TOPPY KEY HANDLING

(Notably for TF5800)

1 INTRODUCTION

The descriptions below are intended to indicate the main data flows as they need to be understood by TAP writers, and are not intended as definitive statements of Topfield's firmware design.

The very early TF5800 firmware used a different system which is not discussed here.

2 STANDARD FIRMWARE PROCESSES

Fig 1 shows the "official" keypress handling within the Toppy. The main stages are:

(1) The input from the remote control unit & front panel are received by the firmware as number in the range 0x10000 to 0x100FF (not all of which are used, course). If the number is greater than 0x1007F then subtract 0x1007F from the value (effectively making 0x10080 to 0x100FF a direct repeat of 0x10000 to 0x1007F). This will actually be done with a bitwise operation of course - probably something silly like they only wired up 7 bits of the connection somewhere. This is the cause of the discrete Power Off code as it is done in firmware so cannot turn the Toppy on.

N.B. The actual key labels differ between models, with some functions moving and others combined, changed, or missing according to model. At this stage there is the same relationship between key code and key position for all models, and not between functions and key codes.

(2) The input key data is mapped onto the values that the TAPs see, still in the range 0x10000 to 0x10050. This mapping is not simple addition of a constant. These are the values that are allocated to the RKEY_xx series of constants in the KEY.h file. The names allocated are based on the Non-TF5800 key labels, so about 30% are a little or completely misleading for the TF 5800.

(3) For the TF5800 only, further mapping is performed in which some keys are combined to form a single code, and others are swapped round. This is done to ensure that basic functions such as V+/-, Tv/Radio & Exit appear to be the same to the TAPs, even though the physical keys are different. However this mapping does not apply to all functions that have moved, such as Text/Teletext.

(4) The results are passed as param1 to the first TAP, together with zero in param2. The TAP returns a key value, normally the param1 value or zero – see next item.

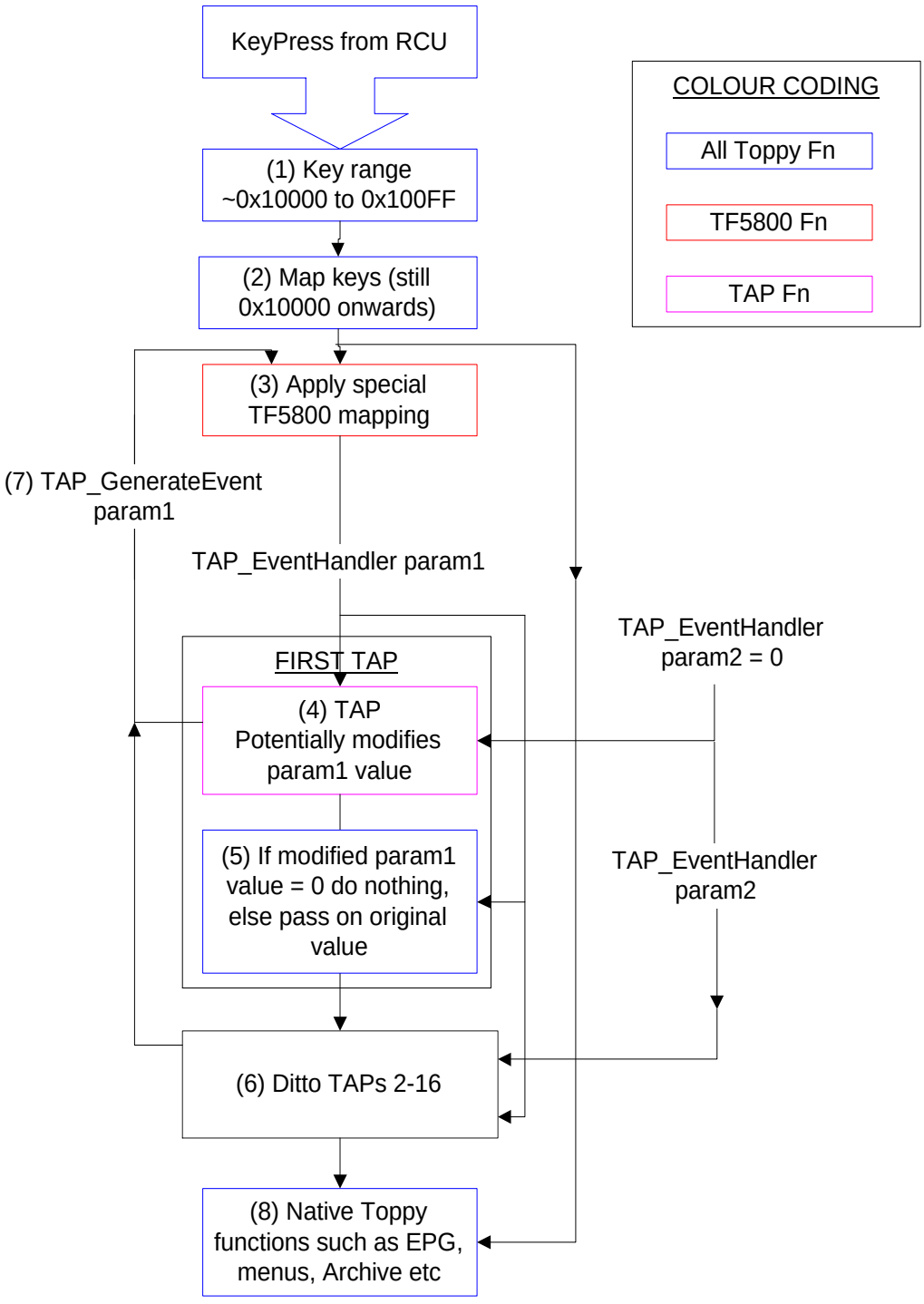
(5) If the return value is zero, the keypress goes no further. If a non-zero value is passed, the original param1 value (NOT the modified value) is passed to the next TAP, or to the native Toppy functions.

N.B. There is one exception – if TAP_ExitNormal() has not been sent or if it has been cancelled with TAP_EnterNormal(), the "<<" ">>" keypresses will seen by the firmware even if zero has been returned.

(6) Processes 4 & 5 are repeated for up to 15 more TAPs.

(7) TAPs may simulate a keypress using the “TAP_GenerateEvent” function. This goes to the input of the special TF5800 mapping function, so the TAP has to invert that mapping to simulate the mapped keys. Param2 in “TAP_GenerateEvent” is ignored.

(8) The native Toppy functions such as EPG etc also have access to the unmapped keys, so they can distinguish between the combined keys.



3 ADDITIONAL FEATURES ADDED BY FIRMWARE “HACKS”

Three additional features are available via hacks to the firmware implemented by TAPs as detailed below. All are specifically aimed at bypassing the special TF5800 mapping with it key combining, and thus allows all the keys to be used by TAPs.

(9) The “Remote Extender” TAP takes the original keypresses, does some mapping, and inserts them into param2 of TAP_EventHandler in place of the standard zero.

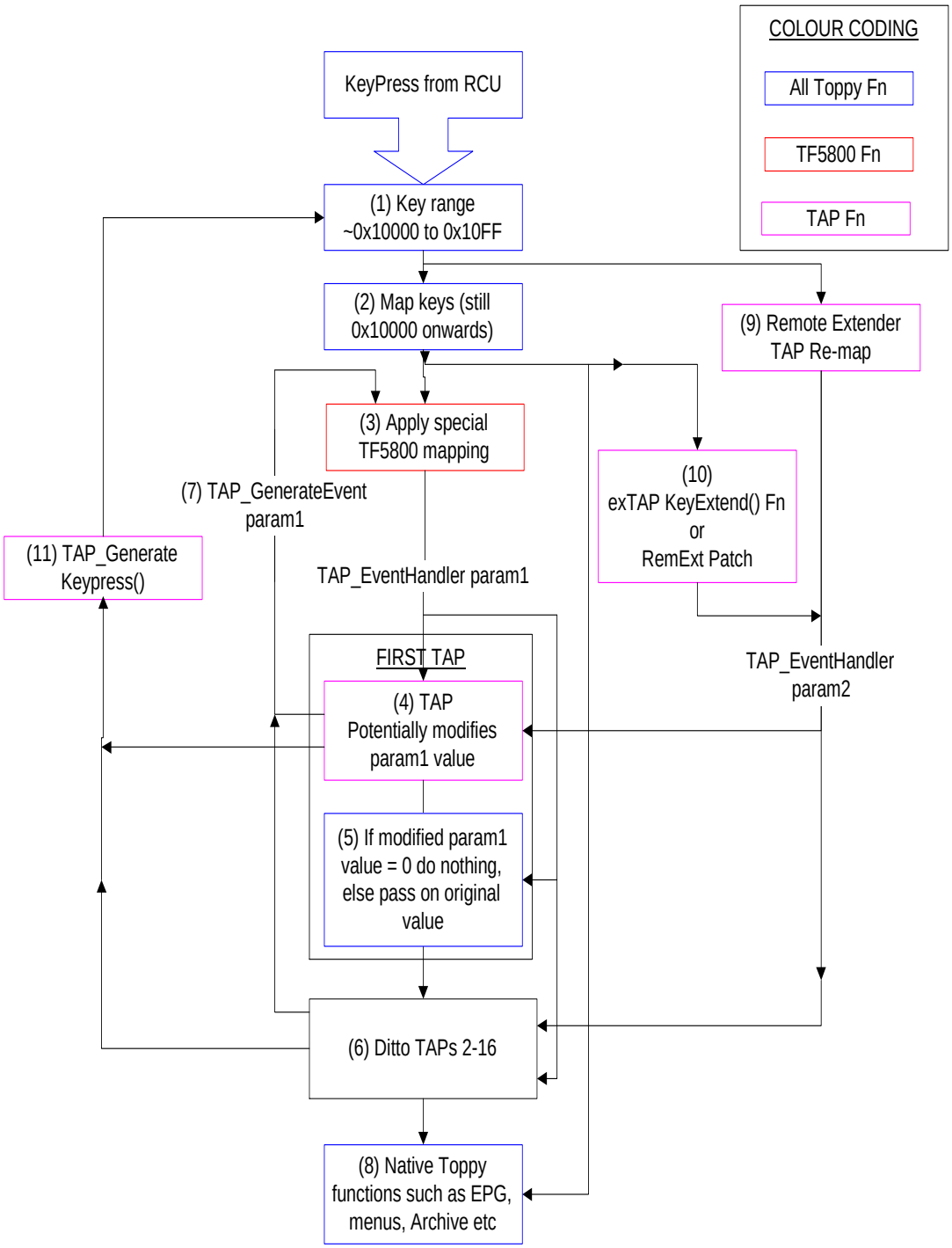
(10) The exTAP_KeyExtend() function and the RemExt patch each puts the value prior to the special UK mapping into param2 of TAP_EventHandler.

(11) The TAP_GenerateKeypress() function simulates a keypress prior to the sample points of (9) & (10), and thus allows them to be used to distinguish between the combined keys. This can be called by any TAP wishing to simulate key presses.

As can be seen from the above, Remote Extender (which is a TAP in its own right) and exTAPKeyExtend() (which can be called up by any other TAP), and the RemExt patch all write into the same data item. As exTAPKeyExtend() and the RemExt patch write the same data there is no conflict. Remote Extender will overwrite the data provided by either of the others, so its data will always take precedence when it is running.

4 COMMAND KEY CODES

Key codes 0x10027 to 0x10036 are used by Topfield for special test purposes, and can do very nasty things to your Toppy. DO NOT USE THEM (at least one is reputed to overwrite the EEPROM including the loader, so cannot be reversed by software means)!


COLOUR CODING
All Toppy Fn
TF5800 Fn
TAP Fn
KeyPress from RCU
(1) Key range ~0x10000 to 0x10FF
(2) Map keys (still 0x10000 onwards)
(9) Remote Extender TAP Re-map
(3) Apply special TF5800 mapping
(7) TAP_GenerateEvent param1
(10) exTAP KeyExtend() Fn or RemExt Patch
(11) TAP_Generate Keypress()
TAP_EventHandler param1
TAP_EventHandler param2
FIRST TAP
(4) TAP Potentially modifies param1 value
(5) If modified param1 value = 0 do nothing, else pass on original value
(6) Ditto TAPs 2-16
(8) Native Toppy functions such as EPG, menus, Archive etc

5 KEY CODES ETC

The table below relates the various key codes & remote control labels for the different models. It is laid out roughly from top to bottom on the Remote Control Unit using the numbering system in the TF5800 manual. As stated above, the two types of RCU have the same physical keys, but different legends, and one row provides the data for one key position on the two model types. The contents of the columns are as follows:

Column 1: The reference number allocated to the key in the TF5800 manual. Where one reference number in the manual refers to a group of keys, they have been subdivided (e.g. the numeric keys are grouped as No 6 in the manual, and are numbered 6.0 to 6.9 in the table).

Column 2: The key caption on the TF5800 remote.

Column 3: The reference number allocated to the key in the TF5000 manual. Where one reference number in the manual refers to a group of keys, they have been subdivided (e.g. the numeric keys are No 6 in the manual, and are numbered 6.0 to 6.9 in the table).

Column 4: The key caption on the TF5000 remote.

Column 5: The RKEY constant equivalent to the param1 value seen by a TAP when that key is pressed on the TF5800 remote.

Column 6: The RKEY constant equivalent to the param1 value seen by a TAP when that key is pressed on the TF5000 remote. This is also the value returned by either the exTAP KeyExtend() function or the RemExt patch in param2 on the TF5800.

Columns 7 & 8: give the hex equivalents of columns 5 & 6.

Column 9: The value of param2 returned by Remote Extender.

Column 11: The RKEY constant to be used to generate the relevant simulated keypress. As one would expect from the logic description above, this is always the same as column 6.

The values to be used in TAP_GenerateKeypress() are different again, but can always be obtained from column 10 with the logic “(Col 10 value & 0xFF) | 0x10000”.

6 KEY LAYOUTS

See http://members.optusnet.com.au/toppytools/gallery.html for UK & Australian remote control unit layouts.

7 ACKNOWLEDGEMENTS

To various contributors to the Toppy.org.uk forums where I have collected most of my information, but particularly to “simonc”.

8 VERSION HISTORY

23/3/2007: Initial “public” version.

8/6/2007: Update to reflect input from http://forum.toppy.org.uk/forum/viewtopic.php?t=8081.

11/9/2007: Update to reflect feedback from R2-D2 and include reference to “RemExt” patch.

EMJB

KEY LABELS & PARAMETER VALUES

Col 1	Col 2	Col 3	Col 4	Col 5	Col 6	Col 7	Col 8	Col 9	Col 10	Col 11
Posn in TF5800 Manual	TF5800 Key Name	Posn in TF5000 Manual	TF5000 Key Name	Key H for UK	Key H Other	TF5800 param1 Hex	TF5000 param1 ExTap K.E. param2 RemExt Patch param2 (Hex)	Remote Extender param2 Hex	TF5800 TAP_Gen_Evt Code	Notes
1	Power	1	Power	RKEY_Power	RKEY_Power	1000A	1000A	1F0	RKEY_Power	
2	Mute	2	Mute	RKEY_Mute	RKEY_Mute	1000C	1000C	10C	RKEY_Mute	
3	TV/Radio	3	UHF	RKEY_TvRadio	RKEY_Uhf	10018	1000E	143	RKEY_Uhf	
4	TvSat	4	TV/STB	RKEY_TvSat	RKEY_TvSat	10022	10022	108	RKEY_TvSat	
5	Opt	5	Sleep	RKEY_Sleep	RKEY_Sleep	10010	10010	144	RKEY_Sleep	
6.0	0	6.0	0	RKEY_0	RKEY_0	10000	10000	110	RKEY_0	
6.1	1	6.1	1	RKEY_1	RKEY_1	10001	10001	111	RKEY_1	
6.2	2	6.2	2	RKEY_2	RKEY_2	10002	10002	112	RKEY_2	
6.3	3	6.3	3	RKEY_3	RKEY_3	10003	10003	113	RKEY_3	
6.4	4	6.4	4	RKEY_4	RKEY_4	10004	10004	114	RKEY_4	
6.5	5	6.5	5	RKEY_5	RKEY_5	10005	10005	115	RKEY_5	
6.6	6	6.6	6	RKEY_6	RKEY_6	10006	10006	116	RKEY_6	
6.7	7	6.7	7	RKEY_7	RKEY_7	10007	10007	117	RKEY_7	
6.8	8	6.8	8	RKEY_8	RKEY_8	10008	10008	118	RKEY_8	
6.9	9	6.9	9	RKEY_9	RKEY_9	10009	10009	119	RKEY_9	
7	Recall	7	Recall	RKEY_Recall	RKEY_Recall	1000B	1000B	11E	RKEY_Recall	
8	Info	8	Info	RKEY_Info	RKEY_Info	10014	10014	11D	RKEY_Info	
9	Guide	9	Guide	RKEY_Guide	RKEY_Guide	10016	10016	11B	RKEY_Guide	
10	Menu	11	Menu	RKEY_Menu	RKEY_Menu	1001C	1001C	11A	RKEY_Menu	
11	List	10	TV/Radio	RKEY_TvRadio	RKEY_TvRadio	10018	10018	104	RKEY_TvRadio	
12	Exit	12	Sound	RKEY_Exit	RKEY_AudioTrk	10017	10013	105	RKEY_AudioTrk	
13.1	V+	16	Fav	RKEY_VolUp	RKEY_Fav	10015	10020	109	RKEY_Fav	
13.2	V-	18	Exit	RKEY_VolDown	RKEY_Exit	1000F	10017	11C	RKEY_Exit	
14.1	P+	17	Subtitle	RKEY_ChUp	RKEY_Subt	10012	10021	107	RKEY_Subt	
14.2	P-	19	Teletext	RKEY_ChDown	RKEY_Teletext	1001D	10037	147	RKEY_Teletext	
15	OK/List	15	OK	RKEY_Ok	RKEY_Ok	1001E	1001E	11F	RKEY_Ok	
16.1	Up	14.1	P+	RKEY_ChUp	RKEY_ChUp	10012	10012	100	RKEY_ChUp	
16.2	Down	14.2	P-	RKEY_ChDown	RKEY_ChDown	1001D	1001D	101	RKEY_ChDown	
17.1	Left	13.1	V-	RKEY_VolDown	RKEY_VolDown	1000F	1000F	103	RKEY_VolDown	

Col 1	Col 2	Col 3	Col 4	Col 5	Col 6	Col 7	Col 8	Col 9	Col 10	Col 11
17.2	Right	13.2	V+	RKEY_VolUp	RKEY_VolUp	10015	10015	102	RKEY_VolUp	
18	<<	20	<<	RKEY_Rewind	RKEY_Rewind	10038	10038	145	RKEY_Rewind	
19	>	21	>	RKEY_Play	RKEY_Play	10039	10039	146	RKEY_Play	
20	>>	22	>>	RKEY_Forward	RKEY_Forward	1003A	1003A	148	RKEY_Forward	
21	Live	24	Stop	RKEY_Stop	RKEY_Stop	1003C	1003C	14A	RKEY_Stop	
22	Record	25	Record	RKEY_Record	RKEY_Record	1003D	1003D	14B	RKEY_Record	
23	Pause	26	Pause	RKEY_Pause	RKEY_Pause	1001A	1001A	106	RKEY_Pause	
24	Slow Motion	23	Slow Motion	RKEY_Slow	RKEY_Slow	1003B	1003B	149	RKEY_Slow	
25	Pip	27	Pip CCW	RKEY_Prev	RKEY_Prev	10041	10041	150	RKEY_Prev	
26	Archive	34	PlayList	RKEY_PLayList	RKEY_PLayList	10042	10042	151	RKEY_PLayList	
27	Text	28	Pip CW	RKEY_Next	RKEY_Next	10043	10043	152	RKEY_Next	
28	Pip Switch	29	Sat	RKEY_Sat	RKEY_Sat	10040	10040	15E	RKEY_Sat	
29	White	35	White	RKEY_Ab	RKEY_Ab	1003E	1003E	14C	RKEY_Ab	
30	Red	30	Red	RKEY_NewF1	RKEY_NewF1	1003F	1003F	14D	RKEY_NewF1	
31	Green	31	Green	RKEY_F2	RKEY_F2	10024	10024	10D	RKEY_F2	
32	Yellow	32	Yellow	RKEY_F3	RKEY_F3	10025	10025	10E	RKEY_F3	
33	Blue	33	Blue	RKEY_F4	RKEY_F4	10026	10026	10F	RKEY_F4	

SPECIAL COMMAND CODES

									0		
				RKEY_Cmd_1	RKEY_Cmd_1	0x10027			0		test colour bars
				RKEY_Cmd_2	RKEY_Cmd_2	0x10028			0		blank screen
				RKEY_Cmd_3	RKEY_Cmd_3	0x1002A			0		all channels go blank
				RKEY_Cmd_4	RKEY_Cmd_4	0x1002B			0		all channels go blank
				RKEY_Cmd_5	RKEY_Cmd_5	0x1002C			0		
				RKEY_Cmd_6	RKEY_Cmd_6	0x1002D			0		jumps to the beginning of the timeshift/record buffer
				RKEY_Cmd_7	RKEY_Cmd_7	0x1002E			0		
				RKEY_Cmd_8	RKEY_Cmd_8	0x1002F			0		
				RKEY_Cmd_9	RKEY_Cmd_9	0x10030			0		
				RKEY_Cmd_a	RKEY_Cmd_a	0x10031			0		unit stops responding
				RKEY_Cmd_b	RKEY_Cmd_b	0x10032			0		test bars – black and white split half screen
				RKEY_Cmd_c	RKEY_Cmd_c	0x10033			0		test bars – black and red split half screen
				RKEY_Cmd_d	RKEY_Cmd_d	0x10034			0		test bars – black and green split half screen
				RKEY_Cmd_e	RKEY_Cmd_e	0x10035			0		test bars – black and blue split half screen
				RKEY_Cmd_f	RKEY_Cmd_f	0x10036			0		**FACTORY** RESET