The Light of Information

SI568 SP24 Course Project

TEAM ELM

Team Members: Yujun Zhang, Qian Dong, Yujia Gao, Jiawei Hu

Project Video

To better showcase our project accomplishments, we created a video demonstrating how light strips can dynamically change colors based on our designed functions, along with our reflections on the project.

Accomplished features

In this project, we accomplished seven features that can change of color expression of the light strip, including

- Adjusting the color of all light nodes
- Alternating the color of every other light nodes
- Rainbow color sequence
- Fading color density of every other light node
- Light “grow”
    - where light nodes gradually illuminate from one end to the other
- Light “chase”
    - where only four light nodes are illuminated simultaneously as the lit light passes from one end to the other
- Light “breathe”
    - where the color density gradually increases to the maximum, then decreases, and repeats

Besides, we opted to utilize a piece of instrumental music with clear beats as our data source and let the light strip interact with it. A Python library called `librosa` was used to help decoding the beats in the ".wav" music file and storing the corresponding timestamps in a Python array. We changed the randomly generated color of the strip every two beats. Additionally, we also attempted to get the bonus point by designing the light strip to change color sequences that match the festive spirit if the input date corresponds to a festival. Valentine’s Day, Saint Patrick’s Day, Independence Day, Halloween, and Christmas were included, with each assigned a festive color sequence. We hope this design can add more festival spirit to the SI base in NQ.

Technical decisions

We made the following major technical decisions.

1. We integrated the functionalities of the light strip as a class, incorporating attributes, such as the number of light nodes, the URL to connect with the strip, the initial value of a single light node, and the color values of the entire strip, because class can simplify our codes. All features that the light strip can perform have been encapsulated as class methods. We chose

to use “Class” as our data structure, as it simplified codes, making our developing work more flexible and easier for maintenance.

2. We utilized the “time.sleep()” method in Python to define the duration the light strip remains on before being turned off. The general rule for setting this time was to strike a balance, ensuring it lasted a reasonable amount that wouldn’t be too quick for viewers to capture the change, but also not too slow to risk boring the audience.
3. We have the color of the light change every two beats, rather than per beat. Our observations revealed that the time difference between each beat was consistently similar. It became challenging for viewers to discern whether the light strip was changing color at an even pace or synchronizing with the music beat. By changing the color every two beats, the pattern became more distinct and apparent.