Simple Socket Server Design on Nios® V/m Processor

Agilex™ 7 FPGA

Last updated: **2024-09-30**

Contents

Send Feedback

1.0 Theory of Operation

The telnet client offers a convenient way of issuing commands over a TCP/IP socket to the Ethernet-connected µC/TCP-IP running on the development board with a simple TCP/IP socket server example. The socket server example receives commands sent over a TCP/IP connection and turns LEDs on and off according to the commands. The example consists of a socket server task that listens for commands on a TCP/IP port and dispatches those commands to a set of LED management tasks.

Figure 1-1. Block Diagram
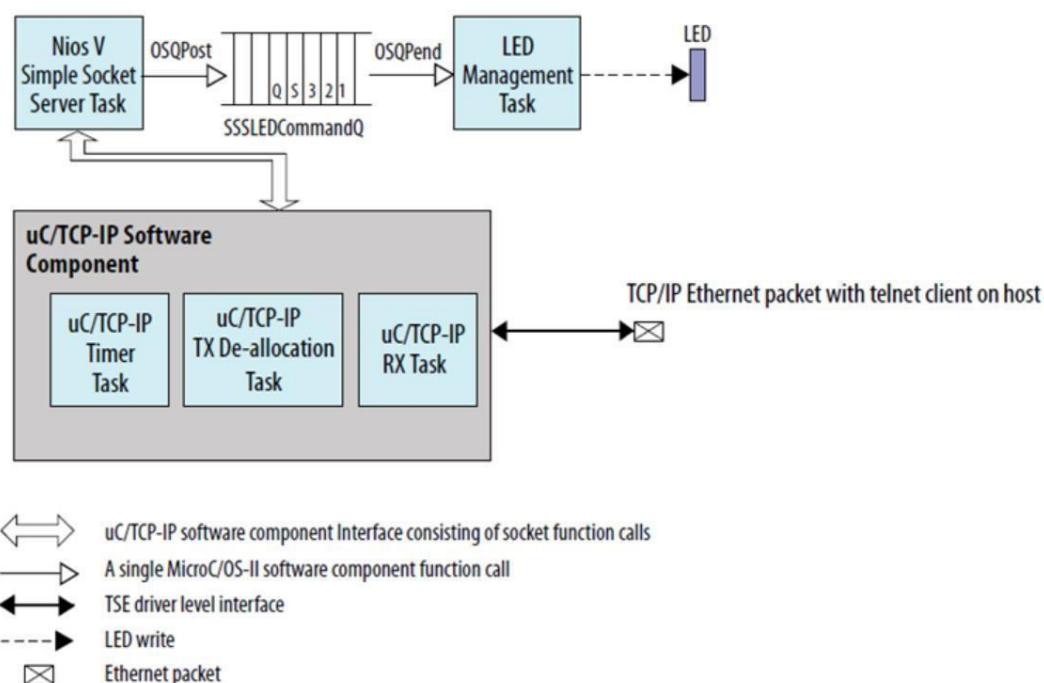
1.1 IP Cores

The following IPs are used in this design.

- Nios V/m soft processor core

- On Chip RAM-II

- JTAG UART

- Triple Speed Ethernet (TSE)

- System ID

- Altera MSGDMA

2.0 Executing the design on Development Kit

2.1 Creating and validating the design

Note: Please refer to the readme.txt file in the package for the steps to create the design, application and generate the programing files.

- Unpackage/extract the design in your working directory.
  Locate the "ready_to_test" folder within the package.
- The folder contains the necessary files for executing the application on the board.
  Refer to the readme file for the steps to program the application files on the board.
- Validate the design by observing the prints on the terminal.

Send Feedback

3.0   Expected Results

The following is the output as observed on the JTAG UART terminal. The output is analogous to the logic from the application code. Users should be able to observe same output on their terminal/setup.

Telnet Server:

```
[crt0.S] Setting up stack and global pointers.
[crt0.S] Clearing BSS
[crt0.S] Calling alt_main.
[alt_main.c] Entering alt_main, calling alt_irq_init.
[alt_main.c] Done alt_irq_init, calling alt_os_init.
[alt_main.c] Done OS Init, calling alt_sem_create.
[alt_main.c] Calling alt_sys_init.
[alt_main.c] Done alt_sys_init.
[alt_main.c] Redirecting IO.
[alt_main.c] Calling C++ constructors.
[alt_main.c] Calling atexit.
[alt_main.c] Calling main.
[main] Main Task TOS: 0x4f808
Print the value of System ID
System ID from Peripheral core is 0xFACECAFE
[uc_main_task]
[uc_main_task] =============================================================
[uc_main_task]                         uC/TCP-IP Setup
[uc_main_task] =============================================================
[uc_main_task] TSE MAC base: 0x212000.
[uc_main_task] Rx csr name: /dev/sys_tse_msgdma_rx_csr.
[uc_main_task] Tx csr name: /dev/sys_tse_msgdma_tx_csr.
[uc_main_task] INFO: Initializing network stack.
[conf_static] Configuring (static) IP address
[conf_static]  * Address: 192.168.1.100
[conf_static]  * Mask:    255.255.255.0
[conf_static]  * Gateway: 192.168.1.254
[uc_main_task] INFO: Initializing network stack: Success. Using interface 1.
[sss_task] Simple Socket Server listening on port 80
[sss_handle_accept] accepted connection request from 192.168.1.50
[sss_handle_receive] processing RX data
Value for LED_PIO_BASE set to 8.
```

Telnet Client:

```
Trying 10.233.123.6...
Connected to sj-oscar-bb.zsc7.intel.com.
Escape character is '^]'.
=================================
Nios V/m Simple Socket Server Menu
=================================
0-3: Toggle board LEDs D0 - D3
S: Board LED Light Show
Q: Terminate session
=================================
Enter your choice & press return:
3
--> Simple Socket Server Command 3.
```

4.0 Document Revision History

Date	Version	Changes
2024-09-30	24.3.0	• Initial release on github

Simple_Socket_Server_design_on_Nios®V_m_Processor: Agilex™ 7 FPGA

💬 **Send Feedback**