

Erfahrungen mit dem CubieTruck (Cubieboard 3)

Dieter Drewanz, September 2016

Inhaltsverzeichnis

1	Einleitung.....	3
2	Die Erstinbetriebnahme.....	4
2.1	Der Zusammenbau.....	4
2.2	Der erste Start.....	5
2.2.1	Start des Androids auf internen Flash-Speicher (NAND).....	5
2.2.2	Vorbereitungen zum Start eines Linux von der SD-Karte.....	5
2.3	Der erste Start des Linux für ARM-Prozessoren.....	8
2.4	Der zweite Start des Linux für ARM-Prozessoren.....	8
3	Erweiterung um SW-Pakete für die Nutzung.....	10
3.1	Vorbereitungen vor der Paketauswahl und Installation einer grafischen Oberfläche.....	10
3.2	Installation von ersten Anwendungen für das Internet und Dokumentenbearbeitung (Office)	11
3.3	Booten von mehreren Systemen bei Bedarf.....	13
4	Kompletieren des Systems und Vornehmen der Einstellungen.....	13
4.1	Vornehmen der Einstellungen und Konfigurationen.....	13
4.1.1	Einstellung des Tastaturlayouts.....	14
4.1.2	Konfiguration des WLAN.....	14
4.1.3	Einbinden von Swap-Partitionen.....	14
4.2	Installation von praktischen Werkzeugen und Anwendungen.....	16
4.2.1	Anwendungen für die Textbearbeitung und Büro-Suiten.....	16
4.2.2	Anwendungen zur Internetnutzung (Browser, Chat).....	17
4.2.3	Werkzeuge, Anwendungen zur Konfiguration und Editoren.....	17
4.2.4	Grafikprogramme, Bildbetrachter und Mediaplayer.....	18
4.2.5	Spiele.....	19
4.3	Zum ersten Start.....	19
4.4	Zu den ersten Schritten mit Armbian.....	20
4.5	Mediacenter ohne Linux/Armbian-Kenntnisse aufsetzen.....	21
4.6	Remote Access (Fernzugriff) auf den CubieTruck.....	22

4.7 Booten von mehreren Systemen bei Bedarf.....	24
5 Installation eines Druckers.....	24
5.1 Brother MFC 795.....	24
5.2 Vorbereitungen für i386er Software auf ARM-Prozessoren.....	25
5.3 Problemlösung zu libc6 bei der Druckerinstallation.....	27
5.4 Installation des Scanners des Multi-Funktions-Druckers.....	29
5.4.1 Installationsversuch des Scanner-Treibers i386 unter der arm-Architektur.....	30
5.4.2 Emulation mit Qemu in einer kleinen virtuellen Debian-Umgebung i386-Architektur.	38
5.4.3 Aus den Quellen compielieren einer aktuelleren Qemu-Version.....	41
5.4.4 Scanner-Treiber unter chroot mit qemu-i386-static installieren.....	46
6 Anwendungen, Installation, Einrichtung und kleine Skript-Werkzeuge.....	56
6.1 Terminal und Text-Editoren.....	56
6.2 Dateimanager.....	56
6.3 Medienplayer „mplayer“.....	57
6.3.1 Installation und Konfiguration.....	57
6.3.2 Kleine Script-Werkzeuge.....	57
6.4 Media Center.....	60
6.5 Datenverkehrsmonitoring.....	61
6.6 Netzwerkanwendungen.....	62
6.7 Anwendungen zur Netzwerkuntersuchung.....	63
6.8 Dokumentenbearbeitung – Office-Suiten.....	64
6.9 Bildbearbeitung.....	65
6.10 Mathematische Anwendungen.....	65
6.11 Virens Scanner.....	65
6.12 Laufzeitumgebungen und Virtualisierung.....	65
7 Varianten zum Vorgehen bei der Installation.....	66
7.1 Verwenden von bereits heruntergeladenen Paketdateien.....	66
7.2 Das System auf eine größere SD-Karte übertragen.....	67
7.3 Updaten des Betriebssystems.....	68
7.3.1 Ungewolltes Update und Änderung des Zustandes.....	68
7.3.2 Trouble-Shooting Zeitangabe.....	70
7.3.3 Trouble-Shooting WLAN geht nicht mehr.....	70
7.3.4 Trouble apt-get update.....	72
7.4 Trouble LibreOffice Erweiterungen und Java-Installation.....	72

8 Aspekte der IT-Sicherheit.....	76
8.1 Zugang beim Starten des Systems.....	76
8.2 Einloggen über das Netzwerk.....	76
8.3 Erste einfache Maßnahmen zur Absicherung.....	77
8.4 Und noch eine kleine Besonderheit des CubieTruck.....	77
9 Zusammenfassung.....	78
10 Ein anderes Linux auf dem CubieTruck.....	78
1 Anlagen – Dokumentteile des Raspian.....	80
1.1 Ein OSMC-Image für Berryboot erzeugen aus einem SD-Image für den Raspberry.....	80
1.2 Installation von Anwendungen und kleinen Werkzeugen.....	84
2 Ab hier Reste von Raspberry Pi.....	87
3 Lösung von einzelnen Problemen und interessante Konfigurationen.....	87
3.1 Bildschirmprobleme.....	87
4 Besondere Netzwerke.....	88

1 Einleitung

Der hier geschriebene Text wurde parallel als kleine Dokumentation angelegt. Eine Nebenabsicht war auch um bei Bedarf schnell ein ähnliches System wieder aufzusetzen und nachzusehen, was gemacht wurde für spätere Änderungen oder Fortsetzungen.

Somit ist dieser sprachlich einfach gehalten und auch nicht weiter quergelesen. Ohne Absichten und Aussichten auf irgendwelche Zuwendungen wurde verständlicherweise hier auf den schriftlichen Ausdruck auch nicht besonders geachtet. Manchmal wurden auch nur die Notizen mit den Links zu Seiten im Internet mit interessanten Informationen, Beschreibungen und Anleitungen gesammelt.

Eingestellt im Netz kann es vielleicht dem Einen oder Anderen nützlich sein zur Information oder Fehlerbehebung.

Über den Jahreswechsel wurde ein Raspberry 2B eingerichtet. Für der Nutzung des Gerätes fand ich die 1GB RAM etwas knapp bemessen, wenn es zum Surfen mit mehreren offenen Tabs (=Reiter auf Deutsch) und parallel noch zur Dokumentenbearbeitung verwendet werden soll. Auf der SD-Karte wurde zwar eine Swap-Partiton eingerichtet, aber Flash hat eine endliche Zahl von Schreibzyklen, welche die Lebensdauer begrenzt. Also hoffte ich auf ein Nachfolgermodell Raspi 3 mit 2GB RAM.

Nachdem dies nicht so war, suchte ich nach Alternativen. Da nicht viel Rechenleistung benötigt wurde für die vorgesehene Verwendung fiel die Wahl letztendlich auf einen CubieTruck. Ein Hauptentscheidungsgrund war der zusätzliche VGA-Anschluss, so dass mit wenig Aufwand ältere Monitore mit dem Gerät genutzt werden können. Beim Raspi verwendete ich eine Adapterkette HDMI-DVI-VGA, jedoch hatte ich beim Anschließen immer Platzprobleme mit dem Adapterwulst auf der Rückseite des Monitors. Auf Grund der zwei Anschlüsse, einmal VGA und einmal HDMI

sollte bei Bedarf auch ein Dual-Screen-Betrieb reibungslos funktionieren. Als weitere Pluspunkte seien noch der SATA-Anschluss für eine Festplatte und der Anschluss für einen Lithium-Akku aufgeführt.

Ein gravierender Nachteil ist jedoch, dass nur zwei USB-Anschlüsse vorhanden sind. Mit Maus und Tastatur sind diese bereits belegt. Glücklicherweise lag noch ein USB-Hub herum, der hier eine dauerhafte Verwendung fand. Im Prinzip gibt es zwei unterschiedlich Verhalten von Hubs. Es gibt Hubs die funktionieren auch ohne, dass eine eigene externe Spannungsversorgung angeschlossen werden muss. An diese sollten aber nur kleine Verbraucher wie eben Tastatur, Maus oder USB-Sticks angeschlossen werden. Es kann sonst der einzelne USB-Anschluss des Raspi überlastet werden. Andere Hubs funktionieren nur mit einer externen Spannungsversorgung und wirklich nur dann sind die USB-Geräte mit Strom versorgt sowie für den Rechner erreichbar. Bei vielen dieser Hubs, wird die Stromversorgung an die USB-Geräte unterbrochen, wenn der Masteranschluss zum Rechner auch stromlos wurde.

Der CubieTruck, Raspberry und ähnliche Computerboards wurden und sind auch weiterhin konzipiert für Technikinteressierte als günstige Systeme zum Experimentieren und Lernen. Daher wird auch kein großer Aufwand betrieben um alle Fallstricke aus dem Weg zu räumen von Seiten der jeweiligen Community betreffend der Software und Anwendungen.

2 Die Erstinbetriebnahme

2.1 Der Zusammenbau

Als komplettes Set von einem bekannten Laden (Reichelt, Conrad, ELV, Völkner, Pollin, Watterott, usw.) ist das Gerät schnell zusammengebaut. Es liegt aber keine Anleitung bei. Im Internet ist diese sicher zu finden, aber ich brauchte diese nicht. Die Platte mit den vielen Ausschnitten ist ganz oben und die die Platte mit einem kleinen Loch in der Nähe einer Ecke ist so zu platzieren, dass dieses Loch in der Nähe der Sata-Anschlüsse liegt. Die Schutzpapiere müssen vor der Montage von der Plexiglasplatten abgezogen werden. Dabei sollten nicht zu harte Gegenstände als Hilfsmittel verwendet werden.

Im Nachhinein wäre das Kompletgerät für ca. 175 Euro mit Akku plus 64GB SATA zu empfehlen, statt ca. 100 Euro für das Gerät im Acrylglasgehäuse, wenn dieser auch öfters transportiert werden soll. Zusätzlich wäre das Gerät Akku-gepuffert bei eventuellen kürzeren Stromausfällen.

Den mitgelieferten Kühlkörper habe ich erst einmal weggelassen und zusammen mit den Kabeln in der Schachtel aufgehoben.

2.2 Der erste Start

2.2.1 Start des Androids auf internen Flash-Speicher (NAND)

Nach dem Zusammenbau schloß ich ein kleines Netzteil an mit nur 1A und 5V. Der CubieTruck startete ohne Probleme in das Android. Bei den Einstellungen wurde das WLAN konfiguriert. Solange kein Google-Account eingerichtet wurde, kann der Google-Play-Store nicht genutzt werden und auch der Browser läßt kein Surfen im Internet zu.

Es handelt sich dabei auf den 8GB internen Flash-Speicher um ein 4.4er Android-Version. Ermittelt wurde, dass vom Hauptspeicher 2GB angezeigt wird, dass noch 0.88GB frei wären und von 8GB Flash noch 4,76GB frei wären. Über die Nacht (als einfacher erster Dauertest um schon mal eine Macke auszuschließen) lief der CubieTruck durch und es wurden in dem Zeitraum insgesamt 47MB Datenverkehr verursacht.

Unter den Einstellungen ist auch einstellbar, ob der CubieTruck vom internen Flash-Speicher oder von einer SATA-Festplatte startet. Wenn eine bootfähige SD-Karte im Gerät steckt, wird immer von dieser gebootet.

An der Seite neben dem IT-Sensor befindet sich ein kleiner Schalter. Dieser ist leicht mit dem Sensor zu verwechseln, da dieser meist für den Taste gehalten wird. Etwa eine Sekunde drücken löst das Herunterfahren des Android aus, wie auch das Einschalten. Langes drücken ab ca. 4 Sekunden bewirkt ein hartes Ausschalten. Gegenüber einigen anderen Einplatinenrechner ist das Vorhandensein eines Tasters zum Ein- und Ausschalten positiv zu bewerten.

Noch zu testen ist, ob bei Wiederkehr der Spannung das Gerät auch wieder von selbst bootet. Im Hinblick auf die Verwendung eines kleinen dauernd-online Servers wäre solches Verhalten praktisch.

Nach dem zweiten Boot-Vorgang stimmte auch die Zeitangabe (statt Januar 2000 nun das richtige Datum), wobei die Zeitzone noch nicht passte. In der Nähe des Stromanschlusses ist auf der Platine ein Bauteil zu erkennen, das wie eine kleine Uhrbatterie mit angelöteten Anschlüssen aussieht. Es scheint sich hier tatsächlich um eine kleine Zelle für eine Echtzeituhr (RTC) zu handeln. Falls diese ausgetauscht werden müßte, wird dies Umständen verbunden sein, da die Zelle mit Lötfahnen verbaut wurde. Bei Gelegenheit werde ich die Spannung messen, da nicht ersichtlich ist, ob es sich um eine 1,5V Standard oder 3,7V Li-Zelle handelt. Es ist leider nicht bekannt, was passiert, wenn diese Zelle ausfallen sollte.

2.2.2 Vorbereitungen zum Start eines Linux von der SD-Karte

Im Gegensatz zum Raspi gibt es hier weniger Informationen zu den Betriebssystemen und auch weniger Images zum Herunterladen für den CubieTruck. Da diese aber einen ARM-Prozessor und eine Mali-GPU wie viele andere Modelle verwendet, gibt es doch eine kleine Auswahlmöglichkeit.

Vor dem Start sollte man sich die Infos unter folgendem Link ansehen:

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

Verwendet wurde ein Debian-basiertes Linux unter:

<http://www.armbian.com/download/>

Auf diesem Link befindet sich eine sehr gute Übersicht über viele verschiedene solcher ARM-Boards von verschiedenen Herstellern.

<http://www.armbian.com/cubietruck/>

Auswahl: Unter der Listenspalte Vanilla wurde Jessie gewählt.

http://mirror.igorpecovnik.com/Armbian_5.04_Cubietruck_Debian_jessie_4.4.3.zip

Die Zip-Datei von ca. 0,25 GB wurde heruntergeladen und entpackt. In dem Paket befindet sich eine kleine exe-Datei für das Schreiben des Images auf eine SD-Karte unter Windows. Die andere große Datei von ca. 1,0 bis 1,3 GB ist ein Image als raw-Datei. Unter Linux schreibt man diese Datei mit dd auf eine **leere** SD-Karte, da bei dem Vorgang alle Daten verloren gehen.

```
'dd bs=1MB if=Armbian_5.04_Cubietruck_Debian_jessie_4.4.3.zip of=/dev/sdx'
```

Achtung bei Nutzung von dd und der Angabe des zu beschreibenden Mediums „of=/dev/sdx“! Wenn das falsche Medium erwischt wird, ist das sehr ärgerlich da alle Daten unwiederbringlich überschrieben wurden. Ich hatte parallel gparted laufen und sicherheitshalber einen weiteren angeschlossenen USB-Stick '/dev/sdb' gezogen. Ein ausgelöster Reload der Partitionen durch eine Betriebssystemroutine verursachte eine Neuordnung genau während jener höchstens drei Sekunden bis dd gestartet wurde, so dass die Lücke zwischen /dev/sda bis /dev/sdd geschlossen wurde und „Schwups“ war das falsche Medium überschrieben. Der USB-Speicherstick enthielt glücklicherweise nur die zip-Datei, die sofort wieder heruntergeladen werden konnte. Somit war kein relevanter Datenverlust entstanden.

Nachdem die SD-Karte beschrieben war, befand sich auf dieser eine primäre Partition von ca. 1,1GB, der restliche Speicherplatz auf der 32GB SD-Karte war noch unbelegt.

Diese SD-Karte könnte nun ohne weitere Bearbeitung in den CubieTruck eingesteckt werden und das System würde booten. Bei dem Bootvorgang würden auch noch Partitionen angelegt oder vergrößert werden. Ich habe aber auf der Karte vorher mit gparted selbst noch Partitionen angelegt und die Größen verändert.

```
Anbei die Ausgabe der Partitionen mit „fdisk -l“:
Disk /dev/mmcblk0: 29 GiB, 31104958464 bytes, 60751872 sectors
Device      Boot      Start          End  Sectors   Size Id Type
/dev/mmcblk0p1                2048 16525311 16523264    7.9G 83 Linux
/dev/mmcblk0p2      16525312 60751871 44226560 21.1G  5 Extended
/dev/mmcblk0p5      35280896 56424447 21143552 10.1G 83 Linux
/dev/mmcblk0p6      56426496 60751871 4325376  2.1G 82 Linux swap / Solaris
```

Wer genau hinsah, fiel vielleicht auf, dass hier eine Lücke am Anfang der erweiterten Partition von ebenfalls fast 8GB vorhanden ist. Das war Absicht um bei Bedarf mittels „gparted“ später die erste Partition zu vergrößern (oder den Weg über die Konsole zu testen). Jedoch zeitnah wurde die Hälfte des freien Bereiches hinzugenommen, da mit den Anwendungen und vor allem Aufheben der Dateien in /var/cache/apt die erste Partition sich doch schneller füllte als angenommen.

```
Device      Boot      Start          End  Sectors   Size Id Type
/dev/mmcblk0p1                2048 26079231 26077184 12.4G 83 Linux
/dev/mmcblk0p2      26079232 60751871 34672640 16.5G  5 Extended
/dev/mmcblk0p5      35280896 56424447 21143552 10.1G 83 Linux
/dev/mmcblk0p6      56426496 60751871 4325376  2.1G 82 Linux swap / Solaris
```

Im Hinblick auf Abstürze und korrupte SD-Karten (dirty bit gesetzt) scheint der CubieTruck beim Booten toleranter als der Raspberry Pi 2 zu sein. Die Notwendigkeit des CubieTruck hier toleranterweise trotzdem von der SD zu booten, würde sonst zu häufig bei einer Fehlbedienung zum Totalverlust auf der SD-Karte führen. Wenn Android starten würde, wäre eine der ersten Fragen während des Startvorgangs, die Frage ob die SD-Karte für Android vorbereitet werden sollte. Wenn diese aus Versehen bejaht werden sollte, wären aller Daten auf der SD-Karte verloren, da diese neu partitioniert, formatiert und ein kleiner Teil überschrieben werden würde.

Wenn das "dirty bit" gesetzt sein sollte, der Reparaturversuch beim Selbstcheck während des Hochfahrens scheiterte, befindet sich die SD-Karte nur im read-only Mode. Ein Versuch hierbei andere Medien zu mounten funktioniert in dem Zustand auch nicht, sei noch angemerkt. Passiert dies während des Betriebes, dass die SD-Karte korrumpiert, dann befindet sich die SD-Karte ebenfalls nur noch im read-only Mode. Ein Versuch andere Medien zu mounten funktioniert in dem Zustand wiederum auch nicht. Noch nicht gespeicherte Daten sind damit in der Regel nicht mehr zu retten.

Eine zweite SD-Karte sollte bereit gehalten werden als eine Sicherungskopie für den Fall der Fälle, dass alles schief gehen sollte. Dies gilt vor allem, wenn das System mit einer schwachen Datenverbindung (mobiles Internet, DSL deutlich unter 2MBit) wieder aufgesetzt werden müsste, viele Anwendungen nachgeladen wurden oder viele Konfigurationen am System durchgeführt wurden. Wenn alles ausreichend dokumentiert wurde, wie zum Beispiel in diesem Dokument, dann dauert das Wiederaufsetzen des Systems nur noch ein Viertel der Zeit oder sogar weniger. Auch sehr zu empfehlen ist dies, wenn das Gerät für Personen ohne Linux-Kenntnisse aufgesetzt wurde, die nicht in der Lage sind mit „fsck“ das Dateisystem wieder zu reparieren.

Getestet wurde was passiert, wenn am USB-Anschluss des CubieTruck ein SmartPhone angeschlossen würde und der CubieTruck nur über ein 1A Netzteil versorgt wäre. Bei diesem Experiment ging das Gerät sofort aus. Als zweiter Versuch wurde der CubieTruck mit nur einem 0,5A Netzteil gestartet. In dem Falle gab es nur ein kurzes Blinken der LED und das Gerät war nach wenigen Sekunden (kleiner 5 Sekunden) sofort aus.

Empfohlen wird ein Netzteil mit mindestens 2A für den Betrieb ohne SATA oder USB-Festplatten, beziehungsweise mindestens 3A, wenn diese vorhanden wären. Eine Obergrenze für die Stromversorgung ist zwar nicht angegeben oder aufgedruckt, aber nach der Dicke der Anschlüsse und Leiterbahnen sollte es 4A nicht übersteigen. Wird mehr benötigt, so muss die Versorgung über einen USB-Hub realisiert werden, der nur funktioniert, wenn dessen externe Versorgungsquelle angeschlossen wurde. Ansonsten kann das Gerät auf diese Art und Weise sehr schnell Elektronikschrott werden.

Nachtrag: Über längere Zeit wurde der CubieTruck mit dem 1A Netzteil betrieben. Ausgefallen auf Grund Strommangels war es nur einmal, als zwischen einer 32GB SD-Karte im USB-Adapter und einem 128GB USB-Stick kopiert wurde. Beide Speicherkomponenten waren günstige Sonderangebote und zeichnen sich nicht durch geringen Stromverbrauch aus. Solche Angaben fehlen in der Regel auf diese Produkten. Aus diesem Grunde gibt es bei den verschiedenen Elektronikhändlern und Elektronikläden USB-Strommessadapter um auch solche Messungen durchzuführen.

2.3 Der erste Start des Linux für ARM-Prozessoren

In dem Gerät steckte die vorbereitete SD-Karte mit dem Armbian Linux. Sobald der CubieTruck an die Stromversorgung angeschlossen wurde, startete er durch. Bei dem Erststart erschien kein Auswahlmenü. Die Stufen des Starts konnten gesehen werden auf dem angeschlossenen VGA-Monitor. Als letzter Schritte war dem Bildschirm zu entnehmen, dass auch versucht wird ein root-System anzulegen und eine minimale Swap von 128MB für den Notfall. Meine Partitionierung der SD-Karte wurde dabei nicht geändert. Als swap war eine Datei der Größe von etwa 100MB unter /var/swap angelegt worden (zu finden in der Ausgabe von "dmesg | less" oder "dmesg | grep swap" auf der Konsole).

Danach hing das System während des Schrittes die Updates zu holen. Nach ungefähr zehn Minuten wurde hier abgebrochen. Über langes Drücken des Tasters wurde der Rechner abgeschaltet.

Für den ersten Start empfiehlt sich es den CubieTruck über ein LAN-Kabel an den Router anzuschließen um zu ermöglichen, dass beim ersten Hochfahren die Updates geladen werden können. Wenn nur das WLAN zur Verfügung steht, gibt es hier ein nicht zu lösendes Henne-Ei-Problem sofern nicht vorher die /etc/network/interfaces entsprechend bearbeitet wurde.

2.4 Der zweite Start des Linux für ARM-Prozessoren

Die SD-Karte wurde nun bewußt nicht mit fsck bearbeitet, so dass das Dateisystem nicht repariert wurde oder das „dirty bit“ zurückgesetzt wurde. Im Hinblick auf durch Abstürze verursachter korrupter SD-Karten (dirty bit gesetzt) scheint der CubieTruck beim Booten toleranter als der Raspberry Pi 2 zu sein (Nachtrag: Der Raspberry Pi 3 scheint hierbei sich auch toleranter zu verhalten). Das wäre auch sinnvoll, denn wenn das Android hoch fährt und die Frage nach der SD-Karte für Android vorbereiten irrtümlich bejaht würde, wäre die SD-Karte überschrieben.

Auf jeden Fall startete der CubieTruck wieder brav bis zum Ende durch. Einige Zeit hing er an der Stelle betreffend der Netzwerkverbindungen (90s Wartezeit) beim Hochfahren. An dieser Stelle wartet das Armbian in der Regel immer einige Zeit auf eventuell vorhandene Netzwerke. Die Wartezeit könnte in den Boot-Optionen auch geändert werden.

Im Anschluss konnte gemäß einer Beschreibung zum Armbian als „root“ mit dem Paßwort „1234“ einloggt werden und es wurde sofort eine Vergabe eines neuen Paßwortes gefordert. Nur wenn dies erfolgte und das Passwort gewissen Mindestkriterien entsprach, wurde der nächsten Schritt erreicht. Es musste auch ein Nutzer mit Paßwort angelegt werden.

Anschließend befand man sich in der Konsole. Eine grafische Oberfläche war zu dem Zeitpunkt noch nicht vorhanden. Bei der Eingabe von „startx“ passierte also nichts. Als nächster Schritt wurde in der Konfigurationsdatei als „root“ oder mittels vorangestellten „sudo“ das WLAN konfiguriert, indem die Angaben des Netzwerknamens (ssid) und Passwortes eingetragen wurden.

```
$ sudo nano /etc/network/interfaces
# Edit your file for adding lines like this:
auto wlan0
    iface wlan0 inet dhcp
    wpa-ssid AndroidAccessPointName
    wpa-psk mypassword
```

Der Netzwerke-Service könnte gestoppt und neu gestartet werden, um die WLAN-Verbindung zum Laufen zu bringen. Hier wurde der einfachere Weg gewählt, das System herunterzufahren und neu

zu starten.

Beim Neustart war zu beobachten, dass etwas mehr als eine Minute Zeit verwendet wird, um die Netzwerke zu finden und sich zu verbinden. Ansonsten verlief der Bootvorgang schnell.

Da unter den Verzeichnissen der User (diese befinden sich unter /home/username/* noch keine Unterverzeichnisse gab, wurden (und sollten) darin Verzeichnisse wie Doku, Documents, Downloads, usw. angelegt werden.

Nun wäre ein guter Zeitpunkt, den Ist-Zustand zu speichern für spätere Vergleiche mit folgenden Befehlen auf der Konsole.

Partitionstabelle:

```
'sudo fdisk -l > fdisk_20160408.txt'
```

Belegung der Speichermedien:

```
'sudo df > df_20160408.txt'
```

Eingebundene Partitionen:

```
'sudo mount > mount_20160408.txt'
```

Installierte Pakete:

```
'sudo dpkg -l > dpkg_20160408.txt'
```

Meldungen beim Hochfahren:

```
'sudo dmesg > dmesg_$(date +%Y%m%d_%H%M)'
```

Später sinnvoll wäre auch:

Unterstützte Grafikmoden mit angeschlossenen Monitor:

```
'sudo xrandr > xrandr_20160408.txt'
```

Liste der Befehle für den späteren Nachvollzug:

```
'sudo history > history_20160408.txt'
```

Alles in einem Rutsch ohne die vorherigen Dateien versehentlich zu überschreiben, kann mit Hilfe der Datumsfunktion und gesetzten Optionen realisiert werden auf folgende Weise:

```
sudo fdisk -l > fdisk_$(date +%Y%m%d_%H%M).txt  
sudo df > df_$(date +%Y%m%d_%H%M).txt  
sudo mount > mount_$(date +%Y%m%d_%H%M).txt  
sudo dpkg -l > dpkg_$(date +%Y%m%d_%H%M).txt  
sudo dmesg > dmesg_$(date +%Y%m%d_%H%M).txt
```

```
'sudo xrandr > xrandr_$(date +%Y%m%d_%H%M).txt'  
'sudo history > history_$(date +%Y%m%d_%H%M).txt'
```

Solche Historien lege ich gerne in einem Verzeichnis INSTALL ab. Aber so etwas ist individuelle Geschmacksache, ob man es macht oder nicht.

3 Erweiterung um SW-Pakete für die Nutzung

Viele umfangreichere Anwendungen, sowohl GUI-basiert oder auf der Konsolenbasis waren bei

dem ursprünglichen Image des Armbian nicht bereits enthalten. Daher stand nun die Erweiterung um meiner Ansicht nach benötigte Pakete im Vordergrund.

3.1 Vorbereitungen vor der Paketauswahl und Installation einer grafischen Oberfläche

Bevor die Pakete ausgewählt werden, sollten die Paketlisten für „apt-get“ aktualisiert werden.

```
'sudo apt-get update'
```

Am Ende wurde angezeigt, dass rund 16MB heruntergeladen wurden.

Als erstes Paket wird ein Datenverkehrszählprogramm installiert.

```
'sudo apt-get install vnstat'
```

Es wurde angezeigt, dass rund 2MB heruntergeladen werden und 0,27MB belegt würden.

Beispiele zur Verwendung:

```
'vnstat' gibt eine Gesamtstatistik aus.  
'vnstat -l -i wlan0' zeigt die aktuelle Datenrate an.  
'vnstat -h -i wlan0' gibt den Datentransfer stundenweise als Liste aus.  
'vnstat -d -i wlan0' gibt den Datentransfer tageweise als Liste aus.  
'vnstat -m -i wlan0' gibt den Datentransfer monatlich als Liste aus.  
'vnstat -l -i wlan0' --style 4' zeigt die aktuelle Datenrate an und schreibt  
dies Zeile für Zeile fort.  
Eine Ergänzung um '>> Datei.txt' oder '| tee datei.txt' funktionierte nicht.
```

Als Zwischenfeststellung wurde mit „top“ ermittelt, dass etwa 270MB des RAM belegt wären beim Betrieb im Konsolenmodus.

Um auf dem Gerät zu arbeiten wird als nächstes eine einfache grafische Oberfläche installiert.

```
'sudo apt-get install xorg'
```

Es wurde angezeigt, dass rund 40MB heruntergeladen werden und etwa 80MB belegt würden.

```
'sudo apt-get install xfce4'
```

Es wurde angezeigt, dass rund 25MB heruntergeladen werden und etwa 92MB belegt würden.

Dabei kam es zu einem Abbruch während des Installationsvorganges. Es konnte auf das Speichermedium nicht mehr schreibend zugegriffen werden. Dies belegte die Ausgabe des Befehls „mount“, da in der Zeile des Mediums „mmcblk0p1 ... (ro,)“ statt „(rw,...)“ stand. Wegen Fehler wurde vom Betriebssystem ein "remount" als "read-only" durchgeführt. Die Fehlermeldungen mit „dmesg“ wurden nicht mehr betrachtet. Es konnte nichts mehr geschrieben werden und ein weiteres Medium konnte auch nicht eingebunden werden ("mount ..."), da auf der SD-Karte nicht mehr geschrieben werden konnte. Somit konnten die Daten der offenen Dateien auch nicht mehr gerettet werden.

Nach wenigen Minuten funktionierten auf dem Rechner immer weniger Anwendungen, bis die Maus auch zu hängen anfang, so dass ein harter Neustart notwendig wurde. Allerdings wurden die Zeiträume bis zum nächsten "remount" als "read-only" recht kurz, so dass mit der Konsolenanwendung "fsck" das Dateisystem repariert werden mußte. Die weiteren Aufrufe von "apt-get" gaben Befehlsvorschläge für die Fehlerbehebung, die das Problem beseitigten sollten, die durch den Abbruch während des Installationsvorganges entstanden.

```
'sudo dpkg --configure -a'
```

Obiges Kommando löste das Problem, so dass die Installationen fortgesetzt werden konnten.

Nun wurde das erste mal mit „startx“ die grafische Oberfläche "xfce" gestartet. Dabei poppte ein Fenster zur Auswahl der Paneleigenschaften auf. Es wurde „use default“ gewählt. Als Anmerkung sei noch ergänzt, dass die Einstellungen der Tastaturbelegung und Sprachen noch offen gelassen wurden. Interessant anzumerken wäre, dass Englisch mit tschechischer Tastaturbelegung die Anfangseinstellungen waren.

Mit „xrandr“ zeigte sich, dass der VGA-Monitor mit 1024x768 angesprochen wurde, dieser aber 1280x1024 hätte. Anscheinend fehlen bei VGA einigen Betriebsmoden. Als Zwischenfeststellung wurde mit „top“ ermittelt, dass etwa 297MB RAM belegt wären bei Verwendung von X11 mit XFCE (minimaler Modus mit GUI).

Ein Test mit 'sudo apt-get upgrade' ergab, dass rund 40MB heruntergeladen würden.

Ein Test mit 'sudo apt-get install kwrite' gab aus, dass rund 123MB heruntergeladen würden und 372 belegt würden. Das liegt darin begründet, dass einige Komponenten von KDE installiert würden.

3.2 Installation von ersten Anwendungen für das Internet und Dokumentenbearbeitung (Office)

Nun sollte ein Programm für die Notizen installiert werden und für die Dokumentenbearbeitung.

Als Befehl für die Installation von abiword im "xterminal" (Terminal/Konsolenfenster unter XFCE) wurde eingegeben:

```
'sudo apt-get install abiword'
```

Es wurde angezeigt, dass rund 40MB heruntergeladen und etwa 127MB belegt würden.

Dabei kam es zu einem Abbruch während des Installationsvorganges. Es konnte auf das Speichermedium nicht mehr schreibend zugegriffen werden. Dies belegte die Ausgabe des Befehls „mount“, da in der Zeile des Mediums „mmcblk0p1 ... (ro,)“ statt „(rw,...)“ stand. Wegen Fehler wurde vom Betriebssystem ein "remount" als "read-only" durchgeführt. Die Fehlermeldungen mit „dmesg“ wurden nicht mehr betrachtet. Es konnte nichts mehr geschrieben werden und ein weiteres Medium konnte auch nicht eingebunden werden ("mount ..."), da auf der SD-Karte nicht mehr geschrieben werden konnte. Somit konnten die Daten der offenen Dateien auch nicht mehr gerettet werden.

Nach wenigen Minuten funktionierten auf dem Rechner immer weniger Anwendungen, bis die Maus auch zu hängen anfang, so dass ein harter Neustart notwendig wurde. Allerdings wurden die Zeiträume bis zum nächsten "remount" als "read-only" recht kurz, so dass mit der Konsolenanwendung "fsck" das Dateisystem repariert werden mußte. Die weiteren Aufrufe von "apt-get" gaben Befehlsvorschläge für die Fehlerbehebung, die das Problem beseitigten sollten, die durch den Abbruch während des Installationsvorganges entstanden.

```
'sudo dpkg --configure -a'
```

Das reichte diesmal nicht aus, da der Abbruch während des „unpacking“ stattfand. Es war

zusätzlich notwendig mit der Option "force" einzelne Schritte erneut zu erzwingen:

```
'sudo apt-get install -f'
```

Es wurde ein Dateimanager auf der Konsolenebene und fürs Terminal installiert.

```
'sudo apt-get install mc'
```

Es wurde angezeigt, dass rund 1,7MB heruntergeladen und 6,7MB belegt würden.

Es wurde ein Browser fürs Internet installiert.

```
'sudo apt-get install iceweasel'
```

Es wurde angezeigt, dass rund 37MB heruntergeladen und 66MB belegt würden.

Es wurde eine Tabellenkalkulation installiert.

```
'sudo apt-get install gnumeric'
```

Es wurde angezeigt, dass rund 4.4MB heruntergeladen und 36MB belegt würden.

Es wurde eine Office-Paket installiert.

```
'sudo apt-get install libreoffice'
```

Es wurde angezeigt, dass rund 88MB heruntergeladen und 272MB belegt würden.

Es wurde ein Editor mit einfacher graphischer Oberfläche installiert:

```
'sudo apt-get install kwrite'
```

Es wurde angezeigt, dass rund 123MB heruntergeladen und 372MB belegt würden. Natürlich hatte mich dieser große Umfang etwas verwundert. Der Grund hierfür ist, dass Grundkomponenten von der Oberfläche KDE zusätzlich installiert werden um Anwendungen von Seiten der KDE-Teams unter Gnome oder anderen Oberflächen verwenden zu können. Interessanterweise wurde auch der VLC-Player mit installiert.

Angemerkt sei noch, dass hier die SD-Karte zweimal in den read-only mode zurückgefallen war und somit zwei Neustarts mit einer Wiederholung der bekannten Befehle zum „Trouble-Shooting“ notwendig wurden:

```
xxxxxxx@cubietruck: mount
dev/mmcblk0p1 on / type ext4 (rw,noatime,nodiratime,errors=remount-ro,commit=600)
xxxxxxx@cubietruck: mount
dev/mmcblk0p1 on / type ext4 (ro,noatime,nodiratime,errors=remount-ro,commit=600)
xxxxxxx@cubietruck: sudo dpkg --configure -a
Wurde nicht fertig, somit Neustart und nochmal:
xxxxxxx@cubietruck: sudo dpkg --configure -a'
xxxxxxx@cubietruck: sudo apt-get install kwrite
Zeigte, dass noch 0 von 30 MB herunterzuladen waren und weiter 40MB belegt wuerden.
xxxxxxx@cubietruck: sudo dpkg --configure -a'
Nicht notwendig war diesmal „sudo apt-get install -f“ anzuwenden.
```

Bei den Neustarts wurde das Verhalten nach dem Ausschalten getestet. Wird die Versorgungsspannung weggenommen, bootet das Gerät bei Wiederkehr der Versorgungsspannung. Wurde das Gerät mit dem Austaster abgestellt, dann musste bei der Wiederkehr der Taster betätigt werden. Zu prüfen wäre dies noch bei langen Ausphasen, da noch Kondensatoren geladen gewesen sein könnten und

die Minute überbrückt haben könnten. Bei längerer Ausschaltphasen wurde neu gestartet ergaben die Test und auch der Zufall eines Stromausfalls.

Es wurde ein einfaches und praktisches Werkzeug zum Einbinden von Datenträgern installiert.

```
'sudo apt-get install pmount'
```

Es wurde angezeigt, dass rund 82kB heruntergeladen und 730kB belegt würden.

Es wurde eine Zeichenprogramm (KDE) installiert.

```
'sudo apt-get install krita'
```

Es wurde angezeigt, dass rund 32MB heruntergeladen und 94MB belegt würden. Da mit der Installation von kwrite bereits die KDE-Komponenten installiert wurden, mussten also nicht mehr etwa 100MB hier heruntergeladen und 450MB belegt werden.

3.3 Booten von mehreren Systemen bei Bedarf

Eine weitere SD-Karte wäre hier mit „berryboot“ herzurichten. Auch möglich wäre es den Boot-Vorgang auf einen USB-Speicherstick umzuleiten. Auf die Dauer wäre diese Realisierung stabiler, da die USB-Sticks in der Regel Chipsätze mit leistungsfähigeren Funktionen zur Ausblendung von ermüdeten Speicherzellen besitzen.

Da SSD (Solid State Disk, Flashdisk) noch leistungsfähigere Funktionen zur Ausblendung von ermüdeten Speicherzellen besitzen, wäre generell die zuverlässigste Lösung das Gerät mit einer SSD am SATA-Anschluss zu betreiben.

4 Komplettieren des Systems und Vornehmen der Einstellungen

4.1 Vornehmen der Einstellungen und Konfigurationen

Damit mit der Konsole bzw. dem Terminal besser gearbeitet werden kann unter der graphischen Oberfläche wird das Werkzeug „gnome-terminal“ installiert.

```
'sudo apt-get install gnome-terminal'
```

Es wurde angezeigt, dass rund 5MB heruntergeladen werden und etwa 18MB belegt würden.

Im „Applications Menu“ war es noch nicht zu finden aber in einem Xterm-Fenster mit dem Befehl „gnome-terminal“ oder mit „Alt+F2“ plus Eingabe von ebenfalls „gnome-terminal“ wird dieses Konsolenfenster aufgerufen. Der Vorteil diese Konsole ist, dass die Eingaben und Ausgaben hier kopiert werden können für die weitere Nutzung oder Dokumentation.

4.1.1 Einstellung des Tastaturlayouts

Eingestellt wurde eine deutsche Tastatur mit 104 Tasten.

```
sudo dpkg-reconfigure keyboard-configuration
```

Die Einstellungen auf deutsches Layout mit dieser Konfiguration werden zwar übernommen, aber die update-rc.d start/stop Funktionen werden nicht mehr unterstützt erschien am Ende in der Konsole als Textausgabe. Somit waren diese Konfigurationen erst nach einem Neustart verfügbar.

4.1.2 Konfiguration des WLAN

Für das WLAN wurde die Datei /etc/network/interfaces editiert:

```
xxxxxx@cubietruck:~$ cat /etc/network/interfaces
# Wired adapter #1
auto eth0
    iface eth0 inet dhcp
#    hwaddress ether # if you want to set MAC manually
#    pre-up /sbin/ifconfig eth0 mtu 3838 # setting MTU for DHCP, static just: mtu 3838
#
# Wired adapter #2
#auto eth1
#    iface eth1 inet dhcp
#    hwaddress ether # if you want to set MAC manually
#    pre-up /sbin/ifconfig eth0 mtu 3838 # setting MTU for DHCP, static just: mtu 3838
#
# Wireless adapter #1
#auto wlan0
#    iface wlan0 inet dhcp
#    wpa-ssid SSID
#    wpa-psk xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
# to generate proper encrypted key: wpa_passphrase yourSSID yourpassword
#

auto wlan0
    iface wlan0 inet dhcp
    wpa-ssid NameNetzwerk
    wpa-psk PassWort

# Local loopback
auto lo
    iface lo inet loopback
```

Aktuell ist die Datei nicht geschützt und das Kennwort steht im Klartext in der Datei. Dies wird später noch geändert werden.

4.1.3 Einbinden von Swap-Partitionen

Bei einem SD-Karten basierten Betrieb des Minirechners bietet eine eigene Swap-Partition am Ende des SD-Kartenbereiches (letzte Partition) eine bessere Lösung, als eine Swap-Datei „/var/swap“. Wenn der obere Bereich der SD-Karte mit der Swap-Partition defekt würde, bliebe der Datenbereich mit Standardmitteln meist noch lesbar. Wenn es zu vielen Swap-Fehlern kommt, dann sollte die SD-Karte zügig ersetzt werden durch eine neue SD-Karte.

Leider fehlen von den Herstellern die Angaben, wie die Funktion des Ausblendens von fehlerhaften Flash-Zellbereichen realisiert wurde. In wenig performanten Produkten sollte von einer Umsetzung in Blöcken (z.B. 1GB) ausgegangen werden, denen jeweils eine kleine Zahl von Ersatzbereichen zugewiesen werden kann, die nicht blockübergreifend anderen Blöcken zugewiesen werden können.

```
sudo swapon
NAME      TYPE  SIZE USED PRIO
/var/swap file 128M  0B  -1
```

```
lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
mmcblk0   179:0    0   29G  0 disk
├─mmcblk0p1 179:1    0   7.9G  0 part /
├─mmcblk0p2 179:2    0    1K  0 part
├─mmcblk0p5 179:5    0  10.1G  0 part
└─mmcblk0p6 179:6    0   2.1G  0 part
```

```
sudo fdisk -l
Disk /dev/mmcblk0: 29 GiB, 31104958464 bytes, 60751872 sectors
Device      Boot      Start          End      Sectors   Size Id Type
/dev/mmcblk0p1                2048  16525311  16523264   7.9G  83 Linux
/dev/mmcblk0p2            16525312  60751871  44226560  21.1G   5 Extended
/dev/mmcblk0p5            35280896  56424447  21143552  10.1G  83 Linux
/dev/mmcblk0p6            56426496  60751871  4325376   2.1G  82 Linux swap / Solaris
```

Also /dev/mmcblk0p6 ist die Swap-Partition, die wir hier verwenden möchten. Mit swapon könnte diese jeweils temporär eingebunden werden, aber in der fstab eingetragen, wird diese immer eingebunden.

```
sudo swapon -p 100 /dev/mmcblk0p6
```

Hinweis: Eine hohe Priority setzen, damit diese bevorzugt verwendet wird.

Dauerhaftes ergänzen der Swap-Partition:

```
sudo nano /etc/fstab
# UNCONFIGURED FSTAB FOR BASE SYSTEM
tmpfs /tmp tmpfs nodev,nosuid,size=256M 0 0
/dev/mmcblk0p1 / ext4
defaults,noatime,nodiratime,data=writeback,commit=600,errors=remount-ro 0
0
/var/swap none swap sw,pri=1 0 0
/dev/mmcblk0p6 none swap sw,pri=5 0 0
```

```
sudo swapon
NAME      TYPE      SIZE USED PRIO
/var/swap file      128M  0B  -1
/dev/mmcblk0p6 partition 2.1G  0B   5
```

Im Terminal mit top oder htop kann nachgesehen werden, ob die Swap (in Summe) mit aufgenommen wurden.

```
KiB Swap: 2293752 total, 0 used, 2293752 free. 260196 cached Mem
```

```
sudo swapoff /var/swap
```

```
sudo swapon
```

```
NAME      TYPE      SIZE USED PRIO
/dev/mmcblk0p6 partition 2.1G  0B   5
KiB Swap: 2162684 total, 0 used, 2162684 free. 260216 cached Mem
```

Somit das Ergebnis des kleinen Tests, dass hier die Swap in Summe ausgegeben werden.

Wenn über die UUID die Swap-Partition eingebunden werden soll, dann kann diese mit diesem Befehl gefunden werden:

```
ls -l /dev/disk/by-uuid/
```

In der /etc/fstab wird z.B. eines der folgenden Beispiele eingetragen:

```
UUID=... none swap defaults,pri=10 0 0
```

```
UUID=... none swap sw,pri=10 0 0
```

Wenn eine alte HD als Swap über USB ergänzt werden soll, dann muss dieser die höchste Priorität vergeben werden.

4.2 Installation von praktischen Werkzeugen und Anwendungen

Nachdem das Betriebssystem armbian für den CubieTruck mit nur sehr wenigen Anwendungen ausgestattet wurde, mußten und müssen die benötigten Anwendungen nachinstalliert werden. Was man hier wählt hängt sehr von den eigenen Vorlieben ab und wie viel Speicherplatz auf der SD-Karte vorhanden ist. Erwähnt werden sollte hier noch, dass die Pakete im Verzeichnis /var/cache/apt/archives/ gespeichert werden. Dieses Verzeichnis wird immer weiter anwachsen, wenn man nicht zum Beispiel mit apt-get und Option autoremove die Cache-Verzeichnisse wieder löscht oder bereinigt. Zum Zeitpunkt der Installation von ersten Paketen (Office) und weniger Anwendungen lagen hier bereits etwa 600 Dateien mit insgesamt fast 400 MB.

```
ls /var/cache/apt/archives/ | wc
    593      593   22548
du /var/cache/apt/archives/
4      /var/cache/apt/archives/partial
384052 /var/cache/apt/archives/
```

4.2.1 Anwendungen für die Textbearbeitung und Büro-Suiten

Bereits installiert wurden gemäß vorderen Kapiteln:

- Abiword
- Gnumeric
- LibreOffice

Ergänzend wurden noch folgende Anwendungen und Ergänzungen installiert.

4.2.2 Anwendungen zur Internetnutzung (Browser, Chat)

Bereits installiert wurden gemäß vorderen Kapiteln:

- iceweasel

Ergänzend wurden noch folgende Anwendungen und Ergänzungen installiert.

w3m (download 1,1MB / 2,1MB disk space)

4.2.3 Werkzeuge, Anwendungen zur Konfiguration und Editoren

Bereits installiert wurden gemäß vorderen Kapiteln:

- mc (Dateimanager)
- kwrite
- pmount

Ergänzend wurden noch folgende Anwendungen und Ergänzungen installiert.

Liste des Wesentlichen für die grafischen Oberfläche:

- xorg
- xfce4

Liste der Werkzeuge:

- mc
- kwrite
- pmount
- vnstat
- gnome-terminal
- lxterminal
- etherape (download 8,5MB / 35MB disk space)
- testdisk (download 0,33MB / 0,87MB disk space)
- fdupes (download 0,02MB / 0,07MB disk space)
- xfce4-screenshooter (download 0,8MB / 3MB disk space)
- gparted
- openjdk (Java, JRE und SDK)

acpitools (download 0,05MB / 0,13MB disk space)

```
sudo acpitool
```

Could not open any of these files : /proc/acpi/info, /sys/module/acpi/parameters/acpica_version

Make sure your kernel has ACPI support enabled.

AcpiTool v0.5.1, released 13-Aug-2009

Funktioniert nicht, da kein BIOS vorhanden ist.

```
$ sudo apt-get install gparted
0 upgraded, 2 newly installed, 0 to remove and 7 not upgraded.
Need to get 2,076 kB of archives.
After this operation, 6,597 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

4.2.4 Grafikprogramme, Bildbetrachter und Mediaplayer

Bereits installiert wurden gemäß vorderen Kapiteln:

- krita

krita (krita stürzt beim Öffnen eines Bildes oder neuer Zeichnung ab);

Ergänzend wurden noch folgende Anwendungen und Ergänzungen installiert.

- vlc
- mplayer (download 1,6MB / 3,4MB disk space)
- inkscape (download 15MB / 86MB disk space)
- okular (download 7,4MB / 31MB disk space)
- xsane/sane, siehe Druckerinstallation

4.2.5 Spiele

Bereits installiert wurden gemäß vorderen Kapiteln:

- (Keine)

Ergänzend wurden noch folgende Anwendungen und Ergänzungen installiert.

- Extremetuxracer
- supertux

extremetuxracer (download 29MB / 35MB disk space)

Ruckartig der Tux und auch der Mauszeiger. Beide Prozessoren sind dabei voll aktiv.

supertux (download 8MB / 15MB disk space)

mesa-utils (download 0,03MB / 0,12MB disk space)

mesa-utils-extra (download 0,03MB / 0,12MB disk space)

Für glxgears und es2gears um die 2D/3D-Beschleunigung zu testen.

Anbei ein Beispiel zum Downloadbedarf für update und upgrade.

Eine Woche später (update lief als cron 1x täglich):

```
sudo apt-get update
Fetched 714 kB in 1min 41s (7,013 B/s)
```

```
sudo apt-get upgrade
61 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 45.0 MB of archives.
After this operation, 1,343 kB of additional disk space will be used.
```

Anmerkung: Vor dem update wären es 39MB zum Herunterladen gewesen.

4.3 Zum ersten Start

Es empfiehlt sich im Internet auffindbare Informationen vorher zu lesen. Läuft das Armbian, muss noch das Tastaturlayout auf Deutsch geändert werden. Etwas Grundwissen über Linux sollte unbedingt vorhanden sein, um hier zurecht zu kommen. Beim allerersten Start des geschriebenen Images das Speichermedium unbedingt den vorgegebenen Usernamen und Passwort bereithalten (armbian für cubitruck user:root pwd:1234). In die grafische Oberfläche kommt man mit „startx“ auf der Konsole.

In der Regel schafft es der xserver als Basis für die grafische Oberfläche sich selbst zu konfigurieren. Bei HDTV-Fernsehern kann es aber zu Problemen mit der richtigen Auflösung kommen. Ein Grund dafür kann sein, dass der Fernseher/Monitor über die HDMI-Schnittstelle seine unterstützten Moden nicht entsprechend meldet (bei Windows muss dies der Treiber beheben). Bei Raspian wird in der Regel davon ausgegangen, dass eher HDMI-Monitore angeschlossen sind, also erfolgen zuerst und vermehrt Abfragen für diese Gerätearten. Bei speziellen Distributionen für Medien-Center ist dies umgekehrt, so dass hier die TV-Geräteunterstützung vorgezogen wurde.

Für VGA muss beim Raspian vorher die Konfigurationsdatei „/boot/config.txt“ editiert werden, damit ein VGA-Monitor über (passive) HDMI-VGA-Adapter funktioniert. Beim CubieTruck funktioniert der VGA-Anschluss in der Regel auf Anhieb, ohne Änderungen in Konfigurationsdateien.

Klappt es nicht mit der Einstellung der optimalen Auflösung der Graphikkarte bzw. Graphikchips passend zum Monitor, kann mit Hilfe von xrandr oder anlegen einer xorg.config manuell noch ein problematischer Monitore oder TV bezwungen werden. Vorher sollte aber im Internet gesucht werden, ob es bereits für dieses Modell eine Lösung gibt oder über Konfigurationsdateien das Problem gemeistert werden könnte.

Mit „xrandr“ auf der Konsole können die aktuellen Werteeinstellungen abgefragt werden und welche unterstützten Modes gefunden wurden. Eventuell hilft auch ein Aufruf von „xrandr --auto“ weiter.

4.4 Zu den ersten Schritten mit Armbian

Zunächst sollten die Paketquellenverzeichnisse auf den aktuellen Stand gebracht:

- `sudo apt-get update`

Programme werden nachinstalliert mit

- `sudo apt-get install Paketname`

Zum Beispiel wird so der Webbrowser iceweasel (entspricht dem firefox) installiert:

- `sudo apt-get install iceweasel`

Wenn man mit einem Stichwort betreffende Paketnamen suchen möchte geht das mit:

- `sudo apt-cache search Stichwort`

Zum Beispiel für die Medientcenter suche man nach xbmc oder kodi.

Entweder wurde sofort in die grafische Oberfläche zu starten eingestellt oder mittels „startx“ wird diese aufgerufen. Die installierten Programme befinden sich dann in der Regel auch im Anwendungsmenü der XFCE-Oberfläche.

4.5 Mediacenter ohne Linux/Armbian-Kenntnisse aufsetzen

Anbei eine Vorgehensweise für Anwender mit wenig oder keinen Linux/Armbian-Kenntnissen um ein Mediacenter aufzusetzen.

Der Anwender ruft die Homepage von ähnlichen Systemen auf (z.B. Raspian) um sich über die Mediacenter zu informieren.

Es wählt sein Media-Center aus, hier OSMC, und landet dort:

<https://osmc.tv/download/>

Wählt dort sein Betriebssystem für die Software, zum Beispiel für Windows, lädt+installiert den Installer:

<https://osmc.tv/download/windows/>

<http://download.osmc.tv/installers/osmc-installer.exe>

Die SD-Karte steckt er in einen SD-Card-Reader, der am Rechner angeschlossen ist (eingebaut oder über USB) und beschreibt die SD-Karte mit dem Programm. Alle anderen externen Datenträger dabei unbedingt abstecken, damit nicht der falsche Datenträger gelöscht und überschrieben wird.

Diese SD-Karte steckt er nun in den Minirechner, einschalten, starten lassen und das war es schon. Wichtig zu wissen wäre noch, dass OSMC hat etwas mehr work around für einige Monitore oder Fernseher integriert hat (d.h. Scripte sind darauf abgestimmt).

Anbei sei noch erwähnt was der Unterschied zwischen Kodi und OSMC ist:

Quelle: <https://osmc.tv/help/wiki/kodi-osmc>

Kodi is a media center application and OSMC is the operating system that runs Kodi and brings it to your device. OSMC is not a fork of Kodi but rather a Linux distribution that ships Kodi as the main application. This is a similar concept to Kodi running on top of Windows or Android.

OSMC is based on Debian Jessie (a flavour of GNU/Linux) and has been heavily optimised to provide the best TV experience possible. OSMC is in charge of:

- Keeping your system up to date
- Hardware management (handling devices such as remotes, keyboards, WiFi adapters etc)
- Managing system resources

OSMC has over 30,000 packages made available via the Debian repositories as well as an App Store.

Problembhebung:

1. Bei OSMC kann es Probleme beim Installieren der Add-Ons geben. Wenn dies der Fall ist, hilft meist sich über einen anderen Rechner über das Netzwerk einzuloggen. Über die Terminal-Sitzung sollte als erstes „sudo apt-get update“ und dann „sudo apt-get upgrade“ durchgeführt werden.
Notwendig wird dies meist, wenn das neuesten Image verwendet wird zu einem Zeitpunkt an dem bereits bald ein noch neueres Image geben würde, wenn das Gerät lange nicht mehr eingeschaltet mit Netzverbindung verwendet oder neugestartet wurde.
2. Es kann zu Fehlern auf der SD-Karte gekommen sein. In dem Falle muss die SD-Karte an einem anderen Rechner mit Hilfe von fsck das Dateisystem repariert werden.

Diese aufgezählten Lösungen zu den genannten Problembeispiele decken die meisten Störungen ab, die so leicht selbst zu beheben sind.

4.6 Remote Access (Fernzugriff) auf den CubieTruck

Bei Linux gibt es mehrere Möglichkeiten im ferngesteuerten Betrieb zu arbeiten. Die erste Variante stellt keinen virtuellen Client im eigentlichen Sinn dar. Dabei wird ausgenutzt, dass unter Linux fast alles als Netzwerk intern strukturiert ist. Darunter fällt auch die klassische graphische X11-Oberfläche ohne 2D/3D-Beschleunigung. Somit ist es möglich den entfernten Rechner im Textmode zu betreiben und trotzdem auf den Clients mit X11-Anwendungen zu arbeiten, die auf dem entfernten Rechner laufen. Es gibt auch Projekte, die eine Unterstützung der 2D/3D-Beschleunigung auf dem Client ermöglichen. Allerdings hat dies immer einen Nachteil in der Performance gegenüber der Karte, die im Rechner steckt und auf die direkt zugegriffen werden kann.

Zuerst muss die aktuelle IP des CubieTruck ermittelt werden:

```
$ sudo ifconfig  
192.168.1.2
```

Filesystemzugriff per ftp/sftp:

```
sftp -o UserKnownHostsFile=.ssh/known_hosts_pi_raspian -o StrictHostKeyChecking=no  
username@192.168.1.2 -v
```

Das strenge Checking abzuschalten ist keine gute Lösung, aber nach dem der Raspi mit verschiedenen SD-Karten betrieben wird und der Key nicht gleich ist, würde es sonst nur mit einer Karte gehen, bzw. Ich immer überlegen müssen welches das richtige File mit den Schlüsseln ist.

Mit “ssh -X ...” (-X option für X11 Unterstützung) eingeloggt werden Programme folgendermaßen ausgeführt

“DISPLAY=:0.0 abiword” auf dem entfernten Rechner
“abiword” auf dem eingeloggten Client

```
# or set the environment variable for all programs  
export DISPLAY=:0.0  
xeyes &  
firefox &
```

```
ssh -o UserKnownHostsFile=.ssh/known_hosts_pi_raspian -o StrictHostKeyChecking=no  
username@192.168.1.2 -v -X
```

So geht es, wenn die ursprüngliche Karte im CubieTruck steckt, wo der Key passt.

```
ssh 192.168.1.2 -v -i .ssh/known_hosts -X  
ssh 192.168.1.2 -v -i .ssh/known_hosts_SD2
```

```
ssh username@192.168.1.2 -v -i .ssh/known_hosts_SD2
```

Anbei noch die Angaben, wie würde ein richtiger virtual Network Client eingerichtet. Beim Raspi wurde dies bereits durchgeführt. Wenn der CubieTruck dran war, wird dies vielleicht noch ergänzt.

“sudo apt-get install x11vnc” benötigt, dass 2MB heruntergeladen werden.

Aus unerfindlichen Gründen fehlte x11vnc auf dem CubieTruck. Dieses wurde wieder installiert.

```
$ sudo apt-get install x11vnc
```

```
0 upgraded, 9 newly installed, 0 to remove and 7 not upgraded.
```

```
Need to get 3,080 kB of archives.
```

```
After this operation, 8,303 kB of additional disk space will be used.
```

“sudo apt-get install xvnc4viewer” benötigt, dass 200kB heruntergeladen werden.

```
$ sudo apt-get install xvnc4viewer
```

```
xvnc4viewer is already the newest version.
```

```
0 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
```

```
“sudo x11vnc -storepasswd”
```

```
“sudo x11vnc -storepasswd KENNWORT /etc/x11vnc.pass”
```

Starten der Anwendung auf dem Rechner des Helfenden bei remoter Unterstützung:

```
“xvnc4viewer -listen 5500”
```

Starten der Anwendung auf dem Rechner des Hilfesuchenden bei remoter Unterstützung:

```
“x11vnc -connect 192.168.1.2”
```

Allerdings ohne Passwortabfrage ist das keine so gute Idee. Ungünstigerweise ist hier der schwierigere Part beim Hilfesuchenden. In dem Falle sollte der Helfende entsprechenden Befehl als Skript hinterlegt haben, so dass nur dieses einfach vom Nutzer aufzurufen wäre.

```
x11vnc -auth guess -forever -loop -noxdamage -repeat -rfbauth /home/USERNAME/.vnc/passwd  
-rfbport 5900 -shared
```

(Ggf auch host:display)

Am besten funktionierte der Zugriff auf den Raspian mit dem tightvncserver, bzw. auf dem Client können die vorher aufgeführten installierten VNC weiter verwendet werden. Ein Pluspunkt bei diesem Server ist, dass ohne angeschlossenen Monitor nicht ein Mini-Bildschirm der Auflösung Standard-VGA (640x480) oft als einzige Auflösung angeboten wird.

```
$ sudo apt-get install tightvncserver
```

```
0 upgraded, 1 newly installed, 0 to remove and 7 not upgraded.
```

```
Need to get 575 kB of archives.
```

```
After this operation, 1,144 kB of additional disk space will be used.
```

4.7 Booten von mehreren Systemen bei Bedarf

Eine weitere SD-Karte wäre hier mit „berryboot“ herzurichten um armbian, raspian oder osmc zu starten. Auch möglich wäre es den Boot-Vorgang auf einen USB-Speicherstick umzuleiten. Auf die Dauer wäre das stabiler, da die USB-Sticks in der Regel Chipsätze mit leistungsfähigeren Funktionen zur Ausblendung von ermüdeten Speicherzellen besitzen.

Angemerkt sei nun einige Monate später, dass dies bei Bedarf angegangen wird. Als eine Möglichkeit andere Linux-Distributionen versuchsweise unter Einschränkungen zu verwenden bietet auch die chroot-Umgebung oder Emulation (z.B. mit qemu) an.

5 Installation eines Druckers

5.1 Brother MFC 795

Die Suche im Internet ergab, dass es für den Drucker nur Treiber für die i386er Prozessorarchitekturen von Intel und AMD gibt. Für Armbian gibt es keine Treiber und in einem Blog steht die Antwort von Brother zitiert, dass es von Brother auch keinen Treiber geben wird für die ARM-Prozessoren. Für Android gibt es aber Treiber im google-play-store. In einer ct 18/2015 gibt es eine Anleitung, wie i386er Treiber mit Hilfe von Komponenten des Emulators qemu zum Laufen gebracht werden können.

Als wichtigster vorbereitender Schritt musste erst lpr und cups installiert werden.

```
sudo apt-get install lpr
```

```
Need to get 83.5 kB of archives.
```

```
After this operation, 294 kB of additional disk space will be used.
```

```
sudo apt-get install cups
```

```
Need to get 3,676 kB of archives.
```

```
After this operation, 10.5 MB of additional disk space will be used.
```

Anbei sei noch eine kleine Panne bei der Installation erwähnt, so dass cups und lpr noch einmal installiert werden mußten:

```
sudo apt-get install lprng
```

```
The following packages will be REMOVED:
```

```
 cups cups-client lpr
```

```
Need to get 1,114 kB of archives.
```

```
After this operation, 386 kB of additional disk space will be used.
```

```
Do you want to continue? [Y/n] y
```

Für den Scanner wird xsane bzw. sane benötigt.

```
sudo apt-get install xsane
```

Need to get 7,434 kB of archives.

After this operation, 26.5 MB of additional disk space will be used.

```
sudo apt-get install sane
```

Need to get 107 kB of archives.

After this operation, 237 kB of additional disk space will be used.

```
sudo apt-get install sane-utils
```

Need to get 257 kB of archives.

After this operation, 650 kB of additional disk space will be used.

5.2 Vorbereitungen für i386er Software auf ARM-Prozessoren

Ermöglichen, dass unter ARM-Architektur auch Binaries i386 von Treibern (z.B. Drucker) laufen.

Quelle: <https://www.lhinderberger.de/pi/2016/01/27/raspberry-pi-binary-x86-drivers.html>

Step 1: Set up dpkg

```
$ sudo dpkg --add-architecture i386
```

Step 2: Edit your apt sources.list and sources.list.d

You have to edit your /etc/apt/sources.list file and every file in /etc/apt/sources.d, adding [arch=armhf] behind each occurrence of dep to prevent confusion of apt-get.

So for example the line

```
deb http://archive.raspbian.org/raspbian jessie main contrib non-free
```

would become

```
deb [arch=armhf] http://archive.raspbian.org/raspbian jessie main contrib non-free
```

Step 3: Add i386 repositories

To do that, you create the file `/etc/apt/sources.list.d/i386.list` and insert the following:

```
deb [arch=i386] http://ftp.debian.org/debian/ jessie main contrib non-free
deb [arch=i386] http://ftp.debian.org/debian/ jessie-updates main contrib non-free
deb [arch=i386] http://security.debian.org/ jessie/updates main contrib non-free
```

Step 4: Update your local apt cache

```
$ sudo apt-get update
```

(Jeweils ca 5MB 3x, wegen eines gemachten Fehlers bei Step 2)

Retrieve GPG public keys for the newly added repositories (man-in-the-middle attacks).

Step 5: Install qemu-user and binfmt-support

```
$ sudo apt-get install binfmt-support qemu-user
```

(Need to get 3,788 kB of archives. After this operation, 29.4 MB of additional disk space will be used.)

Alternativ wurde bei anderen auf den Seiten folgende Pakete angegeben:

```
sudo apt-get install binfmt-support qemu-user-static
```

Need to get 5,155 kB of archives.

After this operation, 54.4 MB of additional disk space will be used.

```
sudo apt-get install qemu binfmt-support qemu-user-static
```

Need to get 16.5 MB/16.5 MB of archives.

After this operation, 89.3 MB of additional disk space will be used.

Step 6: Install x86 libc

Most (if not all) modern binary programs require the C standard library installed on your system to work properly. For ARM, Raspbian ships libc by default, but for x86 we have yet to install it:

```
$ sudo apt-get install libc6:i386
```

Notice the `:i386` suffix? This tells apt-get to explicitly go and fetch the package for x86, rather than for ARM.

If you made it this far: Congratulations! You can now run x86 binaries on your Raspberry Pi!

5.3 Problemlösung zu libc6 bei der Druckerinstallation

Und genau in Step 6: Install x86 libc scheiterte die Aktion auf dem CubieTruck mit Armbian. Die Ergebnisse der Suchmaschinen im Internet ergaben, dass dieses Problem vorliegt, aber es fehlte

überall eine Lösung des Problems. Ohne diese libc6 scheitert die Installation des Druckertreibers kläglich.

```
$ sudo apt-get install libc6:i386
```

```
Package libc6:i386 is not available, but is referred to by another package.
```

```
This may mean that the package is missing, has been obsoleted, or
```

```
E: Package 'libc6:i386' has no installation candidate
```

Mit einigem Probieren, konnte jedoch eine Lösung gefunden werden, die hier im folgenden skizziert wird.

```
sudo apt-get install libc6-pic
```

```
Need to get 1,208 kB of archives.
```

```
After this operation, 5,073 kB of additional disk space will be used.
```

Dies löste das Problem leider nicht. Ich kam auf Grund der Abhängigkeiten auf die Idee in den Debian-Package-Beschreibungen die Abhängigkeiten einzeln abzuklopfen. Die Pakete für i386-Architektur wurden jeweils einzeln herunter geladen und versucht mit dpkg einzeln zu installieren.

Download:

<https://packages.debian.org/jessie/libc6>

```
sudo dpkg -i libc6_2.19-18+deb8u4_i386.deb
```

```
Selecting previously unselected package libc6:i386.  
(Reading database ... 90324 files and directories currently installed.)  
Preparing to unpack libc6_2.19-18+deb8u4_i386.deb ...  
Unpacking libc6:i386 (2.19-18+deb8u4) ...  
dpkg: dependency problems prevent configuration of libc6:i386:  
 libc6:i386 depends on libgcc1.  
dpkg: error processing package libc6:i386 (--install):  
 dependency problems - leaving unconfigured  
Errors were encountered while processing:  
 libc6:i386
```

Zusätzlich wurden noch die folgenden Pakete ermittelt und herunter geladen.

<https://packages.debian.org/jessie/i386/libgcc1/download>

<https://packages.debian.org/jessie/i386/multiarch-support/download>

<https://packages.debian.org/jessie/i386/gcc-4.9-base/download>

```
dpkg: dependency problems prevent configuration of libgcc1:i386:  
 libgcc1:i386 depends on gcc-4.9-base (= 4.9.2-10).  
 libgcc1:i386 depends on libc6 (>= 2.2.4); however:  
 Package libc6:i386 is not configured yet.
```

```
sudo dpkg -i gcc-4.9-base_4.9.2-10_i386.deb
Selecting previously unselected package gcc-4.9-base:i386.
(Reading database ... 90630 files and directories currently installed.)
Preparing to unpack gcc-4.9-base_4.9.2-10_i386.deb ...
Unpacking gcc-4.9-base:i386 (4.9.2-10) ...
Setting up gcc-4.9-base:i386 (4.9.2-10) ...
```

Die Abhängigkeiten auf einem Blatt aufgemalt ergab, dass es ein Henne-Ei-Problem zwischen libc6 und libgcc1 gab. Einzeln ging es nicht, aber alle zusammen mit dpkg aufgerufen, wie unten angegeben führte zu einer erfolgreichen Installation von libc6.

```
sudo dpkg -i libgcc1_4.9.2-10_i386.deb gcc-4.9-base_4.9.2-10_i386.deb
libc6_2.19-18+deb8u4_i386.deb multiarch-support_2.19-18+deb8u4_i386.deb
```

Und nun konnte endlich der Treiber erfolgreich installiert werden.

```
Sudo bash linux-brprinter-installer-2.0.0-1
Input model name ->mfc-795cw
```

```
You are going to install following packages.
mfc795cwlpr-1.1.3-1.i386.deb
mfc795cwcupswrapper-1.1.3-1.i386.deb
brscan3-0.2.13-1.i386.deb
brscan-skey-0.2.4-1.i386.deb
```

Diese Warnung ist ein Hinweis, dass libc6 fehlt:
dpkg: warning: package architecture (i386) does not match system (armhf)
Nachdem libc6 erfolgreich installiert wurde, gab es diese Meldung nicht mehr.

```
mkdir: cannot create directory '/var/spool/lpd/mfc795cw': No such file or directory
chown: cannot access '/var/spool/lpd/mfc795cw': No such file or directory
chgrp: cannot access '/var/spool/lpd/mfc795cw': No such file or directory
chmod: cannot access '/var/spool/lpd/mfc795cw': No such file or directory
error /var/spool/lpd/mfc795cw
```

Diese Datei / dieser Pfad muss vor der Installation manuell angelegt werden. Beim Armbian fehlte der Pfad „/var/spool/lpd“. Nachdem diese angelegt wurden, konnte diese Komponente erfolgreich installiert werden.

```
sudo mkdir -p /var/spool/lpd/mfc795cw
```

Wenn vorher cups nicht installiert wurde, scheitert hier die weitere Installation.
Restarting cups (via systemctl): cups.service.

```
Will you specify the Device URI? [Y/n] ->y
8 (I): Specify IP address.
Es wurde 8 gewählt, da der Drucker am LAN (bzw. WLAN) hängt.
enter IP address ->192.168.xxx.xxx
```

Wenn vorher lpr nicht installiert wurde, scheitert hier die weitere Installation.
lpr -P MFC795CW /usr/share/cups/data/testprint
linux-brprinter-installer-2.0.0-1: line 2408: lpr: command not found

Unter der Liste von Voraussetzungen bei Brother befindet sich auch die eine oder andere

```
Information, was noch zu machen wäre bei verschiedenen Modellen.  
(http://support.brother.com/g/s/id/linux/en/before.html?  
c=us&lang=en&prod=mfc795cw_all&redirect=on)  
sudo lppasswd -g sys -a root  
Enter password: *****  
Enter password again: *****
```

Bei der Ausführung von Skripten zur Installation ist es nicht immer eindeutig, ob diese als user, user mit sudo oder als root (root-Shell) vorteilhafter ausgeführt wären. Als root kann es manchmal passieren, dass beim Anlegen der Pfade und Dateien strikt dessen Ausführungsrechte gesetzt werden und die Anwendung als root einwandfrei funktioniert, aber als user eben nicht. Diese Dateien aufzuspüren und manuell zu ändern, kann ein sehr mühsam werden. Umgekehrt kann es auch passieren, dass nach einem Abbruch der Installation fortgesetzt als root fehlende Installationsanteile ausgelassen werden, da nicht jeder Unterschnitt über das Installationsskript verifiziert wurde. Diese Fehler aufzuspüren können sehr mühsam werden. Die zeitlich bessere Lösung wäre unter diesen Umständen nicht selten, das Betriebssystem mit allen Anwendungen neu aufzusetzen oder auf das Geräte, Komponente, Werkzeug oder Anwendung zu verzichten.

Der Druck einer Testseite funktionierte nun, nachdem unter cups mittels localhost:631 im Browser (hier: iceweasel) alle Druckaufträge gelöscht wurden.

Wenn der Drucker nicht funktionieren sollte, anbei die wichtigsten Schritte zur Überprüfung:

1. Bei Netzwerkdrucker, diesen „anpingen“ ob dieser erreichbar ist:

```
$ ping 192.168.x.x
```

2. Ist der CUPS Daemon aktiv:

```
$ sudo netstat -nap | grep 631
```

Gut dargestellt ist der Vorgang auch unter:

<http://ask.xmodulo.com/install-brother-printer-linux.html>

5.4 Installation des Scanners des Multi-Funktions-Druckers

Vorab sei erwähnt, dass der Weg im Kapitel zum Scanner-Treiber unter chroot mit qemu-i386-static installieren bisher nur erfolgreich war.

5.4.1 Installationsversuch des Scanner-Treibers i386 unter der arm-Architektur

Allerdings funktioniert das scannen noch nicht. Genannte Befehle finden alle den Scanner nicht.

- `scanimage -L`

- xsane

Als root findet jedoch „brsaneconfig3 -d“ den Drucker .

```
Devices on network
```

```
  0 mfc795cw          "MFC-795CW"          I:192.168.xxx.xxx
```

```
ping
```

```
test mfc795cw
```

```
ping 192.168.xxx.xxx -w 10
```

Eingerichtet wurde der Scanner mit:

```
brsaneconfig3 -a name=mfc795cw model=MFC-795CW ip=192.168.xxx.xxx
```

```
sudo brsaneconfig3 -a name=mfc795cw model=MFC-795CW ip=192.168.xxx.xxx
```

Aus diesem Verhalten könnte vermutet werden, dass es sich um ein Problem mit den Rechte handle. Dies konnte nicht auf Anhieb gelöst werden.

Also wurden die letzten Schritte manuell im richtigen root-Account (also nicht mit sudo) noch einmal versucht durchzuführen. Als user (ohne sudo) funktionierten die Schritte nicht sei hier anzumerken.

```
root@cubietruck:# dpkg -i brscan3-0.2.13-1.i386.deb
```

```
root@cubietruck:# dpkg -i brscan-skey-0.2.4-1.i386.deb
```

```
root@cubietruck:/home/username/INSTALL# brsaneconfig3 -p
```

Der Test ergab, es ging noch nicht.

```
root@cubietruck:/home/username/INSTALL# brsaneconfig3 -a name=mfc795cw model=MFC-795CW ip=192.168.xxx.xxx
```

```
root@cubietruck:/home/dieterd/INSTALL# brsaneconfig3 -p
```

```
test mfc795cw
```

Und jetzt wird der Scanner mit dem Befehl angezeigt. Aber unter xsane immer noch Fehlanzeige.

Anbei noch einmal die Ausgaben von der Installation betreffend Brother Druckertreiber für den Scanner:

```
root@cubietruck:# bash linux-brprinter-installer-2.0.0-1
Input model name ->mfc-795cw
You are going to install following packages.
  mfc795cwlpr-1.1.3-1.i386.deb
  mfc795cwcupswrapper-1.1.3-1.i386.deb
  brscan3-0.2.13-1.i386.deb
  brscan-skey-0.2.4-1.i386.deb
OK? [y/N] ->y
... hier weggelassen .....
You are going to install following packages.
```

```

brscan3-0.2.13-1.i386.deb
dpkg -i --force-all brscan3-0.2.13-1.i386.deb
(Reading database ... 91838 files and directories currently installed.)
Preparing to unpack brscan3-0.2.13-1.i386.deb ...
Unpacking brscan3 (0.2.13-1) over (0.2.13-1) ...
Setting up brscan3 (0.2.13-1) ...
You are going to install following packages.
  brscan-skey-0.2.4-1.i386.deb
dpkg -i --force-all brscan-skey-0.2.4-1.i386.deb
(Reading database ... 91838 files and directories currently installed.)
Preparing to unpack brscan-skey-0.2.4-1.i386.deb ...
Unpacking brscan-skey (0.2.4-1) over (0.2.4-1) ...
Setting up brscan-skey (0.2.4-1) ...
brsaneconfig3 -a name=MFC-795CW model=MFC-795CW ip=192.168.xxx.xxx
Hit Enter/Return key.

```

Die Ausgaben zeigen, dass zwar der Scanner von Brother installiert wurde, aber nicht gefunden wird von den Scannprogrammen.

```

dieterd@cubietruck:~/INSTALL/Brother795$ scanimage -L
No scanners were identified. If you were expecting something different,
check that the scanner is plugged in, turned on and detected by the
sane-find-scanner tool (if appropriate). Please read the documentation
which came with this software (README, FAQ, manpages).

$ cat /usr/local/Brother/sane/brsaneconfig3.cfg
DEVICE=MFC-795CW , "MFC-795CW" , 0x4f9:0x22c , IP-ADDRESS=192.168.xxx.xxx

$ brsaneconfig3 -a name="MFC-795CW" model="MFC-795CW" ip=192.168.xxx.xxx
"MFC-795CW" is already registered.

$ sudo ping 192.168.xxx.xxx
PING 192.168.xxx.xxx (192.168.xxx.xxx) 56(84) bytes of data:
64 bytes from 192.168.xxx.xxx: icmp_seq=1 ttl=255 time=1043 ms

```

```

$ sudo service saned restart
Failed to restart saned.service: Unit saned.service is masked.

```

\$ sudo sane-find-scanner

Kein positives Ergebnis.

\$ cat /etc/sane.d/net.conf

Hier ist nichts eingetragen, wie auch beim Netbook, wo der Scanner funktioniert.

Aus kubuntuforums.net:

For network scanning you need to have your local subnet in /etc/sane.d/net.conf like this:

\$ cat /etc/sane.d/net.conf

```
# localhost
```

```
192.168.1.0/24
```

Einen solchen Eintrag habe ich auf dem Netbook auch nicht, wo alles funktioniert.

```

$ sudo systemctl status saned
[sudo] password for marc:
•saned.service

```

```
Loaded: masked (/dev/null)
Active: inactive (dead)
```

```
$ ps -edf | grep sane
$ ps -ax | grep sane
```

Einen solchen Eintrag habe ich auf dem Netbook auch nicht, wo alles funktioniert.

Analog vom Netbook übernommen funktioniert hier nicht:

```
$ scanimage -d "brother3:net1;dev0"
scanimage: open of device brother3:net1;dev0 failed: Invalid argument
```

```
$ ls /etc/systemd/system | grep saned
```

Einen solchen Eintrag habe ich auf dem Netbook auch nicht, wo alles funktioniert.

Mittlerweile generierte ich noch einen alten HP-Scanner für die Verwendung und installierte dessen Treiber, die unter Linux verfügbar waren.

```
$ sudo apt-get install sane-hp
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package sane-hp
dieterd@cubietruck:~/INSTALL/Brother795$ sudo apt-cache show sane-hp
N: Unable to locate package sane-hp
E: No packages found
dieterd@cubietruck:~/INSTALL/Brother795$ sudo apt-cache search sane-hp
libsane-hpaio - HP SANE backend for multi-function peripherals
dieterd@cubietruck:~/INSTALL/Brother795$
```

```
$ sudo apt-get install libsane-hpaio
Recommended packages:
  hplip
Need to get 4,001 kB of archives.
After this operation, 7,504 kB of additional disk space will be used.
```

```
$ scanimage -L
device `hp5590:libusb:001:005' is a HP 4500C/5550C Workgroup scanner
$ xsane
```

Der HP-Scanner am USB-Anschluss funktioniert das Scannen ohne Probleme auf antrieb.

```
$ sudo apt-get install hplip
Need to get 8,614 kB of archives.
After this operation, 17.5 MB of additional disk space will be used.
```

```
$ sudo apt-get install libsane-extras
Need to get 70.9 kB of archives.
After this operation, 215 kB of additional disk space will be used.
```

Texterkennung ergänzt:

```
$ sudo apt-get install goocr
Need to get 188 kB of archives.
After this operation, 364 kB of additional disk space will be used.
```

```
$ sudo modprobe -c | grep sane
alias nfct_helper_sane nf_contrack_sane
```

Das ist auch auf beiden Rechnern identisch.

Das ist auch auf beiden Rechnern ähnlich.

```
/usr/share/sane/
```

Noch nicht geprüft:

```
$ sudo adduser saned scanner
```

```
$ groups
```

```
dieterd dialout sudo audio video plugdev netdev
```

Es gibt dort keine Gruppe scanner.

```
$ cat /etc/group
$ getent group
$ getent passwd group (Gibt auch die berechtigten Pfade aus)
$ id dieterd
```

sane und scanner vorhanden, aber dieterd nicht verknüpft.

```
$ id dieterd
```

```
$ sudo adduser saned scanner
```

Ist schon vorhanden.

```
$ sudo adduser dieterd scanner
```

Dieses wurde ergänzt wie auch folgende Ausgabe zeigt:

```
$ id dieterd
uid=1000(dieterd) gid=1000(dieterd)
groups=1000(dieterd),20(dialout),27(sudo),29(audio),44(video),46(plugdev),108(ne
tdev),117(scanner)
$ groups
$ id -Gn
```

Hier fehlt noch scanner in der Liste;

Dies ist nur vorhanden, wenn rebootet oder erneut eingeloggt wurde. Eine Shell aufgerufen mit „su -l <user>“ zeigt die Gruppe scanner in der Liste. Also hat es funktioniert, aber eine Neustart ist notwendig.

```
sudo mkdir /srv/brscan-skey
/opt/brother/scanner/brscan-skey/brscan-skey-0.2.4-0.cfg
bash /opt/brother/scanner/brscan-skey/script/scantoimage-0.2.4-1.sh
dieterd@cubietruck:~/INSTALL/Brother795$ sh: 1: gimp: not found
```

```
$ sudo apt-get install gimp
Need to get 15.5 MB/15.6 MB of archives.
After this operation, 70.3 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

```
$ sudo systemctl --user start brscan-skey.service
Failed to get D-Bus connection: Connection refused
$ sudo systemctl start brscan-skey.service
Failed to start brscan-skey.service: Unit brscan-skey.service failed to load: No
such file or directory.
```

```
sudo /etc/init.d/dbus restart
```

Das gibt einen Absturz in die Konsole.

Befehle um Statusausgaben der Busse zu bekommen:

```
$ systemctl --user
$ systemctl --system
$ busctl --user
$ busctl --system
$ systemctl status
```

```
sudo apt-get install strace
```

Need to get 236 kB of archives.

After this operation, 468 kB of additional disk space will be used.

```
strace -o strace.out -f scanimage -T
```

```
SANE_DEBUG_DLL=3 scanimage -L
```

```
cat /lib/udev/rules.d/60-libsane.rules | less
```

Soetwas ist nicht vorhanden:

- 80-libsane.rules contains :

```
# Brother DCP-J752DW / added by gui
```

```
ATTRS{idVendor}=="04f9", ATTRS{idProduct}=="02e4", MODE="0660",
```

```
GROUP="lp", ENV{libsane_matched}="yes"
```

Auf beiden Rechnern ist identisch, dass keine solchen Angaben in der libsane.rules vorliegen.

Nun scheint eine Spur gefunden:

```
SANE_DEBUG_DLL=3 scanimage -L
```

```
[dll] add_backend: `hpaio' is already there
```

```
[dll] add_backend: adding backend `brother3'
```

```
[dll] sane_get_devices
```

```
[dll] load: searching backend `brother3' in `/usr/lib/arm-linux-gnueabi/sane:/usr/lib/sane'
```

```
[dll] load: dlopen()ing `/usr/lib/sane/libsane-brother3.so.1'
```

```
[dll] load: dlopen() failed (/usr/lib/sane/libsane-brother3.so.1: cannot open shared object file: No such file or directory)
```

Auf dem funktionierenden Notebook gibt es auch einen „segmentation fault“

```
qemu-i386-static /usr/lib/sane/libsane-brother3.so
```

```
qemu: uncaught target signal 11 (Segmentation fault) - core dumped
```

```
Segmentation fault
```

```
busctl --system | grep D
```

NAME	PID	PROCESS	USER	CONNECTION	UNIT
SESSION CONNECTION-NAME					
org.freedesktop.DBus	--	-	-	-	-

Hier ist alles leer im Vergleich zu dem Rechner mit funktionierenden Scanner.

```
$ brscan-key -l
```

```
MFC-795CW : brother3:net1;dev0 : 192.168.43.128 Active
```

```
qemu: uncaught target signal 11 (Segmentation fault) - core dumped
```

```
dieterd@cubietruck:~$ xsane
```

```
dieterd@cubietruck:~$ brscan-key -l
```

```
dieterd@cubietruck:~$ brscan-key -l
```

```
sudo systemctl start dbus
```

```
dieterd@cubietruck:~$ man systemd
```

```
dieterd@cubietruck:~$ scanimage -L
```

No scanners were identified. If you were expecting something different, check that the scanner is plugged in, turned on and detected by the sane-find-scanner tool (if appropriate). Please read the documentation which came with this software (README, FAQ, manpages).

```
systemctl --failed
UNIT                                LOAD  ACTIVE SUB  DESCRIPTION
• lprng.service                     loaded failed failed LSB: Start lpd to allow print
• systemd-modules-load.service      loaded failed failed Load Kernel Modules
```

```
LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB    = The low-level unit activation state, values depend on unit type.
```

```
2 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
dieterd@cubietruck:~$ dpkg -l | grep lprng
```

```

rc lprng 3.8.B-2 armhf lpr/lpd
printer spooling system
dieterd@cubietruck:~$ sudo apt-get -purge lprng
[sudo] password for dieterd:
E: Command line option 'p' [from -purge] is not known.
dieterd@cubietruck:~$ sudo apt-get --purge lprng
E: Command line option --purge is not understood
dieterd@cubietruck:~$ sudo apt-get install lprng
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  magicfilter lprng-doc
The following packages will be REMOVED:
  cups cups-bsd cups-client hplip printer-driver-hpcups
The following NEW packages will be installed:
  lprng
0 upgraded, 1 newly installed, 5 to remove and 0 not upgraded.
Need to get 0 B/1,114 kB of archives.
After this operation, 1,454 kB disk space will be freed.
Do you want to continue? [Y/n] n
Abort.
dieterd@cubietruck:~$ sudo apt-get purge lprng
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be REMOVED:
  lprng*
0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded.
After this operation, 0 B of additional disk space will be used.
Do you want to continue? [Y/n] y
(Reading database ... 95221 files and directories currently installed.)
Removing lprng (3.8.B-2) ...
Purging configuration files for lprng (3.8.B-2) ...
dieterd@cubietruck:~$

```

systemctl --failed

UNIT	LOAD	ACTIVE	SUB	DESCRIPTION
• lprng.service	not-found	failed	failed	lprng.service
• systemd-modules-load.service	loaded	failed	failed	Load Kernel Modules

LOAD = Reflects whether the unit definition was properly loaded.

ACTIVE = The high-level unit activation state, i.e. generalization of SUB.

SUB = The low-level unit activation state, values depend on unit type.

2 loaded units listed. Pass --all to see loaded but inactive units, too.

To show all installed unit files use 'systemctl list-unit-files'.

```
cat /etc/services | grep 6566
```

```
sane-port 6566/tcp sane saned # SANE network scanner daemon
```

Befehl um eine Anwendung mit X11-Oberfläche als root zu starten.

```
install gksu
```

Need to get 330 kB of archives.

After this operation, 2,211 kB of additional disk space will be used.

erstelle die Datei 55-libsane.rules

```
gksudo gedit /etc/udev/rules.d/55-libsane.rules
```

mit dem Inhalt

```
SYSFS{idVendor=="04f9", MODE="0666", GROUP="scanner", ENV{libsane_matched}="yes"}
ATTRS{idVendor=="04f9", MODE="0666", GROUP="scanner", ENV{libsane_matched}="yes"}
ldconfig
```

Ende Mai 2016, also etwas mehr als ein Monat später, zeigte sich nach einem „sudo apt-get update“ und „sudo apt-get upgrade“, dass einige Treiber und vor allem qemu in einer neuen Version vorlag.

The following packages will be upgraded:

```
imagemagick imagemagick-6.q16 imagemagick-common libarchive13 libexpat1 libidn11
libmagick++-6.q16-5 libmagickcore-6.q16-2 libmagickwand-6.q16-2 libpoppler-glib8 libpoppler-qt4-4
libpoppler46 librsvg2-2 librsvg2-common libssl-dev libssl1.0.0 libtasn1-6 linux-dtb-next-sunxi
linux-firmware-image-next-sunxi linux-headers-next-sunxi linux-image-next-sunxi
linux-jessie-root-next-cubietruck openssl poppler-utils qemu qemu-system qemu-system-arm
qemu-system-common qemu-system-mips qemu-system-misc qemu-system-ppc qemu-system-sparc
qemu-system-x86 qemu-user qemu-user-static qemu-utils
```

36 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

Need to get 52.8 MB of archives.

After this operation, 1,093 kB of additional disk space will be used.

Somit dürfte es sich lohnen nach diesem Update zu testen, ob der Scanner nun funktioniert und ob der Drucker noch läuft. Das Ergebnis war, weiterhin negativ, also keine Änderung.

Versuch zu ergänzen:

```
sudo kwrite /lib/udev/rules.d/60-libsane.rules
```

```
# Brother not USB but Network generally
```

```
ATTRS{idVendor}=="04f9", ENV{libsane_matched}="yes", MODE="0666"
```

```
sudo service saned
```

```
[ ok ] saned disabled; edit /etc/default/saned.
```

Defaults for the saned initscript, from sane-utils

```
# To enable under systemd please read README.Debian
```

```
# Set to yes to start saned under SysV
```

```
# set to yes 13.05.16 and not working;
```

```
RUN=no
```

```
# Set to the user saned should run as  
RUN_AS_USER=saned
```

```
sudo systemctl start saned
```

Failed to start saned.service: Unit saned.service is masked.

```
sudo service saned start
```

Failed to start saned.service: Unit saned.service is masked.

Vorher

```
[ok] saned disabled; edit /etc/default/saned.
```

Nach Rückänderung der /etc/default/saned blieb es geändert.

5.4.2 Emulation mit Qemu in einer kleinen virtuellen Debian-Umgebung i386-Architektur.

Reine Prozessemlation (ubuntuusers)

QEMU beherrscht auch die "reine" Prozessemlation, auch "User-Space-Emulation" genannt. D.h. dass anstatt eines kompletten Systems wird "nur" ein einzelnes Programm ("Binary") im Emulations-Modus ausgeführt. Die Prozess-Emulation für ein 32-bit i386 System wird z.B. mit folgendem Befehl aufgerufen:

```
qemu-i386 PROGRAMMNAME
```

Die Emulation funktioniert natürlich nur, wenn das Programm keine weiteren Bibliotheken dynamisch nachlädt. Bei den Scanner-Treibern wird anscheinend hier dynamisch nachgeladen und andere Programme mit aufgerufen. Somit führte dies nicht zu einem Erfolg.

Link: Anleitung zum Aufruf von Binaries:

<http://unix.stackexchange.com/questions/215864/running-x86-binaries-on-armv7>

<https://wiki.debian.org/RaspberryPi/qemu-user-static>

Somit wurde als neuer Lösungsweg ein minimales Debian in einer virtuellen Umgebung anvisiert, das mit der Virtualisierung qemu betrieben werden soll.

Aus folgender Quelle wurden daher einige Pakete ausprobiert:

<https://wiki.ubuntuusers.de/QEMU/>

https://en.wikibooks.org/wiki/QEMU/Networking#User_mode_networking

<https://wiki.debian.org/QEMU>

<http://qemu-buch.de/de/index.php?title=QEMU-KVM-Buch/Managementtools/Weitere>

Und Images zum Testen heruntergeladen:

<https://people.debian.org/~aurel32/qemu/i386/> (0,3...1,8GB gepackte Dateien)

<http://wiki.qemu.org/Testing> (kleine 8MB gepackte Datei)

```
apt-get install qemu32
```

Need to get 88.5 kB of archives.

After this operation, 251 kB of additional disk space will be used.

```
apt-get install qemulator
```

Need to get 2,140 kB of archives.

After this operation, 13.9 MB of additional disk space will be used.

```
apt-get install virt-goodies
```

Need to get 6,016 kB of archives.

After this operation, 13.9 MB of additional disk space will be used.

```
sudo apt-get install qemu virt-manager virt-viewer libvirt-bin
```

Need to get 4,756 kB of archives.

After this operation, 17.1 MB of additional disk space will be used.

Der Fehler scheint nun eingekreist zu sein. Ein einfaches Image von der qemu-Homepage funktioniert, aber die squeezy und wheezy Debian images eine anderen Quelle stürzen jeweils beim Start von Grub ab. Ein Versionsvergleich mit dem Netbook ergab, dass auf dem Netbook eine 2.5er Version läuft und unter Armbian noch eine 2.1er Version vorhanden ist. Somit funktioniert die Lösung des Aufsetzens einer virtuellen Maschine mit einem i386er Debian mit dem Scanner-Treiber nicht zu funktionieren, bis eine neuere qemu-Version eingespielt wurde.

Versuch mit verschiedenen Prozessoren:

```
qemu-system-i386 -cpu qemu32 debian_wheezy_i386_standard.qcow2
```

```
qemu-system-i386 -cpu kvm32 debian_wheezy_i386_standard.qcow2
```

```
qemu-system-i386 -cpu n270 debian_wheezy_i386_standard.qcow2
```

Führt später (nach Grub) zum Neustart des Images:

```
qemu-system-i386 -cpu 486 debian_wheezy_i386_standard.qcow2
```

Falls das nicht geht, könnte auch mit bochs gearbeitet werden.

```
sudo apt install bochs
```

Need to get 1,427 kB of archives.

After this operation, 4,237 kB of additional disk space will be used.

<http://bochs.sourceforge.net/doc/docbook/user/bochsrc.html>

<http://manpages.ubuntu.com/manpages/wily/man5/bochsrc.5.html>

Verschiedene Quellen und Versionen gleichzeitig Nutzen:

<http://www.binarytides.com/enable-testing-repo-debian/>

<https://www.debian.org/releases/index.de.html>

<http://askubuntu.com/questions/281763/is-there-any-prebuilt-qemu-ubuntu-image32bit-online>

```
sudo apt-get update
```

```
apt-cache policy paketname
```

```
apt-get -t testing install apache2
```

```
apt-get install apache2/testing kwrite/unstable
```

Die Variante mit „/testing“ versucht nicht alle Paket zu aktualisieren, aber die Variante mit „-t testing“ versucht dies nicht, wird aber daher wegen fehlender Abhängigkeiten öfters abbrechen.

<https://wiki.ubuntuusers.de/apt/apt-get/>

```
sudo apt-get [Option(en)] build-dep PAKET1 [PAKET2]
```

```
sudo apt-get check
```

```
sudo apt-get source
```

Adding sources.list:

```
sudo nano /etc/apt/sources.list
```

```
# adding testing and unstable for experiments
deb [arch=armhf] http://http.us.debian.org/debian testing main non-free contrib
deb-src [arch=armhf] http://http.us.debian.org/debian testing main non-free contrib
deb [arch=armhf] http://security.debian.org/ testing/updates main contrib non-free
deb-src [arch=armhf] http://security.debian.org/ testing/updates main contrib non-free
# Unstable repo main, contrib and non-free branches, no security updates here
# deb [arch=armhf] http://http.us.debian.org/debian unstable main non-free contrib
# deb-src [arch=armhf] http://http.us.debian.org/debian unstable main non-free contrib
```

```
sudo nano /etc/apt/preferences.d/my_preferences
```

```
Package: *
Pin: release a=stable
Pin-Priority: 700
```

```
Package: *  
Pin: release a=testing  
Pin-Priority: 650
```

```
Package: *  
Pin: release a=unstable  
Pin-Priority: 600
```

Das neue Paket qemu 2.5.x von Seiten testing zu installieren führte zu Abhängigkeitsproblemen. Am Schluss hatte es diesen Umfang:

```
sudo apt-get -t testing install qemu qemu-utils libgnutls30 qemu-system qemu-user qemu-system-arm qemu-system-mips qemu-system-ppc qemu-system-sparc qemu-system-x86 qemu-system-misc
```

103 upgraded, 23 newly installed, 587 to remove and 430 not upgraded.

Need to get 213 MB of archives.

After this operation, 995 MB disk space will be freed.

Mit der Folge, dass auch noch viele benötigte Anwendungen deinstalliert würden. Dieser Weg brachte also keine Problemlösung.

Ein „sudo apt-get update“ in der ersten Juni-Woche lud 20MB und „sudo apt-get upgrade“ würde 100MB laden, da dies auch Updates von „libreoffice“ enthält. Dies wurde über Nacht mitte Juni 16 durchgeführt.

Wenn ein neues Debian-Release erhältlich ist, führen die Bezeichnungen „unstable“ und „testing“ zu einer Lücke in der Datenbank des Installers apt-get. Für diesen Fall sollten in der sources.list die Namen (jessie, stretch) eingetragen werden.

```
$ sudo nano /etc/apt/sources.list
```

```
$ sudo nano /etc/apt/preferences.d/my_preferences
```

5.4.3 Aus den Quellen compielieren einer aktuelleren Qemu-Version

Unter armbian sind bereits die notwendigen essentiellen Paketeile installiert. Daher wurde bei folgenden Aufruf nichts weiter mehr installiert:

```
$ sudo apt-get install build-essential
```

Compiling für arm auf einem i386er Rechner:

<http://stackoverflow.com/questions/32347898/compiling-qemu-for-arm>

```
./configure --target-list=arm-softmmu
```

```
./configure --target-list=arm-softmmu,arm-linux-user
```

Anleitungen zum Compelieren auf dem Rechner:

<https://wiki.winehq.org/ARM>

<http://www.howtogeek.com/105413/how-to-compile-and-install-from-source-on-ubuntu/?PageSpeed=noscript>

<http://www.aboutdebian.com/compile.htm>

Heruntergeladene qemu-Quelle:

<https://github.com/gnuarmeclipse/qemu/releases/tag/v2.3.1>

(alternativ auch <http://wiki.qemu.org/Download>)

```
./configure
```

```
ERROR: glib-2.12 gthread-2.0 is required to compile QEMU
```

Lösung aus dem Internet:

```
apt-get install libglib2.0-dev zlib1g-dev
```

```
Need to get 6,667 kB of archives.
```

```
After this operation, 19.3 MB of additional disk space will be used.
```

```
$ ./configure
```

```
E: Unable to locate package pixman
```

```
$ apt-cache search pixman
```

```
$ apt-get install libpixman-1-dev
```

```
Need to get 452 kB of archives.
```

```
After this operation, 986 kB of additional disk space will be used.
```

```
$ apt-cache search libfd
```

```
$ apt-get install libfdt-dev
```

```
Need to get 21.4 kB of archives.
```

```
After this operation, 130 kB of additional disk space will be used.
```

```
$ make
```

Der Vorgang mit „make“ dauert sehr lange und wäre ideal um einen kleinen Minirechner bei Abwesenheit von ein paar Stunden durchlaufen zu lassen. Nach einer Stunden nachgesehen, war der Vorgang immer noch nicht abgeschlossen. Nach vielen hw/* war der Vorgang in der nächsten Stunde bei den Architekturen mit „a...-softmmu“ beginnend, nach 2-3 Stunden „p...-softmmu“. Nach 5h gleiches mit Architekturen „p...-linux-user“. Die Temperatur der CPU, die sonst meist um die 32 bis 34 lag, ist nun bei 45 bis 46 Grad. Beide Prozessoren sind meist voll ausgelastet. Der Speicherverbrauch liegt bei 500 bis 700MB, allerdings sind auch libre-office. Browser mit vielen Tabs und noch ein paar kleinere Anwendungen ebenfalls aktiv.

Man beachte, dass das Verzeichnis mit den Dateien von fast 60MB auf 1,1GB anwuchs. Die anschließende Installation belegte fast 110MB auf der Festplatte, beziehungsweise SD-Karte. Auf dem CubieTruck mit 1GHz Dual-Core dauerte es gute 6 Stunden bis alles compeliert war.

```
$ sudo make install
```

Vorher:

```
qemu-system-i386 --version
```

QEMU emulator version 2.1.2 (Debian 1:2.1+dfsg-12+deb8u6), Copyright (c) 2003-2008 Fabrice Bellard

```
apt-cache policy qemu
```

```
qemu:
```

```
  Installed: 1:2.1+dfsg-12+deb8u6
```

```
  Candidate: 1:2.1+dfsg-12+deb8u6
```

```
Version table:
```

```
  1:2.5+dfsg-5+b1 0
```

```
    650 http://httpredir.debian.org/debian/testing/main armhf Packages
```

```
*** 1:2.1+dfsg-12+deb8u6 0
```

```
    700 http://security.debian.org/jessie/updates/main armhf Packages
```

```
    100 /var/lib/dpkg/status
```

```
  1:2.1+dfsg-12+deb8u5a 0
```

```
    700 http://httpredir.debian.org/debian/jessie/main armhf Packages
```

Nachher:

Wurde es noch nicht angezeigt. Entweder muss erst noch ein Neustart erfolgen oder nach einem apt-get update wird das richtige Ergebnis angezeigt.

Nach dem Neustart wurde die richtige Version angezeigt:

```
qemu-system-i386 --version
```

QEMU emulator version 2.3.1, Copyright (c) 2003-2008 Fabrice Bellard

Allerdings der Paketmanager zeigt hier nicht den Ist-Zustand an, sondern wie es theoretisch nach der Paketverwaltung wäre.

```
apt-cache policy qemu
```

```
qemu:
```

```
  Installed: 1:2.1+dfsg-12+deb8u6
```

```
  Candidate: 1:2.1+dfsg-12+deb8u6
```

```
Version table:
```

```
  1:2.5+dfsg-5+b1 0
```

```
    650 http://httpredir.debian.org/debian/testing/main armhf Packages
```

```
*** 1:2.1+dfsg-12+deb8u6 0
```

```
    700 http://security.debian.org/jessie/updates/main armhf Packages
```

```
    100 /var/lib/dpkg/status
```

```
1:2.1+dfsg-12+deb8u5a 0
```

```
700 http://httpredir.debian.org/debian/ jessie/main armhf Packages
```

```
dpkg -l | grep qemu
```

```
ii  qemu                1:2.1+dfsg-12+deb8u6      armhf      fast processor emulator
```

Die Versionen sind nun etwas durcheinander:

```
$ sudo ln -s /usr/bin/qemu-system-i386 /usr/bin/qemu
```

```
$ qemu -version
```

```
QEMU emulator version 2.1.2 (Debian 1:2.1+dfsg-12+deb8u6), Copyright (c) 2003-2008 Fabrice Bellard
```

```
$ qemu-system-i386 -version
```

```
QEMU emulator version 2.3.1, Copyright (c) 2003-2008 Fabrice Bellard
```

```
$ whereis qemu
```

```
qemu: /usr/bin/qemu /usr/lib/qemu /etc/qemu /usr/local/etc/qemu /usr/share/qemu
```

```
$ whereis qemu-system-i386
```

```
qemu-system-i386: /usr/bin/qemu-system-i386 /usr/local/bin/qemu-system-i386  
/usr/share/man/man1/qemu-system-i386.1.gz
```

```
$ /usr/bin/qemu-system-i386 -version
```

```
QEMU emulator version 2.1.2 (Debian 1:2.1+dfsg-12+deb8u6), Copyright (c) 2003-2008 Fabrice Bellard
```

```
$ /usr/local/bin/qemu-system-i386 -version
```

```
QEMU emulator version 2.3.1, Copyright (c) 2003-2008 Fabrice Bellard
```

Der erste Start zeigt, dass sich hier etwas verändert hat. Man erreicht qemu nur über einen VNC-client.

```
qemu-system-i386 debian_wheezy_i386_standard.qcow2
```

```
VNC server running on `:::1:5900'
```

```
sudo apt-get install xvnc4viewer
```

```
Need to get 93.8 kB of archives.
```

```
After this operation, 308 kB of additional disk space will be used.
```

Erreichbar ist die Ausgabe über:

```
$ vncviewer localhost
```

Allerdings hat diese Version weiterhin diese Abstürze beim Start von Grub.

<http://superuser.com/questions/520408/cannot-open-sdl-support-on-qemu>

```
$ ./configure --enable-sdl
```

```
ERROR: User requested feature sdl
       configure was not able to find it.
       Install SDL devel
```

```
$ sudo apt-get install libsdl1.2-dev
```

```
0 upgraded, 52 newly installed, 0 to remove and 0 not upgraded.
```

```
Need to get 8,312 kB of archives.
```

```
After this operation, 31.7 MB of additional disk space will be used.
```

```
(./configure --cross-prefix=arm-linux-gnueabi- --target-list=arm-softmmu --enable-kvm --enable-sdl
--static)
```

```
./configure --enable-sdl
```

```
make
```

Image-Datei umwandeln:

```
cat ubuntu.12-04.xfce.x86.20141001.raw.img.tar.bz2 | bzip2 -d | tar -x
```

```
qemu-img convert -c -O qcow2 ubuntu.12-04.xfce.x86.20141001.raw.img ubuntu.12-
04.xfce.x86.20141001.raw.img.qcow2
```

```
cat ubuntu.12-04.x86.20141001.raw.img.tar.bz2 | bzip2 -d | tar -x
```

```
./configure --enable-sdl --enable-debug
```

```
/qemu-2.5.1$ ./configure --enable-sdl --enable-debug --enable-curses
```

```
Error: missing ncurses dev
```

```
$ sudo apt-cache search ncurses
```

```
libncurses5-dev
```

```
sudo apt-get install libncurses5-dev
```

```
Need to get 224 kB of archives.
```

```
After this operation, 1,089 kB of additional disk space will be used.
```

```
/qemu-2.5.1$ ./configure --enable-sdl --enable-debug --enable-curses
```

```
$ make
```

So geht es schneller, da nur für qemu-system-i386 compelliert wird.

```
/qemu-2.5.1$ ./configure --enable-sdl --enable-debug --enable-curses --target-list=i386-softmmu
/qemu-2.5.1$ sudo make install
```

Weiterhin gleiches Fehlverhalten. Debian (wheezy und stretch) in der VM auf arm-host rebootet jeweils wenn grub startet. Puppylinux 5.7 läuft jedoch, aber ist sehr langsam. Die Maus im Fenster hat sich nicht bewegt.

5.4.4 Scanner-Treiber unter chroot mit qemu-i386-static installieren

Es wurden hier folgende Schritte durchgeführt:

1. Voraussetzung ist eine durchgelaufene Installation des Brother-Installationsvorgangs, wie beschrieben mit qemu i386 von Drucker und Scanner (auch wenn der Scanner nicht funktioniert).
2. Ein lubuntu 10.04 (long term support, CD-Version) wurde heruntergeladen. In dem Falle wurde eine USB-Installation ausgenutzt und dessen squashfs ausgelesen und in ein Verzeichnis kopiert.
3. Es wurden die benötigten Dateien von qemu unter /usr/bin, die auf dem Gastgebersystem bereits installiert waren in das gleiche Verzeichnis /usr/bin der chroot-Umgebung kopiert.
4. Es wurde die Pfade proc, sys, pts mit der Option bind gemounted.
5. Es wurden Teile mit apt-get install beim Gastgeber nachinstalliert.
6. Es wurde die chroot-Umgebung gestartet und Teile mit apt-get install innerhalb der chroot-Umgebung nachinstalliert.
7. Es wurden die Komponenten mit allen Dateien zu dem Brother Drucker in ein Verzeichnis in der chroot-Umgebung heruntergeladen bzw. kopiert.
8. Der Installer von Brother wurde gestartet, lief erfolgreich durch und das Scannen funktioniert nun.

Achtung: Auf dem System Lubuntu 10.04 unter chroot darf kein wine installiert werden, da dabei lpr auf Grund von Abhängigkeitskonflikten deinstalliert wird und somit das System für Drucken zerschossen würde.

Zu Schritt 1:

Bei folgender Fehlerausgabe unter setzen eines Debug-Levels, wie im Beispiel angegeben, sollte es ebenfalls mit der Installation des Treibers funktionieren.

```
$ SANE_DEBUG_DLL=3 scanimage -L
[sanei_debug] Setting debug level of dll to 3.
[dll] sane_init: SANE dll backend version 1.0.13 from sane-backends 1.0.24
[dll] sane_init/read_dll: using config directory `/etc/sane.d/dll.d'
[dll] add_backend: adding backend `hpaio'
....
[dll] add_backend: adding backend `brother3'
[dll] sane_get_devices
[dll] load: searching backend `brother3' in `/usr/lib/arm-linux-gnueabi/h/sane:/usr/lib/sane'
[dll] load: dlopen()ing `/usr/lib/sane/libsane-brother3.so.1'
[dll] load: dlopen() failed (/usr/lib/sane/libsane-brother3.so.1: cannot open shared object file: No
```

```
such file or directory)
[dll] load: searching backend `hpaio' in `/usr/lib/arm-linux-gnueabi/hf/sane:/usr/lib/sane'
[dll] load: dlopen()ing `/usr/lib/sane/libsane-hpaio.so.1'
[dll] init: initializing backend `hpaio'
...
[dll] sane_get_devices: found 0 devices
No scanners were identified. If you were expecting something different,
check that the scanner is plugged in, turned on and detected by the
sane-find-scanner tool (if appropriate). Please read the documentation
which came with this software (README, FAQ, manpages).
[dll] sane_exit: exiting
[dll] sane_exit: calling backend `hpaio's exit function
...
[dll] sane_exit: finished
```

Zu Schritt 2:

Erst nachsehen, wo der Stick für das Einbinden (mount) zu finden wäre.

```
$ lsblk
# sda1 (output on cubietruck)
```

Den Stick Einbinden ist auf folgende zwei Methoden möglich.

```
# In case there is no automount running, you do it manually.
$ pmount /dev/sda1
# USB-stick path is now /media/sda
$ sudo mount /dev/sda1 /home/username/mypath
# USB-stick path is now /home/username/mypath
# Don't forget to create path with mkdir in advance
```

Auf dem Stick wird die Datei `lubuntu-10.04.iso` (500-600MB) für das weitere vorgehen benötigt. Diese Datei (Image) wird eingebunden. In dieser eingebundenen Datei (Image) wird die darin enthaltene Datei `dateiname.squashfs` (500-600MB) eingebunden oder deren Dateien in ein Verzeichnis entpackt.

```
$ ls * | grep lubuntu
# lubuntu-10.04.iso
$ find . | grep lubuntu
# ./lubuntu-10.04.iso
$ sudo mount -o loop lubuntu-10.04.iso /home/username/mypath_iso
# mount: /dev/loop0 is write-protected, mounting read-only
# There you need the file squashfs
$ find . | grep squash
# ./casper/filesystem.squashfs
$ sudo mount -t squashfs -o loop filesystem.squashfs
/home/username/mypath_squash
# Use -t squashfs. Then error message will tell you missing squashfs support.
# Now with sudo (or as root) you can now copy all files to an folder
# Look first how much GB you need on your drive!
$ sudo du --max-depth=1 /home/username/mypath_squash
# 1392292 /home/username/mypath_squash
# That will also need 1.4GB.
# With some additional installations it increased to nearly 1.9GB.
# It is recommended that on media is 4GB reserved for this project.
$ sudo cp -recursive /home/username/mypath_squash /home/mychroots/lubuntu10p04
```

Zu Schritt 3:

```
# Look for your qemu-files on your host system.
$ find /usr/bin | grep qemu | grep i386
# /usr/bin/qemu-system-i386
# /usr/bin/qemu-i386
# /usr/bin/qemu-i386-static
# Copy these files to your chroot /usr/sys
$ sudo cp /usr/bin/qemu-i386-static /home/mychroots/lubuntu10p04/usr/bin
```

Als weitere Option war auch daran gedacht worden unter qemu ein ganzes i386er Linux in einer virtuellen Umgebung zu betreiben. Bei unterschiedlichen Prozessor-Architekturen ist dies aber sehr langsam. Der Versuch ein Knoppix 7.2 zu booten dauerte zwischen zwei bis drei Stunden. Es war so langsam, dass es überhaupt nicht verwendet werden konnte.

```
https://fedoraproject.org/wiki/Using\_UEFI\_with\_QEMU  
http://www.gtkdb.de/index\_7\_2213.html  
wget http://wiki.qemu.org/download/efi-bios.tar.bz2  
tar xjf efi-bios.tar.bz2  
qemu -bios bios.bin -L . -hda  
/media/sda1/home/Qemu/debian_wheezy_i386_standard.qcow2  
Funktioniert nicht, schwarzes Fenster.  
https://wiki.ubuntu.com/UEFI/OVMF  
http://www.linux-kvm.org/downloads/lersek/ovmf-whitepaper-c770f8c.txt
```

Etwas später wurde folgendes gefunden:

<https://wiki.debian.org/QemuUserEmulation>

Installing the target C libraries with dpkg-cross

The target Debian package cannot be installed directly on the host, so we need to use dpkg-cross to "cross-install" the package.

For example purposes, let's assume the target system is "armel".

Install the dpkg-cross package:

```
# apt-get install dpkg-cross
```

Now download the target libc6 package from one of the Debian mirrors and install it using dpkg-cross:

```
# dpkg-cross -i -a arm libc6_<version>_armel.deb
```

Alternatively, you can install the libc6-dev-armhf-cross package from EmDebian:

```
# wget http://www.emdebian.org/debian/pool/main/g/glibc/libc6-armel-cross_2.7-18lenny6_all.deb
```

```
# dpkg -i libc6-armel-cross_2.7-18lenny6_all.deb
```

<http://unix.stackexchange.com/questions/41889/how-can-i-chroot-into-a-filesystem-with-a-different-architechure>

```
sudo service binfmt-support start
```

https://wiki.gentoo.org/wiki/Crossdev_qemu-static-user-chroot

<https://www.plop.at/de/ploplinux/arm/crossbuildchroot.html>

Hier arm unter x86 Architektur. Benötigt wird genau die umgekehrte Richtung.

https://wiki.archlinux.org/index.php/Raspberry_Pi#QEMU_chroot

<https://github.com/AlbrechtL/RPi-QEMU-x86-wine>

Nicht uninteressant. Raspi mit qemu+wine-Kombination.

<https://help.ubuntu.com/community/BasicChroot>

<https://grinux.wordpress.com/2012/02/24/setting-up-a-network-inside-a-chroot/>

In der chroot-Umgebung diese Datei editieren und den Wert vom Gastgebersystem übernehmen. Dann hat diese auch Internetverbindung und ins Netzwerk, z.B. den Drucker.

```
$ sudo mount -o bind /proc /home/i386/mnt3/qemu_chroot_i386/proc
```

```
$ sudo mount -o bind /sys /home/i386/mnt3/qemu_chroot_i386/sys
```

(Für Internet reicht meist nur /proc zu mounten.)

Zur Fehlermeldung „Can not write log, openpty() failed (/dev/pts not mounted?)“ spuckte die Suchmaschine im Internet aus, dass noch etwas gemountet werden muss:

```
$ sudo mount -o bind /dev/pts /home/i386/mnt3/qemu_chroot_i386/dev/pts
```

Danach funktionierte bei lterminal auch die Tastatureingaben.

D-Bus library appears to be incorrectly set up; failed to read ... uuid: Failed to open "/var/lib/dbus/machine-id"

Somit ggf auch /var/lib/dbus ebenfalls mit der Option bin zu mounten.

```
$ cat /etc/resolv.conf
```

```
$ nano /etc/resolv.conf
```

<https://wiki.ubuntuusers.de/QEMU/>

Qemu Optionen und Verwendung von squashfs.

https://fedoraproject.org/wiki/How_to_use_qemu

<http://wiki.qemu.org/download/qemu-doc.html>

Creating squashfs:

http://elinux.org/Squash_FS_Howto

<http://tldp.org/HOWTO/SquashFS-HOWTO/creatingandusing.html>

Making it writeable:

As mentioned, another interesting use for SquashFS is with Unionfs filesystem, which provides copy-on-write semantics for the read-only file systems, enhancing the possibilities.

Create the ro.fs squashed file system and the rw.fs dir.

```
bash# mksquashfs /home/user1 ro.fs
```

```
bash# mkdir /home/rw.fs
```

Mount the squashed ro.fs file system using the loopback device

```
bash# mount -t squashfs ro.fs /mnt -o loop
```

mount the unionfs file system, that makes /mnt and /home/rw.fs apparently merged under /home/user1 location.

```
bash# cd /home
```

```
bash# mount -t unionfs -o dirs=rw.fs=rw:/mnt=ro unionfs user1
```

As you can see, now you can create new files in /home/user1.

Für UEFI/BIOS:

https://fedoraproject.org/wiki/Using_UEFI_with_QEMU

<https://wiki.ubuntu.com/UEFI/OVMF>

<https://github.com/qemu/qemu/tree/master/pc-bios>

<https://forums.gentoo.org/viewtopic-p-7206924.html>:

There's a (undocumented) -no-kvm CMDline switch.

Mit Hilfe von chroot ein defektes Linux reparieren:

<http://superuser.com/questions/111152/whats-the-proper-way-to-prepare-chroot-to-recover-a-broken-linux-installation>

For Old Releases with Ubuntu that apt-get is working again:

<http://askubuntu.com/questions/91815/how-to-install-software-or-upgrade-from-an-old-unsupported-release>

<http://old-releases.ubuntu.com/releases/>

<https://wiki.ubuntuusers.de/sources.list/>

Für die graphische Ausgaben unter chroot:

<https://help.ubuntu.com/community/BasicChroot>

Install xhost: Das war schon vorhanden (Komponente von x-server)

```
$ sudo apt-get install xnest  
(ca. 2,5MB zum Herunterladen)
```

```
$ sudo apt-get install metacity
```

Fehlermeldung im chroot:

```
@chroot@cubietruck $ sudo apt-get install xnest
```

```
Can not write log, openpty() failed (/dev/pts not mounted?)
```

```
(Reading database ... 81597 files and directories currently installed.)
```

```
Preparing to replace xserver-common 2:1.7.6-2ubuntu7 (using .../xserver-common_2%3a1.7.6-2ubuntu7.12_all.deb) ...
```

```
Unpacking replacement xserver-common ...
```

```
Selecting previously deselected package xnest.
```

```
$ apt-cache search xnest
```

```
sabayon - system administration tool to manage GNOME desktop settings
```

```
tsclient - front-end for viewing of remote desktops in GNOME
```

```
xoo - graphical wrapper around Xnest/Xephyr
```

```
xnest - Nested X server
```

```
xserver-xephyr - nested X server
```

https://awesome.naquadah.org/wiki/Using_Xephyr

<https://help.ubuntu.com/community/BasicChroot>

```
$ sudo apt-get install metacity
```

```
0 upgraded, 10 newly installed, 0 to remove and 8 not upgraded.
```

```
Need to get 24.0 MB of archives.
```

```
After this operation, 52.4 MB of additional disk space will be used.
```

<https://help.ubuntu.com/community/DebootstrapChroot>

Das Paket debootstrap erzeugt eine minimale chroot-Umgebung zur weiteren Verwendung.

Nachdem nun einiges experimentiert wurde, wurden noch ein paar Pakete nachgeladen in der chroot-Umgebung.

```
$ sudo apt-get update
```

```
($ sudo apt-get upgrade)
```

```
$ sudo apt-get install lpr
```

Wegen Fehlers beim Installieren mußte noch /dev/pts gemountet werden.

```
$ sudo apt-get purge lpr
```

```
$ sudo apt-get install lpr
```

```
$ sudo apt-get install cups
```

Hier gab es einen DBUS Fehler. Als diese Meldung kopiert werden sollte, verabschiedete sich der Rechner. Die Fenster gingen zu, die Leisten verschwanden. Der DBUS war weg.

```
$ sudo mount /dev/mmcblk0p5 /home/i386/mnt3
```

```
$ sudo mount -o bind /sys /home/i386/mnt3/qemu_chroot_i386/sy
```

```
$ sudo mount -o bind /proc /home/i386/mnt3/qemu_chroot_i386/proc
```

```
$ sudo mount -o bind /dev/pts /home/i386/mnt3/qemu_chroot_i386/dev/pts
```

```
$ xhost +
```

```
$ sudo chroot /home/i386/mnt3/qemu_chroot_i386/ /usr/bin/qemu-i386-static /bin/bash
```

```
@cubietruck:/# export DISPLAY=:0.0
```

```
sudo apt-get install sane
```

```
E: dpkg was interrupted, you must manually run 'sudo dpkg --configure -a' to correct the problem.
```

```
$ sudo apt-get install sane sane-utils sanex
```

Was war hier nur passiert?

```
Setting up sane-utils (1.0.20-13ubuntu2) ...
```

```
Adding saned group and user...
```

```
Removing any system startup links for /etc/init.d/saned ...
```

In dem Fenster zeigt ls nichts mehr an.

```
$ sudo apt-get install tcl
```

```
@cubietruck:/home/dieterd/INSTALL/Brother795# bash linux-brprinter-installer-2.0.0-1
```

```
Input model name ->mfc-795
```

You are going to install following packages.

```
mfc795cwlpr-1.1.3-1.i386.deb
```

```
mfc795cwcupswrapper-1.1.3-1.i386.deb
```

```
brscan3-0.2.13-1.i386.deb
```

```
brscan-skey-0.2.4-1.i386.deb
```

```
OK? [y/N] ->y
```

```
dpkg -x mfc795cwlpr-1.1.3-1.i386.deb /
```

```
dpkg -x mfc795cwcupswrapper-1.1.3-1.i386.deb /
```

```
dpkg-deb: building package `mfc795cwlpr' in `mfc795cwlpr-1.1.3-1a.i386.deb'.
```

```
dpkg -b ./brother_driver_packdir mfc795cwlpr-1.1.3-1a.i386.deb
```

```
dpkg-deb: building package `mfc795cwcupswrapper' in `mfc795cwcupswrapper-1.1.3-1a.i386.deb'.
```

```
dpkg -b ./brother_driver_packdir mfc795cwcupswrapper-1.1.3-1a.i386.deb
```

```
dpkg -i --force-all mfc795cwlpr-1.1.3-1a.i386.deb
```

```
Selecting previously deselected package mfc795cwlpr.
```

```
(Reading database ... 83934 files and directories currently installed.)
```

```
Unpacking mfc795cwlpr (from mfc795cwlpr-1.1.3-1a.i386.deb) ...
```

```
Setting up mfc795cwlpr (1.1.3-1) ...
```

```
dpkg -i --force-all mfc795cwcupswrapper-1.1.3-1a.i386.deb
```

```
Selecting previously deselected package mfc795cwcupswrapper.
```

```
(Reading database ... 83957 files and directories currently installed.)
```

```
Unpacking mfc795cwcupswrapper (from mfc795cwcupswrapper-1.1.3-1a.i386.deb) ...
```

```
Setting up mfc795cwcupswrapper (1.1.3-1) ...
```

```
Stopping printer spooler: lpd [not running]
```

```
* Restarting Common Unix Printing System: cupsd
```

```
...done.
```

```
#
```

```
The security level of AppArmor has been lowered. (aa-complain cups)
```

```
aa-complain cupsd
```

```
Can't find system log.
```

```
Will you specify the Device URI? [Y/n] ->y
```

```
0: ipp
1: socket
2: http
3: lpd
4: scsi
5: smb
6 (I): Specify IP address.
7 (A): Auto. (usb://dev/usb/lp0)
```

select the number of destination Device URI. ->6

enter IP address ->192.168.43.128

```
lpadmin -p MFC795CW -v socket://192.168.43.128 -E
```

```
Test Print? [y/N] ->n
```

You are going to install following packages.

```
brscan3-0.2.13-1.i386.deb
```

```
dpkg -i --force-all brscan3-0.2.13-1.i386.deb
```

Selecting previously deselected package brscan3.

(Reading database ... 83961 files and directories currently installed.)

```
Unpacking brscan3 (from brscan3-0.2.13-1.i386.deb) ...
```

```
Setting up brscan3 (0.2.13-1) ...
```

You are going to install following packages.

```
brscan-skey-0.2.4-1.i386.deb
```

```
dpkg -i --force-all brscan-skey-0.2.4-1.i386.deb
```

Selecting previously deselected package brscan-skey.

(Reading database ... 83981 files and directories currently installed.)

```
Unpacking brscan-skey (from brscan-skey-0.2.4-1.i386.deb) ...
```

```
Setting up brscan-skey (0.2.4-1) ...
```

```
brsaneconfig3 -a name=MFC-795CW model=MFC-795CW ip=192.168.43.128
```

Hit Enter/Return key.

```
root@cubietruck:/home/dieterd/INSTALL/Brother795# scanimage -L
```

```
qemu: Unsupported syscall: 101
qemu: Unsupported syscall: 101
device `brother3:net1;dev0' is a Brother MFC-795CW MFC-795CW
root@cubietruck:/home/dieterd/INSTALL/Brother795# scanimage -L
qemu: Unsupported syscall: 101
qemu: Unsupported syscall: 101
device `brother3:net1;dev0' is a Brother MFC-795CW MFC-795CW
root@cubietruck:/home/dieterd/INSTALL/Brother795# man scanimage
root@cubietruck:/home/dieterd/INSTALL/Brother795# scanimage -d "brother3:net1;dev0"
>image.pnm
scanimage: rounded value of br-x from 215.9 to 215.88
scanimage: rounded value of br-y from 355.6 to 355.567
```

Und nun funktionierte auch das Scannen.

Eine Liste von funktionierenden Anwendung unter der chroot-Umgebung.

Abiword, lxterminal, gparted, xsane, gnumeric, xchat, midori, netsurf

gnumeric, funktioniert trotz der Fehlermeldungen:

```
process 4344: D-Bus library appears to be incorrectly set up; failed to read machine uuid: Failed to
open "/var/lib/dbus/machine-id": No such file or directory
```

See the manual page for dbus-uuidgen to correct this issue.

```
(/usr/bin/gnumeric:4344): Gdk-WARNING **: shmat failed: error 22 (Invalid argument)
```

Eine Liste von nicht funktionierenden Anwendung unter der chroot-Umgebung.

Firefox, cromium-broser, x-www-browser, links2

Folgender Fehler:

```
# firefox
```

```
process 24229: D-Bus library appears to be incorrectly set up; failed to read machine uuid: Failed to
open "/var/lib/dbus/machine-id": No such file or directory
```

See the manual page for dbus-uuidgen to correct this issue.

Segmentation fault

```
sudo mount -o bind /var/lib/dbus /home/i386/mnt3/qemu_chroot_i386/var/lib/dbus
```

Dann bleibt nur noch der Segmentation fault übrig.

Ergänzungen:

```
# sudo apt-get install midori links2 netsurf
```

6 Anwendungen, Installation, Einrichtung und kleine Skript-Werkzeuge

6.1 Terminal und Text-Editoren

Ein Terminal, das auch Reiter zuläßt, wie auch kopieren und einfügen mit der Maus.

```
sudo apt-get install lxterminal
```

Need to get 1,221 kB of archives.

After this operation, 3,259 kB of additional disk space will be used.

Darunter kann nano und vim als Editoren verwendet werden.

6.2 Dateimanager

```
sudo apt-get install dolphin
```

Need to get 2,447 kB of archives.

After this operation, 5,976 kB of additional disk space will be used.

Programm zur grafischen Dateien De/Komprimierung:

```
apt-get install ark
```

Need to get 342 kB of archives.

After this operation, 1,074 kB of additional disk space will be used.

6.3 Medienplayer „mplayer“

6.3.1 Installation und Konfiguration

Installiert wird der Medienabspieler für die Verwendung auf der Konsole mit:

```
sudo apt-get install mplayer
```

Installiert wird der Medienabspieler für die Verwendung auf der graphischen Oberfläche mit:

```
sudo apt-get install gnome-mplayer
```

Einstellungen und Konfigurationsfiles sind zu finden unter:

```
/home/dieterd/.mplayer/
```

```
cat /home/dieterd/.mplayer/config
# Write your default config options here!
# set volume (prevent from shocking full volume at start)
volume=28
# set equalizer
# Possible: [Mixer] No hardware mixing, inserting volume filter.
#af=equalizer=2:1:0:0:0:0:0:0:0
Wegen der Meldung Failed to open /home/dieterd/.mplayer/input.conf wurde die Datei in input.conf
umbenannt. Da die vorherige Datei die einzige Datei in dem Verzeichnis war, wurden die Einstellungen
wohl trotzdem angenommen.
```

Zuerst wurde die Datei gefunden und verwendet. Nach einem Update/Upgrade und Einstellungen unter gnome-mplayer Menu gibt es eine Fehlermeldung, dass diese Datei nicht gefunden wurde:

```
/home/dieterd/.mplayer/input.conf.
```

```
/home/dieterd/.config/gnome-mplayer/
```

6.3.2 Kleine Script-Werkzeuge

Beispiel eines kleinen Scripts zum Internet-Radio hören und einer einfachen Auswahl von Sendern.

```
cat /home/username/tools/radio_01.sh
#!/bin/bash

dummy="n"

while [ $dummy != "q" ] ; do

echo "Select Radio Stream"
echo "1: "
echo "2: "
echo "3: "
echo "q: quit"

read dummy

case $dummy in
q) echo "quit select" ; exit
;;
1) mplayer -playlist http://dir.xiph.org/listen/1685349/listen.m3u
;;
2) mplayer -playlist http://dir.xiph.org/listen/1685348/listen.m3u
;;
3) mplayer -playlist http://dir.xiph.org/listen/1685353/listen.m3u
;;
esac
done
```

Streams verschiedener Codecs sind zum Beispiel zu finden unter folgendem Link. Ein Nachteil ist jedoch, dass jedesmal wenn die Rangliste erneuert wird, diese nicht mehr funktionieren. Praktisch ist jedoch, dass hier mit geeigneter Suche gezielt nach Streams mit nur 32kbit (AAC+) gesucht werden kann. Für Unterwegs mit einem Smartphone und bereits verbrauchten High-Speed-Traffic Guthaben ist das sehr praktisch.

http://dir.xiph.org/by_format/AAC+

http://dir.xiph.org/by_format/Ogg_Vorbis

Anbei ein kleines Script zum zufälligen Abspielen von Mediendateien. Über Tastaturkürzel können Dateien auch in andere Dateiodner verschoben werden, um zum Beispiel diese etwas auszusortieren.

```
cat home/username/tools/mplay_random_02.sh
#!/bin/bash
IFS_SIK=IFS ; IFS='${t\n}' ; echo "how many days ago (max)?" ; read bb ; if [ -z "$bb" ] ; then
bb=99999999 ; echo $bb;fi ; echo "how many days ago (min)?" ; read bbmin ; if [ -z "$bbmin" ] ; then
bbmin=-1 ; echo $bbmin;fi ; echo "how many files to skip?" ; read b ; if [ -z "$b" ] ; then b=0 ; echo
$b;fi ;
a=0 ; for i in $(for j in $(find . "$1" "$2" -type f -mtime -$bb -size +600k | grep -v "jpg$" | grep
-v "JPG$" | sort); do echo $(find "$j" -type f -mtime +$bbmin); done ); do let a=$a+1; echo $a " "
$i ; if [ "$a" -gt "$b" ] ; then
echo $a " " $i ;
# vidarray["$(echo $a)"]="$(echo $i)" ;
vidarray["$a"]="$i" ;
echo ${vidarray["$a"]} ;
fi ; done

echo "number of files: " $a;

for iii in $(seq 1 200) ; do

kk=$((($RANDOM % $a + 1)) ; echo "random: " $kk;

i=${vidarray[$kk]};
# mplayer options for video:
# - for armbian & cubietruck use „-vo sdl“
# - for armbian & cubietruck works too „-vo x11“ but some codecs no fullscreen
# mplayer -vo xv -geometry 0:0 -fs "$i" ; echo $a " " $kk " " $i ; echo "press key for actions";
mplayer -vo sdl -geometry 0:0 -fs "$i" ; echo $a " " $kk " " $i ; echo "press key for
actions";

# mplayer -vo xv -geometry 0:0 -fs $i ; echo $a " " $i ; echo "press key for actions";
read -t 2 -n 1 dd;
if [ "$dd" -eq "1" ] ; then b=$((a+10));fi;
if [ "$dd" -eq "2" ] ; then b=$((a+20));fi;
if [ "$dd" -eq "3" ] ; then b=$((a+50));fi;
if [ "$dd" -eq "4" ] ; then b=$((a+100));fi;
if [ "$dd" = "z" ] ; then exit ;fi;
if [ "$dd" = "k" ] ; then konqueror -select "$i" ;fi;
if [ "$dd" = "d" ] ; then dolphin -select "$i" ;fi;
if [ "$dd" = "p" ] ; then echo "hit key to continue"; read -n 1 dummy ;fi;
if [ "$dd" = "c" ] ; then echo "change category"; read catgor ;
if [ "$catgor" = "m" ] ; then echo "m selected - hit key to continue";
read -n 1 dummy ;
ii=$(echo $i | sed -e "s/\./\./catogorie01/" ) ; mkdir -p $(dirname $ii); mv $i $ii ;
fi;
if [ "$catgor" = "n" ] ; then echo "n selected - hit key to continue";
read -n 1 dummy ;
ii=$(echo $i | sed -e "s/\./\./catogorie02/" ) ; mkdir -p $(dirname $ii); mv $i $ii ;
fi;
if [ "$catgor" = "mn" ] ; then echo "mn selected - hit key to continue";
read -n 1 dummy ;
ii=$(echo $i | sed -e "s/\./\./catogorie03/" ) ; mkdir -p $(dirname $ii); mv $i $ii ;
fi;
fi;
```

```
# fi ;
done ; IFS=IFS_SIK
```

Anbei ein ähnliches Skript zum sortierten Abspielen von Mediendateien. Man kann auch einen Teil der Dateien durch Tastatureingaben überspringen.

```
cat home/username/tools/mplay_all_05.sh
#!/bin/bash
IFS_SIK=IFS ; IFS=$'\t\n' ; echo "how many days ago (max)?" ; read bb ; if [ -z "$bb" ] ; then
bb=999999999 ; echo $bb;fi ; echo "how many days ago (min)?" ; read bbmin ; if [ -z "$bbmin" ] ; then
bbmin=-1 ; echo $bbmin;fi ; echo "how many files to skip?" ; read b ; if [ -z "$b" ] ; then b=0 ; echo
$b;fi;
a=0 ; for i in $(for j in $(find -type f -mtime -$bb -size +600k | grep -v "jpg$" | grep -v "JPG$"
| sort); do echo $(find "$j" -type f -mtime +$bbmin); done ); do let a=$a+1; echo $a " " $i ; if
[ "$a" -gt "$b" ] ; then
mplayer -vo xv -geometry 0:0 -fs $i ; echo $a " " $i ; echo "press key for actions";
read -t 2 -n 1 dd;
if [ "$dd" -eq "1" ] ; then b=$((a+10));fi;
if [ "$dd" -eq "2" ] ; then b=$((a+20));fi;
if [ "$dd" -eq "3" ] ; then b=$((a+50));fi;
if [ "$dd" -eq "4" ] ; then b=$((a+100));fi;
if [ "$dd" = "z" ] ; then exit ;fi;
if [ "$dd" = "k" ] ; then konqueror -select "$i" ;fi;
if [ "$dd" = "d" ] ; then dolphin -select "$i" ;fi;
if [ "$dd" = "p" ] ; then echo "hit key to continue"; read -n 1 dummy ;fi;
if [ "$dd" = "c" ] ; then echo "change category"; read catgor ;
if [ "$catgor" = "m" ] ; then echo "m selected - hit key to continue";
read -n 1 dummy ;
ii=$(echo $i | sed -e "s/\./\./m/" ) ; mkdir -p $(dirname $ii); mv $i $ii ;
fi;
if [ "$catgor" = "n" ] ; then echo "n selected - hit key to continue";
read -n 1 dummy ;
ii=$(echo $i | sed -e "s/\./\./n/" ) ; mkdir -p $(dirname $ii); mv $i $ii ;
fi;
if [ "$catgor" = "mn" ] ; then echo "mn selected - hit key to continue";
read -n 1 dummy ;
ii=$(echo $i | sed -e "s/\./\./mn/" ) ; mkdir -p $(dirname $ii); mv $i $ii ;
fi;
fi;
fi ; done ; IFS=IFS_SIK
```

Anzeige des Spannung und des Stromverbrauchs:

```
cat /sys/power/axp_pmu/ac/
```

```
amperage connected used voltage
```

```
$ cat /sys/power/axp_pmu/ac/*
```

```
$ cat /sys/devices/platform/soc@01c00000/1c2ac00.i2c/i2c-0/0-0034/ac/amperage
```

```
$ cat /sys/power/axp_pmu/ac/amperage
```

```
$ cat /sys/power/axp_pmu/ac/voltage
```

Anzeige der Temperaturen:

```
cat /sys/class/hwmon/hwmon0/device/hwmon/hwmon0/temp1_input
```

```
cat /sys/devices/virtual/thermal/thermal_zone0/temp
```

```
33700
```

Oder Ausgabe über Durchsuche der Verzeichnisse:

```
for i in $(sudo find /sys | grep amper) ; do cat "$i" ; done
for i in $(sudo find /sys | grep temp) ; do cat "$i" ; done
for i in $(sudo find /sys | grep volt | grep "/ac\|/pmu" ) ; do echo "$i" ; cat "$i" ; done
```

Ausgaben nach dem Login:

```
/etc/update-motd.d
bash /etc/update-motd.d/30-sysinfo
```

Alle Ausgaben auf einmal:

```
$ cat /sys/power/axp_pmu/ac/amperage /sys/power/axp_pmu/ac/voltage
/sys/devices/virtual/thermal/thermal_zone0/temp
```

Ausgabe der aktuellen Leistung in mW:

```
$ echo $(( $(cat /sys/power/axp_pmu/ac/amperage) * $(cat /sys/power/axp_pmu/ac/voltage) / 1000 /
1000 /1000 ))
```

Ausgabe: 1470

Mit einem Leistungsmessgerät (KD 302 von www.reichelt.de) wurden ein Verbrauch von maximal 4,1W beim Booten und sonst meist Werte um 2,6...3.1 Watt inklusive Netzteilverluste (Ladegerät eines Smartphone von ZTE) gemessen.

6.4 Media Center

<http://raspberrypi.tips/raspi-media-center/osmc-vs-openelec-unterschiede-der-raspberry-pi-media-center-betriebssysteme/>

Die Auswahl des richtigen Systems solltet ihr daher an euren persönlichen Anforderungen fest machen:

- Wollt ihr ein integriertes System das schnell läuft, einfach zu bedienen ist, eine große Community hat aber dafür keine tieferen Eingriffe ins System zulässt dann seit ihr bei **OpenELEC genau richtig.**
- Wollt ihr ein System mit leichtgewichtiger Oberfläche das zusätzliche Funktionen und Customizing bis in die Linux tiefen ohne Hürden zulässt (was u.U. Probleme mit sich bringt) dann solltet ihr euch für **OSMC entscheiden.**

```
$ sudo apt-get install kodi
Some packages could not be installed. This may mean that you have
requested an impossible situation or if you are using the unstable
distribution that some required packages have not yet been created
or been moved out of Incoming.
The following information may help to resolve the situation:
The following packages have unmet dependencies:
 kodi : Depends: kodi-bin (>= 16.1+dfsg1-1) but it is not going to be installed
        Depends: kodi-bin (< 16.1+dfsg1-1.1~) but it is not going to be installed
E: Unable to correct problems, you have held broken packages.
```

Bei mplayer (ist bereits installiert) gab es eine ähnliche Fehlermeldung. Es kann daran liegen, dass auch die Repositories von „unstable“ und „testing“ eingebunden wurden.

```
$ sudo apt-get install dragonplayer
Need to get 414 kB of archives.
After this operation, 838 kB of additional disk space will be used.
Get:1 http://httpredir.debian.org/debian/ jessie/main dragonplayer armhf 4:4
```

6.5 Datenverkehrsmonitoring

Installation eines Programms für die Zählung des Datenverkehrs, das wenig Ressourcen beansprucht.

"sudo apt-get install vnstat"

```
vnstat -l -i wlan0
```

Zeigt den aktuellen Datenverkehr über das WLAN an.

```
vnstat -h -i wlan0
```

Gibt den Datenverkehr der letzten 24h aus.

```
vnstat -m -i wlan0
```

Gibt den Datenverkehr der letzten 12 Monate aus. Das ist eine gute Möglichkeit zu prüfen, ob das Gerät auch genutzt wurde oder nur herumlag.

```
cat /home/username/tools/vnstat_01.sh
#!/bin/bash
```

```
finish="n"
while [ $finish != "q" ]; do
  echo 'vnstat MENUE'
  echo '1: vnstat -l -i wlan0'
  echo '2: vnstat -h -i wlan0'
  echo '3: vnstat -d -i wlan0'
  echo '4: vnstat -m -i wlan0'
  echo '5: vnstat -l -i eth0'
  echo '6: vnstat -l -i eth0'
  echo '7: vnstat -l -i eth0'
  echo '8: vnstat -l -i eth0'
  echo 'q: Quit'
  read aa
  case $aa in
    q) finish="q"
       exit
       ;;
    Q) finish="q"
       exit
       ;;
    1) vnstat -l -i wlan0
       ;;
    2) vnstat -h -i wlan0
       ;;
  esac
done
```

```
;;
3) vnstat -d -i wlan0
;;
4) vnstat -m -i wlan0
;;
5) vnstat -l -i eth0
;;
6) vnstat -h -i eth0
;;
7) vnstat -d -i eth0
;;
8) vnstat -m -i eth0
;;
esac
done
```

Beispiel einer stündlichen Ausgabe von vnstat in eine Datei:

Hierzu lege man unter /etc/cron.hourly eine Datei vnstat (mit den Rechten -rwxr-xr-x, bzw. chmod 755) mit folgendem Inhalt an:

```
#!/bin/sh
vnstat -h -i wlan0 >> /home/dieterd/INSTALL/vnstat_hourly_$(date +%Y%m%d_%H%M).txt
```

6.6 Netzwerkanwendungen

```
sudo apt-get install ftp
```

Need to get 51.6 kB of archives.

After this operation, 124 kB of additional disk space will be used.

```
sudo apt-get install telnet
```

Need to get 63.8 kB of archives.

After this operation, 131 kB of additional disk space will be used.

```
sudo apt-get install filezilla
```

Need to get 8,066 kB of archives.

After this operation, 26.8 MB of additional disk space will be used.

Installation eines einfachen IRC-Clients zum Chatten unter Nutzung bekannter Servern, wie zum Beispiel freenode.

```
sudo apt-get install xchat
```

Need to get 832 kB of archives.

After this operation, 4,021 kB of additional disk space will be used.

6.7 Anwendungen zur Netzwerkuntersuchung

Um das Netzwerk zu scannen, wurde nmap installiert mittels "sudo apt-get install nmap".

Interessant war, dass manchmal der Port 1352 als offen beim Smartphone ausgegeben wurde.

```
$ nmap 192.168.xxx.xxx
PORT      STATE      SERVICE
53/tcp    open       domain
1352/tcp   filtered   lotusnotes
```

Das Ergebnis ist aber nur selten reproduzierbar.

Um den Netzwerkverkehr sehen zu können, wurde etherape installiert mittels "sudo apt-get install etherape".

Das Startmenü erhielt eine weitere Auswahl "Systemtools" unter dem sich etherape befindet. Erst nach einem Neustart befand sich dort auch etherape (root), welches erst funktioniert, nachdem beim Aufruf die Passwordeingabe erfolgte um Zugriff auf das WLAN zu erhalten. Unter der minimalen Installation der graphischen Oberfläche xfce wurde die zusätzliche Auswahl etherape(root) nicht beim normalen Nutzer angelegt. Die andere Methode war die Eingabe von "sudo etherape" im Terminal, die hier natürlich funktionierte.

Beim Surfen in Internet empfehle ich „etherape“ parallel laufen zu lassen und zuzuschauen, welche Verbindungen beim Aufrufen von wenigen Seiten hier auftauchen. Sehr schön zu sehen finde ich dabei, wie viele Querverbindungen die Scripte auf den Seiten zur Nutzeranalyse von Seiten google, facebook, und alle anderen Anbieter solcher Leistungen auftauchen.

Installation eines Datenverkehrbegrenzungsprogramms auf der Konsolenebene für zum Beispiel einzelne Aktionen mit wget oder apt-get.

```
"sudo apt-get install trickle"
Need to get 35.7 kB of archives.
After this operation, 134 kB of additional disk space will be used.
```

```
"trickle -d 3 -u 1 sudo apt-get install iceweasel"
"sudo trickle -d 3 -u 1 sudo apt-get install iceweasel"
"sudo trickle -d 3 -u 1 apt-get install iceweasel"
Das scheint nur bedingt zu funktionieren. Vermutlich geht es nach einem Neustart besser.
"sudo trickle -d 6 -u 2 apt-get install iceweasel"
```

Die Verwendung von trickle führt zu häufigeren Wiederaufnahmen des Ladevorganges. Die Datenrate schwankt relativ stark, so dass man meint die Begrenzung funktioniert nicht. Aber es läßt sich unter Verwendung eines mobilen GSM-Netz (G2-Mode mit 56/64 kbit/s) hier merklich besser parallel surfen.

6.8 Dokumentenbearbeitung – Office-Suiten

Zwischen den Programmen (Abiword und LibreOffice) und Dateitypen (Abiword) bleibt der Text zwar erhalten, aber die Nummerierungen der Kapitel geht verloren. Somit sollten die umfangreicheren Texte besser mit LibreOffice geschrieben werden. Der Vorteil von Abiword ist, dass es weniger Ressourcen benötigt und noch ein paar weitere seltenere Datei-Typen konvertieren kann.

```
sudo apt-get install libreoffice-help-de
```

<http://extensions.libreoffice.org/extension-center/german-de-de-frami-dictionaries>

```
dict-de_de-frami_2015-12-28.oxt
```

Unter dem Reiter Tools mit Extensionsmanager die Datei auswählen zum Installieren für LibreOffice. Bei Freigabe für alle Nutzer sind Adminrechte erforderlich. Daher wird empfohlen diese nur für den Nutzer auszuwählen. In dem Falle werden diese Dateien in dessen Profilverzeichnis gespeichert.

Werkzeug zur Bearbeitung von PDF-Dokumenten:

```
sudo apt-get install pdftk
Need to get 8,434 kB of archives.
After this operation, 39.8 MB of additional disk space will be used.
for i in 2016_ScannedPage_0*.png ; do convert $i $i.pdf ; done
pdftk 2016_ScannedPage_0*.pdf" cat output 2016_ScannedPagesAll.pdf
```

Erweiterung um unter LibreOffice PDF einzulesen und bearbeiten zu können.

```
$ sudo apt-get install libreoffice-pdfimport
0 upgraded, 1 newly installed, 0 to remove and 8 not upgraded.
Need to get 255 kB of archives.
After this operation, 524 kB of additional disk space will be used.
```

Abiword war aus irgendeinem Grund nach mehreren updates und upgrades nach einem halben Jahr verschwunden und wurde wieder installiert.

```
$ sudo apt-get install abiword
0 upgraded, 25 newly installed, 0 to remove and 8 not upgraded.
Need to get 10.0 MB/10.1 MB of archives.
After this operation, 44.6 MB of additional disk space will be used.
```

LibreOffice:

Vom Extensions-Center LibreOffice wurden verschiedene Erweiterungen installiert.

Unter tools > Extension Manager können Erweiterungen ergänzt werden. Ausgewählt und Heruntergeladen in ein Verzeichnis können diese unter folgenden Link:

<http://extensions.libreoffice.org/extension-center>

Ausgewählt wurde:

- TexMaths, a LaTeX equation editor for LibreOffice
- MultiDiff, issue LibreOffice documents in several formats. eLAIX converts your Writer documents into ILIAS learning modules and ebooks (ePUB 3).
- Writer2xhtml is a collection of export filters to EPUB, HTML5, XHTML and XHTML+MathML
- Writer2LaTeX provides Writer export filters for LaTeX and BibTeX.

Es gab hier noch ein Problem mit der Java-Installation, siehe unter Trouble/Problemlösungen.

6.9 Bildbearbeitung

```
sudo apt-get install imagemagick
Need to get 668 kB of archives.
```

After this operation, 799 kB of additional disk space will be used.

Die Konvertierung von Bildern auf der Konsole werden mit dem Befehl „convert“ aufgerufen.

```
convert $fname.tiff $fname.jpg ;
convert -quality 30 -resize 70% $fname.tiff "$fname"_min.jpg
convert -quality 30 $fname.tiff "$fname"_min.jpg
convert "$fname"_min.jpg "$fname"_min.pdf
```

6.10 Mathematische Anwendungen

Ein Mathematikprogramm, das auch Aufgaben mit Variablen lösen kann.

```
sudo apt-get install wxmaxima
Need to get 11.1 MB of archives.
After this operation, 52.8 MB of additional disk space will be used.
```

Es wird hierbei auch die Anwendung gnuplot mit installiert, die graphische Diagramme auf dem Bildschirm ausgeben kann.

6.11 Virens Scanner

Vor der Installation des Virens Scanners wurde von einem anderen Rechner der Inhalt des ganzen Verzeichnis mit den clamav-Virensignaturdateien auf den Rechner kopiert.

```
/var/lib/clamav/
sudo apt-get install clamav
Need to get 1,946 kB of archives.
After this operation, 3,303 kB of additional disk space will be used.
```

In der Nacht würden in der Regel erst die Virensignaturen ca. 200MB geladen (6-7h bei 64kbit/s).

Die zukünftigen Delta-Dateien sind deutlich kleiner. Details sind zu lesen unter

```
"cat /var/log/clamav/freshclam.log"
"ls /var/lib/clamav"
ls -l /var/lib/clamav
  378368 Apr 19 06:17 bytecode.cld
  6678016 Apr 23 05:14 daily.cld
 109143933 Apr 19 06:19 main.cvd
    4992 Apr 19 06:19 mirrors.dat
```

6.12 Laufzeitumgebungen und Virtualisierung

Die Umgebung „wine“ bietet eine einfache Möglichkeit einige Windows-Programme unter Linux zum Laufen zu bringen, ohne dass ein Windows installiert werden muss. Hierbei handelt es sich um nachprogrammierte Laufzeitumgebungen für die Windows-Programme. Zumeist sind dies benötigte dll-Dateien und deren Schnittstellen für die Windows-Anwendungen.

Es lohnt sich auch die vorgeschlagenen Pakete anzusehen, ob diese sinnvollerweise auch gleich mit installiert werden sollten.

```
$ sudo apt-get install wine
```

Suggested packages:

```
ttf-mscorefonts-installer opencl-icd winetricks playonlinux wine-binfmt dosbox wine32-preloader
Need to get 15.7 MB of archives.
After this operation, 131 MB of additional disk space will be used.
```

Ein Test ergibt hier folgende Informationen:

```
$ sudo apt-get install wine winetricks dosbox playonlinux
```

Need to get 25.4 MB of archives.

After this operation, 169 MB of additional disk space will be used.

Ausgewählt für die Installation wurde folgende Kombination:

```
$ sudo apt-get install wine winetricks dosbox
```

Need to get 17.1 MB of archives.

After this operation, 136 MB of additional disk space will be used.

7 Varianten zum Vorgehen bei der Installation

7.1 Verwenden von bereits heruntergeladenen Paketdateien

Eine gute Idee ist es mittels "sudo dpkg -l > installed_list_1501xx.txt" abzulegen, welche Pakete zu Beginn installiert waren oder aktuell installiert sind. Diese Datei kann später auch genutzt werden, um mit Hilfe von Skripten (die "dpkg" benutzen) das System mit allen Anwendungen wieder aufzusetzen.

Die heruntergeladenen Pakete befinden sich alle unter dem Pfad "/var/cache/apt/archives/", sofern dieses nicht mit Optionen von "apt-get, wie zum Beispiel clean/autoclean/autoremove aufgeräumt wurde. Also sollte man sich dieses Archiv auch in den entsprechenden Pfad beim aufzusetzenden System kopieren. Im Weiteren müssen somit die Dateien auch nicht mehr vom Internet geholt werden von apt-get mit "upgrade" oder "install".

Unter "/var/lib/apt/lists" liegen die Informationen zu den Pakten. Dieses sollte auch kopiert werden, wenn mit "apt-get" ohne Internetverbindung installiert werden soll.

Ein reines lokales Verzeichnis (z.B. USB-Stick voll mit *.deb Dateien) für apt-get einbinden:

```
mkdir MeinRepository
```

```
cd MeinRepository
```

```
dpkg-scanpackages binary /dev/null | gzip -9c > Packages.gz
```

Folgende Zeile in /etc/apt/sources.list einfüegen:

```
deb file:///pfad/zu/MeinRepository ./
```

Anschließend apt-get update ausführen, damit der Installer die Dateien findet und nutzen kann.

Der Versuch funktioniert aber nicht.

```
sudo dpkg-scanpackages binary /dev/null | gzip -9c > Packages.gz
```

```
bash: Packages.gz: Permission denied
```

```
dpkg-scanpackages: error: binary dir binary not found
```

Aber so funktioniert es mit ein paar Fehlermeldungen.

```
sudo $(dpkg-scanpackages ./ /dev/null | gzip -9c > ~/Packages.gz)
```

Und anschließend manuell verschieben der Datei in das Verzeichnis.

7.2 Das System auf eine größere SD-Karte übertragen

Schneller geht es, indem eine Micro-CD-Karte in den SD-Adapter gesteckt wird, die andere in einen USB-Adapter und beide am Netbook angeschlossen werden. Kopiert wird mittels "dd". Hierbei muss besonders aufgepaßt werden, dass man sich nicht die falsche Karte zerschießt oder sogar die HD vom Netbook löscht. Man beachte "of=" gibt das zu schreibende Zielmedium an. Etwas ungünstig ist hier, dass beim den Mini-Einplatinen-Rechnern über USB angeschlossene Speichermedien als „/dev/sdax“ im System erscheinen und auf den Notebooks ist das aber die eingebaute Festplatte mit dem Betriebssystem und der Recovery-Partition. Beim Übernehmen von Anleitungen aus dem Internet, ist daher besondere Vorsicht geboten.

Um das System auf eine größere SD-Karte zu übertragen, sind folgende Schritte notwendig:

1. SD-Karte vorbereiten

Falls Daten auf der SD-Karte sind, diese unbedingt vorher sichern.

SD-Karte ≤ 32GB und SD-Karte > 32GB kein Unterschied, da diese bei den folgenden Schritten ganz platt gemacht werden.

2. Kopieren des Images oder Inhaltes von der kleinen auf die größere SD-Karte

Unbedingt mit „lsblk“ prüfen, als welches Device /dev/sxx die SD-Karten eingebunden wurden.

Hierbei sei /dev/sdb die kleinere SD-Karte und /dev/sdc die größere SD-Karte.

```
dd if=/dev/sdb of=/dev/sdc
```

Bei einer Schreibgeschwindigkeit von ca. 3 bis 4 MB/s dauerte das ganze ungefähr 20-30 Minuten.

3. Partitionen verschieben und vergrößern

Die neue (größere) SD-Karte muss anschließend noch mit "gparted" bearbeitet werden.

Vorher sollte mit „sudo fdisk -l /dev/sdc“ die aktuelle Partition begutachtet werden.

Es sollte dabei die Partition mit dem Betriebssystem vergrößert werden und wenn nicht bereits vorhanden, empfiehlt es sich wieder eine Partition für den Nutzer anzulegen. Es bleibt dabei den Vorlieben des Einzelnen überlassen, ob er es bevorzugt alle Partitionen als primäre Partitionen anzulegen oder ob er eine erweiterte Partition mit logischen Partitionen bevorzugt. Als letzte Partition am Speicherende der SD-Karte sollte wieder eine Swap-Partition eingerichtet werden.

Die fstab muss angepaßt werden, wenn die Swap-Partition nun nicht mehr die dritte sondern die vierte Partition wurde oder UUID-Angaben in der fstab verwendet wurden.

```
sudo cat /etc/fstab
# UNCONFIGURED FSTAB FOR BASE SYSTEM
tmpfs /tmp tmpfs nodev,nosuid,size=256M 0 0
/dev/mmcblk0p1 / ext4 defaults,noatime,nodiratime,data=writeback,commit=600,errors=remount-ro 0 0
/var/swap none swap sw,pri=1 0 0
/dev/mmcblk0p6 none swap sw,pri=5 0 0
```

Aus den Angaben kann entnommen werden, wo die jeweiligen Partitionen zu finden sind. Da diese in der fstab aufgenommen wurde, wird diese automatisch beim Start eingebunden.

Für den Nutzer wird noch ein beschreibbares Verzeichnis angelegt. Darunter kann dieser nun auch Dateien ablegen.

```
sudo mkdir /media/seem/data
sudo chmod 777 /media/seem/data
```

7.3 Updaten des Betriebssystems

Vor dem Updaten sollte unbedingt nachgesehen werden, wie viel Platz noch auf der SD-Karte vorhanden ist. Zum Beispiel Konsolenbefehl „df“

1. Updaten der Paketquellen: `sudo apt-get update`
2. Updaten der Pakete: `sudo apt-get upgrade`

Auf eine neue Version von Armbian umsteigen, empfehle ich dies auf einer richtigen Konsole (mit Strg+Alt+F1...F4) und nicht unter der graphischen Oberfläche in einem dort geöffneten Terminalfenster (xterm) durchzuführen. Manchmal werden bei der Installation Abfragen zur Auswahl von Konfigurationen angeboten (editieren, behalten oder des „maintainers“ verwenden), die einige praktischen Einstellungen überschreiben können. Zum Beispiel, dass Befehle nur noch als root ausgeführt werden können, aber mit sudo nicht, da Berechtigungen geändert wurden.

1. `sudo apt-get dist-upgrade`

7.3.1 Ungewolltes Update und Änderung des Zustandes

Am Morgen des 30.04.16 lädt der Rechner sehr lange etwas aus dem Internet. Mit Etherape wurde die IP 212.211.132.250 und wieck.debian.org ermittelt. Die Spur führt nach <https://db.debian.org/machines.cgi?host=lobos>

Insgesamt wurde etwa 37MB an updates/upgrades heruntergeladen.

Die Ausgabe von „ps ax“ ergab:

```
12449 ?    SN    0:35 /usr/bin/python3 /usr/bin/unattended-upgrade
```

```
12455 ?    SN    0:05 /usr/lib/apt/methods/http
```

Es scheint also etwas mit apt-get im Hintergrund zu laufen. Es erfolgte ein Test mit apt-get:

```
$ sudo apt-get install libreoffice
```

```
E: Could not get lock /var/lib/dpkg/lock - open (11: Resource temporarily unavailable)
```

```
E: Unable to lock the administration directory (/var/lib/dpkg/), is another process using it?
```

Also ist apt-get mit dpkg im Hintergrund aktiv.

Unter /etc/cron.daily/apt befindet sich wohl der Hinweis:

```
# APT::Periodic::Unattended-Upgrade "0";
```

```
# - Run the "unattended-upgrade" security upgrade script
```

```
# every n-days (0=disabled)
```

```
# Requires the package "unattended-upgrades" and will write
```

```
# a log in /var/log/unattended-upgrades
```

In der /etc/apt/apt.conf.d/02periodic Sind die Variablen für das Verhalten gesetzt.

```
// Enable the update/upgrade script (0=disable)
```

```
APT::Periodic::Enable "1";
```

```
// Do "apt-get update" automatically every n-days (0=disable)
APT::Periodic::Update-Package-Lists "1";

// Do "apt-get upgrade --download-only" every n-days (0=disable)
APT::Periodic::Download-Upgradeable-Packages "1";

// Run the "unattended-upgrade" security upgrade script
// every n-days (0=disabled)
// Requires the package "unattended-upgrades" and will write
// a log in /var/log/unattended-upgrades
APT::Periodic::Unattended-Upgrade "1";

// Do "apt-get autoclean" every n-days (0=disable)
APT::Periodic::AutocleanInterval "7";
```

Wenn man dies abstellen will, muss man die Werte alle auf 0 setzen. Wenn man dies schön sauber dokumentiert in der Datei ausführen möchte, dann schreibt man sich einen Kommentar entsprechend dazu, z.B. „// default before manual change was 7 and now 0 (date 01.05.16)“.

```
// default before manual change was 1 and now 1 (date 01.05.16)
APT::Periodic::Enable "1";
// default before manual change was 1 and now 0 (date 01.05.16)
APT::Periodic::Update-Package-Lists "0";
// default before manual change was 1 and now 1 (date 01.05.16)
APT::Periodic::Download-Upgradeable-Packages "1";
// default before manual change was 1 and now 0 (date 01.05.16)
APT::Periodic::Unattended-Upgrade "0";
APT::Periodic::AutocleanInterval "0";
// default before manual change was 7 and now 0 (date 01.05.16)
```

In dieser Konfiguration werden die Dateien für das update/upgrade nur heruntergeladen, aber nicht installiert.

Im Zeitfenster Juni/Juli 2016 war alles überschrieben mit den alten Werten. Dies wurde am 28.07.16 wieder zurück geändert.

Ein apt-get update zu Beginn der vierten Augustwoche in 2016 ergab die Ausgabe, dass über 200MB heruntergeladen werden mußte. Dies führe ich daher nicht durch und wartete noch. Ab der Wochenmitte war die Ausgabe nach weiteren apt-get update, wieder normal mit 19MB herunterzuladen und zu installieren.

7.3.2 Trouble-Shooting Zeitangabe

Nach dem ich eine Woche weg war, hatte die Uhr natürlich eine vollkommen falsche Zeit und Datum. Es funktionierte keine Internetseite und auch die Google-Suche verweigerte den Dienst

wegen eines Zertifikatfehlers. Die Zeit stellte sich auch nicht mehr von selbst ein durch Übernahme aus dem Zeitserver im Internet. Abhilfe war nur möglich, indem die Uhr auf das richtige Datum gestellt wurden, aber die Zeit (Stunden, Minuten) ließ ich bewußt falsch. Erst mit richtigem Datum konnte sich der Rechner aus dem Netz die richtige Zeit nach wenigen Minuten holen.

Schuld daran sind die Zugriffe über „https“ die immer ein Zertifikat benötigen. Reine Zugriffe über „http“ haben das Problem nicht. Wenn der Rechner am Internet hängen soll um Dienste ungewartet auszuführen, gibt es Probleme, wenn dieser mehrere Tage außer Betrieb sein sollte.

7.3.3 Trouble-Shooting WLAN geht nicht mehr

Anfang Juli 2016 wurde noch einmal der ganze Prozess „sudo apt-get update“ und „sudo apt-get upgrade“ durchgelaufen. Gute zwei Wochen später wurde das System neu gestartet und das WLAN funktionierte nicht mehr. Mittel „sudo ifconfig“ wurde nur „eth0“ und „lo“ (loopback) angegeben.

Die folgenden Kommandos brachten auch keine Fehlerbehebung:

```
sudo wpa_action wlan0 restart
```

```
sudo /etc/init.d/networking restart
```

Ein Blick in die Datei /etc/network/interfaces brachte etwas Licht in die Dunkelheit. Die Datei war bei dem Vorgang des „Upgrades“ ausgetauscht worden und somit waren die dortigen Eintragungen weg. Bei der neuen Datei wird die Eingabe des Passwortes als verschlüsselte Zeichenkette verlangt.

```
$ cat /etc/network/interfaces
# Wired adapter #1
allow-hotplug eth0
#no-auto-down eth0
iface eth0 inet dhcp
#       hwaddress ether # if you want to set MAC manually
#       pre-up /sbin/ifconfig eth0 mtu 3838 # setting MTU for DHCP, static just: mtu 3838

# Wired adapter #2
#allow-hotplug eth1
#iface eth1 inet dhcp
#       hwaddress ether # if you want to set MAC manually
#       pre-up /sbin/ifconfig eth0 mtu 3838 # setting MTU for DHCP, static just: mtu 3838

# Wireless adapter #1
#allow-hotplug wlan0
#iface wlan0 inet dhcp
#       wpa-ssid SSID
#       wpa-psk xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
# to generate proper encrypted key: wpa_passphrase yourSSID yourpassword

# Local loopback
auto lo
iface lo inet loopback
```

Nur Eintragen des WLAN und sudo /etc/init.d/networking restart brachte noch keinen Erfolg. Es fällt auf, dass beim Booten nicht mehr die Wartezeit von etwas über einer Minute bis zum Starten der Netzwerke anfällt.

Es scheint, dass die Umstellung etwas mit den neuen Regelung zu den WLAN-Chips zu tun haben könnten, so dass diese nicht mehr so frei wie bisher angesteuert werden können für verschiedene Verwendungen als Accesspoint ab Juli 2016. Zum Ziel führte eine geänderte Konfigurationsdatei und ein Ausführen von sudo apt-get update und sudo apt-get upgrade bei dem auch die Kernels auf eine neue Version angehoben wurden.

```
$ cat /etc/network/interfaces
# Wired adapter #1
# allow-hotplug eth0
#no-auto-down eth0
auto eth0
```

```

iface eth0 inet dhcp
#       hwaddress ether # if you want to set MAC manually
#       pre-up /sbin/ifconfig eth0 mtu 3838 # setting MTU for DHCP, static just: mtu 3838

# Wired adapter #2
#allow-hotplug eth1
#iface eth1 inet dhcp
#       hwaddress ether # if you want to set MAC manually
#       pre-up /sbin/ifconfig eth0 mtu 3838 # setting MTU for DHCP, static just: mtu 3838

# Wireless adapter #1
#allow-hotplug wlan0
#iface wlan0 inet dhcp
#       wpa-ssid SSID
#       wpa-psk xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
# to generate proper encrypted key: wpa_passphrase yourSSID yourpassword

auto wlan0
iface wlan0 inet dhcp
       wpa-ssid SSID
       wpa-psk xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

# Local loopback
auto lo
iface lo inet loopback

```

Hierbei wurde eth0 auf auto geändert, da sonst das Gerät nicht automatisch bei einem Start am LAN in dieses über LAN eingeloggt werden kann um am LAN eines Routers die aufgeführten Befehle mit apt-get auszuführen.

Mit diesen Eintragungen kann das WLAN nun manuell gestartet werden.

```
$ sudo ifup wlan0
```

```
$ sudo dhclient wlan0
```

```
$ ip link set wlan0 up
```

Letzterer Befehl hilft, wenn die Internetanwendung statt auf wlan0 immer noch auf eth0 zugreifen sollte.

Oder Alternativ manuell über

```
$ sudo wpa_supplicant -Dnl80211 -iwlan0 -c/etc/wpa_supplicant.conf
```

```
$ cat /etc/wpa_supplicant.conf
```

```

# allow frontend (e.g., wpa_cli) to be used by all users in 'wheel' group
ctrl_interface=/var/run/wpa_supplicant
#ctrl_interface_group=wheel
#
# home network; allow all valid ciphers
network={
    ssid="MeinAP"
    scan_ssid=1
    key_mgmt=WPA-PSK
    psk="myPhrase"
}
#

```

```
$ sudo dhclient wlan0
```

Damit das WLAN des CubieTruck seine IP-Adresse vom DHCP-Server erhält und übernimmt.

```
$ ip link set wlan0 up
```

Letzterer Befehl hilft, wenn die Internetanwendung statt auf wlan0 immer noch auf eth0 zugreifen sollte.

(Für weitere Details siehe zum Beispiel die Quelle:

https://wiki.debian.org/WiFi/HowToUse#wpa_supplicant)

7.3.4 Trouble apt-get update

```
Get:213 http://httpredir.debian.org jessie/main Translation-en [4,580 kB]
Get:214 http://httpredir.debian.org jessie/non-free Translation-en [72.5 kB]
Fetched 19.5 MB in 1h 3min 44s (5,110 B/s)
E: Could not get lock /var/lib/dpkg/lock - open (11: Resource temporarily unavailable)
E: Unable to lock the administration directory (/var/lib/dpkg/), is another process using it?
dieterd@cubietruck:~$ sudo apt-get update
[sudo] password for dieterd:
Hit http://apt.armbian.com jessie InRelease
Hit http://apt.armbian.com jessie/main armhf Packages
```

vnstat zeigt volle Datenrate an über die GSM-Verbindung.

```
ps ax | less
31302 ?    S    0:00 /usr/sbin/CRON -f
31306 ?    Ss   0:00 /bin/sh -c test -x /usr/sbin/anacron || ( cd / && run-parts --report
/etc/cron.daily )
31307 ?    S    0:00 run-parts --report /etc/cron.daily
31308 ?    S    0:00 /bin/sh /etc/cron.daily/apt
31365 ?    S    0:00 [kworker/1:2]
31398 ?    SN   1:47 /usr/bin/python3 /usr/bin/unattended-upgrade
31404 ?    SN   0:06 /usr/lib/apt/methods/http
```

Die `/etc/apt/apt.conf.d/periodic` ist ebenfalls verändert und die Änderungen nicht mehr enthalten. Im Zeitfenster Juni/Juli 2016 war alles überschrieben mit den alten Werten. Dies wurde am 28.07.16 wieder zurück geändert. Siehe unter "Ungewolltes Update und Änderung des Zustandes".

7.4 Trouble LibreOffice Erweiterungen und Java-Installation

LibreOffice:

Vom Extensions-Center LibreOffice wurden verschiedene Erweiterungen installiert.

Unter tools > Extension Manager können Erweiterungen ergänzt werden. Ausgewählt und Heruntergeladen in ein Verzeichnis können diese unter folgenden Link:

<http://extensions.libreoffice.org/extension-center>

Ausgewählt wurde:

- TexMaths, a LaTeX equation editor for LibreOffice
- MultiDiff, issue LibreOffice documents in several formats. eLAIIX converts your Writer documents into ILIAS learning modules and ebooks (ePUB 3).
- Writer2xhtml is a collection of export filters to EPUB, HTML5, XHTML and XHTML+MathML
- Writer2LaTeX provides Writer export filters for LaTeX and BibTeX.

Bei den letzteren zwei gab es eine Fehlermeldung "could not locate java implementation loader.

Gemäß eine Hilfestellung im Internet sei diese Einstellung zu finden unter dem Menu über tools > options > (im Fenster als Auswahl) LibreOffice > Advanced. Hier würde die zu verwendende Java-Installation auf dem Rechner ausgewählt. Bei der Installation LibreOffice unter Cubian war dieses Feld leer. Das heißt es war gar kein java vorhanden. Auf der Konsole ergab java eingetippt auch Fehlanzeige.

Der erkannte Problem war, dass kein java installiert war auf dem System.

Nachdem "sudo apt-get install java" nichts anzeigte wurde mit "sudo apt-cache search java" gesucht. Die Ausgabe war sehr lang mit wenig passenden Inhalten. Die Suche ergab, dass die Pakete gcj und openjdk eine Lösung bieten könnten.

Nach Durchsicht entsprechender Seite von Ubuntu zu java wurde openjdk als Paket ausgewählt.
<https://wiki.ubuntuusers.de/Java/>

Die Suche ergab, dass es die Versionen openjdk-7-jre und openjdk-8-jre gibt.

```
$ sudo apt-get install openjdk-
```

```
openjdk-7-dbg      openjdk-7-jre-headless openjdk-8-doc      openjdk-8-jre-jamvm
openjdk-7-demo    openjdk-7-jre-lib      openjdk-8-jdk      openjdk-8-source
openjdk-7-doc     openjdk-7-source       openjdk-8-jdk-headless
openjdk-7-jdk     openjdk-8-dbg          openjdk-8-jre
openjdk-7-jre     openjdk-8-demo         openjdk-8-jre-headless
```

```
dieterd@cubietruck:~$ sudo apt-get install openjdk-8-jre
```

```
Some packages could not be installed. This may mean that you have
requested an impossible situation or if you are using the unstable
distribution that some required packages have not yet been created
or been moved out of Incoming.
```

```
The following information may help to resolve the situation:
```

```
The following packages have unmet dependencies:
```

```
openjdk-8-jre : Depends: openjdk-8-jre-headless (= 8u102-b14.1-2) but it is not going to be
installed
```

```
E: Unable to correct problems, you have held broken packages.
```

```
$ sudo apt-get install openjdk-7-jre
```

```
The following packages have unmet dependencies:
```

```
openjdk-7-jre : Depends: openjdk-7-jre-headless (= 7u111-2.6.7-1~deb8u1) but it is not going to be
installed
```

```
E: Unable to correct problems, you have held broken packages.
```

Problem:

```
openjdk-8-jre : Depends: openjdk-8-jre-headless (= 8u102-b14.1-2) but it is not going to be
installed
```

Der erkannte Problem war, dass es hier ein Problem mit den Abhängigkeiten gibt. Solches passiert leider immer wieder. Oft ist der technische Hintergrund, dass nach "update" und "upgrade" Pakete auf eine höhere Version hochgezogen werden, da ein Paket, dessen Abhängigkeiten dies automatisch zurückstellen würde, bisher noch nicht installiert war.

Eine Hilfestellung aus dem Internet empfahl "update" und "upgrade" noch einmal durchlaufen zu lassen. Danach war die Situation weiterhin unverändert.

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo dpkg --configure -a
```

Also gibt es nur die Möglichkeit in die Tiefe zu gehen des Installationsprozesses.

```
$ sudo apt-get -d install openjdk-8-jre
The following packages have unmet dependencies:
 openjdk-8-jre : Depends: openjdk-8-jre-headless (= 8u102-b14.1-2) but it is not going to be
installed
$ sudo apt-get -d install openjdk-8-jre-headless
The following packages have unmet dependencies:
 openjdk-8-jre-headless : Depends: util-linux (>= 2.26.2-4) but 2.25.2-6 is to be installed
$ sudo apt-get -d install util-linux
util-linux is already the newest version.
```

Eine Hilfestellung aus dem Internet empfahl "update" und "upgrade" noch einmal durchlaufen zu lassen. Danach war die Situation weiterhin unverändert.

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo dpkg -configure -a
```

```
$ sudo apt-get -d install util-linux
util-linux is already the newest version.
```

Es gibt auch die Möglichkeit verschiedene Distro-Versionen als Vorgabe für apt-get mitzugeben.

```
$ sudo apt-get -d install openjdk-8-jre-headless/jessie
E: Release 'jessie' for 'openjdk-8-jre-headless' was not found
$ sudo apt-get -u install openjdk-8-jre-headless/jessie
E: Release 'jessie' for 'openjdk-8-jre-headless' was not found
dieterd@cubietruck:~$ sudo apt-get -u install openjdk-8-jre-headless/wheezy
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Release 'wheezy' for 'openjdk-8-jre-headless' was not found
dieterd@cubietruck:~$ sudo apt-get install openjdk-8-jre-headless/wheezy
E: Release 'wheezy' for 'openjdk-8-jre-headless' was not found
$ sudo apt-get install openjdk-8-jre-headless/stretch
The following packages have unmet dependencies:
 openjdk-8-jre-headless : Depends: util-linux (>= 2.26.2-4) but it is not going to be installed
```

Ergebnis dieser Versuche ist, dass es sich hier um eine Version handelt, die nicht bei Debian Jessie vorhanden war.

```
dieterd@cubietruck:~$ sudo apt-get -u install openjdk-7-jre-headless/jessie
```

The following packages have unmet dependencies:

openjdk-7-jre-headless : Depends: tzdata-java but it is not going to be installed

E: Unable to correct problems, you have held broken packages.

dieterd@cubietruck:~\$ sudo apt-get install tzdata-java

The following packages have unmet dependencies:

tzdata-java : Depends: tzdata (= 2016d-0+deb8u1) but 2016f-1 is to be installed

dieterd@cubietruck:~\$ sudo apt-get install tzdata

tzdata is already the newest version.

dieterd@cubietruck:~\$ sudo apt-get install tzdata-java/jessie

The following NEW packages will be installed:

tzdata-java

The following packages will be DOWNGRADED:

tzdata

0 upgraded, 1 newly installed, 1 downgraded, 0 to remove and 8 not upgraded.

\$ sudo apt-get install openjdk-7-jre-headless/jessie

Selected version '7u111-2.6.7-1~deb8u1' (Debian-Security:8/stable [armhf]) for 'openjdk-7-jre-headless'

0 upgraded, 4 newly installed, 0 to remove and 8 not upgraded.

Need to get 37.7 MB of archives.

After this operation, 45.1 MB of additional disk space will be used.

Was ist der Unterschied von java zu java headless?

Headless is the same version than the latter without the support of keyboard, mouse and display systems. Hence it has less dependencies and it makes it more suitable for server application.

<https://www.quora.com/What-is-the-difference-between-openjdk-7-jre-headless-and-openjdk-7-jre-jdk>

Nachdem die "headless"-Variante installiert werden konnte, wurde die nächste vollständigere Variante installiert.

\$ sudo apt-get install openjdk-7-jre/jessie

0 upgraded, 3 newly installed, 0 to remove and 8 not upgraded.

Need to get 207 kB of archives.

After this operation, 511 kB of additional disk space will be used.

\$ sudo apt-get install openjdk-7-jdk/jessie

0 upgraded, 1 newly installed, 0 to remove and 8 not upgraded.

Need to get 13.7 MB of archives.

After this operation, 17.6 MB of additional disk space will be used.

Unter den Optionen konnte das Java nun ausgewählt werden. Danach ist ein Neustart von LibreOffice erforderlich. Jetzt konnten unter dem Extension Manager diese java-abhängigen Erweiterungen aktiviert werden. Auf Grund der Erweiterungen veränderte sich die untere Menüleiste. Unter view > Toolbars wurden bei eLAIX und TexMaths die Häkchen entfernt und somit war wieder alles in Ordnung. In der obersten Zeile gibt es nun eine Auswahl eLAIX mehr.

8 Aspekte der IT-Sicherheit

8.1 Zugang beim Starten des Systems

Das Betriebssystem Raspian ist für die erste Verwendung natürlich besonders nutzerfreundlich ausgelegt. Somit werden keine Nutzernamen und Paßwörter für das Starten erforderlich und der Startvorgang endet direkt in die graphische Desktop-Umgebung. Ein Paßwort wird nur für den ersten Zugang zum jeweiligen geschützten WLAN-Netz benötigt.

Von Seiten der IT-Sicherheit ist das eigentlich eine Katastrophe. Als reines System für Bastler zum Probieren und Experimentieren, kann dies in gewissen Maßen noch toleriert werden, für die ursprüngliche Zielgruppe und deren Verwendung. Wer mehr mit dem System macht, wird sicherlich unvermeidlich so tief in der Materie stecken, so dass er die notwendigen Maßnahmen selbst ergreifen wird sein System abzusichern.

Im Gegensatz hierzu erzwingt das Armbian für den CubieTruck beim ersten Start eine Vergabe von Passwörtern für den Administrator „root“ und für den anzulegenden Nutzer „user“. Dabei werden einfache Passwörter nicht zugelassen.

8.2 Einloggen über das Netzwerk

Wenn auf Rechner über das allgemein bekannte Passwort und Zugangsnamen „remote“ zugegriffen wird im Netzwerk, stellt dies eine große Sicherheitslücke dar, vor allem bei der Nutzung in WLAN-Netzen. Im lokalen Heimnetzwerk hinter einem Router ist zwar schon eine schützende Firewall zur Außenwelt vorhanden, aber dieser Schutz kann auch überwunden werden. Häufige Schwachstelle sind hier andere Rechner im Netz, die vielleicht schon gekapert wurden. Relativ sicher am Internet hängt ein CubieTruck oder Raspi, wenn noch ein alter Router vorhanden ist, dem von Seiten des DSL-Modemeinganges wirklich keine Zugriffe jeglicher Art möglich sind. Bei modernen Routern mit Voice over IP-Funktionen, USB-Anschlüssen für Speicher und Drucker ist das leider nicht mehr so.

In dem Falle sollte für „ssh“ und „sftp“ für gleiche IP und unterschiedliche Systeme bei Verwendung mehrerer SD-Karten entsprechende „UserKnownHostsFile“ verwendet werden so dass „StrictHostKeyChecking“ immer angewendet wird.

Die allgemeinen Passwörter sollten geändert werden so weit dies überhaupt möglich ist. Bei Sonderfällen, wie OSMC kann ein weiterer Nutzer mit einem sichereren Passwort angelegt werden und für den Standardnutzer alles was unter „remote“ fällt gesperrt werden.

Beim Armbian erlaubt es die Standardeinstellung, dass man sich direkt als root über das Netzwerk einloggen kann. Bei anderen Systemen ist dies oft nicht möglich, man sich zuerst als Nutzer erfolgreich einloggen muss und danach erst in den Administratormode root wechseln kann durch einloggen.

8.3 Erste einfache Maßnahmen zur Absicherung

Die Datei "sudoers" wird in der Regel nicht angefaßt werden müssen, um zu verhindern, dass man ohne Passwort root werden kann im Gegensatz zum Raspian. Falls eine Anwendung (Raspian des Raspberry Pi) unbedingt unter dem Nutzernamen "pi" laufen muss, wäre eine Maßnahme für diesen Nutzer den Port für den Empfang von ssh-Zugriffen zu schließen oder und zugleich das Paßwort zu ändern. Ist diese nicht notwendig sollte gleich einen Nutzer mit einem nicht so leicht zu knackenden Paßwort angelegt werden.

Einen Startbildschirm mit Einloggmenü und Paßwortabfrage sollte auch installiert werden. Somit erhält man ein System, das schon mal die wichtigsten Hürden beinhaltet.

8.4 Und noch eine kleine Besonderheit des CubieTruck

Der kleine CubieTruck hat zwar kein BIOS, UEFI oder sonstige Teile, die geflasht werden können. Das hat auf der einen Seite den Vorteil, dass ein internes Android auf dem System vorhanden ist und eine beschränkte Bootmedienauswahl (SATA oder Android) festgelegt werden kann. Dafür hat dieser aber den Nachteil, dass wenn er mal gehackt wurde, dieser nicht wie der Raspi¹ nach der Entnahme des Speichermediums (SD-Karte) wieder als sauber betrachtet werden kann.

In dem Falle muss das System nach Empfang einer frischen bespielten SD-Karte erst der interne Flash (NAND) vollständig geprüft oder beschrieben werden. Erst danach kann dieser wieder als sicher betrachtet werden.

Besser geeignet wäre daher der Raspi zum Löschen von Medien mit kritischen Daten über USB-IDE/SATA-Adapter oder als Paketfilter zwischen sicherheitskritischen Netzwerkverbindungen.

Auf der anderen Platinenseite gegenüber dem SD-Kartenadapter befindet sich der 8GB-Flash Chip. Dieser könnte theoretisch auch entfernt werden. Ob das Gerät ohne diesen Chip noch funktioniert, konnte bisher nicht verifiziert werden. Wenn dies aber möglich wäre, dann wäre hier der CubieTruck gleichwertig mit dem Raspi.

9 Zusammenfassung

Das CubieTruck ist somit konfiguriert für die Verwendung als Mini-Rechner zum Surfen, Bearbeitung von Dokumenten, MediaCenter (Internetradio) und später vielleicht noch mit Kodi. Auf Grund des geringen Stromverbrauchs dürften sich die Kosten der Anschaffung sehr schnell amortisieren.

1 Beim Raspberry Pi Typ 3 können Teile geflasht werden, gemäß einer Beschreibung zum Booten von einer USB-Festplatte. Quelle: <http://www.heise.de/newsticker/meldung/Raspberry-Pi-3-bootet-von-USB-Stick-und-SSD-3288619.html>

10 Ein anderes Linux auf dem CubieTruck

Auf der Suche nach einem Image mit einem Ubuntu-Mate für den CubieTruck im Internet wurde relativ schnell eine Quelle gefunden. Das Image von folgender Internetadresse wurde heruntergeladen:

<https://stefanius.de/tag/docker-sd-card-image-fuer-cubietruck>

CT_CubieSRV_Docker1.8.2_K4.2.1_B151007.img.7z

Die Datei von etwa 230MB komprimiert wurde entpackt und belegte ungefähr 1,8GB auf dem Speichermedium. Mittels „dd“ wurde diese Datei auf eine SD-Karte (32GB) geschrieben.

Anschließend wurde mit dem Werkzeug „gparted“ die zweite Partition mit ungefähr 1,7GB Größe auf 12GB vergrößert. Auf dem noch verbliebenen Speicherbereich wurde eine erweiterte Partition angelegt, die eine weitere große Partition „ext4“ formatiert und eine Swap-Partition enthält.

```
$ sudo fdisk -l /dev/sdb
Disk /dev/sdb: 29 GiB, 31104958464 bytes, 60751872 sectors
Device      Boot      Start         End      Sectors   Size Id Type
/dev/sdb1   *           2048        133119      131072     64M  b W95 FAT32
/dev/sdb2                133120    24709119    24576000    11.7G  83 Linux
/dev/sdb3           24709120    60751871    36042752    17.2G   5 Extended
/dev/sdb5           24711168    54554623    29843456    14.2G  83 Linux
/dev/sdb6           54556672    60751871     6195200       3G  82 Linux swap / Solaris
```

Wenn der CubieTruck (wie alle Arm-Prozessoren) startet, werden die Dateien der ersten Partition verarbeitet und geladen. Im Gegensatz zum Standard-PC besitzt das Board kein BIOS und somit befindet sich alle relevante und board-spezifischen Informationen und Programme für den Start in diesen Dateien.

```
$ ls -l /media/sdb1/
total 5278
-rw----- 1 dieterd dieterd      240 Oct  4  2015 boot.cmd
-rw----- 1 dieterd dieterd      240 Oct  4  2015 boot.cmd.hdd
-rw----- 1 dieterd dieterd      245 Oct  4  2015 boot.cmd.sdcard
-rw----- 1 dieterd dieterd      317 Oct  7  2015 boot.scr
-rw----- 1 dieterd dieterd      312 Oct  5  2015 boot.scr.hdd
-rw----- 1 dieterd dieterd       317 Oct  4  2015 boot.scr.sdcard
-rw----- 1 dieterd dieterd    28110 Oct  4  2015 sun7i-a20-cubietruck.dtb
-rw----- 1 dieterd dieterd     6648 Oct  4  2015 sun7i-a20-cubietruck.dts
-rw----- 1 dieterd dieterd 5366168 Oct  4  2015 uImage
```

In der zweiten Partition befindet sich das eigentliche Betriebssystem, hier ein Ubuntu-Mate mit den Erweiterungsanwendungen Dockers.

```
$ ls /media/sdb2/
bin  dev  home  lost+found  mnt  proc  run  srv  tmp  var
boot  etc  lib  media      opt  root  sbin  sys  usr  webmin-setup.out

$ ls /media/sdb2/home/
linaro
```

Für den ersten Bootvorgang sollte man sich bereits das Paßwort auf der Seite gemerkt (oder notiert parat, usr/pwd, linaro/linaro) haben.

Der erste Start des Systems ging bis auf die Terminal-Ebene. Ein „startx“ war ergebnislos, da dieses Image ohne graphische Oberfläche zur Verfügung gestellt wurde. Hier sind also noch viele Anwendungen nachzuziehen, um es auch als Desktop verwenden zu können.

Natürlich geht noch kein WLAN, da es erst noch eingerichtet werden muss. Allerdings die Übernahme der Einstellungen im Konfigurationsfile identisch zu übernehmen klappte nicht, da das WLAN hier nicht wlan0 lautet sondern lxcbr0 lautet. Zweite Hürde scheint wohl zu sein, dass man das Paßwort zwingend mit wpa_passphrase verschlüsselt in der Datei eintragen muss. Allerdings wpasupplicant war auf dem System noch nicht installiert. Somit kann dieses Images nicht sofort über ein WPA-verschlüsseltes WLAN mittels „apt-get install ...“ hochgezogen werden. Es ist also notwendig den Minirechner mit dem installierten Image auf der SD-Karte über ein LAN-Kabel an einen Router oder Rechner, der seinen Zugang teilt mit dem Gerät am LAN-Anschluß anzuschließen um weitere Anwendungen nachzuinstallieren, wie zum Beispiel wpasupplicant. Wenn dies erfolgreich war (inklusive Neustart), wäre folgende ähnliche Ausgabe auf dem Terminal zu finden:

```
$ ps ax | grep wpa
  455 ?        Ss          0:00 /sbin/wpa_supplicant -s -B -P /run/wpa_supplicant.wlan0.pid
-i wlan0 -D nl80211,wext -C /run/wpa_supplicant
 1378 pts/2    S+          0:00 grep wpa
```

1 Anlagen – Dokumentteile des Raspian

1.1 Ein OSMC-Image für Berryboot erzeugen aus einem SD-Image für den Raspberry

Das OSMC Image downloaden von der Homepage. Auf einen Linux-Rechner. Falls bei Linux noch nicht vorhanden:

```
sudo apt-get install kpartx
sudo apt-get install squashfs-tools
```

Wie es geht steht unter:

http://www.berryterminal.com/doku.php/berryboot/adding_custom_distributions

In der Regel haben viele Distros für den Raspi im Image zwei Partitionen angelegt.

```
sudo kpartx -av OSMC_TGT_rbp2_20150830.img
add map loop0p1 (252:0): 0 497664 linear /dev/loop0 2048
```

Das Image hat nur eine Partition und muss daher nicht wie unter dem Link beschrieben die Dateien verschoben werden und dann die Partiton "deleted" werden.

Verzeichnis als Mountpoint erstellen:

```
mkdir tmpmount
```

```
sudo mount /dev/mapper/loop0p1 tmpmount
```

Ausgabe mit dem Befehl „mount“ um zu prüfen ob das mounten funktioniert:

```
/dev/sdb1 on /media/USB2_DISK type vfat
(rw,nosuid,nodev,uhelper=hal,uid=1000,shortname=winnt)
/dev/mapper/loop0p1 on /media/USB2_DISK/RASPI2/tmpmount type vfat (rw)
```

Erzeugen des Images:

```
sudo mksquashfs tmpmount/ osmc_converted_image_for_berryboot_ohne_libmoduls.img -comp
lzo -e lib/modules
```

mksquashfs: invalid option

SYNTAX:mksquashfs source1 source2 ... dest [options] [-e list of exclude dirs/files]

Options are

-version print version, licence and copy-right message

....

Mein mksquashfs unterstützt die Option des Komprimierens mit „-comp“ nicht.

```
sudo mksquashfs tmpmount osmc_converted_image_for_berryboot_mit_libmoduls.img
```

Parallel mksquashfs: Using 2 processors

Creating 4.0 filesystem on osmc_converted_image_for_berryboot_mit_libmoduls.img, block size 131072.

```
[=====] 1213/1213  
100%
```

Exportable Squashfs 4.0 filesystem, data block size 131072

compressed data, compressed metadata, compressed fragments
duplicates are removed

Filesystem size 154343.39 Kbytes (150.73 Mbytes)

99.83% of uncompressed filesystem size (154600.12 Kbytes)

Inode table size 320 bytes (0.31 Kbytes)

6.28% of uncompressed inode table size (5094 bytes)

Directory table size 132 bytes (0.13 Kbytes)

85.71% of uncompressed directory table size (154 bytes)

Number of duplicate files found 0

Number of inodes 8

Number of files 7

Number of fragments 1

Number of symbolic links 0

Number of device nodes 0

Number of fifo nodes 0

Number of socket nodes 0

Number of directories 1

Number of ids (unique uids + gids) 1

Number of uids 1

root (0)

Number of gids 1

root (0)

```
sudo mksquashfs tmpmount osmc_converted_image_for_berryboot_ohne_libmoduls.img -e  
lib/modules
```

Parallel mksquashfs: Using 2 processors

Creating 4.0 filesystem on osmc_converted_image_for_berryboot_ohne_libmoduls.img, block size 131072.

```
[=====] 1213/1213  
100%
```

Exportable Squashfs 4.0 filesystem, data block size 131072

compressed data, compressed metadata, compressed fragments
duplicates are removed

Filesystem size 154343.39 Kbytes (150.73 Mbytes)

99.83% of uncompressed filesystem size (154600.12 Kbytes)

Inode table size 320 bytes (0.31 Kbytes)

6.28% of uncompressed inode table size (5094 bytes)

Directory table size 132 bytes (0.13 Kbytes)

85.71% of uncompressed directory table size (154 bytes)

Number of duplicate files found 0

Number of inodes 8

Number of files 7

Number of fragments 1

Number of symbolic links 0

Number of device nodes 0

Number of fifo nodes 0

Number of socket nodes 0

Number of directories 1

Number of ids (unique uids + gids) 1

Number of uids 1

root (0)

Number of gids 1

root (0)

Beide Images haben die gleiche Größe von 150MB. Die Images für den Versuch sind somit erstellt. Diese Datei wird zum Beispiel auf einen USB-Stick geschrieben, dieser zusätzlich angesteckt am Raspi und bei BerryBoot die Installation von dieser weiteren Quelle ausgewählt.

1.2 Installation von Anwendungen und kleinen Werkzeugen

Als nächstes soll das bekannte Mediacenter "kodi" installiert werden mit "sudo apt-get install kodi".

Es sind fast 28MB herunterzuladen. Libraries, Teile von mesa, samba und mysql werden nachinstalliert. Fehler "failed to fetch some archives" und somit "sudo apt-get update" notwendig. Es werden fast 10MB heruntergeladen.

Da hier auch Pakete, die die grafische Ausgabe "xorg" betreffen enthalten waren, wird kodi nicht vor einem Neustart gestartet.

ifconfig zeigt wieviel mit surfen und apt-get seit dem Start verbraucht wurde an Daten-Traffic:

RX bytes:56849052 (54.2 MiB) TX bytes:9282326 (8.8 MiB)

Ein Test mit "sudo apt-get upgrade" zeigt, dass ca. 80MB nachgeladen werden müssten. Bei der Installation der einzelnen Pakete wird dies in Schritten auch nachgeholt, wenn es notwendig ist.

"sudo apt-get install kodi"

Es sind fast 30MB von 32MB herunterzuladen. Libraries, Teile von mesa, samba und mysql werden nachinstalliert.

Insgesamt wäre, wenn "sudo apt-get update" vorher ausgeführt worden wäre, etwas weniger Datenverkehr notwendig gewesen, da nun von einigen Libraries die Updates geholt werden und somit die Vorversion nicht mehr verwendet wird. Beim Herunterladen über eine mobile GSM-Verbindung mit 64kbit wäre nun eine Stunde Zeit für eine Mahlzeit und andere Tätigkeiten.

Nach der Installation sind auf der SD 3,58GB belegt.

RX bytes:98098874 (93.5 MiB) TX bytes:11605135 (11.0 MiB)

"sudo apt-get install firefox"

Das Paket ist nicht im Standard-Raspian enthalten. Statt dessen gibt es aber iceweasel. Das ist eigentlich auch der firefox aber mit kleinen Änderungen auf Grund von Differenzen.

"sudo apt-get install firefox"

Herunterzuladen sind hierfür fast 37MB an archiven. Hier beginnt apt-get mit der Installation ohne nach "y/n" zu fragen zur Fortsetzung.

Die Anwendung kodi wurde gestartet und einige Einstellungen vorgenommen. Das Laden der Anwendungen funktionierte. Mit der Auswahl "Verlassen" fällt man wieder in das Raspian zurück und kann hier weiterarbeiten.

Insgesamt sind nun 4,1GB durch das System auf der SD-Karte belegt.

Interessante Browser auf der Konsolenebene sind w3m, lynx, links und links2.

```
“sudo apt-get install w3m lynx links links2“
```

```
0 upgraded, 6 newly installed, 0 to remove and 37 not upgraded.
```

```
Need to get 5,063 kB of archives.
```

```
After this operation, 11.0 MB of additional disk space will be used.
```

```
Install these packages without verification? [y/N] y
```

Mit "links2 -g -driver help" erfolgt zum Beispiel eine graphische Ausgabe. Hier ist sichergestellt, dass zunächst der gesamte Text angezeigt wird, und dann erst die Bilder nachgeladen werden. Alles wird hier schön einfach untereinander aufgelistet. Mit "links2 -g -driver help" wird aufgelistet, welche Graphik-Treiber vorhanden sind und kann diese auch testen.

```
"sudo apt-get install gnome-paint"
```

Installation eines Programms für die Bildbearbeitung das wenig Ressourcen beansprucht.

Herunterzuladen sind hierfür fast 100kB. Bei dem Programm sind nicht alle Funktionen realisiert, so dass ein anderes Programm ausgesucht werden muss für diese Zweck. Das Paket wird daher wieder entfernt.

```
"sudo apt-get purge gnome-paint"
```

Zuerst wird getestet, welchen Installationsumfang die Pakete haben, wenn diese nachinstalliert werden:

```
sudo apt-get install krita, 112MB herunterladen, 315MB benötigen (wegen KDE-Anteilen);
```

```
sudo apt-get install gimp, 15MB herunterladen, 66 MB benötigen;
```

```
sudo apt-get install shutter, 12MB herunterladen, 50MB benötigen;
```

```
sudo apt-get install pinta, 7MB herunterladen, 28MB benötigen;
```

```
sudo apt-get install inkscape, 19MB herunterladen, 102MB benötigen;
```

Somit wird pinta installiert.

```
sudo apt-get install pinta, 7MB herunterladen, 28MB Speicherplatz benötigen;
```

Pinta hat auch einige Macken, zum Beispiel werden nur kleine Flächenbereiche kopiert.

```
“sudo apt-get install mtpaint”
```

Herunterzuladen sind hierfür 500kB.

```
“sudo apt-get install xfig”
```

Ein kleines vektororientiertes Zeichenprogramm. Herunterzuladen sind 4MB.

Eine gute Übersicht über Grafikprogramme bietet Ubuntu:

<https://wiki.ubuntuusers.de/Grafik/>

Insgesamt sind nun 4,2GB durch das System auf der SD-Karte belegt.

Nach ein paar Tagen wurde “sudo apt-get update” und sudo “apt-get upgrade” durchgeführt. Der Rechner lief 9 Tage durch ohne Störungen.

“sudo apt-get install clamav”

Ein kleines Virenschutzprogramm. Herunterzuladen sind fast 2MB. Das Programm lädt danach die Virensignaturen nach (sichtbar in über etherape mirror.cambrium.nl).

Insgesamt sind nun 4,5GB durch das System auf der SD-Karte belegt.

“sudo apt-get install lsof”

Ein kleines Diagnoseprogramm auf der Konsole, das ausgibt welche Dateien von Prozessen verwendet werden. Herunterzuladen sind fast 0,4MB.

Insgesamt sind nun 4,68GB durch das System auf der SD-Karte belegt.

sudo apt-get install browser-plugin-lightspark

Herunterzuladen waren 10MB;

sudo apt-get purge browser-plugin-lightspark

Wieder gelöscht und einige Plugins für Mozilla/Iceweasel installiert. Nun scheinen youtube-videos zu funktionieren.

“sudo apt-get install vino” Sollte remote X11 ermöglichen.

“sudo apt-get purge vino” Somit später wieder entfernt.

“sudo apt-get install testdisk”

“sudo apt-get upgrade” würde 170MB herunterladen, nach dem “sudo apt-get update” und “sudo apt-get upgrade” wäre es nun 450MB und 97 Pakete. Grund sind die neuen Open Driver 2D/3D-Beschleunigung von Raspian. D.h. so nach und nach eines der Pakete, dass erneuert werden würde aus der Liste installieren lassen.

“sudo apt-get install krita” würde benötigen, dass 112MB heruntergeladen werden.

“sudo apt-get install extremetracer” benötigt, dass 32MB heruntergeladen werden.

Start mit “etracer” funktioniert nicht.

*** etracer error: Couldn't initialize video: Couldn't find matching GLX visual (Resource temporarily unavailable)

Über “sudo apt-get upgrade –print-uris” wurden die Links zu den *.deb-Dateien heruntergeladen und extern über einen Linux-Rechner mittels wget und kleinem Script heruntergeladen. Diese Dateien werden nach /var/cache/apt/archives kopiert. Somit entfällt das Herunterladen der Installationsdateien.

Nach dem Upgrade waren 5,9GB belegt.

“sudo apt-get fdupes”, Werkzeug zum Löschen von doppelten Dateien.

“sudo apt-get krita”, Werkzeug zur Grafikbearbeitung. Dieses stürzt leider beim Öffnen eines Bildes ab.

Nun waren 6,3GB belegt. Nach einem “sudo apt-get autoclean” waren es 6.1GB.

“sudo apt-get purge upower”
(War als manuell installiert markiert)

2 Ab hier Reste von Raspberry Pi

3 Lösung von einzelnen Problemen und interessante Konfigurationen

3.1 Bildschirmprobleme

Alle Moden etc. vom HDTV-Fernseher sind zu finden unter:

<http://elinux.org/RPiconfig#Video>

<https://www.raspberrypi.org/documentation/configuration/raspi-config.md>

Aus dem openelec.tv Forum:

Try adding to config.txt:

```
hdmi_group=1
```

```
hdmi_mode=5  
hdmi_force_mode=1
```

If you have any interference effects then add "config_hdmi_boost=7"

Bei Kodi gibt es noch einige Einstellungen für Video. Es müssen noch Codecs geladen werden (die freien Codecs sind nicht so schnell bei der Umsetzung):

<http://mymediaexperience.com/raspberry-pi-xbmc-with-raspbmc/>

Anbei noch ein paar Befehle:

<https://www.raspberrypi.org/forums/viewtopic.php?f=28&t=95378>

Hintergrund:

1360x768 ist die Standard "small HDTV" Auflösung. Diese wird vermutlich über VGA-Protokoll ausgehandelt. Also muss man versuchen HDMI zu erzwingen.

4 Besondere Netzwerke

Linux Magazin mit vielen besonderen Netzwerken:

<http://www.linux-magazin.de/Ausgaben/2014/07>

Sehr gute Anleitung mit Grundwissen für die Raspi-Netzwerkadministration:

<http://www.administrator.de/wissen/netzwerk-management-server-raspberry-pi-191718.html>

Netzwerkkarten Bonding oder Balancing:

Hier ist der Unterschied erklärt:

<http://speedify.com/features/channel-bonding/>

Bonding: Um einen weiteren LAN-Anschluss (USB to LAN, 100MBit) am Raspi zu nutzen um auf 200MBit zu kommen. Der Router/Switch auf der Gegenseite muss dies unterstützen.

Balancing: Verteilen der Zugriffe auf zwei verschiedene Internetzugänge (Hotspots von Mobiltelefonen). Surfen geht dann schneller, aber Dateidownloads, Radiostreams, Videostreams werden dabei nicht schneller.

Bündeln heißt zwei Leitungen zugleich oder Redundant nutzen zu einem Gerät:

https://wiki.ubuntuusers.de/Netzwerkkarten_b%C3%BCndeln

<https://wiki.debian.org/Bonding>

Balancing oder Bonding konfigurieren:

<https://wiki.archlinux.org/index.php/Netctl>

Load Balancing:

<https://blog.linitx.com/load-balancing-multiple-internet-connections/>

<http://lartc.org/howto/lartc.rpdb.multiple-links.html#AEN298>

http://www.debian-administration.org/article/377/Routering_for_multiple_uplinks

Alphabetical Index

abiword.....	10, 20, 58
Android.....	4, 6f., 19, 21, 70
BIOS.....	16, 45, 70, 72
dirty bit.....	6f.
dpkg.....	
dpkg --configure -a.....	9ff., 47, 67
dpkg-scanpackages.....	60
gparted.....	5f., 15f., 49, 61, 71
Kühlkörper.....	4
Raspberry Pi.....	
Raspberry.....	3
SATA.....	3f., 6, 11, 70f.
SD-Karte.....	3ff., 10f., 13f., 18f., 21, 60f., 71, 78f.
startx.....	7, 9, 17f., 72
UEFI.....	43, 45, 70

