Holiday Arboreal Light Project

Design Document

SDDEC18-10
Tom Daniels
Thomas Daniels
Aaron Hudson — Android Dev
Robert Tyynismaa — Android/Web Dev
Rajiv Bhoopala — Web App
Michael Scholl — Android Dev
Mir Ahbab — Electrical Engineer/Microcontroller
Justin Falat — Android Dev/OpenCV

Email: sddec18-10@iastate.edu
Website: http://sddec18-10.sd.ece.iastate.edu/

Revised: 3-4-18-V1

Table of Contents

List of figures/tables/symbols/definitions

PWM - Pulse width modulation

RGB - Red Green Blue

LED - Light emitting diode

1 Introduction

1.1 Acknowledgement

We would like to thank Dr. Tom Daniels for the assistance in researching and developing the plan for the project thus far.

1.2 Problem and Project Statement

Christmas is a time for celebration and with celebration comes the arrival of holiday displays. Many people decorate their homes and other objects like trees with sets of lights. However many products on the market are limited to individual design and creativity by not being customizable. For instance, in order to decorate an arboreal display in a pattern you would have first visualize that display and then lay the lights accordingly. The complexity changes when the person wants the patterns displayed to change. Thus, our team has decided to tackle this problem of being able to create complicated displays in a simple manner.

In order to provide more customization, our team wants to utilize technology in the form of smart phones and web apps. Our idea is for users to set up their lights on a tree and then upload patterns to that string of lights. Smartphone cameras will be used to record the position of LED's within the display. The data will be sent to our web app/android app where a 3d model of the display with LED's will be created. Users are then able to decorate their display through the web app/android app. Thus, the ability for customization is far greater than your average string of lights.

1.3 Operational Environment

Our light display will be set up in a person's home, most likely arranged around a christmas tree. As such our devices must not generate enough heat so as to start a fire. The system we design must also not draw too much power. Lastly, since this will be set up in a person's home and put on display, everything in the final design must aesthetically pleasing.

1.4 Intended Users and Uses

The intended users for our holiday lights are people who are interested in programmable LED lights, but are also interested in arts and gadgets. Our end goal is to create a product that can be used by anyone, even those who are not familiar with technology.

Although our project is mainly focused on creating programmable LED for standard pagan holiday arboreal displays, our final product can be used with anything in mind.

Some other uses include displaying the programmable LED lights on stairs, desks, bed frames, hangers, etc.

1.5 Assumptions and Limitations

Assumptions:

- User will only have one controller per house
- The user has a working WiFi connection
- User owns a smartphone with video functionality

Limitations:

- Keep cost at a minimum for components
- Have to use specific string of lights that are either individually addressable or use PWM

1.6 Expected End Product and Deliverables

Mobile Application - An Android Application for the user to perform set up. The user takes video from different points around the display, and the app will process the video to determine the location of each LED on the string. This will create a 3d representation of the display for the user, and they will be able to select a pattern to displayed. This information will be sent to a web server on the Raspberry Pi.

LED lights w/ Raspberry Pi Controller -The Raspberry Pi will have two modes: set-up and display. For set-up, the Raspberry Pi will send PWM signals to the string of lights so as to only have a certain amount lit up for a set of frames in the video that the Android app is capturing. After the location of each LED has been found and the user has selected a pattern, the Raspberry Pi will send the PWM signal for that pattern.

2. Specifications and Analysis

2.1 Proposed Design

Our design will be split into three component areas: hardware, mobile application, and the web app.

Hardware:

For the hardware, we will use a raspberry pi as a controller. This will be used to receive data in the form of images sent from a camera and then extract data to be sent to our webapp. For communication, we will use wifi to connect our different devices and app. We will use PWM addressable LED light strings for the type of lights and our controller will dictate which lights to turn on and when. A power source will be provided to power the controller and lights. We will include fuses in our design to ensure safety. The circuitry and controller will be kept in a box with the back side left covered with a grate for ventilation. The box will be covered in wrapping paper to look like a present.

The lights are strung up on a tree and powered through a 30V/12A supply, and connected to an Arduino at the moment for testing purposes. It is possible to select the pattern and brightness and send this to the lights.

Mobile Application:

We will use a smartphone camera to obtain data about the positions of each LED on display. The camera will have a user interface in the form of a triangle in the center of the screen. The triangle will line up with the christmas tree. 4 images will need be captured at different angles. The data will be sent to the controller and the position on the LEDs at each angle will be compared to create the final model to be sent to the web app. This has been discussed and a basic app has been pushed to the gitlab, but more work needs to be done. An algorithm has been developed for detecting LED positions but still needs to be implemented.

Web App:

The web app running on the controller will receive information from the mobile application about the location of the lights, as well as the pattern the user would like to display. This has not been implemented yet, and is still in the design phase.

2.2 Design Analysis

The lights have been connected to the power supply as well as to one another. An arduino has been used to run some tests on them to ensure that they are receiving enough power and that all LED's work up to our standards. Both of those statements hold true as all of the LEDs worked and displayed colors correctly.

The next step will be to develop and test a mobile application to detect the location of the LEDs, first in a 2D space then in 3D. Then establish communication between the mobile application and the controller so that in can run a test sequence for calibration. This should be relatively simple (does not require any complex code) and will just be built upon what already exists.

3 Testing and Implementation

Our design can be broken into three functional areas: the circuit, the camera, and the web app. We will need to test all three areas as well as the communication channels between each area.

Hardware/Circuit Tests:

These tests will revolve around making sure the hardware is connected properly and safely. Tests will be conducted so as to make sure that fuses are blown correctly when a threshold current is crossed. Current needs to be measured across components so to make sure they fall within component ratings. This is to ensure that components do not get burned out. Lastly tests in this category will revolve around our controller and whether information can be processed and calculations run correctly.

Mobile Application Tests:

These tests will be conducted so as to make sure that the camera interface we utilize is able to function properly and correctly obtain data. Initial tests will involve obtaining the the types of lights that the camera can detect as well as the environment light intensity that yields the best results. From there, testing will be conducted to see if the camera user interface works correctly.

Web App Tests:

The web app tests will include testing if the web app receives the data from the mobile application, and is also able to send the data to the hardware.

Cross Communications Testing:

These tests are to ensure that the system is able to communicate and transfer data correctly.

Camera to Hardware: Test to determine if data from the camera is received by the controller

Hardware to WebApp: Test to see if the controller can send data to the webapp

WebApp to Hardware: Test to see if the hardware can process data from the webapp to change the light designs.

3.1 Interface Specifications

In our design, there are multiple areas in which our different devices will have to communicate with one another. The mobile application will have to be able to send data to the web application, which then will have to be sent to the hardware, which will power the LED lights.

3.2 Hardware and software

The hardware/software we will be testing include the hardware/circuit tests, the mobile application tests, and web application tests.

The mobile application tests will include taking a picture from the camera and confirming if the application can pick up the LED lights correctly. Next, confirming that the user can select LEDs and choose colors/patterns for them.

The web application tests will include receiving the data from the mobile application and then sending it to the hardware. The first check will be to see if the data is accurate. The second test will be to confirm if the data is sent to the hardware.

The hardware tests include receiving the data from the web application and then converting it into what the user sees on the tree. Best case scenario is exactly what the user picked out on the mobile application (colors/patterns).

3.3 Functional Testing

The mobile application will be tested using jUnit tests. jUnit is common for java applications and allow us to easily find bugs and fix them. jUnit is already built into Android Studio, so the group will have no problem using it. Testing the application on its own will consist of the user navigating through the different parts of the application on several android devices. We are still unsure how the remaining applications will be tested as we still have yet to develop the applications.

3.4 Non-Functional Testing

Some of the big non-functional testing that will have to occur within our product will be usability. The goal for the application is to be very user friendly so that your average consumer understand how to operate the product. This also allows better performance and data acquired making the whole process easier.

3.5 Process

N/A.

3.6 Results

We have not done any testing yet (besides ensuring the LED strings work), so there are no results available at this time.

4 Closing Material

4.1 Conclusion

3 LED strands have been connected and strung up on a tree. They have been connected to a power supply and an Arduino to test to make sure the LEDs have the desired output. An algorithm has been designed to detect the locations of the LEDs given two images (lit and unlit) and to find each individual one after that. Equations have been written to convert 2D coordinates into cylindrical.

The goal is to have a mobile application that can detect the location of the lights in a 3D space. The user will then be able to select a pattern/create their own and have that displayed on the lights through the mobile application communicating with the controller.

At this point, our plan is to begin working on the LED detection in the mobile application and driving the lights through a Raspberry Pi, as well as inter-device communication between the two. This will provide the base for the more complex parts that will still need to be implemented (e.g. matching pattern to location of LEDs, user creating their own pattern).

4.2 References

To run the LED's currently set up on the tree we will require using a library to power them.

Currently we are running the lights off of an Arduino

In the future we will use A Raspberry Pi PWM library for WS281X LEDs

https://github.com/jgarff/rpi_ws281x

From the README: Userspace Raspberry Pi library for controlling WS281X LEDs. This includes WS2812 and SK6812RGB RGB LEDs Preliminary support is now included for SK6812RGBW LEDs (yes, RGB + W) The LEDs can be controlled by either the PWM (2 independent channels) or PCM controller (1 channel) or the SPI interface (1 channel).

OpenCV

```
@article{opencv_library,
    author = {Bradski, G.},
    citeulike-article-id = {2236121},
    journal = {Dr. Dobb's Journal of Software Tools},
    keywords = {bibtex-import},
    posted-at = {2008-01-15 19:21:54},
    priority = {4},
    title = {{The OpenCV Library}},
    year = {2000}
}
```

4.3 Appendices