

Nios[®] V Processor – Simple Socket Server Design for Nios[®] V/m Processor

Date: 06/30/2023

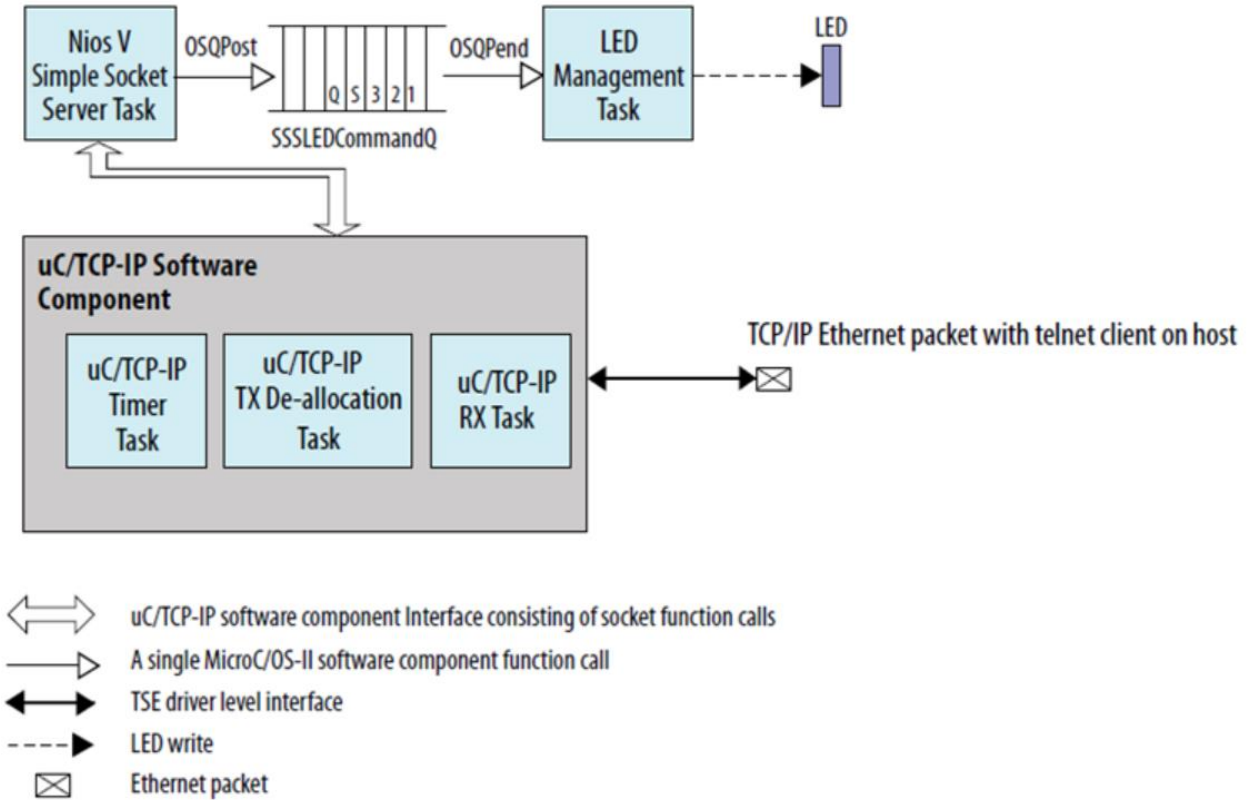
Revision: 1.0

Table of Contents

Theory of Operation	3
1. Block Diagram	3
2. Description	3
3. IP Cores used:	3
Executing the Design on dev kit	4
1. Creating the Design	4
2. Expected Result	4

Theory of Operations

1. Block Diagram



2. DESCRIPTION

The telnet client offers a convenient way of issuing commands over a TCP/IP socket to the Ethernet-connected μ C/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.

3. IP Cores used:

I/O PLL

NIOSV/m soft processor core

Parallel I/O

On Chip RAM

Modular Scatter Gather DMA IP

Triple Speed Ethernet IP

Transceiver ATX PLL IP

Executing the Design on dev kit

1. Creating the Design

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

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

2. Expected Result

The following is the output as observed on the JTAG UART terminal and telnet. The output is analogous to the logic from the application code.

The user should observe same output on their terminal/setup

```
[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: 0x4f750  
[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: 192.168.130.5  
[conf_static] * Mask:    255.255.255.0  
[conf_static] * Gateway: 192.168.130.254  
[uc_main_task] INFO: Initializing network stack: Success. Using interface 1.  
[sss_task] Simple Socket Server listening on port 80
```

telnet client

```
=====
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:
Connection closed by foreign host.

0-3: Toggle board LEDs D0 - D3
```