**EE 445L Lab 11 Report Deliverables**

**Requirements Document: Lightsaber**

1. **Overview**
   1. **Objectives**

The objectives of this project are to design, build, and test a custom embedded system. In practice, we are building a lightsaber, modeled from the movie Star Wars. The lightsaber should contain lighting and sound effects which are played in response to stimuli. Our goal is to demonstrate our understanding of embedded systems design, and showcase our knowledge gained in EE 445L.

* 1. **Process**

The project will be developed using the TM4C microprocessor. A custom PCB will be designed, and the microcontroller and other electrical components soldered to it. A JTAG header will be used to program and debug the embedded system. A battery and charging circuit will be soldered to provide power to the device. The user will use a custom Android app to control the sound and lighting effects of the device. There will be numerous software and hardware modules, all of which will be independently tested and validated. After each module is tested, the system will be built and tested.

* 1. **Roles and Responsibilities**

We are both the engineers and the clients because we are both making a lightsaber and want a lightsaber. Responsibilities will be divided based on accessibility to lab equipment and prior PCB design experience.

* 1. **Interactions with Existing Systems**

Our system will interact with a NeoPixel LED strip to output lighting effects. The NeoPixel’s interface with the TM4C using direct memory access.

Our system will also interact with our custom Android application through an ESP-01 Wi-Fi module. The user will be able to use this app to select color and sound effects to be loaded onto the embedded system. The Blynk app and interface covered in class will not be used. We will instead be using the standard AT command set that ships with the ESP module to interface with it. App communication will be done using a custom TCP protocol developed by us.

1. **Function Description**
   1. **Functionality**

We will attempt to closely resemble the functionality of a “real” lightsaber. There will be a handle for the user to grasp, and an illuminated endpiece. If the user presses the button, the lightsaber will turn on and off. Using the lightsaber’s wi-fi connection, the user will be able to configure lightsaber settings using an app, such as color and sound. Color can be selected from the Adobe RGB color space. Sound can be selected from a pre-loaded library in the firmware. Sound and lighting effects will play depending on the lightsaber’s movement.

There will be 8 different blade effects, 2 unique sound fonts, and a special nyan cat package.

* 1. **Performance**

We will be using three primary qualitative measures to assess the performance of the system. The system will be judged first by the Spring 2021 EE 445L class, at the end of year competition. Second, it will also be judged by our satisfaction of the project. Third, it will be judged by its ability to impress our course instructor and faculty advisor, Dr. Valvano.

Other secondary criteria include software organization and readability, the system’s response to stimuli, and ease-of-use.

The software will be organized in a readable, and sharable manner. Software reviewers will be able to understand the software architecture without needing a call graph. All code will adhere to the *Power Up Coding Standard for C and C++*.

The system will primarily respond to IMU input. The latency must be sufficiently low that the user is not able to observe it without specialized equipment.

* 1. **Usability**

The system will be easy to use, and not require any special training. There will be a single button on the side of the lightsaber toggle the system on and off. The user will download an app onto their phone from the Google play store, which can be used to control the systems sound and lighting effects. Sound effects can be chosen from a list of pre-established choices. Lighting can be chosen from the Adobe RGB color scheme, featuring 16.8 million unique colors. All other features will use input from the IMU to trigger, and not require any specific user input.

1. **Deliverables**
   1. **Objectives**

This 2-page requirements document.

* 1. **Hardware Design**

A detailed circuit diagram of the system will be uploaded to an official github repository, to be shared with the course instructor.

* 1. **Software Design**

A brief description of how our software works will be delivered.

All our software will be uploaded to an official github repository, to be shared with the course instructor and teaching assistants.

* 1. **Measurement Data**

Measurement data will be included as appropriate for our system. Data will describe power consumption of system and LED’s. We will also measure the duration of the Wi-Fi handler, and the smooth swing accuracy.

* 1. **Analysis and Discussion**

We will not be delivering any analysis or discussion regarding our system. A YouTube video featuring our product will be submitted to the course instructor and teaching assistants as requested.

**Software Design:**

Individual drivers have been built for numerous low-level hardware components, including the System Timer, NeoPixel LED’s, IMU, ESP-01, the button, sound, and battery charging.

On top of these drivers, are more complex function libraries, such as WiFi and Utilities.

All the above-mentioned modules are integrated in a main file. This file handles the primary control loop and manages initialization and updates for the other modules.

**Measurement Data:**

Power Measurements:

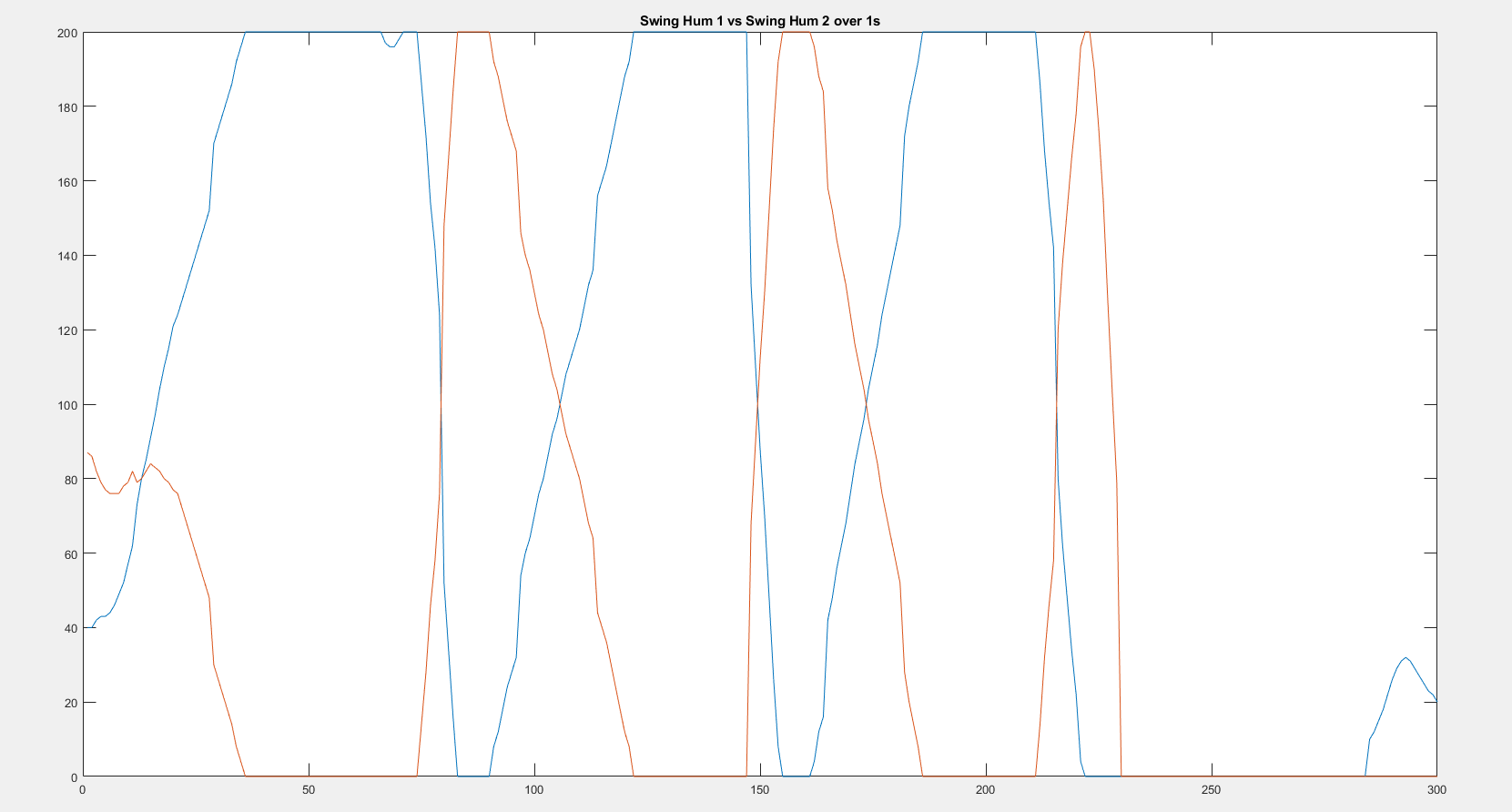
Average current drawn by total PCB: ~80mA

Current drawn by neopixels: Up to 15A

Timing Measurements:

Wifi Handler Duration: 8ms

Smoothswing accuracy:



IMU gyro error when not moving: ~0.9 dps