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





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

© 2017 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 makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.

About Microsemi

Microsemi 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.



Contents

1	Revisi	ion His	tory		1	
	1.1	Revisio	n 7.0		1	
	1.2	Revisio	n 6.0		1	
	1.3	Revisio	n 5.0		1	
	1.4	Revisio	n 4.0		1	
	1.5	Revisio	n 3.0		1	
	1.6	Revisio	n 2.0		1	
	1.7	Revisio	n 1.0		1	
2	Runni FreeR	ng Moo TOS .	dbus TCP Reference Design on SmartFusion2 Devices Using IwIF	^{>} and	2	
	2.1	Introduc	tion		2	
		2.1.1	Using the Modbus Protocol		2	
		2.1.2	Using Modbus Protocol on SmartFusion2 Device		4	
	2.2 Design Requirements					
	2.3	Demo D	Design		5	
		2.3.1	Demo Design Features		6 6	
	2.4	Setting	Up the Demo Design		1	
		2.4.1	Board Setup Snapshot	1	2	
	2.5	Running	g the Demo Design	1	2	
		2.5.1	Running Modbus Functions	1	7	
3	Appendix: Board Setup for Running the Modbus TCP Reference Design					
4	Appendix: Jumper Locations					
5	Appendix: Running the Design in Static IP Mode					



Figures

Figure 1	Modbus Communication Stack	3
Figure 2	Block Diagram of Modbus TCP Server and Application on SmartFusion2 4	1
Figure 3	Demo Design Files Top-Level Structure	3
Figure 4	Libero SoC Top-Level Hardware Design	7
Figure 5	High Speed Serial Interface Configurator Window	3
Figure 6	SoftConsole Project Explorer Window)
Figure 7	Demo Design Driver Versions)
Figure 8	Device Manager Window	1
Figure 9	FlashPro New Project	3
Figure 10	FlashPro Project Configured	1
Figure 11	FlashPro Program Passed	5
Figure 12	HyperTerminal with IP Address	5
Figure 13	Invoking the Modbus Client	3
Figure 14	Modbus Functional Codes Demonstration	3
Figure 15	Read Discrete Inputs	7
Figure 16	Read Holding Registers	7
Figure 17	Read Input Registers	7
Figure 18	Write Multiple Coils	7
Figure 19	SmartFusion2 Advanced Development Kit Board Setup 18	3
Figure 20	SmartFusion2 Advanced Development Kit Silkscreen Top View	9
Figure 21	Project Explorer Window of SoftConsole Project)
Figure 22	Project Explorer Properties Window	1
Figure 23	Host PC TCP/IP Settings	1
Figure 24	Static IP Address Settings	2



Tables

Table 1	Reference Design Requirements and Details	5
Table 2	LED to Package Pins Assignments	8
Table 3	DIP Switches to Package Pins Assignments	8
Table 4	Push Button Switches to Package Pins Assignments	9
Table 5	PHY Interface Signals to Package Pins Assignments	9
Table 6	SmartFusion2 Advanced Development Kit Jumper Settings 1	2



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).



2.1 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 (IwIP) 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

Medhus TCD Application	Modbus TCP Server
	IwIP TCP or IP Stack
FreeRTOS	Firmware
SmartFusion2 Adv	anced Development Kit (HW)



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 IwIP 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 ${\tt Readme.txt}$ 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.





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

○ SerDesIF_0 ○ SerDesIF_1 ○ SerDesIF_	2 SerDesIF_3			Simulation Level RTL -	
Protocol Configuration					
Protocol 1		Protocol 2			
Type EPCS 💌		Type	ne v		
Number of Lanes x1	3 🔹	Number of Lanes	-		
Lane Configuration					
-	Lane 0	Lane 1	Lane 2	Lane 3	
Speed				Custom Speed 👻	
Reference Clock Source				REFCLK1 (Differential) 🗸	
PHY RefClk Frequency (MHz)				125	
Data Rate (Mbps)				1250 Mbps (10 bit) -	
Data Width				10	
FPGA Interface Frequency (MHz)				125	
VCO Rate (MHz) 2500					
Signal Integrity Options	Register Configura	ation			
Signal Integrity Options Edit Registers					

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

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:

- IwIP 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.





The SoftConsole workspace consists of the project, Modbus_TCP_App that has the Modbus TCP application (which uses IwIP 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	V	đ	-	SmartFusion2_CMSIS_0	SmartFusion2_CMSIS	2.3.105 👻	Modbus_TCP_top_sb_MSS
2		Ø,	-	SmartFusion2_MSS_Ethernet_MAC_Driver_0	SmartFusion2_MSS_Ethernet_MAC_Driver	3.1.100 👻	Modbus_TCP_top_sb_MSS:MAC
3	V			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	V			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	V		•	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	V		2 -1	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.
- 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, see the Appendix: Jumper Locations, page 19.

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



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

Table 6 • SmartFusion2 Advanced Development Kit Jumper Settings

- 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 energy patters to the J24 connector.
 - 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:

- 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.



F FlashPro	
File Edit Yiew Tools Programmers Configuration Customize Help D Image: Image	
New Project Configure Device Open Project View Programmers	RUN
New Project Project Name: Modbus Project Location: C: \Users \swapna.onteddhu \Deskt: Browse Programming mode ③ Single device ○ Chain Help OK Cancel	semi .
Info	
Ready	No project loaded

Figure 9 • FlashPro New Project

- Click Browse and navigate to the location where you want to save the project.
 Select Single device as the Programming mode.
 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

FP	FlashPro - [Modbus] *	
Fil	e Edit View Tools Programmers Configuration Customize Help	
З, г		ath ath Hath ath
	」 ┝╾	
	New Project	Configure Device PROGRAM View Programmers PROGRAM
X	Programming file Modbus TCP too sto Brows	
ð	Modify	/
Single Device Configuration Winc	DEVICE M2S150T PACKAGE M2S150T-fc1152 DATE 2015/09/28 STAPL_VERSION JSD71 IDCODE 0F8061CF IDMASK 0FFFFFF DESIGN Modbus_TCP_top CHECKSUM 6E0C SECURITY Disable	Mode: Basic O Advanced Action PROGRAM Procedures
× •	▲ ► All Errors Warnings Info /	

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.



FP	Flas	shPro - [M	odbus] *									
E	ile	<u>E</u> dit <u>V</u> iev	v <u>T</u> ools <u>P</u> rog	rammers	<u>Configuration</u>	n Custo <u>m</u>	<u>ize H</u> elp					
#	D 🛛	2 🖬 🛉	?	a a		🚿 🔝	🕺 🦚	🚸 🦚 🛛 🛤 🛛 🔊				
					New Project	*		Configure Device		PROGRAM		
×										1	D	Programmer
11					Program	ner			Programmer	Port	Programmer	riogrammer
ľ	1	13Y30KNV			Program Name				Programmer Type FlashPro5	Port usb13Y30KNV (US	Status RUN PASSED	Enabled
Programmer List Window	1	13Y30KNV			Program Name		[Refresh/Rescan for f	Programmers	Port usb13Y30KNV (US	RUN PASSED	Frogrammed Babled ₩
 IX Programmer List Window 		13Y30KNV	mmer '13Y3	DKNV' :	Program Name	ner :: :: ::::::::::::::::::::::::::::::	= 2ee0aa	Refresh/Rescan for	Programmer Type RashPro5	Port usb13Y30KNV (US	RUN PASSED	Finabled V
Programmer List Window		progra progra progra progra	mmer '13Y3 mmer '13Y3 mmer '13Y3 mmer '13Y3	DKMA, : DKMA, : DKMA, :	EXPORT DS EXPORT DS Finished: Executing 0 - 0 - 0	N [128]	= 2ee0aa n 29 14: PROGRAM o - o	Refresh/Rescan for f 010000cd5a00f 09:57 2015 (E 1 PASSED.	Programmer Type RashPro5	Port usb13Y30KNV (US):03:47)	RUN PASSED	
< Frogrammer List Window		progra progra progra	nmer '13Y3 nmer '13Y3 nmer '13Y3 nmer '13Y3	DKNV' : DKNV' : DKNV' :	EXPORT DS Finished: Executing 0 - 0 - 0	N[128] ; Thu Ja ; Thu Ja ; action	= 2ee0aa n 29 14: PROGRAM o - 0	Refresh/Rescan for f 010000cd5a00f 09:57 2015 (E 1 PASSED.	Programmer Type RashPro5	Port usb13Y30KNV (US	RUN PASSED	

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.

Fiaure	12•	Hyper]	[erminal	with	IP	Address
iguic	12	пурсп	crimina	WILII	••	Addicoo

9	Modbus - HyperTerminal	
E	je Edit Vjew ⊊all Iransfer Help	
۵) 📽 📨 🐉 🗅 🎦 🖬	
Γ	***************************************	<u> </u>
	## 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	
C	onnected 0:03:21 Auto detect 57600 8-N-1 SCROLL CAPS NUM Capture Print echo	

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

C:\WINDOWS\system32\cmd.exe	- 🗆 🗙
D:\SF2_Modbus_TCP_Ref_Design_DF\HostTool>SmartFusion2_Modbus_TCP_C G	lien 🔺
usage: SmartFusion2_Modbus_TCP_Client.exe	
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	

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

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	-

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

 C:\WINDOWS\system32\cmd.exe
 _ _ _ X

 MB_TCP:03:Read Holding Reg:Address at 0x0 = data is 0xed6
 _ _ _ X

 MB_TCP:03:Read Holding Reg:Address at 0x1 = data is 0x402
 _ _ _ _ X

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

 C:\WINDOWS\system32\cmd.exe
 _ □ ×

 MB_TCP:15:Write Multiple Coils: address Ø, data Øx?
 _



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

Local Area Connection Properties
Networking
Connect using:
😰 Broadcom NetLink (TM) Gigabit Ethemet
Configure
This connection uses the following items:
✓ Client for Microsoft Networks ✓ QoS Packet Scheduler ✓ Image: File and Printer Sharing for Microsoft Networks ✓ Image: File and Printer Sharing for Microsoft Networks ✓ Image: File and Printer Sharing for Microsoft Networks ✓ Image: File and Printer Sharing for Microsoft Networks ✓ Image: File and Printer Sharing for Microsoft Networks ✓ Image: File and Printer Sharing for Microsoft Networks ✓ Image: File and Printer Sharing for Microsoft Networks ✓ Image: File and Printer Sharing for Microsoft Networks ✓ Image: File and Printer Sharing for Microsoft Networks ✓ Image: File and Printer Sharing for Microsoft Networks ✓ Image: File and Printer Sharing for Microsoft Networks ✓ Image: File and Printer Sharing for Microsoft Networks ✓ Image: File and Printer Sharing for Microsoft Networks ✓ Image: File and Printer Sharing for Microsoft Networks ✓ Image: File and Printer Sharing for Microsoft Networks ✓ Image: File and Printer Sharing for Microsoft Networks ✓ Image: File and Printer Sharing for Microsoft Networks ✓ Image: File and Printer Sharing for Micros
Description Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks. OK Cancel



Figure 24 • Static IP Address Settings

Internet Protocol Version 4 (TCP/IPv4)	Properties 8 X				
General					
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 automatical	ly				
• Use the following IP address:					
IP address:	169 . 254 . 1 . 22				
Subnet mask:	255 . 255 . 255 . 0				
Default gateway:	· · ·				
Obtain DNS server address autor	natically				
• Use the following DNS server add	resses:				
Preferred DNS server:	169.254.1.23				
Alternate DNS server:	· · ·				
Validate settings upon exit	Ad <u>v</u> anced				
<u> </u>	OK Cancel				

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