



# Production And Firmware Upgrade Example

*v2.1 April 4th, 2011*



# Table of Contents

---

## Table of Contents

<b>Table of Contents</b> .....	<b>2</b>
<b>About This Guide</b> .....	<b>3</b>
<b>Related Documents</b> .....	<b>3</b>
<b>Production</b> .....	<b>3</b>
Background Knowledge.....	3
Create a DTB Flash Partition.....	4
Update U-boot Environment Variables from U-boot.....	5
Update DTB in U-boot.....	6
<b>Firmware Upgrade</b> .....	<b>6</b>
Step 1: Package the Files Need to Upgrade.....	7
Step 2: Firmware Upgrade.....	7
<b>Revision History</b> .....	<b>9</b>

## About This Guide

---

This document provides an example on how to do production and firmware upgrade with PetaLinux. This document assumes that you have know how to manage your hardware project with Xilinx EDK, you are familiar with Linux and you have experience with PetaLinux.

**The methods described in this document are intended as examples only. PetaLinux SDK users are recommended to have their production and firmware upgrade method based on their requirements. PetaLogix accepts no liability or responsibility for any losses caused due to procedures described in this document.**

## Related Documents

---

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

- PetaLinux SDK Installation Guide
- Getting Started with PetaLinux SDK
- Guide to QEMU System Simulation in PetaLinux SDK
- Guide to Create and Debug User Applications
- PetaLinux Migration Guide

You can also find PetaLinux documentation online at

- [http://www.petalogix.com/resources/documentation/petalinux\\_sdk](http://www.petalogix.com/resources/documentation/petalinux_sdk)

## Production

---

After have testing prototype, we are going to producing our final products. Different product can have slightly different hardware parameters such as MAC address. In this section, we give an example on how to make the same PetaLinux image works with the hardware with slight difference.

## Background Knowledge

Both MicroBlaze and PowerPC Linux systems use Device Trees (DTS/DTB) files to keep the information of the hardware system, we need to have different DTS files for different hardware if we want the Linux driver to notice the hardware difference.

By default, when PetaLinux boots, it uses the DTB file within the PetaLinux image file. A DTB(Device Tree Blob) file is built from a DTS file. PetaLinux can also boot with a DTB file out of the PetaLinux image file. By using different DTB files for different hardware, PetaLinux can boot with different hardware details such as MAC address

Besides the Linux kernel, the u-boot boot loader may also care about the hardware details difference such as MAC address. U-boot doesn't use DTB file for the hardware information, it uses environment variable "ethaddr" for the MAC address. And thus, if different hardware has different MAC address, which is usually the case, we need to have different DTB files and difference u-boot "ethaddr" environment variables.

The next sub-sections are going to talk about how to create a flash partition for the separate DTB file, how to change the MAC address in u-boot and how to edit the DTB file from u-boot.

## Create a DTB Flash Partition

As we have learned from the last sub-section, PetaLinux can boot with different DTB file. We can have different DTB files for the hardware with different MAC addresses. Usually we save our image files in the Flash, we can save the DTB file in the flash too. PetaLinux preserve flash partition name "dtb" for the DTB file. You can use menuconfig to create the DTB partition:

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

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

2. Go to "System Settings" ---> "Flash Partition Table" --->
3. Add a "dtb" partition. Its size is usually 16KByte.:

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

4. Save and exit menuconfig.
5. Rebuild PetaLinux inside the "petalinux-dist/" directory to update the both the u-boot and the kernel images with the new Flash partition settings. If you have already built PetaLinux, remove the "petalinux-dist/romfs/" directory before rebuilding PetaLinux to make sure only the new Flash partition settings are in the romfs.:

```
$ cd ${PETALINUX}/software/petalinux-dist
$ rm -rf romfs
$ make
```

If you boot your board with the newly built u-boot or the kernel image, the u-boot and the kernel can recognize the new “dtb” flash partition.

This procedure uses Flash memory to store the u-boot boot loader, the kernel image and the DTB file. After rebuilding PetaLinux, these images will be stored into flash, using the following procedure:

1. Boot the board into u-boot
2. Since we are going to use TFTP to download the u-boot and the kernel images from the host, make sure TFTP server has installed and worked properly on the host.
3. Make sure the u-boot file “u-boot-s.bin” and “image.ub” and “system.dtb” are in the tftp directory on the TFTP server host.
4. Set the TFTP server IP from u-boot:

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

5. Make sure u-boot IP has been set:

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

6. Make sure the TFTP server can be pinged from u-boot:

```
U-Boot-PetaLinux> ping <HOST IP>
host <HOST IP> is alive
```

7. Update the u-boot boot loader with the “update\_uboot” command:

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

The “update\_uboot” command will tftp the “u-boot-s.bin” file from the TFTP server to u-boot and then save the file into the “boot” flash partition.

8. Update the kernel image with the “update\_kernel” command:

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

The “update\_kernel” command will tftp the “image.ub” file from the TFTP server to u-boot and then save the file into the “image” flash partition.

9. Update the DTB file with the “update\_dtb” command:

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

The “update\_dtb” command will tftp the “system.dtb” file from the TFTP server to u-boot and then save the file into the “dtb” flash partition.

Next time, when you boot the board, it should be able to boot into the kernel with the DTB file saved in the flash from u-boot.

## Update U-boot Environment Variables from U-boot

In u-boot, the MAC address is saved as an environment variable. We can update the MAC address u-boot environment variable if we want to use different MAC address. Here is the way to update the MAC address from u-boot:

1. After booting the board into the u-boot, update the u-boot MAC address environment variable:

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

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

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

## Update DTB in U-boot

It is allowed to edit the DTB file from u-boot although u-boot doesn't use it. This section shows how to update the MAC address in the DTB file from u-boot. Here are the steps:

1. Load the DTB file 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 Ethernet lite:

```
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]"
```

3. Copy the changed DTB file back to flash:

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

When you boot the board next time, it should auto boot into the kernel with the DTB file you have updated from u-boot.

## Firmware Upgrade

---

From PetaLinux v1.3, PetaLogix introduce a two-steps easy way to do firmware upgrade. With the PetaLinux firmware upgrade tools, you just need to package your new versions of hardware bitstream and/or your bootloaders and kernel image at first, and then run the PetaLinux firmware upgrade tool from the running PetaLinux system to upgrade your firmware.

**Please note that PetaLinux assumes all the images and or the bitstream are saved in the Flash and each file corresponds to a flash partition.**

### Step 1: Package the Files Need to Upgrade

PetaLinux tool “petalinux-package-firmware” is used to package the files need to upgrade. E.g. to create a firmware file containing the FPGA bitstream, kernel image and a u-boot image, I 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 are in one line only.

- “- -image” the image file specified with this argument is supposed to be put into the “image” flash partition.
- “- -uboot” the u-boot file specified with this argument is supposed to be put into the “boot” flash partition.
- “- -fpga” the bistream file specified with this argument is supposed to be put into the “fpga” flash partition.
- By default, the package firmware tool assumes the Flash is parallel flash, if you are using SPI flash, please use “- -flash spi” argument. E.g.:

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

You can also generate a firmware upgrade package with jffs2 and other Flash partitions with the PetaLinux packaging firmware command. For more details of the command, please use “-h” option, which will show the usage of the command.

## Step 2: Firmware Upgrade

After you have finished your firmware upgrade package, you can upgrade the firmware with PetaLinux firmware upgrade command “upgrade-firmware” on a running PetaLinux system.

By default, the “upgrade-firmware” command is selected to build into the PetaLinux. You can find it here from the menuconfig:

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

**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 a couple of ways to upgrade the firmware with the “upgrade-firmware” command:

- 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>
```

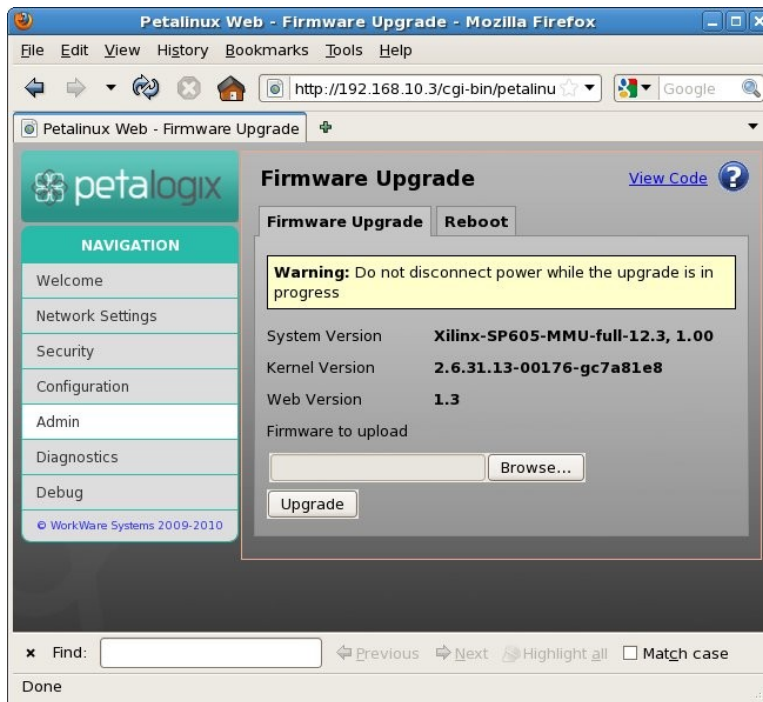
E.g.:

```
# 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
```

- 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” from the “NAVIGATION” list at the right of the web page, and then select the “Firmware Upgrade” tab:



- 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 to wait until it finishes.

Please note:

- All the above several firmware upgrade methods will upgrade the images saved in the Flash partitions and then it will 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 hardware project files of your PetaLinux reference design BSP to see how to configure the system with a soft reset GPIO.
- The firmware upgrade method introduced in this section is a demonstration on how to do firmware upgrade with PetaLinux. You are welcomed to design your own firmware upgrade procedure.

## Revision History

---

Date	Version	Notes
11/24/2010	0.01	Initial version for SDK 1.3 release
04/04/11	2.1	Updated for PetaLinux SDK 2.1 release