

CentiPad Linux Handbuch



Autor: Marcus Hasenstab Version: 1.01 Datum: 0612171535 www.centipad.de



Notizen:

Etc/init.d/S15storage zeigt schön, wie Variablen auf dem System zu verschiedenem Verhalten führen. In diesem Fall STORAGE auswählen sdcard/dataflash

Änderungsliste:

- 061217 direkter Zugriff auf meintech Sourcen, Installationsvorraussetzungen unter Suse, make clean Kommentar
- 061202 image datei erzeugen, erweitert
- 061004 md5sum
- 060902 Eigene Software im CentiPad, packages
- 060826 mercurial setup Beschreibung
- 060807 sudo
- 060801 grep ausführlicher beschrieben
- 060724 cfcard mount beschrieben
- 060720 make bootupdate besser beschrieben



1. Inhaltsverzeichnis

1.	I	Inhaltsverzeichnis	3
2.	E	inführung	8
	2.1.	Fokus	8
	2.2.	Weitere Dokumentation	8
	2.3.	Sicherheitshinweise	8
	2.4.	Garantiebedingungen und Produkthaftung	8
	2.5.	Legende	9
	2.6.	Rechtliches	9
3.	E	Frste Schritte	10
	3.1.	Neue Welten	
	3.2.	Hardware Konfiguration	10
4.	S	Setup unter Linux	12
	4.1.	.1. Erstinstallation	12
	4.1.		
	4.1. 4.1.	···· · · · · · · · · · · · · · · · · ·	
		•	
•	4.2.	Setup unter Windows mit coLinux	
	4.2. 4.2.		
	4.2. 4.2.		
	4.2.		20
	4.2.		
	4.2.	-, -, -, -, -, -, -, -, -, -, -, -, -, -	
		.2.6.1. CygwinX Setup	
_			
5.	D	Der maintech Bootloader	
	5.1.	Überblick	
	5.2.	Werkseinstellung	
	5.3.	Menustruktur	25
	5.4.	Boot Menu	26
	5.5.	Boot Optionen	
	5.5.		
	5.5. 5.5.		
	5.5.		
	5.5.		
		.5.5.1. unter Linux	
		.5.5.2. unter Windows	
	5.5. 5.5.		
6.		CentiTools	
	6.1.	cbupdate	
		-	
	6.2.	ethboot	
	6.3.	genext2fs	31



6.4.	ipkg	31
6.5.	tools	31
6.6.	usbdfu	31
7. En	twicklungswerkzeuge	32
7.1.	GCC	
7.2.	Make	
7.3.	Debugging	32
7.3.1		
7.3.2 7.3.3		
	twicklungszyklus unter Linux	
8.1.	Erstellen eines kompletten Softwarepaketes für CentiPad – 33	make world
8.1.1		33
	.1.1. make world	
	.1.2. make boot	_
	.1.4. make clean	
	.1.5. make distclean	34
	1.6. make app image	
	.1.7. make kernel image	
	.1.9. make run	
	.1.10. make debug	_
	.1.11. make ddd	35
	.1.12. /centidev/make menuconfig	
8.1	.1.13. CentiDev Treiber Setup /centidev/platform/linux/make menuconfig	35
8.2.	packages - Einbinden eigener Software in das CentiPad Sys	tems 36
8.3.	Software ins Zielsystem laden	37
	.1.1. Laden des Zielsystem-RAM über TFTP – make world boot	37
	.1.2. Laden des Zielsystems über SDcard am Hostrechner	37
	.1.3. Laden des Zielsystems über SDcard im CentiPad	
	.1.4. Laden des Permanent Storage Devices	
8.3	.1.5. Laden des Zielsystem-RAM über USB - make world usbboot	
8.4.	Erstellen eines Programms	39
8.4.1		
8.4.2	,	
8.4.3		39
8.4.4		
8.4.5	. Integration eines Anwenderprogramms in die Image-Dateien	42
9. Ar	beitsablauf unter coLinux	43
9.1.	Einführung	43
9.2.	Erstellen und Laden eines kompletten Softwarepaketes für	
9.2.1	•	
9.2.2		
9.3.	coLinux Beenden	
9.4.		A -
10. I	Erstellen und Laden von Programmen für CentiPad	
	Einführung in (Embedded) Linux	47
10.1.	Einführung in (Embedded) Linux Linux Überblick	4 <i>7</i> 47
10.1. 10.2.	Einführung in (Embedded) Linux	4 <i>7</i> 47 47



	Konsole	
10.3.1.1		
10.3.1.2		
10.3.1.3		
10.3.1.4		
10.3.2.	man, info, xman	
10.3.3.	Anzeigebefehle	
10.3.3.1		48
10.3.3.2		
10.3.3.3		
10.3.4.	Verzeichnis- und Dateibefehle	
10.3.4.1		
10.3.4.2		
10.3.4.3		
10.3.4.4 10.3.4.5	,	
10.3.4.6		
10.3.4.7		
10.3.4.8		
10.3.4.9		
10.3.4.1		
10.3.4.1		
10.3.4.1		
10.3.4.1		
10.3.4.1		
10.3.4.1	5. md5sum – Datenkonsistenz überprüfen	51
10.3.5.	Datenfernübertragung	. 51
10.3.5.1		
10.3.5.2	rx – Empfange Xmodem Datei	51
10.3.5.3	minicom	51
10.3.5.4		
10.3.6.	Systeminformation	
10.3.6.1		
10.3.6.2	/ · · · · · · · · · · · · · · · · · · ·	
10.3.6.3		
10.3.6.4	free, df, du - Speicherplatz ermitteln	E 7
10.3.6.5	proc/bootinfoproc/bootinfo	52
10.3.6.5 10.3.7.	proc/bootinfotty – Aktive Konsolen Schnittstelle ausgeben	52 . 52
10.3.6.5 10.3.7. 10.3.8.	proc/bootinfo	52 . 52 . 52
10.3.6.5 10.3.7. 10.3.8. 10.3.9.	proc/bootinfo	52 . 52 . 52 . 52
10.3.6.5 10.3.7. 10.3.8.	proc/bootinfo	52 . 52 . 52 . 52 . 53
10.3.6.5 10.3.7. 10.3.8. 10.3.9.	proc/bootinfo	52 . 52 . 52 . 52 . 53
10.3.6.5 10.3.7. 10.3.8. 10.3.9. 10.3.10.	proc/bootinfo	52 . 52 . 52 . 52 . 53
10.3.6.5 10.3.7. 10.3.8. 10.3.9. 10.3.10.	proc/bootinfo	52 . 52 . 52 . 53 . 53
10.3.6.5 10.3.7. 10.3.8. 10.3.9. 10.3.10. 10.3.11. 10.3.12. 10.3.13.	proc/bootinfo tty – Aktive Konsolen Schnittstelle ausgeben date – Datum und Uhrzeit setzen time – Programmlaufzeit messen hexdump – Datei als Hexdump anzeigen uptime – Betriebsdauer. sudo – Programm als anderer User ausführen watch – Programm periodisch ausführen	52 . 52 . 52 . 53 . 53 . 53
10.3.6.5 10.3.7. 10.3.8. 10.3.9. 10.3.10. 10.3.11. 10.3.12. 10.3.13.	proc/bootinfo tty – Aktive Konsolen Schnittstelle ausgeben date – Datum und Uhrzeit setzen time – Programmlaufzeit messen hexdump – Datei als Hexdump anzeigen uptime – Betriebsdauer sudo – Programm als anderer User ausführen watch – Programm periodisch ausführen	52 . 52 . 52 . 53 . 53 . 53
10.3.6.5 10.3.7. 10.3.8. 10.3.9. 10.3.10. 10.3.11. 10.3.12. 10.3.13.	proc/bootinfo tty – Aktive Konsolen Schnittstelle ausgeben date – Datum und Uhrzeit setzen time – Programmlaufzeit messen hexdump – Datei als Hexdump anzeigen uptime – Betriebsdauer. sudo – Programm als anderer User ausführen watch – Programm periodisch ausführen. *truktur des CentiPad Linux Verzeichnisse.	52 . 52 . 52 . 53 . 53 . 53 . 53
10.3.6.5 10.3.7. 10.3.8. 10.3.9. 10.3.10. 10.3.11. 10.3.12. 10.3.13. 10.4. S	proc/bootinfo tty – Aktive Konsolen Schnittstelle ausgeben date – Datum und Uhrzeit setzen time – Programmlaufzeit messen hexdump – Datei als Hexdump anzeigen uptime – Betriebsdauer. sudo – Programm als anderer User ausführen watch – Programm periodisch ausführen truktur des CentiPad Linux Verzeichnisse dev Verzeichnis	52 . 52 . 52 . 53 . 53 . 53 . 54 . 54
10.3.6.5 10.3.7. 10.3.8. 10.3.9. 10.3.10. 10.3.11. 10.3.12. 10.3.13. 10.4. S	proc/bootinfo tty – Aktive Konsolen Schnittstelle ausgeben date – Datum und Uhrzeit setzen time – Programmlaufzeit messen hexdump – Datei als Hexdump anzeigen uptime – Betriebsdauer. sudo – Programm als anderer User ausführen watch – Programm periodisch ausführen. *truktur des CentiPad Linux* Verzeichnisse. dev Verzeichnis mnt/storage.	52 . 52 . 52 . 53 . 53 . 53 . 53 . 54 54
10.3.6.5 10.3.7. 10.3.8. 10.3.9. 10.3.10. 10.3.11. 10.3.12. 10.3.13. 10.4. S 10.4.1. 10.4.1.1 10.4.1.2 10.4.1.3	proc/bootinfo tty – Aktive Konsolen Schnittstelle ausgeben date – Datum und Uhrzeit setzen time – Programmlaufzeit messen hexdump – Datei als Hexdump anzeigen uptime – Betriebsdauer. sudo – Programm als anderer User ausführen watch – Programm periodisch ausführen *truktur des CentiPad Linux Verzeichnisse. dev Verzeichnis mnt/storage. bin Verzeichnis	52 . 52 . 53 . 53 . 53 . 53 . 54 54 55
10.3.6.5 10.3.7. 10.3.8. 10.3.9. 10.3.11. 10.3.12. 10.3.13. 10.4. 10.4.1. 10.4.1.1 10.4.1.2 10.4.1.3 10.4.1.4	proc/bootinfo tty – Aktive Konsolen Schnittstelle ausgeben date – Datum und Uhrzeit setzen time – Programmlaufzeit messen hexdump – Datei als Hexdump anzeigen uptime – Betriebsdauer. sudo – Programm als anderer User ausführen watch – Programm periodisch ausführen *truktur des CentiPad Linux Verzeichnisse. dev Verzeichnis mnt/storage. bin Verzeichnis var Verzeichnis	52 . 52 . 52 . 53 . 53 . 53 . 53 . 54 54 54 55 55
10.3.6.5 10.3.7. 10.3.8. 10.3.9. 10.3.11. 10.3.12. 10.3.13. 10.4. S 10.4.1. 10.4.1.1 10.4.1.2 10.4.1.3 10.4.1.4 10.4.1.5	proc/bootinfo tty – Aktive Konsolen Schnittstelle ausgeben date – Datum und Uhrzeit setzen time – Programmlaufzeit messen hexdump – Datei als Hexdump anzeigen uptime – Betriebsdauer. sudo – Programm als anderer User ausführen watch – Programm periodisch ausführen *truktur des CentiPad Linux* Verzeichnisse. dev Verzeichnis mnt/storage. bin Verzeichnis var Verzeichnis lib.	52 . 52 . 52 . 53 . 53 . 53 . 53 . 54 54 55 55 55
10.3.6.5 10.3.7. 10.3.8. 10.3.9. 10.3.11. 10.3.12. 10.3.13. 10.4. S 10.4.1. 10.4.1.1 10.4.1.2 10.4.1.3 10.4.1.4 10.4.1.5 10.4.2.	proc/bootinfo tty – Aktive Konsolen Schnittstelle ausgeben date – Datum und Uhrzeit setzen time – Programmlaufzeit messen hexdump – Datei als Hexdump anzeigen uptime – Betriebsdauer. sudo – Programm als anderer User ausführen watch – Programm periodisch ausführen *truktur des CentiPad Linux Verzeichnisse. dev Verzeichnis mnt/storage. bin Verzeichnis var Verzeichnis lib Umgebungsvariablen	52 . 52 . 52 . 53 . 53 . 53 . 53 . 54 54 55 55 55 55
10.3.6.5 10.3.7. 10.3.8. 10.3.9. 10.3.11. 10.3.12. 10.3.13. 10.4. S 10.4.1. 10.4.1.1 10.4.1.2 10.4.1.3 10.4.1.4 10.4.1.5 10.4.2. 10.4.2.1	proc/bootinfo tty – Aktive Konsolen Schnittstelle ausgeben date – Datum und Uhrzeit setzen time – Programmlaufzeit messen hexdump – Datei als Hexdump anzeigen uptime – Betriebsdauer. sudo – Programm als anderer User ausführen watch – Programm periodisch ausführen *truktur des CentiPad Linux Verzeichnisse. dev Verzeichnis mnt/storage. bin Verzeichnis var Verzeichnis lib. Umgebungsvariablen Verwendung von Umgebungsvariablen	52 . 52 . 52 . 53 . 53 . 53 . 53 . 54 54 55 55 55 55
10.3.6.5 10.3.7. 10.3.8. 10.3.9. 10.3.11. 10.3.12. 10.3.13. 10.4. S 10.4.1. 10.4.1.1 10.4.1.2 10.4.1.3 10.4.1.4 10.4.1.5 10.4.2. 10.4.2.1 10.4.2.1	proc/bootinfo tty – Aktive Konsolen Schnittstelle ausgeben date – Datum und Uhrzeit setzen time – Programmlaufzeit messen hexdump – Datei als Hexdump anzeigen uptime – Betriebsdauer. sudo – Programm als anderer User ausführen watch – Programm periodisch ausführen. *truktur des CentiPad Linux Verzeichnisse dev Verzeichnis mnt/storage bin Verzeichnis var Verzeichnis lib Umgebungsvariablen Verwendung von Umgebungsvariablen Wichtige Umgebungsvariablen	52 . 52 . 52 . 53 . 53 . 53 . 53 . 54 55 55 55 55 56
10.3.6.5 10.3.7. 10.3.8. 10.3.9. 10.3.11. 10.3.12. 10.3.13. 10.4. S 10.4.1. 10.4.1.1 10.4.1.2 10.4.1.3 10.4.1.4 10.4.1.5 10.4.2. 10.4.2.1 10.4.2.3	proc/bootinfo tty – Aktive Konsolen Schnittstelle ausgeben date – Datum und Uhrzeit setzen time – Programmlaufzeit messen hexdump – Datei als Hexdump anzeigen uptime – Betriebsdauer. sudo – Programm als anderer User ausführen. watch – Programm periodisch ausführen. *truktur des CentiPad Linux Verzeichnisse. dev Verzeichnis mnt/storage. bin Verzeichnis var Verzeichnis lib. Umgebungsvariablen Verwendung von Umgebungsvariablen Wichtige Umgebungsvariablen Systemspezifische Dateien	52 52 53 53 53 54 55 55 55 55 56 56
10.3.6.5 10.3.7. 10.3.8. 10.3.9. 10.3.11. 10.3.12. 10.3.13. 10.4. S 10.4.1. 10.4.1.1 10.4.1.2 10.4.1.3 10.4.1.4 10.4.1.5 10.4.2. 10.4.2.1 10.4.2.3 10.4.3.	proc/bootinfo tty – Aktive Konsolen Schnittstelle ausgeben date – Datum und Uhrzeit setzen time – Programmlaufzeit messen hexdump – Datei als Hexdump anzeigen uptime – Betriebsdauer sudo – Programm als anderer User ausführen watch – Programm periodisch ausführen **truktur des CentiPad Linux* Verzeichnisse dev Verzeichnis mnt/storage bin Verzeichnis var Verzeichnis var Verzeichnis lib Umgebungsvariablen Verwendung von Umgebungsvariablen Wichtige Umgebungsvariablen Systemspezifische Dateien Die Laufzeitumgebung BusyBox	52 . 52 . 53 . 53 . 53 . 53 . 54 55 55 55 55 56 56
10.3.6.5 10.3.7. 10.3.8. 10.3.9. 10.3.11. 10.3.12. 10.3.13. 10.4. S 10.4.1. 10.4.1.1 10.4.1.2 10.4.1.3 10.4.1.4 10.4.2.1 10.4.2.1 10.4.2.3 10.4.3. 10.4.3.1	proc/bootinfo tty – Aktive Konsolen Schnittstelle ausgeben date – Datum und Uhrzeit setzen time – Programmlaufzeit messen hexdump – Datei als Hexdump anzeigen uptime – Betriebsdauer sudo – Programm als anderer User ausführen watch – Programm periodisch ausführen **truktur des CentiPad Linux** Verzeichnisse dev Verzeichnis mnt/storage bin Verzeichnis var Verzeichnis var Verzeichnis lib Umgebungsvariablen Verwendung von Umgebungsvariablen Wichtige Umgebungsvariablen Systemspezifische Dateien Die Laufzeitumgebung BusyBox Beschreibung	52 . 52 . 53 . 53 . 53 . 53 . 54 55 55 55 55 56 56
10.3.6.5 10.3.7. 10.3.8. 10.3.9. 10.3.11. 10.3.12. 10.3.13. 10.4. S 10.4.1. 10.4.1.1 10.4.1.2 10.4.1.3 10.4.1.4 10.4.2.1 10.4.2.1 10.4.2.3 10.4.3.1 10.4.3.1 10.4.3.1	proc/bootinfo tty - Aktive Konsolen Schnittstelle ausgeben date - Datum und Uhrzeit setzen time - Programmlaufzeit messen hexdump - Datei als Hexdump anzeigen uptime - Betriebsdauer. sudo - Programm als anderer User ausführen watch - Programm periodisch ausführen **truktur des CentiPad Linux** Verzeichnisse. dev Verzeichnis mnt/storage. bin Verzeichnis var Verzeichnis lib. Umgebungsvariablen Verwendung von Umgebungsvariablen Wichtige Umgebungsvariablen Systemspezifische Dateien Die Laufzeitumgebung BusyBox Beschreibung Busybox konfigurieren	52 52 53 53 53 54 54 55 55 56
10.3.6.5 10.3.7. 10.3.8. 10.3.9. 10.3.10. 10.3.11. 10.3.12. 10.3.13. 10.4. S 10.4.1. 10.4.1.2 10.4.1.3 10.4.1.4 10.4.1.5 10.4.2. 10.4.2.1 10.4.2.3 10.4.3. 10.4.3.1 10.4.3.1 10.4.3.2 10.4.4.4.	proc/bootinfo tty - Aktive Konsolen Schnittstelle ausgeben date - Datum und Uhrzeit setzen time - Programmlaufzeit messen hexdump - Datei als Hexdump anzeigen uptime - Betriebsdauer sudo - Programm als anderer User ausführen watch - Programm periodisch ausführen wtruktur des CentiPad Linux Verzeichnisse dev Verzeichnis mnt/storage bin Verzeichnis var Verzeichnis lib Umgebungsvariablen Verwendung von Umgebungsvariablen Wichtige Umgebungsvariablen Systemspezifische Dateien Die Laufzeitumgebung BusyBox Beschreibung Busybox konfigurieren sysfiles / Konfigurations- und Steuerungsdateien	52 . 52 . 53 . 53 . 53 . 53 . 54 55 55 55 56 56 56 56 56 56 56
10.3.6.5 10.3.7. 10.3.8. 10.3.9. 10.3.10. 10.3.11. 10.3.12. 10.3.13. 10.4. S 10.4.1. 10.4.1.2 10.4.1.3 10.4.1.4 10.4.1.5 10.4.2. 10.4.2.1 10.4.2.3 10.4.3. 10.4.3.1 10.4.3.1 10.4.3.2 10.4.4.4 10.4.5.	proc/bootinfo tty - Aktive Konsolen Schnittstelle ausgeben date - Datum und Uhrzeit setzen time - Programmlaufzeit messen hexdump - Datei als Hexdump anzeigen uptime - Betriebsdauer sudo - Programm als anderer User ausführen watch - Programm periodisch ausführen watch - Programm periodisch ausführen truktur des CentiPad Linux Verzeichnisse dev Verzeichnis mnt/storage bin Verzeichnis iib Umgebungsvariablen Verwendung von Umgebungsvariablen Wichtige Umgebungsvariablen Systemspezifische Dateien Die Laufzeitumgebung BusyBox Beschreibung Busybox konfigurieren sysfiles / Konfigurations- und Steuerungsdateien Kernelkonfiguration	52 . 52 . 53 . 53 . 53 . 53 . 54 54 55
10.3.6.5 10.3.7. 10.3.8. 10.3.9. 10.3.10. 10.3.11. 10.3.12. 10.3.13. 10.4. S 10.4.1. 10.4.1.2 10.4.1.3 10.4.1.4 10.4.1.5 10.4.2. 10.4.2.1 10.4.2.3 10.4.3. 10.4.3.1 10.4.3.1 10.4.3.2 10.4.4.4.	proc/bootinfo tty - Aktive Konsolen Schnittstelle ausgeben date - Datum und Uhrzeit setzen time - Programmlaufzeit messen hexdump - Datei als Hexdump anzeigen uptime - Betriebsdauer sudo - Programm als anderer User ausführen watch - Programm periodisch ausführen wtruktur des CentiPad Linux Verzeichnisse dev Verzeichnis mnt/storage bin Verzeichnis var Verzeichnis lib Umgebungsvariablen Verwendung von Umgebungsvariablen Wichtige Umgebungsvariablen Systemspezifische Dateien Die Laufzeitumgebung BusyBox Beschreibung Busybox konfigurieren sysfiles / Konfigurations- und Steuerungsdateien	52 . 52 . 53 . 53 . 53 . 53 . 54 54 55
10.3.6.5 10.3.7. 10.3.8. 10.3.9. 10.3.11. 10.3.12. 10.3.13. 10.4. S 10.4.1. 10.4.1.1 10.4.1.2 10.4.1.3 10.4.1.4 10.4.2.1 10.4.2.1 10.4.2.3 10.4.3. 10.4.3.1 10.4.3.1 10.4.3.1 10.4.3.1 10.4.5. 10.4.5. 10.4.5. 10.4.6.	proc/bootinfo tty - Aktive Konsolen Schnittstelle ausgeben date - Datum und Uhrzeit setzen time - Programmlaufzeit messen hexdump - Datei als Hexdump anzeigen uptime - Betriebsdauer. sudo - Programm als anderer User ausführen watch - Programm periodisch ausführen watch - Programm periodisch ausführen itruktur des CentiPad Linux Verzeichnisse dev Verzeichnis mnt/storage bin Verzeichnis var Verzeichnis lib Umgebungsvariablen Verwendung von Umgebungsvariablen Wichtige Umgebungsvariablen Systemspezifische Dateien Die Laufzeitumgebung BusyBox Beschreibung Busybox konfigurieren sysfiles / Konfigurations- und Steuerungsdateien Kernelkonfiguration Board Konfiguration	52 52 53 53 53 54 55 55 55 55 55 56 56 56 56 56 56 56 55
10.3.6.5 10.3.7. 10.3.8. 10.3.9. 10.3.10. 10.3.11. 10.3.12. 10.3.13. 10.4. S 10.4.1. 10.4.1.2 10.4.1.3 10.4.1.4 10.4.1.5 10.4.2. 10.4.2.1 10.4.2.3 10.4.3. 10.4.3.1 10.4.3.1 10.4.3.1 10.4.3.1 10.4.5. 10.4.5. 10.4.6.	proc/bootinfo tty – Aktive Konsolen Schnittstelle ausgeben date – Datum und Uhrzeit setzen time – Programmlaufzeit messen hexdump – Datei als Hexdump anzeigen uptime – Betriebsdauer. sudo – Programm als anderer User ausführen watch – Programm periodisch ausführen watch – Programm periodisch ausführen truktur des CentiPad Linux Verzeichnisse. dev Verzeichnis mnt/storage. bin Verzeichnis var Verzeichnis lib. Umgebungsvariablen Verwendung von Umgebungsvariablen Wichtige Umgebungsvariablen Systemspezifische Dateien Die Laufzeitumgebung BusyBox Beschreibung Busybox konfigurieren sysfiles / Konfigurations- und Steuerungsdateien Kernelkonfiguration Board Konfiguration	52 52 53 53 54 55
10.3.6.5 10.3.7. 10.3.8. 10.3.9. 10.3.10. 10.3.11. 10.3.12. 10.3.13. 10.4. S 10.4.1. 10.4.1.2 10.4.1.3 10.4.1.4 10.4.1.5 10.4.2. 10.4.2.1 10.4.2.3 10.4.3. 10.4.3. 10.4.3.1 10.4.3.1 10.4.5. 10.4.5. 10.4.5. 10.4.5. 10.5.1	proc/bootinfo tty - Aktive Konsolen Schnittstelle ausgeben date - Datum und Uhrzeit setzen time - Programmlaufzeit messen hexdump - Datei als Hexdump anzeigen uptime - Betriebsdauer sudo - Programm als anderer User ausführen watch - Programm periodisch ausführen truktur des CentiPad Linux Verzeichnisse dev Verzeichnis mnt/storage bin Verzeichnis var Verzeichnis lib Umgebungsvariablen Verwendung von Umgebungsvariablen Wichtige Umgebungsvariablen Systemspezifische Dateien Die Laufzeitumgebung BusyBox Beschreibung Busybox konfigurieren sysfiles / Konfigurations- und Steuerungsdateien Kernelkonfiguration Board Konfiguration Pateisysteme Everything is a file	52 52 53 53 53 54 55 55 55 56 56 56 57 57 57 57 58 58
10.3.6.5 10.3.7. 10.3.8. 10.3.9. 10.3.10. 10.3.11. 10.3.12. 10.3.13. 10.4. S 10.4.1. 10.4.1.2 10.4.1.3 10.4.1.4 10.4.1.5 10.4.2. 10.4.2.1 10.4.2.3 10.4.3. 10.4.3.1 10.4.3.1 10.4.3.1 10.4.5. 10.4.5. 10.4.5. 10.5.1. 10.5.2.	proc/bootinfo tty – Aktive Konsolen Schnittstelle ausgeben date – Datum und Uhrzeit setzen time – Programmlaufzeit messen hexdump – Datei als Hexdump anzeigen uptime – Betriebsdauer sudo – Programm als anderer User ausführen watch – Programm periodisch ausführen truktur des CentiPad Linux Verzeichnisse dev Verzeichnis mnt/storage bin Verzeichnis var Verzeichnis lib Umgebungsvariablen Verwendung von Umgebungsvariablen Wichtige Umgebungsvariablen Systemspezifische Dateien Die Laufzeitumgebung BusyBox Beschreibung Busybox konfigurieren sysfiles / Konfigurations- und Steuerungsdateien Kernelkonfiguration Board Konfiguration Pateisysteme Everything is a file Mounten von Filesystemen	52 52 53 53 53 54 55 55 55 56 56 56 57 57 57 57 58 58
10.3.6.5 10.3.7. 10.3.8. 10.3.9. 10.3.10. 10.3.11. 10.3.12. 10.3.13. 10.4. S 10.4.1. 10.4.1.1 10.4.1.2 10.4.1.3 10.4.1.4 10.4.1.5 10.4.2. 10.4.2.1 10.4.2.3 10.4.3. 10.4.3.1 10.4.3.1 10.4.3.2 10.4.4. 10.4.5. 10.4.6. 10.5.	proc/bootinfo tty - Aktive Konsolen Schnittstelle ausgeben date - Datum und Uhrzeit setzen time - Programmlaufzeit messen hexdump - Datei als Hexdump anzeigen uptime - Betriebsdauer. sudo - Programm als anderer User ausführen watch - Programm periodisch ausführen truktur des CentiPad Linux Verzeichnisse. dev Verzeichnis mnt/storage. bin Verzeichnis ib. Umgebungsvariablen Verwendung von Umgebungsvariablen Wichtige Umgebungsvariablen Wichtige Umgebungsvariablen Systemspezifische Dateien Die Laufzeitumgebung BusyBox Beschreibung. Busybox konfigurieren sysfiles / Konfigurations- und Steuerungsdateien Kernelkonfiguration Board Konfiguration Board Konfiguration Board Konfiguration Boateisysteme Everything is a file. Mounten von Filesystemen Mounten eines USB-Sticks	52 52 53 53 54 55
10.3.6.5 10.3.7. 10.3.8. 10.3.9. 10.3.10. 10.3.11. 10.3.12. 10.3.13. 10.4. S 10.4.1. 10.4.1.2 10.4.1.3 10.4.1.4 10.4.1.5 10.4.2. 10.4.2.1 10.4.2.3 10.4.3. 10.4.3.1 10.4.3.1 10.4.3.1 10.4.5. 10.4.5. 10.4.5. 10.5.1. 10.5.2.	proc/bootinfo tty - Aktive Konsolen Schnittstelle ausgeben date - Datum und Uhrzeit setzen time - Programmlaufzeit messen hexdump - Datei als Hexdump anzeigen uptime - Betriebsdauer sudo - Programm als anderer User ausführen watch - Programm periodisch ausführen Werzeichnisse dev Verzeichnis mnt/storage bin Verzeichnis var Verzeichnis ilb Umgebungsvariablen Verwendung von Umgebungsvariablen Wichtige Umgebungsvariablen Systemspezifische Dateien Die Laufzeitumgebung BusyBox Beschreibung Busybox konfigurieren sysfiles / Konfigurations- und Steuerungsdateien Kernelkonfiguration Board Konfiguration Board Konfiguration Board Konfiguration Mounten von Filesystemen Mounten eines USB-Sticks Mounten des Datenbereichs des CentiPad Dataflashs	52 52 53 53 53 54 55 55 55 56 56 56 57 57 57 57 58 58



10.5.2 10.5.2		
10.5.2	.6. Probleme beim automatischen Mounten	62
10.5.3. 10.5.3	Erzeugen einer Image-Datei	
10.5.3	2.2. Kopieren von Image-Filesystemen	62
10.5.4.	Disk Exercizer	
10.5.5.	Partitionieren eines Speichermediums	
10.6.	Netzwerk	
10.6.1. 10.6.2.	GenerellesMAC-Adresse	
10.6.3.	Netzwerk konfigurieren und starten	
10.6.4.	Standardbenutzer / Passwort	
10.6.5. 10.6.6.	Adresse suchen / festlegenweitere Benutzer auf dem CentiPad anlegen	
10.6.7.	ssh – Secure Shell	
10.6.7		
10.6.8. 10.6.9.	scp – Secure Copyfish / winscp– Fenstergestützter Zugriff auf Dateien	. 65 66
10.6.10		
10.6.11		
10.6.12 10.6.13	<u> </u>	
	e CentiPad Shell ASH	
11.1.	Code Tabelle	
11.2.	Shellskripteerstellen	
11.3.	Shellskripte debuggen	
11.4.	Kommandosubstitution	_
11.5.	Gültigkeitsbereiche von Variablen	
11.6.	Programme in der selben Shell starten	
11.7.	Beispiel: backup.sh	
11.8.	Editieren von Shellskripten	
12. Ce	ntiPad Linux Hilfsprogramme	69
12.1.	picocom	. 69
13. Lir	nux Goodies	69
13.1.	Editores file December serious and dom Co.Co. Linear Costom	60
13.2.	Editoren für Programmierer auf dem SuSe-Linux System	
13.3.	apropos – man-Dateien zum Thema	. 69
	apropos - man-Dateien zum Themaksnapshot - Screenshot	. 69 . 69
13.4.	apropos – man-Dateien zum Themaksnapshot – Screenshotksteak	. 69 . 69 . 69
13.5.	apropos – man-Dateien zum Themaksnapshot – Screenshotksteakkonserve – automatisches Backup	. 69 . 69 . 69 . 70
13.5. 13.6.	apropos – man-Dateien zum Themaksnapshot – Screenshotksteakkonserve – automatisches Backup	. 69 . 69 . 69 . 70
13.5. 13.6. 13.7.	apropos – man-Dateien zum Thema	. 69 . 69 . 69 . 70
13.5. 13.6. 13.7. 13.8.	apropos – man-Dateien zum Themaksnapshot – Screenshotksteakkonserve – automatisches Backup	. 69 . 69 . 70 . 70
13.5. 13.6. 13.7. 13.8.	apropos – man-Dateien zum Thema	. 69 . 69 . 70 . 70 . 70
13.5. 13.6. 13.7. 13.8. Laufwer	apropos – man-Dateien zum Thema	. 69 . 69 . 70 . 70 . 70



14.1. Programmieren	72
14.2. Linux Treiber Entwicklung	72
14.3. Linux Einführung	72
14.4. Hardware	72
15. Glossar	<i>7</i> 3
16. Offene oder fehlende Punkte	75
16.1. Freigeben und Mounten von Netzwerk-Ve	e <mark>rzeichnissen</mark> 75
16.1.1. nbd – Network Block Device	75
16.1.2. <u>ftp</u>	
16.1.3. ftp-server	



2. Einführung

2.1. Fokus

Herzlich willkommen beim "CentiPad Linux Handbuch". Dieses Handbuch soll Ihr Wegweiser für den Einstieg in die Embedded Linux Welt von CentiPad sein. Embedded System Entwicklung ist ein vielschichtiges Thema, der Entwickler muss aus Hardware und Software eine funktionsfähige Einheit herstellen.

Es gäbe viele Wege eine Dokumentation für ein so komplexes Thema aufzubauen. Dieses Handbuch entstand aus einer Sammlung von Fragestellungen, welche der Autor bei seinem Einstieg lösen musste.

Erfahrene Anwender und Linux Fachleute werden bei der Lektüre einige Themen nur überfliegen, der Einsteiger kann jedoch ausführlich den Einstieg des Autors in die Linux-Entwicklung nachverfolgen.

2.2. Weitere Dokumentation

- CentiPad Hardware Dokumentation
- CentiPad Break Out Board Dokumentation mit Quickstart Guide
- CentiPad Break Out Board Schaltplan
- CentiPad Programmers Model
- CentiPad Applikations Handbuch
- Die neuesten Informationen sind jederzeit unter www.centipad.com verfügbar

2.3. Sicherheitshinweise

Bei Schäden, die durch Nichtbeachten der Bedienungsanleitung verursacht werden, erlischt der Garantieanspruch!

Für Folgeschäden wird keine Haftung übernommen!

Bei Sach- oder Personenschäden, die durch unsachgemäße Handhabung oder Nichtbeachten der Sicherheitshinweise verursacht werden, übernehmen wir keine Haftung!

In solchen Fällen erlischt jeder Garantieanspruch.

Verwenden Sie das Gerät nur in trockenen Räumen, in denen keine brennbaren Gase und Dämpfe vorhanden sein können.

Nehmen Sie das Gerät nicht sofort in Betrieb, wenn es von einem kalten in einen warmen Raum gebracht wurde. Das dabei entstandene Kondenswasser kann unter Umständen Ihr Gerät zerstören.

Wenn das Gerät <u>sichtbare Beschädigungen</u> aufweist, <u>nicht mehr arbeitet</u> oder <u>längere Zeit</u> unter <u>ungünstigen</u> Verhältnissen <u>gelagert</u> wurde, so ist anzunehmen, dass ein gefahrloser Betrieb nicht mehr möglich ist. An dieser Stelle ist das Gerät gegen unbeabsichtigte Inbetriebnahmen zu sichern, und falls erforderlich außer Betrieb zu nehmen.

Es dürfen keinerlei technische Veränderungen am Gerät vorgenommen werden. Da das Gerät Wärme erzeugt, ist in der Endanwendung für eine ausreichende Luftzirkulation zum Abtransport der Wärme zu sorgen.

2.4. Garantiebedingungen und Produkthaftung

Die Garantie- und Produkthaftungsansprüche, auch während der gesetzlichen Gewährleistungspflicht entfallen, wenn das Gerät nicht entsprechend den Hinweisen, beschrieben in dieser Betriebsanleitung und am Gerät, betrieben wird.



Die Garantie- und Produkthaftungsansprüche, auch während der gesetzlichen Gewährleistungspflicht entfallen, wenn das Gerät geöffnet und/oder modifiziert bzw. unsachgemäß betrieben wird.

Auf Grund Ihrer begrenzten Lebensdauer sind Teile die einem besonderen Verschleiß unterliegen, von der gesetzlichen Gewährleistungspflicht ausgenommen. Dazu gehören z.B. Batterien, Steckverbinder etc..

2.5. Legende

Verdana Fett Untertitel, markierter Text

Courier New fett Befehl oder Befehlsrückmeldung sowie Programmcode

Courier New Fett Kursiv Pfadangaben und lokale Verzeichnisse

Courier New Lenkt die Aufmerksamkeit auf wichtige oder nützliche

Hinweise

Kommentarzeichen in Shellskripten
\$ Linux Benutzer Eingabeprompt
Linux Root Eingabeprompt

Under Construction grün unterlegte Passagen sind derzeit im Aufbau

2.6. Rechtliches

Das vorliegende Werk ist in all seinen Teilen urheberrechtlich geschützt. Alle Rechte vorbehalten, insbesondere das Recht der Übersetzung, des Vortrags, der Reproduktion, der Vervielfältigung auf fotomechanischem oder anderen Wegen und der Speicherung in elektronischen Medien.

Ungeachtet der Sorgfalt, die auf die Erstellung von Text, Abbildungen und Programmen verwendet wurde, können weder der Autor noch Hersteller für mögliche Fehler und deren Folgen eine juristische Verantwortung oder irgendeine Haftung übernehmen.

Die in diesem Werk wiedergegebenen Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. können auch ohne besondere Kennzeichnung Marken sein und als solche den gesetzlichen Bestimmungen unterliegen.

Technische Änderungen vorbehalten!



3. Erste Schritte

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 Konfiguration

Aller Anfang ist schwer – sagt man. Um Ihnen den Einstieg in die neue Umgebung zu erleichtern, wird jedes CentiPad ab Werk mit einem vorinstallierten Linux ausgeliefert. In dieser Dokumentation wird als Hardwarekonfiguration ein "CentiPad pro" auf einem CentiBOB angenommen.

Bitte nehmen Sie sich vor der ersten Inbetriebnahme etwas Zeit und folgen Sie zunächst dem Kapitel "Quickstart Guide" aus der CentiPad BOB Dokumentation.

Zur Inbetriebnahme des CentiPad EWS benötigen Sie folgende Bestandteile:

- CentiPad
- CentiBOB
- CentiPad EWS CD
- Ethernet Kabel
- RS232-Nullmodem-Kabel



- Steckernetzteil
- X86 Entwicklungsrechner mit Linux oder Windows und ca. 3GB freiem Festplatten Speicherplatz

Bis auf den Entwicklungsrechner sind alle notwendigen Komponenten im CentiPad Development Kit enthalten.

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.

Checkliste:

- CentiPad richtig herum einstecken! 1 Markierung beachten!
- Kabelverbindungen herstellen
- Spannungsversorgung anschließen



4. Setup unter Linux

4.1.1. Vorraussetzungen für die Installation unter Suse

Folgende Pakete sollten im System installiert sein:

- gcc
- g++
- ncurses-devel
- bison
- flex
- alib2-devel
- zlib-devel

4.1.2. Erstinstallation

Das im folgenden beschriebene Setup wurde auf dem frei verfügbaren SUSE-Linux OSS 10.0 durchgeführt (de.opensuse.org).

Zunächst wird eine Standard-SUSE-Installation durchgeführt, z.B. unter Verwendung der KDE-Oberfläche. Da wir Software entwickeln wollen, muss bei der Installation das GCC-Paket installiert werden. Wer seine Quelltexte nicht nur mit dem vi editieren möchte, dem steht nach der Installation von kdevelop eine Entwicklungsumgebung zur Verfügung, die an Microsoft Visual Studio erinnert.

Bitte beachten Sie bei der Installation die SUSE-Dokumentation.

Das CentiPad Entwicklungssystem selbst besteht aus folgenden Teilen:

- 1) Das Paket "maintech" enthält die Systemkomponenten die auf dem x86-Host-System benötigt werden um die Images für das CentiPad zu erzeugen. Das Archiv at91-toolchain-gcc-VERSIONSNAME.tar.bz2/ wird in /opt kopiert und entpackt mittels tar xvjf <archivname> zum Verzeichnis /maintech wird unter /opt kopiert. Dieser Vorgang muss als root durchgeführt werden.
- 2) Das Paket <code>centidev-VERSIONSNAME.tar.gz</code> enthält das CentiPad-Linux-System. Aus diesem Paket wird das CentiPad-Runtime-System erzeugt, in ein Image gepackt und zum CentiPad übertragen. <code>centidev</code> wird unter <code>/home/<username> kopiert</code> und mittels tar xvpf <code>centidev-VERSIONSNAME.tar.gz</code>. Dieser Vorgang muss mit der Benutzerkennung des Entwicklers durchgeführt werden, damit tar beim auspacken automatisch die Zugriffsrechte richtig setzt.
- 3) Um beim Systemstart die Pfade für die maintech-Entwicklungsumgebung richtig zu setzen, muss das Shellscript arm-9tdmi-linux-gnu.sh nach /etc/profile.d/ kopiert werden. Um sicher zu gehen, dass dieses Shellscript geladen wird, sollte man den Rechner neu starten. Das Skript stellt die Pfade ein: 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) Für die Kommunikation mit der Debug-Schnittstelle des CentiPad muss minicom eingerichtet werden (115200-8-N-1).

Hinweis: Ergeben sich nach dem Auspacken Probleme mit den Zugriffsrechten, so kann man mit chown -R myuser:users centidev die Rechte im centidev-Verzeichnis rekursiv für den gewünschten Benutzer "myuser"setzen.

4.1.3. System kopieren / Backup

Hat man einmal ein Entwicklungssystem wunschgemäß eingerichtet, so kann man dieses System problemlos auf einen weiteren Rechner kopieren.

Im Folgenden wird eine Kopie des Linux-Systems von einer Festplatte zur anderen mittels KNOPPIX beschrieben. KNOPPIX (www.knopper.net) ist eine Linux-Distribution,



welche direkt von CD/DVD gebootet wird. Da keine Installation erfolgt, ist KNOPPIX ein ideales Administrationswerkzeug.

- KNOPPIX booten
- Platten mounten
- Backup eines Verzeichnis anfertigen:
 - o tar cvzf /zielverzeichnis/archivname.tar.gz *

Hinweis: Unter KNOPPIX muss man zuerst herausfinden, wie das Benutzerpasswort heißt. Nach dem Start ist man unter KNOPPIX automatisch root. Somit kann man im Zweifelsfall einfach in einer root-Shell mit passwd root das Passwort von root ändern. Mit diesem bekannten Passwort kann man sich nun mittels ssh von einem anderen Rechner aus anmelden.

ssh starten: sudo /etc/init.d/ssh start

4.1.4. Verzeichnisaufbau / Upgrade des Entwicklungssystems

Aktuelle Software wird unter <u>www.centipad.com</u> bzw. <u>www.maintech.de</u> zur Verfügung gestellt.

Das CentiPad EWS hat eine vordefinierte Verzeichnisstruktur. Wenn diese vom Benutzer geändert wird, so müssen Shellskripte und Pfade angepasst werden.

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.

Um nach dem Upgrade keine Probleme mit der Zugriffsverwaltung zu haben, sollte man chmod -R 0777 /home/<user>/centidev chmod -R 0777 /opt/etc/maintech aufzurufen.

/home/<user>

./centidev

- im centidev -Verzeichnis liegen das Makefile und das Menuconfig für den Bau des CentiPad-Systems. Eigene Einträge in den Dateien sollten mit dem jeweiligen Benutzernamen versehen sein, damit diese schnell für die Übertragung in neue Shellskripten identifiziert werden können
- insbesondere centidev/platform/system/sysfiles/etc enthält oft User-Modifikationen, da hier die Konfigurationsskripten für das CentiPad-System gespeichert werden.

./application

- Im application-Verzeichnis liegen typischerweise die User-Projektdateien. Die eigenen Sourcen sind zu sichern. Hier liegt ein Makefile, welches unter APPS einen Eintrag für jedes Benutzerprojekt erhält. Menuconfig enthält Steuerinformationen für das Build-System.
- In jedem Projektverzeichnis liegt ein Makefile für jeweils dieses Projekt

./package

- Beim Erstellen des Firmwarepakets für die Zielplattform werden Packages angelegt. Zusätzlich zu den Benutzerprogrammen können auch Shellscripte und zusätzliche Verzeichnisse erstellt werden. Verzeichnisse und Dateien aus package werden in die Binaries für das Target-System eingebaut.

./target

 Dieses Verzeichnis enthält die übersetzten Daten für das Target-System, sowohl als einzelne Dateien, sowie als Binär-Pakete für den Upload (centipad.bin, initrd, vmlinux).

./platform



- das platform-Verzeichnis enthält das komplette CentiPad-Linux. In diesem Verzeichnis werden Änderungen an der Struktur des Entwicklungssytems eingetragen
 - ./kernel/linux
 - ./drivers
 - hat der Benutzer eigene Treiber geschrieben, so sind deren Sourcen zu sichern
 - in jedem driver-Unterverzeichnis existiert ein Makefile und Kconfig. Diese Dateien enthalten Informationen für die Kernelkonfiguration für die Benutzertreiber. Die Einträge in diesen Dateien sichern!

/opt/maintech

- das maintech-Verzeichnis enthält die Crosscompilertools. Diese sind vom Benutzer normalerweise nicht zu ändern.

/etc/profile.d

 dieses Verzeichnis enthält das Shellskript arm-9tdmi-linux-gnu.sh, welches die Umgebungsvariablen für das Entwicklungssystem setzt. Bei einem Systemupgrade ist zu überprüfen, ob das Shellskript geändert wurde und bei Bedarf angepasst werden muss.

4.1.5. Backup Strategien unter Linux

Das Standard-Entwicklungssystem besteht unter Linux aus folgenden Verzeichnissen und Dateien:

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

Am einfachsten ist es diese Verzeichnisse jeweils mit tar cvzf archivname_datum.tgz quellverzeichnis

zu archivieren. Dabei bleiben automatisch alle Berechtigungen der Files erhalten. Anschließend können die Files z.B. mit кзъ (benötigt evtl. Supervisor-Rechte) auf eine CD gebrannt werden.



4.2. Setup unter Windows mit coLinux

4.2.1. coLinux

coLinux ist eine frei verfügbare virtuelle Maschine, die Linux innerhalb eines Windows 2000/XP Systems ausführt. coLinux kommt ohne Einträge in die Registry aus und ist somit ideal für den Einsatz auf Arbeitsrechnern geeignet, bei denen keine Modifikation an der bestehenden Konfiguration gewünscht wird.

coLinux läuft aus einer Datei, die das komplette File-System enthält. Bei Schäden an diesem System kann der Entwickler einfach wieder die Originaldatei zurückspielen. Komplettsicherungen werden hiermit extrem vereinfacht. Außerdem kann gerade in der Startphase nichts kaputt gemacht werden.

Die Software-Entwicklung kann mit Windows-Werkzeugen durchgeführt werden. Einzig für die Cross-Compilierung wird ein spezieller Gnu-Compiler unter coLinux benötigt.

Hierzu ist eine Ethernet-Treiberinstallation notwendig.

4.2.2. Standardbenutzer

Das Wichtigste kurz und bündig:

Der Standard-User unter coLinux ist centidev, sein Passwort ist centidev.

Selbst wenn unter coLinux ein root Benutzer zur Verfügung steht – dieser darf <u>nie</u> verwendet werden. Die dadurch geänderte Rechtestruktur führt früher oder später zu update- und build-Problemen.

4.2.3. Installation

Hinweis: Da das komplette Linux-Filesystem in einer Image-Datei mit fester Größe (z.B. 2GB) liegt, muss vor der Installation sicher gestellt sein, dass ausreichend freier Speicherplatz zur Verfügung steht.

Starten Sie die Datei mt-centidev-xxx-setup.exe.



Die nachfolgende Installation fragt zunächst nach Installationsort und welche Icons installiert werden sollen.



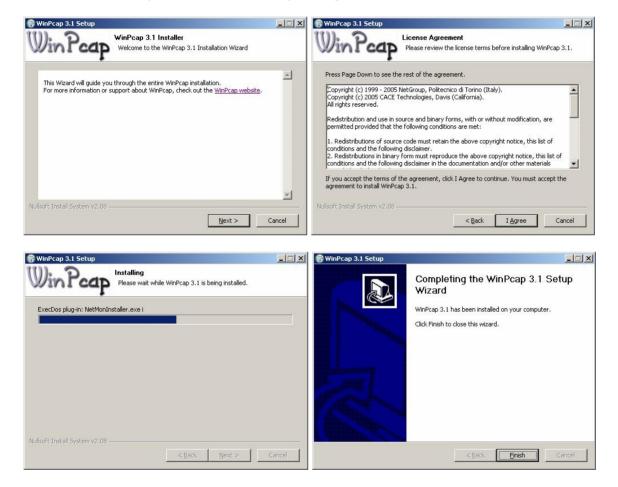


Der TAP-Treiber muss für Netzwerkunterstützung installiert werden. Die unten stehende Meldung erscheint und muss mit "Installation fortsetzen" quittiert werden. Der TAP-Treiber ist für die Verbindung zwischen coLinux und dem Host-Windows vorgesehen. Wird benötigt, wenn kein physisches Netz vorhanden ist.





Als nächstes wird WinPcap installiert, welches die Kommunikation zwischen coLinux und dem LAN (und damit CentiPad) ermöglicht.

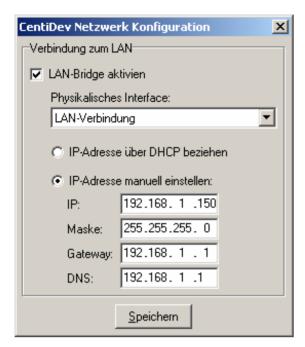


Nachdem Wincap installiert ist, öffnet das netconfig Programm. Dieses kann u.U. von anderen Fenstern verdeckt werden, daher bitte in den Vordergrund holen.

Im NetConfig die **physikalische LAN-Verbindung** zum CentiPad auswählen und die Konfiguration der virtuellen Schnittstelle einstellen.

Hinweis: diese Einstellung muss zum Installationszeitpunkt richtig sein, sonst funktioniert CentiDev nicht und eine Neuinstallation ist notwendig.





Nachdem die Netzwerkkonfiguration gespeichert wurde, startet automatisch coLinux. Die folgende Initialisierung läuft vollautomatisch, kann aber je nach System eine gute halbe Stunde dauern.

```
GentiDev Development System (c) maintech GmbH 2006

CentiDev final installation steps are running.
Please be patient while CentiDev creates and populates your development system.
This is a good moment to fetch a fresh cup of coffee...
...from the other side of town

--- Step 1/4: creating system image (size 1536 MB)
this is going to take a while...ok
--- Step 2/4: creating filesystem inside image
doesn't take too long...ok
--- Step 3/4: unpacking file system archive
please have a bit more patience...ok
--- Step 4/4: returning to Windows installation

The system is going down NOW !!
Sending SIGTERM to all processes.
Sending SIGKILL to all processes.
Sending SIGKILL to all processes.
The system is halted. Press Reset or turn off power
```

coLinux beendet sich nach der Installation automatisch. Die folgende Box signalisiert das Ende der Installation:







4.2.4. Backup Strategien unter Windows/coLinux

Nach der Standardinstallation sind folgende Verzeichnisse und Dateien vorhanden:
- die Standardinstallation von coLinux findet im Verzeichnis C:\Programme\maintech
CentiDev statt. Dieses Verzeichnis enthält sowohl coLinux als auch den TAP und
winpCAP Treiber. Das Unterverzeichnis \hostfs enthält das Linux Dateisystem in der
Datei rootfs.

Eine einfache Backup Strategie unter Windows kann wie folgt aussehen:

- alle Dateien des Linux-Systems befinden sich in der coLinux-Container-Datei. Ein Backup dieser Datei stellt sicher, dass das Entwicklungssystem einfach wiederhergestellt werden kann.
 - In der Standardinstallation ist diese Datei als C:\Programme\maintech
 CentiDev\hostfs\rootfs zu finden. Diese ist standardmäßig 1500MByte groß
 und kann je nach Inhalt auf unter 400MByte gezippt werden
- Colinux hat ein gemeinsames Arbeitsverzeichnis mit Windows. Dieses Verzeichnis kann genutzt werden, um Daten zwischen coLinux und Windows auszutauschen. Dieses Verzeichnis ist unter Windows in der Standardinstallation C:\Programme\maintech CentiDev\hostfs\. Unter coLinux ist dieses Verzeichnis als /windows zu sehen. Hinweis: Da Windows die normale Rechtevergabe eines Linux-Systems nicht kennt, können von diesem Verzeichnis keine Linux-Programme gestartet werden.
- Um Daten mit dem coLinux-System auszutauschen, kann das gemeinsame Verzeichnis zwischen Windows und coLinux verwendet werden.
- Der einfachste Zugriff geht über den Windows-Explorer. Unter Netzwerkumgebung\Gesamtes Netzwerk\Microsoft Windows-Netzwerk\Centidev finden sich die unter coLinux freigegebenen Verzeichnisse.
- Ein volles Backup hat den Vorteil, dass im Bedarfsfall der gesicherte Zustand durch das Rückspielen einer Datei ermöglicht wird. Wenn nur wenige Sourcen zu sichern sind, wird man einer Teilsicherung den Vorzug vor einer 2GB großen Containerdatei geben.

Unter coLinux kann man ein tar-Archiv des centidev Verzeichnisses erstellen: tar cvzf /windows/centidev-<datum>.tgz/home/centidev/centidev
Hiermit wird ein gepacktes Archiv des Entwicklungssystems im Windows-Datenaustausch Verzeichnis erstellt.

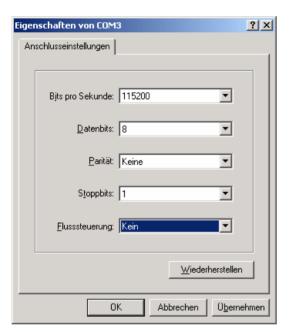
Wahlweise kann der Benutzer die zu sichernden Programme über den Windows-Explorer aus dem coLinux heraukopieren.

4.2.5. Hyperterminal

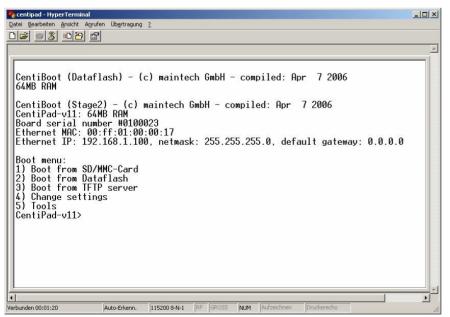
Hyperterminal ist Bestandteil von Windows XP. Im Startmenu ist es normalerweise unter Zubehör\Kommunikation zu finden. Starten Sie Hyperterminal und richten Sie eine neue Verbindung auf der seriellen Schnittstelle, die mit dem CentipAd verbunden ist ein. Die Verbindungsparamter sind 115200-8-N-1, kein Handshake.













4.2.6. Cygwin

Cygwin ist ein Werkzeug welches unter WindowsXP eine Sammlung Linux-ähnlicher Befehle bereitstellt, komplett mit einer Command Shell.

Mittels ssh ermöglicht Cygwin daher den Zugriff auf beliebige Linux-Rechner im Netzwerk – also auch auf ein CentiPad.

4.2.6.1. CygwinX Setup

Cygwin besteht aus einem Textteil "Cygwin" und einem Graphikteil "CygwinX". Zuerst muss Cygwin von http://www.cygwin.com/ installiert werden. Danach wird CygwinX von x.cygwin.com installiert. Bitte unbedingt die Bedienungsanleitung beachten. Angenommen CygwinX wurde im Verzeichnis c.ygwin installiert. Nachdem die X11-Bibliotheken und ssh installiert wurden, kann man die Datei c.\cygwin\usr\X11R6\bin\startwin.bat kopieren und für die eigene Anwendung anpassen. In der Datei werden unter dem Absatz "REM Startup the programs" folgende Zeilen eingefügt:

```
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 -1 <username> 192.168.1.37
```

Damit wird ein X11 initialisiert und der Benutzer mh automatisch beim System 192.168.1.37 angemeldet. Nun kann man hier mit **term&* ein X-Terminal starten.

4.2.6.2. Cygwin für CentiPad

Für den Zugriff auf's CentiPad reicht eine normale Cygwin Installation aus. Das Skript startwin.bat wird nicht benötigt. ssh -1 <username> 192.168.1.100 stellt die Verbindung zu einem CentiPad mit der IP-Adresse 192.168.1.100 her.





5. Der maintech Bootloader

5.1. Überblick

CentiPad wird ab Werk mit dem komfortablen maintech-BootLoader ausgeliefert. Aufgabe des Bootloaders ist es, das CentiPad nach dem Einschalten zu Konfigurieren und anschließend das Betriebssystem vom eingestellten Bootmedium zu laden.

Der Bootloader verwendet standardmäßig folgende Festwertspeicher bzw. Bootdatenträger:

- Konfigurations-EEPROM
- Ethernet
- SPI-DataFlash
- miniSDCARD

5.2. Werkseinstellung

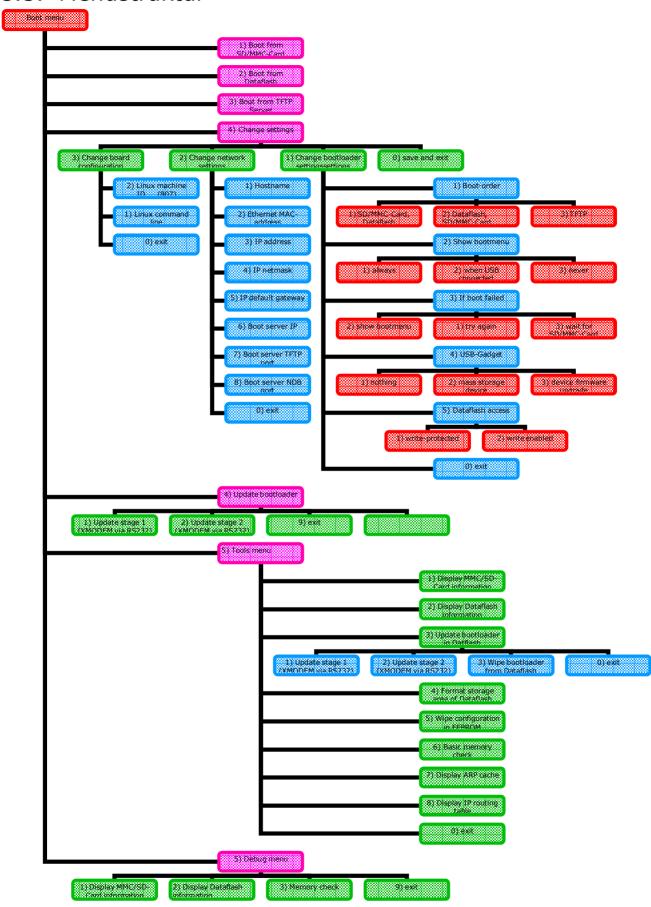
Ab Werk sind folgende Einstellungen im Konfigurations-EEPROM gespeichert:

- Board ID: 807, dieser Code teilt dem Kernel mit, dass er auf einem CentiPad ausgeführt wird
- Die Seriennumer: 0104711 ist z.B. das CentiPad pro mit der Seriennummer 4711
- Die MAC-Adresse: ab Werk ist hier eine Adresse im Bereich 00:FF:xx:xx:xx eingetragen. Die unteren Ziffern der MAC-Adresse entsprechen der Seriennummer, hex-codiert.
- TCP/IP Informationen
- ein vorinstalliertes Linux wird aus dem DataFlash geladen
- das USB Gadget ist auf DFU konfiguriert

Die serielle Debug-Schnittstelle wird für die Konsole verwendet. Die Schnittstellen Einstellungen sind 115200-8-N-1. Diese Einstellung wird vom System auch während des Betriebs beibehalten.



5.3. Menustruktur





5.4. Boot Menu

Boot Menu aufrufen:

Power-Taste halten und beim Erscheinen des Textes "Release power button to enter boot loader menu!" loslassen.

```
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>
```

Das Boot Menu ermöglicht die Auswahl des Bootmediums und ein Update der Boardkonfiguration.

5.5. Boot Optionen

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. Booten über DFU konfigurieren

Steht keine Netzwerkverbindung zur Verfügung, so ist ein Booten über USB möglich. Das verwendete DFU-Verfahren ist jedoch bezüglich der Bus-Topologie sehr empfindlich und wird daher für die Entwicklung nicht empfohlen.

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

Nach dem Einschalten wartet das CentiPad nun auf den USB-DeviceFirmwareUpdate. Die Image Dateien werden durch das Shellskript usbupload.sh übertragen. Anschließend startet das CentiPad-Linux automatisch.

5.5.3. Booten von SDcard

Wird das System seltener geändert, dann ist ein Booten von SDcard sinnvoll. In dieser Konfiguration startet das CentiPad beim Einschalten automatisch. Wird die SDcard im Linux gemounted, dann kann ein neues Linux z.B. über ssh auf die SDcard kopiert werden.

Durch aktivieren des Massenspeicher-USB-Gadget gibt der Bootloader beim Aufruf des Bootmenus die SDcard und das Dataflash über USB frei. Damit können Daten direkt vom Hostbetriebssystem auf die Speichermedien kopiert werden.

- 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. Bootloader neu konfigurieren

Ist bei der Entwicklung etwas schief gegangen und es stehen unbrauchbare Daten in Konfigurations-EEPROM und Dataflash, dann kann man ein Starten des Bootloaders per Hardware verhindern.

J12 ist ein 100Ω Widerstand. Durch Brücken der zum Board-Rand ausgerichteten Seite mit Masse kann das Booten einer defekten Firmware verhindert werden (SPI-MISO wird auf der CPU-Seite kurzgeschlossen).

Nach dem Starten des CentiPad muss der Kurzschluss entfernt werden. Das Board verhält sich nun wie ein CentiPad mit leerem DataFlash.

Die weitere Vorgehensweise entspricht der "Inbetriebnahme eines blanken Boards".

5.5.5. Zur Inbetriebnahme eines blanken Boards

Die folgenden Schritte beschreiben einen Teil der Werksinbetriebnahme. Der Anwender wird dies nur bei beschädigten Bootloader-Daten durchführen müssen.

5.5.5.1. unter Linux

Unter centidev/centitools/cbupdate befindet sich das Programm centipad-coin. Dieses ermöglicht die automatische Neuinstallation eines CentiPad. Bei diesem Vorgang werden auch Seriennummer und MAC-Adresse neu geschrieben. Der letzte Parameter nennt den Namen der Image-Datei, die der Bootloader starten soll (Dataflash oder SDcard).

Syntax:

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

Ein Befehl:

- ./centipad-coin /dev/ttyUSB0¬
- ../../target/centiboot/stage1_xmodem_centipadv11.bin¬
- ../../target/centiboot/stagel_dataflash_centipadvl1.bin¬



../../target/centiboot/stage2.bin¬
0100010¬
00:ff:01:00:00:0A¬
centipad.bin

5.5.5.2. unter Windows

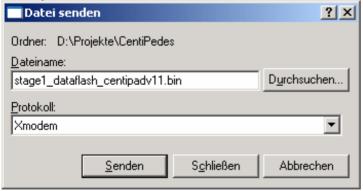
- Hyperterminal auf 115200-8-N-1 einstellen.
- CentiPad einschalten. Im Terminalfenster erscheinen 'C'-Buchstaben die Kennung für XMODEM
- mit Hyperterminal/XMODEM das stage1_xmodem_centipadv11.bin senden



- danach **stage2.bin** senden



- es erscheint das Bootloader Boot Menu
- hier "4) Update bootloader" auswählen
- 1) Update stage 1 (XMODEM via RS232) auswählen, dann **stage1_dataflash_centipadv11.bin** senden (ACHTUNG! Nicht das gleiche File wie oben!)



- 2) Update stage 2 (XMODEM via RS232) direkt danach auswählen und stage2.bin hochschieben
- Reset durchführen



- Dann das Board unter 3) Change settings / Board-Config konfigurieren
- 1) Linux machine ID (807)
- 2) Ethernet MAC-address (00:FF:00:00:01) <- Original MAC-Adresse wieder einstellen
- die Bootloader-Settings nach Bedarf einstellen, z.B.:
- 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...
 - centipad.bin mit einem Kartenlesegerät oder mit dem Bootloader auf die SD-Karte kopieren
 - Reset durchführen
 - CentiPad booted von SDcard

5.5.6. Booten von Dataflash

Im produktiven Betrieb wird man das Linux direkt vom Dataflash booten. Dies entspricht dem Auslieferungszustand von CentiPad.

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

5.5.7. Bootloader für dataflash boot initialisieren

Das Dataflash ist ab Werk formatiert und enthält ein Linux. Das Dataflash kann wie folgt neu formatiert werden.

Beispiel unter Windows:

- Reset durchführen
- Bootloader einstellen
 - Boot menu / Change setting / Change bootloader / USB-Gadget / massstorage
 - Boot menu / Change setting / Change bootloader / Dataflash access / write enabled
- Reset durchführen
- Boot Menu aufrufen
- Unter Windows erscheint die Meldung "neue Hardware gefunden"
- Im Explorer erscheinen die SDcard und das Dataflash als Wechselmedien
- Beim Formatieren ist das Dateisystem "FAT" die Option "Schnellformatierung" zu wählen. Die Low-Level-Speichertests führen beim Dataflash immer zu Fehlermeldungen.





- Wurde das Formatieren erfolgreich abgeschlossen, kann nun centipad.bin aufs Dataflash kopiert werden
- Einstellung im Bootloader überprüfen
 - o Boot menu / Change bootloader / Bootorder / Dataflash
- Reset durchführen

Beispiel unter Linux:

- die Vorgehensweise unter Linux ist analog zur Windows-Variante
- Dataflash formatieren und centipad.bin kopieren
- Da das Dataflash 4MB groß ist, wird als Filesystem FAT12 verwendet
- Formatieren mit mkdosfs -v /dev/sdx
- Mounten mit /dev/sdx /mnt/<mountverzeichnis>

Beispiel auf dem CentiPad:

- das Dataflash kann direkt im Bootloader formatiert werden



6. CentiTools

Das Verzeichnis centitools enthält verschiedene Werkzeuge für die Arbeit mit CentiPad.

6.1. cbupdate

Werkzeuge für die CentiPad Inbetriebnahme, u.a. cbupdate und centipad-coin.

6.2. ethboot

Programm zum Finden von CentiPads und Booten über Ethernet.

6.3. genext2fs

Dieses Werkzeug erzeugt auf dem Hostsystem ein ext2fs-Filesystem in einer Datei, ohne daß man loop-mounten muß

6.4. ipkg

Paketverwaltungswerkzeug

6.5. tools

Shellskripte und Hilfsdateien für die Erstellung der Binaries.

6.6. usbdfu

USB-DFU-Werkzeug welches für make usbboot benötigt wird.



7. Entwicklungswerkzeuge

7.1. GCC

GCC ist der Standard-Compiler/Linker unter Linux. Im CentiPad-Build-System werden zwei Varianten dieses Compilers eingesetzt: die x386-Variante, welche die Host-Utilities aus den Quellen des Build-Systems erzeugt und der ARM-Crosscompiler, welcher die ausführbaren Programme für das CentiPad erzeugt. Die Beschreibung des GCC sei der einschlägigen Fachliteratur vorbehalten. Siehe auch man gcc.

Der Crosscompiler ermöglicht das Erstellen von C-Programmen für CentiPad.

7.2. Make

Make wird verwendet um ganze Programmpakete automatisch zu übersetzen. Ruft man Make einfach mit make auf, so verwendet Make die Datei Makefile im aktuellen Verzeichnis.

Möchte man z.B. mehrere Makefiles im selben Verzeichnis anlegen, so kann man das Makefile für die CentiPad Version Makefile nennen und die Version für das Host-

Debugging MakefileHostDebug. Ein Aufruf von make -f MakefileHostDebug erstellt dann die Debugversion des Programms.

Siehe auch man make.

CentiDev enthält einige umfangreiche Makefiles, welche den kompletten Entwicklungszyklus inklusive automatischem Download zum Target unterstützen.

7.3. Debugging

7.3.1. Hostentwicklung

Normalerweise werden Programme auf einem Computer mit derselben Architektur wie das Zielsystem entwickelt. Der Programmierer kann sein Programm übersetzen und direkt starten und testen.

CentiPad verwendet Linux als Betriebssystem, folglich können Programme prinzipiell auf jedem Linux-Rechner getestet werden. Dies funktioniert solange, wie das Programm keinen Zugriff auf spezielle CentiPad-Hardware benötigt (z.B. TWI-Bus, CAN-Bus, Peripheribus). Häufig bietet es sich an, zumindest den Applikationsrumpf auf einem Linux-Host zu entwickeln und anschließend die Implementation der speziellen Hardwarefunktionen durchzuführen.

7.3.2. Printf-Debugging

Zum Debuggen der der Anwendung auf dem Target-System bietet sich im einfachsten Fall das printf-Debugging an – der Entwickler gibt Statusinformationen mittels printf auf der Konsole aus. Weitere Möglichkeiten sind fprintf(stderr, "Fehlertext\n") und printk zur Ausgabe von Kernel-Fehlermeldungen.

7.3.3. GDB

Zum GCC-Entwicklungssystem gehört der Gnu Debugger GDB. Auf dem CentiPad kann der gdbserver gestartet werden.

Beispiel: CentiPad ist mittels Ethernet mit dem Host verbunden, das Programm lauflicht soll untersucht werden.

Ablauf:

- Host: Bau des Programms mit Debug-Informationen
- Host: make image boot



CentiPad: gdbserver:9999 /usr/bin/lauflicht das Programm wird mit vollem
 Pfad gestartet, der Server hört auf den Port 9999, das CentiPad habe die 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-linuxgnu/lib" -ex "symbol-file lauflicht" -ex "target remote 192.168.1.100:9999"
-ex "break main" -ex "cont"!

- damit wird der DDD auf dem Host im lauflicht-Verzeichnis gestartet. Wichtig ist die Information, dass der ARM-Debugger verwendet werden soll
 - Host/Target: Debuggen

Der gdbserver beendet sich entweder automatisch mit dem verlassen des Programms oder kann von einer anderen Shell aus mittels killall gdbserver gestoppt werden. Siehe auch make debug.

8. Entwicklungszyklus unter Linux

8.1. Erstellen eines kompletten Softwarepaketes für CentiPad – make world

Das CentiPad Entwicklungssystem wird mit Sourcen für ein konsistentes Softwarepaket geliefert. Bei der ersten Inbetriebnahme kann der Anwender dieses Softwarepaket übersetzen und auf dem CentiPad in Betrieb nehmen (bitte Installations-Kapitel beachten). Dieses Softwarepaket bildet die stabile und getestete Basis für die Entwicklung von Anwenderprogrammen.

8.1.1. centidev/make

CentiDev verwendet leistungsfähige Makefiles zur Steuerung der Firmwareerstellung. Zentraler Punkt sind die beiden Files centidev/Makefile und centidev/Menuconf.

8.1.1.1. make world

Das Target Verzeichnis wird gelöscht. Anschließend wird das komplette CentiDev Paket von den Sourcen an gebaut. Ergebnis ist das File centipad.bin.

Mit dem Aufruf werden automatisch zwei Image-Dateien erzeugt – vmlinux enthält den Linux-Kernel und initrd die Init-Ramdisk mit dem Linux-Filesystem, zusammen mit den Anwenderprogrammen. Abgelegt sind diese Dateien unter

centidev/target/
vmlinux
initrd
bzw. kombiniert als
centipad.bin

Ein kompletter Neubau des CentiPad Entwicklungssystems ist z.B. notwendig:

- wenn das Entwicklungsystem neu installiert wurde
- wenn Änderungen an der Runtime-Umgebung (z.B. in etc/init.d) durchgeführt wurden
- um das Embedded-Linux-System für die Auslieferung vorzubereiten

Der erste Aufruf von make world startet die Übersetzung des kompletten Linux-Systems. Dies kann auf aktuellen Rechnern zwischen 5..10 Minuten dauern. Weitere Aufrufe werden üblicherweise in weniger als einer Minute durchgeführt.



Vor dem ersten Aufruf sind unbedingt die Einstellungen gemäß der Erstinstallationsanleitung zu beachten (speziell unter Linux). Falsche Einstellungen beim Erststart können das Build-System unbrauchbar machen. Dies erfordert eine Neuinstallation und anschließende Überprüfung der Einstellungen.

8.1.1.2. make boot

Startet ethboot mit dem dem aktuellen ./target/centipad.bin. Wird idealerweise als make world boot aufgerufen.

8.1.1.3. make usbboot

Bootet das Target über den USB-Port mittels DFU. DFU muss hierzu im Target-Bootloader aktiviert sein.

8.1.1.4. make clean

Säubert die Source-Pfade und löscht das Target-Verzeichnis. make clean führt auf manchen Plattformen zu einem defekten CentiDev. Reparatur über update.sh.

8.1.1.5. make distclean

Säubert noch gründlicher als make clean. Nach make distclean müssen Quelldateien über das Internet neu bezogen werden.

Das Shellscript update.sh führt diesen Vorgang automatisch durch.

8.1.1.6. make app image

Baut nur den Unterbaum application neu in centipad.bin ein. Ideal für die Applikationsentwicklung.

8.1.1.7. make kernel image

Baut nur den Unterbaum kernel neu in centipad.bin ein. Ideal für die Kernelentwicklung.

8.1.1.8. make bootupdate

Lädt einen neuen Bootloader über die serielle Schnittstelle in das Target. Das Target muss hierzu im Bootloader-Menu sein.

Die serielle Schnittstelle wird mit make menuconfig eingestellt. Während des Uploads dürfen keine anderen Programme, z.B. minicom auf die serielle zugreifen.

Die Fehlermeldung "tcflush() failed: Input/output error" signalisiert Probleme mit der seriellen Schnittstelle.

Dieses Update funktioniert derzeit nur unter linux. Unter coLinux/Windows ist das Update über Hypertem/xmodem möglich.

8.1.1.9. make run

Dieses Kommando ist in den Makefiles der Anwender-Projekte eingebaut. Wird make run aufgerufen, so wird die entsprechende Applikation gebaut, in das CentiPad geladen und gestartet. Bei einem erneuten Aufruf von make run wird die laufende Instanz gestoppt und durch die neue Version ersetzt.

8.1.1.10. make debug

Dieses Kommando ist in den Makefiles der Anwender-Projekte eingebaut. Wird make debug aufgerufen, so wird das Anwenderprogramm ins CentiPad geladen und mit dem GDB Debug Server gestartet. Auf dem Host wird der GDB gestartet.



8.1.1.11. make ddd

Dieses Kommando ist in den Makefiles der Anwender-Projekte eingebaut. Wird make debug aufgerufen, so wird das Anwenderprogramm ins CentiPad geladen und mit dem GDB Debug Server gestartet. Auf dem Host wird der GDB mit dem graphischen Frontend DDD gestartet.

8.1.1.12. /centidev/make menuconfig

Mit make menuconfig kann der Benutzer das CentiDev auf seine eigenen Anforderungen einstellen. Die Bedienung entspricht den aus der Kernelentwicklung bekannten Verfahren. Genauere Informationen ergeben sich direkt aus den Menus. Die zugehörige Steuerdatei ist centidev/Menuconf.

Jedes Unterverzeichnis besitzt ein eigenes Makefile/Menuconf – Paar, welches die Konfiguration des Zielsystems ermöglicht.

CentiDev enthält eine Vielzahl von Programmpaketen. Diese kann man mittels centidev/make menuconfig auswählen.

Beispiele:

CentiDev Package Selection Busybox

8.1.1.13. CentiDev Treiber Setup /centidev/platform/linux/make menuconfig

CentiDev enthält eine Vielzahl von Treibern. Diese kann man mittels cd centidev/platform/kernel/linux/
ARCH=arm make menuconfig
auswählen.

Hinweis: Eingriffe in den Kernel haben weitreichende Folgen. Vor jedem Zugriff auf jeden Fall ein Backup machen! Bei Start ohne ARCH=arm kann das Buildsystem beschädigt werden, da dann die Konfiguration für für das x86-Hostsystem aufgerufen wird.



8.2. packages - Einbinden eigener Software in das CentiPad Systems

Das CentiDev baut bei einem make image das File centipad.bin aus verschiedenen Pakten zusammen. Die Benutzerapplikation ist im Paket centidev/application/package untergebracht. Die Daten im package-Verzeichnis werden ins root-Filesystem des CentiPad kopiert. Eine Datei die z.B. in /usr/bin erscheinen soll, bekommt in Ihrem Makefile den Eintrag INSTALLPATH=/usr/bin. Beim Build wird das entsprechende Binary in application/package/usr/bin kopiert. Beim make image werden die Dateien an die entsprechende Stelle im root-Filesystem übertragen.



8.3. Software ins Zielsystem laden

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

Wenn ein CentiPad im Bootloader-Menu 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 Netzwerksegement des Hostsystem verbinden
- CentiPad über Debug-TTY mit dem Host verbinden (für Konsole)
- Bootloader wartet nach dem Einschalten im Menu

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. Laden des Zielsystems über SDcard im CentiPad

Systemkonfiguration:

- Hostrechner mit Linux oder WindowsXP
- CentiPad über Debug-TTY mit dem Host verbinden (für Konsole)
- CentiPad ist über Ethernet mit dem Host verbunden (für Datenübertragung)
- Bootloader ist für Systemstart von SDcard konfiguriert
- SDcard ist gemounted z.B. unter /mnt/sdcard
- ssh-Server dropbear läuft auf dem CentiPad

Entwicklungszyklus:

- 1) Quelltext schreiben
- 2) make world starten
- 3) Dateien centipad.bin aus dem target-Verzeichnis mittels scp auf die SDcard kopieren. Auf der Host-Konsole:
 - scp centipad.bin root@192.168.1.40:/mnt/sdcard (siehe scp)



- 4) CentiPad mittels reboot neu starten. reboot führt einen sauberen Neustart durch. Selbst wenn scp auf dem Host beendet ist, dauert das Schreiben auf die SDcard noch an.
- 5) CentiPad Linux startet automatisch Software testen

8.3.1.4. Laden des Permanent Storage Devices

Wenn man im vorherigen Abschnitt die scp Zeile austauscht, so kann man das Image direkt in das aktuelle permanten storegae device laden, welches je nach Einstellung das Dataflash oder die SDcard ist. Für die regelmäßige Entwicklung wird man das Dataflash aufgrund der Alterung und verhältnismäßig geringen Geschwindigkeit nicht verwenden. 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. Erstellen eines Programms

8.4.1. Übersicht

Dieses Kapitel beschreibt die Vorgehensweise bei der Entwicklung eigener Anwendungsprogramme. Grundlage hierfür ist die erfolgreiche Installation des Build-Systems.

Der typische Entwicklungszyklus besteht aus folgenden Schritten:

- 1) Konzept erstellen
- 2) Quelltext Schreiben
- 3) Quelltext Übersetzen
- 4) Ausführbare Dateien aufs Zielsystem laden
- 5) Testen
- 6) Einfügen der neuen Programme ins Build-System
- 7) System Images erzeugen
- 8) Komplettsystem testen

Je nach Entwicklungsaufgabe kommen mehrere Arbeitsabläufe in Frage, die in den folgenden Abschnitten vorgestellt werden.

8.4.2. Einbindung von Programmen ins Build-System

Das Build-System enthält ein Unterverzeichnis **application**. Alle Anwenderprogramme sind hier in separaten Unterverzeichnissen abgelegt. Dies ermöglicht eine Komplettsicherung der Anwenderprogramme mit der Sicherung des application-Verzeichnisses (z.B. tar cvzf srcarchiv.tgz application).

Das Verzeichnis centidev/application enthält ein Makefile welches vom Build-System aufgerufen wird. Dieses Makefile besteht aus den Sektionen all, build, package, clean und distclean.

Der Benutzer erstellt in der Zeile APPS eine mittels Leerzeichen getrennte Liste von Applikationen, die in die Pakete aufzunehmen sind.

Jedes Anwenderprogramm besitzt wiederum in seinem Unterverzeichnis ein Makefile, welches den Bau dieses Programms beschreibt.

8.4.3. "Hello World" auf dem CentiPad

Der C-Quelltext für "Hello World" auf dem CentiPad besteht aus folgenden Zeilen:

```
#include <stdio.h>
int main(int argc, char** argv)
{
   printf("Hello World!\n");
   return 0;
}
```

Ruft man im Verzeichnis centidev /application/helloworld nun den Linux-Compiler gcc wie folgt auf: gcc -o helloworldhost helloworld.c, so erzeugt gcc eine ausführbare Datei für den Hostrechner mit dem Namen helloworldhost.

Um Programme für das CentiPad zu erzeugen, enthält das Build-System im Verzeichnis /opt/maintech einen Crosscompiler für den ARM-Prozessor des CentiPad. Um diesen Crosscompiler korrekt aufzurufen, enthält das Verzeichnis centidev /application/helloworld ein Makefile zum Erzeugen von ausführbaren Programmen für das CentiPad.



Listing eines Makefiles inkl. Parameterübergabe:

```
TARGET=serialtest
INSTALLPATH=/usr/bin
DEBUG=n
RUNARGS=/dev/tty3 "SerialTest"
   _SOURCES=serialtest.c
  XX_SOURCES=
                                                                                                   -I../../target/staging/include/centidev -
 CXXFLAGS=-I../../target/staging/include
  ../../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
                ../target/staging/bin/g++-uc
 CC=$(MT_CROSS_COMPILE)gcc
LD=../../target/staging/bin/g++-uc
 COBJS=$(C_SOURCES:%.c=%.o)
 CXXOBJS=$(CXX_SOURCES:%.cpp=%.o)
DEPS=$(CXX_SOURCES: %.cpp=%.dep) $(C_SOURCES: %.c: %.dep)
FULLDEPS=$(addprefix $(DEPDIR)/, $(DEPS))
ifeq ($(DEBUG),y)
                  CFLAGS+=-00 -g3
                 CXXFLAGS+=-00
                  LDFLAGS+=-g3
 else
                  CFLAGS+=-02
                 CXXFLAGS+=-02
                  LDFLAGS+=-s
 endif
build: $(DEPDIR) $(TARGET)
-include $(FULLDEPS)
                  T): $(CXXOBJS) $(COBJS)
$(LD) $(LDFLAGS) -0 $(TARGET) $(CXXOBJS) $(COBJS) $(LIBS)
                $(CXX) $(CXXFLAGS) -MD -MP -MF $(patsubst %.o,$(DEPDIR)/%.dep, $@) -c $(patsubst
  s.o,%.cpp, $@) -o $@
                   \$(CC) \$(CFLAGS) - MD - MP - MF \$(patsubst \$.o,\$(DEPDIR)/\$.dep, \$@) - c \$(patsubst \$.o,\$.c, MPDIR) + (patsubst \$.o,\$.c, MPDIR) + (patsubst \$.o, \$.c, MPDIR) + (patsubst \$.o, MPDIR) + (patsubst \$.o,
 $@) -o $@
$(DEPDIR):
                  @mkdir -p $(DEPDIR)
                  mkdir -p ../package/$(INSTALLPATH)
                  cp -v $(TARGET) ../package/$(INSTALLPATH)
                 rm -Rf $(CXXOBJS) $(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
              -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)
```

Das Makefile besteht aus zwei Abschnitten. Im oberen Abschnitt werden folgende Informationen angegeben:

- der Name der ausführbaren Datei angegeben (TARGET=helloworld)
- der Installationspfad der ausführbaren Datei auf in der initrd (INSTALLPATH=/usr/bin)
- eine durch Leerzeichen getrennte Liste der C/C++-Quelldateien (helloworld.c)
- eine Liste der Compilerflags
- Bibliotheksinformationen
- Linkerinformationen

Der untere Abschnitt enthält Arbeitsinformationen für den Crosscompiler und muss vom Anwendungsentwickler normalerweise nicht verändert werden.

Gibt man im Verzeichnis *centidev /application/helloworld* nun make ein, so erzeugt gcc eine ausführbare Datei für das CentiPad mit dem Namen helloworld.

8.4.4. Testen des Anwenderprogramms "Hello World"

In diesem Kapitel wird ein schneller Programmierzyklus beschrieben, mit dessen Hilfe Anwendungsprogramme auf dem Hostsystem Übersetzt werden und in die Ramdisk /tmp eines laufenden CentiPads kopiert werden. Ein kompletter Übersetzungs/Test-Zyklus kann damit ohne Bau der Image-Dateien und CentiPad Neustart durchgeführt werden.

Systemkonfiguration:

- Hostrechner mit Linux oder WindowsXP
- CentiPad über Debug-TTY mit dem Host verbinden (für Konsole)
- CentiPad ist über Ethernet mit dem Host verbunden (für Datenübertragung)
- Bootloader ist f
 ür Systemstart von SDcard konfiguriert
- SDcard ist gemounted z.B. unter /mnt/sdcard
- ssh-Server dropbear läuft auf dem CentiPad

Entwicklungszyklus:

- 1) Quelltext im Verzeichnis centidev/application/helloworld schreiben
- 2) centidev/application/helloworld/Makefile wie beschrieben anpassen
- 3) make starten
- 4) Zieldatei centidev/ application /helloworld/helloworld scp auf die CentiPad-Ramdisk kopieren: Auf der Host-Konsole: scp helloworld root@192.168.1.100:/tmp (siehe scp)
- 5) Zum Test Anwendungsprogramm auf dem CentiPad starten: /tmp/helloworld.



8.4.5. Integration eines Anwenderprogramms in die Image-Dateien

Wie bereits beschrieben ruft centidev/make world das centidev/application/Makefile auf. Hier macht man folgende Einträge:

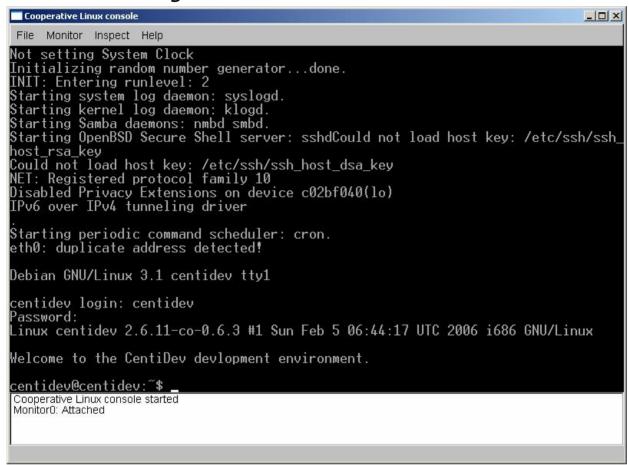
APPS=helloworld

Damit und mit Hilfe der im <code>centidev/application/helloworld/Makefile</code> gemachten Einträge wird das Anwendungsprogramm <code>bin/helloworld</code> in die <code>initrd</code> aufgenommen und steht nach dem <code>reboot</code> des CentiPad mit der neuen Image Datei zur Verfügung.



9. Arbeitsablauf unter coLinux

9.1. Einführung



Der Arbeitsablauf unter coLinux ist derselbe wie unter Linux. Durch die Verwendung virtuellen coLinux Maschine sind jedoch ein paar zusätzliche Punkte zu beachten.

Jeder Entwickler hat seinen gewohnten Arbeitsablauf. Dem Windows-Entwickler ermöglicht coLinux den Einsatz seiner vertrauten Werkzeuge. Letztlich wird jedoch Software für ein Linux-System entwickelt und die Übersetzung geschieht daher auf dem virtuellen System coLinux.

Beim Start von coLinux wird in einer virtuellen Maschine wird nun ein Linux-System gebootet, was je nach Hostsystem ein Weilchen dauern kann. coLinux öffnet zwei Fenster. Im Fenster **maintech CentiDev** werden keine Benutzereingabe gemacht, man kann es minimiert in der Taskleiste ablegen. Arbeiten finden in der **Cooperative Linux Console** statt.



```
Checking connection: LAN-Uerbindung
Bridged listening on: UIA 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 multicast) or (ip broadcast or multicast)
input: AI Translated Set 2 keyboard on cokbd
NET: Registered protocol family 2
IP: routing cache hash table of 1024 buckets, 8Kbytes
ICP established hash table entries: 8192 (order: 4, 65536 bytes)
ICP bind hash table entries: 8192 (order: 3, 32768 bytes)
ICP: Hash tables configured (established 8192 bind 8192)
NET: Registered protocol family 1
NET: Registered protocol family 17
EXI3 FS on cobd0, internal journal
EXI3-fs: recovery complete.
EXI3-fs: mounted filesystem with ordered data mode.
UFS: Mounted root (ext3 filesystem).
Freeing unused kernel memory: 108k freed
kjournald starting. Commit interval 5 seconds
EXI3 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!
```

Außerdem wird die Cooperative Linux Console gestartet – das Arbeitsfenster von coLinux. Das Kernel und das Dateisystem Centipad Linux Entwicklungssystems liegt in der Imagedatei rootfs. Wenn dieses gestartet wurde, sitzt man vor einer Linux-Konsole bei der man sich mit Benutzername centidev und Passwort centidev anmelden kann. Nach der Anmeldung findet man dasselbe Linux Entwicklungssystem wie im Kapitel "Entwicklung unter Linux" beschrieben.

9.2. Erstellen und Laden eines kompletten Softwarepaketes für CentiPad

9.2.1. Images erstellen

Im Verzeichnis /home/centidev/centidev erzeugt man mit ./make world die CentiPad-Linux Image Datei centipad.bin in /home/centidev/centidev/target.

9.2.2. Kopieren auf Bootloader-Massenspeicher

Systemkonfiguration:

- Hostrechner mit Windows XP / coLinux
- 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 und als Massenspeicher konfiguriert

Entwicklungszyklus:

- 1) Quelltext schreiben
- 2) make world starten
- 3) Dateien unter coLinux mit cp /home/centidev/centidev/target/centipad.bin /windows ins Windows-Dateisystem kopieren
- 4) Reset am CentiPad auslösen
- 5) Dem Hostrechner einige Sekunden Zeit lassen, um das DFU-Gadget zu enumerieren
- 6) Dateien mit dem Windows-Explorer auf das gewünschte Bootmedium kopieren
- 7) CentiPad Linux startet Software testen



Hinweis: Um das auf dem CentiPad vorinstallierte Linux zu erhalten, sollte für Experimente eine SDcard als Bootmedium verwendet werden. Damit steht bei Bedarf immer ein lauffähiges CentiPad zur Verfügung.

Steht eine Netzwerkverbindung zwischen coLinux und CentiPad zur Verfügung, kann man die Image-Dateien wie im Kapitel "Entwicklungszyklus unter Linux" beschrieben direkt auf das Bootmedium kopieren.

9.3. coLinux Beenden

Um coLinux zu beenden kann man sudo halt tippen. Danach das Konsole-Fenster schließen.

Oder man schließt das Konsole-Fenster direkt, dann sorgt coLinux intern für einen Shutdown.

9.4. Erstellen und Laden von Programmen für CentiPad

Wie Programme ins CentiPad-Buildsystem eingebunden werden, wurde bereits im Kapitel "Entwicklungszyklus unter Linux" beschrieben.

Steht eine Netzwerkverbindung zwischen coLinux und CentiPad zur Verfügung, kann man Programme zum Testen direkt in die CentiPad Ramdisk /tmp kopieren.

Ohne coLinux-Netzwerkverbindung kann man jedoch wie folgt vorgehen: Man kopiert das Quellverzeichnis des Anwenderprogramms in das Datenaustauschverzeichnis:

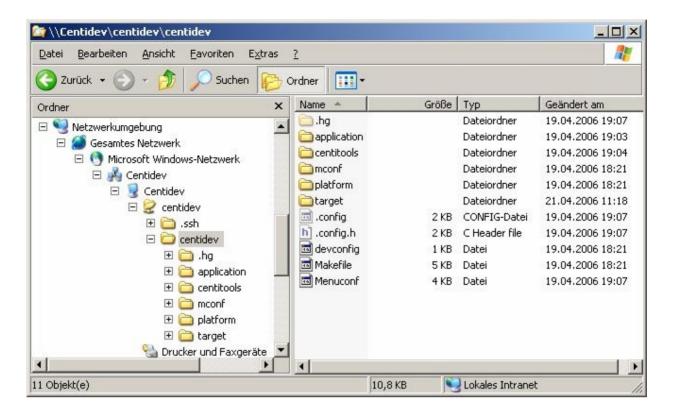
cp -r /home/centidev/centidev /application/helloworld /windows
Hier kann man das Programm übersetzen: /windows/helloworld/make
Nun kann das ausführbare Programm z.B. mit WinSCP in die CentiPad Ramdisk /tmp
kopiert werden. Da Windows die Rechtevergabe von Linux nicht unterstützt, ist ein
Programm aus dem /windows Verzeichnis nicht lauffähig. Nach dem Kopieren müssen auf
dem CentiPad daher noch die Rechte mittels chmod +x 0777 helloworld gesetzt
werden.

Den Vorteil den man durch die Arbeit im /windows Verzeichnis gewinnt ist, dass der Zugriff auf dieses Verzeichnis durch Windows Programme möglich ist. Man kann seine Quelltexte daher mit einem beliebigen Windows Editor erstellen.

Über den Windows-Explorer erhält man Zugriff auf das home-Verzeichnis des Benutzers centidev. Die gewünschten Dateien können wie gewohnt mittels Doppelklick geöffnet werden. Die notwendigen Linux-Rechte bleiben automatisch erhalten.

Beispiel: Netzlaufwerk verbinden, \\192.168.1.150\centidev verbindet mit dem Entwicklungssystem.







10. Einführung in (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 Standardbenutzer

Das Wichtigste kurz und bündig:

Der Standard-User auf dem CentiPad ist root, sein Passwort ist 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:

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 verketten.

Beispiel:

1s -1 | more Leitet die Ausgabe des Is-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. Is - 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 Quellverzecinis und dessen Unterverzeichnissen/-dateien erstellen:

tar cvzf archivdatei.tgz quellverzeichnis

Archiv auspacken:

tar xvzf archivdatei.tgz zielverzeichnis

Ein zusätzliches z in der Parameterlsite 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 iner 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 Dateinnamen an.

Optionen:

- -w Suchmuster muss ein Wort sein
- -i ignoriere Gross/Kleinschreibung
- -r Rekursiy in Verzeichnissen suchen
- -1 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 Such 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 Checksummenprogram, md5sum überprüfen.

Zum Erstellen einer Prüfdatendatei 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 Protokols.

Beispiel:

\$ rx /tmp/foo

10.3.5.3. minicom

Minicom wird so eingerichtet, dass es beim Start automatisch die erste serielle Schnittstelle öffnet und versucht über diese Kontakt zum CentiPad herzustellen. Die Konfigurationsinformation hierzu werden standardmäßig unter /etc/minirc.dfl abgelegt.

Da minicom Zugriff auf Hardware benötigt, muss entweder der Benutzer mit diesen Rechten ausgestattet sein oder man verwendet minicom als Superuser.

Der Superuser kann auch mit chmod dem user Zugriff auf den verwendeten seriellen Port geben – das hält bei USB-Seriell-Umsetzern aber nur so lange wie der Stecker im USB-Port ist.

Wenn noch keine minicom-Konfiguration vorhanden ist, kann mit minicom -s der Konfigurationsmodus eingestellt werden. Nun kann man die gewünschte Schnittstelle mit 115200-8-N-1 konfigurieren. Die erste integrierte serielle Schnittstelle ist z.B.

/dev/tty1, der erste USB-Seriell-Umsetzer hat die Bezeichnung /dev/ttyusb0. Im PC eingebaute serielle Schnittstellen haben die Bezeichnung tty<Schnittstellennummer>. Die voreingestellte Terminalemulation VT102 ermöglicht die Darstellung von erweiterten Text-Attributen.

Anschließend Konfiguration speichern.



Über <ctrl>-a und anschließend z kann ein Menu aufgerufen werden. Verlassen des Programms ist über <ctrl>-a und anschließend 'q' möglich. Für die Verwendung empfiehlt sich die Einstellung 'W' für einen automatischen Zeilenumbruch.

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 cprozessname> 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 Swapspace) 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 cat proc/bootinfo erfährt man das aktuelle Boot-Device.

10.3.7. tty - Aktive Konsolen Schnittstelle ausgeben

tty gibt den Namen der Schnittstell 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 - Programmlaufzeit messen

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

Führt das Programm COMMAND mit den Argumenten ARGS aus. Sobald COMMAND sich beendet hat, wird dessen Resourcenverbrauch 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 all <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. Struktur des CentiPad Linux

10.4.1. Verzeichnisse

Ein Blick in das Wurzelverzeichnis des CentiPad zeigt folgenden Dateibaum:

```
Wurzelverzeichnis des Dateibaums
           System-Programme
/bin
           Device-Treiber-Dateien
/dev
/etc
           Konfigurationsdateien
           Bibliotheken des Laufzeitsystems
/lib
           Mount-Verzeichnis
/mnt
           Hardwareinformationen
/proc
           Administrationsprogramme
/sbin
/sys
           Hardwareinformationen
           Ramdisk für temporäre Dateien, z.B. zum Testen von neuer Software
/tmp
           Verzeichnis für Applikationen
/usr
           Systemvariable (Passwörter, Schlüsseldateien)
/var
           Dateien für den Webserver
/www
```

Die meisten Verzeichnisse sind während der Laufzeit nicht beschreibbar. Der Befehl mount ohne Parameter gibt Informationen über gemountete Verzeichnisse zurück. An der type kann man erkennen, ob es sich um ein prinzipiell schreibbares Dateisystem (tmpfs) oder um eine statisches Dateisystem (squashfs) handelt.

```
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,iocharset=iso8859-1)
```

Beispiel: Schreibbar: /tmp /etc - nur lesen: /

10.4.1.1. dev Verzeichnis

Das dev-Verzeichnis enthält die installierten Device Interface Files. Typischerweise sind folgende Devices verfügbar:

can0	Canbus
dsp	Soundchip Aufnahme
gpio	GPIO-Treiber Interface
i2c-0	TWI-Bus Interface
mixer	Soundchip Lautstärke
mmcblk0	SDcard
nbd116	Network Block Device
sound	Verzeichnis für sound-Interface



spi0...3 SPI Interface Chipselect NPCS0...3

tty0..n Nebenprodukt des udev-Systems, ttys auf Reserve

ttys0..n serielle Schnittstellen des AT91RM9200

urandom Zufallszahlengenerator
vcs virtual console memory
zero Unendlicher Vorrat an Nullen

10.4.1.2. mnt/storage

In centidev/make menuconfig kann unter Target System, configuration ein Permanent storage device für das CentiPad festgelegt werden. Dieses Medium dient als Speicher für Systemdaten (z.B. RSA-Keys) und steht dem Benutzer für eigene Zwecke zur Verfügung. Das Medium wird auf dem CentiPad unter /mnt/storage gemounted.

10.4.1.3. bin Verzeichnis

Das bin Verzeichnis enthält ausführbare Programme. Dies ist der typische Installationsort für eigene Applikationen.

10.4.1.4. var Verzeichnis

Mit more /var/log/messages kann man sich die gesammelten Kernel Messages anschauen.

10.4.1.5. lib

Dieses Verzeichnis enthält die ins CentiDev eingebundenen Bibliotheken. Wenn Applikationen auf bestimmten Zielsystemen nicht laufen, kann man hier die installierten Bibliotheken herausfinden.

```
root@CentiPad-v11:/lib# 1
-rw-r--r-- 1 root
                            root.
                                           4961 Apr 7 2006 base-etc.tar.gz
                                          4692 Apr 7 2006 base-var.tar.gz
89760 Apr 7 2006 ld-2.3.6.so
-rw-r--r--
                1 root
                            root
-rwxr-xr-x
                1 root
                            root
                                              11 Apr 7 2006 ld-linux.so.2 -> ld-2.3.6.so
lrwxrwxrwx 1 root
                           root
                1 root
                                      1125548 Apr 7 2006 libc-2.3.6.so

13 Apr 7 2006 libc.so.6 -> libc-2.3.6.so

19412 Apr 7 2006 libcrypt-2.3.6.so
                            root
-rwxr-xr-x
lrwxrwxrwx
                1 root
                            root
-rwxr-xr-x 1 root root
                                           17 Apr 7 2006 libcrypt.so.1 -> libcrypt-2.3.6.so 9028 Apr 7 2006 libdl-2.3.6.so
             1 root root
1 root root
1rwxrwxrwx
-rwxr-xr-x
                                              14 Apr 7 2006 libdl.so.2 -> libdl-2.3.6.so
lrwxrwxrwx 1 root
                           root
               1 root
                           root
                                       691048 Apr 7 2006 libm-2.3.6.so
13 Apr 7 2006 libm.so.6 -> libm-2.3.6.so
-rwxr-xr-x
lrwxrwxrwx
               1 root
                            root
                                          69784 Apr 7 2006 libnsl-2.3.6.so
-rwxr-xr-x 1 root root
                                           15 Apr 7 2006 libnsl.so -> libnsl-2.3.6.so
15 Apr 7 2006 libnsl.so.1 -> libnsl-2.3.6.so
              1 root
1 root
                            root
root
lrwxrwxrwx
1rwxrwxrwx
                                           14588 Apr 7 2006 libnss_dns-2.3.6.so
19 Apr 7 2006 libnss_dns.so -> libnss_dns-2.3.6.so
19 Apr 7 2006 libnss_dns.so.2 -> libnss_dns-2.3.6.so
-rwxr-xr-x 1 root
                           root
lrwxrwxrwx
                1 root
                            root
              1 root
lrwxrwxrwx
                            root
                                           39108 Apr 7 2006 libnss_files-2.3.6.so
                            root
-rwxr-xr-x 1 root
                                              21 Apr 7 2006 libnss_files.so -> libnss_files-
lrwxrwxrwx 1 root
                            root
.3.6.so
                                               21 Apr 7 2006 libnss_files.so.2 -> libnss_files-
lrwxrwxrwx 1 root
                             root
2.3.6.so
                                           67600 Apr 7 2006 libpthread-0.10.so
             1 root
-rwxr-xr-x
                            root
             1 root
                                           18 Apr 7 2006 libpthread.so.0 -> libpthread-0.10.so 60952 Apr 7 2006 libresolv-2.3.6.so 30480 Apr 7 2006 librt-2.3.6.so
lrwxrwxrwx
                            root
                1 root
                             root
-rwxr-xr-x
             1 root
                            root
                                           14 Apr 7 2006 librt.so -> librt-2.3.6.so
14 Apr 7 2006 librt.so.1 -> librt-2.3.6.so
18884 Apr 7 2006 librt.so.1 -> librt-2.3.6.so
lrwxrwxrwx
               1 root
                            root
lrwxrwxrwx
                1 root
                             root
-rwxr-x-
lrwxrwxrwx 1 root
-rwx 1 root
-rwxr-xr-x 1 root
                            root
               1 root
                                              19 Apr 7
19 Apr 7
                                                           2006 libthread_db.so -> libthread_db-1.0.so
                            root
                                                            2006 libthread_db.so.1 -> libthread_db-
                             root
1.0.so
                                           7836 Apr 7
16 Apr 7
-rwxr-xr-x
                1 root
                            root
                                                           2006 libutil-2.3.6.so
lrwxrwxrwx
                1 root
                             root
                                                            2006 libutil.so -> libutil-2.3.6.so
lrwxrwxrwx 1 root
                                              16 Apr 7 2006 libutil.so.1 -> libutil-2.3.6.so
                             root
```



10.4.2. Umgebungsvariablen

10.4.2.1. Verwendung von Umgebungsvariablen

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. Wichtige Umgebungsvariablen

PATH - der aktuelle Suchpfad bei der Eingabe von Kommandos

10.4.2.3. Systemspezifische Dateien

Das auf dem CentiPad eingesetzte Linux-System besteht aus über 20000 Dateien die mehr als 5 Millionen Zeilen Code enthalten. Die besondere Flexibilität von Linux ermöglicht jedoch die Anpassung dieses Betriebssystems an neue Hardware durch Modifikation eines verhältnismäßig kleinen Anteils der Quellen. Wie unter der GNU-Lizenz üblich, wird CentiPad mit dem kompletten Quelltext geliefert, der Anwender kann dieses System nach eigenem Wunsch erweitern und im Idealfall seine Erweiterungen an die Gemeinschaft zurückgeben. Das mitgelieferte Linux ist eine Arbeitsgrundlage, welche der Benutzer gemäß seinen Anforderungen anpassen kann.

Im Folgenden wird eine Auswahlliste für CentiPad modifizierter oder erstellter Dateien aufgeführt:

Pfad/Dateiname Autor/Lizenz Beschreibung

centidev/platform/linux/arch/arm/mach-at91rm9200/

board-centipad.c tk/maintech/GPL Board Konfigurationsdatei, hier werden

Boardresourcen für das Linux-System allokiert (serielle Schnittstellen, USB, LAN, MMC, SPI,

etc.)

centidev/platform/linux/drivers/char/

at91_gpio.c mk/maintech/GPL GPIO Treiber

at91_gpio.h mk/maintech/GPL GPIO Treiber Include

at91 gpiodev.c mk/maintech/GPL GPIO Treiber Usermode interface

centidev /platform/linux/include/asm-arm/arch-at91rm9200/

in diesem Verzeichnis finden sich die Maschinenspezifischen include-Files für den AT91RM9200



10.4.3. Die Laufzeitumgebung BusyBox

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. Bekannte 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

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 ein Nummerierung SnnXXXX, welche die Reihenfolge der Abarbeitung festlegt.

Das etc-verzeichnis enthält weitere Steuerdateien und Shellskripten 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

automatisch zu starten.

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 Resourcen zugeordnet werden sollen. Unter Umständen wird der Anwender hier spezielle Einstellen 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. 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/sdal /mnt/stick

10.5.2.2. Mounten des Datenbereichs des CentiPad Dataflashs

Das Dataflash ist in verschiedene Sektionen unterteilt.

/dev/mtdblock0 Bootloadercode /dev/mtdblock1 Konfigurationsdaten

/dev/mtdblock2 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

10.5.2.3. 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 Mounten erfolgen:



wenn noch nicht erzeugt

```
# mknod /dev/mmcblk0 b 8 0
# mknod /dev/mmcblk0p1 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 festegestellt, dass nicht alle SD-Cards erkannt werden. Bitte auf jeden Fall Testen!

10.5.2.4. Mounten 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 festegestellt, dass nicht alle CF-Cards erkannt werden. Bitte auf jeden Fall vor dem produktiven Einsatz auf Kompatibilität testen!

10.5.2.5. Mounten eines unter Windows freigegebenen Verzeichnisses

Sobald ein Verzeichnis das unter Windows als "LinuxExchange" freigeben 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:

\$ smbumount /mnt/smb



10.5.2.6. 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. Erzeugen einer Image-Datei

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 CentiPad-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 coLoinux-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: Partitiorien und Formatieren einer SDcard

Partitionieren: cfdisk /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

cfdisk /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. Netzwerk

10.6.1. Generelles

Bei vielen der beschriebenen Versuche ist es notwendig die genauen IP-Adressen der beteiligten Systeme zu kennen. So hilfreich DHCP ist, so sinnvoll kann es sein, jeder an der Entwicklung beteiligten Maschine eine feste IP-Adresse zu geben. Diese wird unter Linux wie unten beschrieben eingegeben (auf einem Linux-Hostrechner wahlweise auch über eine graphische Oberfläche wie YAST). Unter Windows kann man über Netzwerkverbindungen / TCP/IP-Verbindungen die IP-Adresse einstellen.

10.6.2. MAC-Adresse

Jede Netzwerkkarte benötigt eine eindeutige MAC-Adresse. Diese ist im System gespeichert, wird von der Initialisierungssoftware ausgelesen und wird bei der Konfiguration an die Netzwerk-Hardware übergeben.

```
## starte Ethernet-Schnittstelle mit MAC 00:FF:AB:CD:EF:00
ifconfig eth0 hw ether 00:FF:AB:CD:EF:00
```

Derzeit ist für das CentiPad kein eigener Bereich reserviert. Daher wird eine Adresse aus dem internationalen Testbereich eingesetzt. Befinden sich mehrere CentiPads im selben Netzwerksegment, so muss jedes über eine eigene MAC-Adresse verfügen.

10.6.3. Netzwerk konfigurieren und starten

CentiPad hat eine eingebaute Netzwerkkarte "eth0". Um diese zu starten wird eine IP-Adresse und eine Netzmaske benötigt.

```
## Netzwerk starten - up aktiviert Treiber
## IP Adresse: 192.168.1.100
## Netzmaske: 255.255.255.0 (Klasse C)
ifconfig eth0 192.168.1.100 netmask 255.255.255.0 up
## Standard Router einrichten
route add default gw 192.168.1.1
```

Manchmal ist ein Neustart des DHCP-Client notwendig: rcnetwork restart

10.6.4. Standardbenutzer / Passwort

Wichtig: der Standardbenutzer auf dem CentiPad ist root, das Passwort ist admin.

10.6.5. Adresse suchen / festlegen

nslookup hg.maintech.de sucht die IP-Adresse der angegebenen Adresse. In der Datei /etc/hosts kann man als root Adressen vorgeben, z.B. 87.106.18.11 hg.maintech.de.

10.6.6. weitere Benutzer auf dem CentiPad anlegen

Möchte man zusätzlich zum Standardbenutzer weitere Benutzer anlegen, so kann man nach folgendem Schema vorgehen:

- Angelegt werden soll ein Benutzer **myuser** mit dem Passwort **1234**.
- Zuerst in der Datei /etc/groups die Zeile "users:x:100:" einfügen und damit eine Gruppe users mit der User-Id 100 anlegen.



- Dann in der Datei /etc/passwd die Zeile
 myuser::200:100::/var/home/myuser:/bin/ash hinzufügen (200 ist die Benutzer-id und muß eindeutig sein).
- Danach /var/home anlegen und darin das Homeverzeichnis für den User entsprechend dem Eintrag in /etc/passwd: mkdir /var/home/myuser.
- Damit dem User sein Home-Verzeichnis auch selber gehört, chown -R myuser:users /var/home/myuser.
- Und schließlich sollte man dem User auch noch ein Passwort geben: 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 -1 root 192.168.1.100 stellt die Verbindung zu einem CentiPad mit der IP-Adresse 192.168.1.100 her.

ssh -Y -1 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 $\mathfrak{s}\mathfrak{s}\mathfrak{h}$ 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 $\mathfrak{s}\mathfrak{s}\mathfrak{h}$ 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 bietet eine einfache Möglichkeit um Daten von einem System zum anderen zu kopieren scp -r <quelle> <ziel>.

scp -r /home/centidev mah@192.168.1.33:/tmp kopiert das Verzeichnis
/home/centidev und alle Unterverzeichnisse auf den Rechner mit der IP 192.168.1.33,
unter Verwendung des dortigen Users "mah" und in das Verzeichnis \tmp.

Kopieren einer einzelnen ausführbaren Datei:

scp ./dateiname root@192.168.1.35:/tmp SCP fragt hierbei das Benutzerpasswort ab.

SCP ermöglicht einen schnellen Entwicklungsszyklus, da neue Dateien einzeln auf das CentiPad geladen werden können, ohne neu booten zu müssen.



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

Wenn Sicherheit nicht so wichtig ist, kann man auch eine Shell über telnet 192.168.1.108

aufmachen. Der Betrieb erfolgt dann ähnlich ssh.

10.6.11. Samba Verzeichnisse

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 - Daten von einem HTTP/FTP-Server holen

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 Netzwerteilnehmern.

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 -1 -p 9999

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

Empfänger: nc -1 -p 9999 > backup.tgz

Sender: tar cvzf - * | nc 192.168.1.150 9999

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



11. Die CentiPad Shell ASH

CentiPad verwendet als Shell die ASH – eine für embedded Systeme optimierte, da kompakte Version von BASH.

11.1.Code Tabelle

Zur korrekten Darstellung von Umlauten sollte die Codetabelle "utf-8 auf latin1" bzw "iso8859-1" ausgewählt sein. Diese Einstellungen gelten sowohl für den Quelltext-Editor, das Terminalprogramm und alle anderen Programme, mit denen Textdaten ausgetauscht und angezeigt werden.

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 Dateiendung 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 `. cprogrammname' starten. Dies ist die Kurzform von `source cprogrammname'.

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>-Zeilentrennunfg unter Windows. Beheben kann man dieses Problem, indem man unter Linux folgendes Kommado ausführt: recode recode latin1/cr-lf..latin1 <windowsdatei>



12. CentiPad Linux Hilfsprogramme

12.1.picocom

Picocom ist ein kompaktes Terminalprogramm für Embedded Systeme. Dokumentation ist unter centidev/platform/picocom/picocom-1.4/picocom.8.html verfügbar.

Picocom hat folgende Kommandos:

Man muss zuerst die Escape-Kombination drücken (z.B. Ctrl-a) und anschließend mit gedrückter Ctrl-Taste das eigentliche Kommando auslösen.

```
#define KEY_EXIT '\x18' /* C-x: exit picocom */
#define KEY_QUIT '\x11' /* C-q: exit picocom without reseting 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: decrase baudrate (down) */
#define KEY_FLOW '\x06' /* C-f: change flowcntrl 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 mesq 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 -1 -p 9999 > backup.tgz

13.9. Sourcenverwaltung

Bei der Firma maintech liegt CentiDev im Sourcenverwaltungssystem Mercurial. Dieses kann unter Linux wie folgt installiert werden:

http://www.selenic.com/mercurial/wiki/index.cgi/UnixInstall
http://www.selenic.com/mercurial/release/mercurial-0.8.1.tar.gz

Mercurial installation als root:

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

Ist Mercurial installiert, so kann mit centidev/update.sh die neueste CentiDev-Version über die vorhandene Version installiert werden.

Bitte vor jeder Änderung am CentiDev ein Backup machen! Wenn während des Updates etwas schief läuft (Systemabsturz, Unterbrechung der Kommunikation, inkompatible Versionen), kann dies zu einem nicht funktionsfähigen CentiDev führen!

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

Alternativ kann man mit weniger Internet-Bandbreite zunächst centidev/make clean durchführen und anschließend in den Verzeichnissen

centidev

centidev/platform/centiboot/centiboot
centidev/platform/kernel/linux

jeweils ein "hg pull && hg update" durchführen.

Anschließend centidev/make world.



13.10. direkter Zugriff auf das maintech Source-Code-Repository

Falls Sie direkten Zugriff auf unser Source-Code-Repository haben wollen, müssen Sie noch folgende Schritte an Ihrer SuSE unternehmen:

- das Paket "python-devel" installieren
- http://www.selenic.com/mercurial/release/mercurial-0.9.1.tar.gz herunterladen und nach den Anweisungen auf

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

- Danach können Sie den Source-Baum direkt auschecken:
- \$ hg clone http://hg.maintech.de/centidev
- warten diesmal ist aber auch eine flotte Internetanbindung noch zusätzlich hilfreich...

Dieser Schritt ist nicht zwingend notwendig - nur wenn Sie wirklich Bleeding Edge Software haben möchten, was im Zweifelsfall bedeutet, daß sie zumindest einfacher und schneller an Bug-Fixes herankommen.

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.



EWS

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 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.

www.t-online.de 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!

Gateways bzw. Router verbinden verschiedene Netze miteinander wie z.B. das

LAN mit dem Internet.

GMT Greenwich - Mean - Time

HTML Hypertext - Markup – Language. Darstellungsprache, 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 Darstellungsprache



16. Offene oder fehlende Punkte

Geplante Entwicklungen

16.1.Freigeben und Mounten von Netzwerk-Verzeichnissen

Rückfrage cd

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

Freigabe auf NFS-Server:

Willemer:

Datei:/etc/exports

Eintrag des freizugebenden Verzeichnisses.

/tmr

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