Nios II Ethernet Simple Socket Server on MAX10 FPGA Development Kit

Table of Contents

Overview 3

Theory of Operation.. 3

Simple Demo Setup.. 4

How to compile the hardware 7

How to compile the software 8

Overview

The following external parts are needed to demonstrate the design example;

- MAX10 10M50 FPGA Development kit
- Mini-USB cable for programming the device
- Ethernet cable
- **IMPORTANT**: only use the 12V, 2A AC adapter that came with this kit. Do not use other power supplies from other Altera kits, these have higher voltage and may blow out the kit's power circuits

Theory of Operation

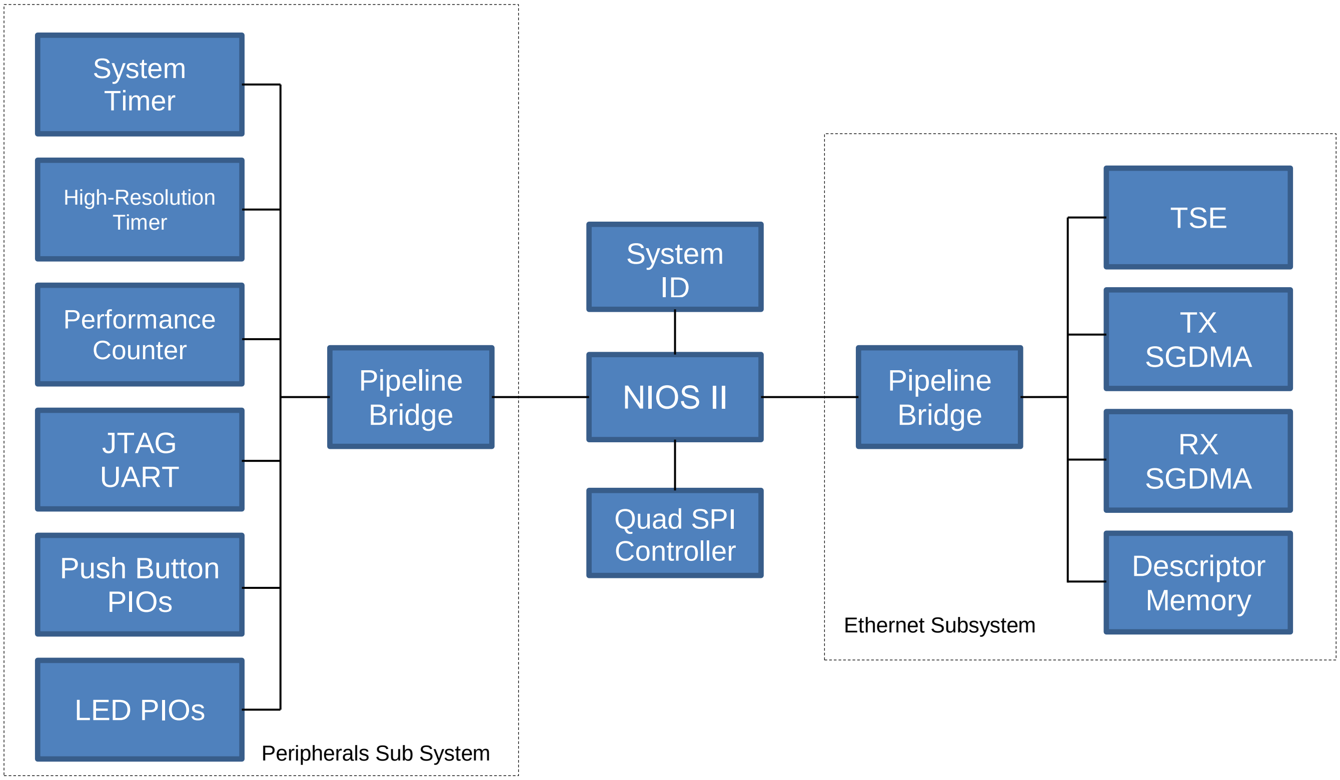


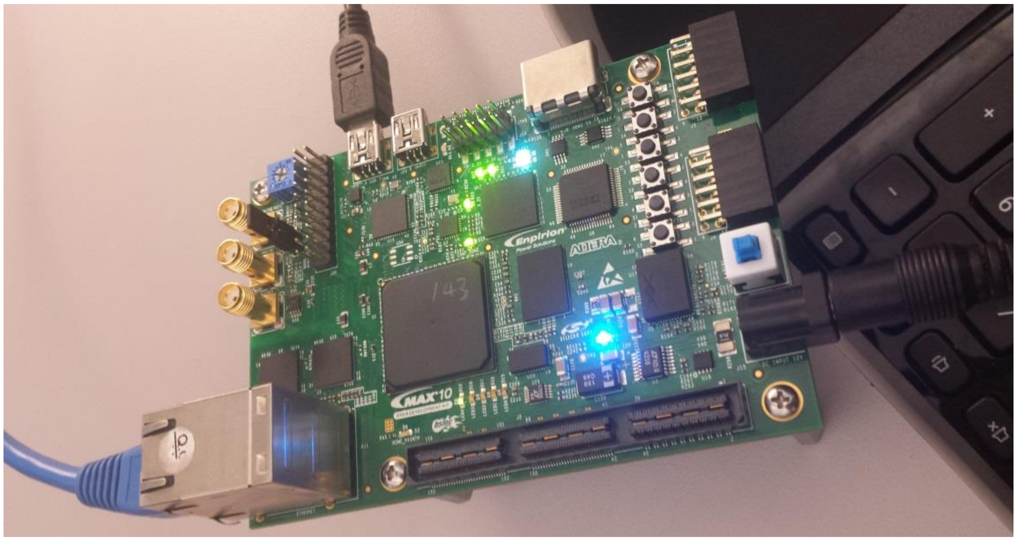
Figure 1: Ethernet Reference Design Block Diagram

The Simple Socket Server design uses a modified version of Altera's Nios® II Ethernet Standard hardware design. It provides a mix of peripherals and memories similar to a typical Nios II processor system. This design interfaces with each hardware component on the Altera® development board, such as SDRAM, LEDs, push buttons, and an Ethernet physical interface or media access control (PHY/MAC). This Qsys-based hierarchical design has a top-level system and two subsystems, namely: peripherals subsystem and Ethernet subsystem as shown in Figure 1.

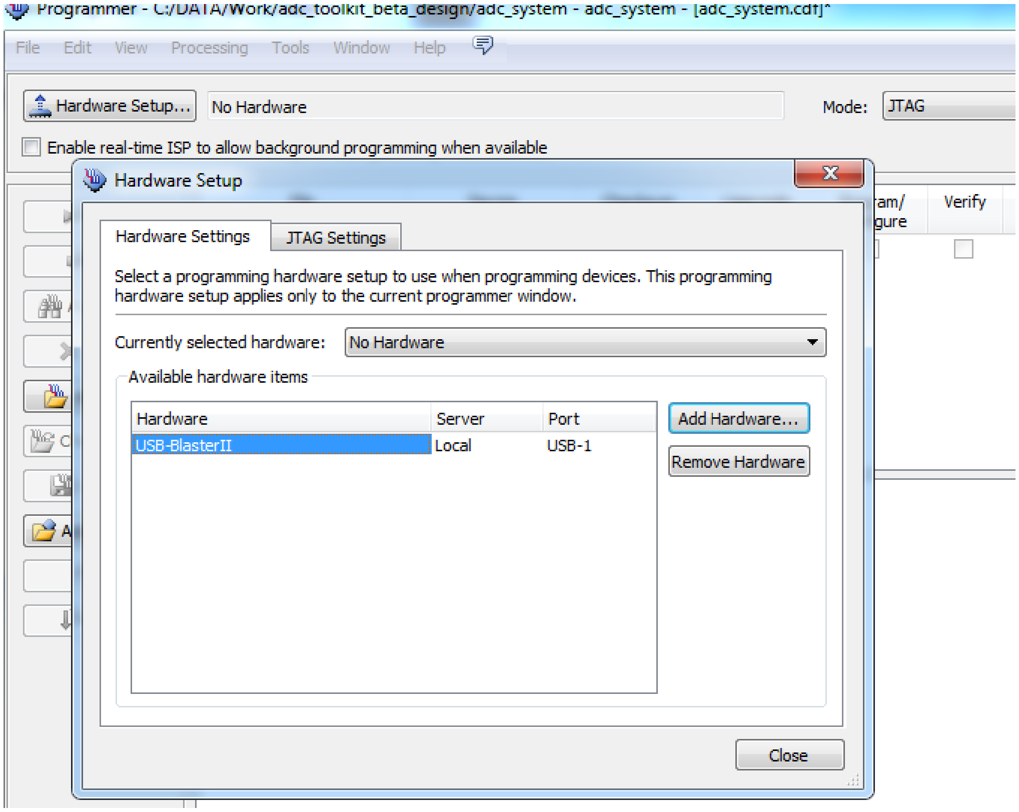
For more details on the Nios II Simple Socket Server implementation, please refer to page 1-16 on the following documentation.

Simple Demo Setup

1. Connect an Ethernet cable to the bottom socket as shown in the pictures.
2. Connect the power cord to the power plug, and connect a mini USB from your PC/laptop to the J12 USB connector (labeled as USB 1 on the silkscreen) on the top left of the kit. Connect the Ethernet cable to the bottom port (Port B). Your set up should look like below:



3. The design package includes a pre-compiled version of the design that downloads to the MAX 10 embedded flash. This version of the configuration includes the combined FPGA configuration and NIOS executable. Follow these instructions to load the design onto your kit:
    a. Make sure the board is powered on (push the blue power button) and that the USB is connected on J12
    b. Startup Quartus. Then start the programmer from Tools → Programmer.
    c. Click Hardware Setup → USBBlaster II and select Add Hardware



Programmer - C:/DATA/Work/adc_toolkit_beta_design/adc_system - adc_system - [adc_system.cdf]*

No Hardware

Mode: JTAG

□ Enable real-time ISP to allow background programming when available

Hardware Setup

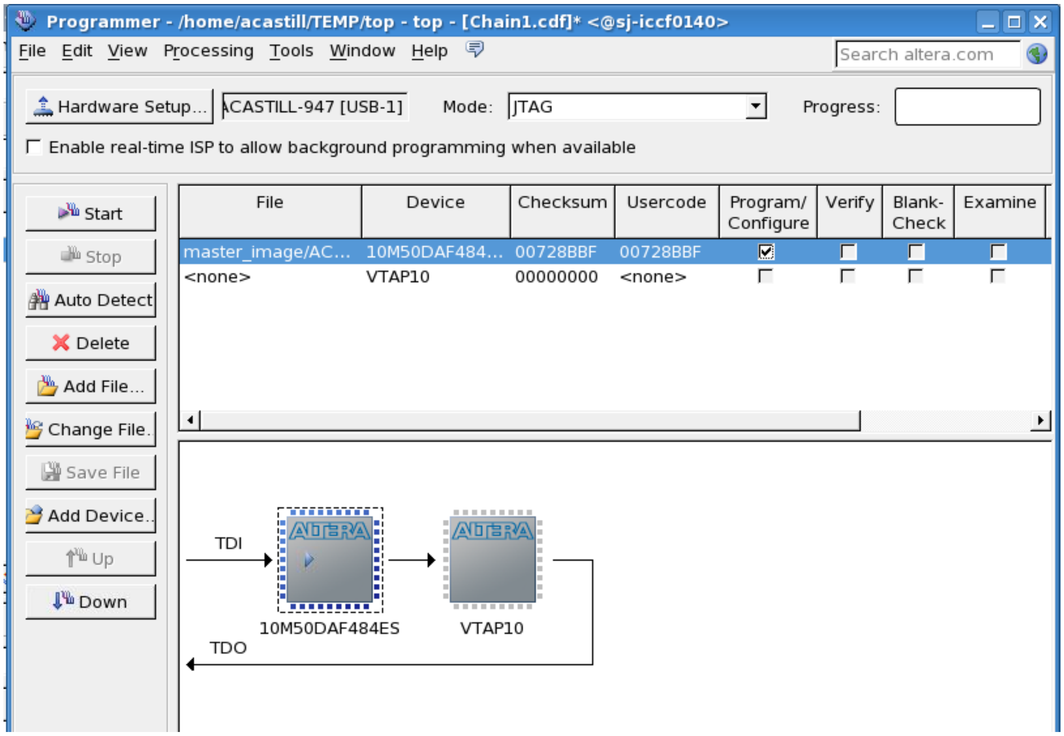
Select a programming hardware setup to use when programming devices. This programming hardware setup applies only to the current programmer window.

Currently selected hardware: No Hardware

Available hardware items

Hardware	Server	Port
USB-BlasterII	Local	USB-1

d. Click “Auto Detect” and double click where it says <none> on the first row to add your .sof/.pof file from the master_image folder. Note: A virtual TAP controller (VTAP) will always show in the JTAG chain – this is correct behavior.



Programmer - /home/acastill/TEMP/top - top - [Chain1.cdf]* <@sj-iccf0140>

ACASTILL-947 [USB-1]

Mode: JTAG

□ Enable real-time ISP to allow background programming when available

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank-Check	Examine
master_image/AC...	10M50DAF484...	00728BBF	00728BBF	☑	□	□	□
<none>	VTAP10	00000000	<none>	□	□	□	□

e. Fill in the checkbox for "Program" and select "Start" to begin downloading the design to the board – your dev kit will start running the design when the process is successful.

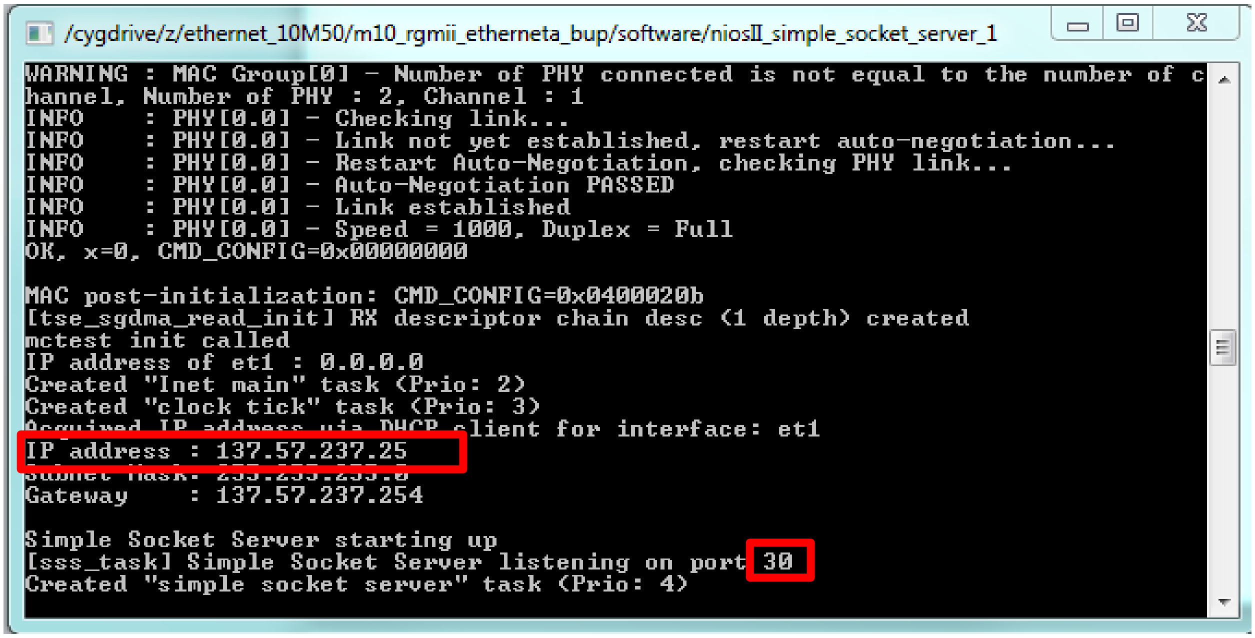
f. You should see LED4 blinking – this indicates that the board is reading to be remotely controlled.

4. Open the Nios II terminal by going to Start → Altera → Nios II EDS → Nios II Command Shell
5. Enter the following command:

   nios2-download –g <file path >/master_image/niosII_simple_socket_server.elf && nios2-terminal

6. If you get a message prompting you to enter a 9 digit number. Enter a potentially correct MAC address like 255255248. This happens because some boards do not have a MAC address dedicated in the CFI Flash. The user has to enter the address if this is the case.

7. This will download the software onto the Nios II processor and run the software. The IP Address, Subnet Mash, and Gateway information will be set up and displayed. After it loads, the console should indicate that the Simple Socket Server is "listening on port <port number> as shown below.


/cygdrive/z/ethernet_10M50/m10_rgmii_etherneta_bup/software/niosII_simple_socket_server_1

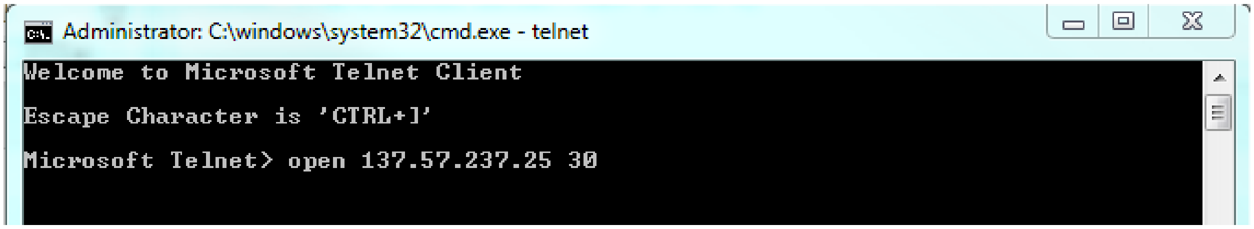
```
WARNING : MAC Group[0] - Number of PHY connected is not equal to the number of c
hannel, Number of PHY : 2, Channel : 1
INFO    : PHY[0.0] - Checking link...
INFO    : PHY[0.0] - Link not yet established, restart auto-negotiation...
INFO    : PHY[0.0] - Restart Auto-Negotiation, checking PHY link...
INFO    : PHY[0.0] - Auto-Negotiation PASSED
INFO    : PHY[0.0] - Link established
INFO    : PHY[0.0] - Speed = 1000, Duplex = Full
OK, x=0, CMD_CONFIG=0x00000000

MAC post-initialization: CMD_CONFIG=0x0400020b
[tse_sgdma_read_init] RX descriptor chain desc (1 depth) created
mctest init called
IP address of et1 : 0.0.0.0
Created "Inet main" task (Prio: 2)
Created "clock tick" task (Prio: 3)
Acquired IP address via DHCP client for interface: et1
IP address : 137.57.237.25
Subnet Mask: 255.255.255.0
Gateway     : 137.57.237.254

Simple Socket Server starting up
[sss_task] Simple Socket Server listening on port 30
Created "simple socket server" task (Prio: 4)
```

8. Open up telnet client in the computer's command window by typing "telnet" and hitting enter.
9. Once you've entered the telnet client, open up the port by entering the command:

   open <IP address shown on the Nios terminal> <port number>

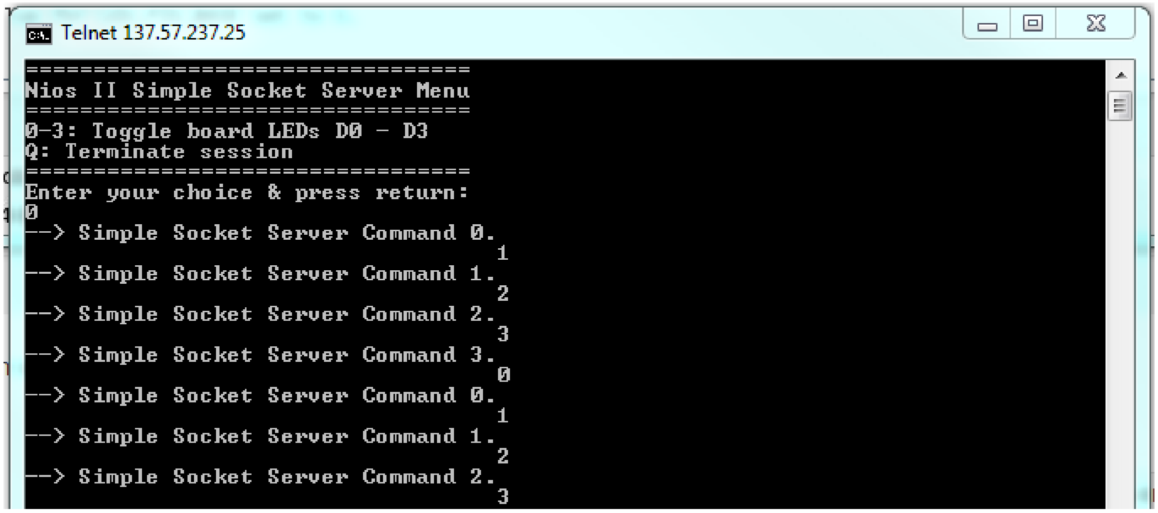


```
Welcome to Microsoft Telnet Client

Escape Character is 'CTRL+]'

Microsoft Telnet> open 137.57.237.25 30
```

10. Once successfully connected, your screen should show the menu for controlling the board remotely. Entering a number between 0-3 will toggle LEDs 0-3 respectively.



```
=================================
Nios II Simple Socket Server Menu
=================================
0-3: Toggle board LEDs D0 - D3
Q: Terminate session
=================================
Enter your choice & press return:
0
--> Simple Socket Server Command 0.
                                   1
--> Simple Socket Server Command 1.
                                   2
--> Simple Socket Server Command 2.
                                   3
--> Simple Socket Server Command 3.
                                   0
--> Simple Socket Server Command 0.
                                   1
--> Simple Socket Server Command 1.
                                   2
--> Simple Socket Server Command 2.
                                   3
```

How to compile the hardware

Follow the steps on the Design Store web page to extract and install the simple_socket_server_10M50 platform file. The following steps describe how to setup a project in Quartus II software in order to program the MAX10 FPGA device with the Simple Socket Server design.

Note – extract platform

i) Launch Quartus II software and open the project top.qpf using File->Open Project.

ii) Compile the Project by clicking the ▶ button.

iii) Launch the Quartus II programmer from the Tools menu or alternatively by clicking the button.

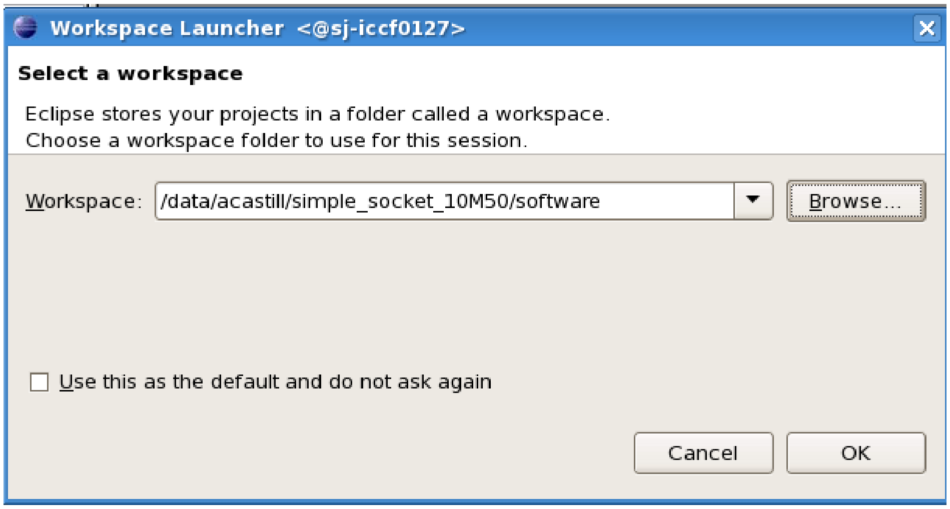
iv) Download the .sof file output_files/top.sof and program the device using the programmer as previously described.

How to compile the software

The following steps describe how to use Nios II Software Build Tools for Eclipse software to perform the following tasks;

i) Create a software project (BSP and application) from a .sopcinfo file
ii) Add source code to the project
iii) Download and run source code on the target processor (NIOS II)

1. Open Quartus II on a windows platform
2. Launch Nios II Software Build Tools for Eclipse from the Tools menu
3. Specify the workspace directory for the project and click ok.



Workspace Launcher <@sj-iccf0127>

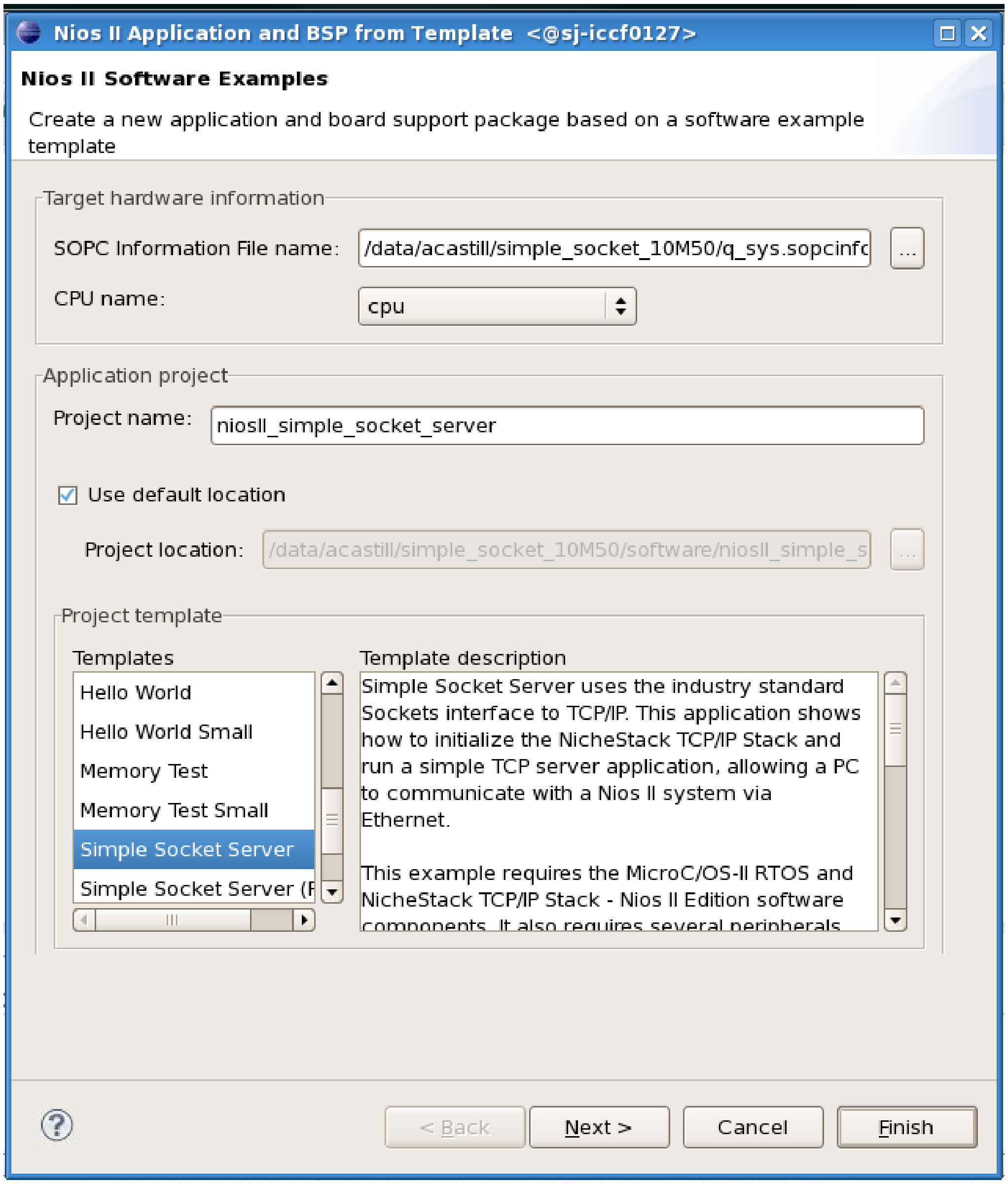
Select a workspace

Eclipse stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: /data/acastill/simple_socket_10M50/software

□ Use this as the default and do not ask again

4. A blank workspace window pops up, right click anywhere in the Project Explorer and select the following to create a new software project; New->Nios II Application and BSP (board support package) from Template.
5. In the window that appears, browse to the location of the .sopcinfo file in the project folder and add it to the project. Provide a name for the software project i.e. niosII_simple_socket_server and select “Simple Socket Server” for your template. Click Finish.


Nios II Application and BSP from Template <@sj-iccf0127>

Nios II Software Examples

Create a new application and board support package based on a software example template

Target hardware information

SOPC Information File name: /data/acastill/simple_socket_10M50/q_sys.sopcinfc

CPU name: cpu

Application project

Project name: niosII_simple_socket_server

☑ Use default location

Project location: /data/acastill/simple_socket_10M50/software/niosII_simple_s

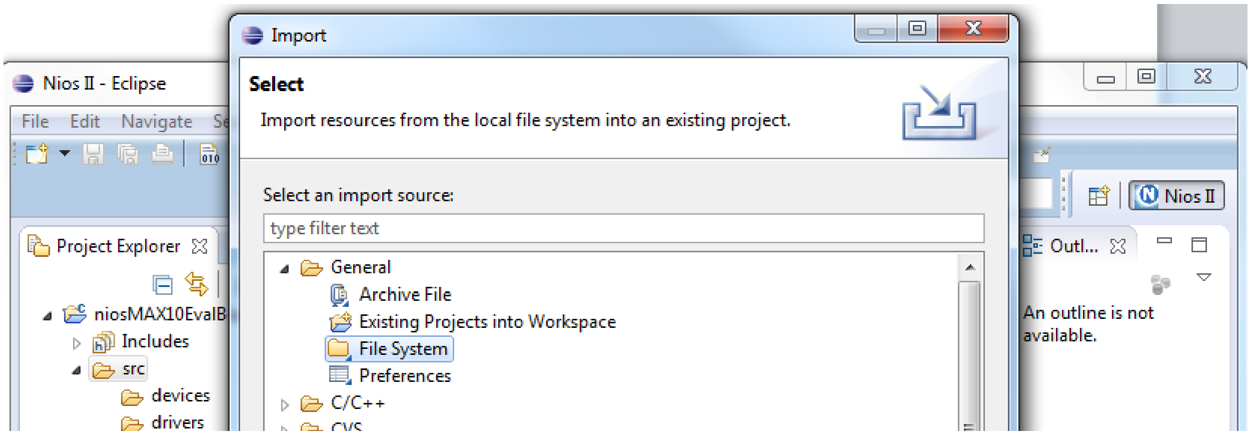
Project template

Templates: Hello World, Hello World Small, Memory Test, Memory Test Small, Simple Socket Server, Simple Socket Server (F

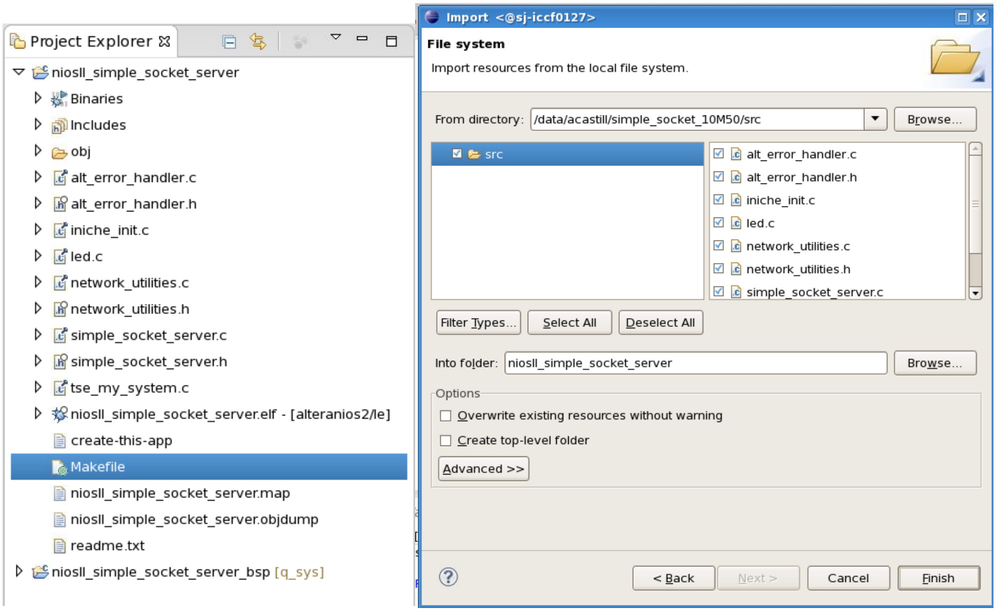
Template description: Simple Socket Server uses the industry standard Sockets interface to TCP/IP. This application shows how to initialize the NicheStack TCP/IP Stack and run a simple TCP server application, allowing a PC to communicate with a Nios II system via Ethernet.

This example requires the MicroC/OS-II RTOS and NicheStack TCP/IP Stack - Nios II Edition software components. It also requires several peripherals

6. A BSP is created and located at niosII_simple_socket_server and an application is created and located at niosII_simple_socket_server_bsp.
7. The application contains source code from the template design "Simple Socket Server" that needs to be replaced with the source code located in the software/src folder. Select all the .c and .h files under the main project explorer tab. Right click and delete.
8. Right click the project name and left click the import command. We will be importing source code that has been modified for this dev kit.
9. Navigate to General → File System
   a. Browse to the location of src, click on the box next to it, add it as shown and click Finish.



10. Import the contents of the src folder as shown below

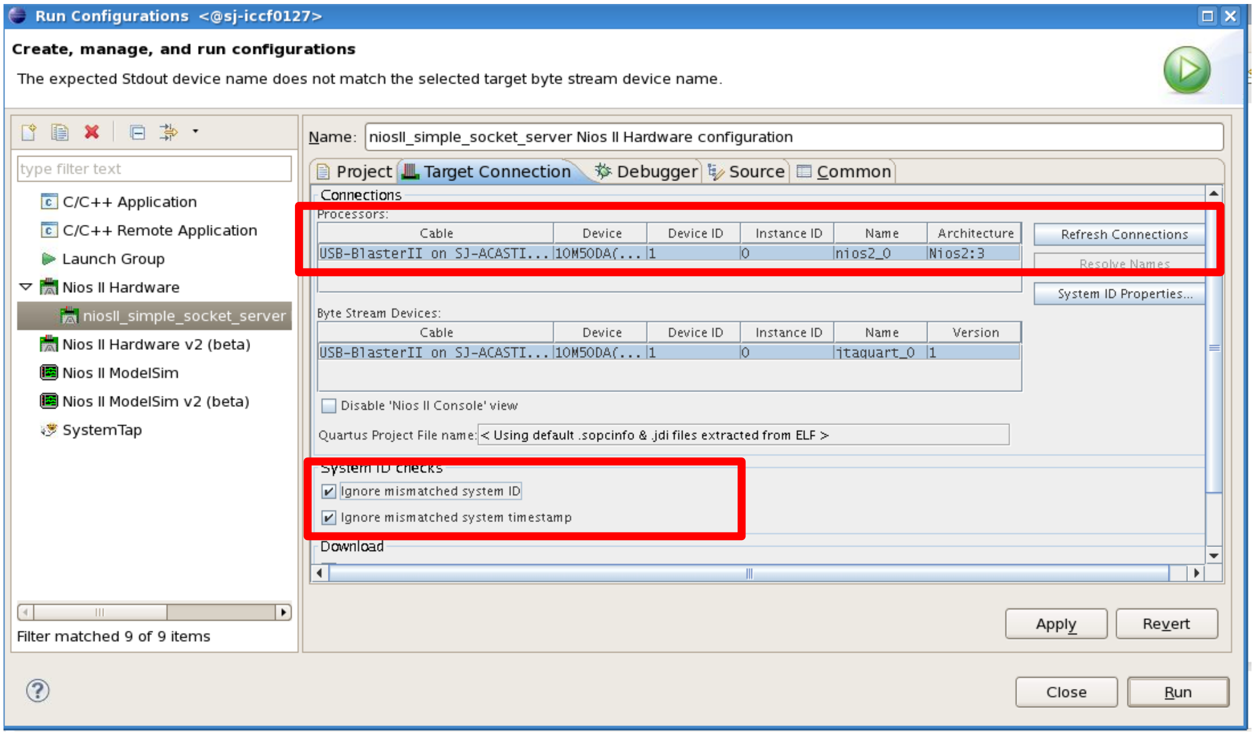


From directory: /data/acastill/simple_socket_10M50/src

Into folder: niosII_simple_socket_server

Your project directory on the left should look similar (mem_init and the .elf will be created later)

11. Compile the project by selecting Build All from the Projects Menu or alternatively Ctrl+B.
12. Program the FPGA device by launching Quartus II programmer from the menu and loading the file output_files/top.sof.
13. Load the executable to the processor by right clicking on the Application project and selecting Run-As Nios II Hardware.
14. In the Run Configurations page under the Target connections tab. Check the options; Ignore mismatched system ID and Ignore mismatched system timestamp. If you don't see the connection specified, click Refresh Connections. Click Apply followed by Run.



Run Configurations <@sj-iccf0127>

Create, manage, and run configurations

The expected Stdout device name does not match the selected target byte stream device name.

Name: niosII_simple_socket_server Nios II Hardware configuration

Processors:

Cable	Device	Device ID	Instance ID	Name	Architecture
USB-BlasterII on SJ-ACASTI...	10M50DA(...	1	0	nios2_0	Nios2:3

Byte Stream Devices:

Cable	Device	Device ID	Instance ID	Name	Version
USB-BlasterII on SJ-ACASTI...	10M50DA(...	1	0	jtaguart_0	1

☐ Disable 'Nios II Console' view

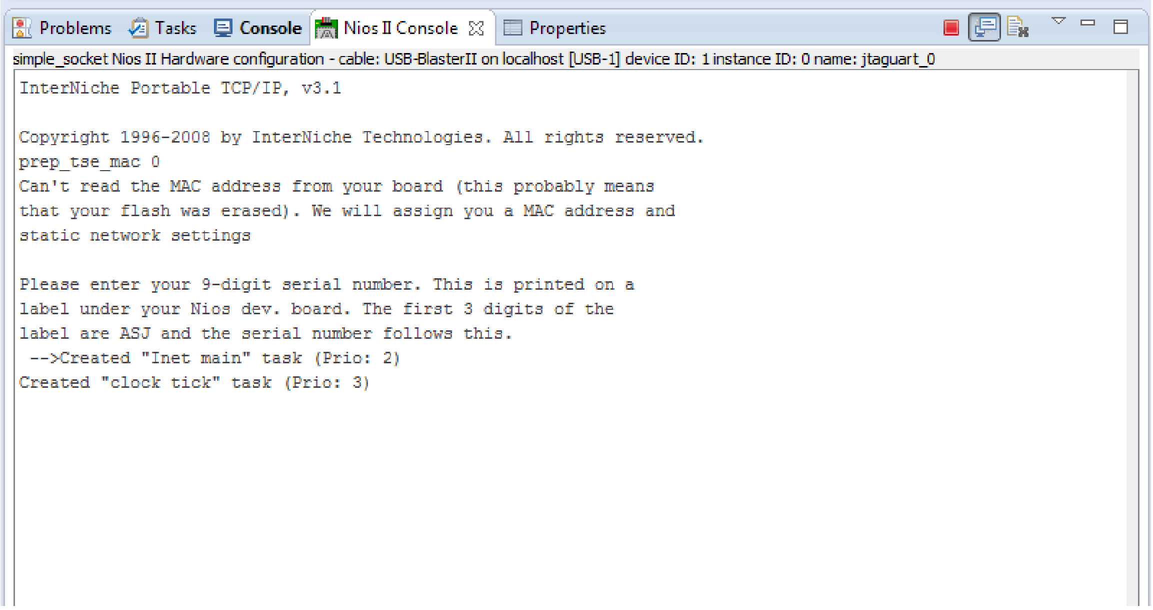
Quartus Project File name: < Using default .sopcinfo & .jdi files extracted from ELF >

System ID checks

☑ Ignore mismatched system ID

☑ Ignore mismatched system timestamp

15. If you get the following window, enter a 9 digit number MAC address that makes sense for the board. For example, if you type 123456789 it won’t work as that is not a potential MAC address. But if you type in 255255248, it will work as that can be a potential MAC address.



simple_socket Nios II Hardware configuration - cable: USB-BlasterII on localhost [USB-1] device ID: 1 instance ID: 0 name: jtaguart_0

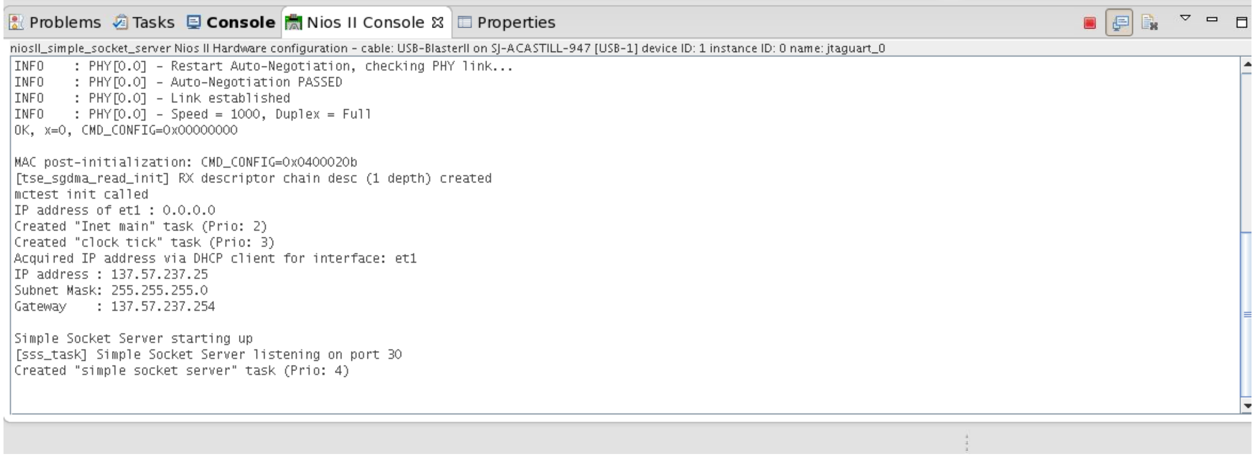
```
InterNiche Portable TCP/IP, v3.1

Copyright 1996-2008 by InterNiche Technologies. All rights reserved.
prep_tse_mac 0
Can't read the MAC address from your board (this probably means
that your flash was erased). We will assign you a MAC address and
static network settings

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)
```

This happens because on some boards there is no MAC address in the CFI flash dedicated section. In such cases, the software is designed to wait for the user to manually type in the MAC address.

16. Now your hardware image is downloaded to the FPGA through the Quartus programmer and the NIOS code has also been downloaded through Eclipse and will automatically start running. Refer to the Nios II console on Eclipse:



niosII_simple_socket_server Nios II Hardware configuration - cable: USB-BlasterII on SJ-ACASTILL-947 [USB-1] device ID: 1 instance ID: 0 name: jtaguart_0

```
INFO    : PHY[0.0] - Restart Auto-Negotiation, checking PHY link...
INFO    : PHY[0.0] - Auto-Negotiation PASSED
INFO    : PHY[0.0] - Link established
INFO    : PHY[0.0] - Speed = 1000, Duplex = Full
OK, x=0, CMD_CONFIG=0x00000000

MAC post-initialization: CMD_CONFIG=0x0400020b
[tse_sgdma_read_init] RX descriptor chain desc (1 depth) created
mctest init called
IP address of et1 : 0.0.0.0
Created "Inet main" task (Prio: 2)
Created "clock tick" task (Prio: 3)
Acquired IP address via DHCP client for interface: et1
IP address : 137.57.237.25
Subnet Mask: 255.255.255.0
Gateway    : 137.57.237.254

Simple Socket Server starting up
[sss_task] Simple Socket Server listening on port 30
Created "simple socket server" task (Prio: 4)
```

17. You can proceed to the demonstration as discussed in the earlier steps. The method described works well if you are modifying your code. You can maintain the same FPGA download and iterate through code revisions.