PhyCORE-AM3517 Linux Quickstart PD11.1.x

1 About this Quickstart

This document describes how to install and work with the Linux Board Support Package (BSP) for the phyCORE-AM3517 platform. This BSP provides a fundamental software platform for development, deployment and execution on the phyCORE-AM3517.

The Quickstart contains instructions for:

- Host Setup
- Boot Options
- Building a BSP (Platform, Kernel, Root Filesystem)
- Flashing images to NAND (X-Loader, U-Boot, Kernel, Root Filesystem)

2 System Requirements

2.1 Hardware Requirements:

The following hardware components are included in the phyCORE-AM4317 Linux Rapid Development Kit (part number KPCM-048-L) and are necessary for completing the instructions in this QuickStart:

- phyCORE-AM3517 System on Module (Part No. PCM-048-5533ER1I0-1A)
- phyCORE-AM3517 Carrier Board (Part# PCM-961)
- AC adapter supplying 5 VDC /3.2A adapter, center positive
- Straight Ethernet cable
- Serial cable (RS-232)
- USB Standard A to mini-B cable

2.2 Software Requirements:

- Ubuntu 10.04 LTS

3 Host Setup

The phyCORE-AM3517 (PCM-048) has been developed and tested with Ubuntu 10.04 LTS Lucid Lynx [Installation Guide]. Although it is possible to use a different OS, some setup information will contain OS-specific commands and paths for settings.

Update repositories and upgrade installed packages:

```
sudo apt-get update
sudo apt-get upgrade
```

3.1 Server Setup

Support for installing and setting up TFTP, NFS, and Samba server settings to enable communication between multiple systems and the target.

3.1.1 TFTP

TFTP allows files to be downloaded from one machine to another. With most embedded Linux devices, TFTP is an efficient way to boot the kernel during development so that the user does not have to flash a new kernel every time it is modified. It is also helpful when updating images in flash from U-Boot.

First, start by installing the TFTP server.

```
sudo apt-get install tftpd-hpa
```

Next, files can be accessed from another machine on the same network by simply using the IP address of the host. Specify a folder where the files will reside on the host by replacing the folder path for TFTP_DIRECTORY with whatever folder you wish to use as your TFTP file storage location, or leave the folder as the default.

```
sudo gedit /etc/default/tftpd-hpa

# /etc/default/tftpd-hpa

TFTP_USERNAME="tftp"
TFTP_DIRECTORY="/var/lib/tftpboot"
TFTP_ADDRESS="0.0.0.0:69"
TFTP_OPTIONS="--secure"
```

If you made any changes to the settings of the TFTP server, you need to restart it for them to take effect.

```
sudo restart tftpd-hpa
```

Lastly, if you would like to grant every user on the system permission to place files in the TFTP directory, use the following command, replacing <TFTP_DIRECTORY> with your chosen location.

```
sudo chmod ugo+rwx <TFTP_DIRECTORY>
```

3.1.2 NFS

A network file-system (NFS) server gives other systems the ability to mount a file-system stored on the host and exported over the network. In embedded development, this is essential for quickly testing applications and evaluating different file-system setups.

To begin the installation process use the following command:

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

Exported filesystems are designated in the "/etc/exports" file and allow you to choose both the directory to be exported and many settings for accessing the exports. Below is an example for exporting a folder called "nfs_export-ex" located in a user's home directory.

```
sudo gedit /etc/exports

# /etc/exports

/home/<user>/nfs_export-ex  *(rw,sync,no_root_squash,no_subtree_check)
```

The options (rw, sync, no_root_squash, no_subtree_check) for this folder are essential in setting up the NFS export correctly. For more information on additional options, refer to the man page for 'exports'.

- rw enables

read and write access when the directory is mounted

- sync

tells the file-system to handle local access calls before remote access

- no_root_squash

allows root access when mounting the file-system

- no_subtree_check

reduces the number of checks the server must make to ensure that an exported sub-directory is within an exported tree and also enables access to root files in conjunction with no_root_squash

After modifying this file, in order to mount the directories as an NFS, you must force the NFS server to export all of the directories listed in "/etc/exports".

```
sudo /usr/sbin/exportfs -va
```

3.1.3 Samba

Samba servers are an excellent way to access a Linux file-system on a Windows machine via a network connection. Using a Samba server, it is quick and easy to transfer files between systems. To install a Samba server, use the following command:

```
sudo apt-get install samba
```

Before the Samba share can be mounted on another machine it's necessary to modify the configuration file to allow write access and access to home directories. Start by editing the "/etc/samba/smb.conf" file.

```
sudo gedit /etc/samba/smb.conf
```

Inside this file there are four specific things that need to be uncommented (remove the ';' at the beginning of the line) to enable the sharing of home folders and write access. Below is the section that must be modified:

```
#======================= Share Definitions =======================

# Un-comment the following (and tweak the other settings below to suit)
# to enable the default home directory shares. This will share each
# user's home directory as \\server\username
;[homes]
;   comment = Home Directories
;   browseable = yes

# By default, the home directories are exported read-only. Change the
# next parameter to 'no' if you want to be able to write to them.
;   read only = no
```

The outcomes after the changes are made follow:

```
#======================= Share Definitions =======================

# Un-comment the following (and tweak the other settings below to suit)
# to enable the default home directory shares. This will share each
# user's home directory as \\server\username
[homes]
   comment = Home Directories
   browseable = yes

# By default, the home directories are exported read-only. Change the
# next parameter to 'no' if you want to be able to write to them.
   read only = no
```

NOTE: It might also be necessary to change the "workgroup = " line to match the workgroup for your machine.

To apply the changes, the next step is to restart all Samba-related processes.

```
sudo restart smbd
sudo restart nmbd
```

Lastly, each user needs to have a password enabled to be able to use the Samba server. There are no rules for this password. The simplest method for choosing this password is to make it the same as the UNIX user's password, but it is not a requirement. After typing in the command below, you will be prompted to enter the password for the specified user.

```
sudo smbpasswd -a <user>
```

As mentioned in the configuration file, the samba share can be connected by accessing "\\<host machine ip>\\<user>" by either mounting a network share or using Windows explorer to navigate to it.

3.2 PTXdist

3.2.1 General Information

PTXdist is a set of tools created by Pengutronix to help manage all of the BSP sources, from x-loader to the filesystem and its applications. It makes it easier for a user to manage everything for specific platforms and toolchains without manually repeating relatively complex commands every time it is necessary to build a new image. It is important to note the PTXdist version because more than one may be necessary for building the toolchain and BSP.

3.2.2 Extracting Sources

Visit the phyCORE-AM3517's BSP page for information on the current versions of BSP tools.

Potentially, two versions of PTXdist will need to be downloaded:

1. ptxdist-1.99.19.1 for installing/constructing the cross-compile toolchain (Optional if using a pre-existing environment)
2. ptxdist-2010.11.1 for building the BSP with the toolchain mentioned above.

The following instructions correspond to downloading PTXdist for building the BSP, *ptxdist-2010.11.1*. Therefore, to download the PTXdist software for building the toolchain, *ptxdist-1.99.19.1*, repeat the instructions replacing *2010.11.1* with *1.99.19.1*. In order to be built, extract the archive containing the PTXdist software:

```
tar -jxvf ptxdist-2010.11.1.tar.bz2
cd ptxdist-2010.11.1
```

Now that the source has been extracted, the next step is to configure it for building.

3.2.3 Pre-Requisites

PTXdist checks for specific packages that must be installed before it can be successfully built. From the PTXdist source directory, *ptxdist-2010.11.1*, execute the command to begin a script that uses GNU autotools to help set up the environment for building the distribution.

```
./configure
```

This command automatically stops if it is missing a package and states why and what package to install to continue with the initial setup. After successfully running the configure script, build and install PTXdist.

From the PTXdist source directory, *ptxdist-2010.11.1*:

```
make
sudo make install
```

The install location is "/usr/local" by default, which is why the **make install** command must be run with root privileges. If another directory is preferred for the install, use the --prefix option when installing, but be sure to add the "bin/" directory of the installed tools in this new folder to $PATH by modifying and sourcing ~/.bashrc.

```
source ~/.bashrc
```

Now that the install is complete, the PTXdist folder can be removed, as well as the original archive.

Optional:

```
cd ..
rm  -rf ptxdist-2010.11.1*
```

3.2.4 Using PTXdist

PTXdist is a console command tool and different functions are run by extending parameters to the **ptxdist** base command.

```
ptxdist <parameter>
```

To generate a full list of parameters and a description of the function, use the **help** command:

```
ptxdist help
```

Since PTXdist versions can be installed in parallel it may be useful to view the version number corresponding to the **ptxdist** command:

```
ptxdist --version
```

If the output of this command does not correlate to the desired version of PTXdist, affix the **ptxdist** command with the version number. For example, for the phyCORE-AM3517 BSP, affix all **ptxdist** commands with **2010.11.1** resulting in **ptxdist-2010.11.1**:

```
ptxdist-2010.11.1 <parameter>
```

Alternatively, to correlate the **ptxdist** command with a specific version create a symbolic link. For example, for **ptxdist** to correlate to **ptxdist-2010.11.1**:

```
sudo ln -fs /usr/local/lib/ptxdist-2010.11.1/bin/ptxdist /usr/local/bin/ptxdist
```

Potential Issues:

- Wrong PTXdist Version

It is important to use the correct version of PTXdist when building the Toolchain or BSP. If the incorrect version is used, the following will result:

error: The ptxconfig file version and ptxdist version do not match:configfile version: 2012.03.0ptxdist version: 2011.11.0

You can either migrate from an older ptxdist release with:'ptxdist migrate'

or, to ignore this error, add '--force'to ptxdist's parameters, e.g.:'ptxdist --force go'

The version of PTXdist used is noted as *ptxdist version* where the one required is specified as *configfile version*. Therefore, rerun the command with the correct version either appended to **ptxdist** or create a symbolic link. If the *merge* or *force* options were used on the Toolchain or BSP they will need to be removed and reinstalled to be built with the correct PTXdist version. Please note that the above is a generic example and may not apply directly to your BSP.

3.2.4.1 Setup

The first time PTXdist is used, there are some setup properties that have to be configured. To run PTXdist's setup, use the following command:

```
ptxdist setup
```

Once in the ptxdist setup, the only settings that should be modified are the *User->Name* and *User->eMail*. This is mainly for general logging purposes. If you wish to modify any other settings, please refer to the Getting Help section for a link to PTXdist documentation.

Since PTXdist works with sources only, it needs to grab source archives from the web using wget as it advances through its setup if they do not exist already. Therefore, an internet connection is required.

3.3 Toolchains

In order to build images or applications for an embedded device, it is necessary to have a cross toolchain that will perform the necessary operations to compile code for a specified processor and system setup.

Each toolchain will have a modified GNU Compiler Collection (gcc) designed for the desired setup. The phyCORE-AM3517 PD11.1.0 uses the arm-cortexa8 toolchain which can be built from the OSELAS.Toolchain-1.99.3.7 and ptxdist-1.99.19.1 sources.

3.3.1 Existing Toolchains

If a working toolchain is already installed for a given architecture, it is possible to use this instead of building an OSELAS Toolchain. Do note that since external toolchains have not been tested it is possible that they may not work properly across all environments.

An extra step needs to be taken if the plan is to use an external toolchain. By default, the software package will be set to check for the vendor toolchain that it was compiled with. In order to change this, use the following command inside the root directory of the BSP:

```
ptxdist platformconfig
    architecture --->
        toolchain --->
            () check for specific toolchain vendor
```

3.3.2 Building OSELAS Toolchains

An OSELAS toolchain, managed by PTXdist, can be easily built for a target architecture.

For the phyCORE-AM3517, the arm-cortexa8 architecture based toolchain is built from OSELAS.Toolchain-1.99.3.7 [Here] and PTXdist 1.99.19.1 [Here]. See Section 2.2 for information regarding the installation of PTXdist sources.

```
tar -jxvf OSELAS.Toolchain-1.99.3.7.tar.bz2
cd OSELAS.Toolchain-1.99.3.7
```

Be sure to use the correct ptxdist version for the toolchain by affixing all **ptxdist** commands with **1.99.19.1** resulting in **ptxdist-1.99.19.1** or see Section 2.2.4 for information on creating a symbolic link between **ptxdist** and **ptxdist-1.99.19.1**.

```
ptxdist select ptxconfigs/arm-cortexa8-linux-gnueabi_gcc-4.3.2_glibc-2.8_binutils-2.18_kernel-2.6.27-sanitized.
ptxconfig
ptxdist go
```

The toolchain is now built and installed in */opt/OSELAS.Toolchain-1.99.3.7/arm-cortexa8-linux-gnueabi* and ready to be used for building the BSP.

If you wish to build applications outside of the BSP directory, add the toolchain location to your PATH. Use the following from the command line or permanently add to your PATH by including it in .bashrc:

```
PATH=/opt/OSELAS.Toolchain-<toolchain version>/arm-<processor>-linux-gnueabi_gcc-linaro-<version>_glibc-
<version>_binutils-<version>_kernel-<version>-sanitized/bin/:$PATH
```

Following a successful build, the OSELAS.Toolchain-1.99.3.7 folder can be removed, as well as the original archive.

Optional:

```
cd ..
rm  -rf OSELAS.Toolchain-1.99.3.7*
```

3.3.2.1 Protecting Toolchains

It is recommended, although optional, to set the /opt/OSELAS.Toolchain-1.99.3.7/arm-cortexa8-linux-gnueabi directory and its contents as read-only to prevent accidental modifications to the toolchain.

4 Board Setup-phyCORE-AM3517

4.1 Overview

The phyCORE-AM3517 comes pre-flashed with x-load, u-boot, linux kernel, and jffs2 filesystem.

Note: If for any reason it is necessary to re-flash the example images, they are located on the PHYTEC America FTP [here]. Refer to Section 6 Flashing Images for information on flashing these images.

After the device is out of the box and setup, make sure that there is a DB9/DB25 cable attached between the host machine and UART3 on the AM3517 carrier board to enable RS-232 serial communication. In a terminal on the host, access minicom and set it up, allowing console access over the serial port.

```
/* If minicom is not installed */
sudo apt-get install minicom

minicom -c on -s
```

Navigate to "Serial port setup" in minicom and modify line "A - Serial Device : " to read */dev/ttyS0*.

Note: The serial device is dependent on what COM port you are connected to on your system, so /dev/ttySO is merely an example.

Next, modify "E - Bps/Par/Bits : " to have a speed of 115200 and 8-N-1 (8N1) for the stop bits. Return to the main menu of minicom and select "Save setup as dfl" to make this the default setup anytime minicom is loaded, meaning *minicom -c on* is all that needs to be done in the future for this machine to be able to communicate with the kit.

4.2 Basic Settings

The settings most necessary for operation at this point are environment variables in U-Boot following the initial setup of minicom (or any other serial monitor):

```
PCM048 # print
PCM048 # setenv ethaddr ##:##:##:##:##:##
PCM048 # setenv ipaddr ###.###.###.###
PCM048 # setenv serverip ###.###.###.###
PCM048 # setenv gatewayip ###.###.###.###
PCM048 # setenv netmask ###.###.###.###
PCM048 # setenv nfshost ###.###.###.###
PCM048 # setenv rootpath /<nfs_mount_location>
PCM048 # save
```

ethaddrMAC address for the device. If this is not already defined in the list of environment variables, it will be located on a small sticker on the SOM.

ipaddrA dedicated IP address for the SOM. This is crucial if TFTP will be used for updating the device's images at any point.

serveripIP address of the host or another machine where the TFTP directory, if it exists, is located.

gatewayipGateway IP for the network. This is only necessary if the TFTP directory is located on another network.

netmaskNetmask for the network: typically 255.255.255.0. This is only necessary if the TFTP directory is located on another network.

nfshost (required for NFS)IP address of the NFS host.

rootpath (required for NFS)Location of the path to the NFS directory on the host system. e.g. /home//nfs-export

Note: help is a useful tool in u-boot to show available commands and usage.

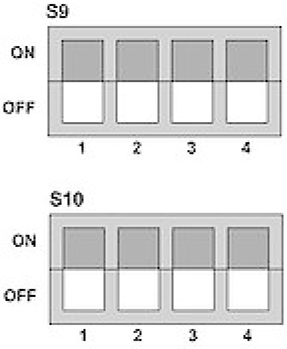
4.3 Selecting Boot Modes

The boot mode is selected using switches S9 and S10 on the carrier board. This selection identifies the location from where the X-Loader and U-Boot binaries are loaded for execution.

To boot (X-Loader and U-Boot) from NAND, use this switch setting:

S9-1 OFF S9-2 OFF S9-3 OFF S9-4 OFF

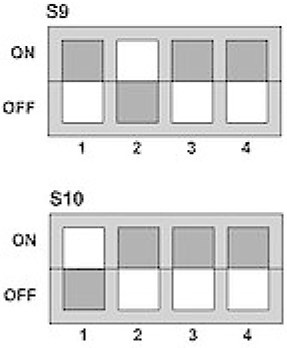
S10-1 OFF S10-2 OFF S10-3 OFF S10-4 OFF



To boot (X-Loader and U-Boot) from MMC, use this switch setting:

S9-1 OFF S9-2 ON S9-3 OFF S9-4 OFF

S10-1 ON S10-2 OFF S10-3 OFF S10-4 OFF



Note: Be careful when changing these switches as the S9 and S10 packages are rotated 180° off of each other so the switches will be on opposite sides for both OFF and ON and count 1-4 in different directions.

4.4 Stand-Alone Booting

By default, the kit comes setup to boot from the provided images loaded on the NAND flash. Executing the following commands in U-Boot tells the system to boot using the linux kernel and filesystem located in NAND.

```
PCM048 # setenv bootcmd 'run boot_nand'
PCM048 # save
```

4.5 Remote Booting

For development it may be beneficial to modify the boot settings to allow the kernel to be loaded from TFTP, and/or mount a network file-system (NFS) so that updates to the kernel can be tested quickly using custom applications hosted on the NFS.

Boot Linux Kernel via NAND with NFS

```
PCM048 # setenv bootcmd 'run boot_nand_nfs'
PCM048 # save
```

Boot Linux Kernel via TFTP with NFS

```
PCM048 # setenv bootcmd 'run boot_tftp_nfs'
PCM048 # save
```

Boot Linux Kernel via TFTP with Root Filesystem in NAND

```
PCM048 # setenv bootcmd 'run boot_tftp_nand'
PCM048 # save
```

4.6 Other Booting Options

There are additional options for booting from MMC that may be handy as well.

Boot Linux Kernel via MMC with NFS

```
PCM048 # setenv bootcmd 'mmc init; run boot_mmc_nfs'
PCM048 # save
```

Boot Linux Kernel via MMC with Root Filesystem in NAND

```
PCM048 # setenv bootcmd 'mmc init; run boot_mmc_nand'
PCM048 # save
```

Boot Linux Kernel via MMC with Root Filesystem on MMC

```
PCM048 # setenv bootcmd 'mmc init; run boot_mmc'
PCM048 # save
```

4.6.1 Display Settings

By default the system is set to support the 5" VGA LVDS TFT-LCD display with integrated touch and adapter board (part# LCD-014). Listed below are settings to modify the bootargs in U-Boot for alternate display options.

Use DVI

```
PCM048 # setenv disp '"dvi"'
PCM048 # setenv fbm 'dvi:1440x900MR@60'
PCM048 # save
```

Use LCD-014-M/LCD-017-050 (5" VGA LVDS TFT-LCD)

```
PCM048 # setenv disp '"lcd"'
PCM048 # setenv fbm 'lcd:640x480@60'
PCM048 # save
```

Use LCD-011 (3.5" QVGA TTL TFT-LCD)

```
PCM048 # setenv disp '"lcd"'
PCM048 # setenv fbm 'lcd:240x320'
PCM048 # save
```

It is also possible to change the display on-the-fly once Linux has been booted.

LCD -> DVI Linux Kernel Settings

```
root@phyCORE:~ echo 0 > /sys/devices/platform/omapdss/display0/enabled
root@phyCORE:~ echo "dvi" > /sys/devices/platform/omapdss/manager0/display
root@phyCORE:~ echo 1 > /sys/devices/platform/omapdss/display2/enabled
```

DVI -> LCD Linux Kernel Settings

```
root@phyCORE:~ echo 0 > /sys/devices/platform/omapdss/display1/enabled
root@phyCORE:~ echo "lcd" > /sys/devices/platform/omapdss/manager0/display
root@phyCORE:~ echo 1 > /sys/devices/platform/omapdss/display0/enabled
```

5 Building a BSP

5.1 Managing Configurations

PTXdist uses the KConfig files present throughout the BSP to allow users to easily configure individual settings and drivers.

NOTE: All target source code is located in the *OSELAS.BSP-Phytec-phyCORE-AM35xx/platform-phyCORE-AM35xx/build-target/* directory. For driver development and general settings or carrier board design it is necessary to know about the following three files to help integrate and modify features on the system:

- linux kernel source is build-target/linux-2.6.32

linux-2.6.32/arch/arm/mach-omap2/board-pcm048.c is the board config file

- u-boot source is build-target/u-boot-2009.11

u-boot-2009.11/include/configs/pcm048.h is the u-boot config file

- x-loader source is build-target/x-load-f243938

x-load-f243938/include/configs/pcm048.h is the x-loader config file

5.1.1 Platform

The platform config contains the default settings for each platform including what bootloaders, kernels, and filesystem images are to be built. The settings for the platform are modified using the following command:

```
ptxdist platformconfig
```

As a user, it is rare to ever modify these settings, but it may be useful to view them.

5.1.1.1 Enabling Graphics SDK

One of those rare modifications to the platformconfig is to enable the *omap35_graphics_sdk*. The image that is pre-installed on the NAND flash for the kit already has the Graphics SDK enabled, but each user must register on TI's website and download the Graphics SDK [here].

After the download is complete, copy the "Graphics_SDK_setuplinux_#_##_##_##.bin" file to the "OSELAS.BSP-Phytec-phyCORE-AM35xx-PD11.1.0 /local_src" directory.

To enable it to be built into the next image, execute the following:

```
ptxdist platformconfig
```

Following that, select the menu option *Graphics driver Menu* ---> and then select the *omap35x_graphics_sdk* option. Beneath that, if the current listed SDK version does not match the #_##_##_## pattern of your download, select and change it accordingly. Exit the configuration completely and select**Yes** when it asks if you would like to save your configuration.

Now, after a successful build, the graphics services will be enabled and automatically load when this image is booted.

5.1.2 Kernel

Use the kernel config to modify what drivers and support are included in a linux kernel build.

```
ptxdist kernelconfig
```

5.1.3 Root Filesystem

In the overall menuconfig it is easy to modify the applications and content that is built into the filesystem by toggling options in the base configuration. This allows both minimal and more complete filesystem builds to be created easily.

```
ptxdist menuconfig
```

5.2 Building Images with PTXdist

Building images with PTXdist is very simple. The command *ptxdist go* does all of the required steps to compile all of the sources and make changes only where necessary. *ptxdist images* builds the filesystem images.

```
ptxdist go
ptxdist images
```

All images are then stored in OSELAS.BSP-Phytec-phyCORE-AM35xx/platform-phyCORE-AM35xx/images.

6 Flashing Images

Flashing images can be relatively complex and there are two main ways to accomplish this: TFTP and MMC. If you wish to flash an image with a different name than what is designated in *italic* below, set the respective U-Boot environment variable to reflect that change in name:

- xload=x-load.bin.ift
- uboot=u-boot.bin
- bootfile=linuximage
- filesys=root.jffs2

Note: If for any reason it is necessary to re-flash the example images, they are located on the PHYTEC America FTP [here].

6.1 X-Loader

Method: TFTP. A valid *x-load.bin.ift* image must be on the TFTP server

```
PCM048 # run update_xload
```

Method: MMC. A valid *MLO* image must be on the SD/MMC card

```
PCM048 # mw.b 0x81600000 0xff 0x20000
PCM048 # mmc init
PCM048 # fatload mmc 0 ${loadaddr} MLO
PCM048 # nand erase 0 20000
PCM048 # nandecc hw
PCM048 # nand write.i ${loadaddr} 0 20000
```

6.2 U-Boot

Method: TFTP. A valid *u-boot.bin* image must be on the TFTP server

```
PCM048 # run update_uboot
```

Method: MMC. A valid *u-boot.bin* image must be on the SD/MMC card

```
PCM048 # mw.b 0x81600000 0xff 0x160000
PCM048 # mmc init
PCM048 # fatload mmc 0 ${loadaddr} ${uboot}
PCM048 # nand erase 80000 160000
PCM048 # nandecc sw
PCM048 # nand write.i ${loadaddr} 80000 160000
```

6.2.1 Restore default U-Boot settings

If the U-Boot environment variables need to be restored for any reason, simply delete the parameter save location in NAND, and the defaults will be restored with the next boot.

```
PCM048 # nand erase 240000 40000
```

6.3 Kernel

Method: TFTP. A valid *linuximage* must be on the TFTP server

```
PCM048 # run update_kernel
```

Method: MMC. A valid *linuximage* must be on the SD/MMC card

```
PCM048 # mw.b 0x81600000 0xff 0x300000
PCM048 # mmc init
PCM048 # fatload mmc 0 ${loadaddr} ${bootfile}
PCM048 # nand erase 280000 300000
PCM048 # nandecc sw
PCM048 # nand write.i ${loadaddr} 280000 300000
```

6.4 Root Filesystem

Method: TFTP. A valid *root.jffs2* image must be on the TFTP server

```
PCM048 # run update_rootfs
```

Method: MMC. A valid *root.jffs2* must be on the SD/MMC card

```
PCM048 # mw.b 0x81600000 0xff 0x6000000
PCM048 # mmc init
PCM048 # fatload mmc 0 ${loadaddr} ${filesys}
PCM048 # nand erase 780000 6000000
PCM048 # nandecc sw
PCM048 # nand write.i ${loadaddr} 780000 6000000
```