

DVB

Komponenten

Manual und Referenz

Die Informationen in diesem Handbuch wurden mit größter Sorgfalt erarbeitet; dennoch können Fehler nicht vollständig ausgeschlossen werden. Wir übernehmen keine juristische Verantwortung oder irgendeine Haftung für eventuell verbliebene Fehler und deren Folgen.

Das Werk ist urheberrechtlich geschützt. Alle Rechte, einschließlich der Vervielfältigung, Übersetzung, Mikroverfilmung sowie Speicherung und Verarbeitung in elektronischen Systemen, vorbehalten.

Kommentare und Fragen können Sie gerne an uns richten:

SR-Systems
Brüder-Grimm-Straße 130
36396 Steinau
Tel.: +49 (66 63) 91 88 66
Fax: +49 (66 63) 91 88 67
eMail: DATV@SR-Systems.de

04. Aug. 2003	Erstausgabe
10. Aug. 2004	Korrektur der Anschlußbelegung des Parallelports
26. Nov. 2004	+Platinenbeschreibungen, +Beispielaufbau, kl. Korrekturen von Tippfehlern, div. Aktualisierungen am Konfigurationsteil.

Autoren: Thomas Sailer, HB9JNX, Stefan Reimann, DG8FAC
Satz und Lektorat: V-Media, Steinau; www.V-Media.biz

Dieses Handbuch ist auf 100% holz- und chlorfreies pdf gedruckt; zur Erstellung wurden keine Tiere gequält oder getötet. Ebenfalls weder gequält noch getötet wurden zahlreiche Kaffee und andere koffeinhaltigen Erfrischungsgetränke, die bei der Erstellung dieses Handbuchs sehr hilfreich waren.

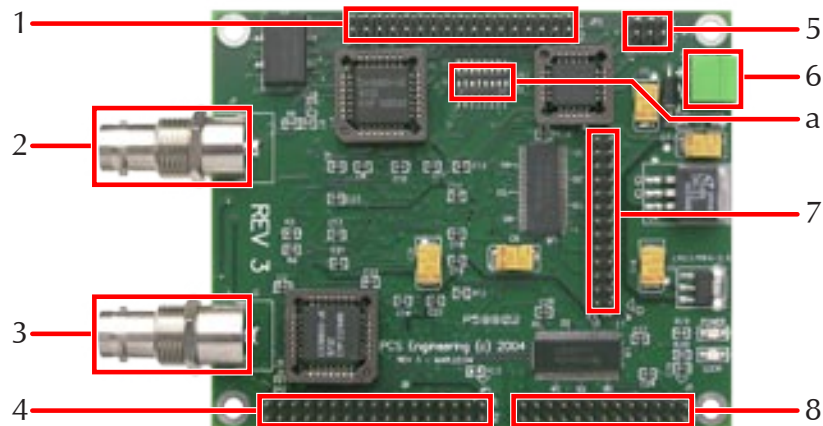
I. Inhaltsverzeichnis

I. Inhaltsverzeichnis	3	4.5.4.12 Video PID	30
1 Board- und Anschlußbeschreibungen	4	4.5.4.13 Audio PID	30
1.1 ASI-I/O-Interface	5	4.5.4.14 Program Map Table (PMT) PID	30
1.2 MPEG Encoder	6	4.5.4.15 Kanalname des Programms	30
1.3 DVB-S/C-Modulator 2×TS	7	4.5.4.16 Sprache	30
1.4 DVB-S/C-Modulator 4×TS	8	4.5.4.17 PID filter	30
1.5 DVB-T-Modulator	9	4.5.4.18 Tunermodus	31
1.6 Vierkanal-Multiplexer	10	4.5.4.19 Tunerfrequenz	31
1.7 I/Q-Modulator UHF	11	4.5.4.20 Tuner FEC Modus	31
1.8 I/Q-Modulator L/S	12	4.5.4.21 Tuner Symbolrate	31
1.9 NIM DVB-S	13	4.5.5 Teletext-Teil	31
1.10 NIM DVB-C	14	4.5.5.1 Program Clock Reference (PCR) PID	31
1.11 NIM DVB-T	15	4.5.5.2 Video PID	32
2 Anwendungsbeispiele	16	4.5.5.3 Teletext PID	32
2.1 Vier analoge und vier digitale Programme	16	4.5.5.4 Program Map Table (PMT) PID	32
2.2 Zehn analoge Programme mit ASI-In/Out	17	4.5.5.5 Program Callsign	32
2.3 10 analoge Kanäle ohne Verschlüsselungsmöglichkeit	18	4.5.5.6 Spracheinstellung	32
2.4 Analogvideo und Digitaltransponder	19	4.5.5.7 Bild-Dateien	32
3 Inbetriebnahme	20	4.5.5.8 VM Code	33
4. Konfiguration	21	4.5.6 Extern-Programm-Teil	33
4.1 Einführung	21	4.5.6.1 Program Clock Reference (PCR) PID	33
4.2 Terminologie	21	4.5.6.2 Program Map Table (PMT) PID	33
4.3 Upgraden der Microcontroller-Firmware	21	4.5.6.3 Sprache	33
4.4 Aufruf von DATVfwtool	22	4.5.6.4 Stream subsections	33
4.4.1 Reaktion auf fehlgeschlagene Schreibvorgänge	22	4.5.6.5 PID	34
4.5 Die Konfigurationsdatei	23	4.5.6.6 Stream Typ	34
4.5.1 Generelle Struktur	23	4.5.6.7 Stream ID	34
4.5.2 Platinenspezifischer Teil	25	4.5.6.8 Component Type	34
4.5.2.1 Taktfrequenz	25	4.5.6.9 Language	34
4.5.2.2 Boardtyp	25	4.6 Der alte Teletext-Encoder	35
4.5.2.3 Tabellengeneratorabschaltung	25	4.6.1 Der Teletext-Teil	35
4.5.3 Modulatorteil	25	4.6.1.1 Teletext Seitenkopf	35
4.5.3.1 Modulation	25	4.6.2 Teletextseiten	35
4.5.3.2 Konstellation	25	4.6.2.1 page number (Seitenzahl)	35
4.5.3.3 FEC Rate, DVB-S	26	4.6.2.2 Teletext lines (Teletextzeilen)	36
4.5.3.4 Sendefrequenz	26	4.7 Beispielkonfiguration	37
4.5.3.5 Symbolrate, DVB-S	26	4.8 Der neue Teletext-Encoder	38
4.5.3.6 Symbolrate, DVB-C	26	4.8.1 C-Code	39
4.5.3.7 Inversion	26	4.8.2 VM Built-In Library Functions	40
4.5.3.8 PTT (nur 4×TS)	27	4.8.2.1 C Typ-Größen	40
4.5.3.9 Transpondererkennung des Senders	27	4.8.2.2 C99 Standard-Makros	40
4.5.4 Transportstream-Teil	27	4.8.2.3 C99 Standard-Typen	40
4.5.4.1 Auswahl der Eingänge	27	4.8.2.4 C99 Standard-Funktionen	40
4.5.4.2 Auswahl der aktiven Taktflanke	28	4.8.2.5 Event-Log-Funktionen	40
4.5.4.3 Clock Debounce Filter	28	4.8.2.6 Zeit- und Datumsfunktionen	40
4.5.4.4 Bitrate	28	4.8.2.7 Parameter und Statistik-Funktionen	41
4.5.4.5 Auswahl des Videoeingangs	28	4.8.2.8 Numeric to String conversion	42
4.5.4.6 Video GOP-Konfiguration	28	4.8.2.9 TS1/TS2 table decoder	42
4.5.4.7 Spatialfilter	28	4.8.2.10 Highlevel teletext encoding functions	43
4.5.4.8 Audioencoder-Bitrate	29	4.8.2.11 Lowlevel Teletext encoding functions	43
4.5.4.9 Audioencoder-Encodiermodus	29	4.9 Anschluß eines PC-Parallels an einen TS-Input	44
4.5.4.10 Audioencoder-Samplingrate	29	5 Anhang	45
4.5.4.11 Program Clock Reference (PCR) PID	29	5.1 Danksagung	45
		5.2 Referenzen	45
		5.3 Autoren	45

1 Board- und Anschlußbeschreibungen

Die folgenden Kapitel beschreiben die auf den einzelnen Platinen vorhandenen Anschlüsse und Bedienelemente sowie die Bedeutung von evtl. vorhandenen LEDs.

1.1 ASI-I/O-Interface

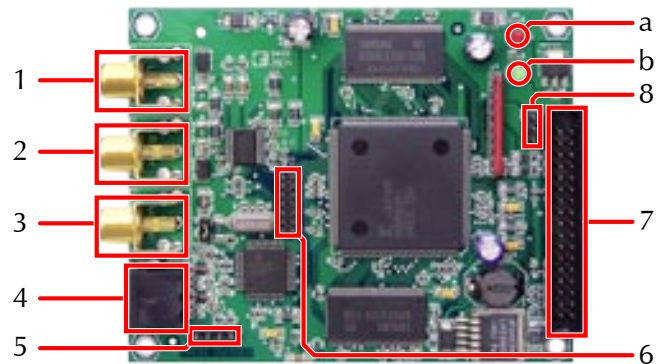


Platinenaußenmaße: 100×80 mm

Connectors	1	34poliger Pfostenstecker	SPI (TS) Eingang
	2	BNC-Buchse 75 Ω	ASI Eingang
	3	BNC-Buchse 75 Ω	ASI Ausgang
	4	34poliger Pfostenstecker	SPI (TS) Ausgang
	5	6poliger Pfostenstecker	JTAG Connector zur Flashprogrammierung
	6	2polige Schraubklemme	Spannungseingang nur 5 V=!
	7	26poliger Pfostenstecker	LVDS (=LVTTTL) Eingang
	8	26poliger Pfostenstecker	LVDS (=LVTTTL) Ausgang

Switches	a	8pol DIP-Switch	Betriebsmodus einstellen								
			Schalterstellung								Operation
			1	2	3	4	5	6	7	8	Result
			on	x	x	x	x	x	x	x	Es wird keine PSI (PAT/PMT) Tabelle erzeugt
			off	x	x	x	x	x	x	x	Alle 400 ms wird eine PSI (PAT/PMT) Tabelle erzeugt
			x	x	off	on	x	x	x	x	Der ASI-Eingang (2) wird mittels Hardware-Loopback auf den ASI-Ausgang (3) durchgeschleift. Hinweis: der ASI-Eingang (2) kann mit den Schaltern 5 und 6 auch zu LVDS/SPI verbunden werden.
			x	x	on	on	x	x	x	x	ASI Out (3) from LVDS In (7)
			x	x	on	off	x	x	x	x	ASI Out (3) from SPI In (1)
			x	x	x	x	off	on	x	x	LVDS (4)/SPI (8) Out from ASI In (2)
			x	x	x	x	on	on	x	x	Der SPI-Eingang (1) wird mittels Hardware-Loopback auf den Ausgang LVDS (8)/SPI (4) durchgeschleift. Hinweis: ASI-Ausgang (3) mit den Schaltern 3 und 4 möglich.
			x	x	x	x	on	off	x	x	Der LVDS-Eingang (7) wird mittels Hardware-Loopback auf den Ausgang LVDS (8)/SPI (4) durchgeschleift. Hinweis: ASI-Ausgang (3) mit den Schaltern 3 und 4 möglich.

1.2 MPEG Encoder

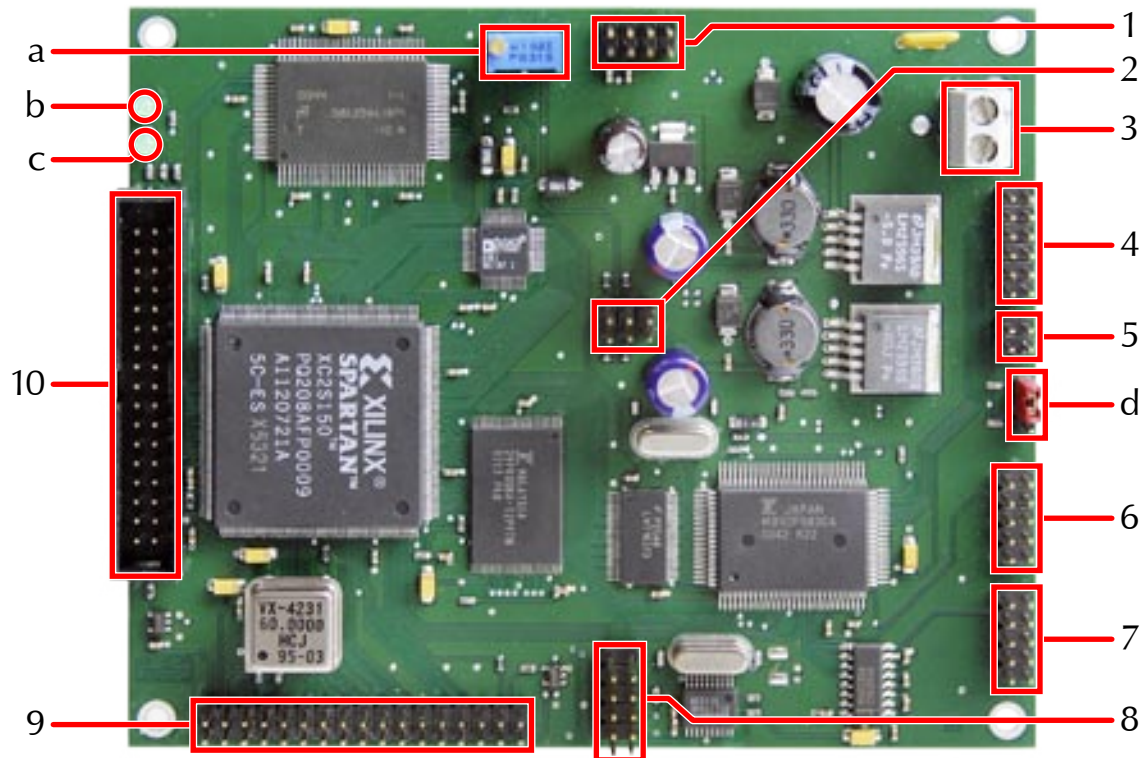


Platinenaußenmaße: 100×80 mm

Connectors	1	Cinch audio links	0 dB Eingangsempfindlichkeit
	2	Cinch audio rechts	Anpassung der Eingangsempfindlichkeit nur durch Schaltungsanpassung auf der Platine möglich!
	3	Cinch Video	Unterstützte Videostandards PAL/NTSC
	4	S-Video	
	5	4pol. Pfostenstecker	ext. I ² C-Bus (+5 V, GND,...)
	6	14pol. Pfostenstecker	Anschluß nach ITU-656
	7	34pol. Pfostenstecker	TS _{out} Verbindung zum Modulator/Multiplexer
	8	3pol. Pfostenstecker	XError vom Encoder-IC und Testausgang des Videocontrollers (zum Fehlerhandling in zukünftigen SW-Versionen)

LEDs	a	LED rot	Eingangsspannung 1,8 V ok
	b	LED green	Eingangsspannung 3,3 V ok

1.3 DVB-S/C-Modulator 2×TS

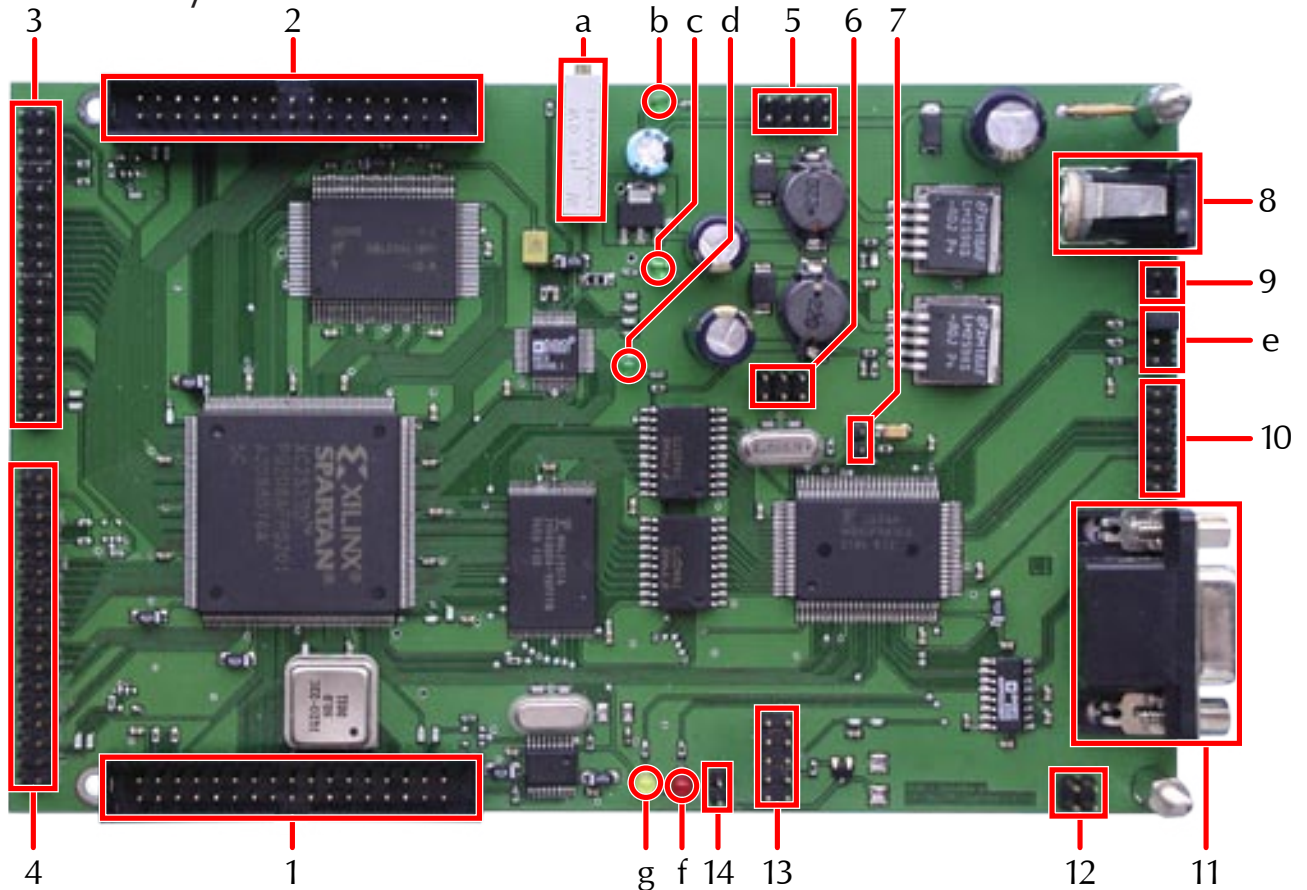


Platinenaußenmaße: 120×100 mm

Connectors	1	Verbindung zum I/Q-Modulator (UB)
	2	Verbindung zum I/Q-Modulator (diff. IQ)
	3	Spannungseingang (10–24 V DC)
	4	optional
	5	IE-Bus Verbinder (unbenutzt)
	6	Serielle Schnittstelle COM1 (wird zum Programmieren benutzt, Baudrate 19k2 8N1)
	7	Serielle Schnittstelle COM2
	8	Verbindung zum I/Q-Modulator (PLL-Control)
	9	TS1 Eingang
	10	TS2 Eingang

LEDs/Switches	a	Potentiometer zur Einstellung der I/Q-Balance
	b	leuchtet auf: Firmware im µC läuft blinkt: Encoder-Firmware wird in eben jenen geladen geht aus: Die Encoder-Software wurde erfolgreich gestartet geht wieder an: einer oder mehrere Encoder konnten nicht gestartet werden.  Beachten Sie bitte, daß Encoder nur an dem Transportstream-Eingang funktionieren, für den sie programmiert wurden! Wird anstelle eines Encoders ein Tuner/ext. Clock angeschlossen, zeigt die LED nach dem Einschalten ein Dauerleuchten.
	c	leuchtet nach dem Laden und nach dem Starten der Firmware
	d	MCU-Modus Jumper (Run/Program-Umschaltung)

1.4 DVB-S/C-Modulator 4×TS

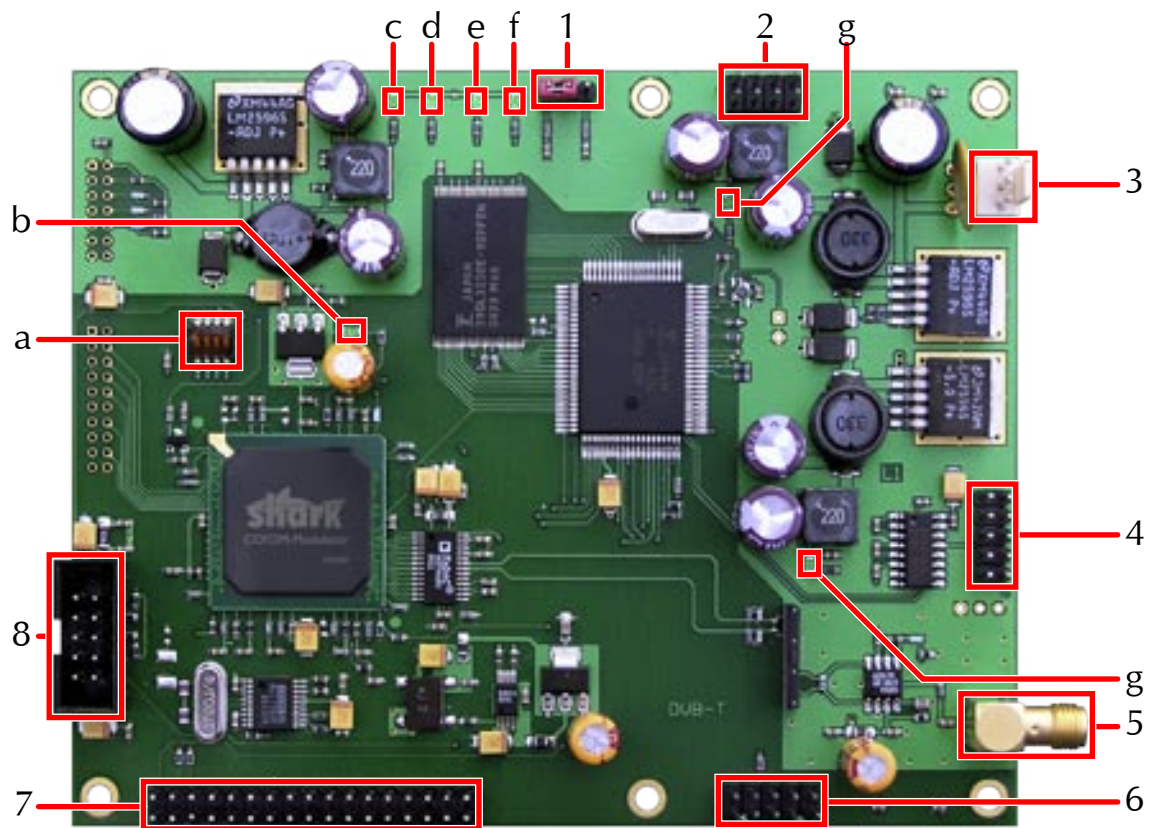


Platinenaußenmaße: 160×100 mm

Connectors	1-4	Transportstream-Anschluß 1-4  An die Anschlüsse 1 und 2 können Encoder, DVB-Tuner (S/C/T), Multiplexer und ext. TS-Inputs angeschlossen werden, an die Anschlüsse 3 und 4 jedoch nur Encoder. Zur leichteren Unterscheidung sind die Anschlüsse 1 und 2 mit, die Anschlüsse 3 und 4 ohne Wanne ausgeführt.
	5	Verbindung zum I/Q-Modulator (diff. IQ)
	6	Verbindung zum I/Q-Mod. (PLL-Control)
	7	ext. Reset
	8	Spannungseingang (10-24 V DC) <i>Achtung:</i> Pluspol innen!
	9	IE-Bus Verbinder (unbenutzt)
	10	DCF77/Analogspannung
	11	Serielle Schnittstelle COM1 wird zum Prog. benutzt, Baudrate 19k2 8N1
	12	Serielle Schnittstelle COM2
	13	Verbindung zum I/Q-Modulator
	14	ext. PTT

LEDs/Switches	a	Poti zur Einstellung der I/Q-Balance
	b	Betriebsspannung 3,3 V i.O.
	c	Betriebsspannung 5,0 V i.O.
	d	Betriebsspannung 2,5 V i.O.
	e	3×Jumper
	f	leuchtet auf: Firmware im µC läuft blinkt: Encoder-Firmware wird in eben jenen geladen geht aus: Die Encoder-Software wurde erfolgreich neu gestartet geht wieder an: einer oder mehrere Encoder konnten nicht gestartet werden.
	 Beachten Sie bitte, daß Encoder nur an dem Transportstream-Eingang funktionieren, für den sie programmiert wurden! Wird anstelle eines Encoders ein Tuner/ext. Clock angeschlossen, zeigt die LED nach dem Einschalten ein Dauerleuchten.	
	g	leuchtet nach dem Laden und nach dem Starten der Firmware

1.5 DVB-T-Modulator

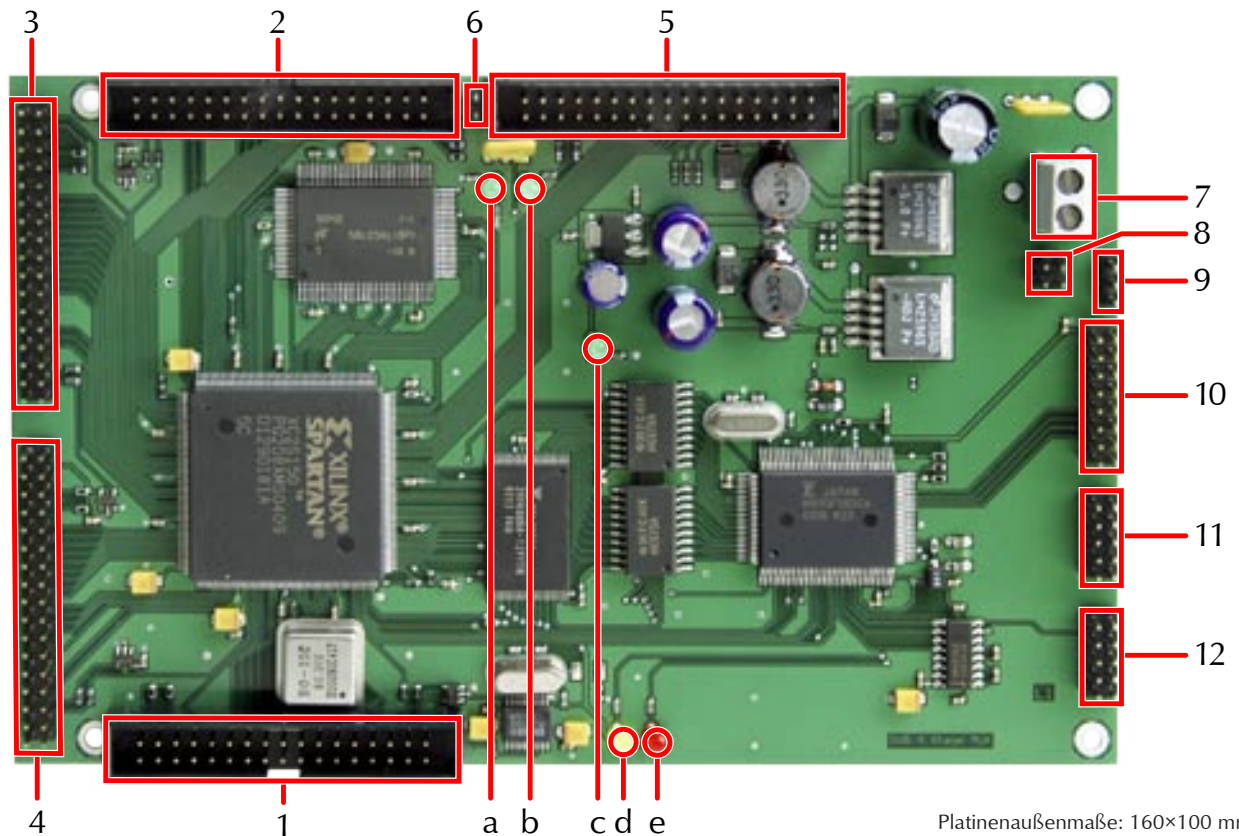


Platinenaußenmaße: 125×100 mm

Connectors/controls	1	3pol. Jumper	Schaltet Betriebsmodus um zwischen <i>program</i> und <i>run</i>
	2	8pol. Pfostenstecker	Spannungsversorgung für das Upconverter-Board
	3	Spannungseingang	Spannungsversorgung für das Board (9–24 V DC)
	4	10pol. Pfostenstecker	RS232 (38k4 8N1)
	5	SMA-Buchse	HF-Ausgang (36,125 MHz Basisband)
	6	10pol. Pfostenstecker	Steuerung für das Upconverter-Board
	7	34pol. Pfostenstecker	Transportstream-Anschluß für Encoder oder Multiplexer
	8	10pol. Pfostenstecker	I2C-Bus Anschluß zum Bedienfeld

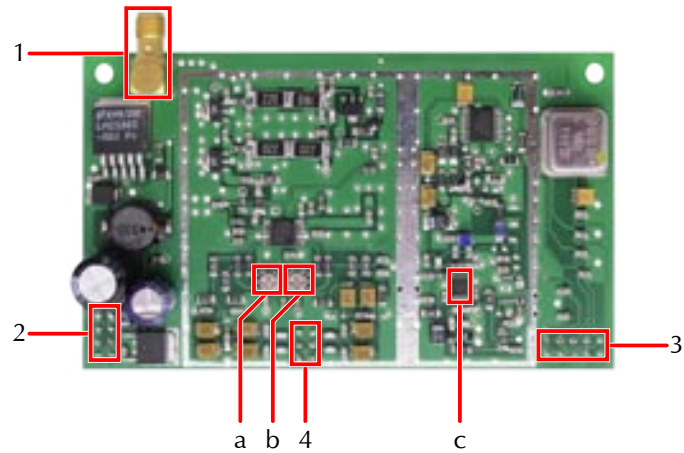
LEDs	a	4pol. DIP-Schalter	optional, z.Z. unbenutzt
	b	LED grün	Spannung o.k. (2,5 V)
	c	LED grün	opt. Status FPGA
	d	LED grün	Firmware läuft (blinkt während der Ausführung!)
	e	LED grün	opt. Status MCU
	f	LED grün	Lädt Firmware (blinkt während des Ladens, nach dem Laden aus => ok)
	g	LED grün	Spannung o.k. (3,3 V)
	h	LED grün	Spannung o.k. (5,0 V)

1.6 Vierkanal-Multiplexer



Connectors	1-4	Transportstream Eingänge 1–4
	5	Transportstream Ausgang
	6	Umschaltung int. Betriebsspannung von 7/ext. Betriebsspannung von Board an 5
	7	Spannungseingang (10–24 V DC)
	8	IE-Bus Connector (not used)
	9	MCU Modus Jumper (Run/Program)
	10	optional, z.Zt. nicht benutzt
	11	Serielle Schnittstelle COM1 (zum Programmieren, Baudrate 19k2 8N1)
LEDs/Switches	12	Serielle Schnittstelle COM2
	a	zeigt an, wenn von einem externen Board an 5 eine Betriebsspannung zur Verfügung steht.
	b	an , wenn TS _{Clock} von Board an 5 benutzt wird, aus bei Benutzung des int. TS _{Clock}
	c	Eingangsspannung 3,3 V i.O.
	d	leuchtet nach dem Laden und nach dem Starten der Firmware
	e	leuchtet auf: Firmware im µC läuft blinkt: Encoder-Firmware wird in eben jenen geladen geht aus: Die Encoder-Software wurde erfolgreich neu gestartet geht wieder an: einer oder mehrere Encoder konnten nicht gestartet werden. ☞ Beachten Sie bitte, daß Encoder nur an dem Transportstream-Eingang funktionieren, für den sie programmiert wurden! Wird anstelle eines Encoders ein Tuner/ext. Clock angeschlossen, zeigt die LED nach dem Einschalten ein Dauerleuchten.

1.7 I/Q-Modulator UHF

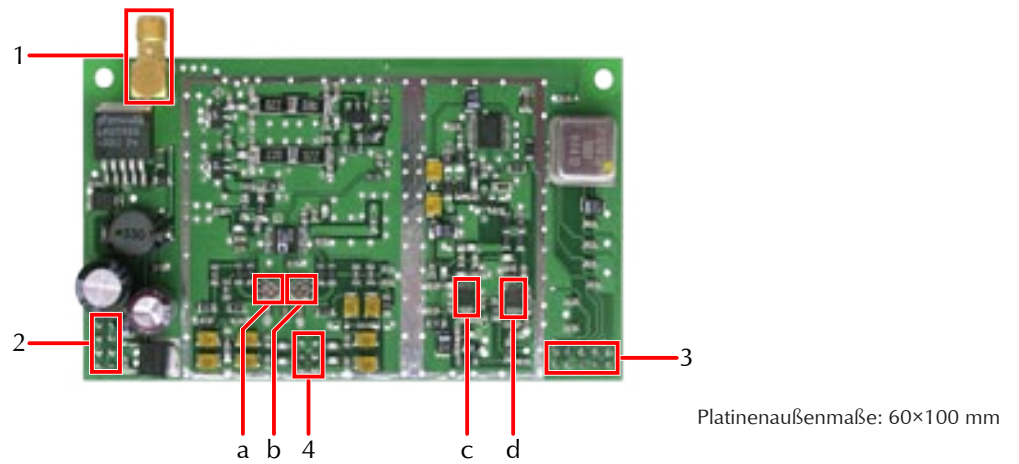


Platinenaußenmaße: 60×100 mm

Anschlüsse	1 HF-Ausgang SMA-Buchse (50 Ohm)
	2 Spannungsversorgung (12 V)
	3 Steuersignale für PLL etc.
	4 Modulationseingang

Anderes	a I-Balanceeinstellung
	b Q-Balanceeinstellung
	c VCO (375–525 MHz)

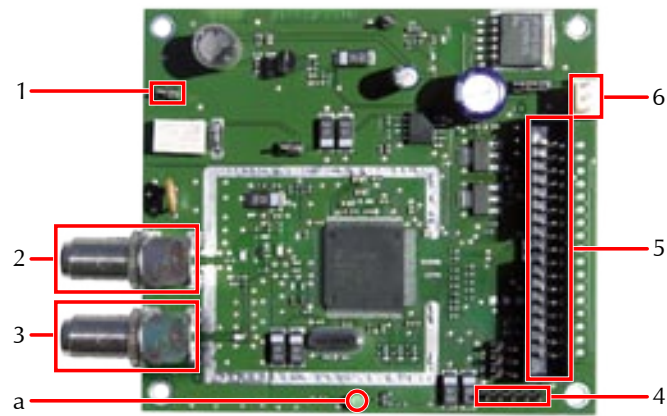
1.8 I/Q-Modulator L/S



Anschlüsse	1 HF-Ausgang SMA-Buchse (50 Ohm)
	2 Spannungsversorgung (12 V)
	3 Steuersignale für PLL etc.
	4 Modulationseingang

Andere	a I-Balanceeinstellung
	b Q-Balanceeinstellung
	c VCO (1175–1325 MHz)
	d VCO (2250–2600 MHz)

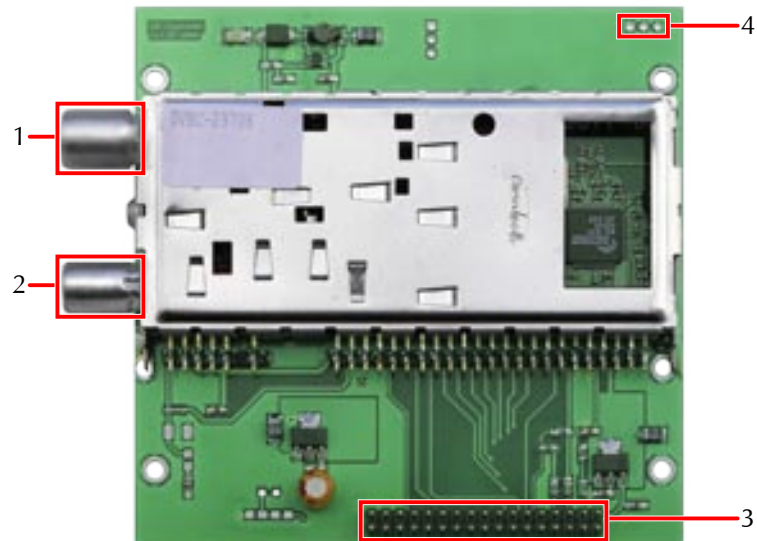
1.9 NIM DVB-S



Platinenaußenmaße: 90×80 mm

Anschlüsse	1 LNB Power on
	2 Antenneneingang
	3 Antennenausgang (durchgeschleift, loopthrough)
	4 Ext. I2C (z.Z. unbenutzt)
	5 TS Out
	6 LNB Power (20–24 V)
LED	a RF Lock

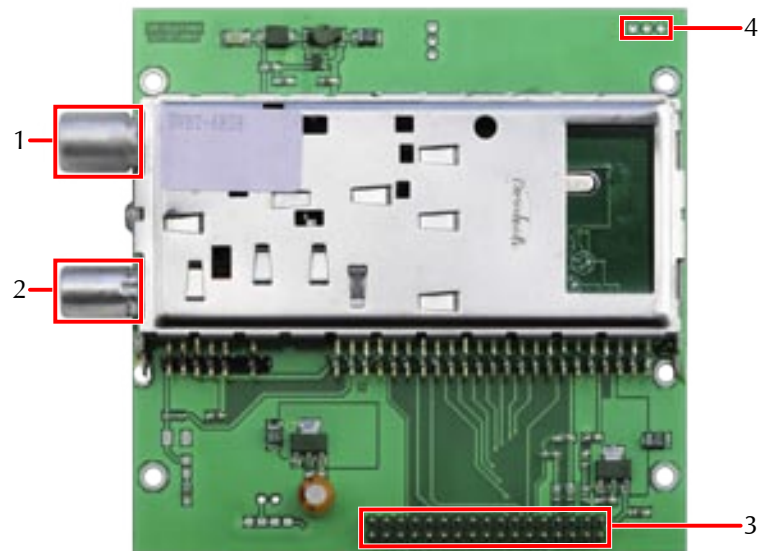
1.10 NIM DVB-C



Platinenaußenmaße: 100×100 mm

Anschlüsse	1 Antenneneingang
	2 Antennenausgang (durchgeschleift, loop-through)
	3 TS out
	4 LNB Power in

1.11 NIM DVB-T



Platinenaußenmaße: 100×100 mm

Anschlüsse

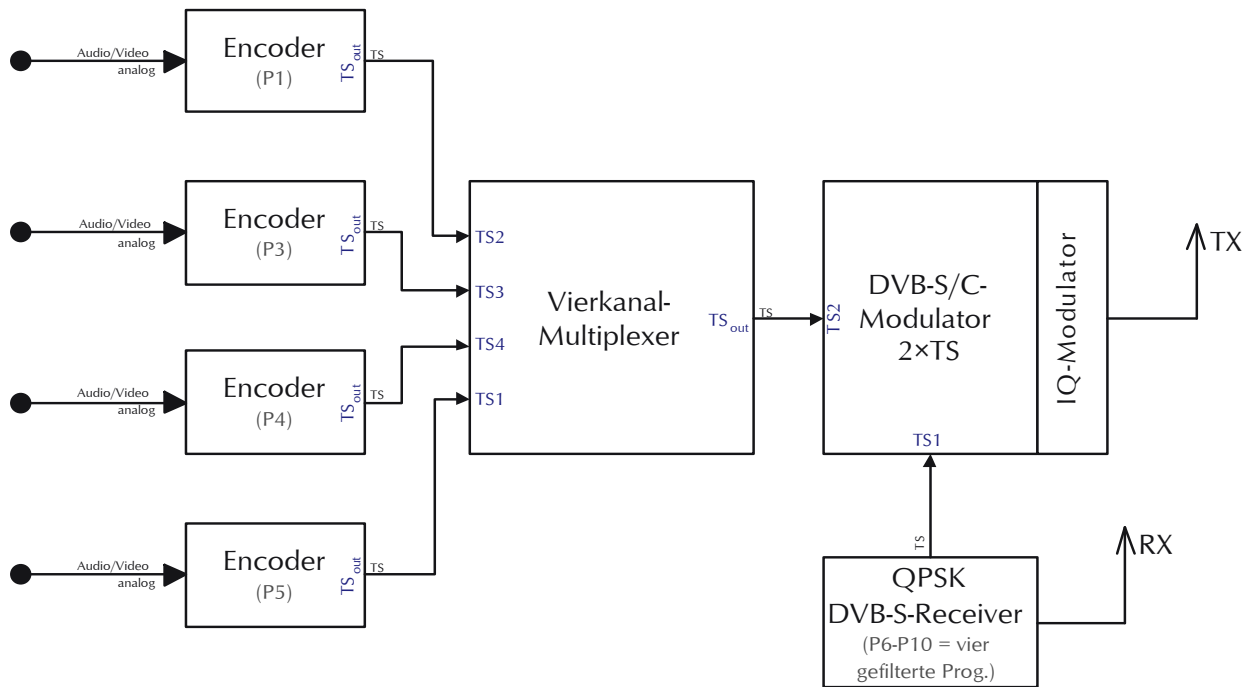
- 1 Antenneneingang
- 2 Antennenausgang (durchgeschleift, loop-through)
- 3 TS out
- 4 LNB Power in

2 Anwendungsbeispiele

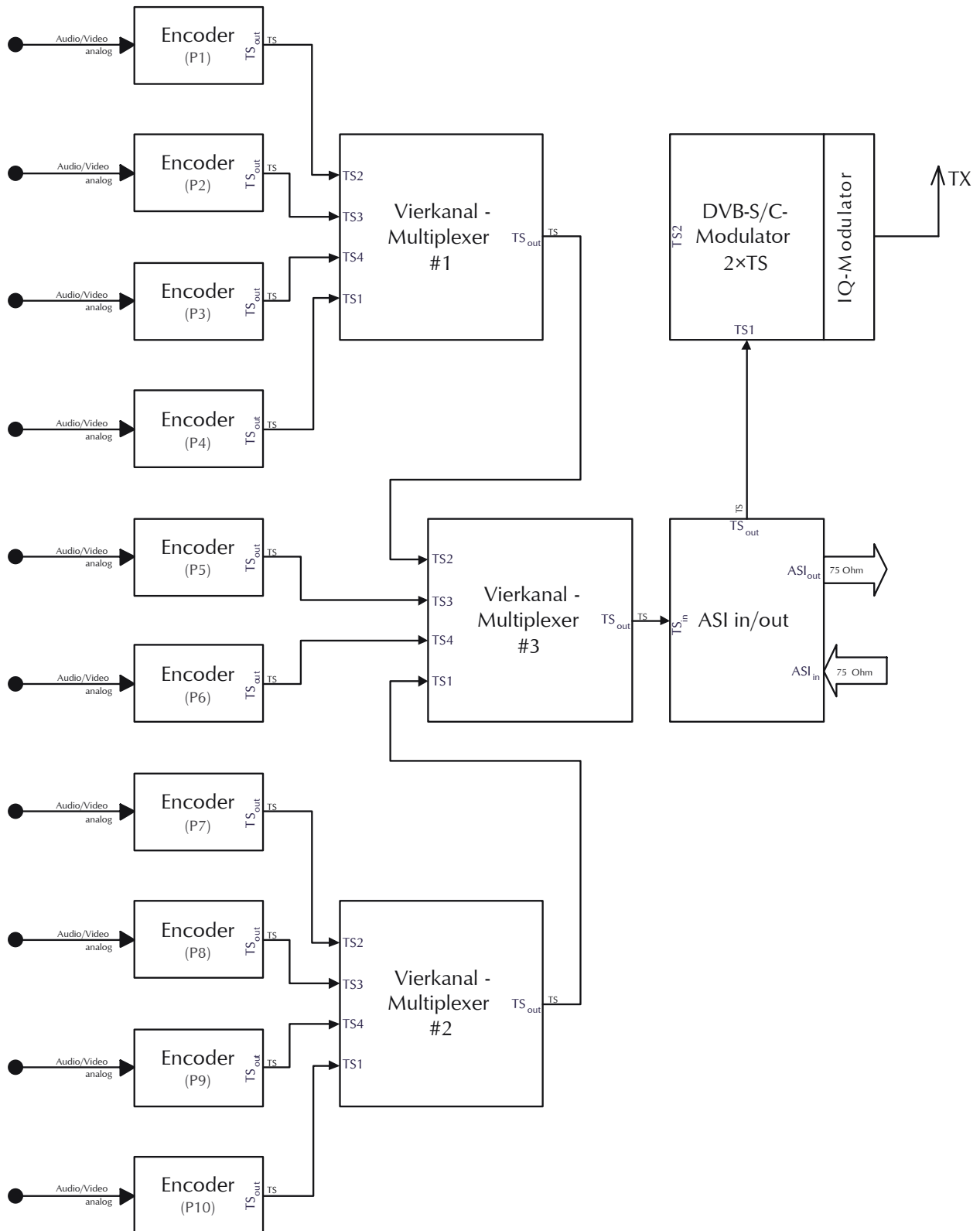
In diesem Kapitel soll anhand einiger Beispiele aufgezeigt werden, wie die verschiedenen Baugruppen zu unterschiedlichen Funktionseinheiten kombiniert werden können.

2.1 Vier analoge und vier digitale Programme

Version mit Mux4-Board, einem DVB-S-Empfänger und dem Zweikanal-Baseband-Board.

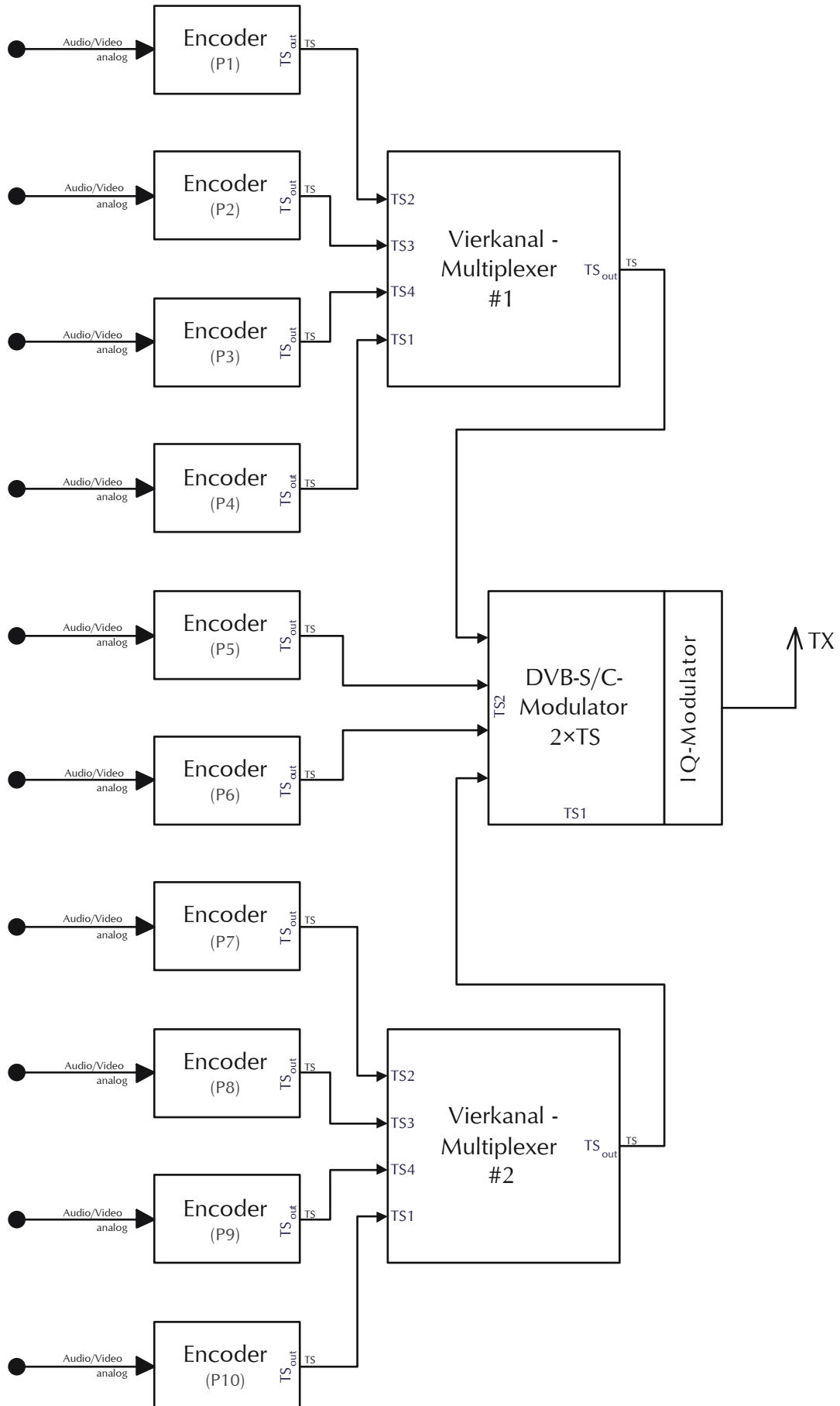


2.2 Zehn analoge Programme mit ASI-In/Out



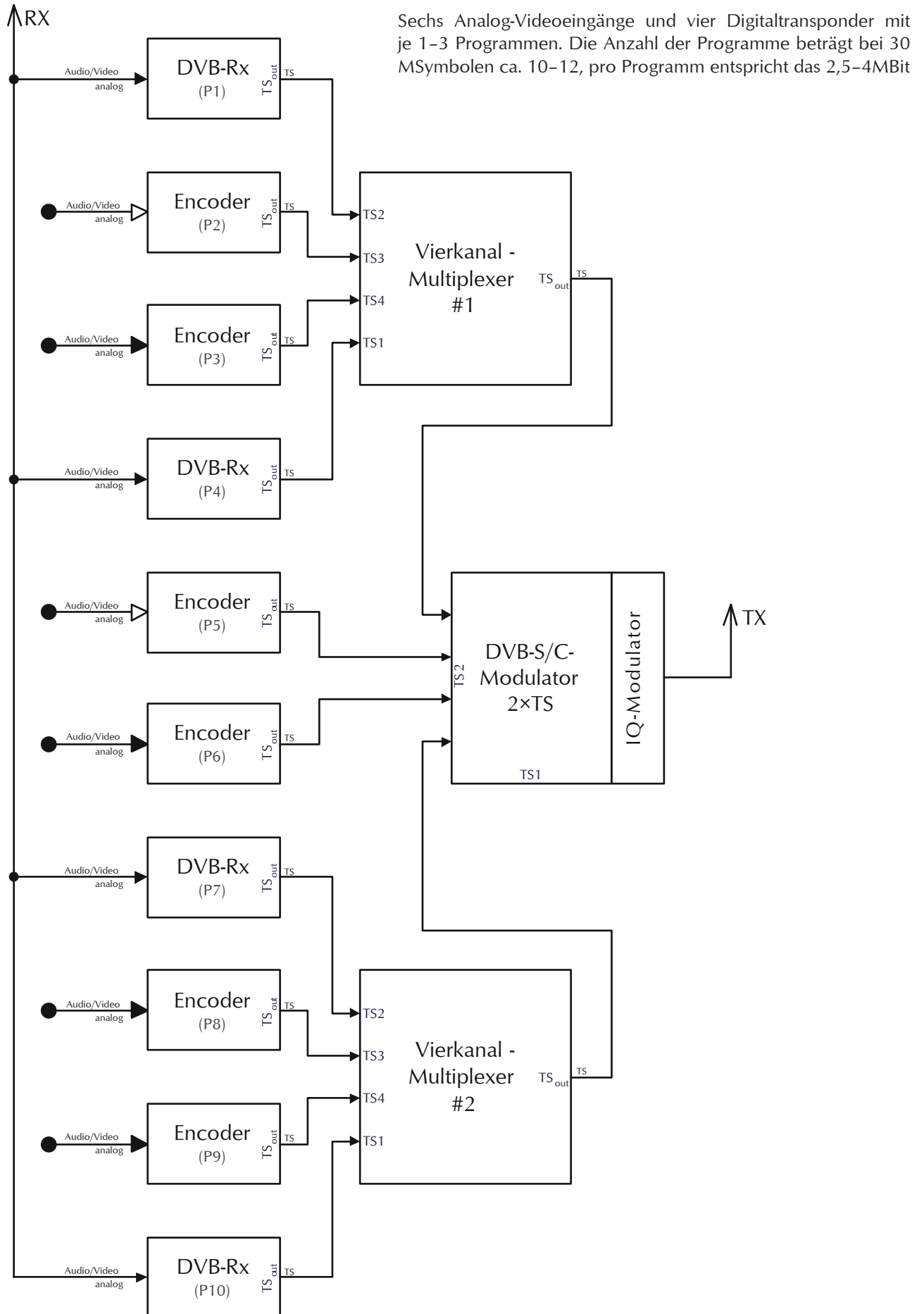
An den Multiplexern #1 und #2 muß „Drive Tx Clock = Off“ eingestellt sein, an Multiplexer #3 hingegen „Drive Tx Clock = On“. Bei Multiplexer #3 müssen die Eingänge TS1 und TS2 auf „fujitsuval“ gestellt werden.

2.3 Zehn analoge Kanäle ohne Verschlüsselungsmöglichkeit



Am DVB-Modulator müssen die Eingänge TS1 und TS2 auf „fujitsueval“ gestellt werden.

2.4 Analogvideo und Digitaltransponder



3 Inbetriebnahme

Aufbau gemäß Kapitel 2.

Schließen Sie an den seriellen Anschluß des DVB-S/C-Modulatorboards Ihren PC an und starten Sie ein Terminalprogramm. Die Parameter sind 19.200 Baud, 8 Datenbits, keine Parität, ein Stopbit (19k2, 8N1). Das Terminalprogramm sollte nun den seriellen Anschluß des DVB-S/C-Modulatorboards abhören.

Schalten Sie jetzt die Versorgungsspannung ein. Bei korrektem Aufbau sollte am DVB-S/C-Modulator Folgendes passieren: die rote LED leuchtet auf, kurz danach die gelbe. Daraufhin blinkt die rote LED und geht anschließend aus. Währenddessen sollte ihr Terminalprogramm einen Text der folgenden Art angezeigt haben:

Ausgabe	Bedeutung
MCU SW V0.9a (jnx)	Softwareversion MCU
(baseband board v35c)	Softwareversion DVB-S/C-Modulatorboard
Bootstrape options: PLL_EN 1 PLL_DATA 1 PLL_CLK 1 XRESET0 1 PLLTHR01	
FLASH manufacturer 004 (Fujitsu) device 22AB (29F400 B) sector 0 protection 0000	verwendeter Flash-Baustein
FLASH test passed	
FPGA configuration start (from FLASH)...	
FPGA configuration completed successfully	
PLL lock indication passed	HF-Modul drauf? passed=ja
Register test passed RAM test passed	ext. SRAM wird getestet
Encoder SIO test passed	Kommunikation mit Encoder wird getestet
01.01.02 00:00:00 D-ATV Transmitter startup, rev MCU SW V0.6(jnx) (baseband board v35c)	DVB-S/C-Modulatorboard startet
01.01.02 00:00:00 Setting mode reg: 0x0083 / 0x0083 FEC mode 2/3 Inversion off Freq 1255000kHz	Einstellungen (DATVfwtool)
01.01.02 00:00:00 Setting PLL frequency to 1255000kHz	
> 01.01.02 00:00:00 Downloading encoder 0 firmware (length 0x027338)	
01.01.02 00:00:00 PLL lock detected	HF startet - ab hier arbeitet der Sender!
01.01.02 00:00:02 Encoder 0 started (Encoder rev 0x00000250 SAA7113 rev 0x11)	Encoder läuft (rote LED aus)
Tabelle 3a: Die Startausgabe des D-ATV-Senders, mit Erklärung der Bedeutung	

4. Konfiguration

4.1 Einführung

Die nachfolgenden Kapitel beschreiben die Vorgehensweise bei Firmwareupgrades, Konfigurationen sowie das Datenformat und die Semantik des D-ATV-Konfigurationsfiles. Dieses Konfigurationsfile wird von DATVfwtool analysiert und über die serielle Schnittstelle in den D-ATV-Flash-Speicher geladen.

4.2 Terminologie

TS, Transportstream	Datenstrom, bestehend aus Transportstream-Paketen, die eine beliebige Anzahl von Video-, Audio- und/oder Daten-Streams beinhalten.
Transportstream Packet	Ein mit fester Länge von 188 Bytes definiertes Paket (4 Bytes Header, 184 Bytes Nutzdaten), welches die Video-, Audio- und andere Daten beinhaltet. Eine PID zeigt dem Empfänger an, zu welchem Daten-Stream die Information gehört.
PID	Packet Identifier Der Packet Identifier ist ein 13-Bit-Zahlenwert (dez. 0–8191), der den Stream identifiziert, zu dem ein Transportstream-Paket gehört. PID 8191 (hex 0x1fff) zeigt ein Paket an, welches keine sinnvollen Daten enthält. Er wird verwendet, um Transportstreams zu füllen, wenn keine sinnvollen Daten zum Senden verfügbar sind. Die PIDs 0–31 (0x00-0x1f) sind reserviert für Systemtabellen.
FEC	Forward Error Correction Die FEC fügt redundante Bits zum Sendestream hinzu, um dem Empfänger die Korrektur kleiner Übertragungsfehler zu ermöglichen.
SI Tables	System Information Tables Die SI-Tables sind Datenstrukturen, die in jedem Transportstream vorhanden sind und dem Empfänger ermöglichen, die Programme zu finden. Sie stellen sozusagen das Inhaltsverzeichnis des Transportstreams dar.
NIT	Network Information Table. Die NIT enthält Informationen über das Netzwerk, wie den Netzwerknamen, die Frequenz, Fehlerkorrektur, Modulationsart und die dem jeweiligen Netzwerk zugehörigen Transportströme sowie deren Parameter.
PAT	Program Association Table Pakete mit der PID 0, sie enthalten Informationen, welche PIDs für die PMTs benutzt werden.
PMT	Program Map Table Enthält Daten über den Inhalt des Programms (Video, Audio, Daten) und welche PIDs die jeweiligen Datenströme enthalten.

4.3 Upgraden der Microcontroller-Firmware

DATVfwtool kann jetzt auch zum Upgrade der Firmware eingesetzt werden. Um das zu erreichen, muß zuerst das Board in den Firmwareupgrade-Modus gejumpert werden (siehe Abb. 4.3a).

Dann muß das Board an die Versorgungsspannung angeschlossen und ein Reset ausgelöst werden. Der Aufruf `DATVfwtool -d /dev/com1 -fm` läßt DATVfwtool überprüfen, ob ein Upgrade notwendig ist und erledigt selbigen dann, sofern dies der Fall sein sollte.

4.4 Aufruf von DATVfwtool

Ein typischer Aufruf von DATVfwtool sieht wie folgt aus:

```
DATVfwtool -d /dev/com1 -v2 -c sample.conf -W
```

Der -d Parameter gibt den seriellen Port an, an dem der DVB-Sender angeschlossen ist. Die Portbezeichnungen für diesen Aufruf sind in Tabelle 4.4a aufgelistet.

Die Kapitel 4.5ff beschreiben das Format der Konfigurationsdatei (im obigen Beispiel heißt die Konfigurationsdatei „sample.conf“) und die Parameter. Nicht erwähnte Parameter werden als experimentell gesehen, sie funktionieren eventuell nicht oder können jederzeit entfernt werden. Tabelle 4.4b enthält eine Liste aller gültigen Parameter für DATVfwtool:

Parameter		Wirkung
normale Schreibweise	alt.*	
-check	-C	überprüft die Konfigurationsdatei
-write	-W	schreibt ins externe Flash. Die Software ändert nur Sektoren, die sich auch geändert haben.
-force-write	-F	schreibt ins externe Flash, auch, wenn sich nichts geändert hat.
-config-unchecked		lädt die angegebene Konfigurationsdatei und fährt auch bei Fehlern fort
-c		lädt die angegebene Konfigurationsdatei und hält bei Fehlern an
-sig-read		read a signal generator register
-sig-write		write to a signal generator register
-s		output to syslog instead of stderr
-d		use the following serial port
-v		set verbosity level to the following number
-T		toggle the table (PAT, PMT, EIT, NIT, SDT,...) transmission
-f		download the flash programmer to the fujitsu microcontroller
-m		update the microcontroller firmware if it has changed
-M		update the microcontroller firmware regardless if it has changed
-R		restart the baseband board
Tabelle 4.4b: Kommandozeilenparameter von DATVfwtool		*alt. - alternative Schreibweise

4.4.1 Reaktion auf fehlgeschlagene Schreibvorgänge

Gelegentlich kann es vorkommen, daß beim Schreiben der Konfiguration in das externe Flash Fehler auftreten. In diesem Fall muß das externe Flash erst gelöscht werden, bevor ein erneuter Schreibvorgang versucht werden kann.

Hierzu ist es beim DVB-Modulator 4×TS notwendig, vor dem Anlegen der Betriebsspannung die Pins 14 und 16 von U10 brücken. Nun erst die Betriebsspannung einschalten und dann die Brücke entfernen. Durch diese Vorgehensweise kann das Flash nicht gefunden werden, und die Firmware startet in einem speziellen Modus (symbolisiert durch das „R>“ als Eingabeaufforderung). Nach dem Drücken der Leertaste sollte nun Folgendes erscheinen:

Ausgabe

```
MCU SW V0.9a (jnx) (baseband board v35c)
Bootstrap options: PLL_EN 1 PLL_DATA 1 PLL_CLK 1 XRESET0 1 PLLTHRO 1FLASH
manufacturer 0004 (Fujitsu) device 22AB (29F400 B) sector 0 protection 0000
FLASH test passed
Firmware error: 0xFFFFFFFF / 0x0000000F
```

```
R> Menu (restricted to FLASH download)
f - Flash                      d - FPGA download from flash
a - FPGA analyze & download from flash  q - Test FPGA CPU interface
Q - Read Test of FPGA CPU interface    r - Test regs
P - Parameters                  t - Timer Registers
T - Start Timer                e - Reset all Encoders
```

Tabelle 4.4.1a: Die Startausgabe des DVB-Senders, wenn die Software im Flash ungültig ist.

Aus diesem Menü heraus kann das Flash gelöscht werden. Hierzu reicht es, einfach „f“ drücken, um das Flashmenü aufzurufen:

Ausgabe

```
R> FLASH: Manufacturer: 0x0004 Model: 0x22AB
Menu
r - Read Sector
e - Erase Sector
S - Program
T - Autoselect Test
```

Tabelle 4.4.1b: Das Flash-Menü

Um das Flash zu löschen, muß nun „e“, gefolgt von der zu löschenden Sektornummer, eingegeben werden. Die Sektoren sind numeriert von 0–7, also lautet die erste Eingabe „e0“. Kurz darauf sollte die Löschbestätigung und das Flash-Menü erscheinen:

Ausgabe

```
Erasing Sector 0: succeeded
Menu
r - Read Sector
e - Erase Sector
S - Program
T - Autoselect Test
```

Tabelle 4.4.1c: Das Flash-Menü, mit Löscherfolgsmeldung in der ersten Zeile

Nun noch die weiteren Sektoren löschen, also nacheinander „e1“ eingeben, Rückmeldung abwarten, „e2“ eingeben, Rückmeldung abwarten, usw. bis einschließlich „e7“. Das externe Flash ist nun gelöscht und ein erneuter Schreibvorgang sollte problemlos funktionieren.

4.5 Die Konfigurationsdatei

4.5.1 Generelle Struktur

Abb. 4.5.1a zeigt die generelle Struktur der Konfigurationsdatei (für einen 4×TS – für einen 2×TS entfallen die Gruppen transportstream 3 und 4 natürlich). Zeilen, die mit # beginnen, sind Kommentare und werden von DATVfw-tool nicht interpretiert. Der board-Teil gruppiert die Parameter der DVB-S/C-Modulatorplatine. Der modulator-Teil gruppiert die Modulationsparameter. Der transportstream-Parameter faßt eine Gruppe von Parametern

zusammen, die zu dem durch die Nummer angegebenen Transportstream-Eingang gehören. Der teletext-Teil schließlich beinhaltet die Parameter für den Teletextdecoder sowie die Standbildparameter.

```
# Sample D-ATV configuration file

board {
};

modulator {
};

transportstream 1 {
};

transportstream 2 {
};

transportstream 3 {
};

transportstream 4 {
};

teletext {
};
```

Abb. 4.5.1a: Die generelle Struktur der Konfigurationsaufbau

Die Parameter sind spezifiziert und verwenden die Syntax Parameter = Wert;; Zahlenwerte können dezimal integer oder hexadezimal integer mit einem **0x**-Präfix sein. Ein **k**- oder **M**-Suffix multipliziert den Zahlenwert mit 1.000 oder 1.000.000. Textwerte müssen in doppelten Anführungszeichen eingeschlossen werden.

Abb. 4.5.1b zeigt den Signalweg durch den DVB-Modulator. Das DVB-S/C-Modulatorboard verbindet die Datenquellen (z.B. MPEG2-Encoder oder DVB-S-Receiver) durch die Transportstream-Schnittstellen mit dem IQ-Modulator.

Die Transportstream-Schnittstellen sind parallele Schnittstellen und beinhalten acht Daten-, eine Takt- und mehrere optionale Synchronisationsleitungen.

Abb. 4.5.1c zeigt die Transportstreamanschlußbelegung. TS1 und TS2 haben beide einen FIFO im Signalweg. Die Richtung des TS-Taktsignals (CK) ist deshalb konfigurierbar. Es kann entweder vom DVB-S/C-Modulatorboard oder aus der Datenstruktur kommen.

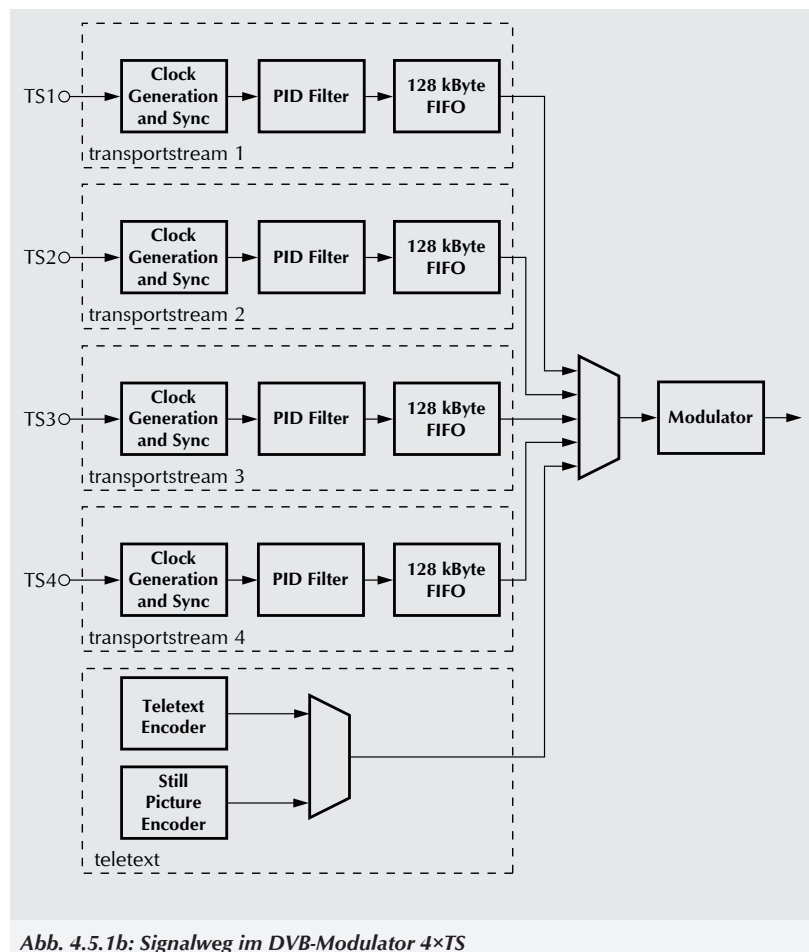


Abb. 4.5.1b: Signalweg im DVB-Modulator 4xTS

TS3 und TS4 haben keinen FIFO; das TS-Taktsignal muß deshalb vom DVB-S/C-Modulatorboard generiert werden, um Datenverluste zu vermeiden. Diese Ports werden deshalb hauptsächlich für die MPEG2-Encoder verwendet.

4.5.2 Platinenspezifischer Teil

4.5.2.1 Taktfrequenz

Der Taktparameter `clock` gibt die Frequenz des Quarzoszillators auf dem DVB-S/C-Modulatorboard an. Alle Timings werden aus diesem Quarz abgeleitet. Das D-ATV-Board wird normalerweise mit 60 MHz ausgeliefert, aber für extrem niedrige Übertragungsraten kann die Quarzfrequenz auch reduziert werden. Das Maximum ist 62 MHz.

Beispiel:

```
clock = 60M;
```

4.5.2.2 Boardtyp

Der Parameter `board` gibt an, um welchen Platinentyp es sich handelt. Mögliche Werte sind `admod` für einen DVB-S/C-Modulator 2×TS, `ham` für den DVB-S/C-Modulator 4×TS und `mux4` für den 4fach-Multiplexer.

Beispiel:

```
board = ham;
```

4.5.2.3 Tabellengeneratorabschaltung

Normalerweise generiert das D-ATV-Basisbandboard alle DVB-Tabellen, die ein Empfänger benötigt, um alle Programme innerhalb eines DVB-Transportstreams zu finden. Diese Tabellen können aber abgeschaltet werden. Dies ist dann nützlich, wenn beispielsweise ein Multiplexer diese Tabellen bereits erzeugt hat.

Beispiel:

```
generate tables = on;
```

4.5.3 Modulatorteil

Dieses Kapitel beschreibt die Konfiguration für die DVB-Modulatoren 4×TS und 2×TS.

4.5.3.1 Modulation

Dieser Parameter wählt den Modulationsstandard. Mögliche Werte sind `dvb-s` und `dvb-c`.

Beispiel:

```
modulation = dvb-s;
```

4.5.3.2 Konstellation

Dieser Parameter wählt die Konstellation. Mögliche Werte sind `qpsk` im DVB-S-Modus und `qam16`, `qam32` oder `qam64` im DVB-C-Modus.

Beispiel:

```
constellation = qpsk;
```

4.5.3.3 FEC Rate, DVB-S

Dieser Parameter spezifiziert die „inner Forward Error Correction Code Rate“. Er erlaubt eine Anpassung des Verhältnisses zwischen übertragener Bitrate und der Robustheit des modulierten Signals. Mögliche Werte sind 1/2, 2/3, 3/4, 5/6, 7/8. Dieser Parameter hat nur Bedeutung im DVB-S-Modus.

Beispiel:

```
fec = 5/6;
```

4.5.3.4 Sendefrequenz

Der `frequency` Parameter spezifiziert die Sendefrequenz. Die gewählte Sendefrequenz sollte im 70-, 23- oder 13 cm-Amateurband liegen und das HF-Modul muß das gewählte Band unterstützen.

Beispiel:

```
frequency = 1275M;
```

4.5.3.5 Symbolrate, DVB-S

Der Symbolraten-Parameter spezifiziert die Bandbreite des Modulatorsignals (siehe Gl. 1) und die Anwenderbitrate (siehe Gl. 2).

Das Verhältnis F_{clk}/SR muß eines aus 2, 2 $\frac{1}{3}$, 2 $\frac{1}{2}$, 2 $\frac{2}{3}$, 3, 3 $\frac{1}{3}$, 3 $\frac{1}{2}$, 3 $\frac{2}{3}$, 4, 4 $\frac{1}{3}$, 4 $\frac{1}{2}$, 4 $\frac{2}{3}$, 5, 5 $\frac{1}{3}$, 5 $\frac{1}{2}$, 5 $\frac{2}{3}$, 6, 6 $\frac{1}{3}$, 6 $\frac{1}{2}$, 6 $\frac{2}{3}$, 7, 7 $\frac{1}{3}$, 7 $\frac{1}{2}$, 7 $\frac{2}{3}$, 8, 8 $\frac{1}{3}$, 8 $\frac{1}{2}$, 8 $\frac{2}{3}$, 9, 9 $\frac{1}{3}$, 9 $\frac{1}{2}$, 9 $\frac{2}{3}$, 10, 11, 12, 13, 14, 15 oder 16 sein. DATVfwtool rundet die Symbolrate zum nächstmöglichen gültigen Verhältnis.

Beispiel:

```
symbol rate = 15000k;
```

$BW \approx \frac{4}{3} SR$ (Gl. 1)	
$BR = 2 \cdot SR \cdot R_{inner} \cdot R_{outer}$ (Gl. 2)	
SR	Symbol Rate (Symbols/s) (siehe 4.5.3.5)
BW	Signal Bandwidth (Hz)
BR	User Bitrate (Bits/s)
R_{inner}	Inner FEC Rate (siehe 4.5.3.3)
R_{outer}	Outer FEC Rate, fixed at 188/204
F_{clk}	Crystal Oscillator Frequency (siehe 4.5.2.1)

4.5.3.6 Symbolrate, DVB-C

Der Symbolraten-Parameter spezifiziert die Bandbreite des Modulatorsignals (siehe Gl. 3) und die Anwenderbitrate (siehe Gl. 4).

Das Verhältnis F_{clk}/SR muß eines von 8, 9, 10, 11, 12, 13, 14, 15 oder 16 sein. DATVfwtool rundet die Symbolrate zum nächstmöglichen gültigen Verhältnis. Es ist zu empfehlen, eine Bandbreite von 8 MHz oder weniger zu verwenden sowie eine Symbolrate von 6,9 MSymbole/s oder weniger, da die Empfänger auf 8 Mhz begrenzt sind. Um 6,9M zu erreichen, wird ein 55,2 MHz-Oszillator benötigt!

Beispiel:

```
symbol rate = 6000k;
```

$BW \approx 1,15 SR$ (Gl. 3)	
$BR = \log 2q \cdot SR \cdot R_{outer}$ (Gl. 4)	
SR	Symbol Rate (Symbols/s) (siehe 4.5.3.6)
BW	Signal Bandwidth (Hz)
BR	User Bitrate (Bits/s)
Q	Constellation QAMq
R_{outer}	Outer FEC Rate, fixed at 188/204
F_{clk}	Crystal Oscillator Frequency (siehe 4.5.2.1)

4.5.3.7 Inversion

Dieser Parameter bestimmt, ob die I&Q-Signale vertauscht werden sollen. Vertauschen der Signale erzeugt einen Effekt gleich dem Empfang eines USB-Signals mit einem LSB-Empfänger. Viele, aber nicht alle Empfänger detek-

tieren automatisch, ob Inversion benutzt wird. In `off`-Programmierung wird das D-ATV-Signal direkt gesendet. Wenn ein spektruminvertierender Transverter benutzt wird, auf `on` stellen.

Beispiel:

```
inversion = off;
```



Die WinTV DVB-S Nova Karte mit convergence.de-Firmware scheint Inversion-Betrieb nicht automatisch zu erkennen!

4.5.3.8 PTT (nur 4×TS)

Dieser Parameter wählt aus, ob der Sender beim Einschalten der Spannungsversorgung ein- oder ausgeschaltet wird. Die PTT kann danach über das Menü geschaltet werden. Der Parameter ist typischerweise auf `on` gesetzt für D-ATV-Relais und auf `off` für Endstellen.

Beispiel:

```
ptt = off;
```

4.5.3.9 Transponderkennung des Senders

Der Netzwerkname-Parameter muß auf die Transponderkennung des Senders eingestellt werden.

Beispiel:

```
network name = "DemoTV";
```

4.5.4 Transportstream-Teil

4.5.4.1 Auswahl der Eingänge

Dieser Parameter spezifiziert den Transportstream-Portmodus. Die folgenden Modi existieren:

<code>off</code>	Port abgeschaltet
<code>datvencoder</code>	D-ATV MPEG2-Encoder an Port angeschlossen
<code>fujitsu eval</code>	Fujitsu MPEG2-Encoder Evaluation Board oder Multiplexer an Port angeschlossen
<code>extclock</code>	Fremdgetaktetes Signal an Port angeschlossen

Das „Fujitsu MPEG2-Encoder Evaluation Board“ ist vergleichbar dem D-ATV Encoderboard. Der Unterschied ist, daß das DVB-S/C-Modulatorboard nicht versucht, die MPEG2-Firmware in den Encoder zu laden. Die `extclock` Option kann nur an TS1 und TS2 verwendet werden.

Beispiel:

```
mode = datvencoder;
```



Wichtiger Hinweis: Die VPIDs sind per Default vorgegeben. Dabei ist TS1=0x20, TS2=0x30, TS3=0x40 und TS4=0x50

4.5.4.2 Auswahl der aktiven Taktflanke

Dieser Parameter hat nur einen Effekt, wenn der Eingang im `extclock`-Modus ist. Er spezifiziert die aktive Flanke des TS-Taktes (CK). Gültige Werte sind `falling`, `rising` und `both`.

Beispiel:

```
clock edge = rising;
```

4.5.4.3 Clock Debounce Filter

Dieser Parameter bestimmt das Verhalten des Clock Debounce Filters. Der Wert muß auf das größte N gestellt werden, welches die Bedingungen aus Abb. 4.5.4.3a erfüllt.

Beispiel:

```
clock filter = 4;
```

$1 \leq N \leq 4$	
$F_{TSCk} \leq \frac{F_{clk}}{2N}$	
F_{TSCk}	Transportstreamtaktfrequenz
N	Clock Debounce Filter Parameter (siehe 4.5.4.3)
Fclk	Quarzoszillatorfrequenz (siehe 4.5.2.1)
Abb. 4.5.4.3a	

4.5.4.4 Bitrate

Dieser Parameter bestimmt die gesamte sinnvolle Bitrate in den Port.

Beispiel:

```
bitrate = 4500k;
```

4.5.4.5 Auswahl des Videoeingangs

Dieser Parameter hat nur im `datvencoder`-Modus einen Effekt. Er spezifiziert die Charakteristik des Videoeingangssignals. Er besteht aus einem oder mehreren Schlüsselwörtern aus der Liste, separiert durch Kommata:

Beispiel:

```
video input = d1, pal, svideo;
```

d1	D1-Auflösung (752×576 px.)
hd1	HD1-Auflösung (384×576 px.)
sif	SIF-Auflösung (384×288 px.)
qsif	QSIF-Auflösung (192×144px.)
ntsc	Eingangssignal ist NTSC
pal	Eingangssignal ist PAL
composite	Benutze Composite-Eingang
svideo	Benutze S-Video-Eingang

4.5.4.6 Video GOP-Konfiguration

Dieser Parameter spezifiziert die Bildencodiersequenz des Encoders. Der voreingestellte Modus liefert gute Encoder-Effektivität zum Preis einer höheren Encoder-Latenz (Zeitverzögerung zwischen Eingangsbild und encodiertem Ausgangsbild). Die Latenz kann durch Verkleinerung der t reduziert werden. Dieser Parameter sollte nur von Experten, die MPEG2-Encodierung wirklich gut verstehen, geändert werden.

Beispiel:

```
video gop = "IBBPBBPBBPBBPBB";
```



Die minimale GOP-Größe ist 1, die maximale 30. Die Größe wird nicht explizit angegeben, sie ergibt sich aus der gewählten Kombination von I-, P- und B-Frames.

4.5.4.7 Spatialfilter

Dieser Parameter stellt die Cut-Off-Frequenz des Spatialfilters sowie die Bildschärfe ein. Mögliche Werte sind `standard`, `soft`, und `sharp`.

Beispiel:

```
spatial filter = "standard";
```

4.5.4.8 Audioencoder-Bitrate

Dieser Parameter hat nur im `datvencoder`-Modus einen Effekt. Er spezifiziert die Bitrate des MPEG2 Layer2 Audioencoders.

Gültige Bitraten sind in der Tab. 4.5.4.8a dargestellt, sind jedoch abhängig vom eingestellten Audio-Encodiermodus (siehe Kap. 4.5.4.9).

Beispiel:

```
audio bitrate = 192k;
```

Bitrate	32k	48k	56k	64k	80k	96k	112k	128k	160k	192k	224k	256k	320k	384k
Stereo				✓		✓	✓	✓	✓	✓	✓	✓	✓	✓
Joint Stereo				✓		✓	✓	✓	✓	✓	✓	✓	✓	✓
Dual Channel				✓		✓	✓	✓	✓	✓	✓	✓	✓	✓
Single Channel	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				

Tab. 4.5.4.8a: Gültige Audioencoder-Bitraten und -Modi

4.5.4.9 Audioencoder-Encodiermodus

Dieser Parameter hat nur im `datvencoder`-Modus einen Effekt. Er spezifiziert den Encodiermodus des MPEG2 Layer2 Audioencoders. Der Wert muß eines dieser Schlüsselwörter sein:

stereo	Eingangssignal ist ein Stereosignal.
joint stereo	Eingangssignal ist ein Stereosignal. Der Encoder soll Redundanzen zwischen beiden Kanälen zur Datenreduktion benutzen.
dual channel	Beide Kanäle sind unabhängig voneinander.
single channel	Nur ein Kanal vorhanden.

Beispiel:

```
audio mode = joint stereo;
```

4.5.4.10 Audioencoder-Samplingrate

Dieser Parameter hat nur im `datvencoder`-Modus einen Effekt. Er spezifiziert die Samplerate des MPEG2 Layer2 Audioencoders. Gültige Werte sind 48000, 44100 und 32000.

Beispiel:

```
audio sample rate = 44100;
```

4.5.4.11 Program Clock Reference (PCR) PID

Dieser Parameter setzt den PID der „Program Clock Reference“ (PCR). Er ist normalerweise auf den PID gesetzt, welcher den Video-Stream enthält.

Beispiel:

```
pcr pid = 0x20;
```

4.5.4.12 Video PID

Dieser Parameter setzt den PID des Video-Streams.

Beispiel:

```
video pid = 0x20;
```

4.5.4.13 Audio PID

Dieser Parameter setzt den PID des Audio-Streams. Die Audio PID muß sich von der PCR und der Video PID unterscheiden.

Beispiel:

```
audio pid = 0x21;
```

4.5.4.14 Program Map Table (PMT) PID

Dieser Parameter setzt den PID der „program map table“ (PMT). Der PMT sagt dem Empfänger, welcher PID zu dem TV-Kanal gehört und verlangt seinen eigenen PID.

Beispiel:

```
pmt pid = 0x22;
```

4.5.4.15 Kanalname des Programms

Dieser Parameter setzt den Programmnamen des TV-Kanals. Der Programmname ist codiert in der SI-Tabelle, welche eine Kanal-Identifizierung im Empfänger ermöglicht.

Beispiel:

```
callsign = "Program1";
```

4.5.4.16 Sprache

Dieser Parameter identifiziert die Sprache des TV-Kanals. Er sollte auf „eng“ für Englisch oder „DEU“ für Deutsch gesetzt werden.

Beispiel:

```
language = "eng";
```

4.5.4.17 PID filter

Der PID Filter erlaubt dem Anwender, selektiv bestimmte PIDs auf dem entsprechenden TS-Eingang zu blockieren oder zu erlauben. Dabei sind zwei Strategien möglich: 1. *Alle PIDs erlauben, Auflisten der zu blockierenden PIDs* oder 2. *Blockiere alle PIDs als Voreinstellung, Auflisten der erlaubten PIDs*.

Voreinstellung ist „alle“ oder „keine“, die Ausnahmen sind mit `minus pid/mask` und `plus pid/mask` markiert. `pid` spezifiziert das passende Zahlwort und `mask` spezifiziert das zu vergleichende Bitmuster. Das resultierende PID filter muß den PIC 8192 (0x1fff), (das NullPacket PID) blockieren. Maximal sind vier Filter setzbar. Bei einer Angabe von mehr als vier Filtern werden nur die ersten vier beachtet, alle danach folgenden werden ignoriert.

Beispiele:

```
pidfilter = all minus 0x1ffe/0x1ffe;  
pidfilter = none plus 0x0020/0x1ffe;
```

4.5.4.18 Tunermodus

Dieser Parameter spezifiziert, ob ein Empfänger an dem Port angeschlossen ist. Folgende Tunermodi bestehen:

off	kein Tuner angeschlossen
dfm	DFM-Analog-Tuner angeschlossen am MPEG2-Encoder
mb86a15	Fujitsu DVB-S-Tuner direkt am TS-Port angeschlossen
nxt6000	DVB-T-Tuner angeschlossen
sv297	DVB-C-Tuner angeschlossen

Beispiel:

```
tuner mode = off;
```

4.5.4.19 Tunerfrequenz

Dieser Parameter hat nur einen Effekt, wenn der Tunermodus nicht auf `off` gesetzt ist. Er gibt die Frequenz an, auf die der Tuner gestellt werden soll.

Beispiel:

```
tuner frequency = 1260M;
```

4.5.4.20 Tuner FEC Modus

Dieser Parameter hat nur einen Effekt, wenn der Tunermodus auf `mb86a15` (siehe Kap. 4.5.4.18) gesetzt ist. Er spezifiziert, welche inneren FEC Einstellungen getestet werden sollen, bis ein gültiges Signal gefunden wird. Er kann entweder auf `auto` oder auf eine innere FEC-Reihe gesetzt werden.

Beispiel:

```
tuner FEC = auto;
```

4.5.4.21 Tuner Symbolrate

Dieser Parameter hat nur einen Effekt, wenn der Tunermodus auf `mb86a15` (siehe Kap. 4.5.4.18) gesetzt ist. Er spezifiziert, welche Symbolrate erwartet werden soll.

Beispiel:

```
tuner symrate = 3000k;
```

4.5.5 Teletext-Teil

4.5.5.1 Program Clock Reference (PCR) PID

Dieser Parameter setzt den PID des „Program Clock Reference“ (PCR). Er ist normalerweise auf den PID, welcher den Video-Stream beinhaltet, gesetzt.

Beispiel:

```
pcr pid = 0x20;
```

4.5.5.2 Video PID

Dieser Parameter wählt den PID des Video-Streams, welcher das Standbild enthält.

Beispiel:

```
video pid = 0x20;
```

4.5.5.3 Teletext PID

Dieser Parameter wählt den PID des Teletext-Streams.

Beispiele:

```
teletext pid = 0x21;
```

4.5.5.4 Program Map Table (PMT) PID

Dieser Parameter setzt den PID der „Program Map Table“ (PMT). Der PMT sagt dem Empfänger, welcher PID zu dem TV-Kanal gehört und verlangt seinen eigenen PID.

Beispiele:

```
pmt pid = 0x22;
```

4.5.5.5 Program Callsign

Dieser Parameter setzt den Programmnamen des Teletext/Standbild-Kanals. Der Programmname ist codiert in der SI-Tabelle, welche eine Kanalidentifizierung ermöglicht.

Beispiel:

```
callsign = "DVB Test";
```

4.5.5.6 Spracheinstellung

Dieser Parameter identifiziert die Sprache des Teletext/Standbildkanals. Er sollte auf `eng` für Englisch oder `DEU` für Deutsch gesetzt werden.

Beispiel:

```
language = "eng";
```

4.5.5.7 Bild-Dateien

Dieser Parameter spezifiziert die Datei, welche das Bild zum Übertragen im Standbild-Kanal beinhaltet. Der Kanal ist dazu bestimmt, beispielsweise ein Logo des Operators anzuzeigen.



Beachten Sie, daß dieser Kanal nicht 100% DVB-kompatibel ist, so daß keine Garantie gegeben werden kann, daß alle Empfänger diesen Kanal anzeigen können.

Die Datei muß entweder vom JPEG-Typ sein oder einen MPEG2 „elementary stream“ beinhalten. MPEG2-Softwareencoder erzeugen üblicherweise Stream-Dateien, die nicht kompatibel sind. Wenn die Datei ein JPEG Bild (muß 704×576 Pixel groß sein) enthält, kann mittels des mpeg2enc-Programms von mjpgtools ein kompatibler „elementary stream“ erzeugt werden.

Die Binärdatei muß im selben Verzeichnis wie DATVfwtools unter Windows oder im `usr/bin/` Verzeichnis unter Linux zu finden sein.

Beispiel:

```
picture file = "mylogo.mpg";
```

Zum Konvertieren von Bildern in eine mpg-Datei kann beispielsweise „Omniformat“, erhältlich unter www.omniformat.com, benutzt werden.

4.5.5.8 VM Code

Dieser Parameter spezifiziert die Datei, welche den *teletext encoder virtual machine bytecode* enthält. Nähere Informationen zum Teletext-Encoder VM-Bytecode siehe Kapitel 4.8.

Beispiel:

```
vm code = "teletext.o";
```

4.5.6 Extern-Programm-Teil

Dieser Sender kann nicht nur lokal encodierte Programme senden, sondern auch Programme von anderen Quellen wie z.B. DVB-S-Receivern oder PCs.

Um Receivern das Auffinden solcher Programme zu ermöglichen, muß der Sender die Program Map Tables (PMT) dieser Programme ebenfalls mitsenden und sie in der Program Association Table (PAT) aufführen.

4.5.6.1 Program Clock Reference (PCR) PID

Dieser Parameter enthält die PID der Program Clock Reference (PCR). Er wird normalerweise auf die PID gesetzt, die den Video-Stream enthält.

Beispiel:

```
pcr pid = 0x420;
```

4.5.6.2 Program Map Table (PMT) PID

Dieser Parameter enthält die PID der Program Map Table (PMT). Die PMT enthält Informationen für den Receiver, welche PID welchen TV-Kanal enthält und besitzt selbst eine PID.

Beispiel:

```
pmt pid = 0x422;
```

4.5.6.3 Sprache

Dieser Parameter gibt die Sprache des TV-Kanals an. Er sollte auf `eng` für Englisch oder `DEU` für Deutsch gesetzt werden.

Beispiel:

```
language = "eng";
```

4.5.6.4 Stream subsections

Die Stream subsections (etwa: Streamunterteilungen) (`video stream`, `audio stream`, `teletext stream` und `stream`) entsprechen den individuellen Strömen, die zusammen ein Programm ergeben.

4.5.6.5 PID

Dieser Parameter gibt die Stream-PID an.

Beispiel:

```
pid = 0x440;
```

4.5.6.6 Stream Typ

Dieser Parameter gibt den Typ des Streams an (siehe [1, Tabelle 2-36]). Dieser Parameter kann nicht manuell für Audio-, Video- und Teletextstreams gesetzt werden (dies geschieht automatisch).

Beispiel:

```
stream type = "0x80";
```

4.5.6.7 Stream ID

Dieser Parameter setzt die ID des Streams, es handelt sich um die Nummer des Streams.

Beispiel:

```
stream id = 1;
```

4.5.6.8 Component Type

Dieser Parameter setzt den Component Type des Streams (siehe [2, Tabelle 24]).

Beispiel:

```
component type = 1;
```

4.5.6.9 Language

Dieser Parameter gibt die Sprache des Streams an. Er sollte auf `eng` für Englisch oder `DEU` für Deutsch gesetzt werden.

Beispiel:

```
language = "eng";
```

4.6 Der alte Teletext-Encoder

Der alte Teletext-Encoder besteht aus einer statischen Tabelle von Teletextseiten mit ihren Zeilen. Er erlaubt wenig dynamische Inhalte und keine Kontrolle über den Encodierprozeß. Er wird vermutlich in Zukunft aus der Konfiguration entfernt. Abb. 4.6a zeigt Konfigurationsanweisungen einer beispielhaft zu encodierenden Teletextseite.

```
teletext {
  page header = "www.D-ATV.de \x92\x20\x08";
  page {
    number = 100;
    line 1 = "";
    line 2 = "\x01 www.D-ATV.de";
    line 3 = "";
    line 4 = "Digital Baseband:";
    line 5 = " Thomas Sailer, HB9JNX/AE4WA";
    line 6 = "";
    line 7 = "RF";
    line 8 = "Wolf-Henning Rech, DF9IC/N1EOW";
    line 9 = "Jens Geisler, DL8SDL";
    line 10 = "";
    line 11 = "Schematics, Boards &";
    line 12 = " Connections to Fujitsu";
    line 13 = " Stefan Reimann, DG8FAC";
    line 14 = "";
    line 15 = "\x03adacom e.V.";
  };
};
```

Abb. 4.6a: Konfigurationsdatei für den alten Teletext-Encoder

4.6.1 Der Teletext-Teil

4.6.1.1 Teletext Seitenkopf

Dieser Parameter setzt die Inhalte des Seitenkopfes (oberste Textzeile), der rechts von der Seitenzahl angezeigt wird.

Beispiel:

```
page header = "www.D-ATV.de \x92\x20\x08";
```

4.6.2 Teletextseiten

Der `teletext`-Teil enthält Seitenuntergliederungen, startend mit einer öffnenden Klammer `{` und endend mit der Zeichenfolge `};`. Jeder dieser Bereiche beschreibt jeweils eine einzelne Teletextseite.

4.6.2.1 page number (Seitenzahl)

Dieser Parameter spezifiziert die Teletextseitenzahl. Der Wert muß zwischen 100 und 899 (jeweils inklusive) sein. Teletextdecoder beginnen automatisch mit Seite 100, deshalb sollten die Seite 100 vorhanden und mit Einführungsinformationen versehen sein.

Beispiel:

```
page number = 100;
```

4.6.2.2 Teletext lines (Teletextzeilen)

Dieser Parameter spezifiziert die einzelnen Zeilen einer Teletextseite. Die Zeilen sind von 1–24 durchnummeriert. Teletextzeilen können bis zu 40 Zeichen lang sein. Kürzere Zeilen werden mit Leerzeichen aufgefüllt. Nichtdruckbare Zeichen können durch Eingabe eines „Backslash“ \, gefolgt von einem x und einer zwei Digit langen Hexadezimalzahl, welche den Zeichencode beinhaltet eingegeben werden. Um zum Beispiel das Zeichen 1 (0x01) einzugeben, muß im File \x01 stehen. Die Zeichencodes von 0-31 (0x00–0x1f) werden für *ETSI Teletext Attribute Markup* verwendet (z.B. Farben), die Zeichencodes 128-255 (0x80–0xff) um dynamische Daten einzugeben, wie etwa Packet Counters.

Beispiel:

```
line 2 = "\x01 www.D-ATV.de";
```

4.7 Beispielkonfiguration

Abb. 4.7a zeigt eine einfache, minimale Konfigurationsdatei. Es wird angenommen, daß ein MPEG2-Encoder an TS4 angeschlossen ist und TS1–TS3 nicht benutzt werden.

```
# Sample configuration file for DATVfwtool

board {
    clock = 60M;
    board = ham;
    generate tables = on;
};

modulator {
    modulation = DVB-S;
    fec = 3/4;
    frequency = 1255M;
    symbol rate = 5000k;
    transport stream id = 0x01234;
    network id = 0x05678;
    network name = "Demo TV";
};

transportstream 1 {
    mode = off;
};
transportstream 2 {
    mode = off;
};
transportstream 3 {
    mode = off;
};
transportstream 4 {
    mode = datvencoder;
    spatial filter = standard;
    video input = dl pal composite;
    bitrate = 6000k;
    audio sample rate = 48000;
    audio bitrate = 192k;
    service id = 0x41;
    service provider name = "Demo TV";
    service name = "Program 1";
    event name = "Program 1";
    event text = "Encoder 1";
    language = "DEU";
};

teletext {
    service id = 0x4242;
    service provider name = "Demo TV";
    service name = "Demo TV";
    event name = "Demo TV";
    event text = "Demo TV on air";
    callsign = "Testbild";
    language = "DEU";
    picture file = "testbild.mpg";
    vm code = "teletext.o";
};
```

Abb. 4.7a: Beispiel einer einfachen, vollständigen Konfigurationsdatei

4.8 Der neue Teletext-Encoder

Der neue Teletext-Encoder erlaubt die volle Kontrolle des Encoding-Prozesses und dynamische Inhalte. Er wird ausgeführt mittels eines Bytecode-Programms, das in einer stack-basierenden, virtuellen Maschine interpretiert wird. Telecode-Teletextprogramme müssen nicht in der stack-basierten Assembler-Sprache der virtuellen Maschine geschrieben werden, sondern können in C programmiert und dann in einen Bytecode kompiliert werden. Die folgende Tabelle zeigt die Executables des Bytecode-Entwicklungssystems:

cpp	C Preprocessor
rcc	C Compiler proper
vm	VM Simulator
vmar	Bytecode Archiver
vmas	Bytecode Assembler
vmdisass	Bytecode Disassembler
vmld	Bytecode Linker
atv2txtvm	Konvertierungstool von DG9MHZ-ATV-Files zu VM Teletext Source Code
Tabelle 4.8a: Executables des Bytecode-Entwicklungssystems	

Angenommen, der Teletext-Encoder-C-Code ist in der Datei `teletext.c` beinhaltet, kann der C-Code durch folgendes Kommando kompiliert und assembliert werden in die Objectcode-Datei `teletext.o`:

```
vmas -c -o teletext.o teletext.c
```

Der Objektcode kann disassembliert werden durch:

```
vmdisass teletext.o
```

Der Objektcode kann simuliert werden durch:

```
vm -c -l -m teletext teletext.o
```

Abb. 4.8b zeigt ein Beispiel des Sourcecodes eines Teletext-Encoders:

Tabelle 4.8b: Beispiel eines Teletext-Encoder-Sourcecodes

4.8.2 VM Built-In Library Functions

4.8.2.1 C Typ-Größen

Typ	Bits
char	8
short	16
int	32
long	32

4.8.2.2 C99 Standard-Makros

`NULL`, `offsetof`

4.8.2.3 C99 Standard-Typen

`ptrdiff_t`, `size_t`, `int8_t`, `u_int8_t`, `int_16_t`, `u_int16_t`, `int32_t`, `u_int32_t`

4.8.2.4 C99 Standard-Funktionen

`memcpy`, `memmove`, `strcpy`, `strncpy`, `strcat`, `strncat`, `memcmp`, `strcmp`, `strncmp`, `memchr`, `strchr`, `strcspn`, `strpbrk`, `strchr`, `strspn`, `strstr`, `memset`, `strlen`, `exit`

4.8.2.5 Event-Log-Funktionen

<code>void logreadinit(unsigned int *p);</code>	
Setzt den Event-Log-Zeiger auf die älteste Nachricht im Buffer	
<code>p</code>	a pointer to an opaque cookie of type <code>unsigned int</code>

<code>unsigned int logreadline(unsigned int *p, char *buf, unsigned int bufsz);</code>	
Liest die nächste Event-Log-Nachricht ein	
<code>p</code>	Zeiger auf einen opaque cookie des Typs <code>unsigned int</code>
<code>buf</code>	Zeiger auf einen Buffer ausreichender Größe
<code>bufsz</code>	die Größe des Buffers
<code>logreadline</code>	Gibt die Anzahl der Zeichen zurück, die im Buffer gespeichert und nicht Null sind. Ein Rückgabewert von Null bedeutet, daß keine weiteren Ereignisse im Event-Log-Buffer mehr vorhanden sind.

4.8.2.6 Zeit- und Datumsfunktionen

struct time	
<code>day</code>	Modifiziertes julianisches Datum (Anzahl der Tage seit dem 17.11.1858)
<code>sec</code>	Anzahl der Sekunden seit Mitternacht
<code>msec</code>	Anzahl der Millisekunden der zuletzt angefangenen Sekunde
<code>valid</code>	wenn <code>valid</code> gesetzt ist, wurden Zeit und Datum per serieller Schnittstelle oder DCF77 eingestellt.

struct timehms		struct date
h Stunden		d Tag
m Minuten		m Monat
m Sekunden		y Jahr

<code>struct time gettime(void);</code>
gibt die aktuelle Zeit zurück

<code>u_int32_t getjiffies(void);</code>
gibt eine monoton steigende Nummer zurück. Sie erhöht sich HZ mal pro Sekunde

<code>struct date mjdto date(u_int16_t mjd);</code>
konvertiert ein modifiziertes julianisches Datum in ein standard-gregorianisches Datum

<code>u_int16_t datetomjd(u_int16_t d, u_int16_t m, u_int16_t y);</code>
konvertiert ein standard-gregorianisches Datum in ein modifiziertes julianisches Datum

<code>char *timedec(char *buf, struct timehms *hms, u_int32_t tm);</code>
nimmt die Sekunden seit Mitternacht und konvertiert sie in Stunden, Minuten und Sekunden und in eine menschenlesbare Form 01:23:45 <i>hms</i> und <i>buf</i> sollten NULL sein. Die Funktion gibt einen Zeiger auf <i>buf[0]</i> zurück.

4.8.2.7 Parameter und Statistik-Funktionen

<code>u_int16_t getadc(unsigned int n);</code>
Gibt den Wert des A/D-Wandlers zurück. <i>n</i> reicht von 0-3, und der Rückgabewert des A/D-Wandlers bewegt sich zwischen 0 und 1023, entsprechend einer Eingangsspannung von 0 bis 5 Volt.

<code>u_int32_t readcounter(unsigned int n);</code>
Gibt den Wert des Zählers <i>n</i> zurück.

```
u_int8_t get_inversion(void);
```

Gibt die „spectral inversion“-Einstellung zurück

<i>n</i>	<i>Wert</i>
0	Local PCR (Program Clock Reference)
1	Total packet count
2	Mux-generated NULL packets
3	Table/teletext packets
4	Transport stream 1 packets
5	Transport stream 2 packets
6	Transport stream 3 packets
7	Transport stream 4 packets
8-15	unused

```
u_int8_t get_fecmode(void);
```

Gibt den FEC-Modus zurück.

<i>Rückgabewert</i>	<i>FEC-Modus</i>
0	Local PCR (Program Clock Reference)
1	Total packet count
2	Mux-generated NULL packets
3	Table/teletext packets
4	Transport stream 1 packets

```
u_int32_t get_frequency(void);
```

Gibt die Übermittlungsfrequenz in kHz an.

```
u_int8_t get_ptt(void);
```

returns whether the PTT is keyed.

4.8.2.8 Numeric to String conversion

flags

INTCONV_SIGN	Number is signed
INTCONV_PLUS	write an explicit + if a signed number is positive
INTCONV_PADZERO	pad buffer to the left with zeros
INTCONV_PADSPACE	pad buffer to the left with spaces
INTCONV_LOWERCASE	use lower case hexadecimal characters

```
char *int2hex(char *buf, u_int16_t len, u_int32_t val, u_int16_t flags);
```

konvertiert `val` in einen Hexadezimalstring, der in den Buffer `buf` gespeichert wird. Bis zu `len` Zeichen werden gespeichert, `buf` muß mindestens `len+1` Zeichen groß sein. Die Funktion gibt einen Zeiger auf den Nummernstring zurück, der sich in `buf` befindet, jedoch nicht notwendigerweise auf dessen Anfang.

```
char *int2dec(char *buf, u_int16_t len, u_int32_t val, u_int16_t flags);
```

konvertiert `val` in einen Dezimalstring, der in den Buffer `buf` gespeichert wird. Bis zu `len` Zeichen werden gespeichert, `buf` muß mindestens `len+1` Zeichen groß sein. Die Funktion gibt einen Zeiger auf den Nummernstring zurück, der sich in `buf` befindet, jedoch nicht notwendigerweise auf dessen Anfang.

4.8.2.9 TS1/TS2 table decoder

Der TS1/TS2 table decoder versucht, die Daten der SI-Tabellen aus den Transportströmen an Port 1 und Port 2 zu extrahieren.

struct portcapture	
event_id	wird inkrementiert, wenn die Service Descriptor Information Table ein Update erfährt
transport_stream_id	Transport Stream ID
nit_pid	PID, mit der die Network Information Table übertragen wird.
service_id	Service ID
network_id	Network ID
service_provider_name	Service Provider Name
service_name	Service Name
Für nähere Informationen über die DVB System Information (SI) tables, siehe [2].	

<code>struct portcapture(unsigned int port);</code>							
returns SI table data for transport stream port Gibt SI-Tabellendaten für den Transportstreamport zurück	<table> <tr> <th>port</th><th>Transport Stream</th></tr> <tr> <td>0</td><td>TS1</td></tr> <tr> <td>1</td><td>TS2</td></tr> </table>	port	Transport Stream	0	TS1	1	TS2
port	Transport Stream						
0	TS1						
1	TS2						

4.8.2.10 Highlevel teletext encoding functions

<code>void teletext encodepage(u_int16_t startline, u_int16_t endline, u_int16_t pagenr, u_int16_t subnr, u_int32_t flags, const char **lines, ...);</code>	
Encodiert mehrere Teletextzeilen, von <code>startline</code> bis <code>endline</code>	
startline	ist normalerweise 0
endline	ist normalerweise 24
pagenr	spezifiziert eine Seitenzahl und sollte zwischen 0x100 und 0x8ff liegen. Seitenzahlen, die Hexadezimalziffern A-F enthalten, sind über den Receiver normalerweise nicht direkt erreichbar.
subnr	spezifiziert die Unterseitenzahl (normalerweise 0)
flags	can be zero or multiple TXTPAGECTRL macros ored.
TXTPAGECTRL	sind detailliert beschrieben bei [3, 9.3.13, p.27]
lines	enthält einen Zeiger auf ein Array von <code>endline-startline+1</code> Strings. Jeder String spezifiziert den Inhalt jeweils einer Zeile. Ein NULL-Pointer unterdrückt das Encodieren der zugehörigen Zeile. Teletextzeilen können auch <code>TXSTATTR</code> -Makros oder <code>TXN_ARGN</code> -Argumente enthalten.
TXSTATTR	Makros (siehe [3, 12.2, p. 76–80])
TXN_ARGN	referenziert auf optionale Argumente. Kann bis zu 64 Zeiger beinhalten, die auf Strings verweisen.

4.8.2.11 Lowlevel Teletext encoding functions

<code>void teletext oddparity(u_int8_t *buf, const u_int8_t *src, unsigned int len);</code>	
encodiert einen Datenbuffer, beginnend bei <code>src</code> mit der Länge <code>len</code> , Teletext-ungleich-Parität benutzend, und speichert es in <code>buf</code> .	

```
void teletext hamming84(u_int8_t *buf, const u_int8_t *src, unsigned
int_nibblelen);
```

encodiert einen Datenbuffer, beginnend bei `src` der `len` Nibbles mit Teletext 8/4-Hamming-Code in `buf` speichert. Zuerst wird das niederwertige Nibble von `src[0]` codiert, dann das höherwertige Nibble von `src[0]`, dann das niederwertige von `src[1]` und so weiter.

```
void teletext hamming2418(u_int8_t *buf, const u_int8_t *src, unsigned int
len);
```

encodiert einen Datenbuffer, beginnend bei `src`, der `len` Triples mit Teletext-24/18-Hamming-Code in `buf` speichert. `src[0]` enthält dabei die niederwertigsten 6 Bits, `src[1]` die mittleren 6 Bits und `src[2]` die höchstwertigen 6 Bits.

```
u_int8_t *teletext currentline(void);
```

Gibt einen Zeiger auf den Buffer der aktuellen Zeile zurück. Der Zeilenbuffer ist 42 Bytes groß und enthält eine komplette Teletext-Zeile ohne clock run-in und framing-code [3, 7.1, p.17ff]

```
u_int8_t *teletext waitline(void);
```

Sendet die aktuelle Zeile und gibt einen Zeiger auf den Buffer der nächsten Zeile zurück

4.9 Anschluß eines PC-Parallelports an einen Transportstream-Input

Der Parallelport eines PC kann als eine einfache Möglichkeit für einen langsamen Datentransfer in den Transportstream benutzt werden. Bis zu 2 MBit/s sind möglich. Tabelle 4.9a zeigt, wie der Parallelport an TS1 oder TS2 angeschlossen werden muß. Der Eingangsport muß auf Modus eingestellt und der TT clock filter sollte auf das Maximum 4 eingestellt werden.

<i>Pin</i>	<i>Parport-Signal</i>	<i>TS-Signal</i>
1	nStrobe	TS-CLK
2	D0	D0
3	D1	D1
4	D2	D2
5	D3	D3
7	D4	D4
8	D5	D5
9	D6	D6
10	D7	D7
11	nAck	SDOUT
12	Busy	ASCLK
13	Perror	SCLK
14	Select	SDIN
15	nAutoFd	TS-SY
16	nFault	XRESET
17	nInit	TS-VL
18	nSelectIn	TS-EN
19	GND	GND

Tabelle 4.9a: Anschlußschema Parallelport – TS Input

5 Anhang

5.1 Danksagung

Wir danken den Firmen [Wacker's Kaffee](#) und [Coca Cola](#) für ihre koffeinhaltigen Produkte; Pasquale für Pizza und Steaks, Naçi für Döner, dem [Bären-Treff](#) für Süßes und [Pepperworld](#) für Scharfes.

5.2 Referenzen

- [1] ETSI EN 300 468 V1.4.1 European Standard (Telecommunications series) Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems, 07 2000.
- [2] Detlef Fliegl, DG9MHZ. VTGEN, der Teletextencoder für IBM-kompatible PCs. [http://www.baycom.org/ftp/local/vt/ { vtpack.exe,vtgendoc.zip }](http://www.baycom.org/ftp/local/vt/{ vtpack.exe,vtgendoc.zip }) , November 1995.
- [3] European Telecommunications Standards Institute (ETSI). ETS 300 706: Enhanced Teletext Specification, May 1997.
- [4] Chris Fraser and David Hanson. lcc, A Retargetable Compiler for ANSI C. <http://www.cs.princeton.edu/software/lcc/>

5.3 Autoren

Softwareentwicklung

Thomas Sailer, **HB9JNX**

Schaltungs- und Boarddesign

Stefan Reimann, **DG8FAC**

HF-Design

Prof. Dr. Ing. Wolf-Henning Rech, **DF9IC**

Ing. Jens Geisler **DL8SDL**