

PetaLinux SDK User Guide

Firmware Upgrade Guide

UG983 (v2012.12) December 17, 2012



Notice of Disclaimer

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of the Limited Warranties which can be viewed at <http://www.xilinx.com/warranty.htm>; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in Critical Applications: <http://www.xilinx.com/warranty.htm#critapps>.

© Copyright 2012 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Vivado, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.

Revision History

Date	Version	Notes
2010-11-24	1.3	Initial version for SDK 1.3 release
2011-04-04	2.1	Updated for PetaLinux SDK 2.1 release
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

Table of Contents

Revision History	1
Table of Contents	2
About this Guide	3
Related PetaLinux Documents	3
Production	4
Background Knowledge	4
Create a DTB Flash Partition	4
Update U-boot Environment Variables from U-boot	6
Update DTB in U-boot	6
Firmware Upgrade	7
Step 1: Package the Files Need to Upgrade	7
Step 2: Firmware Upgrade	8
Upgrading via the command line	8
Upgrading via the Demo Web Server	9

About this Guide

This document provides an example of how to do production configuration and firmware upgrades with PetaLinux.

This document assumes that you know how to manage your hardware project with Xilinx EDK, you are familiar with Linux and you have experience with PetaLinux.



IMPORTANT: *The methods described in this document are intended as examples only. PetaLinux SDK users are recommended to have their production configuration and firmware upgrade methods based on their requirements.*

Related PetaLinux Documents

The following other documents exist to help you to make the most of your PetaLinux experience:

- Application Development Guide
- Eclipse Plugin Guide
- Getting Started Guide
- Installation Guide
- QEMU System Simulation Guide

Production

During production manufacturing productions may require to be configured differently per board, with hardware parameters such as the Ethernet MAC address. In this section, examples are provide which detail how to make the same PetaLinux firmware images work with hardware which have different configuration on a per board basis.

Background Knowledge

MicroBlaze, Zynq and PowerPC Linux systems use Device Tree (DTS/DTB) files to configuration board specific information of the hardware system. Using the the Device Tree configuration different hardware parameters can be set which will configure the Linux Kernel and its respective drivers during boot.

By default, when PetaLinux boots, it uses the DTB within the PetaLinux Kernel image. The DTB (Device Tree Blob) is generated from a DTS file. PetaLinux can also boot with a DTB which is not located inside the PetaLinux Kernel image, this allows for easy modification of the DTB which can be stored in a seperate Flash partition.

Besides the Linux kernel, the U-Boot bootloader may also be configuration with board specific parameters such as Ethernet MAC address. However U-boot does not use a DTB for the hardware configuration, some parameters are available for configuration through the U-Boot environment variables. For example "ethaddr" is used to configure the Ethernet MAC address.

The following sections are going to detail how to create a Flash partition for the DTB, how to change the Ethernet MAC address in U-Boot and how to edit the DTB from U-Boot.

Create a DTB Flash Partition

Firmware images are generally stored in Flash, it is possible to store and boot with a DTB located in a Flash partition.

1. Run "petalinux-config-apps" inside the PetaLinux tree:

```
$ petalinux-config-apps
```

2. Go to "System Settings"
3. Enable the option "Boot image with the DTB partition" to allow U-Boot to load the DTB from a seperate Flash partition.
4. Go to "Flash Partition Table"
5. Add a "dtb" partition. It is recommended to use a size of 16KByte:

```
--- Partition X  
(dtb) name  
(0x4000) size
```

6. Save and exit menuconfig.
7. Rebuild PetaLinux inside the "petalinux-dist/" directory to update the both the firmware images with the new Flash partition settings. If you have already built PetaLinux, run "make clean" first to ensure all changes are propagated to all components.

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

If you boot your board with the newly built U-Boot or the kernel image, the additional "dtb" partition should be available.

The following steps use the Flash memory to store the U-Boot and kernel images as well as the DTB.

1. Boot the board into U-Boot
2. Setup the TFTP server location to load the firmware images from.

```
U-Boot-PetaLinux> set serverip <HOST IP>
```

3. Make sure the U-Boot image ("u-boot-s.bin"), kernel image ("image.ub") and DTB ("system.dtb") are in the TFTP directory on the TFTP server.
4. Make sure U-Boot IP has been set:

```
U-Boot-PetaLinux> print ipaddr
```

5. Update the U-Boot bootloader with the "update_uboot" command:

```
U-Boot-PetaLinux> run update_uboot
```

The "update_uboot" command will use TFTP to load the U-Boot image ("u-boot-s.bin") from the TFTP server and then save the image into the "boot" Flash partition.

6. Update the kernel image with the "update_kernel" command:

```
U-Boot-PetaLinux> run update_kernel
```

The "update_kernel" command will use TFTP to load the kernel image ("image.ub") from the TFTP server and then save the image into the "image" Flash partition.

7. Update the DTB with the "update_dtb" command:

```
U-Boot-PetaLinux> run update_dtb
```

The "update_dtb" command will use TFTP to load the DTB ("system.dtb") from the TFTP server and then save the DTB into the "dtb" Flash partition.

Next time the board is booted, it will boot into the kernel with the DTB located in the "dtb" Flash partition.

Update U-boot Environment Variables from U-boot

In U-Boot the Ethernet MAC address is configured as an environment variable. The MAC address environment variable can be changed from the U-Boot console and then saved into the "bootenv" Flash partition using the "saveenv" command.

1. Boot the board into U-Boot and stop at the console (cancel auto boot if enabled)
2. Change the U-Boot MAC address environment variable:

```
U-Boot-PetaLinux> set ethaddr AA:BB:CC:DD:EE:FF
```

Where "AA:BB:CC:DD:EE:FF" is the MAC address you want to configure.

3. Save the u-boot environment variables into the flash:

```
U-Boot-PetaLinux> saveenv
```

Update DTB in U-boot

This section details how to modify the DTB from within the U-Boot console. The example in this section covers the changing of the MAC address in the DTB.

1. Load the DTB from the "dtb" Flash partition into the memory:

```
U-Boot-PetaLinux> run get_dtb
```

2. Set the MAC address in DTB:

```
U-Boot-PetaLinux> fdt set <PATH TO local-mac-address PROPERTY  
OF ETHERNET CORE> local-mac-address "[AA BB CC DD EE FF]"
```

Please note that the above set MAC address command is in one line only.

e.g.: Change the MAC address of "ethernetlite":

```
u-boot> fdt set /plb@0/ethernet@81000000 local-mac-address  
"[AA BB CC DD EE FF]"
```

e.g.: Change the MAC address of "ll_temac":

```
u-boot> fdt set /plb@0/xps-ll-temac@82780000/ethernet@82780000  
local-mac-address "[AA BB CC DD EE FF]"
```

Where "AA BB CC DD EE FF" is the MAC address you want to configure.

3. Save the changed DTB back to Flash:

```
U-Boot-PetaLinux> run install_dtb
```

Firmware Upgrade

Since PetaLinux v1.3, PetaLinux has an easy process for firmware upgrades. With the PetaLinux firmware upgrade tools upgrading firmware is a two-step process, packaging the firmware images and FPGA bitstream and installing the images into the destination Flash memory.

NOTE:



- That PetaLinux assumes all the images and or the bitstream are saved in the Flash and that each image/bitstream corresponds to a flash partition.
 - "*petalinux-package-firmware*" and the "*firmware-upgrade*" tools do not support Zynq SD Card boot configurations.
-

Step 1: Package the Files Need to Upgrade

The PetaLinux tool "*petalinux-package-firmware*" is used to package the images and bitstream required for the upgrade. e.g. to create a firmware upgrade package containing the FPGA bitstream, kernel image and a u-boot image, you can use the package firmware upgrade command as follows:

```
$ petalinux-package-firmware --image <Path-to-image.ub>
  --uboot <Path-to-u-boot-s.bin>
  --fpga <Path-to-the-FPGA bitstream>
```

Please note that the above command must be execute as a single line only.

- "*--image*" the image file specified with this argument is by default store in the "*image*" flash partition.
- "*--boot*" (*Zynq Only*) the BOOT.BIN file specified with this argument is by default stored in the "*boot*" flash partition.
- "*--uboot*" (*MicroBlaze/PPC only*) the u-boot file specified with this argument is by default stored in the "*boot*" flash partition.
- "*--fpga*" (*MicroBlaze/PPC only*) the bitstream file specified with this argument is by default stored in the "*fpga*" flash partition.
- By default, the package firmware tool automatically detects the configured Flash setup for the selected platform.

You can also generate a firmware upgrade package with jffs2 and other Flash partitions using *petalinux-package-firmware* command. For more details of the *petalinux-package-firmware* command, please use "*-h*" option, which will show the usage of the command.

Step 2: Firmware Upgrade

After you have created the firmware upgrade package, you can upgrade the firmware on the target with PetaLinux firmware upgrade command "upgrade-firmware" on a running PetaLinux system.

By default, the Firmware Upgrade Tools are enabled and built into the image. You can configure this via the `petalinux-config-apps` command and the associated menuconfig:

```
PetaLinux Configuration --->
  PetaLinux Demo Applications --->
    [*] fwupgrade
```



IMPORTANT: Please note that the PetaLinux `upgrade-firmware` command can only be used to upgrade with the package generated by the `petalinux-package-firmware` tool.

There are two different ways to upgrade the firmware with the "upgrade-firmware" command, the first is by using the command on the console and the second is via the demo web server (if configured).

Please note:

- All the below methods of firmware upgrade will upgrade the images which are saved in the Flash partitions and will then do a soft restart automatically to load the system with the new software images.
- To make the soft restart successfully in a MicroBlaze Linux system, you need to have a soft reset GPIO in your system. Please refer to the PetaLinux Documentation for the details on configuration and setup of a Reset GPIO.
- The firmware upgrade method introduced in this section is a demonstration of how to do firmware upgrade with PetaLinux. You are welcome to design your own firmware upgrade procedure.

Upgrading via the command line

- If your upgrade package is on the Internet, you can use the "upgrade-firmware" to wget it from the Internet and then upgrade the system with it:

```
# upgrade-firmware <URL to your upgrade package tar ball>
```

e.g.:

```
# upgrade-firmware http://192.168.0.1/firmware.tar.gz
```

- If your upgrade package is on your host, you can ftp your firmware upgrade package to the running PetaLinux system and then use the "upgrade-firmware" command to upgrade the firmware. e.g.:

```
# upgrade-firmware /var/ftp/firmware.tar.gz
```

Upgrading via the Demo Web Server

You can also use PetaLinux demo web server to do the firmware upgrade:

- Browse to the PetaLinux running system's homepage from your host if you are using PetaLinux uWeb demo web server which is the default web server for PetaLinux.
- Select the "Admin" link from the "NAVIGATION" menu at the left of the web page, and then select the "Firmware Upgrade" tab:

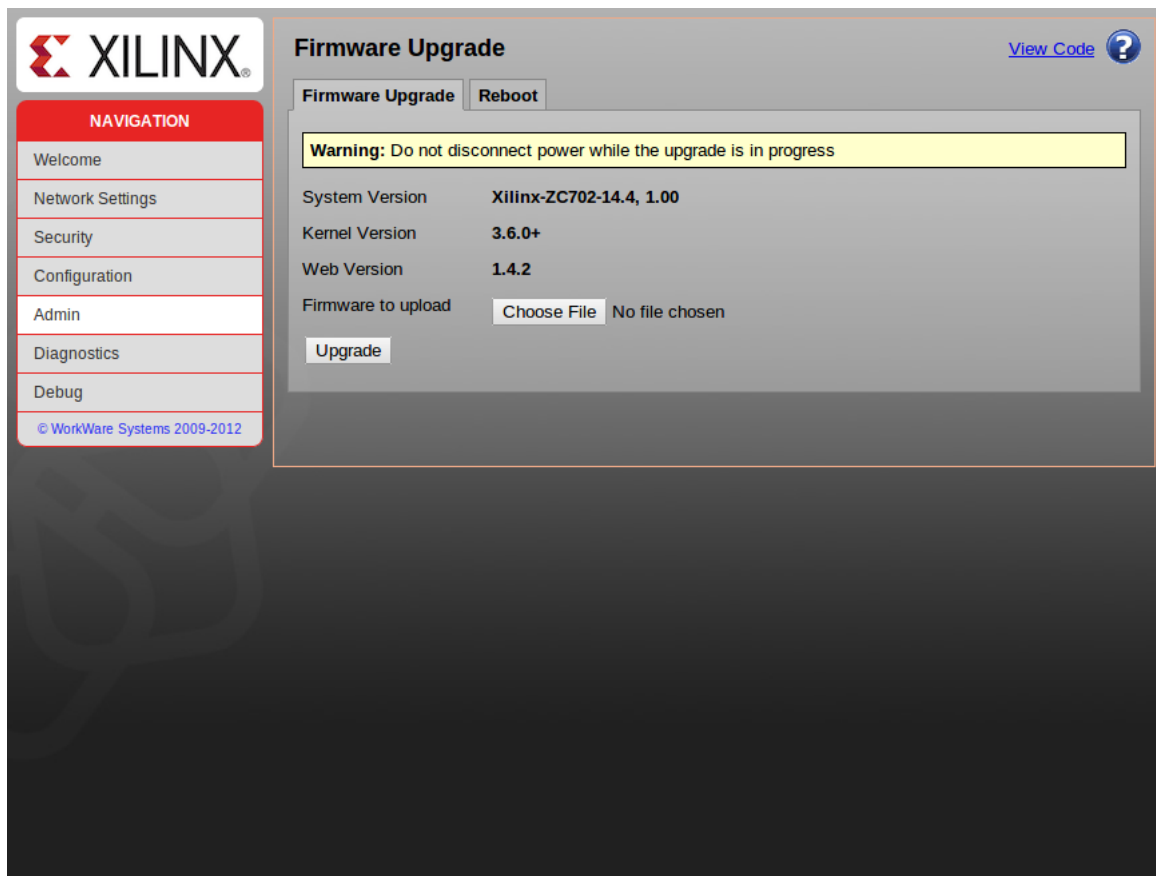


Figure 1: uWeb Firmware Upgrade Page

- Select the firmware upgrade package from your host by clicking the "Browse . . ." button.
- Click "Upgrade" to do the firmware upgrade. Please note that the firmware can take a while, please be patient and wait until it finishes.