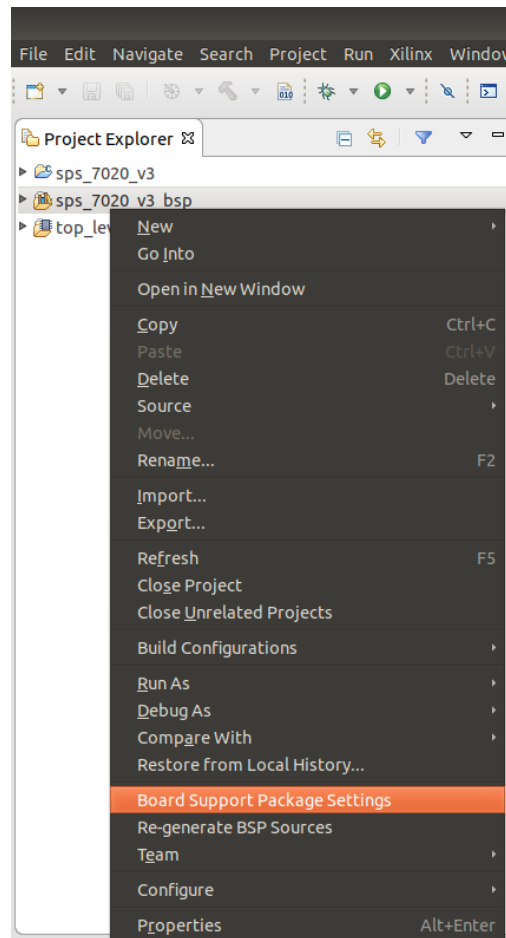


Booting Zynq-7000 from a SD Card

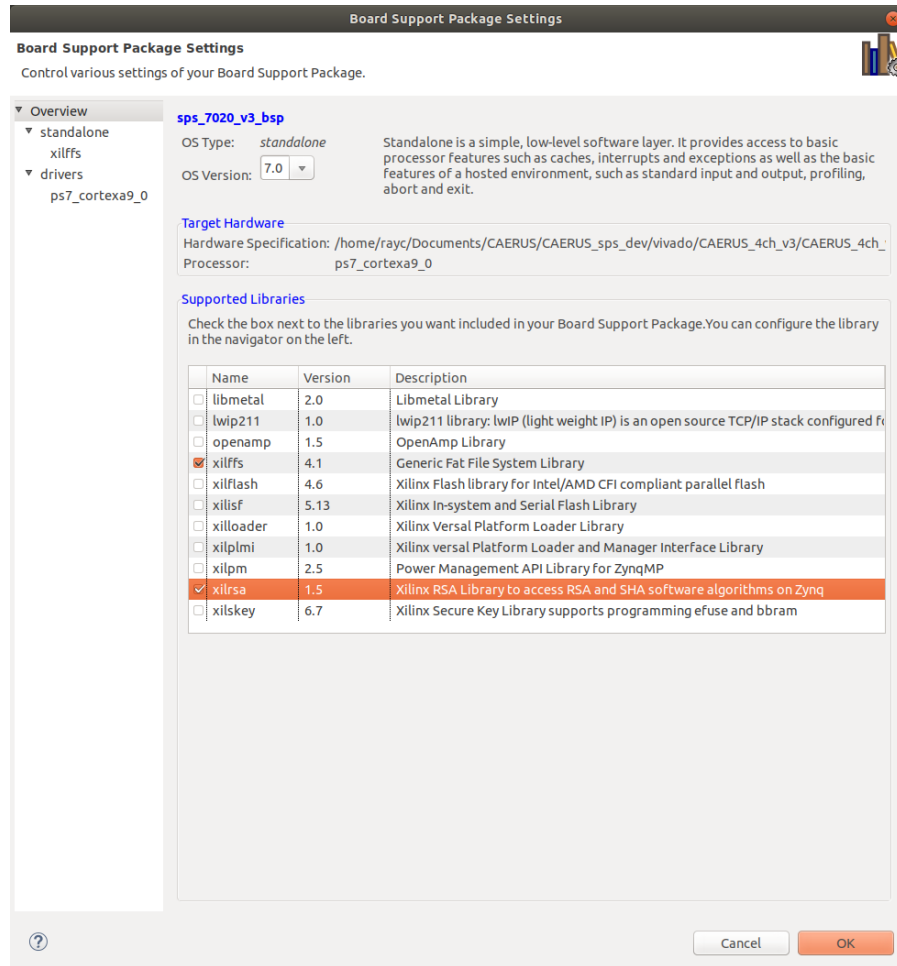
This guide will walk you through the process of creating the projects and files necessary for booting a Xilinx Vivado project from an SD card on a Zynq-7000 series FPGA.

Creating a First Stage Boot Loader and Boot Image

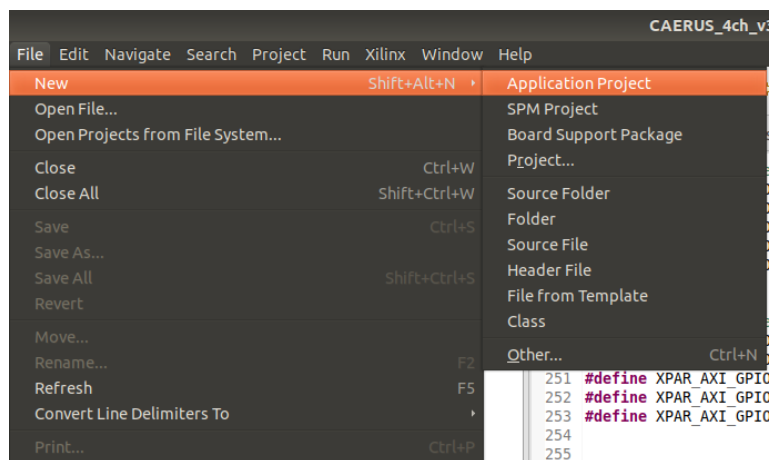
1. Open the Vivado project that you want to load onto an SD card, and open SDK from the project. It is assumed that at this point, the main C project, board support package project, and hardware platform project all work.
2. If xilrsa and xilffs are not enabled on the board support package settings, right click the bsp project and select *“Board Support Package Settings”*. If they are enabled, skip to step 4.



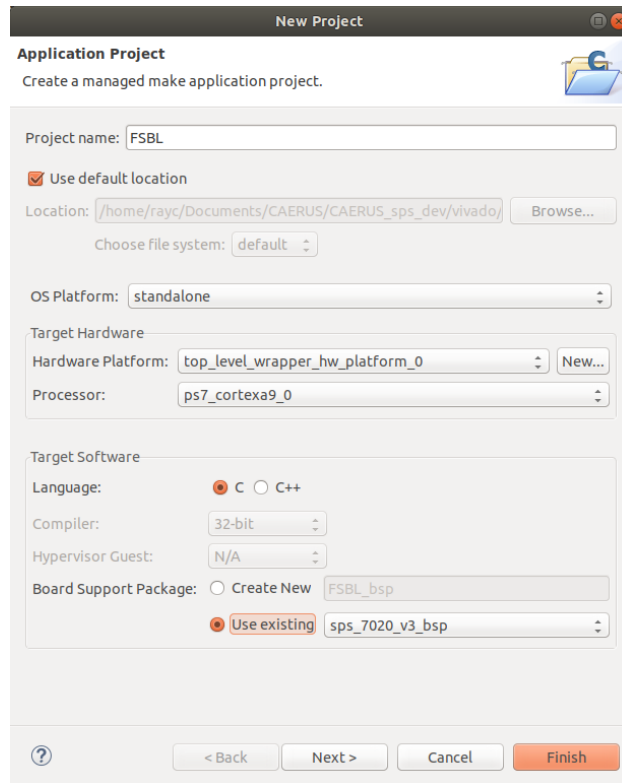
3. In the popup, check the boxes for xilrsa and xilffs then hit okay.



4. Create a new application project by clicking “File”→“New”→“Application Project”

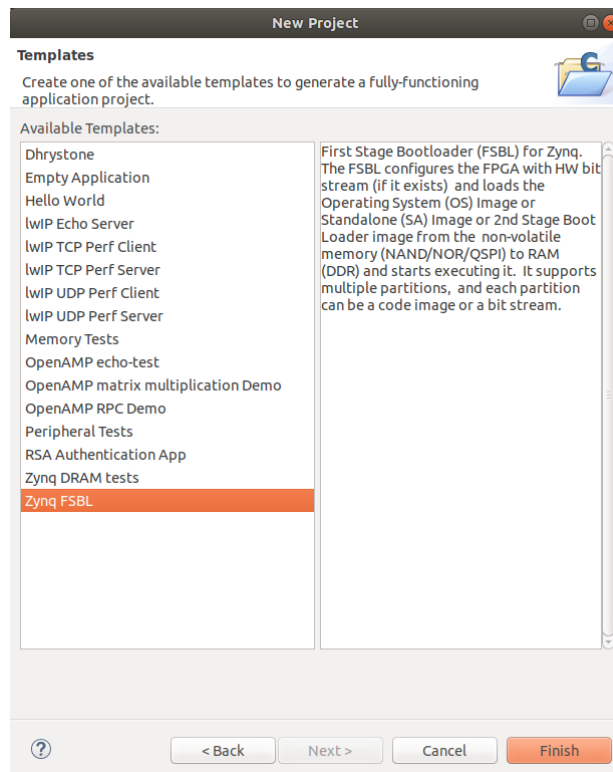


5. Name the project “FSBL” for first stage boot loader and change the “Board Support Package” to the bsp that you already have created.



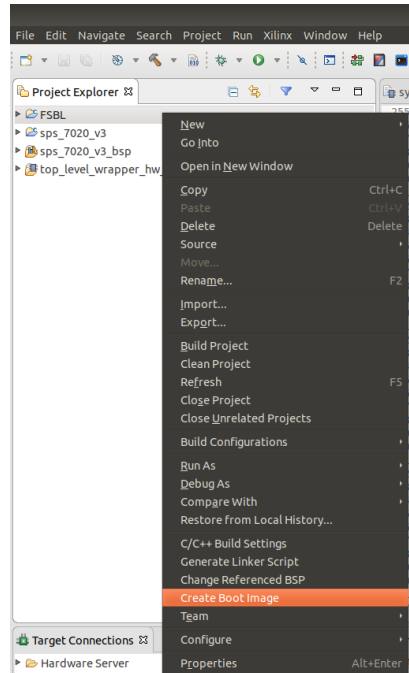
The screenshot shows the 'New Project' dialog box in Vivado. The 'Application Project' section is active, with the instruction 'Create a managed make application project.' The 'Project name' field is set to 'FSBL'. The 'Use default location' checkbox is checked. The 'Location' field shows the path '/home/rayc/Documents/CAERUS/CAERUS_sps_dev/vivado/'. The 'Choose file system' dropdown is set to 'default'. The 'OS Platform' dropdown is set to 'standalone'. The 'Target Hardware' section shows 'Hardware Platform' set to 'top_level_wrapper_hw_platform_0' and 'Processor' set to 'ps7_cortexa9_0'. The 'Target Software' section shows 'Language' set to 'C', 'Compiler' set to '32-bit', 'Hypervisor Guest' set to 'N/A', and 'Board Support Package' set to 'sps_7020_v3_bsp' (selected under 'Use existing'). The 'Finish' button is highlighted in orange.

6. Hit “Next >” then choose the “Zynq FSBL” project template.

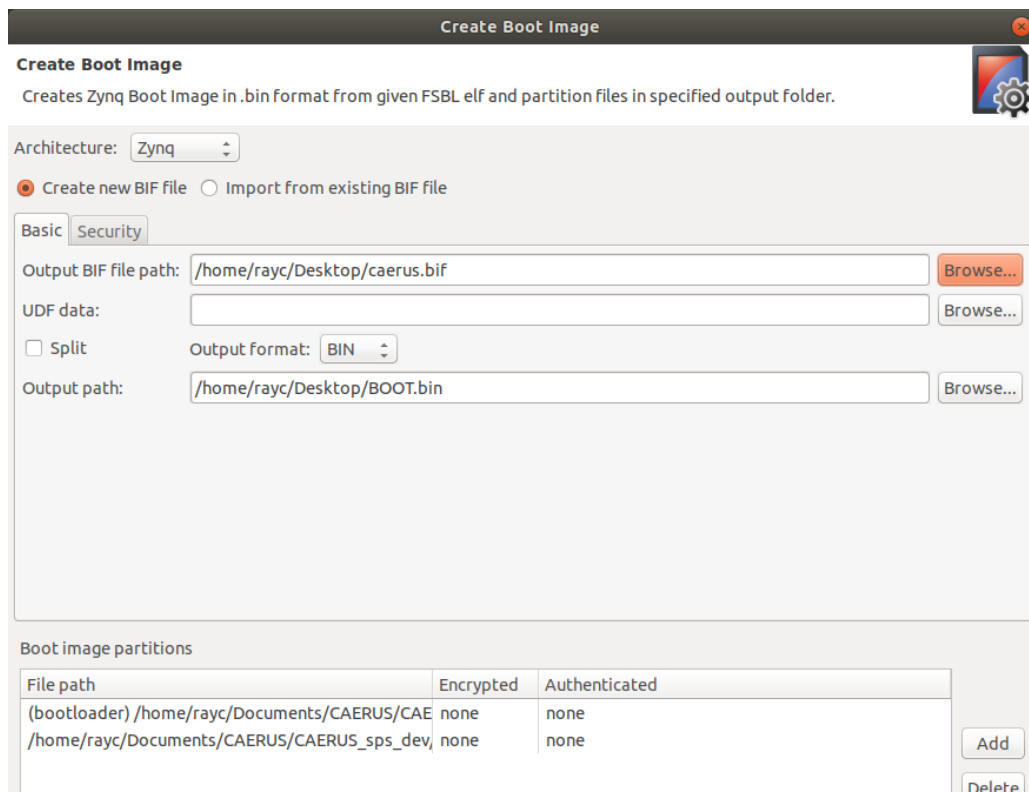


The screenshot shows the 'New Project' dialog box in Vivado, now displaying the 'Templates' section. The instruction is 'Create one of the available templates to generate a fully-functioning application project.' The 'Available Templates' list on the left includes: Dhystone, Empty Application, Hello World, lwIP Echo Server, lwIP TCP Perf Client, lwIP TCP Perf Server, lwIP UDP Perf Client, lwIP UDP Perf Server, Memory Tests, OpenAMP echo-test, OpenAMP matrix multiplication Demo, OpenAMP RPC Demo, Peripheral Tests, RSA Authentication App, Zynq DRAM tests, and Zynq FSBL (highlighted in orange). The description on the right for 'Zynq FSBL' reads: 'First Stage Bootloader (FSBL) for Zynq. The FSBL configures the FPGA with HW bit stream (if it exists) and loads the Operating System (OS) image or Standalone (SA) image or 2nd Stage Boot Loader image from the non-volatile memory (NAND/NOR/QSPI) to RAM (DDR) and starts executing it. It supports multiple partitions, and each partition can be a code image or a bit stream.' The 'Next >' button is highlighted in orange.

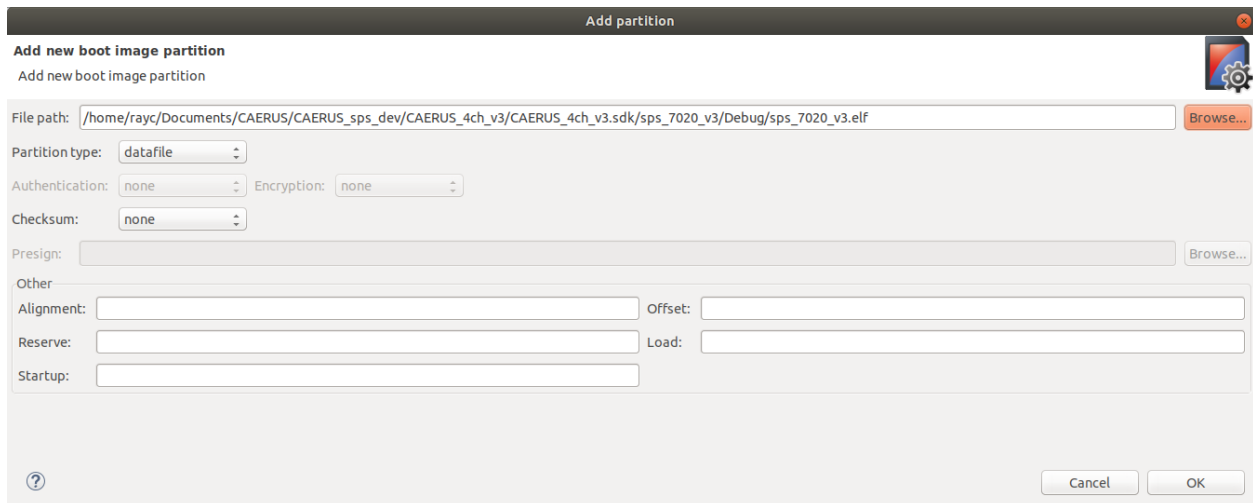
7. Click the “Finish” button.
8. Highlight the “FSBL” project, right click, and choose “Create Boot Image”



9. In the popup, choose a location for the output BIF file so you can easily find it. This example will place it on the desktop. The BOOT.bin file path will be automatically generated from the BIF file path.



10. By creating the boot image from the FSBL, the bootloader .elf file and the bitstream .bit file from the hardware platform will automatically be populated in the “*Boot image partitions*” section. If you do not create the boot image from the FSBL, you will have to manually populate these in the proper order. Hit the “*Add*” button to add another partition.

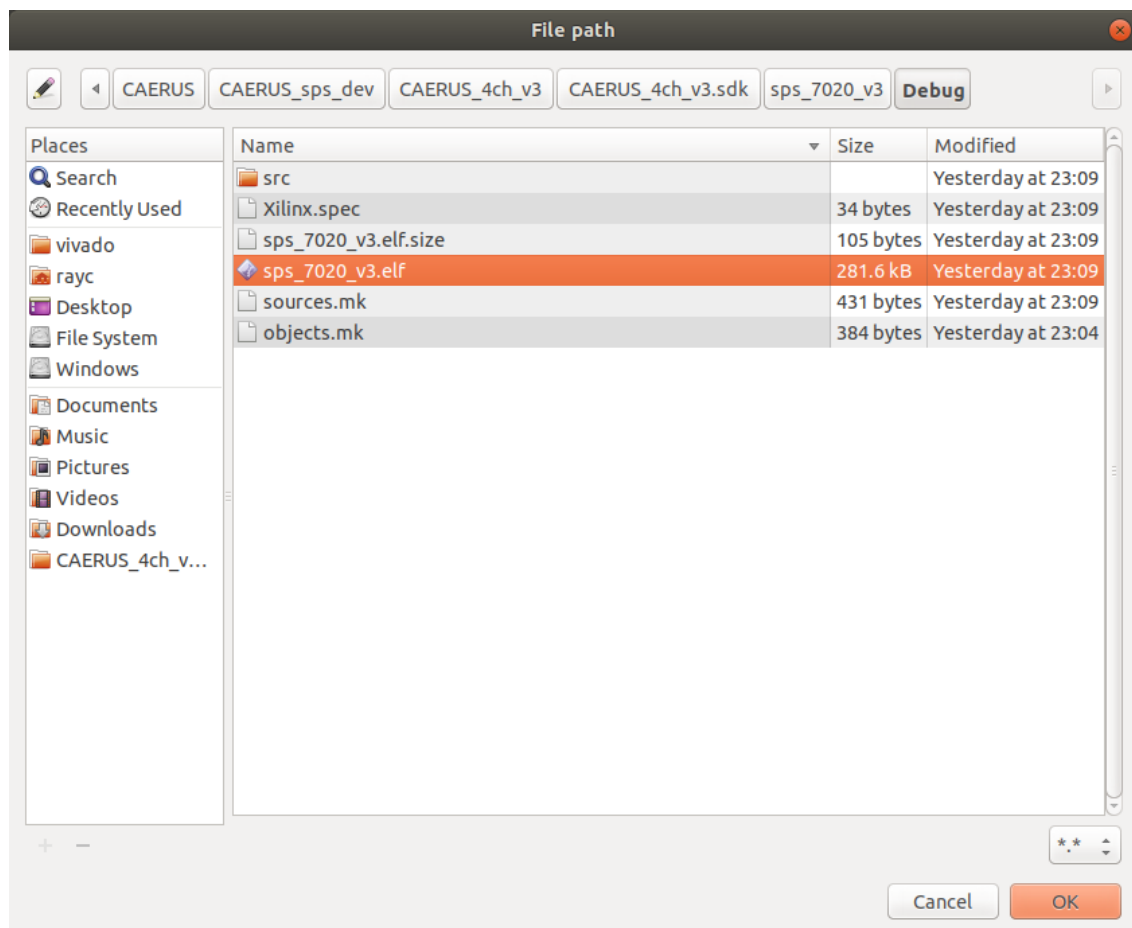


The screenshot shows the 'Add partition' dialog box. The title bar is 'Add partition'. Below the title bar, there is a sub-header 'Add new boot image partition'. The main area contains several fields and buttons:

- File path:** A text field containing the path `/home/rayc/Documents/CAERUS/CAERUS_sps_dev/CAERUS_4ch_v3/CAERUS_4ch_v3.sdk/sps_7020_v3/Debug/sps_7020_v3.elf`. To the right is a 'Browse...' button.
- Partition type:** A dropdown menu set to 'datafile'.
- Authentication:** A dropdown menu set to 'none'.
- Encryption:** A dropdown menu set to 'none'.
- Checksum:** A dropdown menu set to 'none'.
- Presign:** A text field with a 'Browse...' button to its right.
- Other:** A section containing four text fields: 'Alignment:', 'Offset:', 'Reserve:', and 'Load:'. The 'Startup:' field is also present but empty.

At the bottom right, there are 'Cancel' and 'OK' buttons. A help icon (?) is located at the bottom left.

11. Browse to the .elf file for the main C project and hit “OK”.



12. Make sure the .elf file is labeled as a datafile. Only the FSBL.elf file should be a bootloader. The bitstream should always come in between the two .elf files.

Create Boot Image

Create Boot Image

Creates Zynq Boot Image in .bin format from given FSBL elf and partition files in specified output folder.

Architecture: Zynq

Create new BIF file

Import from existing BIF file

Basic

Security

Output BIF file path: /home/rayc/Desktop/caerus.bif

Browse...

UDF data:

Browse...

☐ Split

Output format: BIN

Output path: /home/rayc/Desktop/BOOT.bin

Browse...

Boot image partitions

File path	Encrypted	Authenticated
(bootloader) /home/rayc/Documents/CAERUS/CAE	none	none
/home/rayc/Documents/CAERUS/CAERUS_sps_dev	none	none
/home/rayc/Documents/CAERUS/CAERUS_sps_dev	none	none

Add

Delete

Edit

Up

Down

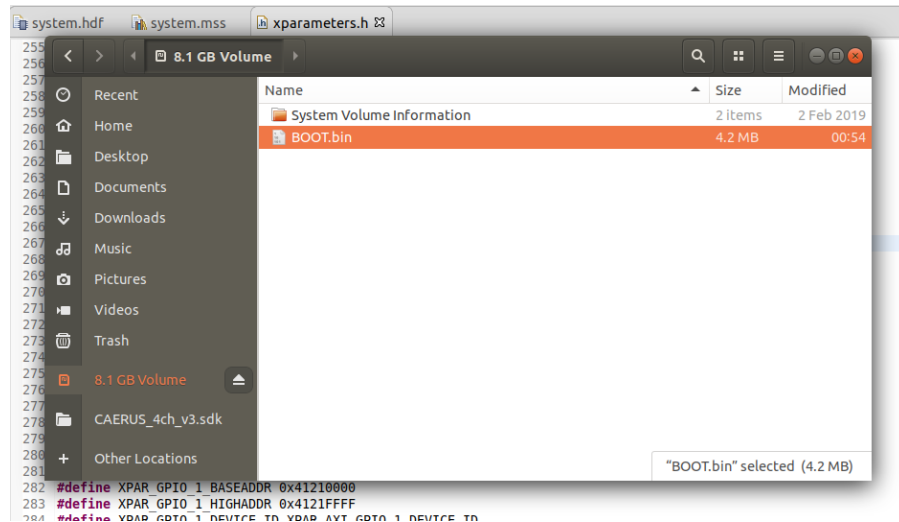
?

Preview BIF Changes

Cancel

Create Image

13. On your desktop, there should be two new files. Only the BOOT.bin file is needed. Copy it onto a microSD card with FAT formatting (if the card came from the ECE parts room with the board, then it is likely already programmed correctly).



14. Put the SD card in the FPGA. Ensure the jumper on the board is on SD and then power the board. If all goes well, the red LED should immediately power on, and after a few seconds, the green "Done" LED should activate.

Useful Links for SD Card Booting and Debugging

Zynq-7000 SoC: SD Programming/Booting Checklist

<https://www.xilinx.com/support/answers/59476.html>

SDK - How to debug FSBL code

<https://www.xilinx.com/support/answers/71671.html>

Files Necessary for Zynq Booting from SD Card

<https://forums.xilinx.com/t5/ACAP-and-SoC-Boot-and/Zynq-booting-from-SD-Card/td-p/470062>

Creating a Boot Image

https://www.xilinx.com/html_docs/xilinx2018_2/SDK_Doc/SDK_tasks/task_creatingabootimage.html

Creating a FSBL Project

https://www.xilinx.com/html_docs/xilinx2018_1/SDK_Doc/SDK_tasks/task_creatinganewzynqfsblapplicationproject.html