PXE-Server

It is possible to boot modules via a PXE server, e.g. to avoid the installation of local disk images.

Installing the PXE server

This tutorial was done with a Debian 8. It will only be possible to automatically boot an OS according to the module's MAC address. In this configuration there is no GUI to select a boot image.

All operations where done with root permissions.

First the following services have to be installed:

```
apt-get install dnsmasq nfs-kernel-server
```

`dnsmasq` serves as the DHCP Server to tell the clients what IP address to use and which kernel they have to load. Also it runs the TFTP Server that provides the kernel images.

`nfs-kernel-server` provides the NFS network share for the RootFS and a persistent storage.

Setting up dnsmasq

The config file can be opened with

```
nano /etc/dnsmasq.conf
```

All DHCP Options used in the following config can be found here Link The following shows an example configuration for 3 different modules and 4 different kernels.

```
#Binds the DHCP server to eth0
interface=eth0
#Sets the IP range with a lease time of 8 hours
dhcp-range=192.168.13.201,192.168.13.249,255.255.255.0,8h

#Enables the TFTP service of dnsmasq
enable-tftp
#Bind the TFTP root folder  to the set folder
tftp-root=/srv/tftpboot/
#DHCP Option 66 sets which TFTP Server will be used by the client
dhcp-option=66,192.168.13.80
#DHCP Option 42 sets a NTP time server
dhcp-option=42,130.133.1.10


#The following sets a flag for all MAC addresses starting with the given
number,
```

```
#so it is possible to set different setups for different modules
dhcp-host=00:14:2d:*:*:*,set:toradexApalis
dhcp-host=00:0e:c6:*:*:*,set:toradexColibriAndroid
#"set:christmannExynosLinux" has to be changed to
"set:christmannExynosAndroid" in order to load
#the Android kernel
dhcp-host=70:b3:d5:56:*:*,set:christmannExynosLinux



#These different tags can be used with "tag:"tagname","option""
#"dhcp-boot" defines which kernel will be loaded from the TFTP root path
#"dhcp-option" gives an option to the set MAC address group

#Kernel filename for Toradex Apalis T30 provided by the TFTP Server
dhcp-boot=tag:toradexApalis,toradexApalis_uImage
#Root path that is shared by the nfs-kernel-server
dhcp-option=tag:toradexApalis,17,/srv/nfs/apalisT30/rootfs/

#Kernel filename for Toradex Colibri T20
dhcp-boot=tag:toradexColibri,toradexColibri_uImage
#Root path
dhcp-option=tag:toradexColibri,17,/srv/nfs/colibriT20/linux/rootfs/

#Kernel filename for Christmann Apalis Exynos Linux
dhcp-boot=tag:christmannExynosLinux,uImage-ExynosLinux
#Root path
dhcp-option=tag:christmannExynosLinux,17,/srv/nfs/apalisExynos/linux/rootfs/

#Kernel filename for Christmann Apalis Exynos Android
dhcp-boot=tag:christmannExynosAndroid,uImage-ExynosAndroid
#Root path
dhcp-
option=tag:christmannExynosAndroid,17,/srv/nfs/apalisExynos/android/rootfs
```

After the `dnsmasq.conf` has been changed, the dnsmasq service has to be restarted in order to apply the changes using:

```
/etc/init.d/dnsmasq restart
```

Then the `uImage` files must be copied to `/srv/tftpboot/*` with the name that was defined in `dhcp-boot`.

Setting up nfs-kernel-server

The RootFS and/or a persistent storage are exported to the client via NFS. This is the location where the root filesystem of the desired distribution must be copied to.

The shared/exported folders can be edited in this file:

```
nano /etc/exports
```

Here is an example configuration:

```
/srv/nfs/apalisT30/rootfs       *(rw,no_root_squash,no_subtree_check)
/srv/nfs/storage/               *(rw,async,no_root_squash)
/srv/nfs/colibriT20             *(rw,no_root_squash,no_subtree_check)
/srv/nfs/apalisExynos/          *(ro,no_root_squash,no_subtree_check)
```

First in line is the folder that will be shared via NFS.
The * in front of the bracket sets the clients that are allowed to access the share. In this case everyone can access the share, but also singe IP addresses or ranges can be set.
The Arguments in the brackets are the following:

rw	Read Write access
ro	Read Only access
async	Better performance with the danger of data loss if the server crashes or is shut down. Default is sync
no_root_squash	Enables access to files that are owned by root on the server
no_subtree_check	disables the subtree checking when accessing a file, this in on by default

A detailed description can be found at Ubuntuusers or with man exports

To apply the changes the nfs-kernel-server has to be restarted.

```
/etc/init.d/nfs-kernel-server restart
```

Client configuration

The clients have to be set up in order to boot from the PXE server and not from the internal flash storage or HHD.
For CXP modules the network boot option in the BIOS has to be activated. In most cases that is enough.

The Apalis and Colibri modules must have an u-boot that supports booting over network (version and higher). And some things ave to be set in the u-boot enviornmentals as described below.

The u-boot can be reached by connecting a serial console to the module and then pressing any key when starting the module to abort the booting process.

Christmann Apalis Exynos

Tested with **U-Boot 2015.07-rc1-00408-g012681b-dirty**

The following variables have to be set in u-boot:

```
setenv usbethaddr
setenv nfsboot 'usb reset; dhcp; mmc dev 1; mmc read 22000000 3000 100;
```

```
bootm 23e00000 - 22000000'
setenv bootcmd 'run nfsboot; usb start; mmc dev 1; mmc read 0x20008000 600
2700; mmc read 0x22000000 3000 100; bootm 0x20008000 - 0x22000000'
setenv bootargs 'root=/dev/nfs rw netdevwait console=ttySAC2,115200n8 init -
-no-log'
```

 But it is necessary at the moment to have an SD card with the original Image, or with the flattend device tree blob at sector 3000 of the SD card.

 At the moment the kernel stops booting with `drm_kms_helper: panic occurred, switching back to text console`

Toradex Apalis T30

Tested with **U-Boot 2014.10**

Only the option for `nfsboot` has to be added to the `bootcmd` in u-boot. Note that in the following steps the MAC address has to be replaced with the actual address of the module. 

If only network boot is desired the following commands have to be executet in the u-boot enviornment:

```
setenv serverip
setenv ipaddr
setenv ethaddr 00:14:2D:00:00:00
setenv nfsboot 'run setup; setenv bootargs ${defargs} ${nfsargs}
${setupargs} ${vidargs}; echo Booting via DHCP/TFTP/NFS...; dhcp
${kernel_addr_r} && bootm ${kernel_addr_r} - ${dtbparam}'
setenv bootcmd 'run nfsboot; echo; echo nfsboot failed'
saveenv
```

If the original bootorder should not be replaced the commands are the following:

```
setenv serverip
setenv ipaddr
setenv ethaddr 00:14:2D:00:00:00
setenv bootcmd 'run nfsboot; echo; echo nfsboot failed; run emmcboot; echo;
echo emmcboot failed'
saveenv
```

But this will result in one error message `*** ERROR: `serverip' not set` because the module then trys to load a flash image from a server whos IP is not set in `serverip`.