Running the Modbus TCP Reference Design on SmartFusion2 Devices using lwIP and FreeRTOS

Demo Guide

May 2014

Microsemi®

Revision History

Date	Revision	Change
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.

Table of Contents

Preface 4
About this document 4
Intended Audience 4
References 4
Modbus Protocol Quick References 4
Microsemi Publications 4

Running the 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
Tools Required 7
Demo Design 8
Introduction 8
Demo Design Features 8
Demo Design Description 9
Setting up the Demo Design 13
Board Setup Snapshot 14
Running the Demo Design 14
Running Modbus Functions 20
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

A List of Changes . . . . -31

B Product Support . . . . -32
Customer Service 32
Customer Technical Support Center 32
Technical Support 32
Website 32
Contacting the Customer Technical Support Center 32
Email 32
My Cases 33
Outside the U.S. 33
ITAR Technical Support 33

Preface

About this document

This demo is for SmartFusion®2 system-on-chip (SoC) field programmable gate array (FPGA) devices. It provides instructions on how to use the corresponding reference design.

Intended Audience

The following designers using the SmartFusion2 devices:

- FPGA designers
- Embedded designers
- System-level designers

References

The following references are used in this document:

- Modbus Organization: www.modbus.org
  - FAQ: www.modbus.org/faq.php
  - Technical resources including specifications and links to free and commercial Modbus tools and resources: www.modbus.org/tech.php
- Wikipedia page on Modbus: http://en.wikipedia.org/wiki/Modbus
- FreeModbus home page: www.freemodbus.berlios.de/
  - API Documentation: www.freemodbus.berlios.de/api/index.html
  - Examples using Modpoll: www.freemodbus.berlios.de/index.php?idx=1
- lwIP TCP/IP stack: www.sics.se/~adam/lwip/
- FreeRTOS stack: 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 with updates for the complete set of features of the Modbus layer.

Microsemi Publications

- SmartFusion2 Microcontroller Subsystem User Guide
- SmartFusion2 SoC FPGA High Speed Serial Interfaces User Guide
- Libero SoC User Guide
- SmartFusion2 Development Kit User Guide

Refer the following web page for a complete and up-to-date listing of SmartFusion2 device documentation: www.microsemi.com/soc/products/smartfusion2/docs.aspx.

Running the Modbus TCP Reference Design on SmartFusion2 Using lwIP and FreeRTOS

Introduction

Microsemi® offers a reference design for SmartFusion2 SoC FPGA devices that demonstrates the TSEMAC features of the SmartFusion2 FPGA and implements the Modbus Protocol. The reference design runs on the *SmartFusion2 Development Kit*. This demo guide describes:

- The use of SmartFusion2 TSEMAC connected to an serial gigabit media independent interface (SGMII) PHY.
- The integration of SmartFusion2 MAC driver with the lwIP TCP/IP stack and the FreeRTOS Operating System.
- The application layer with industrial automation protocol, Modbus on TCP/IP.
- How to run the reference design

The MSS of the SmartFusion2 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 *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/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/reply protocol data units. The components of the Modbus protocol include:

- TCP/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.

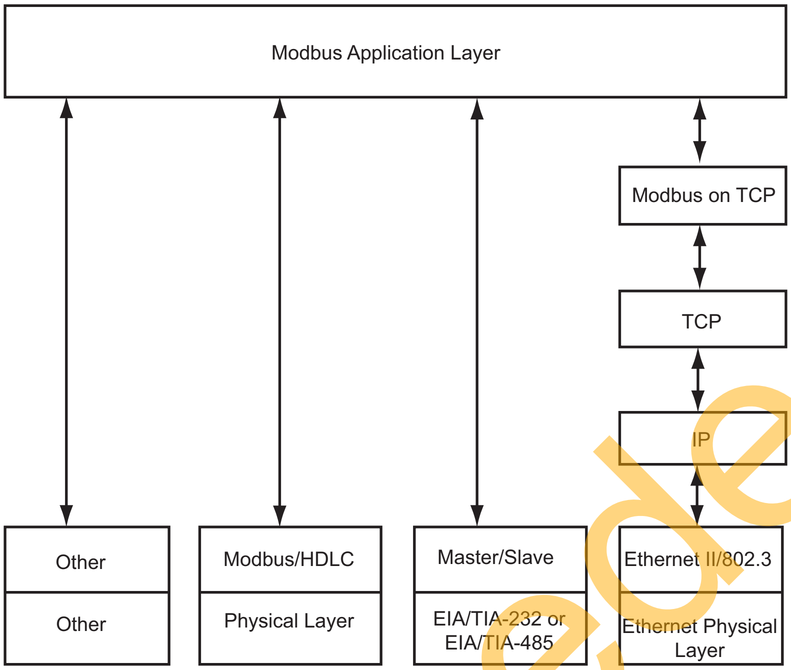


***Figure 1* • Modbus Communication Stack**

For more details 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 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.

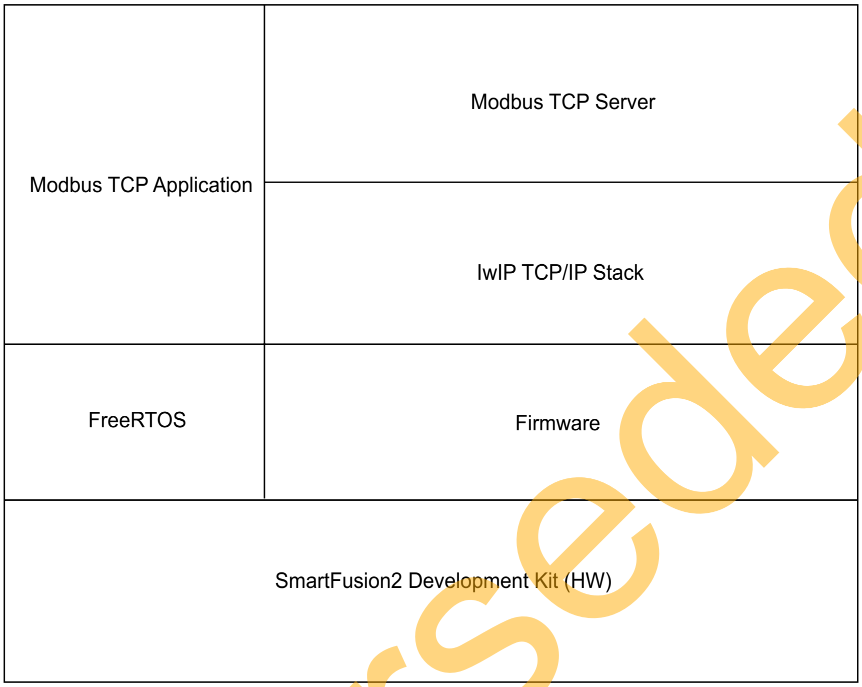


Figure 2 • **Block Diagram of Modbus TCP Server and Application on SmartFusion2 Device**

Tools Required

Table 1 lists the reference design requirements and details for running the modbus TCP reference design demo.

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

Reference Design Requirements and Details	**Description**
Hardware Requirements	
SmartFusion2 Development Kit - FlashPro4 programmer - USB A to mini-B cable - 12 V adapter	Rev C or later
RJ45 cable	
Any one of the following serial terminal emulation programs: - HyperTerminal - Tera Term - PuTTY	
Host PC or Laptop	Any 64-bit Windows Operating System
Software Requirements	
Libero® System-on-Chip (SoC)	11.3

Table 1 • Reference Design Requirements and Details (continued)

SoftConsole	3.4
Flash programming software	11.3
USB to UART drivers	

Demo Design

Introduction

The reference design files are available for download from the Microsemi SoC Products Group website: www.microsemi.com/soc/download/rsc/?f=SF2_Modbus_TCP_Ref_Design_DF.

The design files comprises:

- 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.

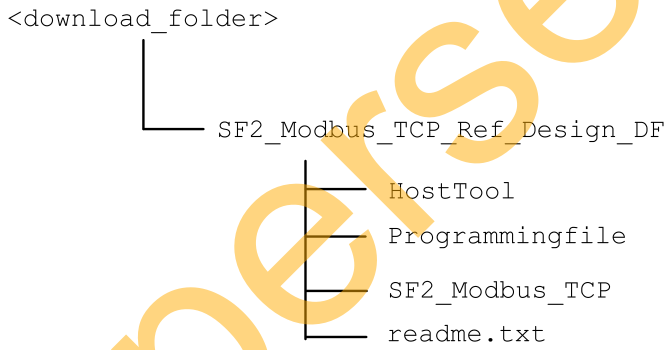


Figure 3 • **Demo Design Files Top-Level Structure**

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/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. The TBI interface is routed through the FPGA fabric onto the SERDES I/Os using the TBI to external physical coding sub layer (EPCS) interface. For more information on the TSEMAC TBI interface, refer to the *SmartFusion2 Microcontroller Subsystem User Guide*.

The design comprises:

- Libero SoC Hardware Project
- SoftConsole Firmware Project

Libero SoC Hardware Project

A Libero SoC v11.3 project uses the SmartFusion2 MSS 1.1.100. Figure 4 shows the hardware design implementation on which the reference design slave firmware runs.

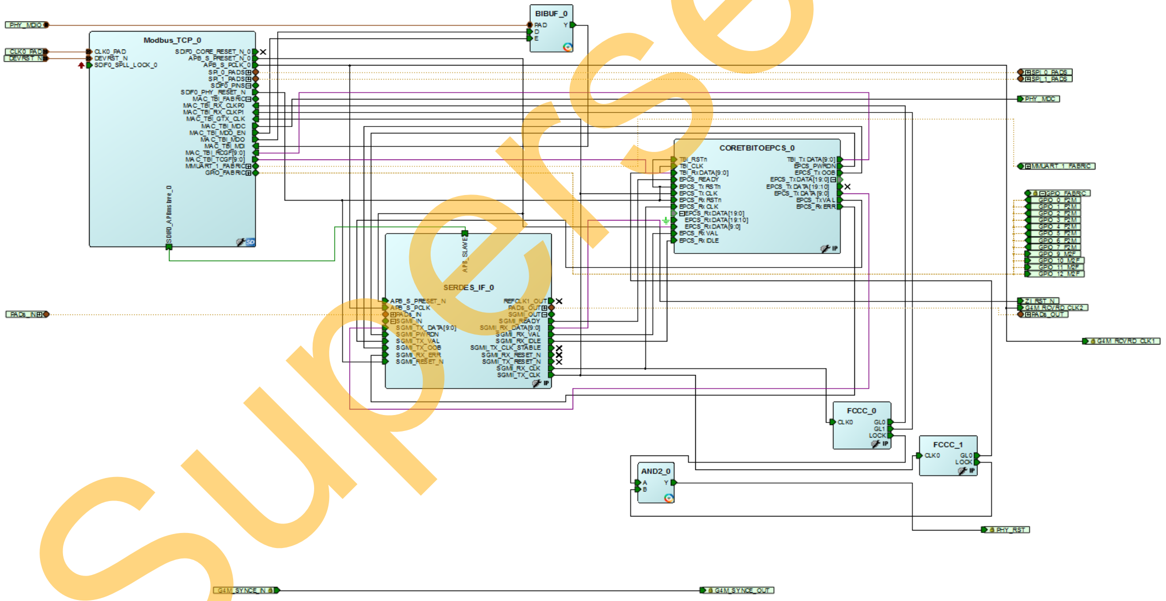


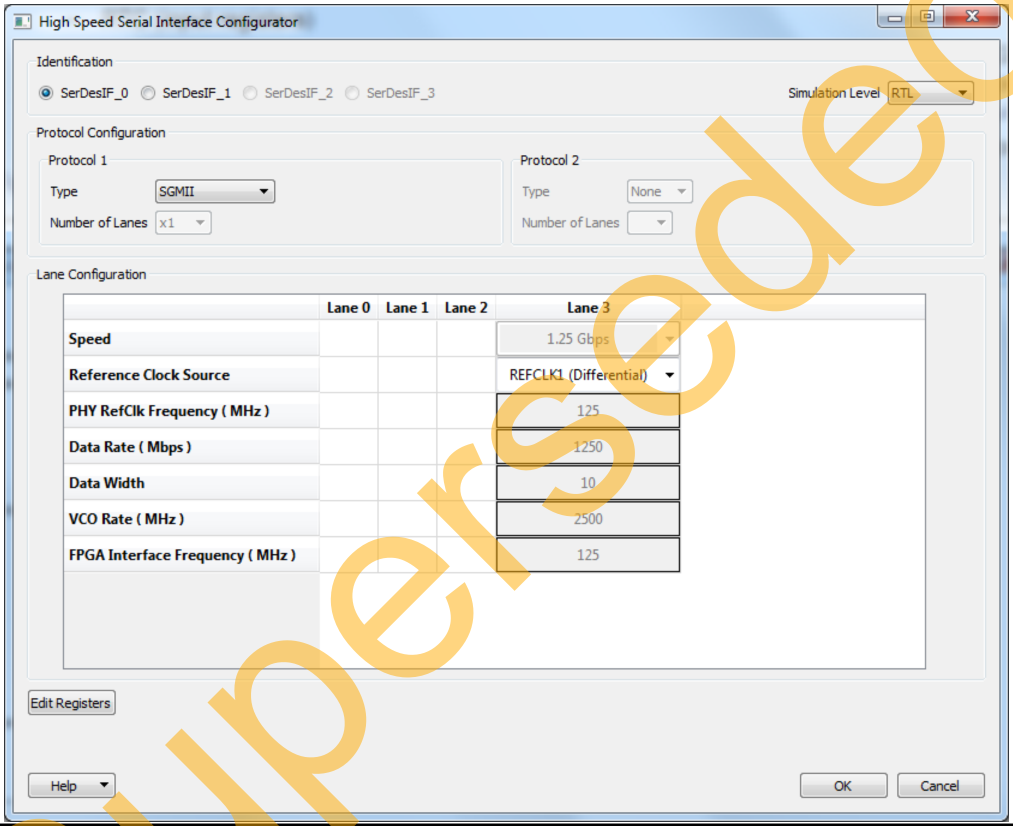
Figure 4 • **Libero Top-Level Design**

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

- TSEMAC TBI interface
- MMUART_1 for RS-232 communications on the development kit
- SPI: Used to configure the ZL30362 clock synthesizer (supplies reference clock to SERDESIF).
- 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_0 SGMII lane3, which is dedicated for the TSEMAC. Refer to Figure 5. To know more about high speed serial interfaces, refer to the *SmartFusion2 SoC FPGA High Speed Serial Interfaces User Guide*.
- CORETBITOEPCS bus interface IP: Acts as a bridge between the TBI and the EPCS interfaces.

Figure 5 shows the **High Speed Serial Interface Configurator** window.



High Speed Serial Interface Configurator

Identification

● SerDesIF_0 ○ SerDesIF_1 ○ SerDesIF_2 ○ SerDesIF_3

Simulation Level: RTL

Protocol Configuration

Protocol 1
Type: SGMII
Number of Lanes: x1

Protocol 2
Type: None
Number of Lanes:

Lane Configuration

	Lane 0	Lane 1	Lane 2	Lane 3
Speed				1.25 Gbps
Reference Clock Source				REFCLK1 (Differential)
PHY RefClk Frequency (MHz)				125
Data Rate (Mbps)				1250
Data Width				10
VCO Rate (MHz)				2500
FPGA Interface Frequency (MHz)				125

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 on page 11.

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

Output	Package Pin
LED_1	A18
LED_2	B18
LED_3	D18
LED_4	E18

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

Output	Package Pin
DIP1	R3
DIP2	R4
DIP3	AE2
DIP4	AD1

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

Output	Package Pin
SWITCH2	AA1
SWITCH3	AA2
SWITCH4	AB1
SWITCH5	AB2

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

Port Name	Direction	Package Pin
PHY_MDC	Output	N5
PHY_MDIO	Input	R7
PHY_RST	Output	N4

SoftConsole Firmware Project

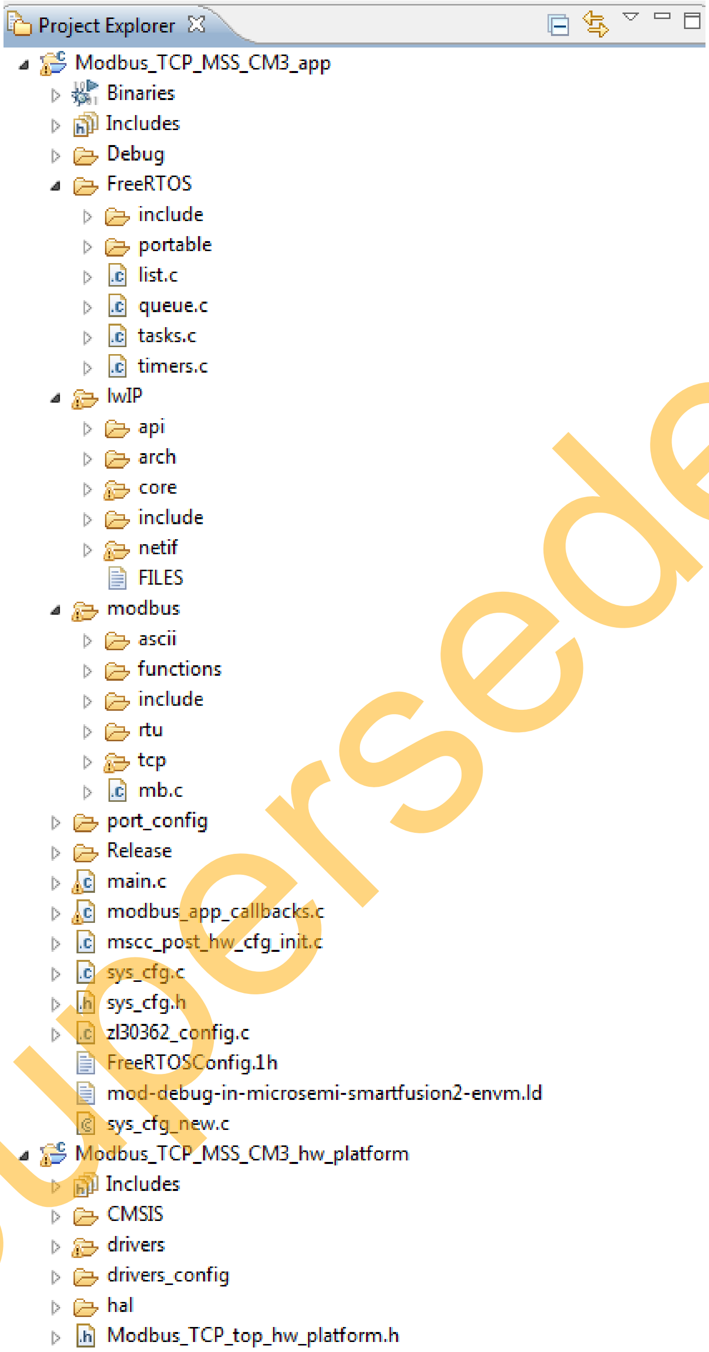
Write Application Code option is available under the **Develop Firmware** tab in the Libero SoC **Design Flow** window to invoke the SoftConsole project. Refer to the *Libero SoC User Guide*.

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

- lwIP TCP/IP stack version 1.3.2 (www.sics.se/~adam/lwip/)
- 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)

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.

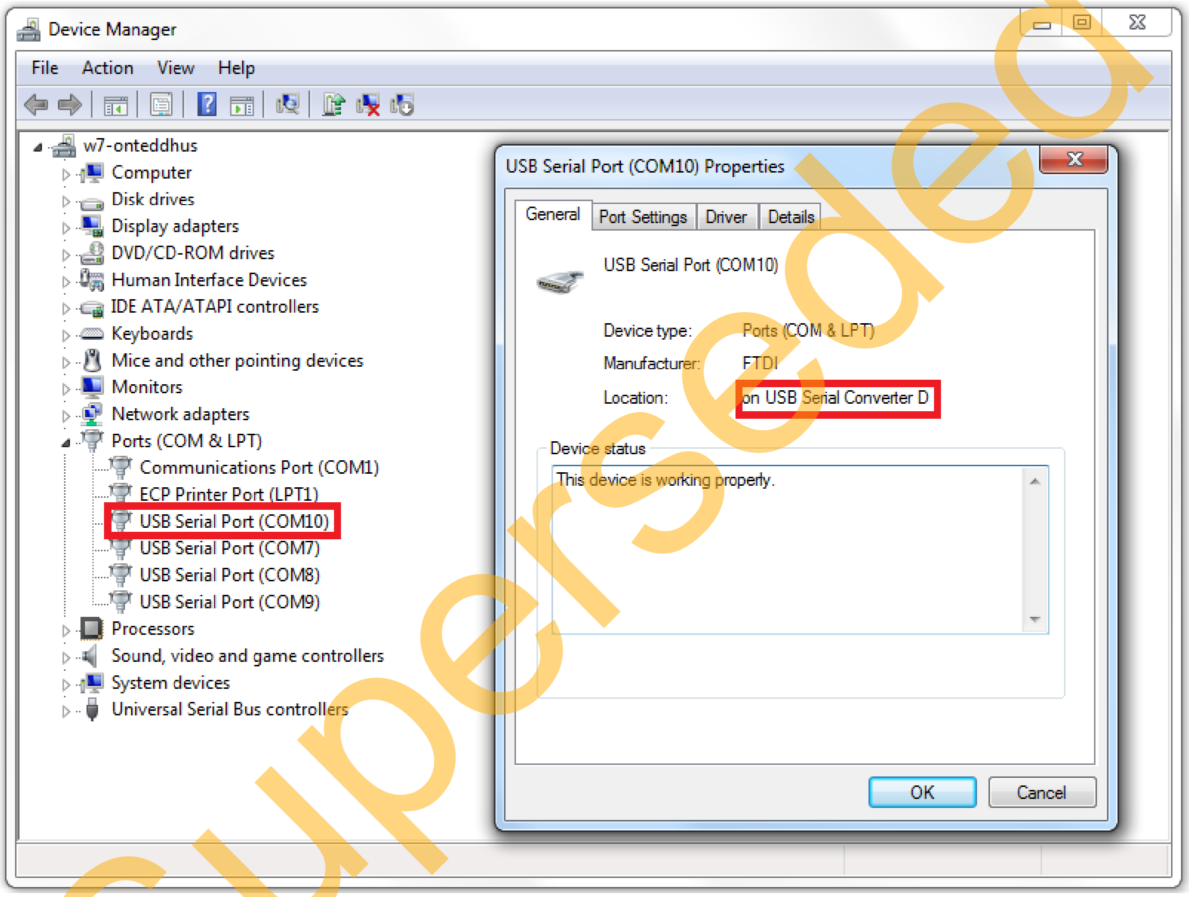


- Modbus_TCP_MSS_CM3_app
  - Binaries
  - Includes
  - Debug
  - FreeRTOS
    - include
    - portable
    - list.c
    - queue.c
    - tasks.c
    - timers.c
  - lwIP
    - api
    - arch
    - core
    - include
    - netif
    - FILES
  - modbus
    - ascii
    - functions
    - include
    - rtu
    - tcp
    - mb.c
  - port_config
  - Release
  - main.c
  - modbus_app_callbacks.c
  - mscc_post_hw_cfg_init.c
  - sys_cfg.c
  - sys_cfg.h
  - zl30362_config.c
  - FreeRTOSConfig.1h
  - mod-debug-in-microsemi-smartfusion2-envm.ld
  - sys_cfg_new.c
- Modbus_TCP_MSS_CM3_hw_platform
  - Includes
  - CMSIS
  - drivers
  - drivers_config
  - hal
  - Modbus_TCP_top_hw_platform.h

Figure 6 • **SoftConsole Project Explorer Window**

Setting up the Demo Design

1. Connect the FlashPro4 programmer to the J59 connector of SmartFusion2 Development Kit Board.
2. Install the USB driver. For serial terminal communication through the FTDI mini USB cable, install the FTDI D2XX driver. Download the drivers and installation guide from: www.microsemi.com/soc/documents/CDM_2.08.24_WHQL_Certified.zip.
3. Connect the host PC to the J24 connector using the USB A to mini-B cable. The USB to UART bridge drivers are automatically detected. Verify if the detection is made in the device manager as shown in Figure 7.



USB Serial Port (COM10)

Device type: Ports (COM & LPT)
Manufacturer: FTDI
Location: on USB Serial Converter D

Device status
This device is working properly.

Figure 7 • Device Manager Window Showing the USB-to-Serial Communication Port

4. Connect the jumpers on the SmartFusion2 Development Kit Board, as described in Table 6. For more information on jumper locations, refer to the "Appendix 2: Jumper Locations" section on page 23.
   - **Caution:** Before making the jumper connections, switch off the power supply switch, SW7.

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

Jumper Number	Settings	Notes
J70, J93, J94, J117, J123, J142, J157, J160, J167, J225, J226, J227	1-2 closed	These are the default jumper settings of the Development Kit. Make sure these jumpers are set properly.
J2	1-3 closed	
J23	2-3 closed	
J129, 133	2-3 closed	These jumpers are required for ZL30362 configuration. All these jumpers are not set by default and must be set manually.
J20, J21, J22, J25	1-2 closed	
J30	2-3 closed	

5. In the SmartFusion2 Development Kit, connect the power supply to the J18 connector.
6. 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 J4 connector of the SmartFusion2 Development Kit Board using an RJ45 cable.
   – For dynamic IP, connect any one of the open network ports to the J4 connector of the SmartFusion2 Development Kit Board using an RJ45 cable.

Board Setup Snapshot

Snapshots of the SmartFusion2 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

1. Download the design file from: www.microsemi.com/soc/download/rsc/?f=SF2_Modbus_TCP_Ref_Design_DF.
2. Switch **ON** the SW7 power supply switch.
3. Start any serial terminal emulation program such as:
   – HyperTerminal
   – PuTTY
   – Tera Term

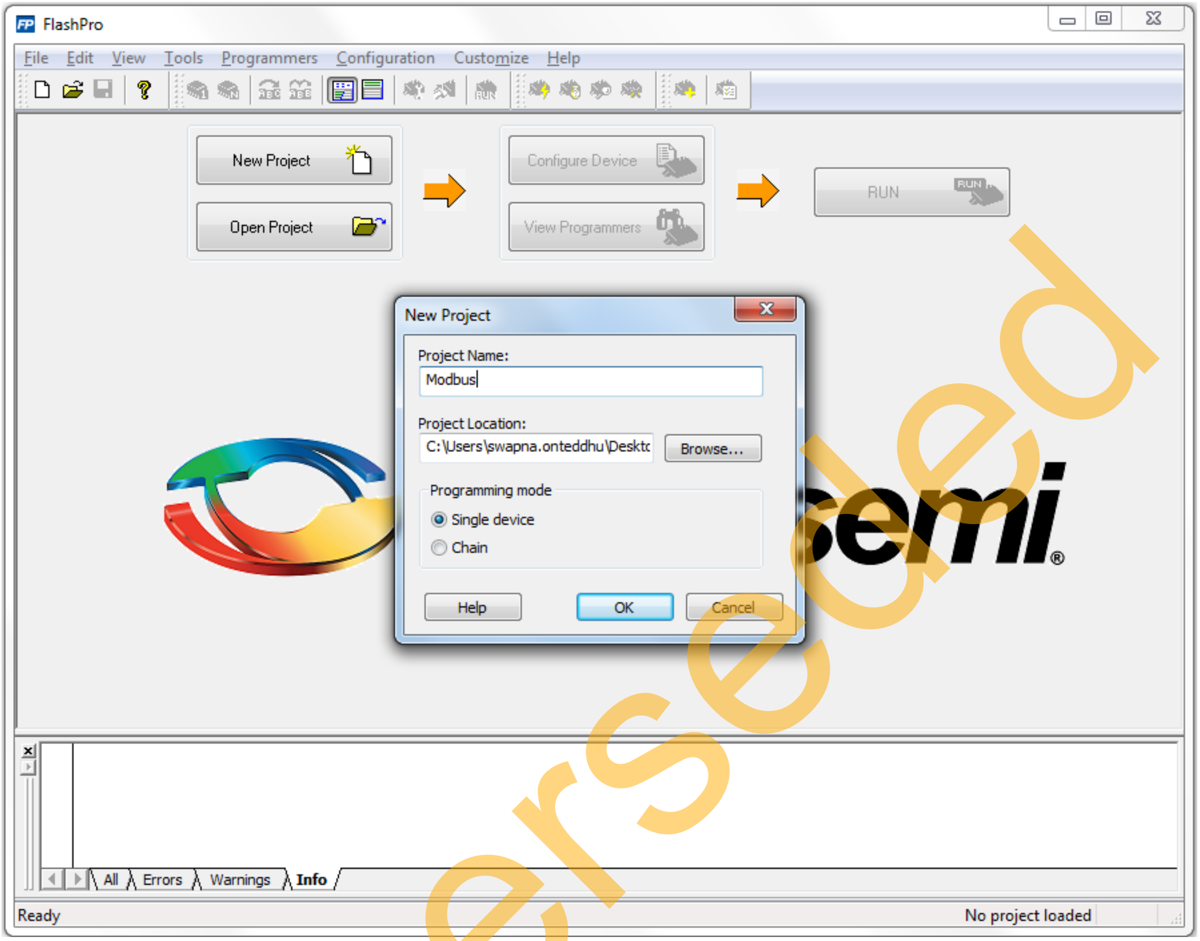
Note: In this demo HyperTerminal is used.
The configuration for the program is:

– Baud Rate: 57600
– 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, type the project name.



Project Name: Modbus

Project Location: C:\Users\swapna.onteddhu\Desktc

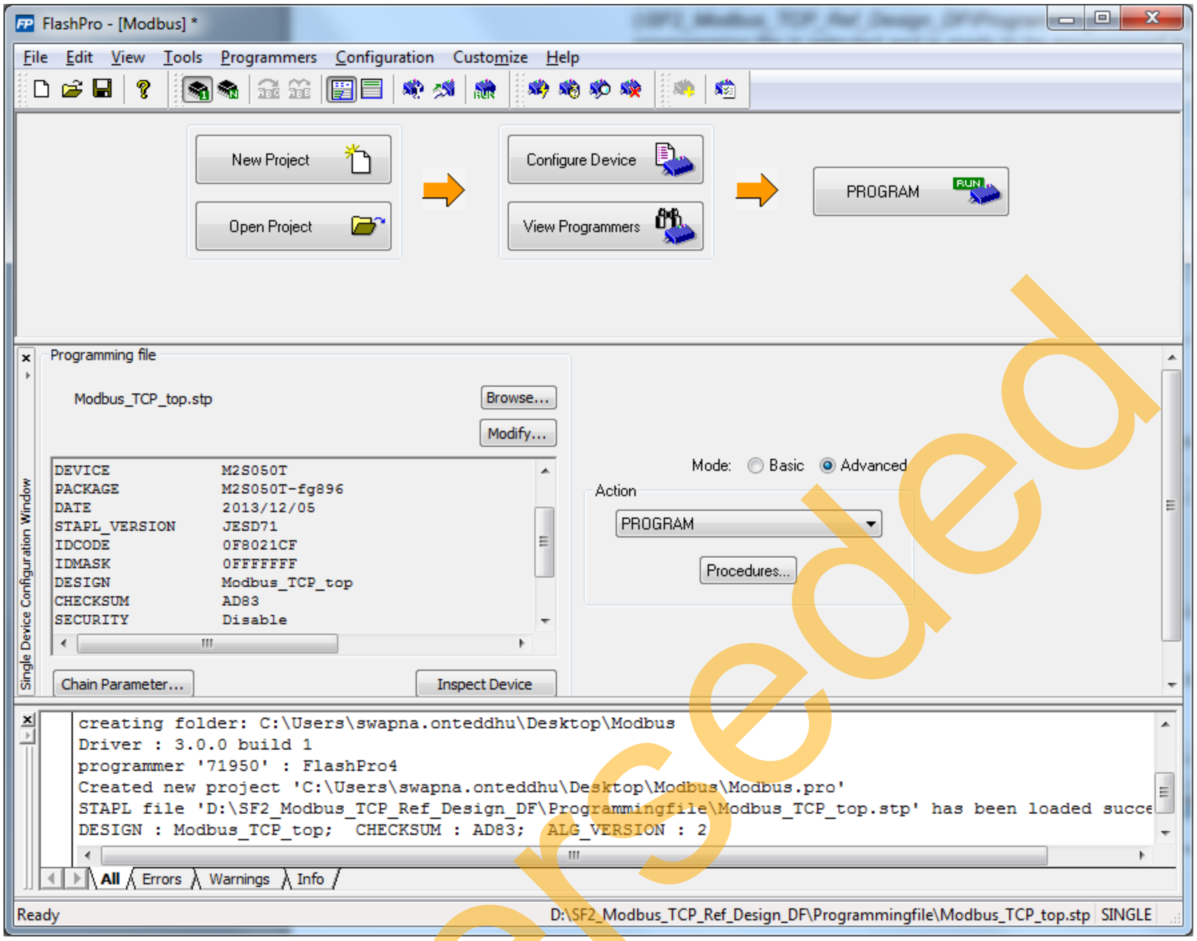
Programming mode

● Single device

○ Chain

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.



Modbus_TCP_top.stp

```
DEVICE          M2S050T
PACKAGE         M2S050T-fg896
DATE            2013/12/05
STAPL_VERSION   JESD71
IDCODE          0F8021CF
IDMASK          0FFFFFFF
DESIGN          Modbus_TCP_top
CHECKSUM        AD83
SECURITY        Disable
```

Mode: ○ Basic ● Advanced

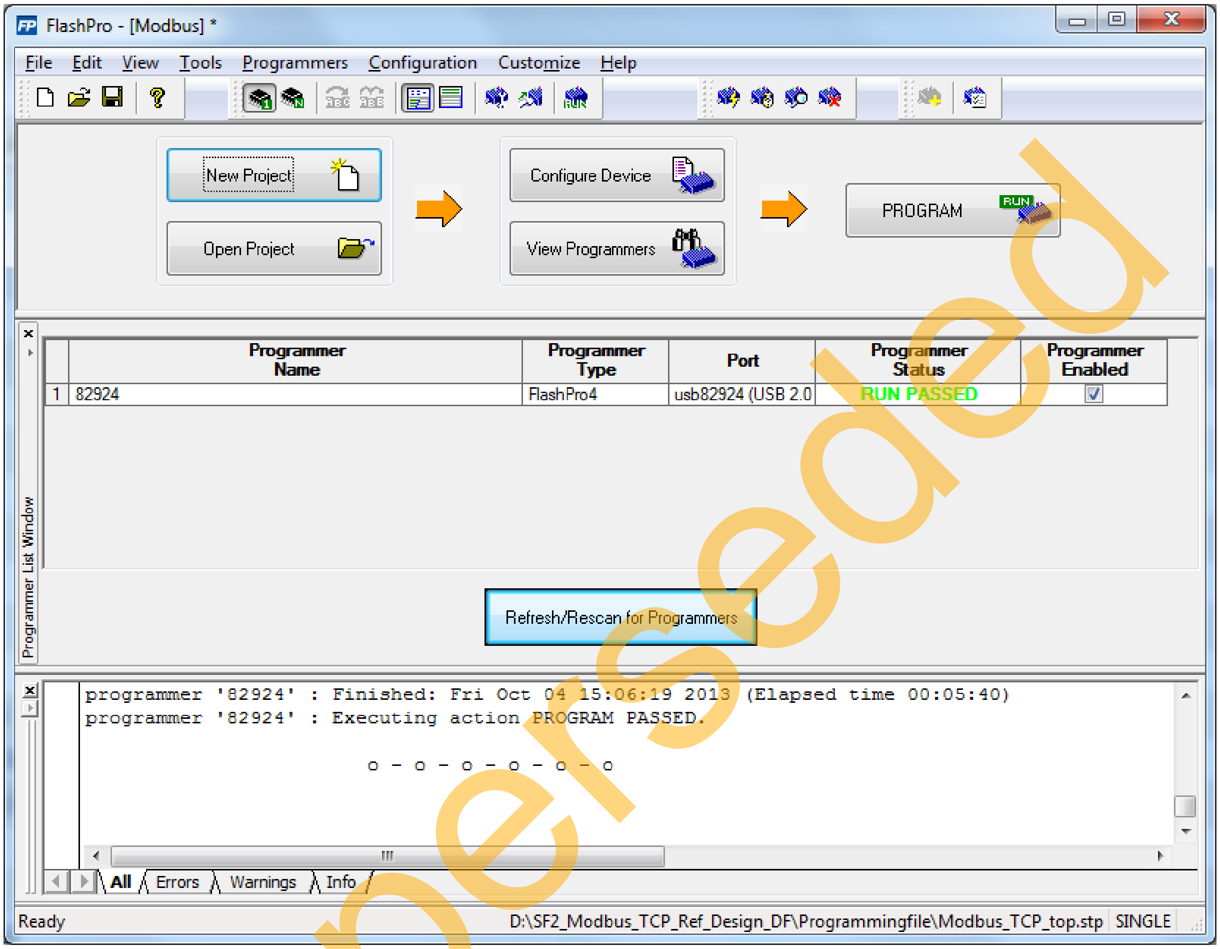
Action: PROGRAM

```
creating folder: C:\Users\swapna.onteddhu\Desktop\Modbus
Driver : 3.0.0 build 1
programmer '71950' : FlashPro4
Created new project 'C:\Users\swapna.onteddhu\Desktop\Modbus\Modbus.pro'
STAPL file 'D:\SF2_Modbus_TCP_Ref_Design_DF\Programmingfile\Modbus_TCP_top.stp' has been loaded succe
DESIGN : Modbus_TCP_top;  CHECKSUM : AD83;  ALG_VERSION : 2
```

Ready D:\SF2_Modbus_TCP_Ref_Design_DF\Programmingfile\Modbus_TCP_top.stp SINGLE

Figure 9 • **FlashPro Project Configured**

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


FlashPro - [Modbus] *
File Edit View Tools Programmers Configuration Customize Help
New Project
Open Project
Configure Device
View Programmers
PROGRAM
RUN
Refresh/Rescan for Programmers
Programmer List Window
All Errors Warnings Info
Ready
D:\SF2_Modbus_TCP_Ref_Design_DF\Programmingfile\Modbus_TCP_top.stp
SINGLE

	Programmer Name	Programmer Type	Port	Programmer Status	Programmer Enabled
1	82924	FlashPro4	usb82924 (USB 2.0	RUN PASSED	☑

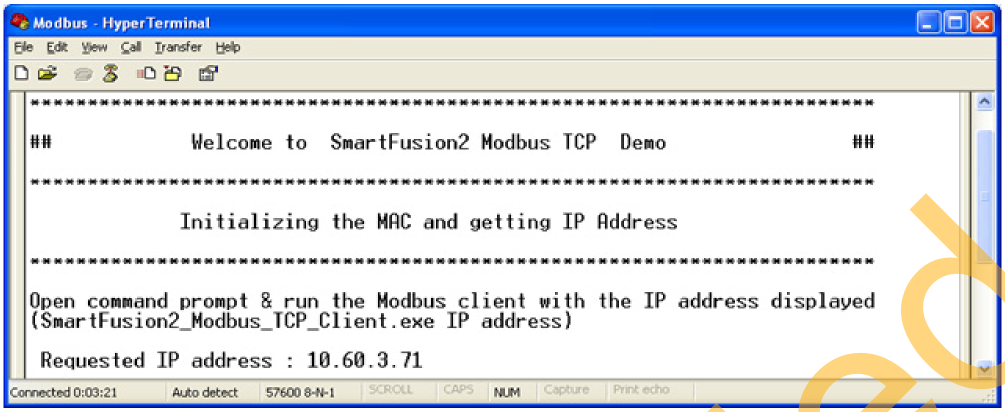
```
programmer '82924' : Finished: Fri Oct 04 15:06:19 2013 (Elapsed time 00:05:40)
programmer '82924' : Executing action PROGRAM PASSED.

                    o - o - o - o - o - o
```

Figure 10 • **FlashPro Program Passed**

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

13. Power cycle the SmartFusion2 Development board.

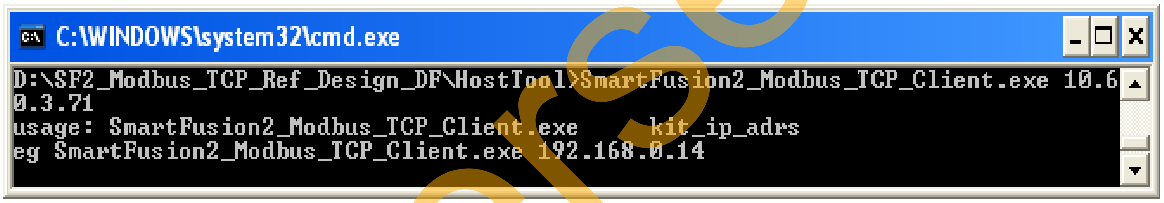
A welcome message with the IP address is displayed in the HyperTerminal window as shown in Figure 11.



```
*************************************************************************
##            Welcome to  SmartFusion2 Modbus TCP  Demo                ##
*************************************************************************
             Initializing the MAC and getting IP Address
*************************************************************************
Open command prompt & run the Modbus client with the IP address displayed
(SmartFusion2_Modbus_TCP_Client.exe IP address)
 Requested IP address : 10.60.3.71
```

***Figure 11* • HyperTerminal with IP Address**

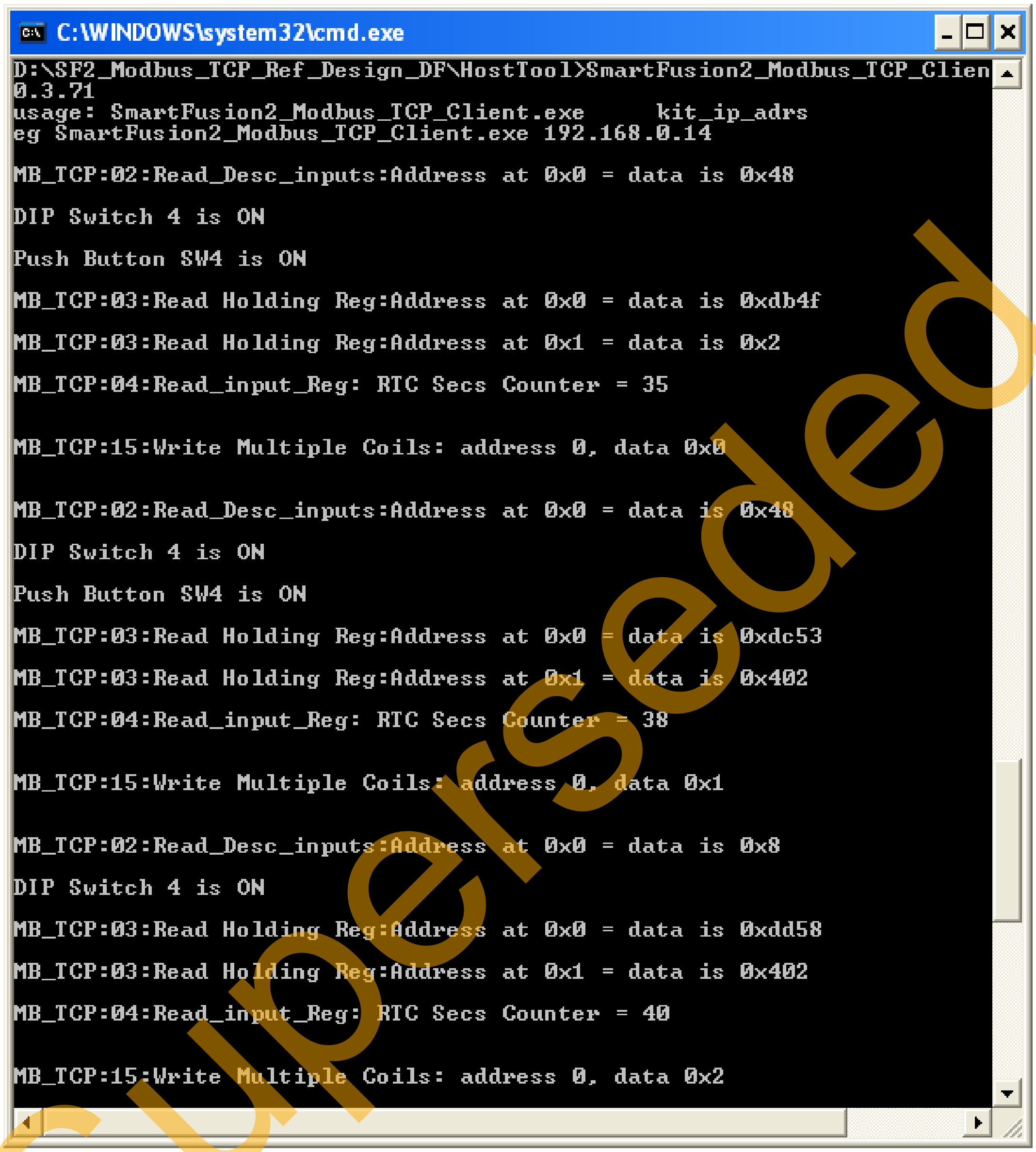
14. Open a new command prompt on the host PC, go to the folder (\SF2_Modbus_TCP_Ref_Design_DF\HostTool) where SmartFusion2_Modbus_TCP_Client.exe file is present, type the below command: `SmartFusion2_Modbus_TCP_Client.exe <IP address>`



```
D:\SF2_Modbus_TCP_Ref_Design_DF\HostTool>SmartFusion2_Modbus_TCP_Client.exe 10.6
0.3.71
usage: SmartFusion2_Modbus_TCP_Client.exe     kit_ip_adrs
eg SmartFusion2_Modbus_TCP_Client.exe 192.168.0.14
```

***Figure 12* • Invoking the Modbus Client**

15. Figure 13 on page 19 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)



```
D:\SF2_Modbus_TCP_Ref_Design_DF\HostTool>SmartFusion2_Modbus_TCP_Clien
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 20 to know about the Modbus functions that are demonstrated in the reference design.

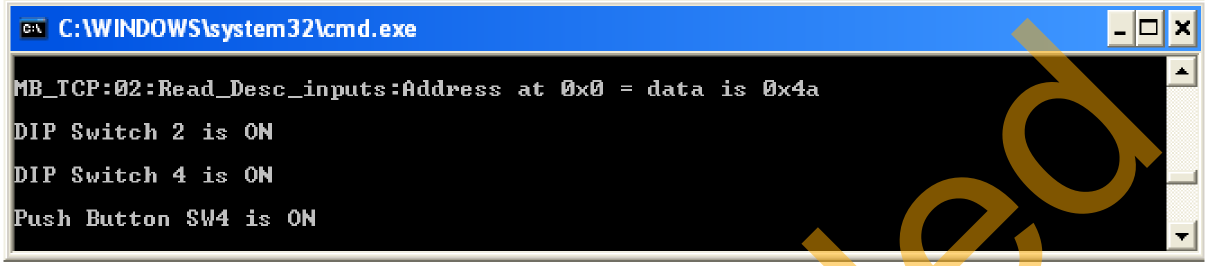
16. After running the demo, close HyperTerminal.

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 SmartFusion2 Development Kit. Read discrete inputs functional code displays the statuses of switches as shown in Figure 14.

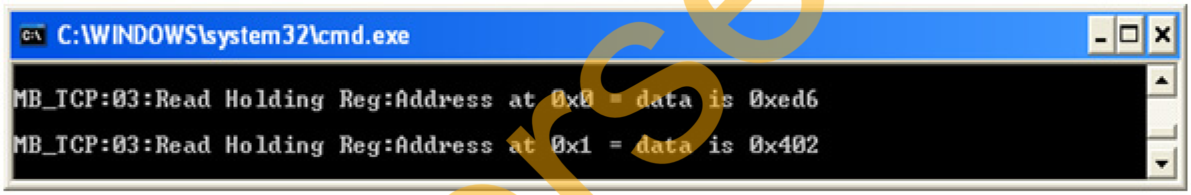


```
MB_TCP:02:Read_Desc_inputs:Address at 0x0 = data is 0x4a
DIP Switch 2 is ON
DIP Switch 4 is ON
Push Button SW4 is ON
```

Figure 14 • **Read Discrete Inputs**

Read Holding Registers (function code 03)

Figure 15 shows the global buffer data defined in the firmware.

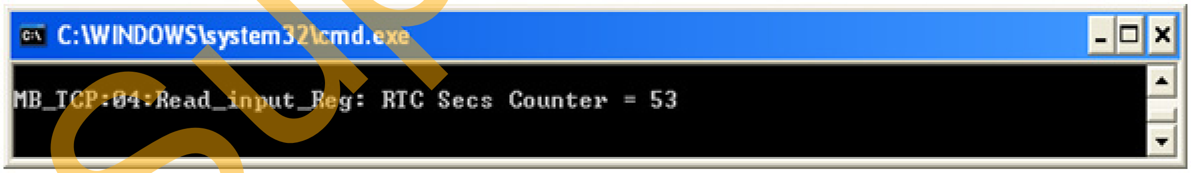


```
MB_TCP:03:Read Holding Reg:Address at 0x0 = data is 0xed6
MB_TCP:03:Read Holding Reg:Address at 0x1 = data is 0x402
```

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.

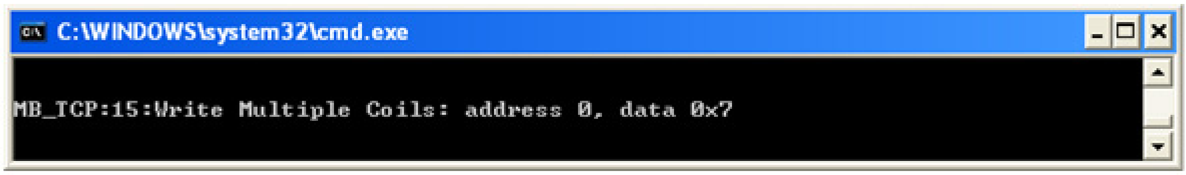


```
MB_TCP:04:Read_input_Reg: RTC Secs Counter = 53
```

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.



```
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 Development Kit Board.

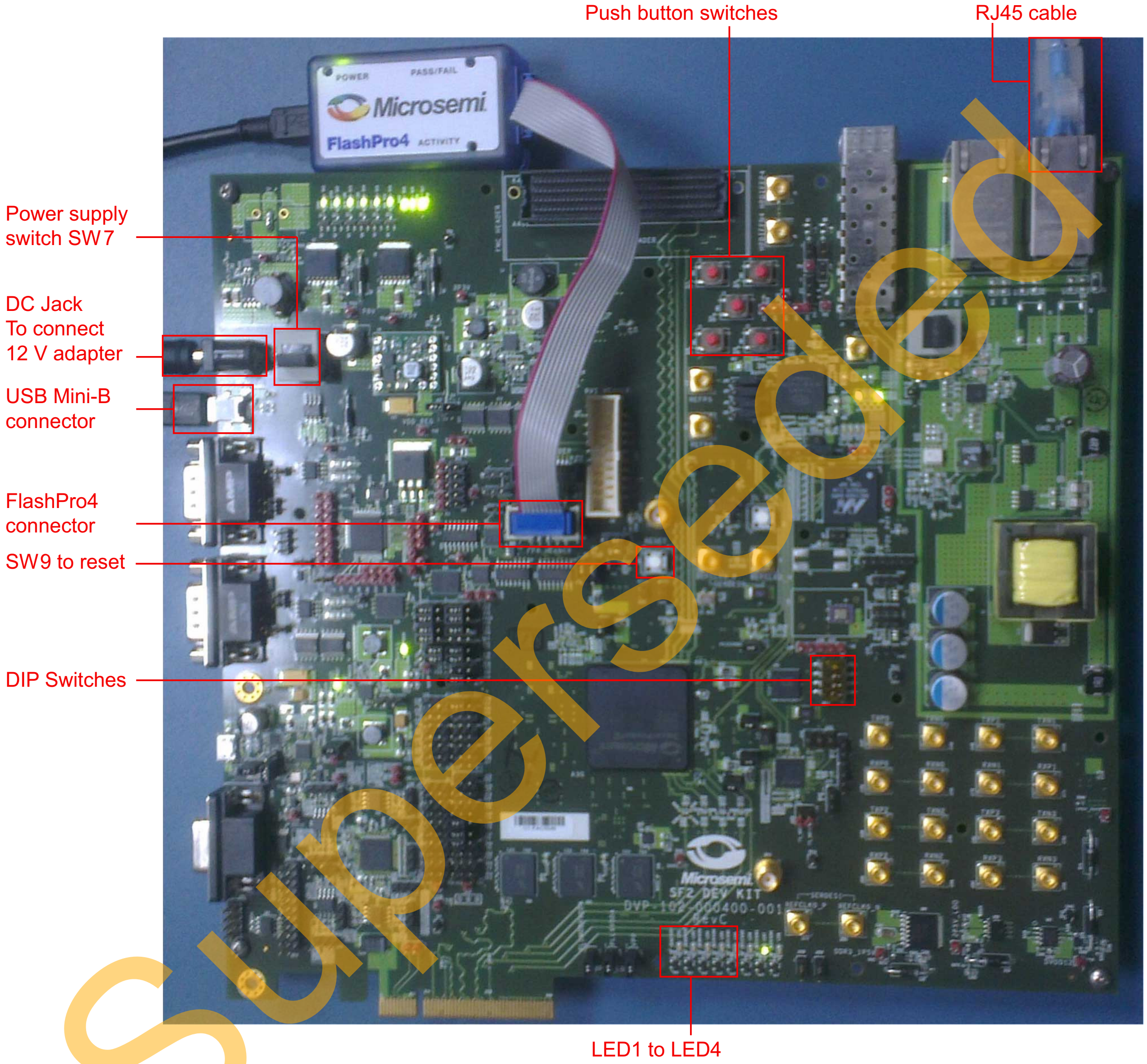


Figure 18 • **SmartFusion2 Development Kit Setup**

Appendix 2: Jumper Locations

Figure 19 shows the jumper locations in the SmartFusion2 Development Kit Board.

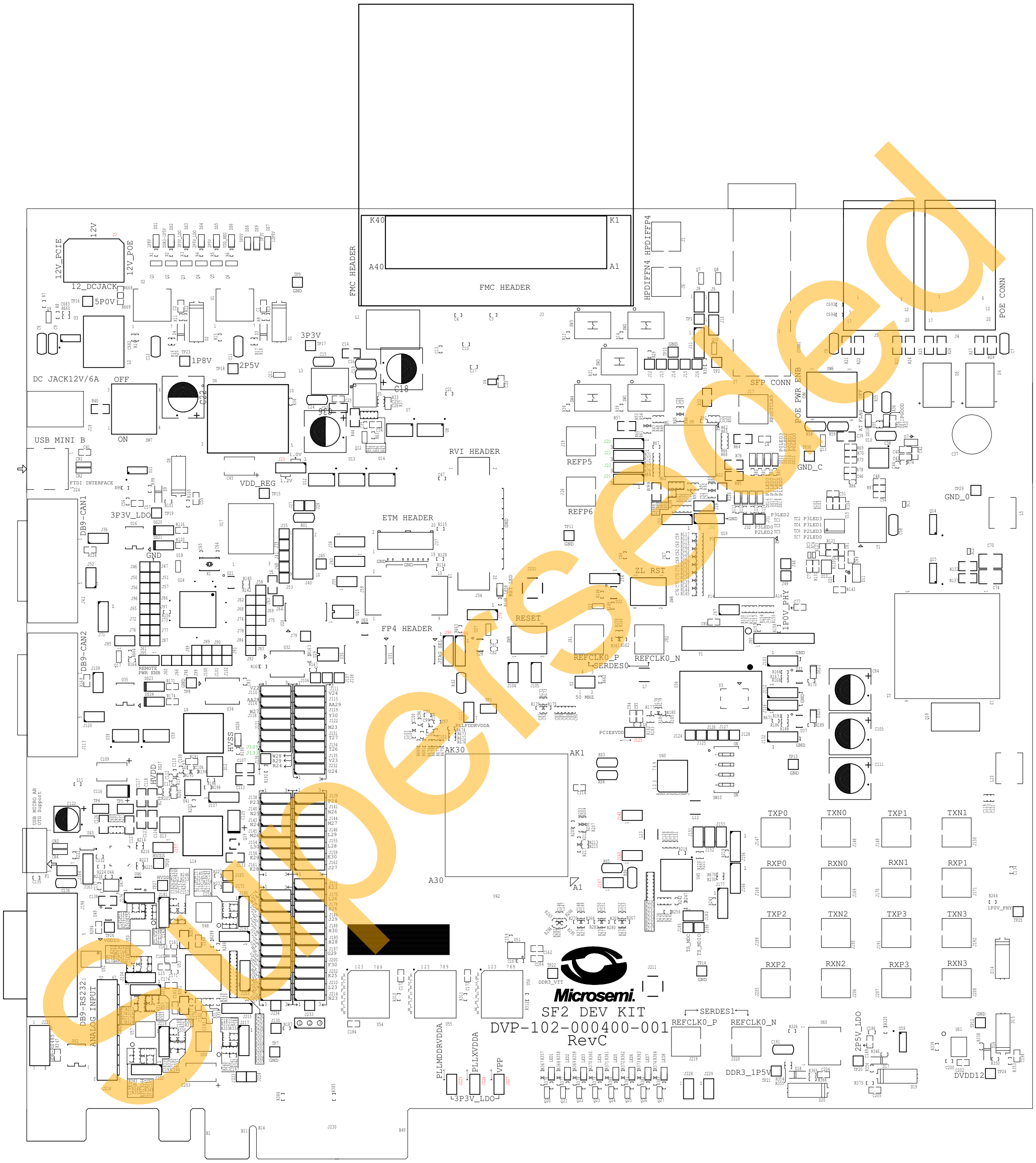


Figure 19 • **SmartFusion2 Development Kit Silkscreen Top View**

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

1. To run the design in Static IP mode, right-click on the **Project Explorer** window of SoftConsole project and go to **Properties** as shown in Figure 20.

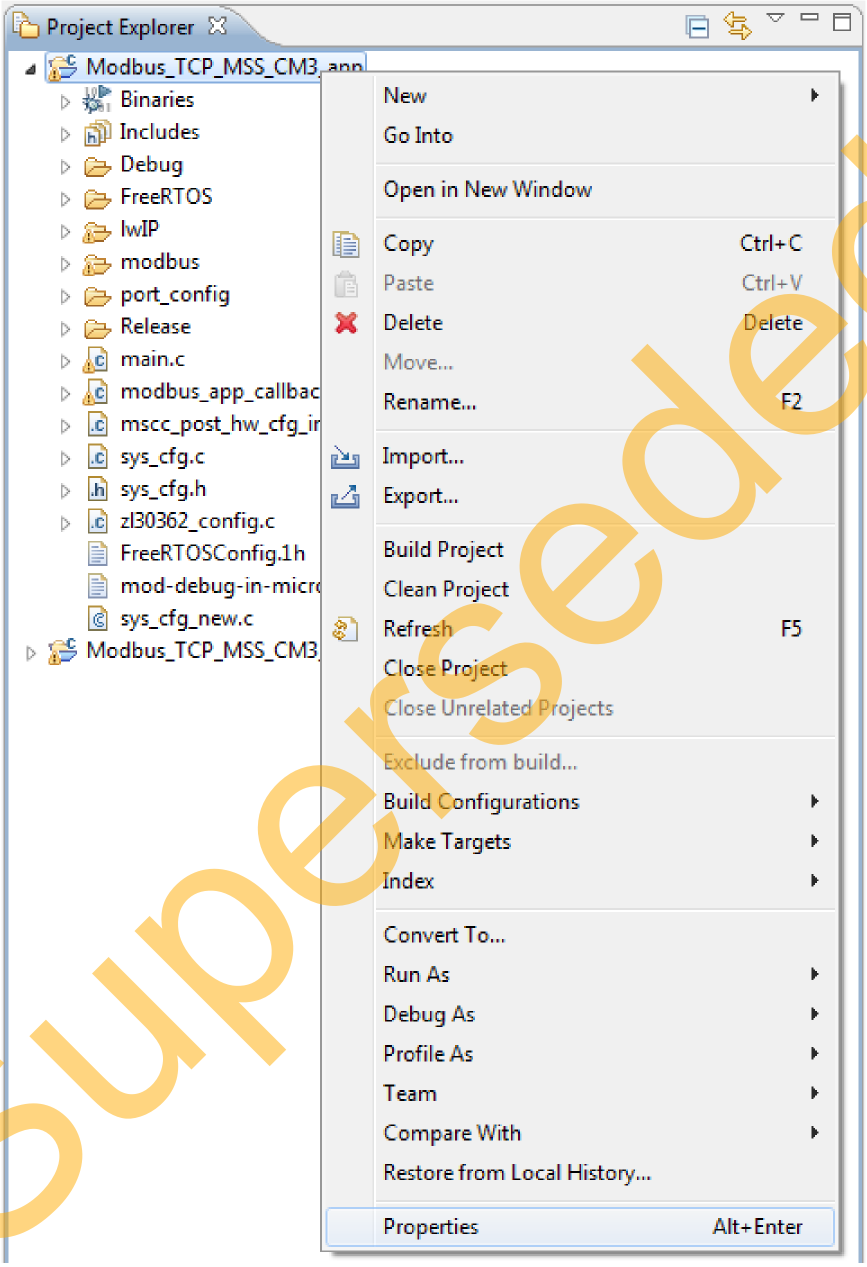
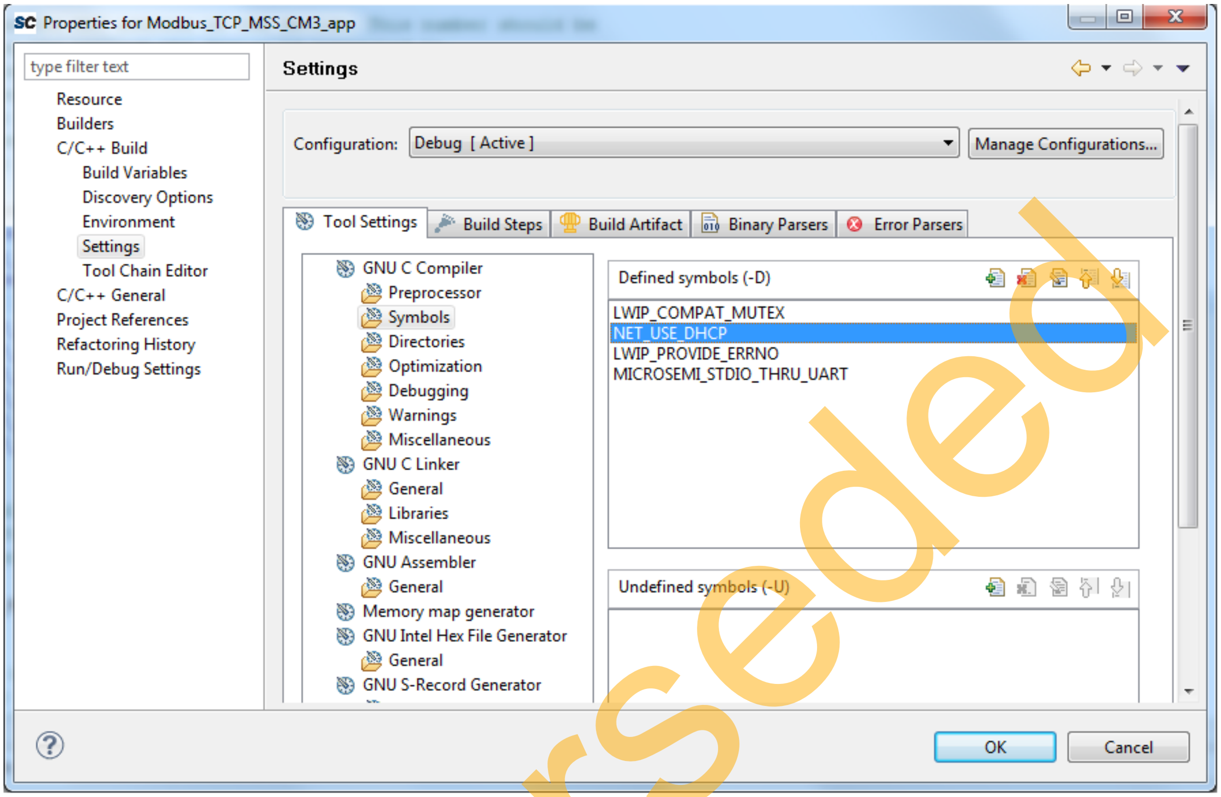


***Figure 20* • Project Explorer Window of SoftConsole Project**

2. Remove the symbol **NET_USE_DHCP** in **Tool Settings** of the **Properties for Modbus_TCP_CM3** window. Figure 21 shows the **Properties for Modbus_TCP_CM3** window.



Configuration: Debug [Active]

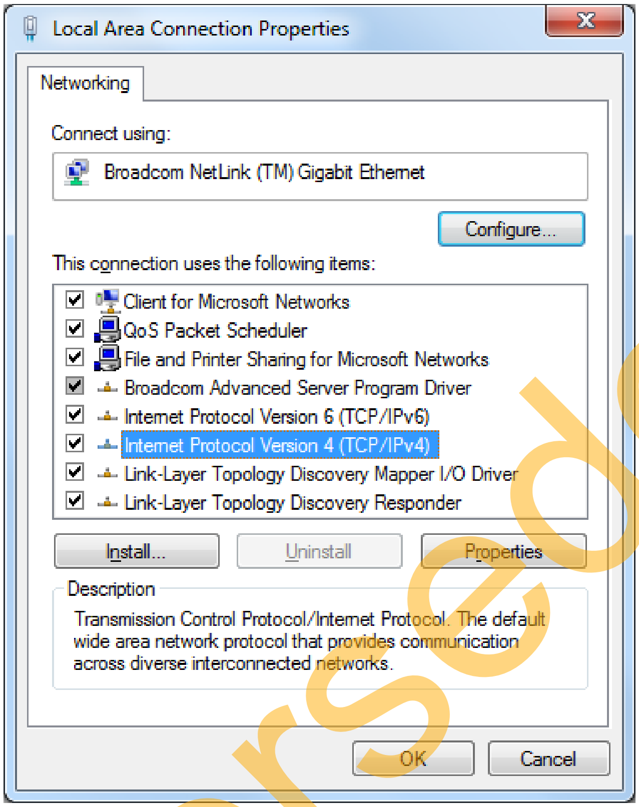
Defined symbols (-D)

LWIP_COMPAT_MUTEX
NET_USE_DHCP
LWIP_PROVIDE_ERRNO
MICROSEMI_STDIO_THRU_UART

Undefined symbols (-U)

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.

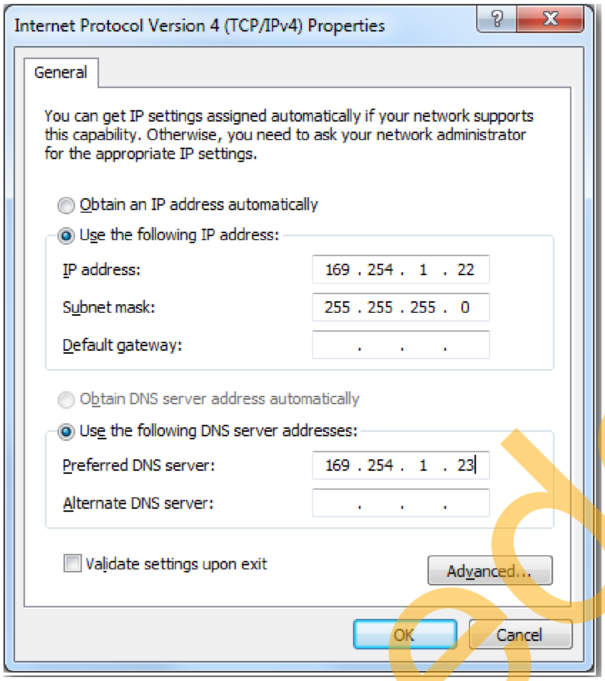


Connect using:
Broadcom NetLink (TM) Gigabit Ethernet

This connection uses the following items:
- ☑ Client for Microsoft Networks
- ☑ QoS Packet Scheduler
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ Broadcom Advanced Server Program Driver
- ☑ Internet Protocol Version 6 (TCP/IPv6)
- ☑ Internet Protocol Version 4 (TCP/IPv4)
- ☑ Link-Layer Topology Discovery Mapper I/O Driver
- ☑ Link-Layer Topology Discovery Responder

Description
Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

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



You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically
● Use the following IP address:
IP address: 169 . 254 . 1 . 22
Subnet mask: 255 . 255 . 255 . 0
Default gateway: . . .

○ Obtain DNS server address automatically
● Use the following DNS server addresses:
Preferred DNS server: 169 . 254 . 1 . 23
Alternate DNS server: . . .

☐ Validate settings upon exit

***Figure 23* • Static IP Address Settings**

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

A – List of Changes

The following table lists critical changes that were made in each revision of the chapter in the demo guide.

Date	Changes	Page
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

B – Product Support

Microsemi SoC Products Group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, electronic mail, and worldwide sales offices. This appendix contains information about contacting Microsemi SoC Products Group and using these support services.

Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From North America, call 800.262.1060
From the rest of the world, call 650.318.4460
Fax, from anywhere in the world, 408.643.6913

Customer Technical Support Center

Microsemi SoC Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions about Microsemi SoC Products. The Customer Technical Support Center spends a great deal of time creating application notes, answers to common design cycle questions, documentation of known issues, and various FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Technical Support

Visit the Customer Support website (www.microsemi.com/soc/support/search/default.aspx) for more information and support. Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on the website.

Website

You can browse a variety of technical and non-technical information on the SoC home page, at www.microsemi.com/soc.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center. The Technical Support Center can be contacted by email or through the Microsemi SoC Products Group website.

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is soc_tech@microsemi.com.

My Cases

Microsemi SoC Products Group customers may submit and track technical cases online by going to My Cases.

Outside the U.S.

Customers needing assistance outside the US time zones can either contact technical support via email (soc_tech@microsemi.com) or contact a local sales office. Sales office listings can be found at www.microsemi.com/soc/company/contact/default.aspx.

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_itar@microsemi.com. Alternatively, within My Cases, select **Yes** in the ITAR drop-down list. For a complete list of ITAR-regulated Microsemi FPGAs, visit the ITAR web page.

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
E-mail: sales.support@microsemi.com

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense and 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; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif. and has approximately 3,400 employees globally. Learn more at **www.microsemi.com**.

50200440-2/05.14