Serial protocol

To eliminate any chances of faulty transmitted data a convenient data transmission protocol was implemented. While it may seem difficult to understand at the beginning it is actually very straight-forward.

At the beginning of the transmission a byte called Start Of Frame (SOF) is sent which indicates the beginning of the frame.
After SOF the data bytes are send which contain payload. To check whether the payload bytes were correctly received another two bytes containing CRC are sent.
Lastly another SOF byte is sent to indicate the end of transmission. CRC is CRC-CCITT with initial value 0xFFFF.

But what happens when payload or CRC byte is accidentally the same as SOF? A transmission would end too soon and produce unexpected results halting system's stability. To solve the problem an escape byte was implemented.

Escape byte is sent before every SOF or escape byte that appears in payload or CRC. The actual byte is then sent right after the escape byte with inverted fifth bit.

Table1 SOF and Escape byte

SOF	0x7E
Escape byte	0x7D

Table2 Basic data frame which sets motor 0 speed to 10 rps

0x7E	0x83	0x00	0x00	0x27	0x10	0x1f	0x83	0x7E
Start of frame	payload byte 0	payload byte 1	payload byte 2	payload byte 3	payload byte 4	CRC byte 0	CRC byte 1	End of frame

Table 2 shows a basic data frame with SOF at the beginning and end and payload containing 4 bytes. An escape byte was not necessary because none of the payload and CRC bytes matches the SOF or escape byte.

Table3 Frame with payload byte matching SOF

0x7E	0x7E	0x7E
Start of frame	payload byte	End of frame

Table4 Frame with payload byte matching SOF and escape byte

0x7E	0x7D	0x5E(1)	0x7E
Start of frame	Escape byte	payload byte	End of frame

(1) 0x7E with inverted fifth bit

Table 4 shows how frame from table 3 is transmitted. Escape byte is added before the problematic payload byte. The payload byte is then transmitted with inverted fifth bit.