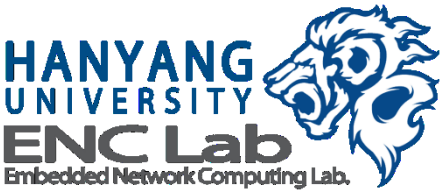
Cosmos OpenSSD+ Tutorial



Revision history

Date	Version	Revision
2016-09-01	1.0	First draft

Table of Contents

Chapter 1: Introduction

1.1. Overview

This document illustrates how to download a prebuilt bitstream to Cosmos OpenSSD and program the Greedy FTL provided as an example FTL.

1.1.1. Materials

The latest technical documents and materials are available at Cosmos OpenSSD wiki http://www.openssd-project.org/wiki/Cosmos_OpenSSD_Technical_Resources

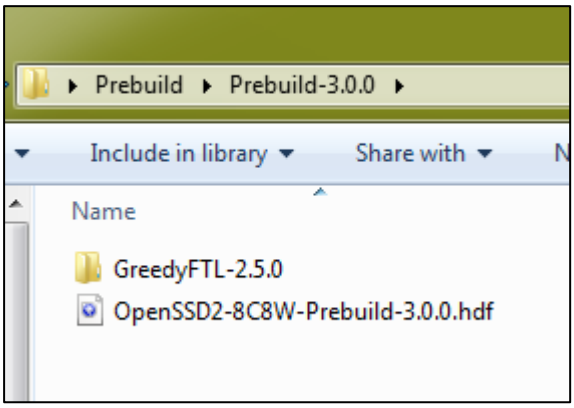
Chapter 2: Get Started

2.1. Required Tools

2.1.1. Xilinx Vivado Design Suite

This document is based on Xilinx Vivado Design Suite: System Edition 2014.4. Especially, Xilinx Software Development Kit (SDK) 2014.4 is required to follow this tutorial.

2.2. Required Files



2.2.1. Xilinx Hardware Description File

Hardware description file (.hdf) is a compressed file which contains a bitstream file (.bit) and multiple source codes for initializing Zynq processing system (PS). A board support package is generated by importing the hardware description file to Xilinx Vivado SDK. The hardware description file is also available at Cosmos OpenSSD wiki.

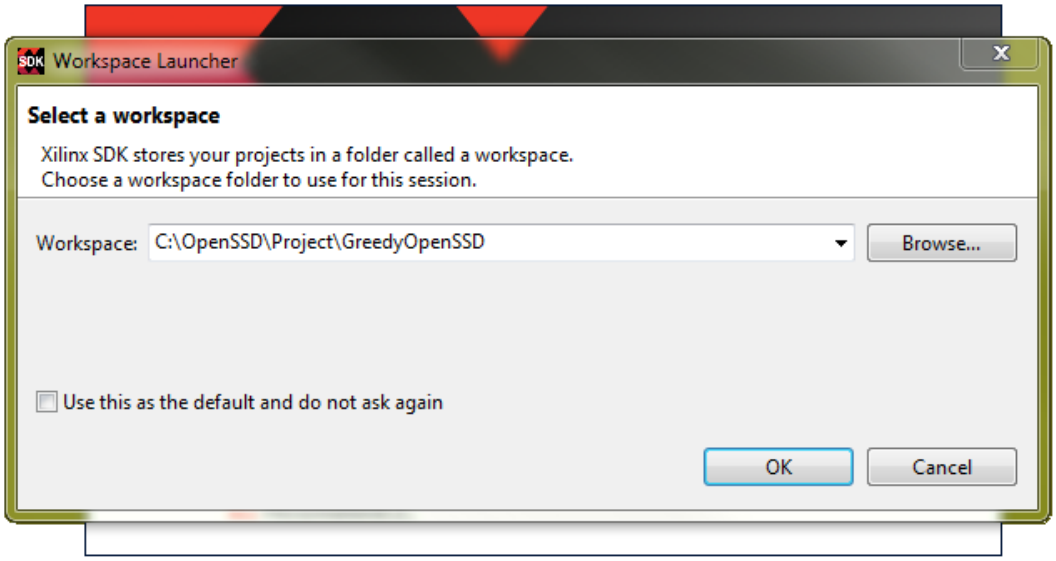
2.2.2. Firmware (flash translation layer) source

The latest Cosmos OpenSSD firmware contains a flash translation layer and a host interface logic to be operated as a solid-state drive. The FTL has a software-level NAND flash scheduler and supports greedy garbage collection mechanism. In addition, it is responsible for managing the NVMe controller.

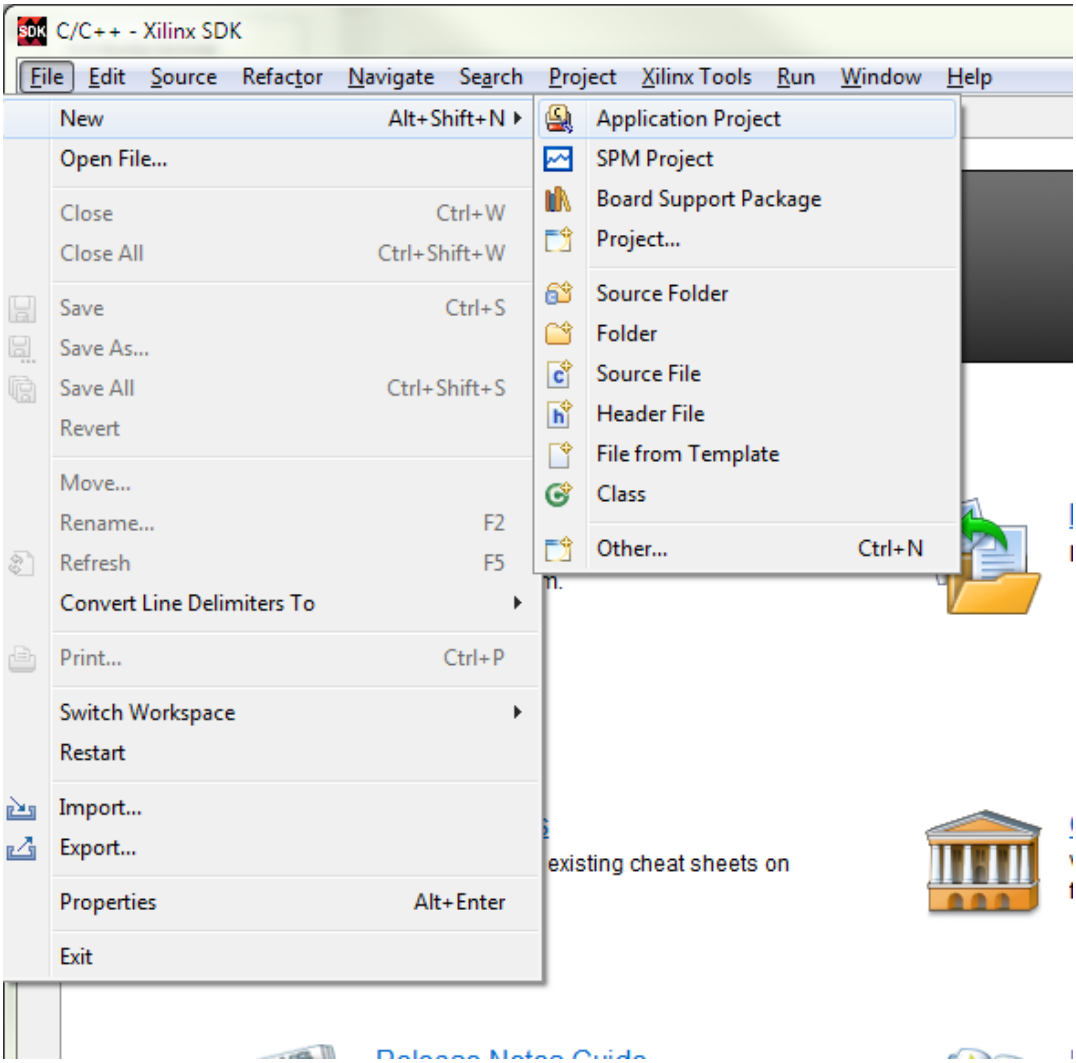
Chapter 3: Deploying Steps

3.1. Download required files and create a workspace directory

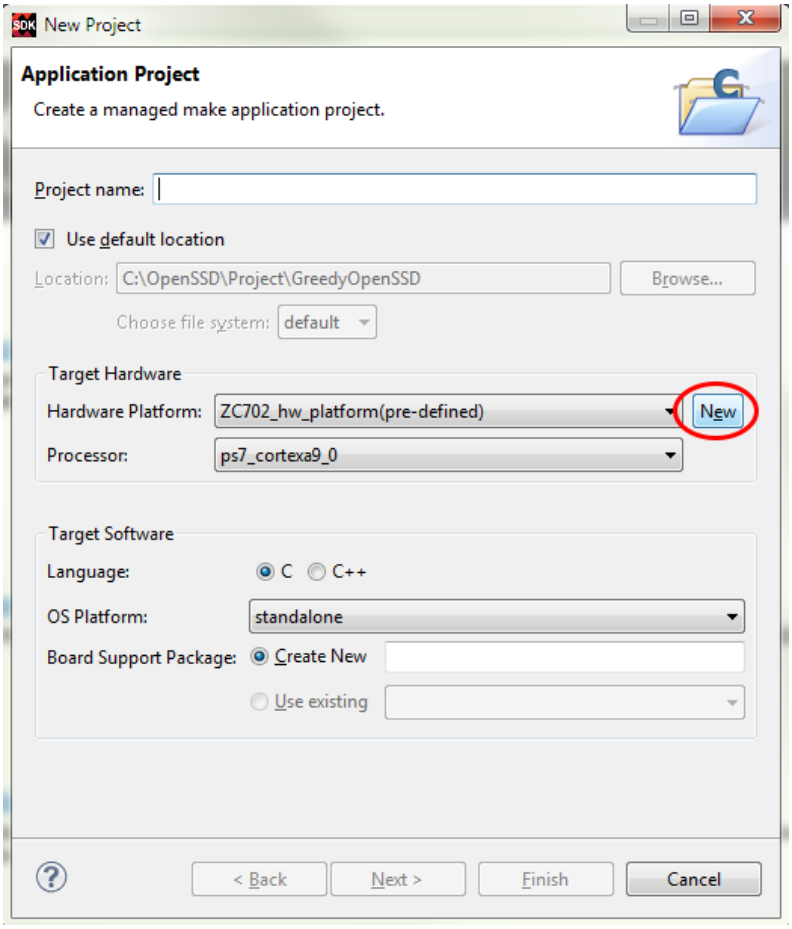
3.2. Launch Xilinx Software Development Kit and designate the workspace



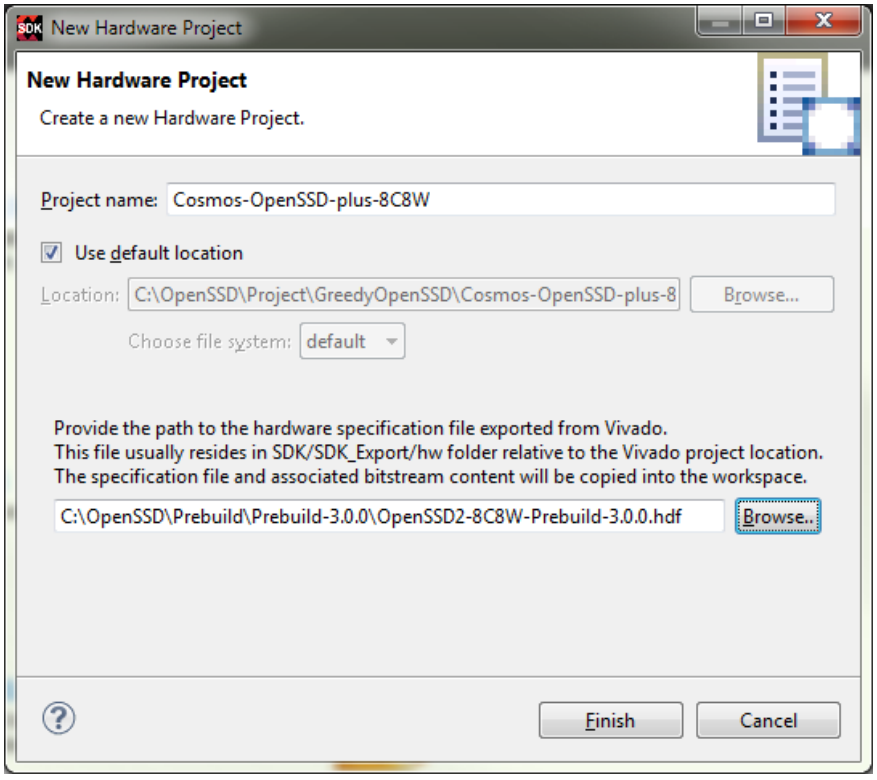
3.3. Go to File->Application Project



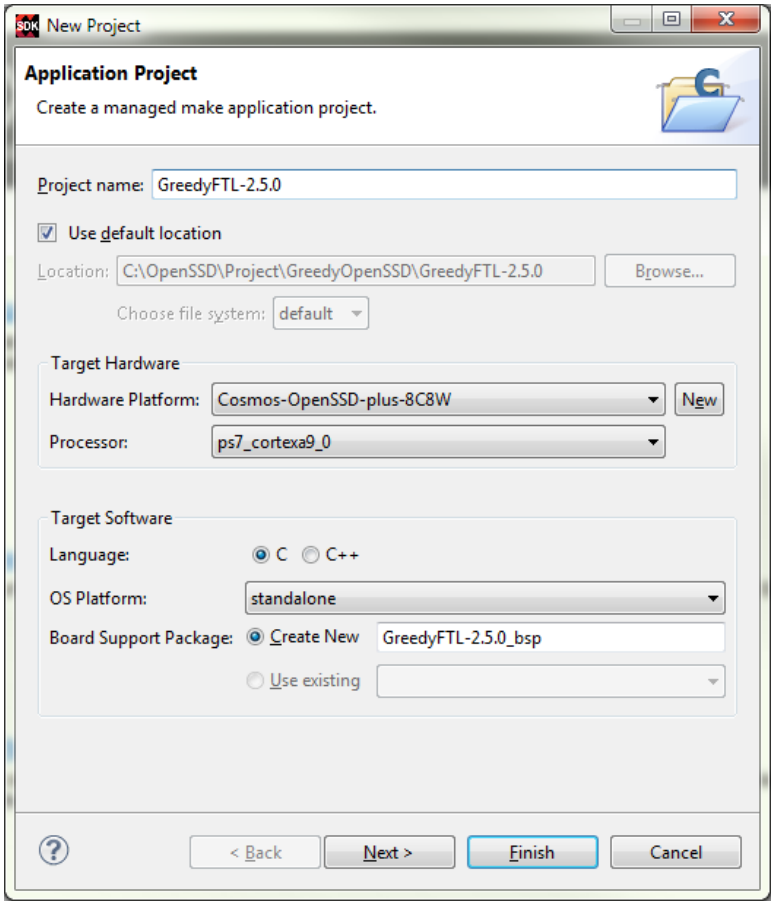
3.4. Press "New" to register the hardware description file (HDF)

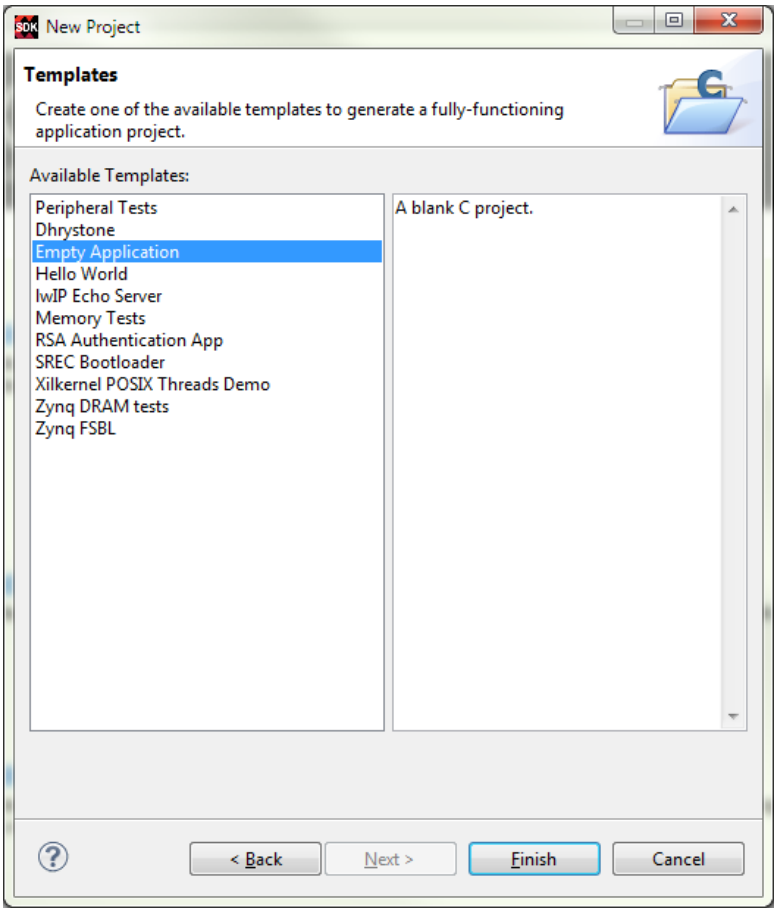


3.5. Name the hardware project and specify a path of the HDF

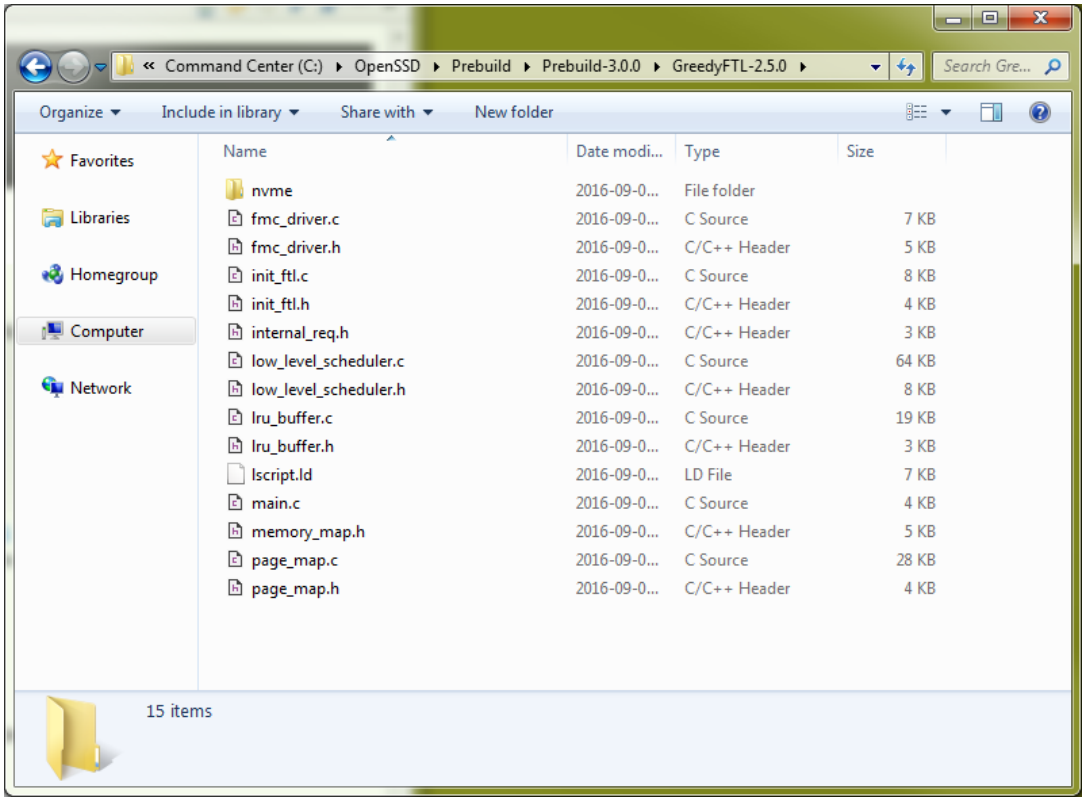


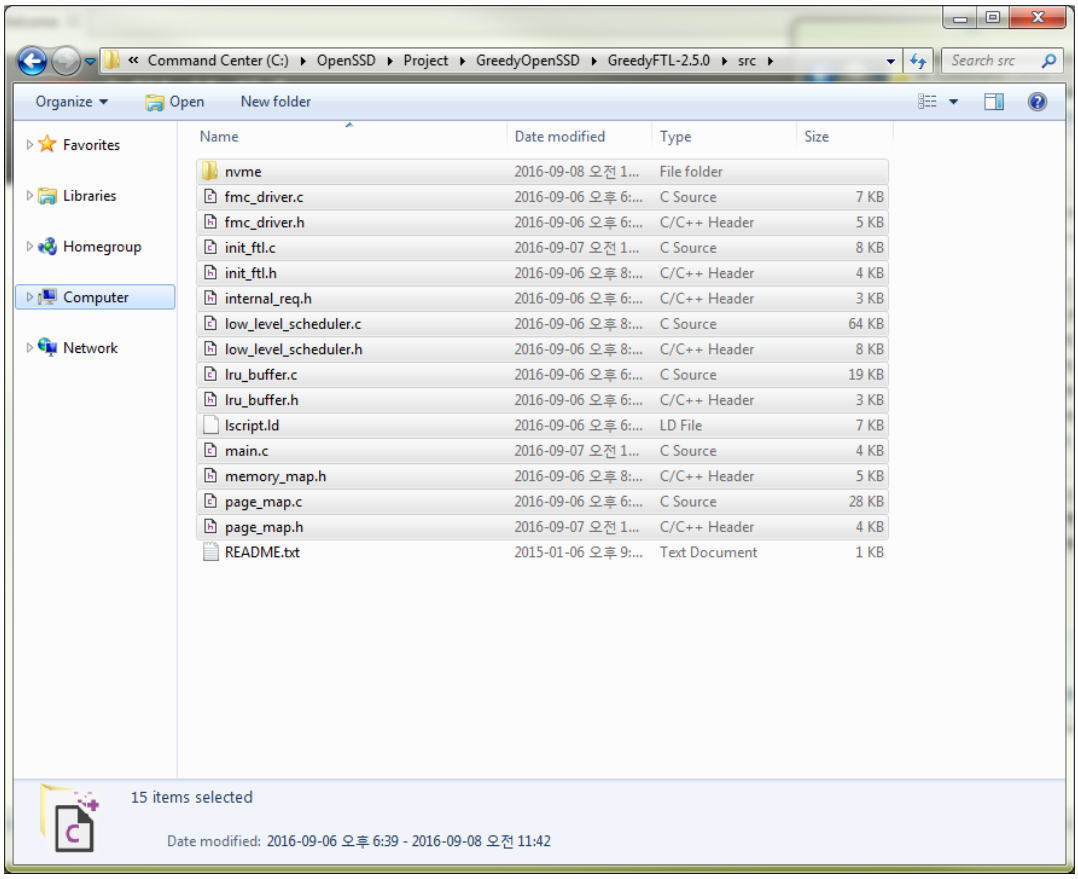
3.6. Name the application project and finish this template wizard



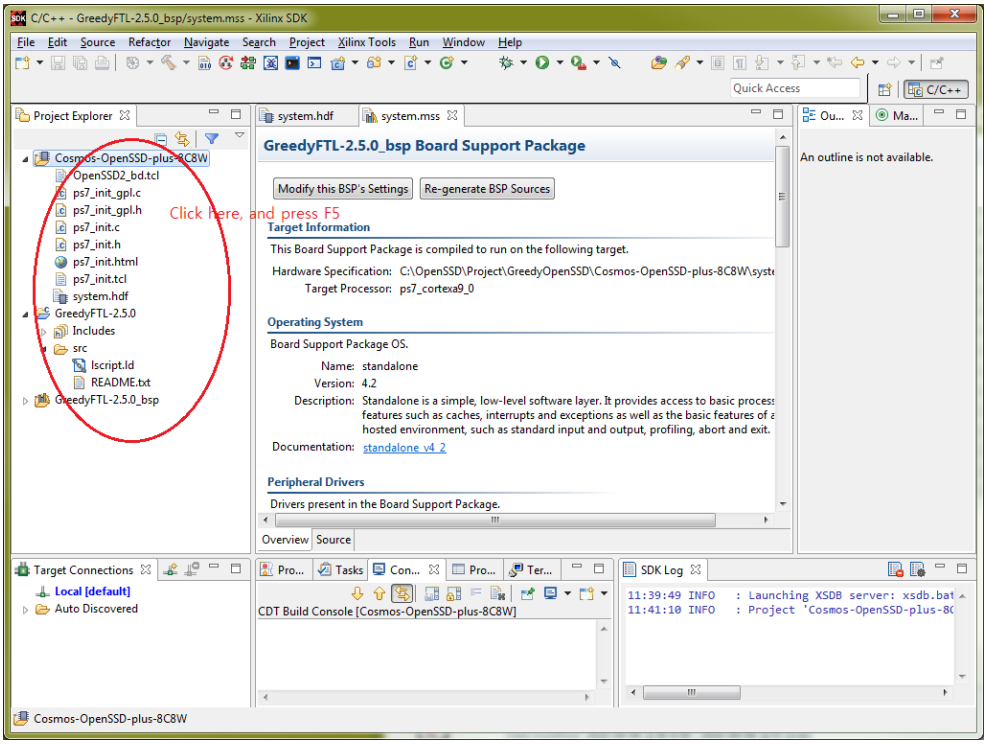


3.7. Locate “src” directory and copy greedy FTL source files to it

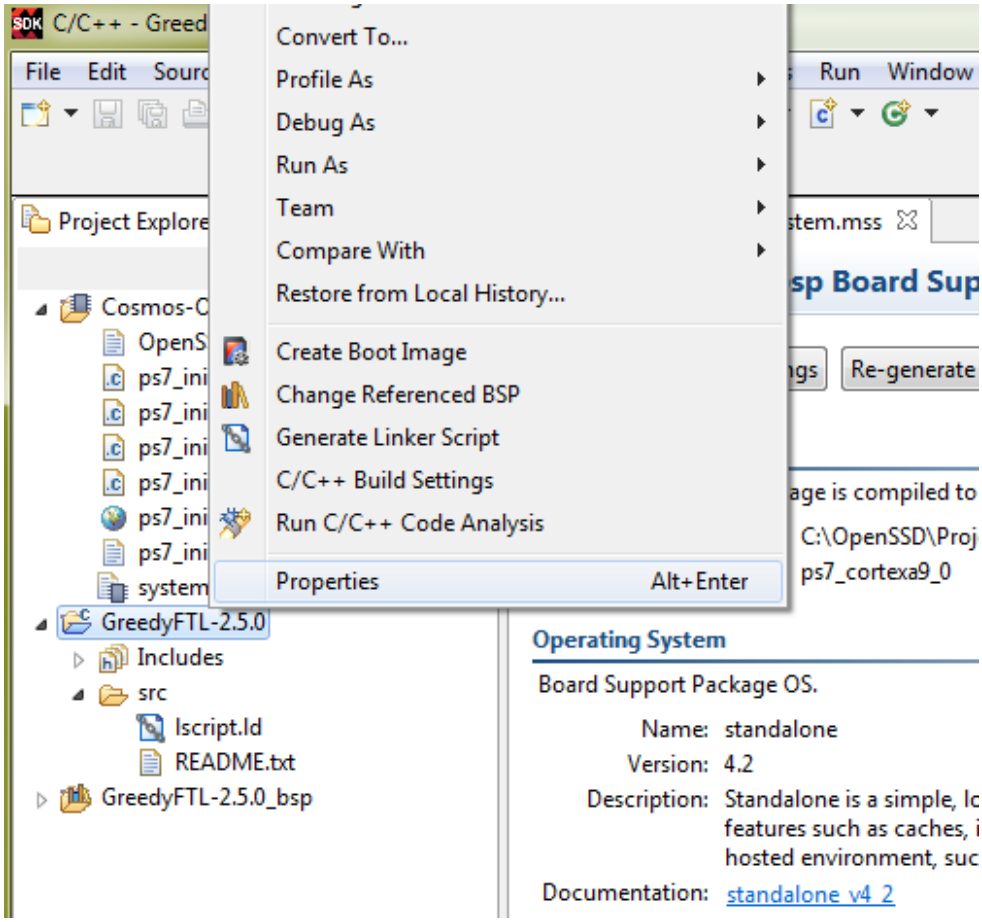




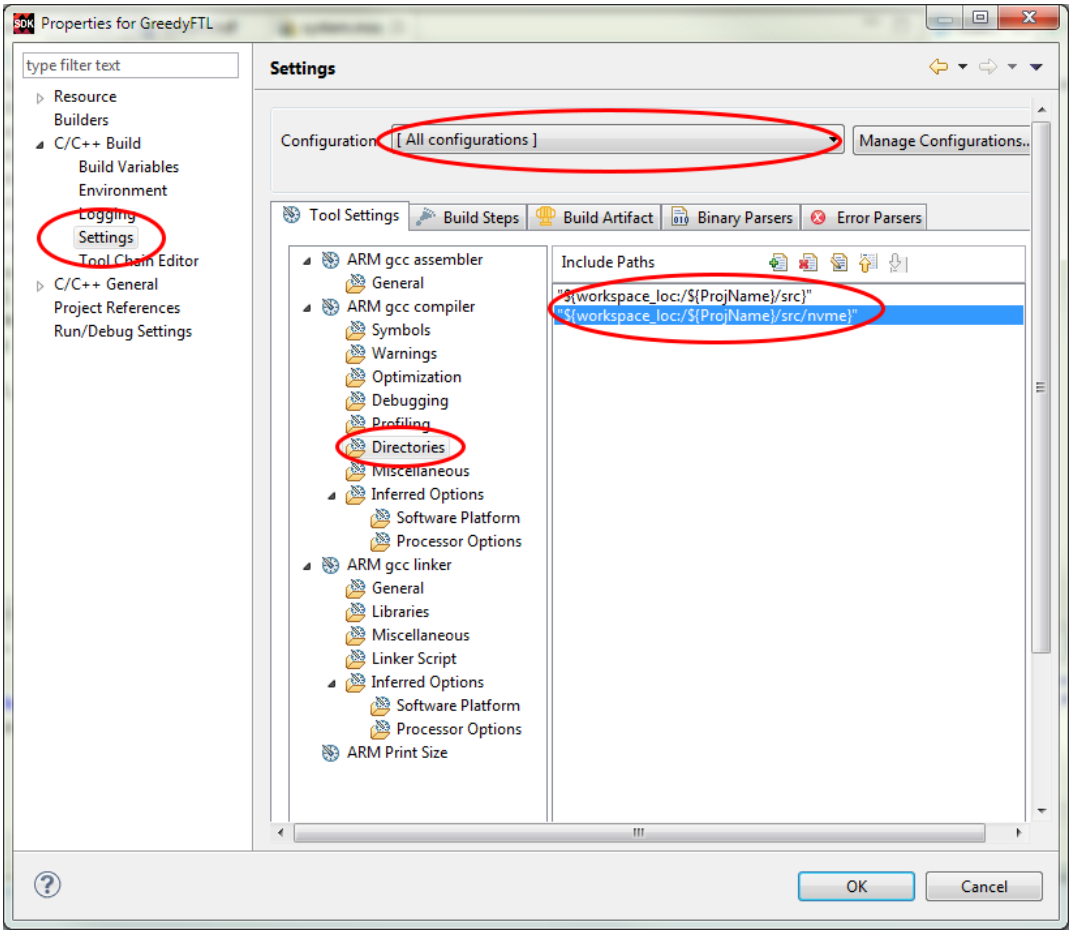
3.8. Get back to SDK and refresh the project explorer



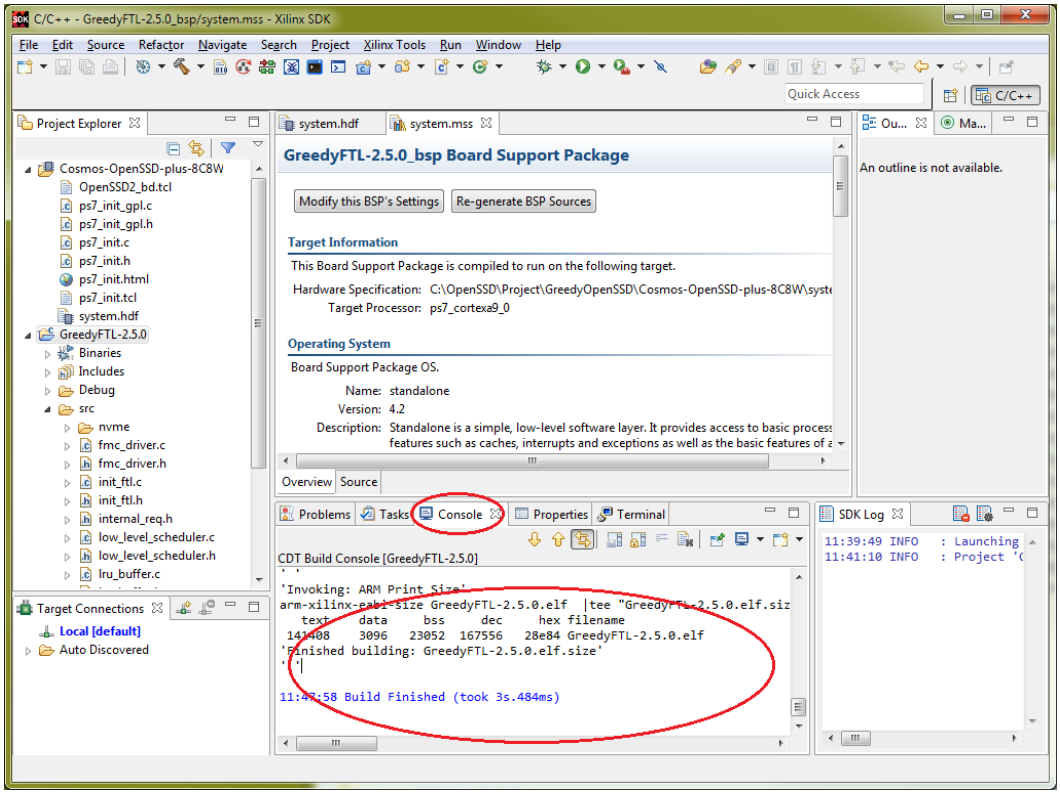
3.9. Go to "properties" menu



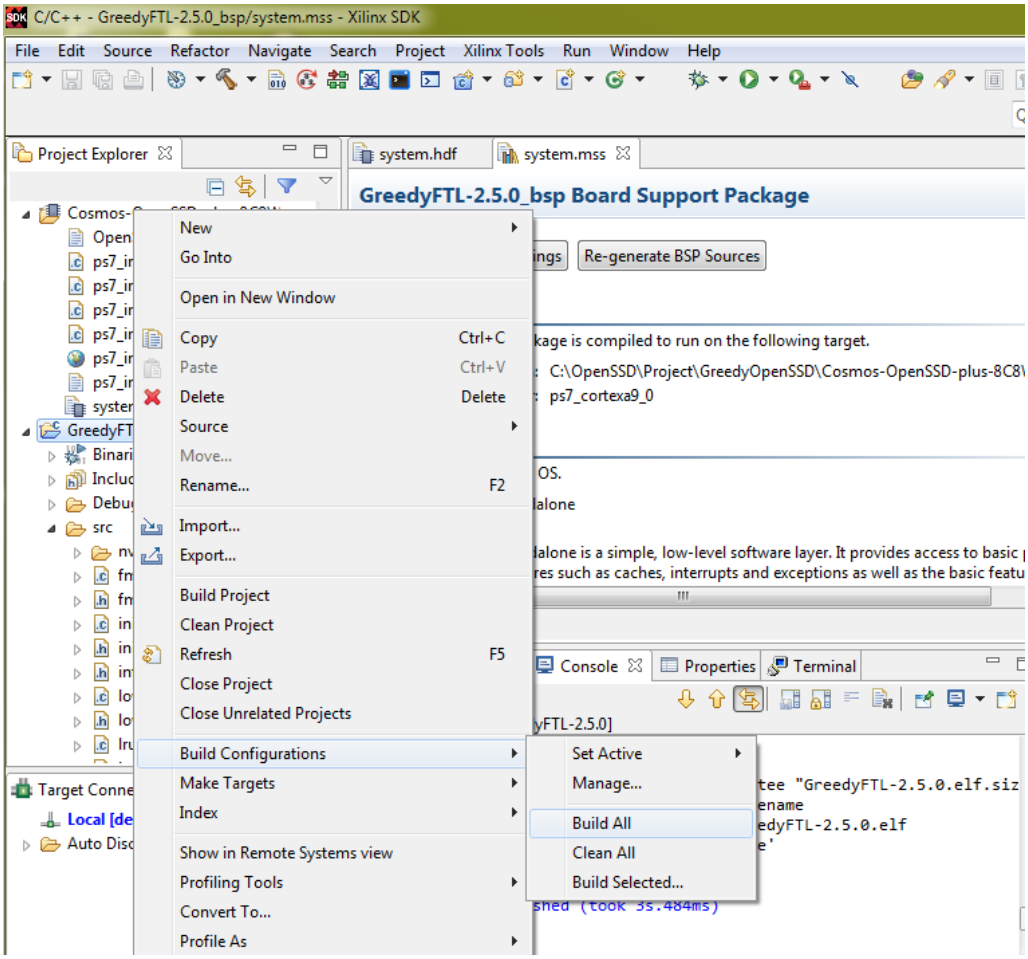
3.10. Go to C/C++ Build->Settings and specify include paths



3.11. If everything goes well, the build process should finish successfully

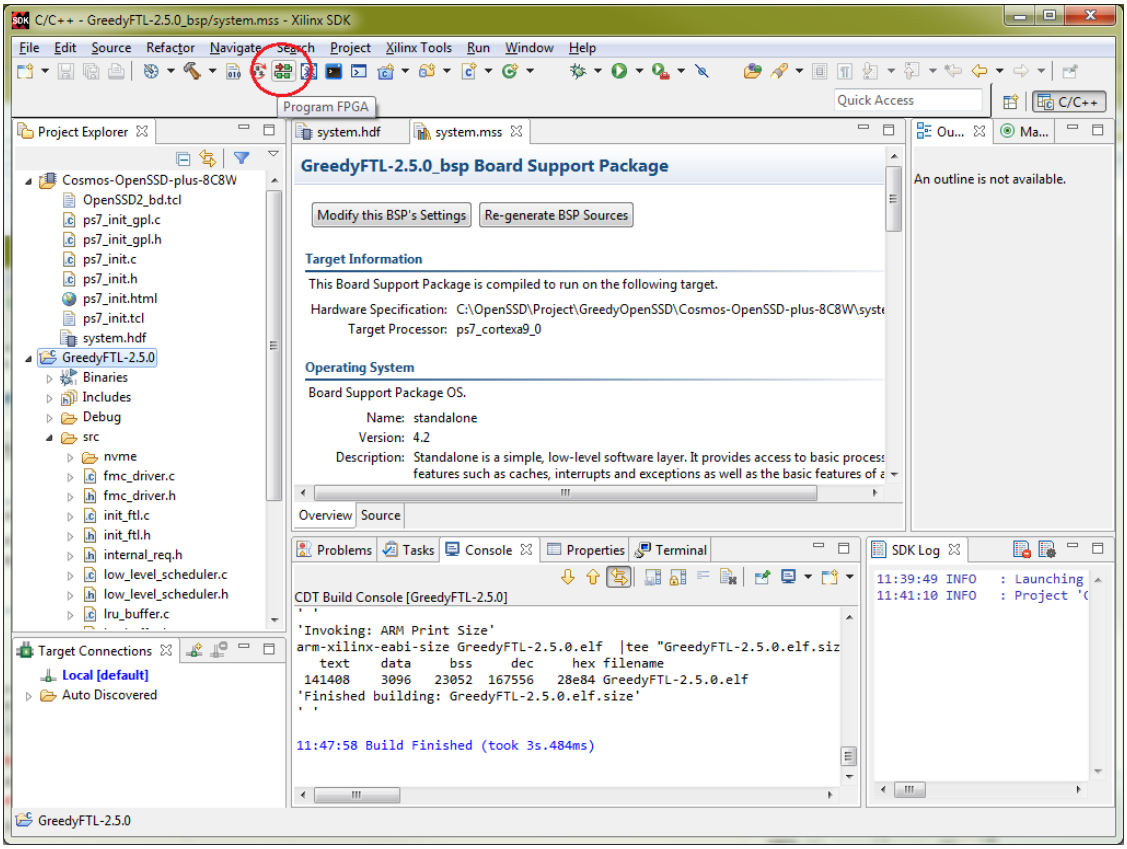


3.12. If then, press “Build All” to make both debug and release executables

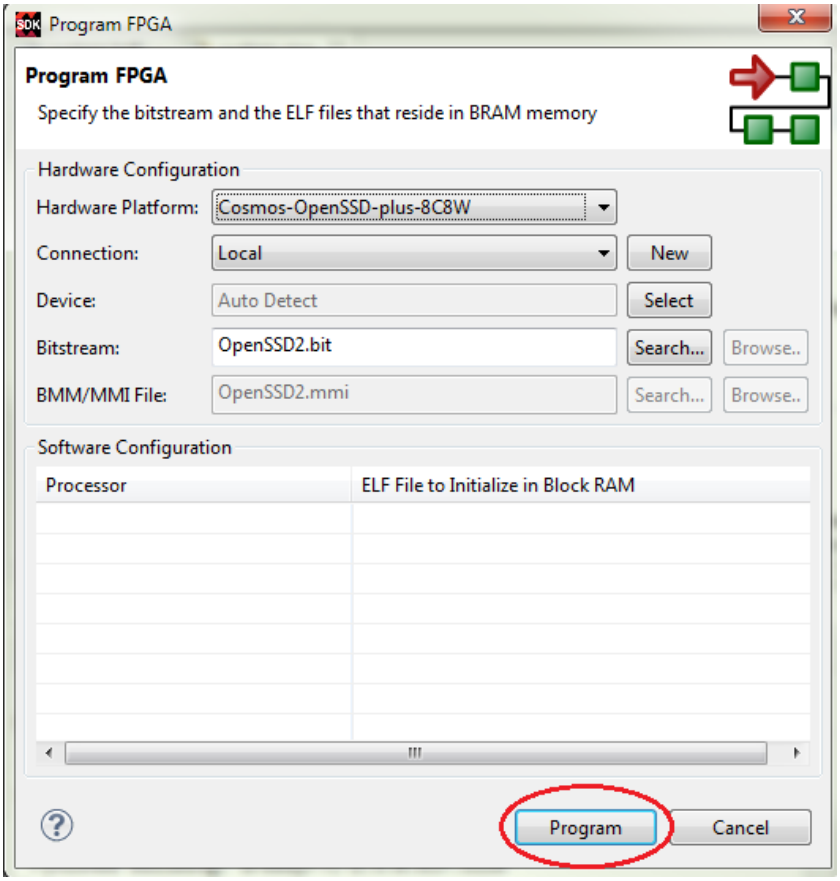


3.13. Launch a UART console program and open the port (baud rate: 115200)

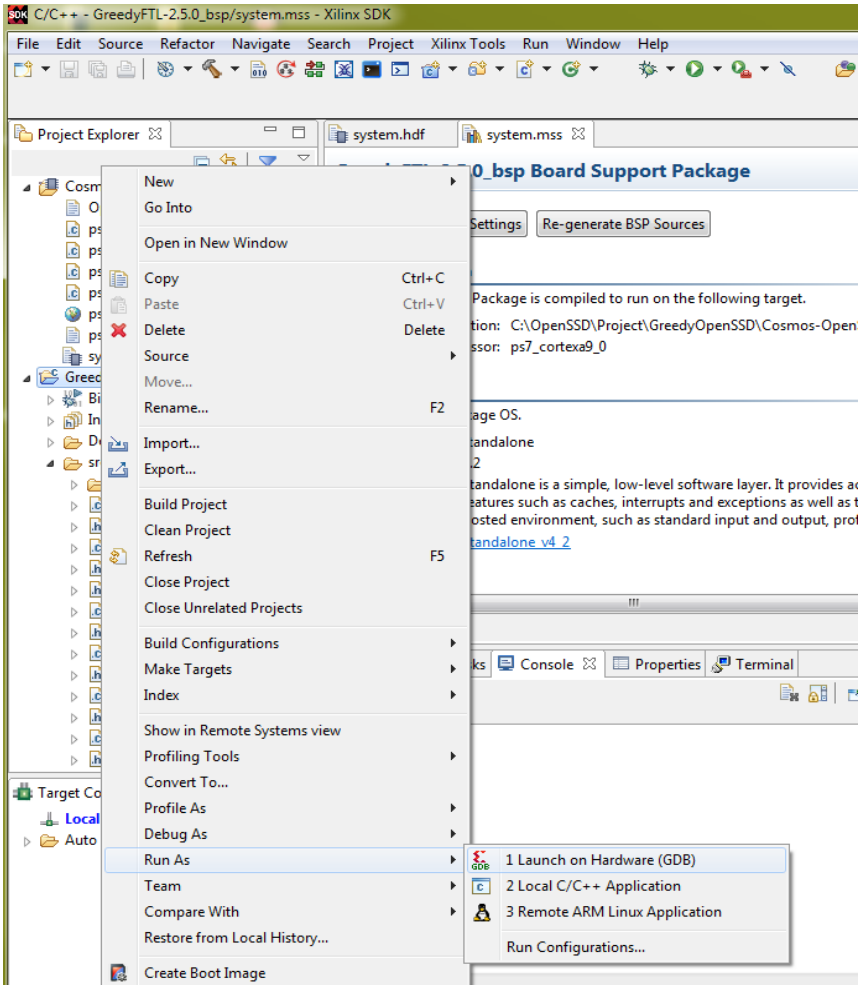
3.14. Press “Program FPGA” button or go to Xilinx Tools->Program FPGA

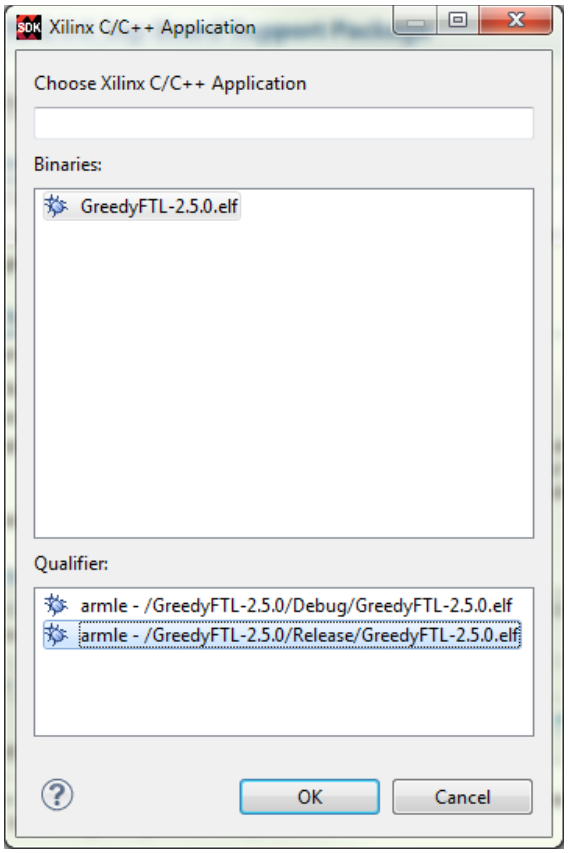


3.15. Press “Program”

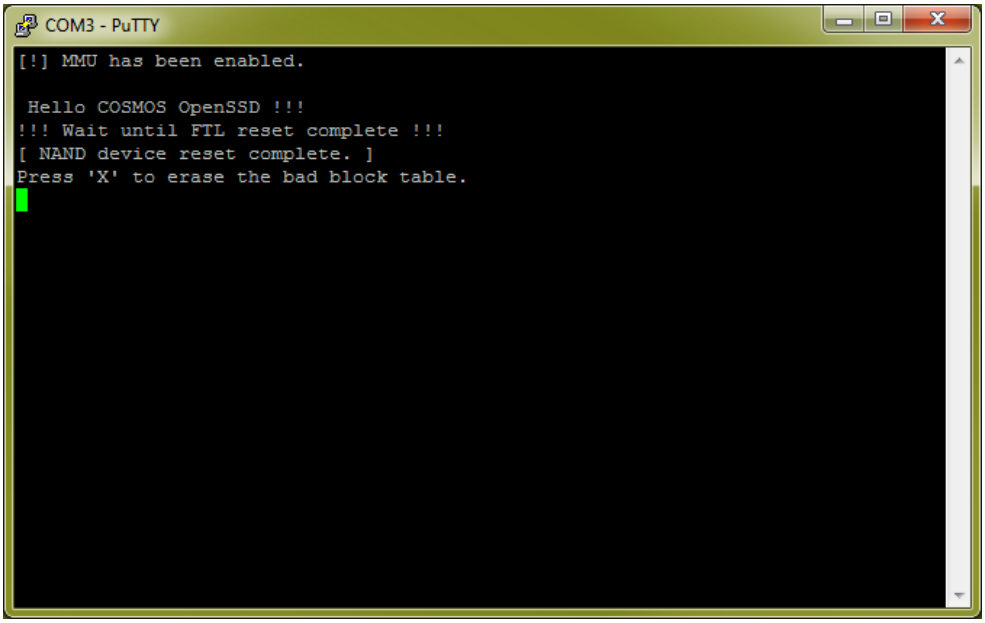


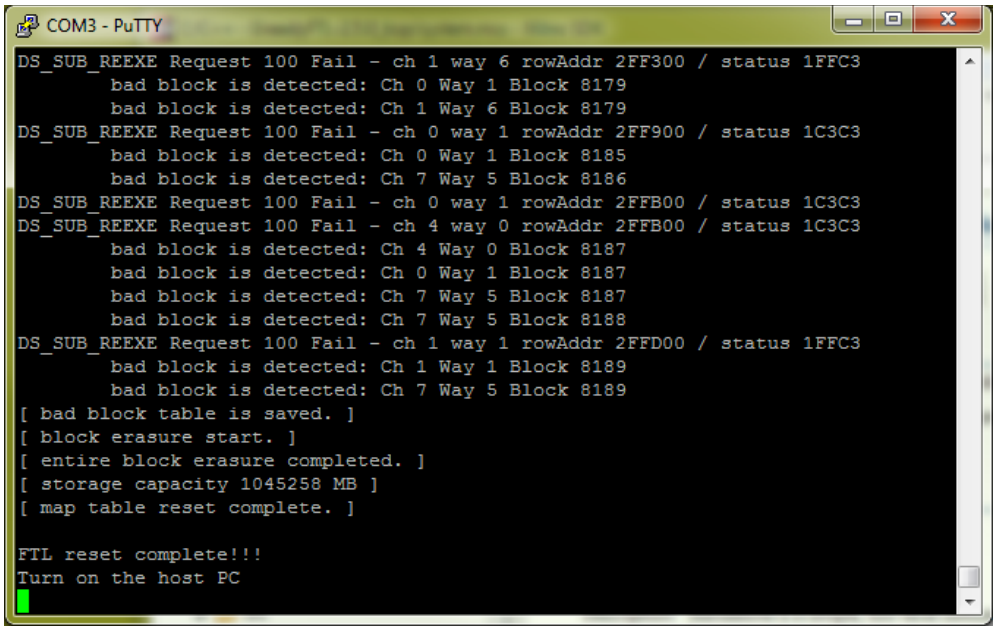
3.16. Launch the firmware





3.17. Watch the UART console





3.18. Turn on the host computer