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

TU0487 Tutorial
# 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 Sysservices_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

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1.</td>
<td>Demo Design Files Top-Level Structure</td>
<td>8</td>
</tr>
<tr>
<td>Figure 2.</td>
<td>Downloading Sample Project from Firmware Catalog</td>
<td>9</td>
</tr>
<tr>
<td>Figure 3.</td>
<td>Generate Sample Options Window</td>
<td>10</td>
</tr>
<tr>
<td>Figure 4.</td>
<td>Libero SoC Project Manager</td>
<td>10</td>
</tr>
<tr>
<td>Figure 5.</td>
<td>Project Details Page</td>
<td>11</td>
</tr>
<tr>
<td>Figure 6.</td>
<td>Device Selection Page</td>
<td>12</td>
</tr>
<tr>
<td>Figure 7.</td>
<td>Device Settings Page</td>
<td>13</td>
</tr>
<tr>
<td>Figure 8.</td>
<td>Device Template Page</td>
<td>13</td>
</tr>
<tr>
<td>Figure 9.</td>
<td>Add HDL Sources Page</td>
<td>14</td>
</tr>
<tr>
<td>Figure 10.</td>
<td>Add Constraints Page</td>
<td>14</td>
</tr>
<tr>
<td>Figure 11.</td>
<td>New Project Information Window</td>
<td>15</td>
</tr>
<tr>
<td>Figure 12.</td>
<td>System Builder Dialog Box</td>
<td>16</td>
</tr>
<tr>
<td>Figure 13.</td>
<td>System Builders - Device Features Page</td>
<td>16</td>
</tr>
<tr>
<td>Figure 14.</td>
<td>System Builder - Peripherals Page</td>
<td>17</td>
</tr>
<tr>
<td>Figure 15.</td>
<td>System Builder - Clocks Page</td>
<td>18</td>
</tr>
<tr>
<td>Figure 16.</td>
<td>Updating Sysse rvices_sb_0</td>
<td>18</td>
</tr>
<tr>
<td>Figure 17.</td>
<td>Sysservices_sb_0</td>
<td>19</td>
</tr>
<tr>
<td>Figure 18.</td>
<td>Opening DESGIN_FIRMWARE Window</td>
<td>20</td>
</tr>
<tr>
<td>Figure 19.</td>
<td>DESIGN_FIRMWARE Window</td>
<td>21</td>
</tr>
<tr>
<td>Figure 20.</td>
<td>Generate Component</td>
<td>22</td>
</tr>
<tr>
<td>Figure 21.</td>
<td>Generate Bitstream Data</td>
<td>22</td>
</tr>
<tr>
<td>Figure 22.</td>
<td>Export Firmware</td>
<td>23</td>
</tr>
<tr>
<td>Figure 23.</td>
<td>Export Firmware Dialog Box</td>
<td>23</td>
</tr>
<tr>
<td>Figure 24.</td>
<td>Firmware Successfully Exported Message</td>
<td>23</td>
</tr>
<tr>
<td>Figure 25.</td>
<td>Firmware Log Window</td>
<td>24</td>
</tr>
<tr>
<td>Figure 26.</td>
<td>Run Program Action</td>
<td>25</td>
</tr>
<tr>
<td>Figure 27.</td>
<td>Invoking SoftConsole IDE</td>
<td>26</td>
</tr>
<tr>
<td>Figure 28.</td>
<td>Importing Projects</td>
<td>26</td>
</tr>
<tr>
<td>Figure 29.</td>
<td>Importing Existing Projects</td>
<td>27</td>
</tr>
<tr>
<td>Figure 30.</td>
<td>Import Window</td>
<td>27</td>
</tr>
<tr>
<td>Figure 31.</td>
<td>Selecting System services</td>
<td>28</td>
</tr>
<tr>
<td>Figure 32.</td>
<td>Adding Projects to SoftConsole IDE</td>
<td>28</td>
</tr>
<tr>
<td>Figure 33.</td>
<td>SoftConsole Workspace</td>
<td>29</td>
</tr>
<tr>
<td>Figure 34.</td>
<td>Sample Project main. c File</td>
<td>29</td>
</tr>
<tr>
<td>Figure 35.</td>
<td>SoftConsole Workspace - main.c File</td>
<td>30</td>
</tr>
<tr>
<td>Figure 36.</td>
<td>Project Explorer Window - SoftConsole Project</td>
<td>30</td>
</tr>
<tr>
<td>Figure 37.</td>
<td>Sysservices_sb_MSS_CM3 Properties Window</td>
<td>31</td>
</tr>
<tr>
<td>Figure 38.</td>
<td>Settings for Clean Build</td>
<td>32</td>
</tr>
<tr>
<td>Figure 39.</td>
<td>Device Manager Window</td>
<td>33</td>
</tr>
<tr>
<td>Figure 40.</td>
<td>Debug Configurations Window</td>
<td>34</td>
</tr>
<tr>
<td>Figure 41.</td>
<td>Confirm Perspective Switch</td>
<td>34</td>
</tr>
<tr>
<td>Figure 42.</td>
<td>SoftConsole Debugger Perspective</td>
<td>35</td>
</tr>
<tr>
<td>Figure 43.</td>
<td>PuTTY Window</td>
<td>36</td>
</tr>
<tr>
<td>Figure 44.</td>
<td>Jumper Locations</td>
<td>38</td>
</tr>
<tr>
<td>Figure 45.</td>
<td>SmartFusion2 Security Evaluation Kit</td>
<td>39</td>
</tr>
<tr>
<td>Table 1.</td>
<td>Design Requirements</td>
<td>7</td>
</tr>
<tr>
<td>Table 2.</td>
<td>SmartFusion2 Security Evaluation Kit Jumper Settings</td>
<td>24</td>
</tr>
</tbody>
</table>
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
2.1 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>
<thead>
<tr>
<th>Design Requirements</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hardware Requirements</strong></td>
<td></td>
</tr>
<tr>
<td>SmartFusion2 Security Evaluation Kit:</td>
<td>Rev D or later</td>
</tr>
<tr>
<td>• FlashPro4 programmer</td>
<td></td>
</tr>
<tr>
<td>• USB A to Mini-B cable</td>
<td></td>
</tr>
<tr>
<td>• 12 V Adapter</td>
<td></td>
</tr>
<tr>
<td>Host PC or Laptop</td>
<td>Any 64-bit Windows Operating System</td>
</tr>
</tbody>
</table>

**Software Requirements**

<table>
<thead>
<tr>
<th>Design Requirements</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Libero SoC</td>
<td>v11.7</td>
</tr>
<tr>
<td>SoftConsole</td>
<td>v3.4 SPI*</td>
</tr>
<tr>
<td>FlashPro programming software</td>
<td>v11.7</td>
</tr>
<tr>
<td>Host PC Drivers</td>
<td>USB to UART drivers</td>
</tr>
<tr>
<td>Any one of the following serial terminal emulation programs:</td>
<td>–</td>
</tr>
<tr>
<td>• HyperTerminal</td>
<td></td>
</tr>
<tr>
<td>• TeraTerm</td>
<td></td>
</tr>
<tr>
<td>• PuTTY</td>
<td></td>
</tr>
</tbody>
</table>

**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

```
<download_folder>
  m2s_tu0487_creating_libero
  _project_libero1ip7_df
    SF2_Sysservices
    Programmingfile
    SF2_GNU_SysServ_read_version
    Source Files
    readme.txt
```

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

**Figure 7 • Device Settings Page**

![Device Settings Page]

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**

![Device Template Page]
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 - 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 INIT_DONE and select Mark Unused.
4. Expand FAB_CCC_PINS, right-click FAB_CCC_GL0 and select Mark Unused.
5. Right-click FAB_CCC_LOCK and select Mark Unused.
6. Right-click FAB_RESET_N and select Tie High.
7. Click File > Save.

The 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
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*
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.
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>
<thead>
<tr>
<th>Jumper Number</th>
<th>Pin (From)</th>
<th>Pin (To)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>J22, J23, J24, J8, J3</td>
<td>1</td>
<td>2</td>
<td>These are the default jumper settings of the SmartFusion2 Security Evaluation Kit board. Ensure that these jumpers are set accordingly.</td>
</tr>
</tbody>
</table>

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**

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 Sysservices 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**

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**
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* • Sysservices_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](http://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**: 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**
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**

![PuTTY Window](image)

19. Terminate execution of the code by choosing Run > Terminate.

20. Close Debug Perspective by selecting Close Perspective from the Window menu.


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.
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.

<table>
<thead>
<tr>
<th>Revision</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revision 5</td>
<td>Updated the document for Libero v11.7 software release (SAR 78428).</td>
</tr>
<tr>
<td>(April 2016)</td>
<td></td>
</tr>
<tr>
<td>Revision 4</td>
<td>Updated the document for Libero v11.6 software release (SAR 72552).</td>
</tr>
<tr>
<td>(October 2015)</td>
<td></td>
</tr>
<tr>
<td>Revision 3</td>
<td>Updated the document for Libero v11.5 software release (SAR 64799).</td>
</tr>
<tr>
<td>(February 2015)</td>
<td></td>
</tr>
<tr>
<td>Revision 2</td>
<td>Updated the document for Libero v11.4 software release (SAR 61636).</td>
</tr>
<tr>
<td>(October 2014)</td>
<td></td>
</tr>
<tr>
<td>Revision 1</td>
<td>Initial release.</td>
</tr>
<tr>
<td>(April 2014)</td>
<td></td>
</tr>
</tbody>
</table>
Product Support

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


6.4 Website


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 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.