Getting Started With PetaLinux

by **NAEastland** on July 15, 2016

Table of Contents

Intro: Getting Started With PetaLinux

Interested in learning a little about embedded Linux? Have you worked with Xilinx FPGAs and want to explore some of the software related to their implementation? If you answered yes, then welcome! If you answered no (to at least the first question) then you should read through anyway, because this is pretty cool stuff.

Welcome to this Getting Started Guide (GSG) to PetaLinux!

PetaLinux is an embedded Linux development solution for Xilinx Zynq chips (an ARM processor with FPGA material, like the ones used here and here) as well as for MicroBlaze designs implemented in fully FPGA chips. This GSG will be using the 2015.4 versions of PetaLinux, SDK, and Vivado and will be targeting the Zybo from Digilent, so we will be implementing a project on a Zynq target rather than the soft processor core of a MicroBlaze design. For some more info on MicroBlaze, check out this guide.

For this Instructable, the following prerequisites apply:

- Some familiarity with Linux
- A Zybo or Zedboard to deploy the project onto
- A Linux machine (VM or dual boot setup) of supported OS: Ubuntu 14.04, CentOS 7, SUSE Enterprise 12, RHEL 6.5/6.6/7. *These are for PetaLinux 2015.4. Newer versions may support more recent OS versions.
- At least 20GB of free hard drive space
- Have Vivado and Xilinx SDK installed (the version you have installed should be the version of PetaLinux you will download)

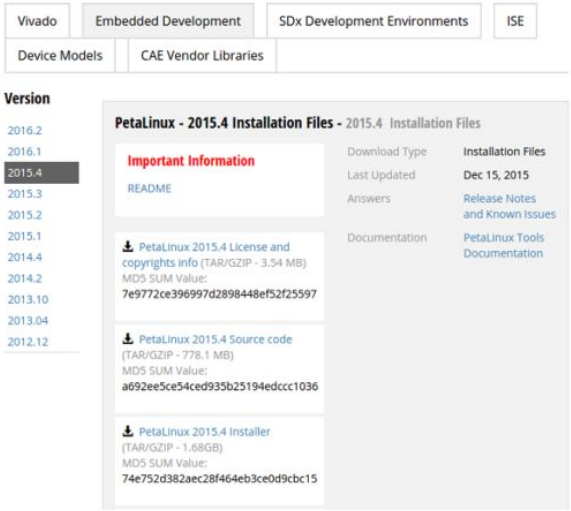
*Note: While I was able to get PetaLinux 2015.4 running within Ubuntu 16.04, this is not a supported version and is not recommended.

This guide will walk you through the steps of initial setup for PetaLinux on your Linux machine and the implementation of a pre-built design on the Zybo. By the end of this Instructable, you will be able to begin tinkering with the design from either the hardware description language (HDL) side, or create custom applications you can run within the Linux system running on your board. Lets get started!

Step 1: Download Installer

You will need to download the installer for PetaLinux of the same version as the Vivado and SDK installation you should have on your system. Meaning if you have Vivado and SDK 2015.4 installed, you should download PetaLinux 2015.4. The download may take a while.

Once the download has completed, make a directory in which you would like the PetaLinux tools to be installed in. From your terminal, change directory (cd) into the directory the installer was downloaded into (likely Downloads) and run the installer with a specified path to the directory you just created.



Step 2: Install PetaLinux

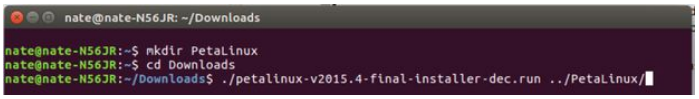
Once the download has completed, make a directory in which you would like the PetaLinux tools to be installed in. In your terminal, change directory (cd) into the directory the installer was downloaded into (likely Downloads) and run the installer with a specified path to the directory you just created. Starting from your home directory, enter the following commands (also in screenshot above).

mkdir PetaLinux

cd Downloads

./petalinux-v2015.4-final-installer-dec.run ../PetaLinux

This is just one option of installation location. You can install PetaLinux into any (typical) directory and it will work fine, as long as you have the tools sourced correctly, which we will cover. The PetaLinux tools have an end user licence agreement (EULA) as part of the tool usage, however PetaLinux does not require a license from Xilinx to run.



Step 3: Download Dependencies

As with many software development tools, there are a variety of dependencies that you will need to have in order for PetaLinux to operate. Many of the packages may already be installed on your computer, but some may not. A full list of the dependencies is included below (for Ubuntu).

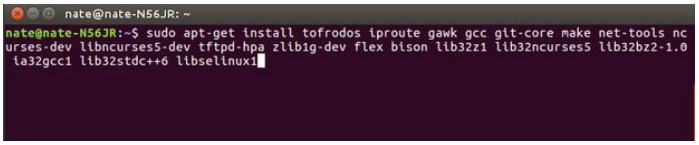
- tofrodos
- iproute
- gawk
- gcc
- git-core
- make
- net-tools
- ncurses-dev
- libncurses5-dev
- tftpd*
- zlib1g-dev
- flex
- bison
- lib32z1
- lib32ncurses5
- lib32bz2-1.0
- ia32gcc1
- lib32stdc++6
- libselinux1

PetaLinux operates using dependencies on 32-bit libraries, so including those specific libraries indicated is required in order for it to operate correctly (even if some may seem redundant). In addition, PetaLinux can utilize a tftp server to streamline the development process, but it is not a requirement for it to function. The specified server package to use is tftpd, however I used tftpd-hpa and would suggest you do the same. If you are unfamiliar with the process of setting it up, you can take a look at this quick Instructable.

Downloading these is the same as any other module that you can get through apt-get, but when you make the following function call (or see image above) you can simply enter each module's name in the same line with spaces between.

sudo apt-get install tofrodos iproute gawk (etc)

The list of dependencies for other Linux distributions can also be found in the PetaLinux documentation, starting on page 10.



Step 4: Set up Bash and Source Settings

The PetaLinux tools require you to use 'bash' as your shell rather than 'dash', which is likely your default shell if you're running Ubuntu. To change this, just enter the following command, which will set your default from 'dash' to 'bash'.

sudo dpkg-reconfigure dash

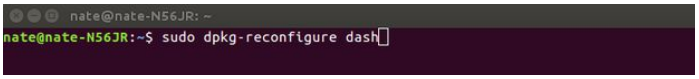
The next thing to take care of will be to source the tools for PetaLinux to use within the terminal window. This includes the 'settings64.sh' and 'settings.sh' files in your Vivado and PetaLinux installation directories, respectively. To avoid needing to type the source commands into the shell every time, you can add a couple lines to the .bashrc script. To modify this system wide, use a text editor to open your .bashrc file. For Ubuntu, this will be bash.bashrc located in the /etc directory (see following command and/or first image above).

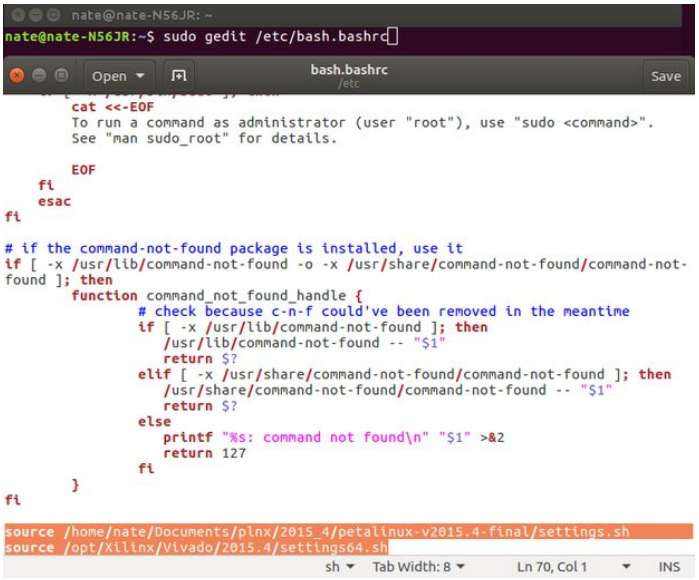
sudo gedit /etc/bash.bashrc

Once you have the script open, add the two commands for sourcing the appropriate files. Note that the path indicated here is simply where my installation directories are, so your specific file path may likely be different.

source /home/nate/Documents/plnx/2015_4/petalinux-v2015.4-final/settings.sh
source /opt/Xilinx/Vivado/2015.4/settings64.sh

With the two lines added, save the changes and close the editor.



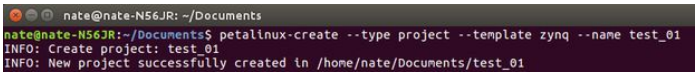


Step 5: Create a Project - New Project (optional)

To use PetaLinux, you will need a PetaLinux project directory to work in. This can be done either by creating a totally new project or by using a reference design provided in a board support package (BSP). Creating a fresh project provides you with a basic template from which you can start your development. Just change to a directory you would like to create your project in and enter the following command.

petalinux-create --type project --template zynq --name test_01

The '--type' parameter should remain 'project', the '--template' parameter should be whatever supported architecture you are targeting (either zynq, zynqMP for Ultrascale chips, or microblaze for soft processors implemented in FPGA fabric), and the '--name' parameter can be whatever you want to name your project. Do note that this simply provides a folder structure for PetaLinux to use and requires you to provide pretty much every part of the build, from the first stage boot loader to the file system, and is not suggested for those new to Linux development. New players should instead use a BSP!

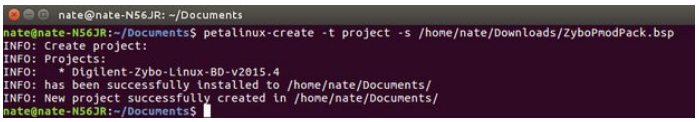


Step 6: Create a Project - Board Support Package

Creating a new project from a BSP is the simplest way to get started with PetaLinux, since it provides you with an already functioning and bootable Linux image that you start playing with. There are several BSPs available for download from Xilinx, as well as a Digilent BSP for the Zybo. Once you have the BSP of your choosing downloaded (and extracted if it was a zip file), go to your terminal and change directory to where you would like to create your new PetaLinux project directory and enter the following command.

petalinux-create -t project -s

Here '-t' is equivalent to '--type' described in the previous step (its parameter should remain 'project') and '-s' is for source and should be followed with the absolute file path to the BSP you want to use. (i.e. /home/nate/Downloads/ZyboPmodPack.bsp).



Step 7: Configure, Build, and Package

This step is very straight forward from an end user's perspective, but will require you to accept a bit of 'magic' in the background if you are not intimately familiar with the process of compiling a Linux image from scratch. Suffice it to say that by the end of the configure and build process in PetaLinux, you will have a kernel, file system, first stage and second stage boot loaders, and device tree compiled and ready to be deployed to your hardware target. To run configuration on the BSP project you just created, change directory into the directory that was made with the 'petalinux-create' command, and type in the following.

petalinux-config

This will initialize a configuration menu for your PetaLinux project (see first image above).Make sure your terminal window is at least its default dimensions or the menu will fail to launch. There are a variety of boot options available to you depending your application. Since the setup and operation of each of the different boot methods is a bit involved, it will be covered in a separate guide. If you are using the Digilent BSP then the default values in the configuration menu will be fine, so just select the 'Exit' option to leave the config menu. The configuration process will then continue on (this will take some time).

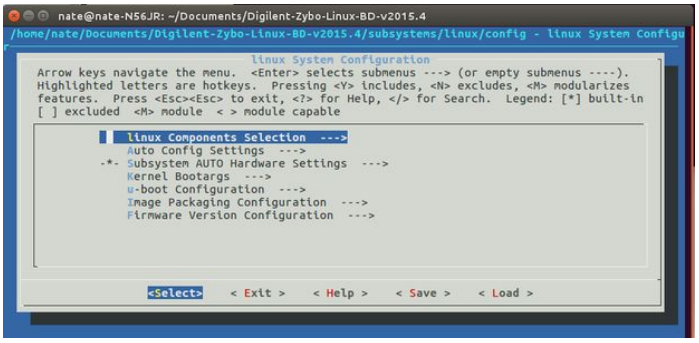
Once the configuration is complete, you will need to build your image by entering the following command.

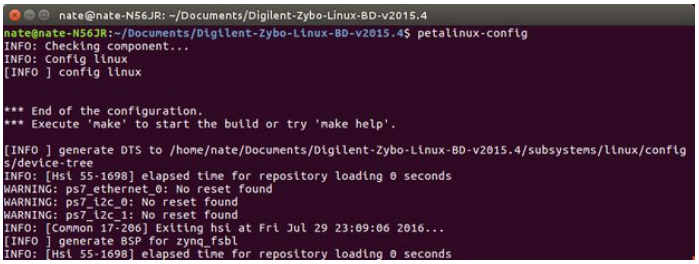
petalinux-build

The execution of this command will also take a few minutes to complete, depending on your system. Once this is finished (again, running the Digilent BSP), enter the following command.

petalinux-package --boot --force --fsbl ./images/linux/zynq_fsbl.elf --fpga ./images/linux/linux_bd_wrapper.bit --u-boot

After this completes you should have your BOOT.bin and U-boot files ready to go.





Step 8: Load onto SD card

The SD you use will need to have two partitions on it. I recommend an 8GB card with the first partition (your BOOT partition) formatted as a **File Allocation Table** (**FAT**) of **1GB**, and the second partition (your rootfs partition) formatted as an **ext4** for the remaining space on the card.The FAT partition will be where your BOOT.bin and image.ub will be stored, while the second partition will be where the file system for your Linux image will be. You can use a utility such as fdisk or gparted in your terminal to do the formatting for your SD card.

Copy these two files into the first partition of your SD card. You can do this from your PetaLinux project root directory with the following commands.

cp images/linux/BOOT.bin /media/BOOT

cp images/linux/image.ub /media/BOOT

Once your boot files have been copied into the BOOT partition of your SD card, copy the root file system into the second partition, your 'rootfs' partition, with the following command.

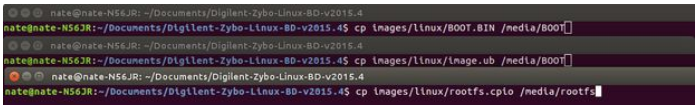
cp images/linux/rootfs.cpio /media/rootfs

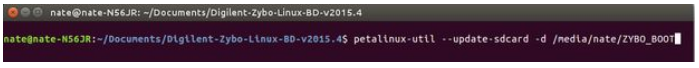
*Note: the paths "/media/BOOT" and "/media/rootfs" may not be where your SD card is mounted, so you should modify the copy command accordingly.

Once the root file system and boot files have been copied to the two partitions of your SD card run the following PetaLinux command, still from the root directory of the PetaLinux project (also in the second image above).

petalinux-util --update-sdcard -d /media/nate/ZYBO_BOOT

With that last command entered in you can now unmount and eject your SD card from your computer and load it into your Zybo. You have now just completed this GSG for PetaLinux, congrats!





Related Instructables

Setting Up TFTP Server for PetaLinux by NAEastland

Embedded Linux Tutorial - Zybo by Commanderfranz

An embedded vision system supporting the home care for convalescent or elderly people by AVADER

The Pathfinder Zybot by ErdÅ‘s Zákány

Quick Start Test Demo: Zybo (Xilinx Zynq 7000) Image Filtering Demo + GoPro by LariSan

Fluid Spectrum Analyser Equipment by emilhun

Comments