



KTH Information and
Communication Technology

IS1500 Computer Engineering and Components 2014

Preparatory Exercise firstproj – Getting to know the Nios II IDE

Latest update: 2014-06-23

Contents

• Files used - a complete list, so you can download them all and then work offline	1
• Read this first - our best advice comes first	1
• Stepwise Instructions	2
• Starting the Nios II Software Build Tools	2
• Creating a new Project	3
• Creating a Source File	5
• Compiling your code	7
• Starting the NIISim simulator	8
• Loading the DE2 Board into the NIISim Simulator.	9
• Loading your compiled program into the NIISim Simulator.	9
• Starting and Stopping Simulation.	10

Files used for this tutorial

All files are available from the course web.

- You need a software installation of the Nios II Software Build Tools for Eclipse. The Nios II Software Build Tools for Eclipse is bundled with Quartus II, version 11 or later. If you install yourself, please select **Quartus II version 13.0sp1**, which is compatible with all laboratory exercises in this course. Later versions are not compatible.
- You need a configuration file for the Nios II Software Build Tools, and a sample program.
- You need the NIISim simulator.

Read This First

This document describes how to create your first project with the Nios II Build Tools for Eclipse.

You will create a new project for each laboratory task. Parts of this first project can be re-used.

This document will also show how to simulate your program with our NIISim simulator.

Some screenshots in this document show earlier software versions. Please read the text.

Stepwise instructions

These instructions refer to version 12.0. Versions 11 and 13.0 are similar. Do not use versions earlier than 11.0, or later than 13.0.

Step 1. Starting the Nios II Software Build Tools

Windows 7: You will find the icon under Start menu - All Programs - Altera 12.0 - Nios II EDS 12.0 - Nios II Software Build Tools. Replace 12.0 with your the actual Altera version number.

Ubuntu 12.04 LTS: use Ctrl-Alt-T to start a Terminal. Type (or copy and paste)

```
/opt/altera12/nios2eds/nios2_command_shell.sh
```

A new shell will start. In that shell, type

```
eclipse-nios2
```

The first time you start the Nios II SBT, use the workspace launcher to set your workspace-folder.

A *workspace folder* is a folder somewhere in the file system. All files you create and use with the Nios II Software Build Tools will be stored in sub-folders of your workspace folder. The path to that folder must have the following required properties:

1. There must be no spaces anywhere in the path.
2. All folders in the path must be readable by you.
3. The workspace-folder itself - the folder at the end of the path - must be both writable and readable by you.
4. The folder must be easy to find for you - you will copy downloaded files to this folder several times.

In the Workspace Launcher, click Browse to change the workspace path.

If you use a computer in a managed computer room, with Windows 7: use a new subfolder of your network-home directory, typically H:\nios2workspace

If you use your own computer, with Windows 7: use a new subfolder of your home directory, typically C:\Users\Student\Documents\nios2workspace

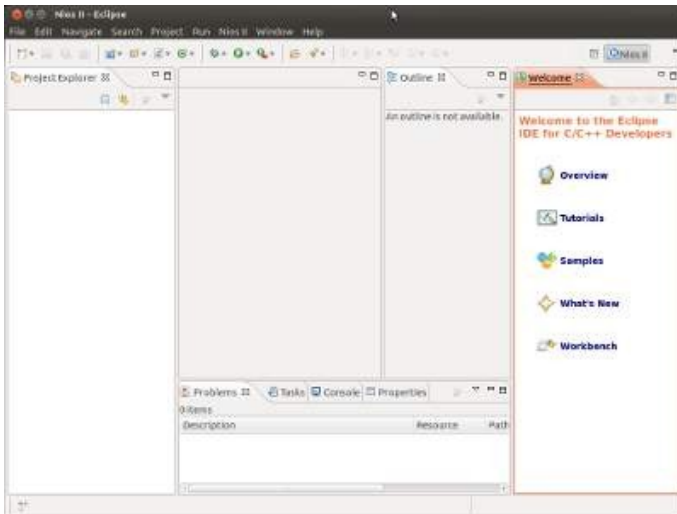
Warning: if your username contains spaces, this will not work.

If you use any computer with Ubuntu 12.04 LTS: use a new subfolder under your home directory - change workspace into nios2. This will fulfil all the required properties, and also avoid possible conflict with other projects using an Eclipse-based development environment.



Note: Changing the workspace folder will hide all old projects and files, until you change the setting back again. Take some care to get this right the first time.

This is the initial, empty workspace.

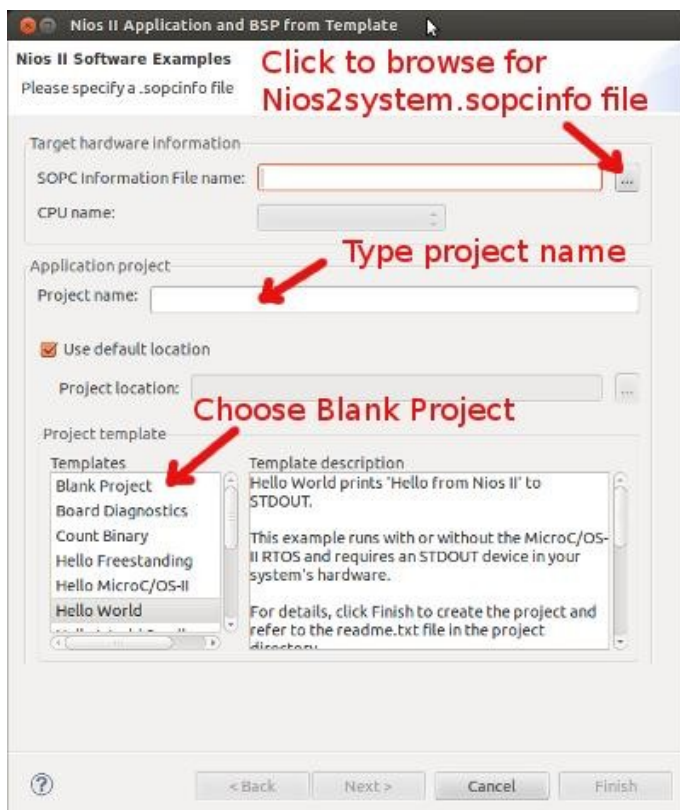


You can close the Welcome tab (rightmost in the picture) to free up screen area for your work.

Step 2. Creating a new Project

If you haven't already done so, now is the time to download the configuration file for the Nios II Software Build Tools – see the course web. **You must save the file into your workspace folder.**

Click "File", then "New", then "Nios II Application and BSP from template" to start the New Project wizard.

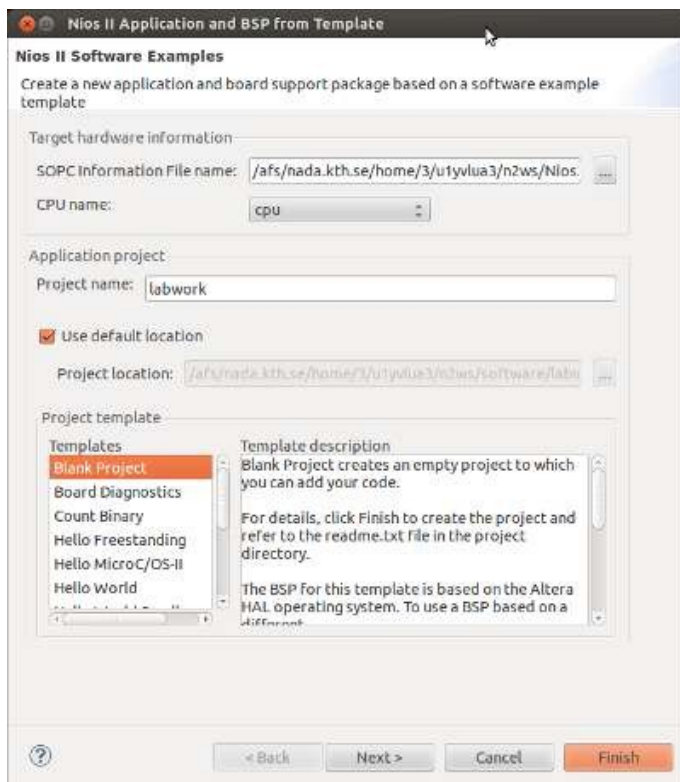


Browse to your **workspace folder** and select the configuration file you copied there. The file should be called something like `Nios2system.sopcinfo`.

If the `Nios2system.sopcinfo` file is **not yet** in your workspace folder, **click Cancel**. Then, copy the file to your workspace folder, and then start again with File – New – Nios II Application and BSP from template.

Type the project name: `labwork`

Select "Blank Project".



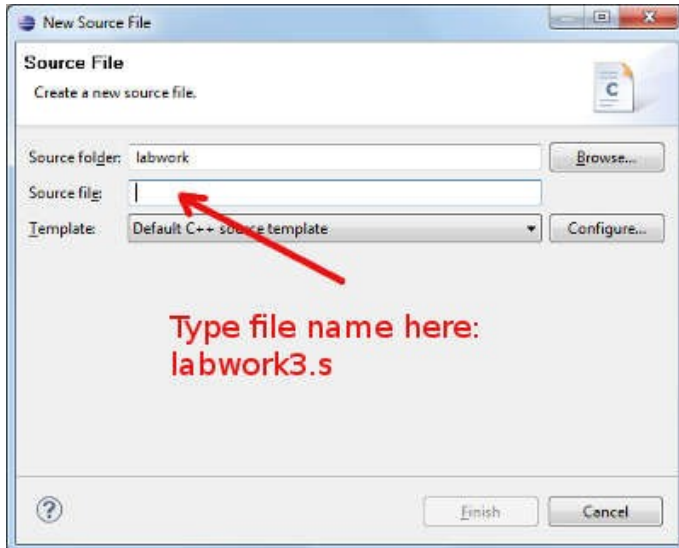
Click "Finish".

Advanced: The Nios II Software Build Tools will now create two projects - your `labwork` project and a Board Support Package project called `labwork_bsp`. The Board Support Package is quite big, and can take a while to compile. In future, you can re-use the same `labwork_bsp` for all projects in laboratory exercises ***nios2time*** and ***nios2io***. Later on in the course, later laboratory exercise ***nios2int*** and the Home Practical #3 (on Threads and Synchronization) will require a different Board Support Package project.

Step 3. Creating a Source File

Select your new project, by clicking on the name.

Now, **right-click** the project name, then select **New**, then **Source File**.



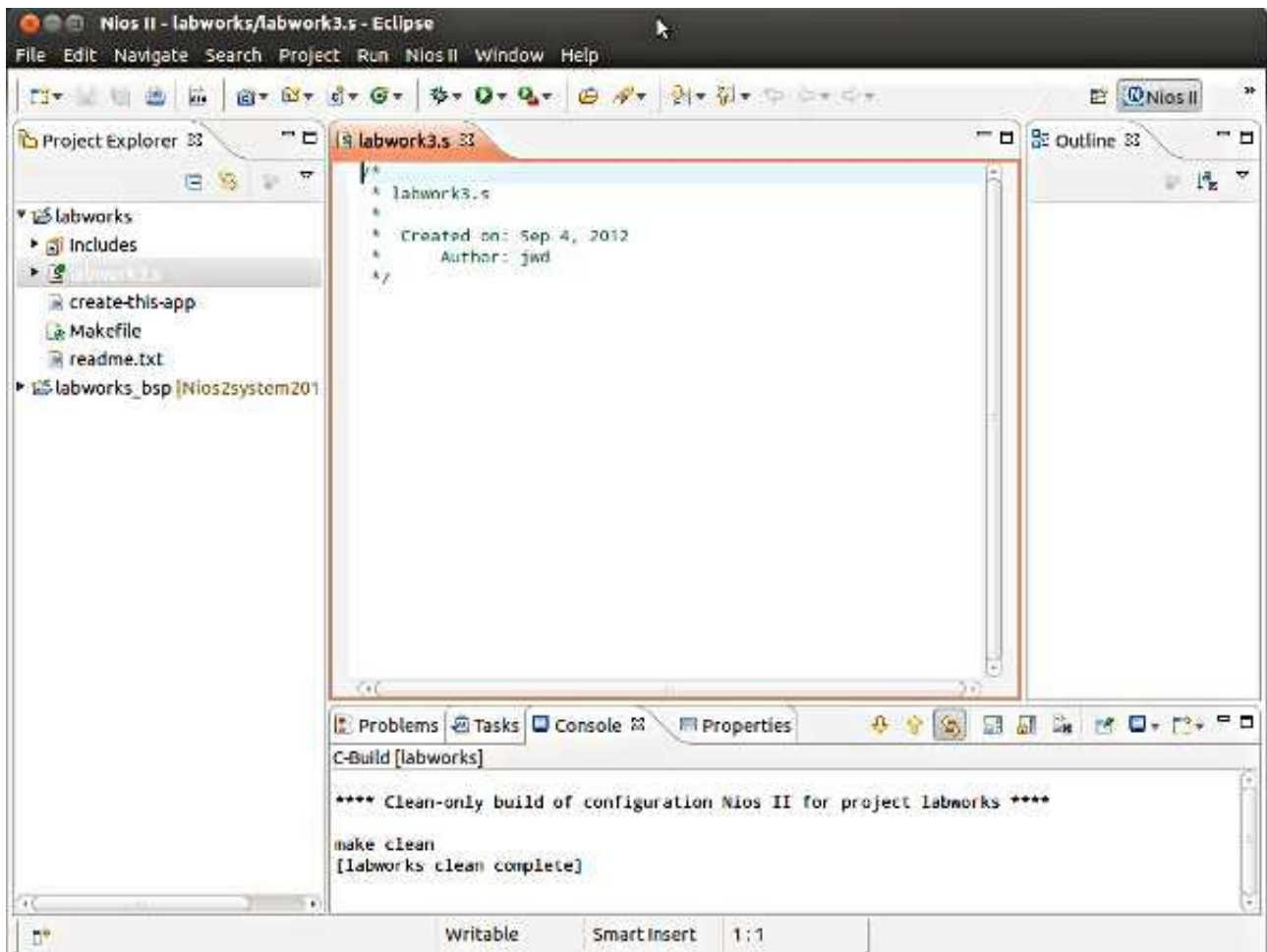
You will see a dialog similar to this one:

Type in the name of your source file. To create an assembly-language source file, the filename must end with ".s" - a period followed by the lower-case letter s. The name can only contain letters a-z and numbers. In this example, use the name: `labwork3.s`

Change the Template to "None", then click "Finish" to continue.

Advanced: in future, you will need to create C-language source files. The name of a C source file must end with ".c".

Your Nios II IDE should now look like this.



Delete all text in the file you just created.

Copy the contents of our `labwork3.s` file, and paste the contents into the file.

After pasting the contents of the file into the source code window, your workspace should look something like this.

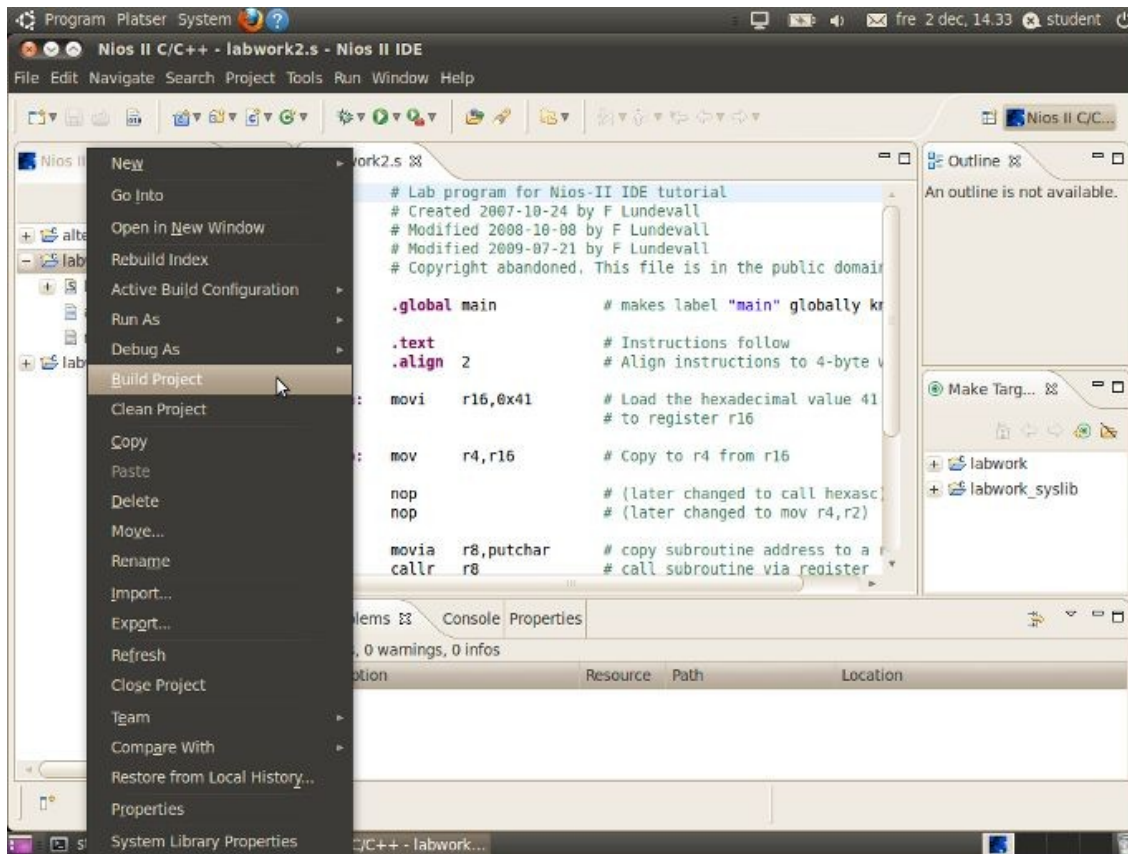
Scroll to the end of the file. Insert a blank line at the end.

Sometimes, the editor inserts whitespace at the beginning of the new line. In that case, delete those whitespace characters. **Always make sure that the end of the file is at the beginning of a new line.** If you ignore this, you will experience some really nasty errors later on.

Click "File", then "Save All" to save the changes you made to the source file `labwork3.s`.

Step 4. Compiling your code

Make sure that all files are saved. If in doubt, select File – Save All.

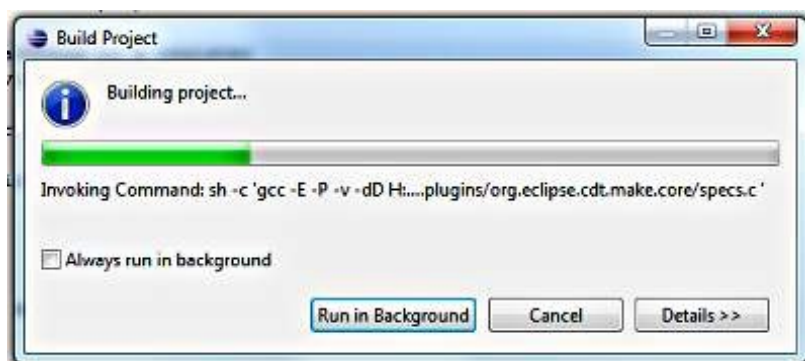


Click once on the project name, to select the project instead of the source file.

Now, right-click the project name. In the pop-up menu, click "Build Project".

This will translate your program into a binary code that the Nios II processor can execute. This can take several minutes. **Most of the time is spent compiling the system library** - that's why you will want to re-use the library as much as possible.

You will see a progress-bar pop-up window while the Nios II Software Build Tools compiles your project.



After compilation, check the Problems tab. You should see "0 items". '

All errors and all warnings must be fixed.

After fixing a warning or an error, perform the following steps (again):

1. Select File – Save All.
 2. Right-click the project name and select "Clean Project". Wait until cleaning is finished.
 3. Right-click the project name and select "Build Project".
-

Step 5. Starting the NIISim simulator

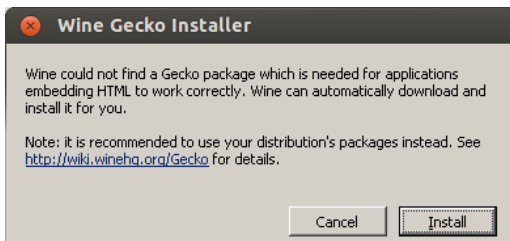
When the project has compiled flawlessly, it's time to start the NIISim simulator.

Download the zipped archive, and open it with a Zip-archive utility (double-clicking works on most systems). Make sure that "Use folder names" is checked (enabled), then extract the contents into your workspace folder.

Use the file manager to navigate to your workspace folder, and open the `niisim` sub-folder.

- On Ubuntu systems, right-click the file `NIISim.exe`, and select Open With Wine Windows Program Starter.
- On Windows systems, just double-click the **NIISim** executable.

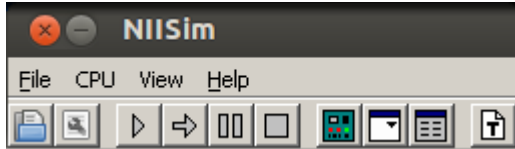
Ubuntu systems only: If this is the first time you start a Wine program, Wine will initialize. You may see this question:



NIISim does not need a Gecko package, so just click "Cancel".

Step 6. Loading the DE2 Board into the NIISim Simulator.

When NIISim has loaded, you will see this small window:



Now and then, you may think the simulator failed to start - you don't see the window. **Look for it!** It's really very small.

Click the Settings icon to *load a system description file*. In case you prefer menus, the same function is available in the File menu.



Choose `datorteknik.sdf` as your system description file. This file is in the `niisim` folder (created when you extracted the simulator archive).

The simulator now loads a description of the particular board we use (the DE2 board), and of the Nios II system we use.

Step 7. Loading your compiled program into the NIISim Simulator.

Next, click the Open icon to *load a program file*. Again, the same function is available in the File menu, should you prefer that.



Navigate to the program file created by the Nios II Software Build Tools. How to find this file:

1. Start in your workspace folder. Open the sub-folder `software`,
2. In the `software` folder, open then the folder with your project name. In our case, this will be the `labwork` folder.
3. In the project folder `labwork`, there will be a file called `labwork.elf` - select that file and click Open.

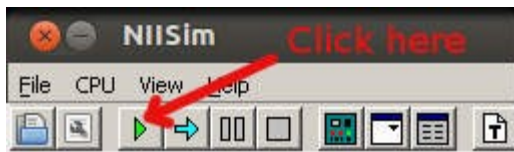
The simulator now loads your compiled program file.

Step 8. Starting and Stopping Simulation.

Click the console-view icon and select JTAG UART. The same function is available as View – Consoles – JTAG UART. A console window will pop up.

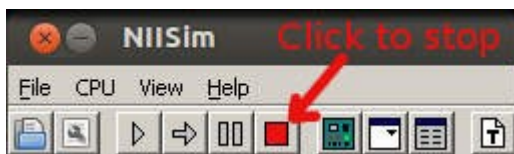


You can now run your program by clicking the green triangle. Alternatively, in the CPU menu, select Run.



The console will start filling up with output from your program.

To stop the program, click the red Stop button. You can also select Stop from the CPU menu.



You can now inspect the program output at your leisure.

Use the left-to-right scroll bars to check out the beginning and end of output-lines, and use the up-down scrollbar to check the first and last lines of output. For simpler viewing, you can resize the console window so that most of the output fits.

Congratulations! You have just finished running your first assembly language program. Please continue with the laboratory exercise *nios2time*.