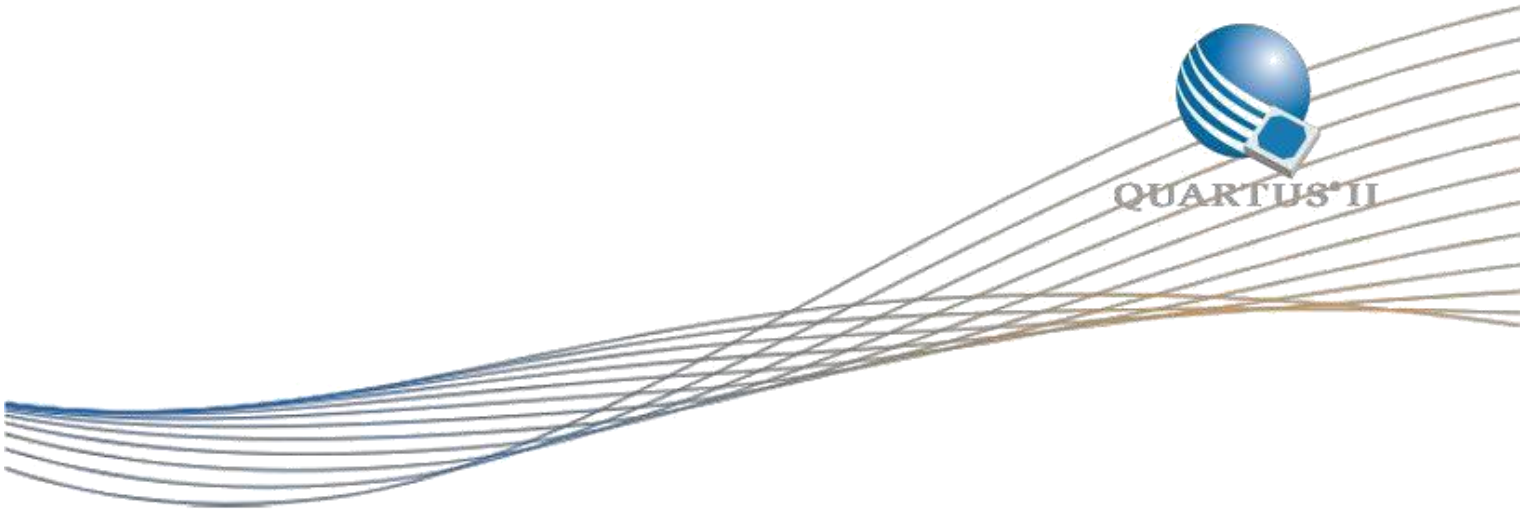


Adapting Digilent PmodCLP LCD to Arduino Development Kit with switches and “Hello World” display

Author: Mehrmah Haider
Date: 11/17/2016
Revision: 1.0

Table of Contents

Overview

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

- MAX10 10M50 DE10 Lite kit
- Mini-USB cable for programming the device
- A Digilent PmodCLP LCD screen – can be purchased here
- Quartus 14.1 or later
    - This is required only if you want to modify or recompile the example)
    - Quartus 14.1 must have "Update 1" installed. Please refer to the Altera Download Center for information on updates.
- **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

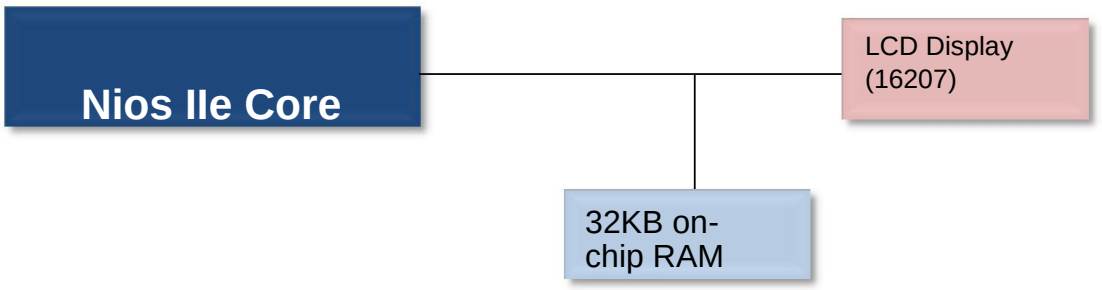


Figure 1: Hello World Design Block Diagram

The NIOS II processor is used to print a basic "hello world" message to an LCD as well as provide a display of 1s and 0s which would indicate the value of each switch (on or off) . The provided Optrex

16207 LCD Controller Core is used to control the LCD through the NIOS. More information on its functionality can be found in Chapter 9 of the IP Peripheral User Guide.

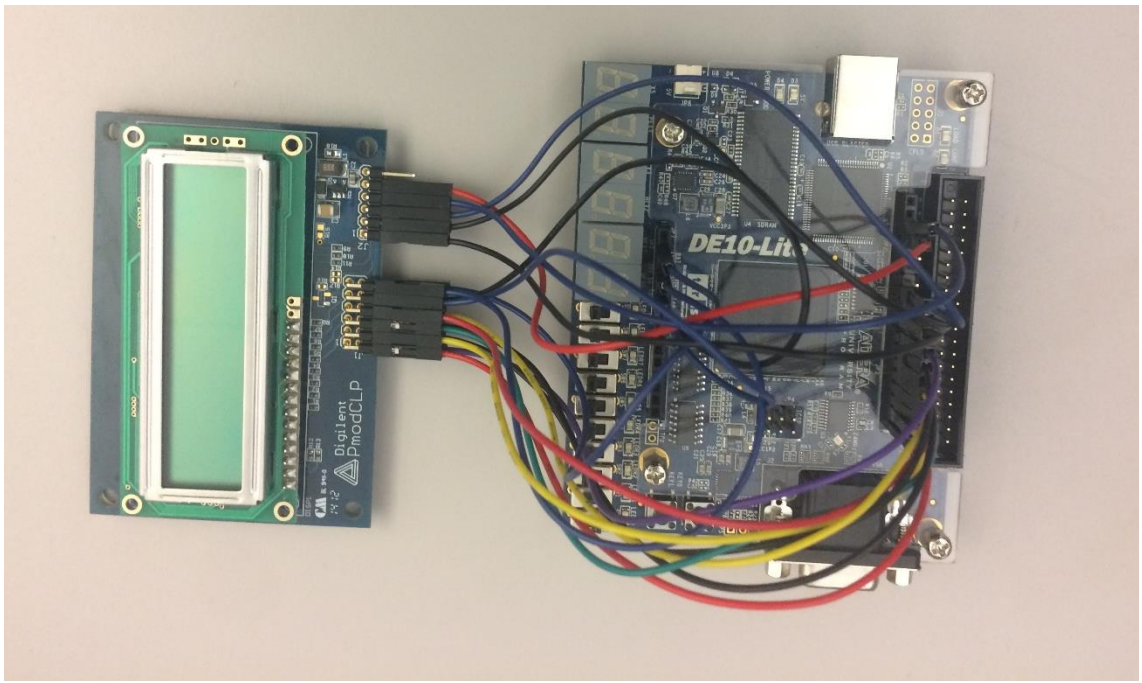
How to Compile the Hardware

1. Connect the LCD module to the dev kit using female to male wires as follows.

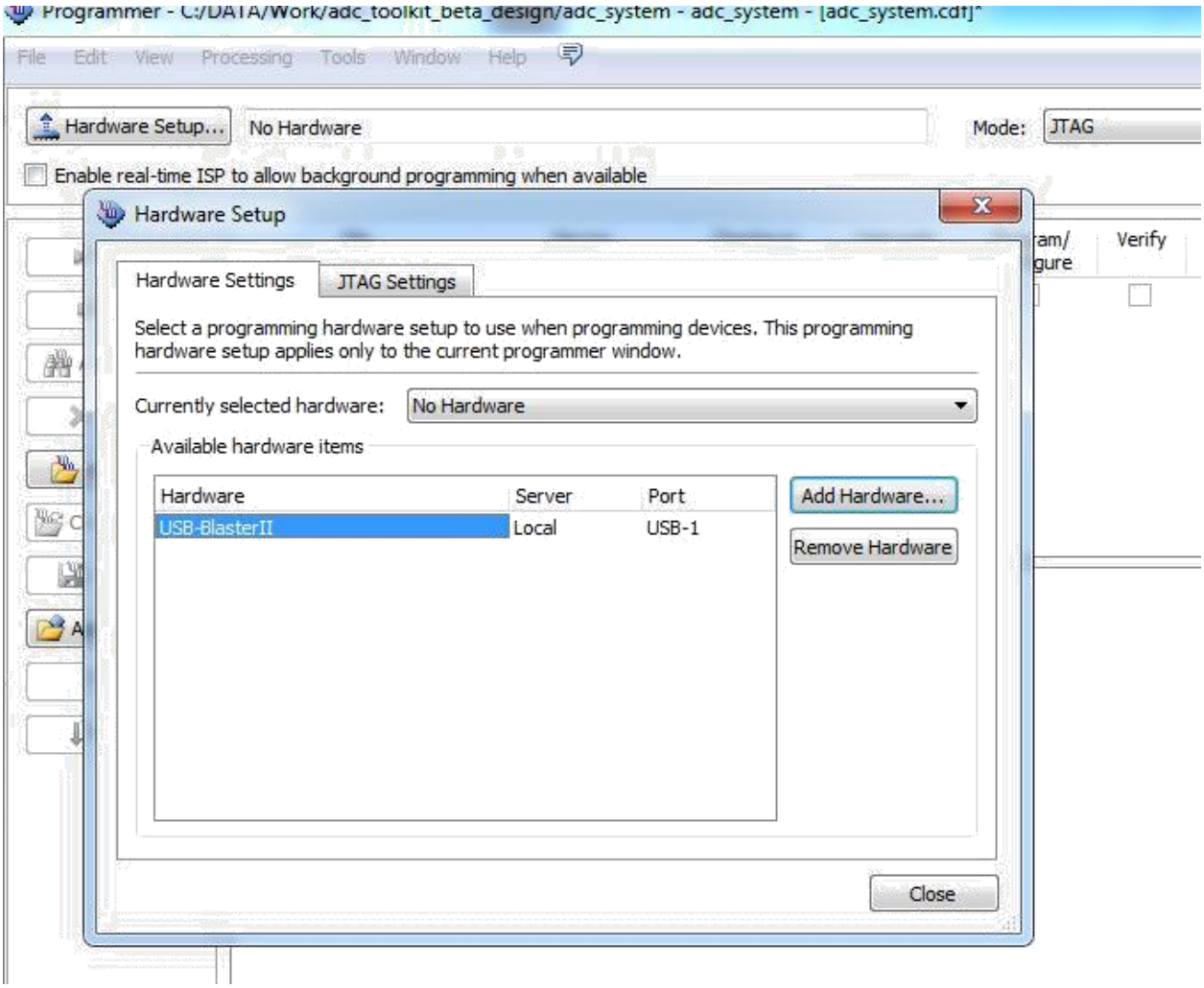
LCD Module	PMOD PIN	Arduino
LCD_DATA[0]	J1-1	JP3_IO0
LCD_DATA[1]	J1-2	JP3_IO1
LCD_DATA[2]	J1-3	JP3_IO2
LCD_DATA[3]	J1-4	JP3_IO3
GND	J1-5	JP7-GND
VCC	J1-6	JP7-3V3
LCD_DATA[4]	J1-7	JP3_IO4
LCD_DATA[5]	J1-8	JP3_IO5
LCD_DATA[6]	J1-9	JP3_IO6
LCD_DATA[7]	J1-10	JP3_IO7
GND	J1-11	JP7-GND
VCC	J1-12	JP7-3V3
LCD_E	J2-3	JP2_IO10
LCD_RS	J2-1	JP2_IO8
LCD_RW	J2-2	JP2_IO9
NC	J2-4	JP2-NC
GND	J2-5	JP2-GND

   Pins J1 - 1-6 are the found in the top half of Header J1 while J1 – 7-12 are found in the bottom half of Header J2

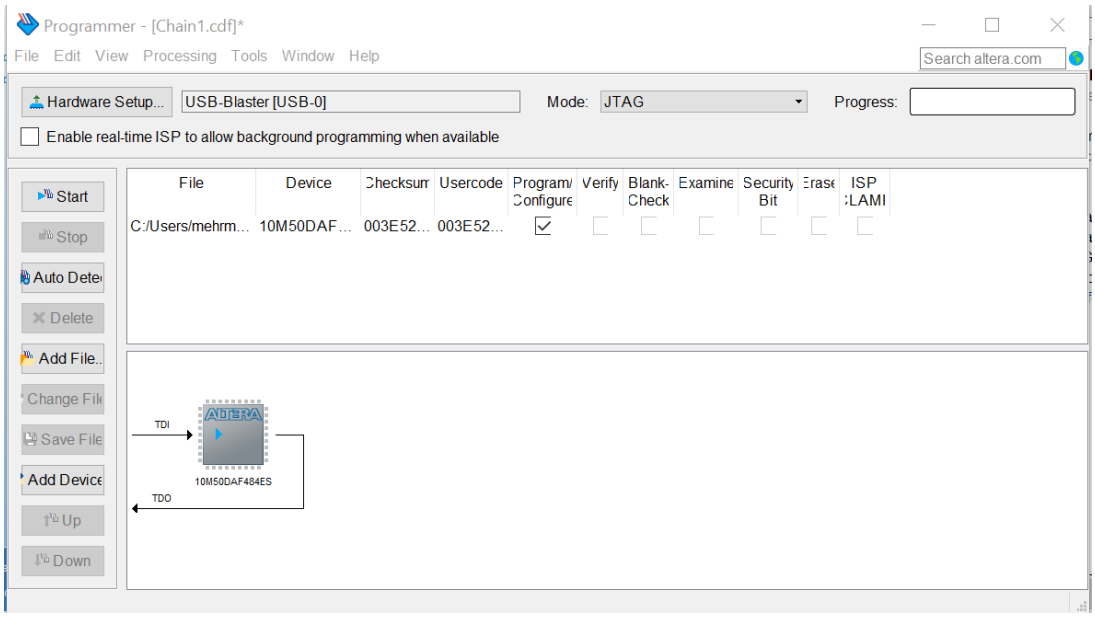
2. Connect the USB from your PC/laptop to the J3 USB connector. You're set up should look like below:



3. The design package includes a pre-compiled version of the design under the master_image folder. This version of the configuration includes the combined FPGA configuration and NIOS executable. Follow these instructions to load the design onto your kit:
    a. Startup Quartus. Then start the programmer from Tools → Programmer.
    b. Click Hardware Setup → USB-Blaster II and select Add Hardware



d. Click “Auto Detect” and double click where it says <none> on the first row to add your hello_LCD.sof or hello_LCD.pof file from the master_image folder. Either one will work, but the .sof will have to be re-downloaded if the board is turned off.



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.

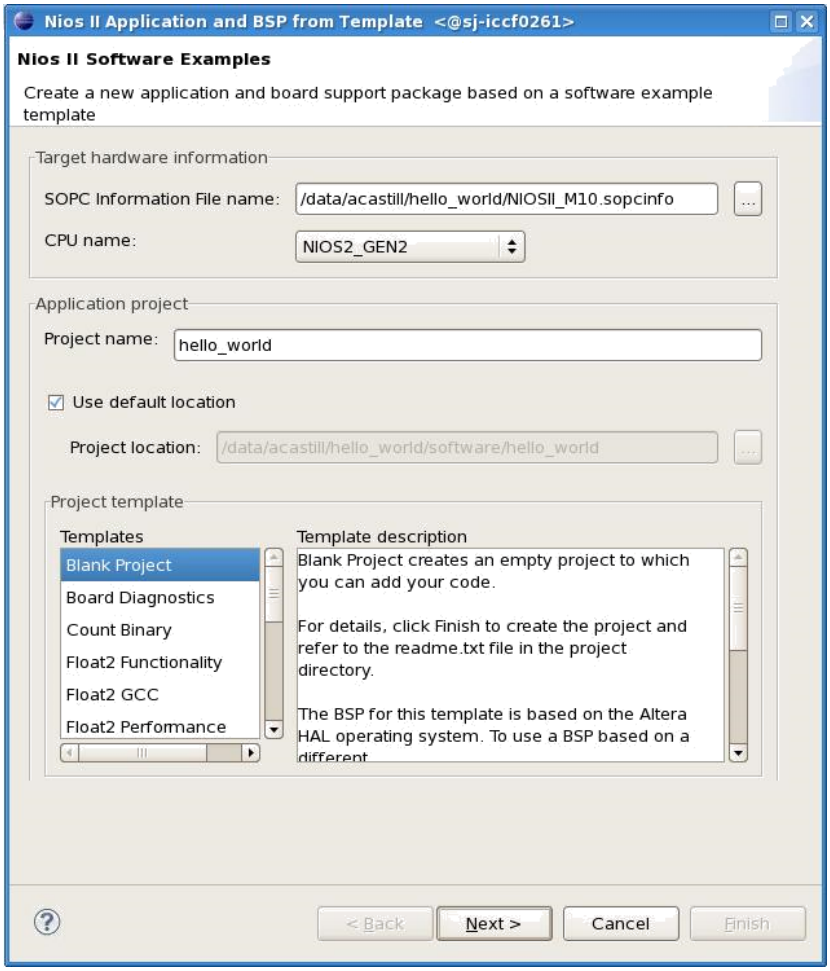
4. You should see the following output on the LCD screen:

How to compile the software

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

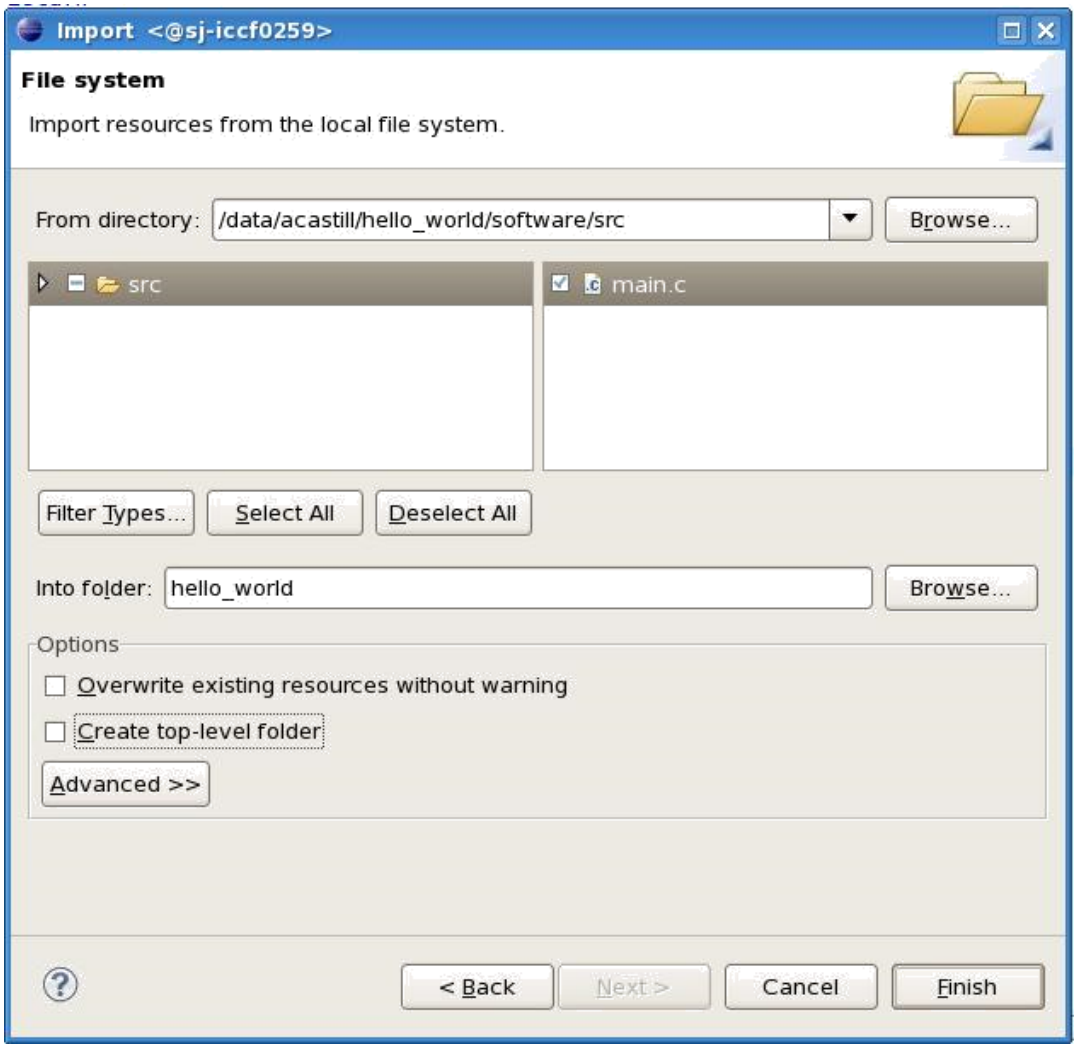
i) Create a software project (BSP and application) from a .sopcinfo file
ii) Add source code (that is used to the program the LCD) to the project
iii) Download and run source code on the target processor (NIOS II)

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.
    a. Ideally your workspace would be in <location_of_Quartus_project>/software
4. When a blank workspace window pops up, right click anywhere in the Project Explorer and select the following to create a new software project;
    a. **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. The .sopcinfo file contains information about the Qsys system, each module instantiated in the project, and parameter names and values contained in the project. The .sopcinfo file is generated by the Qsys file – we have already provided this file for your convenience in **<project folder>/platform/NIOSII_M10**
    a. Provide a name for the software project i.e. hello_world
    b. Select Blank Project for a template and hit “Finish”



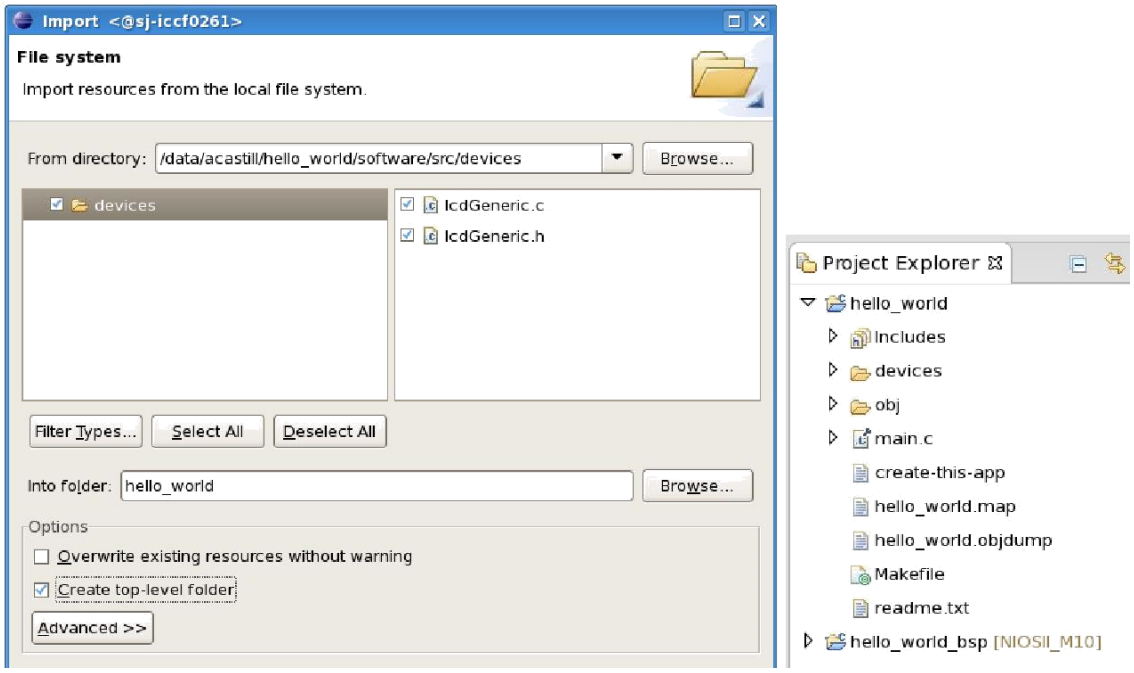
Note: In the future, if you modify the Qsys file for your own design, make sure you are using the most recently generated .sopcinfo file. Check the timestamp for the file.

6. In the Project Explorer, right click the project name and left click the import command.
7. Navigate to **General → File System**
    a. Browse to **software → src**; you should see main.c and devices. Click OK.



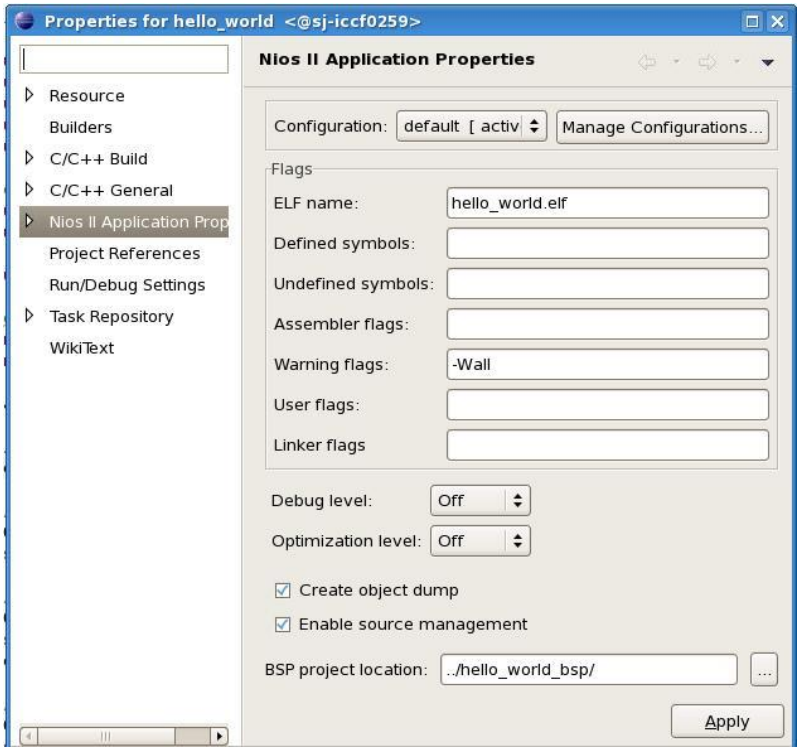
b. Select main.c as shown to add it to your project.

8. Import the contents of the devices folder under software → src
   Check the folder to select all the items at once, and check “Create top-level folder”

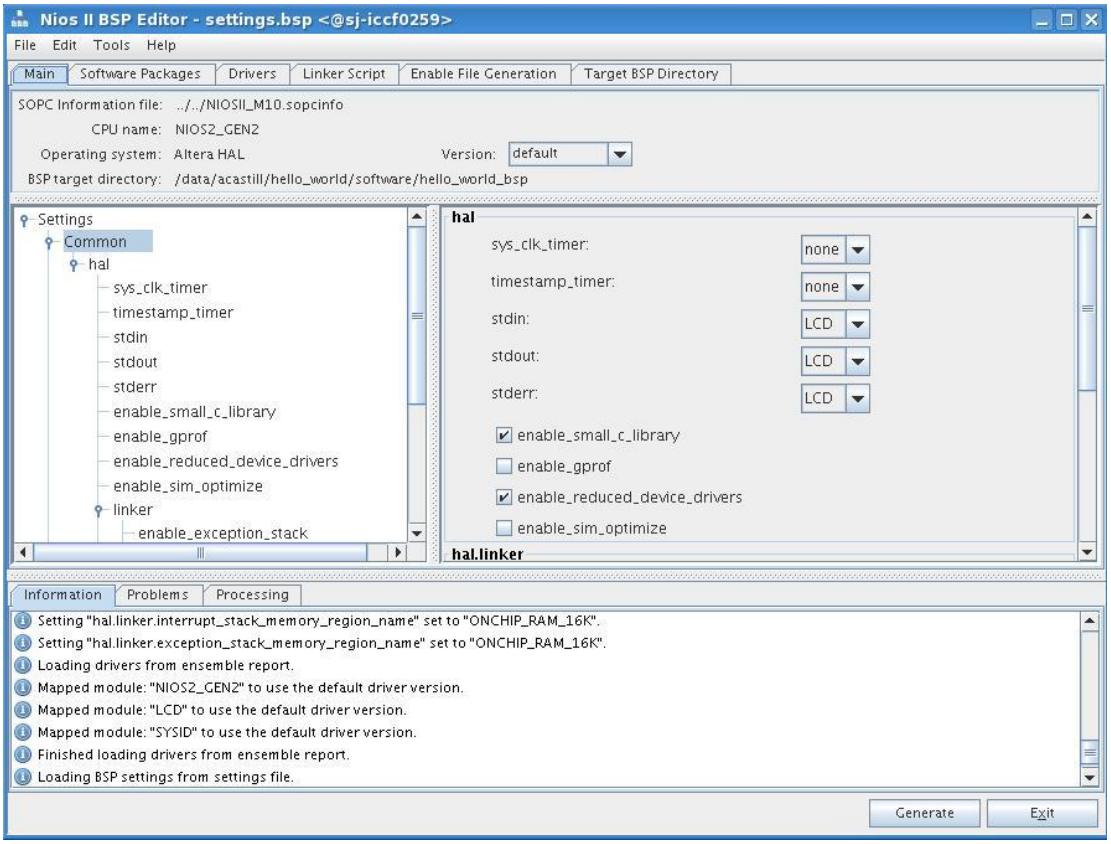


Your project explorer should look similar to the image on the right.

9. Right click on the project's folder and select **Properties**
    a. Under properties, set the Debug Level and Optimization Level to "OFF"



10. Now we will edit the BSP settings for the project:
    a. Right click on the project's BSP folder and navigate to **Properties**
    b. Under the NIOS II BSP properties, set the Debug Level and Optimization Level to "OFF" – hit apply to save the settings.
    c. Click on BSP Editor in the window
    d. Check in enable_small_c_library and enabled_reduced_device_drivers
    e. Hit Generate



11. Compile the project by selecting Build All from the Projects Menu or alternatively Ctrl+B.
    a. It is often better practice to do **Project → Clean** to refresh and build your files.
    b. Once successfully built, a .elf file should appear under the project folder
12. Program the FPGA device by launching Quartus II programmer from the menu and downloading top.sof found in the output folder of the project.
13. Load the executable to the processor by right clicking on the project folder and selecting
    Run → Run Configurations
14. Double click on Nios II Hardware, and new configuration opens on the right pane.
    a. Make sure you select the project name as hello_world and .elf file as hello_world.elf

15. In the Run Configurations page under the Target connections tab.

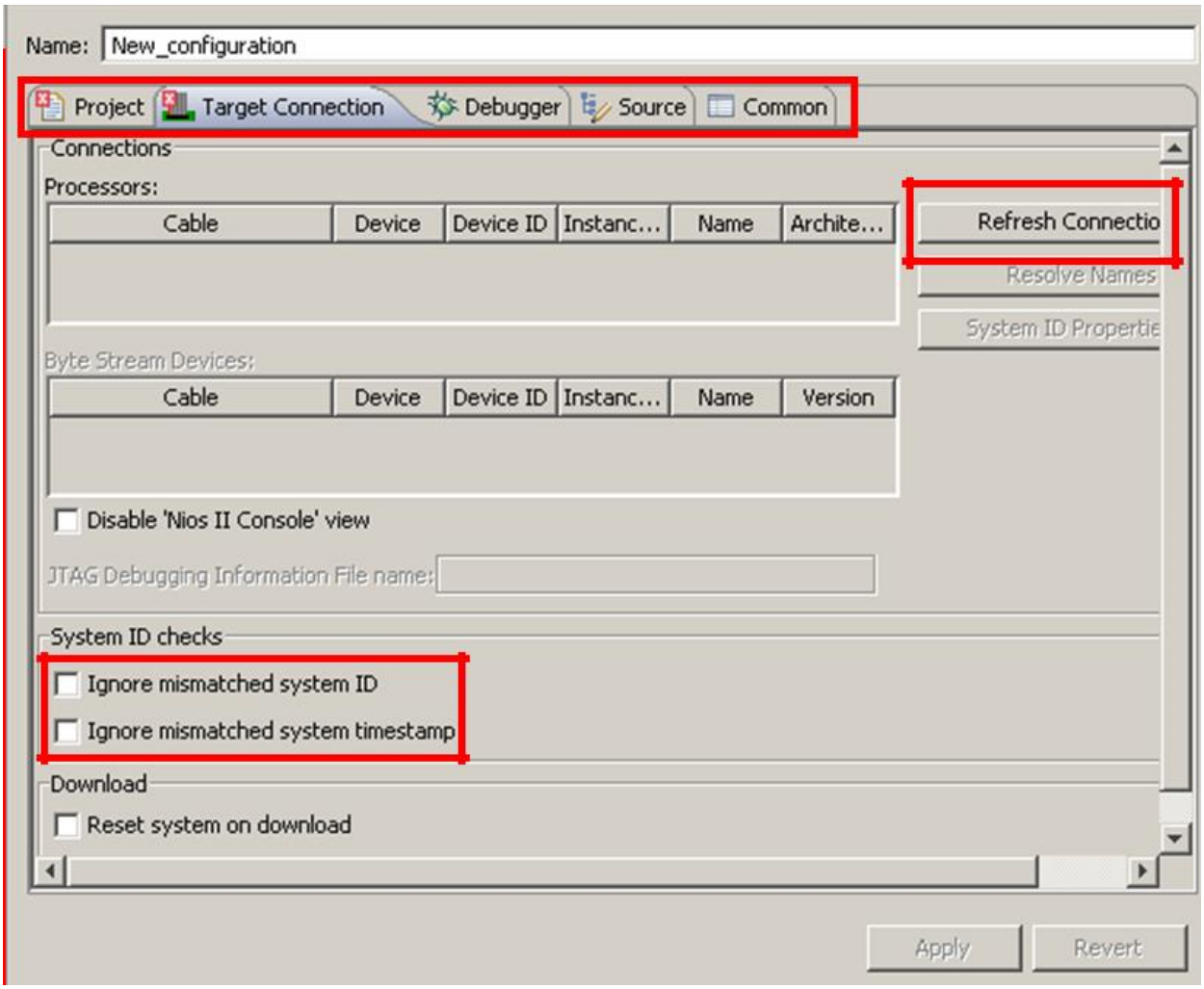
Check the options:

i. **Ignore mismatched system ID**

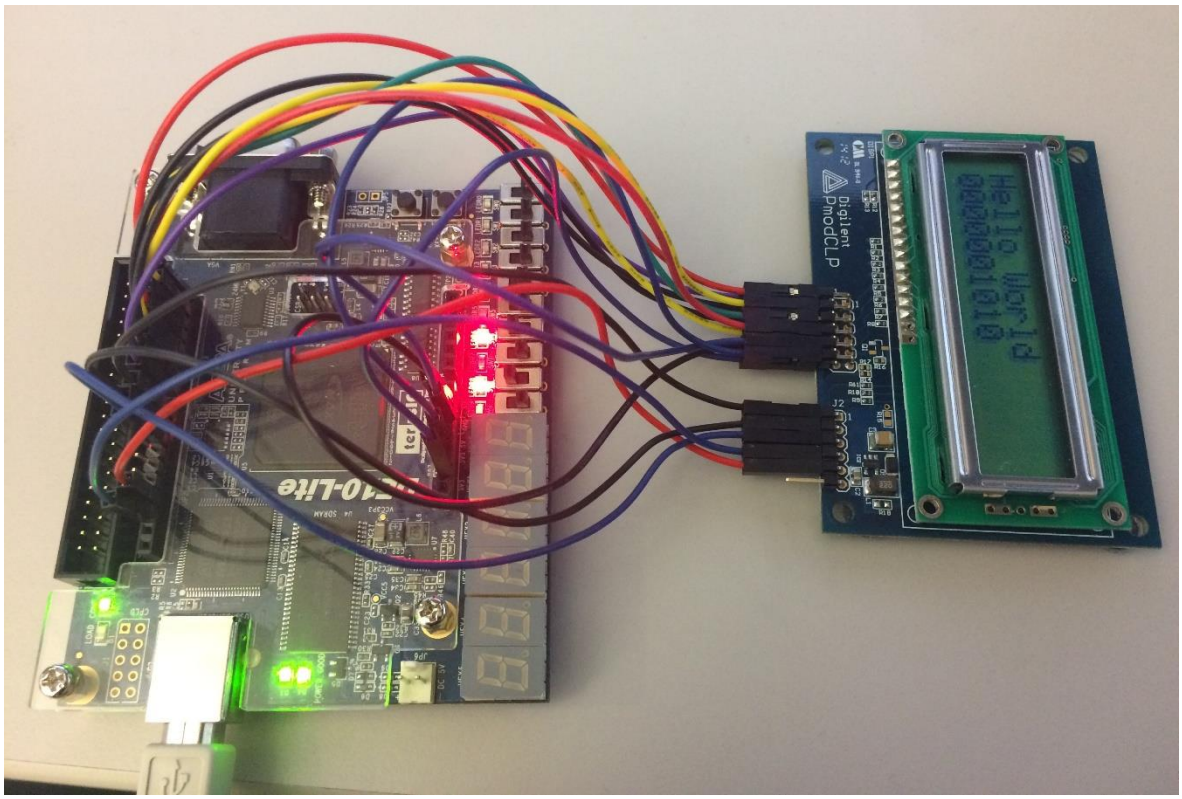
ii. **Ignore mismatched system timestamp.**

b. If you don't see the connection specified, click Refresh Connections.

c. Click Apply followed by Run.



You should see something like this.



16. Now your hardware image is downloaded to the FPGA through the Quartus programmer and the NIOS code has also been downloaded through Eclipse. You can proceed to the demonstration as discussed in the earlier steps.

The method described works well if you are modifying your code – the .sof / .pof file can remain on the board, but you may re-download the .elf file as many times to check the output of your software.

Merging the NIOS executable into the FPGA configuration file

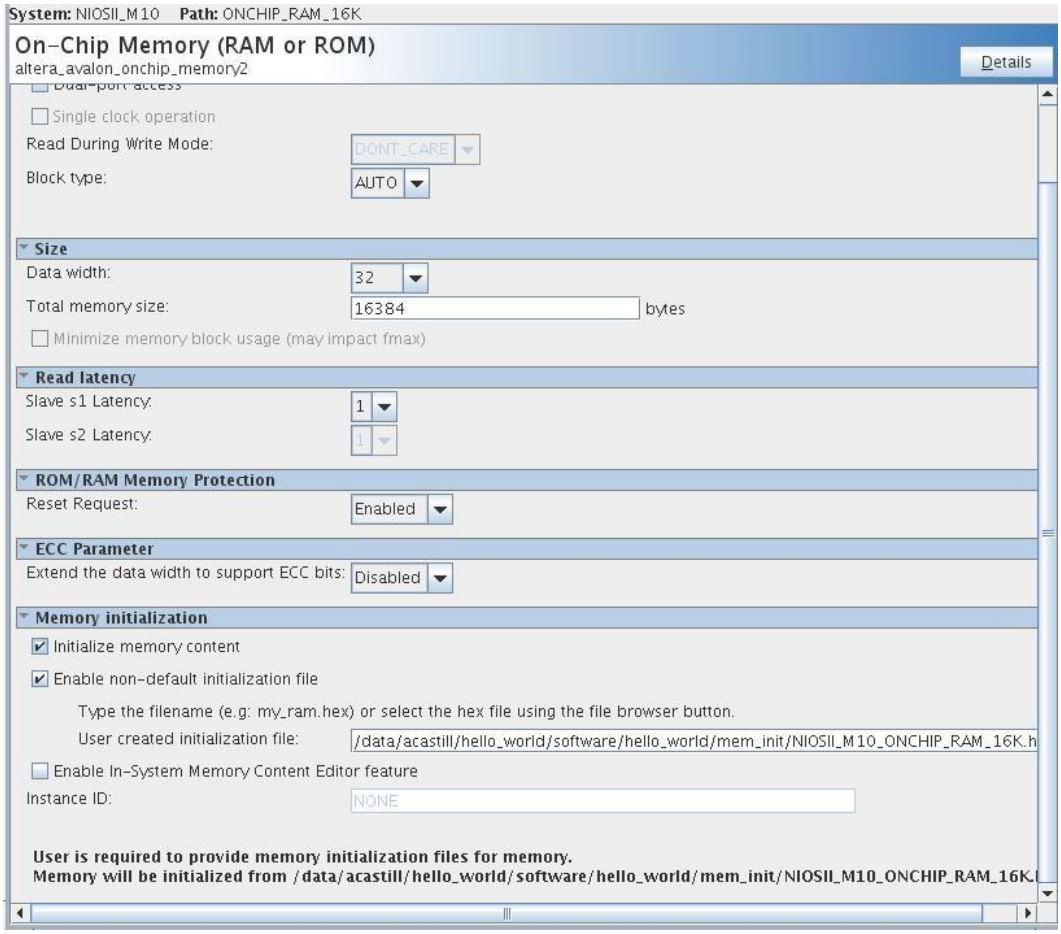
The master_image directory shipped with the design example .sof and .pof files have the NIOS executable incorporated in them. The prior steps detailing how to build your hardware and 21 software show you how to compile the hardware without the image loaded in the .sof or .pof file. In those steps, you load the NIOS executable from Eclipse through the Run → NIOSII Hardware step.

This section will allow you to merge the .elf executable into the .sof once your software is stable. In Eclipse, right click the project and select Make Targets. Click on mem_init_generate. You will generate a .hex file that is in the location where your software build is located (e.g):

<project directory>/mem_init/<name>.hex

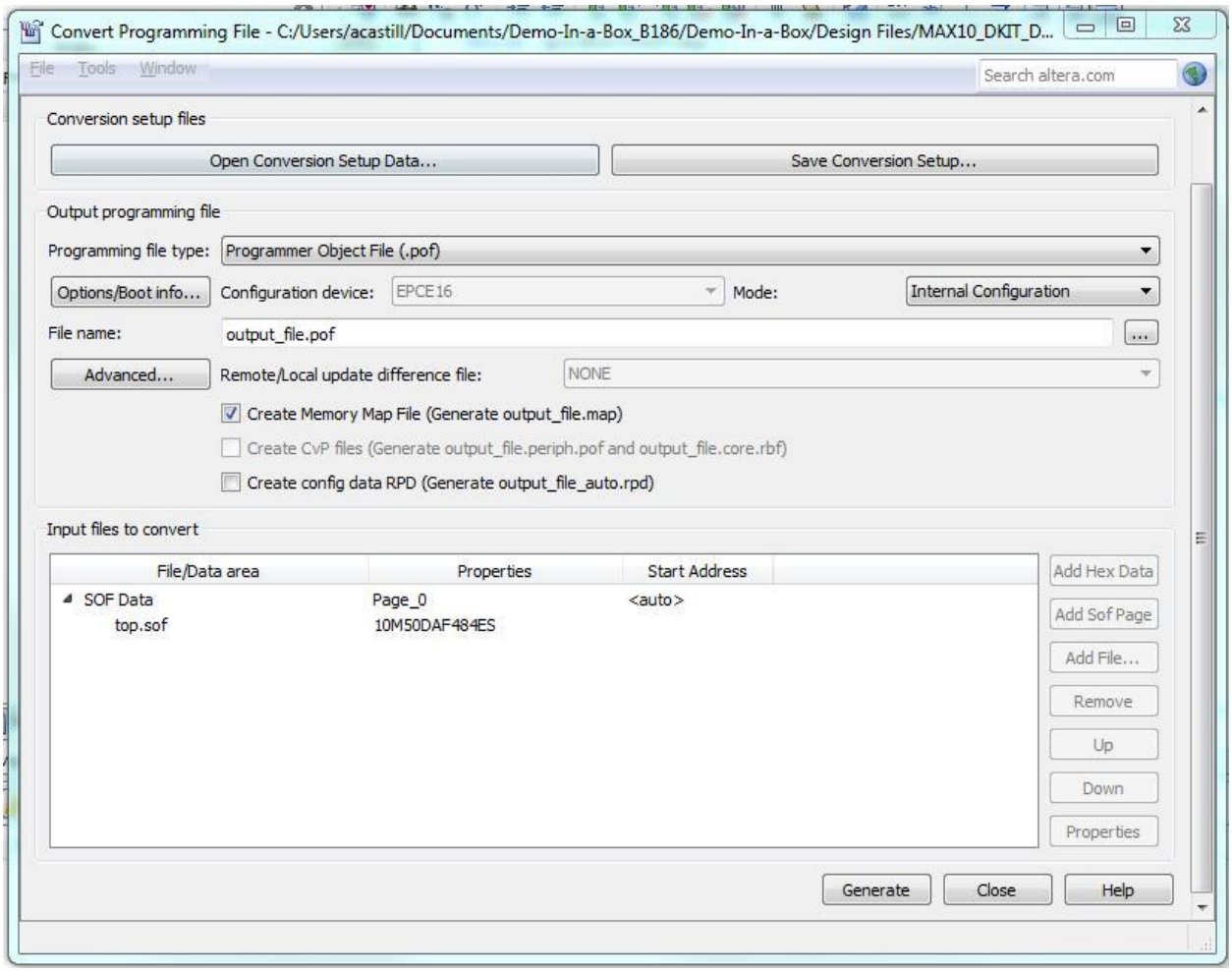
(The name of the .hex file is determined by the settings in your mem_init file under the BSP directory.)

Next, you need to return to Qsys and change the ONCHIP_RAM component by double clicking it. The initialize memory content and enable non-default initialization file needs to on. Enter the location where the .hex file is located.



Next, click Generate HDL → Generate. Return to Quartus and click compile and you will generate a new .sof file with the NIOS executable included. Run the programmer and download the new .sof. Run the demonstration as described in the previous steps.

To generate a .pof file for non-volatile demonstrations, in Quartus run File → Convert Programming Files. Change mode to Internal Configuration. Highlight SOF data, click add file and select output_files/top.sof.



Click Generate, and you will see the output_files/top.pof file to download with the programmer. The demonstration will continue to run even after power cycling the development kit