

---

***Running Modbus TCP Reference Design  
on SmartFusion2 Devices using lwIP and  
FreeRTOS - Libero SoC v11.6***

***DG0440 Demo Guide***

Superseded

---

October 2015

## Revision History

Date	Revision	Change
30 October 2015	4	Fifth release
13 March 2015	3	Fourth release
01 May 2014	2	Third release
09 December 2013	1	Second release
04 October 2013	0	First release

## Confidentiality Status

This is a non-confidential document.

Superseded

---

# Table of Contents

---

Preface .....	4
About this document .....	4
Intended Audience .....	4
References .....	4
Modbus Protocol Quick References .....	4
Microsemi Publications .....	4
Running Modbus TCP Reference Design on SmartFusion2 Using lwIP and FreeRTOS .....	5
Introduction .....	5
Using the Modbus Protocol .....	5
Using Modbus Protocol on SmartFusion2 Device .....	7
Design Requirements .....	8
Demo Design .....	8
Introduction .....	8
Demo Design Features .....	9
Demo Design Description .....	9
Setting uUp the Demo Design .....	14
Board Setup Snapshot .....	15
Running the Demo Design .....	15
Running Modbus Functions .....	21
Appendix 1: Board Setup for Running the Modbus TCP Reference Design .....	22
Appendix 2: Jumper Locations .....	23
Appendix 3: Running the Design in Static IP Mode .....	24
List of Changes .....	28
Product Support .....	29
Customer Service .....	29
Customer Technical Support Center .....	29
Technical Support .....	29
Website .....	29
Contacting the Customer Technical Support Center .....	29
Email .....	29
My Cases .....	30
Outside the U.S. ....	30
ITAR Technical Support .....	30

---

## Preface

---

### About this document

This demo is for SmartFusion<sup>®</sup>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:

- Modbus Organization: [www.modbus.org](http://www.modbus.org)
  - FAQ: [www.modbus.org/faq.php](http://www.modbus.org/faq.php)
  - Technical resources including specifications and links to free and commercial Modbus tools and resources: [www.modbus.org/tech.php](http://www.modbus.org/tech.php)
- Wikipedia page on Modbus: <http://en.wikipedia.org/wiki/Modbus>
- FreeModbus home page: [www.freemodbus.org](http://www.freemodbus.org)
  - API Documentation: [www.freemodbus.org/api/index.html](http://www.freemodbus.org/api/index.html)
  - Examples using Modpoll: [www.freemodbus.org/index.php?idx=1](http://www.freemodbus.org/index.php?idx=1)
- lwIP TCP/IP stack: [www.sics.se/~adam/lwip/](http://www.sics.se/~adam/lwip/)
- FreeRTOS<sup>™</sup> stack: [www.freertos.org](http://www.freertos.org)

### Modbus Protocol Quick References

- The Modbus TCP implementation guidelines can be found in the *Modbus Messaging on TCP/IP Implementation Guide v1.0b*.
- The Modbus protocol source code used for the design example in this document is from [www.freemodbus.org](http://www.freemodbus.org) with updates for the complete set of features of the Modbus layer.

### Microsemi Publications

- *UG0331: SmartFusion2 Microcontroller Subsystem User Guide*
- *UG0447: IGLOO2 FPGA and SmartFusion2 SoC FPGA 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 Modbus TCP Reference Design on SmartFusion2 Using lwIP and FreeRTOS

---

## Introduction

Microsemi® offers a reference design for SmartFusion2 SoC FPGA devices that demonstrates 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 [SmartFusion2 Advanced Development Kit](#). 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 lwIP TCP or IP stack and the FreeRTOS Operating System.
- 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

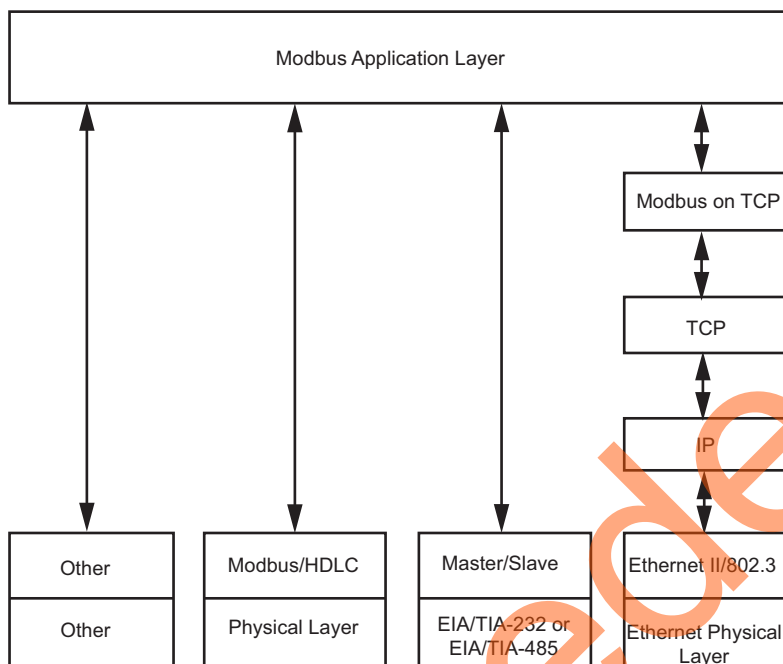
Refer to the [UG0331: SmartFusion2 Microcontroller Subsystem User Guide](#) for more information on the TSEMAC interface for SmartFusion2 devices.

## 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

Figure 1 describes the Modbus communication stacks for various communication networks.

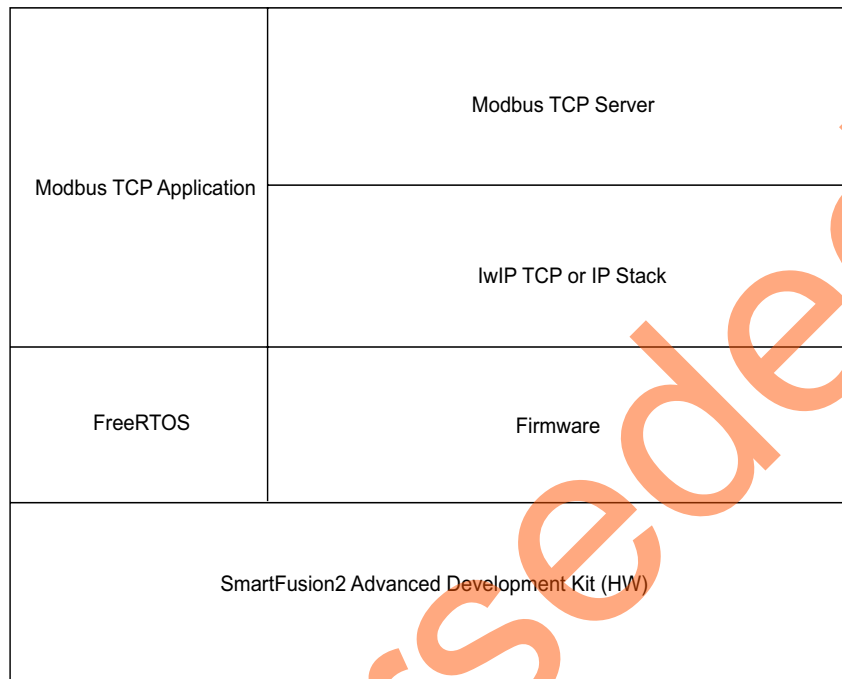


**Figure 1 • Modbus Communication Stack**

For more information on Modbus protocol, refer to the "Modbus Protocol Quick References" section on page 4.

## 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. [Figure 2](#) shows the block diagram of Modbus TCP Server and application on the SmartFusion2 device.



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

## Design Requirements

Table 1 lists the hardware and software design requirements.

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

Reference Design Requirements and Details	Description
<b>Hardware Requirements</b>	
SmartFusion2 Advanced Development Kit <ul style="list-style-type: none"><li>FlashPro5</li><li>USB A to mini-B cable</li><li>12 V adapter</li></ul>	Rev A or later
RJ45 cable	–
Any one of the following serial terminal emulation programs: <ul style="list-style-type: none"><li>HyperTerminal</li><li>TeraTerm</li><li>PuTTY</li></ul>	–
Host PC or Laptop	Any 64-bit Windows Operating System
<b>Software Requirements</b>	
Libero® System-on-Chip (SoC)	v11.6
SoftConsole	v3.4 SP1
FlashPro programming software	v11.6
USB to UART drivers	–

## Demo Design

### Introduction

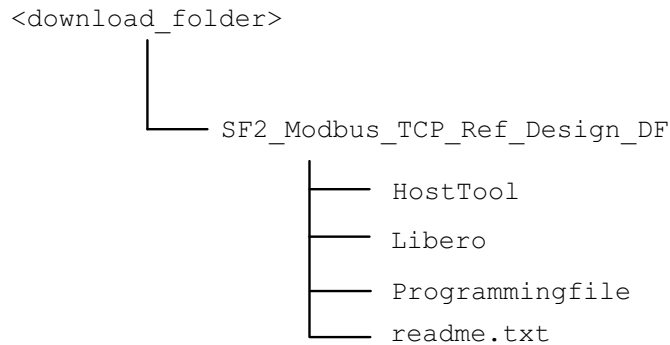
The reference design files are available for download from the Microsemi website:  
[http://soc.microsemi.com/download/rsc/?f=m2s\\_dg0440\\_liberov11p6\\_df](http://soc.microsemi.com/download/rsc/?f=m2s_dg0440_liberov11p6_df)

The demo design files include:

- Libero SoC project
- Programming files
- HostTool
- 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 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 0x04)
  - Read holding registers (function code 0x03)
  - Write single registers (function code 0x06)
  - Write multiple registers (function code 0x10)
  - Read or Write multiple registers (function code 0x17)
  - Read coils (function code 0x01)
  - Write single coil (function code 0x05)
  - Write multiple coils (function code 0x0F)
  - Read discrete inputs (function code 0x02)
- The reference design supports the following Modbus function codes for all Free Modbus communications stack settings:
  - Read input registers (function code 0x04)
  - Read discrete inputs (function code 0x02)
  - Write multiple coils (function code 0x0F)
  - Read holding registers (function code 0x03)

## Demo Design Description

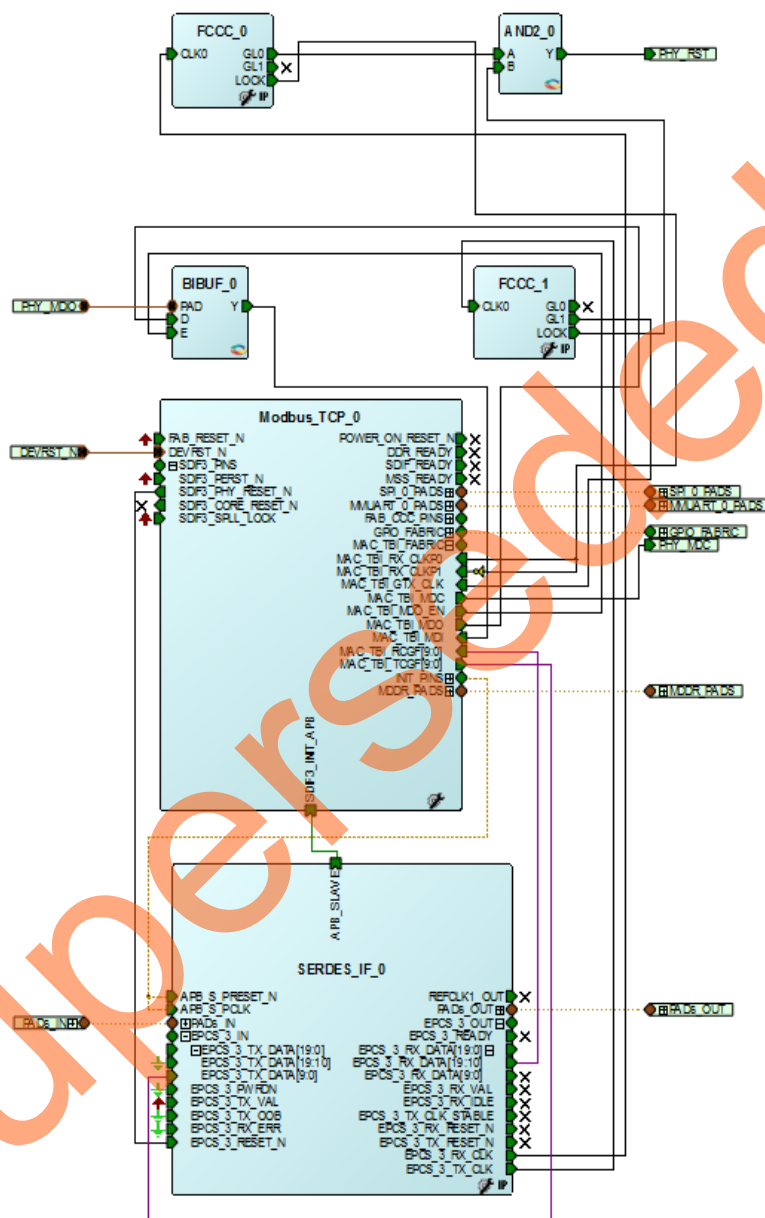
The design is implemented using a SGMII PHY interface by configuring the TSEMAC for the ten-bit interface (TBI) operation. For more information on the TSEMAC TBI interface, refer to the [UG0331: SmartFusion2 Microcontroller Subsystem User Guide](#).

The design comprises:

- [Libero SoC Hardware Project](#)
- [SoftConsole Firmware Project](#)

## Libero SoC Hardware Project

A Libero SoC v11.6 project uses the SmartFusion2 MSS 1.1.400. Figure 4 shows the hardware design implementation on which the reference design slave firmware runs.



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

Libero 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
- General purpose input and output (GPIO) that interfaces the following:
  - Light emitting diodes (LEDs): 4 numbers
  - Push-buttons: 4 numbers
  - DIP switches: 4 numbers

- The following board resources are associated with the Modbus commands:
  - LEDs (coils)
  - DIP switches (discrete inputs)
  - Push-buttons (discrete inputs)
  - RTC (input registers)
- High-speed serial interface (SERDESIF) **SERDES\_IF IP**, configured for **SERDESIF\_3 EPCS lane 3**, refer to [Figure 5](#). To know more about high-speed serial interfaces, refer to the [UG0447: IGLOO2 FPGA and SmartFusion2 SoC FPGA High Speed Serial Interfaces User Guide](#).

Figure 5 shows the High Speed Serial Interface Configurator window.

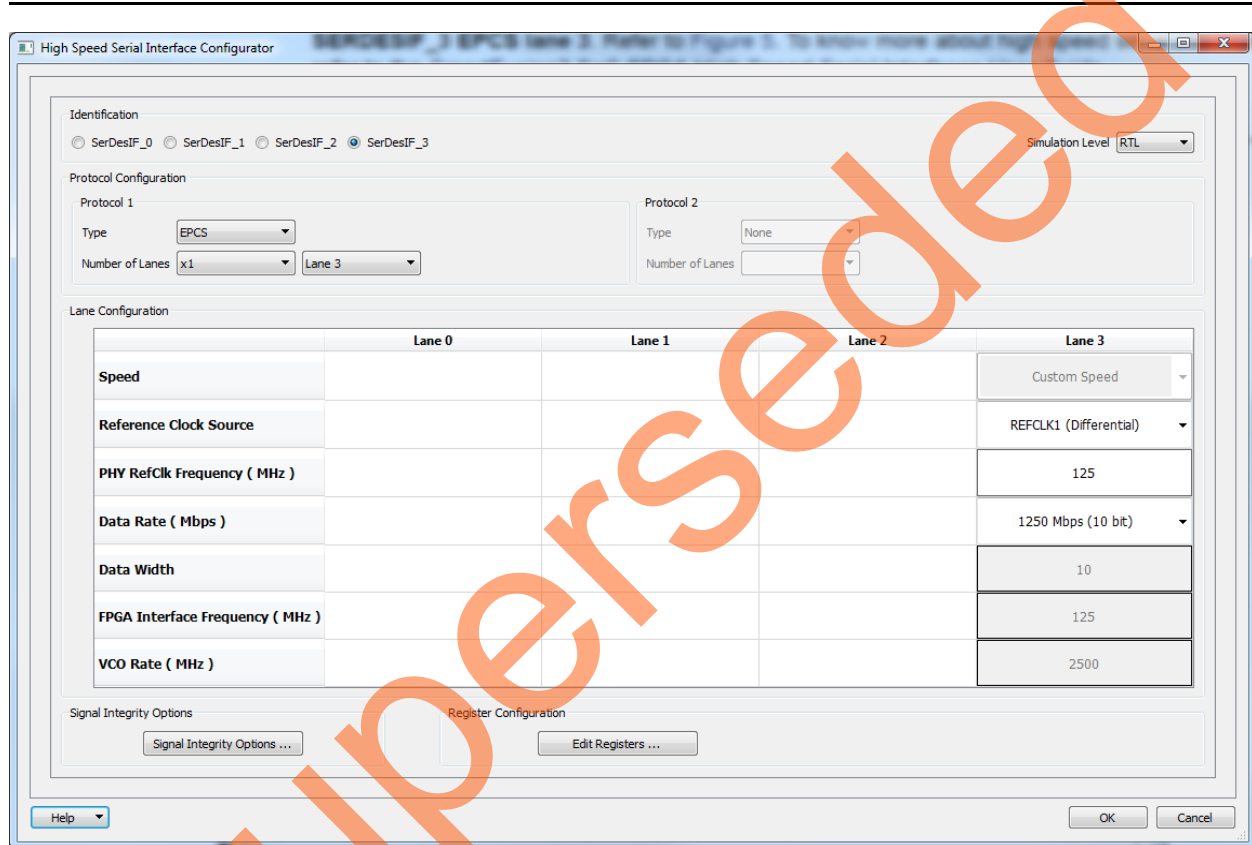


Figure 5 • High Speed Serial Interface Configurator Window

### Package Pin Assignments

Package pin assignments for LED, DIP switches, push-button switches, and PHY interface signals are shown in Table 2 through Table 5.

**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

### 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 ([www.sics.se/~adam/lwip/](http://www.sics.se/~adam/lwip/))
- Modbus TCP server version 1.5 ([www.freemodbus.org](http://www.freemodbus.org)) with enhancements for the complete function code support as Modbus TCP server
- FreeRTOS ([www.freertos.org](http://www.freertos.org))

Figure 2 on page 7 shows the block diagram of the Modbus TCP Server and application on SmartFusion2 used in this design.

Figure 6 shows SoftConsole software stacks directory structure of the design.

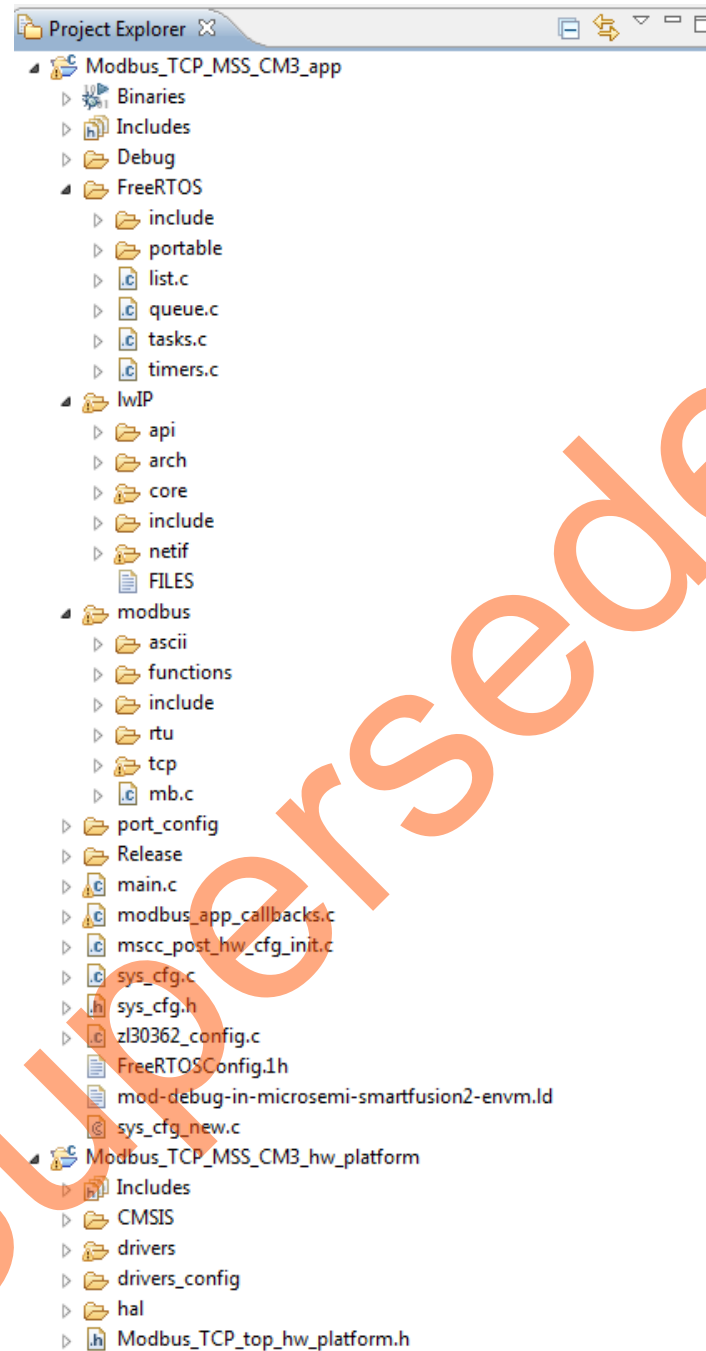
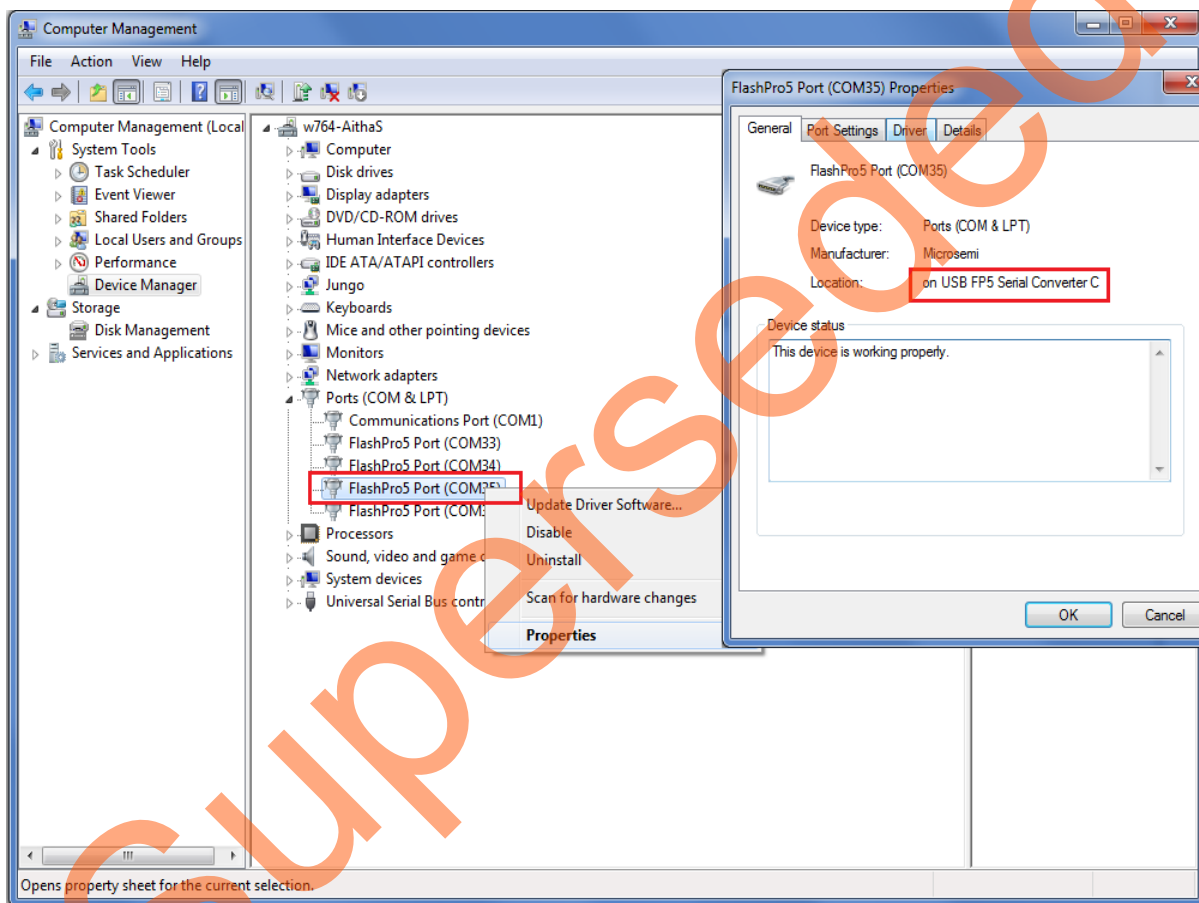


Figure 6 • SoftConsole Project Explorer Window

## 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, right-click any one of the COM ports and select **Properties**. The selected COM port properties window is displayed, as shown in [Figure 7](#).
3. Ensure to have the **Location** as **on USB FP5 Serial Converter C** in the **Properties** window as shown in [Figure 7](#).
4. 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 7 • Device Manager Window**

5. Install the USB driver if the USB drivers are not detected automatically.
6. 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](http://www.microsemi.com/soc/documents/CDM_2.08.24_WHQL_Certified.zip)

7. Connect the jumpers on the SmartFusion2 Advanced Development Kit board as shown in Table 6. For information on jumper locations, refer to "Appendix 2: Jumper Locations" section on page 23.

**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 jumpers are set accordingly.
J123	2	3	
J124, J121, J32	1	2	JTAG programming via FTDI
J118, J119	1	2	Programming SPI Flash

8. Connect the power supply to the **J42** connector in the SmartFusion2 Advanced Development Kit.
9. 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 connections are given in the following appendix: "Appendix 1: Board Setup for Running the Modbus TCP Reference Design" section on page 22

## 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\\_liberov11p6\\_df](http://soc.microsemi.com/download/rsc/?f=m2s_dg0440_liberov11p6_df)
2. Switch ON the **SW7** power supply switch.
3. Start any serial terminal emulation program such as:
  - HyperTerminal
  - PuTTY
  - TeraTerm

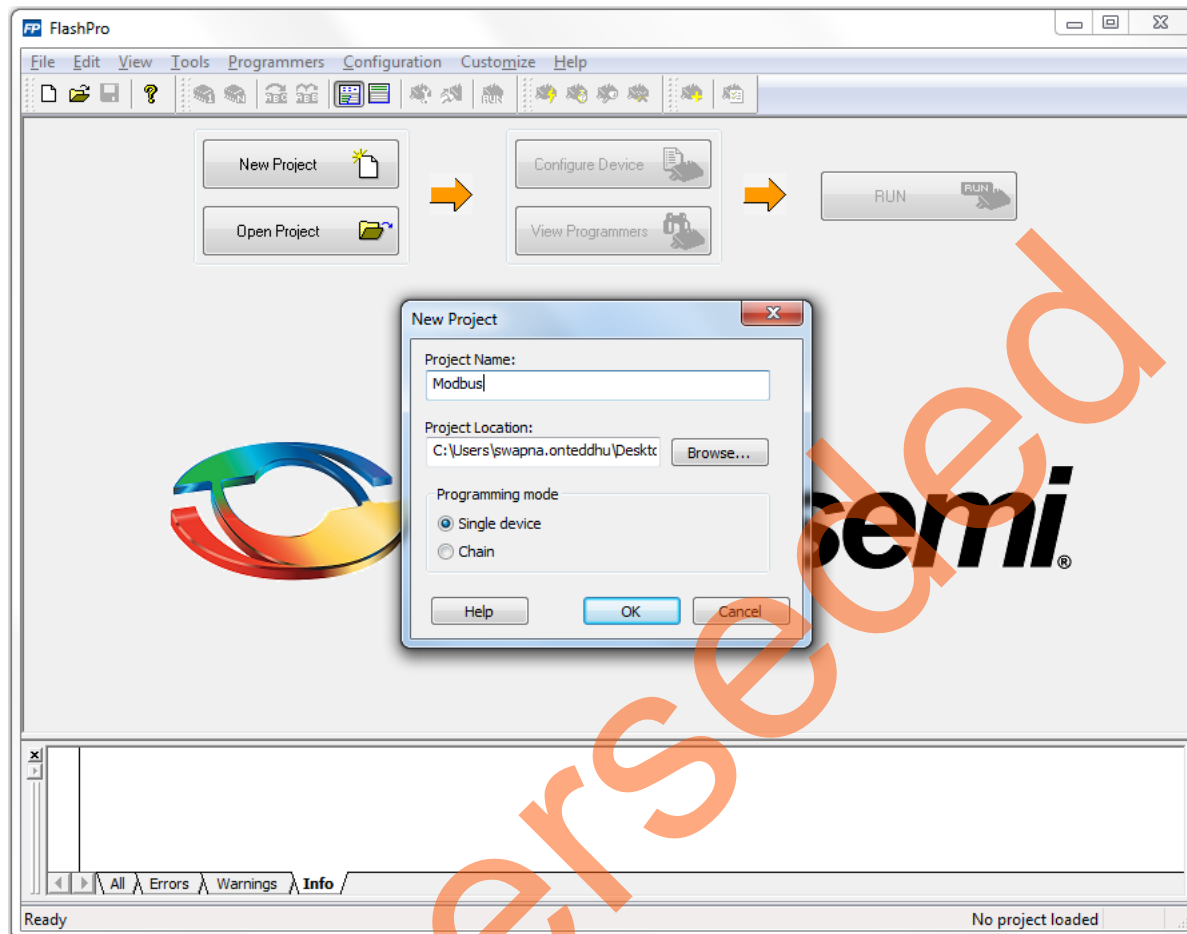
**Note:** In this demo HyperTerminal is used.  
The configuration for the program is:

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

For information on configuring the serial terminal emulation programs, 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.



**Figure 8 • 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.



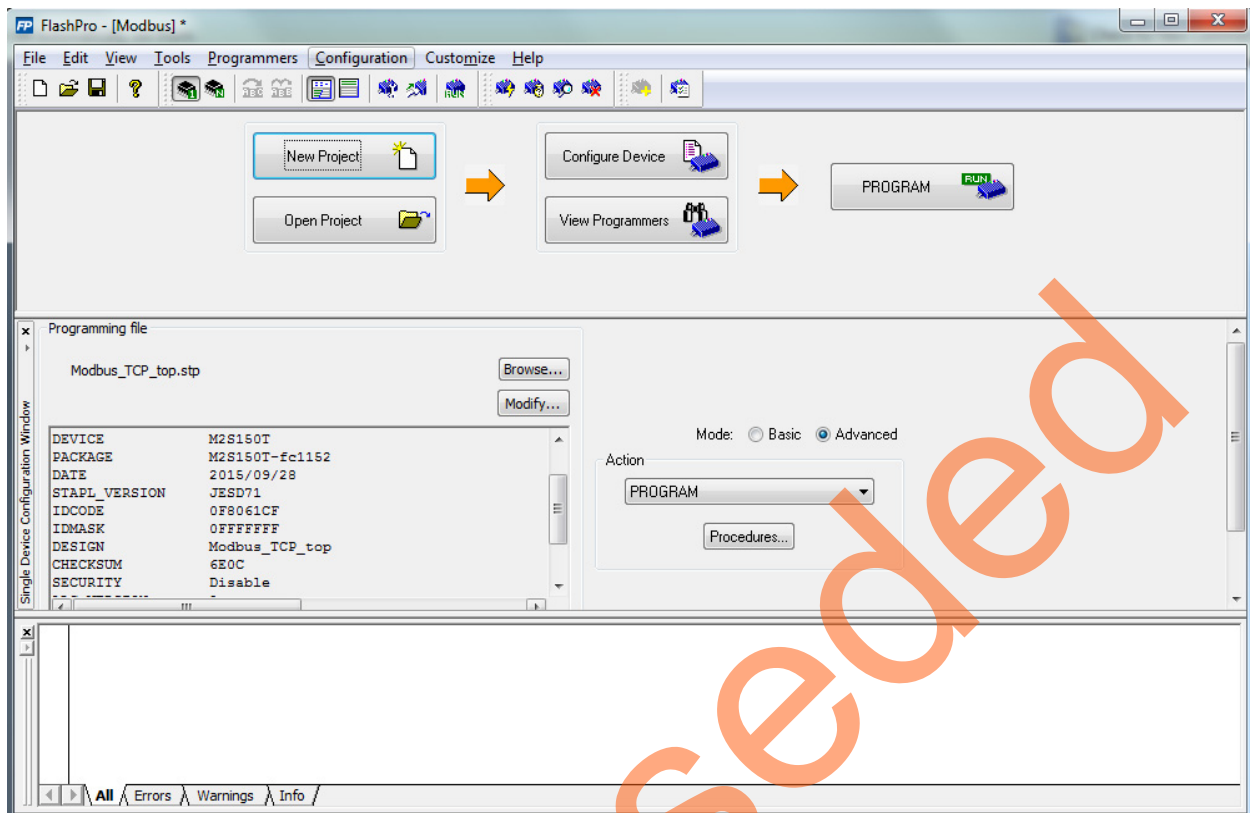
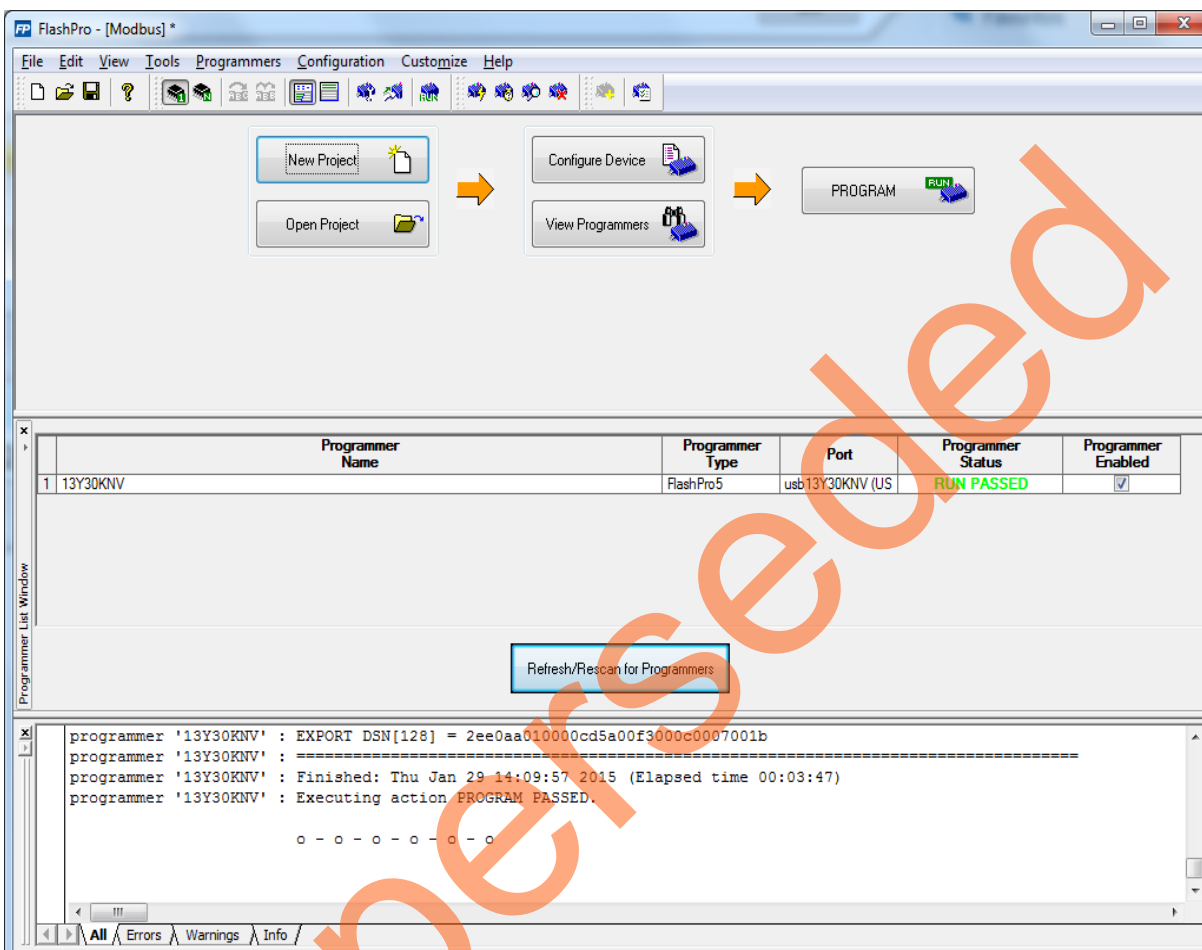


Figure 9 • FlashPro Project Configured

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 10 • FlashPro Program Passed**

Note: To run the design in Static IP mode, follow the steps mentioned in "Appendix 3: Running the Design in Static IP Mode" section on page 24.

13. Power cycle the SmartFusion2 Advanced Development board.

A welcome message with the IP address is displayed in the HyperTerminal window as shown in Figure 11.

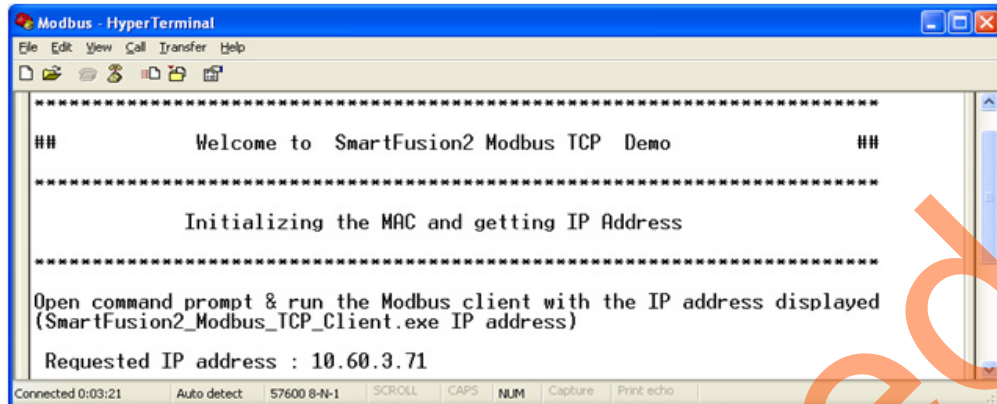


Figure 11 • HyperTerminal with IP Address

14. 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>`

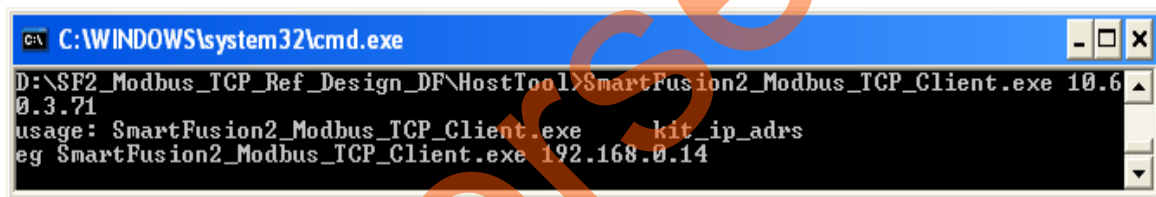


Figure 12 • Invoking the Modbus Client

Figure 13 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)



```

C:\WINDOWS\system32\cmd.exe
D:\SF2_Modbus_TCP_Ref_Design_DF\HostTool>SmartFusion2_Modbus_TCP_Client
0.3.71
usage: SmartFusion2_Modbus_TCP_Client.exe kit_ip_adrs
eg SmartFusion2_Modbus_TCP_Client.exe 192.168.0.14

MB_TCP:02:Read_Desc_inputs:Address at 0x0 = data is 0x48
DIP Switch 4 is ON
Push Button SW4 is ON
MB_TCP:03:Read Holding Reg:Address at 0x0 = data is 0xdb4f
MB_TCP:03:Read Holding Reg:Address at 0x1 = data is 0x2
MB_TCP:04:Read_input_Reg: RTC Secs Counter = 35
MB_TCP:15:Write Multiple Coils: address 0, data 0x0

MB_TCP:02:Read_Desc_inputs:Address at 0x0 = data is 0x48
DIP Switch 4 is ON
Push Button SW4 is ON
MB_TCP:03:Read Holding Reg:Address at 0x0 = data is 0xdc53
MB_TCP:03:Read Holding Reg:Address at 0x1 = data is 0x402
MB_TCP:04:Read_input_Reg: RTC Secs Counter = 38
MB_TCP:15:Write Multiple Coils: address 0, data 0x1

MB_TCP:02:Read_Desc_inputs:Address at 0x0 = data is 0x8
DIP Switch 4 is ON
MB_TCP:03:Read Holding Reg:Address at 0x0 = data is 0xdd58
MB_TCP:03:Read Holding Reg:Address at 0x1 = data is 0x402
MB_TCP:04:Read_input_Reg: RTC Secs Counter = 40
MB_TCP:15:Write Multiple Coils: address 0, data 0x2

```

**Figure 13 • Modbus Functional Codes Demonstration**

Refer to the "Running Modbus Functions" section on page 21 for more information on the Modbus functions that are demonstrated in the reference design.

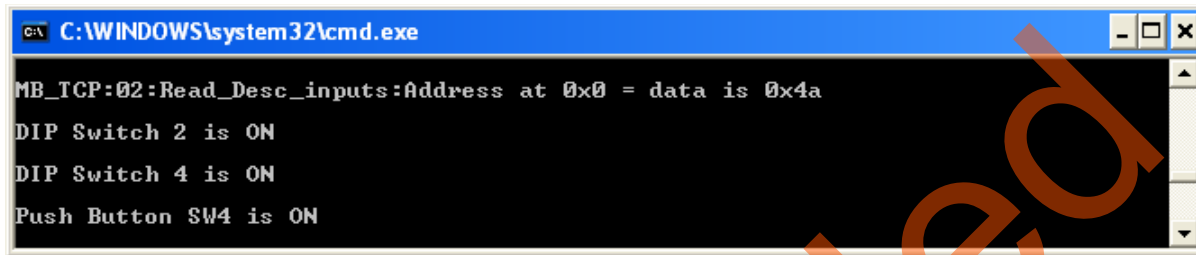
15. After running the demo, close HyperTerminal.

## Running Modbus Functions

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

### **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 [Figure 14](#).



```
C:\WINDOWS\system32\cmd.exe
MB_TCP:02:Read_Desc_inputs:Address at 0x0 = data is 0x4a
DIP Switch 2 is ON
DIP Switch 4 is ON
Push Button SW4 is ON
```

Figure 14 • Read Discrete Inputs

### **Read Holding Registers (function code 03)**

[Figure 15](#) shows the global buffer data defined in the firmware.

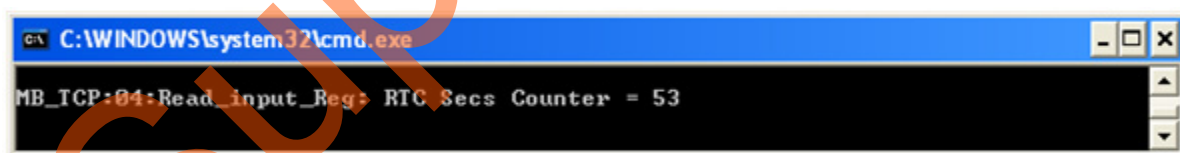


```
C:\WINDOWS\system32\cmd.exe
MB_TCP:03:Read Holding Reg:Address at 0x0 = data is 0xed6
MB_TCP:03:Read Holding Reg:Address at 0x1 = data is 0x402
```

Figure 15 • Read Holding Registers

### **Read Input Registers (function code 04)**

[Figure 16](#) shows the number of seconds that the real-time counter (RTC) has counted.

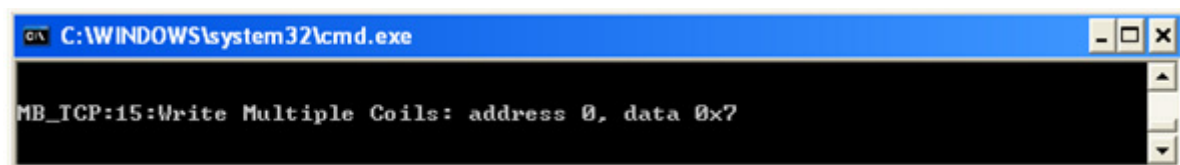


```
C:\WINDOWS\system32\cmd.exe
MB_TCP:04:Read_input_Reg: RTC Secs Counter = 53
```

Figure 16 • Read Input Registers

### **Write Multiple Coils (function code 0x0F)**

[Figure 17](#) shows the Write Multiple Coils register data for toggling the LEDs connected to GPIOs.



```
C:\WINDOWS\system32\cmd.exe
MB_TCP:15:Write Multiple Coils: address 0, data 0x7
```

Figure 17 • Write Multiple Coils

## Appendix 1: Board Setup for Running the Modbus TCP Reference Design

Figure 18 shows the board setup for running the reference design on the SmartFusion2 Advanced Development Kit board.



Figure 18 • SmartFusion2 Advanced Development Kit Setup



## Appendix 2: Jumper Locations

Figure 19 shows the jumper locations on the SmartFusion2 Advanced Development Kit board.



**Figure 19 • SmartFusion2 Advanced Development Kit Silkscreen Top View**

**Notes:**

- Jumpers highlighted in red are set by default.
- Jumpers highlighted in green are must be set manually.
- The location of the jumpers in Figure 19 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 **Project Explorer** window of SoftConsole project and go to **Properties** as shown in Figure 20.

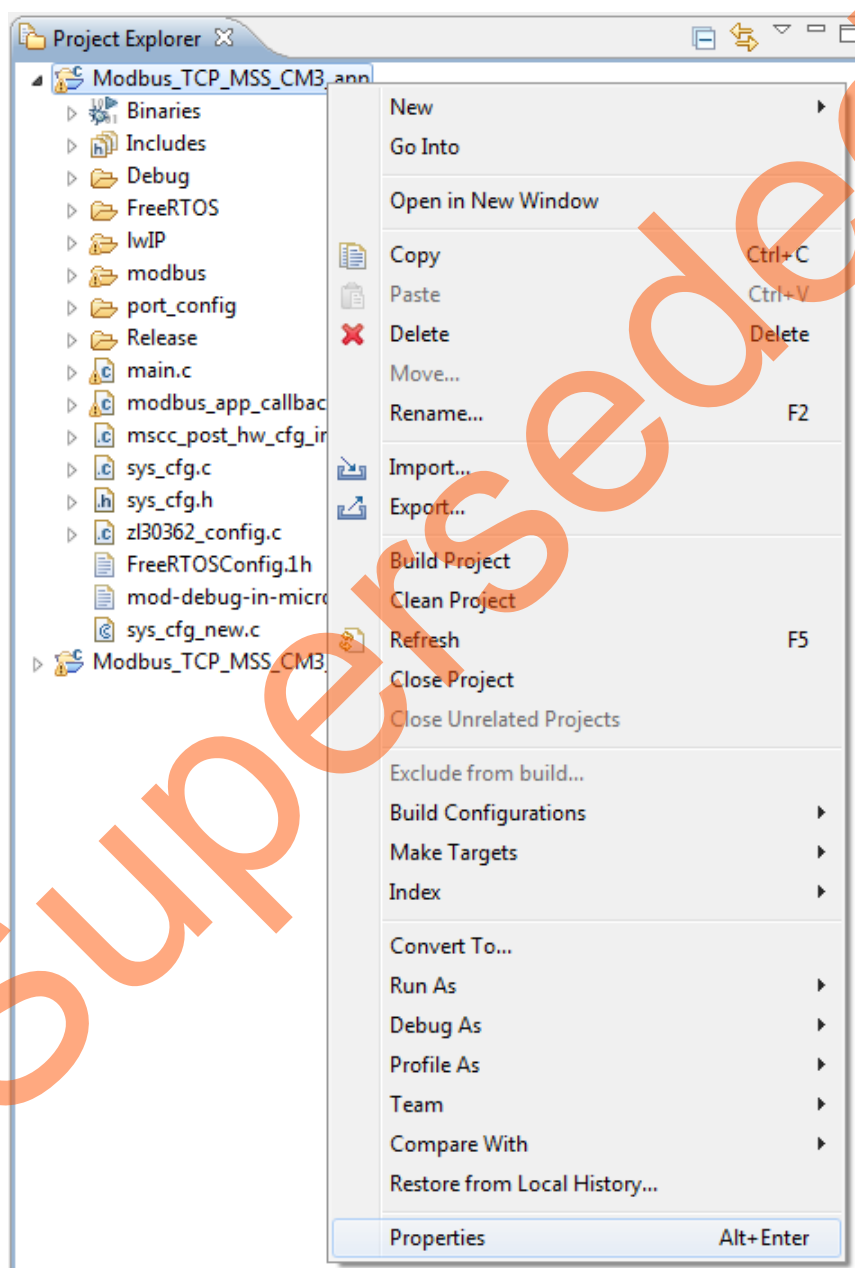


Figure 20 • Project Explorer Window of SoftConsole Project



2. Remove the symbol **NET\_USE\_DHCP** in **Tool Settings** of the **Properties for Modbus\_TCP\_MSS\_CM3\_app** window. Figure 21 shows the **Properties for Modbus\_TCP\_MSS\_CM3\_app** window.

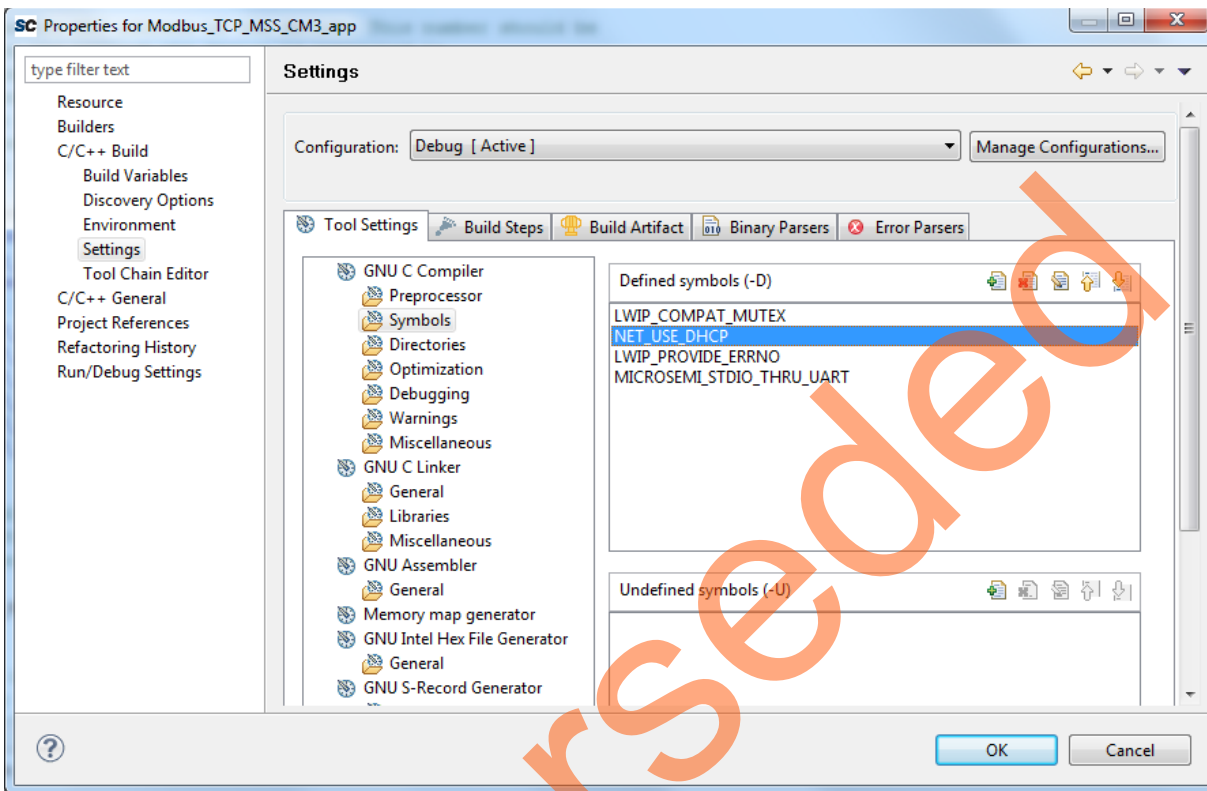
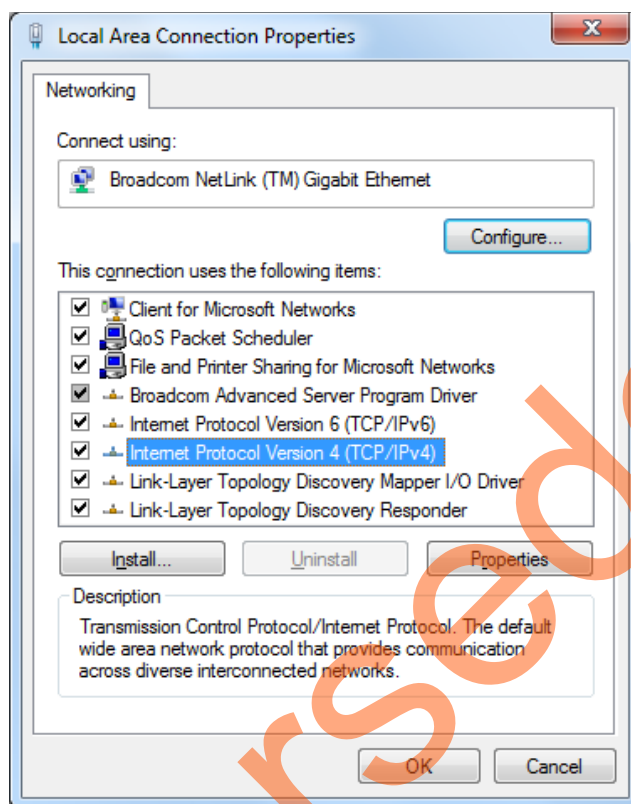
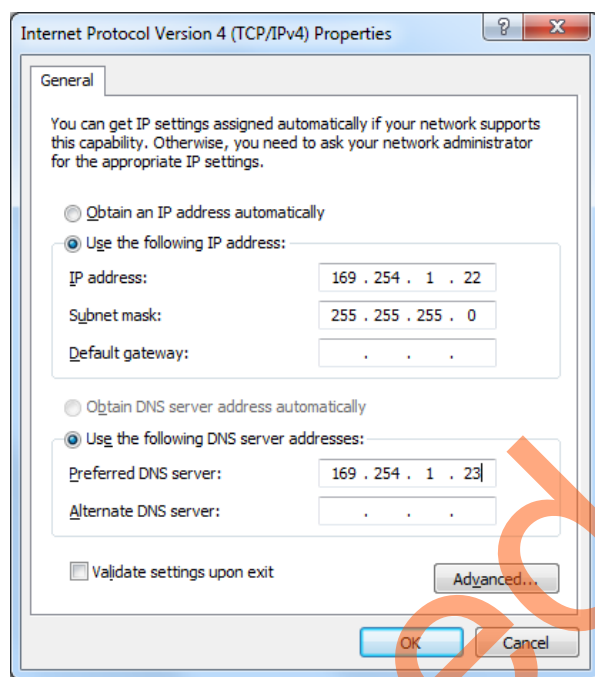


Figure 21 • Project Explorer Properties Window

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. Refer to [Figure 22](#) and [Figure 23](#) on page 27.



**Figure 22 • Host PC TCP/IP Settings**



**Figure 23 • Static IP Address Settings**

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

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

---

## List of Changes

---

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

Date	Changes	Page
Revision 4 (October 2015)	Updated the document for Libero v11.6 software release (SAR 72924).	NA
Revision 3 (March 2015)	Updated the document for Libero v11.5 software release (SAR 63972).	NA
Revision 2 (May 2014)	Updated the document for Libero v11.3 software release (SAR 56538).	NA
Revision 1 (December 2013)	Updated the document for Libero v11.2 software release (SAR 53221).	NA
Revision 0 (October 2013)	Initial release.	NA

Superseded

---

## 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](mailto: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](mailto: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](mailto: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.

Superseded



**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](mailto:sales.support@microsemi.com)

© 2015 Microsemi Corporation. All rights reserved. Microsemi and the Microsemi logo are trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.

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](http://www.microsemi.com).

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.