

DG0516
Demo Guide
Running Secure Webserver on SmartFusion2 Devices
using PolarSSL, lwIP, and FreeRTOS



a  **MICROCHIP** company



a  MICROCHIP company

Microsemi 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

Email: sales.support@microsemi.com

www.microsemi.com

©2021 Microsemi, a wholly owned subsidiary of Microchip Technology Inc. All rights reserved. Microsemi and the Microsemi logo are registered trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.

About Microsemi

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), 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. Learn more at www.microsemi.com.

Contents

1	Revision History	1
1.1	Revision 9.0	1
1.2	Revision 8.0	1
1.3	Revision 7.0	1
1.4	Revision 6.0	1
1.5	Revision 5.0	1
1.6	Revision 4.0	1
1.7	Revision 3.0	1
1.8	Revision 2.0	1
1.9	Revision 1.0	1
2	Running Secure Webserver Demo Design on SmartFusion2 Devices Using PolarSSL, lwIP, and FreeRTOS	2
2.1	Secure Webserver Demo Design Overview	2
2.1.1	Application Layer	3
2.1.2	Security Layer (TLS/SSL Protocol)	4
2.1.3	Transport Layer (lwIP TCP/IP Stack)	4
2.1.4	RTOS and Firmware Layer	4
2.2	Design Requirements	5
2.3	Prerequisites	5
2.4	Demo Design	6
2.4.1	Demo Design Features	6
2.4.2	Demo Design Description	6
2.5	Setting Up the Demo Design	11
2.5.1	Board Setup Snapshot	12
2.6	Running the Demo Design	13
2.6.1	Running the Secure Webserver Demo with Microsoft Internet Explorer	14
2.6.2	Running the Secure Webserver Demo with Mozilla Firefox	15
3	Appendix 1: Programming the Device Using FlashPro Express	19
4	Appendix 2: Board Setup for Running the Secure Webserver	22
5	Appendix 3: Jumper Locations	23
6	Appendix 4: Running the Design in Static IP Mode	24

Figures

Figure 1	Secured Webserver Layers	3
Figure 2	Client Server Communication Block Diagram	3
Figure 3	Demo Design Files Top-Level Structure	6
Figure 4	Libero SoC Top-Level Hardware Design	7
Figure 5	High-Speed Serial Interface Configurator Window	8
Figure 6	SoftConsole Project Explorer Window	9
Figure 7	Demo Design Driver Versions	10
Figure 8	TLS/SSL Handshake Procedure	11
Figure 9	Device Manager Window	12
Figure 10	User Options	13
Figure 11	Microsoft Internet Explorer showing Certificate Error Warning Message	14
Figure 12	Main Menu of Secure Webserver in Internet Explorer	14
Figure 13	Mozilla Firefox showing Warning Message	15
Figure 14	Add Security Exception Window	16
Figure 15	Main Menu of the Secure Webserver in Mozilla Firefox	16
Figure 16	Blinking LEDs Page	17
Figure 17	HyperTerminal Display Page	17
Figure 18	String Display on PuTTY	17
Figure 19	SmartFusion2 Google Search Page	18
Figure 20	FlashPro Express Job Project	19
Figure 21	New Job Project from FlashPro Express Job	20
Figure 22	Programming the Device	20
Figure 23	FlashPro Express—RUN PASSED	21
Figure 24	SmartFusion2 Advanced Development Kit Setup	22
Figure 25	Jumper Locations in Advanced Development Kit Board	23
Figure 26	Project Explorer Window of SoftConsole Project	24
Figure 27	Project Explorer Properties Window	25
Figure 28	Host PC TCP/IP Settings	25
Figure 29	Static IP Address Settings	26

Tables

Table 1	Design Requirements	5
Table 2	LED to Package Pins Assignments	8
Table 3	PHY Interface Signals to Package Pins Assignments	9
Table 4	Macros to Enable or Disable System Controller Services	10
Table 5	SmartFusion2 Advanced Kit Jumper Settings	12

1 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

1.1 Revision 9.0

The following is a summary of the changes made in this revision.

- Updated the document for Libero SoC v2021.1.
- Removed the references to Libero version numbers.

1.2 Revision 8.0

Updated the document for Libero v11.8 SP1 software release.

1.3 Revision 7.0

The following are the changes done in revision 7.0 of this document.

- Libero SoC, FlashPro, and SoftConsole design requirements are updated. For more information, see [Design Requirements](#), page 5.
- Throughout the guide, the names of SoftConsole projects used in the demo design and all the associated figures are updated.

1.4 Revision 6.0

Updated the document for Libero v11.7 software release (SAR 76931) in revision 6.0 of this document.

1.5 Revision 5.0

Updated [SoftConsole Firmware Project](#), page 9 (SAR 73518).

1.6 Revision 4.0

Updated the document for Libero v11.6 software release (SAR 72058).

1.7 Revision 3.0

Updated the document for Libero v11.5 software release (SAR 63973).

1.8 Revision 2.0

Updated the document for Libero v11.4 software release (SAR 60685).

1.9 Revision 1.0

Revision 1.0 was the first publication of this document.

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

This demo explains the secure webserver capabilities using Transport Layer Security (TLS), Secure Sockets Layer (SSL) protocol, and tri-speed ethernet medium access controller (TSEMAC) of the SmartFusion[®]2 devices. This demo describes:

- Using SmartFusion2 Ethernet Media Access Control (MAC) connected to a Serial Gigabit Media Independent Interface (SGMII) PHY.
- Integrating SmartFusion2 MAC driver with the PolarSSL library (free TLS/SSL protocol library), Lightweight IP (lwIP) TCP/IP stack, and the free Real Time Operating System (RTOS).
- Using Microsemi cryptographic system services to implement the TLS/SSL protocol.
- Implementing a secure webserver application on the SmartFusion2 Advanced Development Kit board.
- Running the demo.

The TSEMAC peripheral instance in the SmartFusion2 Microcontroller Subsystem (MSS) can be configured to transfer data between the host PC and the Ethernet network at the following data rates (line speed):

- 10 Mbps
- 100 Mbps
- 1000 Mbps

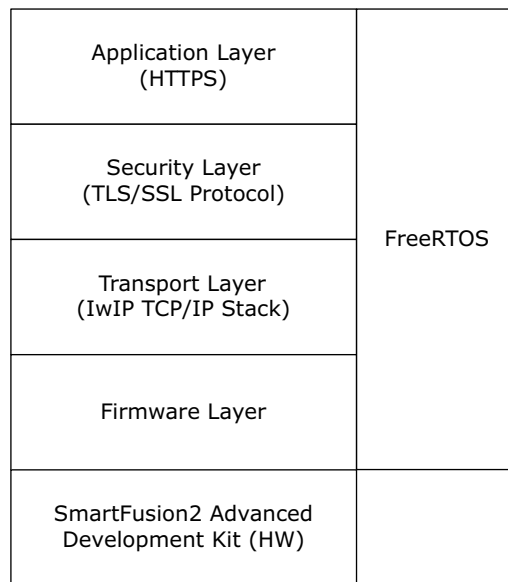
For more information about the TSEMAC interface for SmartFusion2 devices, refer to the [UG0331: SmartFusion2 Microcontroller Subsystem User Guide](#).

2.1 Secure Webserver Demo Design Overview

The secure webserver application supports TLS/SSL security protocol that encrypts and decrypts messages, securing the communication against message tampering. Communication from the secure webserver ensures that sensitive data can be translated into a secret code that makes it difficult to tamper with the data. The secure webserver demo design consists of the following layers, as shown in [Figure 1, page 3](#):

- Application Layer
- Security Layer (TLS/SSL Protocol)
- Transport Layer (lwIP TCP/IP Stack)
- RTOS and Firmware Layer

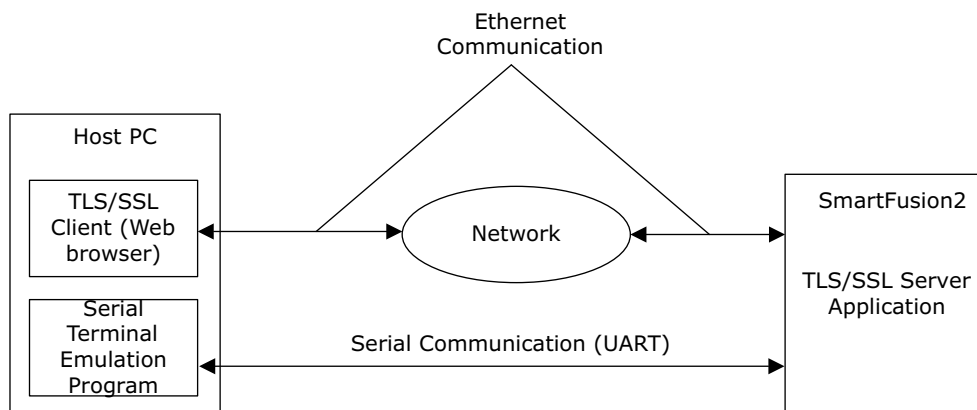
Figure 1 • Secured Webserver Layers



2.1.1 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. The following figure shows the block diagram of the connecting server (Secure webserver application running on the 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.

2.1.2 Security Layer (TLS/SSL Protocol)

Internet browsers and webserver use the TLS/SSL protocol to securely transmit information. TLS/SSL is used to authenticate the server and client to establish secure communication between authenticated parties using encryption. This protocol is layered above the transport protocol, TCP/IP as shown in Figure 1, page 3. 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>
- The TLS 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 cryptographic and TLS/SSL protocol implementations. This library provides application programming interface functions to implement a secure webserver application using the TLS/SSL protocol and the software cryptographic algorithms.

For more information about TLS/SSL protocol library source code written in C and licensing information, refer to the <https://polarssl.org/>.

2.1.3 Transport Layer (lwIP TCP/IP Stack)

The lwIP stack is suitable for embedded systems because it uses few resources, and can be used with or without an operating system. The lwIP consists of actual implementations of the IP, Internet Control Message Protocol (ICMP), User Datagram Protocol (UDP), and TCP protocols, as well as support functions such as buffer and memory management.

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

2.1.4 RTOS and Firmware Layer

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

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

- Ethernet MAC
- System controller services
- Multi-Mode universal Asynchronous/synchronous Receiver/Transmitter (MMUART)
- General Purpose Input and Output(GPIO)
- Serial Peripheral Interface (SPI)

2.2 Design Requirements

The following table lists the hardware and software design requirements for this demo design.

Table 1 • Design Requirements

Requirement	Version
Operating System	64 bit Windows 7 and 10
Hardware	
SmartFusion2 Advanced Development Kit:	Rev A or later
• 12 V adapter • FlashPro5 programmer • USB A to Mini-B cable	
Ethernet cable	RJ45
Host PC or Laptop	
Software	
FlashPro Express	Note: Refer to the <code>readme.txt</code> file provided in the design files for the software versions used with this reference design.
Libero [®] System-on-Chip (SoC) for viewing the design files	
SoftConsole	
MSS Ethernet MAC drivers	
Host PC Drivers	<i>USB to UART drivers</i>
One of the following serial terminal emulation programs:	–
• HyperTerminal • TeraTerm • PuTTY	
Browser	Mozilla Firefox version 24 or later Internet Explorer version 8 or later

Note: Libero SmartDesign and configuration screen shots shown in this guide are for illustration purpose only. Open the Libero design to see the latest updates.

2.3 Prerequisites

Before you begin:

Download and install Libero SoC (as indicated in the website for this design) on the host PC from the following location.

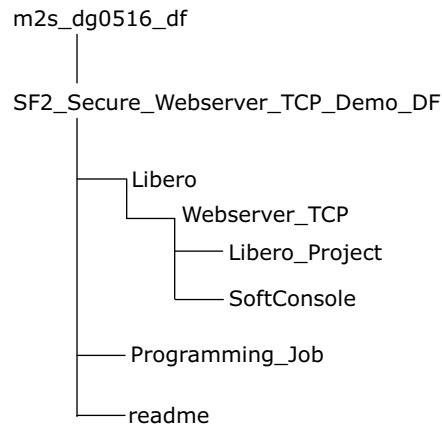
<https://www.microsemi.com/product-directory/design-resources/1750-libero-soc>

2.4 Demo Design

The demo design files are available for download from the following link:
http://soc.microsemi.com/download/rsc/?f=m2s_dg0516_df

The following figure 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



2.4.1 Demo Design Features

The demo design has the following options:

- Blinking LEDs
- HyperTerminal Display
- SmartFusion2 Google Search

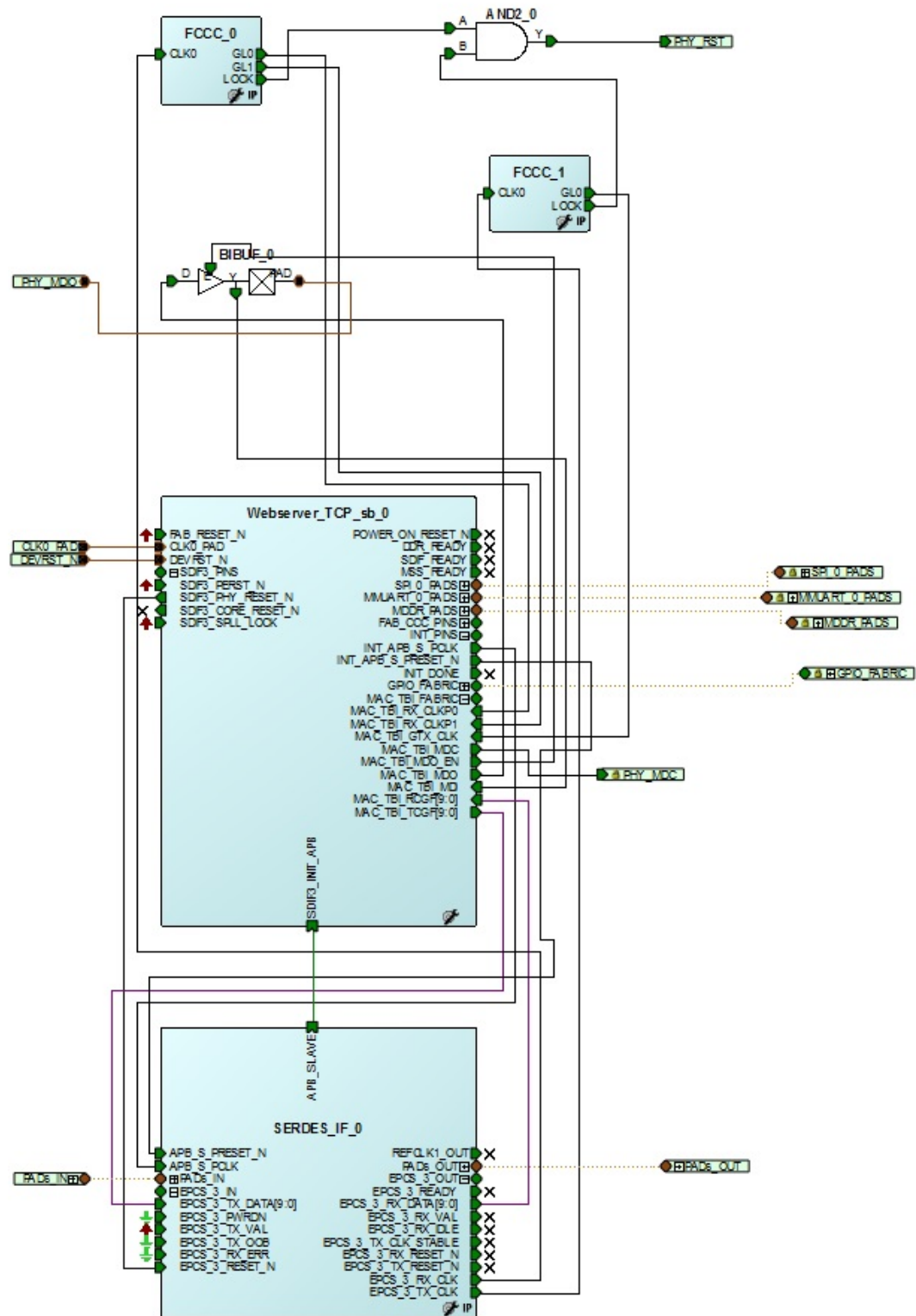
2.4.2 Demo Design Description

The demo design is implemented using an SGMII PHY interface by configuring the TSEMAC for the Ten-Bit Interface (TBI) operation. For more information about the TSEMAC TBI interface, refer to the *UG0331: SmartFusion2 Microcontroller Subsystem User Guide*.

2.4.2.1 Libero SoC Hardware Project

The following figure shows the Libero SoC hardware design implementation for this demo design.

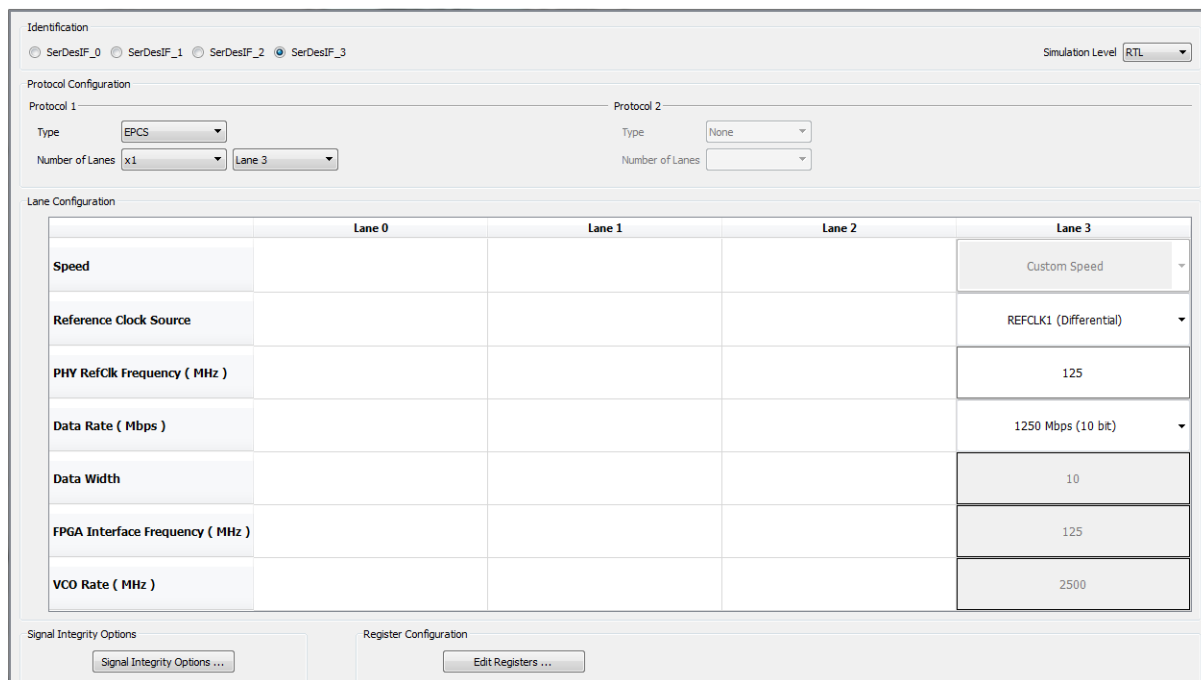
Figure 4 • Libero SoC Top-Level Hardware Design



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

- **TSEMAC TBI** interface.
- **MMUART_0** for RS-232 communications on the SmartFusion2 Advanced Development Kit.
- **GPIO**: Interfaces with the light-emitting diodes (LEDs)
- Dedicated input pad 0 as the clock source
- **High speed serial interface (SERDESIF) SERDES_IF IP**: Configured for **SERDESIF_3 EPCS lane3**, as shown in the following figure.
For more information about 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



The screenshot shows the 'High-Speed Serial Interface Configurator' window. It has a 'Simulation Level' dropdown set to 'RTL'. Under 'Identification', 'SerDesIF_3' is selected. The 'Protocol Configuration' section shows 'Protocol 1' with 'Type' set to 'EPCS' and 'Number of Lanes' set to 'x1' and 'Lane 3'. 'Protocol 2' is set to 'None'. The 'Lane Configuration' table shows settings for Lane 3: Speed (Custom Speed), Reference Clock Source (REFCLK1 (Differential)), PHY RefClk Frequency (125 MHz), Data Rate (1250 Mbps (10 bit)), Data Width (10), FPGA Interface Frequency (125 MHz), and VCO Rate (2500 MHz). At the bottom, there are buttons for 'Signal Integrity Options ...' and 'Edit Registers ...'.

1. **Cryptographic system controller services**: To implement TLS/SSL protocol.

2.4.2.1.1 Package Pin Assignments

Package pin assignments for LEDs and PHY interface signals are shown in the following tables.

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 • PHY Interface Signals to Package Pins Assignments

Port Name	Direction	Package Pin
PHY_MDC	Output	F3
PHY_MDIO	Input	K7
PHY_RST	Output	F2

2.4.2.2 SoftConsole Firmware Project

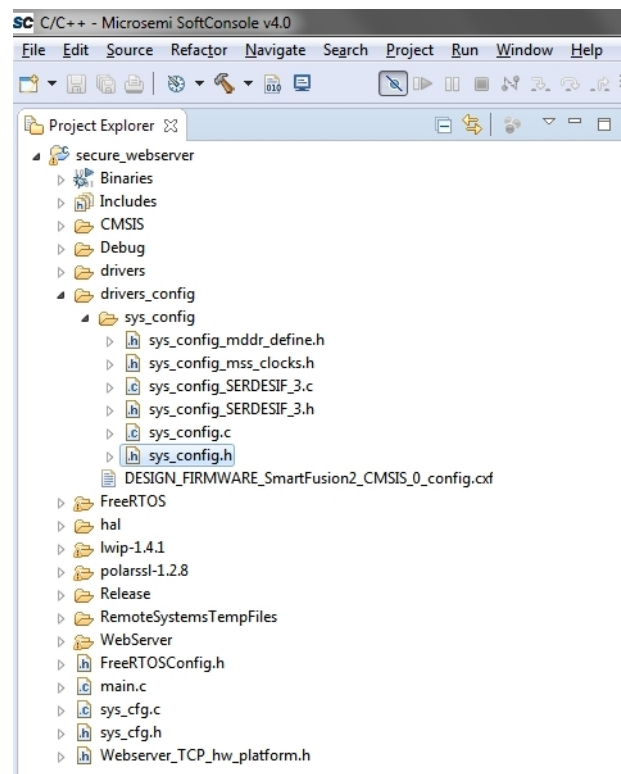
Invoke the SoftConsole project using standalone SoftConsole IDE.

The following stacks are used for this demo design:

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

The following figure shows an example of a SoftConsole software directory structure of the demo design.

Figure 6 • SoftConsole Project Explorer Window



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 the 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	<\$Design_Files_Directory>\m2s_dg0516_d\SF2_Secure_Webserver_TCP_Demo_DF\Libero\Webserver_TCP\SoftConsole\Webserver_TCP_MSS_CM3\polarssl-1.2.8\include\polarssl\aes.h
NRBG	#define HW_NRBG 1	<\$Design_Files_Directory>\m2s_dg0516_d\SF2_Secure_Webserver_TCP_Demo_DF\Libero\Webserver_TCP\SoftConsole\Webserver_TCP_MSS_CM3\polarssl-1.2.8\include\polarssl\ssl.h

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

The following figure shows the driver versions used for the demo.

Figure 7 • Demo Design Driver Versions

	Generate	Instance Name	Core Type	Version	Compatible Hardware Instance
1	☒	SmartFusion2_CMSTIS_0	SmartFusion2_CMSTIS	2.3.105	Webserver_TCP_sb_MSS
2	☒	SmartFusion2_MSS_Ethernet_MAC_Driver_0	SmartFusion2_MSS_Ethernet_MAC_Driver	3.1.100	Webserver_TCP_sb_MSS:MAC
3	☒	SmartFusion2_MSS_GPIO_Driver_0	SmartFusion2_MSS_GPIO_Driver	2.1.102	Webserver_TCP_sb_MSS:GPIO
4	☐	SmartFusion2_MSS_HPDM_A_Driver_0	SmartFusion2_MSS_HPDM_A_Driver	2.2.100	Webserver_TCP_sb_MSS
5	☒	SmartFusion2_MSS_MMUART_Driver_0	SmartFusion2_MSS_MMUART_Driver	2.1.100	Webserver_TCP_sb_MSS:MMUART_0
6	☐	SmartFusion2_MSS_NVM_Driver_0	SmartFusion2_MSS_NVM_Driver	2.4.100	Webserver_TCP_sb_MSS
7	☐	SmartFusion2_MSS_PDMA_Driver_0	SmartFusion2_MSS_PDMA_Driver	2.0.102	Webserver_TCP_sb_MSS:DMA
8	☒	SmartFusion2_MSS_RTC_Driver_0	SmartFusion2_MSS_RTC_Driver	2.2.100	Webserver_TCP_sb_MSS:RTC
9	☒	SmartFusion2_MSS_SPI_Driver_0	SmartFusion2_MSS_SPI_Driver	2.2.101	Webserver_TCP_sb_MSS:SPI_0
10	☒	SmartFusion2_MSS_System_Services_Driver_0	SmartFusion2_MSS_System_Services_Driver	2.9.100	Webserver_TCP_sb_MSS
11	☒	SmartFusion2_MSS_Timer_Driver_0	SmartFusion2_MSS_Timer_Driver	2.2.100	Webserver_TCP_sb_MSS
12	☐	SmartFusion2_MSS_Watchdog_Driver_0	SmartFusion2_MSS_Watchdog_Driver	2.1.100	Webserver_TCP_sb_MSS:WATCHDOG

2.4.2.2.1 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

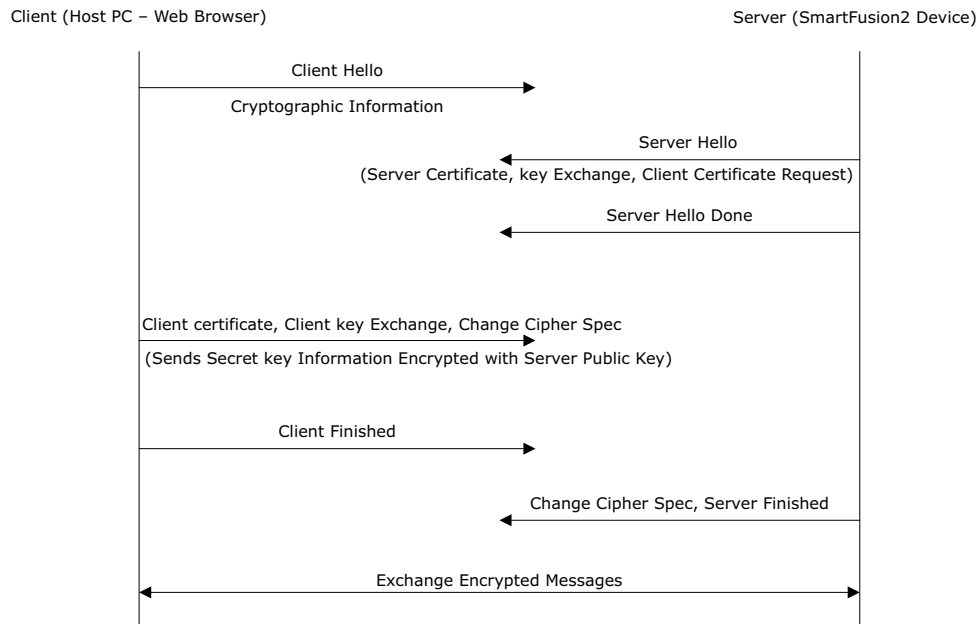
This layer consists of the following sub protocols:

- **Handshake:** Used to negotiate session information between the server and the client. The session information includes the 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.

The following figure shows the overview of the TLS/SSL handshake procedure.

For more information about handshake protocol, record protocol, and cryptographic algorithms, refer to the <http://tools.ietf.org/html/rfc5246>.

Figure 8 • TLS/SSL Handshake Procedure



Record Protocol Layer

The record protocol receives and encrypts data from the application and transfers it 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 keyed-hash message authentication code (HMAC) and encrypts or decrypts the data using the information negotiated during the handshake protocol.

2.5 Setting Up the Demo Design

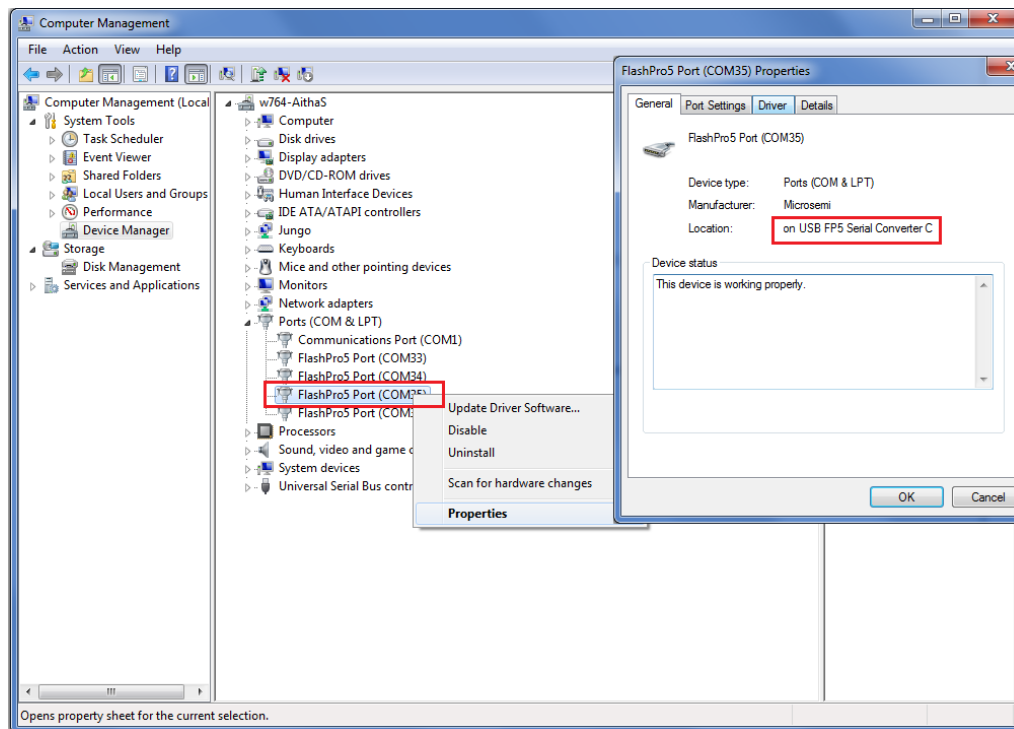
The following steps describe how to set up the demo for the SmartFusion2 Advanced Development Kit board:

1. Connect the host PC to the **J33 Connector** using the USB A to mini-B cable. The USB to universal asynchronous receiver/transmitter (UART) bridge drivers are automatically detected.

Note: If the COM ports are not detected automatically, install the FTDI D2XX driver for serial terminal communication through the FTDI mini-USB cable. The driver, along with the installation guide, is available at www.microsemi.com/soc/documents/CDM_2.08.24_WHQL_Certified.zip.

2. Right-click each of the four detected COM ports, and click **Properties** to find the port with the location on **USB FP5 Serial Converter C**, as shown in the following figure. Make a note of the COM port number for use during serial terminal configuration, as shown in the following figure.

Figure 9 • Device Manager Window



3. Connect the jumpers on the SmartFusion2 Advanced Development Kit board, as shown in the following table. For information about jumper locations, refer to [Appendix 3: Jumper Locations](#), page 23.

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

4. In the SmartFusion2 Advanced Development Kit, connect the power supply to the **J42 connector**.
5. 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.

2.5.1 Board Setup Snapshot

Snapshots of the SmartFusion2 Advanced Development Kit board with all the configured setup is given in [Appendix 2: Board Setup for Running the Secure Webserver](#), page 22.

2.6 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_df
2. Switch ON the **SW7** power supply switch.
3. Start any serial terminal emulation programs such as:
 - HyperTerminal
 - PuTTY
 - TeraTerm

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

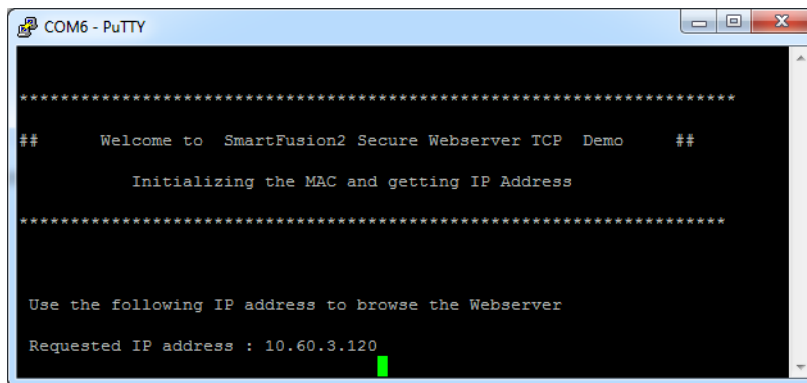
For more information about configuring the serial terminal emulation programs, refer to the [Configuring Serial Terminal Emulation Programs Tutorial](#).

4. Program the SmartFusion2 Advanced Development Kit board with the job file provided as part of the design files using FlashPro Express software, refer to [Appendix 1: Programming the Device Using FlashPro Express](#), page 19.

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 4: Running the Design in Static IP Mode](#), page 24.

5. 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 the following figure.

Figure 10 • User Options



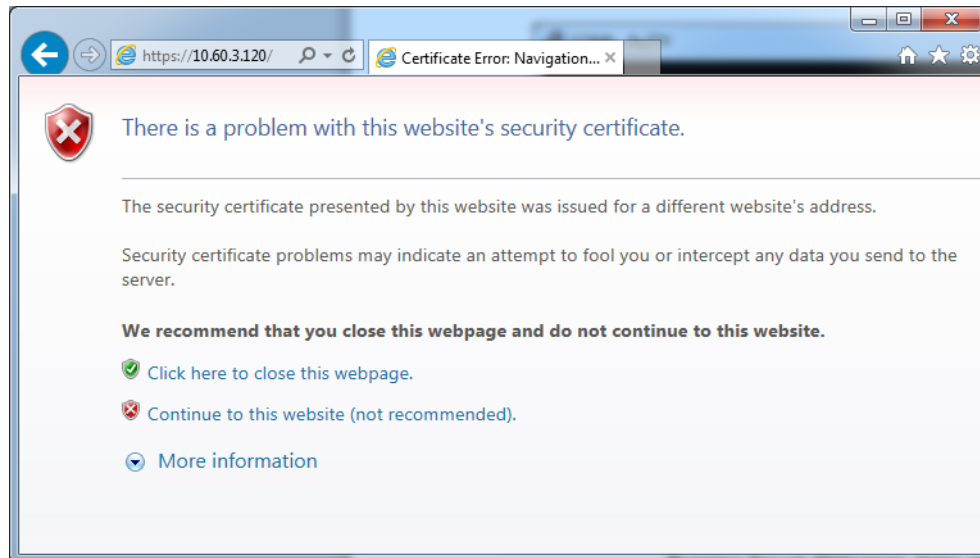
6. The IP address displayed on PuTTY should be entered in the address bar of the browser to run the secure webserver. If the IP address is 10.60.3.120, enter <https://10.60.3.120> in the address bar of the browser. This demo supports both Microsoft Internet Explorer and Mozilla Firefox browsers.

2.6.1 Running the Secure Webserver Demo with Microsoft Internet Explorer

The following steps describe how to run the secure webserver demo with Microsoft Internet Explorer:

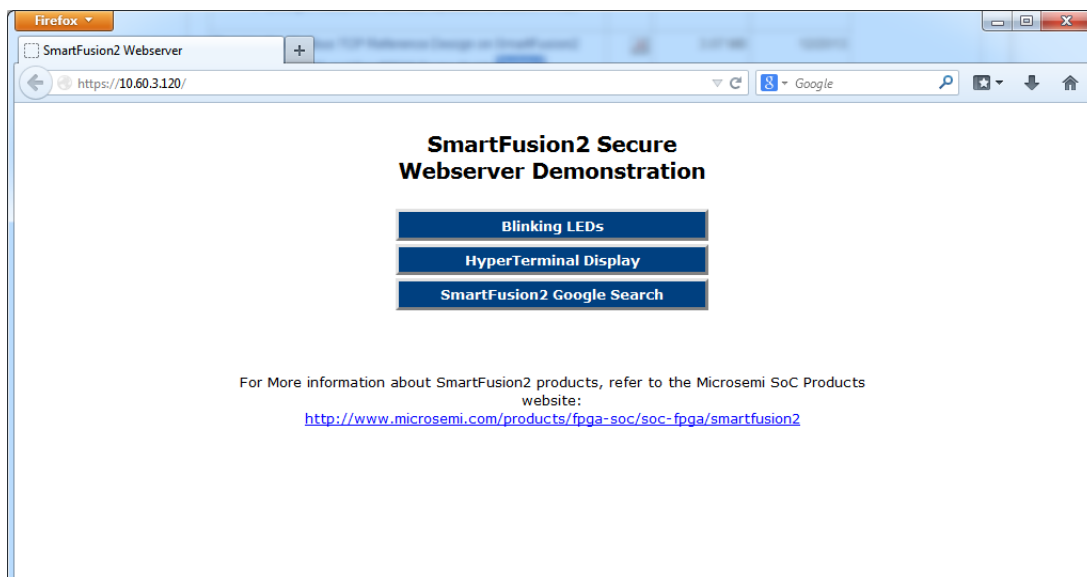
1. Open the Microsoft Internet Explorer and type the URL (for example, <https://10.60.3.120>) in the address bar. The browser shows a warning message, as shown in the following figure.

Figure 11 • 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 the following figure.

Figure 12 • Main Menu of Secure Webserver in Internet Explorer

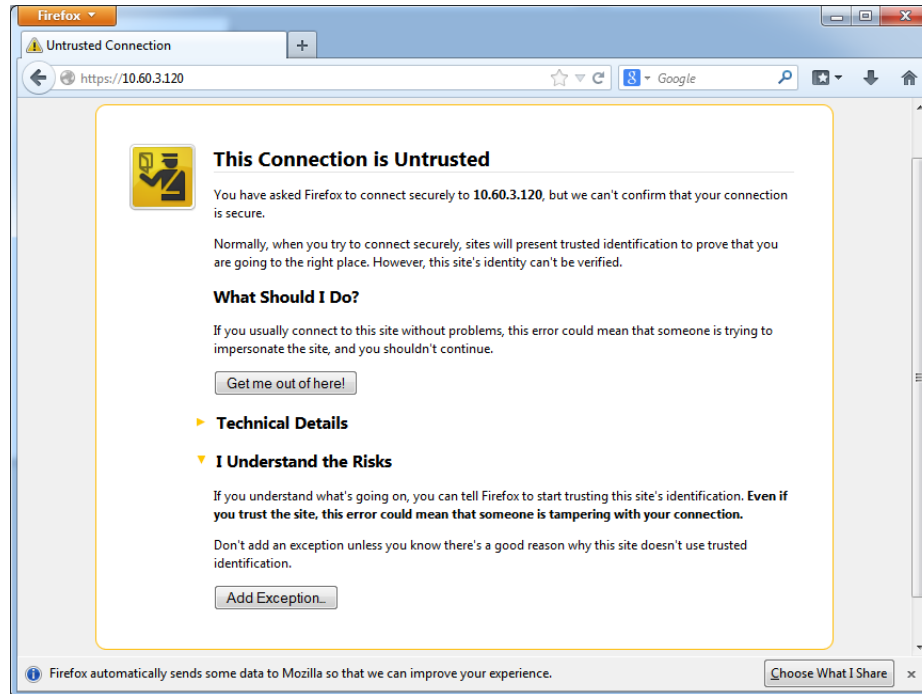


2.6.2 Running the Secure Webserver Demo with Mozilla Firefox

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 the following figure.

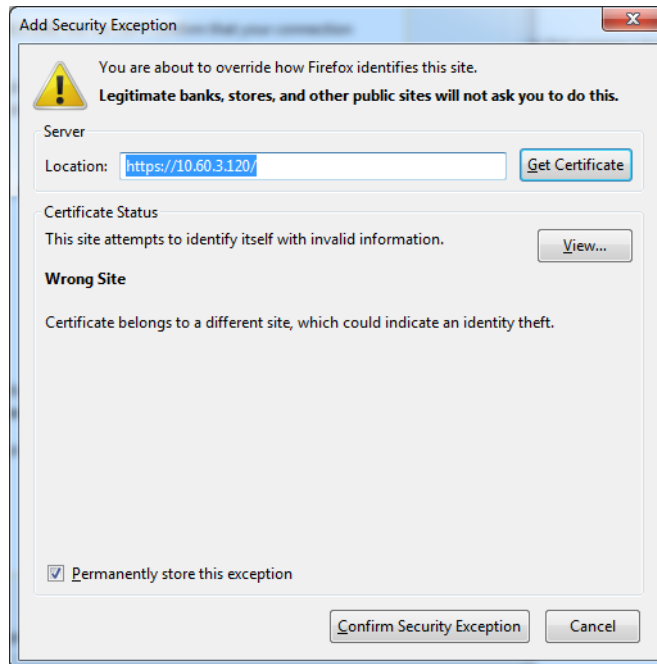
Figure 13 • Mozilla Firefox showing Warning Message



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

3. Click **Confirm Security Exception** in **Add Security Exception** window, as shown in the following figure, to start secure communication with the webserver.

Figure 14 • Add Security Exception Window

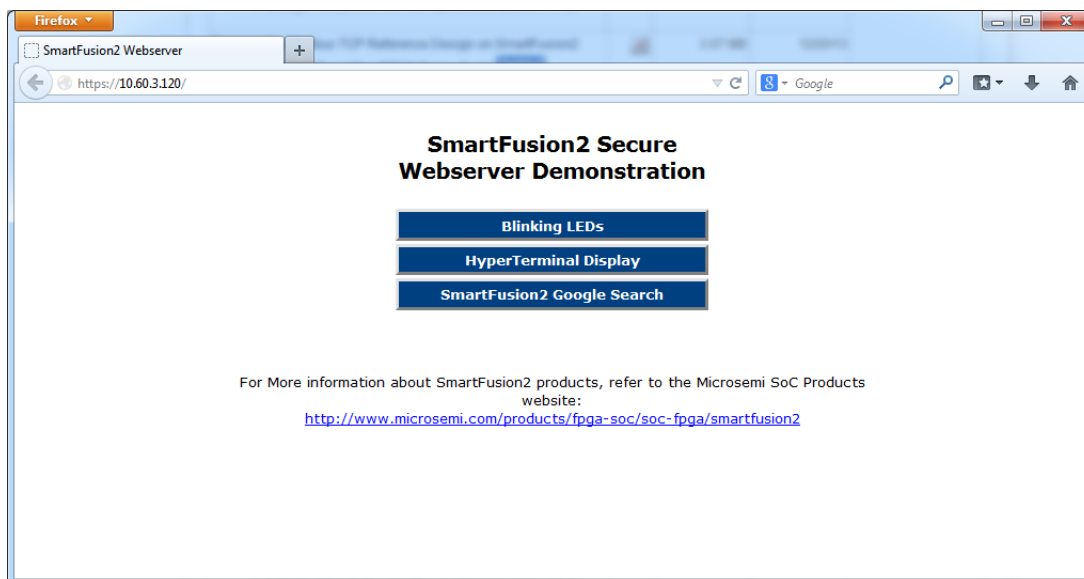


Note: Adding security exception for the IP Address is required for first-time browsing only.

Note: If you get any handshake failed message in the terminal, ignore that message.

4. The Mozilla Firefox browser displays the main menu, as shown in the following figure.

Figure 15 • Main Menu of the Secure Webserver in Mozilla Firefox



The main menu has the following options:

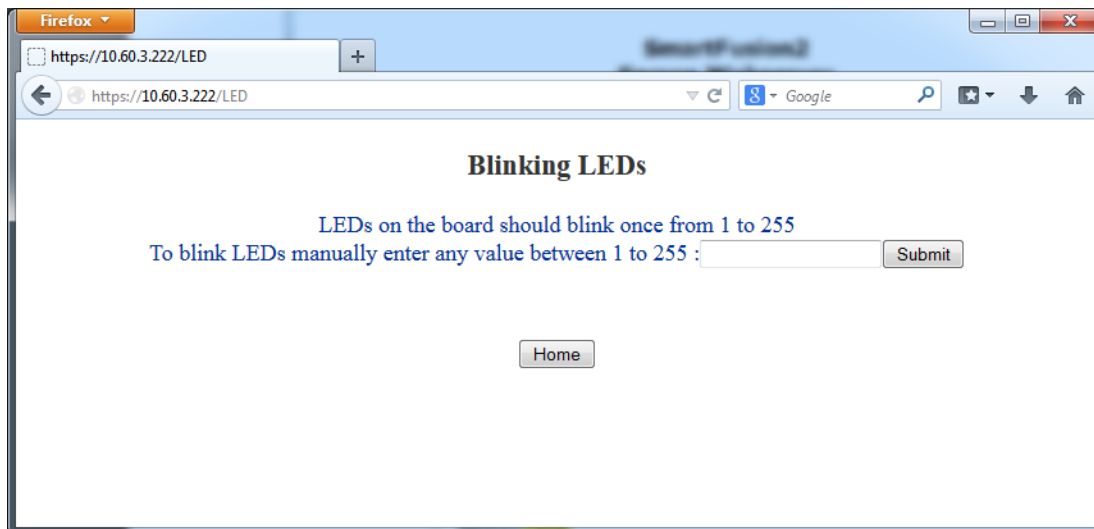
- Blinking LEDs
- HyperTerminal Display
- SmartFusion2 Google Search

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.

2.6.2.1 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 the following figure.

Figure 16 • Blinking LEDs Page



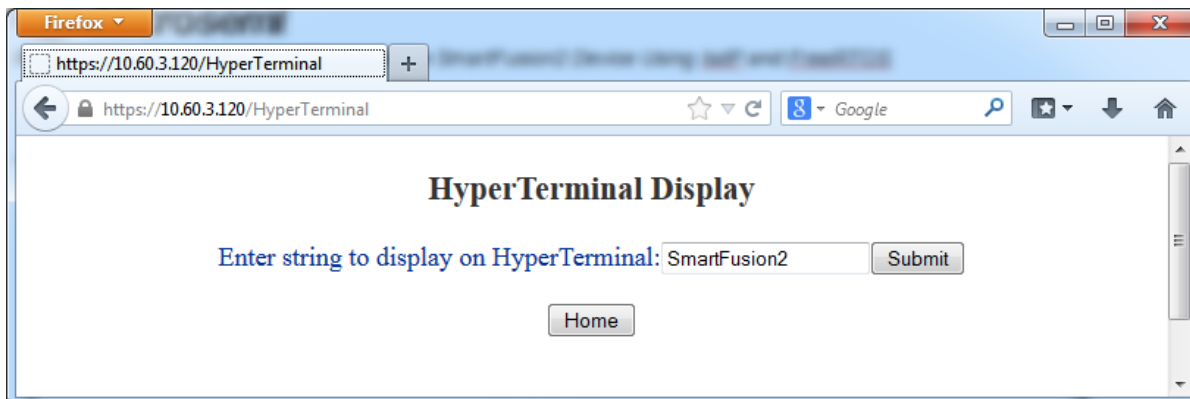
2. Enter any number between 1-255 to light up the LEDs manually. For example, if you enter 1, blinking LED1 goes OFF. If you enter 255, all the eight blinking LEDs go OFF.
3. Click **Home** to return to the main menu.

Note: SmartFusion2 Advanced Development Kit has active low LEDs.

2.6.2.2 HyperTerminal Display

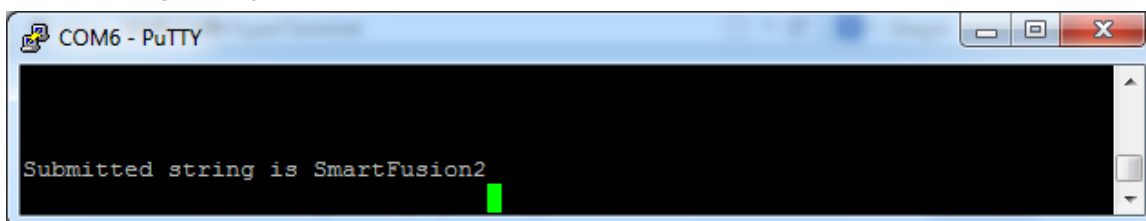
1. Click **HyperTerminal Display** on the main menu. The following figure shows a webpage that gives an option to enter a string value.

Figure 17 • HyperTerminal Display Page



The entered string is displayed on PuTTY, as shown in the following figure.

Figure 18 • String Display on PuTTY



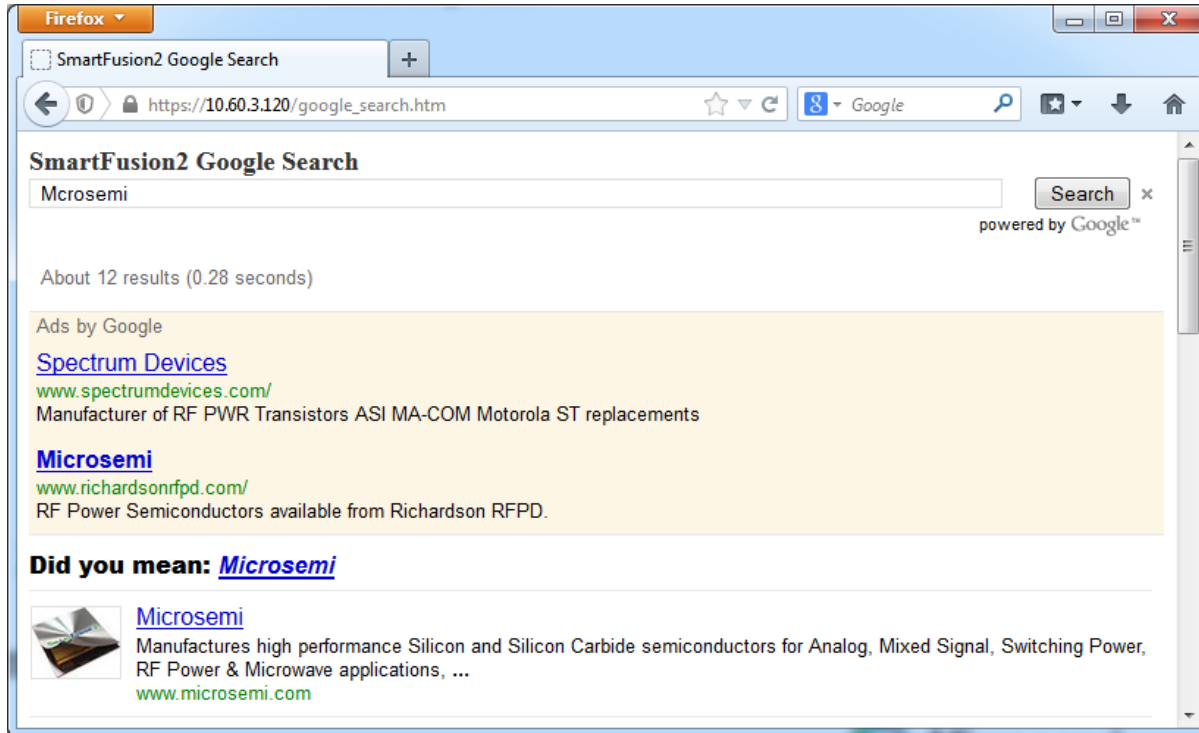
2. Click **Go Back One Page** (arrow button) or **Home** to go back to the main menu.

2.6.2.3 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. The following figure shows a web page with Google search.

Figure 19 • SmartFusion2 Google Search Page



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

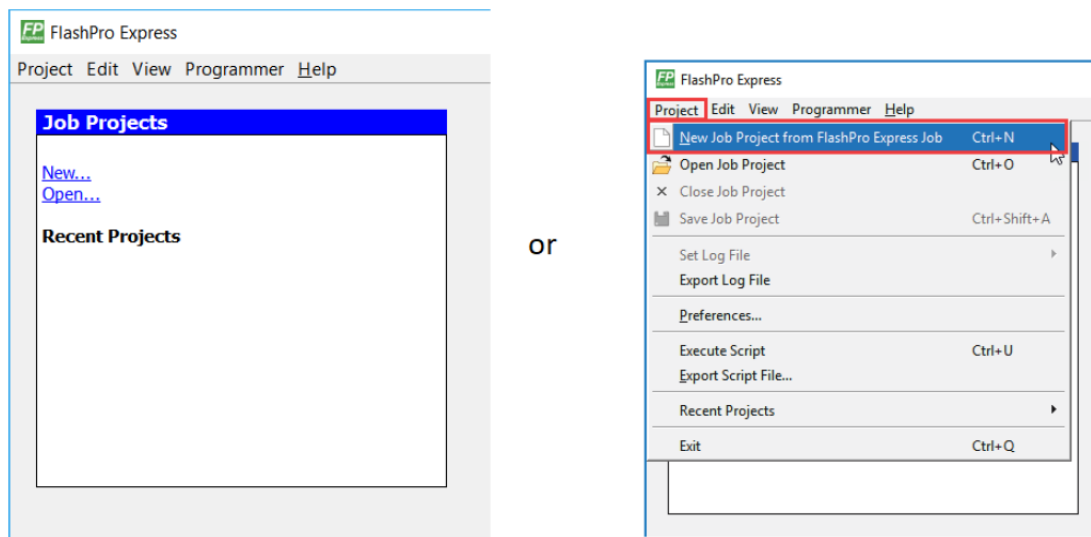
3 Appendix 1: Programming the Device Using FlashPro Express

This section describes how to program the SmartFusion2 device with the programming job file using FlashPro Express.

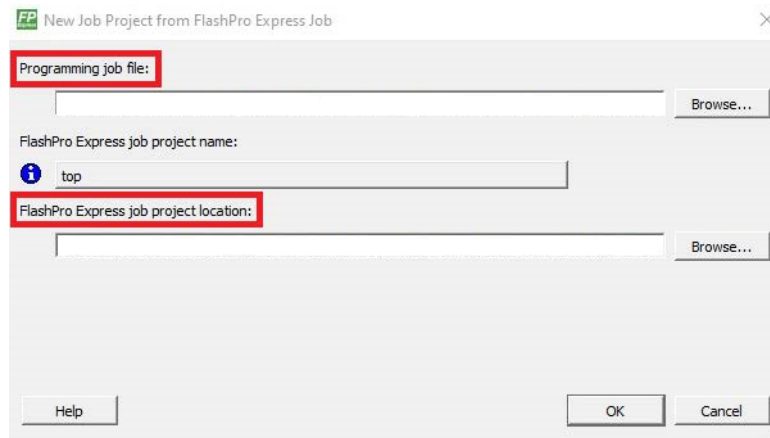
To program the device, perform the following steps:

1. Ensure that the jumper settings on the board are the same as those listed in Table 5, page 12.
- Note:** The power supply switch must be switched off while making the jumper connections.
2. Connect the power supply cable to the **J42** connector on the board.
 3. Power **ON** the power supply switch **SW7**.
 4. On the host PC, launch the **FlashPro Express** software.
 5. Click **New** or select **New Job Project from FlashPro Express Job** from **Project** menu to create a new job project, as shown in the following figure.

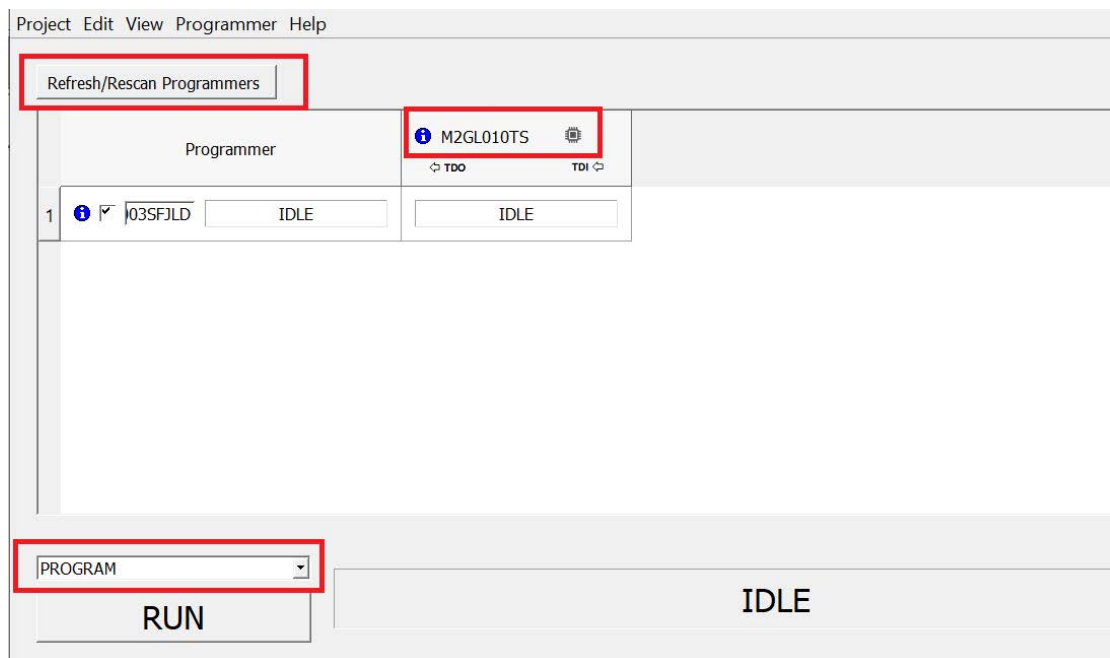
Figure 20 • FlashPro Express Job Project



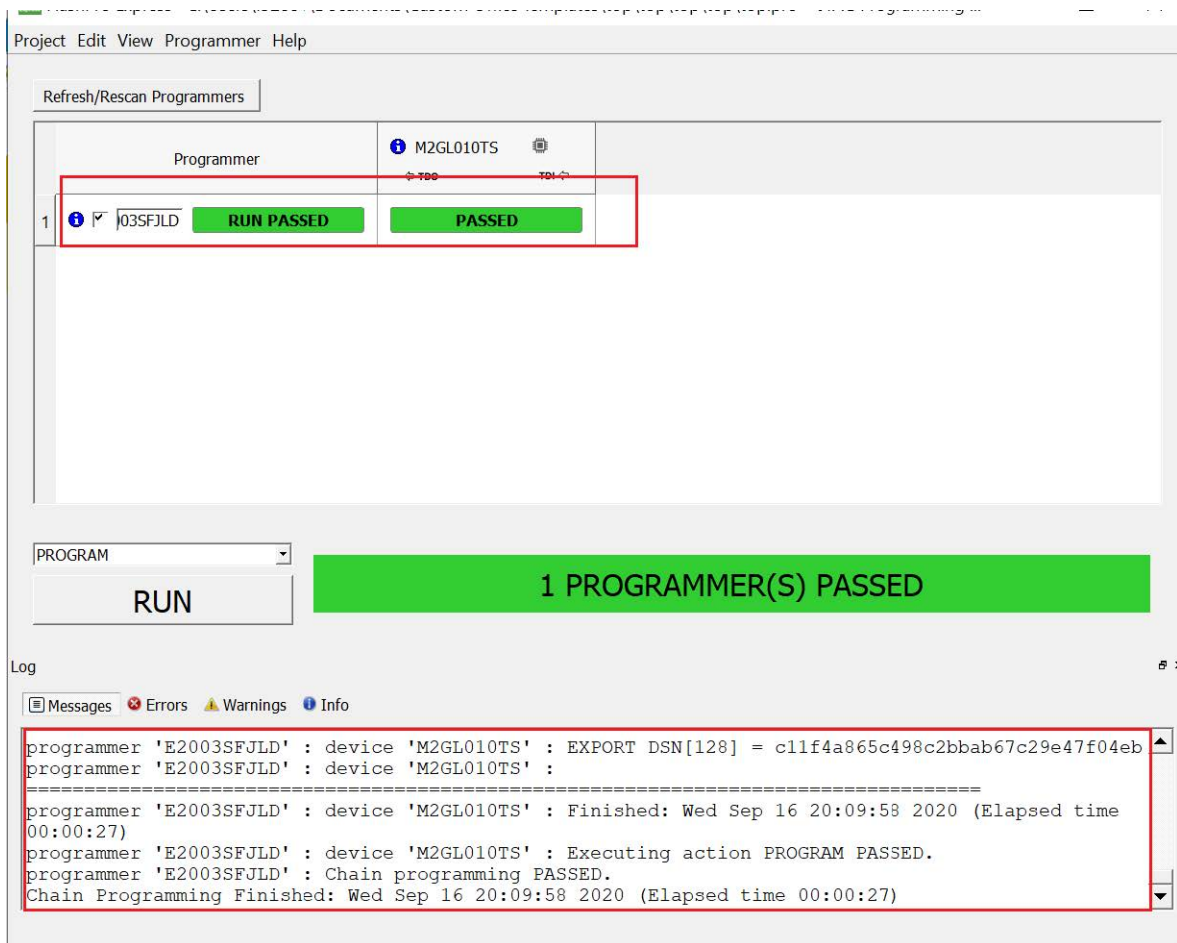
6. Enter the following in the **New Job Project from FlashPro Express Job** dialog box:
 - **Programming job file:** Click **Browse**, and navigate to the location where the .job file is located and select the file. The default location is:
`<download_folder>\m2s_dg0516_df\SF2_Secure_Webserver_TCP_Demo_DF\Programming_Job`
 - **FlashPro Express job project name:** Click **Browse** and navigate to the location where you want to save the project.

Figure 21 • New Job Project from FlashPro Express Job

7. Click **OK**. The required programming file is selected and ready to be programmed in the device.
8. The FlashPro Express window appears as shown in the following figure. Confirm that a programmer number appears in the Programmer field. If it does not, confirm the board connections and click **Refresh/Rescan Programm**ers.

Figure 22 • Programming the Device

9. Click **RUN**. When the device is programmed successfully, a **RUN PASSED** status is displayed as shown in the following figure.

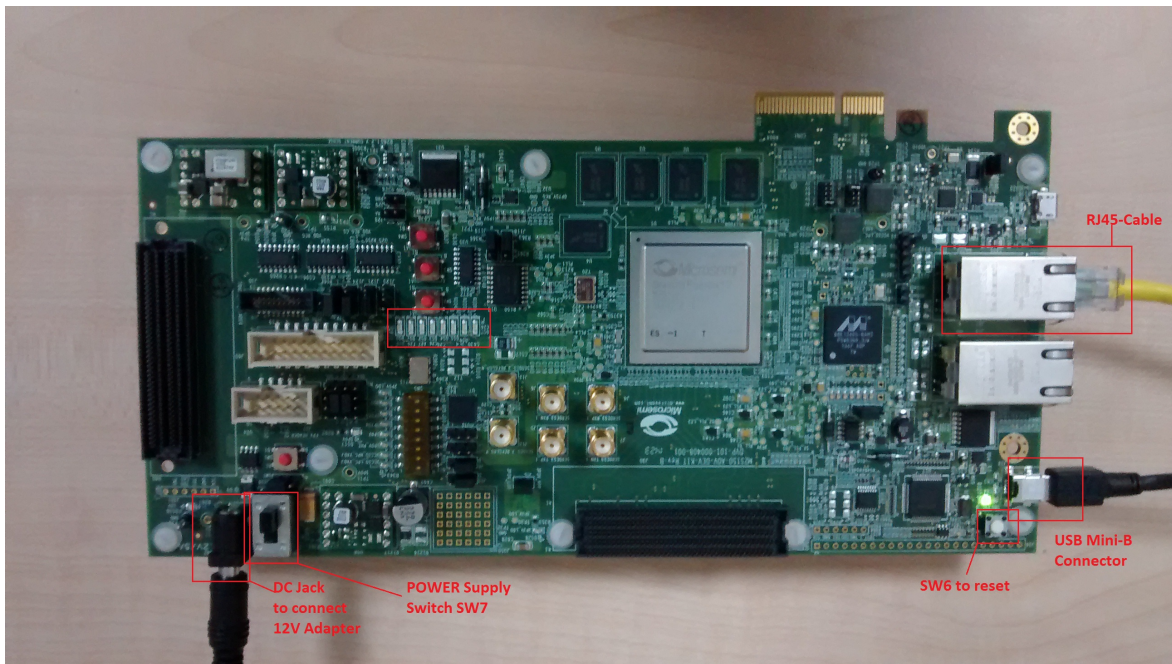
Figure 23 • FlashPro Express—RUN PASSED

10. Close **FlashPro Express** or in the Project tab, click **Exit**.

4 Appendix 2: Board Setup for Running the Secure Webserver

The following figure shows the board setup for running the demo on the SmartFusion2 Advanced Development Kit board.

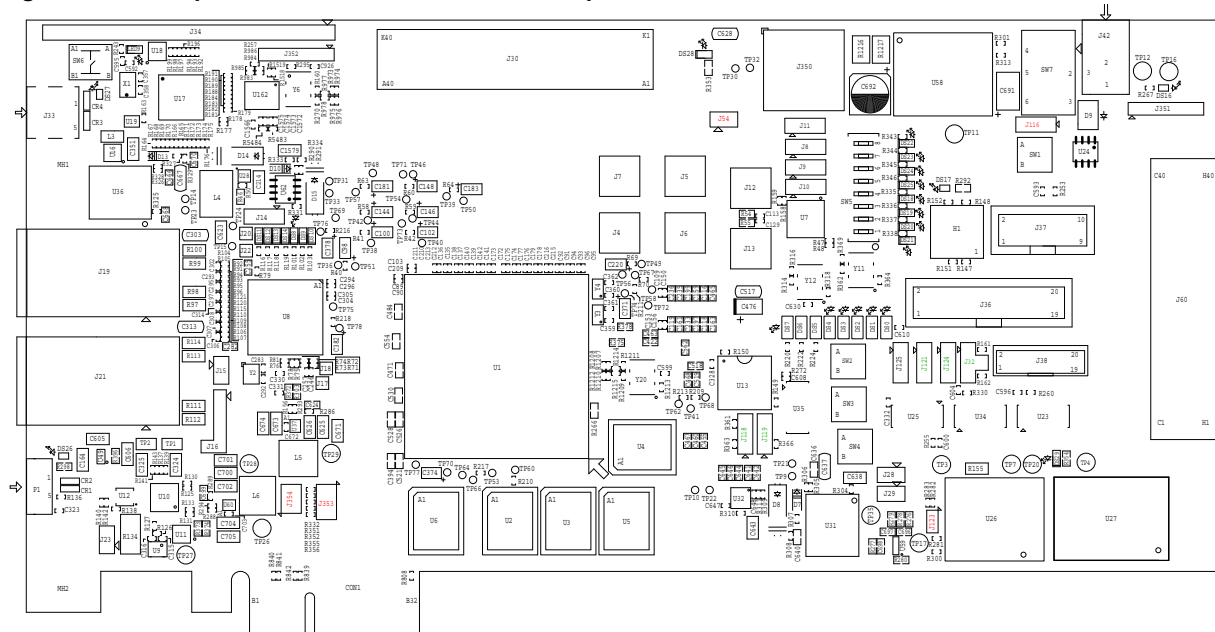
Figure 24 • SmartFusion2 Advanced Development Kit Setup



5 Appendix 3: Jumper Locations

The following figure shows the jumper locations in the SmartFusion2 Advanced Development Kit board.

Figure 25 • Jumper Locations in Advanced Development Kit Board

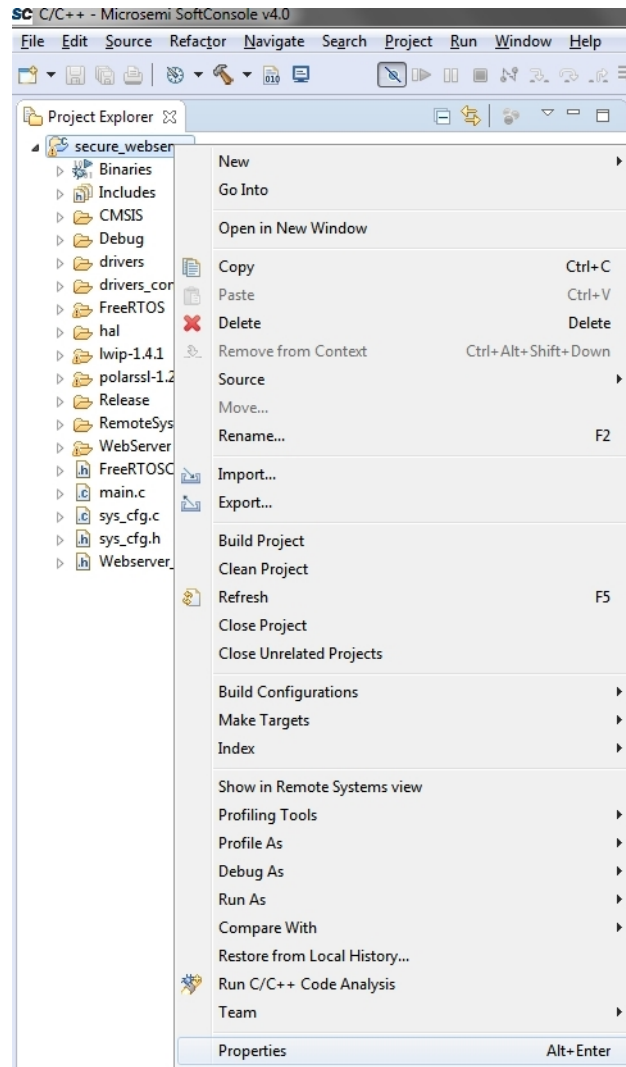


6 Appendix 4: Running the Design in Static IP Mode

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

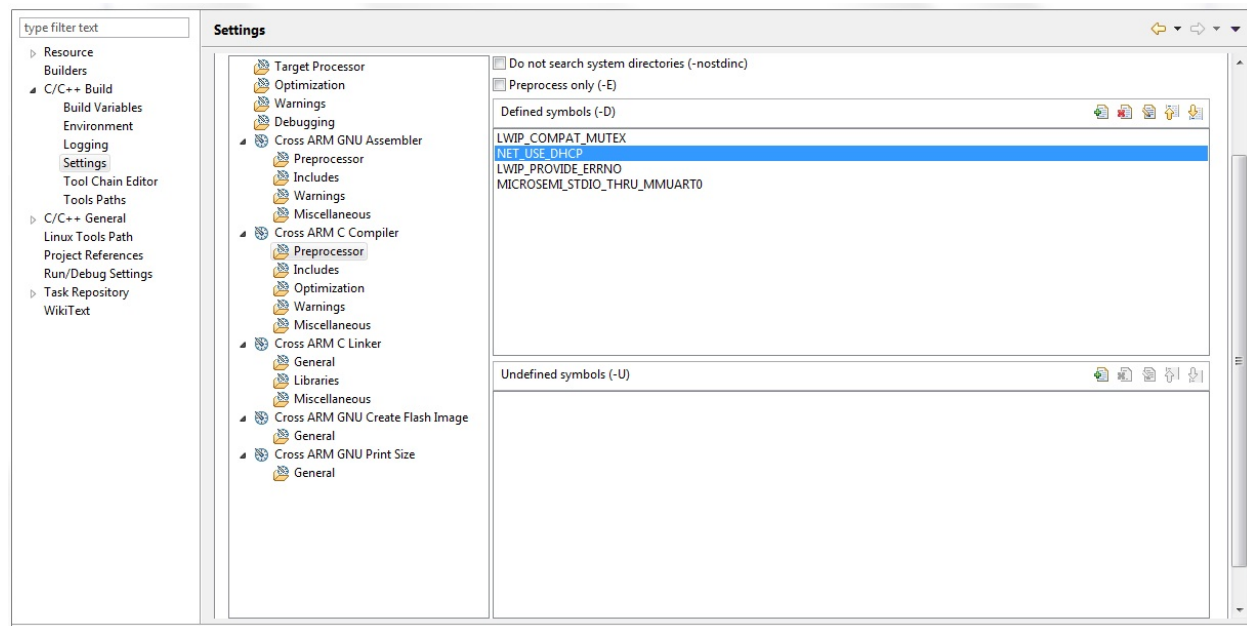
1. Right-click the **secure_webserver** in the **Project Explorer** window of SoftConsole project and select **Properties**, as shown in the following figure.

Figure 26 • Project Explorer Window of SoftConsole Project



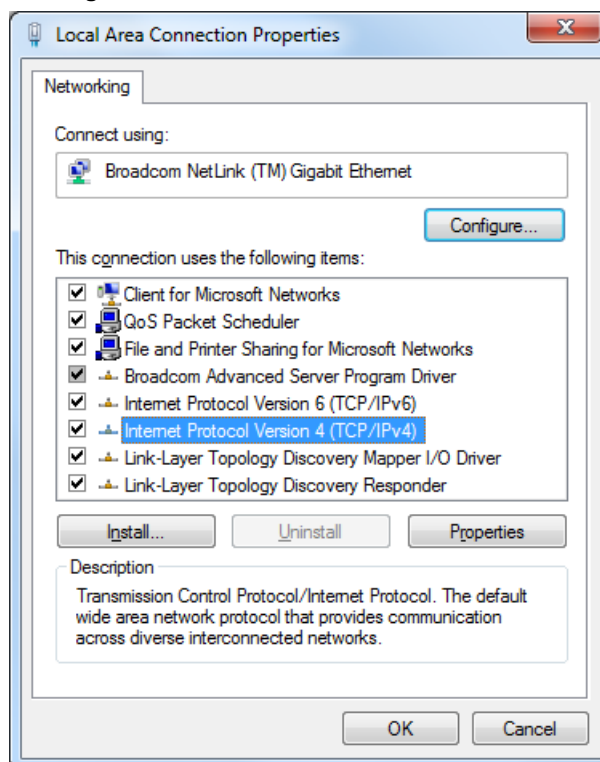
The following figure shows removing the symbol **NET_USE_DHCP** in the **Tool Settings** tab of the **Properties for secure_webserver** window.

Figure 27 • Project Explorer Properties Window



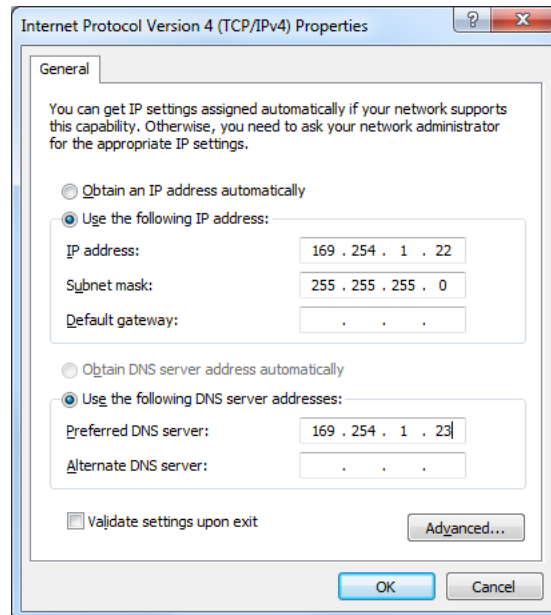
If the device is connected in **static IP** mode, the board static IP address is 169.254.1.23, then change the host TCP/IP settings to reflect the IP address. The following figure shows host PC TCP/IP settings.

Figure 28 • Host PC TCP/IP Settings



The following figure shows static IP address settings.

Figure 29 • Static IP Address Settings



Once these settings are configured, build the firmware, import the latest .hex file into eNVM, and run the Libero design. See [Running the Demo Design](#), page 13 to execute the design in static IP mode, if the SmartFusion2 device is already programmed with top_static.job file.

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