
Linux Development on the Zedboard

Εισαγωγή

Σε αυτό το εργαστήριο θα χρησιμοποιήσουμε το Petalinux SDK ver. 2015.4 για να αναπτύξουμε ένα σύστημα Linux για το Zedboard και μια εφαρμογή που θα τρέχει στο Zedboard.

Το Petalinux SDK είναι ένα open-source λογισμικό από την Xilinx, το οποίο παρέχει τα κατάλληλα εργαλεία για να διευκολύνει έναν developer στην ανάπτυξη ενός πλήρους λειτουργικού συστήματος Linux για Ενσωματωμένα Συστήματα.

Το Petalinux SDK λειτουργεί με command-based διεπαφή. Ο developer χρησιμοποιεί εντολές από ένα terminal για να δημιουργήσει projects, εφαρμογές, να παραμετροποιεί το σύστημα και να δημιουργεί Boot images και Filesystem.

Στόχοι

Μετά την ολοκλήρωση αυτού του εργαστηρίου, θα είστε σε θέση να:

- Αναπτύξετε ένα δικό σας Linux Distribution με βασικές λειτουργικότητες για το Zedboard.
- Δημιουργήσετε custom εφαρμογές για το λειτουργικό σας

Διαδικασία

Για το παρόν εργαστήριο, δεν θα αναπτύξουμε κάποιο Hardware Platform. Το Petalinux παρέχει τη δυνατότητα ανάπτυξης Linux OS για διάφορα SoCs. Θα χρησιμοποιήσουμε ένα έτοιμο BSP φτιαγμένο για δημιουργία Petalinux distribution συγκεκριμένα για το Zedboard.

Συγκεκριμένα

- 1) Θα δημιουργήσουμε ένα Petalinux Project
- 2) Θα διαρυθμίσουμε το project αυτό για να επιτρέψουμε μεταφορά δεδομένων μέσω Ethernet και δημιουργία FileSystem σε SD card.
- 3) Θα δημιουργήσουμε Boot Images και Root FileSystem
- 4) Θα μορφοποιήσουμε μία SD card για να μεταφέρουμε τα Boot Images και το File System που δημιουργήσαμε, στην κάρτα
- 5) Θα αναπτύξουμε μία εφαρμογή για το Linux που θα δημιουργεί 100 τυχαίους integers και θα τους ταξινομεί με αύξουσα σειρά.
- 6) Θα μεταφέρουμε την εφαρμογή μας μέσω Ethernet στο Zedboard και θα την εκτελέσουμε για να ελεγχουμε ότι δεν έχει σφάλματα.

1.1. Create a project using the Avnet-Digilent-Zedboard-2015.4.bsp as source

1.1.1. Open a terminal and type

```
cd ~/Petalinux_Projects
```

1.1.2. Next, type the following command to source the Petalinux SDK settings.sh file and make the build and packaging commands of Petalinux available to us.

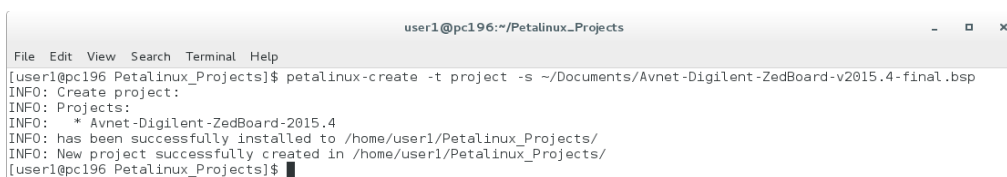
```
source /opt/Petalinux/petalinux-v2015.4-final/settings.sh
```



```
user1@pc196:~/Petalinux_Projects
File Edit View Search Terminal Help
[user1@pc196 Petalinux_Projects]$ source /opt/Petalinux/petalinux-v2015.4-final/settings.sh
Petalinux environment set to '/opt/Petalinux/petalinux-v2015.4-final'
INFO: Checking free disk space
INFO: Checking installed tools
INFO: Checking installed development libraries
INFO: Checking network and other services
[user1@pc196 Petalinux_Projects]$
```

1.1.3. Type the following command to create a Petalinux Project

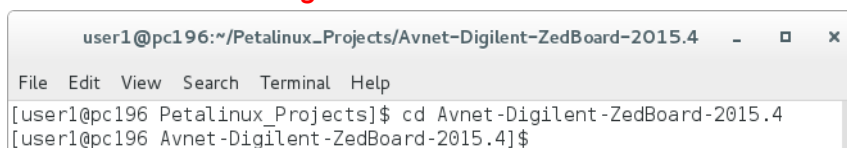
```
petalinux-create -t project \  
-s ~/Documents/Avnet-Digilent-Zedboard-v2015.4-final.bsp
```



```
user1@pc196:~/Petalinux_Projects
File Edit View Search Terminal Help
[user1@pc196 Petalinux_Projects]$ petalinux-create -t project -s ~/Documents/Avnet-Digilent-ZedBoard-v2015.4-final.bsp
INFO: Create project:
INFO: Projects:
INFO: * Avnet-Digilent-ZedBoard-2015.4
INFO: has been successfully installed to /home/user1/Petalinux_Projects/
INFO: New project successfully created in /home/user1/Petalinux_Projects/
[user1@pc196 Petalinux_Projects]$
```

1.1.4. Change current directory to the created Petalinux Project directory

```
cd Avnet-Digilent-ZedBoard-2015.4
```



```
user1@pc196:~/Petalinux_Projects/Avnet-Digilent-ZedBoard-2015.4
File Edit View Search Terminal Help
[user1@pc196 Petalinux_Projects]$ cd Avnet-Digilent-ZedBoard-2015.4
[user1@pc196 Avnet-Digilent-ZedBoard-2015.4]$
```

1.1.5. Execute the following command to build the project.

```
petalinux-build -v
```

1.1.6. Execute this command to create boot images for your project

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

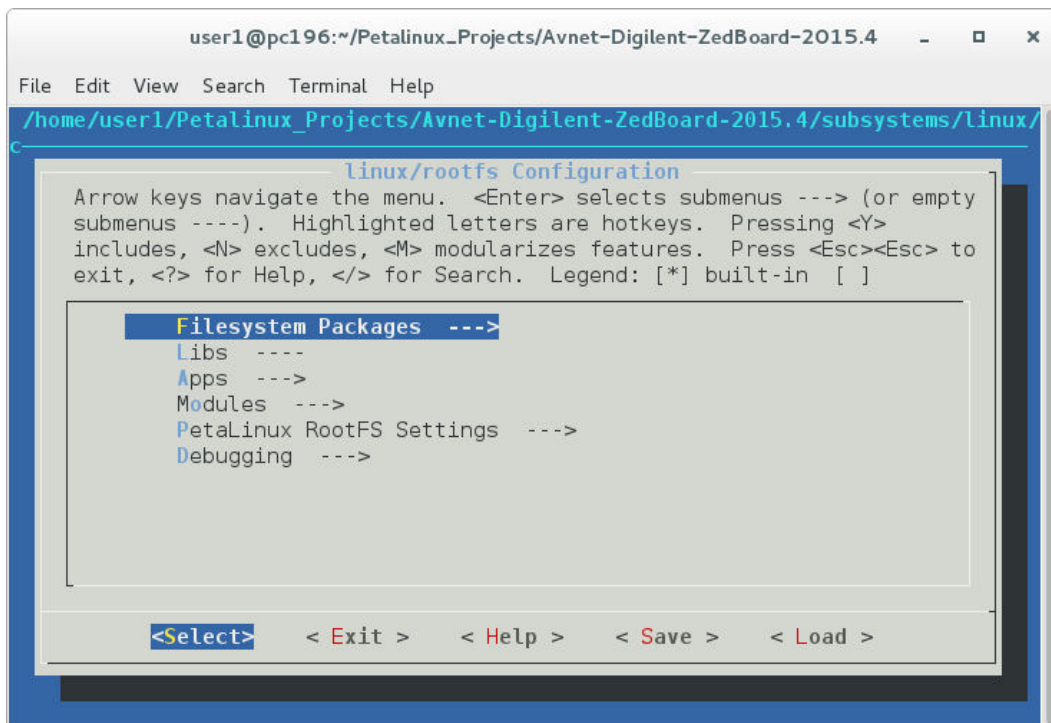
2.1. Configure the created Petalinux project to allow communication via Ethernet with the Host development PC.

2.1.1 While in the root directory of your created Petalinux project, type the following command

```
petalinux-config -c rootfs
```

```
user1@pc196:~/Petalinux_Projects/Avnet-Digilent-ZedBoard-2015.4 -  
File Edit View Search Terminal Help  
[user1@pc196 Avnet-Digilent-ZedBoard-2015.4]$ petalinux-config -c rootfs
```

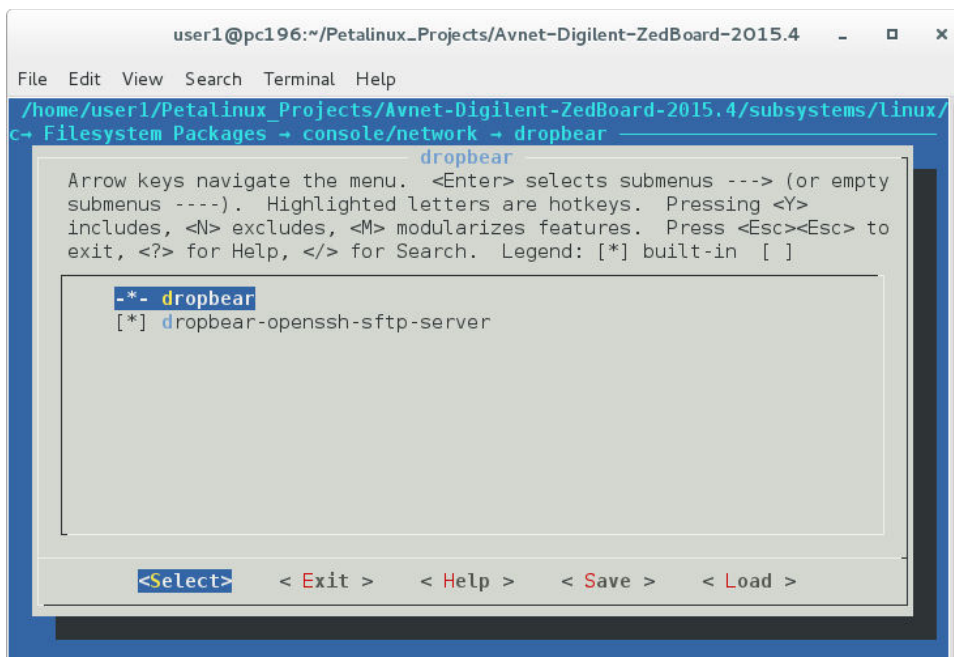
2.1.2 You should see the following screen appear



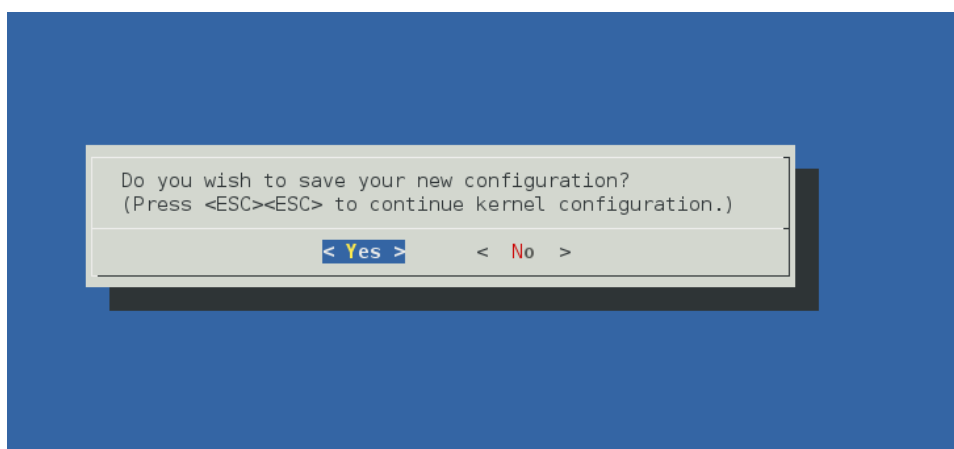
2.1.3 Using the arrow keys and enter, navigate to

```
Filesystem Packages ---> console/network ---> dropbear ---> - - dropbear  
[ ] dropbear-openssh-sftp-server
```

And use the <Y> key to add the asterisk to both those options. This will include the Dropbear package for your Petalinux distribution, allowing you to set up SSH on the Zedboard and securely send files from and to your PC workstation.



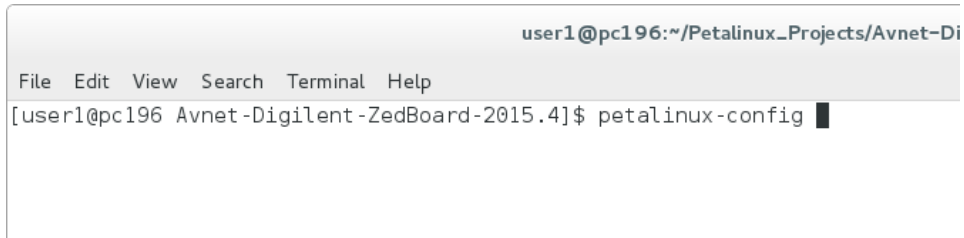
Select exit option as many times as needed to exit the configuration. When prompted to save the new configuration, select yes and press Enter.



2.2. Configure the created Petalinux project to boot from an SD card.

2.2.1 Type the following command while in the root directory of your project

petalinux-config

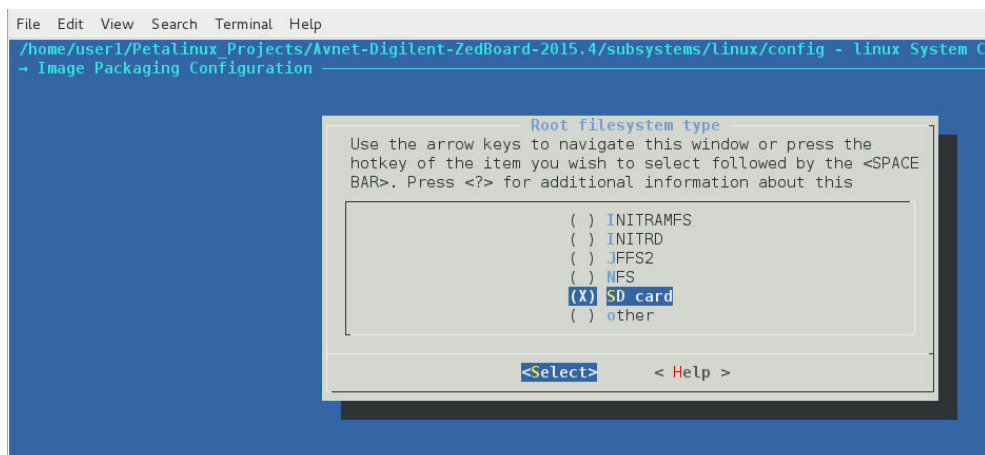


```
user1@pc196:~/Petalinux_Projects/Avnet-Di
File Edit View Search Terminal Help
[user1@pc196 Avnet-Digilent-ZedBoard-2015.4]$ petalinux-config
```

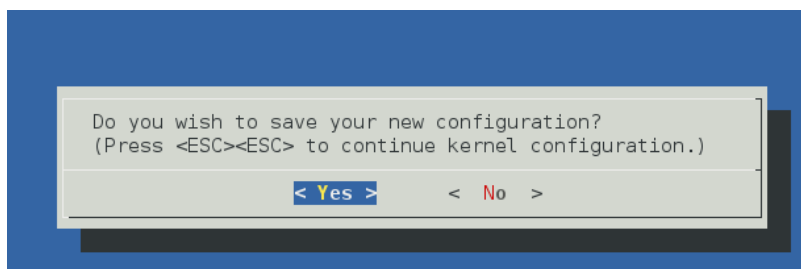
2.2.2 Navigate to

- Image Packaging Configuration ---> Root filesystem type --->
- () INITRAMFS
 - () INITRD
 - () JFFS2
 - () NFS
 - () SD card
 - () other

And select SD card as the root filesystem type



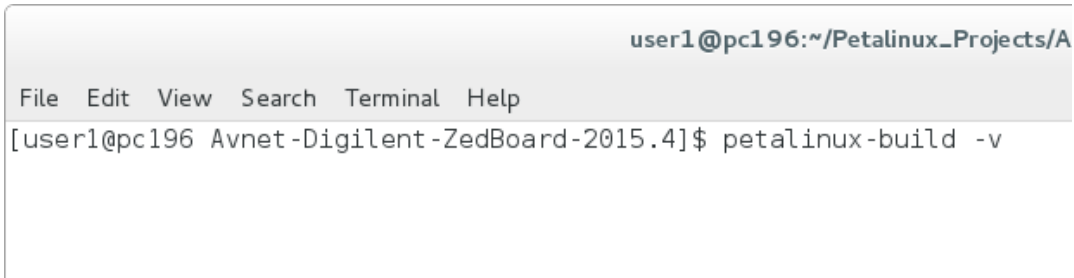
2.2.3 Exit and save the new settings



3.1. Build and package the project

3.1.1. While in your project's root directory, in a terminal type the following command to build the project and create the necessary boot images.

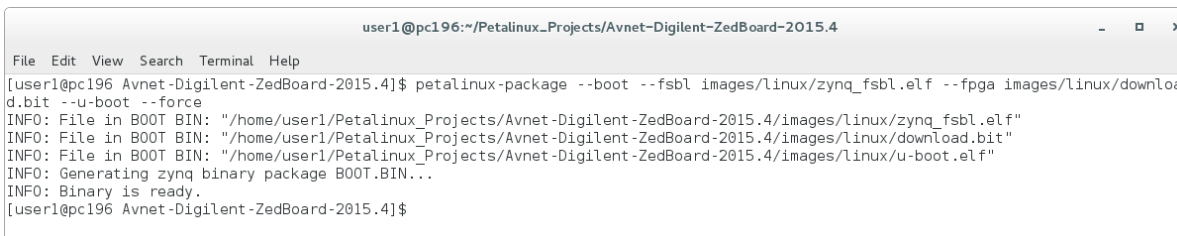
```
petalinux-build -v
```



```
user1@pc196:~/Petalinux_Projects/A
File Edit View Search Terminal Help
[user1@pc196 Avnet-Digilent-ZedBoard-2015.4]$ petalinux-build -v
```

3.1.2. After the build process is over, execute the following command to generate the boot image files

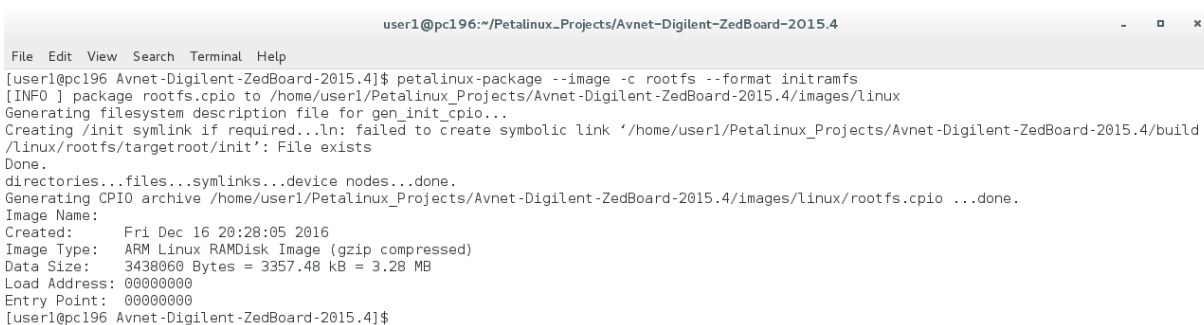
```
petalinux-package --boot --fsbl images/linux/zynq_fsbl.elf --fpga
images/linux/download.bit --u-boot --force
```



```
user1@pc196:~/Petalinux_Projects/Avnet-Digilent-ZedBoard-2015.4
File Edit View Search Terminal Help
[user1@pc196 Avnet-Digilent-ZedBoard-2015.4]$ petalinux-package --boot --fsbl images/linux/zynq_fsbl.elf --fpga images/linux/download.bit --u-boot --force
INFO: File in BOOT BIN: "/home/user1/Petalinux_Projects/Avnet-Digilent-ZedBoard-2015.4/images/linux/zynq_fsbl.elf"
INFO: File in BOOT BIN: "/home/user1/Petalinux_Projects/Avnet-Digilent-ZedBoard-2015.4/images/linux/download.bit"
INFO: File in BOOT BIN: "/home/user1/Petalinux_Projects/Avnet-Digilent-ZedBoard-2015.4/images/linux/u-boot.elf"
INFO: Generating zynq binary package BOOT.BIN...
INFO: Binary is ready.
[user1@pc196 Avnet-Digilent-ZedBoard-2015.4]$
```

3.1.3. Execute the following command to create the Linux Root Filesystem file

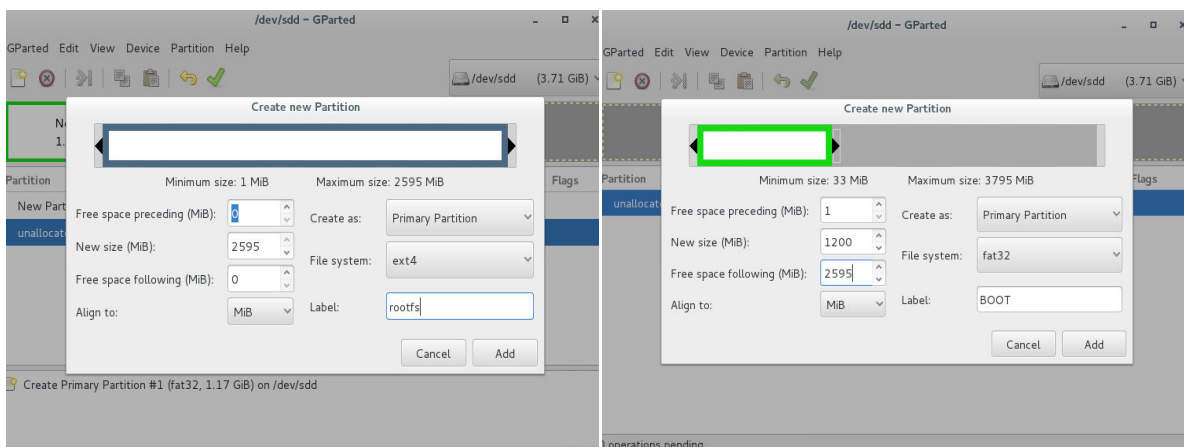
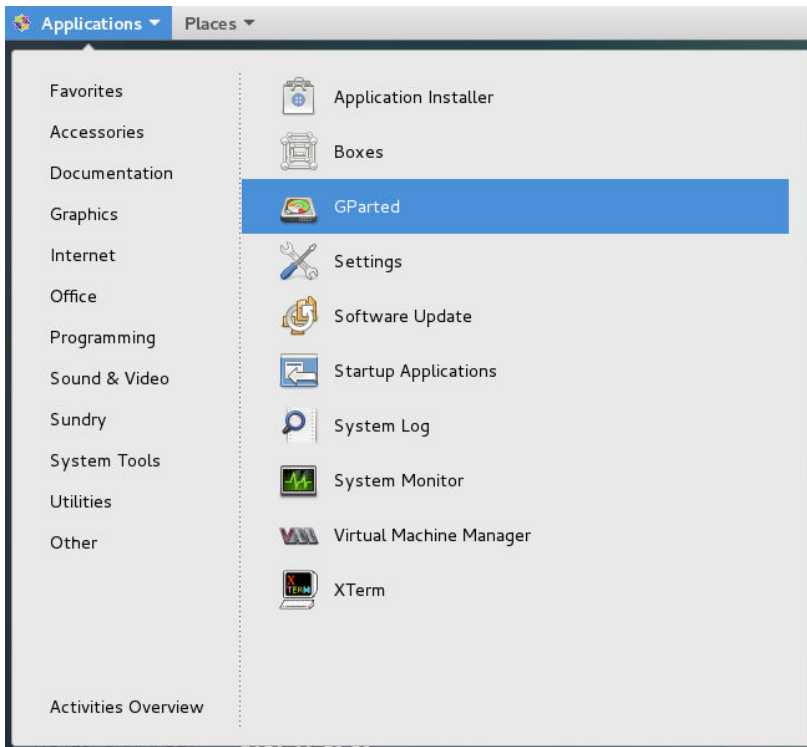
```
petalinux-package --image -c rootfs --format initramfs
```



```
user1@pc196:~/Petalinux_Projects/Avnet-Digilent-ZedBoard-2015.4
File Edit View Search Terminal Help
[user1@pc196 Avnet-Digilent-ZedBoard-2015.4]$ petalinux-package --image -c rootfs --format initramfs
[INFO ] package rootfs.cpio to /home/user1/Petalinux_Projects/Avnet-Digilent-ZedBoard-2015.4/images/linux
Generating filesystem description file for gen_init_cpio...
Creating /init symlink if required...ln: failed to create symbolic link '/home/user1/Petalinux_Projects/Avnet-Digilent-ZedBoard-2015.4/build/linux/rootfs/targetroot/init': File exists
Done.
directories...files...symlinks...device nodes...done.
Generating CPIO archive /home/user1/Petalinux_Projects/Avnet-Digilent-ZedBoard-2015.4/images/linux/rootfs.cpio ...done.
Image Name:
Created:    Fri Dec 16 20:28:05 2016
Image Type: ARM Linux RAMDisk Image (gzip compressed)
Data Size: 3438060 Bytes = 3357.48 kB = 3.28 MB
Load Address: 00000000
Entry Point: 00000000
[user1@pc196 Avnet-Digilent-ZedBoard-2015.4]$
```

4.1. Format the SD card using GParted

Warning!!! *Gnome Partition Editor can also format the partitions of the HDD of the workstation. Consult your instructor for this part.*



4.2. Transfer Boot Images and filesystem to the formatted SD card.

4.2.1. Navigate to {ptlnx-root-dir}/images/linux/ and copy the

- i) BOOT.BIN
- ii) Image.ub

Files to the BOOT partition of the SD card

4.2.2. While in the images/linux/ directory , right click and open a terminal

4.2.3. Log in as Superuser. (required to access the ext4 FileSystem of the rootfs partition of the SD card)

4.2.4. Type the following commands

```
cp rootfs.cpio /run/media/<username>/rootfs/
```

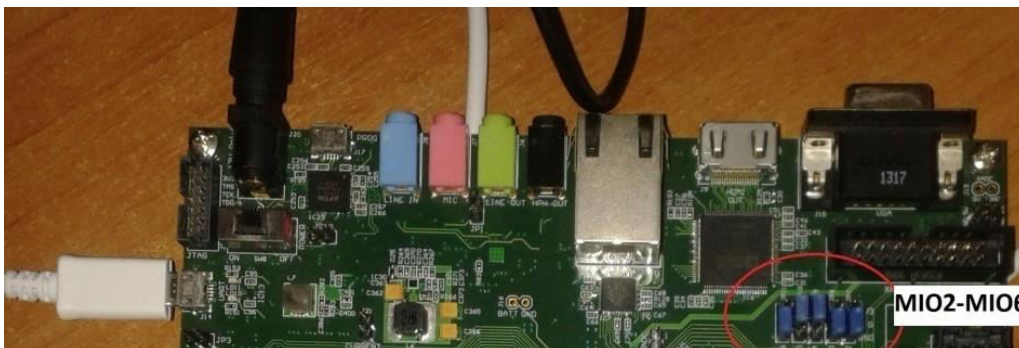
```
cd /run/media/<username>/rootfs/
```

```
cpio -idv < rootfs.cpio
```

4.3. Insert SD card and boot Petalinux.

4.3.1. Insert the SD card on the SD card slot of the Zedboard.

4.3.2. Ensure MIO2-6 pins are in the following configuration to allow for SD Booting



4.3.3. Power on the Zedboard.

4.3.4. Ensure the Blue “Done”LED has been lit.

4.3.5. Connect the USB-UART Cable from the Zedboard to the Development Host PC.

4.3.6. Open a new terminal and type the following command

```
minicom -d /dev/ttyACM0
```

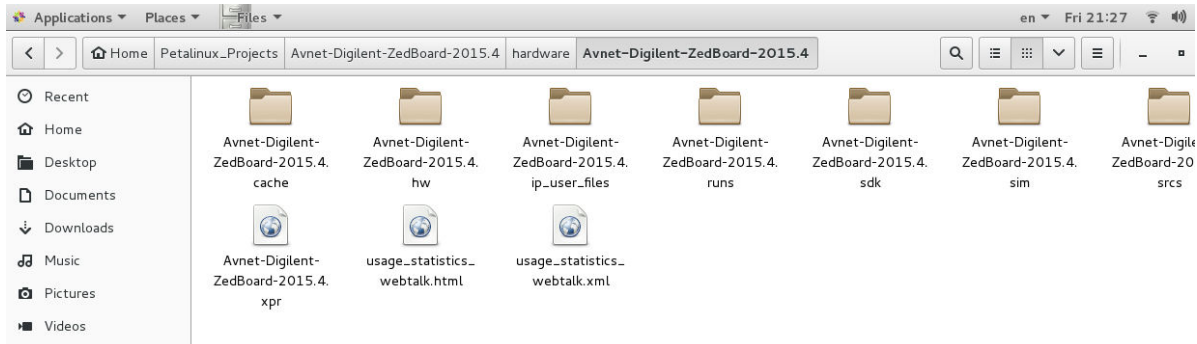
4.3.7. Press enter a few times to show the Petalinux U-Boot command interface.
Type the following command to boot Linux.

```
boot
```

5.1. Open Vivado Hardware project of Petalinux Project

5.1.1. Launch Vivado

5.1.2. Select the Petalinux Project hardware/Avnet-Digilent-Zedboard-2015.4/Avnet-Digilent-Zedboard-2015.4.xpr Vivado project and open it.



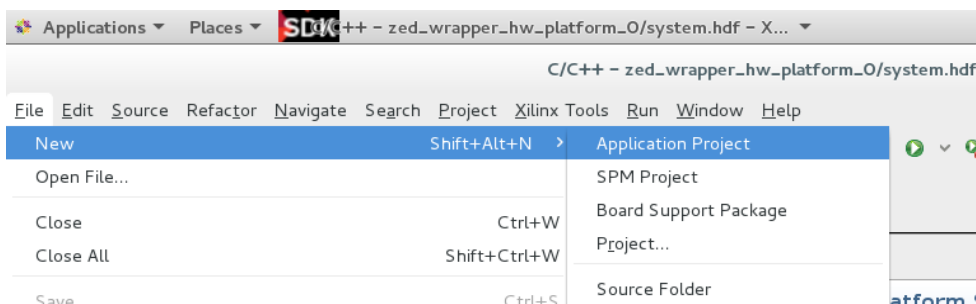
5.1.3. Select the Generate Bitstream option on the Flow Navigator to create the hardware platform.

5.1.4. After the Bitstream has been generated, export the hardware platform to SDK. Select File-> Export -> Export Hardware. Select the “Include Bitstream” option and press yes.

5.1.5. Select File -> Launch SDK

5.2. Create custom Linux application for exported Hardware

5.2.1. After launching XSDK, select File->New->Application project



5.2.2. Name the new application “lab7”. Set the OS Platform to “linux”. Press next.

5.2.3. Select the Linux Empty Application Template. Press Finish.

5.2.4. Right click on the myapp/src/ folder and select Import.

5.2.5. Import the lab7.c file (download from e-class).

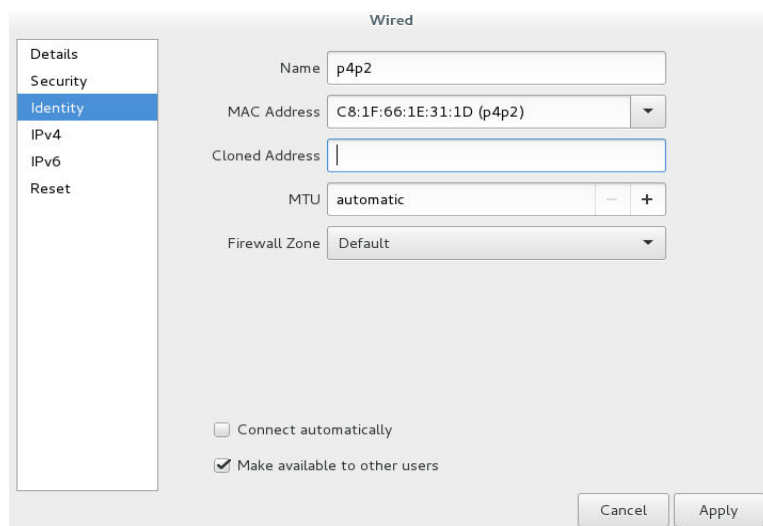
5.2.6. Ensure the .elf file has been compiled and placed in the Debug folder of your application.

6.1. Configure Host (PC) for Wired Ethernet Connection with Zedboard

6.1.1. Open the Network Settings on the Workstation.

6.1.2. Create a Wired profile with the following settings.

- i) Identity -> Mac Address Select the p4p2 device from the drop Down menu



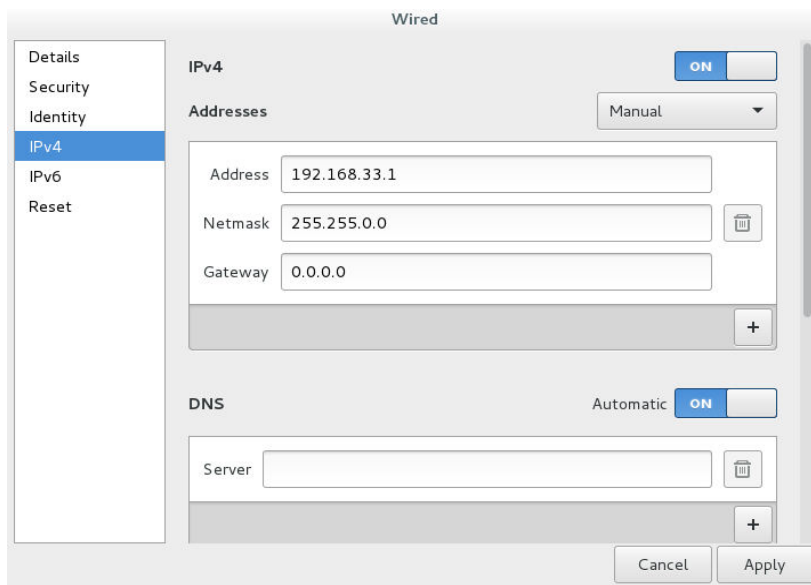
- ii) IPv4 ->

Addresses set to Manual

Address = 192.168.33.1

Netmask = 255.255.0.0

Gateway = 0.0.0.0



6.2. Configure Zedboard for Wired Ethernet Connection

6.2.1. Run the following command on the terminal connected to the Zedboard via Minicom.

```
ifconfig eth0 192.168.33.3
```

```
user1@pc196:~  
File Edit View Search Terminal Help  
root@Avnet-Digilent-ZedBoard-2015_4:~# ifconfig eth0 192.168.33.3
```

6.2.2. On your PC open the Places tab and select the “Connect to Server” option.

6.2.3. Type the following command to open the file explorer for the Zedboard in the /bin directory (many standard commands are stored there, i.e. the echo, cat, ls commands).

```
sftp://192.168.33.2/bin
```

6.2.4. Transfer the compiled lab7.elf file from the SDK application to the Zedboard’s /bin/ directory.

6.2.5. Next, type the command

```
lab7.elf
```

on the terminal connected to the Zedboard via Minicom.