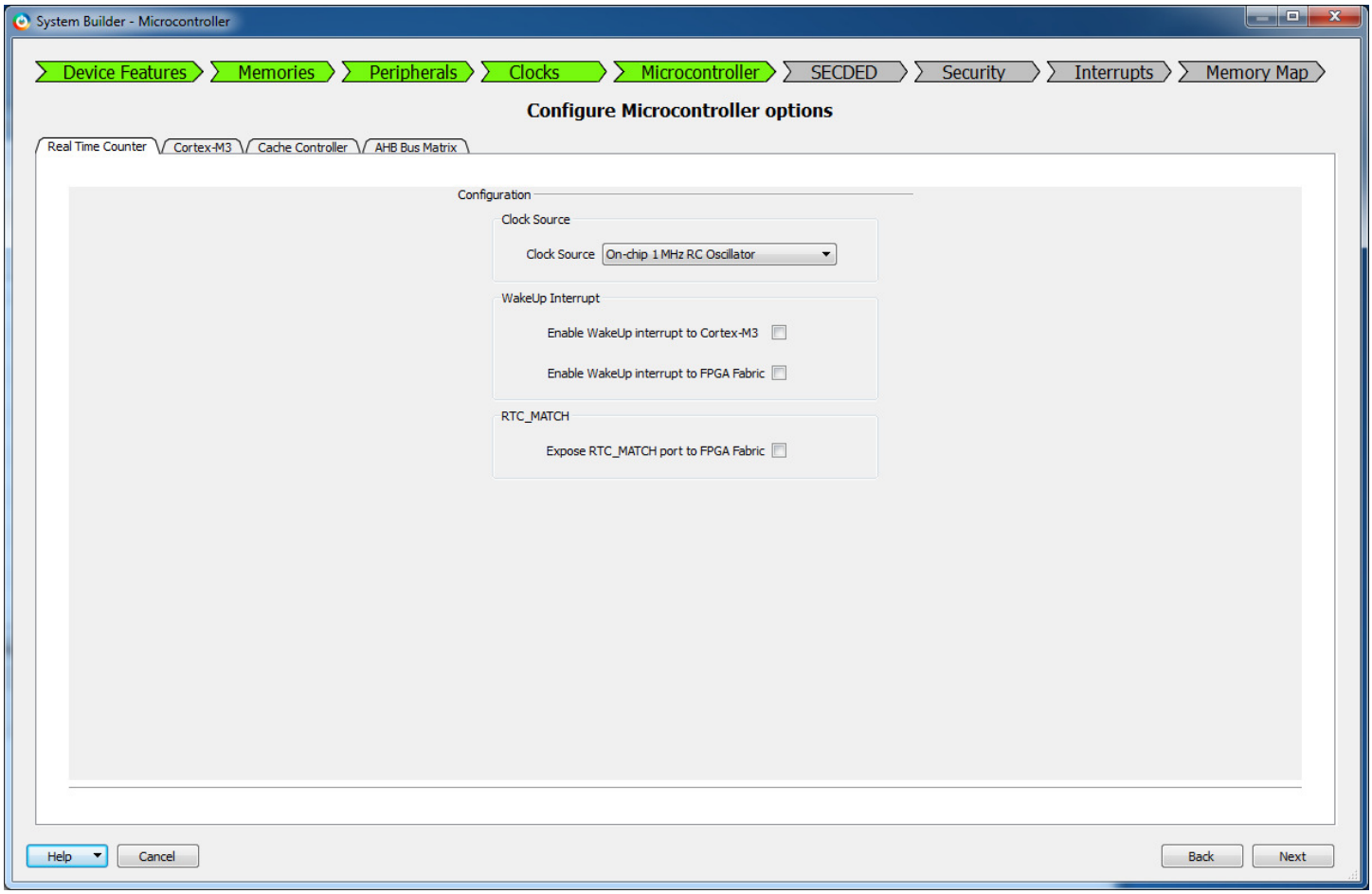


Figure 23 - System Builder Microcontroller options page after selecting the RTC clock source

30. Although we won't be making any other changes, take a moment to become familiar with the contents of each tab.
31. Click **Next** to accept the default settings on the Microcontroller options page.
32. The "System Builder – SECDED" (Single Error Correct / Double Error Detect) page will open. Use this page to enable SECDED for various memory blocks within the SmartFusion2 MSS. Although we won't be using SECDED for this design, take a moment to become familiar with the contents of this page.

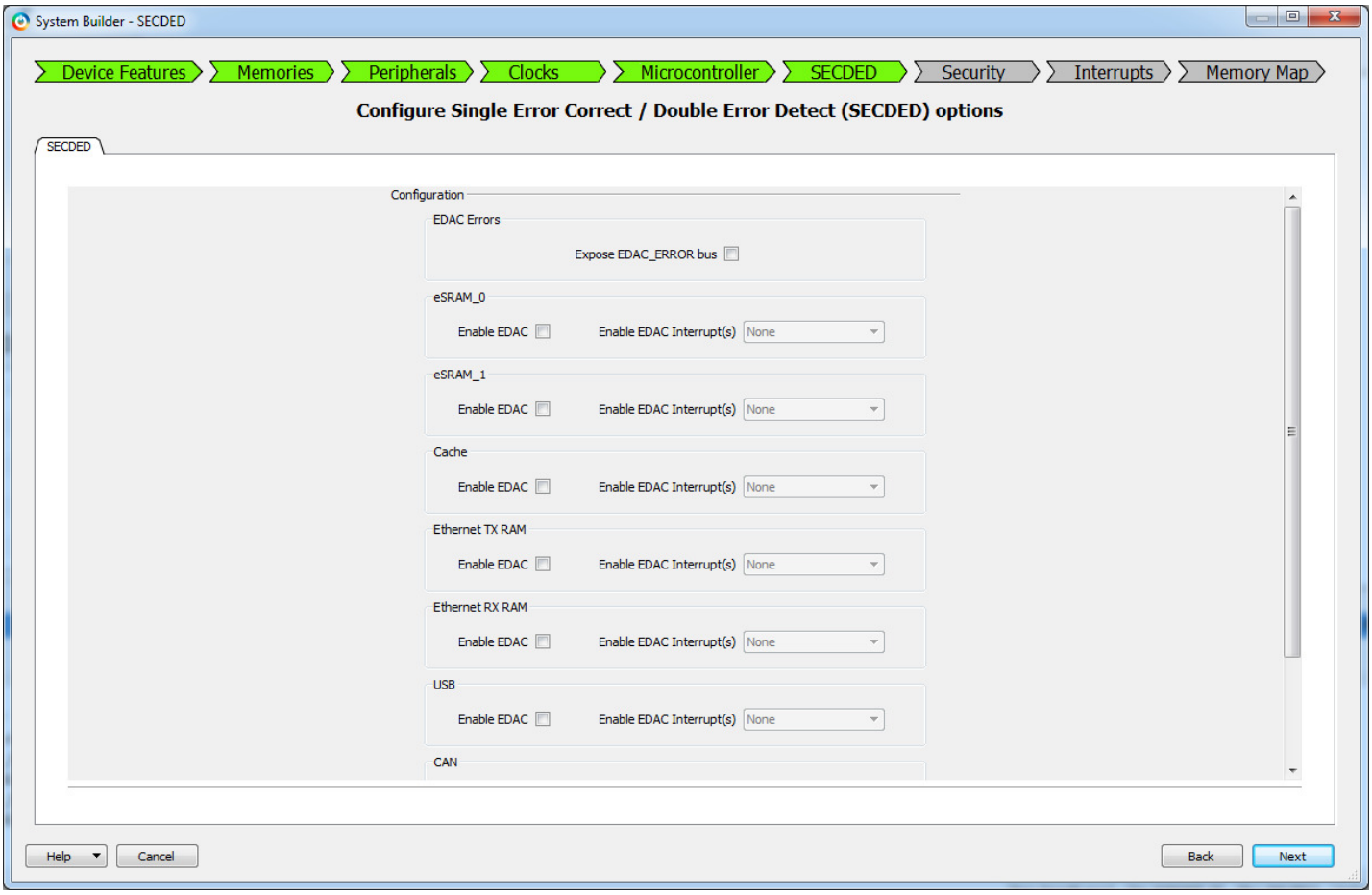


Figure 24 - System Builder SECDED options page

33. Accept the default settings on the page and click **Next**.
34. The “System Builder – Security” page will open. This page is used to set the Master to Slave Read/Write access control registers for devices that support Data Security features. The SmartFusion2 devices on the many of the target boards do not support these security features, so these options may be grayed out.
35. Click **Next** to accept the default settings.
36. The “System Builder – Interrupts” page will open. This page displays interrupt connections generated from attached fabric peripherals. No interrupts are displayed because this design does not include any fabric peripherals.
37. Click **Next**. The “System Builder – Memory Map” page will open. This page displays the addresses of fabric peripherals. This design does not have any fabric peripherals, so no memory map is displayed.
38. Click **Finish**.
39. Confirm that the messages “'Simple_MSS_system_sb_MSS' was successfully generated.”, “'Simple_MSS_system_sb' was successfully generated.” and “System Builder successfully generated 'Simple_MSS_system_sb'” appear in the Libero SoC log window.
40. Select **No** in the Information dialog box when prompted about organizing the constraint files for compile.

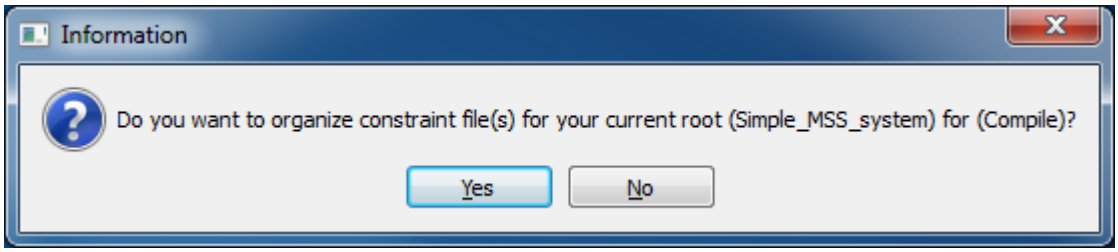


Figure 25 - Information dialog box after importing the PDC constraint file

41. A component named Simple_MSS_system_sb_0 will be visible in the SmartDesign canvas as shown below. If necessary click the Simple_MSS_system tab to display the SmartDesign canvas.

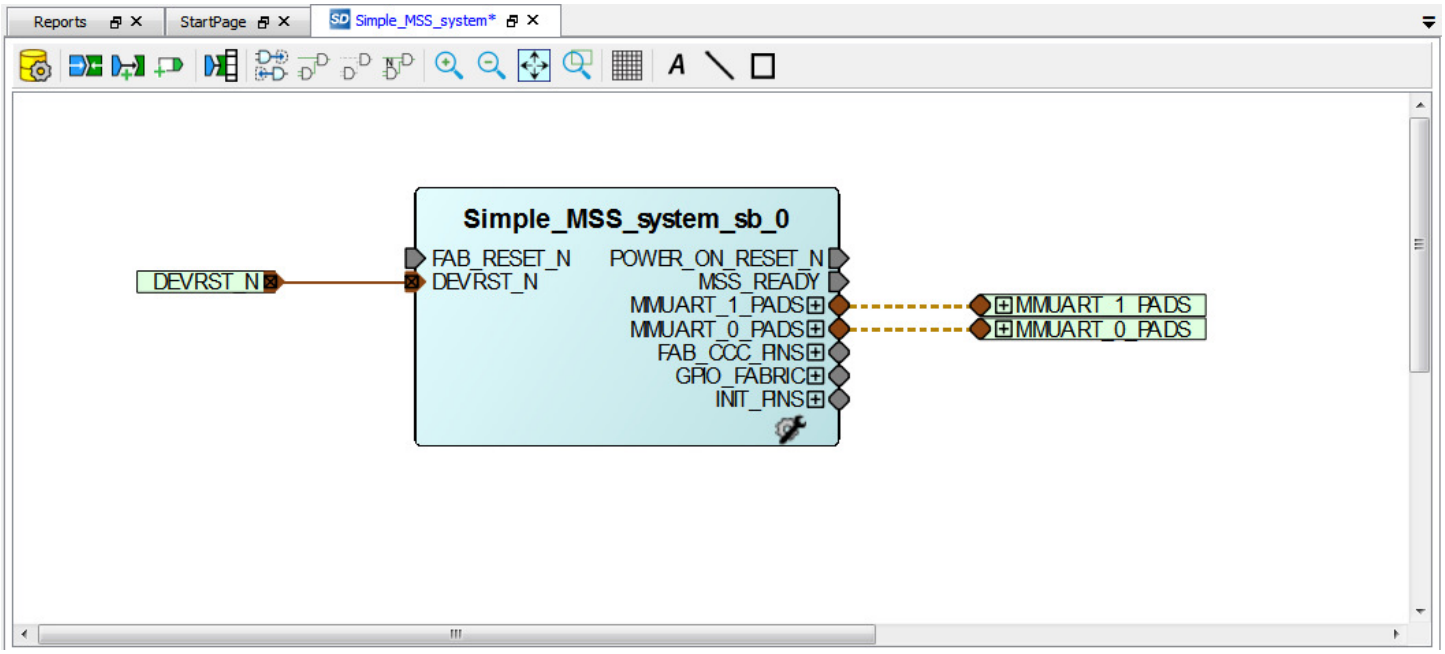


Figure 26 – Simple_MSS_system_sb_0 component

42. If the target board is the SmartFusion2 Development board the SmartDesign canvas will appear as shown in the figure below.

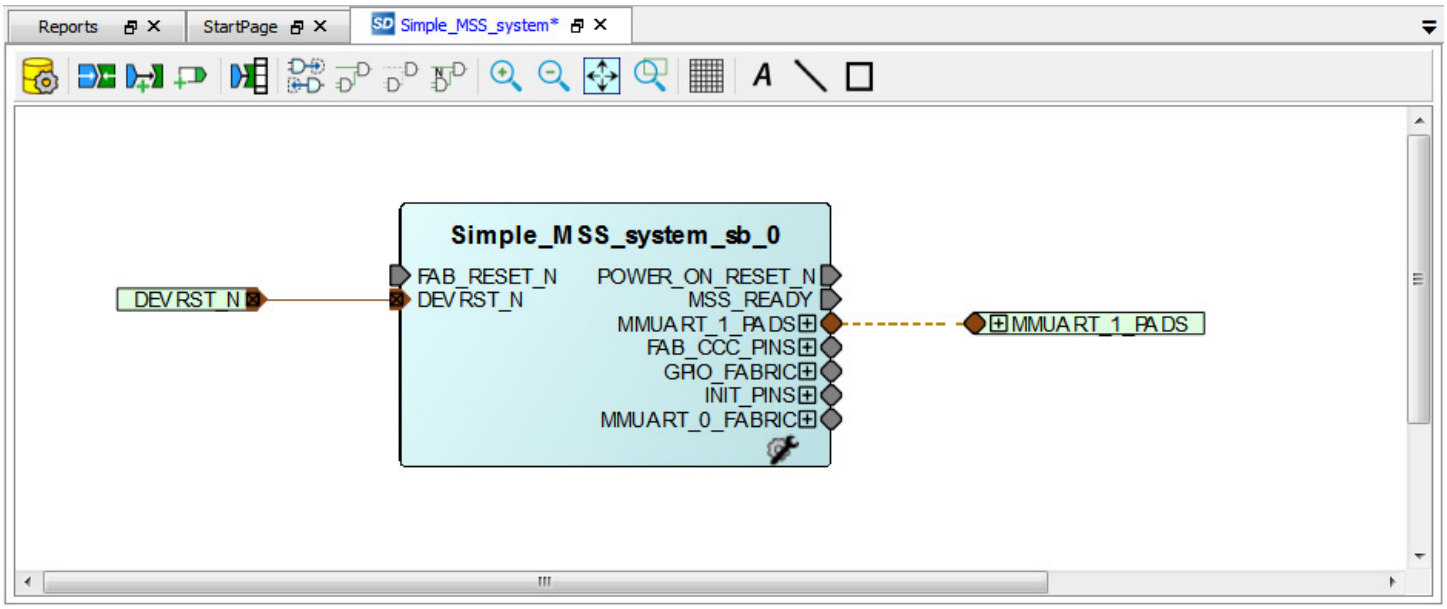


Figure 27 - Simple_MSS_system_0 component (Development kit)

43. Connect the FAB_RESET_N port to VCC by selecting the port, right-clicking and selecting **Tie High**.

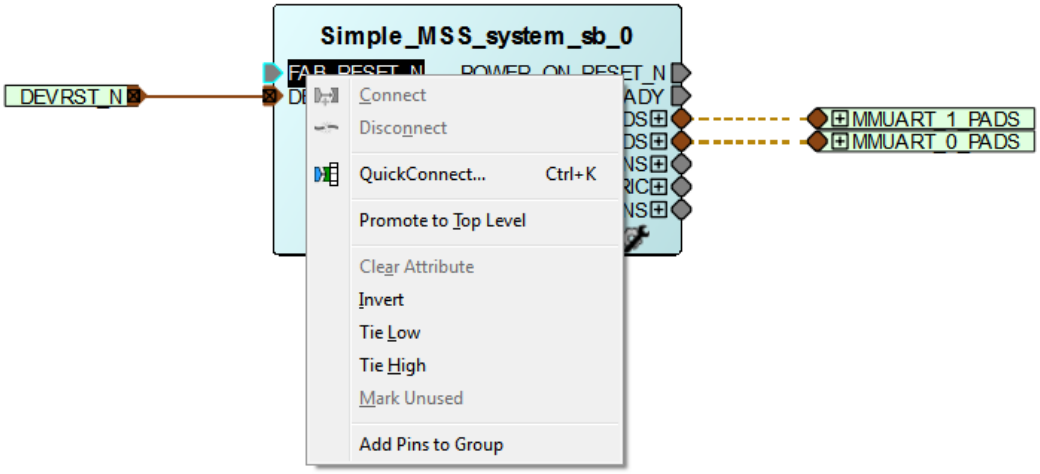


Figure 28 – Connecting FAB_RESET_N port to VCC

44. Mark the POWER_ON_RESET_N and MSS_READY output ports unused by selecting the port, right-clicking and selecting **Mark Unused**.
45. Click the + sign next to FAB_CCC_PINS to expand the pin group. Mark the FAB_CCC_GL0 and FAB_CCC_LOCK pins unused.
46. Click the + sign next to GPIO_FABRIC to expand the pin group. Promote the GPIO_0_M2F, GPIO_1_M2F and GPIO_8_F2M ports to the top level by selecting the port, right-clicking and selecting **Promote to Top Level**.
47. Click the + sign next to the INIT_PINS output port to expand the group. Mark the INIT_DONE output port unused.

If the target board is the SmartFusion2 Development kit, complete the step below. Skip to step 49 if you are using any other board.

48. Click the + sign next to the MMUART_0_FABRIC port to expand the group. Promote the MMUART_0_TXD_M2F and MMUART_0_RXD_F2M ports to the top level.

After making the pin connections the Simiple_MSS_system component will look like Figure 29 or Figure 30 below.

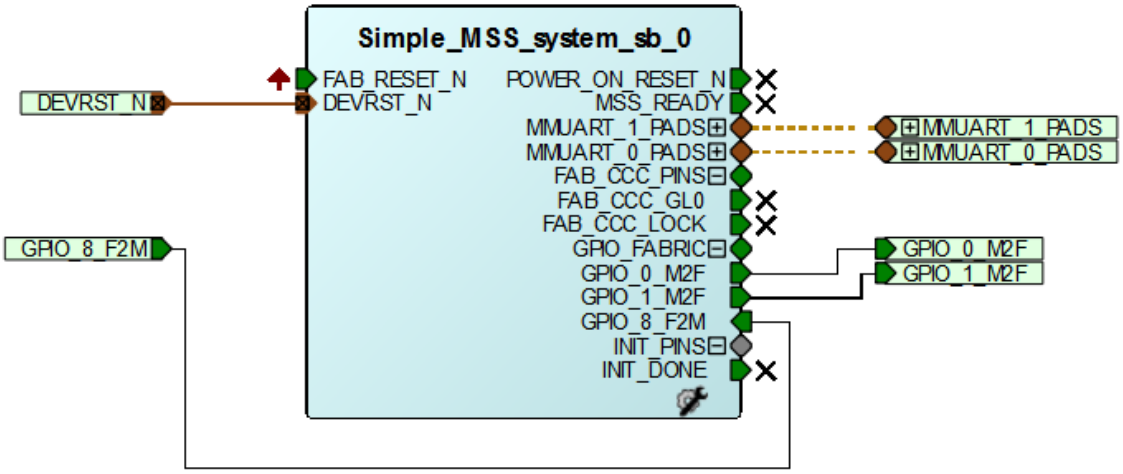


Figure 29 - Simple_MSS_system_sb_0 component after making pin connections (All except SmartFusion2 Development kit)

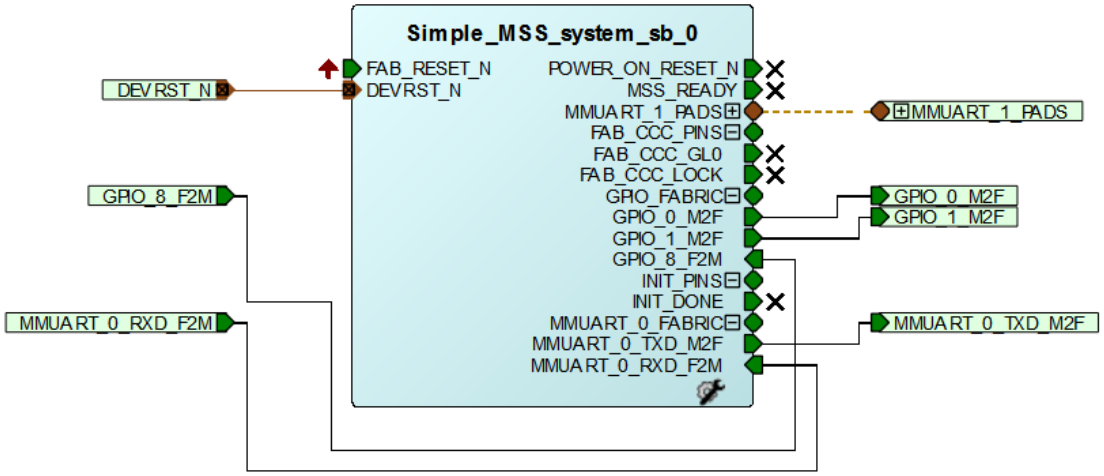


Figure 30 - Simple_MSS_system_sb_0 component after making pin connections (SmartFusion2 Development kit)

49. Save the design **(File > Save Simple_MSS_system**).
50. Click the Simple_MSS_system tab if the SmartDesign canvas is not visible. Generate the design by clicking **SmartDesign > Generate Component** or by clicking the Generate Component icon on the SmartDesign toolbar ( ).
51. The message "'Simple_MSS_system' was successfully generated." will appear in the Libero SoC log window if the design was generated without any errors. In addition, a green check mark will appear in under Create Design in the Design Flow window.

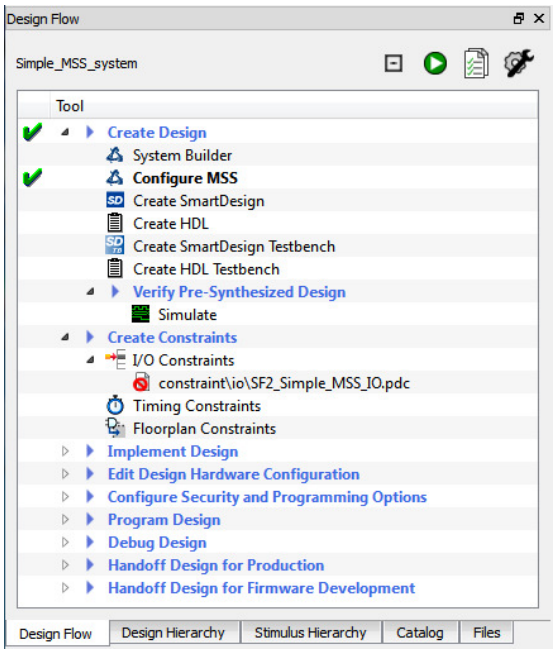


Figure 31 - Successful creation of SmartDesign component

52. Close the design (**File > Close Simple_MSS_system**).

Step 3 - Opening Simple_MSS_system in SmartDesign

To gain an understanding of the system that was created in the previous step you can open the design in SmartDesign.

1. If necessary, restore the work area (**View > Restore Work Area**). Expand Simple_MSS_system on the Libero SoC Design Hierarchy tab to expand the design hierarchy.

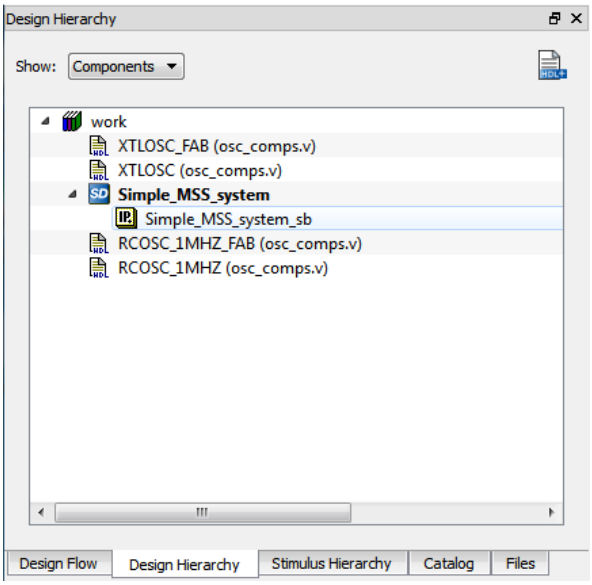


Figure 32 - Expanded design hierarchy

2. Right-click Simple_MSS_system_sb and select **Open as SmartDesign**.

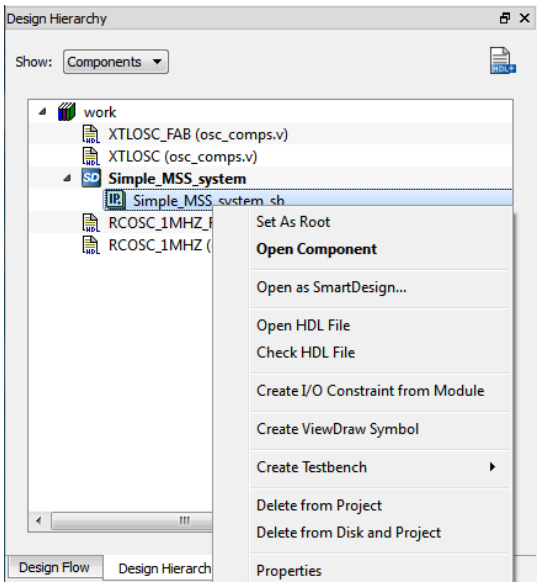


Figure 33 - Opening Simple_MSS_system_sb in SmartDesign

3. Click **OK** in the information dialog box.
4. The design will be visible in the SmartDesign canvas as shown in the figure below. If necessary, drag the components to improve the appearance in the canvas. Notice that the System Builder has automatically instantiated and connected different blocks based on the different options selected in the different pages of the System Builder.
   - **SYSRESET_POR:** This block generates the power on reset signal for the CORESF2RESET block
   - **CORESRESETP_0** (soft core): This block is responsible for managing all the reset mechanism needed for the system
   - **FABOSC_0:** Since we selected that the clock source for the System Clock is the on-chip 25_50MHz, this block is generating the reference clock for the Fabric CCC which intern generating the BASE_CLK for the system.
   - **CCC_0:** This is one of the fabric CCCs which contains a PLL. The PLL is used to generate the clock frequency used in the design.

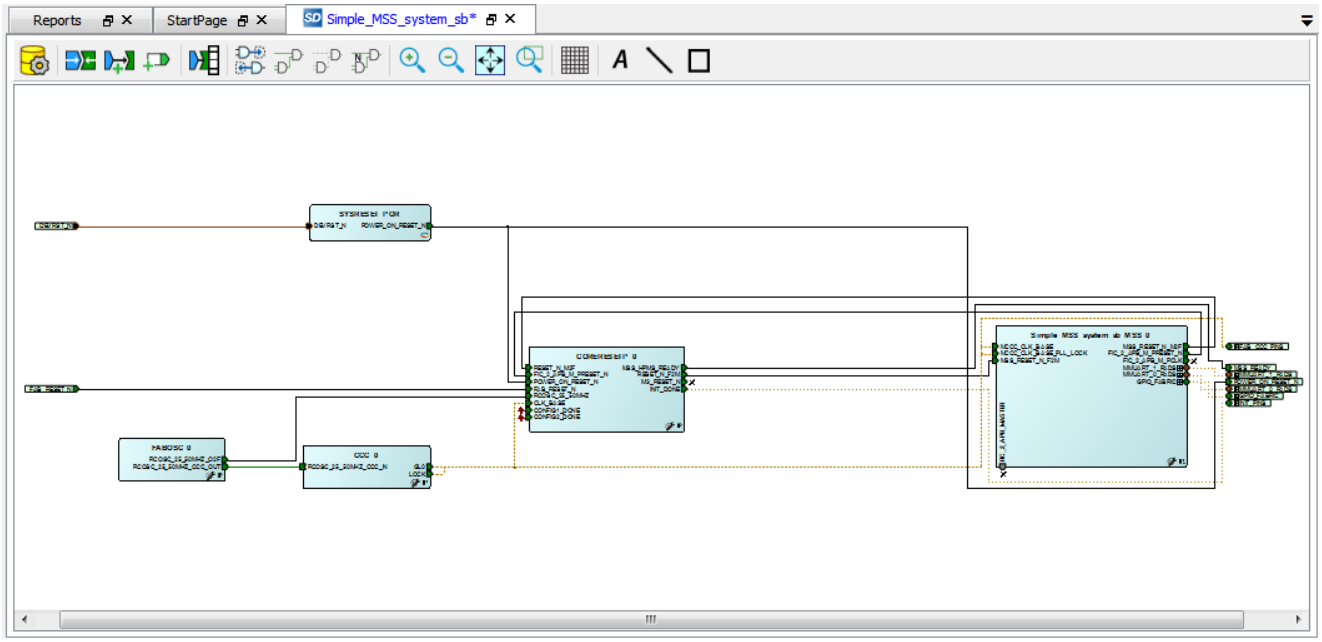


Figure 34 – Simple_MSS_system_sb design in the SmartDesign canvas

5. Generate the Simple_MSS_system component by clicking **SmartDesign > Generate Component** or by clicking the Generate Component icon. Confirm that the message "'Simple_MSS_system_sb' was successfully generated." appears in the Libero SoC Log window.

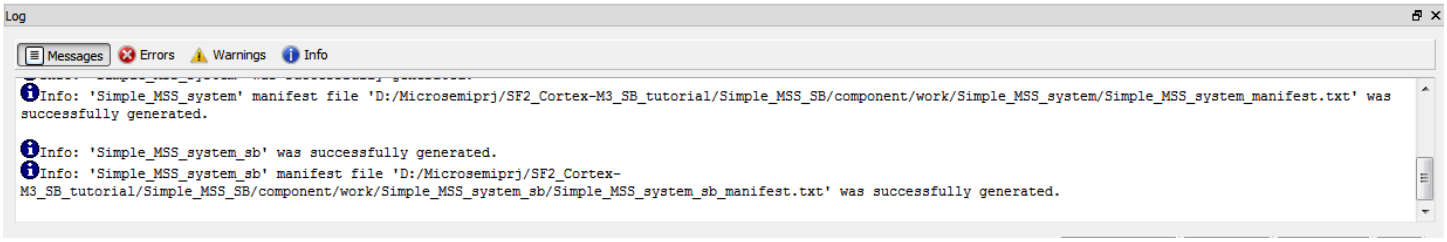


Figure 35 - Messages in Libero SoC Log window

6. Close the SmartFusion2 MSS configurator (**File > Close Simple_MSS_system_sb**).

Confirm that **Simple_MSS_system** appears in bold font on the Design Hierarchy tab indicating that this is the root level. If it does not, select Simple_MSS_system then right-click and select **Set As Root**.

7. Export the design summary and data sheet by selecting **File > Export > Design Summary**. Save the file to the <C: or D:>\Microsemiprj\SF2_Cortex-M3_SB_tutorial\Simple_MSS_SB folder. Click **Save**.
8. A dialog box will appear showing the location of the design summary. Click **OK** in the dialog box.

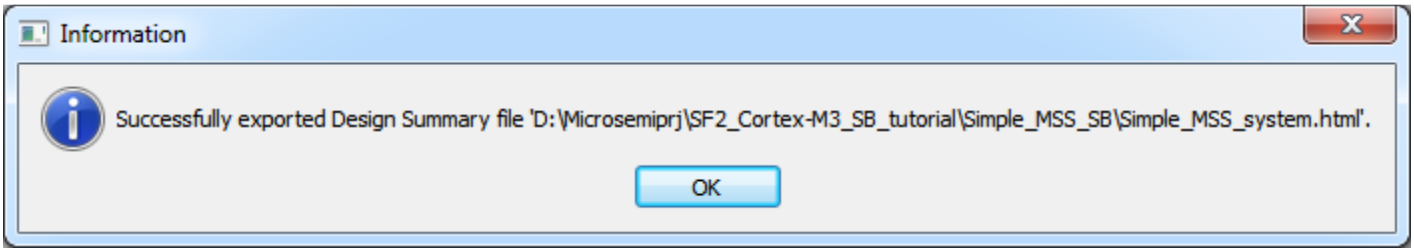
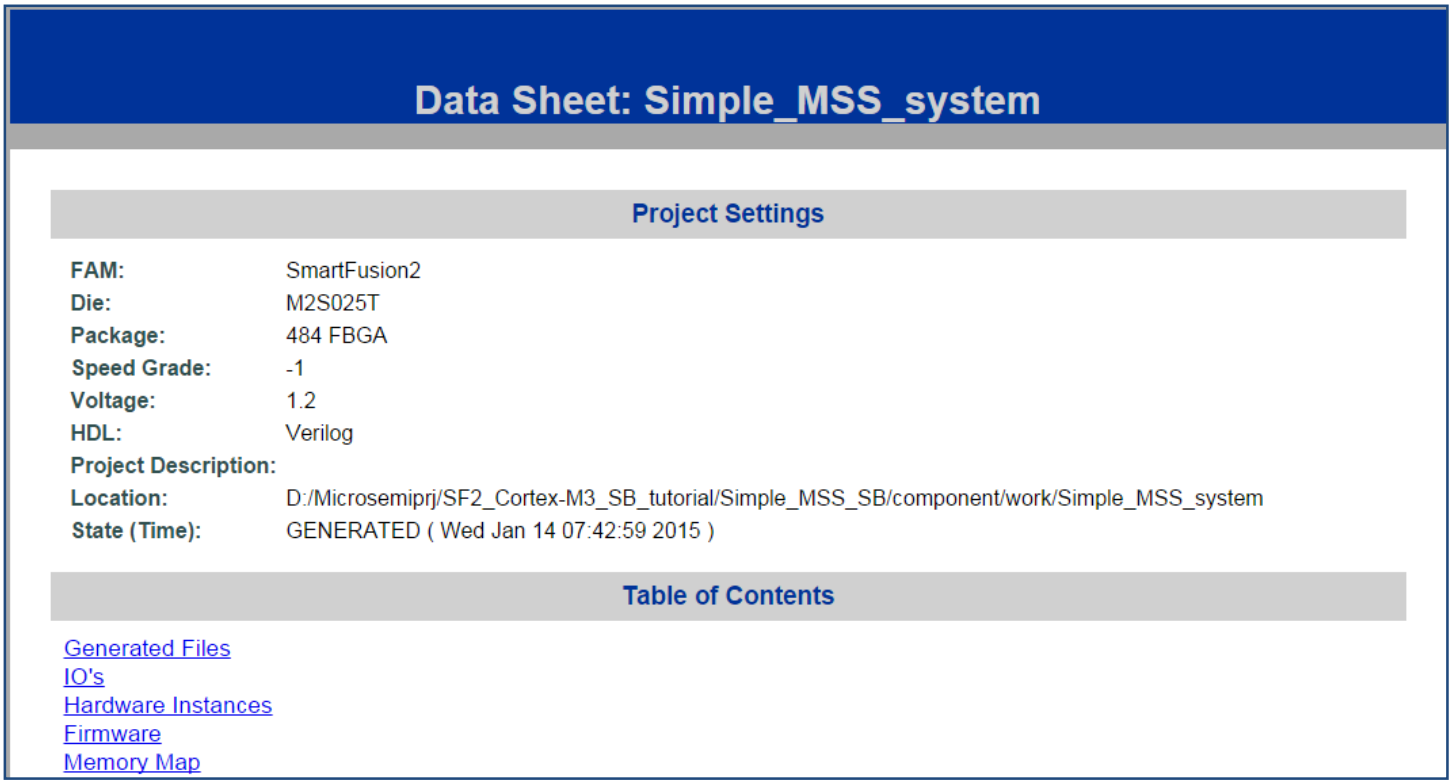


Figure 36 - Design Summary dialog box

9. Navigate to the folder listed above and double-click Simple_MSS_system.html to open the Design Summary in a web browser.



Data Sheet: Simple_MSS_system

Project Settings

FAM:	SmartFusion2
Die:	M2S025T
Package:	484 FBGA
Speed Grade:	-1
Voltage:	1.2
HDL:	Verilog
Project Description:	
Location:	D:/Microsemiprj/SF2_Cortex-M3_SB_tutorial/Simple_MSS_SB/component/work/Simple_MSS_system
State (Time):	GENERATED (Wed Jan 14 07:42:59 2015)

Table of Contents

Generated Files
IO's
Hardware Instances
Firmware
Memory Map

Figure 37 - Design Summary (M2S-EVAL-KIT result shown)

10. Scroll in the Simple_MSS_system data sheet and become familiar with the Generated Files, Firmware and Memory Map sections (click on the hyperlink at the top of the data sheet to move to the section of interest).
11. Select the Memory Map to become familiar with the locations of the peripherals.

    What is the address of MMUART_0? _____

    What is the address of the GPIO? _____

12. Close the web browser.

Step 4 – Editing the I/O Constraint file

There are multiple ways to make I/O Assignments. In this lab we will use the I/O Physical Design Constraint (PDC) file that was imported when the project was created. The I/O pdc file must be edited to match the target board.

1. The I/O constraint file that was imported when the project was created will also be visible on the Files tab under constraint > io.

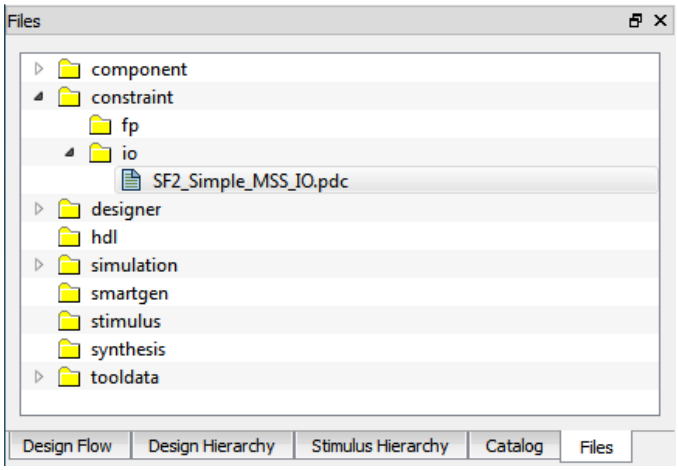


Figure 38 - I/O constraint file in Libero SoC

2. Double-click the PDC file name on the Files tab to open the file in the Libero SoC editor. Scroll in the file to become familiar with the syntax. The constraint set_iobank sets the input voltage of the bank to determine the allowable I/O standards; the constraint set_io sets the pin number and I/O specific attributes. The # symbol is a comment.

   A description of the Designer PDC constraints is available in the Libero Help (**Help > Help Topics > Implement Design > Constrain Place and Route > Assigning Design Constraints > Design Constraints Guide > Reference > Constraints by File Format > PDC Command Reference**).

3. All the assignments in the PDC file are commented. Uncomment the lines per the table below to match the target board. To un-comment a line, highlight one or more lines by dragging in the file while holding the left mouse button down, then right-click and select **Uncomment Selection**.

Board	Lines to un-comment in PDC file
SF2-STARTER-KIT-ES-2	28 – 37
SF2-STARTER-KIT	43 - 51
SF2-484-STARTER-KIT	
M2S-EVAL-KIT	57 – 61
M2S090S-EVAL-KIT	
SF2-DEV-KIT-PP	66 – 78
SF2-DEV-KIT-PP-1	
SF2-DEV-KIT	
M2S150-ADV-DEV-KIT-ES	82 - 84

Table 4 – PDC file edits

4. Save the PDC file after making modifying (**File > Save SF2_Simple_MSS_IO.pdc**).

5. Close the editor (**File > Close SF2_Simple_MSS_IO.pdc**).

   For more information regarding pin assignments for the target boards, refer to the board schematic and Users Guide which are available from the Microsemi website.

Step 5 – Synthesis and Layout

In this step you will use the push-button flow to synthesize the design with Synplify Pro, run layout and generate the programming file.

1. Expand **Create Constraints > I/O Constraints** in the Libero SoC Design Flow window. Right-click SF2_Simple_MSS_IO.pdc under I/O Constraints then right-click and select **Use for Compile**.

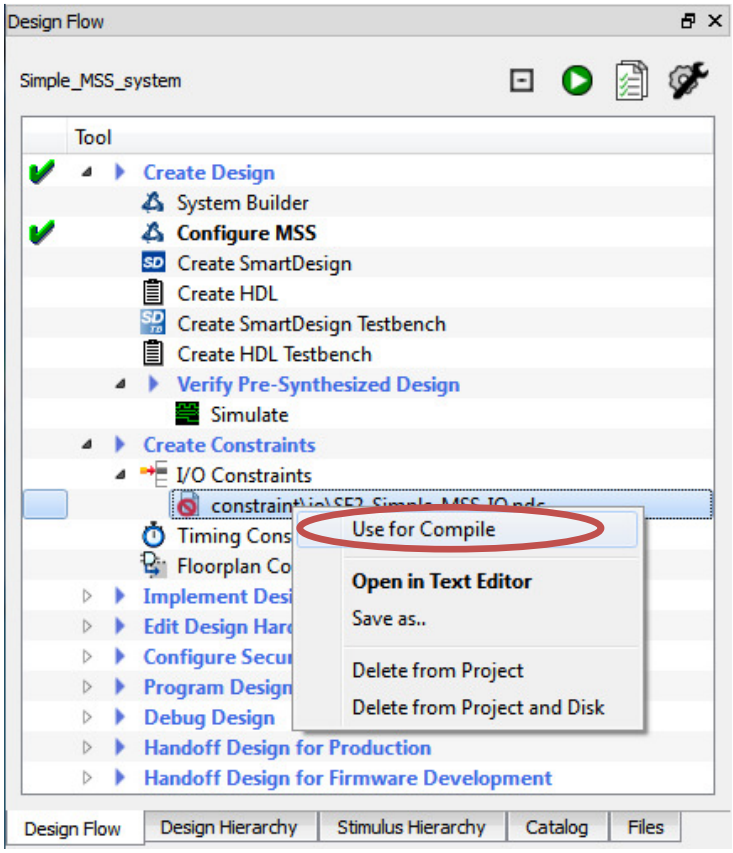


Figure 39 - Selecting the PDC constraint file in the Design Flow window

2. A green check mark will appear next to the constraint file indicating that the file will be used for compile.
3. Expand **Implement Design** on the Design Flow tab. Right-click Compile and select **Configure Options**.

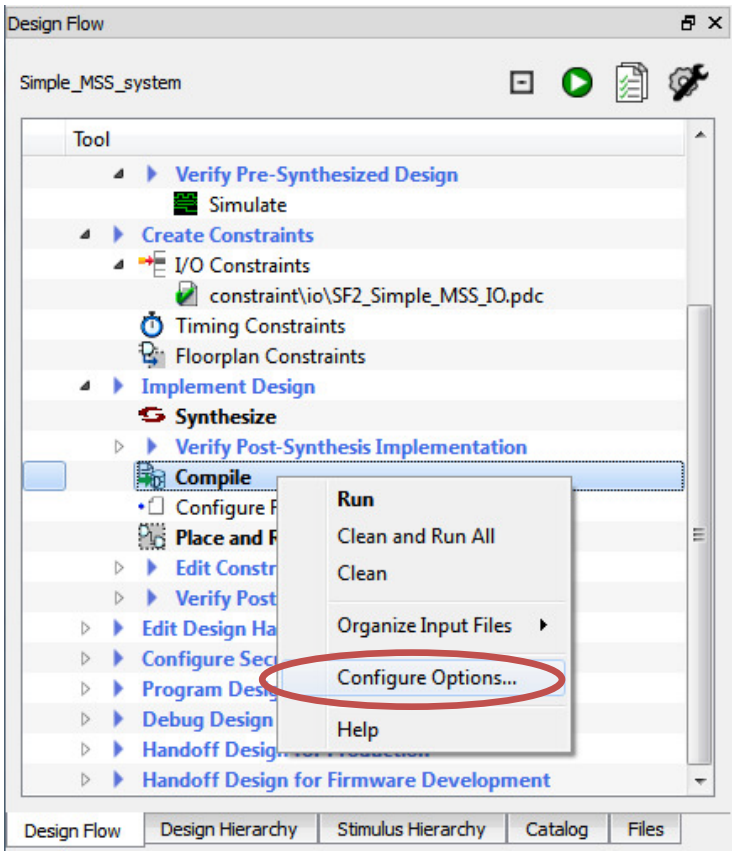


Figure 40 - Configuring Compile Options

4. The Compile Options dialog box will open. Un-check "Abort Compile if errors are found in the physical design constraints" and then click **OK**.

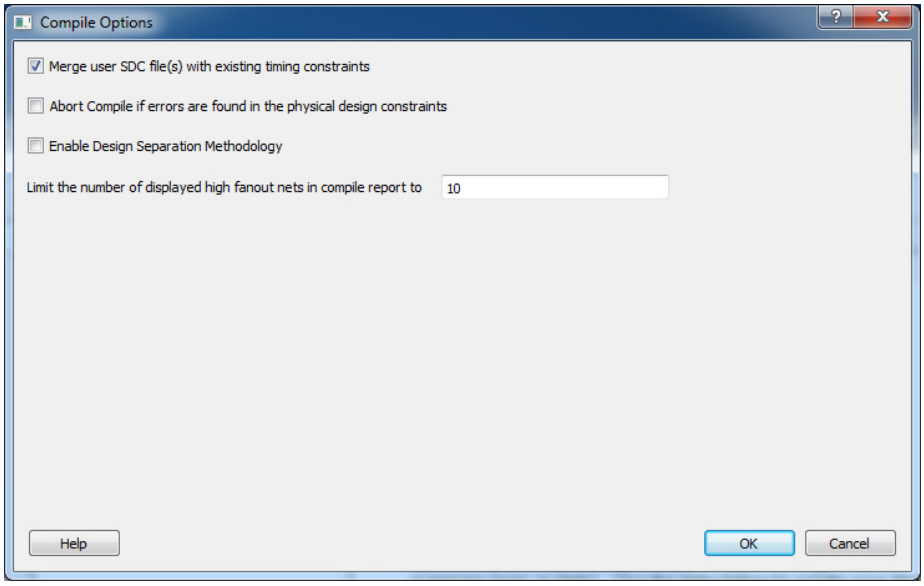


Figure 41 - Configuring the Compile Options

5. Click the Generate Programming Data icon in the Design Flow window (circled in the figure below) or select **Design > Generate Bitstream** to synthesize the design, run layout using the I/O constraints that were created and generate the programming file.

Figure 42 - Generate Bitstream icon

6. The design implementation tools will run in batch mode. Click **OK** in the Information dialog box that appears prior to generating the programming file.

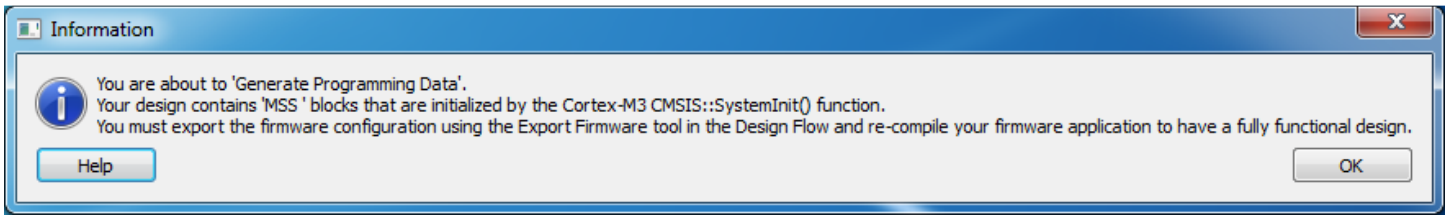


Figure 43 - Information dialog box prior to programming file generation

7. Successful completion of a design step will be indicated by a green check mark next to the Synthesize, Compile, Place and Route and Generate Bitstream under Implement Design in the Design Flow window.

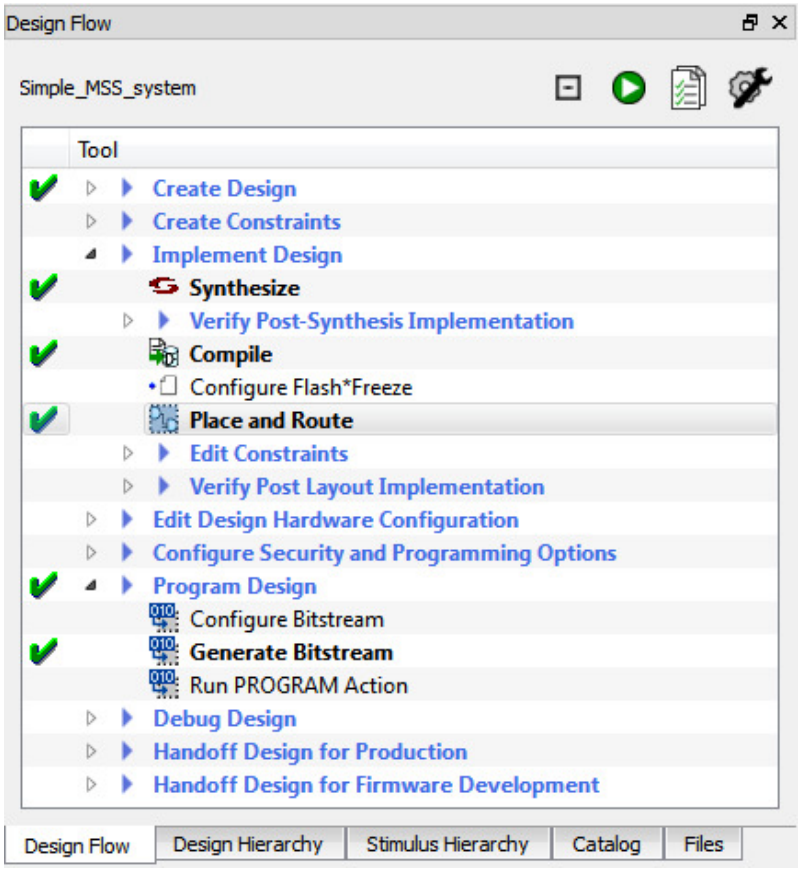


Figure 44 – Successful completion of design implementation

8. The Reports tab will display reports for the tools used to implement the design.

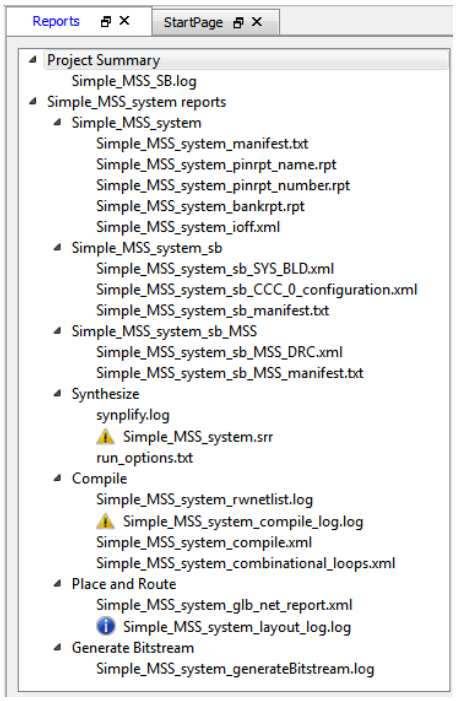


Figure 45 - Reports tab after implementing the design

9. The output files from Synplify Pro and Designer will be visible on the Libero SoC Files tab.

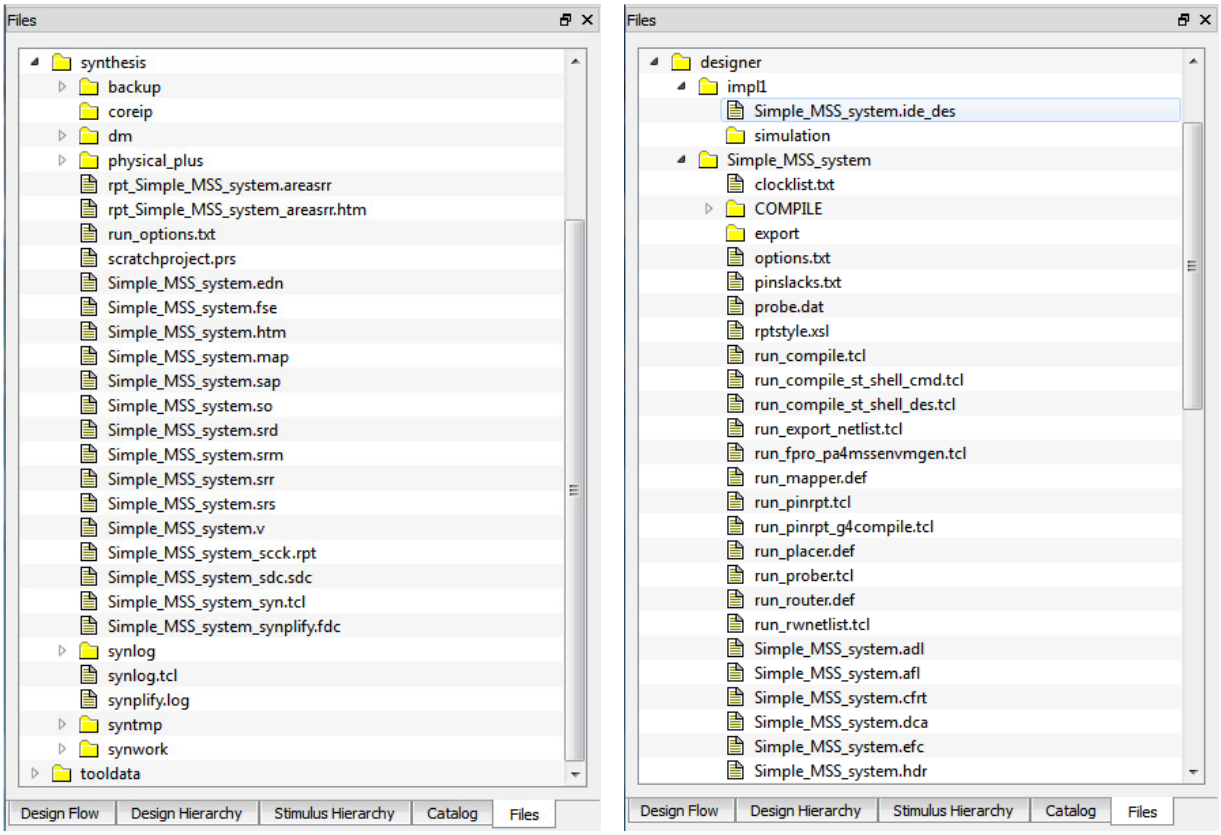


Figure 46 - Synthesis and Designer files on Libero SoC Files tab

Step 6 – Programming

In this step you will run FlashPro in batch mode to program the SmartFusion2 device on the SmartFusion2 target board.

Jumper settings for the supported target boards are shown in the tables in Appendix 1 – 4. Prior to powering up and programming the SmartFusion2 target board, confirm that the jumpers are positioned as shown in the tables and follow the instructions for powering the board.

1. Expand Program Design in the Design Flow window. Right-click **Run PROGRAM Action** and select **Run** to begin programming.

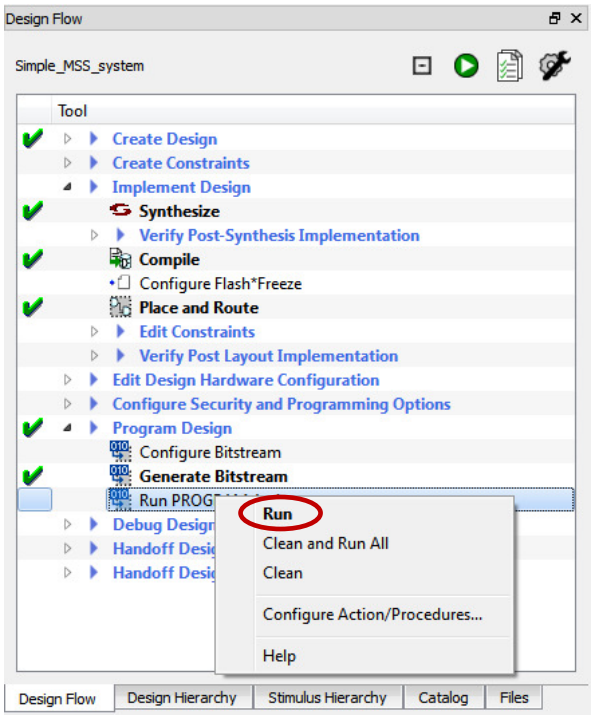


Figure 47 - Launching Programming software from Design Flow window

2. FlashPro will run in batch mode and program the SmartFusion2 device. Programming messages will be visible in the Libero SoC log window (programmer number will differ).

   Note: Do not interrupt the programming sequence; it may damage the device or the -programmer.

3. A message similar to the one below should be visible in the Libero SoC Reports view under Program Device when the device is programmed successfully (programmer number will differ; device will differ depending on the target board):

   ```
   programmer '30237' : device 'M2S025T' : Executing action PROGRAM PASSED.
   ```

4. If you are using the ES silicon a dialog box will remind you to power-cycle the board when programming is complete. Cycle the power on the board by disconnecting and re-connecting the USB cable the mini-USB connector labeled USB Power & USB UART I/F (P1).

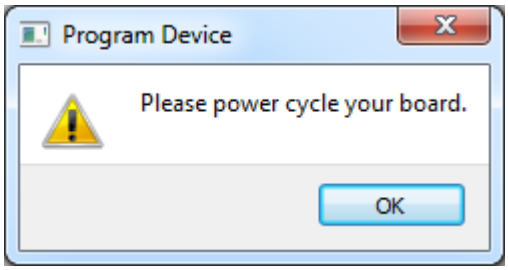


Figure 48 - Reminder to power cycle the board (ES silicon only)

5. Click **OK** to close the Program Device dialog box.

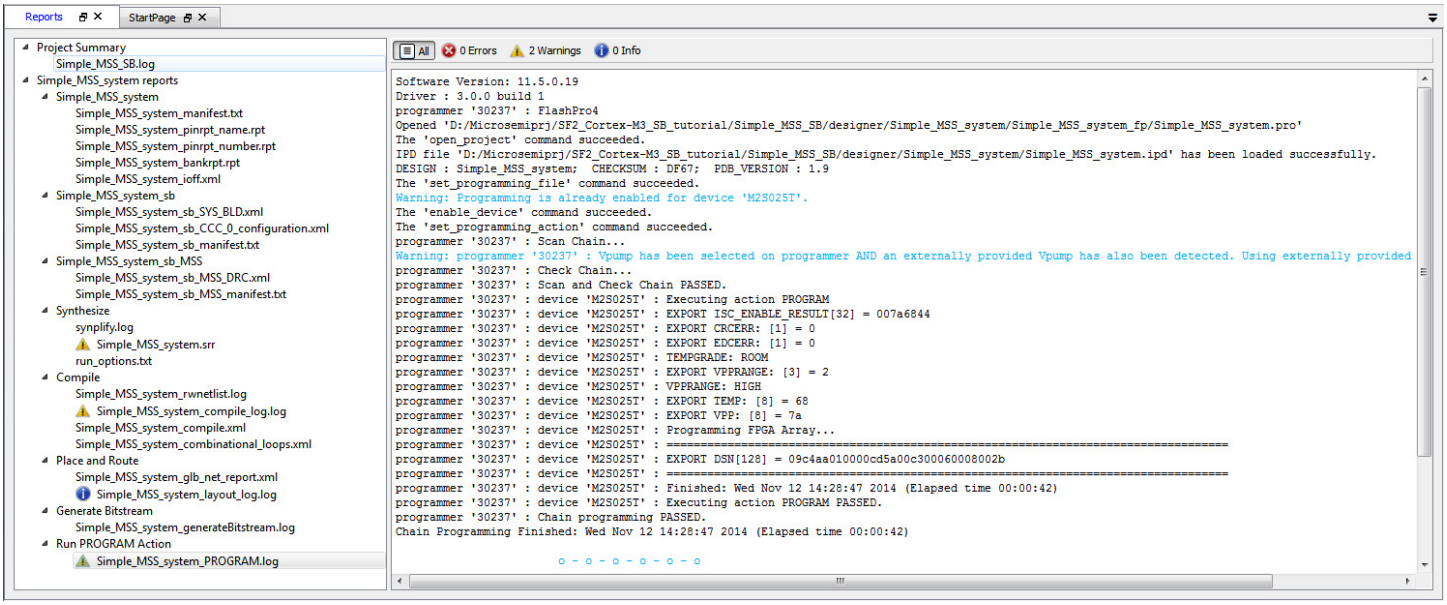


Figure 49 - Programming messages in Libero SoC Reports window (results for M2S-EVAL-KIT shown)

6. A green check mark will appear next to Program Design and Run PROGRAM Action in the Design Flow window to indicate programming completed successfully.

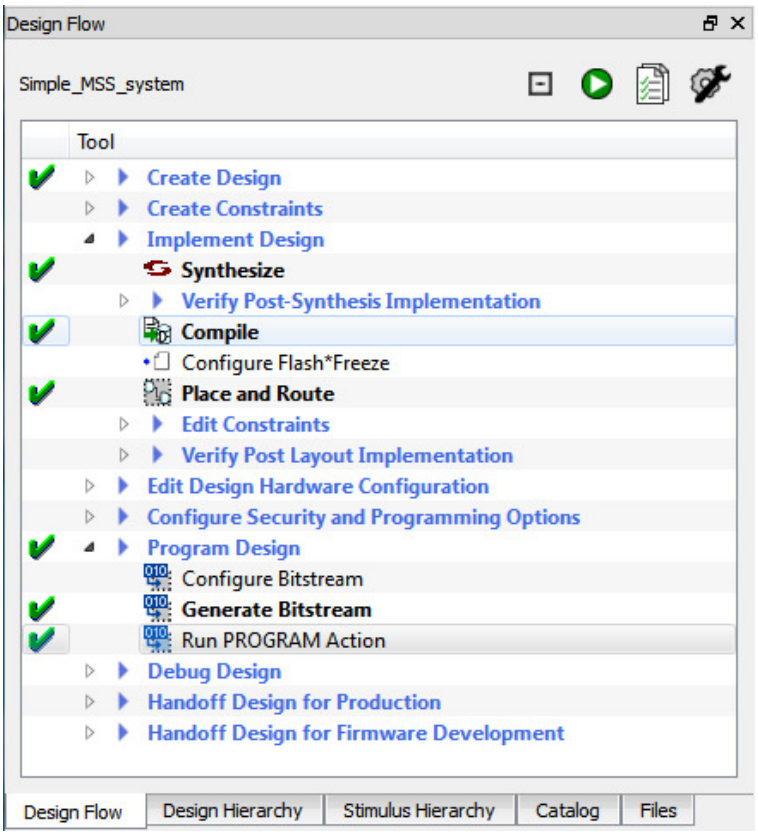


Figure 50 - Design Flow window after programming

Step 7 – Exporting firmware and generating sample projects

In this step you will export the firmware cores for the design and generate sample projects.

Exporting firmware drivers

1. Expand Handoff Design for Firmware Development on the Design Flow tab. Select Export Firmware, then right-click and select **Export Firmware…** to create the firmware drivers for the design.

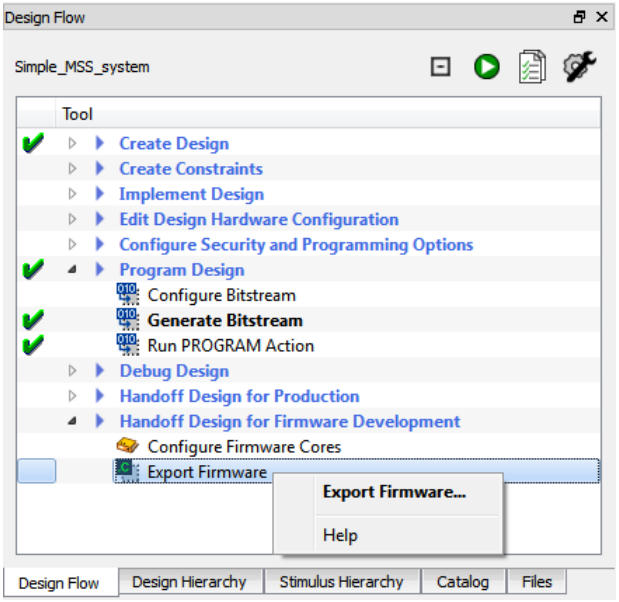


Figure 51 - Exporting firmware drivers

2. Enter the following in the Export Firmware dialog box then click **OK**:
   - Location: Accept the default location
   - Software Tool Chain: Select SoftConsole 3.4 from the pull-down menu
   - Create Project for selected toolchain: Checked (default)

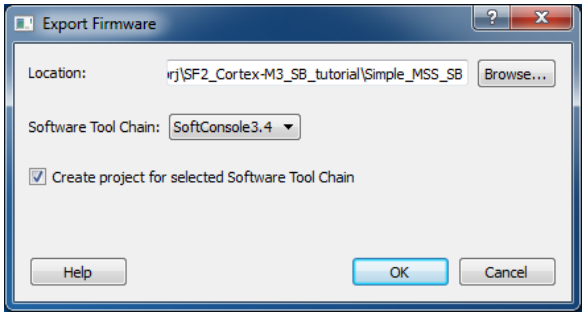


Figure 52 - Export Firmware options

3. Click **OK** in the dialog box that indicates the location of the firmware cores.

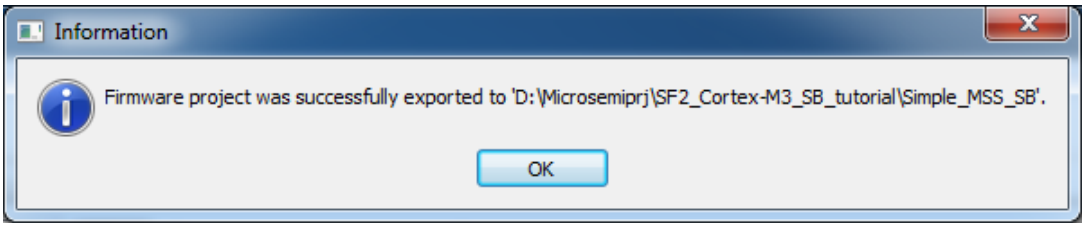
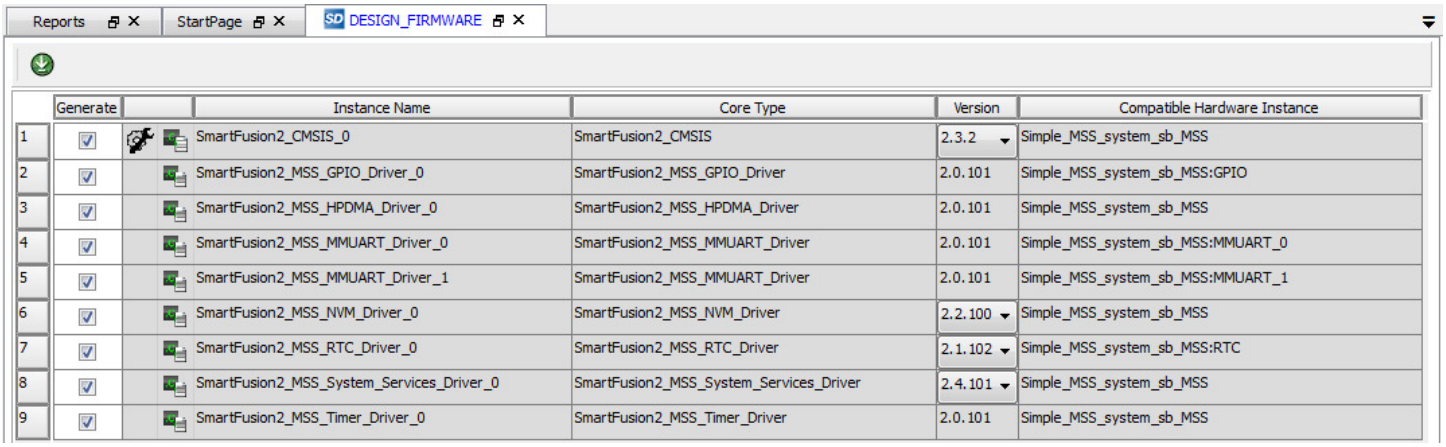


Figure 53 - Firmware driver location

Generating sample projects

4. Select Configure Firmware Cores under Handoff Design for Firmware Development on the Design Flow tab then right-click and select **Open Interactively**. The DESIGN FIRMWARE tab will open.



	Generate	Instance Name	Core Type	Version	Compatible Hardware Instance
1	☑	SmartFusion2_CMSIS_0	SmartFusion2_CMSIS	2.3.2	Simple_MSS_system_sb_MSS
2	☑	SmartFusion2_MSS_GPIO_Driver_0	SmartFusion2_MSS_GPIO_Driver	2.0.101	Simple_MSS_system_sb_MSS:GPIO
3	☑	SmartFusion2_MSS_HPDMA_Driver_0	SmartFusion2_MSS_HPDMA_Driver	2.0.101	Simple_MSS_system_sb_MSS
4	☑	SmartFusion2_MSS_MMUART_Driver_0	SmartFusion2_MSS_MMUART_Driver	2.0.101	Simple_MSS_system_sb_MSS:MMUART_0
5	☑	SmartFusion2_MSS_MMUART_Driver_1	SmartFusion2_MSS_MMUART_Driver	2.0.101	Simple_MSS_system_sb_MSS:MMUART_1
6	☑	SmartFusion2_MSS_NVM_Driver_0	SmartFusion2_MSS_NVM_Driver	2.2.100	Simple_MSS_system_sb_MSS
7	☑	SmartFusion2_MSS_RTC_Driver_0	SmartFusion2_MSS_RTC_Driver	2.1.102	Simple_MSS_system_sb_MSS:RTC
8	☑	SmartFusion2_MSS_System_Services_Driver_0	SmartFusion2_MSS_System_Services_Driver	2.4.101	Simple_MSS_system_sb_MSS
9	☑	SmartFusion2_MSS_Timer_Driver_0	SmartFusion2_MSS_Timer_Driver	2.0.101	Simple_MSS_system_sb_MSS

Figure 54 - DESIGN FIRMWARE tab

5. Confirm that none of the drivers appear in italics. If any drivers appear in italics, click the check box in the Generate column for the missing core. Click **Yes** when prompted about downloading the core.
6. Click **Yes** in the Download Required dialog box to download any firmware cores that are missing from the IP vault.

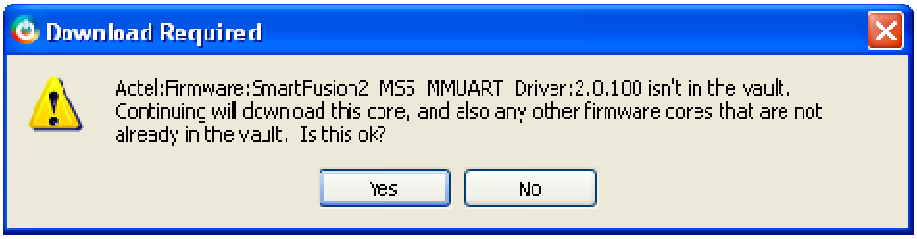


Figure 55 - Downloading missing firmware cores

7. Create the GPIO Simple Blink sample project by selecting SmartFusion2_MSS_GPIO_Driver_0 on the DESIGN FIRMWARE tab, then right clicking and selecting **Generate Sample Project > Cortex-M3 > SoftConsole > Simple Blink**.

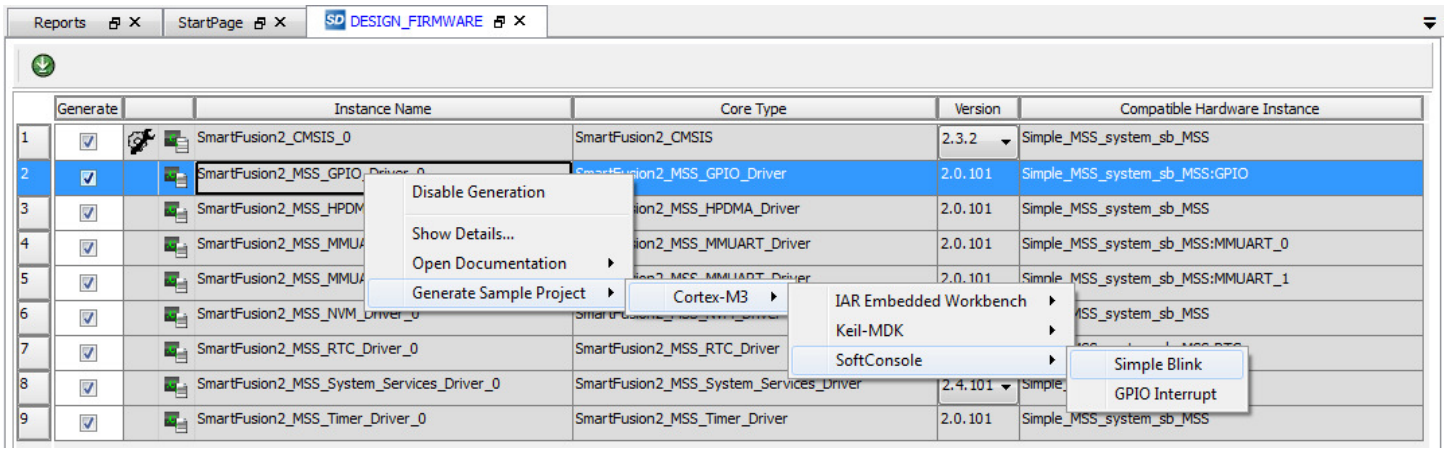


Figure 56 - Generating the Simple Blink sample project

8. Confirm the following settings in the Generate Sample Options dialog box then click **OK**:
   - Samples folder: <C or D>:\Microsemiprj\SF2_Cortex-M3_SB_tutorial\Simple_MSS_SB\SoftConsole
   - Show generation report: checked

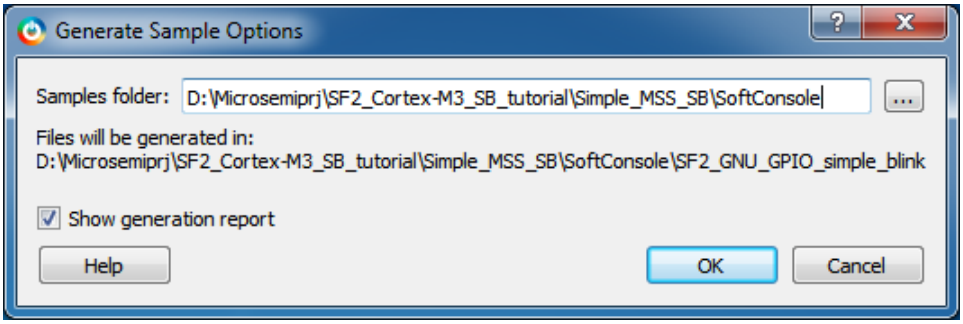


Figure 57 – Simple Blink sample project options

9. The Report dialog box will list all the files generated and the location.

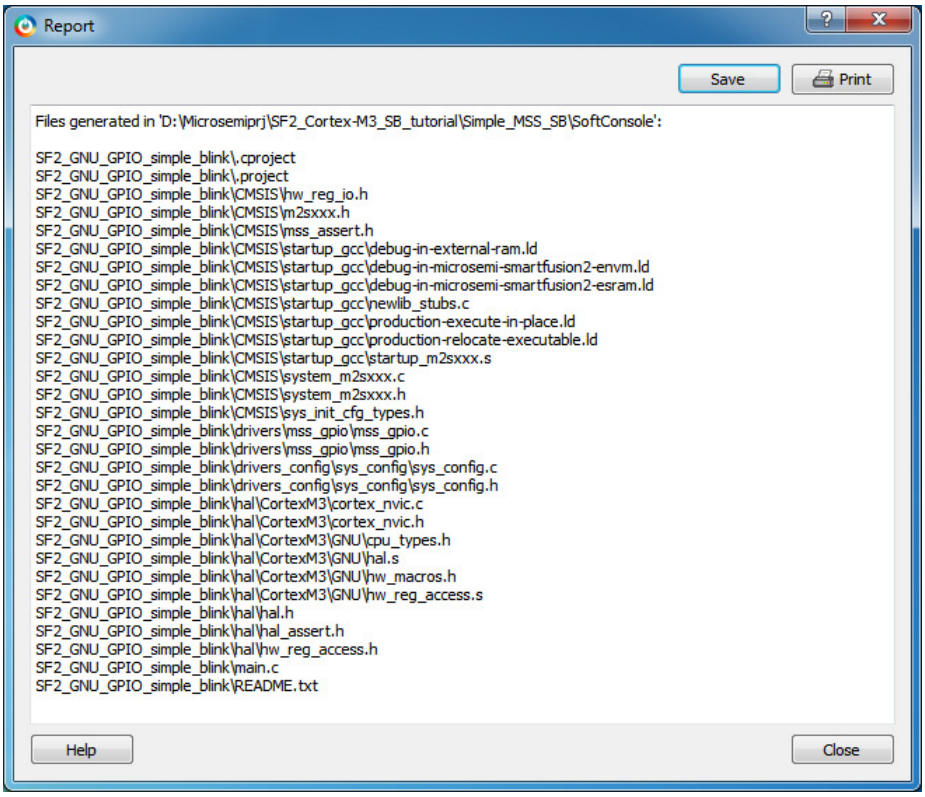


Figure 58 - GPIO Simple Blink project files

10. Click **Close** to close the Report window.
11. Create the Polled UART sample project by selecting SmartFusion2_MSS_MMUART_Driver_0 on the DESIGN FIRMWARE tab, then right clicking and selecting **Generate Sample Project > Cortex-M3 > SoftConsole > Polled UART**.
12. Confirm the following settings in the Generate Sample Options dialog box then click **OK**:
    - Samples folder: <C or D>:\Microsemiprj\SF2_Cortex-M3_SB_tutorial\Simple_MSS_SB\SoftConsole
    - Show generation report: checked

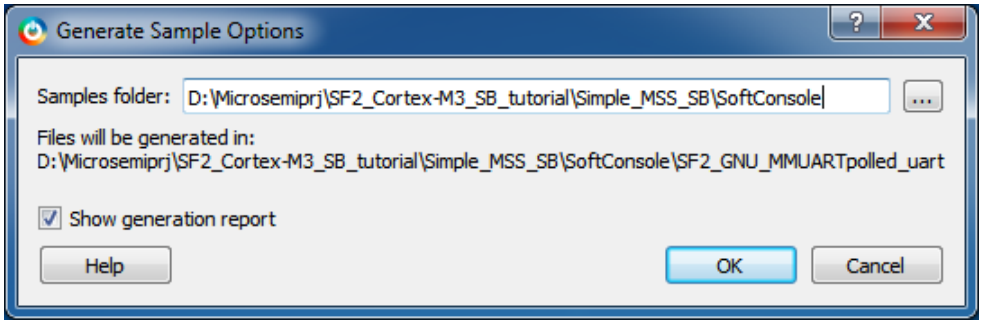


Figure 59 – Polled UART sample project options

13. The Report dialog box will list all the files generated and the location.

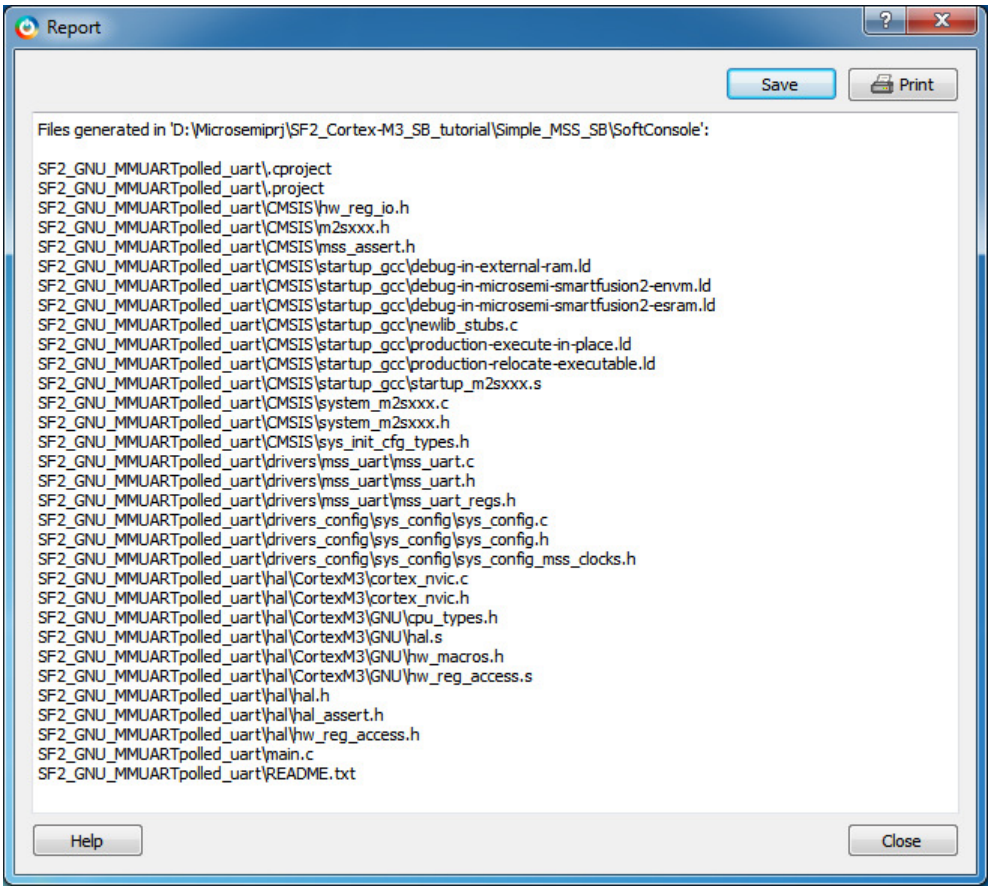


Figure 60 - polled_uart project files

14. Click **Close** to close the Report window.
15. Close the DESIGN FIRMWARE tab (**File > Close DESIGN FIRMWARE**). Select **Yes** if prompted about saving changes to DESIGN_FIRMWARE.
16. The firmware drivers will be visible on the Libero SoC Files tab. The Project for selected toolchain and the sample projects will be visible on the Files tab in the SoftConsole folder. If the projects are not visible, select **View > Refresh Design Hierarchy** from the Libero SoC menu.

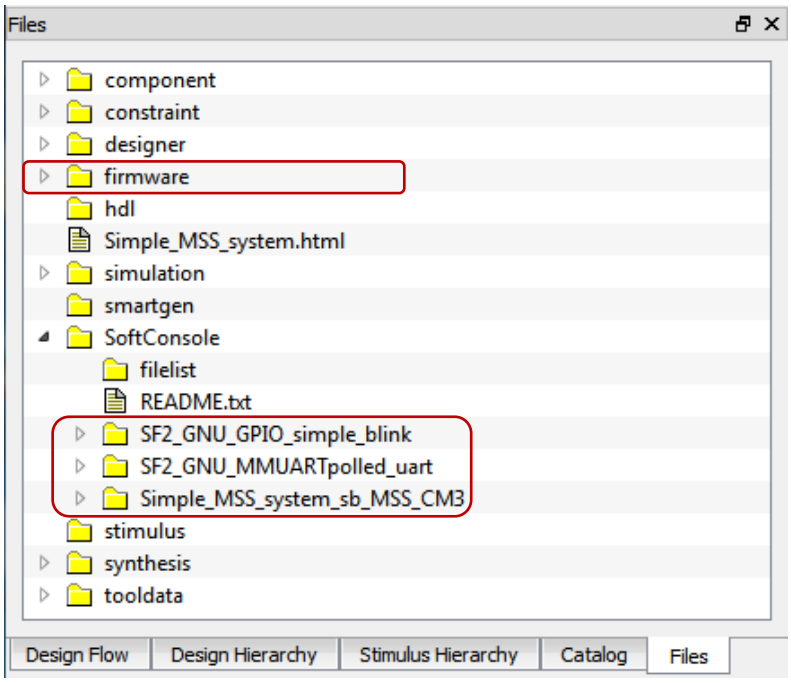


Figure 61 - Sample projects on Libero SoC Files tab

Step 8 - Running the GPIO simple blink application

In this step you will launch SoftConsole v3.4 and run the Simple Blink application.

1. Click **Start > Programs > Microsemi SoftConsole v3.4 > Microsemi SoftConsole IDE v3.4.0.5**.
2. Click the Browse button in the Workspace Launcher dialog box and browse to the <C: or D:>\Microsemiprj\SF2_Cortex-M3_SB_tutorial\Simple_MSS_SB\SoftConsole\Simple_MSS_MSS_CM3 folder and Click **OK**.

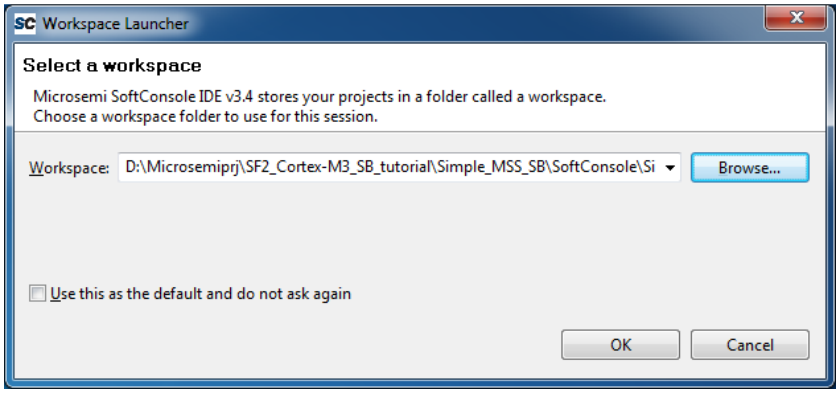


Figure 62 - SoftConsole Workspace Launcher

3. SoftConsole will open with two projects visible in the Project Explorer view (Simple_MSS_system_sb_MSS_CM3_app and Simple_MSS_system_sb_MSS_CM3_hw_platform). Ignore any warning messages that appear in the Problems view.

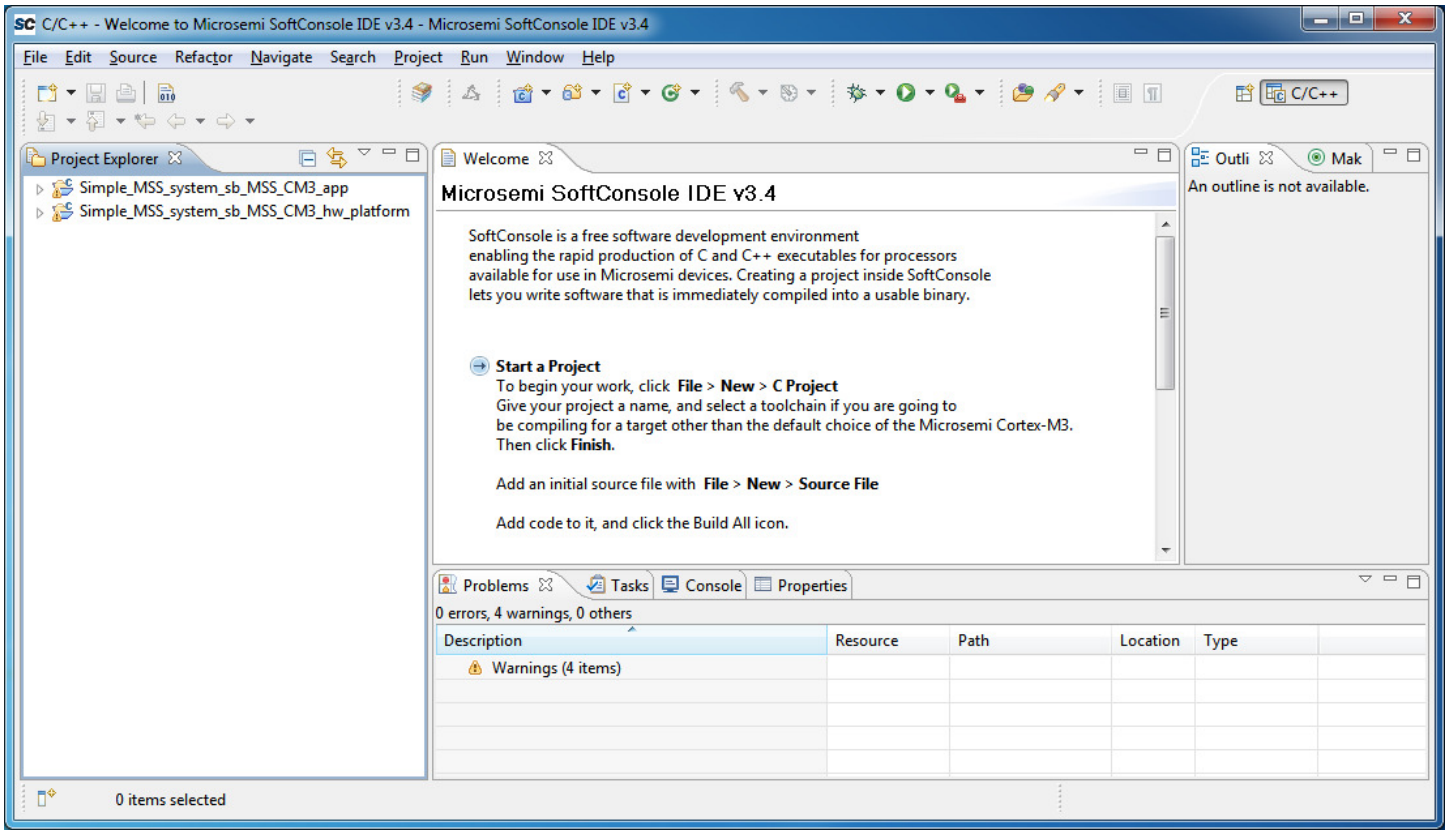


Figure 63 – SoftConsole 3.4

The Simple_MSS_system_sb_MSS_CM3_app project is the starting point for user applications; the file main.c is a template file that you can use to create your firmware application.

The Simple_MSS_system_sb_MSS_CM3_hw_platform project contains the firmware drivers that were created by SmartDesign and the platform header file (Simple_MSS_system_hw_platform.h) which contains the addresses of any fabric peripherals.

4. Expand the folders to become familiar with the contents of each project (Figure 65).
5. Import the main.c file from GPIO simple_blink sample project created earlier by clicking **File > Import** from the SoftConsole menu. The Import dialog box will open.
6. Expand the General category in the Import dialog box and select **File System** and then click **Next**. The Import dialog box will open.
7. Enter the following in the Import Projects dialog box:

   - From directory:

   <C: or D:>\Microsemiprj\SF2_Cortex-M3_SB_tutorial\Simple_MSS_SB\SoftConsole\SF2_GNU_GPIO_simple_blink
   - Select main.c
   - Into Folder: Simple_MSS_system_sb_MSS_CM3_app

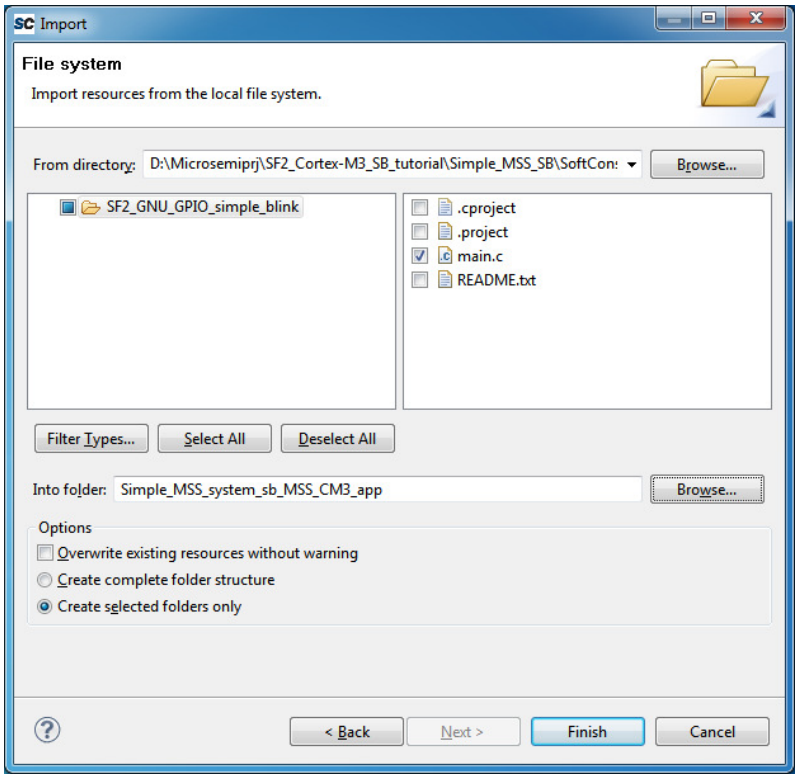


Figure 64 - Importing the main.c file from the simple_blink project

8. Click **Finish**. Select **Yes** when prompted about overwriting main.c in Simple_MSS_system_sb_MSS_CM3_app. The file will be visible in the SoftConsole Project Explorer window as shown below.

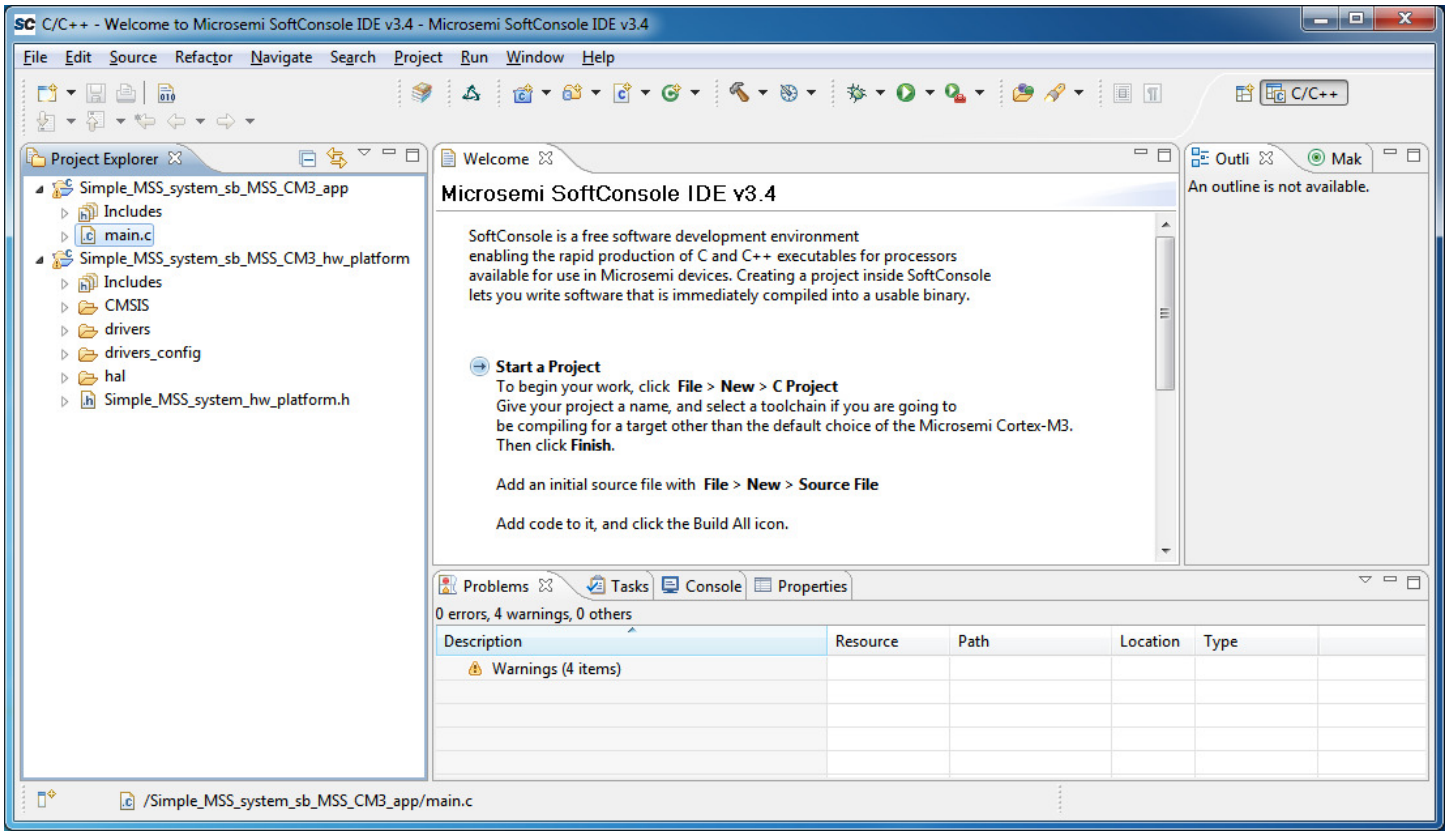


Figure 65 - SoftConsole projects

9. Double click main.c in the Project Explorer window to open the file in the SoftConsole C/C++ editor. Scroll through the file to become familiar with it. The comments at the top of the file describe what the program does.

Checking Project Settings and performing a build

10. Select the project name (Simple_MSS_system_sb_MSS_CM3_app) in the SoftConsole Project Explorer View.
11. Select **Project > Properties**. The project properties dialog box will open.
12. Expand **C/C++ Build** in the properties dialog box and select **Settings**. Confirm that the Debug configuration is selected.

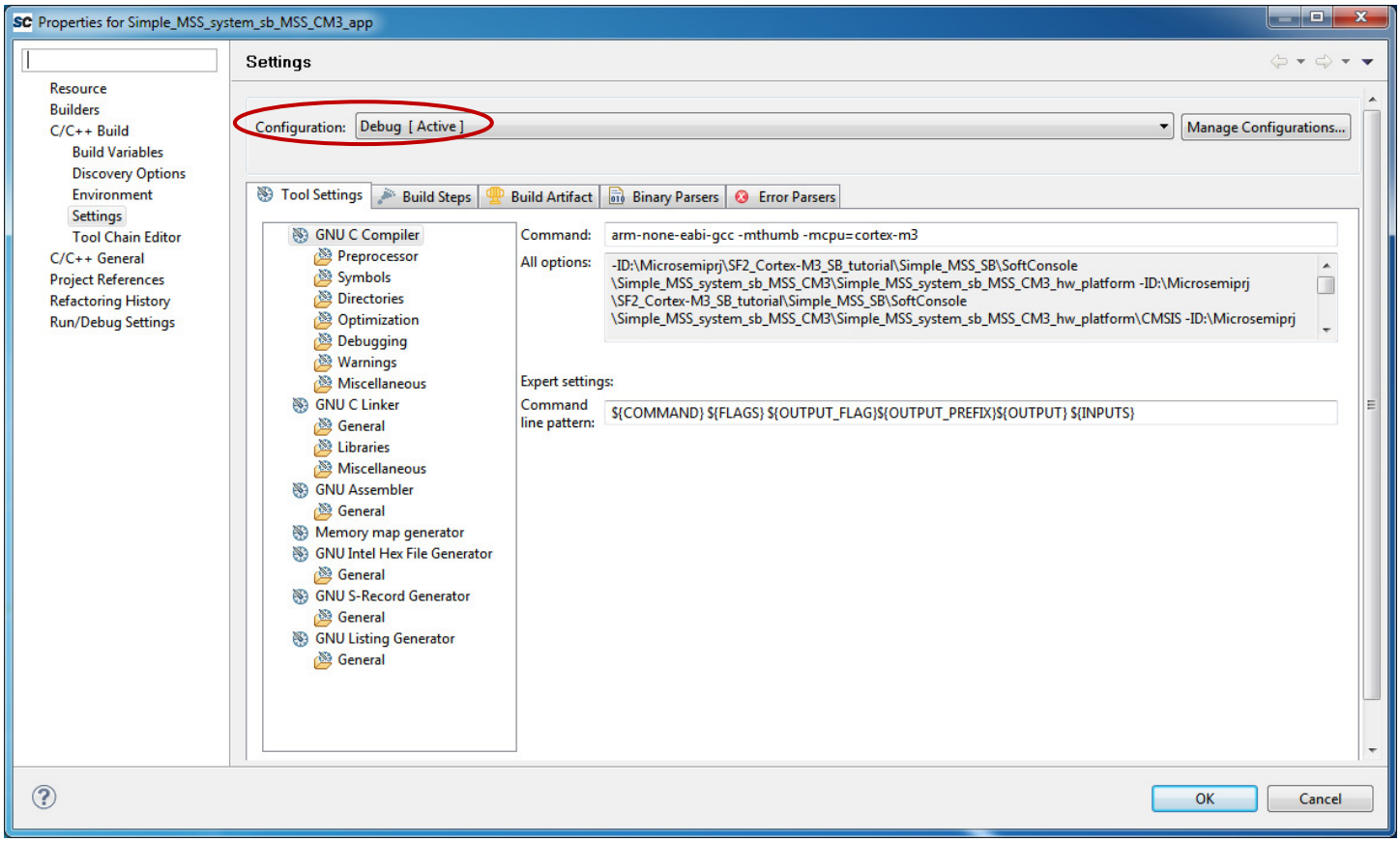


Figure 66 - SoftConsole project settings

13. Select **Miscellaneous** under GNU C Linker. Confirm that the Linker flags field contains
    -T../../Simple_MSS_system_sb_MSS_CM3_hw_platform/CMSIS/startup_gcc/debug-in-microsemi-smartfusion2-esram.ld

    This command uses the linker script that builds an executable that will run from the SmartFusion2 embedded SRAM.

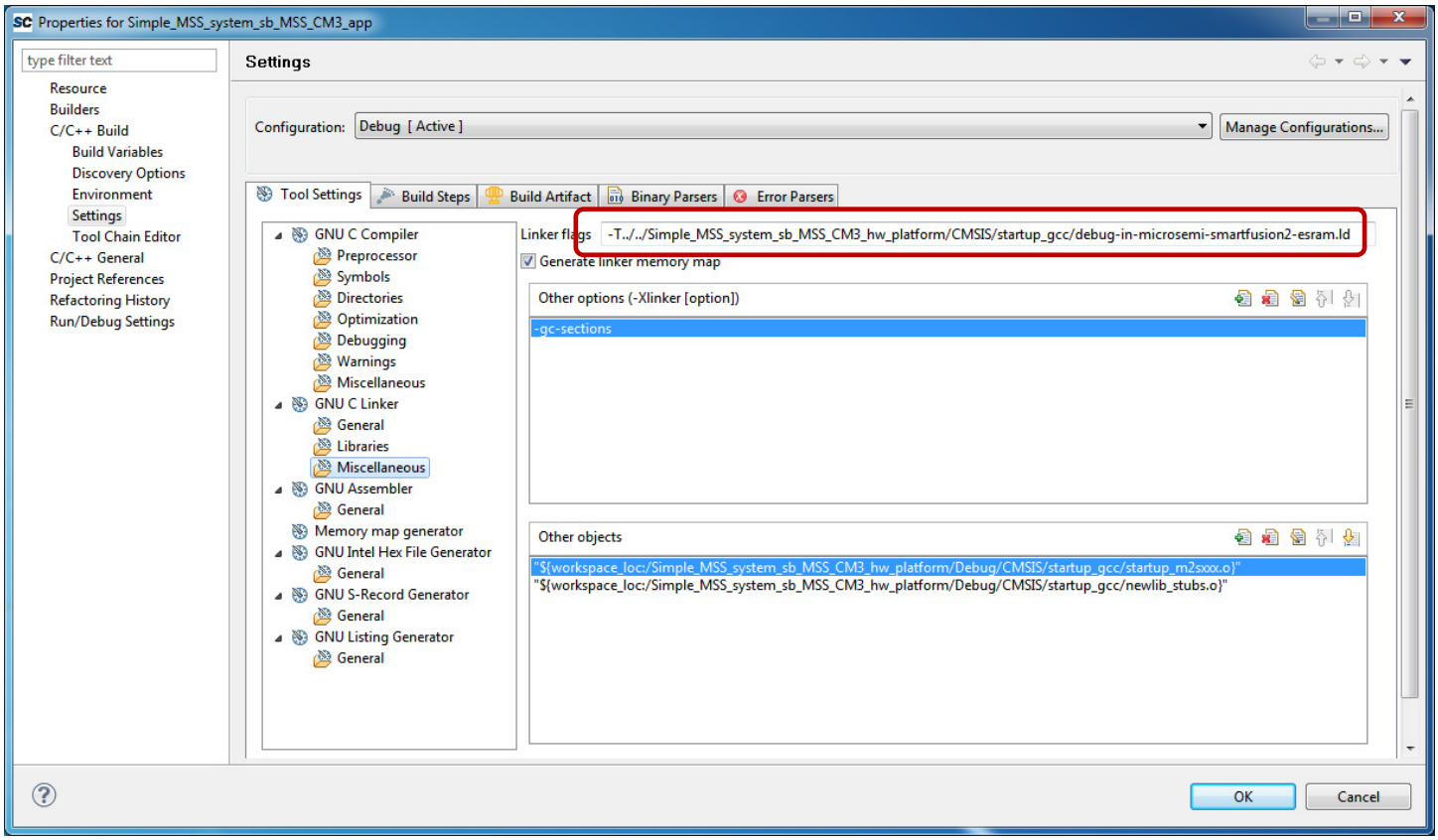


Figure 67 - Confirming the linker script setting

14. After confirming the linker script setting, click **OK** to close the Project Settings dialog box.

15. Perform a build by selecting **Project > Build All** or by clicking the Build All icon ( ) on the SoftConsole toolbar.
16. Confirm that there are no errors in the Problems View.

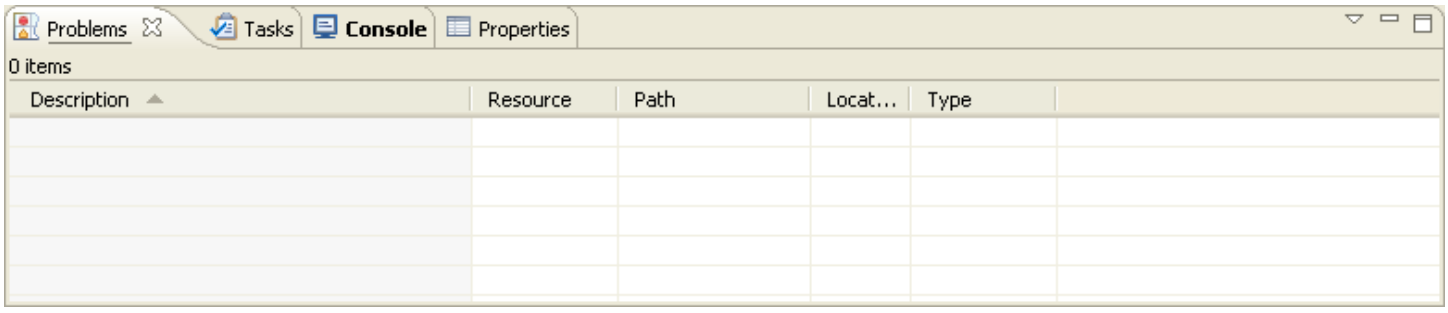


Figure 68 - Problems View after successful build

Debugging with SoftConsole

In this step you will use the SoftConsole debugger to run the GPIO application on the SmartFusion2 Cortex-M3.

Before using the On-chip Debugger (OCD), the debug target must be configured.

17. Select Simple_MSS_system_MSS_CM3_app in the SoftConsole Project Explorer View.
18. Select **Run > Debug Configurations…** from the SoftConsole menu. The Debug Configurations dialog will open.

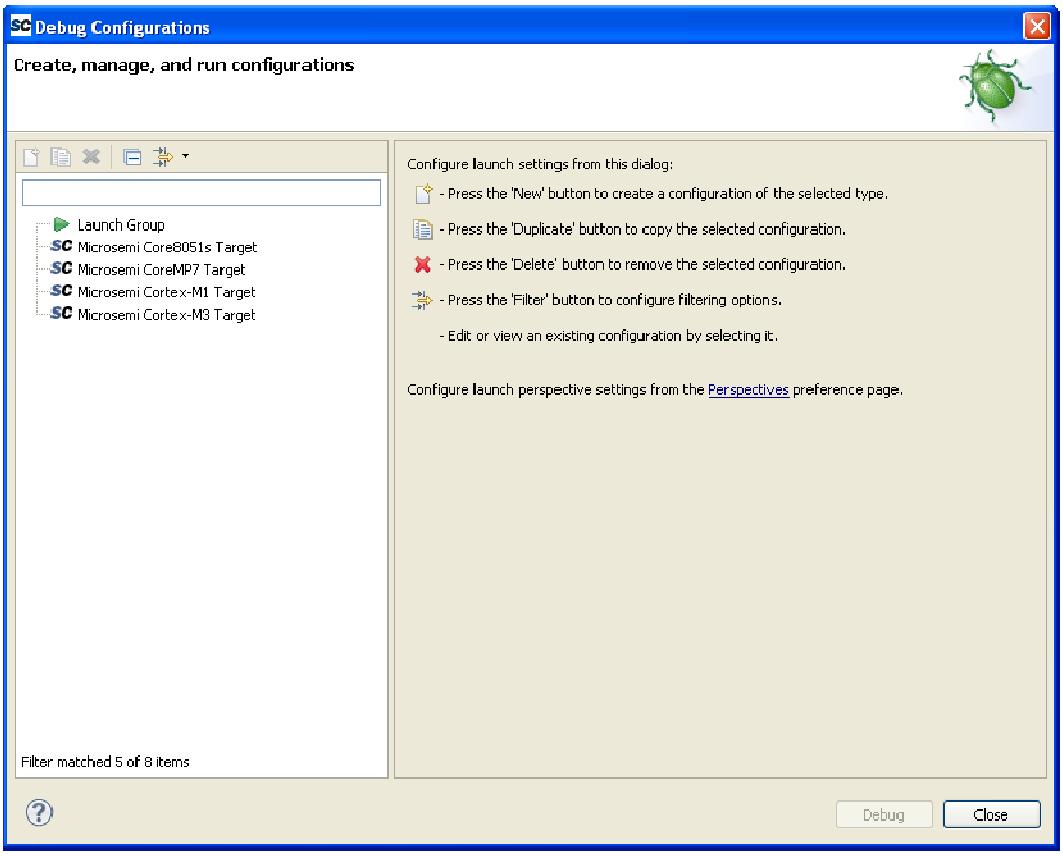


Figure 69 - SoftConsole Debug dialog

19. Select **Microsemi Cortex-M3 Target** then right-click and select **New**. A debug target named Simple_MSS_system_sb_MSS_CM3_app Debug will be visible.
20. Confirm that the following appear on the Main tab in the Debug window (see figure below):
    - Name: Simple_MSS_system_sb_MSS_CM3_app Debug
    - Project: Simple_MSS_system_sb_MSS_CM3_app
    - C/C++ application: Debug\Simple_MSS_system_sb_MSS_CM3_app

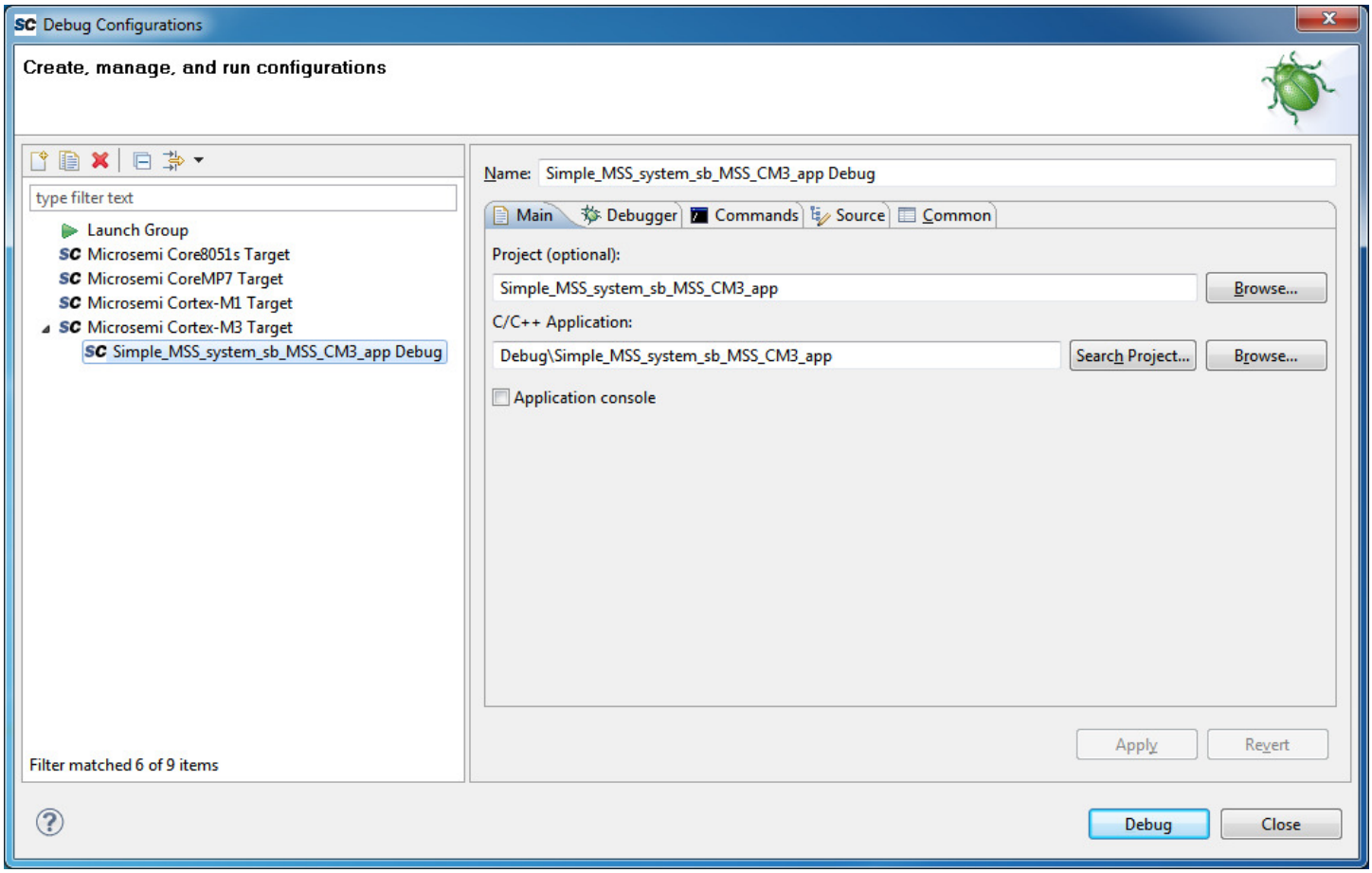


Figure 70 - Debug target for the Simple_MSS_system_MSS_CM3_app project

21. Select the Common tab. Check the box next to **Debug** in the section titled Display in favorites menu then click **Apply**. This will allow you to launch the debugger by clicking the debug icon ( ) in the SoftConsole toolbar.

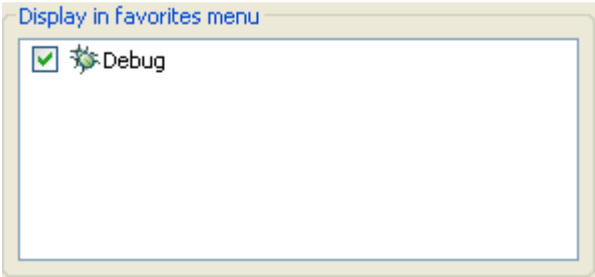


Figure 71 - checking Debug in Display in favorites

22. Click **Debug** to launch the Debugger. Click **Yes** in the **Confirm Perspective Switch** dialog box. The SoftConsole Debug perspective will open as shown in the figure below.
23. The code will be automatically downloaded to the SmartFusion2 starter kit board. The program will be suspended at the first line of main.c.

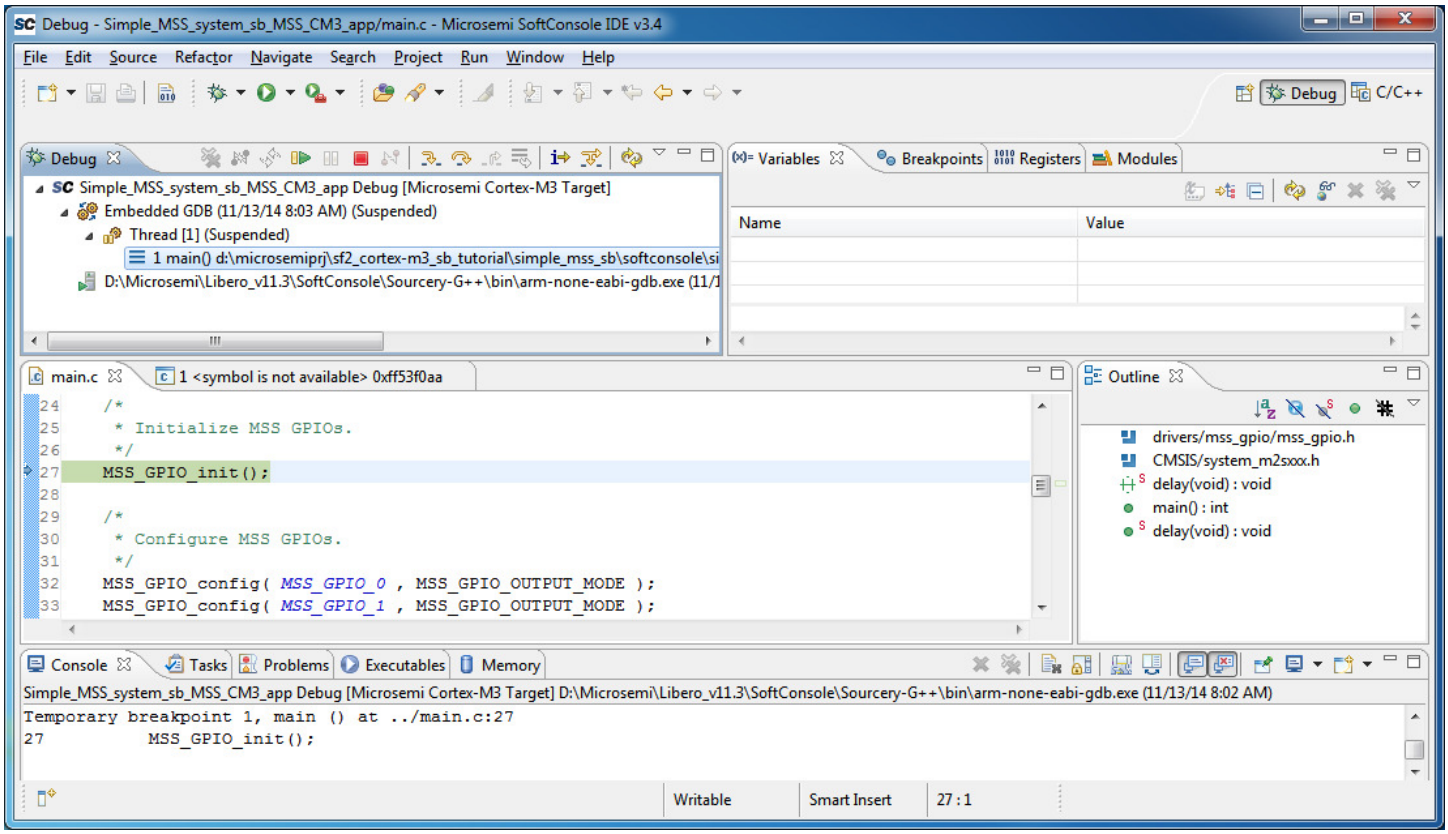


Figure 72 - SoftConsole Debug perspective

Running the simple_blink Application

24. Start the Cortex-M3 software application by clicking **Run > Resume** from the SoftConsole menu. The table below shows which LEDs will blink for the different SmartFusion2 target boards.

Target Board	LEDs
Starter kit (All)	DS4, DS3
Evaluation kit (All)	F4, E1
Development kit (All)	LED2, LED1
Advanced Development kit	DS1, DS0

Table 5 – Reference designators

25. Suspend the software application by clicking **Run > Suspend** from the SoftConsole menu.
26. Select the **Registers** view on the upper right window pane to view the value of the Cortex-M3 internal registers as shown in the figure below. Your values may differ.

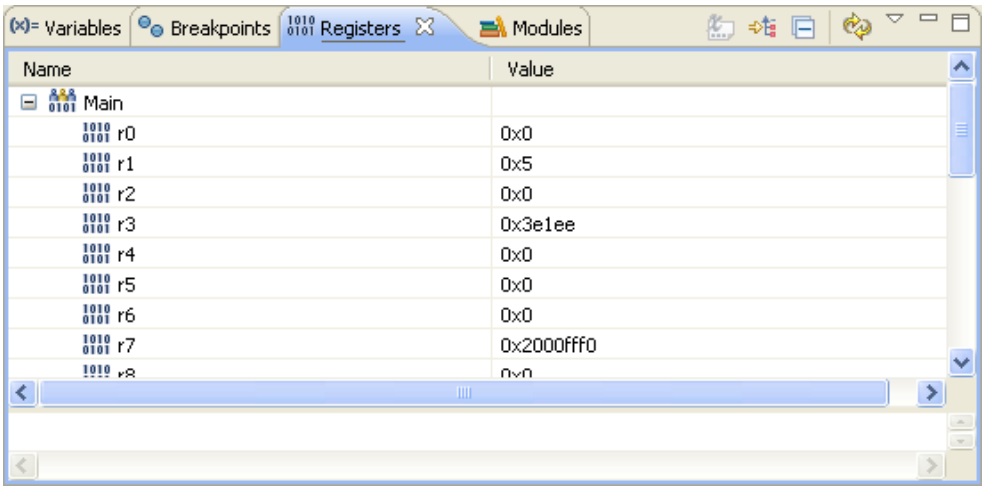


Figure 73 - Cortex-M3 registers

27. Choose **Window > Show View > Disassembly** to display the assembly level instructions. The Assembly window will open on the right side in the middle of the Debug perspective as shown in the figure below.

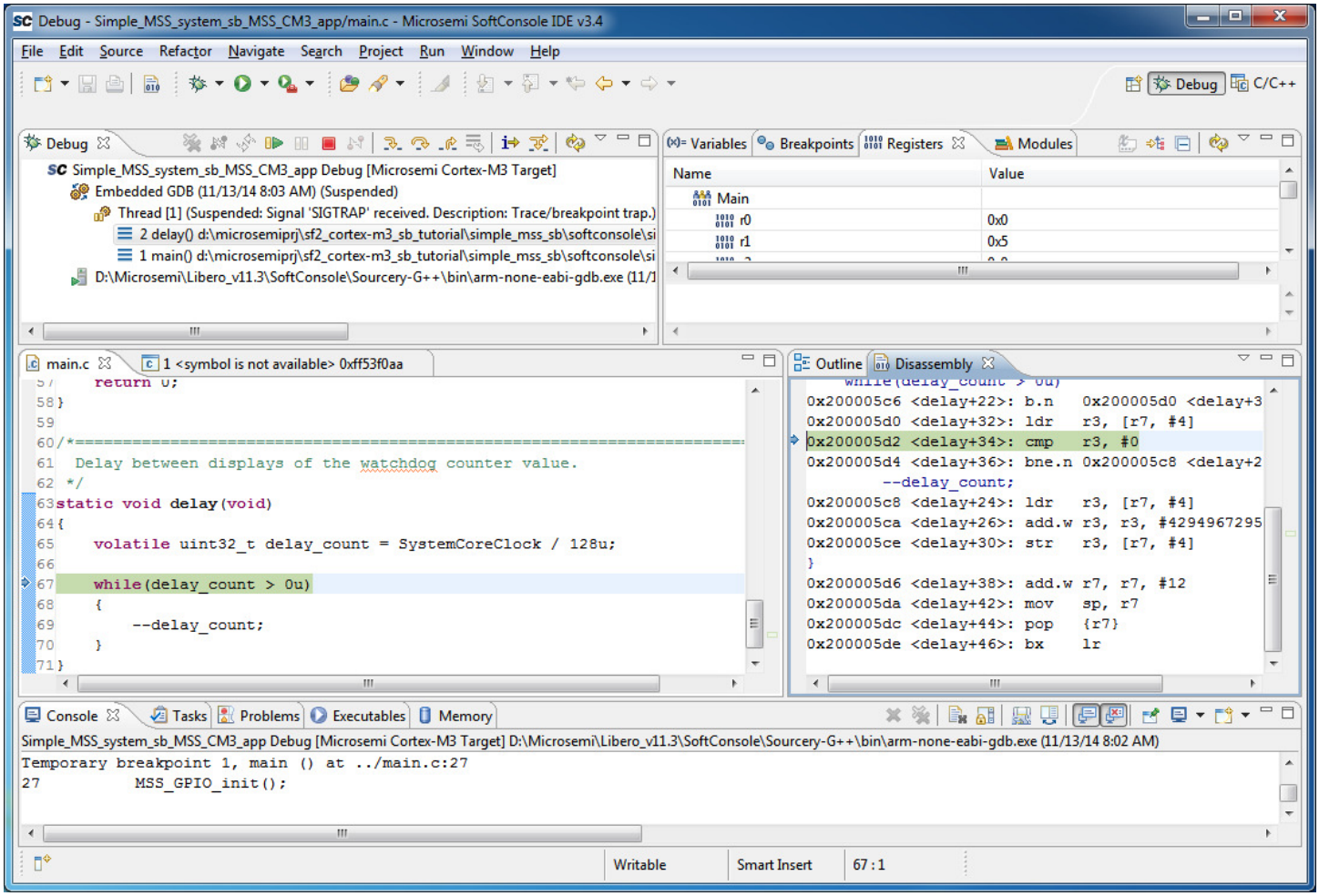


Figure 74 – SoftConsole Disassembly view

28. Scroll in **main.c** and locate the delay function on line 63.

```
Welcome  main.c
56
57      return 0;
58  }
59
60  /*==========================================================================
61    Delay between displays of the watchdog counter value.
62   */
63  static void delay(void)
64  {
65      volatile uint32_t delay_count = SystemCoreClock / 128u;
66
67      while(delay_count > 0u)
68      {
69          --delay_count;
70      }
71  }
72
```

Figure 75 - delay_count variable in main.c

29. Changing the value of the variable delay_count on line 65 will change the blink rate of the LED. Change the value by dividing SystemCoreClock by a larger or smaller value then save the file (**File > Save**).
30. Perform a build by selecting **Project > Build All**. Select the Problems View and confirm that there are no errors.
31. Select Embedded GDB under the Debug tab in the upper left corner of SoftConsole. Right-click and choose **Terminate and Relaunch** to stop the debugger and download the new application.

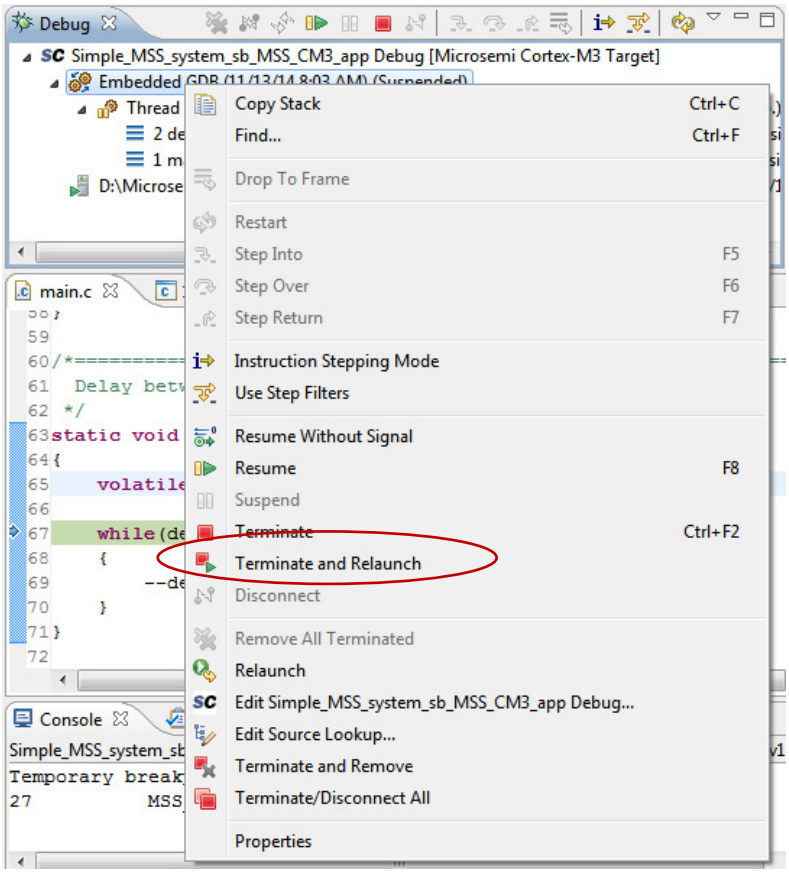


Figure 76 – Terminating and re-launching the debugger

32. Start the Cortex-M3 software application by clicking **Run > Resume** from the SoftConsole menu. The LEDs will blink at a different rate.

33. Repeat steps 30 - 32 to try different values for delay_count and observe the results. When finished, terminate the debugger.
34. If time permits, modify main.c to make the LEDs toggle instead of blinking together.

    Hint: use the GPIO driver function MSS_GPIO_set_outputs (see mss_gpio.h for usage). Setting the GPO to 1 turns the LED on, setting the GPO to 0 turns the LED off.

35. Repeat the steps to perform a build and re-launch the debugger.
36. When finished, terminate the application by selecting Embedded GDB under the Debug tab in the upper left corner of SoftConsole. Right-click and choose **Terminate and Remove** to stop the debugger.

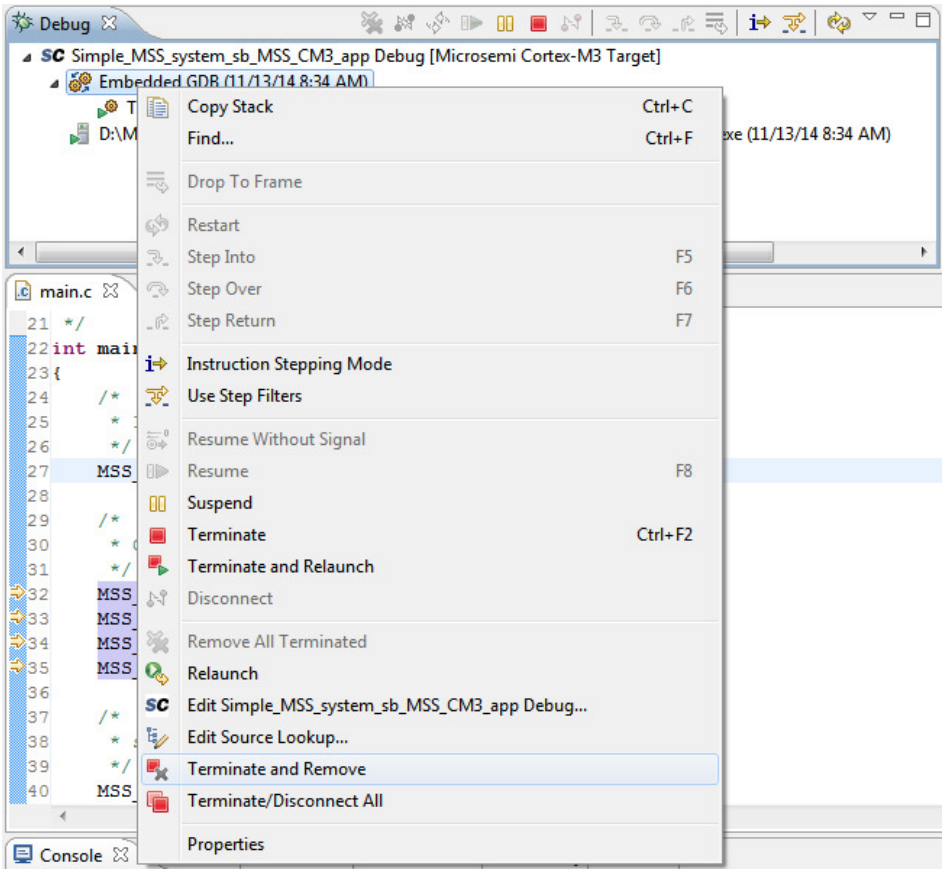


Figure 77 – Terminating the application

37. Close the Debug perspective (**Window > Close Perspective**).

Step 9 - Running the polled_uart application

In this section you will import the main.c file from the polled_uart sample project that was created earlier and run the application.

1. Select **File > Import** from the SoftConsole menu. The Import dialog box will open.
2. Expand the General category in the Import dialog box and select **File System** and then click **Next**.

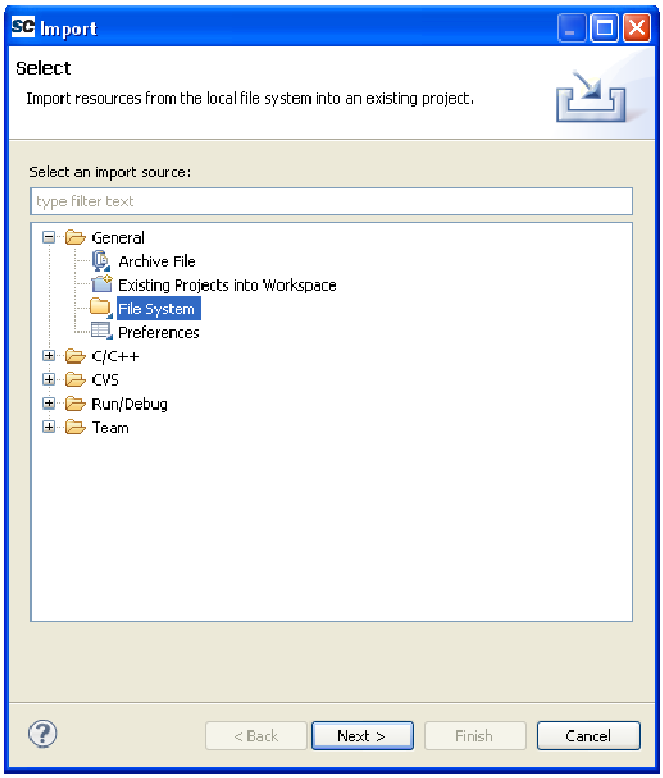


Figure 78 - SoftConsole Import dialog box

3. Enter the following in the Import dialog box then click **Finish**:
   - From directory:
   <C: or D:>\Microsemiprj\SF2_Cortex-M3_SB_tutorial\Simple_MSS_SB\SoftConsole\SF2_GNU_MMUARTpolled_uart
   - Select main.c
   - Into Folder: Simple_MSS_system_sb_MSS_CM3_app

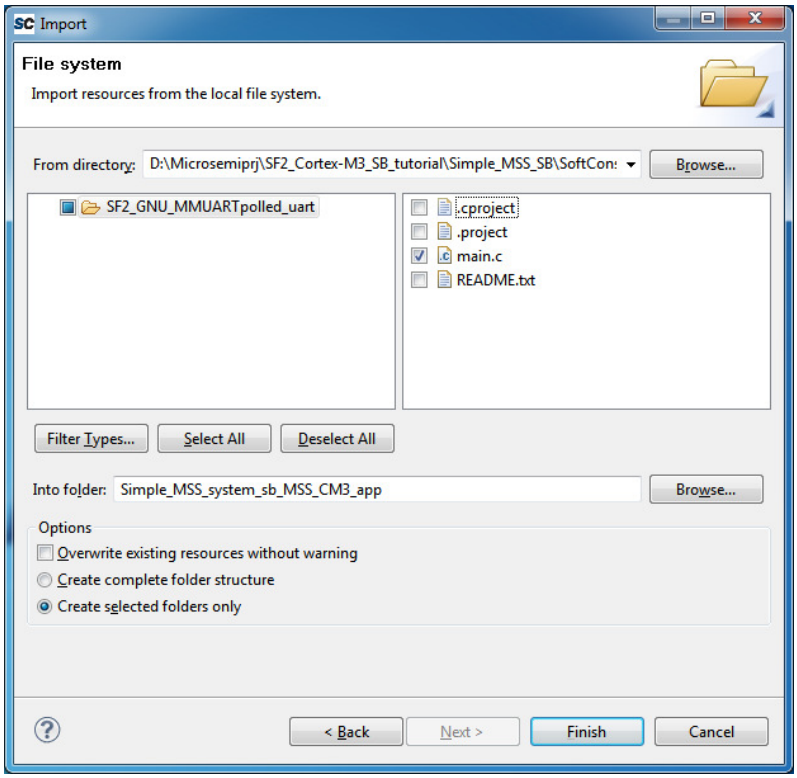


Figure 79 - Importing main.c from the polled_uart project

4. Click **Finish**. Select **Yes** when prompted about overwriting main.c in Simple_MSS_system_sb_MSS_CM3_app.
5. Double click main.c in the Project Explorer window to open the file in the SoftConsole C/C++ editor. Scroll through the file to become familiar with it. The comments at the top of the file describe what the program does.

If the target board is the SmartFusion2 Evaluation kit (M2S-EVAL-KIT) or the SmartFusion2 Security Evaluation Kit (M2S090S-EVAL-KIT), complete step 6 below. Skip step 6 for all other target boards.

6. Change line 20 of main.c to:
   ```
   mss_uart_instance_t * const gp_my_uart = &g_mss_uart1;
   ```
   This change will use MMUART_1 instead of MMUART_0. Save the file after editing.
7. Select Simple_MSS_system_sb_MSS_CM3_app in the SoftConsole Project Explorer. Perform a build by selecting **Project > Build All** or by clicking the Build All icon.
8. Confirm that no errors are shown in the SoftConsole Problems view.

Configuring a terminal emulator program

9. Open the Windows Device Manager by selecting **Start > Settings > Control Panel > System > Hardware > Device Manager**. On Windows 7 PCs select **Start > Control Panel > Device Manager**.
10. Expand the **Ports (COM & LPT)** section and record the COM port assignment for the USB serial port below. If the target board is the SmartFusion2 Evaluation Kit or the SmartFusion2 Advanced Development kit, four consecutive COM ports will be displayed. Select the **Third highest** COM port for the SmartFusion2 Advanced Development kit. Select the **Highest (Fourth)** COM port for the SmartFusion2 Evaluation kit and the SmartFusion2 Development kit. The example below shows the COM port is assigned to COM39 (results for the SmartFusion2 Evaluation kit).

    USB Serial Port: COM_____

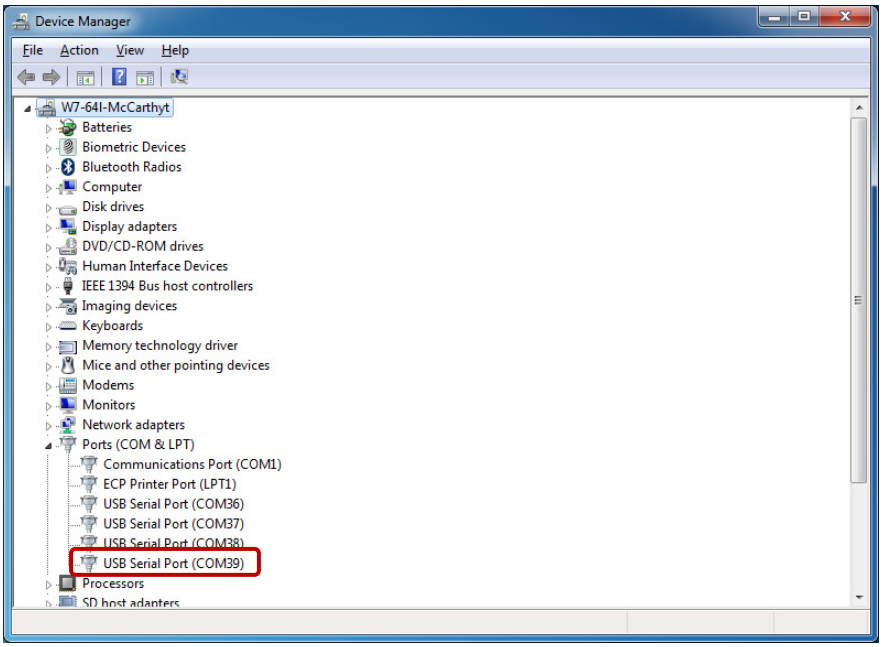


Figure 80 - Windows Device Manager (settings for M2S-EVAL-KIT board shown)

Open a terminal emulator program on your computer. Note: Newer versions of Windows do not include HyperTerminal. See page 5 for a link to HyperTerminal alternatives.

11. Select the COM port identified above and the following in the terminal emulator serial port settings:
    - Port: COM port recorded on the previous page
    - Baud rate: 57600
    - Data bits: 8
    - Parity: None
    - Stop bits: 1
    - Flow control: None

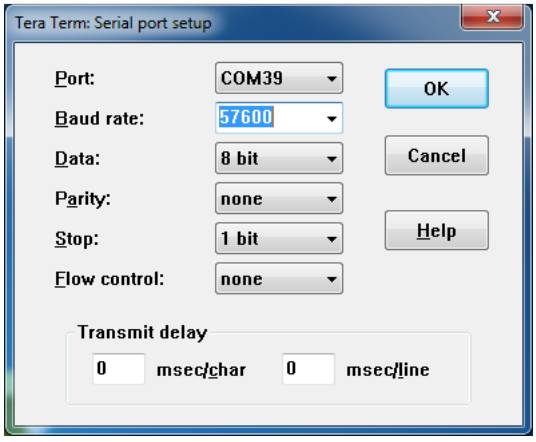


Figure 81 - Terminal emulator serial port settings

12. Click **OK** in the terminal emulator serial port settings dialog box.

Launching the debugger and running the application

13. Select **Run > Debug Configurations** from the SoftConsole menu. The Debug Configurations dialog box will open.
14. Select Simple_MSS_system_MSS_CM3_app Debug under Launch Group then click **Debug** to launch the SoftConsole debugger. You can also launch the debugger by clicking the debug icon () in the SoftConsole tool bar.
15. Click **Yes** in the **Confirm Perspective Switch** dialog box.
16. Select **Run > Resume** to run the application.
17. What is the message displayed on the HyperTerminal window?

_____

18. Click in the HyperTerminal window. Type a message on the keyboard and observe that it is echoed in the terminal emulator.
19. Close the terminal emulator program.
20. Terminate the SoftConsole debugger.
21. Close the Debug perspective (**Window > Close Perspective**).

Appendix 1 - Jumper settings for SmartFusion2 Starter kit

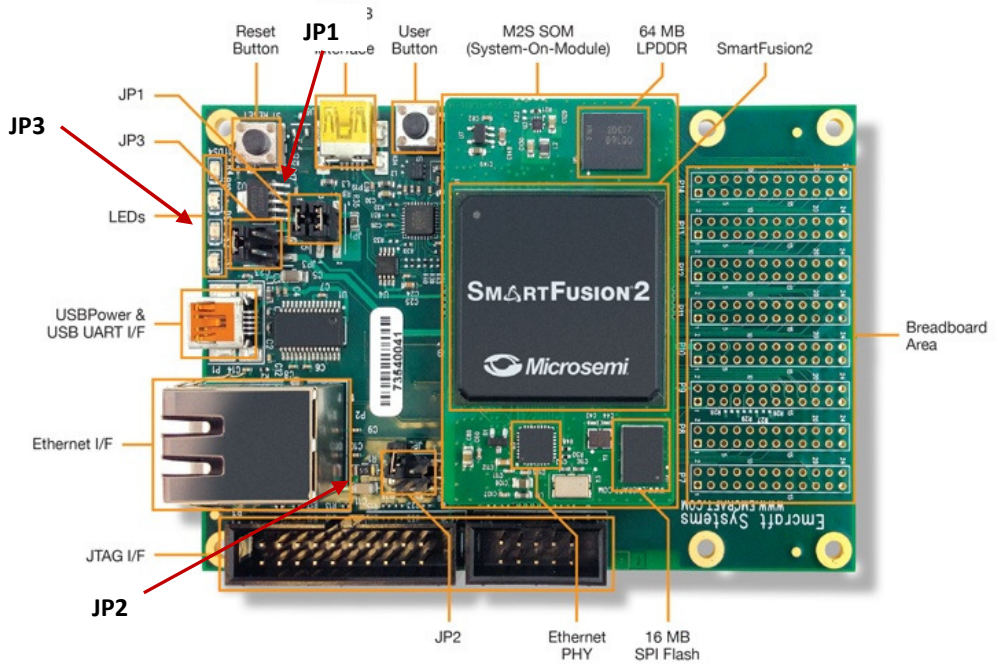


Figure 115 – SmartFusion2 Starter Kit

1. Prior to programming (and powering up) the SmartFusion2 Starter Kit board, confirm that the jumpers are positioned as show in the table below.

Jumper	Location	Function	Setting
JP1	Below the USB OTG connector in picture above	Enable power on the SmartFusion2 SOM (VCC3)	1-2 installed 3-4 open
JP2	To the right of the RJ-45 Ethernet jack in the picture above	Select appropriate JTAG mode and enable power to the SmartFusion2 JTAG connector	1-2 open 3-4 installed
JP3	Above the USB power and UART I/F connector in the picture above	Select the mini-USB port as the power source	1-3 open 2-4 installed

Table 6 – Jumper settings for the SmartFusion2 Starter kit

2. Connect the mini-USB Y-cable into the mini-USB connector labeled USB Power & USB UART I/F (P1) connector on SOM-BSB-EXT board (see figure above). Connect the other end of the cable to a USB port on your PC. As soon as the connection to the PC has been made, the on-board LED DS2 will illuminate, indicating that the board has power.

   A single USB connection provides a 500 mA power to the SF2-STARTER-KIT-ES, which is sufficient for basic functionality. Note however that some advanced operations, such as WiFi connectivity using the USB WiFi module, may require more than 500 mA for reliable operation. Use the second link of the mini-USB Y-cable to connect to the PC for such configurations.

   If prompted, install the FT232R drivers. The drivers can be downloaded from http://www.microsemi.com/document-portal/doc_download/131593-usb-uart-driver-files

3. Plug the FlashPro4 ribbon cable into connector P5 on the Starter Kit board.
4. Connect the mini USB cable between the FlashPro4 and the USB port of your PC.

5. Install the FlashPro4 drivers if prompted. The drivers are located in the <*FlashPro Installation Directory*>*Drivers* folder.

Appendix 2 - Jumper settings for SmartFusion2 Evaluation / Security Evaluation kit

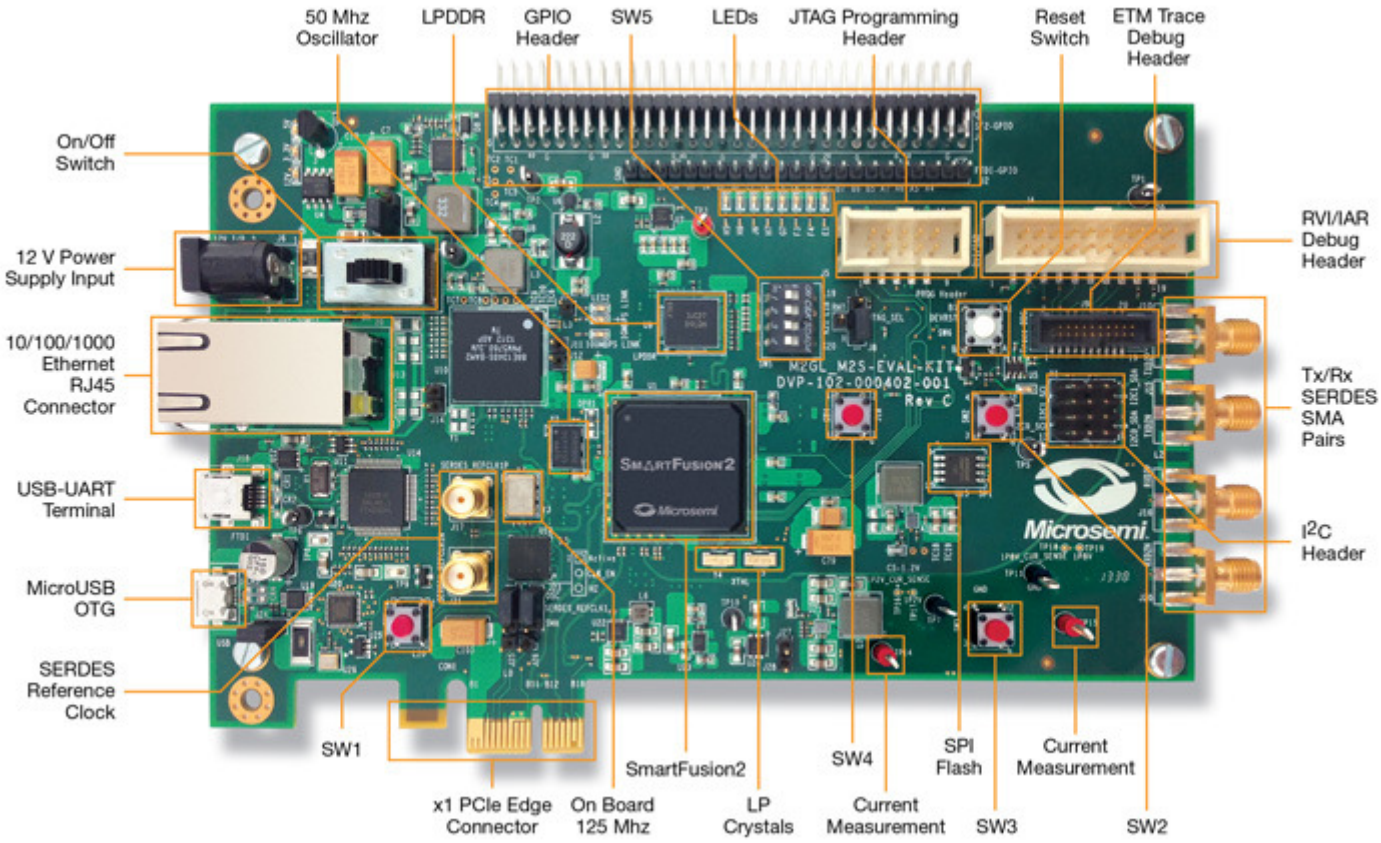


Figure 116 - SmartFusion2 Evaluation kit

1. Prior to programming (and powering up) the SmartFusion2 Evaluation Kit board, confirm that the jumpers are positioned as show in the table below.

Jumper	Location	Function	Setting
J3	Upper left-hand corner of board near the power switch (SW7)	Jumpers to select either SW2 input or signal ENABLE_FT4232 from FT4232H chip.	1-2 installed
J8	Below JTAG programming header	JTAG selection jumper to select between RVI header or FP4 header for application debug. • Pin 1-2 FP4 for SoftConsole/FlashPro • Pin 2-3 RVI for Keil ULINK™/IAR J-Link® • Pin 2-4 for Toggling JTAG_SEL signal remotely using GPIO capability of FT4232 chip.	1-2 installed
J22	Above PCIe connector	Jumper to select the output enables control for the line side outputs • Pin 1-2 (Line side output enabled) • Pin 2-3 (Line side output disabled)	1-2 installed
J23	Above PCIe connector	• Jumper to select switch-side Mux inputs	1-2 installed

		of A or B to the line side. • Pin 1-2 (Input A to the line side) that is on board 125 MHz differential clock oscillator output will be routed to line side. • Pin 2-3 (Input B to the line side) that is external clock required to source through SMA connectors to the line side.	
J24	Lower left-hand corner of board near the USB connector	Jumper to provide the VBUS supply to USB when using in Host mode.	1-2 installed
H1	Right side of board near SMA connectors	Connect I2C0_SCL to I2C1_SCL Connect I2C0_SDA to I2C1_SDA	6-10 installed 7-11 installed

Table 7 – Jumper settings for the SmartFusion2 Evaluation kit

2. Connect USB cable to USB-UART terminal (J18) and the other end of the cable to a USB port on your PC. If prompted, install the USB to UART bridge drivers. The drivers can be downloaded from www.microsemi.com/soc/documents/CDM_2.08.24_WHQL_Certified.zip.
3. Connect 12 V power supply brick to J6 on the Evaluation kit board.
4. Connect the FP4 header to J5. Connect a mini USB cable between the FlashPro4 and a USB port of your PC.
5. Install the FlashPro4 drivers if prompted. The drivers are located in the <*FlashPro Installation Directory*>*Drivers* folder.
6. Slide the main power switch SW7 to ON position.

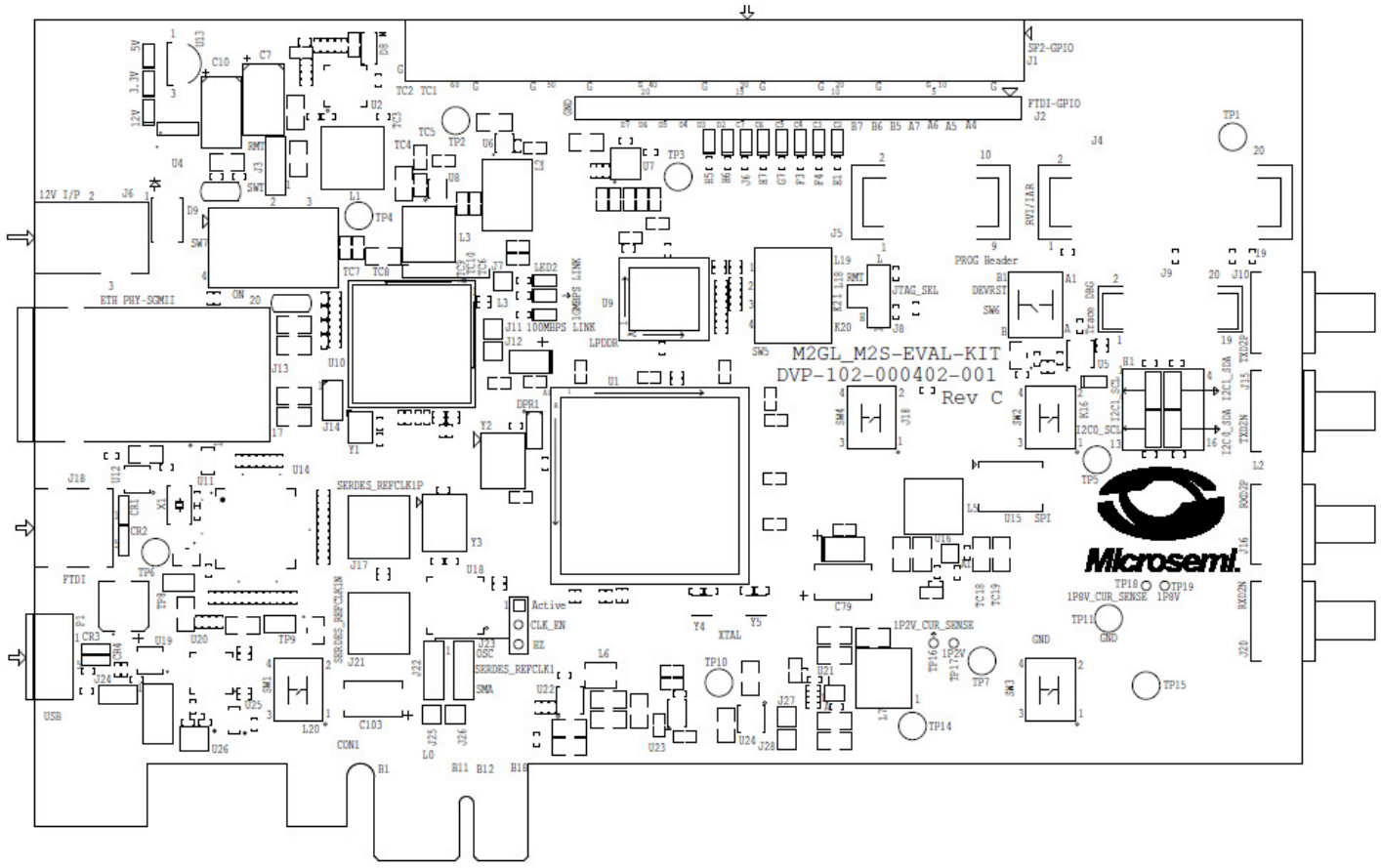


Figure 117 - SmartFusion2 Evaluation kit Silkscreen Top View

Appendix 3 - Jumper settings for SmartFusion2 Development kit

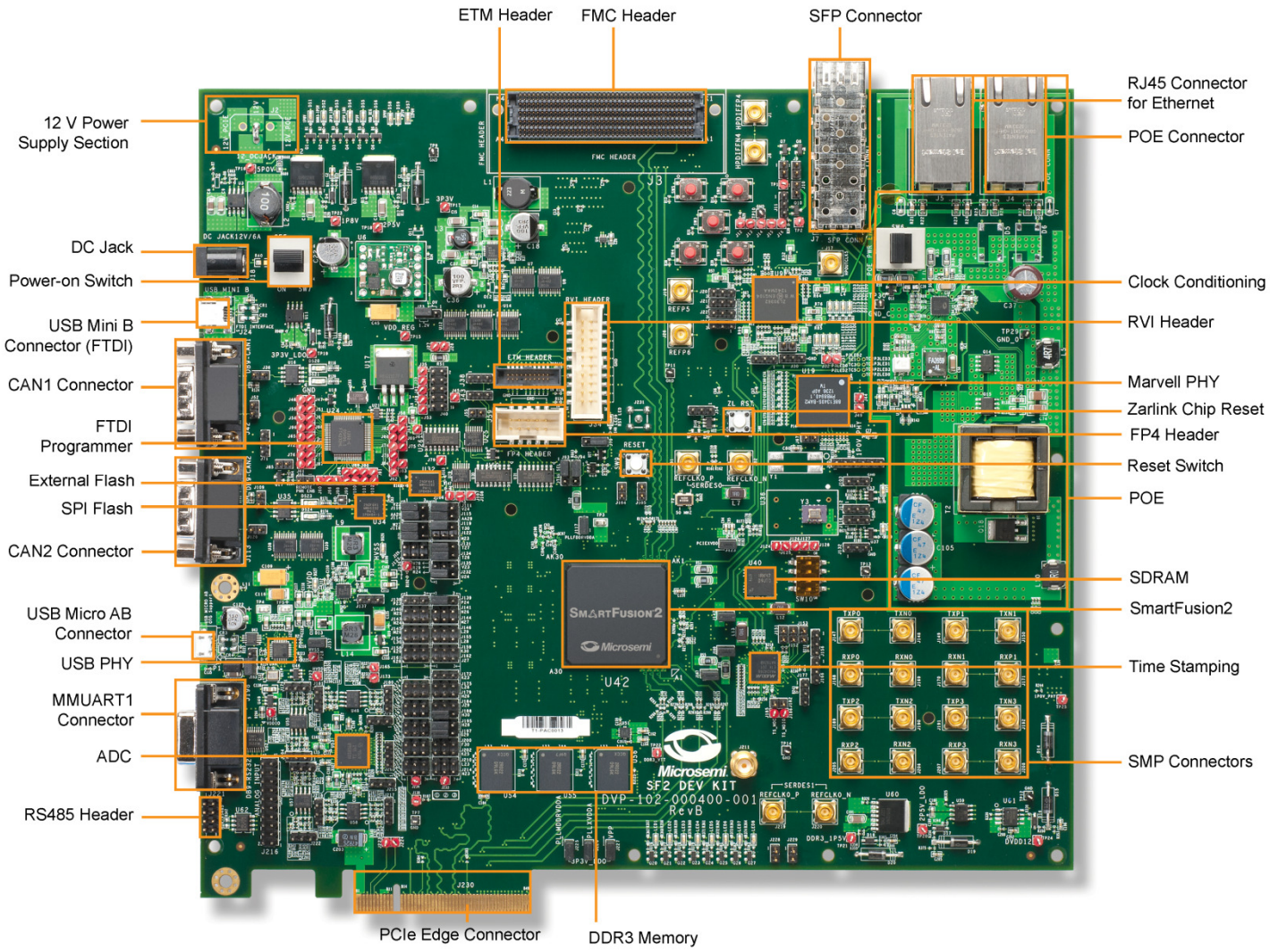


Figure 118 - SmartFusion2 Development kit

1. Prior to programming (and powering up) the SmartFusion2 Development Kit board, confirm that the jumpers are positioned as show in the table below.
2. Connect USB cable to USB mini-B connector (J24) and the other end of the cable to a USB port on your PC.
   If prompted, install the USB to UART bridge drivers. The drivers can be downloaded from www.microsemi.com/soc/documents/CDM_2.08.24_WHQL_Certified.zip.
3. Connect 12 V power supply brick to J18 to power the board.
4. Slide the main power switch SW7 to ON position.
5. Plug the FlashPro4 ribbon cable into connector J59 on the Development Kit board.
6. Connect the mini USB cable between the FlashPro4 and the USB port of your PC.
7. Install the FlashPro4 drivers if prompted. The drivers are located in the *<FlashPro Installation Directory>\Drivers* folder.

Jumper	Location	Function	Setting
J2	Upper left-hand corner of the board	Jumper to select the power supply option	1-3 installed (External DC Jack)
J23	Below regulator U6	Jumper to select the core voltage (VDD_REG) to either 1.0 V or 1.2 V	2-3 installed (1.2V core voltage)
J117	Above SmartFusion2 device (U42)	Jumper to connect 3P3V_LDO to PLLFDDRVDDA	1-2 installed
J123	Above U40	Jumper to connect VDD_REG to PCIExVDD	1-2 installed
J142	To the right of the SmartFusion2 device (U42)	Jumper to connect 3P3V_LDO to PLLPCIExVDDA	1-2 installed
J157	Lower left side of board to the left of L14	Jumper to connect 3P3V_LDO to VPPNVMSA0	1-2 installed
J160	To the right of the SmartFusion2 device (U42)	Jumper to connect VDD_REG to PCIExVDDIOx	1-2 installed
J167	To the right of the SmartFusion2 device (U42)	Jumper to connect 2P5V_LDO to PCIExVDDPLL	1-2 installed
J225	Bottom edge near PCIe connector	Jumper to connect 3P3V_LDO to PLLMDDRVDDA	1-2 installed
J226	Bottom edge near PCIe connector	Jumper to connect 3P3V_LDO to PLLXVDDA	1-2 installed
J227	Bottom edge near PCIe connector	Jumper to connect 3P3V_LDO to VPP	1-2 installed
J70	Below 20 pin RVI Header	Device reset generation selection • Pin 1-2: Reset depends on 3.3 V rail • Pin 2-3: Reset depends on FLASH_GOLDEN pin	1-2 installed
J93	Below 10 pin FP4 header	JTAG selection jumper to select between RVI header or FP4 header for application debug • Pin 1-2: FP4 for SoftConsole / FlashPro4 • Pin 2-3: RVI for Keil ULINK™/IAR J-Link®	1-2 installed 2-3 open
J94	Below 10 pin FP4 header	Jumper to select the JTAG reset • Pin 1-2: selects reset from FP4 header/RVI header depending on JTAG selection • Pin 2-3: SPI flash programming mode	1-2 installed 2-3 open
J110	To the right of the DB9 CAN2 connector	Jumper to select between SPI flash SCK and FMC_V22 • Pin 1-2 SPI flash • Pin 2-3 FMC	open
J118	To the right of the DB9 CAN2 connector	Jumper to select between SPI flash SDO and FMC_W27 • Pin 1-2 SPI flash • Pin 2-3 FMC	1-2 installed
J119	To the right of the DB9 CAN2 connector	Jumper to select between SPI flash SDI and FMC_Y30 • Pin 1-2 SPI flash • Pin 2-3 FMC	1-2 installed
J121	To the right of the DB9 CAN2 connector	Jumper to select between SPI flash SS and FMC_W28 • Pin 1-2 SPI flash	1-2 installed

		• Pin 2-3 FMC	
J129	To the right of the DB9 CAN2 connector	Jumper to select between FT4232 DD0 and FMC_R29 to SPI_1_SS1 • Pin 1-2 FMC • Pin 2-3 FT4232	2-3 installed
J133	To the right of the DB9 CAN2 connector	Jumper to select between FT4232 DD1 and FMC_R24 to SPI_1_SS2 • Pin 1-2 FMC • Pin 2-3 FT4232	2-3 installed
J139	To the left of the SmartFusion2 device (U42)	Jumper to select between USB reset and FMC_P24 • Pin 1-2 USB • Pin 2-3 FMC	1-2 installed
J163	Near the DB9 RS-232 connector	Jumper to select the USB mode of operation • Pin 1-2 On-The-Go (OTG) mode • Pin 2-3 Either host or device mode	1-2 installed
J164	Near the DB9 RS-232 connector	Jumper to provide the VBUS supply to USB when using in Host mode	open

Table 8 - Jumper settings for the SmartFusion2 Development kit

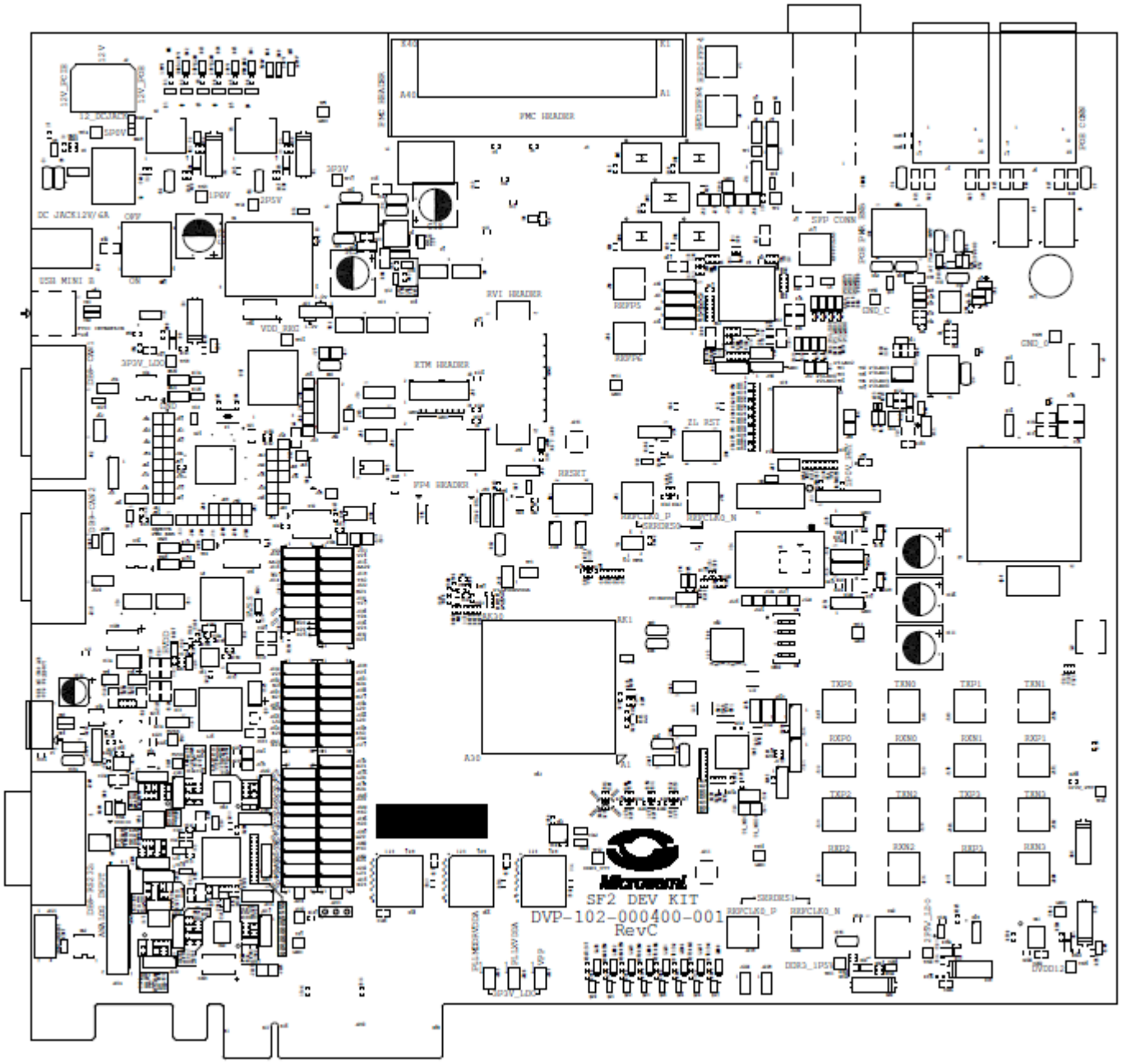


Figure 119 - SmartFusion2 Development kit Silkscreen Top View

Appendix 4 - Jumper settings – SmartFusion2 Advanced Development kit

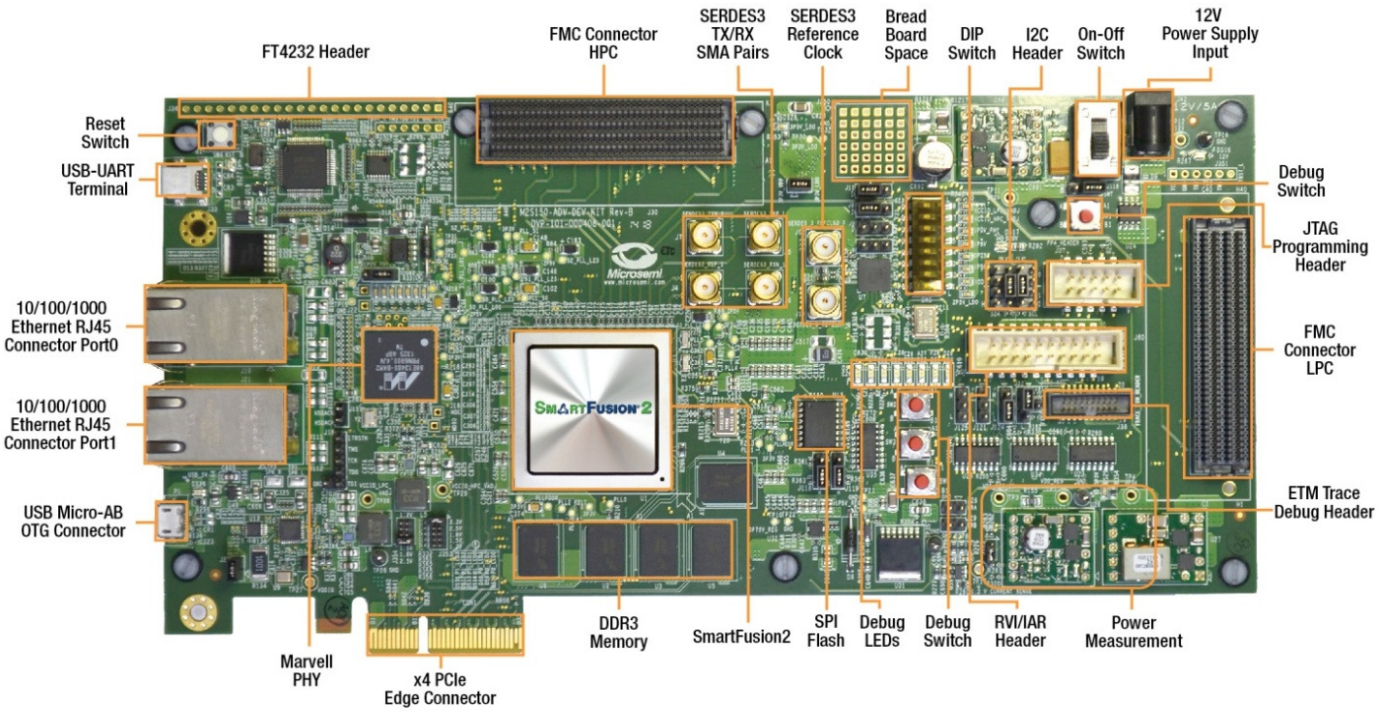


Figure 120 - SmartFusion2 Advanced Development kit

1. Prior to programming (and powering up) the SmartFusion2 Development Kit board, confirm that the jumpers are positioned as show in the table below.
2. Connect USB cable to USB-UART terminal (J33) and the other end of the cable to a USB port on your PC.
3. If prompted, install the USB to UART bridge drivers. The drivers can be downloaded from www.microsemi.com/soc/documents/CDM_2.08.24_WHQL_Certified.zip.
4. Connect the FlashPro4 ribbon cable to J37. Connect the mini USB cable between the FlashPro4 and the USB port of your PC. Install the FlashPro4 drivers if prompted. The drivers are located in the <*FlashPro Installation Directory*>*Drivers* folder.
5. Connect 12 V power supply brick to J42 to power the board.
6. Slide the main power switch SW7 to ON position.

Jumper	Location	Function	Setting
J123	Bottom edge of the board to the right of U31.	Jumper to select the core voltage (VDD_REG) to either 1.0V or 1.2V • Pin 1–2 for 1.0V core voltage • Pin 2–3 for 1.2V core voltage	2-3 installed (1.2V core)
J353	Above PCIe connector.	Jumper to select the core voltage (VCCIO_HPC_VADJ) to either 3.3V or 2.5V or 1.8V or 1.5V or 1.2V • Pin 1–2 for 3.3V • Pin 3–4 for 2.5V • Pin 5–6 for 1.8V • Pin 7–8 for 1.5V • Pin 9-10 for 1.2V	1-2 installed
J354	Above PCIe connector.	Jumper to select the core voltage (VCCIO_LPC_VADJ) to 2.5 V or 1.8 V or 1.5 V or 1.2V • Pin 1–2 for 2.5 V • Pin 3–4 for 1.8 V • Pin 5–6 for 1.5 V • Pin 7-8 for 1.2 V	1-2 installed
J116	Below on/off switch (SW7).	Jumpers to select either SW7 input or signal ENABLE_FT4232 from FT4232H chip • Pin 1–2 for SW7 selection • Pin 2-3 for “Enable_FT4232” signal control	1-2 installed
J8	Left of DIP switch (SW5).	Jumper to select the output enables control for the line side outputs • Pin 1-2 (Line side output enabled) • Pin 2-3 (Line side output disabled)	1-2 installed
J11	Left of DIP switch (SW5).	Jumper to select switch-side Mux inputs of A or B to the line side • Pin 1-2 (Input A to the line side) that is on board 125 MHz differential clock oscillator output is routed to line side • Pin 2-3 (Input B to the line side) that is on board 100 MHz differential clock oscillator output is routed to line side	1-2 installed
J32	Left of TRACE_TEM_HEADER (J38).	JTAG selection jumper to select between RVI header or FP4 header for application debug • Pin 1-2 FP4 for Soft Console/Flash Pro • Pin 2-3 RVI for Keil ULINK™/IAR J-Link® • Pin 2-4 for JTAG_SEL pin to DD1 signal of FT4232H chip	1-2 installed
J121	Left of TRACE_TEM_HEADER (J38).	Select FTDI JTAG/ SPI Slave programming • Pin 1-2 FTDI JTAG programming • Pin 2-3 FTDI SPI Slave Programming	1-2 installed
J124	Left of TRACE_TEM_HEADER (J38).	Select JTAG programming via FP4 or FTDI • Pin 1-2 JTAG programming via FTDI • Pin 2-3 JTAG programming via FP4	2-3 installed
J118	Left of push-button switches.	Select programming SPI-0 flash through FTDI	1-2 installed

		SPI-0 (Port-B) or SmartFusion2 SPI-0 • Pin 1-2 Programming SPI-0 flash via SmartFusion2 SPI-0 • Pin 2-3 Programming SPI-0 flash via FTDI SPI-0 (Port-B) and J125 pin 2-3 must be shorted	
J119	Left of push-button switches.	Select programming SPI-1 flash through FTDI SPI (Port-B) or SmartFusion2 SPI-1 • Pin 1-2 Programming SPI-1 flash via SmartFusion2 SPI-1 • Pin 2-3 Programming SPI-1 flash via FTDI SPI (Port-B) and J125 pin 1-2 must be shorted	1-2 installed

Table 9 - Jumper settings for the SmartFusion2 Advanced Development kit

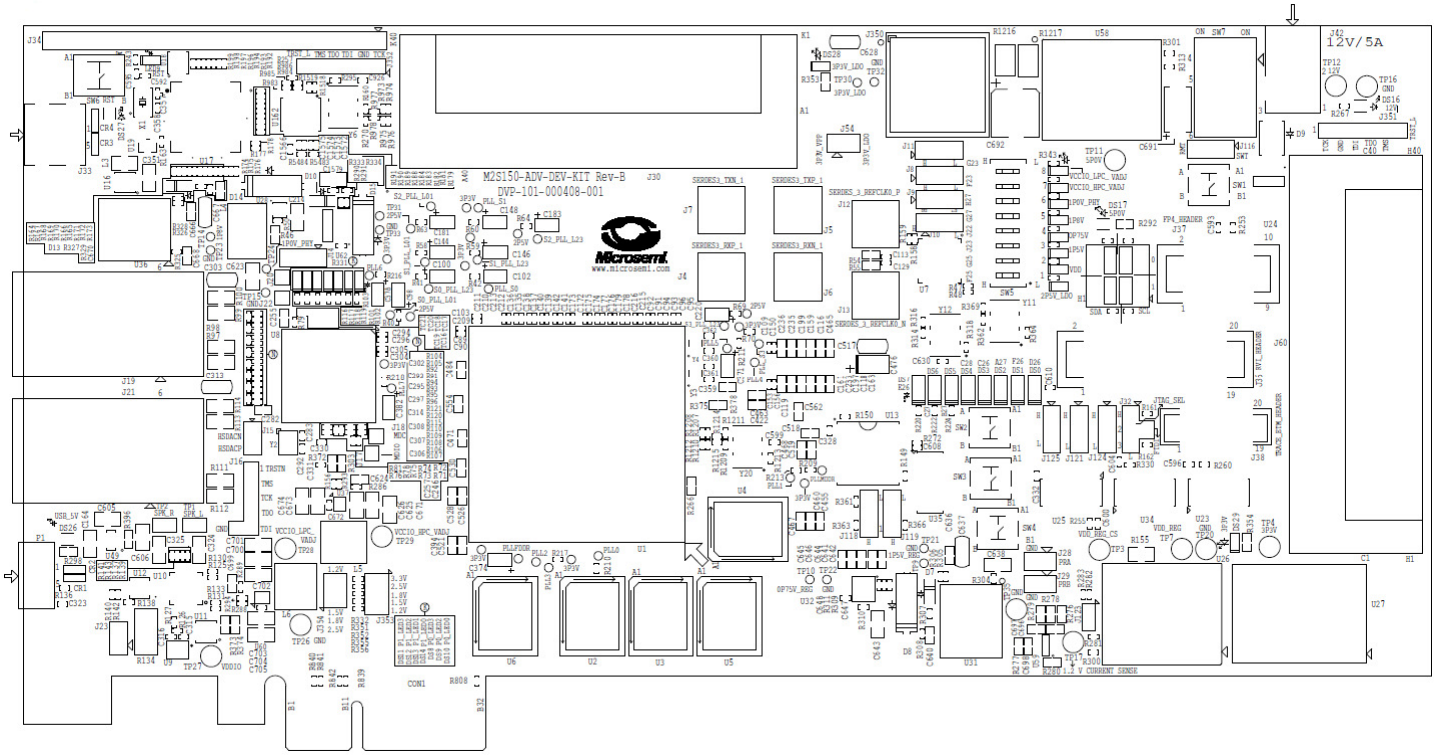


Figure 121 - SmartFusion2 Advanced Development kit Silkscreen Top View

Record Answers to questions below

Name: _____

Page 33

Scroll in the SmartFusion_Simple_MSS data sheet and become familiar with the Generated Files, Firmware and Memory Map sections (click on the hyperlink at the top of the data sheet to move to the section of interest). Answer the following questions:

What is the address of MMUART_0? _____

What is the address of the GPIO? _____

Page 57

If time permits, modify main.c to make LEDs DS2 and DS3 toggle instead of blinking together. Indicate the change you made to the code below.

Page 61

What is the message displayed on the HyperTerminal window?

Answers to Questions

Page 33

Scroll in the SmartFusion_Simple_MSS data sheet and become familiar with the Generated Files, Firmware and Memory Map sections (click on the hyperlink at the top of the data sheet to move to the section of interest). Answer the following questions:

What is the address of MMUART_0? 0x40000000 - 0x40000FFF

What is the address of the GPIO? 0x40013000 - 0x40013FFF

Page 57

If time permits, modify main.c to make LEDs DS2 and DS3 toggle instead of blinking together.

Add the following after line 35 in main.c:

```
MSS_GPIO_set_outputs( MSS_GPIO_0_MASK );
```

Page 61

What is the message displayed on the HyperTerminal window?

```
************************************************************************

******************** SmartFusion2 MMUART Example *********************

************************************************************************

   Characters typed will be echoed back.
```

List of Figures