

Creating a Libero Project for Firmware Catalog Sample Project - Libero SoC v11.7 and SoftConsole Flow Tutorial for SmartFusion2

TU0487 Tutorial



Power Matters.™

Contents

1	Preface	5
1.1	Purpose	5
1.2	Intended Audience	5
1.3	References	5
2	Creating a Libero Project for Firmware Catalog Sample Project - Libero SoC v11.7 and SoftConsole Flow Tutorial for SmartFusion2 6	
2.1	Introduction	6
2.2	Design Requirements	7
2.2.1	Project Files	7
2.3	Design Overview	8
2.4	Downloading SoftConsole Project from Firmware Catalog	9
2.5	Creating a Libero SoC Project	10
2.5.1	Launching Libero SoC	10
2.5.2	Connecting Components in Syservices_sb_0 SmartDesign	19
2.5.3	Configuring and Generating Firmware	20
2.6	Generating the Program File	22
2.7	Programming SmartFusion2 Security Evaluation Board Using FlashPro	24
2.8	Building Software Application Using SoftConsole	26
2.9	Conclusion	37
3	Appendix: Jumper Locations	38
4	Appendix: Board Setup for Running the Tutorial	39
5	Revision History	40
6	Product Support	41
6.1	Customer Service	41
6.2	Customer Technical Support Center	41
6.3	Technical Support	41
6.4	Website	41
6.5	Contacting the Customer Technical Support Center	41
6.5.1	Email	41
6.5.2	My Cases	41
6.5.3	Outside the U.S.	42
6.6	ITAR Technical Support	42

Figures

Figure 1.	Demo Design Files Top-Level Structure	8
Figure 2.	Downloading Sample Project from Firmware Catalog	9
Figure 3.	Generate Sample Options Window	10
Figure 4.	Libero SoC Project Manager	10
Figure 5.	Project Details Page	11
Figure 6.	Device Selection Page	12
Figure 7.	Device Settings Page	13
Figure 8.	Device Template Page	13
Figure 9.	Add HDL Sources Page	14
Figure 10.	Add Constraints Page	14
Figure 11.	New Project Information Window	15
Figure 12.	System Builder Dialog Box	16
Figure 13.	System Builders - Device Features Page	16
Figure 14.	System Builder - Peripherals Page	17
Figure 15.	System Builder - Clocks Page	18
Figure 16.	Updating Syssservices_sb_0	18
Figure 17.	Sysservices_sb_0	19
Figure 18.	Opening DESGIN_FIRMWARE Window	20
Figure 19.	DESIGN_FIRMWARE Window	21
Figure 20.	Generate Component	22
Figure 21.	Generate Bitstream Data	22
Figure 22.	Export Firmware	23
Figure 23.	Export Firmware Dialog Box	23
Figure 24.	Firmware Successfully Exported Message	23
Figure 25.	Firmware Log Window	24
Figure 26.	Run Program Action	25
Figure 27.	Invoking SoftConsole IDE	26
Figure 28.	Importing Projects	26
Figure 29.	Importing Existing Projects	27
Figure 30.	Import Window	27
Figure 31.	Selecting System services	28
Figure 32.	Adding Projects to SoftConsole IDE	28
Figure 33.	SoftConsole Workspace	29
Figure 34.	Sample Project main. c File	29
Figure 35.	SoftConsole Workspace - main.c File	30
Figure 36.	Project Explorer Window - SoftConsole Project	30
Figure 37.	Sysservices_sb_MSS_CM3 Properties Window	31
Figure 38.	Settings for Clean Build	32
Figure 39.	Device Manager Window	33
Figure 40.	Debug Configurations Window	34
Figure 41.	Confirm Perspective Switch	34
Figure 42.	SoftConsole Debugger Perspective	35
Figure 43.	PuTTY Window	36
Figure 44.	Jumper Locations	38
Figure 45.	SmartFusion2 Security Evaluation Kit	39

Tables

Table 1.	Design Requirements	7
Table 2.	SmartFusion2 Security Evaluation Kit Jumper Settings	24

1 Preface

1.1 Purpose

This tutorial is for SmartFusion2® System-on-Chip (SoC) field programmable gate array (FPGA) devices and describes how to download the SoftConsole sample project from Firmware catalog and create a Libero® System-on-Chip (SoC) hardware design for the downloaded sample project.

1.2 Intended Audience

This tutorial is intended for:

- FPGA designers
- Embedded designers
- System-level designers

1.3 References

The following documents are referred in this tutorial:

- *TU0546: SoftConsole v4.0 and Libero SoC v11.7 Tutorial*
- *UG0450: SmartFusion2 FPGA SoC and IGLOO2 FPGA System Controller User Guide*
- *Configuring Serial Terminal Emulation Programs Tutorial*

2 Creating a Libero Project for Firmware Catalog Sample Project - Libero SoC v11.7 and SoftConsole Flow Tutorial for SmartFusion2

2.1 Introduction

Libero SoC firmware catalog shows a list of available firmware cores. Sample projects for each firmware core can be generated from firmware catalog. A sample project is an example of how the firmware core can be integrated in a project. This sample project contains firmware project using SoftConsole, IAR workbench, and Keil tools. This sample project does not have a Libero project for that generated firmware project. Each sample project folder contains a `Readme.text` file which gives an overview of the design and hardware requirements. Using this information, the Libero SoC project can be generated. This tutorial provides an example design for system services.

This tutorial describes the following:

- [Downloading SoftConsole Project from Firmware Catalog](#)
- [Creating a Libero SoC Project](#)
- [Generating the Program File](#)
- [Programming SmartFusion2 Security Evaluation Board Using FlashPro](#)
- [Building Software Application Using SoftConsole](#)

2.2 Design Requirements

Table 1 lists the hardware and software design requirements.

Table 1 • Design Requirements

Design Requirements	Description
Hardware Requirements	
SmartFusion2 Security Evaluation Kit: - FlashPro4 programmer - USB A to Mini-B cable 12 V Adapter	Rev D or later
Host PC or Laptop	Any 64-bit Windows Operating System
Software Requirements	
Libero SoC	v11.7
SoftConsole	v3.4 SPI*
FlashPro programming software	v11.7
Host PC Drivers	USB to UART drivers
Any one of the following serial terminal emulation programs: - HyperTerminal - TeraTerm - PuTTY	–
Note: *For this tutorial, SoftConsole v3.4 SP1 is used. For using SoftConsole v4.0, see the TU0546: SoftConsole v4.0 and Libero SoC v11.7 Tutorial .	

2.2.1 Project Files

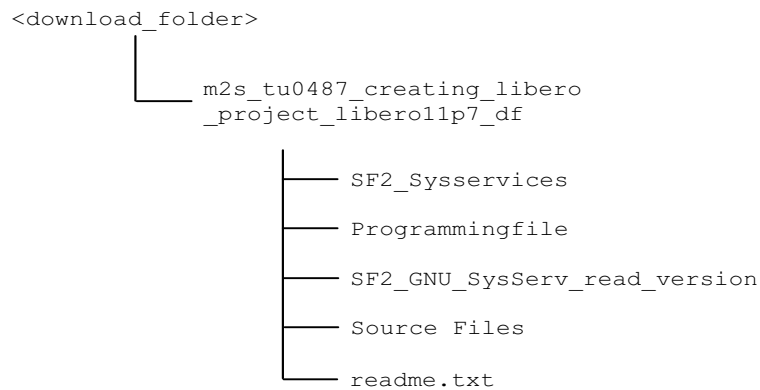
The design files for this tutorial can be downloaded from the Microsemi website:
http://soc.microsemi.com/download/rsc/?f=m2s_tu0487_creating_libero_project_libero11p7_df

The design files include:

- Libero project
- Programming files
- SF2_GNU_SysServ_read_version
- Readme file

Figure 1 shows the top-level structure of the design files. For further details, refer to the `readme.txt` file.

Figure 1 • Demo Design Files Top-Level Structure



2.3 Design Overview

This tutorial demonstrates the following device and design Information services:

- **Serial Number Service:** Fetches the 128-bit device serial number (DSN) and set during manufacturing.
- **USERCODE Service:** Fetches the programmed 32-bit JTAG USERCODE.
- **User Design Version Service:** Fetches the 16-bit user design version.
- **Device Certificate:** Fetches the device certificate.
- **NVM Data Integrity Check Service:** Recalculates and compares cryptographic digests of the selected NVM component(s)—fabric, eNVM0, and eNVM1—to those previously computed and saved in NVM.

Note: In this tutorial, only fabric digest check is demonstrated.

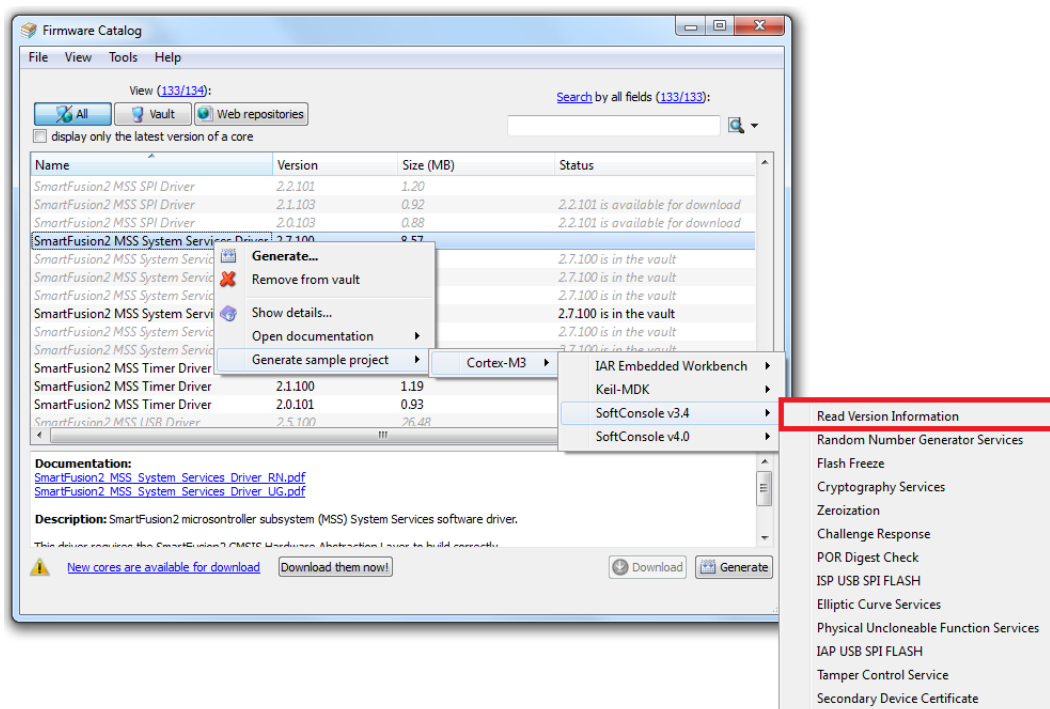
System services Information is displayed on HyperTerminal using MMUART_1 interface. For more information on system services, refer to the [UG0450: SmartFusion2 FPGA SoC and IGLOO2 FPGA System Controller User Guide](#).

2.4 Downloading SoftConsole Project from Firmware Catalog

The following steps describe how to download the SoftConsole project from firmware catalog:

1. Click **Start > Programs > Microsemi SoC Libero SoC 11.7 > Firmware Catalog v11.7 > Firmware Catalog**. This opens the **Firmware Catalog** windows, as shown in Figure 2.
2. Right-click **SmartFusion2 MSS System Services Driver** and select **Generate Sample Project > Cortex-M3 > SoftConsole > Read Version Information**, as shown in Figure 2.

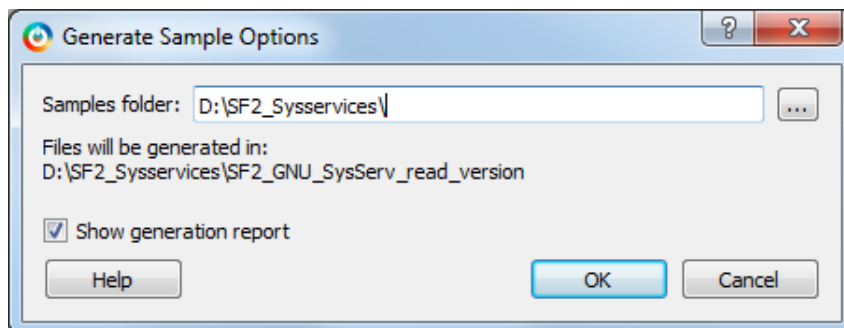
Figure 2 • Downloading Sample Project from Firmware Catalog



Note: Select the latest version of the SmartFusion2 microcontroller subsystem (MSS) services driver.

The **Generate Sample Options** window is displayed, as shown in Figure 3.

Figure 3 • Generate Sample Options Window



3. Browse to a location to save system services **Read Version Information** SoftConsole Project.
4. Open Readme file provided in the SF2_GNU_SysServ_read_version project folder. Readme file gives target hardware information.
5. SF2_GNU_SysServ_read_version project folder is also provided along with design files for reference.

2.5 Creating a Libero SoC Project

The following steps describe how to create SmartFusion2 firmware catalog design using the Libero SoC tool.

2.5.1 Launching Libero SoC

The following steps describe how to launch Libero SoC:

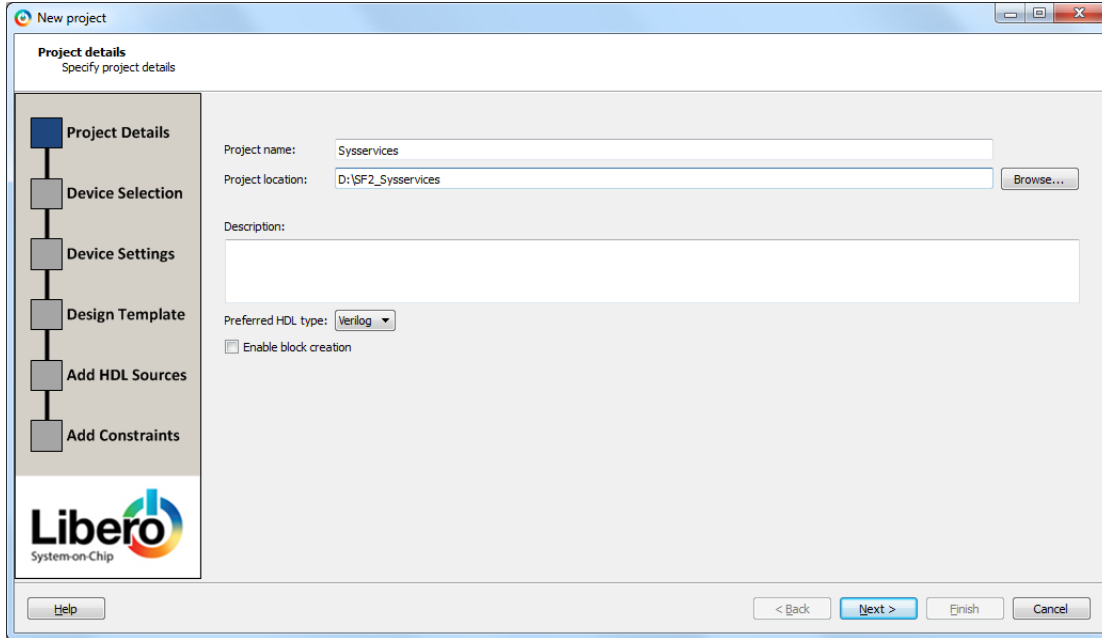
1. Click **Start > Programs > Microsemi Libero SoC v11.7 > Libero SoC v11.7**, or click the shortcut on desktop, to open the Libero v11.7 Project Manager.
2. Create a new project using one of the following options:
 - Select **New** on the **Start Page** tab, as shown in Figure 4.
 - Click **Project > New Project** from the Libero SoC menu.

Figure 4 • Libero SoC Project Manager



3. Enter the following information in the **Project Details** page, as shown in [Figure 5](#):
 - **Project Name:** Syssservices
 - **Project Location:** Select an appropriate location (for example, D:/SF2_Sysservices)
 - **Preferred HDL Type:** Verilog

Figure 5 • Project Details Page



New project

Project details
Specify project details

Project Details

Project name: Syssservices

Project location: D:\SF2_Sysservices

Description:

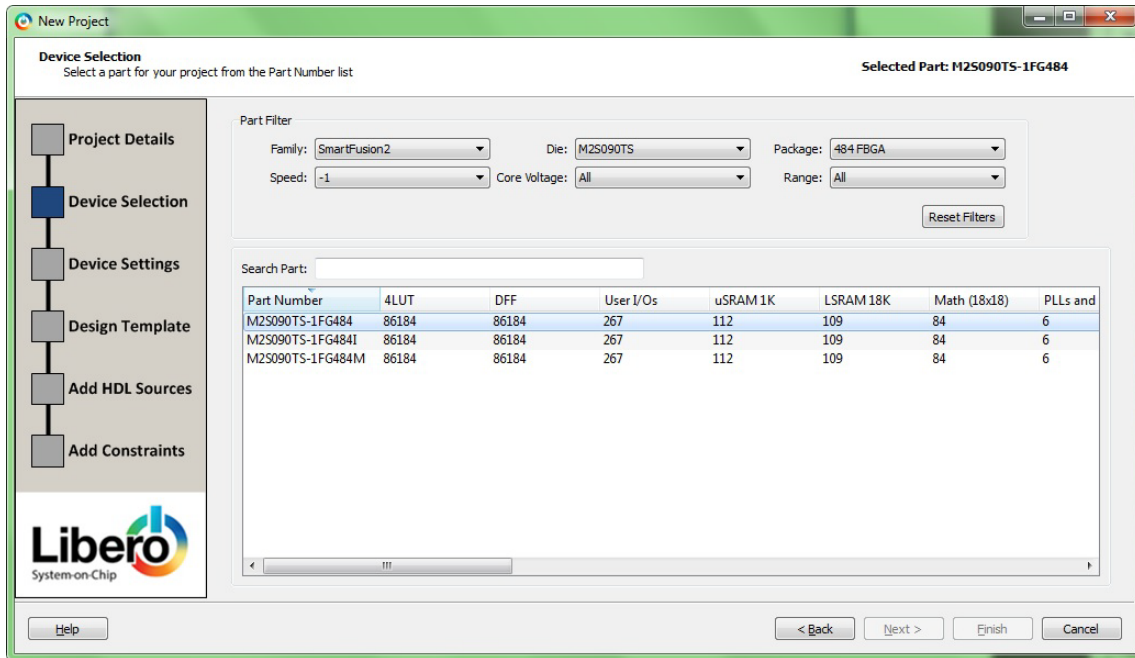
Preferred HDL type: Verilog ▼

☐ Enable block creation

Libero
System-on-Chip

4. Click **Next**. This opens **Device Selection** page, as shown in [Figure 6](#).
Select the following values from the drop-down list:
 - **Family:** SmartFusion2
 - **Die:** M2S090TS
 - **Package:** 484 FBGA
 - **Speed:** -1
 - **Core Voltage:** All
 - **Range:** All

Figure 6 • Device Selection Page



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: All Range: All

Reset Filters

Search Part:

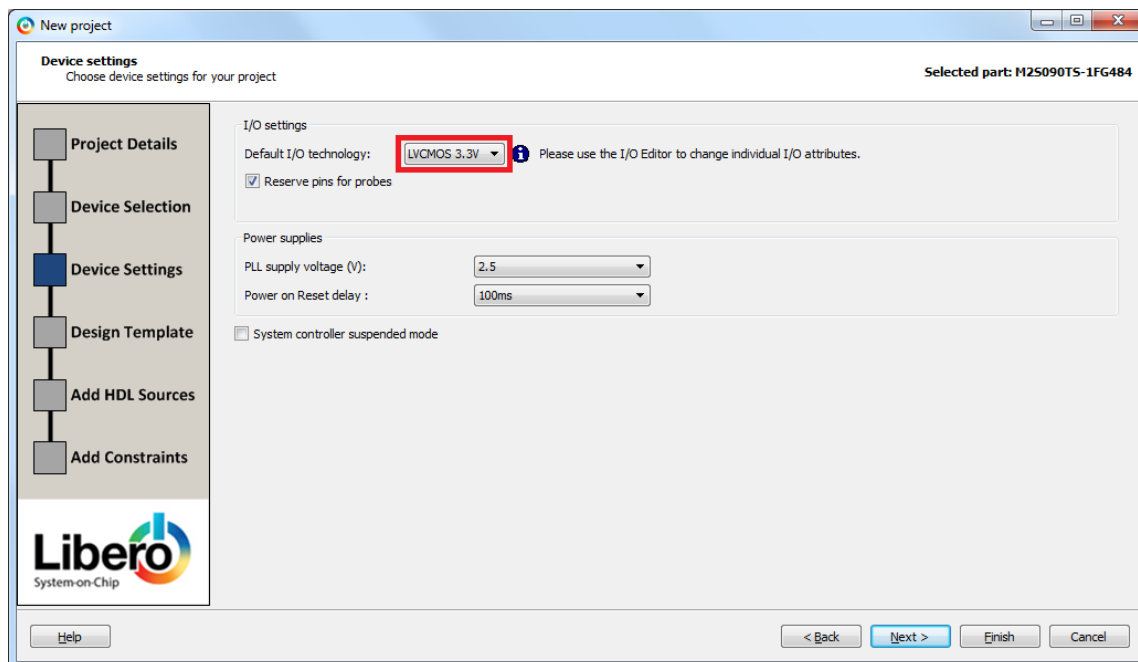
Part Number	4LUT	DFF	User I/Os	uSRAM 1K	LSRAM 18K	Math (18x18)	PLLs and
M2S090TS-1FG484	86184	86184	267	112	109	84	6
M2S090TS-1FG484I	86184	86184	267	112	109	84	6
M2S090TS-1FG484M	86184	86184	267	112	109	84	6

Libero
System-on-Chip

Help < Back Next > Finish Cancel

5. Click **Next**. This opens **Device Settings** page. Change the default I/O technology to LVCMOS 3.3 V.

Figure 7 • Device Settings Page



New project

Device settings
Choose device settings for your project

Selected part: M25090TS-1FG484

I/O settings

Default I/O technology: **LVCMOS 3.3V** ⓘ Please use the I/O Editor to change individual I/O attributes.

☒ Reserve pins for probes

Power supplies

PLL supply voltage (V): 2.5

Power on Reset delay : 100ms

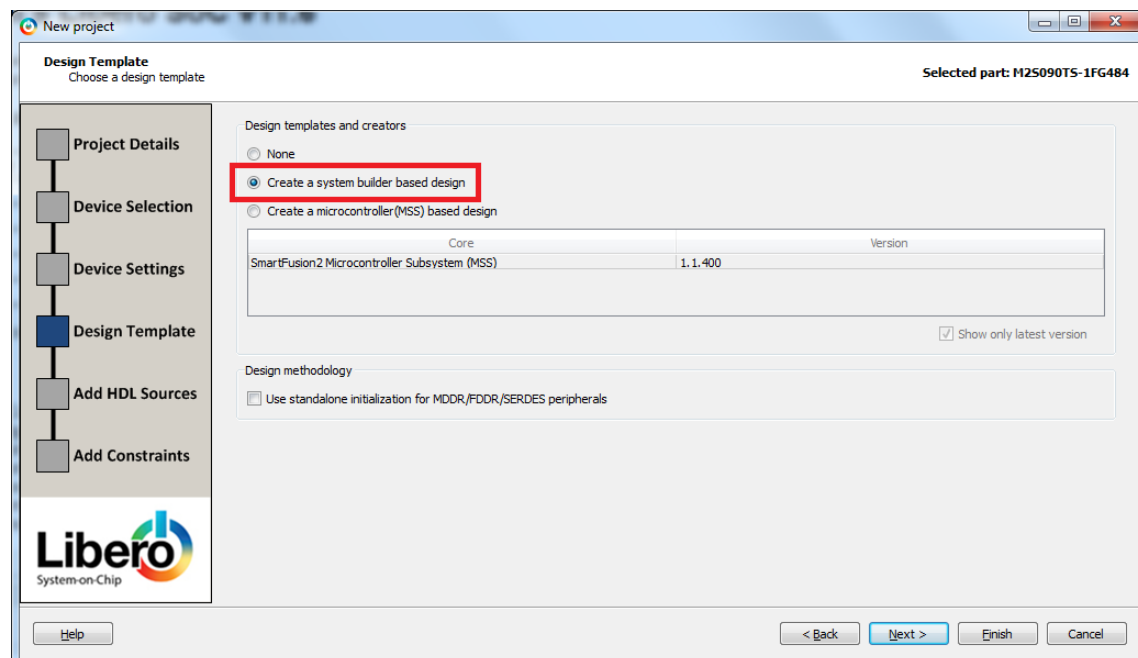
☐ System controller suspended mode

Libero
System-on-Chip

Help < Back Next > Finish Cancel

6. Click **Next**. This opens **Design Template** page, as shown in [Figure 8](#). Under **Design templates and creators**, select **Create a system builder based design**.

Figure 8 • Device Template Page



New project

Design Template
Choose a design template

Selected part: M25090TS-1FG484

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

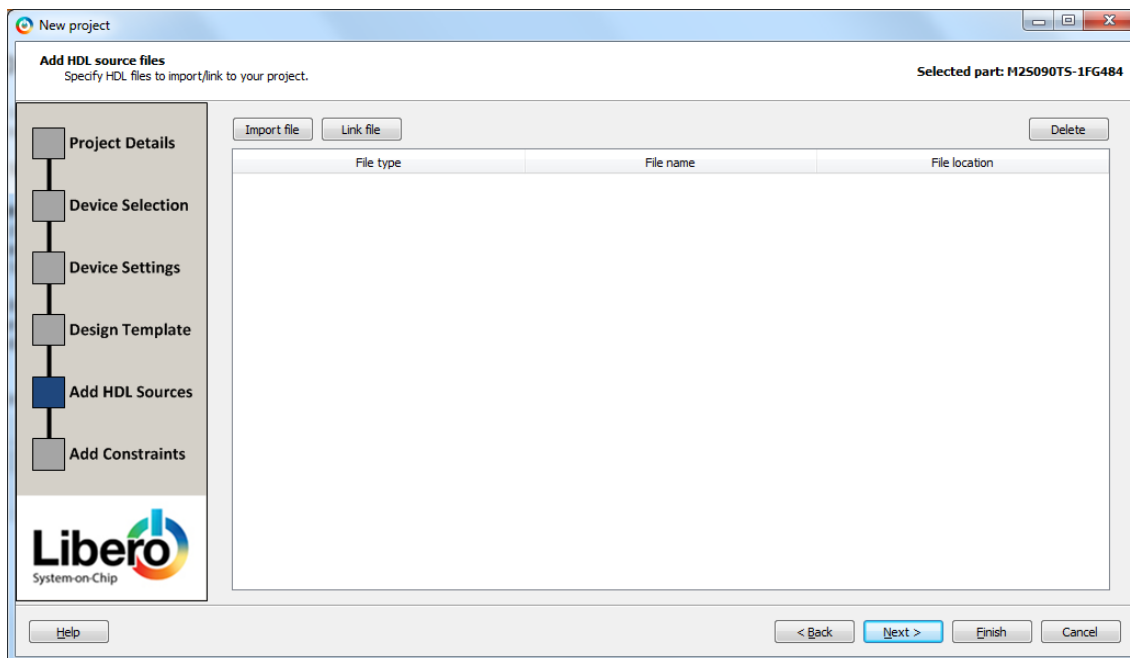
☐ Use standalone initialization for MDDR/FDDR/SERDES peripherals

Libero
System-on-Chip

Help < Back Next > Finish Cancel

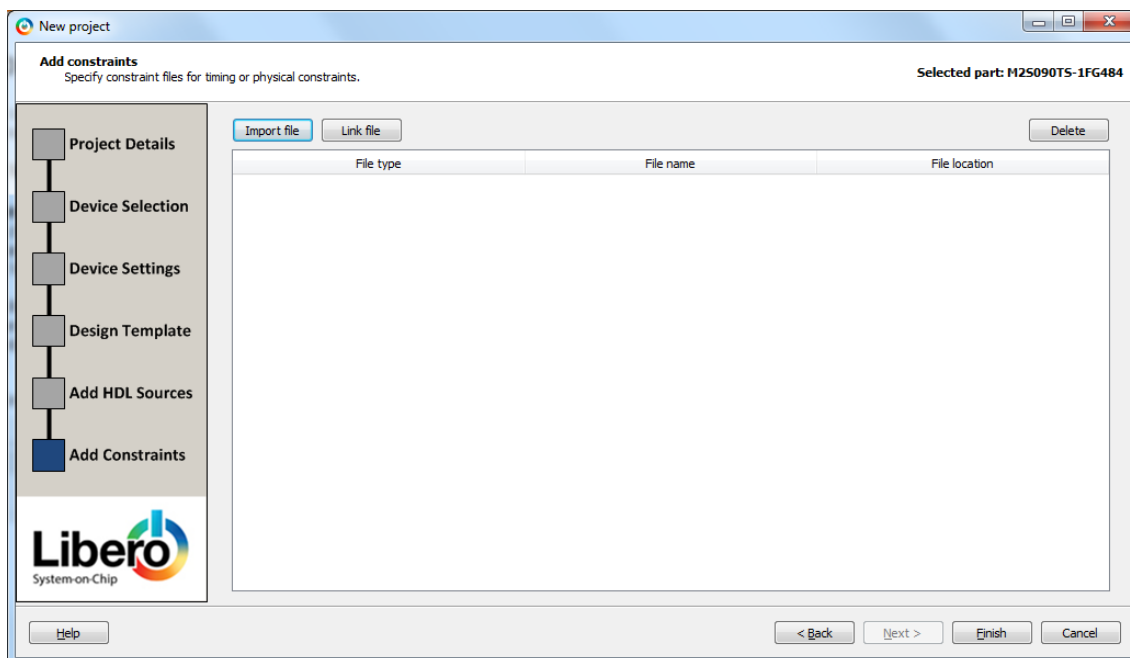
7. Click **Next**. This opens **Add HDL Sources** page, as shown in [Figure 9](#).

Figure 9 • Add HDL Sources Page



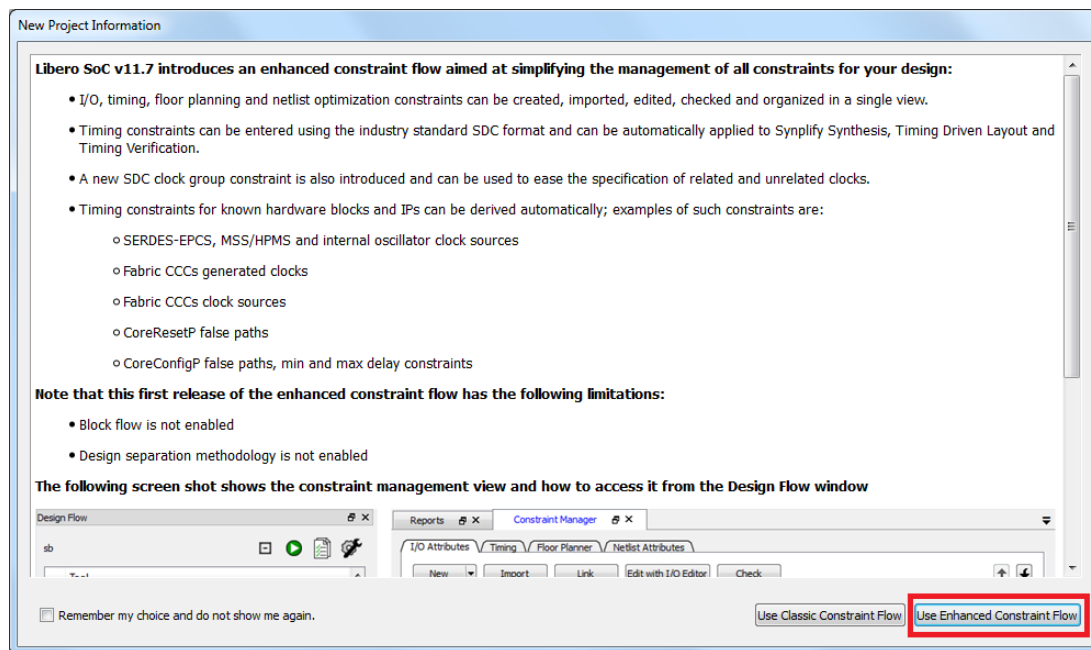
8. Click **Next**. This opens **Add Constraints** page, as shown in [Figure 10](#).

Figure 10 • Add Constraints Page



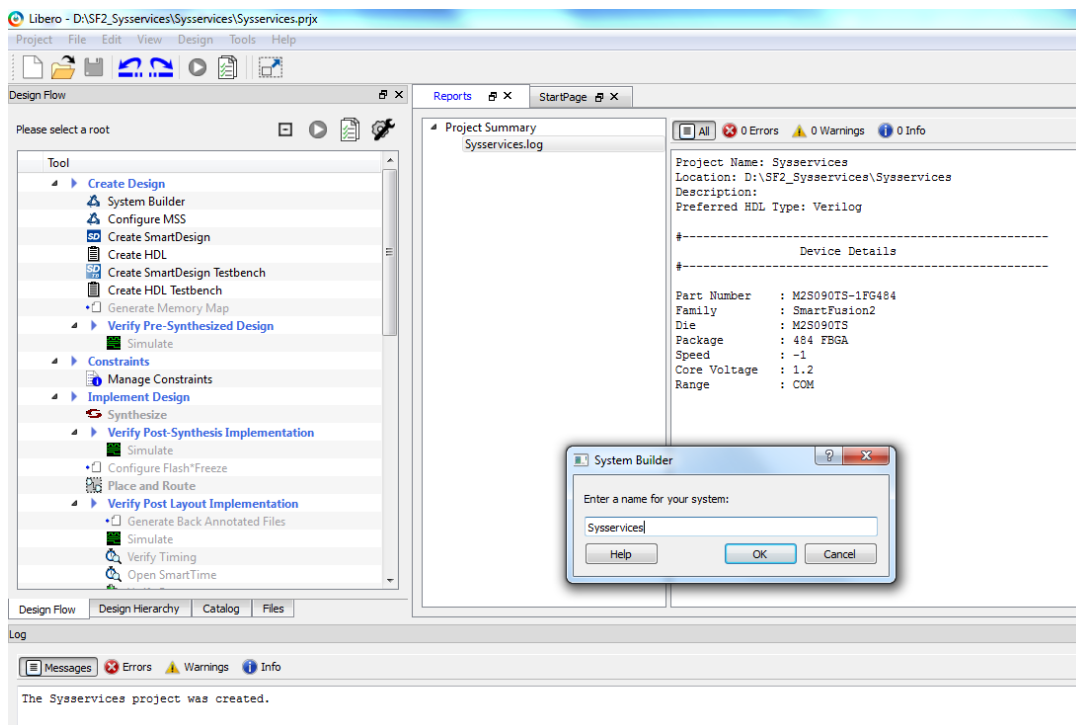
9. Click **Finish**. This displays the **New project information** window. Select **Use Enhanced Constraint Flow** to use the new constraint flow as part of Libero v11.7 SoC, as shown in [Figure 11](#).

Figure 11 • New Project Information Window



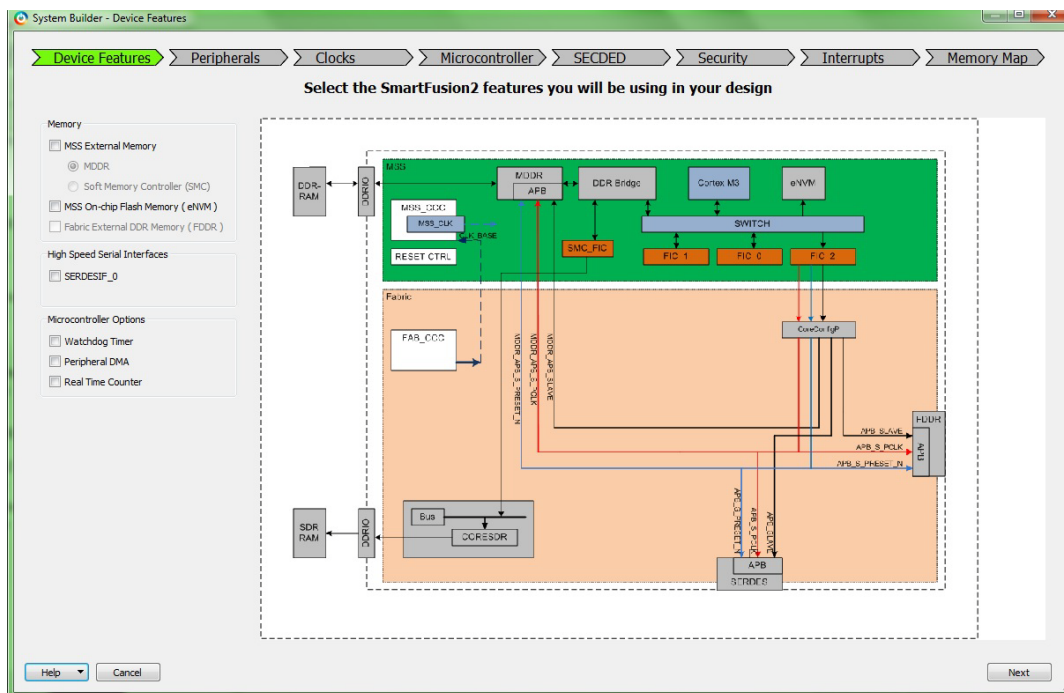
10. Enter **Sysservices** as the name of the system in the **System Builder** dialog box, as shown in Figure 12.

Figure 12 • System Builder Dialog Box



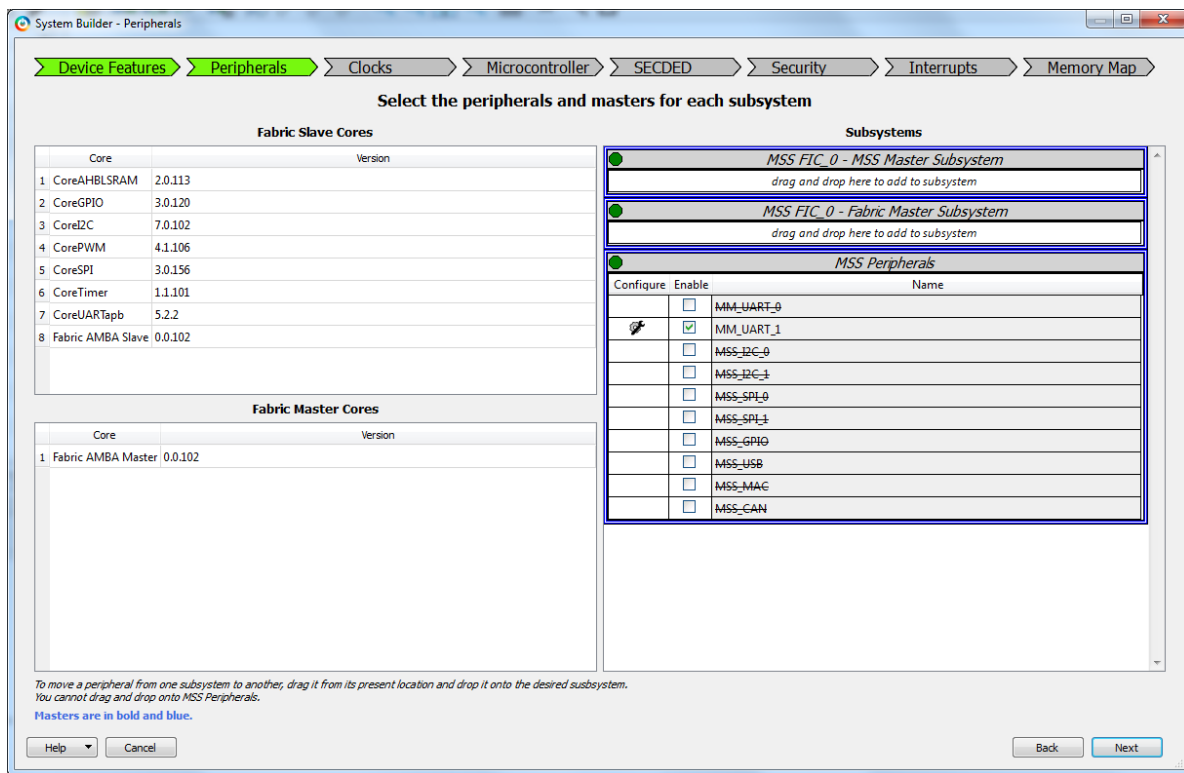
11. Click **OK**. The System Builder - Device Features page open by default, as shown in Figure 13.

Figure 13 • System Builders - Device Features Page



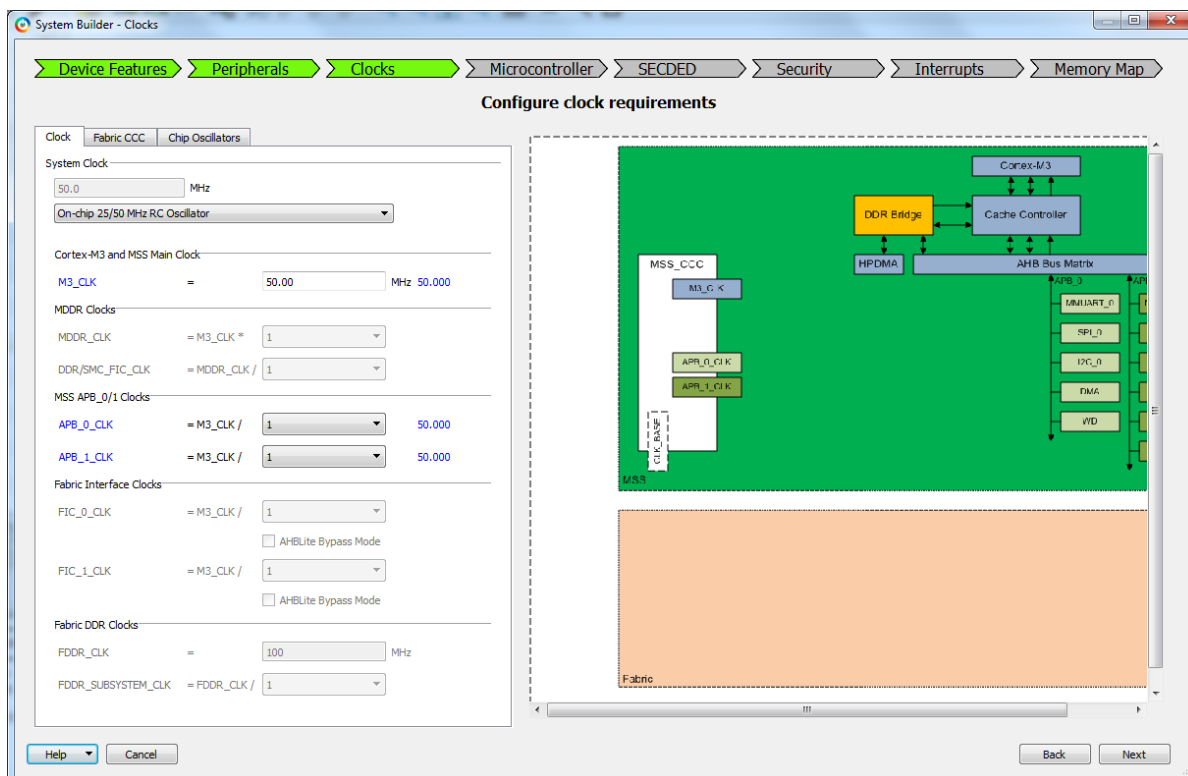
12. Click **Next**. This opens **System Builder - Peripherals** page.
13. Enable the **MM_UART_1** and disable all the peripherals, as shown in [Figure 14](#).

Figure 14 • System Builder - Peripherals Page



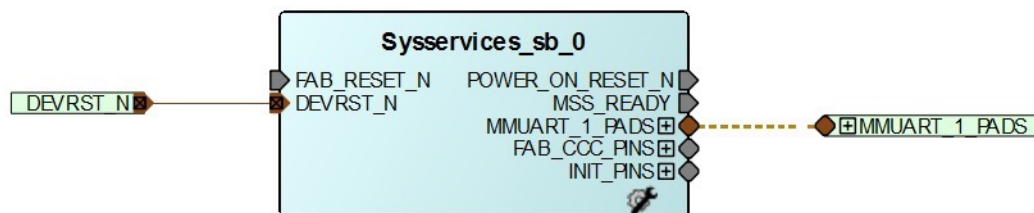
14. Click **Next**. This opens **System Builder - Clocks** page, as shown in Figure 15.
15. In the **System Builder - Clocks** page:
 - Select **System Clock** frequency as **50 MHz** and clock source as **On-chip 25/50 MHz RC Oscillator**
 - Select **M3_CLK** as **50 MHz**
 - Select **APB_0_CLK** and **APB_1_CLK** frequency as **M3_CLK/1**
 - Do not change the default settings of remaining parameters.

Figure 15 • System Builder - Clocks Page



16. Click **Next**. This opens **System Builder - Microcontroller** page. Do not change the default selections.
17. Click **Next**. This opens **System Builder - SECEDED** page. Do not change the default selections.
18. Click **Next**. This opens **System Builder - Security** page. Do not change the default selections.
19. Click **Next**. This opens **System Builder - Interrupts** page. Do not change the default selections.
20. Click **Next**. This opens **System Builder - Memory Map** page. Do not change the default selections.
21. Click **Finish**.
22. Select **File > Save** to save **Sysservices_sb_0**. Select the **Sysservices** tab on the Smart Design canvas, as shown in Figure 16.

Figure 16 • Updating Sysservices_sb_0



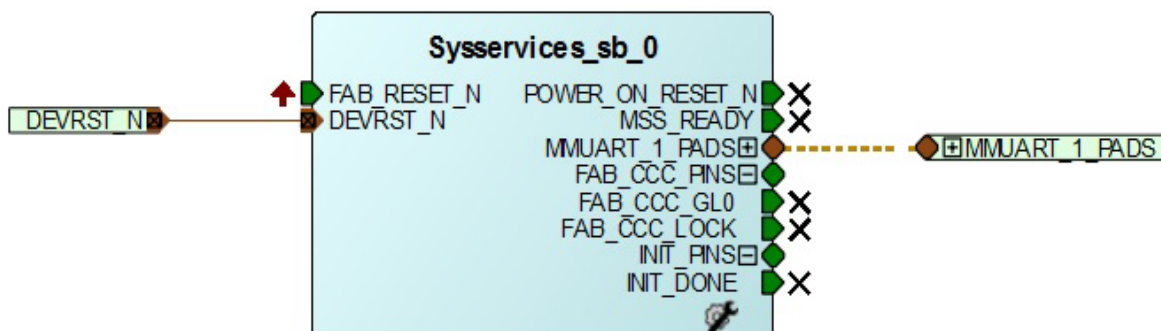
2.5.2 Connecting Components in Syssservices_sb_0 SmartDesign

The following steps describe how to connect the components in the **Sysservices_sb_0** 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 Syssservices_sb_0 design is displayed, as shown [Figure 17](#).

Figure 17 • Syssservices_sb_0

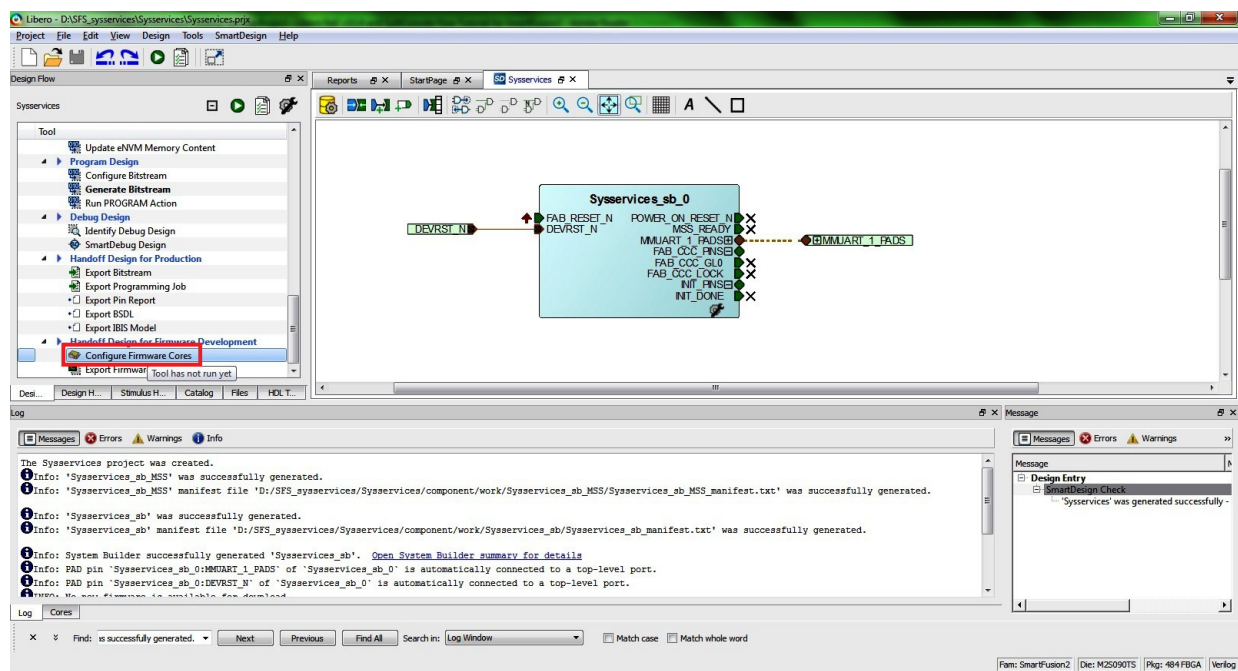


2.5.3 Configuring and Generating Firmware

The following steps describe how to configure and generate firmware:

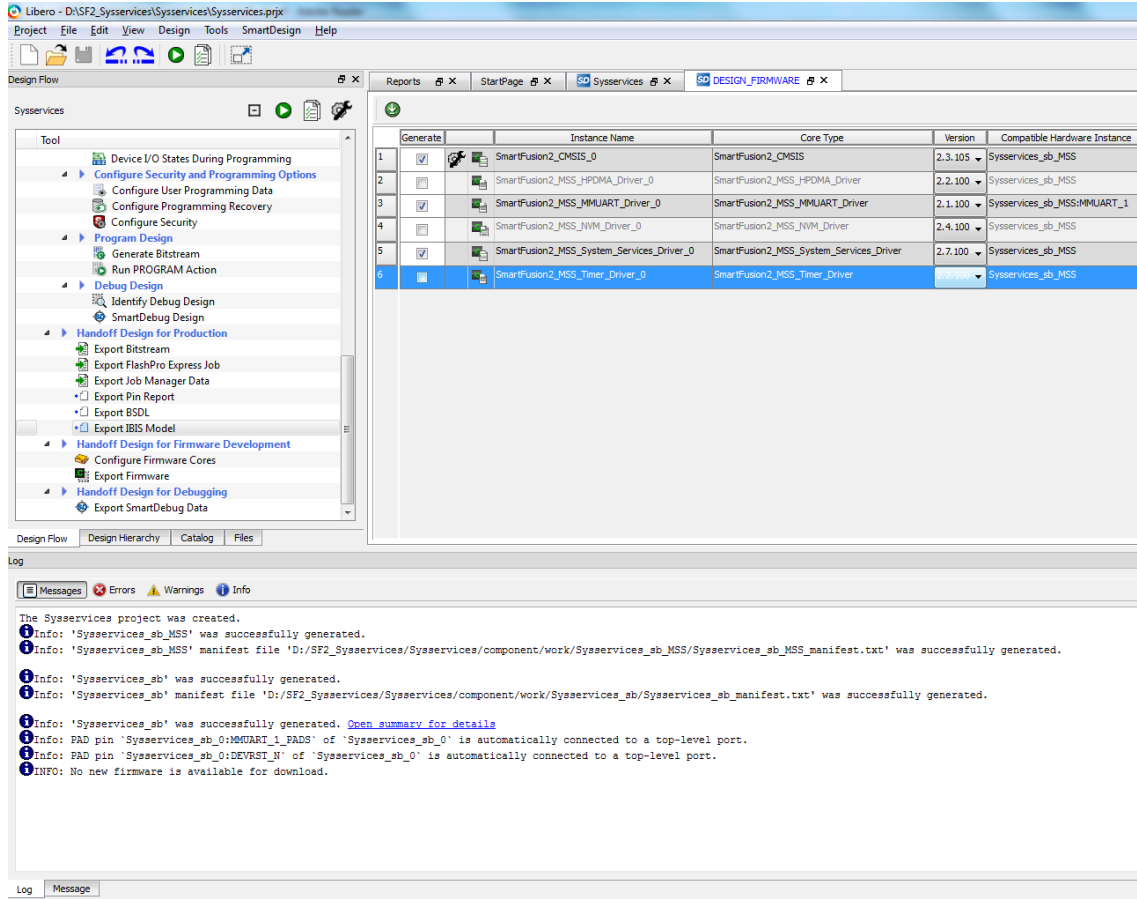
1. In the **Design Flow** tab, double-click **Configure Firmware Cores** under **Handoff design for Firmware Development**. This opens **DESIGN_FIRMWARE** window, as shown in Figure 18.

Figure 18 • Opening DESIGN_FIRMWARE Window



2. Clear all drivers except CMSIS, MMUART_0, and system services, as shown in [Figure 19](#).
- Note:** The SoftConsole sample project for system services driver can also be downloaded from **DESIGN_FIRMWARE** window. Right-click **SmartFusion2_MSS_System_Services_Driver_0** and select **Read Version Information**.

Figure 19 • DESIGN_FIRMWARE Window



The screenshot shows the Libero DESIGN_FIRMWARE window. On the left is a 'Tool' pane with a tree view of design tools. The main area displays a table of drivers. The bottom pane shows a log of messages.

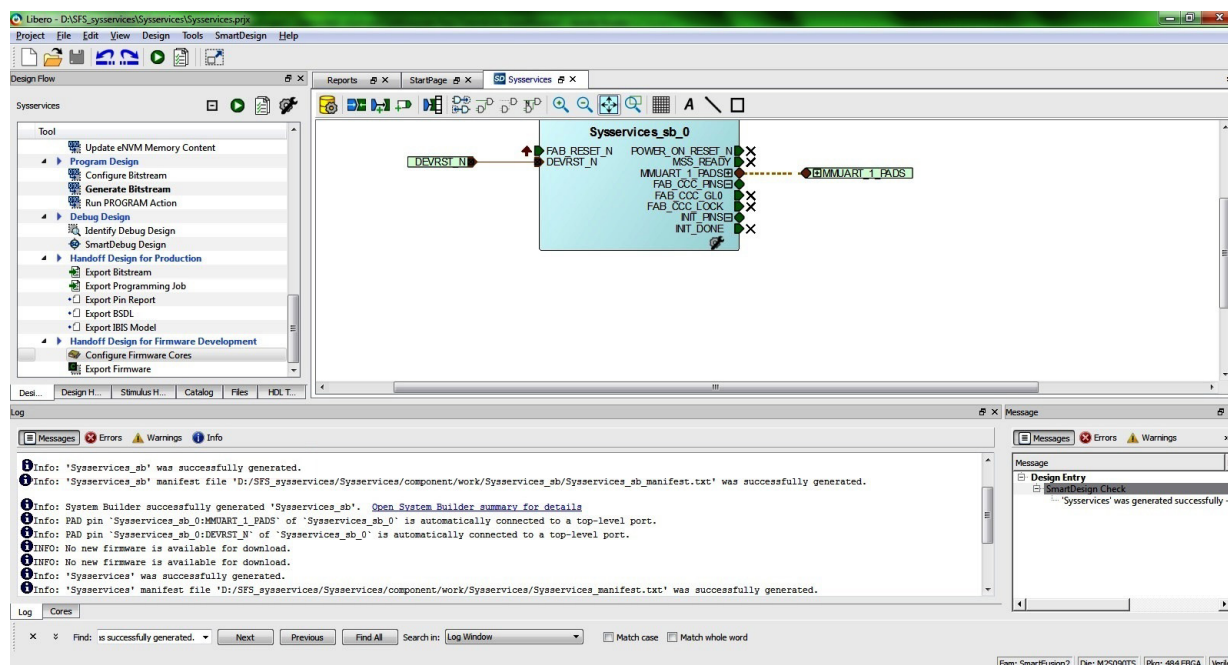
Generate	Instance Name	Core Type	Version	Compatible Hardware Instance
☒	SmartFusion2\CMSIS_0	SmartFusion2\CMSIS	2.3.105	Syservices_sb_MSS
☐	SmartFusion2_MSS_HPOMA_Driver_0	SmartFusion2_MSS_HPOMA_Driver	2.2.100	Syservices_sb_MSS
☒	SmartFusion2_MSS_MMUART_Driver_0	SmartFusion2_MSS_MMUART_Driver	2.1.100	Syservices_sb_MSS:MMUART_1
☐	SmartFusion2_MSS_NVM_Driver_0	SmartFusion2_MSS_NVM_Driver	2.4.100	Syservices_sb_MSS
☒	SmartFusion2_MSS_System_Services_Driver_0	SmartFusion2_MSS_System_Services_Driver	2.7.100	Syservices_sb_MSS
☐	SmartFusion2_MSS_Timer_Driver_0	SmartFusion2_MSS_Timer_Driver	2.7.100	Syservices_sb_MSS

Log:

```

The Syservices project was created.
Info: 'Syservices_sb_MSS' was successfully generated.
Info: 'Syservices_sb_MSS' manifest file 'D:/SF2_Syservices/Syservices/component/work/Syservices_sb_MSS/Syservices_sb_MSS_manifest.txt' was successfully generated.
Info: 'Syservices_sb' was successfully generated.
Info: 'Syservices_sb' manifest file 'D:/SF2_Syservices/Syservices/component/work/Syservices_sb/Syservices_sb_manifest.txt' was successfully generated.
Info: 'Syservices_sb' was successfully generated. Open summary for details
Info: PAD pin 'Syservices_sb_0:MMUART_1_PADS' of 'Syservices_sb_0' is automatically connected to a top-level port.
Info: PAD pin 'Syservices_sb_0:DEVRST_N' of 'Syservices_sb_0' is automatically connected to a top-level port.
Info: No new firmware is available for download.
    
```

Figure 20 • Generate Component



After successful generation of all the components, the following message is displayed on the log window, as shown in Figure 20.

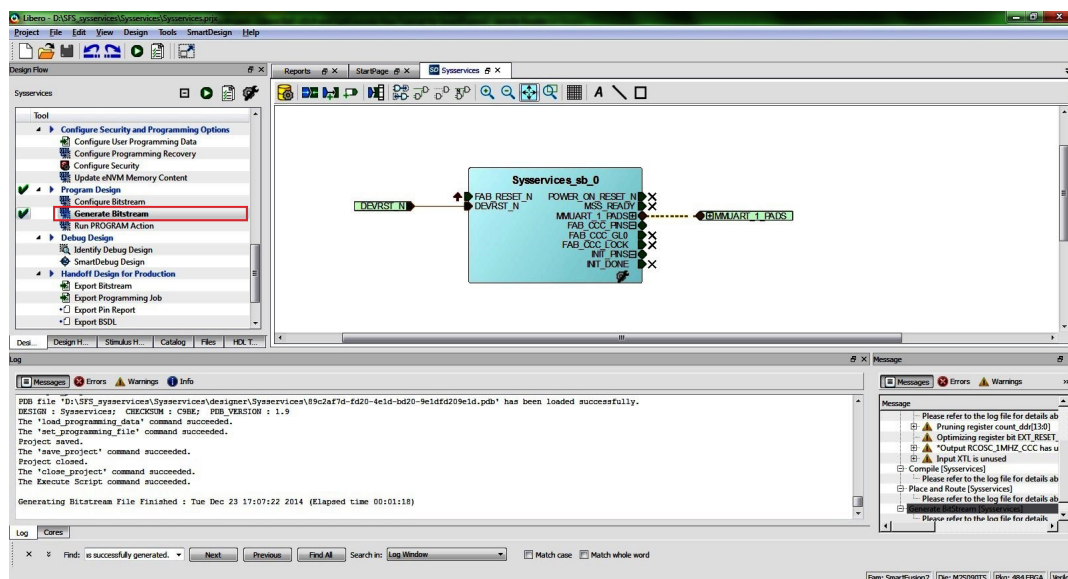
Info: 'Syservices' was successfully generated.

2.6 Generating the Program File

The following steps describe how to generate the program file:

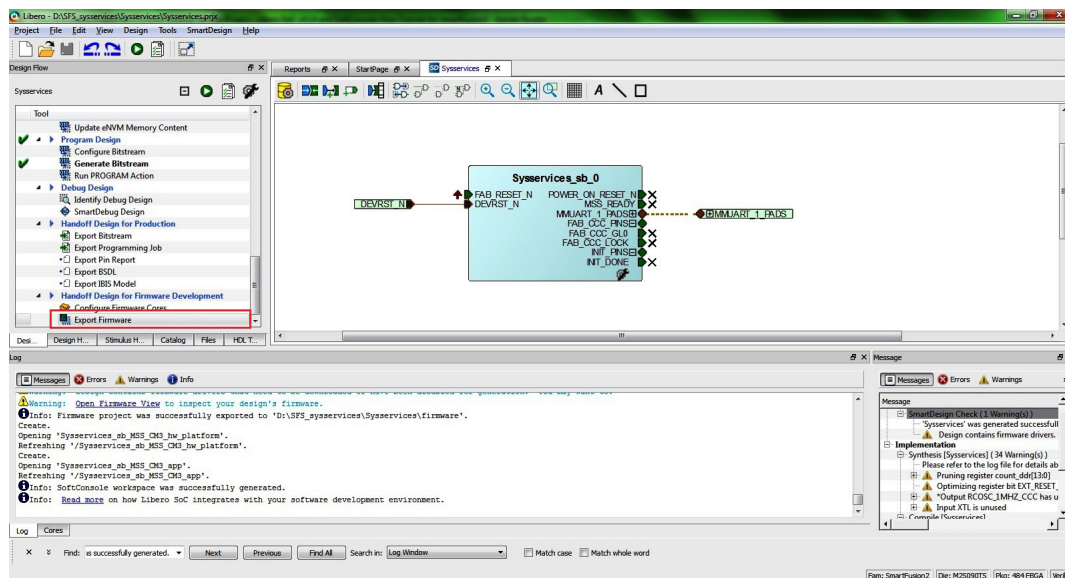
1. Click **Generate Bitstream**, to complete place-and-route and generate the programming file, as shown in Figure 21.

Figure 21 • Generate Bitstream Data



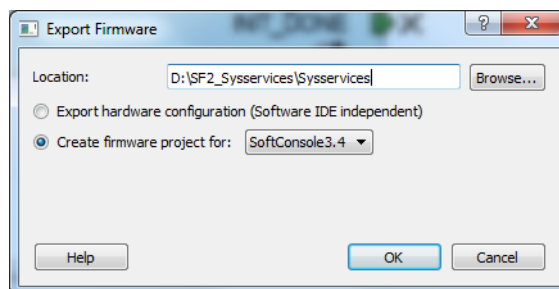
2. Click **Export Firmware**. This opens **Export Firmware** dialog box, as shown in Figure 23 on page 23.

Figure 22 • Export Firmware



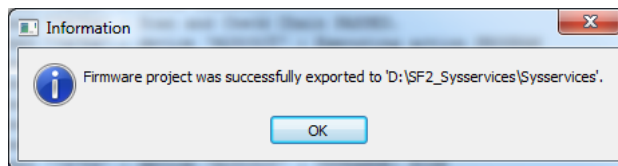
3. In the **Export Firmware** dialog box:
 - Select **Create firmware project for**.
 - Select **SoftConsole3.4** from the drop-down list.

Figure 23 • Export Firmware Dialog Box



4. Click **OK**. The successful firmware generation window is displayed, as shown in Figure 24.

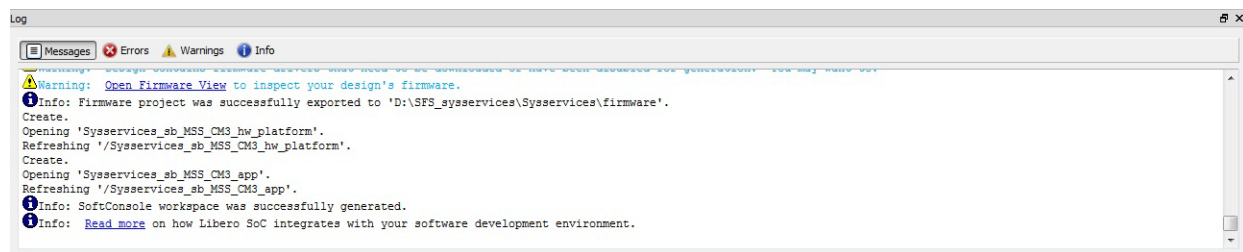
Figure 24 • Firmware Successfully Exported Message



5. Click **OK**.

The log window is displayed, as shown in [Figure 25](#).

Figure 25 • Firmware Log Window



2.7 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 board.
2. Connect the jumpers on the SmartFusion2 Security Evaluation Kit board, as per [Table 2](#). For more information on jumper locations, refer to "[Appendix: Jumper Locations](#)" on page 38.
CAUTION: Ensure that the power supply switch, SW7 is switched OFF while connecting the jumpers on the SmartFusion2 Security Evaluation Kit.

Table 2 • SmartFusion2 Security Evaluation Kit Jumper Settings

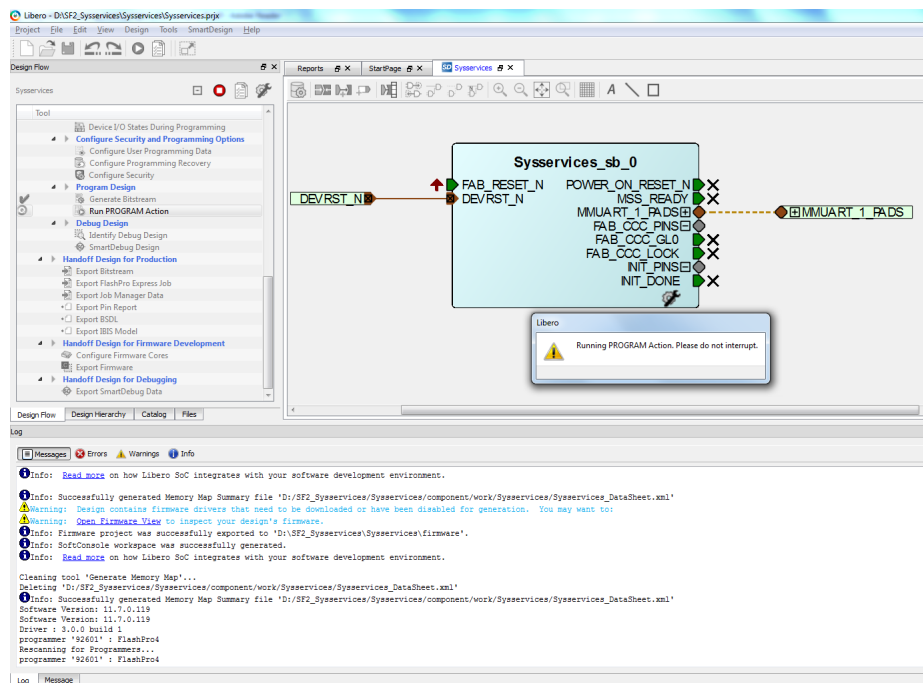
Jumper Number	Pin (From)	Pin (To)	Comments
J22, J23, J24, J8, J3	1	2	These are the default jumper settings of the SmartFusion2 Security Evaluation Kit board. Ensure that these jumpers are set accordingly.

3. Connect the power supply to the J6 connector.
4. Switch ON the power supply switch, SW7. Refer to "[Appendix: Board Setup for Running the Tutorial](#)" on page 39 for information on the board setup for running the tutorial.
5. Program the SmartFusion2 Security Evaluation Kit board with the programming file at <download_folder>\m2s_tu0487_creating_libero_project_libero11p7_d\Programmingfile\Sysservices_top, using the FlashPro software.

Note: This step is required if the *.stp file is used in the design folder.

6. To program the SmartFusion2 device, double-click **Run PROGRAM Action** in the **Design Flow** tab, as shown in [Figure 26](#). This programs the *.stp file to the board.

Figure 26 • Run Program Action



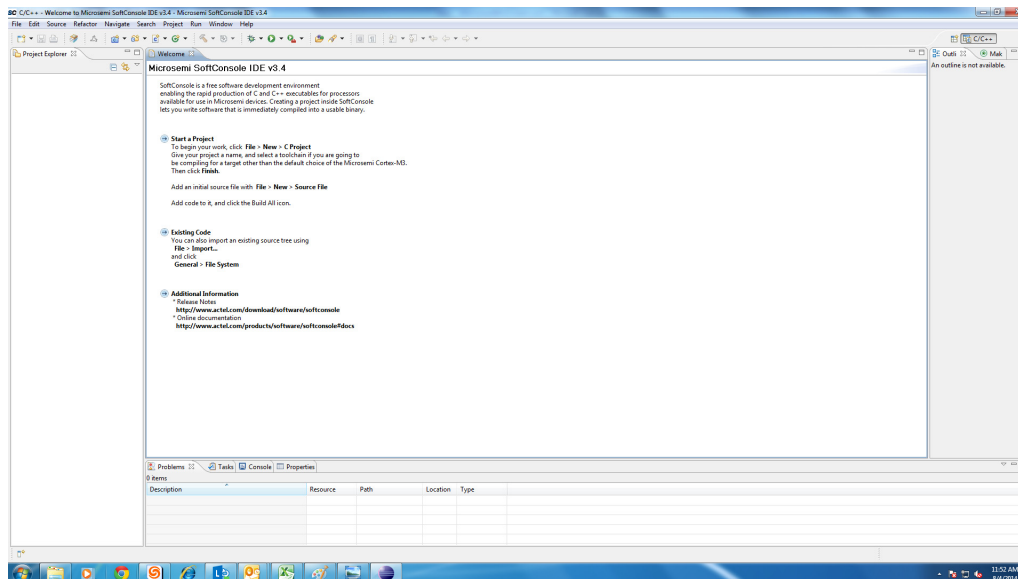
7. After successful programming, press **SW6** switch to reset the board.

2.8 Building Software Application Using SoftConsole

The following steps describe how to build a software application using SoftConsole:

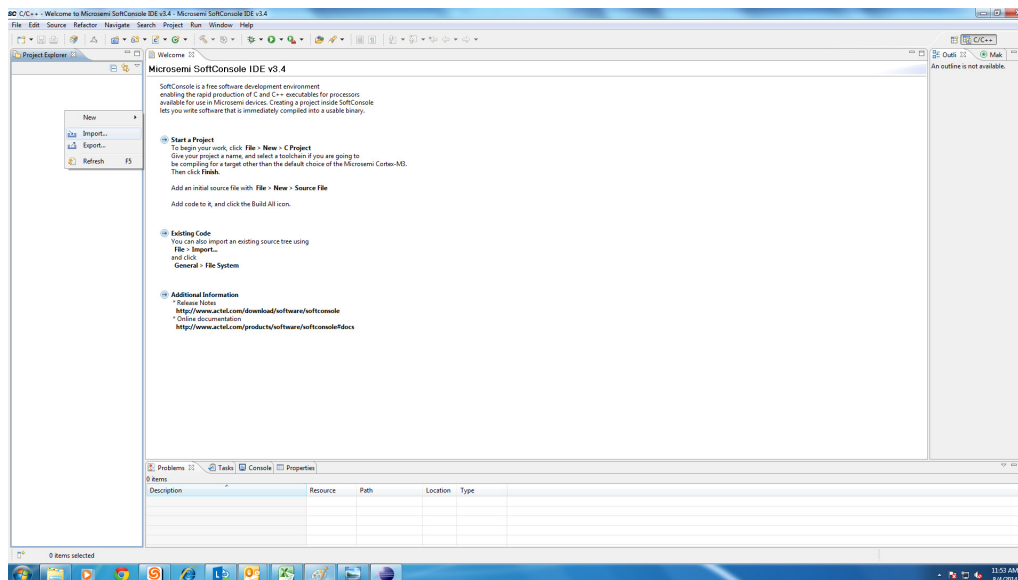
1. Open the standalone SoftConsole IDE.

Figure 27 • Invoking SoftConsole IDE



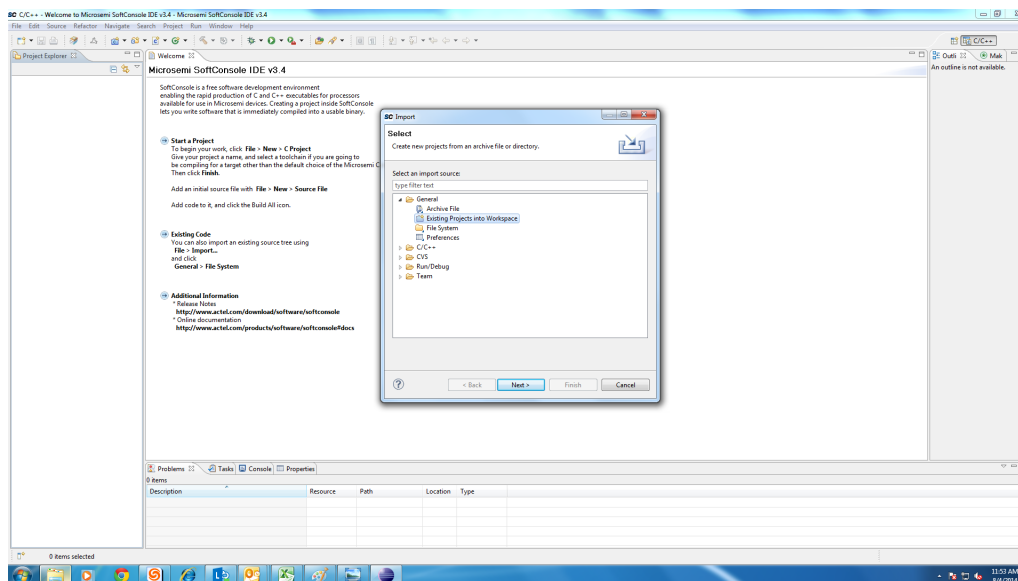
2. Right-click **Project Explorer** window and select **Import** option, as shown in Figure 28.

Figure 28 • Importing Projects



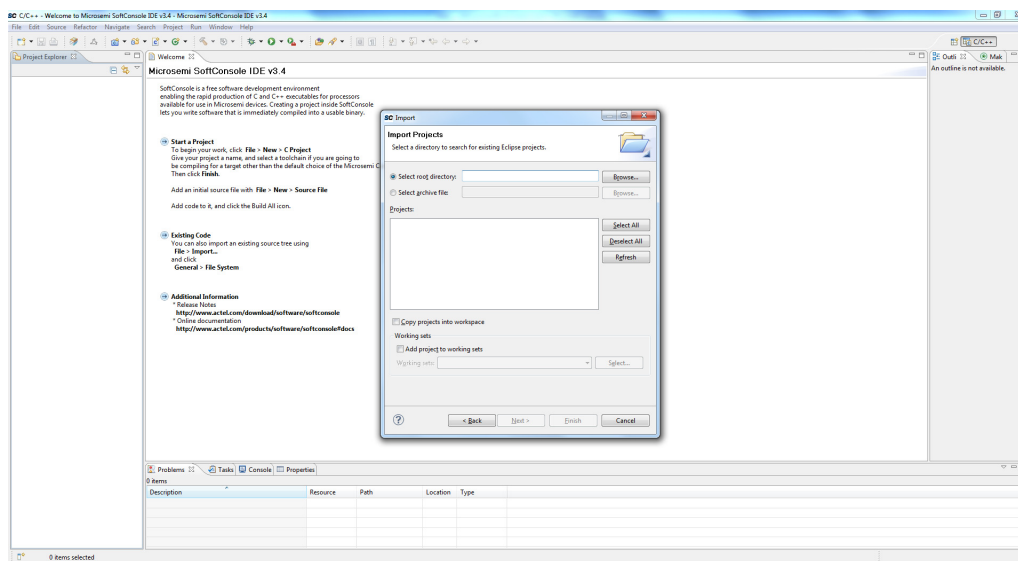
3. Select **General > Existing Projects into Workspace**, as shown in Figure 29.

Figure 29 • Importing Existing Projects



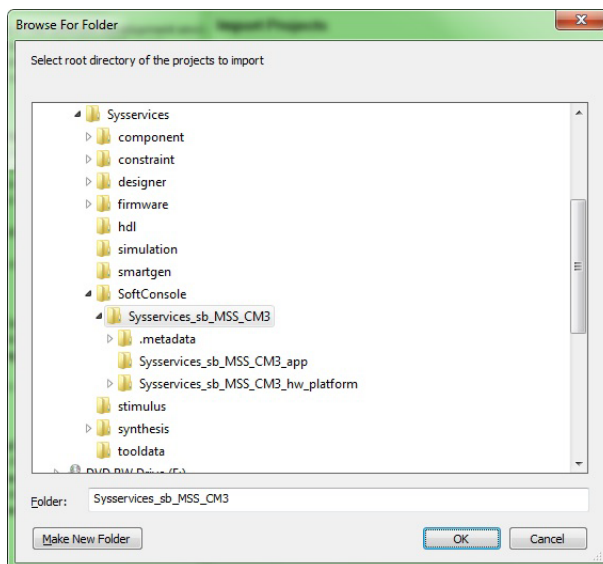
Import window is displayed, as shown in Figure 30.

Figure 30 • Import Window



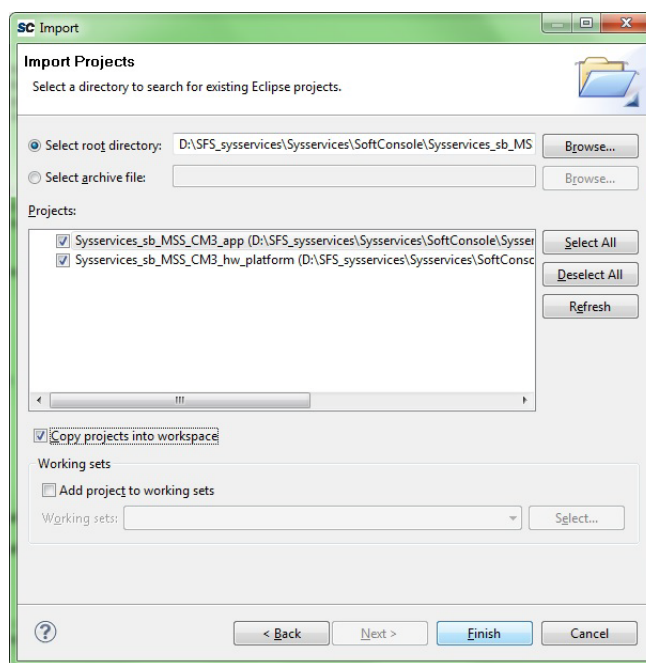
4. Browse through the Syssservices projects folder and select, as shown in Figure 31.

Figure 31 • Selecting System services



5. Click **OK**. This opens the **Import** window, as shown in Figure 32.

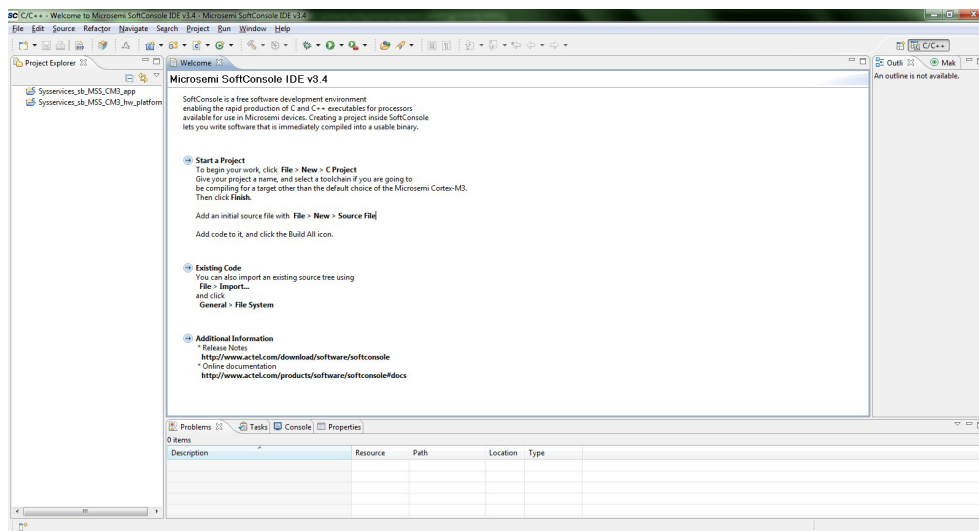
Figure 32 • Adding Projects to SoftConsole IDE



6. Click **Finish**.

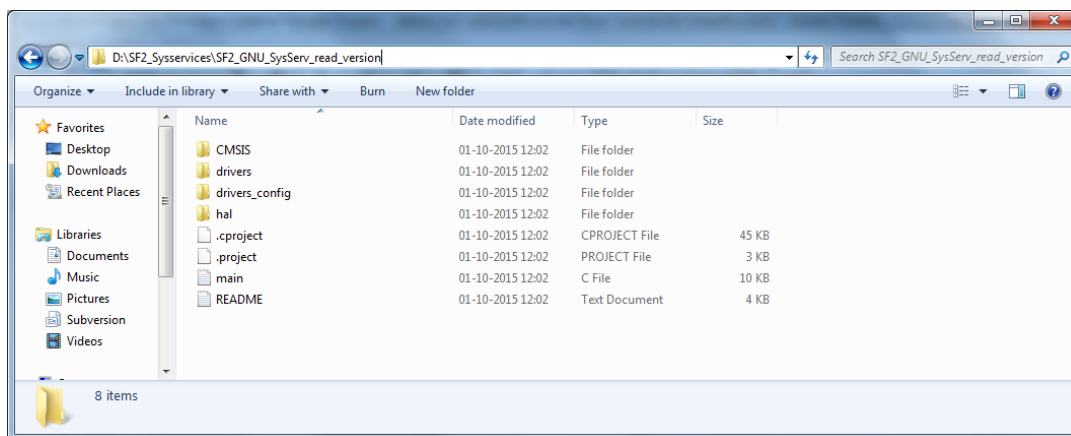
The SoftConsole perspective is displayed, as shown in Figure 33.

Figure 33 • SoftConsole Workspace



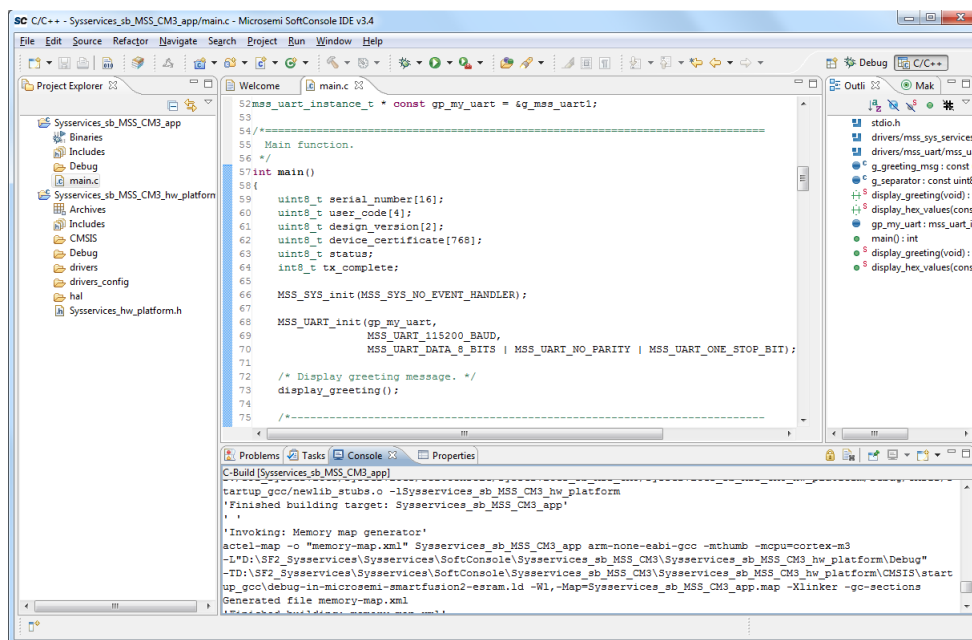
- Go to the location where the SoftConsole sample firmware catalog project is saved, as shown in Figure 34.

Figure 34 • Sample Project main. c File



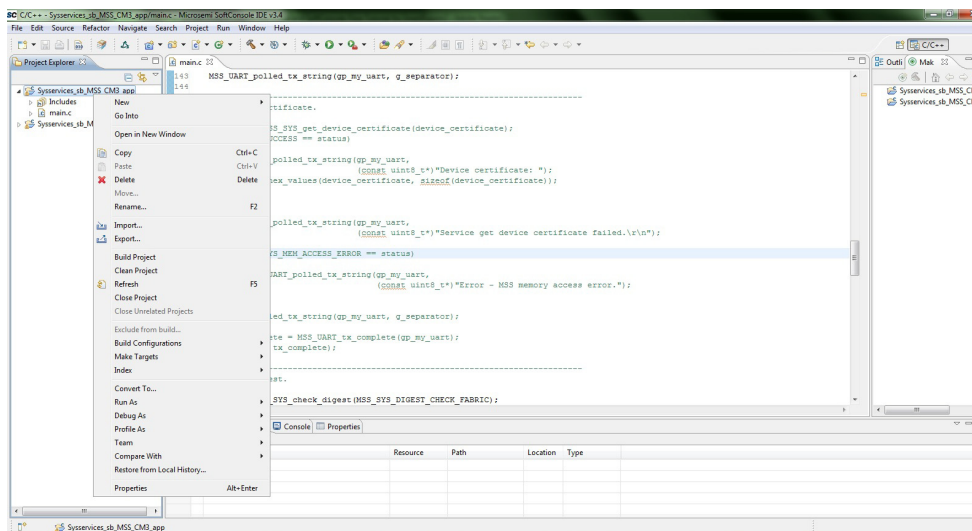
8. Copy the `main.c` file and replace it with the existing `main.c` file under **Sysservices_sb_MSS_CM3** project in the SoftConsole workspace. The SoftConsole window is shown in [Figure 35](#).

Figure 35 • SoftConsole Workspace - main.c File



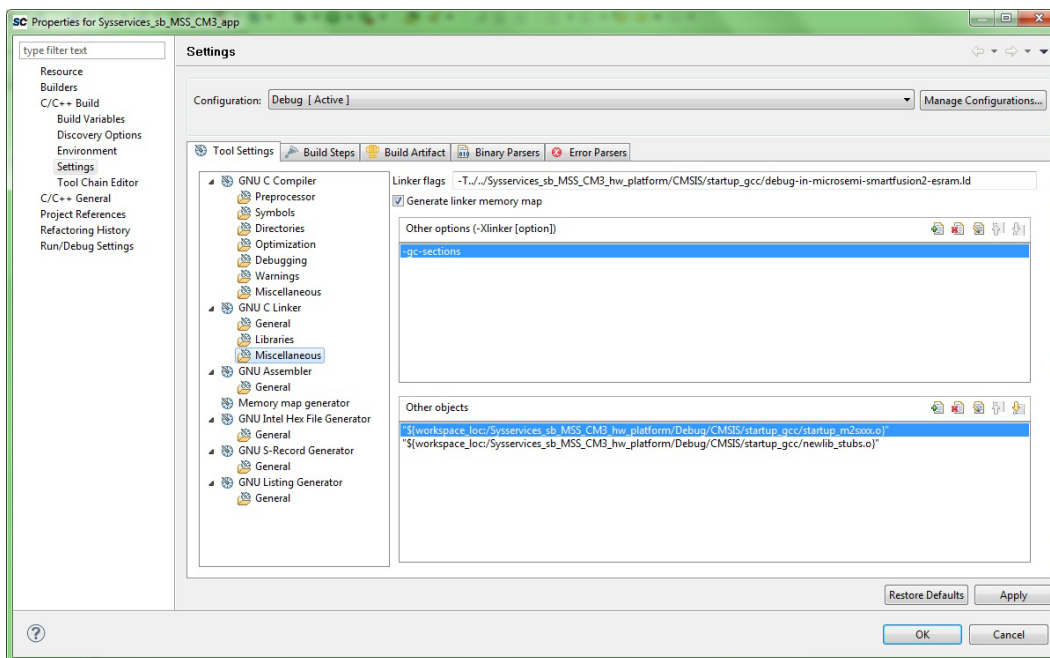
9. Right-click **Sysservices_sb_MSS_CM3** in the **Project Explorer** window of the SoftConsole project and select **Properties**, as shown in [Figure 36](#).

Figure 36 • Project Explorer Window - SoftConsole Project



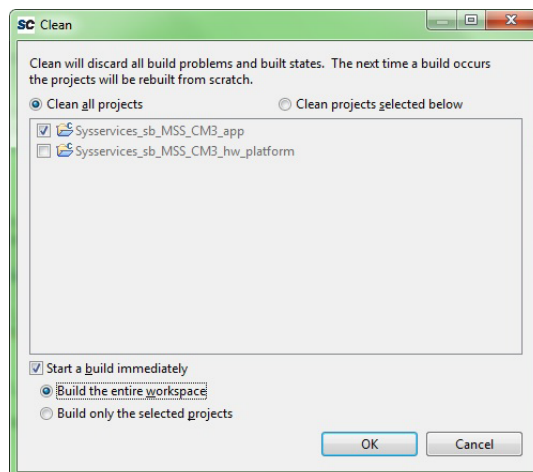
10. In the **Properties** window, go to **Settings** under **C/C++ Build** and select **GNU C linker** as **debug-in-microsemi-smartfusion2-esram.ld**, as shown in [Figure 37](#). Click **Apply** and then **OK**.

Figure 37 • Syservices_sb_MSS_CM3 Properties Window



11. Perform a clean build by selecting **Project > Clean**. Accept the default settings in the **Clean** dialog box and click **OK**, as shown in Figure 38. The SoftConsole project must not have any errors.

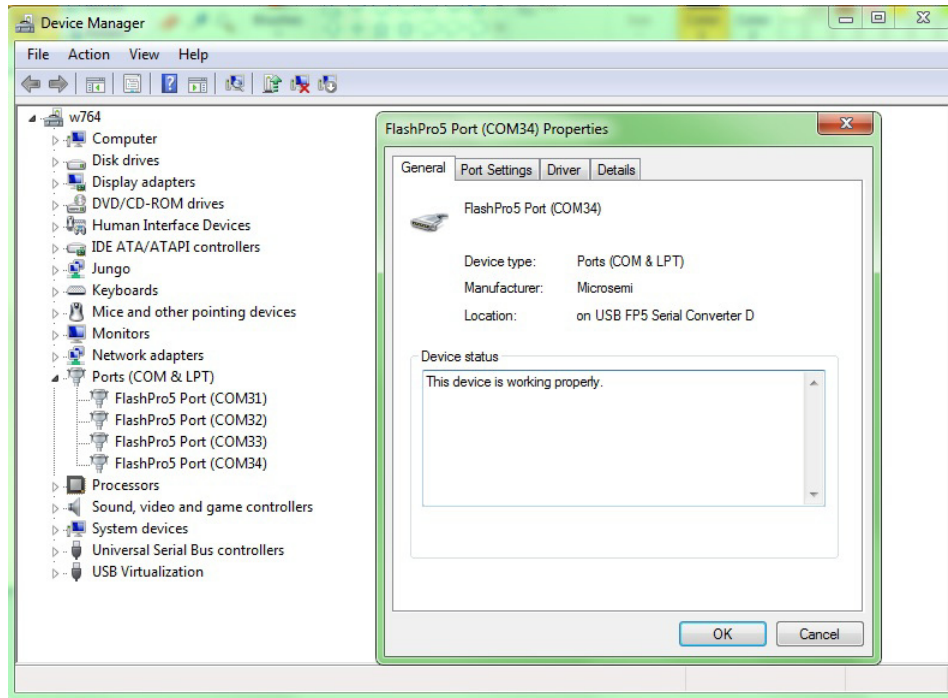
Figure 38 • Settings for Clean Build



12. 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

13. Connect the host PC to the J18 connector using the USB min-B cable. The USB to UART bridge drivers are automatically detected. Verify if the detection is made in the device manager, as shown in [Figure 39](#).

Figure 39 • Device Manager Window



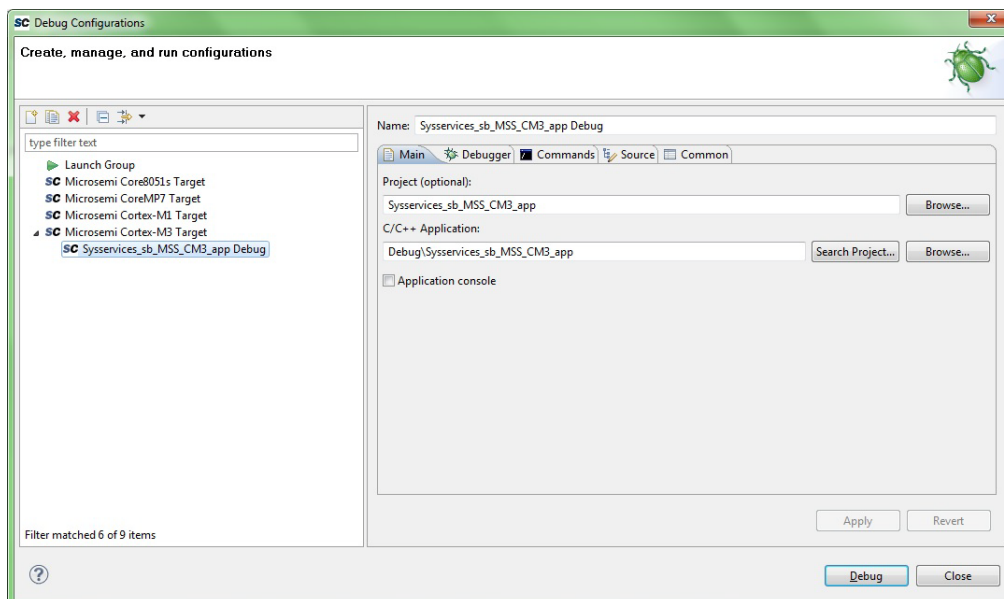
14. Start the PuTTY session. If the PuTTY program is not available in the computer system, use any free serial terminal emulation program such as HyperTerminal or TeraTerm. Refer to the [Configuring Serial Terminal Emulation Programs Tutorial](#) for configuring the HyperTerminal, TeraTerm, or PuTTY.

The PuTTY settings are as follows:

- 115,200 baud rate
- 8 data bits
- 1 stop bit
- No parity
- No flow control

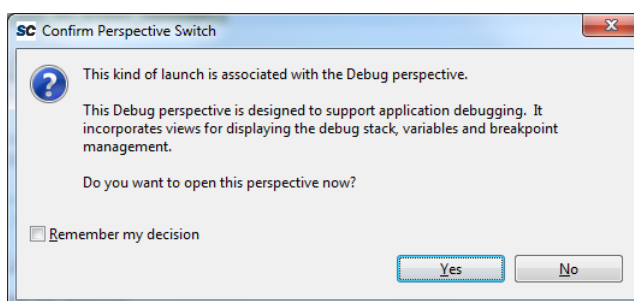
15. Select **Debug Configurations** from the **Run** menu of the SoftConsole. The **Debug** dialog box is displayed. Double-click on **Microsemi Cortex-M3 Target** as shown in Figure 40.

Figure 40 • Debug Configurations Window



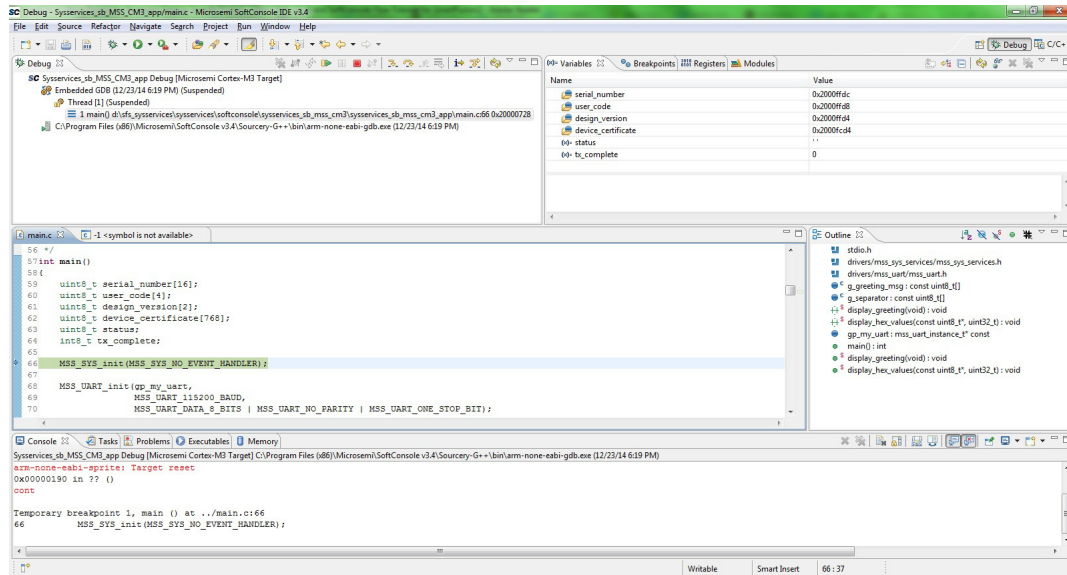
16. Ensure that the following information appears on the **Main** tab in the **Debug Configurations** window, and click **Debug**:
 - **Name:** Syservices_sb_MSS_ CM3 Debug
 - **Project:** Syservices_sb_MSS_ CM3_app
 - **C/C++ Application:** Debug\Syservices_sb_MSS_CM3_app
17. Click **Yes** when prompted for the **Confirm Perspective Switch**, as shown in Figure 41. This displays the debug view mode.

Figure 41 • Confirm Perspective Switch



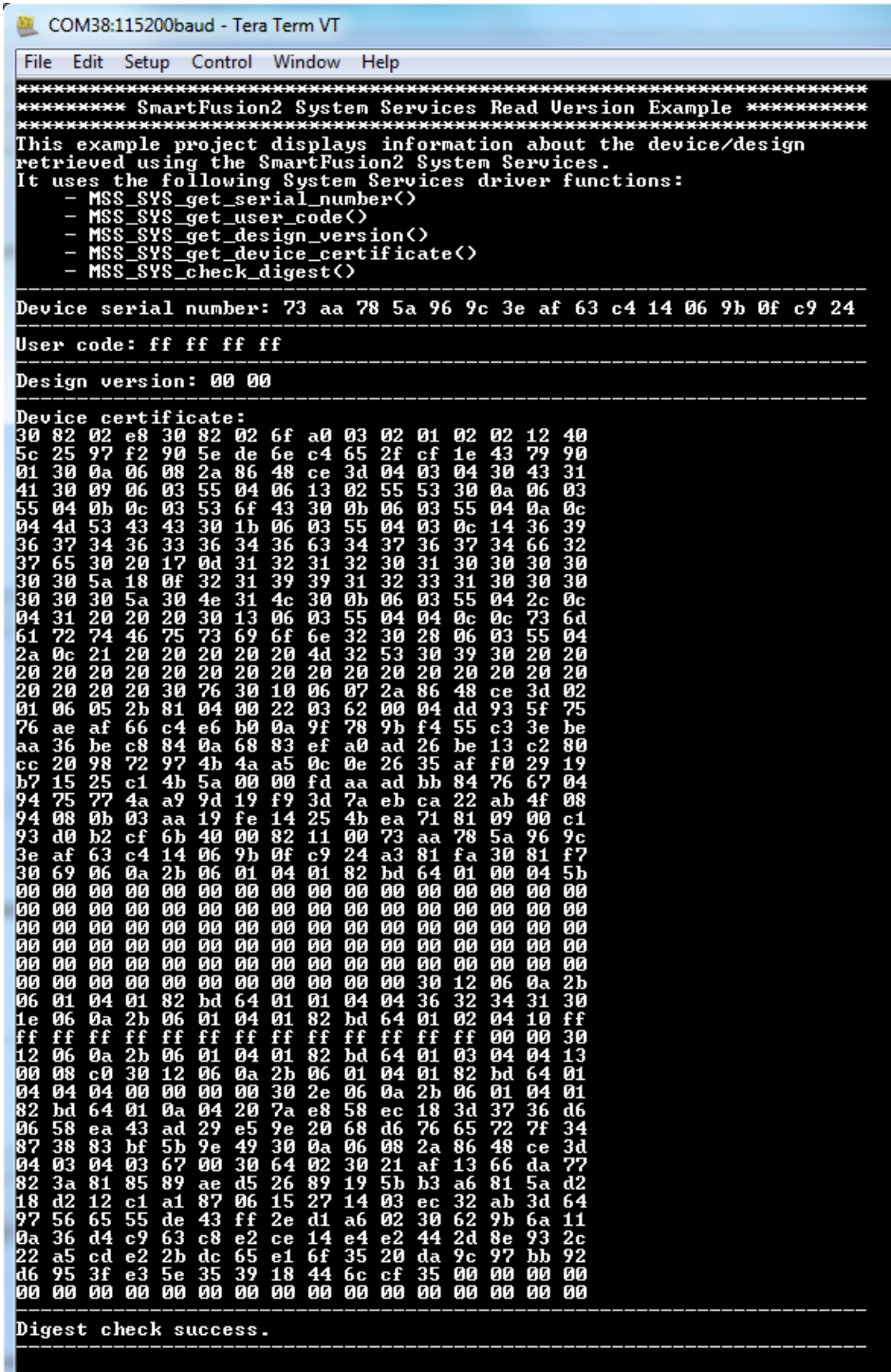
The SoftConsole Debugger Perspective window is displayed, as shown in Figure 42.

Figure 42 • SoftConsole Debugger Perspective



18. Run the application by clicking **Run > Resume**. The Information on the SmartFusion2 device and design, along with a greeting message is displayed on the PuTTY, as shown in Figure 43.

Figure 43 • PuTTY Window



```

COM38:115200baud - Tera Term VT
File Edit Setup Control Window Help
***** SmartFusion2 System Services Read Version Example *****
*****
This example project displays information about the device/design
retrieved using the SmartFusion2 System Services.
It uses the following System Services driver functions:
- MSS_SYS_get_serial_number()
- MSS_SYS_get_user_code()
- MSS_SYS_get_design_version()
- MSS_SYS_get_device_certificate()
- MSS_SYS_check_digest()

-----
Device serial number: 73 aa 78 5a 96 9c 3e af 63 c4 14 06 9b 0f c9 24
-----
User code: ff ff ff ff
-----
Design version: 00 00
-----
Device certificate:
30 82 02 e8 30 82 02 6f a0 03 02 01 02 02 12 40
5c 25 97 f2 90 5e de 6e c4 65 2f cf 1e 43 79 90
01 30 0a 06 08 2a 86 48 ce 3d 04 03 04 30 43 31
41 30 09 06 03 55 04 06 13 02 55 53 30 0a 06 03
55 04 0b 0c 03 53 6f 43 30 0b 06 03 55 04 0a 0c
04 4d 53 43 43 30 1b 06 03 55 04 03 0c 14 36 39
36 37 34 36 33 36 34 36 63 34 37 36 37 34 66 32
37 65 30 20 17 0d 31 32 31 32 30 31 30 30 30 30
30 30 5a 18 0f 32 31 39 39 31 32 33 31 30 30 30
30 30 30 5a 30 4e 31 4c 30 0b 06 03 55 04 2c 0c
04 31 20 20 20 30 13 06 03 55 04 04 0c 0c 73 6d
61 72 74 46 75 73 69 6f 6e 32 30 28 06 03 55 04
2a 0c 21 20 20 20 20 20 4d 32 53 30 39 30 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 30 76 30 10 06 07 2a 86 48 ce 3d 02
01 06 05 2b 81 04 00 22 03 62 00 04 dd 93 5f 75
76 ae af 66 c4 e6 b0 0a 9f 78 9b f4 55 c3 3e be
aa 36 be c8 84 0a 68 83 ef a0 ad 26 be 13 c2 80
cc 20 98 72 97 4b 4a a5 0c 0e 26 35 af f0 29 19
b7 15 25 c1 4b 5a 00 00 fd aa ad bb 84 76 67 04
94 75 77 4a a9 9d 19 f9 3d 7a eb ca 22 ab 4f 08
94 08 0b 03 aa 19 fe 14 25 4b ea 71 81 09 00 c1
93 d0 b2 cf 6b 40 00 82 11 00 73 aa 78 5a 96 9c
3e af 63 c4 14 06 9b 0f c9 24 a3 81 fa 30 81 f7
30 69 06 0a 2b 06 01 04 01 82 bd 64 01 00 04 5b
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
06 01 04 01 82 bd 64 01 01 04 04 36 32 34 31 30
1e 06 0a 2b 06 01 04 01 82 bd 64 01 02 04 10 ff
ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
12 06 0a 2b 06 01 04 01 82 bd 64 01 03 04 04 13
00 08 c0 30 12 06 0a 2b 06 01 04 01 82 bd 64 01
04 04 04 00 00 00 00 30 2e 06 0a 2b 06 01 04 01
82 bd 64 01 0a 04 20 7a e8 58 ec 18 3d 37 36 d6
06 58 ea 43 ad 29 e5 9e 20 68 d6 76 65 72 7f 34
87 38 83 bf 5b 9e 49 30 0a 06 08 2a 86 48 ce 3d
04 03 04 03 67 00 30 64 02 30 21 af 13 66 da 77
82 3a 81 85 89 ae d5 26 89 19 5b b3 a6 81 5a d2
18 d2 12 c1 a1 87 06 15 27 14 03 ec 32 ab 3d 64
97 56 65 55 de 43 ff 2e d1 a6 02 30 62 9b 6a 11
0a 36 d4 c9 63 c8 e2 ce 14 e4 e2 44 2d 8e 93 2c
22 a5 cd e2 2b dc 65 e1 6f 35 20 da 9c 77 bb 92
d6 95 3f e3 5e 35 39 18 44 6c cf 35 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

-----
Digest check success.
-----

```

19. Terminate execution of the code by choosing **Run > Terminate**.
20. Close **Debug Perspective** by selecting **Close Perspective** from the **Window** menu.
21. Close SoftConsole using **File > Exit**.
22. Close the PuTTY. Click **Yes** when prompted for closing.

2.9 Conclusion

This tutorial describes how to download the SoftConsole sample project from the firmware catalog and how to create a Libero SoC project. It explains the procedure to generate the programming file and to run the SoftConsole project on the SmartFusion2 Security Evaluation Kit. A sample project for implementing system services features is created to display the SmartFusion2 device and design information.

Figure 44 shows the jumper locations in the SmartFusion2 Security Evaluation Kit board.

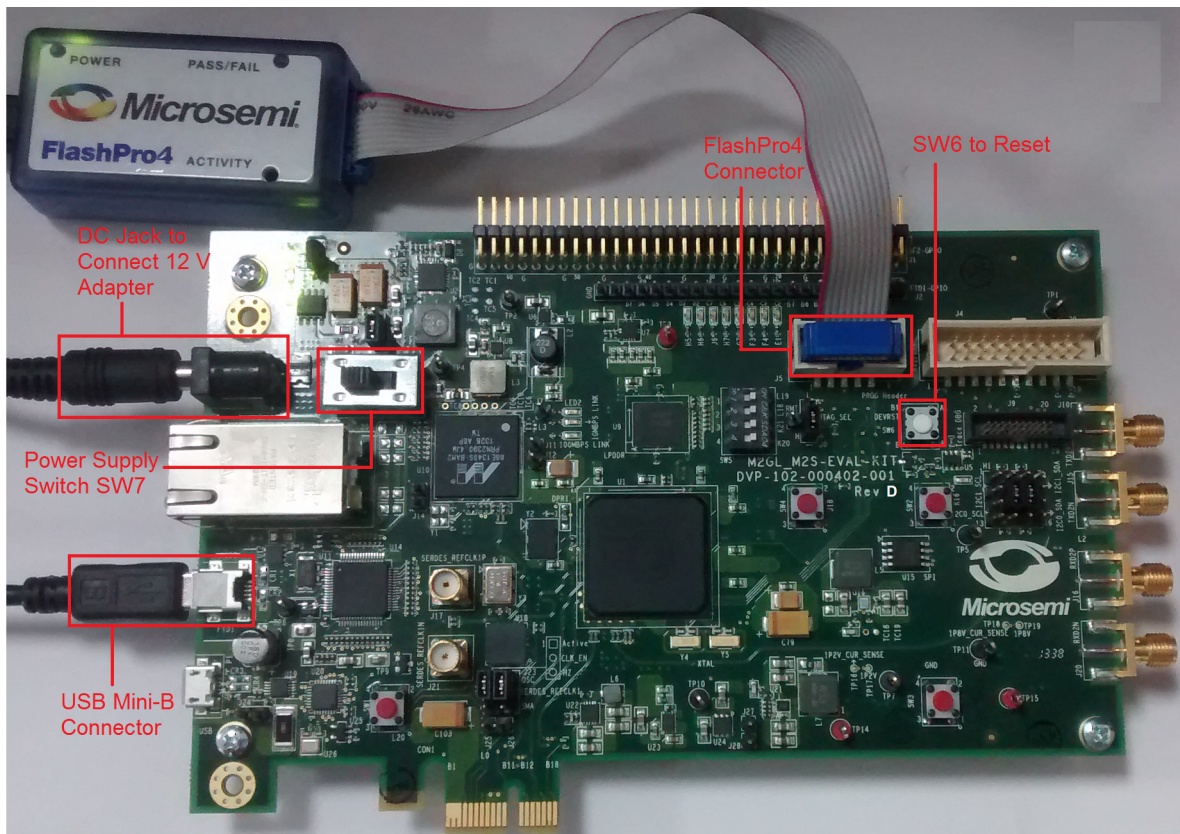
[illegible]

Revision 5

4 Appendix: Board Setup for Running the Tutorial

Figure 45 shows the board setup for running the tutorial on the SmartFusion2 Security Evaluation Kit board.

Figure 45 • SmartFusion2 Security Evaluation Kit



5 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 v11.7 software release (SAR 78428).
Revision 4 (October 2015)	Updated the document for Libero v11.6 software release (SAR 72552).
Revision 3 (February 2015)	Updated the document for Libero v11.5 software release (SAR 64799).
Revision 2 (October 2014)	Updated the document for Libero v11.4 software release (SAR 61636).
Revision 1 (April 2014)	Initial release.

6 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.

6.1 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

6.2 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.

6.3 Technical Support

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

6.4 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>.

6.5 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.

6.5.1 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.

6.5.2 My Cases

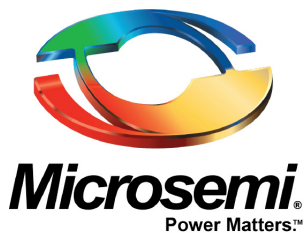
Microsemi SoC Products Group customers may submit and track technical cases online by going to [My Cases](#).

6.5.3 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](#).

6.6 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.