Spike Plan

Name: Serial port -Socket

Context:

For this spike, it only consists of a sketch file, a client file, and a server file. The features are tested by using an Arduino board which consists of LED function and motion function but for this spike, only motion function is tested. For this spike, serial port is used as a communication platform between the Arduino board and the server side. The sketch file will read data from the motion sensor and serial port will retrieve data to the server side and calculate the timestamp of the server. Moreover, the server collects motion data and send the motion data to the client by using Socket.io for display purposes. Lastly, the client side will display the motion data such as motion status, the timestamp of the server and the time taken for the server to send data to the client which labeled as response time.

Gap:

For this spike, bugs can be handled easily because the program is simple and short. The code for motions to function properly in the device was separated into a different file so this reduced the complexity of the code structure.

Goals:

- Make sure the team understands serial port
- Verify the library for serial port
- Set up serial port and verify the requirements to have serial port to function properly
- Check whether the device was connected to the serial port
- Check if data was sent correctly.
- Check the time taken for the data sent from server to client.

Planned start date: 18/4/2017

Deadline: 23/4/2017

Planning notes:

Firstly, we studied the serial port library, syntax, and the requirements so that we are able to understand how to let serial port to function. We verify the spike requirements to give us a better understanding of the features that are needed to be implemented in the program. We shared the same workspace and done our code through Github. Once done, we compare the code structure with another spike to make sure they work the same.