

Nios II Simple Socket Server on C10 LP Development Kit

Date: Jun 2018

Revision: 1.0

©2016 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

Contents

Overview	3
Tool Requirements.....	3
Hardware Design Specifications.....	3
Instruction for running the software	3
Hardware regeneration	6
Software regeneration.....	6
Selection of Static IP or DHCP client	7
Revision History	8

Overview

This design example shows a socket server using the NicheStack TCP/IP stack-Nios® II Edition on MicroC/OS-II on a Cyclone 10 LP development board. The server implements simple commands to control board LEDs through a separate MicroC/OS-II task.

This reference project is bundled with [Synaptic Laboratories Ltd.](#) (SLL) [License Agreement](#) and Free Trial License for the full edition of SLL's [HyperBus Memory Controller \(HBMC\) IP](#). This Free Trial License never expires and can be installed on all your computers without contacting or registering with SLL. The latest HBMC IP and enhancements [can be found here](#).

Tool Requirements

- Altera Cyclone 10 LP Development Kit
- Quartus II software version 17.1
- Nios II Embedded Design Suite (EDS) 17.1
- Mini USB cable for programming the device

Hardware Design Specifications

Major component for hardware design:

- Nios II/f
- Onchip memory
- System timer
- System ID
- JTAG UART
- PIO
- [Synaptic Laboratories Ltd's](#) (SLL) [HyperBus Memory Controller IP](#)
- Modular Scatter Gather DMA
- Triple-Speed Ethernet
- Serial flash controller
- Remote update

Instruction for running the software

Before running the software, connect the USB blaster cable from PC to the on-board connector which shown in Figure 1.

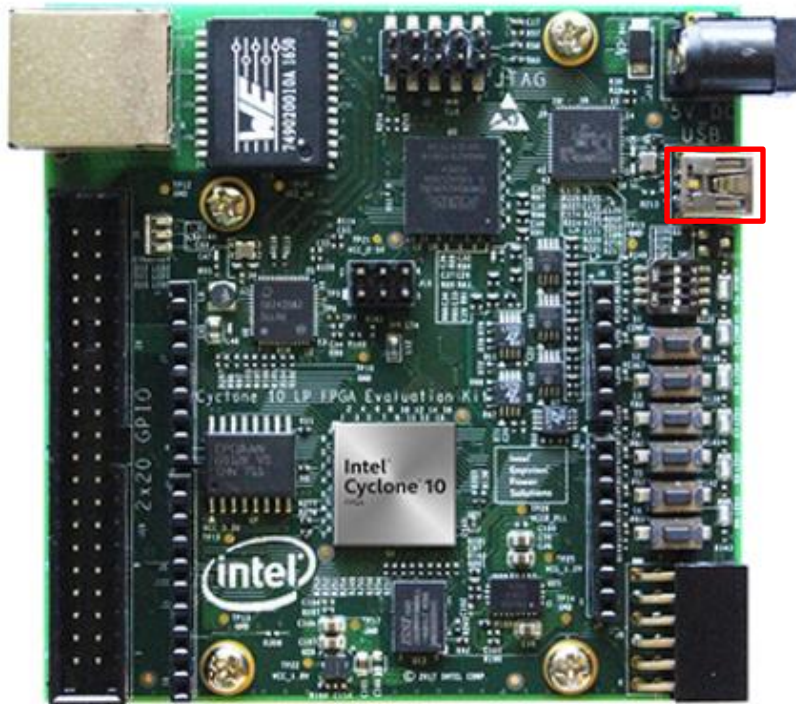


Figure 1: USB Blaster II connection location

1. Open the Quartus Programmer to program the generated .sof file to the FPGA . Master image folder provided in the project directory contain the .sof and .elf file which is ready to be downloaded to the board

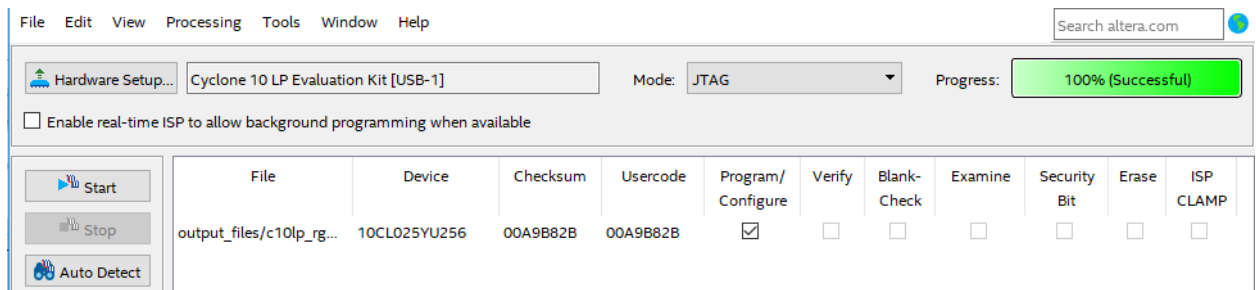


Figure 2: Quartus Prime Programmer

2. In this example, static IP is used. Hence, please make sure that you have configured your PC to have a static IP address before connected to the development kit.

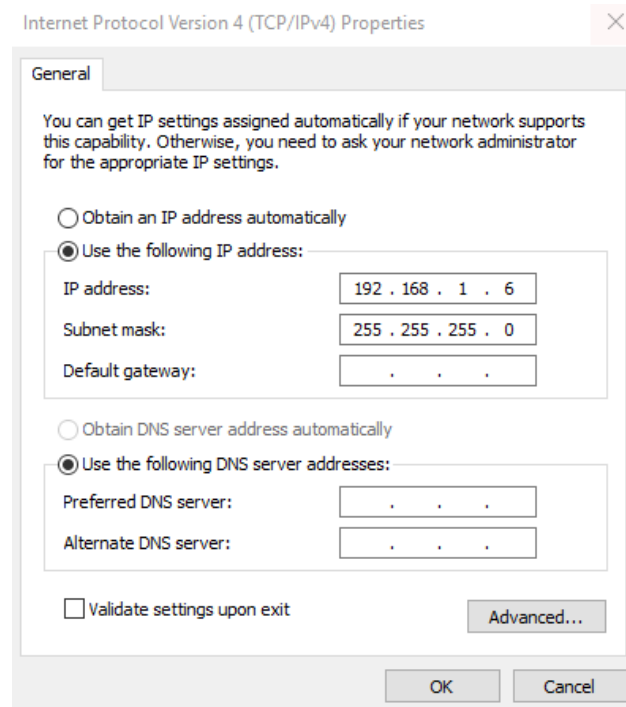


Figure 3: Setting static IP address on PC

3. Open Nios II command shell to download the .elf file and invoke the terminal by typing ***nios2-download -g sss_c10lp.elf && nios2-terminal***

```

InterNiche Portable TCP/IP, v3.1

Copyright 1996-2008 by InterNiche Technologies. All rights reserved.
prep_tse_mac 0

Can't read the MAC address from your board. We will assign you
a MAC address.

Please enter your 9-digit serial number. This is printed on a
label under your Nios dev. board. The first 3 digits of the
label are ASJ and the serial number follows this.
-->Created "Inet main" task (Prio: 2)
Created "clock tick" task (Prio: 3)

```

Figure 4: Start up messages

4. You will be asked for the MAC address. You can simply key in any 9-digit numbers. For example: 123456789.
5. Once the simple socket server is started up successfully. You can open command prompt on your window PC and connect to the simple socket server.
telnet 192.168.1.234 30

```

C:\ Telnet 192.168.1.234
=====
Nios II Simple Socket Server Menu
=====
0-4: Toggle board LEDs D0 - D4
S: LED Light Show
Q: Terminate session
=====
Enter your choice & press return:

```

Figure 5: Messages displayed once successfully connected from command prompt

6. You can control the LED with by enter the choice from 0-4 for controlling 4 LED on the board

Hardware regeneration

You must install the Free License Key included in this reference design's installation package for [SLL's HyperBus Memory Controller IP](#) to enable regeneration and compilation of this project. This Free Trial License never expires and can be installed on all your computers without contacting or registering with SLL. For more information on [SLL's License Agreement](#) and where to access the Free License Key, please read:

c10lp_simple_socket_server_project/ip/..._SLL_HBMC_.../1-ReadMeFirst.txt

Also, the latest version of the Free Trial HyperBus Memory Controller IP and enhancements [can be found here](#).

After you install the Free Trial License Key into Quartus, you can then open the Qsys system with platform designer and add/remove and modify the individual IP components within the design. After you have finished modifying the Qsys system, you can then regenerate the system by clicking on "Generate HDL...".

Software regeneration

The software used in this design is based on the existing Simple Socket Server template available in Nios II Software Build Tools. Minor modification is done on some of the .c and .h files to prevent error due to naming, number of LEDs control and using static IP address.

File	Modification
led.c	Update led driver to map to C10LP Development Kit hardware
simple_socket_server.c	Update text printing on terminal
simple_socket_server.h	Enable Static IP address

After you generate the Simple Socket Server template from Nios II Eclipse, you can replace the default .c and .h files with the files provided with this design example located in <project directory>/software/src folder. Please take note that the naming for the triple speed ethernet IP is correct in tse_my_system.c file. Incorrect naming will cause software compilation error.

The figure below shows how to change the IP address assignment in the file “simple_socket_server.h”

```

simple_socket_server.h
104 * Default IP addresses are set to all zeros so that DHCP server packets will
105 * penetrate secure routers. They are NOT intended to be valid static IPs,
106 * these values are only a valid default on networks with DHCP server.
107 *
108 * If DHCP will not be used, select valid static IP addresses here, for example:
109 *     IP: 192.168.1.234 (default 0.0.0.0)
110 *     Gateway: 192.168.1.1 (default 0.0.0.0)
111 *     Subnet Mask: 255.255.255.0
112 */
113 #define IPADDR0 192
114 #define IPADDR1 168
115 #define IPADDR2 1
116 #define IPADDR3 234
117
118 #define GWADDR0 192
119 #define GWADDR1 168
120 #define GWADDR2 1
121 #define GWADDR3 1
122
123 #define MSKADDR0 255
124 #define MSKADDR1 255
125 #define MSKADDR2 255
126 #define MSKADDR3 0

```

Figure 6: Setting Static IP address

Selection of Static IP or DHCP client

Static IP address is used in this design. Hence, take note that you will need to disable the dhcp client in BSP editor if you rebuild the software as shown below.

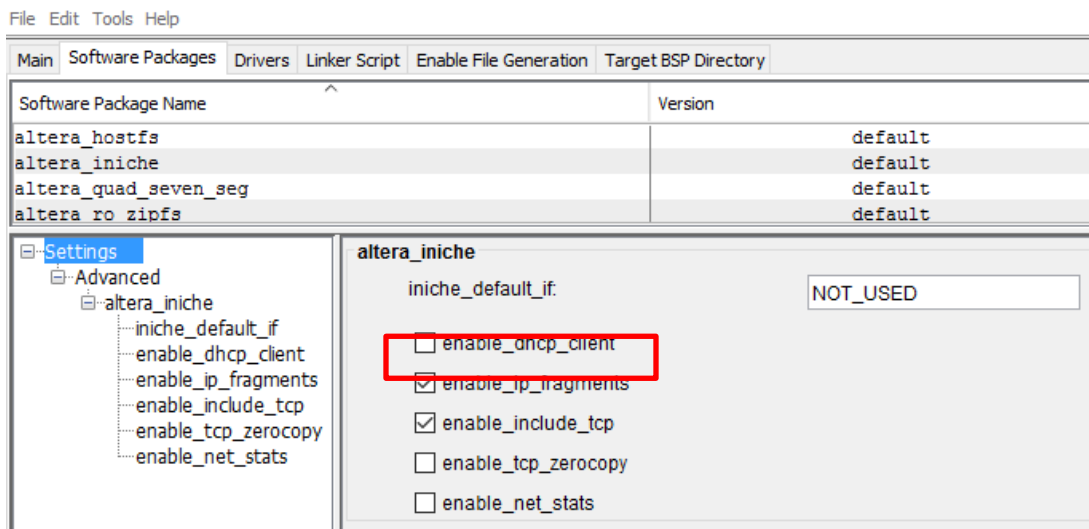


Figure 7: Disable DHCP client

Revision History

Revision	Description
1.0	Simple Socket Server design for C10 LP