Assignment

Implement the parser for AIR GDL 90 protocol extensions and a data message.

The parser should be implemented as a C language function that takes a single byte (next byte of the data stream). The function should output valid packets as they arrive.

The function may take additional arguments (e.g. parser run-time state).

The parser shall not block and shall not use dynamic memory allocation.

Protocol description

FAA and GARMIN define the GDL 90 Data Interface Specification (link). The parser should comply with the specification.

For non-ADS-B related data, the GDL-90 Message ID is used to identify AIR as the sender of this data, using values of 0x40 and 0x41 as identifiers. This method allows us to differentiate between our own data packages using only one Message ID, reducing the potential of a message ID collision and allowing AIR messages to co-exist with standard ADS-B messages on the same link.

The data payload is an encapsulation of SAE J1939 messages (PGs). The basic message structure of an AIR interconnect message embedded into the GDL 90 frame is as follows:

The full PGN consists of 18 bits. The 18-th MSB is always assumed to be zero. The 17-th MSB is derived from the Message ID. The Message ID of 0x40 covers PGN range of 0x0000 to 0xFFFF. The Message ID of 0x41 covers PGN range of 0x10000 to 0x1FFFF.

The SA identifies the sender of the message.

Byte Order

Unless specified otherwise, data packaged in the PDUs is *little endian*.

Bit Order

Bits are numbered from the beginning of the message with respect to endianness (least-significant *byte* first).

For example a field, defined as `bit offset 6, length 4` would have its bits arranged as follows (field bits in brackets):

Byte offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	7 (1)	6 (0)	5	4	3	2	1	0
1	15	14	13	12	11	10	9 (3)	8 (2)
2	23	22	21	20	19	18	17	16
3	31	30	29	28	27	26	25	24

Fixed point (Q notation) numbers

For optimal usage of the communication bandwidth, the numbers communicated over the communication links may use reduced precision and a fixed decimal point, rather than IEEE 754 floating point numbers that always occupy (single precision) 32 or 64 (double precision) bits.

The Q notation used by is defined as follows:

- For signed numbers the notation consists of the letter Q followed by a number ***n*** defining the number of fraction bits not including the sign bit.
    - The total number of bits is ***n+1*** bits where the extra bit is used for the sign.
    - The range of the number is [-1; +1].
- For signed numbers the notation consists of letters UQ followed by a number ***n*** defining the number of fraction bits.
    - The total number of bits used by the number is ***n***.
    - The range of the number is [0; +1].

Numbers that do not fit into the range of the Q-number are mapped to the [-1; +1] range for unsigned or [0; +1] range for signed numbers by using the denominator which is specified after the Q notation and is delimited by a "/" symbol, i.e. *Qn/r*.

For example:

- Q9: represents a signed number in range [-1; +1] and resolution of `1/511=0.001956947`. The number occupies 10 bits. A number of -0.5 will be mapped to a -255 decimal number or `1100000001` two's complement binary.
- UQ9: represents a signed number in range [0; 1] and resolution of `1/511=0.001956947`. The number occupies 9 bits. A number of 0.5 will be mapped to a 255 decimal number or `011111111` two's complement binary.
- Q9/9.81: represents a number ranging from [-9.81; +9.81] and resolution of `9.81/511=0.019197652`. A number of 4 will be mapped to a 208 decimal number or `0011010000` two's complement binary.

Sample message - PGN 65282: Requested Flight Command

Requested command (flight mode, attitude and climb rate) packed into a binary structure:

Bit offset	Bit length	Name	Data type	Units
0	8	Flight mode flags	UINT	bit encoding
8	10	Roll angle command	Q9/90	deg
18	10	Pitch angle command	Q9/90	deg
28	10	Yaw rate command	Q9/90	deg/s
38	10	Climb rate command	Q9/30	m/s