

# Vežba 12 : Podizanje Linux operativnog sistema na Zybo razvojnoj ploči (VGA DMA kontroler)

## 1. Uvod

U ovom uputstvu će biti prikazan proces podizanja *Linux* operativnog sistema na *Zybo* razvojnoj ploči. Koristiće se besplatno razvojno okruženje **PetaLinux** koje obezbeđuje firma *Xilinx* za potrebe podizanja Linux-a na njihovim razvojnim pločama. Iako će ovde biti prikazan postupak za *Zybo Z7* ploču, iste komande se mogu primeniti za ostatak asortimana *Xilinx* proizvoda. Petalinux SDK je moćan alat koji sadrži sve programe koji su potrebni za podizanje, razvijanje, debugovanje i testiranje *Linux* embeded platforme. U ovom dokumentu se neće opisivati alat i njegove funkcionalnosti već će samo biti naveden niz koraka koji je potrebno izvršiti kako bi dobili funkcionalnu *Zybo* ploču za razvijanje drajvera.

Petalinux će generisati četiri glavne komponente koje su neophodne za rad Linux operativnog sistema:

- Linux kernel (*image.ub*)
- Stablo uređaja (*system.dtb*)
- Bootloader (*BOOT.BIN*)
- Root fajl-sistem (*rootfs.tar.gz*)

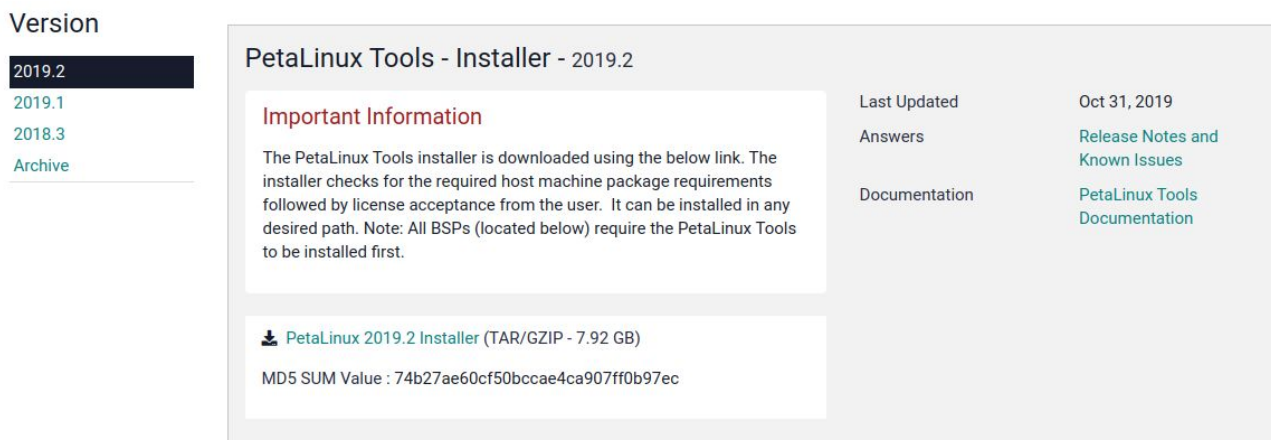
Ovi fajlovi će u pravilnom formatu biti prebačeni na SD karticu sa koje će se *Linux* OS podignuti svaki put kada se ploča upali.

*Petalinux* nije jedini način za podizanje *Linux*-a na *Xilinx* razvojnim pločama. Moguće je ručno kompajlirati kernel za *Zynq 7000 SoC*, zatim pomoću *Xilinx SDK* alata generisati izvorne fajlove za stablo uređaja te ih kompajlirati pomoću *decive-tree compiler (dtc)* alata. Takođe je moguće pomoću *Xilinx SDK* alata napraviti *Bootloader* koji će isprogramirati ploču i podignuti operativni sistem. Pozitivna stvar ovog načina je što se instalira minimalan broj dodatnih softverskih alata, ali sam postupak pravljenja stabla uređaja umje da bude naporan i da oduzme previše vremena pri čestim prepravkama hardvera koji se spušta na *FPGA* ploču.

## 2. Instaliranje PetaLinux alata

Sve aktivne verzije Petalinux alata se mogu videti na sledećoj adresi:

<https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html>



Version

- 2019.2
- 2019.1
- 2018.3
- Archive

PetaLinux Tools - Installer - 2019.2

**Important Information**

The PetaLinux Tools installer is downloaded using the below link. The installer checks for the required host machine package requirements followed by license acceptance from the user. It can be installed in any desired path. Note: All BSPs (located below) require the PetaLinux Tools to be installed first.

⬇️ PetaLinux 2019.2 Installer (TAR/GZIP - 7.92 GB)

MD5 SUM Value : 74b27ae60cf50bccae4ca907ff0b97ec

Last Updated: Oct 31, 2019

Answers: [Release Notes and Known Issues](#)

Documentation: [PetaLinux Tools Documentation](#)

U trenutku pisanja ovog uputstva, poslednja verzija je 2019.2. Potrebno je skinuti fajl “PetaLinux 2019.2 Installer”. Dok se skida instalacioni fajl, možemo preuzeti sva prpratna dokumenta koja se dobijaju uz PetaLinux. Na desnoj strani prozora se nalaze dva linka pod oznakama: “Answers” i “Documentation”.

Klikom na “PetaLinux Tools Documentation” se otvara novi prozor sa kojega se može skinuti pdf fajl koji predstavlja detaljno uputstvo za rukovanje PetaLinux-om. [UG1144 - PetaLinux Tools Documentation: Reference Guide \(v2019.2\)](#) Ovaj fajl se može pogledati ukoliko čitaoca zanimaju funkcionalnosti PetaLinux-a kao i detaljnije uputstvo za rukovanje njime.

Klikom na “Release Notes and Known Issues” se otvara novi prozor, na čijem dnu pod naslovom “Attachments” se može naći link za .xlsx tabelu [2019.2\\_PetaLinux\\_Package\\_List.xlsx](#), u kojoj se nalaze nazivi svih programa koje je neophodno instalirati pre nego što se instalira PetaLinux. U toj tabeli nas zanima rubrika “Ubuntu Desktop/Server (64-bit)” Svaki od navedenih programa je potrebno instalirati pomoću ugrađenog *apt-get* alata. Dovoljno je otvoriti novi terminal (Ctrl+Alt+T) i ukucati sledeću komandu:

```
korisnik ~ $ sudo apt-get install gawk python build-essential gcc git make net-tools libncurses5-dev  
ftpd zlib1g-dev libssl-dev flex bison libselinux1 gnupg wget diffstat chrpath socat xterm autoconf  
libtool tar unzip texinfo zlib1g-dev gcc-multilib zlib1g:i386 screen pax gzip
```

PetaLinux je potrebno instalirati kao običan korisnik (ne kao administrator). U uputstvu se preporučuje da instalacioni direktorijum bude *home* direktorijum “/home/korisnik”. Moguće je instalirati i na privilegovanim putanjama, ali u tom

slučaju je potrebno dati dopuštenje korisniku da pristupi toj putanji (sudo chmod 777 /tools).

Kada je fajl skinut potrebno je preći u Downloads direktorijum, dodati dopuštenje za pokretanje porograma, te pokrenuti Petalinux instaler. Programu se prosleđuje jedan parametar, što je putanja gde se želi instalirati Petalinux (označeno ljubičastom bojom).

```
korisnik ~ $ cd Downloads
korisnik ~/Downloads $ chmod +x petalinux-v2019.2-final-installer.run
korisnik ~/Downloads $ ./petalinux-v2019.2-final-installer.run \ /tools/PetaLinux/2019.2
```

Nakon pokretanja instalera, dobija se sledeći ispis. Program će tražiti da se prihvate licence pritiskom na taster **y** i **enter**. Kada se izlista Licenca može se čitati pomoću tastera **PgUp** i **PgDw**, te se može izaći pritiskom na taster **q**. Kada se prihvate licence počće instalacija Petalinux-a.

```
INFO: Checking installation environment requirements...
INFO: Checking free disk space
INFO: Checking installed tools
INFO: Checking installed development libraries
INFO: Checking network and other services
WARNING: No tftp server found - please refer to "PetaLinux SDK Installation Guide" for its impact
and solution
INFO: Checking installer checksum...
INFO: Extracting PetaLinux installer...

LICENSE AGREEMENTS

PetaLinux SDK contains software from a number of sources. Please review
the following licenses and indicate your acceptance of each to continue.

You do not have to accept the licenses, however if you do not then you may
not use PetaLinux SDK.

Use PgUp/PgDn to navigate the license viewer, and press 'q' to close

Press Enter to display the license agreements
Do you accept Xilinx End User License Agreement? [y/N] > y
Do you accept Webtalk Terms and Conditions? [y/N] > y
Do you accept Third Party End User License Agreement? [y/N] > y
INFO: Installing PetaLinux...
```

Ukoliko dođe do error-a prilikom instalacije, proveriti dopuštenja direktorijuma u kojima se instalira PetaLinux ili instalirati program u home direktorijumu (npr: /home/korisnik/PetaLinux/2019.2). Za detaljniji postupak instalacije, posavetovati se sa " *Reference Guide*" dokumentacijom u kojoj je prvo poglavlje posvećeno ovom problemu.

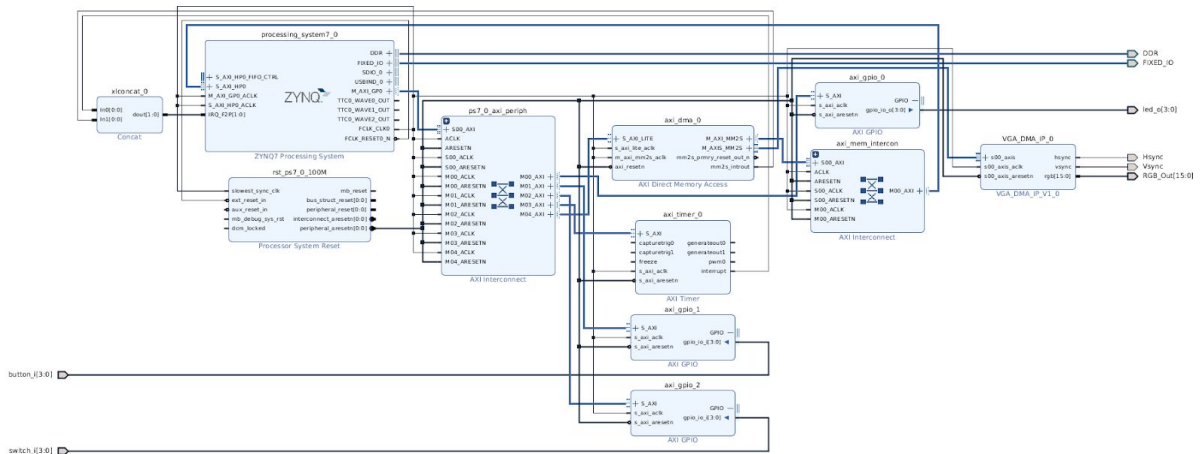
Ukoliko je sve prošlo kako treba, rezultat instalacije izgleda kao na sledećoj slici:

```
INFO: Checking PetaLinux installer integrity ...
INFO: Installing PetaLinux SDK to "/tools/PetaLinux/2019.2/"
INFO: Installing aarch64 Yocto SDK to
"/tools/PetaLinux/2019.2/./components/yocto/source/aarch64"...
INFO: Installing arm Yocto SDK to "/tools/PetaLinux/2019.2/./components/yocto/source/arm"...
INFO: Installing microblaze_full Yocto SDK to
"/tools/PetaLinux/2019.2/./components/yocto/source/microblaze_full"...
INFO: Installing microblaze_lite Yocto SDK to
"/tools/PetaLinux/2019.2/./components/yocto/source/microblaze_lite"...
INFO: PetaLinux SDK has been installed to /tools/PetaLinux/2019.2/.
```

### 3. Generisanje potrebnih fajlova pomoću PetaLinux-a

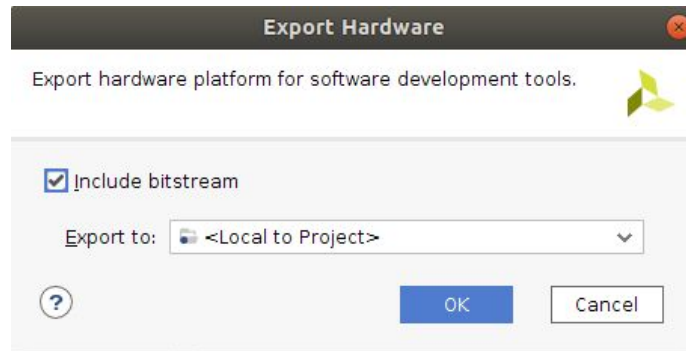
#### 3.1. Eksportovanje hardvera iz Vivado projekta

Petalinux mora da ima sve informacije o hardveru za koji generiše fajlove, te je potrebno imati kompletiran blok dizajn u Vivado IP integratoru. U ovom uputstvu će se napraviti PetaLinux projekat za sistem sa: VGA (DMA) kontrolerom, AXI Brojačem, LED, tasterima i prekidačima.



Napomena: Proveriti da li je ciljana ploča dobro podešena u projektu. U ovom uputstvu je projekat napravljen za ploču Zybo Z7, ukoliko je ploča Zybo Z7-10 promeniti polje “project device” u meniju “Project Manager -> Settings”.

Kada se napravi blok dizajn i constraint fajl, potrebno je odraditi implementaciju i generisanje bitstream fajla. Informacije o sistemu će biti prosleđene PetaLinux-u pomoću “hardware description” fajla (.hdf) ili “xilinx shell architecture” fajla (\*.xsa). Ovaj fajl je neophodno generisati u Vivadu klikom na meni: *File -> Export -> Export Hardware*, čime se otvara prozor sličan prikazanom na sledećoj slici. U prozoru je potrebno kliknuti na opciju *Include bitstream*, jer će se pomoću ovoga fajla programirati FPGA svaki put kada se ploča upali. Polje “Export to” je moguće postaviti na proizvoljnu vrednost, samo je bitno upamtiti lokaciju. Ukoliko se ostavi na opciji “Local to Project”, fajl će biti generisan na putanji projekta u direktorijumu “ime\_projekta.sdk”.



### 3.2. Kreiranje i rad sa PetaLinux projektom

PetaLinux nema grafičko okruženje kao ostatak Xilinx alata, već se komande pokreću direktno u terminalu. Kako bi *bash* terminal prepoznao PetaLinux komande, potrebno je pokrenuti *source* komandu nad fajlom *settings.sh* koji se nalazi u instalacionom direktorijumu.

```
korisnik ~ $ source /tools/PetaLinux/2019.1/settings.sh
```

Ovo je potrebno uraditi u svakom terminalu u kojem se žele pokretati PetaLinux komande.

Kreiranje projekta se vrši pomoću sledeće naredbe:

```
petalinux-create --type project --template <platforma> --name <ime_projekta>
```

Projekat se kreira u direktorijumu u kojem se korisnik trenutno nalazi (u našem slučaju *home* direktorijum, ~). Platforma će za Zybo Z7 ploču biti **zynq** dok se ime projekta može proizvoljno dodeliti. Dakle, prvo pravimo projekat a zatim prelazimo u njega:

```
korisnik ~ $ petalinux-create --type project --template zynq --name VGA  
korisnik ~ $ cd VGA  
korisnik ~/VGA $
```

Sledeće što je potrebno uraditi je konfigurisati projekat na osnovu opisa hardvera koji smo generisali iz Vivado alata. Potrebno je pokrenuti komandu:

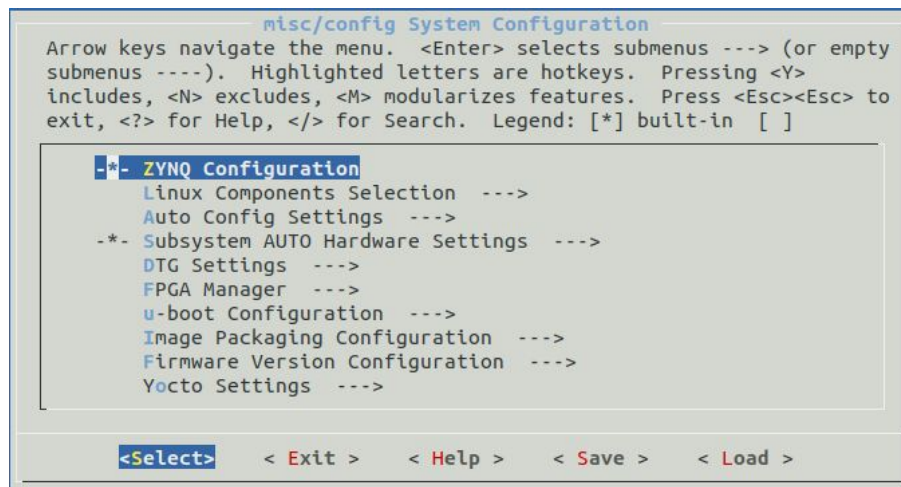
```
petalinux-config --get-hw-description=<Putanja_do_hdf_direktorijuma>
```

Ovoj komandi je potrebno preko parametra *--get-hw-description* proslediti putanju u kojoj se nalazi hdf fajl koji je generisan iz Vivada. U našem slučaju to je:

```
korisnik ~/VGA $ petalinux-config  
--get-hw-description=~/VGA_DMA_Controller/VGA_DMA_Project/Top/result/VGA_DMA_controller.sdk
```

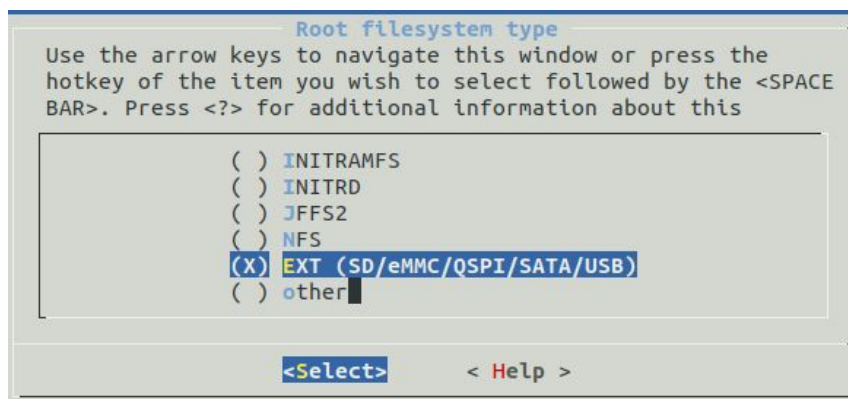
**Napomena:** Od verzije Vivado alata 2019.2 pa na dalje, \*.hdf fajlovi su zamenjeni \*.xsa fajlovima. Te kada se iz Vivada eksportuje hardver dobiće se .xsa fajl. Petalinux podržava obe vrste fajlova uz komandu “petalinux-config --get-hw-description” te je kao u primeru za \*.hdf fajl, samo potrebno proslediti direktorijum u kojem se nalazi \*.xsa fajl.

Ukoliko je komanda dobro napisana, otvara se konfiguracioni prozor kao na sledećoj slici:



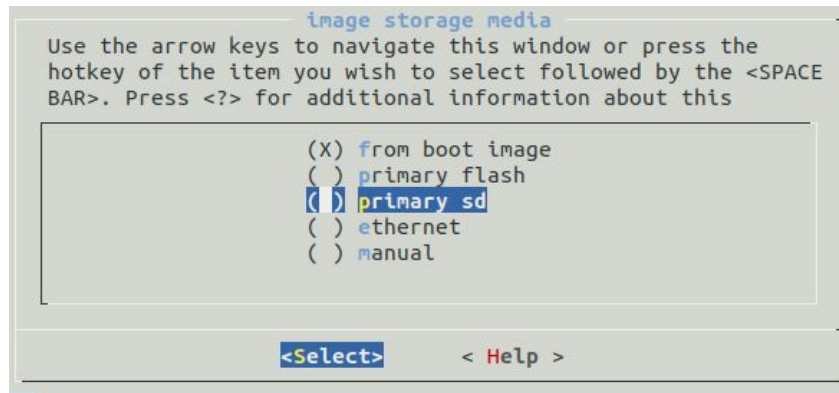
Moguće je prolaziti kroz meni pomoću strelica, opcija se bira strelicama gore-dole, dok akcija strelicama levo-desno, akcija se izvršava pritiskom tastera enter. Od akcija postoji: select, exit, help, save i load. Pomoću akcije “select” je potrebno ući u podmeni: “*image\_packaging\_configuration -> root\_filesystem type*”

Otvara se novi prozor gde je potrebno selektovati opciju EXT kao na sledećoj slici:



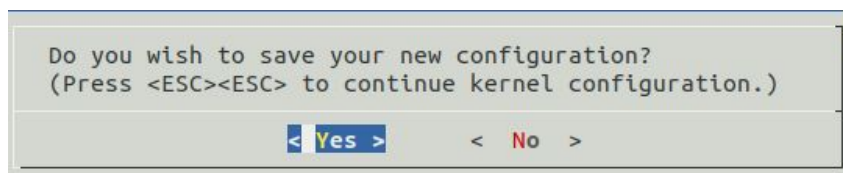
Ovo će nam omogućiti da se Linux učitava sa SD kartice.

Zatim se potrebno pomoću akcije “exit” vratiti nazad do početnog menija, te pomoću akcije “select” izabrati “*Subsystem AUTO Hardware Settings -> Advanced bootable images storage Settings -> dtb image settings -> image storage media*” te izabrati opciju “primary sd” kao na sledećoj slici:



Ova opcija će omogućiti da se radi sa stablom uređaja kao odvojenim fajlom, nezavisnim od kompajliranog kernela.

Sada se može izabrati akcija “exit” i vraćati nazad dok se ne izađe iz glavnog menija. Pri izlasku se otvara prozor koji nas pita da li želimo da snimi konfiguraciju. Izabrati opciju “Yes”.



Sada smo nazad u terminalu, te možemo nastaviti da konfiguriramo projekat. Potrebno je otvoriti fajl *user\_rootfsconfig* koji se nalazi na putanji *~/VGA/project\_spec/meta\_user/conf*, te ubaciti liniju *CONFIG\_kernel-devsrc*. Ova opcija će generisati kernel header fajlove u root fajl sistemu kako bi bilo omogućeno kompajliranje modula.

```
korisnik ~/VGA $ vim project-spec/meta-user/conf/user-rootfsconfig
```

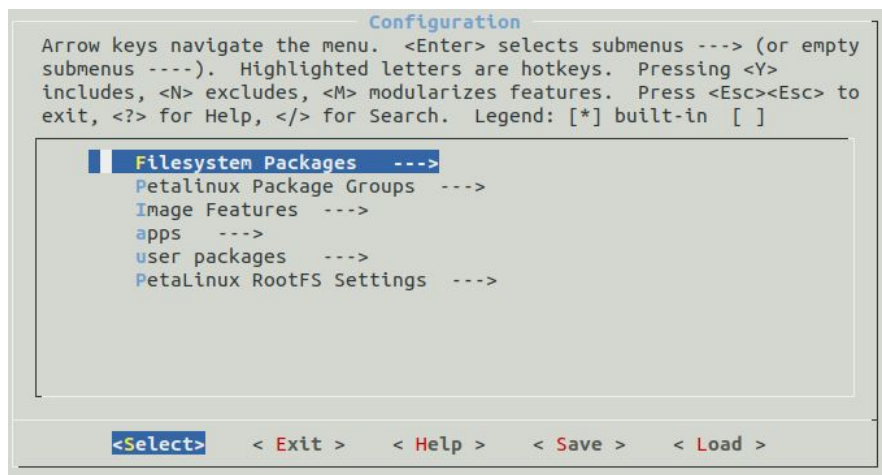
Otvoriti fajl pomoću bilo kojeg tekst editora, dodati liniju kao na slici i snimiti fajl:



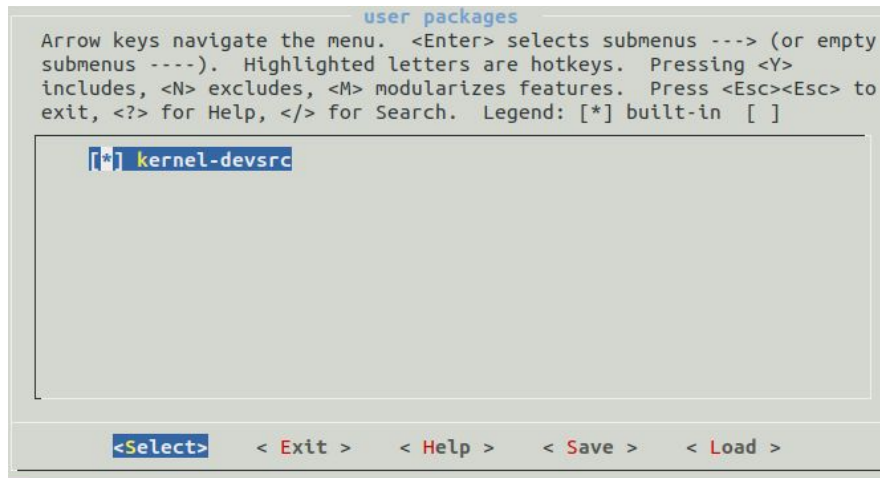
```
user-rootfsconfig (~/VGA/project-spec/meta-user/conf) - VIM
File Edit View Search Terminal Help
p/m/c/user-rootfsconfig buffers
1 #Note: Mention Each package in individual line
2 #These packages will get added into rootfs menu entry
3
4 CONFIG_gpio-demo
5 CONFIG_peekpoke
6 CONFIG_kernel-devsrc
NORMAL <t-spec/meta-user/conf/user-rootfsconfig con... 100% 6/6 ln : 20
"project-spec/meta-user/conf/user-rootfsconfig" 6L, 156C written
```

Zatim je potrebno pokrenuti komandu čime se otvara novi prozor:

```
korisnik ~/VGA $ petalinux-config -c rootfs
```



Selektovati opciju “user packages”, a zatim omogućiti opciju kernel-devsrc klikom na taster “y” kao na sledećoj slici.



Izaći iz ovog menija pomoću akcije exit, te na isti način i iz glavnog menija, pri upitu da li želimo da snimimo konfiguraciju kliknuti “yes”.

Otvoriti fajl *platform-top.h* na putanji *project-spec/meta-user/recipes-bsp/u-boot/files* te na kraj fajla dodati sledeće linije:

```
#ifdef CONFIG_BOOTCOMMAND
#undef CONFIG_BOOTCOMMAND
#define CONFIG_BOOTCOMMAND "run uenvboot; run cp_kernel2ram && run
cp_dtb2ram && bootm ${netstart} - ${dtbnetstart}"
#endif
```

```
korisnik ~/VGA $ vim project-spec/meta-user/recipes-bsp/u-boot/files/platform-top.h
```

```
File Edit View Search Terminal Help
p/m/r/u/f/platform-top.h+ buffers
21 #define CONFIG_BAUDRATE 115200
22 #ifdef CONFIG_DEBUG_UART
23 #undef CONFIG_DEBUG_UART
24 #endif
25 #endif
26
27 /*Dependencies for ENV to be stored in EEPROM. Ensure environment fits in eeprom size*/
28 #ifdef CONFIG_ENV_IS_IN_EEPROM
29 #define CONFIG_SYS_I2C_EEPROM_ADDR_LEN 1
30 #define CONFIG_SYS_I2C_EEPROM_ADDR 0x54
31 #define CONFIG_SYS_EEPROM_PAGE_WRITE_BITS 4
32 #define CONFIG_SYS_EEPROM_PAGE_WRITE_DELAY_MS 5
33 #define CONFIG_SYS_EEPROM_SIZE 1024 /* Bytes */
34 #define CONFIG_SYS_I2C_MUX_ADDR 0x74
35 #define CONFIG_SYS_I2C_MUX_EEPROM_SEL 0x4
36 #endif
37
38 #ifdef CONFIG_BOOTCOMMAND
39 #undef CONFIG_BOOTCOMMAND
40 #define CONFIG_BOOTCOMMAND "run uenvboot; run cp_kernel2ram && run cp_dtb2ram && bootm ${netstart} - ${dtbnetstart}"
41 #endif
42
NORMAL project-spec/meta-user/recipes-bsp/u-boot/files/platform-top.h[+] cpp utf-8[unix] 100% 42/42 ln : 1
```

Ovo ispravlja grešku koja se dešava pri generisanju u-boot komandi za Zybo ploču. Snimiti promene i izaći iz fajla.

Sa ovom ispravkom je projekat spreman za pravljenje, ali je potrebno demonstrirati kako da korisnik napravi promene u stablu uređaja koje će biti generisano. Često je neophodno prepraviti automatski generisane izvorne fajlove stabla uređaja. U ovom primeru ćemo prepraviti “compatible” polja za naše uređaje.

Ukoliko pokrenemo komandu:

```
korisnik ~/VGA $ petalinux-config -c device-tree
```

U direktorijumu *components/plnx\_workspace/device-tree/device-tree* će se pojaviti automatski generisani izvorni fajlovi za stablo uređaja. U fajlu *pl.dtsi* se nalazi izvorni kod za programabilnu logiku koji je sličan sledećem kodnom listingu:

```
/ {
    amba_p1: amba_p1 {
        #address-cells = <1>;
        #size-cells = <1>;
        compatible = "simple-bus";

        axi_dma_0: dma@40400000 {
            #dma-cells = <1>;
            clock-names = "s_axi_lite_aclk", "m_axi_mm2s_aclk";
            clocks = <&clkc 15>, <&clkc 15>;
            compatible = "xlnx,axi-dma-7.1", "xlnx,axi-dma-1.00.a";
            interrupt-names = "mm2s_introut";
            interrupts = <0 30 4>;
            reg = <0x40400000 0x10000>;
            dma-channel@40400000 {
                compatible = "xlnx,axi-dma-mm2s-channel";
                dma-channels = <0x1>;
                interrupts = <0 30 4>;};
        };

        axi_gpio_0: gpio@41200000 {
            #gpio-cells = <3>;
            clock-names = "s_axi_aclk";
            clocks = <&clkc 15>;
            compatible = "xlnx,axi-gpio-2.0", "xlnx,xps-gpio-1.00.a";
            gpio-controller ;
            reg = <0x41200000 0x10000>;
        };

        axi_gpio_1: gpio@41210000 {
            #gpio-cells = <3>;
            clock-names = "s_axi_aclk";
            clocks = <&clkc 15>;
            compatible = "xlnx,axi-gpio-2.0", "xlnx,xps-gpio-1.00.a";
            gpio-controller ;
            reg = <0x41210000 0x10000>;
        };

        axi_gpio_2: gpio@41220000 {
            #gpio-cells = <3>;
            clock-names = "s_axi_aclk";
            clocks = <&clkc 15>;
            compatible = "xlnx,axi-gpio-2.0", "xlnx,xps-gpio-1.00.a";
            gpio-controller ;
            reg = <0x41220000 0x10000>;
        };
    };
};
```

...

Možemo primetiti da su compatible polja za tri uređaja potpuno ista, te bi ovo predstavljalo problem pri uparivanju uređaja sa drajverom. Takođe ukoliko ostavimo generička imena za uređaje, moguće je da će se upariti sa već postojećim drajverima za takve uređaje, a mi želimo da sami napišemo drajver za njih. Korisnik PetaLinux projekta ne sme direktno menjati automatski generisane fajlove, ali izmene možemo uneti u fajl `system_user.dtsi` koji se nalazi na putanji:

`project-spec/meta-user/recipes-bsp/device-tree/files`

```
korisnik ~/VGA $ vim project-spec/meta-user/recipes-bsp/device-tree/files/system-user.dtsi
```

Otvorićemo pomenuti fajl, te u istom formatu opisati naše uređaje. Samo ona polja koja se opisuju će biti redefinisana. Snimamo promene, te izlazimo iz fajla.

```
/include/ "system-conf.dtsi"
/ {
    amba_pl: amba_pl {
        axi_dma_0: dma@40400000 {
            compatible = "vga_dma";
        };
        axi_gpio_0: gpio@41200000 {
            compatible = "led_gpio";
        };
        axi_gpio_1: gpio@41210000 {
            compatible = "button_gpio";
        };
        axi_gpio_2: gpio@41220000 {
            compatible = "switch_gpio";
        };
    };
};
```

U ovom trenutku smo napravili sve promene koje smo želeli, te možemo pokrenuti postupak pravljenja (building) PetaLinux projekta. Ovo je korak koji najduže traje, između 5 i 20 minuta. pokrećemo ga sa:

```
korisnik ~/VGA $ petalinux-build -c device-tree
korisnik ~/VGA $ petalinux-build
```

U direktorijumu `~/VGA/images/linux` će se nalaziti svi generisani fajlovi.

Ukoliko je to potrebno (ne sada), moguće je "očistiti" projekat (izbrisati sve automatski generisane fajlove) naredbom:

```
korisnik ~/VGA $ petalinux-build -x mrproper
```

Poslednje što je potrebno kreirati je bootloader koji će isprogramirati ploču kada se ona upali, te pokrenuti u-boot program za podizanje sistema. Ovo se vrši sledećom komandom:

```
korisnik ~/VGA $ petalinux-package --boot --format BIN --fsbl images/linux/zynq_fsbl.elf --fpga
images/linux/system.bit --u-boot images/linux/u-boot.elf
```

## 4. Prebacivanje sistema na SD karticu

### 4.1. Formatiranje SD kartice

Za podizanje sistema je poželjno imati micro SD karticu veličine 4GB ili više. Micro SD karticu je potrebno povezati sa računarom pomoću adaptera ukoliko računar poseduje slot za SD kartice, ili USB čitača SD kartica. Za svrhu formatiranja SD kartice će se u ovom uputstvu koristiti alat gparted koji se može instalirati pomoću naredbe:

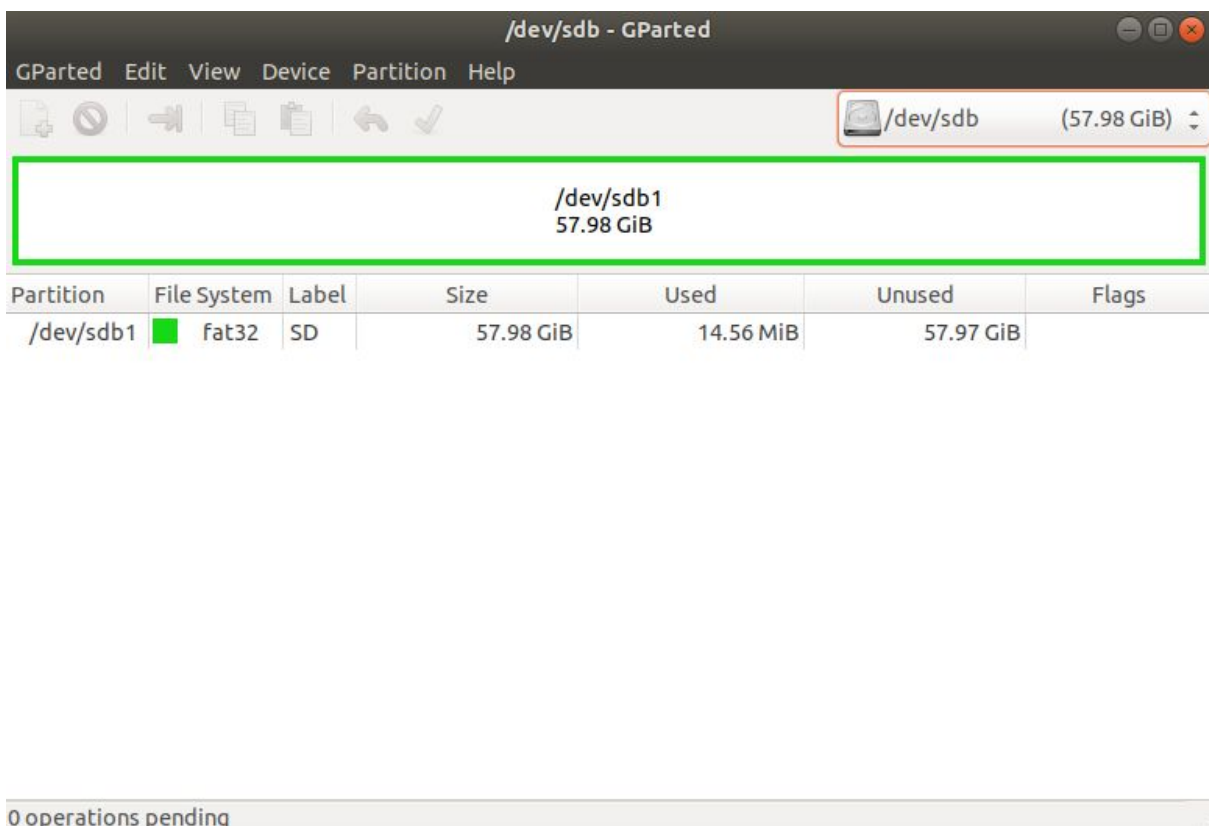
```
korisnik ~/VGA $ sudo apt-get install gparted
```

Alat je moguće pokrenuti komandom gparted. Alat traži administratorsko dopuštenje kako bi mogao menjati sadržaj uređaja.

```
korisnik ~/VGA $ gparted
```

Otvora se prozor kao na sledećoj slici. Bitno je u **gornjem desnom uglu selektovati SD karticu**, koja je u našem slučaju označena sa /dev/sdb.

**Napomena: Veliki oprez obratiti sa kojim uređajem radite! Poslednje što želite je da formatirate hard-disk ili SSD na kojem se nalazi vaš sistem ili podatci!**

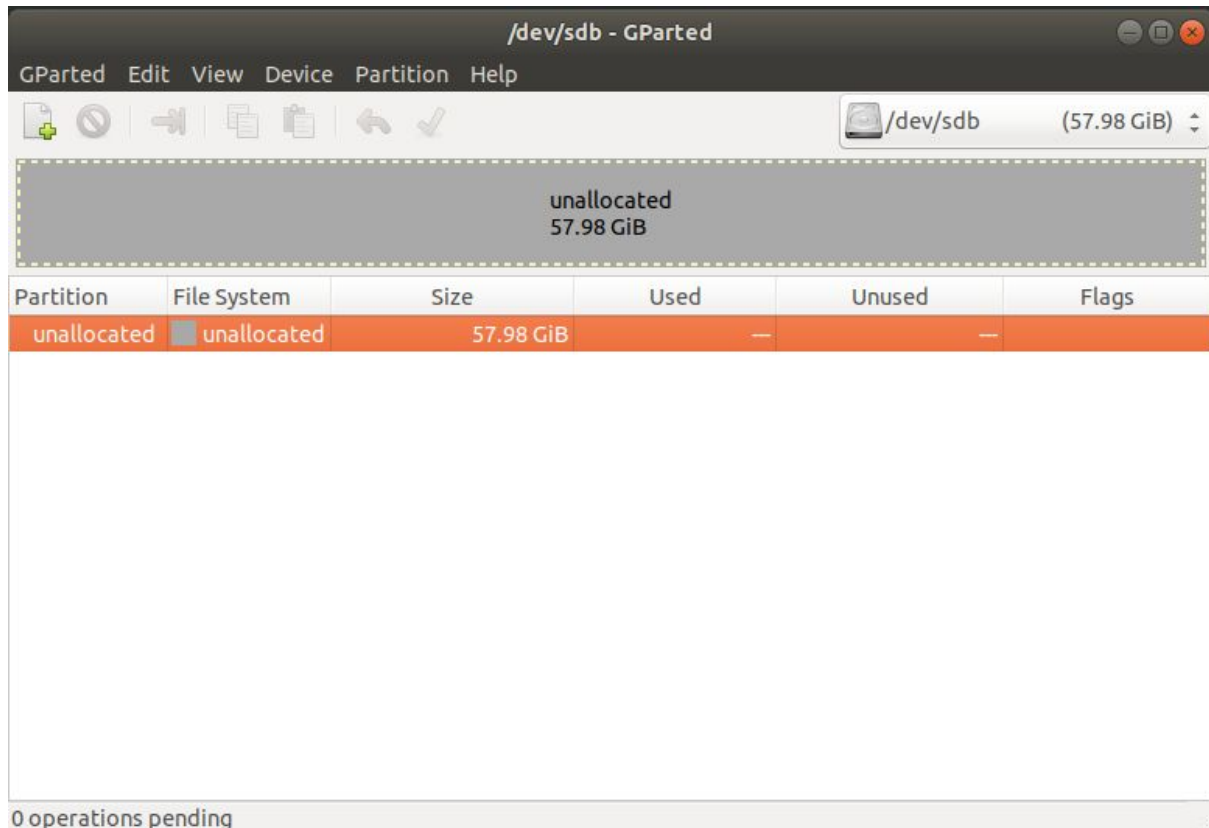


The screenshot shows the GParted application window titled "/dev/sdb - GParted". The window has a menu bar with "GParted", "Edit", "View", "Device", "Partition", and "Help". Below the menu bar is a toolbar with various icons. On the right side of the toolbar, there is a dropdown menu showing "/dev/sdb (57.98 GiB)". The main area of the window displays a large green rectangle representing the selected device, labeled "/dev/sdb1 57.98 GiB". Below this, there is a table with the following data:

Partition	File System	Label	Size	Used	Unused	Flags
/dev/sdb1	fat32	SD	57.98 GiB	14.56 MiB	57.97 GiB	

At the bottom of the window, there is a status bar that reads "0 operations pending".

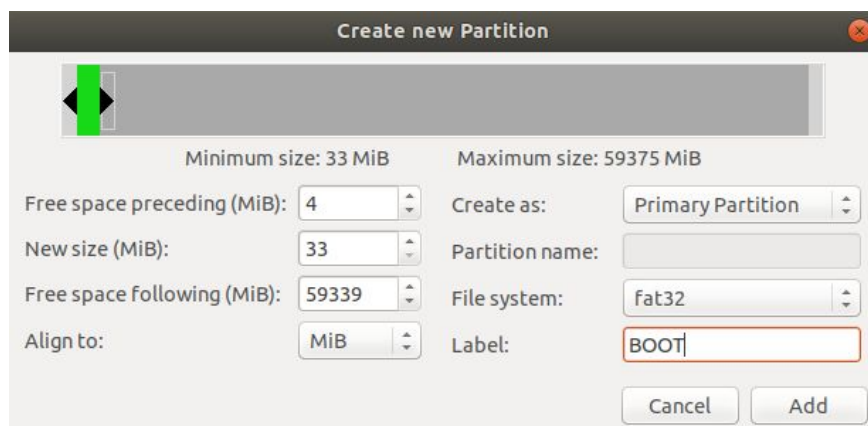
Ukoliko već postoji particija na kartici kao na prethodnj slici, potrebno je desnim klikom izabrati opciju “unmount”, a zatim selektovati particije i pomoću opcije “Delete selected partition” u gornjem levom uglu (crveni znak ⓧ) izbrisati sve particije dok čitav prostor SD kartice nije slobodan za formatiranje kao na sledećoj slici:



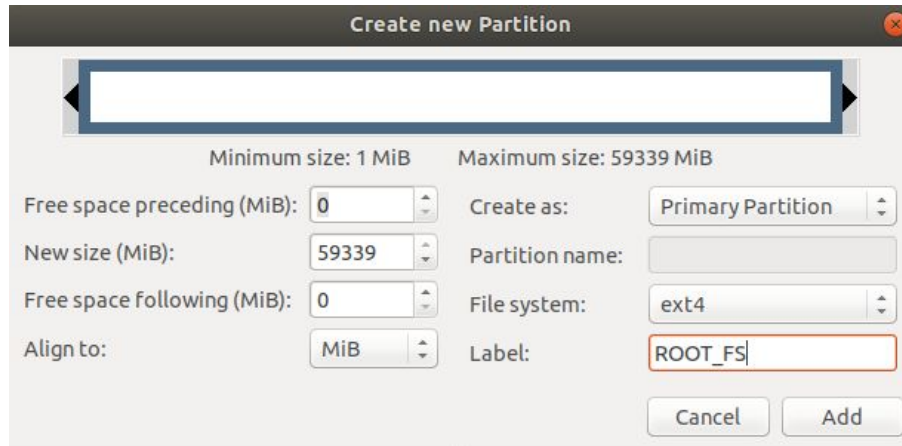
Kada je prostor slobodan za formatiranje, potrebno je napraviti sledeće particije:

- Prva 4 MiB nealocirana
- 32 MiB prostora tipa FAT16 ili FAT32 sa labelom BOOT
- Najmanje 1GiB ext4 prostora sa labelom ROOT\_FS

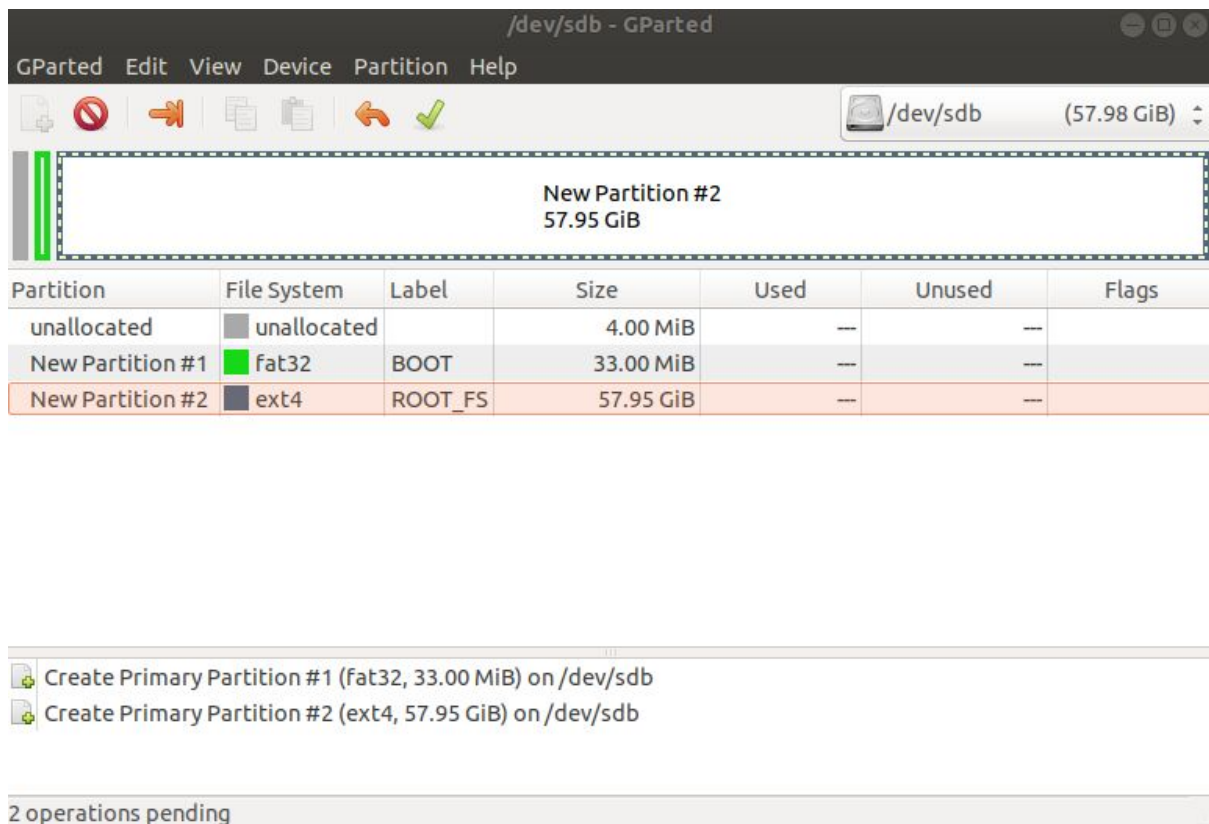
Ovo radimo klikom na opciju “Create New Partition” u gornjem levom uglu (zeleni znak +), te prvo pravimo particiju kao na sledećoj slici:



Zatim ponovo izabiramo opciju "Create New Partition", te prvo pravimo particiju kao na sledećoj slici:



Nakon čega bi izgled prozora trebao izgledati kao na sledećoj slici:



Završavamo formatiranje klikom na opciju Apply all operations (zeleni znak ✓). Sada nam je micro SD kartica formatirana na pravilan način za podizanje Linux sistema. Ovo je potrebno uraditi samo jednom, kada se nabavi kartica. Izvući karticu iz računara te je ponovo ubaciti kako bi računar prepoznao promene.

## 4.2. Prebacivanje boot fajlova na BOOT particiju

Spomenuli smo da se svi generisani fajlovi nalaze na putanji `~/VGA/images/linux`. Na BOOT particiju je potrebno prebaciti tri fajla:

- BOOT.BIN (bootloader)
- system.dtb (stablo uređaja, device-tree)
- image.ub (linux kernel)

Ovo možemo uraditi pomoću komande `cp` (copy).

```
korisnik ~/VGA $ cp -f -t /media/korisnik/BOOT/ images/linux/BOOT.BIN images/linux/system.dtb
images/linux/image.ub
```

Kada su fajlovi kopirani, potrebno je pozvati komandu:

```
korisnik ~/VGA $ umount /media/korisnik/BOOT
```

## 4.3. Prebacivanje fajl sistema na ROOT\_FS particiju

Petalinux je generisao fajl sistem u fajlu `rootfs.tar.gz`. Ukoliko bi ovo otpakovali na ROOT\_FS particiju SD kartice, operativni sistem bi se podigao, ali bi to bila manje popularna distribucija linux-a koju obezbeđuje PetaLinux.

Možemo otpakovati ovaj fajl u posebnom direktorijumu "rootfs" kako bismo imali pristup njegovom sadržaju:

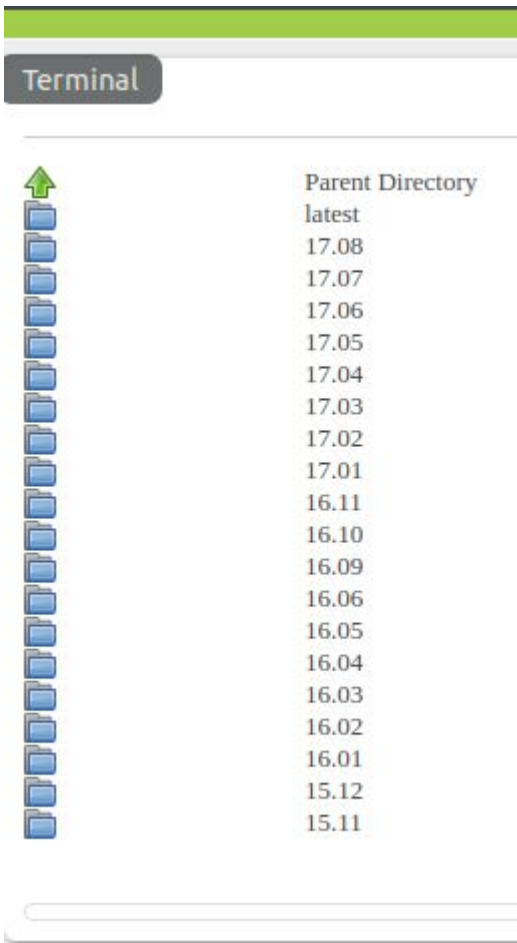
```
korisnik ~/VGA $ mkdir images/linux/rootfs
korisnik ~/VGA $ tar xzf images/linux/rootfs.tar.gz -C images/linux/rootf
```

Budući da mi planiramo praviti module direktno na Zybo ploči, a ne kors-kompajlirati, bolje bi bilo da imamo *debian* distribuciju koja nam omogućuje instaliranje svih potrebnih programa pomoću `apt-get` alata. Možemo otići na link:

<http://releases.linaro.org/debian/images/developer-armhf/>

I skinuti verziju alata 17.2 koja se trenutno koristi na vežbama. Može se skinuti i novija verziju, ali je moguće da će u tom slučaju biti potrebna dodatna podešavanja.





Možemo ući u taj direktorijum, a zatim skinuti upakovani fajl sistem klikom na *linaro-jessie-developer-20161117-32.tar.gz*

	Name	Last modified	Size	License
	Parent Directory			
	linaro-jessie-developer-20161117-32.build-log.txt	13-Jan-2018 10:42	245.1K	open
	linaro-jessie-developer-20161117-32.config.tar.bz2	13-Jan-2018 10:42	7.1K	open
	linaro-jessie-developer-20161117-32.comments	13-Jan-2018 10:42	1.3M	open
	linaro-jessie-developer-20161117-32.md5sums.txt	13-Jan-2018 10:42	404	open
	linaro-jessie-developer-20161117-32.packages	13-Jan-2018 10:42	14.3K	open
	linaro-jessie-developer-20161117-32.sha1sums.txt	13-Jan-2018 10:42	444	open
	linaro-jessie-developer-20161117-32.tar.gz	27-Feb-2018 01:35	335.1M	open

Kada se skinе, ovaj fajl sistem ćemo otpakovati u ROOT\_FS partciju SD kartice.

```
korisnik ~/VGA $ sudo tar xf ~/Downloads/linaro-jesse-developer-20161117-32.tar.gz
--strip-components=1 -C /media/korisnik/ROOT_FS/
```

Takođe ćemo iz PetaLinux-ovog fajl sistema kopirati generisane fajlove za pravljenje modula pomoću komandi:

```
korisnik ~/VGA $ sudo cp -rf images/linux/rootfs/lib/modules/*
/media/korisnik/ROOT_FS/lib/modules/
korisnik ~/VGA $ sudo cp -P images/linux/rootfs/usr/src/kernel
/media/korisnik/ROOT_FS/usr/src/kernel
```

Kada su fajlovi kopirani potrebno je pokrenuti komandu:

```
korisnik ~/VGA $ umount /media/korisnik/ROOT_FS
```

Sačekati da se komanda izvrši, te izvaditi SD karticu iz računara.

## 5. Instaliranje potrebnih programa, i testiranje sistema

### 5.1. Konektovanje preko USB porta

U ovom trenutku je moguće ubaciti SD karticu u Zybo ploču i povezati je sa računarom preko USB kabela. Pomoću preferiranog programa za serijsku komunikaciju (putty, screen) povezati se na odgovarajući USB port.

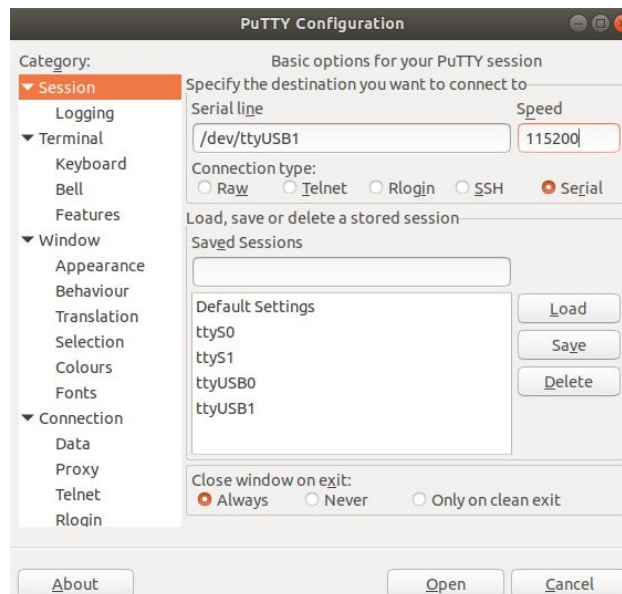
Screen program:

U terminalu ukucati sledeću komandu, nakon čega taj terminal postaje terminal za serijsku komunikaciju.

```
korisnik ~/VGA $ screen /dev/ttyUSB1 115200
```

Putty program:

Podesiti prozor kao na slici te kliknuti "Open"



Zatim sačekati da se sistem podigne...

### 5.2. Konektovanje preko SSH

Kako bi se konektovali preko ssh alata, potrebno je prvo ući preko serijske konekcije. Zatim otvoriti fajl

```
root@linaro-developer:~# vim /etc/ssh/sshd_config
```

i ubaciti liniju:

```
PermitRootLogin yes
```

Snimiti fajl, a zatim restartovati ssh alat pomoću komande

```
root@linaro-developer:~# service sshd restart
```

Takođe je poželjno promeniti lozinku pomoću komande passwd:

```
root@linaro-developer:~# passwd
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@linaro-developer:~#
```

Moguće je i proveriti IP adresu ploče komandom ifconfig:

```
root@linaro-developer:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.14 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::b7d2:f5bc:6692:4f55 prefixlen 64 scopeid 0x20<link>
    ether 00:0a:35:00:1e:53 txqueuelen 1000 (Ethernet)
    RX packets 1691 bytes 85777 (83.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 143 bytes 16696 (16.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 27 base 0xb000
```

Sada se može otvoriti novi prozor i konektovati na ploču pomoću ssh alata, gde je potrebno ukucati IP adresu koju je vratio alat ifconfig u prethodnom koraku.

```
korisnik ~/VGA $ ssh root@192.168.0.14
```

### 5.3. Instalacija potrebnih alata za pravljenje drajvera

Poslednje što je potrebno uraditi je instalirati kernel header fajlove koji su neophodni za pravljenje drajvera (modula) na Zybo ploči. Prvo se ulogovati na ploču preko ssh alata ili serijske komunikacije a zatim kucati sledeće komande:

```
root@linaro-developer:~# apt-get update
root@linaro-developer:~# apt-get install git-core bison flex
```

Zatim preći u direktorijum /lib/modules/4.19.0-xilinx-v2019.2/build i pokrenuti sledeće naredbe:

```
root@linaro-developer:~# cd /lib/modules/4.19.0-xilinx-v2019.2/build/
root@linaro-developer:/lib/modules/4.19.0-xilinx-v2019.2/build# make headers_install
root@linaro-developer:/lib/modules/4.19.0-xilinx-v2019.2/build# make scripts
```

Ovaj korak se izvršava nekoliko minuta, nakon toga je Zybo ploča podešena za pravljenje drajvera (modula).

## 5.4. Testiranje ploče

Možemo to proveriti skidanjem vežbe sa KEEmbedded repozitorijuma, i pokretanjem drajvera.

```
root@linaro-developer:~# git clone https://github.com/KEEmbedded/vezba9
root@linaro-developer:~# cd vezba9/led_driver
root@linaro-developer:~/vezba9/led_driver# make
root@linaro-developer:~/vezba9/led_driver# dmesg -wH &
root@linaro-developer:~/vezba9/led_driver# insmod led_driver.ko
root@linaro-developer:~/vezba9/led_driver# echo 0xf > /dev/led
```

Ukoliko se upale LED na Zybo ploči sve je podešeno kako treba.