Running Secure Webserver on SmartFusion2 Devices Using PolarSSL, lwIP, and FreeRTOS - Libero SoC v11.6

DG0516 Demo Guide

November 2015

Microsemi

Revision History

Date	Revision	Change
18 November 2015	Revision 4	Fourth release
4 March 2015	Revision 3	Third release
16 September 2014	Revision 2	Second release
7 April 2014	Revision 1	First release

Confidentiality Status

This is a non-confidential document.

Table of Contents

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

SmartFusion2 devices are used by:

- FPGA designers
- Embedded designers
- System-level designers

References

The following references are used in this document:

- PolarSSL TLS/SSL protocol: https://tls.mbed.org/
- lwIP TCP/IP stack:
  - www.sics.se/~adam/lwip/
  - http://download.savannah.gnu.org/releases/lwip/
- FreeRTOS™ stack: www.freeRTOS.org

Microsemi Publications

- *UG0331: SmartFusion2 Microcontroller Subsystem User Guide*
- *UG0447: IGLOO2 and SmartFusion2 High Speed Serial Interfaces User Guide*
- *Libero SoC User Guide*
- *UG0557: SmartFusion2 SoC FPGA Advanced Development Kit User Guide*

Refer to the following web page for a complete and up-to-date listing of SmartFusion2 device documentation: *http://www.microsemi.com/products/fpga-soc/soc-fpga/sf2docs*

Running Secure Webserver Demo Design on 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 tri-speed ethernet medium access controller (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 Advanced 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 *UG0331: 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 (TLS/SSL Protocol)
- Transport Layer (lwIP TCP/IP Stack)
- RTOS and Firmware Layer

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

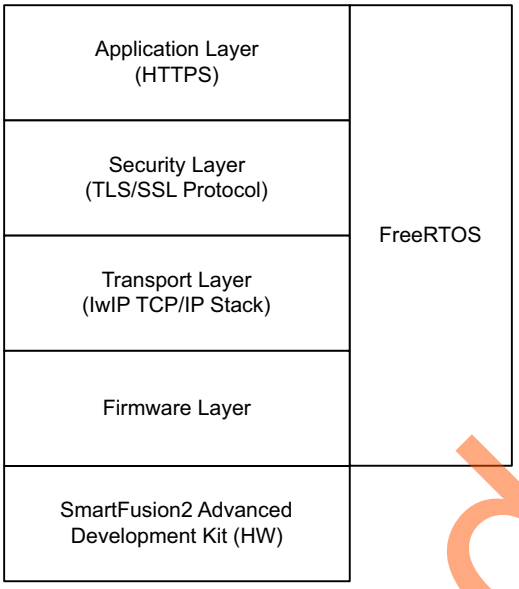

Figure 1 • **Block Diagram of Secure Webserver Demo Design on SmartFusion2**

Application Layer

The Secure Webserver application is implemented on the SmartFusion2 Advanced 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).

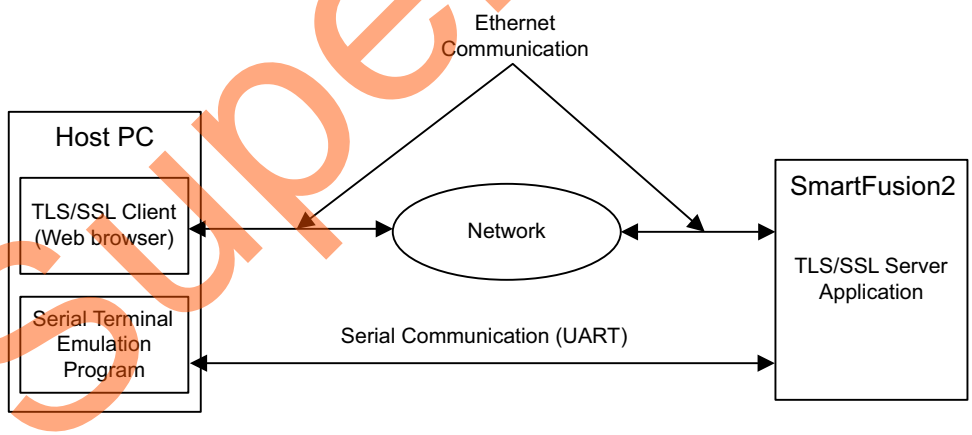

Figure 2 • **Client Server Communication Block Diagram**

When the URL with IP address (for example, https://10.60.3.120) is entered 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

Design Requirements

The following table lists the hardware and software design requirements.

Table 1 • **Design Requirements**

Design Requirements	Description
Hardware Requirements	
SmartFusion2 Advanced Development Kit • 12 V adapter • FlashPro5 • USB A to Mini-B cable	Rev A 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	v11.6
FlashPro Programming Software	v11.6
SoftConsole	v3.4 SP1
Host PC Drivers	USB to UART drivers
One of the following serial terminal emulation programs: • HyperTerminal • Tera-Term • 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=m2s_dg0516_liberov11p6_df

The demo design files include:

- The Libero SoC hardware project with SoftConsole firmware project
- 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.

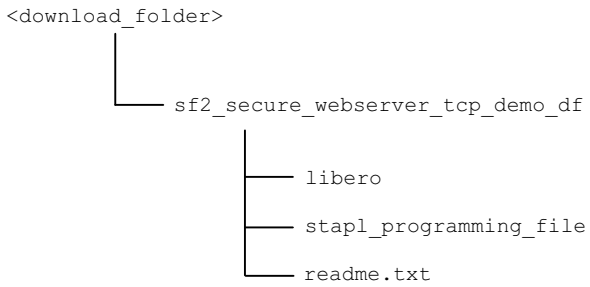

Figure 3 • **Demo Design Files Top-Level Structure**

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. For more information on the TSEMAC TBI interface, refer to the *UG0331: SmartFusion2 Microcontroller Subsystem User Guide*.

The demo design comprises:

- Libero SoC Hardware Project
- SoftConsole Firmware Project

Libero SoC Hardware Project

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

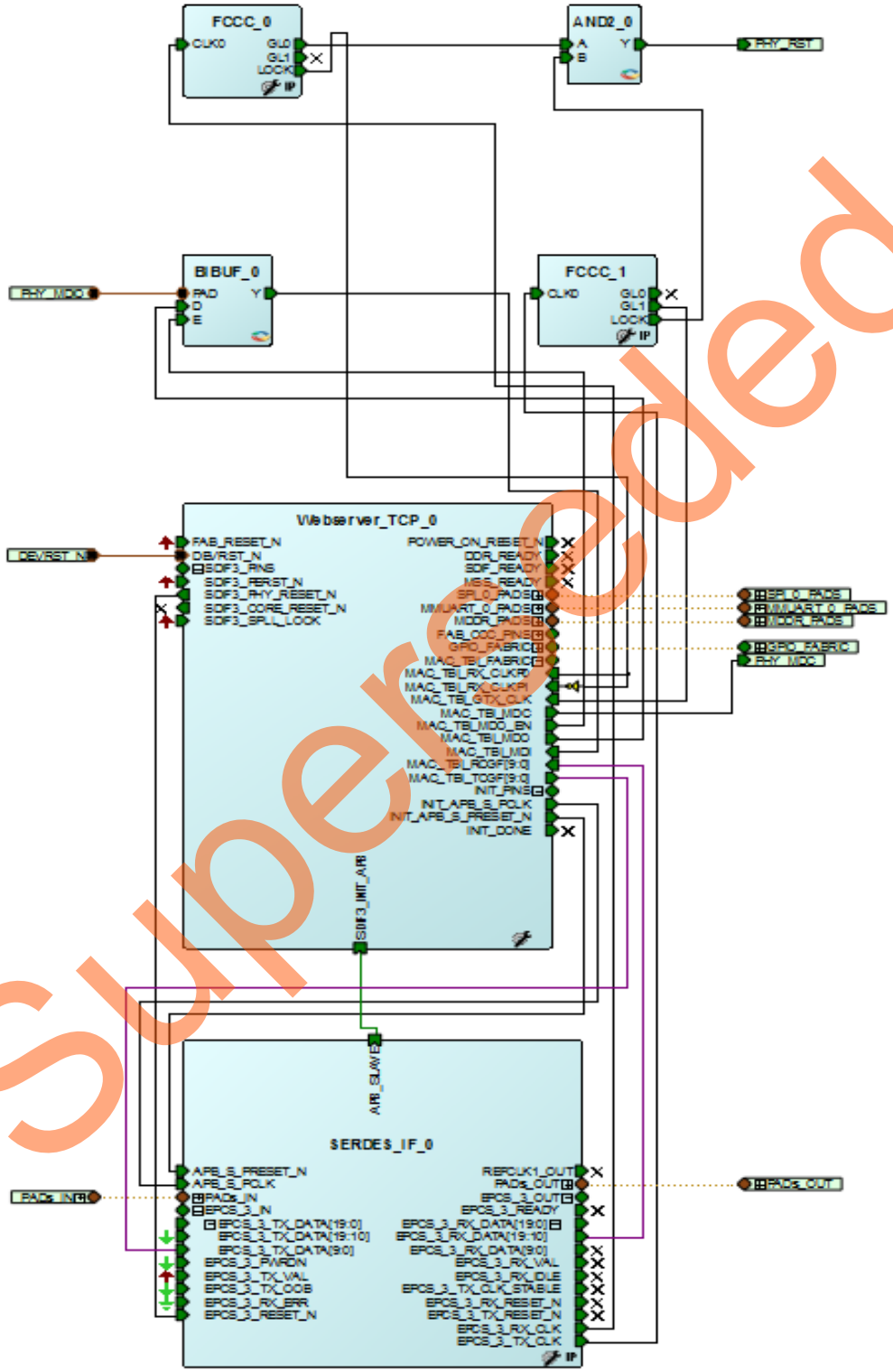

Figure 4 • **Libero Top-Level Design**

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

1. **TSEMAC TBI** interface.
2. **MMUART_0** for RS-232 communications on the SmartFusion2 Advanced Development Kit.
3. **General purpose input and output (GPIO)**: Interfaces with the light emitting diodes (LEDs).
4. **High speed serial interface (SERDESIF) SERDES_IF IP**: Configured for **SERDESIF_3 EPCS lane3** as shown in Figure 5.

   For more information on high-speed serial interfaces, refer to the *UG0447: IGLOO2 and SmartFusion2 High Speed Serial Interfaces User Guide*.

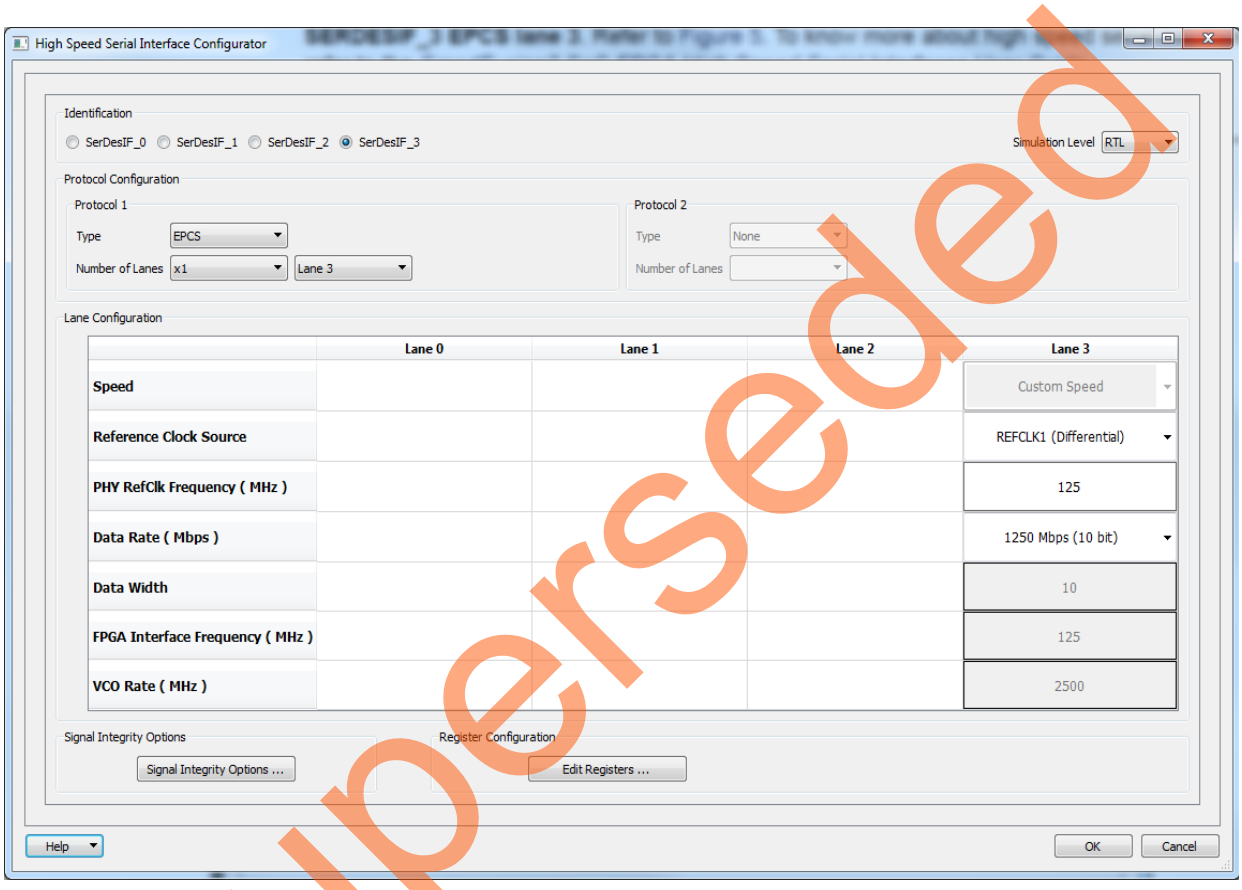

Figure 5 • **High-Speed Serial Interface Configurator Window**

5. 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.

Table 2 lists the port names for the package pins.

Table 2 • **LED to Package Pins Assignments**

Port Name	Package Pin
LED_1	D26
LED_2	F26
LED_3	F27
LED_4	C26
LED_5	C28
LED_6	B27
LED_7	C27
LED_8	E26

Table 3 lists 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	F3
PHY_MDIO	Input	K7
PHY_RST	Output	F2

SoftConsole Firmware Project

Invoke the SoftConsole project using standalone SoftConsole IDE.

The following stacks are used for this demo design:

- **PolarSSL** library version 1.2.8
- **lwIP TCP/IP** stack version 1.4.1
- **FreeRTOS**

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.

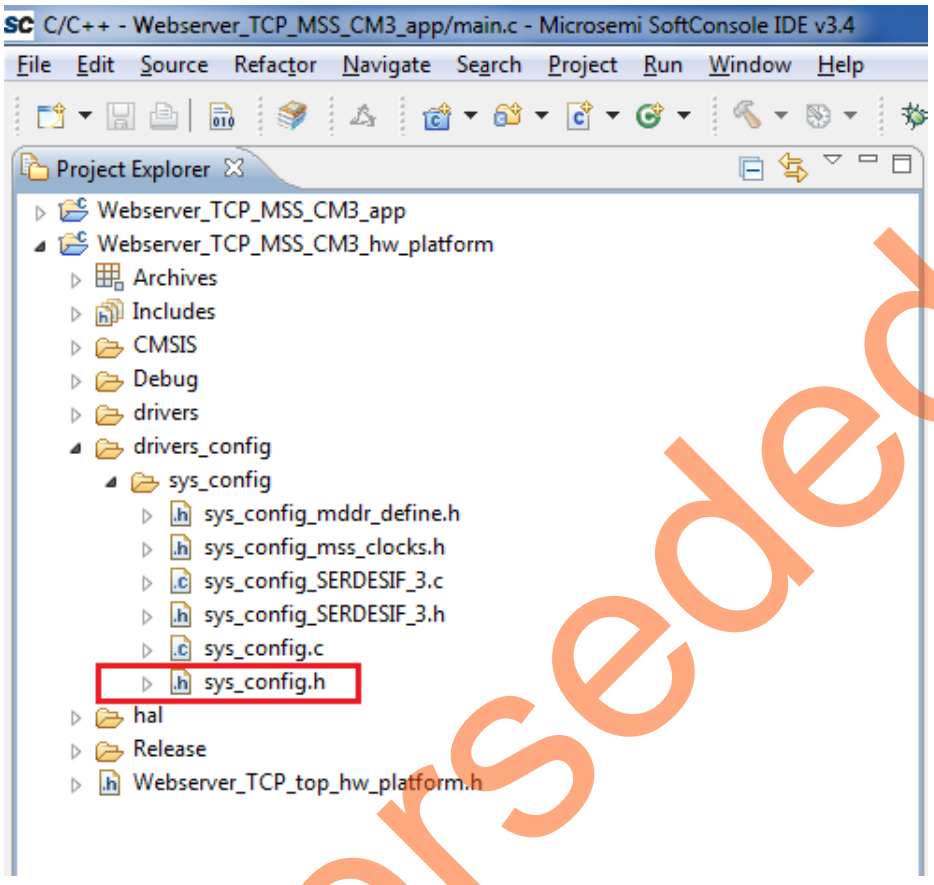

Figure 6 • **Example SoftConsole Project Explorer Window**

The SoftConsole workspace consists of two projects.

1. Webserver_TCP_MSS_CM3_app

   This project contains the Secure Webserver application implementation using PolarSSL, LWIP, and FreeRTOS.

   The advanced encryption standard (AES) and non-deterministic random bit generator (NRBG) system services are used to implement the Secure Webserver application. The AES and NRBG can be implemented using SmartFusion2 hardware engine or software PolarSSL library. In this demo design, AES and NRBG are implemented using SmartFusion2 hardware engine through system services.

Table 4 • **Macros to Enable or Disable System Controller Services**

System Service	Macro	Macro Location
AES	`#define HW_AES 1`	*<download_folder>\sf2_secure_webserver_tcp_demo_df\libero\Webserver_TCP\SoftConsole\Webserver_TCP_MSS_CM3\Webserver_TCP_MSS_CM3_app\polarssl-1.2.8\include\polarssl\aes.h*
NRBG	`#define HW_NRBG 1`	*<download_folder>\sf2_secure_webserver_tcp_demo_df\libero\Webserver_TCP\SoftConsole\Webserver_TCP_MSS_CM3\Webserver_TCP_MSS_CM3_app\polarssl-1.2.8\include\polarssl\ssl.h*

Note: The system services AES and NRBG are supported for data security enabled SmartFusion2 device like M2S0150TS. If the SmartFusion2 device is not data security enabled, disable the macros mentioned in Table 4 to use the software PolarSSL AES and NRBG algorithms.

2. 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 by regenerating the root design every time and exporting the SoftConsole firmware project 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.

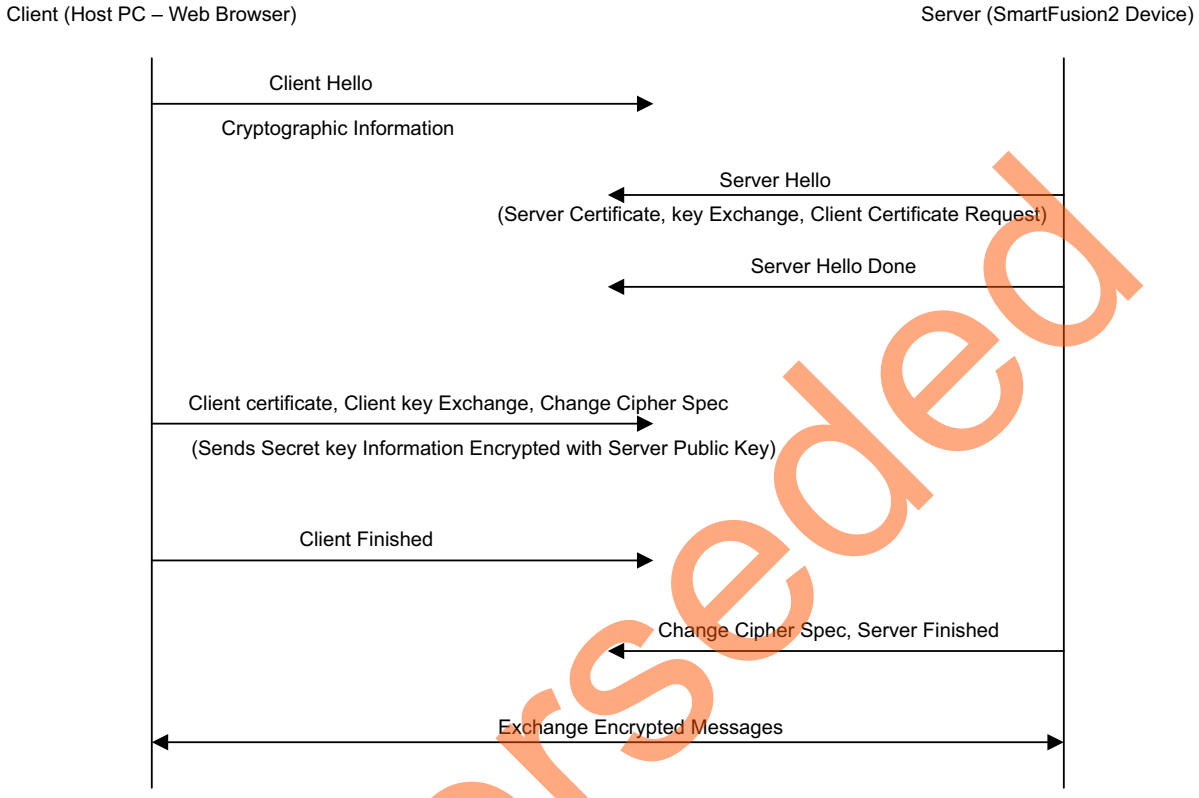

Figure 7 • **TLS/SSL Handshake Procedure**

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.

Setting Up the Demo Design

The following steps describe how to setup the demo for SmartFusion2 Advanced Development Kit board:

1. Connect the host PC to the **J33 Connector** using the USB A to mini-B cable. The USB to UART bridge drivers are automatically detected.
2. From the detected four COM ports, select the one which location on its Properties window must be as **on USB FP5 Serial Converter C**. Make a note of the COM port number for serial port configuration and ensure that the COM port Location is specified as **on USB FP5 Serial Converter C**, as shown in Figure 8.

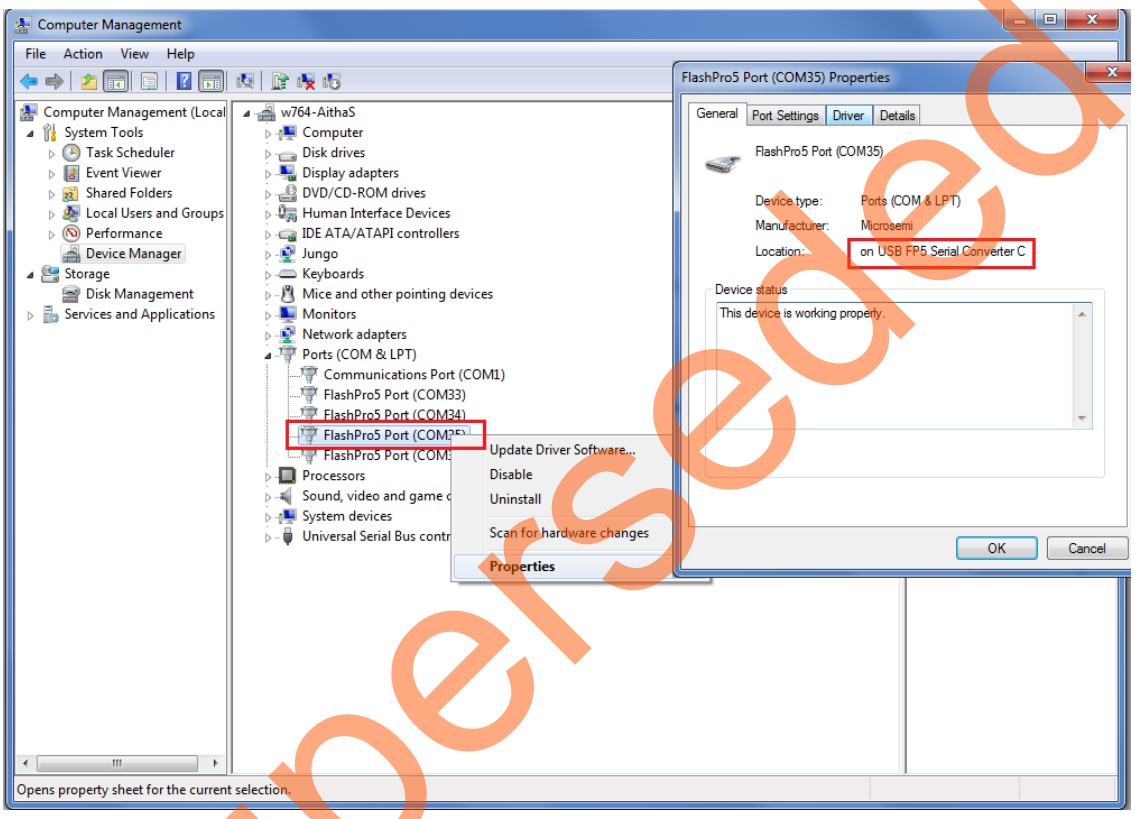

Figure 8 • **Device Manager Window**

3. If USB drivers are not detected automatically, install the USB driver.
4. 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

5. Connect the jumpers on the SmartFusion2 Advanced Development Kit board as shown in Table 5. For information on jumper locations, refer to "Appendix 2: Jumper Locations" on page 30.

   Caution: Switch OFF the power supply switch, **SW7**, before making the jumper connections,

Table 5 • **SmartFusion2 Advanced Kit Jumper Settings**

Jumper	Pin (from)	Pin (to)	Comments
J116, J353, J354, J54	1	2	These are the default jumper settings of the Advanced Dev Kit board. Ensure these jumpers are set accordingly.
J123	2	3	
J124, J121, J32	1	2	JTAG programming via FTDI
J118, J119	1	2	Programming SPI Flash

6. In the SmartFusion2 Advanced Development Kit, connect the power supply to the J42 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 J21 connector of the SmartFusion2 Advanced Development Kit board using an RJ45 cable.
   - For dynamic IP, connect any one of the open network ports to the J21 connector of the SmartFusion2 Advanced Development Kit board using an RJ45 cable.

Board Setup Snapshot

Snapshots of the SmartFusion2 Advanced Development Kit board with all the setup made is given in "Appendix 1: Board Setup for Running the Secure Webserver" on page 29.

Running the Demo Design

The following steps describe how to run the demo design:

1. Download the demo design from:
   http://soc.microsemi.com/download/rsc/?f=m2s_dg0516_liberov11p6_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: 115200
   - 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, enter the project name as shown in Figure 9.

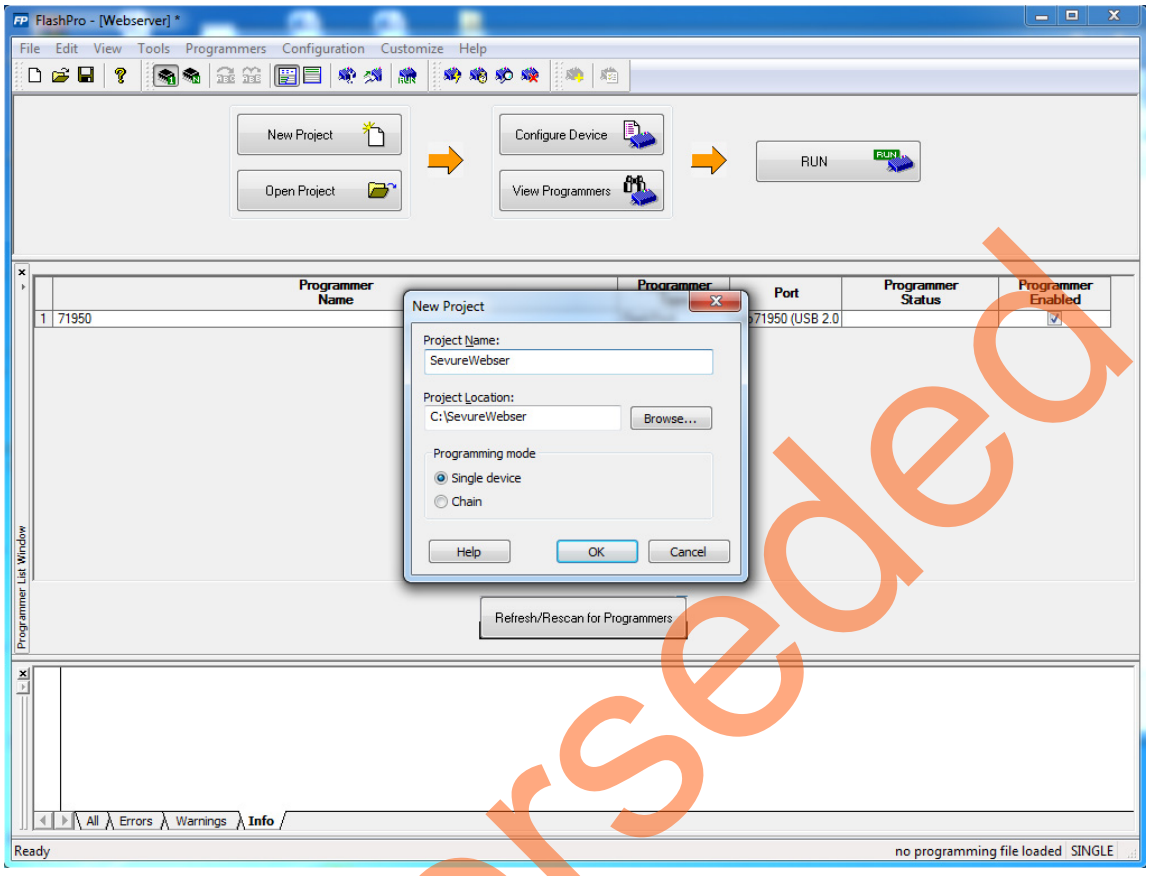

Figure 9 • **FlashPro New Project**

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 `Webserver_tcp_top_Secure_Demo.stp` file is located and select the file. The default location is: *<download_folder>\sf2_secure_Webserver_tcp_demo_df\stapl_programming_file\Webserver_tcp_top_Secure_Demo.stp*

    The required programming file is selected and is ready to be programmed in the device as shown in Figure 10.

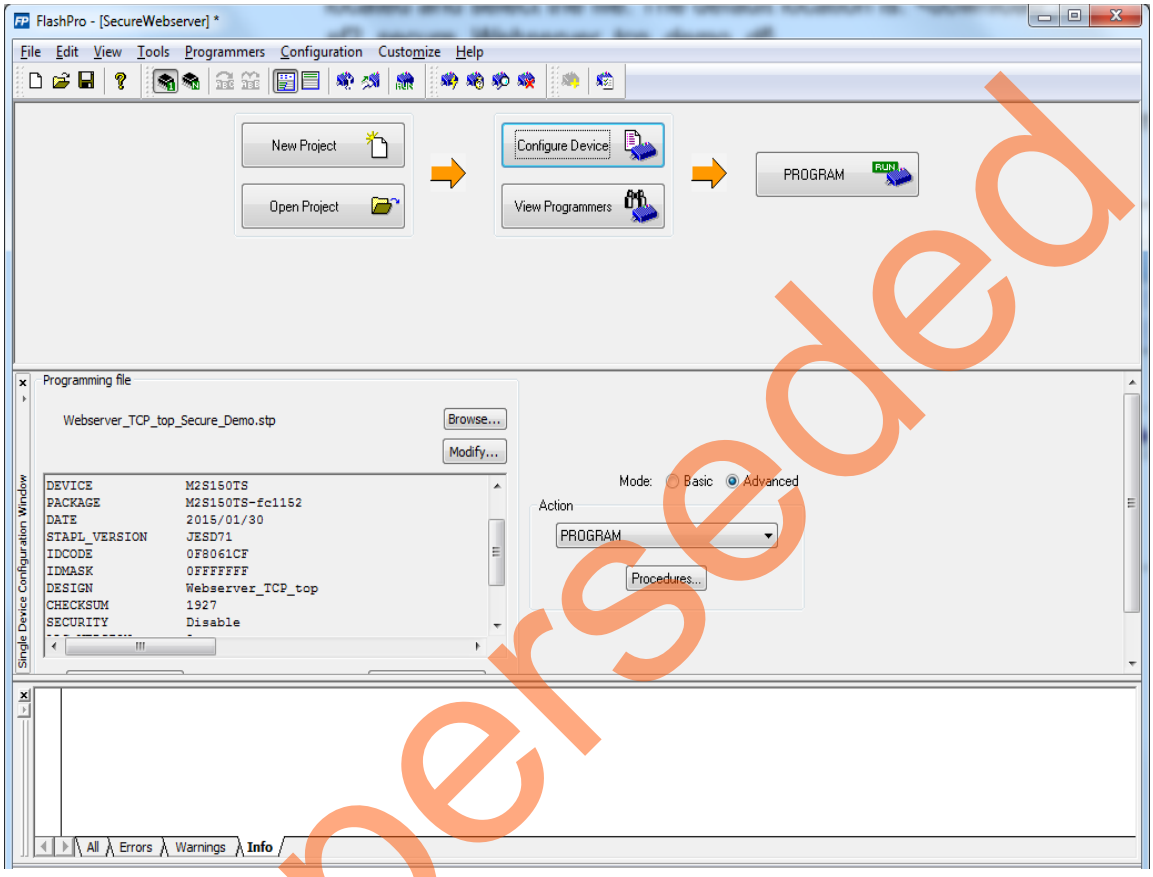

Figure 10 • **FlashPro Project Configured**

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

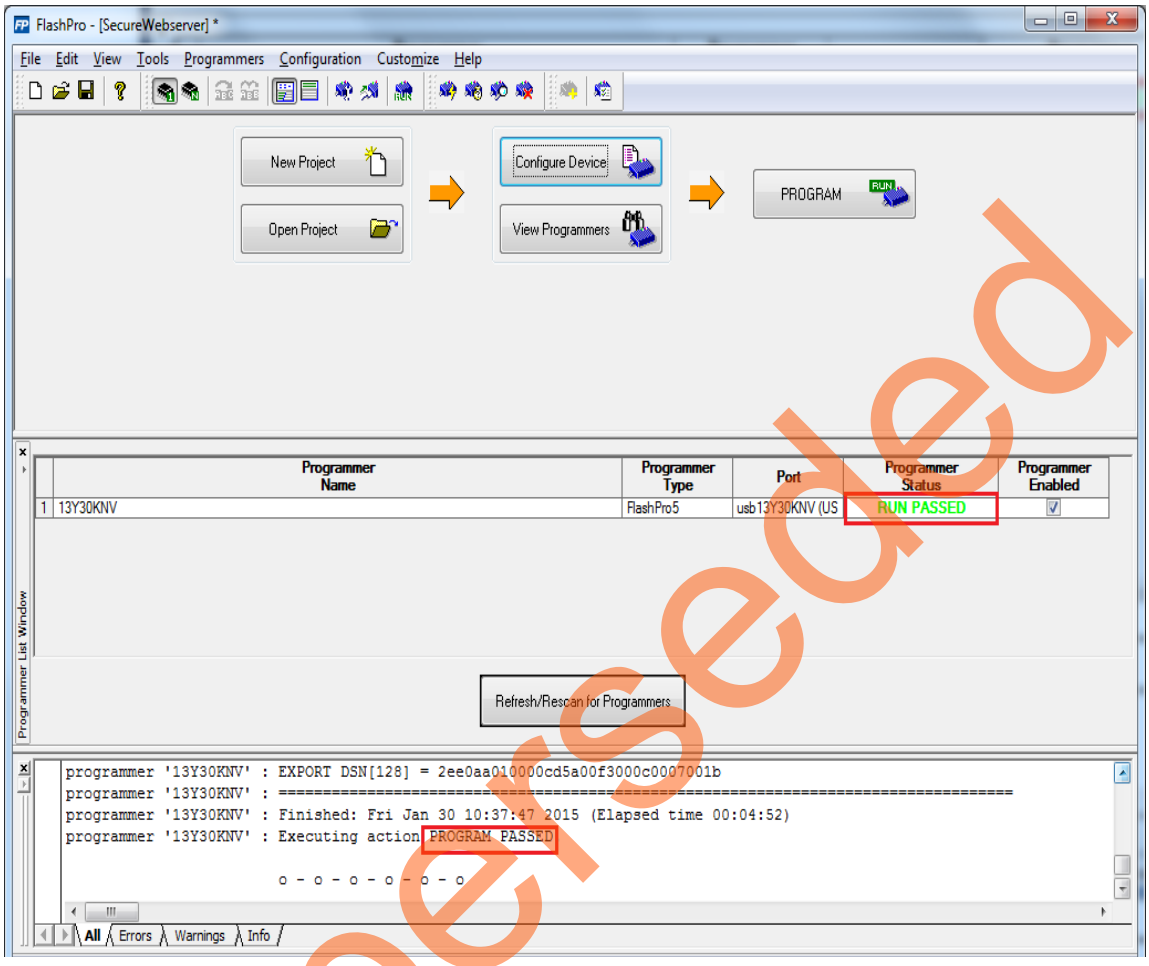

Figure 11 • **FlashPro Program Passed**

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 31.

13. Power cycle the SmartFusion2 Advanced Development Kit board.

    A welcome message with the dynamic IP address is displayed in the serial terminal emulation program as shown in Figure 12.

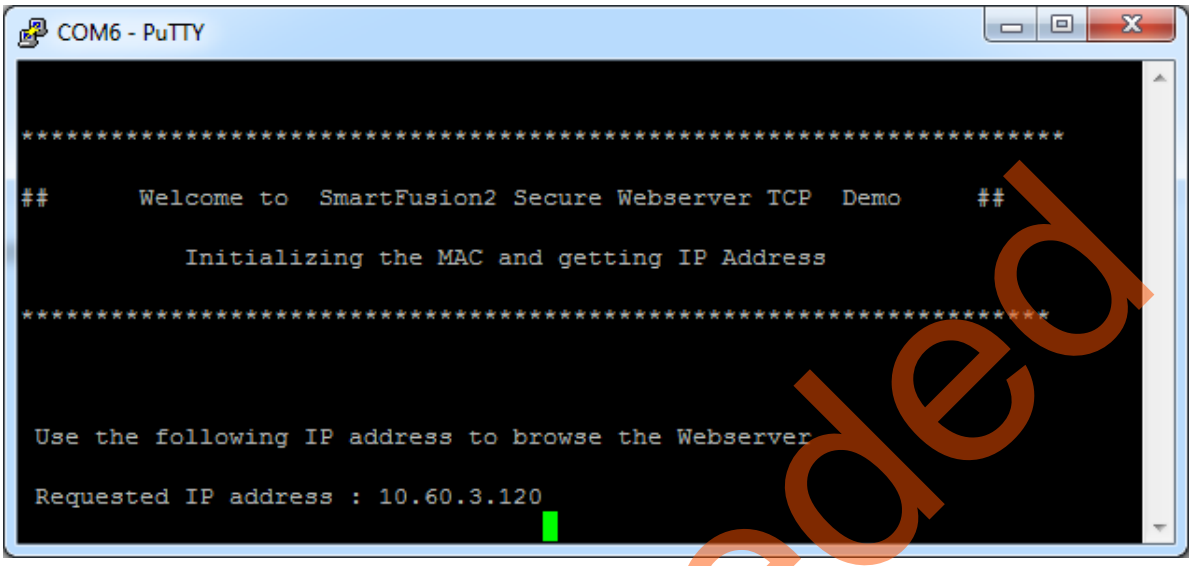

Figure 12 • **User Options**

14. 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, enter 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

The following steps describe how to run 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 13.

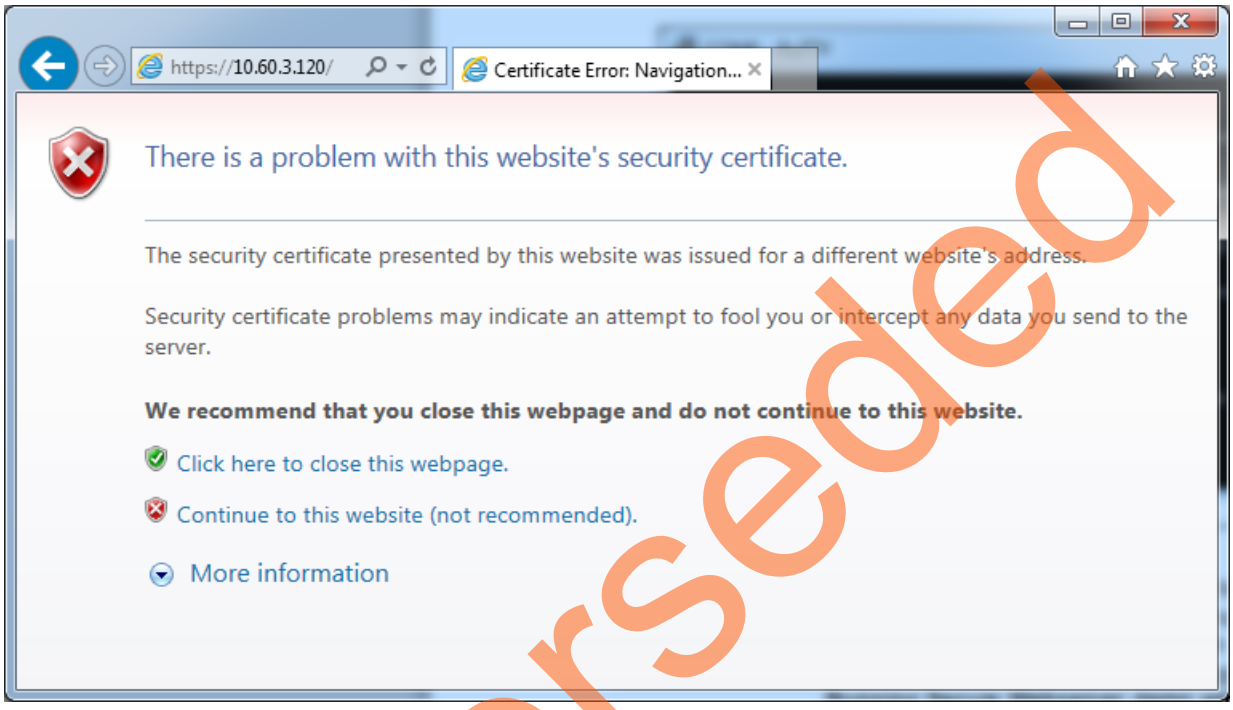

Figure 13 • **Microsoft Internet Explorer showing Certificate Error Warning Message**

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 14.

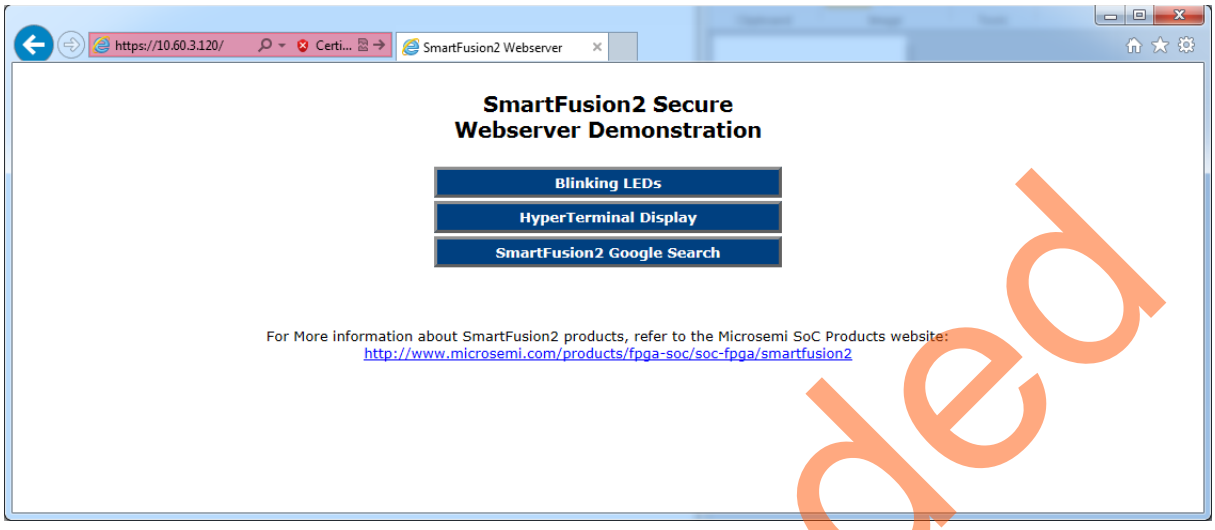

Figure 14 • **Main Menu of Secure Webserver in Internet Explorer**

Running the Secure Webserver Demo with Mozilla Firefox

The following steps describe how to run the Secure Webserver Demo with Mozilla Firefox:

1. Open the Mozilla Firefox browser and enter the URL (for example, https://10.60.3.120) in the address bar. The browser shows a warning message as shown in Figure 15.

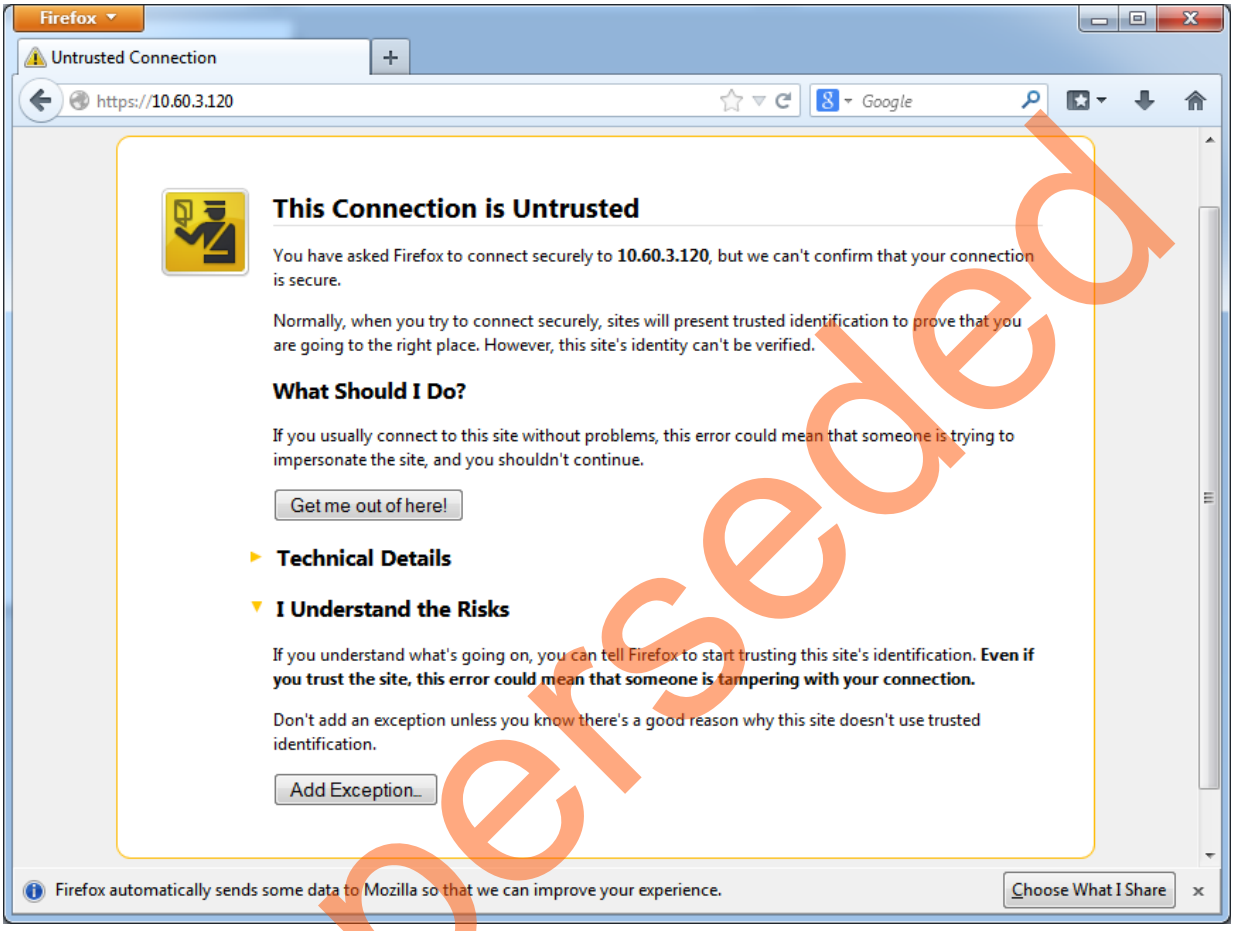

Figure 15 • **Mozilla Firefox showing Warning Message**

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

3. Click **Confirm Security Exception** in **Add Security Exception** window as shown in Figure 16, to start secure communication with the Webserver.

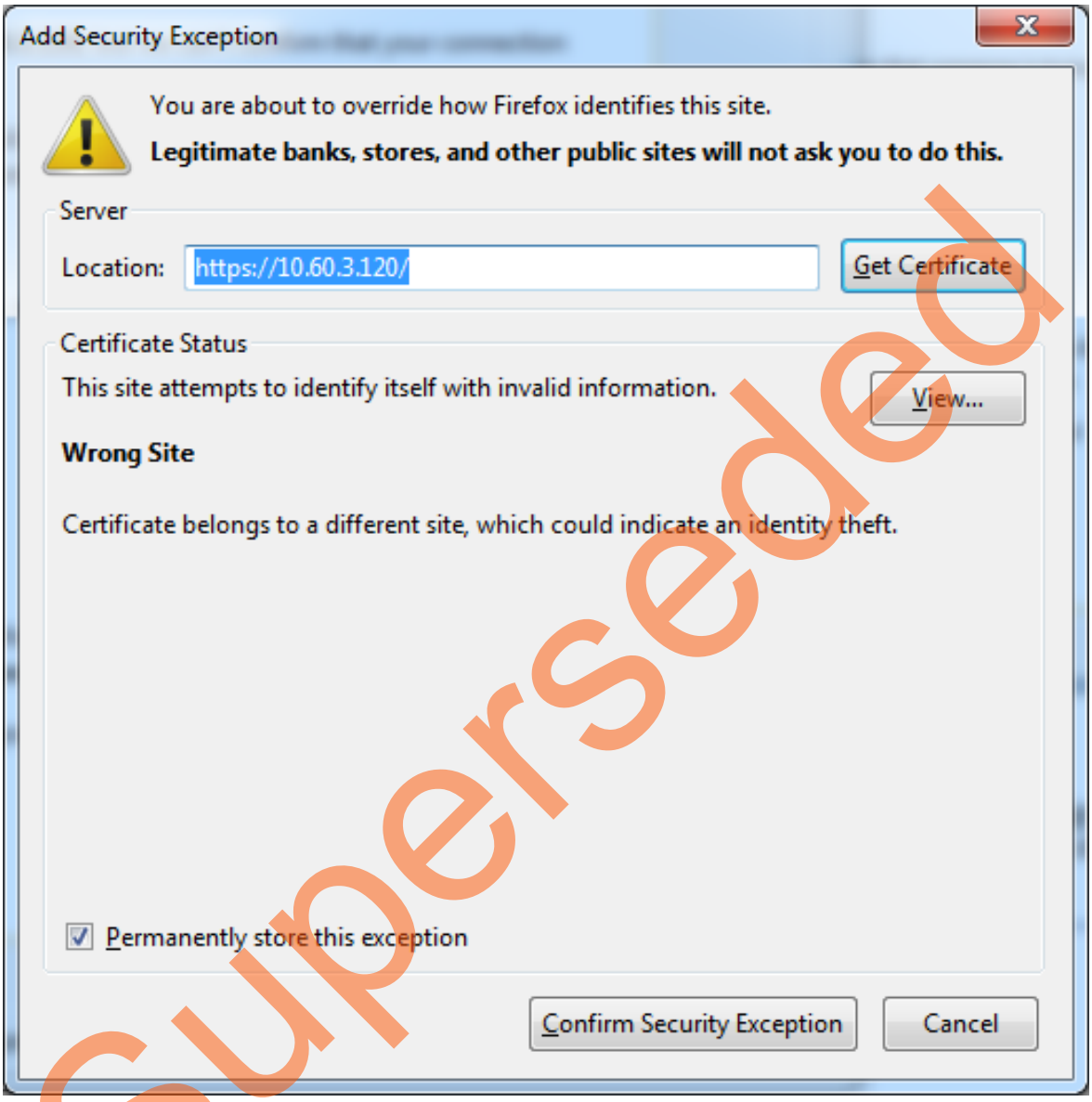

Figure 16 • **Add Security Exception Window**

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 17.

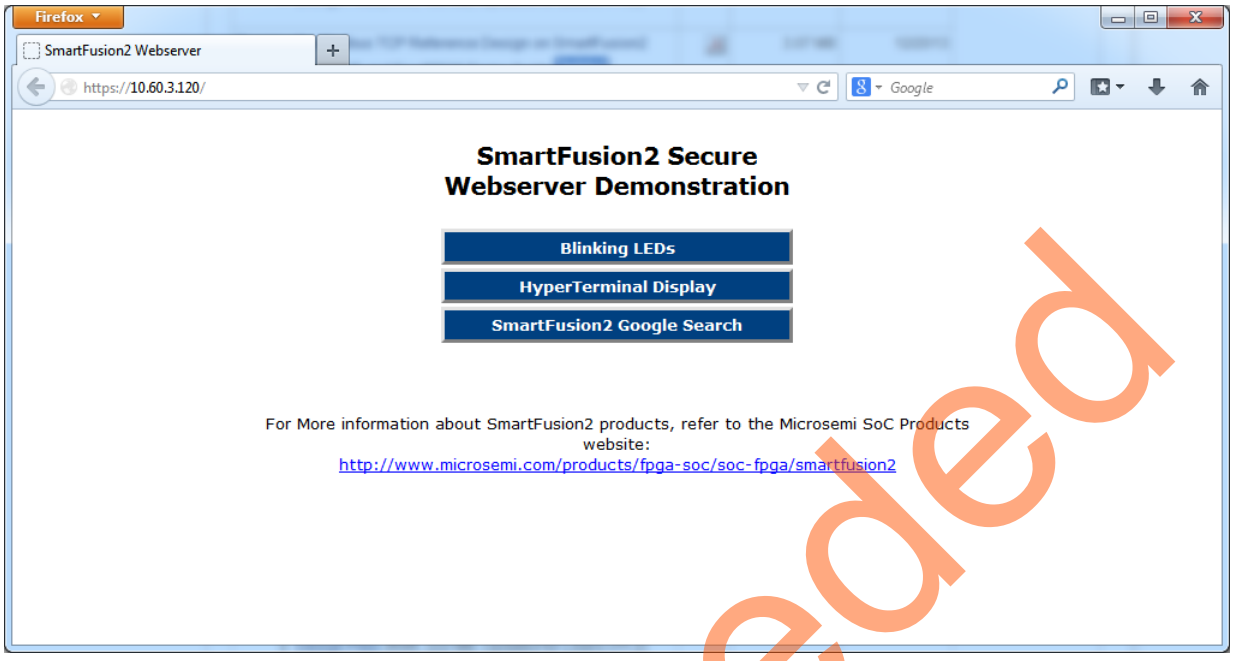

Figure 17 • **Main Menu of the Secure Webserver in Mozilla Firefox**

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 LEDs** 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 18.

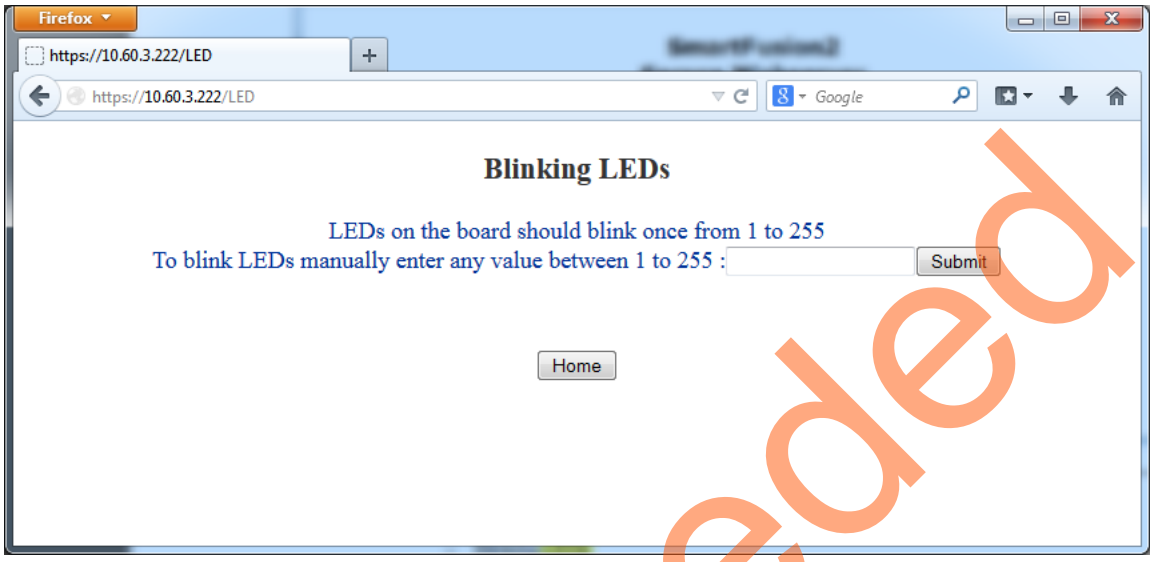

Figure 18 • **Blinking LEDs Page**

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

Note: SmartFusion2 Advanced Development Kit has Active Low LEDs.

HyperTerminal Display

1. Click **HyperTerminal Display** on the main menu. Figure 19 shows a webpage that gives an option to enter a string value.

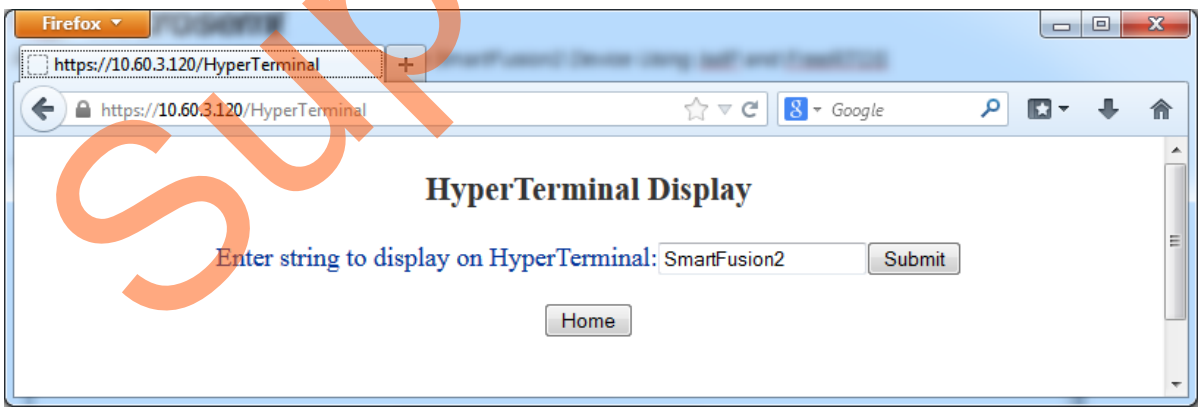

Figure 19 • **HyperTerminal Display Page**

The entered string is displayed on PuTTY as shown in Figure 20.

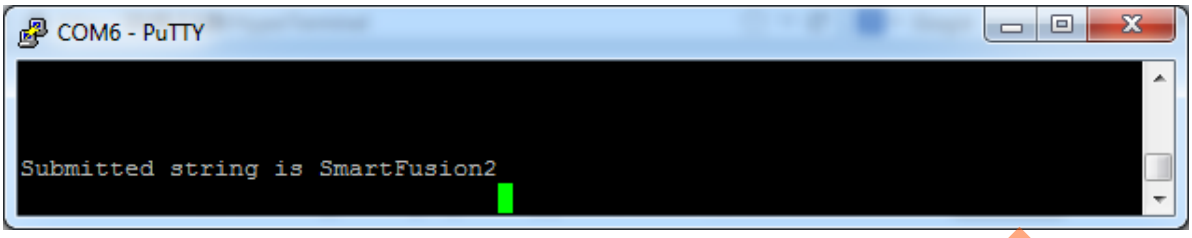

Figure 20 • **String Display on PuTTY**

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 21 shows a web page with Google search.

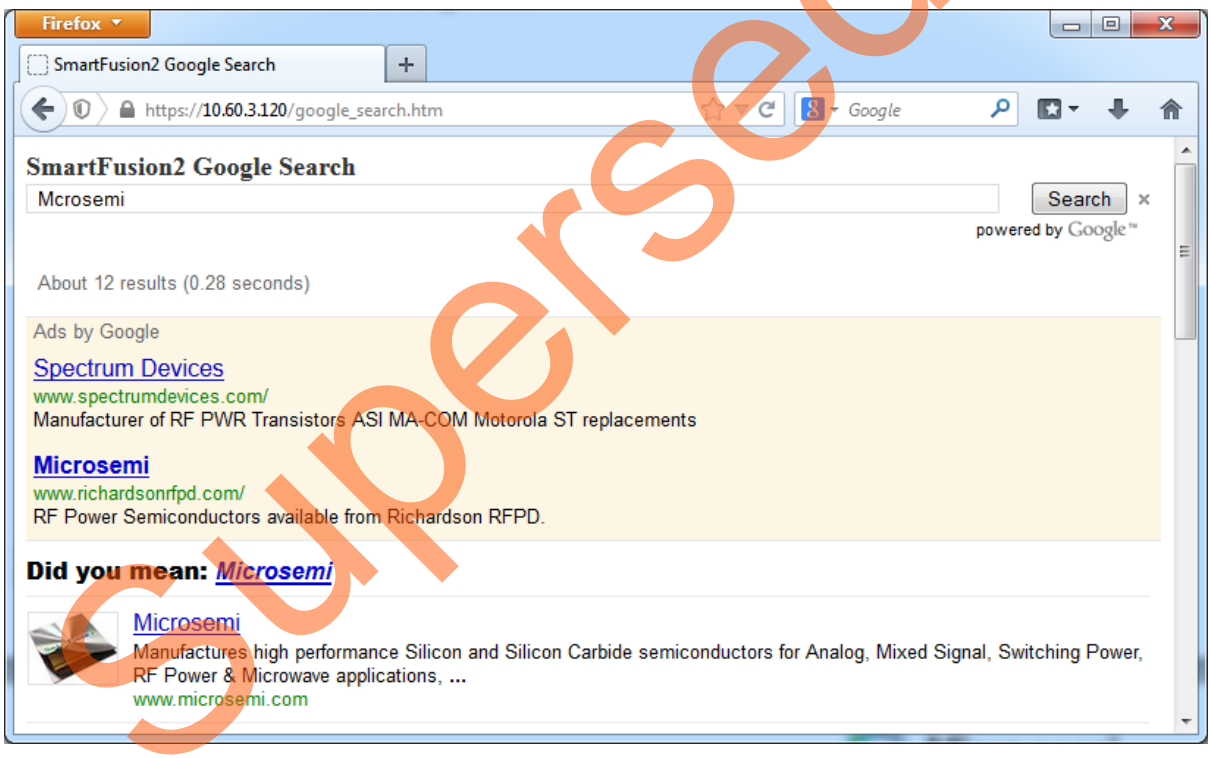

Figure 21 • **SmartFusion2 Google Search Page**

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

Appendix 1: Board Setup for Running the Secure Webserver

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

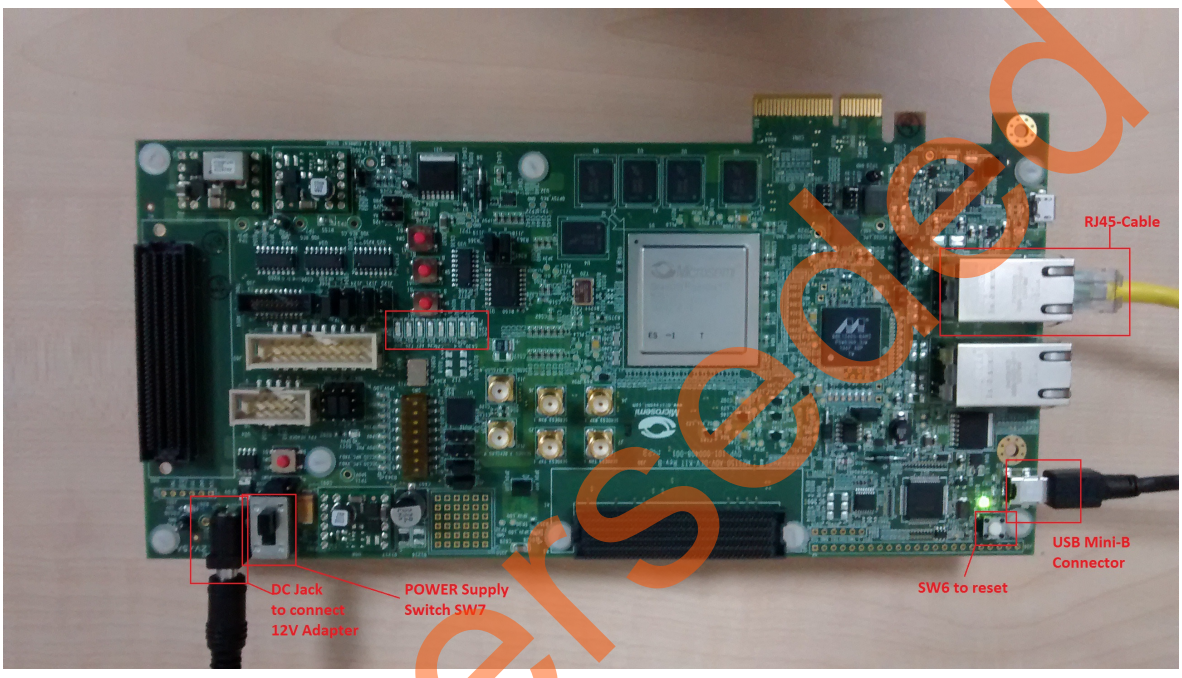

***Figure 1* • SmartFusion2 Advanced Development Kit Setup**

Appendix 2: Jumper Locations

Figure 1 shows the jumper locations in the SmartFusion2 Advanced Development Kit board.

***Figure 1* • Jumper Locations in Advanced Development Kit Board**

Note:

- Jumpers highlighted in red are set by default.
- Jumpers highlighted in green must be set manually.
- The location of the jumpers in Figure 1 are searchable.

Appendix 3: Running the Design in Static IP Mode

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

1. Right-click the **Webserver_TCP_MSS_CM3_app** in the **Project Explorer** window of SoftConsole project and select **Properties** as shown in Figure 1.

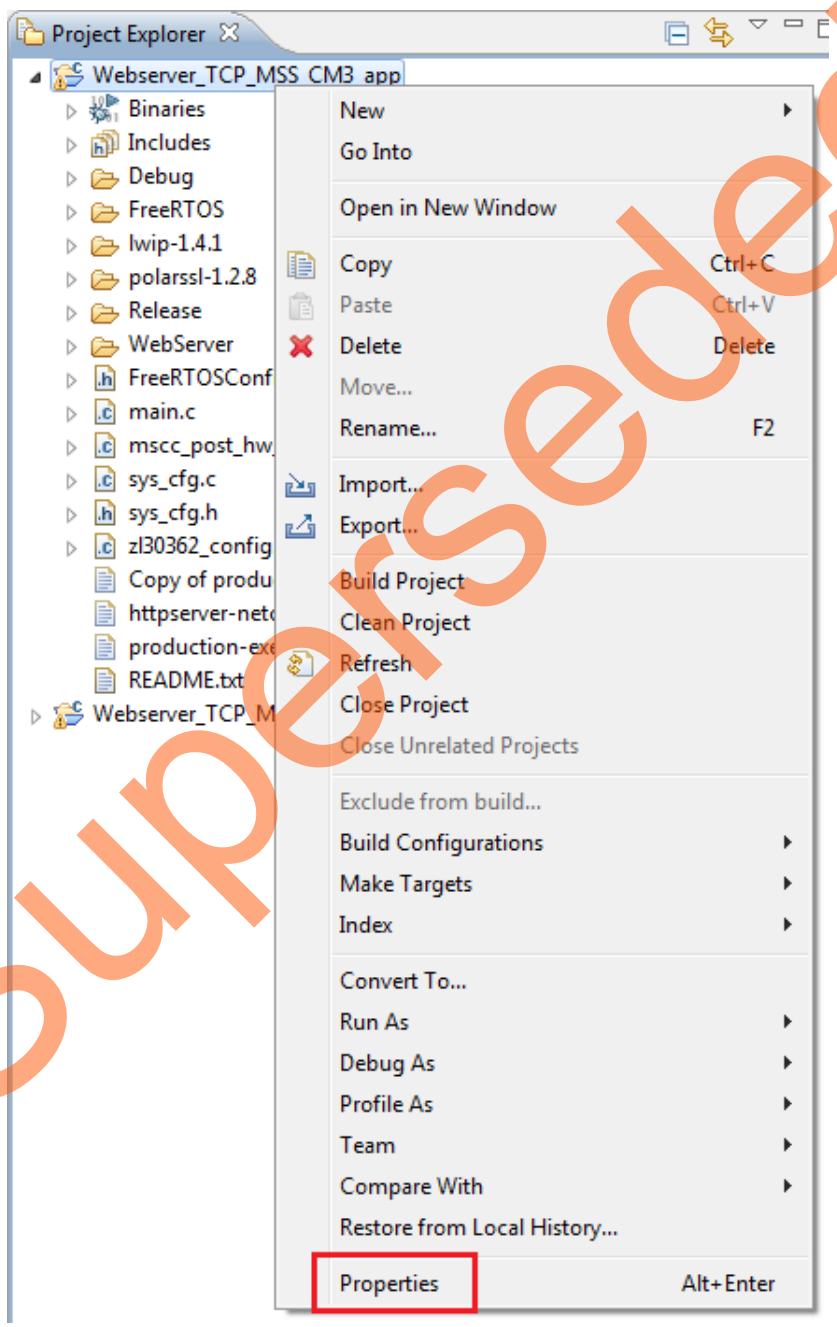

***Figure 1* • Project Explorer Window of SoftConsole Project**

Figure 2 shows removing the symbol **NET_USE_DHCP** in the **Tool Settings** tab of the **Properties for Webserver_TCP_MSS_CM3_app** window.

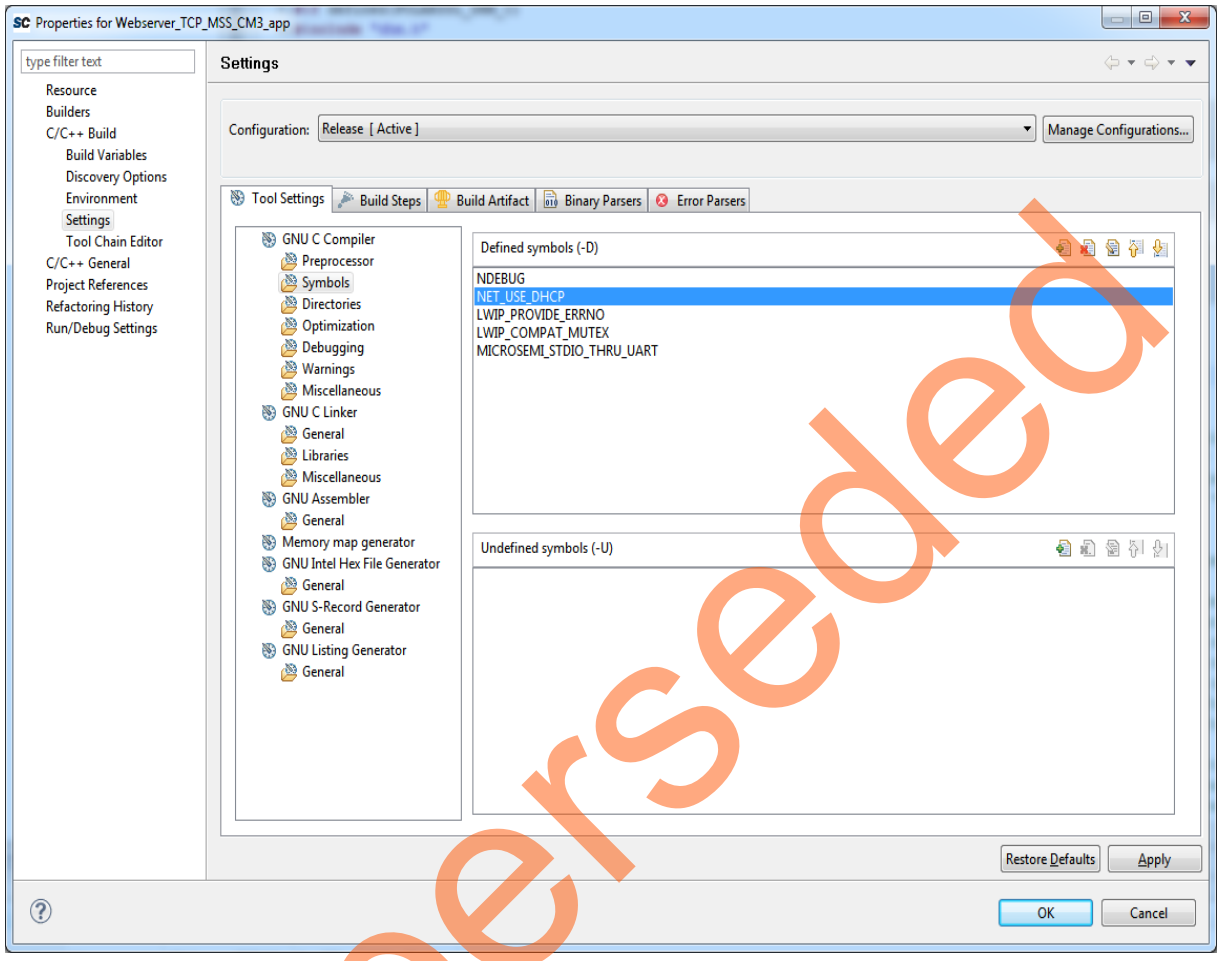

***Figure 2* • Project Explorer Properties Window**

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 3 shows host PC TCP/IP settings.

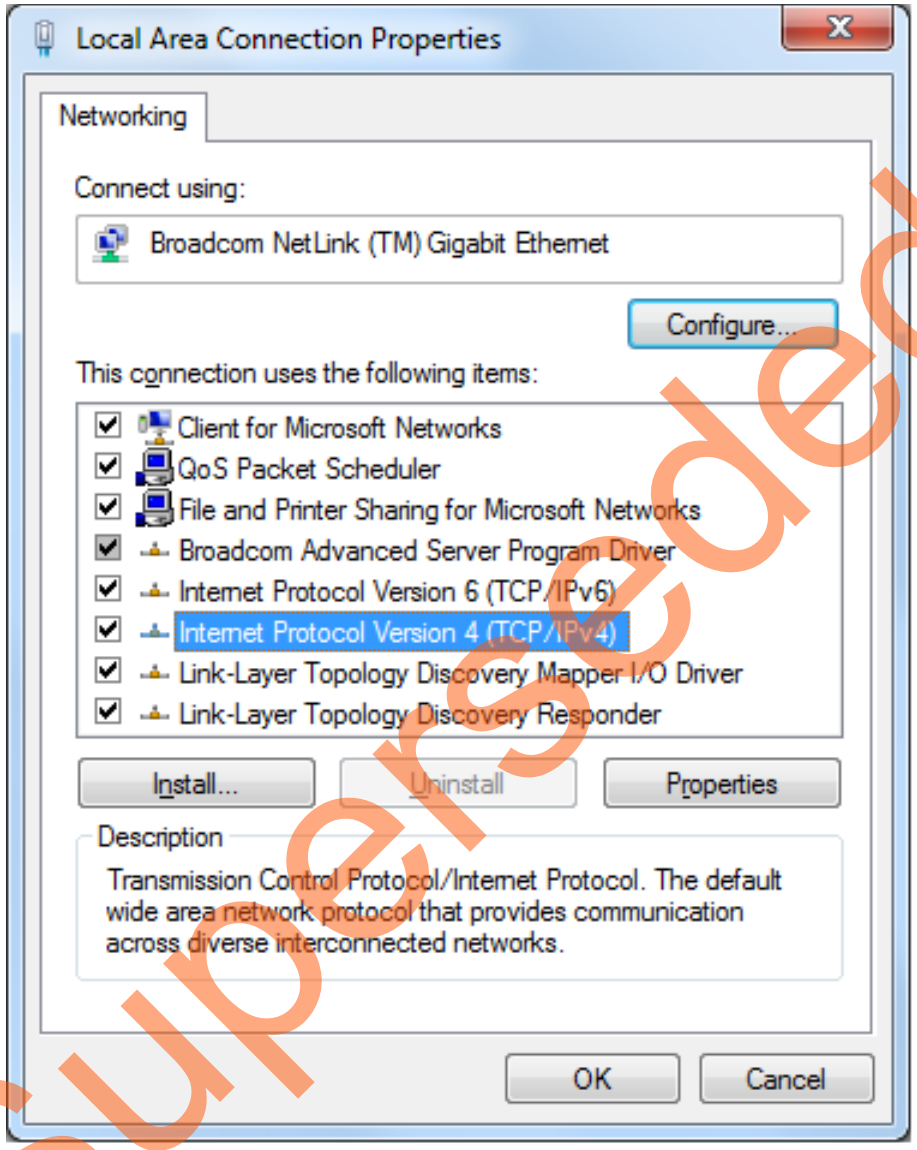

Figure 3 • **Host PC TCP/IP Settings**

Figure 4 shows Static IP address settings.

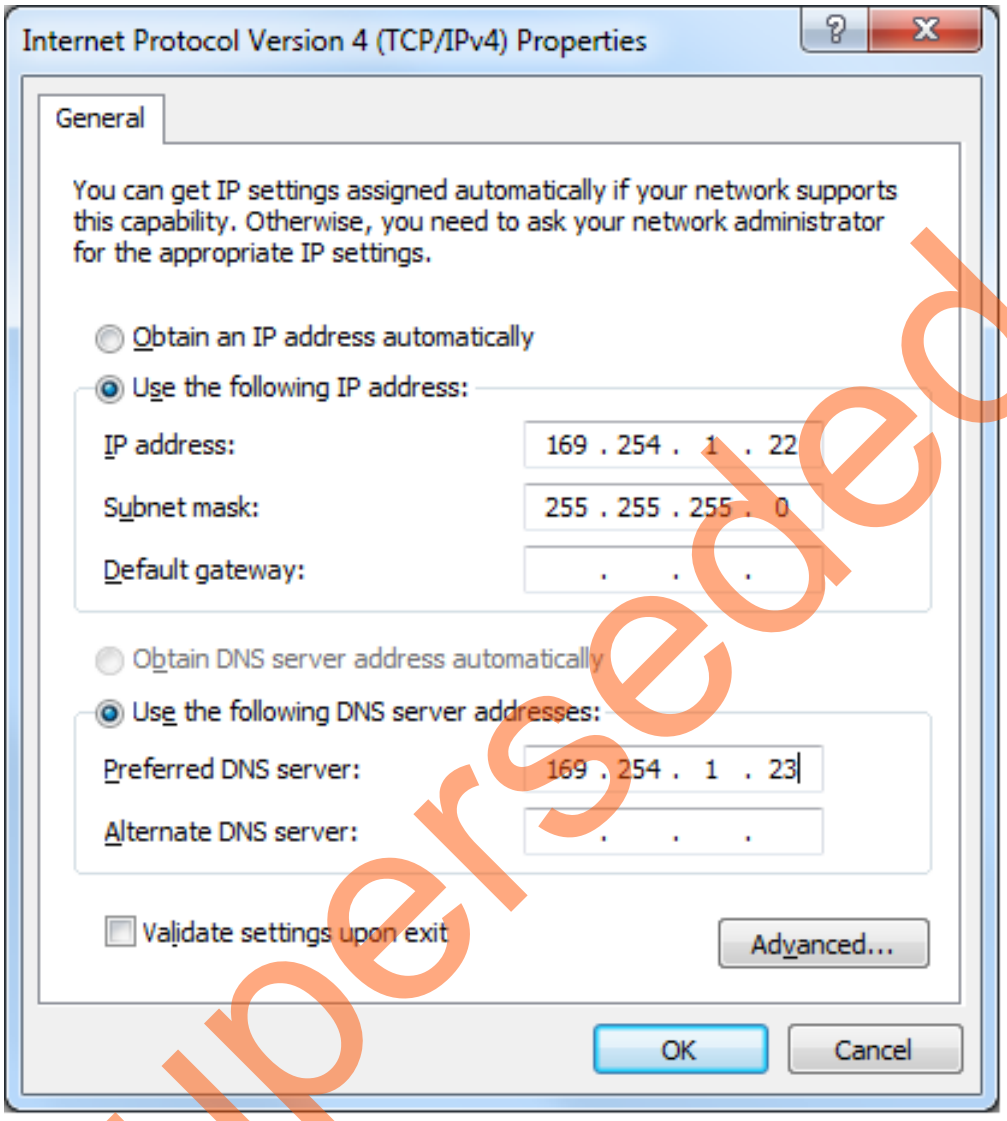

Figure 4 • **Static IP Address Settings**

Once these settings are made, build the design. Refer to "Running the Demo Design" section on page 1-17 to execute the design in static IP mode, if the SmartFusion2 device is already programmed with `Webserver_TCP_top_Secure_Demo.stp` file.

Note: To run the application in debug mode, FlashPro4 JTAG programmer is required.

A – List of Changes

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

Date	Changes	Page
Revision 5 (November 2015)	Updated "SoftConsole Firmware Project" section (SAR 73518).	1-12
Revision 4 (October 2015)	Updated the document for Libero v11.6 software release (SAR 72058).	NA
Revision 3 (March 2015)	Updated the document for Libero v11.5 software release (SAR 63973).	NA
Revision 2 (September 2014)	Updated the document for Libero v11.4 software release (SAR 60685).	NA
Revision 1 (April 2014)	Initial release.	NA
The revision number is located in the part number after the hyphen. The part number is displayed at the bottom of the last page of the document. The digits following the slash indicate the month and year of publication.		

B – 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**.

50200516-5/11.15