ECE 445 Spring 2017 Group 37: Project Proposal

# **Choreographed Light Shows**

By

Jake Tuohy Joshua Schultheiss Miguel Cabrera Del Moral

#### Teaching Assistant Vignesh Sridhar

# **Table of Contents**

1. Introduction	1
1.1 Objective	1
1.2 Background	. 1
1.3 Synopsis	. 1
1.4 High-Level Requirements list	. 2

2. Design and Block Diagrams	
2.1 Block Diagram	
2.2 Physical Designs	
2.3 Functional overviews	
2.4 Block Requirements	11
2.5 Risk Analysis	
3. Ethics & Safety	
4. References	

## **1. Introduction**

#### **1.1 Objective**

LED arrays controlled by an algorithm are nothing in comparison to a professionally choreographed light show at concert. But the software used to design light shows is akin to that of professional photography; it relies on expensive hardware and has difficult-to-use software. With the resurgence of the "internet of things," we wish to bring the life of a professional light show to the living room.

We are going to make a consumer-facing light show recording system. The goal is to turn an array of lights into an instrument playable by a software interface similar to a soundboard. Light show recordings can then be mixed through our software and shared with friends. Playing back any light show will be automated through the Spotify API so that a light show plays tandem to its associated song.

#### 1.2 Background

The market for this project stems off the popular "Philips Hue" light bulbs. These bulbs have been a part of the growth in the lighting market along with the rise of cheap and bright LEDs[1]. While monocolor LEDs now sell for around \$10, multicolored LEDs are still sold around \$50[2,3]. This product could expand from consumers up to professionals. While an apartment is a good starting place, this could be easily scaled up to houses and small concert venues. Future renditions of the product could include hardware that would compete with large concert venues and arenas.

#### **1.3 Synopsis**

- Web application can pair with the wireless hub.
- Light shows can be recorded and saved to a centralized server.
- Manual adjustments to recordings can be made in a precise editor.
- User can play back recordings parallel with Spotify songs.
- Signals relayed from the web application will update the lights in real time.
- Manufacturing costs should be low enough to compete in the market.

#### **1.3 High-level Requirements List**

- Includes 7 main functions:
  - Off
  - Dimmer
  - Brighter
  - Pop
  - $\circ$  Fade
  - Strobe
  - Color Choice
- Step down voltage from 120 Volt outlet to roughly 5 Volts
- Proper synchronization between music and lights instructions

## 2. Design

The core of our project includes two key components: the software application and the hardware peripheral. Each of these include a technology stack as explained in the following sections.

The core of the application will be a MEAN full-stack web application[4]. This allows light show recordings consist only of high-level transition effects. Color palettes are responsive (constantly ebb and flow) but automated. Effects (sudden transitions) are manually triggered.

## 2.1 Block Diagrams



Figure 1: Block Diagram of the Hardware and Software agents of the project

## **2.2 Physical Designs**

#### Software Interface

The main interface of the application will be a "lightboard" similar to the soundboard in the introduction. Each row of the board will correspond to a light(s). Each column will correspond to an effect on the respective light(s). After some market research, the following effects are to be included in the project. The abbreviations to these effects can be found on the lightboard in the accompanying user interface mock-up.

off	Turn the light(s) to zero brightness					
dim	Turn the light(s) to a small brightness					
bri	Turn the light(s) to a large brightness					
рор	Quickly brighten the light(s) and quickly dim to original brightness					
fade	Quickly brighten the light(s) and <i>slowly</i> dim to original brightness					
str	Multicolor strobe					
col	Make light(s) sudden color change					
сус	Cycle the lights with pop effects					
fas	Fast strobe [intense white light only]					
slo	Slow strobe [intense white light only]					
fla	Long white flash [intense white light only]					
lig	Simulates a lightning strike [intense white light only]					

Table 1: Different effects in the lights and its associated abbreviation

1	off	dim	bri	pop	fade	str	col
6	off	dim	bri	pop	fade	str	col
2&3	off	dim	bri	pop	fade	сус	col
4&5	off	dim	bri	pop	fade	сус	col
S	off	dim	bri	fas	slo	fla	lig

Figure 2: View of the MIDI interface

#### Physical component

For the physical design of the LED modules, we have thought of individual blocks that can be separated or connected together into an array of lights. This design allows the user to place each light in any configuration they desire. All of the individual blocks would be wired to the main block that includes the power converters, the microcontroller, the WiFi receiver and the strobe light.

The individual LED blocks are composed of two parts as you can see in figure 3, the base and the LED light enclosure. The front part of the enclosure would be transparent and a curved shaped in order to reflect as much light as possible. Inside the light enclosure, we will have all the analog components and circuitry along with the red, green, and blue LEDs. We have divide it in two different components in order to have a rotating axis between the base and the light component, so the user can manually change the orientation of the light. The expected size of each block are around 60 to 120 mm in length, 20 to 40 mm in width and 60 to 120 mm in depth. The diameter of the light component will be around 40 to 80 mm.



Figure 3: Front view of the main LED block.



Figure 4: Front view of the blocks join together.

As it is shown in figure 4 the shape of the basement component, allows us to connect all of them together making a bigger array of lights. Joining them together with the same mechanism of a puzzle, make connecting and separating the pieces easy but at the same time is a robust way of sticking all the blocks together.



Figure 5: Back view of the blocks join together.

At the back of the block, in the light enclosure we have added two hooks in order to roll the excess of cable, when the physical enclosures are not connected. This will allow the user to have the cables in order, but at the same time have a cable long enough to place the light wherever he wants in the room.

For the speaker block it would be ideal if it has a configuration pretty similar to the LED lights blocks. But maybe in this case more space for the speaker is needed.

Finally figure 6 shows two examples of the possible configurations that this design allows. In the room of the left we see how each light is placed separate, having lights coming from each corner of the room, while in the right room all lights are together some of them pointing up and others pointing down.



Figure 6: Possible configurations of the lights in a room

### **2.3 Functional Overviews**

#### **REMOTE NETWORK**

#### **Remote Server**

This will be the back-end of the service. This will interface with the database and Spotify API to populate the web application with appropriate light show recordings. This will be built off Express to create its own private APIs. This could be hosted on a droplet as a cheap and expandable solution [5].

#### Remote Database

Using MongoDB, a NoSQL schema, the database will store all of the user data, including light shows [6]. The light show file will be a JSON object with an array of timing and color requirements for a light show, a list of authors, and a Spotify song ID. This way, shows can be played, edited, and credited.

#### LOCAL NETWORK

#### Web Application

The front-end application will communicate with the remote server (back-end) in order to save locally recorded shows and load previously recorded shows. It will also communicate with a local receiver, which will be our hardware unit that updates the colors of the lights to create effects.

#### **MIDI Controller**

The MIDI controller could be used as a physical instrument to control the lights as an option compared to the software touchpad. Alternatively, the traditional computer keyboard could be used as an external instrument through the help of a software key realignment guide.

#### CONTROL SYSTEM

#### Microcontroller

The main microcontroller will receive the input signals from the WIFI receiver and output them to each of the analog processors each group of LED's. The microcontroller will interpret the signal coming from the WiFi receiver and output the proper signals for the analog processors to modify the brightness and color of each LED. The microcontroller will also handle the audio signal and sending it to the speaker module.

#### WIFI Receiver

The WIFI Receiver module will be utilizes by receiving the signals stored on the server that were input by the user, and relay to the microcontroller. We are utilizing the use of WIFI because of its larger broadcast range so the device can be controlled from a larger range of points of a user's house. The fallback of using WIFI instead of Bluetooth is its larger power consumption, but because we are not using a battery to power the device this issue is easily avoidable.

#### POWER SYSTEM

#### AC/DC Rectifier

This rectifier would transform the alternating current coming from the grid, to a direct current. Then it would pass this direct current through a buck converter. This block will not be designed by us, as this has to be connected to the grid, and it would require us to work with high voltages and would put us and the grid in danger.

#### Buck converter

With the buck converter, we will eliminate the ripple coming from the rectifier, and reduce the voltage to a value that would probably be around 5V and that would power our entire design. In the event a smaller voltage is required, as it could be the case of the WiFi receiver, we would use a voltage divider to obtain the proper voltage for the device.

#### **DISPLAY SYSTEM**

#### Speaker

The speakers will be an additive module that would be able to be combined into the main housing of all the LED's. The speakers will also be wireless and receive the audio signal from the main microcontroller. As this is a completely additional module that could be easily removed from the LED's, the combination of the speakers is not essential in the completion of the main purpose of our project, but would make a more desirable product.

#### Analog processing

Process all the signal impulses coming from the microcontroller and transform them via analog circuitry of transistors, into different brightness of the RGB LEDs to display different colors and intensities. The control of the brightness of the LEDs will be accomplished changing the duty ratio of the pulse-width modulation (PWM) of the voltage.

#### **RGB LEDs**

Between 2 to 4 LEDs of each color red, green and blue, for each of the modules. They should resist up to 5 voltage.

#### Strobe

The strobe would be composed only of 6 to 12 white LEDs of high brightness, that also need to resist 5 volts. In the case of the strobe there is not any complex circuitry behind as it would always have the same brightness and color. It should only turn on and off at the pace of what the different functions dictate.

#### **2.4 Block Requirements**

#### Remote Server

- Be able to create an account
- Be able to save/load light show recordings to that account
- Spotify integration

#### Remote Database

- Manage the user and show data with correct read write permissions
- Safely store encrypted passwords

#### Web Application

- Users should be able to make an account
- Users are able to search the database for recorded light shows
- Users should be able to create their own light shows

#### Microcontroller

- Receive signal from WIFI receiver
- Be able to send analog signal to LED's
- Be able to send analog signal to speakers
- Run at 5 Volts
- Have precise voltage regulation in order to create accurate and different brightness levels in LED's

#### WIFI Receiver

- Be able to receive signals from software side of project
- Be able to relay the received signals to microcontroller in order to be properly allocated

#### Speaker

- Receive audio signal from microcontroller
- Output audio via speaker

#### Analog processing

- Handle at least 12 bits for RGB colors, 4 bits for each one, ideally 24 bits
- Control the PWM duty ratio of the 5V input, in accordance with the RGB bits values.

#### **RGB LEDs**

• Resist a maximum voltage of 5V

#### AC/DC Rectifier

- Compliance of the safety legislation for all the devices connected to the grid
- Transform 120V to 12V

#### Buck converter

- Transform 12V to 5V
- Ripple less than  $\pm 5\%$

#### 2.5 Risk Analysis

The highest risk item in the project is the voltage regulation to the LEDs and the microcontrollers. If voltage is not regulated correctly to the microcontrollers this could cause them to overheat and possibly burnup making them not useful. This would cause malfunction in signal control and ultimately cause our project to fail. On a slightly smaller scale the voltage regulation to the LEDs could also cause major problems. If voltage again is allowed to go to high, to the LEDs, then this could cause the LEDs to overload and pop or burn out making them no longer useful. While one or two LEDs burning out wouldn't completely destroy the project as in the microcontroller, it would still cause for a faulty a project and one that in neither safe nor complete.

## **3. Ethics & Safety**

The two largest safety concerns with our design is the power usage and the use of strobe/flashing lights. With any item on the market when using voltage converted from an AC 120 Volt outlet you run the risk overheating and burning up our project. This overheating could be a fire hazard, which is why we must take extra precaution in order to guarantee that none of the parts of our project will become to hot in order to cause a fire hazard. The second safety concern is the flashing lights which only concern a small group of people. Flashing lights can induce seizures from people who suffer from epilepsy. The highest risk frequencies are between 10-30 Hz, so we will attempt to avoid those for safety purposes. There will also be a warning on our project advising that any people that may suffer from seizures avoid the use of product in order to attempt to minimize this risk.

From an ethics standpoint one of the things we have to be careful is the rights to the music being played. For this reason in the production of this project no songs would never be loaded into the product when sold as this may infringe on distribution laws. Instead all music would have to be played through a licensed distributor or purchased in a legal manner.

## 4. References

[1] "LED Lighting Forecast." LED Lighting Forecast | Department of Energy. N.p., n.d. Web. 06 Feb. 2017.

[2] "LEDLightingDepot United States." Battdepot Australia. N.p., n.d. Web. 05 Feb. 2017.

[3] "Robot Check." Robot Check. N.p., n.d. Web. 06 Feb. 2017.

[4] "Home." Mongo Express Angular Node. N.p., n.d. Web. 05 Feb. 2017.

[6] "Cloud Computing Designed for Developers." DigitalOcean. N.p., n.d. Web. 08 Feb. 2017.

[7] "MongoDB for GIANT Ideas." MongoDB. N.p., n.d. Web. 06 Feb. 2017.

[8] Schultheiss, Joshua. "Jschlth2/web-keyboard." GitHub. N.p., 02 Mar. 2016. Web. 05 Feb. 2017.