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

TU0487 Tutorial

Microsemi®

Power Matters.™

Contents

Figures

Tables

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

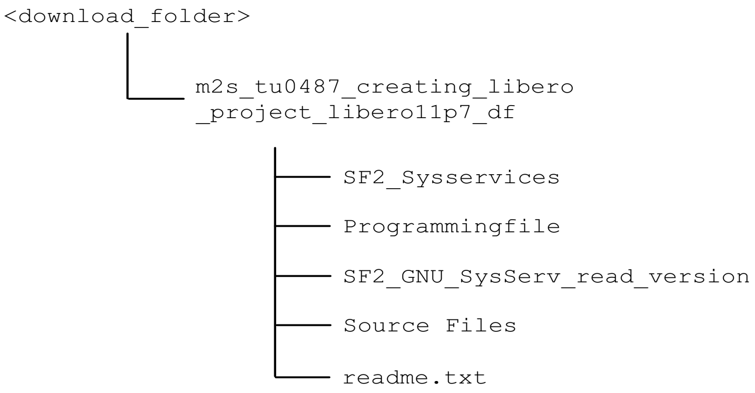
The design files for this tutorial can be downloaded from the Microsemi website: *http://soc.microsemi.com/download/rsc/?f=m2s_tu0487_creating_libero_project_libero11p7_df*

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.

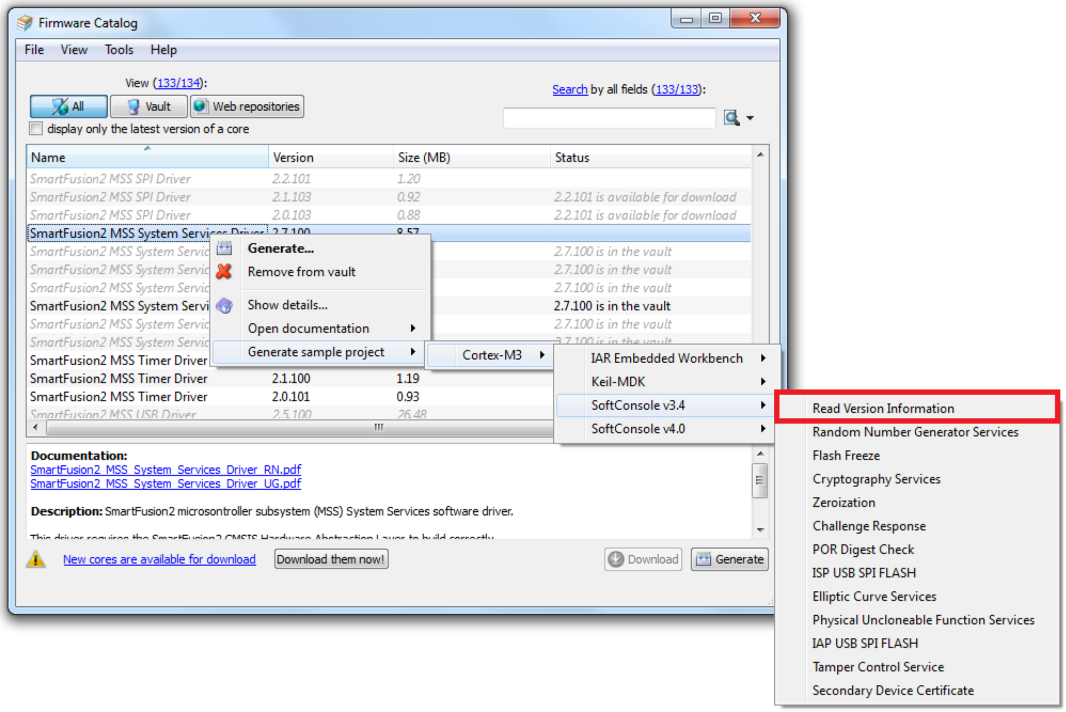
System services Information is displayed on HyperTerminal using MMUART_1 interface. For more information on system services, refer to the *UG0450: SmartFusion2 FPGA SoC and IGLOO2 FPGA System Controller User Guide*.

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**



□ display only the latest version of a core

Name	Version	Size (MB)	Status
SmartFusion2 MSS SPI Driver	2.2.101	1.20	
SmartFusion2 MSS SPI Driver	2.1.103	0.92	2.2.101 is available for download
SmartFusion2 MSS SPI Driver	2.0.103	0.88	2.2.101 is available for download
SmartFusion2 MSS System Services Driver	2.7.100	8.57	
SmartFusion2 MSS System Servic			2.7.100 is in the vault
SmartFusion2 MSS System Servic			2.7.100 is in the vault
SmartFusion2 MSS System Servic			2.7.100 is in the vault
SmartFusion2 MSS System Servi			2.7.100 is in the vault
SmartFusion2 MSS System Servic			2.7.100 is in the vault
SmartFusion2 MSS System Servic			2.7.100 is in the vault
SmartFusion2 MSS Timer Driver			
SmartFusion2 MSS Timer Driver	2.1.100	1.19	
SmartFusion2 MSS Timer Driver	2.0.101	0.93	
SmartFusion2 MSS USB Driver	2.5.100	26.48	

Documentation:
SmartFusion2_MSS_System_Services_Driver_RN.pdf
SmartFusion2_MSS_System_Services_Driver_UG.pdf

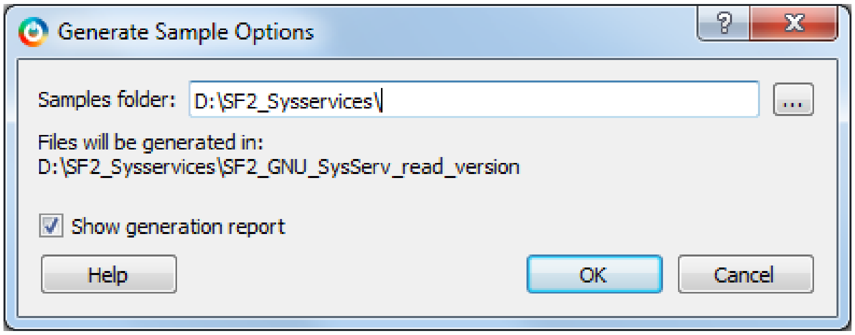
Description: SmartFusion2 microsontroller subsystem (MSS) System Services software driver.

This driver requires the SmartFusion2 CMSIS Hardware Abstraction Layer to build correctly.

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

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

Figure 3 • **Generate Sample Options Window**



Generate Sample Options

Samples folder: D:\SF2_Sysservices\

Files will be generated in:
D:\SF2_Sysservices\SF2_GNU_SysServ_read_version

☑ Show generation report

3. Browse to a location to save system services **Read Version Information** SoftConsole Project.
4. Open Readme file provided in the SF2_GNU_SysSer_read_version project folder. Readme file gives target hardware information.
5. SF2_GNU_SysSer_read_version project folder is also provided along with design files for reference.

2.5 Creating a Libero SoC Project

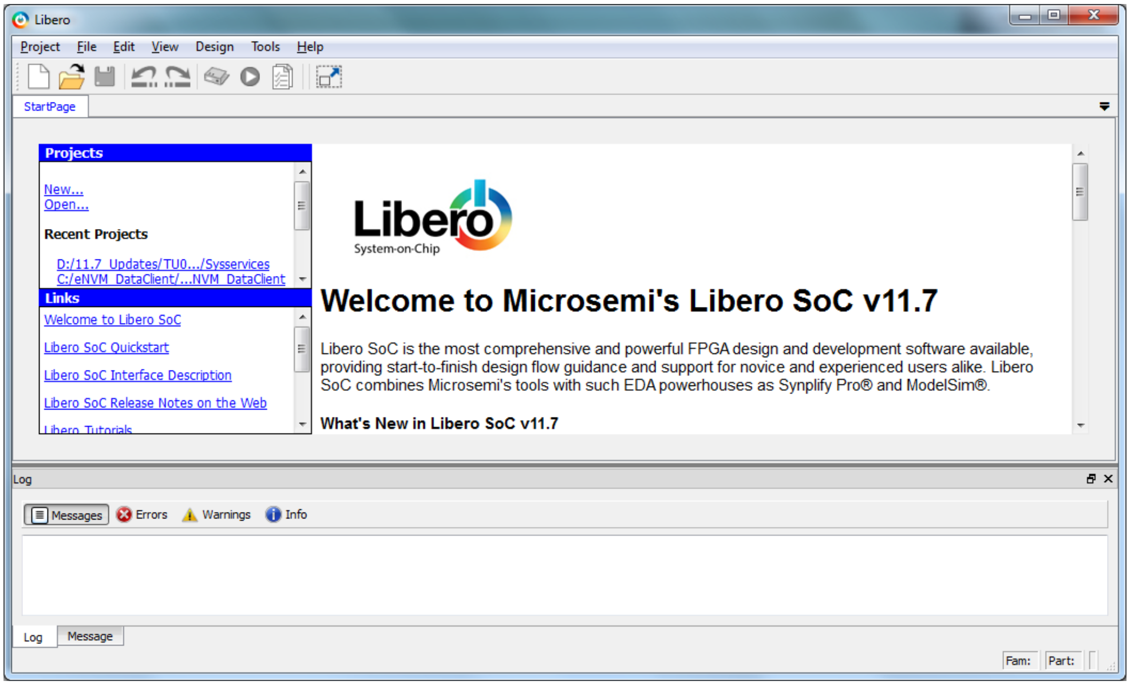
The following steps describe how to create SmartFusion2 firmware catalog design using the Libero SoC tool.

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**



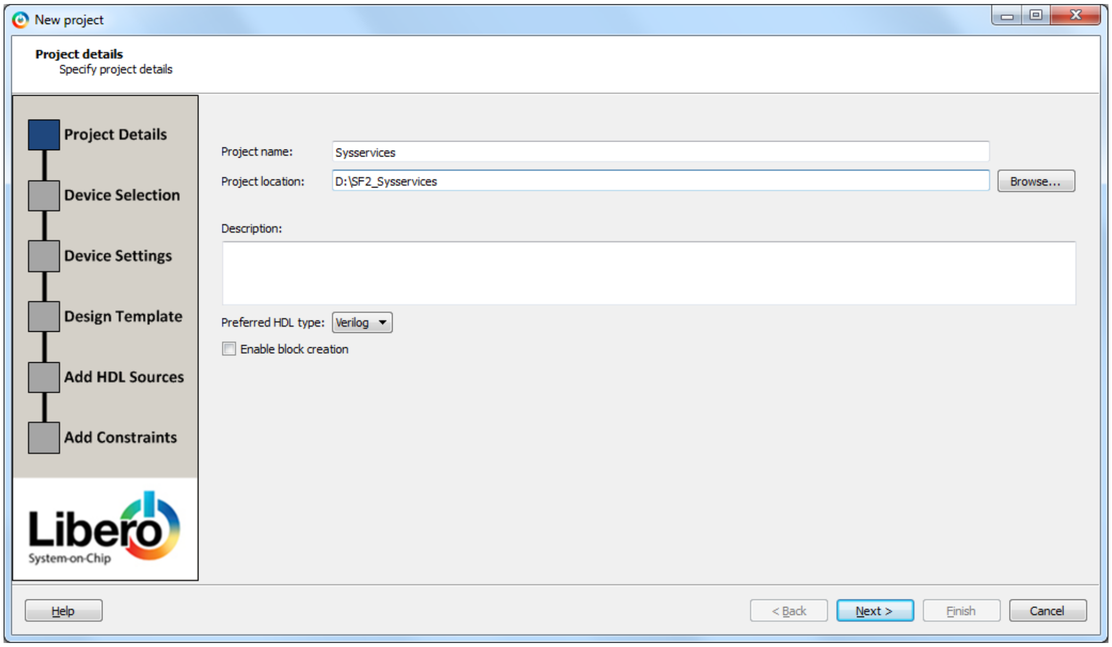
Welcome to Microsemi's Libero SoC v11.7

Libero SoC is the most comprehensive and powerful FPGA design and development software available, providing start-to-finish design flow guidance and support for novice and experienced users alike. Libero SoC combines Microsemi's tools with such EDA powerhouses as Synplify Pro® and ModelSim®.

What's New in Libero SoC v11.7

3. Enter the following information in the **Project Details** page, as shown in Figure 5:
   - **Project Name**: Sysservices
   - **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 name: Sysservices

Project location: D:\SF2_Sysservices

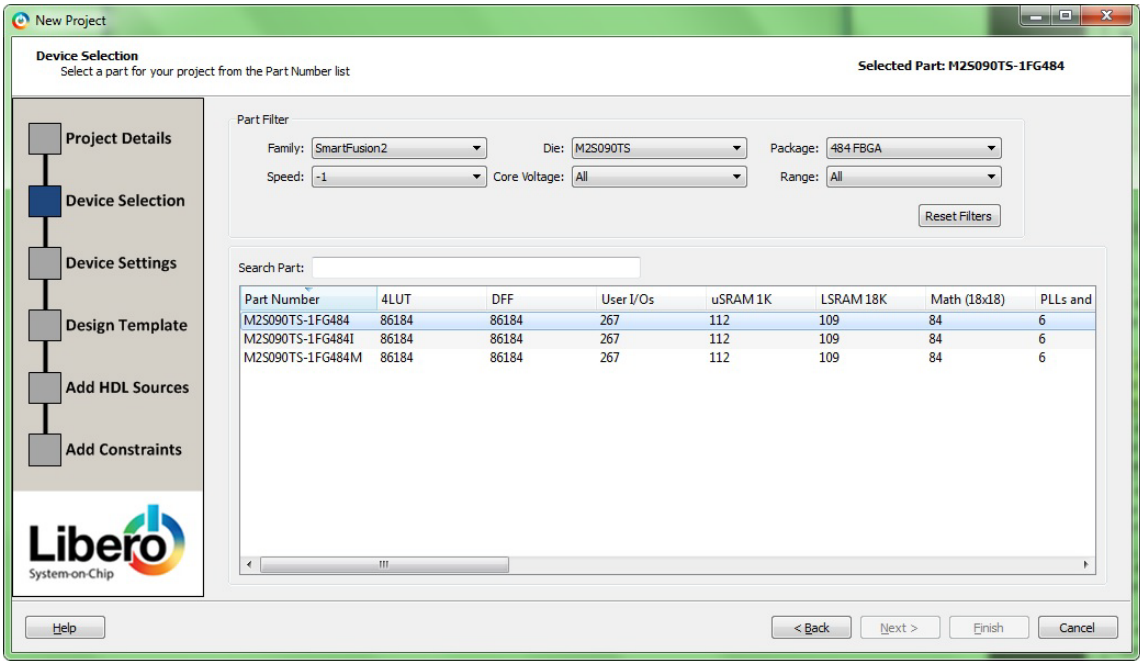
Description:

Preferred HDL type: Verilog

□ Enable block creation

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

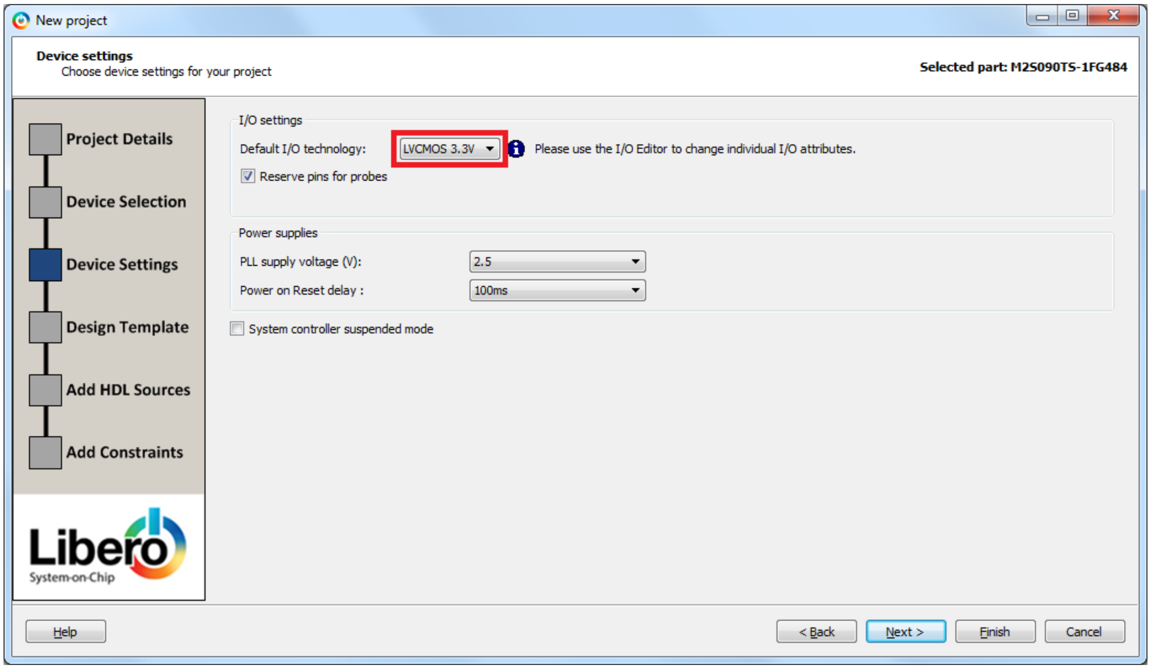
Family: SmartFusion2
Die: M2S090TS
Package: 484 FBGA
Speed: -1
Core Voltage: All
Range: All

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

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: M2S090TS-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

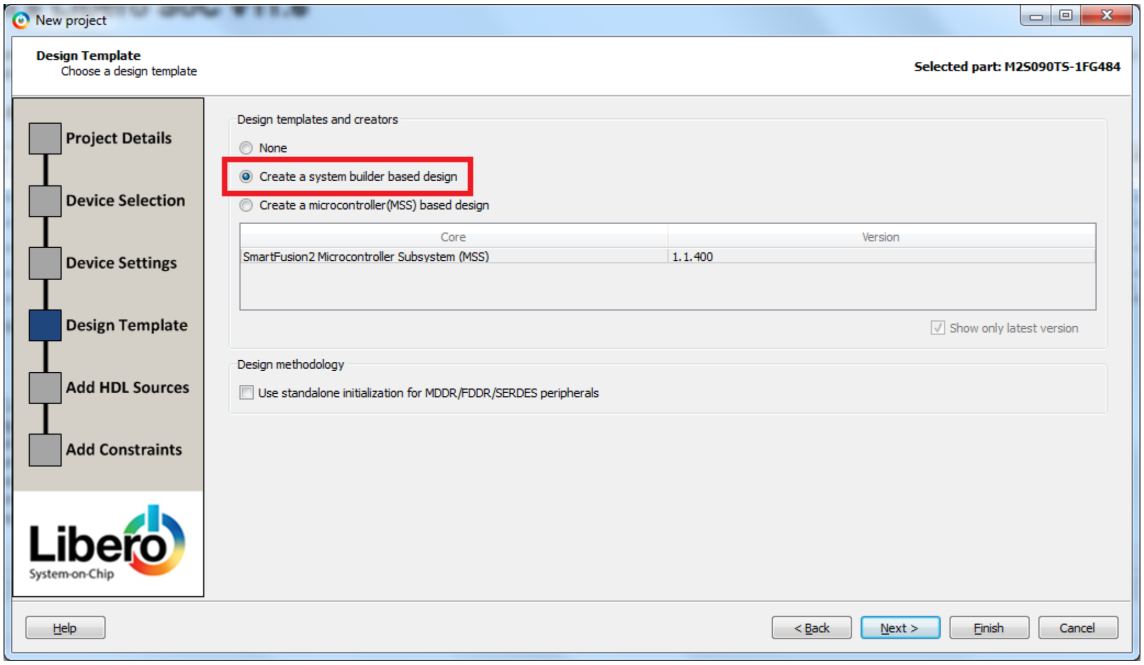
PLL supply voltage (V): 2.5

Power on Reset delay : 100ms

☐ System controller suspended mode

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: M2S090TS-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

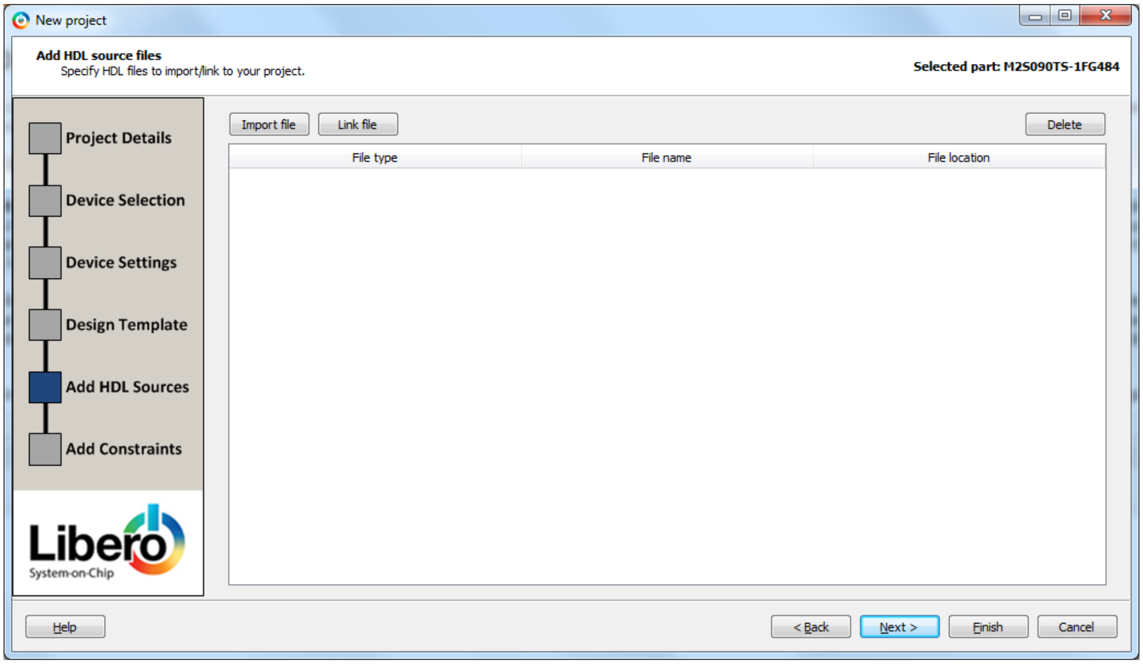
☑ Show only latest version

Design methodology

☐ Use standalone initialization for MDDR/FDDR/SERDES peripherals

7. Click **Next**. This opens **Add HDL Sources** page, as shown in Figure 9.

Figure 9 • **Add HDL Sources Page**



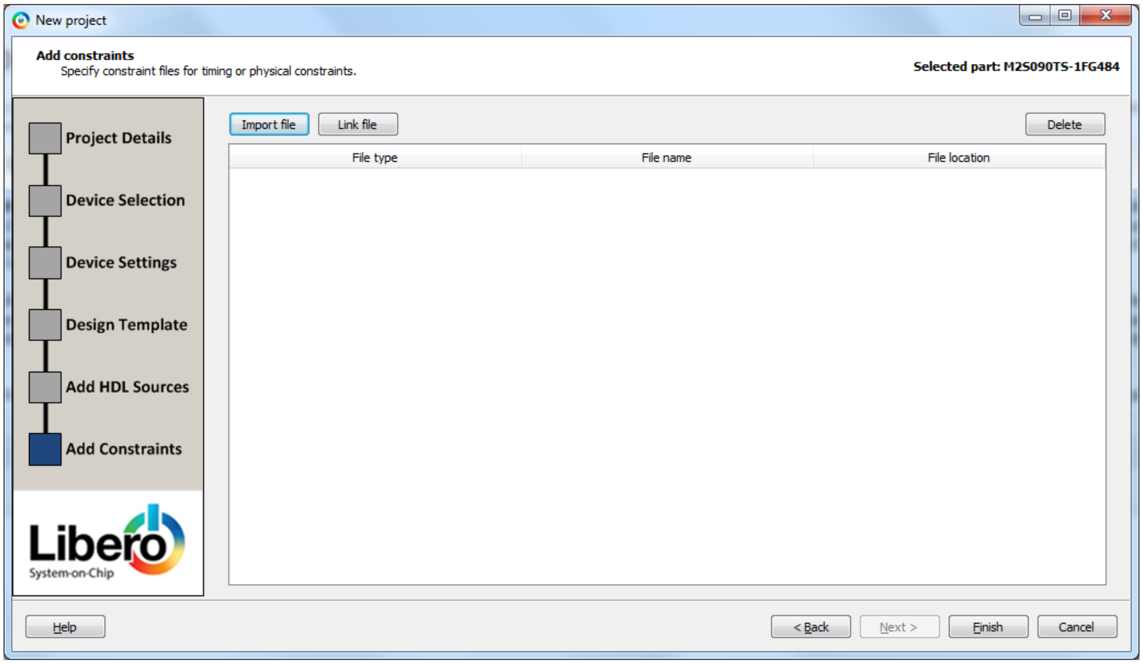
New project

Add HDL source files
Specify HDL files to import/link to your project.

Selected part: M2S090TS-1FG484

8. Click **Next**. This opens **Add Constraints** page, as shown in Figure 10.

Figure 10 • **Add Constraints Page**



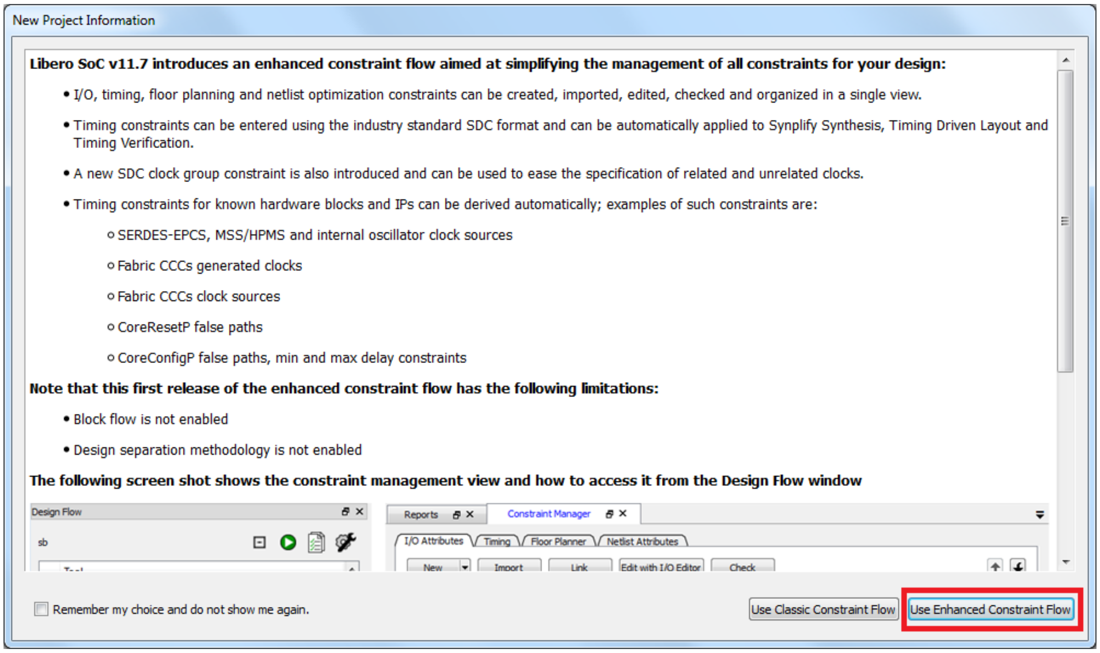
New project

Add constraints
Specify constraint files for timing or physical constraints.

Selected part: M2S090TS-1FG484

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**



Libero SoC v11.7 introduces an enhanced constraint flow aimed at simplifying the management of all constraints for your design:

- I/O, timing, floor planning and netlist optimization constraints can be created, imported, edited, checked and organized in a single view.
- Timing constraints can be entered using the industry standard SDC format and can be automatically applied to Synplify Synthesis, Timing Driven Layout and Timing Verification.
- A new SDC clock group constraint is also introduced and can be used to ease the specification of related and unrelated clocks.
- Timing constraints for known hardware blocks and IPs can be derived automatically; examples of such constraints are:
  - SERDES-EPCS, MSS/HPMS and internal oscillator clock sources
  - Fabric CCCs generated clocks
  - Fabric CCCs clock sources
  - CoreResetP false paths
  - CoreConfigP false paths, min and max delay constraints

Note that this first release of the enhanced constraint flow has the following limitations:

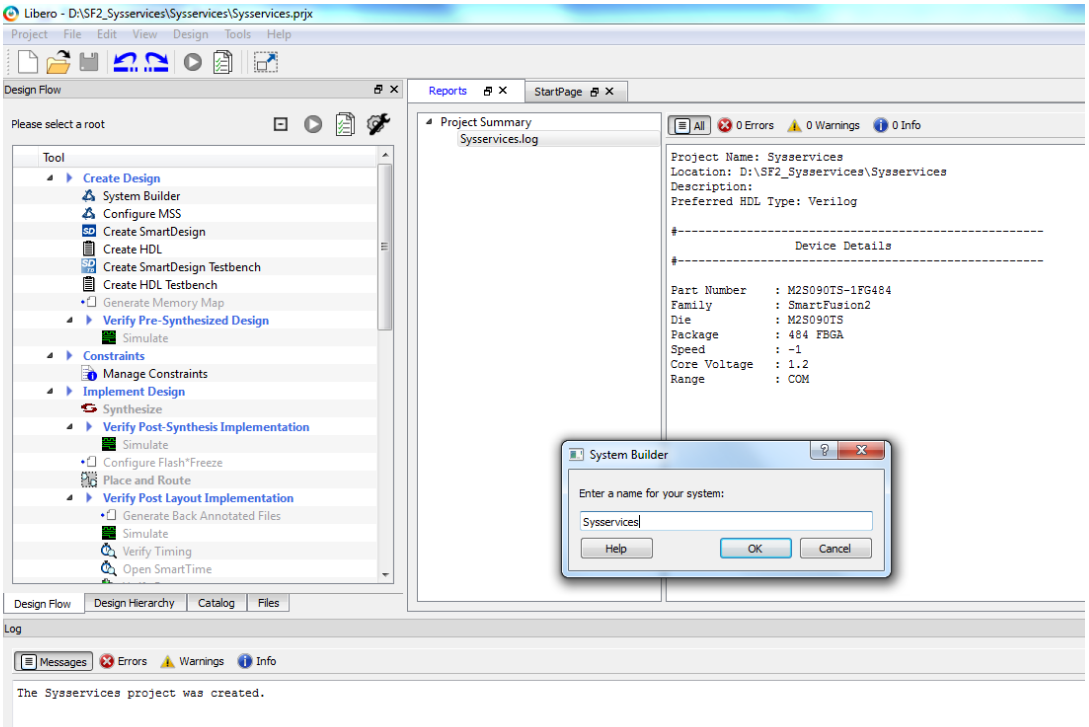
- Block flow is not enabled
- Design separation methodology is not enabled

The following screen shot shows the constraint management view and how to access it from the Design Flow window

□ Remember my choice and do not show me again.

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**



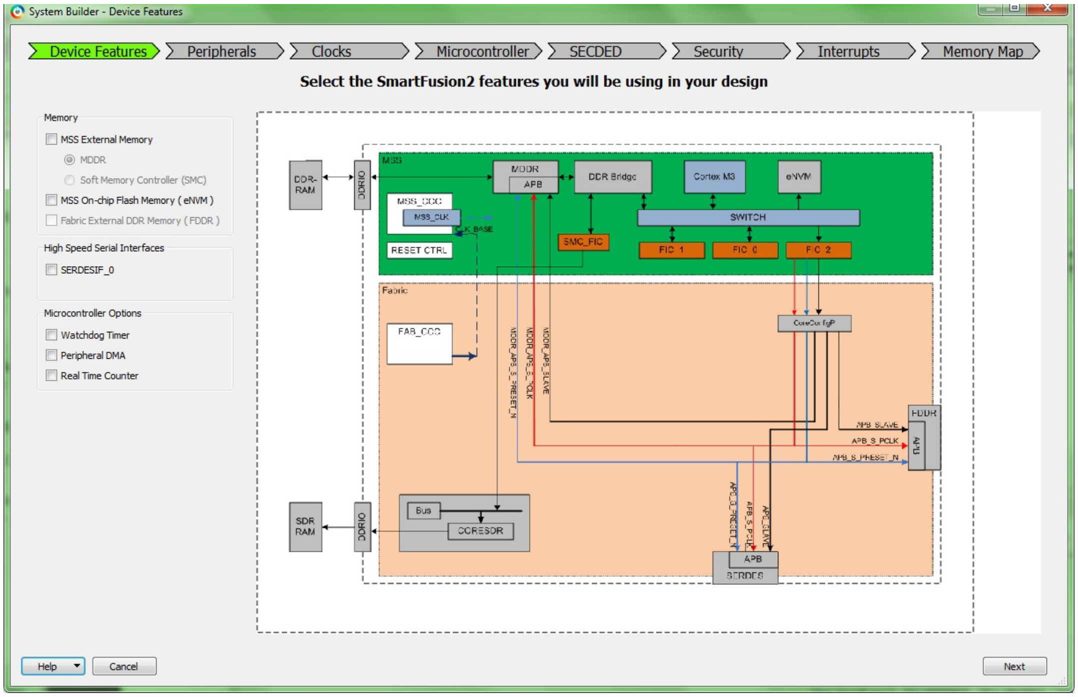
```
Project Name: Sysservices
Location: D:\SF2_Sysservices\Sysservices
Description:
Preferred HDL Type: Verilog

#-----------------------------------------------------
                    Device Details
#-----------------------------------------------------

Part Number    : M2S090TS-1FG484
Family         : SmartFusion2
Die            : M2S090TS
Package        : 484 FBGA
Speed          : -1
Core Voltage   : 1.2
Range          : COM
```

Enter a name for your system: Sysservices

```
The Sysservices project was created.
```

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

Figure 13 • **System Builders - Device Features Page**



Select the SmartFusion2 features you will be using in your design

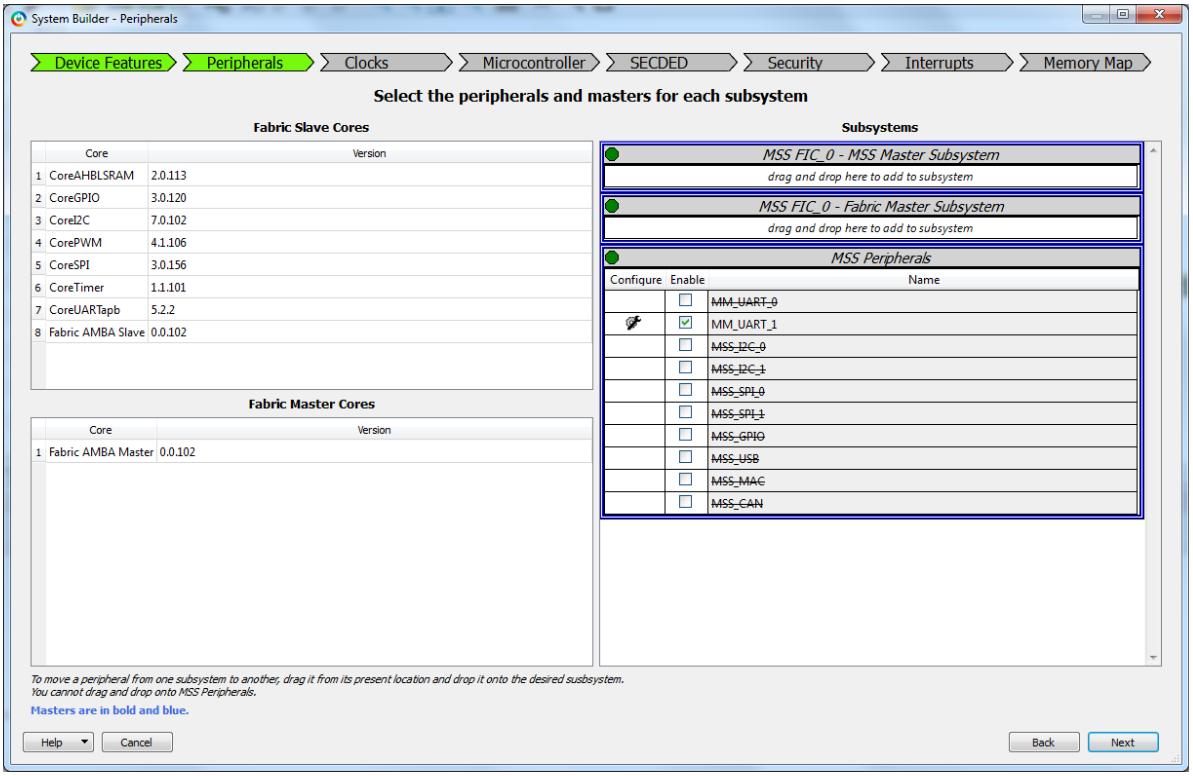
Memory
- □ MSS External Memory
  - ● MDDR
  - ○ Soft Memory Controller (SMC)
- □ MSS On-chip Flash Memory (eNVM)
- □ Fabric External DDR Memory (FDDR)

High Speed Serial Interfaces
- □ SERDESIF_0

Microcontroller Options
- □ Watchdog Timer
- □ Peripheral DMA
- □ Real Time Counter

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**



System Builder - Peripherals

Select the peripherals and masters for each subsystem

Fabric Slave Cores

	Core	Version
1	CoreAHBLSRAM	2.0.113
2	CoreGPIO	3.0.120
3	CoreI2C	7.0.102
4	CorePWM	4.1.106
5	CoreSPI	3.0.156
6	CoreTimer	1.1.101
7	CoreUARTapb	5.2.2
8	Fabric AMBA Slave	0.0.102

Fabric Master Cores

	Core	Version
1	Fabric AMBA Master	0.0.102

Subsystems

MSS FIC_0 - MSS Master Subsystem

drag and drop here to add to subsystem

MSS FIC_0 - Fabric Master Subsystem

drag and drop here to add to subsystem

MSS Peripherals

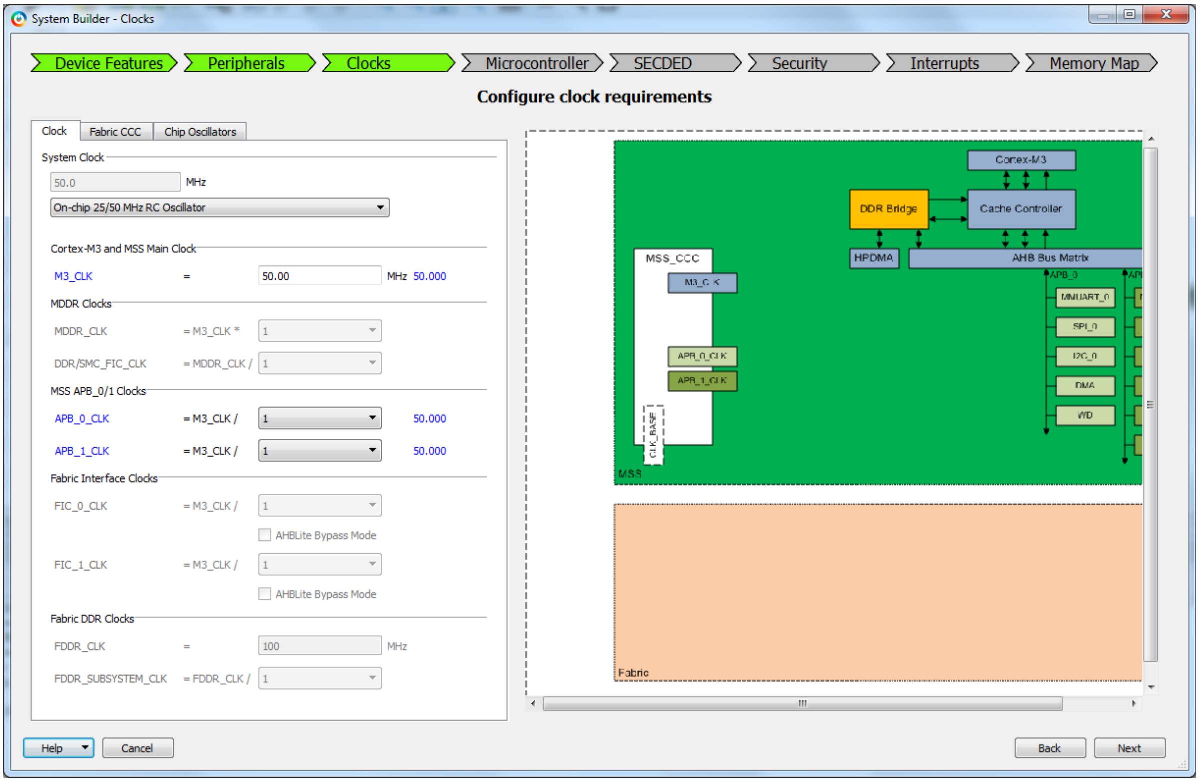
Configure	Enable	Name
	□	~~MM_UART_0~~
	☑	MM_UART_1
	□	~~MSS_I2C_0~~
	□	~~MSS_I2C_1~~
	□	~~MSS_SPI_0~~
	□	~~MSS_SPI_1~~
	□	~~MSS_GPIO~~
	□	~~MSS_USB~~
	□	~~MSS_MAC~~
	□	~~MSS_CAN~~

To move a peripheral from one subsystem to another, drag it from its present location and drop it onto the desired susbsystem.
You cannot drag and drop onto MSS Peripherals.

Masters are in bold and blue.

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**



System Builder - Clocks

Device Features > Peripherals > Clocks > Microcontroller > SECDED > Security > Interrupts > Memory Map

Configure clock requirements

Clock | Fabric CCC | Chip Oscillators

System Clock
50.0 MHz
On-chip 25/50 MHz RC Oscillator

Cortex-M3 and MSS Main Clock
M3_CLK = 50.00 MHz 50.000

MDDR Clocks
MDDR_CLK = M3_CLK * 1
DDR/SMC_FIC_CLK = MDDR_CLK / 1

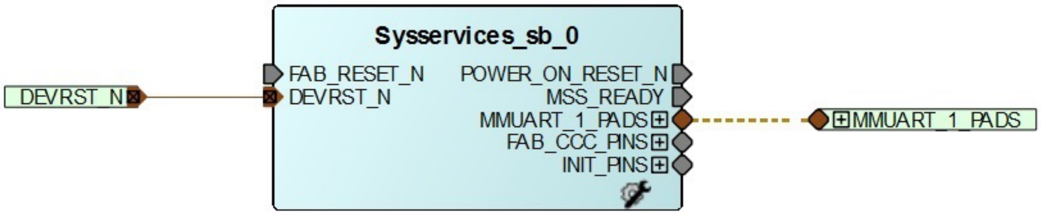
MSS APB_0/1 Clocks
APB_0_CLK = M3_CLK / 1 50.000
APB_1_CLK = M3_CLK / 1 50.000

Fabric Interface Clocks
FIC_0_CLK = M3_CLK / 1
□ AHBLite Bypass Mode
FIC_1_CLK = M3_CLK / 1
□ AHBLite Bypass Mode

Fabric DDR Clocks
FDDR_CLK = 100 MHz
FDDR_SUBSYSTEM_CLK = FDDR_CLK / 1

16. Click **Next**. This opens **System Builder - Microcontroller** page. Do not change the default selections.
17. Click **Next**. This opens **System Builder - SECDED** 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**



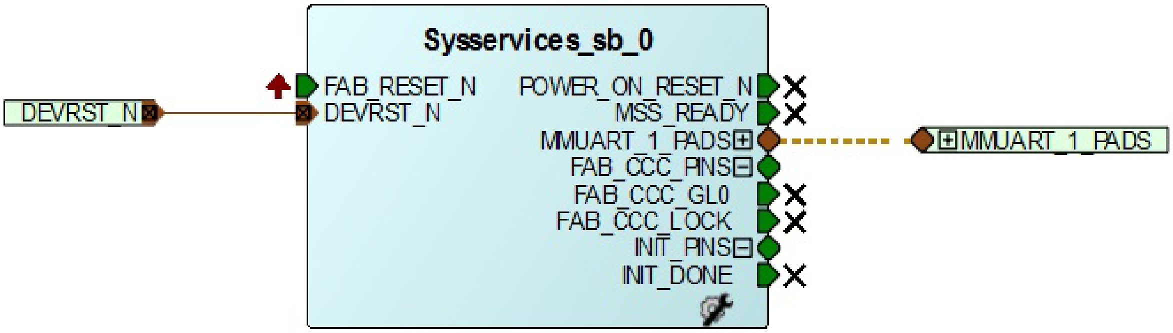
2.5.2 Connecting Components in Sysservices_sb_0 SmartDesign

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 I**NIT_DONE** and select **Mark Unused**.
4. Expand **FAB_CCC_PINS**, right-click **FAB_CCC_GL0** and select **Mark Unused**.
5. Right-click **FAB_CCC_LOCK** and select **Mark Unused**.
6. Right-click **FAB_RESET_N** and select **Tie High**.
7. Click **File** > **Save**.

The Sysservices_sb_0 design is displayed, as shown Figure 17.

Figure 17 • **Sysservices_sb_0**

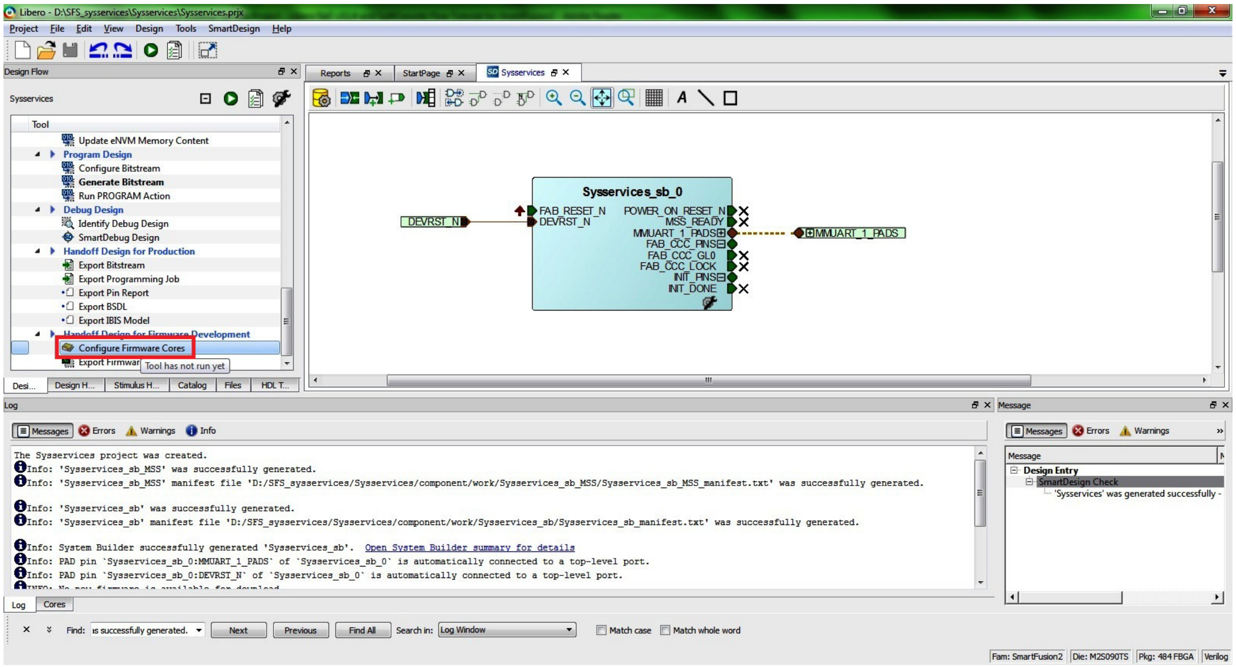


2.5.3 Configuring and Generating Firmware

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 DESGIN_FIRMWARE Window**

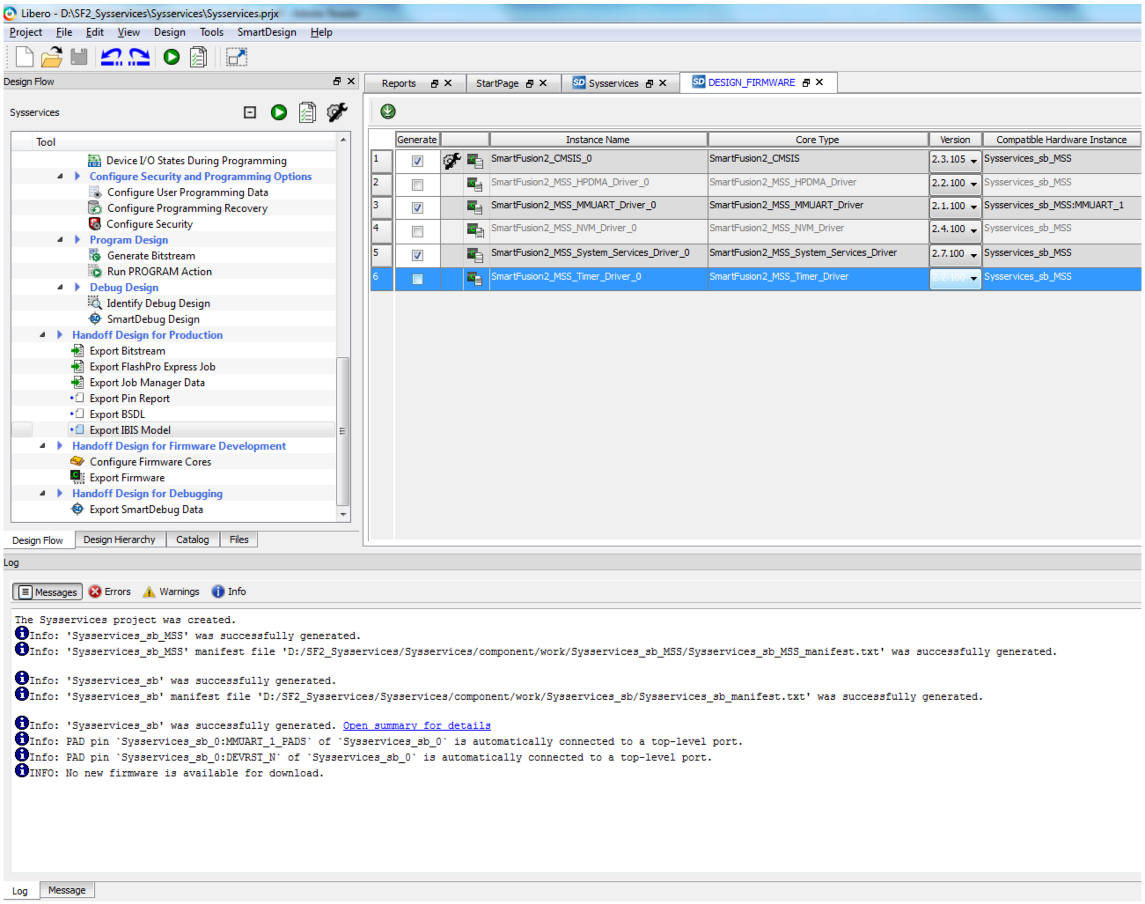

Libero - D:\SFS_sysservices\Sysservices\Sysservices.prjx
Project File Edit View Design Tools SmartDesign Help
Design Flow
Sysservices
Tool
Update eNVM Memory Content
Program Design
Configure Bitstream
Generate Bitstream
Run PROGRAM Action
Debug Design
Identify Debug Design
SmartDebug Design
Handoff Design for Production
Export Bitstream
Export Programming Job
Export Pin Report
Export BSDL
Export IBIS Model
Handoff Design for Firmware Development
Configure Firmware Cores
Export Firmwar
Tool has not run yet
Desi... Design H... Stimulus H... Catalog Files HDL T...
Reports StartPage Sysservices
Sysservices_sb_0
DEVRST_N
FAB_RESET_N
DEVRST_N
POWER_ON_RESET_N
MSS_READY
MMUART_1_PADS
FAB_CCC_PINS
FAB_CCC_GL0
FAB_CCC_LOCK
INIT_PINS
INIT_DONE
MMUART_1_PADS
Log
Messages Errors Warnings Info
Log Cores
Find: is successfully generated. Next Previous Find All Search in: Log Window
□ Match case □ Match whole word
Message
Messages Errors Warnings
Message
Design Entry
SmartDesign Check
'Sysservices' was generated successfully -
Fam: SmartFusion2 Die: M2S090TS Pkg: 484 FBGA Verilog

```
The Sysservices project was created.
Info: 'Sysservices_sb_MSS' was successfully generated.
Info: 'Sysservices_sb_MSS' manifest file 'D:/SFS_sysservices/Sysservices/component/work/Sysservices_sb_MSS/Sysservices_sb_MSS_manifest.txt' was successfully generated.

Info: 'Sysservices_sb' was successfully generated.
Info: 'Sysservices_sb' manifest file 'D:/SFS_sysservices/Sysservices/component/work/Sysservices_sb/Sysservices_sb_manifest.txt' was successfully generated.

Info: System Builder successfully generated 'Sysservices_sb'.  Open System Builder summary for details
Info: PAD pin `Sysservices_sb_0:MMUART_1_PADS` of `Sysservices_sb_0` is automatically connected to a top-level port.
Info: PAD pin `Sysservices_sb_0:DEVRST_N` of `Sysservices_sb_0` is automatically connected to a top-level port.
INFO: No new firmware is available for download
```

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**.

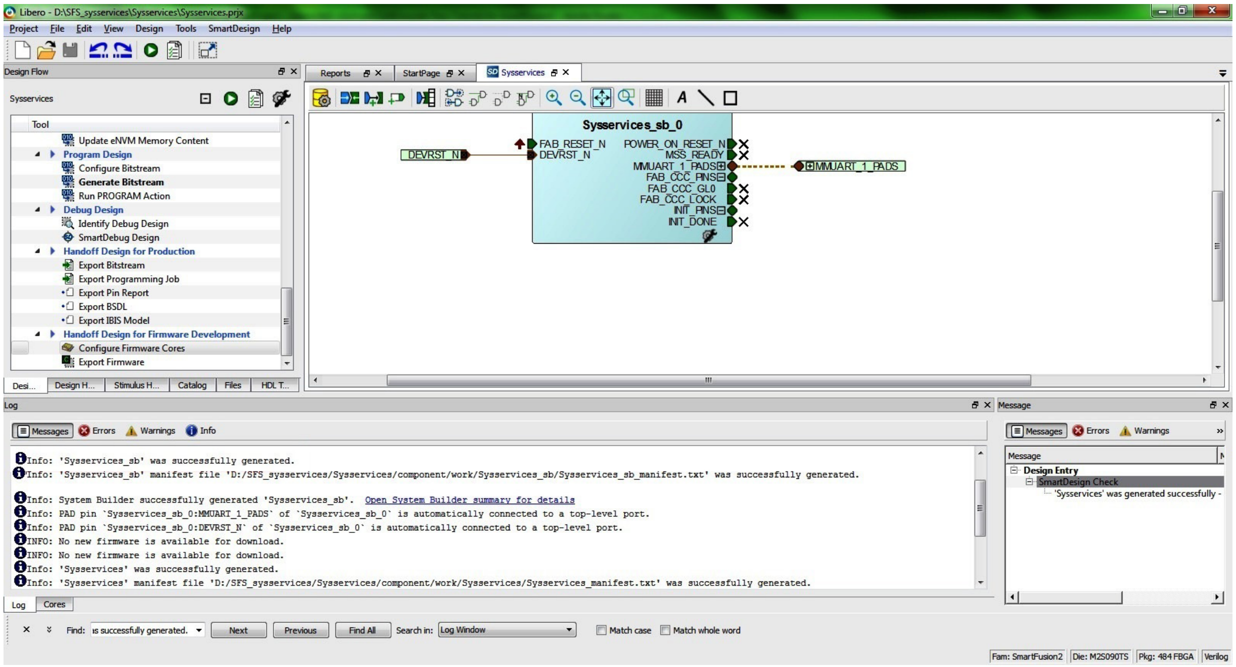
Figure 19 • **DESIGN_FIRMWARE Window**



	Generate		Instance Name	Core Type	Version	Compatible Hardware Instance
1	☑		SmartFusion2_CMSIS_0	SmartFusion2_CMSIS	2.3.105	Sysservices_sb_MSS
2	☐		SmartFusion2_MSS_HPDMA_Driver_0	SmartFusion2_MSS_HPDMA_Driver	2.2.100	Sysservices_sb_MSS
3	☑		SmartFusion2_MSS_MMUART_Driver_0	SmartFusion2_MSS_MMUART_Driver	2.1.100	Sysservices_sb_MSS:MMUART_1
4	☐		SmartFusion2_MSS_NVM_Driver_0	SmartFusion2_MSS_NVM_Driver	2.4.100	Sysservices_sb_MSS
5	☑		SmartFusion2_MSS_System_Services_Driver_0	SmartFusion2_MSS_System_Services_Driver	2.7.100	Sysservices_sb_MSS
6	☐		SmartFusion2_MSS_Timer_Driver_0	SmartFusion2_MSS_Timer_Driver	2.2.100	Sysservices_sb_MSS

```
The Sysservices project was created.
Info: 'Sysservices_sb_MSS' was successfully generated.
Info: 'Sysservices_sb_MSS' manifest file 'D:/SF2_Sysservices/Sysservices/component/work/Sysservices_sb_MSS/Sysservices_sb_MSS_manifest.txt' was successfully generated.

Info: 'Sysservices_sb' was successfully generated.
Info: 'Sysservices_sb' manifest file 'D:/SF2_Sysservices/Sysservices/component/work/Sysservices_sb/Sysservices_sb_manifest.txt' was successfully generated.

Info: 'Sysservices_sb' was successfully generated. Open summary for details
Info: PAD pin `Sysservices_sb_0:MMUART_1_PADS` of `Sysservices_sb_0` is automatically connected to a top-level port.
Info: PAD pin `Sysservices_sb_0:DEVRST_N` of `Sysservices_sb_0` is automatically connected to a top-level port.
INFO: No new firmware is available for download.
```

Figure 20 • **Generate Component**



```
Info: 'Sysservices_sb' was successfully generated.
Info: 'Sysservices_sb' manifest file 'D:/SFS_sysservices/Sysservices/component/work/Sysservices_sb/Sysservices_sb_manifest.txt' was successfully generated.

Info: System Builder successfully generated 'Sysservices_sb'.  Open System Builder summary for details
Info: PAD pin `Sysservices_sb_0:MMUART_1_PADS` of `Sysservices_sb_0` is automatically connected to a top-level port.
Info: PAD pin `Sysservices_sb_0:DEVRST_N` of `Sysservices_sb_0` is automatically connected to a top-level port.
INFO: No new firmware is available for download.
INFO: No new firmware is available for download.
Info: 'Sysservices' was successfully generated.
Info: 'Sysservices' manifest file 'D:/SFS_sysservices/Sysservices/component/work/Sysservices/Sysservices_manifest.txt' was successfully generated.
```

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

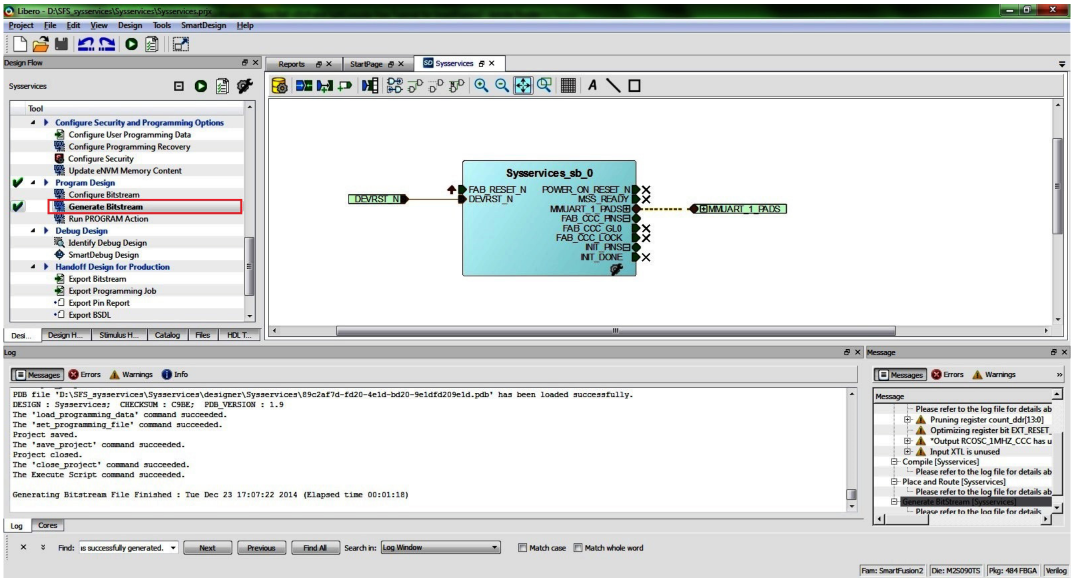
```
Info: 'Sysservices' was successfully generated.
```

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**

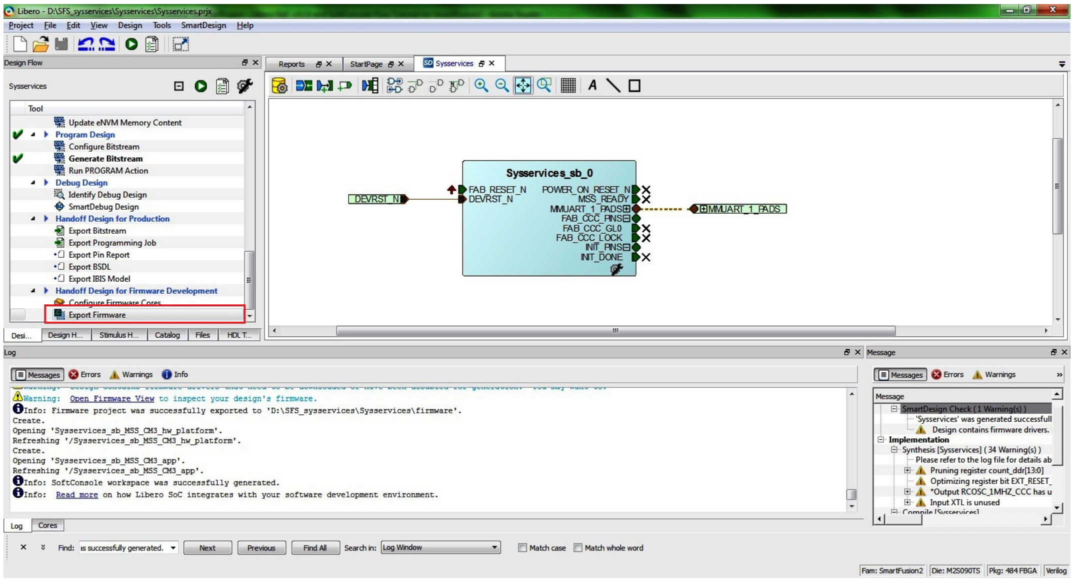


```
PDB file 'D:\SFS_sysservices\Sysservices\designer\Sysservices\89c2af7d-fd20-4e1d-bd20-9e1dfd209e1d.pdb' has been loaded successfully.
DESIGN : Sysservices;  CHECKSUM : C9BE;  PDB_VERSION : 1.9
The 'load_programming_data' command succeeded.
The 'set_programming_file' command succeeded.
Project saved.
The 'save_project' command succeeded.
Project closed.
The 'close_project' command succeeded.
The Execute Script command succeeded.

Generating Bitstream File Finished : Tue Dec 23 17:07:22 2014 (Elapsed time 00:01:18)
```

Libero - D:\SFS_sysservices\Sysservices\Sysservices.prjx
Project File Edit View Design Tools SmartDesign Help
Design Flow
Sysservices
Tool
Configure Security and Programming Options
Configure User Programming Data
Configure Programming Recovery
Configure Security
Update eNVM Memory Content
Program Design
Configure Bitstream
Generate Bitstream
Run PROGRAM Action
Debug Design
Identify Debug Design
SmartDebug Design
Handoff Design for Production
Export Bitstream
Export Programming Job
Export Pin Report
Export BSDL
Reports
StartPage
Sysservices
Sysservices_sb_0
DEVRST_N
FAB_RESET_N
DEVRST_N
POWER_ON_RESET_N
MSS_READY
MMUART_1_PADS
FAB_CCC_PINS
FAB_CCC_GL0
FAB_CCC_LOCK
INIT_PINS
INIT_DONE
MMUART_1_PADS
Log
Messages
Errors
Warnings
Info
Log
Cores
Find: is successfully generated.
Next
Previous
Find All
Search in: Log Window
Match case
Match whole word
Message
Please refer to the log file for details ab
Pruning register count_ddr[13:0]
Optimizing register bit EXT_RESET_
"Output RCOSC_1MHZ_CCC has u
Input XTL is unused
Compile [Sysservices]
Please refer to the log file for details ab
Place and Route [Sysservices]
Please refer to the log file for details ab
Generate Bitstream [Sysservices]
Please refer to the log file for details
Fam: SmartFusion2
Die: M2S090TS
Pkg: 484 FBGA
Verilog

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

Figure 22 • **Export Firmware**

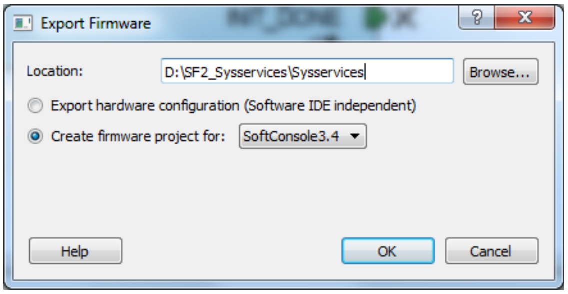

Libero - D:\SFS_sysservices\Sysservices\Sysservices.prjx
Project File Edit View Design Tools SmartDesign Help
Design Flow
Sysservices
Tool
Update eNVM Memory Content
Program Design
Configure Bitstream
Generate Bitstream
Run PROGRAM Action
Debug Design
Identify Debug Design
SmartDebug Design
Handoff Design for Production
Export Bitstream
Export Programming Job
Export Pin Report
Export BSDL
Export IBIS Model
Handoff Design for Firmware Development
Configure Firmware Cores
Export Firmware
Desi... Design H... Stimulus H... Catalog Files HDL T...
Reports StartPage Sysservices
Sysservices_sb_0
DEVRST_N
FAB_RESET_N
DEVRST_N
POWER_ON_RESET_N
MSS_READY
MMUART_1_PADS
FAB_CCC_PINS
FAB_CCC_GL0
FAB_CCC_LOCK
INIT_PINS
INIT_DONE
MMUART_1_PADS
Log
Messages Errors Warnings Info
Log Cores
Find: is successfully generated. Next Previous Find All Search in: Log Window Match case Match whole word
Message
Messages Errors Warnings
Fam: SmartFusion2 Die: M2S090TS Pkg: 484 FBGA Verilog

```
Warning:  Open Firmware View to inspect your design's firmware.
Info: Firmware project was successfully exported to 'D:\SFS_sysservices\Sysservices\firmware'.
Create.
Opening 'Sysservices_sb_MSS_CM3_hw_platform'.
Refreshing '/Sysservices_sb_MSS_CM3_hw_platform'.
Create.
Opening 'Sysservices_sb_MSS_CM3_app'.
Refreshing '/Sysservices_sb_MSS_CM3_app'.
Info: SoftConsole workspace was successfully generated.
Info:  Read more on how Libero SoC integrates with your software development environment.
```

```
Message
SmartDesign Check ( 1 Warning(s) )
    'Sysservices' was generated successfull
    Design contains firmware drivers.
Implementation
  Synthesis [Sysservices] ( 34 Warning(s) )
    Please refer to the log file for details ab
    Pruning register count_ddr[13:0]
    Optimizing register bit EXT_RESET_
    *Output RCOSC_1MHZ_CCC has u
    Input XTL is unused
  Compile [Sysservices]
```

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**



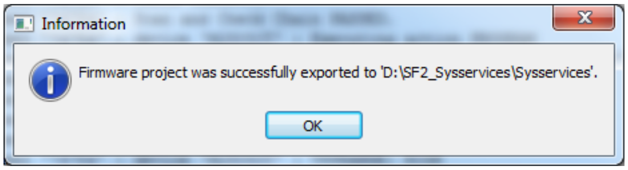
Location: D:\SF2_Sysservices\Sysservices

○ Export hardware configuration (Software IDE independent)

● Create firmware project for: SoftConsole3.4

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

Figure 24 • **Firmware Successfully Exported Message**

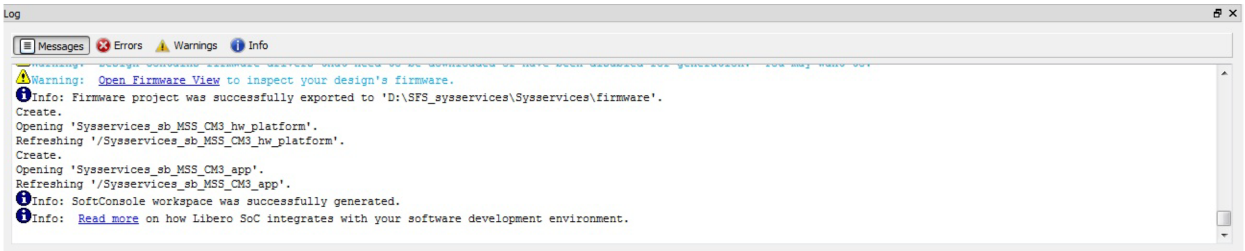


Firmware project was successfully exported to 'D:\SF2_Sysservices\Sysservices'.

5. Click **OK**.

The log window is displayed, as shown in Figure 25.

Figure 25 • **Firmware Log Window**



```
Warning:  Design contains firmware drivers that need to be downloaded or have been disabled for generation.  You may want to
Warning:  Open Firmware View to inspect your design's firmware.
Info: Firmware project was successfully exported to 'D:\SFS_sysservices\Sysservices\firmware'.
Create.
Opening 'Sysservices_sb_MSS_CM3_hw_platform'.
Refreshing '/Sysservices_sb_MSS_CM3_hw_platform'.
Create.
Opening 'Sysservices_sb_MSS_CM3_app'.
Refreshing '/Sysservices_sb_MSS_CM3_app'.
Info: SoftConsole workspace was successfully generated.
Info:  Read more on how Libero SoC integrates with your software development environment.
```

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**

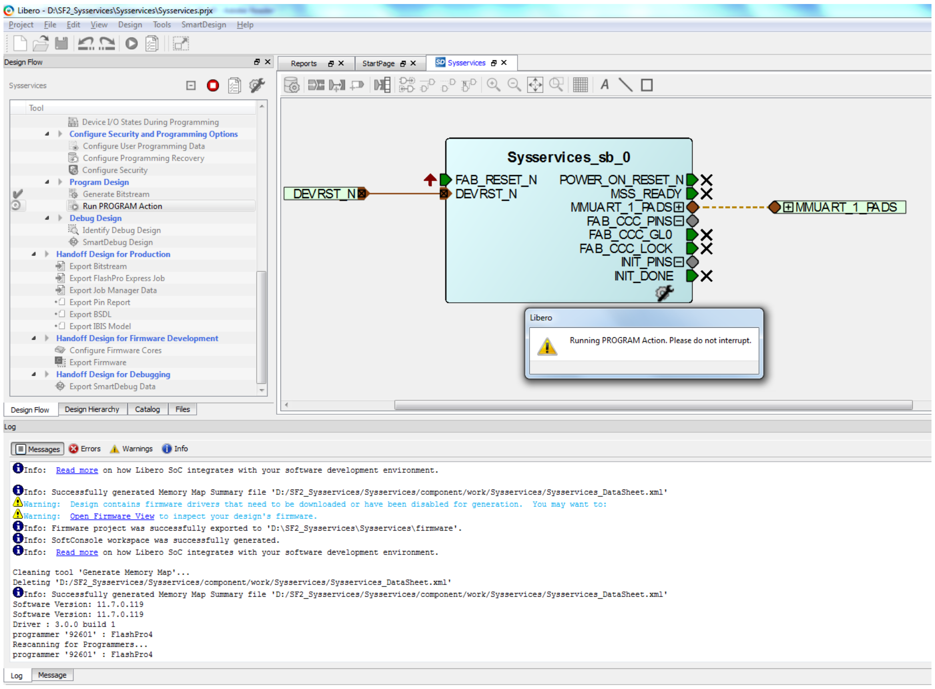
Jumper Number	Pin (From)	Pin (To)	Comments
J22, J23, J24, J8, J3	1	2	These are the default jumper settings of the SmartFusion2 Security Evaluation Kit board. Ensure that these jumpers are set accordingly.

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_df\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**



Running PROGRAM Action. Please do not interrupt.

```
Info:  Read more on how Libero SoC integrates with your software development environment.

Info: Successfully generated Memory Map Summary file 'D:/SF2_Sysservices/Sysservices/component/work/Sysservices/Sysservices_DataSheet.xml'
Warning:  Design contains firmware drivers that need to be downloaded or have been disabled for generation.  You may want to:
Warning:  Open Firmware View to inspect your design's firmware.
Info: Firmware project was successfully exported to 'D:\SF2_Sysservices\Sysservices\firmware'.
Info: SoftConsole workspace was successfully generated.
Info:  Read more on how Libero SoC integrates with your software development environment.

Cleaning tool 'Generate Memory Map'...
Deleting 'D:/SF2_Sysservices/Sysservices/component/work/Sysservices/Sysservices_DataSheet.xml'
Info: Successfully generated Memory Map Summary file 'D:/SF2_Sysservices/Sysservices/component/work/Sysservices/Sysservices_DataSheet.xml'
Software Version: 11.7.0.119
Software Version: 11.7.0.119
Driver : 3.0.0 build 1
programmer '92601' : FlashPro4
Rescanning for Programmers...
programmer '92601' : FlashPro4
```

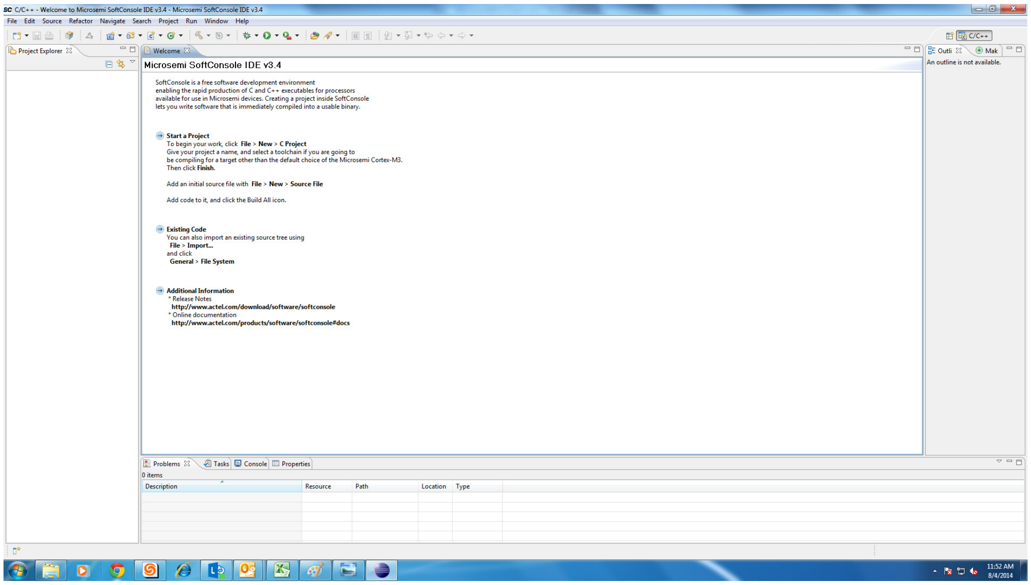
7. After successful programming, press **SW6** switch to reset the board.

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**


C/C++ - Welcome to Microsemi SoftConsole IDE v3.4 - Microsemi SoftConsole IDE v3.4
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
Welcome
C/C++
Outli
Mak
An outline is not available.
Problems
Tasks
Console
Properties
0 items
Description
Resource
Path
Location
Type
11:52 AM
8/4/2014

Microsemi SoftConsole IDE v3.4

SoftConsole is a free software development environment enabling the rapid production of C and C++ executables for processors available for use in Microsemi devices. Creating a project inside SoftConsole lets you write software that is immediately compiled into a usable binary.

Start a Project
To begin your work, click **File** > **New** > **C Project**
Give your project a name, and select a toolchain if you are going to be compiling for a target other than the default choice of the Microsemi Cortex-M3.
Then click **Finish.**

Add an initial source file with **File** > **New** > **Source File**

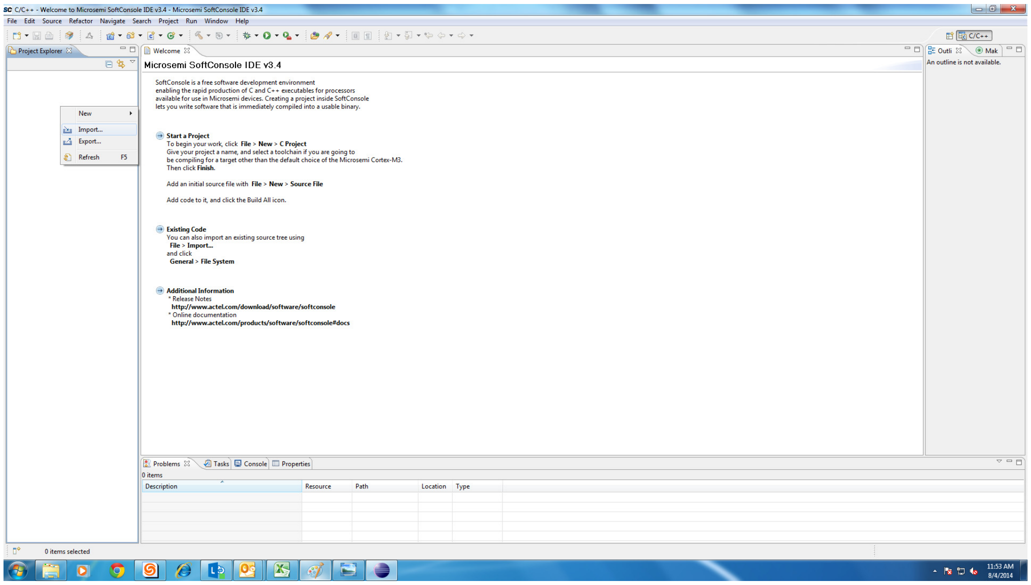
Add code to it, and click the Build All icon.

Existing Code
You can also import an existing source tree using
File > **Import...**
and click
General > **File System**

Additional Information
* Release Notes
http://www.actel.com/download/software/softconsole
* Online documentation
http://www.actel.com/products/software/softconsole#docs

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

Figure 28 • **Importing Projects**


C/C++ - Welcome to Microsemi SoftConsole IDE v3.4 - Microsemi SoftConsole IDE v3.4
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
New
Import...
Export...
Refresh F5
Welcome
C/C++
Outli
Mak
An outline is not available.
Problems
Tasks
Console
Properties
0 items
Description
Resource
Path
Location
Type
0 items selected
11:53 AM
8/4/2014

Microsemi SoftConsole IDE v3.4

SoftConsole is a free software development environment enabling the rapid production of C and C++ executables for processors available for use in Microsemi devices. Creating a project inside SoftConsole lets you write software that is immediately compiled into a usable binary.

Start a Project
To begin your work, click **File** > **New** > **C Project**
Give your project a name, and select a toolchain if you are going to be compiling for a target other than the default choice of the Microsemi Cortex-M3.
Then click **Finish.**

Add an initial source file with **File** > **New** > **Source File**

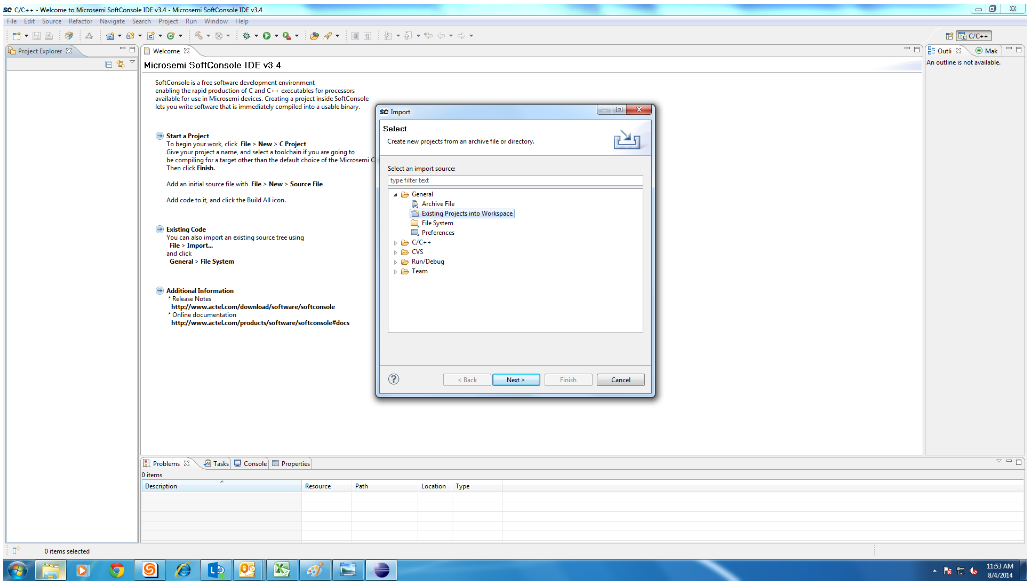
Add code to it, and click the Build All icon.

Existing Code
You can also import an existing source tree using
File > **Import...**
and click
General > **File System**

Additional Information
* Release Notes
http://www.actel.com/download/software/softconsole
* Online documentation
http://www.actel.com/products/software/softconsole#docs

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

Figure 29 • **Importing Existing Projects**



Microsemi SoftConsole IDE v3.4

SoftConsole is a free software development environment enabling the rapid production of C and C++ executables for processors available for use in Microsemi devices. Creating a project inside SoftConsole lets you write software that is immediately compiled into a usable binary.

Start a Project
To begin your work, click **File** > **New** > **C Project**
Give your project a name, and select a toolchain if you are going to be compiling for a target other than the default choice of the Microsemi C
Then click **Finish**.

Add an initial source file with **File** > **New** > **Source File**

Add code to it, and click the Build All icon.

Existing Code
You can also import an existing source tree using
File > **Import...**
and click
General > **File System**

Additional Information
* Release Notes
http://www.actel.com/download/software/softconsole
* Online documentation
http://www.actel.com/products/software/softconsole#docs

An outline is not available.

0 items

0 items selected

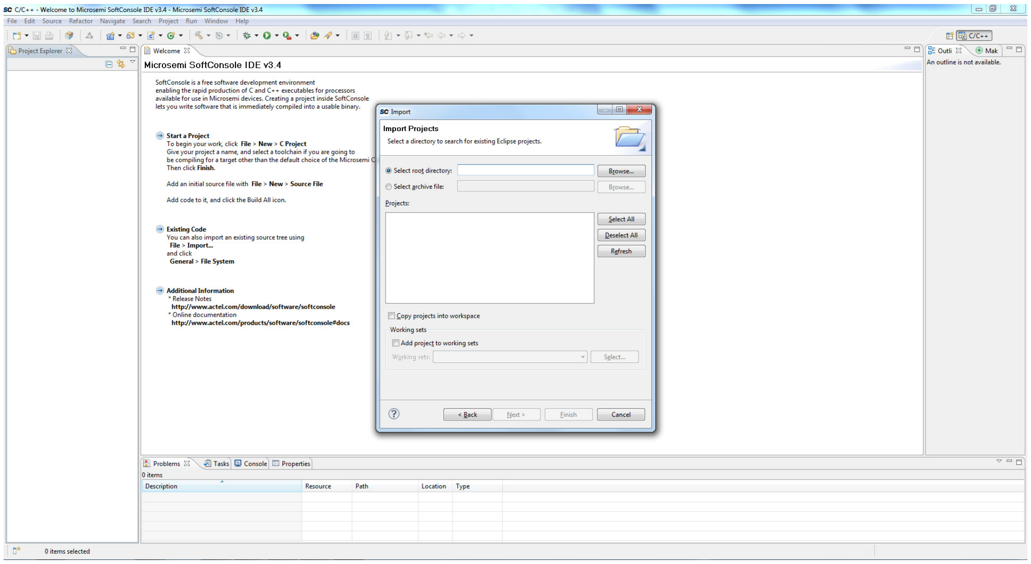
Select
Create new projects from an archive file or directory.

Select an import source:

type filter text

Import window is displayed, as shown in Figure 30.

Figure 30 • **Import Window**



Import Projects
Select a directory to search for existing Eclipse projects.

● Select root directory:

○ Select archive file:

Projects:

□ Copy projects into workspace

Working sets

□ Add project to working sets

Working sets:

Microsemi SoftConsole IDE v3.4

SoftConsole is a free software development environment enabling the rapid production of C and C++ executables for processors available for use in Microsemi devices. Creating a project inside SoftConsole lets you write software that is immediately compiled into a usable binary.

Start a Project
To begin your work, click **File** > **New** > **C Project**
Give your project a name, and select a toolchain if you are going to be compiling for a target other than the default choice of the Microsemi C
Then click **Finish**.

Add an initial source file with **File** > **New** > **Source File**

Add code to it, and click the Build All icon.

Existing Code
You can also import an existing source tree using
File > **Import...**
and click
General > **File System**

Additional Information
* Release Notes
http://www.actel.com/download/software/softconsole
* Online documentation
http://www.actel.com/products/software/softconsole#docs

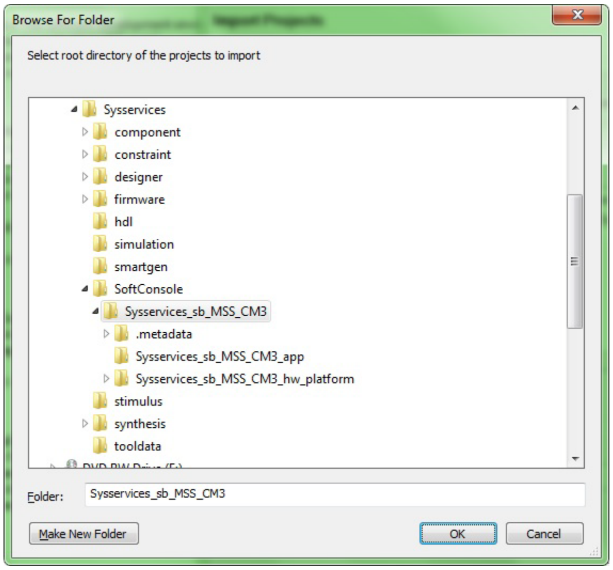
An outline is not available.

0 items

0 items selected

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

Figure 31 • **Selecting System services**



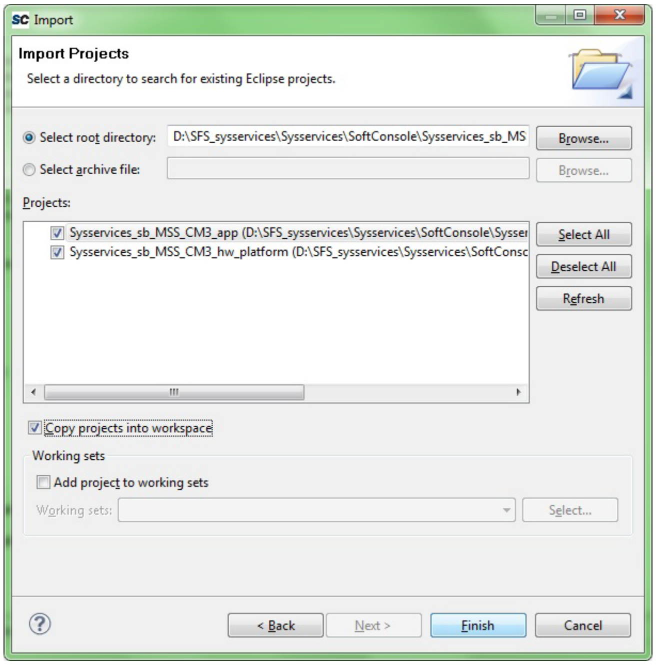
Browse For Folder

Select root directory of the projects to import

Folder: Sysservices_sb_MSS_CM3

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

Figure 32 • **Adding Projects to SoftConsole IDE**



Import

Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: D:\SFS_sysservices\Sysservices\SoftConsole\Sysservices_sb_MS

○ Select archive file:

Projects:

☑ Sysservices_sb_MSS_CM3_app (D:\SFS_sysservices\Sysservices\SoftConsole\Sysser

☑ Sysservices_sb_MSS_CM3_hw_platform (D:\SFS_sysservices\Sysservices\SoftConsc

☑ Copy projects into workspace

Working sets

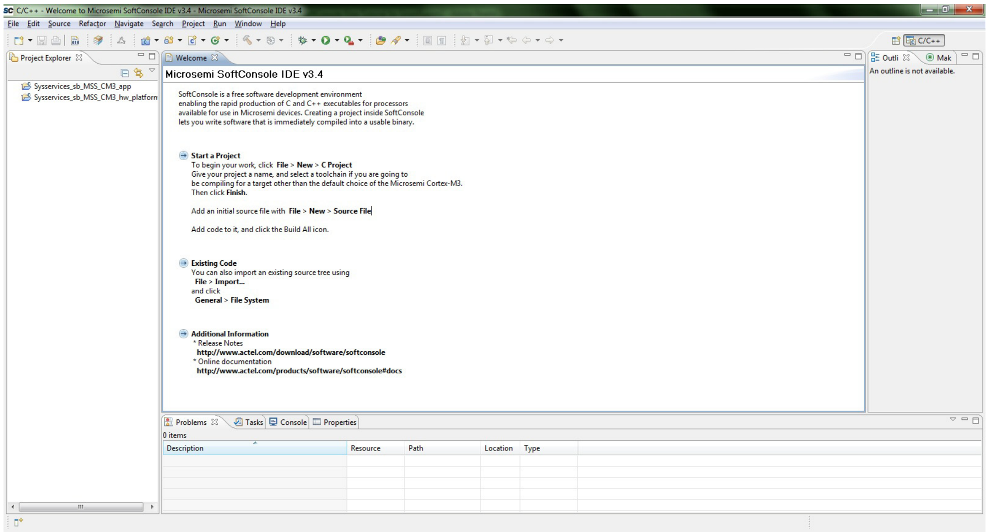
□ Add project to working sets

Working sets:

6. Click **Finish**.

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

Figure 33 • **SoftConsole Workspace**



C/C++ - Welcome to Microsemi SoftConsole IDE v3.4 - Microsemi SoftConsole IDE v3.4
File Edit Source Refactor Navigate Search Project Run Window Help
C/C++
Project Explorer
Sysservices_sb_MSS_CM3_app
Sysservices_sb_MSS_CM3_hw_platform
Welcome
Outli
Mak
Problems
Tasks
Console
Properties

An outline is not available.

Microsemi SoftConsole IDE v3.4

SoftConsole is a free software development environment
enabling the rapid production of C and C++ executables for processors
available for use in Microsemi devices. Creating a project inside SoftConsole
lets you write software that is immediately compiled into a usable binary.

Start a Project
To begin your work, click **File** > **New** > **C Project**
Give your project a name, and select a toolchain if you are going to
be compiling for a target other than the default choice of the Microsemi Cortex-M3.
Then click **Finish.**

Add an initial source file with **File** > **New** > **Source File**

Add code to it, and click the Build All icon.

Existing Code
You can also import an existing source tree using
File > **Import...**
and click
General > **File System**

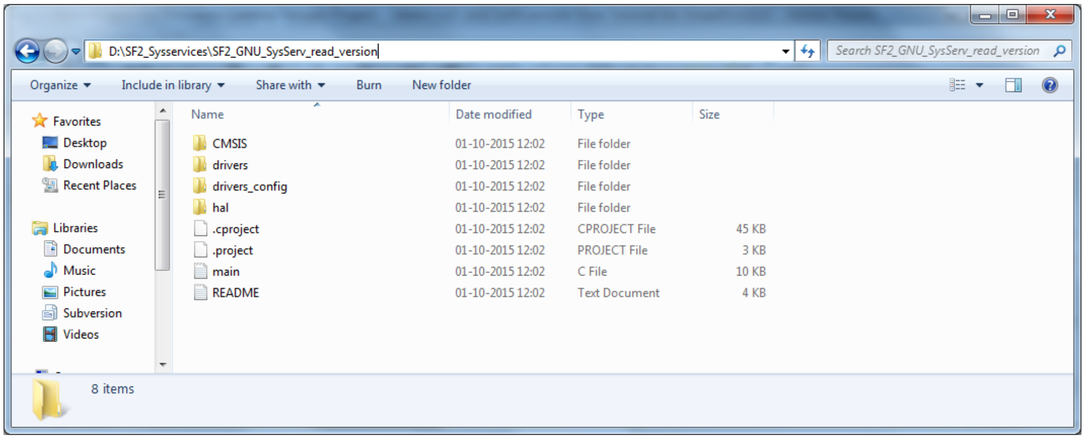
Additional Information
* Release Notes
http://www.actel.com/download/software/softconsole
* Online documentation
http://www.actel.com/products/software/softconsole#docs

0 items

Description	Resource	Path	Location	Type

7. 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**



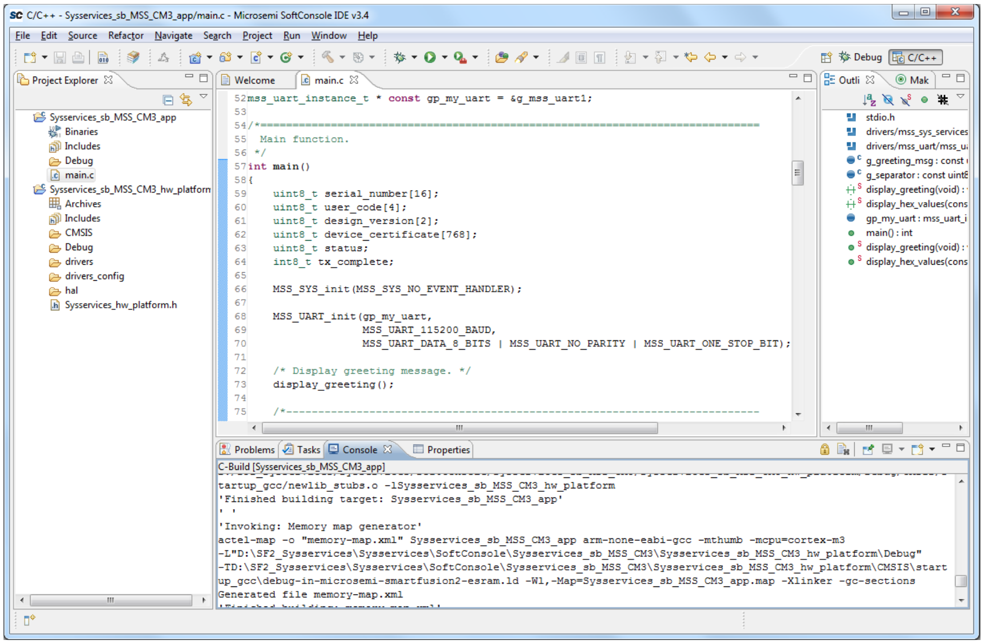
D:\SF2_Sysservices\SF2_GNU_SysServ_read_version

Name	Date modified	Type	Size
CMSIS	01-10-2015 12:02	File folder	
drivers	01-10-2015 12:02	File folder	
drivers_config	01-10-2015 12:02	File folder	
hal	01-10-2015 12:02	File folder	
.cproject	01-10-2015 12:02	CPROJECT File	45 KB
.project	01-10-2015 12:02	PROJECT File	3 KB
main	01-10-2015 12:02	C File	10 KB
README	01-10-2015 12:02	Text Document	4 KB

8 items

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**

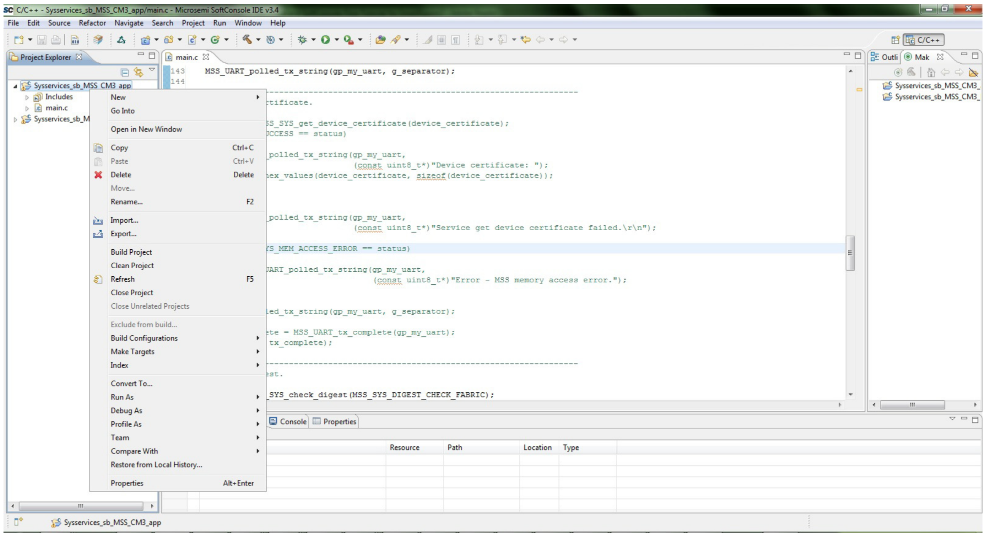


```
mss_uart_instance_t * const gp_my_uart = &g_mss_uart1;

/*==============================================================================
  Main function.
 */
int main()
{
    uint8_t serial_number[16];
    uint8_t user_code[4];
    uint8_t design_version[2];
    uint8_t device_certificate[768];
    uint8_t status;
    int8_t tx_complete;

    MSS_SYS_init(MSS_SYS_NO_EVENT_HANDLER);

    MSS_UART_init(gp_my_uart,
                  MSS_UART_115200_BAUD,
                  MSS_UART_DATA_8_BITS | MSS_UART_NO_PARITY | MSS_UART_ONE_STOP_BIT);

    /* Display greeting message. */
    display_greeting();

    /*------------------------------------------------------------------------
```

```
C-Build [Sysservices_sb_MSS_CM3_app]
tartup_gcc/newlib_stubs.o -lSysservices_sb_MSS_CM3_hw_platform
'Finished building target: Sysservices_sb_MSS_CM3_app'
' '
'Invoking: Memory map generator'
actel-map -o "memory-map.xml" Sysservices_sb_MSS_CM3_app arm-none-eabi-gcc -mthumb -mcpu=cortex-m3
-L"D:\SF2_Sysservices\Sysservices\SoftConsole\Sysservices_sb_MSS_CM3\Sysservices_sb_MSS_CM3_hw_platform\Debug"
-TD:\SF2_Sysservices\Sysservices\SoftConsole\Sysservices_sb_MSS_CM3\Sysservices_sb_MSS_CM3_hw_platform\CMSIS\start
up_gcc\debug-in-microsemi-smartfusion2-esram.ld -Wl,-Map=Sysservices_sb_MSS_CM3_app.map -Xlinker -gc-sections
Generated file memory-map.xml
```

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

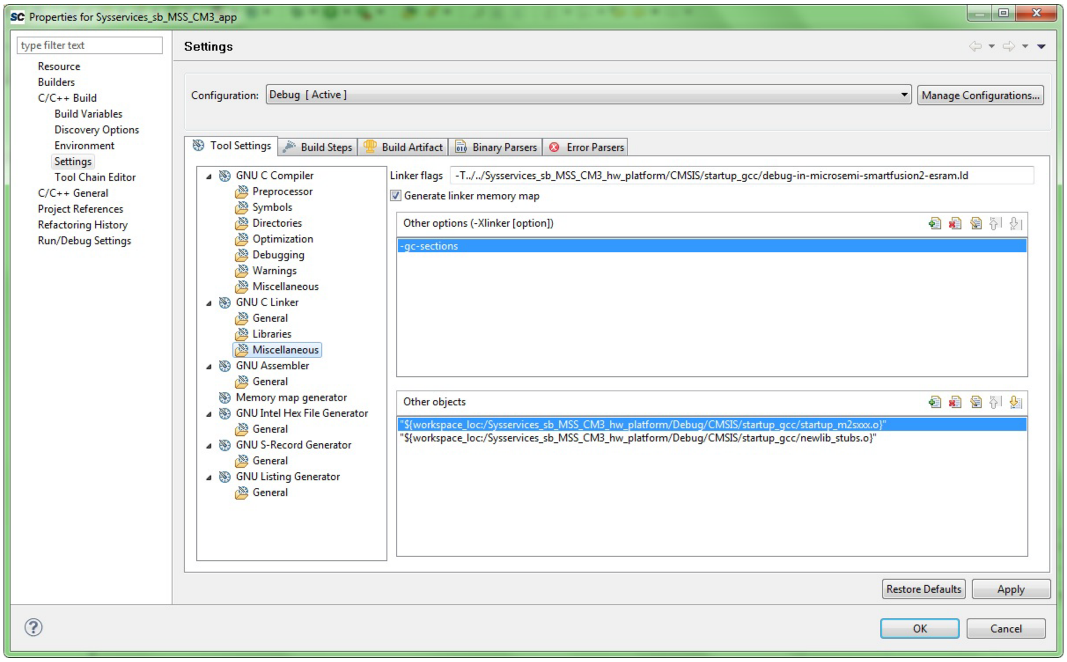
Figure 36 • **Project Explorer Window - SoftConsole Project**



```
    MSS_UART_polled_tx_string(gp_my_uart, g_separator);

-------------------------------------------------------------------
tificate.

S_SYS_get_device_certificate(device_certificate);
CCESS == status)

_polled_tx_string(gp_my_uart,
                 (const uint8_t*)"Device certificate: ");
ex_values(device_certificate, sizeof(device_certificate));


_polled_tx_string(gp_my_uart,
                 (const uint8_t*)"Service get device certificate failed.\r\n");

S_MEM_ACCESS_ERROR == status)

ART_polled_tx_string(gp_my_uart,
                    (const uint8_t*)"Error - MSS memory access error.");

ed_tx_string(gp_my_uart, g_separator);

te = MSS_UART_tx_complete(gp_my_uart);
 tx_complete);

-------------------------------------------------------------------
st.

_SYS_check_digest(MSS_SYS_DIGEST_CHECK_FABRIC);
```

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 • **Sysservices_sb_MSS_CM3 Properties Window**



Configuration: Debug [Active]

Linker flags: -T../../Sysservices_sb_MSS_CM3_hw_platform/CMSIS/startup_gcc/debug-in-microsemi-smartfusion2-esram.ld

☑ Generate linker memory map

Other options (-Xlinker [option]):

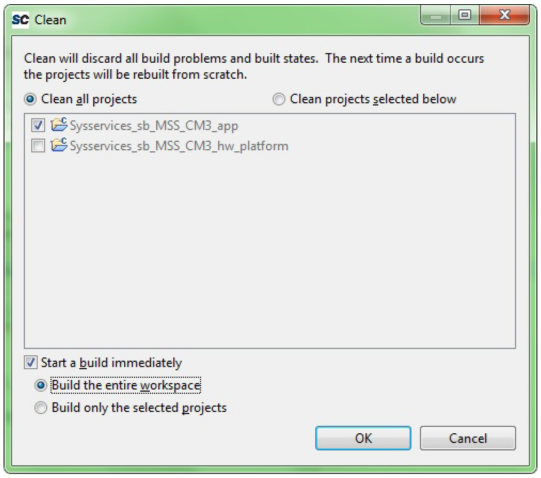
```
-gc-sections
```

Other objects:

```
"${workspace_loc:/Sysservices_sb_MSS_CM3_hw_platform/Debug/CMSIS/startup_gcc/startup_m2sxxx.o}"
"${workspace_loc:/Sysservices_sb_MSS_CM3_hw_platform/Debug/CMSIS/startup_gcc/newlib_stubs.o}"
```

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**



SC Clean

Clean will discard all build problems and built states. The next time a build occurs the projects will be rebuilt from scratch.

● Clean all projects ○ Clean projects selected below

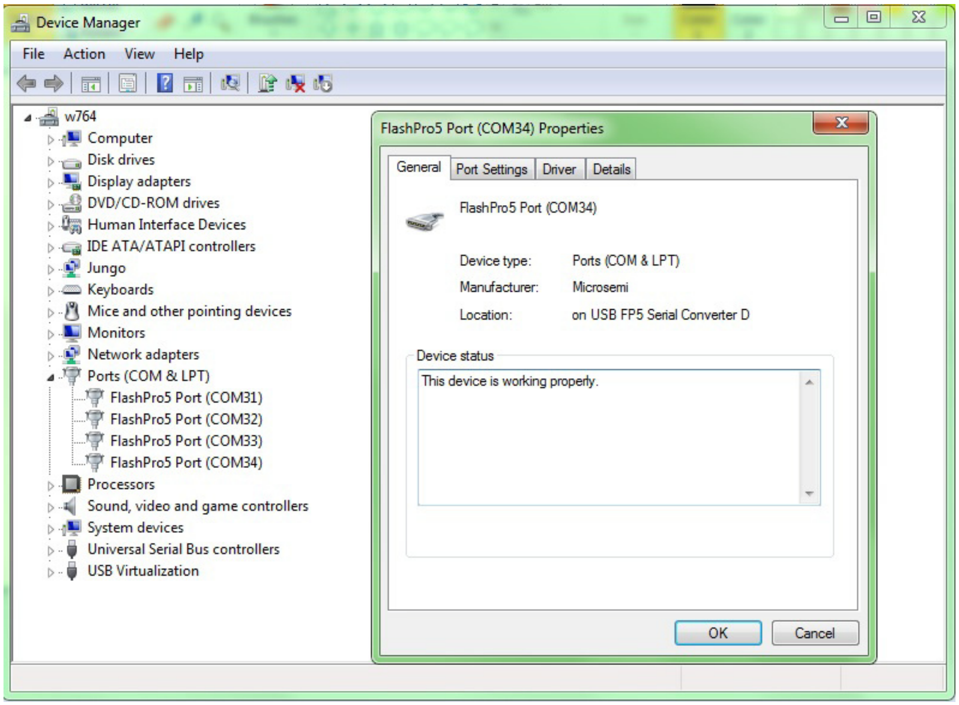
☑ Sysservices_sb_MSS_CM3_app
□ Sysservices_sb_MSS_CM3_hw_platform

☑ Start a build immediately
● Build the entire workspace
○ Build only the selected projects

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**



Device Manager

- w764
  - Computer
  - Disk drives
  - Display adapters
  - DVD/CD-ROM drives
  - Human Interface Devices
  - IDE ATA/ATAPI controllers
  - Jungo
  - Keyboards
  - Mice and other pointing devices
  - Monitors
  - Network adapters
  - Ports (COM & LPT)
    - FlashPro5 Port (COM31)
    - FlashPro5 Port (COM32)
    - FlashPro5 Port (COM33)
    - FlashPro5 Port (COM34)
  - Processors
  - Sound, video and game controllers
  - System devices
  - Universal Serial Bus controllers
  - USB Virtualization

FlashPro5 Port (COM34) Properties

FlashPro5 Port (COM34)

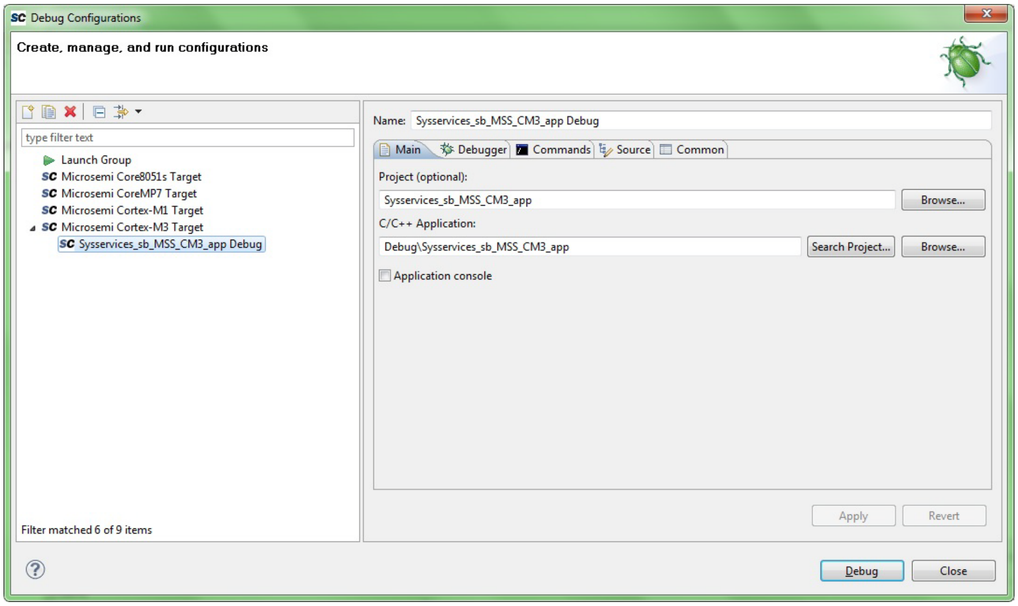
Device type: Ports (COM & LPT)
Manufacturer: Microsemi
Location: on USB FP5 Serial Converter D

Device status
This device is working properly.

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**



SC Debug Configurations
Create, manage, and run configurations
type filter text
Launch Group
SC Microsemi Core8051s Target
SC Microsemi CoreMP7 Target
SC Microsemi Cortex-M1 Target
SC Microsemi Cortex-M3 Target
SC Sysservices_sb_MSS_CM3_app Debug
Main
Debugger
Commands
Source
Common
Browse...
Search Project...
Browse...
Apply
Revert
Debug
Close

Name: Sysservices_sb_MSS_CM3_app Debug

Project (optional): Sysservices_sb_MSS_CM3_app

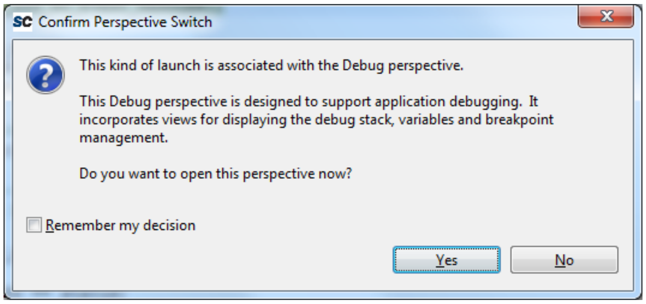
C/C++ Application: Debug\Sysservices_sb_MSS_CM3_app

□ Application console

Filter matched 6 of 9 items

16. Ensure that the following information appears on the **Main** tab in the **Debug Configurations** window, and click **Debug**:
    - **Name**: Sysservices_sb_MSS_ CM3 Debug
    - **Project**: Sysservices_sb_MSS_ CM3_app
    - **C/C++ Application**: Debug\Sysservices_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**



This kind of launch is associated with the Debug perspective.

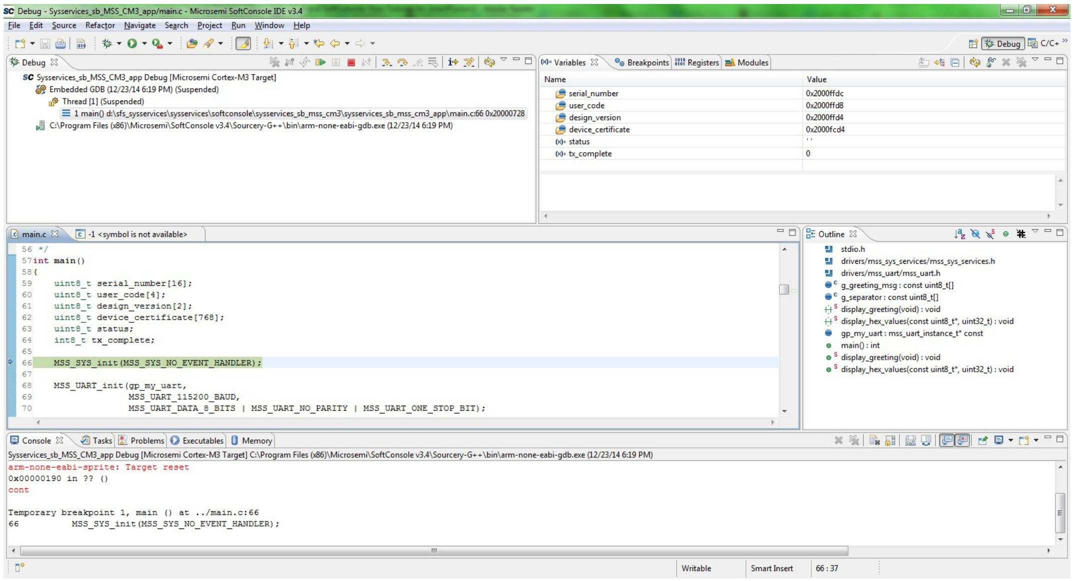
This Debug perspective is designed to support application debugging. It incorporates views for displaying the debug stack, variables and breakpoint management.

Do you want to open this perspective now?

□ Remember my decision

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

Figure 42 • **SoftConsole Debugger Perspective**



Sysservices_sb_MSS_CM3_app Debug [Microsemi Cortex-M3 Target]
Embedded GDB (12/23/14 6:19 PM) (Suspended)
Thread [1] (Suspended)
1 main() d:\sfs_sysservices\sysservices\softconsole\sysservices_sb_mss_cm3\sysservices_sb_mss_cm3_app\main.c:66 0x20000728
C:\Program Files (x86)\Microsemi\SoftConsole v3.4\Sourcery-G++\bin\arm-none-eabi-gdb.exe (12/23/14 6:19 PM)

Name	Value
serial_number	0x2000ffdc
user_code	0x2000ffd8
design_version	0x2000ffd4
device_certificate	0x2000fcd4
status	' '
tx_complete	0

```
*/
int main()
{
    uint8_t serial_number[16];
    uint8_t user_code[4];
    uint8_t design_version[2];
    uint8_t device_certificate[768];
    uint8_t status;
    int8_t tx_complete;

    MSS_SYS_init(MSS_SYS_NO_EVENT_HANDLER);

    MSS_UART_init(gp_my_uart,
                  MSS_UART_115200_BAUD,
                  MSS_UART_DATA_8_BITS | MSS_UART_NO_PARITY | MSS_UART_ONE_STOP_BIT);
```

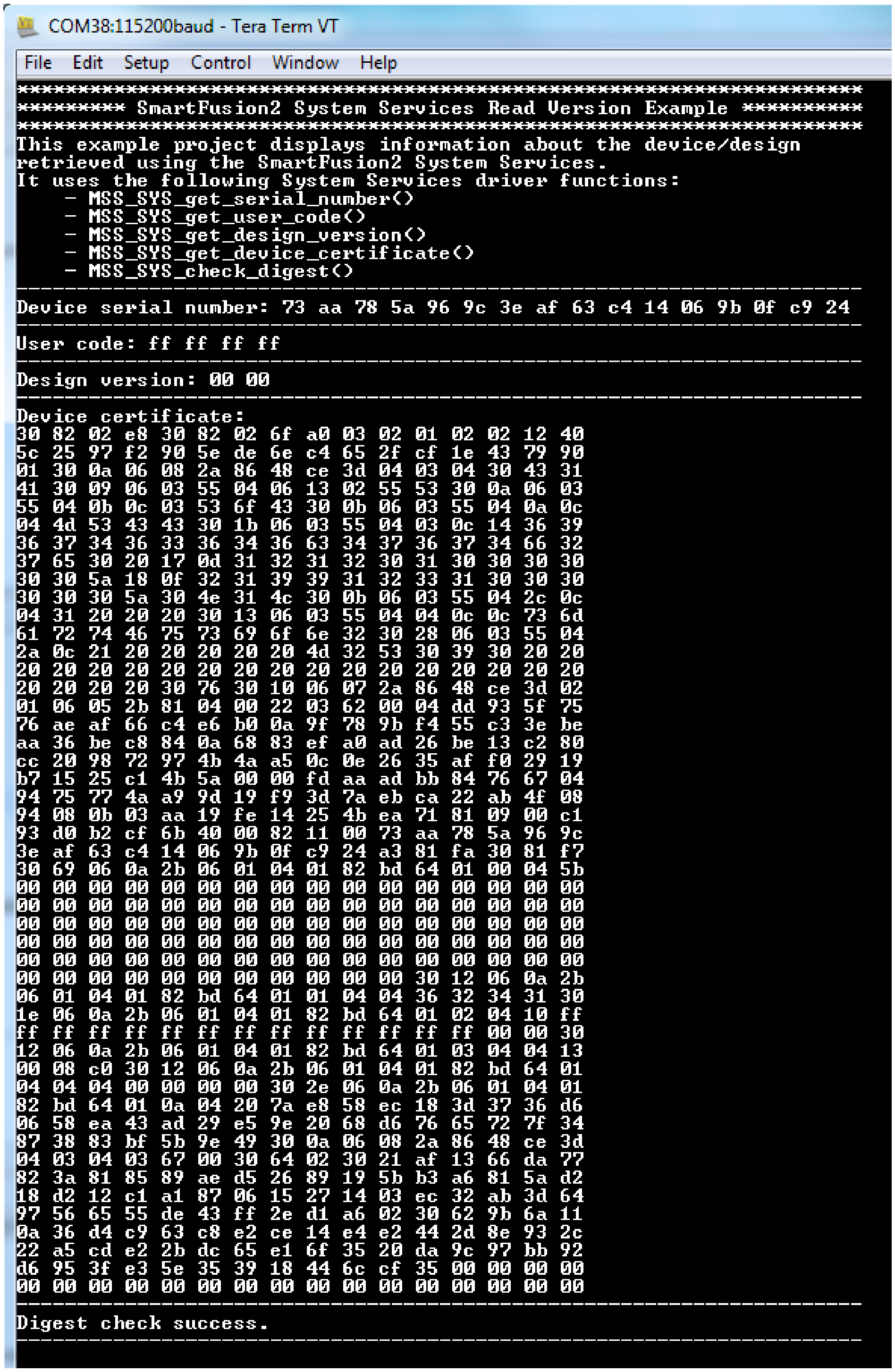
stdio.h
drivers/mss_sys_services/mss_sys_services.h
drivers/mss_uart/mss_uart.h
g_greeting_msg : const uint8_t[]
g_separator : const uint8_t[]
display_greeting(void) : void
display_hex_values(const uint8_t*, uint32_t) : void
gp_my_uart : mss_uart_instance_t* const
main() : int
display_greeting(void) : void
display_hex_values(const uint8_t*, uint32_t) : void

Sysservices_sb_MSS_CM3_app Debug [Microsemi Cortex-M3 Target] C:\Program Files (x86)\Microsemi\SoftConsole v3.4\Sourcery-G++\bin\arm-none-eabi-gdb.exe (12/23/14 6:19 PM)

```
arm-none-eabi-sprite: Target reset
0x00000190 in ?? ()
cont

Temporary breakpoint 1, main () at ../main.c:66
66          MSS_SYS_init(MSS_SYS_NO_EVENT_HANDLER);
```

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 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 00 30 12 06 0a 2b
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 00 00 30
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 97 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 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.

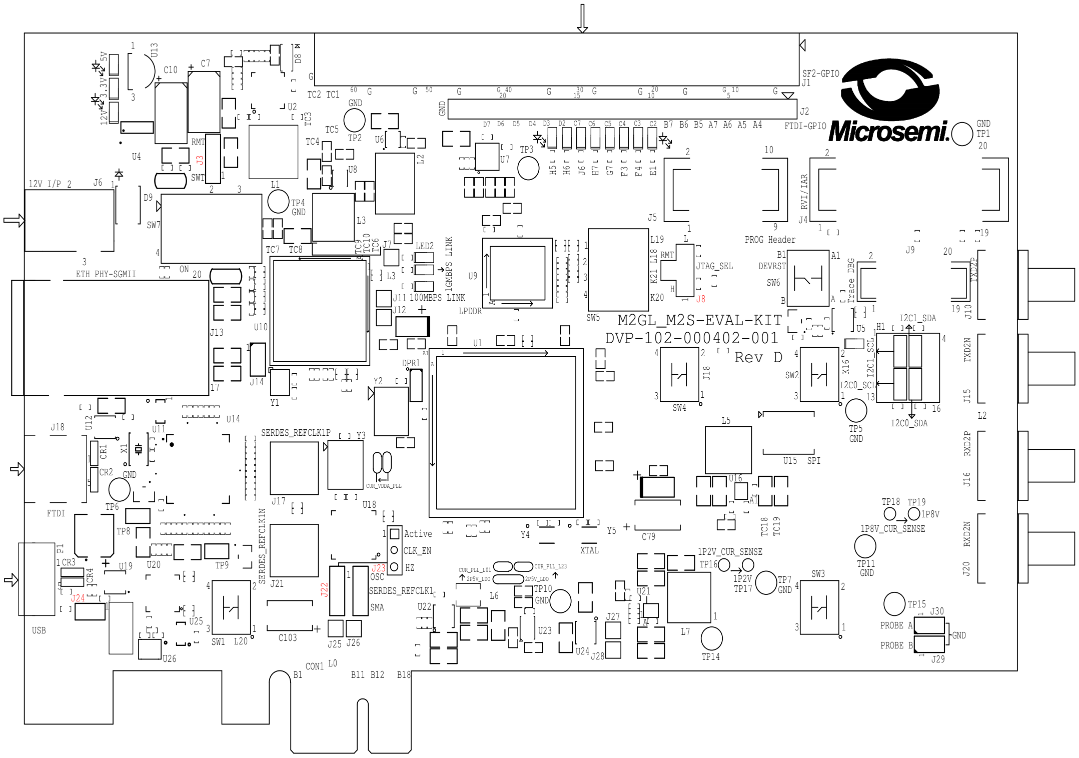
2.9 Conclusion

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.

3 Appendix: Jumper Locations

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

Figure 44 • **Jumper Locations**

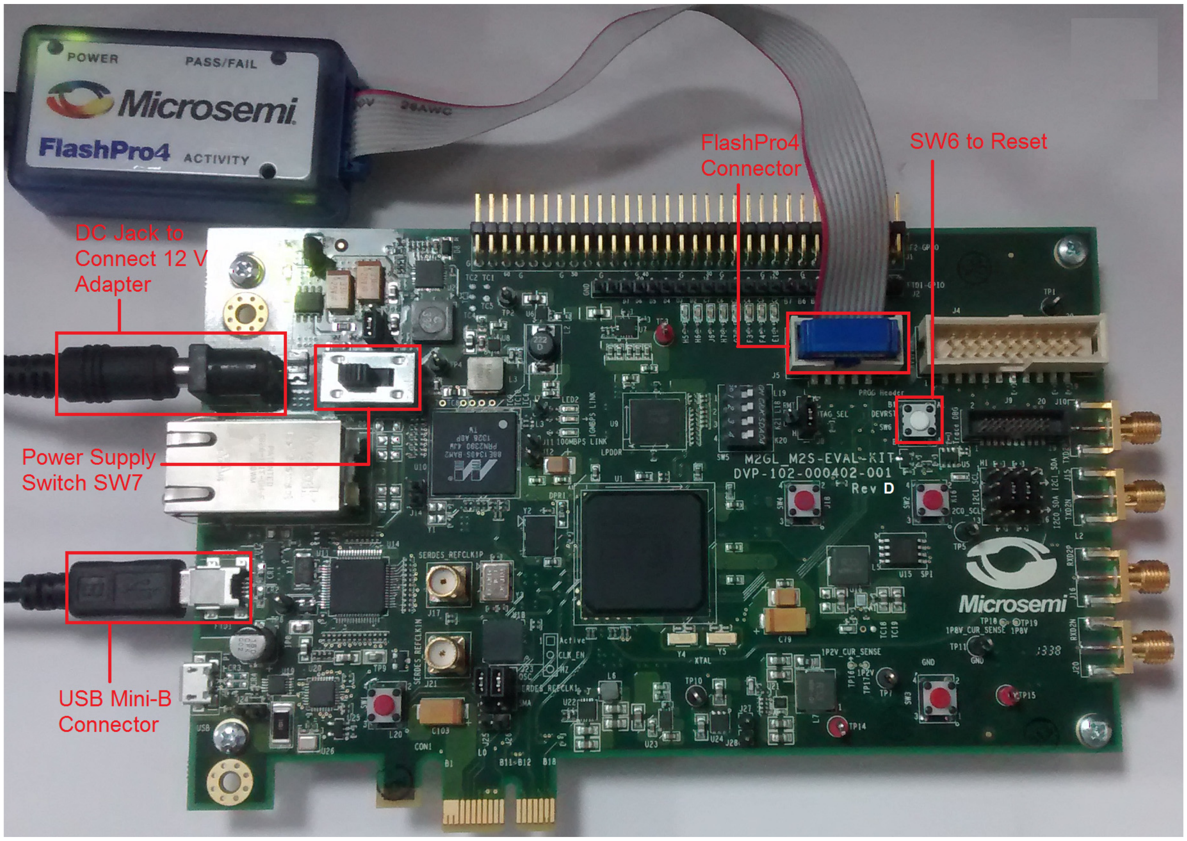


Note: The location of the jumpers in Figure 44 are searchable.

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

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.