DVB

Components

Manual and Reference

The information in this manual is compiled with high care; nevertheless errors can be not completely excluded. We don't take legal or any other responsibility for possibly remained errors and their consequences.

This manual ist protected by copyright laws. All rights, including copying, translation, microfilming as well as storage and processing in electronic systems, reserved.

Comments and questions you can adress to us:

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	First Edition
10. Aug. 2004	Correction of Parallelport-Pinout
26. Nov. 2004	+added board description, +examplifications, updating config documentation.
30. Mar. 2005	translation completed
26. July 2006	V3 of the DVB-S/C-Modulator 4×TS added, completed some configuration parts
11. Nov. 2006	ASI-IO V3 with correct pin no.

Authors: Thomas Sailer, HB9JNX, Stefan Reimann, DG8FAC
Typesetting and editorial office: V-Media, Steinau; www.V-Media.biz

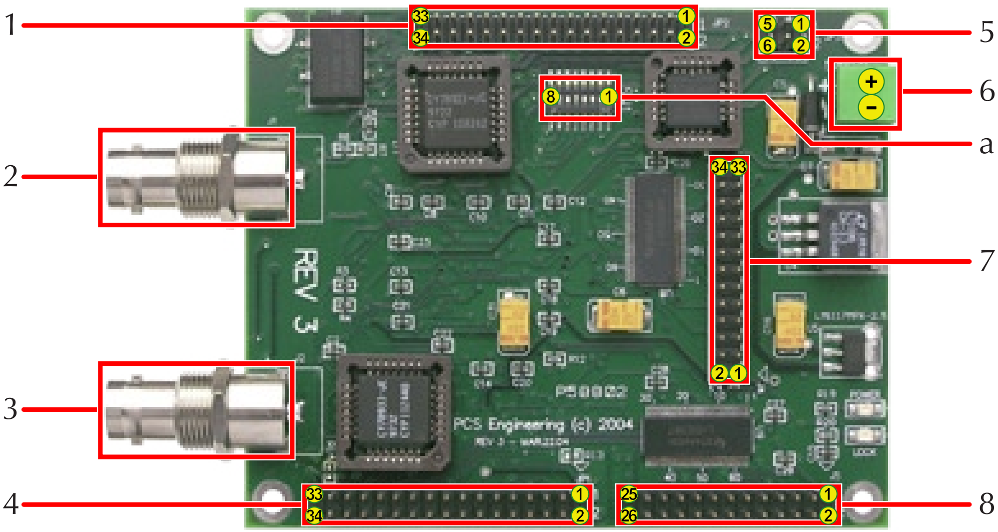
This manual is printed on chlorine-free bleached pdf without cellulose; for its compilation were no animals harmed or killed. Likewise neither were harmed nor killed numerous coffee and other caffeinated dinks, which were very helpful during the compilation of this manual.

I. Table of Contents

1 Board- und Connector descriptions

The following chapters describe the connections and buttons existing on the various boards as well as the meaning of the LEDs on it.

1.1 ASI-I/O-Interface V3



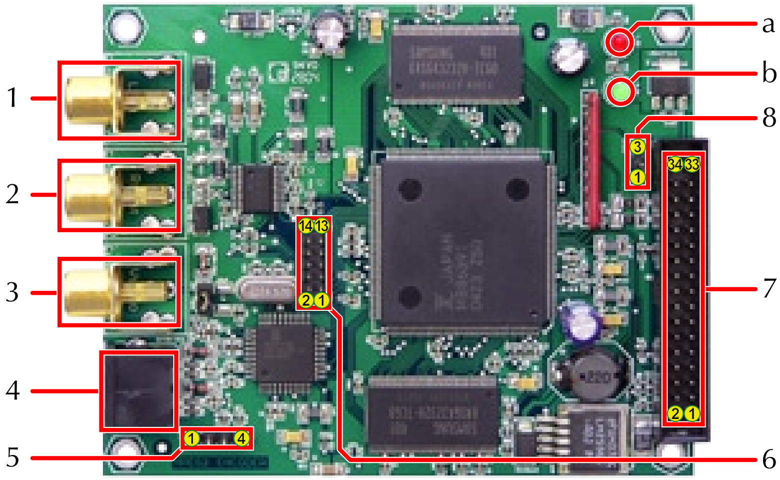
board dimensions: 100×80 mm

Connectors	1	34pin Header	SPI (TS) Input
	2	BNC Connector 75 Ω	ASI Input
	3	BNC Connector 75 Ω	ASI Output
	4	34pin Header	SPI (TS) Output
	5	6pin Header	JTAG Connector for programming the Flash
	6	2-way screw terminal	Power Input 5 V DC only!
	7	26pin Header	LVDS (=LVTTL) Input
	8	26pin Header	LVDS (=LVTTL) Output

Switches — a — 8pin DIP-Switch — set operational mode

Switch Setting								**Operation**
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Result**
on	×	×	×	×	×	×	×	No PSI (PAT/PMT) Tables
off	×	×	×	×	×	×	×	PSI (PAT/PMT) Tables every 400 ms
×	×	off	on	×	×	×	×	ASI Out (3) from ASI In (2) looped via hardware loop back **Note:** ASI In (2) can also connect to LVDS/SPI with Switches 5&6
×	×	on	on	×	×	×	×	ASI Out (3) from LVDS In (7)
×	×	on	off	×	×	×	×	ASI Out (3) from SPI In (1)
×	×	×	×	off	on	×	×	LVDS (8)/SPI (4) Out from ASI In (2)
×	×	×	×	on	on	×	×	LVDS (8)/SPI (4) Out from SPI (1) In via hardwire loop back **Note:** ASI (3) Out with switches 3&4
×	×	×	×	on	off	×	×	LVDS (8)/SPI (4) Out from LVDS (7) In via hardwire loop back **Note:** ASI (3) Out with switches 3&4

1.2 MPEG Encoder V3

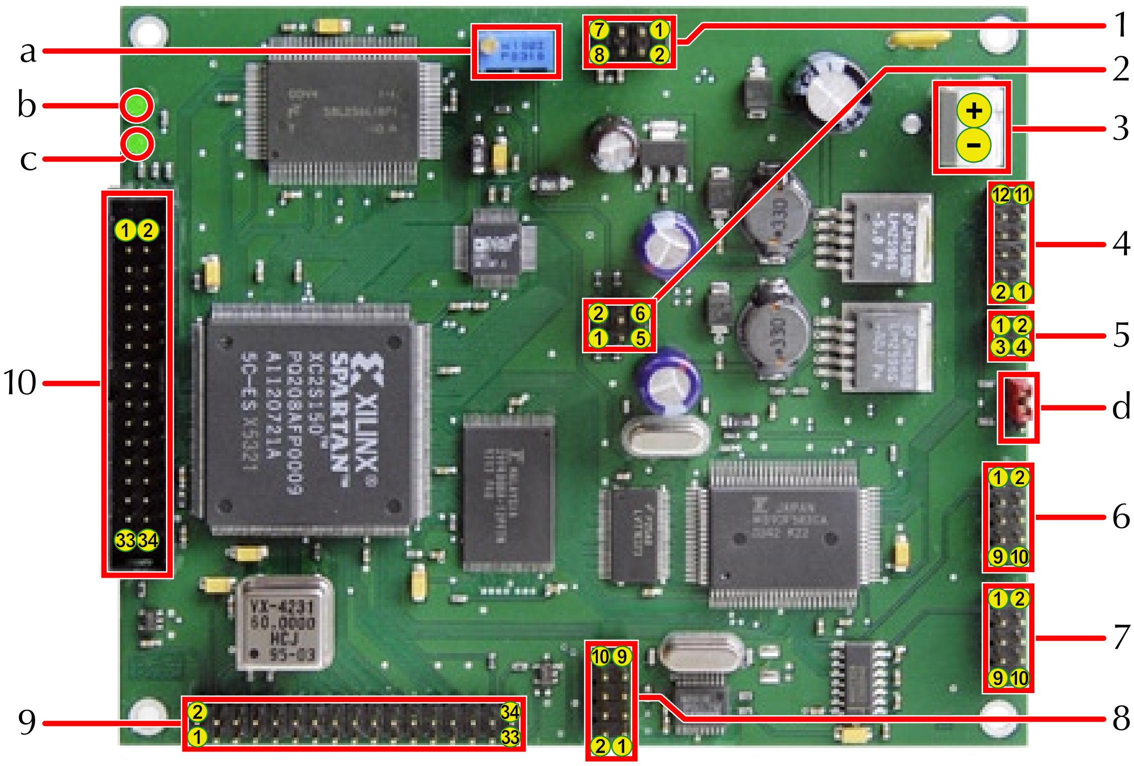


board dimensions: 100×80 mm

Connectors	1	Cinch audio left	0 dB input sensitivity Adjustment of the sensitivity only possible by circuit variation!
	2	Cinch audio right	
	3	Cinch Video	Supported Videostandards: PAL/NTSC
	4	S-Video	
	5	4pol. connector	ext. I²C-Bus (+5 V, GND,...)
	6	14pol. connector	Connector according to ITU-656
	7	34pol. connector	TS_{out} connection to Modulator/Multiplexor
	8	3pol. connector	XError of Encoder-IC and Test-out of Videocontroller (for error-handling in future SW-Versions)

LEDs	a	LED red	Power 1,8 V ok
	b	LED green	Power 3,3 V ok

1.3 DVB-S/C-Modulator 2×TS

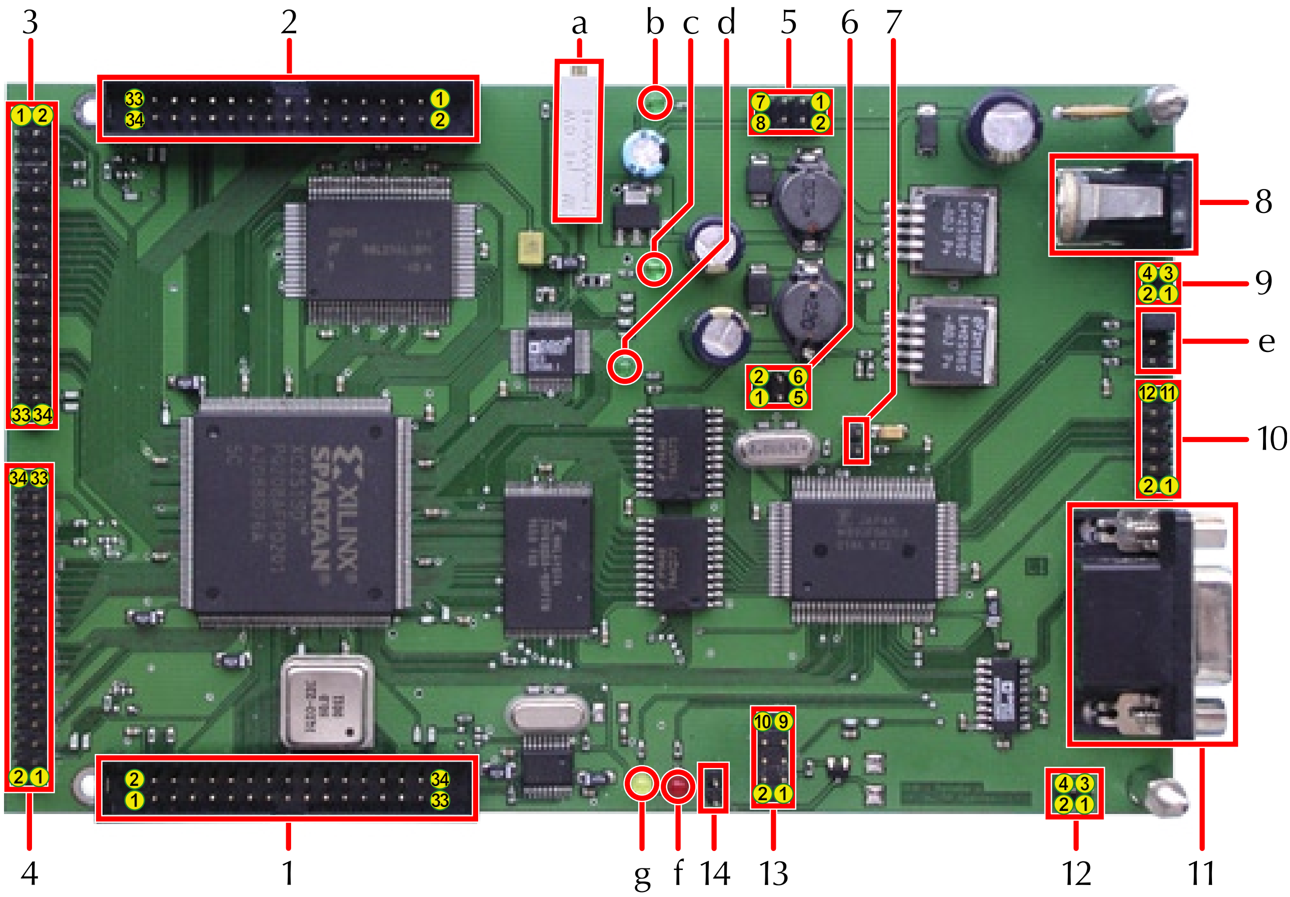


board dimensions: 120×100 mm

Connectors	1	8pin Header	Connector to I/Q-Modulator (U_B)
	2	6pin Header	Connector to I/Q-Modulator (diff. IQ)
	3	2way screw terminal	Power Input (10–24 V DC)
	4	12pin Header	Optional Connector
	5	4pin Header	IE-Bus Connector (not used)
	6	10pin Header	Serial Interface COM1 (used for programming, Baudrate 19k2 8N1)
	7	10pin Header	Serial Interface COM2
	8	10pin Header	Connector to I/Q-Modulator (PLL-Control)
	9	34pin Header	TS1 Input
	10	34pin Header	TS2 Input

LEDs/Switches	a	Potentiometer	for adjusting the I/Q-Balance
	b	LED green	***On:*** µC executes Firmware ***Blinking:*** Encoder-Firmware is being loaded to all encoder boards ***Off:*** Successful start of Encoder SW ***On after off:*** one or more Encoders couldn‘t start. ☞ Note: Encoders can be connected only to the TS-Input, for which these have been programmed! If a tuner/ext. Clock is connected instead an encoder, the LED will be continuous on after switching on the device.
	c	LED green	lightens up after loading and starting code
	d	3pin Jumper	MCU-Mode Jumper (Run/Program)

1.4 a) DVB-S/C-Modulator 4×TS V2

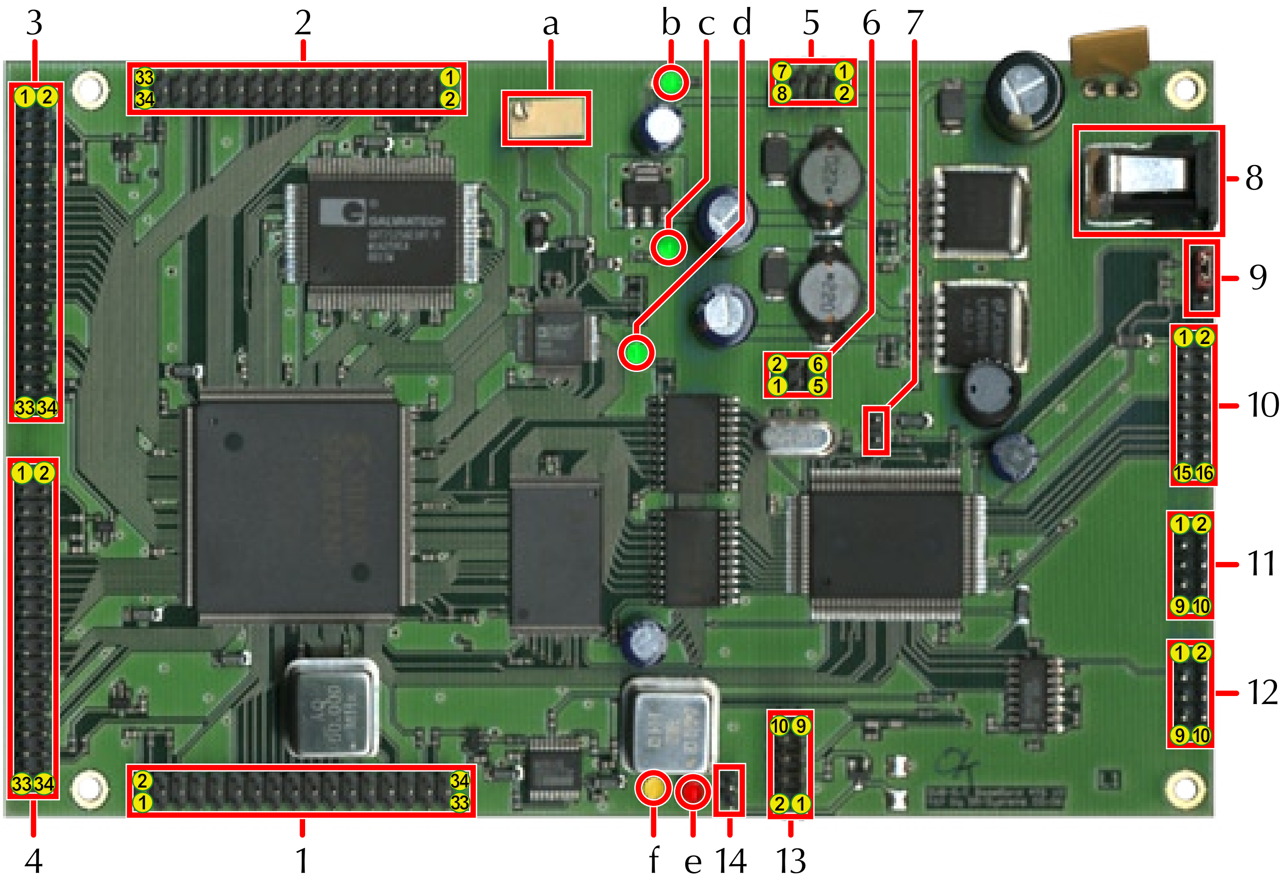


board dimensions: 120×100 mm

Connectors	1–4	Transportstream 1–4 You can use Encoders, DVB-Tuners (S/C/T), Ethernet-Interface and ext. TS-Inputs on the Connectors 1 and 2, Connectors 3 and 4 are **only** for Encoders. For easier identification the connectors 1 and 2 are signed, connectors 3 and 4 are unsigned.
	5	Connection to I/Q-Modulator (diff. IQ)
	6	Connection to I/Q-Mod. (PLL-Control)
	7	ext. Reset
	8	Power Input (10–24 V DC) Attention: positive inside!
	9	IE-Bus connector (not used)
	10	DCF77/analog voltage
	11	Serial Interface COM1 for programming, Baudrate 19k2 8N1
	12	Serial Interface COM2
	13	Connection to I/Q-Modulator
	14	ext. PTT

LEDs/Switches	a	Potentiometer for adjusting of the I/Q-Balance
	b	Power 3,3 V ok
	c	Power 5,0 V ok
	d	Power 2,5 V ok
	e	3×Jumper
	f	***on:*** µC executes firmware ***blinking:*** Download of the Encoder-Firmware ***off:*** Successfull start of Encoder SW ***on after off:*** on or more Encoders couldn‘t start. Note: Encoders can be connected only to the TS-In, for which these have been programmed! If a tuner/ext. Clock is connected instead an encoder, the LED will be continuous on after switching on the device.
	g	lightens up after loading and after starting the firmware

1.4 b) DVB-S/C-Modulator 4×TS V3

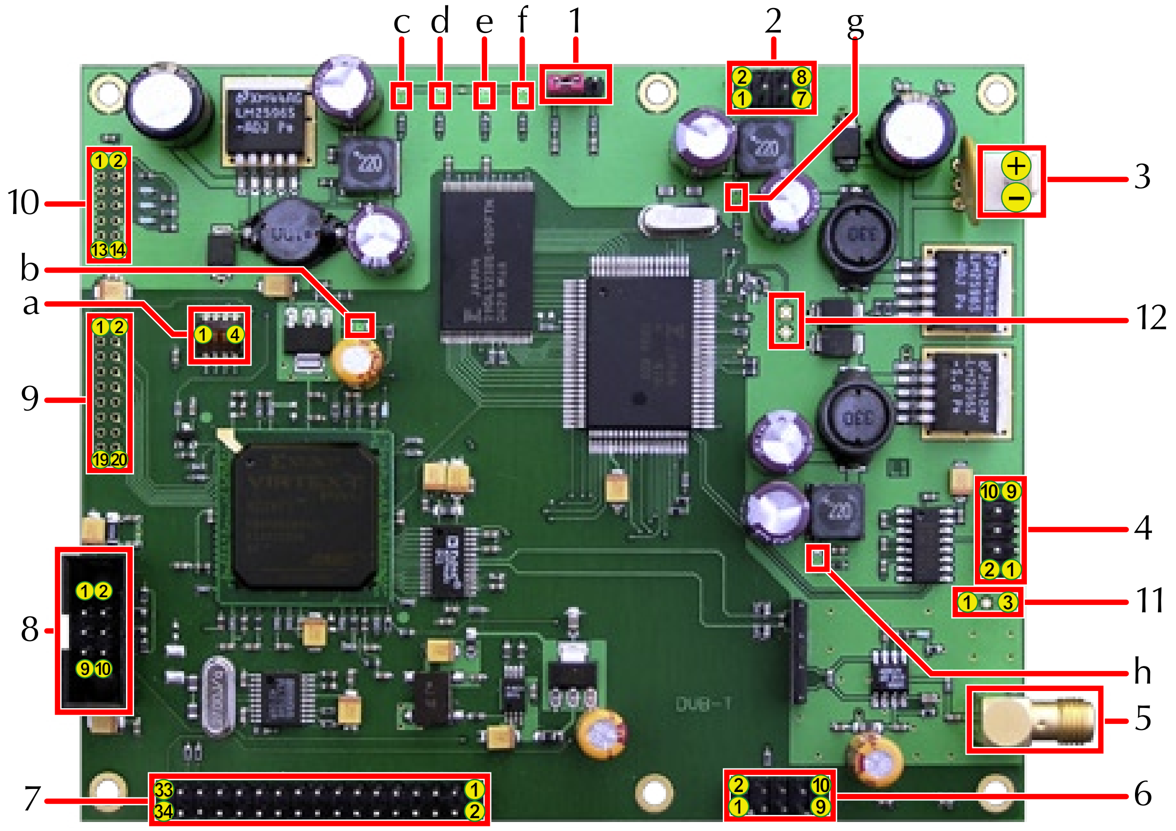


board dimensions: 160×100 mm

Connectors		
	1–4	Transportstream 1–4 You can use Encoders, DVB-Tuners (S/C/T), Ethernet-Interface and ext. TS-Inputs on the Connectors 1 and 2, Connectors 3 and 4 are **only** for Encoders. For easier identification the connectors 1 and 2 are signed, connectors 3 and 4 are unsigned.
	5	Connection to I/Q-Modulator (diff. IQ)
	6	Connection to I/Q-Modulator (PLL-Control)
	7	ext. Reset
	8	Power Input (10–24 V DC) Attention: positive inside!
	9	Mode switch the position shown is *run*, the other is *program*.
	10	optional
	11	Serial interface COM1, RS232 (19k2 8N1)
	12	Serial interface COM2, RS232
	13	Connection to I/Q-Modulator
	14	ext. PTT

LEDs and Switches		
	a	Potentiometer for adjusting the I/Q-Balance
	b	Power 3.3 V ok
	c	Power 5.0 V ok
	d	Power 2.5 V ok
	e	***on:*** µC executes firmware ***blinking:*** Download of the Encoder-Firmware ***off:*** Successfull start of Encoder SW ***on after off:*** on or more Encoders couldn‘t start. Note: Encoders can be connected only to the TS-In, for which these have been programmed! If a tuner/ext. Clock is connected instead an encoder, the LED will be continuous on after switching on the device.
	f	lightens up after loading and after starting the firmware

1.5 DVB-T-Modulator V3

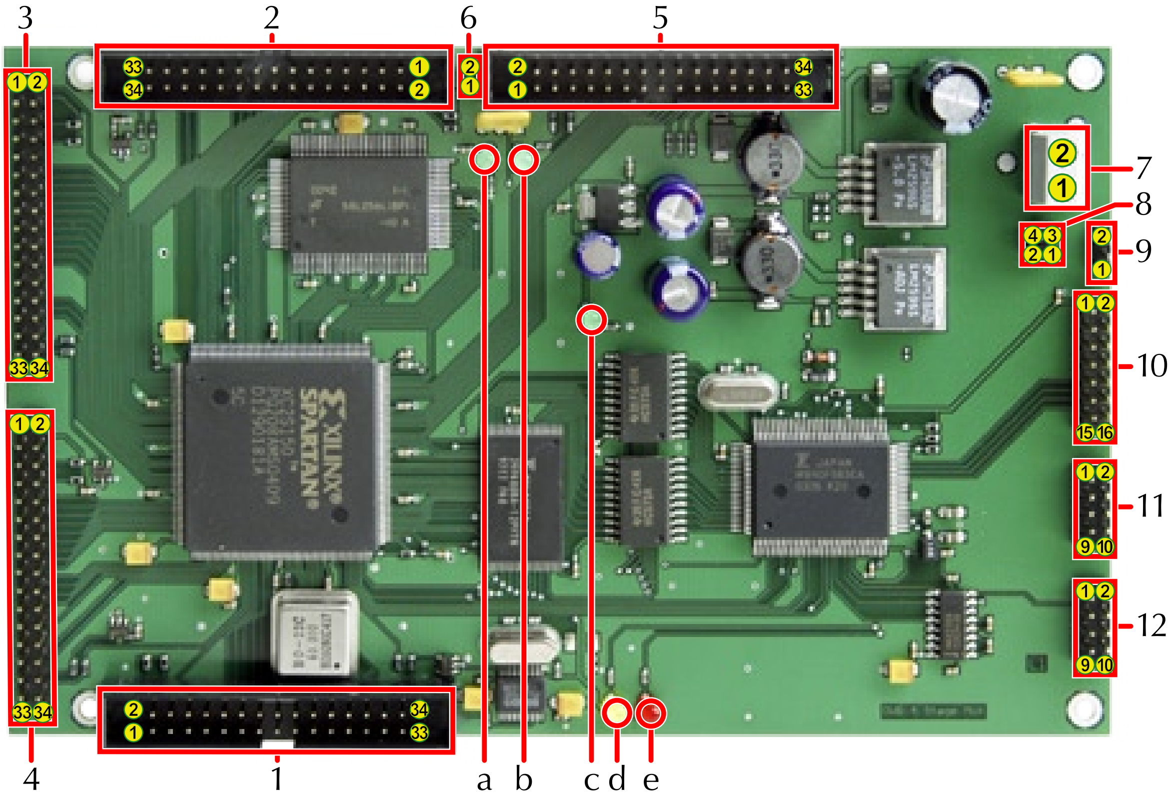


board dimensions: 125×100 mm

Connectors/controls	1	3pin Jumper	Mode jumper. Switches between *program* and *run*
	2	8pin Header	Power supply for Upconverter-Board
	3	Power Input	Power supply for board (9–24 V DC)
	4	10pin Header	RS232 (38k4 8N1)
	5	SMA	RF out (36,125 MHz Baseband)
	6	10pin Header	Control for Upconverter-Board
	7	34pin Header	Transportstream connector for Encoder or Muxboard
	8	10pin Header	I2C-Bus connector for controlpanel

LEDs	a	4pin DIP switch	opt., not used (at the moment)
	b	LED green	Power good (2,5 V)
	c	LED green	opt. Status FPGA
	d	LED green	Firmware running (flashes when running!)
	e	LED green	opt. Status MCU
	f	LED green	Loading Firmware (flashes while loading, off after loading => ok)
	g	LED green	Power good (3,3 V)
	h	LED green	Power good (5,0 V)

1.6 4ch-Multiplexor

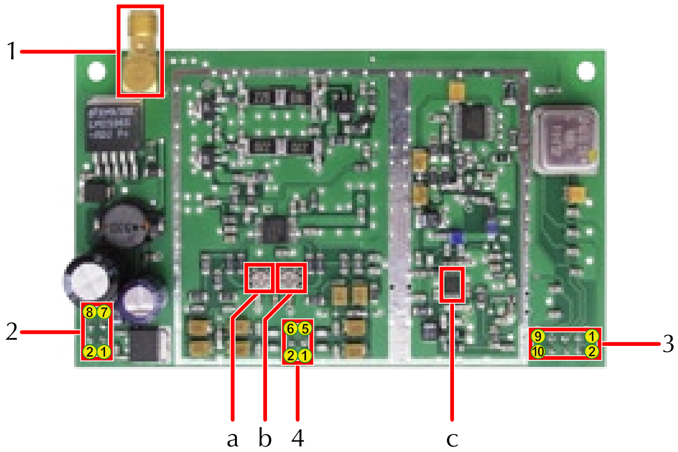


board dimensions: 160×100 mm

Connectors	1-4	34pin Header	Transport Stream Input 1-4
	5	34pin Header	Transport Stream Output
	6	2pin Header	optional, not used yet
	7	2way screw terminal	Power Input (10-24 V DC)
	8	4pin Header	IE-Bus Connector (not used)
	9	3pin Jumper	MCU Mode Jumper (Run/Program)
	10	16pin Header	Optional Connector
	11	10pin Header	Serial Interface COM1 (use for programming)
	12	10pin Header	Serial Interface COM2

LEDs	a	LED green	Indicate External Power TS Output (from other board)
	b	LED green	***Off:*** the internal TS_{Clock} is used ***On:*** an external TS_{Clock} from a board at Connector 5 is used
	c	LED green	Power ok (3,3 V)
	d	LED yellow	lightens up after loading and starting code
	e	LED red	***On:*** µC executes Firmware ***Blinking:*** Encoder-Firmware is being loaded to all encoder boards ***Off:*** Successful start of Encoder SW ***On after off:*** one or more Encoders couldn't start. ☞ Note: Encoders can be connected only to the TS-Input, for which these have been programmed! If a tuner/ext. Clock is connected instead an encoder, the LED will be continuous on after switching on the device.

1.7 I/Q-Modulator UHF V2

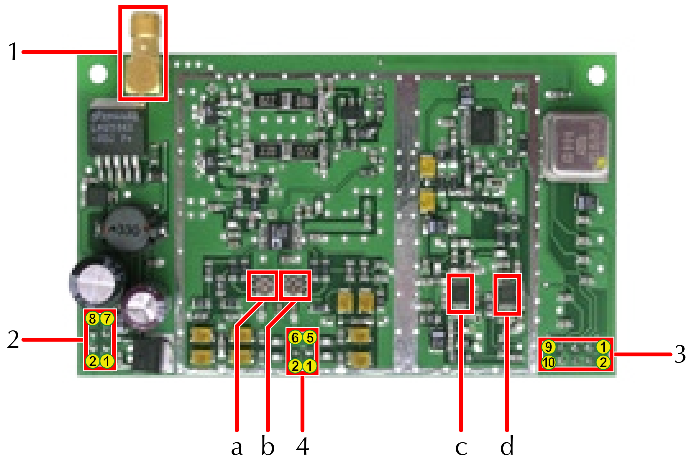


board dimensions: 60×100 mm

Connectors	1	SMA-Connector (50 Ω)	HF Output
	2	8pin Header	Power Input (12 V DC)
	3	10pin Header	Control for PLL etc.
	4	6pin Header	Modulation Input

Others	a	Potentiometer	I-Balance
	b	Potentiometer	Q-Balance
	c	VCO (375–525 MHz)	

1.8 I/Q-Modulator L/S

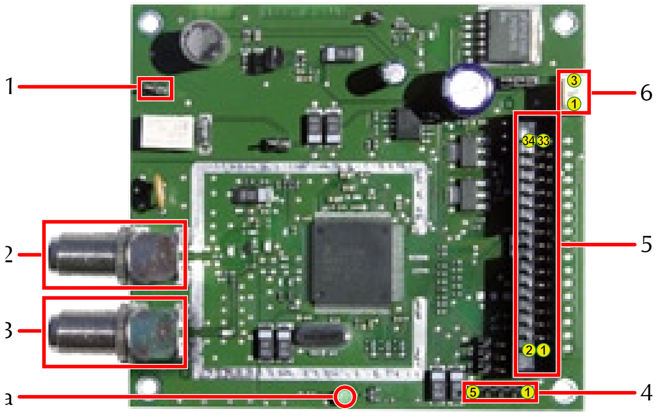


board dimensions: 60×100 mm

Connectors	1	SMA-Connector (50 Ω)	HF Output
	2	8pin Header	Power Input (12 V DC)
	3	10pin Header	Control for PLL etc.
	4	6pin Header	Modulation Input

Others	a	Potentiometer	I-Balance
	b	Potentiometer	Q-Balance
	c	VCO (1175–1325 MHz)	
	d	VCO (2250–2600 MHz)	

1.9 NIM DVB-S V3

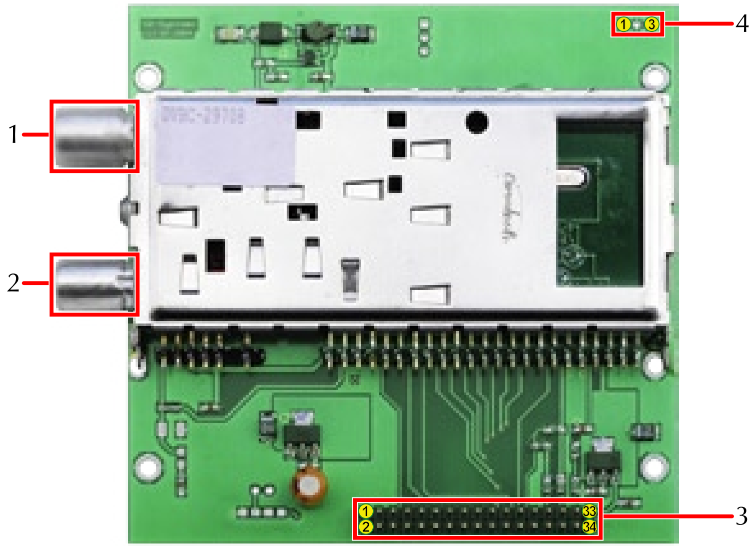


board dimensions: 80×90 mm

Connectors	
	1 LNB Power
	2 Antenna in
	3 Antenna out (loopthrough)
	4 Ext. I²C (unused)
	5 TS Out
	6 LNB Power (20–24 V)

LED	
	a RF Lock

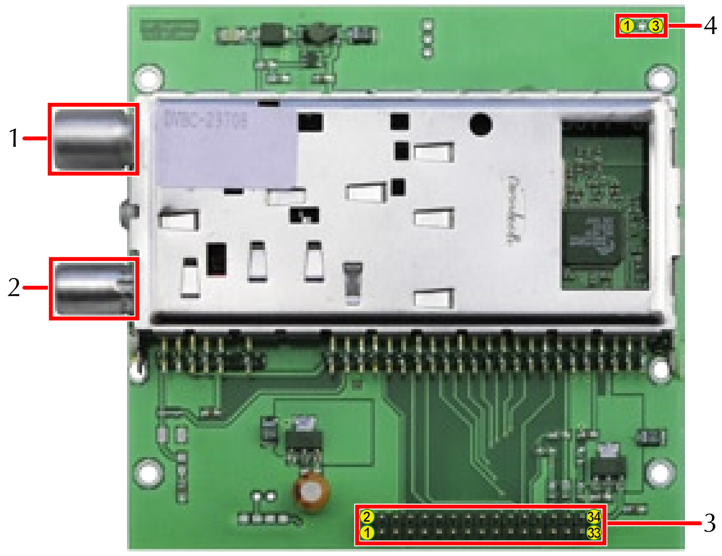
1.10 NIM DVB-C



board dimensions: 100×100 mm

Connectors	1	Antenna in
	2	Antenna out (loop-through)
	3	TS out
	4	LNB Power in

1.11 NIM DVB-T



board dimensions: 100×100 mm

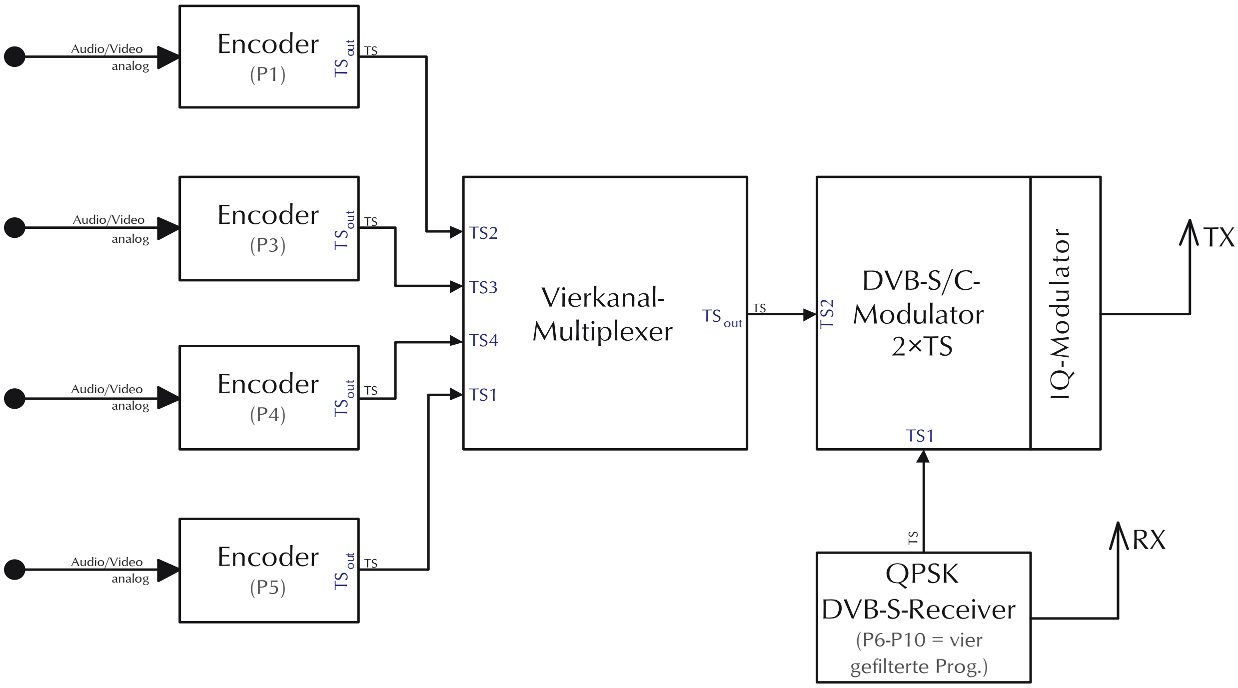
Connectors	
	1 Antenna in
	2 Antenna out (loop-through)
	3 TS out
	4 LNB Power in

2 Examples of application

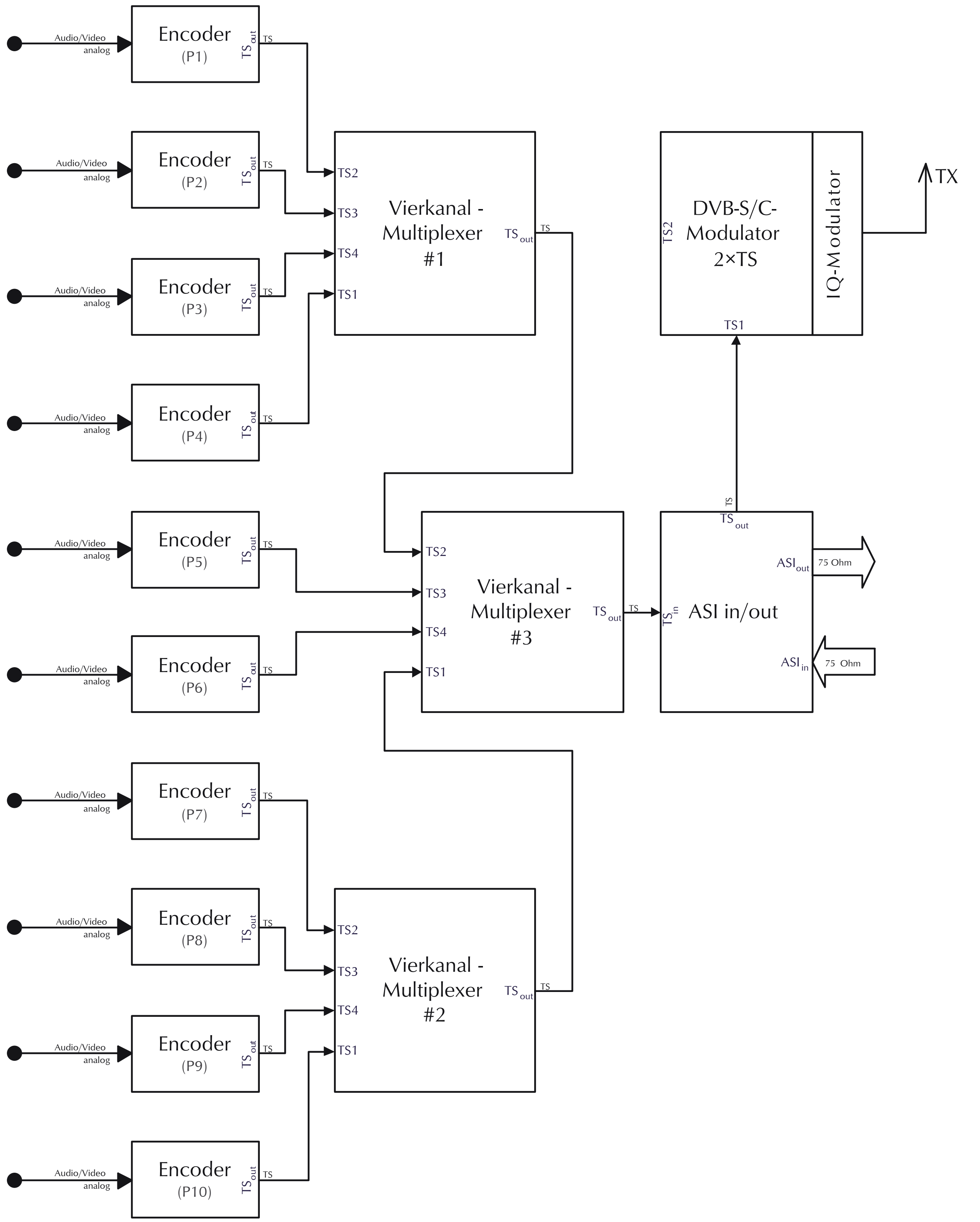
In this chapter is to be pointed out on the basis of some examples, how the different boards can be combined to different functional units.

2.1 Four analogue and four digital channels

using Mux4-Board, a DVB-S-Receiver and the 2ch-Baseband-Board.

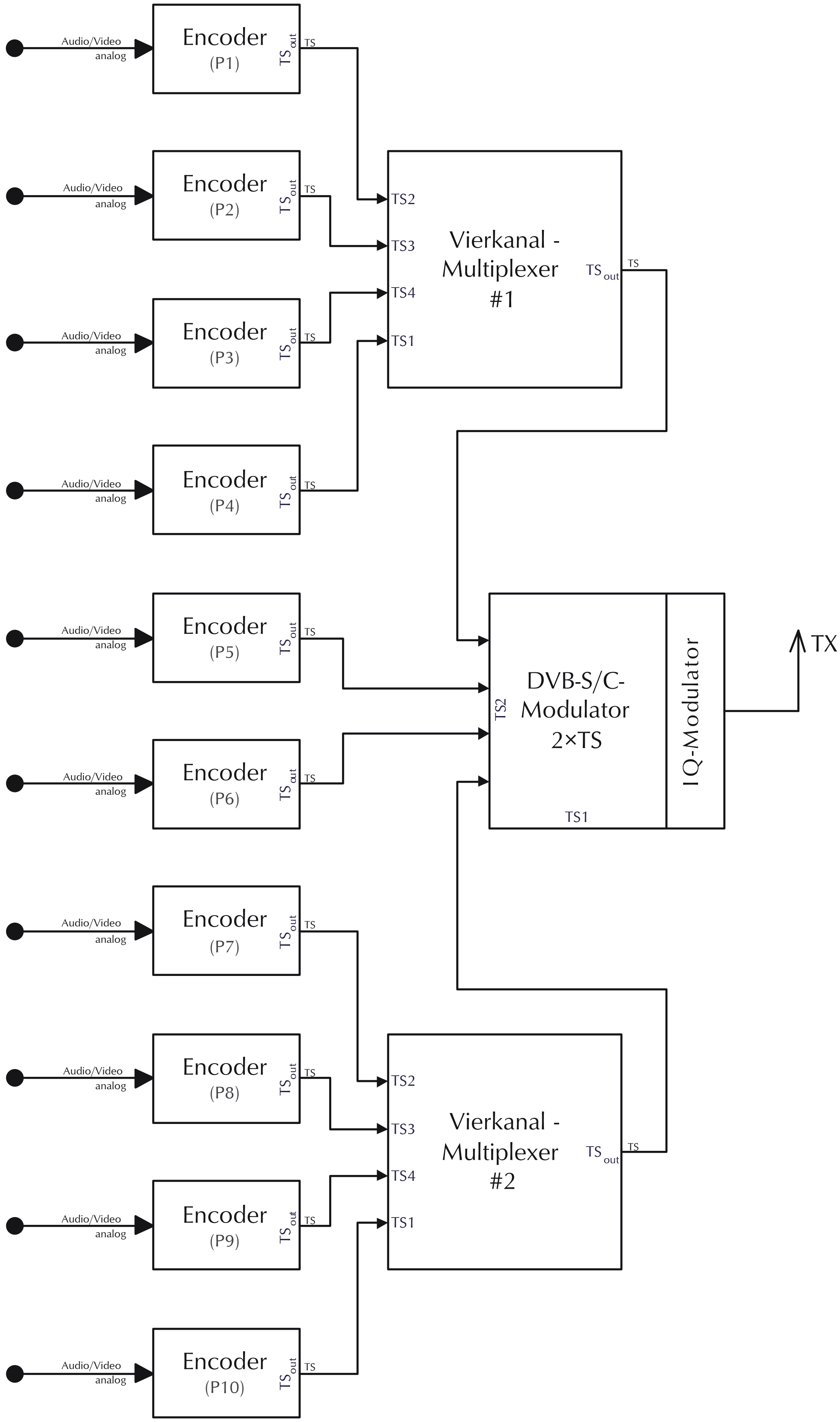


2.2 Ten analogue channels with ASI-In/Out



The Multiplexors #1 und #2 have to be set to „Drive Tx Clock = Off“, Multiplexor #3 to „Drive Tx Clock = On“. The Inputs TS1 and TS2 of Multiplexor #3 have to be configurated as „fujitsueval“.

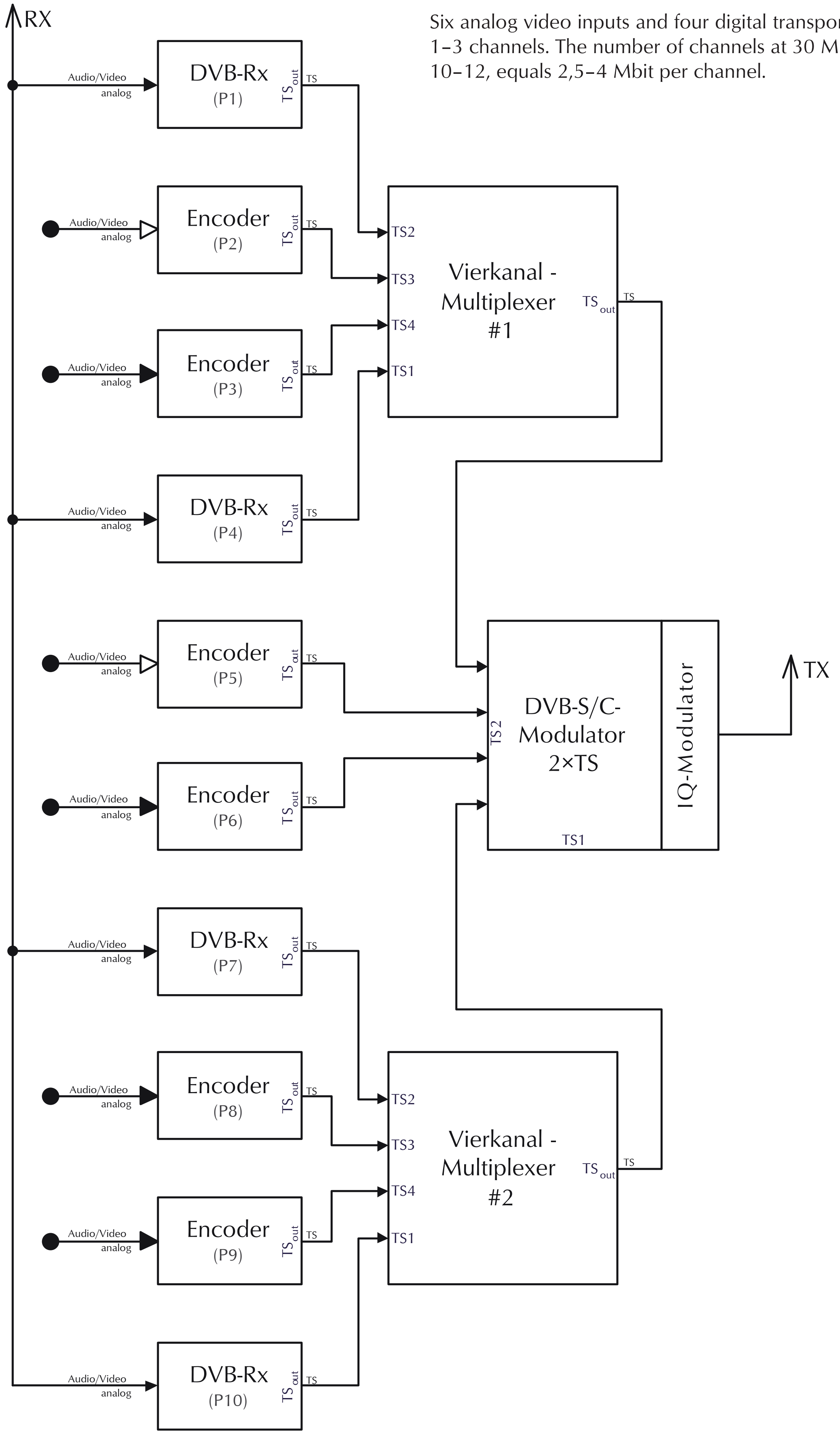
2.3 Ten analogue channels without encryption



The inputs TS1 and TS2 of the DVB-Modulator has to be set to „fujitsueval“.

2.4 Analogue video and digital transponder

Six analog video inputs and four digital transponders, every with 1–3 channels. The number of channels at 30 MSymbols is about 10–12, equals 2,5–4 Mbit per channel.



3 Start-up

Assembly according to chapter 2.

Connect your PC to the serial port of the DVB-S/C-Modulatorboard and start a terminal software. The parameters are 19.200 Baud, 8 data bits, no parity, 1 stop bit (19k2, 8N1). The terminal software should now listen at the serial port of the DVB-S/C-Modulatorboard.

Switch on the system and if the system is set up correctly the red LED will light followed by the yellow LED turning on. After this the red LED will blink stopping after some seconds. During the blinking red LED following messages appear on the terminal window:

terminal output	meaning
`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`	used Flash
`FLASH test passed`	
`FPGA configuration start (from FLASH)...`	
`FPGA configuration completed successfully`	
`PLL lock indication passed`	HF-Modul mounted? passed=yes
`Register test passed` `RAM test passed`	ext. SRAM testing
`Encoder SIO test passed`	testing Communication to Encoder
`01.01.02 00:00:00| D-ATV Transmitter startup, rev MCU SW V0.6(jnx) (baseband board v35c)`	DVB-S/C-Modulatorboard starts up
`01.01.02 00:00:00| Setting mode reg: 0x0083 / 0x0083 FEC mode 2/3 Inversion off  Freq 1255000kHz`	Settings (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 starts - from this point, the transmitter is working!
`01.01.02 00:00:02| Encoder 0 started (Encoder rev 0x00000250 SAA7113 rev 0x11)`	Encoder runs (red LED off)

Table 3a: The intial outputs of the DVB-Transmitter

4. Configuration

4.1 Introduction

This chapter explains the procedure for firmwareupgrades, configurations and the data format and semantic of the DVB configuration file.

The application "DATVfwtool" analyses this configuration file and downloads it to the Flash memory using the serial interface.

4.2 Terminology

TS, Transportstream	Data stream compliant to the DVB specification including split in TS packets. A TS can contain one or multiple broadcast programs that can include video, audio and optional data packets. Video and Audio information is distributed in packetized elementary streams (PES).
TS Packet	**T**ransport**s**tream Packet Stream portion with a fixed length of 188 bytes (4 bytes Header + 184 bytes payload), which contains the video-, audio and other datas. A PID shows the receiver, to which Data-Stream the information belongs.
PID	**P**acket **Id**entifier The Packet Identifier is a 13-Bit-value (dec. 0–8191), it identifies the stream, to which the TS-Packet belongs. Stuffing packets are signalized with the PID 8191 (hex 0x1fff). The PIDs 0x00 ... 0x1f are reserved for System tables.
FEC	**F**orward **E**rror **C**orrection The forward error correction adds redundant information (bits) into the stream, which are used to reconstruct the information after reception.
SI Tables	**S**ystem **I**nformation Tables Service Information tables are used to access to the programs and data in the TS. This information represent a kind of directory of the Transportstream.
NIT	**N**etwork **I**nformation **T**able. The NIT contains information about the network, like the network name, the frequency, error correction, modulation and network-related transportstreams and their parameters.
PAT	**P**rogram **A**ssociation **T**able Packets with PID 0, they contain informations which PIDs are used for the PMTs.
PMT	**P**rogram **M**ap **T**able Contains data about the content of the channel (video, audio, data), and which PIDs contains the related data-streams.

4.3 Upgrading the Microcontroller-Firmware

DATVfwtool can now be used for upgrading the firmware. To do this, the Board has to be switched into the Firmwareupgrade-Mode (see fig. 4.3a)

Then the board should be switched on and resetted. The command `DATVfwtool -d /dev/com1 -fm` proofs if the firmware can be upgraded, if yes the firmware is downloaded.

If this operation fails, please switch off and on the system and try again.

4.4 Using DATVfwtool

Port	Windowsname	Linuxname
COM1	/dev/com1	/dev/ttyS0
COM2	/dev/com2	/dev/ttyS1

Tab. 4.4a: Port names Windows/Linux

A typical call of DATVfwtool looks like this:

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

The `-d`-parameter sets the serial port (for portnames see fig. 4.4a).

The chapter 4.5 describe the format of the configuration file and the corresponding parameters. All parameters, which are not explained are only for test purposes or are not implemented yet or can be removed any time. Table 4.4b gives you a list with all valid parameters for DATVfwtool:

Parameter		Effect
normal syntax	alt.*	
`-check`	`-C`	checks the config file
`-write`	`-W`	Writes into the external flash. The software writes only sectors which have been changed.
`-force-write`	`-F`	Writes into external flash, even is nothing was changed
`-config-unchecked`		loads the given config file and continues even if errors occur
`-c`		loads the given config file and stops if an error occurs
`-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

Table 4.4b: Parameters of DATVfwtool — **alt. - alternative syntax*

4.4.1 Failed flash programming

Sometimes errors occur while writing the configuration into the flash. In this case, the external flash has to erased, before a new programming can be started.

To do this with the DVB-Mod 4×TS, the following steps must be done: set the jumper to pin 14 and 16 of U10, turn on the power, remove the jumper. Because of this procedure the firmware starts in a special mode (symbolized by the „R>“ as prompt). After pressing the space bar the following should appear on the screen:

Output

```
MCU SW V0.9a (jnx) (baseband board v35c)
Bootstrap options: PLL_EN 1 PLL_DATA 1 PLL_CLK 1 XRESET0 1 PLLTHR0 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
```

Fig. 4.4.1a: the initial output of the DVB-Transmitter, if the software in the flash is invalid

From this menu the Flash can be erased. Just press „f“ to call the Flash-Menu:

Output

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

Fig. 4.4.1b: The Flash-Menu

To erase the flash, you have to press „e“, followed by the sector to be erased. The sectors are numberd from 0–7, so the first input is „e0“. After that, the confirmation and the Flash-Menu again shoud appear:

Output

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

Fig. 4.4.1c: the Flash-Menu, confirmation of erasing in the first line

Now you just have to erase the over sectors, so type „e1“, wait for confirmation, type „e2“, wait for confirmation, and so on, until „e7“. The external Flash is erased now completely.

4.5 The configuration file

4.5.1 Structure

Fig. 4.5.1a shows the structure of the configuration file (for a 4×TS – for a 2×TS the groups `transportstream 1` and `transportstream 2` are not needed). Lines beginning with a `#` are comments and will not be interpreted by DATVfwtool. The `board`-part groups the parameters of the DVB-S/C-Modulatorboard. The `modulator`-part groups the parameters of modulation. The `transportstream`-parameter summarize a group of parameters, which are related to the TS input specified by the given number. The `teletext`-part at least contains the parameters for the teletext decoder and the still picture parameters.

```
# Sample D-ATV configura

board {
};

modulator {
};

transportstream 1 {
};

transportstream 2 {
};

transportstream 3 {
};

transportstream 4 {
};

teletext {
};
```

Fig. 4.5.1a: the structure of a configurat

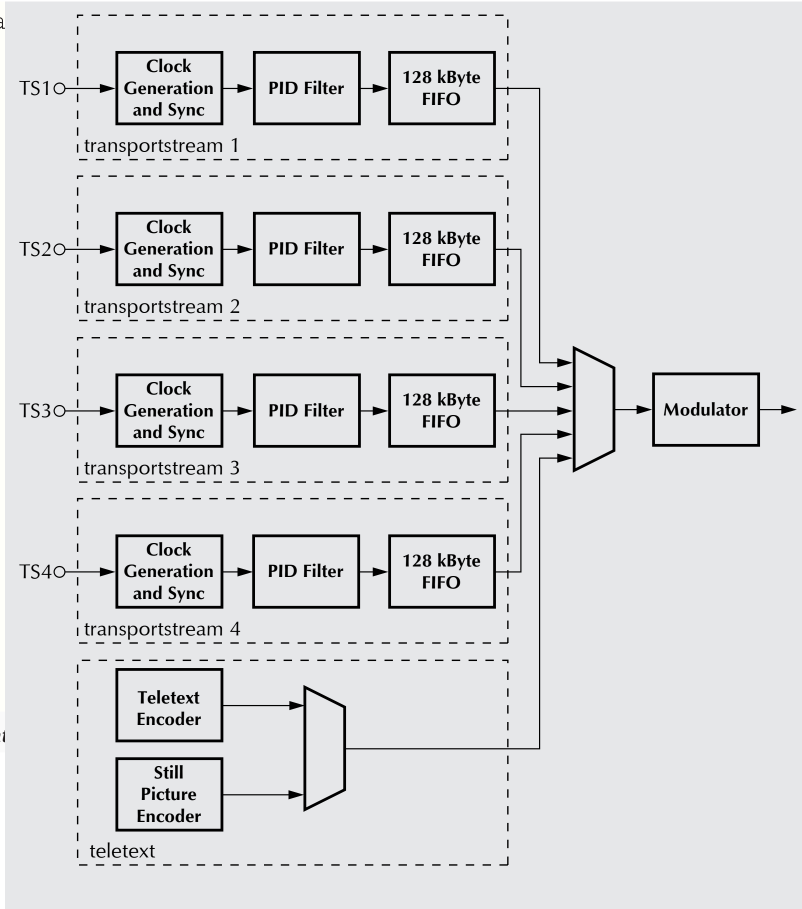


Abb. 4.5.1b: signal way in the DVB-Modulator 4×TS

The parameters are specified and using the syntax `Parameter = value;`, values can be decimal or hexadecimal integer preceded by „**0x**". A **k**- or a **M**-suffix multiply the value by 1,000 or 1,000,000 respectively. Text values must be given in quotation marks.

Fig. 4.5.1b show the signal way through the DVB-Modulator. This board connects the TS sources (e.g. MPEG2-Encoder or DVB-S-Receiver) over the TS I/Fs with the IQ Modulator.

The TS I/Fs are parallel I/Fs with 8 data lines, one clock and more optional synchronization signals. TS1 and TS2 are equipped with a FIFO, therefore the TS clock signal (CK) can be configured. CK can be provided by the DVB Board or generated from the data structure.

TS3 and TS4 don't have any FIFO. For this reason the DVB Multiplexer Board must provide the TS clock in order to avoid data loss. These both ports should be used mainly with the MPEG Encoder boards.

4.5.2 Board description

4.5.2.1 Clock

The parameter `clock` defines the frequency of the XTAL on the DVB Mux-/Mod-Board and is used to define all timings. The DVB boards are equipped with 60 MHz (default), but a reduced clock can be used for very low bit rates. The maximum limit is 62 MHz.

Example:

```
clock = 60M;
```

4.5.2.2 Board type

The parameter `board` defines the board type. Valid values are `admod` for a DVB-S/C-Modulator 2×TS, `ham` for the DVB-S/C-Modulator 4×TS and `mux4` for the 4ch-Multiplexor.

Example:
```
board = ham;
```

4.5.2.3 Generate tables

Normally, the D-ATV baseband board generates all DVB tables a receiver requires to locate programs within a transport stream. The transmission of these tables may be turned off. This is useful if a multiplexor, for example, already has generated the necessary tables.

Example:
```
generate tables = on;
```

4.5.3 Modulator

This chapter describes the configuration for the DVB-Modulators 4×TS und 2×TS.

4.5.3.1 Modulation

Select the modulation standard to be used. Possible values are `dvb-s` and `dvb-c`.

Example:
```
modulation = dvb-s;
```

4.5.3.2 Constellation

Select the constellation. For DVB-S only `qpsk` is possible. For DVB-C three options are possible: `qam16`, `qam32` or `qam64`.

Example:
```
constellation = qpsk;
```

4.5.3.3 FEC Rate, DVB-S

With this parameter the “inner Forward Error Correction code” rate can be selected. This parameter allows defining the adaptation ratio between bit rate and robustness of the modulated signal. Five options are possible: `1/2`, `2/3`, `3/4`, `5/6`, `7/8`. This parameter can only be used for DVB-S.

Example:
```
fec = 5/6;
```

$$BW \approx \frac{4}{3} SR \qquad \text{(Eq. 1)}$$

$$BR = 2 \cdot SR \cdot R_{inner} \cdot R_{outer} \qquad \text{(Eq. 2)}$$

SR	Symbol Rate (Symbols/s) (see 4.5.3.5)
BW	Signal Bandwith (Hz)
BR	User Bitrate (Bits/s)
R_{inner}	Inner FEC Rate (see 4.5.3.3)
R_{outer}	Outer FEC Rate, fixed at 188/204
F_{clk}	Crystal Oscillator Frequency (see 4.5.2.1)

$$BW \approx 1{,}15 SR \qquad \text{(Eq. 3)}$$

$$BR = \log 2q \cdot SR \cdot R_{outer} \qquad \text{(Eq. 4)}$$

SR	Symbol Rate (Symbols/s) (see 4.5.3.6)
BW	Signal Bandwith (Hz)
BR	User Bitrate (Bits/s)
Q	Constellation QAMq
R_{outer}	Outer FEC Rate, fixed at 188/204
F_{clk}	Crystal Oscillator Frequency (see 4.5.2.1)

4.5.3.4 Transmission frequency

The `frequency`-parameter specifies the frequency for transmission. The value must be in the Amateur band of 70, 23, or 13 cm and the HF module must support the selected band.

Example:
```
frequency = 1275M;
```

4.5.3.5 Symbolrate, DVB-S

This parameter specifies the bandwidth for the modulated signal (see Eq. 1) and the available bit rate for the user (see Eq. 2).

One of the following values must be chosen for the ratio F_{clk}/SR: 2, 2⅓, 2½, 2⅔, 3, 3⅓, 3½, 3⅔, 4, 4⅓, 4½, 4⅔, 5, 5⅓, 5½, 5⅔, 6, 6⅓, 6½, 6⅔, 7, 7⅓, 7½, 7⅔, 8, 8⅓, 8½, 8⅔, 9, 9⅓, 9½, 9⅔, 10, 11, 12, 13, 14, 15 or 16. The application DATVfwtool rounds the chosen ratio to the next available one.

Example:
```
symbol rate = 15000k;
```

4.5.3.6 Symbolrate, DVB-C

This parameter specifies the bandwidth for the modulated signal (see Eq. 3) and the available bit rate for the user (see Eq. 4).

One of the following values must be chosen for the ratio F_{clk}/SR: 8, 9, 10, 11, 12, 13, 14, 15 or 16. The application DATVfwtool rounds the chosen ratio to the next available one. It is recommended, to use a bandwith of 8 MHz or less, and a symbolrate of 6.9 MSymbols/s or less, because the receivers are limited to 8 MHz. To get 6.9M, a 55.2 MHz-Oscillator is needed!

Example:
```
symbol rate = 6000k;
```

4.5.3.7 Inversion

This parameter defines if signals I&Q shall be swapped. The effect of this swapping is same as the reception of USB signals with a LSB receptor. Most of the receptors detect automatically if swapped signals are used. The setting `off` for this parameter allows bypassing the DVB Signal. The setting `on` shall be used with a spectrum inverting transverter.

Example:
```
inversion = off;
```

> It seems, that the WinTV DVB-S Nova Card with convergence.de-Firmware doesn‘t detect Inversion-Mode automatically!

4.5.3.8 PTT (4×TS only)

This parameter allows switching on/off the sender by power on/off. After starting it is possible to switch the PTT using the Menu.In case of DVB relays the typical setting is `on`, and accordingly `off` for end stations.

Example:
```
ptt = off;
```

4.5.3.9 Network Name

This parameter must be set to the sender identifier.

Example:
```
network name = "DemoTV";
```

4.5.3.10 Transportstream clock direction

This parameter is valid for the 4TS-Mux-Board only. Typically, it is set to `off` (for using the Mux-Board with the modulator-setting `fujitsueval`). For using with the ASi-Out-Boards this parameter **must** be set to `on`.

Valid parameters are `off` and `on`.

Example:
```
drive ts clock = off;
```

4.5.4 Transportstream Interface

4.5.4.1 Input mode selection

This parameter specifies the TS port mode. The following modes are available:

`off`	Port disabled
`datvencoder`	D-ATV MPEG2-Encoder connected to the port
`fujitsueval`	Fujitsu MPEG2-Encoder Evaluation Board or Multiplexor connected to the port
`extclock`	Any TS source (incl. TS clock)

The „Fujitsu MPEG2-Encoder Evaluation Board" is similar to the D-ATV Encoderboard. The difference is, that the DVB-S/C-Modulatorboard doesn't start the MPEG2-Firmware-Download into the encoder. The `extclock`-Option can only be used at TS1 and TS2 connectors.

Example:
```
mode = datvencoder;
```

☞ Attention: the VPIDs are given by default. Thereby is TS1=0x20, TS2=0x30, TS3=0x40 and TS4=0x50.

4.5.4.2 Clock edge selection

This parameter only takes effect, if the TS input mode is selected as `extclock`. It specifies the active edge of the TS Clock (CK). Valid values are `falling`, `rising` and `both`.

Example:
```
clock edge = rising;
```

$$1 \leq N \leq 4$$

$$F_{TSCK} \leq \frac{F_{clk}}{2N}$$

F_{TSCK}	Transportstream clock frequency
N	Clock Debounce Filter parameter (see 4.5.4.3)
Fclk	Crystal oscillator frequency (see 4.5.2.1)

Fig. 4.5.4.3a

4.5.4.3 Clock Debounce Filter

This parameter defines the behavior of the Clock Debounce Filter. It must be set to the highest value N, that fulfills the condition shown in Fig. 4.5.4.3a.

Example:
```
clock filter = 4;
```

4.5.4.4 Bitrate

Set the system bit rate for the output TS.

Example:
```
bitrate = 4500k;
```

4.5.4.5 Selection of video input

This parameter only takes effect in `datvencoder`-mode. It specifies the characteristic of the video-input signal. More options can be selected separated by commas. The valid values are given in the table below.

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

Resolution	
`d1`	D1-Resolution (752×576 px.)
`hd1`	HD1-Resolution (384×576 px.)
`sif`	SIF-Resolution (384×288 px.)
`qsif`	QSIF-Resolution (192×144 px.)

Signal	
`pal`	PAL
`pal-m`	PAL-M
`secam`	SECAM
`ntsc`	NTSC
`ntsc443`	NTSC with 4.43 MHz color subcarrier

Input	
`composite`	use composite-input
`svideo`	use S-Video-input
`ycbcr`	use YCbCr-component-input
`sdi`	use SDI-input

4.5.4.6 Video GOP-Configuration

This parameter specifies the picture enconding sequence of the encoder. The preselected mode gives you a good encoding efficiency at the price of a higher encoding latency (time lag between input picture and encoded output picture). The latency can be reduced by decreasing the GOP-size. This parameter should be changed only by experts, who **really** understand the MPEG2-encoding.

Example:
```
video gop = “IBBPBBPBBPBBPBB“;
```

The minimum GOP size is 1, the maximum size is 30. The size isn‘t indicated explicitly, it arises as a result of the indication of the I-, P- and B-Frames.

4.5.4.7 Spatialfilter

This parameter specifies the cut-off-frequency of the spatialfilter and the sharpness. Valid values are `standard`, `soft`, and `sharp`.

Example:
```
spatial filter = "standard";
```

4.5.4.8 Audio Encoder Bitrate

This parameter takes only effect in `datvencoder`-mode. It specifies the bitrate of the MPEG2 Layer2 Audioencoder. Valid Bitrates are shown in Fig. 4.5.4.8a, but the allowed bitrates are depending on the selected Audio encoding mode (see chapter 4.5.4.9)

Example:
```
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: Valid audio encoding bitrates and -modes

4.5.4.9 Audio encoding mode

This parameter is only for the `datvencoder`-mode. It specifies the audio mode for encoding and allows the following values:

`stereo`	Usual stereo audio
`joint stereo`	Intensity stereo or mid-/side stereo
`dual channel`	Two independent channels (e.g. two languages)
`single channel`	Mono audio signal

Example:
```
audio mode = joint stereo;
```

4.5.4.10 Audio sampling rate

This parameter takes effect in `datvencoder`-mode only. It specifies the samplerate of the MPEG2 Layer2 Audio-encoder. Valid values are `48000`, `44100` und `32000`.

Example:
```
audio sample rate = 44100;
```

4.5.4.11 Program Clock Reference (PCR) PID

Set the PID for the Program Clock Reference (PCR). Normally this PID has same value as the correspond-ing Video PID.

Example:
```
pcr pid = 0x20;
```

4.5.4.12 Video PID

Sets the PID for the video-stream.

Example:
```
video pid = 0x20;
```

4.5.4.13 Audio PID

Sets the PID for the audio-stream. This parameter must differ from PCR and Video PID.

Example:
```
audio pid = 0x21;
```

4.5.4.14 Program Map Table (PMT) PID

Set the PID for the Program Map Table (PMT). The PMT provides the mappings between program numbers and the program elements that comprise them; i.e. the PMT is the complete collection of all program definitions for TS.

Example:
```
pmt pid = 0x22;
```

4.5.4.15 Program callsign

Set the callsign for the TV channel. This identifier is inserted in the SI Table, which allows identifying the TV channel at reception.

Example:
```
callsign = “Program1“;
```

4.5.4.16 TV Channel Language

This parameter identifies the language of the TV channel. Use `“eng”` for English and `“DEU”` for German.

Example:
```
language = “eng”;
```

4.5.4.17 PID filter

The purpose of this parameter is to select which PIDs from the selected TS input shall be passed or blocked. Two strategies are possible:

1. All PIDs are passed excluding a list of PIDs
2. All PIDs are blocked excluding a list of PIDs, which are released

Pre-setting are `all` or `none` with a list of the exceptions using the keywords `minus pid/mask` and `plus pid/mask`. The value `pid` specifies the corresponding numeral and `mask` specifies the bit pattern for comparison. The resulting PID Filter has to block the PID 8192 (0x1fff), which corresponds to stuffings. The maximum number of filters is four. If more than four filters are given, only the first four will be used, all following will be ignored.

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

4.5.4.18 Tuner Mode Selection

This parameter specifies if a receptor is connected to the port. Following modes are possible:

`off`	No tuner connected
`dfm`	DFM-Analogue-Tuner connected to the SR-Systems MPEG-Encoder
`mb86a15`	Fujitsu DVB-S-Tuner connected to the TS-Port
`nxt6000`	DVB-T-Tuner connected
`sv297`	DVB-C-Tuner connected

Beispiel:
```
tuner mode = off;
```

4.5.4.19 Tuner frequency

This parameter is activated only if the Tuner Mode is not set to `off` (see chapter 4.5.4.18). It defines the frequency to which the Tuner has to be tuned.

Example:
```
tuner frequency = 1260M;
```

4.5.4.20 Tuner FEC Modus

This parameter can be used only with Tuner Mode `mb86a15` (see chapter 4.5.4.18) gesetzt ist. It specifies which internal FEC settings shall be tested until a valid signal is found. It can be set to `auto` or to an internal FEC rate ratio.

Example:
```
tuner FEC = auto;
```

4.5.4.21 Tuner Symbolrate

This parameter can be used only with Tuner Mode `mb86a15` (see chapter 4.5.4.18) gesetzt ist. It specifies which symbol rate can be expected.

Example:
```
tuner symrate = 3000k;
```

4.5.5 Teletext

4.5.5.1 Program Clock Reference (PCR) PID

The PID for the PCR packet can be set with this parameter. Usually this PID is same as the corresponding Video PID.

Example:
```
pcr pid = 0x20;
```

4.5.5.2 Video PID

This parameter sets the Video PID for the video stream that contains a still picture.

Example:
```
video pid = 0x20;
```

4.5.5.3 Teletext PID

This parameter sets the PID for the teletext packets.

Example:
```
teletext pid = 0x21;
```

4.5.5.4 Program Map Table (PMT) PID

This parameter sets the PMT PID. The PMT contains the information, which PID belongs to which TV channel.

Example:
```
pmt pid = 0x22;
```

4.5.5.5 Program Callsign

This parameter sets the callsign of the channel with Teletext/still picture. This information is inserted in the SI Table and allows to the receptor to identify the channels.

Example:
```
callsign = "DVB Test";
```

4.5.5.6 Language Setting

This parameter is used to identify the language for the channel with Teletext/Still picture. Two options are available: `eng` for English and `DEU` for German.

Example:
```
language = "eng";
```

4.5.5.7 Still picture files

This parameter specifies the file with the still picture that shall be transmitted. This channel may be used e.g. for the operator's logo.

☞ Be aware that this channel is not DVB compliant, thus it may happen that some receptors will not decode and show it.

The file shall be of type JPEG or alternatively MPEG-2 elementary stream. MPEG-2 software encoders generate usually non-compatible streams. If the file contains a JPEG picture with the size 704×576 pixels, then it is possible to use the application mpeg2enc from mjptools, which generates a compatible stream. The binary file must be located in same directory as fwtools under Windows or in the directory usr/bin under Linux.

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

To convert a picture into a mpg-file a converter like „Omniformat“ (available at www.omniformat.com) can be used.

4.5.5.8 VM Code

This parameter specifies the file, which contains the teletext encoder virtual bytecode. See chapter 4.8 for more details.

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

4.5.6 Programs from external sources

The DVB sender is able also to send programs from alternative sources like PCs or DVB-S receivers. In order to make possible for the STBs to find these programs, the DVB sender must actualize the PMT and PAT with the new PIDs.

4.5.6.1 Program Clock Reference (PCR) PID

The PID for the PCR packet can be set with this parameter. Usually this PID is same as the corresponding Video PID.

Example:
```
pcr pid = 0x420;
```

4.5.6.2 Program Map Table (PMT) PID

This parameter sets the PMT PID. The PMT contains the information, which PID belongs to which TV channel.

Example:
```
pmt pid = 0x422;
```

4.5.6.3 Language

This parameter specifies the language of the tv-channel. It should be set to `eng` for English or `DEU` for German.

Example:
```
language = "eng";
```

4.5.6.4 Stream subsections

The streams subsections (e.g. stream divisions: (`video stream`, `audio stream`, `teletext stream` und `stream`) correspond to the individual streams, to which a program consists in.

4.5.6.5 PID

This parameter defines the Stream PID.

Example:
```
pid = 0x440;
```

4.5.6.6 Stream Type

This parameter defines the stream type (see [1, tab. 2-36]). This parameter is set automatically for Video-, Audio- and Teletext-Streams.

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

4.5.6.7 Stream ID

This parameter sets the ID for the stream.

Example:
```
stream id = 1;
```

4.5.6.8 Component Type

It sets the component type of the stream (see [2, tab. 24]).

Example:
```
component type = 1;
```

4.5.6.9 Language

This parameter sets the language of the stream. Two options are available: `eng` for English or `DEU` for German.

Example:
```
language = "eng";
```

4.6 The old Teletext-Encoder

The old Teletext-Encoder is a static table with the teletext-pages and its lines. There are just a few dynamic contents and no control about the encoding. Probably it will be removed from the configuration in the future. Fig. 4.6a shows the configuration commands of an exemplary teletext-page.

```
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.";
   };
};
```

Fig. 4.6a: configuration commands for the old Konfigurationsdatei für den alten Teletext-Encoder

4.6.1 Teletext section

4.6.1.1 Teletext page header

This parameter sets the contents of the topmost teletext line that is displayed right of the pagenumber.

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

4.6.2 Teletext page section

The `teletext`-section may contain `page { };` -subsections, each specifying a single teletext page.

4.6.2.1 page number

This parameter specifies the teletext page number. Its value must be between `100` and `899` inclusive. Teletext decoders start with page 100, so a page 100 should be present and display introductory material.

Example:
```
page number = 100;
```

4.6.2.2 Teletext lines

This parameter specify the teletext page lines. The lines are numbered from 1 to 24 inclusive. Teletext line decoders can be up to 40 characters long. Shorter lines are padded with space characters. Nonprinting characters can be entered by a backslash `\`, followed by an `x`, and a two digit hexadecimal number that encodes the character code. For example, to enter character 1 (0x01) type `\x01`. Character codes 0–31(0x00–0x1f) are used for ETSI Teletext Attribute markup (eg. colours), and character codes 128–255 (0x80–0xff) are used to insert dynamic data, such as packet counters.

Example:

```
line 2 = “\x01 www.D-ATV.de“;
```

4.7 Sample configuration

Fig. 4.7a shows a simple minimalistic configuration file. It assumes that there is one MPEG2 encoder connected to TS1, and that TS2–TS4 are left unconnected.

```
# 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 = d1 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";
};
```

Fig. 4.7a: Example of a simple, minimalistic configuration file

4.8 The new Teletext-Encoder

The new Teletext-Encoder allows full control over the encoding process and arbitrary dynamic content. It is driven by a user bytecode program that is interpreted by a stack-based virtual machine.

Bytecode teletext programs need not to be written in the stack-based assembly language of the virtual machine (VM). They can be written in the C programming language and then compiled into the bytecode. The following table shows the executables that constitute the bytecode development system:

`cpp`	C Preprocessor
`rcc`	C Compiler proper
`vm`	VM Simulator
`vmar`	Bytecode Archiver
`vmas`	Bytecode Assembler
`vmdisass`	Bytecode Disassembler
`vmld`	Bytecode Linker
`atv2txtvm`	conversion utility from DG9MHZ ATV files to VM Teletext source code

Fig. 4.8a: Executables of the bytecode development system

Assuming the teletext encoder C code is contained in a file named `teletext.c`, the C code can be compiled and assembled in to an object file `teletext.o` using the following command:

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

The object code file can be disassembled with:

```
vmdisass teletext.o
```

The object code file can be simulated with:

```
vm -c -1 -m teletext teletext.o
```

Fig. 4.8b shows an example source code of a teletext encoder.

DG9MHZ ATV files can be converted to VM teletext object code using:

```
atv2txtvm -c -o teletext.o -i "D-ATV" -p 10 100_0000.ATV 101_0000.ATV
```

ATV Files can be written by vtedit from [2].

```
/* sample teletext encoder */
#include “dvbs.h”
static const char pg_header[] = TXT_ARG0 “ www.D-ATV.de “ TXT_ARG1;

static const char *pg_100[] = {
      pg_header,
      NULL,
      TXTATTR_ALPHA_RED “ www.D-ATV.de”,
      NULL,
      “Digital Baseband:”,
      “ Thomas Sailer, HB9JNX/AE4WA”,
      NULL,
      “RF”,
      “ Wolf-Henning Rech, DF9IC/N1EOW”,
      “ Jens Geisler, DL8SDL”,
      NULL,
      “Schematics, Boards &”,
      “ Connections to Fujitsu”,
      “ Stefan Reimann, DG8FAC”,
      NULL,
      TXTATTR_ALPHA_YELLOW “adacom e.V.”,
      NULL,
      NULL,
      NULL,
      NULL,
      NULL,
      NULL,
      NULL,
      NULL,
      NULL
};

void teletext(void)
{
      char t[9];
      for (;;) {
            timedec(t, NULL, gettime());
            teletext_encodepage(0, 24, 0x100, 0, 0, pg_100, “100”, t);
            teletext_encodepage(0, 0, 0x1ff, 0, 0, pg_100, “100”, t);
      }
}
```

Tabelle 4.8b: Example of a Teletext encoder source code

4.8.1 C-Code

The header file `dvbs.h` contains prototypes for the built-in library functions. The VM starts the teletext encoder by calling the function `teletext`, with the prototype `void teletext(void)`

4.8.2 VM Built-In Library Functions

4.8.2.1 C Type sizes

Type	Bits
char	8
short	16
int	32
long	32

4.8.2.2 C99 standard macros

`NULL`, `offsetof`

4.8.2.3 C99 standard types

`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 functions

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

4.8.2.5 Event log functions

`void logreadinit(unsigned int *p);`	
Rewinds the event log to the oldest log message still in the circular buffer.	
`p`	A pointer to an opaque cookie of type `unsigned int`

`unsigned int logreadline(unsigned int *p, char *buf, unsigned int bufsz);`	
Reads the next event log message	
`p`	A pointer to an opaque cookie of type `unsigned int`
`buf`	Pointer to a buffer
`bufsz`	Size of the buffer
`logreadline`	Returns the number of non-null characters, which are stored in the buffer. A return value of Null means the end of the Event-Log buffer.

4.8.2.6 Time and date functions

struct time	
`day`	Modified julian date (Number of days since 17.11.1858)
`sec`	Number of seconds since midnight
`msec`	Number of ms from current second
`valid`	if `valid` is set, time and date have been actualized via serial port or via DCF77

struct timehms	
h	Hours
m	Minutes
s	Seconds

struct date	
d	Day
m	Month
y	Year

```
struct time gettime(void);
```

returns the actual time

```
u_int32_t getjiffies(void);
```

Returns a monotone increasing value. It increases HZ times per second

```
struct date mjdtodate(u_int16_t mjd);
```

Converts a modified julian date into the standard gregorian date

```
u_int16_t datetomjd(u_int16_t d, u_int16_t m, u_int16_t y);
```

Converts a standard gregorian date into the modified julian date

```
char *timedec(char *buf, struct timehms *hms, u_int32_t tm);
```

Converts the number of seconds since midnight into hours, minutes and seconds using a readable format 01:23:45hms and buf shall be NULL. This function returns a pointer referencing the buf[0].

4.8.2.7 Parameter and statistic funktions

```
u_int16_t getadc(unsigned int n);
```

Returns the value from the n-th AD Converter. The range for n = 0 ... 3, and the range for the 10 bit AD conversion value is between 0 ... 1023, which corre-sponds to an input voltage of 5 Volt.

```
u_int32_t readcounter(unsigned int n);
```

Returns the value of the n-th counter.

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

returns the „spectral inversion“-setting

n	*Value*
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);
```

Returns the FEC-Mode.

Return value	*FEC-Mode*
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);
```

Returns the transmission center frequency in kHz.

```
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);
```

Converts `val` into a hexadecimal string, which has been stored into the buffer `buf`. The length of the stored string is `len`, and `buf` must be defined with the character length of `len+1`. This function returns a pointer to the string stored in `buf` but not necessarily to the address of the first character.

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

Converts `val` into a hexadecimal string, which has been stored in the Buffer `buf`. The length of the stored string is `len`, and `buf` must be defined with the character length of `len+1`. This function returns a pointer to the string stored in `buf` but not necessarily to the address of the first character.

4.8.2.9 TS1/TS2 table decoder

The TS1/TS2 table decoder tries to extract data from the System Information tables received on transport stream ports 1 and 2.

struct portcapture	
`event_id`	Increments, when an update for the Service Descriptor Information is received
`transport_stream_id`	Transport Stream ID
`nit_pid`	PID of the transmitted Network Information Table
`service_id`	Service ID
`network_id`	Network ID
`service_provider_name`	Service Provider Name
`service_name`	Service Name
For more details to the DVB System Information (SI) tables, see [2].	

```
struct portcapture(unsigned int port);
```

Returns SI table data for transport stream port

Port	***Transport Stream***
`0`	TS1
`1`	TS2

4.8.2.10 Highlevel teletext encoding functions

```
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, ...);
```

Encodes multiple teletext lines, from `startline` up to `endline`	
`startline`	Usually 0
`endline`	Usually 24
`pagnr`	Specifies the number of pages and shall be in the range between 0x100 and 0x8ff. Page numbers that contain the hexadecimal values A-F can not be usually accessed.
`subnr`	Specifies the number of sub-pages (usually 0)
`flags`	Can be zero or multiple ored TXTPAGECTRL macros.
`TXTPAGECTRL`	See detailed description in [3, 9.3.13, p.27]
`lines`	Contains a pointer to an array with `endline-startline+1` strings. Every string specifies the content of one line. A `NULL`-Pointer disables the encoding process for the corresponding line. Teletextlines can contain also `TXTATTR`-Macros or `TXT_ARGn`-Arguments.
`TXTATTR`	Macros (see [3, 12.2, p. 76–80])
`TXT_ARGn`	References to optional arguments. A maximal number of 64 pointers can be used.

4.8.2.11 Lowlevel Teletext encoding functions

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

Encodes a data buffer starting at `src` with the length `len`, using unequal Teletext parity, and stores it in `buf`.

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

Encodes a data buffer starting at `src`, with `len` nibbles and stores with Teletext 8/4-Hamming-Code in `buf`. The coding order is low nibble of `src[0]`, followed by the high nibble of `src[0]`, followed by the low nibble of `src[1]`, etc.

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

Encodes a data buffer starting at `src`, with `len` triples and stores with Teletext-24/18-Hamming-Code in `buf`. `src[0]` contains the lowest 6 Bits, `src[1]` the middle 6 Bits and `src[2]` the high 6 Bits.

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

Returns the pointer to the buffer with the actual TXT line. The line buffer has the length of 42 bytes and contains a full TXT line excluding clock run-in and framing code [3, 7.1, p.17ff].

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

Sends the actual line and returns a pointer to the next line in the buffer.

4.9 Connection of a PC Parallel Port to a TS Input port

The parallel port of a PC can be used as a simple (but slow) input for a Transport Stream. The maximal transmission speed is limited to 2 Mbps. Table 4.9a shows how to connect a PC parallel port to TS1 oder TS2. The input port must be configured to extclock mode and theTT clock filter shall be set to maximal 4.

Pin	*Parport-Signal*	*TS-Signal*
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: Corresponding signals for connection

5 Annex

5.1 Credits

We want to thank Wacker's Kaffee and Coca Cola for their caffeinated products; Pasquale for Pizza and Steaks, Naçi for Sis Kebap, also we wish to thank the Bären-Treff for Sweets and Pepperworld for hot stuff.

5.2 References

[1] ETSI EN 300 328 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 } , 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 Authors

Software engineering	Thomas Sailer, **HB9JNX**
Circuit and board design	Stefan Reimann, **DG8FAC**
HF-Design	Prof. Dr. Ing. Wolf-Henning Rech, **DF9IC** Ing. Jens Geisler **DL8SDL**