

# PetaLinux KICKstart

PetaLinux is brand name used by Xilinx, it is based on Yocto and pretty decent mainstream kernel, what Petalinux adds is the HSI (Hardware Software Interface from Vivado) and special tools for boot image creation.

## Table of Content

- 1 [Table of Content](#)
- 2 [Petalinux Project Creation- Short HOWTO](#)
- 3 [PetaLinux 2018.3](#)
  - [3.1 PetaLinux Installation](#)
  - [3.2 Creating a Project from Vivado Project](#)
  - [3.3 Petalinux Configuration](#)
- 4 [PetaLinux 2018.2](#)
  - [4.1 PetaLinux Installation](#)
  - [4.2 Creating a Project from Vivado Project](#)
  - [4.3 Petalinux Configuration](#)
- 5 [PetaLinux 2017.4](#)
  - [5.1 PetaLinux Installation](#)
  - [5.2 Creating a Project from Vivado Project](#)
  - [5.3 Petalinux Configuration](#)
- 6 [PetaLinux 2017.2](#)
  - [6.1 PetaLinux Installation](#)
  - [6.2 Creating a Project from Vivado Project](#)
  - [6.3 Petalinux Configuration](#)
- 7 [PetaLinux 2017.1](#)
  - [7.1 PetaLinux Installation](#)
  - [7.2 Creating a Project from Vivado Project](#)
  - [7.3 Petalinux Configuration](#)
- 8 [PetaLinux 2016.4](#)
  - [8.1 PetaLinux Installation](#)
  - [8.2 Creating a Project from Vivado Project](#)
  - [8.3 Petalinux Configuration](#)
- 9 [PetaLinux 2016.2](#)
  - [9.1 PetaLinux Installation](#)
  - [9.2 Creating a Project from Vivado Project](#)
  - [9.3 Petalinux Configuration](#)
- 10 [Additional Descriptions](#)
- 11 [References](#)

---

## Petalinux Project Creation- Short HOWTO

Getting Linux working on Zynq is very simple, following steps are required

1. Vivado/SDK/PetaLinux 20xx.x installed (important do not mix versions!)
2. Create Vivado Project, configure PS, Export HDF
3. Ceate new PetaLinux project
4. Import HDF into project
5. petalinux-build
6. copy boot.bin and image.ub to SD Card (only 2 files no more)

This is generic how-to, everything is setup for your by the Vivado->PetaLinux flow. Note, the boot.bin generated by PetaLinux may not always work, in such case it is recommended to make the boot.bin with SDK-GUI or command line tools manually.

There is no need to install anything else, or to fetch anything from any github repos, etc.

# PetaLinux 2018.3

## PetaLinux Installation

- (optional) Create new VM with supported Linux OS.
  - **Attention:** Use English as OS language for your Linux System (Keyboard language can be any language). Other languages may cause errors on PetaLinux build process.
- Download PetaLinux from Xilinx Website: <http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html>
  - Choose a PetaLinux Version, that's corresponding to the installed Vivado and SDK Version.
    - Example: Use Vivado 2018.3 with SDK 2018.3 and PetaLinux 2018.3
- Use UG1144 "PetaLinux Tools Documentation - Reference Guide" that's corresponding with your PetaLinux Version
  1. Change owner of installation directory to non root (see Troubleshoot page)
  2. Check "PetaLinux Tools Installation Requirements" chapter and install missing tool/libraries
  3. Use installation instructions from chapter "PetaLinux Tools Installation Steps"
  4. Deactivate Webtalk: **\$ petalinux-util --webtalk off**
  5. See also [Petalinux Troubleshoot#Petalinux2018.3](#)
- Note:
  - There is no need to install anything else, or to fetch anything from any github repos, etc.
  - It is recommended to test the installation by creating a dummy template project and building it.
    - Download one of the BSP Examples from Xilinx Website (Only to test your installation)
    - Install: **\$ petalinux-create -t project -s <path-to-bsp>**
    - Build: **\$ petalinux-build**

## Creating a Project from Vivado Project



Some reference designs contains a preconfigured PetaLinux project as template. This can be used instead of creating a new project described on step two. Basic Note to TE Petalinux Templates, see: [PetaLinux TE-Template#Template-PetaLinux2018.3](#)

1. PetaLinux Working Environment (see UG1144)
  - PetaLinux Tools Installation is completed.
  - "/bin/sh" is bash
  - Set Working Environment:
    - **\$ source <path-to-installed-PetaLinux>/settings.sh**
2. Create a New Project (see UG1144):
  - **\$ petalinux-create --type project --template <CPU\_TYPE> --name <PROJECT\_NAME>**
    - <CPU\_TYPE>: zynqMP, zynq, microblaze
    - <PROJECT\_NAME>: The name of the project you are building
3. Import Hardware Configuration (see UG1144):
  - Export Hardware Definition File (\*.hdf) from the Vivado Project into the PetaLinux root folder "<plnx-proj-root>":
    - change to PetaLinux root folder:
    - Run: **\$ petalinux-config --get-hw-description**
4. (optional) Configure your PetaLinux:
  - While anywhere in the project folder tree:
  - Run: **\$ petalinux-config**
  - Run: **\$ petalinux-config -c u-boot**
  - Run: **\$ petalinux-config -c kernel**
  - Run: **\$ petalinux-config -c rootfs**
5. Build System Image (see UG1144):
  - While anywhere in the project folder tree:
  - Run: **\$ petalinux-build**
6. Take u-boot.elf, image.ub, bl31.elf (ZynqMP only) from "<plnx-proj-root>/images/linux" and make BOOT.BIN. It is recommended to create the FSBL and PMU Firmware (ZynqMP only) with SDK/HSI tools.

## Petalinux Configuration

Most settings can be changed with menu-config:

- \$ **petalinux-config**
- \$ **petalinux-config -c u-boot**
- \$ **petalinux-config -c kernel**
- \$ **petalinux-config -c rootfs**

Manual changes can be done in the subfolder "<plnx-proj-root>/project-spec/meta-user/"

U-Boot	recipes-bsp/u-boot/files/platform-top.h	overwrite, add UBoot settings
Device Tree	recipes-bsp/device-tree/files/system-user.dtsi recipes-bsp/device-tree/files/zynqmp-qemu-arm.dts	overwrite, add device tree attributes
Kernel	recipes-kernel/linux/linux-xlnx/	changes with " <b>petalinux-config -c kernel</b> " will be add here automatically
Apps	project-spec/meta-user/recipes-apps	add simple new app with " <b>petalinux-create -t apps -n myapp --enable</b> " enable/disable with " <b>petalinux-config -c rootfs</b> "

## PetaLinux 2018.2

### PetaLinux Installation

- (optional) Create new VM with supported Linux OS.
  - **Attention:** Use English as OS language for your Linux System (Keyboard language can be any language). Other languages may cause errors on PetaLinux build process.
- Download PetaLinux from Xilinx Website: <http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html>
  - Choose a PetaLinux Version, that's corresponding to the installed Vivado and SDK Version.
    - Example: Use Vivado 2018.2 with SDK 2018.2 and PetaLinux 2018.2
- Use UG1144 "PetaLinux Tools Documentation - Reference Guide" that's corresponding with your PetaLinux Version
  1. Change owner of installation directory to non root (see Troubleshoot page)
  2. Check "PetaLinux Tools Installation Requirements" chapter and install missing tool/libraries
  3. Use installation instructions from chapter "PetaLinux Tools Installation Steps"
  4. Deactivate Webtalk: \$ **petalinux-util --webtalk off**
  5. See also [Petalinux Troubleshoot - Petalinux2018.2](#)
- Note:
  - There is no need to install anything else, or to fetch anything from any github repos, etc.
  - It is recommended to test the installation by creating a dummy template project and building it.
    - Download one of the BSP Examples from Xilinx Website (Only to test your installation)
    - Install: \$ **petalinux-create -t project -s <path-to-bsp>**
    - Build: \$ **petalinux-build**

### Creating a Project from Vivado Project



Some reference designs contains a preconfigured PetaLinux project as template. This can be used instead of creating a new project described on step two.



Some 18.2 designs include `./init_config.sh` to change directory path (`CONFIG_TMP_DIR_LOCATION` on `<plnx-proj-root>/project-spec/configs/config`). This is not longer necessary with petalinux 2018.2. this files will be removed on later updates

1. PetaLinux Working Environment (see UG1144)
  - PetaLinux Tools Installation is completed.

- "/bin/sh" is bash
  - Set Working Environment:
    - \$ **source <path-to-installed-PetaLinux>/settings.sh**
2. Create a New Project (see UG1144):
    - \$ **petalinux-create --type project --template <CPU\_TYPE> --name <PROJECT\_NAME>**
      - <CPU\_TYPE>: zynqMP, zynq, microblaze
      - <PROJECT\_NAME>: The name of the project you are building
  3. Import Hardware Configuration (see UG1144):
    - Export Hardware Definition File (\*.hdf) from the Vivado Project into the PetaLinux root folder "<plnx-proj-root>":
      - change to PetaLinux root folder:
      - Run: \$ **petalinux-config --get-hw-description**
  4. (optional) Configure your PetaLinux:
    - While anywhere in the project folder tree:
    - Run: \$ **petalinux-config**
    - Run: \$ **petalinux-config -c kernel**
    - Run: \$ **petalinux-config -c rootfs**
  5. Build System Image (see UG1144):
    - While anywhere in the project folder tree:
    - Run: \$ **petalinux-build**
  6. Take u-boot.elf, image.ub, bl31.elf (ZynqMP only) from "<plnx-proj-root>/images/linux" and make BOOT.BIN. It is recommended to create the FSBL and PMU Firmware (ZynqMP only) with SDK/HSI tools.

## Petalinux Configuration

Most settings can be changed with menu-config (\$ **petalinux-config**, **petalinux-config -c kernel**, **petalinux-config -c rootfs**).

Manual changes can be done in the subfolder "<plnx-proj-root>/project-spec/meta-user/"


U-Boot	recipes-bsp/u-boot/files/platform-top.h	overwrite, add UBoot settings
Device Tree	recipes-bsp/device-tree/files/system-user.dtsi recipes-bsp/device-tree/files/zynqmp-qemu-arm.dts	overwrite, add device tree attributes
Kernel	recipes-kernel/linux/linux-xlnx/	changes with " <b>petalinux-config -c kernel</b> " will be add here automatically

## PetaLinux 2017.4

### PetaLinux Installation

- (optional) Create new VM with supported Linux OS.
  - **Attention:** Use English as OS language for your Linux System (Keyboard language can be any language). Other languages may cause errors on PetaLinux build process.
- Download PetaLinux from Xilinx Website: <http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html>
  - Choose a PetaLinux Version, that's corresponding to the installed Vivado and SDK Version.
    - Example: Use Vivado 2017.4 with SDK 2017.4 and PetaLinux 2017.4
- Use UG1144 "PetaLinux Tools Documentation - Reference Guide" that's corresponding with your PetaLinux Version
  1. Change owner of installation directory to non root (see Troubleshoot page)
  2. Check "PetaLinux Tools Installation Requirements" chapter and install missing tool/libraries
  3. Use installation instructions from chapter "PetaLinux Tools Installation Steps"
  4. Deactivate Webtalk: \$ **petalinux-util --webtalk off**
  5. See also [Petalinux Troubleshoot - Petalinux2017.4](#)
- Note:
  - There is no need to install anything else, or to fetch anything from any github repos, etc.
  - It is recommended to test the installation by creating a dummy template project and building it.
    - Download one of the BSP Examples from Xilinx Website (Only to test your installation)
    - Install: \$ **petalinux-create -t project -s <path-to-bsp>**
    - Build: \$ **petalinux-build**

# Creating a Project from Vivado Project

 Some reference designs contains a preconfigured PetaLinux project as template. This can be used instead of creating a new project described on step two.



PetaLinux 17.4 is using one absolute path in "<plnx-proj-root>/project-spec/configs/config". Change path of CONFIG\_TMP\_DIR\_LOCATION variable to your project path "<plnx-proj-root>/build/tmp" manually or use provided "/init\_config.sh" to change path variable automatically. If missing, change execution rights for init\_config.sh with **chmod**

1. PetaLinux Working Environment (see UG1144)
  - PetaLinux Tools Installation is completed.
  - "/bin/sh" is bash
  - Set Working Environment:
    - \$ **source <path-to-installed-PetaLinux>/settings.sh**
2. Create a New Project (see UG1144):
  - \$ **petalinux-create --type project --template <CPU\_TYPE> --name <PROJECT\_NAME>**
    - <CPU\_TYPE>: zynqMP, zynq, microblaze
    - <PROJECT\_NAME>: The name of the project you are building
3. Import Hardware Configuration (see UG1144):
  - Export Hardware Definition File (\*.hdf) from the Vivado Project into the PetaLinux root folder "<plnx-proj-root>":
    - change to PetaLinux root folder:
    - Run: \$ **petalinux-config --get-hw-description**
4. (optional) Configure your PetaLinux:
  - While anywhere in the project folder tree:
    - Run: \$ **petalinux-config**
    - Run: \$ **petalinux-config -c kernel**
    - Run: \$ **petalinux-config -c rootfs**
5. Build System Image (see UG1144):
  - While anywhere in the project folder tree:
    - Run: \$ **petalinux-build**
6. Take u-boot.elf, image.ub, bl31.elf (ZynqMP only) from "<plnx-proj-root>/images/linux" and make BOOT.BIN. It is recommended to create the FSBL and PMU Firmware (ZynqMP only) with SDK/HSI tools.

## Petalinux Configuration

Most settings can be changed with menu-config (\$ **petalinux-config**, **petalinux-config -c kernel**, **petalinux-config -c rootfs**).

Manual changes can be done in the subfolder "<plnx-proj-root>/project-spec/meta-user/"

U-Boot	recipes-bsp/u-boot/files/platform-top.h	overwrite, add UBoot settings
Device Tree	recipes-bsp/device-tree/files/system-user.dtsi recipes-bsp/device-tree/files/zynqmp-qemu-arm.dts	overwrite, add device tree attributes
Kernel	recipes-kernel/linux/linux-xlnx/	changes with " <b>petalinux-config -c kernel</b> " will be add here automatically


## PetaLinux 2017.2


### PetaLinux Installation

- (optional) Create new VM with supported Linux OS.

- **Attention:** Use English as OS language for your Linux System (Keyboard language can be any language). Other languages may cause errors on PetaLinux build process.
- Download PetaLinux from Xilinx Website: <http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html>
  - Choose a PetaLinux Version, that's corresponding to the installed Vivado and SDK Version.
    - Example: Use Vivado 2017.2 with SDK 2017.2 and PetaLinux 2017.2
- Use UG1144 "PetaLinux Tools Documentation - Reference Guide" that's corresponding with your PetaLinux Version
  1. Change owner of installation directory to non root (see Troubleshoot page)
  2. Check "PetaLinux Tools Installation Requirements" chapter and install missing tool/libraries
  3. Use installation instructions from chapter "PetaLinux Tools Installation Steps"
  4. See [Troubleshoot PetaLinux2017.2](#)
- Note:
  - There is no need to install anything else, or to fetch anything from any github repos, etc.
  - It is recommended to test the installation by creating a dummy template project and building it.
    - Download one of the BSP Examples from Xilinx Website (Only to test your installation)
    - Install: `$ petalinux-create -t project -s <path-to-bsp>`
    - Build: `$ petalinux-build`

## Creating a Project from Vivado Project

 Some reference designs contains a preconfigured PetaLinux project as template. This can be used instead of creating a new project described on step two.

 PetaLinux 17.2 is using one absolute path in "`<plnx-proj-root>/project-spec/configs/config`". Change path of `CONFIG_TMP_DIR_LOCATION` variable to your project path "`<plnx-proj-root>/build/tmp`" manually or use provided "`init_config.sh`" to change path variable automatically.

1. PetaLinux Working Environment (see UG1144)
  - PetaLinux Tools Installation is completed.
  - `/bin/sh` is bash
  - Set Working Environment:
    - `$ source <path-to-installed-PetaLinux>/settings.sh`
2. Create a New Project (see UG1144):
  - `$ petalinux-create --type project --template <CPU_TYPE> --name <PROJECT_NAME>`
    - `<CPU_TYPE>`: zynqMP, zynq, microblaze
    - `<PROJECT_NAME>`: The name of the project you are building
3. Import Hardware Configuration (see UG1144):
  - Export Hardware Definition File (`.hdf`) from the Vivado Project into the PetaLinux root folder "`<plnx-proj-root>`":
    - change to PetaLinux root folder:
    - Run: `$ petalinux-config --get-hw-description`
4. (optional) Configure your PetaLinux:
  - While anywhere in the project folder tree:
    - Run: `$ petalinux-config`
    - Run: `$ petalinux-config -c kernel`
    - Run: `$ petalinux-config -c rootfs`
5. Build System Image (see UG1144):
  - While anywhere in the project folder tree:
    - Run: `$ petalinux-build`
6. Take `u-boot.elf`, `image.ub`, `bl31.elf` (ZynqMP only) from "`<plnx-proj-root>/images/linux`" and make `BOOT.BIN`. It is recommended to create the FSBL and PMU Firmware (ZynqMP only) with SDK/HSI tools.

## Petalinux Configuration

Most settings can be changed with `menu-config` (`$ petalinux-config`, `petalinux-config -c kernel`, `petalinux-config -c rootfs`).

Manual changes can be done in the subfolder "`<plnx-proj-root>/project-spec/meta-user/`"

U-Boot	<code>recipes-bsp/u-boot/files/platform-top.h</code>	overwrite, add UBoot settings
--------	--	-------------------------------


Device Tree	recipes-bsp/device-tree/files/system-user.dtsi recipes-bsp/device-tree/files/zynqmp-qemu-arm.dts	overwrite, add device tree attributes
Kernel	recipes-kernel/linux/linux-xlnx/	changes with " <b>petalinux-config -c kernel</b> " will be add here automatically


# PetaLinux 2017.1

## PetaLinux Installation

- (optional) Create new VM with supported Linux OS.
  - Attention:** Use English as OS language for your Linux System (Keyboard language can be any language). Other languages may cause errors on PetaLinux build process.
- Download PetaLinux from Xilinx Website: <http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html>
  - Choose a PetaLinux Version, that's corresponding to the installed Vivado and SDK Version.
    - Example: Use Vivado 2017.1 with SDK 2017.1 and PetaLinux 2017.1
- Use UG1144 "PetaLinux Tools Documentation - Reference Guide" that's corresponding with your PetaLinux Version
  - Check "PetaLinux Tools Installation Requirements" chapter and install missing tool/libraries
  - Use installation instructions from chapter "PetaLinux Tools Installation Steps"
  - See [Troubleshoot PetaLinux2017.1](#)
- Note:
  - There is no need to install anything else, or to fetch anything from any github repos, etc.
  - It is recommended to test the installation by creating a dummy template project and building it.
    - Download one of the BSP Examples from Xilinx Website (Only to test your installation)
    - Install: `$ petalinux-create -t project -s <path-to-bsp>`
    - Build: `$ petalinux-build`

## Creating a Project from Vivado Project

 Some reference designs contains a preconfigured PetaLinux project as template. This can be used instead of creating a new project described on step two.

 PetaLinux 17.1 is using one absolute path in "`<plnx-proj-root>/project-spec/configs/config`" manually. Change path of `CONFIG_TMP_DIR_LOCATION` variable to your project path "`<plnx-proj-root>/build/tmp`" manually.

- PetaLinux Working Environment (see UG1144)
  - PetaLinux Tools Installation is completed.
  - "/bin/sh" is bash
  - Set Working Environment:
    - `$ source <path-to-installed-PetaLinux>/settings.sh`
- Create a New Project (see UG1144):
  - `$ petalinux-create --type project --template <CPU_TYPE> --name <PROJECT_NAME>`
    - `<CPU_TYPE>`: zynqMP, zynq, microblaze
    - `<PROJECT_NAME>`: The name of the project you are building
- Import Hardware Configuration (see UG1144):
  - Export Hardware Definition File (\*.hdf) from the Vivado Project into the PetaLinux root folder "`<plnx-proj-root>`":
    - change to PetaLinux root folder:
    - Run: `$ petalinux-config --get-hw-description`
- (optional) Configure your PetaLinux:
  - While anywhere in the project folder tree:

- Run: \$ **petalinux-config**
  - Run: \$ **petalinux-config -c kernel**
  - Run: \$ **petalinux-config -c rootfs**
5. Build System Image (see UG1144):
    - While anywhere in the project folder tree:
    - Run: \$ **petalinux-build**
  6. Take u-boot.elf, image.ub, bl31.elf (ZynqMP only) and pmufw.elf (ZynqMP only) from "<plnx-proj-root>/images/linux" and make BOOT.BIN. It is recommended to create the FSBL with SDK/HSI tools.

## Petalinux Configuration

Most settings can be changed with menu-config (\$ **petalinux-config**, **petalinux-config -c kernel**, **petalinux-config -c rootfs**).

Manual changes can be done in the subfolder "<plnx-proj-root>/project-spec/meta-user/"


U-Boot	recipes-bsp/u-boot/files/platform-top.h	overwrite, add UBoot settings
Device Tree	recipes-bsp/device-tree/files/system-user.dtsi	overwrite, add device tree attributes
Kernel	recipes-kernel/linux/linux-xlnx/	changes with " <b>petalinux-config -c kernel</b> " will be add here automatically

## PetaLinux 2016.4

### PetaLinux Installation

- (optional) Create new VM with supported Linux OS.
  - **Attention:** Use English as OS language for your Linux System (Keyboard language can be any language). Other languages may cause errors on PetaLinux build process.
- Download PetaLinux from Xilinx Website: <http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html>
  - Choose a PetaLinux Version, that's corresponding to the installed Vivado and SDK Version.
    - Example: Use Vivado 2016.4 with SDK 2016.4 and PetaLinux 2016.4
- Use UG1144 "PetaLinux Tools Documentation - Reference Guide" that's corresponding with your PetaLinux Version
  1. Check "PetaLinux Tools Installation Requirements" chapter and install missing tool/libraries
  2. Use installation instructions from chapter "PetaLinux Tools Installation Steps"
  3. See [Troubleshoot PetaLinux2016.4](#)
- Note:
  - There is no need to install anything else, or to fetch anything from any github repos, etc.
  - It is recommended to test the installation by creating a dummy template project and building it.
    - Download one of the BSP Examples from Xilinx Website (Only to test your installation)
    - Install: \$ **petalinux-create -t project -s <path-to-bsp>**
    - Build: \$ **petalinux-build**

## Creating a Project from Vivado Project

 Some reference designs contains a preconfigured PetaLinux project as template. This can be used instead of creating a new project described on step two.



PetaLinux 16.4 is using one absolute path in "<plnx-proj-root>/project-spec/configs/config". Change path of CONFIG\_TMP\_DIR\_LOCATION variable to your project path "<plnx-proj-root>/build/tmp" manually.



- PetaLinux Working Environment (see UG1144)
  - PetaLinux Tools Installation is completed.
  - "/bin/sh" is bash
  - Set Working Environment:
    - \$ **source <path-to-installed-PetaLinux>/settings.sh**
- Create a New Project (see UG1144):
  - \$ **petalinux-create --type project --template <CPU\_TYPE> --name <PROJECT\_NAME>**
    - <CPU\_TYPE>: zynqMP, zynq, microblaze
    - <PROJECT\_NAME>: The name of the project you are building
- Import Hardware Configuration (see UG1144):
  - Export Hardware Definition File (\*.hdf) from the Vivado Project into the PetaLinux root folder "<plnx-proj-root>":
    - change to PetaLinux root folder:
    - Run: \$ **petalinux-config --get-hw-description**
- (optional) Configure your PetaLinux:
  - While anywhere in the project folder tree:
  - Run: \$ **petalinux-config**
  - Run: \$ **petalinux-config -c kernel**
  - Run: \$ **petalinux-config -c rootfs**
- Build System Image (see UG1144):
  - While anywhere in the project folder tree:
  - Run: \$ **petalinux-build**
- Take u-boot.elf, image.ub and bl31.elf (zynqMP only) from "<plnx-proj-root>/images/linux" and make BOOT.BIN. It is recommended to create the FSBL with SDK/HSI tools.

## Petalinux Configuration

Most settings can be changed with menu-config (\$ **petalinux-config**, **petalinux-config -c kernel**, **petalinux-config -c rootfs**).

Manual changes can be done in the subfolder "<plnx-proj-root>/project-spec/meta-user/"

U-Boot	recipes-bsp/u-boot/files/platform-top.h	overwrite, add UBoot settings
Device Tree	recipes-dt/device-tree/files/system-top.dts	overwrite, add device tree attributes
Kernel	recipes-kernel/linux/linux-xlnx/	changes with " <b>petalinux-config -c kernel</b> " will be add here automatically

## PetaLinux 2016.2

### PetaLinux Installation

- (optional) Create new VM with supported Linux OS.
  - Attention:** Use English as OS language for your Linux System (Keyboard language can be any language). Other languages may cause errors on PetaLinux build process.
- Download PetaLinux from Xilinx Website: <http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html>
  - Choose a PetaLinux Version, that's corresponding to the installed Vivado and SDK Version.
    - Example: Use Vivado 2016.2 with SDK 2016.2 and PetaLinux 2016.2
- Use UG1144 "PetaLinux Tools Documentation - Reference Guide" that's corresponding with your PetaLinux Version
  - Check "PetaLinux Tools Installation Requirements" chapter and install missing tool/libraries
  - Use installation instructions from chapter "PetaLinux Tools Installation Steps"
  - Additional packages for PetaLinux 2016.2 + MicroBlaze projects:
    - \$ **sudo apt-get install libc6-i386 lib32stdc++6 lib32gcc1 lib32ncurses5 lib32z1**
- Note:
  - There is no need to install anything else, or to fetch anything from any github repos, etc.
  - It is recommended to test the installation by creating a dummy template project and building it.
    - Download one of the BSP Examples from Xilinx Website (Only to test your installation)

- Install: \$ **petalinux-create -t project -s <path-to-bsp>**
- Build: \$ **petalinux-build**

## Creating a Project from Vivado Project



Some reference designs contains a preconfigured PetaLinux project as template. This can be used instead of creating a new project described on step two.

1. PetaLinux Working Environment (see UG1144)
  - PetaLinux Tools Installation is completed.
  - "/bin/sh" is bash
  - Set Working Environment:
    - \$ **source <path-to-installed-PetaLinux>/settings.sh**
  - Set cross compiler:
    - \$ **export CROSS\_COMPILE=arm-xilinx-linux-gnueabi-**
    - \$ **export ARCH=arm**
2. Create a New Project (see UG1144):
  - \$ **petalinux-create --type project --template <CPU\_TYPE> --name <PROJECT\_NAME>**
    - <CPU\_TYPE>: zynqMP, zynq, microblaze
    - <PROJECT\_NAME>: The name of the project you are building
3. Import Hardware Configuration (see UG1144):
  - Export Hardware Definition File (\*.hdf) from the Vivado Project into the PetaLinux subfolder "<plnx-proj-root>/hw-description":
    - change to the hardware description folder:
    - Run: \$ **petalinux-config --get-hw-description**
4. (optional) Configure your PetaLinux:
  - While anywhere in the project folder tree:
  - Run: \$ **petalinux-config**
    - It's recommended to deactivate FSBL-Configuration and build FSBL and Boot.bin with SDK/HSI: "linux Components Selection --->" [ ] First Stage Bootloader
  - Run: \$ **petalinux-config -c kernel**
  - Run: \$ **petalinux-config -c rootfs**
5. Build System Image (see UG1144):
  - While anywhere in the project folder tree:
  - Run: \$ **petalinux-build**
  - Build log-file: "<plnx-proj-root>/build/build.log"
6. Take u-boot.elf and image.ub from "<plnx-proj-root>/images/linux" and make BOOT.BIN. It is recommended to create the FSBL with SDK/HSI tools.

## Petalinux Configuration

Most settings can be changed with menu-config (\$ **petalinux-config**).

There are 3 more files that user can edit, they are

File	Description	Location
system-top.dts	Device tree changes are to be applied here	<petalinux-project>/subsystems/linux/configs/device-tree
platform-top.h	U-boot changes are to be applied here	<petalinux-project>/subsystems/linux/configs/u-boot
fsbl_hooks.c	FSBL changes are to be applied here	SDK/HSI FSBL-Template

There should be no reason to modify any other files by editing them (most of them are generated and would be overwritten).

## Additional Descriptions

- [Debian Installation](#)
- [How to install the linux-rt \(Real-Time\) patch](#)
- [PetaLinux TE-Template](#)
- [Petalinux Troubleshoot](#)
- [Reference Designs with PetaLinux](#)

## References

1. [PetaLinux Tools Documentation - Reference Guide \(UG1144\)](#)
2. [PetaLinux Tools Documentation - PetaLinux Command Line Reference \(UG1157\)](#)
3. <https://www.devicetree.org/>
  - a. <https://www.devicetree.org/downloads/devicetree-specification-v0.1-20160524.pdf>
4. <https://github.com/Xilinx/linux-xlnx/tree/master/Documentation/devicetree/bindings>