HowTo: Zedoled Controller

Introduction

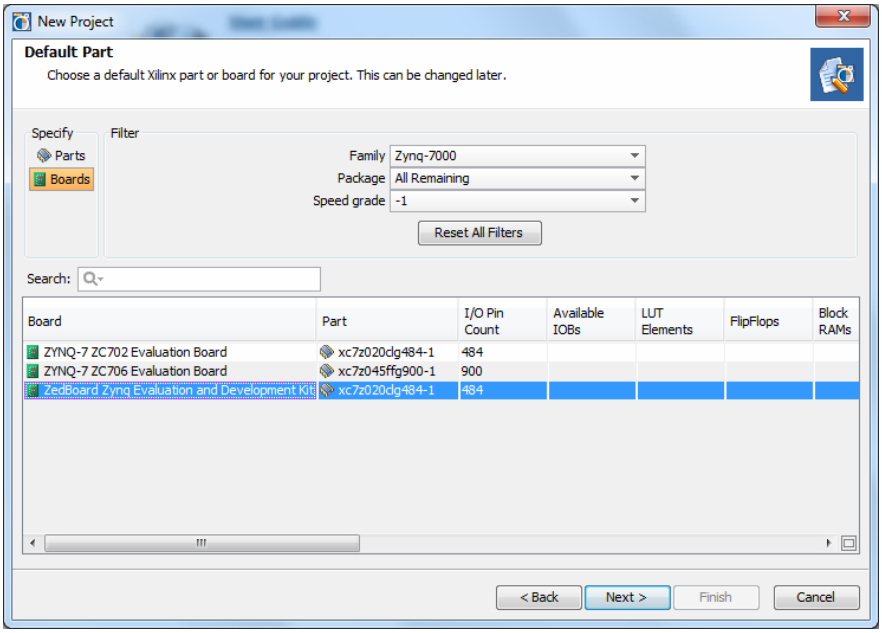
This document is a tutorial. It describes how to create and add a Zedoled Controller IP in a design and how to use software to control it.
Details about the Zedoled Controller can be found in the Reference Manual.
Requirements:

- ISE Design Suite 14.4, with at least a webpack license (free)
- Zedboard
- Zedoled source files

Note: for ISE versions older than 14.4, please read tutorial by ZynqGeek: http://zedboard.org/content/zedboard-create-planahead-project-embedded-processor

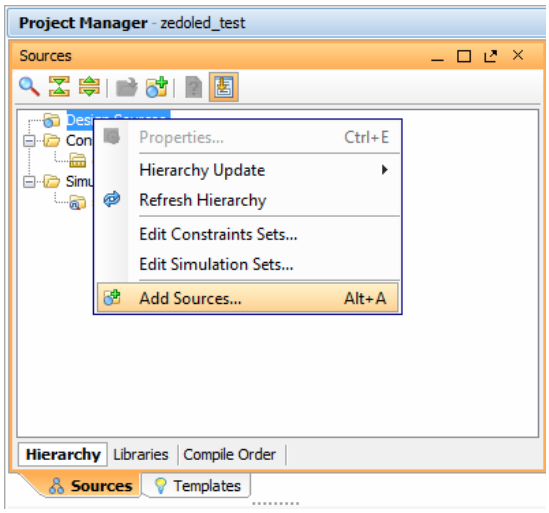
Create a new Project

Launch PlanAhead and select "Create New Project".
Choose a name and a directory, and then click next until the "Default Part" selection.
Select Zedboard part and click next and finish.



Now our project is created, we want to add our embedded processor to the design.

Add a source => Add or Create Embedded source => Create Sub-Design. When clicking "Finish", XPS starts.

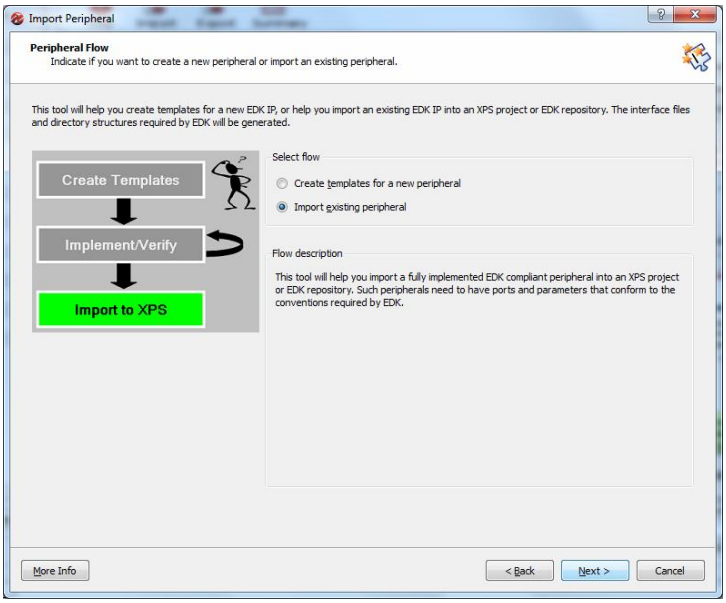


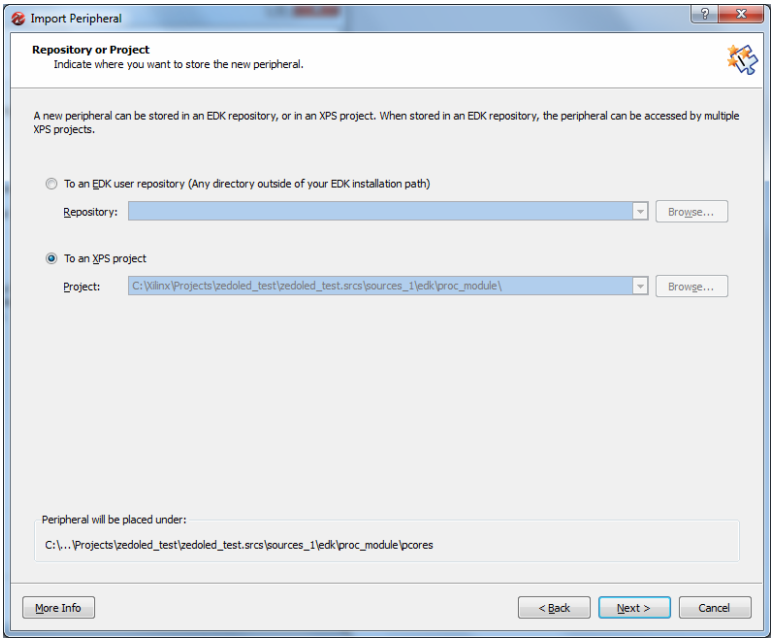
Note: if you have a message box with a license error, just close the box. XPS will start anyway.

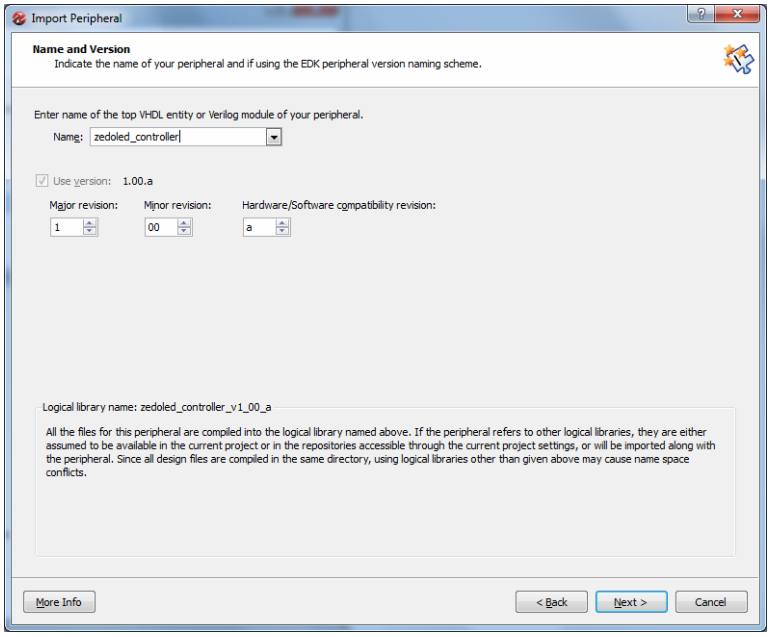
XPS proposes to create a Base System. Click Yes. Follow default settings except Peripheral Configuration: remove everything in "Included Peripherals" list.

Create a new IP

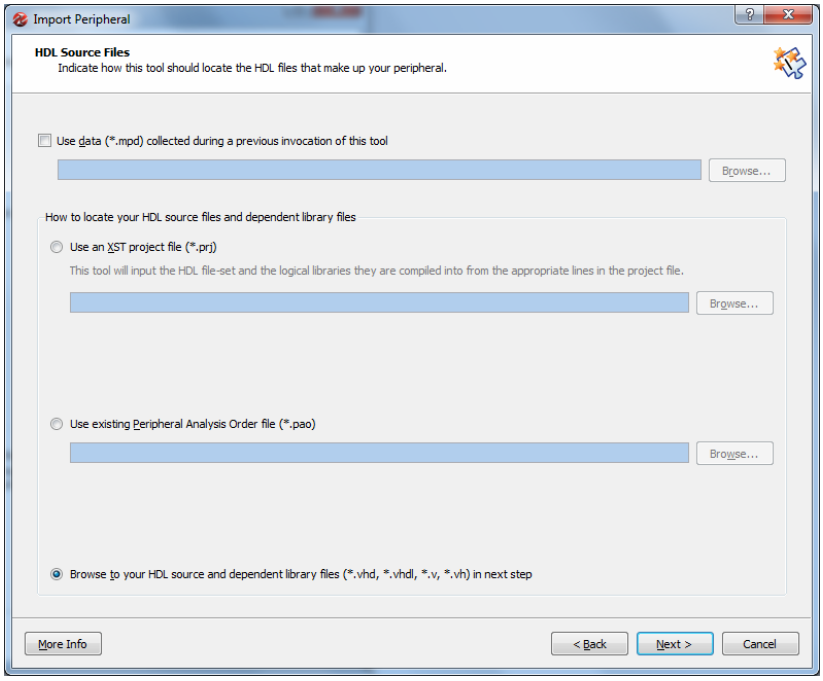
Launch Create and Import Peripheral Wizard: Hardware=>Create or Import Peripheral… Select "Import Peripheral" and name it zedoled_controller.



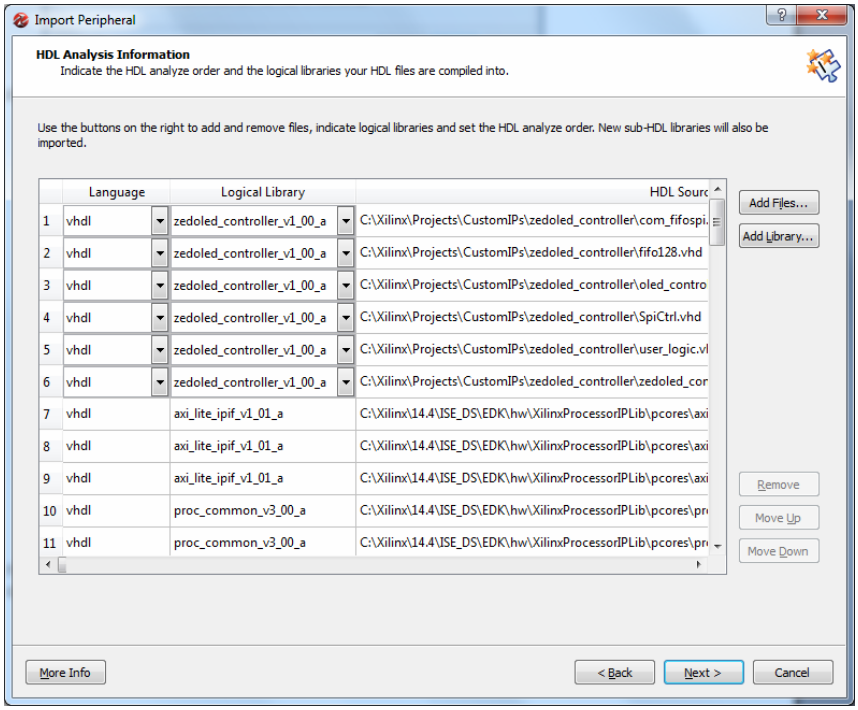

Import Peripheral
Repository or Project
Indicate where you want to store the new peripheral.
A new peripheral can be stored in an EDK repository, or in an XPS project. When stored in an EDK repository, the peripheral can be accessed by multiple XPS projects.
To an EDK user repository (Any directory outside of your EDK installation path)
Repository:
Browse...
To an XPS project
Project:
C:\Xilinx\Projects\zedoled_test\zedoled_test.srcs\sources_1\edk\proc_module\
Browse...
Peripheral will be placed under:
C:\...\Projects\zedoled_test\zedoled_test.srcs\sources_1\edk\proc_module\pcores
More Info
< Back
Next >
Cancel


Import Peripheral
Name and Version
Indicate the name of your peripheral and if using the EDK peripheral version naming scheme.
Enter name of the top VHDL entity or Verilog module of your peripheral.
Name:
zedoled_controller
Use version:
1.00.a
Major revision:
Minor revision:
Hardware/Software compatibility revision:
1
00
a
Logical library name: zedoled_controller_v1_00_a
All the files for this peripheral are compiled into the logical library named above. If the peripheral refers to other logical libraries, they are either assumed to be available in the current project or in the repositories accessible through the current project settings, or will be imported along with the peripheral. Since all design files are compiled in the same directory, using logical libraries other than given above may cause name space conflicts.
More Info
< Back
Next >
Cancel

We will use directly vhdl files to create the IP. So select the third option.

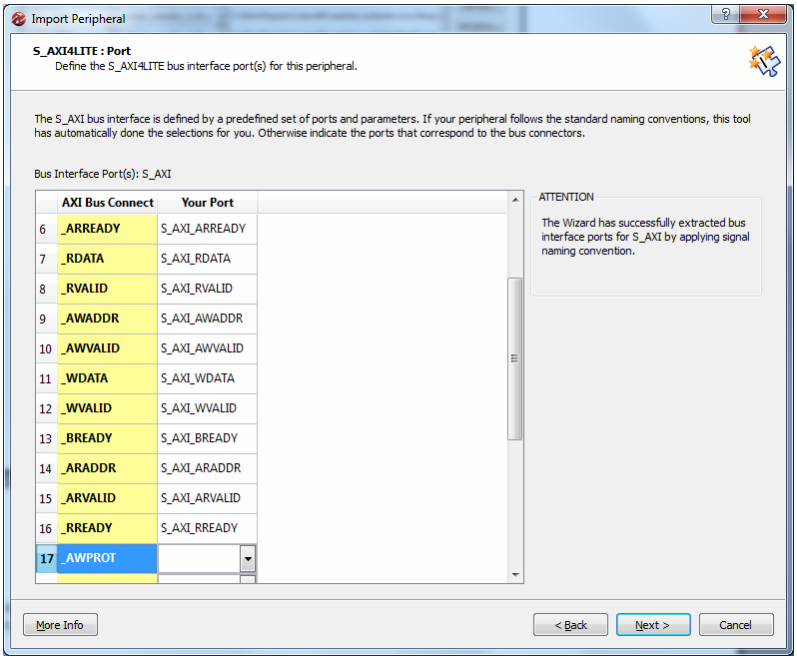


Click "Add Files" and select the 6 files provided with the project (available at https://github.com/vcesson/zedoled_controller.git).
Then, as this peripheral uses the AXI Lite bus to communicate with the processor, we need to add two libraries: axi_lite_ipif_v1_01 and proc_common_v3_00.

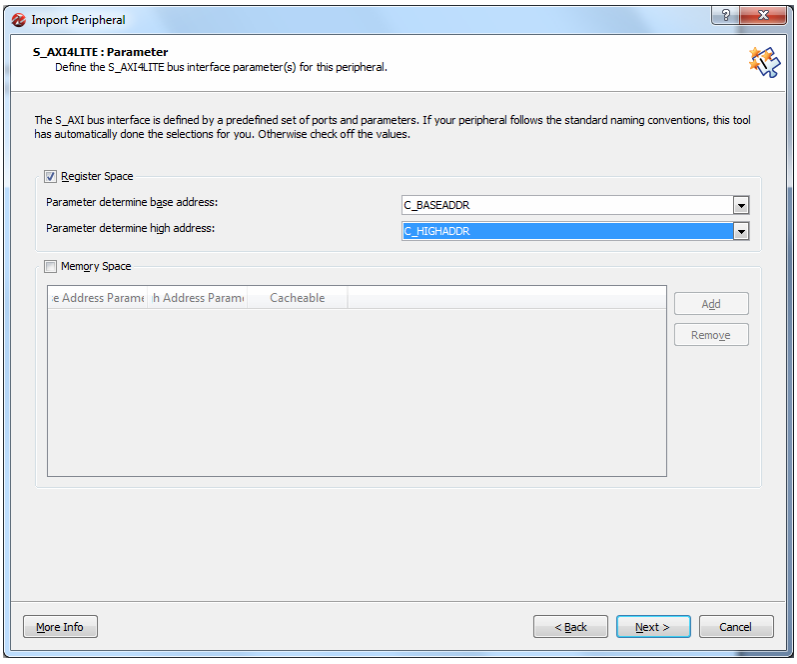


If an error occurs when clicking “Next”, be sure you have added the correct libraries with the correct version.

On the next page, select the “AXI4Lite” bus interface, in slave mode.
When clicking next again, the wizard should connect automatically the signals.

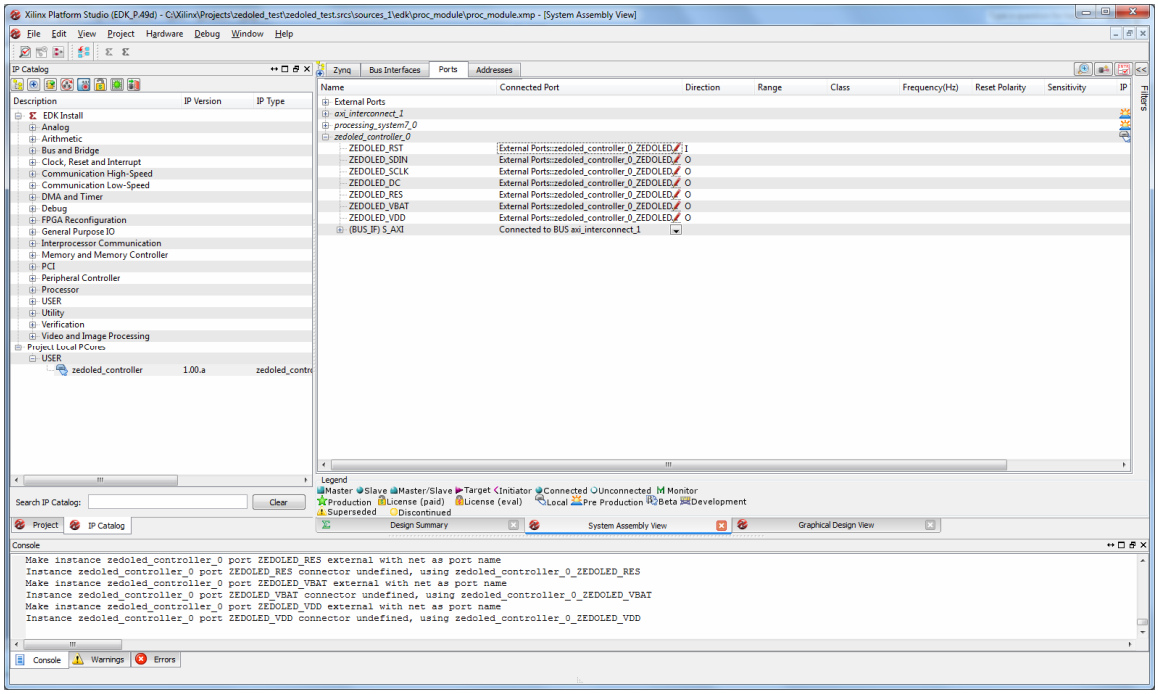


On next page, check “Register Space” box and select “C_BASEADDRESS” for base address and “C_HIGHADDRESS” for high address.



Click next until finished.

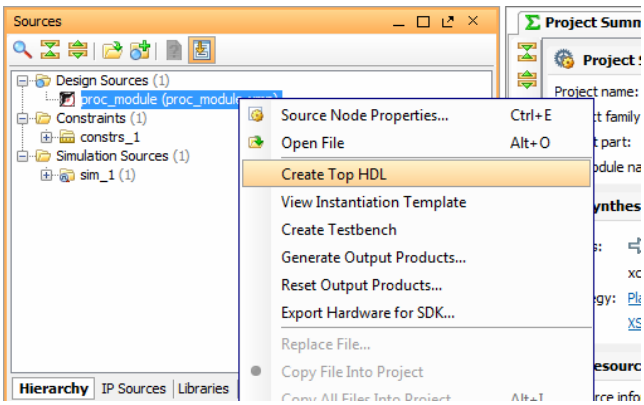
Now our new IP is available in the IP Catalog, under Local PCores=>USER. Double click on zedoled_controller to add an instance of this IP to the design and keep the default settings on the two windows which pop up.
Go to "Ports" tab and make port connections of zedoled_controller to external.



The IP is now successfully added to the design.
Close XPS.

Implement design

Add the constraint file provided in the project: Add sources…=> Add or Create Constraint => Add files…
Note: if the IP instance name is not zedoled_controller_0, the UCF file must be changed.
Now we want to generate all the necessary wrapped files from our design. Right click on "proc_module" => Create TOP HDL.



Once it is done, run implementation and generate the bitstream.

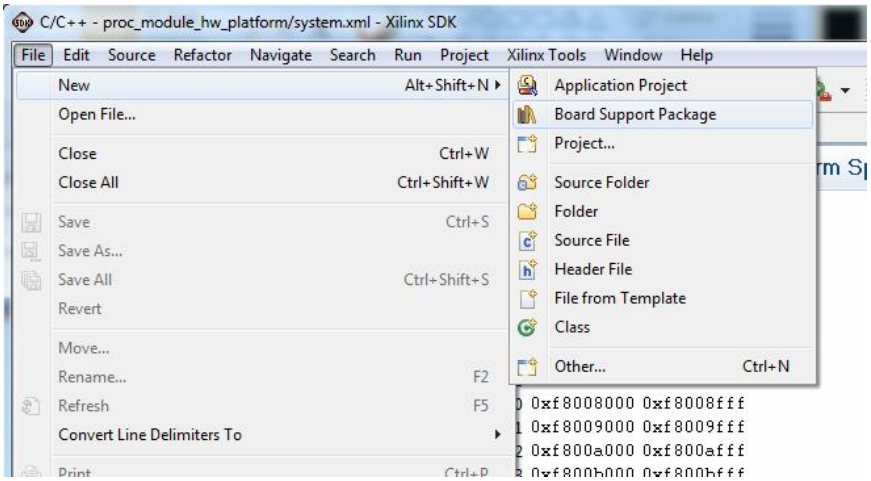
Note: there may be 3 critical errors. Those errors are not important.

Export to SDK

Once the bitstream is generated, we can export the project to SDK to develop the software:
File=>Export=>Export Hardware for SDK
Check all 3 boxes: include bitstream, Export Hardware and Launch SDK.

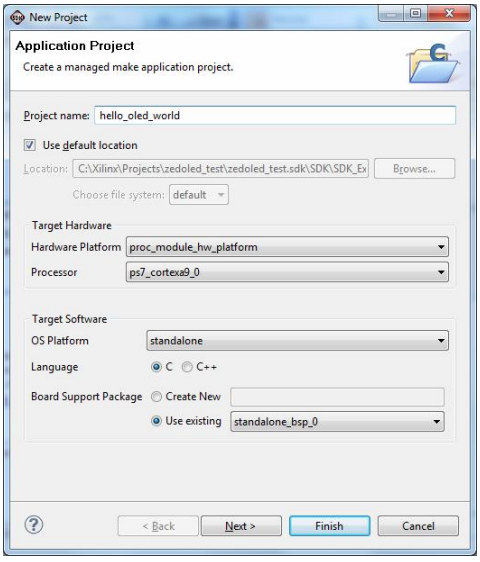
Note: if "include bitstream" is not checkable, first click on "Open Implemented Design" in the Flow Navigator. After Implemented Design is opened, export the project to SDK.

First thing to do in SDK is to create the BSP. It creates the good environment (libraries) corresponding to our device and allows us to develop software easily. Go to File=>New=>Board Support Package, and keep default settings.



Wait until compiling is finished.

Then, create an application project, name it as you want, and select the BSP we have just created.



Click next and Select a “Hello World” type project.

Software

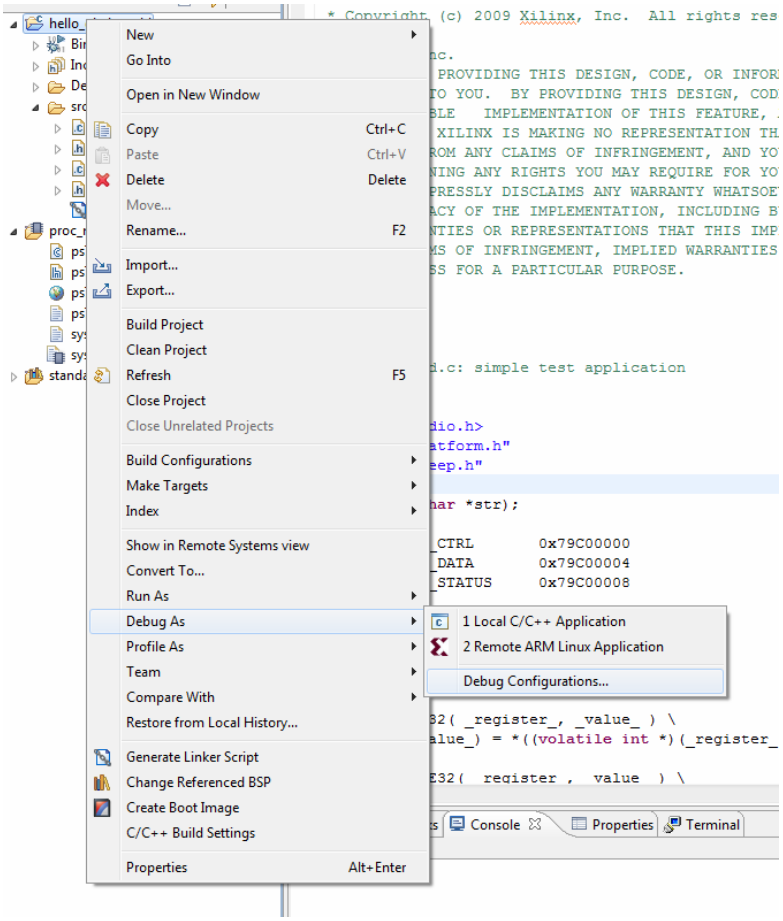
You can test the software like this. Connect the PC to the uart usb plug of the board and use a terminal to see “Hello World!” appears on your screen.

But we want to use the OLED display. Replace the code in helloworld.c file with the code in zedoled.c file provided. This program initializes the Oled and it displays a logo.
You can easily modify the code. Refer to the SSD1306 document for information about the different commands.

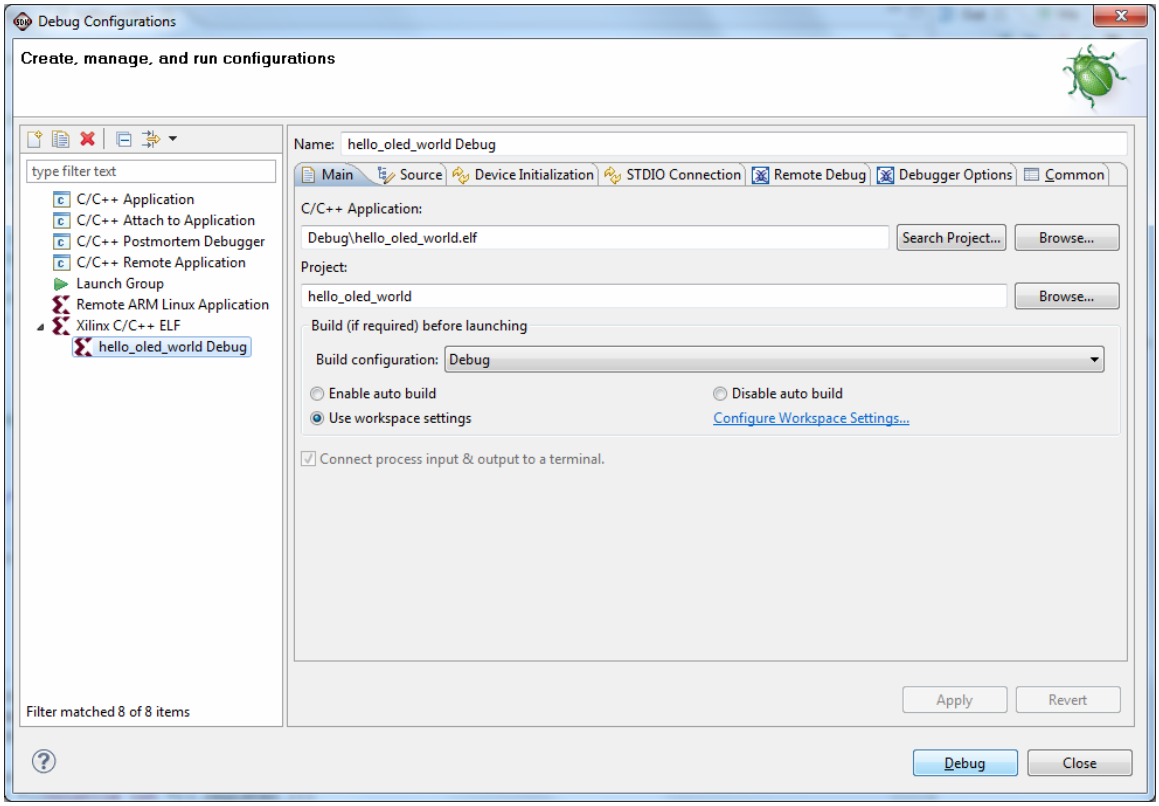
Run on board

The PL is configured with a Xilinx tool. Go to Xilinx Tools=>Program FPGA. The selected bitstream should be \SDK\SDK_Export\proc_module_hw_platform\system.bit
When the PL is configured, the blue led “DONE” turns on.

To run the application, right click on your project and select Debug as=>debug configuration…



Select Xilinx C/C++ ELF, keep default settings and clock on "Debug".



When running the program, you should see the Zedboard logo scrolling on your Oled screen.

Change the logo

A very friendly easy-to-use free software is available at http://en.radzio.dxp.pl/bitmap_converter/ to create a C table from a bmp image. The image should contain just black and white pixels for a better result.
Note: on Zedboard, the display is 128x32 px. But the ram size is actually 128x64. So it is possible to send a 128x64px image and display it entirely thanks to scrolling!