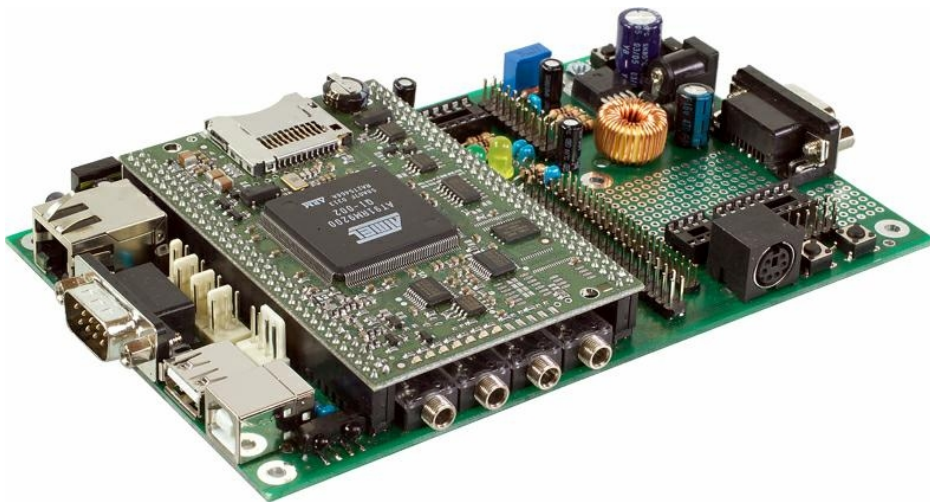




# CentiPad Linux Handbook (English)



Author: Marcus Hasenstab  
Version: 1.01  
Date: 0704112205  
[www.centipad.de](http://www.centipad.de)

Notes:

Änderungsliste:

- 070411 first English draft

# 1. Inhaltsverzeichnis

<b>1. Inhaltsverzeichnis .....</b>	<b>3</b>
<b>2. Introduction.....</b>	<b>8</b>
<b>2.1. Preface to the English version .....</b>	<b>8</b>
<b>2.2. Focus .....</b>	<b>8</b>
<b>2.3. additionaldocumentation .....</b>	<b>8</b>
<b>2.4. Safetyguidelines.....</b>	<b>8</b>
<b>2.5. Life support policy.....</b>	<b>9</b>
<b>2.6. Protection against electrostatic discharge (ESD) .....</b>	<b>9</b>
<b>2.7. operationalsafety .....</b>	<b>9</b>
<b>2.8. Warrenty regulations / product liability.....</b>	<b>9</b>
<b>2.9. Legend .....</b>	<b>9</b>
<b>2.10. Legal stuff.....</b>	<b>10</b>
<b>3. First steps.....</b>	<b>11</b>
<b>3.1. Neue Welten .....</b>	<b>11</b>
<b>3.2. Hardwareconfiguration .....</b>	<b>11</b>
<b>4. Setup on Linux .....</b>	<b>13</b>
4.1.1. Requirements for installation on Suse .....	13
4.1.2. First installation .....	13
4.1.3. System copy / Backup.....	13
4.1.4. directory structure / development system upgrade.....	13
4.1.5. Backup Strategien unter Linux.....	14
<b>4.2. Setup under Windows with coLinux .....</b>	<b>16</b>
4.2.1. coLinux.....	16
4.2.2. Standard User.....	16
4.2.3. Installation.....	16
4.2.4. Backup strategies for Windows/coLinux.....	20
4.2.5. Hyperterminal.....	20
4.2.6. Cygwin .....	22
4.2.6.1. CygwinX Setup.....	22
4.2.6.2. Cygwin for CentiPad .....	22
<b>5. the maintech Bootloader .....</b>	<b>24</b>
<b>5.1. Overview .....</b>	<b>24</b>
<b>5.2. Factoryconfiguration .....</b>	<b>24</b>
<b>5.3. Menustructure.....</b>	<b>25</b>
<b>5.4. Boot Menu .....</b>	<b>26</b>
<b>5.5. Boot Options.....</b>	<b>26</b>
5.5.1. Booten über TFTP konfigurieren .....	26
5.5.2. Config for DFU boot .....	27
5.5.3. boot from SDcard .....	27
5.5.4. reconfig Bootloader.....	27
5.5.5. Setup of a new board.....	27
5.5.5.1. on Linux .....	27
5.5.5.2. on Windows .....	28
5.5.6. Bootfrom Dataflash.....	29
5.5.7. Setup bootloader for dataflash.....	29

<b>6.</b>	<b><i>CentiTools</i></b> .....	<b>31</b>
<b>6.1.</b>	<b>cbupdate</b> .....	<b>31</b>
<b>6.2.</b>	<b>ethboot</b> .....	<b>31</b>
<b>6.3.</b>	<b>genext2fs</b> .....	<b>31</b>
<b>6.4.</b>	<b>ipkg</b> .....	<b>31</b>
<b>6.5.</b>	<b>tools</b> .....	<b>31</b>
<b>6.6.</b>	<b>usbdfu</b> .....	<b>31</b>
<b>7.</b>	<b><i>development tools</i></b> .....	<b>32</b>
<b>7.1.</b>	<b>GCC</b> .....	<b>32</b>
<b>7.2.</b>	<b>Make</b> .....	<b>32</b>
<b>7.3.</b>	<b>Debugging</b> .....	<b>32</b>
7.3.1.	Host development .....	32
7.3.2.	Printf-Debugging .....	32
7.3.3.	GDB.....	32
<b>8.</b>	<b><i>Development cycle on Linux</i></b> .....	<b>34</b>
<b>8.1.</b>	<b>Building the whole CentiPad package – make world</b> .....	<b>34</b>
8.1.1.	centidev/make .....	34
8.1.1.1.	make world.....	34
8.1.1.2.	make boot .....	34
8.1.1.3.	make usbboot .....	34
8.1.1.4.	make clean .....	34
8.1.1.5.	make distclean .....	34
8.1.1.6.	make app image.....	35
8.1.1.7.	make kernel image .....	35
8.1.1.8.	make bootupdate.....	35
8.1.1.9.	make run.....	35
8.1.1.10.	make debug.....	35
8.1.1.11.	/centidev/make menuconfig .....	35
8.1.1.12.	CentiDev driver setup /centidev/platform/linux/make menuconfig .....	35
<b>8.2.</b>	<b>packages – including your own software into CentiDev</b> .....	<b>36</b>
<b>8.3.</b>	<b>Uploading software to the target system</b> .....	<b>37</b>
8.3.1.1.	Laden des Zielsystem-RAM über TFTP – make world boot .....	37
8.3.1.2.	Laden des Zielsystems über SDcard am Hostrechner.....	37
8.3.1.3.	starting from SDcard in CentiPad .....	37
8.3.1.4.	Upload to Permanent Storage Device .....	38
8.3.1.5.	Laden des Zielsystem-RAM über USB - make world usbboot.....	38
<b>8.4.</b>	<b>User program development</b> .....	<b>39</b>
8.4.1.	Overview .....	39
8.4.2.	including your programs into the build system.....	39
8.4.3.	„Hello World“ on the CentiPad.....	39
8.4.4.	Testing the user program „Hello World“ .....	41
8.4.5.	Integrating application-files into image-files.....	41
<b>9.</b>	<b><i>development cycle on coLinux</i></b> .....	<b>43</b>
<b>9.1.</b>	<b>introduction</b> .....	<b>43</b>
<b>9.2.</b>	<b>Build and upload of a CentiPad Linux</b> .....	<b>44</b>
9.2.1.	image build .....	44
9.2.2.	data transfer to target system .....	44
9.2.3.	copy to bootloader mass storage.....	44
<b>9.3.</b>	<b>coLinuxshutdown</b> .....	<b>45</b>
<b>9.4.</b>	<b>build and upload of programs to CentiPad</b> .....	<b>45</b>

<b>10.</b>	<b>Introduction to (Embedded) Linux .....</b>	<b>46</b>
<b>10.1.</b>	<b>LinuxÜberblick .....</b>	<b>46</b>
<b>10.2.</b>	<b>CentiPad Standard User .....</b>	<b>46</b>
<b>10.3.</b>	<b>Kleine Auswahl von Linux Kommandos .....</b>	<b>46</b>
10.3.1.	Konsole .....	46
10.3.1.1.	Tastatursteuerung der Konsole.....	46
10.3.1.2.	Standard Eingabe/Ausgabe.....	47
10.3.1.3.	Piping.....	47
10.3.1.4.	which – welches Programm würde ausgeführt werden? .....	47
10.3.2.	man, info, xman.....	47
10.3.3.	Anzeigebefehle .....	47
10.3.3.1.	cat – zeige Daten an .....	47
10.3.3.2.	more/less – Konsolenausgabe komfortabel .....	47
10.3.3.3.	tail – Ende einer Datei anschauen.....	48
10.3.4.	Verzeichnis- und Dateibefehle.....	48
10.3.4.1.	cd – Verzeichniswechsel .....	48
10.3.4.2.	pwd – Print Work Directory .....	48
10.3.4.3.	ls – List Directory.....	48
10.3.4.4.	md, mkdir – Make Directory.....	48
10.3.4.5.	mv – move file or directory .....	48
10.3.4.6.	rm .....	48
10.3.4.7.	touch .....	48
10.3.4.8.	cp – Dateien kopieren .....	48
10.3.4.9.	tar – Backup und Archive.....	49
10.3.4.10.	chmod - Change file access permissions.....	49
10.3.4.11.	find - Suche Dateien im Verzeichnisbaum:.....	49
10.3.4.12.	grep – Dateien mit definiertem Inhalt suchen.....	49
10.3.4.13.	diff – Unterschiede zwischen Dateiversionen finden .....	49
10.3.4.14.	xargs.....	50
10.3.4.15.	md5sum – Datenkonsistenz überprüfen.....	50
10.3.5.	Datenfernübertragung.....	50
10.3.5.1.	telnetd .....	50
10.3.5.2.	rx – Empfange Xmodem Datei.....	50
10.3.5.3.	minicom .....	50
10.3.5.4.	xterm / konsole.....	51
10.3.6.	Systeminformation .....	51
10.3.6.1.	ps – Prozessliste anschauen .....	51
10.3.6.2.	kill/killall – Prozess stoppen .....	51
10.3.6.3.	top – Systembelastung anzeigen lassen.....	51
10.3.6.4.	free, df, du - Speicherplatz ermitteln.....	51
10.3.6.5.	proc/bootinfo .....	51
10.3.7.	tty – Aktive Konsolen Schnittstelle ausgeben.....	51
10.3.8.	date – Datum und Uhrzeit setzen .....	51
10.3.9.	time – Programmlaufzeit messen .....	51
10.3.10.	hexdump – Datei als Hexdump anzeigen .....	52
10.3.11.	uptime – Betriebsdauer .....	52
10.3.12.	sudo – Programm als anderer User ausführen .....	52
10.3.13.	watch – Programm periodisch ausführen .....	52
<b>10.4.</b>	<b>Structure of the CentiPadLinux .....</b>	<b>53</b>
10.4.1.	directories.....	53
10.4.1.1.	dev directory.....	53
10.4.1.2.	mnt/storage.....	54
10.4.1.3.	bin.....	54
10.4.1.4.	var.....	54
10.4.1.5.	lib.....	54
10.4.2.	environment variables.....	55
10.4.2.1.	using environment variables .....	55
10.4.2.2.	Important environment variables.....	55
10.4.2.3.	system specific files .....	55
10.4.3.	BusyBox runtime environment.....	56
10.4.3.1.	Beschreibung .....	56
10.4.3.2.	Busybox konfigurieren.....	56
10.4.4.	sysfiles / Konfigurations- und Steuerungsdateien.....	57
10.4.5.	Kernelkonfiguration .....	57
10.4.6.	Board Konfiguration.....	57

<b>10.5.</b>	<b>Dateisysteme .....</b>	<b>59</b>
10.5.1.	Everything is a file.....	59
10.5.2.	Mounten von Filesystemen.....	59
10.5.2.1.	Allgemein .....	59
10.5.2.2.	Mounten eines USB-Sticks .....	59
10.5.2.3.	Mounten des Datenbereichs des CentiPad Dataflashes.....	59
10.5.2.4.	Mounten einer SDCARD .....	59
10.5.2.5.	Mounten einer CF-CARD .....	60
10.5.2.6.	Mounten eines unter Windows freigegebenen Verzeichnisses.....	60
10.5.2.7.	Probleme beim automatischen Mounten.....	61
10.5.3.	Image-Dateien verwenden.....	61
10.5.3.1.	Mounten einer Image-Datei.....	61
10.5.3.2.	Kopieren von Image-Filesystemen .....	61
10.5.4.	Disk Exercizer.....	61
10.5.5.	Partitionieren eines Speichermediums.....	62
<b>10.6.</b>	<b>Networking.....</b>	<b>63</b>
10.6.1.	General .....	63
10.6.2.	MAC-address.....	63
10.6.3.	Network configuration and start.....	63
10.6.4.	Standard user / Password.....	63
10.6.5.	Address search / set .....	63
10.6.6.	additional users on CentiPad .....	63
10.6.7.	ssh – Secure Shell.....	64
10.6.7.1.	RSA-Schlüsselliste löschen.....	64
10.6.8.	scp – Secure Copy.....	64
10.6.9.	fish / winscp– Fenstergestützter Zugriff auf Dateien .....	64
10.6.10.	telnet .....	64
10.6.11.	Samba directories .....	65
10.6.12.	wget – file transfer from HTTP/FTP-server .....	65
10.6.13.	nc - netcat .....	65
<b>11.</b>	<b><i>The CentiPad Shell ASH.....</i></b>	<b>66</b>
<b>11.1.</b>	<b>Code Table .....</b>	<b>66</b>
<b>11.2.</b>	<b>Shellskripteerstellen .....</b>	<b>66</b>
<b>11.3.</b>	<b>Shellskriptedebuggen .....</b>	<b>66</b>
<b>11.4.</b>	<b>Kommandosubstitution .....</b>	<b>66</b>
<b>11.5.</b>	<b>Gültigkeitsbereiche von Variablen.....</b>	<b>66</b>
<b>11.6.</b>	<b>Programme in der selben Shell starten .....</b>	<b>66</b>
<b>11.7.</b>	<b>Beispiel: backup.sh .....</b>	<b>66</b>
<b>11.8.</b>	<b>Editieren von Shellskripten.....</b>	<b>67</b>
<b>11.9.</b>	<b>.....</b>	<b>67</b>
<b>12.</b>	<b><i>CentiPad Linux Hilfsprogramme .....</i></b>	<b>68</b>
<b>12.1.</b>	<b>picocom .....</b>	<b>68</b>
<b>13.</b>	<b><i>Linux Goodies.....</i></b>	<b>68</b>
<b>13.1.</b>	<b>Editoren für Programmierer auf dem SuSe-Linux System.....</b>	<b>68</b>
<b>13.2.</b>	<b>apropos – man-Dateien zum Thema .....</b>	<b>68</b>
<b>13.3.</b>	<b>ksnapshot – Screenshot .....</b>	<b>68</b>
<b>13.4.</b>	<b>ksteak .....</b>	<b>68</b>
<b>13.5.</b>	<b>konserve – automatisches Backup .....</b>	<b>69</b>
<b>13.6.</b>	<b>Nachrichten an angemeldete Userschicken .....</b>	<b>69</b>

<b>13.7.</b>	<b>tthttpd – ein kleiner Webserver für das Hostsystem .....</b>	<b>69</b>
<b>13.8.</b>	<b>mbuffer – Puffer zur Datentransferoptimierung bei schnellen Laufwerken.....</b>	<b>69</b>
<b>13.9.</b>	<b>Source management.....</b>	<b>69</b>
<b>13.10.</b>	<b>direct access to the maintech Source-Code-Repository.....</b>	<b>70</b>
<b>13.11.</b>	<b>Disassembler.....</b>	<b>70</b>
<b>14.</b>	<b><i>Literaturverzeichnis .....</i></b>	<b>71</b>
<b>14.1.</b>	<b>Programmieren .....</b>	<b>71</b>
<b>14.2.</b>	<b>Linux Treiber Entwicklung.....</b>	<b>71</b>
<b>14.3.</b>	<b>Linux Einführung .....</b>	<b>71</b>
<b>14.4.</b>	<b>Hardware.....</b>	<b>71</b>
<b>15.</b>	<b><i>Glossar .....</i></b>	<b>72</b>
<b>16.</b>	<b><i>Offene oder fehlende Punkte .....</i></b>	<b>74</b>
<b>16.1.</b>	<b><i>Freigeben und Mounten von Netzwerk-Verzeichnissen .....</i></b>	<b>74</b>
16.1.1.	nbd – Network Block Device.....	74
16.1.2.	ftp.....	74
16.1.3.	ftp-server.....	74

## 2. Introduction

### 2.1. Preface to the English version

This document is a draft translation of the central chapters of the "CentiPad Linux Handbuch". It should guide the user through system setup and initial development. The complete handbook provides notes taken during Linux development. This information is not CentiPad specific, but included into the handbook for convenience. Chapter in orange writing will be translated when required and time permits.

### 2.2. Focus

Welcome to the „CentiPad Linux Handbook“. This handbook help you with your first steps into the embedded Linux world of CentiPad.

EmbeddedSystem development is a complex task. The developer must create a functional unit of hard- and software.

Different approaches would be possible. This handbook is a collection of answers to questions the author had solve when starting CentiPad development.

Linux specialists will only skim some of the chapters, the beginner can follow the authors path in to embedded Linux development.

### 2.3. additional documentation

- CentiPad Hardware Documentation
- CentiPad Breakout Board Documentation with Quickstart Guide
- CentiPad Breakout Board Schematic
- CentiPad Programming Model
- CentiPad Application Handbook
- The latest documentation is available at [www.centipad.com](http://www.centipad.com)

### 2.4. Safety guidelines

For your own safety and correct operation please follow all of these safety guidelines. Warranty will be void for damage resulting from disregarding this manual. No responsibility will be taken by the manufacturer for secondary damages! No responsibility will be taken for material or personal damage resulting from disregarding this manual.

Only use in dry environment with no flammable gases.

When changing the environmental conditions let the equipment acclimate.

Don't modify the device.

Don't operate devices showing visible damage, no response, or after long storage in an unfavorable environment.

Supply sufficient cooling.

Remove power supply before accessing internals of a device.

Handle with care.

Avoid storage or operation in high humidity conditions.



## 2.5. Life support policy

HAREROD does not authorize or warrant any of its products for use in life support systems, without the specific written consent of Marcus Hasenstab Ingenieurdienstleitungen.

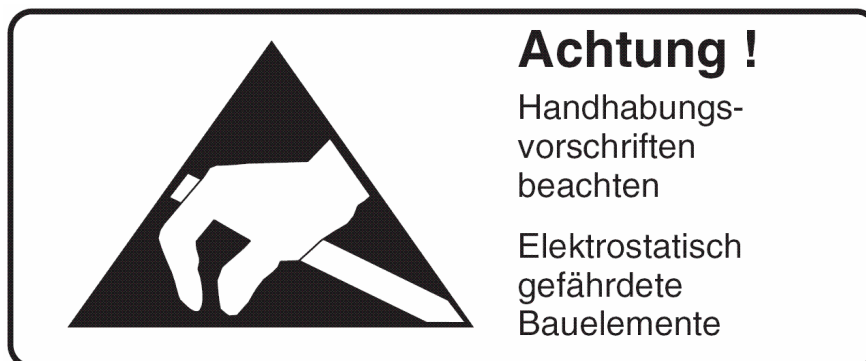
Life support systems are equipment intended to support or sustain life, and whose failure to perform, when properly used in accordance with instructions provided, can be reasonably expected to result in personal injury or death.

## 2.6. Protection against electrostatic discharge (ESD)

This device contains components sensitive to electrostatic discharge. Even a discharge too small to notice may result in destruction or reduced performance.

Use the following precautions against ESD:

- use a properly installed anti-ESD-mat
- wear ESD-bracelets
- follow ESD grounding techniques
- leave the device in its ESD-shielding package until installation. When outside its shielding lay the device on a grounded surface
- don't touch components in the device
- hold PCB at the edges



## 2.7. operational safety

All devices are submitted to extensive functional tests before leaving the factory. But even with very reliable devices defects are possible.

A defect in the device may result in consequential damages. It's the users responsibility to provide protection accordingly. The manufacturer denies responsibility for an kind of defects.

## 2.8. Warrenty regulations / product liability

Warranty and product liability claims are void even during the legal warranty period

- if the device is not operated according to the manuals.
- If the device is opened
- If the device is modified
- If the device is improperly operated

Components which are inherently subject to wear (e.g. batteries, connectors) are exempt from warranty.

## 2.9. Legend

**Verdana Bold**

subtitle, highlighted text

---

<b>Courier New bold</b>	command, command result, programm code
<b><i>Courier New bold italic</i></b>	path or local directory
Courier New	captures the readers attention
<b>Under Construction</b>	passage under construction

## *2.10. Legal stuff*

This manual and all its parts are copyright. All rights reserved, especially right of translation, lecture, reproduction or storage on electronic media.

The author can not be held liable for possible errors in text, pictures or programs or be held liable for their consequences.

Names used in this manual may be trademarks and may have legal restrictions even if not specifically indicated as such.

The right to make technical changes is reserved!

## 3. First steps

### 3.1. Neue Welten

Nun ist es also soweit. Als erfahrener Windows-Entwickler haben Sie dem Druck der Zeit nachgegeben und wagen nun den großen Schritt ins Ungewisse: Linux-Entwicklung. Plötzlich zählen die in der Windowswelt über Jahre hinweg aufgebauten Erfahrungen scheinbar nicht mehr. Man ist wieder an diesem grauen Punkt in der Vergangenheit angelangt, als man das erste Mal einen PC bediente. Alles ist ungewohnt und neu. So ähnlich stellte sich die Situation für den Autor dar, als er vor einiger Zeit diesen Schritt wagte. Trotz aller Unterschiede zu Windows hat es der Einsteiger mit Vorerfahrungen jedoch wesentlich leichter. Immerhin versammelt sich unter dem Dachbegriff Linux auch nur eine Sammlung von Software die ein Rechnersystem zum Leben erweckt. Das feine Detail, dass Linux unter der GNU Lizenz eine schier unerschöpfliche Quelle von Software-KnowHow zur freien Verfügung stellt, sollte man aber in Erinnerung behalten. Das ist ja auch der Grund, warum sich so viele Entwickler für dieses System entscheiden.

Die Erfahrung des Einstiegs in Linux ist dem Autor noch lebhaft im Gedächtnis. Dieses Papier soll die ersten Schritte festhalten und wiedergeben, die der Autor unter Führung der erfahrenen Linux-Entwickler der Firma maintech GmbH gemacht hat.

Der Autor war im Rahmen der CentiPad Entwicklung als ständiger Fürsprecher der Neueinsteiger aus der Windowswelt tätig, mit dem Ziel ein Entwicklungssystem zusammenzustellen, welches einem Anwender den Einstieg mit minimalem Aufwand ermöglicht.

Das CentiDev, das CentiPad Entwicklungssystem, erhebt den Anspruch, dass ein erfahrener Windows-Entwickler, mit Hilfe der mitgelieferten Beispiele und Dokumentation, innerhalb kurzer Zeit in der Lage ist eigene Applikationen zu schreiben.

Der Einsteiger sollte sich zunächst die Kapitel Installation und erste Schritte durchlesen und verstehen. Danach wird im Kapitel Entwicklungszyklus der typische Arbeitsablauf der Anwendungsentwicklung für das CentiPad beschrieben. Diese wenigen Kapitel reichen aus, um auf dem CentiPad eigene Programme zum Laufen zu bringen.

Nachdem die ersten Erfahrungen gemacht sind, empfiehlt sich das weitere Studium der CentiPad Dokumentation und der Application Notes.

### 3.2. Hardware configuration

The hardware configuration expects a CentiPad DevKit with a CentiPad pro. Please take the time to read the „Quickstart Guide“ in the CentiPad BOB Dokumentation.

To setup your CentiPad development system the following items are required:

- CentiPad
- CentiBOB
- CentiDev CD
  
- Ethernet cable
- RS232-cable
- Power supply
  
- X86 host system running (SUSE) Linux or Windows and ca. 3GB freedisk space

Except the host all items are included in the CentiPad Development Kit.

Bitte beachten Sie je nach der von Ihnen angestrebten Konfiguration die Kapitel „Setup unter Linux“ bzw. „Setup unter Windows“. Für beide Installationsvarianten wird grundsätzlich dasselbe Entwicklungssystem verwendet, beide Softwarepakete sind, was den Crosscompiler und die CentiPad-Linux-Plattform betrifft, identisch.

Wichtige Unterschiede ergeben sich jedoch in der Hardware/Treiber-Anbindung, dem Entwicklungsablauf und dem Backup-Konzept.

Ziel ist die Software-Entwicklung für Linux. Auch wenn Sie die Entwicklung unter Windows XP planen, sollten Sie das Kapitel „Entwicklung unter Linux“ lesen, da hier der grundlegende Systemaufbau erklärt wird.

**Checklist:**

- Insert CentiPad correctly in CentiBOB! Check Pin 1 markings!
- Connect cables
- Connect power supply

## 4. Setup on Linux

### 4.1.1. Requirements for installation on Suse

The following packages must be installed:

- gcc
- g++
- ncurses-devel
- bison
- flex
- glib2-devel
- zlib-devel

### 4.1.2. First installation

The following setup description is based on setup on the freely available SUSE-Linux OSS 10.0 (de.opensuse.org).

First complete a standard SUSE installation, e.g. using KDE graphical interface. Since we want to develop software, the GCC-package must be installed. If you don't want to edit your sources with vi, you could use the kdevelop development environment, which slightly resembles Microsoft Visual Studio.

Please follow the SUSE documentation.

The CentiDev development system consists of the following parts:

- 1) The archive `at91-toolchain-gcc-XXX.tar.gz` compiler components for the x86-host-system. The archive `at91-toolchain-gcc-VERSIONNAME.tar.bz2/` has to be copied to `/opt` and extracted via `tar xvjf <archivename>`. This must be done by **root**.
- 2) The archive `Centidev-VERSIONNAME.tar.gz` contains the CentiPad-Linux-System. From this package the CentiPad runtime system build, packed into an image file and transferred to the CentiPad. Copy `Centidev-VERSIONNAME.tar.gz` to `/home/<username>` and extract via `tar xvpf centidev-VERSIONSNAME.tar.gz`. This should be done with the developers user-id, to automatically set the access rights.
- 3) To automatically set the paths at systemstart, the shellscript `arm-9tdmi-linux-gnu.sh` should be copied to `/etc/profile.d/`. Restart the computer to insure the script is run. The script sets the following paths:
 

```
export PATH=/opt/maintech/arm-9tdmi-linux-gnu/bin:$PATH
export PATH=/opt/maintech/gcc-3.4.5-glibc-2.3.6/arm-920t-linux-gnu/bin:$PATH
```
- 4) Setup `minicom` (115200-8-N-1) for communication to th CentiPad debug interface.

Hint: Should acces right problems occur, try `chown -R myuser:users centidev` to set ownership of the `centidev`-directory recursively for „myuser“.

### 4.1.3. System copy / Backup

Once CentiDev is setup satsifactorly, it can be easily copied to other computers.

Just make an archive of the CentiDev directory: `tar cvpzf CentiDev_Archive.tgz CentiDev`

### 4.1.4. directory structure / development system upgrade

New software will be available from [www.centipad.com](http://www.centipad.com) or [www.maintech.de](http://www.maintech.de).

Das CentiDev has a predefined file structure. If this is changed by the user, shelscripts and paths must be adjusted.

When upgrading the development system, care must be taken not lose user files and adjustments. Backup first!

The following list includes some files from the standard installation which are typically changed:

Bei einem Upgrade des Entwicklungssystems ist darauf zu achten, dass keine Benutzerdateien und Anpassungen verloren gehen. Unten folgt eine Liste von Dateien aus der Standardinstallation, die typischerweise geändert werden.

To resolve access problems try:

```
chmod -R 0777 /home/<user>/centidev
chmod -R 0777 /opt/etc/maintech
```

*/home/<user>*

*./centidev*

- The centidev – directory contains makefile and menuconfig to build the CentiPad-System. Your own entry should have a comment with your username to identify those entries for copying to updated scripts/files.
- especially `centidev/platform/system/sysfiles/etc` often contains user-modifications, since the config scripts for the CentiPad-system are stored here.

*./application*

- the application-directory typically contain user-projekt files. Backup your own sources. Here is one `Makefile`, which contains an entry for each user project. `menuconfig` contains control information for the build-system.
- In jedem Projektverzeichnis liegt ein `Makefile` für jeweils dieses Projekt

*./package*

- The build the firmware image for the target system packages are generated. In addition to user programs scripts or additional directories might be generated. Directories and files from `package` will be included in the binaries for the target-system.

*./target*

- This directory contains the binary images for the upload to the target system (`centipad.bin`, `initrd`, `vmlinux`).

*./platform*

- This directory contains the CentiPad-Linux. Changes in the development system structure must be entered here

*./kernel/linux*

*./drivers*

- backup the sources to your own drivers
- each driver-subdirectory contains a `Makefile` und `Kconfig`. These files contains info about the kernel config for the users drivers. Backup!

*/opt/maintech*

- contains the cross compiler tools. Usually not changed by the user.

*/etc/profile.d*

- contains the shell script `arm-9tdmi-linux-gnu.sh`, sets the system paths. Adapt if necessary.

#### 4.1.5. Backup Strategien unter Linux

The standard CentiDev under Linux consists of the following files and directories::

- `/opt/maintech`
- `/home/<USER>/centidev`
- `/etc/profile.d/arm-9tdmi-linux-gnu.sh`

The easiest way is just to backup via

`tar cvzf archivename_date.tgz sourcedirectory`. This will keep the access rights. The files can be burned on CD by e.g. `k3b` (root!).



## 4.2. Setup under Windows with coLinux

### 4.2.1. coLinux

coLinux is freely available virtual machine, which runs Linux under Windows 2000/XP. coLinux needs no registry entries and is therefore ideal for systems where no configuration modification is permitted.

coLinux runs out of a file which contains the complete Linux-file-system. If this file is damaged, the user can simply play back this file.

Complete backups are extremely easy and damage during the first developments can easily be repaired.

The software development can be done with Windows tools, only the cross-compilation requires the Gnu-Compiler under coLinux.

A ethernet driver installation is necessary.

### 4.2.2. Standard User

The most important things:

The standard-user-name under coLinux is **centidev**, the password is **centidev**.

Even though a root user is available under coLinux – never use it. The resulting changes in file access rights will lead to update- and build-problems.

### 4.2.3. Installation

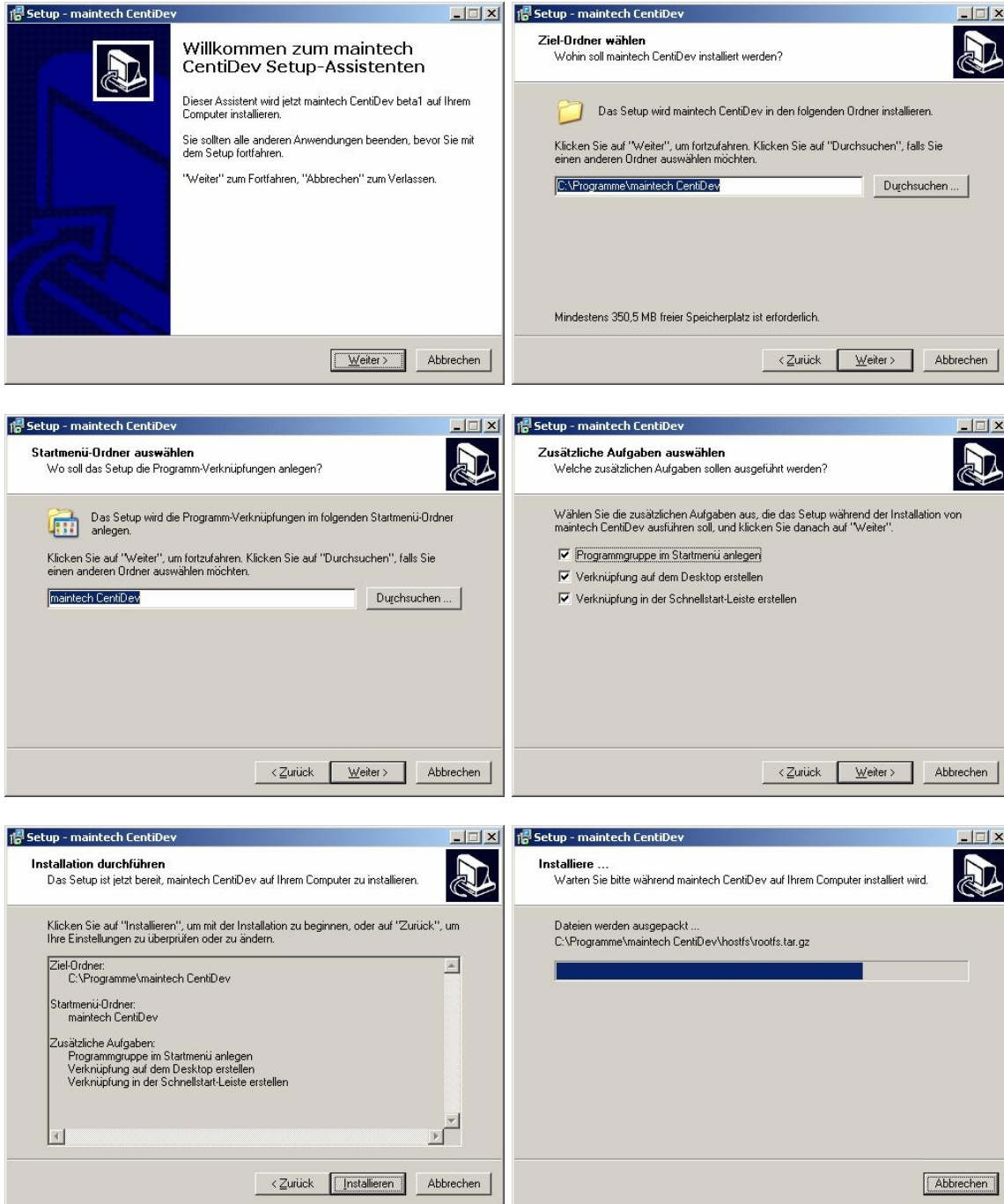
Hint: Since the complete Linux-filesystem is in one fixed-size image-file (e.g. 2GB), this amount of free space must be available before starting installation.

Start the file mt-centidev-xxx-setup.exe:



The following installation asks for the installation directory and which icons should be installed.



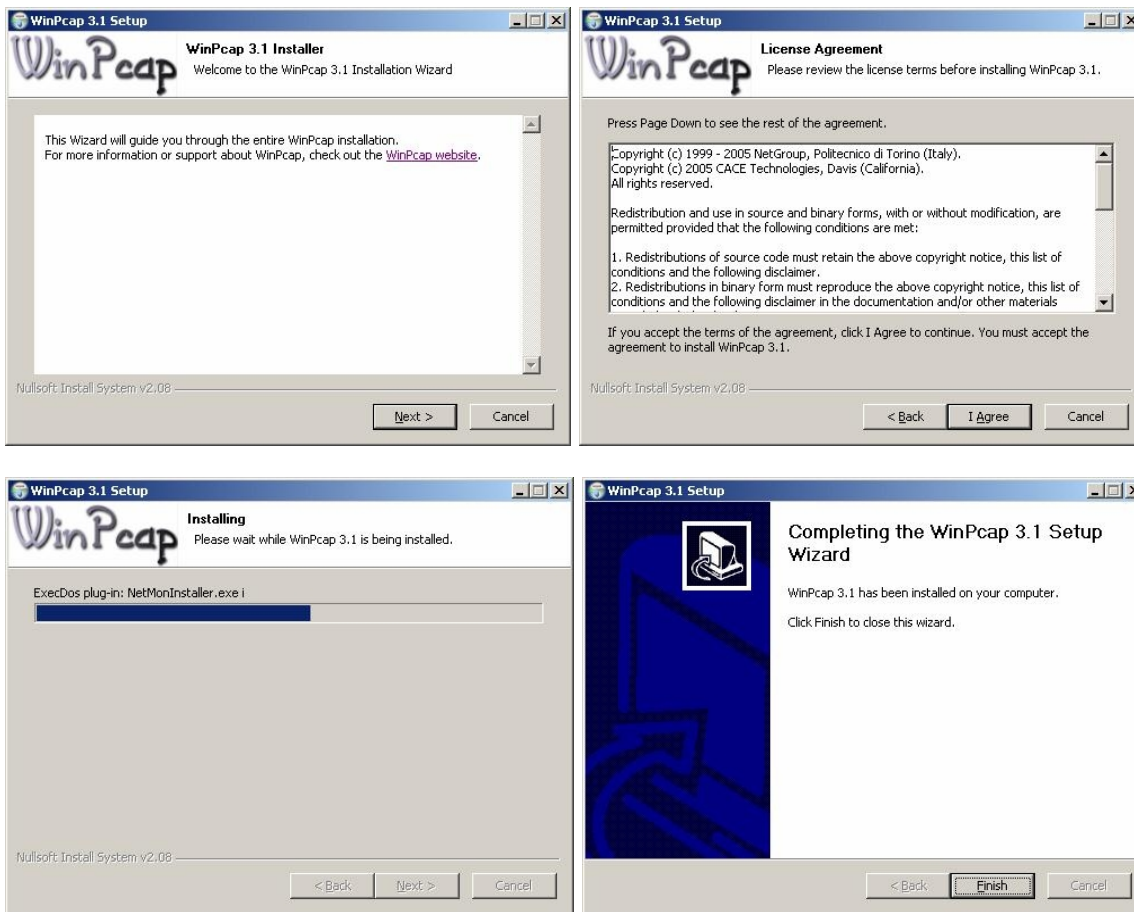


For networking support the TAP-driver must be installed. The following message will appear and must be acknowledged with „Installation fortsetzen“ (continue installation).

The TAP-driver provides the connection between coLinux and the host-Windows.

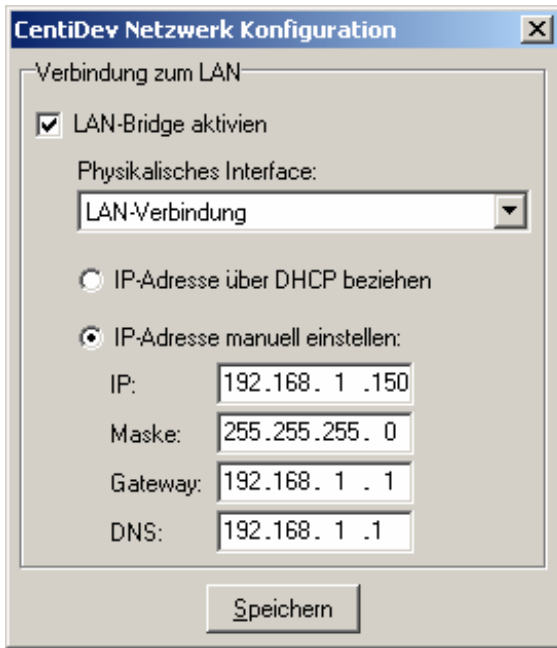


Next step is WinPcap installation, which provides communication from coLinux to LAN (including CentiPad).

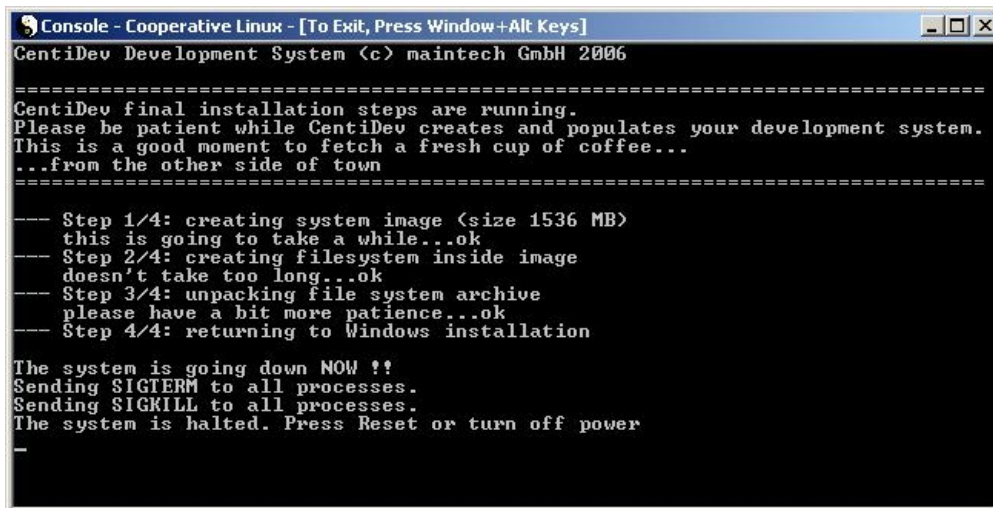


After Wincap installation, the netconfig program starts. Sometimes it becomes covered by other windows – please bring to top.

In NetConfig requires the **physikalische LAN-Verbindung** (the actual LAN-Interface-card you're using, NOT the virtual interfaces we just installed) to the CentiPad and the config for the virtuellen Schnittstelle (virtual interface).  
Hinweis: diese Einstellung muss zum Installationszeitpunkt richtig sein, sonst funktioniert CentiDev nicht und eine Neuinstallation ist notwendig.



After saving the network configuration, coLinux starts automatically. The following setup runs fully automatic, but can take about half an hour, depending on the host system.



coLinux exists automatically after installation. The following box will appear:



#### 4.2.4. Backup strategies for Windows/coLinux

After a standard installation the following directories and files exist:

- the standard installation of coLinux is in the directory `C:\programs\maintech CentiDev`. This directory contains coLinux and the TAP and winpCAP drivers. The subdirectory `\hostfs` contains the Linux filesystem in the containerfile **rootfs**.

A simple backup strategy under Windows could be:

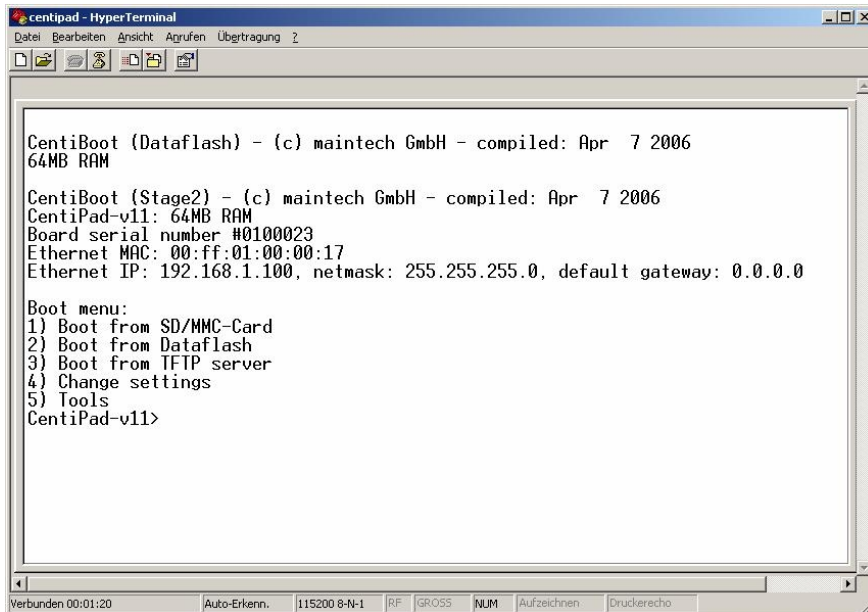
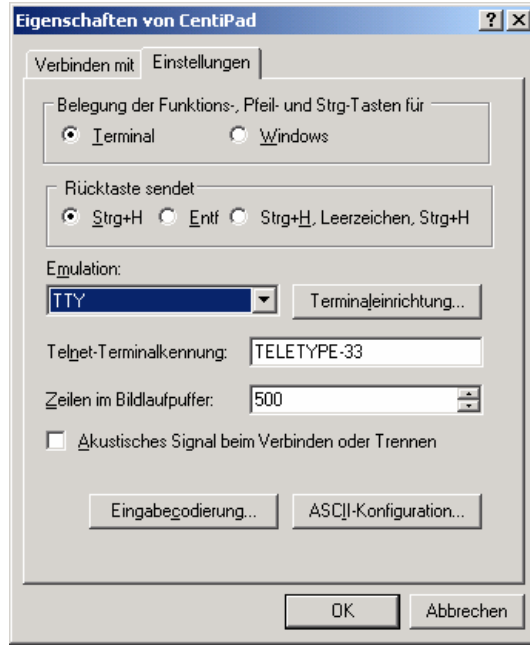
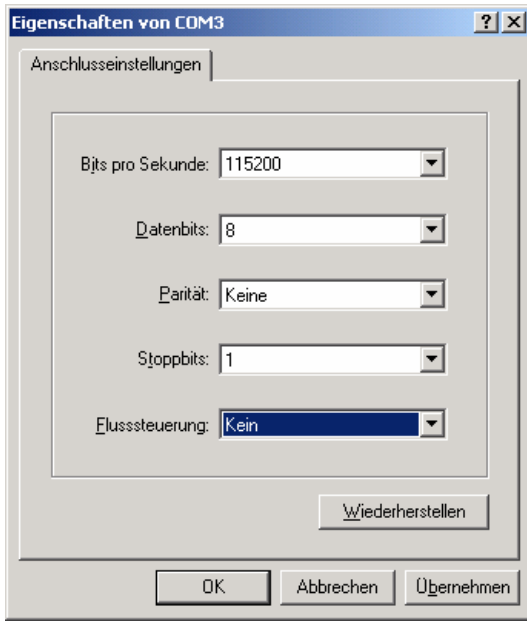
- The complete CentiDev is in the coLinux-Container-file. A backup of this file ensures simple restoring of the CentiDev. For a standard installation you would find this file under `C:\programs\maintech CentiDev\hostfs\rootfs`. The would have a size of 1500MByte but could be zipped to about 400MByte depending on the content
- Colinux has a shared work directory with Windows. This can be used to exchange data between coLinux and Windows. For a standard installation this would be the Windows-path `C:\programs\maintech CentiDev\hostfs\`. Under coLinux this is the path `/windows`. Hint: Since Windows doesn't feature Linux access rights, no Linux-programs can be started from this directory.
- The most simple access is via Windows-Explorer. Under `Netzwerkumgebung\Gesamtes Netzwerk\Microsoft Windows-Netzwerk\Centidev` you'll find coLinux shared directories.
- A complete backup has the advantage of restoring the system by copying a single file – which is about 2GB in size, though.

Under coLinux a tar-archive of the `/home/centidev` directory can be made. The following command would store the actual development sources to the Windows-work directory:  
`tar cvzf /windows/centidev-<datum>.tgz /home/centidev/centidev`

#### 4.2.5. Hyperterminal

Hyperterminal is part of Windows XP. In the startmenu it is found under accessories\communication. Start Hyperterminal and setup the serial interface which is connected to the CentiPad. The connection parameters are 115200-8-N-1, no handshake.





## 4.2.6. Cygwin

Cygwin provides a collection of Linux-like commands under Windows, including a command shell.

Via ssh Cygwin permits access to any Linux hosts in the network – including CentiPad.

### 4.2.6.1. CygwinX Setup

Cygwin consists of the text based „Cygwin“ and the graphics based „CygwinX“.

First Cygwin from <http://www.cygwin.com/> must be installed.

Next CygwinX from [x.cygwin.com](http://x.cygwin.com) must be installed. Please read the manual.

Let's assume CygWinX was installed to `c:\cygwin`. After installing the X11-libraries and ssh, copy the file `c:\cygwin\usr\X11R6\bin\startwin.bat` and adjust it for your requirements. Add the following lines to the section „REM Startup the programs“:

```
run XWin -multiwindow -clipboard -silent-dup-error -unixkill
echo 0
run xsetroot -cursor_name left_ptr -fg white -bg black
echo 1
REM try to login to main unix system
ssh -Y -l <username> 192.168.1.37
```

This initializes X11 and the user `mh` is automatically logged into `192.168.1.37`. In the Cygwin console you can start an X11-terminal with `xterm&`.

### 4.2.6.2. Cygwin for CentiPad

For access to the CentiPad a normal CygWin installation is sufficient. The `startwin.bat` script is not required. `ssh -l <username> 192.168.1.100` connects to a CentiPad at IP-address `192.168.1.100`.



## 5. the maintech Bootloader

### 5.1. Overview

CentiPad is shipped with the comfortable maintech-BootLoader installed. The Bootloaders task is, to configurate CentiPad at power-up and to load the operating system image.

The Bootloader uses the following boot- and configuration memories:

- configuration-EEPROM
- ethernet
- SPI-DataFlash
- miniSDCARD

### 5.2. Factory configuration

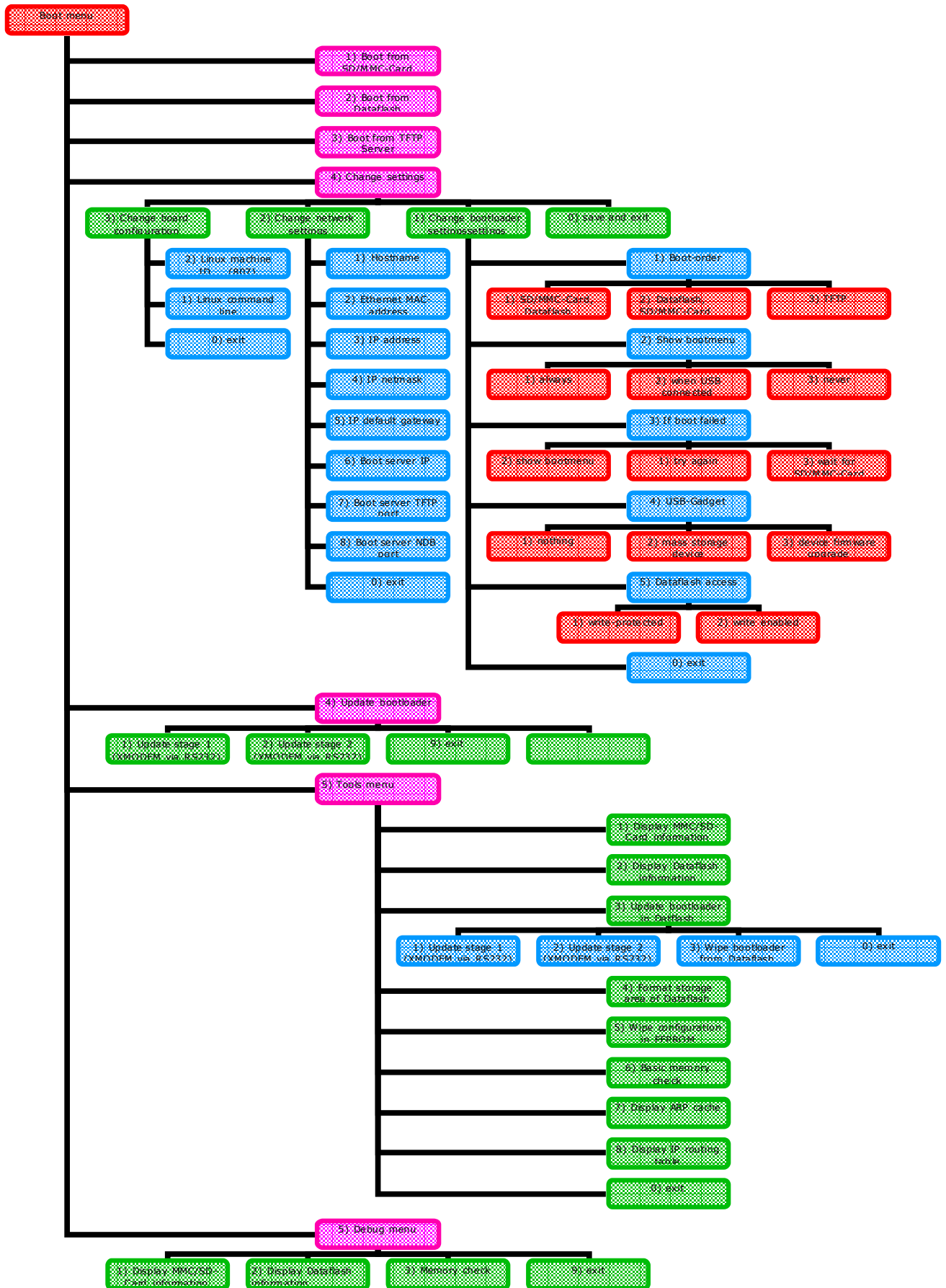
Initialy the following data is stored in the configuration-EEPROM:

- Board ID: 807, this code tells the kernel, that Linux runs on a CentiPad
- Serial number: 0204711 is a CentiPad pro with the serial number 4711
- MAC-address: in the factory an address in the range `00:FF:xx:xx:xx:xx` is entered. The lower digits of the MAC-address resemble the hex-coded serial number.
- TCP/IP informations
- A pre-installed Linux is loaded from DataFlash
- the USB Gadget is configured for DFU

The serial Debug-interface is used as console. The interface settings are 115200-8-N-1. The same settings are used by the CentiPad-Linux.



### 5.3. Menu structure



## 5.4. Boot Menu

Entering the Boot Menu:

Hold Power-Key during reset and release on the message“ Release power button to enter boot loader menu!“.

```
CentiBoot (Stage2) - (c) maintech GmbH - compiled: Apr  7 2006
CentiPad-v11: 64MB RAM
Board serial number #0100008
Ethernet MAC: 00:ff:01:00:00:08
Ethernet IP: 192.168.1.100, netmask: 255.255.255.0, default gateway:
0.0.0.0

Boot menu:
1) Boot from SD/MMC-Card
2) Boot from Dataflash
3) Boot from TFTP server
4) Change settings
5) Tools
CentiPad-v11>
```

The Boot Menu permits selection of the Boot media and an update of the board configuration.

## 5.5. Boot Options

### 5.5.1. Booten über TFTP konfigurieren

Soll das komplette CentiPad Linux übersetzt und getestet werden (z.B. bei Anpassungen am Kernel oder der Startskripte), dann ist folgende Konfiguration sinnvoll:

- Change Settings / Change bootloader settings / Show bootmenu / always
- Change Settings / Change bootloader settings / Boot-order / TFTP

Nach dem Einschalten wartet das CentiPad nun auf den Firmware via TFTP.

Auf dem Hostsystem muss ein TFTP-Server installiert sein. Unter Windows kann dies z.B. <http://perso.wanadoo.fr/philippe.jounin/download/tftpd32.284.zip> sein. Das Programm wird in ein Unterverzeichnis kopiert und gestartet. Als Current Directory wird nun das centidev\centidev\target Verzeichnis ausgewählt (hier liegt das centipad.bin).

Nach einem CentiPad-Reset wählt man im Bootloader-Menu den Punkt 3) Boot from TFTP server. Auf dem Server wird nun die Datei angezeigt, die CentiPad laden will. Der Dateiname setzt sich (seit 060424) zusammen aus "%s-%s", hostname, bootfilename, also per default CentiPad-v11-centipad.bin.

Eine sehr schöne Möglichkeit mehrere CentiPads in einem Netzwerk immer mit dem aktuellen System laufen zu lassen, ist folgende Einstellung:

- Bootloader auf Booten über TFTP einstellen
- Kein Bootmenu anzeigen lassen

Bei einem auf der Konsole eingetippten `reboot` bzw. nach jedem Reset, zieht das CentiPad automatisch das aktuelle Binary über den TFTP-Server.

## 5.5.2. Config for DFU boot

If no network connection is available, a boot via USB-DFU is possible. DFU is quite sensitive regarding Bus-Topologie and therefore not recommended for development. If the complete CentiPad Linux shall be translated and tested (e.g. for Kernel or script updates), the following configuration is recommended:

- Change Settings / Change bootloader / Show bootmenu / always
- Change Settings / Change bootloader / USB-Gadget / device firmware

After PowerUp CentiPad waits for an USB-DeviceFirmwareUpdate. The image file is transferred by `usbupload.sh`. Afterwards CentiPad-Linux automatically.

## 5.5.3. boot from SDcard

In this configuration CentiPad starts automatically after power-up. If the SDcard is mounted under Linux, a new image file can be copied to the SDcard via `ssh`. If the mass-storage-USB-Gadget is activated in the Bootloader, the SDcard and the Dataflash are available as USB devices while the bootmenu is active. This enables copying of from the host-system to the CentiPad.

- Change Settings / Boot-order / SD/MMC-Card
- Change Settings / Change bootloader / Show bootmenu / never
- Change Settings / Change bootloader / USB-Gadget / mass storage

## 5.5.4. reconfig Bootloader

If something went wrong during development and config-EEPROM or dataflash data is damaged, the bootloader start can be prevented by hardware. J12 is a 100Ω resistor. If the pin pointing to the PCB-border is shorted to ground, booting of defective firmware is prevented (SPI-MISO is ground-shortened at the CPU-side). When CentiPad is started the short must be removed. The board now behaves like a factory new device with empty DataFlash. See next chapter for following steps.

## 5.5.5. Setup of a new board

The following steps a part of the factory setup. For the user they're only of interest when a bootloader is damaged or shall be upgraded.

### 5.5.5.1. on Linux

Under `centidev/centitools/cbupdate` is the program `centipad-coin`. This permits automatic reinstallation of a CentiPad. This process rewrites serial number and MAC-adresse neu geschrieben. The last parameter defines the name of the image-file, which the bootloader executes (Dataflash oder SDcard).

Syntax:

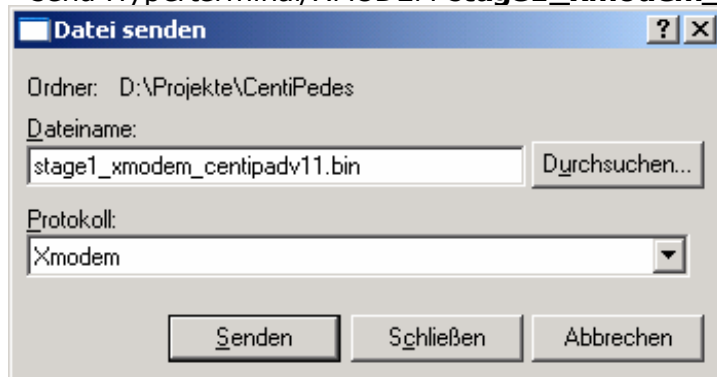
```
centipad-coin /dev/ttyS0 stage1-xmodem.bin stage1.bin stage2.bin serialno
features mac bootfile-name
```

command:

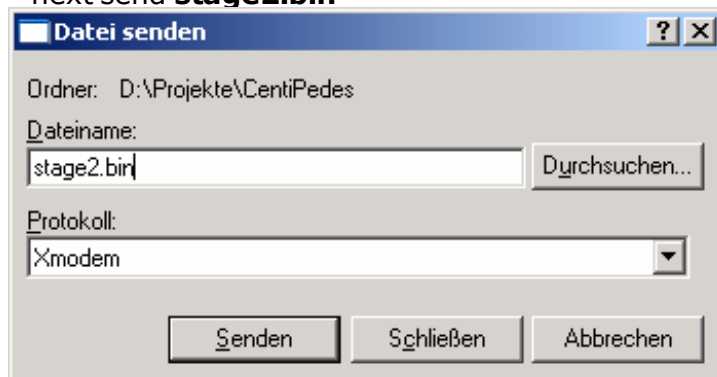
```
./centipad-coin /dev/ttyUSB0↵
../../../../target/centiboot/stage1_xmodem_centipadv11.bin↵
../../../../target/centiboot/stage1_dataflash_centipadv11.bin↵
../../../../target/centiboot/stage2.bin↵
2↵
0100010↵
00:ff:01:00:00:0A↵
centipad.bin
```

### 5.5.5.2. on Windows

- Hyperterminal setup to 115200-8-N-1.
- CentiPad power-up. The terminal-window shows ,C' – the identifier for XMODEM
- send Hyperterminal/XMODEM **stage1\_xmodem\_centipadv11.bin**

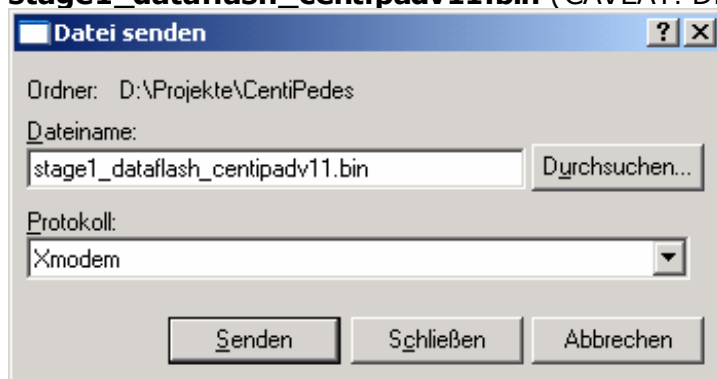


- next send **stage2.bin**



- now the bootloader menu appears
- select "4) Update bootloader"

- select 1) Update stage 1 (XMODEM via RS232), then send **stage1\_dataflash\_centipadv11.bin** (CAVEAT! Different stage1 file!)



- select 2) Update stage 2 (XMODEM via RS232) next and transfer stage2.bin

- reset board
- setup board under 3) Change settings / Board-Config
  - 1) Linux machine ID (807)
  - 2) Ethernet MAC-address(00:FF:00:00:00:01) <- restore original MAC-address

- Bootloader-Settings asrequired, e.g.:
  - 1) Boot-order (SD/MMC-Card, Dataflash)
  - 2) Show bootmenu (always)
  - 3) If boot failed (show bootmenu)

- 4) USB-Gadget (mass storage device)
- 5) Dataflash access (write-protected)
- 6) IP-Adresse etc...

- copy `centipad.bin` with a card reader or the bootloader to the SDcard
- Reset board
- CentiPad boots from SDcard

### 5.5.6. Bootfrom Dataflash

For production operation Linux can be booted from DataFlash without showing the bootloadermenu. This is factory default.

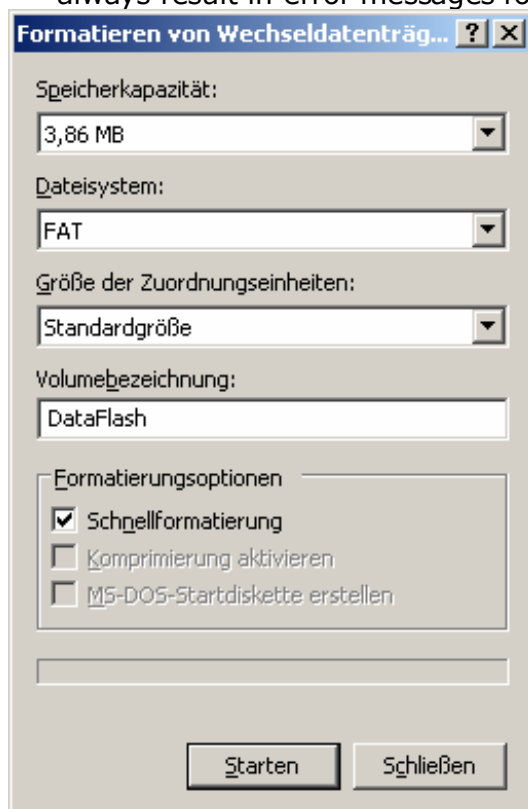
- Change Settings / Boot-order / Dataflash
- Change Settings / Change bootloader / Show bootmenu / never

### 5.5.7. Setup bootloader for dataflash

The Dataflash is formatted at factory and contains a Linux. The Dataflash can be reformatted.

Example under Windows:

- Reset board
- Bootloader setup
  - o Boot menu / Change setting / Change bootloader / USB-Gadget / mass-storage
  - o Boot menu / Change setting / Change bootloader / Dataflash access / write enabled
- Reset board
- Enter Boot Menu
- Windows shows the message „new hardware found“
- Explorer shows SDcard and Dataflash as mass storage
- Formatting „FAT“ and the option „fast format“. The Low-Level-memory tests will always result in error messages for the Dataflash.



- After format `centipad.bin` can be copied to Dataflash
- Check bootloader settings
  - o Boot menu / Change bootloader / Bootorder / Dataflash
- Reset board

#### Under Linux:

- The procedure for Linux is essentially like under Windows
- Dataflash format and `centipad.bin` copy
- Since Dataflash has a size of 8MB, FAT12 is used
- Format with `mkdosfs -v /dev/sdx`
- Mount with `/dev/sdx /mnt/<mountdirectory>`

#### Example on CentiPad:

- theDataflash be formatted directly in the Bootloader

## 6. CentiTools

The directory centitools contains tools for the work with CentiPad.

### 6.1. *cbupdate*

Tools for factory initialization, like `cbupdate` and `centipad-coin`.

### 6.2. *ethboot*

Tool for finding and booting CentiPads via ethernet.

### 6.3. *genext2fs*

This tool generates an ext2fs-filessystem in a file, without loop-mount.

### 6.4. *ipkg*

Paket management tools.

### 6.5. *tools*

Shell scripts and helper files for binaries generation.

### 6.6. *usbdfu*

USB-DFU-tool for `make usbboot`.

## 7. development tools

### 7.1. GCC

GCC is the Linux standard-compiler/linker. The CentiPad-Build-System utilizes two variants of the compiler: the x386-version generates the host-utilities from the build-system sources and the ARM-crosscompiler generates the binaries for the CentiPad. Please have look at `man gcc`.

### 7.2. Make

Make is used to automatically translate wholeprogram packages.

If `make` is called, the `Makefile` in the current directory is used.

`make -f MakefileHostDebug` runs make using the file `MakefileHostDebug`.

Please see `man make`.

CentiDev contains comprehensive Makefiles, which support the complete translation and upload of the target binaries.

### 7.3. Debugging

#### 7.3.1. Host development

Since CentiPad runs Linux, CentiPad programs can be translated and tested on any Linux-computer. This works as long as the program doesn't access CentiPad-specific hardware (e.g. TWI-Bus, CAN-Bus, peripheral-bus). Quite often it's faster to develop the rump-application on the Linux-Host and only do the hardware-specific development on the CentiPad.

#### 7.3.2. Printf-Debugging

The simplest debugging setup is printf-debugging – the program outputs status info via `printf` on the console. Additional variations are `fprintf(stderr, "Error text\n")` and `printk` for Kernel-messages.

#### 7.3.3. GDB

The CentiDev-GCC contains the Gnu Debugger GDB. On the CentiPad `gdbserver` can be run. GDB is utilized bei the `make run`.

Example: CentiPad is connected to the host via Ethernet, the program `lauflicht` shall be tested. Remark: `lauflicht` is a scanning KIT-Light on the GPIO-LEDs.

Procedure:

- Host: build program with debug information
- Host: `make image boot`
- CentiPad: `gdbserver :9999 /usr/bin/lauflicht` the program is started with full path, the server listens to port 9999, CentiPad has the IP-adresse 192.168.1.100

Host:

```
ddd --debugger 'arm-920t-linux-gnu-gdb -ex "set solib-absolute-prefix /opt/maintech/gcc-3.4.5-glibc-2.3.6/arm-920t-linux-gnu/arm-920t-linux-gnu/lib" -ex "symbol-file lauflicht" -ex "target remote 192.168.1.100:9999" -ex "break main" -ex "cont"'
```

– this starts DDD on the Host in the `lauflicht`-directory. Important is the information to use the ARM-Debugger

- Host/Target: Debug



gdbserver either quits automatically when the program exits or can be killed from another shell via `killall gdbserver`.  
See also `make debug`.

## 8. Development cycle on Linux

### 8.1. Building the whole CentiPad package – *make world*

CentiDev is shipped with sources for a consistent software package. The user can start the build of the via `make world`. The software package is a stable and tested base for your own developments.

#### 8.1.1. centidev/make

CentiDev's central files for system build are `centidev/Makefile` and `centidev/Menuconf`.

##### 8.1.1.1. make world

First the target directory is deleted. Afterwards the CentiDev is build from the sources. Result is the file `centipad.bin`.

Two image files are generated – `vmlinux` contains the Linux-kernel and `initrd` the initramdisk with the Linux-filesystem and the user programs. These files are stored under

```
centidev/target/  
    vmlinux  
    initrd  
or combined as  
    centipad.bin
```

A complete rebuild of the CentiPad development system is necessary:

- after CentiDev installation
- after changes in the runtime environment (e.g. in `etc/init.d`)
- to prepare the Embedded-Linux-System for shipment

The first call to `make world` starts the translation of the complete Linux-System. This will take 5..10 minutes. Subsequent calls are usually processed in less than a minute. Before the first call it is imparative to have the setup right (especially under Linux). Wrong setups can render the build system unusable – which would require a re-installation.

##### 8.1.1.2. make boot

Starts `ethboot` with the current `./target/centipad.bin`.

Usually called in combination as `make world boot`.

##### 8.1.1.3. make usbboot

Boots the target via USB- DFU. DFU must be activated in the target-bootloader.

##### 8.1.1.4. make clean

Clears the source-paths and deletes the target-directory. Can lead to a defect build system on some host-configurations. Repair via `update.sh`.

##### 8.1.1.5. make distclean

Cleans cleaner than `make clean`. After `make distclean` source files must be reloaded via internet.

The shell script `update.sh` perorms this process automatically.

### 8.1.1.6. make app image

Only rebuilds the application-directory and creates a new centipad.bin. Ideal for application development.

### 8.1.1.7. make kernel image

Only rebuilds the kernel-directory and creates a new centipad.bin. Ideal for kernel development.

### 8.1.1.8. make bootupdate

Update the Bootloader via the serial. The target mst be in the bootloader menu. The serial interface is setup via `make menuconfig`. During uploads no other programs may access the serial interface (e.g. minicom). The error message "`tcflush() failed: Input/output error`" signals problems with the serial interface. This update only works under Linux. Under coLinux/Windows update is possible via Hypertem/xmodem.

### 8.1.1.9. make run

This command is build into user-program Makefiles. If `make clean run` is called, the respective application is rebuild, uploaded to CentiPad and executed. On the next call the running instance is stopped and replaced by the new version.

### 8.1.1.10. make debug

This command is build into user-program Makefiles. If `make clean debug` is called, the user-program is uploaded to CentiPad and run with the GDB Debug Server. On the Host the GDB is started.

### 8.1.1.11. /centidev/make menuconfig

with `make menuconfig` CentiDev can be adjusted to user requirements. The handling is as known from kernel development procedures. Additional information is found in the menus. The controlling file is `centidev/Menuconf`. Each subdirectory contains a Makefile/Menuconf – pair, which allows configuration for the target system.

CentiDev contains man program packages which can be selected by `centidev/make menuconfig`.

Example:

```
CentiDev Package Selection
Busybox
```

### 8.1.1.12. CentiDev driver setup/centidev/platform/linux/make menuconfig

CentiDev contains several drivers. These can be selected via:  
`cd centidev/platform/kernel/linux/  
ARCH=arm make menuconfig`

Hint: Kernel adjustments can have big consequences. Backup before each access! When started without `ARCH=arm` the build system will be damaged, since the configuration for the x86-hostsystem is called.

## 8.2. *packages* – including your own software into CentiDev

On `make image` CentiDev builds the file `centipad.bin` from the different packages. The user application is contained in `centidev/application/package` untergebracht. The is copied from the package-directory to CentiPads root-filesystem. A file which should appear in `/usr/bin` will have the entry `INSTALLPATH=/usr/bin` in its `Makefile`. During build the binary will be copied to `application/package/usr/bin`. During `make image` the files will be entered into the root-filesystem.

## 8.3. Uploading software to the target system

### 8.3.1.1. Laden des Zielsystem-RAM über TFTP – make world boot

Wenn ein CentiPad im Bootloader-Menü steht, ist ein Booten über TFTP möglich. Die Befehlskombination `make world boot` erstellt das File `centipad.bin` und lädt dieses über Ethernet das angeschlossene CentiPad.

Das Makefile geht hierbei von einem wartenden CentiPad im Netzwerksegment aus. Andere Kombinationen sind durch `ethboot` möglich.

Die Daten werden in das RAM geladen und anschließend ausgeführt. Das Verfahren eignet sich daher für den Test des kompletten Linux Pakets.

Systemkonfiguration:

- Hostrechner mit Linux
- CentiPad über Ethernet-Schnittstelle mit dem Netzwerksegment des Hostsystem verbinden
- CentiPad über Debug-TTY mit dem Host verbinden (für Konsole)
- Bootloader wartet nach dem Einschalten im Menü

Entwicklungszyklus:

- 1) Quelltext schreiben
- 2) Reset am CentiPad auslösen
- 3) `make world boot` starten
- 4) CentiPad Linux startet automatisch – Software testen

### 8.3.1.2. Laden des Zielsystems über SDcard am Hostrechner

Systemkonfiguration:

- Hostrechner mit Linux oder WindowsXP
- CentiPad über Debug-TTY mit dem Host verbinden (für Konsole)
- Bootloader ist für Systemstart von SDcard konfiguriert

Entwicklungszyklus:

- 1) Quelltext schreiben
- 2) `make world image` starten
- 3) auf dem Hostrechner die Datei `centipad.bin` auf die SDcard kopieren
- 4) SDcard ins CentiPad einlegen
- 5) Reset am CentiPad auslösen
- 6) CentiPad Linux startet automatisch – Software testen

### 8.3.1.3. starting from SDcard in CentiPad

System configuration:

- Hostrechner with Linux or WindowsXP
- CentiPad's Debug-TTY connected to host (for console)
- CentiPad's Ethernet connected to host (for data transfer)
- Bootloader is configured for system start from SDcard
- SDcard is mounted e.g. at `/mnt/sdcard`
- ssh-server is running on CentiPad

Development cycle:

- 1) write source code
- 2) `make world`
- 3) copy `centipad.bin` from target-directory to SDcard via `scp`. On the host-console:  
`scp centipad.bin root@192.168.1.40:/mnt/sdcard` (see `scp`)

- 4) restart CentiPad with `reboot`. `reboot` performs a a clean reboot, even if scp has finished, writing to SDcard may still be going on!
- 5) CentiPad Linux starts automatically – test your software

### 8.3.1.4. Upload to Permanent Storage Device

By adjusting the scp command in the last chapter, the image can be uploaded to the current permanent storage device. Depending on your settings this could be either the Dataflash or the SDcard. During development the Dataflash should not be used due to aging and comparatively low speed.

```
scp target/centipad.bin root@192.168.1.108:/mnt/storage
```

### 8.3.1.5. Laden des Zielsystem-RAM über USB - make world usbboot

DFU steht für Device Firmware Upgrade – ein Verfahren um neue Firmware in USB-Devices zu laden. DFU steht derzeit nur unter Linux zur Verfügung.

Wenn im Bootloader das DFU-USB-Gadget aktiviert wurde, wartet der Bootloader beim CentiPad-Neustart auf DFU. `make world usbboot` sendet die Image-Datei `centipad.bin` via USB zum CentiPad. Die Daten werden in das RAM geladen und anschließend ausgeführt. Das Verfahren eignet sich daher für den Test des kompletten Linux Pakets.

Systemkonfiguration:

- Hostrechner mit Linux
- CentiPad über USB-Device-Schnittstelle mit dem Hostsystem verbinden
- CentiPad über Debug-TTY mit dem Host verbinden (für Konsole)
- DFU-USB-Gadget im CentiPad Bootloader ist aktiv, Bootloader wartet nach dem Einschalten auf DFU

Entwicklungszyklus:

- 1) Quelltext schreiben
- 2) Reset am CentiPad auslösen
- 3) Dem Hostrechner einige Sekunden Zeit lassen, um das DFU-Gadget zu enumerieren
- 4) `make world usbboot` starten
- 5) CentiPad Linux startet automatisch – Software testen

## 8.4. User program development

### 8.4.1. Overview

This chapter describes the procedures for user program development after successful installation of the build-system.

The typical development cycle consists of the following steps:

- 1) Create your concept
- 2) Write sources
- 3) Translate sources
- 4) Upload binaries to target system
- 5) test
- 6) add your new programs to the build system
- 7) create System Images
- 8) test of the complete system

Dependant of your development task, different approaches might be used.

### 8.4.2. including your programs into the build system

The build-system contains a directory **application**. The application programs are stored here, in different subdirectories. This allows a complete application backup by `tar cvzf srcarchiv.tgz application`.

The directory `centidev/application` contains a **Makefile** which is called by the build-system. This **Makefile** contains the sections `all`, `build`, `package`, `clean` and `distclean`. The user creates in the line `APPS` a blank separated list of the applications, which are to be included in the packages.

Each application program contains a **Makefile** in its subdirectory which controls the programs build process.

### 8.4.3. „Hello World“ on the CentiPad

The C-source of „Hello World“ on the CentiPad consists of the following lines:

```
#include <stdio.h>

int main(int argc, char** argv)
{
    printf("Hello World!\n");
    return 0;
}
```

When in the directory `centidev/application/helloworld` the Linux-Compiler `gcc` is called: `gcc -o helloworldhost helloworld.c`, `gcc` generates an executable on the host called `helloworldhost`.

To create binaries for CentiPad, the build-system contains a cross-compiler for the ARM-processor in `/opt/maintech`. To correctly call the crosscompiler, enthält das Verzeichnis `centidev/application/helloworld` contains a **Makefile**.

#### Listing of a Makefile including parameter passing:

```
TARGET=serialtest
INSTALLPATH=/usr/bin

DEBUG=n
RUNARGS=/dev/tty3 "SerialTest"
```

```

C_SOURCES=serialtest.c
CXX_SOURCES=

CXXFLAGS=-I../target/staging/include -I../target/staging/include/centidev -
I../platform/kernel/linux/include -I.
LDFLAGS=-L../target/staging/lib
LIBS=../target/staging/lib/libsigc-2.0.so ../target/staging/lib/libcentidev.so

DEPDIR=./deps

#####

include ../devconfig
include ../config

CXX=../target/staging/bin/g++-uc
CC=$(MT_CROSS_COMPILE)gcc
LD=../target/staging/bin/g++-uc

COBJS=$(C_SOURCES%.c%.o)
CXJOBJS=$(CXX_SOURCES%.cpp%.o)
DEPS=$(CXX_SOURCES%.cpp%.dep) $(C_SOURCES%.c%.dep)
FULLDEPS=$(addprefix $(DEPDIR)/,$(DEPS))

ifeq ($(DEBUG),y)
    CFLAGS+=-O0 -g3
    CXXFLAGS+=-O0 -g3
    LDFLAGS+=-g3
else
    CFLAGS+=-O2
    CXXFLAGS+=-O2
    LDFLAGS+=-s
endif

build: $(DEPDIR) $(TARGET)

-include $(FULLDEPS)

$(TARGET): $(CXJOBJS) $(COBJS)
    $(LD) $(LDFLAGS) -o $(TARGET) $(CXJOBJS) $(COBJS) $(LIBS)

$(CXJOBJS):
    $(CXX) $(CXXFLAGS) -MD -MP -MF $(patsubst %.o,$(DEPDIR)/%.dep, $@) -c $(patsubst
%.o,%.cpp, $@) -o $@

$(COBJS):
    $(CC) $(CFLAGS) -MD -MP -MF $(patsubst %.o,$(DEPDIR)/%.dep, $@) -c $(patsubst %.o,%.c,
$@) -o $@

$(DEPDIR):
    @mkdir -p $(DEPDIR)

install:
    mkdir -p ../package/$(INSTALLPATH)
    cp -v $(TARGET) ../package/$(INSTALLPATH)

clean:
    rm -rf $(CXJOBJS) $(COBJS) $(DEPDIR) $(TARGET) *~

distclean: clean

run: build
    ../centitools/bin/ethboot kill $(CFG_HW_ETH)
    ../centitools/bin/ethboot upload $(CFG_HW_ETH) $(TARGET) /tmp/$(TARGET)
    ../centitools/bin/ethboot exec $(CFG_HW_ETH) "/tmp/$(TARGET) $(RUNARGS)"

debug: build
    ../centitools/bin/ethboot upload $(CFG_HW_ETH) $(TARGET) /tmp/$(TARGET)
    ../centitools/bin/ethboot exec $(CFG_HW_ETH) "gdbserver :26200 /tmp/$(TARGET)"
    $(MT_CROSS_COMPILE)gdbtui \
        -ex "set solib-absolute-prefix $(MT_CROSS_LIBDIR)" \
        -ex "symbol-file $(TARGET)" \
        -ex "target remote ../centitools/bin/ethboot getip $(CFG_HW_ETH):26200" \
        -ex "break main" \
        -ex "cont" \

```



```

        -ex "clear main"
        ../../centitools/bin/ethboot kill $(CFG_HW_ETH)

ddd: build
        ../../centitools/bin/ethboot upload $(CFG_HW_ETH) $(TARGET) /tmp/$(TARGET)
        ../../centitools/bin/ethboot exec $(CFG_HW_ETH) "gdbserver :26200 /tmp/$(TARGET)"
        ddd --debugger "$(MT_CROSS_COMPILE)gdb \
        -ex "set solib-absolute-prefix $(MT_CROSS_LIBDIR)" \
        -ex "symbol-file $(TARGET)" \
        -ex "target remote `../../centitools/bin/ethboot getip $(CFG_HW_ETH)`:26200" \
        -ex "break main" \
        -ex "cont" \
        -ex "clear main"
        ../../centitools/bin/ethboot kill $(CFG_HW_ETH)
    
```

The Makefile is two-parted. The first part contains the following information:

- The name of the executable (**TARGET=helloworld**)
- The installation path of the executable `initrd` (**INSTALLPATH=/usr/bin**)
- A blanks separated list of C/C++ source files (**helloworld.c**)
- A list of compiler flags
- Library information
- linker information

The lower part contains work information for the cross compiler and is usually not to be changed by the user. If in `centidev/application/helloworld` `make` is entered, `gcc` builds an executable for CentiPad named `helloworld`.

## 8.4.4. Testing the user program „Hello World“

This chapter describes a quick development cycle, which translates application programs on the host system and uploads them to the CentiPad ramdisk `/tmp`. This allows a complete application build/test cycle without rebuilding the image files.

System configuration:

- Host system running Linux or WindowsXP
- CentiPad connected to host via Debug-TTY (for console)
- CentiPad connected to host via Ethernet (for data transfer)
- Bootloader is configured for systemstart from SDcard
- SDcard is mounted e.g. at `/mnt/sdcard`
- ssh-server dropbear runs on the CentiPad

Development cycle bare-boned:

- 1) write source code in `centidev/application/helloworld`
- 2) `centidev/application/helloworld/Makefile` adapted as required
- 3) `make`
- 4) copy `centidev/application/helloworld/helloworld` via `scp` to the CentiPad-ramdisk: on the host-console: `scp helloworld root@192.168.1.100:/tmp` (siehe `scp`)
- 5) start the program on the CentiPad: `/tmp/helloworld`.

Hint: the same result can be achieved by typing `make clean run`. The Makefile will then call `ethboot` to upload the executable to the target system.

## 8.4.5. Integrating application-files into image-files

As already described `centidev/make world` calls `centidev/application/Makefile`. Add the following entries:

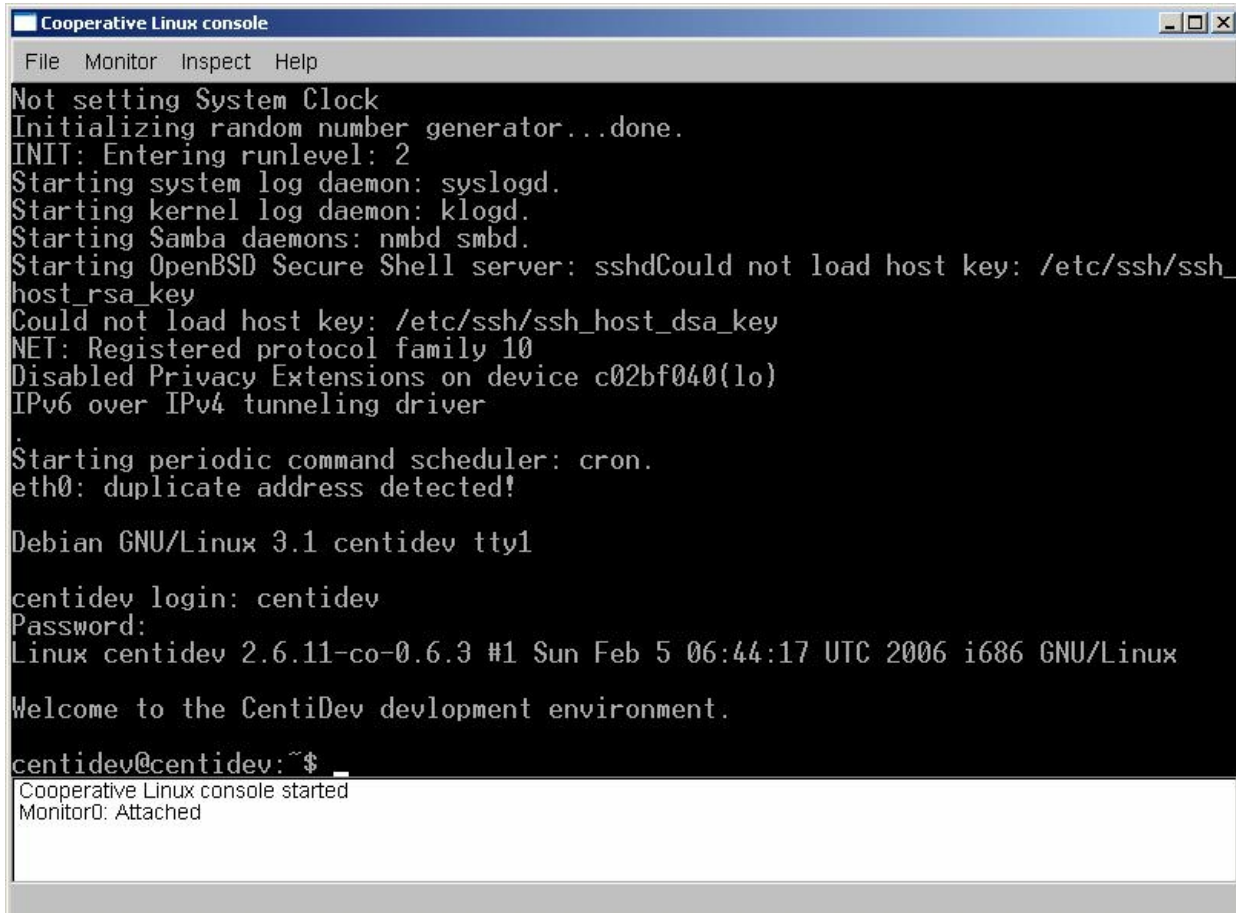
```
APPS=helloworld
```

Together with the entries into *centidev/application/helloworld/Makefile* the application program *bin/helloworld* is included in *initrd* and will be part of the CentiPad software package after the next *reboot*.

Hint: To trigger application rebuild be sure to type *make clean* in the application directory. To make sure the whole system is rebuild type *make world* in the *centidev*-directory.

## 9. development cycle on coLinux

### 9.1. introduction



```

Cooperative Linux console
File Monitor Inspect Help
Not setting System Clock
Initializing random number generator...done.
INIT: Entering runlevel: 2
Starting system log daemon: syslogd.
Starting kernel log daemon: klogd.
Starting Samba daemons: nmbd smbd.
Starting OpenBSD Secure Shell server: sshdCould not load host key: /etc/ssh/ssh_
host_rsa_key
Could not load host key: /etc/ssh/ssh_host_dsa_key
NET: Registered protocol family 10
Disabled Privacy Extensions on device c02bf040(lo)
IPv6 over IPv4 tunneling driver
.
Starting periodic command scheduler: cron.
eth0: duplicate address detected!

Debian GNU/Linux 3.1 centidev tty1

centidev login: centidev
Password:
Linux centidev 2.6.11-co-0.6.3 #1 Sun Feb 5 06:44:17 UTC 2006 i686 GNU/Linux

Welcome to the CentiDev development environment.

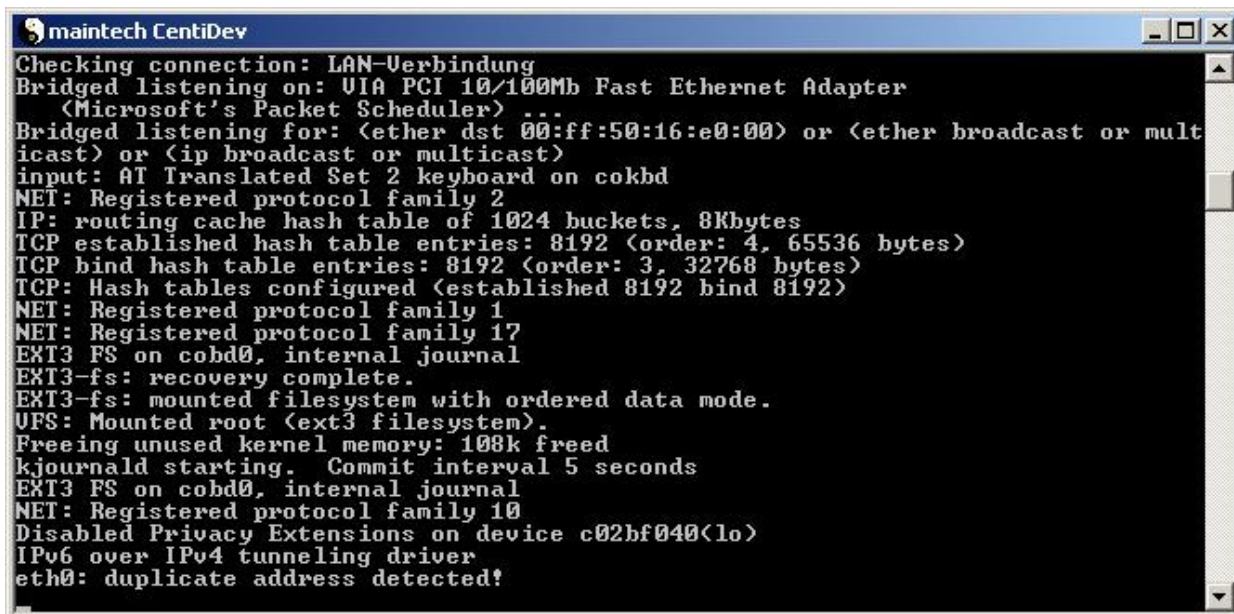
centidev@centidev:~$
Cooperative Linux console started
Monitor0: Attached
    
```

The development cycle on coLinux is essentially the same as on Linux. Some issues arise due to usage of the virtual machine.

Every developer has his preferred workflow. coLinux enables a Windows developer to use his familiar tools. The end result is firmware for a Linux system, the translation runs on a virtual machine.

At startup coLinux boots a Linux in ist virtual machine, which will take a few seconds, depending on the host system.

coLinux opens two windows. The window **maintechCentiDev** is for status only, it can be minimized to the taskbar. The work is done in the **CooperativeLinuxConsole** window.



```

maintech CentiDev
Checking connection: LAN-Verbindung
Bridged listening on: VIA PCI 10/100Mb Fast Ethernet Adapter
(Microsoft's Packet Scheduler) ...
Bridged listening for: (ether dst 00:ff:50:16:e0:00) or (ether broadcast or mult
icast) or (ip broadcast or multicast)
input: AT Translated Set 2 keyboard on cokbd
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
EXT3 FS on cobd0, internal journal
EXT3-fs: recovery complete.
EXT3-fs: mounted filesystem with ordered data mode.
UFS: Mounted root (ext3 filesystem).
Freeing unused kernel memory: 108k freed
kjournald starting. Commit interval 5 seconds
EXT3 FS on cobd0, internal journal
NET: Registered protocol family 10
Disabled Privacy Extensions on device c02bf040(lo)
IPv6 over IPv4 tunneling driver
eth0: duplicate address detected!

```

The kernel and CentiDev filesystem are inside the image file `rootfs`. After startup a Linux-console appears, ready for logon as user `centidev` with password `centidev`.

## 9.2. Build and upload of a CentiPad Linux

### 9.2.1. image build

in the directory `/home/centidev/centidev` the command `./make world` creates CentiPad-Linux image file `centipad.bin` in `/home/centidev/centidev/target`.

### 9.2.2. data transfer to target system

The same transfer techniques via LAN (make run, or scp) are normally available for coLinux. The following chapters describe additional/alternative ways for data upload.

### 9.2.3. copy to bootloader mass storage

System configuration:

- Hostrechner with Windows XP / coLinux
- CentiPad connected to host via USB-Device-interface
- CentiPad connected to host via debug-TTY (for console)
- DFU-USB-Gadget in the CentiPad Bootloader is activated and configured as mass storage

Development cycle:

- 1) Write source code
- 2) `make world`
- 3) copy files to the Windows work directory  
`cp /home/centidev/centidev/target/centipad.bin /windows`
- 4) Reset CentiPad
- 5) Wait for gadget enumeration
- 6) Copy files with Windows-Explorer to target
- 7) Start CentiPad Linux – test software

Hint: To keep the preinstalled Linux on the CentiPad dataflash, development should be done with the SDcard as boot media. This keeps a working Linux on CentiPad.

### 9.3. coLinux shutdown

Type `sudo halt` to stop coLinux. Afterwards close the console window.

If the console window is closed, coLinux performs an automatic shutdown.

### 9.4. build and upload of programs to CentiPad

How to include application programs into the CentiPad-buildsystem is described in the chapter „development cycle on Linux“.

If a network connection between coLinux and CentiPad is available, programs can be directly uploaded to the CentiPad ramdisk `/tmp`.

Without coLinux-tetzwork connection the following procedure is necessary, though somewhat extensive.

Copy the applications source directory to the Windows work directory:

```
cp -r /home/centidev/centidev /application/helloworld /windows
```

Translate the program: `/windows/helloworld/make`

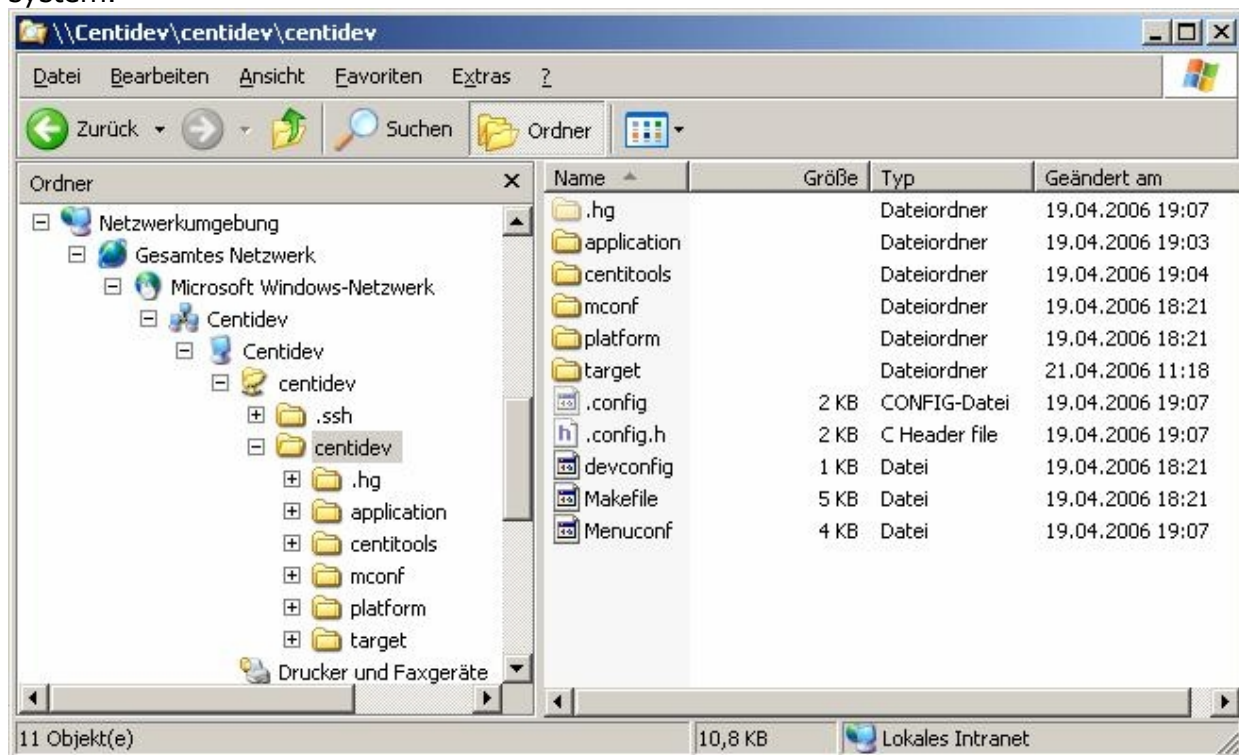
Copy the executable e.g. with WinSCP to the CentiPad ramdisk `/tmp`. Since Windows doesn't support Linux access rights, programs can't be run from the `/windows` directory. After upload to CentiPad the permissions must be set by `chmod +x 0777 helloworld`.

The advantage of work in the `/windows` directory is access for any Windows Programs. Your code can be edited by any Windows editor.

If network connection is available:

The Windows-Explorer allows access to the home-directory of the user centidev. Files can be opened as usual, permits are automatically kept.

Example: connect network drive, `\\192.168.1.150\centidev` connects to developments system.



## 10. Introduction to (Embedded) Linux

### 10.1. Linux Überblick

Eine Einführung in Linux als Betriebssystem würde den Rahmen dieses Skripts bei Weitem sprengen. Unser Anspruch im vorliegenden Zusammenhang ist aber gerade, dass Linux nicht als Betriebssystem gelernt werden muss. Wir nutzen nur einige Vorteile und Annehmlichkeiten die uns die Betriebssystementwickler für unser CentiPad zusammengestellt haben.

Wird die Entwicklung auf einem Hostsystem mit dem frei verfügbaren SUSE 10 OSS durchgeführt, so erinnert vieles an Windows. Die unter SUSE 10 verfügbare graphische Benutzeroberfläche KDE lehnt sich stark an Windows an. Dies erleichtert dem Windows Anwender den Einstieg in Linux enorm. Der KDE Konqueror erinnert stark an den Windows Explorer, so dass einfache Arbeiten problemlos von der Hand gehen. Gelegentlich hat zum das Kopieren mit Konqueror nicht den gewünschten Erfolg. Linux ist und bleibt Kommandozeilen-orientiert, daher lohnt es sich für administrative Arbeiten einige wichtige Kommandos und Werkzeuge zu kennen.

Wird die Entwicklung mit coLinux unter Windows XP durchgeführt, so benötigt man auf dem Hostsystem nur wenige Linux Kommandos. Auf dem CentiPad sind diese Kommandos zumindest zu Debug-Zwecken hilfreich.

Für einen tieferen Einstieg seien die Bücher aus der Literaturliste empfohlen.

### 10.2. CentiPad Standard User

The Standard-User on the CentiPad is **root**, the password is **admin**.

### 10.3. Kleine Auswahl von Linux Kommandos

In diesem Kapitel wird auch eine kurze Übersicht einiger Linux-Anwender-Kommandos gegeben, welche die Arbeit mit dem CentiPad-Entwicklungssystem einfacher machen. Administrative Kommandos die im fortgeschrittenen Einsatz interessant werden, sind weiter unten ausführlicher beschrieben.

#### 10.3.1. Konsole

Die Konsole ist das Standard-Benutzerinterface unter Linux. Sobald ein CentiPad mittels der seriellen Debug-Schnittstelle an einen Hostrechner mit Terminalprogramm angeschlossen ist, steht ein Linux-Rechner zur Verfügung, welcher sich in der Bedienung von einem Desktop-System kaum unterscheidet.

##### 10.3.1.1. Tastatursteuerung der Konsole

- Gestartet werden Programme durch Eingabe des kompletten Pfades und Dateinamens
- Programme im aktuellen Verzeichnis können mittels `./dateiname` gestartet werden
- `<ctrl>-c` beendet das aktuell im Vordergrund laufende Programm
- Ein `&` nach dem Programmnamen startet ein Programm direkt in den Hintergrund
- `<ctrl>-z` schiebt das aktuell im Vordergrund laufende Programm in den Hintergrund und ermöglicht den Zugriff auf die Shell
- Das Kommando `fg` bringt ein Programm wieder in den Vordergrund

### 10.3.1.2. Standard Eingabe/Ausgabe

Linux verfügt über folgende Standardkanäle, auf welche Programm Ein-/Ausgaben zugreifen können: `stdin`, `stdout`, `stderr`.

Wird ein Programm von der Konsole aufgerufen, so kann man die Ein-Ausgabe wie folgt umlenken:

```
< InputDatei      Standardeingabe wird aus der InputDatei gezogen
> OutputDatei     Standardausgabe wird in die OutputDatei geschoben
2> ErrorDatei     Fehlermeldungen werden in die ErrorDatei geschoben
2>&1 AllOutputDatei Standard und Fehlermeldungen werden in die AllOutputDatei
geschoben
>> OutputDatei   das doppelte Größerzeichen sorgt dafür, dass Ausgaben an eine
bestehende Datei angehängt werden (z.B. auch log 2>>&1)
```

Beispiel:

```
make world > log 2>&1
```

Mitloggen der Ausgaben des System-Builds, besonders hilfreich bei der Suche von versteckten Fehlern.

### 10.3.1.3. Piping

Über Pipes können Ausgaben eines Programms als Eingabe für das nächste Programm verwendet werden. Damit kann man unter Linux sehr leicht Programme miteinander verketteten.

Beispiel:

```
ls -l | more
```

Leitet die Ausgabe des `ls`-Kommandos an das Werkzeug `more` weiter.

Mittels `tee` kann man in eine Pipe auch eine Abzweigung einbauen, und damit z.B. gleichzeitig eine Programmausgabe auf der Konsole und in eine Datei erreichen:

```
ls -l | tee OutputDatei | more
```

### 10.3.1.4. which – welches Programm würde ausgeführt werden?

Mit `which <Kommando>` kann man herausfinden, welches Programm die Shell ausführen würde. Dies ist sehr hilfreich, wenn mehrere Programme gleichen Namens vorhanden sind und die Suchpfade nicht eindeutig bekannt sind.

## 10.3.2. man, info, xman

Wenn man beim Lesen fremder Quellen über unbekannte Befehle stolpert, so kann man sich mit diesen Befehlen Hilfestellung geben lassen. Linux ist größtenteils elektronisch dokumentiert. Die Eingabe von `man man` gibt Auskunft über das Hilfesystem selbst.

Wichtig zu wissen ist, dass das Manual-System in mehrere Hauptkapitel unterteilt ist. So sucht `man 2 read` die Beschreibung der `read`-Funktion im Handbuch für Systemaufrufe. Ohne Zusatzparameter werden Shellkommandos gesucht.

## 10.3.3. Anzeigebefehle

### 10.3.3.1. cat – zeige Daten an

Der `cat dateiname` zeigt die Datei `dateiname` auf der Standardausgabe an.

### 10.3.3.2. more/less – Konsolenausgabe komfortabel

`more` und `less` sind Programme welche die Ausgaben auf die Standardausgabe filtern und besser lesbar machen. Während `more` die Ausgabe zwischen puffert und bei jedem Tastendruck eine Seite weiterblättert, ermöglicht `less` sogar Scrolling in beide Richtungen.

`cat ./readme` | `less` Zeigt den Inhalt der Datei `readme` komfortabel an.

### 10.3.3.3. `tail` – Ende einer Datei anschauen

`tail` zeigt das Ende einer Datei an. Mit `tail /var/log/messages` kann man sich z.B. die letzten Einträge in die Datei mit den Systemmeldungen anschauen.

## 10.3.4. Verzeichnis- und Dateibefehle

### 10.3.4.1. `cd` – Verzeichniswechsel

`cd pfadname` wechselt in den Pfad `pfadname`, relativ zum aktuellen Pfad.

`cd /pfadname` wechselt in den Pfad `pfadname` beginnend im Wurzelverzeichnis.

### 10.3.4.2. `pwd` – Print Work Directory

`pwd` listet das aktuelle Arbeitsverzeichnis der Shell auf.

### 10.3.4.3. `ls` – List Directory

`ls` zeigt das aktuelle Verzeichnis an. In der Kombination `ls -la` ähnelt die Anzeige dem MSDOS `dir`-Befehl.

### 10.3.4.4. `md, mkdir` – Make Directory

`mkdir pfadname` erzeugt das Verzeichnis `pfadname`.

### 10.3.4.5. `mv` – move file or directory

`mv quellname zielname` benennt `quellname` in `zielname` um.

### 10.3.4.6. `rm`

`rm` löscht Dateien.

`rm filename` löscht `filename`.

`rm -rf *` ist einer der gefährlicheren Befehle – hiermit wird ein Verzeichnis ohne weitere Rückfrage rekursiv gelöscht.

### 10.3.4.7. `touch`

Der `touch`-Befehl erlaubt es Dateien ein bestimmtes Datum zu geben.

Damit können z.B. bei der Auslieferung von Software alle Quelltexte und Dokumente auf das gleiche Datum gebracht werden.

Die folgende Kombination versieht alle Dateien im Verzeichnis `firmware` mit dem Datum 19.01.2006,18:16h:

```
find firmware -print -exec touch -t 0601191816 {} \;
```

### 10.3.4.8. `cp` – Dateien kopieren

`cp` ist der klassische Kopier-Befehl unter Linux. Mit Hilfe der Optionen kann man jedoch während des Kopiervorgangs auf die Eigenschaften der Kopien Einfluss nehmen.

`cp -a` behält die Zugriffsrechte bei. Dies ist zum Beispiel beim Setup oder Backup wichtig, wenn komplette Systeme kopiert werden sollen.

`cp -a * /tmp/susi` kopiert alle Dateien aus dem aktuellen Verzeichnis in das Verzeichnis `tmp/susi`



### 10.3.4.9. tar – Backup und Archive

tar ist unter Linux das Standard Packwerkzeug.

Der Tape ARchiver ist unter Linux das gängige Archivierungswerkzeug.

Archiv aus dem Quellverzeichnis und dessen Unterverzeichnissen/-dateien erstellen:

```
tar cvzf archivdatei.tgz quellverzeichnis
```

Archiv auspacken:

```
tar xvzf archivdatei.tgz zielverzeichnis
```

Ein zusätzliches z in der Parameterliste erzeugt/entpackt zip-Dateien, ein j erzeugt/entpackt bz2 Dateien.

Wie immer unter Linux ist auch bei tar auf die Zugriffsrechte zu achten!

Der Parameter p erhält die Zugriffsrechte der Originaldateien.

### 10.3.4.10. chmod - Change file access permissions.

chmod -R 0777 <Zielverzeichnis> gibt jedem uneingeschränkten Zugriff auf das Zielverzeichnis. Dies ist zwar nicht besonders sicher, spart aber gerade beim Testen eines neuen Systems eine Menge Zeit bei der Suche nach nicht geschalteten Berechtigungen.

Ausführbare Dateien erkennt man unter linux mit Hilfe des x-Bits in den Datei-Rechten. Um ein File ausführbar zu machen genügt es chmod +x dateiname einzugeben.

Die Angabe von chmod 4755 dateiname ermöglicht das Öffnen einer Datei mit den Rechten des Eigentümers. Damit kann root einem Anwender Programme zur Verfügung stellen, welche mit den Rechten von root ausgeführt werden.

### 10.3.4.11. find - Suche Dateien im Verzeichnisbaum:

find pfadname -name dateimaske sucht Dateien in pfadname auf die dateimaske zutrifft.

find . -name test.h sucht die Datei test.h im aktuellen Verzeichnis und dessen Unterverzeichnissen:

### 10.3.4.12. grep – Dateien mit definiertem Inhalt suchen

grep ist ein universelles Suchwerkzeug.

```
grep suchmuster suchmaske
```

grep AT91\_SYS \*.h sucht im aktuellen Verzeichnis in den Dateien mit der Endung .h nach der Zeichenkette AT91\_SYS.

grep -i -l -r „at91\_sys\_write“ ./\* sucht im aktuellen Verzeichnis und allen Unterverzeichnissen Dateien mit dem Inhalt at91\_sys\_write und zeigt deren Dateinamen an.

Optionen:

- w Suchmuster muss ein Wort sein
- i ignoriere Gross/Kleinschreibung
- r Rekursiv in Verzeichnissen suchen
- l nur Dateinamen auflisten
- n Zeilennummer ausgeben

### 10.3.4.13. diff – Unterschiede zwischen Dateiversionen finden

Mit diff -u -r <original> <geändert> kann man die Unterschiede zwischen Originaldateien und neueren Versionen herausfinden. Diff ist ein wichtiges Werkzeug auf der Suche nach Fehlern in neueren Software Revisionen.

Will man ganze Sourcen-Bäume durchsuchen, so sollte man vorher `make clean` aufrufen, um nicht zusätzlich die Compiler-Ausgabefiles zu durchsuchen.

### 10.3.4.14. xargs

`xargs [COMMAND] [OPTIONS] [ARGS...]` führt `COMMAND` für jede stdin-Eingabe aus.

Optionen:

- p Prompt the user about whether to run each command
- r Do not run command for empty read lines
- x Exit if the size is exceeded
- 0 Input filenames are terminated by a null character
- t Print the command line on stderr before executing it.

Beispiel:

```
ls | xargs gzip
find . -name '*.c' -print | xargs rm
```

### 10.3.4.15. md5sum – Datenkonsistenz überprüfen

Bei Bedarf kann der Benutzer Dateien mit dem bekannten Checksummenprogramm, `md5sum` überprüfen.

Zum Erstellen einer Prüfdatei `md5sum * > checksums` ausführen. Die Gegenprobe wird mit `md5sum -c checksums` durchgeführt.

## 10.3.5. Datenfernübertragung

### 10.3.5.1. telnetd

`telnet HOST [PORT]` Telnet wird verwendet um eine interaktive Verbindung mit einem anderen Computer über das TELNET Protokoll zu erstellen.

### 10.3.5.2. rx – Empfange Xmodem Datei

`rx FILE` Empfängt ein File unter Verwendung des xmodem Protokolls.

Beispiel:

```
$ rx /tmp/foo
```

### 10.3.5.3. minicom

The Linux terminal program minicom stores its config on `/etc/minirc.dfl`.

Since minicom needs hardware access, the user must have the appropriate access permits (or `sudo minicom`).

root can give access to the serial port via `chmod`, for USB-serial-converters this works only as long the converter is not unplugged.

Minicom can be configured by `minicom -s`. Set the interface and transfer parameters 115200-8-N-1. The first integrated serial interface usually is `/dev/tty1`, the first USB-serial-converter usually is `/dev/ttyUSB0`.

The VT102 terminal emulation allows display of additional text attributes.

Save config!

Via `<ctrl>-a` and `,z'` a menu is called.

Quit the program via `<ctrl>-a ,q'`.

The setting `,W'` set automatic line wrap.

### 10.3.5.4. xterm / Konsole

Diese Programme stellen Text-Konsolen auf dem Hostsystem zur Verfügung.

## 10.3.6. Systeminformation

### 10.3.6.1. ps – Prozessliste anschauen

Um herauszufinden welche Tasks gerade auf dem System gestartet sind, ist diese Funktion nützlich. Gleichzeitig erfährt man noch die Prozessnummer, die für administrative Zwecke, z.B. kill gebraucht wird.

### 10.3.6.2. kill/killall – Prozess stoppen

Nicht mehr gewünschte Prozesse kann man mit `kill -9 prozessnummer` stoppen.

Wem die Eingabe einer Prozessnummer zu umständlich ist, der kann mittels `killall <prozessname>` den unerwünschten Prozess stoppen. `killall` funktioniert bei bestimmten Prozessen nicht, dann muss man doch mit `kill -9` arbeiten.

### 10.3.6.3. top – Systembelastung anzeigen lassen

`top` zeigt eine Liste der aktiven Prozesse und die Systembelastung an. Das Programm kann mit `,q'` verlassen werden.

### 10.3.6.4. free, df, du - Speicherplatz ermitteln

Das Kommando `free` zeigt den Füllstand des Systemspeichers (RAM und Swap-space) an.

Mit `df -h` (Diskfree, human readable) bekommt man den Füllstand der angeschlossenen Festplatten angezeigt. Eine Besonderheit auf dem CentiPad ist die Systemramdisk, deren Größe abhängig vom Füllstand ist und die bei Bedarf immer größere Teile des Hauptspeichers belegt.

Mit `du -h` (Disk Unterverzeichnis, human readable) bekommt man die Größe aller Dateien des aktuellen Verzeichnisses angezeigt.

### 10.3.6.5. proc/bootinfo

Mit `catproc/bootinfo` erfährt man das aktuelle Boot-Device.

## 10.3.7. tty – Aktive Konsolen Schnittstelle ausgeben

`tty` gibt den Namen der Schnittstelle aus die mit der Konsole verbunden ist.

## 10.3.8. date – Datum und Uhrzeit setzen

Mit dem Kommando `date -s 121413232005.27` wird das Datum auf den 14.12.2005 13:23:27h gesetzt.

## 10.3.9. time – Programmlaufzeitmessen

`time [OPTION]... COMMAND [ARGS...]`

Führt das Programm `COMMAND` mit den Argumenten `ARGS` aus. Sobald `COMMAND` sich beendet hat, wird dessen Ressourcenverbrauch angezeigt.

Optionen: `-v` zeigt umfangreicher Informationen an.

Nützlich um die Systembelastung durch bestimmte Programme zu überprüfen.

### 10.3.10. hexdump – Datei als Hexdump anzeigen

Mit `hexdump` kann man Dateien hexadezimal ausgeben lassen. `hexdump /dev/urandom` liefert eine hexadezimale Reihe von Zufallszahlen.

### 10.3.11. uptime – Betriebsdauer

Der Befehl `uptime` verrät, wie lange das System seit dem letzten Booten in Betrieb ist. Zusätzlich wird noch die Systemlast ausgegeben.

### 10.3.12. sudo – Programm als anderer User ausführen

Dieser Befehl ermöglicht die Ausführung eines Programms als anderer Benutzer – standardmäßig der Superuser.

Beispiel:

`sudo minicom` fragt nach dem root-Passwort und startet anschließend `minicom`.

### 10.3.13. watch – Programm periodisch ausführen

`watch [-n <seconds>] COMMAND...`

Führt ein Programm periodisch alle `<seconds>` aus.

Beispiel:

```
$ watch date
Mon Dec 17 10:31:40 GMT 2000
Mon Dec 17 10:31:42 GMT 2000
Mon Dec 17 10:31:44 GMT 2000
```

## 10.4. Structure of the CentiPad Linux

### 10.4.1. directories

CentiPad has the following root tree:

/	root
/bin	system programs
/dev	device driver files
/etc	config files
/lib	runtime libraries
/mnt	mount directory
/proc	hardware informations
/sbin	administration programs
/sys	hardware informations
/tmp	ramdisk for temporary files, e.g. for testing new software
/usr	application directory
/var	system variables (passwords, encryption key files)
/www	files for the webserver

Most directories are write protected. The mount command without parameters yields information about mounted directories. The type identifies a principally writable directory (tmpfs) or a static directory (squashfs).

```
root@CentiPad-v11:/# mount
/dev/root on / type squashfs (rw)
none on /etc type tmpfs (rw)
none on /var type tmpfs (rw)
none on /dev type tmpfs (rw)
none on /sys type sysfs (rw)
none on /proc type proc (rw,nodiratime)
none on /proc/bus/usb type usbfs (rw)
none on /dev/pts type devpts (rw)
none on /tmp type tmpfs (rw)
none on /mnt type tmpfs (rw)
/dev/mtdblock2 on /mnt/storage type vfat
(rw,sync,nodiratime, fmask=0022,dmask=00
22,codepage=cp437,icharset=iso8859-1)
```

Example: writable: /tmp /etc - read only: /

#### 10.4.1.1. dev directory

The dev directory contains the installed Device Interface Files. Typically the following devices are available:

can0	Canbus
dsp	Soundchip
gpio	GPIO driver Interface
i2c-0	TWI-Bus Interface
mixer	Soundchip volume
mmcblk0	SDcard
nbd1..16	Network Block Device
sound	sound interfaces
spi0..3	SPI interface Chipselect NPCS0..3

**tty0..n** by-product of the udev-systems, ttys on reserve  
**ttys0..n** AT91RM9200 serial interfaces  
**urandom** random number generator  
**vcs** virtual console memory  
**zero** zero generator

### 10.4.1.2. mnt/storage

With `centidev/make menuconfig` in the section target system, configuration a permanent storage device for CentiPad can be selected. This media provides memory for system dats (e.g. RSA-Keys) and is available for user purposes. The media is mounted on `/mnt/storage`.

### 10.4.1.3. bin

The bin directory contains executables. This is the typical installation location for user applications.

### 10.4.1.4. var

`more /var/log/messages` shows the collected kernel messages.

### 10.4.1.5. lib

lib contains the CentiDev runtime libraries. Missing libraries are a common problem for failure for executables.

```

root@CentiPad-v11:/lib# l
-rw-r--r-- 1 root root 4961 Apr 7 2006 base-etc.tar.gz
-rw-r--r-- 1 root root 4692 Apr 7 2006 base-var.tar.gz
-rwxr-xr-x 1 root root 89760 Apr 7 2006 ld-2.3.6.so
lrwxrwxrwx 1 root root 11 Apr 7 2006 ld-linux.so.2 -> ld-2.3.6.so
-rwxr-xr-x 1 root root 1125548 Apr 7 2006 libc-2.3.6.so
lrwxrwxrwx 1 root root 13 Apr 7 2006 libc.so.6 -> libc-2.3.6.so
-rwxr-xr-x 1 root root 19412 Apr 7 2006 libcrypt-2.3.6.so
lrwxrwxrwx 1 root root 17 Apr 7 2006 libcrypt.so.1 -> libcrypt-2.3.6.so
-rwxr-xr-x 1 root root 9028 Apr 7 2006 libdl-2.3.6.so
lrwxrwxrwx 1 root root 14 Apr 7 2006 libdl.so.2 -> libdl-2.3.6.so
-rwxr-xr-x 1 root root 691048 Apr 7 2006 libm-2.3.6.so
lrwxrwxrwx 1 root root 13 Apr 7 2006 libm.so.6 -> libm-2.3.6.so
-rwxr-xr-x 1 root root 69784 Apr 7 2006 libnsl-2.3.6.so
lrwxrwxrwx 1 root root 15 Apr 7 2006 libnsl.so -> libnsl-2.3.6.so
lrwxrwxrwx 1 root root 15 Apr 7 2006 libnsl.so.1 -> libnsl-2.3.6.so
-rwxr-xr-x 1 root root 14588 Apr 7 2006 libnss_dns-2.3.6.so
lrwxrwxrwx 1 root root 19 Apr 7 2006 libnss_dns.so -> libnss_dns-2.3.6.so
lrwxrwxrwx 1 root root 19 Apr 7 2006 libnss_dns.so.2 -> libnss_dns-2.3.6.so
-rwxr-xr-x 1 root root 39108 Apr 7 2006 libnss_files-2.3.6.so
lrwxrwxrwx 1 root root 21 Apr 7 2006 libnss_files.so -> libnss_files-
.3.6.so
lrwxrwxrwx 1 root root 21 Apr 7 2006 libnss_files.so.2 -> libnss_files-
2.3.6.so
-rwxr-xr-x 1 root root 67600 Apr 7 2006 libpthread-0.10.so
lrwxrwxrwx 1 root root 18 Apr 7 2006 libpthread.so.0 -> libpthread-0.10.so
-rwxr-xr-x 1 root root 60952 Apr 7 2006 libresolv-2.3.6.so
-rwxr-xr-x 1 root root 30480 Apr 7 2006 librt-2.3.6.so
lrwxrwxrwx 1 root root 14 Apr 7 2006 librt.so -> librt-2.3.6.so
lrwxrwxrwx 1 root root 14 Apr 7 2006 librt.so.1 -> librt-2.3.6.so
-rwxr-xr-x 1 root root 18884 Apr 7 2006 libthread_db-1.0.so
lrwxrwxrwx 1 root root 19 Apr 7 2006 libthread_db.so -> libthread_db-1.0.so
lrwxrwxrwx 1 root root 19 Apr 7 2006 libthread_db.so.1 -> libthread_db-
1.0.so
-rwxr-xr-x 1 root root 7836 Apr 7 2006 libutil-2.3.6.so
lrwxrwxrwx 1 root root 16 Apr 7 2006 libutil.so -> libutil-2.3.6.so
lrwxrwxrwx 1 root root 16 Apr 7 2006 libutil.so.1 -> libutil-2.3.6.so

```

## 10.4.2. environmentvariables

### 10.4.2.1. using environment variables

Die Linux-Shell verwaltet verschiedene Umgebungsvariablen, die Programmen, welche aus dieser Shell gestartet werden, zur Verfügung stehen.

Durch die Eingabe von `$PATH` kann der Zustand dieser Umgebungsvariablen abgefragt werden. Die Eingabe von `export` zeigt uns sämtliche definierten Umgebungsvariablen.

Eine Umgebungsvariable kann gesetzt und weiteren Programmen zur Verfügung gestellt werden. Eine Erweiterung des Suchpfades um das Verzeichnis `NEUERPFAD` erreicht man wie folgt:

```
export -p NEUERPFAD:$PATH
```

### 10.4.2.2. Important environment variables

**PATH** – the current search path for shell commands

### 10.4.2.3. system specific files

The CentiDev Linux-System consists of over 20000 files with more than 5 million lines of code. Linux' inherent flexibility allows the adaption of the operating system to new hardware by modification of a small part of the sources. As usual with the GPL, CentiPad is shipped with a complete set of sources, the user can adapt the system to his own requirements and contribute to the community. The CentiDev is a base for work.

Below is a short excerpt of files modified for CentiPad:

Path/filename	author/licence	description
<i>centidev/platform/linux/arch/arm/mach-at91rm9200/</i> board-centipad.c	tk/maintech/GPL	Board configuration file for allocation of board resources to Linux. (serial interfaces, USB, LAN, MMC, SPI, etc.)
<i>centidev/platform/linux/drivers/char/</i> at91_gpio.c	mk/maintech/GPL	GPIO driver
at91_gpio.h	mk/maintech/GPL	GPIO driver includes
at91_gpiodev.c	mk/maintech/GPL	GPIO driver usermode interface
<i>centidev /platform/linux/include/asm-arm/arch-at91rm9200 /</i>		this dir contains the machine specific include-files for AT91RM9200

## 10.4.3. BusyBox runtimeenvironment

### 10.4.3.1. Beschreibung

Bei jeder Linux-Installation sind zusätzlich zum Betriebssystemkern noch eine Anzahl von Hilfsprogrammen mit vorhanden. Diese Werkzeuge möchten wir in der CentiPad-Umgebung nicht missen. Um jedoch Speicherplatz zu sparen werden kompakte Versionen dieser Standardtools verwendet.

Die folgenden Absätze sollen dem Benutzer einen Überblick verschaffen. Weitere Informationen finden sich in der BusyBox-Dokumentation. Das File `centidev/platform/busybox/busybox-1.1.1/docs/BusyBox.html` gibt einen Überblick der zur Verfügung stehenden Befehle. Aufgrund des kompakten Aufbaus von Busybox eignet sich dieses File auch für den Einsteiger um die wichtigsten Linux-Kommandos kennenzulernen.

### 10.4.3.2. Busybox konfigurieren

Im Verzeichnis `/centidev/platform/busybox` kann man zum Anzeigen und Bearbeiten der aktuellen Konfiguration `make menuconfig` aufrufen.



#### 10.4.4. sysfiles / Konfigurations- und Steuerungsdateien

Ähnlich den von DOS bekannten Dateien `autoexec.bat` und `config.sys` gibt es auch unter Linux eine Reihe von Konfigurationsdateien. Diese Dateien liegen, abhängig von der jeweiligen Linux-Distribution, in bestimmten Verzeichnissen.

Die Konfigurationsdateien für das CentiPad-Linux liegen im Verzeichnis `centidev/platform/sysfiles/sysfiles/` und können dort direkt editiert werden.

Bekanntere Vertreter sind zum Beispiel `etc/init.d/rcs` z.B. für Autostart-Funktionen.

Damit ist dieses Shellskript die beste Möglichkeit, um als User das Systemverhalten zu beeinflussen, z.B. Benutzeranwendungen eintragen, um diese beim Systemstart automatisch zu starten.

Diese Dateien werden beim `make image` mitkopiert.

In Ergänzung zu diesen Dateien können Konfigurationsdateien über die Applikations-Packages hinzugefügt werden.

Die Dateien unter `etc/init.d` haben eine Nummerierung `SnnXXXX`, welche die Reihenfolge der Abarbeitung festlegt.

Das `etc`-verzeichnis enthält weitere Steuerdateien und Shellskripte für das CentiPad. Die Dateien können bei Bedarf editiert werden.

Erstellt die Image-Dateien des Zielsystems. Folgende Dateien in der Init-Ramdisk des Zielsystems werden hier erzeugt:

```
/etc/ld.so.conf
/etc/inittab
/etc/profile
/etc/fstab
/etc/group
/etc/nsswitch.conf
```

#### 10.4.5. Kernelkonfiguration

Im Verzeichnis `/platform/kernel/linux` kann man `ARCH=arm make menuconfig` aufrufen. Dies startet die Kernelkonfiguration für die ARM Architektur des CentiPad.

#### 10.4.6. Board Konfiguration

Die Datei `/home/<username>/centidev/platform/kernel/linux-`

`centipad/arch/arm/mach-at91rm9200/board-centipad.c` enthält die

Hardwarekonfiguration des CentiPad Systems. Der Kernel erhält hierüber Informationen, wie welche Ressourcen zugeordnet werden sollen. Unter Umständen wird der Anwender hier spezielle Einstellungen vornehmen wollen (z.B. Zuordnung serielle Schnittstellen physisch/logisch).

## Demo-Programme

Für CentiDev werden im Verzeichnis `centidev/platform/demos` verschieden Beispielprogramme angeboten. Diese Beispiele sind bewusst nicht im Applikations-Verzeichnis abgelegt, um die einfache Auskopplung der Demos aus dem Produktivsystem zu ermöglichen.

## 10.5. Dateisysteme

### 10.5.1. Everything is a file

Linux verwendet einen einzigen Baum für das komplette Dateisystem. Beginnend vom Root-Verzeichnis / werden weitere Elemente in den Baum gehängt – eben ge-mount-ed.

Dieses Konzept wird von Linux dermaßen konsequent durchgezogen, dass neben ‚normalen‘ Dateien auch Hardwareinterfaces (Treiber) in Files abgebildet werden.

### 10.5.2. Mounten von Filesystemen

#### 10.5.2.1. Allgemein

Für jedes Dateisystem existiert ein Dateiname – entweder ein Device-Filename (z.B. bei SDcards) oder ein echter Dateiname (bei Image-Dateien).

Mit `mount` wird dieser Dateiname an den Dateibaum angehängt mit `umount` wieder entfernt. `sync` erzwingt das Schreiben aller Pufferspeicher auf die jeweiligen Speichermedien.

#### 10.5.2.2. Mounten eines USB-Sticks

In den folgenden Zeilen wird beschrieben, wie ein USB-Stick mit FAT Filesystem gemounted wird.

Die folgenden Zeilen müssen ausgeführt sein, bevor der Stick eingesteckt wird. Bei bekanntem Bedarf werden diese in der Datei `rcS.sh` ausgeführt:

Wenn noch nicht automatisch geschehen, dann müssen unter `/dev` die mount-Knoten für den Stick erzeugt werden. Der Stick wird als SCSI-Device eingebunden, daher heißt der Knoten auch `sda`. Der SCSI Treiber hat die major number 8, `sda` ist die erste SCSI Festplatte, `sda1` ist die erste primäre Partition auf der ersten SCSI Festplatte.

```
# mknod /dev/sda b 8 0
# mknod /dev/sda1 b 8 1
```

Nach dem Einstecken kann das eigentliche Mounten erfolgen:

```
## wenn noch nicht vorhanden mount Verzeichnis für den Stick kreieren
mkdir /mnt/stick
```

```
## den Stick mit FAT-System mounten
# mount /dev/sda1 /mnt/stick
```

#### 10.5.2.3. Mounten des Datenbereichs des CentiPad Dataflashs

Das Dataflash ist in verschiedene Sektionen unterteilt.

<code>/dev/mtdblock0</code>	Bootloadercode
<code>/dev/mtdblock1</code>	Konfigurationsdaten
<code>/dev/mtdblock2</code>	Userdatenbereich, kann gemounted werden. Hier kann auch das Linux-Binary liegen, wenn das CentiPad aus dem Flash läuft.

```
mkdir /mnt/dataflash
mount /dev/mtdblock2 /mnt/dataflash
```

#### 10.5.2.4. Mounten einer SDCARD

In den folgenden Zeilen wird beschrieben, wie eine SDCARD mit FAT Filesystem gemounted wird.

Die folgenden Zeilen müssen ausgeführt sein, bevor der Stick eingesteckt wird. Bei bekanntem Bedarf werden diese in der Datei rcS.sh ausgeführt:

Im vorliegenden System sind die Mountknoten schon eingetragen,

Nach dem Einstecken kann das eigentliche Mouneten erfolgen:

```
# wenn noch nicht erzeugt
# mknod /dev/mmcbk0 b 8 0
# mknod /dev/mmcbk0p1 b 8 1
```

```
## mount Verzeichnis für die SDCARD kreieren
#mkdir /tmp/sdcard
```

```
## den Stick mit FAT-System mit allgemeinen Einstellungen mounten
# mount /dev/sda1 /tmp/stick -a
```

```
## den Stick mit FAT-System mounten, wahlweise geht auch -f
# mount /dev/sda1 /tmp/stick -f
```

Bei Tests wurde festgestellt, dass nicht alle SD-Cards erkannt werden. Bitte auf jeden Fall Testen!

#### 10.5.2.5. Mouneten einer CF-CARD

In den folgenden Zeilen wird beschrieben, wie eine SDCARD mit FAT Filesystem gemounted wird.

Die folgenden Zeilen müssen ausgeführt sein, bevor der Stick eingesteckt wird. Bei bekanntem Bedarf werden diese in der Datei rcS.sh ausgeführt:

Im vorliegenden System sind die Mountknoten schon eingetragen,

```
## mount Verzeichnis für die CFCARD anlegen
#mkdir /tmp/cfcard
```

```
## den CFCARD mit FAT-System mit allgemeinen Einstellungen mounten
# mount /dev/hda1 /tmp/cfcard
```

Bei Tests wurde festgestellt, dass nicht alle CF-Cards erkannt werden. Bitte auf jeden Fall vor dem produktiven Einsatz auf Kompatibilität testen!

#### 10.5.2.6. Mouneten eines unter Windows freigegebenen Verzeichnisses

Sobald ein Verzeichnis das unter Windows als „LinuxExchange“ freigegeben wurde kann es wie folgt in den Linux-Dateibaum unter /mnt/smb eingehängt werden:

```
# mkdir /mnt/smb
# smbmount //192.168.1.33/LinuxExchange /mnt/smb -o username=Administrator
```

Es kann wahlweise der Name des Windowsrechners „Surfer“ verwendet werden statt direkt dessen IP-Adresse. Die Option guest verhindert die Passwortabfrage (wenn der Windowsrechner entsprechend offenherzig ist):

```
$ smbmount //Surfer/LinuxExchange /mnt/smb -o guest
```

Die Verbindung kann unter Angabe des Mountpoints wieder gelöst werden:

```
$ smbmount /mnt/smb
```

### 10.5.2.7. Probleme beim automatischen Mounten

Das Starten von USB-Devices wird mittels Bootskripten gesteuert. Diese funktionieren nicht für alle Geräte gleichgut.

Bei Bedarf kann daher nach dem Systemstart noch `udevstart` aufgerufen werden. Dies löst eine erneute Initialisierung aus. Anschließend können die Geräte gemounted werden.

## 10.5.3. Image-Dateien verwenden

Zunächst muss eine n der passenden Größe erzeugt werden:

```
dd if=/dev/zero of=imagedatei bs=1048576 count=2048
```

erzeugt eine datei, die nur Nullen enthält und 2GB groß ist.

Formatiert wird die Datei mit:

```
/sbin/mke2fs -F -j -m0 imagedatei
```

### 10.5.3.1. Mounten einer Image-Datei

Mittels Loop-Devices kann Linux auch Verzeichnisbäume die in Dateien liegen einbinden. Das coLinux des Centi Pad-EWS legt seinen kompletten Dateibaum in eine Image-Datei. Einzige Ausnahme ist ein Windows-Verzeichnis, welches zum Datenaustausch hinzugemounted wird.

```
# mkdir /mnt/LinuxExchange
```

```
# mount /LinuxExchange/image.img /mnt/LinuxExchange -o loop
```

### 10.5.3.2. Kopieren von Image-Filesystemen

Will man z.B. eine größere Image-Datei für coLinux erstellen, so muss man folgende Schritte durchführen (auf einem echten Linux-System, erstellt eine neue Imagedatei mit 4GB):

Imagedatei auf Linux-Maschine kopieren.

```
dd if=/dev/zero of=neueimagedatei bs=1048576 count=4096
```

```
/sbin/mke2fs -F -j -m0 neueimagedatei
```

```
sudo mkdir /mnt/image1
```

```
sudo mkdir /mnt/image2
```

```
mount alteimagedatei /mnt/image1
```

```
mount neueimagedatei /mnt/image2
```

```
cp -pRP /mnt/image1/* /mnt/image2/
```

Neue Datei auf coLinux-System kopieren und in den Namen der vorhandenen Datei umbenennen.

## 10.5.4. Disk Exercizer

```
dd if=/dev/sda of=/dev/null;
```

Dieses kleine Shellskript kopiert Daten vom Datenträger sda zum Nulldevice. Damit kann man das System schön mit sich selbst beschäftigen und gleichzeitig testen wie sich das frisch gemountete Device verhält.

## 10.5.5. Partitionieren eines Speichermediums

Beispiel: Partitionieren und Formatieren einer SDcard

Partitionieren: `cdisk /dev/sdc`  
Formatieren: `mkdosfs /dev/sdc1`

Beispiel: Auf einer SDcard sollen zwei Partitionen angelegt werden:

`fdisk -l` Angeschlossene Festplatten anzeigen. Die SDcard wird als `/dev/sd?` angezeigt, Partitionen auf der sdcard als `/dev/sd?n`

`cdisk /dev/sdc` ruft das Partitionier und Formatier Programm auf  
`mkdosfs /dev/sdc2` formatiert die zweite Partition auf sdc mit einem Dos-Filesystem

`mount /dev/sdc2 /mnt/` zweite Partition ins Verzeichnis /mnt mounten

`umount /mnt/` Unmount device an /mnt

## 10.6. Networking

### 10.6.1. General

Most examples rely on the knowledge of the host/target IP-addresses. During development it is useful not to rely on DHCP, but use fixed IP-addresses. Setting IP-addresses on Linux is described below, on Windows the menu Network connections / TCP/IP can be used.

### 10.6.2. MAC-address

Each network adapter needs a unique MAC-address. The address is stored in the system, is read by the initialisation software and written into the network-hardware.

```
## start Ethernet-interface with MAC 00:FF:AB:CD:EF:00
ifconfig eth0 hw ether 00:FF:AB:CD:EF:00
```

Centipads use the international test range. Make sure that each device has its unique MAC.

### 10.6.3. Network configuration and start

CentiPad's network adapter is „eth0“. For starting an IP-address and a netmask is required.

```
## Network start - up activates driver
## IP Adresse: 192.168.1.100
## Netzmaske: 255.255.255.0 (class C)
ifconfig eth0 192.168.1.100 netmask 255.255.255.0 up
```

```
## Standard Router setup
route add default gw 192.168.1.1
```

sometimes a DHCP-Client restart is necessary:

```
rcnetwork restart
```

### 10.6.4. Standard user / Password

**Important:** the CentiPad standard user is **root**, the password is **admin**.

### 10.6.5. Address search / set

`nslookup hg.maintech.de` searches the IP-Adresse of hg.maintech.de. In the file `/etc/hosts` root can preset addresses, e.g. `87.106.18.11 hg.maintech.de`.

### 10.6.6. additional users on CentiPad

How set up additional users:

- the new username is **myuser** with password **1234**.
- In the file `/etc/groups` add the line `"users:x:100:"` to get a new group **users** with the **User-Id 100**.
- In the file `/etc/passwd` add the line `myuser::200:100::/var/home/myuser:/bin/ash` (200 is the unique users-id).
- `mkdir /var/home` anlegen and create a home directory for the user according to the entry in `/etc/passwd: mkdir /var/home/myuser`.
- Give directory rights to the user by: `chown -R myuser:users /var/home/myuser`.
- Create a password for the user: `passwd myuser`.

## 10.6.7. ssh – Secure Shell

SSH steht für SecureShell und gibt dem Benutzer die Möglichkeit Daten sicher über ein ungesichertes Netzwerk zu transportieren.

Wenn noch nicht geschehen, kann ssh auf dem Hostrechner mittels `sudo /etc/init.d/ssh start` gestartet werden.

Auf dem CentiPad wird SSH steht mittels `dropbear` ein SSH-Server zur Verfügung.

Die SSH bietet Zugriff auf Linux-Rechner über ein Netzwerk. Dies ermöglicht neben (oder statt) der Debug-Konsole mehrere Sitzungen mit dem CentiPad. Verwendung:

`ssh <ipadresse>`

`ssh -l root 192.168.1.100` stellt die Verbindung zu einem CentiPad mit der IP-Adresse 192.168.1.100 her.

`ssh -Y -l mh 192.168.1.100` stellt die Verbindung zu einem X11-System mit der IP-Adresse 192.168.1.40 her und ermöglicht die Übertragung von Graphikdaten.

Wahlweise ist auch die Syntax `ssh mh@192.168.1.100` möglich.

Wurde z.B. das CentiPad ausgeschaltet, während die `ssh` auf dem Hostsystem noch lief, dann kann man von einer anderen Konsole aus `killall ssh` aufrufen.

### 10.6.7.1. RSA-Schlüsselliste löschen

Wenn der Zielrechner seine RSA-Schlüssel ändert, bekommt man, je nach Einstellung, eine Warnung oder `ssh` verhindert den Zugriff komplett. In diesem Fall bekommt man noch eine Zusatzinfo, wo die RSA-Schlüsselliste gespeichert ist. Diese kann man dann einfach löschen. Beim nächsten Zugriff fragt `ssh` dann nach, ob man die neuen Schlüssel akzeptieren will.

CentiDev ist so eingestellt, dass die RSA-Keys nur bei Bedarf neu erstellt werden. Sind diese bereits vorhanden, so

## 10.6.8. scp – Secure Copy

scp provides a way of copying data between systems `scp -r <source> <destination>`.

`scp -r /home/centidev mah@192.168.1.33:/tmp` copies the dir `/home/centidev` and all subdirs to the host IP 192.168.1.33, using the remote user „mah“ and the directory `/tmp`.

Copying a single executable:

`scp ./dateiname root@192.168.1.35:/tmp` SCP asks the remote users password.

SCP simplifies the development through copying single files.

## 10.6.9. fish / winscp – Fenstergestützter Zugriff auf Dateien

Wem `scp` zu umständlich ist, für den stellt im KDE-Konqueror

`fish://mh@192.168.1.35`

eine Verbindung zu dem Rechner mit der IP-Adresse 192.168.1.35 her und meldet sich dort als Benutzer mh an. Wenn nach der IP-Adresse des Zielrechners ein Verzeichnis angegeben wird, so wechselt fish direkt dorthin.

Für Fish läuft auf dem Hostsystem ein PEARL-Script. Dieses könnte theoretisch auch in CentiPad-Linux eingebaut werden.

Das Windows-Gegenstück hierzu ist das Programm `winscp`.

## 10.6.10. telnet

If security is not paramount , a shell can be opened by

`telnet 192.168.1.108`



. Operation is similar to ssh.

### 10.6.11. Samba directories

Verzeichnis unter Windows mit Schreib/Lesezugriff freigeben.

Unter linux ein Verzeichnis suchen, wo der Share gemounted werden soll, z.b. `mkdir /mnt/net`

Dann den `smbmount //rechnerip/sharename /mnt/net -o username=Administrator` ausführen.

Den benutzernamen kannst man anpassen - je nachdem was Windows da gerne hätte, Administrator müsste eigentlich immer gehen.

### 10.6.12. wget – file transfer from HTTP/FTP-server

wget ermöglicht das Laden von Dateien über Ethernet unter Angabe einer URL. Auf dem Hostrechner muss ein Webserver laufen.

Möglicher Arbeitsablauf:

- Mit wget URL neue Daten zum CentiPad spielen
- Neu starten

Beispiel: `wget 192.168.1.32:/testfile.txt`

### 10.6.13. nc - netcat

netcat ermöglicht den Austausch von Daten zwischen Netzwerkteilnehmern.

Netcat ist unidirektional und unverschlüsselt.

Beispiel 1: Übertragen einer Datei filename an IP 192.168.1.100 über Port 9999

Sender: `nc 192.168.1.100 9999 < filename`

Empfänger: `nc -l -p 9999`

Beispiel 2: Netzwerksicherung: Sicherung aller Daten im lokalen Verzeichnis in eine Datei backup.tgz auf dem Remote Rechner.

Empfänger: `nc -l -p 9999 > backup.tgz`

Sender: `tar cvzf - * | nc 192.168.1.150 9999`

Von Netcat existiert noch eine FTP-Variante: `ncftpput` / `ncftpget`

## 11. The CentiPad Shell ASH

CentiPad uses ASH – a compact version of BASH, optimized for embedded systems.

### 11.1. Code Table

To correctly display special characters codetable „utf-8 to latin1“ or „iso8859-1“ should be selected. These settings are valid for the source-editor, the terminal-program and other programs which exchange textfiles.

### 11.2. Shellskripte erstellen

Die erste Zeile in einem Shellskript `#!/bin/sh` zeigt dem System beim Starten, welcher Interpreter verwendet werden soll.

Kommentarzeilen beginnen mit `#`.

Dateien wie z.B. Shellskripte werden unter Linux mit `chmod +x dateiname` ausführbar gemacht. Die Dateierdung ist unter Linux ohne Bedeutung.

Bekanntlich lernt man aus Beispielen am besten, daher sei ein Blick z.B. in vorhandene Skripten empfohlen.

### 11.3. Shellskripte debuggen

Aufschluss über den Ablauf eines Shellskripts gibt `set -x`. In die zweite Zeile eines Skripts eingefügt, zeigt ASH vor der Ausführung jedes Kommandos an, wie die Zeile interpretiert wurde. Die Kommandozeile wird beginnend mit einem `+` ausgegeben, inklusive aller Parameter und Variablen Auflösungen.

### 11.4. Kommandosubstitution

Die Verwendung des umgekehrten Hochkommas ``` bewirkt die Ersetzung eines Kommandos durch sein Ergebnis.

```
datestring=`date +%y%m%d%H%M`
```

### 11.5. Gültigkeitsbereiche von Variablen

Variable sind zunächst nur innerhalb eines Shellskripts gültig. Mit dem Kommando `export v2=45` steht der Wert von `v2` auf außerhalb des Skripts zur Verfügung. Da jedes Shellskript standardmäßig in einer eigenen Sub-Shell gestartet wird, muss man ein Skript mit `. <skriptname>` starten, um in der selben Umgebung zu arbeiten.

### 11.6. Programme in der selben Shell starten

Wenn der aktuelle Umgebungsvariablensatz in einem Programm benötigt wird, so kann man dieses mit ``. <programmname>'` starten. Dies ist die Kurzform von ``source <programmname>'`.

### 11.7. Beispiel: backup.sh

Folgendes Beispielskript erzeugt ein Archivfile mit dem Namen `server+<date>.tgz`. In dieses Archiv werden die Verzeichnisse `server` und `teststep` kopiert.

```
#!/bin/sh
#set -x
echo Make Backup of server-project
datestring=`date +%y%m%d%H%M`
echo $datestring
```

```
filename="server_`${datestring}.tgz`
tar -cvpf $filename server
tar -rvpf $filename teststep
ls -la $filename
echo
```

## *11.8. Editieren von Shellskripten*

Wenn Shellskripten unter Windows editiert werden, kann dies zur Fehlermeldung "bad interpreter" führen. Ursache ist die <CR><LF>-Zeilentrennung unter Windows. Beheben kann man dieses Problem, indem man unter Linux folgendes Kommando ausführt:  
`recode recode latin1/cr-lf..latin1 <windowsdatei> <linuxdatei>`

## *11.9.*

## 12. CentiPad Linux Hilfsprogramme

### 12.1. picocom

Picocom is a compact terminal program for embedded systems. Documentation is available at [centidev/platform/picocom/picocom-1.4/picocom.8.html](http://centidev/platform/picocom/picocom-1.4/picocom.8.html).

Picocom has the following commands:

Press the escape-combination first (e.g. Ctrl-a), keep ctrl pressed and then enter the command.

```
#define KEY_EXIT '\x18' /* C-x: exit picocom */
#define KEY_QUIT '\x11' /* C-q: exit picocom without resetting port */
#define KEY_PULSE '\x10' /* C-p: pulse DTR */
#define KEY_TOGGLE '\x14' /* C-t: toggle DTR */
#define KEY_BAUD_UP '\x15' /* C-u: increase baudrate (up) */
#define KEY_BAUD_DN '\x04' /* C-d: decrease baudrate (down) */
#define KEY_FLOW '\x06' /* C-f: change flowctrl mode */
#define KEY_PARITY '\x19' /* C-y: change parity mode */
#define KEY_BITS '\x02' /* C-b: change number of databits */
#define KEY_STATUS '\x16' /* C-v: show program option */
#define KEY_SEND '\x13' /* C-s: send file */
#define KEY_RECEIVE '\x12' /* C-r: receive file */
#define KEY_BREAK '\x1c' /* C-\\: break */
```

## 13. Linux Goodies

Dieses Kapitel enthält Hinweise auf im SUSE 10 OSS enthaltene Programme, die den Arbeitsalltag einfacher gestalten.

### 13.1. Editoren für Programmierer auf dem SuSe-Linux System

Unter SuSe-Linux ist eine große Auswahl brauchbarer Editoren vorhanden. Beispiele:

- kdevelop ist eine integrierte Entwicklungsumgebung ähnlich Visual Studio.
- kwrite ist ein Texteditor/Textverarbeitungsprogramm
- kate ist ein Editor ähnlich Notepad.
- xedit ist ein Editor im klassischen X11-Stil
- ein Blick auf den Editor vi lohnt sich, da dieser auf jedem Linux-System (inkl. CentiPad) vorhanden ist.

### 13.2. apropos – man-Dateien zum Thema

Tippt man `apropos <thema>` so wird eine Liste von man-Files zu diesem Thema ausgegeben.

### 13.3. ksnapshot – Screenshot

`ksnapshot` erzeugt Kopien direkt vom Linux-Bildschirm.

### 13.4. ksteak

Wörterbuch Deutsch/Englisch

## 13.5.konserve – automatisches Backup

`konserve` stellt eine graphische Oberfläche zur Verfügung, in der man Dateien oder Verzeichnisse für ein automatisches Backup auswählen kann. Wenn man `konserve` ein Zielverzeichnis angibt, dann werden automatisch tar.gz-Backups mit Datum und Uhrzeit erzeugt.

## 13.6.Nachrichten an angemeldete User schicken

An einem Host sind zwei User ‚mah‘ und ‚jk‘ angemeldet.

Mittels `mesg y` wurde Messaging eingeschaltet.

Benutzer mah tippt: `talk jk`

jk bekommt nun auf seiner Konsole die Nachricht angezeigt und kann antworten. Der Gedankenaustausch kann mit `<ctrl>-d` beendet werden.

## 13.7.thttpd – ein kleiner Webserver für das Hostsystem

Wem Apache zu aufwendig ist, dem sei ein Blick auf den `thttpd` empfohlen.

## 13.8.mbuffer – Puffer zur Datentransferoptimierung bei schnellen Laufwerken

`mbuffer` ist unter GPL verfügbar. Z.B. bei Kopiervorgängen über eine Pipe kann `mbuffer` zwischen Sender und Empfänger gehängt werden.

Beispiel: Netzwerkbackup

Sender: `tar cvzf - * | mbuffer | nc 192.168.1.150 9999`

Empfänger: `nc -l -p 9999 > backup.tgz`

## 13.9.Source management

maintech manages CentiDev with Mercurial. With the following steps this can be installed on Linux:

<http://www.selenic.com/mercurial/wiki/index.cgi/UnixInstall>

<http://www.selenic.com/mercurial/release/mercurial-0.8.1.tar.gz>

**Mercurial installational root:**

`python setup.py install # change python to python2.3 if 2.2 is default`

When Mercurial is installed, `centidev/update.sh` can be used to get the newest CentiDev version.

**Make a backup of CentiDev before each update! If update runs afoul (system crash, communication interrupt, incompatible versions), CentiDev will be damaged! Take care to recopy your application files if necessary!**

<http://www.selenic.com/mercurial/wiki/index.cgi/UnixInstall>

With low internet-bandwidth execute `centidev/make clean` and execute in each of the following directories

`centidev`

`centidev/platform/centiboot/centiboot`

`centidev/platform/kernel/linux`

`"hg pull && hg update"`.

Finally run `centidev/make world`.

## 13.10. *direct access to the maintech Source-Code-Repository*

For direct access to the source repository from SUSE:

- install "python-devel"
  - <http://www.selenic.com/mercurial/release/mercurial-0.9.1.tar.gz> download
- <http://www.selenic.com/mercurial/wiki/index.cgi/UnixInstall> install.
- check out the source tree:  
\$ hg clone <http://hg.maintech.de/centidev>
- wait...

This is only necessary for bleeding edge software and last nights bug fixes.

## 13.11. *Disassembler*

Mit `objdump -d <objectfile> <ausgabedatei>` kann man Objektcode disassemblieren.

## 14. Literaturverzeichnis

### 14.1. Programmieren

Wolf, Jürgen: *Linux – Unix Programmierung*, Galileo Press GmbH, Bonn, 1. Auflage 2005

### 14.2. Linux Treiber Entwicklung

Quade, Jürgen / Eva-Katharina Kunst: *Linux Treiber entwickeln*. dpunkt.verlag, Heidelberg, 1. Auflage 2004

### 14.3. Linux Einführung

Kofler, Michale: *Linux*, Addison Wesley, 7. Auflage – aktualisiert 2005

Willemer, Arnold: *Wie werde ich Linux Guru*, Galileo Press GmbH, Bonn, 1. Auflage 2004

<http://www.novell.com/documentation/>

SUSE Linux Reference, pdf-File

SUSE Linux Start-up, pdf-File

### 14.4. Hardware

Atmel: *Datasheet\_doc1768.pdf*. Datenblatt des AT91RM9200.

## 15. Glossar

Distribution	Die Zusammenstellung eines Linux Systems durch entsprechende Anbieter (Suse, redhat, debian, gentoo). Jede Distribution hat ihre eigenen Spezialitäten. Das CentiPad-EWS besteht gleich aus zwei Distributionen: eine für das Hostsystem und die andere die tatsächlich auf dem CentiPad läuft.
Cross Compiler	Compiler auf einem Entwicklungsrechner der Code für eine andere Systemplattform erstellen kann
ESD	Electro Static Discharge / Electrostatically Sensitive Device. Verschiedene Beschreibungen der Tatsache, dass viele elektronische Bauteile empfindlich gegenüber elektrostatischen Entladungen sind
EWS	Entwicklungssystem, in diesem Fall die Software Pakete für CentiPad inkl. aller notwendigen Programmbibliotheken und include-Files
MAC	Media Access Control – weltweit eindeutige Nummer einer Netzwerkkarte
3964(R)	Kommunikations - Protokoll, zur Punkt zu Punkt Kopplung von Industriesteuerungen über eine serielle Schnittstelle RS232/485/TTY.
CSV	Datei Comma - Separated - Values, Datenbank Dateien in denen das zum Austausch zwischen Programmen die Passwörter und Namen, die Texte für die Sprachumschaltung und die Bedienungen für den Versand der E-Mails speichert.
DHCP	Dynamic - Host - Configuration - Protocol, Dynamische Zuteilung von IPAdressen aus einem Adressenpool in einem LAN.
DNS	Domain - Name - Service, Suchdienst der im Internet zu einem Namen wie z.B. <a href="http://www.t-online.de">www.t-online.de</a> die numerische IP - Adressen findet.
FTP	File - Transfer - Protocol, FTP ist ein auf TCP/IP aufsetzendes Protokoll, das es ermöglicht, ganze Dateien zwischen zwei Netzwerkteilnehmern zu übertragen. Ungesichert!
Gateway	Gateways bzw. Router verbinden verschiedene Netze miteinander wie z.B. das LAN mit dem Internet.
GMT	Greenwich - Mean - Time
HTML	Hypertext - Markup – Language. Darstellungssprache, die über Schlüsselwörter vorgibt, wie die Inhalte im Browser angezeigt werden.
HTTP	Hyper - Text - Transfer – Protocol. Datenübertragungs Protokoll für Seiten im Internet.
IP	Internet – Protocol. Protokoll, das die Verbindung von Teilnehmern ermöglicht, die in unterschiedlichen Netzwerken positioniert sind.
IP – Adresse	Eindeutige 32Bit Adresse eines Computers in einem IP - Netzwerk wie dem Internet oder Ihrem LAN.
LAN	Lokal - Area - Network, lokales Netz innerhalb eines begrenzten Gebiets unter Anwendung eines schnellen Übertragungsmediums wie z.B.Ethernet. Eine IP - Adresse in einem LAN beginnt häufig mit 192.168.
MAC	Adresse Media - Access - Control, Die bei vielen Geräten unveränderbare, physikalische Adresse einer Netzwerkkomponente.
MTD	Memory Technologie Devices, Überbegriff für Bausteine wie Flash, RAM, EEPROM
NFS	Network File System
NTP	Network - Time - Protocol, Das NTP Protokoll dient zur Übertragung von Zeitinformationen über das Ethernet.
SMTP	Simple - Mail - Transfer - Protocol, SMTP regelt den Versand von E-Mails vom Mail - Client (DigiWEB) zum Mailserver (SMTP - Server).
SPS	peicher - Programmierbare - Steuerung.
SQL	tructured - Query - Language, Sprache zum Steuern einer Datenbank.
Subnetmask	2-Bit-Wert, der festlegt, welcher Teil der IP-Adresse das Netzwerk und welcher den Netzwerkteilnehmer adressiert.
Target	Zielsystem für die Embedded Software Entwicklung



---

TCP	Transmission - Control - Protocol, Protokoll das für eine gesicherte Verbindung der Teilnehmer während der Datenübertragung sorgt.
TCP/IP	Kombination von IP und TCP, TCP setzt auf IP auf.
URL	Uniform - Ressource - Locator, Adresse einer Internetseite.
WAN	Wide - Area – Network. Netzwerk über ein großes Gebiet wie z.B. das Internet.
WWW	World - Wide – Web. Wird häufig mit dem Internet gleichgesetzt.
XML	eXtensible - Markup - Language, erweiterbare Darstellungssprache

## 16. Offene oder fehlende Punkte

### Geplante Entwicklungen

### 16.1. Freigeben und Mounten von Netzwerk-Verzeichnissen

#### Rückfrage cd

NFS ist nicht sicher und eine recht un stabile Geschichte. Wir empfehlen daher andere Verfahren: xyz...

#### Freigabe auf NFS-Server:

##### Willemer:

Datei: **/etc/exports**

Eintrag des freizugebenden Verzeichnisses:

/tmp

Linux mitteilen, dass die Datei **/etc/exports** neu zu lesen ist.

#### Mounten auf NFS-Client:

```
# mkdir /tmp/testmnt  
# mount -f NFS //192.168.1.34/tmp /tmp/testmnt
```

#### 16.1.1. nbd – Network Block Device

Network Block Devices sind Filesysteme die in Image-Dateien auf einem Hostrechner liegen und über ein Netzwerk freigegeben werden. Ein Serverdienst ist derzeit in Entwicklung.

Auf dem CentiPad sind standardmäßig 16 NBDs verfügbar.  
Nbd-client verbindet /dev/d.

Das NBD wird mit **mount /dev/nbd1 /mnt/nbdmountname** ins System eingehängt.

#### 16.1.2. ftp

in Bearbeitung

#### 16.1.3. ftp-server

in Bearbeitung