Documentation Part 2

# System integration

This project uses the LIS2DH accelerometer on the Thingy:52. The accelerometer outputs three floats, one for each of the x, y and z axis. The mobile node (Thingy:52) can transmit the sensor data over BLE at 40Hz, providing a good resolution data. The base node (NRF52840) received the BLE information, converts it to JSON format and send to the PC via UART.

After arriving at the PC, the acceleration data is passed into a machine learning model to predict the user’s activity/action. This machine learning model is a significant data processing procedure that processes acceleration data and turn it into desired information, thus integrating the sensor data into a usable product.

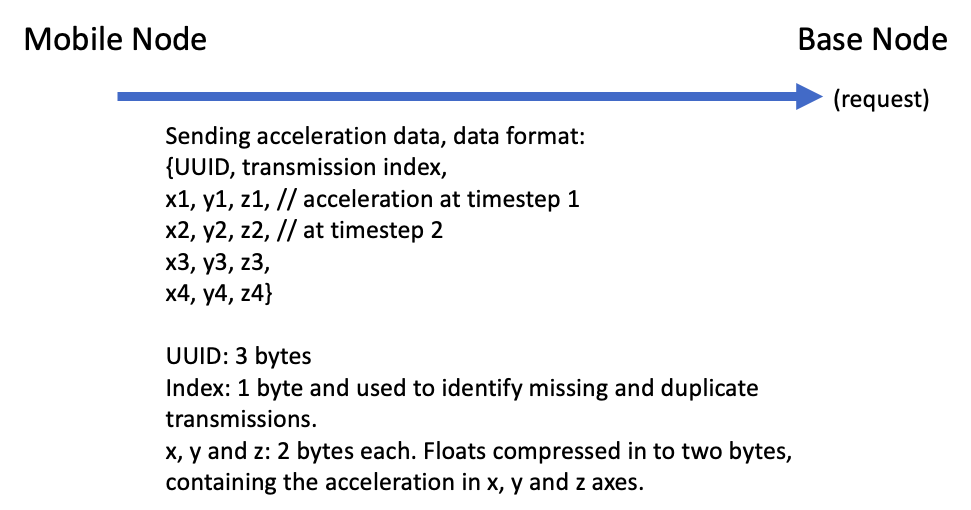
# Wireless Network Communications

The diagram below shows how the information is transferred through the network.

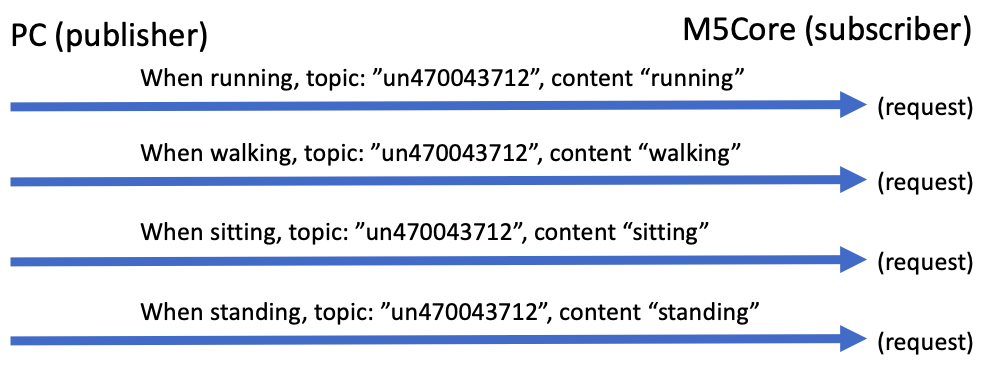
A diagram of a computer

Description automatically generated

The diagram below shows the message protocol between the mobile and base node:



The diagram below shows the message protocol between the PC and M5Core:



# Key Performance Indicator

**Test accuracy** measures the performance of the machine learning model on unseen data. The fact of being unseen ensures that the accuracy values is reflective of real-life applications and performance. The accuracy is measured in percentages:

|  |  |
| --- | --- |
| KPI ranking | Test accuracy |
| Excellent | >90% |
| Good | 80-90% |
| Satisfactory | 70-80% |
| Poor | 60-70% |
| Terrible | <60% |

**Latency** measures the delay between the user performing the action and the system displaying the action. This can be affected by data transmission time, delays in the machine learning inference and GUI refresh rate. Latency can be measured in seconds.

|  |  |
| --- | --- |
| KPI ranking | Delay (sec) |
| Excellent | <1 |
| Good | 1-3 |
| Satisfactory | 3-5 |
| Poor | 5-10 |
| Terrible | >10 |

**Number of parameters in the ML model** measures the complexity and how resource-intensity the model is. Having many parameters requires more training data and resource, and thus harder to train.

|  |  |
| --- | --- |
| KPI ranking | Number of parameters |
| Excellent | <5,000 |
| Good | 5,000-10,000 |
| Satisfactory | 10,000-50,000 |
| Poor | 50,000-100,000 |
| Terrible | >100,000 |

**Software functionality** measures how complete the GUI’s functionality and how easy it is to use.

|  |  |
| --- | --- |
| KPI ranking | Extent of software functionality |
| Excellent | All functionalities implemented. The GUI is intuitive and easy to use. The GUI provides additional relevant features/displayed information for use and debugging. |
| Good | All functionalities implemented. The GUI is intuitive and easy to use. |
| Satisfactory | All basic functionalities are implemented. |
| Poor | Some of functionalities are missing, impeding use. |
| Terrible | The GUI/software is missing. |

**Software reliability** measures how long the software can operate on average before crashing or freezing.

|  |  |
| --- | --- |
| KPI ranking | Software reliability |
| Excellent | It always works without crashes/freezes. |
| Good | It never crashes, but it may freeze briefly after 30 minutes of use and recovers by itself. |
| Satisfactory | Crashes or requires user intervention after 30 minutes of use. |
| Poor | Crashes/freezes between 10 to 30 minutes of use. |
| Terrible | Crashes/freezes within 10 minutes of use. |