
Running the Secure Webserver on SmartFusion2 Devices Using PolarSSL, lwIP and FreeRTOS

Demo Guide

Superseded

April 2014

Revision History

Date	Revision	Change
7 April 2014	Revision 1	First release

Confidentiality Status

This is a non-confidential document.

Superseded

Table of Contents

Preface	4
About this document	4
Intended Audience	4
References	4
Microsemi Publications	4
Running the Secure Webserver Demo Design on the SmartFusion2 Devices Using PolarSSL, lwIP and FreeRTOS	5
Introduction	5
Secure Webserver Demo Design Overview	5
Requirements and Details	8
Demo Design	8
Introduction	8
Demo Design Features	9
Demo Design Description	9
Setting up the Demo Design	16
Board Setup Snapshot	18
Running the Demo Design	18
Running the Secure Webserver Demo with Microsoft Internet Explorer	25
Running the Secure Webserver Demo with Mozilla Firefox	26
Appendix 1: Board Setup for Running the Secure Webserver	31
Appendix 2: Jumper Locations	32
Appendix 3: Running the Design in Static IP Mode	33
Product Support	36
Customer Service	36
Customer Technical Support Center	36
Technical Support	36
Website	36
Contacting the Customer Technical Support Center	36
Email	36
My Cases	37
Outside the U.S.	37
ITAR Technical Support	37

Preface

About this document

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

Intended Audience

The following designers using the SmartFusion2 devices:

- FPGA designers
- Embedded designers
- System-level designers

References

The following references are used in this document:

- PolarSSL TLS/SSL protocol: <https://polarssl.org/>
- lwIP TCP/IP stack:
 - www.sics.se/~adam/lwip/
 - <http://download.savannah.gnu.org/releases/lwip/>
- FreeRTOS stack: www.freertos.org

Microsemi Publications

- SmartFusion2 Microcontroller Subsystem User Guide
- SmartFusion2 SoC FPGA High Speed Serial Interfaces User Guide
- Libero SoC User Guide
- SmartFusion2 Development Kit User Guide

Refer to the following web page for a complete and up-to-date listing of SmartFusion2 device documentation: www.microsemi.com/products/fpga-soc/soc-fpga/smartfusion2.

Running the Secure Webserver Demo Design on the SmartFusion2 Devices Using PolarSSL, lwIP and FreeRTOS

Introduction

This demo explains the Secure Webserver capabilities using transport layer security (TLS) and secure sockets layer (SSL) protocol and TSEMAC of the SmartFusion2 devices. This demo describes:

- Use of SmartFusion2 Ethernet MAC connected to a serial gigabit media independent interface (SGMII) PHY.
- Integration of SmartFusion2 MAC driver with the PolarSSL library (free TLS/SSL protocol library), lwIP TCP/IP stack and the FreeRTOS operating system.
- Use of Microsemi[®] cryptographic system services in the implementation of TLS/SSL protocol.
- Implementation of the Secure Webserver application on the SmartFusion2 Development Kit board.
- Procedure to run the demo.

The microcontroller subsystem (MSS) of the SmartFusion2 device has an instance of the TSEMAC peripheral. The TSEMAC can be configured between the host PC and the Ethernet network at the following data transfer rates (line speeds):

- 10 Mbps
- 100 Mbps
- 1000 Mbps

Refer to the [SmartFusion2 Microcontroller Subsystem User Guide](#) for more information on the TSEMAC interface for SmartFusion2 devices.

Secure Webserver Demo Design Overview

The Secure Webserver application supports TLS/SSL security protocol that encrypts and decrypts the messages to secure the communication against message tampering. Communication from the Secure Webserver ensures that the sensitive data can be translated into a secret code that is difficult to tamper the data. The Secure Webserver demo design consists of the following layers:

- Application Layer
- Security Layer
- Transport Layer
- Firmware Layer

Figure 1 shows the block diagram of the Secure Webserver demo design.

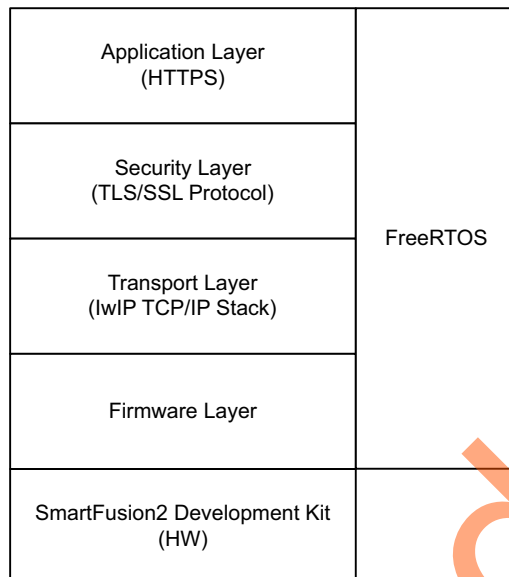


Figure 1 • Block Diagram of Secure Webserver Demo Design on SmartFusion2 Device

Application Layer

The Secure Webserver application is implemented on the SmartFusion2 Development Kit board. The application handles the HTTPS request from the client browser and transfers the static pages to the client in response to their requests. These pages run on the client (Host PC) browser. Figure 2 shows the block diagram of the connecting server (Secure Webserver application running on SmartFusion2 device) and client (web browser running on Host PC).

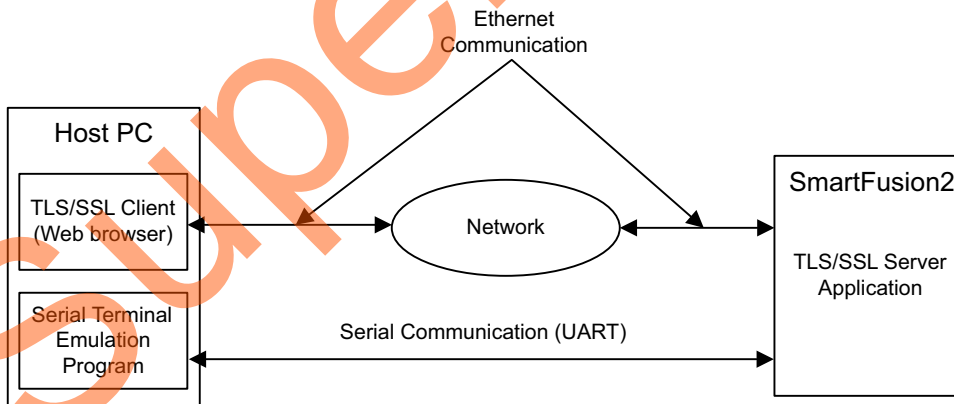


Figure 2 • Client Server Communication Block Diagram

When the URL with IP address (for example, <https://10.60.3.120>) is typed in the browser, the HTTPS request is sent to the port on the Secure Webserver. The Secure Webserver then interprets the request and responds to the client with the requested page or resource.

Security Layer (TLS/SSL Protocol)

Internet browsers and Webservers use TLS/SSL protocol to transmit information securely. TLS/SSL protocol is used to authenticate the server and client to establish the secure communication between authenticated parties using encrypted messages. This protocol is layered above the transport protocol, TCP/IP as shown in [Figure 1 on page 6](#). This protocol provides privacy and reliability in data transfers between the client (internet browser) and the Webserver. An Open Source PolarSSL library is used to implement the TLS/SSL protocol for the Secure Webserver application in this demo.

Refer to the following URLs for complete TLS/SSL protocol implementation details:

Transport Layer Security protocol Version 1.2: <http://tools.ietf.org/html/rfc5246>

Transport Layer Security protocol Version 1.1: <http://tools.ietf.org/html/rfc4346>

Transport Layer Security protocol Version 1.0: <http://tools.ietf.org/html/rfc2246>

Secure Sockets Layer protocol Version 3.0: <http://tools.ietf.org/html/rfc6101>

The PolarSSL library includes the cryptographic and TLS/SSL protocol implementations. This library provides the application programming interface functions to implement Secure Webserver application using the TLS/SSL protocol and the software cryptographic algorithms.

Refer to <https://polarssl.org/> for TLS/SSL protocol library source code written in C and licensing information.

Transport Layer (lwIP TCP/IP Stack)

The lwIP stack is suitable for the embedded systems because of less resource usage. It can be 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.

The lwIP is available (under a BSD license) in C source-code format for download from the following address: <http://download.savannah.gnu.org/releases/lwIP/>

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. Refer to <http://www.freertos.org> for more information and the latest source code.

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

- Ethernet MAC
- System controller services
- MMUART
- GPIO
- SPI

Requirements and Details

Table 1 • Reference Design Requirements and Details

Reference Design Requirements and Details	Description
Hardware Requirements	
SmartFusion2 Development Kit <ul style="list-style-type: none">• 12 V adapter• FlashPro4 programmer• USB A to Mini-B cable	Rev C or later
RJ45 cable	-
Host PC or Laptop	Windows 64-bit Operating System
Software Requirements	
Libero® System-on-Chip (SoC) for viewing the design files <ul style="list-style-type: none">• FlashPro Programming Software v11.3• SoftConsole v3.4	11.3
Host PC Drivers	USB to UART drivers
One of the following serial terminal emulation programs: <ul style="list-style-type: none">• HyperTerminal• TeraTerm• PuTTY	-
Browser	Mozilla Firefox version 24 or later Internet Explorer version 8 or later

Demo Design

Introduction

The demo design files are available for download from the Microsemi® website:
http://soc.microsemi.com/download/rsc/?f=SF2_Secure_Webserver_tcp_DF

The demo design files include:

- The Libero SoC hardware project with the SoftConsole firmware project
- SPI flash loader
- STAPL programming file
- readme.txt file

Figure 3 shows the top-level structure of the design files. For further details, refer to the readme.txt file.

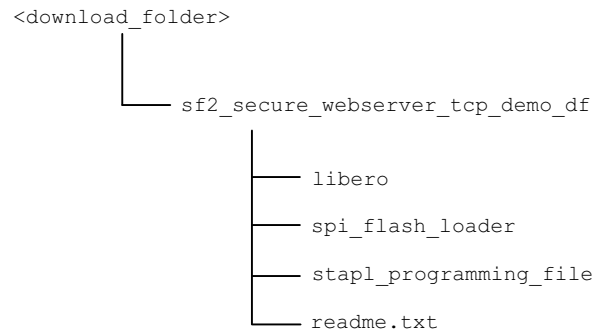


Figure 3 • Demo Design Files Top-Level Structure

Demo Design Features

The demo has the following options:

- Blinking LEDs
- HyperTerminal Display
- SmartFusion2 Google Search

Demo Design Description

The demo design is implemented using an SGMII PHY interface by configuring the TSEMAC for the ten-bit interface (TBI) operation. The TBI is routed through the FPGA fabric onto the SERDES I/Os using the CORETBITOEPCS (TBI to the external physical coding sublayer Soft IP) interface. For more information on the TSEMAC TBI interface, refer to the [SmartFusion2 Microcontroller Subsystem User Guide](#).

The demo design comprises:

- [Libero SoC Hardware Project](#)
- [SoftConsole Firmware Project](#)
- [SPI Flash Memory Loader](#)

Libero SoC Hardware Project

Figure 4 shows the Libero SoC hardware design implementation for this demo design.

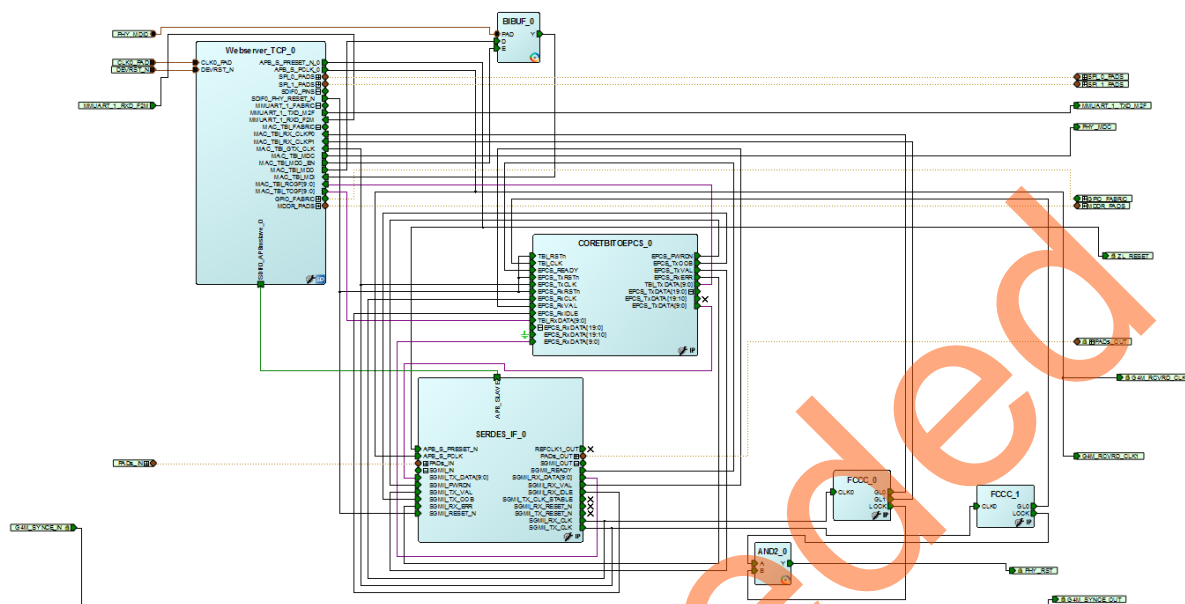


Figure 4 • Libero Top-Level Design

The Libero Hardware project uses the following SmartFusion2 MSS resources and IPs:

1. **TSEMAC TBI** interface.
2. **MMUART_1** for RS-232 communications on the development kit.
3. **SPI**: Used to configure the ZL30362 clock synthesizer (supplies reference clock to **SERDESIF**).
4. General purpose input and output (GPIO): Interfaces with the light emitting diodes (LEDs).
5. High speed serial interface (SERDESIF) **SERDES_IF** IP, configured for **SERDESIF_0** SGMII lane3 as shown in Figure 5.

For more information on high speed serial interfaces, refer to the [SmartFusion2 SoC FPGA High Speed Serial Interfaces User Guide](#).

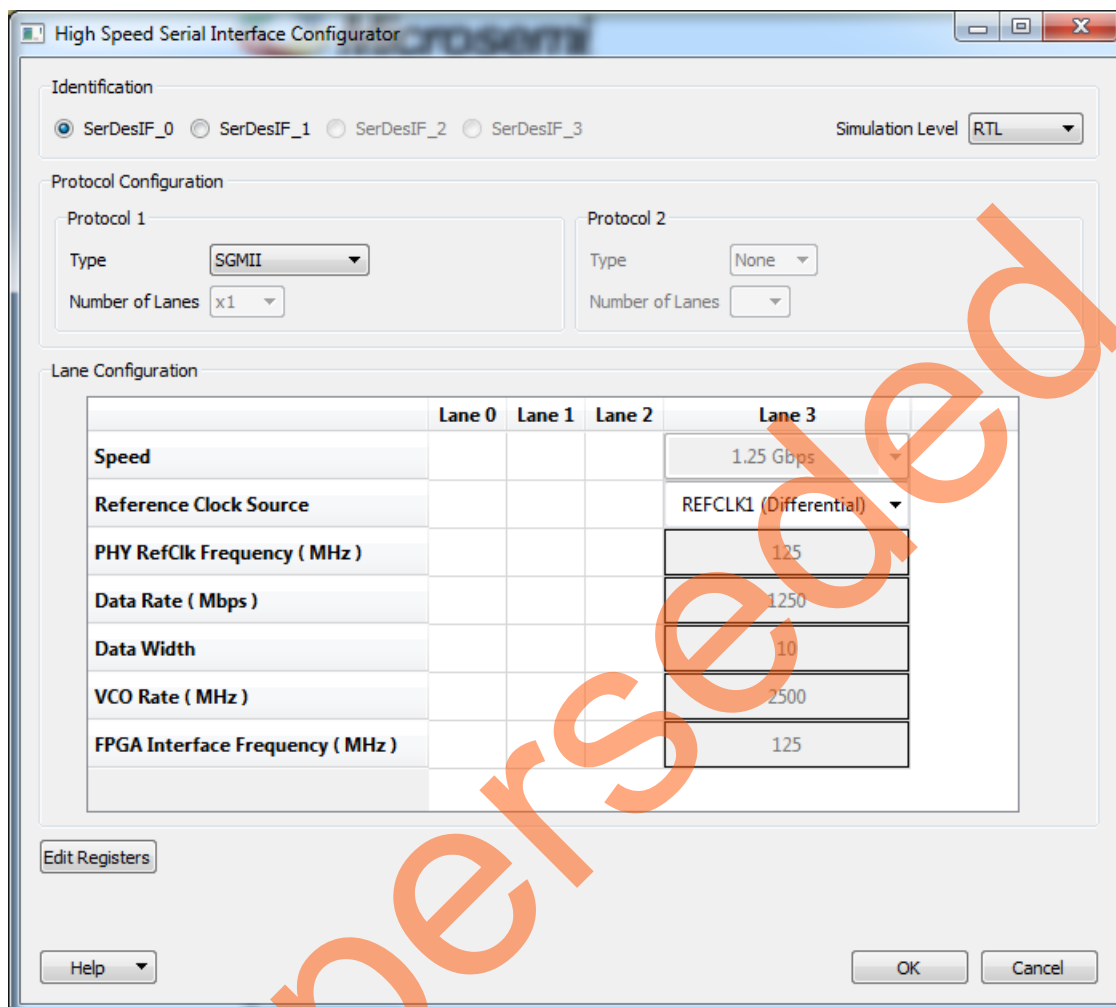


Figure 5 • High Speed Serial Interface Configurator Window

6. **CORETBITOEPCS** bus interface: Acts as a bridge between the TBI and the EPCS interfaces.
7. Cryptographic system controller services: To implement TLS/SSL protocol.

Package Pin Assignments

Package pin assignments for LEDs and PHY interface signals are shown in Table 2 and Table 3 on page 12.

Table 2 shows the port names for the package pins.

Table 2 • LED to Package Pins Assignments

Port Name	Package Pin
LED_1	A18
LED_2	B18
LED_3	D18
LED_4	E18
LED_5	A20
LED_6	D20
LED_7	E20
LED_8	B20

Table 3 shows the port names and directions for the package pins.

Table 3 • PHY Interface Signals to Package Pins Assignments

Port Name	Direction	Package Pin
PHY_MDC	Output	N5
PHY_MDIO	Input	R7
PHY_RST	Output	N4

SoftConsole Firmware Project

Invoke the SoftConsole project using the **Write Application Code** option available under **Develop Firmware** in the Libero SoC **Design Flow** window. Refer to the [Libero SoC User Guide](#) for more information.

The following stacks are used for this demo design:

- **PolarSSL** library version 1.2.8 (<https://polarssl.org/>)
- **lwIP TCP/IP** stack version 1.4.1 (www.sics.se/~adam/lwIP/)
- **FreeRTOS** (www.freertos.org)

Figure 1 on page 6 shows the block diagram of the Secure Webserver application on the SmartFusion2 devices used in this demo design.

Figure 6 shows an example SoftConsole software directory structure of the demo design.

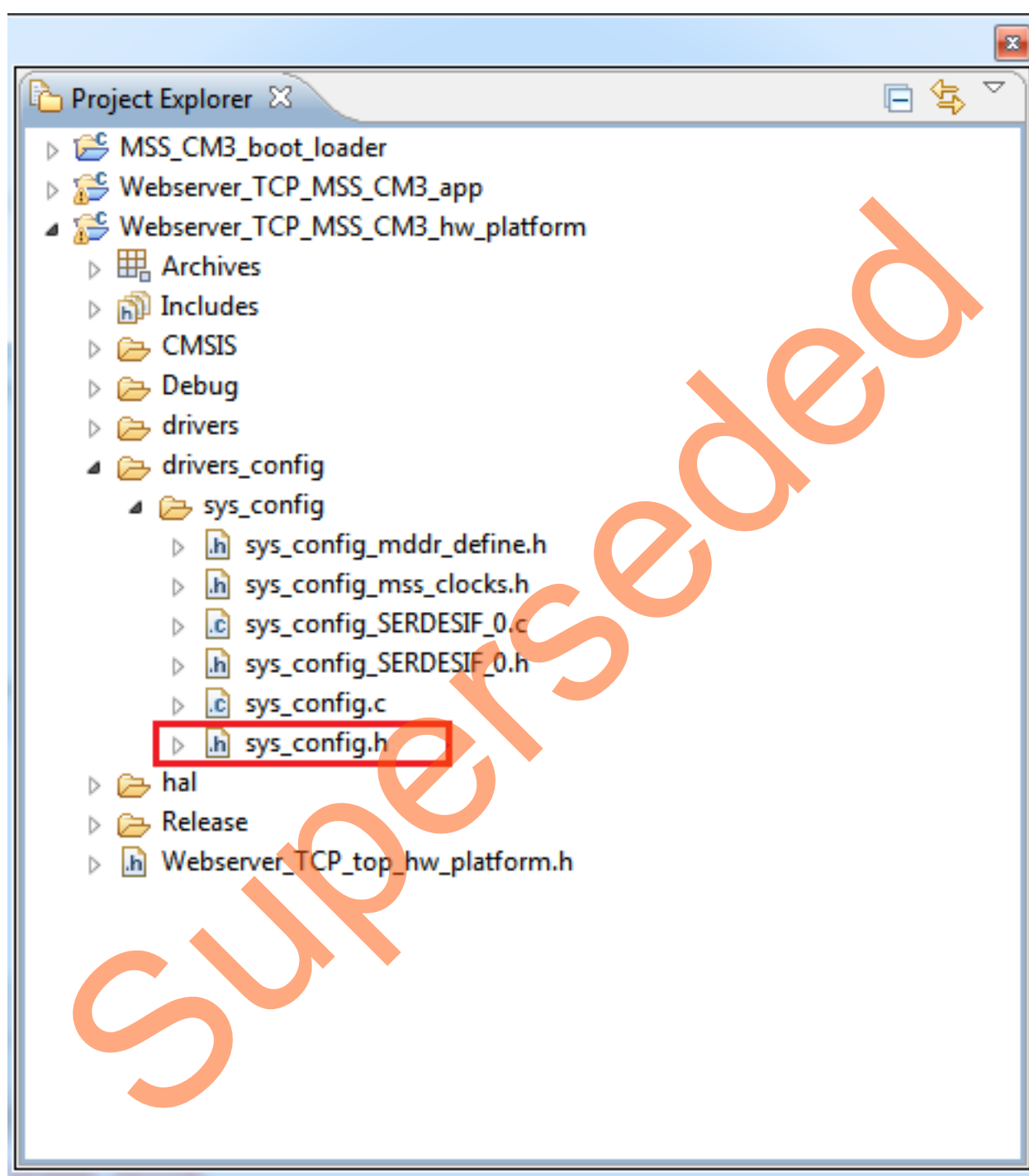


Figure 6 • Example SoftConsole Project Explorer Window

The SoftConsole workspace consists of three projects.

1. MSS_CM3_boot_loader
 - Receives the secure Webserver application image from the host PC using UART interface and stores the image to the SPI flash memory.
 - Copies the application image from the SPI flash memory to the DDR memory and runs the image from the DDR memory.

This project loads the Secure Webserver application image from the SPI flash memory to the DDR memory and runs the application from the DDR memory. Running the application from the DDR memory is required, if the application image does not fit into the eNVM memory. This demo runs the secure Webserver application image from the DDR memory with cache controller enabled. See the [SmartFusion2 SoC FPGA Code Shadowing from SPI Flash to DDR Memory Demo User Guide](#) for more information on running the application image from the DDR memory.

2. Webserver_TCP_MSS_CM3_app

This project contains the Secure Webserver application implementation using PolarSSL, LWIP, and FreeRTOS. If the Libero hardware project is regenerated, set the `#define MSS_SYS_MDDR_CONFIG_BY_CORTEX` macro to 0 in `sys_config.h` file. [Figure 6 on page 13](#) shows the `sys_config.h` file selected.

The advanced encryption standard (AES) and non-deterministic random bit generator (NRBG) system services are used to implement the Secure Webserver application. The following macros need to be commented for using the PolarSSL AES and software NRBG algorithms.

Table 4 • Macros to Enable or Disable System Controller Services

System Service	Macro	Macro Location
AES	<code>#define HW_AES 1</code>	<code><download_folder>\sf2_secure_webserver_tcp_demo_dfl\libero\SoftConsole\Webserver_TCP_MSS_CM3\Webserver_TCP_MSS_CM3_app\polarssl-1.2.8\include\polarssl\aes.h</code>
NRBG	<code>#define HW_NRBG 1</code>	<code><download_folder>\sf2_secure_webserver_tcp_demo_dfl\libero\SoftConsole\Webserver_TCP_MSS_CM3\Webserver_TCP_MSS_CM3_app\polarssl-1.2.8\include\polarssl\ssl.h</code>

Note: The system services AES and NRBG are supported for data security enabled SmartFusion2 device like M2S050TS. If the SmartFusion2 device is not data security enabled, disable the macros mentioned in [Table 4](#).

3. Webserver_TCP_MSS_CM3_hw_platform

This project 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 the `Webserver_TCP_MSS_CM3_app` application project. The contents of this folder get over-written every time the root design is regenerated in the Libero SoC software.

TLS/SSL Protocol Implementation using PolarSSL Library

The TLS/SSL protocol is divided into the following two protocol layers:

- Handshake protocol layer
- Record protocol layer

Handshake Protocol Layer

This layer consists of the following sub protocols:

- **Handshake:** Used to negotiate session information between the server and the client. The session information includes session ID, peer certificates, the cipher spec, the compression algorithm, and a shared secret code that is used to generate required keys.
- **Change Cipher spec:** Used to change the key used for encryption between the client and the server. The key is computed from the information exchanged during the client-server handshake.
- **Alert:** Alert messages are generated during the client-server handshake to report an error or a change in status to the peer.

Figure 7 shows the overview of the TLS/SSL handshake procedure. Refer to <http://tools.ietf.org/html/rfc5246> for detailed information on handshake protocol, record protocol and cryptographic algorithms.

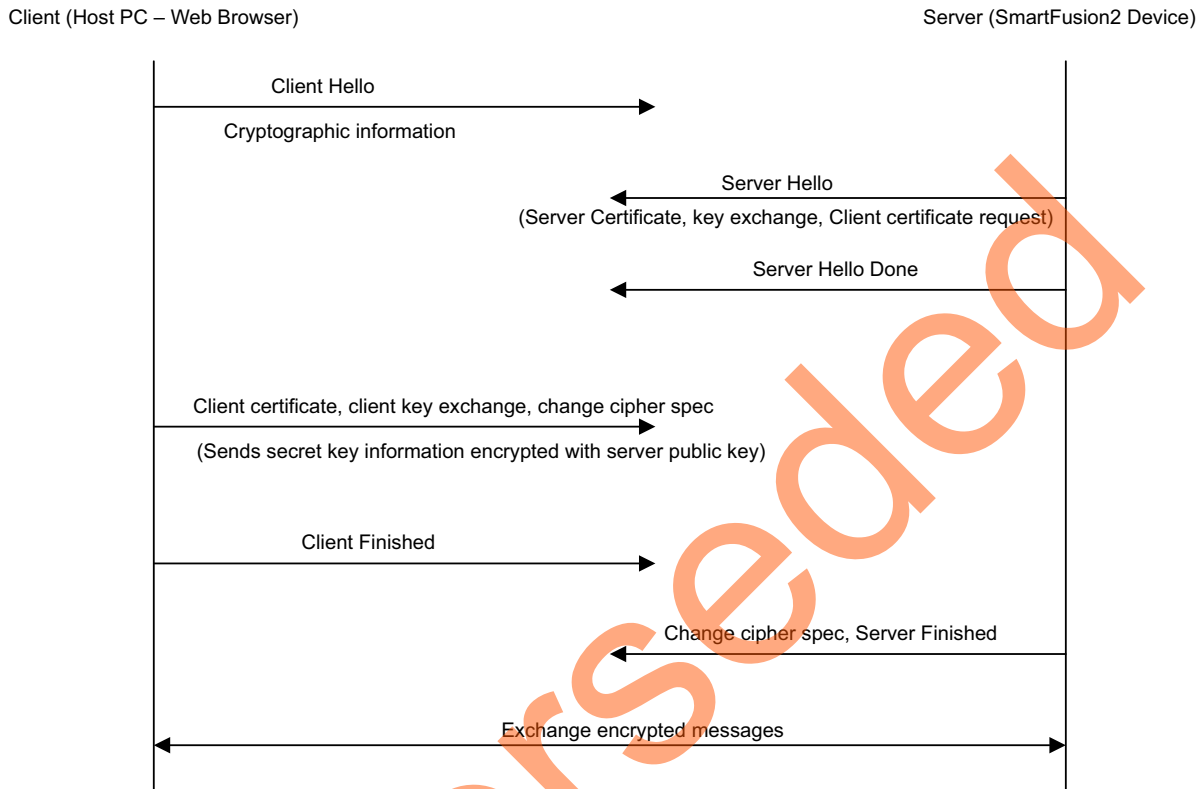


Figure 7 • TLS/SSL Handshake Procedure

Record Protocol Layer

The record protocol receives and encrypts data from the application and transfers to the transport layer. The record protocol fragments the received data to a size appropriate to the cryptographic algorithm and optionally compresses the data. The protocol applies a MAC or HMAC and encrypts or decrypts the data using the information negotiated during the handshake protocol.

SPI Flash Memory Loader

SPI-Flash memory loader (m2s_spi_flash_loader.exe) is an executable program that transfers the secure Webserver application image (Webserver_TCP_MSS_CM3_app.bin) from the host PC to the SPI flash memory of the SmartFusion2 Development Kit board. The m2s_spi_flash_loader.exe file is executed from the command prompt. It is located at:

<download_folder>\sf2_secure_webserver_tcp_demo_dfs\spi_flash_loader.

The syntax is:

```
m2s_spi_flash_loader.exe <*.bin> <COM Port number>
```

Arguments:

- *.bin file (secure Webserver application image Webserver_TCP_MSS_CM3_app.bin)
- COM Port number

Generating *.bin File for Secure Webserver Application Image

1. Invoke the SoftConsole workspace using the **Write Application Code** option available under **Develop Firmware** in the Libero SoC **Design Flow** window.
2. Build the **Webserver_TCP_MSS_CM3_app** project from SoftConsole workspace.
3. Double-click `Bin_File_Generator.bat` file located at `<download_folder>\sf2_secure_webserver_tcp_demo_df\libero\SoftConsole\Webserver_TCP_MSS_CM3\Webserver_TCP_MSS_CM3_app` and copy the `Webserver_TCP_MSS_CM3_app.bin` file to the `spi_flash_loader` folder located at:
`<download_folder>\sf2_secure_webserver_tcp_demo_df\spi_flash_loader`

Setting up the Demo Design

1. Connect the FlashPro4 programmer to the J59 connector of SmartFusion2 Development Kit board.
2. Install the USB driver.
3. For serial terminal communication through the FTDI mini USB cable, install the FTDI D2XX driver.
Download the drivers and installation guide from:
www.microsemi.com/soc/documents/CDM_2.08.24_WHQL_Certified.zip

4. Connect the host PC to the J24 connector using the USB A to mini-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 8](#).

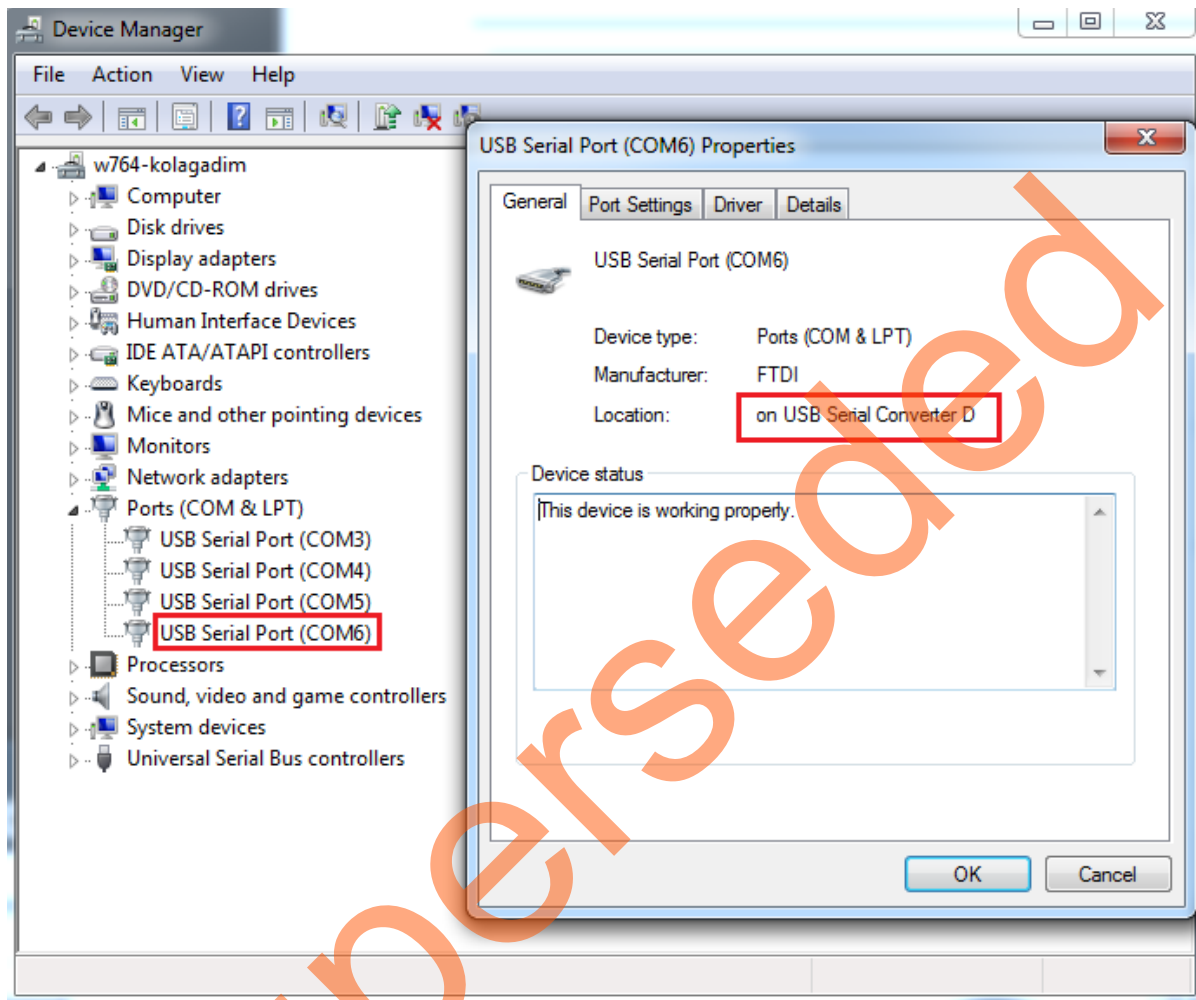


Figure 8 • Device Manager Window Showing the USB-to-Serial Communication Port

5. Connect the jumpers on the SmartFusion2 Development Kit board as shown in [Table 5](#). For information on jumper locations, refer to "[Appendix 2: Jumper Locations](#)" on page 32.

Caution: Before making the jumper connections, switch off the power supply switch, SW7.

Table 5 • SmartFusion2 Development Kit Jumper Settings

Jumper Number	Settings	Notes
J70, J93, J94, J117, J123, J142, J157, J160, J167, J225, J226, J227	1-2 closed	These are the default jumper settings of the Development Kit board. Make sure that these jumpers are set properly.
J2	1-3 closed	
J23	2-3 closed	
J129, J133	2-3 closed	These jumpers are required for the ZL30362 configuration. All these jumpers are not set by default and must be set manually.
J20, J21, J22, J25	1-2 closed	
J30	2-3 closed	

6. In the SmartFusion2 Development Kit, connect the power supply to the J18 connector.
7. This design example can run in both Static IP and Dynamic IP modes. By default, programming files are provided for dynamic IP mode.
 - For static IP, connect the host PC to the J4 connector of the SmartFusion2 Development Kit Board using an RJ45 cable.
 - For dynamic IP, connect any one of the open network ports to the J4 connector of the SmartFusion2 Development Kit Board using an RJ45 cable.

Board Setup Snapshot

Snapshots of the SmartFusion2 Development Kit board with all the setup made is given in "[Appendix 1: Board Setup for Running the Secure Webserver](#)" on page 31.

Running the Demo Design

1. Download the demo design from:
http://soc.microsemi.com/download/rsc/?f=SF2_Secure_Webserver_tcp_DF
2. Switch ON the SW7 power supply switch.
3. 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: 57600
- Eight data bits
- One stop bit
- No Parity
- No flow control

For information on configuring the serial terminal emulation programs, refer to the [Configuring Serial Terminal Emulation Programs Tutorial](#).

4. Launch the **FlashPro** software.
5. Click **New Project**.

6. In the **New Project** window, type the project name as shown in [Figure 9](#).

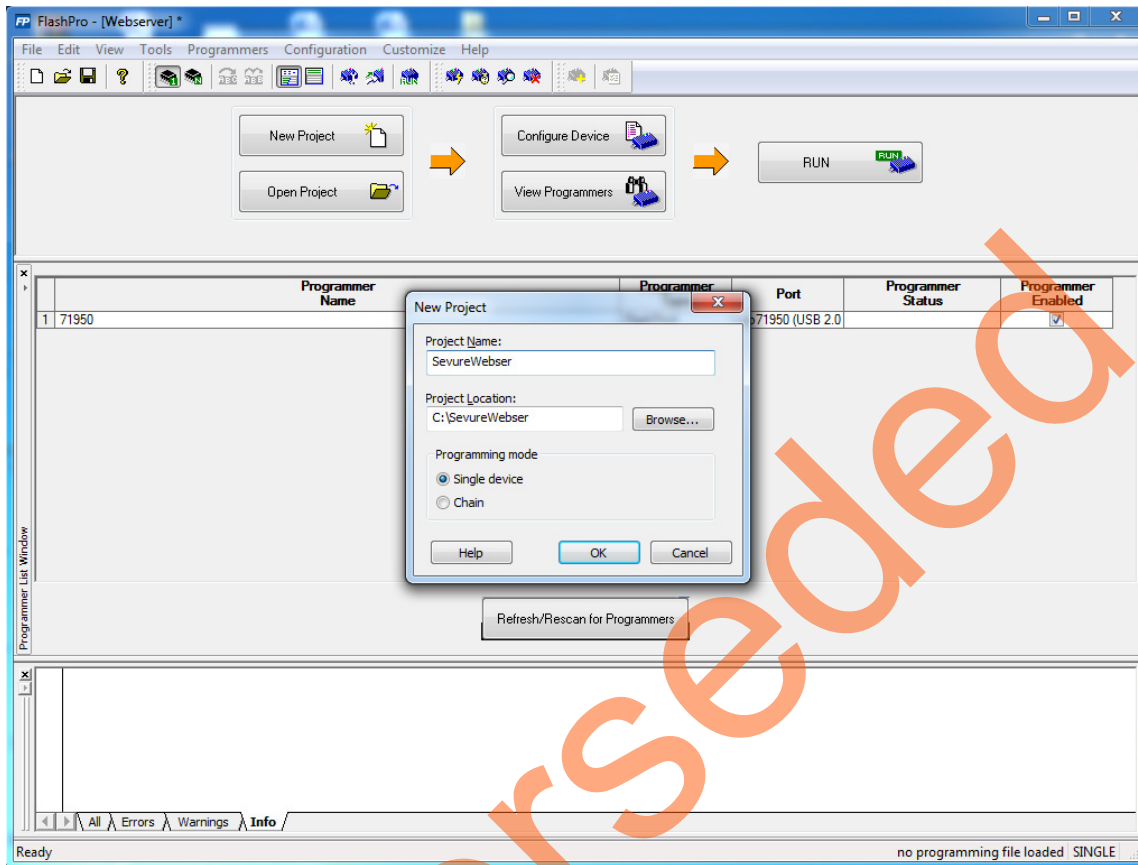


Figure 9 • FlashPro New Project

7. Click **Browse** and navigate to the location where the project is required to be saved.
8. Select **Single device** as the **Programming mode**.
9. Click **OK** to save the project.
10. Click **Configure Device**.

11. Click **Browse** and navigate to the location where the `secure_Webserver_tcp_top.stp` file is located and select the file. The default location is: `<download_folder>\sf2_secure_Webserver_tcp_demo_df\stapl_programming_file\secure_Webserver_tcp_top.stp`
The required programming file is selected and is ready to be programmed in the device as shown in Figure 10.

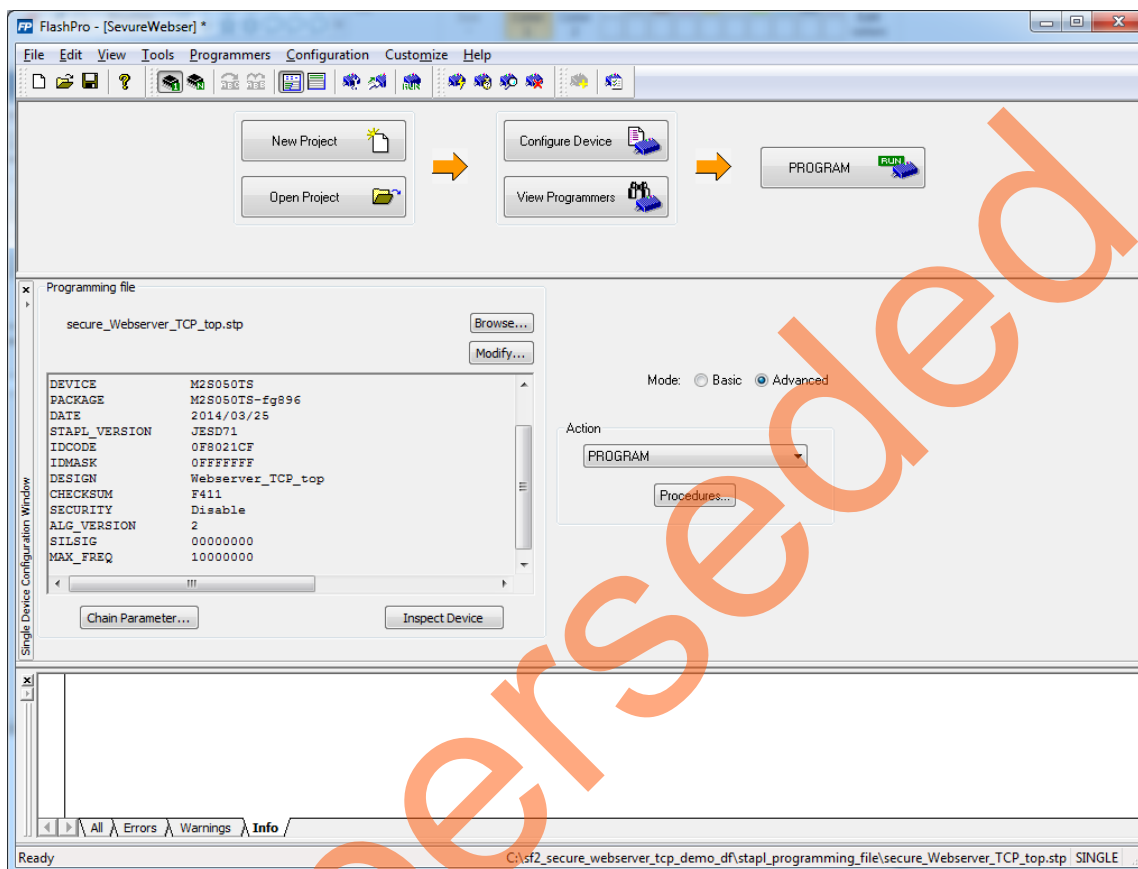


Figure 10 • FlashPro Project Configured

12. Click **PROGRAM** to start programming the device. Wait until a message is displayed, indicating that the program has passed.

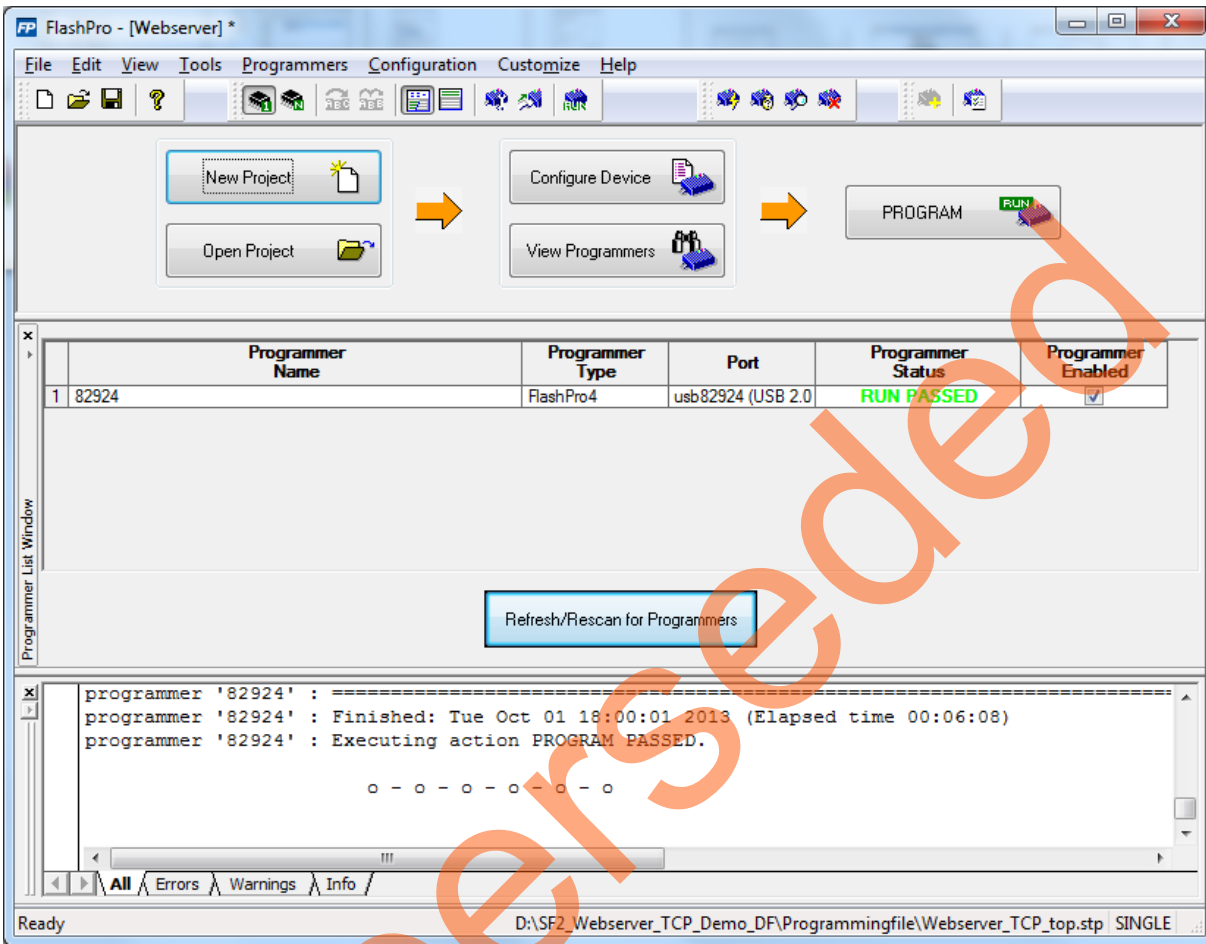


Figure 11 • FlashPro Program Passed

Note: The demo can be run in static and dynamic modes. To run the design in Static IP mode, follow the steps mentioned in the "Appendix 3: Running the Design in Static IP Mode" on page 33.

13. Power cycle the SmartFusion2 Development Kit board.

Figure 12 shows the serial terminal emulation program options.

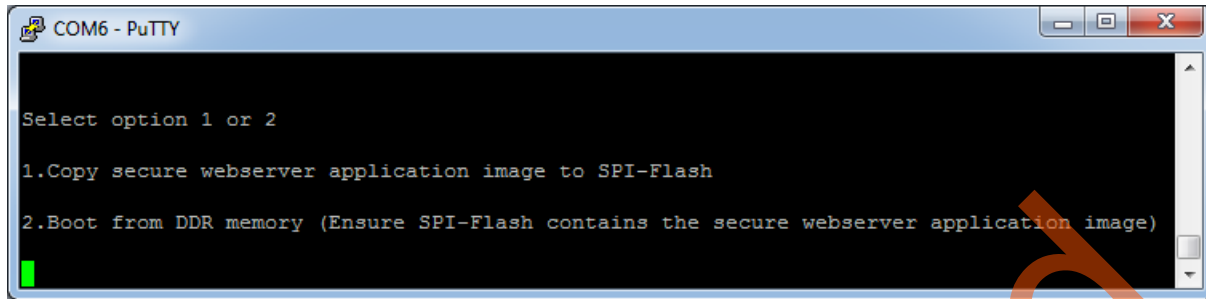


Figure 12 • User Options

14. Type 1 to copy the Secure Webserver application image to the SPI flash memory using the Host PC SPI flash loader program.

15. Close the serial terminal emulation program window.

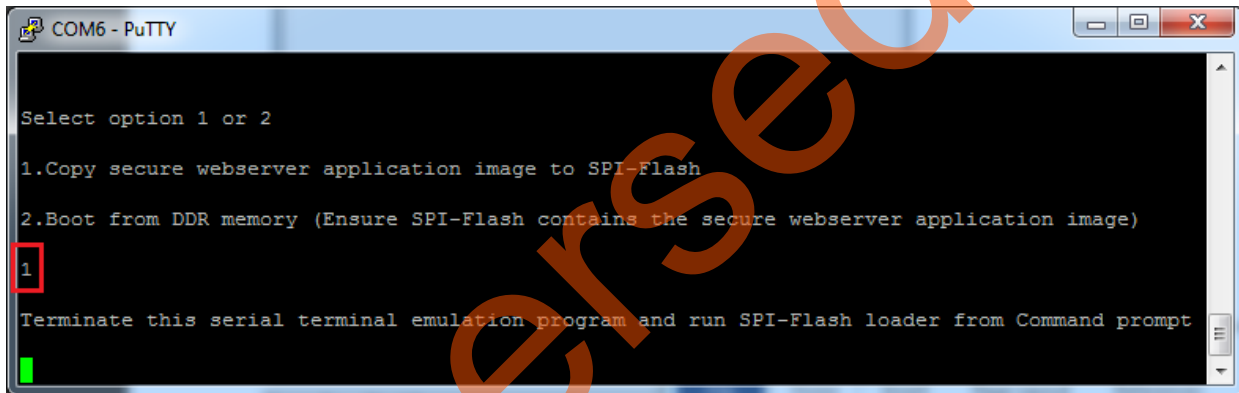


Figure 13 • Selecting Option 1

16. Open the command prompt in the Host PC.

17. Navigate to the directory, where the SPI flash memory loader (m2s_spi_flash_loader.exe) is located. The default location is:

`<download_folder>\sf2_secure_webserver_tcp_demo_df\spi_flash_loader.`

18. Execute the m2s_spi_flash_loader.exe file and launch the SPI flash memory loader to copy the secure webserver application image to the SPI flash memory of the SmartFusion2 device.

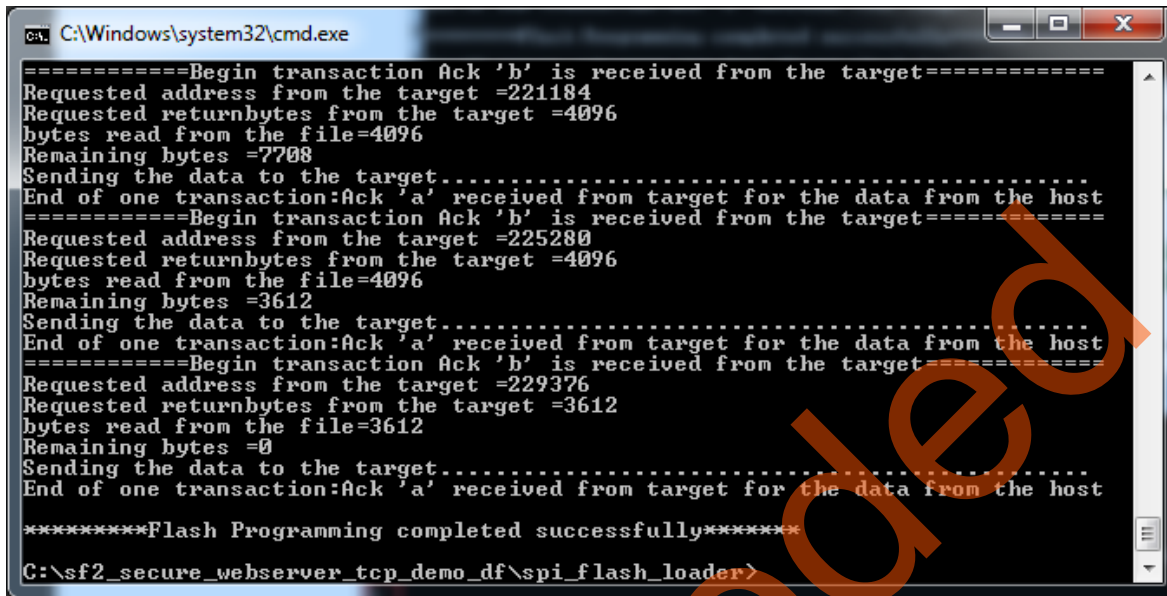
Use the following command to execute the SPI flash loader from command prompt as shown in Figure 14.

`m2s_spi_flash_loader.exe Webserver_TCP_MSS_CM3_app.bin 6`
where 6 is the COM port number.



Figure 14 • Command Syntax

On successful completion of copying the application image, the following message is displayed in the command prompt Flash Programming completed successfully.



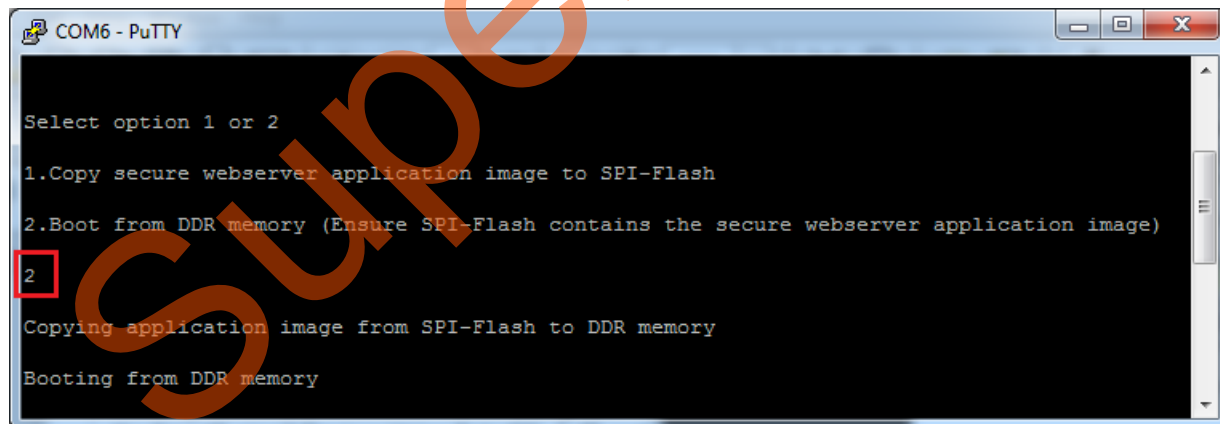
```

C:\Windows\system32\cmd.exe
=====Begin transaction Ack 'b' is received from the target=====
Requested address from the target =221184
Requested returnbytes from the target =4096
bytes read from the file=4096
Remaining bytes =7708
Sending the data to the target.....
End of one transaction:Ack 'a' received from target for the data from the host
=====Begin transaction Ack 'b' is received from the target=====
Requested address from the target =225280
Requested returnbytes from the target =4096
bytes read from the file=4096
Remaining bytes =3612
Sending the data to the target.....
End of one transaction:Ack 'a' received from target for the data from the host
=====Begin transaction Ack 'b' is received from the target=====
Requested address from the target =229376
Requested returnbytes from the target =3612
bytes read from the file=3612
Remaining bytes =0
Sending the data to the target.....
End of one transaction:Ack 'a' received from target for the data from the host
*****Flash Programming completed successfully*****
C:\sf2_secure_webserver_tcp_demo_df\spl_flash_loader>

```

Figure 15 • Programmed SPI Flash Successfully

19. Start the serial terminal emulation program as mentioned in [Step 3](#), and power cycle the SmartFusion2 Development Kit board.
20. Type 2 to select Boot from the DDR memory. The Cortex-M3 processor copies the application image from the SPI-Flash memory to the DDR memory and runs the Secure Webserver application image from the DDR memory as shown in [Figure 16](#).



```

COM6 - PuTTY

Select option 1 or 2

1.Copy secure webserver application image to SPI-Flash
2.Boot from DDR memory (Ensure SPI-Flash contains the secure webserver application image)
2
Copying application image from SPI-Flash to DDR memory
Booting from DDR memory

```

Figure 16 • Selecting Option 2

21. A welcome message with the dynamic IP address is displayed in the serial terminal emulation program as shown in Figure 17.

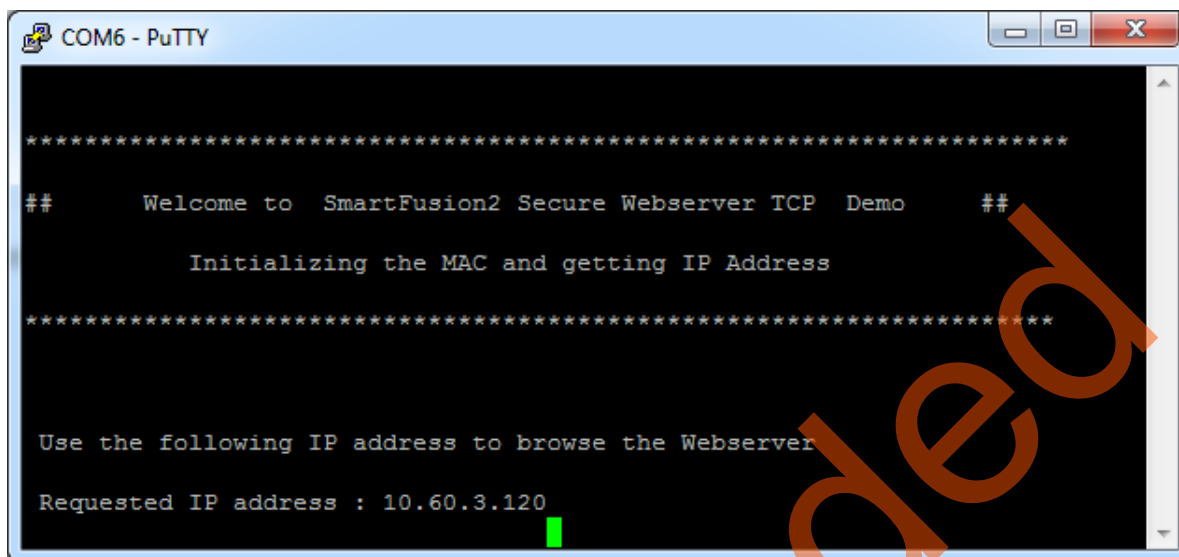


Figure 17 • PuTTY with IP Address

22. The IP address displayed on PuTTY should be entered in the address bar of the browser to run the Secure Webserver. If the IP address is 10.60.3.120, type <https://10.60.3.120> in the address bar of the browser. This demo supports both Microsoft Internet Explorer and Mozilla Firefox browsers.

Running the Secure Webserver Demo with Microsoft Internet Explorer

1. Open the Microsoft Internet Explorer and type the URL (for example, <https://10.60.3.120>) in the address bar. The browser shows a warning message as shown in Figure 18.

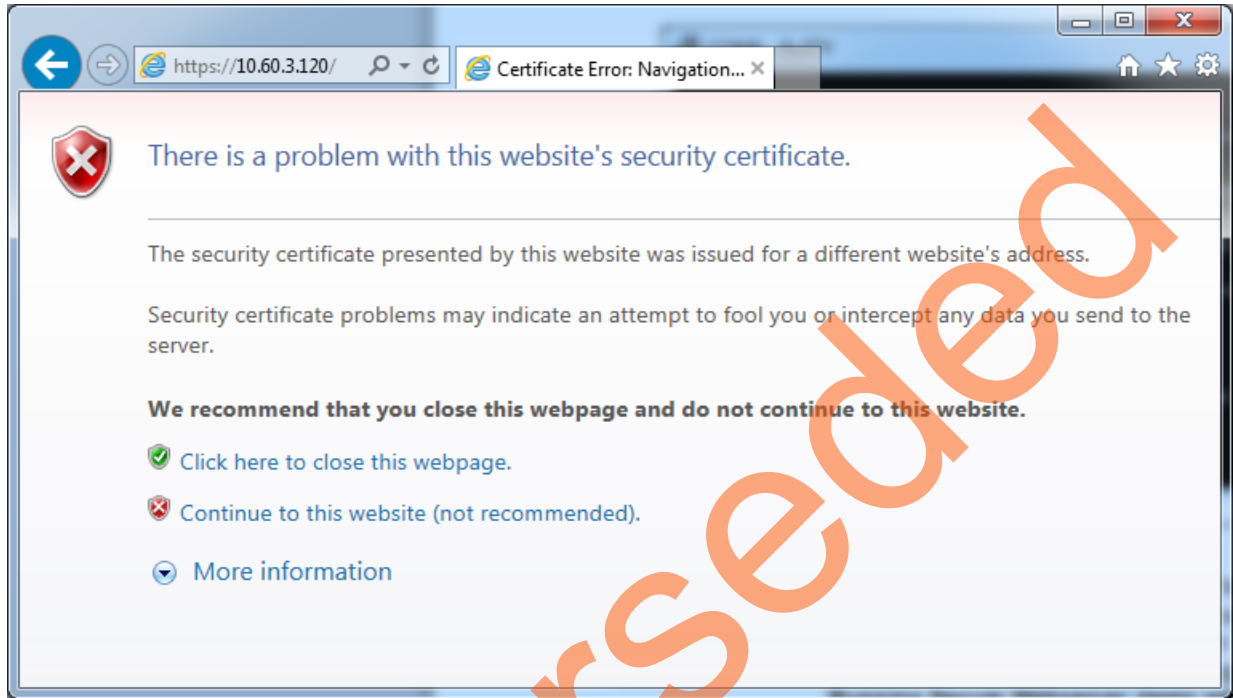


Figure 18 • Microsoft Internet Explorer showing Certificate Error Warning Message

2. Click **Continue to this website (not recommended)** to start secure communication with the Webserver. The Microsoft Internet Explorer displays the main menu of the Secure Webserver as shown in Figure 19.

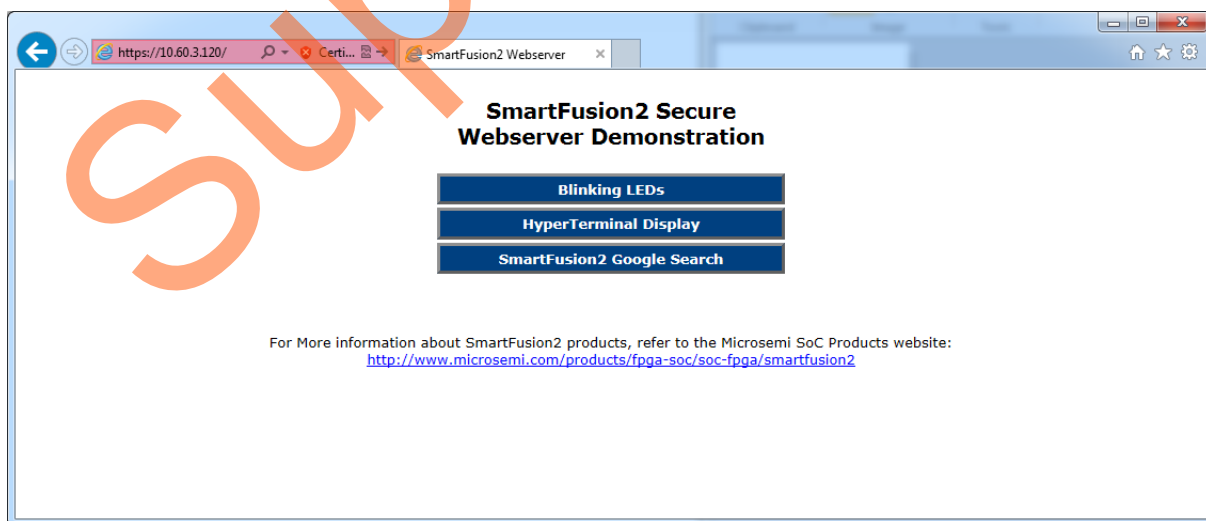


Figure 19 • Main Menu of Secure Webserver in Internet Explorer

Running the Secure Webserver Demo with Mozilla Firefox

1. Open the Mozilla Firefox browser and type the URL (for example, <https://10.60.3.120>) in the address bar. The browser shows a warning message as shown in Figure 20.

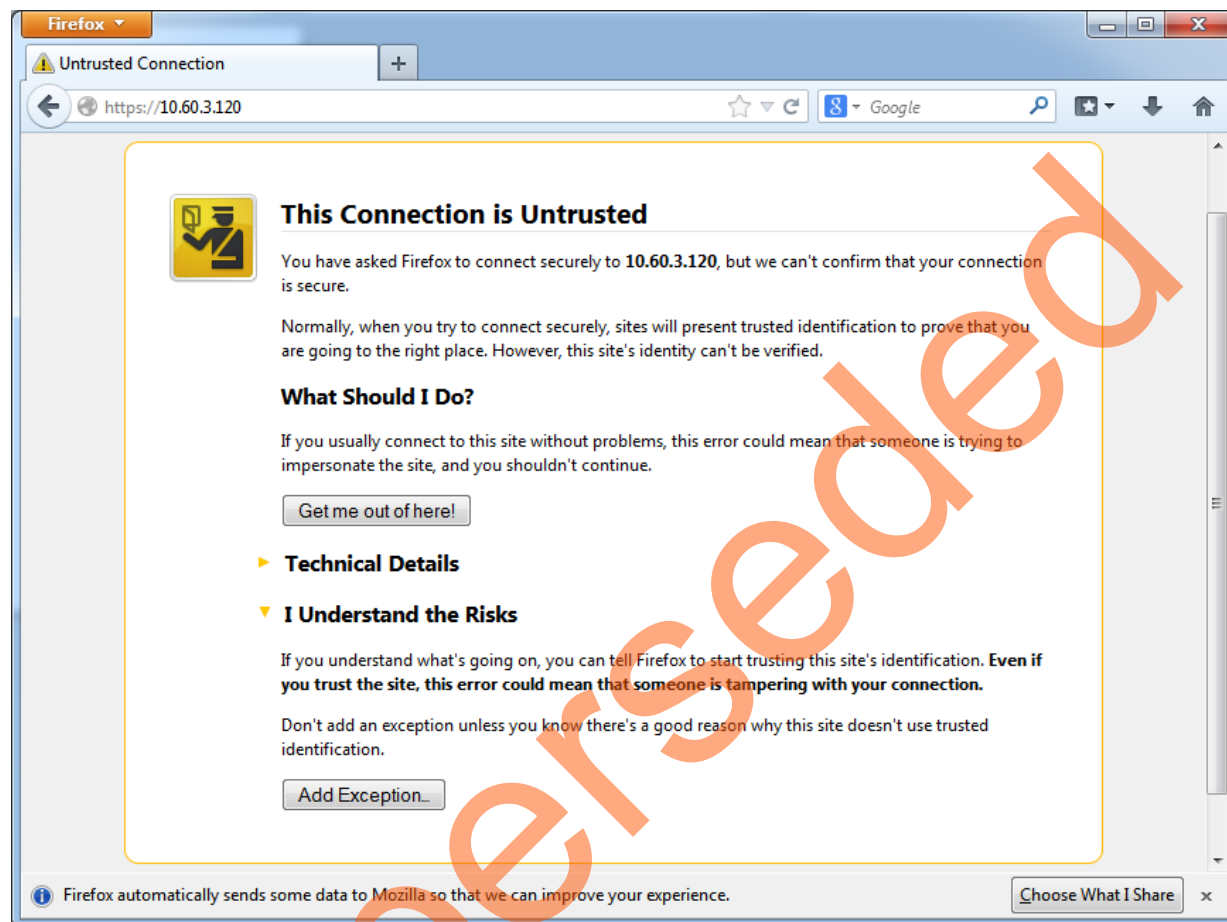


Figure 20 • Mozilla Firefox showing Warning Message

2. Select **I Understand the Risks** and click **Add Exception....**

3. Click **Confirm Security Exception** in **Add Security Exception** window as shown in Figure 21, to start secure communication with the Webserver.

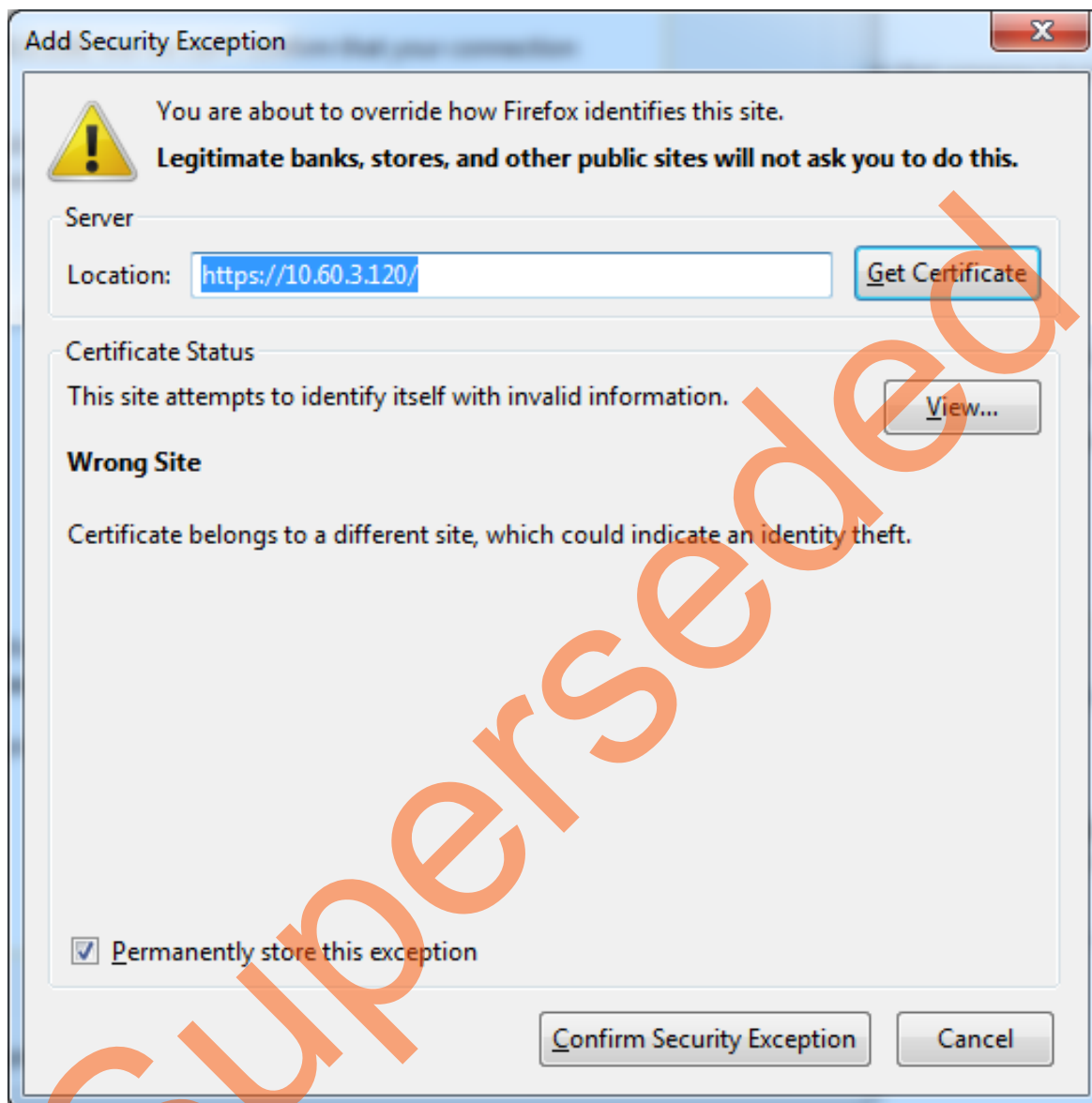


Figure 21 • Add Security Exception Window

Note: Adding security exception for the IP Address is required for first-time browsing only.

The Mozilla Firefox browser displays the main menu as shown in Figure 22.

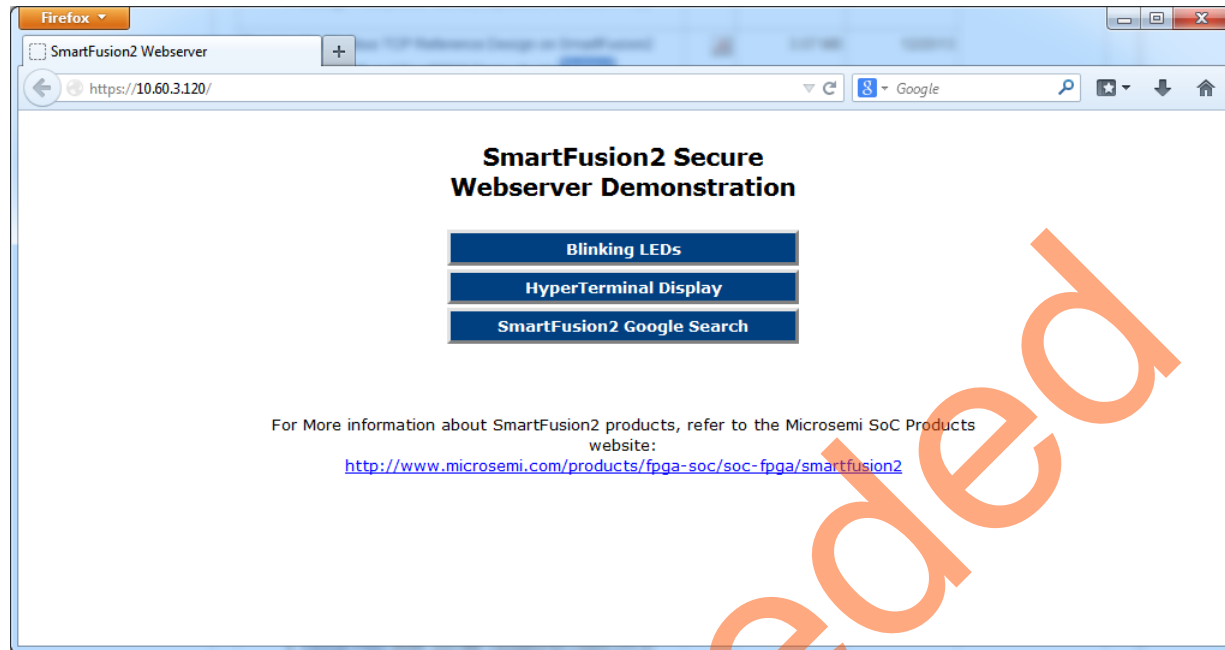


Figure 22 • Main Menu of the Secure Webserver in Mozilla Firefox

The main menu has the following options:

- Blinking LEDs
- HyperTerminal Display
- SmartFusion2 Google Search

Note: These options can be verified using either Microsoft Internet Explorer or Mozilla Firefox web browsers. In this demo, the options are demonstrated using Mozilla Firefox web browser.

Blinking LEDs

1. Click **Blinking LED's** on the main menu. You can observe a running LED pattern on the SmartFusion2 board. The webpage gives an option to enter the values to blink the LEDs manually as shown in Figure 23.

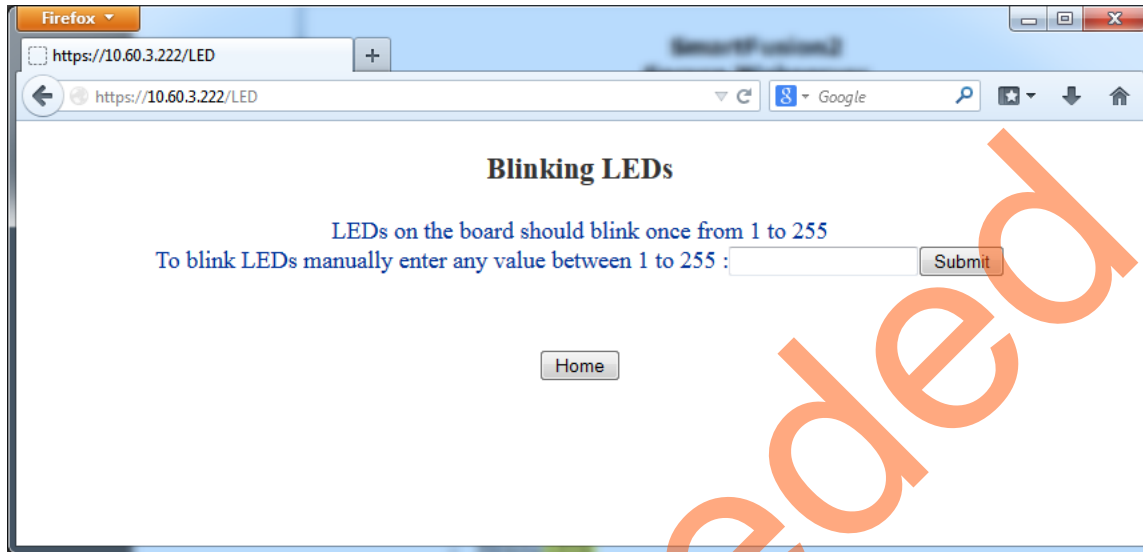


Figure 23 • Blinking LEDs Page

2. Enter any number between 1-255 to lit the LEDs manually. For example, if you enter 1, LED1 blinks. If you enter 255, all the eight LEDs blink.
3. Click **Home** to go back to the main menu.

HyperTerminal Display

1. Click **HyperTerminal Display** on the main menu. Figure 24 shows a webpage that gives an option to enter a string value.

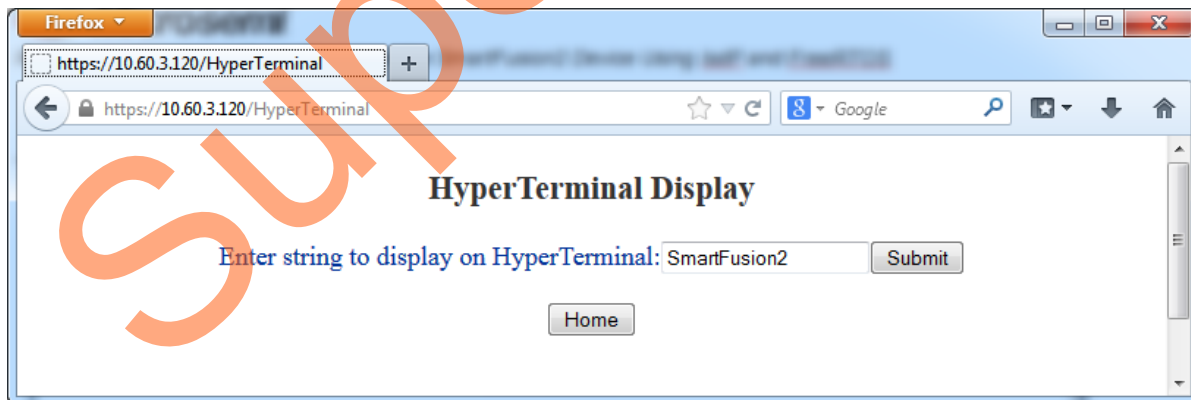


Figure 24 • HyperTerminal Display Page

The entered string is displayed on PuTTY as shown in Figure 25.

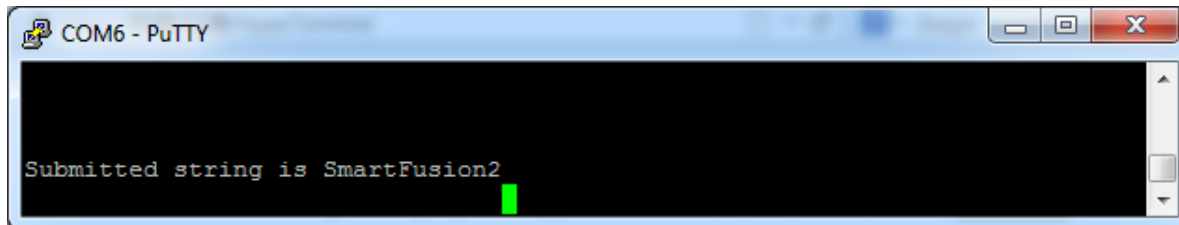


Figure 25 • String Display on PuTTY

2. Click **Go Back One Page** (arrow button) or **Home** to go back to the main menu.

SmartFusion2 Google Search

1. Click **SmartFusion2 Google Search** on the main menu.

Note: Internet connection is required with proper access rights to get to the SmartFusion2 Google Search page.

Figure 26 shows a webpage with Google search.

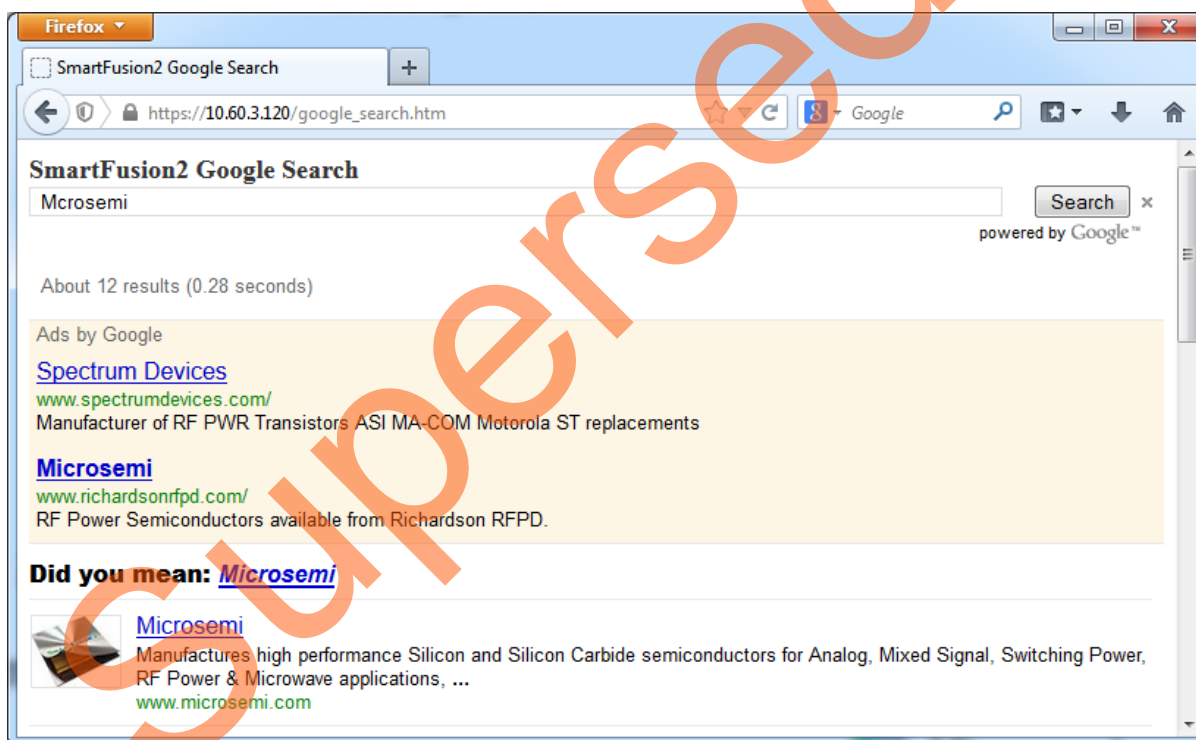


Figure 26 • SmartFusion2 Google Search Page

2. Click **Home** to go back to the main menu.

Appendix 1: Board Setup for Running the Secure Webserver

Figure 27 shows the board setup for running the demo on the Development Kit Board.

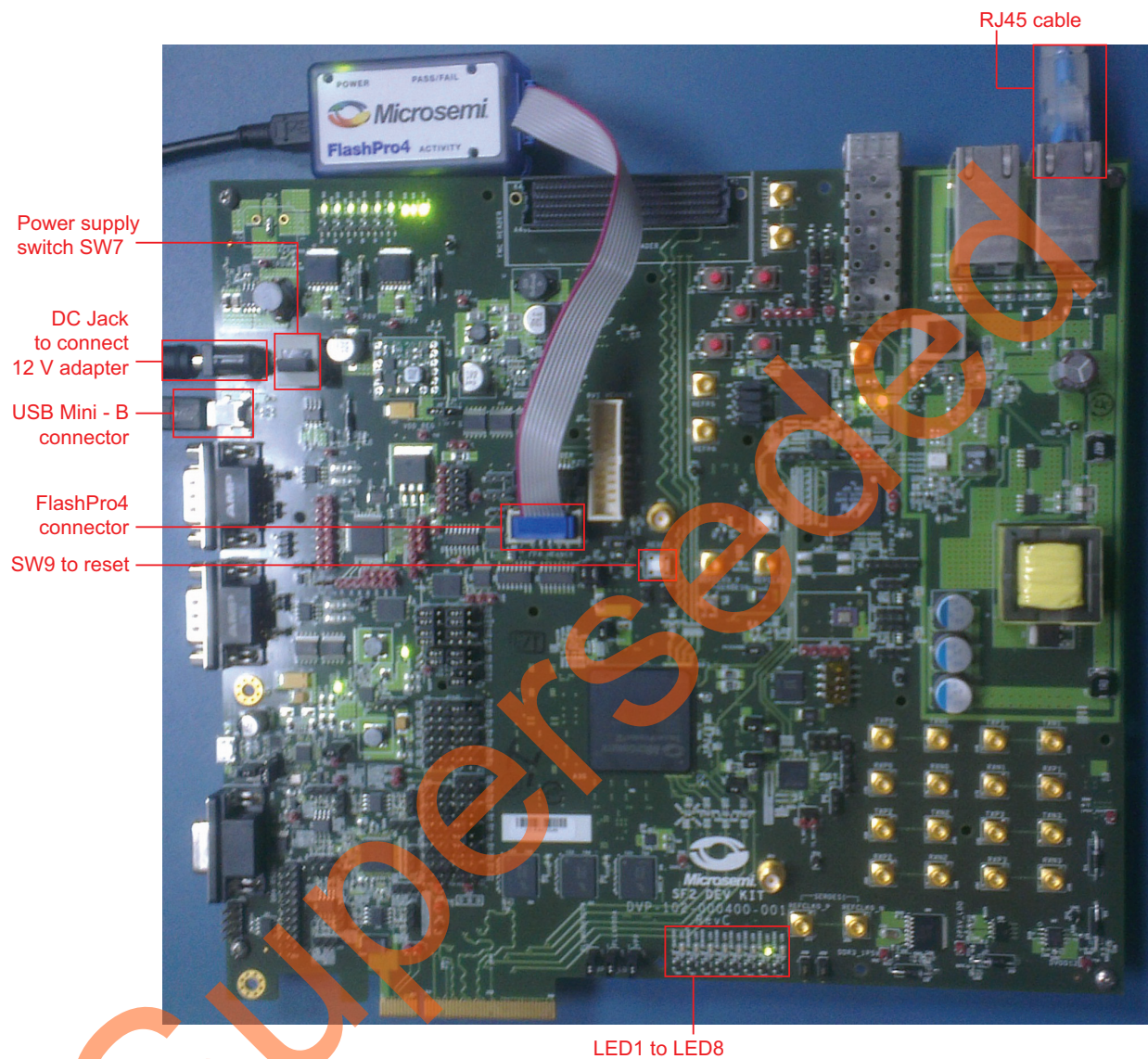


Figure 27 • SmartFusion2 Development Kit Setup

Figure 28 shows the jumper locations in the SmartFusion2 Development Kit board.

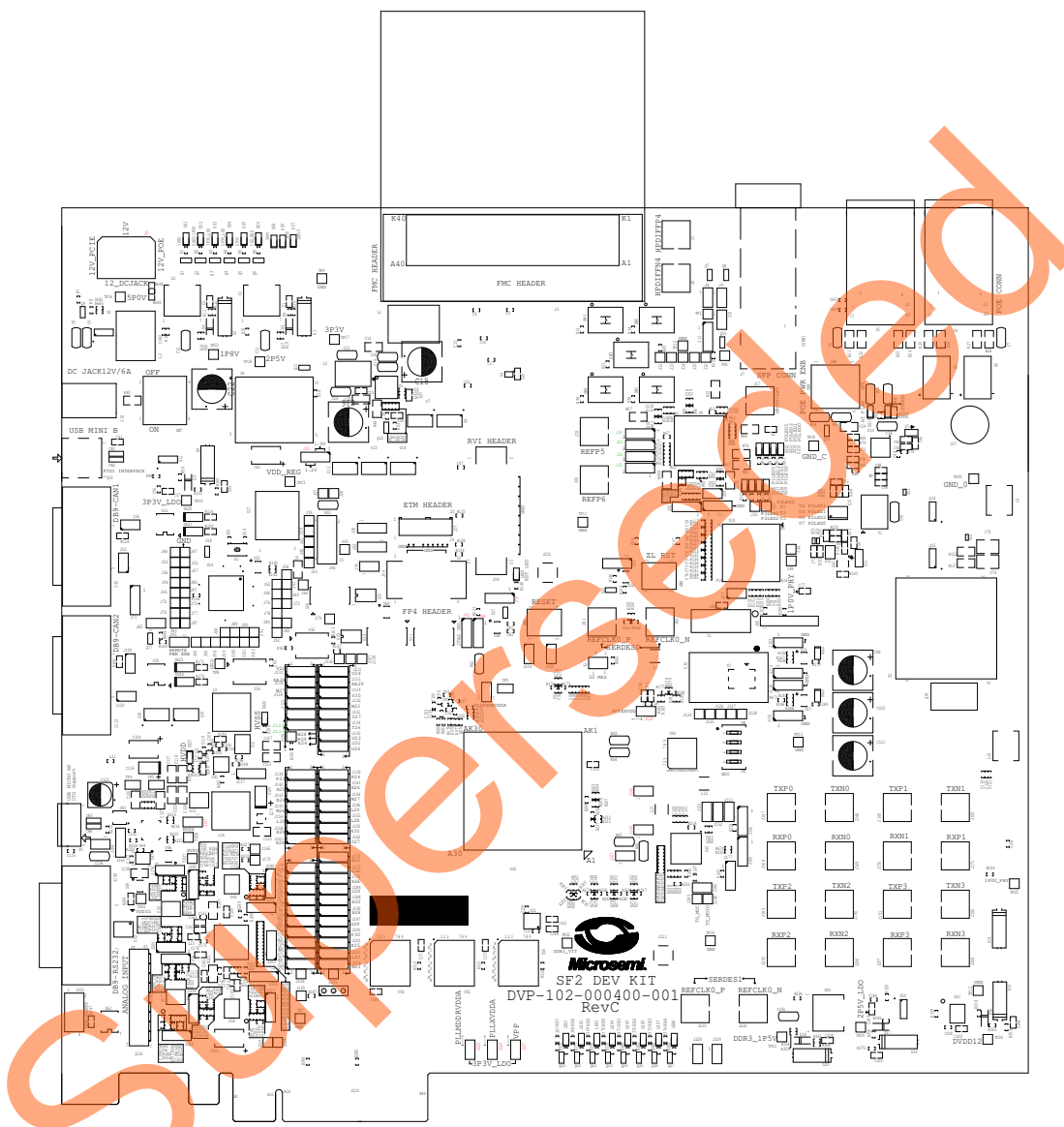


Figure 28 • Jumper Locations in Development Kit Board

- Jumpers highlighted in red are set by default.
- Jumpers highlighted in green must be set manually.
- The location of the jumpers in [Figure 28](#) are searchable.

Appendix 3: Running the Design in Static IP Mode

1. To run the design in Static IP mode, right-click the **Webserver_TCP_MSS_CM3_app** in the **Project Explorer** window of SoftConsole project and select **Properties** as shown in Figure 29.

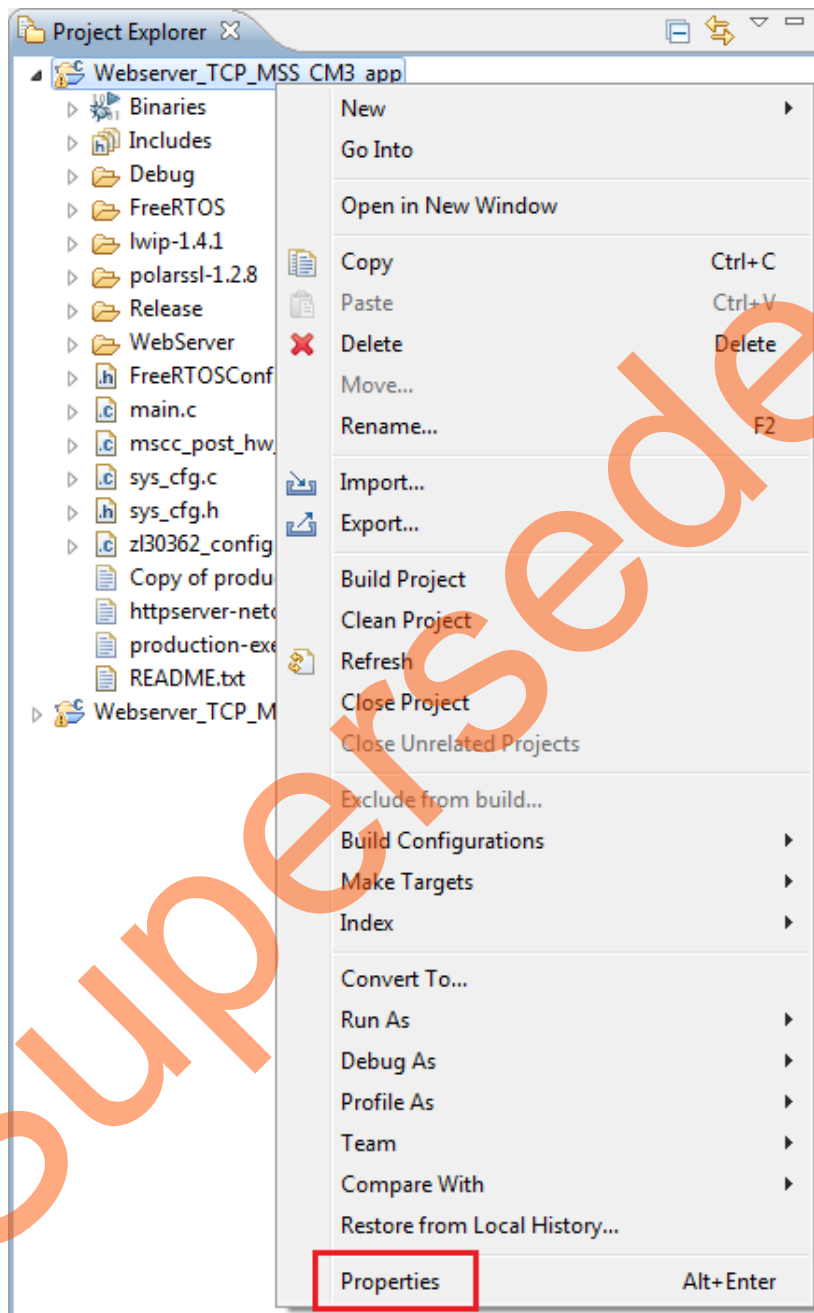


Figure 29 • Project Explorer Window of SoftConsole Project

Figure 30 shows removing the symbol **NET_USE_DHCP** in **Tool Settings** tab of the **Properties** for **Webserver_TCP_MSS_CM3** window.

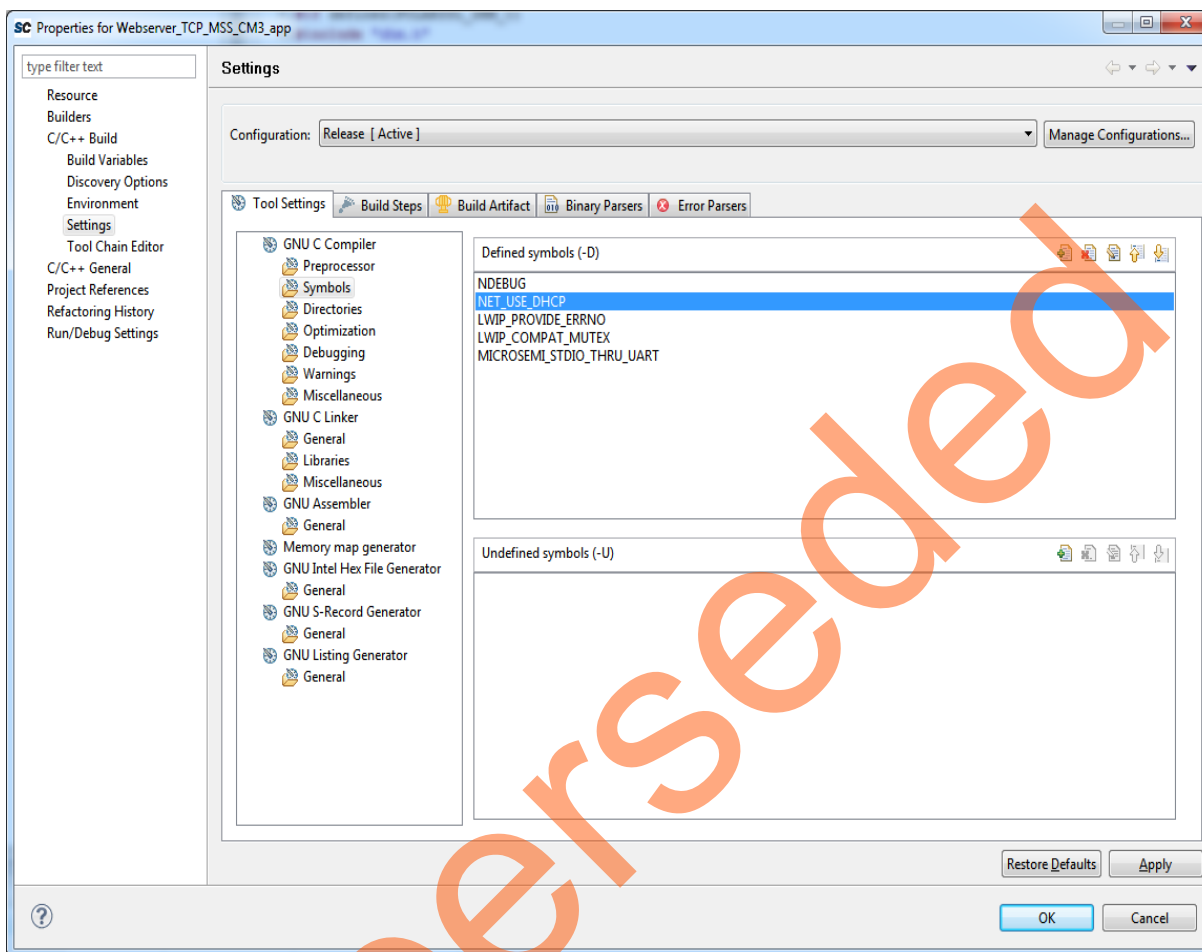


Figure 30 • Project Explorer Properties Window

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 31 shows Host PC TCP/IP settings.

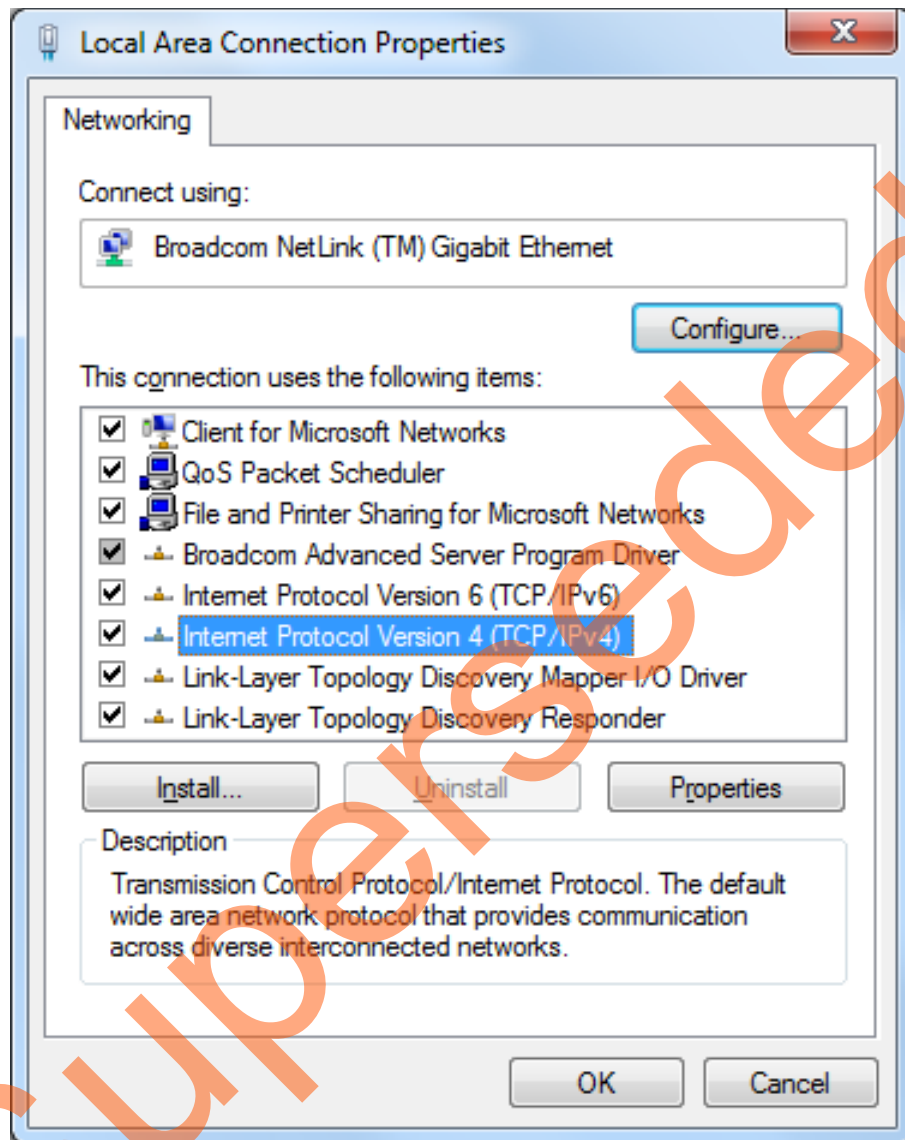


Figure 31 • Host PC TCP/IP Settings

Figure 32 shows Static IP address settings.

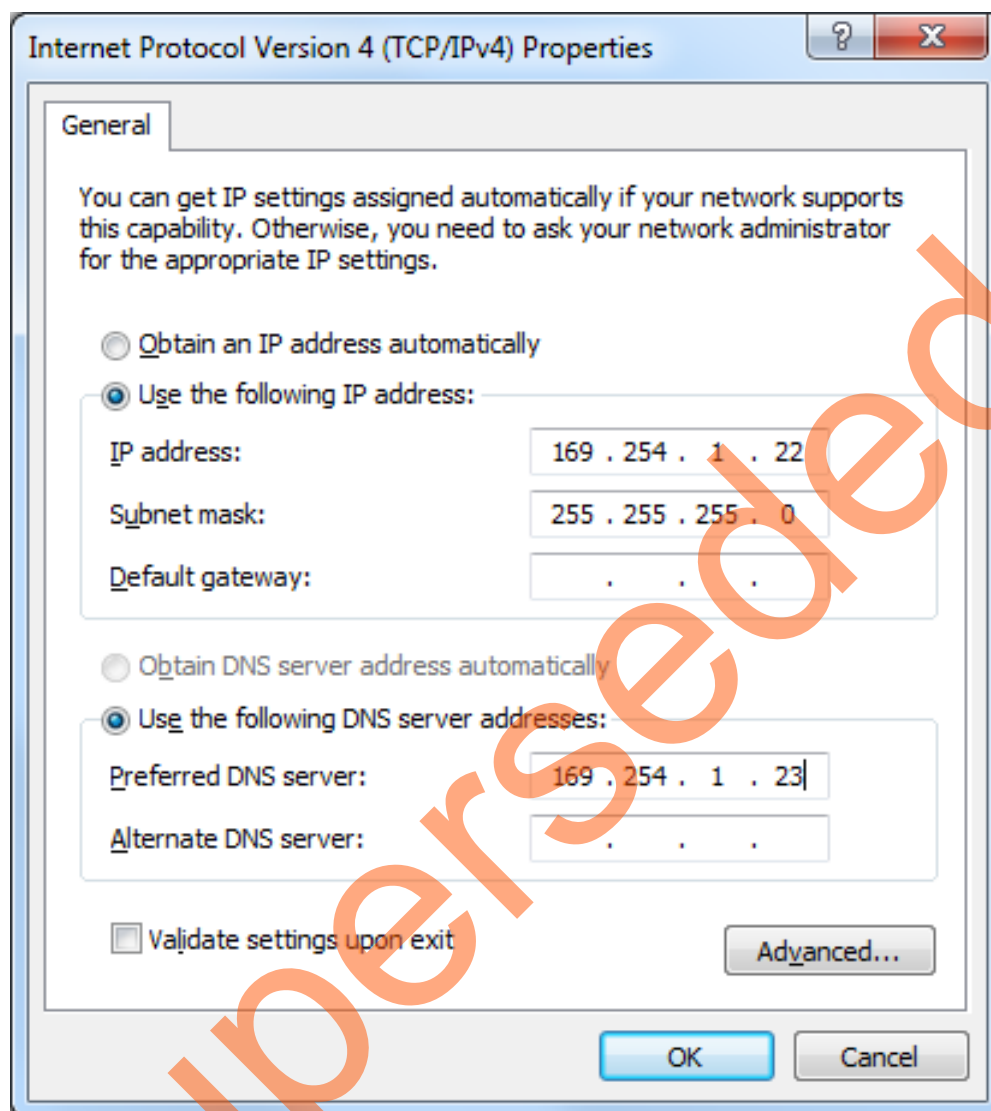


Figure 32 • Static IP Address Settings

Once these settings are made, build the design. Refer to "Generating *.bin File for Secure Webserver Application Image" section on page 16 on how to generate the *.bin file. Follow from Step 13. in the "Running the Demo Design" section on page 18 to execute the design in static IP mode, if the SmartFusion2 device is already programmed with `secure_webserver_tcp_top.stp` file.

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

Visit the Customer Support website (www.microsemi.com/support/search/default.aspx) for more information and support. Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on the website.

Website

You can browse a variety of technical and non-technical information on the SoC home page, at www.microsemi.com.

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. [Sales office listings](#) can be found at www.microsemi.com/company/contact/default.aspx.

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_itar@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.

Superseded

Superseded



Microsemi®

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 and 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; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif. and has approximately 3,400 employees globally. Learn more at www.microsemi.com.

© 2014 Microsemi Corporation. All rights reserved. Microsemi and the Microsemi logo are trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.