PetaLinux SDK User Guide

Getting Started Guide

UG977 (v2013.04) April 22, 2013

XILINX®

Revision History

Date	Version	Notes
2009-11-19	1.1	Initial version for SDK 1.1 release
2011-11-26	1.3	Updated for PetaLinux SDK 1.3 release - PowerPC 440 support
2011-04-01	2.1	Updated for PetaLinux SDK 2.1 release - new procedure for rebuilding reference designs based on Xilinx 13.1
2012-08-03	3.1	Updated for PetaLinux SDK 3.1 release
2012-09-03	12.9	Updated for PetaLinux SDK 12.9 release
2012-12-17	2012.12	Updated for PetaLinux SDK 2012.12 release
2013-04-22	2013.04	Updated for PetaLinux SDK 2013.04 release

XILINX

Table of Contents

About this Guide

This document provides basic information on how to start working with the PetaLinux SDK. PetaLinux is the only Embedded Linux System Development Kit specifically targeting FPGA-based System-on-Chip designs. With PetaLinux, you can:

- Synchronise your hardware platform and software platform in one step
- Easily propagate your user application to MicroBlaze or Zynq Embedded Linux systems
- Test your MicroBlaze or Zynq Linux system in a virtual machine environment using QEMU.

The following sections will describe PetaLinux installation, reference design verification with both QEMU and hardware, reference design software recompilation and verification, and reference design hardware rebuild and verification.

Please note: the reader of this document is assumed to have basic Linux knowledge such as how to run Linux commands.

Prerequisites

This getting started document assumes that the following prerequisites have been satisfied:

- Minimum workstation requirements:
  - 2GB RAM (recommended minimum for Xilinx tools)
  - Pentium 4 2GHz CPU clock or equivalent
  - 5 GB free HDD space
  - Recommended OS: CentOS / RHEL 5 (32-bit), or Ubuntu 10.04 (32-bit or 64-bit)
  - Xilinx ISE and EDK Tools 14.4 or 14.5 (required for building and downloading FPGA hardware bitstreams)
- Your workstation has `tftpd` server running.
- There exists a `"/tftpboot"` directory on your workstation and all users have read/write permissions to it.
- A serial communication program such as `minicom` or `kermit` has been installed; the baud rate of the serial communication program has been set to 115200bps.
- If you wish to work on hardware designs, Xilinx tools must be installed. Please refer to Xilinx installation documentation and procedures.

Installation

Please refer to the *PetaLinux SDK Installation Guide (UG976)* to install PetaLinux and PetaLinux BSPs (Board Support Package). A PetaLinux BSP includes at least one hardware platform configuration, software platform configuration, pre-built hardware bitstream and software images for a reference design.

WARNING: *Ensure that at least one PetaLinux BSP has been installed.*

Environment Setup

After PetaLinux has been successfully installed, setup the PetaLinux working environment by running the PetaLinux setup script as follows:

1. Source the set up script.

```
$ cd <path-to-installed-PetaLinux>
$ source settings.sh
```

WARNING:

- *Only run one of these scripts - whichever is appropriate for your terminal shell*
- *You must run the settings script each time you open a new terminal window or shell. The PetaLinux tools will fail otherwise.*
- *You must be within the PetaLinux root directory (e.g. "/home/user/petalinux-v2013.04-final-full") to source the settings file.*

2. Verify that the PetaLinux working environment has been set:

```
$ echo $PETALINUX
/home/user/petalinux
```

Environment variable "$PETALINUX" should point to the path to the installed PetaLinux. Your echo output may be different from this example, it depends on where you installed PetaLinux.

Test a Pre-built PetaLinux Image

So far, you have successfully installed PetaLinux, one or more BSPs and setup the PetaLinux working environment. Now, you can try one of the reference designs shipped with your BSP package. This is achieved with the `petalinux-boot-prebuilt` command, which is used to boot reference designs under software simulation (QEMU) as well as on a hardware board.

Test Pre-Built PetaLinux Image with QEMU

PetaLinux provides QEMU support such that the PetaLinux software image can be tested in a simulated environment, without any hardware.

WARNING: *In order to use the virtual networking features of QEMU you must have root access on the local machine. Please refer to the PetaLinux SDK QEMU System Simulation Guide (UG982) for more details.*

To test the PetaLinux reference design with QEMU, follow these steps:

1. Boot a reference design PetaLinux image with `petalinux-boot-prebuilt` command:

   ```
   $ petalinux-boot-prebuilt -p <reference design name> -q
   ```

   The `-q` option tells `petalinux-boot-prebuilt` to boot QEMU, instead of real hardware via JTAG. e.g, for the `"Xilinx-ZC702-14.5"` reference design:

   ```
   $ petalinux-boot-prebuilt -p Xilinx-ZC702-14.5 -q
   ```

   To see a list of the installed reference designs, run `petalinux-boot-prebuilt` with the `--help` option:

   ```
   $ petalinux-boot-prebuilt --help
   ```

   The available PetaLinux reference designs will be listed at the bottom of the help text. On the console where you boot the reference design with QEMU, you should see the PetaLinux booting messages similar to this example:

```
INIT: version 2.88 booting
Starting Bootlog daemon: bootlogd.
Creating /dev/flash/* device nodes
Configuring network interfaces... udhcpc (v1.20.2) started
Sending discover...
xemacps e000b000.ps7-ethernet: Set clk to 124999998 Hz
xemacps e000b000.ps7-ethernet: link up (1000/FULL)
Sending select for 192.168.11.3...
Lease of 192.168.11.3 obtained, lease time 864000
/etc/udhcpc.d/50default: Adding DNS 172.19.128.1
/etc/udhcpc.d/50default: Adding DNS 172.19.129.1
done.
starting Busybox inet Daemon: inetd... done.
Starting uWeb server:
INIT: Entering runlevel: 5
Starting network benchmark server: netserver.
Starting syslogd/klogd: done
Stopping Bootlog daemon: bootlogd.


 _____       _            _      _
| ___ \     | |          | |    (_)
| |_/ / ___ | |_   __ _ | |     _  _ __    _   _ __  __
|  __/ / _ \| __| / _` || |    | || '_ \ | | | |\ \/ /
| |   |  __/| |_ | (_| || |____| || | | || |_| | >  <
\_|    \___| \__| \__,_|\_____/|_||_| |_| \__,_|/_/\_\

PetaLinux v2013.04 (Yocto 1.3) Xilinx-ZC702-14_5 ttyPS0

Xilinx-ZC702-14_5 login:
```

Figure 1: `petalinux-boot-prebuilt` with QEMU

2. Login to PetaLinux with the default user name `root` and password `root`.
3. Try the PetaLinux web demo:
   (a) Determine the IP address of the QEMU system. By default, all reference designs use DHCP to obtain an IP address. When booting in QEMU, a DHCP server is automatically started on the virtual network port.

   You can see the IP address of the PetaLinux system by reading the boot message similar to this:

   ```
   udhcpc (v1.14.3) started
   Sending discover...
   Sending select for 192.168.10.2...
   Lease of 192.168.10.2 obtained, lease time 864000
   adding dns 192.168.0.1
   ```

   above the PetaLinux banner on the QEMU console.

   In this example, the IP address of the QEMU system is `192.168.10.2`.

   Alternatively, you can use the ifconfig command on the QEMU console to get the PetaLinux System's IP address:

```
~ # ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0A:35:00:22:01
          inet addr:192.168.10.2  Bcast:192.168.10.255
Mask:255.255.255.0
          UP BROADCAST RUNNING  MTU:1500  Metric:1
          RX packets:82 errors:0 dropped:0 overruns:0 frame:0
          TX packets:67 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:13069 (12.7 KiB)  TX bytes:11936 (11.6 KiB)
          Interrupt:2

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

The `inet addr` of `eth0` is the IP address of the QEMU system.

By default, the QEMU subnet is configured as `192.168.10.0/24`. If this conflicts with your local network, please refer to Appendix A: QEMU with Customised Subnet Settings to manually specify an alternative virtual subnet.

If DHCP acquisition fails under QEMU, the most likely cause is your local firewall blocks the DHCP request. In such a case, please refer to Appendix B:IP Address Configuration to set the PetaLinux system IP address manually.

(b) Open a web browser on your workstation. Type:

`"http://<PetaLinux system IP Address>"`

in the web browser's address bar. In this example, it is

`"http://192.168.10.2"`

You should see the homepage of the PetaLinux web demo:

Figure 2: PetaLinux Web Demo

Test Pre-Built PetaLinux Image on Hardware

PetaLinux BSPs include pre-built FPGA bitstreams for each reference design, allowing you to quickly boot PetaLinux on your hardware. Here are the steps to test a pre-built PetaLinux image with hardware:

1. Power cycle the board.
2. Choose the correct board for the reference design. The naming mechanism of a PetaLinux reference design is as follows:

   `"<a>-<b>-<c>-<d>-<e>"`

   a) Vendor: the vendor of the board e.g. Xilinx.
   b) Board: the name of the board e.g. SP605.
   c) Bus Architecture: the type of architecture e.g. AXI.

IMPORTANT: *The Bus Architecture section has replaced the 'MMU' string since PetaLinux SDK v2.1.*

   d) Base Name: the base name e.g. 'full'.
   e) Xilinx Version: the version of the Xilinx tools that were used to synthesize the design e.g. 14.5.

   e.g.: `"Xilinx-SP605-AXI-full-14.5"` means:

   - the reference design uses the Xilinx SP605 development board as the hardware platform;
   - the hardware design of this reference design targets Xilinx tools 14.5 version;

- the reference design uses the AXI Bus Architecture.

3. Connect the JTAG port on the board with the JTAG cable to your workstation.
4. Connect the serial port on the board to your workstation. If there are both DCE and DTE connectors on the board, use DCE.
5. Connect the Ethernet port on the board to the local network via a network switch.
6. Power on the board.
7. Open a console on your workstation and then start your preferred serial communication program (e.g. `kermit`, `minicom`) with the baud rate set to 115200 on that console.
8. Run the petalinux-boot-prebuilt command as follows on your workstation:

```
$ petalinux-boot-prebuilt -p <reference design name>
```

This command will take some time to finish, please wait until you see the shell prompt again on the command console.

The figures below are examples of the messages on the workstation command console and on the serial console:

```
$ petalinux-boot-prebuilt -p Xilinx-ZC702-14.5
LEVEL 3 BOOT:
Configuring the FPGA...
FPGA configuration completed.
Downloading FSBL
FSBL download completed.
Downloading and booting Linux Kernel from RAM memory, this may take a few minutes.
Downloading and booting U-Boot from RAM memory, this may take a few minutes.
Launching XMD for file download and boot.
This may take a few minutes, depending on the size of your image.
Done..
Connect hyperterminal or kermit at 115200 baud.
Type "help" on the U-Boot prompt.

Launching XMD for file download and boot.
This may take a few minutes, depending on the size of your image.
Done..
Connect hyperterminal or kermit at 115200 baud.
Login using root/root as the username and password
```

Figure 3: Workstation console output for successful `petalinux-boot-prebuilt`

```
INIT: version 2.88 booting
Starting Bootlog daemon: bootlogd.
Creating /dev/flash/* device nodes
Configuring network interfaces... udhcpc (v1.20.2) started
Sending discover...
xemacps e000b000.ps7-ethernet: Set clk to 124999998 Hz
xemacps e000b000.ps7-ethernet: link up (1000/FULL)
Sending select for 192.168.11.3...
Lease of 192.168.11.3 obtained, lease time 864000
/etc/udhcpc.d/50default: Adding DNS 172.19.128.1
/etc/udhcpc.d/50default: Adding DNS 172.19.129.1
done.
starting Busybox inet Daemon: inetd... done.
Starting uWeb server:
INIT: Entering runlevel: 5
Starting network benchmark server: netserver.
Starting syslogd/klogd: done
Stopping Bootlog daemon: bootlogd.


 _____       _            _      _
| ___ \     | |          | |    (_)
| |_/ / ___ | |_   __ _  | |     _  _ __   _   _ __  __
|  __/ / _ \| __| / _` || |    | || '_ \ | | | |\ \/ /
| |   |  __/| |_ | (_| || |____| || | | || |_| | >  <
\_|    \___| \__| \__,_|\_____/|_||_| |_| \__,_|/_/\_\

PetaLinux v2013.04 (Yocto 1.3) Xilinx-ZC702-14_5 ttyPS0

Xilinx-ZC702-14_5 login:
```

Figure 4: Serial console output of `petalinux-boot-prebuilt`

By default, network settings for PetaLinux reference designs are configured using DHCP. The output you see may be slightly different from the above example, depending upon which PetaLinux reference design you test.

9. Type user name root and password root on the serial console to log into the PetaLinux system.
10. Determine the IP address of the PetaLinux system by reading the boot messages above the PetaLinux banner or alternatively, by running `ifconfig` on the system console.
11. Open a web browser and type:

    `"http://<PetaLinux System IP Address>"`

    in the address bar of the web browser. You should be able to see the PetaLinux web demo homepage.

Troubleshooting

If your local network does not have a DHCP server, the PetaLinux system will fail to acquire an IP address. If so, follow the same procedure described for the QEMU boot to manually specify the address. If the

`petalinux-boot-prebuilt` command fails, it is typically from a JTAG connectivity failure. Please ensure the board is powered on and your JTAG cable is properly connected. Please refer to the Xilinx JTAG cable and tools documentation for more detailed troubleshooting.

Rebuilding the Reference Design Software Image

So far, you have tested the PetaLinux reference design pre-built software image both with QEMU and on hardware. You can also rebuild the software image of a reference design software platform. The following subsections describe how to do it and how to test the resulting image.

Compile PetaLinux Reference Design Software

First of all, let's look at how to re-compile the PetaLinux reference design software image.

1. Go to the petalinux-dist directory by running this command on the workstation:

```
$ cd $PETALINUX/software/petalinux-dist
```

2. Run `make menuconfig` inside the petalinux-dist directory:

```
$ make menuconfig
```

TIP: *Remember to always source the "`settings.sh`" script in each new terminal window, before working with PetaLinux.*

PetaLinux SDK uses a cross-compiler to build PetaLinux. The cross-compiler should be from the PetaLinux SDK toolchains. If it is detected that a different compiler is provided which is not from the PetaLinux SDK toolchain an error is generated. If you want to use an external compiler, you will need to set the `EXTERN_COMPILER` environment variable.

This command will show the PetaLinux configuration menu on the console. Here is an example:

Figure 5: PetaLinux configuration menu

3. Select the **Vendor** sub-menu and then select the vendor of of the PetaLinux reference design you want to rebuild, e.g., Xilinx.
4. Select the **Vendor Products** sub-menu, e.g., **Xilinx Products**, and then select the reference design platform you want to rebuild, e.g., `"Xilinx-SP605-AXI-full-14.5"`. Here is an example of the platform selection result:

Figure 6: PetaLinux configuration menu - Vendor/Product Selection default settings

5. Exit the menuconfig and select **<Yes>** to question **Do you wish to save your new kernel configuration?**
6. It takes some time for PetaLinux to configure the Linux kernel settings. Please wait until the shell prompt shows again on the command console.
7. Run make to compile the PetaLinux software image:

```
$ make clean
$ make
```

8. The PetaLinux compilation progress will show on the console. Wait until the compilation finishes.

TIP:

- *A detailed compilation log will be in "`$PETALINUX/software/petalinux-dist/build.log`" file.*
- *Running `make PV=1` will both show the compilation details on the console and save it in the "`build.log`" file.*

When the PetaLinux compilation finishes, the generated images will be in the "`$PETALINUX/software/petalinux-dist/images`" and "`/tftpboot`" directories.

Here is an example of the PetaLinux compilation progress output:

```
...
[INFO ] Building ../user-modules
[INFO ]  Building kernel modules
[INFO ] Building ../user-apps
[INFO ] Building ../demo-apps
[INFO ]  Building demoapp:fwupgrade
[INFO ]  Building demoapp:gpio-demo
[INFO ]  Building demoapp:uWeb
[INFO ] Generating romfs:vendors
[INFO ] Generating romfs:include
[INFO ]  Installing headers
[INFO ] Generating romfs:rootfs
[INFO ]  Setting up romfs config
[INFO ]  Setting up stage config
[INFO ]  Updating for armv7a-vfp-neon
[INFO ]  Updating package manager
[INFO ]  Expanding rootfs
[INFO ]  Setting up romfs:sys_init
[INFO ] Generating romfs:lib
[INFO ] Generating romfs:include
[INFO ]  Installing headers
[INFO ] Generating romfs:../user-modules
[INFO ] Generating romfs:../user-apps
[INFO ] Generating romfs:../demo-apps
[INFO ]  Installing demoapp:fwupgrade
[INFO ]  Installing demoapp:gpio-demo
[INFO ]  Installing demoapp:uWeb
[INFO ] Installing kernel modules
[INFO ] Stripping kernel modules
[INFO ] romfs postprocessing
[INFO ] Configuring u-boot
[INFO ] Building u-boot
[INFO ] Relocating u-boot
[INFO ] Building image files
```

Figure 7: PetaLinux compilation progress output

The final software image is the image.elf file, living in the "`$PETALINUX/software/petalinux-dist/images`" folder. A copy is also placed in the "`/tftpboot`" directory or your development workstation, to support network-based kernel boot.

Test New Software Image with QEMU

Now you have successfully rebuilt the software system image, it is time to test it out.

1. Use `petalinux-qemu-boot` command to test the newly built PetaLinux software image:

   ```
   $ petalinux-qemu-boot
   ```

   The system boot messages will be shown on the console where QEMU is running.

2. When you see the login prompt on the QEMU console, login as `root`.

You may follow the same instructions described previously for exploring the QEMU image, for accessing the web demo, finding and changing the system IP address, and troubleshooting DHCP / firewall issues.

TIP:

- *With no options specified, `petalinux-qemu-boot boots` the most recently built software image*
- *You may run `petalinux-qemu-boot --help` to see how to specify an alternative software image rather than the default, and see other options of for command.*

Test New Software Image on Hardware

Next, let's test the rebuilt software image on the real hardware. Follow the instructions from the previous Test Pre-built PetaLinux Image on Hardware section, to connect the board, serial and JTAG correctly.

1. Use `petalinux-boot-prebuilt` to program the FPGA with the reference design pre-built bitstream:

   ```
   $ petalinux-boot-prebuilt -p <reference design name> -l 1
   ```

   This command will take a few moments, please wait until you see the shell prompt shows again on the command console.

   The `-l 1` option to `petalinux-boot-prebuilt` signals to do a Level 1 boot, that is, only configure the FPGA. Level 2 is FPGA + u-boot, and Level 3 is FPGA + pre-built Linux image.

2. Use `petalinux-jtag-boot` to download the built Linux image to the board and boot it:

   ```
   $ petalinux-jtag-boot
   ```

   This command will take a few minutes, downloading the entire kernel image over the JTAG link. Please wait until the shell prompt displays again on the serial console.

   IMPORTANT: *Currently Direct Kernel Boot via `petalinux-jtag-boot` supports MicroBlaze only, for Zynq, you can copy the kernel image to SD card or QSPI flash or use u-boot netboot to boot the kernel.*

3. Watch the serial console, you should see the Linux booting messages shown on the serial console.

You can now repeat the previous steps for connecting to the board via the serial console and the network demo.

Software Image Network Download

As the name suggests, the `petalinux-jtag-boot` command uses the Xilinx JTAG cable to download the system software image, and takes several minutes. This process can be made much more rapid by using the networking capabilities of the u-boot bootloader, which is enabled by default in all PetaLinux systems.

1. Make sure that the Linux image is in your TFTP directory `"/tftpboot"`.

2. Use `petalinux-boot-prebuilt` to program the FPGA with the reference design pre-built bitstream:

```
$ petalinux-boot-prebuilt -p <reference design name> -l 1
```

3. This command will take a few moments, please wait until you see the shell prompt again on the command console.

4. Use `petalinux-jtag-boot` to download the u-boot image to the board and boot it:

```
$ petalinux-jtag-boot -u
```

5. Watch the serial console. When you see `Hit any key to stop autoboot` on the console, press a key to stop auto boot, as shown below.

```
U-Boot 2013.01 (Apr 17 2013 - 10:58:28)

DRAM:  1 GiB
WARNING: Caches not enabled
MMC:   zynq_sdhci: 0
SF: Detected N25Q128 with page size 64 KiB, total 16 MiB
*** Warning - bad CRC, using default environment

In:    serial
Out:   serial
Err:   serial
Net:   Gem.e000b000
U-BOOT for Xilinx-ZC702-14.5

BOOTP broadcast 1
DHCP client bound to address 192.168.11.2
Hit any key to stop autoboot:  0
U-Boot-PetaLinux>
```

Figure 8: U-boot

6. Check if the u-boot environment variable serverip is set to your workstations IP by running the U-boot print command on the serial console:

```
U-Boot-PetaLinux> print serverip
```

The serverip variable defines the IP address from which u-boot will attempt to load the kernel image, using the TFTP protocol. It should correspond to the IP address of your workstation. When PetaLinux builds u-boot, it automatically set your workstations IP as the default u-boot serverip. However, any previously saved u-boot settings in flash will override this default.

7. Set the u-boot serverip to the IP of your workstation by running this command on the u-boot:

```
U-Boot-PetaLinux> set serverip <Your workstation IP>
```

8. Download the PetaLinux image with TFTP and then boot the image by running this command on u-boot:

```
U-Boot-PetaLinux> run netboot
```

You should be able to see the PetaLinux booting messages on the serial communication console.

Using u-boot and TFTP to load new kernel images is a great time-saver during the system development process, taking just seconds, instead of minutes that are required for a full JTAG boot of the Linux image.

Customising a Hardware Reference Design

So far, we have tested the pre-built hardware bitstream and the software image of the reference design, and have rebuilt the software image and tested it. In this section, we will customise the reference design.

Build a Hardware Project Based on Reference Design

The PetaLinux reference designs contain hardware project files which you can rebuild and customise. Here are the instructions to rebuild a reference design hardware system:

1. Go to the "user-platforms" directory in the PetaLinux tree:

   ```
   $ cd $PETALINUX/hardware/user-platforms
   ```

   This directory is intended to hold your own hardware projects, keeping them separate from Xilinx provided BSPs.

2. Copy the existing reference design:

   ```
   $ cp -r ../reference-designs/<Reference Design Name> <My Design Name>
   ```

   e.g.

   ```
   $ cp -r ../reference-designs/Xilinx-SP605-AXI-full-14.5 my-hw-project
   ```

3. Go to the the "<My Design Name>" directory:

   ```
   $ cd my-hw-project
   ```

4. Launch Xilinx XPS in command mode:

   ```
   $ xps -nw system.xmp
   ```

   It may take a few moments for XPS to load the hardware project, please wait until you see the XPS shell prompt:

   ```
   XPS%
   ```

5. Build the hardware in the XPS shell:

   ```
   XPS% run bits
   ```

   It will take quite some time to rebuild the hardware project. When it finishes building hardware project, you can see the XPS shell prompt shows again. Once this process completes, the FPGA bitstream is ready.

6. Next, you need to build the fs-boot bootloader.

Building and installing fs-boot

1. Export to SDK:

```
XPS% run exporttosdk
```

2. Close XPS, by entering the `exit` command:

```
XPS% exit
```

3. Change directory to the fs-boot subdirectory:

```
$ cd fs-boot
```

4. Launch the `init_bram` build step:

```
$ make init_bram
```

   This will build the PetaLinux BSP, the fs-boot bootloader application, and initialise the FPGA bitstream with the fs-boot program.

5. Change directory back up to the reference design folder:

```
$ cd ..
```

Creating a New PetaLinux Software Platform

You could use the existing reference design software platform for this new hardware project, but it is a good chance to try the `petalinux-new-platform` command instead. This section introduces how to create a new software platform with this command.

1. Use `petalinux-new-platform` to create a new software platform

```
$ petalinux-new-platform -v <Vendor Name> -p <Platform Name>
```

   e.g.:

```
$ petalinux-new-platform -v vendor -p my-hw-project
```

Note: You don't need to run `make menuconfig` to select the newly created platform; it has been automatically selected by the `petalinux-new-platform` command.

Synchronise Hardware and Software Platforms

The software platform we created in the last subsection doesn't know anything about the hardware platform yet. We need to tell the software platform the hardware platform settings so that we can build the software image for that hardware platform. PetaLinux provides a simple way to synchronise the hardware and software platforms. This section describes how to synchronise the hardware and software platforms using PetaLinux tools.

1. Go to the hardware project console to confirm the hardware build has finished. When it finishes, you should be able to see the XPS shell prompt. If not, wait until it finishes.
2. After XPS finishes building the hardware, exit the XPS shell by running this command in the XPS shell:

```
XPS% exit
```

3. Change directory to the "fs-boot" subdirectory:

```
$ cd fs-boot
```

4. Propagate the hardware parameters to the software platform by running PetaLinux command on the hardware project console:

```
$ petalinux-copy-autoconfig
```

   It will take a few moments for PetaLinux tools to configure the software platform with the hardware parameters. You should expect to see output similar to the following:

```
$ petalinux-copy-autoconfig
INFO: Using MSS file ./system.mss
INFO: Attempting vendor/platform auto-detect
INFO: Auto-detected platform "vendor/my-hw-project"
INFO: Using generic kernel platform
INFO: Merging platform settings into kernel configuration
Auto-config file successfully updated for platform "vendor/my-hw-project"
$
```

Rebuilding the new software image

1. After PetaLinux finishes configuring the software platform, go to the "petalinux-dist" directory in another console on the workstation:

```
$ cd $PETALINUX/software/petalinux-dist
```

2. Compile the PetaLinux with the new software platform by simply running make in the "petalinux-dist" directory:

```
$ make clean
$ make
```

   The PetaLinux compilation progress will be shown on the console, please wait until the compilation finishes.

Test the Customised PetaLinux Image with QEMU

Use the same procedure as described in the previous section Test the Newly Built PetaLinux Software with QEMU to test your new PetaLinux Image with QEMU:

1. Run the `petalinux-qemu-boot` command on the workstation:

```
$ petalinux-qemu-boot
```

2. Repeat the earlier steps try the PetaLinux web demo on your newly rebuilt software image.

Test the Customised PetaLinux Image on Hardware

Once the hardware build is complete, you can now download it to the board and boot the new software image.

1. Setup the board and connections as previously described.
2. Select the console window that is open on the hardware project directory (where you ran `xps` previously). Confirm you are in the main hardware project directory, not a subdirectory such as `"fs-boot"`.
3. Run the Xilinx impact command to program the hardware:

```
$ impact -batch etc/download.cmd
```

   This command takes a few moments to finish.

4. Boot the newly rebuilt u-boot image with `petalinux-jtag-boot` command as follows:

```
$ petalinux-jtag-boot -u
```

5. Watch the serial communication console. When you see `Hit any key to stop autoboot` on the console, type any key to stop auto boot.
6. As before, ensure the u-boot serverip setting is the IP of your workstation by running this command on u-boot terminal:

```
U-Boot-PetaLinux> print serverip
```

   If it is not, set it to your workstation IP:

```
U-Boot-PetaLinux> set serverip <Your workstation IP>
```

7. Download the PetaLinux image with TFTP and then boot the image by running this command on u-boot:

```
U-Boot-PetaLinux> run netboot
```

   You should see the PetaLinux booting messages on the serial communication console.

Appendix A: QEMU with Customised Subnet Settings

By default, PetaLinux uses `192.168.10.*` as the virtual subnet for QEMU. If it has been used by your local network or other subnets on your system, you need to use a customised subnet setting rather than the default one to run QEMU.

QEMU with petalinux-boot-prebuilt

If you use `petalinux-boot-prebuilt` command to test a reference design with QEMU, you can run the command as follows to specify a QEMU subnet:

```
$ petalinux-boot-prebuilt -p <Reference Design> -q --qboot-args
        "--subnet <subnet>/<bit mask>"
```

WARNING: *Be sure to enter these commands on one line*

e.g., to use subnet `192.168.20.*`:

```
$ petalinux-boot-prebuilt -p Xilinx-SP605-AXI-full-14.5 -q
        --qemu-args "--subnet 192.168.20.1/24"
```

QEMU with petalinux-qemu-boot

If you use `petalinux-qemu-boot` to test your newly built PetaLinux software with QEMU, you can run the command as follows to specify a QEMU subnet:

```
$ petalinux-qemu-boot --subnet <subnet IP>/<bit mask>
```

e.g:

```
$ petalinux-qemu-boot --subnet 192.168.20.1/24
```

Appendix B: IP Address Configuration

IP Address Configuration with ifconfig

After the PetaLinux system boots, you can set or change its IP address manually.

1. First determine which network the PetaLinux system is connected to.
   - If you are booting on real hardware, and the board is connected to a local network, the IP address of the system should be in the subnet of the local network.
   - If you are booting in QEMU, the IP address of the system should be in the QEMU subnet, which defaults to `192.168.10.0/24` (unless you have followed the instructions in Appendix A: Run QEMU with Customised Subnet Settings to specify a different subnet).
2. Use ifconfig command to set the systems IP address on the login console:

   ```
   # ifconfig eth0 <IP>
   ```

   e.g:

   ```
   # ifconfig eth0 192.168.10.10
   ```

3. Use `ifconfig` on the system login console again to confirm whether the IP address has been successfully set:

   ```
   # ifconfig
   ```

   You should be able to see the IP has been set to the interface eth0.

IP Address Configuration through PetaLinux Web Demo

You can change the network settings of the PetaLinux system through the PetaLinux web demo.
Note that you may see slightly different contents of the web page from the following examples, depending on the actual network settings of your system.

1. Open the PetaLinux web demo homepage by opening a web browser on your workstation, and then type `"http://<PetaLinux system IP Address>"` in the web browsers address bar.
2. Click **Network Settings** on the **NAVIGATION** menu on the left most of the homepage.

   The **Network Settings** page will be opened on the web browser showing the current network settings of the PetaLinux system. Here is an example:

Figure 9: PetaLinux Web Demo - Network Settings

3. Click **Edit** button on the **Network Settings** web page. The network settings editor page will be opened. Here is an example:

Figure 10: PetaLinux Web Demo - Network Settings Editor

4. Change the network settings in the network settings editor. Here is an example:

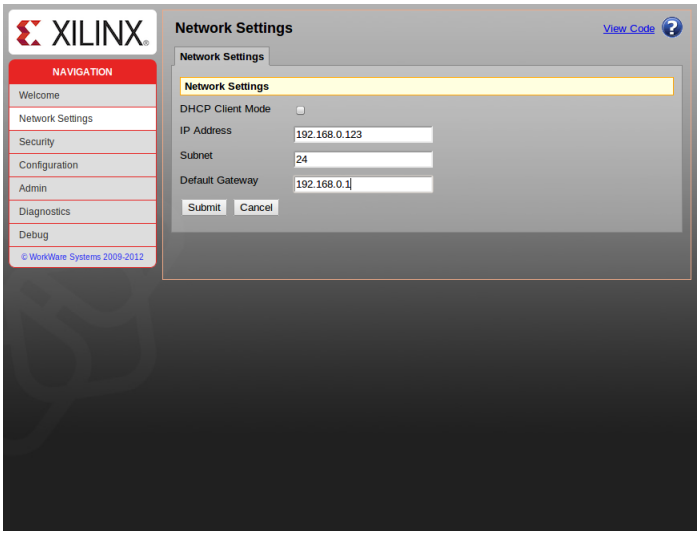

Figure 11: PetaLinux Web Demo - Change Network Settings

You may set different network settings from the above example.

5. Click **Submit** button below the network settings. The new network settings will appear on the web browser. Here is an example:

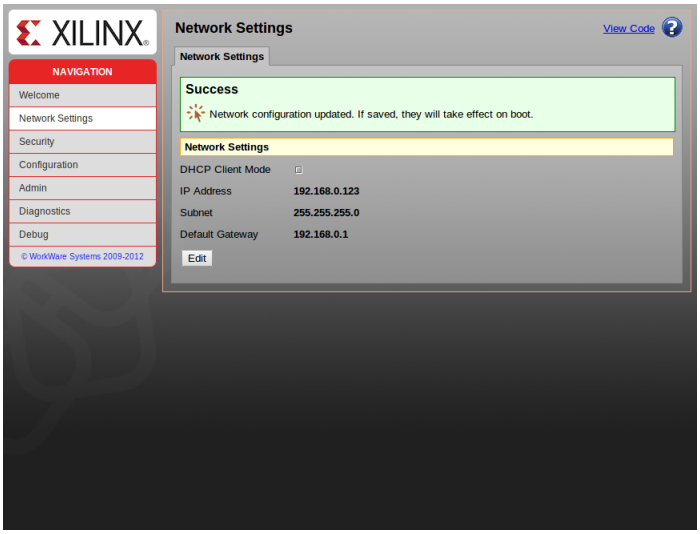

Figure 12: PetaLinux Web Demo - New Network Settings

WARNING: *The new network settings will not take effect until you restart your network interface.*

6. To restart your network interface, disable the network interface first by running `ifdown` command on the system login console:

```
# ifdown eth0
```

7. Then enable the network interface by running `ifup` command on the console:

```
# ifup eth0
```

   This command may take a few seconds to finish.

8. After the `ifup` command finishes, re-run `ifconfig` command on the console to confirm the new network settings

```
# ifconfig
```

Additional Resources

References

- PetaLinux SDK Application Development Guide (UG981)
- PetaLinux SDK Board Bringup Guide (UG980)
- PetaLinux SDK Eclipse Plugin Guide (UG979)
- PetaLinux SDK Firmware Upgrade Guide (UG983)
- PetaLinux SDK Getting Started Guide (UG977)
- PetaLinux SDK Installation Guide (UG976)
- PetaLinux SDK QEMU System Simulation Guide (UG982)