

Anleitung camd3

Inhaltsangabe:

1. Einleitung/Vorwort
2. [Info über die Image-Arten](#) und welche Werkzeuge benötigt werden
3. [Einbau/Update der camd3 ins DBox2-Image](#)
 - 3.1. Einbau der camd3 (Image ist noch ohne camd3)
 - 3.2. Nachdem wir das Binary eingebaut haben kommen nun die Konfigs
 - 3.3. Ein erster Test...
 - 3.3.1 Einbau der camd3_start,stop.pcamd !
 -
 - 3.4. Update einer neueren camd3-Version (Image hatte bereits camd3 an Board)
4. [Bearbeiten eines Nicht-Beschreibbaren Bereiches im Image](#)
 - 4.1. Bearbeiten eines CRamFS-Images mit New-Tux-Flash-Tools unter WinXp
 - 4.2. Bearbeiten eines CRamFS-Images
 - 4.3. Bearbeiten eines Squash-Images
5. [Konfiguration der camd3](#)
 - 5.1. Konfigdateien/Localhost/etc.
 - 5.2. Heimnetz-Sharing/Internet-Sharing
6. [Nachtrag/Credits/Dat](#)

Was ist die camd3 ?

Die camd3 ist ein Smartcard-Emulator für verschlüsselte, digitale TV- und Radiosender. Sie unterstützt aber auch originale Smartcards. Ebenso kann man mit ihr Cardsharing via Heimnetzwerk und via Internet betreiben, und sie unterstützt das Multicam...

Was heisst denn camd3 überhaupt ?

CAM steht für "Conditional Access Module", was übersetzt etwa heisst "Modul für bedingten Zugang"... Das Teil ist im Grunde ein Datenfilter, der sich nur dann angesprochen fühlt, wenn tatsächlich die Sprache an ihm vorbeirauscht, die er auch versteht. Das macht er nur für die Smartcard (aufopferungsvoll), weil die sonst so überfordert wäre... Was nun ein MultiCAM sein soll erklärt sich von selbst dann ;-)) Kann halt mehr "Sprachen"... Und kann somit auch mit anderen Smartcards umgehen. Super, weil sonst müsste man für jede Originalkarte die mit einer anderen Verschlüsselungstechnik arbeitet, eine Extra-Settopbox kaufen, bzw. ein Extra-CAM (falls das bei dem jeweiligen Receiver überhaupt geht), bzw. ein CI-Modul (Conditional Interface Module; da kann man die Smartcard direkt ins Modul schieben; is aber im Grunde intern dieselbe Technik)

Und dann gibts ja noch die camd... der Daemon zum CAM (deswegen der Name), also ein Stück Software, bzw. eine Software-Schnittstelle zwischen CAM (Hardware) und dem Tuxbox-Betriebssystem (Software). Könnte man vielleicht als "Treiber" fürs CAM bezeichnen. Wenn man in die Dbox2 reinguckt, dann sieht man allerdings nicht viel konventionelles CAM. Nur so ein paar festverlötete Hardwarebrocken. Die "cam-alpha.bin" spielt auch noch als "Treiber" mit (manche sagen es wäre das CAM-an-sich)... Jedenfalls bestimmt die cam-alpha.bin schonmal von wo das Signal kommt (Kabel oder SAT; die entsprechende Version von ihr muss ins jeweilige Image, sonst geht nix). Danach reiht sich der CAM-Daemon ein. Man könnte die camd so manipulieren, dass das interne "CAM" auch andere "Sprachen" durchlässt. Z.B. Seca oder sowas, statt immer nur Betacrypt. Mithilfe eines MultiCAMs geht das hardwareunterstützt mit Originalkarten. Aber was ist, wenn man keine Originalkarte hat ? Geht das auch softwareunterstützt ? Tatsächlich: Das geht ! Nennt sich dann camd-Emulator oder auch Emulator bzw. Emu ! Ohne Keys geht aber nix; also eine Schnittstelle in den Emulator, der dann die zu entschlüsselnden Keys nicht mit der Smartcard aushandelt, sondern mit ein paar Dateien, die sich im Image befinden; also ein Emu-Image ! Jetzt isses aber so, dass diese Emu-camd auch nur eine Software ist, die wieder Libraries und weitere externe Dateien benötigt, die ausserdem nur in bestimmten Pfaden liegen sollen/können/dürfen. Gibt viele Hacker-camds... z.B. camd3, newcamd, gbox, evocamd, mgcamd, etc. Da es privat-geschriebene Tools sind, und es keine Emu-Union

gibt *rofl* ist es halt so, dass man jeden Emu irgendwie anders ins Image frickeln muss...

Der CAM-Daemon (camd), der von der offiziellen Tuxbox-Developer-Crew geschrieben wurde, nennt sich camd2 und in den meisten Images ist schon eine camd2 drinnen; jedoch bietet jener (jene ?) keinen Premiere-Support, was soviel bedeutet, dass man zwar eine originale Premiere-Karte in die DBox2 stecken kann, jedoch Premiere dunkel bleibt ! Es sei denn man hat noch die Betanova am Laufen ;-)

Der Grund: Premiere hat dies bei der Tuxbox-Developer-Crew erwirken können, da sie sonst keine Lizenzen für Hardcore-Erotik in Deutschland bekommen hätten. Denn mit der Betanova-Alternativ-Software Tuxbox kann man die Jugendschutz-Sperre umgehen. Deswegen hat die Tux-Crew kurzerhand den gesamten Premiere-Empfang in ihrer camd2 abgeschaltet !

Wer nun nur Premiere mit einer originalen Premiere-Karte auf einer DBox2 mit Tuxbox-Firmware gucken möchte, muss eine gepatchte camd2 einspielen. Oder man spielt sich die camd3 ein, welche das auch kann (neben noch weiteren Features ;-). Zwar zählt die camd3 auch zur Kategorie "Emu", aber wer auch ein Geplagter ist, der die Betanova-Firmware Schei**e findet, und dennoch Premiere unter Tuxbox-Oberflächen schauen möchte, findet in der camd3 eine Alternative zur gepatchten camd2 🤖

2. Info über die Image-Arten und welche Werkzeuge benötigt werden

Welche Image-Arten gibt es ?

Es gibt die unterschiedlichsten Images ! Sie unterscheiden sich idR. erstmal durch 1xI oder 2xI (die Bezeichnung bedeutet, dass eine DBox2 entweder *einen* FlashROM-Baustein hat, oder *zwei* ! Das "I" steht für Intel, welches in den meisten DBoxen verbaut wurde. Es gibt auch AMD-FlashROMs. In dem Fall müsste es eigentlich 2xAMD heißen, aber der Kern-Unterschied liegt nicht am Hersteller, sondern an der Chip-Anzahl...) Und in diesen FlashROMs kann die Firmware "Tuxbox" (sei es neutrino oder enigma oder both-in-one) in mittlerweile drei Filesystemen stecken. Das sind JFFS2, CramFS und Squash. Ersteres ist re-writable, die anderen beiden sind read-only, aber enthalten eine bessere Realtime-Komprimierung als JFFS2 ! Squash die Beste ! Wenn mans genau nimmt, sollte man eigentlich nur noch entweder über Squash, oder JFFS2 nachdenken. Squash weils am besten komprimiert, JFFS2, weils auch re-writable ist ! Wer zwar JFFS2 haben möchte (wegen Beschreibbarkeit), aber mehr Platz braucht, kann [sb-shrink](#) zurhilfe nehmen. Mag zwar verwirrend sein, aber in eigentlich jedem CramFS/Squash-Image liegt auch mind. eine JFFS2-Partition. Warum ? Weil dort so Dinge wie Channel-Search, zuletzt geschauter Kanal, oder Auto-Update-Keys irgendwo gespeichert werden müssen, bevor die Box ausgeschaltet wird. Ein JFFS2-Only ist was anderes... Dort ist wirklich *alles* wiederbeschreibbar ! Jedenfalls wenn die camd2 in /bin liegt (was nur innerhalb eines JFFS2-Only-Images beschreibbar ist), dann führt kein Weg am Re-Imaging vorbei. Sofern es sich um ein JFFS2-Only-Image handelt, kann man das weglassen. Aber sobald man z.B. in einem Squash-Image die camd2 ersetzen will, und jene nicht im beschreibbaren Bereich (z.B. /bin) liegt, dann muss halt gebastelt werden, leider ! (wie das geht, wird hier beschrieben)

Wichtig wäre es natürlich als Erstes einmal nachzuschauen ob überhaupt noch genügend Platz im Image zur Verfügung steht ! Verbindet Euch mal mit Telnet oder Putty auf eure Box (wie das funktioniert steht gleich darunter) und gebt mal "df" ein. Werte bei knapp 90% sind idR. Ende der Fahnenstange ! Dann müsst Ihr Plugins/Spiele rausschmeissen, sonst gehts nicht ! Oder Ihr müsst anderwärtig aufräumen (z.B. [sb-shrink](#) nehmen, etc.). Durch Eingabe von "ps" unter Telnet/Putty sieht man hier schon meistens wie die eingebaute camd heisst und wo sie liegt ! In seltenen Fällen wird nur der Symlink der laufenden camd angezeigt (muss man dann selbst ausfindig machen). Symlinks sind mittels "ls" unter Telnet/Putty idR. unsichtbar ! Eingabe "ls -la" zeigt dann wirklich alles an ! Mittels FTP kann man auch einiges angezeigt bekommen...

Also zuerst benötigt Ihr ein FTP-Programm (für Windows z.B. Total Commander; Download hier <http://www.ghisler.com/> ; für Linux gehts auch mit dem Midnight-Commander [mc], der idR. vorinstalliert ist), oder Telnet (Boardmittel unter Windows und Linux) .Telnet lässt sich mit Windows via Start-Button/"Ausführen" aufrufen. Alternative für Windows ist Putty... Download hier <http://www.uni-giessen.de/hrz/software/putty/> !!

FTP-Verbindung zur DBox2 herstellen:

Windows: Total-Commander starten. Netz/neue Verbindung/dbox-Namen eingeben (zb "meine dbox"), IP der DBox eingeben; Login "root" und "dbox2" als Passwort (Achtung: Passwort kann auch ein anderes, oder auch **ohne PW** sein) !! Mit "ok" abschliessen und verbinden !

Linux: Durch Eingabe von "mc" den Midnight-Commander starten. Dann F9 drücken, mit den Cursortasten zu dem Menüeintrag "Links"- oder "Rechts" gehen, "FTP-Verbindung" auswählen und "root@192.168.x.y"(IP der Box) eingeben, und mit Eingabe des Passwortes abschliessen (wenn kein PW, dann einfach ENTER). Als Alternative könnt Ihr auch kbear verwenden - der ist sehr einfach und intuitiv zu verwenden

Jetzt solltet Ihr auf Eurer DBox2 via FTP eingeloggt sein !

Telnet/Putty-Verbindung zur DBox2 herstellen:

(Anmerkung: Nur eins der folgenden Varianten wird benötigt !)

Putty: Bei "Host-Name" gebt Ihr die IP der DBox2 ein; darunter auf Telnet klicken (Port 23) und unter "saved sessions" gebt Ihr zb "meine dbox" ein. Dannach auf "save" ! Nun Doppelklick auf "meine dbox", root/dbox2 (oder Euer eigenes Passwort eingeben)... Nun seid Ihr auch mit Putty auf der DBox2 !!

Telnet unter Windows: Start-Button/Ausführen/Eingabe: "telnet (IP der DBox2)", dann root/dbox2 (oder anderes PW, bzw. auch ohne PW)... Nun seid Ihr auch unter Windows mit Telnet auf der DBox2 eingeloggt !

Telnet unter Linux (grafische Oberfläche): Ein Konsolen-Fenster öffnen. Eingabe dort: "telnet (IP der DBox2)", dann root/dbox2 (oder anderes PW, bzw. auch ohne PW)... Nun seid Ihr mit Telnet in der DBox2 eingeloggt !

Telnet unter Linux (Textkonsole): Mittels ALT-F2 auf eine zweite Konsole wechseln (für später: zurück gehts mit ALT-F1). Dort einloggen. Eingabe dort: "telnet (IP der DBox2)", dann root/dbox2 (oder anderes PW, bzw. auch ohne PW)... Nun seid Ihr mit Telnet in der DBox2 eingeloggt !

Welche Werkzeuge werden noch benötigt ?

- mkfs.jffs2, mkcramfs, mksquashfs (je nach Image-Art nur das zutreffende benötigt; liegt alles dem Anhang bei)
- Optional: NewTuxBoxFlashTools (NTBFT) <--- gibts hier in der Database !
- DBox2-Boot-Manager... Hiermit kann man unter Windows Images flashen, aber auch Bootlogs erstellen und Win-Partitionen mounten (und vieles mehr)... (Download: <http://dbox.feldtech.com>)
- Un-TAR Utilities für Windows zum enttaren von Files ! Kann man machmal brauchen... Auch Files mit .gz und .bz2 Endung sind hiermit unter Windows entpackbar (Download: <http://dbox.feldtech.com>)
- Linux auf dem PC ! Optional: Eine der vielen Linux-Emulationen unter Windows wie cygwin oder Bochs; oder VMware oder Virtual-PC als "PC-Emulator" für einen PC-im-PC in dem ein Linux unter Windows läuft , etc.
- Optional zum Editieren von Texten unter Windows im Linux-Format: Ultraedit (Download: <http://www.ultraedit.com/>)

3. Einbau/Update der camd3 ins DBox2-Image

Wie bereits erwähnt, ist die camd3 die Alternative zur "offiziellen" camd2 ! Das einfachste wäre nun, einfach die camd3 in camd2 umzubenennen, und im Image unter /bin zu überschreiben. Bei einem JFFS2-Only-Image wär das dann auch schon fast alles. 😊 Da aber bei allen anderen Images das Verzeichnis /bin in einem NICHT-beschreibbaren Bereich liegt, sollte man sich was anderes ausdenken...

Jedes Image verfügt über einen beschreibbaren Bereich (JFFS2)... Auch ein Cramfs- oder ein Squash-Image (!) Idee also, die camd3 in den JFFS2-Bereich eines Cramfs/Squash-Images abzulegen !

Was aber machen, wenn das Image beim Booten bereits die "offizielle" camd2 startet und die camd3 hinterher ? Das geht leider nicht... Geht immer nur eins (!) Lüftung des Geheimnisses ist die Datei "start_neutrino" oder "start_enigma" ! Dort könnte man alle Einträge "/bin/camd2" einfach in /var/camd3" abändern ! Fehlt dann nur noch die Konfiguration (detaillierter beschrieben unter Punkt 5.)

Einige Image-Ersteller haben da sogar schon was vorbereitet; lies´ weiter unter Punkt 3.1. (!)

3.1. Einbau der camd3 (Image ist noch ohne camd3)

Wie eben schon beschrieben, haben einige Image-Ersteller schon eine "Schnittstelle" zum Laden einer Alternativen camdX eingebaut. Schauen wir uns die "start_neutrino" einiger Images mal an...

AlexW-Image (1.7.1)

```
code:
1: if [ -x /var/bin/camd2 ]; then
2: /var/bin/camd2
3: else
4: if [ -e /var/bin/camd2 ]; then
5: chmod +x /var/bin/camd2
6: /var/bin/camd2
7: else
8: /bin/camd2
9: fi;
```

Yadi-Image

```
code:
1: #!/bin/sh
2: sectionsd
3: timerd
4: if [ -e /var/bin/camd2 ]; then
5: /var/bin/camd2
6: else
7: /bin/camd2
```

JackTheGrabber-Image (JTG)

```
code:
1: fi;
2: /bin/sections
3: /bin/timerd
4: /bin/zapit
5: if [ -x /var/bin/camd2 ]; then
6: /var/bin/camd2
7: else
8: if [ -e /var/bin/camd2 ]; then
9: chmod +x /var/bin/camd2
10: /var/bin/camd2
11: else
12: /bin/camd2
13: fi;
14: fi;
```

Bits4Fun-Image

```
code:
```

```

1: zapit
2: if [ -x /var/bin/camd2 ]; then
3: /var/bin/camd2
4: else
5: if [ -e /var/bin/camd2 ]; then
6: chmod +x /var/bin/camd2
7: /var/bin/camd2
8: else
9: /bin/camd2

```

Das sind jeweils nur ausgeschnittene Teile der start_neutrino... Wie man sieht, wiederholt sich ein Teil dessen in vielen Images... Nehmen wir's mal unter die Lupe:

Auszug der start_neutrino:

```

code:
1: if [ -x /var/bin/camd2 ]; then
2: /var/bin/camd2
3: else
4: if [ -e /var/bin/camd2 ]; then
5:   chmod +x /var/bin/camd2
6:   /var/bin/camd2

```

Es wird nach einer camd2 im beschreibbaren Teil /var/bin gesucht und wenn eine gefunden wurde, werden die Rechte zum Ausführen vergeben, sofern nicht schon erfolgt; dann wird gestartet !!

```

code:
1: else
2: /bin/camd2

```

Wenn keine gefunden wird, wird die originale camd2 aus /bin gestartet (die jedoch ohne Tüfteleien ohne Premiere-Support daherkommt) !!

ALLES WAS WIR ALSO NUN MACHEN MÜSSEN, IST DIE CAMD3 IN CAMD2 UMZUBENENNEN UND AN DIE STELLE IM /VAR BEREICH ZU SCHIEBEN, WIE ES IN DER START_NEUTRINO DEKLARIERT WURDE !

*Angenommen Ihr habt die *brandneue* camd3 aus dem Streamboard runtergeladen, und jene heisst "camd3630"... Jene kopiert Ihr dann via FTP nach /var/bin der DBox ! Nun muss sie nur noch umbenannt werden... Zuvor muss aber die laufende camd (sei es die camd2, oder die camd3, oder die in camd2 umbenannte camd3) **abgeschossen werden***

Ihr öffnet telnet/putty und gebt folgendes ein:

killall camd2 (oder wie auch immer sie bei Euch heisst)

Hierdurch wird die camd2 aus dem RAM gelöscht.

Als nächstes müsst Ihr das Verzeichnis /var/bin anlegen (falls das Verzeichnis noch nicht existiert)... Der ganze Vorgang incl. /var/bin erstellen hier im Detail...

Windows: Via FTP unter /var das Verzeichnis /bin erstellen. Nun in das Verzeichniss /var/bin wechseln und die camd3630 dort reinkopieren; danach in camd2 umbenennen !

Linux-"mc": Mit TAB auf die "DBox2-Seite" wechseln, in /var wechseln, F7 drücken, "bin" eingeben (erstellt /var/bin), in /var/bin gehen, mit TAB auf die Festplattenseite gehen, F5 drücken (kopiert die camd3630 in /var/bin), mit TAB wieder auf die DBox-Seite wechseln, den Cursor auf camd3630 legen, F6 drücken und eingeben "camd2"+ ENTER (benennt die camd3 um)

Telnet/Putty: Eingabe "cd /var", dann "mkdir bin", dann mit FTP die camd3630 in /var/bin uppen, dann wieder

unter Telnet/Putty Eingabe (unter /var/bin) "mv camd3606 camd2"

Natürlich könnt Ihr diese camd3630 auch *gleich* in camd2 umbenennen und nach /var/bin kopieren; das spart Zeit. Jeder wie er will...

Jetzt geben wir der "camd2" noch die erforderlichen Rechte. Wenn Ihr nach Attributen schaut, steht dort möglicherweise "644" ! Das ändern wir nun auf 755. Geht wie folgt:

Windows-FTP: camd2 anklicken im Menü oben auf Dateien/Dateiattribute ändern und überall Häkchen bei "Ausführen" setzen. Mit "ok" abschließen !!

Linux/Windows via Putty/Telnet: Mit Telnet/Putty-Konsole dorthin wechseln, wo die in camd3 liegt. Also "cd /var/bin" und gebt dann "chmod 755 camd2" ein !

Nun sind die Rechte auf 755 (bzw. halt auf "Ausführen") gesetzt !

ALS NÄCHSTES EDITIERT IHR DIE START_NEUTRINO (ODER START_ENIGMA) SO, DASS DIE WEITER OBEN BESCHRIEBENEN PASSAGEN DRINSTEHEN, FALLS SIE NICHT SOGAR SCHON DRINSTEHEN !!!!

Das Verzeichnis /var/keys anlegen und Konfigs kopieren

In der selben Weise wie wir oben /var/bin angelegt haben, erstellen wir noch /var/keys... Und in der selben Weise, wie wir die camd3 (als camd2) ins Image kopiert haben, kopieren wir die Dateien camd3.config, camd3.keys, camd3.users, camd3.servers in **/var/keys!**

Wenn wir mit der camd3 Premiere-Unterstützung durch die Originale Smartcard haben wollen, machen wir folgendes:

Wir öffnen die camd3.config (liegt fast bei jedem camd3-Release von doz21 dabei) unter Windows z.B. mit Ultraedit, oder unter Linux mit dem "mc" mittels F4, und ändern z.B. wie folgt (man könnte durch Telnet/Putty auch direkt innerhalb der DBox2 mittels "vi camd3.config" ändern, aber der vi ist umständlich zu bedienen; wer sich auskennt kann den vi aber nutzen):

(ACHTUNG ! NICHT MIT NOTEPAD ODER WORD, ETC. EDITIEREN)

Bei den meisten Dateien ist das zwar egal, aber in "camd3.users" kann es Probleme mit den Passwörtern geben.

```
code:
1: Port=24110
2: Osd=1
3: Debug=1
4: Slot0_Start=0
5: Slot1_Start=0
6: Slot2_Start=0
7: Slot3_Start=1
8: Slot0_AU_G=0
9: Slot0_AU_S=0
10: Slot0_AU_U=0
11: Slot1_AU_G=0
12: Slot1_AU_S=0
13: Slot1_AU_U=0
14: Slot3_AU_A=1
```

Windows: Unter Ultraedit auf Konvertieren/DOS auf Unix gehen und abspeichern !

Linux: Beim "mc" einfach 2x Escape drücken und die Nachfrage zum Abspeichern mit "Ja" quittieren ! Bei anderen Editoren (z.B. unter KDE) halt eben abspeichern; ein Konvert DOS/Unix nicht erforderlich !

Die Dateirechte müssen Lesezugriff haben. Können auf 644 bleiben (also nicht-ausführbar) !

3.2. Nachdem wir das Binary eingebaut haben kommen nun die Konfigs

Da sich ab- und an mal bei den Files was ändert, sollte man in dem entsprechenden Thread von doz21 mal nachlesen, bzw. die meist bei Änderungen von doz21 beiliegenden Beispieldateien mit denen im Image vergleichen... "Hell" wirds natürlich nur mit korrekten Konfigs und korrekten Keys... Genauer wird das unter Punkt 4. erklärt !

3.3. Ein erster Test...

Da wir ja vorhin die laufende camd "abgeschossen" haben, werden wir nun mal einen ersten Test der camd3 mittels Telnet/Putty starten:

Einfach (in unserem Beispiel) /var/bin/camd2 eingeben... danach sollte so ein ähnlicher Log erscheinen/kommen...

```
code:
1: [camd]: Slot0: deaktiviert
2: [camd]: Slot1: deaktiviert
3: [camd]: Slot2: deaktiviert
4: [camd]: Slot3: aktiviert
5: [camd]: OSD aktiviert
6: [camd]: DEBUG aktiviert (mit Verbindungsinformationen)
7: [camd]: UserDB: USER=benutzer1 PASS=password1 CAIDs=Alle
8: [camd]: /var/keys/camd3.servers: HOST=192.168.0.7 PORT=24110 : alle
9: CAIDs
10 [camd]: /var/keys/camd3.users - No such file or directory
   : [camd]: KeyDB: gestartet
11 camd !!!STREAMBOARD!!! v3.606 - erfolgreich gestartet
   :
```

Wichtig ist die letzte Zeile ".....**erfolgreich gestartet**" ! Wenn die nicht kommen sollte, dann habt Ihr irgendwas falsch gemacht !

Das war es im Groben ! Nun die DBox neu booten... Die originale Premiere-Karte sollte nun im Originalslot der DBox2 funzen.

In manchen Images (Boxcracker) ist die camd3 unter /emu oder (Zg0re + SB-Image) unter /bin/emu zu finden und heißt evtl. camd3.bin, oder gar camd3.ppc ! Aber es ist fast immer der gleiche Ablauf ! Symlinks nicht ausser Acht lassen !!!

Wer Plugins benötigt um die camd2 (die umbenannte camd3606) zu starten/beenden findet diese im Anhang ! Dieses Plugin kann jederzeit umgebaut werden womit zb der start einer camd3 mit den Namen camd3633.bin möglich ist !!

Wichtig ist, in neueren Versionen der camd3 von doz21 die mitgelieferten Dateien camd3.config, camd3.users camd3.keys, camd3.servers anzusehen da sich dort öfters mal was ändert !

3.3.1 Einbau der Dateien "camd3_start", "camd3_stop" und "pcamd"

Die Datei **camd3_start** dient zum Start von camd3. Zusätzlich wird der Treiber "multicam.o" geladen, der für ein Funktionieren eines erweiterten Kartenlesers in der dBox unabdingbar ist.

Die Datei **camd3_stop** dient dazu, die camd3 ordnungsgemäß zu beenden ohne die Keydatei zu zerstören, wie es aus früheren Versionen bekannt ist. Dazu wird nach Aufruf des "kill"-Befehles für camd3 in einer Schleife gewartet, bis sich der Prozess ordnungsgemäß beendet hat.

Zuerst müsst ihr in den Dateien "camd3_start" und "camd3_stop" (editieren z.B. mit Ultraedit) den Namen eurer Camd korrekt eintragen.

Hier dazu ein Auszug der Datei "camd3_stop":

```
code:
1 #!/bin/sh
2 :
3 # Diese Zeile ANPASSEN !!!
4 # Du muss hier den Name von deinem camd eingeben (camd2 oder camd3 oder
5 camd3606 usw.)
6 : CAMD_NAME="camd3" <--- hier ändern
7 :
8 :
9 :
10 :
```

Die Datei **pcamd** ist im Gegensatz zu den vorgenannten Scripten ein Binärfile, das ebenfalls für ein korrektes Beenden des "camd3-Prozesses sorgt. Derzeit existiert nur der Aufrufparameter "pcamd -kill". Es macht Sinn, den Aufruf dieses Programmes in die Start-Datei ("start_neutrino" oder "start_enigma") einzuarbeiten, damit auf jeden Fall beim Herunterfahren der dBox camd3 ordnungsgemäß beendet wird und nicht die Datei "camd3.keys" zerschossen wird.

Sinnvollerweise wird z.B. in "start_neutrino" der Aufruf "pcamd" vor der Zeile "pzapit -kill" eingefügt. Das Ganze sollte dann so z.B. aussehen:

```
code:
1: ...
2: /bin/neutrino -u -f
3: /bin/pcamd -kill
4: pzapit -kill
5: ...
```

Danach einfach mit einem FTP Programm auf die D-Box einloggen, und in den Ordner "/var/bin" die Dateien "camd3_start", "camd3_stop" und "pcamd" hineinkopieren, Rechte auf "755" setzen, zuletzt die Dateien "start_neutrino"/"start_enigma" noch anpassen. Siehe dazu auch die Beispieldatei von Doz, die jeder neuen camd3-Version beiliegt. Nach dem Neustart der Box sollte das Ganze korrekt laufen.

Die Methode ist zwar schön- und gut, aber in vielen Fällen ist nicht genug Platz in dem bei Cramfs- und Squash-Images zur Verfügung gestellten JFSS2-Bereich ! In solch einem Fall bitte hier weiterlesen (verschiedene Varianten):

Bitte unter 4. weiterlesen...

3.4. Update einer neueren camd3-Version (Image hatte bereits camd3 an Board)

Zuerst schaut man mal nach unter welchem Namen hier nun tatsächlich eure camd3 gestartet wird. Dazu gibt es mehrere Möglichkeiten:

- Ihr verbindet Euch mit FTP auf die DBox und sucht in den folgenden Verzeichnissen /var/bin oder /emu nach einer camd3
- Ihr grast das ganze Image nach einer camd3 ab (aufwendig, aber man lernt gleich ein wenig sein Image kennen)
- Ihr Schaut Euch das Plugin an mit dem die camd3 neu gestartet wird !!
- Ihr schaut Euch die start_neutrino an. Meist steht auch da die genaue Bezeichnung und das Verzeichnis !!
- Ihr gebt unter telnet/putty den Befehl "ps -fe" ein, und werdet fündig !!

Anmerkung: Manchmal ist die camd3 auch verlinkt mittels Symlinks (aufpassen). Telnet/putty zeigt die Symlinks an mit "ls -la" an !!

*Angenommen Ihr habt die *brandneue* camd3 aus dem Streamboard runtergeladen, und jene heisst "camd3630"... In Eurem Image heisst die Datei aber "camd3.bin" statt "camd3630" (und liegt in /emu). Kein Problem... wird gleich erklärt...*

Anmerkung: Die camd3 ist beliebig umbenennbar !!! Nun zu dem Vorgang im Detail:

Ihr öffnet telnet/putty und gebt folgendes ein:

killall camd3.bin (oder wie auch immer sie bei Euch heisst)

Hierdurch wird die camd3.bin aus dem RAM gelöscht. Danach:

```
rm /emu/camd3.bin
```

Hiermit wird die camd3.bin auch aus dem FlashROM in /emu gelöscht !

Nun in das FTP-Programm wechseln und Eure neue "camd3630" nach /emu kopieren und dort in "camd3.bin" umbenennen.

Grund: Die bestehenden Plugins funzen weiterhin, und die start_neutrino muss nicht geändert werden !!

Jetzt vergeben wir der "neuen camd3" noch die erforderlichen Rechte. Wenn Ihr nach Attributen schaut, steht dort möglicherweise "644" ! Das ändern wir nun auf 755. Geht wie folgt:

Windows-FTP: camd3.bin anklicken im Menü oben auf Dateien/"Dateiattribute ändern" und überall Häkchen bei "Ausführen" setzen. Mit "ok" abschließen !!

Linux/Windows via Putty/Telnet: Mit Telnet/Putty-Konsole dorthin wechseln, wo die in camd3.bin liegt (cd /emu). Dann dort "chmod 755 camd3.bin" eingeben!

Nun sind die Rechte auf 755 (bzw. halt auf "Ausführen") gesetzt!

Ein Neustart der Box kann danach nicht schaden und alles sollte wie zuvor funktionieren halt nur mit neuer Camd3 !

4. Bearbeiten eines Nicht-Beschreibbaren Bereiches im Image

Wichtig!! Denkt immer daran, vorher eine Sicherung eures kompl. Images zu machen, falls mal was daneben geht !!

4.1. Bearbeiten eines CRamFS-Images mit New-Tux-Flash-Tools unter WinXp

Das Programm hat zwar einen Fehler, aber wenn man weiss wie man damit umgehen muss, funzt es einwandfrei !

Auslesen des Cramfs-Bereiches

Unter der DBox selbst: Cramfs auslesen per Service/Software Aktualisierung/Experten Funktion, via FTP aus /tmp der Box auf den PC holen...

|----->

Alternative:

Via telnet/putty mit der DBox verbinden...

```
cat /proc/mtd
```

dannach erhält man eine Auflistung die so ähnlich aussehen könnte !!

code:

```
1: mtd0: 00020000 00004000 "BR bootloader"
2: mtd1: 00020000 00020000 "flfs (ppcboot)"
```

```

3: mtd2: 006e0000 00020000 "root (cramfs)"
4: mtd3: 000e0000 00020000 "var (jffs2)"
5: mtd4: 007e0000 00020000 "flash without bootloader"
6: mtd5: 00800000 00020000 "complete flash"

```

nun wechseln wir nach /tmp mit

```
cd /tmp
```

Ist das Cramfs nun zb. ein mtd2 wie im beispiel oben, dann folgendes eingeben

```
cat /dev/mtd/2 > mtd2.img
```

Nun liegt das Cramfs mit namen mtd2.img in /tmp der DBox und kann per FTP auf eurem Computer kopiert werden !

```
<-----|
```

Dann das Cramfs mit New-Tux-Flash-Tools öffnen, und mal Platz machen... Z.B. alle Spiele raus; danach kann man ohne Prob. den ganzen Ordner \share\tuxbox\sokoban löschen unter \share\fonts md_khmrabi_10.ttf löschen die braucht niemand. Ausserdem kann man z.B. die Sprachunterstützungen bis auf Deutsch rausnehmen. Liegt in \share\tuxbox\neutrino\locale ! Und wenn man eine SAT-DBox2 besitzt, könnte man auch noch in \share\tuxbox die cables.xml löschen.

Nun ist Gelegenheit, weitere Dinge in das Image reinzukopieren... (RECHTE NICHT VERGESSEN)
Z.B. camd36xx umbenennen in camd2 und ab nach /bin ! Rechte nicht vergessen... Plugins einfügen nach /lib/tuxbox/plugins einfügen, wenn ´s jemand unbedingt dort haben möchte !!

Jetzt geht man auf Extras/Einstellungen, gibt die IP-Adresse, Com-Port und Passwort der DBox ein, und schließt mit ok ab.

Als nächsten Schritt geht man mit einem Rechtsklick auf der linken Seite auf den Cramfs-Heimordner, und klickt auf "Filesystem zur box uploaden" !!!

Nun befindet sich das soeben geänderte Cramfs in /var/tmp Eurer DBox (als cramfs.img), und kann wieder bequem per Expertenfunktion in die Box geflasht werden !!!
(Service/Software Aktualisierung/Experten Funktion/Einzelne Partition einspielen/cramfs)

Man sollte drauf achten, dass man unter 4950 kB bleibt !!

wenn das neue cramfs mal über 4950 kb hat und nach dem flashen kein system kommt nehmt einfach bei einer sat box die zuvor beschriebene cables.xml aus dem cramfs raus bzw bei einer Kabel Box umgekehrt und dann funzt es auch wieder man muss oft nur eine ganz geringfügige änderung am cramfs vornehmen und es kommt keine fehlermeldung wie zb kein system zu!

4.2. Bearbeiten eines CRamFS-Images

Verbindung mit telnet/putty zur DBox herstellen; folgendes eingeben...

```
cat /proc/mtd
```

Könnte dann z.B. sowas kommen...

```

code:
1: dev:      size  erasesize  name
2: mtd0: 00020000 00004000 "BR bootloader"
3: mtd1: 00020000 00020000 "flfs (ppcboot)"
4: mtd2: 003e0000 00020000 "root (cramfs)"
5: mtd3: 00300000 00020000 "emu (jffs2)"
6: mtd4: 000e0000 00020000 "var (jffs2)"

```

```
7: mtd5: 007e0000 00020000 "flash without bootloader"
8: mtd6: 00800000 00020000 "complete flash"
```

Hier liegt der Cramfs-Bereich im mtd2, und die Grösse ist \$3e0000, bzw. 4063232 Bytes. Diese Info brauchen wir ! Die folgende Anleitung bezieht sich mal auf dieses Beispiel ! Wenn das root(cramfs) bei Euch in einem anderen Bereich liegt und eine andere Grösse hat, müsst Ihr entsprechend handeln.

Als nächstes holt Ihr Euch mit der DBox selber den Bereich root(cramfs) auf den Linux-PC. Dazu geht Ihr in Menü/Service/Software-Aktualisierung/Experten-Funktion/individuelle Partition auslesen und wählt root(cramfs). Das erstellt ein File in /tmp der Box namens "mtd2.img"; dieses holt Ihr Euch mittels FTP auf Euren Linux-PC !

|----->

Alternative:

Via telnet/putty mit der DBox verbinden... Ist das Cramfs nun ein mtd2 wie im Beispiel oben, dann folgendes eingeben

```
cat /dev/mtd/2 > /var/tmp/mtd2.img
```

Nun liegt das Cramfs mit Namen mtd2.img in /tmp der DBox und kann per FTP auf eurem Computer kopiert werden !

<-----|

Weiter geht´s auf dem **Linux-PC**:

Konsole öffnen... Eingabe:

```
cd $HOME
mkdir test
cd test
```

Nun haben wir ein Verzeichnis "test" angelegt, und dort kopieren wir das "mtd2.img" rein, sowie die beiden beiliegenden Files "cramfsck" und "mkcramfs" ! Dann müssen wir den beiden Files noch Ausführ-Rechte vergeben. Eingabe

```
chmod +x cramfsck
chmod +x mkcramfs
```

Nun entpacken wir das "mtd2.img" mit cramfsck... Eingabe

```
./cramfsck -x outpacked mtd2.img
```

Da kommt zwar häufig ein "warning: file extends past end of filesystem", aber es passiert nix Schlimmes 🤖 Das Verzeichnis "outpacked" muss nicht vorher angelegt werden ! Das übernimmt das Tool cramfsck mit dem Namen als Parameter... Nun nehmt Ihr im Verzeichnis "outpacked" Eure Änderungen vor, und es geht wieder ans ver-cramfs´en ! Folgende Eingabe

```
cd $HOME/test
./mkcramfs outpacked new-mtd2.img
```

Auch hier gibts häufig ein "warning: gids truncated to 8 bits (this may be a security concern)", aber auch hier passiert nix wirklich schlimmes 🤖

Nun habt Ihr Euer bearbeitetes cramfs, welches Ihr dann mittels FTP wieder in die DBox in /tmp uppen und mit den Expertentools einspielen könnt.

(Service/Software-Aktualisierung/Experten-Funktion/Einzelne Partition einspielen/root(cramfs))

Anmerkung: Wie wir vorhin gesehen haben, ist der reservierte Platz im Image \$3e0000 (4063232 Bytes) gross ! Das von der DBox geholte mtd2.img hat auch eben diese Grösse. Macht man ohne Änderung gleich wieder ein cramfs draus, dann ist es kleiner ! Warum ? Weil zuvor das ganze Device ausgelesen wurde, auch wenn der beschriebene Datenteil ein kleinerer ist ! Das eben erstellte new-mtd2.img darf nicht grösser als die obige Byte-Angabe sein. Es darf aber durchaus kleiner sein !!!! Dann ist halt ein bisschen "luftleerer, ungenutzter" Raum im Image... Ist Euer new-mtd2.img z.B. dann 3854336 Bytes gross, so ist ein "Loch" von 208896 Bytes im Image... Dieses Loch kann nicht genutzt werden, aber es passiert auch nix Schlimmes; will sagen: es muss nicht 4063232

Bytes gross sein !!!! Ihr habt also alles richtig gemacht...

4.3. Bearbeiten eines Squash-Images

Als erstes bootet Ihr mal Euren Linux-PC und startet eine Konsole. Dann gebt Ihr mal "mksquashfs" ein. Wenn da ein Error kommt, beendet Ihr die Konsole, loggt Euch als root ein und kopiert das beiliegende "mksquashfs" in /usr/bin/ Eures Linux-PCs ! Durch Eingabe von

```
chmod +x /usr/bin/mksquashfs
```

vergebt Ihr die Rechte zum Ausführen. Von nun an ist Eurem Linux-PC der Befehl sowohl als root, als auch als user bekannt. Nun loggt Ihr Euch wieder als user in Euren Linux-PC ein...

Verbindung mit telnet/putty zur DBox herstellen; folgendes eingeben...

```
cat /proc/mtd
```

Könnte dann z.B. sowas kommen...

code:	dev:	size	erasesize	name
1:	mtd0:	00020000	00004000	"BR bootloader"
2:	mtd1:	00020000	00020000	"flfs (u-boot)"
3:	mtd2:	000a0000	00020000	"kernel (jffs2)"
4:	mtd3:	005c0000	00020000	"root (squashfs)"
5:	mtd4:	00160000	00020000	"var (jffs2)"
6:	mtd5:	007e0000	00020000	"flash without bootloader"
7:	mtd6:	00800000	00020000	"complete flash"

Hier liegt der Squash-Bereich im mtd3, und die Grösse ist \$5c0000, bzw. 6029312 Bytes. Diese Info brauchen wir ! Die folgende Anleitung bezieht sich mal auf dieses Beispiel ! Wenn das root(squashfs) bei Euch in einem anderen Bereich liegt und eine andere Grösse hat, müsst Ihr entsprechend handeln. Als nächstes folgendes eingeben...

```
mkdir /var/tmp/root
```

Damit wird erst einmal ein Verzeichniss Namens root in /var/tmp erstellt... Danach folgende Eingabe...

```
mount -t squashfs -o ro /dev/mtdblock/3 /var/tmp/root
```

Wenn alles exakt eingetippt wurde, wurde nun der Inhalt vom Squash-Bereich in das Verzeichniss /var/tmp/root gemountet ! Ins Verzeichnis /var/tmp wechseln. Eingabe...

```
cd /var/tmp <<--- genau *so* eingeben; NICHT in /var/tmp/root wechseln !!!
```

Nun muss das Ganze "getart" werden, damit die Rechte und Symlinks nicht futsch gehen. Eingabe...

```
tar -cf root.tar root
```

Nun liegt das getarte File unter /var/tmp/ und lautet root.tar. Das kann durchaus 15 Megabytes gross sein, denn das Squash-Filesystem ist ja ein Komprimierendes, und nun liegt es unkomprimiert, aber ge-tart da ! Via FTP auf den Linux-PC holen... Nun könnt Ihr das root.tar unter /var/tmp wieder löschen und durch Eingabe von "umount /var/tmp/root" das Mount des Squash-Bereiches wieder un-mounten und mit "rm -f /var/tmp/root" das Verzeichnis wieder löschen... Ihr braucht ja für später wieder etwas RAM-Platz !

Weiter geht ´s auf dem **Linux-PC**:

Konsole öffnen... Eingabe:

```
cd $HOME
```

mkdir test

Das eben von der DBox2 heruntergeladene root.tar in \$HOME/test kopieren; dann weiter...

```
cd /test
tar -xf root.tar
```

Hiermit wurde es wieder ent-tart. Nun alle Änderungen in \$HOME/test/root/.. vornehmen...
Wenn Ihr damit fertig seid, geht Ihr mit "cd \$HOME/test" wieder in die Ebene zurück in der sich das /root als Verzeichnis befindet ! Zum wieder ver-squashen gebt Ihr folgendes ein:

```
fakeroot mksquashfs root mtd3.img -be
```

Wenn Ihr Probleme mit dem Befehl "fakeroot" haben solltet, dann loggt Euch halt von Anfang an als root in Eurem Linux-PC ein, und die Befehlszeile lautet dann "mksquashfs root mtd3.img -be"... Übrigens steht "-be" für Big-Endian. Nun habt Ihr das root-Image im Squash-Format, welches Ihr dann mittels FTP wieder in die DBox in /tmp uppen und mit den Expertentools einspielen könnt.
(Service/Software-Aktualisierung/Experten-Funktion/Einzelne Partition einspielen/root (squashfs))

Anmerkung: Wie wir vorhin gesehen haben, ist der reservierte Platz im Image \$5c0000 (6029312 Bytes) gross ! Das eben erstellte mtd3.img darf nicht grösser als diese Byte-Angabe sein. Es darf aber durchaus kleiner sein !!!! Dann ist halt ein bisschen "luftleerer, ungenutzter" Raum im Image... Ist Euer mtd3.img z.B. dann 5779456 Bytes gross, so ist ein "Loch" von 249856 Bytes im Image... Dieses Loch kann nicht genutzt werden, aber es passiert auch nix Schlimmes; will sagen: es muss nicht 6029312 Bytes gross sein !!!! Ihr habt also alles richtig gemacht...

5. Konfiguration der camd3

5.1. Konfigdateien

Folgende Dateien MÜSSEN (neben der binären camd3) in Euer Image:

```
camd3.config
camd3.keys
camd3.users (muss für Cardsharing auf den Server)
camd3.servers (muss für Cardsharing auf den Client)
```

Alle Dateien müssen nach /var/keys kopiert werden !!!

camd3.config

Beispieldatei von DOZ21

```
Zitat:
# Wenn Parameter gebraucht wird - '#' entfernen und eure Werte reinsetzen!!!!
#
#
# Boxytyp(optional): 1-dbox2, 2-dm7000, 3-dm7020, 4-dm500, 5-dm56xx/Triax; Default ist 2
# BOXTYPE=1
#
# UDP Port(optional); Default - 20248
#UDP_PORT=24110

# TCP Port(optional); Default - 0 (TCP-server ausgeschaltet)
#TCP_PORT=24110

# Log(optional): 0 - keine Ausgaben, 1-Datei, 2 - Console, 4 - UDP(syslog), 3 - Console+Datei, 5 - UDP+Datei,
6 - UDP+Console, 7-UDP+Console+Datei; Default ist 2;
#LOG=2
```

```
# Host für UDP-logging
#LOG_HOST=192.168.1.1

# Port für UDP-logging(optional); Default ist 514
#LOG_PORT=514

# Logfile
#LOG_FILE=/mnt/hdd/var/log/camd3.log

# DNS cache in sekunden; Default ist 600 sec
#DNS_CACHE=600

# Pfad zum User-File(optional); Default ist /var/keys/camd3.users
#USERS=/var/keys/camd3.users

# Pfad zum Server-File(optional); Default ist /var/keys/camd3.servers
#SERVERS=/var/keys/camd3.servers

# Pfad zum Key-File(optional); Default ist /var/keys/camd3.keys
#KEYS=/var/keys/camd3.keys

# Pfad zum Socket-File(optional); Default ist /tmp/camd.socket
#SOCKET=/tmp/camd.socket

# Multichannel mode (optional): 0-aus, 1-ein; Default ist 0
#MULTI=1

# Kartenslots (anzahl beliebig)
# je eine Zeile für Slot:
# SLOT=devicename:type:au_g:au_s:au_u:port:username:password:deskey
#
# deskey für normale Slots ist optional, aber port,username und password müssen immer angegeben
werden!!!
#
# devicename: Pfad (vollständig) zum Device, oder Hostname von cardserver(newcamd)
# type: 1-serial, 2-intern(dreamox), 4-cardserver(newcamd)
# au_g: 1-eingeschaltet, 0-ausgeschaltet
# au_s: 1-eingeschaltet, 0-ausgeschaltet
# au_u: 1-eingeschaltet, 0-ausgeschaltet
# port: port
# username: loginname
# password: loginpassword
# deskey: cardserver deskey (14 byte)

# für Dreambox
# Serial
#SLOT=/dev/tts/0:1:1:1:1:997:mastercrd:password1
# Slot oben
#SLOT=/dev/sci1:2:1:1:1:998:slotoben:password2
# Slot unten
#SLOT=/dev/sci0:2:1:1:1:999:slotunten:password3
# Newcamd Cardserver
#SLOT=localhost:4:1:1:1:2222:dummy:dummy:0102030405060708091011121314

# für dbox2
# Serial
#SLOT=/dev/tts/0:1:1:1:1:997:mastercrd:password1
# Multicam
#SLOT=/dev/tts/1:1:1:1:1:998:multicam:password2

# Au für den original Slot der dbox2: 0-aus, 1-an; Default ist 1
#DBOX2_AU=0
```

Was bedeutet Boxtyp ?

Mit Hilfe des Boxtyp stellt man die verwendete Box ein auf der ihr die camd3 installieren wollt, ändert einfach die Zahl entsprechend der verwendeten Box.

Für was ist der UDP / TCP Port ?

Die Übertragung wird bei der camd3 entweder über das Protokoll UDP oder TCP gemacht. Bei einer der Serverbox legt dies die udp oder tcp_Portnummer fest, unter der der Server auf eingehende Anfragen lauscht. Genau der gleiche muß bei den Clients in der camd3.servers eingetragen werden damit eine Kommunikation ermöglicht wird. Weiterhin wird über den Porteintrag in camd3.config festgelegt, auf welchem Absende-Port der Client seine Anfragen an den Server stellt.

Beispiel:

Die Serverbox hat Port 20248, Clientbox Port 20250 in der camd3.config und Port 20248 in der camd3.servers stehen.

Der Client sendet nun mit Absendeport 20250 an den Zielport 20248 des Servers. Bei der Antwort ist es umgekehrt. Der Server nutzt Port 20248 als Absender und schickt seine Daten an Port 20250 der Clientbox.

Für was ist der Logeintrag ?

Die camd3 gibt verschiedene Informationen aus die z.B. Informationen über die Cardsharinganfragen und Verarbeitung enthalten. Dabei gibt es verschiedene Möglichkeiten diese Informationen anzeigen zu lassen welche mit diesem Eintrag geregelt werden können.

Datei:

Die Loginformationen werden in einer Datei auf der Box gespeichert

Console:

Gibt die Informationen über Telnet aus wenn man die camd3 manuell über Telnet gestartet hat.

UDP:

Die Loginformationen werden über einen UDP Port an eine beliebige IP gesendet und können dann z.B. am PC abgefragt werden.

Was bedeutet LOG_HOST ?

Hier wird die IP Adresse angegeben an welche die Log-Informationen gesendet werden sollen fall UDP als Logart gewählt wurde.

Was bedeutete LOG_PORT ?

Dies regelt den Port über welchen das Log an die zuvor eigestellt IP gesendet wird.

Was bedeutet LOG_FILE ?

Hiermit wird der Speicherort der Logdatei angegeben fall Datei als Logart gewählt wurde.

Bedeutung der Log-Ausgaben:

Zitat:

```
00 - ECM (request)
01 - ECM (response)
02 - EMM (für client - set EMM, für server - EMM data) -> veraltet, bleibt aus Kompatibilitätsgründen
03 - ECM (cascading request)
04 - ECM (cascading response)
```

05 - Kartendaten zum Client (eingehend beim Client)
06 - EMMs (eingehende EMM-Daten beim Server)

normales Sharing Client - Server:

172.16.0.121 = Client

172.16.0.123 = Server

- auf dem Server eingehende Anfrage:

Oct 5 00:54:26 camd3: UDP 00 -> 1702:000000:100A:000A 0.286008s from 172.16.0.121 (User1) decoded on /dev/sci1

- die Antwort des Servers auf dem Client:

Oct 5 00:54:28 camd3: UDP 01 -> 1702:000000:100A:000A 0.002755s from 172.16.0.123 (User1)

Sharing mit kaskadierten Servern:

172.16.0.121 = Client

172.16.0.123 = 1. Server

172.16.0.124 = 2. Server

- Anfrage des Clients beim 1. Server:

Oct 5 00:54:26 camd3: UDP 00 -> 1702:000000:100A:000A 0.286008s from 172.16.0.121 (User1)

- Anfrage des 1. Servers bei einem 2. Server

Oct 5 00:54:28 camd3: UDP 03 -> 1702:000000:100A:000A 0.286008s from 172.16.0.123 (User2) decoded on /dev/sci1

- Antwort des 2. Servers an den 1. Server:

Oct 5 00:54:30 camd3: UDP 04 -> 1702:000000:100A:000A 0.286008s from 172.16.0.124 (User2)

- eingehende Antwort beim Client:

Oct 5 00:54:32 camd3: UDP 01 -> 1702:000000:100A:000A 0.286008s from 172.16.0.123 (User1)

Was bedeuten die Pfadangaben?

Die camd3 nimmt an, dass die key, server und user Dateien sich im Ordner var/keys befinden, falls dies jedoch nicht der Fall sein sollte kann man hier den Ort angeben an dem sich die Dateien befinden.

Was bedeutet DNS cache in Sekunden?

Hiermit kann man die Zeit in Sekunden einstellen in welchen Abständen DNS aktualisiert werden soll.

Was ist der Multichannel mode ?

Die camd3 ab Version 3.800 unterstützen den Multichannel mode, dabei kann man z.B. einen verschlüsselten Kanal aufnehmen und dennoch weitere verschlüsselte Sender auf dem gleichen Transponder entschlüsseln. Der Eintrag MULTI aktiviert oder deaktiviert diese Funktion.

Welcher slot steht für was?

dBox 2:

/dev/tts/0 externe com mit Cardreader

/dev/tts/1 Multicam

Ab camd3.807 wird wieder der original Slot der dbox2 unterstützt, dieser ist immer aktiviert, er wird nicht über die Config geregelt. Er kann nicht für CS genutzt werden.

Dreambox:

/dev/tts/0 externe com mit Cardreader

/dev/sci0 unterer Slot

/dev/sci1 oberer Slot

Des weiteren gibt es noch die Möglichkeit sich mit der camd3 am Newcamd Cardserver anzumelden um auf dessen Karten zuzugreifen, dies geschieht mit dem Eintrag:

SLOT=localhost:4:1:1:1:22222:dummy:dummy:0102030405060708091011121314

Warum liest die camd3 meine Original Karte nicht?

Eine Möglichkeit wäre, dass ihr vergessen habt den entsprechenden Slot in der camd3.config zu aktivieren, überprüft also ob die # vor dem Slot entfernt ist.

Was ist das Kommando D0,D2 bzw. D3?

D0 -> G=Global -> Globale Kommandos an alle Karten
D2 -> S=Shared -> Shared Kommandos an die eigene Kartengruppe
D3 -> U=Unique -> Unique Kommandos nur an die eigene Karte

Wie blocke ich um z.B. mein Abo zu verlängern ?

Um die Karte zu blocken müßt ihr das Autoupdate der Karte deaktivieren, dies geschieht in der SLOT Zeile:

```
SLOT=/dev/sci0:2:1:1:1:999:slotunten:password3
```

Ändert dazu die roten Zahlen 1:1:1 in 0:1:0 und die Karte wird nur noch D2 Zeitverlängerungskommandos erhalten und ist somit geblockt.

Falls ihr die Smartkarte im original Schacht der dbx2 verwendet und diese blocken wollt müßt ihr überprüfen ob der Eintrag:

```
DBOX2_AU=0
```

aktiviert und auf 0 gesetzt ist. Damit ist die Karte geblockt.

Welchen Sinn machte es, bei der Multicam und dem com-Port nochmal in G,U und S zu unterteilen?

Weil es eben "Sinn" macht entweder nur D0, D2, D3 oder eben auch "alles" durchzulassen!
So kann man regelt ob man nur Kommandos speziell für seine Karte durchlassen will oder auch Globale Kommandos.

camd3.keys

Beispieldatei von DOZ21

Zitat:

```
#  
# Eine Zeile - Ein Eintrag, der so aussehen muss:  
# [-]AAAA:BBBBBB:CCCCCCCC:DD:[-]E.....  
# oder so:  
#  
# [-]AAAA:BBBBBB:CCCCCCCC:DD:FILE=filename:OFFSET=0xXXXX:SIZE=0xYYYY  
#  
# AAAA - caid  
# BBBBBB - provider  
# CCCCCCCC - ppua  
# DD - keynr  
# EEE..... - data  
# X - offset  
# Y - SIZE  
# wenn '-' vor caid -> wird das Keys aus DB gelöscht(wenn vorhanden)  
# wenn '-' vor data -> wird das Datateil bytesweise rotiert  
#  
# oder SO:
```

```

#
#
# Für OP.keys muss provider 0000000000 sein

# Beispiel Seca
0100:0000A8:0000000000:0C:1234567890ABCDEF
0100:0000A8:0000000000:0D:1234567890ABCDEF
0100:0000A8:0000000000:0F:1234567890ABCDEF

# Beispiel Conax
0B00:000000:0000000000:E0:1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1
234567890ABCDEF1234567890ABCDEF1234567890ABCDEF.....
0B00:000000:0000000000:B0:1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1
234567890ABCDEF1234567890ABCDEF1234567890ABCDEF.....
0B00:000000:0000000000:E1:1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1
234567890ABCDEF1234567890ABCDEF1234567890ABCDEF.....
0B00:000000:0000000000:B1:1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1
234567890ABCDEF1234567890ABCDEF1234567890ABCDEF.....

# Beispiel Nagra
1800:007001:0000000000:00:6FF06F47E957D3C8
1800:007001:0000000000:01:BD422C6AD2B58C24
1800:007001:0000000001:00:... - PK0_E1
1800:007001:0000000001:01:... - PK0_N1
1800:007001:0000000001:02:... - PK2_E1
1800:007001:0000000001:03:... - PK2_E1
1800:007001:0000000001:04:... - PK_N2
1800:007001:0000000001:05:... - EMMK1

# Beispiel Via2
0500:015000:0000000000:08:11223344556677881122334455667780

# Beispiel AES
0500:007C00:0000000000:80:11223344556677881122334455667781
0500:007C00:0000000000:81:11223344556677881122334455667782
.....
0500:007C00:0000000000:8A:1122334455667788112233445566778

# Beispiel BISS AAAA - service id (hexadizimal)
2600:00AAAA:0000000000:00:112233445566

# Beispiel ConstandDW AAAA - service id (hexadizimal)
0100:00AAAA:0000000000:00:11223344556677881122334455667788

```

Die Datei "camd3.keys" muss zwingend !! ins Verzeichnis /var/keys. Grundsätzlich könnte man dort auch einen symbolischen Link anlegen, der auf eine in einem anderen Verzeichnis abgelegte "camd3.keys" zeigt. Spätestens nach der Aktualisierung der Datei durch eine Autoupdate eines Senders wird allerdings der Link gelöscht und die Datei in /var/keys abgelegt.

5.2. Heimnetz-Sharing/Internet-Sharing

camd3.servers

(wird nur in der Clientbox benötigt)

Beispieldatei von DOZ21

Zitat:

```
# cs357x - UDP
# cs378x - TCP
#
# Sende alle ECMs
#cs357x://benutzer1:password1@132.23.78.133:32897
#
# Sende gefiltert nach Services (z.B nur Start und Film)
#
#cs357x://benutzer3:password3:SERVICES=/var/keys/pw_start.services&SERVICES
=/var/keys/pw_film.services@www.myhost.com:16663
#
#
#
# Sende alle ECMs via TCP
#cs378x://benutzer1:password1@132.23.78.133:32897
```

Via LAN:

cs357x:// => UDP-Protokoll
cs378x:// => TCP-Protokoll

benutzer1:password1 => name und passwort zur authentifizierung am Server

:SERVICES=/var/keys/pw_start.services&SERVICES=/var/keys/pw_film.services => Hier ist ein Filter nach Services gesetzt.

@132.23.78.133:32897 => IP und Port des Servers; Client und Server regeln voll automatisch welche Anfragen per CS gesendet werden und welche nicht.

Der eingetragene Port muß identisch sein mit dem Port aus der "camd3.config" der Serverbox.

Die rausgehende Anfragen kann man einschränken, in dem man z.B.
":SERVICES=/var/keys/pw_start.services&SERVICES=/var/keys/pw_film.services"
hinter dem Passwort anhängt, somit lässt er z.B. nur Anfragen raus die vorher in der services Datei eingetragen wurden. Damit kann man manuell den Netzverkehr reduzieren, da z.B. eine Anfrage für einen Kanal, welcher nicht in der Services Datei steht, erst gar nicht an den Server geschickt wird, sondern rein lokal über den passenden Key in der Datei "camd3.keys" beantwortet wird, falls möglich.

Via Internet:

Sieht die camd3.servers genauso aus, man muss nur die Netzwerk-IP durch eine Internetadresse (zB. www.myhost.com) ersetzen.

camd3.users

(wird nur in der Serverbox benötigt)

Beispieldatei von DOZ21

Zitat:

```
# Alle services erlaubt, IN_CASC erlaubt, OUT_CASC erlaubt
benutzer1:password2

# REMM
#benutzer1:password2:REMM=/dev/sci1 (kann auch mit FROM und(oder) SERVICES benutzt werden)
```

```

# nur Start
#benutzer2:password2:SERVICES=/var/keys/pw_start.services

# nur Film
#benutzer3:password3:SERVICES=/var/keys/pw_film.services

# nur Start UND Film
#benutzer3:password3:SERVICES=/var/keys/pw_start.services&SERVICES=/var/key
s/pw_film.services

# Cascadingabfragen von benutzer5 werden akzeptiert und local bearbeitet, wenn es in services steht
#benutzer5:password5:SERVICES=/var/keys/pw_start_in

# Cascading für benutzer6 aktiviert
#benutzer6:password6:SERVICES=/var/keys/pw_start_out.services

#Cascadingabfragen von benutzer7 werden akzeptiert, local bearbeitet und anderen Server weitergeleitet
#benutzer7:password7:SERVICES=/var/keys/pw_start_all.services

# Nur Anfragen von einer bestimmten IP-Adresse sind erlaubt (kann auch mit REMM und(oder) SERVICES
benutzt werden)
#benutzer8:password8:FROM=sein.hostname.tv
#benutzer9:password9:FROM=192.168.1.210

# DISABLE=1 -> Sende Disable-Service, wenn nicht in Service-List (kann mit Anderen Parameters kombiniert
werden)
benutzer110:password10:DISABLE=1&SERVICES=/var/keys/pw_start.services

# DISABLE=2 -> Sende Fake-DW, wenn nicht in Service-List (kann mit Anderen Parameters kombiniert
werden)
benutzer110:password10:DISABLE=2&SERVICES=/var/keys/pw_start.services

```

Hinweis: Bei der dbox2 kan man nur mit multicam bzw. Cardreader und geladenem "multicam.o" Modul einen Server mit Cardsharing betreiben.

Grundsätzlich kann "camd3" auch ohne Cardsharing Server für jegliches Cryptsystem spielen. So ist es denkbar, dass in einem lokalen Netz eine Box Server für andere Boxen spielt, aber keine Karte shared. Dies erspart z.B. Arbeit beim Pflegen der "camd3.keys", da man nur auf einer Box die Datei pflegen muss.

In der camd3.user sind die freigegebenen Clienten, welche mit "camd3" kommunizieren dürfen.

benutzer1:password2 => alle eingehenden Anfragen von den Clienten werden von camd3 angenommen

Filtern mit Service Dateien

SERVICES=/var/keys/pw_start.services => Dem User wird nur erlaubt Sender zu sehe welche in der pw_start.services Datei eingetragen sind.
Alle anderen Anfragen werden von der camd3 ignoriert.

SERVICES=/var/keys/pw_start.services&SERVICES=/var/keys/pw_film.services => Dem User wird nur erlaubt Sender zu sehe welche in der pw_start.services UND der pw_film.services Datei eingetragen sind.
Alle anderen Anfragen werden von der camd3 ignoriert.

REMM - Remote EMM

Über diesen zusätzlichen Parameter in der Datei "camd3.users" wird gesteuert, von welchem Benutzer Auto-Update Daten für die Karte die geshared wird angenommen werden.

Hinweis:

REMM funktioniert zur Zeit nur bei Verwendung des UDP Protokoll, also wundert euch nicht wenn bei TCP keine

Kartenupdates per REMM kommen ;)

Des weiteren funktioniert REMM bis jetzt nur mit Irdeto (und seine Abarten, also Betacrypt oder getunneltes Nagar) Seca und seit 3.812 mit Cryptoworks.

Es gibt folgende Einstellungen:

dBox2:

REMM=/dev/tts/0 => remote EMMs gehen an den seriellen Port = Karte im externen Kartenleser

REMM=/dev/tts/1 => remote EMMs gehen an das Multicam = Karte im Multicam der dBox2

Dreambox

REMM=/dev/tts/0 => remote EMMs gehen an den seriellen Port = Karte im externen Kartenleser

REMM=/dev/sci1 => remote EMMs gehen an den oberen Slot = Karte im oberen Slot der Dreambox

REMM=/dev/sci0 => remote EMMs gehen an den unteren Slot = Karte im unteren Slot der Dreambox

Das Senden von Remote-EMMs muss beim Client nicht extra aktiviert werden.

Bei korrekter Konfiguration liefert der Server nach der ersten Verbindung seine Kartendaten (HEX-Seriennummer) an den Client, dieser filtert anhand der übermittelten Kartendaten die zutreffenden EMMs aus dem Datenstrom heraus und übermittelt die Daten an den Server. Dort werden je nach AU-Konfiguration D0, D2 oder D3 an die Karte weitergeleitet.

Bei erfolgreicher Weiterleitung schickt der Server eine "Quittung" an den Client.

Wenn man die Logausgabe aktiviert hat, sieht man folgende Sequenzen im Log:

"Paket von CMD=05 ..." => beim Client eingehende Kartendaten

"Paket von CMD=06 ..." => beim Server eingehende EMMs

Im Erfolgsfalle protokolliert der Server folgendes:

"Oct 5 01:18:26 camd3: emm D0 auf Slot /dev/tts/1 erfolgreich" (... oder D2 bzw. D3)

Was ist cascading

Wenn ihr einen Server betreibt, welcher eine CW-Anfrage eines Clients nicht beantworten kann (da er die nicht entschlüsseln kann), kann die Anfrage an einen anderen Server weitergereicht werden (der muss sich dann in camd3.servers befinden). Die Anfrage kann maximal 3 mal an einen anderen Server weitergereicht werden. Seit der Version 3.668 wird Cascading User-bezogen gesteuert, d.h die Berechtigungen werden einzelnen Benutzern erteilt

Falls keine weiteren Parameter bei einem User angegeben sind ist Cascading in & out für diesen User aktiviert. Falls man für einen User Cascading regeln will muss man eine Services Datei für diesen Anlegen und dem Usernamen zuweisen s.o.

Das Cascading selber wird dann in der services Datei geregelt.

Services Datei

Beispieldatei von doz21:

Zitat:

```
# AAAA:BBBBBB:CCCCCC:D:E
#
# AAAA caid
#BBBBBB provider
# CCCC service
# D IN_CASC (1-ja, 0-nein)
# E OU_CASC (1-ja, 0-nein)
#
#
#
# Start
```

```
1702:000000:0008:0:0
1722:000000:0008:0:0
```

1702 & 1722 => Sind die ChannelIDs, hier Premiere für Sat und Kabel.

000000 => Ist der Provider

0008 => ist der Services oder anders gesagt die Hex des Kanals. 0008 steht z.B. für Premiere Start. Mit dem Eintrag FFFF als Platzhalter werden alle Services aktiviert, so dass nicht alle Sender einzeln eingetragen werden müssen (seit 3.812)

0:0 => regelt das Cascading In und Out. 1 erlaubt dem User Cascading 0 unterbindet dieses.

Die Anfragen können nicht beliebig oft weitergeleitet werden, da sonst irgendwann der Traffic zu groß wird. Der Server der die Anfrage weiterleitet wird muss einem Zugriff erlauben. (Der Client muss also dort in der camd3.users eingetragen sein).

Weiterhin muss natürlich der weiterleitende Server selbst als Client bei einem anderen Server eingetragen sein.

"IN_CASC" und "OUT_CASC" muss nur bei weiterleitenden Servern hinter dem 1. Server aktiviert sein. Beim 1. Server muss nur OUT_CASC aktiviert sein, da die eingehende Anfrage des Clients eine "normale" Anfrage ohne Cascading-Kennung ist. Beim Client selbst müssen keine Cascading-Parameter aktiviert sein, da dieser ja nur eine Anfrage an "seinen" Server stellt und eine normale Antwort erhält. Beim letztendlichen Server, der die Anfrage beantwortet, muss nirgends mehr "OUT_CASC" stehen, da er nicht mehr weitere Server befragt, mindestens eine Nutzerkennung muss aber "IN_CASC" aktiviert haben, da sonst Kaskadeanfragen nicht bearbeitet werden.

Beispiel:

Ein Client möchte ein Programm sehen, das der Server nicht entschlüsseln kann, dann gibt er (der Server) die Anfrage weiter an einen anderen Server, wo er (der erst angefragte Server) vorher als Client eingetragen wurde. Dieser entschlüsselt dann das Signal und schickt es, über den erst angefragten Server, an den Client zurück.

Verschlüsselung:

Weiterhin wird die Kombination aus Username und Passwort zur Generierung eines Verschlüsselungsstrings genutzt. In den aktuellen Versionen von camd3 wird die Übertragung zwischen Client und Server immer verschlüsselt. Vor allem beim Cardsharing über das Internet ist dies mit Sicherheit von Interesse. Bei der Vorgängerversion musste noch ein Schlüssel "von Hand" eingegeben werden und die Verschlüsselung konnte auch abgeschaltet werden. Wenn die Kennungen auf Client- und Serverseite nicht übereinstimmen ist es nicht nur so, dass der Server wegen fehlender Berechtigung keine Antwort gibt, er kann die Anfrage erst gar nicht "verstehen".

6. Nachtrag/Credits/Dateianhang:

Top 10 der Fehler, die gemacht werden...

1. die neue camd3 getauscht, aber die neuen Config-Dateien nicht ! Z.B. stehen in der camd3.servers der camd3.570 vorne

"cs357x://....", in der älteren camd3.servers aber z.B. noch "cs353x://...." Das muss schon passen !

2. respektive zur camd3.servers muss auch die camd3.users passen wenn ihr euch zu einer anderen Serverbox verbinden wollt. Passend zum obigen Beispiel würde dann dies hier reichen

```
benutzer1:password1
```

Anmerkung: Da muss man nichtmal die Wörter tauschen ! Der benutzer1 kann ruhig auch so heissen, warum nicht !? Dasselbe mit seinem Passwort "password1" 😊

3. Falsch editiert ! Mann sollte drauf achten, dass man grundsätzlich einen Unix-Editor benutzt, und nicht das

"Notepad" von

Windows, oder Word benutzt... Schliesslich läuft auf der Dbox2 ja das Unix-kompatible Operating System "Linux" (Problem tritt de facto nur bei "camd3.users" bei Einträgen ohne Filter auf, aber sollte man aus Prinzip tun.)

4. Image voll ! Sowohl bei JFFS2-Only Images, als auch bei Cramfs/Squash-Images kann ein Tausch der "libc.so.6" und der "libcrypto.so.0.9.7" das Image zu voll werden (ein Problem älterer camd3-Versionen)... Dann geht NIX mehr ! Also mächtig freischaufeln.. Z.B. Spiele rausschmeissen. Bei einem Fehlversuch und nachdem die Message kam "no space left" hilft meist auch kein manuelles entleeren mehr. IdR ist das Image dann komplett futsch und muss erst nochmal neu geflasht werden ! Erst danach neuen Versuch wagen.

5. Nix geht ! Geht mal mit telnet auf die Dbox2 und schaut mal was die Eingabe von "kill -9 camd3", oder "kill -term camd3", oder "killall camd3" ergibt... Wenn da steht, das es nix zum killen gäbe, hat die camd3 niemals richtig gestartet... Ein manuelles Starten der camd3 durch Eingabe von "/var/keys/camd3" kann da schon so manche überraschende Hinweise ausspucken...

6. Alles okay, aber nix wird hell ! Nun, vielleicht vergessen die camd3.keys zu füllen, oder gefüllt zu kopieren ? Übrigens kann man die "camd3.key" und "camd3.key1" nun vergessen; die waren in älteren Versionen... Nun heisst sie "camd3.keys" Dafür kann man aber durchaus die alte camd3.key in camd3.keys umbenennen - das geht (bis auf ein paar Änderungen abgesehen) !

7. Dennoch geht nix ! Rechte vergeben ? Macht doch einfach mal mit telnet in dem Verzeichnis in dem die camd3 liegt "chmod +x camd3", dann kann sicher nix mehr anbrennen 🤖

8. Und immer noch geht nix ! Vielleicht die Keys von einer anderen camd (emu) einfach in die camd3.keys kopiert ? Die hat ihr eigenes Format... Bitte unter Linux, oder im Win mittels Unix-Editor passend um-editieren !

9. Mein unter Linux neu erstelltes jffs2.img oder cramfs.img hat 8257536 Bytes (bzw. typische Grössen für das jeweilige mtd). Warum lüppt es dennoch nicht ? Es kann durchaus sein, dass so eine Datei zwar die passende Dateigrösse hat, aber dennoch zu gross ist (vielleicht wurde bei Erreichen der Grösse abgeschnitten ? Bzw. ein JFFS2-Only braucht um zu funzen ein bissie Rest-Platz zur eigenen Verwaltung... Ca. 500 KByte sollten *wenigstens* frei bleiben !)... Als Tipp hier nur: Ausprobieren !

10. Nix geht bei Cardsharing ! Die Konfiguration richtig gemacht und die richtigen Files in die jeweilige DBox2 geuppt ? Die Datei camd3.users muss auf die Server-Box, die camd3.servers muss auf die Client-Box. Wenn Cascading betrieben werden soll, muss zum Betrieb auch auf eine Server-Box eine camd3.servers und auf eine Client-Box eine camd3.users ! Cascading bedeutet, dass eine Kette von Boxen sich jeweils aushilft, je nachdem welche Box welche Antwort liefern kann... Demnach ist jede Box in dem Verbund sowohl mal Client, als auch mal Server !

Tipp zum Austauschen der libc.so.6 und libcrypto.so.0.9.7 im JFFS2-Only (ältere Versionen der camd3)... Mit FTP in /tmp uppen... Dann mit telnet auf die Dbox2 gehen, und mit "cd /var/tmp" dorthineinwechseln... Dann dort eingeben "mv libc.so.6 /lib"... Dann abwarten !!! Kann sein, dass da sekundenlang *nichts* passiert... Wenn der Prompt zurückkommt, dann wars erfolgreich !

PS: Wer will, kann die Top10 auch gerne in Top20 erweitern 🤖

Moral von der Geschichte´

Beachtet, dass eine Smartcard zwar im Besitz des Abonnenten ist, aber der Provider/Programmanbieter der Eigentümer bleibt. Ihr solltet also wissen was Ihr damit tut. Man kann es allerdings keinem Abonnenten verbieten, "seine" Smartcard mit zu einem Kumpel zu nehmen (der

z.B. auch eine Box hat, aber kein Sport-Abo besitzt) um dort mit der Karte ein Fussballspiel anzuschauen. Etwa ähnlich verhält es sich bei dem Cardsharing. Sowohl bei Heimnetz- als auch bei Internetsharing bleibt die Karte zu Hause im Receiver, und die Verbindung auf Eure Box ist im Grunde nur so eine Art "Verlängerungskabel" auf Eure Smartcard 🤖

Ohne gültiges Abo macht Ihr Euch strafbar !

Gegen ein einzelnes Sharing einer Doku, eines Spielfilms etc. wird keiner was sagen. Wird mit dem Feature des Cardsharings z.B. Brummieri regelrecht *ausgenutzt*, und im übelsten Fall in der halben Strasse für "lau geglotzt", dann entspricht das in keinster Weise den Vorstellung der Autoren des Tools/dieser Anleitung, der Tester und dem Team dieses Boards !

Weniger "schlimm" finden wir es allerdings mit den Emu-Funktionen einen Sender aufzumachen, der in Eurem Land gar nicht abonniebar ist. Das ist zwar trotzdem untersagt, aber 🤖 Die camd3 ist auf keinen Fall eine End-Anwender-Software, sondern ist und bleibt eine Technik-Demonstration, an der stetig entwickelt wird.

Diese Anleitung soll sowohl den erfahreneren User, als auch den Newbee ansprechen. Man kann sich entweder alles komplett durchlesen, oder es auch als "Nachschlagewerk" nutzen, und nur diese Infos entnehmen, an denen man grade knabbert... Viel Erfolg beim Einbau der camd3 !

Credits

Wenn hier irgendwo Fehler enthalten sein sollten, verbessert sie, und gebt Bescheid (an einen der Autoren) ! Es wird dann eingepflegt...

Editiert von limette: [Hier ist die Anlaufstelle... Fragen, Anregungen zu \[HowTo\] Anleitung camd3](#)

Grosses Dankeschön an das Streamboard, und an doz21 für seine unermüdliche Arbeit an der camd3, und an alle, die das Board zu dem machen was es ist.

Gewidmet an: doz21

Erstellt von: imbécil, limette,murphy, tom_63, zor und Just_Me

© 2004 und 2005 by Streamboard

Was ist im Dateianhang enthalten ?

- cramfsck
- mkcramfs
- mkfs.jffs2
- mksquashfs
- camd3-Plugins
- dieses HowTo im .doc, .pdf und .html - Format zum OFFLINE lesen
- ein "Erste-Hilfe-Paket" als HTML-Dokument zur Konfiguration der camd3

Update --> Dateianhang 30.06.04, 23.50 Uhr (waren Verlinkungsfehler im "Erste-Hilfe-Paket"); 26 Downloads bis dahin.

Update --> 03.07.04, 17.00 Uhr - Ergänzung "Was ist debug" unter Punkt 5 (tom_63) ! Entsprechendes Update der Offline-Dokumente im Dateianhang (tom_63) ! Update des "Erste Hilfe-Pakets", ebenso 03.07.04 (zor); 322 Downloads bis dahin..

Update --> Dateianhang 08.11.04, 00.05 Uhr (Updates, in Punkt 5. und Punkt 3. - Edits von imbécil und tom_63, Dateianhang von tom_63); 1672 Downloads bis dahin...

Update --> Dateianhang 29.01.05, 22.00 Uhr; Stand vom 17.01.05 (Updates in Punkt 5. und 6. - Edits von tom_63 - Das HowTo auf den aktuellen Stand der camd3.681 gebracht); 828 Downloads bis dahin...

Ersetzen von Kapitel 5 nach einer Überarbeitung von Just_Me am 22.10.2005 (PDFs folgen noch - TOM)

Ersetzen von Kapitel 5 nach einer Überarbeitung von Just_Me am 27.12.2005 (Anpassen nach Version 3.830)

Stand: Dezember/2005