PetaLinux SDK User Guide

Firmware Upgrade Guide

UG983 (v2013.10) November 25, 2013

XILINX®

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
2013-04-29	2013.04	Updated for PetaLinux SDK 2013.04 release
2013-11-25	2013.10	Updated for PetaLinux SDK 2013.10 release

Online Updates

Please refer to the PetaLinux v2013.10 Master Answer Record (Xilinx Answer Record #55776) for the latest updates on PetaLinux SDK usage and documentation.

XILINX

Table of Contents

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 Embedded hardware tools (EDK or Vivado), you are familiar with Linux and you have experience with PetaLinux.

WARNING: *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.*

Production

During production boards may require individual configuration, with hardware parameters such as the Ethernet MAC address. In this section, examples are provided which detail how to make the same firmware images work with hardware which has different configurations on a per board basis.

Background Knowledge

Zynq and MicroBlaze Linux systems use Device Tree (DTS/DTB) files to configure board specific information of the hardware system. Using 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 Linux boots, it uses the DTB within the kernel image. The DTB (Device Tree Blob) is generated from a DTS file. Linux can also boot with a DTB which is not located inside the kernel image, this allows for easy modification of the DTB which can be stored in a separate flash partition.

Besides the Linux kernel, the U-Boot bootloader may also be configured with board specific parameters such as Ethernet MAC address. Although U-boot does not use a DTB for the hardware configuration, some parameters are available for configuration through the U-Boot environment variables.

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. Change into the directory of your project.

```
$ cd <project-root>
```

2. Run `petalinux-config` from within the PetaLinux project directory:

```
$ petalinux-config
```

3. Enable the option **Boot image with the DTB partition** to allow U-Boot to load the DTB from a separate flash partition.

```
*** Bootloader settings ***
[*] Boot image with the DTB partition
```

4. Go to **Flash Partition Table**

5. Add a **dtb** partition. It is recommended to use a size of at least 16 KByte (or 64 KByte for SPI flash) and must aligned to flash sector size:

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

6. Save and exit menuconfig.
7. Build the project to update both the firmware images with the new flash partition settings. If you have previously built the project, run `petalinux-build -x clean` first to ensure all changes are propagated to all components.

```
$ petalinux-build -x clean
$ petalinux-build
```

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

Update Flash Partition from U-boot

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.

TIP: *`update_uboot` is a prepackaged u-boot script, generated by the PetaLinux build system. Run `printenv update_uboot` within u-boot to examine its contents.*

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.

1. Load the DTB from the `dtb` flash partition into the memory:

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

2. Make any required changes using `fdt set` command.
3. Set the size of the file to save to flash:

```
U-Boot-PetaLinux> set filesize ${dtbsize}
```

4. Save the changed DTB back to flash:

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

Firmware Upgrade

PetaLinux contains some basic firmware upgrade and management tools, intended to demonstrate one possible approach to these important product capabilities.

With these upgrade tools, upgrading firmware is a two-step process: packaging the firmware images and FPGA bitstream, and installing the images into the board flash memory.

IMPORTANT:

- *PetaLinux assumes that 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 `upgrade-firmware` tools do not support Zynq SD Card boot configurations.*

Step 1: Package the Upgrade Image

The PetaLinux `petalinux-package` tool has a `--firmware` option 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 --linux <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 executed as **a single line** only.

- `--image` - the Linux kernel image file specified with this argument is by default stored 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 only)* - the u-boot file specified with this argument is by default stored in the `boot` flash partition.
- `--fpga` *(MicroBlaze 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.

TIP: *The `--uboot` and `--fpga` options do not apply for Zynq, because the bitstream and u-boot firmware are instead bundled as part of `boot.bin`, using the Xilinx `bootgen` tool.*

You can also generate a firmware upgrade package with `jffs2` and other flash partitions using `petalinux-package --firmware` command. For further details on the petalinux firmware packaging tool, please use the `petalinux-package --firmware --help` command, which will show the full usage of the tool.

Step 2: Firmware Upgrade

After you have created the firmware package, you can upgrade the firmware on the target with the `upgrade-firmware` tool 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 -c rootfs` command and the associated menuconfig:

```
Apps --->
    [*] fwupgrade
```

WARNING: *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 reset to boot the system with the new software images.
- To make the soft reset successfully in a MicroBlaze Linux system, you need to have a soft reset GPIO in your system.
- The firmware upgrade method introduced in this section is a demonstration of how to do firmware upgrade with PetaLinux. It is recommended to design your own firmware upgrade procedure.

Upgrading via the command line

- The `upgrade-firmware` tool can either fetch image files over the network using the HTTP protocol, or directly install a package from the local root file system. e.g.:

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

or

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

WARNING: *Please do not interrupt (powercycle) the board while it is upgrading the firmware.*

Upgrading via the uWeb Demo Web Server

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

IMPORTANT: *This can only be completed if the uWeb demo webserver is installed on the target system (it is enabled by default).*

1. Using a Web Browser, navigate to the target system by IP or hostname.
2. Select the `Admin` link from the **NAVIGATION** menu at the left of the web page, and then select the `Firmware Upgrade` tab:

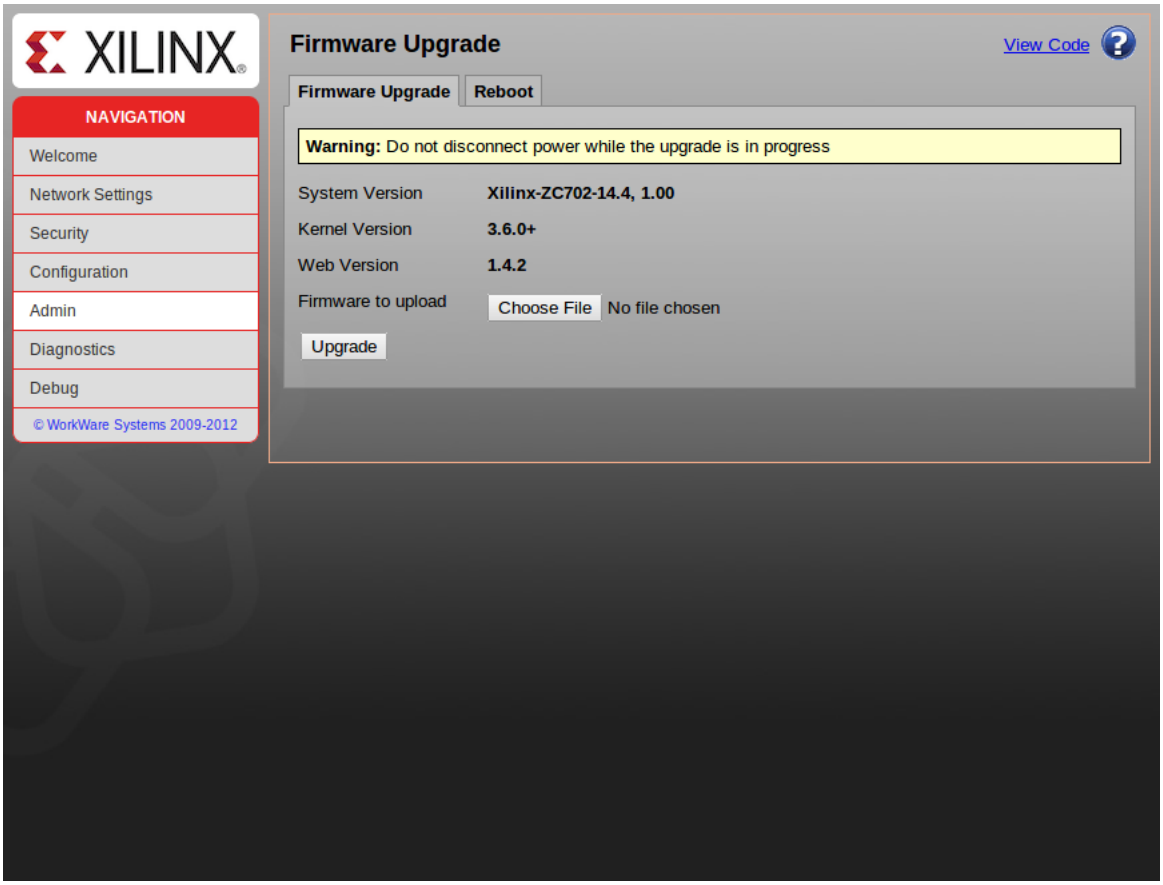

Figure 1: uWeb Firmware Upgrade Page

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

WARNING: *Please do not interrupt (powercycle) the board while it is upgrading the firmware.*

Additional Resources

References

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

PetaLinux SDK Documentation is available at http://www.xilinx.com/petalinux.