

## **TOPFIELD INTERNAL INFORMATION**

**(2008-12-23)**

### **HISTORY**

- 2008-12-23 R2 provided the additional fields from the TF5700PVRt [transponder block](#)  
Updated the [Topy / SysID](#) list
- 2008-08-09 Added a note from R2-D2 about the [REC header](#) on the TF5810/TF500/TF600
- 2008-07-30 Added a note about a [timer structure](#) bug (thx to ibbi)
- 2008-05-08 Updated the TF5800 [transponder struct](#) and the [flash header](#) (another one from the patch and information factory R2-D2 ☺ )
- 2008-04-29 Added the 3 DOW bits to the [frontpanel's](#) new date format (thx to R2-D2)
- 2008-04-13 Updated [EEPROM](#) ListInfo+0x04 and EtcInfo+0x0d/0x0e (thx to EMJB & R2-D2)
- 2008-02-17 Added some details to the [FP Comm](#) table
- 2008-01-17 Got some more [EEPROM](#) details from EMJB  
Added a relative address column to the [EEPROM info](#), which is compatible with all firmware releases
- 2008-01-13 Updates several [FP Comm](#) commands  
Updated the [SysID Table](#)
- 2007-12-30 [SetRemoteFilter](#): Added code for Mode 2  
Fixed a false offset value in the service name offsets of the [5800](#) and [5200](#) (thx to EMJB)
- 2007-12-13: [12/24h-Flag in the EEPROM etc-Block](#) (thx to R2-D2)
- 2007-09-11: Added the [WinFileAttribute flags](#) in the directory slot description (thx to Aldarin)  
Corrected some details of the [root directory description](#) (thx to ibbi)
- 2007-07-24: Added a note about the TF5700PVRt [transponder block size](#)
- 2007-06-03: More details about the [header of the compressed loader and firmware](#)
- 2007-05-10: Added some details to the [FP communication table](#) (thx to DeadBeef)  
Added the S3\_CRC description in the [DataFiles Directory section](#)
- 2007-04-28: Added the Network ID to the [Sat Timer Block](#) (thx to R2-D2)
- 2007-01-27: [Meaning of unknown\[3\] in the ServiceInfo block of the REC header](#)  
[How to auto detect a REC header type](#) (thx to jkIT)
- 2007-01-14: [MHEG-Flag in the EEPROM etc-Block](#) (thx to SimonC)

## CONTENTS OF THE FLASH MEMORY

The decompressed flash memory is divided into several blocks. Every block starts with a CRC-16 ([see below for the algo](#)) followed by a 0xFFFF. The CRC calculation begins after the 0xFFFF (block start address + 4).

TF5000/5500/5000 Masterpiece

TF5800 PVRt

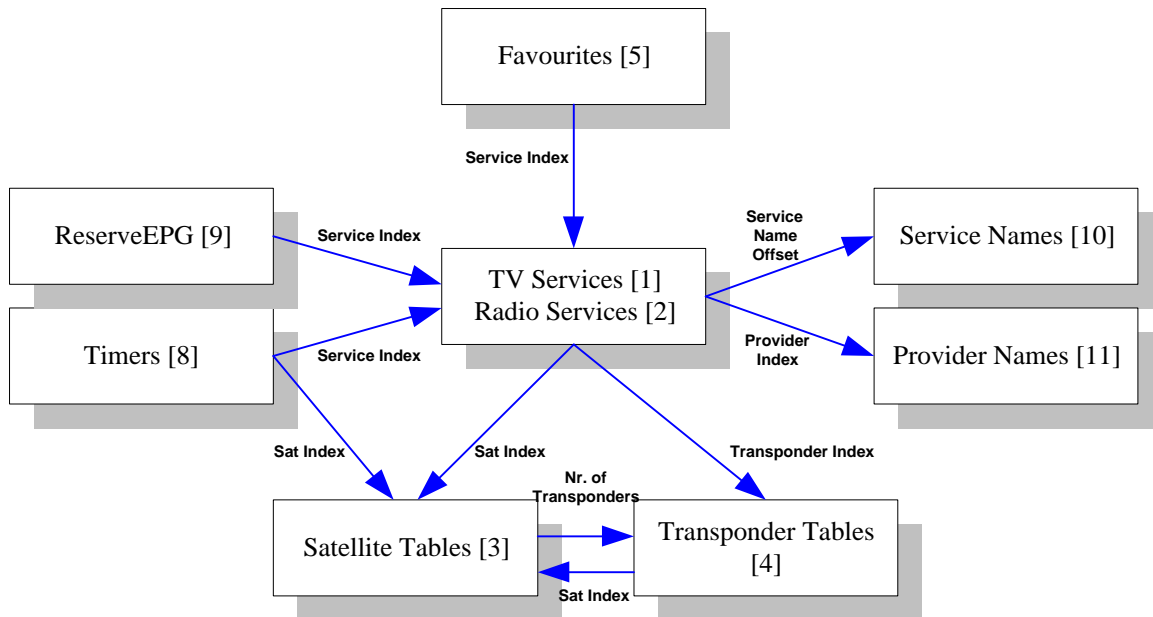
TF5200 PVRc

Block	Remarks	Start	End	Length	Rec Size	Records	cross links to block
1	TV Service Data	00000000 00000000 00000000	0001B583 00007D03 00007D03	112004 32004 32004	32 32 32	3500 1000 1000	3, 4, 10, 11
2	Radio Service Data	0001B584 00007D04 00007D04	00027107 0000FA07 0000FA07	48004 32004 32004	32 32 32	1500 1000 1000	3, 4, 10, 11
3	Satellite Tables	00027108 0000FA08 0000FA08	0002bcbf 0000FA2B 0000FA2B	19384 36 36	76 - -	255 - -	4
4	Transponder Data	0002BCC0 0000FA2C 0000FA2C	00037847 00010A23 00010627	48008 4088* 3068	16 16 12	3000 255 255	3
5	Favorites	00037848 00010A24 00010628	00039D17 00012EF3 00012AF7	9424 9424 9424	314 314 314	30 30 30	1, 2
6	Game Data	00039D18 00012EF4 00012AF8	00039D5F 00012F3B 00012B3F	72 72 72	- - -	- - -	
7	Clock Setup	00039D60 00012F3C 00012B40	00039D6B 00012F47 00012B4B	12 12 12	- - -	- - -	(4)
8	Timer	00039D6C 00012F48 00012B4C	0003C29F 0001547B 00014F67	9524 9524* 9244	136 136 132	70 70 70	1, 2, 3, 4
9	Reserve EPG	0003C2A0 0001547C 00014F68	0003C4D3 000156AF 0001519B	564 564 564	8 8 8	70 70 70	1, 2
10	Service Names	0003C4D4 000156B0 0001519C	0004C2BF 0002549B 00024F87	65004 65004 65004	- - -	- - -	(1), (2)
11	Provider Names	0004C2C0 0002549C 00024F88	0004D7C3 0002699F 0002648B	5380 5380 5380	21 21 21	256 256 256	(1), (2)
12	OTA	0004D7C4 - -	0004D7E7 - -	36 - -	- - -	- - -	
13	Unused	0004D7E8 000269A0 0002648C	0007FFFF 0007FFFF 0007FFFF	206872 366176 367476	- - -	- - -	
14	WLAN	0004DBEC - -	0004DFEB - -	1024 - -	- - -	- - -	
15	Unused	0004DFEC - -	0007FFFF - -	204820 - -	- - -	- - -	

Green background: all models use the same record structure.

\* The transponder block on a TF5700PVRt (HDMI) is 8 bytes longer than on other machines

How the tables are cross linked:



## TF5000/5500/5000 MASTERPIECE TV SERVICE DATA [1]

A single record contains 32 bytes. The allocated space can hold up to 3500 services. Unused records are marked by a Tuner value of 00 (this is used to detect the last entry).

```
01 00 0D 74 32 C9 03 E9 00 A0 00 A0 80 A1 00 00 00 00 4E 09 00 2D 00 00 00 00 00 00 00 00 00 00
01 00 0D 50 32 D6 03 EC 00 AA 00 AA 80 AB 80 00 00 00 4D 30 00 06 00 00 00 00 00 00 00 00 00 00
```

Offset	Length	Field	Remarks
0x0000	1	Sat Index	
0x0001	1	Unused	
0x0002	2	Transponder Index & Flags	0000 1101 01.. .... Transponder Index with the same Sat Index (Block [4]) .... .... ..11 .... Tuner (11= both Tuner, 10 =Tuner 2, 01 = Tuner 1) .... .... .... 0... Delete .... .... .... .1.. CAS .... .... .... ..0. Lock .... .... .... ...0 Skip
0x0004	2	ServiceID	
0x0006	2	PMT PID	
0x0008	2	PCR PID	
0x000A	2	Video PID	
0x000C	2	Audio PID	100. .... .... .... Auto 001. .... .... .... Dolby Digital channel 000. .... .... .... Fixed language
0x000E	1	Flags	1... .... Rename Flag .000 00.. unused .... ..00 Sound Mode
0x000F	1	NVOD Index	
0x0010	4	Service Name Offset	Points to the beginning of a string in block 10 (=0x0003C4D8 + Offset)
0x0014	2	Provider Index	Points to then n <sup>th</sup> string in block 11
0x0016	10	Unused	

## RADIO SERVICE DATA [2]

See [TV Service Data \[1\]](#). The VideoPID is always 0x0000.

## SATELLITE TABLES [3]

A single record contains 76 bytes.

```

00 75 FF FF H o t b i r d 00 00 00 00 00 00 00 00 00
82 00 A6 16 29 68 7F 00 00 01 00 00 00 00 00 00
82 00 A6 16 29 68 7F 00 00 89 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 82 00 00 00 00 00 00

00 4A FF FF A s t r a 00 00 00 00 00 00 00 00 00 00
81 00 A6 16 29 68 7F 00 00 01 00 00 00 00 00 00
81 00 A6 16 29 68 7F 00 00 81 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 c0 00 00 00 00 00 00
  
```

Offset	Length	Field	Remarks
0x0000	2	Number of Transponders	
0x0002	2	Reserved	
0x0004	16	Satellite Name	
0x0014	1	Tuner 1: LNB-Flags & DiSEqC 1.0	1... .. LNB Power .0... .. SW12V (12V switched output is not implemented on (most?) Toppies. ..00 0010 DiSEqC 1.0 (0 = Disable, 1 = 1 of 4, 2 = 2 of 4, 3 = 3 of 4, 4 = 4 of 4, 5 = Mini A, 6 = Mini B)
0x0015	1	Tuner 1: DiSEqC 1.2	0... .. Motorized .000 0000 stabPosition
0x0016	2	Tuner 1: LNB Low Band Oscillator	1... .. Universal LNB .0... .. 22kHz Switchbox ..10 0110 0001 0110 Frequency
0x0018	2	Tuner 1: LNB High Band Oscillator	0010 1001 0110 1000 Frequency
0x001A	1	Tuner 1: DiSEqC 1.2 USALS	0... .. USALS .111 1111 posIndex
0x001B	1	Tuner 1: DiSEqC 1.2 Vertical Skew	
0x001C	1	Tuner 1: DiSEqC 1.2 Hor. Skew	
0x001D	1	Tuner 1: Tuner Loop Trough	0000 00.. Magic .... ..01 Connection Type (00=sep., 01=loop)
0x001E	1	Tuner 1: DiSEqC 1.1	0 = Disable, 1 = 1 of 4, 2 = 2 of 4, 3 = 3 of 4, 4 = 4 of 4
0x001F	5	Reserved	
0x0024	1	Tuner 2: LNB-Flags & DiSEqC 1.0	0... .. USALS .111 1111 posIndex
0x0025	1	Tuner 2: DiSEqC 1.2	
0x0026	2	Tuner 2: LNB Low Band Oscillator	
0x0028	2	Tuner 2: LNB High Band Oscillator	
0x002A	1	Tuner 2: DiSEqC 1.2 USALS	
0x002B	1	Tuner 2: DiSEqC 1.2 Vertical Skew	
0x002C	1	Tuner 2: DiSEqC 1.2 Hor. Skew	
0x002D	1	Tuner 2: Tuner Loop Through	1000 00.. Magic .... ..01 Connection Type (00=sep., 01=loop)
0x002E	1	Tuner 2: DiSEqC 1.1	
0x002F	5	reserved	
0x0034	12	DefaultChannel	define.h is talking about a TYPE_DefaultChannel (maybe outdated info?)
0x0040	4	SatAngle	
0x0044	2	Satellite Position	Longitude*10
0x0046	6	reserved	

## TRANSPONDER DATA [4]

A single transponder record consists of 16 bytes. The first four bytes hold the number of used transponder records.

```

00 00 01 7A
00 00 00 00 00 00 29 DF 6B 6C 2A F8 00 00 00 00
00 80 00 00 00 00 29 E3 74 CC 2B 5C 00 00 00 01
01 80 00 00 00 00 2A 50 55 F0 04 21 80 00 00 85
  
```

Offset	Length	Field	Remarks
	4	Number of Entries	
0x0000	1	Sat Index	
0x0001	1	Polarisation	1 . . . . . Hor. Pol. .000 . . . . . Mode (0=Normal, 1=SmaTV) . . . . . 0000 unused
0x0002	2	unused	
0x0004	4	Frequency	
0x0008	2	Symbol Rate	
0x000A	2	Transport Stream ID	
0x000C	1	User selected Time Sync	1 . . . . . selected .000 0000 unused
0x000D	1	unused	
0x000E	2	Network ID	



## CLOCK SETUP [7]

The following structure shows all flags of all models seen so far. As these are implemented by software, future firmware version might add more flags. The 5000 Masterpiece does not use the Automatic Time Offset-flag.

00 78 00 00 08 FF FF FF

Offset	Length	Field	Remarks
0x0000	2	UTC Offset	Offset in minutes
0x0002	2	Sleep Timer	
0x0004	1	Flags1	000. .... unused ...0 00.. GMT collection (see Transponder Block Offset 0x10) (00=Normal, 01=CAS Only, 10=User Select) .... ..00 Mode (0=Auto, 1=Manual)
0x0005	1	Flags2	1111 111. unused .... ...1 Time Offset (0=Manual, 1=Automatic)
0x0006	1	Flags3	11... .... DST (00=off, 11=on) ..11 1111 unused
0x0007	1	Flags4	1111 1111 unused

## TIMER BLOCK [8]

A single timer record consists of 136 bytes. The maximum length of the file name is 98 characters. This includes the terminating 0x00. On non-PVRs (e.g. the 5000CI), bytes 0x14 to 0x87 are unused.

```
00000000h: 01 40 01 01 00 0A 00 00 D0 38 13 2D D0 38 13 37 01 01 32 C9
00000014h: Wetter-1.rec...
00000074h: 00 00 01 DE 01 80 00 00 00 00 31 94 55 F0 04 5D 80 00 00 01
```

Offset	Length	Field	Remarks
0x0000	1	TunerIndex	
0x0001	1	Flags	010. .... Rec. Mode (0=none, 1=recording, 2=reserved recording, 3=temp. recording, 4=copy) ...0 0... Demux Path .... .00. Res. Type (1=manual recording (no SID, EventID)) .... ...0 unused
0x0002	1	SatIndex	
0x0003	1	Type & Reservation	0... .... Service Type (0= TV, 1=Radio) .000 0001 0 = One Time, 1 = Every Day, 2 = Every Weekend, 3 = Weekly, 4 = Every Weekday
0x0004	2	Duration	In minutes
0x0006	2	Service Index	
0x0008	4	Start Time	DDHM (DD=days since 17.11.1858)
0x000C	4	End Time <sup>*1</sup>	DDHM
0x0010	1	isRec	P = 0, R = 1
0x0011	1	setName	
0x0012	2	ServiceID	
0x0014	96	File Name	
0x0074	2	EPGMarker	2 = Rt
0x0076	2	EventID	
The following part has the same structure as block 4			
0x0078	1	SatIndex	
0x0079	1	Polarization	B7=Hor. Pol.
0x007A	4	Unused	
0x007E	2	Frequency	
0x0080	2	SymbolRate	
0x0082	2	TransportStreamID	
0x0084	1	Scrambled Service	1... .... Scrambled .000 0000 unused
0x0085	1	Unused	
0x0086	2	NetworkID	

<sup>\*1</sup> Due to a bug, this field may contain the duration. It is safer to calculate the end time by adding the start time and the duration

## RESERVE EPG BLOCK [9]

This block contains 70 EventID-StartTime pairs. The most recent entry found in this block was from December 2004 and on the 5800 & 5200 models, this block isn't used (0xFF). So it seems that this block is not used in current firmware versions any more.

```
00 00 04 39 D0 6E 00 33
```

Offset	Length	Field	Remarks
0x0000	2	Flags	00.. .... reserved ..0. .... Service Type (0=Radio, 1=TV) ...0 0000 0000 0000 Service Index
0x0002	2	EventID	
0x0004	4	Start Time	

### [SERVICE NAMES \[10\]](#)

This block simply contains a list of NULL-terminated strings.

### [PROVIDER NAMES \[11\]](#)

This table contains the name strings of the service providers. Every record is 21 bytes long. The end of the table is indicated by a zero length string.

### [OTA \[12\]](#)

This block does not contain any useful data.

## TF5800 PVRt TV SERVICE DATA [1]

A single record contains 32 bytes. The allocated space can hold up to 1000 services. Unused records are marked by a Tuner value of 00 (this is used to detect the last entry).

```
00 00 0E B0 57 00 02 C0 01 91 01 91 81 92 20 00 00 00 01 35 00 03 00 0C 00
00 00 00 00 00 00 00
```

Offset	Length	Field	Remarks
0x0000	2	Unused	
0x0002	2	Transponder Index & Flags	0000 1110 10.. .... Transponder Index (Block [4]) .... .... ..11 .... Tuner (11= both Tuner, 10 = Tuner 2, 01 = Tuner 1) .... .... .... 0... Delete .... .... .... .0.. CAS .... .... .... ..0. Lock .... .... .... ...0 Skip
0x0004	2	ServiceID	
0x0006	2	PMT PID	
0x0008	2	PCR PID	
0x000A	2	Video PID	
0x000C	2	Audio PID	B15 = always set
0x000E	1	Flags	0... .... Rename Flag .010 00.. <b>unknown</b> .... ..00 Sound Mode
0x000F	1	NVOD Index	
0x0010	4	Service Name Offset	Points to the beginning of a string in block 10 (=0x000156B0 + Offset)
0x0014	2	Provider Index	Points to then n <sup>th</sup> string in block 11
0x0016	2	Logical Channel Number	
0x0018	8	Unused	

## RADIO SERVICE DATA [2]

See [TV Service Data \[1\]](#). The VideoPID is always 0xFFFF.

## SATELLITE TABLES [3]

```
00 72 44 56 42 54 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Offset	Length	Field	Remarks
0x0000	2	Number of Transponders	
0x0002	4	String "DVBT"	
0x0004	26	unused	

## TRANSPONDER DATA [4]

A single transponder record consists of 16 bytes. The first four bytes hold the number of used transponder records. On Australian machines, the channel number is coded in BCD instead of binary. In addition the channel 9A is shown as 0x009A.

00 00 00 72

00 2A 08 00 00 09 CB D0 50 00 00 00 23 3A 00 00

Offset	Length	Field	Remarks
	dword	Number of Entries	
0x0000	byte	SatIdx	Always 0
0x0001	byte	Channel Number	
0x0002	byte	Bandwidth	[MHz]
0x0003	byte	unused	
0x0004	dword	Frequency	[kHz]
0x0008	word	Transport Stream ID	
0x000A	byte	LP/HP Stream	0 = LP Stream, 1= HP Stream
0x000B	byte	unused	
0x000C	word	OriginalNetwork ID	
0x000E	word	NetworkID	
<b>TF5700PVRt only</b>			
0x0010	byte	Code Rate – HP Stream	1/2, 2/3, 3/4, 5/6, 7/8
0x0011	byte	Guard Interval	1/32, 1/16, 1/8, 1/4
0x0012	byte	Transmission Mode	2k, 8k, 4k
0x0013	byte	unused	
0x0014	byte	Constellation	QPSK, 16-QAM, 64-QAM
0x0015	byte	unused	
0x0016	byte	unused	
0x0017	byte	unused	

Note: The transponder block on a TF5700PVRt (HDMI) is 8 bytes longer than on other machines.



## CLOCK SETUP [7]

The following structure shows all flags of all models seen so far. As these are implemented by software, future firmware version might add more flags. The 5800 does not use the GMT Collection-flags and the DST-flags.

00 3C 00 00 E0 01 FF FF

Offset	Length	Field	Remarks
0x0000	2	UTC Offset	Offset in minutes
0x0002	2	Sleep Timer	
0x0004	1	Flags1	111. .... unused ...0 00.. GMT collection (see Transponder Block Offset 0x10) (00=Normal, 01=CAS Only, 10=User Select) .... ..00 Mode (0=Auto, 1=Manual)
0x0005	1	Flags2	0000 000. unused .... ...1 Time Offset (0=Manual, 1=Automatic)
0x0006	1	Flags3	11... .... DST (00=off, 11=on) ..11 1111 unused
0x0007	1	Flags4	1111 1111 unused

## TIMER BLOCK [8]

A single timer record consists of 136 bytes. The maximum length of the file name is 98 characters. This includes the terminating 0x00.

```
00000000h: 03 42 00 00 00 40 00 01 D1 9A 14 04 D1 9A 15 08 01 01 10 85
00000014h: Secret of Drawing_10-15.rec
00000074h: 00 00 00 00 00 35 08 00 00 0B 23 90 10 05 00 00 23 3A 00 00
```

Offset	Length	Field	Remarks
0x0000	1	TunerIndex	
0x0001	1	Flags	010. .... Rec. Mode (0=none, 1=recording, 2=reserved recording, 3=temp. recording, 4=copy) ...0 0... Demux Path .... .00. Res. Type (1=manual recording (no SID, EventID)) .... ...0 unused
0x0002	1	unused	
0x0003	1	Mode & Reservation Type	0... .... Service Type (0= TV, 1=Radio) .000 0001 0 = One Time, 1 = Every Day, 2 = Every Weekend, 3 = Weekly, 4 = Every Weekday
0x0004	2	Duration	In minutes
0x0006	2	Service Index	
0x0008	4	Start Time	DDHM (DD=days since 17.11.1858)
0x000C	4	End Time <sup>*1</sup>	DDHM
0x0010	1	isRec	P = 0, R = 1
0x0011	1	setName	
0x0012	2	ServiceID	
0x0014	96	File Name	
0x0074	2	EPGMarker	2 = Rt
0x0076	2	EventID	
The following part has the same structure as block 4			
0x0078	byte	SatIdx	Always 0
0x0079	byte	Channel Number	
0x007A	byte	Bandwidth	[MHz]
0x007B	byte	unused	
0x007C	dword	Frequency	[kHz]
0x0080	word	Transport Stream ID	
0x0082	byte	LP/HP Stream	0 = LP Stream, 1= HP Stream
0x0083	byte	unused	
0x0084	word	OriginalNetwork ID	
0x0086	word	NetworkID	
<b>TF5700PVRt only</b>			
0x0088	byte	Code Rate – HP Stream	1/2, 2/3, 3/4, 5/6, 7/8
0x0089	byte	Guard Interval	1/32, 1/16, 1/8, 1/4
0x008A	byte	Transmission Mode	2k, 8k, 4k
0x008B	byte	unused	
0x008C	byte	Constellation	QPSK, 16-QAM, 64-QAM
0x008D	byte	unused	
0x008E	byte	unused	
0x008F	byte	unused	

<sup>\*1</sup> Due to a bug, this field may contain the duration. It is safer to calculate the end time by adding the start time and the duration

Note: The transponder block on a TF5700PVRt (HDMI) is 8 bytes longer than on other machines.

### RESERVEEPG BLOCK [9]

This doesn't contain any data on the 5800.

### SERVICE NAMES [10]

This block simply contains a list of 0x00-terminated strings.

### PROVIDER NAMES [11]

This table contains the name strings of the service providers. Every record is 21 bytes long. The end of the table is indicated by a zero length string.

## TF5200 PVRc TV SERVICE DATA [1]

A single record contains 32 bytes. The allocated space can hold up to 1000 services. Unused records are marked by a Tuner value of 00 (this is used to detect the last entry).

```
00 00 02 F0 00 12 00 67 0A FF 0A FF 8B 00 20 00 00 00 04 FE 00 06 04 56 00
00 00 00 00 00 00 00
```

Offset	Length	Field	Remarks
0x0000	2	Unused	
0x0002	2	Transponder Index & Flags	0000 0010 11... .. Transponder Index (Block [4]) ..... .. .11 ..... Tuner (11= both Tuner, 10 = Tuner 2, 01 = Tuner 1) ..... .. .0... Delete ..... .. .0.. CAS ..... .. .0. Lock ..... .. .0 Skip
0x0004	2	ServiceID	
0x0006	2	PMT PID	
0x0008	2	PCR PID	
0x000A	2	Video PID	
0x000C	2	Audio PID	B15 = always set
0x000E	1	Flags	0... .. Rename Flag .010 00.. <b>unknown</b> .... ..00 Sound Mode
0x000F	1	NVOD Index	
0x0010	4	Service Name Offset	Points to the beginning of a string in block 10 (=0x0001519C + Offset)
0x0014	2	Provider Index	Points to then n <sup>th</sup> string in block 11
0x0016	2	Logical Channel Number	
0x0018	8	Unused	

## RADIO SERVICE DATA [2]

See [TV Service Data \[1\]](#). The VideoPID is always 0xFFFF.

## SATELLITE TABLES [3]

```
00 72 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Offset	Length	Field	Remarks
0x0000	2	Number of Transponders	
0x0002	30	Unused	



## GAME DATA [6]

```

00 07 A1 20 00 0F 42 40 00 0F 42 40 00 0F 42 40
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 02 FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF

```

Offset	Length	Field	Remarks
0x0000	4 x 4	Money	
0x0010	4 x 4	Win	
0x0020	4 x 4	Lose	
0x0030	4	exblockClearStage	
0x0034	16	reserved	

## CLOCK SETUP [7]

The following structure shows all flags of all models seen so far. As these are implemented by software, future firmware version might add more flags. The 5200 does not use the GMT Collection-, Time Offset- and the DST- flags.

```

00 78 00 00 E0 00 FF FF

```

Offset	Length	Field	Remarks
0x0000	2	UTC Offset	Offset in minutes
0x0002	2	Sleep Timer	
0x0004	1	Flags1	111. . . . unused . . . 0 00.. GMT collection (see Transponder Block Offset 0x10) (00=Normal, 01=CAS Only, 10=User Select) . . . . . 00 Mode (0=Auto, 1=Manual)
0x0005	1	Flags2	0000 000. unused . . . . . 0 Time Offset (0=Manual, 1=Automatic)
0x0006	1	Flags3	11.. . . . DST (00=off, 11=on) . . 11 1111 unused
0x0007	1	Flags4	1111 1111 unused

## TIMER BLOCK [8]

A single timer record consists of 132 bytes. The maximum length of the file name is 98 characters. This includes the terminating 0x00.

```
00000000h: 03 42 00 00 00 18 00 0D D1 9D 12 3B D1 9D 13 17 01 01 6D 67
00000014h: heute-Nachrichtenmagazin des ZDF-18-10.rec
00000074h: 00 00 00 00 00 06 03 10 1A F4 04 37 00 01 02 00
```

Offset	Length	Field	Remarks
0x0000	1	TunerIndex	
0x0001	1	Flags	010. .... Rec. Mode (0=none, 1=recording, 2=reserved recording, 3=temp. recording, 4=copy) ...0 0... Demux Path .... .00. Res. Type (1=manual recording (no SID, EventID)) .... ...0 unused
0x0002	1	unused	
0x0003	1	Mode & Reservation Type	0... .... Service Type (0= TV, 1=Radio) .000 0001 0 = One Time, 1 = Every Day, 2 = Every Weekend, 3 = Weekly, 4 = Every Weekday
0x0004	2	Duration	In minutes
0x0006	2	Service Index	
0x0008	4	Start Time	DDHM (DD=days since 17.11.1858)
0x000C	4	End Time <sup>*1</sup>	DDHM
0x0010	1	isRec	P = 0, R = 1
0x0011	1	setName	
0x0012	2	ServiceID	
0x0014	96	File Name	
0x0074	2	EPGMarker	2 = Rt
0x0076	2	EventID	
The following part has the same structure as block 4			
0x0078	4	Frequency	[kHz]
0x007C	2	Symbol Rate	
0x007E	2	Transport Stream ID	
0x0080	2	Network ID	
0x0082	1	Modulation	0=16QAM, 1=32QAM, 2=64QAM, 3=128QAM, 4=256QAM
0x0083	1	unused	

<sup>\*1</sup> Due to a bug, this field may contain the duration. It is safer to calculate the end time by adding the start time and the duration

## RESERVEEPG BLOCK [9]

This doesn't contain any data on the 5800.

## SERVICE NAMES [10]

This block simply contains a list of 0x00-terminated strings.

## PROVIDER NAMES [11]

This table contains the name strings of the service providers. Every record is 21 bytes long. The end of the table is indicated by a zero length string.

## CONTENTS OF THE EEPROM

The size of the EEPROM is 256 bytes (24WC02). It is divided into several blocks, which start with a CRC16 (see below for the algo). Some models do not use all blocks.

```

00000000h: 06 0A 00 1A 00 3E 00 02 FF FF FF FF 11 02 0F 20
00000010h: 05 04 FF 0C 0C 0C 01 00 FF FF 24 00 00 00 00
00000020h: 00 00 00 00 00 00 00 00 31 C0 00 00 FF 02 1B 00
00000030h: FF FF 00 00 00 00 3A C0 00 A4 01 E2 F4 76 00 FF
00000040h: 00 03 00 00 00 00 00 00 9E 5B FC 05 FF FF FF FF
00000050h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000060h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000070h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000080h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000090h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000000A0h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000000B0h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000000C0h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000000D0h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000000E0h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000000F0h: FF FF FF FF FF FF FF FF FF FF FF FF 44 FF FF 00
  
```

Absolute	Relative	Length	Field	Remarks
<b>Etc Info</b>				
0x00	----	2	CRC16	
0x02	0x00	2	TV Channel Sub	
0x04	0x02	2	TV Channel Main	
0x06	0x04	2	Radio Channel	
0x08	0x06	2	Old TV Channel	
0x0A	0x08	2	Old Radio Channel	
0x0C	0x0A	1	Volume	
0x0D	0x0B	1	Flags	0... .. Service Type (0 = TV, 1 = Radio) .00. .... Sound Mode (0=Stereo, 1=Mono, 2=Left, 3=Right) ...0 0... Scart Output Source ... .01. Video Output (0=CVBS, 1=RGB, 2=S-Video, 3=YUV) ... ..0 TV Aspect Ratio (0=4:3, 1=16:9)
0x0E	0x0C	1	Configuration Flags	.0... .. N(C)PC (has to do with the 5500 VFD) ..00 .. ??
0x0F	0x0D	1	Flags	001. .... RF Output (0=NTSC-M, 1=PAL-G, 2=PAL-I, 3=PAL-K) ...0 .... Disable Subtitle ... 0... Disable Teletext ... .00. 16:9 Display Format (0=letterbox, 1=center extract) ... ..0 Dolby
0x10	0x0E	1	InfoBox Hide Time	In seconds
0x11	0x0F	1	Active Fav. Number	
0x12	0x10	1	Current Fav. Group Nr.	(0xFF means that no group is active)
0x13	0x11	1		0000 11.. Infobox Position ... ..00 TV Type (0=Multi, 1=PAL, 2=NTSC)
0x14	0x12	1	Transparency	
0x15	0x13	1		0... .. MovingConfirmBox .0... .. satDelete ..0. .... Scart Type (0=Standard, 1=External A/V) ...0 1... PIP Position ... .1.. Svc Help Window ... ..0. Time Shift ... ..0 unused
0x16	0x14	1	Current Tuner Sub	0=Tuner 1, 1=Tuner 2
0x17	0x15	1	Current Tuner Main	0=Tuner 1, 1=Tuner 2
0x18	0x16	2	MultiFeed Service Nr.	
0x1A	0x18	1	rfChannel	UHF channel number
0x1B	0x19	1	Unused	
0x1C	0x1A	1	Unused	
0x1D	0x1B	1	Factory Country Code	1 for "Italy", 2 for "Other Europe"
0x1E	0x1C	1	Flags	0... .. 12/24h formt (1=12h Mode) ..0. .... MHEG Disable (1=disabled)
0x1F	0x1D	1	OldTVFavGroup	

Absolute	Relative	Length	Field	Remarks
0x20	0x1E	1	OldRadioFavGroup	
0x21	0x1F	1	unused	

#### Parental Info

0x22	----	2	CRC16	
0x24	0x00	2	PIN Code	
0x26	0x02	1	Rate	0=No block, 1=4 years, 2=5 years, ..., 15=18 years, 16=Total block
0x27	0x03	1	Lock Flags	B7 = Time Setting locked/unlocked B6 = Language Setting B5 = A/V Output Setting B4 = Organizing Services B3 = Organizing Favorites B2 = Common Interface B1 = Installation B0 = Boot Lock

#### Language Info

0x28	----	2	CRC16	
0x2A	0x00	1	Menu	0 = English, 1 = French, 2 = Deutsch, 3 = Italian, 4 = Spanish, 5 = Arabic, 6 = Greek, 7 = Türkce, 8 = Danish, 9 = Swedish, 10 = Norwegian, 11 = Dutch, 12 = Russian, 13 = Polish, 14 = Persian, 15 = Suomi, 16 = Slovak, 17 = Thai, 18 = Czech
0x2B	0x01	1	Teletext	See above
0x2C	0x02	1	1 <sup>st</sup> Subtitle	See above
0x2D	0x03	1	1 <sup>st</sup> Audio	See above

#### Recent Info

0x2E	----	2	CRC16	
0x30	0x00	1	2 <sup>nd</sup> Subtitle	See Language info
0x31	0x01	1	2 <sup>nd</sup> Audio	See Language info
0x32	0x02			
0x35	0x35			

#### MyPos Info (USALS)

0x36	----	2	CRC16	
0x38	0x00	2	Longitude	*10
0x3A	0x02	2	Latitude	*10

#### List Info

0x3C	----	2	CRC16	
0x3E	0x00	1	Show Provider	(0=off, 1=on)
0x3F	0x01	1	Sat Index	
0x40	0x02	1	Sort Mode	(0=none, 1=name, 2=Satellite, 3=FTA/CAS, 4=CAS/FTA, 5=Provider, 6=FAV/No FAV)
0x41	0x03	1	Tuner	(1=Tuner 1, 2=Tuner 2, 3=All Tuner)
0x42	0x04	1	Sort Order	(0=Sort on time/date, 1=Sort alphabetically, 2=Sort on file size)
0x43	0x05	5	Unused	

#### ?

0x48	----	2	CRC16	
0x4A	0x00			
0x51	0x49			

#### ?

0x52	----	170		Unused <sup>1</sup>
0xFC	----	1	Power Status	0x44 = Standby, 0x88 = On
0xFD	----	1		Unused <sup>1</sup>
0xFE	----	1		Unused <sup>1</sup>
0xFF	----	1		0x00

<sup>1</sup> the default pattern differs on different machines. Maybe these locations aren't touched by factory reset and contain the original manufacturer's pattern.

## STRUCTURE OF THE .STD FILE

This file holds the user settings located in the flash memory and the contents of the EEPROM memory. While the EEPROM data is uncompressed, the data inside of the flash memory is compressed using Haruhiko Okomuras AR002 algorithm (used by the old DOS LHArc V2.x). These up to 64 x 8kB blocks have nothing in common with the internal data structures.

00000-0003F: The .std file header.  
 00040-0007F: The flash header. This is the real start of the flash memory area 2 (user settings).  
 00080-0107F: The directory for the compressed data blocks is located here.  
 01080-0143F: 0x00  
 01440-2003F: Compressed flash data blocks.  
 20040-2013F: Uncompressed contents of the EEPROM

### .STD-FILE HEADER

```
00000000: ToPFIeLd
00000008: 00 02 (File Version)
0000000A: 01 96 (System ID = 406)
0000000C: 1F 1D 1D FF 01 01 96 06 00 00 00 8F 22 1E D0 90 (ModelID)
0000001C: 0F 04 00 88 10 13 FF 0B B8 0D AC 05 DC 00 17 02 (ModelID contd.)
0000002C: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF (Padding)
```

On machines with the ST5518 processor (e.g. TF6060CI), the ModelID is 36 bytes long due to the way how the compiler manages the bitfields on this little endian processor. In this case only 16 padding bytes are used so that the header is always 64 bytes long.

### FLASH HEADER

```
00000040: 1F 1D 1D FF 01 01 96 06 00 00 00 8F 22 1E D0 90 (ModelID)
00000050: 0F 04 00 88 10 13 FF 0B B8 0D AC 05 DC 00 17 02 (ModelID contd.)
00000060: 00 40 00 00 00 00 00 00 00 00 00 00 00 00 00 (Nr. of 8k Blocks)
00000070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

### DIRECTORY

Every entry is 64 bytes long.

	CRC	-File Offset-Packed Size-Unpkd. Size	
00000080h	AE 70-18 9A-00 00 14 00-00 00 04 D1-00 00 20 00		CRC Packed Data=AE70; CRC Unpacked Data=189A, Offset=00001400 (00001440 in the .std-File because of the .std file header), Packed Size=04D1, Unpacked Size=2000
000000c0h	A0 85-91 BC-00 00 1C 00-00 00 00 42-00 00 20 00		The next directory entry
00000100h	A0 85-91 BC-00 00 20 00-00 00 00 42-00 00 20 00		
00000140h	A0 85-91 BC-00 00 24 00-00 00 00 42-00 00 20 00		
00000180h	A0 85-91 BC-00 00 28 00-00 00 00 42-00 00 20 00		
000001c0h	F7 CC-8A E0-00 00 2C 00-00 00 00 46-00 00 20 00		
00000200h	BD 27-59 1E-00 00 70 00-00 00 00 3E-00 00 20 00		
00000240h	BD 27-59 1E-00 00 78 00-00 00 00 3E-00 00 20 00		

Every data block is a full TFD file (SystemID=0xFFFF, Nr of blocks=1). See below for format details. Every data block has to start at a 1k boundary. There is a minor difference in the default-block of a TFD-file: it just contains only two valid entries. The first one is for the default settings and the second for the default EEPROM. Therefore the default settings block contains more than one TFD block.

## STRUCTURE OF THE .TFD FILE

The .tfd-format is used for PC => Topfield transfers. If Vega is used to upload settings, a temporary .tfd file is created and its contents is transferred to the Topyy. The .tfd-file may contain more than one data type (see [Available blocks](#)). The maximum size of the uncompressed blocks is 32762 (\$7FFA) bytes. The data is compressed using the same AR002 algorithm as the .std file.

Sample .tfd file:

```
File Header
00000000h: 00 08 0E 98 01 96 00 01 00 06 - HeaderLength-2 Header CRC (SystemID+FileVersion+NrofBlock)
                                         SystemID FileVersion NrofBlocks

Data blocks:
00000000ah: 10 DE 3A F9 00 02 7F FA 0F D7 87 5D ...- DataBlockSize (incl. Header-2) CRC (BlockType+
000010e7h: 0A 8D DD 97 00 02 7F FA 0A 04 7E DB ...- UncompressedSize+Data) BlockType UncompressedSize
00001b76h: 05 7E 1A 9A 00 02 7F FA 05 15 75 EE ...
000020f6h: 00 8F 6F 5E 00 02 7F FA 00 B7 54 54 ...
00002187h: 00 0F BF 86 00 02 00 18 00 02 20 04 ...
00002198h: 00 4C 5C 8D 00 03 01 00 00 33 4A 6D ...
000021e6h: <EOF>
```

The above sample consists of 5 type-2 blocks (Flash settings) and a single type-3 block (EEPROM).

Inside of the flash memory, the block header is a little bit different and endian format is inconsistent. On a 5000MP, the secondary loader uses little endian and the firmware uses big endian words.

Sample secondary loader:

```
bfc02000h: 00 80 5E 2E 20 A2 85 EF 5F 7D ...- UncompressedSize CompressedSize
bfc04e62h: 00 80 8D 1E 15 DF 83 F0 FB D6 ...
...
bfc0c2f9h: FE FE DE 77 FF FF ...- End of data
```

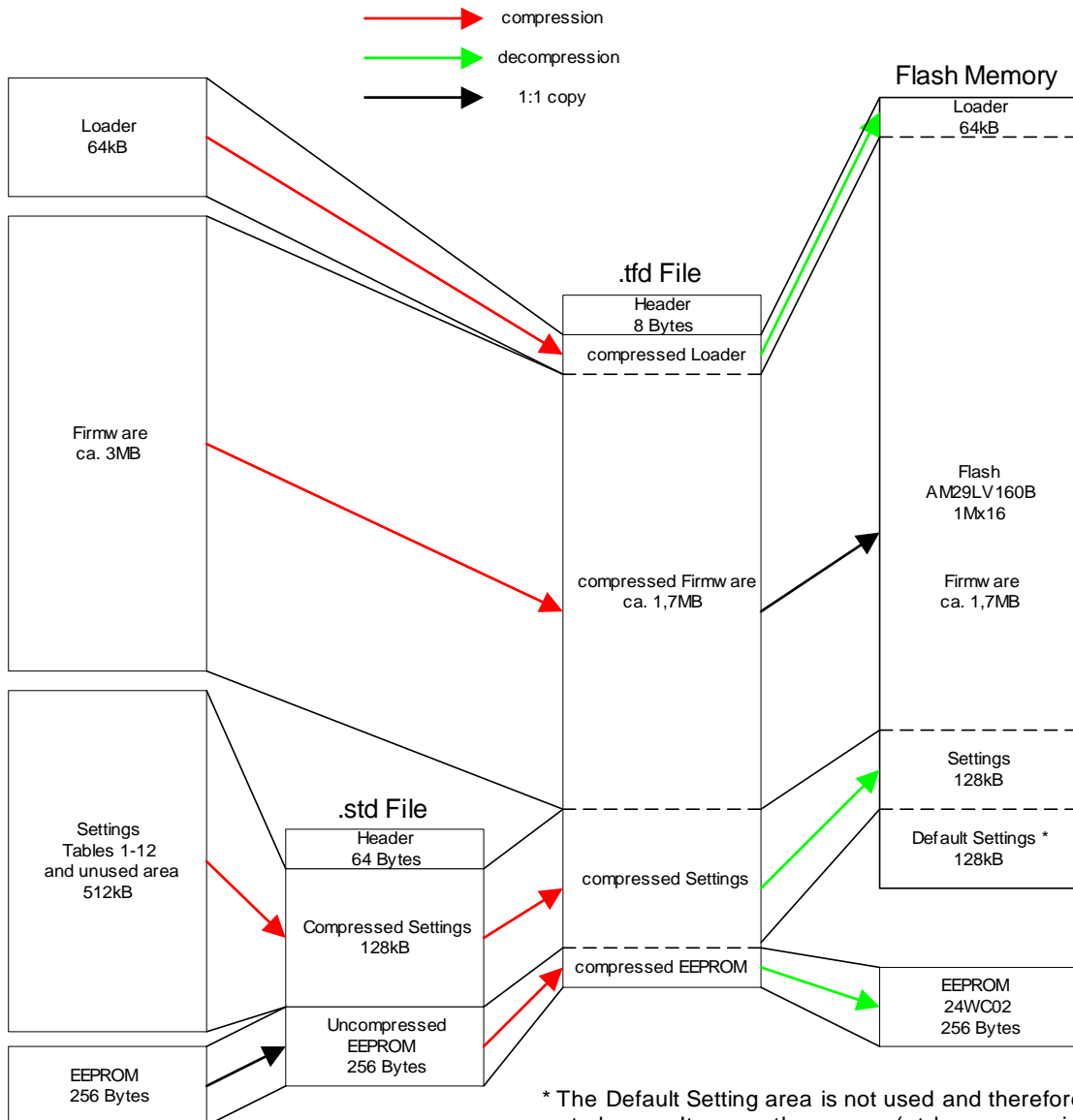
Sample firmware:

```
bfc20000h: 7F FA 2E 42 41 D0 20 8A 84 DD ...- UncompressedSize CompressedSize (incl. CRC-16) CRC-16
bfc22e62h: 7F FA 27 A0 30 4F 1B 14 84 F7 ...
...
bfd88ce3h: FE FE DE 77 FF FF ...- End of data
```

## COMMUNICATING WITH THE LOADER

Note: While Topfield uses the terms „Download“ and „Upload“ from the STBs standpoint (download means a transfer from the PC to the STB), this document uses “PC => STB” and “STB => PC” to minimize confusion.

When transferring data from the Topfield, all blocks are transferred as they are saved in the flash/EEPROM memory. This means that the EEPROM and loader are uncompressed and the user and default settings use the std-format. The only exception is the firmware, which is stored in the tfd-format but is decompressed on the STB => PC transfer. On PC => STB transfers, all blocks have to be packed into a single tfd file. The loader, firmware and EEPROM are uncompressed and the settings and default settings use the std-format.



Serial communication parameters: 115200 / N / 8 / 1

## DATA PACKETS

The transfer follows the same scheme for both directions. Every command (with the exception of CMD06 and CMD0B) has to be acknowledged.

Serial: ToFi <Seq:1> <Cmd:1> <Len:1> <Data:n> <CRC:1>

USB: ToFi <Seq:1> <Cmd:1> <Len:2> <Data:n> <CRC:1>

Seq is the sequence number. Every answer to a command needs to have the same sequence number as the command itself. Commands which belong together are marked with the same color in the following table.

CRC The CRC uses a very simple formula: (Summ of all bytes after the ToFi-string) AND 0xFF.

Cmd	Name	Data Len	Direction	Data	Notes
01	CMD_ChkReq	34	P ← T	<SystemID:2> + <ModelID:32>	Identification of the receiver
02	CMD_ChkReply	00	P → T	-	CMD01-answer to initiate a PC => STB transfer
03	CMD_UploadReq	00	P → T	-	CMD01-answer to initiate a STB => PC transfer
04	CMD_DataReq	04 or 05	P ⇆ T	NumberOfBytes:1 or 2 StartAddress:3	Request data. This command may reset the sequence number. Maximum blocksize: serial=255, USB=32762
05	CMD_DataReply	n	P ⇆ T	Data:n	Answer to CMD04
06	CMD_Terminate	00	P ⇆ T	-	End of the communication (End). There's no answer to this CMD.
07	CMD_DataSizeReq	01	P ⇆ T	Block:1	Prepare transfer (see below for available blocks)
08	CMD_DataSizeReply	03	P ⇆ T	BlockLength:3	Answer to CMD07 (total length of the requested block)
09		00	P → T	-	Request STB-identification
0A		34	P ← T	<SystemID:2> + <ModelID:32>	Answer to CMD09 (same content as CMD01 after a Topyy reset)
0B	CMD_Reboot	00	P → T	-	STB Reboot; the sequence number restarts at 0x00; no answer
FF		n	P ← T	Message:n	Loader information in clear text. Sent by the STB after a reset and doesn't expect an answer.

## AVAILABLE BLOCKS

Block	Name	Display	
00	DATA_TYPE_LoaderPgm	L	Loader
01	DATA_TYPE_ApplPgm	A	Firmware
02	DATA_TYPE_FlashData	C	User Settings
03	DATA_TYPE_EepromData	E	EEPROM
04	DATA_TYPE_PrePgm	P	Factory Settings
05	DATA_TYPE_Background	?	Radio Background Picture <sup>*1</sup>

Trying to access blocks 06 to FF results in an error E-08 (invalid format). The loader seems to be write-protected on some machines and trying to write that block produces an E-04 (Error writing Flash).

\*1 seen in the TF6060CI firmware. It contains a tfd compressed MPEG PS stream with a single frame.

## ORDER OF THE COMM EVENTS

Open the interface (some DeviceIOControls have to be sent to correctly initiate USB transfers)

Send a Reset (CMD0B)

Wait about 300ms to make sure that the Toppo has shut down its interfaces

Close and reopen the interface

Receive and ignore the loader data

Receive and ignore the loader info strings (CMDFF)

Receive the STB ID (CMD01) and transmit a download request (CMD03) to initiate a STB => PC sequence...

...or receive a CMD01 and transmit an upload request (CMD02) to initiate an PC => STB sequence

(Note: the above CMD02/CMD03 is time critical. If the response isn't sent within some ms, the Toppo will continue to boot into user mode!)

(Note: on USB transfers, the Toppo will send the CMD01 after it has received the CMD02/CMD03. Therefore the CMD02/CMD03 command has to be sent after receiving the CMDFF string)

(Note: there seems to be a bug in the USB implementation: the command byte of the CMD02 has to be 0x01 instead of 0x02 to be accepted by the Toppo!)

On STB => PC transfers:

Send a CMD07 and receive the CMD08 response to get the size of the block

Repeatedly send CMD04s and receive the requested data via CMD05s

To receive another block move back to the CMD07, otherwise stop with a CMD06 (END) and an optional CMD0B (REBOOT).

On PC => STB transfers:

Repeatedly receive a CMD04 and answer with a CMD05 until you receive a CMD06 (END)

Send an optional CMD0B (REBOOT)

Close the interface

## EXAMPLE OF A SERIAL STB => PC TRANSFER WITH VEGA

Colors: <Seq> <Cmd> <Len> <CRC>

Vega (V) sends a reset to the Toppo (T) who responds with the boot message and the first "real" packet.

```
V: ToFi 00 0B 00 0B
T: 0D - 0D \ 0D | 0D 0A 0D 0A 'LOAD : NEC uPD613x Embedded Controller...' 0D 0A
T: ToFi 34 FF 18 0D 0A '<<< Loader L4.06 >>>' 0D 0A B2
```

Afterwards the Toppo sends the model and system ID, Vega answers with a download request (time critical!)

```
T: ToFi 35 01 22 01 96 1F 1D 1D FF 01 01 96 06 00 00 00 8F 22 1E D0 90 0F 04 00 88
    10 13 FF 0B B8 0D AC 05 DC 00 17 02 47
V: ToFi 35 03 00 38
```

Vega now wants to receive the user settings, the Toppo answers with its length (0x020000)

```
V: ToFi 36 07 01 02 40
T: ToFi 36 08 03 02 00 00 43
```

Vega demands the first 255 bytes starting at 0x000000

```
V: ToFi 37 04 04 FF 00 00 00 3E
T: ToFi 37 05 FF 1F 1D 1D FF 01 ... 00 00 5F
```

and so on.....

Vega demands the last block (2 bytes starting at 0x01FFFE)

```
V: ToFi 39 04 04 02 01 FF FE 41
T: ToFi 39 05 02 00 00 40
```

Now Vega requests the EEPROM; Toppo: its size is 0x000100 bytes

```
V: ToFi 3A 07 01 03 45
T: ToFi 3A 08 03 00 01 00 46
```

Vega: send them

V: ToFi 3B 04 04 FF 00 00 00 42  
T: ToFi 3B 05 FF C3 CE 00 1A 00 1F ... 44 FF FF AA  
V: ToFi 3C 04 04 01 00 00 FF 44  
T: ToFi 3C 05 01 00 42

Vega: End of the communication and do a reboot

V: ToFi 3D 06 00 43  
V: ToFi 3E 0B 00 49

## CRC16 ALGORITHM

The calculation is done via a 256 word lookup table.

0000	C0C1	C181	0140	C301	03C0	0280	C241
C601	06C0	0780	C741	0500	C5C1	C481	0440
CC01	0CC0	0D80	CD41	0F00	CFC1	CE81	0E40
0A00	CAC1	CB81	0B40	C901	09C0	0880	C841
D801	18C0	1980	D941	1B00	DBC1	DA81	1A40
1E00	DEC1	DF81	1F40	DD01	1DC0	1C80	DC41
1400	D4C1	D581	1540	D701	17C0	1680	D641
D201	12C0	1380	D341	1100	D1C1	D081	1040
F001	30C0	3180	F141	3300	F3C1	F281	3240
3600	F6C1	F781	3740	F501	35C0	3480	F441
3C00	FCC1	FD81	3D40	FF01	3FC0	3E80	FE41
FA01	3AC0	3B80	FB41	3900	F9C1	F881	3840
2800	E8C1	E981	2940	EB01	2BC0	2A80	EA41
EE01	2EC0	2F80	EF41	2D00	EDC1	EC81	2C40
E401	24C0	2580	E541	2700	E7C1	E681	2640
2200	E2C1	E381	2340	E101	21C0	2080	E041
A001	60C0	6180	A141	6300	A3C1	A281	6240
6600	A6C1	A781	6740	A501	65C0	6480	A441
6C00	ACC1	AD81	6D40	AF01	6FC0	6E80	AE41
AA01	6AC0	6B80	AB41	6900	A9C1	A881	6840
7800	B8C1	B981	7940	BB01	7BC0	7A80	BA41
BE01	7EC0	7F80	BF41	7D00	BDC1	BC81	7C40
B401	74C0	7580	B541	7700	B7C1	B681	7640
7200	B2C1	B381	7340	B101	71C0	7080	B041
5000	90C1	9181	5140	9301	53C0	5280	9241
9601	56C0	5780	9741	5500	95C1	9481	5440
9C01	5CC0	5D80	9D41	5F00	9FC1	9E81	5E40
5A00	9AC1	9B81	5B40	9901	59C0	5880	9841
8801	48C0	4980	8941	4B00	8BC1	8A81	4A40
4E00	8EC1	8F81	4F40	8D01	4DC0	4C80	8C41
4400	84C1	8581	4540	8701	47C0	4680	8641
8201	42C0	4380	8341	4100	81C1	8081	4040

And here is the main loop in Basic:

```
CRC = 0
For Addr = StartAddr To EndAddr
  CRC = CRCLookupTable(Data(Addr) Xor (CRC And 255)) Xor (CRC \ 256)
Next
```

## ERROR CODES

E-01 Data error in downloaded data via UART (corrupted data)  
E-02 Corrupted firmware in Flash memory  
E-03 Communication error with UART  
E-04 Error in writing Flash memory  
E-05 Too big downloaded data for the Flash memory  
E-06 Error in system ID. Not compatible firmware with the receiver.  
E-07 Error in system ID. Not compatible firmware with the receiver.  
E-08 Error in downloaded data format  
E-09 Error in reading EEPROM  
E-10 Error in writing EEPROM  
E-11 Unsupported Flash memory  
E-13 Timeout error in OTA  
E-14 Data error in downloaded data via OTA (corrupted data)  
E-15 Communication error via USB

## MODELID

The ModelID is a 32 bytes data block, which contains information about the hard- and software of the settopbox.

```
typedef struct
{
    word           Magic;           // = 0x1F1D
    byte           DataLen;         // = 0x1D
    byte           Checksum;
    byte           Version;         // = 0x01
    word           SystemID;
    word           ProjectID;
    byte           AreaID;
    byte           OEMID;
    CASSpecType   CASSpec;
    HWSpecType    HWSpec;
    SWSpecType    SWSpec;
} ModelIDType; // 32 Bytes
```

```
typedef struct
{
    byte          NrCIs:2;
    byte          Conax:1;
    byte          Irdeto:1;
    byte          Res:4;
} CASSpecType; // 1 Byte
```

```

typedef struct
{
    byte      Tuner1:4;
    byte      Tuner2:4;

    word      V12:1;
    word      RF:3;
    word      tvScart:2;
    word      vcrSCart:1;
    word      RCA:2;
    word      SVideo:1;
    word      SPDIF:2;
    word      pstnModem:3;
    word      CableModem:1;

    byte      RS232C:1;
    byte      USB:3;
    byte      IEE1394:1;
    byte      Ethernet:2;
    byte      WirelessHi:1;

    dword     WirelessLo:1;
    dword     nSmartCard:2;
    dword     nFrontKeys:4;
    dword     Segment7:3;
    dword     BoxSize:2;
    dword     DRAMSize:8;
    dword     FlashSize:6;
    dword     EEPROMSize:4;
    dword     Positioner:1;
    dword     Skew:1;
} HWSpecType; //8 bytes

```

```

typedef struct
{
    byte      RTOS:4;
    byte      MiddleWare:4;

    byte      OSD:5;
    byte      BPP:3;

    byte      MaxSat;
    word      MaxTP;
    word      MaxTV;
    word      MaxRadio;
    word      MaxSvcName;
    byte      DataSize;
} SWSpecType; //12 Bytes

```

#### TF5000/5500:

```

<ModelID> = 1F 1D 1D FF 01 01 96 06 00 00 00 8F 22 1E D0 90
            0F 04 00 88 10 13 FF 0B B8 0D AC 05 DC 00 17 02

```

```

SetEdit    = 1F 00 00 00 1D 00 00 00 1D 00 00 00 84 DF 45 00
            84 00 00 00 01 00 00 00 40 48 C8 01 84 DF 45 00

```

## **MODEL / SYSTEMID TABLE**

<b>Device</b>	<b>SystemID</b>	<b>App Type</b>
Procaster VF PVR5101C	10416	
Procaster VF PVR5101T	10426	
TF100C	40053	TF-CCMTC
TF100T	30053	TF-CCMTF
TF3000CI	101	
TF3000CI	121	
TF3000CI	161	
TF3000CIC	1201	
TF3000CIP	102	
TF3000CIPpro	202	
TF3000CIpro	201	
TF3000CIpro	211	
TF3000COCI	10201	
TF3000COT	10203	
TF3000COT-d	10213	
TF3000COT-N	10223	
TF3000FA	100	
TF3000FE	103	
TF3000FEI	203	
TF3000FEI	223	
TF3000FI	200	
TF3000FI, TF3000FI-N (NTSC)	240	
TF3000PVR	208	
TF3000T(유럽향) (EUR)	243	
TF3000T(호주향)(AUS)	233	
TF3030F	290	
TF3030FE	293	
TF3100C	1205	
TF3100CO-C	10215	
TF3100COT	10225	
TF3100FE	105	
TF3100FE	145	
TF3100FEI	205	
TF3100FEI	245	
TF3100FEI	295	
TF3100FEP	104	
TF3100FEP	134	
TF3100FEPpro	204	
TF3100T	1215	
TF3200IR	3200	
TF3500FE	10200	
TF4000C	508	
TF4000CO	10500	
TF4000COC	10508	
TF4000COC(러시아향)	10538	
TF4000COC(우크라이나향)	10528	
TF4000COC(핀란드향)	10518	
TF4000CO-N	10510	
TF4000COT	10503	
TF4000COT	10523	
TF4000COT	10533	
TF4000COT-N	10513	
TF4000Fe (GER)	1505	TF-DSFE
TF4000FE, TF5000FE	505	TF-MSFE
TF4000Fi	1500	
TF4000Fi	2500	

TF4000Fi	5500	
TF4000Fi (ME)	500	TF-MSFI4
TF4000Fi (Telran)	560	
TF4000Fi (UAE)	530	
TF4000Fi Plus(중동향)	50610	
TF4000Fi-N	540	
TF4000Fi-NA	550	
TF4000Fi-NB	520	
TF4000IR Plus	14000	TF-GNIrP
TF4000NA	30503	
TF4000PVR	206	TF-JPC
TF4000PVR COCI	10206	
TF4000PVR Plus	3406	TF-NPC4P
TF4000PVR Plus	3506	
TF4000PVR Plus(영국향)	3446	
TF4000PVR-N	296	
TF4000T	523	
TF4000T(일반향) (EUR)	533	TF-SFT
TF4000T(호주향) (AUS)	543	TF-SFTA
TF400PVRc	3457	
TF400PVRt	3467	TF-NPC4PT
TF4010PVR Plus	3436	TF-NPCAP
TF4010T	563	TF-SFTA
TF4100Fi	3500	TF-MSFI
TF4100Fi	4500	TF-ESFI
TF4100PVRc	3456	TF-NPC4PC
TF4100PVRt	3466	TF-NPC4PT
TF4100PVRt	3496	
TF4400PVRt (AUS)	1496	TF-NPTA4
TF4410PVRt (AUS)	1497	TF-NPTA4
TF4500T (호주향)	30202	
TF5000CI	2501	
TF5000CI (EUR)	1501	TF-ESCI
TF5000CI (GER)	3501	TF-SCI
TF5000CI (ME)	501	TF-MSCI
TF5000CI Plus	4411	TF-ESCI16
TF5000CI Plus	4501	TF-MSCI16
TF5000CI Plus	4611	TF-ESCI16
TF5000CI Plus (EUR)	4511	TF-ESCI16
TF5000CI Plus(베네룩스향)	4521	
TF5000CIP	502	
TF5000Fe	2505	
TF5000Fe (EUR), TF4000Fe	1505	TF-ESFE
TF5000Fi	507	
TF5000PVR	2466	
TF5000PVR Black Panther / White Polar	446	TF-NPCS
TF5000PVR Cd	407	TF-NCPC
TF5000PVR Masterpiece	1416	TF-NNPC
TF5000PVR Masterpiece	1516	
TF5000PVR Masterpiece (JPN)	1417	TF-NNPCJ
TF5000PVR, TF5500PVR	406	TF-NPC, TF-NCPC
TF5000PVR, TF5500PVR	506	TF-NPC, TF-NCPC
TF5000PVRc	2406	
TF5000PVR-N	426	
TF5000PVRt (EUR & AUS)	416	TF-NPT, TF-NPTA
TF5000PVRt Black Panther / White Polar (AUS)	466	TF-NPTSA
TF5000PVRt Masterpiece (AUS)	1426	TF-NNPTA
TF5000PVRt-N	1406	
TF5000T	503	
TF5000T(베트남향)	513	

TF500PVRc	12417	TF-NCPCf
TF500PVRt	13417	TF-NCPTf
TF5010PVR Black Panther / White Polar	486	TF-NPCS
TF5010PVR Masterpiece	1456	TF-NNPC
TF5010PVR, TF5510PVR	436	TF-NPC, TF-NCPC
TF5010PVRtH	13446	TF-NCPT
TF5020PVR HDMI	437	TF-NPC
TF5030PVR HDMI	447	
TF5050CI	5501	TF-ES50CID
TF5050CI HDMI	34010	
TF5050CI HDMI	34110	
TF5050CI(동유럽향)	5511	
TF5050CI(러시아향)	5521	TF-ES50CID
TF5050DVR-c	32406	
TF5050PDR	30406	
TF5100PVRc (FIN & RUS)	12406	TF-NCPCf, TF-NCPCr
TF5100PVRc HDMI	12416	TF-NCPCf
TF5100PVRc HDMI-N(북미향)(가칭)	12516	
TF5100PVRc HDMI-S(남미향)(가칭)	12316	
TF5100PVRc Masterpiece (FIN)	1486	TF-NNCPCf
TF5100PVRc Masterpiece (RUS)	2486	TF-NCPCr
TF5100PVRcE	12426	TF-NCPCf
TF5100PVRt (FIN)	13406	TF-NCPTf
TF5100PVRt HDMI (CZ)	13516	
TF5100PVRt HDMI (FIN)	13416	TF-NCPTf
TF5100PVRt Masterpiece (FIN)	1466	TF-NNCPTf
TF5110PVRc(불가리아향)	12506	TF-NCPCb
TF5200PVRc	10446	TF-NCPCd
TF5200PVRc(동유럽향)	10456	TF-NCPCe
TF5200PVRt-N	11406	
TF5300CI	541	
TF5300d	10406	
TF5300FTA	527	
TF5300k	10436	TF-NPTk
TF5400PVR combo (S&T)	1446	TF-NCNPstD
TF5410PVR HDMI (C&T)	2496	TF-NCNPct
TF550PVR	32430	TF-NCPCtct
TF5600PVR	1436	
TF5700PVRt (SWE)	2426	TF-NPTs
TF5700PVRt HDMI	13426	TF-NPTs
TF5700PVRt Masterpiece	2436	
TF5710PVRt HDMI	13436	TF-NPTs
TF5720PVRt HDMI	13536	TF-NPTs
TF5800PVR	456	TF-NPT
TF5800PVR Masterpiece	1476	
TF5800PVRt	458	TF-NPT
TF5810PVRt	457	TF-NPT
TF5900PVR	2476	
TF5950PVR(칠레향)	2477	
TF6000COC(러시아향)	40252	TF-CCMRC
TF6000COC(핀란드향)	40052	TF-CCMFC
TF6000COCI	50070	
TF6000COK	30652	
TF6000COT	30052	TF-CCMFT
TF6000COT-k	30252	
TF6000CR	50602	
TF6000F	20000	
TF6000F	20400	
TF6000F	50010	
TF6000F	50200	

TF6000F (ME)	50000	TF-MCMFI
TF6000F New OSD(가칭)	50600	
TF6000F(동유럽향)	50700	
TF6000Fe	20002	
TF6000Fe (GER)	50202	TF-ECMFG
TF6000Fe(독일향)	50002	TF-MCMFE
TF6000Fe(독일향)	50052	TF-ECMFE
TF6000Fe, TF6200CO	50502	TF-ECMFG
TF6000FT	20200	
TF6000FT	20300	
TF6000FT	20600	
TF6000IR, TF6200IR	16200	
TF6000IRC	26000	
TF6000PVR	2416	TF-NNPCW
TF6000PVR	2516	
TF6000PVR ES	2457	TF-NNPCW
TF6000PVR WS	2417	TF-NNPCW
TF6000PVRE	2456	TF-NNPCW
TF6000PVRE	2458	
TF6000PVRE	2556	
TF6000PVRt (AUS)	2446	TF-NNPCTA
TF6000SF	50051	
TF6000SFP	50451	
TF6000T (EUR)	30002	TF-CNFET
TF6000T(싱가폴향)	30102	
TF6000T(핀란드향)	30402	
TF6000TS	30552	TF-CMFETS
TF6000TS HDMI	30062	TF-CFCZT
TF6000VI	50702	
TF600PVRc	32416	TF-NNPCCF
TF600PVRt	2447	TF-NNPCTf
TF6010FT	20500	
TF6010FT	20700	
TF6010PVR	3416	TF-NNPCW
TF6010PVR WF	3456	
TF6010PVRE	3426	TF-NPCWB
TF6060CI	10066	
TF6060CI	21031	TF-MRCI
TF6060CI(러시아향) (RUS)	21231	
TF6100COC	40452	TF-CC7MEC
TF6100COC	40962	
TF6100COC(러시아향)	40652	TF-CC7MRC
TF6100COC(루마니아향)	40952	TF-C7RC
TF6100COC(슬로베니아향)	40852	
TF6100COC(우크라이나향)	40552	
TF6100COT	30452	
TF6100CR	50612	TF-ECNCRG
TF6100DCC	40753	TF-C7BDE
TF6100EMC	40752	
TF6100F	50400	
TF6100F(중동향)	50900	
TF6100GRC(ELoB향)	40553	TF-C7GRC
TF6100GRC(슬로베니아향)	40453	
TF6100IR	16100	
TF6100SF	50050	
TF6100SF(이집트향)	50150	
TF6200CO	50504	
TF6200COC(가칭)	40152	
TF6200COT	30152	
TF6200F	50100	

TF6200F	50500	
TF6200GR	50212	TF-GMFMK
TF6200GR(BBM)	50312	TF-EGMFMB
TF6200GR(ELOB)	50412	TF-GMFME
TF6200Gr(동유럽향)	50512	
TF6300IR	16300	TF-CFIR
TF6300IRc	26300	TF-CIRC
TF6310IR(태국향)	16310	
TF6400IR	16400	TF-GNIR
TF6400IRc	26500	
TF6400SF	50251	
TF6410IR(인도향)	16410	
TF6500F	20001	
TF6500F	20201	
TF6500F	50101	
TF6500F	50201	
TF6500F (ME), TF6400FTA	50001	TF-MCNFE
TF6500F New OSD(가칭)	50601	
TF6500F Plus(중동향)	50003	TF-MCNFE
TF6500IRc	26400	
TF6500T_HDMI	30063	
TF6700CO(브라질향)	50603	
TF6700CO(칠레향)	50503	
TF6700COC(브라질향)	40454	
TF6700IRt(아르헨티나향)	16700	
TF6700SF	50053	
TF6800F	50103	
TF6800F	50203	
TF6900Fe	50402	
TF7000HDPVRt	32040	
TF7000HT	31019	
TF700HSCI	23042	
TF7010HT	31020	TF-HFTA-VE
TF7050HDRt	33032	
TF7700HCCF (Cabletel)	43031	TF-HCCF
TF7700HCCF (DCC)	43032	TF-HCCF
TF7700HCCF (MSAT)	43030	TF-HCCF
TF7700HCCI (EU)	43022	
TF7700HCCI (GER)	43022	
TF7700HCCI(러시아향)	43122	
TF7700HDPVR (AT)	23531	
TF7700HDPVR (EUR), TF7710HDPVR	23031	
TF7700HDPVR (GER)	23231	
TF7700HDPVR, TF7710HDPVR (CZ)	23131	
TF7700HSCI	23122	
TF7700HSCI (POL)	23322	
TF7700HSCI (RUS)	23222	
TF7700HSCI(BOE)(가칭)	23025	
TF7700HSCI(독일향) (EUR)	23022	
TF7700HTCI	33022	
TF7700HTCO(ntv향)	33150	
TF7710HCCI	43023	
TF7710HDPVR	23431	
TF7710HDPVR (NL)	23331	
TF7710HSCI	23023	
TF7710HSCI (GER)	23123	
TF7710HTCI	33023	
TF7710HTCO	33050	
TF7720HSCI	23224	
TF7720HSCI	23324	

TF7720HSCI	23424	
TF7720HSCI (동유럽향)	23124	
TF7720HSCI(유럽향)	23024	
TF7720HSIR	23026	
TF8000	27000	
TF800PVR HDMI	439	
TMS (SR-2100)	22010	
Wildcard	39321	

## FRONTPANEL COMMUNICATION

The communication runs with the following scheme: <STX> **Command [Parameters]** <ETX>. The lower nibble of 'Command' is the number of parameters to follow. <STX> is the character 0x02, <ETX> is 0x03.

Yellow: FP => Mainboard

Cmd		Reply	Function	Sample	Comments
<b>Time</b>					
10	O	13	ReqFPTime	10	Request time from front panel clock
11	O	--	FPTimeFormat	11 81 11 80	Time format in standby (81=24h, 80=12h)
13	I		FPTime	13 0f 28 12	Answer to command 0x10. This sample shows the response for 15:40:18
30	O	25	ReqFPDateTimeNew	30	The command 0x25 is the response to this request.
25	I		FPDateTimeNew	25 0c 96 ae 28 01	This is the response to Cmd 30 and seems to have the same purpose as Cmd 15 except it uses the new data format (see also commands 74 & 84). See below for the date format.
30	I	15		30	The FP wants to know the time
15	O		FPDateTime	15 d2 57 0e 28 01	This command sends the current time to the front panel. The format is "15 DD DD HH MM SS" where DD is a MJD number. The following sample sets the front panel to 22.04.2006 14:40:01 local time.
<b>Display</b>					
35	O		SetLED	35 af 86 c6 a3 ff	This command sends the pattern for the front panel LED display. The bits are inverted therefore a 1 means that the segment is switched off. The sample to the left shows the string "rECo".
91	O	--	FPDisplayBufferFill	91 xx	MP VFD: Fills the internal 48 bytes buffer with xx. Only useful xx: 00 (display off) and ff (all segments)
99	O		MPVFDData	99 00 c0 0e 0c 78 31 61 c2 08 99 08 02 00 00 00 00 00 00 00 99 10 3c 00 00 00 00 00 00 00 99 18 00 00 00 04 20 f0 00 00 99 20 03 c1 40 fd 40 00 d0 eb 99 28 00 00 00 00 00 00 00 00	The 0x99 command writes into the 48 byte Masterpiece Display Buffer. The second byte contains the start address of the buffer and the next 8 bytes are written into the buffer (see the MP VFD PDF for details).
A2	O		MPBrightness	A2 08 xx	Sets the brightness of the MP's VFD. 0x20 is the maximum. 01, 02, 08, 10, 20
Cx	O		5500VFDData		5500 VFD

Timer					
72	O	--	FPWakeupClear	72 ff ff	This command seems to delete all wakeup timers in the front panel
74	O	--	SetFPWakeup	74 d2 58 06 00	This command seems to inform the front panel about the wakeup time. The format is "74 DD DD HH MM" where DD is a MJD number. The sample sets the wakeup timer to 23.04.2006 06:00 local time. The Topyy seems to send the next 3 entries to the FP (see also commands 15 & 25).
84	O	--	SetFPWakeupNew	84 0c 97 c6 00	The command 84 seems to have the same purpose as command 74 but uses a different format. See below for the date format.
Keys					
51	I		FPKey	51 06	This command is used to send front panel keystrokes to the main board (e.g. the Vol Up button).
51	O				
61	I		RemoteKey	61 1c	This command is used to send remote keystrokes to the main board (e.g. the OK button). Add 0x0080 for repeated key codes.
A0	O	A5	GetRemoteFilter	A0	Lets the FP generate 4 A5 messages with the current IR filter values.
A5	I	--	ReportRemoteFilter		See below for the meaning of the bytes
A5	O	--	SetRemoteFilter	A5 13 20 DF 0A 1B  Mode 1 = 02 34 0A Mode 2 = 49 00 0A Mode 3 = 49 99 0A Mode 4 = 20 DF 0A	Sets one of four filter for the remote control. The second byte contains the filter number (x0 to x3) and if the filter should be enabled (0x or 1x), the bytes 3 to 5 contain the remote code and byte 6 is a checksum (chk = ([1] + [2] + [3] + [4] - 1) & 0xff).
Power					
20	I		ShutdownReq	20	This command is repeatedly sent by the front panel in case somebody hit the power button.
31	O			31 00	Locks the FP buttons (incl. remote), VFD and shutdown event
31	O	--	PowerOff	31 01	Turns the Topyy off
31	O		ShutdownAck	31 02	Used by the firmware to acknowledge the ShutdownReq 0x20. If this isn't sent within a specific time, the FP processor assumes that the main processor is stuck and shuts the Topyy down. This command unlocks the FP buttons (opposed to 31 00) but disables the standby event. In this state, pressing standby will generate a 61 0a (remote) or 51 00 (FP).
31	O			31 03	???
31	O			31 04	Unlocks buttons, VFD and shutdown event
31	O			31 05	Unlocks buttons and shutdown event
31	O			31 06	Reboots the Topyy by toggling the power line for about 1 second.
21	O		Shutdown	21 01	This sequence turns off the Topyy.
40	O	--	GetDisplayControl	40	Stops the FP shutdown watchdog and returns the display control to the firmware.
41	O	--		41 44 41 88	Last Power Status (44=on, 88=off). This variable is responsible for the VF&F.
80	O	81	ReqBootReason	80	Ask the front panel why it has bootet up the Topyy.

<b>81</b>	I		BootReason	81 01	The response to the 0x80 command. Known responses: 00=front panel button, 01=remote power button or power failure, 02=timer
<b>Other</b>					
<b>B2</b>	O	B3	FPPeek	B2 F0 FF B2 F1 FF B2 F2 FF B2 F3 FF B2 F5 FF	Peeks a byte from the FP CPU memory. F0 to F3 is used as a FP-ID (MP=A6 00 00 00; TF5500=A5 56 46 44)
<b>B3</b>	I		FPPeekAnswer	B3 Fx FF xx	
<b>B3</b>	O	--	FPPoke	B3 F5 FF 01	Pokes a byte into the FP CPU
<b>Unknown</b>					
<b>54</b>	O	--		54 FF FF 00 00	
<b>E0</b>	O	E4			
<b>E4</b>	I			E4 0B 05 00 00	

The new date word is calculated as following:

```
{
    word    Date;
    byte    DayOfWeek:3;
    byte    Hour:5;
    byte    Minute;
    byte    Seconds;
}
```

Date = (((year - 2000) & 0x7f) << 9) | ((month & 0xf) << 5) | (day & 0x1f)

e.g. 0c 97 = 2006-04-23

## **HARDWARE**

### **MEMORY ADDRESS MAP**

Taken from a TF5000.

80000000-83FFFFFF: Cached RAM (64MB)  
84000000-87FFFFFF: RAM Shadow of 80000000 (64MB)  
88000000-8BFFFFFF: RAM Shadow of 80000000 (64MB)  
8C000000-8FFFFFFF: RAM Shadow of 80000000 (64MB)

9F000000-9FBFFFFFFF: unknown nearly constant pattern (12MB)

9FC00000-9FDFFFFFFF: Flash Shadow  
9FE00000-9FFFFFFF: Flash Shadow

A0000000-A3FFFFFF: Uncached RAM Shadow of 80000000 (64MB)  
A4000000-A7FFFFFF: RAM Shadow of 80000000 (64MB)  
A8000000-ABFFFFFF: RAM Shadow of 80000000 (64MB)  
AC000000-AFFFFFFF: RAM Shadow of 80000000 (64MB)

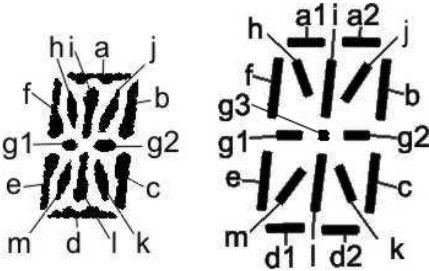
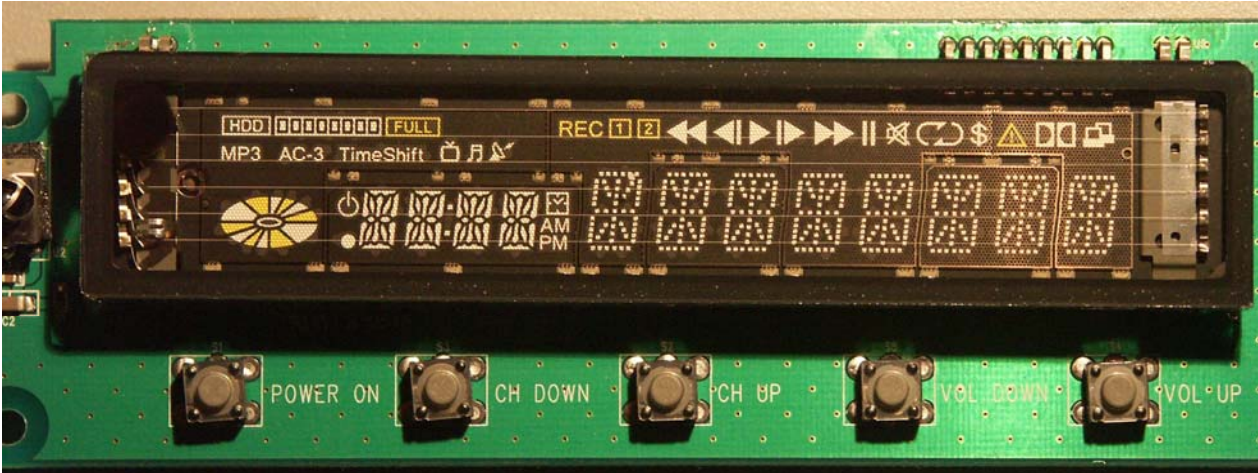
B0000000-BFBFFFFFFF: EMMA Hardware Register

BFC00000-BFC01FFF: Flash - Primary Loader (8kB)  
BFC02000-BFC0FFFF: Flash - Secondary Loader (56kB, compressed)  
BFC10000-BFDBFFFF: Flash - Firmware (1,7MB compressed)  
BFDC0000-BFDDFFFF: Flash - Sat/Timer... Tables (128kB compressed)  
BFDE0000-BFDEFFFF: Flash - Factory Settings (128kB compressed)

BFE00000-BFFFFFFF: Flash Shadow

**TF5000 MASTERPIECE VFD**

The display is a Samsung HCV-03SM06T.



The digits are counted from left.

	7	6	5	4	3	2	1	0
00								
01			Power	Remote Dot (?)	14 Seg / Digit 1 / d	14 Seg / Digit 1 / e	14 Seg / Digit 1 / c	14 Seg / Digit 1 / l
02	14 Seg / Digit 1 / m	14 Seg / Digit 1 / k	14 Seg / Digit 1 / g1	14 Seg / Digit 1 / g2	14 Seg / Digit 1 / b	14 Seg / Digit 1 / f	14 Seg / Digit 1 / j	14 Seg / Digit 1 / h
03	14 Seg / Digit 1 / i	14 Seg / Digit 1 / a	14 Seg / Digit 2 / d	14 Seg / Digit 2 / e	14 Seg / Digit 2 / c	14 Seg / Digit 2 / l	14 Seg / Digit 2 / m	14 Seg / Digit 2 / k
04	14 Seg / Digit 2 / g1	14 Seg / Digit 2 / g2	14 Seg / Digit 2 / b	14 Seg / Digit 2 / f	14 Seg / Digit 2 / j	14 Seg / Digit 2 / h	14 Seg / Digit 2 / i	14 Seg / Digit 2 / a
05	Clock Colon	14 Seg / Digit 3 / d	14 Seg / Digit 3 / e	14 Seg / Digit 3 / c	14 Seg / Digit 3 / l	14 Seg / Digit 3 / m	14 Seg / Digit 3 / k	14 Seg / Digit 3 / g1
06	14 Seg / Digit 3 / g2	14 Seg / Digit 3 / b	14 Seg / Digit 3 / f	14 Seg / Digit 3 / j	14 Seg / Digit 3 / h	14 Seg / Digit 3 / i	14 Seg / Digit 3 / a	PM
07	AM	Timer Symbol	14 Seg / Digit 4 / d	14 Seg / Digit 4 / e	14 Seg / Digit 4 / c	14 Seg / Digit 4 / l	14 Seg / Digit 4 / m	14 Seg / Digit 4 / k
08	14 Seg / Digit 4 / g1	14 Seg / Digit 4 / g2	14 Seg / Digit 4 / b	14 Seg / Digit 4 / f	14 Seg / Digit 4 / j	14 Seg / Digit 4 / h	14 Seg / Digit 4 / i	14 Seg / Digit 4 / a
17			17 Seg / Digit 2 / d1	17 Seg / Digit 3 / d1	17 Seg / Digit 2 / d2	17 Seg / Digit 3 / d2	17 Seg / Digit 2 / e	17 Seg / Digit 3 / e
18	17 Seg / Digit 2 / c	17 Seg / Digit 3 / c	17 Seg / Digit 2 / l	17 Seg / Digit 3 / l	17 Seg / Digit 2 / m	17 Seg / Digit 3 / m	17 Seg / Digit 2 / k	17 Seg / Digit 3 / k
19	17 Seg / Digit 2 / g1	17 Seg / Digit 3 / g1	17 Seg / Digit 2 / g2	17 Seg / Digit 3 / g2	17 Seg / Digit 2 / g3	17 Seg / Digit 3 / g3	17 Seg / Digit 2 / b	17 Seg / Digit 3 / b
20	17 Seg / Digit 2 / f	17 Seg / Digit 3 / f	17 Seg / Digit 2 / j	17 Seg / Digit 3 / j	17 Seg / Digit 2 / h	17 Seg / Digit 3 / h	17 Seg / Digit 2 / i	17 Seg / Digit 3 / i
21	17 Seg / Digit 2 / a1	17 Seg / Digit 3 / a1	17 Seg / Digit 2 / a2	17 Seg / Digit 3 / a2	17 Seg / Digit 6 / d1	17 Seg / Digit 7 / d1	17 Seg / Digit 6 / d2	17 Seg / Digit 7 / d2
22	17 Seg / Digit 6 / e	17 Seg / Digit 7 / e	17 Seg / Digit 6 / c	17 Seg / Digit 7 / c	17 Seg / Digit 6 / l	17 Seg / Digit 7 / l	17 Seg / Digit 6 / m	17 Seg / Digit 7 / m
23	17 Seg / Digit 6 / k	17 Seg / Digit 7 / k	17 Seg / Digit 6 / g1	17 Seg / Digit 7 / g1	17 Seg / Digit 6 / g2	17 Seg / Digit 7 / g2	17 Seg / Digit 6 / g3	17 Seg / Digit 7 / g3
24	17 Seg / Digit 6 / b	17 Seg / Digit 7 / b	17 Seg / Digit 6 / f	17 Seg / Digit 7 / f	17 Seg / Digit 6 / j	17 Seg / Digit 7 / j	17 Seg / Digit 6 / h	17 Seg / Digit 7 / h
25	17 Seg / Digit 6 / i	17 Seg / Digit 7 / i	17 Seg / Digit 6 / a1	17 Seg / Digit 7 / a1	17 Seg / Digit 6 / a2	17 Seg / Digit 7 / a2	CD Symbol - Center Circle	CD Symbol - 12h
26	CD Symbol - 1h	CD Symbol - 2h	CD Symbol - 3h	CD Symbol - 4h	CD Symbol - 5h	CD Symbol - 6h	CD Symbol - 7h	CD Symbol - 8h
27	CD Symbol - 9h	CD Symbol - 10h	CD Symbol - 11h	Sat Dish	Music Symbol	TV Symbol	TimeShift	AC-3
28	MP3	HDD FULL	HDD Fullness Frame	HDD Fullness 8	HDD Fullness 7	HDD Fullness 6	HDD Fullness 5	HDD Fullness 4
29	HDD Fullness 3	HDD Fullness 2	HDD Fullness 1	HDD Symbol				
33			17 Seg / Digit 1 / d1	17 Seg / Digit 8 / d1	17 Seg / Digit 1 / d2	17 Seg / Digit 8 / d2	17 Seg / Digit 1 / e	17 Seg / Digit 8 / e
34	17 Seg / Digit 1 / c	17 Seg / Digit 8 / c	17 Seg / Digit 1 / l	17 Seg / Digit 8 / l	17 Seg / Digit 1 / m	17 Seg / Digit 8 / m	17 Seg / Digit 1 / k	17 Seg / Digit 8 / k
35	17 Seg / Digit 1 / g1	17 Seg / Digit 8 / g1	17 Seg / Digit 1 / g2	17 Seg / Digit 8 / g2	17 Seg / Digit 1 / g3	17 Seg / Digit 8 / g3	17 Seg / Digit 1 / b	17 Seg / Digit 8 / b
36	17 Seg / Digit 1 / f	17 Seg / Digit 8 / f	17 Seg / Digit 1 / j	17 Seg / Digit 8 / j	17 Seg / Digit 1 / h	17 Seg / Digit 8 / h	17 Seg / Digit 1 / i	17 Seg / Digit 8 / i
37	17 Seg / Digit 1 / a1	17 Seg / Digit 8 / a1	17 Seg / Digit 1 / a2	17 Seg / Digit 8 / a2	17 Seg / Digit 4 / d1	17 Seg / Digit 5 / d1	17 Seg / Digit 4 / d2	17 Seg / Digit 5 / d2
38	17 Seg / Digit 4 / e	17 Seg / Digit 5 / e	17 Seg / Digit 4 / c	17 Seg / Digit 5 / c	17 Seg / Digit 4 / l	17 Seg / Digit 5 / l	17 Seg / Digit 4 / m	17 Seg / Digit 5 / m
39	17 Seg / Digit 4 / k	17 Seg / Digit 5 / k	17 Seg / Digit 4 / g1	17 Seg / Digit 5 / g1	17 Seg / Digit 4 / g2	17 Seg / Digit 5 / g2	17 Seg / Digit 4 / g3	17 Seg / Digit 5 / g3
40	17 Seg / Digit 4 / b	17 Seg / Digit 5 / b	17 Seg / Digit 4 / f	17 Seg / Digit 5 / f	17 Seg / Digit 4 / j	17 Seg / Digit 5 / j	17 Seg / Digit 4 / h	17 Seg / Digit 5 / h
41	17 Seg / Digit 4 / i	17 Seg / Digit 5 / i	17 Seg / Digit 4 / a1	17 Seg / Digit 5 / a1	17 Seg / Digit 4 / a2	17 Seg / Digit 5 / a2	REC	Tuner 1
42	Tuner 2	<<	<	>	>	>>		Mute
43	AutoRewind Left Circle	Auto Rewind Right Circle	\$	Attention	Dolby			Computer Symbol

## TOPFIELD TF-5XXX HARD DISK STRUCTURE

The following samples are based on a TF5000 80GB HDD. It has been formatted with 2068 sectors per cluster. The first cluster is called the System Cluster, but Topfield doesn't count it as a cluster. Therefore the first cluster (cluster 0) starts at sector 2068. The data uses the big endian (Motorola) format. In addition every 4 bytes block is mirrored. For example, the string „ABCDEFGHIJKL“ is saved as „DCBAHGFELKJI“.

Following the cluster structure:

Cluster#	Data
-	System Cluster
0	Root Directory
1	Recycle Directory
2	DataFiles Directory
3	ProgramFiles Directory
4	MP3 Directory
5...n	Data

The cluster size is based on the least common multiple of a disk sector (512) and a TS packet (188) = 24064. It is calculated the following way:

- Get the numbers of tracks, heads and sectors of the disk (depends on the OS)
- Calculate the disk size in GB (DiskSize = Tracks \* Heads \* Sectors / 1048576 / 2)
- Calculate the LCM factor from the disk size (LCMFactor = Ceil (Ceil (DiskSize / 47 \* 16) / 4) \* 4)
- The minimum LCM factor is 44 (If LCMFactor < 44 Then LCMFactor = 44)
- Now calculate the cluster size (SectorsPerCluster = 24064 \* LCMFactor / 512)

The last cluster number is calculated with the following formula: LastCluster = (DiskSize / SectorsPerClusters / 512) - 1.

### SYSTEM CLUSTER

The system cluster contains the super block, the root directory and the FAT. The size of the root directory is 254 sectors and the FAT can allocate up to 768 sectors. With 2068 sectors per cluster, a 130GB disk can be managed ((1023-256+1 FAT-Sectors) \* 512 BytesPerSector / 3 BytesPerCluster \* 2068 SectorsPerCluster \* 512 BytesPerSector).

Sector#	Sectors	Data
0	1	Super Block
1	1	Copy of the Super Block
2-255	254	Root Directory
256-698	443	FAT24-1
699-1023	325	FAT24-1 (not used by the Toppy: 80GB-border)
1024-1466	443	FAT24-2
1467-1791	325	FAT24-2 (not used by the Toppy: 80GB-border)
1792-2067	276	unused

The above sector numbers are always the same, regardless of the disk size. The root directory in sector 2 is an incomplete copy of the root directory in cluster 0.

## SUPER BLOCK (SECTOR 0)

```

00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
00000000h: 07 08 26 07 54 4F 50 46 49 45 4C 44 20 54 46 35 30 30 30 50 56 52 20 48 44 44 00 00 00 00 00 ; ..&.TOPFIELD TF5000PVR HDD.....
00000020h: 01 01 08 14 00 00 00 00 00 00 E8 7E 00 10 25 80 00 A3 DA CF 00 00 00 00 00 00 00 00 00 00 00 ; .....à~..%€.fúí.....
00000040h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
00000060h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
00000080h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
000000a0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
000000c0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
000000e0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
00000100h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
00000120h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
00000140h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
00000160h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
00000180h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
000001a0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
000001c0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
000001e0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....

```

```

0x0000 Magic number      0x07082607
0x0004 HD ID 5000:      TOPFIELD TF5000PVR HDD
0x0004 HD ID 4000:      TOPFIELD PVR HDD
0x0020 Layout Version:  V1.1
0x0022 Sectors per Cluster: 2068
0x0024 Root Start Cluster
0x0028 Nr. of used Clusters
0x002C Empty Bytes in Root
0x0030 FAT_CRC32

```

The FAT\_CRC32 is based on the same polynomial as WinZIP. The only difference is that Topfield doesn't use a reflection and the last XOR is not done. The checksum is calculated over the FAT, rounded to the next sector boundary. The following formula used:

```
FatSize:=512 * Ceil (3 * LastCluster / 512);
```

```

Bit size      = 32
Polynomial    = 0x04C11DB7
Start         = 0xFFFFFFFF
Data Reflection = no
Output Reflection = no
XOR Out      = 0x00000000

```





## DIRECTORY DATAFILES (CLUSTER 2)

Following are some sample recordings. The directories also use the famous „,“ and „,“ directories.

```

00000000h: F1 00 00 00 00 00 00 00 00 00 00 02 00 00 00 01 00 10 1C 80 2E 00 00 00 00 00 00 00 00 00 00 00 ; ñ.....€.....
00000020h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
00000040h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
00000060h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....i...

00000080h: F0 FF FF FF FF FF FF FF 00 00 00 00 00 00 00 01 00 10 26 80 2E 2E 00 4F 4F 54 5F 5F 00 5F 5F 00 ; ð.....&€...OOT...
00000a0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
00000c0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
00000e0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....{...

00000100h: D1 CF 99 15 34 00 00 00 00 00 00 05 00 00 12 CA 00 0B C0 00 46 61 63 65 2F 4F 66 66 20 2D 20 49 ; Ñï™.4.....Ê..Ä..Face/Off - I
00000120h: 6D 20 4B F6 72 70 65 72 20 64 65 73 20 46 65 69 6E 64 65 73 2D 31 2E 72 65 63 00 00 00 00 00 ; m Körper des Feindes-1.red.....
00000140h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 50 72 6F 53 69 65 62 65 6E 00 00 00 ; ProSieben.....
00000160h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 04 03 00 00 50 00 00 00 ; .....P...

```

e.g.: Face/Off, 20.05.2004 21:52, Start Cluster 5, Size: 4810 Clusters = 4.74GB ((0x12CA\*2068-0x0BC0/2-1)\*512+512), partly scrambled (yellow/green key)

<p><b>File Attrib:</b></p> <p>D0 ... File (created by Altair)</p> <p>D1 ... File (created by the Topy)</p> <p>F0 ... ‘.’ Directory</p> <p>F1 ... ‘.’ Directory</p> <p>F2 ... Subdirectory</p> <p>F3 ... Recycle-directory</p> <p>FF ... unused</p> <p><b>Flags:</b></p> <p>0000 ... not scrambled</p> <p>0400 ... descrambled (by a copy) (green key)</p> <p>0401 ... scrambled (yellow key)</p> <p>0402 ... descrambled (green key)</p> <p>0403 ... partly scrambled (yellow/green key)</p> <p>0200 ... File is locked</p>	<p><b>File Date/Time:</b> DD DD HH MM SS OO OO (DDDD=MJD=days since 17.11.1858)</p> <p>OOOO = local offset (0000 if set manually)</p> <p><b>Start-Cluster</b></p> <p><b>File Size:</b></p> <p>0x0C ... Number of Clusters (let’s call the field S1)</p> <p>0x10 ... Number of unused sectors in the last cluster in bytes (S2)</p> <p>0x7E ... Number of bytes in the last sector (mod 0x200) (S3)</p> <p><b>S3_CRC:</b></p> <p>0x7D ... = ((S3 &gt;&gt; 8) + (S3 &amp; 0xff) + 1) &amp; 0xff</p> <p>Size = S1*SectorsPerCluster * 512 - S2 - 512 + S3</p> <p>(add 512 instead of S3 if S3_CRC == false)</p> <p><b>File name:</b> terminated by a 0x00</p> <p><b>Service name:</b> terminated by a 0x00</p> <p><b>WinFileAttribute:</b> Contains the Windows FILE_ATTRIBUTE flags if a file has been copied from the PC (little endian, not used by the firmware).</p> <p><b>Unknown fields:</b></p> <p>0x73 ... __temprec__.* has a 0x04, all other a 0x00</p> <p>0x74 ... .taps and .mp3 have a 0x20, all other a 0x00</p> <p>0x7C ... seems to be a checksum (starting from Layout v1.1)</p>
---	---

The S1 field of the „,“-directory contains the size of the current directory and therefore is always 1. The field S2 holds the number of free bytes in that current directory and can be calculated by:  $S2 = 512 * \text{SectorsPerCluster} - (128 * \text{UsedSlots})$ . A slot is a single directory entry. In other words:  $\text{NextSlot} = (512 * \text{SectorsPerCluster} - S2) / 128$

The S1 field of the „,“-directory contains the same value as the S2-field of the parent directory while it as created.

The S1-field of a subdirectory contains the sum of all used clusters plus the size of the subdirectory itself; S2=0; S3=0

# TOPFIELD REC HEADER

## TF5000

```

00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f
00000000h: 54 46 72 63 50 00 00 00 00 01 00 01 00 00 01 00 12 34 32 CA 03 EA 01 F4 01 F4 81 F5 4F 52 46 32 ; TFrcP.....DRF2
00000020h: 00 69 73 63 68 65 73 20 46 53 00 00 00 00 00 00 00 00 00 00 01 80 00 00 00 00 31 94 55 F0 04 5D ; isches FS.....
00000040h: 80 00 00 01 80 00 01 1F 00 00 16 82 D2 9A 12 0F D2 9A 13 2E 04 08 00 05 48 65 6C 70 20 54 56 05 ; .....Help TV.
00000060h: 48 65 6C 70 20 54 56 00 20 4B 75 6C 74 75 72 40 6E 65 72 00 00 00 00 05 41 6C 70 65 6E 77 65 74 ; Help TV. Kultur@ner....Alpenwet
00000080h: 74 65 72 00 6C 00 00 00 00 00 00 00 00 00 00 00 40 00 00 00 00 00 05 41 52 54 45 20 4B 75 6C ; ter.l.....ARTE Kul
000000a0h: 74 75 72 00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 00 00 00 01 05 44 61 73 20 47 65 73 74 ; tur.....Das Gest
000000c0h: FC 74 05 31 34 2E 20 45 69 6E 20 70 61 61 72 20 5A 65 6E 74 69 6D 65 74 65 72 00 00 00 00 00 00 ; ..... Ein paar Zentimeter.....
000000e0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 00 00 00 01 05 49 63 68 2C 20 48 75 62 ; .....Ich, Hub
00000100h: 65 72 74 75 73 05 4B 75 72 7A 66 69 6C 6D 20 44 65 75 74 73 63 68 6C 61 6E 64 20 32 30 30 36 00 ; ertus.Kurzfilm Deutschland 2006.
00000120h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 00 00 00 01 05 5A 61 70 70 69 6E 67 20 ; .....Zapping
00000140h: 49 6E 74 65 72 6E 61 74 69 6F 6E 61 6C 05 44 61 73 20 69 73 72 61 65 6C 32 CA 00 00 00 00 00 00 ; International.Das israel.....
00000160h: 00 00 00 00 02 00 00 00 32 CA 01 6D 00 00 16 82 05 2D 20 53 65 78 65 6E 74 7A 75 67 20 66 FC 72 ; .....- Sexenzug für
00000180h: 20 52 61 73 65 72 20 52 6F 6C 61 6E 64 20 41 73 73 69 6E 67 65 72 2C 20 42 72 75 64 65 72 20 64 ; Raser Roland Assinger, Bruder d
000001a0h: 65 73 20 4D 69 6C 6C 69 6F 6E 65 6E 73 68 6F 77 2D 4D 6F 64 65 72 61 74 6F 72 73 2C 20 73 74 65 ; es Millionenshow-Moderators, ste
000001c0h: 6C 6C 74 20 73 65 69 6E 20 56 65 72 6B 65 68 72 73 70 72 6F 6A 65 6B 74 20 76 6F 72 20 2D 20 44 ; llt sein Verkehrsprojekt vor - D
000001e0h: 65 72 20 4B 61 6D 70 66 20 67 65 67 65 6E 20 64 61 73 20 4B 75 63 6B 75 63 6B 73 6B 69 6E 64 20 ; er Kampf gegen das Kuckuckskind
...
00000520h: 6D 61 73 20 57 FC 70 70 65 72 20 28 52 69 63 68 61 72 64 20 47 72 69 65 73 65 29 2C 20 55 64 6F ; mas Wüpper (Richard Griese), Udo
00000540h: 20 53 63 68 65 6E 6B 20 28 45 77 61 6C 64 20 48 6F 72 74 65 6E 29 20 75 2E 61 2E 20 52 65 67 69 ; Schenk (Ewald Horten) u.a. Regi
00000560h: 65 3A 20 4D 69 63 68 61 65 6C 20 4D 61 63 6B 00 00 00 00 00 02 00 00 00 00 00 00 00 52 00 00 00 B6 ; e: Michael Mack.....
00000580h: 00 00 01 32 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
...

```

## TF5800

```

00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f
00000000h: 54 46 72 63 50 10 00 00 00 00 00 01 00 00 00 00 06 30 10 84 10 84 02 62 02 62 82 64 42 42 43 20 ; TFrcP.....0...b.b,dBBC
00000020h: 54 57 4F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 07 B8 90 ; Two
00000040h: 10 04 00 00 23 3A 30 05 80 00 02 0A 00 00 A1 AE D2 9A 10 32 D2 9A 13 00 04 0E 00 57 69 6D 62 6C ; ...#:0.e.....;@š.2š....Wimbl
00000060h: 65 64 6F 6E 20 32 30 30 36 53 70 6F 72 74 00 72 72 65 6E 74 20 61 66 00 00 00 00 54 6F 64 61 79 ; edon 2006Sport.rrent af...Today
00000080h: 20 61 74 20 57 69 6D 62 6C 65 64 6F 6E 53 70 6F 72 74 00 65 6E 20 22 00 00 00 00 54 68 65 20 43 ; at WimbledonSport.en "...The C
000000a0h: 6F 6E 76 65 6E 74 44 6F 63 75 6D 65 6E 74 61 72 79 00 74 60 42 75 69 00 00 00 00 52 6F 6F 6D 20 ; onventDocumentary.t`Bui...Room
000000c0h: 31 30 31 45 6E 74 65 72 74 61 69 6E 6D 65 6E 74 00 6E 20 62 54 68 65 00 00 00 01 4E 65 77 73 6E ; 10lEntertainment.n bThe...Newsn
000000e0h: 69 67 68 74 4E 65 77 73 20 61 6E 64 20 43 75 72 72 65 6E 74 20 41 66 66 61 69 72 73 00 4D 5A 4F ; ightNews and Current Affairs.MZO
00000100h: 4F 4B 69 43 68 69 6C 64 72 65 6E 00 69 6C 64 72 65 6E 20 62 54 68 65 00 00 00 02 57 68 65 6E 20 ; OKiChildren.ildren bThe...When
00000120h: 54 6F 62 79 20 4D 65 74 20 4A 75 6C 69 65 3A 20 54 68 65 20 53 74 6F 72 79 20 6F 66 20 74 68 65 ; Toby Met Julie: The Story of the
00000140h: 20 4D 6F 64 65 72 6E 20 52 65 76 69 65 77 44 6F 63 75 6D 65 6E 74 61 72 79 20 22 42 10 84 00 00 ; Modern ReviewDocumentary "B...
00000160h: 00 00 05 63 05 77 00 00 02 00 00 10 84 01 1E 00 00 A1 AE 54 68 65 20 61 63 74 69 6F 6E 20 63 ; ...c.w.....;@The action c
00000180h: 6F 6E 74 69 6E 75 65 73 20 66 72 6F 6D 20 74 68 65 20 41 6C 6C 20 45 6E 67 6C 61 6E 64 20 43 6C ; ontinues from the All England Cl
000001a0h: 75 62 2C 20 77 69 74 68 20 74 68 65 20 62 65 73 74 20 6F 66 20 74 6F 64 61 79 27 73 20 6D 61 74 ; ub, with the best of today's mat
...
00000540h: 74 61 63 6B 6C 65 20 74 68 65 20 70 61 69 6E 20 6F 66 20 68 65 72 20 63 68 69 6C 64 68 6F 6F 64 ; tackle the pain of her childhood
00000560h: 2E 20 5B 53 5D 20 5B 57 5D 00 68 65 60 62 65 73 00 00 00 00 00 00 00 00 00 00 00 00 00 26 ; . [S] [W].he`bes.....&
00000580h: 00 00 00 4A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ...J.....
...
00000660h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
00000680h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
...
00000e60h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
00000e80h: 01 DD 0A 9B C0 A9 35 1C 09 4E 20 CD 7C 0D 81 D6 83 49 9C 0A 21 3F 81 55 E0 09 06 74 FE 2D 5C A0 ; .Ý.>À@5..N í|. ÖfIe.!? Uà..tp-\

```

<pre>typedef struct {   dword MagicNumber; //0x54467263 (TFrc)   word Version; //0x5000, 0x5010   byte reserved1 [2];   word Duration; //in minutes   word ServiceNr;   word ServiceType; //0=TV, 1=Radio } <b>tRECHheader</b>; //14 bytes</pre>	<pre>typedef struct {   byte SatIndex; //DVB-s only   byte reserved1;   word TPIIdx:10;   word TunerNum:2;   word DelFlag:1;   word CASFlag:1;   word LockFlag:1;   word SkipFlag:1;   word ServiceID;   word PMTPID;   word PCRPID;   word VideoPID;   word AudioPID; //Note 1   char ServiceName [24]; } <b>tRECSERVICEInfo5000</b>; //38 bytes</pre>	<pre>typedef struct {   byte SatIndex; //DVB-s only   byte reserved1;   word TPIIdx:10;   word TunerNum:2;   word DelFlag:1;   word CASFlag:1;   word LockFlag:1;   word SkipFlag:1;   word ServiceID;   word PMTPID;   word PCRPID;   word VideoPID;   word AudioPID; //Note 1   char ServiceName [28]; } <b>tRECSERVICEInfo5010</b>; //42 bytes</pre>
<pre>typedef struct {   byte SatIndex;    byte Polarization:1;   byte Mode:3; //0=Normal, 1=SmaTV   byte reserved1:4;    byte reserved2 [2];   dword Frequency; //in MHz   word SymbolRate;   word TSID;   byte reserved3 [2];   word NetworkID; } <b>tRECTPIInfoSat</b>; //16 bytes</pre>	<pre>typedef struct {   word ChannelNumber; //Note 2   byte Bandwidth; //in MHz   byte reserved1;   dword Frequency; //in kHz   word TSID;   byte LPHPSstream; //0=LP, 1=HP   byte reserved2;   word NetworkID;   byte <b>unknown1</b> [2]; } <b>tRECTPIInfoTer</b>; //16 bytes //Note 3</pre>	<pre>typedef struct {   dword Frequency;   word SymbolRate;   word TSID;   word NetworkID;   byte Modulation; //0=16QAM, 1=32QAM,...   byte unused1; } <b>tRECTPIInfoCable</b>; //12 bytes</pre>

Note 1: 0x8xxx = Auto, 0x2xxx = AC3 channel, 0x0xxx = fixed language

Note 2: In Australian F/W versions, the ChannelNumber is coded in BCD while a decimal number is used in European versions.

Note 3: The transponder block on a TF5700PVRt (HDMI) is 8 bytes longer than on other machines. The meaning of the extra data is not yet known.

<pre>typedef struct { byte reserved1 [2]; byte DurationHour; byte DurationMin; dword EventID; word StartTimeMJD; byte StartTimeHour; byte StartTimeMin; word EndTimeMJD; byte EndTimeHour; byte EndTimeMin; byte reserved2; byte TextLength; byte ParentalRate; char EventName [TextLength]; //no 0x00 terminator! char EventDescription [257 - TextLength]; //Note 4 char unknown1 [18]; } tRECEventInfo; //294 bytes</pre>	<pre>typedef struct { word TextLength; dword EventID; char Text [1024]; } tRECExtendedEventInfo; //1030 bytes</pre>	<pre>typedef struct { byte reserved1 [4]; byte CryptFlag; //Note 5 byte reserved2 [3]; } tRECCryptInfo; //8 bytes</pre>
<pre>typedef struct { dword Bookmark [64]; //Note 6 } tRECBookmarks; //256 bytes</pre>	<pre>dword Resume; byte reserved [2048];</pre>	

Note 4: The EventDescription buffer might contain other information

Note 5: Crypt Flag: 00=FTA, 01=scrambled, 02=descrambled, 03=partly scrambled

Note 6: Bookmarks are saved in 512 x (Size of a TS Packet) units (=94kB). This makes them independent of the cluster size.

A note from R2-D2:

The .rec header is different on the TF5810. There are only 48 bookmarks, and so the later sections are moved down by 64 bytes. This is the same on the TF500/TF600, which share a lot of things in common with our firmwares and the TF5810 in particular. The "reserved[2048]" section is actually a collection of up to 512 block indexes (like bookmarks) that specify the places where a recording was paused ("cut points") -- the firmware uses these to cover up the glitch at the cut point, but it's very badly programmed. Now, the Resume bookmark seems to be a feature that's only in our TF5800 and TF5810 firmwares (plus almost all TF500 and TF600 firmwares) -- if you've got it then there is an option to press Play (rather than OK) on a recording in the archive to "resume" it from the last played position. On all other firmwares this is the count of the "cut points". But it is actually that on our firmwares, too... So the clever covering up of the glitch points would only be seen on our firmwares the first time the file is played. DOH! I can't see any way to judge whether there are 64 or 48 bookmarks in any given .rec header without knowing the model that made it. :(

## HOW TO AUTO DETECT THE REC HEADER TYPE

Use the following rules for the type decision: count 1 point for every matching rule

for dvb-s:

RECTPInfoSat.Polarization & 0x6F == 0 ?  
RECTPInfoSat.SymbolRate in [2000...30000] ?  
RECTPInfoSat.Frequency in [10700...12800] ?

for dvb-t

RECTPInfoTer.Bandwidth in [6..8] ?  
RECTPInfoTer.Frequency in [174000...230000] || [470000...862000] ?  
RECTPInfoTer.LPHPStream & 0xFE == 0 ?

for dvb-c

RECTPInfoCable.Frequency in [47000...862000] ?  
RECTPInfoCable.SymbolRate in [2000...30000] ?  
RECTPInfoCable.Modulation <= 4 ?

If one system gets 3 points and all other less than 3, you've found the headers system type.