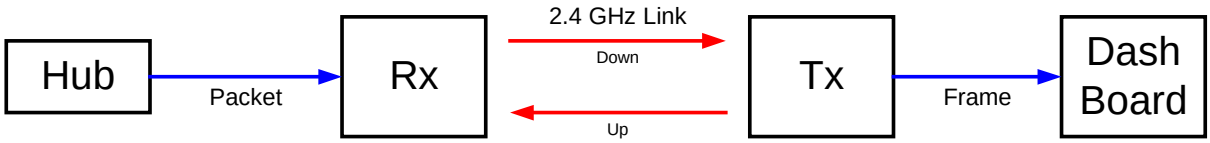
FrSky Two-Way Protocol Mechanisms

1 Introduction

Abstract:
Present document reflects the understandings of the authors of the FrySky Two-way Protocol Mechanisms as observed by our tests with the FrSky Transmitter module DFT (V2), Receiver module D8R (V2).

Definitions:



Synoptic

Hub: a device sending entities of n raw bytes per Packet over RS232, no “Byte Stuffing” applied.

Serial setting of the Hub: TTL-Level,
9600 bps,1 start bit, 8 data bits, 1 stop bit, inverted
or
RS232 Level,
9600 bps,1 start bit, 8 data bits, 1 stop bit, non-inverted

Rx: The FrSky receiver.

Tx: The FrSky transmitter module.

Dash Bord: a device receiving entities of 11 bytes per Frame over RS232. “Byte Stuffing” must be applied to the input stream.

Test Conditions:

Input for setting alarm threshold has been not used. No alarm frames send by Tx. Thus the output of alarm threshold frames has been not taken into account.

2 Frame protocol

2.1 Remote Voltage and Link Quality

Frame Contents

1	2	3	4	5 **(1)**	6	7 .. 10	11
Head	PRIM	Analog value for	Analog value for	(Up) Link Quality	(Down) Link Quality	4 Bytes	End Byte
0x7E	0xFE	Port1	Port2	Min ≈ 40 Max 110	2 * RSSI	00	0x7E

Remark (1): With shielded Rx and Tx antenna and reduced Power Mode (Tx) measured min (Up) Link Quality was 39 and than no reception of (Down) Link Quality.

Transmission:

Mechanism: time triggered, measured frame cycle ≈ 36 ms

2.2 User Data

Frame Contents

1	2	3	4	5	6	7	8	9	10	11
Head	PRIM	Length of valid bytes in frame	Frame Counter Modulo 32	User byte	User byte	User byte	User byte	User byte	User byte	End Byte
0x7E	0xFD	No. of valid bytes	0 .. 0x1F	byte1	byte2	byte3	byte4	byte5	byte6	0x7E

Transmission:

Mechanism: event and time triggered.

Model (so far as understood)

Incoming raw User Data of a Packet are buffered. The buffer is cyclically (time triggered) scanned and up to 6 bytes are sent in one frame. Empty buffer is never sent.

Event: buffer written

Buffer length TBD, but at least 8 (tested) bytes.
Scan Cycle is TBD, but assumed to be ≈ 36 ms.

A collision occurs if the buffer is written while scanned. In that case, the actual availabe bytes are sent in a first frame. Remaining bytes are sent in consecutive frames, this is called “Packet Splitting”. In that case “Packet Assembling” must be performed in order to not loose data. Consecutive frames are scheduled to be send every Scan Cycle (i.e. every ≈ 36 ms).

Since the Remote Voltage Frame is cyclically scheduled to be send too, it happens that either

- a Remote Voltage Frame and a User Data Frame or
- two User Data Frames

are sent quasi “back to back” , i.e. about every 15ms.

Note: Proper Packet Assembling requires that the Dashbord HW and SW is able to receive and process every frame received with a rate <= 14 ms.

If the Packet contains more than 6 bytes, the Packet is **always** split into several frames sent consecutively. A real capture of 8 byte Packets is provided by example 2.

Packet Assembling:

Packet Assembling without loss or erroneous data is not trivial. See example 1 in the appendix. A suitable and reliable algorithm could base on the Frame Counter, since for a splitted Packet with up to six bytes the Frame Counter doesn’t increment for the consecutive remainder Frame.
We never have observed that, for Packets with up to six bytes, the Packet was splitted in more than two Frames.

Packets larger than 6 bytes can be split in more than two Frames. In that case the Frame counter increments still every second Frame. Refer to example 2, 2nd Packet. Hence, an algorithm based on the Frame Counter is no longer reliable.

Finaly, we highly discourage the reader from employing Packets larger than 6 bytes.

3 Requirements for a simple protocol mechanism

User Data packets shall be send to the Rx without additional gaps between the bytes. This is **even** a general requirement.

User Data shall be send cyclically in packets of up to 6 bytes.

The User Data frame cycle shall be longer than the buffer scan cycle. Proven frame cycles are >= 100 ms.

At the reception on the Dash Board, only those frames shall be used were the length of valid bytes matches the number of the bytes sent per packet. Thus, some packets are lost.

4 Appendix

4.1 Examples

4.1.1 Frame Assembling Example

Frame Assembling vs start point

	Frame Bytes																	
									1	2	3	4	5	6	7	8	Sent Packet	
Assembly starts here >	1	2	3	4	5	6												
	7	8							1	2	3	4	5	6	7	8	Assembled Packet	
									1	2	3	4	5	6	7	8	Sent Packet	
	1	2	3	4	5	6												
	7	8							1	2	3	4	5	6	7	8	Assembled Packet	
									1	2	3	4	5	6	7	8	Sent Packet	
	1	2	3	4	5	6												
	7	8							1	2	3	4	5	6	7	8	Assembled Packet	
									1	2	3	4	5	6	7	8	Sent Packet	
	1	2	3	4	5	6												
Assembly starts here >	7	8															Assembled Packet	
									1	2	3	4	5	6	7	8	Sent Packet	
	1	2	3	4	5	6			7	8	1	2	3	4	5	6	Assembled Packet	
	7	8																
									1	2	3	4	5	6	7	8	Sent Packet	
	1	2	3	4	5	6			7	8	1	2	3	4	5	6	Assembled Packet	
	7	8																

4.1.2 Recorded Frame Sequence Example

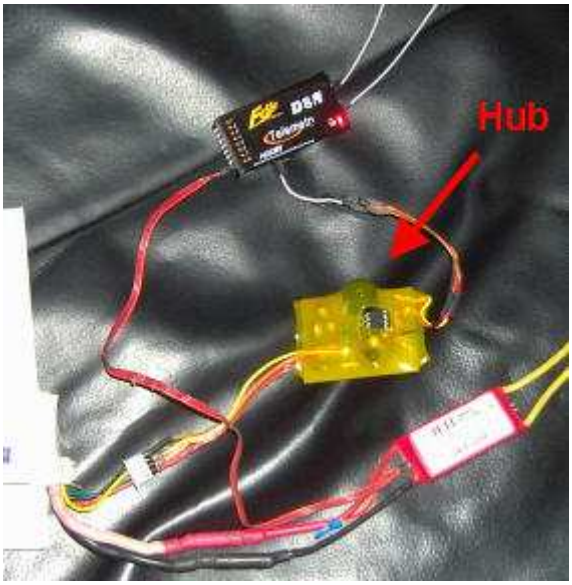
Recorded Frame sequence for 8 byte Packets

Elapsed Time [ms]	User Data Frame	Sent Packet
T= 453	F= 7E FD 3 11 1 2 3 4E E F 7E	0x1,0x2,0x3,0x4D,0xA,0xB,0xC,0xD
T= 32	F= 7E FD 5 12 4D A B C D 13 7E	
T= 468	F= 7E FD 6 13 4 5 6 4E E F 7E	0x4,0x5,0x6,0x4E,0xE,0xF,0x10,0x11
T= 16	F= 7E FD 1 13 10 D 0 0 E F 7E	
T= 31	F= 7E FD 1 14 11 5 6 4E E F 7E	
T= 454	F= 7E FD 6 15 7 8 9 4F 12 13 7E	0x7,0x8,0x9,0x4F,0x12,0x13,0x14,0x15
T= 15	F= 7E FD 2 15 14 15 0 0 12 13 7E	
T= 500	F= 7E FD 6 16 1 2 3 4D A B 7E	0x1,0x2,3,0x4D,0xA,0xB,0xC,0xD
T= 16	F= 7E FD 2 16 C D 0 0 A B 7E	
T= 484	F= 7E FD 6 17 4 5 6 4E E F 7E	0x4,0x5,0x6,0x4E,0xE,0xF,0x10,0x11
T= 16	F= 7E FD 2 17 10 11 0 0 E F 7E	
T= 484	F= 7E FD 6 18 7 8 9 4F 12 13 7E	0x7,0x8,0x9,0x4F,0x12,0x13,0x14,0x15
T= 16	F= 7E FD 2 18 14 15 0 0 12 13 7E	

4.2 Dash Board and Hub Hardware

The pictures are showing a Hub and a Dash Board.

The Hub is a LiPo Monitor for 3 Cells and is realised with a PIC12F683.



The display shows the cell voltage 1 to 3, the total voltage and the (Up) Link Quality.
The Dash Board is realised with a PIC16F687.