

1. Quartus Howto: Linux 3P28

1.1. Your Environment

On the command line, enter

```
echo $PATH
```

This will print the current contents of the `PATH` environment variable. ie. the search path for executables. If it contains the following

```
/usr/local/quartus/bin  
/usr/local/quartus/sopc_builder/bin  
/usr/local/quartus/sopc_builder/kits/nios/bin  
/usr/local/quartus/sopc_builder/kits/nios/bin/nios-gnupro/bin
```

and executing

```
echo $sopc_builder
```

gives

```
/usr/local/quartus/sopc_builder/
```

then you may ignore Section 1.1.1 and goto 1.2

1.1.1. Setting the environment variables

Using an editor, eg emacs, do the following (note the `.` at the start of the file name):

```
emacs .bash_profile &
```

when the file is open add the following lines to it.

```
export PATH=$PATH:/usr/local/quartus/bin  
export PATH=$PATH:/usr/local/quartus/sopc_builder/bin  
export PATH=$PATH:/usr/local/quartus/sopc_builder/kits/nios/bin  
export PATH=$PATH:/usr/local/quartus/sopc_builder/kits/nios/bin/nios-gnupro/bin  
export socp_builder=/usr/local/quartus/sopc_builder/
```

save the file and on the command line enter

```
. .bash_profile
```

which is (dot space dotbash_profile) You only need to do the above once only, it will run automatically next time you log in.

1.2. Setting up the nios example

For the first exercise you will need to copy an example project into a new directory. First, create a new directory for this module.

For example ...

```
mkdir Quartus_Projects
```

change into this directory and create a directory for the first exercise.

```
mkdir nios_tutorial_1
```

change directory into it.

```
cd nios_tutorial_1
```

Use the following to copy the example to this directory, type it as one long line, without the \ after examples

```
cp -a /usr/local/quartus/sopc_builder/kits/nios/examples\  
/vhdl/nios_apex_20k200/standard_32 ./
```

do an ls to check that you now have a directory full of stuff.

1.3. Starting Quartus:

For some strange reason Quartus does not always start properly in the lab, so you may need to do this every time you start quartus :-(. To start Quartus ...

on the command line enter

```
quartus &
```

This will start quartus,

you should see a process number on the screen, something like

```
[1] 23157
```

The large number is the process id. You have to kill quartus and then restart it.

On the command line enter

```
kill -9 the process number on the screen
```

for example

```
kill -9 23157
```

If there was no process number then, on the command line enter

```
ps
```

and use the first number for the first quartus in the list.

Quartus should now start. It's a bit slow at first so be patient.

2. Setting up quartus

one time only, hopefully.

2.1. License Server setup

On the quartus menu bar go to

```
tools ->license setup
```

in the field labelled **license file** enter

```
2700@164.11.222.200:/usr/local/flexlm/licenses/license.dat
```

and click ok.

You should see some new info appear.

This means that quartus can now see the license server.

2.2. Proxies and Libraries

Some of the settings have to be made via the 'SOPC' tool. This can only be started from within Quartus. It also requires that a project is open. Therefore go to ...

```
file ->open project
and select the file with the blue symbol. This will add some files to
the left-hand window in quartus. Now go to
tools ->sopc builder.
eventually this will open a new large window.
in this window (sopc_builder) go to
file -> socp builder setup
In the field 'additional directories' enter
/usr/local/quartus/sopc_builder/kits/nios/
tick the box 'Use Proxy Host'
and in the 2 fields enter
cache01.uwe.ac.uk 80
click ok and exit socp_builder.
```

3. Getting Started

You can now start on the tutorials. The tutorial files are all written from a Microsoft Windows perspective. You will need to translate references to 'C:\path\to\project' as references to your current project directory. There are many manuals, reference guides and tutorials for the Quartus and SOPC tools, mostly with cryptic file names. Most of the ones you will need to refer to can be found in

```
/usr/local/quartus/sopc_builder/kits/nios/documents
```

The important ones for now are the 'Nios User Guide: Getting Started Guide' and the 'Nios Tutorial, Apex Hardware'. The filenames for these are

```
ug_nios_gsg_apex_20k200e.pdf
tt_nios_hw_apex_20k200e.pdf
```

The 'Getting Started Guide' provides an overview of the documentation, installation instructions (which you can ignore) and a quick run through of software development based on an existing project that you copied across. Again there are a number of issues to note; the user guide and nios tutorial are written on the assumption that you are running Quartus via cygwin¹.

- o) Ignore all references/sections to installing the software (obviously)
- i) Ignore all references to the `nios sdk shell`. This is a standard terminal started by a script which also sets up ONLY the paths needed by `quartus/sopc-builder` and wipes out all other paths from the environment!
- ii) Substitute `/dev/ttyS0` in references to 'COM 1' and `/dev/ttyS1` in references to COM 2. The systems in the lab are set-up as standard to use `ttyS0` for communication with the board and `ttyS1` for debug.
- iii) Not all of the boards will have a default nios image on them as they are used for other projects. If attempts to communicate with the board via `nios-run` then skip forward to the section on downloading a nios image to the board found in section 4 of this worksheet and the 'Nios Tutorial, Apex Hardware')
- iv) Page 16 of the *Getting Started User Guide* refers to the verilog example files in the default installation directory. We will be using the VHDL examples which you copied in step 1.2.

¹ Cygwin: Gnu/Linux emulator for Microsoft Windows

4. Hardware Development

In this section you will create a new embedded system from scratch. You will follow the tutorial found in `tt_nios_hw_apex_20k200e.pdf`. The general comments, given above, about interpreting the documents also apply here. Specific issues are as follows.

- `cd` into your *Quartus_projects* directory and create a new folder for this tutorial.
- **Page 13:** If quartus is not running then follow the instructions in section 1.3 of this document.
- **Page 14.3:** Set the working directory to your newly created directory.
- **Page 18.5:** Specify VHDL as the output type.
- **Page 51:** Ignore the EDA settings.
- **Page 56:** There are sometimes issues getting the jtagd and byteblaster module to cooperate. Hopefully these will have been resolved.
- **PAGE 57: Always check that the switch positions on the board are as shown in the diagram!!!**
- **Page 59:** Interpret references to the SDK shell as references to a standard xterm running bash. Remember it is written for MSWindows users not for those familiar with unix shells.
- **Page 60:** Ignore MS_windows references.
- **Page 60-1:** Substitute the path to your project directory in path references. Downloads via the serial connection can take a while, for small programs it is ok but for large configuration files it can take quite a while.
- **Page 62 :** Stop at the point where it says 'Download design to flash memory'