Development & Debug
EE332/493 Embedded Systems Hardware/Software Fall 2021
Lab #8
Andrew Catalano
8 December 2021
https://github.com/embedded-systems-2-master/lab-8-development-debug-acats525

Objective:

The purpose of this lab was to give an introduction to cross-compilation and debugging an user application. This lab was a guide to creating a simple application with PetaLinux tools and building this application by cross-compiling into the embedded Linux environment. Once the application was running, the lab instructed how to debug the application with the System Debugger

Theory of Operation:

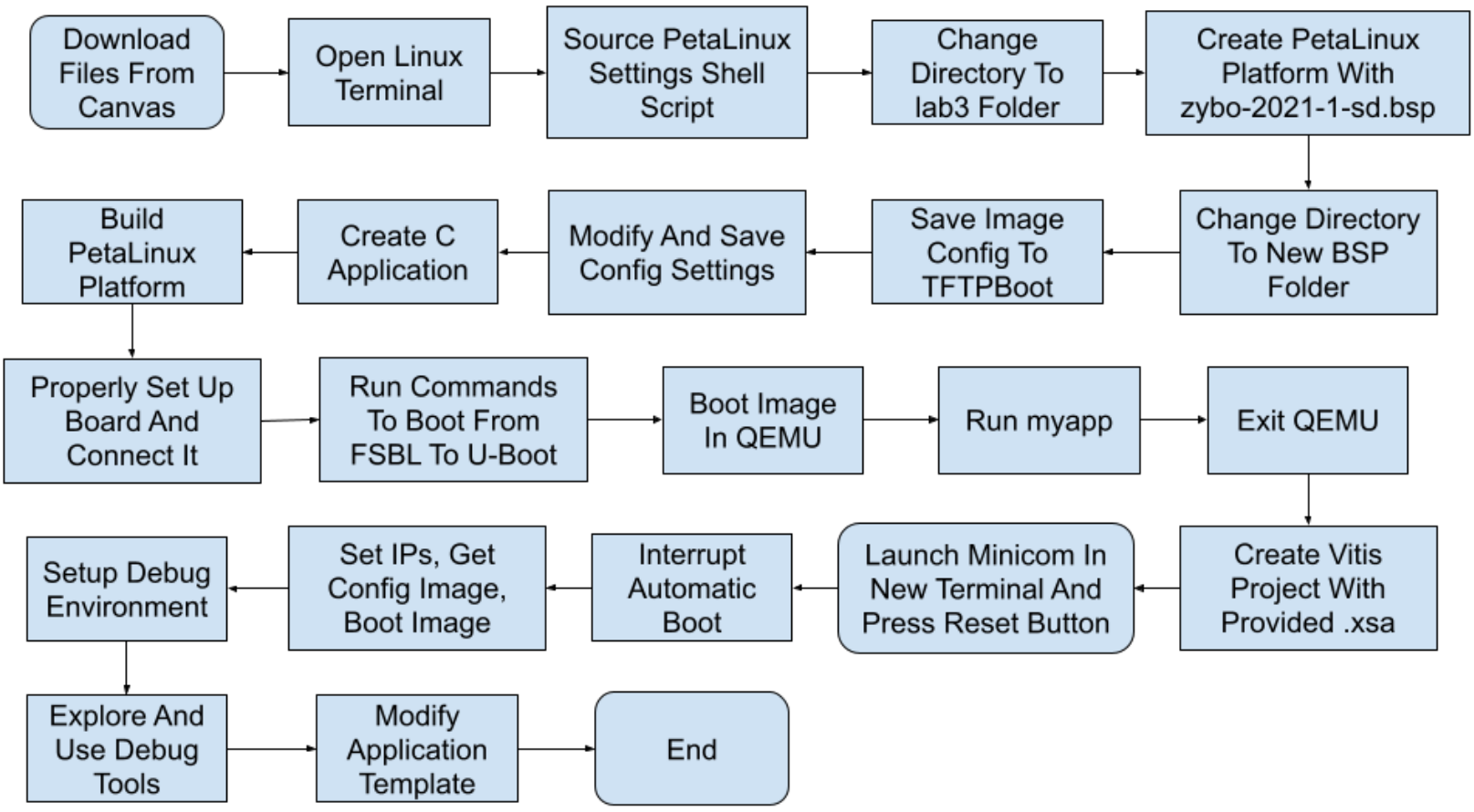

Figure 1. Flow Diagram

Development & Debug
EE332/493 Embedded Systems Hardware/Software Fall 2021
Lab #8
Andrew Catalano
8 December 2021
https://github.com/embedded-systems-2-master/lab-8-development-debug-acats525

Conclusion:

This lab was an interesting experience for me because it was a lot more involved than the previous two labs. Unfortunately, due to time constraints and the lab being outdated, I was not able to complete it to the end to demo the debugging part and modifications to the C program part.

This lab began like Lab 7 by downloading the same folder from Canvas, opening a terminal (which I used for the host machine) and I ran the settings shell script (settings.sh) with the source Linux command (this shell script is the one for PetaLinux). I changed directories to the lab3 folder, specified by the lab, and then I created a new PetaLinux platform with zybo-2021-1-sd.bsp as my board support package. I changed directories to the new folder that was created named zybo-2021-1. I saved the image configuration to tftpboot and then I followed the instructions to do all the configuration settings and saved them. I created a C application with the command provided in the lab, made sure to locate it, and then began the build process which took about 30 minutes.

Once done, I set up the board properly with the MicroSD card and necessary jumper tags and ran two more commands which I believe helped boot the image from the first stage boot loader to u-boot. I then booted the image to qemu, located my application which was called myapp, and then ran it which printed “Hello World” to the terminal. I exited qemu and proceeded by opening Vitis and creating a new application with the .xsa file provided by the petalinux-create command.

From here I pressed the reset button on the Zybo board, interrupted the boot process by pressing a key on the keyboard. I followed a series of Linux commands that set the server IP to 192.168.1.1 (which I assumed is the host), set the IP address to 192.168.1.2 (which I assumed was the board), get the config file from tftpboot I saved earlier, and then boot the device with the new image and config settings.

Unfortunately, this is where I had to stop since I couldn’t set up the debug environment since the version in the lab was older than what I was using. From what I could gather from reading the rest of the lab, we had to set up the debug environment to be for a Linux application and set the Host field to be the server IP address I mentioned earlier. From there it looked like the application file paths had to be configured and then the debug window was opened for use. The lab goes over some of the basics of the debugger by showing local variables in one tab and explaining how to use breakpoints, which pause execution of the program at the breakpoint. These can be very useful when narrowing in on a problem in source code. The rest of the lab was customizing the application template and making different strings print out in the terminal. Overall, I think this lab was a nice look into embedded Linux.