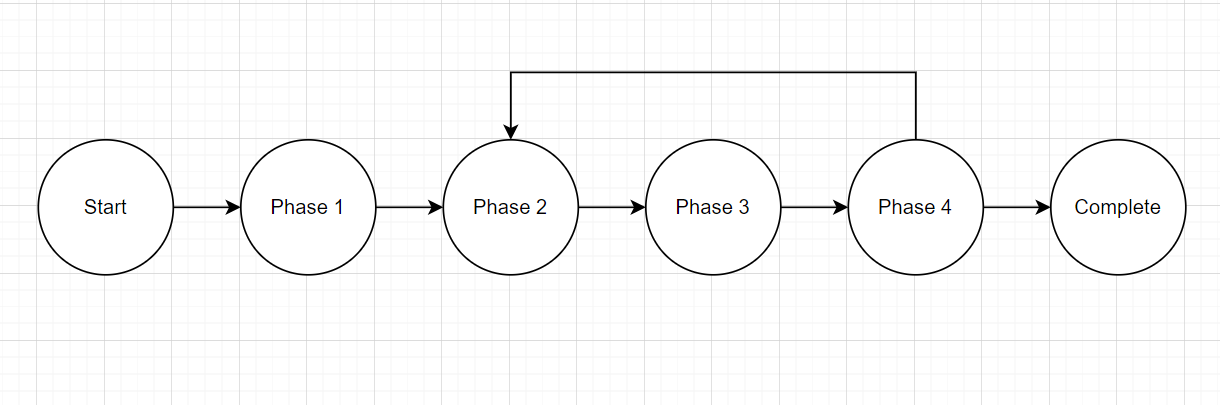
**SUMMARY MOVELLA IMU EXPERIENCE**

****

1. **Phase 1:** Receive imu Xsens DOT movella and approach related documentation

* You can get some tools and documentation on:

<https://www.movella.com/products/wearables/movella-dot>

<https://www.movella.com/support/software-documentation>

* Some information about how the sensor works, communicates with others is on:

<https://base.movella.com/s/xsens-dot-landing-page?language=en_US>

* We can study more about imu on youtube or some reliable pages
* Some useful information about Xsens such as:
  + Xsens take 3 seconds to init (Tips for best practise – Xsens user manual)
  + Signal process architecture:

Ảnh có chứa văn bản, biểu đồ, Kế hoạch, hàng

Mô tả được tạo tự động

* + Strap down integration is a method used to compute orientation and velocity increments by integrating angular velocity from gyroscope and acceleration from the accelerometer.
  + The 3D orientation of the sensor is computed by Movella’ latest Kalman filter core algorithm (XKFCore) for sensor fusion, which is optimized for human motions. If you want, you can research more about Kalman filter from the internet.
  + State transition:

Ảnh có chứa văn bản, biểu đồ, Kế hoạch, sơ đồ

Mô tả được tạo tự động

* + By default, the local earth-fixed reference coordinate system (L) used is defined as a right-handed Cartesian coordinate system.
  + Calibrated data has been going through Strapdown Integration and Inverse Strapdown Integration.
  + Output of calibrated 3D linear acceleration, 3D rate of turn (gyro) and 3D magnetic field data is in sensor coordinate system (S).
  + Yaw, pitch, roll (ZYX Euler Sequence. Earth fixed type, also known as Cardan or aerospace sequence)
  + Free acceleration is the acceleration in the local earth coordinate system (L) from which the local gravity is deducted.
  + Max frequency in realtime streaming mode is 60Hz
* After we read some related posts and documents on the website, we could know some base knowledge such as:
  + We have Xsens DOT app to connect to sensor and it can do all functions of sensor like measurement, synchronization, config, change mode, …
  + We have some tools for developers to make their own projects: PC SDK, data exporter for recording mode, mobile SDK, or it can connect through BLE protocol base on published manufacturer’s documentation. Please note that SDK tools require contacting manufacturers for installation, I don’t know details about this because I connect through BLE. The reason why I chose BLE is that I want to have wireless communication in realtime, and it is available option at the current.
  + In BLE documentation, you can find about its services UUID, characteristics, description, length, size, property, data structure, …
* Choose framework and mcu for your project. I choosen ESP32 and arduino framework because it is popular and easy to approach.
* Search for some example code about BLE client on ESP32, code and adjust based on Xsens BLE document.
* Some examples of BLE code:

<https://randomnerdtutorials.com/esp32-ble-server-client/>

<https://gist.github.com/elktros/6236b54d0ebe74c9a347b3d5d55993df>

1. **Phase 2:** Survey data received from sensor

* Conduct survey data. In my project, I am using DOT app to read the sensor in realtime mode with different configs. Some test cases such as:
  + Pull sensor forward, stop and pull backward
  + Rotate the sensor around
  + Rotate the sensor right at 90 degree.
  + Pull the sensor in quare
  + Pull up, down
* Cause using DOT app, I send the data file (.csv) to my computer to visualize it using python. You can refer to some examples about how to visualize data with python on ChatGPT or Google.
* Visualize data to see what events occur during the testcase.
* A few takeaways from the survey are:
  + Xsens DOT takes 3 seconds to init (confirm)
  + When it rotates, it got some magnetic noise and cost about 15 seconds to normalize.
  + The free acceleration x, y in stationary state fluctuates around [-0.05;0.05] while z is around 0.33 + [-0.05,0.05]. But in this project we do not use facc z.
  + Euler angle is around [-180;180]

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Phông chữ

Mô tả được tạo tự động

1. **Phase 3:** Research some ways to process data on MCU or embedded computer

* Implement code on ESP32, because it has dual core, so we can include free RTOS library and divide it into 3 tasks:
  + Task 1: Process notification data from notify callback function of BLE
  + Task 2: Process device state to control device
  + Task 3: Start measurement service, inform connection state
* According to BLE documentation, you can check the length of measurement data for each mode. The data received in byte with first byte is LSB.
* Add some crutial functions in my case that is data converter, distance calculation, angle calculation, motor control driver, queue, …
* To calculate distance, we just integrate the free acceleration value twice.

<https://www.youtube.com/watch?v=BoMO-Peobsw>

1. **Phase 4:** Implement and testing

* Place the sensor on the car, run and monitor the problem. In this phase, you can estimate qualitative and quantitative by making a way or putting the pen on the car to check the routine.
* If you have any problem back to phase 2.

1. **Reference**

* Source code: <https://github.com/DgTanDat/IMU_AI>