

# Using MicroC/OS-II RTOS with the

## **Nios II Processor Tutorial**



101 Innovation Drive San Jose, CA 95134 (408) 544-7000 http://www.altera.com

Copyright © 2007 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. 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. Alter 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 Corporation. 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



TU-NIOSII-MCRC/OS-II-1.2



## Contents

About this Tutorial	\	V
How to Find Information		v
How to Contact Altera		v
Typographic Conventions	v	i

Tutorial	1
Introduction	1–1
Hardware & Software Requirements	1–1
Tutorial Files	1–1
Hello MicroC/OS-II Design	1–2
Create a New Nios II IDE Project	1–3
Configure the System Library	1–7
Build & Run the Nios II Software Project	1–11



## About this Tutorial

This tutorial introduces you to the Nios<sup>®</sup> II integrated development environment (IDE) and MicroC/OS-II development flow. It shows you how to use the Nios II IDE to create a new MicroC/OS-II project to configure, build, and run a MicroC/OS-II program on the Nios development board.

Table 1–1 shows the tutorial revision history.

Table 1–1. Tutorial Revision History		
Date	Description	
January 2007	Updates for Nios II 6.1 release.	
September 2004	Updates for Nios II 1.01 release.	
May 2004	First publication.	

## How to Find Information

- The Adobe Acrobat Find feature allows you to search the contents of a PDF file. Click the binoculars toolbar icon to open the Find dialog box.
- Bookmarks serve as an additional table of contents.
- Thumbnail icons, which provide miniature previews of each page, provide a link to the pages.
- Numerous links, shown in green text, allow you to jump to related information.

## How to Contact Altera

For the most up-to-date information about Altera<sup>®</sup> products, go to the Altera world-wide web site at www.altera.com. For technical support on this product, go to www.altera.com/mysupport. For additional information about Altera products, consult the sources shown below.

Information Type	USA & Canada	All Other Locations	
Technical support	www.altera.com www.altera.com		
	(800) 800-EPLD (3753) (7:00 a.m. to 5:00 p.m. Pacific Time)	(408) 544-7000 (1) (7:00 a.m. to 5:00 p.m. Pacific Time)	
Product literature	www.altera.com	www.altera.com	
Altera literature services	literature@altera.com (1)	literature@altera.com (1)	

Information Type	USA & Canada	All Other Locations
Non-technical customer service	(800) 767-3753	(408) 544-7000 (7:30 a.m. to 5:30 p.m. Pacific Time)
FTP site	ftp.altera.com	ftp.altera.com

Note to table:

(1) You can also contact your local Altera sales office or sales representative.

### Typographic Conventions

This document uses the typographic conventions shown below.

Visual Cue	Meaning			
Bold Type with Initial Capital Letters	Command names, dialog box titles, checkbox options, and dialog box options as shown in bold, initial capital letters. Example: <b>Save As</b> dialog box.			
bold type	External timing parameters, directory names, project names, disk drive names, filenames, filename extensions, and software utility names are shown in bold type. Examples: $f_{MAX}$ , <b>\qdesigns</b> directory, <b>d:</b> drive, <b>chiptrip.gdf</b> file.			
Italic Type with Initial Capital Letters	Document titles are shown in italic type with initial capital letters. Example: AN 75: High-Speed Board Design.			
Italic type	Internal timing parameters and variables are shown in italic type. Examples: $t_{PIA}$ , $n + 1$ .			
	Variable names are enclosed in angle brackets (< >) and shown in italic type. Example: <i><file name=""></file></i> , <i><project name=""></project></i> . <b>pof</b> file.			
Initial Capital Letters	Keyboard keys and menu names are shown with initial capital letters. Examples: Delete key, the Options menu.			
"Subheading Title"	References to sections within a document and titles of on-line help topics are shown in quotation marks. Example: "Typographic Conventions."			
Courier type	Signal and port names are shown in lowercase Courier type. Examples: data1, tdi, input. Active-low signals are denoted by suffix n, e.g., resetn.			
	Anything that must be typed exactly as it appears is shown in Courier type. For example: c:\qdesigns\tutorial\chiptrip.gdf. Also, sections of an actual file, such as a Report File, references to parts of files (e.g., the AHDL keyword SUBDESIGN), as well as logic function names (e.g., TRI) are shown in Courier.			
1., 2., 3., and a., b., c., etc.	Numbered steps are used in a list of items when the sequence of the items is important, such as the steps listed in a procedure.			
••	Bullets are used in a list of items when the sequence of the items is not important.			
$\checkmark$	The checkmark indicates a procedure that consists of one step only.			
R	The hand points to information that requires special attention.			

Visual Cue	Meaning		
CAUTION	A caution calls attention to a condition or possible situation that can damage or destroy the product or the user's work.		
WARNING	A warning calls attention to a condition or possible situation that can cause injury to the user.		
4	The angled arrow indicates you should press the Enter key.		
	The feet direct you to more information on a particular topic.		



# **Tutorial**

Introduction	This tutorial familiarizes you with the Nios II IDE and MicroC/OS-II development flow. The Nios II IDE offers designers a rich development platform for Nios II applications. The Nios II IDE contains the MicroC/OS-II real-time operating system, providing designers with the ability to build MicroC/OS–II applications for the Nios II processor quickly. This tutorial provides step-by-step instructions for building a simple program based on the MicroC/OS-II RTOS. For complete details on MicroC/OS-II for the Nios II processor, refer to the "MicroC/OS-II Real-Time Operating System" chapter in the <i>Nios II Software Developer's Handbook</i> .		
Hardware & Software Requirements	<ul> <li>This tutorial requires the following hardware and software:</li> <li>Quartus<sup>®</sup> II Development Software version 6.1 or later</li> <li>One of the following Nios Development Boards: <ul> <li>Stratix<sup>®</sup> II Edition</li> <li>Cyclone<sup>™</sup> Edition</li> <li>Cyclone<sup>™</sup> Edition</li> <li>Stratix Edition</li> <li>Stratix Professional Edition</li> </ul> </li> <li>Altera USB-Blaster<sup>™</sup> or similar cable</li> </ul> To complete this tutorial, you must have the Nios II IDE installed, and your Nios development board must be connected to a host PC. For details on how to install the software and connect the board, see the Nios II Development Kit Getting Started User Guide.		
Tutorial Files	<ul> <li>The Nios II Embedded Design Suite (EDS) includes the tutorial designs. The hardware design is the standard reference design and the software design is a Nios II IDE template. After you install the Nios II EDS, you can find the design files in the Nios II EDS directory structure.</li> <li>The standard reference hardware design files are located in the &lt;<i>Nios II EDS install path</i>&gt;\examples\<i><verilog i="" or="" vhdl<="">&gt;\<i><nios board<="" development="" i="">&gt;\standard directory.</nios></i></verilog></i></li> </ul>		

The MicroC/OS-II tutorial software design files are located in the <Nios II EDS install path>\examples\software\ucosii\_tutorial directory.

You do not need to move or copy these software files because the Nios II IDE copies them to your working project directory automatically.

#### Hello MicroC/OS-II Design

The example design you use for this tutorial is a simple design that exercises some of the basic features of the operating system. Figure 1–1 is a simplified diagram of the application.

Figure 1–1. Tutorial Example Design



As shown in Figure 1–1 the design has five active tasks. **send\_task** fills up a message queue with incrementing data. **receive\_task1** and **receive\_task2** periodically pull messages out of the message queue. **getsem\_task1** and **getsem\_task2** compete over a shared resource that is protected by a semaphore. The design also has two tasks not shown in Figure 1–1: one for initialization and one for printing status information. The process for creating a MicroC/OS-II software image for the Nios II processor involves the following steps:

- 1. Create a new Nios II IDE project.
- 2. Configure the Nios II system library project.
- 3. Build and run the Nios II software project.

These steps are described in detail in the following sections.

#### **Create a New Nios II IDE Project**

In this section you create a new Nios II IDE project using a software template. Perform the following steps:

- 1. On the Windows Start menu, point to **Programs**, **Altera**, **Nios II EDS** *<version number>*, and click **Nios II** *<version number>* **IDE**.
- On the File menu, point to New, and click Nios II C/C++ Application. The first page of New Project wizard opens.
- 3. Under **Select Project Template**, select the **MicroC/OS-II Tutorial**. The project name and project path are filled in for you automatically. Keep these defaults.
- 4. Click Browse under Select Target Hardware.
- 5. Browse to the standard example directory for the Nios development board that you are targeting, i.e., <Nios II EDS install path>\examples\<verilog or vhdl>\<Nios development board>\standard directory, where <Nios development board> is one of the following options:
  - **niosII\_cyclone\_1c20** for the Cyclone EP1C20 board
  - niosII\_cycloneII\_2c35 for the Cyclone EP2C35 board
  - **niosII\_stratix\_1s10** for the Stratix EP1S10 board
  - niosII\_stratix\_1s10\_ES for the Stratix EP1S10ES board
  - niosII\_stratix\_1s40 for the Stratix EP1S40 board
  - niosII\_stratixII\_2s60 for the Stratix EP2S60 board
  - niosII\_stratixII\_2s60\_ES for the Stratix EP2S60ES board
  - niosII\_stratixII\_2s60\_rohs for the Stratix EP2S60 ROHS board
- 6. Click the file **std**\_<*device name*>.**ptf**.

7. Click Open. You return to the New Project wizard. As shown in Figure 1–2, the SOPC Builder System box under Select Target Hardware contains the path to the .ptf file for the standard example design. Additionally, the CPU box contains the name of the CPU in the example SOPC Builder system.

Figure 1–2. New Project Wizard Page 1

C:\altera\61\nios2eds\	oplication with a default system library as examples/verilog\niosII_cycloneII_2c35
Name: ucosii_tuto	rial_0
Specify Location	:1\nios2eds\examples\verilog\nios11_cyclone11_2d [Browse]
Select Target Hardwa	re
SOPC Builder System	C:\altera\61\nios2eds\examples\verilog\nios 🗸 Browse
CPU:	cpu 🗸
Select Project Templa Hello World Small Host File System Memory Test MicroC/OS-11 Messag MicroC/OS-11 Tutoria Simple Socket Server Tightly Coupled Mem Web Server Zip File System	te

- 8. Click Next to go the second page of the New Project wizard.
- 9. Select the Select or create a system library option.
- 10. Click **New System Library Project** to open the Nios II System Library page. See Figure 1–3.



<b>New Project</b> Nios II System Library Create a new Nios II	system library project.
Name: std_system	n_lib
Location: C:\altera	61\nios2eds\examples\verilog\niosII_cycloneII_2
SOPC Builder System	are,
CPU:	cbn
Select Type of RTOS	Э.
<mark>Inone (single-thread</mark> MicroC/OS-II	ed) Nios II Hardware Abstraction Layer
Ø	Einish Cancel

- 11. Type std\_system\_lib in the Name box.
- 12. Select MicroC/OS-II in the Select Type of RTOS box.

13. Click **Finish** to copy the template files and return to the **New Project** wizard. See Figure 1–4.

Figure 1–4. New Project Wizard Page 2

New Project
ios II C/C++ Application
Choose (or create) a system library A system library is a set of device drivers that provides access to the target
Create a new system library named:
ucosii_tutorial_0_syslib
This new system library project will be located relative to the application project.
∋ Select or create a system library
Available System Library Projects for: std_2C35
☑ ≌std_system_lib [std_2C35]
New System Library Project
Image: Second state         Einish         Cancel

Click Finish to complete creation of your new project. The wizard creates two projects in the Nios II C/C++ Projects view, as shown in Figure 1–5. ucosii\_tutorial\_0 is the application project, and std\_system\_lib is the system library project.

Figure 1–5. Nios II C/C++ Projects



#### **Configure the System Library**

In general, after you create a new system library you must configure it, e.g., defining stdin, stdout, stderr, etc. For this tutorial, you must configure MicroC/OS-II. During configuration, the Nios II IDE saves the appropriate values into the **system.h** file. Perform the following steps to configure the MicroC/OS-II kernel.

- Right-click on the system library, std\_system\_lib, in the Nios II C/C++ Projects view.
- 2. Click **System Library Properties** in the pop-up menu to open the **Properties** dialog box. If you are using uC/OS-II in evaluation mode as distributed by Altera, you might see the license notification in Figure 1–6 on page 1–8. Click **OK**. The system library options dialog box appears as shown in Figure 1–7.



#### Figure 1–6. uC/OS-II License Notification

Figure 1–7. System Library Options

Properties for std	_system_lib				
type filter text	System Library			⇔ • ⇔ -	
Info Builders C/C++ Build C/C++ Documentation C/C++ File Types C/C++ File Types C/C++ Indexer Project References Refactoring History System Library	Target Hardware SOPC Builder System: C:\altera\61\nios2 CPU: cpu System Library Contents RTOS: RTOS Options I stdout: stderr: stdin: System clock timer: Timestamp timer: Max file descriptors: Program never exits	e stem: C:\altera\61\nlos2eds\examples\verilog\nlos11_cyclone11_2c35\standard\std_2C35.ptf  cpu  Contents  none (single-threaded)  linker Script  contents  inga_uart  inga_uart		Browse Browse ddr_sdram v ddr_sdram v ddr_sdram v ddr_sdram v ddr_sdram v	
Q	Support C++     Reduced device drivers       Lightweight device driver API     Small C library       Link with profiling library     ModelSim only, no hardware support       Emulate multiply and divide instructions     Run time stack checking       Software Components     Software Components		Exception stack memory: Maximum exception stack size (bytes): Help Restore Del	aults Apply	

3. Click **RTOS Options**. The **MicroC/OS-II RTOS Options** dialog box opens, as shown in Figure 1–8.

Figure 1–8. MicroC/OS-II RTOS Options

🕼 MicroC/OS-II RTOS Options 📃 🗖 🔀								
<ul> <li>MicroC/OS-II</li> <li>Event Flags</li> <li>Mutex</li> <li>Semaphores</li> <li>Mailboxes</li> <li>Queues</li> <li>Memory Managem</li> <li>Miscellaneous</li> <li>Task Managemen</li> <li>Time Managemen</li> <li>Timer Managemen</li> </ul>	MicroC/OS-II							
	MicroC/OS-II General Options         Maximum number of tasks       10         Lowest assignable priority       20         Image: Thread Safe C Library       20         Code Generation Options       20         Code for Semaphores is always on, as it is required by the HAL       20         Enable code for Event Flags (used by UART and JTAG UART drivers)       20         Enable code for Mutex Semaphores       20         Enable code for Mailboxes       20         Enable code for Memory Management       20         Enable code for Timers       20							
<								
0	OK Cancel							

- 4. Click the "+" in the left hand panel to expand the contents under MicroC/OS-II. MicroC/OS-II is highly configurable. The options you select in this dialog box are saved to the system.h file and determine which MicroC/OS-II options are included in the binary image. Examine the options you can select by clicking each of the categories listed at the left side of the dialog box, under MicroC/OS-II.
- For details on the various MicroC/OS-II features, refer to the "MicroC/OS-II Real Time Operating System" chapter in the Nios II Software Developer's Handbook.
  - 5. Click **OK** to use the default settings. You return to the system library properties dialog box.
  - 6. Click **OK** to complete the configuration.

You have finished configuring the system library and are ready to build and run the example as described in the following section. • For additional details on how to build and run programs with the Nios II IDE, see the *Nios II Software Development Tutorial* in the Nios II IDE help system.

#### **Build & Run the Nios II Software Project**

In this section, you run the example design on a development board. Using Nios II IDE, you build the application, configure the development board with a valid configuration (**.sof**) file, and download the Executable and Linkable Format File (**.elf**).

- In the Nios II C/C++ Projects view, select the ucosii\_tutorial\_0 project.
- 2. On the Tools menu, click **Quartus II Programmer** ... to open the Quartus II Programmer.
- 3. On the Quartus II Programmer's File menu, click **Open**.
- 4. Select **standard.sof** as shown in Figure 1–9.

Figure 1–9. Open Dialog Box



5. Click **Open**. You return to the Quartus II Programmer.

	II Programmo	r - IChain 1	edf*1							
File Edit O	ntions Processing	Help	.cai j				_			
A Hardware Setup	Mode: J	▼ Pr	ogress:	0:	0%					
Enable real-time ISP to allow background programming (for MAX II devices)										
🏓 Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check Exa	mine Secur		
🗰 Stop	C:/altera/61/nios2eds/e	EP2C35F672	0068A8B7	FFFFFFF	4					
Auto Detect					М					
🗙 Delete										
🍰 Add File										
🔛 Change File										
Save File										
😂 Add Device										
📫 Up										
Down	<							>		
Sustem										
Ready										

Figure 1–10. Quartus II Programmer

- 6. Turn on the **Program/Configure** check box to the right of the **.sof** filename, as shown in Figure 1–10.
- 7. Click **Start** to configure the FPGA on the development board with the SOF. Upon successful completion, the Quartus II Programmer displays messages similar to this:

```
Info: Configuration succeeded -- 1 device(s) configured
Info: Successfully performed operation(s)
Info: Ended Programmer operation at Wed Jan 03 17:50:03 2007
```

8. On the File menu, click **Exit** to close the Quartus II Programmer. You return to the Nios II IDE.

If the Quartus II Programmer asks if you want to save changes to the **chain1.cdf** file, click **No**.

9. On the Run menu, click **Run As**, and click **Nios II Hardware** to build the program, download it to the board, and run it.

•••

For additional information on the Nios II IDE build process and run configuration setup, see the *Nios II Software Development Tutorial* within the Nios II IDE online help.

After download is complete, the Nios II IDE **Console** view is updated periodically by print\_status\_task() as shown in Figure 1–11. The numbers displayed vary depending on the type of Nios development board the example runs on.

Figure 1–11. print\_status\_task() output

Hello From MicroC/OS-II Running on NIOS II. Here is the status: The number of messages sent by the send\_task: 123 The number of messages received by the receive\_task1: 73 The number of messages received by the receive\_task2: 24 The shared resource is owned by: getsem\_task2 The Number of times getsem\_task1 acquired the semaphore 240 The Number of times getsem\_task2 acquired the semaphore 185

Congratulations! You have successfully configured, built, and run a MicroC/OS-II program.



For further details on this tutorial, you can examine the source file **ucosii\_tutorial.c**. For complete reference on MicroC/OS-II, refer to the "MicroC/OS-II Real Time Operating System" chapter in the *Nios II Software Developer's Handbook*.