Design Requirements: EEE3097S

Background: Some academics in the department of electrical engineering work in the larger RDI consortium called SCALE. In one of the projects where we are involved, we are building a generic flexible buoy that could be installed on the pancake ice in Southern Ocean to gather environmental data. You can read more about this buoy, which we have named as SHARC buoy (!) in Jamie's MSc dissertation. His dissertation is also a good example of how to approach a design problem.
One of the crucial sensors in our buoy is the IMU. It gives information about the ice as well as wave dynamics. Ideally, we would like as much of the IMU data as possible. But, transmitting the data using Iridium is extremely costly. So, we want to compress the IMU data. We also want to encrypt the data.

Problem Statement: Design an ARM based digital IP using the Raspberry-Pi to encrypt and compress the IMU data. Further requirements are given below. These can be used to derive the exact specifications for your design.

Req1. The IMU to be used is ICM-20649.
However, we shall not provide you these IMUs. We shall provide a different IMU. So, you shall need to find out ways to design your IP without the exact hardware! Here is a link to the device we will give you: Waveshare Sense Hat (B)

Req2. Oceanographers have indicated that they would like to be able to extract at least 25% of the Fourier coefficients of the data. Make sure that your compression satisfies this.

Req3. In addition to reducing the amount of data we also want to reduce the amount of processing done in the processor (as it takes up power which is limited). Try to minimize the computation required for your IP.

Suggested Methodology: This is a task with a few limitations. The biggest of which might be the lack of the actual sensor! Here is a suggested methodology.

1) **Requirement Analysis or Paper Design**: The first step is to understand the requirements and convert them into hard specifications. Then you must divide your IP into sub-systems and specify the inter and intra sub-system interactions. Lastly, you should work on detailed acceptance test procedures, i.e. how would you demonstrate that your design satisfies all the specifications.
2) **Implementation and Demonstration using Data from Computer:** In checking the functionality of your codes, you can set-up tests by sending data from your computer. These set of validations can be your next deliverable.

3) **Implementation and Demonstration using Data from an On-board Sensor:** Lastly, you need to show that the Pi can process data from an on-board sensor. Choose any sensors you think to be pertinent and use them to validate that your algorithms work.