DG0516
Demo Guide
Running Secure Webserver on SmartFusion2 Devices using PolarSSL, lwIP, and FreeRTOS - Libero SoC v11.8 SP1
# Contents

1 Revision History ................................................................. 1
  1.1 Revision 8.0 ................................................................. 1
  1.2 Revision 7.0 ................................................................. 1
  1.3 Revision 6.0 ................................................................. 1
  1.4 Revision 5.0 ................................................................. 1
  1.5 Revision 4.0 ................................................................. 1
  1.6 Revision 3.0 ................................................................. 1
  1.7 Revision 2.0 ................................................................. 1
  1.8 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 ......................................................... 4
  2.3 Demo Design ................................................................. 5
      2.3.1 Demo Design Features ................................................ 5
      2.3.2 Demo Design Description ............................................ 5
  2.4 Setting Up the Demo Design ................................................ 11
      2.4.1 Board Setup Snapshot ............................................... 12
  2.5 Running the Demo Design ................................................... 12
      2.5.1 Running the Secure Webserver Demo with Microsoft Internet Explorer .................................................. 15
      2.5.2 Running the Secure Webserver Demo with Mozilla Firefox .................................................. 16

3 Appendix: Board Setup for Running the Secure Webserver ..................... 20

4 Appendix: Jumper Locations ..................................................... 21

5 Appendix: Running the Design in Static IP Mode ................................ 22
### Figures

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

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>Design Requirements</td>
<td>4</td>
</tr>
<tr>
<td>Table 2</td>
<td>LED to Package Pins Assignments</td>
<td>8</td>
</tr>
<tr>
<td>Table 3</td>
<td>PHY Interface Signals to Package Pins Assignments</td>
<td>8</td>
</tr>
<tr>
<td>Table 4</td>
<td>Macros to Enable or Disable System Controller Services</td>
<td>9</td>
</tr>
<tr>
<td>Table 5</td>
<td>SmartFusion2 Advanced Kit Jumper Settings</td>
<td>11</td>
</tr>
</tbody>
</table>
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 8.0
Updated the document for Libero v11.8 SP1 software release.

1.2 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 4.
• Throughout the guide, the names of SoftConsole projects used in the demo design and all the associated figures are updated.

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

1.4 Revision 5.0
Updated SoftConsole Firmware Project, page 8 (SAR 73518).

1.5 Revision 4.0
Updated the document for Libero v11.6 software release (SAR 72058).

1.6 Revision 3.0
Updated the document for Libero v11.5 software release (SAR 63973).

1.7 Revision 2.0
Updated the document for Libero v11.4 software release (SAR 60685).

1.8 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
SmartFusion2 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 of 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

See the UG0331: SmartFusion2 Microcontroller Subsystem User Guide for more information about the
TSEMAC interface for SmartFusion2 devices.

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
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 for 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 SmartFusion2 device) and client (web browser running on host PC).

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 webservers use 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.

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


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.

See https://polarssl.org/ for TLS/SSL protocol library source code written in C and licensing information.

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. See http://www.freertos.org for more information and the latest source code.

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.

<table>
<thead>
<tr>
<th>Table 1 • Design Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
</tr>
<tr>
<td>Hardware</td>
</tr>
<tr>
<td>SmartFusion2 Advanced Development Kit:</td>
</tr>
<tr>
<td>– 12 V adapter</td>
</tr>
<tr>
<td>– FlashPro5</td>
</tr>
<tr>
<td>– USB A to Mini-B cable</td>
</tr>
<tr>
<td>Ethernet cable</td>
</tr>
<tr>
<td>Host PC or Laptop</td>
</tr>
</tbody>
</table>
2.3 Demo Design

The demo design files are available for download from the following link:
http://soc.microsemi.com/download/rsc/?f=m2s_dg0516_liberov11p8_sp1_df

The following figure shows the top-level structure of the design files. For further details, see the Readme.txt file.

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

```
<m2s_dg0516_liberov11p8_sp1_df.rar>
   m2s_dg0516_liberov11p8_sp1_df
      Libero
      Stapl_Programming_File
      Readme.txt
```

### 2.3.1 Demo Design Features

The demo design has the following options:

- Blinking LEDs
- HyperTerminal Display
- SmartFusion2 Google Search

### 2.3.2 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 on the TSEMAC TBI interface, see the **UG0331: SmartFusion2 Microcontroller Subsystem User Guide**.
2.3.2.1 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:

1. **TSEMAC TBI** interface.
2. **MMUART_0** for RS-232 communications on the SmartFusion2 Advanced Development Kit.
3. **GPIO**: Interfaces with the light-emitting diodes (LEDs)
4. Dedicated input pad 0 as the clock source
5. **High speed serial interface (SERDESIF) SERDES_IF IP**: Configured for SERDESIF_3 EPCS lane3 as shown in the following figure. For more information on high-speed serial interfaces, see the UG0447: IGLOO2 and SmartFusion2 High Speed Serial Interfaces User Guide.

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

6. **Cryptographic system controller services**: To implement TLS/SSL protocol.
2.3.2.1 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**

<table>
<thead>
<tr>
<th>Port Name</th>
<th>Package Pin</th>
</tr>
</thead>
<tbody>
<tr>
<td>LED_1</td>
<td>D26</td>
</tr>
<tr>
<td>LED_2</td>
<td>F26</td>
</tr>
<tr>
<td>LED_3</td>
<td>F27</td>
</tr>
<tr>
<td>LED_4</td>
<td>C26</td>
</tr>
<tr>
<td>LED_5</td>
<td>C28</td>
</tr>
<tr>
<td>LED_6</td>
<td>B27</td>
</tr>
<tr>
<td>LED_7</td>
<td>C27</td>
</tr>
<tr>
<td>LED_8</td>
<td>E26</td>
</tr>
</tbody>
</table>

**Table 3 • PHY Interface Signals to Package Pins Assignments**

<table>
<thead>
<tr>
<th>Port Name</th>
<th>Direction</th>
<th>Package Pin</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHY_MDC</td>
<td>Output</td>
<td>F3</td>
</tr>
<tr>
<td>PHY_MDIO</td>
<td>Input</td>
<td>K7</td>
</tr>
<tr>
<td>PHY_RST</td>
<td>Output</td>
<td>F2</td>
</tr>
</tbody>
</table>

2.3.2.2 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 SmartFusion2 hardware engine or software PolarSSL library. In this demo design, AES and NRBG are implemented using SmartFusion2 hardware engine through system services.

<table>
<thead>
<tr>
<th>System Service</th>
<th>Macro</th>
<th>Macro Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES</td>
<td>#define HW_AES 1</td>
<td><code>&lt;$Design_Files_Directory\m2s_dg0516 Liberov11p8_sp1_df\Liberov\SoftConsole4.0\secure_webserver\polarssl-1.2.8\include\polarssl\aes.h</code></td>
</tr>
<tr>
<td>NRBG</td>
<td>#define HW_NRBG 1</td>
<td><code>&lt;$Design_Files_Directory\m2s_dg0516 Liberov11p8_sp1_df\Liberov\SoftConsole4.0\secure_webserver\polarssl-1.2.8\include\polarssl\ssl.h</code></td>
</tr>
</tbody>
</table>

**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.
2.3.2.2.1 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 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. See [http://tools.ietf.org/html/rfc5246](http://tools.ietf.org/html/rfc5246) for detailed information on handshake protocol, record protocol, and cryptographic algorithms.
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 keyed-hash message authentication code (HMAC) and encrypts or decrypts the data using the information negotiated during the handshake protocol.

2.4 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 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.

![Device Manager Window](image)

3. Connect the jumpers on the SmartFusion2 Advanced Development Kit board as shown in the following table. For information on jumper locations, refer to Appendix: Jumper Locations, page 21.

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

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

<table>
<thead>
<tr>
<th>Jumper</th>
<th>Pin (From)</th>
<th>Pin (To)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>J116, J353, J354, J54</td>
<td>1</td>
<td>2</td>
<td>These are the default jumper settings of the Advanced Dev Kit board. Ensure these jumpers are set accordingly.</td>
</tr>
<tr>
<td>J123</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>J124, J121, J32</td>
<td>1</td>
<td>2</td>
<td>JTAG programming via FTDI</td>
</tr>
<tr>
<td>J118, J119</td>
<td>1</td>
<td>2</td>
<td>Programming SPI Flash</td>
</tr>
</tbody>
</table>
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.

2.4.1 Board Setup Snapshot

Snapshots of the SmartFusion2 Advanced Development Kit board with all the configured setup is given in Appendix: Board Setup for Running the Secure Webserver, page 20.

2.5 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_liberov11p8_sp1_df](http://soc.microsemi.com/download/rsc/?f=m2s_dg0516_liberov11p8_sp1_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 information on configuring the serial terminal emulation programs, see 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 the following figure.

**Figure 10** • **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:
<\$Design_Files_Directory\m2s_dg0516_liberov11p8_sp1_df\Stapl_Programming_File\Webservet_TCP_top_Secure_Demo.stp>
The required programming file is selected and is ready to be programmed in the device, as shown in the following figure.

**Figure 11 • FlashPro Project Configured**

![FlashPro Project Configured](image)

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

**Figure 12 • FlashPro Program Passed**

![FlashPro Program Passed](image)

**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: Running the Design in Static IP Mode, page 22.
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 the following figure.

   **Figure 13 • User Options**

   ![User Options with dynamic IP address](image)

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.

### 2.5.1 Running the Secure Webserver Demo with Microsoft Internet Explorer

The following steps describe how to run the secure webserver demo with the 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 14 • Microsoft Internet Explorer showing Certificate Error Warning Message**

   ![Certificate Error Warning Message](image)
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 15 • Main Menu of Secure Webserver in Internet Explorer*

---

### 2.5.2 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 16 • 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 17 • Add Security Exception Window**

![Add Security Exception](image)

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

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

**Figure 18 • Main Menu of the Secure Webserver in Mozilla Firefox**

![Main Menu](image)

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.
2.5.2.1 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.

![Blinking LEDs Page](image)

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.

2.5.2.2 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.

![HyperTerminal Display Page](image)

2. Click **Go Back One Page** (arrow button) or **Home** to go back to the main menu.
2.5.2.3 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.

   ![SmartFusion2 Google Search Page](image)

2. Click Home to go back to the main menu.
The following figure shows the board setup for running the demo on the SmartFusion2 Advanced Development Kit board.

*Figure 23 • SmartFusion2 Advanced Development Kit Setup*
Appendix: Jumper Locations

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

**Figure 24 • Jumper Locations in Advanced Development Kit Board**

- **Note:** Jumpers highlighted in red are set by default. Jumpers highlighted in green must be set manually.
- **Note:** The location of the jumpers in the preceding figure are searchable.
Appendix: 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 25 • Project Explorer Window of SoftConsole Project**
Appendix: Running the Design in Static IP Mode

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

*Figure 26 • 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 27 • Host PC TCP/IP Settings*
The following figure shows static IP address settings.

**Figure 28 • 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 12 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.