

# PR-HD1000-Secrets

by Colibri

27. April 2007 / Version 1.02

Die neueste Version gibt's hier: <http://colibri.de.ms>  
colibri\_dvb@lycos.com

Der PR-HD1000 ist ein Premiere zertifizierter HDTV-Receiver von Humax.

Das Einspielen selbst entwickelter Software oder das Ändern der Programmliste über die serielle Schnittstelle auf der Receiverrückseite ist nicht möglich da Humax „vergessen“ hat entsprechende Tools oder Whitepapers die das Übertragungsprotokoll beschreiben öffentlich zur Verfügung zu stellen. Schlimmer noch, sie haben sogar den Flash-Baustein im Inneren so mit Harz verklebt, so das ein Auslöten und ein externes Programmieren auch nicht möglich ist.

Wenn man einfach versucht den Programmlisteneditor „Formula For NEO“ der eigentlich für andere Humax Receiver gedacht ist zu verwenden wird man enttäuscht. Zwar kann mit dem Tool die Programmliste aus dem Receiver gelesen, angezeigt, geändert und als Datei auf dem PC abgespeichert werden, jedoch schlägt ein Zurückspielen auf den Receiver fehl.

Durch Reverse Engineering ist es mir gelungen ein eigenes Tool „PR-HD1000-Heaven“ zu schreiben das ohne öffnen des Receivers nur über die auf der Rückseite vorhandene serielle Schnittstelle folgendes kann:

- Vollständiges Auslesen des 4 MByte Flashes, des 8 kByte EEPROM und der UniqueID des Receivers
- Ändern des kompletten EEPROMs und Teile des Flashes die nicht schreibgeschützt sind (z.B. ist der Bootloader geschützt die Programmliste oder Stringressourcen hingegen nicht)

Alle Daten die zum Receiver übertragen werden müssen signiert (digital unterschrieben) sein, damit er sie akzeptiert. Durch eine (vergessene) Backdoor im Receiver ist es jedoch möglich durch senden einer speziellen Bytefolge die Signaturprüfung zu umgehen, dadurch ist das Beschreiben mit eigenen Daten überhaupt erst möglich. Man sollte sich also überlegen ob man ein Over-The-Air firmware update durchführt, denn es könnte ja die Backdoor schliessen.

In den folgenden Kapiteln ist nun einiges beschrieben das ich herausgefunden habe und auch ein paar Erklärungen zum PR-HD1000-Haven Tool.

## **Neues in der Version 1.02:**

- Settingeditor ist integriert. Man kann jetzt seine Programmliste am PC nach belieben sortieren. Das Kapitel 7 „Programmliste des Receivers am PC ändern“ wurde deswegen komplett überarbeitet.
- Das Speichern vom Receiver gelesene Daten in eine **HDF** Datei war bei der 1.01er Version kompliziert. Ab der Version 1.02 genügt nach dem Lesen „Save HDF file“ anzuklicken.

## **Neues in der Version 1.01:**

- Testmode aktivierung
- Ausführen von eigenen Code ist jetzt durch einen zweiten Bug (Testmode-Bug) möglich.

1	Gültigkeitsprüfung der Premiere Seriennummer .....	2
2	Over-The-Air (OTA) update .....	3
3	Komplettes Flash und EEPROM Backup vom Receiver .....	4
4	Humax-Download-File (HDF) .....	5
4.1	Zerlegen eines HDFs .....	6
4.2	Zusammenbauen eines HDFs .....	6
4.3	Gültig machen eines vorhandenen HDFs .....	7
5	(De)komprimierung des Flashes .....	8
6	String Resource .....	8
7	Programmliste des Receivers am PC ändern .....	10
7.1	Programmliste von Receiver zum PC übertragen .....	10
7.2	Programmliste am PC bearbeiten .....	11
7.3	Programmliste vom PC zum Receiver übertragen .....	12
8	Memory Map .....	13
9	Troubleshooting .....	14
10	Testmode .....	15
11	Ausführen von eigenen Code .....	16

## 1 Gültigkeitsprüfung der Premiere Seriennummer

Die Premiere Seriennummer ist 14 Ziffern lang. Die letzten beiden Ziffer sind Prüfziffern. Zur Berechnung der Prüfziffern teilt man die Nummer in drei Bereiche auf

1	2	3	4	5	6	7	8	9	10	11	12	13	14
Nummer1										Nummer2		Prüfnummer	

$\text{Prüfnummer} = ((\text{Nummer1} \bmod 23) + \text{Nummer2}) \bmod 100$

Beispiel:

123422443399xx

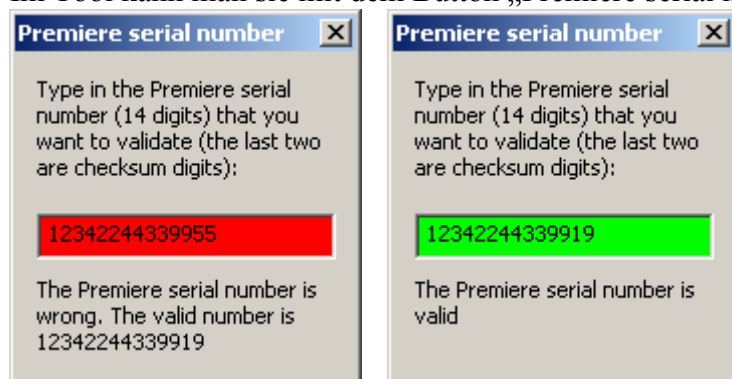
$1234224433 \bmod 23 = 20$

$20 + 99 = 119$

$119 \bmod 100 = 19$

Die Prüfnummer ist also 19. Die gültige Premiere Seriennummer ist somit 12342244339919.

Im Tool kann man sie mit dem Button „Premiere serial number“ berechnen lassen.



Die Premiere Seriennummer ist ab Offset 100 hex im Flashdump zu finden und sollte überschrieben werden bevor man ihn irgendwo ins Internet stellt.

## 2 Over-The-Air (OTA) update

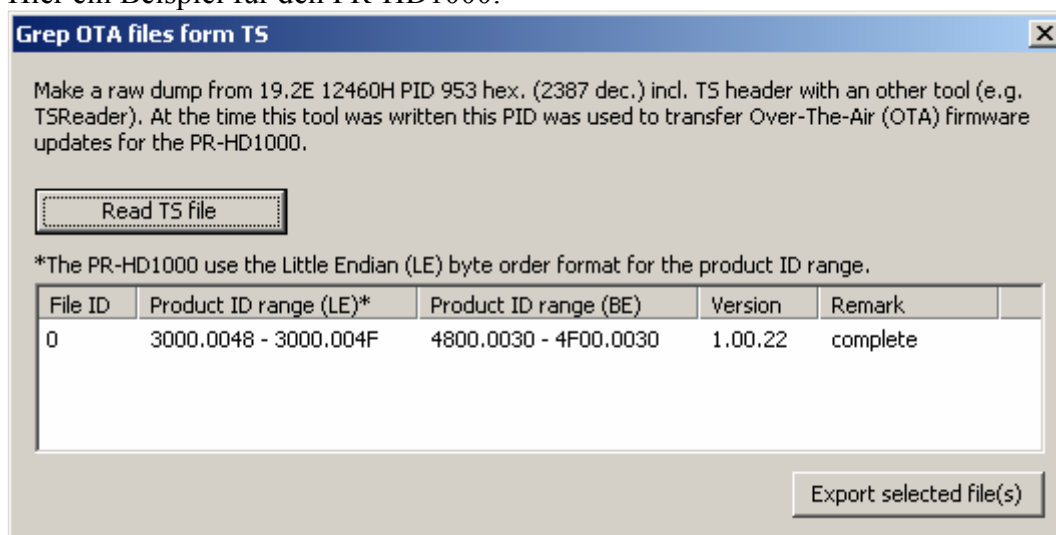
Humax stellt für den PR-HD1000 keine Firmware updates als Dateiform zur Verfügung. Vermutlich deshalb damit man nicht einfach eine alte Version einspielt, wenn über ein OTA-update ein bug gefixt wird. Beispielsweise gab es mal in einer alten Version den Bug das der Receiver bei einem durch HDCP geschützten Film den YUV-Ausgang nicht abgeschalten hatte. Einige Besitzer eines nicht HD-ready Beamer würden gerne diese alte Version einspielen. Die einzige Ausnahme die (versehentlich) public wurde ist die Version 1.00.13 („prhd1000\_10013\_ssd1.hdf“), aber bei dieser war der Bug bereits behoben.

Zwar könnte man mit einer DVB-Karte für den PC die OTA-updates ja einfach loggen, das Problem ist aber aus den aufgezeichnetem Stream die HDF-Datei zusammenzusetzen. Die Datei ist nämlich nicht als Ganzes zu finden sondern in Hunderten von Einzelteilen im Stream verteilt. Nach dem mühsamen loggen und analysieren von diversen Download Frequenzen (die in den Foren Gepostete war leider falsch bzw. für ein anderes Humax Model) habe ich sie endlich gefunden und dem Tool eine Funktion hinzugefügt die aus dem aufgezeichnetem Stream die HDF-Dateien herauslösen kann.

Satellit	Frequenz	Pol.	Symb. Rate	PID (hex.)	PID (dez.)	Bemerkung
Astra 19,2° Ost	12460 MHz	H	27500	953	2387	<b>PR-HD1000</b>
Astra 19,2° Ost	12148 MHz	H	27500	969	2409	iPDR-9800
Astra 19,2° Ost	12670 MHz	V	22000	141	321	CR-FOX+ und Andere
Astra 19,2° Ost	12604 MHz	H	22000	4C0	1216	
Astra 19,2° Ost	12070 MHz	H	27500	972	2418	PR-HD1000C

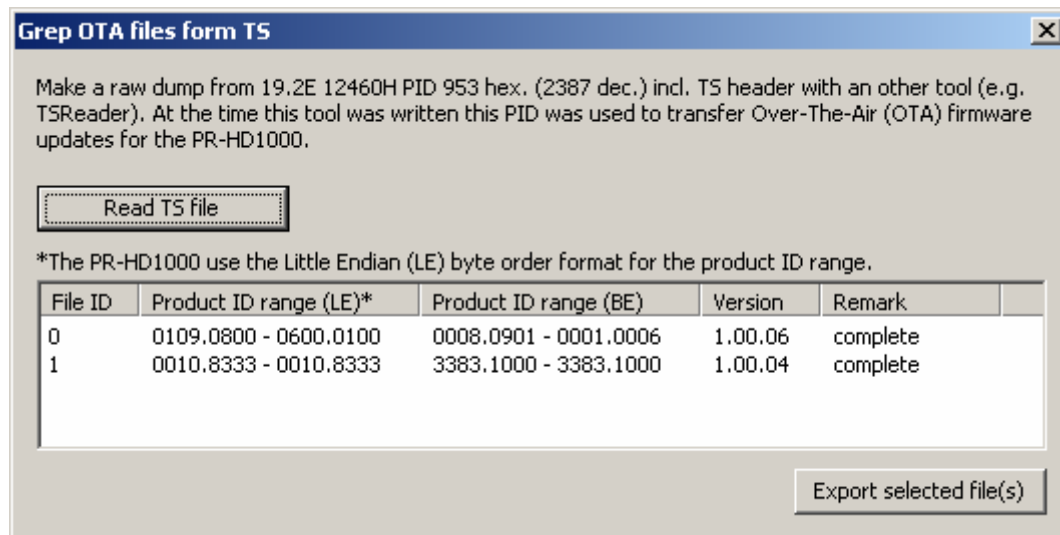
Es ist möglich das sich die PID ändert, auch können auf einer PID Version für verschiedene Modelle ausgestrahlt werden.

Hier ein Beispiel für den PR-HD1000:



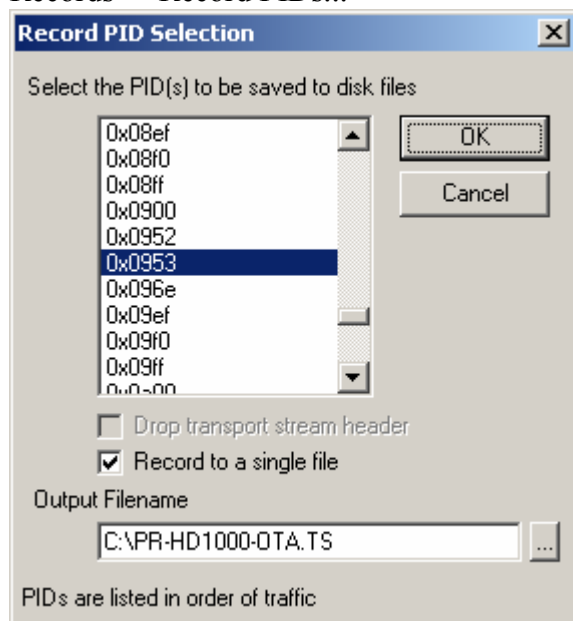
Damit der Receiver entscheiden kann ob das Update für ihn bestimmt ist prüft er ob seine Produkt ID im Produkt ID Bereich der mit dem Update mitübertragen wird liegt. Der PR-HD1000 verwendet die Little Endian (LE) Spalte.

Hier ein Beispiel von einer anderen Frequenz (12670V, 22000, PID 0141 hex.) das für andere Modelle gedacht ist:



Zum Erzeugen des benötigten RAW dump habe ich das Tool TSReader verwenden und die [WinTV-NOVA-S PCI](http://www.hauppage.com) von <http://www.hauppage.com> .

Records -> Record PIDs...



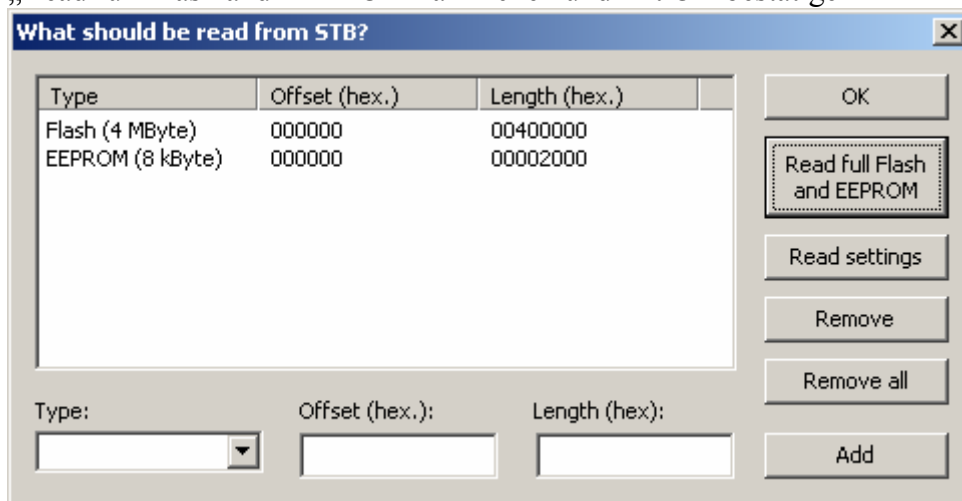
Am Besten solange laufen lassen bis man 10 MByte zusammen hat. Dann im PR-HD1000-Heaven „OTA extractor“ und „Read TS file“ anklicken. Anschliessend die Zeile markieren und „Export selected file(s)“ anklicken um die HDF-Datei zu speichern. Falls die gewünschte Zeile den Remark „incomplete“ enthält dann muss man nochmal loggen. Am Besten wartet man diesmal bis die Datei 20 MByte gross ist.

### 3 Komplettes Flash und EEPROM Backup vom Receiver

Bevor man das Flash/EEPROM im Receiver ändert, sollte man unbedingt ein Kompletbackup durchführen. Das kann man wie im Folgenden dargestellt durchführen:

- Den Receiver und den PC mit einem Nullmodemkabel (<http://de.wikipedia.org/wiki/Nullmodem-Kabel>) verbinden
- Im Feld „STB is connected to“ den COM-Port einstellen an dem der Receiver angeschlossen ist

- „Read from STB“ anklicken
- „Read full Flash and EEPROM“ anklicken und mit OK bestätigen



- Den Receiver über den Netzschalter an der Rückseite aus- und wieder einschalten (das Schalten in den Standby und anschliessendes Einschalten über die Fronttaste genügt nicht).
- Nach ein paar Sekunden beginnt der Transfer von Receiver zu PC. Die geschätzte Restzeit wird im Tool angezeigt. Der Transfer zum PC dauert protokollbedingt sehr lange, da jedes Byte bestätigt werden muss bevor das nächste gesendet wird. Das Kompletbackup wird also über eine Stunde in anspruch nehmen (die Gegenrichtung z.B. beim Einspielen einer Programliste geht dabei rasend schnell da für einen grossen Block eine Bestätigung ausreicht).
- Den Transfer kann man jederzeit gefahrlos durch einen Klick auf „Cancel“ abbrechen.
- Es kann manchmal auch zu einem Abbruch ganz am Anfang des Transfers kommen (mittendrin ist es mir noch nie passiert), dann einfach nochmal mit „Read from STB“ anfangen
- Wenn der Transfer abgeschlossen ist „Save HDF file“ anklicken.

## 4 Humax-Download-File (HDF)

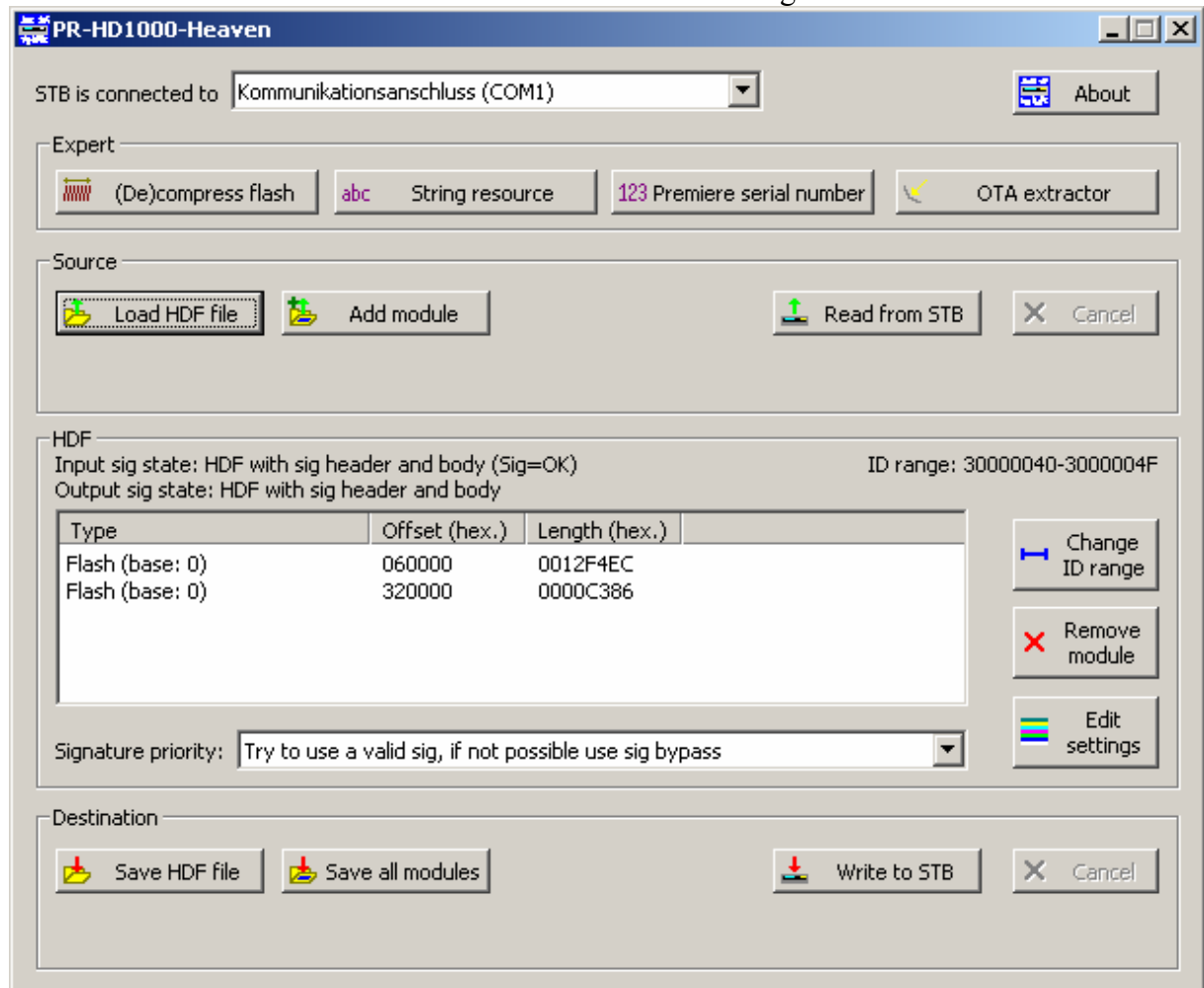
Die Daten die zum Receiver geschickt werden sollen müssen vorher in das Humax-Download-File (HDF) Format konvertiert werden. In der HDF Datei sind neben den Nutzdaten zusätzliche Header vorhanden die angeben wohin diese Daten geschrieben werden sollen (z.B. ins Flash oder ins EEPROM). Ausserdem ist am Header erkennbar ob die Nutzdaten unverändert geschrieben werden sollen oder ob die Nutzdaten im HDF komprimiert vorliegen und deshalb vor dem Schreiben entpackt werden müssen. Eine HDF Datei muss auch mit einem speziellen RSA-Private-Key signiert sein damit sie von Receiver angenommen wird. Der Receiver prüft dann die Unversehrtheit der Daten mit seinem RSA-Public-Key. Den RSA-Private-Key haben wir nicht, er ist auch nicht im Receiver gespeichert, nur Humax hat diesen Key. Er lässt sich auch nicht aus dem im Receiver vorhandenen Public-Key berechnen. Wäre also die Firmware im Receiver fehlerfrei, dann könnten wir keine eigenen Daten zum Receiver schicken, sondern nur von Humax Zertifizierte.

Wir haben aber Glück, den momentan befindet sich noch eine Backdoor in der Firmware die es ermöglicht die Signaturprüfung zu umgehen, womit beliebige Daten akzeptiert werden. Das PR-HD1000-Heaven Tool unterstützt diese Funktion (er tauscht die vier Bytes des Signaturheaders der ein Offset auf den Signaturbody darstellt gegen die Bytefolge FFFF0000 aus).

## 4.1 Zerlegen eines HDFs

Um an den Inhalt von HDF Dateien (z.B. die Inhalte vom Firmwareupdate „prhd1000\_10013\_ssd1.hdf“ oder selbst geloggte OTA-Updates) ranzukommen kann man wie folgt vorgehen:

- „Load HDF file“ anklicken und die HDF Datei angeben
- Die enthaltenen Module werden nun in Tabellenform dargestellt:



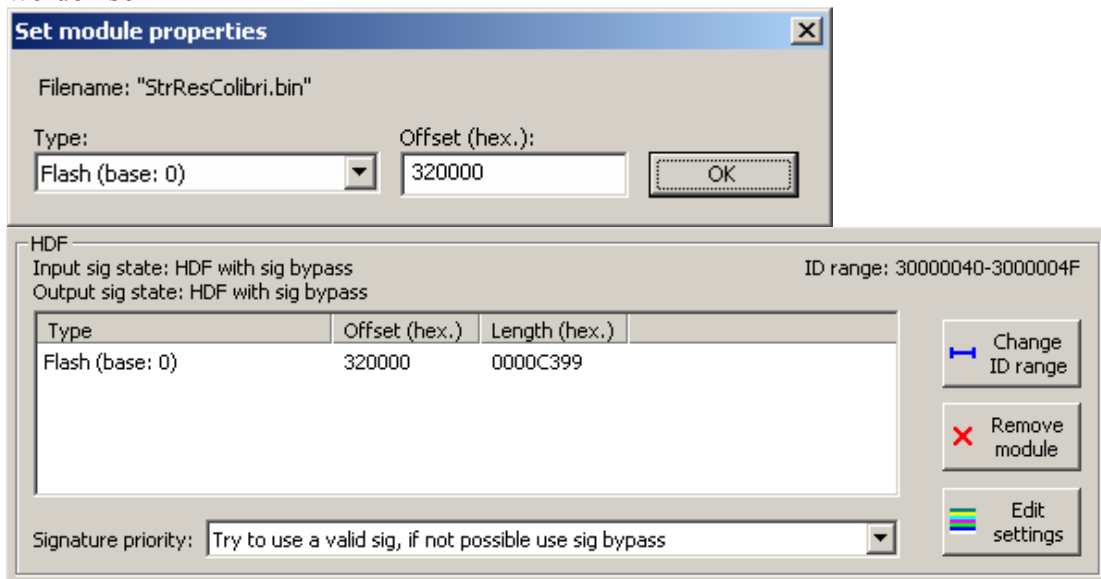
- „Input sig state“ zeigt an ob die HDF Datei signiert ist und falls ja ob die Signatur gültig ist. In diesem Beispiel ist alles OK, da es ein offizielles File („prhd1000\_10013\_ssd1.hdf“) ist.
- „ID range“ zeigt an für welches Modell das File gedacht ist. Mein PR-HD1000 hat z.B. die Product ID 3000004F, er würde das Update also akzeptieren da es innerhalb des Bereiches ist.
- „Save all modules“ anklicken und man wird nach den Namen für die zwei Dateien gefragt (der vorgeschlagene Text ist „Flash\_060000.bin“ und „Flash\_320000.bin“). Die Dateien enthalten also 1:1 genau die Data, die nach dem Einspielen in den Receiver so auch im Flash stehen würden.

## 4.2 Zusammenbauen eines HDFs

Man kann auch wie folgt ein eigenes HDF zusammenbauen, wenn die einzelnen Daten die ins Flash oder EEPROM geschrieben werden sollen als Dateien vorliegen:

- Nach dem Starten des Tools ist die Tabelle leer die „ID range“ ist bereits richtig vorbelegt
- „Add module“ anklicken und den Namen der eigenen Datei angeben

- Den Typ auswählen und den Offset angeben welche stelle die Datei geschrieben werden soll



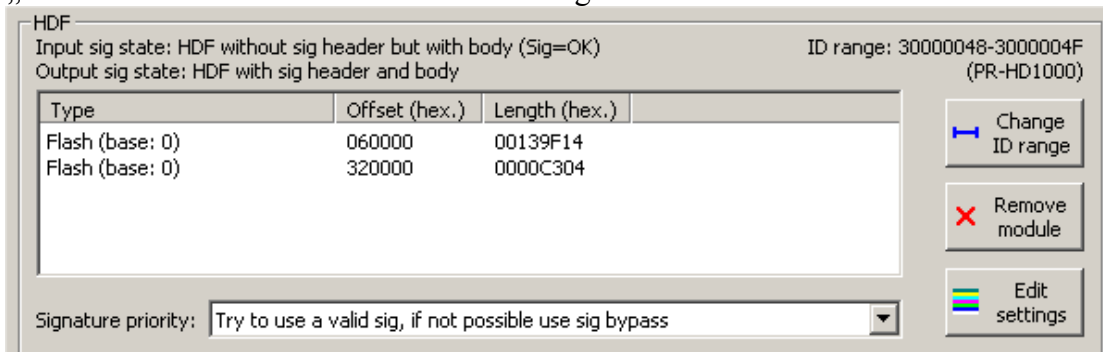
- „Signature priority“ so wie im Screenshot dargestellt einstellen damit unser selbst erstelltes HDF auch von Receiver angenommen wird
- Das erstellte HDF kann man jetzt mit „Save HDF file“ entweder abspeichern oder auch gleich mit „Write to STB“ zum Receiver schicken

### 4.3 Gültig machen eines vorhandenen HDFs

Es gibt auch Fälle bei denen man bereits ein HDF besitzt, dies aber keine gültige Signatur besitzt.

Ein Beispiel ist ein geloggtetes OTA firmware update (siehe auch Kapitel „Over-The-Air (OTA) update“). Ein solches HDF enthält keinen Signatur Header, sondern nur einen Signatur Body. Dieser ist jedoch nutzlos da der Receiver den Signatur Body nicht beachtet wenn er in Header nicht angekündigt wird. Der Receiver würde solch ein HDF also zurückweisen wenn er es über die serielle Schnittstelle empfangen würde. Über das eingebaute Satellitenempfangsteil jedoch akzeptiert er alle Updates, da findet generell keine Signaturprüfung statt. Um die HDF auch für die serielle Schnittstelle akzeptabel zu machen kann man wie folgt vorgehen:

- „Load HDF file“ anklicken und die Datei angeben



- „Input sig state“ zeigt an das die Datei keinen Signatur Header hat (aber einen gültigen Body mit korrekter Signatur)
- „Signatur priority“ wie im Screenshot dargestellt einstellen
- „Output sig state“ zeigt jetzt an das die Datei nun einen Signatur Header und Body hat. Das Benutzen des Signaturbypasses (der Backdoor) ist in diesem Fall also gar

nicht nötig und die Datei würde auch nach dem Beseitigen des Bugs akzeptiert werden.

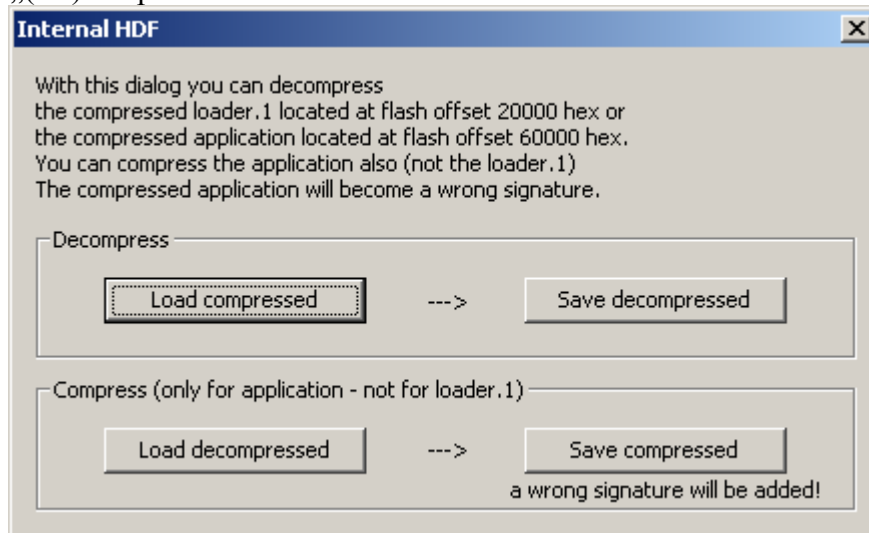
Ein anders Beispiel bei dem die HDF Datei weder Signatur Header noch Body hat und deshalb der Signaturbypass aktiviert werden muss, ist wenn die Datei von einem Programmliedeneditor stammte der keine Signaturfunktion unterstützt.

## 5 (De)komprimierung des Flashes

Dieses Kapitel ist nur für Entwickler interessant die den Loader.1 oder die Applikation disassembeln wollen. Im Gegensatz zum Bootloader und Loader.0 die unkomprimiert im Flash liegen und direkt disassembled werden können, sind der Loader.1 und die Applikation nur komprimiert im Flash vorhanden. Diese werden zur Laufzeit ins wesentlich grössere RAM dekomprimiert und danach ausgeführt.

Die Daten können wie folgt dekomprimiert werden:

- „(De)compress flash“ anklicken



- „Load compressed“ anklicken und die komprimierte Datei angeben. Die Applikation kann man z.B. von einem zerlegtem Firmwareupdate nehmen. Alternativ kann man auch Beide aus einem entsprechend gemachten Backup entnehmen (eins vom Offset 20000 hex das den Loader.1 enthält oder eins von Offset 60000 hex das die Applikation enthält).
- „Save decompressed“ anklicken um die entpackten Daten zu speichern

Seit der Version 1.01 kann man die Applikation auch wieder komprimieren. Dies ist in Verbindung mit dem Kapitel „Ausführen von eigenen Code“ nützlich.

## 6 String Resource

Wie folgender Screenshot zeigt kann man auch die Texte ändern die der Receiver am Bildschirm ausgibt:

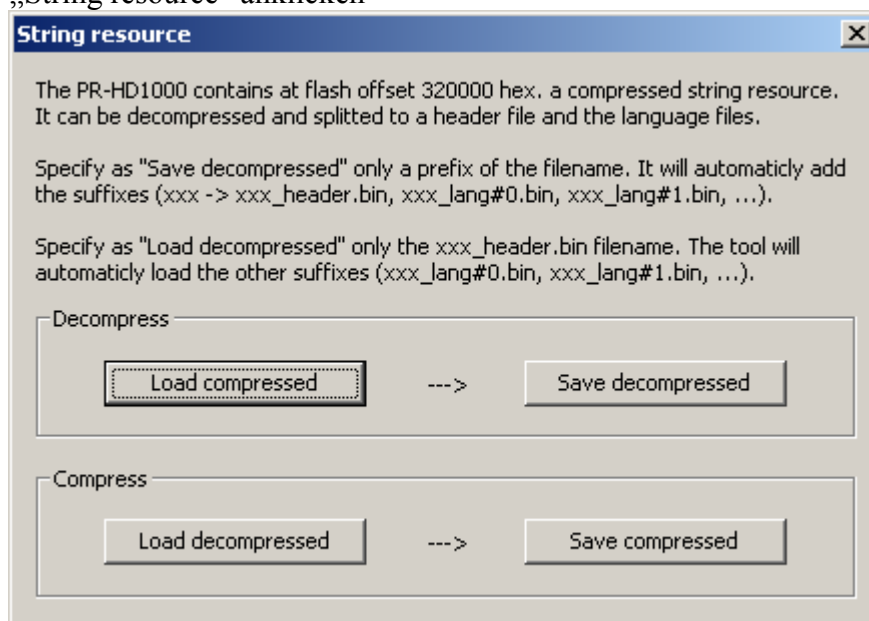




Die ganzen Texte der vier Sprachen die der Receiver unterstützt sind getrennt von der Applikation (die bei Offset 60000 hex liegt) in einem String-Resource-File bei Offset 320000 hex im Flash gespeichert. Da das String-Resource-File komprimiert ist kann man es nicht so einfach patchen.

Zum Entpacken kann man wie folgt vorgehen:

- „String resource“ anklicken



- „Load compressed“ anklicken und die komprimierte Datei angeben. Das String-Resource-File kann man z.B. von einem zerlegtem Firmwareupdate nehmen. Alternativ kann man es auch einem entsprechend gemachten Backup entnehmen (eins vom Offset 320000 hex).
- „Save decompressed“ anklicken und den vorderen Teil des Dateinamens eingeben. Daraus werden automatisch mehrere Dateien erzeugt, eine Header Datei und vier Sprachdateien. Die Sprachdateien enthalten einzelne Texte die durch ein Nullzeichen getrennt sind, deshalb braucht man einen Hexeditor zum patchen. Es gibt kein

Problem wenn sich die Länge der Texte und somit die Gesamtlänge der Datei ändert, man muss nur darauf achten das die Anzahl der Nullzeichen gleichbleibt.

Zum Packen kann man wie folgt vorgehen:

- „Load decompressed“ anklicken und die beim vorherigen Decomprimieren erzeugte Headerdatei angeben. Es werden dann automatisch auch die vier Sprachdateien geladen.
- „Save compressed“ anklicken und den Dateinamen angeben. Diese Datei kann man dann wie im Kapitel „Zusammenbauen eines HDFs“ beschrieben zum Receiver schicken.

## **7 Programmliste des Receivers am PC ändern**

Viele Leute haben sich gewünscht die Programmliste des Receivers auch am PC ändern zu können. Leider gibt es kein offizielles Programm vom Hersteller für den PR-HD1000. Es gibt zwar z.B. „Formula For NEO“ von HUMAX aber das ist für andere Modelle geschrieben worden und ist für den PR-HD1000 nicht geeignet. Mit PR-HD1000-Heaven kann man aber seit der Version 1.02 die Programmliste am PC editieren.

Um die Programmliste zu ändern muss man diese von Receiver lesen. Das dauert protokollbedingt relativ lange (ca. 20 Minuten), da der Receiver Daten zum PC immer unkomprimiert sendet. Dann kann man die Programme am PC umsortieren und als Datei abspeichern. Anschliessend muss man die geänderte Programmliste wieder zum Receiver übertragen. Das geht protokollbedingt sehr schnell (ca. 10 Sekunden), da in diese Richtung der Receiver den Empfang von komprimierten Daten unterstützt. Es empfiehlt sich also die Settings nur einmal zu holen und am PC zu speichern. Bei einem Änderungswunsch dann die Datei am PC öffnen, Änderungen machen, unter neuen Namen am PC zu speichern und dann zum Receiver übertragen, was ja nur ca. 10 Sekunden dauert.

### **7.1 Programmliste von Receiver zum PC übertragen**

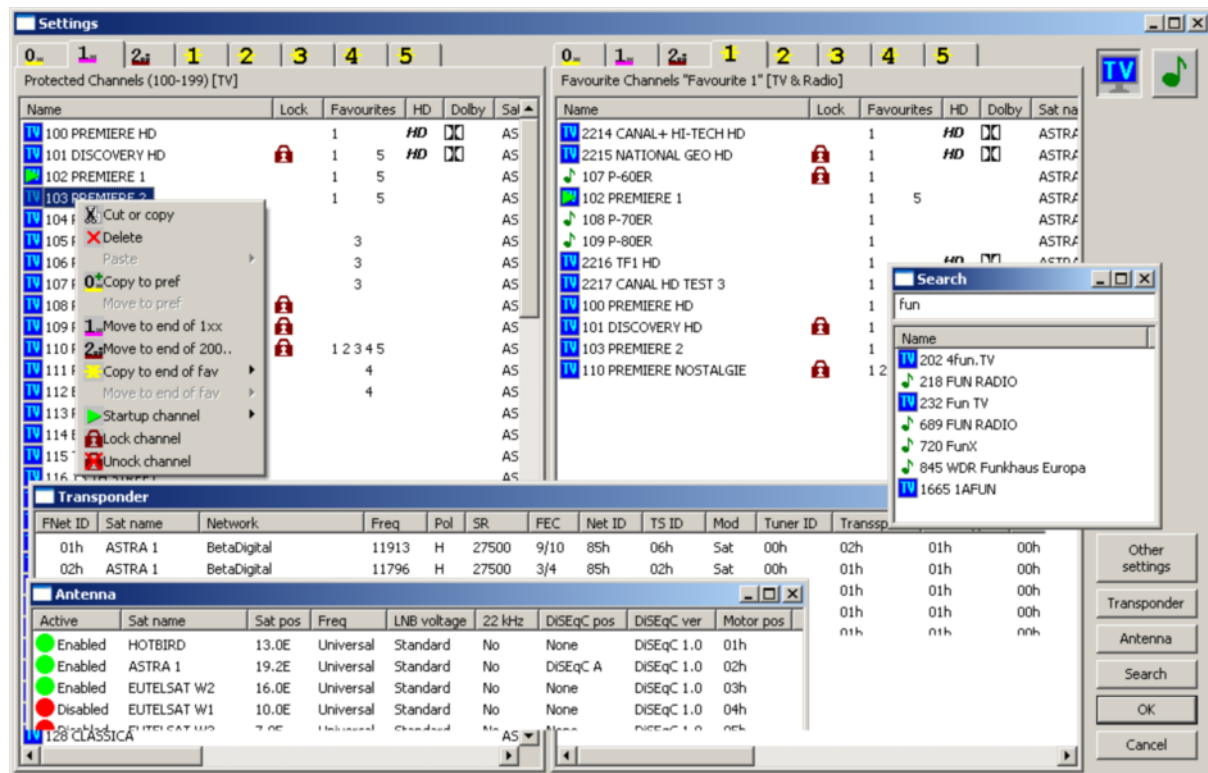
Um die Programmliste vom Receiver zu holen kann man wie folgt vorgehen:

- Den Receiver und den PC mit einem Nullmodemkabel (<http://de.wikipedia.org/wiki/Nullmodem-Kabel>) verbinden
- Im Feld „STB is connected to“ den COM-Port einstellen an dem der Receiver angeschlossen ist
- „Read from STB“ anklicken
- „Read settings“ und „OK“ anklicken
- Den Receiver über den Netzschalter an der Rückseite aus- und wieder einschalten (das Schalten in den Standby und anschliessendes Einschalten über die Fronttaste genügt nicht).
- Die Daten werden jetzt zum PC übertragen was ca. 20 Minuten dauert.

Die Daten enthalten nicht nur die Programmliste sondern auch andere Informationen wie z.B. die System ID, PIN, usw. Beim allerersten Mal noch bevor man die Programmliste durch einen Klick auf „Edit settings“ bearbeitet, sollte man unbedingt die Daten mit „Save HDF file“ sichern und sich diese Datei gut aufheben. Sollte es durch einen Bug im „Edit settings“ Dialog später im Receiver zu problem kommen, kann man das original Settings backup wieder zurückspielen.

## 7.2 Programmliste am PC bearbeiten

Nachdem die Daten wie im vorherigen Abschnitt beschrieben vom Receiver gelesen wurden, oder alternativ über „Load HDF file“ geladen worden sind kann man jetzt durch einen Klick auf „Edit settings“ den Programmlisteneditor aufrufen.



An der rechten Seite kann man mit „Search“ einen Suchdialog anzeigen lassen. In der ersten Zeile kann man einen Teil des Programmnamen eingeben. In der Liste unterhalb werden dann alle Programme aufgeführt die den Text enthalten. Durch einen Doppelklick auf ein Suchergebnis wird dieses im Hauptfenster angezeigt.

Durch einen Klick auf „Antenna“ kann man sich die Antennenkonfiguration anzeigen lassen.

Ein Klick auf „Transponder“ zeigt die Transponderkonfiguration an.

Unter „Other settings“ kann man sich die Geräte-PIN anzeigen lassen und auch ändern. Auch die System/Produkt ID lässt sich hier ändern.

Das Hauptfenster enthält zwei Programmlisten. Jede List enthält Reiter mit denen man zwischen den bevorzugten Programmen (0..99), den geschützten Programmen (100..199), den normalen Programmen (ab 200) und den fünf Favoriten Gruppen mit je 100 Programmen wechseln kann. Die Programme 100 bis 199 sind zwar am Receiver vor dem Verändern geschützt, nicht jedoch am PC.

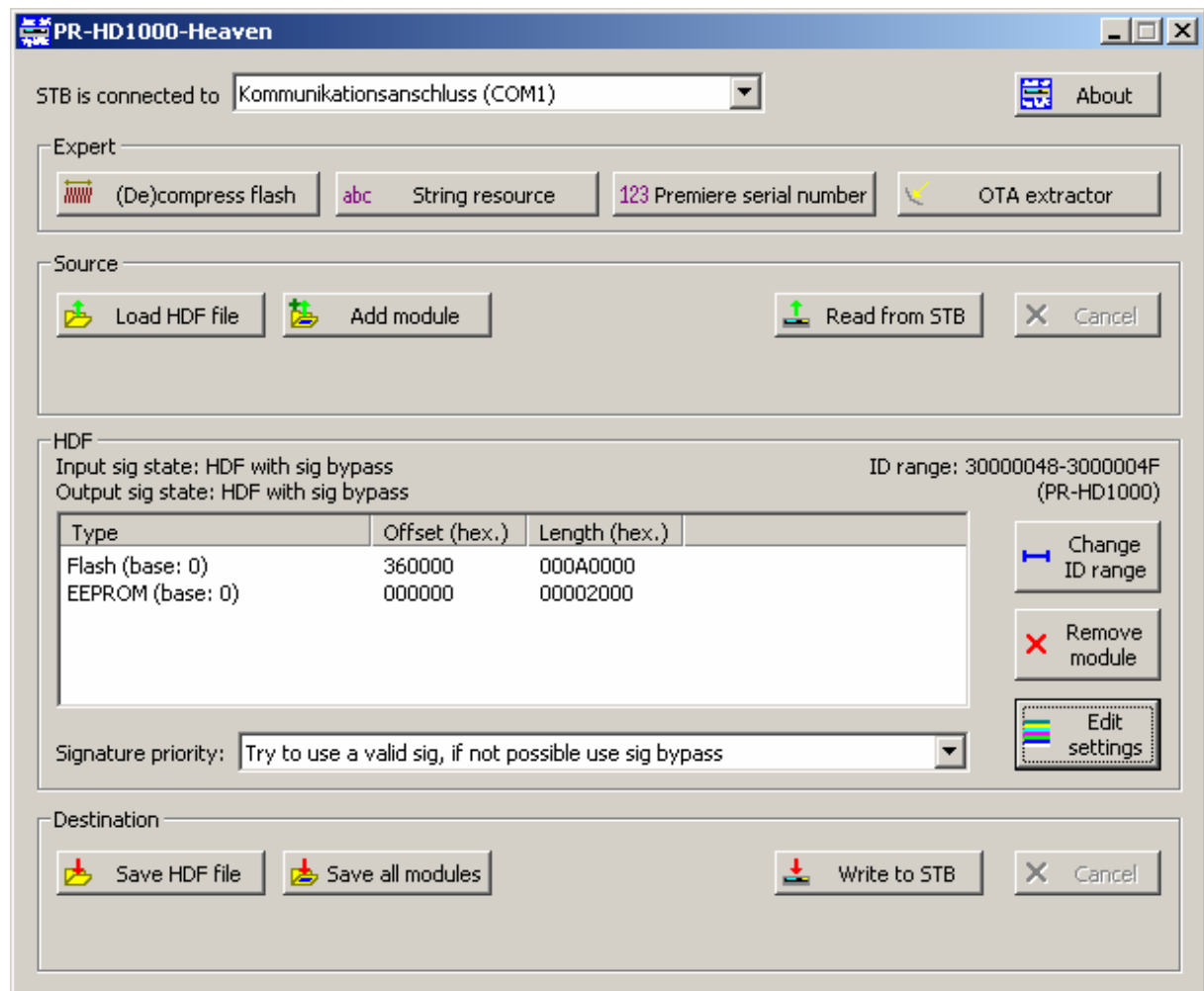
Zum Verschieben oder Kopieren von Programmen markiert man das erste Programm das man ändern will mit der Maus, hält die <Strg>-Taste gedrückt und klickt weitere Programme mit der Maus an. Danach lässt man die <Strg>-Taste los und klickt man mit der rechten Maustaste

auf eines dieser markierten Programme. Jetzt wird ein PopUp-Menü dargestellt bei der man die gewünschte Aktion auswählt.

Wenn man alle gewünschten Aktion durchgeführt hat übernimmt man die Settings durch den klick auf „OK“. Jetzt kann man die Settings mit „Save HDF file“ auf dem PC sichern und wie folgt zum Receiver übertragen

### 7.3 Programmliste vom PC zum Receiver übertragen

Die Programmliste kann jetzt durch einen Klick auf „Write to STB“ zum Receiver übertragen werden. Auch hier startet der Transfer nach dem kurzen Aus- und Einschalten mit dem Netzschalter auf der Receiverrückseite. Der Transfer dauert ca. 10 Sekunden. Die „Signature priority“ sollte wie im Bild dargestellt auf dem Defaultwert „Try to use a valid sig, if not possible use sig bypass“ eingestellt sein. Mit „Change ID range“ kann man falls man seine ID geändert hat den Bereich anpassen.



## 8 Memory Map

Hier die physikalische Speicheraufteilung von Flash. Die virtuelle Basisadresse die der Prozessor beim Flashzugriff anspricht ist F0000000.

Adresse (hex)	Länge (hex)	
0 -5FFF	6000	Bootstrap
6000 -1F3FF	19400	Loader.0 Nur ca. 4100 hex Bytes benutzt Wird nach RAM 8000 kopiert und ausgeführt
1F400 -1FBFF	800	RSA-Modulus für Transfer, Loader.1 und Applikation *
1FC00 -1FFFF	400	RSA-Modulus für die Signatur von den drei mit einem Stern gekennzeichneten Blöcken
20000 -5F7FF	3F800	Loader.1 Wird entpackt nach RAM 1000000 hex kopiert und ausgeführt
5F800 -5FFFF	800	RSA-Modulus für Transfer, Loader.1 und Applikation *
60000 -31FFFF	2C0000	Applikation Wird entpackt nach RAM 400 hex kopiert und ausgeführt
320000 -35BBFF	3BC00	String-Resource
35BC00 -35F800	3C00	?
35F800 -35FFFF	800	RSA-Modulus für Transfer, Loader.1 und Applikation *
360000 -3FFFFFF	A0000	u.a. Programmliste

\* Die drei 800 hex Bytes langen Blöcke sind identisch

Der Flashbereich 0 – 1FFFF ist anscheinend schreibgeschützt.

Der vom Loader.1 und der Applikation benutzte Bereich und die RSA-Daten sind kryptografisch gesichert. Eine Manipulation (die ja durch den Sig-Bypass-Bug möglich ist) führt zu einem unbrauchbaren Receiver. Harmlos ist anscheinend eine Manipulation der String-Resource und der Programmliste, falls man vorher ein Backup gemacht hat.

Das DWORD im Little Endian Format an der EEPROM Adresse 0 ist die Produkt ID.

Der Bootvorgang läuft im Groben wie folgt ab:

- Der Bootstrap prüft die Unversehrtheit von Teilen von sich selbst, der RSA-Daten, Loader.0, Loader.1 und der Applikation. Falls der Check fehlschlägt bricht er ab, ansonsten kopiert der den Loader.0 ins RAM und führt ihn aus.
- Der Loader.0 dekomprimiert den Loader.1 ins RAM und führt ihn aus.
- Der Loader.1 sendet Bytes über die serielle Schnittstelle die anzeigen das er bereit für einen Datenaustausch ist, falls der PC nicht sofort antwortet dann dekomprimiert er die Applikation ins RAM und führt sie aus.

## 9 Troubleshooting

Während der Übertragung von Daten zum Receiver können Fehlermeldungen [E-xx] am Display/Bildschirm erscheinen.

Fehlermeldung	Grund
[E-01] ID-Error	<p>Die Produkt ID des Receivers ist nicht innerhalb des Bereiches das im empfangenen Dateiheader steht.</p> <p>Man kann entweder die Daten nochmal zum Receiver schicken, aber vorher „Change ID range“ und „Any“ anklicken oder man kann seine ID wie folgt wieder auf Standart stellen:</p> <p>Man kann wie in Kapitel 7 (Programmliste des Receivers am PC ändern) vorgehen, da hier auch die Produkt ID wieder berichtigt werden kann, die unter „Other settings“ zu finden ist. Bei meinen PR-HD1000 ist diese z.B. „30000048“.</p> <p>Bevor diese neue Produkt ID mit „Write to STB“ zum Receiver geschickt wird muss man mit durch Klick auf „Change ID range“ und „Any“ den Range auf 0-FFFFFFFF ändern. Somit akzeptiert der Receiver mit seiner momentan noch falschen Produkt ID die neue richtige Produkt ID.</p>
[E-02] HC-Error	
[E-03] HF-Error	
[E-04] DC-Error	
[E-05] AC-Error	
[E-06] UT-Error	Übertragungsfehler. Dann einfach die Übertragung wiederholen. Erst wenn die Übertragung komplett ist, werden die Daten ins Flash/EEPROM geschrieben. Dieser Fehler ist also harmlos.
[E-07] AD-Error	
[E-08] FLASH-Error	Es wurde versucht einen schreibgeschützten Bereich (z.B. Bootstrap) zu ändern.
[E-09] FLASH-Error	Siehe [E-08]
[E-10] IBC-Error	
[E-11] AD-Error	
[E-12] FLASH-Error	Siehe [E-08]
[E-14] OTA-Error	
[E-15] IS-Error	Die empfangene Datei hat keine gültige Signatur. Dann einfach mit dem Tool den Signatur-Bypass aktivieren.
[E-16] DEC-Error	
[E-17] EEPROM-Error	
[E-18] IS-Error	Siehe [E-15]
[E-19] UNKNOWN-Error	

## 10 Testmode

Man kann bei dem Receiver auch temporär einen Testmode aktivieren. Nach dem Aus- und Einschalten mit dem Schalter auf der Receiverrückseite ist dann der normale Betriebsmodus wieder aktiv.

Der Testmode wird aktiviert in dem man ein HDF zum Receiver schickt bei dem der ID range nicht wie gewöhnlich auf 30000048-3000004F eingestellt ist, sondern auf den Bereich FFFFFFFF-FFFFFFF. Dieser ignoriert dann die eigentlichen Nutzdaten im HDF, setzt aber wegen des speziellen Ranges im EEPROM des Receivers das Testmodeflag, startet anschliessend die Applikation, die das Testmodeflag ausliest und dann den Testmode ausführt. Das Testmodeflag im EEPROM wird während des bootens immer zurückgesetzt.

Obwohl der Receiver anscheinend den HDF-Inhalt ignoriert, empfehle ich trotzdem keinen Müll reinzuschreiben, sondern stattdessen mit dem Hexeditor eine Datei erstellen die nur ein Byte mit dem Wert 0 enthält. Diese Datei mit „Add module“ einlesen, als Type „Flash (base: 0)“ auswählen, als Offset (hex.) „60000“ verwenden, den Range FFFFFFFF-FFFFFFF setzen und die Datei mit „Save HDF file“ unter dem Namen „ActivateTestmode.hdf“ abspeichern.

Der Offset 60000 (erstes Headerbyte der Applikation = MSByte vom Signaturoffset) hat immer den Wert 0 und das HDF ändert damit nie etwas am Flash. So ist man auf der sicheren Seite falls die Daten doch mal ins Flash geschrieben werden.



Mit dem Testmode können diverse Receiverfunktionen getestet werden:

### Tasten an der Frontseite:

Nacheinander die 7 Tasten drücken. Zur jeder Taste erscheint ein Text im Display. Nach der siebten Taste erscheint KEY\_PASS (=Tastentest war erfolgreich).

### Tasten auf der Fernbedienung:

Taste 1: Schaltet alle Pixel am Display an

Taste 2: Wiederholtes drücken schaltet zwischen diversen Tests um

Taste 3: Colorbar

Taste 4: CAS CRYPTO

Taste 5: Arirang

Taste 6: CAS IRDETO

Taste 7: Aktiviert nacheinander die Symbole im Display

Taste 8: Schaltet Farbbalken auf dem TV-Gerät ein und aus

Taste 9: HD STREAM

Taste 0: Wiederholtes drücken schaltet von Step1 – Step4

Bei den restlichen Tasten bleibt der Test hängen.

Auch beim Einschieben und Herausziehen einer Smartcard erscheinen entsprechende Meldungen.

## 11 Ausführen von eigenen Code

Nach langem Suchen habe ich endlich eine Möglichkeit gefunden die Signaturprüfung der Applikation temporär zu umgehen und somit eigenen Code auszuführen (z.B. eine gepatchte Applikation). Denkbar wäre den eigenen Code so zu schreiben das der Schreibschutz des Bootstraps aufgehoben wird und danach alle Signaturprüfungen aus dem Bootstrap zu löschen. Danach wäre die Signaturprüfung nicht nur temporär sondern dauerhaft ausgeschaltet.

Der Bootstrap prüft die Unversehrtheit folgender Daten (Blocklänge/Offset zur Sig steht im ersten DWORD des Blocks):

- (return 0 bei Fehler) Block bei 1FC00 (enthält RSA-Modulus)
- (return 0 bei Fehler) Block bei 500 (umfasst Teile des Bootstraps und den Loader.0)
- (return 0 bei Fehler) Block bei 300 (umfasst Teile des Bootstraps)
- (return 1 bei Fehler) Block bei 5F800 (enthält RSA-Modulus für Transfer, Loader.1 und Applikation)
- (return 102 bei Fehler) Block bei 2000 (umfasst den Loader.1)
- (return 202 bei Fehler) Block bei 6000 (umfasst die Applikation)

Return 3 falls alle Signaturen in Ordnung waren

Der Returnwert wird nach 0xE0400050 gespeichert.

Der Loader.0 list den Inhalt von 0xE0400050 (erg sigcheck bootstrap) aus

- Falls der Inhalt 0 oder 1 ist wird versucht die RSA-Backups (35F800=LastKnownGood, 1F400=Default) nach 5F800 zurückzupielen (Endlosschleife falls das Default-Backup bereits die gleichen Dateninhalt von 5F800 hat).
- Falls der Inhalt 3=AlleSigsOk ist wird der Wert E in 0xE0400024 gespeichert.
- Falls der Inhalt 202=NurAppUngültig ist wird der Wert F in 0xE0400024 gespeichert.
- Bei den restlichen Inhalten (z.B. 102=Loder.1 ungültig) springt der sofort in eine Endlosschleife (siehe Anweisung bei F00040C4).

Der Loader.1 prüft ob ein Datentransfer von der seriellen Schnittstelle ansteht. Falls dies der Fall ist empfängt er ihn und booted dann.

Falls 0xE0400024 den Wert F=AppSigUngültig hat, versucht der Receiver ein OTA-Update der Applikation durchzuführen und bleibt mit einen Fehler stehen falls der das nicht schafft (weil z.B. die Satantenne abgesteckt ist).

Die Applikation wird nur gestartet wenn 0xE0400024 einen anderen Wert als F hat (also z.B. E=AlleSigsOk).

Im Gegensatz zum Loader ist es also harmlos wenn man sich die Applikation überschreibt, denn selbst ohne OTA, kann man über die serielle Schnittstelle eine originale Applikation zurückspielen und der Receiver funktioniert wieder einwandfrei.

Zwar kann man, nachdem die Signaturprüfung der Applikation im Bootstrap stattgefunden hat diese nachträglich über die serielle Schnittstelle ersetzen, jedoch wird nach dem Ende des



Transfers diese Applikation nicht gestartet, sondern gebootet womit wieder eine Signaturprüfung stattfindet.

**Nun zum Trick, der es ermöglicht eigenen Code ausführen zu lassen:**

- Antenne temporär abstecken (damit ein OTA-Update verhindert wird)
- Die original Applikation entpacken
- Die gewünschten Patches durchführen (auf jeden Fall wie weiter unten beschrieben zusätzlich den Testmode Code rauspatchen)
- Die gepatchte Applikation (kann auch kompletter eigener Code sein) wieder packen (das wird erst seit der Version 1.01 von PR-HD1000-Heaven unterstützt)
- Die gepackte Applikation mit „Add module“ einlesen, als Type „Flash (base: 0)“ auswählen, als Offset (hex.) „60000“ verwenden
- den Range 30000048-3000004F setzen
- mit „Write to STB“ die gepatchete Applikation zum Receiver übertragen. Diese wird dann ins Flash geschrieben und anschliessend gebootet. Dann stellt der Receiver fest das die Signatur falsch ist und versucht deshalb ein Update über die serielle Schnittstelle und über OTA zu bekommen. Da die Antenne im ersten Schritt abgesteckt wurde schafft er das nicht und bleibt mit Fehler [E-14] OTA-Error stehen.
- Die Datei „ActivateTestmode.hdf“ wie im vorherigen Kapitel „Testmode“ beschrieben erstellen und zum Receiver schicken. Dieser setzt wegen dem Range FFFFFFFF-FFFFFFF das Testmodeflag im EEPROM, das ja während des Bootens zuvor gelöscht worden war und startet dann direkt die Applikation ohne nochmal zu booten (sonst würde das Testmodeflag ja wieder gelöscht werden). **Der Bug dabei ist, das die Testmodeabfrage und das Aufrufen der Applikation durchgeführt wird, bevor der Inhalt der Adresse 0xE0400024 ausgewertet wird. Dieser enthält, wie vorher beschrieben, die Info ob die Applikation unverfälscht ist.**
- Jetzt kann man die Antenne wieder anstecken und seine gepatchte Applikation genießen

Warum habe ich aber am Anfang geschrieben das der eigene Code nur temporär ausgeführt werden kann? Nun der Nachteil dieser Methode ist das nach dem Aus- und Einschalten des Receivers der Bootstrap die Applikationssignatur wieder prüft, feststellt das diese falsch ist und sie deswegen die Applikation nicht startet.

Man kann jetzt ...

- ... die Datei „ActivateTestmode.hdf“ (mit abgesteckter Antenne) zum Receiver schicken (dies geht blitzschnell) und die noch im Flash vorhandene gepatchte Applikation wird wieder gestartet.
- ... mit angesteckter Antenne booten, dann bekommt man die originale Applikation via OTA wieder draufgespielt (**oder ein Sicherheitsupdate das die Backdoor und den Bug für immer schliesst**).
- ... über die serielle Schnittstelle die originale Applikation draufspielen.

Zusätzlich zum eigentlichen Patch, muss man die Testmodeabfrage aus der Applikation löschen, sonst würde die gepatchte Applikation (bedingt durch den Trick) nie in den normalen Modus gehen.

Der folgende Ausschnitt stammt von der dekomprimierten Applikation Version 1.00.13 (andere Versionen haben andere Offsets).

```
RAM:000FB1B6 00 28                                CMP     R0, #0
RAM:000FB1B8 02 D0                                BEQ     NORMAL_MODE
RAM:000FB1BA 58 F7 83 FC                          BL      TEST_MODE
RAM:000FB1BE
RAM:000FB1BE                                LOOP_FOREVER
RAM:000FB1BE FE E7                                B       LOOP_FOREVER
RAM:000FB1C0                                ; -----
RAM:000FB1C0
RAM:000FB1C0                                NORMAL_MODE
RAM:000FB1C0 79 F7 2C FC                          BL      sub_74A1C
```

Um immer den Normalmode zu starten, kann man z.B. die gelb markierten Bytes durch 00 ersetzen. Die Bytefolge „00 00“ steht für den Befehl „LSL R0, R0, #0“ (es wird also nichts verändert - also wie ein „NOP“).

In der entpackten Applikationsdatei der Version 1.00.13 befindet sich der zu patchende Bereich ab FADB8 (RAM-Offset – 400h = FileOffset)

Als erste kleine Patchübung kann man z.B. in der Version 1.00.13 der Applikation am Fileoffset 124BCC den Text „STANDBY“ durch „COLIBRI“ ersetzen, dann erscheint beim Schalten in den Standby folgender Displaytext am Receiver:



Viel Spass,  
Colibri