Spike Plan 3

Name: SocketIO/JohnnyFive based communication spike.

Context:

For the upcoming project to create a morse code decoder, we need to decide which frameworks for communication are appropriate on both the server-client and server-IoT directions.

Gap:

We hope to understand the consequences of our choice of client-server communication method, especially one where no intermediary server is involved (SocketIO). We should take care to see how much of a delay this approach has compared to Firebase, and the reliability of the connection (making sure that no data is lost, etc.).

Goals/Deliverables:

The main goal here is to discover how difficult creating each of the two communication directions (server-client and server-IoT) is, and how difficult it is to integrate the two.

- A `server.js` file that will simply push the time of motion start and end events to the client using SocketIO. It will determine when motion starts and ends using JohnnyFive to communicate between IoT and server.
- A `client.js` file that will read from SocketIO and output the latency between actual motion event and display time in the client.

Planned start date: Apr 23, 2017
Deadline: Apr 23, 2017

Planning notes:

First, the server-IoT communication will be established using JohnnyFive. Once this is working and the server is capable of printing the motion ON/OFF and time of the event in milliseconds (ms) to the console, this information will simply be sent over SocketIO and client will be able to retrieve it. Finally, the client side code will be written to display the event, including the latency.