

# Inhaltsverzeichnis (Stand: So. 15. Apr. 2012)

Neuerungen in <i>dFlash</i> .....	2
Anleitung für <i>dFlash</i> -Installation/Update ( hier: GP3 / Dreambox 8000 ) .....	3
<i>dFlash</i> starten und konfigurieren .....	5
Sichern mit <i>dFlash</i> .....	7
Dreambox mit einem Image flashen .....	10
Das Web-Interface von <i>dFlash</i> .....	11
Deinstallieren von <i>dFlash</i> .....	12
Ausblick ( in nur drei Schritten zu einem aktualisierten Image ) .....	13
<i>dFlash</i> anpassen .....	13
Flashspeicher überprüfen .....	15
Mit <i>dFlash</i> den Speicher aufräumen .....	16
Einen Boot-Log erstellen .....	17
Einschätzungen & Risiken .....	19
Schlussbemerkungen .....	20

Diese Anleitung enthält alle Neuerungen bis zur Version 3.2

## Vorsichtsmaßnahmen

Der Haftungsausschluss)<sup>1</sup> hat durchaus seine Berechtigung. Vorsichtigen Zeitgenossen sei daher vor der Verwendung von *dFlash* die Lektüre der Abschnitte „*Neuerungen in dFlash*“, „*Flashspeicher überprüfen*“ und die „*Schlussbemerkungen*“ ans Herz gelegt.

Mit *dFlash* Vertraute sollten sich mit der Lektüre des Abschnitts *Neuerungen in dFlash* bereits auf den neuesten Stand bringen können.

Der nachfolgend Abschnitt richtet sich vor allem an jene, die *dFlash* bisher nur in älteren Versionen verwendet haben.

---

)<sup>1</sup> disclaimer ☺

## Neuerungen in *dFlash*

DMM benutzt bei einigen Boxen ein neues Dateiformat (NFI2) für die Images, lediglich die DM8000 verwendet wohl noch die alte Version (NFI1). Konnte das alte *writenfi* nicht mit dieser neuen Imagestruktur umgehen, so hat sich dies nun geändert. Zwar stellt *writenfi* noch immer kein sog. *bad block handling* bereit, jedoch werden Sektoren, die als schlecht (etwa durch das Betriebssystem oder DreamUP) gekennzeichneten sind, nicht mehr verwendet bzw. übersprungen. Wer größtmögliche Sicherheit bevorzugt, sollte auf *nfidump* zurückgreifen; hier werden schlechte Sektoren auch erkannt und vermieden, aber es werden nur noch die Bestandteile<sup>2</sup> *root* und *jffs2* des Images gesichert und geschrieben, der SSL<sup>3</sup> bleibt unangetastet.

Bitte beachtet, dass von *nfidump* nicht alle Teile<sup>4</sup> eines Images in den Flash geschrieben werden: Nur noch die *boot.jffs* und *root.jffs* finden den Weg in den Flashspeicher der Box. Dies hat gleich drei Konsequenzen zur Folge:

- Da der SSL derzeit bei der Verwendung von *nfidump* nicht mehr in den Flash geschrieben wird, bedeutet dies, dass die Box stets über den neuesten SSL bereits verfügen muss (derzeit #84 oder höher).
- Will man den SSL erneuern, muss man ein neueres Image mittels WebIF, BIOS/DreamUP flashen oder (das neue in der Version 2.8 vorhandene) *writenfi* verwenden.
- Da *nfidump* sowohl *boot.jffs2* als auch *root.jffs2* aus den NFI-Dateien extrahiert, wird entsprechend temporär recht viel Speicher in */tmp* belegt. So kann bei (zu) großen Images schon einmal der Speicher knapp werden.

Diesem Nachteil steht jedoch auch ein gewichtiger Vorteil gegenüber. Kümmerte sich das alte *writenfi* in keiner Weise um schlechte Sektoren im Flash<sup>5</sup>, so kann dies mit *nfidump* nicht passieren. Dieses Werkzeug schreibt unter Verwendung des Betriebssystems – und somit mit *bad block handling* – in den Flashspeicher.

Zwei weitere Optionen, die Netzwerkressourcen betreffen, wurden eingebaut:

- Um eine Swapdatei auf einer Netzwerkressource anzulegen, muss ein sog. *loop device* fest im Kernel des Betriebssystems eingefügt werden. In den Images von OoZooN ist dieses bereits fest integriert.
- Um von Netzwerkressourcen ein Image flashen zu können, wird das Image vor dem Flashvorgang in den internen Speicher (RAMdisk) kopiert. Dies sollte jedoch nur den Boxen mit kleinen 64MB Flash vorbehalten bleiben. Größere Images, wie sie etwa in der 7020HD und der DM8000 vorliegen könnten dort nicht mehr reinpassen.

Ab der Version 3.3 ist in *dFlash* das Werkzeug *nfidump 4.3* integriert. Es ist nicht mehr in *dFlash* (statisch) enthalten, sondern nur noch verlinkt, wodurch die neue *dFlash*-Version deutlich kleiner geworden ist.

---

<sup>2</sup> Ein Image besteht aus den Teilen *root*, *jffs2* und *SSL* – was wohl Betriebssystemkern, *enigma2* und Bootloader entspricht.

<sup>3</sup> **Second Stage boot Loader**

<sup>4</sup> Eine NFI-Datei enthält am Anfang die Angabe des Imagetyps (NFI1/2) gefolgt von *Second Stage Loader* (SSL), *Boot* und *Root*, wobei die letzten beiden Teile im *jffs2*-Dateiformat (also hoch komprimiert) vorliegen.

<sup>5</sup> Was vor allem bei Boxen, die schon ein paar Jahre auf dem Buckel haben, zu Abstürzen führen kann.

## Anleitung für *dFlash*-Installation/Update ( hier: GP3 / Dreambox 8000 )

Das Vorgehen für Neuinstallation und Update ist identisch! Die Anleitung basiert auf Gemini Project 3 (GP3). Als Skin wurde *dmm-hdr2* verwendet.

### Vorab noch dies ..

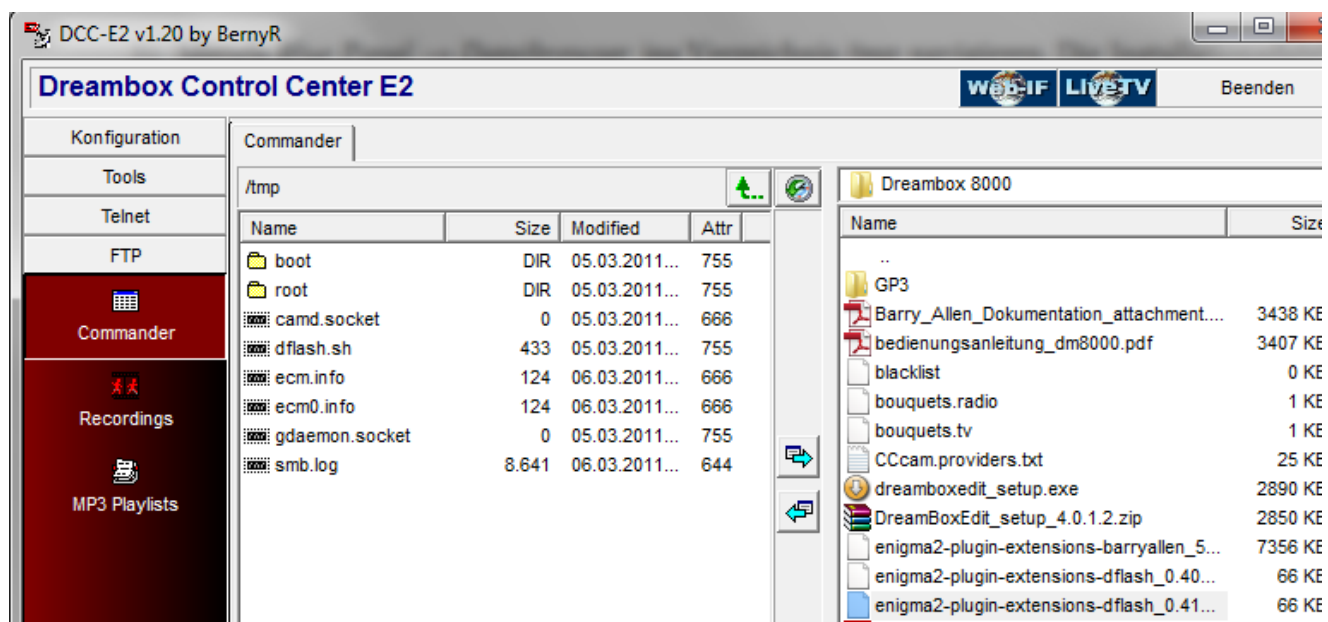
Wer noch kein Jumbo-Image auf seiner Box hat, sollte vorab mittels DMM's WebIf seine Box mit einem dieser neuen, großen Images flashen. *dFlash* kann zwar alte, kleine Images handhaben, aber problemloser Übergang von den alten zu den neuen will *gutemine* nicht garantieren. Hier hilft: ausprobieren! )<sup>6</sup> Derzeit sollte eure Box über Images 3.2 und mit SSL #84 oder neuer verfügen.

Bitte verwendet stets die neuesten Versionen! Bedenkt, dass die Imagefeeds nicht immer aktuell sind. Sehr gute Erfahrungen habe ich bei OoZooN gemacht.

*dFlash* ist derzeit getestet auf 8000/800/800SE/500HD/7020HD/7025)<sup>7</sup> (Stand 8.Dez. 2011; Version 2.0)

### Und jetzt geht's los ..

- 1) Download von *enigma2-plugin-extensions-dFlash\_0.40\_mipsel.ipk* (oder eine neuere Version). Wer's bequemer mag, wählt den Weg über die Online-Installation/Aktualisierung im Blue Panel. ( → Addons → Plugins (extensions) .. )
- 2) Diese Datei mittels FTP ins Verzeichnis */tmp* der Dreambox kopieren. Hier geschieht dies mit dem Dreambox Control Center (DCC E2 1.20).

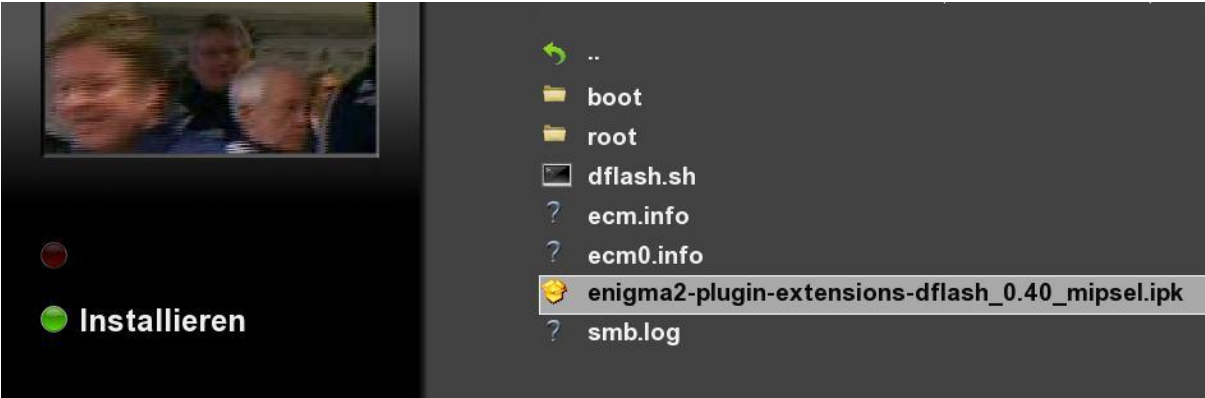


<sup>6</sup>Wer sich den äußerst zeitaufwendigen Weg (Dauer 15-20Minuten) über das Web-Interface von DMM ersparen möchte, kann getrost bereits die *dFlash*-Version 0.40 verwenden. Mir gelang es seinerzeit mit dieser *dFlash* Version problemlos in weniger als vier Minuten aus einem alten, kleinen Image heraus ein neues Jumbo-Image zu flashen.

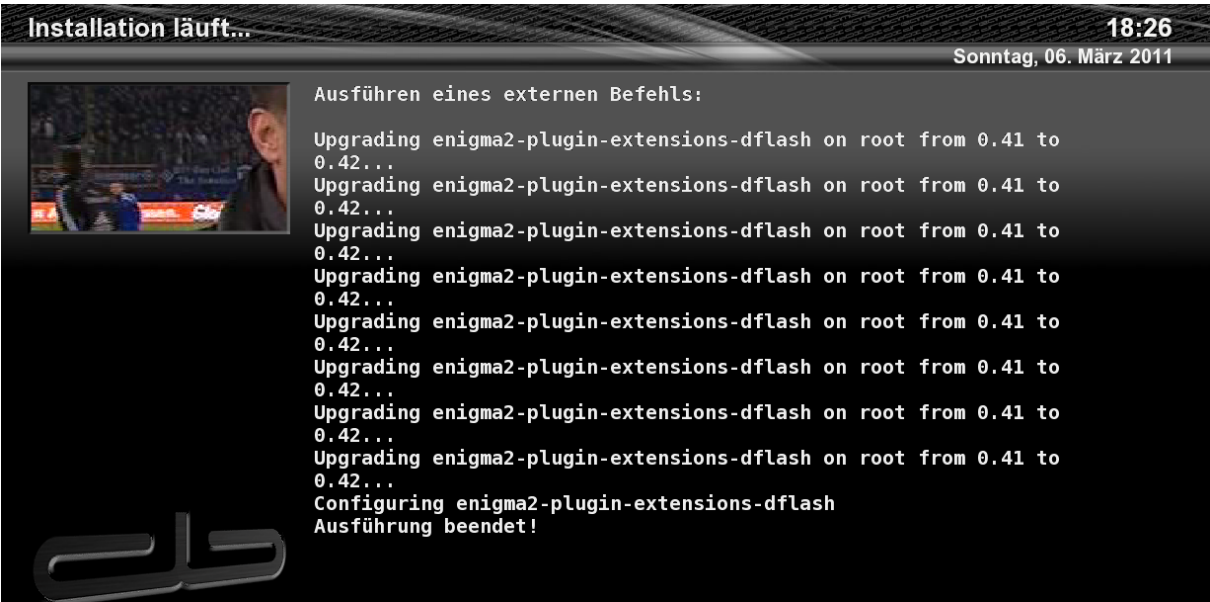
*maxroot* und Jumboimages vertragen sich nicht. Diese neuen, großen Images nutzen ja nunmehr den gesamten Flash-speicher (256MB statt nur 128MB). Wer den ersten Ratschlag befolgt, macht diesbezüglich ohnehin schon alles richtig.

<sup>7</sup> Das Flashen geht derzeit bei der 7025 nur über den Umweg über das WebIf der Dreambox.

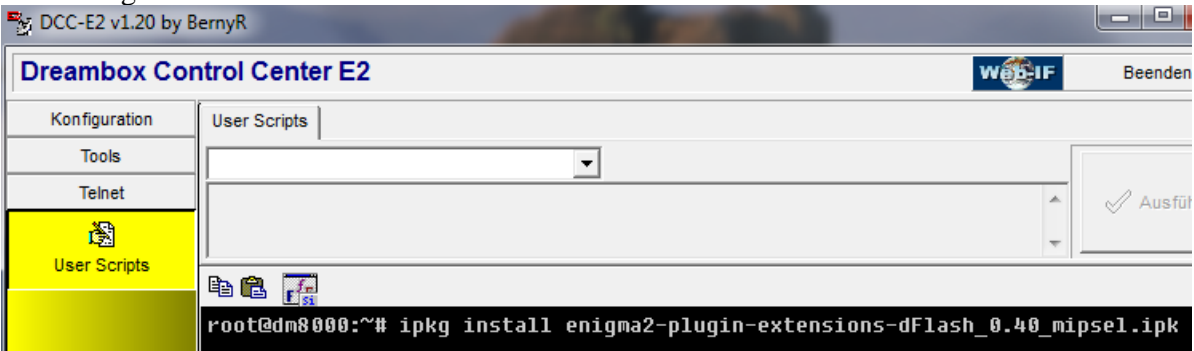
Mittels *Blue Panel* → *Dateibrowser* ins Verzeichnis */tmp* navigieren. Die Installationsdatei von *dFlash* durch Drücken der grünen Taste starten.



Bei einem Update sieht die Meldung nach Abschluss der Installation wie folgt aus:



Alternativ kann man natürlich auch die Installation von Hand vornehmen. Hierzu verbindet man sich über Telnet mit der Dreambox und gibt in der Console folgenden Befehl ein: *ipkg install enigma2-plugin-extensions-dFlash\_0.40\_mipsel.ipk*. Die Abbildung zeigt diesen Vorgang bei Verwendung des DCC. Hierbei ist zu beachten, dass die Installationsdatei sich im eingestellten Verzeichnis befindet, andernfalls ist ggf. der Dateiname um einen entsprechende Pfadangabe zu ergänzen.



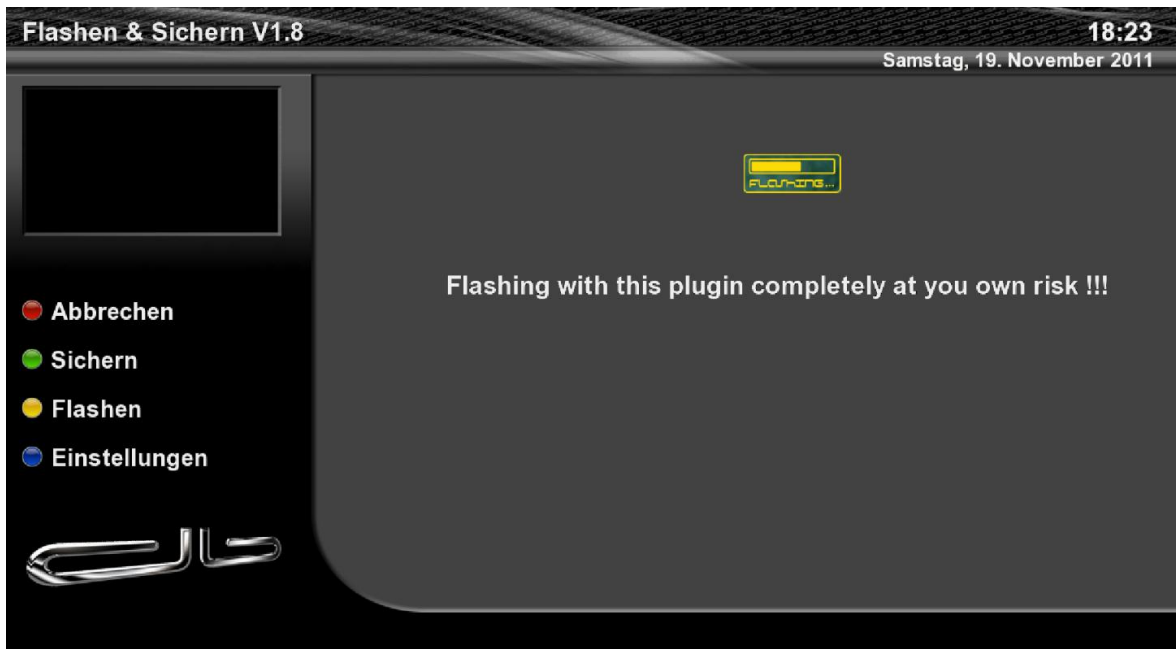
3) Über *Menü* → *Standby/Neustart* die Box neu starten.

## dFlash starten und konfigurieren

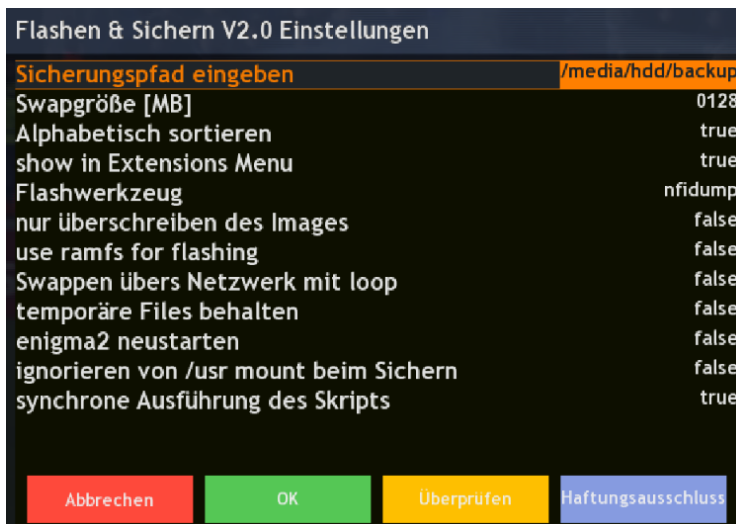
Man kann *dFlash* über ..

- 1) Über *Menü* → *Einstellungen* → *Flashen & Sichern* das *dFlash*-Plugin oder ..
- 2) .. das Extension Menü (falls man dies bereits so konfiguriert hat) ..
- 3) .. oder über das WebIF der Dreambox unter Extras ..

starten. *dFlash* meldet sich wie folgt ..



Vor der ersten Verwendung sollte man *dFlash* zunächst einmal nach seinen Wünschen konfigurieren. Hierzu aktiviert man das folgende Einstellungsmenü, das jedoch nur sichtbar ist, wenn man unter *Menü* → *Einstellungen* → *System* → *Anpassen* → *Benutzermodus = Experte* gewählt hat.



### - **Swapgröße [MB]**

Vorab: Wird das Image mittels *nfidump* in den Flash geschrieben, kann der RAM-Speicher schon mal ausgehen, da dieses Werkzeug die drei Teile (SSL, root, boot) des Images aus der NFI-Datei auspackt und einzeln (außer SSL) in den Flash schreibt. Bei Boxen ohne lokalen Speicher sollte das sog. *loop device* unter *kernel-module-loop\_2.6.18-r10.1\_dm8000.ipk* aus dem GIT geladen und installiert werden. Im OoZooN-Image ist es bereits fest im Kernel enthalten. Dessen Verwendung erfolgt nach Aktivieren in den Konfigurationseinstellungen von *dFlash* stets automatisch.

Die Konfiguration einer lokalen Swapdatei ist je nach Image höchst unterschiedlich aufwendig. Ist die Einstellungen im *iCVS* im *Blue Panel* sehr benutzerfreundlich implementiert, so muss etwa im *OoZooN* hierzu zunächst ein auf dem Feed vorhandenes Script installiert und ggf. angepasst werden.

Optionen: Wird der Wert 0 eingetragen, so wird keine Swap-Datei angelegt. In diesem Fall wird jedoch Enigma2 herunter gefahren, damit *dFlash* nicht das RAM ausgeht. Ist bereits eine mindestens 128MB große Swapdatei vorhanden, so wird diese verwendet. In allen anderen Fällen wird im Zielverzeichnis für das Image eine Swapdatei angelegt. Dies sogar auch auf NAS, wenn das hierzu erforderliche sog. *loopdevice* konfiguriert und aktiviert wurde.

- **Sicherungspfad eingeben**

In diesem Feld kann man (leider nur) mit der Fernbedienung den Sicherungspfad eingeben. Bitte achtet darauf, dass der Pfad auch existiert (z.B. Tippfehler), sonst wird *dFlash* scheitern.

- **Alphabetisch sortieren**

Hat man mehrere Verzeichnisse, in denen Images abgelegt wurden (z.B.: */MB\_Images*, */hdd/backup*, ..) so werden alle Dateinamen in diesen Verzeichnissen als eine sortierte Liste angezeigt anstatt die Dateien Verzeichnis für Verzeichnis nacheinander anzuzeigen.

- **Show in Exensions Menu**

Bei True wird *dFlash* auch im sog. Erweiterungsmenü angezeigt.

- **Flashwerkzeug**

Die Standardvorgabe ist *wirtenfi*. *nfidump* sichert und stellt nur die Teile *root* und *jffs2* eines Images wieder her. Daher muss zum Erneuern des SSL entweder DreamUP, das WebIF des Boot-ROMs, *writenfi* oder einer anderen Methode, welche den SSL in den Flash schreibt, verwendet werden. Wenn man schlechte Sektoren im Flash seiner Box hat/vermutet, sollte man einmal mit DreamUP den Flash beschreiben, weil hierdurch schlechte Sektoren erkannt und als solche markiert werden. *writenfi* überspringt dann diese Sektoren.

- **Nur überschreiben des Images**

Das Image ersetzt nicht mehr komplett den Inhalt des Filesystems *root.jffs2*, das sich im Flash befindet. Es werden im Flashimage der Box nur noch jene Teile überschrieben, die im neuen Image enthalten sind. Alle zusätzlich vorhandenen Elemente im alten Image bleiben unverändert erhalten. Auf diese Weise kann man ein neues Image flashen, ohne sich die vorhandenen Plugins, CAMs, Konfigurationen, .. zu zerstören.

- **Use ramfs for flashing**

Die Option sollte nur<sup>8</sup> fürs Flashen von einer Netzwerkressource verwendet werden, da *dFlash* nach wie vor nicht direkt von einer Netzwerkressource flashen kann. *dFlash* legt eine RAM-Disk an, d.h.: Es wird ein Laufwerk fester Größe im RAM emuliert, sodass der Box garantiert der Speicher nicht ausgehen kann, was beim Kopieren eines (zu großen) Images in ein Verzeichnis – z.B. */var* oder */tmp* – natürlich passieren kann. Hier hinein wird das Image von der Netzwerkressource kopiert und von dort anschließend geflasht. Dieses Vorgehen ist jedoch nur ratsam, wenn die Images klein genug sind, um auf diese Weise ins RAM der Box zu passen. Für Boxen, die über lokale Speichermedien in Form von Festplatten, USB-Sticks, CF, .. verfügen, macht diese Option eigentlich keinen Sinn, kann man doch auf diese externen lokalen Speicher das zu flashende Image vorab kopieren und eben von dort aus flashen.

- **Swappen übers Netzwerk mit loop**

Diese Option ist für all jene interessant, deren Boxen über keine lokalen Speichermedien

---

<sup>8</sup> außer bei einer 500HD, wenn kein USB-Speicher oder eine Festplatte verwendet wird; ansonsten ist hier die Verwendung dieser Option unumgänglich.



verfügen. Hiermit kann nicht nur eine Swapdatei auf einer Netzwerkressource verwendet werden, sie ermöglicht darüber hinaus auch das Flashen von Netzwerkressourcen. Einzige Einschränkung: Die Netzwerkressource muss ein echter Mount sein und kein Link.

- **Temporäre Files behalten**

*dFlash* legt während seiner Laufzeit in im Verzeichnis */tmp* Hilfsdateien an und löscht diese nach seiner Beendigung. Hiermit kann man deren Löschung verhindern. Diese Option entspricht der Eintragung *config.plugins.dflash.keep=true* in den Settings und eignet sich besonders gut für Fehlersuche und eigene Anpassungen.

- **enigma2 neu starten**

Während der Backup-Erstellung wird das Betriebssystem *enigma2* sanft gestoppt (ähnlich wie auch im *Barry Allen*) und nach deren Abschluss erneut gestartet. Geht man direkt nach diesem Neustart in das *dFlash*-Plugin und drückt die grüne Taste, erhält man alle Informationen über die zuvor erstellte nfi-Image-Datei – sozusagen den Ergebnislog. Bestätigt man dessen Anzeige mit *OK*, wird die Logdatei gelöscht und *dFlash* verhält sich beim Drücken der grünen Taste wieder wie gewohnt.

- **Ignorieren von /usr mount beim Sichern**

Gegen seine Überzeugung hat *gutemine* ab der Version 0.92 das Standardverhalten zum Mitsichern des auf einem FE device <sup>9</sup> ausgelagerten */usr* Verzeichnisses von *nein* auf *ja* geändert. Bei der 7025 ändert sich nichts, da hier ohnehin nur der Flashspeicher gesichert wird. Jeder sollte bedenken, dass ein um das */usr* Verzeichnis vergrößertes Image ggf. nicht in den internen Flash passt und *dFlash* bei seiner Ausführung scheitert. Diese Option entspricht dem Eintrag *config.plugins.dflash.usr=true* auf *false* in den Settings.

- **Synchrone Ausführung des Skripts**

Hierdurch wird *dFlash* sozusagen modal ausgeführt; d.h.: Es können keine weiteren Aktivitäten parallel zu *dFlash* ablaufen bzw. gestartet werden. Der aktuelle Betriebszustand bleibt sozusagen solange eingefroren, bis *dFlash* seine Arbeit beendet hat.

## Sichern mit dFlash

Um möglichst allen Schwierigkeiten aus dem Weg zu gehen, sollte ein lokales <sup>10</sup> **Swap-File eingerichtet** sein. Bereits 128MB reichen völlig aus. Wird auf NAS gesichert, muss ein solches File sogar zwingend vorhanden sein.

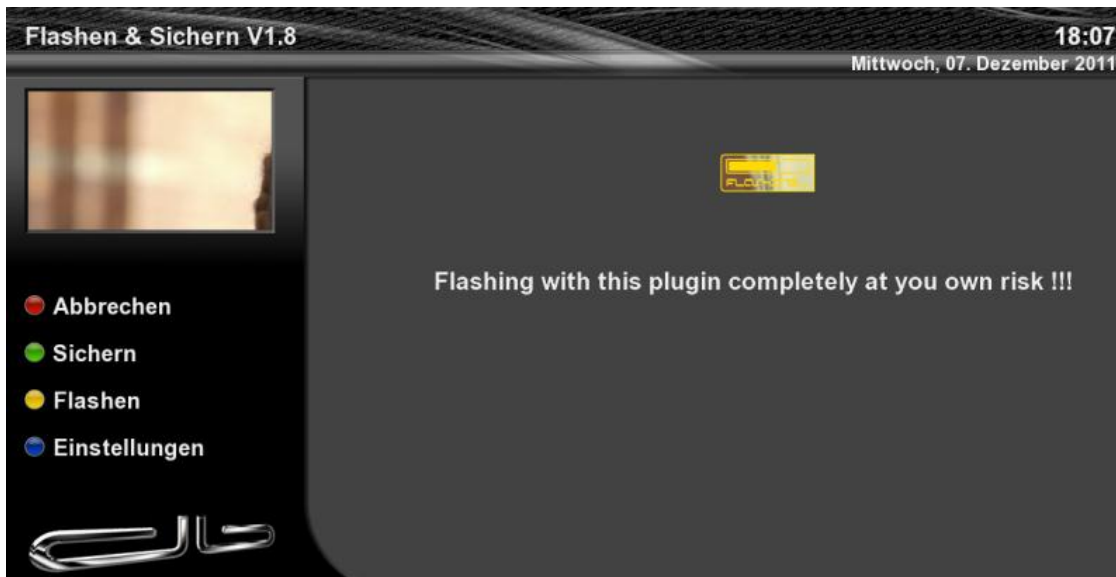
### 4) Über Menü → Einstellungen → Flashen & Sichern das *dFlash*-Plugin starten.



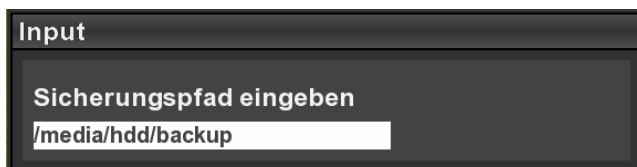
<sup>9</sup> Flash Extender

<sup>10</sup> Eine gute Empfehlung – sofern die Box dies unterstützt – ist, die Swapdatei auf einer externen SD/CF-Speicherkarte oder einem USB-Stick anzulegen, damit die Festplatte schlafen kann, wenn sie während des Betriebs der Box nicht benötigt wird. Dank des sog. *loop device* ist es auch möglich, eine Swap-Datei auf dem Zielverzeichnis des Images auf einer Netzwerk-Ressource anzulegen.

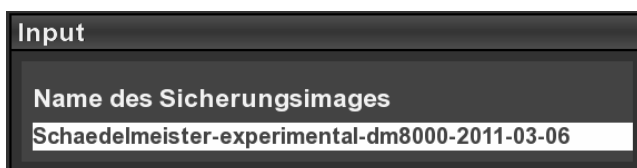
5) Jetzt sollte der Bildschirm so aussehen.



6) Ein Druck auf die grüne Taste leitet die Imageerstellung ein. Nach Wahl des Speicherortes ..



7) .. wird nach dem Namen der Imagedatei gefragt. Dieser sollte 40 Zeichen nicht überschreiten und keine Leerzeichen, [, {, (, .. enthalten.



### **Anmerkung zu Pfadangaben**

Die Pfadangaben dürfen keine Symlinks sein. Welches Verzeichnis als Standardvorgabe von *dFlash* verwendet wird, ist durch die hier vorgenommene Eingabe festgelegt. *dFlash* merkt sich die Eingabe des Pfads. Gibt man in Punkt „6“ einen neuen Pfad ein, so wird beim nächsten Mal dieser als Standardvorgabe für die Ablage der Images verwendet. Auch beim Wiederherstellen eines Images sucht *dFlash* in diesem Verzeichnis. Beim Flashen sucht *dFlash* darüber hinaus standardmäßig zusätzlich in */MB\_Images* und */tmp*.<sup>11</sup>

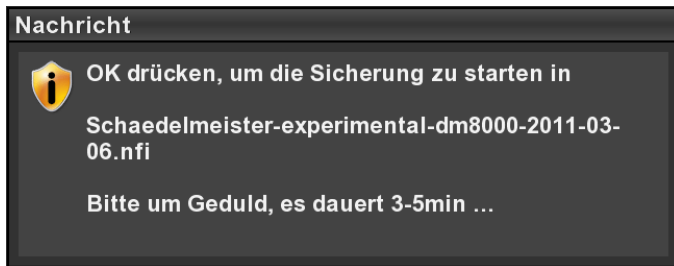
Im Prinzip sind alle (lokalen) Speicherorte als Ziel möglich. Hier hilft Ausprobieren! Selbst NAS ist erlaubt, wenn man ein lokales Swapfile<sup>12</sup> (empfohlen: 256MB) eingerichtet hat. Nur unter Einschränkungen sind Netzwerkressourcen erlaubt, will man das Image auch wieder unter aus demselben Verzeichnis zurückflashen – siehe hierzu die Option *use ramfs for flashing*.

<sup>11</sup> Findet *dFlash* in */tmp* ein Image, so wird nur dieses zum Flashen angeboten.

<sup>12</sup> .. außer bei 500HD. Diese Box kann auch ohne Swapfile eine Sicherung durchführen



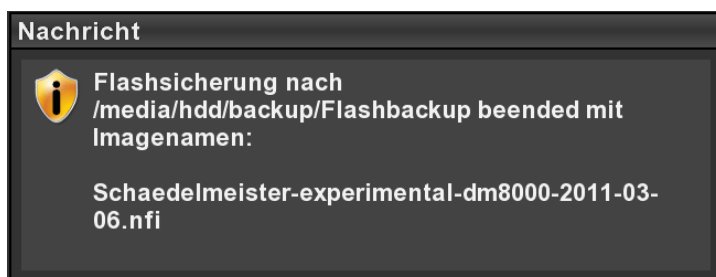
8) Das Drücken der OK-Taste leitet die Imageerstellung ein.



9) Nun erhält man für ca. fünf Minuten folgenden Bildschirm)<sup>13</sup> angezeigt. Die Einblendung einer Fortschrittsanzeige mit dem Untertitel *loading*.. ist nur ein statisches Bild. Wenn nach ..



10) .. dieser Zeit nicht folgende Anzeige erscheint, ..



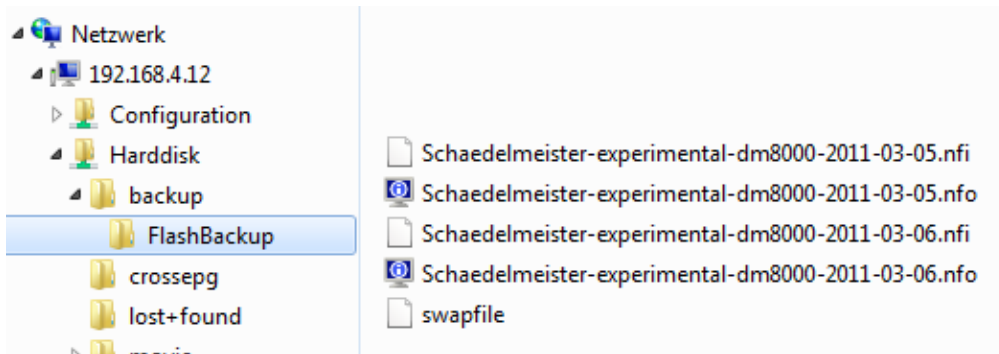
.. sollte man durch Drücken der grünen Taste versuchen abzurufen, ob der Sicherungsvorgang erfolgreich abgeschlossen wurde. Siehe hierzu auch Fußnote )<sup>8</sup>!

Wenn man die oben gewählten Standardangaben für die Imagespeicherung verwendet, so findet man auf der Festplatte der Dreambox die erstellten Images wie hier abgebildet wieder. In dem hier dargestellten Beispiel sind bereits zwei Images vorhanden. )<sup>14</sup>

<sup>13</sup> Wenn *dFlash* kein Swapfile vorfindet und auch temporär keines anlegen kann (etwa auf network mounts), wird man mittels eines Popup informiert, dass – ganz wie im *Barry Allen* – das Betriebssystem der Box *enigma2* sanft heruntergefahren wird. Man sieht die nachfolgend gezeigten Nachricht in diesem Falle natürlich nicht mehr auf dem Bildschirm. Aber keine Panik! Anschließend wird *enigma2* wieder gestartet.

Da man ja nach dem Sicherungsvorgang nicht so recht weiß, ob alles funktioniert hat, sollte man nach dem Neustart von *enigma2* ins *dFlash*-Plugin gehen; hier liefert ein erster Druck auf die grüne Taste Informationen über das gerade zuvor erstellte Image; anschließend arbeitet die grüne Taste wieder wie gewohnt.

<sup>14</sup> Die hier gezeigte Darstellung erhält man jedoch nur, wenn man mittels Samba auf der Dreambox die HDD als *Harddisk* freigegeben hat.



## Dreambox mit einem Image flashen

- 1) Hierzu wählt man mit der gelben Taste die Optionen Flashen.

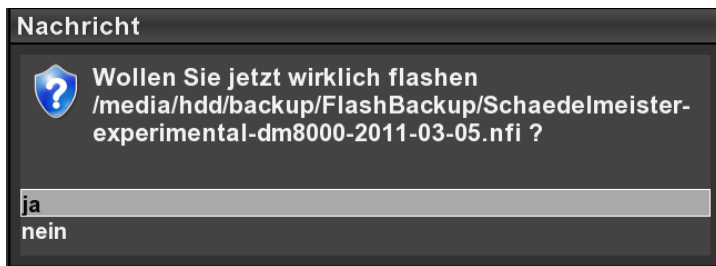


- 2) Es erscheint eine Auswahl der bereits in `/media/hdd/backup/..` <sup>15</sup> vorhandenen Images. Falls *dFlash* jedoch ein Image im Verzeichnis `/tmp` vorfindet, wird nur dieses zum Flashen angeboten.



- 3) Nach Auswahl eines Images erhält man eine letzte Gelegenheit, den Flashvorgang noch abzubrechen.

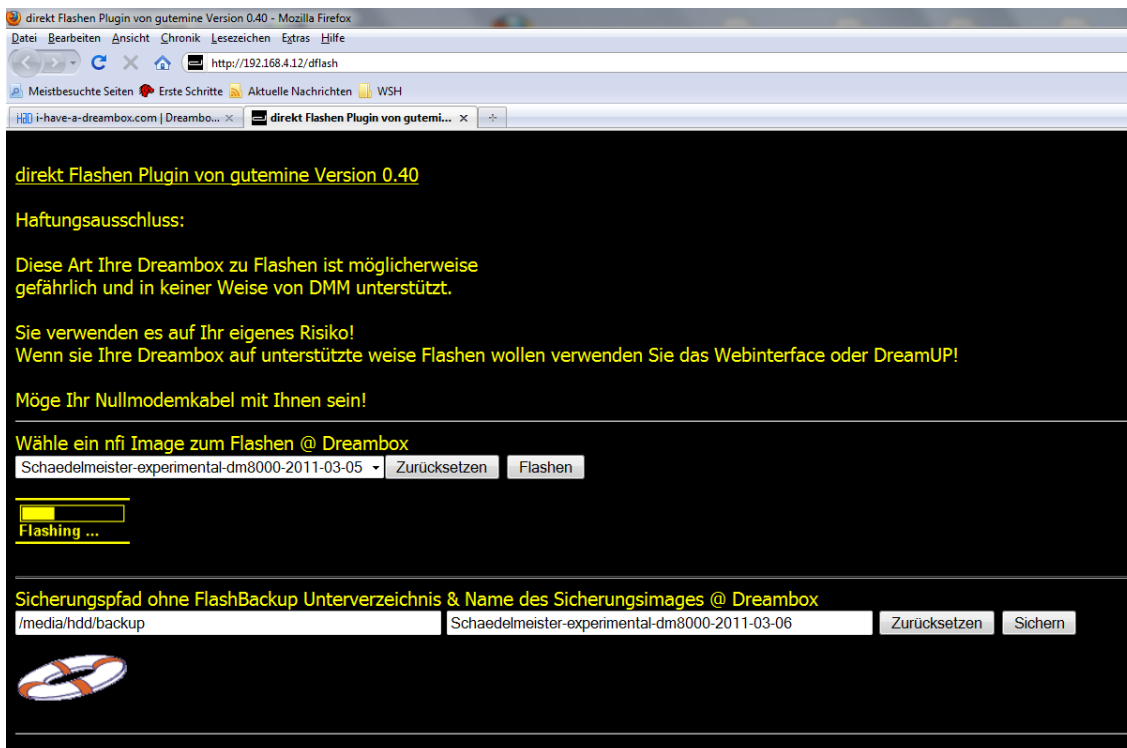
<sup>15</sup> Hier wird der Inhalt des Verzeichnisse angezeigt, in welches dFlash das Image der letzten Sicherung geschrieben hat.



Nach dem Flashvorgang startet die Dreambox neu.

## Das Web-Interface von *dFlash*

Wer's noch bequemer mag, sollte sich mit dem Web-Interface von *dFlash* vertraut machen. Zum Aufruf gibt man im Webbrowser *http://*"IP der Dreambox"/*dflash* ein.



### 1) Image erstellen

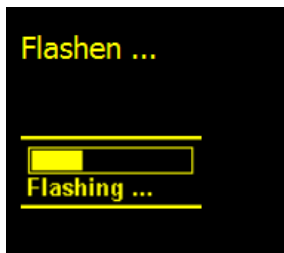
Wählt man einen anderen Dateinamen, so ist darauf zu achten, dass keine unerlaubten Zeichen (etwa Leerzeichen) verwendet werden; auch darf die Anzahl der Zeichen nicht die Grenze 40 überschreiten. Während des Erstellens zeigt der Browser den Text „*Sicherung wird erstellt*“ und einen Rettungsring.



Während des Erstellens kann man die Seite im Browser neu laden lassen. Erst wenn Vorgang abgeschlossen ist, wird wieder die Startseite von *dFlash* im Browser erneut angezeigt.

### 2) Dreambox flashen

Imagedatei auswählen und Flashen drücken lässt den Browser zur Anzeige ..

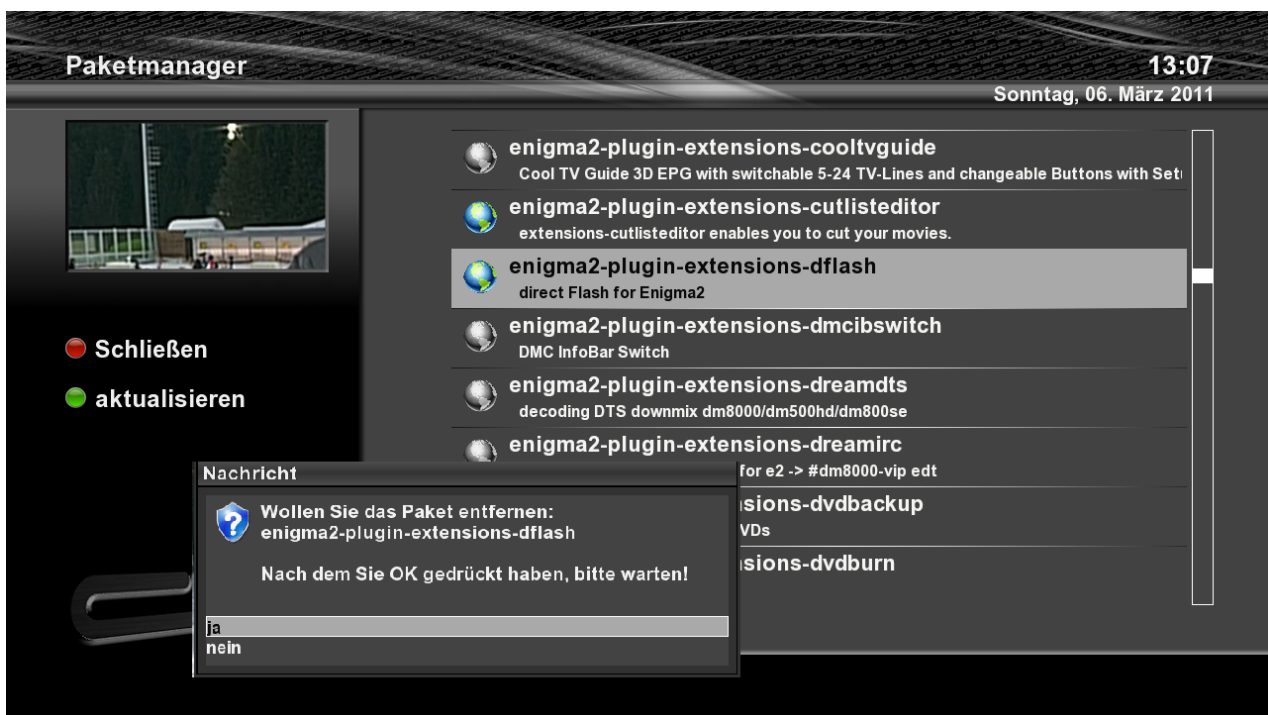


.. wechseln. Wiederum handelt sich bei der Fortschrittsanzeige lediglich um ein statisches Bild. Während und vor allem nach Abschluss des Flashvorganges sollte man nicht auf die Schaltfläche „Seite neu laden“ im Browser klicken )<sup>16</sup>, sondern das Verhalten der Dreambox beobachten, um zu erkennen, wann der Vorgang abgeschlossen ist. Die Dreambox startet nach Abschluss des Flashvorgangs, der zwischen drei und fünf Minuten dauert, neu und es wird der beim Sichern des Images eingestellte Sender wiedergegeben.

## Deinstallieren von *dFlash*

Hierzu gibt es zwei Möglichkeiten. Entweder per Konsolenbefehl oder über die *Paketverwaltung* der Dreambox kann man *dFlash* wieder loswerden.

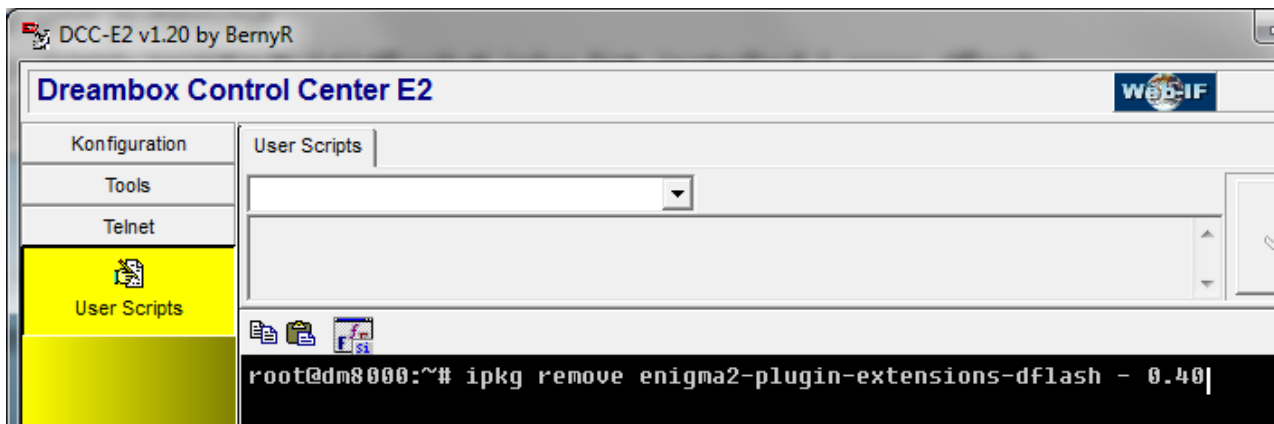
- 1) Über *Menü*→*Einstellungen*→*Softwareverwaltung*→*erweiterte Optionen*→*Paketverwaltung* gelangt man in den sog. *Paketmanager*. Hier muss man den Eintrag *enigma2-plugin-extensions-dflash* suchen und selektieren. Nach Drücken der OK-Taste erhält man eine letzte Möglichkeit die Deinstallation abubrechen (siehe kleines Fenster).



- 2) Per Telnet verbindet man sich mit der Konsole der Dreambox. Hier gibt man den Text „*ipkg remove enigma2-plugin-extensions-dflash - 0.40*“<sup>17</sup> ein, um *dFlash* wieder loszuwerden. In der folgenden Abbildung wird Dreambox Control Center für Enigma 2 (DCC E2 1.20) verwendet.

<sup>16</sup> Wenn man Pech hat, beginnt der Flashvorgang ansonsten von Neuem

<sup>17</sup> Wenn man nicht weiß, unter welchem Namen *dFlash* in Linux registriert ist, kann man mittels *ipkg list\_installed | grep dflash* dies abfragen. Mit dem Filter *grep* beschränkt man die Anzeige auf alle Namen, welche *dflash* als Bestandteil enthalten.



## Ausblick ( in nur drei Schritten zu einem aktualisierten Image )

Die Dreambox auf dem neuesten Stand zu halten ist dank *gutemine's dFlash* mit Unterstützung des *GP3SettingsManager's* <sup>18</sup> recht einfach geworden. In nur drei Schritten erhält man ein aktualisiertes Image. Ob und wie ggf. der *GP3SettingsManager* noch stärker eingebunden kann, sollte man im entsprechenden Support-Thread versuchen in Erfahrung zu bringen.

- In ein fertig konfiguriertes Image, das alle Settings, Plugins, .. enthält, den *GP3SettingsManager* installieren; mit diesem die gewünschten ipk-Dateien aktualisiert vom jeweiligen Feed<sup>19</sup> auf die Festplatte sichern. Die übrigen Einstellungen kann man ja mit standardmäßig mit vorhandenen Boardmitteln (zurück)sichern.
- Neues Image downloaden, ins Verzeichnis */hdd/backup* (oder in ein Verzeichnis deiner Wahl) kopieren und mit *dFlash* die Box flashen.
- Die mit dem *GP3SettingsManager* zuvor gesicherten \*.ipk-Dateien nach */tmp* kopieren und installieren. Nach einem Neustart sind alle Plugins aus dem alten Image schon wieder im neuen Image.

## dFlash anpassen

### Scripting mit dFlash

Wer sich einen Cron-Job zum automatischen Sichern mittels *dFlash* erstellen möchte, sollte sich die Datei */tmp/dbackup.sh* näher anschauen. Diese Datei existiert jedoch nur während Laufzeit von *dFlash* und wird nach dessen Abschluss gelöscht. Man kann nun während des Backup-Vorganges sich diese Datei an einen sicheren Ort kopieren oder aber man setzt in der Datei */etc/enigma2/settings* den Parameter *config.plugins.dflash.keep=true*. Dieser Eintrag verhindert nicht nur das Löschen dieser Datei, sondern darüber hinaus auch aller übrigen von *dFlash* während des Backups angelegten Dateien.

Einfach dieses Script mit Cron planen und schon hat man eine automatische Sicherung.

Es geht noch einfacher: mache ein backup im Webif von *dFlash* und kopiere das, was in der Adresszeile steht <sup>20</sup> für ein *wget -o /tmp/log "http://127.0.0.1/dflash/.."* Aufruf.

<sup>18</sup> Diesen zum Download zu finden, ist wohl nicht immer einfach. Ich habe mich damit beholfen, OoZoon's sehr gut sortierten und aktualisierten Feed zu verwenden. Mit dem Befehl *ipkg list* in Telnet kann man sich eine Übersicht über die auf dem Feed vorhandenen ipk-Pakete verschaffen. Den korrekten Paketnamen kann man mittels *ipkg info <Stichwort+Wildcards>* ausfindig machen. Diese kann man sich schließlich mit *ipkg download <ipk-Paketname>* herunterladen, nach */tmp* kopieren und wie oben beschrieben installieren.

<sup>19</sup> Bitte bedenkt die höchst unterschiedliche Aktualität der Feeds der jeweiligen Imagehersteller!

<sup>20</sup> z.B.: *http://127.0.0.1/dflash?directory=%2Fmedia%2Fhdd%2F&file=Schaedelmeister-exp-dm8000-2011-06-27-16-52&command=Backup*  
*wget -O /tmp/log "http://127.0.0.1/dflash?directory=%2Fmedia%2Fhdd%2Fbackup&file=scripted-backup&command=Backup"*

wget fungiert als eine Art Interpreter, der über die Adresszeile von Internet-Browsern eingegebene (meist Download-)Befehle ausführt. Der Parameter `-o /tmp/log` sorgt dafür, dass die Ausgaben von *dFlash* in die hierdurch spezifizierte Datei geschrieben werden. Die IP 127.0.0.1 ist für ein im Netzwerk befindliches Gerät stets eine Referenz auf sich selbst. Der Dateiname beginnt hier mit *Schaedelman*.. und kann natürlich beliebig geändert werden. Alles Übrige muss bleiben, wie es hier steht.

Und hier ein Script von *dreamon256*, welches das alte Backup löscht, automatisch ein neues erstellt und in *cron* eingebunden werden kann:

```
rm /media/usb/backup/autoback.nfi
rm /media/usb/backup/autoback.nfo
wget -O /tmp/log
"http://192.168.178.26/dflash?directory=%2Fmedia%2Fusb%2Fbackup&file=autoback&command=Backup"
rm /tmp/log
```

Will man Backups aufheben, hilft die Lösung von *allow2010* weiter, welche an den Dateinamen das Datum anfügt:

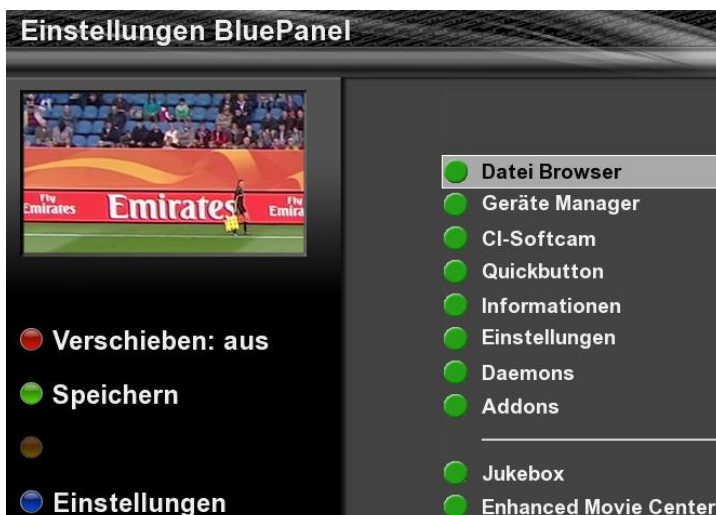
```
"http://127.0.0.1/dflash?directory=%2Fmedia%2Fhdd%2Fbackup&file=backup$(date +%Y%m%d) &command=Backup"
```

### ***dFlash ins Blue Panel und Extensions-Menü integrieren***

Hierzu lädt man sich das (neueste) Installationspaket und installiert es wie gewohnt unter GP3. Nach einem Neustart geht man erneut ins Blue Panel und drückt die <Menü>-Taste (s. Abb.!).



Ein Druck auf die blaue Taste ..



.. ermöglicht das Hinzufügen eines weiteren Plugins ins *blue panel* (BP), indem man ..





.. in der nachfolgenden Anzeige *dFlash* auswählt.

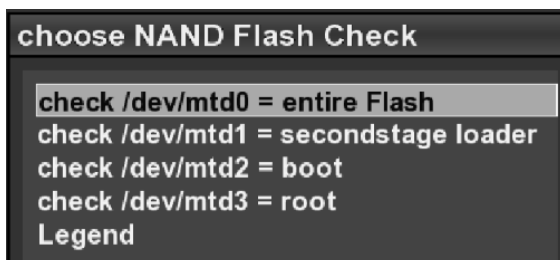
Wenn man alles richtig gemacht hat, kann man im BP das neue Icon für *dFlash* bewundern.



## Flashspeicher überprüfen

Wer *dFlash* benutzt, sollte sich immer wieder darüber im Klaren sein, dass er es mit Bedacht anwenden sollte. Flashspeicher haben eine begrenzte Lebensdauer, und *bad blocks* machen *dFlash* das Leben schwer. Wer möglichst jedes Risiko ausschließen möchte, sollte vor dem Flashen den Flashspeicher auf *bad blocks* überprüfen. Zwar überprüft *dFlash* ab der Version 0.102 den Speicherbereich des *Second Stage Loader* (SSL), aber eben nicht die übrigen Bereiche. Zwar sollte die „Error“ Meldung, die auf einen defekten SSL)<sup>21</sup> hindeutet, beim Neustart sicher vermieden werden, jedoch rät *gutemine*, ggf. bei einer (zu) großen Anzahl an *bad blocks* DMM's *DreamUP* zum Flashen zu verwenden.

*nand\_check* ist ab der Version 0.103 direkt in *dFlash* integriert. Wenn man in *dFlash* auf die Info-Taste drückt, wird das Menü von *nand\_check* eingeblendet, ..



.. in dem man nun die gewünschte Testoptionen auswählen kann. Dauert die – von *dFlash* zu Beginn automatisch durchgeführte – Überprüfung des SSL-Bereichs aufgrund seiner geringen Größe (nur ein Sektor ist komplett ein zweiter teilweise belegt) – nur wenige Sekunden, ..

<sup>21</sup> second stage loader; für diesen Falls hilft nur noch ein Nullmodemkabel ☺

```
Ausführen eines externen Befehls:

Flash type is 4
Block size 131072, page size 2048, 00B size 64
1048576 bytes, 8 blocks

=-.....

Ausführung beendet!
```


.. so muss man für die beiden übrigen Optionen schon mal eine Minute auf deren Abschluss warten. Und wenn die Box einen bösen Block enthält, sieht das Ganze so aus ..

A diagram of a 2D hexagonal lattice. The lattice is represented by a grid of vertices (dots) connected by lines. The top row is labeled 'S' on the left and '=SSSSSSSSSS' on the right. The bottom row is labeled 'B=' on the right. The lattice is composed of horizontal and diagonal lines forming a honeycomb pattern.

Das „B“ weist den schlechten Sektor aus, das nachfolgende Zeichen „=“ beschreibt einen vollbelegten Sektor.

Die Bedeutung der einzelnen Zeichen liefert die Option Legende.

**Nachricht**

-  **B** Bad block
  - . Empty block
  - Partially filled block
  - = Full block, no summary node
- S** Full block, summary node
- Ignore B at the end of Flash

## Mit *dFlash* den Speicher aufräumen

[illegible]

```
Flash type is 4
Block size 131072, page size 2048, OOB size 64
268435456 bytes, 2048 blocks
```

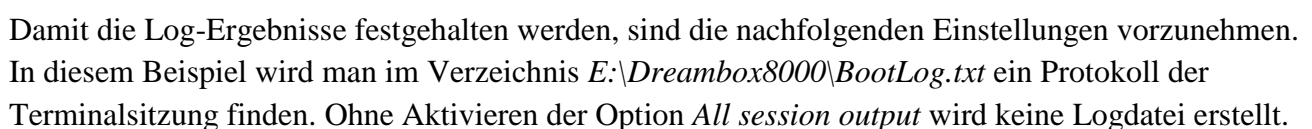
S

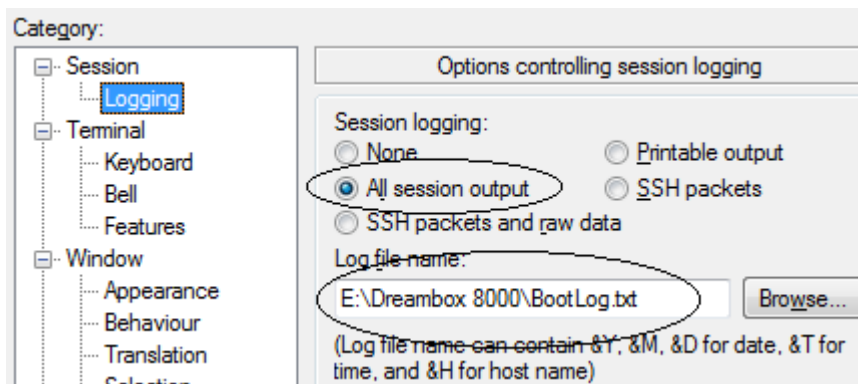
-SSSSSSSSSSS

B=

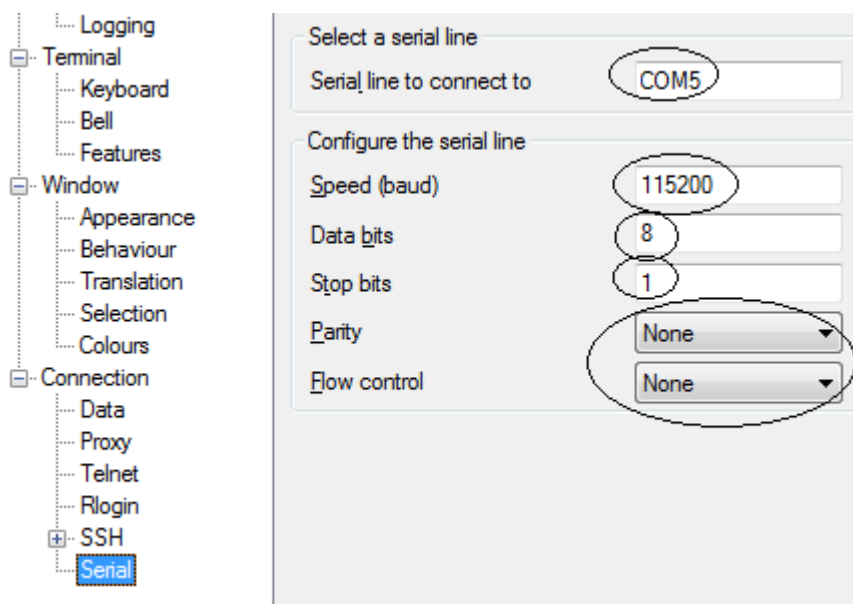
Die Vorbereitungen teilen sich in zwei Arbeitsgänge. Zunächst gilt es, ein Terminal-Programm für den Zugriff auf die Dreambox via serielltem Nullmodemkabel vorzubereiten. Anschließend müssen beim Neustart im BIOS der Box einige Einstellungen vorgenommen werden.

Am Beispiel von *PuTTY* werden die nötigen Einstellungen aufgezeigt. Im ersten Fenster kommt es auf die Eingaben unter *Serial* und *Speed* an; diese sind wie hier gezeigt zu wählen. Sinnvollerweise sollte man sich diese Einstellungen (und alle weiteren!) speichern, was im Abschnitt *Saved Sessions* geschieht.





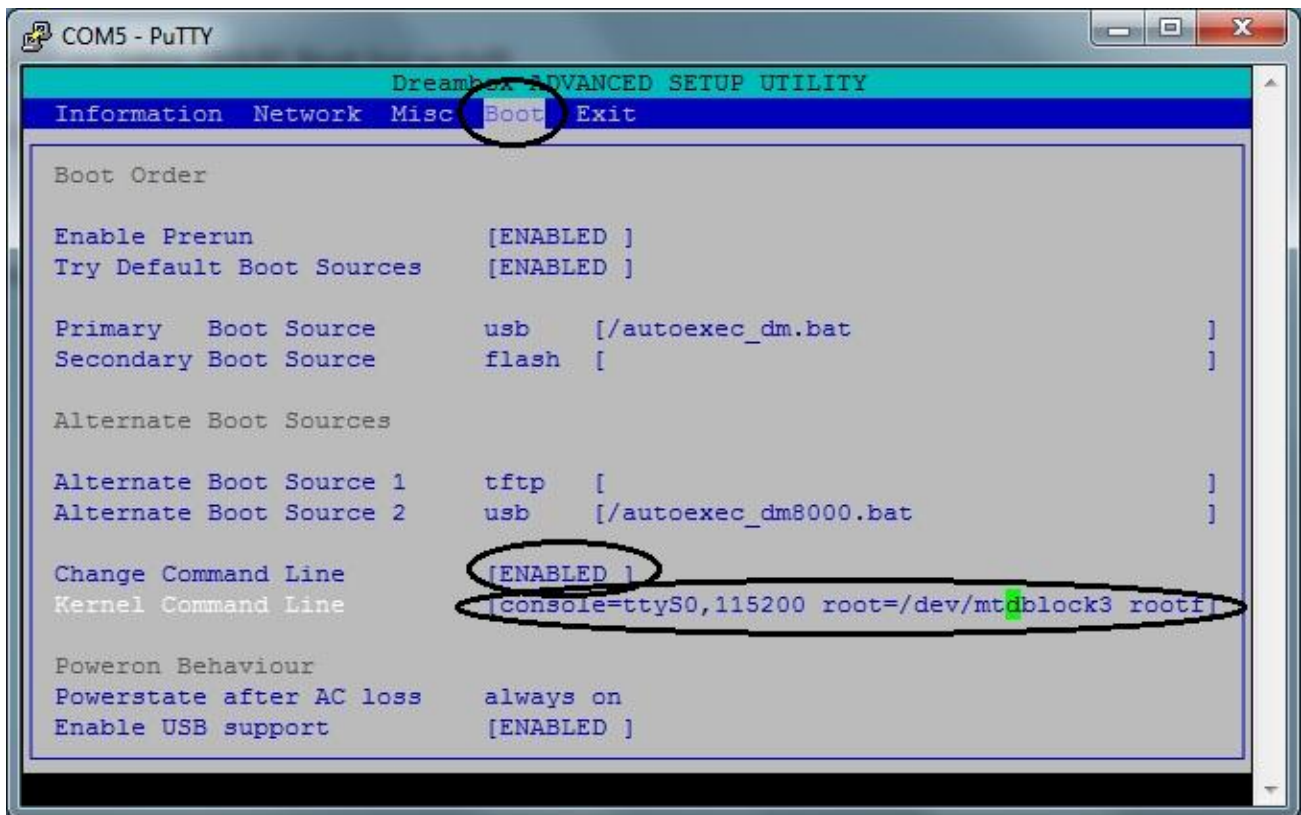
Nun verbleibt nur noch, die erforderlichen physikalischen Verbindungs- und Protokollparameter einzutragen. Die nachstehende Abbildung zeigt diese Einstellungen. Mit Bedacht gilt es den Eintrag im Feld *Serial line to connect to* zu wählen. Hier muss man die richtige COM-Schnittstelle wählen, an der das Nullmodemkabel der Dreambox angeschlossen ist.



### ***Box starten und BIOS-Einstellungen vornehmen***

Bevor die Box gestartet wird, ist das Terminal-Programm zu aktivieren, damit dieses bereits an der seriellen Schnittstelle, an der die Dreambox angeschlossen ist, auf Verbindung wartet.

Sobald die Box mit dem Neustart beginnt, muss man die obere Taste links neben dem OLED-Display gedrückt halten, bis die Meldung >>STOP<< in der Anzeige erscheint. Die Box meldet sich im Terminalprogramm mit der Anzeige des Registers *Information*. Nun gilt es zum Register *Boot* zu wechseln und hierin die Einstellungen *Change Command Line [ENABLED]* und den Eintrage in der *Kernel Cammand Line* auf ***console=ttyS0,115200 root=/dev/mtdblock3 rootfstype=jffs2 rw*** zu ändern.



Jetzt kann es an das Erstellen des Bootlogs gehen. Sinnvollerweise startet man auch die Terminal-sitzung neu, damit ein neuer Log begonnen wird. Nun kann die Dreambox neu gestartet werden. Wenn alles richtig gemacht wurde, kann man nun im Terminalfenster die Ausgaben der Dreambox während des Bootvorganges sehen und in der Logdatei nachlesen.

### Abschlussarbeiten

Für den normalen Betrieb der Box, sollten die Einstellungen im BIOS wieder rückgängig gemacht werden, sonst gibt die Box bei jedem Neustart an der COM-Schnittstelle einen vollständigen Log (und auch alle weiteren Aktivitäten während ihrer Laufzeit) aus. Ohne diese Einstellung gibt die Box bei jedem Neustart lediglich einen verkürzten Log aus und kappt anschließend die Verbindung zu ihrer seriellen Schnittstelle.

## Einschätzungen & Risiken

Nach all dem was ich gelesen und verstanden habe, scheint sich folgendes Bild zu ergeben ..

Das von *dFlash* verwendete *writenfi* führt keinerlei *bad block* Prüfungen durch. Dies hat zur Folge, dass beim Schreiben in einen schlechten Sektor das Image defekt ist.

Die einzige Möglichkeit, dies zu vermeiden, scheint die Verwendung vom *WebIF* oder *DreamUP* zu sein. Es hat den Anschein, dass diese Hilfsmittel das Schreiben in schlechte Sektoren sicher vermeiden.

Die Option *recover bad blocks* versucht durch mehrmaliges Schreiben schlechte Sektoren doch noch mit korrekten Daten zu füllen. Erst wenn dies scheitert, wird der Sektor endgültig als schlecht (bad) gekennzeichnet und fortan von *DreamUP* und dem *WebIF* nicht mehr verwendet.

Wenn man das Pech hat, dass *writenfi* in schlechte Sektoren schreibt, hat dies zunächst die unangenehme Folge, dass das Image an den betreffenden Stellen defekt ist. Je nachdem welche Teile des

Images betroffen sind, zeigen sich höchst unterschiedliche Fehlverhalten der Box. Im schlimmsten Fall bootet die Box nicht einmal mehr.

Wer von derartigen *bad blocks* im Flashspeicher seiner Box geplagt ist, sollte/kann sie sinnvollerweise nur noch mit *DreamUP* und *recover bad blocks* flashen oder sie halt reparieren lassen.

Mein Fazit in der Angelegenheit: Die Ausstattung und Verwaltung des Flashspeichers derart hochpreisiger Boxen ist eine einzige Enttäuschung. Eine DM8000 hat in ihrer Grundausstattung m.E. lediglich das Prädikat ‚Baussatz‘ verdient und kostet dennoch derzeit rund stolze 800€. Umso enttäuschender ist, dass nicht nur an der Ausstattung sondern offenbar an der auch Qualität gespart wurde. In dieser Preisklasse sollte heutzutage ein mit hochwertigeren Speicherchips bestückter Flashspeicher, der auch über Fehlerredundanzen verfügt, selbstverständlich sein.

## Schlussbemerkungen

Seine Box zu flashen ist gewiss für so manchen ängstlichen Zeitgenossen eine Nerven aufreibende Angelegenheit. Anhand der Bildfolgen und Beschreibung kann man sich jedoch vorab über den Ablauf der Vorgänge einen Überblick verschaffen, sodass mit *gutemine's dFlash* und dieser kleinen Anleitung jeder die Angst davor verlieren sollte.

Dennoch sollte man dieses Werkzeug nicht bedenkenlos verwenden. Einerseits wird nun mal diese Art seine Box zu flashen von DMM offiziell nicht unterstützt, andererseits belastet häufiges Flashen die Hardware und lässt sie schneller altern und somit fehleranfälliger werden.

Wer eine DM8000 sein Eigen nennt, kann die hiermit verbunden Gefahren minimieren, indem man *gutemine's Barry Allen* – etwa in Verbindung mit einer CF-Karte – einsetzt. Meine Erfahrungen mit der Kombination Flash: OoZooN stable + beliebige Images in BA habe ich beste Erfahrungen gemacht. Erfahrungsgemäß werden die sog. stabilen Versionen weitaus weniger häufig aktualisiert – das schont den Flashspeicher der Dreambox –, dennoch waren die von BA verwendeten Treiber aus dem Flash-Image stets aktuell genug, meine Konfigurationen fehlerfrei arbeiten zu lassen.

So lautet trotz alle Bedenken mein Fazit: Wer's jetzt immer noch nicht schafft, seine Dreambox immer auf dem aktuellsten Stand zu halten, der hat sie eigentlich erst gar nicht verdient. 😊

Wysiwyg

---

---

---



## Anhang

Hier stehen all jene Beiträge aus dem *dFlash*-Support-Thread, die noch nicht in die Anleitung aufgenommen wurden.

-----

Was solls, im Anhang ist ein dummy SSL 84 Paket für die 7025 damit das Gejammere endlich aufhört.

Im Prinzip ist das natürlich kein echter loader 84 sonder indem man das ipk installiert repariert man einfach den ssl Eintrag in der opkg status database.

Wenn DMM nicht bereit ist es zu fixen habe ich auch kein schlechtes Gewissen dabei sowas zur Verfügung zu stellen !

Also nichts mir herumeditieren, einfach das ipk aus dem Anhang VOR dem BA 6.0 kit installieren und schon sollte Ruhe sein mit allen Plugins die Loader 84 als Voraussetzung haben.

Bitte auf der 7025 testen und berichten ob es funktioniert. Sollte DMM je einen loader 84 für die 7025 rausbringen (was wahrscheinlich nicht der Fall sein wird weil die alten 7025 ja keinen security chip haben) können sie den dann gerne 84a taufen. Indem sie es monatelang nicht gefixed haben denke ich haben wir auch ein moralisches Recht zur Selbsthilfe zu greifen.

Insofern kann man das ipk sogar auf einen Software Update feed machen damit es automatisch installiert wird sobald die Abhängigkeiten anderer pakete es verlangen, aber das muss OoZooN entscheiden.

LG  
gutemine

EDIT: Anhang wurde entfernt, weil der ist jetzt auf der ersten Seite

-----

BA 6.0 hat im ipk als abhängigkeit das der Loader mindestens Version 84 haben soll. Der aktuelle Loader der 7025 ist aber 83, womit eigentlich die 7025 nur mehr BA 5.\* verwenden kann/soll.

ABER nachdem die loader Version auf der 7025 in aktuellen images sowieso NICHT in der opkg status database steht und DMM scheinbar nicht bereit ist das zu fixen sollte man das sowieso reparieren damit man wenn man ein Plugin mit Loader Anhängigkeit installiert dieser nicht nachinstalliert wird obwohl eh schon vorhanden, weil das meistens nicht gut geht und man dann mit DreamUP seriell flashen muss.

Dafür ein Auszug aus dem Dumbo Wiki:

Paketverwaltung austricksen

Wir gaukeln jetzt der Paketverwaltung vor, dass der benötigte Second Stage Loader schon installiert ist.

Zuerst erstellen wir die Datei dreambox-secondstage.list mit folgendem Befehl.

touch /usr/lib/opkg/info/dreambox-secondstage.list

Nun müsst ihr noch die Datei **/usr/lib/opkg/status** editieren und am Ende folgenden Eintrag machen.  
Benutzt dazu bitte einen unixkonformen Editor (z.B. Notepad++).  
Wobei ich den vi Editor direkt auf der box verwende.

Package: dreambox-secondstage  
Version: 83-r0a  
Depends: dreambox-bootlogo (>= 7.0)  
Provides:  
Status: install ok installed  
Architecture: dm7025  
Installed-Time: 1315676295  
Auto-Installed: yes

Na ja und im Prinzip kann man hier auch lügen und einfach 84 als Version hinschreiben, dann installiert auch BA 6.0 und sollte auch wie gewohnt funktionieren.

Einfacher ist aber du verwendest das dummy loader ipk aus meiner nächsten Reply :-D

LG  
gutemine

-----

Bei Problemen ..

config.plugins.dflash.keep=true bei mit init 4 gestopptem enigma2 in die /etc/enigma2/settings machen  
dann bleibt dbackup.sh über und du kannst es mit sh -x /tmp/dbackup.sh ausführen und den output hier  
posten

☺