True Anomaly

Code Test, FSW

1. An IMU outputs a byte stream at 921600 baud with a packet definition described below. Assuming some flavor of Linux running and data formatted according to IEEE-754 in network byte order, write the C++ code to parse data over a UART (/dev/tty1) and decommutate the byte stream into the four fields:
   Packet_Count
   X_Rate_rdps
   Y_Rate_rdps
   Z_Rate_rdps

Byte Offset	Size	Type	Description
0	1	Raw byte	Start of frame byte – 7F
1	1	Raw byte	Start of frame byte – F0
2	1	Raw byte	Start of frame byte – 1C
3	1	Raw byte	Start of frame byte – AF
4	4	Unsigned Integer	Packet count
8	4	Float	X-axis gyro rate
12	4	Float	Y-axis gyro rate
16	4	Float	Z-axis gyro rate

2. What modifications, if any, would you need to make to the code from 1) if the Linux host was a little-endian processor? Describe those modifications.
3. Write a C++ program that runs on Linux designed to execute in a resource constrained, multi-threaded environment that will execute the IMU parsing code every 80ms, and then broadcast the parsed results on the localhost network.
4. In Python, design a simple simulator/tester to drive your IMU parser. The simulator should send a string of bytes in the proper format over the UART for your parser to accept. The data contained in the byte string can be generated in any way of your choosing to best test the parser. Subsequently, it should then read the broadcasted results from 3) on the localhost network to validate the output.