Studienarbeit

Systemfamilienbasierte Dokumentation des Digital Video Projektes (SW-Modul „vdr 1.1.20“)

Anne Preiß
Studentin der Ingenieurinformatik
an der Technischen Universität Ilmenau
Fachrichtung Integrierte Hard- und Softwaresysteme
M98, Matrikel-Nr.: 27659

6. März 2003

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Erklärung

Hiermit bestätige ich, daß ich die vorliegende Arbeit selbständig verfasst und keine anderen, als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ilmenau, den 6. März 2003

.....
Anne Preiß

.....
Detlef Streitferdt/Betreuer

2 Einleitung

Ziel des Dokumentes ist eine systemfamilienbasierte Dokumentation des „Digital Video“-Projektes VDR in der Version 1.1.20. Dies soll folgenden Studien- und Diplomarbeiten als Grundlage für eine schnelle Einarbeitung in Konzept und Quelltext des gesamten Softwaremoduls ermöglichen.

Aufgrund der Komplexität des Softwaremodules und der schnellen Versionsentwicklung, wurde die Aufteilung in konzeptionelle Module vorgenommen. Diese konnten keineswegs bis ins Detail dokumentiert werden. Im Zuge dessen enthält dieses Dokument im ersten Teil Gedanken zur automatischen Generierung von Dokumenten. Es wurde mit der Dokumentierung des Quelltextes im DocBook-Format begonnen um daraus HTML- und LaTeX-Dokumente zu erzeugen, die in digitaler Form vorliegen.

3 Reverse Engineering

3.1 Definitionen

„Ein technischer Prozeß des Verstehens, des Analysierens und der Abstraktion eines Systems zu einer neuen Form auf einem höheren Abstraktionsniveau.“

Das bedeutet die Analyse von existierender Software, die bereits erfolgreich im Einsatz ist. Ziel ist es ihre Komponenten und deren Beziehungen zu identifizieren, sowie Darstellungen auf einem hohen Abstraktionsniveau zu erzeugen. Dabei kommt es zu keiner Änderung des Systems, sondern zur Redokumentation des Quellcodes und des Designs.

Die wesentlichen Aufgaben des Reverse Engineerings bestehen darin, die Struktur und das Verhalten des Softwareanteils vorhandener IT-Systeme zu verstehen, die Software aufzubereiten, sowie nicht oder nicht mehr vorhandene Informationen über die Software wiederzugewinnen und auf abstrakten Ebenen zu beschreiben.

Reverse Engineering wird oft als präventive Softwarepflegemaßnahme immer dann durchgeführt, wenn die Softwaredokumentation nur unvollständig oder gar nicht mehr zur Verfügung steht, wenn sie unverständlich ist, wenn sie nicht den aktuellen Zustand der Software widerspiegelt oder nicht den geforderten Dokumentationsstandards entspricht.

Im Zuge des Reverse Engineerings soll für die VDR-Version 1.1.20 ein Klassendiagramm, sowie eine ansatzweise Dokumentation des Softwaremoduls erstellt, und ein Überblick über das Projekt gegeben werden.

3.2 Automatische Generierung von Dokumentationen

Die wachsende Komplexität von Softwareprodukten erfordert einen zunehmend hohen Aufwand bei der Erstellung von Dokumentation und Handbüchern.

Oftmals stellt eine große Anzahl von Produkten oder auch eine große Anzahl von Produktversionen, die heutzutage meist international verteilbar sind, hohe Anforderungen an die Entwickler, ihre Software ausreichend zu kommentieren. Hohe Innovationsgeschwindigkeiten und damit verbundene kurze Versionslebenszyklen machen eine ständige Aktualisierung von Dokumentationen notwendig.

Das ursprüngliche Zielmedium Papier findet bei der Dokumentation von Software kaum noch Anwendung. In diesem Fall bieten elektronische Medien weitaus mehr Handhabbarkeit. So erleichtert das einfache Suchen nach Begriffen und Textstellen in einer Menge von Dokumenten, und das Springen und Verzweigen innerhalb dieser Dokumente (Hypertext-Konzept) grundsätzlich das Verständnis komplexer Systeme. Darüber hinaus ermöglichen technische Fortschritte in Bereichen wie Rechengeschwindigkeit und Speichervolumina die Verwendung neuer, dynamischer Medientypen wie Audio und Animation.

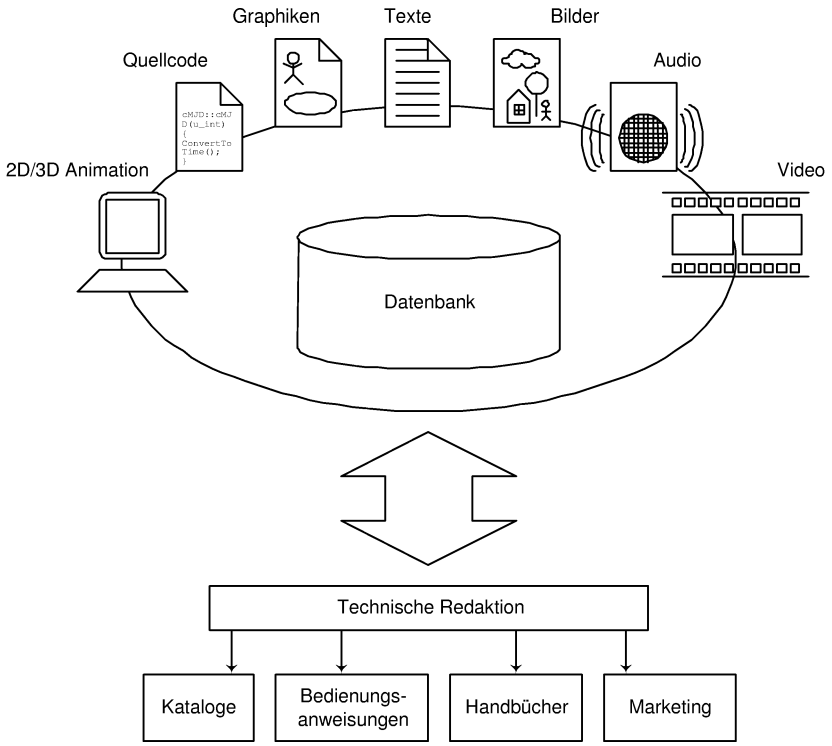

Abbildung 1: Automatische Dokumentengenerierung

Durch die automatische Generierung solcher Dokumente ergeben sich folgende Vorteile:

- Bei der automatisierten Erstellung von Dokumentationen ergeben sich enorme Einsparungen hinsichtlich Aufwand und Zeit.
- Es wird zur Senkung der Fehlerquote hinsichtlich struktureller Darstellung und in Querverweisen beigetragen.
- Es können vohandene Datenbestände genutzt werden.
- Ein schneller Datenzugriff und Austausch wird ermöglicht.
- Es ergibt sich eine höhere Aktualität.
- Es wird die Erzeugung von generischen Katalogen und verschiedenen Sichten für unterschiedliche Benutzer vereinfacht.

Für den Endbenutzer von Produkten und deren Dokumentationen bieten sich letzlich folgende Vorteile:

- Der Aktualisierungsaufwand wird verringert.
- Eventuell ergibt sich ein Mehrnutzen in Form eines Update-Mechanismus.
- Es wird ein schnellerer Zugriff auf die gewünschten und relevanten Informationen ermöglicht.
- Ausdruck von Dokumentationen nur auf Wunsch nötig.
- Es ergeben sich verbesserte Präsentationsmöglichkeiten durch dynamische Medien.

Ziel ist die Erzeugung und Verarbeitung strukturierter Dokumente, um den Prozeß der Dokumentationserstellung effizienter, flexibler und zuverlässiger zu gestalten.

Allerdings ergeben sich bei der automatischen Generierung von Dokumentationen auch wesentliche Nachteile.

Synthesetools sind im Falle undokumentierter Quelltexte nur in der Lage diese hinsichtlich ihrer Klassenstruktur zu analysieren und beispielweise in UML-Diagrammen darzustellen. Die Aussagekraft solcher Notationen ist jedoch beschränkt, da die beabsichtigte Wirkung von Operationen im allgemeinen nicht ablesbar ist. Die Erkennung größerer Zusammenhänge, die oftmals nur mit natürlicher Sprache beschreibbar sind, bleibt daher nach wie vor schwierig.

3.3 Together

Das Together ControlCenter von TogetherSoft ist eine sogenannte „Model-Build-Deploy Plattform“ und für Softwareentwicklungsteams gedacht. Es ermöglicht Reverse Engineering, durch automatische Generierung von Klassendiagrammen aus bereits vorhanden Quelltexten. Weiterhin unterstützt es die automatische Dokumentengenerierung, bspw. in Form von HTML-Dokumenten.

Es gilt somit als schnelles Software-Modellierungstool, mit modellgetriebener Architektur, das zur Optimierung von Entwicklungszeit und -kosten beitragen soll.

Zusammenfassung der Features

- simultanes Round-Trip-Engineering
- UML Diagramme
- JUnit Testing
- Visual J2EE deployment profiling
- Verteiltes Testen
- Entwicklung von Web Services
- Reverse Engineering
- Qualitätssicherung
- Audits und Messungen
- Support für bedeutende Applikationsserver

Together ist unter Ausschluß kommerzieller Nutzung als Freeware erhältlich.

Im Rahmen der Studienarbeit wurde Together in folgender Weise genutzt:

1. Erstellung von UML-Diagrammen (Klassendiagramme)
2. Analyse des Quelltextes
3. Dokumentierung des Quelltextes

Als Vorteil erwies sich hier die einfache Navigation durch verschiedene Ansichten (Diagramm, Klassenbaum, Quelltext in gekoppelten Fenstern).

3.4 Doxygen

Doxygen ist ein Dokumentationssystem für C++, C, Java, IDL (Corba, Microsoft, sowie KDE-DCOP) und teilweise auch für PHP und C#.

Es kann zur Generierung von Online-Dokumentation in HTML und/oder eines Offline-Referenz-Handbuches aus dokumentierten/kommentierten Quelltexten verwendet werden. Weiterhin wird die Generierung von RTF (MS-Word) und Postscript Dokumenten, hyperlinked PDF, komprimiertem HTML, und UNIX-Hilfeseiten unterstützt.

Die Dokumentation wird direkt aus dem Quellcode extrahiert, was die Erhaltung der Konsistenz von Dokumentation und Quellcode vereinfacht.

Doxygen kann ähnlich dem Reverse Engineering, wie es Together vornimmt, auch aus unkommentiertem Quellcode eine Strukur erstellen. Das vereinfacht das Verständnis großer Distributionen. Die Beziehungen zwischen den verschiedenen Elementen werden durch das Einbinden von automatisch generierten Abhängigkeitsgraphenen, Vererbungsdiagrammen, sowie Kollaboratonsdiagrammen visualisiert.

Zusammenfassung der Features

- Unterstützt C/C++, Java, (Corba, Microsoft, und KDE-DCOP) Java, IDL, und teilweise auch C# und PHP Quelltexte.
- Unterstützt die Dokumentation von Files, Namensräumen, Klassen, Structs, Unions, Templates, Variablen, Functionen, Typedefs, Aufzählungen und Defines.
- JavaDoc (1.1), Qt-Doc, und KDOC kompatibel.
- Generiert automatisch HTML-Klassendiagramme (anklickbare Image Maps) und EPS Bilder.
- Benutzt das Zeichenwerkzeug des Graphviz, um Abhängigkeitsgraphen, Kollaborationsdiagramme und graphische Klassenhierarchien zu erzeugen.
- Erlaubt das Anlegen der Dokumentation im Header-File (vor der Deklaration einer Instanz/Entity), im Quellcode (vor der Definition einer Instanz/Entity) oder in einem gesonderten File.
- Kann eine Liste aller Mitglieder einer Klasse (einschließlich vererbter) auf ihrem Protection-Level erzeugen. Erkennt automatisch öffentliche, geschützte und private Bereiche. Die Extrahierung privater Klassenelemente ist optional.
- Erzeugt gleichzeitig eine Dokumentation im Online-Format (HTML und UNIX-Hilfeseiten) und Offline-Format (und RTF). Die Erzeugung dieser Dokumente kann auf Wunsch auch deaktiviert werden. Weiterhin, kann aus der HTML-Ausgabe compressed HTML, sowie PDF-Dokumente erzeugt werden.
- Schließt einen kompletten C-Präprozessor ein, der eine Syntaxanalyse der bedingter Codefragmente erlaubt und die Auswertung eines Teiles oder der gesamten Makrodefinitionen erlaubt.
- Generiert automatisch Referenzen zu dokumentierten Klassen, Dateien, Namensräumen und Elementen. Die Dokumentation globaler Funktionen, globaler Variablen, Typendefinitionen, defines und Aufzählungen wird ebenfalls unterstützt.
- Schließt eine schnelle, Ebenen basierte Suchmaschine ein. Damit können Strings oder Worte in den Klassen- oder Elementdokumentationen gesucht werden.
- Erlaubt ortsunabhängige Verweise auf Dokumentationen für andere Projekte oder auf andere Teile des selben Projektes.

- Die Einbindung von undokumentierten Klassen wird ebenfalls unterstützt, so daß ein schneller Überblick über Struktur und Schnittstellen großer Programmteile ohne Blick auf die Implementationsdetails möglich gemacht wird.
- Erlaubt automatisches Querverweise von (dokumentierten) Instanzen mit ihrer Definition in Quellcode.
- Erlaubt die Einbindung von Funktions-, Element- und Klassendefinitionen in die Dokumentation.
- Alle Optionen werden aus einer einfach zu ändernden Datei und einer kommentierten Konfigurationsdatei (optional) ausgelesen.

Auch bei Doxygen handelt es sich um Freeware. Es wurde unter Linux entwickelt, ist aber portierbar, und läift somit unter den meisten UNIX-Distributionen, Windows 9x/NT und Mac OS X.

Im Rahmen der Studienarbeit wurde Doxygen zur Erstellung einer HTML-Dokumentation genutzt. Diese hätte auch mit Toghether erstellt werden können, jedoch bindet Klaus Schmiedinger ab Version 1.2.20 in den VDR-Quelltext die Dokumentation für *Doxygen* ein. Weiterhin wird eine Doxygen-Konfigurationsfile mit Informationen für den Präprozessor angeboten. Dies kann für zukünftige Dokumentationen als Erweiterung zu dieser genutzt werden.

4 Das Projekt VDR

4.1 Video Disc Recorder

„With the availability of high capacity hard disks and MPEG encoders/decoders the idea of a fully digital video recorder comes to mind.

Meanwhile there are several commercial products on the market that implement this functionality. Some of these systems are mainly targetted on working together with special services provided by the system manufacturer, which offer the consumer a personalized recording strategy - and sometimes are available only at extra costs. The downside of these systems is often the lack of a really flexible „timer recording“ facility, and the manufacturers' tendencies to implement new ways of imposing advertisements on the viewer (since it would become too easy to skip the commercial breaks). They also don't support what one would expect to come natural with digital recording: on-disk editing. And finally they won't let you get access to the actual recorded digital data (for instance for archiving). This project describes how to build your own digital satellite receiver and video disk recorder. It is based mainly on the DVB-S digital satellite receiver card, and the driver software developed by the LinuxTV project.“[1]

Das unter Linux programmierte Softwaremodul VDR implementiert einen vollständigen Digitalempfänger und Videorekorder. Er ermöglicht die Verarbeitung digitaler Signale aus Satelliten-, Kabel- und terrestrischem Empfang, und bietet damit Fernsehen und Aufnahme in annähernd DVD-Qualität.

Dafür wird mindestens eine DVB-Karte benötigt, die den zu verarbeitenden Mpeg2-Datenstrom liefert. Entsprechend der Empfangsart unterscheidet man zwischen DVB-S-, DVB-C- und DVB-T-Karten. Um parallel Fernsehen und Aufnehmen zu können, oder auch zeitversetzt mit dem Anschauen aufgenommener Sendungen noch während der laufenden Aufnahme zu beginnen (sogen. *Timeshifting*), sind allerdings 2 solcher Karten nötig. Grundsätzlich können per VDR bis zu vier Karten in einem unterstützt werden.

Weiterhin ist damit eine problemlose Nachbarbeitung von Filmen, wie z. B. das Herausschneiden von Werbeblöcken, sowie die Integration zusätzlicher Features wie DVD-Player, MP3-Player und Fernbedienung möglich.

Die primäre Karte, also jene, die mit dem Fernsehgerät verbunden ist, sollte eine „full featured“ Karte sein. Das bedeutet, es sollte ein MPEG-Dekoder implementiert sein, sowie Audio- und Video-Signalausgänge unterstützt werden.

VDR nutzt das *On Screen Display* (OSD) der DVB-Karte zur Anzeige eines Menüs. Dieses kann entweder per PC-Tastatur, einer selbstgebauten *Remote Control Unit* (RCU) oder über die *Linux Infrared Remote Control* (LIRC) bedient werden. So kann auf die zusätzlich abgestrahlten EPG-Programminformationen zugriffen, und über diese der Video-Rekorder programmiert werden. Auch ein bequemes Zuschneiden ist per OSD möglich.

[1] Klaus Schmidinger, www.cadsoft.de/vdr

Der Fernzugriff ist über das *Simple Video Disk Recorder Protocol*[2] möglich. Auf das kann über Port 2001 zu gegriffen werden, beispielweise per Telnet. Tools wie VdrAdmin oder KVDR erleichten die Bedienung.

VDR wird als Freewarelösung von Klaus Schmidinger zu Verfügung gestellt, und von ihm, mit Unterstützung weiterer Programmierer, fortlaufend weiterentwickelt. Das umfasst die Programmierung neuer Plugins und Tools durch engagierte Personen, aber auch die ständige Verbesserung des VDR und Erstellung von Bugfixes durch den Autor selbst. [3]

Zusammenfassung der Features

- Die Bedienung erfolgt gänzlich über das OSD der DVB-Karte, die Infrarot-Fernbedienung (LIRC/ RCU) oder die Tastatur.
- Unterstützt verschiedene DVB-Karten. Es wird lediglich eine vollständig mit Video-Ausgang ausgestattete Karte benötigt, insgesamt können bis zu vier Karten betrieben werden, genauso wie ein *Conditional Access*-Modul.
- Kanalgruppen
- Anzeige der EPG-Daten nach Kanal und Zeit („What's on now/next“)
- Timer: Manuelle Programmierung per EPG-Daten, Prioritäts/Lebensdauer Modell, einmalige oder sich automatisch wiederholende Timer, die EPG Untertitel-Informationen selbstständig als Aufnahmetitel heranziehen.
- Aufnahmespeicher auf der Platte: Automatische Aufteilung der Aufnahme in Dateien kleiner 2 GB, Unterstützung von verschiedenen Speicherverzeichnissen, die auch über mehrere Platten verteilt sein können, Unterstützung für hierarchische Speicherung
- Unterstützung für verschiedene Audio-Tracks und Dolby Digital
- Sofortaufnahmen (Instant Recording)
- Verschiedene Playback-Modi: Normal, Pause, Vorspulen/Zurückspulen (Verschiedene Geschwindigkeiten), Springen an bestimmte Markierungen, Springen 60 Sekunden
- Unterstützung bei der Bearbeitung von Aufnahmen (I-frame Genauigkeit: $\tilde{0}$.5s)
- Multiple Language Support
- Unterstützung für die Ausführung von Systembefehlen und deren Bildschirmausgabe
- Netzwerk Unterstützung (SVDRP): Timer- und Aufnahmeverwaltung via Telnet
- Automatischer Shutdown/Wakeup (für bestimmte Mainboards)
- Unterstützung für die automatische Ausführung von Befehlen für Aufnahme-Start und -Ende, sowie die -Bearbeitung
- Unterstützung von MP3/DVD/(S)VCD/DivX-Playback und DivX-Aufnahme über Patches

[2] SVDRP
[3] Nähere Informationen dazu sind auch unter www.cadsoft.de/vdr oder www.vdrportal.de zu finden.

Hauptkomponenten (grundlegend benötigt)

Rechner mit Betriebssystem	ab PII (mind. 32 MB RAM, große Festplatte, CD/DVD-Brenner zwecks Archivierung) ausgestattet mit Linux
DVB-Karte + Treiber	DVB(-s-), DVB(-t-), DVB(-c-) Karte mit Treibern, von der LinuxTV Homepage oder vom Hersteller zu beziehen
Remote Control	Keybord/RCU/LIRC
VDR	Der Hauptteil dieses Projektes beschäftigt sich mit der aktuellen Implementation eines On Screen Menu Systems, daß die vollständige Kontrolle über die Kanaleigenschaften, Timer und Aufnahmen ermöglichen soll.

4.2 Die Version 1.1.20

Mit Beginn dieser Studienarbeit war die Entwickler-Version 1.1.20 als aktuellste Version verfügbar. Somit habe ich für die Dokumentation nicht die an der TU Ilmenau für das Digitale Video Projekt verwendete Entwickler-Version 1.1.14 dokumentiert. Das bedeutet, daß sicherlich hier einige Funktionen mehr dokumentiert sind.

Mit Abschluß der Arbeit war bereits die Version 1.1.25 verfügbar, was verdeutlicht wie rasend schnell die Weiterentwicklung des VDR voranschreitet.

Allgemein kann man sagen, daß etwa seit Version 1.1.0 mit der Implementation des Plugin-Interfaces begonnen wurde. Seit Version 1.1.1 existiert das „hello“-Plugin als Beispiel für die Integration eines Plugins.

Die Version 1.1.20 ist seit dem 15.12.2002 verfügbar.

Sie bietet im Vergleich zur vorhergehenden Version keine gravierenden Neuerungen, sondern eher kleinere Anpassungen.

So wird beispielsweise automatisch überprüft, ob eine Verbindung zur Tastatur besteht. Daraufhin wird die Bedienung per Keyboard nur erzeugt, wenn VDR im Vordergrund ausgeführt wird.

Wird die Tastatur als Fernbedienung genutzt, akzeptieren die Texteingabefelder im Menü die direkte Eingabe (funktioniert nur für Tasten, die nicht anderweitig auf der Fernbedienung verwendet werden). Auch Plugins können die *cKbdRemote* in den *raw mode* umschalten, in dem jede Tastatureingabe über den neuen *kKbd*-Tastencode verfügbar ist und nicht als normale Fernbedienungsfunktion verarbeitet wird. Das Löschen von Buchstaben im *Einfügen*-Modus von Textfeldern ist möglich.

Weiterhin ist die Übernahme einer aktiven SVDRP-Verbindung in den Account für den Fall eines Shutdowns möglich. Hinzu kommen noch kleinere Änderungen im Makefile.

Entscheidend ist sicher auch im Rahmen dieser Dokumentation, daß die Nutzung von „Doxygen“ zur Generierung einer Quellcode-Dokumentation vereinfacht wird. Zukünftig wird eine Doxyfile-Konfiguration zu Verfügung gestellt, und im Quelltext mit entsprechenden Kommentaren gearbeitet.

5 Systemfamilien

„Ein aufkommender Ansatz zur schnelleren und kostengünstigeren Softwareentwicklung sind Systemfamilien. Eine Systemfamilie beschreibt eine Gruppe ähnlicher Software-Systeme, denen eine gemeinsame Architektur sowie gemeinsam genutzte Komponenten zugrundeliegen. Darüber hinaus implementiert jedes Mitglied der Systemfamilie nur noch seine spezifischen Besonderheiten. Durch den hohen Grad an Wiederverwendung sowohl auf Modell- als auch auf Implementierungsebene können neue Familienmitglieder in kurzer Zeit mit hoher Qualität von kleinen Teams entwickelt werden.“[4]

Entsprechend dem Pluginkonzept ist das VDR-Modul modularisierbar. Ausgehend von einem grundlegenden Software-Modul, kann das Paket in seiner Funktionalität durch verschiedene Tools und Plugins ergänzt werden. Diese Aufteilung in verschiedene Systemkomponenten soll die grundlegenden Funktionen und Variationsmöglichkeiten in der Gestaltung des digitalen Videorekorders aufzeigen.

Es sollen also gemeinsame und variable Merkmale des Systems, deren Wechselwirkungen, sowie Bereiche mit hoher Variabilität identifiziert werden.

5.1 Aufspaltung in Bereiche

Im Zuge einer Dokumentation habe ich zunächst begonnen die vorhandene Software in folgende funktionale „Module“ aufzuspalten.

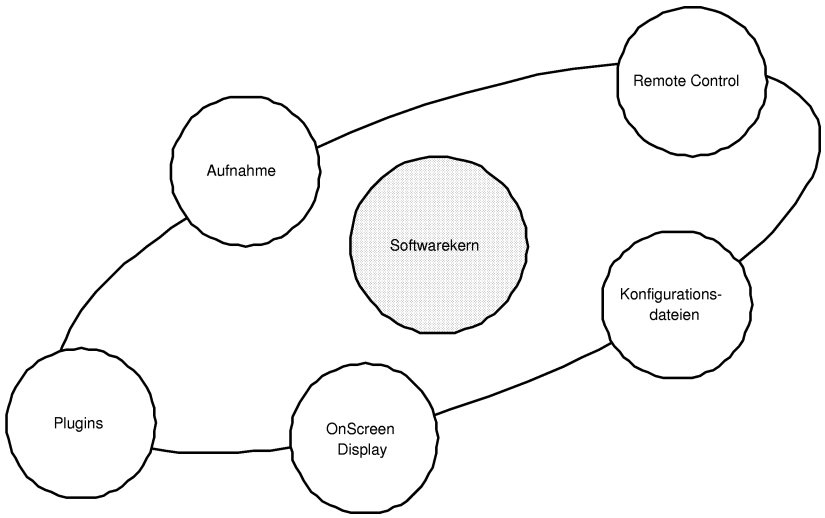

Abbildung 2: Komponenten

[4] Matthias Riebisch, Detlef Streitferdt, Kai Böllert, Methoden und Werkzeuge zur Entwicklung von Systemfamilien, Technische Universität Ilmenau

1. *Softwarekern* - er enthält grundlegende, für alle Komponenten benötigten Dateien

Beschreibung	***C-File***	***Klassen***
Video Disk Recorder Hauptprogramm	Vdr.c	
Handling der Konfigurationsdateien	Config.c	cCommand cSVDRPhost cCaDefinition cConfig CDiseqcs cSources cKeys cSetup cTimers cSVDRhosts cCaDefintions cKeyMacros cChannels cCommands cSetupLine
DiSEqC-Handling	Diseqc.c	cDiseqc cDiseqcs
Handling der Kanäle	Channels.c	cChannel cChannels
Basis Geräte-Interface	Device.c	cDevice cTSBuffer
Handling der Datenquellen	Sources.c	cSource cSources
Status Überwachung	Status.c	cStatus
Einfache Thread-basierte Klassen	Thread.c	cThread cCondVar cMutex cMutexLock cThreadLock cPipe
Verschiedene Tools	Tools.c	cPoller cFile cSafeFile cLockFile cListObject cListBase cList
Transfer Modus	Transfer.c	cTransfer cTransferControl

SPU Dekoder-Prototyp	Spu.c	cSpuDecoder
Aktualisierung der EPG-Daten	Eit.c	cEventInfo cSchedule cSchedules cSIprocessor
EPG-Daten Scanner	EitScan.c	cEitScanner
Basis Audio-Interface	Audio.c	cAudio cAudios cExternalAudio
Basis Player-Interface	Player.c	cPlayer cControl
VDR Plugin-Interface	Plugin.c	cPlugin cDll cDlls cPluginManager
Basis Empfänger-Interface	Receiver.c	cReceiver
Streaming MPEG2-Remultiplexer	Remux.c	cRemux
DVB Geräte-Interface	Dvdbdevice.c	cDvbDevice cDvbTuner
DVB Player	Dvbplayer.c	cDvbPlayerControl cBackTrace
SPU-Dekoder für DVB-Geräte	Dvbspu.c	cDvbSpuBitmap cDvbSpuPalette cDvbSpuDecoder

2. *Konfigurationsdateien* - sie enthalten für die verschiedenen Komponenten benötigte Konfigurationsdaten

Beschreibung	***Konfigurationsdatei***
DiSEqC-Konfiguration	Diseqc.conf
User definable setup	Setup.conf
Timer-Konfiguration	Timers.conf
Kanal-Konfiguration	Channels.conf
Quell-Konfiguration	Sources.conf
Konfiguration der SVDRP-Hosts	Svdrphosts.conf

Konfiguration für bedingten Zugriff	Ca.conf
Benutzerdefinierte Befehle	Commands.conf
Tastenbelegung für die Fernbedienung	Remote.conf
Tastenbelegung für eine LIRC-fernbedienung	Lirc.conf
Definition von Kurzbefehlen	Keymacros.conf

3. *On Screen Display* - für die Darstellung der Bedienoberfläche notwendigen Dateien

Beschreibung	***C-File***	***Klassen***
Aktuelle Menü-Implementationen	Menu.c	cMenuMain cDisplayChannel cDisplayVolume cMenuRecordings cRecordControl cRecordControls cReplayControl
Elemente des universellen Menüs	Menuitems.c	cMenuEditItem cMenuEditIntItem cMenuEditBoolItem cMenuEditChrItem cMenuEditStrItem cMenuEditStraItem cMenuTextItem cMenuSetupPage
Abstract On Screen Display layer	Osd.c	cOsd cOsdItem cOsdObject cOsdMenu
Basis-Interface zum On Screen Display	OsdBase.c	cOsdBase cWindow CPalette cBitmap
Implementation des DVB On Screen Display	Dvbosd.c	cDvbOsd
Font-Handling für das DVB On Screen Display	Font.c	cFont
Abstract user interface layer	Interface.c	cInterface

4. *Remote Control* - für die Fernbedienung nötige Dateien und mögliche Varianten

Beschreibung	***C-File***	***Klassen***
Allgemeine Fernbedienungbehandlung	Remote.h	cRemote cRemotes cKbdRemote
Key Handling für die Fernbedienung	Keys.c cKeys cKeyMacro cKeyMacros	cKey
LIRC Fernbedienung	Lirc.c	cLircRemote
RCU Fernbedienung	Rcu.c	cRcuRemote
Simple Video Disk Recorder Protocol	Svdrp.c cPuteHandler cSVDRP	cSocket

5. *Aufnahme* - für die Aufnahme vorhandene Dateien und zusätzliche Weiterbearbeitungskomponenten

Beschreibung	***C-File***	***Klasse***
Der aktuelle DVB Recorder	Recorder.c	cRecorder
Handling aufgenommener Dateien	Recording.c	cResumeFile cRecording cRecordings cMark cMarks cRecordingUserCommand cIndexFile cFileName
Timer Handling	Timers.c	cTimer cTimers
Videoschnitt-Funktionen	Cutter.c	cCutter cCuttingThread
Funktionen für ein verteiltes Videoverzeichnis	Videodir.c	cVideoDirectory

6. *Plugins* - mögliche bzw. im Paket der aktuellen Entwicklerversion bereits enthaltene Plugin-Dateien

Beschreibung	***C-File***	***Klassen***

Internationalisierung	I18n.c	cI18Entry

5.2 Der DVB-Treiber

Bevor die nähere Beschreibung der einzelnen Systemkomponenten erfolgt, soll kurz auf den DVB-Treiber eingegangen werden, der die Schnittstelle zwischen VDR und DVB-Karte bildet.

Digitales Fernsehen soll unabhängig von Software und Betriebssystem der TV-Anlage betrieben werden können. Die Standards, die diese Interoperabilität garantieren, werden von den Mitgliedern des DVB definiert. Da diese Standards offen sind, müssen alle Hersteller von entsprechenden DVB-Systemen, garantieren können, daß ihr DVB-Equipment auch mit dem anderer Hersteller funktioniert. Leider versuchten mit dem Beginn digitalen Fernsehens in Europa einige Firmen wie OpenTV, Beta Research oder Canal+ eine eigene Software-Schicht auf die offenen DVB-Standards zu setzen. Die ersten auf dem Markt verfügbaren Digitalempfänger waren folglich inkompatibel - elektronische Programmguides, die für den deutschen Markt produziert wurden, konnten nicht mit französischen Digitalempfängern verwendet werden, und umgekehrt. Wie vermutet, ist keiner der Hersteller groß genug, um seinen eigenen Standard dem europäischen oder weltweiten Markt aufzuerlegen.

Der offene Standard für zufriedenstellende Darstellung digitalen Fernsehens wird MHP genannt (Multimedia-Home-Plattform). Die MHP1.0-Spezifikation wurde von den DVB Mitgliedern vervollständigt und kann von ETSI heruntergeladen werden. Jedoch basiert MHP sehr auf der Java Technologie, die von Sun Microsystems herausgegeben wurde. Infolgedessen wird die Offenheit der MHP-Implementierungen durch sogenannte Implementationsvereinbarungen begrenzt.

Linux kommt als eingebettetes Betriebssystem für Digitalempfänger, digitale TV-Anlagen und mobile Geräte zunehmend in Mode. Viele Hersteller der sogenannten Verbraucherelektronik experimentieren bereits damit. Bedauerlicherweise geben nicht alle den Quellcode für ihre Treiber für entsprechende En- und Dekoder frei. So kann die Open-Source-Community derzeit nicht viel zur Behebung von aktuellen Treibermängeln und der Entwicklung digitalen Fernshens beitragen. Dies beeinträchtigt teilweise auch die Funktionalität des VDR.

5.3 Softwarekern

Der Softwarekern enthält grundlegende, für alle Komponenten benötigte Dateien. Er setzt in seiner Funktionalität auf die DVB-Karte auf, und ermöglicht des Betreiben des digitalen Fernsehers.

5.3.1 Vdr.c

Das *„Video Disk Recorder Main Program“* startet die Anwendung, initialisiert zu Beginn alle Einstellungen und enthält die *„main program loop“*. In Abbildung 3 sind die beim Aufruf des Hauptprogramms abgearbeiteten Funktionen aufgelistet.

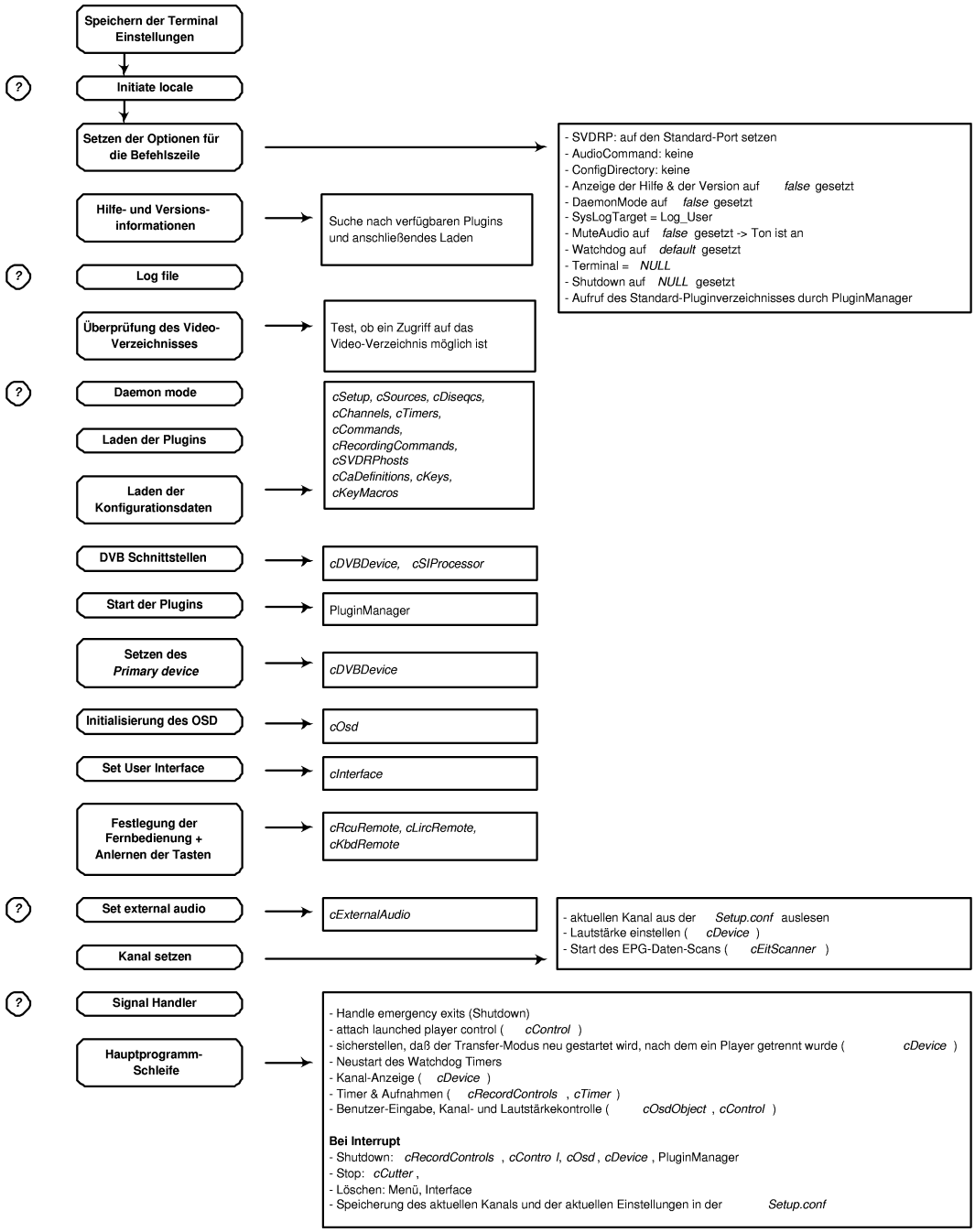

Abbildung 3: Vdr Main

5.3.2 Config.c

In dieser Datei sind alle für das Handling der Konfigurationsdateien notwendigen Klassen enthalten.

So stellt *cConfig* als parametrisierte Klasse, die Eigenschaften für die folgenden Unterklassen zur Verfügung. Diese erlauben den Zugriff anderer Klassen auf die entsprechenden Konfigurationsdateien.

- cDiseqcs
- cSources
- cKeys
- cSetup
- cTimers
- cSVDRPhosts
- cCaDefinitions
- cKeyMacros
- cChannels
- cCommands

cSetupLine ist für das Setup der Zeilen im Hauptmenü des Onscreendisplays verantwortlich.

cCommands dient der Abfrage der Befehlskonfiguration aus der Konfigurationsdatei *Commands.conf*.

Die Klasse *cSVDRPhost* übernimmt die Konfiguration und Analyse der SVDRP-Hosts. Diese sind in der Konfigurationsdatei *SVDRPhosts.conf* angegeben. Diesen Clients soll der Zugriff auf den VDR über das Web ermöglicht werden. Dabei kann es sich um einzelne Host, aber auch ganze Unternetze handeln. Dies über folgende Methoden realisiert:

- cSVDRPhost()
- Parse()
- Accepts() - Prüfung gültiger Hosts

Die Klasse *cCaDefinition* regelt die Konfiguration der verschlüsselt empfangenen Kanäle. Diese sind in der Konfigurationsdatei *Ca.conf* eingetragen. Es werden folgende Methoden verwandt:

Die Klasse *cSetupLine* übernimmt die Konfiguration der Setup-Zeile im OSD Menü. Dazu wird geprüft, ob eventuell Plugins vorhanden sind und deren Name wird ermittelt

cSetup konfiguriert mit Hilfe der in der *Setup.conf* gespeicherten Nutzer-Einstellungen das Setup.

5.3.3 Diseqc.c

Diese Datei enthält alle zur DiSEqC-Konfiguration benötigten Klassen. Das allgemeine Handling wird über *cDiseqc* realisiert. *cDiseqcs* übernimmt die Konfiguration durch Nutzung der *cConfig*, die wiederum den Zugriff auf die Konfigurationsdatei *Diseqc.conf* ermöglicht.

5.3.4 Sources.c

cSource übernimmt das Handling der Kanalquellen.

cSources übernimmt die Konfiguration durch Nutzung der *cConfig*, die wiederum den Zugriff auf die Konfigurationsdatei *Sources.conf* ermöglicht.

5.3.5 Channels.c

Diese Datei enthält alle für die Kanalbehandlung benötigten Klassen.

cChannel ist für die Zusammenstellung des Kanalmenüs verantwortlich. Dabei geht es zunächst grundlegend um die Abfrage von kanalspezifischen Variablen, wie der Transponderfrequenz, der Signalquelle, der Symbolrate, der Video-, Audio- und Teletext-PID. Weiterhin müssen Daten für den bedingten Zugriff auf verschlüsselte Kanäle, die Transport Stream ID, die Sequenznummer, die Polarisation, sowie die Service-, Network- und Radio-ID abgefragt werden.

cChannels übernimmt die Konfiguration durch Nutzung der *cConfig*, die wiederum den Zugriff auf die Konfigurationsdatei *Channels.conf* ermöglicht. Diese wird teilweise durch die *Sources.conf* gespeist.

5.3.6 Status.c

Status.c enthält alle zur Anzeige von Statusinformationen des VDR notwendigen Klassen. Die in *cStatus* enthaltenen Funktionen können von abgeleiteten Klassen zum Empfangen dieser Informationen verwendet werden. Sie werden aufgerufen, wenn sich die entsprechenden Status-Informationen ändern.

5.3.7 Thread.c

Enthält einfache Thread-basierte Klassen, wie *cMutex*, *cThread*, *cMutexLock*, *cThreadLock*, *cPipe*.

cThread kann verwendet werden, wann immer ein eigener Thread laufen soll. Dies ist beispielweise der Fall, wenn Daten einer Datei ausgelesen werden soll. Dazu muß lediglich eine von Klasse von cThread abgeleitet, und die notwendige Funktionalität implementiert werden.

cMutexLock kann dazu verwendet werden, einen Mutex zu sperren und dabei sicherzustellen, daß dieser beim Verlassen wieder entsperrt wird. Dabei können mehrere Sperren gespeichert werden. So kann beispielsweise eine Funktion, die eine andere, *cMutexLock* verwendende, Funktion aufruft, ebenfalls *cMutexLock* verwenden um die Sperre zu verlängern.

cThreadLock stellt die selbe Funktion wie cMutexLock zu Verfügung, allerdings für Threads.

cPipe implementiert eine Pipeline, die alle unnötigen Datei-Deskriptoren eines Child-Prozesses schließt.

5.3.8 Tools.c

Enthält verschiedene Tools zu Nutzung durch andere Klassen.

cPoller übernimmt die periodische Abfrage bestimmter Daten.

Für die Behandlung von Dateien werden die Klassen *cFile*, *cSafeFile* und *cLockFile* zu Verfügung gestellt.

cListObject übernimmt das Handhabung der Objekte der Menülisten. *cListBase* ist für die allgemeine Auflistung von Objekten verantwortlich. *cList* wiederum bietet als Template, die Möglichkeit zu Ableitung von Unterklassen, in denen Listen realisiert werden.

5.3.9 Transfer.c

Übernimmt das Handling des Transfers zwischen *cReceiver* und *cPlayer*. Dafür stehen die Klassen *cTransfer* und *cTransferControl* zur Verfügung.

5.3.10 Plugin.c

Die Plugin-Schnittstelle des VDR. Sie enthält die Klassen *cPlugin*, *cDll*, *cDlls* (Dll-Listing), sowie *cPluginManager*. Die somit zur Verfügung gestellten Container-Funktionen können zur Integration neuer Plugins genutzt werden.

5.3.11 Dvddevice.c

Diese Datei stellt die Schnittstelle zu den DVB-Geräten (*DvbDevices*) dar.

cDvbDevice setzt auf bestehenden *DvbDevices* auf und implementiert diese. Darauf kann dann über die Programmierschnittstelle des Linux-DVB-Treibers zugegriffen werden. Ebenso wird ein neues OSD an definierten Koordinaten initialisiert. Die in *Device.c* enthaltenen Klassen und Funktionen setzen auf die Implementationen der Klassen in *DvbDevice.c* auf.

5.3.12 Device.c

Zusammenfassend handelt es sich hier um die Basis-Device-Schnittstelle, die die Behandlung der einzelnen Karten regelt.

Mit den Definitionen in *ePlayMode* werden vier verschiedene Wiedergabe-Modi zu Verfügung gestellt. Zum einen die Wiedergabe von Audio und Video über den Dekoder oder den Player, weiterhin die Audio-Wiedergabe über dem Player, verbunden mit der Video-Wiedergabe über den Dekoder. Der letzte Wiedergabe-Modus unterstützt die Audiowiedergabe vom Player ohne Video-Wiedergabe (Black Screen).

Aus der Klasse *cDevice* können alle Geräte abgeleitet werden. Sie stellt Funktionen zur Verfügung, die zum einen die Gesamtzahl von Karten angibt, davon eine primäre Karte auswählt, und die Verwendung anderer Karten festlegt. Dazu gehört auch die Festlegung welche Kanäle von welcher Karte empfangen werden (entsprechend terrestrischer Austrahlung, oder per Kabel oder Satellit).

Das dient der Verwaltung mehrerer Karten, die parallel arbeiten. Weiterhin ermöglicht es die gleichzeitige Aufnahme mehrerer verschiedener Sendungen, oder das parallele Fernschauen auf einem anderen Kanal. Ebenso wird hier die Kanalumschaltung vorgenommen.

Zudem erzeugt es ein neues *cOsdBase*-Objekt, um die Darstellung von Informationen auf dem Display zu ermöglichen.

Außerdem werden Funktionen für den Player, das PID-Handling, Audio-, Video- und Empfängereigenschaften, sowie das Aufnehmen von Bildern zur Verfügung gestellt. Eine ausführlichere Beschreibung ist in der HTML-Dokumentation zu finden.

Die Klasse *cTSBuffer* implementiert letztendlich noch einen Buffer, der dem *Device* erlaubt größere Datenmengen mit lediglich einem *Read()*-Aufruf vom Treiber zu erhalten. Zudem übernimmt sie die Resynchronisation bei verloren gegangen Daten des Transport-Streams.

5.3.13 Dvbspu.c

Diese Datei stellt mit der Klasse *cSpuDecoder* einen SPU[5]-Dekoder für DVB-Geräte zur Verfügung.

cDvbSpuPalette stellt die DVB-Farbpalette zur Verfügung. Weiterhin existiert die Klasse *cDvbSpuBitmap*, die das Setzen der Bilder auf dem DVB-Diplay übernimmt. Die Klasse *cDvbSpuDecoder* implementiert den in *Spu.c* gegebenen SPU-Dekoder.

5.3.14 Spu.c

Enthält einen Prototypen für einen SPU-Dekoder.

5.3.15 Dvbplayer.c

Implementiert den DVB-Player. Enthalten ist die Klasse *cDvbPlayerControl*, die sämtliche Informationen zur Player-Steurung über die Schnittstelle zu den in *Player.c* enthaltenen Klassen erhält.

5.3.16 Player.c

cPlayer liefert die Basis-Schnittstelle zum DVB-Player, oder genauer gesagt zum entsprechenden *Device*. Über diese Schnittstelle werden die abzuspielenden Daten an das *DvbDevice* gesendet. Zudem wird mit der Klasse *cControl* der Zugriff auf den Player über das OSD-Menü zur Verfügung gestellt.

5.3.17 Eit.c/EitScan.c

EIT-Scanner, der die mitgesandten EPG-Daten zur Verfügung zu stellt.

5.3.18 Audio.c

Liefert die Basis-Audio-Schnittstelle. Über die Klasse *cAudio* werden alle grundlegenden Funktionen zur Steuerung von dem im Menü enthaltenen Objekt hinsichtlich seiner Audioeigenschaften geboten. *cAudios* übernimmt die entsprechende Konfiguration. *cExternalAudio* übernimmt die Behandlung externer Audioeinstellungen.

[5] Signal Processing Unit

5.3.19 Receiver.c

Bietet mit der Klasse *cReceiver* eine Basis-Schnittstelle zum empfangenden *Device*.

5.3.20 Remux.c

Enthält einen Streaming-MPEG2-Remultiplexer, der die Umkodierung bestimmter Formate übernehmen kann.

Die aufrufende Schnittstelle ist *cRemux::Process()*. Diese Funktion funktioniert folgendermaßen:

Data zeigt auf eine Menge von Daten, die aus *Count*-Bytes besteht. Die *Process*- Funktion soll versuchen so viele Daten wie möglich zu remultiplexen, und einen Zeiger auf den entstehenden Buffer zurück geben. Dieser Buffer unterscheidet sich typischerweise von den Eingangsdaten, aber im einfachsten Fall, wenn *Process* nichts tut, kann er auf die Originaldaten zeigen.

Bei der Rückgabe wird *Count* auf die Zahl der verarbeiteten Bytes gesetzt (beispielsweise aus den Eingabedaten), die verarbeitet wurden, während *Result* zurückgibt wieviele Bytes der Rückgaeb-Buffer enthält.

PictureType soll auf *NO_PICTURE* gesetzt werden, wenn die zurückgegebenen Daten kein neues Bild starten, oder auf *I_FRAME*, *P_FRAME* oder *B_FRAME*, falls ein Startpunkt für ein neues Bild gefunden wurde.

Das bedeutet auch, daß die zurückgegebenen Daten höchstens einen gesamten Video-Frame enthalten können, da der nächste Frame mit seinen eigenen Wert für *PictureType* zurück gegeben werden muß.

Process sollte dabei die Latenzzeit so kurz wie möglich halten, um einen schnellen Start des VDR Tranfer Mode zu ermöglichen (Anzeige des Signale von einer der DVB- oder anderen Karten). Zu diesem Zweck, kann diese Funktion entscheiden, ob der erste Durchsatz eingehender Daten tatsächlich verarbeitet werden soll, und kann den aktuellen Prozess nach einige Sekunden abbrechen (soweit das für den Algorithmus möglich ist). Das kann zu Beginn einen nicht-optimalen Stream ergeben, der für normale Aufnahmen nicht von Bedeutung ist, aber das Durchschalten verschlüsselter Kanäle im *Tranfer Modus* verschnellert.

In dem sich ergebenden Datenstrom wird ein neues Paket immer dann begonnen, wenn eine Frame-Grenze erreicht ist. VDR benötigt dies zum Erkennen und zur Speicherung der Frame-Indizes, und zur vereinfachten Anzeige einzelne Frames im Forward/Back Modus. Der erste zurückgegebene Datenblock soll dabei dem Startpunkt eines *I_FRAME* entsprechen. Alle vorhergehenden Daten werden fallen gelassen.

Reichen die Eingangsdaten zum Remultiplexing nicht aus, wird der Wert NULL zurückgegeben, und *Result* besitzt dann keinerlei Bedeutung mehr. Das erklärt dem Aufrufenden auf mehr Daten, die erst im nächsten Aufruf präsentiert werden können, zu warten. Wird NULL zurückgegeben und *Count* ist ungleich 0, entfernt der Aufrufende die *Count*-Bytes vom Beginn von *Data* vor dem nächsten Aufruf. Auf diese Weise wird *Process* angezeigt, daß es diese Daten überspringen muß.

Alle während dem Aufruf nicht verwendeten Daten, erscheinen beim nächsten Aufruf am Anfang der Eingangsdaten, plus alle weiteren verfügbaren Daten.

Es wird garantiert daß der Aufrufende vor dem nächsten Aufruf von *Process* alle zurückgegebenen Daten verarbeitet. So kann *Process* seinen Rückgabe-Puffer dynamisch reservieren, und sicherstellen daß der Aufrufer keine weiteren Zeiger auf diesen Buffer aufrecht erhält.

5.4 Konfigurationsfiles

Es existieren standardmäßig folgende Konfigurationsfiles

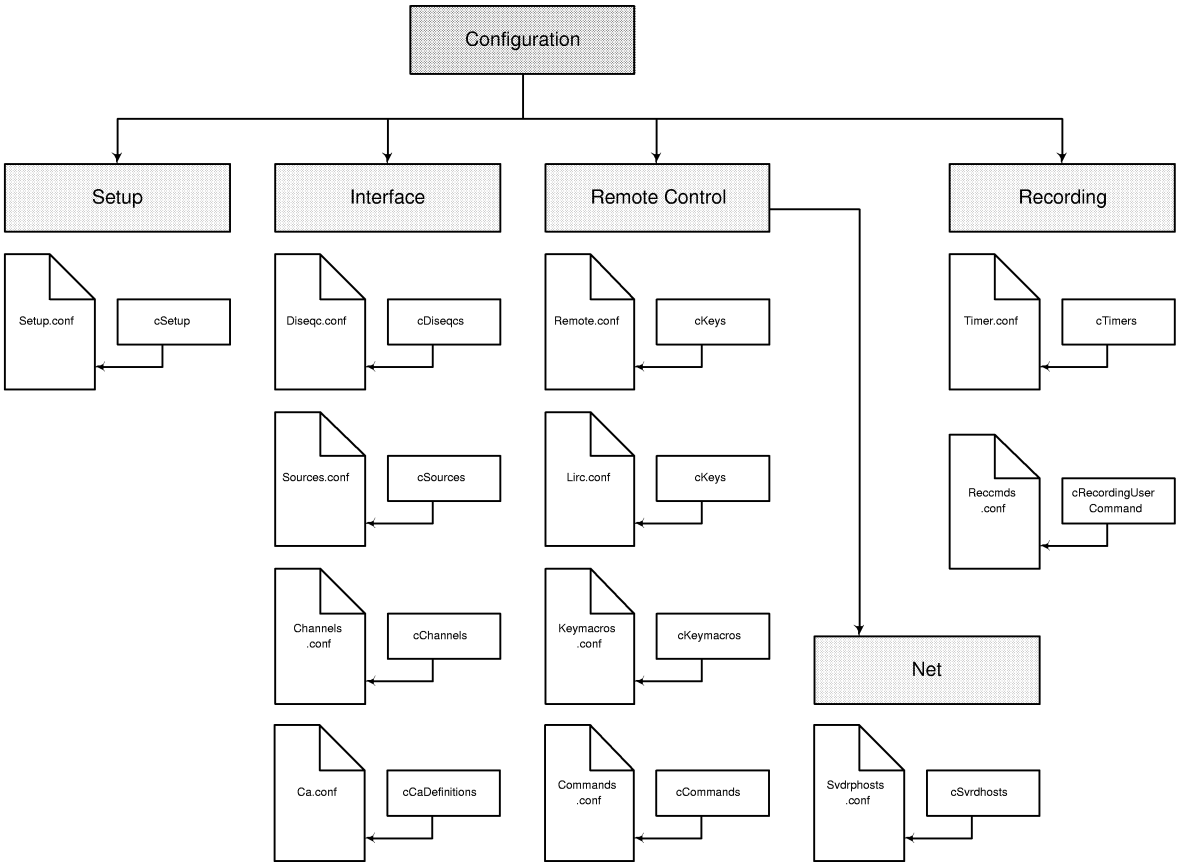

Abbildung 4: Konfigurationsdateien

5.4.1 Ca.conf

Die die Datei *Ca.conf* enthält die Daten zur Konfiguration des Zugriffs auf verschlüsselte Kanäle, den *conditional Access*.

Mit der Verwendung eines CI-Moduls[6] für eine DVB-Karte und einer entsprechenden Abo-Karte, können auch verschlüsselte Sendungen betrachtet werden. Das verwendete Verschlüsselungsverfahren muss im OSD für jede Karte ausgewählt werden.

Die in der Datei *Ca.conf* vorgesehenen Werte entsprechen dem vorletzten Wert jeder Zeile in der *Channels.conf*.

Die Datei *Ca.conf* definiert die im „*Conditional Access*“-Feld verwendeten Werte, und fügt eine Be-

[6]Common-Interface-Modul, Decoder der ein verschlüsseltes Programm mittels vorhandener SmartCard decodiert, vergleichbar mit einer PCMCIA-Karte, wird an den dafür vorgesehenen CI-Steckplatz des Receivers gesteckt

schreibung hinzu.

Beispiel: 101 Premiere World

Die definierenden Zeilen bestehen aus einer Integer-Zahl, gefolgt von einem die Entschlüsselungsmethode beschreibenden Text. Dieser enthält im Allgemeinen den Namen des Pay TV Dienstleisters, der diese Dekodierungsvariante nutzt.

Der spezielle Wert 0 bedeutet *Free to Air*, und kann für Kanäle verwendet werden, die keiner zusätzlichen Entschlüsselungshardware bedürfen.

Die Werte 1 bis 4 können für Kanäle verwendet werden, die aus irgendeinem bestimmten Grund eine bestimmte der verfügbaren DVB-Karten verwenden müssen.

Es sind maximal 2 Verschlüsselungen je DVB-Karte möglich.

```
# Conditional Access configuration for VDR
#
# Format:
#
# number  description
#
# Please contact kls@cadsoft.de before assigning a new number
# to a description, in order to keep them unique.

0      Free To Air

# BetaCrypt

101    Premiere World
102    ORF
103    DIGI-Kabel

...
```

5.4.2 Channels.conf

Die mitgelieferte Datei *Channels.conf* für DVB-s enthält eine Senderbelegung des digitalen ASTRA-Programmangebotes, sowie jeweils eine Vorlagedatei für DVB-c und DVB-t.

```
ORF1:12692:h:S19.2E:22000:160:161:165:102:13001:0:0:0
ORF2:12692:h:S19.2E:22000:500:501:505:102:13002:0:0:0
ZDF.info:11954:h:S19.2E:27500:610:620:0:0:28011:0:0:0
```

```
CNN:12168:v:S19.2E:27500:165:100:0:0:28512:0:0:0
Super RTL:12188:h:S19.2E:27500:165:120:65:0:12040:0:0:0
VOX:12188:h:S19.2E:27500:167:136:71:0:12060:0:0:0
Kabel 1:12480:v:S19.2E:27500:511:512:33:0:899:0:0:0
Neun Live:12480:v:S19.2E:27500:767:768:35:0:897:0:0:0
DSF:12480:v:S19.2E:27500:1023:1024:0:0:900:0:0:0
HOT:12480:v:S19.2E:27500:1279:1280:0:0:40:0:0:0
Bloomberg TV Germany:12552:v:S19.2E:22000:162:99:0:0:12160:0:0:0
Bloomberg TV France:11817:v:S19.2E:27500:163:92:0:0:8004:0:0:0
Bloomberg TV Spain:12168:v:S19.2E:27500:167:112:0:0:12721:0:0:0
Sky News:12552:v:S19.2E:22000:305:306:0:0:3995:0:0:0
Fox Kids Netherlands:12574:h:S19.2E:22000:163:92:0:0:5020:0:0:0
Alice:12610:v:S19.2E:22000:162:96:0:0:12200:0:0:0
n-tv:12670:v:S19.2E:22000:162:96:55:0:12730:0:0:0
Grand Tourisme:12670:v:S19.2E:22000:289:290:0:0:17300:0:0:0
TW1:12692:h:S19.2E:22000:166:167:0:0:13013:0:0:0
Eurosport:11954:h:S19.2E:27500:410:420:0:0:28009:0:0:0
EinsExtra:12110:h:S19.2E:27500:101:102:0:0:28201:0:0:0
EinsFestival:12110:h:S19.2E:27500:201:202:0:0:28202:0:0:0
EinsMuXx:12110:h:S19.2E:27500:301:302:0:0:28203:0:0:0
ZDF Theaterkanal:11954:h:S19.2E:27500:1110:1120:0:0:28016:0:0:0
ZDF.doku:11954:h:S19.2E:27500:660:670:0:0:28014:0:0:0
MDR:12110:h:S19.2E:27500:401:402:404:0:28204:0:0:0
ORB:12110:h:S19.2E:27500:501:502:504:0:28205:0:0:0
B1:12110:h:S19.2E:27500:601:602:604:0:28206:0:0:0
:Premiere World
Premiere Start:11797:h:S19.2E:27500:255:256:0:101:8:0:0:0
Premiere 1:11797:h:S19.2E:27500:511:512,513;515:0:101:10:0:0:0
Premiere 2:11797:h:S19.2E:27500:1791:1792,1793;1795:0:101:11:0:0:0
Premiere 3:11797:h:S19.2E:27500:2303:2304:0:101:43:0:0:0
Premiere 4:11797:h:S19.2E:27500:767:768:0:101:9:0:0:0
Premiere 5:11797:h:S19.2E:27500:1279:1280:0:101:29:0:0:0
Premiere 6:11797:h:S19.2E:27500:1535:1536:0:101:41:0:0:0
Premiere 7:11797:h:S19.2E:27500:1023:1024:0:101:20:0:0:0
...
```

Zeilenaufbau:

Programmbezeichnung : Transponderfrequenz : Polarisation : DiSEqC-Quelle : Symbolrate : Video-PID : Audio-PID : Teletext-PID : Verschlüsselungsmethode : Service-ID : Network-ID : Transport Stream-ID : Radio-ID.

Die verschiedenen Programmgruppen können durch eine Zeile mit vorangestelltem Doppelpunkt zusammengefasst werden, damit sie im Menü auf kurzem Wege angewählt werden können. Zusätzlich kann hier nach einem „@“ noch die nächste Kanalnummer vorgegeben werden.

5.4.3 Commands.conf

Die Datei *Commands.conf* enthält benutzerdefinierte Befehle, die vom „*Commands*“-Menü aus ausgeführt werden können.

Das bedeutet: Es besteht die Möglichkeit, selbst definierte Befehle über das OSD per Fernbedienung ausführen zu lassen. Dazu muß die Datei *Commands.conf* angelegt werden. Aus dieser lassen sich beliebige Skripte ausführen und deren Ausgaben formatiert im OSD anzeigen.

```
TDSL-Verbindung ein       : sudo /usr/sbin/cinternet -start; echo "PPPD gestartet"
TDSL-Verbindung aus       : sudo /usr/sbin/cinternet -stop; echo "PPPD gestoppt"
```

Den verschiedenen Befehlen wird beim Aufruf automatisch eine laufende Nummer zur Kurzwahl vorangestellt. Durch ein Fragezeichen hinter den korrespondierenden Text, wird vor der Ausführung noch eine Bestätigung verlangt. Die Kommandos werden nur mit der Berechtigung des Users der VDR-Software ausgeführt. Bei Befehlen, die nur durch den Superuser *root* ausgeführt werden können, muß auf die *sudo*-Anweisung zurückgegriffen werden, falls die Software nicht mit der Berechtigung von *root* gestartet wurde.

5.4.4 Remote.conf

Die Datei *Remote.conf* enthält die aktuelle Tastenbelegung für die Tastatur oder eine Fernbedienung.

Beim erstmaligen Aufruf der VDR-Software ist zunächst die PC-Tastatur als Standard für die Bedienung eingestellt. Das Programm geht automatisch in den Lernmodus um die Tasten wuschgemäß zuzuordnen. Wird eine Fernbedienung verwendet und läuft der LIRC-Daemon, können die Signale zur Steuerung der VDR-Funktionen angelernt werden. Während der Lernphase können Korrekturen durchgeführt oder der Vorgang auch vorzeitig beendet werden. Auch lassen sich nicht benötigte Tasten einfach überspringen.

Zusätzlich zu den Standardfunktionen können spezielle Schlüssel für den direkten Zugiff zu einigen Menüpunkten angelernt werden. Genauso lassen sich einige weitere Schlüsselwörter für bestimmte Tasten einer Fernbedienung anlernen. Diesen kann z. B. in den Plugins eine besondere Bedeutung zugewiesen werden.

In der Datei *Remote.conf* sind diese angelernten Zuordnungen gespeichert. Wird die Datei gelöscht, startet die VDR-Software den Lernmodus beim nächsten Aufruf erneut.

Jede Zeile enthält eine Tastenbelegung im folgenden Format:

name.key code

name definiert den Namen der Fernbedienung, z. B. KBD für die PC-Tastatur, RCU für die selbst gebaute *„Remote Control Unit“*, oder LIRC für die *„Linux Infrared Remote Control“*. *key* ist der Name der zu definierenden Taste, z. B. Up, Down, Menu usw.. *code* definiert das Zeichen, daß die Fernbedienung liefert, wenn die entsprechende Taste betätigt wird.

Im folgenden Beispiel sind die für die Grundfunktionen des VDR nötigen Elemente in einer LIRC-Zuordnung aufgeführt.

```
LIRC.Up         UP
LIRC.Down       DOWN
LIRC.Menu       MENU
LIRC.Ok         OK
LIRC.Back       BACK
LIRC.Left       LEFT
LIRC.Right      RIGHT
LIRC.Red        RED
LIRC.Green      GREEN
LIRC.Yellow     YELLOW
LIRC.Blue       BLUE
LIRC.0          0
LIRC.1          1
LIRC.2          2
LIRC.3          3
LIRC.4          4
LIRC.5          5
LIRC.6          6
LIRC.7          7
LIRC.8          8
LIRC.9          9
LIRC.Power      POWER
LIRC.Volume+    VOLUME+
LIRC.Volume-    VOLUME-
LIRC.Mute       MUTE
```

In früheren Versionen (bis 1.1.11) musste, wie im obigen Beispiel, die Tastenbenennung in der *lirc.conf* den VDR-Befehlen angepasst werden. In den aktuellen Versionen ist grundsätzlich eine freie Belegung möglich.

5.4.5 Lirc.conf

In der Datei *Lirc.conf* sind die Tastenbelegungen für mit LIRC betriebene Fernbedienungen enthalten.

5.4.6 Reccmds.conf

Wird im Auswahlmenü der Aufzeichnungen einen bestimmter Eintrag markiert, lassen sich dafür frei definierbare Befehle ausführen. Das Format der *Reccmds.conf* entspricht dabei den Vorgaben für die *Commands.conf*. Aufgerufen wird es bei der markierten Aufnahme durch Drücken einer Zifferntaste.

Im folgenden Beispiel wird nach einer Bestätigung der Name der markierten Aufnahme einer Datei zur späteren Konvertierung dieser Aufnahme hinzugefügt.

```
Konvertierungsliste ergänzen?     : echo $1 >> /video/toconvert.txt
```

5.4.7 Keymacros.conf

Die Datei *Keymacros.conf* ermöglicht die Definition von Kurzbefehlen, die z. B. durch Drücken der Farbtasten direkt während einer Wiedergabe die direkte Ausführung von Menüpunkten erlaubt.

Dabei können verschiedene Tastendrücke hintereinander simuliert werden.

In *Keymacros.conf* sind als benutzerdefinierte Makros enthalten, die immer mit dem Betätigen der entsprechenden Taste ausgeführt werden. Die Definition erfolgt in folgender Syntax:

macrokey [@plugin] key1 key2 key3...

macrokey entspricht der Taste, die die Ausfährung des Makros initiiert. Das kann ein der 4 Sondertasten *Red*, *Green*, *Yellow* oder *Blue*, bzw. *User1* bis *User9* sein. Der Rest der Zeile besteht aus einer Reihe von Tasten, deren Funktionen ausgeführt werden, wenn sie in der entsprechenden Reihenfolge betätigt werden. Hier sind bis zu 15 Tasten insgesamt möglich.

Das optionale *@plugin* kann für die automatische Auswahl eines vorhandenen Plugins aus dem Hauptmenü verwendet werden, vorausgesetzt, daß Plugin hat einen Hauptmenü-Eintrag. *plugin* ist dabei der Name des Plugins. Es kann lediglich ein Plugin pro Makro gewählt werden, und es wird automatisch ein *Ok*-Button in die Makro-Definition eingefügt.

Beispiel: User1 @abc Down Down Ok

Dies würde im Hauptmenü die Funktion „abc“-Plugin aufrufen, und 2 Tastendrücke nach unten veranlassen, gefolgt von einem *Ok*.

Dabei ist zu beachten, daß die farbigen Tasten nur ihre Makro-Funktion ausüben, wenn normal fern gesehen wird, daß heißt, wenn kein anderes Menü geöffnet oder der Player aktiv ist. Die *User1...User9* Tasten führen hingegen immer ihre Makro-Funktion aus.

```
# Remote control key macros for VDR
#
# Format:
#
# macrokey  key1 key2 key3...
#
# See man vdr(5)
```

```
#
Red       Recordings
Green     Schedule Ok
Blue      Timers
Yellow    Commands
```

5.4.8 Diseqc.conf

Die Datei *Diseqc.conf* übernimmt die DiSEqC-Konfiguration, das heißt, die Steuerung einer Anlage mit mehreren Satelliten mittels der Signalisierung nach DiSEqC[7]. Die Angaben dieser Konfigurationsdatei werden nur benötigt, wenn mehr als ein Satellit über z. B. einen entsprechenden Umschalter an einer DVB-Karte betrieben wird.

Für spezielle Fälle sind auch Alternativen zur DiSEqC-Ansteuerung aufgeführt.

In der *Diseqc.conf* sind die DiSEqC-Kontrollsequenzen definiert, die an die DVB-S Karte gesendet werden, um von einer bestimmten Satellitenposition/einem bestimmten Satellitenband zu empfangen.

Beispiel

S19.2E 11700 V 9750 t v W15 [E0 10 38 F0] W15 A W15 t

Das erste Wort der Parameterzeile muß ein in der Datei *Sources.conf* definierter Code sein, und bezeichnet den Satelliten von dem empfangen werden soll.

Es folgt die Umschaltfrequenz des LNB (*slof*), also die Transponderfrequenz oberhalb der empfangen werden soll. Typischerweise existiert lediglich eine *slof* pro LNB, aber die Syntax erlaubt die Definition einiger verschiedener Frequenzbereiche.

Der dritte Parameter definiert die Polarisation. Diese kann entweder H für horizontal oder V für vertikal sein.

Der vierte Parameter gibt die „local oscillator frequency“, kurz *lof*, des LNB's für den verwendeteten Frequenzbereich an. Dieser Wert wird von der aktuellen Transponderfrequenz abgezogen, wenn auf den entsprechenden Kanal umgeschaltet wird.

Der Rest der Zeile enthält aktuelle DiSEqC-Sequenzen.

```
# DiSEqC configuration for VDR
#
# Format:
#
# satellite slof polarization lof command...
#
# satellite:      one of the 'S' codes defined in sources.conf
```

[7] Digital Satellite Equipment Control

```
# slof:           switch frequency of LNB; the first entry with
#                 an slof greater than the actual transponder
#                 frequency will be used
# polarization:   V = vertical, H = horizontal
# lof:            the local oscillator frequency to subtract from
#                 the actual transponder frequency
# command:
#   t         tone off
#   T         tone on
#   v         voltage low (13V)
#   V         voltage high (18V)
#   A         mini A
#   B         mini B
#   Wnn       wait nn milliseconds (nn may be any positive integer number)
#   [xx ...]  hex code sequence (max. 6)
#
# The 'command...' part is optional.
#
# Examples:
#
# Full DiSEqC sequence:
#
S19.2E  11700 V  9750  t v W15 [E0 10 38 F0] W15 A W15 t
S19.2E  99999 V 10600  t v W15 [E0 10 38 F1] W15 A W15 T
S19.2E  11700 H  9750  t V W15 [E0 10 38 F2] W15 A W15 t
S19.2E  99999 H 10600  t V W15 [E0 10 38 F3] W15 A W15 T

S21.5E  11700 V  9750  t v W15 [E0 10 38 F4] W15 B W15 t
S21.5E  99999 V 10600  t v W15 [E0 10 38 F5] W15 B W15 T
S21.5E  11700 H  9750  t V W15 [E0 10 38 F6] W15 B W15 t
S21.5E  99999 H 10600  t V W15 [E0 10 38 F7] W15 B W15 T

...
```

5.4.9 Sources.conf

Der VDR arbeitet grundsätzlich mit den verschiedenen DVB-Karten für Kabel (DVB-c), terrestrischen Empfang (DVB-t) und Empfang über Satellit (DVB-s) zusammen.

In den Menüeinstellungen kann die Quelle eines Kanals festgelegt werden. Diese Daten werden aus der Datei *Sources.conf* gelesen und finden sich dann entsprechend in der *Channels.conf* wieder. Das bedeutet, daß ein als „terrestrisch“ markierter Kanal über eine DVB-T-Karte empfangen wird.

Die Datei *Sources.conf* definiert die Codes, die im Source-Feld der Kanäle in der Datei *Channels.conf* verwendet werden, und fügt ihnen eine Beschreibung hinzu.

Beispiel: S19.2E Astra 1

Das erste Zeichen muß entweder eine S für Satellite, ein C für Cable oder ein T für Terrestrial,

entsprechend dem Empfang, sein. Die folgenden Daten betreffen dann die gewählte Quelle. Für das Beispiel bedeutet das: die Orbitposition in Grad, gefolgt von einem E für East oder einem W für West.

```
# Sources configuration for VDR
#
# Format:
#
# code  description
#
# S (satellite) xy.z (orbital position in degrees) E or W (east or west)
# Note: only the first part is actually used by VDR. The description part
# is for the "human" interface for clarity.
#
# '&' means same orbital position but different host company.
# '/' means same (or very little deviation) orbital position & host.
# A value in () means this satellite is still in it's test phase.
#
# Please contact kls@cadsoft.de before assigning a new code
# to a description, in order to keep them unique.
#
# Satellites

S5E     Sirius 2/3
S7E     Eutelsat W3
S10E    Eutelsat W1R
S13E    Hotbird 1-(5)-6
S16E    Eutelsat W2
S19.2E  Astra 1B/C/E/F/G/H/2C

...

# Cable

C       Cable

# Terrestrial

T       Terrestrial
```

5.4.10 Svdrphosts.conf

Mit dem Simple Video Disk Recorder Protocol (SVDRP) kann die VDR-Software auch über das Netzwerk gesteuert werden. Dazu werden Programme wie *kvdr* und *vdradmin* benötigt. Allerdings kann lediglich ein Programm über diese Schnittstelle zugreifen.

In der Datei *SVDRPhosts.conf* werden eine Reihe verschiedener Host-Adressen angegeben, denen es möglich ist mit dem SVDRP-Port des VDR zu verbinden. Als Standard ist zunächst nur der Zugriff von *localhost* mit der IP-Adresse 127.0.0.1 freigegeben. Im weiteren können beliebige Adressen eingestellt werden. So ist beispielweise auch die Freigabe ganzer Subnetze, wie im Beispiel Subnetz 192.168.1.0 mit der Netzmaske 255.255.255.0, möglich. Jede Zeile enthält eine IP-Adresse im Format *IP-Address[/Netmask]*, wobei die Netzmaske optional ist.

In der Voreinstellung wird Port 2001 zur Steuerung genutzt. (telnet localhost 2001)

Auszug aus *Svdrphosts.conf* (Zeichen hinter einer Raute gelten als Kommentar.):

```
#
# svdrphosts    This file describes a number of host addresses that
#               are allowed to connect to the SVDRP port of the Video
#               Disk Recorder (VDR) running on this system.
# Syntax:
#
# IP-Address[/Netmask]
#
#
127.0.0.1             # always accept localhost
192.168.1.0/24        # any host on the local net
# 204.152.189.113     # a specific host
# 0.0.0.0/0           # any host on any net (USE THIS WITH CARE!)
```

5.4.11 Make.config template

Ein benutzerdefiniertes Makefile mit Optionen für den VDR, C-Compiler und die Verzeichnis-Umgebung.

5.4.12 Reccmds.conf

Die neue Konfigurationsdatei *Reccmds.conf* kann zur Definition von Befehlen verwendet werden, die aus dem *Recordings*-Menü heraus ausgeführt werden sollen. Die Syntax entspricht der für *Commands.conf* beschriebenen.

5.4.13 Timers.conf

Die Datei *Timers.conf* enthält das Timer-Setup. Jede Zeile enthält eine Timer-Definition, mit indviduellen Feldern, die durch das Zeichen „:“ voneinander getrennt sind.

Beispiel

```
1:10:-T-----:2058:2150:50:5:Quarks \& Co:
```

Die Felder einer Timer-Definition besitzen folgende Bedeutung (von links nach rechts zu lesen):

Status - Definiert, ob der Timer inaktiv (inactive - 0) oder aktiv (active - 1) ist. Der Wert 3 wird für momentane Aufnahmen vergeben (instant - 3). Andere Werte können von externen Programme vergeben werden, z.B. zur Markierung aktiver Timer und der Erkennung, ob diese vom Nutzer verändert wurden. Modifiziert ein Nutzer einen aktiven Timer, so wird das *Status*-Feld explizit auf „1“ gesetzt (oder „0“, wenn der Nutzer den Timer deaktiviert).

Um zukünftig die Erweiterbarkeit zu erlauben, sollten externe Programme nur die oberen 16 bis 32 bit des *Status*-Parameters nutzen und die unteren 16 bit unangetastet lassen.

Channel - Der Kanal, von dem aufgenommen werden soll. Dabei handelt es sich um die Kanalnummer, wie sie auch im OSD-Menü angezeigt wird, oder eine komplette *Channel ID*. Beim Lesen der *Timers.conf* werden einige Kanal-Nummern auf die entsprechenden *Channel ID*s abgebildet. Wird die Datei neu geschrieben, existieren so lediglich die genannten *Channel ID*s. Kanalnummern werden als Eingabe akzeptiert, um bei einer manuellen Bearbeitung der *Timers.conf* eine einfachere Erzeugung von Timern zu ermöglichen. Auch bei der Auflistung der Timer über SVDRP-Befehle, werden die Kanäle als Nummern übergeben.

Day - Der Tag an dem der Timer die Aufnahme starten soll. Hierbei handelt es sich um einen „singleshot“ Timer, der den Tag im Monat definiert, an dem der Timer aufnehmen soll. Der Wert dieser Zahl muß also im Bereich 1 bis 31 liegen.

Im Falle eines periodischen Timers handelt es sich um eine Zeichenkette mit genau sieben Zeichen. Dabei korrespondiert jede Zeichenposition mit einem Wochentag, wobei Montag als erster Tag angesehen wird.

Das Zeichen „-“ auf einer Position bedeutet, daß die Aufnahme an diesem Tag ausgelassen wird. Jedes andere Zeichen veranlasst die Aufnahme an diesem Tag. Das könnte z. B. so aussehen: *MTWTF–*

Damit wird ein Timer definiert, der von Montag bis Freitag aufnimmt, und das Wochenende dabei auslässt. Das selbe Ergebnis könnte auch durch die Zeichenkette *ABCDE–* erzielt werden.

Der Definition des Aufnahmetages eines sich wiederholenden Timers könnte das erste Aufnahmedatum folgen. Das Format sieht folgendenmaßen aus: *YYYY-MM-DD*,

Eine vollstandige Defintion würde also wie folgt aussehen: *MTWTF–@2003-02-18*. Das implementiert einen Timer, der beginnend am 18. Februar 2003, von Montag bis Freitag aufnimmt.

Dieses Feature *first day* kann auch dazu verwendet werden, einen Timer für ein paar Tage zu deaktivieren. Das Datum muß nicht mit einem Wochentag übereinstimmen, an dem der Timer wirklich aufnimmt. Im Beispiel bedeutet das, es könnte auch der Samstag als *first day* angegeben werden, ohne, daß am Samstag tatsächlich eine Aufnahme stattfindet.

Start - Eine 4-stellige Integer-Zahl, die angibt wann der Timer mit der Aufnahme beginnen soll. Das Format sieht wie folgt aus: *hhmm*, so daß *1430* halb drei nachmittags bedeutet.

Stop - Eine 4-stellige Integer-Zahl, die angibt, wann der Timer die Aufnahme stoppt. Das Format gleicht dem der Startzeit.

Priority - Ein Integer-Wert im Bereich 0 bis 99, der die Priorität des Timers und der von ihm gemachten Aufnahmen definiert. 0 repräsentiert dabei den kleinsten Wert, 99 den höchsten. Die Priorität wird für die Entscheidung genutzt, welcher Timer gestartet wird, falls 2 oder mehr Timmer zu exakt der selben *Start*-Zeit aufnehmen sollen. Dabei wird der erste Timer in der Liste, mit der höchsten Priorität gewählt. Dieser Wert wird mit der Aufnahme gespeichert und später dafür verwendet, zu entscheiden welche Aufnahme von der Festplatte gelöscht wird um Platz für neue Aufnahmen zu schaffen. Ist die Festplatte voll und eine neue Aufnahme benötigt weiteren Speicherplatz, wird dementsprechend die Aufnahme mit der niedrigsten Priorität, sowie überschrittener Lebensdauer (*lifetime*) gelöscht.

Sind alle verfügbaren DVB-Karten ausgelastet, kann eine Timer mit höherer Priorität, einen Timer niedrigerer Priorität unterbrechen, um seine Aufnahme zu starten.

Lifetime - Die garantierte Lebensdauer, in Tagen, einer Aufnahme wird von ihrem Timer erzeugt. 0 bedeutet, daß diese Aufnahme jederzeit automatisch durch eine Aufnahme höherer Priorität gelöscht werden kann. 99 definiert, daß diese Aufnahme niemals automatisch gelöscht werden kann. Jede andere Nummer zwischen 1 und 98 bedeutet, daß diese Aufnahme nicht von einer anderen gelöscht werden kann, bevor nicht die genannte Anzahl an Tagen, ausgehend von der *Start*-Zeit vergangen ist.

File - Der Datei-Name, der der Aufnahme vom Timer zugewiesen wird. Enthält der Name ein „:“, muß dies durch ein **|** ersetzt werden. Soll der Name Unterverzeichnisse enthalten, müssen diese durch ein **~** getrennt angegeben werden.

Die speziellen Schlüsselwörter *Title* und *Episode* werden durch den Titel und die Folge, die aus den Informationen der EPG-Daten zur Zeit der Aufnahme entnommen werden, ersetzt. Sind diese nicht verfügbar, wird *Title* durch den Kanal-Namen ersetzt und *Episode* frei gelassen.

Summary - beliebiger Text der den Inhalt der Aufnahme beschreibt. Der Eintrag der Zusammenfassung wird ebenfalls vom Timer vorgenommen.

Jedes Zeichen, das einen Zeilenumbruch festlegt, muß durch ein **|** ersetzt werden. Dafür darf die Zusammenfassung, das Zeichen **:** enthalten. Wird das *Summary*-Feld gefüllt, wird diese in die Datei *Summary.vdr* der Aufnahme geschrieben.

5.4.14 Setup.conf

In der Datei *Setup.conf* sind alle aktuellen benutzerdefinierten Einstellungen verzeichnet. Die Basis-Konfigurationen sind dabei im Format *Name* = *Value* enthalten.

5.4.15 epg.data

Bei *Epg.data* handelt es sich um keine wirkliche Konfigurationsdatei. Diese Datei wird mit dem Programmstart gelesen um die Ergebnisse früherer EPG-Scans wieder herzustellen.

Epg.data enthält EPG-Daten in einem einfach zu analysierenden Format.

Der erste Buchstabe jeder Zeile definiert, was für eine Art von Daten, die Zeile enthält. Dabei sind folgende Buchstaben definiert:

- C <channel id> <channel name>

- E <event id> <start time> <duration> <table id>
- T <title>
- S <subtitle>
- D <description>
- e @
- c @

Kleingeschriebene Buchstaben markieren das Ende eine Sequenz, die von einem entsprechenden Großbuchstaben begonnen wurde. Der äußere Rahmen besteht aus einer Sequenz mehrerer „C“...“c“ (Channel) Eingaben. In diesem sind ein Zahl von „E“...“e“ (Event) Eingaben erlaubt. Die Eingabe von „T“, „S“ und „D“ ist optional (obwohl jedes Eventmindestens einen „T“ Eintrag besitzen sollte).

- <channel id> @„Channel ID“, zusammengesetzt aus aus in der *channels.conf* definierten Parametern
- <channel name> @„Name“ wie in *channels.conf* (nur zur Information, kann auch weggelassen werden)
- <start time> @Zeit (als time_t Integer) in UTC wenn das Event startet
- <duration> @Zeit (in Sekunden) die das Event dauern wird
- <table id> @Hexadezimal-Zahl, die angibt, daß das Event enthalten ist
- in (wird dies leer gelassen oder ist 0, wird dieses Event nicht überschrieben
- or verändert durch die Daten vom DVB Stream)
- <title> @Titel des Events
- <subtitle> @Untertitel (z. B. Name der Seria oder ähnliches)
- <description> @Beschreibung des Events (jedes ’|’ Zeichen wird als neue Zeile interpretiert)

5.5 Fernbedienung

Der VDR kann grundsätzlich über die PC-Tastatur bedient werden. Steht eine Fernbedienung zur Verfügung kann im Remote-Makro zwischen zwei Einstellungen gewählt werden:

remote = RCU oder
remote = LIRC

Letztendlich ist auch noch mit Tools wie *vdradmin* oder *kvdr* die Fernbedienung über das eigens dafür programmierte *Simple Video Disk Recorder Protocol* möglich. Dies ermöglicht den Zugriff auf den VDR über das Netzwerk.

Zusammenfassend übernehmen folgende Quellcode-Datein das Handling der Fernbedienung

- Remote.c - Allgemeines Handling der Fernbedienung
- Keys.c - Handling der Tastenbelegung der Fernbedienung
- Lirc.c - Bedienung über die „Linux Infrared Remote Control“ (http://www.lirc.org)
- Rcu.c - Bedienung über die „Remote Control Unit“ (http://www.cadsoft.de/vdr/remote.htm)

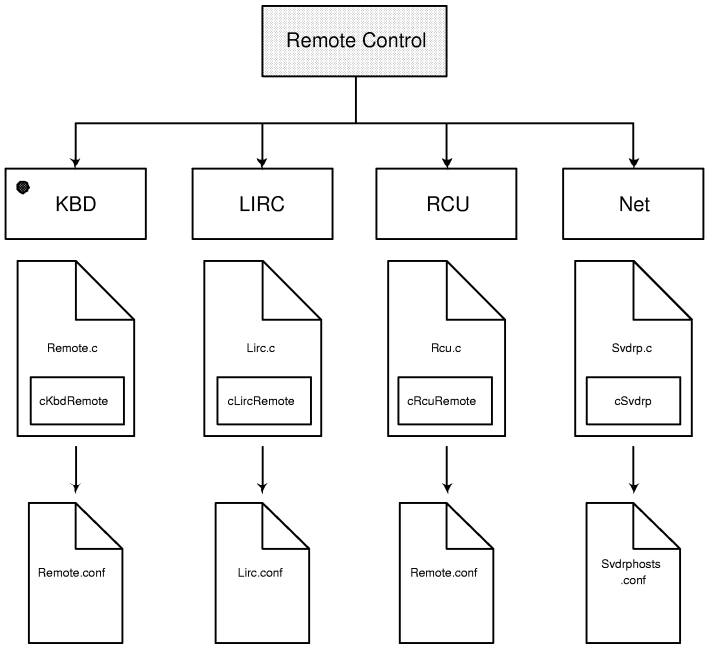

Abbildung 5: Remote Control

5.5.1 Linux Infrared Remote Control

LIRC ist ein Packet, das die Dekodierung und das Senden von Infrarot-Signalen der üblichen Fernbedienungen erlaubt.

Frühere Versionen fokussierten die selbstgebastelte Hardware, verbunden mit einem seriellen oder per Parallelport. Aktuelle LIRC-Versionen unterstützen allerdings auch verschiedenste Hardware. Bastelanleitungen, eine Up-to-date-Liste aller unterstützten Geräte, sowie der Status entsprechender Treiber finden sich beispielsweise unter www.lirc.org.

Der bedeutendste Teil von LIRC ist der *lircd* Daemon. Dieser dekodiert die IR-Signale der Geräte-Treiber und unterstützt die Socketinformationen. Außerdem akzeptiert er auch Kommandos für IR-Signale, falls die Hardware das unterstützt. Das zweite Daemon-Programm, *lircmd*, arbeit zusammen mit *lircd* und übersetzt die dekodierten IR-Signale in Mausbewegungen.

Im folgenden werden die grundlegenden Programme des Paketes kurz vorgestellt.

lircd - LIRC Daemon - Dekodiert Infrarot-Signale und stellt ein einheitliches Interface für Client-Applikationen zur Verfügung. Clients können über einen UNIX Domain Socket nutzen. Dies ermöglicht ihnen das Empfangen von Infrarot-Codes über *lircd*, sowie das Senden an lircd.

lircmd **-** LIRC Maus Daemon **-** übersetzt Infrarot-Signale zu Maus-Events

Dieser Daemon ist in der Lage eine Maus zu simulieren (MouseSystems, IntelliMouse oder IMPS/2 Maus). Es erhält die empfangenen Tastendrücke über lircd und setzt diese in Maus-Events um. Um das sicherzustellen, benötigt *lircmd* eine Konfigurationsdatei, in der festgelegt wird, welche Taste eine

Mausbewegung oder einen Mausklick verursachen soll.

So kann auf Wunsch eine IMPS/2-kompatible Maus simuliert und unter X11 neben jeder gewöhnlichen Maus als alternatives Eingabegerät genutzt werden. Das IMPS/2-Protokoll unterstützt dabei auch die Simulation von Mausrädern, mit denen sich Zusatzfunktionen wie Scrollen realisieren lassen.

irexec - Führt Programme ohne Tastendruck aus

Dieses Programm ermöglicht die Ausführung beliebiger Befehle über ein IR-Signal durch *lircd*, den LIRC Daemon, und dient damit als Ausgangspunkt für das Starten von Programmen.

Clients werden von der Konfigurationsdatei /~.lircrc, die im Home-Verzeichnis des Anwenders liegen muss, verwaltet. Einträge in dieser Datei legen fest, was bei Betätigung einer bestimmten Taste zu geschehen hat. In Ermangelung an Tasten für zu viele gewünschte Funktionen, ist es möglich für zu startende Applikationen unterschiedlichen Betriebsmodi zu definieren, so daß Tasten mit mehreren Funktionen belegt werden können, die abhängig vom Modus ausgeführt werden.

Es wird *irexec* ein Befehlszeilenparameter übergeben, der mit dem Namen einer gültigen Konfigurationsdatei übereinstimmen muß. Werden keine Parameter übergeben, liest *irexec* die Standard-Konfigurationsdatei, die sich üblicherweise in *~/.lircrc* befindet. Führt *irexec* ein Programm aus, wartet es bis dies beendet wird.

ircat - Ausgabe von Strings beim Betätigen von Tasten

Dieses Programm gibt Konfigurationsstrings auf den Standardausgang aus. Das kann zur Ausgabe der Eingänge der Fernbedienung in Skripte, und zum Debuggen der *.lircrc*-Datei verwendet werden. Das Argument zum Programm ist der Programmname, wie er auch in der Eingaben in *.lircrc* erscheint.

irpty - Pseudo-tty-Treiber

Irpty verbindet zu *lircd* um die IR-Codes zu empfangen und in Tastenanschläge umzuwandeln. Natürlich muß vorher eine entsprechende Konfigurationsdatei erzeugt werden.

irxevent - Sender für Infrarot-X-Events

Irxevent ist ein Programm, das Tastatur-Events bzw. Events, die durch das Betätigen der Buttons einer Fernbedienung erzeugt werden, an X-Applikationen sendet. Vorraussetzung ist natürlich, daß die Fernbedienung LIRC verwendet.

So können CD/Mp3-Player oder Fernsehempfänger und andere X-Applikationen, die auf Tastatur oder Mauseingaben reagieren, gesteuert werden.

Irxevent stellt eine Ergänzung zu *irexec* und *irpty* dar.

Ein Programm mit dessen Hilfe man an jede beliebige X11-Anwendung Events verschicken kann. Die am häufigsten benötigten Kommandos, werden lediglich in die Konfigurationsdatei eingetragen.

Das Anklicken von Schaltflächen kann ebenfalls simuliert werden.

Irxevent arbeitet mit der selben Konfigarationsdatei wie *irexec* und *irpty* (in *~/.lircrc*). Ein Teil der *.lircrc*-Datei könnte beispielsweise so aussehen:

```
begin
        prog = irxevent
        button = VIDEO_UP
        config = Key SHIFT-KP_Add CurrentWindow
end
begin
        prog = irxevent
        button = VIDEO_DOWN
        config = Key SHIFT-KP_Subtract CurrentWindow
end
begin
        prog = irxevent
        button = STOP
        config = Key ctrl-c CurrentWindow
end
```

Das bedeutet, die Konfigurationszeilen sollten so aussehen

```
config = Key [shift-][ctrl-][alt-]<key> <windowname> | WindowID id
        | CurrentWindow | RootWindow
config = Button <button> <x> <y> <windowname> | CurrentWindow
config = xy_Key <x> <y> [shift-][ctrl-][alt-]<key> <windowname>
        | WindowID id | CurrentWindow | RootWindow

/*
 * Keyname - Tastensymbol/-name
 * Windowname - Fensternamen
 * CurrentWindow - aktives Fenster
 * RootWindow - Hauptfenster
 * Window-ID - Dezimalzahl, wird benötigt um an die Koordinaten der
 * einzelnen Buttons zu gelangen
 */
```

irrecord - Applikation für die Aufnahme der IR-Codes, für die Verwendung mit LIRC

Dieses Programm nimmt die Signale der Fernbedienung auf und erzeugt daraus eine Konfigurationsdatei für *lircd*. Diese Konfigurationsdatei ist vermutlich der entscheidendste Teil dieses Paketes, so daß auf die Erzeugung einer funktionsfähigen Datei besonderer Wert gelegt werden muß.

Existiert diese allerdings und enthält bereits eine zulässige Konfiguration, nutzt *irrecord* die aufgefundene Protokoll-Beschreibung und nimmt lediglich die Tasten auf. Dies funktioniert natürlich nur, wenn die Fernbedienungen dasselbe Protokoll nutzen. Es existieren allerdings für die meisten Protokolle im Verzeichnis *remotes/generic/* des Paketes bereits Template-Dateien. Der Name der neuen Datei wird aus dem gegebenen Dateinamen durch Anhängen des *.conf* ereugt.

irw **-** sendet Daten vom Unix Domain Socket an *stdout*.

Irw verbindet mit jedem beliebigen Unix Domain Socket und gibt die empfangenen Daten an *sdtout* weiter. Wird kein Argument für den Socket-Namen übergeben, wird dieser in */dev/lircd* nachgeschlagen. Das Ganze erweist sich als durchaus nützlich bei der Fehlersuche.

XMODE2 *Mode2*, *SMode2*, *XMode2* **-** zeigen die Impulselänge der Infrarotsignale an.

Der hauptsächliche Zweck dieses Programms ist, die Funktionsweise eigens gebauter LIRC-Empfänger zu überprüfen,und die IR-Frequenzgänge der Fernbedienung ohne ein Oszilloskop zu visualisieren. So bietet es sich zur Fehlersuche an. Das Programm läuft allerdings nicht mit Hardware, die die Signale bereits selbst codiert, wie beispielsweise TV-Karten.

Mode2 liefert eine einfache Ausgabe von Impulslängen an *stdout*.

SMode2 verwendet die *svgalib* zur Anzeige des IR-Frequenzganges in einer graphischen Repräsentation. Der Zeitmultiplex ist variabel zwischen Werten von 1ms pro Einheit bis zu extrem hohen Werten (Integer), allerdings nie größer als 20ms pro Einheit, da eine Impulsdauer etwa 1ms beträgt. Diese Art und Weise der Präsentation bietet einen spannendere Visualisierung als eine einfache Impulse Ausgabe wie bei *Mode2*.

XMode2 basiert auf *SMode2* von Sinkovics Zoltan. Es handelt sich hierbei lediglich um eine Konvertierung von SVGA in X, mit grundlegender Unterstützung zur Skalierung. Der einzige Unterschied, ist die Möglichkeit, daß hier noch wechselnde Anzeigemodi gewählt werden können.

rc - Das Basis-LIRC-Programm zum Senden von Infrarot-Kommandos

Es kommuniziert mit dem *lircd*-Daemon für das Senden eines oder mehrerer CIR[8]-Befehle. Diese Funktion kann für die Fernbedienung von elektronischen Geräten, wie Fernseher, HiFi-Anlagen u.ä. verwendet werden.

Rc verwendet ebenfalls die *lircd*-Konfigurationsdatei, in der alle Fernbedienungen mit den entsprechenden Codes und den dazugehörigen Zeit- und Frequenzdetails enthalten sein sollten.

Um den PC zu einer fernbedienbaren Hi-Fi-Anlage ungestalten wird lediglich eine freie serielle Schnittstelle, eine gewöhnliche Infrarot-Fernbedienung und die passende Software für Linux oder Windows benötigt.

LIRC ist auf der LIRC-Projekt-Homepage www.lirc.org zu finden. Dieses Paket enthält entsprechende Treiber, das Systemprogramm zur Dekodierung der Infrarotsignale und Client-Anwendungen, die auf Tastendrücke reagieren.

Das Modul gibt empfangene Signale als Puls- und Pausenlängen über ein „character device“ aus. Dazu misst der Treiber die Abstände zwischen Interrupts, die der Zustandswechsel der „Data Carrier

[8]Consumer Infra-Red

Detect“-Leitung (DCD) am seriellen Port auslöst. Auf PC-Seite werden durch den Treiber alle seriellen Schnittstellen mit einem 16450/16550A-UART-kompatiblen Chipsatz unterstützt. Die Hauptaufgabe des Treibers ist das korrekte Timing, da bei einigen IR-Protokollen Pulslängen auftreten können, die kürzer als 100 Mikrosekunden sind. Da Auflösung der Systemuhr für solche Anwendungen nicht geeignet ist, verwendet Treiber für Zeit-Messungen die Linux-Kernelfunktion *do_ gettimeofday()*, deren Auflösung Mikrosekunden-Genauigkeit erreicht. Zur Signalauswertung werden Interrupts verwendet, was die CPU entlastet.

5.5.2 Remote Control Unit

Die Remote Control Unit, kurz RCU, ist eine selbstgebaute Fernbedienung. Sie besteht hauptsächlich aus einem Infrarot-Empfänger und einem vierstelligen Display, gesteuert von einem PIC 16C84.

Die gedruckte Schaltung wurde so entworfen, daß sie in einen etwa 5.25“ Laufwerksschacht passt. Um die Herstellung des Boards zu vereinfachen, wurden alle Leiterbahnen so entworfen, daß sie als Drahtbrücken implementiert werden können, es existiert also nur eine Schicht. Die Daten werden zwischen RCU und dem PC durch einen IrDa-Connector übertragen, der heutzutage auf den meisten Boards integriert ist. Dieser Connector unterstützt außerdem die +5V-Spannungsversorgung.

Die kompletten Baupläne für die RCU und das Board-Layout können unter *Download* auf *www.cadsoft.de/vdr* heruntergeladen werden.

5.6 OnScreen Display

Das OSD ist ein, auf dem Display der DVB-Karte basierendes, Menü zur Steuerung des VDR. Folgende C-Files enthalten die dafür benötigten Klassen.

- *Menu.c* - Aktuelle Menü-Imlementation
- *Menuitems.c* - Universelle Menüelemente
- *Osd.c* - Abstrakte OSD-Schicht
- *Osdbase.c* - Basis-Schnittstelle zum OSD
- *Dvbosd.c* - Implementation des DVB-Onscreen Displays
- *Font.c/font.h/fontfix.c/fontosd.c/Genfontfile.c* - Schriftenhandling
- *Interface.c* - Schicht für eine abstrakte Benutzerschnittstelle

Die folgenden elf Screenshots sollen einen kurzen Überblick über das Benutzerinterface, das OSD, geben.

Das ***Hauptmenü*** erlaubt einen Zugriff auf die primären VDR-Funktionen, wie die „programme schedules“ (EPG), Kanäle, Timer, Aufnahmen, System-Setup und sogar auf benutzerdefinierte Linux-Shell-Befehle.

Das Betätigen der roten Taste der Fernbedienung ermöglicht die sofortige Aufnahme der aktuell laufenden Sendung.

Die Titelzeile zeigt die aktuelle Speicherplatzauslastung (Plattenauslastung 7%) und der voraussichtlich verbleibende Speicherplatz in Stunden und Minuten (20 Stunden und 12 Minuten für dieses 40 GB System).

Abbildung 6: Hauptmenü

Programme Schedule - Die rote Taste erlaubt die einfache Programmierung eines Timers, zur Aufnahme des hervorgehobenen Programms. Die grüne und gelbe Taste dient zum Umschalten auf die „What's on now?“ und „What's on next?“ Screens.

Abbildung 7: Programme

Wird während dem Fernschauen die *OK*-Taste betätigt, wird das ***Channel Display*** angezeigt. Darauf ist der aktuelle Kanal, Datum und Uhrzeit, sowie das gerade laufende und folgende Programme angezeigt.

Abbildung 8: Kanäle

Die ***Programmbescheibung*** gibt eine ausführliche Beschreibung des laufenden Programmes. Dieser Menüpunkt kann durch ein *OK* auf die im Programmmenü angezeigte Sendung aufgerufen werden.

Das ***„What's on now?“***-Menü listet die aktuell laufenden Sendungen aller Kanäle auf. Durch Betätigung des Buttons kann direkt zu einer anderen Sendung gewechselt werden. Über den roten Button kann der Timer zur Aufnahme der Sendung programmiert werden.

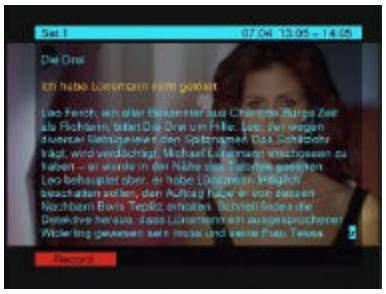

Abbildung 9: Programmbeschreibung

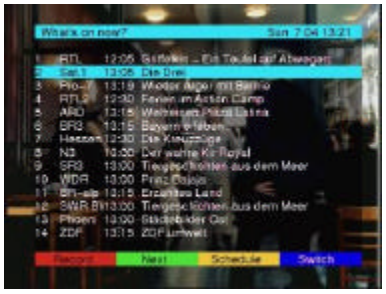

Abbildung 10: What's on now?

Das ***Kanalmenü*** erlaubt das Umschalten zwischen den Kanälen, das Editieren der Kanaleinstellungen, das Hinzufügen neuer Kanäle oder auch das Löschen bereits existierender, sowie das Ändern der Kanalfolge.

Abbildung 11: Kanal-Menü

Das Menü zur ***Kanaleditierung***, ermöglicht die Einstellung alle Kanalparameter.

Auf die Liste aller Timer kann über das ***Timer Menü*** zugegriffen werden. Es zeigt die Kanalnummer und den Tag an dem die Aufnahme erfolgen soll, die Start- und Stopzeit und den Namen den die Aufnahme erhalten soll.

Die Liste ist chronologisch geordnet, so daß der als nächstes startende Timer am oberen Ende steht.

Die Raute an der linken Seite zeigt an, ob der Timer inaktiv, aktiv oder bei der Aufnahme ist, oder aber mit der Aufnahme erst nach einem gegebenen „ersten Tag“ Datum beginnt.

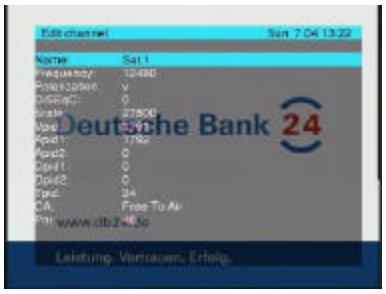

Abbildung 12: Kanal-Editierung

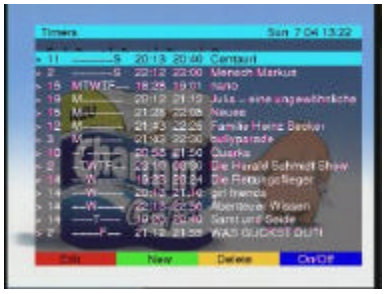

Abbildung 13: Timer Menü

Das Menü zur ***Timer-Editierung*** ermöglicht die manuelle Einstellung verschiedener Timer-Parameter.

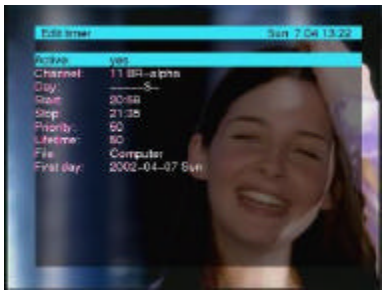

Abbildung 14: Timer Editierung

Die ***Aufnahmeliste*** enthält alle Aufnahmen des Timers. Besitzt ein Eintrag Datum und Zeit in den zwei linken Spalten, muß lediglich die rote Taste oder *OK* gedrückt werden um die Aufnahme abzuspielen. Sind dort lediglich einfache Zahlen angegeben, handelt es sich um ein Aufnahmeverziechnis, daß ebenfalls über die rote Taste oder *OK* geöffnet werden kann. In diesem Fall gibt die erste Nummer die Gesamtzahl der Aufnahmen in diesem Verzeichnis an, die zweite hingegen die Anzahl neuer Aufnahmen.

Wird während dem Abspielen einer Aufnahme *OK* gedrückt, erscheint das ***Progress Display***. Dieses zeigt Datum, Uhrzeit und Name der Aufnahme an. Weiterhin kann die aktuelle Position in der Aufnahme und ihre Gesamtlänge angezeigt werden.

Abbildung 15: Aufnahmeliste

Abbildung 16: Progress Display

Hier wird gezeigt, daß bereits einige Editier-Markierungen definiert werden, die es erlauben eine aufbereitete Version der Aufnahme zu erzeugen, aus der beispielsweise alle Werbespots herausgeschnitten sind.

5.6.1 Osdbase.c

Osdbase bietet mit den enthaltenen Klassen die Basis-Schnittstelle zum OSD. Hier wird die Definition und das Setzen möglicher DVB-Farben übernommen, sowie das Festlegen von Fenster-Attributen. Das ermöglicht letztendlich das Setzen der Bilder auf dem OSD.

eDvbColor definiert mögliche DVB-Farben. Zu den bereits definierten Farben können noch eigens definierte hinzugefügt werden.

Die Klasse *cPalette* übernimmt das Setzen der Farben. Mit jedem Aufruf von *NewColors()* wird eine fortlaufende Reihe von Farbeinträgen zurückgegeben, die seit dem letzten Aufruf hinzugefügt wurden. Der Rückgabewert ist die Adresse der ersten neu hinzugefügten Farbe, und der Index der ersten und letzten neuen Farbe (im gegebenen int-Parameter). Existieren keine neuen Einträge wird NULL zurückgegeben.

Mit der Klasse *cBitmap* können Bilder auf definierte Positionen gesetzt werden. Zusätzlich wird die Schriftart enthalter Texte mittels der Funktion *SetFont()* ermöglicht. Über die Funktion *Dirty()* wird die Abfrage ermöglicht, welcher Bereich der OSD-Daten aktuell übertragen werden soll. Letztendlich füllt die Operation *Fill()* das über die Koordinaten der linken oberen und rechten unteren Ecke definierte Fenster mit festgelegten Farben.

Die Klasse *cWindow* ermöglicht die Festlegung der Fensterattribute. So wird zum Beispiel festgelegt, welche Koordinaten (x0, y0) ein Fenster besitzt, ob es sich um geteilte oder alleinstehende Fenster handelt. Zudem bekommt jedes Fenster eine Identifikationsnummer (*handle*) zugeordnet. Auch enthaltene Daten können festgelegt und beispielsweise die Schriftart enthaltenen Textes definiert werden.

Weiterhin kann die Sichtbarkeit ein Fenster über die folgenden Werte festgelegt werden

(-1) als Fehler-Rückgabewert verwendet
(-2) entspricht *ALL_ WINDOWS*
(-3) entspricht *ALL_ TILED_ WINDOWS*
(-4) entspricht *LAST_ CREATED_ WINDOW*

Die maximal mögliche Anzahl von OSD-Fenster beträgt sieben. Diese können grundsätzlich alle geöffnet sein, aber nicht gleichzeitig angezeigt werden.

Die Klasse *cOsdBase* bildet die Basis-Schnittstelle zum OSD. Das bedeutet die Übermittlung der Indentifikationsnummer eines Fensters, der Fensteranzahl, den Koordinaten des Fensters. Weiterhin werden Fenster über die OSD-Hardware geöffnet, so Daten- und Farbdefinitionen geschrieben und modifiziert. Geöffnete Fenster können verschoben, oder auch temporär ausgeblendet werden. Die Funktion *Dirty()* ermöglicht die Feststellung, welcher Bereich der Daten aktuell an die Hardware gesendet wird. Mit der Operation *Flush()* ist dann die Übergabe der Daten aller Fenster an das OSD möglich. Zudem wird hier das Schließen nicht benötigter Fenster überwacht, und damit die Freigabe unnötig belegter Ressourcen.

5.6.2 Osd.c

Die Datei enthält die allgemeine OSD-Schicht.

cOsd überwacht den Status des OSD, mittels in *eOSState* definierter Werte.

Die Klasse *cOsdItem* übernimmt die Behandlung der OSD-Elemente, wie deren Anzeige oder das Setzen ihrer Farben und Schriftarten der enthalten Textelemente.

cOsdObject ermöglicht das Handling von OSD-Objekten. *cOsdMenu* ist als Unterklasse von *cOsdObject* für das Interface zur Bedienung des OSD verantwortlich.

5.6.3 Dvbosd.c

Die in der Datei *DvbOsd.c* enthaltene Klasse *cDvbOsd* ist für die Implementation der DVB-Onscreen-Displays zuständig. Ihre Methoden ermöglichen das Öffnen, Binden, Anzeigen, Verbergen, Verschieben und Schließen von Fenstern. Die in der Datei *Osdbase.c* enthalten Klassen setzen auf diese Funktionalität auf.

5.6.4 Menu.c

Hier sind alle aktuellen Menü-Implementationen enthalten. Zum Beispiel wird über die enthaltenen Klassen das Hauptmenü definiert, sowie die Anzeige von Kanälen und der Lautstärke. Ebenso ermöglichen sie die Kontrolle von Aufnahme und Wiedergabe.

cMenuMain bildet dabei eine Unterklasse von *cOsdMenu*, und ist damit ebenfalls für die Bedienung des OSD durch die Ausführung von Tastenbefehlen zuständig.

cDisplayChannel ist für die Kanalanzeige und deren Aktualisierung zuständig. *cDisplayVolume* liefert äquivalent dazu die Anzeige der Lautstärke.

cMenuRecordings ist für das Aufnahmemenü zuständig. *cRecordControl* kontrolliert die Aufnahme durch ständige Abfrage eventueller Ereignisse, des Timers und des verwendeten Device. *cRecordControls* hingegen übernimmt die direkte Kontrolle über Start und Beginn von Aufnahmen, sowie die Aktivierung von Timern.

cReplayControl ist eine Unterklasse von *cDvbPlayerControl* und kümmert sich entsprechend um die Kontrolle der Wiedergabe von Aufnahmen.

5.6.5 Menuitems.c

Die in *MenuItems.c* enthaltenen Klassen stellen universelle Menüelemente zur Verfügung. Das ermöglicht die Bearbeitung von Text und weiteren Menüelementen.

Die Klasse *cMenuEditItem* ist eine Unterklasse von *cOsdItem* und ermöglicht über ihre *SetValue()*-Funktion das Editieren von Menü-Elementen. Ihre Unterklassen *cMenuEditIntItem*, *cMenuEditBoolItem*, *cMenuEditChrItem* und *cMenuEditStrItem* ermöglichen entsprechend die Bearbeitung nach Datentypen, wie Integer- und Boolschen Werten, sowie Zeichen und Zeichenketten. Dieselbe Funktion besitzt auch die Klasse *cMenuEditStraItem*. Letztendlich ist mit der Klasse *cMenuTextItem* die Konfiguration von Text-Attributen (Position, Frabe, Schriftart) möglich.

cMenuSetupPage als Unterklasse von *cOsdMenu* ermöglich die Konfiguration der Setup-Seite des Menüs.

5.6.6 Font.c

Die Klasse *cFont* übernimmt das Schriften-Handling für das Onscreen-Display. Schriften sind dazu in *eDvbFont* definiert.

Weitere verwendbare Dateien sind: *fontfix.c*, *fontosd.c*, *Genfontfile.c*.

5.6.7 Interface.c

In dieser Datei wird eine abstrakte Schicht für die Benutzerschnittstellen zur Verfügung gestellt, das bedeutet die Schnittstelle zwischen OSD und Fernbedienung.

cInterface ist dabei für das Handling zwischen Fernbedienung und OSD zuständig. Es wird beispielsweise auf Tastenbedeutungen und SVDRP-Befehle geprüft.

5.7 Aufnahme

Dieser Abschnitt beschreibt die, für die Aufnahme und deren Bearbeitung, benötigten Dateien.

- *Recorder.c* - Der aktuelle DVB-Rekorder
- *Recording.c* - behandlung der Aufnahme-Dateien
- *Timers.c* - Timer-Handling
- *Videodir.c* - Funktionen für ein verteiltes Videoverzeichnis
- *Cutter.c* - Videoschnitt-Funktionen

5.7.1 Recorder.c

cRecorder als Unterklasse von *cReceiver* und *cThread* enthält alle Methoden für einen funktionierenden DVB-Rekorder: Empfang über einen implementierten Empfänger in einem unabhängigen Thread. So wird ein neuer Recorder mit definierter Priorität, gegebenem Zugriff auf eventuell verschlüsselte Kanäle und der Fähigkeit zur Aufnahme der gegebenen PIDs in eine Datei mit definiertem Namen erzeugt.

5.7.2 Recording.c

Die enthaltenen Klassen übernehmen das Handling aufgenommener Daten: Namensgebung, Setzen von Markierungen und Einordnung in das Videoverzeichnis.

Die Klasse *cResumeFile* erstellt eine Zusammenfassung der Aufnahme.

cRecording übernimmt das Handling der Aufnahme. Das bedeutet die Erstellung einer inhaltlichen Zusammenfassung, die Vergabe eines Titels und eventuell eines Serien-/Episodennamens, die Festlegung von Priorität und Lebenzeit der Aufnahme,.

Sind Aufnahmen fertiggestellt werden sie in das Video-Verzeichnis verschoben und aus dem Aufnahme-Verzeichnis gelöscht.

Die maximale Größe eines einzelnen Frames beträgt 192 KB. Die maximale Datei-Größe ist begrenzt durch den Bereich des Datentyps Integer. So sind im Falle von *unsigned int* 4GB, und im Falle von *signed int* 2GB möglich. Um also die sichere Aufnahme zu gewährleisten werden maximal 2000MB Dateigröße für Aufnahmen angenommen. Die minimale Größe eine Aufnahme beträgt 100MB.

Die Klasse *cRecordings* übernimmt die Konfiguration der Aufnahme-Liste.

cMark übernimmt die Markierung bestimmter Stellen, für beispielsweise das spätere Schneiden der Aufnahme. *cMarks* übernimmt die Konfiguration der Markierungsliste.

cRecordingUserCommand übernimmt die Aufnahme von benutzerdefinierten Befehlen.

Die Klasse *cIndexFile* legt den Index der Aufnahme fest. Die Klasse *cFileName* übernimmt die Namensgebung für aufgenommene Dateien.

5.7.3 Timers.c

Diese Datei stellt alle Klassen für die allgemeine Behandlung der Timer zur Verfügung.

In *eTimerActive* sind die möglichen Timer-Aktivitäten definiert.

Die Klasse *cTimer* dient der Auflistung der Timer im Menü unter Zuhilfenahme der Klasse *cMenuEditTimer*. Hier sind Start, Ende, Datum und Lebensdauer der Aufnahme, die Priorität des Timers, sowie der aktuelle Status (Aufnahme/noch bevorstehende Aufnahme), eine Zusammenfassung der Aufnahme und der Aufnahmekanal definiert. Zudem wird auf verändernde Events gewartet, sowie eine Überprüfung vorgenommen, ob das eingestellte Startdatum und die eingestellte Startzeit bereits erreicht ist. Außerdem wird mit Beginn der Aufnahme der Timer auf *Aktiv* gesetzt.

cTimers übernimmt die Konfiguration durch Nutzung der *cConfig*, die wiederum den Zugriff auf die Konfigurationsdatei *Timers.conf* ermöglicht.

5.7.4 Videodir.c

Diese Datei stellt Funktionen für ein verteiltes Videoverzeichnis zur Verfügung.

Das beinhaltet Funktionen zum Öffnen und Schließen, Umbenennen und Schließen von Video-Dateien, sowie zur Prüfung des vorhandenen Festplatten- oder Video-Disk-Speichers. Leere Verzeichnisse werden gelöscht.

5.7.5 Cutter.c

Stellt mit der Klasse *cCutter* Videoschnitt-Funktionen zur Verfügung.

5.8 Variable Anteile der Architektur/Plugins

5.8.1 Pluginkonzept

Zur Integration variabler Komponenten ist der VDR über das Plugin-Konzept erweiterbar. In dem Paket *Plugin* sind zunächst die Pakete *lib* und *src* enthalten. *lib* kann dabei Bibliotheken mit Standardfunktionen enthalten, wie für die Unterstützung von Mp3 oder Ac3 (*mad*, *liba52dec*), oder auch das Abspielen von DVDs (*libdvdread*). *src* enthält hingegen die Plugins. Die in *Plugin.c* enthaltenen Klassen bieten alle Funktionen zum Einbinden neuer Komponenten.

Zusätzlich zu Plugins können auch Skripte und Applikationen eingebunden werden, die beispielsweise über *Reccmds.conf* oder *Commands.conf* angesprochen werden. Als Beispiel kann hier die Applikation zur Markierung von Werbeblöcken[9] genannt werden.

Weitere Beispiele zu bereits realisierten Skripten/Plugins/Applikationen

- Nutzung der Status-Funktionen für Plugins
- TvTv-Skript, daß die Timer-Programmierung mittels EPG-Daten aus dem Web ermöglicht
- Skript zur Umwandlung von VDR-Aufnahmen ins AVI-Format
- Sendersuchlauf-Tool

Außerdem kann ein Plugin auch die Funktion des VDR so erweitern, daß es für weitere Plugins neue Klassen und Funktionen zur Verfügung stellt. Dies ist bereits im MPlayer-Plugin realisiert.

[9] Implementierung siehe Studienarbeit von Andreas Regel

So bietet dieses Plugin-Konzept hinsichtlich der Verwendung der OSD auch die Möglichkeit zu Verwendung völlig unabhängiger Anwendungen. So könnten sich beispielsweise Datenbankanwendungen einbinden lassen, die über das OSD steuerbar sind.

Ideen für die Weiterentwicklung des VDR hinsichtlich des Komponentenkonzepts wären außerdem:

1. Eine Möglichkeit wäre die ***Bereitstellung von Beispielen***. Das heißt, daß für jede vorstellbare Anwendungen ein Plugin geschrieben und kommentiert wird. Die Zusammenstellung eines neuen Plugin aus diesem Pool von Beispielen würde so um einiges erleichtert. So etwas ist seit Version 1.1.11 bereits als *Hello*-Plugin integriert.
2. Eine weitere Möglichkeit wäre eine ***Bibliothek, die Standardfunktionen*** wie beispielsweise *„Erzeuge Menü“* oder *„Greife auf X zu“* ***erzeugt*** und die eingebunden werden können. Der Entwickler hätte dementsprechend die Möglichkeit auf diese Funktionen zuzugreifen. Auf diese Weise funktioniert derzeitig implizit das Plugin-Konzept des VDR. Problem ist hier, daß sich dies relativ aufwendig gestaltet und zudem diese Quellcodebibliothek bereits aus vorcompilierten Funktionen besteht, die nicht mehr geändert werden können.
3. Eine sehr praktische Variante wäre ein ***Tool zur Codegenerierung*** ähnlich Visual Basic für Windows. Das würde die Entwicklung neuer Plugins und Skripte sehr vereinfachen, ist aber für Linux eher nicht denkbar.
4. Die gängige Variante: Der Quellcode steht meistens unter GPU/GNU als Freeware zur Verfügung, unter der Voraussetzung, daß Veränderungen ebenfalls veröffentlicht werden. So kann man sich ***Plugins erweitern oder zusammensetzen***. Diese recht einfache Art der Weiterentwicklung von Software wird auf die Weise unter UNIX/LINUX-Nutzer praktiziert. Problematisch ist hierbei, daß es oft an Dokumentation mangelt (siehe VDR), und daher ohne weitere Vorkenntnisse eine Einarbeitung meist schwierig ist.

5.8.2 Weitere Variationsmöglichkeiten

VDR ist hinsichtlich der optischen Erscheinung seines OSD variabel. Das bedeutet beispielsweise eine mögliche Änderung Menüanzeige, auch hinsichtlich des Hinzufügens von Farben (in *eDvbColors*) oder Schriften (in *fontosd.c* und *fontfix.c*). Solche Veränderungen können relativ üver frei verfügbare Patches vorgenommen werden.

Beispiel: Beauty Patch

5.8.3 Aktuelle Plugins

- MP3/MPlayer (Stefan Huelswitt, www.muempf.de)
- Teletext (Peter Seyringer, www.dapeace.de/teletext.htm)
- DVD Player (Andreas Schultz, warp10.net/dvd/plugin-devel), Abspielen von DVDs mittels libdvdread und libdvdcss (greift für Darstellung des mpeg2-Datenstromes auf Decoder der DVB-Karte zurück, somit sind Rechner ab 300 Mhz ausreichend)
- DXR3 Device (Andreas Schultz, Stefan Schluenss, et al www.schluenss.de/DXR3.html)
- S/VCD (Thomas Heiligenmann, www.heiligenmann.de/download)
- LCDproc (Martin Hammerschmid, home.pages.at/linux/dvb.html)
- Image Viewer (Alessio Sangalli)

- Remote Control (Oliver Endriss, endriss.escape.bei.t-online.de/vdr)
- Email Reader (Peter Seyringer, www.dapeace.de/vdr-mail.htm)
- Streaming (Sascha Volkenandt, www.magoa.net/linux)
- Tetris (Clemens Kirchgatterer, www.thf.ath.cx)
- Multitainer LCD (Meinrad Sauter, galileo.spaceports.com/m̃einrad)
- Media detection (Bernd Schweikert, bernd.schweikert.bei.t-online.de/vdr/plugin)
- Preferred channels (Olivier Jacques, www.olivierjacques.com/vdr/prefermenu)
- Console (Jan Rieger, ricomp.de/vdr)
- TVTV interface (Gerald Berwolf, www.genka.de) - Epg2Timers ermöglicht die Timer Programmierung via www.tvtv.co.uk. Der VDR ist auch per Internet programmierbar. Die EPG-basierte TV-Programmzeitschrift TVTV ermöglicht eine übersichtliche und nach Sparten und Kategorien geordnete Markierung zur Aufnahme gewünschter Sendungen und Serien. Die Nutzung des werbefinanzierten Dienstes ist kostenfrei, und erfordert lediglich eine Anmeldung. Für einen schnellen Zugriff können persönliche Einstellungen gespeichert werden. (www.tvtv.de) Das Programmpaket *epg2timers* sorgt dafür, dass die persönliche Merkliste aus dem Internet heruntergeladen, passend aufbereitet und dem VDR-Timer zur Verfügung gestellt wird. (ftp.cadsoft.de/pub/people/kls/vdr/Tools)
- File manager (Gerald Berwolf, www.genka.de)

5.8.4 Aktuelle Applikationen

Java Runtime Environment **-** Für VDR2DivX verwendet. (avifile.sourceforge.net)

MPlayer + Fonts **-** Movieplayer

LIRC **-** Ansteuerung der Fernbedienung

KVDR **-** Frontend für den KDE-Desktop. Damit lässt sich auf dem Rechner eine Bedienung per Tastatursteuerung realisieren. Zudem wird die Bildschirmausgabe des TV-Ausgangs der DVB-Karte in einem Fenster der X-Oberfläche oder im Vollbild angezeigt. Es lassen sich Helligkeit, Farbe und Kontrast einstellen und abspeichern, sowie Bilschirmsnapshot machen.

(www.s.netic.de/gfiala)

LCDProc **-** Für die Anzeige von Informationen auf einen LCD-Display

VDRAdmin - Timerprogrammierung via LAN. Ebenso wie das graphische Frontend *kvdr* setzt das für die Websteuerung benutzte Perl-Programmpaket *vdradmin* auf die Möglichkeit der Steurung über das Netzwerk auf. Viele Funktionen lassen sich nach dessen Installation dann komfortabel per Browsersteuerung erledigen. So ist auch eine elektronische Programmzeitschrift mit Suchfunktion integriert, die aus gespeicherten EPG-Daten gespeist wird. Mit der aktivierten Autotimer-Funktion können anhand von Suchbegriffen Timer-Aufnahmen automatisch generieren werden.

(www.LinVDR.org/download/vdradmin)

ToSvcd - SVCD-Konvertierung. Diese Applikation nimmt alternativ zur dauerhaften Archivierung der VDR-Aufnahmen im DivX-Format die Umwandlung ins das SuperVideo-Format vor. Es lassen sich bei Bedarf Optionen wie z. B. die maximalen Größe der Rohlinge und deren Anzahl übergeben. Zudem findet automatisch eine Umrechnung auf die Auflösung von 480x576 statt. Abweichend vom

Standard für SVCD's wird der Ton aber mit einer Samplingfrequenz von 48 kHz statt 44,1 kHz erstellt. (muse.seh.de/tosvcd)

Webmin **-** Kontrolle des Systems via WAN/LAN

WakeonLan **-** Hochfahren des VDR über LAN

NVRam-Wakeup **-** Hochfahren des VDR via Bios-Timer

5.8.5 Libraries

- *libsndfile* **-** Lesen und Schreiben von Dateien, die Sampled Sound enthalten
- *mjpegtools* - Tools für die Mpeg-Editierung
- *a52dec* **-** frei verfügbarer ATSC A/52 Stream-Dekoder
- *nasm* **-** Netwide Assembler
- *mad* - (M)PEG (A)udio (D)ecoder
- *ffmpeg* - Kommandozeilen-Tool zur Konvertierung von einem Videoformat in ein anderes
- *lame* **-** MP3-Codierer
- *CDfs* **-** Datei-System für Linux
- *libspopc* - einfache Pop3-Client-Bibliothek
- *libdv* **-** Software-Codec für DV-Video

5.8.6 Skripte

VDR2DivX (auch Avifile) **-** Diese Applikation nimmt eine automatische Konvertierung von Aufnahmen in das Speicherplatz sparende DivX-Format vor. Auf Wunsch können Vorgaben für Benennung und Verzeichnis vorgegeben werden. Dabei wird die Bitrate so berechnet, daß das Ergebnis auf einen einzelnen Rohling zur dauerhaften Archivierung passt. Zunächst ist der DivX4-Codec zum Wandeln des von der DVB-Karte gelieferten MPEG2-Formats nach DivX erforderlich. Hierzu nutzt man z. B. das für Linux erhältliche *divx4linux*-Paket.

mplayer.sh - Shell-Skript für Mplayer. Ruft den MPlayer mit der Möglichkeit zur Übergabe selbst definierbarer Parameter auf. Diese können in der *MPlayer.sh.conf* festgelegt werden.

convert.pl - Konvertierung zu SVCD

RunVDR

5.8.7 Tools

- *vdrscan* **-** Scan-Programm
- *vdradmin* - Programmierung des Timers via LAN
- *ds.jar* **-** Java-Konvertierungstool
- *transcode* **-** „Linux Video Stream Processing Tool"
- *netpbm* - Manipulation von Grafiken

- *tosvcd* - Konvertierungstool zu (S)VCD
- *nvram-wakeup* - Wake-Up via Bios-Timer
- *master-timer* - Skript zur automatisierten VDR-Timer-Programmierung
- *epg2timers* - VDR-Timer-Programmierung via www.tvtv.co.uk
- *conv2conf* - Kommandozeilen-Kanal-Konvertierer

5.9 Ausblick

Hier sind kurz Ideen für neue Plugins aufgeführt, die auch als Anregung für kommende Studien- und Diplomarbeiten genutzt werden können.

- Scan-Funktion für die Suche nach neuen Kanälen im Sinne eines regelmäßigen Abgleichs der Kanaldaten in der *Channels.conf*.
- Automatische Suche des VDR im Internet nach zusätzlichen Information zu den EPG-Daten zu aufgenommenen oder aufzunehmenden Sendungen. Das könnten beispielsweise Kritiken sein, die dem Nutzer zur Durchsicht zur Verfügung gestellt werden, und auf Wunsch in die Zusammenfassung integriert werden können.
- Erweiterte Filmdatenbank zur Sortierung nach Genres, die bei der Aufnahme hinterlegt werden können. Diese soll dementsprechend mit einer Suchfunktion ausgestattet werden, die dem Nutzer die bequeme Auswahl eines Filmes gestattet.

6 Ausblick

In diesem Kapitel sollen kurz Mängel und mögliche Schritte zur Behebung aufgezeigt werden, die ebenfalls als Anregung für neue Studien- oder Diplomarbeiten genutzt werden könnten.

Im Gespräch mit VDR-Nutzern konnten folgende Mängel identifiziert werden.

- Teilweise instabile Komponenten durch die mangelhafte Firmware der DVB-Karten. Dieses Problem wurde bereits im Kapitel 5.2 angesprochen und begründet, und ist zur Zeit schwer behebbar.
- Lange Kanalwechselzeiten, die durch die Hardware bedingt sind. Bei schnellem Umschalten kann dies sogar zum Absturz führen. Die Behebung könnte als Handshake zwischen 2 Karten realisiert werden.
- Probleme/Störungen/Grafikfehler im OSD, die ebenfalls auf die unausgereifte Firmware zurückzuführen sind.
- Teilweise ist die Installation von Plugins für unerfahrene Nutzer recht kompliziert, da ein einheitliches Konzept fehlt. Dies könnte durch ein neues, vereinfachtes Konzept zur Entwicklung hinsichtlich der Einbindung von Plugins mit definierten Schnittstellen vermieden werden.

A Glossar

CI	Common Interface, eine von der DVB Plattform spezifizierte Schnittstelle für digitale Receiver zum Anschluß eines Conditional- Access-Moduls. Dieses Modul enthält alle Komponenten, welche für den Descrambler und die Freischaltung des Teilnehmers notwendig sind.
File descriptor	Bevor ein Prozeß auf die Daten in einer Datei zugreifen kann, muß er sie mit dem Systemaufruf *open()* öffnen. UNIX überprüft hierbei die Zugriffsrechte an der Datei und gibt dem aufrufenden Prozeß im Erfolgsfall einen *file descriptor* zurück, der für alle weiteren Operationen auf dieser Datei notwendig ist. Mit dem Systemaufruf *close()* gibt der Prozeß die Rechte an einer Datei wieder auf, die Datei wird geschlossen.
DiSEqC	Digital Satellite Equipment Control, für den Empfang analoger und digitaler Programme von Astra, Eutelsat u.a.. Der Digitalempfänger liefert Steuersignale in digitaler Form, mit denen sich mehrere Antennen anwählen lassen. Im Gegensatz zu bisherigen analogen Steuersignalen (14 / 18 Volt und 22 kHz) bietet das DiSEqC-System eine wesentlich unfangreichere Anzahl von Schaltmöglichkeiten zur Steuerung und Überwachung von Ausrüstungsgegenständen einer Empfangsanlage.
DVB	Digital Video Broadcasting
EPG	Electronic Programme Guide. Dabei handelt es sich um mitgesandte Programminformationen für jeden Kanal, die Detailinformationen über aktuelle und zukünftige Sendungen liefern.
LNB/LNC	Low Noise Block Amplifier /Low-Noise-Blockkonverter, Empfangseinheit einer Satellitenantenne. Hardware zur Verstärkung und Konvertierung von Satellitensignalen eines bestimmten Bereiches von einer hohen Frequenz (gewöhnlich GHz) in eine niedrigere Zwischenfrequenz (gewöhnlich MHz). Der LNB wird im Brennpunkt einer Parabolantenne installiert, und wandelt die empfangene Frequenz in die erste Satelliten-Zwischenfrequenz um. Diese kann dann vom Receiver zur Darstellung des gewünschten Programmes oder Dienst genutzt werden.
MPEG 2	Kompressionsstandard als Erweiterung von MPEG-1 bis zu einer Gesamtdatenrate von 100 Mbit/s.
OSD	On-Screen-Display, zur Bedienung des VDR werden die Funktionen, auf Grundlage des Displays der DVB-Karte, als Bildschirm-Menü ins Fernsehbild eingeblendet.

Pipeline	Die Dateitypen p (pipeline) und s (socket) bezeichnen Dateien, die der Kommunikation von Prozessen dienen: Jeweils ein Prozeß versucht aus einer solchen Datei zu lesen, ein oder mehrere Prozesse können hineinschreiben. Während eine Pipeline auf Prozesse auf demselben Rechner beschränkt ist, stellt ein Socket eine Netzwerkverbindung dar.
PID	Packed Identification, jedes im Transportstream enthaltene Paket ist durch einen vierstelligen Code, getrennt nach Audio und Video, gekennzeichnet. So wird z. B. eine Überschreibung von SCPC-Signalen verhindert.
Polarisation	Schwingungsrichtung einer elektromagnetischen Welle, Ausrichtung des elektrischen Feldes eines Signals. Das ASTRA-Satellitensystem sendet Signale aus, die dieselbe Frequenz haben, jedoch eine unterschiedliche Polarisierung (horizontal oder vertikal), so daß das verfügbare Spektrum optimal genutzt wird. In der Regel wird die lineare Polarisation verwendet (also horizontal oder vertikal), stellenweise aber auch zirkulare Polarisation (rechtszirkular und linkszirkular). Durch die Polarisation können zwei Programme auf fast der gleichen Frequenz ausgestrahlt werden, ohne sich zu stören.
SPU	Signal Processing Unit (Signalverarbeitungseinheit)
Streaming	Die Streaming-Technologie (a. d. eng. stream= strom) ist ein Verfahren zur Übertragung von zeitabhängigen Medien (Video, Audio, Animation) in Echtzeit über Netze mit geringer Bandbreite (z.B. Internet). Dazu werden Komprimierungsalgorythmen, sogenannte „Codecs“ (a. d. eng. coder=Kodierer und decoder=Dekodierer) verwendet. Diese Codecs unterscheiden sich je nach der Art des Ausgangsmaterials (audio mono, audio stereo, video, etc.). Eine so komprimierte digitale Mediendatei wird in der Regel auf einem Server bereitgestellt und in ein Hypertextsystem eingebunden. Die Nutzer des Hypertextsystems können durch Aufruf eines Hyperlinks die Mediendatei aufrufen. Die Mediendatei wird dann in Datenpakteten von dem Server (a. d. eng. to serv=bedienen; ein Rechner der in Netzwerken Daten, Dienste und Programme bereitstellt) an den abrufenden Nutzerrechner übertragen und sofort abgespielt.
SVDRP	Simple VDR Protocol, eigens für den Netz-Zugriff auf den VDR entwickeltes Protokoll.
Transponder	Elektronische Einrichtung auf einem Satelliten, die das zu übertragende (Rundfunk-)Signal von einer Bodenstation empfängt, auf eine andere Frequenz umsetzt, verstärkt und zur Erde zurückschickt. Wird auch als Übertragungskanal eines Satelliten bezeichnet. Der Transponder empfängt Funksignale und sendet sie auf einer anderen Frequenz (in einen anderen Kanal umgesetzt) wieder aus (z. B. bei Rundfunksatelliten).
TS	Transport Stream

VDR	Video Disc Recorder

Literatur

[1] Paper/Report zu METODOC (Methodik und Werkzeuge für multimedial Dokumentation und elektronische Kataloge), Frauenhofer Institut, Arno Hitzges, Michael Krieger

[2] www.cadsoft.de/vdr

[3] www.linvdr.org

[4] www.lirc.org

[5] diverse c't - Artikel

[6] www.vdrportal.de