Loading Instruction Set on to uZed EEPROM

SDK = Software Design Kit

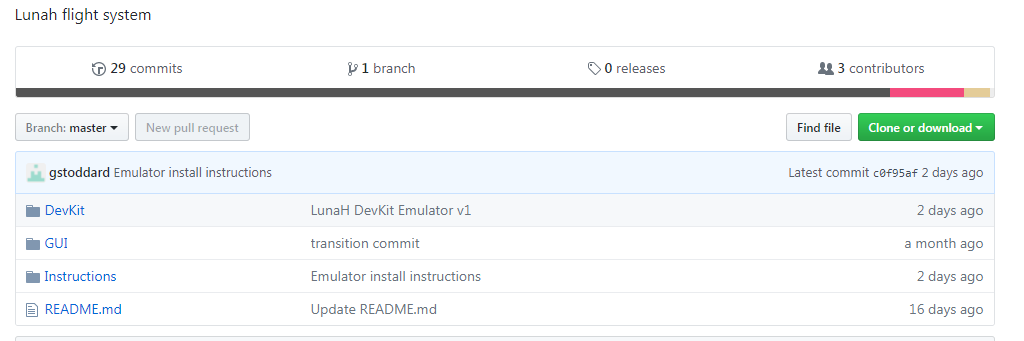
uZed = Avnet development kit consisting of Zynq SOC and peripherals.

Workspace = Local Directory consisting of the instructions for the FPGA and processor on the Zynq

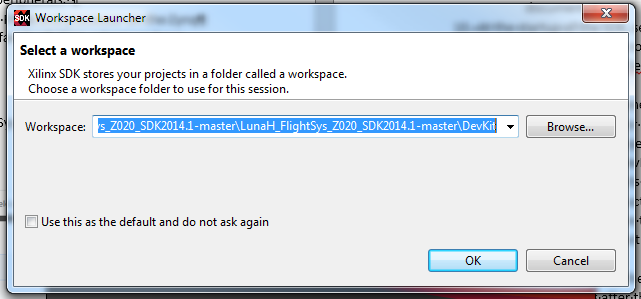
Zynq = Xilinx System on a Chip. This is the name of a large family of chips, where we have utilized only the Z010 and Z020 chips.

BSP = ………………Board Support Package

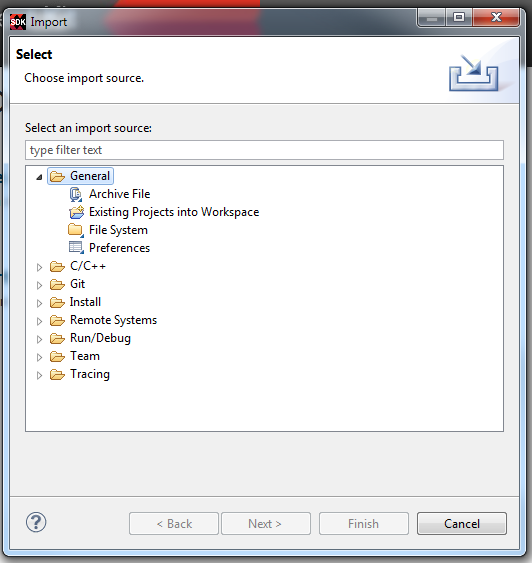
1. Go to github ([link](https://github.com/RMDInc/LunaH_FlightSys_Z020_SDK2014.1)) and download the project “LunaH\_FlightSys\_Z020\_2014\_1”.
2. Click “Clone or Download”



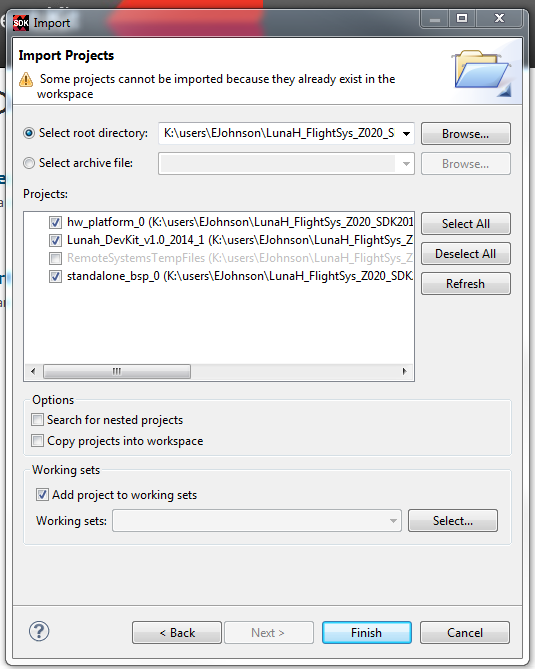
1. Unzip the data to your [Local Director]/../LunaH\_FlightSys\_Z020\_SDK2014.1-master/
2. Start Xilinx 2015.3 SDK
3. Set the [WORKSPACE] = “[Local Director]/../LunaH\_FlightSys\_Z020\_SDK2014.1-master/DevKit”



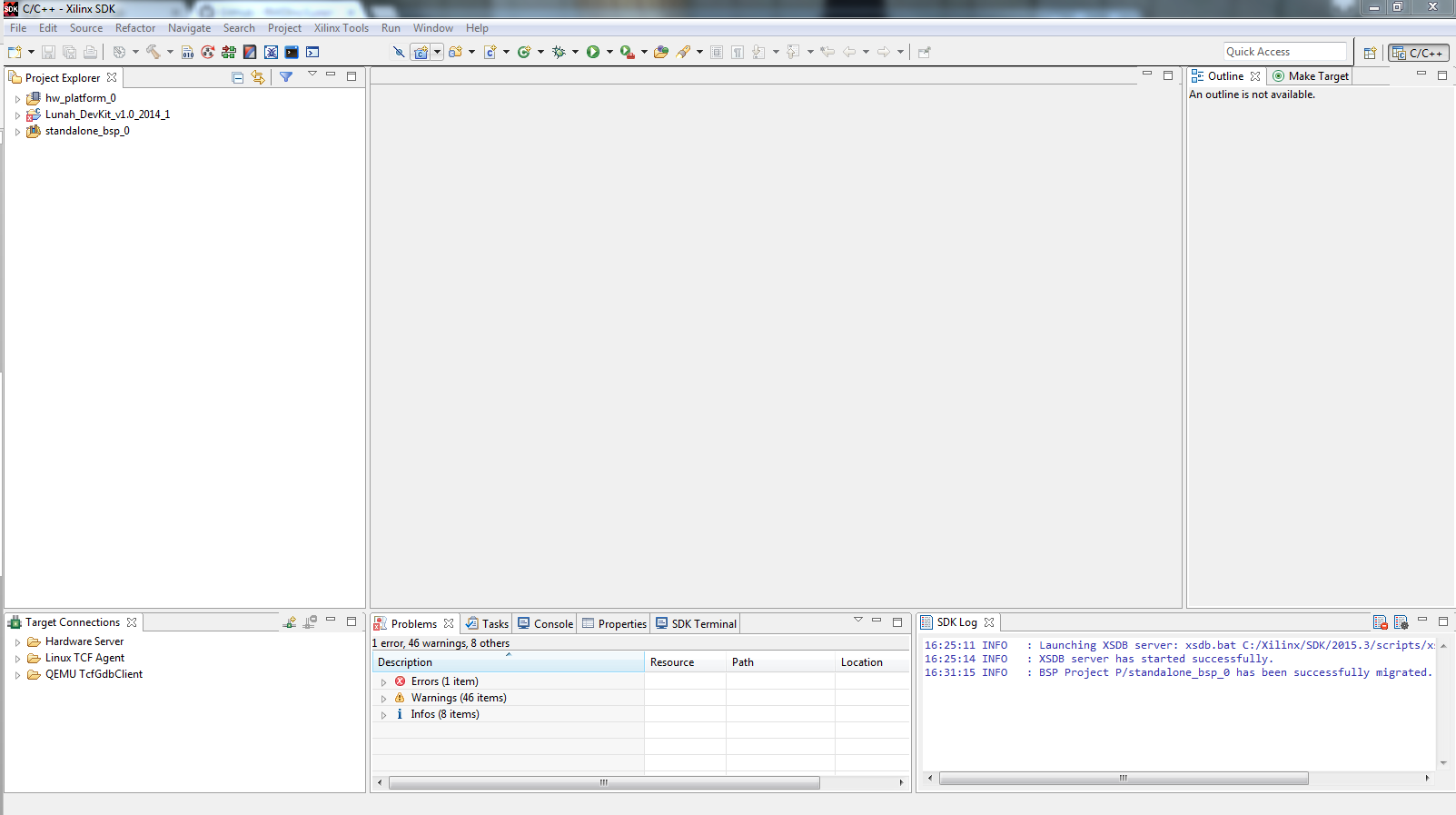
1. Import Files:
   1. File > Import > General > Existing projects into workspace



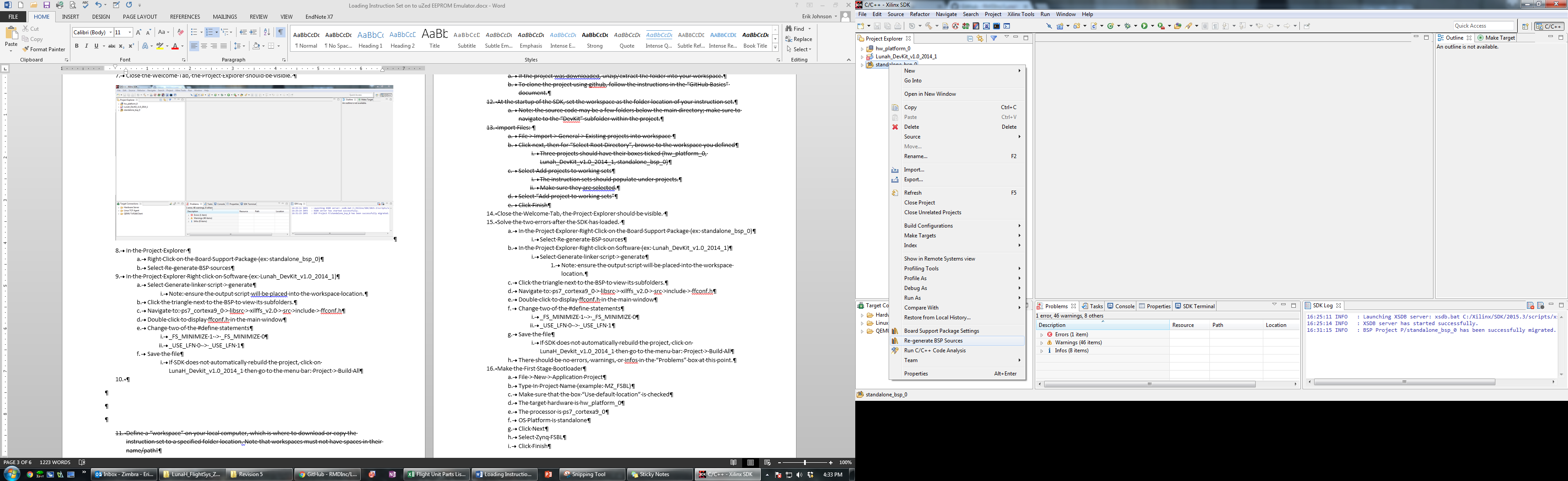
* 1. Click next,
  2. In Select Root Directory Field, browse to [WORKSPACE].
  3. Three projects should have their boxes ticked
     1. hw\_platform\_0
     2. Lunah\_DevKit\_v1.0\_2014\_1
     3. standalone\_bsp\_0
  4. Under Working sets region, Click the “Add project to working sets” button.
  5. Click Finish
  6. Wait until Building workspace is completed. May take 1-2 minutes.



1. Close the Welcome Tab, the Project Explorer should be visible.

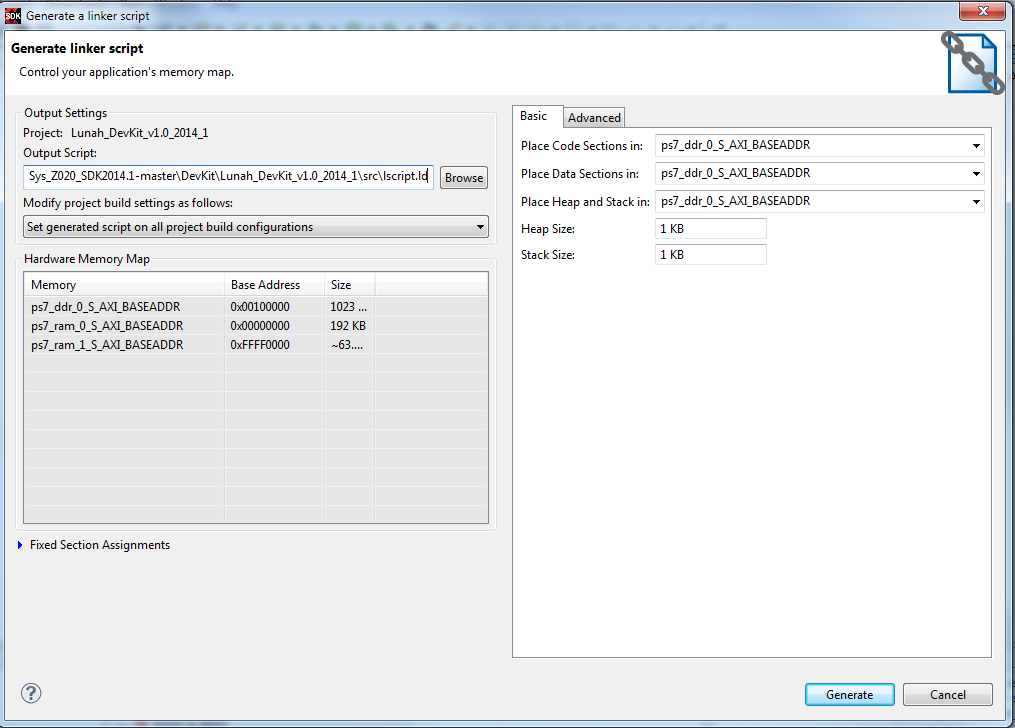


1. In the Project Explorer
   1. Right Click on the Board Support Package (standalone\_bsp\_0)
   2. Select Re-generate BSP sources
   3. Click “Yes”



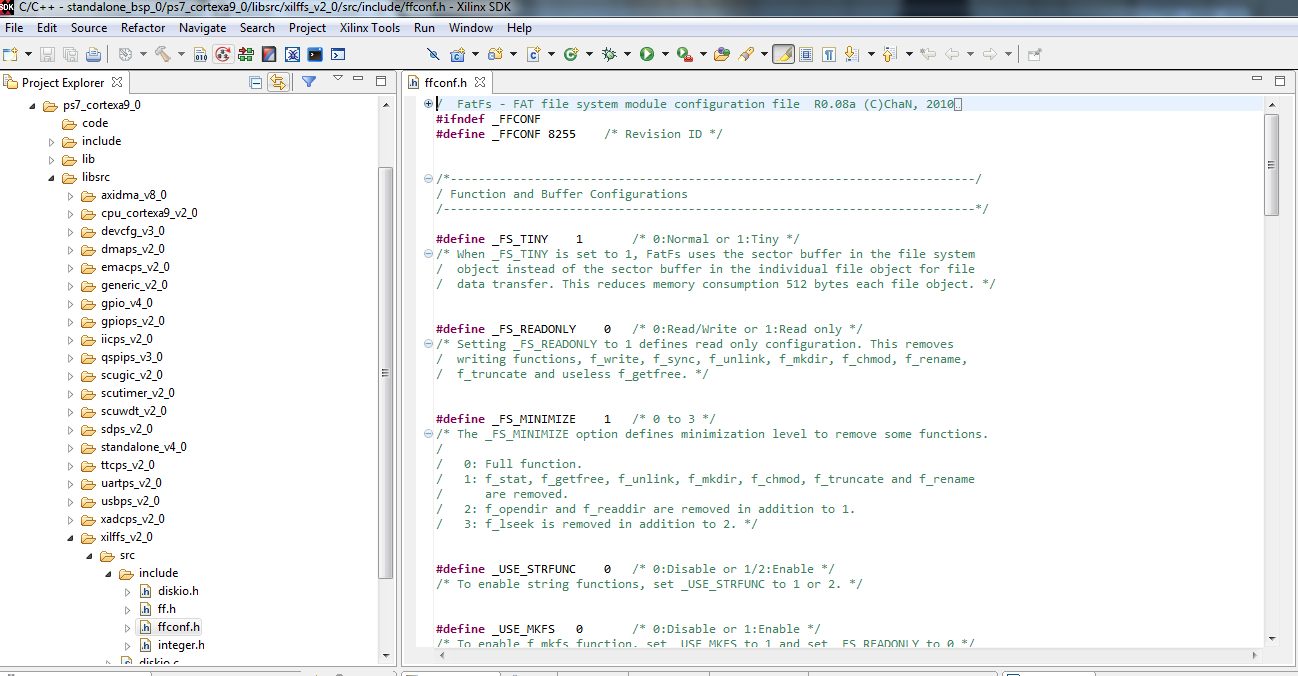
Ignore any errors.

1. In the Project Explorer Right click on Software (Lunah\_DevKit\_v1.0\_2014\_1)
   1. Select Generate Linker Script > generate
      1. Note: ensure the output script will be placed into the workspace location.



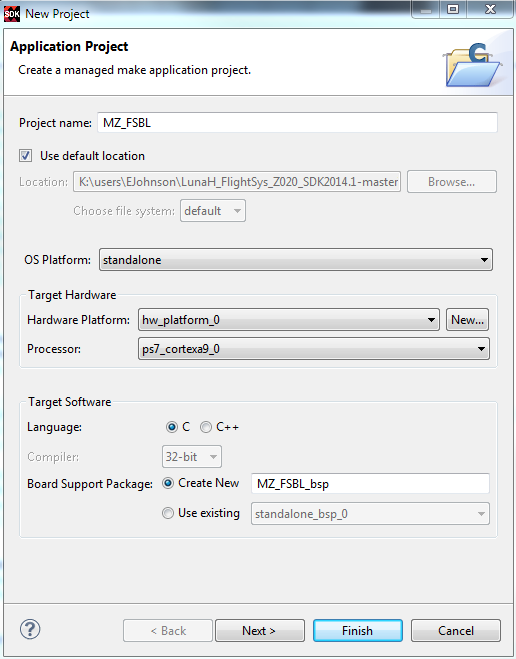
There will be Errors in the Problems dialog. Ignore them.

* 1. Click the triangle next to the “standalone\_bs\_0” to view its subfolders.
  2. Navigate to: ps7\_cortexa9\_0 > libsrc > xilffs\_v2.0 > src >include > ffconf.h
  3. Double click to display ffconf.h in the main window

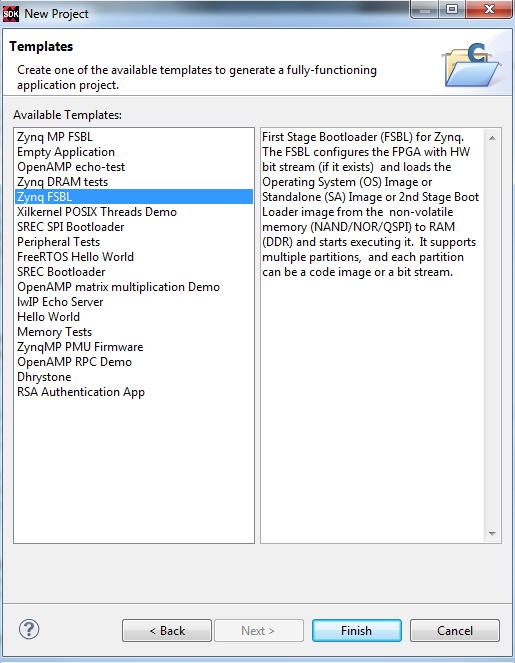


* 1. Change two of the #define statements
     1. \_FS\_MINIMIZE 1 🡪 \_FS\_MINIMIZE 0 (line 29)
     2. \_USE\_LFN 0 🡪 \_USE\_LFN 1 (line 93)
  2. Save the file
  3. Highlight (one click) LunaH\_Devkit\_v1.0\_2014\_1 in the Project Explorer
  4. Click Project > Build All (cltr+b)

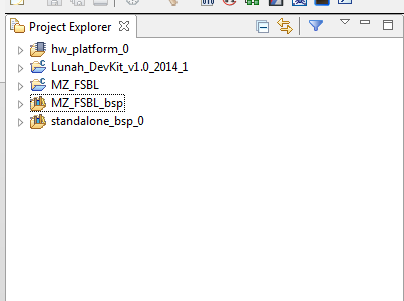
1. Make the First Stage Bootloader
   1. In the menu banner, File > New > Application Project
   2. Type In Project Name : “MZ\_FSBL”
   3. Make sure that the box “Use default location” is checked
   4. The target hardware is hw\_platform\_0
   5. The processor is ps7\_cortexa9\_0
   6. OS Platform is standalone
   7. Click Next



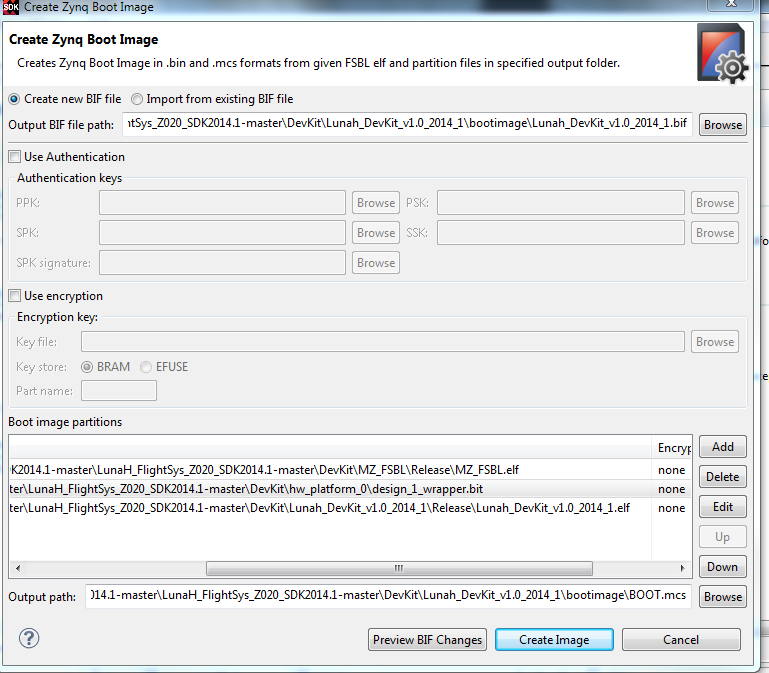
* 1. Select Zynq FSBL
  2. Click Finish



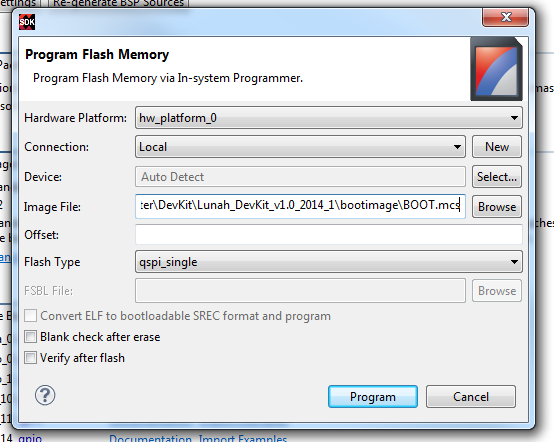
1. The first Stage bootloader files will be generated: MZ\_FSBL and MZ\_FSBL\_bsp.



1. Build the FSBL and Software in Release mode
   1. Right Click on “MZ\_FSBL”
   2. Select Build Configurations > Set Active > Release
   3. Right Click on Software “Lunah\_DevKit\_v1.0\_2014\_1”
   4. Select Build Configurations > Set Active > Release
2. In the menu banner, click Project> Build All (ctrl+b)
3. In the menu banner, click Xilinx Tools > Create Zynq Boot Image
4. Create New BIF file
   1. Browse to the Software folder [WORKSPACE]../Lunah\_DevKit\_v1.0\_2014\_1/
   2. Create New Folder called “bootimage”
   3. Set filename to bootimage.bif in that folder
   4. Under boot image partitions add:
      1. Delete any existing file paths
      2. Click Add and a window will pop up.
      3. Set filepath as [WORKSPACE]../MZ\_FSBL/MZ\_FSBL.elf
      4. Set Partition type as “bootloader”
      5. Add [WORKSPACE]../hw\_platform\_0/design\_1\_wrapper.bit (Partition type “datafile”)
      6. Add [WORKSPACE]../Lunah\_DevKit\_v1.0\_2014\_1/Release/ Lunah\_DevKit\_v1.0\_2014\_1.elf (Partition type “datafile”)
      7. Set the output path as [WORKSPACE]/../Lunah\_DevKit\_v1.0\_2014\_1/bootimage/BOOT.mcs   
         in the bootimage folder created above.
5. Click “Create Image”



1. Program the EEPROM
   1. On the uZed board, Check Jumpers
      1. JP1: Jumper Pin 1 and Pin 2
      2. JP2: Jumper Pin1 and Pin 2
      3. JP3: Jumper Pin 2 and Pin3
   2. (Optional) download and install TerraTerm https://osdn.jp/projects/ttssh2/downloads/65898/teraterm-4.91.exe/
   3. Download and install USB Driver for UART https://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx
   4. Insert SD Flash memory card into the uZed board.
   5. Connect the board (uZed) to the computer via the JTag cable (Digilent JTAG-HS3 Rev. A)
   6. Connect the micro USB to the computer.
      1. The board should be powered
      2. LEDs are On
      3. Silicon Labs 210x USB to UART Bridge device is recognized
      4. Digilent USB device is recognized.
   7. Select Xilinx Tools > Program Flash
   8. Set Image File as BOOT.mcs generated in previous step.
   9. Click Program. This may take 1-2 min.



* 1. Note the comments in Console window; the operation is completed and successful when it reports “Flash Operation Successful” – See Example of report below
  2. A red LED will be lit on the uZed board.
  3. Press the reset button (Blue button labelled “RST”)
  4. A blue LED will be lit if the system boot was successful. The system is now prepared.

**Programming Flash Report (Example)**

The bold text below are key operations. The file to load should be correct. The JTAG should be reconized, and the processor should be reset. The EEPROM should be erased and then we should have successful flash operation.

cmd /C program\_flash -f \

**K:\users\EJohnson\LunaH\_FlightSys\_Z020\_SDK2014.1-master\LunaH\_FlightSys\_Z020\_SDK2014.1-master\DevKit\Lunah\_DevKit\_v1.0\_2014\_1\bootimage\BOOT.mcs \**

-offset 0 -flash\_type qspi\_single -cable type xilinx\_tcf url TCP:127.0.0.1:3121

\*\*\*\*\*\* Xilinx Program Flash

\*\*\*\*\*\* Program Flash v2015.3 (64-bit)

\*\*\*\* SW Build 1368829 on Mon Sep 28 20:06:43 MDT 2015

\*\* Copyright 1986-2015 Xilinx, Inc. All Rights Reserved.

Connecting to hw\_server @ TCP:127.0.0.1:3121

WARNING: Failed to connect to hw\_server at TCP:127.0.0.1:3121

Attempting to launch hw\_server at TCP:127.0.0.1:3121

Connected to hw\_server @ TCP:127.0.0.1:3121

Available targets and devices:

**Target 0 : jsn-JTAG-HS3-210299A181D7**

**Device 0: jsn-JTAG-HS3-210299A181D7-4ba00477-0**

JTAG chain configuration

--------------------------------------------------

Device ID Code IR Length Part Name

1 4ba00477 4 arm\_dap

2 23727093 6 xc7z020

--------------------------------------------------

Enabling extended memory access checks for Zynq.

Writes to reserved memory are not permitted and reads return 0.

To disable this feature, run "debugconfig -memory\_access\_check disable".

--------------------------------------------------

CortexA9 Processor Configuration

-------------------------------------

Version.............................0x00000003

User ID.............................0x00000000

No of PC Breakpoints................6

No of Addr/Data Watchpoints.........4

**Processor Reset .... DONE**

SF: Detected S25FL128S\_64K with page size 256 Bytes, erase size 64 KiB, total 16 MiB

**Performing Erase Operation...**

**Erase Operation successful.**

INFO: [Xicom 50-44] Elapsed time = 11 sec.

Performing Program Operation...

0%............................................................................................................................................................................40%..........................................50%............................................................................................................................................................................90%..........................................100%

............Program Operation successful.

INFO: [Xicom 50-44] Elapsed time = 85 sec.

**Flash Operation Successful**

**To run the system in debug mode:**

1. Build the FSBL and software in debug mode
   1. Right Click on FSBL (example: MZ\_FSBL)
   2. Select Build Configurations > Set Active > Debug
   3. Right Click on Software (example: Lunah\_DevKit\_v1.0\_2014\_1)
   4. Select Build Configurations > Set Active > Debug
2. Create a debug configuration
   1. Run > Debug Configurations…
   2. Right click on Xilinx C/C++ application (GDB) from the left box and choose New
   3. For the bitstream file, browse to the \DevKit folder in your workspace, then to \hw\_platform\_0 and design\_1\_wrapper.bit should be your selection.
   4. Go to the Application tab and across from Project Name, click browse and select Lunah\_Devkit\_v1.0\_2014\_1
   5. For Application, click browse and navigate to the DevKit\Lunah\_Devkit\_v1.0\_2014\_1\Release folder where Lunah\_Devkit\_v1.0\_2014\_1.elf should be. Select this file and click open.
   6. The option to change the name of the configuration at the top of the box, this can be useful for future debug sessions.
   7. Click Apply then Debug to begin the session.
      1. As a shortcut, once the debug configuration has been created, press the “Bug” button in the toolbar to launch the currently selected debug configuration.
   8. As a default, SDK will stop at main(), so in the toolbar near the top of the screen press the green ‘Resume’ button (F8) to start the debug session.