DG0440
Demo Guide
Running Modbus TCP Reference Design on SmartFusion2 Devices using lwIP and FreeRTOS - Libero SoC v11.8

Microsemi
Power Matters.™

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

About Microsemi

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, California, and has approximately 4,800 employees globally. Learn more at www.microsemi.com.

50200440. 7.0 3/17

Contents

Figures

Tables

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 7.0

Updated the document for Libero v11.8 software release.

1.2 Revision 6.0

The following changes are done in revision 6.0 of this document.

- Libero SoC, FlashPro, and SoftConsole design requirements are updated in the Design Requirements, page 5.
- Throughout the guide, the names of SoftConsole projects used in the demo design and all the associated figures are updated.

1.3 Revision 5.0

Updated the document for Libero v11.7 software release (SAR 76559).

1.4 Revision 4.0

Updated the document for Libero v11.6 software release (SAR 72924).

1.5 Revision 3.0

Updated the document for Libero v11.5 software release (SAR 63972).

1.6 Revision 2.0

Updated the document for Libero v11.3 software release (SAR 56538).

1.7 Revision 1.0

Updated the document for Libero v11.2 software release (SAR 53221).

2 Running Modbus TCP Reference Design on SmartFusion2 Devices Using lwIP and FreeRTOS

2.1 Introduction

Microsemi offers a reference design for SmartFusion®2 SoC FPGA devices that demonstrate the tri-speed ethernet medium access controller (TSEMAC) features of the SmartFusion2 SoC FPGA and implements the Modbus protocol. The reference design runs on the *UG0557: SmartFusion2 SoC FPGA Advanced Development Kit User Guide*. This demo guide describes.

- Usage of SmartFusion2 TSEMAC connected to a serial gigabit media independent interface (SGMII) PHY.
- Integration of SmartFusion2 MAC driver with the lightweight IP (lwIP) transmission control protocol (TCP) or IP stack and the free real time operating system (RTOS).
- Application layer with industrial automation protocol, Modbus on TCP or IP.
- How to run the reference design

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

- 10 Mbps
- 100 Mbps
- 1000 Mbps

For more information on the TSEMAC interface for SmartFusion2 devices, see the *UG0331: SmartFusion2 Microcontroller Subsystem User Guide*.

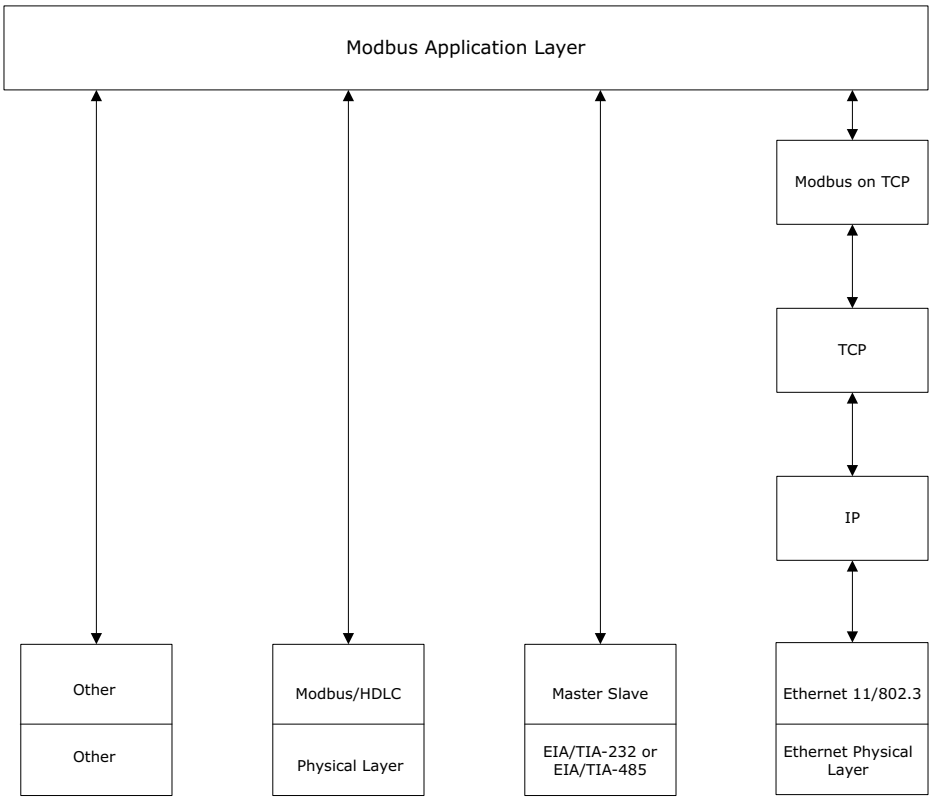
2.1.1 Using the Modbus Protocol

Modbus is an application layer messaging protocol present at the level seven of the open systems interconnection (OSI) model. It enables client or server communication between the devices connected in different types of buses or networks. It is a service protocol that offers many services specified by the function codes. The Modbus function codes are elements of Modbus request or reply protocol data units. The components of the Modbus protocol include:

- TCP or IP over Ethernet
- Asynchronous serial transmission over a variety of media
  - Wire:
    – EIA/TIA-232-E
    – EIA-422
    – EIA/TIA-485-A Fiber
  - Radio
- Modbus PLUS, a high-speed token passing network

The following figure describes the Modbus communication stacks for various communication networks.

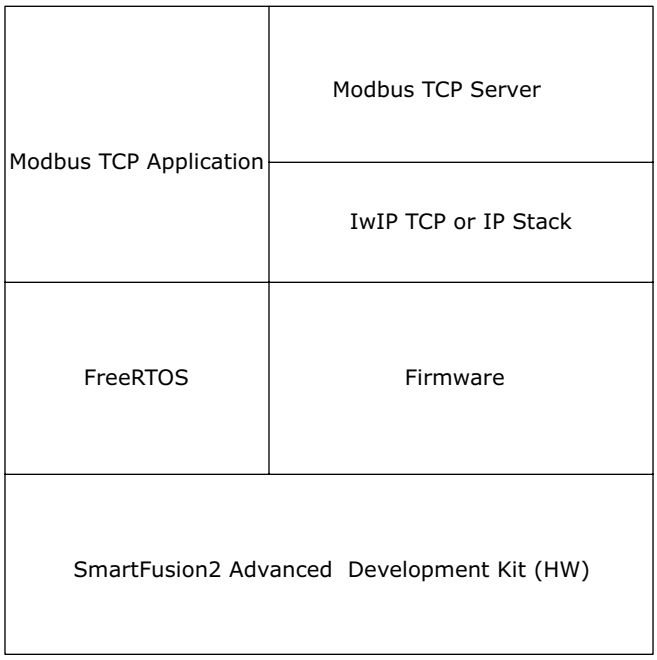
Figure 1 • **Modbus Communication Stack**



2.1.2 Using Modbus Protocol on SmartFusion2 Device

The Modbus TCP server runs on the SmartFusion2 Advanced Development Kit and responds to the Modbus TCP client running on the host PC. The following figure shows the block diagram of the Modbus TCP server and application on the SmartFusion2 device.

Figure 2 • **Block Diagram of Modbus TCP Server and Application on SmartFusion2**



2.2 Design Requirements

The following table lists the hardware and software design requirements.

Table 1 • **Reference Design Requirements and Details**

Design Requirements	Description
Hardware	
SmartFusion2 Advanced Development Kit – USB A to mini-B cable – 12 V adapter	Rev A or later
Ethernet cable	RJ45
Any one of the following serial terminal emulation programs: – HyperTerminal – TeraTerm – PuTTY	–
Host PC or Laptop	Windows 64-bit Operating System
Software	
Libero® System-on-Chip (SoC)	v11.8
SoftConsole	v4.0
FlashPro programming software	v11.8
USB to UART drivers	–
MSS Ethernet MAC drivers	v3.1.100
A serial terminal emulation program	HyperTerminal, TeraTerm, or PuTTY
Browser	Mozilla Firefox or Internet Explorer

2.3 Demo Design

The following sections describe the demo design of the Modbus TCP reference design on SmartFusion2 devices using lwIP and FreeRTOS.

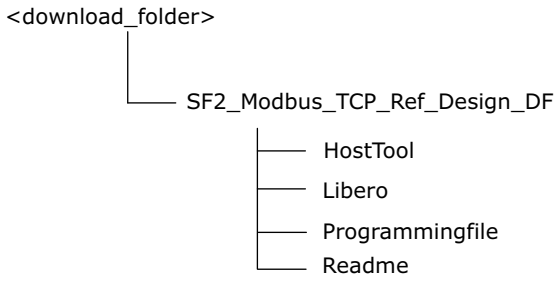
The demo design files are available for download at:
http://soc.microsemi.com/download/rsc/?f=m2s_dg0440_liberov11p8_df

The demo design files include:

- Libero
- Programming files
- HostTool
- Readme

The following figure shows the top-level structure of the design files. For more information, see the `Readme.txt` file.

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



2.3.1 Demo Design Features

The reference design includes:

- Complete Libero SoC Verilog project
- SoftConsole firmware project

The reference design can support the following Modbus function codes depending on the free Modbus communications stack settings:

- Read input registers (function code 0×04)
- Read holding registers (function code 0×03)
- Write single registers (function code 0×06)
- Write multiple registers (function code 0×10)
- Read or Write multiple registers (function code 0×17)
- Read coils (function code 0×01)
- Write single coil (function code 0×05)
- Write multiple coils (function code 0×0F)
- Read discrete inputs (function code (0×02)

The reference design supports the following Modbus function codes for all free Modbus communications stack settings:

- Read input registers (function code 0×04)
- Read discrete inputs (function code (0×02)
- Write multiple coils (function code 0×0F)
- Read holding registers (function code 0×03)

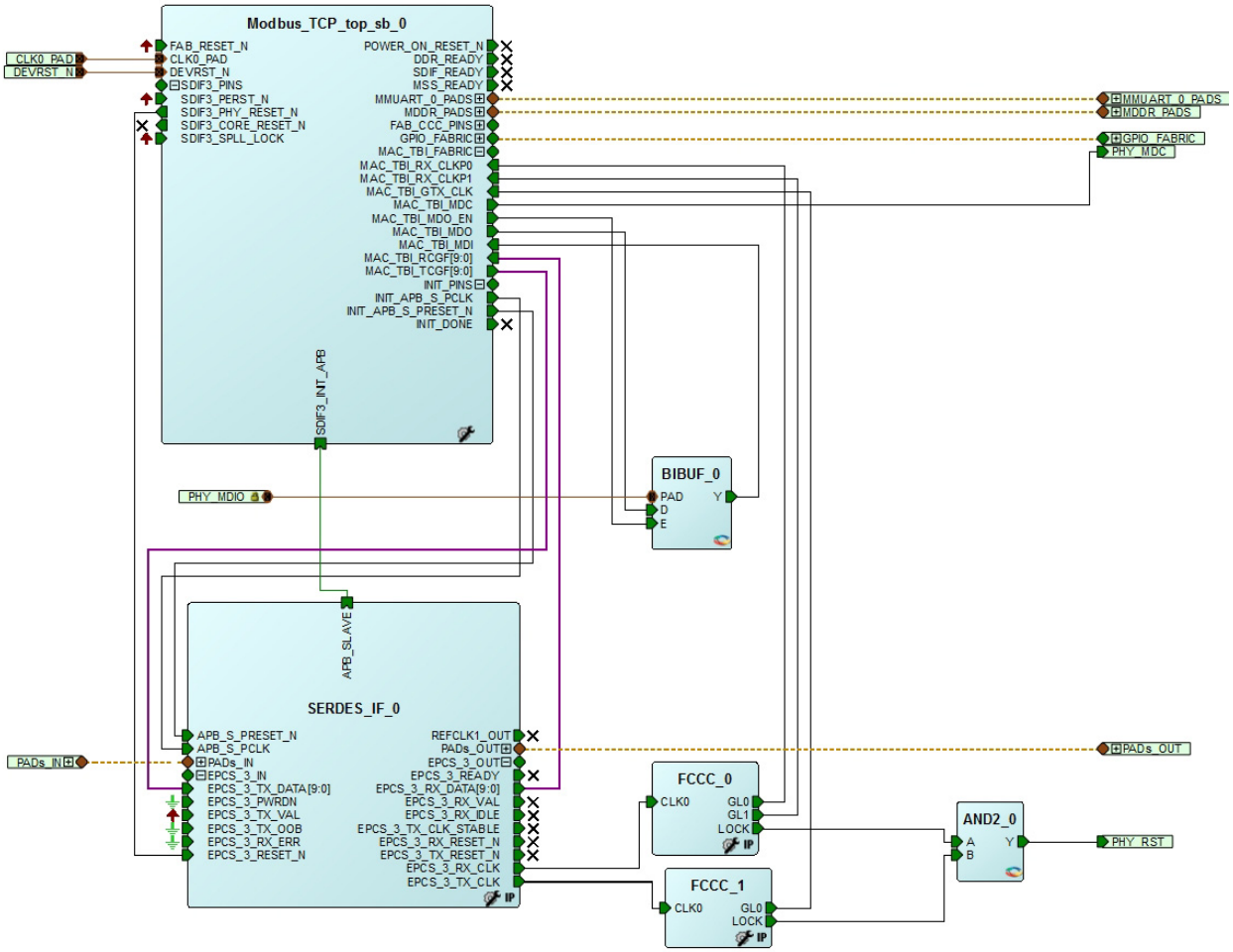
2.3.2 Demo Design Description

The design is implemented using a 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 hardware design implementation on which the reference design slave firmware runs.

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

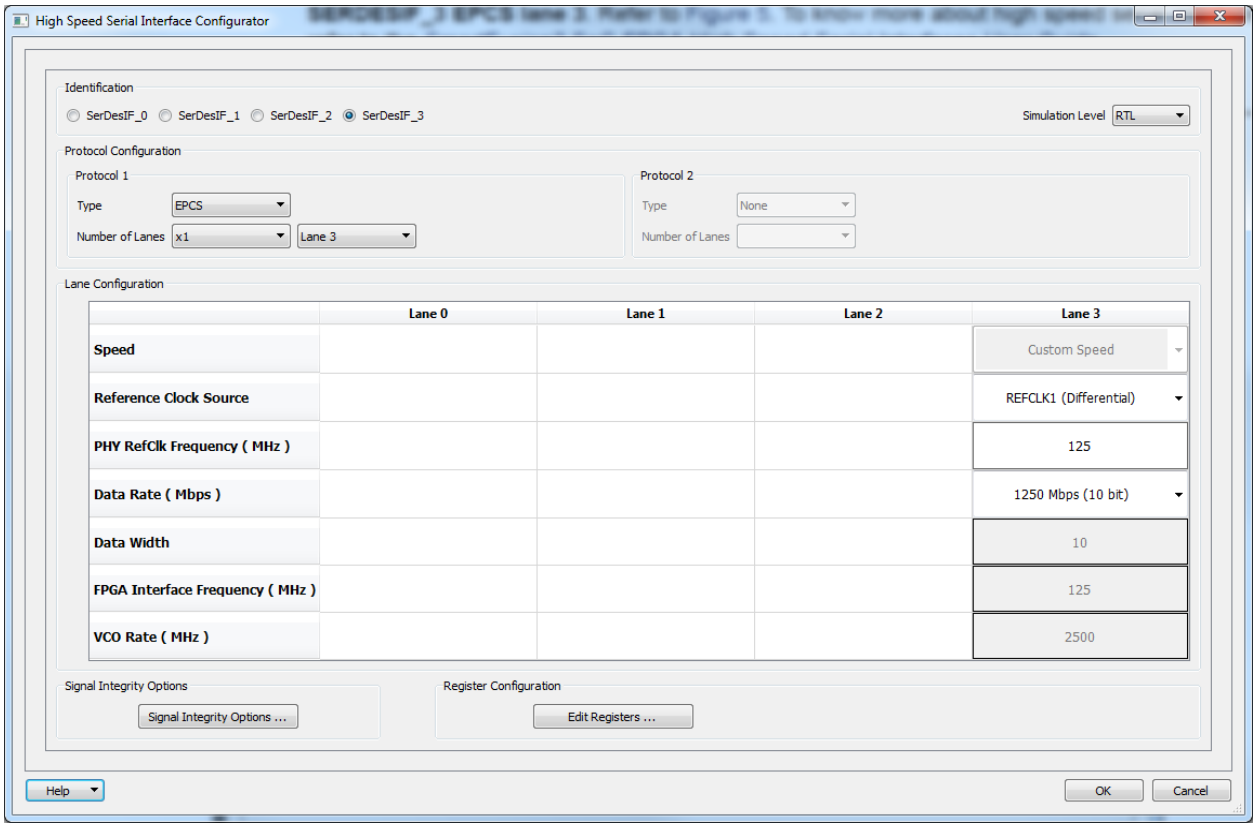


The Libero SoC hardware project uses the following SmartFusion2 MSS resources and IPs:

- **TSEMAC TBI** interface
- **MMUART_0** for RS-232 communications on the SmartFusion2 Advanced Development Kit
- **Dedicated input pad 0 as the clock source**
- General purpose input and output (GPIO) that interfaces the following:
  - Light emitting diodes (LEDs): 4 numbers
  - Push-buttons: 4 numbers
  - Dual in-line package (DIP) switches: 4 numbers
- The following board resources are associated with the Modbus commands:
  - LEDs (coils)
  - DIP switches (discrete inputs)
  - Push-buttons (discrete inputs)
  - Real time clock (RTC) (input registers)
- High-speed serial interface (SERDESIF) **SERDES_IF IP**, configured for **SERDESIF_3 EPCS lane 3**, see the following figure. To know more about high-speed serial interfaces, see the *UG0447- SmartFusion2 and IGLOO2 FPGA High Speed Serial Interfaces User Guide*.

The following figure shows the **High Speed Serial Interface Configurator** window.

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



2.3.2.1.1 Package Pin Assignments

Package pin assignments for LED, DIP switches, push-button switches, and PHY interface signals are shown in the following table through Table 5, page 9.

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

Output	Package Pin
LED_1	D26
LED_2	F26
LED_3	A27
LED_4	C26

Table 3 • **DIP Switches to Package Pins Assignments**

Output	Package Pin
DIP1	F25
DIP2	G25
DIP3	J23
DIP4	J22

Table 4 • **Push Button Switches to Package Pins Assignments**

Output	Package Pin
SWITCH1	J25
SWITCH2	H25
SWITCH3	J24
SWITCH4	H23

Table 5 • **PHY Interface Signals to Package Pins Assignments**

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

2.3.2.2 SoftConsole Firmware Project

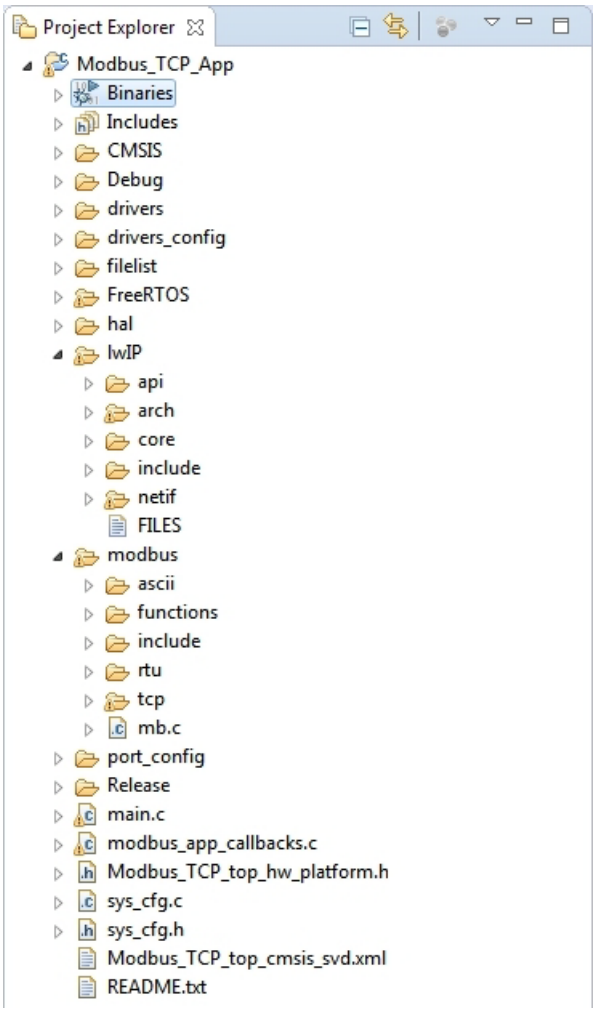
Invoke the SoftConsole project using standalone SoftConsole IDE.

The following versions of the stack are used for the reference design:

- **lwIP TCP or IP** stack version 1.3.2
- **Modbus TCP** server version 1.5 (*www.freemodbus.org*) with enhancements for the complete function code support as Modbus TCP server
- **FreeRTOS** (*www.freertos.org*)

The following figure shows SoftConsole software stacks directory structure of the design.

Figure 6 • **SoftConsole Project Explorer Window**



The SoftConsole workspace consists of the project, Modbus_TCP_App that has the Modbus TCP application (which uses lwIP and FreeRTOS) and all the firmware and hardware abstraction layers that correspond to the hardware design.

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_CMSIS_0	SmartFusion2_CMSIS	2.3.105	Modbus_TCP_top_sb_MSS
2	☑		SmartFusion2_MSS_Ethernet_MAC_Driver_0	SmartFusion2_MSS_Ethernet_MAC_Driver	3.1.100	Modbus_TCP_top_sb_MSS:MAC
3	☑		SmartFusion2_MSS_GPIO_Driver_0	SmartFusion2_MSS_GPIO_Driver	2.1.102	Modbus_TCP_top_sb_MSS:GPIO
4	☐		SmartFusion2_MSS_HPDMA_Driver_0	SmartFusion2_MSS_HPDMA_Driver	2.2.100	Modbus_TCP_top_sb_MSS
5	☑		SmartFusion2_MSS_MMUART_Driver_0	SmartFusion2_MSS_MMUART_Driver	2.1.100	Modbus_TCP_top_sb_MSS:MMUART_0
6	☐		SmartFusion2_MSS_NVM_Driver_0	SmartFusion2_MSS_NVM_Driver	2.4.100	Modbus_TCP_top_sb_MSS
7	☑		SmartFusion2_MSS_RTC_Driver_0	SmartFusion2_MSS_RTC_Driver	2.2.100	Modbus_TCP_top_sb_MSS:RTC
8	☐		SmartFusion2_MSS_System_Services_Driver_0	SmartFusion2_MSS_System_Services_Driver	2.7.100	Modbus_TCP_top_sb_MSS
9	☑		SmartFusion2_MSS_Timer_Driver_0	SmartFusion2_MSS_Timer_Driver	2.2.100	Modbus_TCP_top_sb_MSS

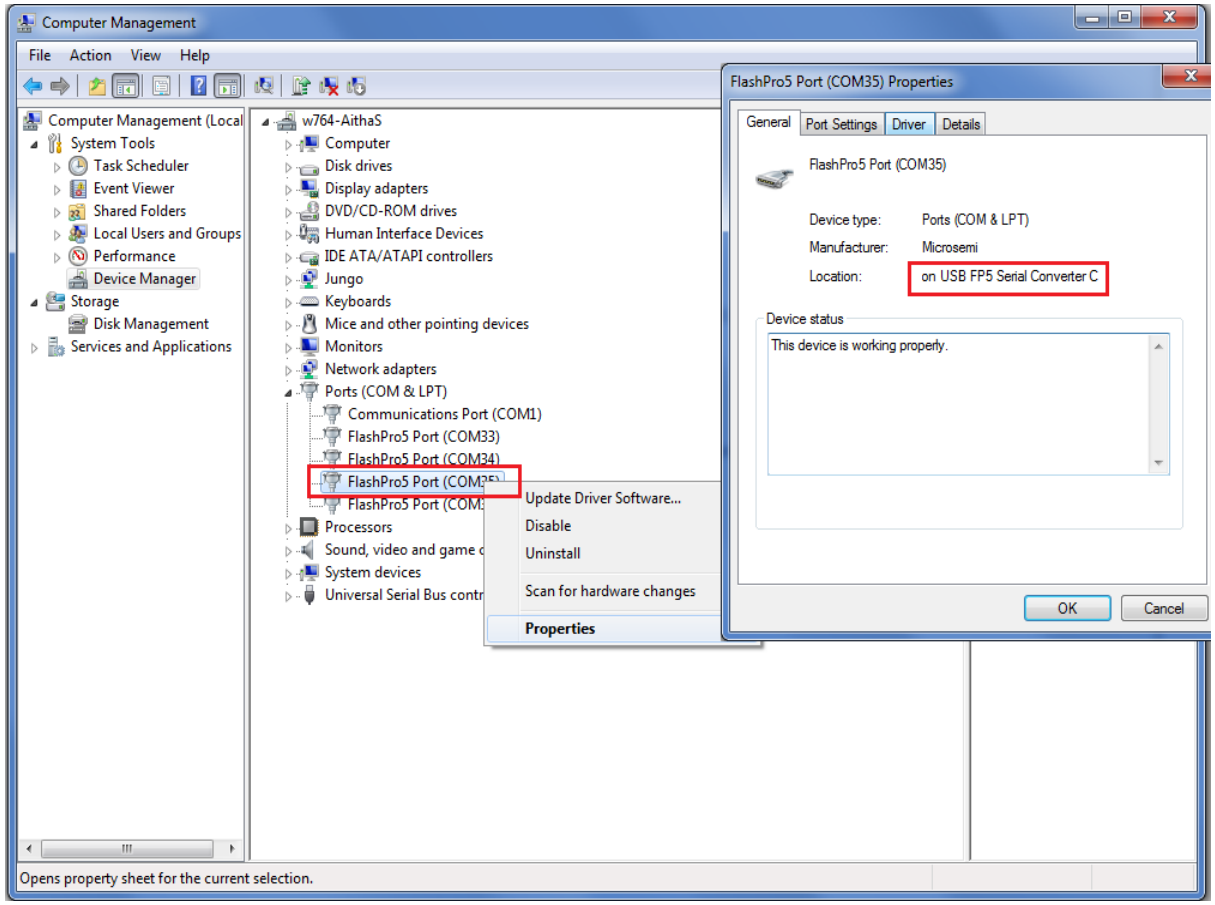
2.4 Setting Up the Demo Design

The following steps describe how to setup 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.
2. From the detected four communication (COM) ports, right-click any one of the COM ports and select **Properties**. The selected COM port properties window is displayed, as shown in the following figure.
3. Ensure to have the **Location** as **on USB FP5 Serial Converter C** in the **Properties** window as shown in the following figure.

Note: 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**.

Figure 8 • **Device Manager Window**



4. Install the USB driver if the USB drivers are not detected automatically.
5. Install the FTDI D2XX driver for serial terminal communication through the FTDI mini USB cable. Download the drivers and installation guide from: *www.microsemi.com/soc/documents/CDM_2.08.24_WHQL_Certified.zip*
6. Connect the jumpers on the SmartFusion2 Advanced Development Kit board as shown in the following table. For information on jumper locations, seethe Appendix: Jumper Locations, page 19.

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

Table 6 • **SmartFusion2 Advanced Development Kit Jumper Settings**

Jumper	Pin From	Pin To	Comments
J116, J353, J354,J54	1	2	These are the default jumper settings of the Advanced Development Kit board. Ensure that the jumpers are set accordingly.
J123	2	3	
J124, J121, J32	1	2	JTAG programming via FTDI

7. Connect the power supply to the **J42** connector in the SmartFusion2 Advanced Development Kit board.
8. 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 setup connections are given in the Appendix: Board Setup for Running the Modbus TCP Reference Design, page 18.

2.5 Running the Demo Design

The following steps describe how to run the demo design:

1. Download the design file from: *http://soc.microsemi.com/download/rsc/?f=m2s_dg0440_liberov11p8_df*
2. Switch ON the power supply switch, **SW7**.
3. Start any serial terminal emulation program such as:
    - HyperTerminal
    - PuTTY
    - TeraTerm

Note: In this demo HyperTerminal is used.

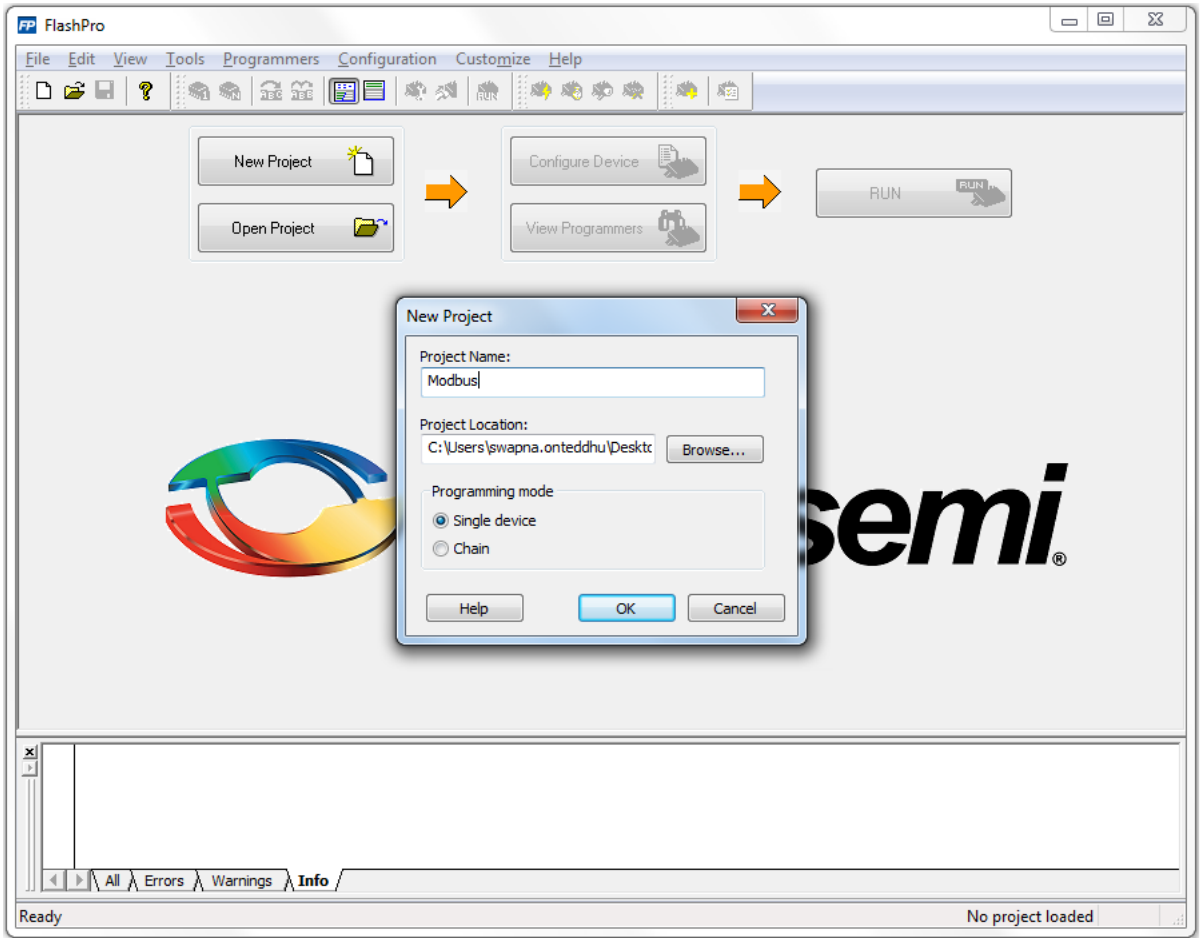
The configuration for the program is:

- Baud Rate: 115200
- 8 Data bits
- 1 Stop bit
- No parity
- No flow control

For information on configuring the serial terminal emulation programs, see the *Configuring Serial Terminal Emulation Programs*.

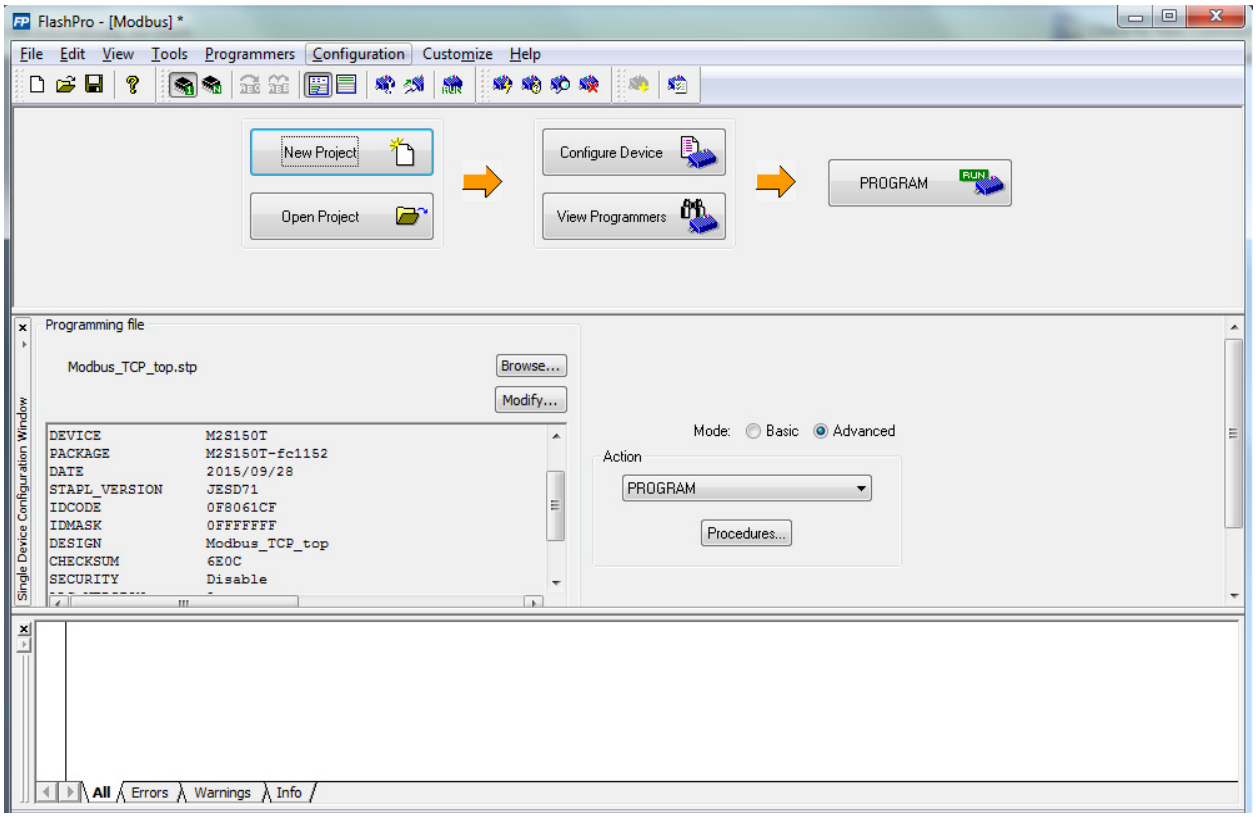
4. Launch the FlashPro software.
5. Click **New Project**.
6. In the **New Project** window, enter the Project Name, as shown in the following figure.

Figure 9 • **FlashPro New Project**



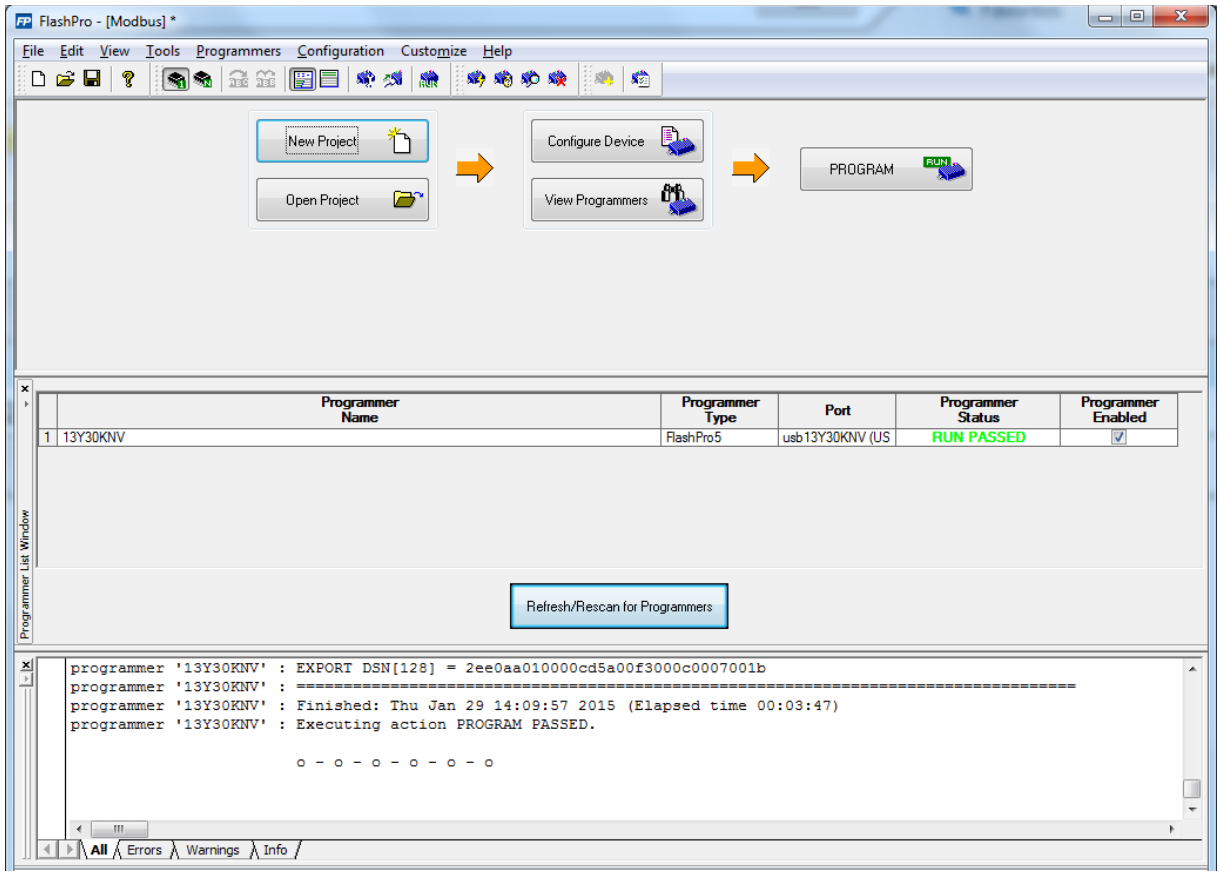
7. Click **Browse** and navigate to the location where you want to save the project.
8. Select **Single device** as the **Programming mode**.
9. Click **OK** to save the project.
10. Click **Configure Device**.
11. Click **Browse** and navigate to the location where the `Modbus_TCP_top.stp` file is located and select the file. The default location is: (*\SF2_Modbus_TCP_Ref_Design_DF\Programmingfile\Modbus_TCP_top.stp*). The required programming file is selected and is ready to be programmed in the device as shown in the following figure.

Figure 10 • **FlashPro Project Configured**



12. Click **PROGRAM** to start programming the device. Wait until a message is displayed indicating that the program passed. This demo requires the SmartFusion2 device to be preprogrammed with the application code to activate the Modbus application. The SmartFusion2 device is preprogrammed with the `Modbus_TCP_top.stp` using FlashPro software.

Figure 11 • **FlashPro Program Passed**

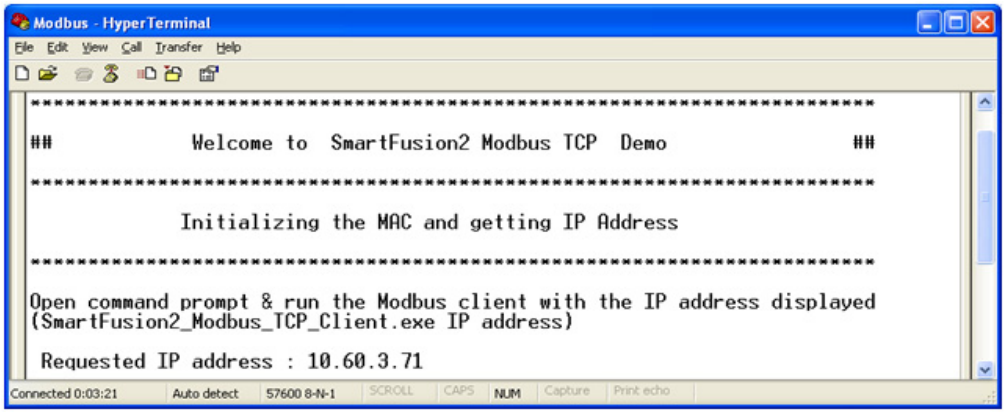


Note: To run the design in static IP mode, follow the steps mentioned in Appendix: Running the Design in Static IP Mode, page 20.

13. Power cycle the SmartFusion2 Advanced Development board.

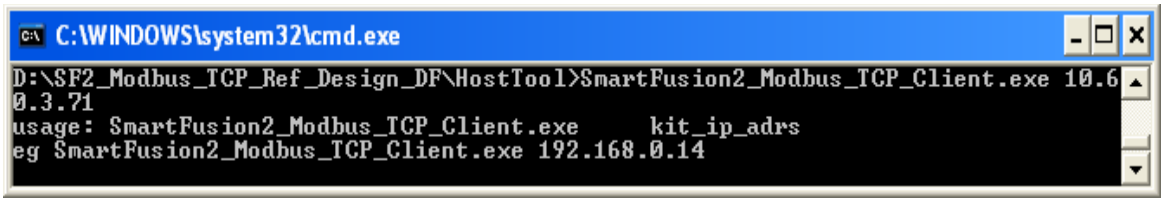
A welcome message with the IP address is displayed in the HyperTerminal window, as shown in the following figure.

Figure 12 • **HyperTerminal with IP Address**



Open a new command prompt on the host PC, go to the folder (*\SF2_Modbus_TCP_Ref_Design_DF\HostTool*) where `SmartFusion2_Modbus_TCP_Client.exe` file is present, enter the command: `SmartFusion2_Modbus_TCP_Client.exe <IP address>` as shown in the following figure.

Figure 13 • **Invoking the Modbus Client**



The following figure shows the Modbus TCP functions that are running. The functions are:

- Read discrete inputs (function code 02)
- Read holding registers (function code 03)
- Read input registers (function code 04)
- Write multiple coils (function code 15)

Figure 14 • **Modbus Functional Codes Demonstration**

See the Running Modbus Functions, page 17 for more information on the Modbus functions that are demonstrated in the reference design.

14. After running the demo, close HyperTerminal.

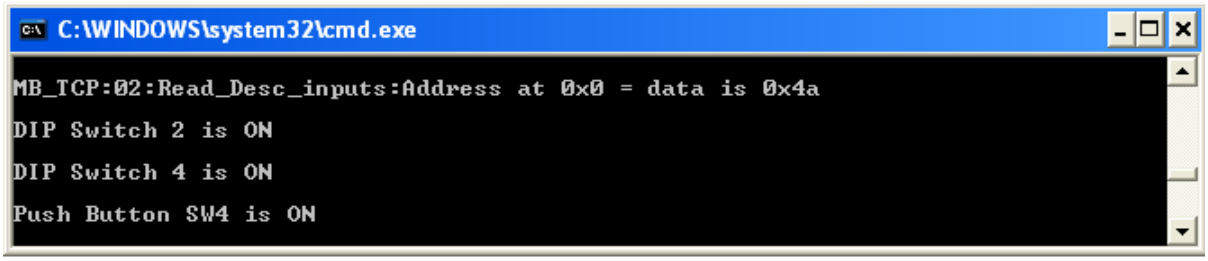
2.5.1 Running Modbus Functions

This section describes the Modbus functions that are demonstrated in the reference design.

2.5.1.1 Read Discrete Inputs (function code 02)

GPIOs are connected to 4 DIP switches and 4 push-button switches. Switch ON and switch OFF the DIP switches and push-button switches on the SmartFusion2 Advanced Development Kit. Read discrete inputs functional code displays the statuses of switches as shown in the following figure.

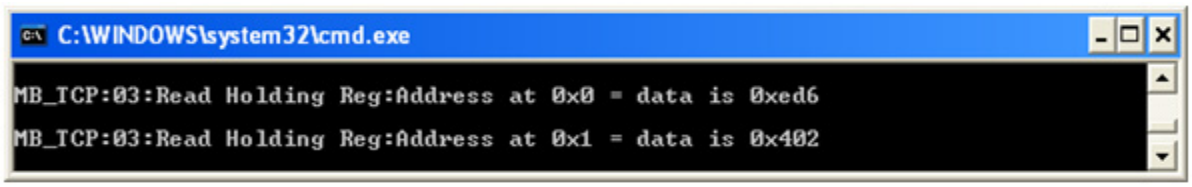
Figure 15 • **Read Discrete Inputs**



2.5.1.2 Read Holding Registers (function code 03)

The following figure shows the global buffer data defined in the firmware.

Figure 16 • **Read Holding Registers**



2.5.1.3 Read Input Registers (function code 04)

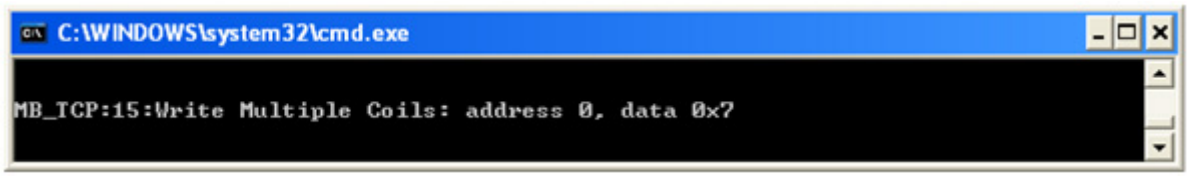
The following figure shows the number of seconds that the real-time counter (RTC) has counted.

Figure 17 • **Read Input Registers**

2.5.1.4 Write Multiple Coils (function code 0×0F)

The following figure shows the Write Multiple Coils register data for toggling the LEDs connected to GPIOs.

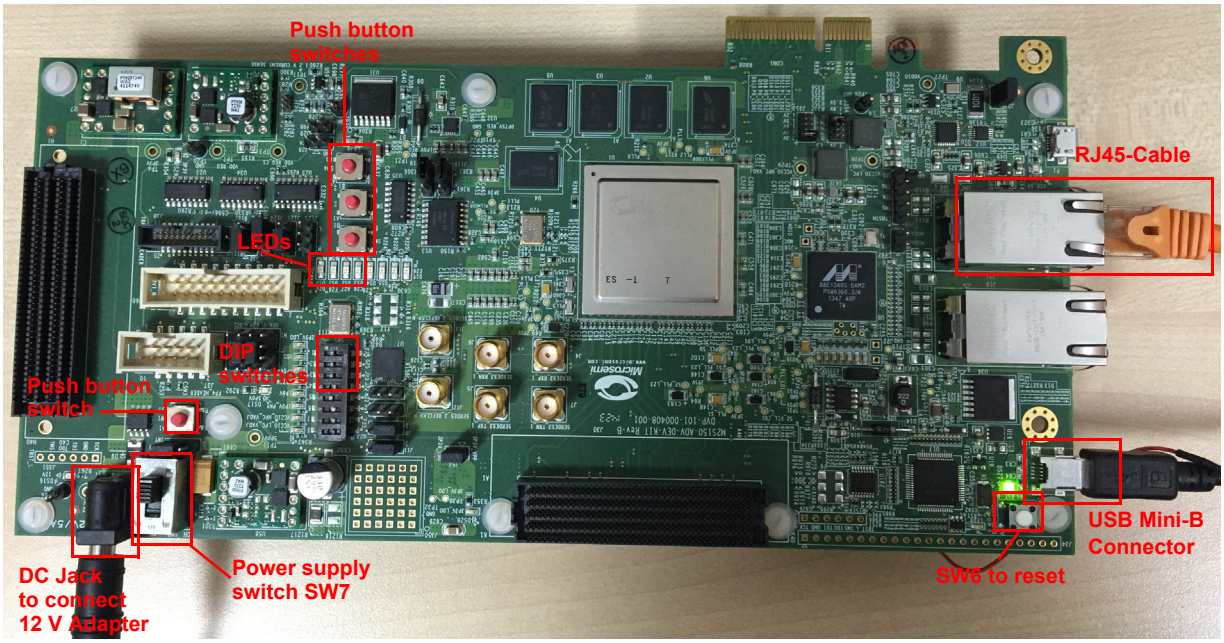
Figure 18 • **Write Multiple Coils**



3 Appendix: Board Setup for Running the Modbus TCP Reference Design

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

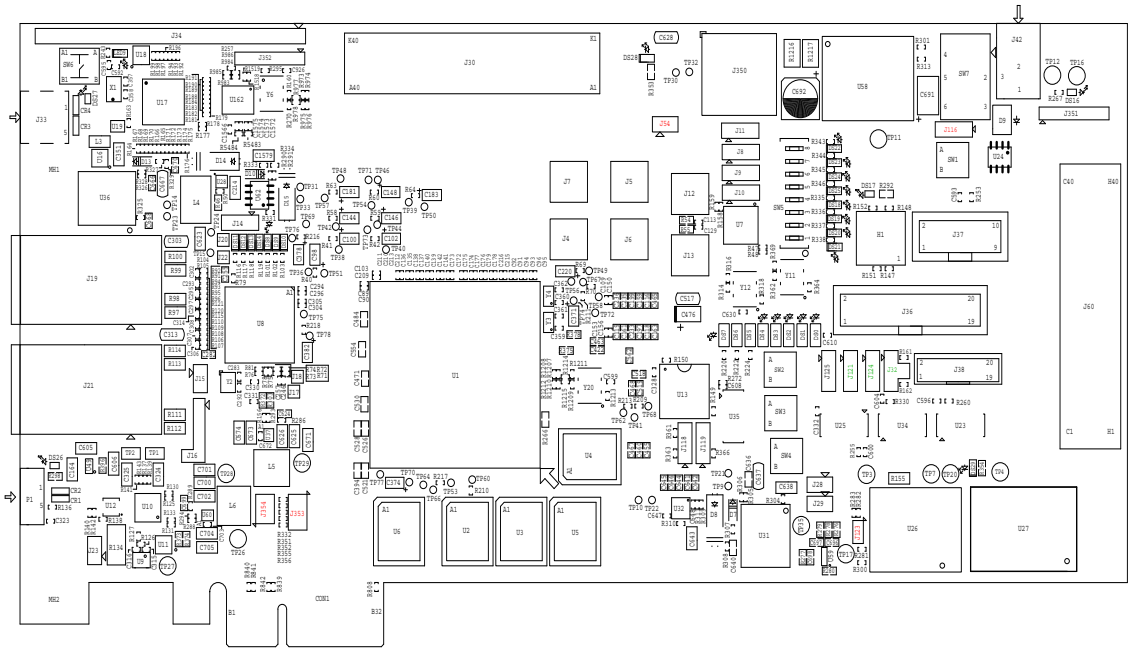
Figure 19 • **SmartFusion2 Advanced Development Kit Board Setup**



4 Appendix: Jumper Locations

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

Figure 20 • **SmartFusion2 Advanced Development Kit Silkscreen Top View**



Note: Jumpers highlighted in red are set by default. Jumpers highlighted in green must be set manually.

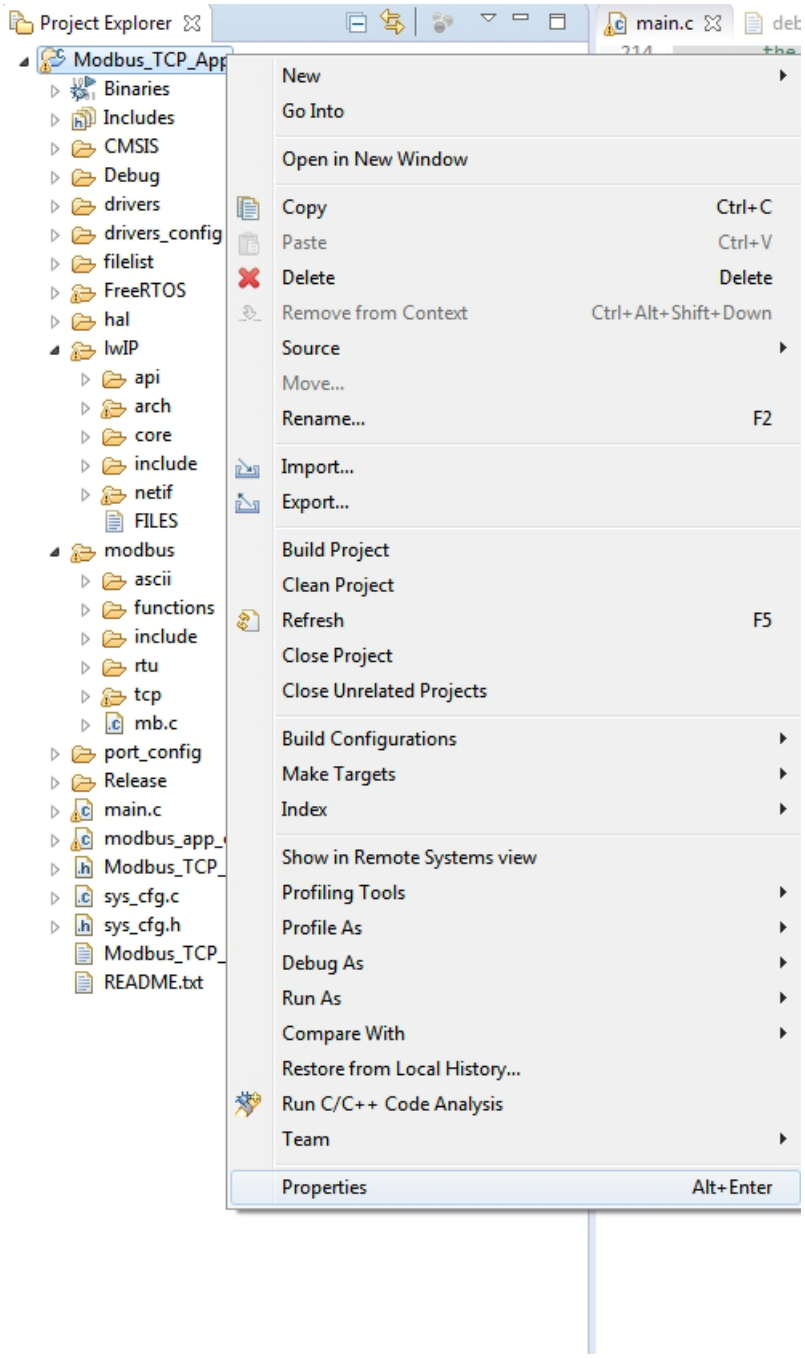
Note: The location of the jumpers in the preceding figure are searchable.

5 Appendix: Running the Design in Static IP Mode

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

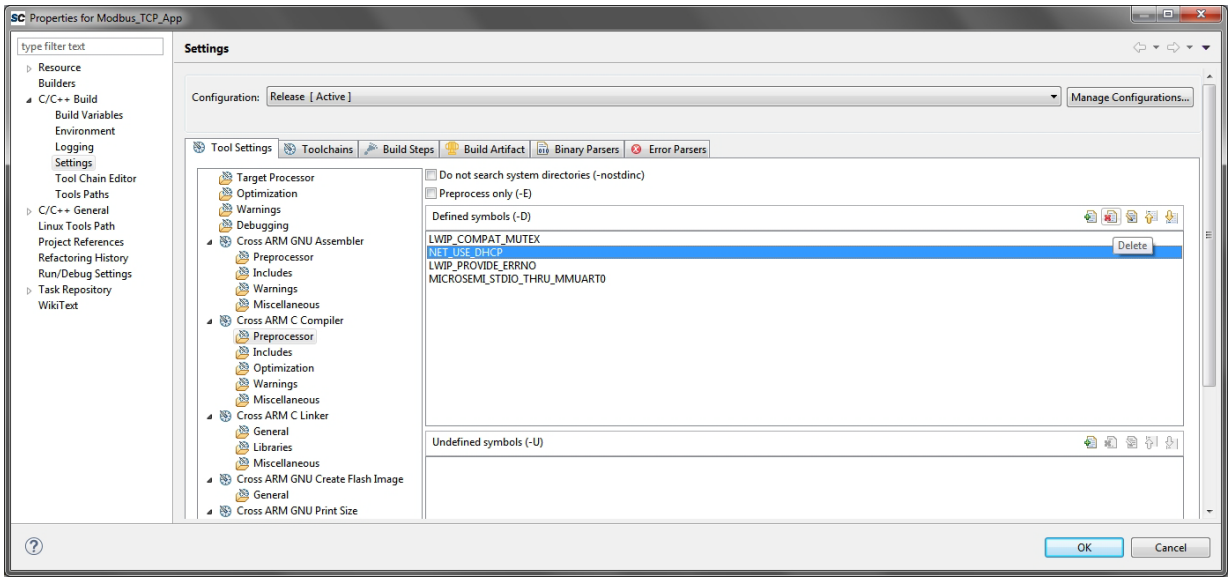
1. Right-click the **Project Explorer** window of SoftConsole project and go to **Properties** as shown in the following figure.

Figure 21 • **Project Explorer Window of SoftConsole Project**



2. Remove the symbol **NET_USE_DHCP** in **Tool Settings** of the **Properties for Modbus_TCP_App** window. The following figure shows the **Properties for Modbus_TCP_App** window.

Figure 22 • **Project Explorer Properties Window**



3. 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. See the following figure and Figure 24, page 22.

Figure 23 • **Host PC TCP/IP Settings**

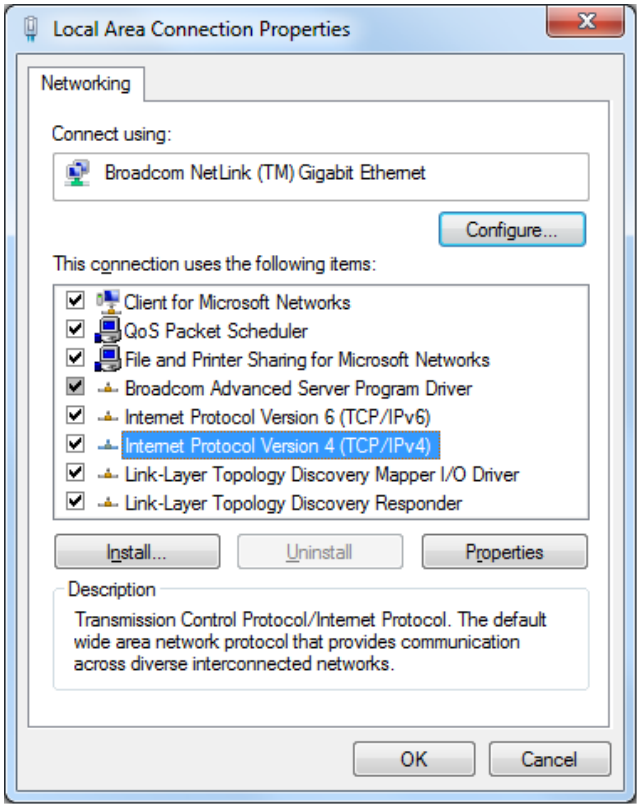
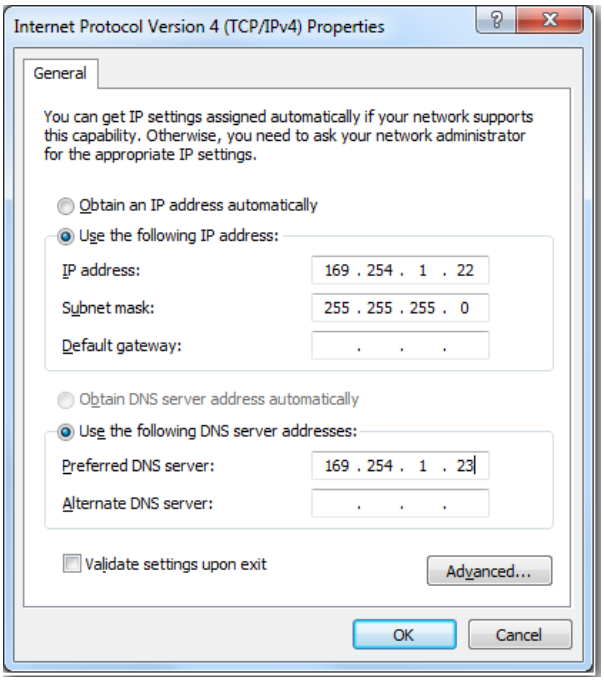


Figure 24 • **Static IP Address Settings**



Note: When these settings are configured, compile the design, load the design into Flash memory, and run the design using SoftConsole.