SoftConsole v4.0 and Libero SoC v11.6

TU0546 Tutorial

Microsemi

Table of Contents

SoftConsole v4.0 and Libero SoC v11.6

Introduction

This tutorial describes how to implement an ARM® Cortex®-M3 design using Libero® System-on-Chip v11.6 and describes the process to build a simple LED blink application using SoftConsole v4.0.

After completing this tutorial, you will be able to perform the following tasks:

- Create a Libero SoC project using System Builder
- Generate the programming file to program the SmartFusion®2 SoC field programmable gate array (FPGA) device
- Creating a SoftConsole v4.0 project
- Compile application code
- Debug and run code using SoftConsole

Design Requirements

Table 1 • **Design Requirements**

Design Requirements	Description
Hardware Requirements	
SmartFusion2 Security Evaluation Kit: • FlashPro4 or FlashPro5 programmer • 12 V adapter	Rev D or later
SmartFusion2 Advanced Development Kit: • FlashPro4 or FlashPro5 programmer • 12 V adapter	Rev B or later
SmartFusion2 Starter Kit: • FlashPro4 or FlashPro5 programmer • USB A to Mini-B cable	SmartFusion2-484-Starter-Kit (M2S010-FGG484)
Host PC or Laptop	Any 64-bit Windows Operating System
Software Requirements	
Libero SoC	v11.6
SoftConsole	v4.0
FlashPro programming software	v11.6

Associated Project Files

Download design files for this tutorial from the Microsemi website:
http://soc.microsemi.com/download/rsc/?f=m2s_tu0546_liberov11p6_df

The demo design files include:

- LiberoProject
- Sourcefiles
- Readme file

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

Design Overview

This tutorial demonstrates the simple LED blink application for SmartFusion2 Device. MSS GPIOs are configured as outputs and connected to LEDs using Fabric I\Os. This Tutorial can be used on the following SmartFusion2 boards:

- SmartFusion2 Security Evaluation Kit
- SmartFusion2 Advanced Development Kit
- SmartFusion2 Starter Kit (M2S010-FGG484)

Step 1: Creating a Libero SoC Project

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

Launching Libero SoC

1. Click **Start > Programs > Microsemi Libero SoC v11.6 > Libero SoC v11.6**, or double-click the shortcut on desktop to open the Libero SoC v11.6 Project Manager.
2. Create a new project by selecting **New** on the **Start Page** tab (highlighted in Figure 1), or by clicking **Project > New Project** from the Libero SoC menu.

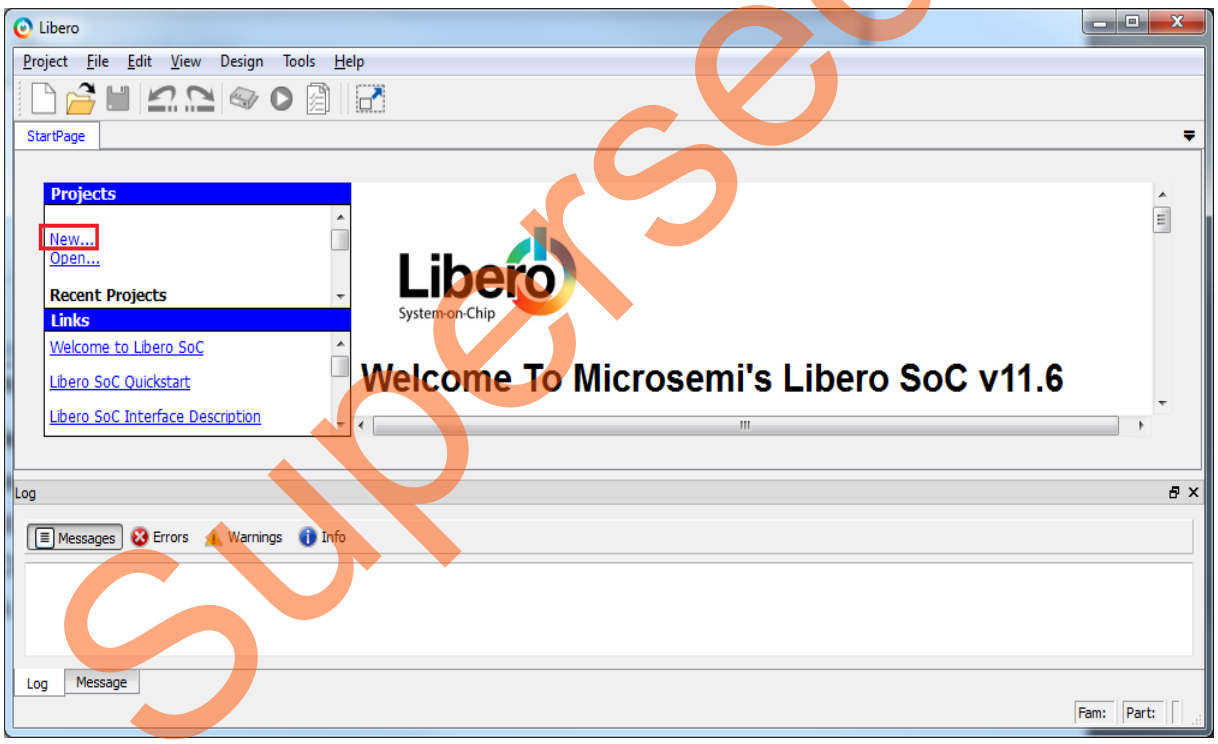


***Figure 1* • Libero SoC Project Manager**

3. In the **Project Details** window, enter the following information as shown in Figure 2.
   – **Project Name**: LED_Blink
   – **Project Location**: Select an appropriate location (for example, *D:/Microsemi_prj*)
   – **Preferred HDL type**: Verilog or VHDL
   – **Enable Block Creation**: Unchecked

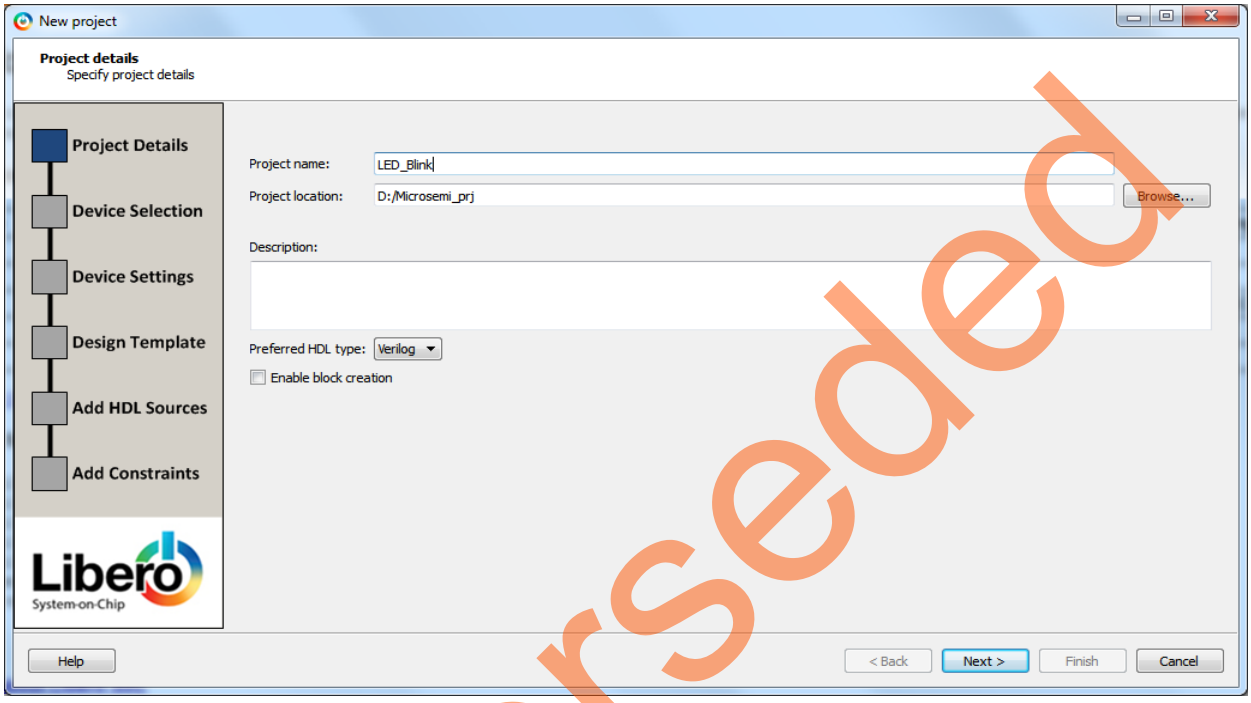


Figure 2 • **Project Details Window**

4. Click **Next**. In the **Device Selection** window, select the information displayed in Figure 3 on page 7. In the Part Filter, select the values using the drop-down lists, as shown in Table 2.
   – **Family**: SmartFusion2

Table 2 • **SmartFusion2 Devices Selection**

Board	Die	Package	Speed	Core Voltage	Range	PLL Supply Voltage
SmartFusion2 Security Evaluation Kit	M2S090TS	484 FBGA	-1	1.2	COM	3.3
SmartFusion2 Advanced Development Kit	M2S150T	1152 FC	-1	1.2	COM	3.3
SmartFusion2 Starter Kit	M2S010	484 FBGA	STD	1.2	COM	2.5

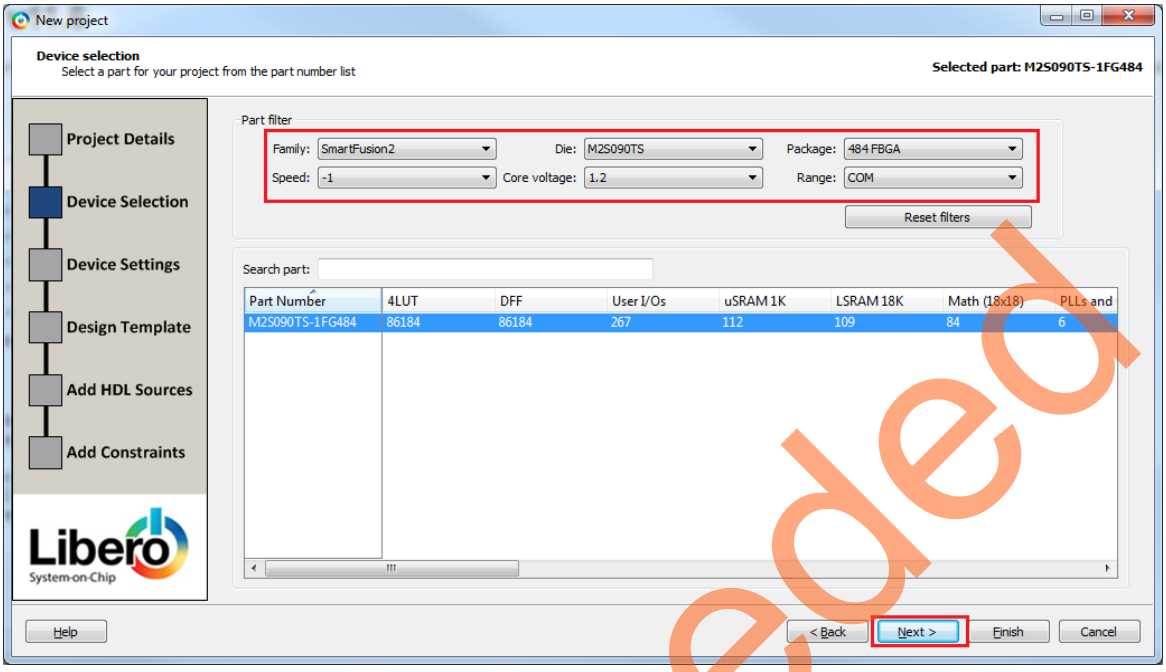


Figure 3 • **Device Selection Window**

5. Click **Next**. The **Device Settings** window is displayed. Select PLL Supply Voltage (V), as shown in Figure 4.

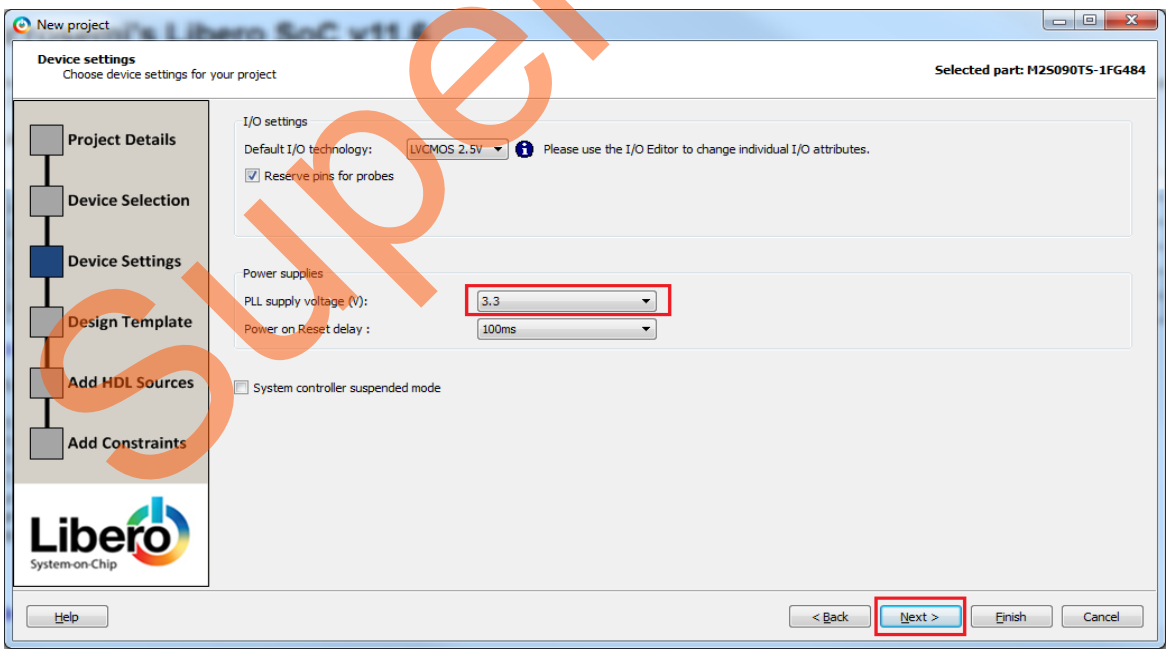


Figure 4 • **Device Settings**

Refer to Table 2 on page 6 for specific board values.

6. Click **Next**. In the **Design Template** page, select **Create a system builder based design** check box under the **Design Templates and Creators** as shown in Figure 5.

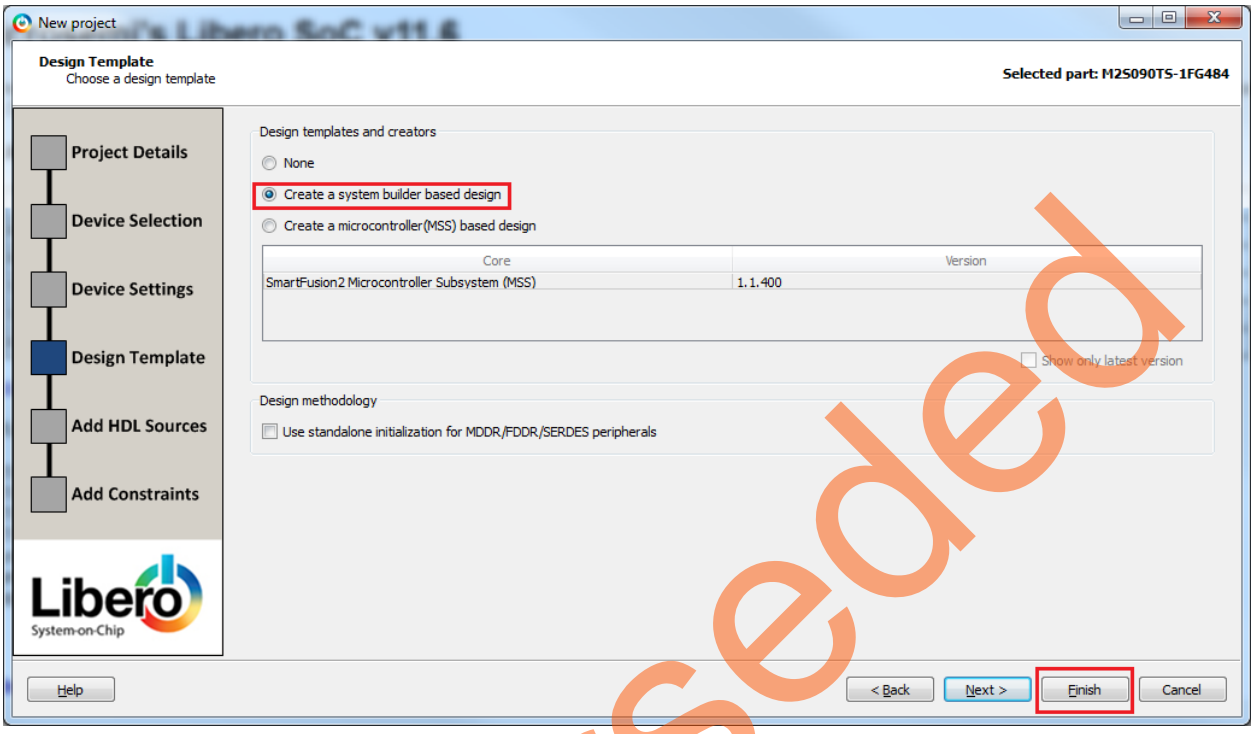


***Figure 5* • Design Template Window**

7. Click **Finish**. A System Builder dialog box is displayed.

   Note: System Builder is a graphical design wizard. It creates a design based on high-level design specifications by taking the user through a set of high-level questions that define the intended system.

8. Enter a name for your system as **LED_Blink** and click **OK**, as shown in Figure 6.

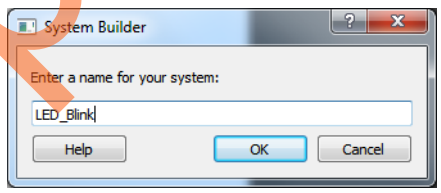


***Figure 6* • System Builder Dialog Box**

System Builder – Device Features page is displayed, as shown in Figure 7.

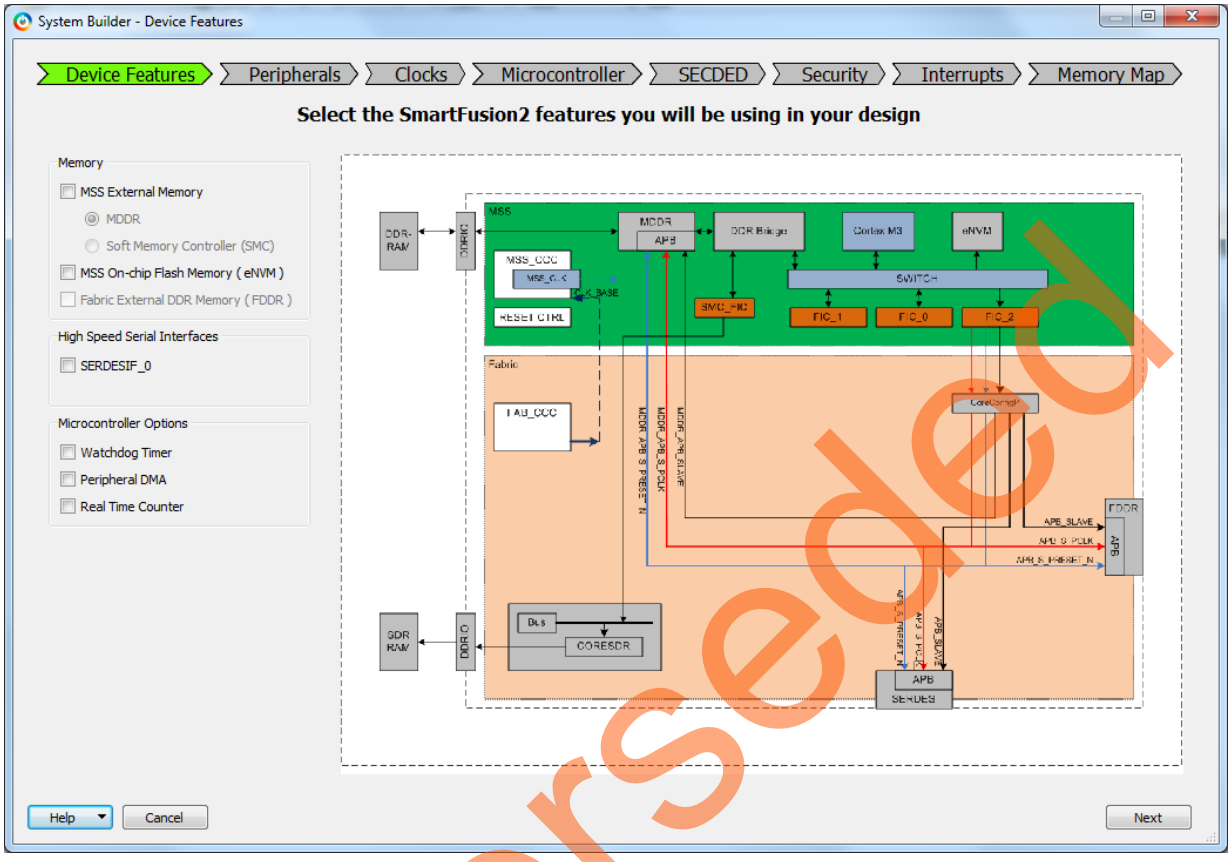


***Figure 7* • System Builder – Device Features Page**

9. Retain the default values. Click **Next**, the **System Builder – Peripherals** page is displayed. Under the MSS Peripherals section, clear all the check boxes except **MSS_GPIO**, as shown in Figure 8.

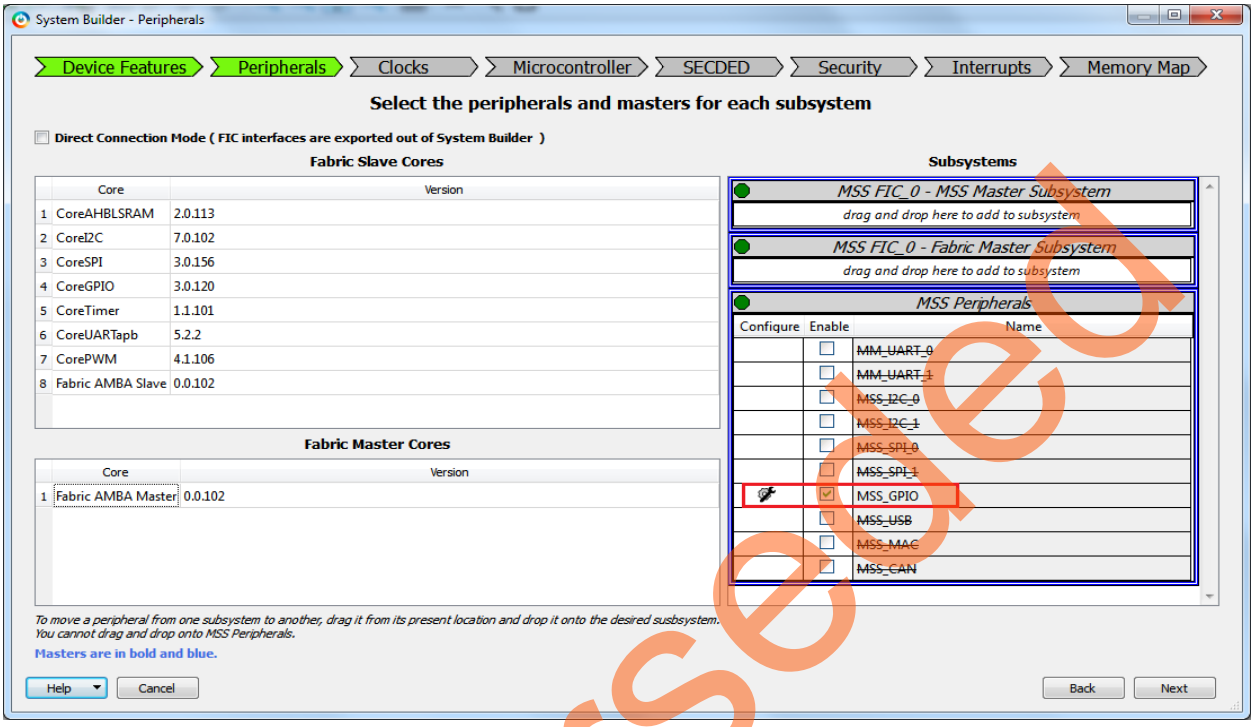


***Figure 8* • System Builder Configurator – Select Peripherals Page**

10. Double-click the **wrench** symbol for the MSS_GPIO peripheral to open the MSS_GPIO Configurator.

This design requires configuring GPIOs to drive LEDs on the board, Configure the GPIOs as shown below:

- **Set/Reset Definition** accept default settings
- Configure GPIO as per Table 3

***Table 3* • SmartFusion2 GPIO Configuration**

Board	Die	GPIO ID	Direction	Package Pin	Connectivity
SmartFusion2 Security Evaluation Kit	M2S090TS	GPIO_0 to GPIO_7	Output	NA	FABRIC_A
SmartFusion2 Advanced Development Kit	M2S150T	GPIO_0 to GPIO_7	Output	NA	FABRIC_A
SmartFusion2 Starter Kit	M2S010	GPIO_0 to GPIO_1	Output	NA	FABRIC_A

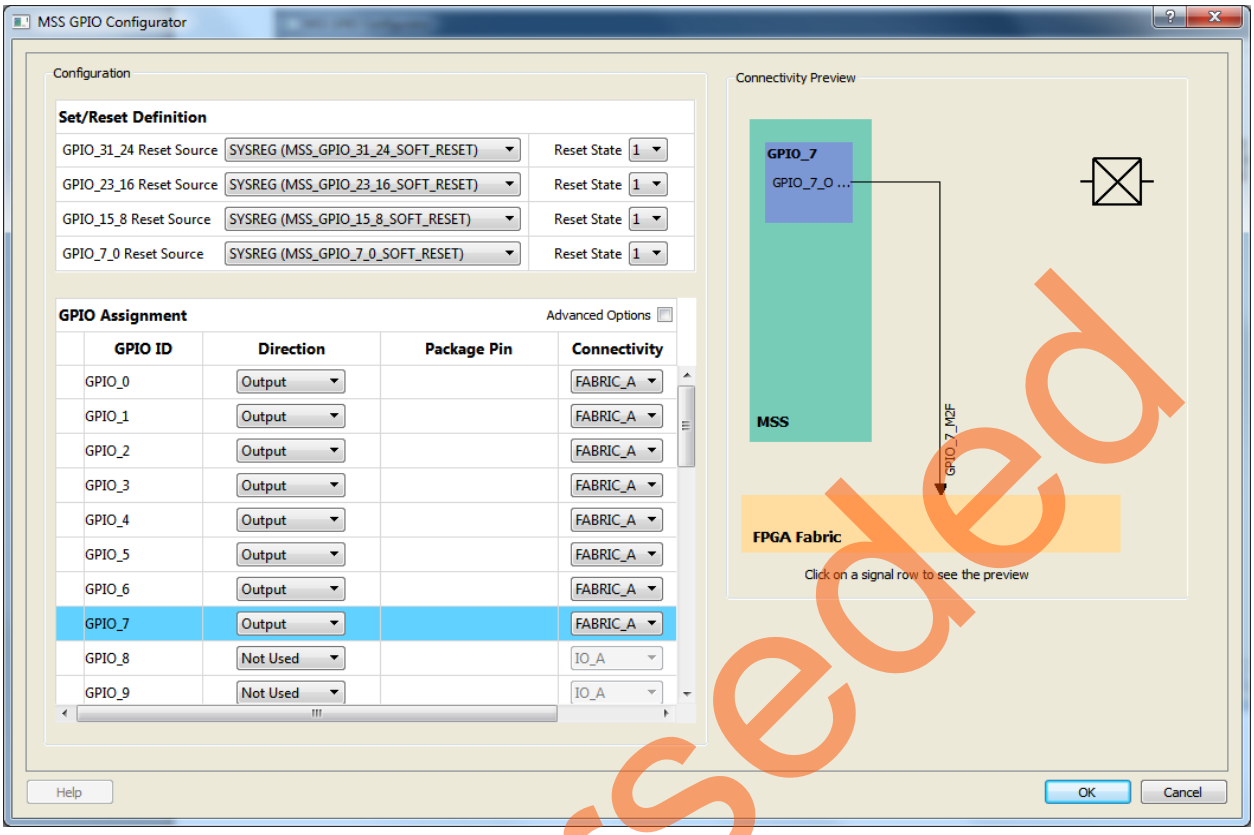


***Figure 9* • MSS_GPIO Configurator**

11. Click **Next**, the **System Builder – ClocksSettings** page is displayed, as shown in Figure 10. Select **System Clock** source as **On-chip 25/50 MHz RC Oscillator**. The M3_CLK is configured to 100 MHz by default.

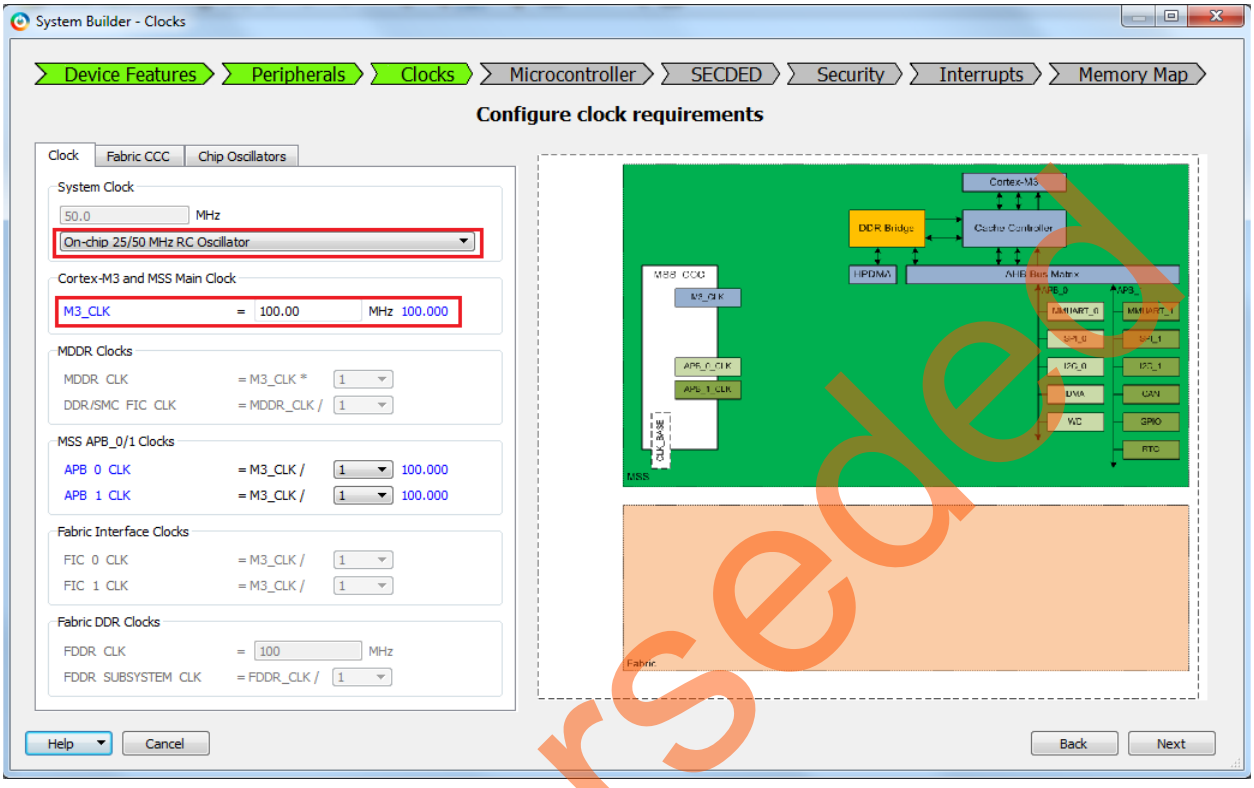


***Figure 10* • System Builder Configurator – Clock Settings Page**

12. Click **Next**, the **System Builder – Microcontroller Options** page is displayed.
    – Retain the default values.
13. Click **Next**, the **System Builder – SECDED Options** page is displayed.
    – Retain the default values.
14. Click **Next**, the **System Builder – Security Options** page is displayed.
    – Retain the default values.
15. Click **Next**, the **System Builder – Interrupts Options** page is displayed.
    – Retain the default values.
16. Click **Next**, the **System Builder – Memory Map Options** page is displayed.
    – Retain the default values.
17. Click **Finish**.
    System Builder generates the system based on the selected options.

System Builder block is created and added to the Libero SoC project, as shown in Figure 11.

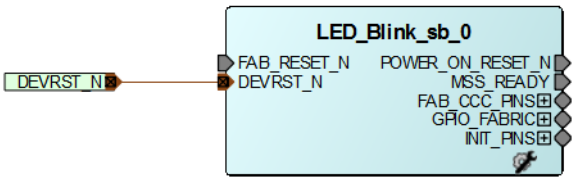


Figure 11 • **System Builder Generated System**

Connecting Components in LED_Blink SmartDesign

The following steps describe connecting components in LED_Blink SmartDesign:

1. Connect the pins as follows:
   - Tie the **FAB_RESET_N** to high by right-clicking and selecting **Tie High**.
   - Mark the output port **POWER_ON_RESET_N** as unused by right-clicking and selecting **Mark Unused**.
   - Mark the output port **MSS_READY** as unused by right-clicking and selecting **Mark Unused**.
   - Expand **INIT_PINS**, right-click **INIT_DONE** and select **Mark Unused**.
   - Expand **FAB_CCC_PINS**, right-click **FAB_CCC_GL0** and select **Mark Unused**.
   - Right-click **FAB_CCC_LOCK** and select **Mark Unused**.
   - Right-click **GPIO_FABRIC** and select **Promote to Top Level**.
2. Click **File** > **Save**. The LED_Blink design is displayed, as shown in Figure 12.

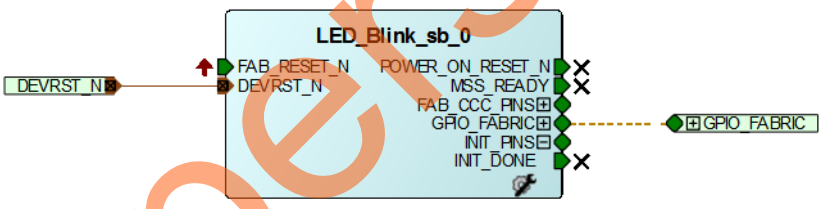


Figure 12 • **LED_Blink Design**

3. Generate the LED_Blink Smart Design by clicking **SmartDesign** > **Generate Component** or by clicking Generate Component on the SmartDesign toolbar.

Figure 13 • **Generate Component**

After successful generation of the system, the message **'info: LED_Blink' was successfully generated** is displayed on the Libero SoC **Log** window, as shown in Figure 14.

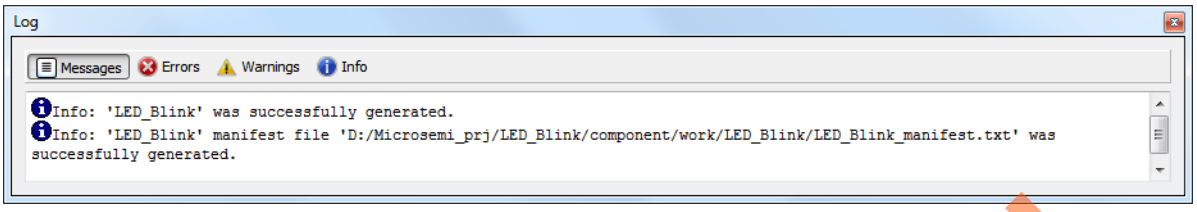


Figure 14 • **Log Window**

Step 2: Generating the Program File

1. Double-click **I/O Constraints** in the **Design Flow** window, as shown in Figure 15. The **I/O Editor** window is displayed after completing Synthesize and Compile.

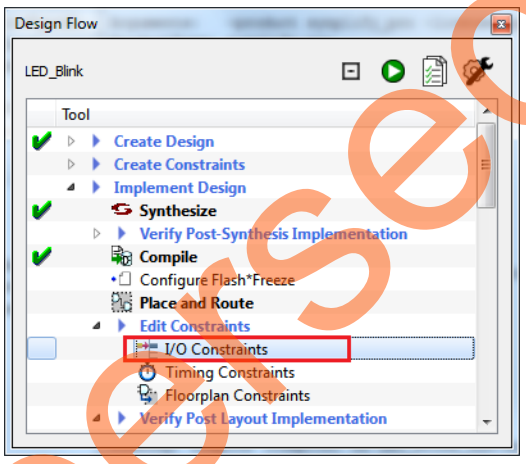


Figure 15 • **I/O Constraints**

2. Make the pin assignments, as shown in Table 4. After the pins have been assigned, the I/O Editor is displayed, as shown in Figure 16 on page 15.

Table 4 • **Port to Pin Mapping**

Port Name	Pin Number
SmartFusion2 Security Evaluation Kit	
GPIO_0_M2F	H5
GPIO_1_M2F	H6
GPIO_2_M2F	J6
GPIO_3_M2F	H7
GPIO_4_M2F	G7
GPIO_5_M2F	F3
GPIO_6_M2F	F4
GPIO_7_M2F	E1

Table 4 • **Port to Pin Mapping (continued)**

SmartFusion2 Advanced Development Kit	
GPIO_0_M2F	D26
GPIO_1_M2F	F26
GPIO_2_M2F	A27
GPIO_3_M2F	C26
GPIO_4_M2F	C28
GPIO_5_M2F	B27
GPIO_6_M2F	C27
GPIO_7_M2F	E26
SmartFusion2 Starter Kit	
GPIO_0_M2F	AB18
GPIO_1_M2F	P1

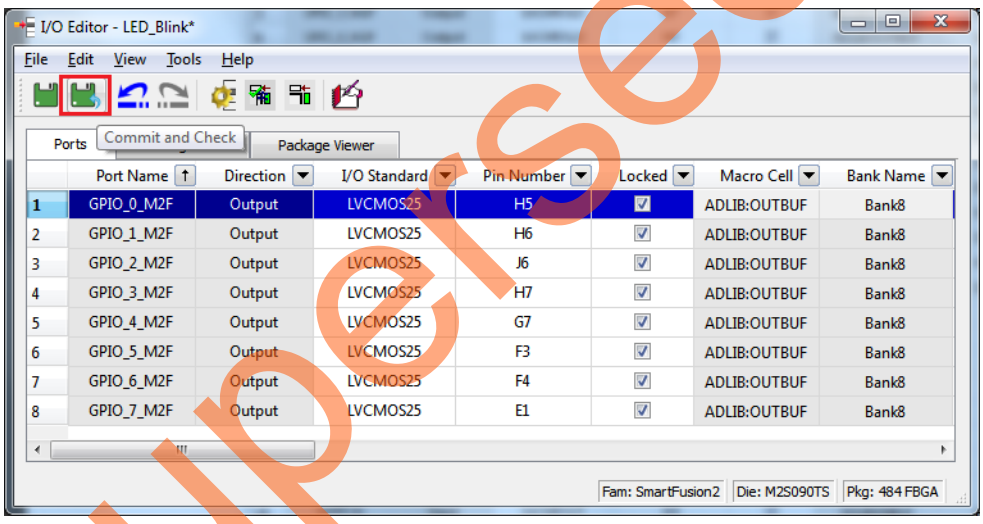


Figure 16 • **I/O Editor**

3. After updating I/O editor, click **Commit and Check**.
4. Close the **I/O Editor**.
5. Click **Generate Bitstream** as shown in Figure 17 to generate the programming file.

Figure 17 • **Generate Bitstream**

Step 3: Programming the SmartFusion2 Board Using FlashPro

Jumper settings for the supported target boards and Board setup for running the tutorial are given in "Appendix A: Board Setup for SmartFusion2 Security Evaluation Kit" section, "Appendix B: Board Setup for SmartFusion2 Advanced Development Kit" section, and "Appendix C: Board Setup for SmartFusion2 Starter Kit" section.

1. To program the SmartFusion2 device, double-click **Run PROGRAM Action** in the **Design Flow** window, as shown in Figure 18.

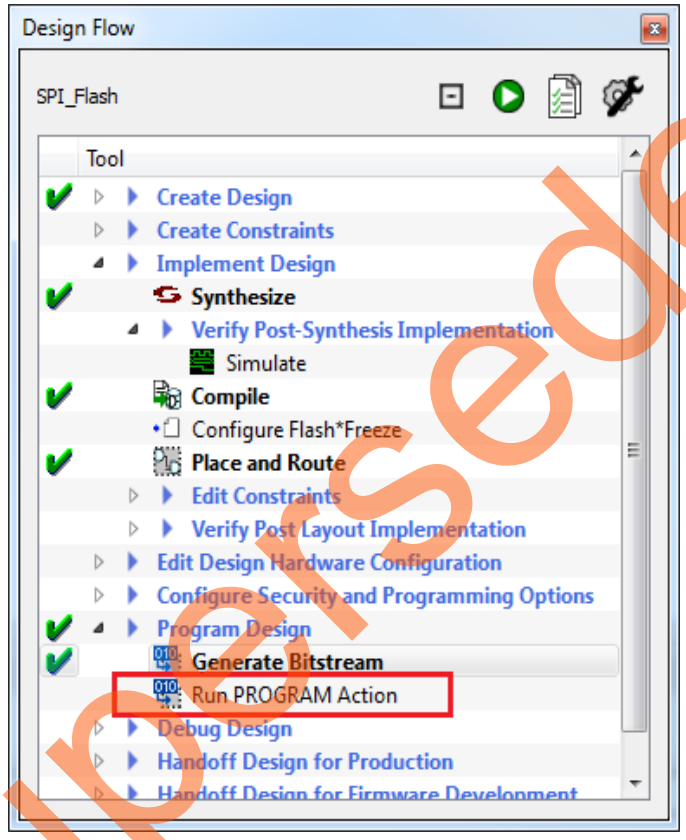


Figure 18 • **Run Programming Action**

Step 4: Creating Software Project using SoftConsole 4.0

Configuring and Generating Firmware

The **Design Firmware** window displays compatible firmware drivers based on peripherals configured in the design. Following drivers are used in this tutorial:

- CMSIS
- GPIO

To generate the required drivers:

1. Double-click **Configure Firmware Cores** under **Handoff design for Firmware Development** in **Design Flow** window, as shown in Figure 19.

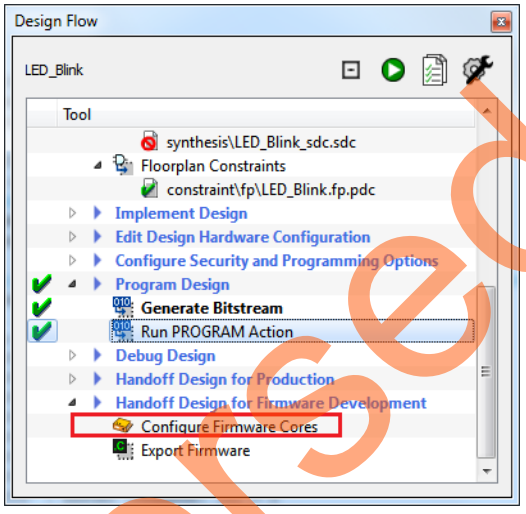


***Figure 19* • Design Flow**

2. Clear all the drivers check boxes, except **CMSIS**, **GPIO** in **Design firmware** window, as shown in Figure 20.

   Note: Select the latest version of the drivers.

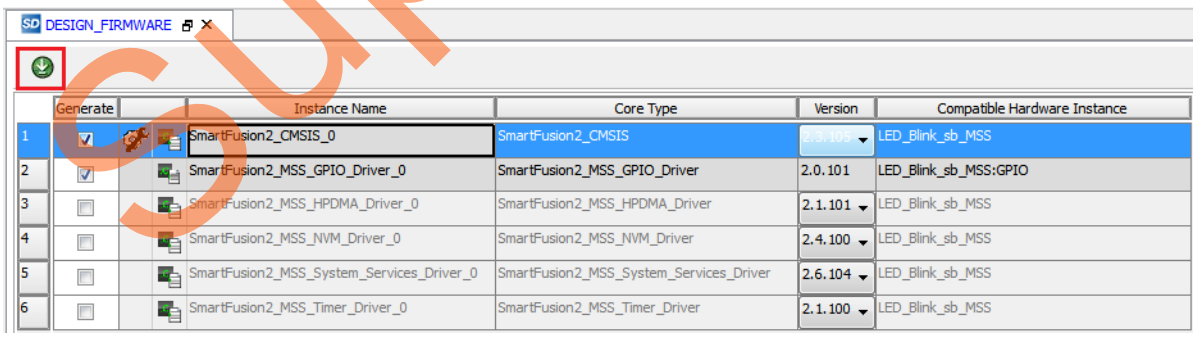


***Figure 20* • Design Firmware**

3. Click **Download** highlighted in red to download the latest version of drivers for peripherals, as shown in Figure 20. Close **DESIGN_FIRMWARE** tab.
4. Double-click on **Export Firmware** in **Handoff design for Firmware Development** in the Design Flow window.

Export Firmware dialog box is displayed as shown in Figure 21.

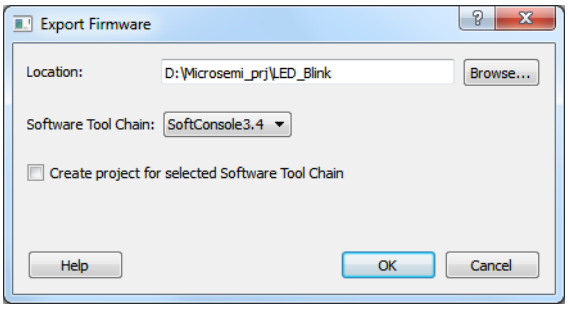


***Figure 21* • Export Firmware Dialog Box**

5. Click **OK**, a notification window appears saying **Firmware project was successfully exported to <drive:\>Microsemi_prj\LED_Blink**, as shown in Figure 22.
6. Click **OK**

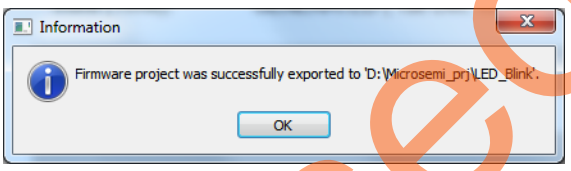


***Figure 22* • Firmware Export Successful**

Building Software Application using SoftConsole 4.0

1. Click **Start > Programs > Microsemi SoftConsole v4.0 > Microsemi SoftConsole v4.0** to open the SoftConsole IDE. The SoftConsole **Workspace Launcher** window is displayed.
2. Browse to the location to select D:\Microsemi_prj\LED_Blink\SoftConsole, as shown in Figure 23.

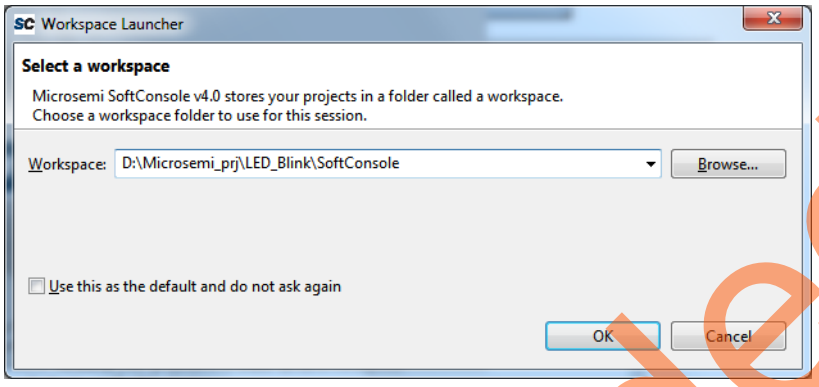


Figure 23 • **Workspace Launcher**

The SoftConsole workspace is displayed, as shown in Figure 24.

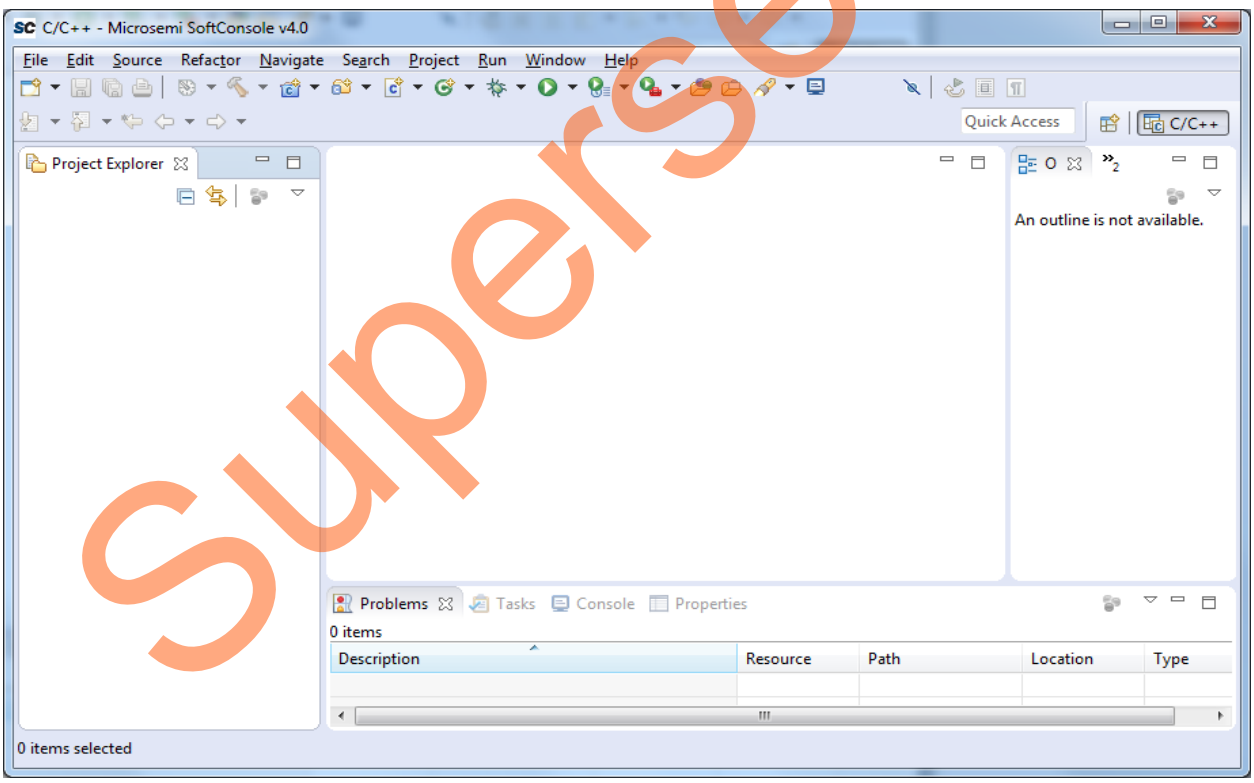


Figure 24 • **SoftConsole Window**

3. Click **File >New >C project** as shown in Figure 25.

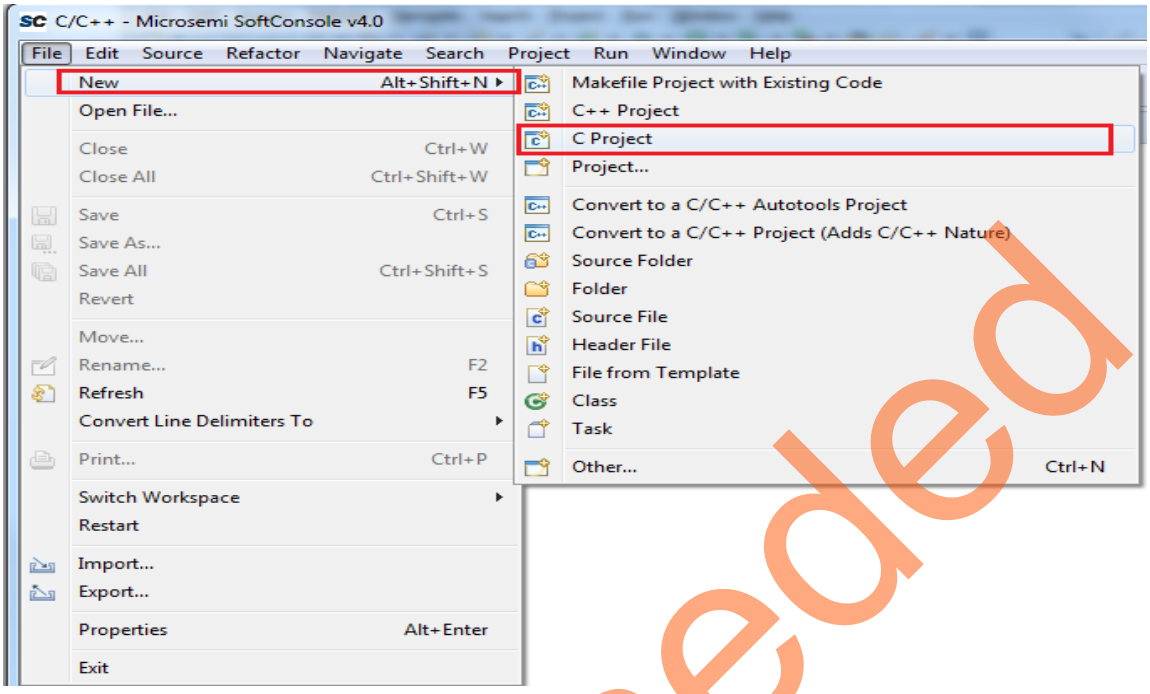


***Figure 25* • Creating New C Project**

4. Enter Project name as **LED_Blink**, as shown in Figure 26.

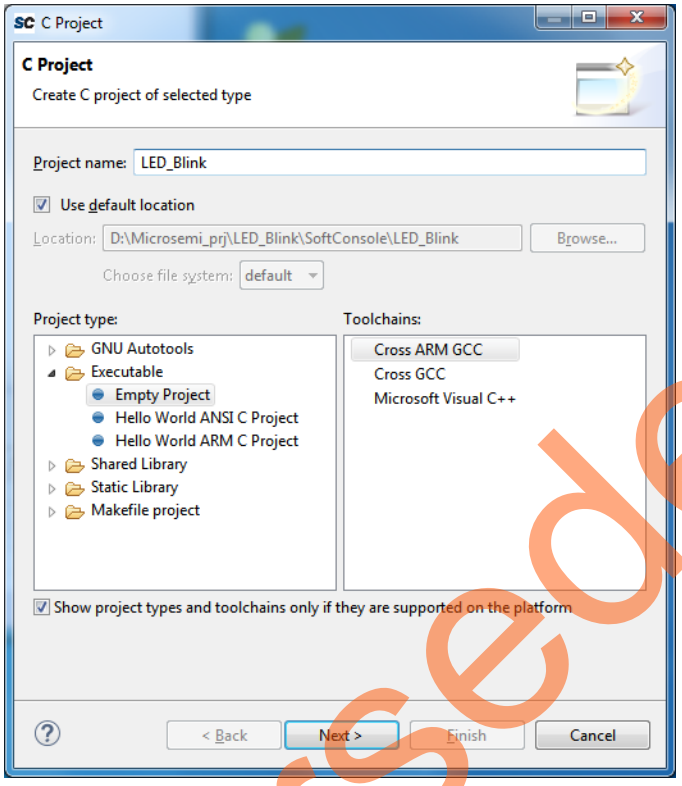


***Figure 26* • C Project Window**

5. Click **Next**, **Select Configurations** window is displayed, as shown in Figure 27.

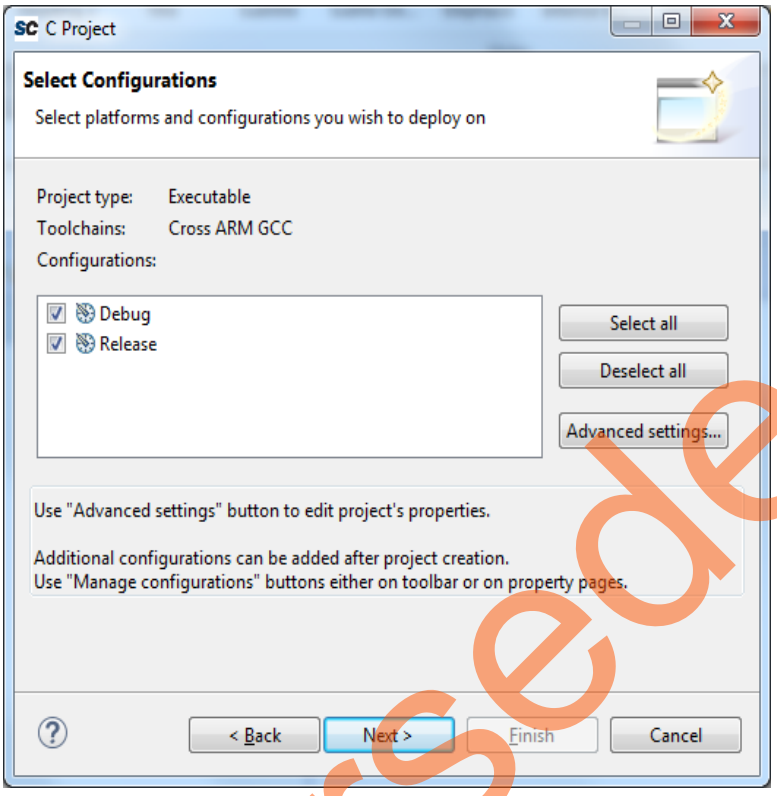


***Figure 27* • C Project - Select Configuration**

6. Do not change default settings. Click **Next**.
   Cross GNU ARM Tool chain window is displayed, as shown in Figure 28.

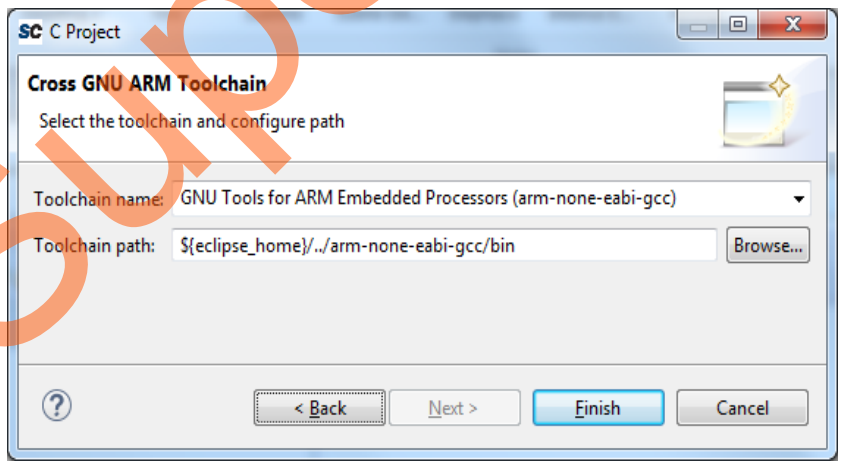


***Figure 28* • C Project - Cross GNU ARM Tool Chain**

7. Click **Finish**.

8. Right-click **LED_Blink** and then click **Import** as shown in Figure 29.

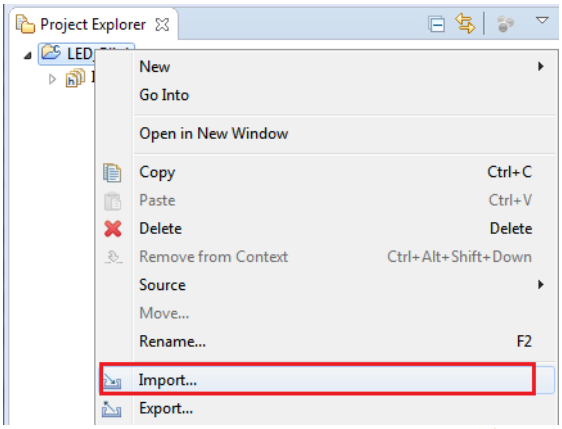


***Figure 29* • Project Explorer - Import**

9. **Import** window is displayed, as shown Figure 30.
10. Click **File System** and then click **Next** as shown in Figure 30.

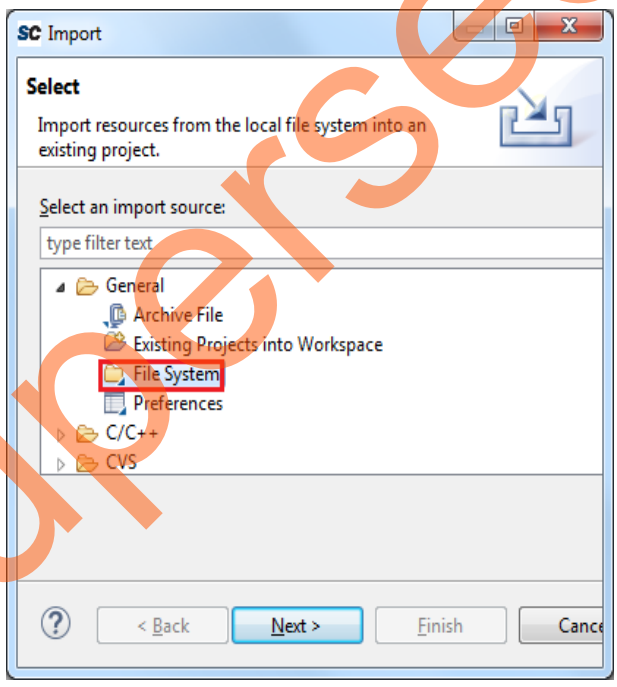


***Figure 30* • Import Window**

11. Browse to **D:\Microsemi_prj\LED_Blink\firmware** and select **firmware** check box, as shown in Figure 31.

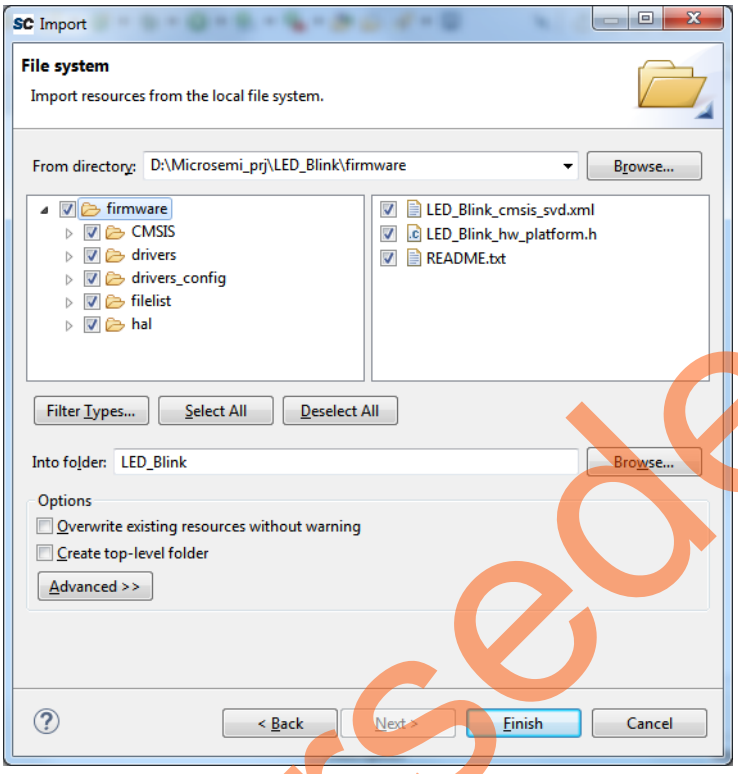


***Figure 31* • Import - File System**

12. Click **Finish**.

    Note: If any changes are made to the Libero project. Firmware need to be exported from Libero and new firmware should be imported to **LED_Blink**.

13. Using Windows explorer, browse to the main.c file location in the design files folder:
    - For SmartFusion2 Security Evaluation Kit:
      <download_folder>\SF2_LED_Blink_SC_Tutorial_DF\Source Files\SF2_Security_Kit.
    - For SmartFusion2 Advanced Development Kit:
      <download_folder>\SF2_LED_Blink_SC_Tutorial_DF\Source Files\SF2_Adv_Dev_Kit.
    - For SmartFusion2 Starter Kit:
      <download_folder>\SF2_LED_Blink_SC_Tutorial_DF\Source Files\SF2_Starter_Kit.

14. Copy the main.c file to the **LED_Blink** project in the SoftConsole workspace, as shown in Figure 32.

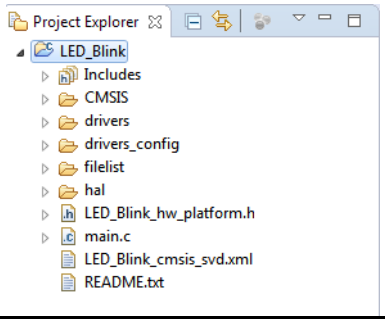


***Figure 32* • Project Explorer**

15. Right-click **LED_Blink** and click **Properties,** as shown in Figure 33.

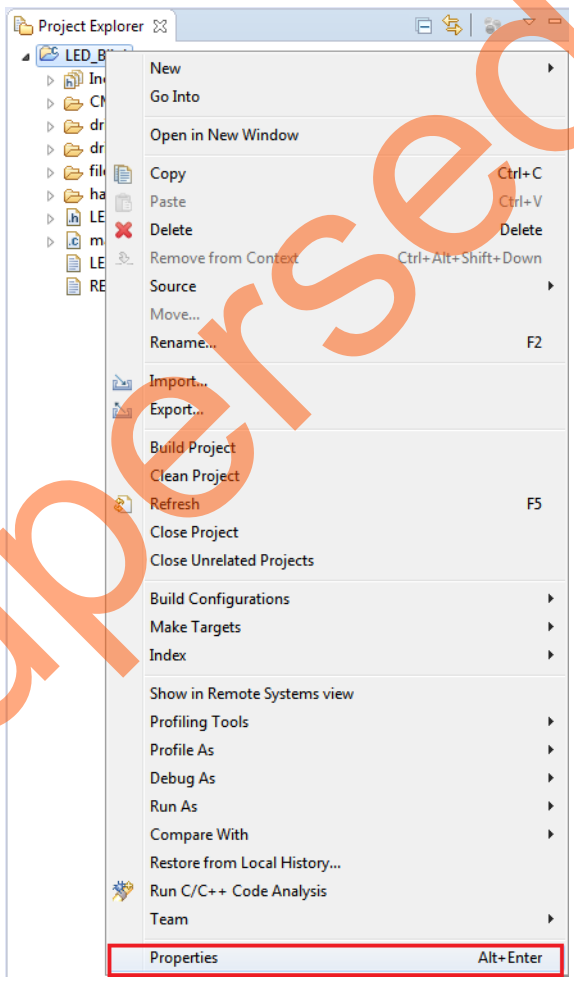


***Figure 33* • Project Explorer window - Properties**

16. click **Settings** under the **C/C++ Build** tab, as shown in Figure 34.

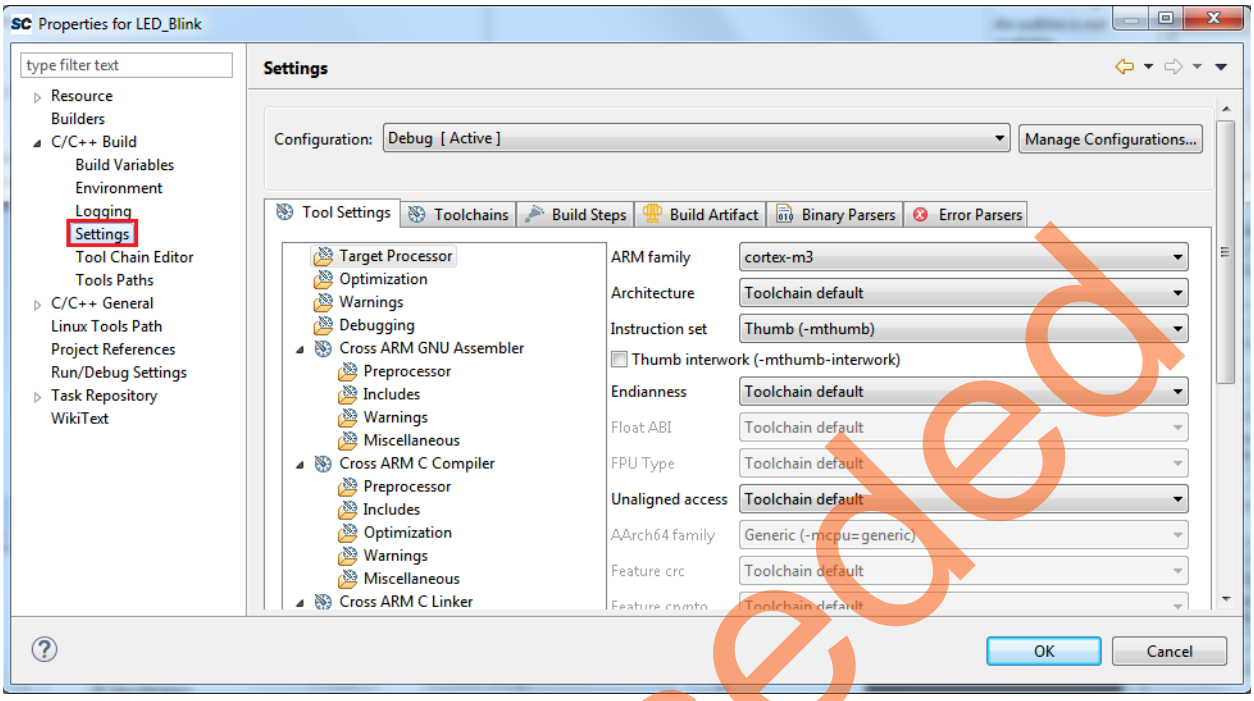


***Figure 34* • Properties for LED_Blink**

17. Under **Cross ARM C compiler**, click **Miscellaneous** and enter **--specs=cmsis.specs**, in **Other compiler flags** text box as shown in Figure 35.

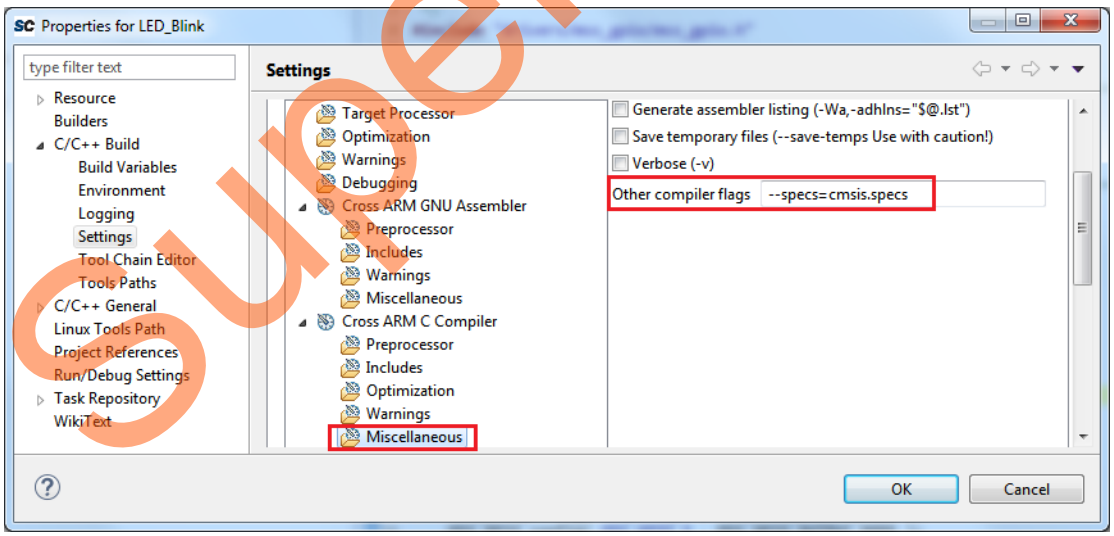


***Figure 35* • Properties for LED_Blink - Miscellaneous**

18. Under **Cross ARM C Linker**, click **General** as shown in Figure 36.
19. Click **add button** and add following linker Script path:
    ../../LED_Blink/CMSIS/startup_gcc/debug-in-microsemi-smartfusion2-esram.ld
    After adding Linker Script, **Properties for LED_Blink** window is displayed, as shown in Figure 36.

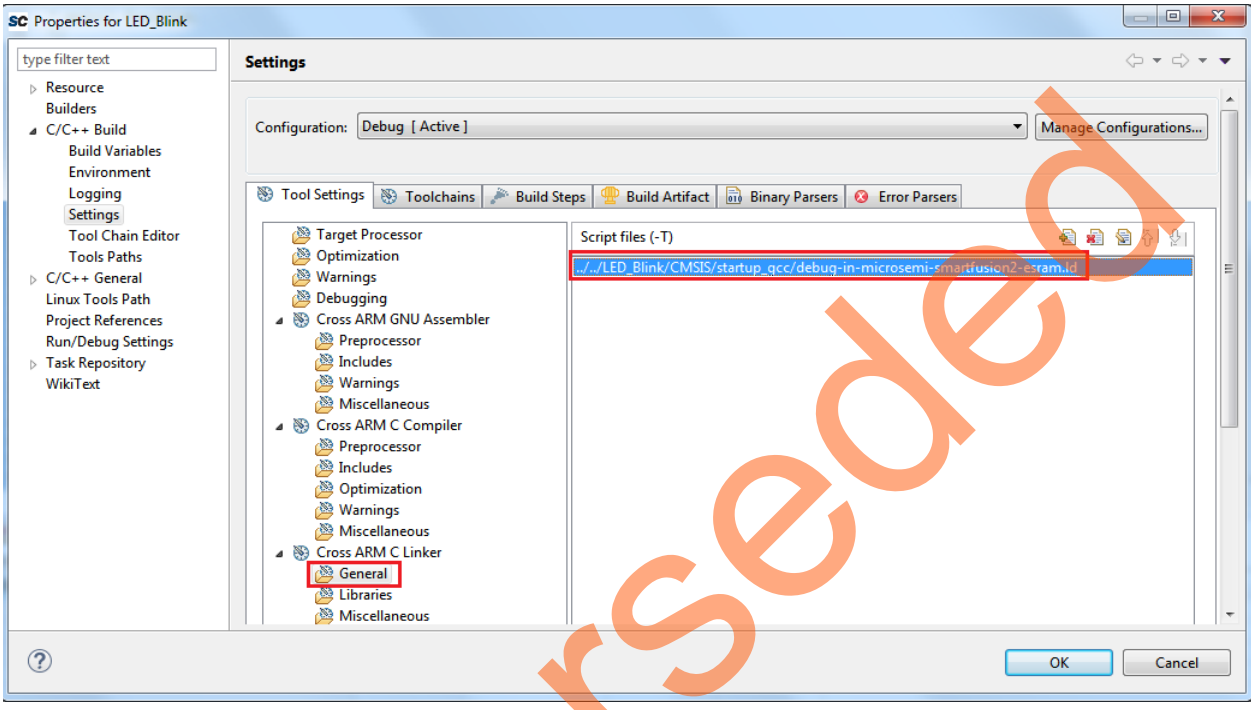


***Figure 36* • Properties for LED_Blink - General**

20. Under **Cross ARM C Linker**, click **Miscellaneous**.
    Check **Use newlib-nano(--specs=nano.specs)** option, as shown in Figure 37.

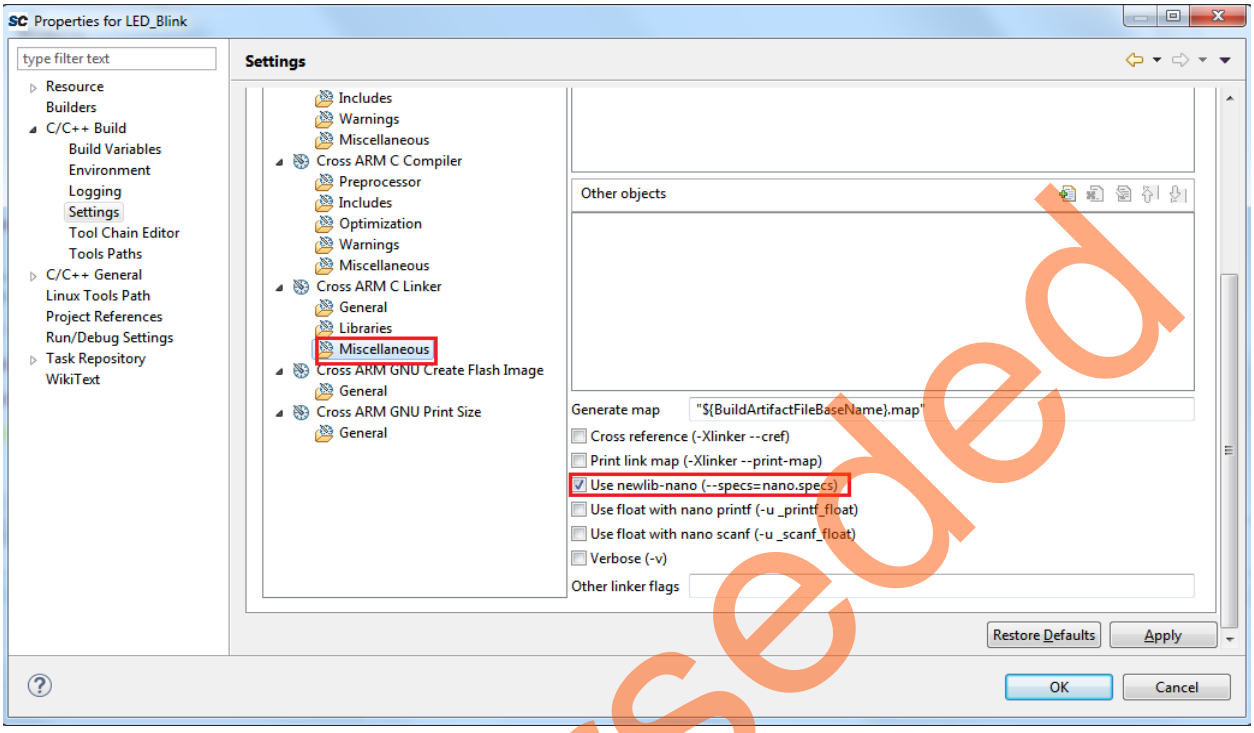


***Figure 37* • Properties for LED_Blink - Miscellaneous**

21. Click **Ok**.
22. Click **Project** and then click **Build All**, as shown in Figure 38.

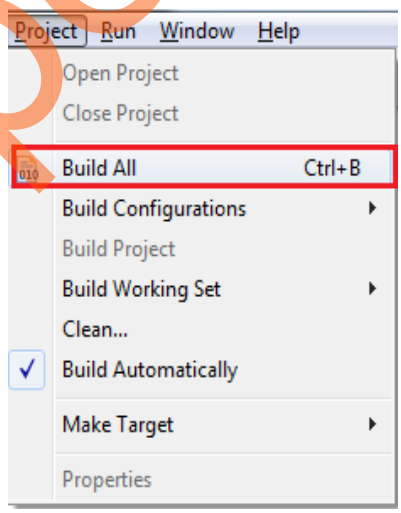


***Figure 38* • Project - Build All**

23. Make sure **Problems** window must not have any errors, as shown in Figure 39.

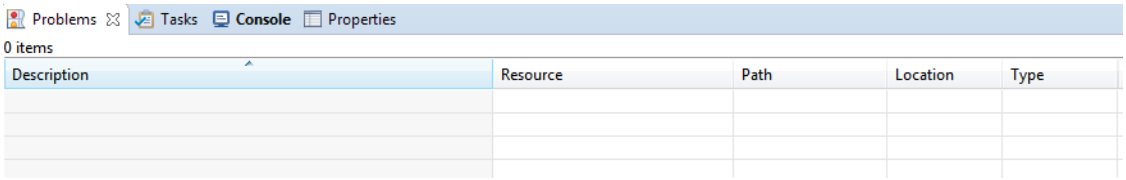


***Figure 39* • Problem Window**

Debugging the Application Project using SoftConsole 4.0

The following steps describe how to debug the application project using SoftConsole 4.0:

1. Click **Debug Configurations** in the **Run** menu of the SoftConsole, as shown in Figure 40. The **Debug Configurations** window is displayed as shown in Figure 41 on page 30.

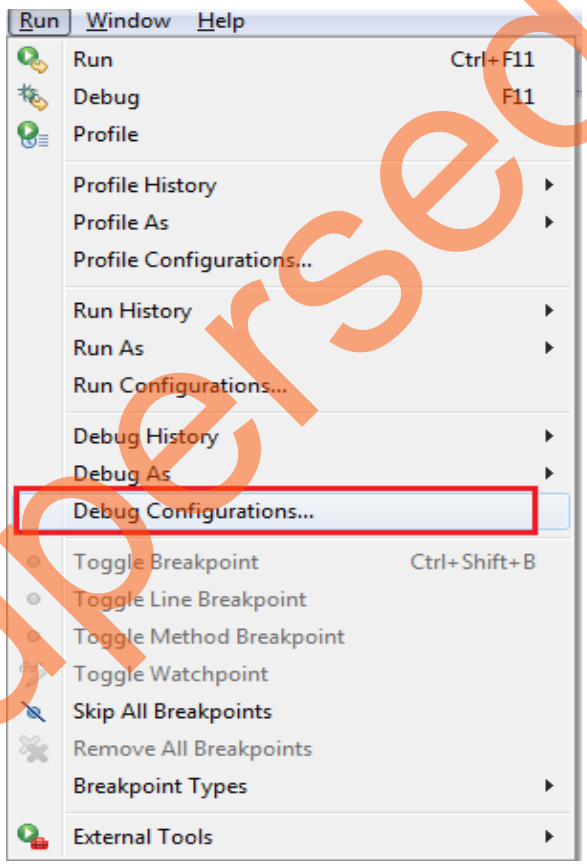


***Figure 40* • Run - Debug Configurations**

2. Double-click **GDB OpenOCD Debugging** to view the configurations, as shown in Figure 41.

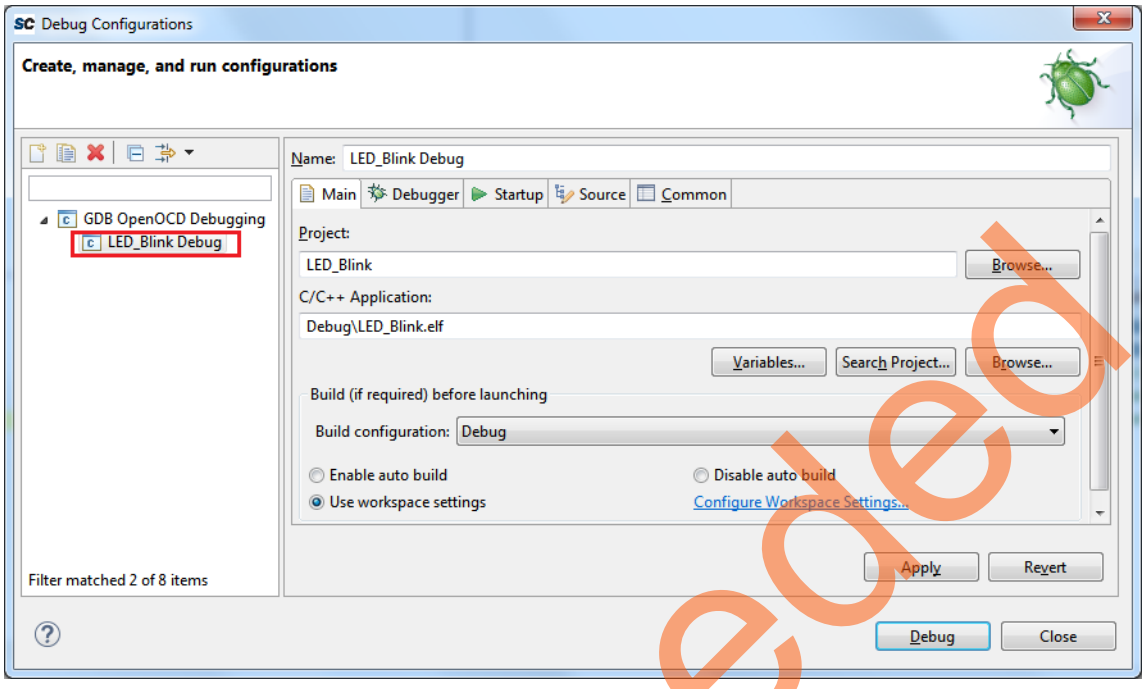


Figure 41 • **Debug Configurations**

3. Ensure that the following values are filled in the corresponding fields:
   - **Name**: LED_Blink Debug
   - **Project**: LED_Blink
   - **C/C++ Application**: Debug\LED_Blink.elf
4. Select the **Debugger** tab in the **Debug Configurations** dialog box.**--command "set DEVICE M2S090"** specifies the target device, as shown in Figure 42 on page 31. This command needs to be modified based on the target silicon.
   - SmartFusion2 Security Evaluation Kit - set DEVICE M2S090
   - SmartFusion2 Advanced Development Kit - set DEVICE M2S150
   - SmartFusion2 Starter Kit - set DEVICE M2S010

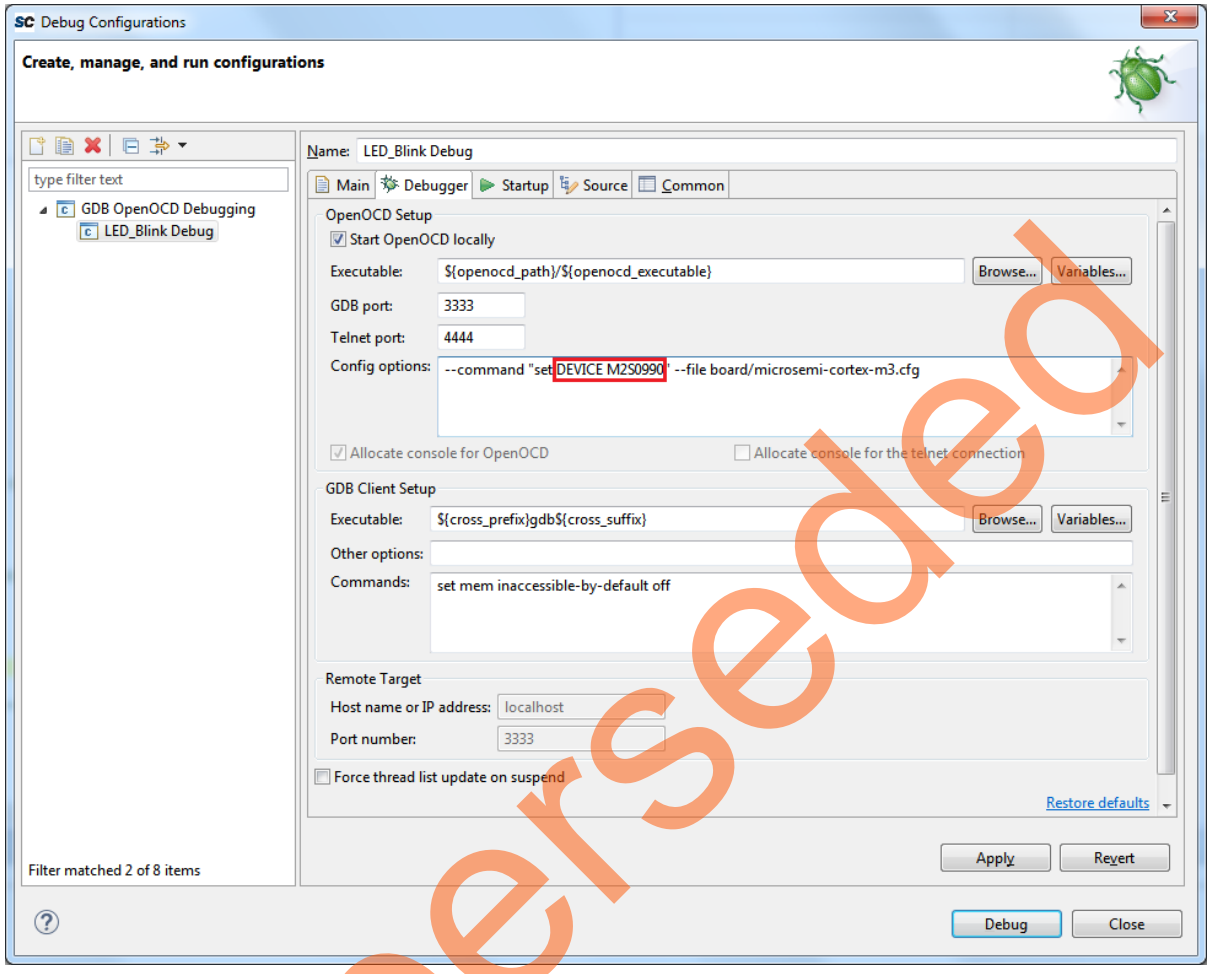


***Figure 42* • Debug Tab**

5. Click **Debug**.
6. On the **Confirm Perspective Switch** window, click **Yes** as shown in Figure 43.

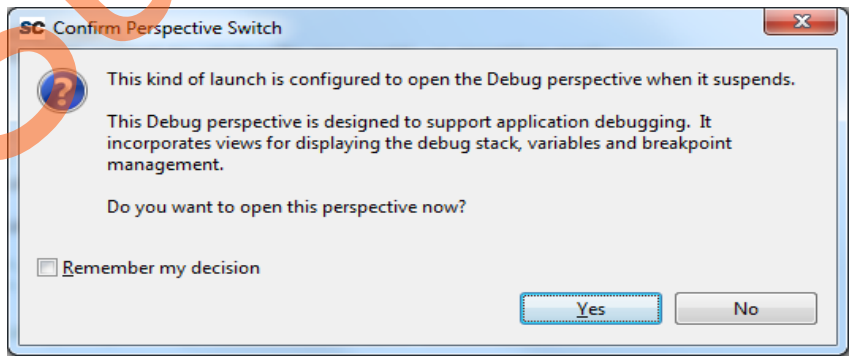


***Figure 43* • Confirm Perspective Switch**

The **SoftConsole Debugger Perspective** window is displayed, as shown in Figure 44.

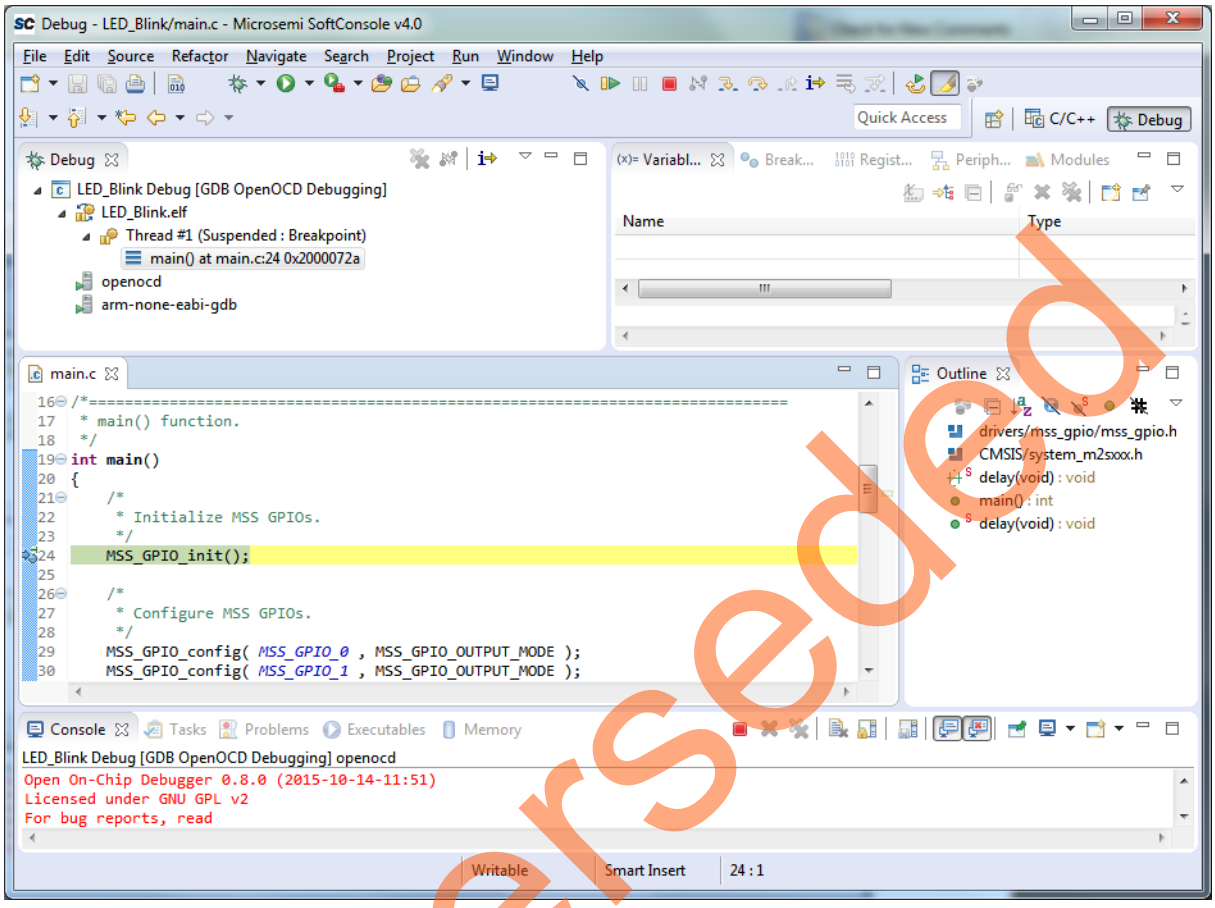


***Figure 44* • Debugger Perspective Window**

7. Click **Run > Resume** to run the application. LED's start blinking on SmartFusion2 target boards. Table 5 shows which LEDs blinks for the different SmartFusion2 target boards.

***Table 5* • LED Target Board**

Target Board	LEDs
SmartFusion2 Security Evaluation Kit	H5, H6, J6, H7, G7, F3, F4, and E1
SmartFusion2 Advanced Development Kit	DS0, DS1, DS2, DS3, DS4, DS5, DS6, and DS7
SmartFusion2 Starter Kit	DS4, DS3

8. Launch the debug session:
   – By selecting **Debug Configurations** from the **Run** menu of SoftConsole.
   or
   – By selecting the **Debug Configurations** using the Debug Button as shown in Figure 45.

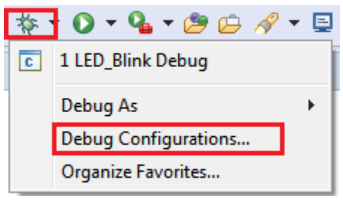


***Figure 45* • Debug Configurations Option**

9. Click the **Registers** tab to view the values of the ARM Cortex-M3 processor internal registers, as shown in Figure 46.

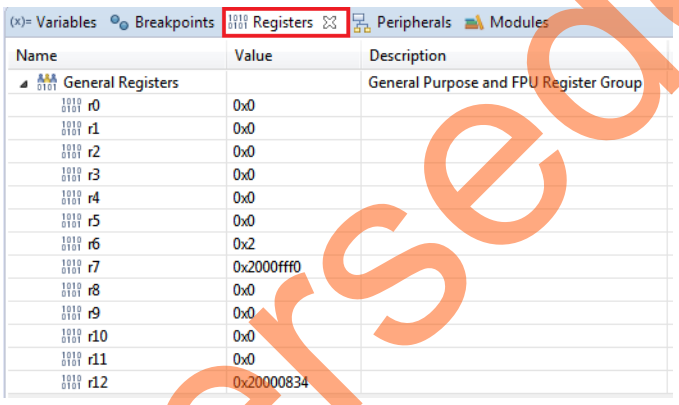


***Figure 46* • Values of Cortex-M3 Internal Registers**

10. Click the **Variables** tab to view the values of variables in the source code, as shown in Figure 47.

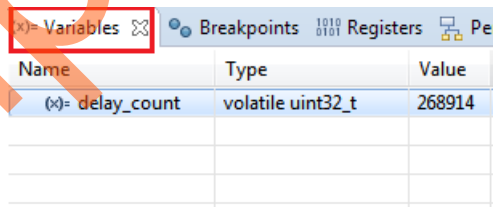


***Figure 47* • Values of the Variables in the Source Code**

11. In the **Debug** window, click **Window > Show View > Disassembly** to display the assembly level instructions. The **Disassembly** window with assembly instructions is displayed on the right-side of the Debug perspective, as shown Figure 48.

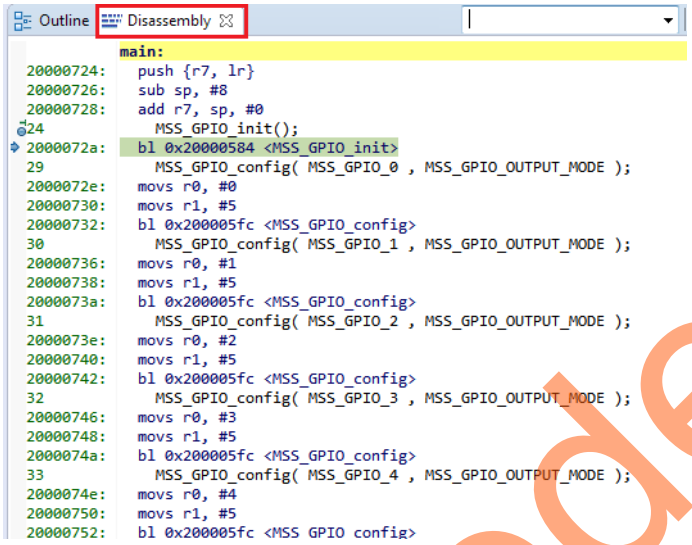


Figure 48 • **Assembly Level Instructions**

12. Source code can be single-stepped by choosing **Run > Step Into or Run > Step Over**. Observe the changes in the source code window and Disassembly view. Performing a Step Over provides an option for stepping over functions. The entire function is run but there is no need to single-step through each instruction contained in the function.
13. Click **Instruction Stepping** ( ) and perform **Step Into** operations. Observe that **Step Into** executes a single line of assembly code.
14. Click **Instruction Stepping** to exit the instruction stepping mode. Single-step through the application and observe the instruction sequence in the source code window of the Debug perspective, and the values of the variables and registers.
15. Add breakpoints in the application to force the code to halt, then single-step and observe the instruction sequence.
16. When debug process is finished, terminate execution of the code by choosing **Run > Terminate**.
17. Close Debug Perspective by selecting **Close Perspective** from the **Window** menu.
18. Close SoftConsole using **File > Exit**.
19. Close the HyperTerminal using **File > Exit**.

Conclusion

This tutorial provides steps to create a Libero SoC design using System Builder. It describes the procedure to build, debug, and run a SoftConsole application. It also provides a simple design to blink LEDs.

Appendix A: Board Setup for SmartFusion2 Security Evaluation Kit

Figure 1 shows the board setup for running the tutorial on the SmartFusion2 Security Evaluation Kit board.

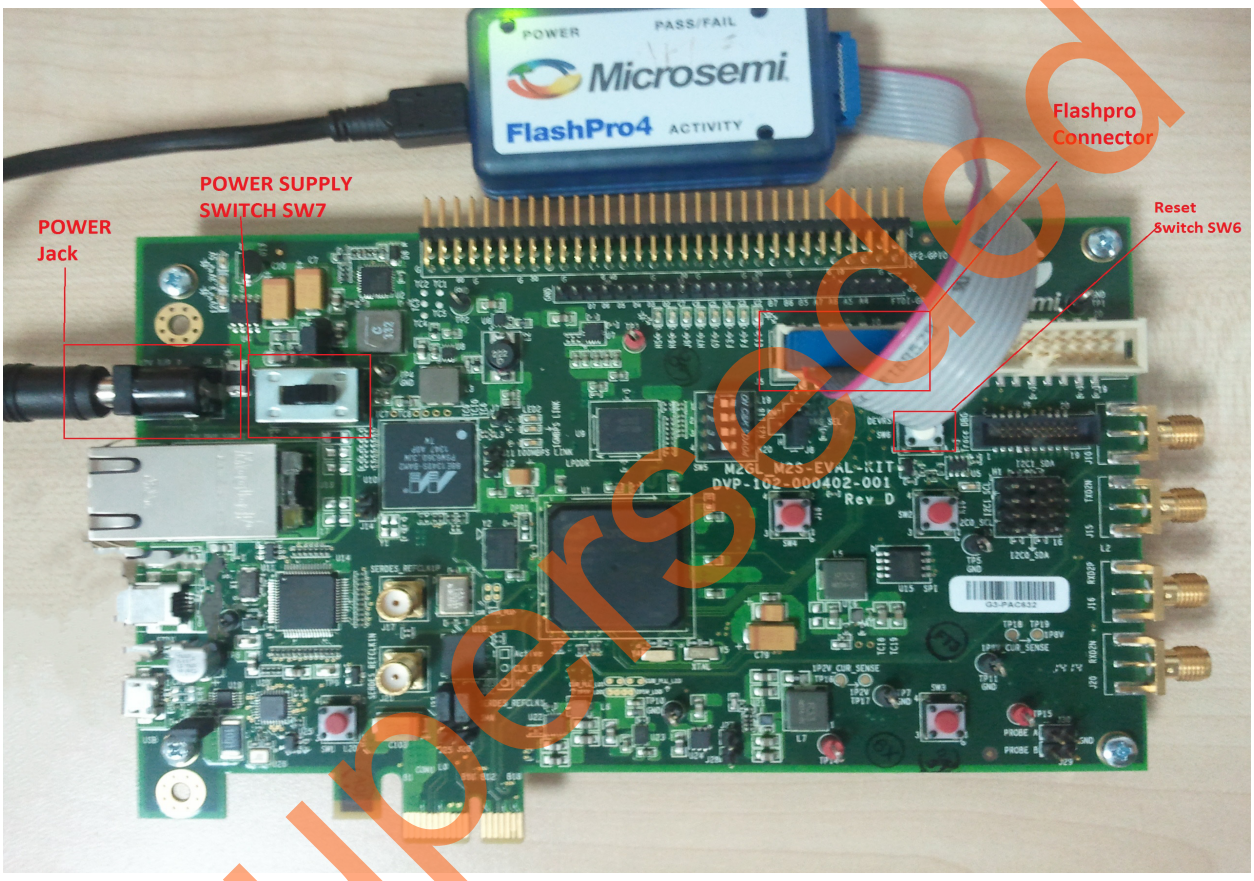


Figure 1 • **SmartFusion2 Security Evaluation Kit Setup**

1. Connect the jumpers on the SmartFusion2 Security Evaluation Kit board as listed in Table 1. For more information on jumper locations, refer Figure 2 on page 36 for SmartFusion2 Security Evaluation Kit board Jumper Locations.

   CAUTION: While making the jumper connections, the **SW7** power supply switch on the board must be in **OFF** position.

Table 1 • **SmartFusion2 Security Evaluation Kit Jumper Settings**

Jumper	Pin (from)	Pin (to)	Comments
J22, J23, J8, and J3	1	2	These are the default jumper settings of the SmartFusion2 Security Evaluation Kit board. Ensure, these jumpers are set accordingly.

2. Connect the FlashPro4 or FlashPro5 programmer to the **J5** connector of the SmartFusion2 Security Evaluation Kit.
3. Connect the power supply to the **J6** connector.
4. Switch **ON** the **SW7** power supply switch.

Figure 2 shows the jumper locations on the SmartFusion2 Security Evaluation Kit board.

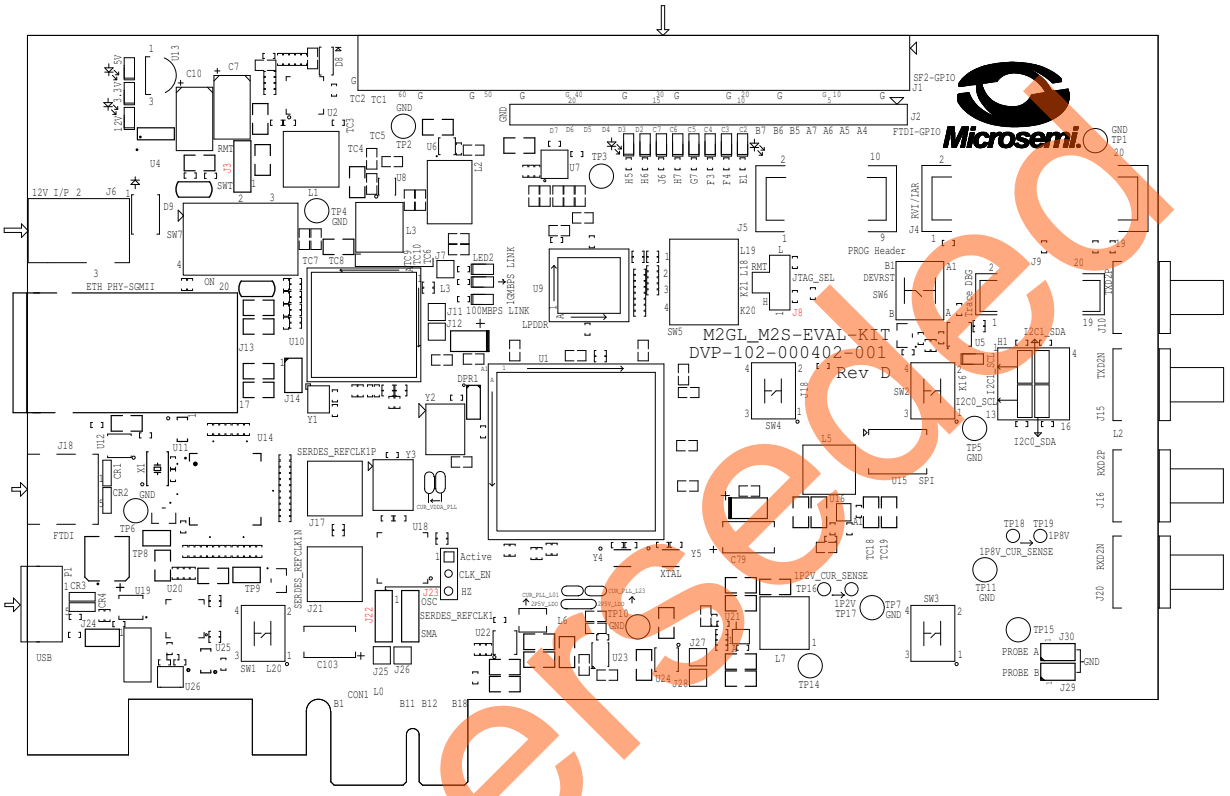


Figure 2 • **SmartFusion2 Security Evaluation Kit Board Jumper Locations**

Notes:

- Jumpers highlighted in red are set by default.
- The locations of the jumpers in Figure 2 are searchable.

Appendix B: Board Setup for SmartFusion2 Advanced Development Kit

Figure 1 shows the board setup for running the demo on the SmartFusion2 Advanced Development Kit board.

Figure 1 • **SmartFusion2 Advanced Development Kit Setup**

1. Connect the jumpers on the SmartFusion2 Advanced Development Kit board as listed in Table 1. For more information on jumper locations, refer Figure 2 on page 38 of SmartFusion2 Advanced Development Kit board Jumper Locations.

   CAUTION: While making the jumper connections, the **SW7** power supply switch on the board must be in **OFF** position.

Table 1 • **SmartFusion2 Advanced Development Kit Jumper Settings**

Jumper	Pin (from)	Pin (to)	Comments
J116, J353, J354, and J54	1	2	These are the default jumper settings of SmartFusion2 Advanced Development Kit board. Ensure, these jumpers are set accordingly.

2. Connect the FlashPro4 or FlashPro5 programmer to the **J37** connector of the SmartFusion2 Advanced Development Kit.
3. Connect the power supply to the **J42** connector.
4. Switch **ON** the **SW7** power supply switch.

Figure 2 shows the jumper locations on the SmartFusion2 Advanced Development Kit board.

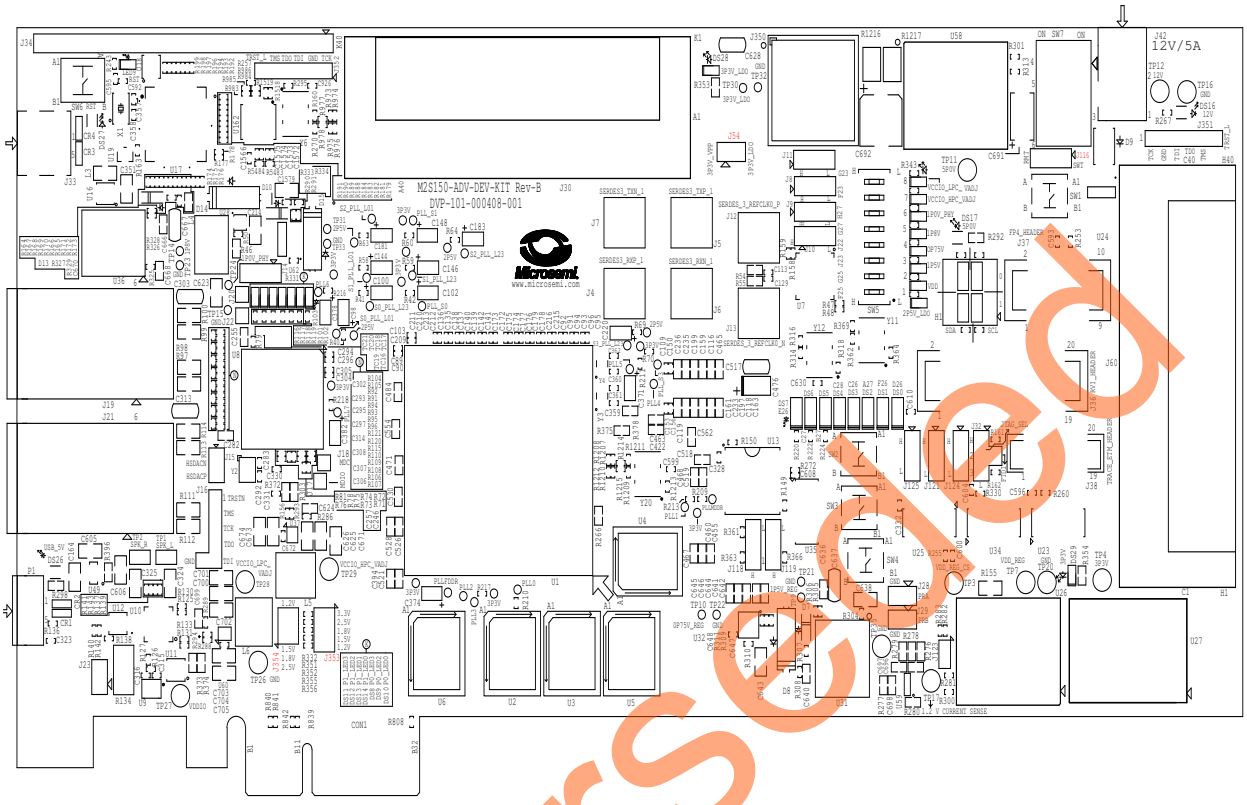


Figure 2 • **SmartFusion2 Advanced Development Kit Board Jumper Locations**

Notes:

- Jumpers highlighted in red are set by default.
- The locations of the jumpers in Figure 2 are searchable.

Appendix C: Board Setup for SmartFusion2 Starter Kit

Figure 1 shows the board setup for running the demo on the SmartFusion2 starter Kit board.

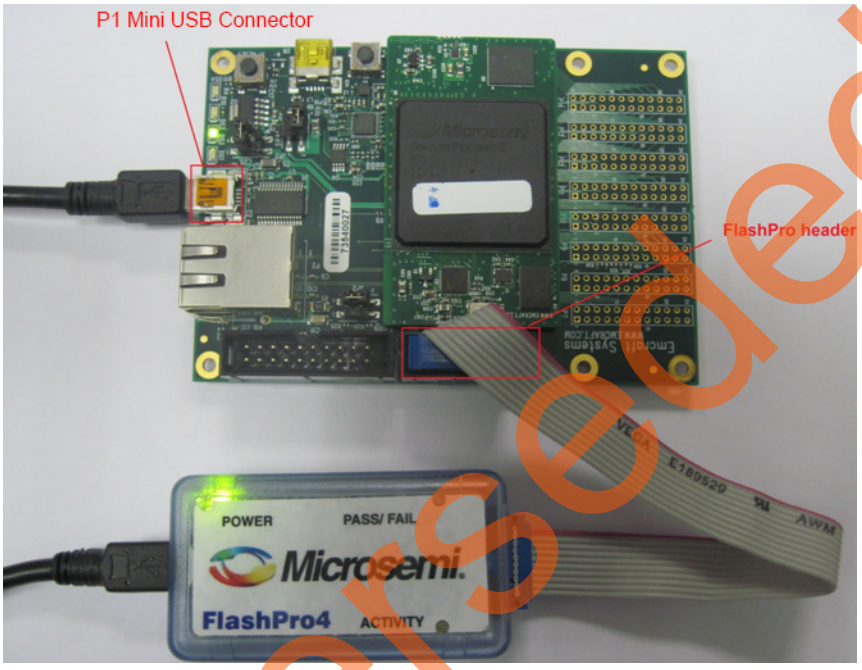


Figure 1 • **SmartFusion2 Starter Kit Setup**

1. Connect the jumpers on the SmartFusion2 Starter Kit board as listed in below table.

Table 1 • **SmartFusion2 Starter Kit Jumper Settings**

Jumper	Pin (from)	Pin (to)	Comments
JP1	1	2	These are the default jumper settings of SmartFusion2 Starter Kit board. Ensure, these jumpers are set accordingly.
JP2	3	4	
JP3	2	4	

2. Connect the FlashPro4 or FlashPro5 programmer to the **P5** connector of the SmartFusion2 Starter Kit.
3. Connect the host PC USB port to the P1 Mini USB connector on the SmartFusion2 Starter Kit board using the USB Mini-B cable. As soon as the connection to the PC is made, the on-board LED DS2 will illuminate, indicating that the board has power.

List of Changes

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

Revision	Changes	Page
Revision 4 (December 2015)	Updated the document for Libero SoC v11.6 and SoftConsole v4.0 software release (SAR 72814)	N/A
Revision 3 (March 2015)	Updated the document for Libero SoC v11.5 software release (SAR 64190).	N/A
Revision 2 (October 2014)	Updated the document for Libero SoC v11.4 software release (SAR 61627).	N/A
Revision 1 (April 2014)	Initial release.	N/A

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.

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

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.

Technical Support

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

Website

You can browse a variety of technical and non-technical information on the SoC home page, at http://www.microsemi.com/products/fpga-soc/fpga-and-soc.

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.

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.

My Cases

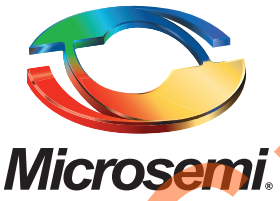
Microsemi SoC Products Group customers may submit and track technical cases online by going to My Cases.

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.

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; 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 3,600 employees globally. Learn more at **www.microsemi.com**.

50200546-4/12.15