Implementing Auto Update and Programming Recovery Features (Using Ethernet Interface) for SmartFusion2 Devices - Libero SoC v11.7

DG0636 Demo Guide

Microsemi
Power Matters.™

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
www.microsemi.com

About Microsemi

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center 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.

50200636-2/7.16

2 Revision History

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

Revision	Changes
Revision 2 (July 2016)	Updated the document for Libero v11.7SP1 software release (SAR 80381). Updated the document for Libero v11.7 software release (SAR 77891). Updated the document with MSS SPI0 enhancement to support Auto Update while sharing SPI0 pins with SPI Controller in Fabric (SAR 74734).
Revision 1 (September 2016)	Initial release.

Contents

Figures

Tables

1 Preface

1.1 Purpose

This demo is for SmartFusion®2 system-on-chip (SoC) field programmable gate array (FPGA) device. It provides instructions on how to use the corresponding reference design.

This document contains the following sections:

- Implementing Auto Update and Programming Recovery Features (Using Ethernet Interface) for SmartFusion2 Devices

1.2 Intended Audience

This demo guide is intended for:

- FPGA designers
- System-level designers

1.3 References

The following documents are referred in this demo guide.

1.3.1 Microsemi Publications

UG0594- M2S090TS-EVAL-KIT SmartFusion2 Security Evaluation Kit User Guide

UG0451- IGLOO2 and SmartFusion2 Programming User Guide

UG0331- SmartFusion2 Microcontroller Subsystem User Guide

The following web-page provides a complete and up-to-date listing of SmartFusion2 device documentation:

http://www.microsemi.com/products/fpga-soc/soc-fpga/smartfusion2

2 Implementing Auto Update and Programming Recovery Features (Using Ethernet Interface) for SmartFusion2 Devices

2.1 Introduction

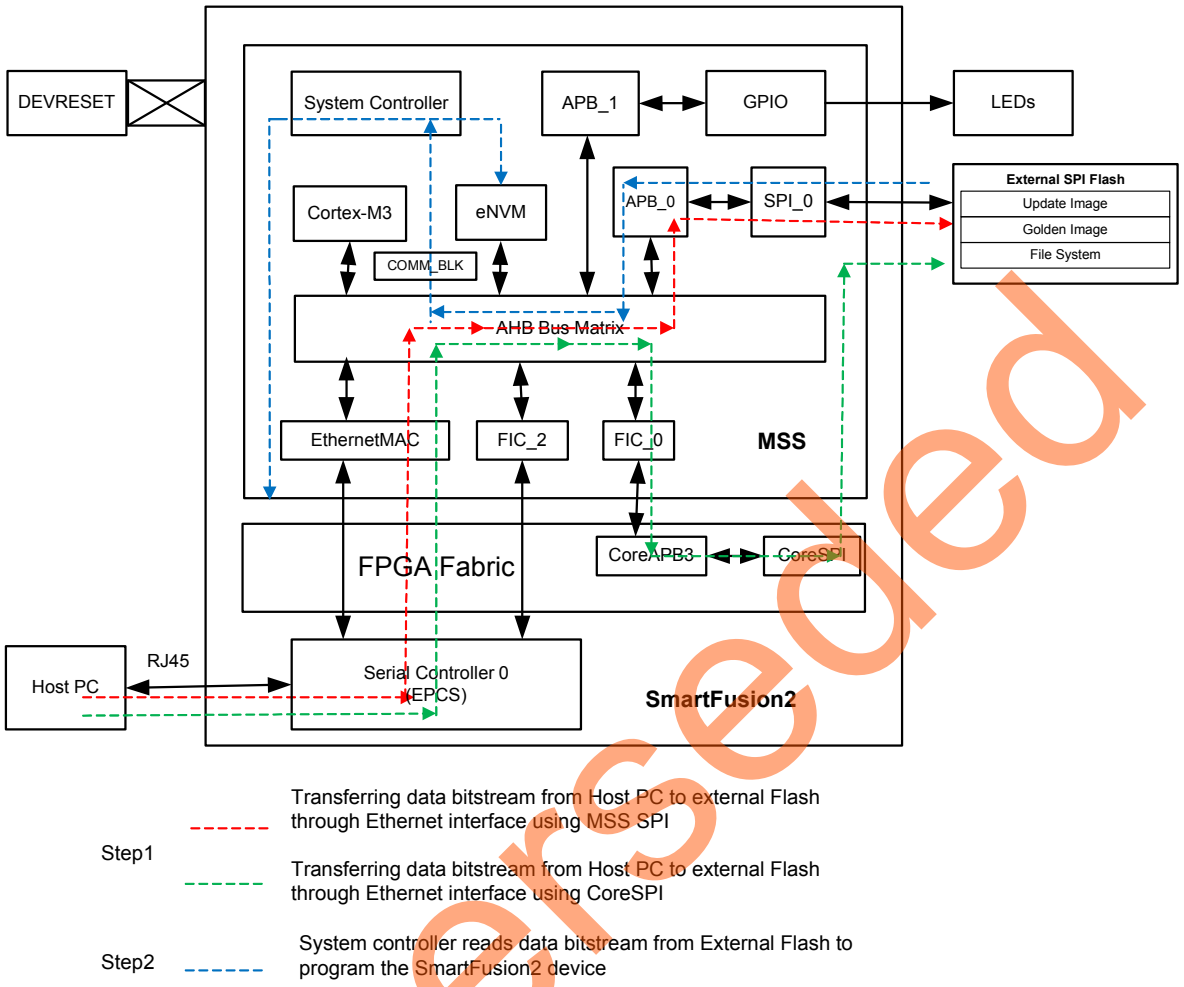
This demo design explains how to implement the Auto Update and Programming Recovery features of the SmartFusion2 device. Auto Update is a programming feature where a pre-programmed device is auto programmed with an Update Image on power-up or assertion of the DEVRST_N pin. The Update Image is typically stored in an on-board external flash memory and its design version is greater than the image design version that is already programmed in the device. Programming Recovery feature allows the device to automatically recover from a power failure during a programming operation. When Programming Recovery option is enabled, the device is programmed with the Golden Image that is stored in the external flash memory.

To transfer the Update and Golden images from Host PC to on-board external SPI flash, you can use different interfaces. For example, SmartFusion2 supported MSS peripherals such as UART, USB, PCIe, and Ethernet interfaces are used to load the images to the SPI flash memory. In this demo TFTP/Ethernet interface is used for loading SPI flash memory. The SPI flash memory can be programmed using the MSS SPI0 or Fabric CoreSPI controller.

The MSS SPI0 controller is configured to share the SPI0 pins with a CoreSPI controller implemented in the FPGA fabric. To share the SPI0 port a multiplexer logic is implemented in the FPGA fabric to switch the SPI0 pins between MSS SPI0 and the Fabric CoreSPI controller.

Figure 1 on page 9 shows the dataflow of the design. The arrows that are highlighted in red show the data flow between the Host PC and on-board external SPI flash memory using the MSS SPI and Ethernet Interfaces. The arrows that are highlighted in green show the data flow between the Host PC and on-board external SPI flash memory using the CoreSPI and Ethernet interfaces. The ARM® Cortex®-M3 processor copies the programming data from the Host PC to the SPI flash using the Ethernet interface. The arrows that are highlighted in blue show the System Controller reading the data from external SPI flash memory to program the SmartFusion2 device. In this demo design, the SPI flash images contain information to program both eNVM and FPGA fabric.

Figure 1 • **SmartFusion2 Demo Design**



For more information about Programming Recovery and Auto Update features of SmartFusion2, refer to the *UG0451- IGLOO2 and SmartFusion2 Programming User Guide*.

For more information about SPI and Ethernet Interfaces, refer to the *UG0331- SmartFusion2 Microcontroller Subsystem User Guide*.

2.2 Design Requirements

Table 1 • **Design Requirements**

Design Requirements	Description
Hardware Requirements	
SmartFusion2 Security Evaluation Kit: • 12 V adapter • FlashPro4 • USB A to Mini-B cable	Rev D or later
RJ45 Cable (not provided with the kit)	
Host PC or Laptop	Windows 64-bit Operating System
Software Requirements	
Libero® System-on-Chip (SoC) for viewing the design files	v11.7 SP1
FlashPro Programming Software	v11.7 SP1
SoftConsole	v3.4 SP1*
Host PC Drivers	USB to UART drivers
One of the following serial terminal emulation programs: • HyperTerminal • TeraTerm • PuTTY	

Note: **For this demo guide, SoftConsole v3.4 SP1 is used. For using SoftConsole v4.0, see the TU0546: SoftConsole v4.0 and Libero SoC v11.7 Tutorial.*

2.3 Demo Design

2.3.1 Introduction

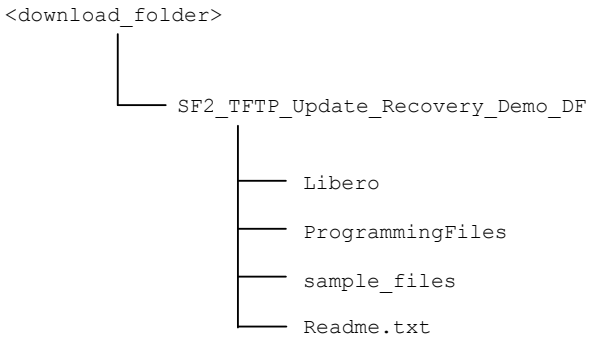
The demo design files are available for download from the following path in the Microsemi® website: http://soc.microsemi.com/download/rsc/?f=m2s_dg0636_liberov11p7sp1_df

The demo design files include:

- Libero
- Sample files
- Programming files
- Readme.txt file

Figure 2 shows the top-level structure of the design files. Refer to `readme.txt` file for the complete directory structure.

Figure 2 • **Directory Structure**



2.3.2 Demo Design Features Summary

The following are the demo design features:

- Program Recovery is enabled and to demonstrate this feature, only Golden Image is loaded into the SPI flash memory. When a power failure scenario is created, the demo design programs the device with the Golden Image after a power failure during a programming operation.
- Auto Update is enabled and both Golden and Update images (with a higher version) are loaded into the flash memory. When the device is reset, the demo design programs the device with a higher design version image, that is, Update Image to demonstrate Auto Update feature.
- Transfer of programming files from the Host PC to an external SPI flash is done through Ethernet Interface using the TFTP application.
- External SPI flash can be programmed using MSS SPI0 or CoreSPI controller in fabric.
- Both eNVM and FPGA Fabric are programmed during Programming Recovery and Auto Update.
- LED blinking patterns and Serial Terminal messages are different for Golden and Update images to indicate Programming Recovery and Auto Update are applied correctly.

2.3.3 Demo Design Description

The demo design comprises of the following features:

- Programming Recovery
- Auto Update
- TFTP Server Application

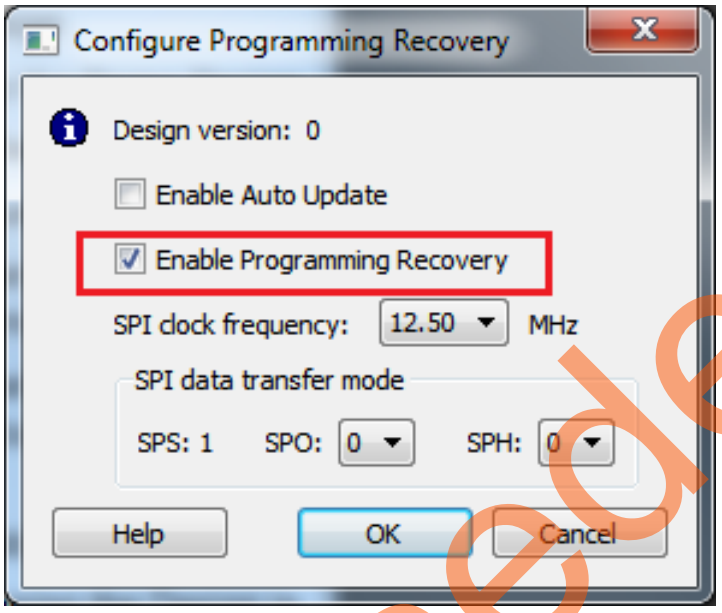
2.3.3.1 Programming Recovery

Programming recovery, if enabled, automatically recovers from a power failure during a programming operation. The Programming Recovery option is enabled using the Libero SoC software. To enable the Programming Recovery feature, open the Libero Project and select **Configure Programming Recovery** option in **Design Flow** window.

Programming recovery can be enabled in two ways:

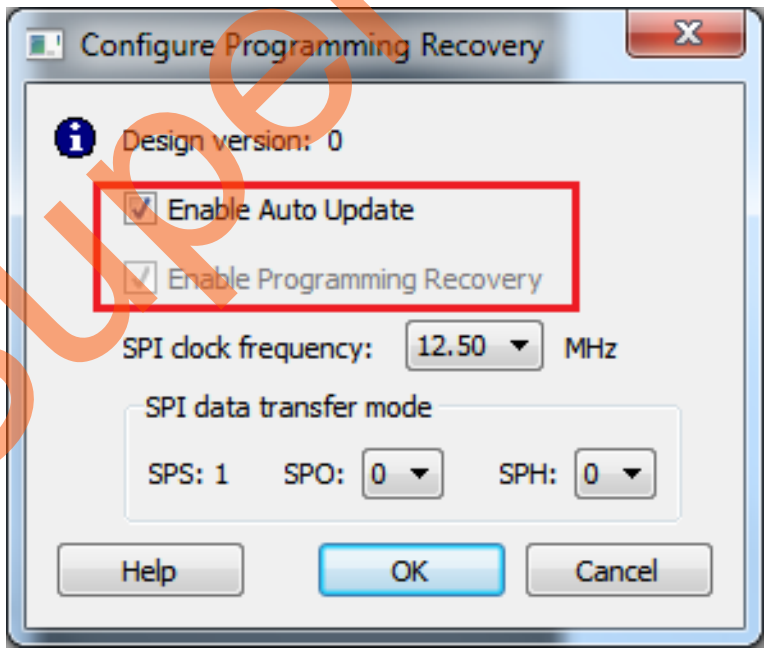
1. In the **Configure Program Recovery** window, select **Enable Programming Recovery** check box as shown in Figure 3.

Figure 3 • **Configure Programming Recovery - Enable Programming Recovery**



2. In **Configure Program Recovery** window, enabling Auto Update option enables the Programming Recovery as shown in Figure 4, this configuration is used in this demo.

Figure 4 • **Configure Programming Recovery - Enable Auto Update**



Based on the content provided in the Recovery programming file, three types of programming are possible:

- **eNVM programming:** This file has only eNVM content.
- **FPGA Fabric programming:** This file has only the FPGA fabric content.
- **eNVM and FPGA Fabric programming:** This file has both the FPGA fabric and eNVM content.

Program Recovery Image provided with Design files in this demo has both eNVM and Fabric content.

The Programming Recovery option requires an external SPI flash to be connected to MSS SPI_0. External SPI flash needs to be loaded with a SPI directory, Golden Image and Update Image. The SPI directory provides the address of the Golden or Update images and its design versions as shown in Table 2.

Note: When only Programming Recovery option is enabled. Update Image address and version are not required while creating SPI directory. This demo demonstrates both Programming Recovery and Auto Update features.

Table 2 • **SPI Flash Directory**

Offset	Name	Size (Bytes)	Description
0	GOLDEN_IMAGE_ADDRESS	[3:0]	Address where the golden image starts.
4	GOLDEN_IMAGE_DESIGNVER	[1:0]	Design version of the golden image.
6	UPDATE_IMAGE_ADDRESS	[3:0]	Address where the update image starts.
10	UPDATE_IMAGE_DESIGNVER	[1:0]	Design version of the update image.

Create a directory in the SPI flash to provide the address for Auto Update Image, Golden Image, and their design versions. On the SmartFusion2 Security Evaluation Kit board, the flash memory size is 8 MB. Depending on the size of the programming file, the SPI flash memory is configured in the SoftConsole Project as follows:

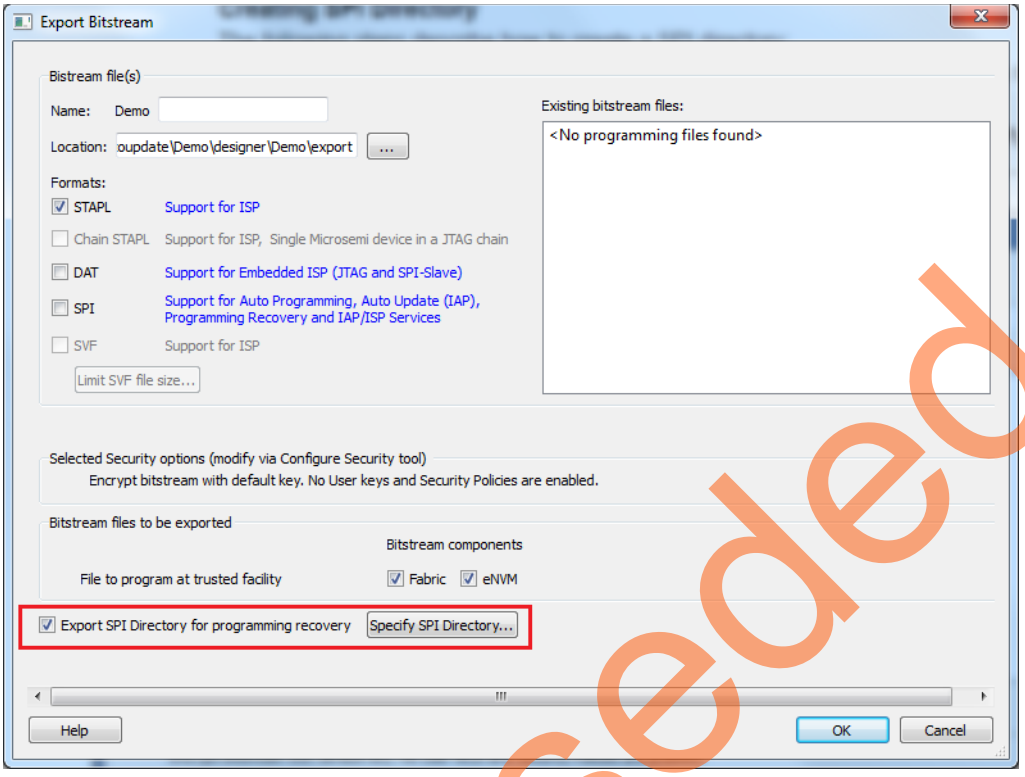
- 0x0 - 0xFFF is for loading SPI Directory
- 0x1000 - 0x3FFFFF is for loading Golden Image
- 0x400000 - 0x7FFFFF is for loading Auto Update Image

2.3.3.1.1 Creating SPI Directory

The following steps describe how to create a SPI directory:

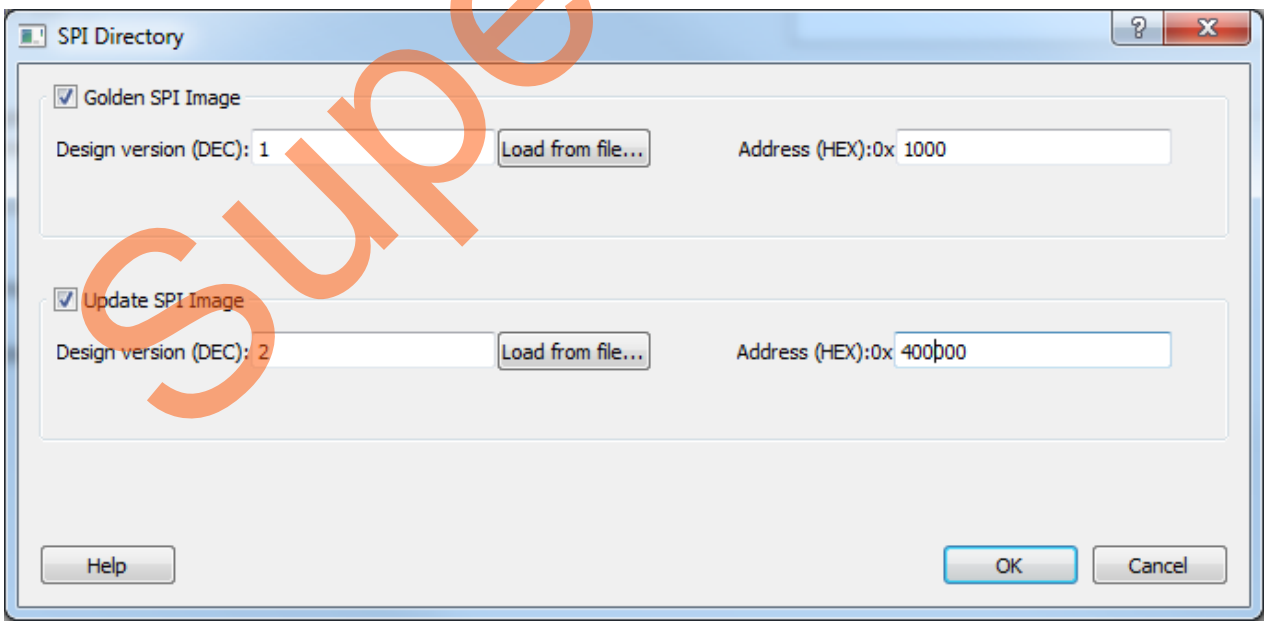
1. To create SPI directory, open Libero project and select **Export Bit Stream** from **Design Flow** window. **Export Bitstream** window is displayed as shown in Figure 5.
2. Select **Export SPI Directory for programming recovery** and click **Specify SPI Directory**.

Figure 5 • **Export Bitstream**



The **SPI Directory** window is displayed as shown in Figure 6. Enter the design version and address for Golden Image and Update Image. Golden and Update image addresses, which are configured in the SoftConsole Project need to be entered in the SPI Directory window.

Figure 6 • **SPI Directory**



Design version size is 16-bit. Address indicates the starting address of Golden and Update images and address field size is 32-bit. Using the TFTP/Ethernet SPI Directory, Golden Image, and Auto Update

images are transferred to SPI addresses 0x0, 0x1000 and 0x400000 respectively. Golden and Update images are provided with the file extension *.spi*.

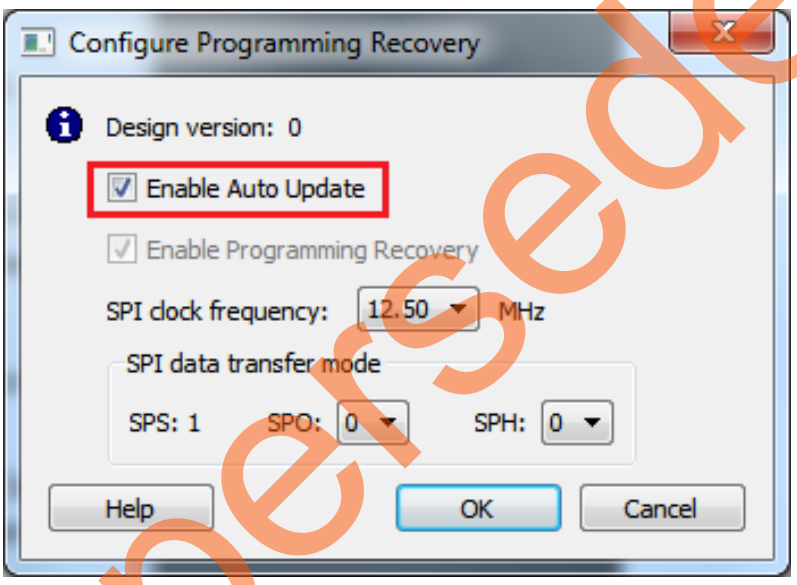
The System Controller reads the Golden Image address from SPI directory and programs the Golden Image available at 0x1000 in case of power failure during programming operation.

2.3.3.2 Auto Update

When the Auto Update feature is enabled, the device is auto programmed with an Update Image on power-up or assertion of the DEVRST_N pin. This applies when the Update Image design version (that is, stored in SPI flash) is greater than the image design version that is already programmed in the device. Auto Update feature is enabled using Libero SoC software. To enable the Auto Update feature, open the provided Libero Project and select **Configure Programming Recovery** option in **Design Flow** window, select **Enable Auto Update** in Configure Programming Recovery as shown in Figure 7 on page 15.

Enabling the Auto Update enables the Programming Recovery by default. During the Auto Update, in case of power failure the Programming Recovery is run automatically.

Figure 7 • **Configure Programming Recovery - Enable Auto Update**



Based on the inputs provided by the Auto Update programming file, three types of programming are possible:

- **eNVM programming:** This file has only eNVM content.
- **FPGA Fabric programming:** This file has only the FPGA fabric content.
- **eNVM and FPGA Fabric programming:** This file has both the FPGA fabric and eNVM content.

Auto Update Image provided with design files in this demo has both eNVM and Fabric content.

The Auto Update option requires an external SPI flash to be connected to MSS SPI_0. The External SPI flash needs to be loaded with SPI directory, Golden Image and Update Image. The SPI directory provides the address for Golden Image, Update Image and their design versions. For more information, refer to Table 2 on page 13.

If the flash memory has enough memory, then it can have more than one Auto Update Image. To update from different Auto Update images, the user needs to create a new SPI directory with new Auto Update Image Address and its Design Version.

In this demo, Auto Update Image is available at 0x400000 address. The same address must be entered when creating the SPI directory as shown in Figure 6 on page 14.

To update from different Auto Update Images, new Update Image address and version must be entered in the GUI when creating the SPI directory. This new SPI directory needs to be programmed to the SPI flash.

For more information about how to create SPI directory, refer to "Creating SPI Directory" section.

On power-up or assertion of the DEVRST_N pin, the System Controller reads the image version in the SPI directory and programs the Update Image available at 0x400000, if it contains the higher image version than the image already programmed in the FPGA.

2.3.3.3 TFTP Server Application

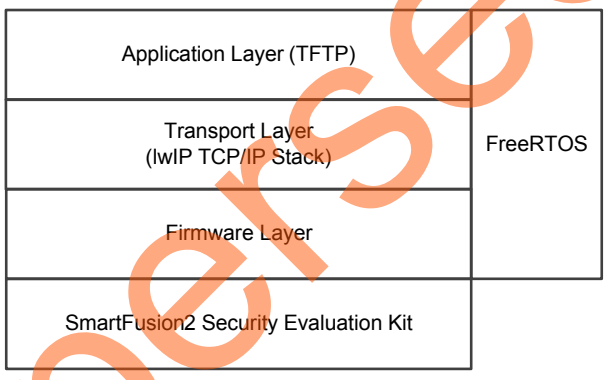
To transfer the Programming Images (.spi format) and load them to the SPI flash memory, the Trivial File Transfer Protocol (TFTP) is used. The TFTP server application is implemented in the firmware project to transfer the SPI images from the Host PC to the external flash memory (available on SmartFusion2 Security Evaluation Kit board).

TFTP server application has following layers:

- Application Layer
- Transport Layer
- Firmware Layer

Figure 8 shows the block diagram of the TFTP server application on SmartFusion2 device used in this demo design.

Figure 8 • **TFTP Server Application - Block Diagram**



2.3.3.3.1 Application Layer

The TFTP protocol is implemented in the application layer. TFTP is used to transfer the files between client and server. A file transfer is initiated by the client issuing a request to read or write a particular file on the server.

The TFTP client (Host PC) transfers the file using TFTP PUT command to the SmartFusion2 device (TFTP server). Transferred files are stored in the external flash memory connected to the MSS SPI_0 on SmartFusion2 Security Evaluation Kit board.

2.3.3.3.2 Transport Layer (lwIP TCP/IP Stack)

The lwIP stack is suitable for the embedded systems because of its less resource usage. It is used with or without the operating system. The lwIP consists of the actual implementations of the IP, ICMP, UDP, and TCP protocols, as well as the support functions such as buffer and memory management.

For more information on the design and implementation, refer to www.sics.se/~adam/lwip/doc/lwip.pdf.

2.3.3.3.3 RTOS and Firmware Layer

FreeRTOS is an open source real time operating system kernel. FreeRTOS is used in this demo to prioritize and schedule the tasks. For more information about FreeRTOS and the latest source code, refer to http://www.freertos.org.

The firmware provides the software driver implementation to configure and control the following MSS components:

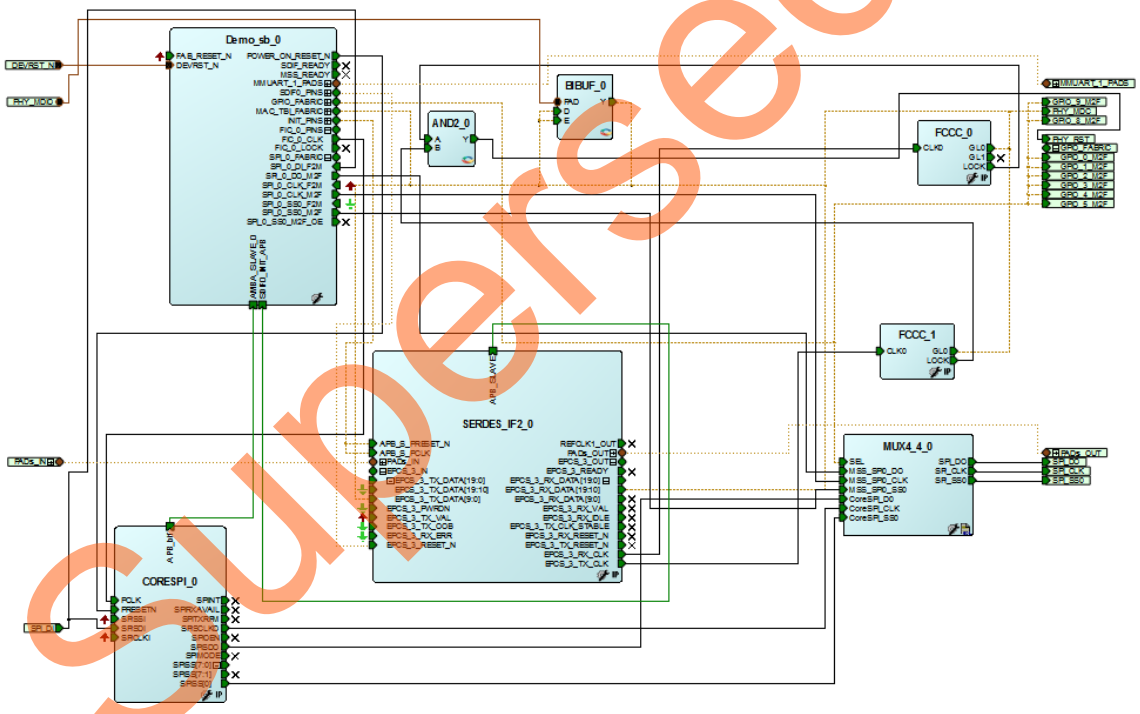
- Ethernet MAC
- MMUART
- GPIO
- SPI
- RTC

2.3.4 Hardware Implementation

In this demo design, the following blocks are configured in Libero hardware project:

- MSS GPIO block is enabled and configured as: GPIO_0 to GPIO_7 as outputs and connected LEDs.
- M3_CLK clock is configured to 100 MHz.
- MSS SPI_0 controller is enabled to access the external SPI flash memory.
- CoreSPI controller in fabric is configured to access the external Flash Memory.
- MMUART1 is enabled for RS-232 communication on the SmartFusion2 Security Evaluation Kit board.
- The MSS TSEMAC is configured for the ten-bit interface (TBI) operation.
- The SERDES_IF_2 in the SmartFusion2 device is configured for EPCS Lane3.

Figure 9 • **Libero SmartDesign**



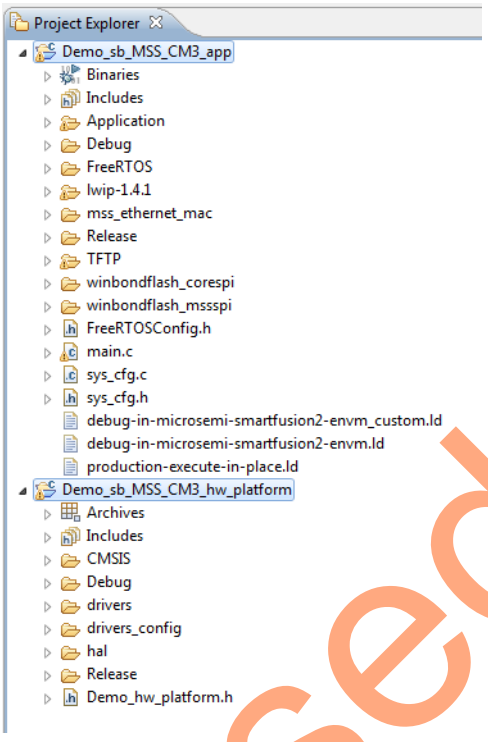
2.3.5 SoftConsole Firmware Project

The following stacks are used for this demo design:

- lwIP TCP/IP stack version 1.4.1
- FreeRTOS

Figure 10 shows SoftConsole software directory structure of the demo design.

Figure 10 • **SoftConsole Project Explorer Window**



The SoftConsole workspace consists of the following projects.

- **Demo_sb_MSS_CM3_app:** Contains TFTP server application using LWIP and FreeRTOS. This SoftConsole project transfers only the SPI directory, Golden and Update Images to SPI Flash memory using Ethernet Interface.
- **Demo_sb_MSS_CM3_hw_platform:** Contains all the firmware and hardware abstraction layers that correspond to the hardware design. This project is configured as a library and is referenced by Demo_sb_MSS_CM3_app project.

Note: External SPI flash can be programmed using MSS SPI0 or Fabric CoreSPI controller. By default CoreSPI is configured in SoftConsole project provided in the design files. To configure MSS SPI0 to transfer .spi files, see "Appendix: Configuring MSS SPI0 in SoftConsle Project" section on page 42.

2.4 Setting Up the Demo Design

The following steps describe how to setup the hardware demo for the SmartFusion2 Security Evaluation Kit board:

1. Connect the jumpers on the SmartFusion2 Security Evaluation Kit board, as shown in Table 3.

Table 3 shows the jumper settings.

Table 3 • **SmartFusion2 Security Evaluation Kit Jumper Settings**

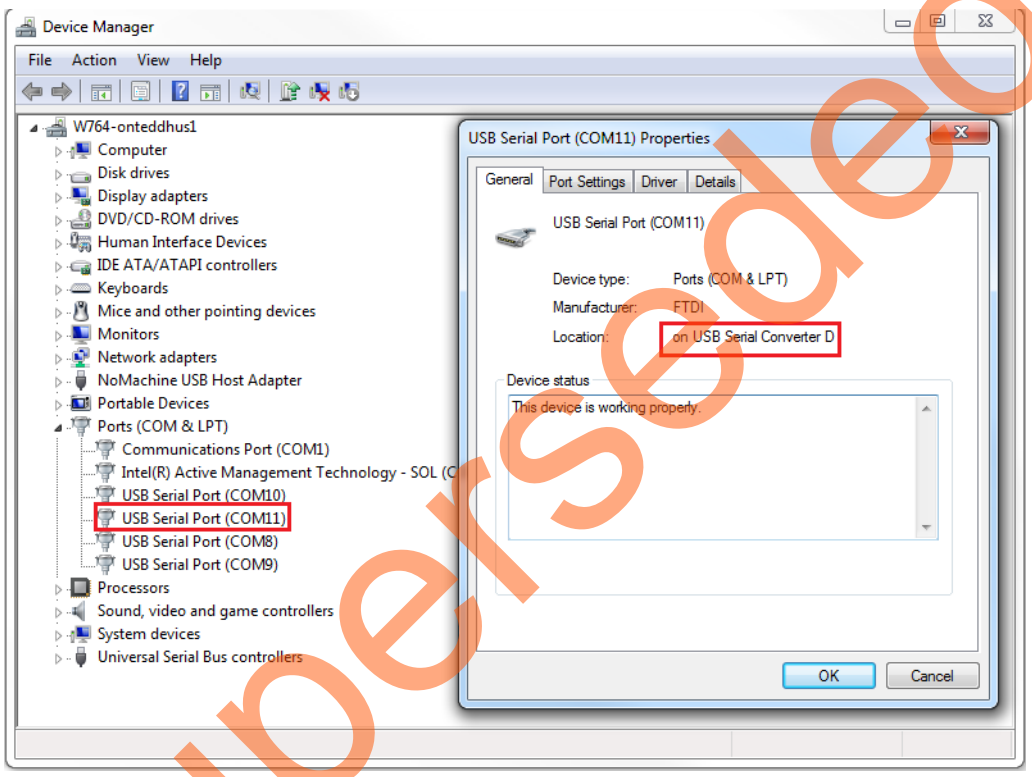
Jumper	Pin (From)	Pin (To)	Comments
J22, J23, J24, J8, J3	1	2	Default These are the default jumper settings of the SmartFusion2 Security Evaluation Kit board. Ensure that these jumpers are set properly.

"Appendix: SmartFusion2 Security Evaluation Kit Board Jumper Locations" section on page 29 provides the SmartFusion2 Security Evaluation kit silk screen to identify the jumper locations on board.

Note: Ensure that the power supply switch, **SW7** is switched OFF while connecting the jumpers on the SmartFusion2 Security Evaluation Kit board.

2. Connect the Host PC to the J18 connector using the USB Mini-B cable. The USB to UART bridge drivers are automatically detected.
3. From the detected four COM ports, right-click any one of the COM ports and select Properties. The selected COM port properties window is displayed, as shown in Figure 11.
4. Ensure to have the Location as **on USB Serial Converter D** in the Properties window as shown in Figure 11.

Figure 11 • **Device Manager Window**



5. Install the USB driver, if the USB drivers are not detected automatically.
6. Install the FTDI D2XX driver for serial terminal communication through the FTDI mini USB cable. Download the drivers and installation guide from: www.microsemi.com/soc/documents/CDM_2.08.24_WHQL_Certified.zip
7. Connect the FlashPro4 programmer to the J5 connector of the SmartFusion2 Security Evaluation Kit board.
8. Connect the power supply to the J6 connector of the SmartFusion2 Security Evaluation Kit.
9. This design example can run in both Static IP and Dynamic IP modes. By default, the programming files are provided for dynamic IP mode.
   - For static IP, connect the Host PC to the J13 connector of the SmartFusion2 Security Evaluation Kit board using an RJ45 cable.
   - For dynamic IP, connect any one of the open network ports to the J13 connector of the SmartFusion2 Security Evaluation Kit board using an RJ45 cable.

2.4.1 Board Setup Snapshot

Snapshots of the SmartFusion2 Security Evaluation Kit board with all the setup made is given in "Appendix: Board Setup for Running the Demo" section on page 28

2.5 Running the Demo Design

The following steps describe how to program the demo design:

10. Download the demo design from:
    http://soc.microsemi.com/download/rsc/?f=m2s_dg0636_liberov11p7sp1_df
1. Switch **ON** the SW7 power supply switch.
2. Start any serial terminal emulation program such as:
    - HyperTerminal
    - PuTTY
    - Tera Term

Note: In this demo PuTTY is used.

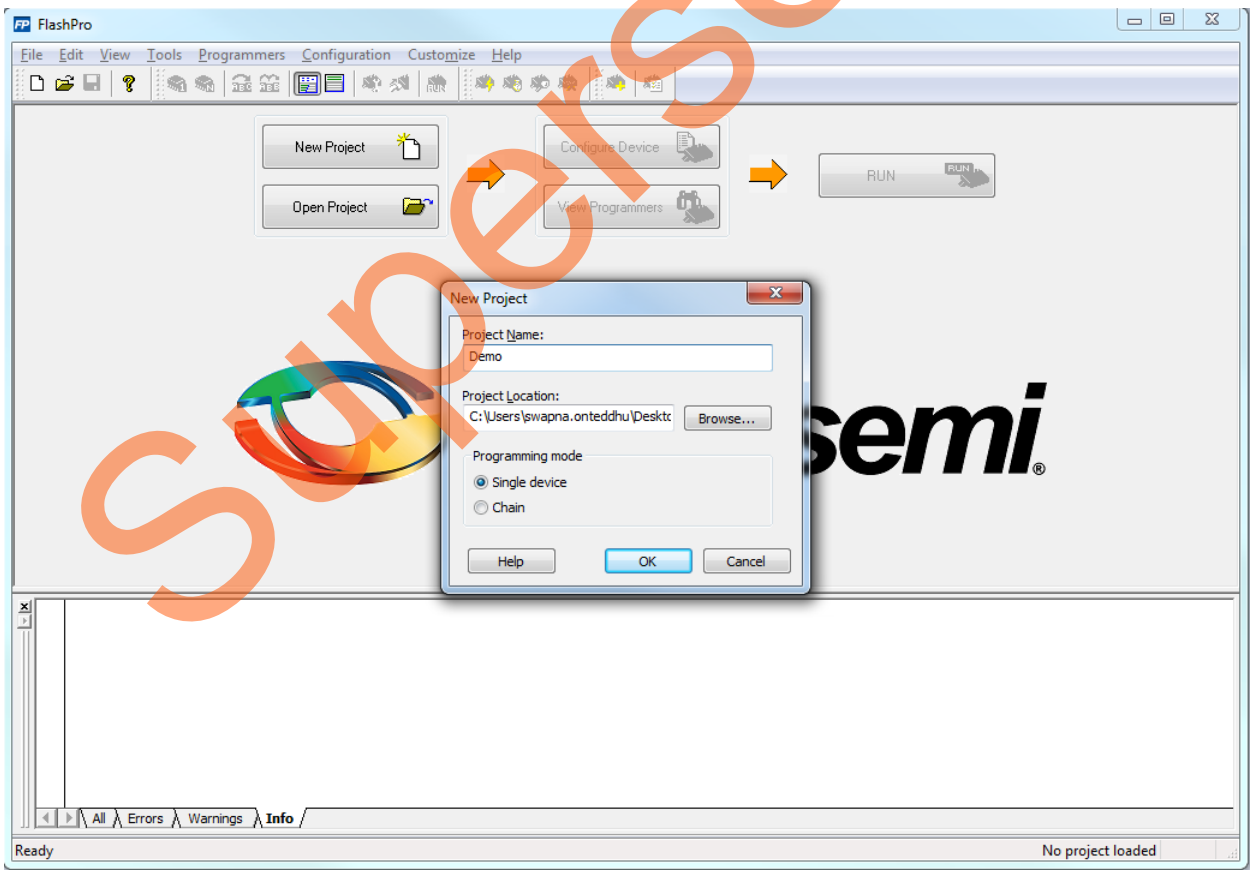
The configuration for the program is:

- Baud Rate: 115200
- Eight data bits
- One stop bit
- No Parity
- No flow control

For more information about how to configure the serial terminal emulation programs, refer to the *Configuring Serial Terminal Emulation Programs Tutorial*.

3. Launch the FlashPro software.
4. Click **New Project**.
5. In the **New Project** window, enter the project name.

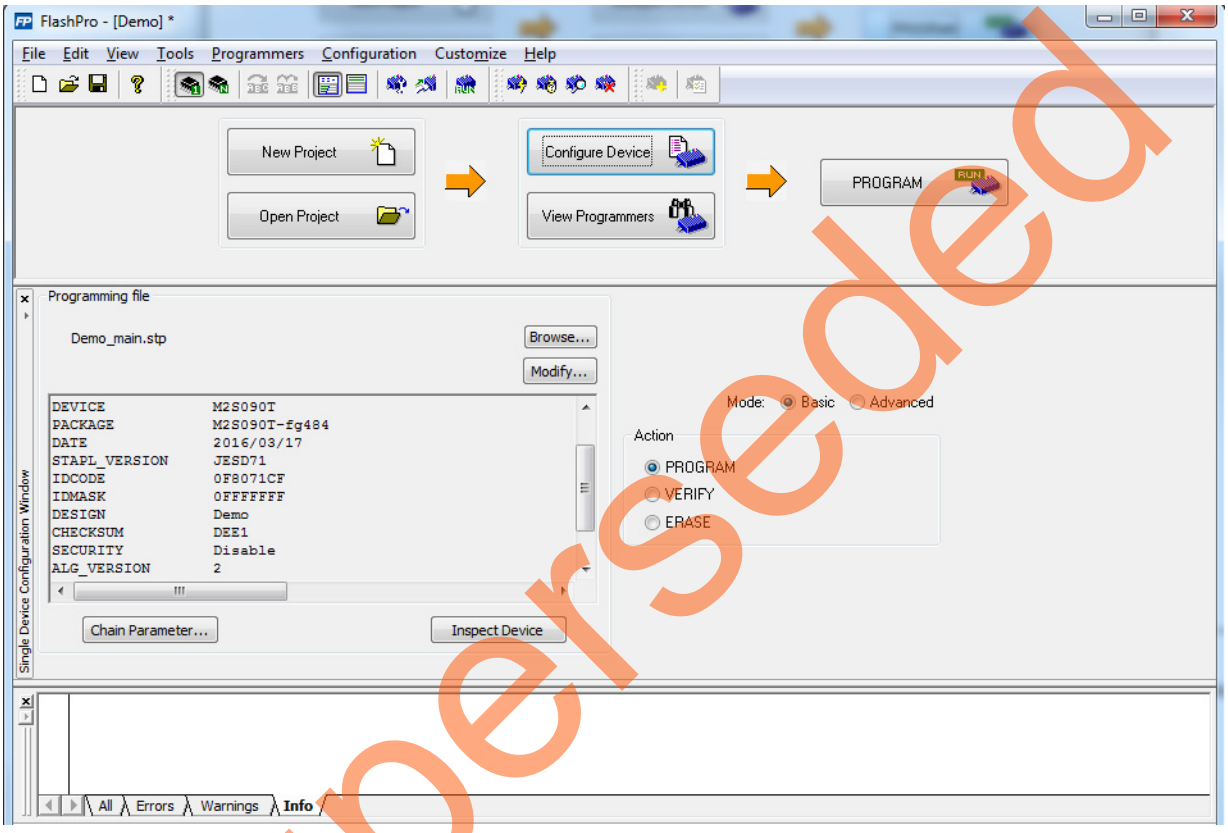
Figure 12 • **FlashPro - New Project Dialog Box**



6. Click **Browse** and navigate to the location where the project needs to be saved.
7. Select **Single device** as the **Programming mode**.

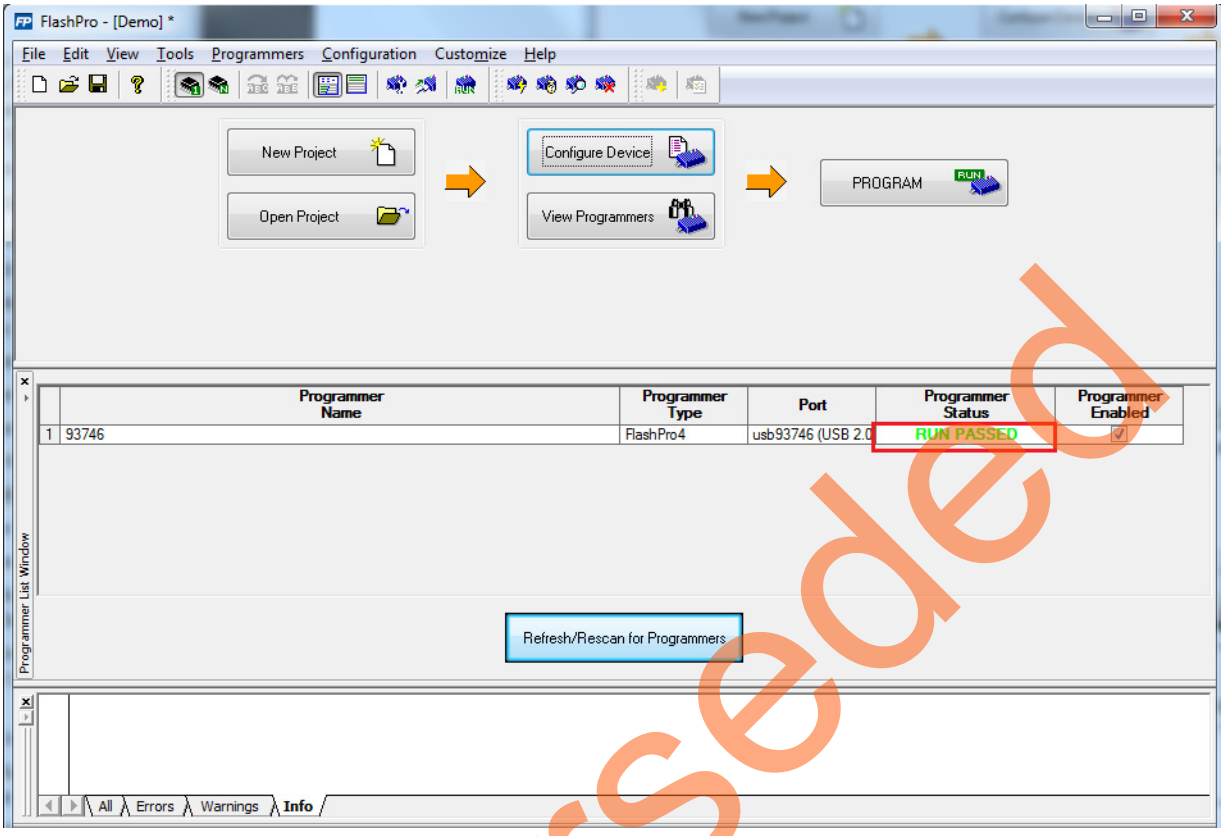
8. Click **OK** to save the project.
9. Click **Configure Device**.
10. Click **Browse** and navigate to the location where the `Demo_main.stp` file is located, and select the file. The default location is: *<download_folder>\sf2_tftp_update_recovery_demo_df\ProgrammingFile\DynamicIP\core-spi\Demo_main.stp.* The required programming file is selected and is ready to be programmed in the device.
11. Click **Open** in **Load Programming File** dialog.

Figure 13 • **Configured FlashPro Project Window**



12. Click **PROGRAM** to start programming the device. Wait until the Programmer Status is changed to **RUN PASSED** as shown in Figure 14 on page 22.

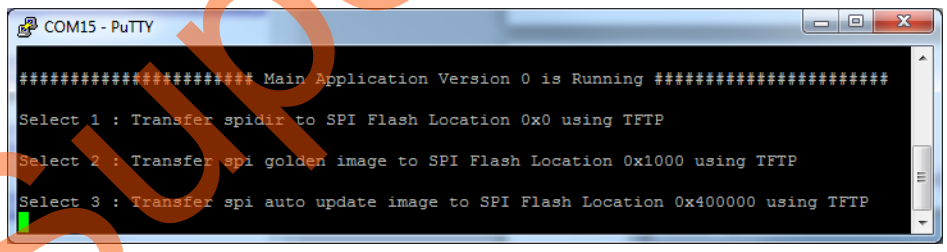
Figure 14 • **FlashPro Program Passed Window**



Note: The demo can be run in both static and dynamic modes. To run the design in Static IP mode, refer to "Appendix: Running the Design in Static IP Mode" section on page 35.

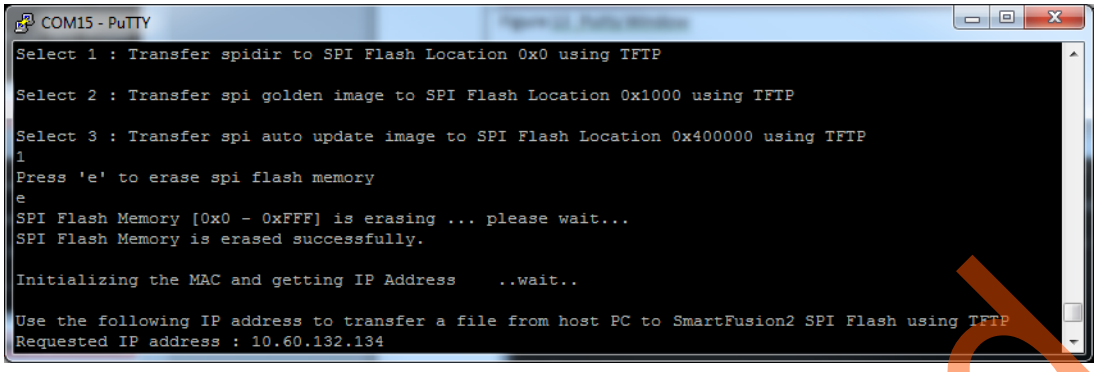
PuTTY displays the message as shown in Figure 15.

Figure 15 • **PuTTY Window**



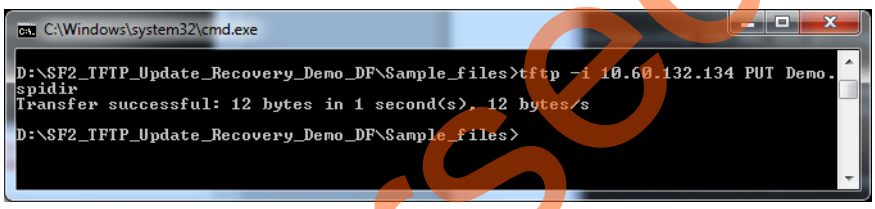
13. Enable TFTP client in Host PC. To enable the TFTP client in Host PC, refer to "Appendix: Enable TFTP Client" section on page 30.
14. Enter **1** to initiate SPI directory transfer.
15. Enter **e** to erase the SPI flash memory location (0x0 – 0xFFF).

Figure 16 • **Erasing the SPI Flash Memory Location [0x0 - 0xFFF]**



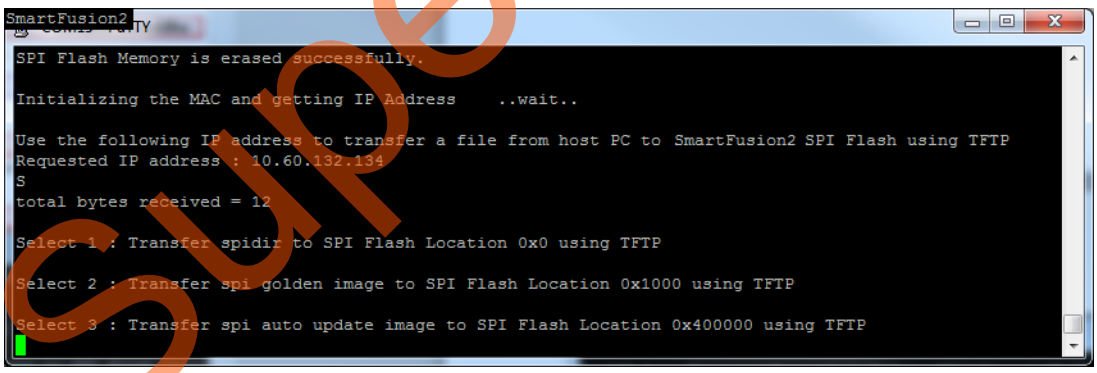
16. After completion of SPI flash erase operation, the Ethernet link is up and IP address is displayed on PuTTY terminal. The LED 2 on the SmartFusion2 Security Evaluation Kit board starts blinking.
17. On the Host PC command prompt, browse to the folder downloadfolder>\SF2_TFTP_Update_Recovery_Demo_DF\Sample_files
18. Type the following command to transfer SPI Directory as shown in Figure 17.
    `tftp -i 10.60.132.134 PUT Demo.spidir`

Figure 17 • **Transfer SPI Directory**



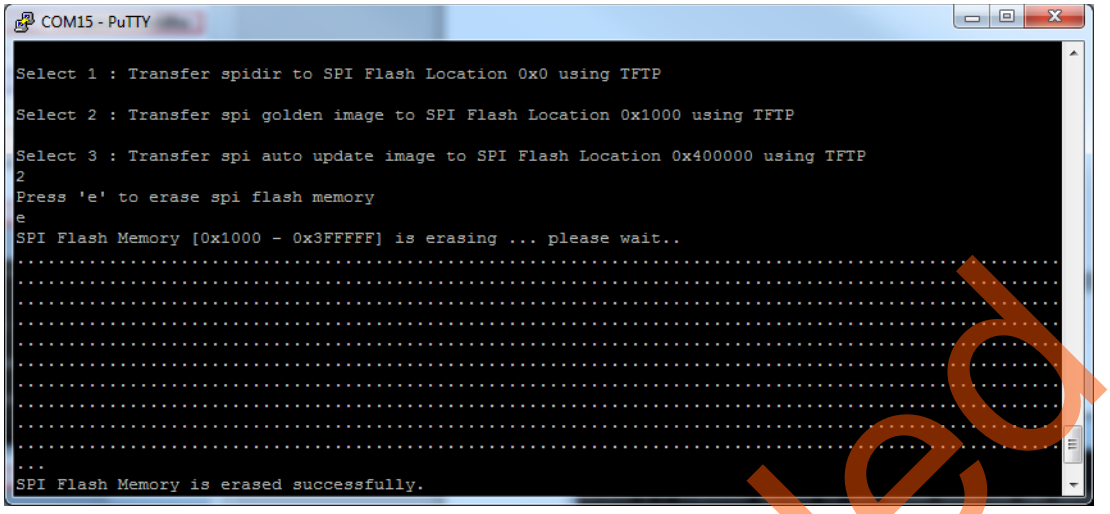
19. Wait until total bytes received message is displayed on the PuTTY terminal, to ensure that the SPI directory TFTP transfered to SPI Flash.

Figure 18 • **SPI Directory Transferred Successfully Window**



To run the Programming Recovery option, select Option 2 and enter **e** to erase the SPI flash memory location [0x1000 – 0x3FFFFF] for transferring the Golden Image version 1 to the address 0x1000.

Figure 19 • **Erase the SPI Flash Memory [0x1000 - 0x3FFFFF]**



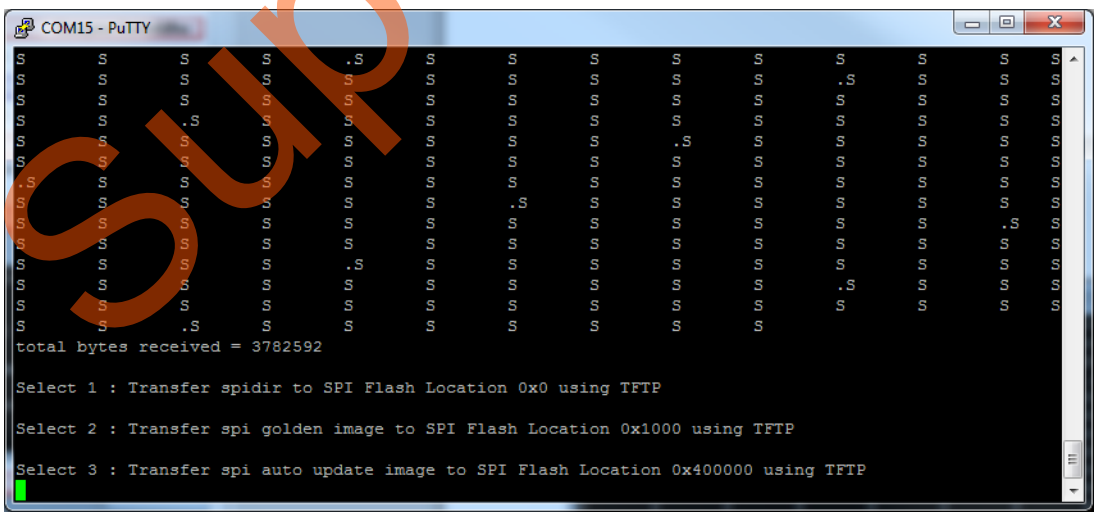
1. On the Host PC command prompt, browse to the folder *downloadfolder>\SF2_TFTP_Update_Recovery_Demo_DF\Sample_files*
2. Type the following command to transfer the Golden Image Version 1 to 0x1000 memory location of SPI Flash.

```
tftp -i 10.60.132.134 PUT Demo_Golden_V1.spi
```

Figure 20 • **Transferring Golden Image Version 1**

Wait until “total bytes received” message is displayed on the PuTTY terminal, to ensure Golden Image Version 1 TFTP transfer to SPI Flash is completed.

Figure 21 • **Successful Transfer of Golden Image Window**

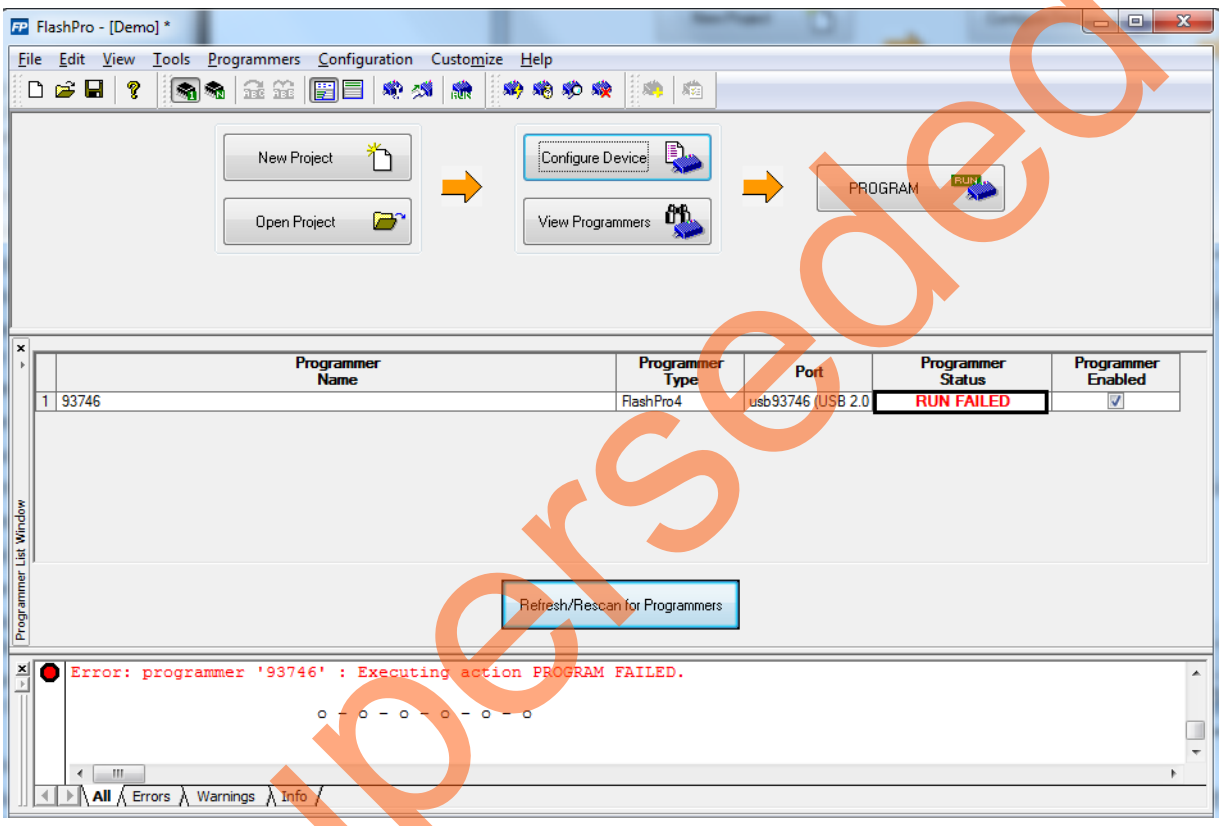


In order to run the Programming Recovery option, create a scenario where the programming operation meets a power failure. One of the ways to perform this task, is to start programming the device using the FlashPro tool. Ensure to switch off the power supply to the board before programming operation is completed by FlashPro programmer. After the power failure situation is created, then the FlashPro

programmer can be removed. The SmartFusion2 System Controller performs Programming Recovery operation, as described below:

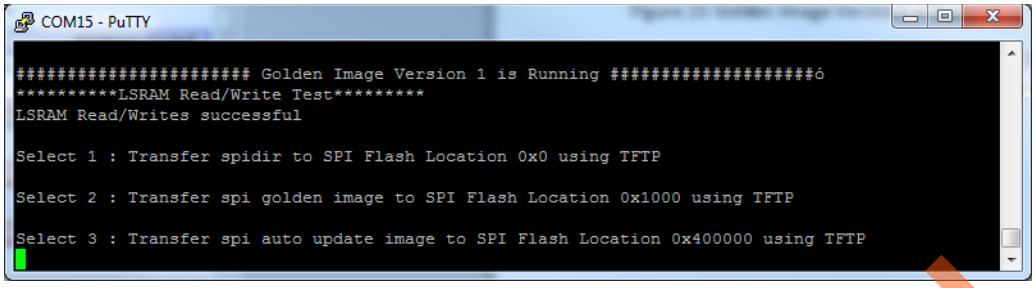
1. Run FlashPro software and select the same `Demo_main.stp` file provided in the Programming File folder.
2. Click **PROGRAM**.
3. While the programming is going ON, user can observe LEDs (H5, H6, J6, H7, F3, F4, and E1) are turned OFF on the SmartFusion2 Security Evaluation Kit board and after seeing LED's turn OFF, immediately switch OFF the board using SW7.
4. FlashPro hardware should stop programming the device and FlashPro software should display an error message as shown in Figure 22.

Figure 22 • **FlashPro Error Display Window**



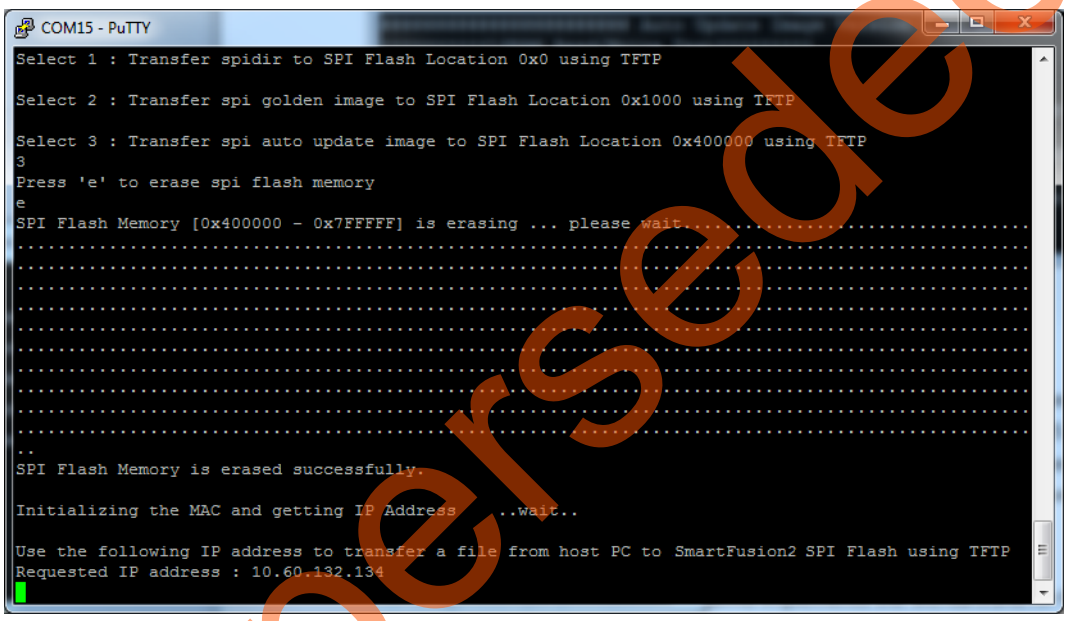
5. Switch ON the SmartFusion2 Security Evaluation Kit board. The System Controller reads the Golden Image address from SPI directory and programs the Golden Image available at 0x1000. The Programming Recovery process takes approximately four to five minutes. After successful Programming Recovery, PuTTY terminal displays a message "Golden Image Version 1 is Running" and Ensure LEDs (H5, J6, G7, and F4) are ON. This activity confirms that the Golden Image Version 1 is recovered.

Figure 23 • **Running Golden Image Version 1**



To run the Auto Update option, select **3** and enter **e** to erase the SPI flash memory location [0x400000 – 0x7FFFFF] for transferring the Auto Update Image Version 2 to address 0x400000.

Figure 24 • **Erasing the SPI Flash Memory Location [0x400000 - 0x7FFFFF]**



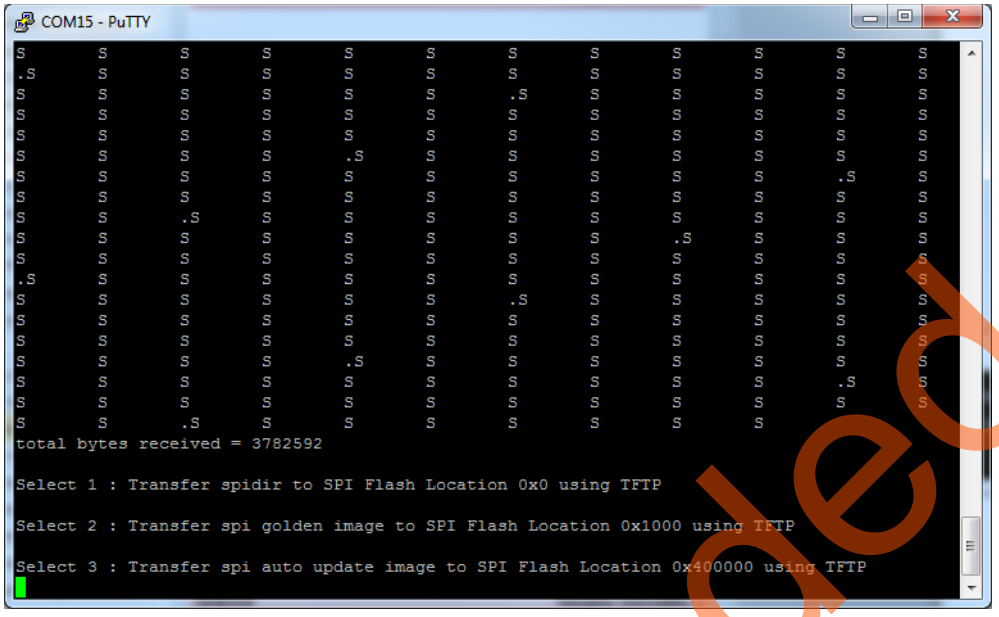
1. On the Host PC Command prompt, browse to the *downloadfolder>\SF2_TFTP_Update_Recovery_Demo_DF\Sample_files* folder
2. Type the following command to transfer Auto Update Image Version 2 to 0x400000 memory location of the SPI flash.

```
tftp -i10.60.132.134 PUT Demo_Update_V2.spi
```

Figure 25 • **Transferring Auto Update Image Version 2**

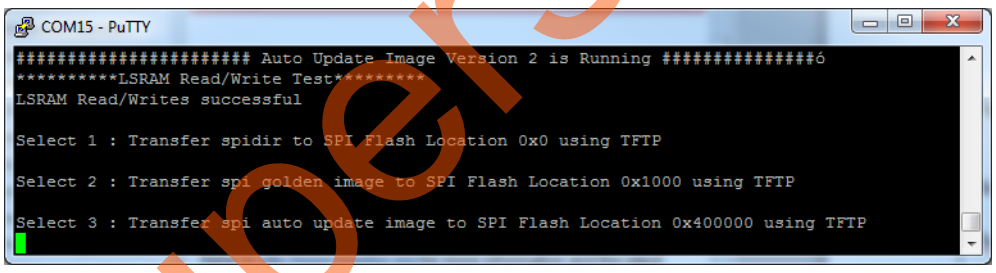
Wait until a message “total bytes received” is displayed on the PuTTY terminal, which ensures that the Auto Update Image Version 2 TFTP transfer to SPI Flash is completed.

Figure 26 • **Transfer Successful Auto Update Image**



3. Select **SW6** to reset the SmartFusion2 Security Evaluation Kit board and wait for approximately four to five minutes.
4. During Auto Update, LEDs (H5, H6, J6, H7, F3, F4, and E1) are turned OFF.
5. After successful Auto Update, PuTTY terminal displays a message “Auto Update Image Version 2 is Running” and also ensure LEDs (H5, H6, J6, are H7) are ON.

Figure 27 • **Auto Update Image**



2.6 Known Issue

The System Controller is unable to change the oscillator frequency after a successful Programming Recovery. The oscillator frequency is 25 MHz instead of 50 MHz. This is a known silicon issue, which is documented in *ER0196- SmartFusion2 Device, Errata*.

After Recovery Operation, junk messages are displayed on the Serial Terminal. The workaround is to apply a system reset after a successful recovery. It is recommended that this workaround is implemented for any design, which uses the Programming Recovery. For more information about how to implement this workaround, refer to "Appendix: Implementing Workaround to Reset the Device after Programming Recovery" section on page 38.

The design example provided in this demo implements the workaround for Programming Recovery issue and the design files are available in the following location:

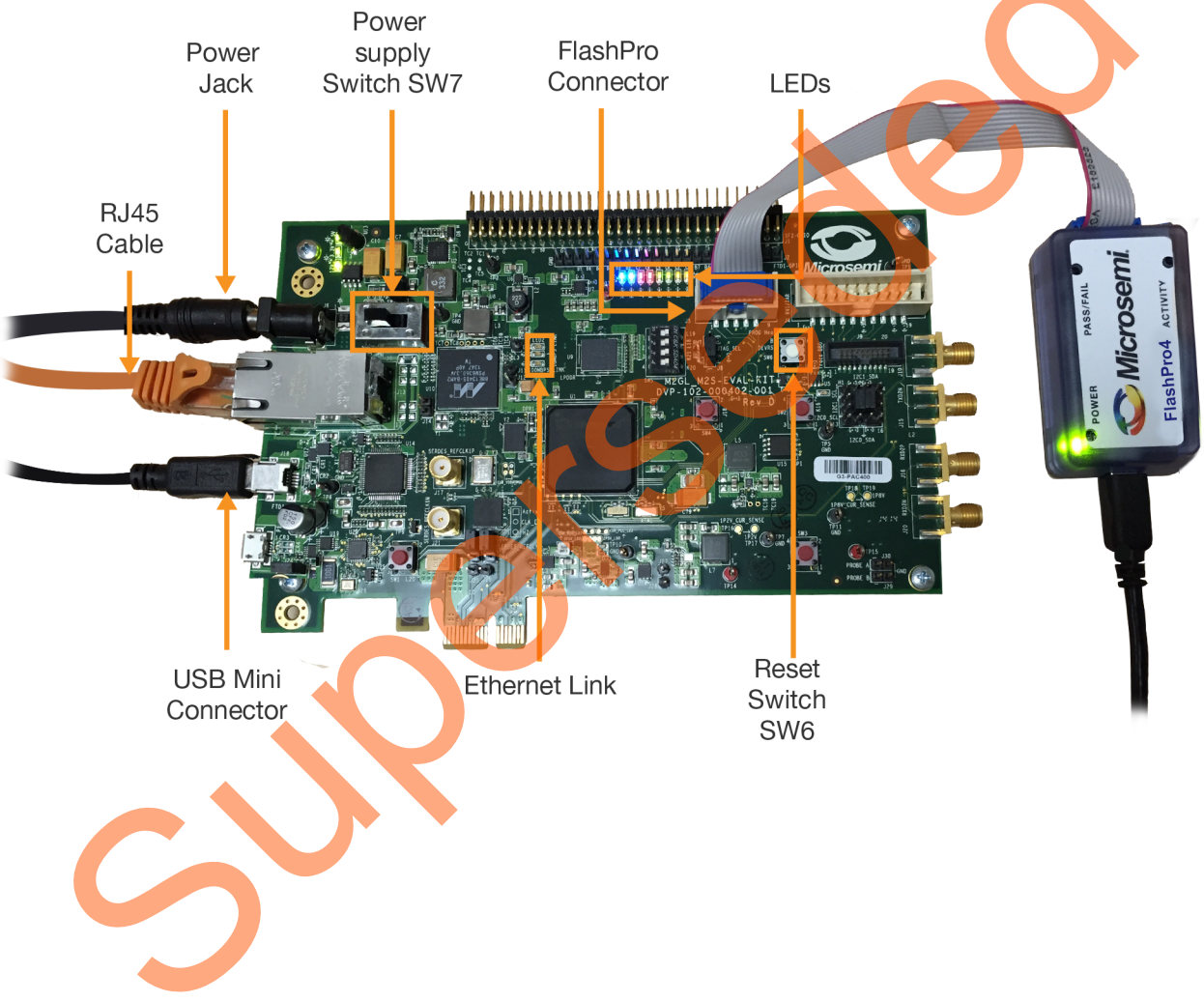
```
<downloadfolder>\SF2_TFTP_Update_Recovery_Demo_DF\Sample_files\Recovery-_WA
```

The same workaround is also implemented in the Program Recovery Image that is, `Demo_Golden_V1.spi` which is demonstrated in "Running the Demo Design" section on page 20 already.

3 Appendix: Board Setup for Running the Demo

Figure 28 shows the board setup for running the demo on the SmartFusion2 Security Evaluation Kit board.

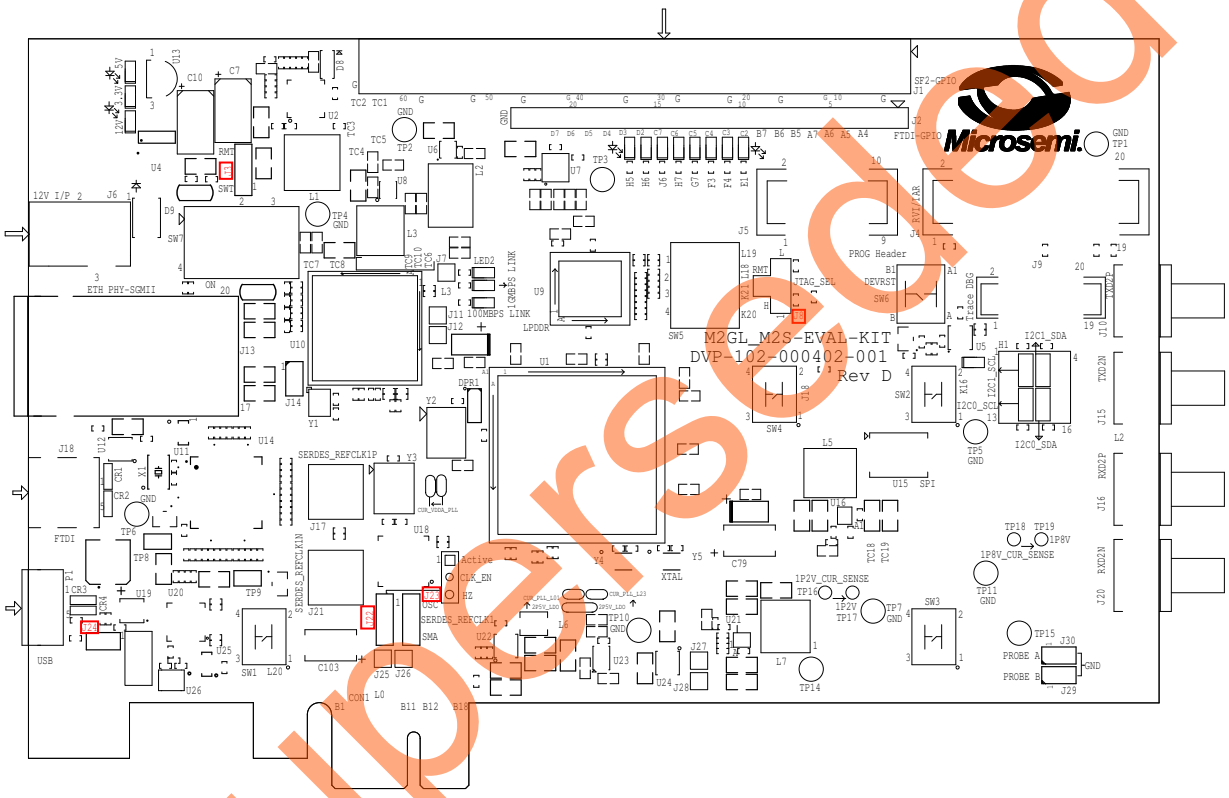
Figure 28 • **SmartFusion2 Security Evaluation Kit Setup**



4 Appendix: SmartFusion2 Security Evaluation Kit Board Jumper Locations

Figure 29 shows the jumper locations on the SmartFusion2 Security Evaluation Kit board.

Figure 29 • **SmartFusion2 Security Evaluation Kit Board Jumper Locations**



Notes:

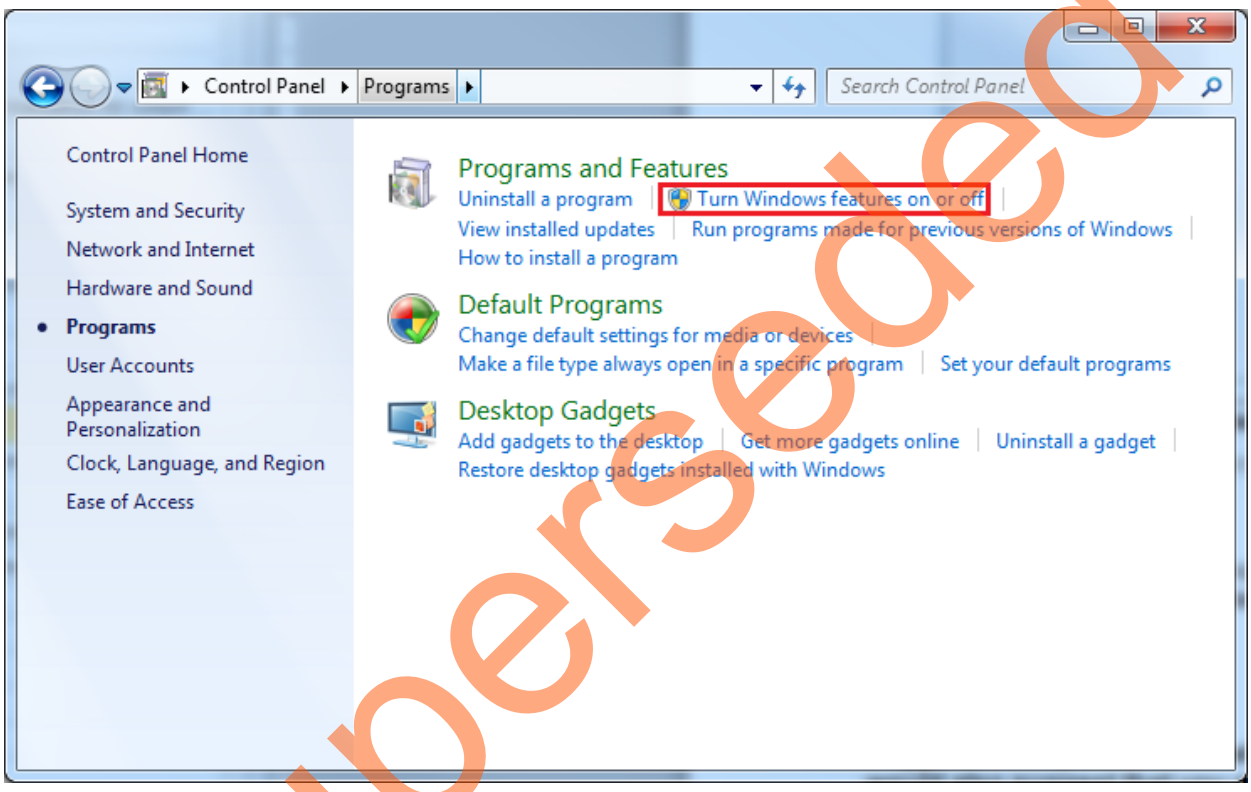
- Jumpers highlighted in red (J22, J23, J24, J3, J8) are set by default.
- The location of the jumpers in Figure 29 are searchable.

5 Appendix: Enable TFTP Client

The following steps describe how to enable TFTP client:

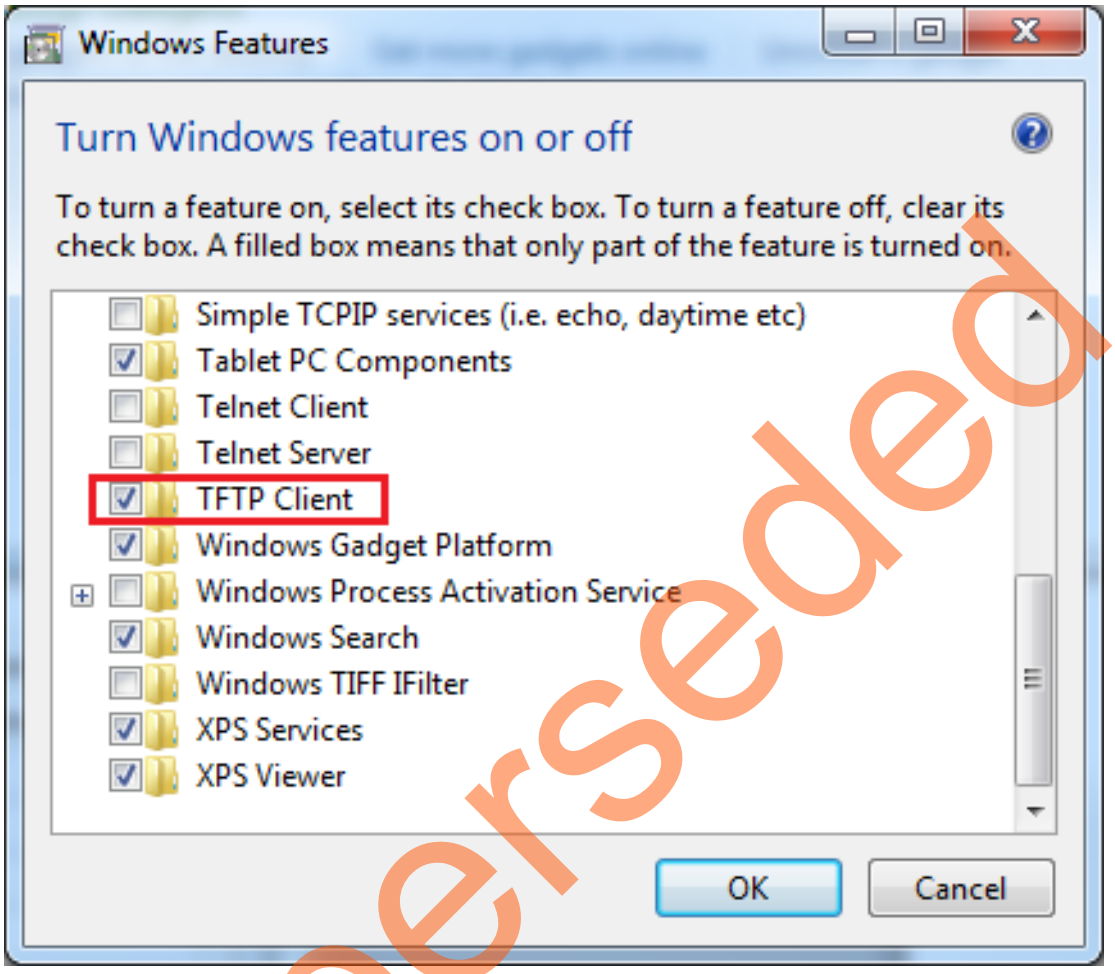
1. Navigate to **Control Panel** > **Programs**. Click **Turn Windows Features On or Off** as shown in Figure 30.

Figure 30 • **Control Panel - Programs and Features**



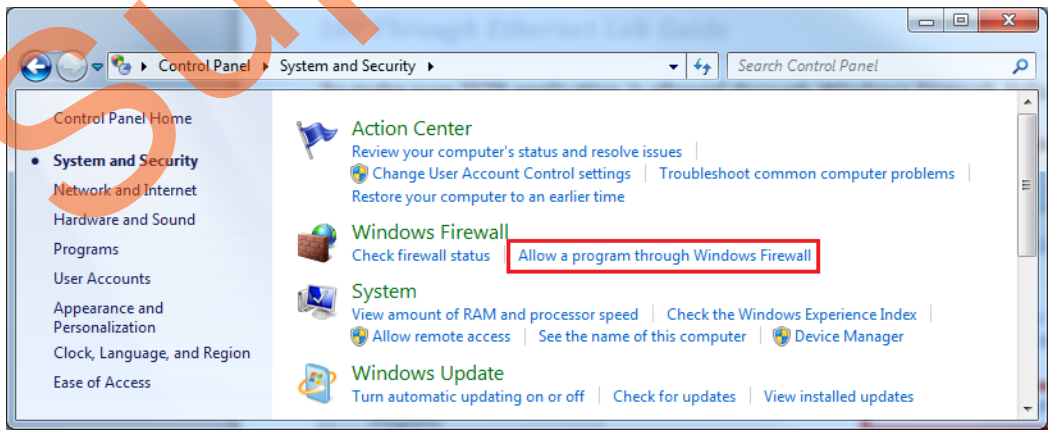
2. Select the **TFTP Client** check box from Windows Features as shown in Figure 31.

Figure 31 • **Selecting TFTP Client from Windows Features**



3. Browse through **Control Panel** > **System and Security**, click **Allow a program through Windows Firewall**.

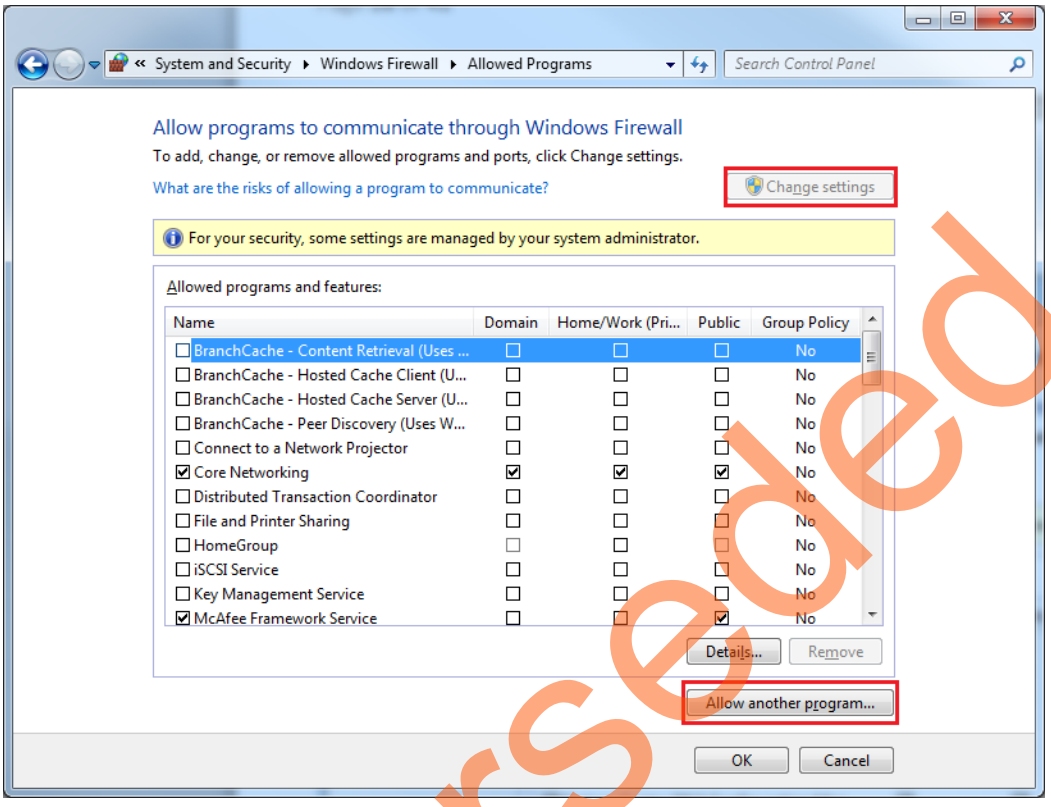
Figure 32 • **System and Security Window**



Note: If the System and Security option is not available, then enter the firewall in the search window to perform step 3.

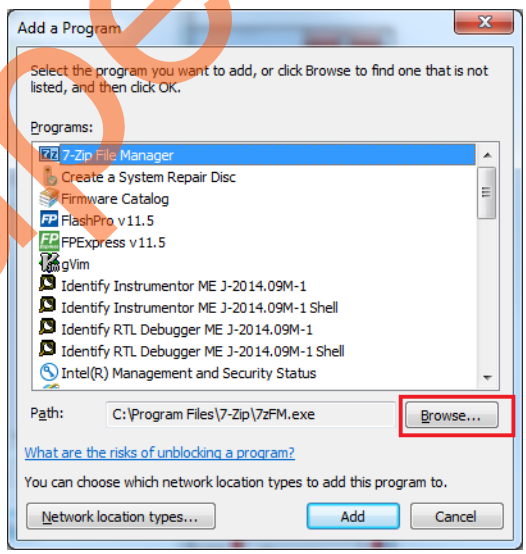
4. Click **Change settings** and choose **Allow another program…**

Figure 33 • **Allowed Programs Window**



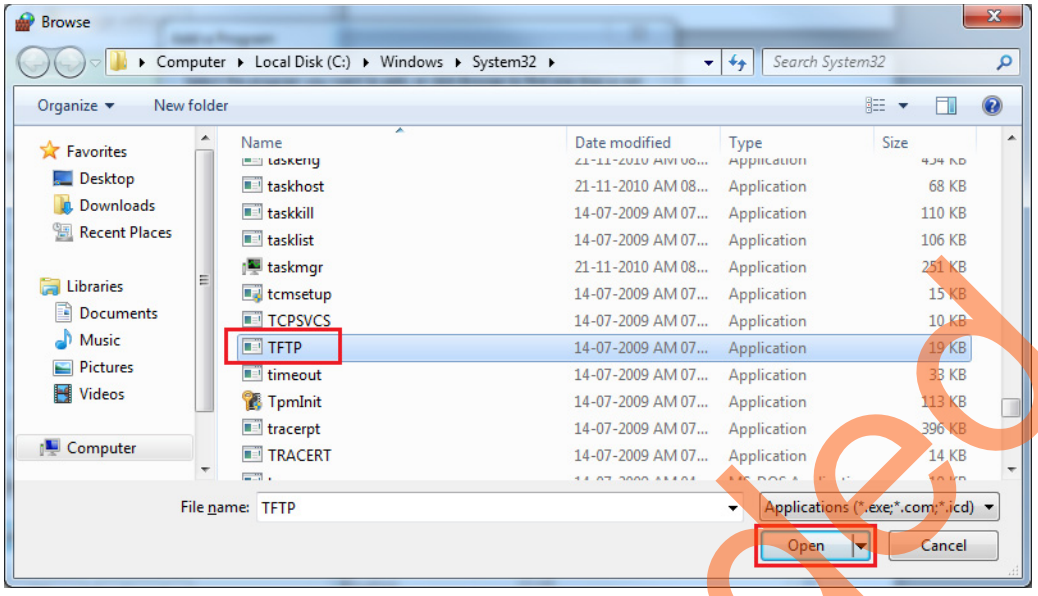
5. The **Add a Program** window is displayed and click **Browse…**

Figure 34 • **Add a Program Window**



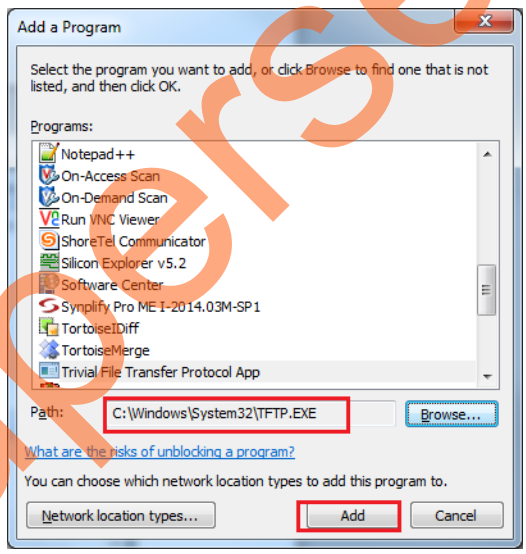
6. Browse through `C:\ -> Windows->System32` and choose **TFTP.exe** and click **Open**.

Figure 35 • **Selecting the TFTP Executable File**



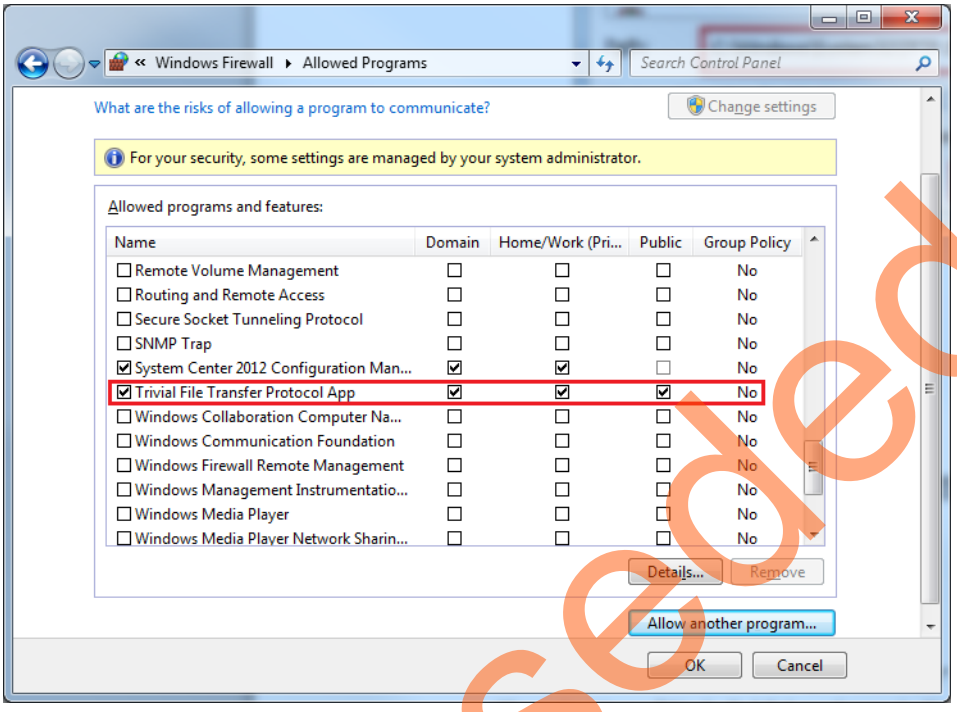
7. Ensure that the TFTP.EXE path (C:\Windows\System32\TFTP.EXE) is selected correctly and Click **Add.**

Figure 36 • **TFTP.EXE Path in Add a Program Window**



8. Ensure that the **Trivial File Transfer protocol App** is added and also select all the check boxes (Domain, Home/Work, Public) as shown in Figure 37.

Figure 37 • **Selecting Trivial File Transfer Protocol App in Allowed Programs Window**



9. Click **OK**.

6 Appendix: Running the Design in Static IP Mode

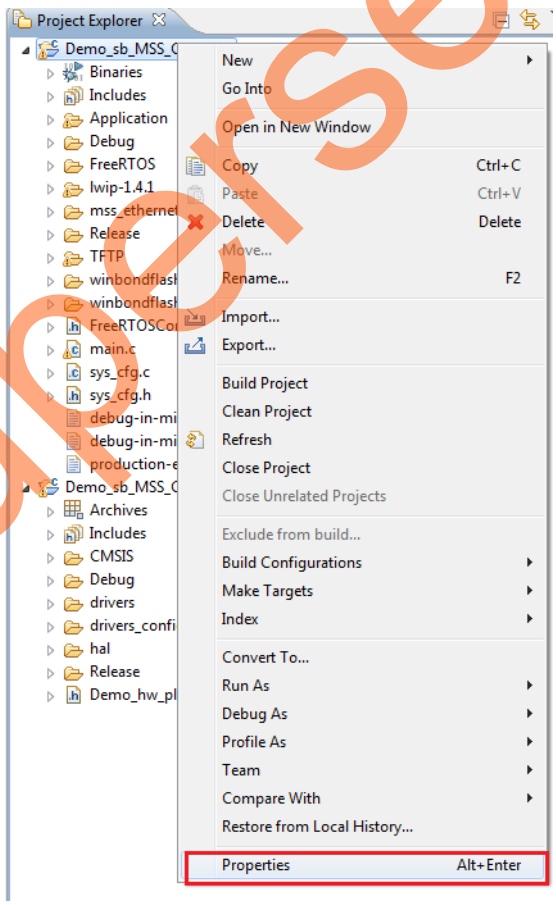
A static IP address will never change until it is explicitly changed and it is a permanent address assigned to a device to access the Internet. A device using a static IP address must be reconfigured each time it switches the networks. A dynamic IP address is a temporary address that is assigned each time a computer or device accesses the Internet. Dynamic IP address requires a DHCP-capable router.

Note: In this demo, the design files are provided with the dynamic IP address settings.

The following steps describe how to run the design in Static IP mode:

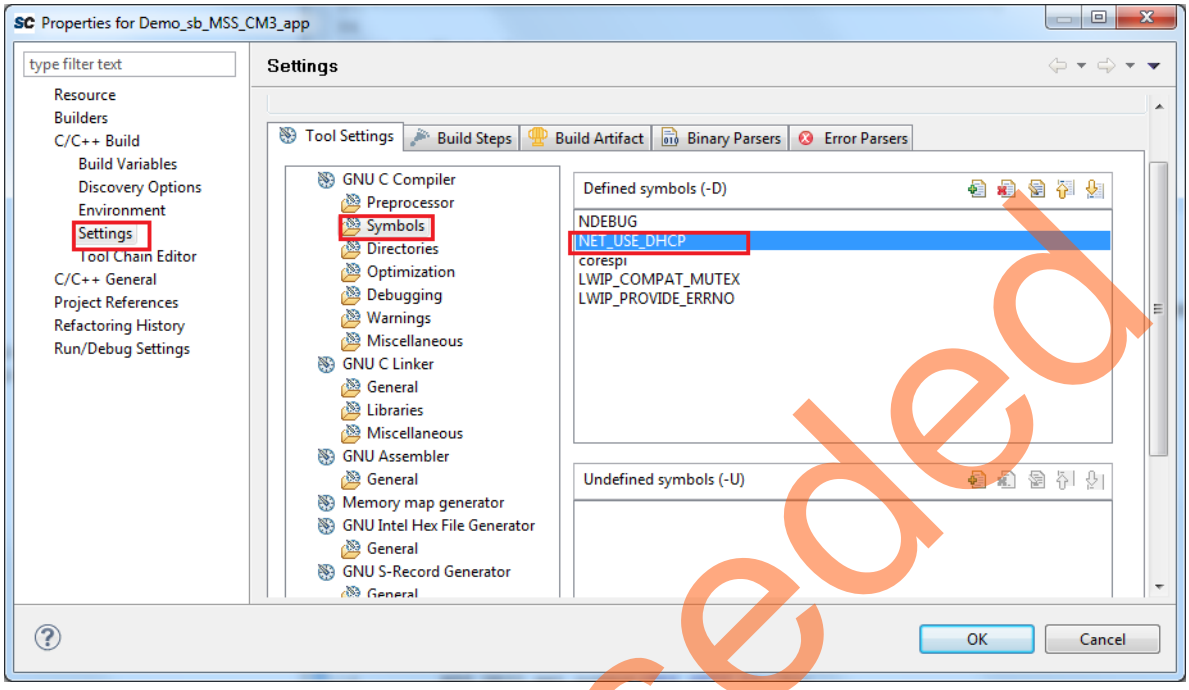
1. Click **Start > Programs > Microsemi SoftConsole v3.4 > Microsemi SoftConsole v3.4.0.5** to open SoftConsole IDE.
2. Browse to the project Location *<downloadfolder>\SF2_TFTP_Update_Recovery_Demo_DF\Libero\Demo\SoftConsole\Demo_sb_MSS_CM3\projects*.
3. To run the design in Static IP mode, right-click the **Demo_sb_MSS_CM3_app** project and select **Properties** as shown in Figure 38.

Figure 38 • **Project Explorer Window SoftConsole Project**



4. Remove the symbol **NET_USE_DHCP** in **Tool Settings** of the Properties for Demo_sb_MSS_CM3_app window as shown in Figure 39.

Figure 39 • **Properties for Demo_sb_MSS_CM3_app**



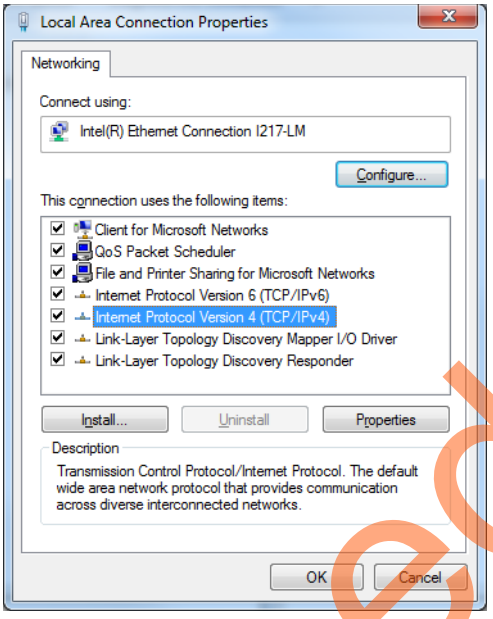
5. Rebuild the SoftConsole Project. Load the design in to the eNVM.

Note: Programming file with static IP settings is available in the following path. *<Downloadfolder>SF2_TFTP_Update_Recovery_Demo_DF\Programmingfile\StaticIP*.

6. Navigate to Control Panel and enter **Network and Sharing Center** in search window.
7. In **Network and Sharing Center** window, select **Change Adapter Settings**.
8. Right-click on **Local Area Connection** and select **Properties**.

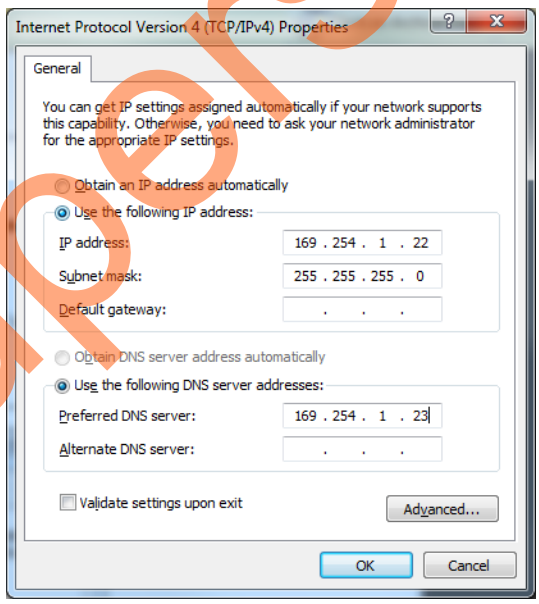
9. If the device is connected in Static IP mode, the board static IP address is 169.254.1.23, then change the host TCP/IP settings to reflect the IP address. Figure 40 shows Host PC TCP/IP settings.

Figure 40 • **Local Area Connection Properties Window**



10. Update the static IP settings as shown in the Figure 41. Click **OK**.

Figure 41 • **Internet Protocol Version 4 (TCP/IPv4) Properties**



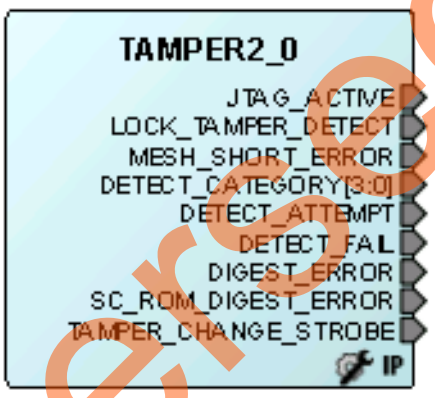
11. Run the demo with Static IP address 169.254.1.23 as described in the Running the Demo Section.

7 Appendix: Implementing Workaround to Reset the Device after Programming Recovery

The following changes are required in the Libero design.

1. Select **File** > **New** > **SmartDesign**.
2. Enter the name as Program_Recovery_WA in **Create New SmartDesign** window.
   - Browse to the Libero catalog to open **Tamper Macro**. The Tamper Macro resets the device when the RESET_N input port is connected to the logic 0.
   - Drag-and-drop the **Tamper Macro** that is available in the Libero catalog to the Program_Recovery_WA SmartDesign canvas as shown in Figure 42.

Figure 42 • **Tamper Macro**



- Select the **Enable RESET** function check box in the Configuring Tamper 2_0 window.
- Click **OK**. The System Reset option is enabled.

Figure 43 • **Tamper Macro Configuration Window**

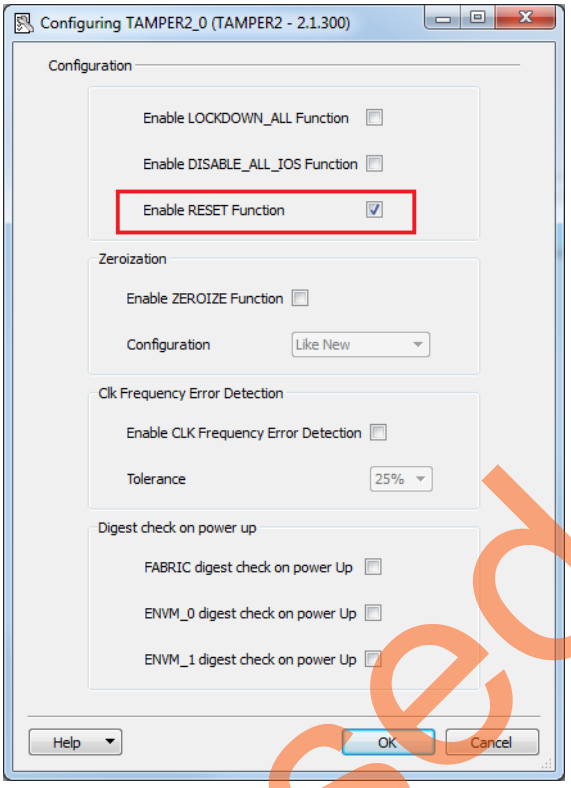
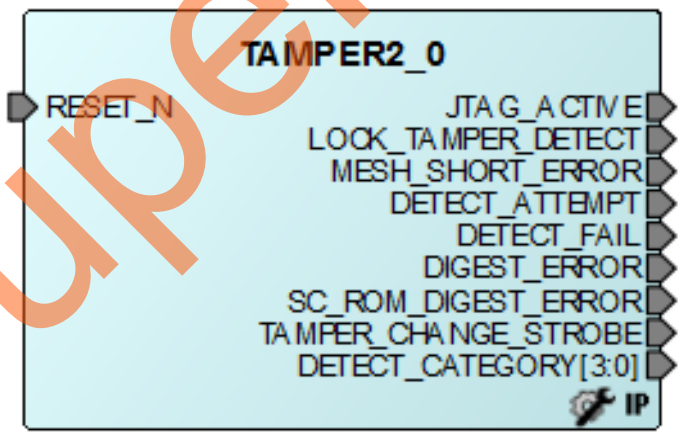


Figure 44 shows the TAMPER2_0 macro after configuration.

Figure 44 • **Tamper Macro**



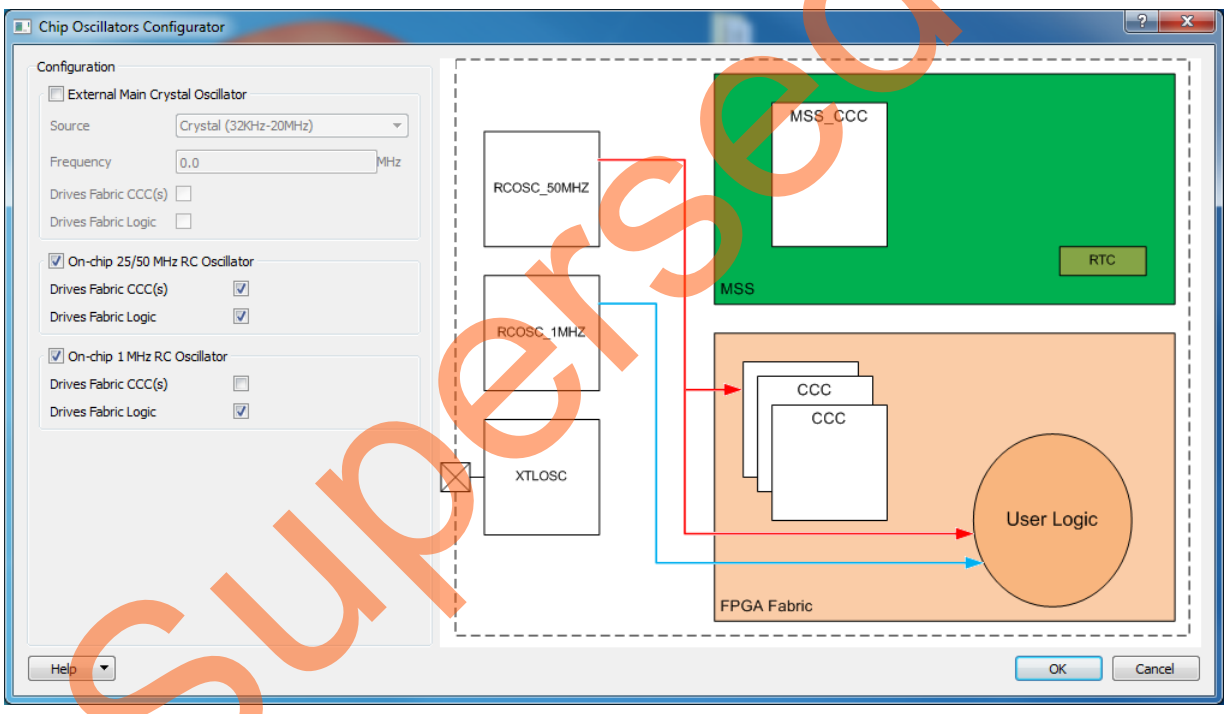
3. Instantiate the Clock_check HDL Module that is provided in the design files. The HDL module is a 25 bits ring counter and it counts the number of pulses coming into clock input pin and it is enabled by the logic high on the reset pin. The pulse output pin is asserted high only when any of the ring counter bits 10, 11, 12, 13, or 14 is high, and it occurs only at the negative edge of reset. So, this module generates output pulses only for 25 MHz and not for 50 MHz. Follow the steps to add Clock_check HDL module to Libero design.
   - Choose File > Import > HDL Source Files.
   - Browse to the following `Clock_check.v` file location in the design files folder.
   - `<downloadfolder>\SF2_TFTP_Update_Recovery_Demo_DF\Sample_files\Recovery_WA\Sourcefiles`

- Click **Program_Recovery_WA** tab and drag-and-drop the `Clock_check.v` component from the from the Design Hierarchy to the Program_Recovery_WA SmartDesign canvas. Figure 45 shows the Clock_check HDL module.

Figure 45 • **Clock_check HDL Component**

4. Right-click **Demo_sb** under **Demo** in **Design Hierarchy** window and select **Open As SmartDesign**.
5. Select **Demo_sb** tab and double-click **FABOSC_0 (On Chip Oscillator)**.
6. Configure the FABOSC_0 with the following settings as shown in Figure 46.
   - On-chip 25/50 MHz RC Oscillator should be enabled to Drive Fabric Logic

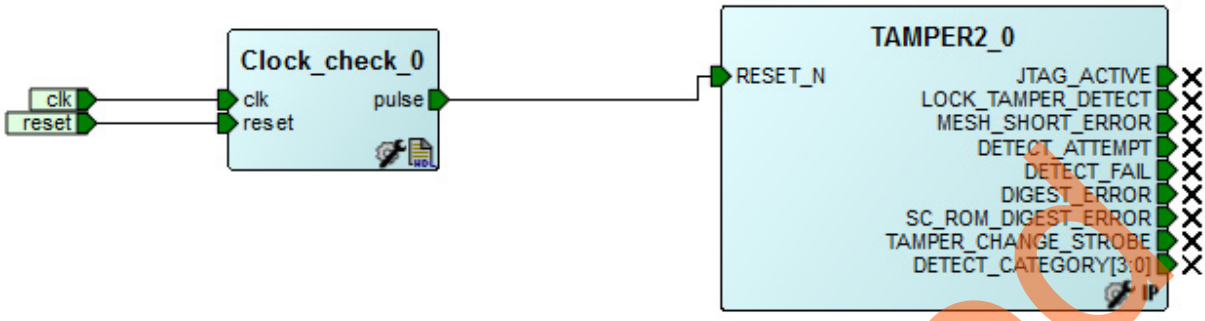
Figure 46 • **Chip Oscillators Configurator**



7. Select **Demo** tab.
8. Right-click **Demo_sb_0** and select **Update Instance** with the latest component.

9. Click **Program_Recovery_WA** tab and make the connections as shown in Figure 47.

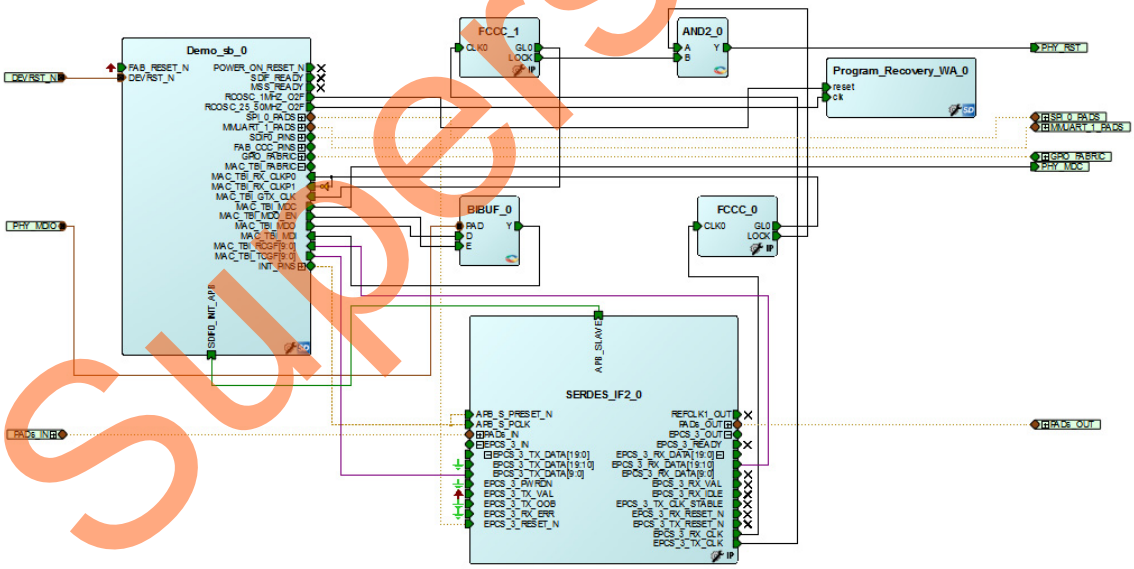
Figure 47 • **Program_Recovery_WA**



10. Click **Demo** tab and drag-and-drop the **Program_Recovery_WA** component from the Design Hierarchy to the Demo SmartDesign canvas.

Make the connection as shown in Figure 48 and generate Demo SmartDesign. This completes the implementation of the workaround.

Figure 48 • **Demo Smart Design**



Notes:

1. If the Auto Update feature is enabled and update image is available in the Flash memory, then after recovery, a default reset occurs due to the workaround. As a result, Update Image is programmed, if the Update Image version is greater than the Golden Image.
2. This workaround only works for Mode 1.2 settings in the SPI images.

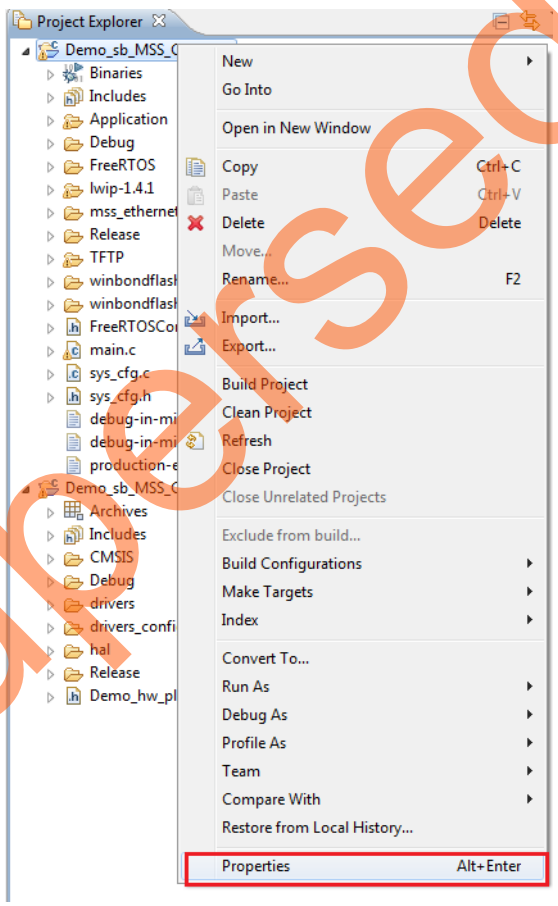
Reason: If the SPI images are set to Mode 1 V settings, then the device may undergo permanent reset. As it continuously holds reset by the workaround since, the oscillator frequency is 25 MHz.

8 Appendix: Configuring MSS SPI0 in SoftConsle Project

The following procedure allows you to configure MSS SPI0 in the SoftConsole project.

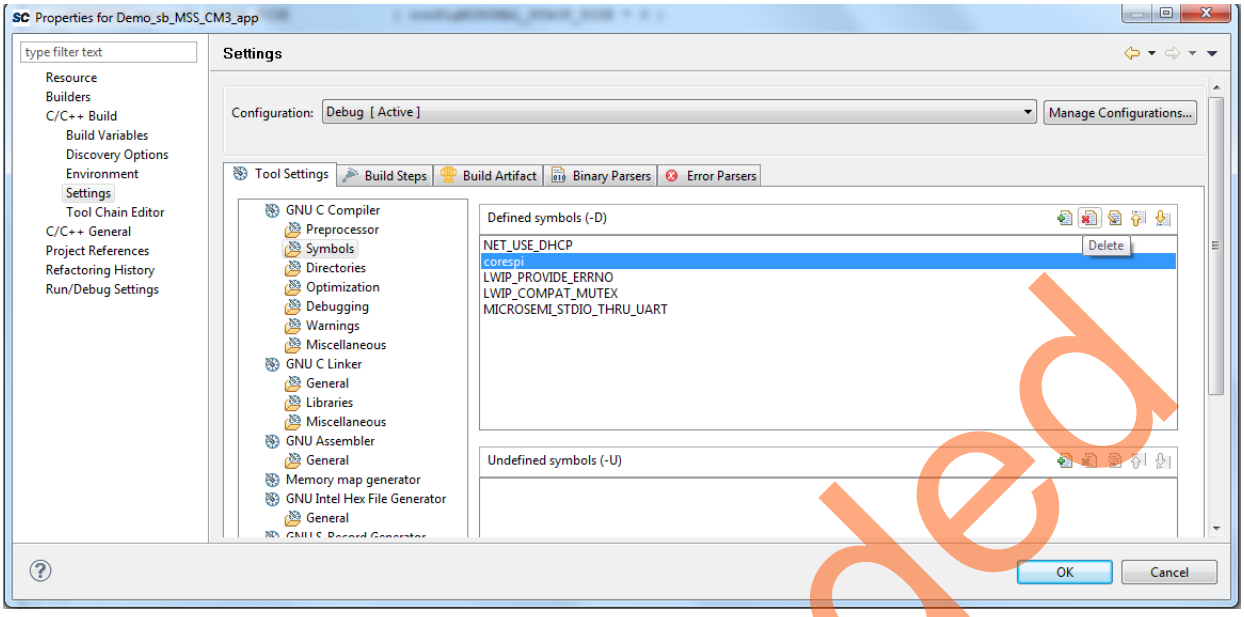
1. Click Start > Programs > Microsemi SoftConsolev3.4 > Microsemi SoftConsole v3.4.0.5 to open SoftConsole IDE.
2. Browse to the project location: *<downloadfolder>\SF2_TFTP_Update_Recovery_Demo_DF\Libero\Demo\SoftConsole\Demo_sb_MSS_CM3\projects*
3. To configure MSS SPI0, right-click the **Demo_sb_MSS_CM3_app** project and select **Properties** as shown in Figure 49.

Figure 49 • **Project Explorer - Properties**



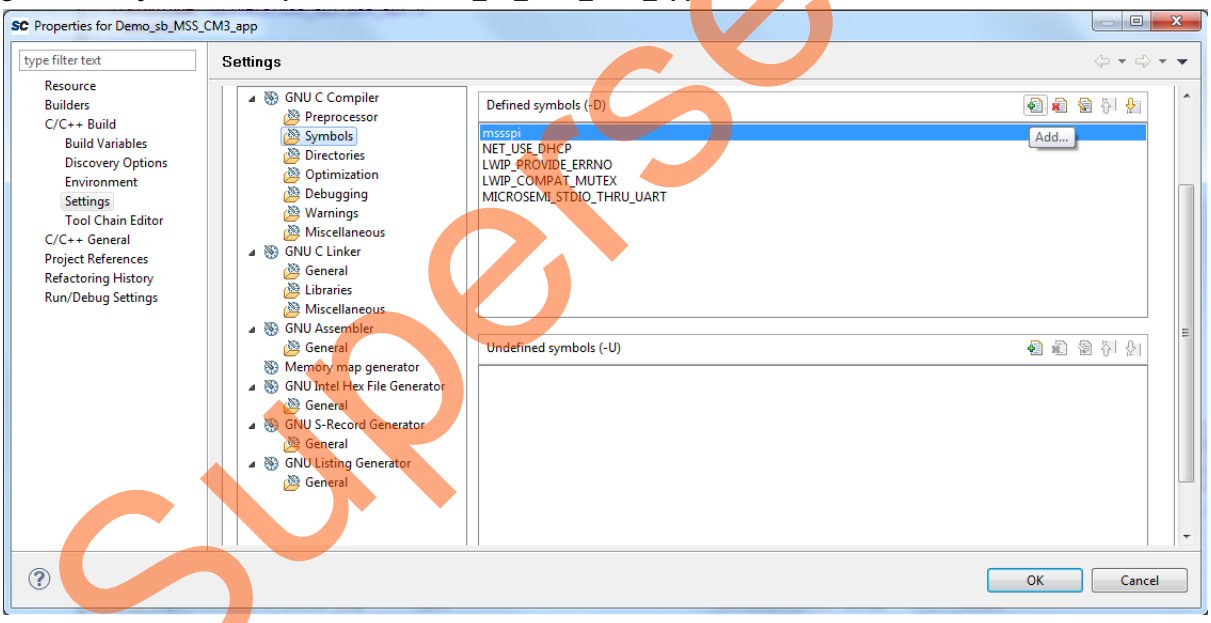
4. Remove the symbol **corespi** in Tool Settings of the Properties for **Demo_sb_MSS_CM3_app** window as shown in Figure 50 on page 43.

Figure 50 • **Properties for Demo_sb_MSS_CM3_app**



5. Add symbol **mssspi** in the properties for **Demo_sb_MSS_CM3_app** window as shown in Figure 51.

Figure 51 • **Symbols - Properties for Demo_sb_MSS_CM3_app**



6. Rebuild the SoftConsole Project. Load the design in to the eNVM.
7. Programming file with MSSSPI0 settings is available in the following path:
   - *<Downloadfolder>SF2_TFTP_Update_Recovery_Demo_DF\Programmingfile\StaticIp\mssspi\Demo_main.stp*
   - *<Downloadfolder>SF2_TFTP_Update_Recovery_Demo_DF\Programmingfile\DynamicIp\mssspi\Demo_main.stp*

9 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.

9.1 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

9.2 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.

9.3 Technical Support

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

9.4 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.

9.5 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.

9.5.1 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.

9.5.2 My Cases

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

9.5.3 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.

9.6 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.