

# Journal 11

Wyatt Marvil Per. 5

25 November 2019

## 1 Progress Log

Re-fixed the issue with null devices appearing in the list of scanned devices.

Changed the connected device interface to only display the specific service of the sensors we wish to use. This service is denoted by a UUID beginning in FFE0.

Also query specifically for the read/write/notify characteristic of the sensors, denoted by a UUID beginning in FFE1.

Automatically connect to the correct service and characteristic and begin receiving data from the BLE module. Previously you had to choose the correct service and characteristic from a table menu.

Modified the Teensyduino's code to relay data from sensor 3 at 5 second intervals. For testing purposes.

## 2 Next Steps

Clean up the connected device interface and show a live feed of the data stream.

Parse the incoming data from hexadecimal to base 10.

Write characteristic data into a CSV file. Give the option to store the data locally.

Give the user the option to send the data to a registered email account. Once this component is implemented we are ready to begin training the neural network our team has developed.

Replace the temporary application icon with a custom designed logo. Spice up the application interface to feel more like a lacrosse app than a BLE area scanning app.

Create different modes for training the neural network. For example, in one mode the data could be specifically for passing vs non-passing data. This will help in the initial stages of data creation.

Continue learning gradle. At some point rewrite/edit the current automatically generated gradle system to ensure it is operating optimally. (I don't have a great deal of faith in IDE's like AndroidStudio. I prefer to work in more minimalist text editors.)

### 3 Reflection

This week was slow and aside from quality of life improvements, no real functionality was added. Over the upcoming Thanksgiving break I hope to implement several new functions, most importantly the generation of CSV files.

On a separate note, we have different modes we might like to have different neural networks as well. If we can create a specific NN for each mode with a narrower range of possible outputs and a more consistent set of inputs, the accuracy should be significantly greater than with a single generalized NN for every mode.