IM3080 Design and Innovation Project (AY2022/23 Semester 1)
Individual Report

Name: Lim Bing Hong

Group No: 3

Project Title: Tubes

Contributions to the Project (1 page)

Ideation of communication protocol and infrastructure

- -Research on frontend and backend libraries and hardware devices that fit the project requirements
- -Research on viable communication protocols between frontend and backend, and backend and hardware
- -Proposed communication network that is currently implemented

Hardware circuit design

- -Research on implementation of capacitors and resistors to reduce attenuation
- -Designed existing circuit by taking into regards of wire management

Implementation of circuit design

- -Modelled and 3D printed T junction for wire management
- -Implemented current hardware circuit with Ryan based on circuit designed
- -De-bugging hardware issues that arises

Hardware Ad-hoc tasks

- -Soldering of wires to LED Strips
- -Testing of LED strips and securing strips onto tubes

Backend

- -Research on Polling via HTTP request and WebSocket
- -Research on Flask Alchemy and SQLite to store data
- -Research on Nginx and Gunicorn
- -Implementation of Flask Server with WebSocket and SQLite
- -Creation of Queue Management system to grant access to users via a queue system

Frontend

- -Created the structure of Frontend interface using React
- -Linking of Frontend to Backend via WebSocket
- -Minor assistance in CSS for frontend design

Arduino

- -Research on multitasking on the single core Arduino Mega (multithreading is not possible) and shared with the team
- -Written code to link the Backend and Arduino Mega via serial communication
- -Assistance in debugging of Arduino Code

Raspberry PI

- -Wrote bash script to automatically run React and Flask servers when it is boot up

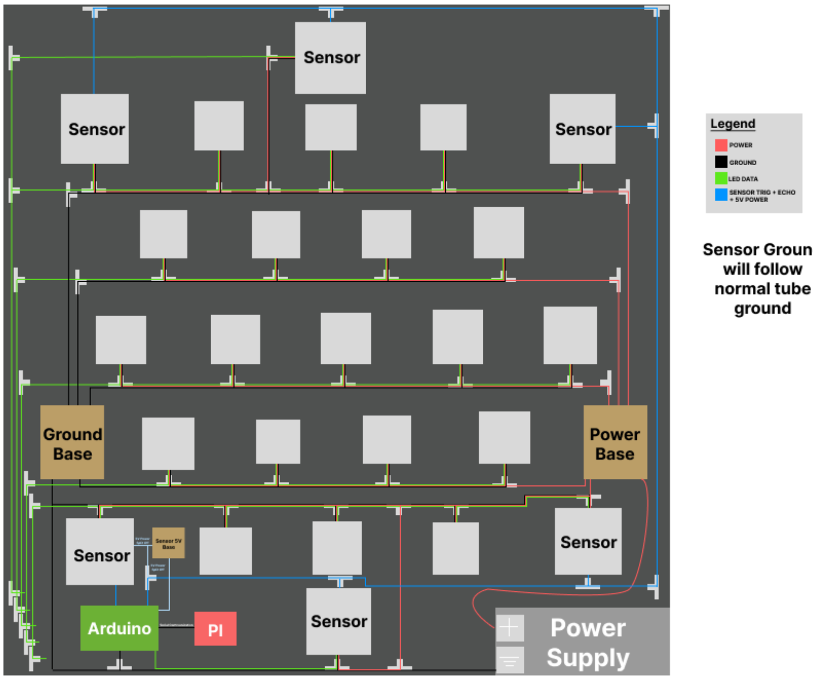
Reflection on Learning Outcome Attainment

Reflect on your experience during your project and the achievements you have relating to at least two of the points below:

(a) Engineering knowledge
(b) Problem Analysis
(c) Investigation
(d) Design/development of Solutions
(e) Modern Tool Usage
(f) The Engineer and Society
(g) Environment and Sustainability
(h) Ethics
(i) Individual and Team Work
(j) Communication
(k) Project Management and Finance
(l) Lifelong Learning

Point 1: Problem Analysis + Design/Development of Solutions

The initial implementation of the circuit was messy as wire management was not considered. This makes debugging extremely hard as finding and tracing the wire was extremely difficult. To resolve this issue, I understood that a circuit design is extremely important. Therefore, I used Figma to create a circuit design with wire management in mind. One of the faults in the previous implementation was that the parts were not modular. Therefore, I designed the new circuit such that each individual ground, data and power line will stop at the base of where the tubes will be located. I would then use a screw terminal wire connector to connect the wires from the tubes to where I laid the wires. This ensures that the wire from the tubes can be easily connected and disconnected whenever the situation arises. In addition, I also modelled and 3D printed a T junction to hold down and bundle the wires together for wire management.



Sensor Ground will follow normal tube ground

After implementation of the circuit design, the LED strips were able to light up. However, some of the LED strips did not light up according to the code I programmed in the Arduino Mega. Firstly, I analyzed the problem to try and determine its cause. All the LEDs on the LED strips were able to light up, however, the issue was that only the LED strips that I programmed were performing intended light patterns while the LED strips that I did not program were lighting up random lights. Therefore, I was able to isolate the issue to the data line of the LED strips. By researching on the problem online, I found out that by adding resistor to the data wire of the LED strips help to limit the amplitude of the rebound data signal on the led strip. When researching, I also found out that capacitor can be added across the + and – terminal to buffer sudden changes in current which may help with the issue or prevent future bugs. Therefore, based on the research I had done, I added capacitor across the + and – terminal and a 475-ohm resistor on the data line. Sadly, this did not solve the issue. This allows me to conclude that the problem lies with the data wire being too long which introduces noise into it as all the wires are bundled up together. Therefore, to resolve this issue, I reduced the length of the data line from the Arduino to the LED strips while keeping the capacitors and resistors. With these, the LED strips finally perform as intended.

Another problem that we faced was the interactions between users and our light display. As we want to integrate a software application to control the light display, we needed a system to prevent all the users from controlling the light display all at once which may cause unintended issues. Thus, I came up with a queue system implemented using the software. When entering the software, the user will be placed onto the queue.. Every user will only have a set amount of time to interact with the light display using the software. When the time is up, the user will lose control of the light display and the next user will gain control.

Point 2: Investigation

While designing the frontend and backend, I was looking into the best method for communication. I am used to connecting the frontend and backend through HTTP request. However, HTTP request is unidirectional, which requires the user to send a request so that the backend can reply to the user with data. What I wanted to achieve was a queue management system such that the server can notify the frontend when it is the user’s turn. One way to solve it was through polling whereby the user’s frontend will send a GET request to the backend at certain intervals. However, this seems to me as an inefficient method due to the overheads from HTTP request. Also, the server does not scale well with the number of users as the increase in number of users means that the server will be spammed with numerous GET request. Therefore, I investigate on other ways for communication between frontend and backend. This is when I come across WebSocket which allows a two-way interactive communication session between the user's browser and a server. Unlike HTTP request, WebSocket enables bidirectional connection between client and server and this connection will be keep alive until it is terminated by either party. This fits better in the circumstances of this project where real-time data is required.