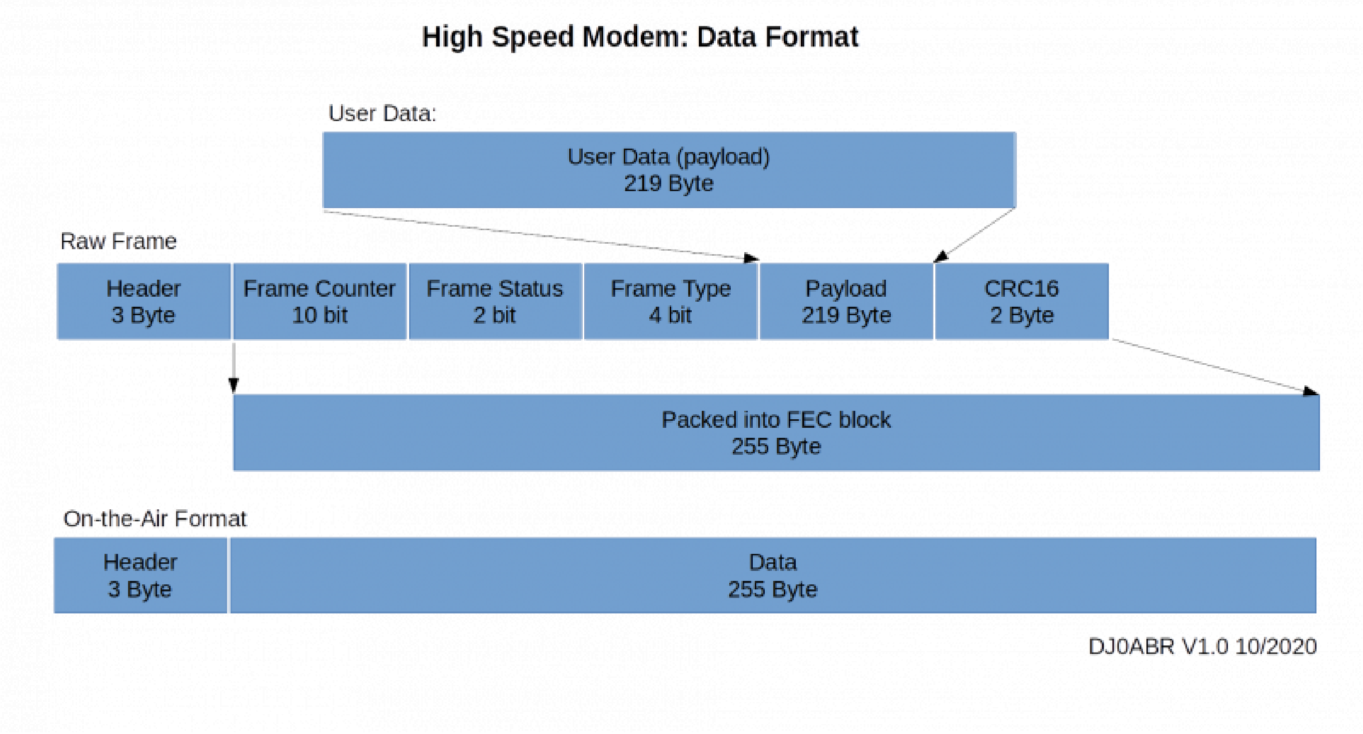
Frame Format



Data Processing Sequence - Transmitter:

(see file: frame_packer.cpp)

- The GUI (user interface or other application that needs to send data) delivers data in blocks of 219 bytes (payload). If a larger file has to be transmitted then the application has to split it into 219 byte blocks.
- this 219 byte payload is extended with additional information with 2 Byte (16 bit) length:

Frame Counter: a 10 bit counter (0..1023) which starts with value 0 at the beginning of a file transmission and then automatically increases by one for each sent frame.

Frame Status: a two bit status information:
0 … first frame of a larger file
1 … next frame
2 … last frame of a larger file
3 … first and last frame of a short file which takes only one frame.

Frame Type: specifies the type of the file, which has no meaning for the modem itself, but is important for the application:
1 … BER Test
2 … Image
3 … Ascii File
4 … HTML File
5 … Binary File

6 … Audio (i.e. Codec2)
7 … User Info (Callsign, Locator…)

- these 2 Bytes are added in front of the payload. Now we get 2 byte + 219 byte = 221 byte.
- in the next step the CRC16 is calculated over all 221 bytes. The two CRC16 bytes are added at the end, which results in a length of 223 byte
- these 223 bytes are fed into the Shifra-FEC function which generates 32 bytes of FEC information
- the 32 byte FEC is added at the end. Now we got: 223 byte + 32 byte = 255 byte
- finally the fixed 3 byte header is added and we got the complete frame of 258 byte ready to be sent
- convert bits to symbols according to the selected mode (BPSK, QPSK, 8APSK). See functions in constellation.cpp
- send these symbols to the modulator
- the modulator generates the audio samples
- send samples to sound card

Data Processing Sequence - Receiver:

receiving is almost the same procedure as transmitting, in reverse order:

- receive audio samples from the sound card
- demodulate the samples and generate symbols
- fill a received symbol in a FIFO which can hold a complete frame
- try to find the header-symbols at the beginning of the FIFO. Do this for each possible rotation.
- when a valid header is detected, check rotation, if required back-rotate the complete FIFO
- convert symbols to bits
- de-scramble it (length: 255 byte)
- run the FEC, cancel the frame in case of FEC error. Remaining length: 255-32 = 223 bytes.
- run the CRC16 check, cancel the frame in case of CRC error. Remaining length: 223-2 = 221 bytes.
- read the 2 byte frame status/counter. Remaining length: 221-2 = 219 bytes.
- send the 219 byte payload to the application

From:
https://wiki.amsat-dl.org/ - **Satellite Wiki**

Permanent link:
https://wiki.amsat-dl.org/doku.php?id=en:hsmodem:packer

Last update: **2021/03/28 19:14**