

## ***Cosmos OpenSSD+ Tutorial***

## ***Revision history***

<b>Date</b>	<b>Version</b>	<b>Revision</b>
2016-09-01	1.0	First draft

# Table of Contents

<b>Chapter 1: Introduction .....</b>	<b>1</b>
1.1. Overview.....	1
1.1.1. Materials.....	1
<b>Chapter 2: Get Started.....</b>	<b>2</b>
2.1. Required Tools.....	2
2.1.1. Xilinx Vivado Design Suite .....	2
2.2. Required Files.....	2
2.2.1. Xilinx Hardware Description File .....	2
2.2.2. Firmware (flash translation layer) source .....	2
<b>Chapter 3: Deploying Steps.....</b>	<b>3</b>
3.1. Download required files and create a workspace directory.....	3
3.2. Launch Xilinx Software Development Kit and designate the workspace .....	3
3.3. Go to File->Application Project.....	4
3.4. Press "New" to register the hardware description file (HDF).....	5
3.5. Name the hardware project and specify a path of the HDF .....	5
3.6. Name the application project and finish this template wizard.....	6
3.7. Locate "src" directory and copy greedy FTL source files to it.....	7
3.8. Get back to SDK and refresh the project explorer .....	8
3.9. Go to "properties" menu.....	8
3.10. Go to C/C++ Build->Settings and specify include paths .....	9
3.11. If everything goes well, the build process should finish successfully.....	10
3.12. If then, press "Build All" to make both debug and release executables.....	11
3.13. Launch a UART console program and open the port (baud rate: 115200)....	12
3.14. Press "Program FPGA" button or go to Xilinx Tools->Program FPGA.....	12
3.15. Press "Program" .....	13
3.16. Launch the firmware .....	14

3.17.	Watch the UART console .....	15
3.18.	Turn on the host computer.....	15

# 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](http://www.openssd-project.org/wiki/Cosmos_OpenSSD_Technical_Resources)



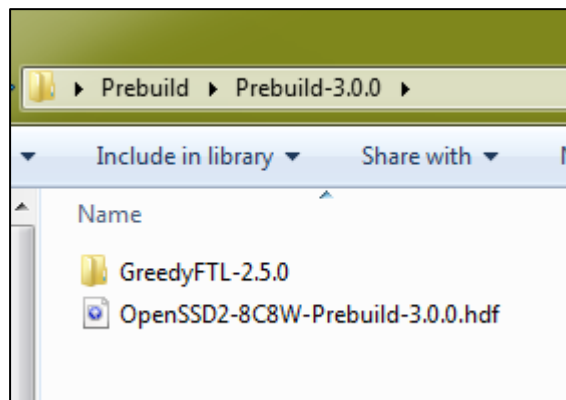
## Chapter 2: Get Started

### 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

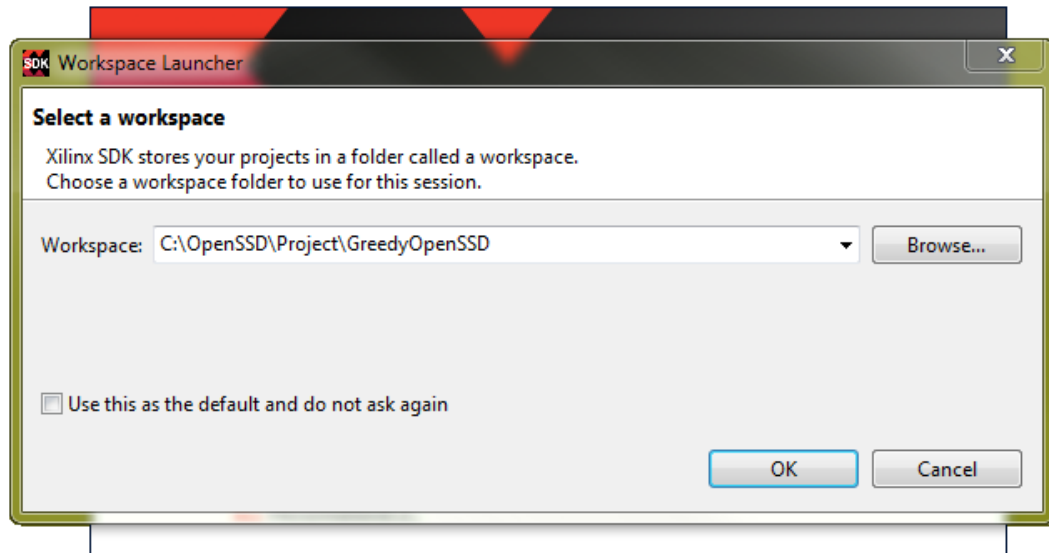
Hardware description file (.hdf) is a compressed file which contains a bitstream file (.bit) and multiple source codes for initializing Zynq processing system (PS). A board support package is generated by importing the hardware description file to Xilinx Vivado SDK. The hardware description file is also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

#### 2.2.2. Firmware (flash translation layer) source

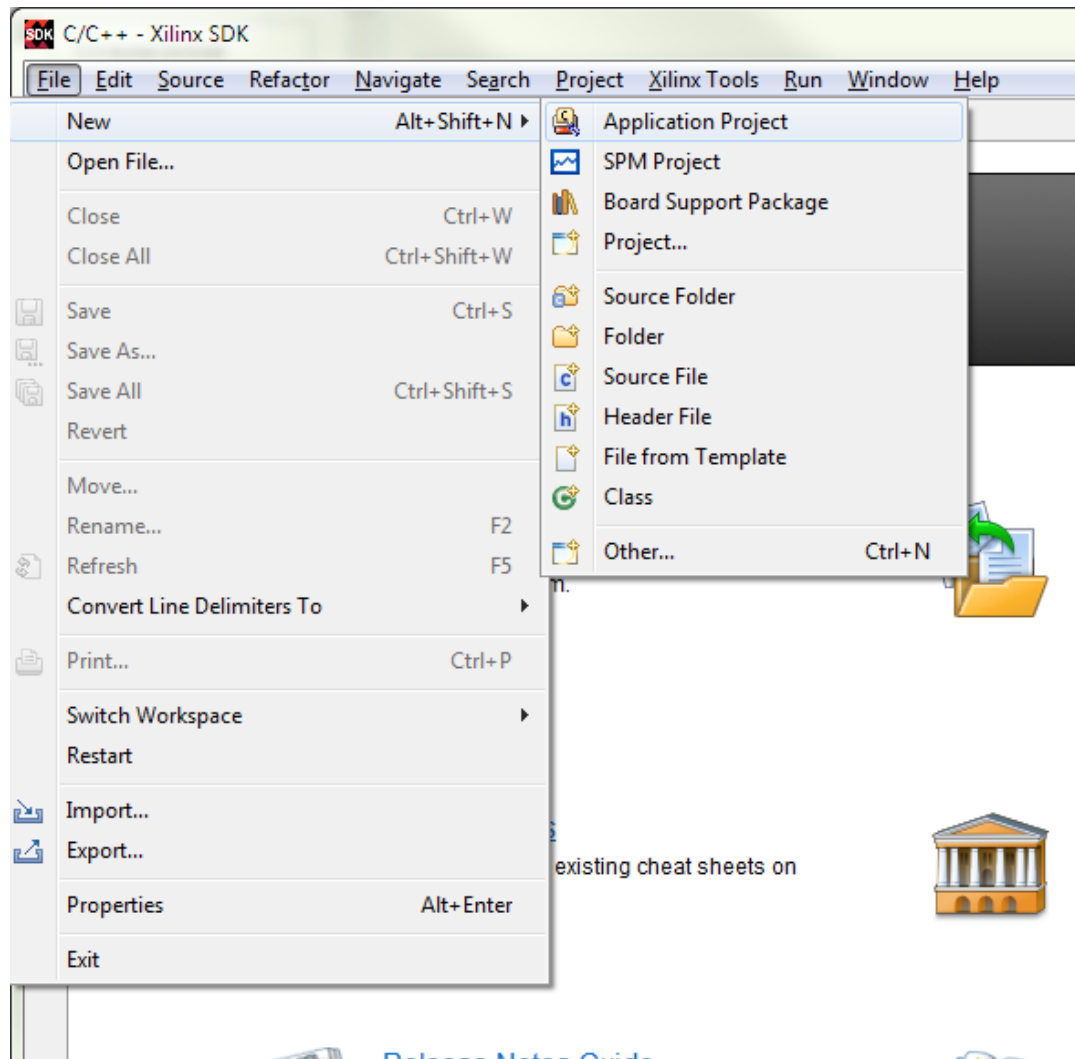
The latest Cosmos OpenSSD firmware contains a flash translation layer and a host interface logic to be operated as a solid-state drive. The FTL has a software-level NAND flash scheduler and supports greedy garbage collection mechanism. In addition, it is responsible for managing the NVMe controller.

## 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

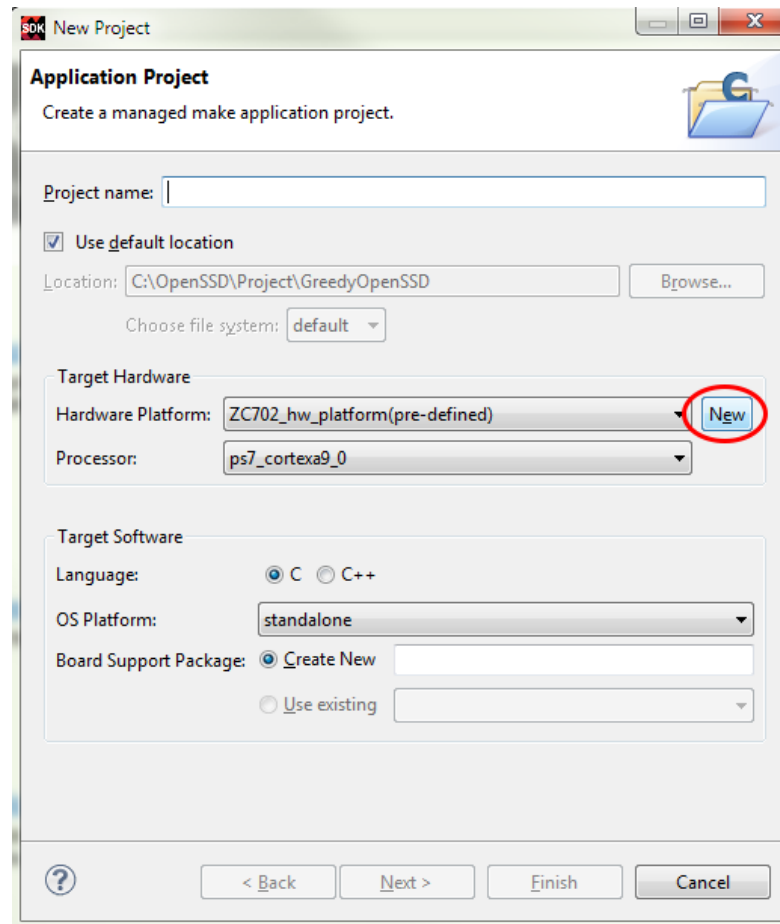


### 3.3. Go to File->Application Project

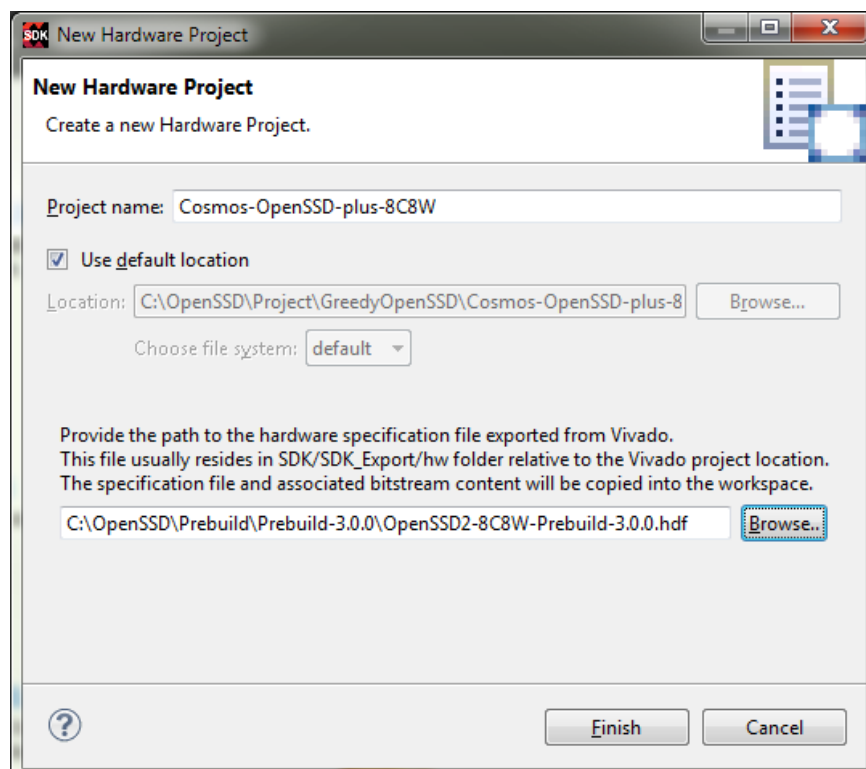




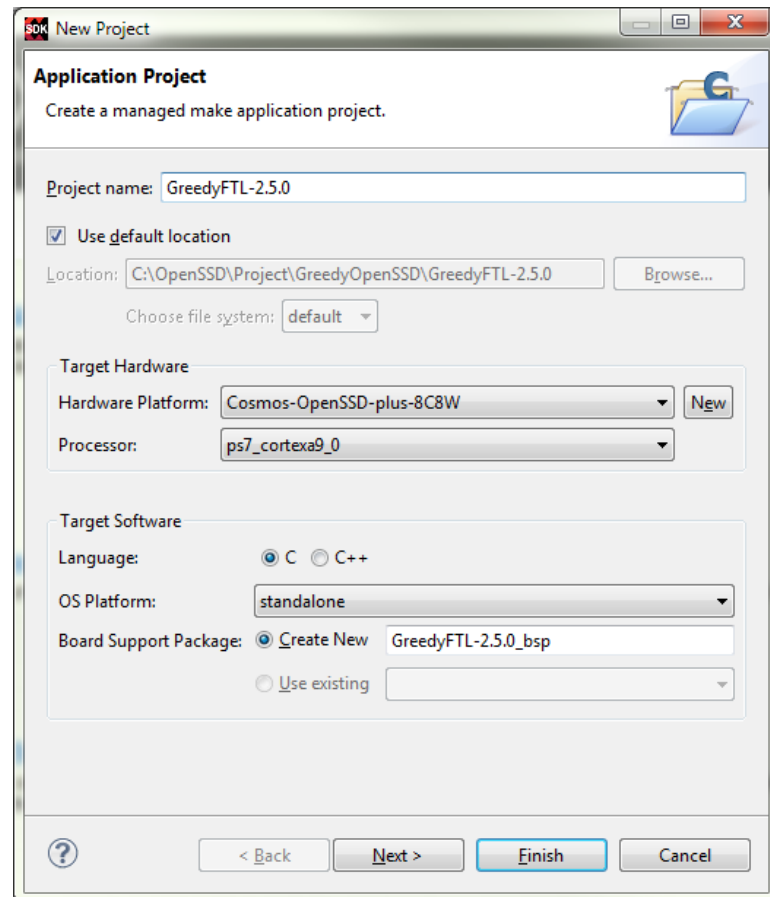
### 3.4. Press “New” to register the hardware description file (HDF)



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



### 3.6. Name the application project and finish this template wizard



**New Project**

**Application Project**  
Create a managed make application project.

Project name: GreedyFTL-2.5.0

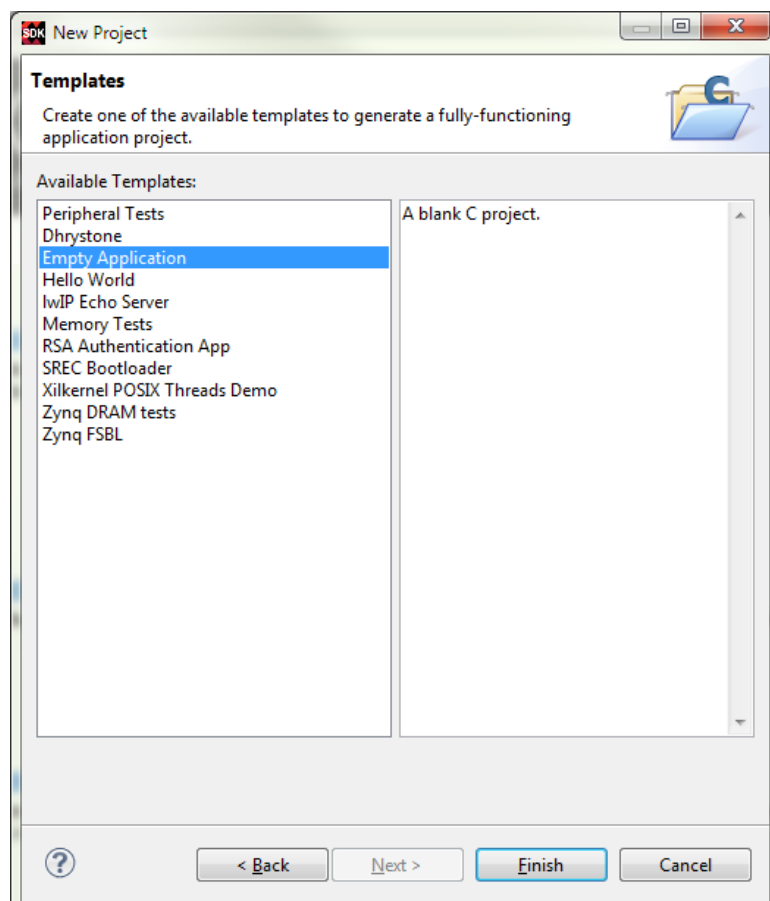
☒ Use default location  
Location: C:\OpenSSD\Project\GreedyOpenSSD\GreedyFTL-2.5.0 Browse...

Choose file system: default

**Target Hardware**  
Hardware Platform: Cosmos-OpenSSD-plus-8C8W New  
Processor: ps7\_cortexa9\_0

**Target Software**  
Language: ☒ C ☐ C++  
OS Platform: standalone  
Board Support Package: ☒ Create New GreedyFTL-2.5.0\_bsp  
☐ Use existing

? < Back Next > Finish Cancel



**New Project**

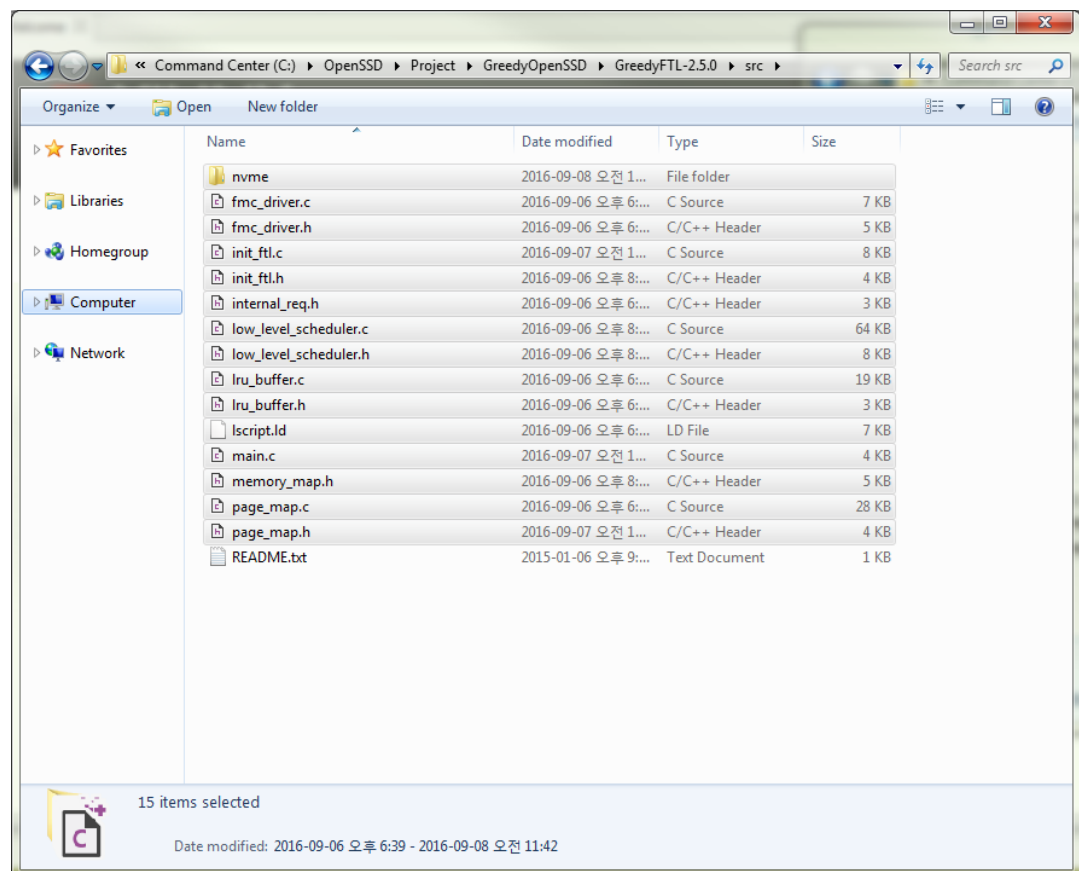
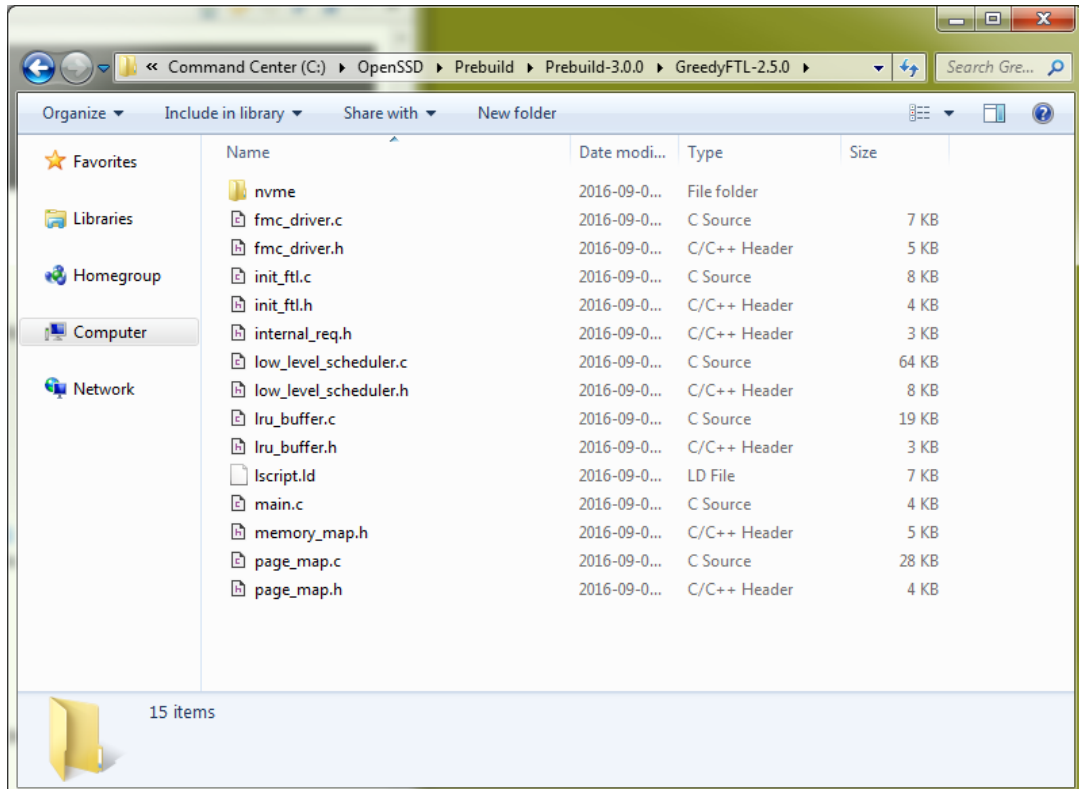
**Templates**  
Create one of the available templates to generate a fully-functioning application project.

**Available Templates:**

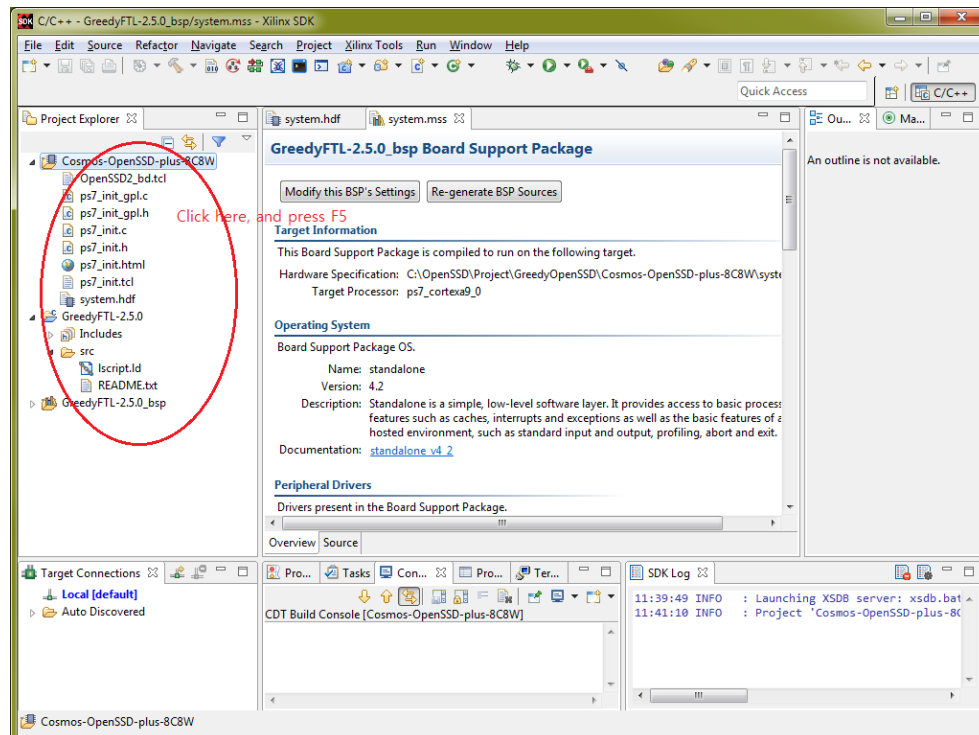
Peripheral Tests	A blank C project.
Dhrystone	
<b>Empty Application</b>	
Hello World	
IwIP Echo Server	
Memory Tests	
RSA Authentication App	
SREC Bootloader	
Xilkernel POSIX Threads Demo	
Zynq DRAM tests	
Zynq FSBL	

? < Back Next > Finish Cancel

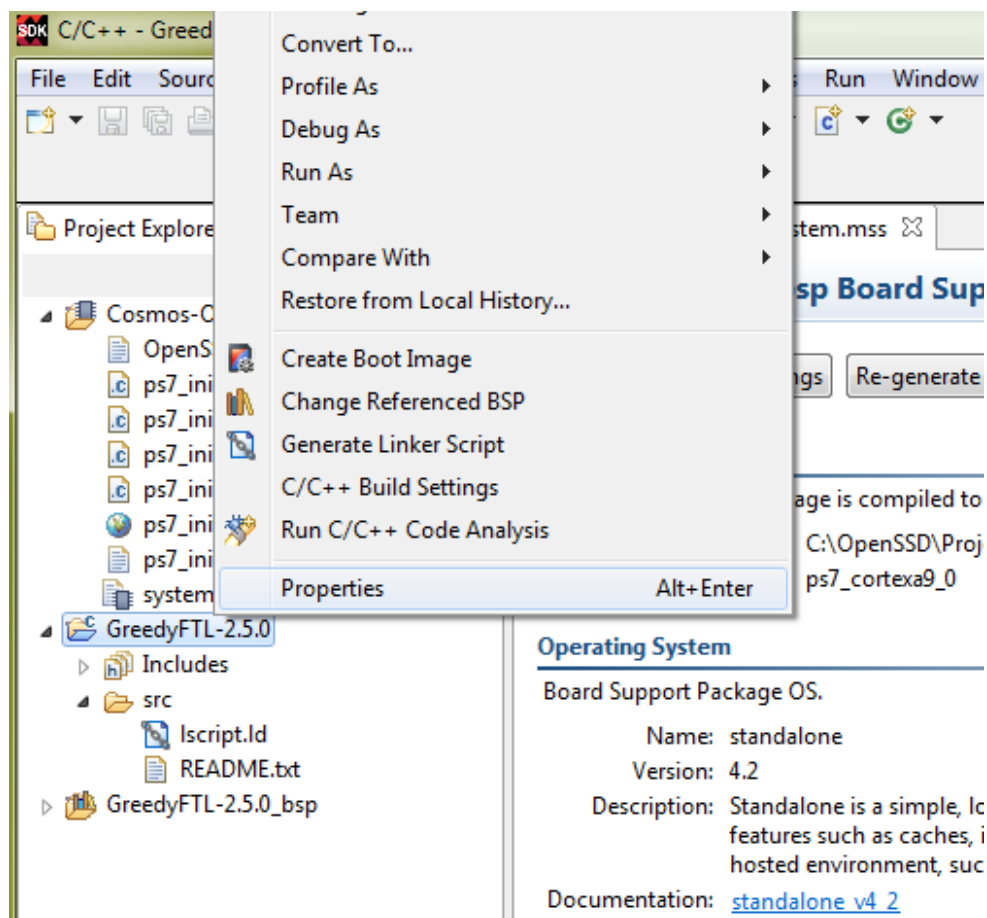
### 3.7. Locate “src” directory and copy greedy FTL source files to it



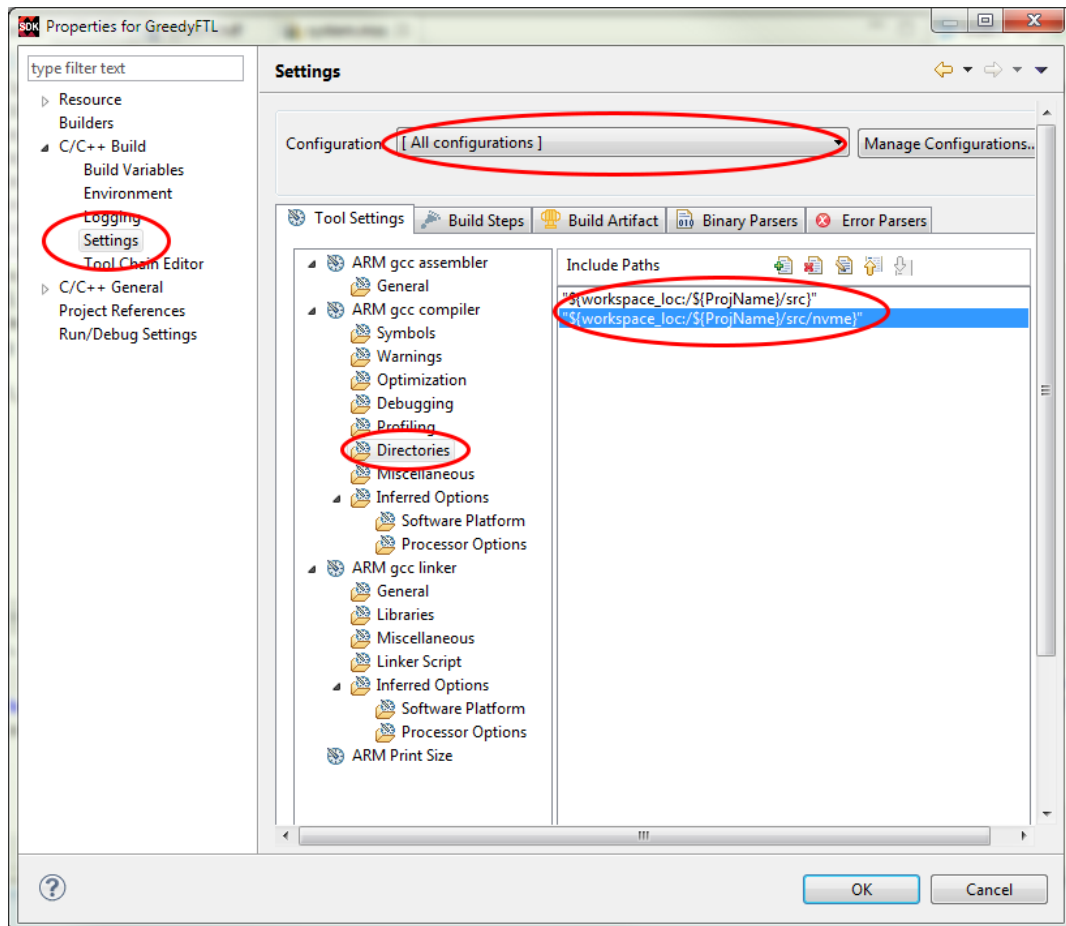
### 3.8. Get back to SDK and refresh the project explorer



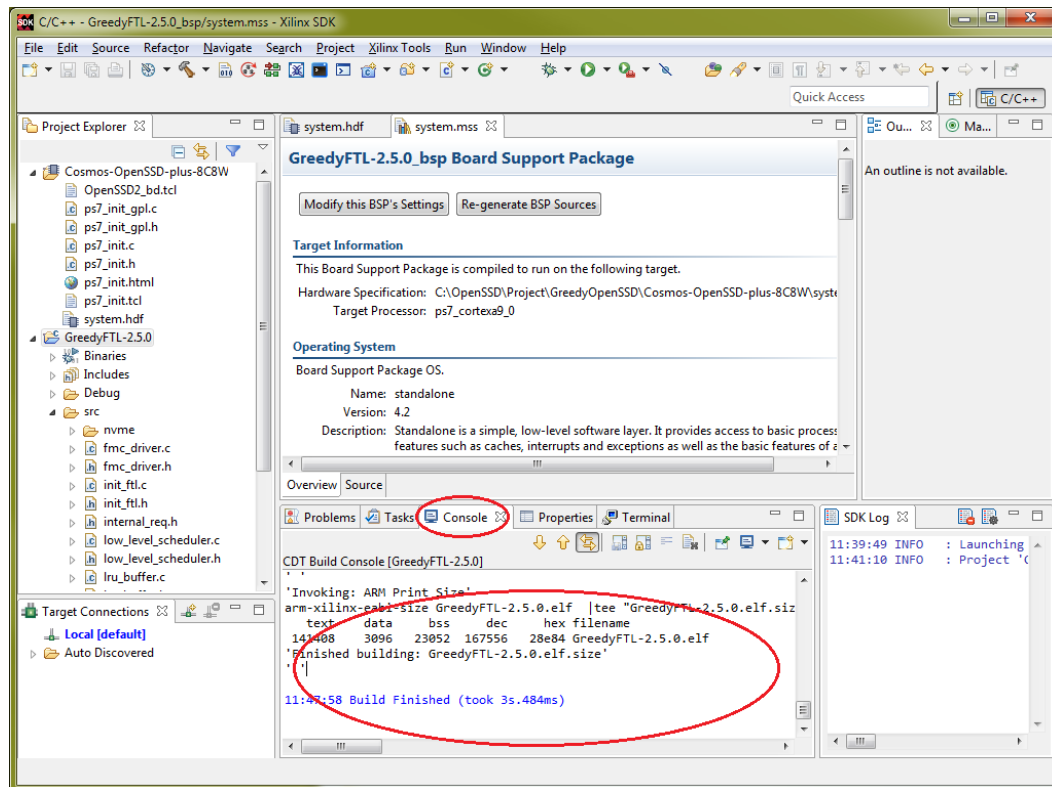
### 3.9. Go to “properties” menu



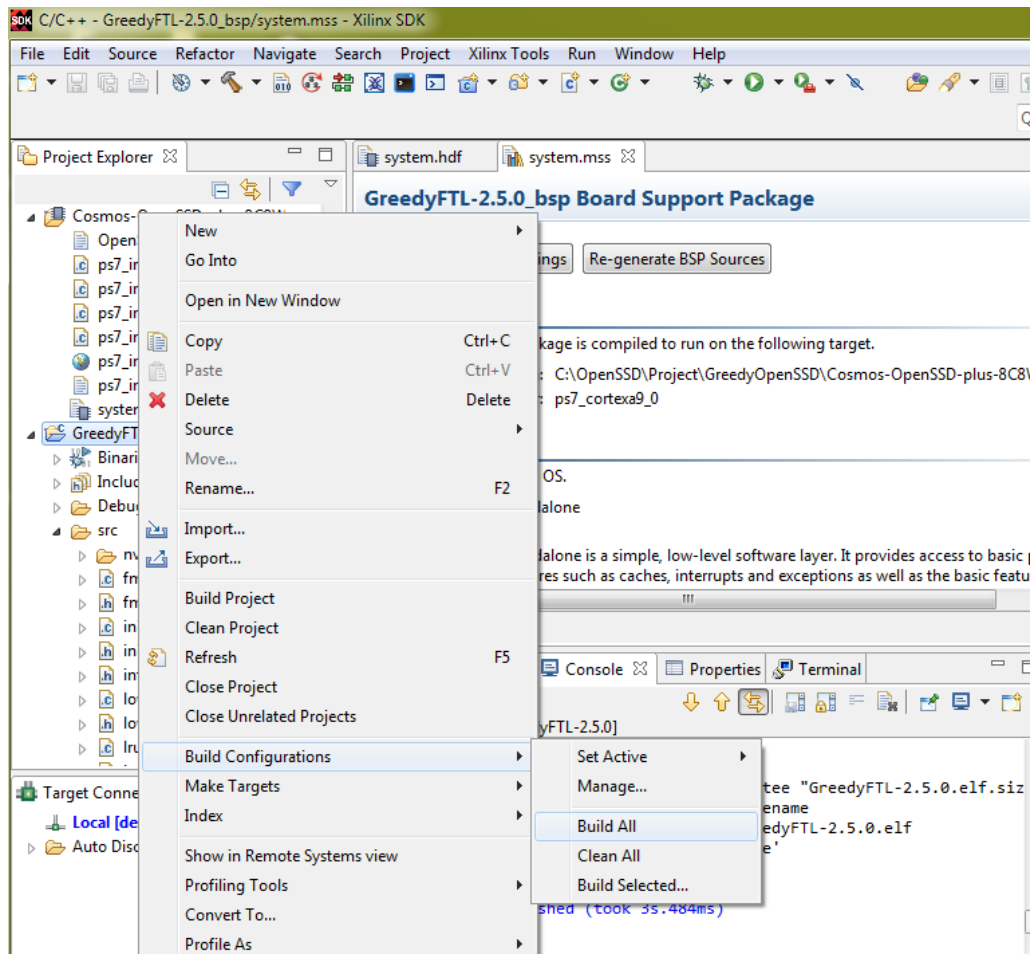
### 3.10. Go to C/C++ Build->Settings and specify include paths



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

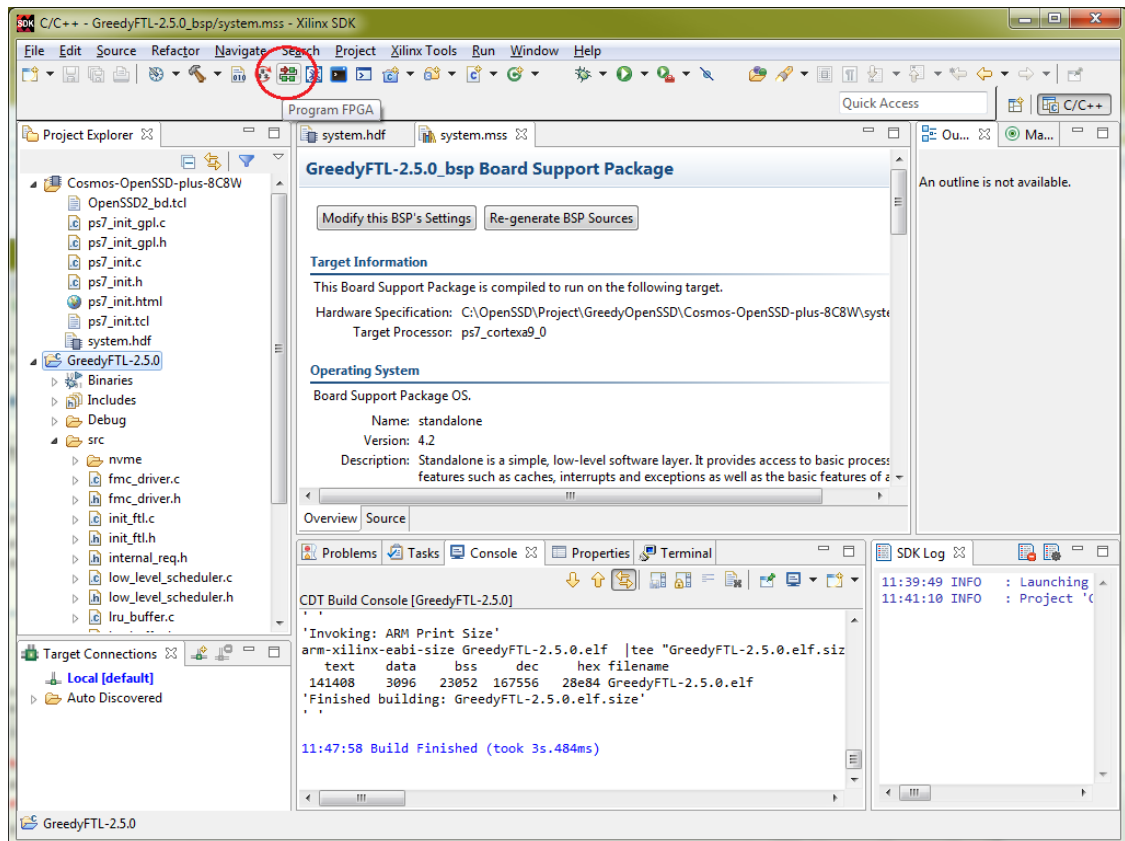


### 3.12. If then, press “Build All” to make both debug and release executables



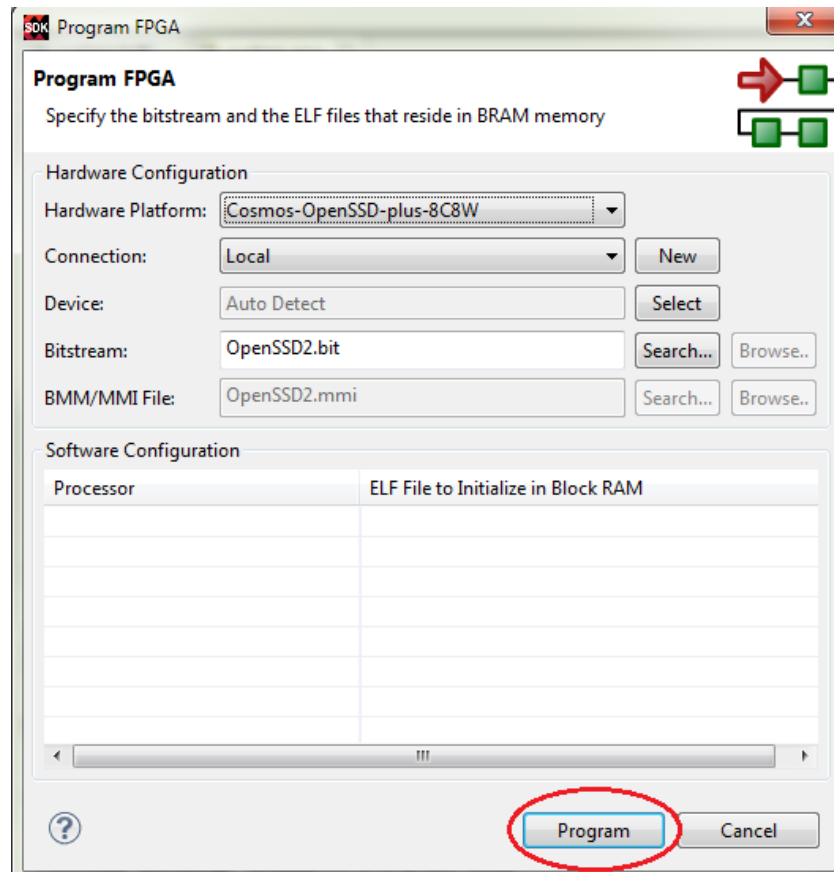
3.13. Launch a UART console program and open the port (baud rate: 115200)

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

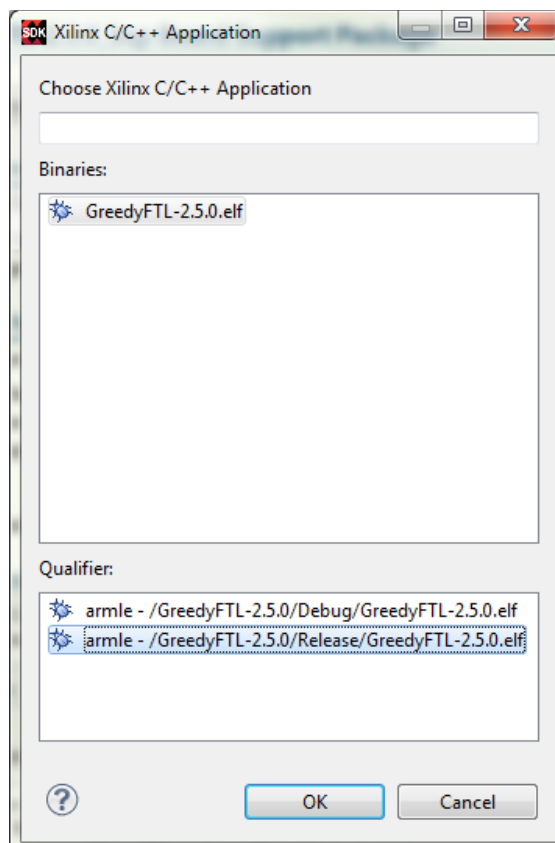
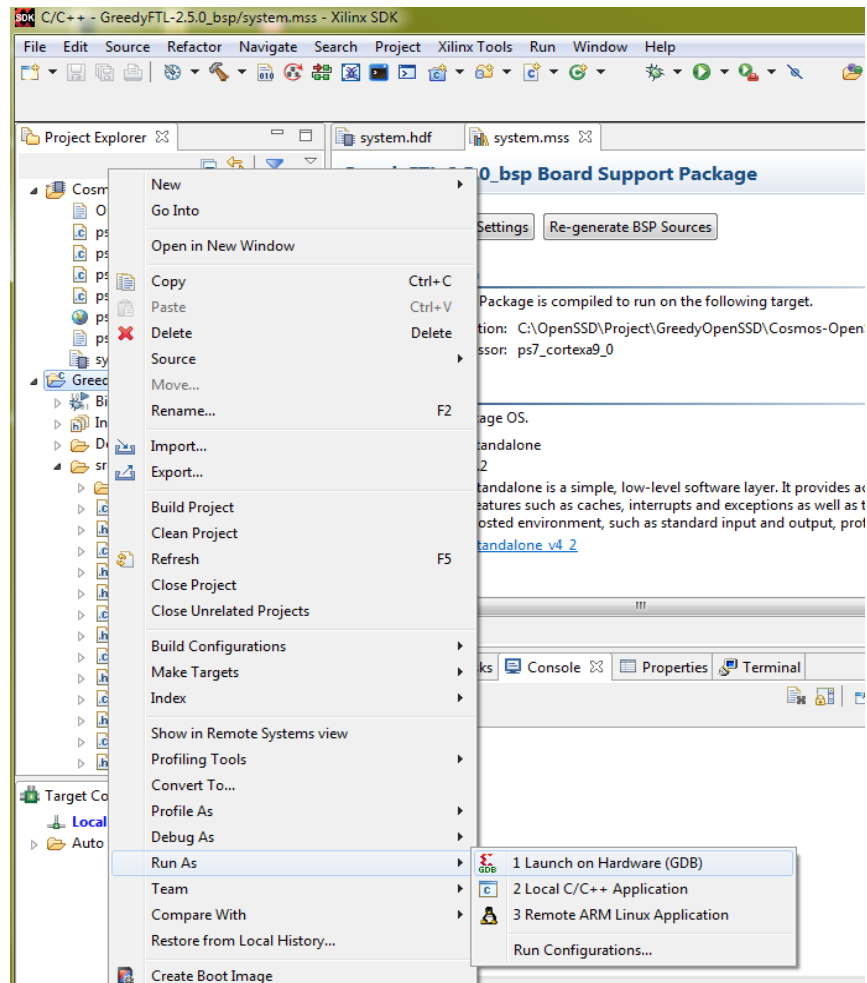




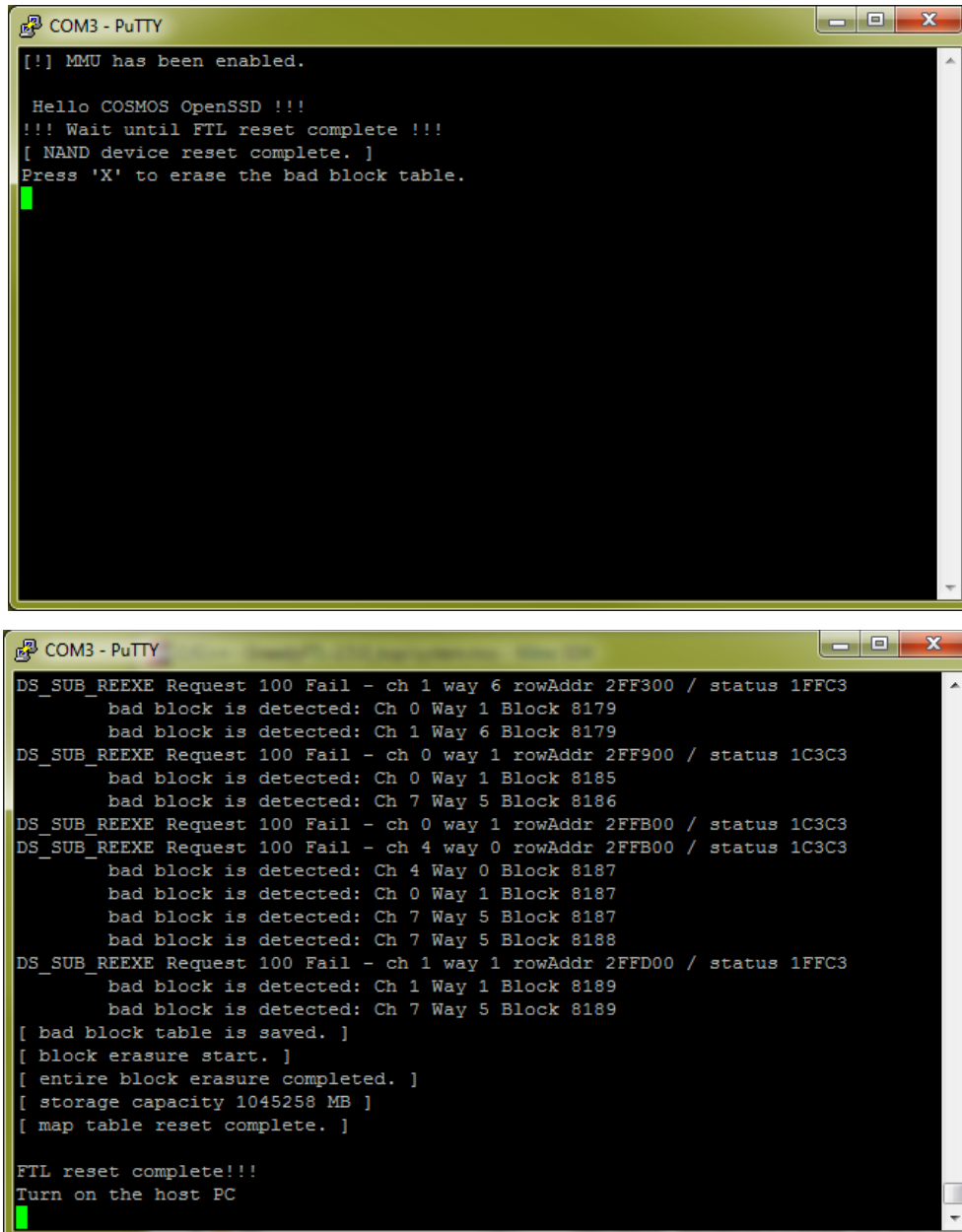
### 3.15. Press “Program”



## 3.16. Launch the firmware



## 3.17. Watch the UART console



```

COM3 - PuTTY
[!] MMU has been enabled.

Hello COSMOS OpenSSD !!!
!!! Wait until FTL reset complete !!!
[ NAND device reset complete. ]
Press 'X' to erase the bad block table.
█

COM3 - PuTTY
DS_SUB_REEXE Request 100 Fail - ch 1 way 6 rowAddr 2FF300 / status 1FFC3
    bad block is detected: Ch 0 Way 1 Block 8179
    bad block is detected: Ch 1 Way 6 Block 8179
DS_SUB_REEXE Request 100 Fail - ch 0 way 1 rowAddr 2FF900 / status 1C3C3
    bad block is detected: Ch 0 Way 1 Block 8185
    bad block is detected: Ch 7 Way 5 Block 8186
DS_SUB_REEXE Request 100 Fail - ch 0 way 1 rowAddr 2FFB00 / status 1C3C3
DS_SUB_REEXE Request 100 Fail - ch 4 way 0 rowAddr 2FFB00 / status 1C3C3
    bad block is detected: Ch 4 Way 0 Block 8187
    bad block is detected: Ch 0 Way 1 Block 8187
    bad block is detected: Ch 7 Way 5 Block 8187
    bad block is detected: Ch 7 Way 5 Block 8188
DS_SUB_REEXE Request 100 Fail - ch 1 way 1 rowAddr 2FFD00 / status 1FFC3
    bad block is detected: Ch 1 Way 1 Block 8189
    bad block is detected: Ch 7 Way 5 Block 8189
[ bad block table is saved. ]
[ block erasure start. ]
[ entire block erasure completed. ]
[ storage capacity 1045258 MB ]
[ map table reset complete. ]

FTL reset complete!!!
Turn on the host PC
█

```

## 3.18. Turn on the host computer