HOCHSCHULE DER MEDIEN

Institut für Rundfunktechnik

Bachelorarbeit

Test und Analyse von Möglichkeiten zur Online-Distribution von Mehrkanalton für HbbTV und PC über das offene Internet

zur Erlangung des akademischen Grades

Bachelor of Engineering

vorgelegt von:

Sebastian Siepe

am 28.02.2013

Audiovisuelle Medien

Hochschule der Medien, Stuttgart

Erstprüfer: Prof. Oliver Curdt, HdM

Zweitprüfer: Martin Schmalohr, IRT

Abstract

Die Verbreitung von Audio- und Videoinhalten über das Internet weitet sich ständig aus. Während Videos in HD-Auflösung bereits über das Internet angeschaut werden können, werden die meisten Audioinhalte auf Portalen wie Youtube oder den Mediatheken der Rundfunkanstallten nur im Stereo-Format angeboten. Um die Verbreitung von Mehrkanalton-Inhalten im Internet weiter voranzutreiben, wird in dieser Abschlussarbeit nach Möglichkeiten gesucht, eine Mehrkanalton-Übertragung über das Internet für HbbTV-Geräte und PCs zu realisieren.

Dazu sollen durch ausführliche Tests verschiedener Mehrkanalton-fähiger Audiocodecs Codierprofile erarbeitet werden, welche eine möglichst hohe Interoperabilität mit vielen Endgeräten gewährleisten.

The distribution of audio- and video-content across the internet is constantly increasing. Today videos can be watched in HD quality over the internet on portals such as Youtube or the media libraries of the TV broadcasting companies. Unfortunately audio content ist most often delivered only in stereo. In order to find a way to enable a transmission of multichannel audio content over the internet, this thesis searches for possibilities to realise a distribution of multichannel audio content for HbbTV-devices and computers.

For this purpose, a variety of tests with different multichannel audio codecs were run in order to identify codec profiles, that ensure a high interoperability with many HbbTV-devices and computers.

Danksagung

Diese Abschlussarbeit wurde von Herrn Martin Schmalohr am Institut für Rundfunktechnik betreut. Für seinen unermüdlichen Einsatz und seine stets konstruktive Hilfe möchte ich ihm ausdrücklich danken.

Des weiteren gilt mein Dank Herrn Prof. Oliver Curdt, der ebenfalls jederzeit für Fragen zur Verfügung stand.

Zuletzt möchte ich Herrn Arnd Paulsen von Dolby Laboratories danken, der für diese Arbeit Testmaterial und Software der Firma Dolby zur Verfügung stellte.

Eidesstattliche Erklärung

„Hiermit versichere ich, Sebastian Siepe, an Eides Statt, dass ich die vorliegende Bachelorarbeit mit dem Titel: „Test und Analyse von Möglichkeiten zur Online-Distribution von Mehrkanalton für HbbTV und PC über das offene Internet“ selbstständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe.

Die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen wurden, sind in jedem Fall unter Angabe der Quelle kenntlich gemacht.

Die Arbeit ist noch nicht veröffentlicht oder in anderer Form als Prüfungsleistung vorgelegt worden.

Ich habe die Bedeutung der eidesstattlichen Versicherung und die prüfungsrechtlichen Folgen (§26 Abs. 2 Bachelor-SPO (6 Semester), § 23 Abs. 2 Bachelor-SPO (7 Semester) bzw. § 19 Abs. 2 Master-SPO der HdM) sowie die strafrechtlichen Folgen (gem. § 156 StGB) einer unrichtigen oder unvollständigen eidesstattlichen Versicherung zur Kenntnis genommen.“[1]

Ort, Datum

Name

[1] Festgestellte Plagiate führen gemäß § 26 Abs. 2 der SPO der 6-semestrigen bzw. §23 Abs. 2 der 7-semesterigen grundständigen Studiengänge bzw. § 19 Abs.1 Satz 3 der SPO der Masterstudiengänge zum Verlust des Prüfungsanspruches und damit zur Exmatrikulation.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

AAC	Advanced Audio Coding
AC-3	Adaptive Transform Coder 3
AIT	Application Information Table
ATSC	Advanced Television Systems Committee
AVC	Advanced Video Coding
AVI	Audio Video Interleave
CBR	Konstante Bitrate
CD	Compact Disc
CEA	Consumer Electronic Association
DMG	Dolby Media Generator
DSL	Digital Subscriber Line
DVB	Digital Video Broadcasting
E-AC-3	Enhanced-Adaptive Transform Coder 3
EBU	European Broadcasting Union
ETSI	European Telecommunications Standards Institute
FFT	Fast Fourier Transformation
FLV	Flash Video
GoP	Group of Pictures
HbbTV	Hybrid Broadcast Broadband TV
HD	High Definition
HDMI	High Definition Media Interface
HE-AAC	High Efficiency Advanced Audio Coding
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
IIS	(Fraunhofer) Institut für integrierte Schaltungen
IRT	Institut für Rundfunktechnik GmbH
LC-AAC	Low Complexity Advanced Audio Coding
LFE	Low Frequency Effects
Lo/Ro	Left Only / Right Only
LS	Left Surround

M3U	MP3-URL
MDCT	Modifizierte diskrete Kosinustransformation
MNR	Mask to Noise Ratio
MP3	MPEG1/2 Layer III
MPEG	Moving Picture Experts Group
OIPF	Open IPTV Forum
OSMF	Open Source Media Framework
PCM	Pulse Code Modulation
PNG	Portable Network Graphics
RS	Right Surround
RTP	Real-Time Transport Protocol
SBR	Spectral Band Replication
SMR	Signal to Mask Ratio
S/PDIF	Sony/Philips Digital Interface
SWF	Shockwave FLASH
TPC	Technology and Production Center Switzerland AG
URL	Uniform Resource Locator
VBR	Variable Bitrate
W3C	World Wide Web Consortium
XML	Extensible Markup Language

1 Einleitung

Die Vernetzung von Fernsehen und Internet wird immer enger. Neben dem klassischen, linearen Fernsehangebot werden von den Rundfunkanstalten zunehmend auch TV-Inhalte im Internet bereitgestellt. So können in den Mediatheken der Fernsehsender TV-Sendungen auch zeitversetzt angeschaut werden.

Dieser Trend, des zeitversetzten Fernsehens und des Medienkonsums über das Internet, hat in den letzten Jahren einen massiven Zuwachs bekommen. Laut „ARD/ZDF-Onlinestudie 2012“ konsumieren mittlerweile 30% aller Internetnutzer gelegentlich TV-Inhalte über das Internet[2]. Um diesen Trend zu bedienen, sind immer mehr Fernseher mit einem Zugang zum Heimnetz für die Internetnutzung ausgestattet. Laut dem Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e.V. (BITKOM) waren 46% aller im Jahr 2012 verkauften Fernseher fähig, Informationen aus dem Internet zu beziehen. Somit verfügten Ende 2012 ca. 22% aller bundesdeutschen Haushalte über einen Fernseher mit Internetanschluss[3].

Die Verbreitung und Nutzung von internetfähigen Fernsehern bringt jedoch nur dann einen Mehrwert für den Nutzer, wenn auch geräte-spezifische Inhalte angeboten werden. Zum Vergleich kann hier die Internetnutzung auf Smartphones heran gezogen werden. Denn der Durchbruch gelang vor allem mit eigens für die Endgeräte entwickelten Applikationen und Funktionen, die spezifische Informationen aus dem Internet zur Verfügung stellten.

Wünschenswert für internetfähige Fernseher wäre, Medieninhalte möglichst in einer gleichen Qualität anzubieten, wie sie über den linearen Fernsehweg empfangen werden können. Im Videobereich werden teilweise HD[4]-Videos auf verschiedenen Portalen im Internet angeboten. Im Audiobereich jedoch werden bei fast allen Anbietern, wie auch den Mediatheken der öffentlich rechtlichen Rundfunkanstalten, Beiträge nur im Stereo-Format angeboten. Da vor allem TV-Inhalte oft mit Mehrkanalton produziert und ausgestrahlt werden, wäre es wünschenswert, dem Zuschauer auch über das Internet TV-Sendungen mit Mehrkanalton anbieten zu können.

Deswegen wird im ersten Teil dieser Abschlussarbeit nach Möglichkeiten gesucht, eine Online-Distribution von Mehrkanalton für Internetfähige Fernseher zu ermöglichen.

Im zweiten Teil der Arbeit wird nach Verfahren gesucht, ein Online-Angebot von Mehrkanalton für PCs zu gewährleisten. Hierbei werden Mediennutzer berücksichtigt, die mit ihrem Computer auf die Mediatheken der Fernsehanstalten zugreifen.

[2] (ARD/ZDF, 2012)
[3] (BITKOM, 2012)
[4] High Definition

1.1 Aufgabenstellung

Die vorliegende Bachelorarbeit „Test und Analyse von Möglichkeiten zur Online-Distribution von Mehrkanalton für HbbTV und PC über das offene Internet“ besteht aus zwei Teilen.

1. **Test und Analyse von Möglichkeiten zur Online-Distribution von Mehrkanalton für HbbTV**

   Im ersten Teil soll durch ausgiebige Tests das Abspielverhalten verschiedener Mehrkanaltonfähiger Audiocodecs auf ausgewählten Hybrid Broadcast Broadband TV-Geräten untersucht werden.

   Ein Codec beschreibt ein Verfahren zur Codierung und Decodierung von Daten. Das Wort wird aus den englischen Bezeichnungen Coder und Decoder gebildet. Bei den eingesetzten Codecs handelt es sich um die in der HbbTV 1.5 Spezifikation der „European Telecommunications Standards Institute“ (ETSI) festgelegten Audioformate „HE-AAC“, „LC-AAC“, „AC-3“ und „E-AC-3“[5], welche bereits ab HbbTV 1.0 unterstützt werden..

   Die Tests werden auf einer vollständigen HbbTV-Teststrecke am „Institut für Rundfunktechnik GmbH“ (IRT) mit einem angepassten Player, der eine vereinfachte Mediathek darstellt, durchgeführt. Jeder Audiocodec liegt dabei in verschiedenen Bit- und Abtastraten, im Mehrkanal- und Stereo-Format, im MPEG2-Transportstrom- und MP4-Container, sowie mit HD und SD Videoinhalten und auch alleine als Audio-Elementarstrom vor.

   Untersucht wird das Abspielverhalten dieser Files mit Hinblick auf die Interoperabilität, A/V-Synchronität, Kanalzuordnung, Standartkonformität, Klangqualität, Lautstärke, sowie einem eventuellen Downmix- und Transcodierverhalten.

   Durch die Analyse des Abspielverhaltens soll ein Audiocodierprofil erarbeitet werden, das von allen, oder möglichst vielen HbbTV-Geräten in der Praxis unterstützt und wiedergegeben wird.

2. **Test und Analyse von Möglichkeiten zur Online-Distribution von Mehrkanalton für PC**

   Im zweiten Teil der Abschlussarbeit soll ebenfalls durch ausgiebige Tests das Abspielverhalten von mehrkanaltonfähigen Audiocodecs in ausgewählten Web-Browsern untersucht werden.

   Die eingesetzten Codecs ergeben sich aus den von den Web-Browsern unterstützen Formaten. Hierbei handelt es sich um die Codecs „HE-AAC“, „LC-AAC“, „OggVorbis“ und „Opus“, sowie die offiziell nicht unterstützen Formate „AC-3“ und „E-AC-3“.

   Die codierten Files sollen durch verschiedene Einbindungsvarianten auf einer HTML5-Testseite hinterlegt werden, um von mehreren Web-Browsern abgespielt werden zu können. Auch hier wird das Abspielverhalten auf die Interoperabilität, A/V-Synchronität, Kanalzuordnung, Standartkonformität, Lautstärke, sowie einem eventuellen Downmixverhalten untersucht.

   Durch die Analyse des Abspielverhaltens soll auch hier ein Audiocodierprofil bestimmt werden, das von allen, oder möglichst vielen Web-Browsern unterstützt wird. Auf Grund der gewonnenen Erkenntnisse, soll eine vereinfachte Referenz-Implementierung einer HTML5-

[5](ETSI TS 102 796, 2012), Abschnitt 7.3.1

Seite erstellt werden, mit der es allen Mediennutzern ermöglicht wird, Mehrkanaltoninhalte über eine Mediathek abzurufen.

1.2 Institut für Rundfunktechnik

Das Institut für Rundfunktechnik GmbH (IRT) ist das zentrale Forschungs- und Entwicklungsinstitut von ARD, ZDF, DRadio, ORF und der Schweizer Radio- und Fernsehgesellschaft, mit Sitz in München.

Im Laufe des über 50 jährigen Bestehens, hat das IRT viele Erfindungen und Lösungen erarbeitet, die Einzug in unsere heutige Medienwelt gefunden haben. Darunter fällt die erste 5.1 Mehrkanaltonübertragung über DVB im Jahre 1998, oder das neu entwickelte Kardioid-Ebenen-Mikrofon im Bundestag.

Ein großes Aufgabenfeld des IRT stellt zurzeit die Forschung und Entwicklung von HbbTV dar, bei dem das Institut eine wichtige Rolle einnimmt. So ist das IRT Mitglied der Standardisierungsgruppe der ETSI und hatte Anteil an allen bisher erschienenen HbbTV-Spezifikationen. Des weiteren versucht das IRT die Funktionalität von HbbTV stetig weiter zu entwickeln und bietet jedes Quartal einen sogenannten „HbbTV Interoperabilitätsworkshop“ an, bei dem Entwickler ihre Geräte und Applikationen testen können.

Im Zuge der Entwicklung an HbbTV ist auch der Wunsch nach einer Mehrkanaltonübertragung für HbbTV-Geräte entstanden. Dies war der Grund für die Bearbeitung dieser Abschlussarbeit.

2 Grundlagen

2.1 HbbTV

„Hybrid Broadcast Broadband TV“ (HbbTV) ist eine Spezifikation, die beschreibt, wie TV-Inhalte und sendungsbegleitende Informationen über Breitband auf TV-Geräte mit Internetanschluss bereitgestellt werden können. Die Motivation ist, einen offenen Standard zu schaffen, der die Verknüpfung zwischen dem linearen Fernsehinhalt und programmbegleitenden Informationen aus dem Internet ermöglicht.

HbbTV wurde von einem Konsortium aus Unternehmen der europäischen Fernsehindustrie, wie Phillips, Loewe, RTL, Samsung, der European Broadcasting Union (EBU) und des IRT, erarbeitet. Die eingesetzten Techniken sind marktübliche Standards wie Digital Video Broadcasting (DVB) und Internet Technologien von den Konsortien Open IPTV Forum (OIPF), Consumer Electronic Association (CEA) und des World Wide Web Consortium (W3C). Die Spezifikation 1.1.1 wurde 2010 durch die ETSI standardisiert. Im November 2012 wurde der Standard durch die Spezifikation 1.5 erweitert[6].

Mit HbbTV ist es möglich, sendungsbegleitende Inhalte aus dem Internet auf den Fernseher zu bringen. Dies können digitale Teletexte sein, Abstimmungen, Wetterinformationen oder für HbbTV-Geräte angepasste Mediatheken, in denen zeitversetzt TV-Inhalte über das Internet angeboten werden. Auch Untertitel, eine alternative Tonspur oder Gebärdendolmetscher können übertragen werden. Durch das Drücken des sogenannten „Red Button“ auf der Fernbedienung, gelangt man in das jeweilige Portal der Rundfunkanstalt und kann dort je nach Sender, verschiedene Zusatzinformationen erhalten[7]. Diese Portale und Dienste mit Zusatzinformationen werden auch als Applikationen bezeichnet.

Abbildung 2.1: *ARD Mediathek für HbbTV-Geräte*

[6](ETSI TS 102 796, 2012)
[7](IRT, 2009)

Bietet ein Sender eine HbbTV-Applikation an, so tastet er in sein Sendesignal eine so genannte AIT[8] ein. In dieser AIT-Tabelle steht eine URL[9], welche die Adresse einer speziellen HTML[10]-Seite darstellt und vom Empfangsgerät ausgelesen wird. Wird nun der „Red Button“ auf der Fernbedienung gedrückt, gelangt das HbbTV-Gerät mithilfe der URL über den eigenen Internetanschluss zum gewünschten HTTP-Server, und kann von dort Inhalte wie Videos beziehen.

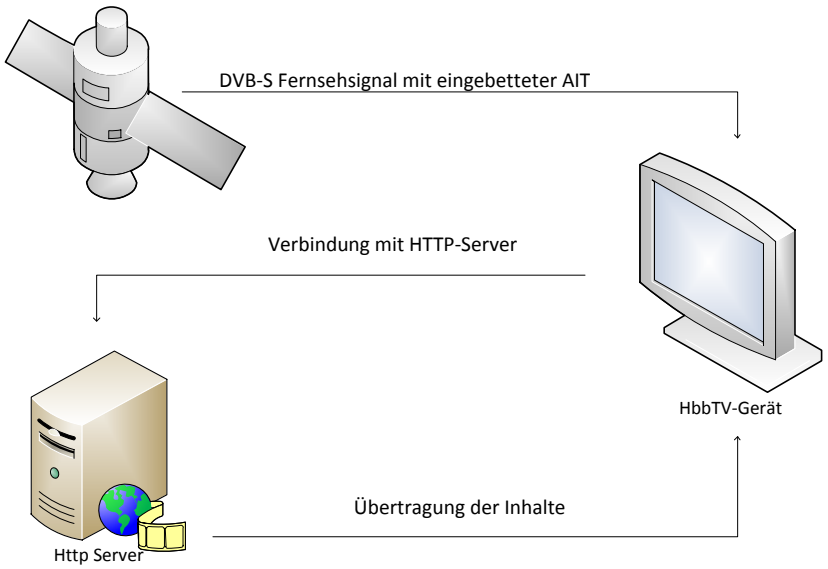


Abbildung 2.2: *Signalfluss HbbTV*

Die AIT kann in das Fernsehsignal aller üblichen digitalen Verbreitungswege eingetastet werden, ganz gleich, ob Satellit (DVB-S), Antenne (DVB-T) oder Kabel (DVB-C). Neben der URL können in der AIT weitere, senderseitige Einstellungen für die HbbTV-Applikation vorgegeben werden. So kann der Sender festlegen, ob eine Applikation automatisch startet, oder erst durch den Tastendruck des Nutzers auf den „Red Button“. Auch ob die Applikation bei einem Senderwechsel weiterlaufen darf oder nicht, kann in der AIT festgelegt werden.

2.2 HTML5

Beim Surfen durch das Internet trifft man immer häufiger auf Audio- und Videoinhalte. Durch Portale wie Youtube oder auch die Mediatheken der Rundfunkanstalten steht eine enorme Anzahl an Videos zur Verfügung, die mit einem Webbrowser am PC angeschaut werden können.

Die Einbindung von Videos in Webseiten erfolgte bisher hauptsächlich über einen FLASH-Player. FLASH ist eine Plattform der Firma Adobe, die es ermöglicht, multimediale Inhalte darzustellen und abzuspielen. In der Vergangenheit hat sich jedoch gezeigt, dass FLASH für die Wiedergabe von Videoinhalten nicht optimal ist. So benötigt FLASH verhältnismäßig viel Rechenleistung und Rescourcen des PCs. Zudem läuft FLASH in manchen Umgebungen nicht stabil, was zum Absturz des FLASH-Players oder des Browsers führen kann. Auch hatte Flash in der Vergangenheit häufig Sicherheitslücken. Ein weiterer Nachteil von FLASH ist, dass es eine nicht offene, proprietäre Technologie ist, die nur von einem Hersteller kontrolliert wird.

[8] Application Information Table
[9] Uniform Resource Locator
[10] Hypertext Markup Language

Eine standardisiert und effiziente Variante, Videos in Webseiten einzubinden, wird nun durch HTML5 möglich. HTML ist eine Auszeichnungssprache, mit der Elemente wie Texte, Bilder, Hyperlinks oder Metainformationen auf Webseiten integriert werden können. Der Browser liest den Inhalt der HTML-Seite aus und stellt diesen am Bildschirm dar. Zurzeit befindet sich das W3C in der Endphase der Standardisierung von HTML5, welche bis 2014 abgeschlossen sein soll.

Eine der wesentlichen Neuerungen von HTML5 sind die standardisierten Multimediaelemente, die in Webseiten eingebunden werden können. Mit dem so genannten „Audio"- und „Video-Tag" kann die Quelle eines Audio- oder Videofiles angegeben werden, welches dann direkt vom Browser abgespielt wird. Es sind also keine externen Plugins, wie beispielsweise ein FLASH-Player mehr nötig, um das gewünschte Video wiederzugeben. Dadurch, dass der Browser das Video direkt abspielt, ist die Auslastung zudem häufig geringer[11].

Das Video-Tag bietet verschiedene Attribute, die angeben, wie das Abspielverhalten und Erscheinungsbild des Videos aussehen soll. Durch das Attribut „controls" wird festgelegt, ob Bedienelemente wie ein Start/Stop-Button angezeigt werden sollen. Der Befehl „autostart" regelt, ob der Inhalt automatisch startet, sobald die Seite aufgerufen wird oder erst mit dem Klick auf den Play-Button. Mit „poster" können ausgewählte Vorschau-Bilder im Player angezeigt werden, bevor das Video startet. Zusätzlich kann mit „preload" festgelegt werden, ob der Inhalt sofort geladen wird, sobald die Seite aufgerufen wurde oder erst nach einem Klick auf den Play-Button.

Analog dazu kann das Audio-Tag genutzt werden, einzig das Attribut „poster" ist hier nicht definiert.

Die Quelle des Inhalts wird durch eine URL im „src"-Attribut (source) angegeben. Die URL verweist auf einen HTTP-Server, auf dem die Video- oder Audiodatei hinterlegt ist. Wird der Inhalt nun vom HTTP-Server abgerufen, werden die Daten durch einen Progessive Download (s. 2.3.2), oder auch per Adaptive Streaming übertragen.

2.3 Internet Distribution

Die Verbreitung von Medieninhalten über das Internet unterscheidet sich grundlegend von der Verbreitung von Medieninhalten über den Fernsehweg. Im Broadcast werden alle Empfänger mit einem einzigen Signal bedient. Dies kann ein DVB-S Signal sein, welches von einem Satelliten an beliebig viele Empfangsgeräte gesendet wird. Bei der Verbreitung über das Internet, muss der Sender hingegen jeden Empfänger einzeln versorgen. Deshalb spricht man hier vom so genannten „Unicast".

Generell gibt es bei einer Übertragung über das Internet immer einen Server, der die Informationen bereitstellt und einen Client, der die Informationen abruft. Die Übertragung der Informationen kann über unterschiedliche Methoden realisiert werden. Hier wird zwischen „Download" und „Streaming" unterschieden. Die Grenzen sind jedoch fließend, sie werden ausschließlich durch die genutzte Speicherdauer im Client definiert.

2.3.1 Download

Beim klassischen Download muss eine Datei vollständig vom Server heruntergeladen werden, bevor sie vom Client benutzt werden kann. Die Informationen werden beim Client gespeichert und können auch nach dem Download oft weiter genutzt oder weitergegeben werden, sofern die

[11](ix, 7/2012), S. 132

Applikation dies zulässt. Beim Download entspricht die Größe des Zwischenspeichers immer der Größe des Inhaltes selbst.

2.3.2 Progressive Download

Beim Progressive Download können schon während der Datenübertragung bereits heruntergeladene Informationen genutzt werden. Wird ein Video über einen Progressive Download angeboten, kann es abgespielt werden, noch bevor alle Informationen beim Client verfügbar sind, vorausgesetzt, die verfügbare Bandbreite ist mindestens so groß wie die Medienbitrate. Mithilfe von Metadaten ist es sogar möglich, an beliebige Stellen des Videos zu springen, die bereits heruntergeladen wurden. Außerdem kann durch den Server eine variable Gesamtspieldauer simuliert werden. In diesem Fall wird vom so genannten „Pseudo Streaming“ gesprochen.

Die Übertragung wird durch das HTTP[12]-Protokoll geregelt, über welches beispielsweise auch die Übertragung der Inhalte von Webseiten an den Browser abläuft.

Bei allen Audio- und Videoinhalten dieser Arbeit, die zu Testzwecken über ein Netzwerk übertragen wurden, ist die Übertragung durch einen Progressive Download realisiert worden.

2.3.3 Streaming

Unter Streaming versteht man die Übertragung von Informationen in nahezu Echtzeit. Wird z.B. ein Video „gestreamt“, so wird vom Server ein kontinuierlicher Datenstrom erzeugt, der vom Client angezeigt wird. Das Zwischenspeichern des gesamten Inhalts ist beim Streaming meist nicht nötig. Die übertragenen Informationen werden üblicherweise in Echtzeit genutzt. Eine Navigation innerhalb eines Videos ist jedoch beim On-Demand-Streaming für Inhalte auf Abruf dennoch möglich. Dies gilt nicht für das Live-Streaming, bei dem der Server ein Live-Signal bereitstellt, welches vom Client abgerufen werden kann. Hierbei wird der Stream erst gestartet, wenn das Video vom Client angefordert wird.

Beim Streaming wird die Übertragung mit dem RTP[13]-Protokoll geregelt. Da RTP im Gegensatz zu HTTP ein verbindungsloses Protokoll ist, können Informationen, die während der Übertragung verloren gehen, nicht wiederhergestellt oder neu angefordert werden. RTP setzt spezielle Streaming-Server voraus, was einen höheren Serverseitigen Realisierungsaufwand für eine Übertragung darstellt. Zudem kann es passieren, dass das Protokoll Ports benutzt, die von der Firewall des Computers blockiert werden und somit eine Übertragung der Informationen nicht stattfinden kann.

Aus diesen Gründen wird versucht, alternativ zum RTP-Protokoll, die Übertragung auf einen HTTP basierten Datentransfer umzustellen.

2.3.4 Adaptive Streaming

Dieser Gedanke wird beim Adaptive Streaming mit Hilfe einer Segmentierung des Datenstromes umgesetzt, wobei hier die Übertragung mit Hilfe des HTTP-Protokolls stattfindet.

Beim Adaptive Streaming liegen am Server mehrere Versionen des Videos in unterschiedlichen Bitraten vor. Bevor der Client ein Video anfordert, vermisst er zunächst die Bandbreite seiner Internetverbindung. An Hand der Messung wählt der Client ein Video aus, dessen Medienbitrate kleiner als die momentan vorhandene Bandbreite ist. Somit kann für jedes Endgerät eine

[12] Hypertext Transfer Protocol
[13] Real-Time Transport Protocol

lückenlose Übertragung des Videos gewährleistet werden. Wird also ein Video mit einer hohen Bitrate vom Client angefordert und die Bandbreite des Internetzuganges reduziert sich, wechselt der Client den Stream automatisch zu einem Video mit niedrigerer Bitrate. Es kommt also nur zu einer Verschlechterung der Videoqualität, wobei die Wiedergabe aufrecht erhalten wird[14].

In dieser Arbeit wurde die Adaptive Streaming-Übertragung aus HbbTV 1.5 nicht eingesetzt, da dafür noch keine Geräte am Markt verfügbar waren. Nach Fertigstellung der Arbeit soll jedoch eine weitere Untersuchung für HbbTV-Geräte in einer Adaptive Streaming-Umgebung hinsichtlich einer Mehrkanalton-Übertragung realisiert werden. Hierbei soll der gleiche Bildinhalt einmal mit einer geringen Datenrate (SD-TV) und Ton im Stereo-Format, sowie mit einer hohen Datenrate (HD-TV) und Ton im Mehrkanalton-Format als Video auf einem Streaming-Server vorliegen. Untersucht wird dann das Abspielverhalten des Videos bei schwankender Bandbreite und damit sich ändernden Tonformaten.

2.4 Mehrkanalton

Unter dem Begriff Mehrkanalton versteht man das Vorhandensein von mehreren verschiedenen Audiokanälen. Streng genommen stellt also auch das Stereo-Format ein Mehrkanalton-System dar, da dort zwei getrennte Audiokanäle vorliegen. Im allgemeinen medientechnischen Sprachgebrauch bezeichnet Mehrkanalton jedoch eine Erweiterung des Stereo-Format-Systems.

Diese Arbeit orientiert sich an der von der ITU standardisierten Anordnung eines Mehrkanalton-Systems[15] (s. Abb. 2.3). Hierbei werden die beiden Stereo Kanäle Links (L) und Rechts (R) um die Kanäle Center (C), Left Surround (LS), Right Surround (RS) sowie den Low Frequency Effects-Kanal (LFE) erweitert.

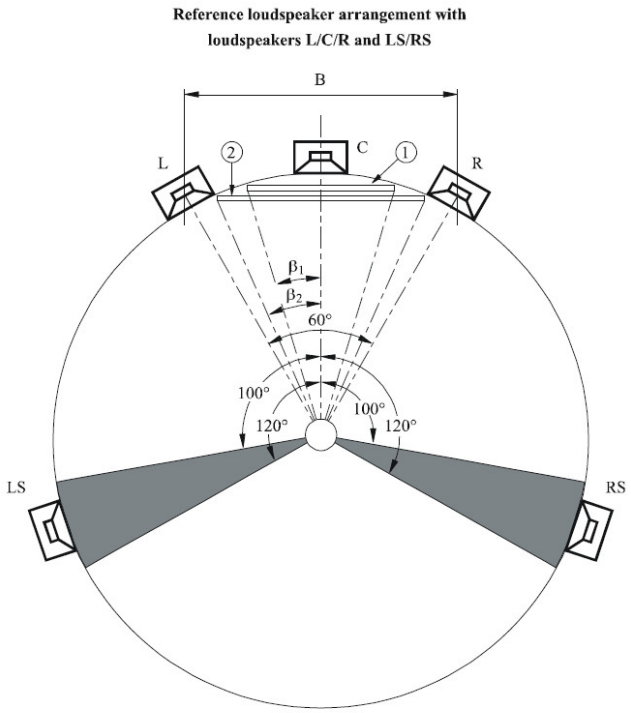


Abbildung 2.3: *Mehrkanalton Anordnung nach ITU-R BS.775-2*

[14](c't, 12/2010), S. 158ff
[15](ITU-R BS.775-2, 2006)

Diese Anordnung wird auch als 5.1-Surround-Anordnung oder 3/2LFE-System bezeichnet. Da das 5.1-System die am weitesten verbreitete Surround-Anordnung darstellt, werden in der Fernsehproduktion fast alle Surround-Mischungen an dieses System angepasst. Aus diesem Grund wurden alle Mehrkanalton-Audiosequenzen in dieser Arbeit ebenfalls im 5.1-Format erstellt.

2.5 Audiocodierung

Der Begriff Codierung beschreibt einen Vorgang, bei dem mit Hilfe eines Codes, die Symbole eines Zeichenvorrates, in die entsprechenden Symbole eines anderen Zeichenvorrates übertragen werden. Bei der digitalen Audiocodierung werden einzelne Datenwerte eines Audioformates ebenfalls in Datenwerte eines anderen Audioformates übertragen. Meist wird die Audiocodierung eingesetzt, um eine Datenkompression zu erreichen.

Die digitale Codierung von Audiosignalen ist seit den 1990er Jahren zu einem wichtigen Gebiet der Audiosignalverarbeitung geworden. Besonders im Bereich des Rundfunks und des Internets, wo die Übertragungsbandbreiten begrenzt sind, werden effiziente, meist verlustbehaftete Audiocodierungsverfahren eingesetzt, um auch mit geringen Datenübertragungsraten eine akzeptable Audioqualität am Empfänger zu gewährleisten.

Bereits die lineare Übertragung digitaler Audiodaten, wie sie beispielsweise auf einer CD[16] aufgezeichnet ist, benötigt eine enorm große Bandbreite. Da sich die Berechnung der Datenrate aus

$$Abtastfrequenz[Hz] * Abtasttiefe[Bit] = Datenrate[Bit/s]$$

ergibt, würde für die Übertragung eines einzelnen Kanals eine Bandbreite von ca. 700kb/s benötigt werden. Bei sechs Kanälen und einer Beitragslänge von einer Stunde, käme eine Datenmenge von ca. 2GByte/h zusammen. Da die Bereitstellung solcher Datenmengen teuer ist, wird versucht, durch die Audiocodierung die Datenmenge zu reduzieren, ohne das hörbare Artefakte auftreten. Mit aktuellen Kompressionsverfahren kann eine Datenreduktion eines 5.1 Surround-Audiosignals um den Faktor 100 erzielt werden[17].

Bei der Audiocodierung unterscheidet man zwischen der verlustlosen Audiocodierung und der verlustbehafteten Audiocodierung.

2.5.1 Verlustlose Audiocodierung

Die verlustlose Audiocodierung zeichnet sich dadurch aus, dass kein Qualitätsverlust des Audiosignals entsteht und eine exakte Replikation des Eingangssignals nach dem Decodierprozess erzeugt wird[18]. Um eine exakte Replikation zu ermöglichen, wird versucht die Redundanzen eines Signals zu minimieren. Als Redundanz wird das mehrfache Vorkommen der gleichen Information bezeichnet.

Da die Häufigkeit der Redundanzen je nach Signal jedoch sehr unterschiedlich sein kann, hängt der mögliche Kompressionsfaktor maßgeblich vom Inhalt des Signals ab. Ein weißes Rauschen lässt sich beispielsweise überhaupt nicht verlustlos komprimieren, ein einfaches Sinussignal hingegen sehr gut. Somit lässt sich eine bestimmte Kompressionsrate bei verlustlosen Kompressionsverfahren für natürliche Signale nicht vorab festlegen. Deshalb wurde die verlustlose Audiocodierung in dieser Arbeit auch nicht eingesetzt.

[16] Compact Disc
[17] (Dickreiter, 2008), Vgl. S.649
[18] (Zölzer, 2005), Vgl. S.279

2.5.2 Verlustbehaftete Audiocodierung

Ein weitaus höherer und vom Signalinhalt weniger stark abhängiger Kompressionsfaktor kann mit der verlustbehafteten Audiocodierung erzielt werden. Hierbei wird keine exakte Replikation des Eingangssignals erzeugt. Vielmehr wird zusätzlich zur Minimierung von Redundanzen versucht, das Signal von irrelevanten Anteilen zu befreien (sog. Irrelevanzreduktion)[19]. Bei irrelevanten Anteilen handelt es sich um Informationen, die das menschliche Gehör nicht wahrnehmen kann.

Aufgrund der hohen Kompressionsraten findet die verlustbehaftete Audiocodierung im Bereich des Rundfunks und des Internets ihren Einsatz. Auch in dieser Arbeit wurden alle Audiosequenzen verlustbehafteten Kompressionsverfahren unterzogen.

2.5.2.1 Psychoakustische Grundlagen

Mithilfe der Psychoakustik und der genauen Beschreibung des menschlichen Hörsinns, können redundante und irrelevante Informationen eines Audiosignals identifiziert werden. Je besser die psychoakustischen Effekte unseres Hörsinns beschrieben werden können, umso effizienter kann ein Codierverfahren diese irrelevanten Informationen heraus filtern und somit eine gesteigerte Datenreduktion erzielen.

Irrelevante Signalanteile sind Informationen, die von uns durch verschiedene Verdeckungseffekte unseres Hörsinns nicht wahrgenommen werden können. Diese Verdeckungseffekte werden auch als Maskierung bezeichnet. Es gibt mehrere Arten von Maskierungseffekten, die im Kontext der Audiocodierung relevant sind.

Frequenzabhängige Maskierung

Ein wichtiger Begriff der Maskierung ist die Maskierungsschwelle, auch Mithörschwelle genannt. Sie legt fest, wann ein Schallereignis vom menschlichen Gehör wahrgenommen wird. Liegt ein Schallereignis über der Maskierungsschwelle, ist dieses wahrnehmbar, wird die Maskierungsschwelle nicht erreicht, können wir das Schallereignis nicht hören. Die Mithörschwelle ist ein komplexes Phänomen, das von vielen unterschiedlichen Faktoren abhängt. Die wichtigsten Faktoren sind dabei die Frequenz, der Pegel und die Art der Schallquelle. Alle diese Faktoren müssen gemeinsam betrachtet werden, um eine Aussage über die Veränderung der Mithörschwelle treffen zu können.

Einen besonderen Fall der Maskierung stellt die Ruhehörschwelle dar. Sie definiert, welcher Schalldruckpegel bei absolut stiller Umgebung mindestens erreicht werden muss, damit ein Schallereignis wahrgenommen wird. Alle Schallquellen, die diese Ruhehörschwelle nicht erreichen, müssen somit nicht übertragen werden. Dies ist der Fall beim so genannten Quantisierungsrauschen, welches beim Wandel von analogen zu digitalen Signalen auftritt.

Treten nun lautere Schallquellen auf, verändert sich die Mithörschwelle in den benachbarten Frequenzbereichen dieser Schallquellen deutlich. Die Schallquellen werden zu sog. Maskierern, welche benachbarte Schallquellen, die einen geringeren Pegel haben, überdecken können.

Wichtig in diesem Zusammenhang ist der Begriff der Signal to Mask Ratio (SMR). Der SMR gibt die Differenz des Maskierers zur Mithörschwelle an. Er wird meist in dB angegeben.

Eine Maskierung tritt beispielsweise auf, wenn man drei schmalbandige Rauschsignale mit einem Schalldurckpegel von 60dB über den Frequenzbereich verteilt (s. Abbildung 2.4). Die Mithörschwellen, die durch diese Signale entstehen, können Schallquellen überdecken, welche

[19](Dickreiter, 2008), Vgl. S.652

je nach Frequenz einen Schalldruckpegel von bis zu 55dB aufweisen. Bei tieferen Frequenzbereichen ab 250Hz ist die Überdeckung im Frequenz- und Pegelbereich größer, sie erreicht einen SMR von bis zu 3dB. Bei höheren Frequenzen ab 4kHz nimmt sie auf bis zu 5dB ab. Diese SMR-Werte werden jedoch nur für Frequenzen erreicht, die sehr nah an der Frequenz des Maskierers liegen. Je größer der Frequenzabstand ist, desto kleiner wird der Überdeckungseffekt. Dabei ist es egal, welche Art von Schallquelle maskiert wird[20].

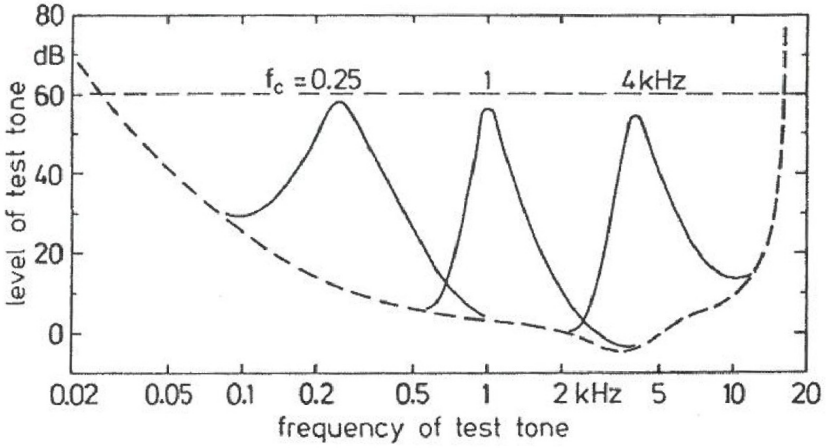


Abbildung 2.4: *Frequenzsmaskierung schmalbandiger Rauschsignale*

Nicht alle Signale haben einen derart starken Maskierungseffekt wie Rauschsignale. So maskieren einzelne Töne deutlich schwächer und verhalten sich auch hinsichtlich der Frequenzen und Lautstärken unterschiedlich. Generell maskiert ein Ton zu niedrigeren Frequenzen sehr viel schwächer als ein Rauschsignal mit dem gleichen Schalldruckpegel. Das bedeutet, dass eine Schallquelle frequenztechnisch noch näher am Maskierer sein muss, um verdeckt zu werden[21]. Zu höheren Frequenzen ist der Maskierungseffekt eines einfachen Sinustones jedoch mit dem eines rauschartigen Signals vergleichbar. Auch der SMR-Wert ist bei Tönen um einiges höher, als das bei einem Rauschen der Fall ist. Doch selbst in ungünstigen Fällen wird die Differenz zwischen Maskierer und Mithörschwelle selten größer als ca. 25dB[22].

Befinden sich mehrere Schallquellen frequenztechnisch nahe beieinander, beeinflussen sich ihre Maskierungseffekte gegenseitig. Zwei Töne, die an den Grenzfrequenzen eines schmalbandigen Rauschens liegen, erzielen einen ähnlichen Überdeckungseffekt, wie eben dieses schmalbandige Rauschen[23]. Also erzielen mehrere Maskierer eine weitere, gemeinsame Maskierung. Dabei kann man jedoch nicht von einer einfachen Addition der Maskierungseffekte ausgehen. Wie sich Maskierungen gegenseitig beeinflussen ist wiederum von den Frequenzen, den Pegelen und von der Art der Schallquellen abhängig.

Unser Hörsinn wertet Schallquellen, die in einem gemeinsamen, bestimmten Frequenzbereich liegen, stets ganzheitlich aus. Diese Frequenzgruppen, die als so genannte Critical Bands bezeichnet werden, definieren auch, wie weit die Schallquellen auseinander liegen dürfen um eine gemeinsame Maskierung zu erzielen. Ungefähr 24 dieser Bänder sind über das hörbare Frequenzspektrum verteilt, wobei der Frequenzabstand der einzelnen Bänder nicht einheitlich ist. Bis 500Hz hat ein Band die Breite von 100 Hz, ab 13kHz beträgt die Breite jedoch 3.5kHz.

Bei realen Schallereignissen kann man von einer Mischung aus rauschartigen und tonalen Schallinhalten ausgehen. Daraus kann grob geschlossen werden, dass in den einzelnen Frequenzbändern

[20](Fastel, Zwicker, 2006), Vgl. S. 64
[21](Fastel, Zwicker, 2006), Vgl. S.68
[22](Dickreiter, 2008), Vgl. S. 655
[23](Fastel, Zwicker, 2006), Vgl. S.73

der SMR-Wert zwischen 5dB und 25dB schwankt.

Die Gesamtmaskierung im Frequenzbereich ergibt sich dann aus der Summe und Überlagerungen der einzelnen Maskierungen der Critical Bands, die sich wiederum aus den Maskierungen der einzelnen Schallquellen ergeben.

Zeitabhängige Maskierung

Zusätzlich zur frequenzabhängigen Maskierung gibt es auch das Phänomen der zeitlichen Maskierung. Hierbei unterscheidet man zwischen Vorverdeckung, Simlutanverdeckung und Nachverdeckung.

Bei der Vorverdeckung werden leise Schallereignisse von lauten, nachfolgenden Schallereignissen maskiert. Die zeitliche Spanne für diese Verdeckung ist sehr kurz und liegt im Bereich von wenigen Millisekunden.

Die Simultanverdeckung entspricht im wesentlichen der frequenzabhängigen Maskierung.

Bei der Nachverdeckung werden nachfolgende, leisere Schallereignisse unhörbar. Die Verdeckungszeiten können dabei bis zu 150ms betragen. Wie lange die Verdeckung anhält, wird wieder maßgeblich von der Frequenz, der Amplitude und der Art der Schallquelle bestimmt. Die Nachverdeckung ist dem Umstand geschuldet, dass das Gehör einige Zeit benötigt, um nach einem lauten Schallereignis, wieder die Empfindlichkeit für ein leiseres Schallereignis zu erreichen[24].

2.5.2.2 Funktionsweise verlustbehafteter Kompressionsverfahren

Alle heute verwendeten verlustbehafteten Kompressionsverfahren benutzen die selben Grundbausteine. Aus diesem Grund wird in diesem Abschnitt die generelle Funktionsweise eines verlustbehafteten Codierverfahrens, anhand des MPEG[25]1 Layer III Verfahrens vorgestellt. Dieses Codierverfahren ist besser bekannt als „MP3“, welches Anfang der 1990er Jahre vom Fraunhofer-Institut für Integrierte Schaltungen (IIS) entwickelt wurde und auf psychoakustische Forschungen des IRT zurückgreift. Noch heute wird das Verfahren im Bereich des Internets und der mobilen Musikabspielgeräte (MP3-Player) eingesetzt.

Ziel eines Audiokompressionsverfahrens ist es, aus einem digitalen, linearen PCM-Signal[26] einen encodierten binären Datenstrom (Bitstream) zu erzeugen. Die Abbildung 2.5[27] zeigt die grundlegenden Elemente, die ein Kompressionsverfahren benutzt um einen Bitstream zu erstellen.

Filterbank (Mapping)

Durch die Filterbank wird das Signal so verändert, dass anschließend das psychoakustische Modell auf die gefilterten Daten angewendet werden kann. Dies passiert durch folgende Arbeitsschritte:

- Redundanzreduktion durch Frequenzanalyse oder Zeitanalyse
- Zerlegung des Signals in die Frequenzgruppen des Gehörsinns (Critical Bands)

Zunächst wird das PCM-Eingangs-Signal in kleine Blöcke unterteilt, die je nach Audiocodec bis zu 1024 Abtastwerte groß sein können. Diese Blöcke überlappen sich teilweise und werden einzeln

[24] (Dickreiter, 2008), Vgl. S.658
[25] Moving Picture Experts Group
[26] Pulse Code Modulation
[27] (ISO-IEC 11172-3, 1993), S. v

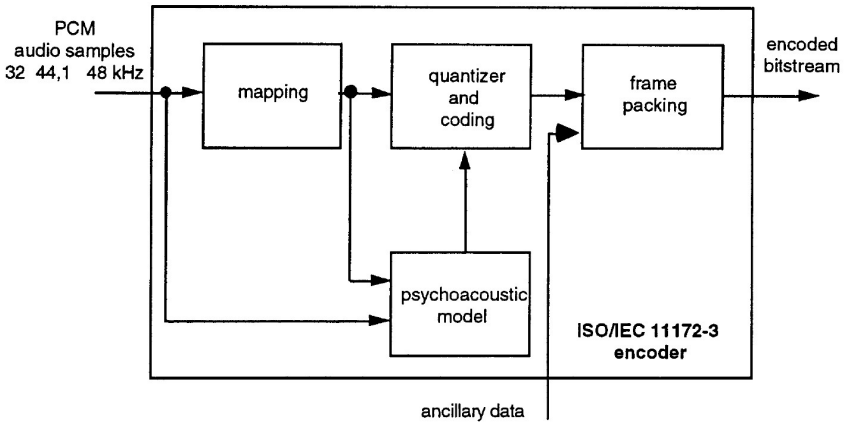


Abbildung 2.5: *Encoder MPEG 1 Layer III*

weiterverarbeitet. Je nach Eigenschaft des Blockes, kann zwischen der Frequenzauflösung und der Zeitauflösung gewechselt werden.

Bei der Frequenzauflösung findet eine Transformation des Signals vom Zeitbereich in den Frequenzbereich statt. Durch dieses Vorgehen kann analysiert werden, in welchen Frequenzsspektren ein hoher Informationsgehalt vorhanden ist und in welchen nur ein niedriger. Befinden sich in einzelnen Frequenzsspektren wenige oder gar keine Informationen, können diese beim Encodierungsprozess entweder weggelassen, oder ungenauer weiterverarbeitet werden. In diesem Fall wird von der so genannten Transformationscodierung gesprochen. In diesem Verfahren wird das volle Spektrum in bis zu 576 Sub-Frequenzsspektren unterteilt.

Die Zeitauflösung wird bei perkussiven Signalen angewendet. Also beispielsweise bei Signalen, die längere Zeit Stille beinhalten und nur für einen kurzen Zeitraum einen Ausschlag, der sich über große Teile des Frequenzsspektrums verteilt. Würde man hier die Frequenzanalyse anwenden, müsste ein großes Frequenzsspektrum übertragen werden. Da jedoch die meiste Zeit keine, oder sehr wenige Informationen im Signal vorhanden sind, wird das Spektrum an den Positionen mit einem kleinen Amplitudenausschlag nicht weiterverarbeitet, da sich dort der Informationsgehalt nicht ändert.

Mit einem weiteren Prozess in der Filterbank, wird das Signal in die nach dem Gehörsinn eingeteilten Critical Bands (siehe 2.5.2.1) aufgeteilt, um jedes Band nach den entsprechenden psychoakustischen Annahmen weiterverarbeiten zu können. Beim Layer III Verfahren wird das Frequenzspektrum zunächst in 32 Sub-Bänder gegliedert, die dann je nach Signal, noch ein mal in sechs oder 18 Teilbänder unterteilt werden[28]. Diese Zerlegung wird als Polyphase/MDCT[29] Hybrid-Filterbank bezeichnet.

Psychoakustisches Modell

Zur Bestimmung der Mithörschwelle wird zunächst untersucht, ob das Signal in den einzelnen Teilbändern einen rauschartigen, oder tonalen Charakter besitzt. Dazu wird die spektrale Zusammensetzung der einzelnen Teilbänder im Block des psychoakustischen Modells (s. Abbildung 2.5) ermittelt.

Für die Analyse wird eine Fast Fourier Transformation (FFT) durchgeführt, was zu einer Frequenzauflösung des betrachteten Teilbandes führt. Durch die Analyse der spektralen Hüllkurve

[28](ISO-IEC 11172-3, 1993), Annex C, C.1.1.2
[29]Modifizierte diskrete Kosinustransformation

jedes Bandes kann nun eine Aussage über dessen Charakter getroffen werden. Treten deutliche Ausschläge in der spektralen Hüllkurve des Bandes auf, so befinden sich in diesem Frequenzabschnitt tonale Signalanteile. Ist die Hüllkurve jedoch flach, so wird für dieses Teilband ein rauschartiger Signalinhalt angenommen. Für tiefe Frequenzen reicht die Analyse der Hüllkurve jedoch nicht aus, um zwischen tonal oder rauschartig entscheiden zu können. Deswegen werden mehrere Spektralanalysen kleinerer Sub-Abschnitte des Bandes durchgeführt. Durch den Umstand, dass sich die tonalen Hüllkurven eines Bandes vorhersagbar verhalten, kann durch das Vergleichen der Sub-Abschnitte eine Entscheidung getroffen werden[30].

Aus der Bestimmung, ob ein Teilband tonal oder rauschartig ist, lässt sich nun die Grundmaskierung jedes einzelnen Bandes errechnen. Der SMR-Wert schwankt dabei zwischen ca. 5dB und ca. 25dB, je nach Tonalität des Teilbandes (s. 2.5.2.1).

Zusätzlich kann nun noch die Maskierungswirkung eines gesamten Teilbandes auf seine benachbarten Bänder betrachtet werden. In die Betrachtung fließen jedoch nicht alle, sondern nur die acht niedrigeren und drei höheren Teilbänder mit ein. Grob betrachtet werden dazu die relevanten tonalen und rauschartigen Maskierungswerte aufsummiert und später addiert. Diese Berechnung ergibt dann für jede Frequenz die globale Mithörschwelle.

Quantisierung und Codierung

Im nächsten Block (s. Abbildung 2.5) werden die von der Filterbank kommenden Werte mit Hilfe des psychoakustischen Modells quantisiert und anschließend codiert. Durch die Errechnung des SMR in jedem Teilband kann nun entschieden werden, wie viele Bits einem Teilband zugeordnet werden müssen. Ist der SMR groß, so werden für dieses Teilband mehr Bits benötigt, als für einen kleineren SMR. Die Zuweisung der Bits pro Teilband erfolgt jedoch nicht ausschließlich aufgrund des SMR. In die Berechnung muss auch der so genannte Mask to Noise Ratio (MNR) mit einbezogen werden, der angibt, wie groß der Rauschabstand zur Maskierungsschwelle ist. Je nach Größe des MNR wird die Anzahl der Bits pro Teilband erhöht, da sonst das Quantisierungsrauschen nicht von der Mithörschwelle maskiert und damit hörbar wird[31].

Nach der Bitzuweisung pro Teilband werden die Abtastwerte linear quantisiert und anschließend codiert.

Im letzten Schritt werden die quantisierten Werte in Blöcke gepackt, die für jeden Audiocodec vordefiniert sind. Die Aneinanderreihung dieser Blöcke stellt den encodierten Datenstrom dar.

Decodierung

Beim Decodierprozess entsteht aus dem encodierten Bitstrom wieder ein PCM-Signal. Dazu wird der Codierprozess invertiert (s. Abbildung 2.6[32]).

Nachdem die Informationen der einzelnen Teilbänder aus dem Datenstrom entpackt wurden, werden die Teilbänder mithilfe einer inversen Filterbank wieder zu einem breitbandigen PCM-Signal zusammengesetzt.

[30](Dickreiter, 2008), Vgl. S. 662
[31](Zölzer, 2005), Vgl. S. 298f
[32](ISO-IEC 11172-3, 1993), S. vi

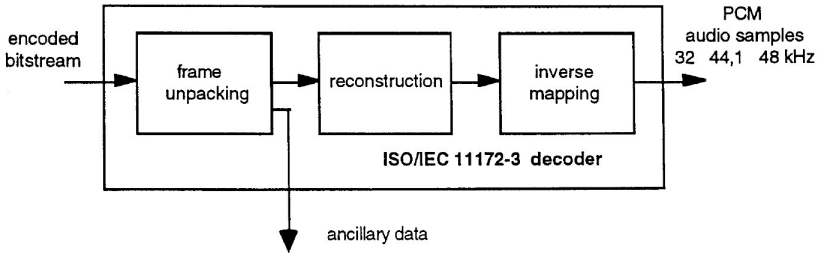


Abbildung 2.6: *Decoder MPEG 1 Layer III*

2.6 Eingesetzte Audiocodecs

In diesem Abschnitt werden alle Audiocodecs beschrieben, die in dieser Arbeit eingesetzt wurden. Für die HbbTV-Tests kamen die Codecs LC-AAC, HE-AAC, AC-3 sowie E-AC-3 zum Einsatz. Für die Web-Browser-Tests wurde ebenfalls AAC und die Codecs der Firma Dolby Laboratories verwendet, sowie die freien Codecs Ogg Vorbis und Opus.

2.6.1 Advanced Audio Coding (AAC)

Das Advanced Audio Coding-Verfahren ist eine Weiterentwicklung des MPEG1 Layer III (MP3) Verfahrens und wurde erstmals 1996 im MPEG2-Standard beschrieben. Mit kleineren Erweiterungen wurde es dann in den MPEG4-Standard übernommen und ist dort im Part 7 festgelegt[33].

AAC benutzt eine höhere Frequenzauflösung als MPEG1 Layer III. Bis zu 1024 Sub-Frequenzspektren werden in der Filterbank erzeugt und analysiert. Dies führt dazu, dass das psychoakustische Modell noch präziser angewendet werden kann, was eine weitere Datenreduktion ermöglicht. Die Anzahl der Kanäle wurde bei AAC auf bis zu 48 erweitert, die Samplingraten auf bis zu 96kHz erhöht und AAC ist in der Lage ein Eingangssignal mit 24bit Abtasttiefe zu codieren. AAC verwendet eine variable Datenrate (VBR).

Wie auch schon beim MP3-Verfahren wird nicht mehr jeder Kanal separat codiert. Kanäle mit hoher Informationsübereinstimmung, beispielsweise L und R, werden zusammengefasst. Übertragen wird dann nur noch ein Kanal (L) sowie die abweichenden Informationen des anderen Kanals (R) zum übertragenen Kanal (L). Dieses Verfahren wird auch als Joint-Stereo-Coding bezeichnet[34].

AAC wird ständig weiterentwickelt und ist in mehreren Versionen spezifiziert. In dieser Arbeit wurden die beiden folgenden AAC-Versionen verwendet.

2.6.1.1 Low Complexity Advanced Audio Coding

Low Complexity Advanced Audio Coding (LC-AAC) ist die Grundvariante des AAC-Verfahrens und setzt die in Abschnitt 2.6.1 beschriebenen Eigenschaften um.

2.6.1.2 High Efficiency Advanced Audio Coding

High Efficiency Advanced Audio Coding (HE-AAC) basiert auf dem LC-AAC, wurde jedoch durch das so genannte Spectral Band Replication-Verfahren (SBR) erweitert.

[33](ISO-IEC 13818-7, 2006)
[34](Chiariglione, 2012), S.118f

Beim SBR wird das Frequenzspektrum des Eingangssignals in zwei Hälften zerteilt. Die untere Hälfte des Spektrums wird normal quantisiert und codiert. Die obere Hälfte des Spektrums wird jedoch aus den Informationen des tiefen Anteils abgeleitet. Nach der Erzeugung des oberen Frequenzsspektrums aus dem unteren, wird dieses mit dem Eingangssignal verglichen. Die auftretenden Differenzen werden nun in den Metainformationen des Datenstroms weitergegeben. Der Decoder führt diese Generierung der hohen Frequenzen aus den tiefen ebenfalls durch und kann dann anhand der Differenzinformationen im Datenstrom die obere Spektumshälfte relativ genau wieder herstellen. Die beiden Hälften werden danach wieder zusammengefügt und ergeben das Ausgangssignal[35].

Das SBR-Verfahren kann deshalb angewendet werden, da unser Hörsinn für hohe Frequenzen weniger empfindlich ist, als für tiefe. Dieses Verfahren führt zu einer weiteren, erheblichen Datenreduktion.

2.6.2 Dolby® Laboratories

Die amerikanische Firma Dolby Laboratories steht mit ihren Codecs AC-3 und E-AC-3 in Konkurrenz zu den vom Fraunhofer IIS oder IRT entwickelten Audioformaten. So wird das AC-3-Format im DVB-Standard als Mehrkanalton-Codec verwendet, während MPEG1 Layer II als Stereo-Format eingesetzt wird. Auch auf DVDs, sowie als Tonformat im Kino, findet AC-3 seinen Einsatz.

2.6.2.1 Adaptive Transform Coder 3 (AC-3)

Das AC-3-Verfahren unterstützt die üblichen Abtastfrequenzen zwischen 32kHz und 48kHz, erzeugt einen Datenstrom mit konstanter Datenrate (CBR) und codiert bis zu sechs Audiokanäle, da es speziell für das 5.1-Audioformat entwickelt wurde. AC-3 wird von Dolby unter dem Namen „Dolby Digital“ vermarktet und unter diesem Namen beispielsweise in Fernsehsendungen kenntlich gemacht.

AC-3 unterteilt das Frequenzsspektrum lediglich in bis zu 256 Sub-Spektren. Dies ist weniger als beim Layer III-Verfahren (576) und deutlich weniger als bei den AAC-Codecs (1024). Da der Kompressionsfaktor maßgeblich von der Genauigkeit der spektralen Auflösung abhängt, kann der AC-3-Codec bei einer gleichen Datenrate nicht die selbe Audioqualität wie die AAC-Codecs liefern. Dennoch erzeugen aktuelle Implementierungen bei einer Datenrate von 448kb/s eine nahezu transparente Audioqualität. Die Quantisierung erfolgt bei AC-3 ebenfalls mit einer MDCT[36].

2.6.2.2 Enhanced - Adaptive Transform Coder 3 (E-AC-3)

E-AC-3 wird von Dolby auch als „Dolby Digital Plus“ bezeichnet und unter diesem Namen vertrieben. Der Codec stellt eine Weiterentwicklung des AC-3-Codecs dar. E-AC-3 unterstützt bis zu 13 Kanäle, erzeugt eine CBR und Abtastraten zwischen 32kHz und 96kHz.

Auch hier wird eine verbesserte Frequenzauflösung erzielt, indem nach der AC-3-Filterbank und der Unterteilung des Frequenzsspektrums in 256 Sub-Frequenzsspektren, jedes Spektrum noch ein mal in sechs weitere Teilspektren unterteilt wird. Außerdem werden von Dolby entwickelte,

[35](Chiariglione, 2012), S.127f
[36](Dickreiter, 2008), S. 689f

proprietäre Verfahren eingesetzt, welche ähnliche Ideen wie das SBR- und Joint-Stereo-Coupling-Verfahren des HE-AAC verfolgen[37].

E-AC-3 ist zu AC-3 nicht abwärtskompatibel. Eine Kompatibilität hinsichtlich der Endgeräte kann von einem E-AC-3-Decoder nur durch eine Transcodierung von E-AC-3 nach AC-3 gewährleistet werden. Diese Transcodierung ist in allen E-AC-3-Decodern implementiert und wird je nach Audioausgang des Decoders automatisch eingesetzt. So wird gewährleistet, dass jeder AC-3 fähige Audioverstärker indirekt auch mit einem E-AC-3-Signal bespielt werden kann, da dieses vom vorgeschalteten Gerät, beispielsweise einer Set-Top-Box, transcodiert werden muss.

2.6.3 Ogg Vorbis

Ogg Vorbis ist ein freier und quelloffener Audiocodec, der von der Xiph.Org Foundation entwickelt wurde.

Der Codec unterstützt Abtastraten zwischen 8kHz und 192kHz, bis zu 255 diskrete Kanäle sowie eine Abtasttiefe von bis zu 24bit. Ogg Vorbis ist in der Lage eine konstante, sowie eine variable Datenrate zu erzeugen. Zudem codiert der Codec nicht jeden Kanal einzeln, sondern fasst Kanäle zusammen und überträgt nur die Differenzen. Die beiden eingesetzten Verfahren werden „Channel interleaving“ und „Square polar mapping“ genannt. Zusätzlich benutzt die Ogg Vorbis Filterbank eine MDCT für die Quantisierung der Werte[38].

2.6.4 Opus

Opus ist ein relativ neuer, lizenzfreier Codec, der von den Firmen Skype und Mozilla Corporation speziell für den Einsatz im Internet entwickelt wurde. Die Anforderungen an den Codec waren eine möglichst kurze Latenzzeit, was für Kommunikationsplattformen wie Skype sehr wichtig ist und eine akzeptable Sprachqualität bei Datenraten unter 10kb/s. Opus wurde 2012 von der Internet Engineering Task Force (IETF) standardisiert.

Der Codec unterstützt Abtastraten zwischen 8kHz und 48kHz, Bitraten zwischen 6kb/s und 510kb/s je Kanal und bis zu 255 Kanäle. Eine CBR sowie eine VBR sind möglich. Für Datenraten bis ca. 40kb/s verwendet Opus den SILK-Codec, der von Skype entwickelt wurde und auf die Kompression von Sprachsignalen optimiert ist. Bei besonders kleinen Datenraten codiert SILK nicht das komplette Frequensspektrum, sonder filtert das Eingangssignal mit einem Tiefpass. Übertragen wird dann nur der Frequenzsbereich bis beispielsweise 4kHz. Bei höheren Datenraten kommt der CELT-Codec zum Einsatz, der nicht nur für Sprache, sondern auch für Musiksignale geeignet ist und das gesamte Frequenzspektrum überträgt[39].

Je nach Datenrate und Signalinhalt kann Opus nun entscheiden, ob das Eingangssignal mit SILK oder CELT codiert werden soll.

2.7 Videocodierung

Auch bei Videosignalen wird versucht durch den Einsatz verschiedener Codecs mit verlustbehafteten Kompressionsverfahren, eine Datenrreduktion zu erreichen. Dazu wird im Videosignal nach irrelevanten und redundanten Informationen gesucht, die durch das Kompressionsverfahren herausgefiltert werden können.

[37](Dickreiter, 2008), S. 691
[38](Vorbis I specification, 2012), S.4ff
[39](IETF RFC: 6716, 2012), S. 5ff

Eine mögliche Irrelevanzreduktion stellt beispielsweise die Unterabtastung der Farbinformationen eines Videosignals dar. Unser Auge ist für Helligkeitsunterschiede viel empfindlicher als für Farbunterschiede. Dadurch können Farbinformationen seltener als Helligkeitsinformationen abgetastet werden, ohne dass eine subjektive Verschlechterung des Videosignals auftritt.

Bei der Redundanzreduktion wird versucht, Informationen, die wiederholt auftreten zu ermitteln und im Kompressionsprozess heraus zu filtern. Redundante Informationen können innerhalb eines Bildes auftreten, oder auch zwischen aufeinander folgenden Bildern. In der Sequenz eines Nachrichtensprechers beispielsweise ändert sich die Bildinformation der einzelnen Bilder nur wenig. Hier kann eine hohe Datenreduktion erzielt werden, indem nur die Unterschiede zum vorherigen Bild angegeben und übertragen werden[40].

2.8 Eingesetzte Videocodecs

In diesem Abschnitt werden alle Videocodecs näher beschrieben, die in dieser Arbeit eingesetzt wurden.

Für die HbbTV-Test wurde der H264 Codec eingesetzt und für die Browser-Tests H264, Theora und VP8.

2.8.1 H264

H246 ist ein verlustbehaftetes Kompressionsverfahren, welches auch als Advanced Video Coding (AVC) bezeichnet wird und im MPEG4 Standard Part 10 festgelegt ist. Der Codec erzielt eine wesentlich effizientere Kompression als beispielsweise MPEG2 und wird deshalb oft bei der Übertragung von Videoinhalten im Internet eingesetzt.

Um eine Redundanzreduktion zur erzielen, wird zunächst eine Intraframe-Codierung vorgenommen. Dazu werden alle Pixel eines Einzelbildes Blöcken zugeteilt und diese Blöcke miteinander verglichen. In der Intraframe-Prädiktion wird versucht, ähnliche Blöcke innerhalb eines Frames zu detektieren und nur noch deren Differenzen anzugeben. Zudem benutzt H264 eine verbesserte Bewegungskompensation. Dabei wird versucht Formen und ihre Bewegungen zu erkennen, um die weitere Bewegung voraussagen zu können. In die Analyse fließen mehrere, bereits codierte Frames ein um Bewegungen und Objekte optimal erfassen zu können. Hier wird von der so genannten Interframe-Codierung gesprochen (s. Abb. 2.7[41]).

MPEG definiert so genannte I-Frames. Dies sind vollständig in sich codierte Bilder, die in bestimmten Zeitabständen unabhängig von Differenzinformationen übertragen werden. Die I-Frames werden als Ausgangspunkte für nahe liegende Bilder benutzt, aus denen dann die zeitlich nahe liegenden Bilder anhand des Interframe-Verfahrens berechnet werden können. H264 ist in der Lage die Interframe-Codierung bidirektional durchzuführen. Das bedeutet, dass auch Informationen aus zeitlich nachfolgenden Bildern für die Berechnung eines Frames mit einbezogen werden können. Außerdem ist es möglich, einzelnen Blöcke eines Frames unterschiedlichen Prädiktionsverfahren zu unterziehen. Dies führt zu einer weiteren Datenersparnis.

2.8.2 Theora

Wie Ogg Vorbis ist Theora ebenfalls ein lizenfreier Codec, der von der Xiph.Org Foundation entwickelt wurde.

[40](Schmidt, 2009), Vgl. S. 158ff
[41](Schmidt, 2009), S. 218

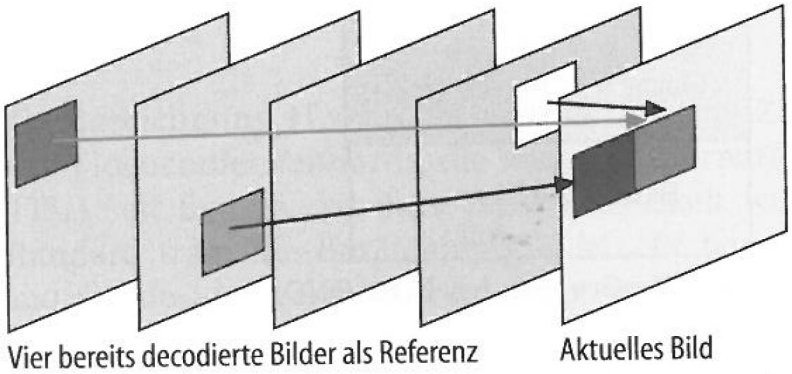


Abbildung 2.7: *Bewegungskompensation*

Ähnlich wie bei H264 werden redundante Informationen durch Intraframe- und Interfram-Verfahren analysiert und herausgefiltert. Die Bewegungskompensation wird durch eine, auf Pixel-Gruppen basierte Interframe-Codierung realisiert. Im Gegensatz zu H264 wird das Interframe-Verfahren jedoch nicht bidirektional angewendet. Die Vorhersage der Bewegung wird aus den bereits codierten Bildern abgeleitet. Die Pixel-Werte werden anschließend mit einer Diskreten Kosinustransformation quantisiert.

Theora lässt sich hinsichtlich der Qualität mit dem H263-Codec vergleichen, dem Vorgänger von H264[42].

2.8.3 VP8

VP8 ist ein von Google lizenzfrei gesteller Video-Codec, der auf Basis vorheriger kommerzieller Versionen der Firma TrueMotion weiter entwickelt wurde. Die Motivation von Google war, eine Alternative zum H264-Codec zu entwickeln. Da für H264 Lizenzgebühren anfallen, würde der Einsatz von VP8 unter anderem bei Googles Videoplattform Youtube.com zu einer erheblichen Kostenersparnis führen.

Auch VP8 bedient sich zur Datenkompression der Intraframe- und Interframe-Verfahren. Beim Interframe-Verfahren kommen, analog zu den H264 I-Frames, so genannte Key-Frames zum Einsatz. Ausgehend von den Key-Frames werden die Differenzen der nachfolgenden Bilder angegeben. Allerdings können durch das Interframe-Verfahren bereits berechnete Bilder im nachhinein zu Key-Frames werden. Eine bidirektionale Bewegungskompensation ist jedoch auch bei VP8 nicht vorhanden[43].

2.9 Container-Formate

Als Container-Format wird eine Datei bezeichnet, die mehrere, einzelne Elementardatenströme beinhalten kann. Dies können beispielsweise Video- und Audio-Datenströme sein. Dadurch, dass die Audio- und Video-Spur in einen Container verpackt werden, kann das Signal A/V-synchron decodiert werden, was zu einem abspielbaren Film führt.

Nicht jeder Audio- und Video-Codec wird von jedem Container unterstützt. Die in dieser Arbeit eingesetzten Container ergeben sich aus den jeweiligen Audio- und Video-Codecs für HbbTV

[42](Theora Specification, 2011), Vgl. S. 2.
[43](RFC 6386, 2011), Vgl. S.4ff

und den getesteten Browsern.

2.9.1 MP4

Der MP4-ISO-Container ist ein, im MPEG4 Standard Part 14[44] festgelegtes Medien File Format. Laut HbbTV-Standard müssen alle HbbTV-Geräte in der Lage sein, MP4-Dateien wiederzugeben.

MP4 kann folgende Codierungen aufnehmen:

- Audiocodecs: LC-AAC / HE-AAC / AC-3 / E-AC-3
- Videocodec: H264

2.9.2 MPEG2-TS

Der MPEG2-Transportstrom ist ein streaming-fähiges Container-Format, welches für die Übertragung von Fernsehsignalen entworfen wurde. Da es für fehleranfällige Übertragungswege ausgelegt ist, wird es heute bei der digitalen Fernsehübertragung über DVB-S, DVB-T und DVB-C eingesetzt. Der MPEG2-TS ist im MPEG2 Standard Part 1 näher spezifiziert.

Neben den Audio- und Videodaten können Zusatzinformationen, wie beispielsweise eine AIT, in das Fernsehsignal mit eingetastet werden (s. 2.1).

Der HbbTV-Standard schreibt auch eine Unterstützung des MPEG2-TS-Containers für HbbTV-Geräte, insbesondere beim Live-Streaming vor. MPEG2-TS kann folgende Codecs beinhalten.

- Audiocodecs: LC-AAC / HE-AAC / AC-3 / E-AC-3
- Videocodec: H264

2.9.3 OGG

Der OGG-Container wurde von der Xiph.org Foundation entwickelt um ein Container-Format für die lizenzfreien Codecs Ogg Vorbis und Theora bereitzustellen. OGG wurde speziell für das Streaming von Audio- und Video-Inhalten über das Internet entwickelt. Aus diesem Grund wird eine Fehlererkennung unterstützt und die Möglichkeit im Video zu navigieren. Der OGG-Container wird aus diesen Gründen von verschiedenen Browsern unterstützt und wurde in dieser Arbeit eingesetzt.

Folgende Codierungen können im OGG-Container beinhaltet sein:

- Audiocodecs: Ogg Vorbis / Opus
- Videocodec: Theora

2.9.4 WebM

Der WebM-Container ist eine von Google eingeschränkte Variante des lizenfreien Matroska Containers. WebM wird insbesondere von freien und quelloffenen Browsern favorisiert und wurde in dieser Arbeit eingesetzt.

WebM wurde speziell für folgende Codecs optimiert:

[44](ISO-IEC 14496-14, 2003), Vgl. S. vf

- Audiocodec: Ogg Vorbis
- Videocodec: VP8

2.9.5 FLV

Flash Video (FLV) ist ein von der Firma Adobe entwickeltes, proprietäres Containerformat. Der FLV-Container kann in kompilierten Shockwave FLASH-Anwendungen (SWF) mit Hilfe der FLASH-Laufzeitumgebung am PC genutzt werden. Dadurch ist es möglich, den Inhalt des Containers durch einen FLASH-Player wieder zu geben. In dieser Arbeit wurden folgende Codecs in den FLV-Container integriert.

- Audiocodecs: LC-AAC / HE-AAC
- Videocodec: H264

3 Erstellung der Testsequenzen

3.1 Webformate der ARD Mediathek

Diese Arbeit ist entstanden um speziell für die Mediatheken der Rundfunkanstalten eine Möglichkeit zu finden, Mehrkanalton-Inhalte anbieten zu können. Mediatheken bieten sich als Plattformen an, um bereits in Mehrkanalton produzierte TV-Inhalte auch online verbreiten zu können.

Aus diesem Grund wurden für die Erstellung der Testsequenzen zunächst die Codier- und Format-Einstellungen übernommen, die zur Zeit in den Mediatheken der ARD genutzt werden. Besonders im Bereich der Video-Codierung wurden hinsichtlich der Übertragung via Streaming und Progressive Download bereits ausgiebige Tests vom IRT durchgeführt, die zu mehreren Codierprofilen geführt haben. Diese Profile sind im „ARD Webtechnik-Handbuch“ als technische Richtlinie des IRT beschrieben (s. Abb. 3.1[45]).

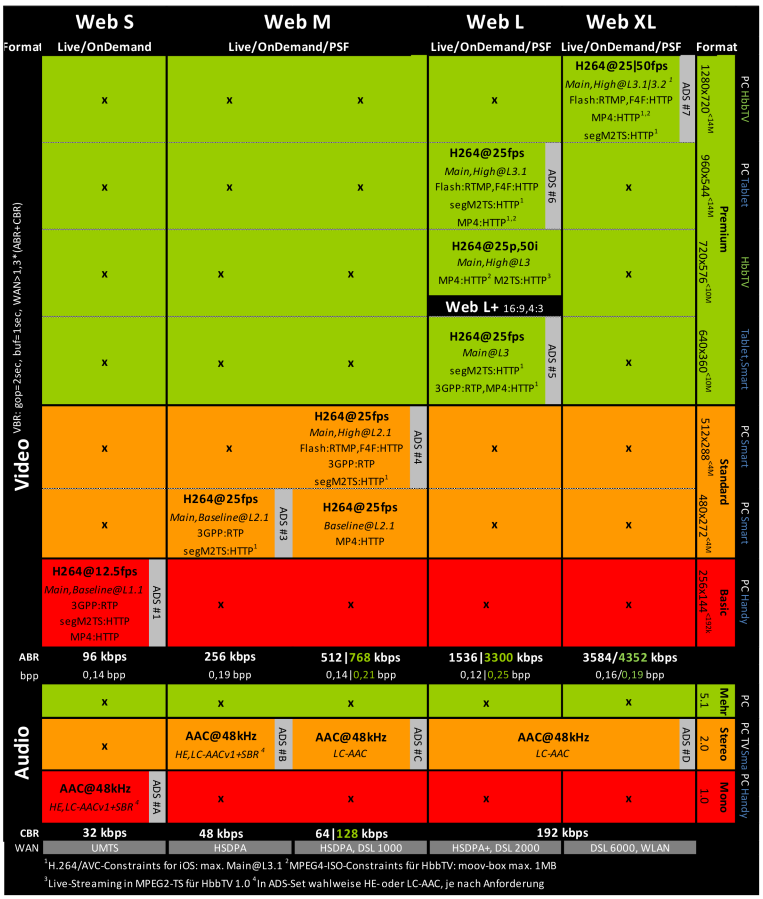


Abbildung 3.1: *Codierprofile ARD Webtechnik Handbuch*

[45](Schmalohr, 2012), S. 39

Die Abbildung 3.1 zeigt, welche Codier-Parameter für unterschiedliche Übertragungsbandbreiten vorgesehen sind. Dabei werden die Formate zur Veranschaulichung nach Kleidergrößen von S bis XL eingeteilt. Web S beschreibt beispielsweise die Audio- und Video-Codiereinstellungen für eine mobile Funkübertragung über UMTS oder EDGE. Dieses Format ist speziell für eine mobile Nutzung der Inhalte auf Handys und Smartphones ausgelegt. Dazu wird ein H264-codiertes Video mit 96kb/s Datenrate (1k bit ≙ 1000 bit) bei 12 Frames/s und einer Auflösung von 256x144 Pixel, sowie ein HE-AAC Mono-Audiostream mit 32kb/s bereitgestellt.

Um Videoinhalte mit einem HbbTV-Gerät anschauen zu können, wird eine breitbandige DSL[46]-Verbindung vorausgesetzt. Dies liegt in der Videoqualität begründet, die vergleichbar mit dem linearen Broadcast-Signal sein soll. Um eine annähernd gute subjektive Videoqualität zu erreichen, ist eine Übertragungsbandbreite von mindestens ca. 1-2Mbit/s erforderlich. Des weiteren benötigen selbst sehr effiziente mehrkanalton-fähige Audiocodecs, wie beispielsweise HE-AAC, für ein 5.1-Signal Datenraten ab 200kb/s um eine annähernd transparente Audioqualität zu erreichen. Daraus ergeben sich die Einstellungen für die Video-Codierung aus den Formaten L und XL des ARD Webtechnik-Handbuchs, welche speziell für DSL-Anschlüsse mit einer Bandbreite zwischen 2Mbit/s und 6Mbit/s definiert sind.

3.2 Codierparameter

Um ein geeignetes Codierformat für eine Mehrkanalton-Übertragung zu ermitteln, müssen mehrere Testsequenzen in möglichst vielen unterschiedlich codierten Varianten erzeugt werden. Die Testsequenzen sollten dabei so codiert sein, dass möglichst viele typische, in der Praxis vorkommende, Anwendungsfälle abgedeckt sind. Daraus muss eine präzise Auswahl gefunden werden, mit der eine hohe Interoparabilität zwischen möglichst vielen Geräten gewährleistet wird.

3.2.1 Audio-Codierparameter

Bei den Parametern für die Mehrkanalton-Audiocodierung lagen außer den Standards noch keine praktischen Vorgaben für HbbTV und PC vor. Somit sollten alle Codier-Einstellungen, die möglicherweise einmal eingesetzt werden könnten, getestet werden.

3.2.1.1 HbbTV-Tests

Bei den HbbTV-Tests stand im Vordergrund, ob die bestimmten Parameter einer Codierung (s. Tab. 3.1) das Abspielverhalten beeinflussen können.

Deshalb sollte jeder Codec in verschiedenen Bitraten vorliegen. Die Bitraten und damit die Kompression des Audiosignals, sollten so gewählt werden, dass von einer sehr starken Kompression mit hörbaren Codier-Artefakten, bis hin zu einer leichten Kompression mit annähernd transparenter Audioqualität, alle Datenraten abgedeckt sind.

Zum Einsatz kamen deshalb Datenraten zwischen 128kb/s und 640kb/s. Bei E-AC-3 wurden sogar Datenraten bis 3096kb/s eingesetzt, da der Codec auch solch hohe Datenraten unterstützt. Als Anhaltspunkt wurde die Datenrate heran gezogen, die für den AC-3-Codec im DVB-Signal verwendet wird. Für ein 5.1-Audiosignal werden 448kb/s benötigt um eine annähernd transparente Audioqualität zu erreichen.

Das codierte Audiosignal sollte zudem in konstanter und in variabler Bitrate vorliegen, sofern

[46] Digital Subscriber Line

der Audiocodec beide Bitraten-Modi unterstützt. Damit soll überprüft werden, ob eine variable Bitratenverteilung Einfluss auf die Interoperabilität des Audio-Codecs hat.

Des weiteren sollten verschieden Abtastraten getestet werden. Hier kamen die bei Videoinhalten, sowie im Rundfunkbereich und auf DVDs üblichen 48kHz zum Einsatz, sowie die bei der CD eingesetzten 44.1kHz. Der HbbTV-Standard schreibt für die meisten Codecs eine Abtastrate von 48kHz vor, für einzelne Codecs wie beispielsweise AAC gibt es jedoch keine expliziten Angaben zur Abtastfrequenz[47]. Aus diesem Grund wurden alle Codecs mit den beiden beschrieben Abtastraten getestet.

Auch sollte ein möglicherweise unterschiedliches Wiedergabeverhalten des 5.1- und Stereo-Formates untersucht werden. Wird beispielsweise ein Codec im 5.1-Format nicht unterstützt, so kann durch den Test der Stereo-Variante geprüft werden, ob der Codec generell zu Interoperabilitätsproblemen führt, oder nur die 5.1-Codierung.

Die Kombination der Codecs mit den beiden Videoformaten (s. 3.2.2) soll zeigen, ob mögliche Interoperabilitätsprobleme mit der Videoauflösung oder Bildwiederholungsrate zusammen hängen.

Tabelle 3.1: Audio-Codierparameter der HbbTV-Testsequenzen

Bitrate*	**Bitratentyp***	**Abtastrate***	**Format**	**Video**
128kb/s	CBR/VBR	48kHz/44.1kHz	5.1/2.0	HD/SD
192kb/s	CBR/VBR	48kHz	5.1	HD
256kb/s	CBR/VBR	48kHz	5.1	HD
320kb/s	CBR/VBR	48kHz	5.1	HD
448kb/s	CBR/VBR	48kHz	5.1	HD
640kb/s	CBR/VBR	48kHz/44.1kHz	5.1/2.0	HD/SD
1024kb/s	CBR	48kHz	5.1	HD
3096kb/s	CBR	48kHz	5.1	HD

* Falls vom Codec unterstützt.

3.2.1.2 Browser-Tests

Bei den Browser-Tests für PCs stand nicht mehr der Test jedes einzelnen Codier-Parameters im Vordergrund. Dies liegt darin begründet, dass die Interoperabilität zwischen Codec und Browser weniger von den Codier-Parametern, als von der Tatsache abhängt, wie ein Audio- oder Video-Inhalt in eine Web-Seite integriert ist. Außerdem konnten durch die zuvor durchgeführten HbbTV-Tests Rückschlüsse über das Verhalten der einzelnen Datenströme gezogen werden, sodass bestimmte Codier-Parameter am PC nicht getestet werden mussten.

Die eingesetzten Testsequenzen wurden mit einer Bitrate von 320kb/s, einer Abtastrate von 48kHz, je nach Codec mit einer variablen oder konstanten Datenrate codiert und einem Videoinhalt im HD-Format getestet. Bei einzelnen Browsern wurde auch ein Test im Stereo-Format durchgeführt.

[47](ETSI TS 102 796, 2012) Abschnitt 7.3.1.4

Tabelle 3.2: Audio-Codierparameter der Browser-Testsequenzen

Bitrate	Bitratentyp*	Abtastrate	Format	Video
320kb/s	CBR/VBR	48kHz	5.1/2.0	HD

* Falls vom Codec unterstützt.

3.2.2 Video-Codierparameter

Da sich die Videocodierung dieser Arbeit an den Formaten L und XL des ARD Webtechnik-Handbuchs orientiert, wurden folgende Video-Codiereinstellungen eingesetzt.

Tabelle 3.3: Video-Codierparameter der Testsequenzen

Video	Auflösung	Framerate	Wiederholungstyp	Datenrate
HD	1280x720	50	progressiv	ca. 3-4 Mbit/s
SD	720x576	25	interlaced	ca. 1-3 Mbit/s

Die HD-Auflösung wird in den Tests als Standard betrachtet, da der größte Teil der TV-Inhalte in HD produziert wird und teilweise auch schon jetzt in den ARD Mediatheken in HD abgerufen werden kann. Das HD-Format wurde in den HbbTV- und Browser-Tests eingesetzt.

Die SD-Variante wurde erstellt um festzustellen, ob möglicherweise auftretende Interoperabilitätsprobleme durch die Video-Auflösung hervor gerufen werden können. Das SD-Format wurde nur in den HbbTV-Tests eingesetzt. Dies liegt darin begründet, dass kein im Browser des PCs integrierter Player bislang in der Lage ist, Bilder im Zeilensprungverfahren adäquat anzuzeigen. Der Bildwiederholungstyp, den ein Computer nutzt ist immer progressiv.

3.3 Testkriterien

Die codierten Files sollten auf bestimmte Testkriterien hin untersucht werden. Diese Testkriterien waren:

- **Interoperabiltät**

  Unter dem Gesichtspunkt der Interoperabilität wird untersucht, ob ein, mit bestimmten Parametern komprimierter Datenstrom generell abgespielt wird. Ist die Codierung interoperabel, so können die folgenden Testkriterien angewendet werden. Besteht keine Interoperabilität, kann keine weitere Betrachtung des Abspielverhaltens mehr erfolgen.

- **Standardkonformität**

  Durch die Analyse der Standardkonformität soll ermittelt werden, ob der Decoder einen Datenstrom erzeugt, der zu den jeweiligen Rundfunk-, Industrie- und Web-Standards konform ist. Dies ist besonders bei einer Transcodierung oder bei einem so genannten „Pass-Through“ des Datenstroms wichtig. Beim „Pass Through“ wird das codierte Audiosignal an ein anderes Gerät, beispielsweise einen Audioverstärker, unverändert weitergereicht und dort decodiert.

- **Kanalzuordnung / LFE**

  Hier wird die richtige Kanalzuordnung analysiert. Als Standardbelegung wurde in dieser Arbeit die Mehrkanaltonanordnung eines WAV-Containers herangezogen. Diese Belegung ist L-R-C-LFE-LS-RS und wird von den meisten Decodern angewendet. Es wird darauf geachtet, ob Kanäle vertauscht oder weggelassen werden. Des weiteren wird untersucht, ob der LFE-Kanal im Signal enthalten ist oder nicht.

- **Downmix**

  Falls ein Downmix von einem 5.1-Signal auf ein Stereo-Signal erfolgt, wird die Art des Downmix analysiert. Es wird darauf geachtet, ob die Kanäle C, LS, RS und LFE in den Downmix mit einbezogen werden und wenn ja, wohin diese im Stereo-Panorama verteilt werden. Des weiteren wird die Lautstärke dieser Kanäle im Downmix überprüft.

- **A/V-Synchronität**

  Unter dem Kriterium der A/V-Synchronität wird untersucht, ob der Ton und das Bild synchron wiedergegeben werden. Ist dies nicht der Fall, wird zunächst darauf geachtet, ob der Ton zu früh, oder zu spät kommt. Danach wird die Dauer der Latenzzeit ermittelt und ob sich die Dauer der Latenz im Laufe der Wiedergabe ändert.

- **Lautstärke**

  Zunächst wird betrachtet, ob die verschiedenen Datenströme in unterschiedlichen Lautstärken wiedergegeben werden. Danach wird auf Unterschiede der subjektiven Lautstärke zum Quellsignal geachtet, sowie ob bei gleichem Inhalt ein Lautstärkesprung zwischen Broadcast und Broadband auftritt. Auch wird untersucht, ob sich die Lautstärke zwischen verschiedenen Audioausgängen eines Endgerätes, beziehungsweise eines Players unterscheidet.

- **Klangqualität**

  Hier wird die Klangqualität der einzelnen komprimierten Signale bei unterschiedlichen Bitraten untersucht. Auch wenn der Testaufbau nicht für einen subjektiven Qualitätstest der verschiedenen Audiocodecs bei unterschiedlichen Bitraten geeignet ist, soll hier grob abgeschätzt werden, welche Bitrate bei den unterschiedlichen Codecs eingesetzt werden muss, um eine akzeptable Audioqualität zu gewährleisten.

Aus diesen Testkriterien kann ein exakter Rückschluss darauf gezogen werden, ob die jeweilige Testsequenz vom Decoder korrekt ausgegeben wird oder nicht. Wird die Testsequenz nicht korrekt wiedergegeben, können die Ursachen analysiert werden.

3.4 Auswahl der Testsequenzen

Ziel bei der Auswahl der Testsequenzen war es, Audio- und Video-Inhalte zu finden, die sich eignen um die Testkriterien anwenden zu können. Zu diesem Zweck wurden vorhande Inhalte genutzt und eigene erstellt, die klar erkennen lassen, ob ein Testkriterium eingehalten wird oder nicht.

3.4.1 HbbTV-Tests

Folgende Testsequenzen wurden für die HbbTV-Tests ausgewählt.

- **CI-SHORT**

  Die Testsequenz CI-SHORT (Channel Identification) wurde für diese Arbeit erstellt, um die Kanalzuordnung und die A/V-Synchronität überprüfen zu können (s. 3.5). Dies wird dadurch erreicht, dass für einen Zeitraum von jeweils 2 Sekunden ein Audiokanal einzeln angesprochen wird. Es ist eine Stimme zu hören, die erklärt welcher Kanal gerade wiedergegeben wird. Der LFE-Kanal wird mit einem 100Hz Sinuston bespielt, da eine Stimme die von einem Subwoofer wieder gegeben wird, nur sehr schlecht wahrgenommen werden kann.

  Das Bild dazu besteht aus mehreren Farbbalken. In den Farbbalken wird der jeweilige Kanal, welcher gerade wiedergegeben wird, visuell mit seine Bezeichnung kenntlich gemacht. Somit kann sofort erkannt werden, ob die Kanalzuordnung vom Decoder korrekt wiedergegeben wird oder nicht.

  Des weiteren wird alle zwei Sekunden, also quasi zwischen jedem Kanalwechsel, ein schwarzes und ein weißes Frame angezeigt. Zeitgleich und nur für die Dauer dieser Blende, werden von allen sechs Kanälen Sinustöne in verschiedenen Frequenzen wiedergegeben.

  Durch diesen „Beep“ und die zeitgleiche visuelle Bildveränderung wird erkenntlich, ob der Ton und das Bild synchron wiedergegeben werden.

Abbildung 3.2: *Testsequenz: CI-SHORT*

- **Eishockey**

  Die Testsequenz Eishockey wurde aus einem Mitschnitt der Technology and Production Center Switzerland AG (TPC), einer Produktionsfirma des SRF erstellt. Der Mitschnitt zeigt eine Eishockey-Partie des Spengler-Cups 2009, die im selben Jahr vom Sender HD-Swiss ausgestrahlt wurde. Die Tonmischung setzt sich aus den Spielgeräuschen, den Fangesängen und einem Kommentator zusammen.

  Der Produktionsmitschnitt lag dem IRT im Farbraum YUV (10Bit) bei einer Auflösung von 1280x720 Pixel sowie mit einer Bildwiederholungsrate von 50 Bildern/s vor. Der Ton wurde als Mehrkanalton-Format PCM im WAV-Container bei 48kHz und einer Abtasttiefe von 24bit bereitgestellt. Durch die unkomprimierte Audioaufnahme konnten Rückschlüsse auf die Klangqualität der einzelnen Codecs im Hinblick auf Sprachsignale gezogen werden.

Außerdem wurde der Mitschnitt als Testsequenz ausgewählt, da er einen typischen Fernsehinhalt darstellt, der ausgestrahlt und auch über eine Mediathek abgerufen werden kann.

Abbildung 3.3: *Testsequenz: Eishockey*

- **LaTraviata**

  La Traviata ist ebenfalls ein Produktionsmitschnitt der TCP, welcher dem IRT vorliegt. Zu sehen ist eine live Aufführung der Oper La Traviata von Guiseppe Verdi im Gebäude des Züricher Bahnhofs.

  Der Mitschnitt liegt ebenfalls im unkomprimierten YUV-Farbraum vor. Die Auflösung beträgt 1920x1080 Pixel bei einer Bildwiederholungsrate von 25 Bildern/s. Der Ton ist als Mehrkanalton-Signal (PCM) im WAV-Container vorhanden und hat eine Abtastrate von 48kHz bei 24bit Abtasttiefe.

  Der Mitschnitt wurde als Testsequenz ausgewählt, um einen subjektiven Eindruck der Audio-Codierverfahren hinsichtlich der Qualität von Live-Musiksignalen einer Fernsehproduktion zu gewinnen.

Abbildung 3.4: *Testsequenz: LaTraviata*

- **Musik**

  Das Video Musik stellt eine weitere, selbst erstellte Testsequenz dar (s. 3.5), mit der gleichzeitig auf die A/V-Synchronität, Kanalzuordnung sowie der subjektiven Qualität der

Codecs bei Musikproduktionen geachtet werden kann.

Dazu wurden Audiosequenzen hintereinander gesetzt, die verschiedene Musik-Genres abdecken. Am Anfang der Testsequenz ist für 15 Sekunden das Lied „I can't dance“ der Band Genesis zu hören. Darauf folgt eine Sequenz des Soundtracks aus dem Film „Das fünfte Element“, gefolgt von einem Ausschnitt des Werkes „Stabat Mater“, Opus 58, von Antonin Dvořák. Zuletzt ist ein großes Publikum zu hören, dass nach einem Konzert applaudiert.

Diese Audiobeispiele wurden gewählt, da sie einerseits ein großes Spektrum verschiedener Genres abdecken und andererseits als kritische Signale betrachtet werden können, bei deren Kompression erfahrungsgemäß auch deutlich hörbare Codierartefakte auftreten können. Gerade bei Signalanteilen mit rauschartigem Inhalt, wie beispielsweise der Sequenz „Applaus“, können deutlich hörbare Unterschiede zwischen den einzelnen Codierungen auftreten.

Zwischen die Sequenzen wurde wieder ein „Beep“-Ton mit zeitgleicher schwarz-weiß Blende eingefügt, um die A/V-Synchronität zu verfolgen.

Als Bildmaterial wurden verschiedene Farbbalken gewählt, sowie ein Text, der angibt, welches Genre gerade abgespielt wird. Für die Band Genesis wurde die Bezeichnung „Pop“, gewählt, für den Soundtrack „Film“, für die klassische Aufnahme „Klassik“ und für die letzte Sequenz „Applaus“.

Abbildung 3.5: *Testsequenz: Musik*

- **Kroemer**

  Eine Testsequenz sollte aus Material bestehen, welches bereits über den Broadcast-Sendeweg in Mehrkanalton ausgestrahlt wurde um einen typischen TV-Inhalt zu testen. Des weiteren wurde eine Sendung gesucht, die aus dem Genre der Unterhaltung kommt.

  Hierzu wurde die Sendung „Krömer - Late Night Show“ aufgezeichnet, die am 13. Oktober 2012 von der ARD ausgestrahlt wurde. Das Video lag nach der Aufzeichnung im H264-Format in einer Auflösung von 1280x720 Pixel, bei 50 Bildern/s vor. Im aufgezeichneten Transportstrom-Container waren mehrere Audiosignale enthalten. Ausgewählt wurde die dritte Audiospur, welche im 5.1-Format als AC-3-Datenstrom bei einer Datenrate von 448kb/s vorlag.

  Bei dieser Testsequenz wurde besonders auf die Lippensynchronität geachtet. Denn bei sprechenden Personen wird bereits ein kleiner zeitlicher Versatz zwischen dem Audiosignal und der Mundbewegung als sehr störend empfunden. Auch bei dieser Testsequenz wurde

alle 30 Sekunden ein „Beep“-Ton und eine zeitgleiche schwarz-weiß-Blende eingefügt.

Abbildung 3.6: *Testsequenz: Kroemer*

- [redacted]

[redacted]

3.4.2 Browser-Tests

In den folgenden Browser-Tests wurde lediglich die Testsequenz CI-SHORT verwendet. Dies ist dadurch begründet, dass Testkriterien, wie beispielsweise die Klangqualität, bereits im HbbTV-Test hinreichend untersucht wurden. Die bereits gewonnenen Erkenntnisse zur Klangqualität können grundsätzlich auch auf die Browser-Tests übertragen werden, da durch den Decoder keine, oder nur eine minimale Beeinflussung der Klangqualität zu erwarten ist. Zudem stellte

sich heraus, dass in der Untersuchung der Testsequenz CI-SHORT alle anderen Testkriterien wie A/V-Synchronität, Kanalzuordnung und Downmix-Verhalten bereits hinreichend abgedeckt sind.

3.5 Komposition der Testsequenzen

Zur Komposition der einzelnen Testsequenzen wurden verschiedene Hilfsprogramme benutzt, um die Audio- und Video-Inhalte zu erstellen, zu schneiden und zusammen zu fügen. Da die Vorgehensweise in dieser Arbeit möglichst transparent und jederzeit reproduzierbar sein sollte, wurde darauf geachtet, weitestgehend mit lizenz- und kostenfreien Programmen zu arbeiten.

Die, für die Komposition der Testsequenzen benutzen Programme und Werkzeuge waren:

- **Csound**

  Csound ist eine, auf Quelltext basierte Audio-Programmiersprache, die in den frühen 1980er Jahren von Barry Vercoe am MIT entwickelt wurde. Sie wurde aus der Sprache C abgeleitet und kann als digitaler Audio-Synthesizer beschrieben werden. Mit dieser Sprache ist es möglich, beliebige Töne und Klänge mittels Klangsynthese zu erzeugen, zu routen und auch Audio-Files wiederzugeben oder zu schneiden[48].

  Die eingesetzte Csound-Version war 5.1.8.1. Die Bearbeitung und das Bouncing wurde mit dem Editor Csound-QT Version 0.7-alpha durchgeführt.

- **AviSynth**

  AviSynth ist ein skript-basiertes Werkzeug und ein so genannter Frameserver für das unkomprimierte Windows AVI[49]-Format, zur Erstellung und Bearbeitung von Videos. Mit AviSynth können einfache Bildinhalte erzeugt, oder Videos aus verschiedenen Bildern und Sequenzen zusammen gestellt werden. Die Berechnung und Erstellung eines Ausgabe-Videos nach der Interpretation des programmierten Skriptes, wurde in dieser Arbeit durch das Programm VirtualDub in der Version 1.9.11 übernommen.

- **FFmpeg**

  FFmpeg ist ein Framework zur Encodierung, Decodierung, Transcodierung und zum Multiplexen von Audio- und Video-Inhalten. Das Programm wird über die Kommandozeile gesteuert und unterstützt eine Vielzahl an Audio- und Videocodecs. Das Framework steht seit dem Jahr 2000 zur Verfügung und wurde seitdem kontinuierlich erweitert. Mit FFmpeg ist es zudem möglich Audio- und Video-Inhalte zu schneiden und beliebig zusammenzufügen.

  In dieser Arbeit wurde die Version 0.10.6 „Freedom“ eingesetzt.

- **GIMP**

  GIMP ist eine lizenzfreies Programm zur Bildbearbeitung. Es unterstützt alle gängigen Bildformate und stellt diverse Filter zur Bildmanipulation zur Verfügung.

  Es wurde in dieser Arbeit die Version 2.6.11 verwendet.

Ziel bei der Komposition der Testsequenzen war es, den Audio- und Video-Inhalt unkomprimiert bereitzustellen, sodass die unkomprimierten Inhalte anschließend in die verschiedenen Ausgabe-Formaten umgesetzt werden können. Um dies zu ermöglichen und dabei die in Abschnitt 3.2

[48] (Aikin, 2012), Vgl. S. 4f
[49] Audio Video Interleave

erwähnten Codierparameter anwenden zu können, sollten die Testsequenzen als Ausgangsdateien in folgenden Formaten vorliegen:

Der Videoinhalt sollte einmal in HD-Qualität mit 1280x720 Pixel und einer Bildwiederholungsrate von 50 Bildern/s, sowie in SD-Qualität mit 720x576 Pixel und einer Bildwiederholungsrate von 25 Bildern/s (interlaced), im YUV-Farbraum in einem AVI-Container vorliegen.

Das Audio sollte im PCM-Format einmal in Mehrkanalton mit 48kHz und 44.1kHz sowie in einer Stereo-Variante mit 48kHz jeweils in einem WAV-Container vorliegen.

Diese Vorgaben werden ebenfalls dadurch begründet, damit eine unerwünschte Änderung der Abtastrate, des Kanalformates sowie der Bildauflösung und Wiederholungsrate durch die Encoder-Implementierungen vermieden wird.

- **CI-SHORT**

  Die Komposition des Audiosignals wurde für diese Testsequenz mit Csound durchgeführt. Zunächst mussten die Audiofiles, die angeben welcher Kanal gerade abgespielt wird, in Csound eingebunden werden. Dazu wird ein so genanntes „Instrument“ gebildet, mit dem es möglich ist das File abzuspielen und auf den richtigen Kanal zu routen.

```
instr 1                                     ;Kanal L
aL, aR, aC, aLS, aRS, aLFE = 0              ;Kanalerstellung
aL diskin "Mono_Audio.wav", 1               ;Routing
outc aL, aR, aC, aLFE, aLS, aRS             ;Ausgabe
endin

instr 2                                     ;Kanal R
...
```

  Anschließend wurden die Sinustöne erzeugt, die während der schwarz-weiß Blende wiedergegeben werden. Die Klangsynthese und das Routing wurde ebenfalls innerhalb eines Instruments realisiert.

```
instr 100
aOscL oscil 1, 400, giSine                  ;Sinus
...
kEnv linen 1, giAtt, p3, giDec              ;Hüllkurve
aL = aOscL*kEnv                             ;Routing
...
outc aL, aR, aC, aLFE, aLS, aRS             ;Ausgabe
endin
```

  Zum Schluss wurde innerhalb des Score-Abschnittes von Csound angegeben, zu welcher Zeit die einzelnen Instrumente und damit die verschiedenen Audiofiles und Sinustöne, wiedergegeben werden sollen.

```
i1      0       2                           ;Audiofile
i2      2       0.04                        ;Sinustöne
...
e
```

  Nach der Komposition wurde das Audio als PCM-Signal mit sechs Kanälen und einer Abtasttiefe von 24bit heraus gerechnet. Es wurde eine 48kHz Datei und eine 44.1kHz Datei erzeugt.

Der Downmix des Audiosignals wurde mit FFmpeg realisiert. Es wurde die so genannte „Left Only / Right Only“ -Variante (Lo/Ro) verwendet. Die Einstellungen wurden so gewählt, dass dem linken Kanal das C-Signal sowie das LS-Signal mit -3dB beigemischt wurde, sowie dem rechten Kanal das C-Signal und das RS-Signal mit ebenfalls -3dB. Der LFE-Kanal wurde weggelassen.

```
ffmpeg.exe -i Mehrkanal.wav -ac 2 -dmix_mode 2 Stereo.wav
```

Das Videosignal wurde mit AvSiynth und GIMP erzeugt.

Zunächst wurden mit GIMP die Kanalbezeichnungen erstellt, die später über die Farbbalken gelegt werden sollten. Die Kanäle wurden durch die Abkürzungen L, R, C, LS, RS und LFE kenntlich gemacht und als Bilder im PNG[50]-Format erzeugt. Da PNG einen transparenten Alpha-Kanal besitzt, konnten die Abkürzungen später über die Farbbalken gelegt werden, ohne den kompletten Bildbereich mit der Hintergrundfarbe zu überdecken.

Danach wurden die Farbbalken mithilfe eines AviSynth-Skriptes mit einer Auflösung von 1280x720 Pixel und einer Bildwiederholungsrate von 50 Bildern/s erstellt.

```
fps      = 50
Video    = ColorBars(1280, 720)
```

Jedes PNG-Bild musste mit AviSynth zwei mal eingelesen werden. Einmal um die Farbinformationen zu erhalten und ein zweites mal um die Informationen des Alpha-Kanals erkennen zu können.

```
Bild     = ImageReader("Bezeichnung.png",0,500,fps,false, //
         pixel_type="RGB32")
Alpha    = ImageReader("Bezeichnung.png",0,500,fps,false, //
         pixel_type="RGB32").showAlpha(pixel_type="RGB32")
```

Die PNG-Bilder konnten nun anhand der Informationen im Alpha-Kanal transparent über die Farbbalken gelegt werden, sodass nur die jeweilige Kanalbezeichnung sichtbar war.

```
Vid_Ov   = Overlay(Video, Bild, mask=Alpha, opacity=1)
```

Bei Erstellung der schwarz-weiß Blende wurden zunächst zwei einzelne Frames mit den jeweiligen Farben in der HD-Auflösung erstellt. Anschließend wurden die beiden Frames zu einem Video zusammen gefügt.

```
Black    = BlankClip(length=1, width=1280, height=720, //
         fps=50, color=$000000)
White    = BlankClip(length=1, width=1280, height=720, //
         fps=50, color=$FFFFFF)
Vid_BW   = Black + White
```

Im letzten Schritt wurden die einzelnen Videos mit den Kanalbezeichnungen und der schwarz-weiß Blende zusammen geschnitten.

```
Vid_HD   = Vid_Ov + Vid_BW + ...
return Vid_HD
```

Nun interpretiert VirtualDub das AviSynth-Skript und erzeugt eine AVI-Datei mit dem Video im YUV-Farbraum (YV12, 10Bit) und den gewünschten Einstellungen und Inhalten.

[50] Portable Network Graphics

Um daraus ein SD-Format zu erzeugen, wurde das Video nicht neu zusammengestellt, sondern aus dem HD-Material generiert.

Zunächst muss dazu das HD-Video importiert und auf die SD-Auflösung heruntergerechnet werden. Anschließend folgt ein so genanntes „Interlacing“. Das bedeute, dass aus den 50 Vollbildern des HD-Videos 50 zeitlich aufeinander folgende Halbbilder extrahiert werden, die im AVI-Format als 25 zusammengesetzte Vollbilder abgelegt werden. Zudem wird angegeben, welches Halbbild zuerst im Video angezeigt werden soll. Hier wurde festgelegt, dass die Sequenz mit dem unteren Halbbild beginnt (BFF).

```
Video   = AviSource("Video_HD.avi").BicubicResize(720, 576)
Video   = Video.AssumeFrameBased()
Video   = Video.AssumeBFF()
Vid_SD  = Video.SeparateFields().SelectEvery(4,0,3).Weave()
return Vid_SD
```

Auch dieses AviSynth-Skript wird von VirtualDub als Datei abgespeichert. Somit wird ein SD-Video mit der gewünschten Auflösung und Bildwiederholungsrate erzeugt.

Die Testsequenz CI-SHORT liegt nun in allen gewünschten Formaten unkomprimiert vor und kann mit den verschiedenen Zielprofilen komprimiert werden.

- **Eishockey**

  Der Ton der Eishockey Testsequenz lag in sechs einzelnen WAV-Dateien vor. Der Inhalt dieser WAVs musste zunächst extrahiert und dann in ein WAV-File zusammen gepackt werden. Dies wurde wieder mit Csound realisiert, indem für jeden Kanal ein Instrument gebildet wurde, welches das jeweilige Audiofile wiedergeben sollte. Anschließend wurde wieder eine 48kHz- und eine 44kHz-Datei erzeugt.

  Der Downmix wurde ebenfalls mit FFmpeg im Lo/Ro-Verfahren erzeugt.

  Der Bildinhalt lag bereits im richtigen HD-Format vor, sodass das HD-Video nicht neu berechnet werden musste. Das SD-Format wurde analog zur Berechnung des SD-Videos der Testsequenz CI-SHORT erzeugt.

- **LaTraviata**

  Der Mehrkanalton der Testsequenz LaTraviata lag bereits als sechs kanaliges PCM-Signal in einem WAV-Container mit 48kHz Abtastrate und 24bit Abtasttiefe vor. Somit musste lediglich die fehlende Stereo-Variante mit FFmpeg erzeugt werden.

  Da der Bildinhalt jedoch in der Auflösung 1920x1080 Pixel bei einer Bildwiederholungsrate von 25 Bildern/s im Interlacing-Verfahren vorlag, musste für die gewünschte HD-Auflösung das Video mit der richtigen Auflösung und Bildwiederholungsrate neu berechnet werden.

  Dies geschah ebenfalls mit FFmpeg.

```
ffmpeg.exe -i Video.avi -vcodec rawvideo -r 50 //
           -s 1280x720 Video_HD.avi
```

  Die SD-Variante mit der Berechnung des Interlacing sowie der neuen Auflösung wurde anschließend wieder mit AviSynth und VirtualDub erzeugt.

- **Musik**

  Die Audio-Abschnitte dieser Testsequenz wurden mit Csound zusammengefügt. Dazu wurde jeder Abschnitt von einem eigenen Instrument wiedergegeben. Die Abschnitte wurden zwischen die gleichen Sinustöne eingefügt, die schon zum Test der A/V-Synchronität in

der CI-SHORT Testsequenz zum Einsatz kamen.

Nach der Komposition des Audio-Signals wurden ebenfalls die beiden Abtastraten herausgerechnet, sowie ein Downmix mit FFmpeg durchgeführt.

Das Video-Signal wurde analog zur Vorgehensweise von CI-SHORT erstellt.

- **Kroemer**

  Der Ton für diese Testsequenz lag als AC-3-codiertes Audiosignal mit einer Abtastrate von 48kHz bei einer Datenrate von 448kb/s im aufgezeichneten MPEG2-TS-Container vor. Um das gewünschte unkomprimierte Audiosignal zu erhalten, wurde der AC-3-Datenstrom aus dem Container in ein PCM-Signal gewandelt und in einen WAV-Container eingepackt. Gleichzeitig wurde der Datenstrom in 30 Sekunden lange Blöcke geschnitten, da auch bei dieser Testsequenz eine schwarz-weiß Blende mit begleitendem „Beep"-Ton eingesetzt werden sollte.

```
ffmpeg.exe -i TS.ts -map 0:3 -acodec raw -vn //
           -ss 0 -t 30 Audio_1.wav
ffmpeg.exe -i TS.ts -map 0:3 -acodec raw -vn //
           -ss 30 -t 30 Audio_2.wav
...
```

  Die Audioabschnitte wurden zusammen mit den Sinustönen anschließend ebenfalls mit Csound wieder zusammengefügt, in den verschiedenen Abtastraten heraus gerechnet und mit FFmpeg in ein Stereo-Signal gewandelt.

  Der Bildinhalt wurde wiederum mit FFmpeg zunächst von einem H264-Datenstrom in einen umkomprimierten Farbraum gewandelt und in einen AVI-Container integriert. Da die Auflösung und die Bildwiederholungsrate bereits in den gewünschten HD-Einstellungen vorlagen, mussten sie hier nicht geändert werden. Auch das Videosignal wurde in Blöcke von jeweils 30 Sekunden Länge geschnitten.

```
ffmpeg.exe -i TS.ts -vcodec rawvideo -an  //
           -ss 0 -t 30 Video_1.avi
ffmpeg.exe -i TS.ts -vcodec rawvideo -an  //
           -ss 30 -t 30 Video_2.avi
...
```

  Die Erstellung des SD-Formates wurde bei dieser Testsequenz ebenfalls mit AviSynth und VirtualDub realisiert.

3.6 Benennung der Testsequenzen

Die Anforderung an die Benennung der Testsequenzen war, dass durch den Namen erkannt werden kann, welche Testsequenz mit welchen Codier-Parametern vorliegt. Jede codierte Sequenz musste einen aussagekräftigen Namen besitzen, sodass keine Verwechslungen auftreten konnten.

Als Namenskonvention wurde folgende Bezeichnung, Syntax und Reihenfolge gewählt:

Testsequenz_Audiocodec_Audiodatenrate_Kanalformat
_Abtastrate_Bitratenmodus_Videoformat.Container

Dadurch, dass jeder Codier-Parameter im Dateinamen der Testsequenz angegeben wurde, war gewährleistet, dass die Eigenschaften jeder Datei zweifelsfrei erkannt und nicht verwechselt werden konnten. Folgendes Beispiel könnte eine Bezeichnung einer Testsequenz sein:

Kroemer_AAC-LC_192_51_48_VBR_HD.mp4

3.7 Codierung der Testsequenzen

Für die Audio- und Videocodierung sollten - schon wie bei der Komposition der Testsequenzen (s. 3.5) - wenn möglich freie Programme und Werkzeuge zum Einsatz kommen, um eine einfache Reproduzierbarkeit der Ergebnisse für Dritte zu gewährleisten. Neben dem bereits beschriebenen FFmpeg wurden folgende Programme zur Audio- und Videocodierung in dieser Arbeit eingesetzt.

- **NeroAAC**

  NeroAAC ist ein frei verfügbarer En- und Decoder der Firma NERO, der sowohl ein PCM-Signal in einen AAC-Datenstrom wandeln kann, als auch umgekehrt. NeroAAC unterstützt dabei die AAC-Profile LC-AAC und HE-AAC in verschiedenen Bit- und Abtastraten. Das Programm ist in der Lage, einen Datenstrom variabler Bitrate zu erzeugen, so wie es im Standard vorgesehen ist. Allerdings kann NeroAAC aber auch einen Datenstrom erzeugen, der sich sehr nahe an einem vorgegebenen Wert orientiert. Der Datenstrom ist zwar grundsätzlich variabel, aber die Schwankung der Datenrate ist nur noch sehr gering.

  Das Programm hat keine grafische Oberfläche und wird über die Kommandozeile gesteuert. In dieser Arbeit wurde die Version 1.5.4.0 eingesetzt.

- **Aften**

  Aften ist ebenfalls ein frei verfügbares Codier-Werkzeug um AC-3-Datenströme zu erzeugen. Der Name setzt sich aus dem ATSC[51]-Standard A/52, welcher das AC-3-Verfahren beschreibt, sowie dem Wort Encoder zusammen.

  Die Steuerung von Aften erfolgt über die Kommandozeile. Die eingesetzte Version trug den Namen SVN.

- **Dolby® Media Generator**

  Der Dolby Media Generator (DMG) ist ein Encodierungs-, Decodierungs- und Multiplexing-Werkzeug der Firmal Dolby Laboratories. Das Programm ist nicht frei verfügbar, sondern muss von Dolby erworben werden. Dem IRT wurde während der Bearbeitungszeit dieser Arbeit eine Testversion zur Verfügung gestellt. Das Programm wurde in dieser Arbeit eingesetzt, da es eines der wenigen verfügbare Programme ist, welches einen standardkonformen E-AC-3-Datenstrom erzeugen kann. Neben dem Erstellen von E-AC-3-Dateien ist der Dolby Media Generator auch in der Lage, einen AAC- sowie H264-Codec einzubinden.

  Auch dieses Programm wird über die Kommandozeile gesteuert und wurde in der Version 3.5.1 verwendet.

- **OggEnc**

  OggEnc ist ein Kommandozeilenprogramm für die Erzeugung von Ogg Vorbis-Datenströme und wird von der Xiph.org Foundation frei zur Nutzung bereitgestellt. Wie es der Standard vorschreibt, ist OggEnc in der Lage, variable- und konstante Datenströme zu erzeugen.

[51] Advanced Television Systems Committee

Die eingesetzte Version war 2.8.7.

- **OpusEnc**

  OpusEnc ist ein Kommandozeilenprogramm, was ebenfalls von der Xiph.org Foundation zur freien Nutzung bereitgestellt wird, um Opus-Datenströme zu erzeugen.

  Es wurde die Version 0.1.6 eingesetzt.

- **MP4Box**

  Das Programm MP4Box ist in der Lage, verschiedene Datenströme in einen MP4-Container zu multiplexen und wird auch über die Kommandozeile gesteuert. Die verwendete Version ist 0.5.1.

3.7.1 Audio

Die Codierung der PCM-Signale erfolgte nach den, im Abschnitt 3.2.1 vorgegebenen Codierparametern. Die Codierungen wurden dabei für alle Testsequenzen immer mit den gleichen Programmen und Codier-Einstellungen erzeugt. Somit wurde gewährleistet, dass alle Testsequenzen hinsichtlich ihrer Datenstrukturen, identisch sind.

Des weiteren wurden die bereits codierten Audio-Dateien nach der, in Abschnitt 3.6 festgelegten Bezeichnung benannt. Mit dem einzigen Unterschied, dass die Dateiendung nicht den Container beschreibt, sondern den verwendete Audiocodec des vorliegenden Elementarstromes. Somit konnten die Codierparameter der bereits codierten Audiofiles jederzeit zweifelsfrei erkannt werden, was auch für das spätere Multiplexing sehr hilfreich war. Folgende Benennung könnte die Bezeichnung eines codierten Audiofiles sein:

Kroemer_AAC-LC_192_51_48_VBR_HD.aac

Für die HbbTV-Tests wurden die Codecs LC-AAC, HE-AAC, AC-3 sowie E-AC-3 eingesetzt.

- **LC-AAC**

  Die Codierung erfolgte durch das Programm NeroAAC. Bei der Erzeugung eines Datenstroms mit variabler Datenrate muss der Parameter „-vbr“ angegeben werden. Wird eine annähernd konstante Datenrate gewünscht, wird dieser Parameter durch „-cbr“ ersetzt. Des weiteren muss die Datenrate in Bit/s angegeben werden und da die Codierung in der LC-AAC-Variante erfolgen sollte, wurde der Parameter „-lc“ verwendet.

```
1 NeroAAC.exe -vbr 192000 -lc -if Input.wav -of Output.aac
```

- **HE-AAC**

  Die Codierung der HE-AAC-Variante erfolgte identisch wie die der LC-AAC-Variante, nur dass der Parameter „-he“ angegeben wurde.

```
1 NeroAAC.exe -vbr 192000 -lc -if Input.wav -of Output.aac
```

- **AC-3**

  Zur Codierung der AC-3-Datenströme wurde Aften eingesetzt. Da der AC-3-Datenstrom im Standard schon als konstanter Datenstrom festgelegt wird, ist auch Aften nicht in der Lage einen variablen Datenstrom zu erzeugen. Die Bitrate wird durch den Parameter „-b“ definiert und in kBit/s angegeben. Der Parameter „-s“ gibt an, aus wie vielen Abtastwerten die einzelnen Blöcke bestehen, die in der Filterbank analysiert werden. Durch den

gewählten Wert „1“ erzeugt Aften Blöcke aus jeweils 512 Abtastwerten. Mit „-smix“ kann die Lautstärke der Kanäle LS und RS angegeben werden. Der Wert „2“ bedeutet, dass das Signal in seiner Lautstärke nicht verändert werden soll.

```
aften.exe -b 192 -s 1 -smix 2 Input.wav Output.ac3
```

- **E-AC-3**

  Hier wurde der Dolby Media Generator zur Codierung eingesetzt. Die Bitratensteuerung wird beim DMG mit dem Parameter „–audio-cbr-rate“ angegeben und die Datenrate selbst in kBit/s eingestellt. Obwohl der E-AC-3 Standard eine Unterstützung aller gängigen Abtastrate beschreibt, ist der der DMG nur in der Lage, einen E-AC-3 Datenstrom mit einer Abtastrate von 48kHz zu erzeugen. Aus diesem Grund konnte kein E-AC-3-Datenstrom mit einer Abtastrate von 44.1kHz erstellt werden.

  Das Dialnorm-Verfahren ist eine von Dolby entwickelte Methode um die Lautstärke eines Signals anpassen zu können. Dazu wird ein Wert über die Metadaten im Datenstrom an das Engerät weitergegeben, der angibt wie laut der Dialoglevel des Audiosignals ist. Der Decoder übernimmt diesen Wert und passt alle Audiosignale so an, dass der Dialoglevel immer die gleiche subjektive Lautstärke hat. Da der Mensch für die Lautstärke von Sprache, also auch bei Dialogen besonders empfindlich ist, soll so unabhängig vom restlichen Signal stets eine einheitliche subjektive Sprachverständlichkeit gewährleistet werden, die vom Nutzer am Endgerät über die Dialoglautstärke individuell angepasst werden kann. Dieser Wert kann mit „–input-dialnorm“ angegeben werden. Durch den eingesetzten Wert „-31“ wird die Lautstärke des Eingangssignals nicht verändert.

```
DMG.exe -i Input.wav -o Output.ec3 --audio-cbr-rate 192 //
        --audio-samplerate 48000 --input-dialnorm -31.0
```

Für die Browser-Tests kamen neben den AAC- und Dolby-Varianten auch die freien Audiocodecs Ogg Vorbis und Opus zum Einsatz.

- **Ogg Vorbis**

  Durch den Parameter „–max-bitrate“ wird angegeben, dass die variable Datenrate einen definierten Wert nicht überschreiten darf. Dieser Wert wird in kBit/s angegeben. Die Ausgangsdatei wird bei OggEnc nicht angegeben, da sie mit dem gleichen Namen in das gleiche Verzeichnis wie die Eingangsdatei gespeichert wird.

```
OggEnc.exe --max-bitrate 320 Input.wav
```

- **Opus**

  Mit dem Parameter „–hard-cbr“ wird eine konstante Bitrate in der zuvor festgelegten Größe vom Encoder erzeugt.

```
OpusEnc.exe --bitrate 320 --hard-cbr Input.wav Output.opus
```

3.7.2 Video

Für die HbbTV-Testsequenzen wurde einzig der H264-Codec eingesetzt.

- **H264**

  Aus den vorliegenden HD- und SD-Quelle wurde mit FFmpeg ein standardkonformer H264-Datenstrom erzeugt. Der zu verwendende Codec wird mit „-vcodec“angegeben. Durch

den Wert „libx264“ wird angegeben, dass die Codierungdurch durch die x264-Bibliothek durchgeführt werden soll, die als sehr effiziente Implementierung eines H264-Encoders hinsichtlich der erzielbaren subjektiven Qualität bei gegebener Bitrate gilt. Der Parameter „g“ gibt die Länge der so genannten Group of Pictures (GOP) an, die definiert, wie viele Bilder zwischen zwei I-Frames liegen dürfen. Mit „-b:v“ wird die durchschnittliche Datenrate des Videos angegeben und „-maxrate“ definiert die maximale Datenrate, in Abhängigkeit der Größe des genutzten Encoderpuffers.

```
ffmpeg.exe -i Input.avi -vcodec libx264 -g 100 //
-b:v 3000k -maxrate 4000k Output.h264
```

Bei den Browser-Tests kamen neben H264 auch die Videocodecs Theora, und VP8 zum Einsatz. Für die verschiedenen Codierungen wurde wiederum FFmpeg eingesetzt.

- **Theora**

```
ffmpeg.exe -i Input.avi -vcodec theora //
-b:v 3000k -maxrate 4000k Output.ogv
```

- **VP8**

```
ffmpeg.exe -i Input.avi -vcodec libvpx //
-b:v 3000k -maxrate 4000k Output.webm
```

3.7.3 Multiplexing

Die nun erzeugten Audio- und Video-Elementar-Datenströme müssen im letzten Arbeitsschritt in die entsprechenden Containerdateien intergriert werden. Diesen Vorgang bezeichnet man als Multiplexing.

Für die HbbTV-Tests wurde der MP4- und MPEG2-TS-Container verwendet.

- **MP4**

  Für das Multiplexing der AAC-Varianten mit den H264-Dateien wurde MP4Box verwendet. Das Programm erfordert eine erneute Angabe der Bildwiederholungsrate, da sonst automatisch ein Video mit 25 Bildern/s erstellt wird.

```
MP4Box.exe -add Video.h264 -add Audio.aac -fps 50 Output.mp4
```

  MP4Box war jedoch in der vorliegenden Version nicht in der Lage die Dolby-Codecs AC-3 und E-AC-3 in einen MP4-Container zu multiplexen. Aus diesem Grund wurde für diesen Arbeitsschritt der Dolby Media Generator eingesetzt.

```
DMG.exe -i Video.h264 -i Audio.ac3 -o Output.mp4
```

- **MPEG2-TS**

  Für die Transportströme kam FFmpeg zum Einsatz.

```
ffmpeg.exe -i Video.h264 -i Audio.aac -f mpegts //
           -codec copy Output.ts
```

  Für die Broadcast-Tests mussten die Transportströme eine konstante Datenrate aufweisen. Da die Elementardatenströme meist eine variable Datenrate besitzen, wurde der

Transportstrom mit so genannten Stuffing Bits aufgefüllt. Das bedeutet, dass die Datenrate bis zu einem definierten Wert mit leeren Bits aufgefüllt wird. Das Stuffing wird mit dem Paramteter „-muxrate“ angegeben. Liegt der Transportstrom nicht mit einer konstanten Datenrate vor, kann das Video vom Playoutserver (s. 4.1) nicht flüssig ausgegeben werden.

```
ffmpeg.exe -i Video.h264 -i Audio.aac -f mpegts //
           -muxrate 8000 -codec copy Output.ts
```

Für die Browser-Tests kamen neben MP4 die Container OGG, WebM und FLV zum Einsatz. Das Multiplexing wurde bei diesen Formaten jeweils mit FFmpeg durchgeführt.

- **OGG**

  In den Ogg-Container können sowohl ein Ogg Vorbis-, als auch ein Opus-Elementarstrom in Kombination mit Theora gemultiplext werden.

```
ffmpeg.exe -i Video.ogv -i Audio.ogg -codec copy Output.ogv
```

```
ffmpeg.exe -i Video.ogv -i Audio.opus -codec copy Output.ogv
```

- **WebM**

  Der WebM-Container unterstützt die Codecs Ogg Vorbis und VP8.

```
ffmpeg.exe -i Video.webm -i Audio.ogg -codec copy //
           Output.webm
```

- **FLV**

  Zuletzt wurden der H264- und die AAC-Codecs für die Wiedergabe in FLASH-basierten Playern in den FLV-Container gemultiplext.

```
ffmpeg.exe -i Video.h264 -i Audio.aac -codec copy Output.flv
```

4 HbbTV Versuchsaufbau

Für die HbbTV-Tests sollte eine Umgebung aufgebaut werden, in der verschiedene Videos aus einer Mediathek abgerufen werden können. Um die Versuche so praxisnah wie möglich durchführen zu können, sollte die Testumgebung technisch nahe an den bereits verfügbaren HbbTV-Angeboten realisiert werden.

Dazu wurde ein Playout-Server eingerichtet, der ein DVB-S-Signal mit HbbTV-Signalisierung erzeugt. Dieser Transportstrom mit entsprechend eingetasteter AIT wurde an sechs ausgewählte HbbTV-Geräte gesendet. Alle Geräte waren über ein lokales Netzwerk mit einem Apache-Server verbunden. So konnten diese anhand der AIT die darin eingebetteten URLs aufrufen, welche auf eine HTML-Seite des Apache-Servers verwiesen. Diese HTML-Seite realisierte eine Testapplikation, die als Test-Mediathek für die zuvor erzeugten Datenströme betrachtet werden kann. Die Applikation konnte über die Fernbedienung der Geräte gesteuert werden. Wurde nun ein Video ausgewählt, so stellte der Apache-Server den Inhalt über das Netzwerk zur Verfügung, welcher dann vom Gerät decodiert werden sollte.

Da die meisten Fernseher und Set-Top-Boxen nicht in der Lage sind Merhkanalton wiederzugeben, wurde das Audiosignal aus dem jeweiligen Gerät an einen mehrkanalton-fähigen Audioverstärker weiter geleitet und und von den angeschlossenen Lautsprechern wiedergegeben.

Die Abbildung 4.1 beschreibt den Versuchsaufbau für HbbTV fähige Fernseher.

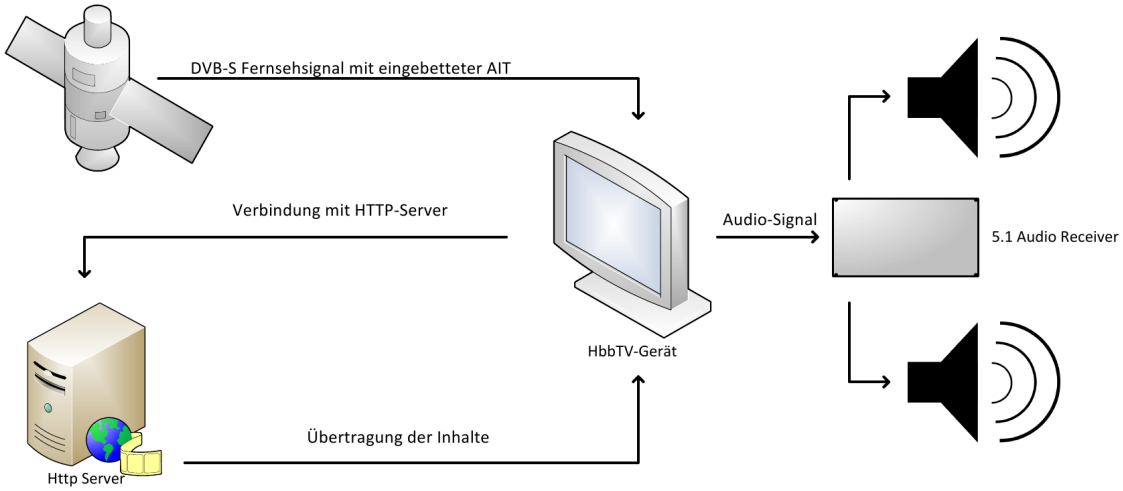


Abbildung 4.1: *Versuchsaufbau: HbbTV Fernseher*

Die Abbildung 4.3 stellt den Versuchsaufbau für HbbTV fähige Set-Top-Boxen dar.

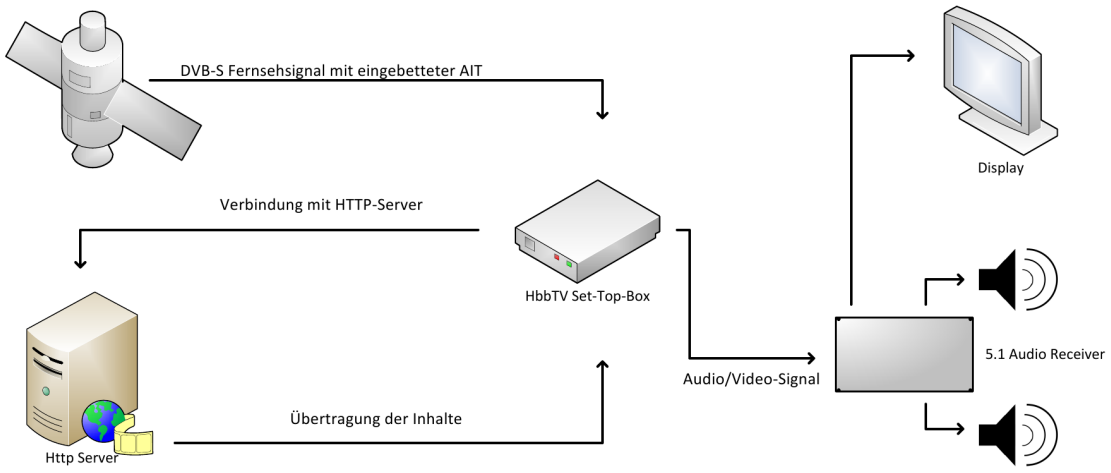


Abbildung 4.2: *Versuchsaufbau: HbbTV Set-Top-Box*

4.1 Elemente der Teststrecke

MPEG2-Transportstrom-Server

Für die Tests sollte ein standardkonformes DVB-S-Signal mit eingebetteter AIT erstellt werden. Dazu wurde ein Computer genutzt, der mit Hilfe einer DVB-S Modulator Karte der Firma „DekTec“ in der Lage war, einen MPEG2-Transportstrom zu erzeugen.

Die Zusammenstellung des Signals wurde mit der „DVB Playout Server“-Software des IRT realisiert. Mit diesem Programm ist es möglich, den Audio- und Videoinhalt zu bestimmen, der im Transportstrom beinhaltet sein soll. Des weiteren können die Parameter der AIT verändert, sowie die gewünschte URL in das Signal eingebettet werden. Die AIT wurde so konfiguriert, dass die Applikation nach einem Wechsel zum eigenen Testservice automatisch gestartet wurde und damit die Betätigung des „Red Button“ nicht mehr erforderlich war.

Mit Hilfe des Playout-Servers (s. Abb. 4.3) wurden vergleichbare Transponder-Einstellungen gewählt, wie die der öffentlich rechtlichen Programme, die über den ASTRA-Satelliten auf 19.2° Ost ausgestrahlt werden. Somit gab es für die HbbTV-Geräte praktisch keinen Unterschied zwischen dem Test- und einem frei empfangbaren Fernsehsignal.

Apache HTTP-Server

Die CE-HTML-Seiten, sowie die einzelnen abzurufenden Videos wurden auf einem Apache-Server im Testnetzwerk abgelegt. Die Software entstammt eines freien und quelloffenen Projektes, dass von der Apache Software Foundation entwickelt wird. Die Apache-Server haben die höchste Verbreitung im Internet. Dieser Umstand trug weiter dazu bei, dass die Testumgebung so praxisnah wie möglich gestaltet werden konnte.

Der Server wurde auf einer 64 Bit-Plattform mit „Windows 7 Professional“-Betriebssystem aufgebaut.

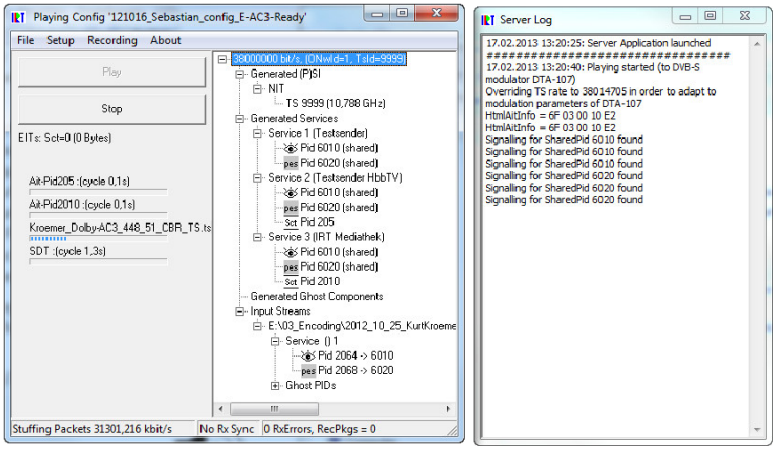


Abbildung 4.3: *DVB Playout Server*

HbbTV-Test Applikation

Die Test-Applikation, welche auf dem Server vorgehalten wurde, besteht aus einer HbbTV-konformen CE-HTML-Seite. Die abzuspielenden Inhalte müssen zunächst in eine XML[52]-Seite integriert werden. Auf dieser XML-Seite wird die Quelle des Inhalts sowie deren Name und Mime-Type angegeben. Der Mime-Type (type="video/Format") beschreibt die Codecs und das Containerformat, in denen das jeweilige verlinkte Video vorliegt. Das folgende Beispiel beschreibt die Einbindung eines Videos auf einer XML-Seite:

```
<item>
<title>AAC-LC</title>
  <pubDate/>
  <description>320kbps, 5.1, 48kHz</description>
  <link type="video/mp4">http://Video.mp4</link>
<guid/>
</item>
```

Dazu liest eine JavaScript-Funktion der Web-Seite diese Informationen aus und reicht sie an einen Player weiter, der wiederum den HbbTV-Geräten die Datenströme zur Decodierung bereitstellt.

Abbildung 4.4 zeigt die Test-Applikation, wie sie am HbbTV-Gerät angezeigt wurde.

Mit den Fernbedienungen der Geräte konnte nun in der Applikation navigiert werden. „Next" rief eine weitere Seite mit eingebundenen Videos auf. Mit „Go Back" wurde die Applikation und damit auch der HbbTV-Modus beendet und man gelangte zurück zum laufenden Satelliten-Fernsehsignal.

[52] Extensible Markup Language

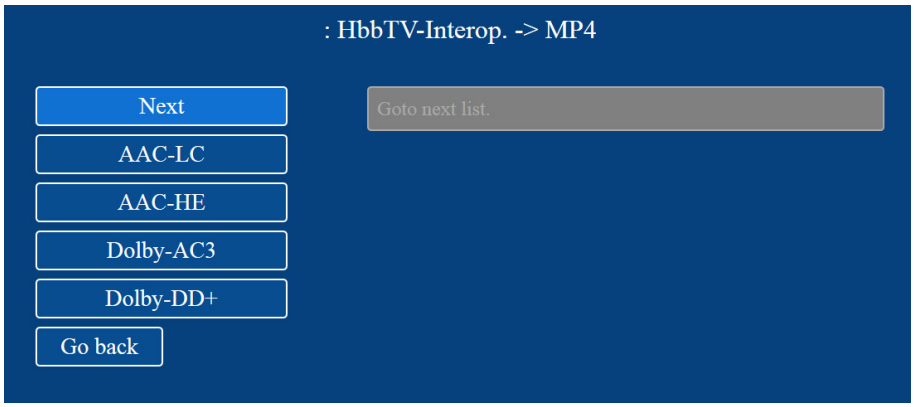


Abbildung 4.4: *HbbTV Test-Applikation*

Audio Wiedergabe

Die Mehrkanalton-Inhalte der HbbTV-Tests wurden von einem AVR-3312 Audio Receiver der Firma „Denon“ wiedergegeben. Dieser Receiver ist in der Lage neben dem PCM-Format die Audiocodierungen AC-3 und E-AC-3 zu decodieren und wiederzugeben. Des weiteren wird die Codierung, sowie das Kanalformat des am Eingang vorliegenden Datenstromes angezeigt. Der Receiver verfügt neben mehreren HDMI[53]-Schnittstellen auch über optische und koaxiale S/PDIF[54]-Eingänge. Somit stand für jeden verfügbaren Audioausgang der HbbTV-Geräte ein passender Audioeingang am Receiver zur Verfügung. Des weiteren ist der AVR-3312 in der Lage von einem HDMI-Eingangssignal den Ton zu extrahieren, diesen wiederzugeben und das Bildsignal über den HDMI-Ausgang des Receivers einem externen Display zur Verfügung zu stellen.

Abbildung 4.5: *HbbTV Testaufbau: Denon / Transportstrom-Server*

Zum Abhören kamen Lautsprecher das 5.1-Systems „VM 7“ der Firma „Bowers&Wilkins“ zum Einsatz. Da mit den Tests keine belastbaren qualitativen Aussagen gemacht werden sollten, war eine professionelle Studio-Abhörumgebung nicht erforderlich.

Bei den HbbTV-Tests wurde der AVR-3312 über die entsprechenden Audioausgänge des Gerätes angesteuert. Für die angeschlossenen Fernseher wurden in der Regel die dort vorhandene S/PDIF-Schnittstellen zur Audioübertragung genutzt.

[53] High Definition Media Interface
[54] Sony/Philips Digital Interface

Bei den getesteten Set-Top-Boxen sollten alle verfügbaren Audioausgänge überprüft werden. Über HDMI kann sowohl das Bild, als auch der Ton übertragen werden. Die Schnittstelle ermöglicht einerseits die Weitergabe des komprimierte Datenstroms per „Pass Through“, als auch eine unkomprimierte Übertragung eines 5.1-Signals im PCM-Format. Bei den Tests wurde das Audio- und Videosignal über den HDMI-Ausgang in den Verstärker geleitet. So konnte der Ton vom Receiver wiedergegeben und das Bild an ein externes Display weitergeleitet werden.

Bei der Audioübertragung über die S/PDIF-Schnittstellen wurde der Ton über das jeweilige S/PDIF-Kabel an den Receiver weitergeleitet. S/PDIF ist in der Lage einen komprimierten 5.1-AC-3-Datenstrom weiterzuleiten („Pass Through“), sowie ein PCM-Signal im Stereo-Format zu übertragen. Das Bild wurde von der Set-Top-Box über HDMI ausgegeben und in den Tests ebenfalls durch den Receiver geleitet. Somit musste die Verkabelung nicht für jeden Test geändert werden, sondern der zu testende Audioausgang konnte durch die Umschaltung der Eingänge am Receiver gewechselt werden.

Dolby® Bitstream Analyzer DM100

Der DM100 ist ein Gerät der Firma Dolby Laboratories, das in der Lage ist AC-3- und PCM-Datenströme zu analysieren. Der Datenstrom wird vom Gerät ausgelesen und auf seine Standardkonformität, Datenrate, Abtastrate, Kanalkonfiguration und Metadaten hin untersucht. Bei den Tests wurde der optische S/PDIF-Eingang des DM100 genutzt.

4.2 Auswahl der Testgeräte

Die Tests der HbbTV-Geräte wurden in zwei Teile gegliedert. Zunächst wurden HbbTV-Geräte getestet, die bereits auf dem Markt erhältlich sind. Dabei sollten drei Fernseher sowie drei Set-Top-Boxen untersucht werden, um ein möglichst breites Spektrum an Geräten zu testen. Die Testgeräte sollten zusätzlich aus verschiedenen Preisklassen, von „preiswert“ bis „premium“ stammen.

Im zweiten Teil wurden im Rahmen des 12. HbbTV-Interoperabilitätsworkshops, der Anfang Dezember 2012 vom IRT organisiert wurde, mehrere HbbTV-Geräte getestet, die sich noch in der Entwicklungsphase befinden. Da es während des Workshops nicht möglich war, die Geräte in die aufgebaute Testumgebung zu integrieren, wurden die Datenströme der Ausgänge anhand des DM100 analysiert. Über den Miniklinke-Stereo-Ausgang des DM100 konnte mit einem Kopfhörer, neben der Überprüfung der technischen Parameter, auch auf die A/V-Synchronität sowie die Lautstärke der wiedergegeben Datenströme geachtet werden. Durch die zuvor durchgeführten Tests der bereits erhältlichen Geräte und die daraus gewonnen Erkenntnisse konnte auch ohne die Testumgebung zuverlässig bestimmt werden, welche Codecs unterstützt werden und wie diese sich bei der Wiedergabe verhalten.

Des weiteren stand bei diesen Tests nicht die Interoperabilität der einzelnen Codecs im Vordergrund, sondern auch der Austausch mit den verschiedenen Geräteherstellern. So sollten diese für das Thema der Mehrkanalton-Interoperabilität sensibilisiert werden, da dieses Thema in der Geräteentwicklung bislang meist nur eine untergeordnete Rolle spielt.

4.2.1 Geräte am Markt

Folgende HbbTV-Fernseher wurden getestet:

ITT LCD 42-3475

Der ITT-Fernseher ist im unteren Preissegment angesiedelt. Als digitaler Audioausgang steht bei diesem Gerät eine optische S/PDIF-Schnittstelle zur Verfügung. Das Gerät wurde in der Software-Version V.0.11.6.1 getestet.

Abbildung 4.6: *HbbTV Testgerät: ITT LCD 42-3475*

Toshiba REGZA - 32SL883G

Dieses Testgerät stellt einen HbbTV-Fernseher der mittleren Preisklasse dar. Zur Audioübertragung stehen ein HDMI-Ausgang, sowie eine S/PDIF-Schnittstelle zur Verfügung. Da das Testgerät keinen DVB-S Eingang besitzt, wurde der Transportstrom mit einer DVB-T Signalisierung erzeugt, der vom Gerät empfangen werden konnte. Dazu wurde eine DVB-T Modulator Karte der Firma „DekTec“ eingesetzt. Die Komposition des Fernsehsignals erfolgte ebenfalls mit dem in Abschnitt 4.1 beschriebenen Playout-Server.

Abbildung 4.7: *HbbTV Testgerät: Toshiba REGZA*

Loewe ART 40 3D DR+

Der dritte getestete Fernseher ist im oberen Preissegment angesiedelt. Als digitaler Audioausgang konnte die koaxiale S/PDIF-Schnittstelle genutzt werden. Die getestete Software-Version war V.7.1.0.

Abbildung 4.8: *HbbTV Testgerät: Loewe ART 40 3D DR+*

Des weiteren wurden diese Set-Top-Boxen in die Testumgebung integriert:

SMART CX-10 HD+

Die Smart Set-Top-Box stellt ein Testgerät der unteren Preisklasse dar. Das Gerät verfügt über eine HDMI-Schnittstelle sowie einen optischen und einen koaxialen S/PDIF-Ausgang. Die Set-Top-Box wurde in der Software-Version V1_37_2_43 ausführlich getestet. Nach den Tests war die Version V1_38_2_47 verfügbar, die ebenfalls, wenn auch nicht im gleichen Umfang, getestet wurde.

Abbildung 4.9: *HbbTV Testgerät: SMART CX-10 HD+*

HUMAX iCord HD+

Die Humax Set-Top-Box ist im mittleren Preissegment angesiedelt. Neben einem HDMI-Ausgang, stellt die Humax-Box einen optischen S/PDIF Ausgang zur Verfügung. Die getestete Software-Version ist GHSNA 1.02.12.

Abbildung 4.10: *HbbTV Testgerät: HUMAX iCord HD+*

TechniSat DIGIT ISIO S

Die letzte getestete Set-Top-Box ist ist im oberen Preissegment zu finden. Neben der üblichen HDMI-Schnittstelle bietet die Technisat Set-Top-Box auch einen optischen und einen koaxialen S/PDIF-Ausgang. Das Gerät wurde in der Software-Version 2.52.0.1 (2303) BL3 (347) getestet.

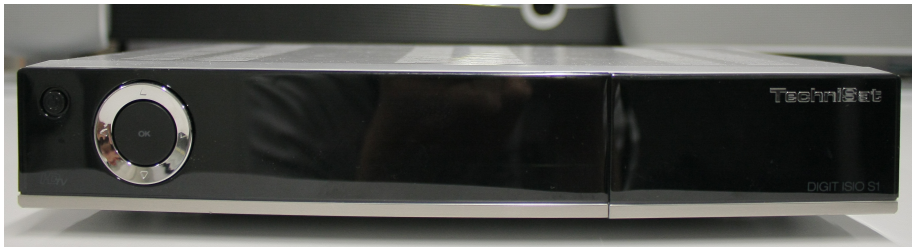


Abbildung 4.11: *HbbTV Testgerät: TechniSat DIGIT ISIO S*

4.2.2 Entwicklunsgeräte

Folgende Geräte wurden auf dem HbbTV-Interoperabilitätsworkshop getestet:

Tabelle 4.1: Getestete Entwicklungsgeräte

5 HbbTV Versuchsdurchführung

In diesem Kapitel werden die Versuchsdurchführung, sowie die daraus gewonnenen Erkenntnisse beschrieben. Die ausführlichen Testergebnisse im Einzelnen befinden sich im Anhang (s. XXII).

5.1 Test der am Markt verfügbaren Geräte

Für die Tests der einzelnen Geräte wurden verschieden codierte Dateien aus den Testsequenzen ausgewählt, die zusammen das gesamte Testspektrum abdecken sollten. Somit musste nicht jede Testsequenz in allen vorliegenden Codierungen getestet werden. Dies war auch nicht nötig, da einzelne Testkriterien schon durch eine Testsequenz hinreichen untersucht werden konnten. So musste beispielsweise die A/V-Synchronität nach den Tests der Sequenz CI-SHORT nicht erneut durch die Sequenz Musik getestet werden.

Des weiteren lag jedes Kompressionsverfahren mit mindestens 14 verschiedenen Codier-Einstellungen vor. Wäre jede Codier-Einstellung anhand der fünf Testsequenzen auf den sechs Geräten über die bis zu drei Audioausgänge getestete worden, hätte dies zu ca. 1200 Testfällen geführt, die manuell im gegebenen Zeitraum nicht zu bewältigen gewesen wären.

Aus diesem Grund wurden folgende Kombinationen an Sequenzen und Codierungen untersucht.

Tabelle 5.1: Testkombinationen

Codecs/Container	Bitrate	Freq.	Format	Video	Testsequenz
Alle/MP4	128kb/s	48kHz	5.1	HD	LaTraviata
Alle/MP4	192kb/s	48kHz	5.1	HD	Musik
Alle/MP4	256kb/s	48kHz	5.1	HD	Eishockey
Alle/MP4	320kb/s	48kHz	5.1	HD	CI-SHORT
Alle/MP4	448kb/s	48kHz	5.1	HD	Kroemer
Alle/MP4	640kb/s	48kHz	5.1	HD	CI-SHORT
E-AC-3/MP4	1024kb/s	48kHz	5.1	HD	Eishockey
E-AC-3/MP4	3096kb/s	48kHz	5.1	HD	Eishockey
AAC + AC-3/MP4	128kb/s	44.1kHz	5.1	HD	CI-SHORT
AAC + AC-3/MP4	640kb/s	44.1kHz	5.1	HD	CI-SHORT
Alle/MP4	128kb/s	48kHz	2.0	HD	Musik
Alle/MP4	640kb/s	48kHz	2.0	HD	Musik
Alle/MP4	128kb/s	48kHz	5.1	SD	LaTraviata
Alle/MP4	640kb/s	48kHz	5.1	SD	LaTraviata
Alle/MPEG2-TS	128kb/s	48kHz	5.1	HD	CI-SHORT
Alle/MPEG2-TS	640kb/s	48kHz	5.1	HD	CI-SHORT
Alle/Audio Only	128kb/s	48kHz	5.1	HD	Musik
Alle/Audio Only	640kb/s	48kHz	5.1	HD	Musik

Diese 64 Kombinationen wurden an allen HbbTV-Geräte über jeden Audioausgang getestet.

ITT LCD 42-3475

Die folgende Tabelle 5.2 stellt die Testergebnisse des ITT-Fernsehers dar. Die Legende zur Tabelle ist am Ende des Abschnittes 5.1 zu finden.

Der ITT-Fernseher war leider nicht in der Lage, Mehrkanalton-Inhalte korrekt wiederzugeben. Bei jeder Mehrkanalton-Codierung wurde ein Stereo-Downmix seitens des Decoders durchgeführt, der als PCM-Datenstrom ausgegeben wurde. Jedoch waren alle Codecs nutzbar, sodass der Ton zumindest immer decodiert und wiedergegeben wurde. Die Wiedergabelautstärke war bei allen Tests ca. 6dB geringer, als die Lautstärke des gleichen Inhalts im Broadcast-Signal.

Tabelle 5.2: Testergebnis: ITT LCD 42-3475

Codec	Interop.	Ausgang	Kanalz.	Lauts.	Sync.
LC-AAC	Ja_1	PCM_1	2.0_1	$—_1$	✓
HE-AAC	Ja_1	PCM_1	2.0_1	$—_1$	✓
AC-3	Ja_1	PCM_1	2.0_2	$—_1$	✓
E-AC-3	Ja_2	PCM_1	2.0_1	$—_1$	✓

Toshiba REGZA

Der Toshiba Regza-Fernseher war in der Lage jede getestete Codierung in Mehrkanalton wiederzugeben. Dies wird durch eine Transcodierung der AAC- und des E-AC-3-Ströme nach AC-3 realisiert. Bei AAC sowie AC-3 werden Ton und Bild nicht zeitgleich decodiert. Dieser Versatz kann jedoch in beide Richtungen zeitlich manuell am Gerät ausgeglichen werden. Die Wiedergabelautstärke wurde durch die Transcodierung nicht verändert.

Tabelle 5.3: Testergebnis: Toshiba REZA

Codec	Interop.	Ausgang	Kanalz.	Lauts.	Sync.
LC-AAC	Ja_3	$AC\text{-}3_2$	5.1	✓	X_1
HE-AAC	Ja_3	$AC\text{-}3_2$	5.1	✓	X_1
AC-3	Ja_4	$AC\text{-}3_1$	5.1	✓	X_2
E-AC-3	Ja_4	$AC\text{-}3_2$	5.1	✓	✓

Loewe ART 40 3D DR+

Das Testgerät von Loewe war nur mit dem AC-3-Codec bei einer Abtastrate von 48kHz in der Lage Mehrkanalton-Inhalte wiederzugeben. Die AAC-Varianten führten zu einem Downmix und E-AC-3 wurde gar nicht wiedergegeben. Einzig der AC-3-Codec wurde in der korrekten Lautstärke decodiert.

Tabelle 5.4: Testergebnis: Loewe ART 40 3D DR+

Codec	Interop.	Ausgang	Kanalz.	Lauts.	Sync.
LC-AAC	Ja_1	PCM_2	2.0_1	—$_1$	✓
HE-AAC	Ja_1	PCM_2	2.0_1	—$_1$	✓
AC-3	Ja_1	AC-3_1	5.1	✓	✓
E-AC-3	Nein	PCM_2			

SMART CX-10 HD+

Die Set-Top-Box war nach dem Update auf die Software V38 in der Lage, beide Dolby-Codecs AC-3 und E-AC-3 korrekt in Mehrkanalton zu nutzen. E-AC-3 wurde dabei durch das Gerät nach AC-3 transcodiert und so an den Receiver weitergeleitet. Auch über die HDMI-Schnittstelle fand eine Transcodierung statt, obwohl hier eigentlich der E-AC-3 Datenstrom direkt zum E-AC-3-fähigen Receiver geleitet werden könnte. Die AAC-Varianten wurde zwar wiedergegeben, jedoch mit erheblichen Lautstärke-Unterschieden und nur als Downmix.

Tabelle 5.5: Testergebnis: SMART CX-10 HD+

Codec	Interop.	Ausgang	Kanalz.	Lauts.	Sync.
LC-AAC	Ja_1	PCM_1	2.0_1	—$_2$	✓
HE-AAC	Ja_1	PCM_1	2.0_1	—$_2$	✓
AC-3	Ja_1	AC-3_1	5.1	✓	✓
E-AC-3	Ja_5	AC-3_3	5.1	✓	✓

HUMAX iCord HD+

Auch die Humax-Box konnte beide Dolby-Codecs in Mehrkanalton in der richtigen Lautstärke für Breitband nutzen. Bei den AAC-Varianten kam es wiederum zu einem Downmix und es ergaben sich erhebliche Lautstärkeunterschiede zwischen dem HDMI- und S/PDIF-Ausgang.

Tabelle 5.6: Testergebnis: HUMAX iCord HD+

Codec	Interop.	Ausgang	Kanalz.	Lauts.	Sync.
LC-AAC	Ja_1	PCM_1	2.0_1	—$_3$	✓
HE-AAC	Ja_1	PCM_1	2.0_1	—$_3$	✓
AC-3	Ja_6	AC-3_1	5.1	✓	✓
E-AC-3	Ja_6	AC-3_2	5.1	✓	✓

TechniSat DIGIT ISIO S

Die TechniSat Set-Top-Box gibt die Ströme der Dolby-Codecs AC-3 und E-AC-3 korrekt im Mehrkanal-Format wieder. Bei AC-3 treten leichte Synchronisationsprobleme auf, die aber am

Gerät manuell wieder ausgeglichen werden können. Bei den AAC-Datenströmen kommt es erneut zu einem Downmix. Alle Codecs werden in der korrekten Lautstäke decodiert.

Tabelle 5.7: Testergebnis: TechniSat DIGIT ISIO S

Codec	Interop.	Ausgang	Kanalz.	Lauts.	Sync.
LC-AAC	Ja_1	PCM_1	2.0_1	✓	✓
HE-AAC	Ja_1	PCM_1	2.0_1	✓	✓
AC-3	Ja_6	$AC\text{-}3_1$	5.1	✓	X_2
E-AC-3	Ja_6	$AC\text{-}3_2$	5.1	✓	✓

Test der Klangqualität

Durch die zahlreichen Tests konnte nach einiger Zeit abgeschätzt werden, welche Datenraten bei den einzelnen Codecs ungefähr eingesetzt werden sollten, um eine annähernd transparente Audioqualität zu erzielen.

Tabelle 5.8: Erforderliche Datenrate für transparente Audioqualität

Codec	Datenrate
LC-AAC	ca. 320kb/s
HE-AAC	ca. 256kb/s
AC-3	ca. 448kb/s
E-AC-3	ca. 320kb/s

Diese Annahmen wurden jedoch unter nicht optimalen Abhör-Bedingungen und anhand der subjektiven Wahrnehmung einer einzigen Person durchgeführt. Sie decken sich jedoch mit vorausgegangenen subjektiven Evaluierungen Mehrkanalton-fähiger Audiocodecs des IRT.

Legende:

Ja_1 - Interoperabel bei allen Containern, Datenraten, Abtastraten, Kanalkonfigurationen, Videoformaten und Audio Only.

Ja_2 - Interoperabel bei allen Containern, Datenraten, 48kHz Abtastrate, Kanalkonfigurationen, Videoformaten und Audio Only.

Ja_3 - Interoperabel nur im MP4-Container, bei allen Datenraten, Abtastraten, Kanalkonfigurationen, Videoformaten und Audio Only.

Ja_4 - Interoperabel nur im MP4-Container, bei allen Datenraten, Abtastraten, Kanalkonfigurationen, Videoformaten und kein Audio Only.

Ja_5 - Bei der zuerst getestete Software V37 wurde der Codec nicht wiedergegeben. Nach Update auf V38 wurde er jedoch unterstützt.

Ja_6 - Interoperabel bei allen Containern, Datenraten, Abtastraten, Kanalkonfigurationen, Videoformaten, kein Audio Only.

Nein - Ton wird in keiner Codierung wiedergegeben.

PCM_1 - PCM-Datenstrom mit gleicher Abtastrate wie Testfile. Analyse durch DM100.

PCM_2 - PCM-Datenstrom. Analyse durch Denon AVR-3312.

$AC\text{-}3_1$ - "Pass Through" des AC-3-Datenstroms in der entsprechenden Datenrate.

$AC\text{-}3_2$ - Transcodierung in AC-3-Datenstrom mit 640kb/s über S/PDIF und E-AC-3, „Pass-Through" über HDMI

$AC\text{-}3_3$ - Transcodierung in AC-3-Datenstrom mit 640kb/s über S/PDIF und HDMI.

2.0_1 - Downmix: C auf L+R, LS auf L, RS auf R. Kein LFE.

2.0_2 - Downmix: C auf L+R, LS und RS auf Front Mitte. Kein LFE.

$—_1$ - Broadband-Signal ca. 6dB Leiser als Broadcast-Signal.

$—_2$ - Broadband-Signal ca. 15dB Leiser als Broadcast-Signal.

$—_3$ - Broadband-Signal über HDMI ca. 30dB Leiser als Broadcast-Signal, über S/PDIF kein Lautstärkeunterschied.

X_1 - Ton ca. 60ms zu spät. Tonversatz ändert sich nicht und kann durch das Gerät manuell ausgeglichen werden.

X_2 - Ton ca. 60ms zu früh. Tonversatz ändert sich nicht und kann durch das Gerät manuell ausgeglichen werden.

X_3 - Die Abtastrate 48kHz wird synchron decodiert. Bei 44.1kHz jedoch Läuft der A/V-Sync auseinander und wird im Laufe der Wiedergabe größer.

5.2 Tests der Entwicklungsgeräte

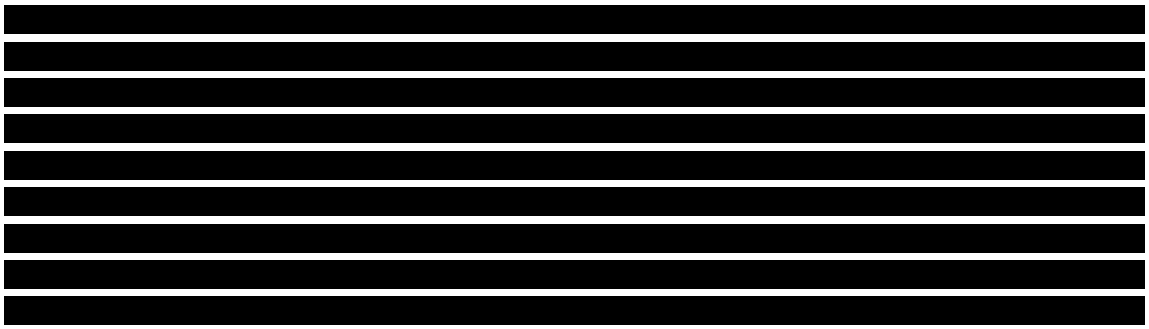
Da es sich bei den Geräten auf dem IRT HbbTV-Interoperabilitätsworkshop um Entwicklungsgeräte handelt, konnten nur begrenzt Rückschlüsse auf die Interoperabilität der späteren Produkte gezogen werden. Dies liegt darin begründet, dass teilweise Komponenten vom Hersteller noch nicht vollständig getestet, oder Software noch nicht vollständig implementiert war. Getestet wurden auf dem Workshop alle Codierungen im MP4- und im MPEG2-TS-Container mit der Sequenz CI-SHORT, in einer Datenrate von jeweils 320kb/s sowie einer Abtastfrequenz von 48kHz. In den Tests der Seriengeräte hat sich ergeben, dass weder die Datenrate, noch die Abtastfrequenz Einfluss auf das Wiedergabeverhalten der Geräte hatte. Des weiteren waren die Testzeiten auf dem Workshop sehr begrenzt, was umfangreichen Untersuchungen verhinderte.

Tabelle 5.9: Testergebnis: Entwicklungsgeräte

✓ - Codec wird mit Mehrkanalton unterstützt.

X - Codec wird nicht wiedergegeben.

2.0 - Stereo Downmix.



5.3 Fazit

Die Tests haben ergeben, dass nur Dolby AC-3-Datenströme von den meisten Geräten korrekt im Mehrkanalton wiedergegeben werden können. Die weit aus effizienteren AAC-Kompressionsverfahren führten leider bei fast allen Geräten nur zu einem Downmix des Mehrkanalton-Signals. E-AC-3-Datenströme wurden von ungefähr der Hälfte der Geräte korrekt in Mehrkanalton genutzt, die andere Hälfte zeigte jedoch massive Interoperabilitätsprobleme.

Somit sind die Codierungen LC-AAC, HE-AAC sowie E-AC-3 noch nicht für eine Übertragung von Mehrkanalton-Inhalte einsetzbar.

Deshalb schlägt diese Arbeit den Dolby AC-3 Codec mit einer Datenrate von 384kb/s für eine Übertragung von Mehrkanalton-Inhalten über das Internet für HbbTV-Geräte vor.

6 HTML5 Versuchsaufbau

Für die PC-Tests sollten die Videos auf einer HTML5-Webseite eingebunden werden, damit sie von mehreren Browser eines Computers im Netzwerk wiedergegeben werden können. Dazu wurde die HTML-Seite sowie die Testsequenzen auf einen Apache-Server gelegt, der die Inhalte über ein internes Netzwerk zur Verfügung stellte. Die Audio- und Video-Inhalte konnten so per progressive Download über das Netzwerk übertragen werden.

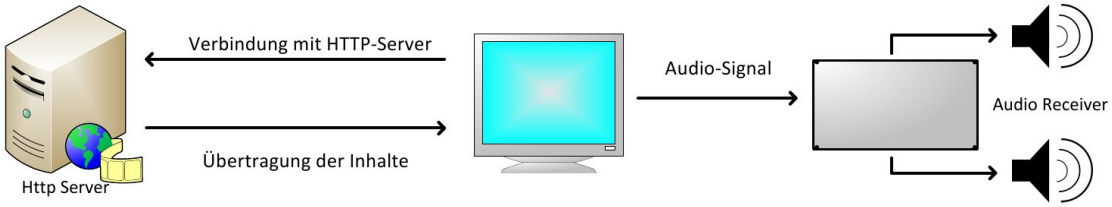


Abbildung 6.1: *HTML5 Versuchsaufbau*

Die Tests sollten einmal unter einem Windows-Betriebssystem, sowie einem Mac-Betriebssystem durchgeführt werden, um eventuelle Unterschiede der Plattformen zu analysieren. Des weiteren sollte nach den Tests und den daraus gewonnen Erkenntnissen eine vereinfachte Referenz-Implemetierung einer HTML5-Seite erstellt werden, die es möglichst vielen Nutzern unterschiedlicher System-Umgebungen ermöglicht, Mehrkanaltoninhalte wiederzugeben.

Audiowiedergabe

Zur Audiowiedergabe wurde der mehrkanalton-fähige Audio-Receiver RX-V530RDS der Firma Yamaha eingesetzt. Im gegensatz zum Denon AVR-3312 besitzt der Yamaha-Receiver sechs analoge Audioeingänge, über die auch ein analoges Mehrkanalton-Signal wiedergegeben werden kann. Neben den analogen Eingängen waren auch digitale optische S/PDIF Audioeingänge verfügbar, so dass jedem Audioausgang der Computer im Netztwerk ein passender Eingang am Receiver gegenüber stand.

Testcomputer

Für die Tests unter dem „Windows 7 Professional“-Betriebssystem wurde ein Computer mit 64Bit-Plattform, acht Gigabyte Arbeitsspeicher und einem Intel i7 Prozessor mit acht Kernen eingesetzt, welcher derzeit dem Stand der Technik für Desktop-PCs entspricht. Es wurden die Audioausgänge, des auf dem Mainboard integrierten „Realtek High Definition Audio“-Chipsatzes der Firma Asus genutzt. Hierbei handelte es sich um sechs analoge Miniklinke-Ausgänge sowie einen optischen S/PDIF-Ausgang.

Des weiteren wurde für die Tests die externe Soundkarte „Gigaport HD+“ der Firma ESI eingesetzt. Die Soundkarte kann über eine USB-2.0-Schnittstelle mit dem Computer verbunden werden und besitzt acht Cinch-Ausgänge für die analoge Audioübertragung.

Abbildung 6.2: *GigaportHD+*

Die Tests unter dem MAC OS/X-Betriebssystem wurden auf einem „Macbook Air“ der Firma Apple durchgeführt. Hier wurde ebenfalls die externe USB-Soundkarte „Gigaport HD+“ verwendet.

6.1 HTML5 Testseite

Für die Browser-Tests sollte eine standardkonforme HTML5-Webseite erstellt werden, auf der die codierten Videos in unterschiedlichen Varianten eingebunden werden können. Die Einbindung erfolgte durch so genannten „Tags“. Das sind Elemente, die einem Browser angeben, dass sich ein Medien-Element auf der Seite befindet. Dazu wurden die, in HTML5 standardisierten „Audio“- und „Video“-Tags eingesetzt, sowie das „Object-Tag“, welches in älteren Webstandards zum Einbinden von Medieninhalten genutzt wurde. Des weiteren sollten zwei FLASH-Plugins zur Wiedergabe in die Seite integriert werden.

Tabelle 6.1: Einbundungsvarianten HTML5-Testseite

HTML-Element	Quelle	Mime Type
<video>	Container	video/Codec
<video>	Container	video/alternativer Codec
<video>	M3U8	application/x-mpegURL
<a>	Container	—
<a>	M3U8	—
<audio>	Audio Only	audio/Codec
<object>	Container	video/Codec
FLASH-Plugin	Container	—

6.1.1 Einbindungsvarianten

Einbindung über das <video>-Tag

Wie schon in Abschnitt 2.2 beschrieben, unterstützt das <video>-Tag verschiedene Attribute. Mit „src=“ wird die Quelle des Inhaltes angegeben, mit „width=“ und „height=“ die angezeigte Größe. „controls“ erzeugt eine Leiste mit Elementen zum navigieren und „preload“ legt fest, dass der Inhalt bereits beim Aufruf der Seite übertragen werden soll.

Des weiteren kann innerhalb des <video>-Tags ein Text angegeben werden der angezeigt wird, falls ein Browser nicht in der Lage sein sollte das Element zu interpretieren. Dies kann bei älteren Browser-Versionen vorkommen.

Zur Untersuchung, ob der Mime Type Einfluss auf das Wiedergabeverhalten hat, wurde das selbe Video ein mal mit der richtigen und ein mal mit einer alternativen Mime Type-Bezeichnung in die Seite eingebunden.

Außerdem wurde für einen Testfall der Quellpfad des Audio- und Video-Inhalts in eine M3U[55]-Playlist als Re-Direktor angegeben. So sollte analysiert werden, ob der Browser, beziehungsweise der genutzte Player in der Lage ist einen Quellpfad aus einer M3U-Datei zu extrahieren.

```
<video  src="http://Video.mp4" width="480" height="320" //
        type="video/mp4" controls preload>
        Der Browser unterstützt das &lt;video&gt;-Tag nicht!
</video>
```

Einbindung über das <audio>-Tag

Das <audio>-Tag wurde auf der Testseite mit den gleichen Attributen wie das <video>-Tag deklariert. Einzig die Attribute für die angezeigte Bildgröße wurden nicht angegeben, da sie für das <audio>-Tag nicht spezifiziert sind.

```
<audio  src="http://Audio.aac" type="audio/aac" //
        controls preload="">
        Der Browser unterstützt das &lt;audio&gt;-Tag nicht!
</audio>
```

Einbindung über das <object>-Tag

Das <object>-Tag wurde vor HTML5 benutzt, um Medieninhalte in eine HTML-Seite einzubetten. Die Medienwiedergabe über das <object>-Tag setzt normalerweise das Vorhandensein einer externen Playerkomponente im Betriebssystem vorraus, an die der Browser die Wiedergabe delegieren kann. Durch die <audio>- und <video>-Tags ist es jedoch in neueren Browsern mittlerweile obsolet geworden.

Mit „param“ können Attribute definiert werden, die das Verhalten des <object>-Tags näher beschreiben.

```
<object data="http://Video.mp4" type="video/mp4" //
        width="480" height="320">
        <param name="autoplay" value="false">
        <param name="scale" value="exactfit">
        Der Brwoser kann das &lt;object&gt; nicht wiedergeben!
</object>
```

Einbindung über Verlinkung

Audio- und Video-Inhalte können auch über eine Verlinkung auf einer HTML-Seite hinterlegt werden. Dazu wird die Quelle des Inhalts in einen Link integriert. Wird dieser Link angeklickt,

[55] MP3-URL

übergiebt der Browser die angegebenen Quellen normalerweise direkt an das Betriebssystem, das seinerseits an Hand des Mime Type dem Nutzer die Wiedergabe in einem vorhandenen Player anbietet. Als Quelle lagen ebenfalls eine Container-Datei und eine M3U-Datei vor.

```
<a href="http://Video.mp4">MP4</a>
<a href="http://Video.m3u">M3U</a>
```

Einbindung über FLASH-Plugin

Die meisten Video-Inhalte werden im Internet immer noch für den PC über ein FLASH-Plugin genutzt. Auch in dieser Arbeit sollten zwei FLASH-Plugins zum Einsatz kommen um das Wiedergabeverhalten von Mehrkanalton-Inhalten eines FLV-Containers zu beobachten. Folgende Plugins wurden in dieser Arbeit eingesetzt:

- **JW Player 6**

  Der JW-Player ist ein Plugin der Firma LongTail Video, welches im Internet weit verbreitet ist, um Videos in der FLASH-Umgebung wiederzugeben. Der Player ist quelloffen in JavaScrip implementiert, was eine Konfiguration und Anpassung des Aussehens und der Wiedergabemöglichkeiten zulässt. Auch in der ARD Mediathek wird der JW-Player von einigen Anstalten verwendet. In dieser Arbeit kam die Version 6 zum Einsatz.

  Um das Plugin in eine HTML-Seite zu integrieren, muss zunächst die FLASH-Komponente sowie die JavaScript-Implementierung des Players von der LongTail Website[56] heruntergeladen und dann in das gleiche Verzeichnis wie die HTML-Seite auf dem Server hinterlegt werden.

  Damit das Plugin verwendet werden kann, muss folgender Quellcode in den <head>-Bereich der Seite aufgenommen werden:

```
<script type="text/javascript" //
        src="jwplayer/jwplayer.js" ></script>
```

  Der Player selbst wird dann ebenfalls in einen <script>-Bereich der Seite eingebunden. Diese Bereiche werden vom Browser nicht interpretiert, sondern rufen ihrerseits Programme oder Funktionen auf, die meist in einer anderen Programmiersprache geschrieben sind. In diesem Falle JavaScript.

```
<div    id="myElement2">Loading the player...</div>
        <script type="text/javascript">
        jwplayer("myElement2").setup({
        file: "http://Video.flv", });
</script>
```

- **Strobe Media Playback**

  Das Strobe Media Playback-Plugin ist ein Teil des Open Source Media Framework (OSMF), welches von der Firma Adobe zur Verfügung gestellt wird. Da das OSMF von Adobe entwickelt wurde, kann das Strobe Media Playback-Plugin als Referenzimplementierung für die Wiedergabe von FLASH-Inhalten betrachtet werden, da FLASH ebenfalls aus dem Hause Adobe stammt.

[56] http://www.longtailvideo.com/jw-player/

Um das Plugin benutzen zu können, muss auch hier die Implementierung des Players heruntergeladen[57] und in das selbe Verzeichnis wie die HTML-Seite gespeichert werden.

Das Video sowie der Player, werden dann über das <object>-Tag mit verschiedenen „param“-Attributen in die Seite aufgenommen.

```
<object width="470" height="320">
<param name="movie" //
value="http://.../StrobeMediaPlayback.swf"></param>
<param name="flashvars" value="src=http.//Vid.flv"></param>
<param name="allowFullScreen" value="true"></param>
<param name="allowscriptaccess" value="always"></param>
<param name="wmode" value="direct"></param>
<embed src="http://.../StrobeMediaPlayback.swf" //
type="application/x-shockwave-flash" //
allowscriptaccess="always" allowfullscreen="true" //
wmode="direct" width="470" height="320" //
flashvars="src=http://Video.flv"></embed>
</object>
```

Die Abbildung 6.3 zeigt die erstellte Testseite.

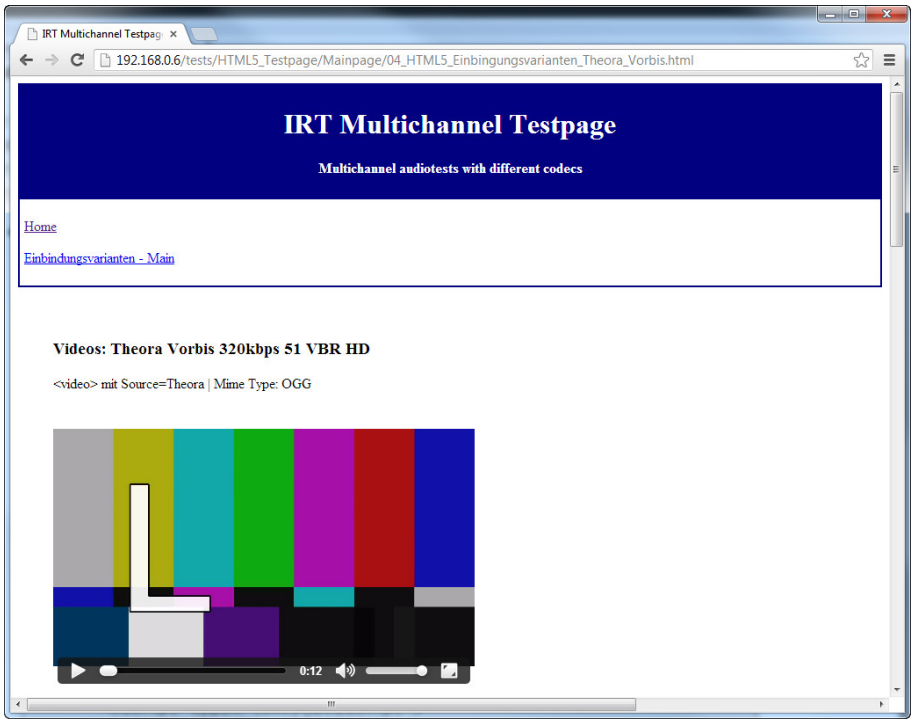


Abbildung 6.3: *HTML5 Testseite*

6.1.2 Browserweiche

Aufgrund der, in Abschnitt 7.1 durchgeführten Tests und den daraus gewonnenen Erkenntnissen sollte eine HTML5-Seite realisiert werden, die es möglichst vielen Browsern ermöglicht, Mehrkanalton-Inhalte wiederzugeben. Dazu war es nötig, eine so genannte Browserweiche zu

[57] http://sourceforge.net/projects/osmf.adobe/files/

implementieren. Diese Weiche erkennt, welcher Browser die Inhalte gerade abruft und stellt den verschiedenen Browsern die passenden Formate in der jeweiligen Einbindung zur Verfügung.

Die Implementierung der Weiche wurde in JavaScript realisiert. Zunächst muss von der HTML-Seite erkannt werden, welcher Browser gerade auf den Inhalt zugreift. Da jeder Browser eine eigene Kennung zurückgibt, die ausgelesen werden kann, ist es möglich, die Browser und das Betriebssystem zu identifizieren. Die Kennung für den Firefox-Browser unter Windows lautet beispielsweise:

Mozilla/5.0 (Windows NT 6.1; WOW64; rv:18.0) Gecko/20100101 Firefox/18.0

Diese Kennung wurde nun mit der Methode „checkBrowserName(name)“ abgefragt und mit den Kennungen der einzelnen Browser verglichen.

```
/* User Agent (Browserkennung) auf einen Browsertyp prüfen */
function checkBrowserName(name)
{
        var agent = navigator.userAgent.toLowerCase();
        if (agent.indexOf(name.toLowerCase())>-1)
        {
                return true;
        }
        return false;
}
```

Wird nun eine Kennung und somit ein Browser identifiziert, können Werte für globale Variablen definiert werden, die das vorgesehene Video für den Browser bereitstellen.

```
// Globale Variablen
var browser = "";
var kompatibel;
var Ausgabe = "";
var VideoType = "";

...

// Browserabfrage
if(checkBrowserName('MSIE'))
{
        browser="Internet Explorer";
        kompatibel=true;
        VideoType = "MP4";
}

...

//Kompatibilität
if(kompatibel==true)
{
  Ausgabe="Der Browser kann Mehrkanalton wiederzugeben!"'
}
else
{
```

```
26    Ausgabe="Der Browser kann kein Mehrkanalton wiedergeben!"'
27 }
28
29 ...
30
31 // Videozuordnung
32 if(VideoType=="MP4")
33 {
34         <source src=\"http://Video.mp4\" type=\"video/mp4\">";
35 }
```

6.2 Auswahl der Browser

Die HTML5-Testseite sollte von den gängigen Browsern abgerufen werden, die derzeit zum Darstellen von Inhalten im Internet vorwiegende eingesetzt werden. Aus diesem Grund wurden folgende Browser ausgewählt.

- **Internet Explorer**

  Der Internet Explorer ist ein Web-Browser der Firma Microsoft und wird mit jedem Windows-Betriebssystem ausgeliefert. Getestet wurden die Versionen 9 und 10.

- **Safari**

  Der von der Firma „Apple“ entwickelte Safari-Browser ist standardmäßig auf allen MAC/-OS-Betriebssystemen vorinstalliert. Zusätzlich stellt Apple auch eine Version für Windows-Betriebssysteme zur Verfügung, welche jedoch stets älter ist, als die für MAC/OS verfügbare Implementierung. Somit wurde unter Windows die Version 5.1.7 getestet und unter MAC/OS die Version 6.0.2.

- **Chrome**

  Chrome ist ein quell-offener Browser, der von der Firma Google entwickelt wird. Der Browser ist seit 2008 verfügbar und seine Verbreitung steigt seit dem kontinuierlich. Unter Windows sowie unter MAC/OS wurde die Version 23.0.1271.97 m getestet.

- **Firefox**

  Der Firefox-Browser wurde von der Mozilla-Stiftung entwickelt. Er ist ein quell-offenes Programm und kann sowohl unter Windows, als auch unter MAC/OS genutzt werden. Unter Windows wurden die Versionen 18 und die sich noch in der Entwicklungsphase befindende Version 20 getestet. Die Entwicklungsvariante des Browsers nennt Mozilla „Firefox Nightly“. Unter MAC/OS wurde ebenfalls die Version 18 analysiert.

- **Opera**

  Der von der Firma „Opera Software ASA“ entwickelte, gleichnamige Browser, ist als Freeware für Windows- und Mac/OS-Betriebssysteme verfügbar. Getestet wurde unter Windows wie unter MAC/OS die Version 12.12.

- **Maxthon**

  Maxthon stellt einen Cloud-Browser dar, der auf Smartphones, Tablets sowie Computern verwendet werden kann. Die Nutzerprofile werden ständig über eine Cloud synchronisiert, damit für jedes Gerät die gleichen Einstellungen gelten. Somit wird gewährleistet, dass der gleiche Inhalt auf verschiedenen Geräten nutzbar ist. Der Browser wird von der gleich-

namigen Firma „Maxthon“ entwickelt, stand zur Zeit der Tests in der Version 4.0.3 zur Verfügung und wurde in dieser unter Windows analysiert.

Es ist schwer eine zuverlässige Angabe darüber zu finden, welche Verbreitung die einzelnen Browser im Internet haben. Laut der Webseite `http://gs.statcounter.com/` benutzen jedoch ca. 98% aller deutschen Internetnutzer eines der getesteten Programme.

7 PC Versuchsdurchführung

In diesem Kapitel werden die gewonnenen Erkenntnisse aus den Browser-Tests beschrieben, die am PC über den digitalen S/PDIF-Ausgang, sowie die analogen Ausgänge am PC und über die „GigaportHD+“-Soundkarte durchgeführt wurden. Die ausführlichen Testergebnisse befinden sich im Anhang (s. XLIV).

7.1 Test der Web-Browser

Internet Explorer

Die Tabelle 7.1 stellt das Mehrkanalton-Abspielverhalten des Internet Explorers dar. Die Legende zur Tabelle befindet sich am Ende dieses Abschnittes.

Der Internet Explorer zeigte in beiden getesteten Versionen (9 und 10) das gleiche Abspielverhalten. Korrekt in Mehrkanalton wurden nur die AAC-Ströme wiedergegeben, wenn diese in einem Container über das <video>-Tag in die HTML-Seite eingebunden waren. Das <audio>-Tag funktionierte nicht und wurde bei der Einbindung von Mehrkanalton- sowie Stereo-Elementarströme nicht angezeigt. Die Einbindung über eine Verlinkung führte zu einem Downmix. Des weiteren waren die eingesetzten FLASH-Plugins nicht interoperabel. Sie erzeugten eine Fehlermeldung und spielten die Testsequenzen nicht ab.

Tabelle 7.1: Testergebnis: Internet Explorer

HTML5	**Interop.**	**Einbindung**	**Kanalz.**	**Lauts.**	**Sync.**
LC-AAC	WIN	<video>$_2$	5.1	✓	✓
HE-AAC	WIN	<video>$_2$	5.1	✓	✓
AC-3	Nein				
E-AC-3	Nein				
Opus	Nein				
Vorbis$_{OGG}$	Nein				
Vorbis$_{WebM}$	Nein				

FLASH	**Interop.**	**Kanalz.**	**Lauts.**	**Sync.**
JW-Player	Nein			
Strobe	Nein			

Safari

Der Safari-Browser war ebenfalls in der Lage, AAC-Datenströme wiederzugeben. Diese konnten über alle eingesetzten Tags eingebunden werden, jedoch nur, wenn die Quelle einen Container

oder Elementardatenstrom darstellte. Während unter Windows die Inhalte jeweils als Downmix wiedergegeben wurden, nutze Safari unter MAC/OS beide Codecs korrekt mit Mehrkanalton. Die eingesetzten FLASH-Player funktionierten nur unter Windows und gaben jeweils nur einen Stereo-Downmix wieder.

Tabelle 7.2: Testergebnis: Safari

HTML5	**Interop.**	**Einbindung**	**Kanalz.**	**Lauts.**	**Sync.**
LC-AAC	Ja	<tag>$_1$	5.1_{MAC}	✓	✓
HE-AAC	Ja	<tag>$_1$	5.1_{MAC}	✓	✓
AC-3	Nein				
E-AC-3	Nein				
Opus	Nein				
$Vorbis_{OGG}$	Nein				
$Vorbis_{WebM}$	Nein				

FLASH	**Interop.**	**Kanalz.**	**Lauts.**	**Sync.**
JW-Player	WIN	2.0_1	✓	✓
Strobe	WIN	2.0_1	✓	✓

Chrome

Der Chrome-Browser konnte AAC, sowie Ogg Vorbis im OGG- und WebM-Container unter beiden Betriebssystemen korrekt wiedergeben. Voraussetzung dafür war, dass die Quelle nicht als M3U vorlag und nicht über das <object>-Tag eingebunden wurde. Die FLASH-Player wurden unter beiden Betriebssystemen unterstützt, gaben jedoch nur einen Downmix in niedrigerer Lautstärke wieder.

Tabelle 7.3: Testergebnis: Chrome

HTML5	**Interop.**	**Einbindung**	**Kanalz.**	**Lauts.**	**Sync.**
LC-AAC	Ja	<tag>$_2$	5.1	✓	✓
HE-AAC	Ja	<tag>$_2$	5.1	✓	✓
AC-3	Nein				
E-AC-3	Nein				
Opus	Nein				
$Vorbis_{OGG}$	Ja	<tag>$_1$	5.1	✓	✓
$Vorbis_{WebM}$	Ja	<tag>$_1$	5.1	✓	✓

FLASH	**Interop.**	**Kanalz.**	**Lauts.**	**Sync.**
JW-Player	Ja	2.0_1	—$_1$	✓
Strobe	Ja	2.0_1	—$_1$	✓

Firefox

Der Firefox-Browser war unter Windows in keiner getesteten Version in der Lage Datenströme im Mehrkanalton wiederzugeben, auch nicht im Downmix. Unter MAC/OS hingegen war eine korrekte Wiedergabe mit dem Ogg Vorbis-Codec möglich, wenn der Datenstrom im Container vorlag und das <object>-Tag nicht genutzt wurde. Die Kanalzuordnung war jedoch L-C-R-LS-RS-LFE und wich somit vom Standard und der Kanalzuordnung der anderen Browser (L-R-C-LFE-LS-RS) ab. Die FLASH-Player arbeiteten nur unter Windows und gaben beide lediglich einen Downmix wieder.

Tabelle 7.4: Testergebnis: Firefox

HTML5	**Interop.**	**Einbindung**	**Kanalz.**	**Lauts.**	**Sync.**
LC-AAC	Nein				
HE-AAC	Nein				
AC-3	Nein				
E-AC-3	Nein				
Opus	Nein				
$\text{Vorbis}_{\text{OGG}}$	MAC	$<\text{tag}>_3$	5.1	✓	✓
$\text{Vorbis}_{\text{WebM}}$	MAC	$<\text{tag}>_3$	5.1	✓	✓

FLASH	**Interop.**	**Kanalz.**	**Lauts.**	**Sync.**
JW-Player	WIN	2.0_1	$—_1$	✓
Strobe	WIN	2.0_1	✓	✓

Opera

Auch der Opera-Browser konnte unter Windows keine Mehrkanalton-Inhalte wiedergeben. Im Gegensatz zum Firefox-Browser wird der Vorbis Codec jedoch abgespielt, wenn auch nur im Downmix. Unter MAC/OS war eine korrekte Mehrkanalton-Wiedergabe mit Vorbis möglich, die Einbindung muss jedoch über das <video>-Tag oder über eine Verlinkung erfolgen. Stellte die Quelle eine M3U-Playlist dar, so konnte der Inhalt ebenfalls nicht abgespielt werden. Die FLASH-Player wurden auch hier nur unter Windows unterstützt und erzeugten einen Downmix.

Tabelle 7.5: Testergebnis: Opera

HTML5	Interop.	Einbindung	Kanalz.	Lauts.	Sync.
LC-AAC	Nein				
HE-AAC	Nein				
AC-3	Nein				
E-AC-3	Nein				
Opus	Nein				
$\text{Vorbis}_{\text{OGG}}$	Ja	$<\text{tag}>_4$	5.1_{MAC}	✓	✓
$\text{Vorbis}_{\text{WebM}}$	Ja	$<\text{tag}>_4$	5.1_{MAC}	✓	✓

FLASH	Interop.		Kanalz.	Lauts.	Sync.
JW-Player	WIN		2.0_1	✓	✓
Strobe	WIN		2.0_1	✓	✓

Maxthon

Der Maxthon-Browser konnte nur unter Windows getestet werden. Dabei wurden die AAC-Ströme sowie Ogg Vorbis korrekt im Mehrkanalton-Format wiedergegeben. Bei AAC durfte die Einbindung jedoch nicht über das <object>-Tag erfolgen. Des weiteren mussten beide Formate als Quelle in einer Container-Datei vorliegen. Die FLASH-Player wurden beide unterstützt, gaben den Mehrkanalton-Inhalt jedoch ebenfalls nur als Downmix wieder.

Tabelle 7.6: Testergebnis: Maxthon

HTML5	Interop.	Einbindung	Kanalz.	Lauts.	Sync.
LC-AAC	WIN	$<\text{tag}>_5$	5.1	✓	✓
HE-AAC	WIN	$<\text{tag}>_5$	5.1	✓	✓
AC-3	Nein				
E-AC-3	Nein				
Opus	Nein				
$\text{Vorbis}_{\text{OGG}}$	WIN	$<\text{tag}>_6$	5.1	✓	✓
$\text{Vorbis}_{\text{WebM}}$	WIN	$<\text{tag}>_6$	5.1	✓	✓

FLASH	Interop.		Kanalz.	Lauts.	Sync.
JW-Player	WIN		2.0_1	5.1	✓
Strobe	WIN		2.0_1	5.1	✓

Legende:

WIN - Interoperabel nur unter Windows.

MAC - Interoperabel nur unter MAC/OS.

Ja - Interoperabel unter Windows und MAC/OS.

Nein - Nicht Interoperabel unter Windows und MAC/OS.

$<\text{video}>_2$ - Interoperabel nur mit Container als Quelle und nur im <video>-Tag

$<\text{tag}>_1$ - Interoperabel nur mit Container als Quelle jeweils im <video>-, <audio>- und <object>-Tag.

$<\text{tag}>_2$ - Interoperabel nur mit Container als Quelle jeweils im <video>-, <audio>- und unter MAC/OS auch im <object>-Tag.

<tag>$_3$ - Interoperabel nur mit Container als Quelle und nur im <video>- und <audio>–Tag

<tag>$_4$ - Interoperabel nur mit Container als Quelle und nur im <video>-Tag sowie über eine Verlinkung.

<tag>$_5$ - Interoperabel nur mit Container als Quelle und nur im <video>- und <audio>-Tag sowie über eine Verlinkung.

<tag>$_6$ - Interoperabel nur mit Container als Quelle in allen Tags.

2.0_1 - Stereo Downmix. C auf L+R, LS auf L, RS auf R, kein LFE.

—$_1$ - Wiedergabelautstärke ca. 6dB leiser als Referenz.

7.2 Browserweiche

Anhand der in Abschnitt 7.1 durchgeführten Tests wurde festgestellt, dass unter einem Windows-Betriebssystem nicht alle getesteten Browser in der Lage sind, Mehrkanalton-Inhalte wiederzugeben. Dadurch, dass Firefox und Opera derzeit grundsätzlich nicht Mehrkanalton-fähig sind, kann ihnen auch eine Browserweiche keine Audio-Inhalte zuweisen, die korrekt wiedergegeben werden. Deshalb musste die Weiche so implementiert werden, dass die Benutzer eines Firefox- oder Opera-Browsers gebeten werden, zu einen anderen Mehrkanalton-fähigen Browser wie Chrome oder dem Internet Explorer zu wechseln, sofern sie nicht nur einen Downmix erhalten möchten.

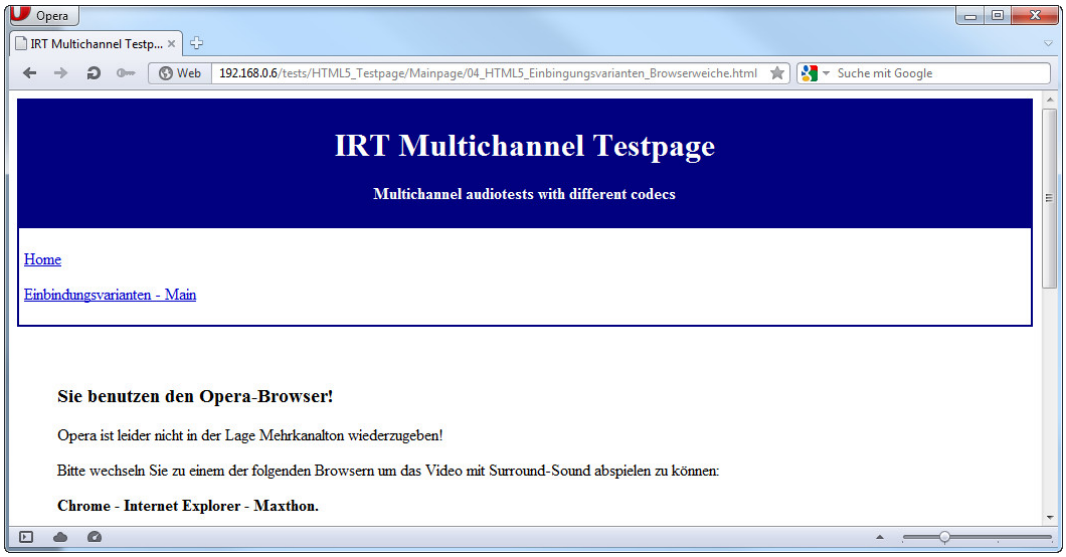


Abbildung 7.1: *Browserweiche Opera*

Da unter MAC/OS alle getesteten Browser in der Lage waren, Mehrkanalton-Inhalte wiederzugeben, konnte die Weiche so implementiert werden, dass jeder Browser den für ihn passenden Audio- und Video-Inhalt erhält.

8 Ergebnis und Ausblick

Die Verbreitung von Medieninhalten über das Internet weitet sich ständig aus. Über Portale wie Youtube oder die Mediatheken der Rundfunkanstalten steht uns jederzeit eine Vielfalt an Audio- und Video-Inhalten zur Verfügung, die wir über das Internet abrufen können. Die Verbreitung von Mehrkanalton-Inhalten im Internet spielt zur Zeit aber noch eine untergeordnete Rolle. Aus diesem Grund sollte in dieser Arbeit durch ausführliche Tests das Wiedergabeverhalten verschiedener Audiocodecs auf mehrkanalton-fähigen HbbTV-Geräten und PCs mit Web-Browsern untersucht werden. Ziel war eine Erarbeitung von Audiocodierprofilen, welche von allen oder möglichst vielen Geräten und Browsern unterstützt werden, um eine Mehrkanalton-Übertragung über das Internet zu ermöglichen.

Test und Analyse von Möglichkeiten zur Online Distribution von Mehrkanalton für HbbTV

Im ersten Teil dieser Arbeit wurden ausführliche Tests mit bereits am Markt erhältlichen Geräten, sowie mit Geräten die sich noch in der Entwicklungsphase befinden, durchgeführt. Die Analyse der Tests ergab, dass 83% der bereits am Markt erhältlichen Geräte in der Lage sind, den Inhalt mit Datenströmen im Format

Dolby AC-3

korrekt mit Mehrkanalton wiederzugeben. Dieser Codec wies damit die mit Abstand größte Interoperabilität über alle getesteten Geräte auf.

[redacted]

Die AAC-Kompressionsverfahren stellten sich als nicht einsetzbar heraus, da sie von der Mehrzahl der Geräte nur als Downmix wiedergegeben wurden. E-AC-3 wurde von einigen Geräten zwar korrekt im Mehrkanalton wiedergegeben, führte jedoch oft zu erheblichen Interoperabilitätsproblemen. Es ist jedoch zu erwarten, dass E-AC-3 in Zukunft von immer mehr Geräten unterstütz wird.

Somit sind die effizientesten Kompressionsverfahren für eine Mehrkanalton-Übertragung leider nicht einsetzbar.

Aus diesen Gründen empfiehlt diese Arbeit für die Mehrkanalton-Übertragung mit HbbTV den Dolby AC-3-Codec mit einer Datenrate von 384kb/s einzusetzen. Diese Datenrate ermöglicht eine annähernd transparente Audioqualität und entspricht zudem der Datenrate, die auch für Mehrkanalton-Inhalte im linearen Fernsehsignal verwendet wird. Da auch über Broadcast der Dolby AC-3-Codec zum Einsatz kommt, kann mit den gleichen Codierparametern über HbbTV die selbe Audioqualität wie über das lineare Fernsehsignal erzielt werden.

Die Verwendung des AC-3-Codecs bringt zusätzlich den Vorteil, dass die Medienanbieter keine neuen Codier-Werkzeuge und Lizenzen einsetzten müssen, um Mehrkanalton-Inhalte zu verbreiten. Dem gegenüber steht die hohe Datenrate, die aufgewendet werden muss, um eine transpa-

rente Audioqualität zu gewährleisten. Jedoch stellt es heute einerseits technisch keine Probleme dar, Mehrkanalton-Inhalte über Broadband anzubieten und andererseits ist der Realisierungsaufwand sehr gering, da die meisten Mehrkanalton-TV-Inhalte schon im Dolby AC-3-Codec vorliegen.

Für die Zukunft wird es auch interessant zu beobachten sein, wie sich das Abspielverhalten der Codecs bei einer segmentierten Übertragung für Adaptive Streaming verhält. Dies soll in anschließenden Tests an diese Arbeit analysiert werden.

Test und Analyse von Möglichkeiten zur Online Distribution von Mehrkanalton für PC

Im zweiten Teil dieser Arbeit wurden verschiedene mehrkanalton-fähige Audiocodecs durch unterschiedliche Einbindungsvarianten, sowie durch zwei FLASH-Plugins auf einer HTML5-Seite integriert. Diese Seite wurde von allen gängigen Browsern aufgerufen um ihr Wiedergabeverhalten der unterschiedlichen Codecs und Einbindungen zu analysieren. Leider stellte sich durch die Tests heraus, dass kein eindeutiges Audiocodierprofil identifiziert werden kann, welches von allen Browsern korrekt wiedergegeben wird. Alleine unter verschiedenen Betriebssystemen ist das Wiedergabeverhalten selbst von den selben Browsern unterschiedlich. Dies hängt auch damit zusammen, dass die Browser nicht auf die Codecs der Betriebssysteme zurückgreifen. Das bedeutet, das eine Codierung die beispielsweise unter Windows wiedergegeben wird, nicht auch vom Browser unterstütz werden muss.

Aus den Testergebnissen ergibt sich, dass für ein Windows-Betriebssystem der Mehrkanalton-Inhalt mit den Codecs

LC-AAC / HE-AAC

codiert und über das <video>- oder <audio>-Tag in HTML5 eingebunden werden sollte. Diese Codierung und Einbindung wird von den Browsern Chrome, Internet Explorer und Maxthon korrekt wiedergegeben. Die Benutzer des Opera-, Firefox- und Safari-Browsers müssen gebeten werden, einen anderen Mehrkanalton-fähigen Browser zu nutzen.

Unter einem MAC/OS-Betriebssystem ist es möglich, jedem Browser einen entsprechenden Datenstrom zur Verfügung zu stellen, der korrekt wiedergegeben werden kann. Für Chrome und Safari sind das die

LC-AAC / HE-AAC

Codecs und für Firefox und Opera der

Ogg Vorbis

Codec. Die Einbindung sollte jeweils über das <video>- oder <audio>-Tag erfolgen und die Datenrate 320kb/s für eine annähernd transparente Audioqualität nicht unterschreiten.

Mit den FLASH-Plugins war es leider in keiner Konstellation möglich Mehrkanalton-Inhalte korrekt wiederzugeben. Dies ist in so fern bedauerlich, als dass die meisten Videos mit Hilfe der FLASH-Umgebung im Internet abgespielt werden.

Diese Codec- und Einbindungs-Empfehlungen können jedoch nur als aktueller Zwischenstand betrachtet werden. Denn die Interoperabilität zwischen den einzelnen Browsern und Codecs ist ständig im Wandel. So kündigte beispielsweise der Firefox-Browser für das Frühjahr 2013 ebenfalls eine Unterstützung der AAC-Codecs an. Es bleibt zu hoffen, dass sich alle Browser-Hersteller in Zukunft für einen einzigen Codec entscheiden und für alle Betriebssysteme eine einheitliche Wiedergabemöglichkeit implementieren. Bis es so weit ist, sollten diese Test in regelmäßigen Zeitabständen wiederholt werden, da die Testergebnisse schon durch das nächste Update eines Browsers nicht mehr aktuell sein könnten.

Diese Arbeit konnte hoffentlich dazu beitragen, die Übertragung von Mehrkanalton-Inhalten über das Internet zu ermöglichen oder weiter zu vereinfachen. Denn besonders in den Mediatheken wäre ein Medienerlebnis mit Mehrkanalton sehr wünschenswert und durch die heutigen technischen Möglichkeiten eigentlich fast schon eine Pflicht.

Literaturverzeichnis

[Aikin(2012)] Aikin, Jim: „CSOUND POWER! - The Comprehensive Guide“, Course Technology, 2012

[ARD/ZDF(2012)] ARD/ZDF-Onlinestudie 2012: *Internetnutzung in Deutschland steigt weiter - Tablets und Smartphones sorgen für stärkere Nachfrage nach TV-Inhalten* , Mainz/Frankfurt, Pressemitteilung: 20. August 2012.

[BITKOM(2012)] Bundesverband Informationswirtschaft Telekommunikation und neue Medien e.V.: *Fernseher mit Internet-Anschluss werden Standard* , Pressemitteilung: 20. Mai 2012

[Chiariglione(2012)] Chiariglione, Leonardo: „The MPEG Representation of Digital Media“, Vol. 1, Springer Science+Business Media, 2012

[c't(12/2010)] Dr. Zota, V.: *Mundgerecht zubereitet - Adaptives HTTP-Streaming verspricht Qualitätssprung bei Internet-Streaming*, *c't* Vol. 12, 2010

[Dickreiter(2008)] Dickreiter, M., Dittel, V., Hoeg, W., & Wöhr, M.: „Handbuch der Tonstudiotechnik Band 2“, Vol. 7, K. G. Sauer, München, 2008

[ETSI TS 102 796(2012)] ETSI TS 102 796, v1.2.1: „Technical Specification: Hybrid Broadcast Broadband TV“, Nov. 2012

[ISO-IEC 11172-3(1993)] ISO/IEC 11172-3: „Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s - Part 3: Audio“, 1993

[ISO-IEC 13818-7(2006)] ISO-IEC 14496-14: „Information technology - Generic coding of moving pictures and associated audio information - Part 7: Advanced Audio Coding (AAC)“, 2006

[ISO-IEC 14496-14(2003)] ISO/IEC 13818-7: „Information technology - Coding of audio-visual objects - Part 14: MP4 file format“, 2003

[IRT(2009)] Institut für Rundfunktechnik GmbH: *Neue europäische Initiative verbindet Fernsehen mit der Dynamik des Internets* , München, Pressemitteilung: 27. August 2009.

[ITU-R BS.775-2(2006)] ITU-R BS.775-2: „Multichannel stereophonic sound system with and without accompanying picture“, 2006

[ix(7/2012)] Heck, J., Lo Iacano, L., Ta, S. T.: *Der Umgang mit Video und Audio in HTML5*, *ix* Vol. 7, 2012

[IETF RFC: 6716(2012)] Internet Engineering Task Force: „Definition of the Opus Audio Codec“, Sep. 2012

[Theora Specification(2011)] Xiph.Org Foundation: „Theora Specification“, Mar. 2011

[Schmidt(2009)] Schmidt, Ulrich: „Professionelle Videotechnik“, Vol. 5, Springer, 2009

[RFC 6386(2011)] Request for Comments: 6386: „VP8 Data Format and Decoding Guide“, Nov. 2011

[Schmalohr(2012)] Schmalohr, M., Vogl, A.: „ARD Webtechnik Handbuch - Empfehlungen zur Umsetzung des AV-Standards für die Internetdistribution von Audio und Video in der ARD“, Institut für Rundfunktechnik GmbH, 2012, Verfügbar unter http://webdb.irt.de/richtlinien

[Vorbis I specification(2012)] Xiph.Org Foundation: „Vorbis I specification“, Feb. 2012

[Zölzer(2005)] Zölzer, Udo: „Digitale Audiosignalverarbeitung“, Vol. 3, Vieweg+Teubner, 2005

[Fastel, Zwicker(2006)] Fastel, Hugo, Zwicker, Eberhard: „Psychoacoustics“. Vol. 3, Springer-Verlag Berlin-Heidelberg, 2006

Anhang

Komposition der Testsequenzen am Beispiel CI-SHORT

```
<CsoundSynthesizer>
<CsOptions>
</CsOptions>
<CsInstruments>
sr              = 48000
ksmps   = 1
nchnls  = 6
0dbfs   = 1
giAmp           = 1
giAmpBeep       = 0.4
giAmpSinus      = 0.3
giSine ftgen 0, 0, 8192, 10, 1
giAtt   = 0.01
giDec   = 0.01
instr 1
aL, aR, aC, aLS, aRs, aLFE = 0
aL diskin "CI_MC_SHORT_L.wav", 1
outc aL, aR, aC, aLFE, aLS, aRS
endin
instr 10
aL, aR, aC, aLS, aRs, aLFE = 0
aOscL oscil giAmpSinus, cpspch(9.00), giSine
kEnv linen 1, giAtt, p3, giDec
aL = aOscL*kEnv
outc aL, aR, aC, aLFE, aLS, aRS
endin
instr 2
aL, aR, aC, aLS, aRs, aLFE = 0
aR diskin "CI_MC_SHORT_R.wav", 1
outc aL, aR, aC, aLFE, aLS, aRS
endin
instr 20
aL, aR, aC, aLS, aRs, aLFE = 0
aOscR oscil giAmpSinus, cpspch(9.04), giSine
kEnv linen 1, giAtt, p3, giDec
aR = aOscR*kEnv
outc aL, aR, aC, aLFE, aLS, aRS
endin
instr 3
aL, aR, aC, aLS, aRs, aLFE = 0
aC diskin "CI_MC_SHORT_C.wav", 1
outc aL, aR, aC, aLFE, aLS, aRS
```

```
endin
instr 30
aL, aR, aC, aLS, aRs, aLFE = 0
aOscC oscil giAmpSinus, cpspch(9.07), giSine
kEnv linen 1, giAtt, p3, giDec
aC = aOscC*kEnv
outc aL, aR, aC, aLFE, aLS, aRS
endin


instr 4
aL, aR, aC, aLS, aRs, aLFE = 0
aLS diskin "CI_MC_SHORT_LS.wav", 1
outc aL, aR, aC, aLFE, aLS, aRS
endin

instr 40
aL, aR, aC, aLS, aRs, aLFE = 0
aOscLS oscil giAmpSinus, cpspch(9.10), giSine
kEnv linen 1, giAtt, p3, giDec
aLS = aOscLS*kEnv
outc aL, aR, aC, aLFE, aLS, aRS
endin

instr 5
aL, aR, aC, aLS, aRs, aLFE = 0
aRS diskin "CI_MC_SHORT_RS.wav", 1
outc aL, aR, aC, aLFE, aLS, aRS
endin

instr 50
aL, aR, aC, aLS, aRs, aLFE = 0
aOscRS oscil giAmpSinus, cpspch(9.10), giSine
kEnv linen 1, giAtt, p3, giDec
aRS = aOscRS*kEnv
outc aL, aR, aC, aLFE, aLS, aRS
endin

instr 60
aL, aR, aC, aLS, aRs, aLFE = 0
aOscLFE oscil giAmpSinus, cpspch(6.00), giSine
kEnv linen 1, giAtt, p3, giDec
aLFE = aOscLFE*kEnv
outc aL, aR, aC, aLFE, aLS, aRS
endin

instr 100
aOscL oscil giAmpBeep, cpspch(9.00), giSine
aOscR oscil giAmpBeep, cpspch(9.04), giSine
aOscC oscil giAmpBeep, cpspch(9.07), giSine
aOscLFE oscil giAmpBeep, cpspch(6.00), giSine
```

```
aOscLS oscil giAmpBeep, cpspch(9.10), giSine
aOscRS oscil giAmpBeep, cpspch(09.10), giSine
aOscLFE oscil giAmpBeep, 100, giSine
kEnv linen 1, giAtt, p3, giDec
aL              = aOscL*kEnv
aR              = aOscR*kEnv
aC              = aOscC*kEnv
aLS     = aOscLS*kEnv
aRS     = aOscRS*kEnv
aLFE    = aOscLFE*kEnv

outc aL, aR, aC, aLFE, aLS, aRS
endin

</CsInstruments>
<CsScore>

i1              0               2
i100    2               0.04

i2              2.04    2
i100    4.04    0.04

i3              4.08    2
i100    6.08    0.04

i4              6.12    2
i100    8.12    0.04

i5              8.16    2
i100    10.16   0.04

i60             10.20   2
i100    10.20   0.04
e

</CsScore>
</CsoundSynthesizer>
```

```
#_________________PARAMTER_________________________
xsolution = 720
ysolution = 576
#FARBRAUM FÜR TESTELEMENTE
colorMatrix = "Rec601"
#_________________AVI_QUELLEN_____________________
#AVIQUELLEN IMORTIEREN
Vid01 = AviSource("CI-SHORT_HD.avi").//
BicubicResize(xsolution, ysolution)
# Wenn AviSource nicht geht -> DirectShowSource()
#__________AUDIO___________________
#INTERLACING
```

```
Vid01 = Vid01.AssumeFrameBased()
Vid01 = Vid01.AssumeBFF() # HALBBILDREIHENFOLGE FESTLEGEN
VidFinal = Vid01.SeparateFields().SelectEvery(4,0,3).Weave()
#_____________________VIDEO_AUSGEBEN_______________
return VidFinal
```

Codierung der Testsequenzen am Beispiel CI-SHORT

AAC

```
@ echo off
cd \
cls
REM ********************************************************
REM BITTE SETZEN!
set BitrateName=320
set Format=51
REM Mehrkanal -> 51  |  Stereo -> 20
set FrequenzName=48
set BitrateSpec=cbr
set BitrateSpecName=CBR
REM CBR -> cbr  |  VBR -> br
set VideoFormat=HD
set FrameRate=50
set TestName=CI_SHORT
set MainPath=E:\
REM ********************************************************
set InputPath=%MainPath%01_QuellAudio\WAV\
set OutputPathLC=%MainPath%01_QuellAudio\AAC\LC\
set OutputPathHE=%MainPath%01_QuellAudio\AAC\HE\
set LCName=LC
set HEName=HE
set CodecLCName=AAC-%LCName%
set CodecHEName=AAC-%HEName%
set CodecHE=he
set CodecLC=lc
set Bitrate=%BitrateName%000
set Extension=.aac
set InputName=%TestName%_%Format%_%FrequenzName%.wav
set OutputNameLC=%TestName%_%CodecLCName%_%BitrateName%_//
%Format%_%FrequenzName%_%BitrateSpecName%_%VideoFormat%
set OutputNameHE=%TestName%_%CodecHEName%_%BitrateName%_//
%Format%_%FrequenzName%_%BitrateSpecName%_%VideoFormat%
set neroexe=E:\01_Tools\NeroAAC\win32\neroaacenc.exe
echo *
echo ****************************************
echo Um %OutputNameLC% und %OutputNameHE% mit //
NEROAACENC zu encodieren
pause
%neroexe% -%BitrateSpec% %Bitrate% -%CodecLC% -if //
%InputPath%%InputName% -of %OutputPathLC%%OutputNameLC%
```

```
%neroexe% -%BitrateSpec% %Bitrate% -%CodecHE% -if //
%InputPath%%InputName% -of %OutputPathHE%%OutputNameHE%
REM VideoMultiplexing
set InputPathAudioLC=%MainPath%01_QuellAudio\AAC\LC\
set InputPathAudioHE=%MainPath%01_QuellAudio\AAC\HE\
set InputPathVideo=%MainPath%02_QuellVideo\H264\
set OutputPathLC=%MainPath%03_Mp4s\AAC\LC\
set OutputPathHE=%MainPath%03_Mp4s\AAC\HE\
set Extension=.mp4
set InputNameVideo=%TestName%_%VideoFormat%.h264
set InputNameAudioLC=%OutputNameLC%
set InputNameAudioHE=%OutputNameHE%
set OutputNameLC=%TestName%_%CodecLCName%_//
%BitrateName%_%Format%_//
%FrequenzName%_%BitrateSpecName%_%VideoFormat%%Extension%
set OutputNameHE=%TestName%_%CodecHEName%_//
%BitrateName%_%Format%_//
%FrequenzName%_%BitrateSpecName%_%VideoFormat%%Extension%
set Mp4Boxexe=E:\01_Tools\GPAC\mp4box.exe
echo *
echo **********************************************
echo Um %OutputNameLC% und %OutputNameHE% mit //
 Mp4Box zu multiplexen
pause
%Mp4Boxexe% -add %InputPathVideo%%InputNameVideo% -add //
%InputPathAudioLC%%InputNameAudioLC% -fps %FrameRate%//
 %OutputPathLC%%OutputNameLC%
%Mp4Boxexe% -add %InputPathVideo%%InputNameVideo% -add //
%InputPathAudioHE%%InputNameAudioHE% -fps %FrameRate%//
 %OutputPathHE%%OutputNameHE%
echo *
echo **************
echo JUHU!! FERTIG!
pause
```

AC-3

```
REM **********************************************
set InputPath=%MainPath%01_QuellAudio\WAV\
set OutputPath=%MainPath%01_QuellAudio\Dolby\AC3\
set Block=1
set Codec=Dolby-AC3
set SMixLevel=2
set BitrateSpec=cbr
set BitrateSpecName=CBR
set Extension=.ac3
set InputName=%TestName%_%Format%_%FrequenzName%.wav
set OutputName=%TestName%_%Codec%_%BitrateName%_%Format%_//
%FrequenzName%_%BitrateSpecName%_%VideoFormat%%Extension%
set aftenexe=E:\01_Tools\Aften\aften_x86\aften.exe
echo *
echo ******************************************************
```

```
16 echo Um %OutputName% mit AFTEN zu encodieren
17 pause
18 %aftenexe% -b %BitrateName% -s %Block% -pad 0 -smix %SMixLevel%
19 %InputPath%%InputName% %OutputPath%%OutputName%
20 REM VIDEO MULTIPLEXING MIT DMG
21 set DOLBY-MUX="E:\01_Tools\DolbyMedieGenerator\Dolby//
22  Media Generator\dmg-umux.exe"
23 set InputNameVideo=%TestName%_%VideoFormat%.h264
24 set InputNameAudio=%outputName%
25 set InputPathAudio=%MainPath%01_QuellAudio\Dolby\AC3\
26 set InputPathVideo=%MainPath%02_QuellVideo\H264\
27 set OutputPathVideo=%MainPath%03_Mp4s\Dolby\AC3\
28 set Extension=.mp4
29 set OutputName=%TestName%_%Codec%_%BitrateName%_%Format%_//
30 %FrequenzName%_%BitrateSpecName%_%VideoFormat%%Extension%
31 echo *
32 echo **********************
33 echo Um %InputNameAudio% und %InputNameVideo% mit //
34 FFmpeg zu %outputName% zu multiplexen
35 pause
36 %DOLBY-MUX% -i %InputPathVideo%%InputNameVideo% //
37 -i %InputPathAudio%//
38 %InputNameAudio% -o %OutputPathVideo%%OutputName% //
39 --overwrite --progress
40 echo *
41 echo ***************
42 echo JUHU!! FERTIG!!
43 pause
```

E-AC-3

```
1  @ echo off
2  cd \
3  cls
4  REM DD+ AUDIO ENCODING MIT DOLBY MEDIA GENERATOR
5  REM ************************************
6  REM BITTE SETZEN!
7  set BitrateName=320
8  set Format=51
9  REM Mehrkanal -> 51  |  Stereo -> 20
10 set FrequenzName=48
11 set Frequenz=48000
12 set VideoFormat=HD
13 set TestName=CI_SHORT
14 set MainPath=E:\03_Encoding\2012_11_16_HbbTV-Interop\
15 REM ****************************
16 set InputPath=%MainPath%01_QuellAudio\WAV\
17 set OutputPath=%MainPath%01_QuellAudio\Dolby\DD+\
18 set Block=1
19 set Codec=Dolby-DD+
20 set BitrateSpecName=CBR
21 set Extension=.ec3
```

```
set InputName=%TestName%_%Format%_%FrequenzName%.wav
set OutputName=%TestName%_%Codec%_%BitrateName%_%Format%_//
%FrequenzName%_%BitrateSpecName%_%VideoFormat%%Extension%
set DOLBY="E:\01_Tools\DolbyMedieGenerator\Dolby //
Media Generator\dmg.exe"
echo *
echo ********************************************************
echo Um %OutputName% mit DMG zu encodieren
pause
%DOLBY% -i %InputPath%%InputName% -o %OutputPath%%OutputName% //
 --audio-cbr-rate %BitrateName% --audio-samplerate %Frequenz% //
 --input-dialnorm -31.0 --overwrite --progress
REM VIDEO MULTIPLEXING MIT DMG
set DOLBY-MUX="E:\01_Tools\DolbyMedieGenerator\Dolby //
 Media Generator\dmg-umux.exe"
set InputNameVideo=%TestName%_%VideoFormat%.h264
set InputNameAudio=%outputName%
set InputPathAudio=%MainPath%01_QuellAudio\Dolby\DD+\
set InputPathVideo=%MainPath%02_QuellVideo\H264\
set OutputPathVideo=%MainPath%03_Mp4s\Dolby\DD+\
set Extension=.mp4
set OutputName=%TestName%_%Codec%_%BitrateName%_%Format%_//
%FrequenzName%_%BitrateSpecName%_%VideoFormat%%Extension%
echo *
echo ****************************************
echo Um %InputNameAudio% und %InputNameVideo% mit DMG zu //
 %outputName% zu multiplexen
pause
%DOLBY-MUX% -i %InputPathVideo%%InputNameVideo% -i //
%InputPathAudio% %InputNameAudio% -o %OutputPathVideo%//
%OutputName% --overwrite --progress
echo *
echo ***************
echo JUHU!! FERTIG!!
pause
```

Ogg Vorbis

```
REM @ echo off
cd \
cls
REM OGG AUDIO ENCODING MIT FFMPEG
REM ****************************************
REM BITTE SETZEN!
set BitrateName=320
set Format=51
REM Mehrkanal -> 51   |  Stereo -> 20
set FrequenzName=48
set Frequenz=48000
set VideoFormat=HD
set TestName=CI_SHORT
set MainPath=E:\03_Encoding\2012_11_16_HbbTV-Interop\
```

```
REM ****************************************
set AudioInputPath=%MainPath%01_QuellAudio\WAV\
set AudioOutputPath=%MainPath%01_QuellAudio\OggVorbis\
set VideoInputPath=%MainPath%02_QuellVideo\AVI\
set TheoraOutputPath=%MainPath%03_Mp4s\Theora\
set Block=1
set Codec=OggVorbis
set BitrateSpecName=VBR
set Extension=.ogg
set AudioInputName=%TestName%_%Format%_%FrequenzName%
set VideoInputName=%TestName%_HD.avi
set AudioOutputName=%TestName%_%Codec%_%BitrateName%_//
%Format%_%FrequenzName%_%BitrateSpecName%_%VideoFormat%
set FFMPEG="E:\01_Tools\FFmpegNeu\bin\ffmpeg.exe"
set OggEnc2="E:\OggEnc\oggenc2.87-1.3.3-x64\oggenc2.exe"
echo *
echo ********************************************************
echo Um %AudioOutputName% mit OggEnc2 zu encodieren
pause
%OggEnc2% --max-bitrate %BitrateName% %AudioInputPath%//
%AudioInputName%.wav
REM ****************************************
set OutputName=%AudioOutputName%
echo *
echo ************************************
echo Um %AudioOutputName% mit FFmpeg zu multiplexen
pause
%FFMPEG% -i %AudioInputPath%%AudioInputName%.ogg //
-i %VideoInputPath%%VideoInputName% -acodec copy //
-vcodec theora -vb 5000k %TheoraOutputPath%%OutputName%.ogv
echo *
echo ***************
echo JUHU!! FERTIG!!
pause
```

Opus

```
REM @ echo off
cd \
cls
REM OPUS AUDIO ENCODING MIT OPUESENC
REM ************************************************
REM BITTE SETZEN!
set BitrateName=320
set Format=51
REM Mehrkanal -> 51   |  Stereo -> 20
set FrequenzName=48
set Frequenz=48000
set VideoFormat=HD
set TestName=CI_SHORT
set MainPath=E:\03_Encoding\2012_11_16_HbbTV-Interop\
REM ************************************************
```

```
set AudioInputPath=%MainPath%01_QuellAudio\WAV\
set AudioOutputPath=%MainPath%01_QuellAudio\Opus\
set VideoInputPath=%MainPath%02_QuellVideo\AVI\
set TheoraOutputPath=%MainPath%03_Mp4s\Theora\Opus\
set Codec=Opus
set BitrateSpecName=CBR
set Extension=.opus
set AudioInputName=%TestName%_%Format%_%FrequenzName%
set VideoInputName=%TestName%_HD.avi
set AudioOutputName=%TestName%_%Codec%_%BitrateName%_//
%Format%_%FrequenzName%_%BitrateSpecName%_%VideoFormat%
set FFMPEG="E:\01_Tools\FFmpegNeu\bin\ffmpeg.exe"
set Opus="E:\01_Tools\Opus\opusenc.exe"
echo *
echo **********************************
echo Um %AudioOutputName% mit OggEnc2 zu encodieren
pause
%Opus% --bitrate %BitrateName% %AudioInputPath%//
%AudioInputName%.wav %AudioOutputPath%%AudioOutputName%.opus
REM ********************************************
set OutputName=%AudioOutputName%
echo *
echo Um %AudioOutputName% mit FFmpeg zu multiplexen
pause
%FFMPEG% -i %AudioOutputPath%%AudioOutputName%.opus//
 -i %VideoInputPath%%VideoInputName% -acodec copy //
-vcodec theora -vb 5000k %TheoraOutputPath%%OutputName%.mp4
echo *
echo ***************
echo JUHU!! FERTIG!!
pause
```

HTML5 Testseite

```
<!DOCTYPE html>
<html>
<style>

.header-design
{
width:  1280;
color:  white;
background-color:       navy;
border: 2px solid navy;
padding:5px;
}

.center
{
text-align: center;
```

```
}


.nav-design
{
width:  400;
color:  navy;
background-color:       white;
border: 2px solid navy;
padding: 5px;
}

article
{
width: 950px;
padding: 40px;
display:
table;
}

.footer-design
{
width:  1280;
color:  white;
background-color:       navy;
border: 2px solid white;
padding:5px;
}

</style>

<head lange="de">
<title>IRT Multichannel Testpage</title>
</head>

<body>
<header class="header-design">
<h1 class="center">IRT Multichannel Testpage</h1>
<p class="center"><strong>Multichannel audiotests with //
different codecs</strong></p>   <!-- Untertitel -->
</header>

<nav class="nav-design">

<p><a href="01_HTML5_Mainpage_02.html">Home</a></p>
<p><a href="04_HTML5_Einbingungsvarianten_Main.html">//
Einbindungsvarianten - Main</a></p>

</nav>

<article>
```

```
68
69 <h3>Videos: MP4 AAC-LC 320kbps 51 VBR HD</h3>
70
71 <!--  Element 1 -->
72 <p> &lt;video&gt; mit Source=MP4 | Mime Type: MP4</p>
73 <video src="http://CI_SHORT_AAC-LC_320_51_48_CBR_HD.mp4" //
74 width="480" height="320" type="video/mp4" controls preload>
75 Der Brwoser unterstützt das Format nicht!
76 </video>
77
78 <!--  Element 2 -->
79 <p>&lt;video&gt; mit Source=MP4 | Mime Type: MP2T</p>
80 <video src="http:CI_SHORT_AAC-LC_320_51_48_CBR_HD.mp4" //
81 width="480" height="320" type="video/MP2T" controls preload>
82 Der Brwoser unterstützt das Format nicht!
83 </video>
84
85 <!--  Element 3 -->
86 <p>&lt;video&gt; mit Source=M3U | Mime Type: audio/x-mpegurl</p>
87 <video src="http://CI_SHORT_AAC-LC_320_51_48_CBR_HD.m3u" //
88 width="480" height="320" type="audio/x-mpegurl"//
89  controls preload>
90 <div>Der Brwoser unterstützt das Format nicht!</div>
91 </video>
92
93 <!--  Element 4 -->
94 <p>Verlinkung auf
95 <a href="http://CI_SHORT_AAC-LC_320_51_48_CBR_HD.m3u">M3U</a>
96 Der Brwoser unterstützt das Format nicht!
97 </p>
98
99 <!--  Element 5 -->
100 <p>Verlinkung auf
101 <a href="http:///CI_SHORT_AAC-LC_320_51_48_CBR_HD.mp4">MP4</a>
102 Der Brwoser unterstützt das Format nicht!
103 </p>
104
105 <!--  Element 6 -->
106 <p>&lt;object&gt; mit Source=MP4 | Mime Type: MP4</p>
107
108 <object data="http://CI_SHORT_AAC-LC_320_51_48_CBR_HD.mp4"//
109 type="video/mp4" width="1300" height="740">
110 <param name="autoplay" value="false">
111 <param name="scale" value="exactfit">
112 Der Brwoser kann das Video nicht wiedergeben!
113 </object>
114
115 <p> &lt;audio&gt; mit Source=AAC-LC | Mime Type: aac</p>
116 <audio type="audio/aac" controls preload="auto">
117 <source src="CI_SHORT_AAC-LC_320_51_48_CBR_HD.aac">
118 Der Brwoser unterstützt das Format nicht!
```

```
</audio>

<p> &lt;audio&gt; mit Source=AAC-LC | Mime Type: aac|STEREO</p>
<audio type="audio/aac" controls preload="auto">
<source src="CI_SHORT_AAC-LC_320_20_48_CBR_HD.aac">
Der Brwoser unterstützt das Format nicht!
</audio>

</article>

<footer class="footer-design">
<h4>Institut für Rundfunktechnik GmbH</h4>
<h5>Bachelorarbeit: Test und Analyse von Möglichkeiten //
zur Online-Distribution von Mehrkanalton für PC //
über das offene Internet</h5>
</footer>
</body>
</html>
```

HbbTV Testergebnisse

Eigenschaften										Ergebnis											
Decoder	Testname	Codec	Bitrate	CBR/VBR	Frequenz	Audio Format	Video Format	Kabel	Interoperabilität	DM100	Kanalzuordnung	LIPE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	Erklärung
AAC-LC																	
AAC-LC																	
Smart	LaTraviata	AAC-LC	128	Pseudo CBR	48	5.1	HD	HDMI	ja	---	falsch	Nein	Ja	Zu Leise -15dB	Schlecht	Schlecht	5.1 -> 2.0, LS auf Ln9, RS auf Ln9, Abstürz nach abspielen des Files -> Reboot
								SPDIF	ja	POM	falsch	Nein	Ja	Zu Leise -15dB	Schlecht	Schlecht	5.1 -> 2.0, LS auf Ln9, RS auf Ln9, Abstürz nach abspielen des Files -> Reboot
								TOSLINK	ja	---	falsch	Nein	Ja	Zu Leise -15dB	Schlecht	Schlecht	5.1 -> 2.0, LS auf Ln9, RS auf Ln9, Abstürz nach abspielen des Files -> Reboot
Smart	Musik	AAC-LC	192	Pseudo CBR	48	5.1	HD	HDMI							Akzeptabel		KO.
								SPDIF							Akzeptabel		KO.
								TOSLINK							Akzeptabel		KO.
Smart	Eishockey	AAC-LC	256	Pseudo CBR	48	5.1	HD	HDMI	ja	---	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0, LS auf Ln9, RS auf Ln9, Abstürz nach abspielen des Files -> Reboot
								SPDIF	ja	POM	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0, LS auf Ln9, RS auf Ln9, Abstürz nach abspielen des Files -> Reboot
								TOSLINK	ja	---	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0, LS auf Ln9, RS auf Ln9, Abstürz nach abspielen des Files -> Reboot
Smart	Musik	AAC-LC	320	Pseudo CBR	48	5.1	HD	HDMI							Gut		KO.
								SPDIF							Gut		KO.
								TOSLINK							Gut		KO.
Smart	Kroemer	AAC-LC	448	Pseudo CBR	48	5.1	HD	HDMI	ja	---	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0, LS auf Ln9, RS auf Ln9, Abstürz nach abspielen des Files -> Reboot K.Q.
								SPDIF	ja	POM	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0, LS auf Ln9, RS auf Ln9, Abstürz nach abspielen des Files -> Reboot K.Q.
								TOSLINK	ja	---	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0, LS auf Ln9, RS auf Ln9, Abstürz nach abspielen des Files -> Reboot K.Q.
Smart	Channelidentification	AAC-LC	640	Pseudo CBR	48	5.1	HD	HDMI	ja	---	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0, LS auf Ln9, RS auf Ln9, Abstürz nach abspielen des Files -> Reboot
								SPDIF	ja	POM	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0, LS auf Ln9, RS auf Ln9, Abstürz nach abspielen des Files -> Reboot
								TOSLINK	ja	---	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0, LS auf Ln9, RS auf Ln9, Abstürz nach abspielen des Files -> Reboot
WAV to AAC-HE Encodierung mit NeroAAC. AAC und H.264 in MP4 gemuxt mit FFmpeg.																	
AAC-HE																	
Smart	LaTraviata	AAC-HE	128	Pseudo CBR	48	5.1	HD	HDMI	ja	---	falsch	Nein	Ja	Zu Leise -15dB	Schlecht	Schlecht	5.1 -> 2.0, LS auf Ln9, RS auf Ln9, Abstürz nach abspielen des Files -> Reboot
								SPDIF	ja	POM	falsch	Nein	Ja	Zu Leise -15dB	Schlecht	Schlecht	5.1 -> 2.0, LS auf Ln9, RS auf Ln9, Abstürz nach abspielen des Files -> Reboot
								TOSLINK	ja	---	falsch	Nein	Ja	Zu Leise -15dB	Schlecht	Schlecht	5.1 -> 2.0, LS auf Ln9, RS auf Ln9, Abstürz nach abspielen des Files -> Reboot
Smart	Musik	AAC-HE	192	Pseudo CBR	48	5.1	HD	HDMI							Akzeptabel		KO.
								SPDIF							Akzeptabel		KO.
								TOSLINK							Akzeptabel		KO.
Smart	Eishockey	AAC-HE	256	Pseudo CBR	48	5.1	HD	HDMI	ja	---	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0, LS auf Ln9, RS auf Ln9, Abstürz nach abspielen des Files -> Reboot
								SPDIF	ja	POM	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0, LS auf Ln9, RS auf Ln9, Abstürz nach abspielen des Files -> Reboot
								TOSLINK	ja	---	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0, LS auf Ln9, RS auf Ln9, Abstürz nach abspielen des Files -> Reboot
Smart	Musik	AAC-HE	320	Pseudo CBR	48	5.1	HD	HDMI							Gut		KO.
								SPDIF							Gut		KO.
								TOSLINK							Gut		KO.
Smart	Kroemer	AAC-HE	448	Pseudo CBR	48	5.1	HD	HDMI	ja	---	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0, LS auf Ln9, RS auf Ln9, Abstürz nach abspielen des Files -> Reboot K.Q.
								SPDIF	ja	POM	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0, LS auf Ln9, RS auf Ln9, Abstürz nach abspielen des Files -> Reboot K.Q.
								TOSLINK	ja	---	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0, LS auf Ln9, RS auf Ln9, Abstürz nach abspielen des Files -> Reboot K.Q.
Smart	Channelidentification	AAC-HE	640	Pseudo CBR	48	5.1	HD	HDMI	ja	---	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0, LS auf Ln9, RS auf Ln9, Kein Audio nach abspielen des Files -> Reboot
								SPDIF	ja	POM	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0, LS auf Ln9, RS auf Ln9, Kein Audio nach abspielen des Files -> Reboot.
								TOSLINK	ja	---	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0, LS auf Ln9, RS auf Ln9, Kein Audio nach abspielen des Files -> Reboot.
WAV zu AAC-HE Encodierung mit NeroAAC. AAC und H.264 in MP4 gemuxt mit FFmpeg.																	
Dolby-AC3																	
Smart	LaTraviata	AC3	128	CBR	48	5.1	HD	HDMI	ja	---	Richtig	Ja	Ja	Gleich	Schlecht	Gut	Kein Audio nach abspielen des Files -> Reboot
								SPDIF	ja	AC3	Richtig	Ja	Ja	Gleich	Schlecht	Gut	Kein Audio nach abspielen des Files -> Reboot
								TOSLINK	ja	---	Richtig	Ja	Ja	Gleich	Schlecht	Gut	Kein Audio nach abspielen des Files -> Reboot
Smart	Musik	AC3	192	CBR	48	5.1	HD	HDMI							Schlecht		KO.
								SPDIF							Schlecht		KO.
								TOSLINK							Schlecht		KO.
Smart	Eishockey	AC3	256	CBR	48	5.1	HD	HDMI							Schlecht		KO.
								SPDIF							Schlecht		KO.
								TOSLINK							Schlecht		KO.
Smart	Musik	AC3	320	CBR	48	5.1	HD	HDMI	ja	---	Richtig	Ja	Ja	Gleich	Akzeptabel	Gut	Kein Audio nach abspielen des Files -> Reboot
								SPDIF	ja	AC3	Richtig	Ja	Ja	Gleich	Akzeptabel	Gut	Kein Audio nach abspielen des Files -> Reboot
								TOSLINK	ja	---	Richtig	Ja	Ja	Gleich	Akzeptabel	Gut	Kein Audio nach abspielen des Files -> Reboot
Smart	Kroemer	AC3	448	CBR	48	5.1	HD	HDMI	ja	---	Richtig	Ja	Ja	Gleich	Gut	Gut	Kein Audio nach abspielen des Files -> Reboot. K.Q.
								SPDIF	ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Kein Audio nach abspielen des Files -> Reboot. K.Q.
								TOSLINK	ja	---	Richtig	Ja	Ja	Gleich	Gut	Gut	Kein Audio nach abspielen des Files -> Reboot. K.Q.
Smart	Channelidentification	AC3	640	CBR	48	5.1	HD	HDMI	ja	---	Richtig	Ja	Ja	Gleich	Gut	Gut	Kein Audio nach abspielen des Files -> Reboot. K.Q.
								SPDIF	ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Kein Audio nach abspielen des Files -> Reboot
								TOSLINK	ja	---	Richtig	Ja	Ja	Gleich	Gut	Gut	Kein Audio nach abspielen des Files -> Reboot
WAV zu AC3 Encodierung mit Aften. AC3 und H.264 in MP4 gemuxt mit FFmpeg.																	
Dolby-DD+																	
Smart	LaTraviata	AC3-DD+	192	CBR	48	5.1	HD	HDMI	Nein	---						Schlecht	File wird abgespielt, aber kein Audio, Abstürz nach abspielen des Files -> Reboot
								SPDIF	Nein	Notinput						Schlecht	File wird abgespielt, aber kein Audio, Abstürz nach abspielen des Files -> Reboot
								TOSLINK	Nein	---						Schlecht	File wird abgespielt, aber kein Audio, Abstürz nach abspielen des Files -> Reboot
Smart	Eishockey	AC3-DD+	256	CBR	48	5.1	HD	HDMI	Nein	---						Schlecht	File wird abgespielt, aber kein Audio, Abstürz nach abspielen des Files -> Reboot
								SPDIF	Nein	Notinput						Schlecht	File wird abgespielt, aber kein Audio, Abstürz nach abspielen des Files -> Reboot
								TOSLINK	Nein	---						Schlecht	File wird abgespielt, aber kein Audio, Abstürz nach abspielen des Files -> Reboot
Smart	Musik	AC3-DD+	320	CBR	48	5.1	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
Smart	Kroemer	AC3-DD+	448	CBR	48	5.1	HD	HDMI	Nein	---						Schlecht	File wird abgespielt, aber kein Audio, Abstürz nach abspielen des Files -> Reboot
								SPDIF	Nein	Notinput						Schlecht	File wird abgespielt, aber kein Audio, Abstürz nach abspielen des Files -> Reboot
								TOSLINK	Nein	---						Schlecht	File wird abgespielt, aber kein Audio, Abstürz nach abspielen des Files -> Reboot
Smart	Channelidentification	AC3-DD+	640	CBR	48	5.1	HD	HDMI	Nein	---						Schlecht	File wird abgespielt, aber kein Audio, Abstürz nach abspielen des Files -> Reboot
								SPDIF	Nein	Notinput						Schlecht	File wird abgespielt, aber kein Audio, Abstürz nach abspielen des Files -> Reboot
								TOSLINK	Nein	---						Schlecht	File wird abgespielt, aber kein Audio, Abstürz nach abspielen des Files -> Reboot
Smart	Eishockey	AC3-DD+	1024	CBR	48	5.1	HD	HDMI	Nein	---						Schlecht	File wird abgespielt, aber kein Audio, Abstürz nach abspielen des Files -> Reboot
								SPDIF	Nein	Notinput						Schlecht	File wird abgespielt, aber kein Audio, Abstürz nach abspielen des Files -> Reboot
								TOSLINK	Nein	---						Schlecht	File wird abgespielt, aber kein Audio, Abstürz nach abspielen des Files -> Reboot
Smart	Eishockey	AC3-DD+	3024	CBR	48	5.1	HD	HDMI	Nein	---						Schlecht	Kein Audio, kein Video -> Reboot
								SPDIF	Nein	Notinput						Schlecht	Kein Audio, kein Video -> Reboot
								TOSLINK	Nein	---						Schlecht	Kein Audio, kein Video -> Reboot
WAV zu DD+ Encodierung. WAV zu H.264 Encodierung und DD+ und H.264 Muxing mit TotalCodeStudio																	
44.1 kHz																	
Smart		AAC-LC	128	Pseudo CBR	44.1	5.1	HD	HDMI									N.G.
								SPDIF									N.G.

								TOSLINK									N.G.
Smart	Channelldentification	AAC-LC	640	Pseudo CBR	44.1	5.1	HD	HDMI	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Kein Audio nach abspielen des Files -> Reboot
								SPDIF	Ja	PCM	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Kein Audio nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Kein Audio nach abspielen des Files -> Reboot.
Smart		AAC-HE	128	Pseudo CBR	44.1	5.1	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
Smart	Channelldentification	AAC-HE	640	Pseudo CBR	44.1	5.1	HD	HDMI	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Kein Audio nach abspielen des Files -> Reboot
								SPDIF	Ja	PCM	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Kein Audio nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Kein Audio nach abspielen des Files -> Reboot.
Smart		AC3	128	CBR	44.1	5.1	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
Smart	Channelldentification	AC3	640	CBR	44.1	5.1	HD	HDMI	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Kein Audio nach abspielen des Files -> Reboot
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Kein Audio nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Kein Audio nach abspielen des Files -> Reboot
Smart		AC3-DD+	128	CBR	44.1	5.1	HD	HDMI									N.V.
								SPDIF									N.V.
								TOSLINK									N.V.
Smart	Channelldentification	AC3-DD+	640	CBR	44.1	5.1	HD	HDMI									N.V.
								SPDIF									N.V.
								TOSLINK									N.V.
Siehe AAC, AC3 und DD+																	
VBR									Interoperabilität	DM 100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
Smart		AAC-LC	128	VBR	48	5.1	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
Smart	Eishockey	AAC-LC	640	VBR	48	5.1	HD	HDMI	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Kein Audio nach abspielen des Files -> Reboot
								SPDIF	Ja	PCM	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Kein Audio nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Kein Audio nach abspielen des Files -> Reboot.
Smart		AAC-HE	128	VBR	48	5.1	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
Smart	Eishockey	AAC-HE	640	VBR	48	5.1	HD	HDMI	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Kein Audio nach abspielen des Files -> Reboot
								SPDIF	Ja	PCM	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Kein Audio nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Kein Audio nach abspielen des Files -> Reboot.
Siehe AAC																	
STEREO									Interoperabilität	DM 100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
Smart		AAC-LC	128	Pseudo CBR	48	2.0	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
Smart	Musik	AAC-LC	640	Pseudo CBR	48	2.0	HD	HDMI	Ja	-----	Richtig	Nein	Ja	Zu Leise -15dB	Gut	Akzeptabel	Kein Audio nach abspielen des Files -> Reboot
								SPDIF	Ja	PCM	Richtig	Nein	Ja	Zu Leise -15dB	Gut	Akzeptabel	Kein Audio nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	Richtig	Nein	Ja	Zu Leise -15dB	Gut	Akzeptabel	Kein Audio nach abspielen des Files -> Reboot
Smart		AAC-HE	128	Pseudo CBR	48	2.0	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
Smart	Musik	AAC-HE	640	Pseudo CBR	48	2.0	HD	HDMI	Ja	-----	Richtig	Nein	Ja	Zu Leise -15dB	Gut	Akzeptabel	Kein Audio nach abspielen des Files -> Reboot
								SPDIF	Ja	PCM	Richtig	Nein	Ja	Zu Leise -15dB	Gut	Akzeptabel	Kein Audio nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	Richtig	Nein	Ja	Zu Leise -15dB	Gut	Akzeptabel	Kein Audio nach abspielen des Files -> Reboot
Smart		AC3	128	CBR	48	2.0	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
Smart	Musik	AC3	640	CBR	48	2.0	HD	HDMI	Ja	-----	Richtig	Nein	Ja	Gleich	Gut	Gut	Kein Audio nach abspielen des Files -> Reboot
								SPDIF	Ja	AC3	Richtig	Nein	Ja	Gleich	Gut	Gut	Kein Audio nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	Richtig	Nein	Ja	Gleich	Gut	Gut	Kein Audio nach abspielen des Files -> Reboot
Smart		AC3-DD+	128	CBR	48	2.0	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
Smart	Musik	AC3-DD+	640	CBR	48	2.0	HD	HDMI	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								SPDIF	Nein	No Input						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								TOSLINK	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
Siehe AAC, AC3 und DD+																	
SD									Interoperabilität	DM 100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
Smart		AAC-LC	128	Pseudo CBR	48	5.1	SD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
Smart	LaTraviata	AAC-LC	640	Pseudo CBR	48	5.1	SD	HDMI	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Kein Audio nach abspielen des Files -> Reboot
								SPDIF	Ja	PCM	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Kein Audio nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Kein Audio nach abspielen des Files -> Reboot.
Smart		AAC-HE	128	Pseudo CBR	48	5.1	SD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
Smart	LaTraviata	AAC-HE	640	Pseudo CBR	48	5.1	SD	HDMI	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Kein Audio nach abspielen des Files -> Reboot
								SPDIF	Ja	PCM	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Kein Audio nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Kein Audio nach abspielen des Files -> Reboot.
Smart		AC3	128	CBR	48	5.1	SD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
Smart	LaTraviata	AC3	640	CBR	48	5.1	SD	HDMI	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Kein Audio nach abspielen des Files -> Reboot
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Kein Audio nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Kein Audio nach abspielen des Files -> Reboot
Smart		AC3-DD+	192	CBR	48	5.1	SD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
Smart	LaTraviata	AC3-DD+	640	CBR	48	5.1	SD	HDMI	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								SPDIF	Nein	No Input						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								TOSLINK	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
Siehe AAC, AC3 und DD+																	
TS									Interoperabilität	DM 100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
								HDMI	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot

Smart	Channelldentification	AAC-LC	448	Pseudo CBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
Smart	Channelldentification	AAC-HE	448	Pseudo CBR	48	5.1	HD	HDMI	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
								SPDIF	Ja	PCM	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
Smart	Channelldentification	AC3	448	CBR	48	5.1	HD	HDMI	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Kein Audio und Video nach abspielen des Files -> Reboot
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Kein Audio und Video nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Kein Audio und Video nach abspielen des Files -> Reboot
Alle .ts mit TotalCodeStudio erstellt, um DVB Konformität zu gewährleisten.																	
Audio Only Elementary									Interoperabilität	DM 100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
Smart	Musik	AAC-LC	640	Pseudo CBR	48	5.1	HD	HDMI	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
								SPDIF	Ja	PCM	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
Smart	Musik	AAC-HE	640	Pseudo CBR	48	5.1	HD	HDMI	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
								SPDIF	Ja	PCM	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
Smart	Musik	AC3	640	CBR	48	5.1	HD	HDMI	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Absturz nach abspielen des Files -> Reboot
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Absturz nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Absturz nach abspielen des Files -> Reboot
Smart	Musik	AC3-DD+	640	CBR	48	5.1	HD	HDMI	Nein	-----	Richtig	Ja	Nein	Gleich	Gut	Schlecht	File wird abgespielt, aber Audio ruckelt. Absturz nach abspielen des Files -> Reeboot.
								SPDIF	Nein	No Input						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								TOSLINK	Nein	-----	Richtig	Ja	Nein	Gleich	Gut	Schlecht	File wird abgespielt, aber Audio ruckelt. Absturz nach abspielen des Files -> Reeboot.
WAV zu AAC Encodierung mit NeroAAC. WAV zu AC3 Encodierung mit Aften. WAV zu DD+ Encodierung mit TotalCode Studio																	
Audio Only MP4									Interoperabilität	DM 100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
Smart	Channelldentification	AAC-LC	448	Pseudo CBR	48	5.1	HD	HDMI	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
								SPDIF	Ja	PCM	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
Smart	Channelldentification	AAC-HE	448	Pseudo CBR	48	5.1	HD	HDMI	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
								SPDIF	Ja	PCM	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
Smart	Channelldentification	AC3	448	CBR	48	5.1	HD	HDMI	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Absturz nach abspielen des Files -> Reboot
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Absturz nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Absturz nach abspielen des Files -> Reboot
Smart	Channelldentification	AC3-DD+	448	CBR	48	5.1	HD	HDMI	Nein	-----	Richtig	Ja	Nein	Gleich	Gut	Schlecht	File wird abgespielt, aber Audio ruckelt. Absturz nach abspielen des Files -> Reeboot.
								SPDIF	Nein	No Input						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								TOSLINK	Nein	-----	Richtig	Ja	Nein	Gleich	Gut	Schlecht	File wird abgespielt, aber Audio ruckelt. Absturz nach abspielen des Files -> Reeboot.
Siehe AAC, AC3 und DD+																	
Smart	Dolby Test Material	AC3-DD+	192	CBR	48	5.1	SD	HDMI	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								SPDIF	Nein	No Input						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								TOSLINK	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
Smart	Dolby Test Material	AC3-DD+	320	CBR	48	7.1	SD	HDMI	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								SPDIF	Nein	No Input						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								TOSLINK	Nein	-----						Schlecht	Bitrate: 384kbps. File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
DMG									Interoperabilität	DM 100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
Smart	Channelldentification	AC3-DD+	448	CBR	48	5.1	HD	HDMI	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								SPDIF	Nein	No Input						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								-------	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
Smart	Eishockey	AC3-DD+	448	CBR	48	5.1	HD	HDMI	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								SPDIF	Nein	No Input						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								-------	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
Smart	Musik	AC3-DD+	448	CBR	48	5.1	HD	HDMI	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								SPDIF	Nein	No Input						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								-------	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
Smart	Kroemer	AC3-DD+	448	CBR	48	5.1	HD	HDMI	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								SPDIF	Nein	No Input						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								-------	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
Smart	LaTraviata	AC3-DD+	448	CBR	48	5.1	HD	HDMI	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								SPDIF	Nein	No Input						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								-------	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
Smart	Aida	AC3-DD+	448	CBR	48	5.1	HD	HDMI	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								SPDIF	Nein	No Input						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								-------	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
Smart																	

Legende: **Erkläung:**

N.G.	Nicht Getestet, da sich die einzelnen Bitraten gleich verhalten.
N.V.	Nicht vorhanden, da TotalCode Studio kein 44kHz DD+ erzeugen kann.
K.Q.	Klangqualitätstest durchgeführt mit LaTraviata.

Set-Top-Box Details:

Modellnummer:	cx10
Hardware Version	14000569180000065001301000000000
Software Version:	V1_36_2_2
Build Version:	ker126-sdk4384-fs262-thr692-build754
Bootloader Version:	V0.1.5-1
Erstellungsdatum Software:	2012-06-11
Seriennummer:	5N605110310044 61
IRD Seriennummer:	17 3784 0918 18
Chipset UID:	00 1363 2651 C8 Rev:00, UID: 22 1961 2345 64

Smart	Channelldentification	AAC-LC	448	Pseudo CBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
Smart	Channelldentification	AAC-HE	448	Pseudo CBR	48	5.1	HD	HDMI	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
								SPDIF	Ja	PCM	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
Smart	Channelldentification	AC3	448	CBR	48	5.1	HD	HDMI	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Kein Audio und Video nach abspielen des Files -> Reboot
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Kein Audio und Video nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Kein Audio und Video nach abspielen des Files -> Reboot
Alle .ts mit TotalCodeStudio erstellt, um DVB Konformität zu gewährleisten.																		
Audio Only Elementary									Interoperabilität	DM 100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis		
Smart	Musik	AAC-LC	640	Pseudo CBR	48	5.1	HD	HDMI	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
								SPDIF	Ja	PCM	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
Smart	Musik	AAC-HE	640	Pseudo CBR	48	5.1	HD	HDMI	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
								SPDIF	Ja	PCM	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
Smart	Musik	AC3	640	CBR	48	5.1	HD	HDMI	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Absturz nach abspielen des Files -> Reboot
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Absturz nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Absturz nach abspielen des Files -> Reboot
Smart	Musik	AC3-DD+	640	CBR	48	5.1	HD	HDMI	Nein	-----	Richtig	Ja	Nein	Gleich	Gut	Schlecht	File wird abgespielt, aber Audio ruckelt. Absturz nach abspielen des Files -> Reeboot.
								SPDIF	Nein	No Input						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								TOSLINK	Nein	-----	Richtig	Ja	Nein	Gleich	Gut	Schlecht	File wird abgespielt, aber Audio ruckelt. Absturz nach abspielen des Files -> Reeboot.
WAV zu AAC Encodierung mit NeroAAC. WAV zu AC3 Encodierung mit Aften. WAV zu DD+ Encodierung mit TotalCode Studio																		
Audio Only MP4									Interoperabilität	DM 100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis		
Smart	Channelldentification	AAC-LC	448	Pseudo CBR	48	5.1	HD	HDMI	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
								SPDIF	Ja	PCM	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
Smart	Channelldentification	AAC-HE	448	Pseudo CBR	48	5.1	HD	HDMI	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
								SPDIF	Ja	PCM	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	falsch	Nein	Ja	Zu Leise -15dB	Gut	Schlecht	5.1 -> 2.0. LS auf L+R, RS auf L+R. Absturz nach abspielen des Files -> Reboot
Smart	Channelldentification	AC3	448	CBR	48	5.1	HD	HDMI	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Absturz nach abspielen des Files -> Reboot
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Absturz nach abspielen des Files -> Reboot
								TOSLINK	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Absturz nach abspielen des Files -> Reboot
Smart	Channelldentification	AC3-DD+	448	CBR	48	5.1	HD	HDMI	Nein	-----	Richtig	Ja	Nein	Gleich	Gut	Schlecht	File wird abgespielt, aber Audio ruckelt. Absturz nach abspielen des Files -> Reeboot.
								SPDIF	Nein	No Input						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								TOSLINK	Nein	-----	Richtig	Ja	Nein	Gleich	Gut	Schlecht	File wird abgespielt, aber Audio ruckelt. Absturz nach abspielen des Files -> Reeboot.
Siehe AAC, AC3 und DD+																		
Smart	Dolby Test Material	AC3-DD+	192	CBR	48	5.1	SD	HDMI	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								SPDIF	Nein	No Input						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								TOSLINK	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
Smart	Dolby Test Material	AC3-DD+	320	CBR	48	7.1	SD	HDMI	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								SPDIF	Nein	No Input						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								TOSLINK	Nein	-----						Schlecht	Bitrate: 384kbps. File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
DMG									Interoperabilität	DM 100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis		
Smart	Channelldentification	AC3-DD+	448	CBR	48	5.1	HD	HDMI	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								SPDIF	Nein	No Input						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								-------	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
Smart	Eishockey	AC3-DD+	448	CBR	48	5.1	HD	HDMI	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								SPDIF	Nein	No Input						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								-------	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
Smart	Musik	AC3-DD+	448	CBR	48	5.1	HD	HDMI	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								SPDIF	Nein	No Input						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								-------	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
Smart	Kroemer	AC3-DD+	448	CBR	48	5.1	HD	HDMI	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								SPDIF	Nein	No Input						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								-------	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
Smart	LaTraviata	AC3-DD+	448	CBR	48	5.1	HD	HDMI	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								SPDIF	Nein	No Input						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								-------	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
Smart	Aida	AC3-DD+	448	CBR	48	5.1	HD	HDMI	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								SPDIF	Nein	No Input						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
								-------	Nein	-----						Schlecht	File wird abgespielt, aber kein Audio. Absturz nach abspielen des Files -> Reeboot.
Smart																		

Legende: **Erkläung:**

N.G.	Nicht Getestet, da sich die einzelnen Bitraten gleich verhalten.
N.V.	Nicht vorhanden, da TotalCode Studio kein 44kHz DD+ erzeugen kann.
K.Q.	Klangqualitätstest durchgeführt mit LaTraviata.

Set-Top-Box Details:

Modellnummer:	cx10
Hardware Version	14000569180000065001301000000000
Software Version:	V1_36_2_2
Build Version:	ker126-sdk4384-fs262-thr692-build754
Bootloader Version:	V0.1.5-1
Erstellungsdatum Software:	2012-06-11
Seriennummer:	5N605111031004461
IRD Seriennummer:	17 3784 0918 18
Chipset UID:	00 1363 2651 C8 Rev:00, UID: 22 1961 2345 64

Eigenschaften									Ergebnis								
Decoder	Testname	Codec	Bitrate	CBR/VBR	Frequenz	Audio Format	Video Format	Kabel	Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	Erklärung
AAC-LC									**AAC_LC**								
HUMAX	Channelldentification	AAC-LC	128	Pseudo CBR	48	5.1	HD	HDMI	Ja	----	falsch	Nein	Ja	Zu Leise -30dB	Schlecht	Schlecht	5.1 -> 2.0, ohne LFE
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Schlecht	Schlecht	5.1 -> 2.0, ohne LFE
HUMAX	Musik	AAC-LC	192	Pseudo CBR	48	5.1	HD	HDMI	Ja	----	falsch	Nein	Ja	Zu Leise -30dB	Akzeptabel	Schlecht	5.1 -> 2.0, ohne LFE
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Akzeptabel	Schlecht	5.1 -> 2.0, ohne LFE
HUMAX	Eishockey	AAC-LC	256	Pseudo CBR	48	5.1	HD	HDMI	Ja	----	falsch	Nein	Ja	Zu Leise -30dB	Gut	Schlecht	5.1 -> 2.0, ohne LFE
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	5.1 -> 2.0, ohne LFE
HUMAX	Musik	AAC-LC	320	Pseudo CBR	48	5.1	HD	HDMI	Ja	----	falsch	Nein	Ja	Zu Leise -30dB	Gut	Schlecht	5.1 -> 2.0, ohne LFE
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	5.1 -> 2.0, ohne LFE
HUMAX	Kroemer	AAC-LC	448	Pseudo CBR	48	5.1	HD	HDMI	Ja	----	falsch	Nein	Ja	Zu Leise -30dB	Gut	Schlecht	5.1 -> 2.0, ohne LFE
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	5.1 -> 2.0, ohne LFE
HUMAX	Channelldentification	AAC-LC	640	Pseudo CBR	48	5.1	HD	HDMI	Ja	----	falsch	Nein	Ja	Zu Leise -30dB	Gut	Schlecht	5.1 -> 2.0, ohne LFE
								SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0, ohne LFE
WAV to AAC-HE Encodierung mit NeroAAC. AAC und H264 in MP4 gemuxt mit FFmpeg.																	
AAC-HE									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
HUMAX	LaTraviata	AAC-HE	128	Pseudo CBR	48	5.1	HD	HDMI	Ja	----	falsch	Nein	Ja	Zu Leise -30dB	Schlecht	Schlecht	5.1 -> 2.0, ohne LFE
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Schlecht	Schlecht	5.1 -> 2.0, ohne LFE
HUMAX	Musik	AAC-HE	192	Pseudo CBR	48	5.1	HD	HDMI	Ja	----	falsch	Nein	Ja	Zu Leise -30dB	Akzeptabel	Schlecht	5.1 -> 2.0, ohne LFE
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Akzeptabel	Schlecht	5.1 -> 2.0, ohne LFE
HUMAX	Eishockey	AAC-HE	256	Pseudo CBR	48	5.1	HD	HDMI	Ja	----	falsch	Nein	Ja	Zu Leise -30dB	Gut	Schlecht	5.1 -> 2.0, ohne LFE
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	5.1 -> 2.0, ohne LFE
HUMAX	Musik	AAC-HE	320	Pseudo CBR	48	5.1	HD	HDMI	Ja	----	falsch	Nein	Ja	Zu Leise -30dB	Gut	Schlecht	5.1 -> 2.0, ohne LFE
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	5.1 -> 2.0, ohne LFE
HUMAX	Kroemer	AAC-HE	448	Pseudo CBR	48	5.1	HD	HDMI	Ja	----	falsch	Nein	Ja	Zu Leise -30dB	Gut	Schlecht	5.1 -> 2.0, ohne LFE
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	5.1 -> 2.0, ohne LFE
HUMAX	Channelldentification	AAC-HE	640	Pseudo CBR	48	5.1	HD	HDMI	Ja	----	falsch	Nein	Ja	Zu Leise -30dB	Gut	Schlecht	5.1 -> 2.0, ohne LFE
								SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0, ohne LFE
WAV zu AAC-HE Encodierung mit NeroAAC. AAC und H264 in MP4 gemuxt mit FFmpeg.																	
Dolby-AC3									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
HUMAX	LaTraviata	AC3	128	CBR	48	5.1	HD	HDMI	Ja	----	Richtig	Ja	Ja	Lauter	Schlecht	Akzeptabel	Ausgangsmaterial ruckelt.
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Lauter	Schlecht	Akzeptabel	
HUMAX	Musik	AC3	192	CBR	48	5.1	HD	HDMI	Ja	----	Richtig	Ja	Ja	Lauter	Schlecht	Akzeptabel	
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Lauter	Schlecht	Akzeptabel	
HUMAX	Eishockey	AC3	256	CBR	48	5.1	HD	HDMI	Ja	----	Richtig	Ja	Ja	Lauter	Akzeptabel	Akzeptabel	
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Lauter	Akzeptabel	Akzeptabel	
HUMAX	Musik	AC3	320	CBR	48	5.1	HD	HDMI	Ja	----	Richtig	Ja	Ja	Gleich	Gut	Gut	
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	
HUMAX	Kroemer	AC3	448	CBR	48	5.1	HD	HDMI	Ja	----	Richtig	Ja	Ja	Gleich	Gut	Gut	
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Lauter	Gut	Gut	
HUMAX	Channelldentification	AC3	640	CBR	48	5.1	HD	HDMI	Ja	----	Richtig	Ja	Ja	Gleich	Gut	Gut	
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	
WAV zu AC3 Encodierung mit Aften. AC3 und H264 in MP4 gemuxt mit FFmpeg.																	
Dolby-DD+									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
HUMAX	Musik	AC3-DD+	192	CBR	48	5.1	HD	HDMI	Ja	----	Richtig	Ja	Ja	Gleich	Akzeptabel	Gut	DD+ pass through.
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Akzeptabel	Gut	AC3 Transcoding mit 640kbps.
HUMAX	Eishockey	AC3-DD+	256	CBR	48	5.1	HD	HDMI	Ja	----	Richtig	Ja	Ja	Gleich	Gut	Gut	DD+ pass through.
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	AC3 Transcoding mit 640kbps.
HUMAX	Musik	AC3-DD+	320	CBR	48	5.1	HD	HDMI	Ja	----	Richtig	Ja	Ja	Gleich	Gut	Gut	DD+ pass through.
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	AC3 Transcoding mit 640kbps.
HUMAX	Kroemer	AC3-DD+	448	CBR	48	5.1	HD	HDMI	Ja	----	Richtig	Ja	Ja	Gleich	Gut	Gut	DD+ pass through.
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	AC3 Transcoding mit 640kbps.
								HDMI	Ja	----	Richtig	Ja	Ja	Gleich	Gut	Gut	DD+ pass through.

HUMAX	Channelldentification	AC3-DD+	640	CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	AC3 Transcoding mit 640kbps.

HUMAX	Eishockey	AC3-DD+	1024	CBR	48	5.1	HD	HDMI	Ja	----	Richtig	Ja	Ja	Gleich	Gut	Gut	DD+ pass through.
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	AC3 Transcoding mit 640kbps.

HUMAX	Eishockey	AC3-DD+	3024	CBR	48	5.1	HD	HDMI	Nein							Schlecht	Kein Audio.
								SPDIF	Nein							Schlecht	Kein Audio.

WAV zu DD+ Encodierung, YUV zu H264 Encodierung und DD+ und H264 Muxing mit TotalCode Studio																	
44.1 kHz									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
HUMAX		AAC-LC	128	Pseudo CBR	44.1	5.1	HD	HDMI									N.G.
								SPDIF									N.G.

HUMAX	Channelldentification	AAC-LC	640	Pseudo CBR	44.1	5.1	HD	HDMI	Ja	----	falsch	Nein	Ja	Zu Leise -30dB	Gut	Schlecht	5.1 -> 2.0, ohne LFE
								SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0, ohne LFE

HUMAX		AAC-HE	128	Pseudo CBR	44.1	5.1	HD	HDMI									N.G.
								SPDIF									N.G.

HUMAX	Channelldentification	AAC-HE	640	Pseudo CBR	44.1	5.1	HD	HDMI	Ja	----	falsch	Nein	Ja	Zu Leise -30dB	Gut	Schlecht	5.1 -> 2.0, ohne LFE
								SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0, ohne LFE

HUMAX		AC3	128	CBR	44.1	5.1	HD	HDMI									N.G.
								SPDIF									N.G.

HUMAX	Channelldentification	AC3	640	CBR	44.1	5.1	HD	HDMI	Ja	----	Richtig	Ja	Ja	Gleich	Gut	Gut	
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	

HUMAX		AC3-DD+	128	CBR	44.1	5.1	HD	HDMI									N.V.
								SPDIF									N.V.

HUMAX	Channelldentification	AC3-DD+	640	CBR	44.1	5.1	HD	HDMI									N.V.
								SPDIF									N.V.

Siehe AAC, AC3 und DD+																	
VBR									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
HUMAX		AAC-LC	128	VBR	48	5.1	HD	HDMI									N.G.
								SPDIF									N.G.

HUMAX	Eishockey	AAC-LC	640	VBR	48	5.1	HD	HDMI	Ja	----	falsch	Nein	Ja	Zu Leise -30dB	Gut	Schlecht	5.1 -> 2.0, ohne LFE
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	5.1 -> 2.0, ohne LFE

HUMAX		AAC-HE	128	VBR	48	5.1	HD	HDMI									N.G.
								SPDIF									N.G.

HUMAX	Eishockey	AAC-HE	640	VBR	48	5.1	HD	HDMI	Ja	----	falsch	Nein	Ja	Zu Leise -30dB	Gut	Schlecht	5.1 -> 2.0, ohne LFE
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	5.1 -> 2.0, ohne LFE

Siehe AAC																	
STEREO									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
HUMAX		AAC-LC	128	Pseudo CBR	48	2.0	HD	HDMI									N.G.
								SPDIF									N.G.

HUMAX	Channelldentification	AAC-LC	640	Pseudo CBR	48	2.0	HD	HDMI	Ja	----	Richtig	Nein	Ja	Zu Leise -30dB	Gut	Schlecht	
								SPDIF	Ja	PCM	Richtig	Nein	Ja	Gleich	Gut	Gut	

HUMAX		AAC-HE	128	Pseudo CBR	48	2.0	HD	HDMI									N.G.
								SPDIF									N.G.

HUMAX	Channelldentification	AAC-HE	640	Pseudo CBR	48	2.0	HD	HDMI	Ja	----	Richtig	Nein	Ja	Zu Leise -30dB	Gut	Schlecht	
								SPDIF	Ja	PCM	Richtig	Nein	Ja	Gleich	Gut	Gut	

HUMAX		AC3	128	CBR	48	2.0	HD	HDMI									N.G.
								SPDIF									N.G.

HUMAX	Channelldentification	AC3	640	CBR	48	2.0	HD	HDMI	Ja	----	Richtig	Nein	Ja	Gleich	Gut	Gut	
								SPDIF	Ja	AC3	Richtig	Nein	Ja	Gleich	Gut	Gut	

HUMAX		AC3-DD+	128	CBR	48	2.0	HD	HDMI									N.G.
								SPDIF									N.G.

HUMAX	Channelldentification	AC3-DD+	640	CBR	48	2.0	HD	HDMI	Ja	----	Richtig	Nein	Ja	Gleich	Gut	Gut	DD+ pass through.
								SPDIF	Ja	AC3	Richtig	Nein	Ja	Gleich	Gut	Gut	AC3 Transcoding mit 640kbps.

Siehe AAC, AC3 und DD+																	
SD									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
HUMAX		AAC-LC	128	Pseudo CBR	48	5.1	SD	HDMI									N.G.
								SPDIF									N.G.

HUMAX	LaTraviata	AAC-LC	640	Pseudo CBR	48	5.1	SD	HDMI	Ja	----	falsch	Nein	Ja	Zu Leise -30dB	Gut	Schlecht	5.1 -> 2.0, ohne LFE
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	5.1 -> 2.0, ohne LFE

HUMAX		AAC-HE	128	Pseudo CBR	48	5.1	SD	HDMI									N.G.
								SPDIF									N.G.

HUMAX	LaTraviata	AAC-HE	640	Pseudo CBR	48	5.1	SD	HDMI	Ja	----	falsch	Nein	Ja	Zu Leise -30dB	Gut	Schlecht	5.1 -> 2.0, ohne LFE
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	5.1 -> 2.0, ohne LFE

HUMAX		AC3	128	CBR	48	5.1	SD	HDMI									N.G.
								SPDIF									N.G.

HUMAX	LaTraviata	AC3	640	CBR	48	5.1	SD	HDMI	Ja	----	Richtig	Ja	Ja	Gleich	Gut	Gut	
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	

HUMAX		AC3-DD+	192	CBR	48	5.1	SD	HDMI									N.G.
								SPDIF									N.G.

HUMAX	LaTraviata	AC3-DD+	640	CBR	48	5.1	SD	HDMI	Ja	----	Richtig	Ja	Ja	Gleich	Gut	Gut	DD+ pass through.
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	AC3 Transcoding mit 640kbps.

Siehe AAC, AC3 und DD+																		
TS									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis		
HUMAX	Channelldentification	AAC-LC	448	Pseudo CBR	48	5.1	HD	HDMI	Ja	----	falsch	Nein	Ja	Zu Leise -30dB	Gut	Schlecht	5.1 -> 2.0, ohne LFE
								SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0, ohne LFE

HUMAX	Channelldentification	AAC-HE	448	Pseudo CBR	48	5.1	HD	HDMI	Ja	----	falsch	Nein	Ja	Zu Leise -30dB	Gut	Schlecht	5.1 -> 2.0, ohne LFE
								SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0, ohne LFE

HUMAX	Channelldentification	AC3	448	CBR	48	5.1	HD	HDMI	Ja	----	Richtig	Ja	Ja	Gleich	Gut	Gut	
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	

Alle .ts mit TotalCodeStudio erstellt, um DVB Konformität zu gewährleisten.																		
Audio Only Elementary									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis		
HUMAX	Musik	AAC-LC	640	Pseudo CBR	48	5.1	HD	HDMI	Ja	----	falsch	Nein	Ja	Zu Leise -30dB	Gut	Schlecht	5.1 -> 2.0, ohne LFE
								SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0, ohne LFE

HUMAX	Musik	AAC-HE	640	Pseudo CBR	48	5.1	HD	HDMI	Ja	----	falsch	Nein	Ja	Zu Leise -30dB	Gut	Schlecht	5.1 -> 2.0, ohne LFE
								SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0, ohne LFE

HUMAX	Musik	AC3	640	CBR	48	5.1	HD	HDMI	Nein	----						Schlecht	Transitioned to error state for -> Unidentified Error. Box-Reset.
								SPDIF	Nein	----						Schlecht	Transitioned to error state for -> Unidentified Error. Box-Reset.

HUMAX	Musik	AC3-DD+	640	CBR	48	5.1	HD	HDMI	Nein	----						Schlecht	Transitioned to error state for -> Unidentified Error. Box-Reset.
								SPDIF	Nein	----						Schlecht	Transitioned to error state for -> Unidentified Error. Box-Reset.

WAV zu AAC Encodierung mit NeroAAC. WAV zu AC3 Encodierung mit Aften. WAV zu DD+ Encodierung mit TotalCode Studio																		
Audio Only MP4									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis		
HUMAX	Channelldentification	AAC-LC	448	Pseudo CBR	48	5.1	HD	HDMI	Ja	----	falsch	Nein	Ja	Zu Leise -30dB	Gut	Schlecht	5.1 -> 2.0, ohne LFE
								SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0, ohne LFE

HUMAX	Channelldentification	AAC-HE	448	Pseudo CBR	48	5.1	HD	HDMI	Ja	----	falsch	Nein	Ja	Zu Leise -30dB	Gut	Schlecht	5.1 -> 2.0, ohne LFE
								SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0, ohne LFE

HUMAX	Channelldentification	AC3	448	CBR	48	5.1	HD	HDMI	Ja	----	Richtig	Ja	Ja	Gleich	Gut	Gut	
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	

HUMAX	Channelldentification	AC3-DD+	448	CBR	48	5.1	HD	HDMI	Ja	----	Richtig	Ja	Ja	Gleich	Gut	Gut	DD+ pass through.
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	AC3 Transcoding mit 640kbps.

Siehe AAC, AC3 und DD+																		
Audio Only MP4									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis		
HUMAX	Dolby TestMaterial	AC3-DD+	192	CBR	48	5.1	SD	HDMI	Ja	----	Richtig	Ja	Ja	Gleich	Gut	Gut	DD+ kommt beim Receiver an.
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	transcodiert in AC3 640kbps.

HUMAX	Dolby TestMaterial	AC3-DD+	320	CBR	48	7.1	SD	HDMI	Ja	----	Richtig	Ja	Ja	Gleich	Gut	Gut	DD+ pass through in 7.1.
								SPDIF	Ja	AC3	falsch	Ja	Ja	Gleich	Gut	Gut	transcodiert in AC3, 5.1, 640kbps.

DMG									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis		
HUMAX	Channelldentification	AC3-DD+	448	CBR	48	5.1	HD	HDMI	Ja	----	Richtig	Ja	Ja	Gleich	Gut	Gut	DD+ pass through.
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	AC3 Transcoding mit 640kbps.

HUMAX	Eishockey	AC3-DD+	448	CBR	48	5.1	HD	HDMI	Ja	----	Richtig	Ja	Ja	Gleich	Gut	Gut	DD+ pass through.
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	AC3 Transcoding mit 640kbps.

HUMAX	Musik	AC3-DD+	448	CBR	48	5.1	HD	HDMI	Ja	----	Richtig	Ja	Ja	Gleich	Gut	Gut	DD+ pass through.
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	AC3 Transcoding mit 640kbps.

								HDMI	Ja	----	Richtig	Ja	Ja	Gleich	Gut	Gut	DD+ pass through.

HUMAX	Kroemer	AC3-DD+	448	CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	AC3 Transcoding mit 640kbps.
								HDMI	Ja	---	Richtig	Ja	Ja	Gleich	Gut	Gut	DD+ pass through.
HUMAX	LaTraviata	AC3-DD+	448	CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	AC3 Transcoding mit 640kbps.
								HDMI	Ja	---	Richtig	Ja	Ja	Gleich	Gut	Gut	DD+ pass through.
HUMAX	Aida	AC3-DD+	448	CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	AC3 Transcoding mit 640kbps.

	Legende:	**Erkläung:**
	N.G.	Nicht Getestet, da sich die einzelnen Bitraten gleich verhalten.
	N.V.	Nicht vorhanden, da TotalCode Studio kein 44kHz DD+ erzeugen kann.

	Set-Top-Box Details:	
	Gerätebezeichnung:	HUMAX iCord HD+
	Software Version:	GHSNA 1.02.12
	Updtae Datum:	30 MAR 2012
	MICOM-Version:	Jul 72
	System_ID:	709E.5100
	CI+-Version:	1.00.00

Eigenschaften									Ergebnis								
Decoder	Testname	Codec	Bitrate	CBR/VBR	Frequenz	Audio Format	Video Format	Kabel	Interoperabilität	DM 100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	Erklärung
AAC-LC									**AAC_LC**								
TechniSat	LaTraviata	AAC-LC	128	Pseudo CBR	48	5.1	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	Musik	AAC-LC	192	Pseudo CBR	48	5.1	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	Eishockey	AAC-LC	256	Pseudo CBR	48	5.1	HD	HDMI	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								TOSLINK	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
TechniSat	Musik	AAC-LC	320	Pseudo CBR	48	5.1	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	Kroemer	AAC-LC	448	Pseudo CBR	48	5.1	HD	HDMI	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								TOSLINK	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
TechniSat	Channelldentification	AAC-LC	640	Pseudo CBR	48	5.1	HD	HDMI	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								TOSLINK	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
WAV to AAC-HE Encodierung mit NeroAAC. AAC und H264 in MP4 gemuxt mit FFmpeg.																	
AAC-HE									Interoperabilität	DM 100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
TechniSat	LaTraviata	AAC-HE	128	Pseudo CBR	48	5.1	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	Musik	AAC-HE	192	Pseudo CBR	48	5.1	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	Eishockey	AAC-HE	256	Pseudo CBR	48	5.1	HD	HDMI	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								TOSLINK	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
TechniSat	Musik	AAC-HE	320	Pseudo CBR	48	5.1	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	Kroemer	AAC-HE	448	Pseudo CBR	48	5.1	HD	HDMI	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								TOSLINK	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
TechniSat	Channelldentification	AAC-HE	640	Pseudo CBR	48	5.1	HD	HDMI	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								TOSLINK	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
WAV zu AAC-HE Encodierung mit NeroAAC. AAC und H264 in MP4 gemuxt mit FFmpeg																	
Dolby-AC3									Interoperabilität	DM 100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
TechniSat	LaTraviata	AC3	128	CBR	48	5.1	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	Musik	AC3	192	CBR	48	5.1	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	Eishockey	AC3	256	CBR	48	5.1	HD	HDMI	Ja	-----	Richtig	Ja	Ja	Gleich	Akzeptabel	Gut	Tonversatz aufgrund des Files nicht erkennbar. Receiver empfängt AC3.
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Akzeptabel	Gut	Tonversatz aufgrund des Files nicht erkennbar. Receiver empfängt AC3, 256kbps.
								TOSLINK	Ja	-----	Richtig	Ja	Ja	Gleich	Akzeptabel	Gut	Tonversatz aufgrund des Files nicht erkennbar. Receiver empfängt AC3.
TechniSat	Musik	AC3	320	CBR	48	5.1	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	Kroemer	AC3	448	CBR	48	5.1	HD	HDMI	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Akzeptabel	Ton leicht zu früh. Versatz lässt sich durch Gerät ausgleichen, ca +50ms. Receiver empfängt AC3.
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Akzeptabel	Ton leicht zu früh. Versatz lässt sich durch Gerät ausgleichen, ca +50ms. Receiver empfängt AC3, 448kbps.
								TOSLINK	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Akzeptabel	Ton leicht zu früh. Versatz lässt sich durch Gerät ausgleichen, ca +50ms. Receiver empfängt AC3.
TechniSat	Channelldentification	AC3	640	CBR	48	5.1	HD	HDMI	Ja	-----	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	Ton leicht zu früh. Versatz lässt sich durch Gerät ausgleichen, ca +50ms. Receiver empfängt AC3.
								SPDIF	Ja	AC3	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	Ton leicht zu früh. Versatz lässt sich durch Gerät ausgleichen, ca +50ms. Receiver empfängt AC3, 640kbps.
								TOSLINK	Ja	-----	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	Ton leicht zu früh. Versatz lässt sich durch Gerät ausgleichen, ca +50ms. Receiver empfängt AC3.
WAV zu AC3 Encodierung mit Aften. AC3 und H264 in MP4 gemuxt mit FFmpeg.																	
Dolby-DD+									Interoperabilität	DM 100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
TechniSat	LaTraviata	AC3-DD+	192	CBR	48	5.1	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	Eishockey	AC3-DD+	256	CBR	48	5.1	HD	HDMI	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Receiver empfängt DD+.
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Transcodierung in AC3 640kbps.
								TOSLINK	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Receiver empfängt AC3.
TechniSat	Musik	AC3-DD+	320	CBR	48	5.1	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	Kroemer	AC3-DD+	448	CBR	48	5.1	HD	HDMI	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Receiver empfängt DD+.
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Transcodierung in AC3 640kbps.
								TOSLINK	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Receiver empfängt AC3.
TechniSat	Channelldentification	AC3-DD+	640	CBR	48	5.1	HD	HDMI	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Receiver empfängt DD+.
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Transcodierung in AC3 640kbps.
								TOSLINK	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Receiver empfängt AC3.
TechniSat	Eishockey	AC3-DD+	1024	CBR	48	5.1	HD	HDMI	Nein	-----	Richtig	Ja	Ja	Gleich	Gut	Schlecht	Kein Video. Receiver empfängt DD+.
								SPDIF	Nein	AC3	Richtig	Ja	Ja	Gleich	Gut	Schlecht	Kein Video. Transcodierung in AC3 640kbps.
								TOSLINK	Nein	-----	Richtig	Ja	Ja	Gleich	Gut	Schlecht	Kein Video. Receiver empfängt AC3.
TechniSat	Eishockey	AC3-DD+	3024	CBR	48	5.1	HD	HDMI	Nein	-----	Richtig	Ja	Ja	Gleich	Gut	Schlecht	Kein Video. Receiver empfängt DD+.
								SPDIF	Nein	AC3	Richtig	Ja	Ja	Gleich	Gut	Schlecht	Kein Video. Transcodierung in AC3 640kbps.
								TOSLINK	Nein	-----	Richtig	Ja	Ja	Gleich	Gut	Schlecht	Kein Video. Receiver empfängt AC3.
WAV zu DD+ Encodierung, YUV zu H264 Encodierung und DD+ und H264 Muxing mit TotalCode Studio																	
44.1 kHz									Interoperabilität	DM 100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
TechniSat		AAC-LC	128	Pseudo CBR	44.1	5.1	HD	HDMI									N.G.
								SPDIF									N.G.

								TOSLINK										N.G.
TechniSat	Channelldentification	AAC-LC	640	Pseudo CBR	44.1	5.1	HD	HDMI	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								TOSLINK	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
TechniSat		AAC-HE	128	Pseudo CBR	44.1	5.1	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	Channelldentification	AAC-HE	640	Pseudo CBR	44.1	5.1	HD	HDMI	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								TOSLINK	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
TechniSat		AC3	128	CBR	44.1	5.1	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	Channelldentification	AC3	640	CBR	44.1	5.1	HD	HDMI	Ja	-----	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	Ton leicht zu früh. Versatz lässt sich durch Gerät ausgleichen, ca +50ms. Receiver empfängt AC3.
								SPDIF	Ja	AC3	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	Ton leicht zu früh. Versatz lässt sich durch Gerät ausgleichen, ca +50ms. Receiver empfängt AC3, 640kbps.
								TOSLINK	Ja	-----	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	Ton leicht zu früh. Versatz lässt sich durch Gerät ausgleichen, ca +50ms. Receiver empfängt AC3.
TechniSat		AC3-DD+	128	CBR	44.1	5.1	HD	HDMI									N.V.
								SPDIF									N.V.
								TOSLINK									N.V.
TechniSat	Channelldentification	AC3-DD+	640	CBR	44.1	5.1	HD	HDMI									N.V.
								SPDIF									N.V.
								TOSLINK									N.V.
				Siehe AAC, AC3 und DD+					N.V.								
				VBR					Interoperabilität	DM 100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
TechniSat		AAC-LC	128	VBR	48	5.1	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	Eishockey	AAC-LC	640	VBR	48	5.1	HD	HDMI	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								TOSLINK	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
TechniSat		AAC-HE	128	VBR	48	5.1	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	N.G.
TechniSat	Eishockey	AAC-HE	640	VBR	48	5.1	HD	HDMI	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								SPDIF	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								TOSLINK									
				Siehe AAC													
				STEREO					Interoperabilität	DM 100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
TechniSat		AAC-LC	128	Pseudo CBR	48	2.0	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	Musik	AAC-LC	640	Pseudo CBR	48	2.0	HD	HDMI	Ja	-----	Richtig	Nein	Ja	Gleich	Gut	Gut	
								SPDIF	Ja	PCM	Richtig	Nein	Ja	Gleich	Gut	Gut	
								TOSLINK	Ja	-----	Richtig	Nein	Ja	Gleich	Gut	Gut	
TechniSat		AAC-HE	128	Pseudo CBR	48	2.0	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	Musik	AAC-HE	640	Pseudo CBR	48	2.0	HD	HDMI	Ja	-----	Richtig	Nein	Ja	Gleich	Gut	Gut	
								SPDIF	Ja	PCM	Richtig	Nein	Ja	Gleich	Gut	Gut	
								TOSLINK	Ja	-----	Richtig	Nein	Ja	Gleich	Gut	Gut	
TechniSat		AC3	128	CBR	48	2.0	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	Musik	AC3	640	CBR	48	2.0	HD	HDMI	Ja	-----	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	Ton leicht zu früh. Versatz lässt sich durch Gerät ausgleichen, ca +50ms. Receiver empfängt AC3.
								SPDIF	Ja	AC3	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	Ton leicht zu früh. Versatz lässt sich durch Gerät ausgleichen, ca +50ms. Receiver empfängt AC3, 640kbps.
								TOSLINK	Ja	-----	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	Ton leicht zu früh. Versatz lässt sich durch Gerät ausgleichen, ca +50ms. Receiver empfängt AC3.
TechniSat		AC3-DD+	128	CBR	48	2.0	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	Musik	AC3-DD+	640	CBR	48	2.0	HD	HDMI	Ja	-----	Richtig	Nein	Ja	Gleich	Gut	Gut	Receiver empfängt DD+.
								SPDIF	Ja	PCM	Richtig	Nein	Ja	Gleich	Gut	Gut	Transcodierung in AC3 640kbps.
								TOSLINK	Ja	-----	Richtig	Nein	Ja	Gleich	Gut	Gut	Receiver empfängt AC3.
				Siehe AAC, AC3 und DD+													
				SD					Interoperabilität	DM 100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
TechniSat		AAC-LC	128	Pseudo CBR	48	5.1	SD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	LaTraviata	AAC-LC	640	Pseudo CBR	48	5.1	SD	HDMI	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								TOSLINK	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
TechniSat		AAC-HE	128	Pseudo CBR	48	5.1	SD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	LaTraviata	AAC-HE	640	Pseudo CBR	48	5.1	SD	HDMI	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								TOSLINK	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
TechniSat		AC3	128	CBR	48	5.1	SD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	LaTraviata	AC3	640	CBR	48	5.1	SD	HDMI	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Tonversatz aufgrund des Files nicht erkennbar. Receiver empfängt AC3.
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Tonversatz aufgrund des Files nicht erkennbar. Receiver empfängt AC3, 640kbps.
								TOSLINK	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Tonversatz aufgrund des Files nicht erkennbar. Receiver empfängt AC3.
TechniSat		AC3-DD+	192	CBR	48	5.1	SD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	LaTraviata	AC3-DD+	640	CBR	48	5.1	SD	HDMI	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Receiver empfängt DD+.
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Transcodierung in AC3 640kbps.
								TOSLINK	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Receiver empfängt AC3.
				Siehe AAC, AC3 und DD+													
				TS					Interoperabilität	DM 100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
								HDMI	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.

								TOSLINK										N.G.
TechniSat	Channelldentification	AAC-LC	640	Pseudo CBR	44.1	5.1	HD	HDMI	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								TOSLINK	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
TechniSat		AAC-HE	128	Pseudo CBR	44.1	5.1	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	Channelldentification	AAC-HE	640	Pseudo CBR	44.1	5.1	HD	HDMI	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								TOSLINK	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
TechniSat		AC3	128	CBR	44.1	5.1	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	Channelldentification	AC3	640	CBR	44.1	5.1	HD	HDMI	Ja	-----	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	Ton leicht zu früh. Versatz lässt sich durch Gerät ausgleichen, ca +50ms. Receiver empfängt AC3.
								SPDIF	Ja	AC3	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	Ton leicht zu früh. Versatz lässt sich durch Gerät ausgleichen, ca +50ms. Receiver empfängt AC3, 640kbps.
								TOSLINK	Ja	-----	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	Ton leicht zu früh. Versatz lässt sich durch Gerät ausgleichen, ca +50ms. Receiver empfängt AC3.
TechniSat		AC3-DD+	128	CBR	44.1	5.1	HD	HDMI									N.V.
								SPDIF									N.V.
								TOSLINK									N.V.
TechniSat	Channelldentification	AC3-DD+	640	CBR	44.1	5.1	HD	HDMI									N.V.
								SPDIF									N.V.
								TOSLINK									N.V.
				Siehe AAC, AC3 und DD+					N.V.								
				VBR					Interoperabilität	DM 100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
TechniSat		AAC-LC	128	VBR	48	5.1	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	Eishockey	AAC-LC	640	VBR	48	5.1	HD	HDMI	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								TOSLINK	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
TechniSat		AAC-HE	128	VBR	48	5.1	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	N.G.
TechniSat	Eishockey	AAC-HE	640	VBR	48	5.1	HD	HDMI	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								SPDIF	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								TOSLINK									
				Siehe AAC													
				STEREO					Interoperabilität	DM 100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
TechniSat		AAC-LC	128	Pseudo CBR	48	2.0	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	Musik	AAC-LC	640	Pseudo CBR	48	2.0	HD	HDMI	Ja	-----	Richtig	Nein	Ja	Gleich	Gut	Gut	
								SPDIF	Ja	PCM	Richtig	Nein	Ja	Gleich	Gut	Gut	
								TOSLINK	Ja	-----	Richtig	Nein	Ja	Gleich	Gut	Gut	
TechniSat		AAC-HE	128	Pseudo CBR	48	2.0	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	Musik	AAC-HE	640	Pseudo CBR	48	2.0	HD	HDMI	Ja	-----	Richtig	Nein	Ja	Gleich	Gut	Gut	
								SPDIF	Ja	PCM	Richtig	Nein	Ja	Gleich	Gut	Gut	
								TOSLINK	Ja	-----	Richtig	Nein	Ja	Gleich	Gut	Gut	
TechniSat		AC3	128	CBR	48	2.0	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	Musik	AC3	640	CBR	48	2.0	HD	HDMI	Ja	-----	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	Ton leicht zu früh. Versatz lässt sich durch Gerät ausgleichen, ca +50ms. Receiver empfängt AC3.
								SPDIF	Ja	AC3	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	Ton leicht zu früh. Versatz lässt sich durch Gerät ausgleichen, ca +50ms. Receiver empfängt AC3, 640kbps.
								TOSLINK	Ja	-----	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	Ton leicht zu früh. Versatz lässt sich durch Gerät ausgleichen, ca +50ms. Receiver empfängt AC3.
TechniSat		AC3-DD+	128	CBR	48	2.0	HD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	Musik	AC3-DD+	640	CBR	48	2.0	HD	HDMI	Ja	-----	Richtig	Nein	Ja	Gleich	Gut	Gut	Receiver empfängt DD+.
								SPDIF	Ja	PCM	Richtig	Nein	Ja	Gleich	Gut	Gut	Transcodierung in AC3 640kbps.
								TOSLINK	Ja	-----	Richtig	Nein	Ja	Gleich	Gut	Gut	Receiver empfängt AC3.
				Siehe AAC, AC3 und DD+													
				SD					Interoperabilität	DM 100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
TechniSat		AAC-LC	128	Pseudo CBR	48	5.1	SD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	LaTraviata	AAC-LC	640	Pseudo CBR	48	5.1	SD	HDMI	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								TOSLINK	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
TechniSat		AAC-HE	128	Pseudo CBR	48	5.1	SD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	LaTraviata	AAC-HE	640	Pseudo CBR	48	5.1	SD	HDMI	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
								TOSLINK	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.
TechniSat		AC3	128	CBR	48	5.1	SD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	LaTraviata	AC3	640	CBR	48	5.1	SD	HDMI	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Tonversatz aufgrund des Files nicht erkennbar. Receiver empfängt AC3.
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Tonversatz aufgrund des Files nicht erkennbar. Receiver empfängt AC3, 640kbps.
								TOSLINK	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Tonversatz aufgrund des Files nicht erkennbar. Receiver empfängt AC3.
TechniSat		AC3-DD+	192	CBR	48	5.1	SD	HDMI									N.G.
								SPDIF									N.G.
								TOSLINK									N.G.
TechniSat	LaTraviata	AC3-DD+	640	CBR	48	5.1	SD	HDMI	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Receiver empfängt DD+.
								SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Transcodierung in AC3 640kbps.
								TOSLINK	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Receiver empfängt AC3.
				Siehe AAC, AC3 und DD+													
				TS					Interoperabilität	DM 100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
								HDMI	Ja	-----	falsch	Nein	Ja	Gleich	Gut	Schlecht	LS auf L und RS auf R gefaltet.

Eigenschaften									Ergebnis								
Decoder	Testname	Codec	Bitrate	CBR/VBR	Frequenz	Audio Format	Video Format	Kabel	Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	Erklärung
AAC-LC									**AAC_LC**								
ITT	LaTraviata	AAC-LC	128	Pseudo CBR	48	5.1	HD	SPDIF									N.G.
ITT	Musik	AAC-LC	192	Pseudo CBR	48	5.1	HD	SPDIF									N.G.
ITT	Eishockey	AAC-LC	256	Pseudo CBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 Downmix.
ITT	Musik	AAC-LC	320	Pseudo CBR	48	5.1	HD	SPDIF									N.G.
ITT	Kroemer	AAC-LC	448	Pseudo CBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 Downmix.
ITT	ChannelIdentification	AAC-LC	640	Pseudo CBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 Downmix.
WAV to AAC-HE Encodierung mit NeroAAC. AAC und H264 in MP4 gemuxt mit FFmpeg.																	
AAC-HE									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
ITT	LaTraviata	AAC-HE	128	Pseudo CBR	48	5.1	HD	SPDIF									N.G.
ITT	Musik	AAC-HE	192	Pseudo CBR	48	5.1	HD	SPDIF									N.G.
ITT	Eishockey	AAC-HE	256	Pseudo CBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 Downmix.
ITT	Musik	AAC-HE	320	Pseudo CBR	48	5.1	HD	SPDIF									N.G.
ITT	Kroemer	AAC-HE	448	Pseudo CBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 Downmix.
ITT	ChannelIdentification	AAC-HE	640	Pseudo CBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 Downmix.
WAV zu AAC-HE Encodierung mit NeroAAC. AAC und H264 in MP4 gemuxt mit FFmpeg.																	
Dolby-AC3									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
ITT	LaTraviata	AC3	128	CBR	48	5.1	HD	SPDIF									N.G.
ITT	Musik	AC3	192	CBR	48	5.1	HD	SPDIF									N.G.
ITT	Eishockey	AC3	256	CBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 Downmix. LS+RS auf Front Mitte zusammengelegt.
ITT	Musik	AC3	320	CBR	48	5.1	HD	SPDIF									N.G.
ITT	Kroemer	AC3	448	CBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 Downmix. LS+RS auf Front Mitte zusammengelegt.
ITT	ChannelIdentification	AC3	640	CBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 Downmix. LS+RS auf Front Mitte zusammengelegt.
WAV zu AC3 Encodierung mit Aften. AC3 und H264 in MP4 gemuxt mit FFmpeg.																	
Dolby-DD+									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
ITT	LaTraviata	AC3-DD+	192	CBR	48	5.1	HD	SPDIF									N.G.
ITT	Eishockey	AC3-DD+	256	CBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	5.1 -> 2.0 Downmix.
ITT	Musik	AC3-DD+	320	CBR	48	5.1	HD	SPDIF									N.G.
ITT	Kroemer	AC3-DD+	448	CBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	5.1 -> 2.0 Downmix.
ITT	ChannelIdentification	AC3-DD+	640	CBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	5.1 -> 2.0 Downmix.
ITT	Eishockey	AC3-DD+	1024	CBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	5.1 -> 2.0 Downmix.
ITT	Eishockey	AC3-DD+	3024	CBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	5.1 -> 2.0 Downmix.
WAV zu DD+ Encodierung, YUV zu H264 Encodierung und DD+ und H264 Muxing mit TotalCode Studio																	
44.1 kHz									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
ITT		AAC-LC	128	Pseudo CBR	44.1	5.1	HD	SPDIF									N.G.

ITT	ChannelIdentification	AAC-LC	640	Pseudo CBR	44.1	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 Downmix.	
ITT		AAC-HE	128	Pseudo CBR	44.1	5.1	HD	SPDIF									N.G.	
ITT	ChannelIdentification	AAC-HE	640	Pseudo CBR	44.1	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 Downmix.	
ITT		AC3	128	CBR	44.1	5.1	HD	SPDIF									N.G.	
ITT	ChannelIdentification	AC3	640	CBR	44.1	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 Downmix. LS+RS auf Front Mitte zusammengelegt.	
ITT		AC3-DD+	128	CBR	44.1	5.1	HD	SPDIF									N.V.	
HUMAX	ChannelIdentification	AC3-DD+	640	CBR	44.1	5.1	HD	SPDIF									N.V.	
Siehe AAC, AC3 und DD+																		
VBR									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis		
ITT		AAC-LC	128	VBR	48	5.1	HD	SPDIF									N.G.	
ITT	Eishockey	AAC-LC	640	VBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 Downmix.	
ITT		AAC-HE	128	VBR	48	5.1	HD	SPDIF									N.G.	
ITT	Eishockey	AAC-HE	640	VBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 Downmix.	
Siehe AAC																		
STEREO									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis		
ITT		AAC-LC	128	Pseudo CBR	48	2.0	HD	SPDIF									N.G.	
ITT	Musik	AAC-LC	640	Pseudo CBR	48	2.0	HD	SPDIF	Ja	PCM	Richtig	Nein	Ja	Leiser	Gut	Gut		
ITT		AAC-HE	128	Pseudo CBR	48	2.0	HD	SPDIF									N.G.	
ITT	Musik	AAC-HE	640	Pseudo CBR	48	2.0	HD	SPDIF	Ja	PCM	Richtig	Nein	Ja	Leiser	Gut	Gut		
ITT		AC3	128	CBR	48	2.0	HD	SPDIF									N.G.	
ITT	Musik	AC3	640	CBR	48	2.0	HD	SPDIF	Ja	PCM	Richtig	Nein	Ja	Leiser	Gut	Gut		
ITT		AC3-DD+	128	CBR	48	2.0	HD	SPDIF									N.G.	
ITT	Musik	AC3-DD+	640	CBR	48	2.0	HD	SPDIF	Ja	PCM	Richtig	Nein	Ja	Gleich	Gut	Gut		
Siehe AAC, AC3 und DD+																		
SD									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis		
ITT		AAC-LC	128	Pseudo CBR	48	5.1	SD	SPDIF									N.G.	
ITT	LaTraviata	AAC-LC	640	Pseudo CBR	48	5.1	SD	SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 Downmix.	
ITT		AAC-HE	128	Pseudo CBR	48	5.1	SD	SPDIF									N.G.	
ITT	LaTraviata	AAC-HE	640	Pseudo CBR	48	5.1	SD	SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 Downmix.	
ITT		AC3	128	CBR	48	5.1	SD	SPDIF									N.G.	
ITT	LaTraviata	AC3	640	CBR	48	5.1	SD	SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 Downmix.	
ITT		AC3-DD+	192	CBR	48	5.1	SD	SPDIF									N.G.	
ITT	LaTraviata	AC3-DD+	640	CBR	48	5.1	SD	SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	5.1 -> 2.0 Downmix.	
Siehe AAC, AC3 und DD+																		
TS									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis		
ITT	ChannelIdentification	AAC-LC	448	Pseudo CBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 Downmix.	
ITT	ChannelIdentification	AAC-HE	448	Pseudo CBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 Downmix.	

ITT	Channel Identification	AC3	448	CBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	5.1 -> 2.0 Downmix. LS+RS auf Front Mitte zusammengelegt.
Alle .ts mit TotalCodeStudio erstellt, um DVB Konformität zu gewährleisten.																
Audio Only Elementary									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
ITT	Musik	AAC-LC	640	Pseudo CBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 Downmix.
ITT	Musik	AAC-HE	640	Pseudo CBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 Downmix.
ITT	Musik	AC3	640	CBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 Downmix.
ITT	Musik	AC3-DD+	640	CBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	5.1 -> 2.0 Downmix.
WAV zu AAC Encodierung mit NeroAAC. WAV zu AC3 Encodierung mit Aften. WAV zu DD+ Encodierung mit TotalCode Studio																
Audio Only MP4									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
ITT	Channel Identification	AAC-LC	448	Pseudo CBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 Downmix.
ITT	Channel Identification	AAC-HE	448	Pseudo CBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 Downmix.
ITT	Channel Identification	AC3	448	CBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 Downmix. LS+RS auf Front Mitte zusammengelegt.
ITT	Channel Identification	AC3-DD+	448	CBR	48	5.1	HD	SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	5.1 -> 2.0 Downmix.
Siehe AAC, AC3 und DD+																
Dolby Testmaterial									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
ITT	Dolby TestMaterial	AC3-DD+	192	CBR	48	5.1	SD	SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	5.1 -> 2.0 Downmix.
ITT	Dolby TestMaterial	AC3-DD+	320	CBR	48	7.1	SD	SPDIF	Ja	PCM	falsch	Nein	Ja	Gleich	Gut	Schlecht	7.1 -> 2.0 Downmix. Bitrate: 384kbps.
TotalCode Stuio DD+									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
ITT	Channel Identification	AC3-DD+	448	CBR	48	5.1	HD	SPDIF	Nein	PCM						Schlecht	Video wird abgespielt, aber kein Audio. Denon zeigt jedoch PCM als Input an.
ITT	Eishockey	AC3-DD+	448	CBR	48	5.1	HD	SPDIF	Nein	PCM						Schlecht	Video wird abgespielt, aber kein Audio. Denon zeigt jedoch PCM als Input an.
ITT	Musik	AC3-DD+	448	CBR	48	5.1	HD	SPDIF	Nein	PCM						Schlecht	Video wird abgespielt, aber kein Audio. Denon zeigt jedoch PCM als Input an.
ITT	Kroemer	AC3-DD+	448	CBR	48	5.1	HD	SPDIF	Nein	PCM						Schlecht	Video wird abgespielt, aber kein Audio. Denon zeigt jedoch PCM als Input an.
ITT	LaTraviata	AC3-DD+	448	CBR	48	5.1	HD	SPDIF	Nein	PCM						Schlecht	Video wird abgespielt, aber kein Audio. Denon zeigt jedoch PCM als Input an.
ITT	Aida	AC3-DD+	448	CBR	48	5.1	HD	SPDIF	Nein	PCM						Schlecht	Video wird abgespielt, aber kein Audio. Denon zeigt jedoch PCM als Input an.

Legende: **Erklärung:**
N.G. Nicht Getestet, da sich die einzelnen Bitraten gleich verhalten.
N.V. Nicht vorhanden, da TotalCode Studio kein 44kHz DD+ erzeugen kann.
K.Q. Klangqualitätstest durchgeführt mit LaTraviata.

HbbTV-Gerät Details:
Laut Manual:
Input Froemat: Output optisch
AC3 -> AC3
DD+ -> AC3
MPEG -> PCM
AAC-HE -> AC3

Gerät Infos:
Name: ITT LCD 42-3475

Eigenschaften									Ergebnis								
Decoder	Testname	Codec	Bitrate	CBR/VBR	Frequenz	Audio Format	Video Format	Kabel	Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	Erklärung
		AAC-LC											**AAC_LC**				
Toshiba	La Traviata	AAC-LC	128	Pseudo CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Nein	Gleich	Schlecht	Schlecht	AAC-LC zu AC3 640kbps transcodiert. Ton zu spät, ca 2 Frames versatz. Tonversatz kann manuell durch das Gerät nicht ausgeglichen werden.
Toshiba	Musik	AAC-LC	192	Pseudo CBR	48	5.1	HD	SPDIF									N.G.
Toshiba	Eishockey	AAC-LC	256	Pseudo CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	AAC-LC zu AC3 640kbps transcodiert. Tonversatz aufgrund des Testfiles nicht erkennbar.
Toshiba	Musik	AAC-LC	320	Pseudo CBR	48	5.1	HD	SPDIF									N.G.
Toshiba	Kroemer	AAC-LC	448	Pseudo CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	AAC-LC zu AC3 640kbps transcodiert. Ton zu spät, ca 2 Frames versatz. Tonversatz kann manuell durch das Gerät nicht ausgeglichen werden.
Toshiba	ChannelIdentification	AAC-LC	640	Pseudo CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	AAC-LC zu AC3 640kbps transcodiert. Ton zu spät, ca 2 Frames versatz. Tonversatz kann manuell durch das Gerät nicht ausgeglichen werden.
	WAV to AAC-HE Encodierung mit NeroAAC. AAC und H264 in MP4 gemuxt mit FFmpeg.																
		AAC-HE							Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
Toshiba	La Traviata	AAC-HE	128	Pseudo CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Nein	Gleich	Schlecht	Schlecht	AAC-HE zu AC3 640kbps transcodiert. Ton zu spät, ca 2 Frames versatz. Tonversatz kann manuell durch das Gerät nicht ausgeglichen werden.
Toshiba	Musik	AAC-HE	192	Pseudo CBR	48	5.1	HD	SPDIF									N.G.
Toshiba	Eishockey	AAC-HE	256	Pseudo CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	AAC-HE zu AC3 640kbps transcodiert. Tonversatz aufgrund des Testfiles nicht erkennbar.
Toshiba	Musik	AAC-HE	320	Pseudo CBR	48	5.1	HD	SPDIF									N.G.
Toshiba	Kroemer	AAC-HE	448	Pseudo CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	AAC-HE zu AC3 640kbps transcodiert. Ton zu spät, ca 2 Frames versatz. Tonversatz kann manuell durch das Gerät nicht ausgeglichen werden.
Toshiba	ChannelIdentification	AAC-HE	640	Pseudo CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	AAC-HE zu AC3 640kbps transcodiert. Ton zu spät, ca 2 Frames versatz. Tonversatz kann manuell durch das Gerät nicht ausgeglichen werden.
	WAV zu AAC-HE Encodierung mit NeroAAC. AAC und H264 in MP4 gemuxt mit FFmpeg.																
		Dolby-AC3							Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
Toshiba	La Traviata	AC3	128	CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Nein	Gleich	Schlecht	Schlecht	Ton zu früh, ca 2 Frames versatz. Tonversatz kann manuell durch das Gerät ausgeglichen werden, ca +60ms.
Toshiba	Musik	AC3	192	CBR	48	5.1	HD	SPDIF									N.G.
Toshiba	Eishockey	AC3	256	CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Akzeptabel	Gut	Tonversatz aufgrund des Testfiles nicht erkennbar.
Toshiba	Musik	AC3	320	CBR	48	5.1	HD	SPDIF									N.G.
Toshiba	Kroemer	AC3	448	CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	Ton zu früh, ca 2 Frames versatz. Tonversatz kann manuell durch das Gerät ausgeglichen werden, ca +60ms.
Toshiba	ChannelIdentification	AC3	640	CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	Ton zu früh, ca 2 Frames versatz. Tonversatz kann manuell durch das Gerät ausgeglichen werden, ca +60ms.
	WAV zu AC3 Encodierung mit Alten. AC3 und H264 in MP4 gemuxt mit FFmpeg.																
		Dolby-DD+							Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
Toshiba	La Traviata	AC3-DD+	192	CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Zu AC3 640kbps transcodiert.
Toshiba	Eishockey	AC3-DD+	256	CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Zu AC3 640kbps transcodiert.
Toshiba	Musik	AC3-DD+	320	CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Zu AC3 640kbps transcodiert.
Toshiba	Kroemer	AC3-DD+	448	CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Zu AC3 640kbps transcodiert.
Toshiba	ChannelIdentification	AC3-DD+	640	CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Zu AC3 640kbps transcodiert.
Toshiba	Eishockey	AC3-DD+	1024	CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Zu AC3 640kbps transcodiert.
Toshiba	Eishockey	AC3-DD+	3024	CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Zu AC3 640kbps transcodiert.
	WAV zu DD+ Encodierung, YUV zu H264 Encodierung und DD+ und H264 Muxing mit TotalCode Studio																
		44.1 kHz							Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
Toshiba		AAC-LC	128	Pseudo CBR	44.1	5.1	HD	SPDIF									
																	N.G.
Toshiba	ChannelIdentification	AAC-LC	640	Pseudo CBR	44.1	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	AAC-LC zu AC3 640kbps, 48kHz transcodiert. Ton zu früh, ca 2 Frames Tonversatz. Tonversatz kann manuell durch das Gerät nicht ausgeglichen werden.
Toshiba		AAC-HE	128	Pseudo CBR	44.1	5.1	HD	SPDIF									N.G.
Toshiba	ChannelIdentification	AAC-HE	640	Pseudo CBR	44.1	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	AAC-HE zu AC3 640kbps, 48kHz transcodiert. Ton zu früh, ca 2 Frames Tonversatz. Tonversatz kann manuell durch das Gerät nicht ausgeglichen werden.

Toshiba		AC3	128	CBR	44.1	5.1	HD	SPDIF									N.G.
Toshiba	ChannelIdentification	AC3	640	CBR	44.1	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	44.1kHz Output. Ton zu spät, ca 2 Frames Tonversatz. Tonversatz kann manuell durch das Gerät ausgeglichen werden, ca +60ms.
Toshiba		AC3-DD+	128	CBR	44.1	5.1	HD	SPDIF									N.V.
Toshiba	ChannelIdentification	AC3-DD+	640	CBR	44.1	5.1	HD	SPDIF									N.V.
				Siehe AAC, AC3 und DD+													
				VBR					Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
Toshiba		AAC-LC	128	VBR	48	5.1	HD	SPDIF									N.G.
Toshiba	Eishockey	AAC-LC	640	VBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	AAC-LC zu AC3 640kbps, 48kHz transcodiert. Ton zu früh, ca 2 Frames Tonversatz. Tonversatz kann manuell durch das Gerät nicht ausgeglichen werden.
Toshiba		AAC-HE	128	VBR	48	5.1	HD	SPDIF									N.G.
Toshiba	Eishockey	AAC-HE	640	VBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	AAC-HE zu AC3 640kbps, 48kHz transcodiert. Ton zu früh, ca 2 Frames Tonversatz. Tonversatz kann manuell durch das Gerät nicht ausgeglichen werden.
				Siehe AAC													
				STEREO					Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
Toshiba		AAC-LC	128	Pseudo CBR	48	2.0	HD	SPDIF									N.G.
Toshiba	Musik	AAC-LC	640	Pseudo CBR	48	2.0	HD	SPDIF	Ja	AC3	Richtig	Nein	Nein	Gleich	Gut	Akzeptabel	AAC-LC zu AC3 384kbps, 48kHz transcodiert. Ton zu früh, ca 2 Frames Tonversatz. Tonversatz kann manuell durch das Gerät nicht ausgeglichen werden.
Toshiba		AAC-HE	128	Pseudo CBR	48	2.0	HD	SPDIF									N.G.
Toshiba	Musik	AAC-HE	640	Pseudo CBR	48	2.0	HD	SPDIF	Ja	AC3	Richtig	Nein	Nein	Gleich	Gut	Akzeptabel	AAC-HE zu AC3 384kbps, 48kHz transcodiert. Ton zu früh, ca 2 Frames Tonversatz. Tonversatz kann manuell durch das Gerät nicht ausgeglichen werden.
Toshiba		AC3	128	CBR	48	2.0	HD	SPDIF									N.G.
Toshiba	Musik	AC3	640	CBR	48	2.0	HD	SPDIF	Ja	AC3	Richtig	Nein	Nein	Gleich	Gut	Akzeptabel	Ton zu spät, ca 2 Frames Tonversatz. Tonversatz kann manuell durch das Gerät ausgeglichen werden, ca +60ms.
Toshiba		AC3-DD+	128	CBR	48	2.0	HD	SPDIF									N.G.
Toshiba	Musik	AC3-DD+	640	CBR	48	2.0	HD	SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Zu AC3 640kbps transcodiert.
				Siehe AAC, AC3 und DD+													
				SD					Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
Toshiba		AAC-LC	128	Pseudo CBR	48	5.1	SD	SPDIF									N.G.
Toshiba	La Traviata	AAC-LC	640	Pseudo CBR	48	5.1	SD	SPDIF	Ja	AC3	Richtig	Nein	Nein	Gleich	Gut	Akzeptabel	AAC-LC zu AC3 640kbps, 48kHz transcodiert. Ton zu früh, ca 2 Frames Tonversatz. Tonversatz kann manuell durch das Gerät nicht ausgeglichen werden.
Toshiba		AAC-HE	128	Pseudo CBR	48	5.1	SD	SPDIF									N.G.
Toshiba	La Traviata	AAC-HE	640	Pseudo CBR	48	5.1	SD	SPDIF	Ja	AC3	Richtig	Nein	Nein	Gleich	Gut	Akzeptabel	AAC-HE zu AC3 640kbps, 48kHz transcodiert. Ton zu früh, ca 2 Frames Tonversatz. Tonversatz kann manuell durch das Gerät nicht ausgeglichen werden.
Toshiba		AC3	128	CBR	48	5.1	SD	SPDIF									N.G.
Toshiba	La Traviata	AC3	640	CBR	48	5.1	SD	SPDIF	Ja	AC3	Richtig	Nein	Nein	Gleich	Gut	Akzeptabel	Ton zu spät, ca 2 Frames Tonversatz. Tonversatz kann manuell durch das Gerät ausgeglichen werden, ca +60ms.
Toshiba		AC3-DD+	192	CBR	48	5.1	SD	SPDIF									N.G.
Toshiba	La Traviata	AC3-DD+	640	CBR	48	5.1	SD	SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Zu AC3 640kbps transcodiert.
				Siehe AAC, AC3 und DD+													
				TS					Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
Toshiba	ChannelIdentification	AAC-LC	448	Pseudo CBR	48	5.1	HD	SPDIF	Nein							Schlecht	File wird nicht abgespielt.
Toshiba	ChannelIdentification	AAC-HE	448	Pseudo CBR	48	5.1	HD	SPDIF	Nein							Schlecht	File wird nicht abgespielt.
Toshiba	ChannelIdentification	AC3	448	CBR	48	5.1	HD	SPDIF	Nein							Schlecht	File wird nicht abgespielt. Error-Meldung -> A/V Format not supoorted.
				Alle .ts mit TotalCodeStudio erstellt, um DVB Konformität zu gewährleisten.													
				Audio Only Elementary					Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
Toshiba	Musik	AAC-LC	640	Pseudo CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	AAC-LC zu AC3 640kbps transcodiert. Tonversatz kann manuell durch das Gerät nicht ausgeglichen werden.
Toshiba	Musik	AAC-HE	640	Pseudo CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	AAC-HE zu AC3 640kbps transcodiert. Tonversatz kann manuell durch das Gerät nicht ausgeglichen werden.
Toshiba	Musik	AC3	640	CBR	48	5.1	HD	SPDIF	Nein							Schlecht	File wird nicht abgespielt.

Toshiba	Musik	AC3-DD+	640	CBR	48	5.1	HD	SPDIF	Nein								Schlecht	File wird nicht abgespielt.
WAV zu AAC Encodierung mit NeroAAC. WAV zu AC3 Encodierung mit Aften. WAV zu DD+ Encodierung mit TotalCode Studio																			
Audio Only MP4									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis		
Toshiba	ChannelIdentification	AAC-LC	448	Pseudo CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	AAC-LC zu AC3 640kbps transcodiert. Tonversatz kann manuell durch das Gerät nicht ausgeglichen werden.	
Toshiba	ChannelIdentification	AAC-HE	448	Pseudo CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Nein	Gleich	Gut	Akzeptabel	AAC-HE zu AC3 640kbps transcodiert. Tonversatz kann manuell durch das Gerät nicht ausgeglichen werden.	
Toshiba	ChannelIdentification	AC3	448	CBR	48	5.1	HD	SPDIF	Nein	AC3						Schlecht	File wird abgespielt, aber kein Audio. Zeit läuft dreifach zu schnell. DM100 -> Konstanter Datenstrom mit 640kbps.	
Toshiba	ChannelIdentification	AC3-DD+	448	CBR	48	5.1	HD	SPDIF	Nein	AC3						Schlecht	File wird abgespielt, aber kein Audio. Zeit läuft dreifach zu schnell. DM100 -> Konstanter Datenstrom mit 640kbps.	
Siehe AAC, AC3 und DD+																			
Dolby Testmaterial									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis		
Toshiba	Dolby TestMaterial	AC3-DD+	192	CBR	48	5.1	SD	SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Zu 640kbps transcodiert. Tonversatz auf Grund des Files nicht erkennbar.	
Toshiba	Dolby TestMaterial	AC3-DD+	320	CBR	48	7.1	SD	SPDIF	Ja	AC3	falsch	Ja	Ja	Gleich	Gut	Gut	Zu 640kbps transcodiert. Kanäle 6+7 nicht hörbar da AC3 nicht 7.1 fähig ist. Tonversatz aufgrund des Files nicht erkennbar.	
DMG									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis		
Toshiba	ChannelIdentification	AC3-DD+	448	CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Zu AC3 640kbps transcodiert.	
Toshiba	Eishockey	AC3-DD+	448	CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Zu AC3 640kbps transcodiert.	
Toshiba	Musik	AC3-DD+	448	CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Zu AC3 640kbps transcodiert.	
Toshiba	Kroemer	AC3-DD+	448	CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Zu AC3 640kbps transcodiert.	
Toshiba	La Traviata	AC3-DD+	448	CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Zu AC3 640kbps transcodiert.	
Toshiba	Aida	AC3-DD+	448	CBR	48	5.1	HD	SPDIF	Ja	AC3	Richtig	Ja	Ja	Gleich	Gut	Gut	Zu AC3 640kbps transcodiert.	

	Legende:	**Erklärung:**
	N.G.	Nicht Getestet, da sich die einzelnen Bitraten gleich verhalten.
	N.V.	Nicht vorhanden, da TotalCode Studio kein 44kHz DD+ erzeugen kann.
	K.Q.	Klangqualitätstest durchgeführt mit La Traviata.
	HbbTV-Gerät Details:	
	Laut Manual:	
	Input Froemat:	Output optisch
	AC3 ->	AC3
	DD+ ->	AC3
	MPEG ->	PCM
	AAC-HE ->	AC3
	Gerät Infos:	
	Name:	Toshiba REZA 2011 - 32SL883G

Eigenschaften									Ergebnis								
Decoder	Testname	Codec	Bitrate	CBR/VBR	Frequenz	Audio Format	Video Format	Kabel	Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	Erklärung
AAC-LC													**AAC_LC**				

Loewe	LaTraviata	AAC-LC	128	Pseudo CBR	48	5.1	HD	TOSLINK	Ja	------	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 | JavaScript Error in KeyEvent.

Loewe	Musik	AAC-LC	192	Pseudo CBR	48	5.1	HD	TOSLINK									N.G.

Loewe	Eishockey	AAC-LC	256	Pseudo CBR	48	5.1	HD	TOSLINK	Ja	------	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 | JavaScript Error in KeyEvent.

Loewe	Channelldentification	AAC-LC	320	Pseudo CBR	48	5.1	HD	TOSLINK	Ja	------	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 | JavaScript Error in KeyEvent.

Loewe	Kroemer	AAC-LC	448	Pseudo CBR	48	5.1	HD	TOSLINK	Ja	------	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 | JavaScript Error in KeyEvent.

Loewe	Musik	AAC-LC	640	Pseudo CBR	48	5.1	HD	TOSLINK									N.G.
WAV to AAC-HE Encodierung mit NeroAAC. AAC und H264 in MP4 gemuxt mit FFmpeg.																	
AAC-HE									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	

Loewe	LaTraviata	AAC-HE	128	Pseudo CBR	48	5.1	HD	TOSLINK	Ja	------	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 | JavaScript Error in KeyEvent.

Loewe	Musik	AAC-HE	192	Pseudo CBR	48	5.1	HD	TOSLINK									N.G.

Loewe	Eishockey	AAC-HE	256	Pseudo CBR	48	5.1	HD	TOSLINK	Ja	------	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 | JavaScript Error in KeyEvent.

Loewe	Channelldentification	AAC-HE	320	Pseudo CBR	48	5.1	HD	TOSLINK	Ja	------	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 | JavaScript Error in KeyEvent. | JavaScript Error in KeyEvent.

Loewe	Kroemer	AAC-HE	448	Pseudo CBR	48	5.1	HD	TOSLINK	Ja	------	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 | JavaScript Error in KeyEvent.

Loewe	Musik	AAC-HE	640	Pseudo CBR	48	5.1	HD	TOSLINK									N.G.
WAV zu AAC-HE Encodierung mit NeroAAC. AAC und H264 in MP4 gemuxt mit FFmpeg.																	
Dolby-AC3									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	

Loewe	LaTraviata	AC3	128	CBR	48	5.1	HD	TOSLINK	Ja	------	Richtig	Ja	Ja	Gleich	Gut	Gut	Denon zeigt als Input AC3. | JavaScript Error in KeyEvent.

Loewe	Musik	AC3	192	CBR	48	5.1	HD	TOSLINK									N.G.

Loewe	Eishockey	AC3	256	CBR	48	5.1	HD	TOSLINK	Nein	------	Richtig	Ja	Ja	Gleich	Gut	Schlecht	Denon zeigt als Input AC3. Video ruckelt leicht | JavaScript Error in KeyEvent.

Loewe	Channelldentification	AC3	320	CBR	48	5.1	HD	TOSLINK	Ja	------	Richtig	Ja	Ja	Gleich	Gut	Gut	Denon zeigt als Input AC3. | JavaScript Error in KeyEvent.

Loewe	Kroemer	AC3	448	CBR	48	5.1	HD	TOSLINK	Ja	------	Richtig	Ja	Ja	Gleich	Gut	Gut	Denon zeigt als Input AC3. | JavaScript Error in KeyEvent.

Loewe	Musik	AC3	640	CBR	48	5.1	HD	TOSLINK									N.G.
WAV zu AC3 Encodierung mit Aften. AC3 und H264 in MP4 gemuxt mit FFmpeg.																	
Dolby-DD+									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	

Loewe	LaTraviata	AC3-DD+	128	CBR	48	5.1	HD	TOSLINK	Nein							Schlecht	Kein Audio. Denon PCM als Input | JavaScript Error in KeyEvent.

								SPDIF									
Loewe	Eishockey	AC3-DD+	256	CBR	48	5.1	HD	TOSLINK	Nein							Schlecht	Kein Audio. Denon PCM als Input | JavaScript Error in KeyEvent.

Loewe	Channelldentification	AC3-DD+	320	CBR	48	5.1	HD	TOSLINK									N.G.

Loewe	Kroemer	AC3-DD+	448	CBR	48	5.1	HD	TOSLINK	Nein							Schlecht	Kein Audio. Denon PCM als Input | JavaScript Error in KeyEvent.

Loewe	Musik	AC3-DD+	640	CBR	48	5.1	HD	TOSLINK	Nein							Schlecht	Kein Audio. Denon PCM als Input | JavaScript Error in KeyEvent.

Loewe	Eishockey	AC3-DD+	1024	CBR	48	5.1	HD	TOSLINK	Nein							Schlecht	Kein Audio. Denon PCM als Input | JavaScript Error in KeyEvent.

Loewe	Eishockey	AC3-DD+	3024	CBR	48	5.1	HD	TOSLINK	Nein							Schlecht	Kein Audio. Denon PCM als Input | JavaScript Error in KeyEvent.
WAV zu DD+ Encodierung, YUV zu H264 Encodierung und DD+ und H264 Muxing mit TotalCode Studio																	

44.1 kHz									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
Loewe		AAC-LC	128	Pseudo CBR	44.1	5.1	HD	--------									

								TOSLINK									N.G.
Loewe	ChannelIdentification	AAC-LC	640	Pseudo CBR	44.1	5.1	HD	--------									

								TOSLINK	Ja	------	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 | JavaScript Error in KeyEvent.
Loewe		AAC-HE	128	Pseudo CBR	44.1	5.1	HD	--------									

								TOSLINK									N.G.
Loewe	ChannelIdentification	AAC-HE	640	Pseudo CBR	44.1	5.1	HD	--------									

								TOSLINK	Ja	------	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 | JavaScript Error in KeyEvent.
Loewe		AC3	128	CBR	44.1	5.1	HD	--------									

								TOSLINK									N.G.
Loewe	ChannelIdentification	AC3	640	CBR	44.1	5.1	HD	--------									

								TOSLINK	Ja	------	Richtig	Ja	Nein	Gleich	Gut	Schlecht	Denon zeigt als Input AC3. Ton zu schnell. Versatz wird im Laufe des Files größer.
Loewe		AC3-DD+	128	CBR	44.1	5.1	HD	--------									Vermutung: Ton mit 44.1 wird mit 48 abgespielt. Jedoch kein Pitch . | JavaScript Error in KeyEvent.
								--------									N.V.
								TOSLINK									
Loewe	ChannelIdentification	AC3-DD+	640	CBR	44.1	5.1	HD	--------									
								--------									N.V.
								TOSLINK									
Siehe AAC, AC3 und DD+																	
VBR									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
Loewe		AAC-LC	128	VBR	48	5.1	HD	--------									

								TOSLINK									N.G.
Loewe	Eishockey	AAC-LC	640	VBR	48	5.1	HD	--------									

								TOSLINK	Ja	------	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 | JavaScript Error in KeyEvent.
Loewe		AAC-HE	128	VBR	48	5.1	HD	--------									

								TOSLINK									N.G.
Loewe	Eishockey	AAC-HE	640	VBR	48	5.1	HD	--------									

								TOSLINK	Ja	------	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 | JavaScript Error in KeyEvent.
Siehe AAC																	
STEREO									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
Loewe		AAC-LC	128	Pseudo CBR	48	2.0	HD	--------									

								TOSLINK									N.G.
Loewe	Musik	AAC-LC	640	Pseudo CBR	48	2.0	HD	--------									
								TOSLINK	Ja	------	Richtig	Nein	Ja	Leiser	Gut	Akzeptabel	
Loewe		AAC-HE	128	Pseudo CBR	48	2.0	HD	--------									

								TOSLINK									N.G.
Loewe	Musik	AAC-HE	640	Pseudo CBR	48	2.0	HD	--------									

								TOSLINK	Ja	------	Richtig	Nein	Ja	Leiser	Gut	Akzeptabel	
Loewe		AC3	128	CBR	48	2.0	HD	--------									

								TOSLINK									N.G.
Loewe	Musik	AC3	640	CBR	48	2.0	HD	--------									

								TOSLINK	Ja	------	Richtig	Nein	Ja	Gleich	Gut	Gut	
Loewe		AC3-DD+	128	CBR	48	2.0	HD	--------									

								TOSLINK									N.G.
Loewe	Musik	AC3-DD+	640	CBR	48	2.0	HD	--------									

								TOSLINK	Nein							Schlecht	Kein Audio. Denon PCM als Input | JavaScript Error in KeyEvent.
Siehe AAC, AC3 und DD+																	
SD									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
Loewe		AAC-LC	128	Pseudo CBR	48	5.1	SD	--------									

								TOSLINK									N.G.
Loewe	LaTraviata	AAC-LC	640	Pseudo CBR	48	5.1	SD	--------									

								TOSLINK	Ja	------	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 | JavaScript Error in KeyEvent.
Loewe		AAC-HE	128	Pseudo CBR	48	5.1	SD	--------									

								TOSLINK									
Loewe	LaTraviata	AAC-HE	640	Pseudo CBR	48	5.1	SD	--------									
								--------									N.G.
								TOSLINK	Ja	------	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 | JavaScript Error in KeyEvent.
Loewe	LaTraviata	AC3	128	CBR	48	5.1	SD	--------									

								TOSLINK	Nein	------	Richtig	Ja	Ja	Gleich	Gut	Schlecht	Denon zeigt als Input AC3. Video ruckelt. | JavaScript Error in KeyEvent.
Loewe	LaTraviata	AC3	640	CBR	48	5.1	SD	--------									

								TOSLINK	Nein	------	Richtig	Ja	Ja	Gleich	Gut	Schlecht	Denon zeigt als Input AC3. Video ruckelt. | JavaScript Error in KeyEvent.
Loewe		AC3-DD+	192	CBR	48	5.1	SD	--------									

								TOSLINK									N.G.

Loewe	LaTraviata	AC3-DD+	640	CBR	48	5.1	SD	--------									

								TOSLINK	Nein	-----						Schlecht	Kein Audio. Denon PCM als Input | JavaScript Error in KeyEvent.
Siehe AAC, AC3 und DD+																	
TS									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
Loewe	ChannelIdentification	AAC-LC	320	Pseudo CBR	48	5.1	HD	--------									

								TOSLINK	Ja	-----	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 | JavaScript Error in KeyEvent.
Loewe	ChannelIdentification	AAC-HE	320	Pseudo CBR	48	5.1	HD	--------									

								TOSLINK	Ja	-----	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0. Wrong MimeType -> AV-Format not supported | JavaScript Error in KeyEvent.
Loewe	ChannelIdentification	AC3	320	CBR	48	5.1	HD	--------									

								TOSLINK	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Denon zeigt als Input AC3. | JavaScript Error in KeyEvent.
Loewe	ChannelIdentification	AC3-DD+	320	CBR	48	5.1	HD	--------									

								TOSLINK	Nein	-----						Schlecht	Kein Audio. Denon PCM als Input | JavaScript Error in KeyEvent.
Alle .ts mit TotalCodeStudio erstellt, um DVB Konformität zu gewährleisten.																	
Audio Only Elementary									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
Loewe	Musik	AAC-LC	640	Pseudo CBR	48	5.1	HD	--------									

								TOSLINK	Ja	-----	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 | JavaScript Error in KeyEvent.
Loewe	Musik	AAC-HE	640	Pseudo CBR	48	5.1	HD	--------									

								TOSLINK	Ja	-----	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0. | JavaScript Error in KeyEvent.
Loewe	Musik	AC3	640	CBR	48	5.1	HD	--------									

								TOSLINK	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Denon zeigt als Input AC3. | JavaScript Error in KeyEvent.
Loewe	Musik	AC3-DD+	640	CBR	48	5.1	HD	--------									

								TOSLINK	Nein	-----						Schlecht	AV-Format not supported | JavaScript Error in KeyEvent.
WAV zu AAC Encodierung mit NeroAAC. WAV zu AC3 Encodierung mit Aften. WAV zu DD+ Encodierung mit TotalCode Studio																	
Audio Only MP4									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
Loewe	ChannelIdentification	AAC-LC	448	Pseudo CBR	48	5.1	HD	--------									

								TOSLINK	Ja	-----	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0 | JavaScript Error in KeyEvent.
Loewe	ChannelIdentification	AAC-HE	448	Pseudo CBR	48	5.1	HD	--------									

								TOSLINK	Ja	-----	falsch	Nein	Ja	Leiser	Gut	Schlecht	5.1 -> 2.0. | JavaScript Error in KeyEvent.
Loewe	ChannelIdentification	AC3	448	CBR	48	5.1	HD	--------									

								TOSLINK	Ja	-----	Richtig	Ja	Ja	Gleich	Gut	Gut	Denon zeigt als Input AC3. | JavaScript Error in KeyEvent.
Loewe	ChannelIdentification	AC3-DD+	448	CBR	48	5.1	HD	--------									

								TOSLINK	Nein	-----						Schlecht	AV-Format not supported | JavaScript Error in KeyEvent.
Siehe AAC, AC3 und DD+																	
Dolby Testmaterial									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
Loewe	Dolby TestMaterial	AC3-DD+	192	CBR	48	5.1	SD	--------									

								TOSLINK	Nein							Schlecht	Kein Audio. Denon PCM als Input | JavaScript Error in KeyEvent.
Loewe	Dolby TestMaterial	AC3-DD+	320	CBR	48	7.1	SD	--------									

								TOSLINK	Nein							Schlecht	Kein Audio. Denon PCM als Input | JavaScript Error in KeyEvent.
TotalCodeStudio									Interoperabilität	DM100	Kanalzuordnung	LFE	Synchronität	Lautstärke zu DVB	Klangqualität	Ergebnis	
Loewe	ChannelIdentification	AC3-DD+	448	CBR	48	5.1	HD	--------									

								TOSLINK	Nein							Schlecht	Kein Audio. Denon zeigt keinen Input an. Video ruckelt. | JavaScript Error in KeyEvent.
Loewe	Eishockey	AC3-DD+	448	CBR	48	5.1	HD	--------									

								TOSLINK	Nein							Schlecht	Kein Audio. Denon zeigt keinen Input an. Video ruckelt. | JavaScript Error in KeyEvent.
Loewe	Musik	AC3-DD+	448	CBR	48	5.1	HD	--------									

								TOSLINK									N.G.
Loewe	Kroemer	AC3-DD+	448	CBR	48	5.1	HD	--------									

								TOSLINK	Nein							Schlecht	Kein Audio. Denon zeigt keinen Input an. Video ruckelt. | JavaScript Error in KeyEvent.
Loewe	LaTraviata	AC3-DD+	448	CBR	48	5.1	HD	--------									

								TOSLINK									N.G.
Loewe	Aida	AC3-DD+	448	CBR	48	5.1	HD	--------									

								TOSLINK									N.G.

Legende:	**Erkläung:**
N.G.	Nicht Getestet, da sich die einzelnen Bitraten gleich verhalten.
N.V.	Nicht vorhanden, da TotalCode Studio kein 44kHz DD+ erzeugen kann.
K.Q.	Klangqualitätstest durchgeführt mit LaTraviata.
HbbTV-Gerät Details:	
Name:	Loewe ART 40 3D DR+
Software:	V.7.1.0
TV-Basis-Software:	V16.1.0
DVB-Software:	V16.1.33
DVB-Bootloader:	V2.3.0
FRC Software	V11.5.5
MSP Software:	V8.106.0
MPEG Encoder Software:	V8.22.105
Standby-Software:	V1.10.0
Feature-Prozessor-Soft.	V1.1.10
MHP-Software:	V16.1.33
CH-TYP	SL151
Artikel Nr.:	51413W47

Browser Tests

Browser	Testname	Audio Codec	Bitrate	CBR/VBR	Frequenz	Audio Format	Container	Video Codec	HTML5 Tag	Source	Mime Type	Betriebssystem	Video Format	Ausgang	Interoperabilität	Kanalzuordnung	LFE	Downmix	Synchronität
Eigenschaften																			
Firefox	CI_SHORT	OggVorbis	320	VBR	48	5.1	Ogg	Theora	<video>	Container	video/ogg	Win7 / 64Bit	HD	Gigaport	Nein				
Opera	CI_SHORT	OggVorbis	320	VBR	48	5.1	Ogg	Theora	<video>	Container	video/ogg	Win7 / 64Bit	HD	Gigaport	Ja	falsch	Nein	Faltung	Ja
Safari	CI_SHORT	AAC-LC	320	Pseudo CBR	48	5.1	MP4	H264	<video>	Container	video/mp4	Win7 / 64Bit	HD	Gigaport	Ja	falsch	Nein	Faltung	Ja
Chrome	CI_SHORT	OggVorbis	320	VBR	48	5.1	WebM	VP8	<video>	Container	video/webm	Win7 / 64Bit	HD	Gigaport	Ja	falsch	Ja	-	Ja
IE	CI_SHORT	AAC-LC	320	Pseudo CBR	48	5.1	MP4	H264	<video>	Container	video/mp4	Win7 / 64Bit	HD	Gigaport	Ja	Richtig	Ja	-	Ja
Maxthon	CI_SHORT	AAC-LC	320	Pseudo CBR	48	5.1	MP4	H264	<video>	Container	video/mp4	Win7 / 64Bit	HD	Gigaport	Ja	Richtig	Ja	-	Ja
Browser	Testname	Audio Codec	Bitrate	CBR/VBR	Frequenz	Audio Format	Container	Video Codec	HTML5 Tag	Source	Mime Type	Betriebssystem	Video Format	Ausgang	Interoperabilität	Kanalzuordnung	LFE	Downmix	Synchronität
Firefox	CI_SHORT	OggVorbis	320	VBR	48	5.1	Ogg	Theora	<video>	Container	video/ogg	MAC / OS X	HD	Gigaport	Ja	falsch	Ja	-	Ja
Opera	CI_SHORT	OggVorbis	320	VBR	48	5.1	Ogg	Theora	<video>	Container	video/ogg	MAC / OS X	HD	Gigaport	Ja	Richtig	Ja	-	Ja
Safari	CI_SHORT	AAC-LC	320	Pseudo CBR	48	5.1	MP4	H264	<video>	Container	video/mp4	MAC / OS X	HD	Gigaport	Ja	Richtig	Ja	-	Ja
Chrome	CI_SHORT	OggVorbis	320	VBR	48	5.1	WebM	VP8	<video>	Container	video/webm	MAC / OS X	HD	Gigaport	Ja	Richtig	Ja	-	Ja
IE	CI_SHORT	AAC-LC	320	Pseudo CBR	48	5.1	MP4	H264	<video>	Container	video/mp4	MAC / OS X	HD	Gigaport					
Maxthon	CI_SHORT	AAC-LC	320	Pseudo CBR	48	5.1	MP4	H264	<video>	Container	video/mp4	MAC / OS X	HD	Gigaport					

Legende:	**Erkläung:**	
Faltung:	Downmix: C auf L+R, LS auf L, RS auf R. Kein LFE.	
Faltung -3dB:	Downmix: C auf L+R, LS auf L, RS auf R, LS+RS -3DB. Kein LFE.	
Faltung -6dB:	Downmix: C auf L+R, LS auf L, RS auf R, LS+RS -6DB. Kein LFE.	
Faltung Mono:	Downmix: C auf L+R, LS-RS auf Front Mitte. Kein LFE.	
Browser Details:		
Firefox	18.0.1	
Opera	12.12.	
Safari	5.1.7.	6.0.2 unter MAC
Chrome	24.0.1312.56 m	
IE	10.0.9200.16439	
Maxthon	4.0.3.1000	
Soundkarte:		
Gigaport:	USB Audiogerät	
Treiberversion:	6.1.7601.17514	
Treiberdatum:	19.11.2010	

Ergebnis			
Lautstärke zu Original	Klangqualität	Ergebnis	Erklärung
		Schlecht	Kein Audio. Mixer zeigt auch keinen Ausschlag an.
Gleich	Gut	Schlecht	LFE in der Downmix Mischung.
Gleich	Gut	Schlecht	Kein LFE im Downmix.
Gleich	Gut	Schlecht	LS und RS werden auf den Ausgängen 7-8 des Interfaces ausgegeben. Mit VLC z.B. aber auf 5-6. Vermutung: Chrome unterstützt 7.1 Surround.
Gleich	Gut	Gut	LS und RS werden auf den Ausgängen 5-6 des Interfaces ausgegeben.
Gleich	Gut	Gut	LS und RS werden auf den Ausgängen 5-6 des Interfaces ausgegeben.
Lautstärke zu Original	Klangqualität	Ergebnis	Erklärung
Gleich	Gut	Schlecht	Kanalzuordnung: L-C-R-LS-RS-LFE
Gleich	Gut	Gut	LS und RS werden auf den Ausgängen 5-6 des Interfaces ausgegeben.
Gleich	Gut	Gut	LS und RS werden auf den Ausgängen 5-6 des Interfaces ausgegeben.
Gleich	Gut	Gut	LS und RS werden auf den Ausgängen 5-6 des Interfaces ausgegeben.
			Nicht verfügbar für Mac OS x
			Nicht getestet.