VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

T-MOBILE MDA II V LINUXU

T-MOBILE MDA II IN LINUX

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — Bc. ZBYNĚK MICHL
AUTHOR

VEDOUCÍ PRÁCE — Ing. TOMÁŠ KAŠPÁREK
SUPERVISOR

BRNO 2009

Abstrakt

Diplomová práce se zabývá mobilním digitálním asistentem T-Mobile MDA II v operačním systému Linux. První část práce je zaměřena na identifikaci zařízení a specifikaci parametrů MDA II. Druhá část popisuje výběr GNU distribuce s Linuxovým zavaděčem a porovnává současný stav podpory MDA II v různých verzích Linuxového jádra. Poslední část práce se věnuje implementaci kódu některých komponent a jeho začlenění do Linuxového jádra.

Klíčová slova

Ångström, ARM, ASIC3, bootldr, distribuce, Familiar, GPIO, Handhelds, HaRET, Himalaya, HTC, jádro, LinExec, Linux, MDA, mobilní, modul, ovladač, Pocket PC, PXA263, T-Mobile, vanilla, Xanadux, zařízení, zavaděč

Abstract

MSc. thesis deals with mobile digital assistant T-Mobile MDA II running Linux operating system. The first part presents device identification and parameters' specification of the MDA II. The second part focuses on selection of GNU distribution with Linux bootloader and Linux kernel support comparison. The subject of the last part is MDA II component code implementation and its merging into Linux kernel.

Keywords

Ångström, ARM, ASIC3, bootldr, bootloader, device, distribution, driver, Familiar, GPIO, Handhelds, HaRET, Himalaya, HTC, kernel, LinExec, Linux, MDA, mobile, module, Pocket PC, PXA263, T-Mobile, vanilla, Xanadux

Citace

Zbyněk Michl: T-Mobile MDA II in Linux, diplomová práce, Brno, FIT VUT v Brně, 2009

T-Mobile MDA II in Linux

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Tomáše Kašpárka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

....
Zbyněk Michl
January 21, 2009

Contents

Chapter 1

Prologue

A mobile device is a pocket-sized computing device, typically comprising of a small visual display screen for user output and a miniature keyboard or touchscreen for user input. There are many of different variants of mobile devices on the market using various software equipment.

This MSc. thesis focuses on mobile digital assistant *T-Mobile MDA II*. MDA is a device which combines a personal digital assistant with a mobile phone and offers all standard operations like an office computer for an user. Furthermore the device is able to communicate with a personal computer.

The most widespread mobile digital assistants on the market run Microsoft Windows CE family operating system, like Pocket PC or Windows Mobile. There are also a few models with another operating system like Symbian, iPhone OS or various forms of GNU/Linux.

We intend to run GNU/Linux on T-Mobile MDA II which originally comes with Microsoft Pocket PC 2003 operating system. MDA II has many brand names and is supplied by miscellaneous mobile operators, however with Microsoft operating system in all cases and no other OS is officially supported.

As mentioned above, there is no official GNU/Linux distribution for T-Mobile MDA II, but despite it we could find many projects which make an effort to create minimalistic distributions specially for embedded and mobile devices including mobile digital assistants.

Another issue could be device hardware support by the Linux kernel. Mobile devices from miscellaneous suppliers have various hardware modifications and furthermore many manufacturers do not provide their hardware specifications publicly. As a result, only some mobile device drivers are fully implemented in the mainline Linux kernel. Fortunately, we can find a few projects with their own kernel branches which put effort into implementing drivers that are not in the mainline kernel.

Brief summary of chapters follow:

Chapter 2 — presents identification and specification of the T-Mobile MDA II, procedures of resetting device and connecting to MDA II from a personal computer. Default bootloader usage is mentioned as well.

Chapter 3 — focuses on support of MDA II in GNU/Linux system. It includes GNU distribution selection (Familiar GNU/Linux and Ångström GNU/Linux) with Linux kernel (2.6.3-rmk0-hh0-xda0, 2.6.12-hh2-xda0-ba0, 2.6.21-devel, 2.6.28-devel) and Linux kernel bootloader (bootldr, LinExec, HaRET). For distributions and kernels the text describes how to get sources from repository and compile them. What follows is comparison of Linux kernel features for each Linux kernel and description of hardware parts of MDA II currently supported and not supported. The end of the chapter focuses on running above kernels with Familiar and Ångström distributions on the T-Mobile MDA II device.

Chapter 4 — describes MDA II components' code implementation. The first part provides information about general-purpose input/output which is needed for device components' control. The second part presents kernel GPIO's macros and functions used for future driver implementation. Third part implements components' code in the 2.6.21-devel branch (USB communication, power and battery management, touchscreen and buttons) and in the 2.6.28-devel branch (MDA II hardware definitions, LCD display). The fourth part summarizes implementation result with improvements description. The last, fifth, part tells the way how to create and submit Linux kernel patches.

Chapter 2

T-Mobile MDA II Description

2.1 Identification of MDA II

Real name of the T-Mobile MDA II is *HTC Himalaya* [1] which is manufactured by *High Tech Computer Corporation* from Taiwan. This device can be sold under different names depending on a supplier or a seller. In addition to T-Mobile MDA II, HTC Himalaya is sold as well as the Qtek 2020/2060, I-Mate Pocket PC Phone Edition, O2 XDA II, Krome Navigator F1, Orange SPV M1000, Vodafone WirelessPDA, Dopod 696/696i or Movistar TSM500 [2]. All of these types have the same hardware, but there are little differences in software.

Figure 2.1: T-Mobile MDA II from the front and back side [2]

2.2 MDA II Specification

Table 2.1 summarise brief specification of the HTC Himalaya device used for development. Detailed specification [3] can be found in appendix A.

Resetting of the HTC Himalaya:

Soft reset — reset the unit by pressing the *reset* button.

Hard reset — press *power button* and *reset* the unit simultaneously.

Note that after hard reset all user's data and settings are lost and default factory settings are loaded.

Item	Specification
Operating System	Microsoft Pocket PC 2003 Phone Edition
Main Applications	Pocket Word, Pocket Excel, xBackup, MMS Composer, ActiveSync, Album, Calculator, ClearVUE PDF, ClearVUE PPT, Microsoft Reader, MSN Messenger, Terminal Services Client
Keyboard	No external keyboard attached
Processor	Intel XScale PXA263 400 MHz
Memory	128 MB RAM, 64 MB ROM
Display	240 × 320 TFT screen, 64k colours
Band	Tri-band (900/1800/1900 MHz)
Wi-Fi	Not available, but Wi-Fi unit can be added
Bluetooth	Version 1
Battery	1200 mAh Li-Pol
Built-in Camera	CMOS sensor, 0.3 Mpixel
Audio	MP3, WAV and WMA
Video	3GP and MPEG
I/O Interface	USB, serial, IrDA
Expansion	SD/MMC card

Table 2.1: Brief specification of the HTC Himalaya

2.3 Connecting to MDA II

Although there are several ways, how to connect HTC Himalaya with a desktop computer, it is necessary to use RS232 or USB cable for service purposes. It is because other connection types like IrDA or bluetooth are not supported by all service utilities. Besides RS232 cable is the only way to capture output written by a kernel while the Linux kernel is booting. It can be very useful for kernel debugging in the case of making some changes in the kernel.

Himalaya has one connector at the bottom with 22 contacts and USB port, serial port COM1, power and audio (in and out) are available there. Picture 2.2 shows orientation of the connector. Pins are connected [4] according to table 2.2.

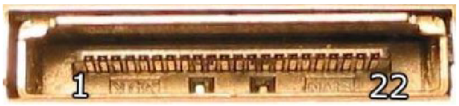

Figure 2.2: Connector of the HTC Himalaya [4]

If we have not got original RS232 cable, USB cable or USB cradle (all of these types are standard accessories), we can use home-made cable connected according to table mentioned above.

2.4 Default MDA II Bootloader

The bootloader is invoked [5] by pressing down both the *power* and the *navigator/action* button, while pressing the *reset* button.

Pin	Signal	Pin	Signal
1	CAR_MIC_IN	12	IN_CALL
2	GND/SENSE	13	USB_VDD
3	CAR_AUDIO_R	14	UDC+
4	CAR_AUDIO_L	15	UDC−
5	ANALOG GND	16	GND
6	RS232_DCD	17	GND
7	RS232_CTS	18	GND
8	RS232_TXD	19	CAR_ON#
9	RS232_RTS	20	V_ADP
10	RS232_RXD	21	V_ADP
11	RS232_DTR	22	V_ADP

Table 2.2: Pins description of the HTC Himalaya

We can connect to the bootloader either via a RS232 or via USB (see section 2.3). The first mentioned variant needs only a terminal program like a *hyperterm* (MS Windows[1]) or *minicom*[2] (GNU/Linux). For USB connection we can use *mtty*[3] (MS Windows) and for GNU/Linux the procedure goes as follows:

1. Compile[4] kernel modules *usbcore*, *usbserial* and *ipaq*,
2. Load these modules into a kernel,
3. Create `/dev/ttyUSB0` using `mknod /dev/ttyUSB0 c 188 0` if it does not exists,
4. Connect via USB cable/cradle,
5. Use *minicom* or other terminal program.

In the case of successful connection you can see the following command line:

```
For a help screen, use command ? or h
USB>
```

however both commands *?* and *h* do not work properly[5].

Now we can do many operations, e.g. copy ROM to SD-card, examine memory, write to RAM, etc.

If you want to print help for a command, just enter a command name without any arguments and the bootloader will show all possible options. Complete list of commands with their usage can be find at [6].

[1]Make sure you have disabled the RS232 or USB connection in MS ActiveSync, before trying to connect to the bootloader.

[2]http://alioth.debian.org/projects/minicom

[3]http://wiki.xda-developers.com/uploads/mtty.exe

[4]If all of these modules are already compiled, skip this step.

[5]Probably bug in the bootloader.

Chapter 3

GNU/Linux Support for MDA II

Porting GNU/Linux to HTC phones is provided by the *Xanadux*[1] project.

Project goal is to produce working and usable Linux on the device. This includes getting a working base system, making sure all the features of the device (such as the phone, GPRS, etc.) can be used, and creating a distribution that everybody can actually install and use.

Xanadux team have chosen already existing *Familiar*[2] distribution (see subsection 3.1.1) and they are going to make patches for Familiar's kernel to add a support for HTC phones (including HTC Himalaya). However Familiar seems to be dead nowadays[3], but *Ångström* distribution (see subsection 3.1.2) could be a convenient replacement for many PDA machines.

3.1 GNU/Linux Distributions

3.1.1 Familiar GNU/Linux

The Familiar project is composed of a group of loosely knit developers (*Handhelds*[4] team) all contributing to creating the next generation of PDA OS. Currently, most of development time is being put towards producing stable, and full featured Linux distribution for HP iPAQ series and other handheld computers, as well as applications to run on top of the distribution.

Currently Familiar's Linux distribution supports some of the following key features [10]:

- Choice of user environments, both with a full personal information management (PIM) suite and other applications
  - GPE Palmtop Environment (GPE),
  - Open Palmtop Integrated Environment (Opie),
- Full package management based on *ipkg*,
- Many system programs are implemented using *BusyBox*, saving space,
- *Dropbear* SSH server included by default,

[1] http://wiki.xda-developers.com/index.php?pagename=Xanadux
[2] http://familiar.handhelds.org
[3] The status is current as of 1 November 2008.
[4] http://handhelds.org/moin/moin.cgi/HandheldsPeople

- Built entirely using the *OpenEmbedded* build system.

The current stable release versions are:

- Familiar 0.8.4 for the iPAQ h3600, h3700, h3800, h3900, h5400, h5500 series, and Siemens Simpad,
- Familiar 0.7.2 for the iPAQ h3100 series.

3.1.2 Ångström GNU/Linux

Another way is to use *Ångström*[5] distribution. Ångström was started by a small group of people who worked on OpenEmbedded, OpenZaurus and OpenSimpad projects to unify their effort to make stable and user-friendly distribution for embedded devices like handhelds, set top boxes and network-attached storage devices and more.

Ångström will be available [11] for the at least Sharp Zaurus (SL-5500, SL-5600, SL-6000, SL-C7x0, SL-C1000, SL-C3xxx), Hewlett Packard iPAQ (h2200, h4000, hx4700, h5000), Nokia 770 internet tablet, HTC universal, Motorola A780 and Psion Teklogix NetBook Pro.

Despite the list above, distribution could be experimentally built for many other devices mentioned in the OpenEmbedded Device List[6] including HTC Himalaya.

EABI and OABI Interfaces

An *Embedded-Application Binary Interface* (EABI) specifies standard conventions[7] for file formats, data types, register usage, stack frame organization, and function parameter passing of an embedded software program. It is the next generation of the *Old-Application Binary Interface* (OABI) and its advantage is that it runs applications more effectively.

Ångström sources could be compiled either with EABI or OABI interface, but only 2.6.16 and newer kernels support EABI interface. That is why we have to choose OABI because we want to use both pre-2.6.16 and post-2.6.16 kernels (see subsection 3.3.3).

Getting Sources and Compilation

Following instructions [12] will tell you how to setup a build environment, how to get stable Ångström sources and compile them to create binary root image.[8]

All Ångström images are built using *OpenEmbedded*[9] cross-compile environment with *BitBake*[10] task executing tool and *Monotone*[11] version control system.

Note that you do not have to be *root* to build Ångström with OpenEmbedded. It is even recommended to always work as a user, not as *root*.

Let assume `$HOME/OE` as a default work directory.

Get BitBake and Monotone tools:

[5] http://www.angstrom-distribution.org
[6] http://www.linuxtogo.org/gowiki/OeDeviceList
[7] http://www.arm.com/products/DevTools/ABI.html
[8] The status is current as of 14 October 2008.
[9] http://www.openembedded.org
[10] http://bitbake.berlios.de
[11] http://www.monotone.ca

```
$ wget http://download.berlios.de/bitbake/bitbake-1.8.10.tar.gz
$ tar zvxf bitbake-1.8.10.tar.gz
$ wget http://www.monotone.ca/downloads/0.41/mtn-0.41-linux-x86.bz2
$ bunzip2 mtn-0.41-linux-x86.bz2
$ ln -s mtn-0.41-linux-x86 mtn
```

Get OpenEmbedded (OE) snapshot:

```
$ wget http://wiki.openembedded.net/snapshots/OE.mtn.bz2
$ bunzip2 OE.mtn.bz2
```

Upgrade OE database version:[12]

```
$ ./mtn --db=OE.mtn db migrate
```

Extract and update OE stable branch metadata (use the latest stable revision):

```
$ ./mtn --db=OE.mtn pull monotone.openembedded.org org.openembedded.stable
$ ./mtn --db=OE.mtn --revision=0fa016c65abfeab8646d405ea9db417c1c179b26 \
  checkout --branch=org.openembedded.stable
$ cd org.openembedded.stable
$ ../mtn pull monotone.openembedded.org org.openembedded.stable
$ ../mtn update
```

Setup build environment:

```
$ BBPATH=$HOME/OE:$HOME/OE/build:$HOME/OE/org.openembedded.stable
$ PKGDIR=$HOME/OE/build
$ DL_DIR=$HOME/OE/downloads
$ if [ -z ${ORG_PATH} ] ; then ORG_PATH=${PATH}; export ORG_PATH; fi
$ if [ -z ${ORG_LD_LIBRARY_PATH} ] ; then ORG_LD_LIBRARY_PATH=\
  ${LD_LIBRARY_PATH}; export ORG_LD_LIBRARY_PATH; fi
$ PATH=$HOME/OE/bitbake-1.8.10/bin:${ORG_PATH}
$ LD_LIBRARY_PATH=
$ export PATH LD_LIBRARY_PATH BBPATH
$ export LANG=C
$ export BB_ENV_EXTRAWHITE="MACHINE DISTRO ANGSTROM_MODE"
```

Create local configuration (see `$HOME/OE/org.openembedded.stable/conf/local.conf.sample` file). Put the following in `$HOME/OE/build/conf/local.conf`:

```
# Where to store sources
DL_DIR = "${HOME}/OE/downloads"
# Which files do we want to parse
BBFILES := "${HOME}/OE/org.openembedded.stable/packages/*/*.bb"
BBMASK = ""
# ccache always overfill $HOME....
CCACHE = ""
# What kind of images do we want?
```

[12] The database format has changed through monotone development. OE still uses old version, therefore we need to upgrade the db to be able to get it compatible with monotone release 0.41.

```
IMAGE_FSTYPES = "tar.gz"
# Set Angstrom distribution (version 2007-12.r18)
DISTRO = "angstrom-2007.1"
# Set TMPDIR instead of defaulting it to $pwd/tmp
TMPDIR = "${HOME}/OE/${DISTRO}-stable/"
# Set HTC Himalaya device
MACHINE = "htchimalaya"
```

Change *TARGET_OS* variable value to *linux* (to compile sources with OABI):

```
$ vi conf/distro/include/angstrom-glibc.inc
```

Remove *openmoko-dialer2* from *ANGSTROM_EXTRA_INSTALL* variable (otherwise distribution build fails):

```
$ vi packages/images/x11-image.bb
```

Start building:[13]

```
$ bitbake x11-image
```

Other useful packages not included in *x11-image* could be built optionally:[14]

```
$ bitbake bash && bitbake mc && bitbake less && bitbake nano
```

Final *.tar.gz* image of root filesystem will be saved in `$HOME/OE/angstrom-stable/deploy/glibc/images/htchimalaya/` directory in the case of sucessful compilation. The software as separate packages including optionally built ones will be found in `$HOME/OE/angstrom-stable/deploy/glibc/ipk/` directory. How to use the built image will be discussed in subsection 3.4.2.

3.2 Linux Bootloaders

bootldr — is one of the best-known Linux PocketPC bootloaders. It supports HP PDAs iPAQ h36xx, h37xx, h38xx, h39xx and h5xxx series, but HTC Himalaya is supported only partially at this time.

LinExec — bootloader which was initially developed to allow booting Linux kernel from MS Windows CE on a PDA with XScale processor.

HaRET — stands for Handheld Reverse Engineering Tool. It is a Linux bootloader which works from MS Windows CE environment (like a *Loadlin* for DOS or older *LinExec* tool above). It is also a tool for accessing the hardware internals of a Windows CE handheld to help get Linux up and running on it.

Warm-boot is the only way how to launch Linux on HTC Himalaya device at this time (e.g. using *HaRET* or *LinExec*). More information about using HaRET could be found at [13].

[13] *console-image* can be built as well, but *x11-image* is more complex. Note that *x11-image* build process needs a lot of free disk space (20 GiB should be enough).

[14] Use *ipkg* tool to install them ones the Ångström has booted up.

3.3 Linux Kernel

In HTC Himalaya kernel development following branches are used:

- *vanilla* — mainline released at ftp://ftp.kernel.org,
- *rmk* — patched by Russell King (the core ARM Linux maintainer),
- *hh* — patched by Handhelds team (team ports Linux to PDAs and creates Familiar distribution),
- *xda* — patched by Xanadux team (they are using *xda* as a common name for all HTC's phones),
- *ba* — stands for BlueAngel (name of HTC's phone),
- *un* — stands for Universal (name of HTC's phone),
- *ma* — stands for Magician (name of HTC's phone).

Figure 3.1 shows connections between significant branches. The base is created by *vanilla* kernel with *rmk* patches applied. Most of patches from *rmk* branch are merged into *vanilla* branch later. *hh* branch is derived from *rmk*, but there is almost no effort to merge changes back into *vanilla* kernel hence *hh* branch is considered as a base kernel for many handheld devices. Extended *xda* branch contains generic patches for all HTC phones. Finally we can focus our attention on *ba* branch, where HTC BlueAngel and HTC Himalaya specific patches are developed. Both devices are very similar thus development is done in a single branch. There is some effort to port changes from *ba* branch to *hh* branch, but most effort is put into HTC BlueAngel phone's specific parts and HTC Himalaya falls behind.

Note that as of 2.6.0 based *rmk* branch, sufficient patches have been merged into mainline to make the release of *rmk* kernels unnecessary. Therefore, there are no further *rmk* patch releases as could be seen in figure 3.2.

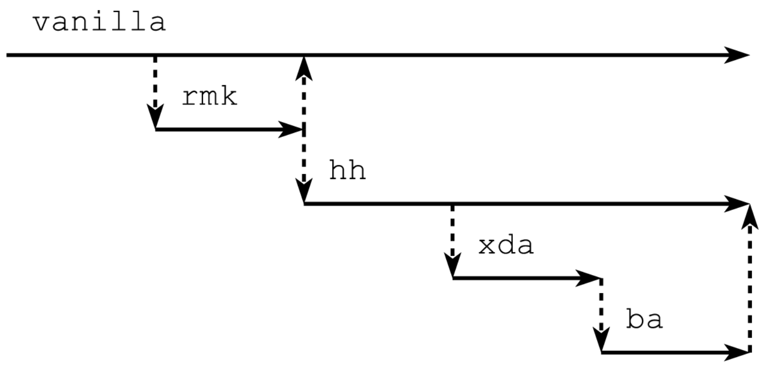

Figure 3.1: Development branches prior to 2.6.0 kernel

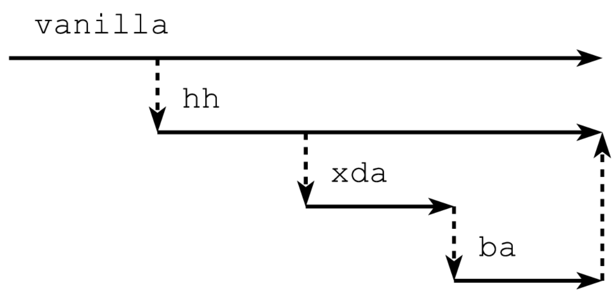

Figure 3.2: Development branches since 2.6.0 kernel

3.3.1 Kernel Sources

Xanadux Repository

All work of Xanadux team is located in the *SourceForge CVS* under the project name *xanadux* [7]. In this CVS they keep the Linux kernels they are working on and any tools that are specific to their development.

The CVS can be used in two ways:

- via web interface at `http://xanadux.cvs.sourceforge.net/xanadux/` to browse the trees, or
- with client tools like *cvs* we can checkout the sources directly to our local disk.

We can find there the following CVS modules:

```
acx100/          haret/             qomunicator/
bluez-utils/     linexec/           qpegps/
bootldr/         linux-2.4.19-xda/  roadmap/
cyace-arm/       linux-2.6-xda/     roadmap_editor/
gomunicator/     oe/
gpsd-2.19/       opie2/
```

As we can see above, currently[15] two Linux kernel trees are checked in the CVS. The *linux-2.4.19-xda* is currently used for HTC Wallaby[16] development, while the *linux-2.6-xda* is used for other HTC phones including HTC Himalaya.

There are following, among others, branches available:

```
K2-6-12-hh2-xda0-ma1   K2-6-3-hh2
K2-6-12-hh2-xda0-ma0   K2-6-3-rmk0-hh0-xda0
K2-6-12-hh2-xda0-un0   K2-6-3-hh0
K2-6-12-hh2-xda0-ba0   K2-6-2-hh0
K2-6-12-hh2            K2-6-1-rmk0-hh0-xda0
K2-6-3-hh2-xda0        K2-6-1-rmk0-hh0
```

How to get *2.6.3-rmk0-hh0-xda0* and *2.6.12-hh2-xda0-ba0* sources and compile them is described in subsection 3.3.2.

[15] The status is current as of 3 January 2008.

[16] Wallaby is name of another HTC's phone (Himalaya predecessor).

Handhelds Repository

Handhelds team has their own CVS repository for source code accessible via web interface at http://handhelds.org/cgi-bin/cvsweb.cgi/linux/ or with *cvs* command line tool.

There are the following, among others, branches available:

HEAD	K2-6-12-hh3	K2-6-3-hh2
K2-6-21-hh20	K2-6-12-hh2	K2-6-3-hh0

How to get *HEAD* sources (currently[17] based on 2.6.21-hh20 release) and compile them is described in subsection 3.3.2. Sources are marked as *2.6.21-devel*.

ARM Repository

This repository concentrates on ARM platform patches to be submitted into *vanilla* kernel. Repository is maintained by Russell King and could be find via web interface at http://ftp.arm.linux.org.uk/pub/linux/arm/kernel/git-cur/linux-2.6-arm.git or with *git* command line tool.

In GIT repository following branches are available:

devel	for-linus	master	origin
dma	machtypes	omap-clks1	

How to get *devel* sources (currenty[18] based on 2.6.28 release) and compile them is described in subsection 3.3.2. Sources are marked as *2.6.28-devel*.

3.3.2 Kernel Compilation

Here we present how to compile kernels *2.6.3-rmk0-hh0-xda0*, *2.6.12-hh2-xda0-ba0* [8], *2.6.21-devel* and *2.6.28-devel*. More about these kernels and their features could be seen in subsection 3.3.3.

Let assume `$HOME/kernel` as a default work directory.

Log on to Xanadux and Handhelds CVS servers:

```
$ cvs -d:pserver:anonymous@xanadux.cvs.sourceforge.net:/cvsroot/xanadux \
  login
CVS password:
$ cvs -d:pserver:anoncvs@anoncvs.handhelds.org:/cvs login
CVS password: anoncvs
```

2.6.3-rmk0-hh0-xda0

Get kernel sources from CVS:

```
$ cvs -z3 -d:pserver:anonymous@xanadux.cvs.sourceforge.net:/cvsroot/\
  xanadux co -ko -r K2-6-3-rmk0-hh0-xda0 -d linux-2.6.3-rmk0-hh0-xda0 \
  -P linux-2.6-xda
```

Get 3.3.2 GNU toolchain needed for cross-compilation:

[17] The status is current as of 2 September 2008.
[18] The status is current as of 16 December 2008.

```
$ wget http://www.handhelds.org/download/projects/toolchain/\
  arm-linux-gcc-3.3.2.tar.bz2
$ tar jvxf arm-linux-gcc-3.3.2.tar.bz2
$ export PATH=$HOME/kernel/usr/local/arm/3.3.2/bin:$PATH
```

Generate *.config* file and compile sources:

```
$ cd linux-2.6.3-rmk0-hh0-xda0
$ make himalaya_defconfig && make menuconfig
$ make
```

2.6.12-hh2-xda0-ba0

Get kernel sources from CVS:

```
$ cvs -z3 -d:pserver:anonymous@xanadux.cvs.sourceforge.net:/cvsroot/\
  xanadux co -ko -r K2-6-12-hh2-xda0-ba0 -d linux-2.6.12-hh2-xda0-ba0 \
  -P linux-2.6-xda
```

Get 3.4.1 GNU toolchain needed for cross-compilation:

```
$ wget http://www.handhelds.org/download/projects/toolchain/\
  arm-linux-gcc-3.4.1.tar.bz2
$ tar jvxf arm-linux-gcc-3.4.1.tar.bz2
$ export PATH=$HOME/kernel/usr/local/arm/3.4.1/bin:$PATH
```

Generate *.config* file and compile sources:

```
$ cd linux-2.6.12-hh2-xda0-ba0
$ make himalaya_defconfig && make menuconfig
$ make
```

2.6.21-devel

Get kernel sources from CVS:

```
$ cvs -z3 -d:pserver:anoncvs@anoncvs.handhelds.org:/cvs co -ko \
  -r HEAD -d linux-2.6.21-devel -P linux/kernel26
```

Get 4.2.4 GNU toolchain needed for cross-compilation:

```
$ wget http://nihilisme.ca:8080/arm/files/\
  Generic-arm_gcc-4.2.4-glibc-2.3.3.tar.bz2
$ tar jvxf Generic-arm_gcc-4.2.4-glibc-2.3.3.tar.bz2
$ export PATH=$HOME/kernel/usr/local/arm/gcc-4.2.4-glibc-2.3.3/\
  arm-unknown-linux-gnu/bin:$PATH
```

Edit *Makefile* to adapt cross-compiler:

```
$ cd linux-2.6.21-devel
$ sed -i 's/arm-linux-/arm-unknown-linux-gnu-/' Makefile
```

Generate *.config* file and compile sources:

```
$ make htchimalaya_defconfig && make menuconfig
$ make
```

2.6.28-devel

Get kernel sources from GIT:

```
$ git clone ftp://ftp.arm.linux.org.uk/pub/linux/arm/kernel/git-cur/\
  linux-2.6-arm.git linux-2.6.28-devel
$ cd linux-2.6.28-devel
$ git checkout -t -b devel origin/devel
```

Get 4.2.4 GNU toolchain if has not already done it (see 2.6.21-devel above).

Edit *Makefile* to adapt cross-compiler:

```
$ sed -i 's/arm-linux-/arm-unknown-linux-gnu-/' Makefile
```

It is not possible to compile sources for HTC Himalaya until device specific code is implemented (see section 4.4).

3.3.3 Kernel Features Comparison

Table 3.1 summarizes current status of HTC Himalaya code implementation across various Linux 2.6 kernels.

2.6.3-rmk0-hh0-xda0 has been the first 2.6 kernel where Himalaya specific code is implemented. However only a few components are supported in this branch. This branch could be found in Xanadux CVS (see subsection 3.3.1).

2.6.12-hh2-xda0-ba0 has the best support for basic components what makes device quite usable with this kernel. On the other hand phone part of the MDA device is still unimplemented. This branch also could be found in Xanadux CVS (see subsection 3.3.1). It seems that Xanadux branches are not being actively developed anymore.

2.6.21-devel is the latest[19] *Handhelds* version (based on *2.6.21-hh20* release) that is being actively developed. Some components of the MDA are supported, but not everything has been ported from *2.6.12-hh2-xda0-ba0* kernel yet. This branch could be located in Handhelds CVS (see subsection 3.3.1).

2.6.28-devel is the current[20] *vanilla* ARM development branch (based on *2.6.28* release). Booting this kernel on MDA device might be possible, however there is no way to display kernel output messages. That makes hardware component support status unidentified. Branch could be found in ARM GIT repository (see subsection 3.3.1).

Some of the Himalaya components such as GSM/GPRS, camera or bluetooth are still not implemented in any Linux kernel branch.

Kernel Feature Notes

Here are some useful extended notes about component research and their current support in Linux 2.6 kernels.

For example *(1, 3)* stands for row 1 and column 3 of table 3.1, thus *Booting on 2.6.21-devel* kernel.

(1, 1), *(1, 2)*, *(1, 3)*
Only warm-boot (e.g. by HaRET) is working.

[19] The status is current as of 30 October 2008.
[20] The status is current as of 26 December 2008.
[21] *rmk* patches not included in *2.6.3 vanilla* were probably ported from older *2.6.0-rmk* branch

		1	2	3	4
	Component	**2.6.3 -rmk0-hh0-xda0**[21]	**2.6.12 -hh2-xda0-ba0**	**2.6.21 -devel**	**2.6.28 -devel**
1	Booting	Part+	Part+	Part+	?
2	HW Buttons	No	Yes	No	?
3	Touchscreen	Yes	Yes	Part−	?
4	Serial Ports	Yes	Yes	Yes	?
5	USB	Yes	Yes	Part−	?
6	SD/SDIO Card	Part−	Yes	Yes	?
7	DiskOnChip	No	Part−	Part−	?
8	Strata Flash	No	Yes	Part−	?
9	GSM/GPRS	No	No	No	?
10	LCD	Part+	Yes	Part+	Part−
11	Sound	Part−	Part−	Part−	?
12	Battery	No	Part+	Part−	?
13	LEDs	Part−	Part+	No	?
14	Camera	No	No	No	?
15	Bluetooth	No	No	No	?

Legend:	No	unimplemented
	Part−	partially implemented (component is unusable)
	Part+	partially implemented (component is usable)
	Yes	fully implemented
	?	unidentified

Table 3.1: Features comparison across various Linux 2.6 kernels

(3, 1)
Module name: `himalaya_ts`, `evdev`, `tsdev`
Device path: `/dev/input/event0 (13, 64)`
Environment variable: `TSLIB_TSDEVICE=/dev/input/event0`
Utils: `ts_test`, `ts_calibrate`

(6, 1)
Module name: `mmc_block`
No ASIC3 MMC/SD card controller support that makes device unusable.

(10, 1)
Module name: `himalaya_lcd`
No control interface.

(11, 1)
Module name: `pxa-uda1380`
Driver does not compile.

(13, 1)
Module name: `himalaya_leds`
No control interface.

(2, 2)

Module name: `himalaya_kp`
Device path: `/dev/input/event1 (13, 65)`

(3, 2)
Module name: `himalaya_ts3`, `evdev`, `tsdev`
Device path: `/dev/input/event0 (13, 64)`
Environment variable: `TSLIB_TSDEVICE=/dev/input/event0`
Utils: `ts_test`, `ts_calibrate` (*tslib-tests* and *tslib-calibrate* packages in Ångström)

(5, 2), *(5, 1)*
Module name: `g_ether`
This module have to be loaded for main battery to charge from USB power.

(6, 2), *(6, 3)*
Module name: `asic3_mmc`, `mmc_block`
Device path for SD card: `/dev/mmcblk0 (254, 0)`
Device path for 1st partition of SD card: `/dev/mmcblk0p1 (254, 1)`
Device path for 2nd partition of SD card: `/dev/mmcblk0p2 (254, 2)`

(7, 2), *(7, 3)*
Module name: `diskonchip`
Contains the bootsplash, extended ROM (with CAB files) and storage.
Only 16 MiB version is properly supported by the driver, but Himalaya has a 32 MiB version.

(8, 2)
Module name: `blueangel_rom`, `mtdblock`
Device path: `/dev/mtdblock0 (31, 0)`
Contains 32 MiB RAW partition with Microsoft Pocket PC operating system.

(10, 2)
Module name: `himalaya_lcd`, `w100fb`
Control directory: `/sys/class/lcd/w100fb/`, `/sys/class/backlight/w100fb/`, `/sys/class/graphics/fb0/`
Framebuffer control, contrast and backlight brightness level control.

(11, 2)
Module name: `snd-uda1380`
Control directory: `/proc/asound/`
Driver successfully loads, but no soundcard found.

(12, 2)
Module name: `himalaya_battery`
Control directory: `/sys/class/battery/main battery/`
Battery monitoring.

(13, 2)
Module name: `himalaya_leds`
Control directory: `/sys/class/leds/`

(8, 3)
Module name: `blueangel_rom`, `mtdblock`
Device path: `/dev/mtdblock0 (31, 0)`
Error while inserting `blueangel_rom` module (no such device).

(10, 3)
Module name: `himalaya_lcd`, `w100fb`
Control directory: `/sys/class/lcd/w100fb/`, `/sys/class/graphics/fb0/`
Contrast level and framebuffer control, no backlight brightness level control.

(11, 3)
Module name: `snd-soc-uda1380`
Control directory: `/proc/asound/`
Driver successfully loads, but no soundcard found.

(10, 4)
LCD framebuffer does not work thus kernel output could not be displayed.

3.4 Running GNU/Linux on MDA II

This section describes how to use compiled Linux kernel and Ångström (Familiar) distribution together on HTC Himalaya. Booting will be done with HaRET bootloader (see section 3.2). No hardware modification or flashing of the device is necessary and afterwards we can just continue using it with original Microsoft Pocket PC. Note that HaRET can hard reset a device and therefore all data may be lost. It is necessary to make a backup if we need that data.

Linux kernel booting can be done with or without initial ramdisk (InitRD). Because SD card will be used for the root filesystem, we do not need InitRD for *2.6.12-hh2-xda0-ba0* and *2.6.21-devel* kernel and can use the rootfs directly. On the other hand InitRD must be used with *2.6.3-rmk0-hh0-xda0* kernel [9], because there is no SD card support (see subsection 3.3.3).

InitRD will be used with Familiar distribution (shell and no GUI — small space needed) while Ångström distribution will use SD card ext3 fs (GUI included — more space needed).

Linux booting sequence with InitRD and with SD card ext3 fs is shown in table 3.2.

3.4.1 Running GNU/Linux with InitRD

HaRET + Familiar + 2.6.3-rmk0-hh0-xda0

Let assume `/mnt/sd-vfat` as a default working directory with SD card mounted on it (vfat filesystem).[22]

Get HaRET bootloader:[23]

```
$ wget http://www.handhelds.org/~koconnor/haret/haret-0.5.1.exe
```

Copy compiled kernel image (see subsection 3.3.2):

```
$ cp $HOME/kernel/linux-2.6.3-rmk0-hh0-xda0/arch/arm/boot/zImage \
  zImage-2.6.3-rmk0-hh0-xda0
```

Get InitRD image, copy kernel modules and update start script:[24]

[22] Also Pocket PC filesystem could be used instead of SD card, but SD card is recommended to preserve all the saved files.
[23] Sometimes HaRET may hang during boot. Just soft reset (see section 2.2) and try again.
[24] To be able to connect HTC Himalaya using USB cable, it is necessary to edit downloaded InitRD image in order to load kernel module *g_ether* automatically during system boot.

Initial ramdisk	**SD card ext3 fs**
Operating system (Pocket PC)	*Operating system (Pocket PC)*
Bootloader (HaRET) – Loads kernel image to RAM – Loads initRD image to RAM – Executes the kernel image	*Bootloader (HaRET)* – Loads kernel image to RAM – Loads initRD image to RAM – Executes the kernel image
Kernel (Linux) – Uncompresses itself – Initializes statically compiled drivers – Mounts the initrd ext3 image as root – Executes the first userspace program `/linuxrc`	*Kernel (Linux)* – Uncompresses itself – Initializes statically compiled drivers – Mounts the ext3 fs from SD card as root – Executes the first userspace program `/sbin/init`
Userspace: (Familiar) `/linuxrc` (stored in the initrd) – Runs commands to configure the devices – Loads kernel modules (drivers) – Executes `/sbin/init`	
Userspace: `/sbin/init` (stored in the initrd) – Starts up services and user programs	*Userspace: (Ångström)* `/sbin/init` (stored in the SD card ext3 fs) – Runs commands to configure the devices – Loads kernel modules (drivers) – Starts up services and user programs

Table 3.2: Linux booting sequence with or without InitRD

```
$ wget http://wiki.xda-developers.com/uploads/initrd-2.6.3.gz
$ gunzip initrd-2.6.3.gz
# mount -o loop initrd-2.6.3 $MOUNT_DIR
# rm -rf $MOUNT_DIR/lib/modules/2.6.3-rmk0-hh0-xda0
# cd $HOME/kernel/linux-2.6.3-rmk0-hh0-xda0
# export INSTALL_MOD_PATH=$MOUNT_DIR
# make modules_install
# cd /mnt/sd-vfat
# echo -e "10a11\n> modprobe g_ether" | patch $MOUNT_DIR/linuxrc
# umount $MOUNT_DIR
$ gzip initrd-2.6.3
```

`$MOUNT_DIR` is a directory where you want to mount image.

Create HaRET `startup.txt` file in `/mnt/sd-vfat/`:

```
set KERNEL "zImage-2.6.3-rmk0-hh0-xda0"
set MTYPE 448
set INITRD "\Storage Card\initrd-2.6.3.gz"
set CMDLINE "root=/dev/ram0 init=/linuxrc ramdisk_size=14336 \
             keepinitrd console=ttyS1,115200"
bootlinux
```

Note 1: Machine type `448` matches the HTC Himalaya device.

Note 2: The name `Storage Card` may be localized depending on Windows version.
Note 3: For RS232 console use these parameters: 115200 Bauds, 8 data bits, no parity, 1 stop bit, no flow control.

Unmount vfat filesystem and continue host setup in subsection 3.4.3.

Output of *2.6.3-rmk0-hh0-xda0* kernel's `dmesg` is available in appendix B.1.

3.4.2 Running GNU/Linux without InitRD

HaRET + Ångström + 2.6.12-hh2-xda0-ba0

Let assume SD card with two partitions. The first is vfat type and is mounted in `/mnt/sd-vfat` directory and the later one is of ext3 type and is mounted in `/mnt/sd-ext3` directory.

Get HaRET bootloader:

```
$ cd /mnt/sd-vfat
$ wget http://www.handhelds.org/~koconnor/haret/haret-0.5.1.exe
```

Copy compiled kernel image (see subsection 3.3.2):

```
$ cp $HOME/kernel/2.6.12-hh2-xda0-ba0/arch/arm/boot/zImage \
  zImage-2.6.12-hh2-xda0-ba0
```

Extract rootfs image (see subsection 3.1.2) and copy kernel modules:

```
# cd /mnt/sd-ext3
# tar zvxf $HOME/OE/angstrom-stable/deploy/glibc/images/htchimalaya/\
  x11-image-htchimalaya.tar.gz
# rm -rf lib/modules/2.6.12-hh2-xda0-ba0
# cd $HOME/kernel/linux-2.6.12-hh2-xda0-ba0
# export INSTALL_MOD_PATH=/mnt/sd-ext3
# make modules_install
# cd /mnt/sd-ext3
```

Modify system files:

- `etc/X11/gpe-login.setup` to disable starting of *ipaq-sleep*,
- `etc/X11/Xserver` to setup touchscreen device,
- `etc/modutils/g_ether` to load USB ethernet module,
- `etc/network/interfaces` to setup network interfaces,
- `etc/default/zeroconf` to disable *zeroconf*,
- `etc/profile.d/tslib.sh` to setup *TSLIB_TSDEVICE* variable,
- `home/root/.ssh/authorized_keys` to use key for SSH login.

Create HaRET `startup.txt` file in `/mnt/sd-vfat/`:

```
set KERNEL "zImage-2.6.12-hh2-xda0-ba0"
set MTYPE 448
set CMDLINE "root=/dev/mmcblk0p2 rootdelay=1 console=ttyS0,115200n8 \
             console=tty0 psplash=false"
bootlinux
```

Note 1: Machine type `448` matches the HTC Himalaya device.
Note 2: `/dev/mmcblk0p2` represents ext3 partition on SD card.
Note 3: With `psplash` an startup image could be disabled.
Note 4: For RS232 console use these parameters: 115200 Bauds, 8 data bits, no parity, 1 stop bit, no flow control.

Unmount vfat and ext3 filesystem and follow host setup in subsection 3.4.3.

Output of *2.6.12-hh2-xda0-ba0* kernel's `dmesg` is available in appendix B.2.

HaRET + Ångström + 2.6.21-devel

Running Ångström distribution with *2.6.21-devel* kernel is very similar to Ångström with *2.6.12-hh2-xda0-ba0* kernel (see subsection 3.4.2). The only difference is substitution of *2.6.12-hh2-xda0-ba0* for *2.6.21-devel* in instructions above.

Output of *2.6.21-devel* kernel's `dmesg` is available in appendix B.3.

3.4.3 Host Configuration

Insert SD card to HTC Himalaya and connect Himalaya with RS232 or USB cable to the host to be able to log in (see section 2.3). Then tap on `haret-0.5.1.exe` on Himalaya and watch the screen or serial terminal. Linux kernel boots and all necessary services should start.

Network interface *usb0* is configured on HTC Himalaya as follows:

```
iface usb0 inet static
    address 192.168.0.206
    netmask 255.255.255.0
    network 192.168.0.0
    gateway 192.168.0.205
```

On the host we need to compile and load kernel modules *usbcore*, *usbnet* and *cdc_subset*. Then interface *usb0* should be visible on the host as well, so we can configure it this way:

```
# ifconfig usb0
usb0      Zapouzdření:Ethernet  HWadr EA:FA:BD:48:A7:33
          VŠESMĚROVÉ_VYSÍLÁNÍ MULTICAST  MTU:1500  Metrika:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          kolizí:0 délka odchozí fronty:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

# ifconfig usb0 192.168.0.205 netmask 255.255.255.0 up
```

For connection test we use *ping* and on success *ssh* for remote administration (user *root*, password *rootme*):

```
# ping -c 3 192.168.0.206
PING 192.168.0.206 (192.168.0.206) 56(84) bytes of data.
64 bytes from 192.168.0.206: icmp_seq=1 ttl=64 time=1.36 ms
...

# ssh root@192.168.0.206
```

Chapter 4

MDA II Component Code Implementation

4.1 General-Purpose Input/Output

The PXA263 processor [14] provides 90 *General-Purpose Input/Output* (GPIO) pins for use in generating and capturing application-specific input and output signals. Each pin can be programmed as either input or output. When programmed to be input, a GPIO can also serve as an interrupt source. Note that the first 86 GPIO pins (GPIO[85:0]) are configured as inputs during the assertion of all resets, and remain as inputs until they are configured otherwise. Primary GPIO pins are not shared with peripherals while secondary ones have alternate functions which can be mapped to peripherals.

Figure 4.1 shows PXA263 processor's block diagram with General-Purpose Input/Output interface that could be used for peripheral's control.

Kernel functions and macros to operate PXA263 GPIO's pins are described in subsection 4.2.1.

4.1.1 GPIO Registers

The PXA263 processor enable and control its GPIO pins through the use of special registers which configure:

- pin direction (*input* or *output*),
- pin state (*low* or *high*, outputs only),
- pin level detection (*low* or *high*, inputs only),
- selection of alternate functions.

There are a total of twenty-seven 32-bit registers to GPIO pins control (see table 4.1):

- *GPDR* — GPIO Pin Direction Registers (read/write),
- *GPLR* — GPIO Pin Level Registers (read),
- *GPSR* — GPIO Pin output Set Registers (write),
- *GPCR* — GPIO Pin output Clear Registers (write),

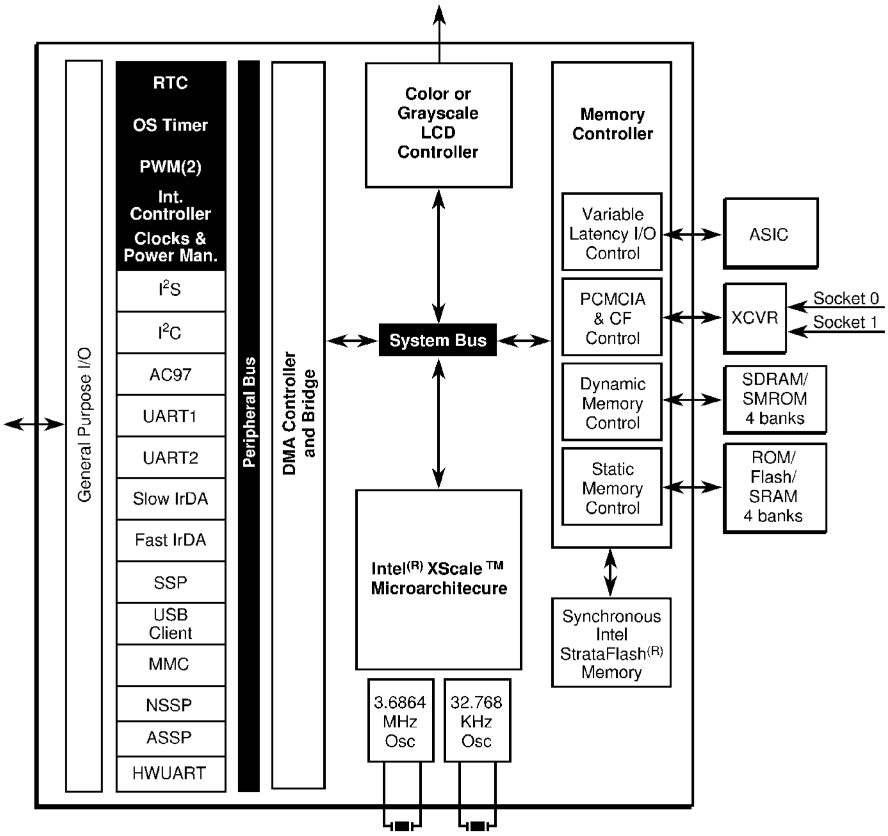

Figure 4.1: PXA263 processor's block diagram [14]

- *GRER* — GPIO Rising-Edge detect Registers (read/write),
- *GFER* — GPIO Falling-Edge detect Registers (read/write),
- *GEDR* — GPIO Edge detect Status Registers (read/write),
- *GAFR* — GPIO Alternate Function Registers (read/write).

Use the *GPDR* to set (or get) whether the GPIO pins are outputs or inputs. When programmed as output, the pin can be set high by writing to the *GPSR* and cleared low by writing to the *GPCR*. The set and clear registers can be written regardless of whether the pin is configured as input or output. If a pin is configured as input, the programmed output state occurs when the pin is reconfigured to be output. To validate each GPIO pin's state read the *GPLR*. You can read this register any time to confirm the state of a pin. In addition, use the *GRER* and *GFER* to detect either a rising edge or falling edge on each GPIO pin. Use the *GEDR* to read edge detect state. These edge detects can be programmed to generate interrupts.

Most peripherals connect to the external pins through GPIOs. To use a peripheral connected through a GPIO, the software must first configure the GPIO by writing to the *GAFR* so that the desired peripheral is connected to its pins. The default state for most of the pins is input. Some of the GPIOs default to their alternate function and do not need to be configured for use.

GPIO registers' schema is shown in the PXA263 GPIO block diagram (see figure 4.2).

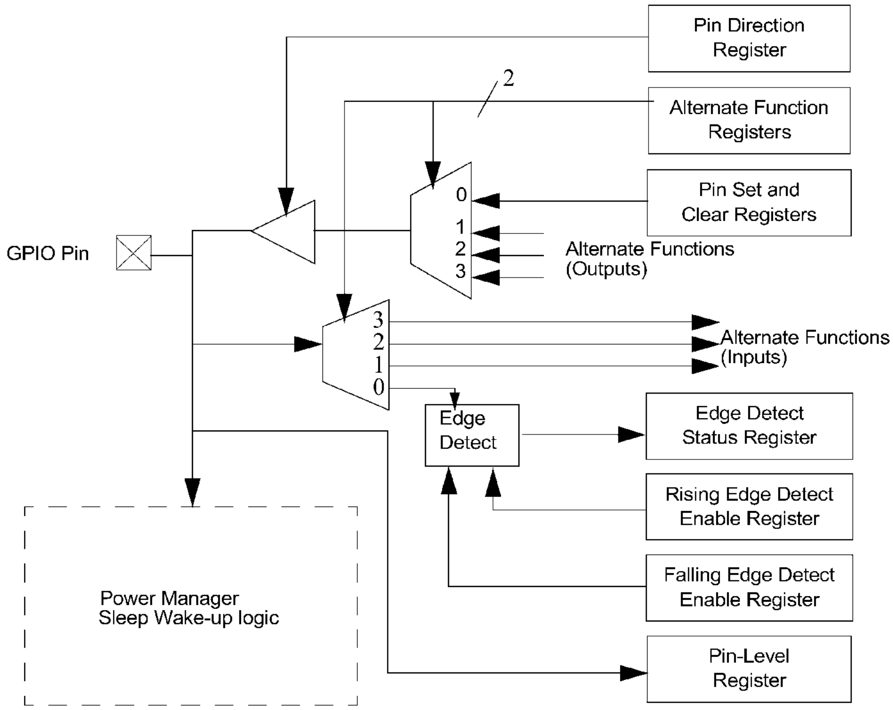

Figure 4.2: PXA263 GPIO block diagram [14]

4.1.2 GPIO Register Definitions

In order to be able to proceed to implementation (see sections 4.3 and 4.4) we need to familiarize ourselves with PXA263 GPIO registers.

GPIO Pin Level Registers

Check the state of each of the GPIO pins by reading the *GPLR* [14]. Each bit in the *GPLR* corresponds to one pin in the GPIO. According to table 4.1 there are three *GPLR* registers (*GPLR0*, *GPLR1*, *GPLR2*). GPLR0[31:0] correspond to GPIO[31:0], GPLR1[31:0] correspond to GPIO[63:32] and GPLR2[25:0] correspond to GPIO[89:64]. Use the *GPLR0–2* read-only registers to determine the current value of a particular pin (regardless of the programmed pin direction). For reserved bits (GPLR2[31:26]), reads return zero.

Physical addresses of *GPLR* registers are mentioned in table 4.2.

GPIO Pin Direction Registers

Whether a pin is input or output is controlled by programming the *GPDR* [14]. According to table 4.1 there are three *GPDR* registers (*GPDR0*, *GPDR1*, *GPDR2*). The *GPDR* registers contain one direction control bit for each of the 90 GPIO pins. For GPIO[85:0], if a direction bit is programmed to one, the GPIO is output. If it is programmed to zero, it is input. For GPIO[89:86], if a direction bit is programmed to one, the GPIO is input.

If it is programmed to zero, it is output. Reserved bits (GPDR2[31:26]), should be written with zeros and reads of the reserved bits should be ignored.

Physical addresses of *GPDR* registers are mentioned in table 4.2.

GPIO Pin Output Set and Pin Output Clear Registers

When a GPIO is configured as output, you control the state of the pin by writing to either the *GPSR* or the *GPCR* [14]. According to table 4.1 there are three *GPSR* registers (*GPSR0*, *GPSR1*, *GPSR2*) and three *GPCR* registers (*GPCR0*, *GPCR1*, *GPCR2*). Output pin is set high by writing one to its corresponding bit within the *GPSR*. To clear an output pin, one is written to the corresponding bit within the *GPCR*. *GPSR* and *GPCR* are write-only registers. Reads return unpredictable values.

Writing zero to any of *GPSR* or *GPCR* bits has no effect on the state of the pin. Writing one to a *GPSR* or *GPCR* bit corresponding to a pin that is configured as input is effective only after the pin is configured as output. Reserved bits (GPSR2[31:26] and GPCR2[31:26]), must be written with zeros and reads must be ignored.

Physical addresses of *GPSR* and *GPCR* registers are mentioned in table 4.2.

GPIO Alternate Function Registers

According to table 4.1 there are three *GAFR* [14] registers (*GAFR0*, *GAFR1*, *GAFR2*). The *GAFRs* contain select bits that correspond to the 90 GPIO pins. Each GPIO can be configured to be either a generic GPIO pin, one of 3 alternate input functions, or one of 3 alternate output functions. To select any of the alternate input functions, the *GPDR* register must configure the GPIO to be input. Similarly, only GPIOs configured as outputs by the *GPDR* can be configured for alternate output functions. Each GPIO pin has a pair of bits assigned to it whose values determine which function (normal GPIO, alternate function 1, alternate function 2 or alternate function 3) the GPIO performs.

The function selected is determined by writing the *GAFR* bit pair as below:

- *00* indicates normal GPIO function for GPIO[85:0]. Indicates default dedicated functionality for GPIO[89:86].
- *01* selects alternate input function 1 (*ALT_FN_1_IN*) or alternate output function 1 (*ALT_FN_1_OUT*) for GPIO[85:0]. Selects GPIO function for GPIO[89:86].
- *10* selects alternate input function 2 (*ALT_FN_2_IN*) or alternate output function 2 (*ALT_FN_2_OUT*).
- *11* selects alternate input function 3 (*ALT_FN_3_IN*) or alternate output function 3 (*ALT_FN_3_OUT*).

Physical addresses of *GAFR* registers could be found in table 4.2.

4.1.3 ASIC3 GPIO

Besides PXA263 General-Purpose Input/Output HTC Himalaya has *Application-Specific Integrated Circuit* (ASIC3) for peripherals control. Chip has 64 pins divided into four 16 bit registers (*GPIOA[15:0]*, *GPIOB[15:0]*, *GPIOC[15:0]*, *GPIOD[15:0]*).

Kernel functions and macros to operate ASIC3 GPIO's pins are described in subsection 4.2.2.

Register	GPIO[15:0]	GPIO[31:16]	GPIO[47:32]	GPIO[63:48]	GPIO[79:64]	GPIO[80]
GPLR	GPLR0		GPLR1		GPLR2	
GPSR	GPSR0		GPSR1		GPSR2	
GPCR	GPCR0		GPCR1		GPCR2	
GPDR	GPDR0		GPDR1		GPDR2	
GRER	GRER0		GRER1		GRER2	
GFER	GFER0		GFER1		GFER2	
GEDR	GEDR0		GEDR1		GEDR2	
GAFR	GAFR0_L	GAFR0_U	GAFR1_L	GAFR1_U	GAFR2_L	GAFR2_U

Table 4.1: PXA263 GPIO register definitions

Register name	Physical address	Register name	Physical address
GPLR0	0x40E0_0000	GPSR0	0x40E0_0018
GPLR1	0x40E0_0004	GPSR1	0x40E0_001C
GPLR2	0x40E0_0008	GPSR2	0x40E0_0020
GPDR0	0x40E0_000C	GPCR0	0x40E0_0024
GPDR1	0x40E0_0010	GPCR1	0x40E0_0028
GPDR2	0x40E0_0014	GPCR2	0x40E0_002C
GAFR0_L	0x40E0_0054	GAFR1_U	0x40E0_0060
GAFR0_U	0x40E0_0058	GAFR2_L	0x40E0_0064
GAFR1_L	0x40E0_005C	GAFR2_U	0x40E0_0068

Table 4.2: PXA263 registers' physical addresses

4.2 Kernel GPIO's Macros and Functions

Handhelds Linux kernel provides the following macros and functions for PXA263/ASIC3 GPIO operations.

4.2.1 PXA263 GPIO

Macros manipulating PXA263 GPIO registers:

- *GPIO_bit(x)* — chooses appropriate GPIO bit,
- *GPDR(x)* — finds appropriate GPDR register and reads from or writes to it,
- *GPLR(x)* — finds appropriate GPLR register and reads from it,
- *GPSR(x)* — finds appropriate GPSR register and writes to it,
- *GPCR(x)* — finds appropriate GPCR register and writes to it,
- *GAFR(x)* — finds appropriate GAFR register and reads from or writes to it.

Basic definitions could be found in `include/asm-arm/arch-pxa/pxa-regs.h` file:

```
#define GPIO_bit(x)   (1 << ((x) & 0x1f))
#define GPLR(x)   __REG2(0x40E00000, ((x) & 0x60) >> 3)
#define GPDR(x)   __REG2(0x40E0000C, ((x) & 0x60) >> 3)
```

```
#define GPSR(x)    __REG2(0x40E00018, ((x) & 0x60) >> 3)
#define GPCR(x)    __REG2(0x40E00024, ((x) & 0x60) >> 3)
#define GAFR(x)    __REG2(0x40E00054, ((x) & 0x70) >> 2)
```

Examples

1. Set PXA263 GPIO pin *5* as output (see subsection 4.1.2):

```
GPDR(5) |= GPIO_bit(5)
```

GPDR(5) finds direction register for GPIO pin *5* (thus *GPDR0* at address 0x40E0000C — see tables 4.1 and 4.2) and *GPIO_bit(5)* sets the bit *5* to one for that register. Note that there is an assignment by bitwise OR because we need to set just bit *5* and the other bits leave as they are.

2. Set PXA263 GPIO pin *49* as input (see subsection 4.1.2):

```
GPDR(49) &= ~GPIO_bit(49)
```

GPDR(49) finds direction register for GPIO pin *49* (thus *GPDR1* at address 0x40E00010 — see tables 4.1 and 4.2) and *GPIO_bit(49)* sets the bit *49* to zero for that register. Note that there is an assignment by bitwise AND and bitwise one's complement because we need to clear just bit *49* and the other bits leave as they are.

3. Set PXA263 GPIO pin *16* to one (see subsection 4.1.2):

```
GPSR(16) = GPIO_bit(16)
```

GPSR(16) finds set register for GPIO pin *16* (thus *GPSR0* at address 0x40E00018 — see tables 4.1 and 4.2) and *GPIO_bit(16)* sets the bit *16* to one for that register. Note that there is a basic assignment because writing zero to the other bits has no effect on the state of the pins.

4. Set PXA263 GPIO pin *18* to zero (see subsection 4.1.2):

```
GPCR(18) = GPIO_bit(18)
```

GPCR(18) finds clear register for GPIO pin *18* (thus *GPCR0* at address 0x40E00024 — see tables 4.1 and 4.2) and *GPIO_bit(18)* sets the bit *18* to one for that register. Note that there is a basic assignment because writing zero to the other bits has no effect on the state of the pins.

5. Check PXA263 GPIO pin *62* value (see subsection 4.1.2):

```
GPLR(62) & GPIO_bit(62)
```

GPLR(62) finds level register for GPIO pin *62* (thus *GPLR1* at address 0x40E00004 — see tables 4.1 and 4.2) and *GPIO_bit(62)* checks the bit *62* for a value for that register.

4.2.2 ASIC3 GPIO

Functions manipulating ASIC3 GPIO registers:

- *asic3_set_gpio_dir_a()* — sets ASIC3 GPIOA pin direction,
- *asic3_set_gpio_out_a()* — sets ASIC3 GPIOA pin value,
- *asic3_get_gpio_status_a()* — gets ASIC3 GPIOA pin value.

Analogous functions are available for ASIC3 *GPIOB*, *GPIOC* and *GPIOD* registers.

Examples

1. Set ASIC3 GPIOA pin *6* as output:

```
asic3_set_gpio_dir_a(&himalaya_asic3.dev, 1<<6, 1<<6)
```

2. Set ASIC3 GPIOB pin *6* to one:

```
asic3_set_gpio_out_b(&himalaya_asic3.dev, 1<<6, 1<<6)
```

3. Set ASIC3 GPIOB pin *6* to zero:

```
asic3_set_gpio_out_b(&himalaya_asic3.dev, 1<<6, 0)
```

4. Check ASIC3 GPIOC pin *8* value:

```
asic3_get_gpio_status_c(&himalaya_asic3.dev) & (1<<8)
```

4.3 2.6.21-devel Drivers Implementation

This section describes how to implement component code for Linux 2.6.21-devel branch (the latest development *handhelds* kernel — see subsection 3.3.3).

There are a few important directories where HTC Himalaya's specific code is placed:

- `include/asm-arm/arch-pxa/` containing PXA CPUs' and Himalaya's header files,
- `arch/arm/mach-pxa/htchimalaya/` containing Himalaya's component code.

The driver implementation will be done in directories mentioned above.

`arch/arm/mach-pxa/htchimalaya/` directory contains two files used for kernel configuration and compilation:

- `Kconfig` contains kernel menu configuration of HTC Himalaya,
- `Makefile` contains build configuration.

Tables 4.3 and 4.4 summarize major PXA263 GPIO and ASIC3 GPIO pins used for code implementation (see following subsections).

[1]The TSC2200 is analog interface circuit implementing, among others, touchscreen and button interface.

Pin	Pin and its IRQ alias	I/O	Description
03	GPIO_NR_HIMALAYA_USB_DETECT_N IRQ_NR_HIMALAYA_USB_DETECT_N	input	USB cradle (0=dock, 1=undock)
05	GPIO_NR_HIMALAYA_TS_IRQ_N IRQ_NR_HIMALAYA_TSC2200_TS	input	TSC2200[1]touchscreen IRQ (0 on change)
07	GPIO_NR_HIMALAYA_TSC2200_KB_N IRQ_NR_HIMALAYA_TSC2200_KB	input	TSC2200 buttons IRQ (0 on change)
60	GPIO_NR_HIMALAYA_USB_PULLUP_N —	output	USB power (0=on, 1=off)
63	GPIO_NR_HIMALAYA_CHARGER_EN —	output	battery charger (0=off, 1=on)

Table 4.3: Major PXA263 General-Purpose Inputs/Outputs

Pin	Pin alias	I/O	Description
D14	GPIOD_AC_CHARGER_N	input	AC adapter present (0=yes, 1=no)

Table 4.4: Major ASIC3 General-Purpose Inputs/Outputs

4.3.1 USB Communication

Implementation Goal

USB power enabling or disabling does not work in 2.6.21-devel kernel when connecting or disconnecting Himalaya with host computer (via USB cradle or cable). It makes USB interface absolutely unusable. My goal is to fix this issue.

Affected Files

`himalaya_udc.c`, `htchimalaya-gpio.h`, `Kconfig`, `Makefile`

Code with Comments

Function for detecting USB cradle/cable plugging. It uses PXA263 GPIO bit *3* (see table 4.3).

```
static int himalaya_udc_is_connected(void)
{
  int ret = !(GPLR(GPIO_NR_HIMALAYA_USB_DETECT_N) &
            GPIO_bit(GPIO_NR_HIMALAYA_USB_DETECT_N));
  printk(KERN_DEBUG "udc_is_connected returns %d\n", ret);
  return ret;
}
```

Function for activating USB power (low means enabled). It uses PXA263 GPIO bit *60* (see table 4.3).

```
static void himalaya_udc_command(int cmd)
{
```

```
  /* activates USB power (low = enabled) */
  switch (cmd) {
  case PXA2XX_UDC_CMD_DISCONNECT:
    printk(KERN_DEBUG "_udc_control: disconnect\n");
    GPSR(GPIO_NR_HIMALAYA_USB_PULLUP_N) = /* usb power disable */
        GPIO_bit(GPIO_NR_HIMALAYA_USB_PULLUP_N);
    break;
  case PXA2XX_UDC_CMD_CONNECT:
    printk(KERN_DEBUG "_udc_control: connect\n");
    GPCR(GPIO_NR_HIMALAYA_USB_PULLUP_N) = /* usb power enable */
        GPIO_bit(GPIO_NR_HIMALAYA_USB_PULLUP_N);
    break;
  default:
    printk(KERN_WARNING "_udc_control: unknown command!\n");
    break;
  }
}
```

Function for setting PXA263 GPIO bit *60* as output.

```
static int himalaya_udc_probe(struct device *dev)
{
  printk(KERN_INFO "himalaya udc register\n");
  GPDR(GPIO_NR_HIMALAYA_USB_PULLUP_N) |= /* set GPIO bit as output */
      GPIO_bit(GPIO_NR_HIMALAYA_USB_PULLUP_N);
  pxa_set_udc_info(&himalaya_udc_mach_info);
  return 0;
}
```

Kernel Configuration

Kconfig

```
config MACH_HIMALAYA_UDC
  tristate "HTC Himalaya UDC support"
  depends on MACH_HIMALAYA_ASIC3 && MACH_HIMALAYA
  help
    Module to handle USB client enable and disable.
```

Makefile

```
obj-$(CONFIG_MACH_HIMALAYA_UDC) += himalaya_udc.o
```

4.3.2 Power and Battery Management

Implementation Goal

Power and battery charging does not work in 2.6.21-devel kernel when connecting Himalaya and host computer via USB interface or AC adapter. Device is powered only by internal battery and die on battery discharge. My goal is to fix this issue.

Affected Files

`himalaya_power.c`, `htchimalaya-gpio.h`, `htchimalaya-asic.h`, `Kconfig`, `Makefile`

Code with Comments

Function for enabling or disabling battery charging follows. It uses PXA263 GPIO bit *63* (see table 4.3). *flags* variable is *2* in the case of USB cradle/cable charging, *1* in the case of AC power charging, and *0* in the case of charging is disabled.

```
static void himalaya_set_charge(int flags)
{
  printk(KERN_DEBUG "himalaya_set_charge: enabled=%d\n", flags);
  GPDR(GPIO_NR_HIMALAYA_CHARGER_EN) |=   /* set GPIO bit as output */
      GPIO_bit(GPIO_NR_HIMALAYA_CHARGER_EN);
  /* Activates charging (low = disabled) */
  if (flags) {   /* battery charger enable */
    GPSR(GPIO_NR_HIMALAYA_CHARGER_EN) =
        GPIO_bit(GPIO_NR_HIMALAYA_CHARGER_EN);
  } else {       /* battery charger disable */
    GPCR(GPIO_NR_HIMALAYA_CHARGER_EN) =
        GPIO_bit(GPIO_NR_HIMALAYA_CHARGER_EN);
  }
}
```

Function for detecting AC power (power adapter). It uses ASIC3 GPIO bit *D14* (see table 4.4).

```
static int himalaya_is_ac_online(void)
{
  return (asic3_get_gpio_status_d(&himalaya_asic3.dev) &
         (1<<GPIOD_AC_CHARGER_N)) == 0;
}
```

Function for detecting USB cradle/cable plugging. It uses PXA263 GPIO bit *3* (see table 4.3).

```
static int himalaya_is_usb_online(void)
{
  return (GPLR(GPIO_NR_HIMALAYA_USB_DETECT_N) &
          GPIO_bit(GPIO_NR_HIMALAYA_USB_DETECT_N)) == 0;
}
```

Function for defining AC adapter and USB cradle/cable interrupt requests. It uses PXA263 GPIO bit *3* IRQ (see table 4.3) and ASIC3 GPIO bit *D14* IRQ.

```
static int himalaya_power_init(void)
{
  int ret;
  unsigned int ac_irq, usb_irq;
```

```
  ac_irq = asic3_irq_base(&himalaya_asic3.dev) + ASIC3_GPIOD_IRQ_BASE +
           GPIOD_AC_CHARGER_N;
  usb_irq = IRQ_NR_HIMALAYA_USB_DETECT_N;

  himalaya_power_pdev.resource[0].start = ac_irq;
  himalaya_power_pdev.resource[0].end = ac_irq;
  himalaya_power_pdev.resource[1].start = usb_irq;
  himalaya_power_pdev.resource[1].end = usb_irq;

  ret = platform_device_register(&himalaya_power_pdev);
  if (ret)
    printk(KERN_ERR DRIVER_NAME ": registration failed\n");
  return ret;
}
```

Kernel Configuration

Kconfig

```
config MACH_HIMALAYA_POWER
  tristate "HTC Himalaya power support"
  depends on MACH_HIMALAYA
  help
    Module to handle HTC Himalaya battery charging via AC adapter or USB
    cradle/cable.
```

Makefile

```
obj-$(CONFIG_MACH_HIMALAYA_POWER) += himalaya_power.o
```

Improvement Suggestions

Battery monitoring implementation (voltage, temperature and capacity measuring).

4.3.3 Touchscreen and Buttons

Implementation Goal

Touchscreen and buttons do not work in 2.6.21-devel kernel. My goal is to make touchscreen and TSC2200 buttons (*calendar*, *contacts*, *phone*, *cancel*) working.

Affected Files

`himalaya_tsc2200.c`, `himalaya.c`, `htchimalaya-gpio.h`, `Kconfig`, `Makefile`

Code with Comments

Structure defining touchscreen parameters. It uses PXA263 GPIO bit *5* (see table 4.3).

```
struct tsadc_platform_data himalaya_ts_params = {
  .pen_gpio         = GPIO_NR_HIMALAYA_TS_IRQ_N,
  .x_pin            = "tsc2200-adc.0:x",
```

```
  .y_pin           = "tsc2200-adc.0:y",
  .z1_pin          = "tsc2200-adc.0:z1",
  .z2_pin          = "tsc2200-adc.0:z2",
  .pressure_factor = 100000,
  .min_pressure    = 2,
  .max_jitter      = 8,
};
```

Structure defining Himalaya's TSC2200 buttons. There are *keyname*, *keyindex* and *keycode*.

```
static struct tsc2200_key himalaya_tsc2200_keys[] = {
  {"contacts",        1,    KEY_F5},    /* KEY_CONTACTS */
  {"calendar",        7,    KEY_F6},    /* KEY_CALENDAR */
  {"phone_lift",     11,    KEY_F7},    /* KEY_PHONE */
  {"phone_hangup",   15,    KEY_F8},    /* KEY_CANCEL */
};
```

Structure defining button interrupt request. It uses PXA263 GPIO bit *7* IRQ (see table 4.3).

```
static struct tsc2200_buttons_platform_info himalaya_tsc2200_buttons = {
  .keys     = himalaya_tsc2200_keys,
  .num_keys = ARRAY_SIZE(himalaya_tsc2200_keys),
  .irq      = IRQ_NR_HIMALAYA_TSC2200_KB,
};
```

Structure defining touchscreen interrupt request. It uses PXA263 GPIO bit *5* IRQ (see table 4.3).

```
struct tsc2200_ts_platform_info himalaya_tsc2200_ts = {
  .irq      = IRQ_NR_HIMALAYA_TSC2200_TS,
};
```

Kernel Configuration

Kconfig

```
menuconfig MACH_HIMALAYA
  bool "HTC Himalaya"
  depends on ARCH_PXA
  select PXA25x
  select PXA26x
  select BOARD_IRQ_MAP_BIG
  select PXA_SSP
  select HTC_ASIC3
  select TOUCHSCREEN_TSC2200
  select KEYBOARD_TSC2200
```

Makefile

```
obj-$(CONFIG_MACH_HIMALAYA) += himalaya_tsc2200.o
```

Improvement Suggestions

ASIC3 buttons implementation (*record*, *camera*, *volume up*, *volume down*, *select*, *up*, *down*, *left*, *right*).

4.4 2.6.28-devel Drivers Implementation

Section presents how to implement code for Linux 2.6.28-devel branch (the latest *vanilla* ARM development kernel — see subsection 3.3.3).

There are a few important directories where HTC Himalaya's specific code is placed:

- `arch/arm/mach-pxa/include/mach/` containing PXA CPUs' header files,
- `arch/arm/mach-pxa/` containing PXA CPUs' components code.

The driver implementation will be done in directories mentioned above.

`arch/arm/mach-pxa/` directory has two files used for kernel configuration and compilation:

- `Kconfig` contains kernel menu configuration of HTC Himalaya,
- `Makefile` contains build configuration.

4.4.1 Hardware Definitions and LCD Display

Implementation Goal

There is no specific support for HTC Himalaya in 2.6.28-devel kernel. My goal is to create platform hardware definitions and to make LCD display framebuffer to get working.

Affected Files

`himalaya.c`, `Kconfig`, `Makefile`

Code with Comments

Structure with platform definitions.

```
MACHINE_START(HIMALAYA, "HTC Himalaya")
  .phys_io      = 0x40000000,
  .io_pg_offst  = (io_p2v(0x40000000) >> 18) & 0xfffc,
  .boot_params  = 0xa0000100,
  .map_io       = pxa_map_io,
  .init_irq     = pxa25x_init_irq,
  .init_machine = himalaya_init,
  .timer        = &pxa_timer,
MACHINE_END
```

Function for setting LCD framebuffer mode. There are two modes depending on the Himalaya *BoardID*. Its value is hardcoded here because detection needs an ASIC3 functions that are not available in this kernel.

```
static void __init himalaya_lcd_init(void)
{
  int himalaya_boardid;

  himalaya_boardid = 0x4; /* hardcoded (detection needs ASIC3 functions) */
  printk(KERN_INFO "himalaya LCD Driver init. boardid=%d\n",
         himalaya_boardid);

  switch (himalaya_boardid) {
  case 0x4:
    himalaya_fb_info.modelist = &himalaya4_lcd_mode;
  break;
  case 0x6:
    himalaya_fb_info.modelist = &himalaya6_lcd_mode;
  break;
  default:
    printk(KERN_INFO "himalaya lcd_init: unknown boardid=%d. Using 0x4\n",
           himalaya_boardid);
    himalaya_fb_info.modelist = &himalaya4_lcd_mode;
  }
}
```

Structure defining LCD framebuffer registers.

```
static struct w100_gen_regs himalaya_lcd_regs = {
  .lcd_format   = 0x00000003,
  .lcdd_cntl1   = 0x00000000,
  .lcdd_cntl2   = 0x0003ffff,
  .genlcd_cntl1 = 0x00fff003,
  .genlcd_cntl2 = 0x00000003,
  .genlcd_cntl3 = 0x000102aa,
};
```

Structure defining framebuffer mode parametres for BoardID *4*. Structure defining parametres for BoardID *6* is very similar.

```
static struct w100_mode himalaya4_lcd_mode = {
  .xres                  = 240,
  .yres                  = 320,
  .left_margin           = 0,
  .right_margin          = 31,
  .upper_margin          = 15,
  .lower_margin          = 0,
  .crtc_ss               = 0x80150014,
  .crtc_ls               = 0xa0fb00f7,
  .crtc_gs               = 0xc0080007,
  .crtc_vpos_gs          = 0x00080007,
  .crtc_rev              = 0x0000000a,
  .crtc_dclk             = 0x81700030,
  .crtc_gclk             = 0x8015010f,
```

```
  .crtc_goe              = 0x00000000,
  .pll_freq              = 80,
  .pixclk_divider        = 15,
  .pixclk_divider_rotated = 15,
  .pixclk_src            = CLK_SRC_PLL,
  .sysclk_divider        = 0,
  .sysclk_src            = CLK_SRC_PLL,
};
```

Structure defining LCD framebuffer resources.

```
static struct resource himalaya_fb_resources[] = {
  [0] = {
    .start = 0x08000000,
    .end   = 0x08ffffff,
    .flags = IORESOURCE_MEM,
  },
};
```

Kernel Configuration

Kconfig

```
config MACH_HIMALAYA
  bool "HTC Himalaya Support"
  select CPU_PXA26x
  select FB_W100
```

Makefile

```
obj-$(CONFIG_MACH_HIMALAYA) += himalaya.o
```

Improvement Suggestions

LCD control implementation (contrast, backlight brightness, etc.).

4.5 Code Implementation Results

Table 4.5 summes up Linux *2.6.21-devel* and *2.6.28-devel* kernel support before and after my code implementation.

Please note that code mentioned in sections 4.3 and 4.4 is not complete (full version can be found on the attached CD).

Output of *2.6.21-devel* kernel `dmesg` after my implementation is available in appendix B.4. Output of *2.6.28-devel* kernel `dmesg` is not available because a serial cable has not been accessible therefore kernel output has been visible only on Himalaya's LCD display.

[2]Linux *2.6.21-devel* kernel before my code implementation.
[3]Linux *2.6.21-devel* kernel after my code implementation.
[4]Linux *2.6.28-devel* kernel before my code implementation.
[5]Linux *2.6.28-devel* kernel after my code implementation.

		1	*2*	*3*	*4*
	Component	**2.6.21-devel** *orig*[2]	**2.6.21-devel** *my*[3]	**2.6.28-devel** *orig*[4]	**2.6.28-devel** *my*[5]
1	Booting	Part+	Part+	?	**Part**+
2	HW Buttons	No	**Part**+	?	?
3	Touchscreen	Part−	**Yes**	?	?
4	Serial Ports	Yes	Yes	?	?
5	USB	Part−	**Yes**	?	?
6	SD/SDIO Card	Yes	Yes	?	?
7	DiskOnChip	Part−	Part−	?	?
8	Strata Flash	Part−	Part−	?	?
9	GSM/GPRS	No	No	?	?
10	LCD	Part+	Part+	?	**Part**+
11	Sound	Part−	Part−	?	?
12	Battery	Part−	**Part**+	?	?
13	LEDs	No	No	?	?
14	Camera	No	No	?	?
15	Bluetooth	No	No	?	?

Legend:	No	unimplemented
	Part−	partially implemented (component is unusable)
	Part+	partially implemented (component is usable)
	Yes	fully implemented
	?	unidentified

Table 4.5: Features comparison between Linux *orig* and *my* kernels

Kernel Feature Notes

Here are some useful extended notes about current support of components in Linux 2.6 kernel after my implementation (see table 3.1 for other kernel branches).

For example *(1, 2)* stands for row 1 and column 2 of table 4.5, thus *Booting on 2.6.21-devel new* kernel.

(2, 2)
Module name: `himalaya_tsc2200` (compiled in `himalaya`)
Device path: `/dev/input/event0 (13, 64)`
Implements HTC Himalaya TSC2200 button support.

(3, 2)
Module name: `himalaya_tsc2200` (compiled in `himalaya`), `evdev`, `tsdev`
Device path: `/dev/input/event1 (13, 65)`
Environment variable: `TSLIB_TSDEVICE=/dev/input/event1`
Utils: `ts_test`, `ts_calibrate` (*tslib-tests* and *tslib-calibrate* packages in Ångström)
Implements HTC Himalaya TSC2200 touchscreen support.

(5, 2)
Module name: `g_ether`
This module have to be loaded for IP communication via USB interface.

(12, 2)
Module name: `himalaya_power`, `pda_power`
This module have to be loaded for battery charging by AC adapter or USB power.

(10, 4)
Module name: `himalaya`, `w100fb`
Implements HTC Himalaya LCD framebuffer support.

4.6 Kernel Patch Submitting

This section provides information about how to create patches for Linux kernel sources and their submitting procedure. All patches can be found on the attached CD.

4.6.1 2.6.21-devel Branch

Patch creating:

```
$ cd $HOME/kernel
$ diff -Naru linux-2.6.21-devel-orig linux-2.6.21-devel-my \
  > htchimalaya.patch
```

Description: Patch adds support for TSC2200 touchscreen, TSC2200 buttons, battery charge handling and includes USB power fixes. Furthermore patch contains other minor fixes and Himalaya code clean-up (all of the code is now compliant to `checkpatch.pl` script available in recent Linux kernels).

Submit address: *kernel-bugs@handhelds.org*

URL: http://www.handhelds.org/hypermail/kernel-bugs/current/0238.html

Current status[6]**:** Should be merged into *Handhelds* CVS by *cr2* shortly. This patch has been already merged into *Linux-To-Go* GIT[7] repository by *pH5*.

Comments: Many useful advices about *Handhelds* kernel patching provided at irc.freenode.net#htc-linux by *cr2*, *pH5*, *tmzt*, *BabelO*, *dcordes*, *marex* and *Kevin2*.

4.6.2 2.6.28-devel Branch

Patch creating[8]**:**

```
$ cd $HOME/kernel/linux-2.6.28-devel
$ git add arch/arm/mach-pxa/Kconfig arch/arm/mach-pxa/Makefile \
  arch/arm/mach-pxa/himalaya.c
$ git commit -m "HTC Himalaya framebuffer support"
$ git log
$ git format-patch 4655a0de36e8e903e99a8d152818e3aae86dae1a
$ scripts/checkpatch.pl 0001-HTC-Himalaya-framebuffer-support.patch
```

[6] The status is current as of 21 January 2009.
[7] See commit log at http://git.linuxtogo.org/?p=groups/mobile-linux/kernel.git;a=commit;h=b4891d61094c52304cdb184084e7278947d55eaa. Linux-To-Go GIT contains a copy of the Handhelds CVS repository with some more extra patches.
[8] *git format-patch* uses ID of the previous commit that we are making patch towards.

Description: Patch adds support for the HTC Himalaya device. It includes hardware definitions and w100fb support.

Submit address: *patches@arm.linux.org.uk*

URL: http://www.arm.linux.org.uk/developer/patches/viewpatch.php?id=5355/1

Current status[9]**:** Incoming, no reply yet.

Comments: Many useful advices about *ARM* kernel patching provided by *Marek Vašut* <*marek.vasut@gmail.com*> and *Ian Molton* <*spyro@f2s.com*>.

[9]The status is current as of 19 January 2009.

Chapter 5

Epilogue

This MSc. thesis describes *T-Mobile MDA II* in Linux, which is on the market since August 2003. The goal of the thesis was to make an analysis of the MDA II device, find suitable GNU distribution and bootloader for it, identify current status of support in Linux kernel and implement code of some components which are not currently supported.

Two GNU distributions have been found, *Familiar* and *Ångström* GNU/Linux. The first of them is quite small and compact, but there is no release since August 2006, so currently it seems to be dormant. However the second of them, Ångström, is actively developed and could be a sufficient replacement of Familiar distribution. Its build process takes many hours on personal computer and needs more than 10 GiB of free disk space. It was quite difficult to find and configure appropriate *OpenEmbedded* files to build Ångström successfully with *OABI* interface to be able to run distribution with pre-2.6.16 Linux kernels.

Booting can be done by *HaRET* or *LinExec* bootloader. HaRET is not only a bootloader but also reverse engineering tool thus it can be used for accessing the hardware internals of MDA II to give helping hand in getting Linux up and running on it. Currently the only way how to launch Linux kernel on MDA II device is using warm-boot. For cold-boot *bootldr* might be used but device is supported only partially at this time.

There are a few projects that engage in porting Linux kernel to MDA II. The first is *Xanadux*, but their last kernel is 2.6.12 and project development has ceased. On the other hand kernel has good support for MDA II components. The second project is *Handhelds* and their last kernel, 2.6.21, has support only for a few device components, however there is chance that support will be better in time. The third project is *ARM mainline* with 2.6.28 kernel. It is an ARM development branch of the *vanilla* kernel found at *kernel.org*. Currently there is no MDA II support. The build process of kernels mentioned above needed an effort to find suitable GNU toolchains because each of the kernel require specific version of the cross-compiler. Linux kernel comparison needed a lot of kernel source and output log analyses and testing experiments because there are no much infomation about these kernels available. Some information about kernels support could be found on the web however most of them are outdated or they are valid only for old Xanadux kernel branches.

For the last two kernels mentioned above, 2.6.21 and 2.6.28, code has been implemented to increase kernel support of the device. In the case of the 2.6.21, USB communication, power and battery management, touchcsreen and buttons have been made supported. In the case of the 2.6.28, device hardware definitions and LCD framebuffer support have been implemented. Code implementation in these kernels was difficult, because I have not got any experience with Linux kernel programming before. Furthermore Handhelds 2.6.21 kernel has many lines of extra code in comparison with vanilla release and no documentation or code comments are available. Therefore programming required a lot of code analyses, testing experiments and consultations with people throughout the world via IRC channel. On the other hand most of the code has been implemented in earlier Xanadux kernels which made things easier and no hardware reverse engineering was necessary to do.

Patches implementing component support have been submitted to relevant mailing lists to be merged into next kernel releases. Based on response, 2.6.21 kernel patch has been improved several times to increase its chances to be accepted shortly (there is already one successful integration). 2.6.28 patch is in incoming state and no reply has been received yet. Kernel patch submitting can occasionally take a lot of effort since kernel maintainers are very strict about coding style and code cleanliness.

The future support of T-Mobile MDA II in Linux kernel is hard to predict. In fact, this device is over five years on the market and GNU/Linux support is still insufficient to consider it to be user-friendly. Perhaps some of MDA II components used in next device generations can possibly expect better support in Linux kernel one day and my patches could help it.

MSc. thesis gave me a great opportunity to create complete GNU/Linux system that runs on MDA II device and I have gained a lot of experience with Linux kernel programming and patches submitting.

Bibliography

[1] High Tech Computer Corporation. *Wikipedia, the free encyclopedia*, [online], [cit. 2007-12-31].
URL: http://en.wikipedia.org/wiki/High_Tech_Computer_Corporation

[2] HTC Himalaya Brands Comparison Table. *Comprehensive Database of PDA, PDA Phone, PNA & Smartphone Specifications*, [online], [cit. 2007-12-31].
URL: http://pdadb.net/index.php?m=pdacomparer&id1=261&id2=69&id3=548&id4=322&id5=359&id6=68&id7=67&id8=66

[3] HTC Corp.: Service Manual for MDA2. [online], Revision 1.0 (September 12, 2003), [cit. 2007-12-31].
URL: ftp://xda:xda@ftp.xda-developers.com/Himalaya/Himalaya-Service-Manual/HTCXDAII-MDAIIServiceManual.pdf

[4] The Connectors on the Wallaby and Himalaya. *XDADeveloperWiki*, [online], [Last edited 2007-01-24], [cit. 2007-12-31].
URL: http://wiki.xda-developers.com/index.php?pagename=Connectors

[5] Himalaya Bootloader. *XDADeveloperWiki*, [online], [Last edited 2004-09-09], [cit. 2008-01-01].
URL:
http://wiki.xda-developers.com/index.php?pagename=HimalayaBootloader

[6] Himalaya Bootloader Commands. *XDADeveloperWiki*, [online], [Last edited 2005-06-19], [cit. 2008-01-01].
URL: http://wiki.xda-developers.com/index.php?pagename=HimalayaBootloaderCommands

[7] Xanadux CVS. *XDADeveloperWiki*, [online], [Last edited 2006-09-05], [cit. 2008-05-04].
URL: http://wiki.xda-developers.com/index.php?pagename=XanaduxCVS

[8] How To Compile 4 Himalaya. *XDADeveloperWiki*, [online], [Last edited 2006-07-07], [cit. 2008-05-04].
URL:
http://wiki.xda-developers.com/index.php?pagename=HowToCompile4Himalaya

[9] Himalaya Linux Booting. *XDADeveloperWiki*, [online], [Last edited 2006-02-12], [cit. 2008-01-01].
URL:
http://wiki.xda-developers.com/index.php?pagename=HimalayaLinuxBooting

[10] The Familiar Project. *Handhelds.org — Open source for handheld devices*, [online], [Last edited 2007-11-27], [cit. 2008-01-02].
URL: http://familiar.handhelds.org

[11] Target Ångström Platforms. *LinuxToGo GoWiki*, [online], [Last edited 2008-09-20], [cit. 2008-09-22].
URL: http://www.linuxtogo.org/gowiki/AboutAngstrom

[12] Building Ångström. *The Ångström Distribution*, [online], [cit. 2008-10-13].
URL: http://www.angstrom-distribution.org/building-%C3%A5ngstr%C3%B6m

[13] HaRET. *Handhelds.org MoinMoin Wiki*, [online], [Last edited 2008-06-18], [cit. 2008-06-25].
URL: http://www.handhelds.org/moin/moin.cgi/HaRET

[14] Intel Corp.: PXA26x Processor Family Developer's Manual. [online], Release 3 (January, 2004), [cit. 2008-10-15].
URL: http://int.xscale-freak.com/XSDoc/PXA26X/27863803.pdf

Appendix A

MDA II Detailed Specification

Item	**Specification**
Platform	– Microsoft Pocket PC 2003 Phone Edition – Combined GSM/GPRS and PDA – 2 logical block (PDA and GSM/GPRS) solution, layout is integrated into one module – GSM/GPRS can be turned off to let PDA to run alone
Dimensions	– 70 mm (W) × 130 mm (H) × 19 mm (D)
Weight	– Less than 200 g
Band	– Tri-band (900/1800/1900 MHz)
GPRS	– GPRS Class B – Multi-slot Class 10
Battery	– Removable rechargeable Lithium Polymer battery – 1200 mAh (Typical) – Data retention time: 72 hrs – Separated backup battery (25 mAh, rechargeable), data retention time above 0.5 hrs – Battery life: 15 hrs of PDA – Talk time: 2.5 ∼ 4 hrs – Standby: 180 hrs
AC Adapter	– AC input 100 ∼ 240 V AC, 50/60 Hz – AC input current : 0.2 A max – Output voltage : 5 V DC – Output current : 2 A (typical)
ROM RAM	– 32 MB NOR Flash (CPU embedded), 32bit data bus + M-Systems DiskOnChip 32 MB NAND Flash, 16bit data bus) – 128 MB SDRAM
Processor	– Intel XScale PXA263 CPU (Low power, High Performance, 32bit)
Clock speed	– 400 MHz
LCD Module	– 3.5” transflective TFT-LCD with back light LEDs, 240 × 320, 65536 colours – Sensitive touch screen
Graphic Controller	– MPEG4 decoder – 2D graphic accelerator

Interface	– One Infrared port IrDA SIR – One 22pin individual port for signals (for USB, Serial, Car kit, Power and Audio) – One SIM card slot – One SDIO/MMC card slot (B-Square SDIO driver) – One 34pin Back pack connector – One Audio Jack (2.5)
Stylus	– Lock type mechanism
Keyboard/ Button/ Switch	– One five-way navigation button (includes *action* key) – One Power button (wake-up key) – One volume control button (up & down) – Two phone button, SEND (Yes) & END (No), (Wake-up keys), with LED backlight – Two programmable AP buttons (Wake up key) – Camera (Side button) – Voice command/Voice recorder (side button) – Reset Switch – Key lock function support by software
Notification	– One Bi-color LED for GSM standby, GSM message, GSM network status, PDA notification, PDA Charging status. – One Blue LED for Bluetooth system notification of powered -ON and ready to transmit RF signal. – Notification by sound, Message, Vibration on the display.
Audio	– Microphone build in – Software Echo cancellation – Receiver – Hardware AGC – Hardware Full duplex – WAV/WMA/MP3 stereo – 16bits with 8 kHz, 11 kHz, 16 kHz, 22 kHz, 44.1 kHz, 48 kHz sampling rate
CMOS Camera	– Color CMOS camera module – VGA (480 $\times$ 640) resolution with JPEG encoder – ALC (Auto Light Control) – AWB (Auto White Balance) – Preview Mirror on battery cover – Camcorder QCIF
Crandle	– Cradle connector connected to main unit – Audio jack (2.5) – Microphone – 2 slots, one for main unit + battery, another is capable of charging 2nd battery – Serial or USB cable between cradle and PC – LED indicator for 2nd battery charging
Bluetooth	– Bluetooth 1.1 compliant – Power class 2 – Support Profile

Appendix B

GNU/Linux Outputs

B.1 2.6.3-rmk0-hh0-xda0 dmesg

```
Linux version 2.6.3-rmk0-hh0-xda0 (maik@dracula) (gcc version 3.3.2) #1
  Wed Oct 1 18:02:28 CEST 2008
CPU: XScale-PXA255 [69052d06] revision 6 (ARMv5TE)
CPU: D undefined 5 cache
CPU: I cache: 32768 bytes, associativity 32, 32 byte lines, 32 sets
CPU: D cache: 32768 bytes, associativity 32, 32 byte lines, 32 sets
Machine: HTC Himalaya
Memory policy: ECC disabled, Data cache writeback
Memory clock: 99.53MHz (*27)
Run Mode clock: 398.13MHz (*4)
Turbo Mode clock: 398.13MHz (*1.0, inactive)
On node 0 totalpages: 32768
  DMA zone: 32768 pages, LIFO batch:8
  Normal zone: 0 pages, LIFO batch:1
  HighMem zone: 0 pages, LIFO batch:1
Built 1 zonelists
Kernel command line: root=/dev/ram0 init=/linuxrc ramdisk_size=14336 keepinitrd
Relocating machine vectors to 0xffff0000
IRQ27 (GPIO4): rising falling edges
PID hash table entries: 1024 (order 10: 8192 bytes)
Console: colour dummy device 80x30
Memory: 128MB = 128MB total
Memory: 123296KB available (1295K code, 365K data, 64K init)
Calibrating delay loop... 397.31 BogoMIPS
Dentry cache hash table entries: 16384 (order: 4, 65536 bytes)
Inode-cache hash table entries: 8192 (order: 3, 32768 bytes)
Mount-cache hash table entries: 512 (order: 0, 4096 bytes)
checking if image is initramfs...it isn't (no cpio magic); looks like an initrd
CPU: Testing write buffer coherency: ok
POSIX conformance testing by UNIFIX
NET: Registered protocol family 16
himalayafb: register driver
pxafb: request mem region
```

```
pxafb: deferring startup
get_machine_info: didn't get himalayafb mach info
pxafb: deferring startup
Checking ASIC3 idstring: HTC-SDIO P/N:30H80028-00
lcd_module_register: name=himalayafb
pxafb: deferring startup
get_machine_info: got himalayafb mach info
init_fbinfo
inf=c016d5bc inf->xres=240 inf->yres=320 inf->bpp=16
himalayafb: framebuffer at 0x8100000, mapped to 0xc8803000, size 150k
NetWinder Floating Point Emulator V0.97 (double precision)
devfs: v1.22 (20021013) Richard Gooch (rgooch@atnf.csiro.au)
devfs: devfs_debug: 0x0
devfs: boot_options: 0x1
Initializing Cryptographic API
Console: switching to colour frame buffer device 60x53
pty: 256 Unix98 ptys configured
ttyS0 at MMIO 0x40100000 (irq = 15) is a FFUART
ttyS1 at MMIO 0x40200000 (irq = 14) is a BTUART
ttyS2 at MMIO 0x40700000 (irq = 13) is a STUART
Console: switching to colour frame buffer device 60x53
RAMDISK driver initialized: 16 RAM disks of 14336K size 1024 blocksize
mice: PS/2 mouse device common for all mice
NET: Registered protocol family 2
Enabling LCD
MODULE: Power on!
himalaya LCD power on
IP: routing cache hash table of 1024 buckets, 8Kbytes
TCP: Hash tables configured (established 8192 bind 16384)
NET: Registered protocol family 1
LCD enable
MODULE: Enable on!
himalaya_lcd_set_enable
RAMDISK: Compressed image found at block 0
VFS: Mounted root (ext2 filesystem).
Mounted devfs on /dev
Freeing init memory: 64K
pxa2xx_udc: version 14-Dec-2003
_udc_control: disconnect
usb0: Ethernet Gadget, pxa2xx, version: Bastille Day 2003
_udc_control: connect
udc: USB reset
udc_is_connected: request returns 1
udc: USB reset
udc_is_connected: request returns 1
udc: USB reset
udc: USB reset
usb0: full speed config #3: Ethernet Gadget
```

```
himalaya_ts: module license 'unspecified' taints kernel.
H3600 Registering HAL abstraction layer
h3600_ts_init_module: registering char device
h3600_ts_reset_filters
h3600_ts_touchpanel_event: x=0 y=0 down=0
Using anticipatory io scheduler
Battery driver v0.1
himalaya_leds: module license 'unspecified' taints kernel.
udc_is_connected: request returns 1
udc_is_connected: request returns 1
```

B.2 2.6.12-hh2-xda0-ba0 dmesg

```
Linux version 2.6.12-hh2-xda0-ba0 (maik@dracula) (gcc version 3.4.1) #2
  Sun Jul 20 16:44:24 CEST 2008
CPU: XScale-PXA255 [69052d06] revision 6 (ARMv5TE)
CPU0: D VIVT undefined 5 cache
CPU0: I cache: 32768 bytes, associativity 32, 32 byte lines, 32 sets
CPU0: D cache: 32768 bytes, associativity 32, 32 byte lines, 32 sets
Machine: HTC Himalaya
Memory policy: ECC disabled, Data cache writeback
Memory clock: 99.53MHz (*27)
Run Mode clock: 398.13MHz (*4)
Turbo Mode clock: 398.13MHz (*1.0, inactive)
Himalaya Board ID 0x4
On node 0 totalpages: 32768
  DMA zone: 32768 pages, LIFO batch:15
  Normal zone: 0 pages, LIFO batch:1
  HighMem zone: 0 pages, LIFO batch:1
Built 1 zonelists
Kernel command line: root=/dev/ram0 init=/linuxrc ramdisk_size=14336 keepinitrd
PID hash table entries: 1024 (order: 10, 16384 bytes)
Console: colour dummy device 80x30
Dentry cache hash table entries: 32768 (order: 5, 131072 bytes)
Inode-cache hash table entries: 16384 (order: 4, 65536 bytes)
Memory: 128MB = 128MB total
Memory: 121984KB available (1763K code, 353K data, 88K init)
Calibrating delay loop... 397.31 BogoMIPS (lpj=1986560)
Mount-cache hash table entries: 512
CPU: Testing write buffer coherency: ok
checking if image is initramfs...it isn't (no cpio magic); looks like an initrd
NET: Registered protocol family 16
platform_device_register asic3
platform_device_register pxa2xx-mci
platform_device_register pxa2xx-udc
platform_device_register pxa2xx-fb
platform_device_register pxa2xx-uart
platform_device_register pxa2xx-uart
```

```
platform_device_register pxa2xx-uart
platform_device_register pxa2xx-i2c
platform_device_register pxa2xx-uart
asic3: using irq 104-173 on irq 31
Register NSSP bus ...  done.
tsc2200_init: 10F/80.
Register TSC2200 driver ...  done.
tsc2200_init: 10F/80.
nssp_bus_match: ts_tsc2200 -> tsc2200
tsc2200_probe: initializing NSSP.
tsc2200_probe: initializing the tsc2200.
tsc2200_reset: 4000.
tsc2200_reset: 4000.
tsc2200_probe: SPI: cr0 0000098f cr1 60000010 sr: 0000f024 it: 00000000
  to: 00000000 ps: 00000000
nssp_bus_match: kp_tsc2200 -> tsc2200
tsc2200_probe: initializing NSSP.
tsc2200_probe: initializing the tsc2200.
tsc2200_reset: 4000.
tsc2200_reset: 4000.
tsc2200_probe: SPI: cr0 0000098f cr1 60000010 sr: 0000f024 it: 00000000
  to: 00000000 ps: 00000000
himalaya_lcd_init
platform_device_register w100fb
NetWinder Floating Point Emulator V0.97 (double precision)
devfs: 2004-01-31 Richard Gooch (rgooch@atnf.csiro.au)
devfs: boot_options: 0x1
Found w3200 at 0x08000000.
Console: switching to colour frame buffer device 60x53
fb0: w100fb frame buffer device
platform_device_register sa-pxa-rtc
SA1100 Real Time Clock driver v1.03
SA1100/PXA2xx Watchdog Timer: timer margin 60 sec
ttyS0 at MMIO 0x40100000 (irq = 15) is a FFUART
ttyS1 at MMIO 0x40200000 (irq = 14) is a BTUART
ttyS2 at MMIO 0x40700000 (irq = 13) is a STUART
ttyS3 at MMIO 0x41600000 (irq = 0) is a HWUART
io scheduler noop registered
io scheduler anticipatory registered
io scheduler deadline registered
io scheduler cfq registered
RAMDISK driver initialized: 16 RAM disks of 14336K size 1024 blocksize
loop: loaded (max 8 devices)
tun: Universal TUN/TAP device driver, 1.6
tun: (C) 1999-2004 Max Krasnyansky <maxk@qualcomm.com>
mice: PS/2 mouse device common for all mice
ts: Compaq touchscreen protocol output
evbug.c: Connected device: "tsc2200_ts", touchscreen/tsc2200
```

```
evbug.c: Connected device: "himalayakb", keyboard/himalayakb
NET: Registered protocol family 2
IP: routing cache hash table of 1024 buckets, 8Kbytes
TCP established hash table entries: 8192 (order: 4, 65536 bytes)
TCP bind hash table entries: 8192 (order: 3, 32768 bytes)
TCP: Hash tables configured (established 8192 bind 8192)
NET: Registered protocol family 1
NET: Registered protocol family 17
RAMDISK: Compressed image found at block 0
VFS: Mounted root (ext2 filesystem).
Mounted devfs on /dev
Freeing init memory: 88K
pxa2xx_udc: version 14-Dec-2003
_udc_control: disconnect
udc_is_connected: request returns 1
usb0: Ethernet Gadget, version: Equinox 2004
usb0: using pxa2xx_udc, OUT ep2out-bulk IN ep1in-bulk
usb0: MAC fe:f4:b0:52:a7:d3
_udc_control: connect
udc_is_connected: request returns 1
udc_is_connected: request returns 1
udc_is_connected: request returns 1
udc_is_connected: request returns 1
usb0: full speed config #1: 100 mA, Ethernet Gadget, using CDC Ethernet Subset
Probing PXA Flash 1 at physical address 0x00000000 (32-bit bankwidth)
PXA Flash 1: Found 2 x16 devices at 0x0 in 32-bit bank
 Intel/Sharp Extended Query Table at 0x0031
Using buffer write method
cfi_cmdset_0001: Erase suspend on write enabled
0: offset=0x0,size=0x40000,blocks=128
mymtd=c7c6fe00
mtd: Giving out device 0 to PXA Flash 1
himalaya_battery_get_voltage: in.
1: 2CA4
himalaya_battery_get_voltage: BAT1: 4FF
BAT2: 0
AUX1: fff
AUX2: fff
TEMP1: 0
TEMP2: 0
retval=4095
himalaya_battery_get_temp: in.
retval=0x0
himalaya_battery_get_temp: BAT1: 4FF
BAT2: 000
AUX1: fff
AUX2: fff
TEMP1: 3ec
```

```
TEMP2: 000
evbug.c: Event. Dev: touchscreen/tsc2200, Type: 0, Code: 0, Value: 0
platform_device_register himalaya_led
himalaya_set_led
himalaya_set_led
himalaya_set_led
Registered led device: number=red, color=red
Registered led device: number=green, color=green
Registered led device: number=yellow, color=yellow
Registered led device: number=phone, color=<NULL>
Registered led device: number=vibra, color=<NULL>
No valid DiskOnChip devices found
```

B.3 2.6.21-devel-orig dmesg

```
Linux version 2.6.21-devel-orig (maik@dracula) (gcc version 4.1.3 20080623
  (prerelease) (Debian 4.1.2-23)) #1 Wed Jan 7 23:34:20 CET 2009
CPU: XScale-PXA255 [69052d06] revision 6 (ARMv5TE), cr=0000397f
Machine: HTC Himalaya
Memory policy: ECC disabled, Data cache writeback
On node 0 totalpages: 32768
  DMA zone: 256 pages used for memmap
  DMA zone: 0 pages reserved
  DMA zone: 32512 pages, LIFO batch:7
  Normal zone: 0 pages used for memmap
Memory clock: 99.53MHz (*27)
Run Mode clock: 398.13MHz (*4)
Turbo Mode clock: 398.13MHz (*1.0, inactive)
CPU0: D VIVT undefined 5 cache
CPU0: I cache: 32768 bytes, associativity 32, 32 byte lines, 32 sets
CPU0: D cache: 32768 bytes, associativity 32, 32 byte lines, 32 sets
Built 1 zonelists.  Total pages: 32512
Kernel command line: root=/dev/mmcblk0p2 rootdelay=1 console=ttyS0,115200n8
  console=tty0 psplash=false
PID hash table entries: 512 (order: 9, 2048 bytes)
Console: colour dummy device 80x30
Dentry cache hash table entries: 16384 (order: 4, 65536 bytes)
Inode-cache hash table entries: 8192 (order: 3, 32768 bytes)
Memory: 128MB = 128MB total
Memory: 127104KB available (2408K code, 177K data, 100K init)
Calibrating delay loop... 397.31 BogoMIPS (lpj=1986560)
Mount-cache hash table entries: 512
CPU: Testing write buffer coherency: ok
NET: Registered protocol family 16
asic3: using irq 108-177 on irq 31
irda_init()
NET: Registered protocol family 23
Time: pxa_timer clocksource has been installed.
```

```
NET: Registered protocol family 2
IP route cache hash table entries: 1024 (order: 0, 4096 bytes)
TCP established hash table entries: 4096 (order: 3, 32768 bytes)
TCP bind hash table entries: 4096 (order: 2, 16384 bytes)
TCP: Hash tables configured (established 4096 bind 4096)
TCP reno registered
himalaya udc registerhimalaya LCD Driver init. boardid=4
Himalaya Board ID 0x4
NetWinder Floating Point Emulator V0.97 (double precision)
JFFS2 version 2.2. (NAND) (C) 2001-2006 Red Hat, Inc.
io scheduler noop registered
io scheduler anticipatory registered (default)
io scheduler deadline registered
Found w3200 at 0x08000000.
Console: switching to colour frame buffer device 60x53
fb0: w100fb frame buffer device
pxa2xx-uart.0: ttyS0 at MMIO 0x40100000 (irq = 15) is a FFUART
pxa2xx-uart.1: ttyS1 at MMIO 0x40200000 (irq = 14) is a BTUART
pxa2xx-uart.2: ttyS2 at MMIO 0x40700000 (irq = 13) is a STUART
pxa2xx-uart.3: ttyS3 at MMIO 0x41600000 (irq = 0) is a HWUART
RAMDISK driver initialized: 16 RAM disks of 4096K size 1024 blocksize
loop: loaded (max 8 devices)
NFTL driver: nftlcore.c $Revision: 1.98 $, nftlmount.c $Revision: 1.41 $
INFTL: inftlcore.c $Revision: 1.19 $, inftlmount.c $Revision: 1.18 $
No valid DiskOnChip devices found
Registering tsc2200 buttons driver
i2c /dev entries driver
I2C: i2c-0: PXA I2C adapter
TCP cubic registered
NET: Registered protocol family 1
XScale DSP coprocessor detected.
asic3_mmc: ASIC3 MMC/SD Driver, controller at 0xc800000
Waiting 1sec before mounting root device...
mmcblk0: mmc0:b368 SD    993792KiB
 mmcblk0: p1 p2
kjournald starting.  Commit interval 5 seconds
EXT3 FS on mmcblk0p2, internal journal
EXT3-fs: mounted filesystem with ordered data mode.
VFS: Mounted root (ext3 filesystem).
Freeing init memory: 100K
pxa2xx_udc: version 4-May-2005
_udc_control: disconnect
ether gadget: using random self ethernet address
ether gadget: using random host ethernet address
usb0: Ethernet Gadget, version: May Day 2005
usb0: using pxa2xx_udc, OUT ep2out-bulk IN ep1in-bulk
usb0: MAC 92:7b:a1:bd:d8:bd
_udc_control: connect
```

```
NET: Registered protocol family 10
ADDRCONF(NETDEV_UP): usb0: link is not ready
udc_is_connected returns 1
udc_is_connected returns 1
```

B.4 2.6.21-devel-my dmesg

```
Linux version 2.6.21-devel-my (maik@dracula) (gcc version 4.1.3 20080623
  (prerelease) (Debian 4.1.2-23)) #1 Wed Jan 7 23:34:03 CET 2009
CPU: XScale-PXA255 [69052d06] revision 6 (ARMv5TE), cr=0000397f
Machine: HTC Himalaya
Memory policy: ECC disabled, Data cache writeback
On node 0 totalpages: 32768
  DMA zone: 256 pages used for memmap
  DMA zone: 0 pages reserved
  DMA zone: 32512 pages, LIFO batch:7
  Normal zone: 0 pages used for memmap
Memory clock: 99.53MHz (*27)
Run Mode clock: 398.13MHz (*4)
Turbo Mode clock: 398.13MHz (*1.0, inactive)
CPU0: D VIVT undefined 5 cache
CPU0: I cache: 32768 bytes, associativity 32, 32 byte lines, 32 sets
CPU0: D cache: 32768 bytes, associativity 32, 32 byte lines, 32 sets
Built 1 zonelists.  Total pages: 32512
Kernel command line: root=/dev/mmcblk0p2 rootdelay=1 console=ttyS0,115200n8
  console=tty0 psplash=false
PID hash table entries: 512 (order: 9, 2048 bytes)
Console: colour dummy device 80x30
Dentry cache hash table entries: 16384 (order: 4, 65536 bytes)
Inode-cache hash table entries: 8192 (order: 3, 32768 bytes)
Memory: 128MB = 128MB total
Memory: 127104KB available (2408K code, 178K data, 100K init)
Calibrating delay loop... 397.31 BogoMIPS (lpj=1986560)
Mount-cache hash table entries: 512
CPU: Testing write buffer coherency: ok
NET: Registered protocol family 16
asic3: using irq 108-177 on irq 31
Initializing himalaya tsc2200 ssp... done.
tsc2200_probe: SPI: cr0 0000138f cr1 60000010 sr: 0000f024 it: 00000000
  to: 00000000 ps: 00000000
irda_init()
NET: Registered protocol family 23
Time: pxa_timer clocksource has been installed.
NET: Registered protocol family 2
IP route cache hash table entries: 1024 (order: 0, 4096 bytes)
TCP established hash table entries: 4096 (order: 3, 32768 bytes)
TCP bind hash table entries: 4096 (order: 2, 16384 bytes)
TCP: Hash tables configured (established 4096 bind 4096)
```

```
TCP reno registered
himalaya udc register
himalaya LCD Driver init. boardid=4
Himalaya Board ID 0x4
NetWinder Floating Point Emulator V0.97 (double precision)
JFFS2 version 2.2. (NAND) (C) 2001-2006 Red Hat, Inc.
io scheduler noop registered
io scheduler anticipatory registered (default)
io scheduler deadline registered
Found w3200 at 0x08000000.
Console: switching to colour frame buffer device 60x53
fb0: w100fb frame buffer device
pxa2xx-uart.0: ttyS0 at MMIO 0x40100000 (irq = 15) is a FFUART
pxa2xx-uart.1: ttyS1 at MMIO 0x40200000 (irq = 14) is a BTUART
pxa2xx-uart.2: ttyS2 at MMIO 0x40700000 (irq = 13) is a STUART
pxa2xx-uart.3: ttyS3 at MMIO 0x41600000 (irq = 0) is a HWUART
RAMDISK driver initialized: 16 RAM disks of 4096K size 1024 blocksize
loop: loaded (max 8 devices)
NFTL driver: nftlcore.c $Revision: 1.98 $, nftlmount.c $Revision: 1.41 $
INFTL: inftlcore.c $Revision: 1.19 $, inftlmount.c $Revision: 1.18 $
No valid DiskOnChip devices found
Registering tsc2200 buttons driver
input: tsc2200-keys as /class/input/input0
evbug.c: Connected device: "tsc2200-keys", <NULL>
tsc2200_ts_probe IRQ: 28
input: tsc2200-ts as /class/input/input1
evbug.c: Connected device: "tsc2200-ts", touchscreen/tsc2200-ts
i2c /dev entries driver
I2C: i2c-0: PXA I2C adapter
TCP cubic registered
NET: Registered protocol family 1
XScale DSP coprocessor detected.
asic3_mmc: ASIC3 MMC/SD Driver, controller at 0xc800000
Waiting 1sec before mounting root device...
mmcblk0: mmc0:b368 SD    993792KiB
 mmcblk0: p1 p2
kjournald starting.  Commit interval 5 seconds
EXT3 FS on mmcblk0p2, internal journal
EXT3-fs: mounted filesystem with ordered data mode.
VFS: Mounted root (ext3 filesystem).
Freeing init memory: 100K
pxa2xx_udc: version 4-May-2005
_udc_control: disconnect
ether gadget: using random self ethernet address
ether gadget: using random host ethernet address
usb0: Ethernet Gadget, version: May Day 2005
usb0: using pxa2xx_udc, OUT ep2out-bulk IN ep1in-bulk
usb0: MAC 5e:3f:96:87:52:52
```

```
_udc_control: connect
udc_is_connected returns 1
udc_is_connected returns 1
pda-power pda-power: charger on (USB)
himalaya_set_charge: enabled=2
 ac: uevent
 ac: No power supply yet
 ac: power_supply_changed
 usb: uevent
 usb: No power supply yet
 usb: power_supply_changed
 ac: power_supply_changed_work
 ac: power_supply_update_gen_leds 0
 ac: uevent
 ac: POWER_SUPPLY_NAME=ac
 ac: Static prop TYPE=Mains
 ac: 1 dynamic props
 ac: prop ONLINE=0
 usb: power_supply_changed_work
 usb: power_supply_update_gen_leds 1
 usb: uevent
 usb: POWER_SUPPLY_NAME=usb
 usb: Static prop TYPE=USB
 usb: 1 dynamic props
 usb: prop ONLINE=1
udc_is_connected returns 1
udc_is_connected returns 1
usb0: full speed config #1: 100 mA, Ethernet Gadget, using CDC Ethernet Subset
usb0: full speed config #1: 100 mA, Ethernet Gadget, using CDC Ethernet Subset
NET: Registered protocol family 10
udc_is_connected returns 1
udc_is_connected returns 1
```