Learning Holiday Lights Project

SDDEC19-10

PROJECT PLAN

Adviser
Thomas Daniels

Client
Thomas Daniels

Team
Jacob Grace
Chad Griggs
Thien Nguyen
Joseph Nunez
Valery Smith
Steven Williams

Email
sddec19-10@iastate.edu

Website
sddec19-10.sd.ece.iastate.edu

Table Of Contents

Frontal Material

List of Figures

List of Tables

List of Definitions

1) Raspberry Pi - Also referred to as “Pi”, is a “low cost, credit-card sized computer” that “enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python” [1].
2) LED - Also known as a “Light Emitting Diode”, these “produce light approximately 90% more efficiently than incandescent light bulbs” [2], are substantially smaller and are more readily individually programmable.
3) SSH - The Secure Shell Protocol “is a method for secure remote login from one computer to another” and allows for the use of the File Transfer Protocol [3].
4) RGB - Red, Green, Blue. The three colors human eyes are tuned to receive and the three colors used in a pixel
5) PWM - Pulse Width Modulation. The variance in time high over time total (duty cycle) in a time varying digital signal

Introductory Material

Acknowledgment

We would like to acknowledge Dr. Tom Daniels for his time and expertise with this project, as well as the previous senior design group for laying a foundation onto which we can build our design.

Problem Statement

General Problem Statement

During the winter holiday season, it is traditional for many families to set up interesting light displays to impress peers. Previous methods of putting static lights on trees or on the roof of the house are considered outdated by many standards. Here in 2019, we at Team 10 want to take the holiday season to the next level! The idea is instead of plain static lighting, we want to make dynamic lighting for all your holiday needs. With our product, you will be able to set up the lights on the object of your choosing and then make breathtaking visual effects of your choosing.

General Solution Approach

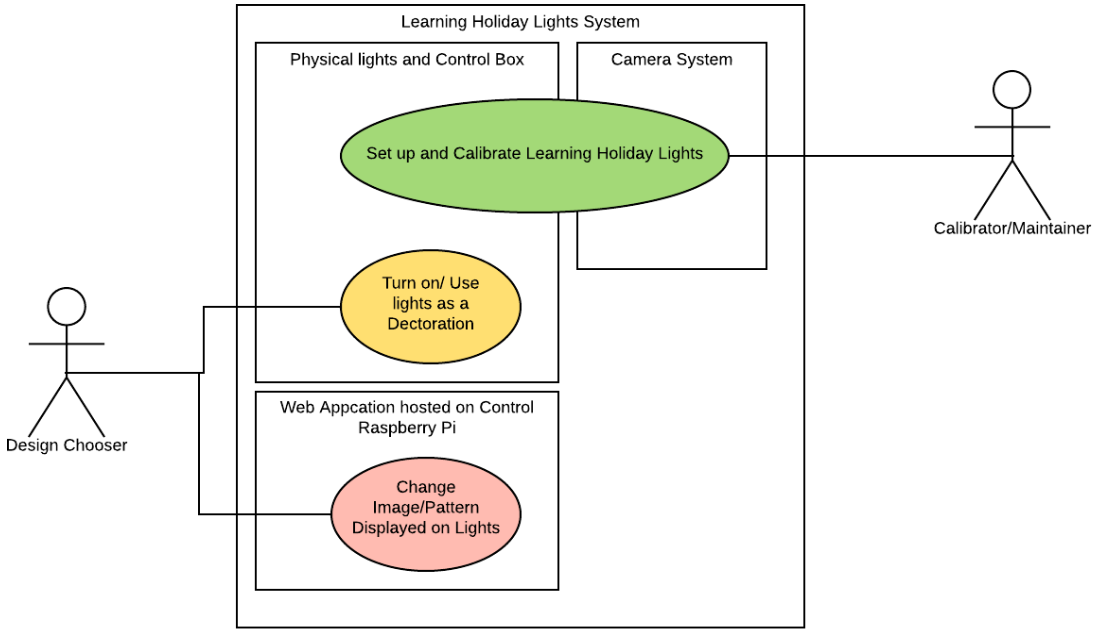
Our proposed solution is to take the current system created by the last group and improve it in a few ways. Our first improvement is to use a second Raspberry Pi as the calibration device instead of a phone app; this way we will not have to deal with compatibility issues with all the different phone options. We also removed the addition of a private server and instead are hosting all data (such as web-hosting, calibrations, and display).Our second improvement is a revised calibration system in which the Camera Pi will take photos of the tree during a calibration system, process those photos using OpenCV, then pass the results to the control Pi. At this point, the control Pi will interpret the results and control the tree.

Operating Environment

The system should function in controlled indoor environments, regardless of any visual noise such as reflective windows. The system may be moved after it is calibrated and the calibrating environment may have certain settings like less noise or lights on/off.

The enclosure for the Raspberry Pi and power distribution is designed to operate in climate controlled environment in order to allow adequate cooling and .

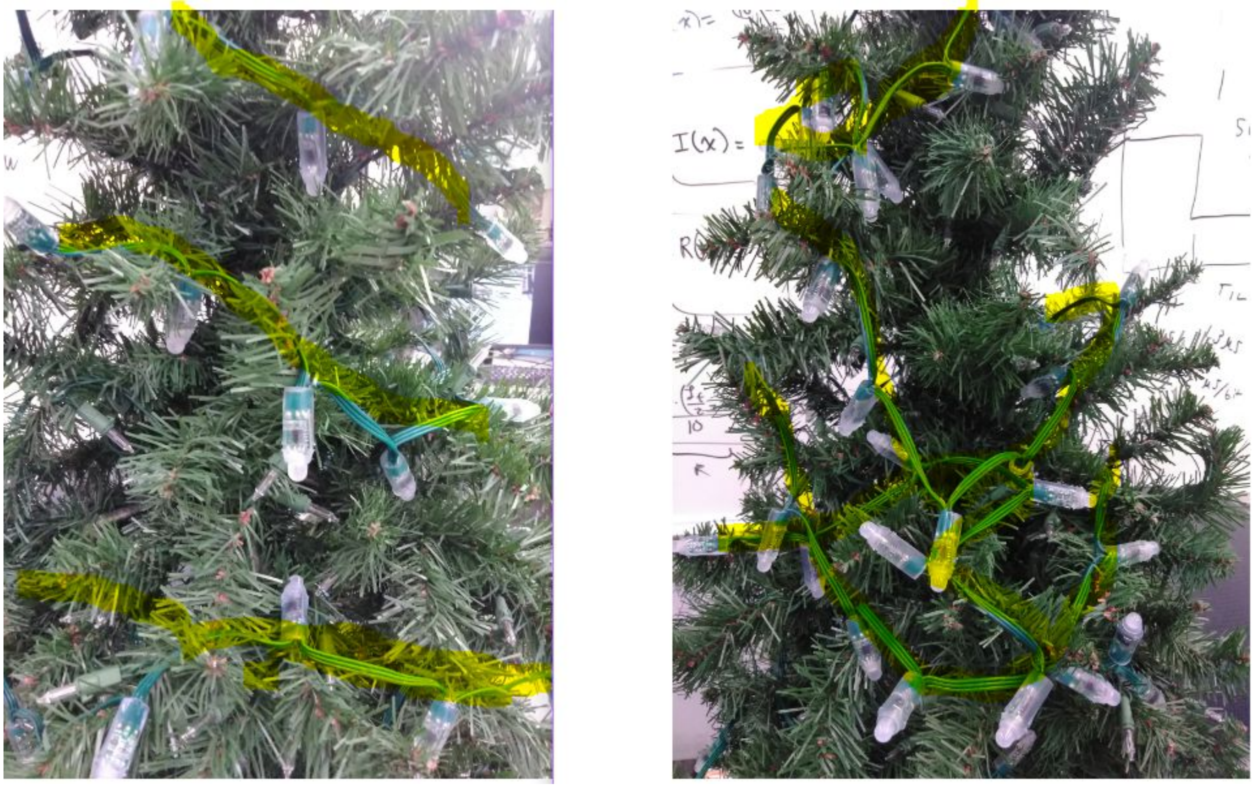
Intended Users and Intended Uses


Learning Holiday Lights System
Physical lights and Control Box
Camera System
Set up and Calibrate Learning Holiday Lights
Calibrator/Maintainer
Turn on/ Use lights as a Dectoration
Design Chooser
Web Appcation hosted on Control Raspberry Pi
Change Image/Pattern Displayed on Lights

[Figure 1]

Our intended users will fall into two role categories the Calibrator/Maintainer and the Design Chooser. The Calibrator/Maintainer has higher technological literacy requirements than the Design Chooser as they must have knowledge of IP addresses are able to follow instructions, and exhibit patience during calibration.

The calibrator is responsible for setting the system up, and ‘teaching’ the system where each LED is (calibrating). The set up consists of first stringing the lights around the tree or other objects in a neat manner then connecting the light string to the control box, see image below.



[Figure 2] The left side shows lights arranged correctly in a roughly parallel and linear fashion, the right side shows lights arranged incorrectly in a messy fashion

The process of calibrating the system that the Calibrator/Maintainer will use is

1. Identify the Control Pi IP
2. Enter the IP address in the browser and click calibrate button
3. Enter the IP address for Calibration/Camera Pi
4. User Follows remaining few on-screen instructions
5. User validates a successful calibration

The design chooser uses the system's UI to select a pattern or image for the system to display. The design chooser also presses the button on the control box to turn the system on and display a previously loaded/pattern image, and turn the system off. This UI is intended to be very user-friendly. Therefore, the design chooser could be almost anyone from older children up to elderly people.

The system is intended to be used as a fancy decoration primarily around the holiday season, but with the possibility of any RGB combination you can dream of, this could be used year round.

Assumptions and Limitations

Assumptions

1. The user has a working WiFi network.
2. The user has electricity.
3. The user has ample space to safely store the system.

Limitations

1. We have a string of lights that we are expected to use from the previous senior design group.
2. The speed of the system is limited by how fast each component can communicate with each other and how fast our image processing software solution can auto-calibrate the tree.
3. The system should not pose any safety concerns.
4. The system should be easy to set up for the user.

Expected End Product and Other Deliverables

User Interface: The user interface will be a user-friendly web application hosted on our senior design website. We would like to use WebSockets to have the server push events to the pi to control the lights. On the user interface the user can either upload images files to be texture mapped to the LEDs on the tree or design a basic pattern to be displayed such as a rotating rainbow or other select colors.

Lights and Control/Powerbox: The control/power box contains a 12v power supply, a 5v:3v step-down converter, and a Raspberry Pi. The lights are WS2811 LED/chipsets. Python files will be controlling the Raspberry Pi's PWM pin that is connected to the WS2811 LEDs.

Calibration Method: The method by which the system will determine the location of each individual LED on the tree. It will utilize camera images captured by a PiCam to be processed on the Raspberry Pi.

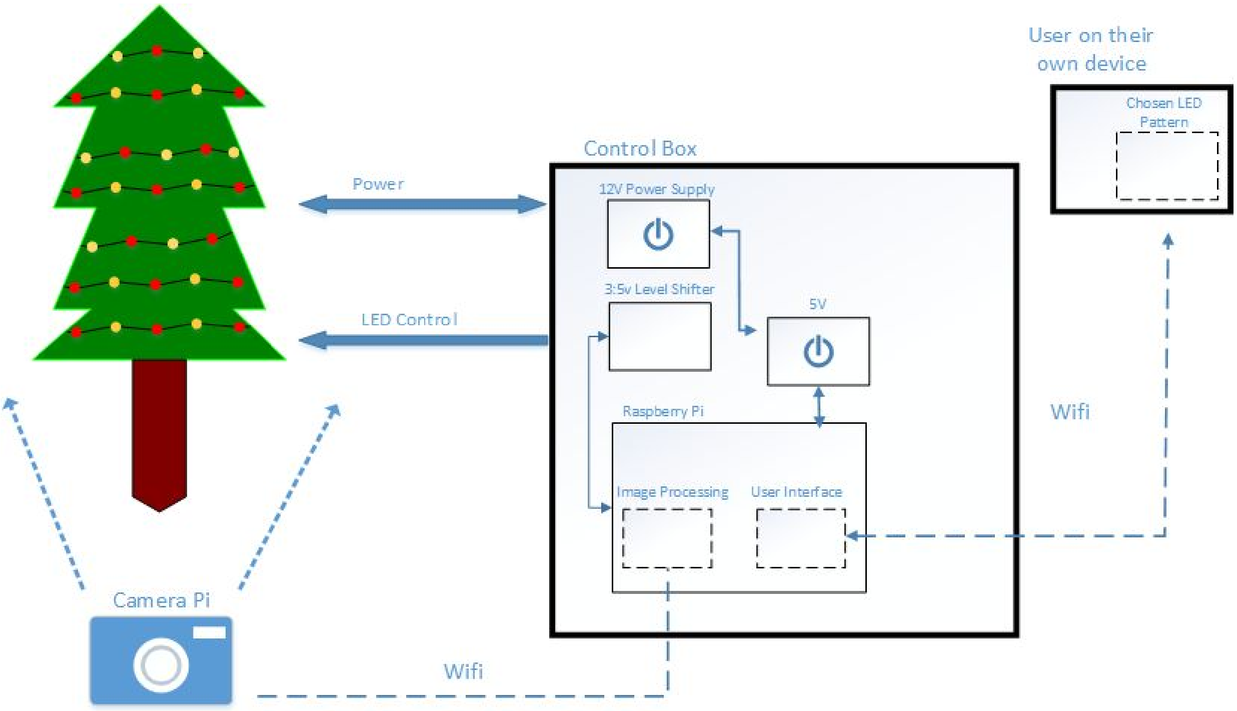
User Manual: a pdf detailing how to set-up, calibrate, change designs, and take down the system in an easily readable way. This document will contain diagrams and images to help guide the user.

Proposed Approach

Objective of the Task

The objective of this project is to engineer a customizable LED light display for use on a Christmas tree. After initial calibration is performed, the location of each LED will be learned by the product. Then through a user interface, the user will be able to customize the LED patterns displayed on the tree. This could come in the form of images being scrolled across the tree or various color gradients being displayed as well as countless other possibilities. The custom patterns will be sent through a web server and to the raspberry pi which will then control each individual LED.

High-level Block Diagram of System



[Figure 3]

Functional Requirements

FR.1: The system shall display user-defined patterns on
FR.2: The system shall utilize a calibration algorithm to determine the three-dimensional location of each individual LED

FR.3: The system shall be capable of operating for several hours at a time without overheating or posing any other safety threats

Constraints Considerations

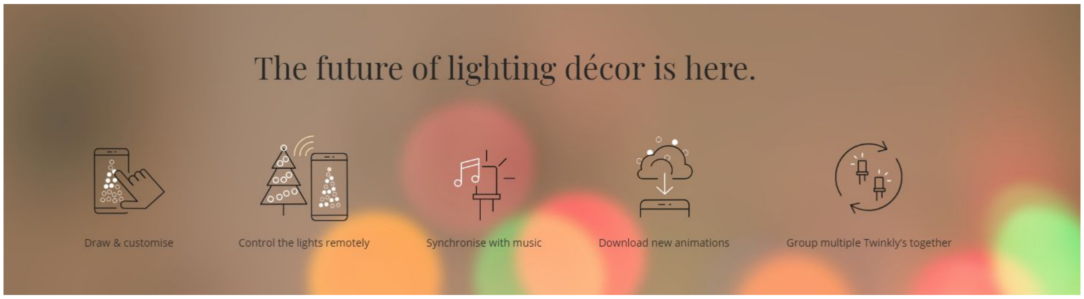
CC.1: The overall system design should not need to be over-budget (>$300).
CC.2: The system should not have security vulnerabilities from wireless communication.
CC.3: The system should be easily portable.
CC.4: The system should be understandable to use.
CC.5: The system should be easily expandable with more lights.

Relevant Standards and Specifications

1. IEEE 802.11 - Standard for Wi-Fi communication over 2.4, 5, and 60 GHz frequencies. The calibration Pi, control Pi, and user's device all communicate with each other using devices that conform to this standard.
2. IEEE Light emitting diode safety and safety standards.
   Our system needs to be safe for all users, LEDs can cause damage to eyesight, and other bodily harm if used improperly.
3. IEC 60320 - Standard for power supplies not exceeding 250V
   Power supplies can easily cause fire or overheating if not complying with this standard.

Previous Work / Literature Review

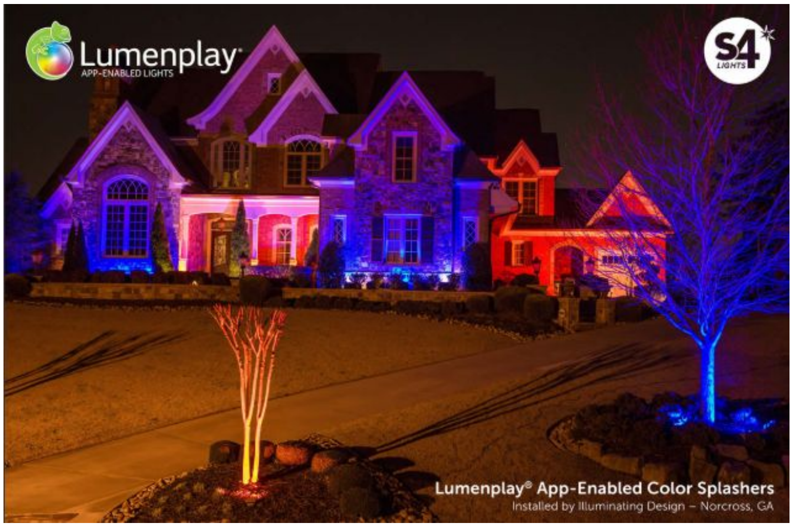
There is currently at least one product on the market similar to our project - Twinkly [4]. After some research, we assume their LED strand and controller communicate with an Android or iPhone app through wifi. The application takes into account that the LEDs are placed in specific locations on the tree; calibration is decided by how the strand of lights is laid onto the tree. For the calibration of the Twinkly tree the LEDs light up in a pattern and the user must place them correctly onto the tree. The application assumes the locations of the LEDs based on that process to display patterns. The Twinkly product is very similar to this project in terms of design and feel of a product. We would like for the application to change the design of the tree then, if time allows, draw and upload new designs. The difference between this project and the Twinkly product are the ability to analyze the light locations so that the lights would not need to be hung in a specific pattern, but rather arbitrarily; theoretically, they could be placed not just in a tree shape.


The future of lighting décor is here.
Draw & customise
Control the lights remotely
Synchronise with music
Download new animations
Group multiple Twinkly's together

[Figure 4]

[Figure 5]

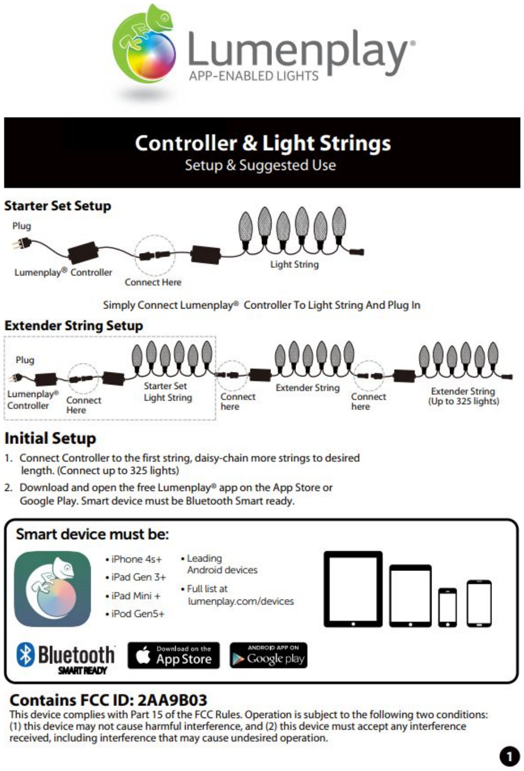
A second device on the market that will function similar to our intended product is the Lumenplay series of app-enabled lights [5]. These lights will connect to a wifi network and allow a user on their smartphone to control the lights allowing them to “choose colors, effects, and play games” [6]. Based on the companies ‘About’ and “FAQ’ pages, there is no reference to any sort of auto calibration for the lights. The website also fails to mention anywhere in their support documents how the user should string their lights.



Lumenplay® App-Enabled Color Splashers
Installed by Illuminating Design – Norcross, GA

[Figure 6]

Users are meant to figure out how to strand the lights themselves based on what they're trying to accomplish, such as covering the roof in symmetrical rows of lights. It is then assumed that the user inputs the number of lights they strung and a general location of where each strand is. In the end, this is going to be incredibly complex to set up on the user's end and is clearly not directed towards a casual audience. For example, the following image is the provided PDF document detailing how to set up and use the lights, but now how/where to place them:


Lumenplay
APP-ENABLED LIGHTS

Controller & Light Strings

Setup & Suggested Use

Starter Set Setup

Plug
Lumenplay® Controller
Connect Here
Light String

Simply Connect Lumenplay® Controller To Light String And Plug In

Extender String Setup

Plug
Lumenplay® Controller
Connect Here
Starter Set Light String
Connect here
Extender String
Connect here
Extender String (Up to 325 lights)

Initial Setup

1. Connect Controller to the first string, daisy-chain more strings to desired length. (Connect up to 325 lights)
2. Download and open the free Lumenplay® app on the App Store or Google Play. Smart device must be Bluetooth Smart ready.

Smart device must be:

- iPhone 4s+
- iPad Gen 3+
- iPad Mini +
- iPod Gen5+
- Leading Android devices
- Full list at lumenplay.com/devices

Contains FCC ID: 2AA9B03

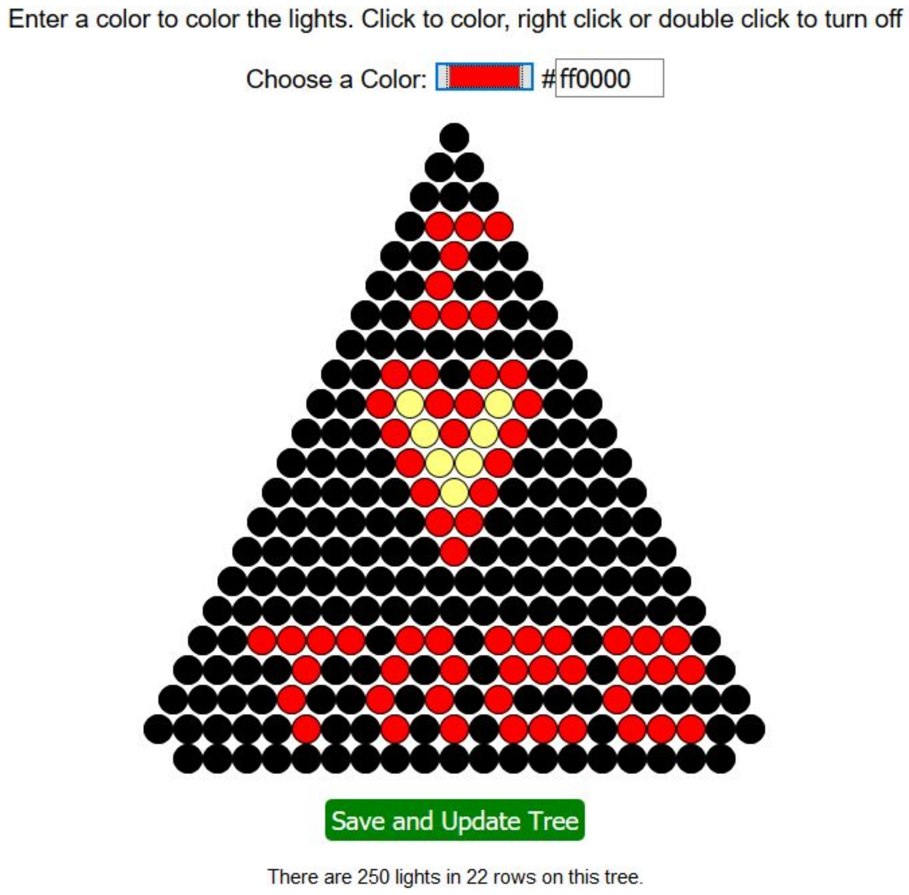
This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

[Figure 7]

Proposed Design

User Interface

The user interface will be a user-friendly web application hosted on the control Raspberry Pi using Apache HTTP/PHP that will transfer post data to a Python script. Having it be a web-app allows the user to access it from any device with an internet browser of their choosing. We will use WebSockets to have the server push events to the control Raspberry Pi to control the lights. On the user interface, the user can either upload images files to be texture mapped to the LEDs on the tree or design a basic pattern to be displayed such as a rotating rainbow or other select colors.



Enter a color to color the lights. Click to color, right click or double click to turn off

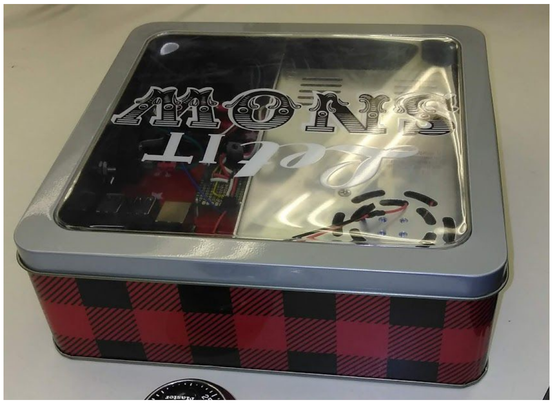
Choose a Color: 🟥 # ff0000

There are 250 lights in 22 rows on this tree.

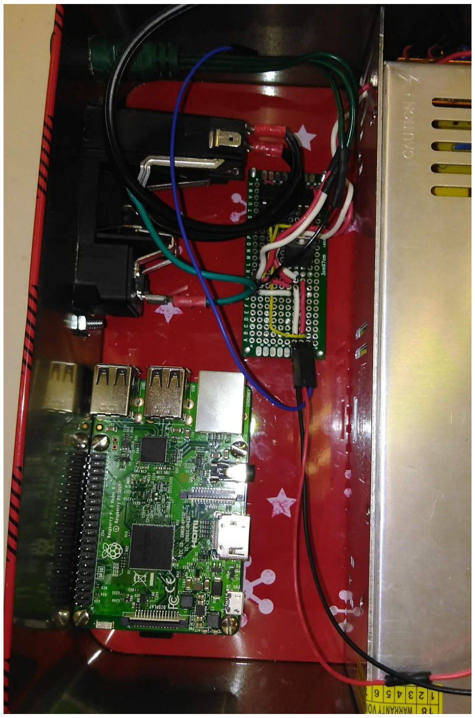
[Figure 8]

Lights and Control/Powerbox

The power and control box contains a 12v power supply, a 3v to 5v step up, a 12v to 5v voltage regulator and a Raspberry Pi. Both the 3v to 5v step up converter and 12v to 5v voltage regulator are soldered onto the same piece of perfboard to conserve space. The Raspberry Pi will have an attached LCD screen to display the IP address and other important data from the Raspberry Pi. This will be useful for the user to connect to the system without an additional display. The lights utilize the WS2811 LED chipset. Python files will control the Raspberry Pi’s PWM pin that is connected to the WS2811 LEDs to send data. The power and control box is housed in a holiday-themed metal box selected by the client. Further support will be added to aid in the structure especially on the side of the plugs. The structure of the case is too flimsy by itself to support repetitive plugging and unplugging of the power cable and light strand. Screws have also been added to the base so that the modules inside do not move around and potentially be damaged.



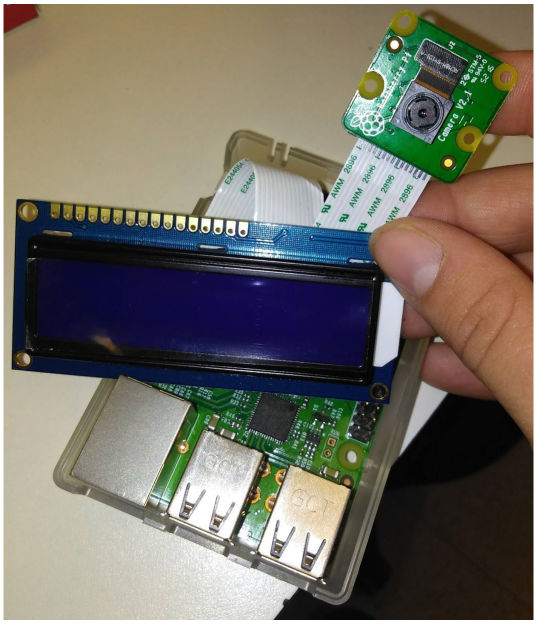
[Figure 9]



[Figure 10]

Calibration Device and Process

The calibration device is composed of a Raspberry Pi and Raspberry Pi camera housed in a 3D printed case. It collects images to be used for processing and sends them to the control Pi via a samba file share. The control Pi sends a command via SSH to instruct the Camera Pi to take a picture. The calibration device will have an attached LCD screen that displays the IP address of the calibration Raspberry Pi. The camera Pi will communicate with the LCD screen via the I2C protocol (NXP, 2014). This will be very useful for the user to connect the calibration system to the control system without needed to set up their own display.



[Figure 11]

The calibration process starts with the control system lighting up the light that is at the end of the strand that the user was instructed to place at the highest point on the tree. The calibrator then rotates the tree and takes pictures until the system is able to identify which view of the tree the top light is most visible from. It then lights pairs of proceeding lights purple and red until it identifies the next highest lights on the left and right edges of this view. It uses both purple and red at the same time because we can easily filter by color in software and these colors produce the least amount of noise. This allows us to process information about two lights with one picture. It repeats in this manner until all the lights that make up the left and right edges of the tree within the starting view are identified. The position of the remaining lights that are within view are then determined by their numerical position on a given strand. The tree is then rotated to the next view such that the lights that define the left edge of the previous face compose the right edge of the new face/view (i.e. 180 degrees). This ensures the system will learn how each light is mapped to the object in an efficient fashion.

Assessment of Proposed Solution

The user interface of the design allows the user to use their preferred web-browser and web-enabled device. The user will appreciate this as often members of one household have different brands/devices for web access. Such as one person could have an iPhone, another an android device, and a third may only use a desktop. Our design allows everyone to bring their own device and not go through the hassle of using an unfamiliar device.

The process for calibration fits our goal of being scalable and independent of the number of lights, allowing the customer to use light strands of varying lengths. However, the calibration method may be touchy with regards to lighting and window refraction in the calibration location. Particularly with depth perception and identifying if a light is on the target view or peeking in from the backside. To counteract this the user will be given instructions to modify the light conditions of the system to be more preferable. A working Internet connection will also be needed for both the system and the calibration/camera device.

We include the power source of the system to make sure the system is as streamlined as possible so the user will not have to fret with a multitude of cords needed to be plugged in. Having a few cords makes the system easy to set-up for the user. This is part of our goal of the end product being useable regardless of the user's technological background.

Displaying the IP of the devices helps us achieve of goal of being user-friendly by making the URL for the user to access for UI easily visible to anyone regardless of technological ability.

Our choice of using a web-app allows the flexibility for the design-chooser to be able to use any device with an internet connection. This means they can also access the design selection interface from any location as well, especially because the system does not have to be on to access the web server.

Technology Considerations

Most of the technology used in the project was passed down to us from the previous group. The hardware technologies we are currently using are two Raspberry Pi Bs, a Pi camera, a PWM light strand. The softwares in use or will be used are; OpenCV, NeoPixel library by Adafruit, Pi camera API, and PHP/HTTP server.

The previous group passed down a PWM light strand for us, we were given the requirement that we have to use this strand because of the original price paid for it. Therefore, we are using it. We were also passed down a Pi B, this is a fine choice for our project because of the open source software, size, and the nearly limitless options we can pursue to solve our problem.

We also choose to purchase another Pi B and Pi camera. This second Pi will serve as our calibration system. We choose to use a second Pi instead of a phone app (like the previous group) for a few reasons. First, an app is a terrible idea. Specifically, because you have to develop an app for a specific OS (apple or android). To maintain maximum usability we would need to develop two apps to run on all phones or require the user to have a phone with the OS we designate. But even if we ignore the OS issues, using a phone app would mean that when doing light calibration we would be relying on the phone hardware to do the computation or if we only use the phone as a camera we then could run into issues with people that have broken cameras or poor quality. By including a second Pi with a camera in the system we solve all of these problems at once.

The other consideration we had to evaluate is our calibration method. This is still in the early stages for our current project but our current idea is to use OpenCV software to analyze and interpret the image data received from the Pi camera into light positions on the object. We have settled on this software because it is free and has the option to be executed from a command line. We want the calibration process to work completely on its own and we believe OpenCV can do that for us.

Some previous ideas we had come up with were using two cameras to perform parallax calculations. We decided against this idea because it increased the number of cameras required and could have some mathematical difficulties because it would require exact positioning of the two cameras to perform the calculations.

Security Considerations

The system needs to be secured via a strong password such that it won't be rudimentary for a potential hacker to break into the system. We also need to consider locking down the system's communication to only listen to incoming traffic on certain devices or solely on the network that the system is on. Finally, permissions to edit aspects of the customer's systems, such as files on each respective Pi, will be locked to an admin account with randomized password that the user may change.

Safety Considerations

There are a few safety concerns to consider with this project. Shock, choking hazards, fire, and security are all possible areas of concern. Because our system is plugged into a wall socket, there is a chance of electrical shock. Beyond that, if some wiring is faulty or the voltage is not properly controlled there is a risk of fire. When dealing with any length of cord-like material there is always a risk of, tripping or strangulation. Finally, when any device is connected to the internet there is always the chance that the system could be hacked for nefarious purposes. This could be something harmless like someone changing how the lights look to something more severe like an attacker using the Pi as a hub to gain access to the users private WiFi network.

Possible Risks and Risk Management

Title: Inability to meet on Schedule
Risk: Accept
Information: Due to conflicts within our personal schedules, some members may not be able to make it to scheduled meetings. We can resolve some of these issues by having a written scribe to take note of what happened in the meetings to pass along the information. The scribe would also be responsible to communicate what other questions that needed to be answered. We also have the ability to communicate via online services.
Mitigation Action: Utilize free tools such as doodle polls to determine team eligibility, post meeting notes, and communicate through online channels

Title: Calibration Inaccuracies
Risk: Mitigate
Information: The tree has several lights winded around it, and it won't always be wired the same way. We're assuming that the client is able to use their camera, properly coordinate and calibrate the signaling lights so that the web-server can properly interpret any images based on the calibration.
Mitigation Action: Include in the user manual the proper procedure for calibration to minimize the noise in the images as well as proper hanging techniques.

Title: Inexperience with Python
Risk: Accept
Information: The Software Development Team is working over the previous team's development, which used Python for a majority of their project. Our team does not have a strong familiarity with Python; however, we can use C/C++ to call upon their Python Code.
Mitigation Action: As we continue to work through the project and take further courses our knowledge of python will increase

Title: OpenCV
Risk: Accept
Information: None of us have experience with OpenCV. This will most likely be a steep learning curve when designing the calibration process. There exists a lot of documentation on OpenCV and tutorials with examples of how to use it.
Mitigation Action: Ensure we have ample time to struggle through the processes of learning what OpenCV can do for us

Project Proposed Milestones and Evaluation Criteria

The major milestones of our project are as follows:

1. Turning on the lights using the power distribution system/control box housing we developed

2. Displaying a pattern that the user input on the user-Interface (i.e. alternate rows of green and Red.
3. Displaying an image that the user inputted on the user-interface.
4. Testing with users similar to our intended users(non-ECpE/SE background) and having them set-up and use the system.

Milestone One

This milestone is important because it requires us to have the power distribution design complete and tested, and familiar with the libraries and software needed for the lights to work. Without this milestone complete we would be unable to fully test most of the software and networking aspects. Testing and debugging will be the most difficult portion of this milestone as bad solder joint etc. can take weeks to reveal themselves. We reached this milestone in mid-April 2019.

Milestone Two

This milestone is important because it requires us to have the user-interface functional so the user can access it. Having the user-interface complete and communicating with the lights. The portion of this milestone that we are anticipating as the most difficult is implementing the interfacing between the light controlling code and the user interface. It will be a real test of our design choices. We expect to hit this milestone in early September 2019.

Milestone Three

This milestone is important because it requires us to have the image processing complete and functional, as well as the calibration system. We are anticipating these portions being the most difficult and time-consuming portions of our project so we have given ourselves until late October 2019 to reach this milestone.

Milestone Four

This milestone is important because it is necessary to know how the target user groups react to using our system, and their feelings about the user experience. This Milestone, we expect to hit early November 2019.

Project Tracking Procedures

We added progress reminders to our team's slack channel to keep us always aware of the deadlines we set, keep us on track, and hold us accountable. We will be dutiful in documenting our progress with pictures of our progress, updates to the Gitlab repository, weekly reports, and the Gantt chart.

Expected Results and Validation

The end goal is to have a fully-functioning and fully customizable light display. A new user should be able to lay out the lights on their tree and be able to perform the chosen calibration method. The user then should be able to upload their desired pattern to the tree through the use of the user interface.

The Raspberry Pi will be used for communicating the calibration information back to the web server. The image processing will then be handled on the web server where it will determine the position of each LED. Once the positions of the LED are learned, the web server will then map the desired pattern to the corresponding LEDs. The web server will then communicate this to the Raspberry Pi which will then control the LEDs.

Test Plan

FR.1: The system shall control individual RGB LEDs to display user-defined patterns

Test Case: This requirement is simply verified by using the raspberry pi to turn on various LEDs with various colors.

Test Steps:

1. Ensure the device is correctly installed
2. Load test_pattern.py
3. Ensure the LEDs on the tree are lit up sequentially

Expected Results:

It is expected that the python script will light up each LED with various coloring.

FR.2: The system shall utilize a calibration algorithm to determine the three-dimensional location of each individual LED

Test Case:

Test Steps:

1. Ensure the device is correctly installed
2. Begin calibration algorithm
3. Check the number of LEDs found and their general distribution

Expected Results:

It is expected that the calibration algorithm will find the correct number of LEDs. It is also expected that the precise location of each LED will be within a margin of error small enough to be negligible to the user when displaying a pattern or image.

FR.3: The system shall be capable of operating for several hours at a time without overheating or posing any other safety threats

Test Case: This requirement is verified by allowing the box to run for 24 hours while monitoring its temperature. It is expected the product will never run for a continuous 24 hours in consumer use, therefore this test will more than satisfy this requirement.

Test Steps:

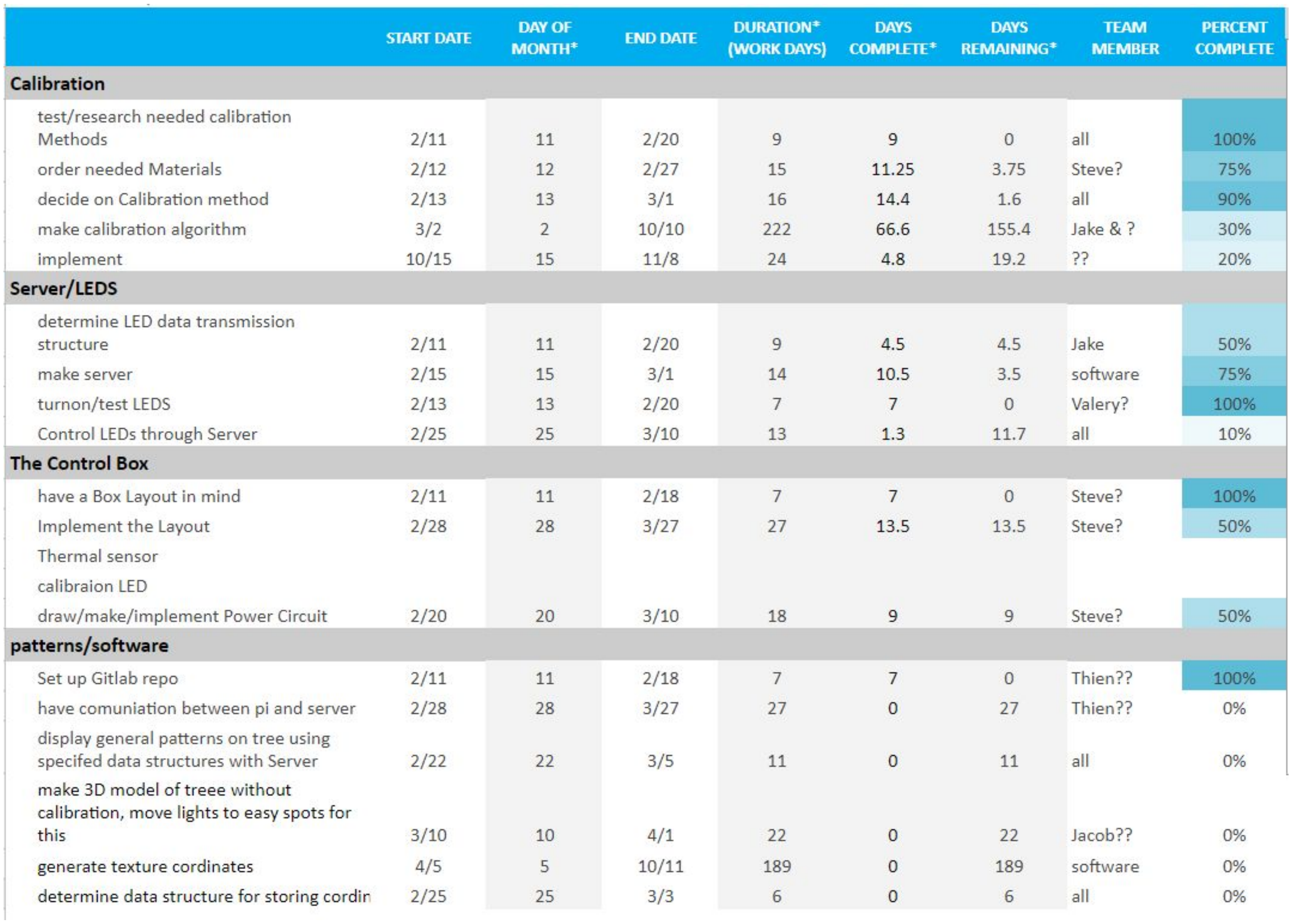
1. Ensure the device is correctly installed
2. Begin running chosen test pattern script
3. Allow lights to be on for 24 hours while monitoring temperature and performance

Expected Results:

It is expected that the box will increase in temperature with time but will not overheat or cause

Project Timeline, Estimated Resources, and Challenges

Project Timeline



	START DATE	DAY OF MONTH*	END DATE	DURATION* (WORK DAYS)	DAYS COMPLETE*	DAYS REMAINING*	TEAM MEMBER	PERCENT COMPLETE
Calibration								
test/research needed calibration Methods	2/11	11	2/20	9	9	0	all	100%
order needed Materials	2/12	12	2/27	15	11.25	3.75	Steve?	75%
decide on Calibration method	2/13	13	3/1	16	14.4	1.6	all	90%
make calibration algorithm	3/2	2	10/10	222	66.6	155.4	Jake & ?	30%
implement	10/15	15	11/8	24	4.8	19.2	??	20%
Server/LEDS								
determine LED data transmission structure	2/11	11	2/20	9	4.5	4.5	Jake	50%
make server	2/15	15	3/1	14	10.5	3.5	software	75%
turnon/test LEDS	2/13	13	2/20	7	7	0	Valery?	100%
Control LEDs through Server	2/25	25	3/10	13	1.3	11.7	all	10%
The Control Box								
have a Box Layout in mind	2/11	11	2/18	7	7	0	Steve?	100%
Implement the Layout	2/28	28	3/27	27	13.5	13.5	Steve?	50%
Thermal sensor								
calibraion LED								
draw/make/implement Power Circuit	2/20	20	3/10	18	9	9	Steve?	50%
patterns/software								
Set up Gitlab repo	2/11	11	2/18	7	7	0	Thien??	100%
have comuniation between pi and server	2/28	28	3/27	27	0	27	Thien??	0%
display general patterns on tree using specifed data structures with Server	2/22	22	3/5	11	0	11	all	0%
make 3D model of treee without calibration, move lights to easy spots for this	3/10	10	4/1	22	0	22	Jacob??	0%
generate texture cordinates	4/5	5	10/11	189	0	189	software	0%
determine data structure for storing cordin	2/25	25	3/3	6	0	6	all	0%

[Figure 12]

We divided the project into several blocks, Calibration, Server/LEDs/ Physical Control Box, and Patterns Software as shown in our Gantt Chart above. Each block has steps that can be implemented in parallel with the steps in other blocks., save for the final integration portion. This prevents the project from getting bottlenecked around any one person or sub-groups' work. Having our project displayed in a Gantt chart allows us to easily see which parts of the project are progressing and if any need added attention. It is also very easily modifiable so we can add new tasks/milestones as our design choices create them.

The design portions of the project have all been scheduled to be completed in the first semester of senior design. The data structure that the LED update data will be transmitted in was decided and evaluated on February 20th. The calibration method was determined and evaluated on February 27th. Our control box circuits and layout was decided and evaluated on. Our server framework was determined and evaluated on. Completing and evaluating the design decisions in the first semester allows us to validate our designs and choices before they are implemented next semester, and be prepared and aware of any issues we may encounter in integration.

Feasibility Assessment

We believe this project is completely feasible. Picking up from the previous group we need to determine a concrete method of LED detection and then an accurate system to map images to the LEDs. When calibrating the LED positions the tree will have to remain steady as well as the camera. Issues that may occur are seeing an LED on the other side of the tree through a gap in a tree branch. Also depending on the tree, if it is made of plastic, the light may reflect off its “needles”. Another challenge in displaying images and text on the tree is that the density of lights will need to be very high to provide the highest “pixel” density to be able to have clearly defined images.

Personnel Effort Requirements

Further hardware development will be necessary from the hardware team including reinforcing the metal box, isolating power and data lines, affixing the LCD screen, providing adequate airflow for cooling, and maintaining the overall aesthetic of the devices.

Further software development will be needed for creating a usable web interface, an image capturing device, and the light calibration process. The light calibration process will require a tremendous amount of research and testing to ensure the process works in a multitude of environments.

Other Resource Requirements

Other physical requirements include a 3D printed case for the Camera Pi, a spare light strand in the event of an accident and to test scalability of our design. Overall, we will require at least three more months in order to complete, this lines up with the time left in the next semester and is verified by our timeline.

Financial Requirements

There will be very few additional items to purchase as a significant amount of materials were inherited.. Other monetary requirements are the internal Raspberry Pi power supply and spare light strands in the event of an accident damaging the equipment. It would be possible to include the original Raspberry Pi power supply within the container. However, that adds significant complexity and removes vital space from the container which could otherwise be used for ventilation or otherwise. All software will be free and/or open-source. The total price of the project can be found in the Bill of Materials [Table 1].

Closure Materials

Closing Summary

Our project is to create a dynamic, fully-customizable LED tree light display, a vast improvement over today's standard, static light displays. Our group has made the first steps to accomplishing our goals for this project. As we continue to work through the progress already done by a previous team and learn more about it, we can make the best decisions to improve and complete this project. The web server on the Raspberry Pi is already up and running, the LEDs can be controlled through the Pi and the Pi's web interface, and the rest of the required hardware including the power supply has been obtained. The team will implement a method of calibration that will allow for the highest accuracy on light positions. This information will then be used for a system to map textures and patterns to LEDs to generate the images on the tree.

References

[1] “What is a Raspberry Pi?,” *Raspberry Pi*. [Online]. Available: https://www.raspberryPi.org/help/what- is-a-raspberry-Pi/. [Accessed: 20-Apr-2019].

[2] “Learn About LED Lighting,” Learn About LED lights | ENERGY STAR. [Online]. Available: https://www.energystar.gov/products/lighting_fans/light_bulbs/learn_about_led_bulbs. [Accessed: 20-Apr-2019].

[3] “Home,” – *Secure Remote Login and File Transfer* | *SSH.COM*. [Online]. Available: https://www.ssh.com/ssh/protocol/. [Accessed: 20-Apr-2019].

[4] Info@twinkly.com, “Homepage,” Twinkly, 13-Nov-2018. [Online]. Available: https://www.twinkly.com/. [Accessed: 20-Apr-2019].

[5] “Home,” Lumenplay®. [Online]. Available: https://lumenplay.com/. [Accessed: 20-Apr-2019].

[6] “About,” Lumenplay®. [Online]. Available: https://lumenplay.com/about/. [Accessed:20-Apr-2019].

[7] “802.11-2012 - IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” IEEE. [Online]. Available: https://standards.ieee.org/standard/802_11-2012.html. [Accessed: 24-Apr-2019].

[8] R. Altkorn, S. Milkovich, and G. Rider, Light emitting diode safety and safety standards - IEEE Conference Publication. [Online]. Available: https://ieeexplore.ieee.org/document/1529515. [Accessed: 24-Apr-2019].

[9] 1st Initial. , "IEC 60320-1 ," *IEC* , Vol. , no. , pp. , 2015.[Standard]. https://webstore.iec.ch/publication/63846. [Accessed 4/24/2019].

Appendices

Bill of Materials

Item	Price	Source	Notes
Metal Box	$0	Client	inherited
12V Power Supply	$18	Amazon	inherited
Raspberry Pi 3B	$35	Adafruit	inherited
Light Strands (x5)	$15	Amazon	inherited
3.3v-5v step up (74AHCT125)	$0.50	ETG	inherited
12V-5V step down (7805)	$0.50	ETG	Integrated chip #7805
Power plug and fused switch	$1.50	Amazon	Inherited
Raspberry Pi 3B	$35	ETG	For running setup
PiCam	$14	ETG	
Total Cost to Date	**$119.50**		

[Table 1]