Light Show Recording System

By:

Jake Tuohy

Joshua Schultheiss

Miguel Cabrera Del Mora

Final Report for ECE 445, Senior Design, Spring 2017

TA: Vignesh Sridhar

May 2017

Project No. 37

Abstract

The Lightshow Recording System allows for users to bring professional light shows to their apartment or living room. Our design takes a simple user interface and outputs the show to a comprehensive light array. By using a simple user interface, it allows consumers to create complex light shows, as you would see professionally, without having to go through the complex process of programming of the lights.

Table of Contents

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 ecosystem. Our goal is to design an array of lights using LEDs that can illuminate a room. Those lights will be a playable instrument by an interface similar to a soundboard. A web application will emulate a software version of this interface called a “light board.” Light show recordings can then be mixed through our software and shared with friends.

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. While controllable mono-color LED bulbs now sell for around $15, controllable multi-colored LED bulbs are still sold around $45[1,2]. 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.

Our product will be competing with the controllable multi-colored bulbs, but instead of just a single bulb being placed in an existing lighting socket, the user will be able to have a greater amount of lights in any area of their home. Our design has a base of 6 controllable multi-colored LED blocks, where each one is as bright as a single multi-colored LED bulb. This will mean that the user will have a greater amount different effects to use for a more professional looking personal light show.

2 Design

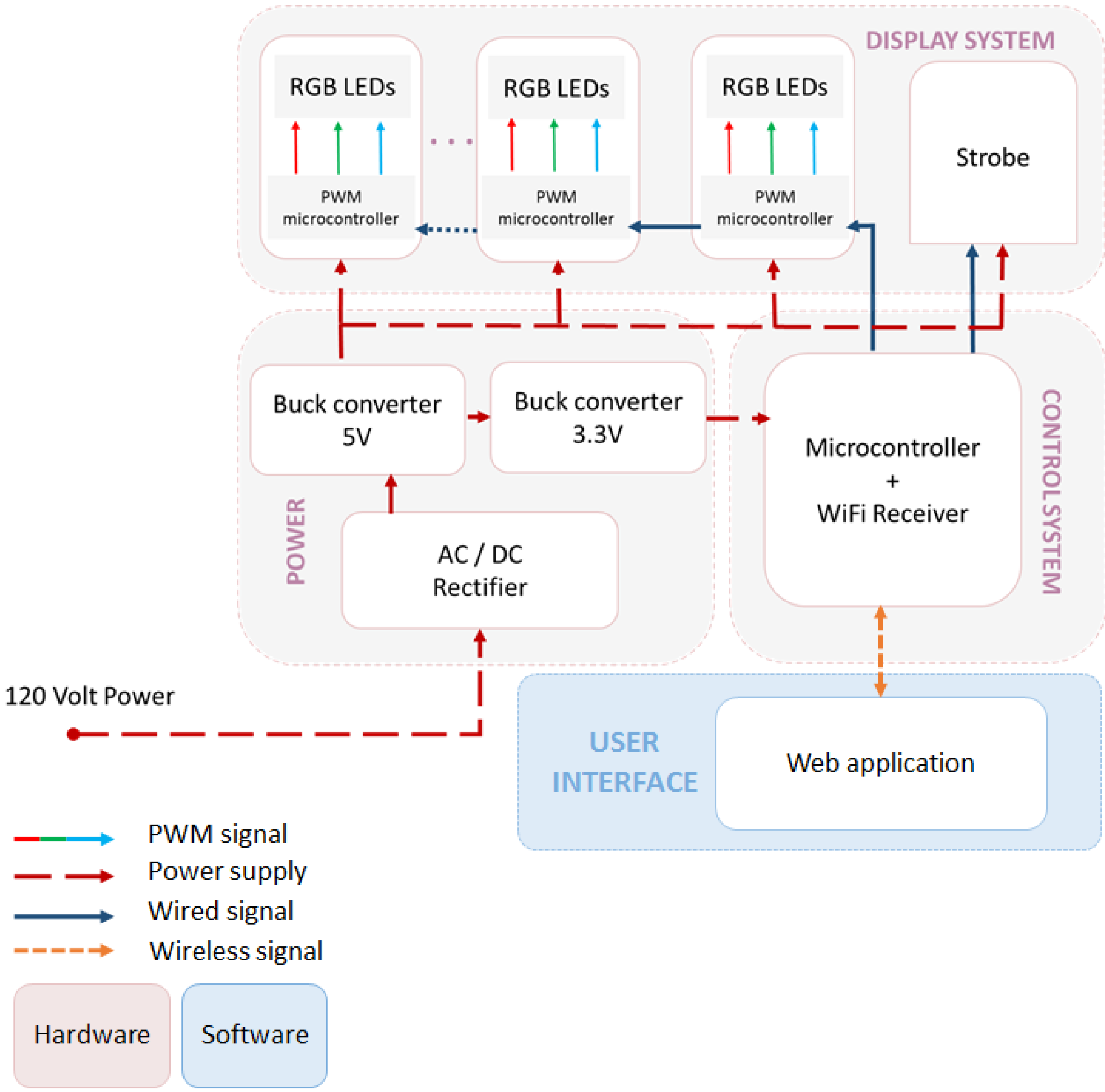

Figure 1: Block Diagram of the Hardware and Software agents of the project

The four main block portions of the project are the two software blocks, the web application and the control system, and the two hardware blocks, the power and display system. Each of these blocks are broken into more detail later in this report.

2.1 Web Application

The main application will be a web application that can be run on any platform. However, the design was inspired by the iPad since a large touch interface would be ideal to quickly press many virtual buttons. The grid of effects that the application contains is structured by rows. Each row of the application corresponds to a group of lights. For example, a row might correspond to every-other light. This flexible group definition allows for a setup with any number of lights. Buttons in the rows correspond to light-effect functions. Each light effect executes three functions for the red, green, and blue channels of the LEDs and lasts for a given transition time. Table 1 contains the variables that can be used in the effect functions.

Table 1: Light Functions API

curr	Index of light being addressed
L	Total number of strings
LIGHTS[curr]	Object containing light data
LIGHTS[curr].t	Time elapsed for function
LIGHTS[curr].r/g/b	Color values at start of function

For advanced effects, the application can make loops. For instance, if one effect turned the lights off and another turned the lights on, a user could loop these two effects in a short amount of time to simulate a strobe. Furthermore, each light holds two sets of effect functions: an “unlock” function and a “lock” function. The “unlock” function executes perpetually, and when the time for the function overshoots the transition time the function repeats. However, if a user triggers a “lock” function the light immediately switches to perform that effect. Once the “lock” function expires, the “lock” “falls off” and resumes displaying the “unlock” function.

2.2 Control System

2.2.1 ESP8266 Microcontroller

This is the main microcontroller that receives the input signals through the Wi-Fi. The WIFI Receiver receives the signals stored on the server that are input by the user, and communicating 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 area in 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.

The data used to update the state of the LEDs is sent from a web client. The data sent includes: brightness, color, transition time, and type of effect. An effect is a function that executes on the processor over a transition time while changing the brightness and color of the lights. That data outputs to each of the PIC16F1508 Microcontrollers for each LED block using SPI communication. The signal is 24-bits in length; 8-bits per red, green, and blue brightness levels.

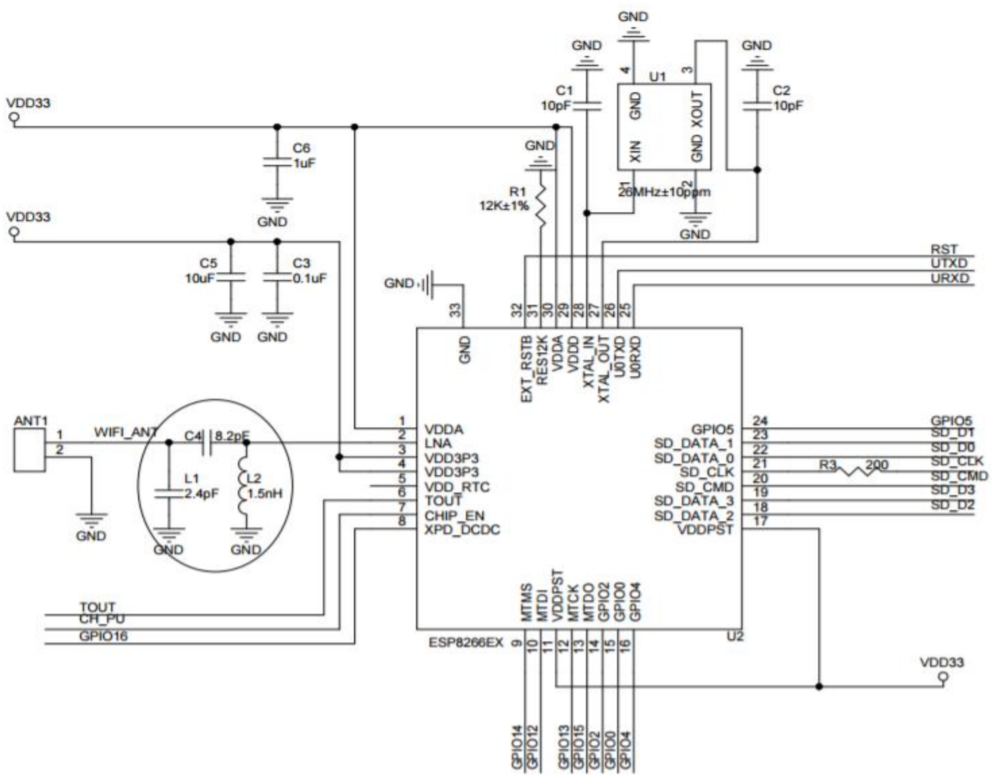

Figure 2: ESP8266 Circuit Diagram [6]

2.3 Power

2.3.1 5V Buck converter

The DC-DC buck converter takes in 9 volts DC from the AC/DC rectifier powered from the wall, and converts it to a 5V ± 5% DC voltage source. All components used in our project, with the exception of the ESP8266 microchip, are rated for a 5V input, so this will be the main voltage level used. In order to drive the PWM signal shown in figure 3 we use a TL5001 timing chip, which has an internal voltage regulator so it can be powered from the 9-volt source.

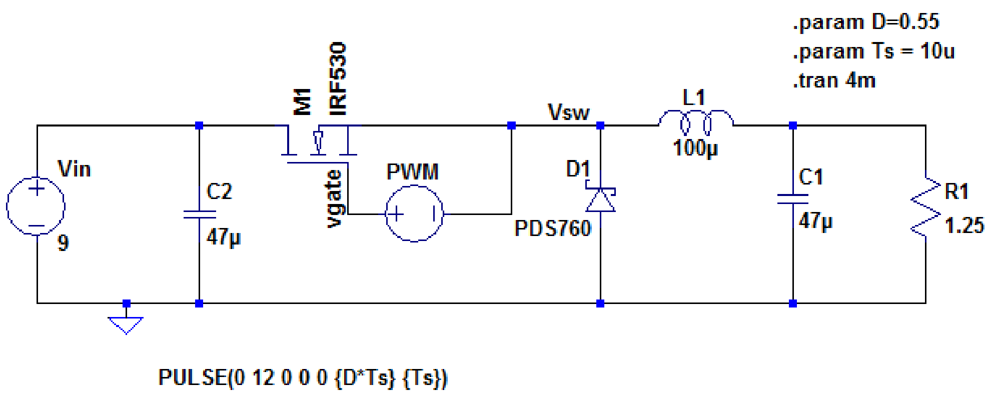

Figure 3: 9V to 5V buck converter schematics

2.3.2 3.3V Buck converter

The DC-DC buck converter takes the 5V DC voltage and converts it to a 3.3V ± 10% DC voltage source. This voltage level is only used to power the ESP8266 microchip. In order to drive the PWM signal shown in figure 4 we again use a TL5001 timing chip.

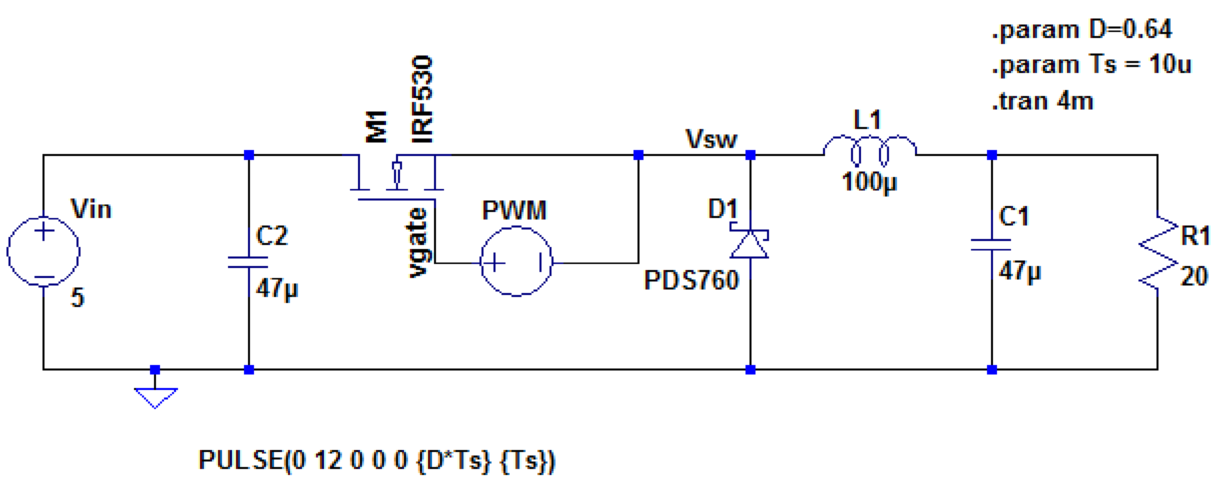

Figure 4: 5V to 3.3V buck converter schematics

2.4 Display System

2.4.1 PIC16F1508 Microcontroller

The PIC16F1508 microcontroller has 14 pins while our project only utilizes 9 of them, this is the smallest microcontroller to control up to 4 PWM modules, which of those 3 of them are used, one for each RGB color [12]. One of the main reasons this microcontroller fits our project is its compatibility with the SPI communication system between ICs. We use the SPI communication with a daisy chain configuration. This cascading communication between blocks allows the project to be scaled independently of the number of ports of the master microcontroller. One concern with the cascading communication use is a timing delay in the light signals reaching every block, as it can be seen from figure 6 and equation 2 this timing delay will not be an issue.

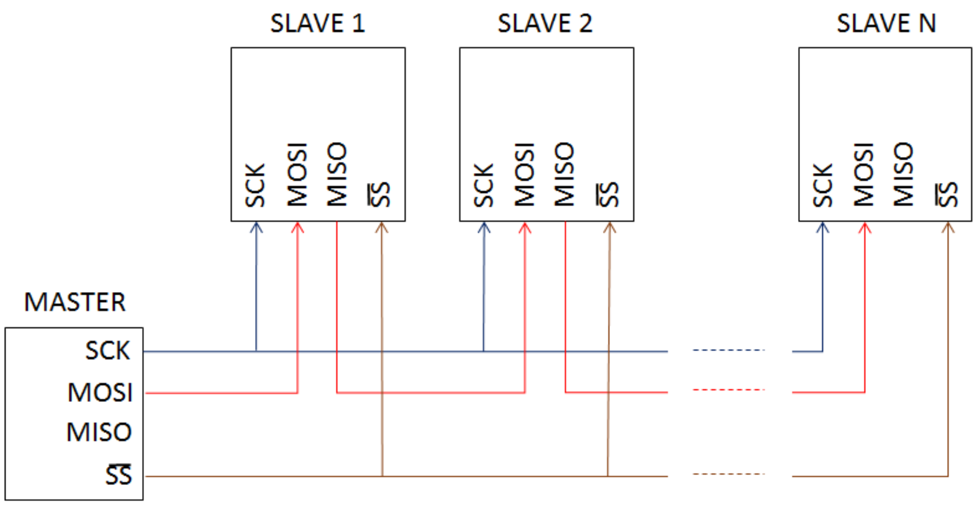

Figure 5: Daisy chain connection with a SPI bus.

In daisy-chaining what our microcontroller would do is register a single 8-bit command (8 bits for each RGB color) through DIN (Data In) during a given command-cycle (8 clock cycles), then the microcontroller would output the same command it has register through DOUT (Data Out) during the subsequent command-cycle. Because there is a DIN-to-DOUT delay of one command-cycle, our total delay would be equal to time of a command cycle multiplied by the number of blocks in the array. The slave, would execute the new command written to it on the rising edge of active-low SS (Slave Select) signal. This means that as long as active-low SS remains low, the slave would ignore the command and would output it to DOUT on the following command-cycle. If active-low SS goes high after a given command-cycle, all slaves execute the commands just written to their respective DIN inputs. If active-low SS goes high, data is not output at DOUT. This process makes it possible for every slave in the chain to execute a different command. The only problem with this protocol is that if you want to change only the value of one of the blocks you would have to update all the values in the array.

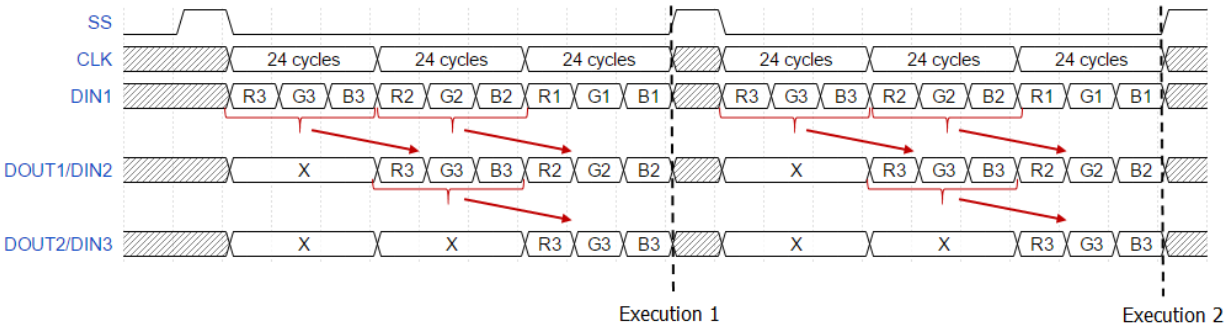

Figure 6: Time diagram daisy chain protocol

In the following equations, we calculated the total delay of our daisy chain communication protocol:

T_{delay}=Total delay n_{bits}=number of bits in one command N_{blocks}=number of blocks in the array

$$T_{delay}=(Clk_{frequency})^{-1}*n_{bits}*N_{blocks} \quad (1)$$
$$T_{delay} = 2*10-10*24*12=7.2\ \mu s \quad (2)$$

As this project is meant to be used in synchronization with music, we had to make sure the delay of the lights wound not affect the user's experience. The Advanced Television and Systems Committee says that the light delay should be no more than 45 milliseconds [10]. As shown in equation (2), the delay of our lights is well below this threshold.

2.4.2 RGB LEDs

Each block is composed of 6- 259RGBM5C-013 LEDs each of which are each powered by the 5V output of the buck converter. Each color is connected in parallel in order to use a single pull down resistor to ensure consistent current through each LED and in turn consistent colors. Because of the nonlinear nature of LED outputs once the value of 0-255 has been received it is linearized in the PICF1508 to create a smoother ramping of colors and a truer color scale.

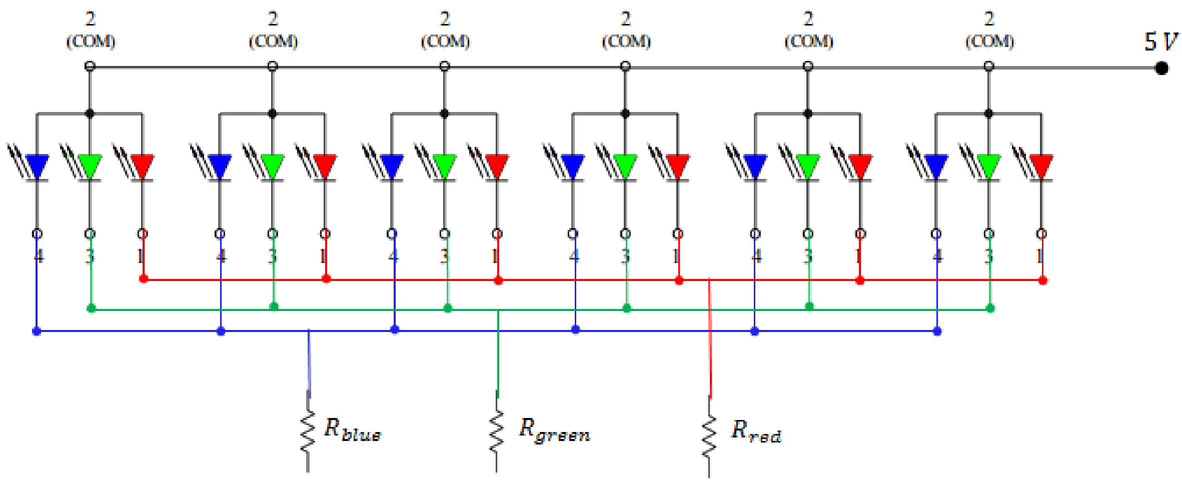

Figure 7: Partial schematic view of the connection of the LEDs

To calculate the value needed for each color resistor, we made use of the datasheet for the 259RGBM5C-013 LEDs. As we are controlling the LEDs brightness with Pulse Width Modulation, we determined the value of the resistors at full brightness when the duty ratio of the PWM is equal to 1, this means maximum forward current in each LED for a continuous mode, no pulse modulation. The values used for each of the following currents are the maximum values per the data sheet:

$$V_{color} = V_{input} - V_{forward} \quad (3)$$

$$I_{color} = (\text{\# of LEDs}) * I_{colormax} \quad (4)$$

$$R_{color} = V_{color}/I_{color} \quad (5)$$

Calculations for the red resistor

$$V_{Rred} = 5\ V - 2.1\ V = 2.9\ V \quad (6)$$

$$I_{Rred} = 6 * I_{LED\ red\ max} = 6 * 25\ mA = 150\ mA \quad (7)$$

$$R_{red} = V_{Rred} / I_{Rred} = 2.9\ V / 150\ mA = 19.33\ \Omega \approx 20\ \Omega \quad (8)$$

Calculations for green resistor:

$$V_{Rgreen} = 5\ V - 2.1\ V = 2.9\ V \quad (9)$$

$$I_{Rgreen} = 6 * I_{LED\ green\ max} = 6 * 25\ mA = 150\ mA \quad (10)$$

$$R_{green} = V_{Rgreen} / I_{Rgreen} = 2.9\ V / 150\ mA = 19.33\ \Omega \approx 20\ \Omega \quad (11)$$

Calculations for blue resistor:

$$V_{Rblue} = 5\ V – 3.3\ V = 1.7\ V \tag{12}$$

$$I_{Rblue} = 6 * I_{LED\ blue\ max} = 6 * 25\ mA = 150\ mA \tag{13}$$

$$R_{blue} = V_{Rblue} / I_{Rblue} = 1.7\ V / 150\ mA = 11.33\ \Omega \approx 12\ \Omega \tag{14}$$

Values are rounded up to give more available resistor values. Also since these are max values by increasing resistance slightly we go to a safer level below max.

Figure 8 shows the complete schematic circuit of 1 LED block. To control the sinking of the current with our microcontroller we use a BS170G small signal MOSFET, this MOSFET has a maximum drain current of 500mA and a maximum drain to source voltage of 60 V. These rated values are enough for our project as we work with 5V and sinking a maximum current of 450 mA. We attached to each MOSFET to a pulldown resistor of 10KΩ between the gate and the source, in order to create a truer on-off switch.

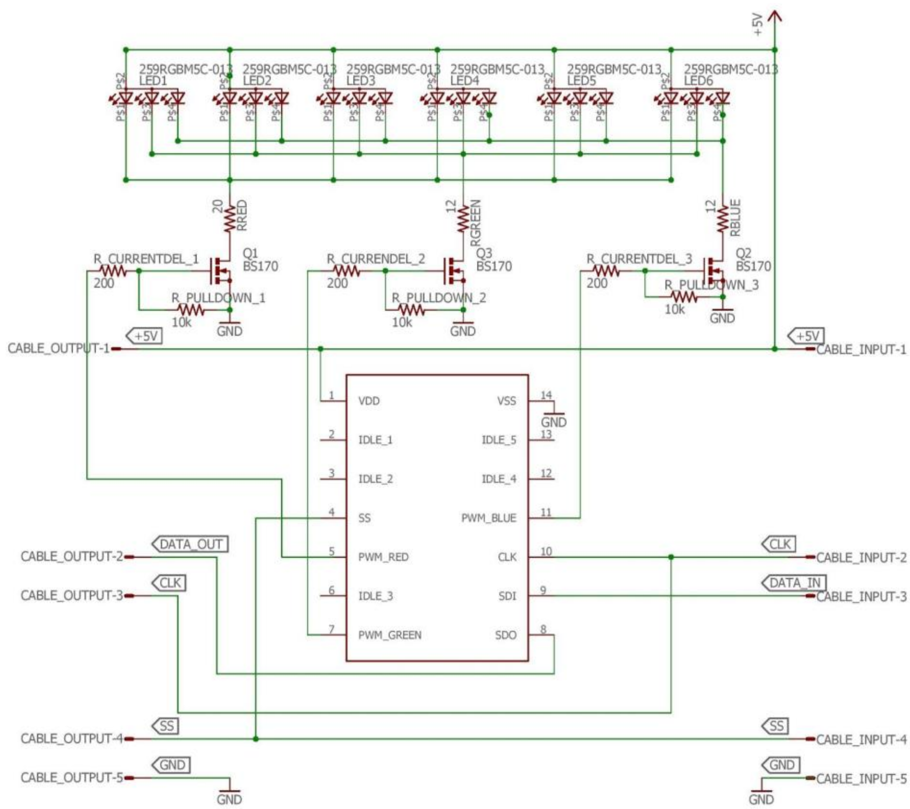

Figure 8: Complete schematic circuit of each LED block.

2.4.3 Strobe Light

The strobe light is composed of 6-FLR-50T04-HW7 LED which are controlled by the main microcontroller, the ESP8266. These LEDs are driven in a different manner because the ESP8266 has a logic high of 3.3 volts so it can be used for the switching and a MOSFET is not needed. Simply when the LEDs are to be turned on, a logic "1" is outputted from the ESP8266. A pulldown resistor is used to make sure maximum current does not exceed rated value.

Calculations for the white resistor using maximum rated current per the datasheet and equations 3,4 and 5:

$$V_{White} = 3.3\ V - 3\ V = .3\ V \tag{15}$$

$$I_{White} = 6 * I_{White} = 6 \cdot 20\ mA = 120\ mA \tag{16}$$

$$R_{White} = V_{RWhite} / I_{RWhite} = .3\ V / 120\ mA = 2.5\ \Omega \approx 3\ \Omega \tag{17}$$

3 Verifications

3.1 ESP8266 Microcontroller

All requirements were able to be verified for the ESP8266 microcontroller. The Wi-Fi requirements was able to be verified in a simple manner by sending a signal from the web application and have be outputted to the LEDs. Second the speed requirement of the ESP8266 was verified by sending 1440 color signals and receiving all signals back through the daisy chaining loop in under one second, the total time was roughly .2 seconds because it was able to run at around 400 Hertz after adding a small delay after each signal in order to increase signal reliability, which is well above what the human eye can detect. Lastly, signal reliability was tested by sending 1000 signals and seeing what error was received, which was on average 2/1000, which would not be able to be seen by the human eye.

3.2 5V Buck converter

Because we only had 6 LED blocks for our demonstration we were unable to test at what we decided our maximum expandable load would be of 6 Amps. When testing the 5-volt buck converter at our maximum load, which consisted of all lights fully on white and the strobe light completely illuminated, our maximum voltage level measured was 4.86 volts and a minimum value of 5.22 volts, which is completely in our 5% percent threshold. The buck converter was more stable at full load while at smaller load still did not reach high or lower than the max values stated but did experience a higher amount of fluctuation.

3.3 3.3V Buck converter

The 3.3 Volt buck converter only had to be able to sustain the load of the main micro controller and the strobe light LEDs. While measuring minimum and maximum values for this buck converter the range stayed within 3.20-3.46 which is within the 10% threshold. This threshold was maintained with the 200 mA draw of the ESP8266 microcontroller.

3.4 PIC16F1508 Microcontroller

Because of the daisy-chain setup of our microcontrollers the timing for this microcontroller was able to be tested using the same process as the ESP8266 microcontroller. Each PIC16F1508 was able to process the RGB values to an output faster than the receiving of a new value, therefore the colors were also outputted at roughly a speed of 400 hertz, which again is faster than the human eye can detect. The second requirement was not completely satisfied because of the inability to achieve a true “off” position in the MOSFET. Different PWM cycles corresponding to 25-255 were able to be achieved.

3.5 RGB LEDs

Because of testing restrictions, we were not able to completely test the brightness levels of each of the LED blocks, but based on data sheets by using six of the LEDs chosen we were able to achieve the required lumen values for each of the colors. For the second requirement, using our calculations shown in the design above we were able to calculate the resistance value needed to supply less than rated current through each LED. When calculating we used a max value of 25 mA through each LED and when testing the highest value recorded was 24 mA which ensures safety from burning out LEDs.

3.6 Strobe Light

Because of testing restrictions, we were not able to completely test the brightness levels of the strobe light block, but based on data sheets by using six of the high intensity LEDs chosen we were able to achieve the required lumen values for each of the colors. For the second requirement, using our calculations shown in the design above we were able to calculate the resistance value needed to supply less than rated current through each LED. When calculating we used a max value of 20 mA through each LED and when testing the highest value recorded was 18 mA which ensures safety from burning out LEDs.

4 Cost

Table 2: Cost

Item	**Quantity**	**Cost (Unit)**	**Cost (Total)**
ESP8266 Microcontroller	1	$6.95	$6.95
7-PIC16F1508 Microcontrollers	6	$1.36	$8.16
TL 5001 Timing chip	2	$.88	$1.76
36 - 259RGBM5C-013 LEDs	36	$.66	$23.67
6- FLR-50T04-HW7 LEDs	6	$.28	$1.68
Plastic housing	1	$25.00	$25.00
Assorted resistors, capacitors, transistors, inductors	x	$10.00	$10.00
LED PCBs	6	$1.00	$6.00
Buck Converter PCBs	2	$1.00	$2.00
Labor	300	$30*2.5	$22,500
Total Cost			$22,586.34

5 Conclusion

Because the LED light show is a customizable product there is still more progression that could be made in order to finalize this product. One main thing is making the product more visually appealing, as the final product turned out bulkier than our original design. In the future of production working with a different manufacturer could greatly help in making a sleeker design. More issues that can be looked at in our project but cannot be changed is the safety and ethics issues involved with our project, which will be looked at a little deeper.

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 too 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 5-30 Hz, but because this is the most prominent range that the lights will be used at, there will 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 [5].

From an ethics standpoint, the above risk of seizures could be in violation of the IEEE ethics code number 9 stating that we should not injure others [7]. We believe that proper labeling and warning will a way to avoid breaking this ethical code as we will do our best to let people know the risks so they can ultimately make the decision to use the product. Another thing we have to be careful is the rights to the music being played as going along with ACM code 1.6[8]. For this reason, in the production of this project, 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. This would be a way of complying to code 1.6 in effort to protect others intellectual property.

References

[1] "Philips - hue A19 Smart LED Light Bulb - White Only,". [Online]. Available: http://www.bestbuy.com/site/philips-hue-a19-smart-led-light-bulb-white-only/4374300.p?skuId=4374300&ref=212&loc=1&ksid=8ab71c6b-584c-449e-91fe-5aa715945c01&ksprof_id=401&ksaffcode=pg223111&ksdevice=c&lsft=ref:212,loc:2&gclid=CjwKEAiArbrFBRDL4Oiz97GP2nISJAAmJMFaE4BEC9BEsNTJs-axZquFs-1TWUbo_NtutxcwtLLMpRoCsLPw_wcB. Accessed: Feb. 24, 2017.

[2] "Wi-Fi multi-color smart LED bulb, 2-Pack," Walmart.com, 2017. [Online]. Available: https://www.walmart.com/ip/WiFi-Multi-Color-Smart-LED-Bulb-2-Pack/50820255?wmlspartner=wlpa&selectedSellerId=0&adid=22222222227039581096&wl0=&wl1=g&wl2=c&wl3=85303117610&wl4=pla-193197925970&wl5=9022196&wl6=&wl7=&wl8=&wl9=pla&wl10=8175035&wl11=online&wl12=50820255&wl13=&veh=sem. Accessed: Feb. 24, 2017.

[3] "Home - Mongo express angular Node," Mongo Express Angular Node, 2014. [Online]. Available: http://mean.io/. Accessed: Feb. 24, 2017.

[4] IncDigitalOcean™, "DigitalOcean: Cloud computing designed for developers," DigitalOcean, 2017. [Online]. Available: https://www.digitalocean.com/. Accessed: Feb. 24, 2017.

[5] P. Shafer and J. Sirven, "Photosensitivity and seizures," Epilepsy Foundation, 2013. [Online]. Available: http://www.epilepsy.com/learn/triggers-seizures/photosensitivity-and-seizures. Accessed: Feb. 23, 2017.

[6] "File: ESP8266 specifications English.pdf," 2014. [Online]. Available: https://nurdspace.nl/File:ESP8266_Specifications_English.pdf. Accessed: Feb. 23, 2017

[7] "IEEE IEEE code of ethics," 2017. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. Accessed: Feb. 23, 2017.

[8] "ACM code of ethics and professional conduct," 2017. [Online]. Available: https://www.acm.org/about-acm/acm-code-of-ethics-and-professional-conduct. Accessed: Feb. 23, 2017.

[9] SRC: https://i.ytimg.com/vi/rc7oV5PVxpw/hqdefault.jpg

[10] R. Hoffner, "ATSC recommends AV sync rules," TV Technology, 2015. [Online]. Available: http://www.tvtechnology.com/opinions/0004/atsc-recommends-av-sync-rules/184350. Accessed: Feb. 24, 2017.

[11] Adafruit. [Online]. Available: https://cdn-shop.adafruit.com/product-files/2739/p2739.pdf. Accessed: Feb. 24, 2017.

[12] Microchip. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/41440B.pdf. Accessed: Feb. 24, 2017.

Appendix A - Requirements and Verification Table

ESP8266 Microcontroller

Requirements	Verifications	Points
1) Pair to Wi-Fi network 2) Generate digital signals at a minimum rate of 60 Hz 3) Output data to slave controllers with error rate of less than 1%	1) Connected to the internet 2) Speed Check a) Connect microcontroller to computer b) Send 600 commands c) Measure total time to be under 10 seconds (see no flicker) 3) Error Rate a) Connect microcontroller to computer b) Send 1000 commands c) At least 990 correct commands are received	**10**

5V Buck converter

Requirements	Verifications	Points
1) Transform 9V to 5V ± 5% 2) Output current has ripple of less than 5% 3) Handle current draw of up to 6 amps	1) Monitor voltage output over standard load shown in figure 3 to make sure voltage varies within 5% of 5 volts 2) Monitor voltage output over standard load shown in figure 3 to make sure current varies within 5% 3) Measure current at maximum load of all LEDs outputting white light	**10**

3.3V Buck converter

Requirements	Verifications	Points
1) Transform 5V 3.3V ± 10% 2) Handle current draw of 200 milli-amp	1) Monitor output voltage make sure voltage varies within 10% of 3 volts 2) Measure current output with a load of the microcontroller	**10**

PIC16F1503 Microcontroller

Requirements	Verifications	Points
1) Receive, process and output signals to LEDs at a rate of 100 Hz 2) 255 PWM different duty ratio configurations from 0% to a 100% in each of the pins with a switching frequency of 1 kHz	1) Timing a) Send 1000 repetitive signals b) Measure total output time of LEDs being on c) Divide total time by 1000 to calculate rate 2) Use a multimeter to monitor the PWM signal at the pins, check the frequency and duty ratio for a sample of 25, randomly chosen colors. Acceptable error of 5%	**5**

RGB LEDs

Requirements	Verifications	Points
1) Luminosity of 2000 mcd ± 10% for red color at maximum current, binary code value (255,0,0), 4000 mcd ± 10% for green (0,255,0), and 1500 ± 10% mcd for blue (0,0,255). 2) Maximum current through each LED of 25 mA working at full brightness.	1) Brightness a) Send max binary value (255) for each color to microcontroller b) Measure mcd value with lumen meter 2) Current a) Send maximum values for all colors (255,255,255) b) Measure current through LED using ammeter	**10**

Strobe Light

Requirements	Verifications	Points
1) Luminosity of 4000 mcd ± 10% for all colors at maximum current, giving white light 2) Maximum current through each LED of 20 mA working at full brightness.	1) Brightness a) Send high voltage signal to strobe b) Measure mcd value with lumen meter 2) Current a) Send high voltage signal to strobe b) Measure current through LED using ammeter	**5**

Appendix B - Enclosure Models

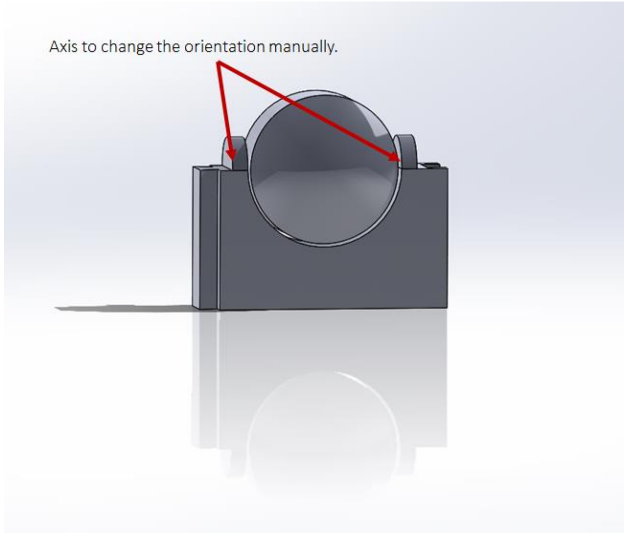

Figure 9: Single LED Block

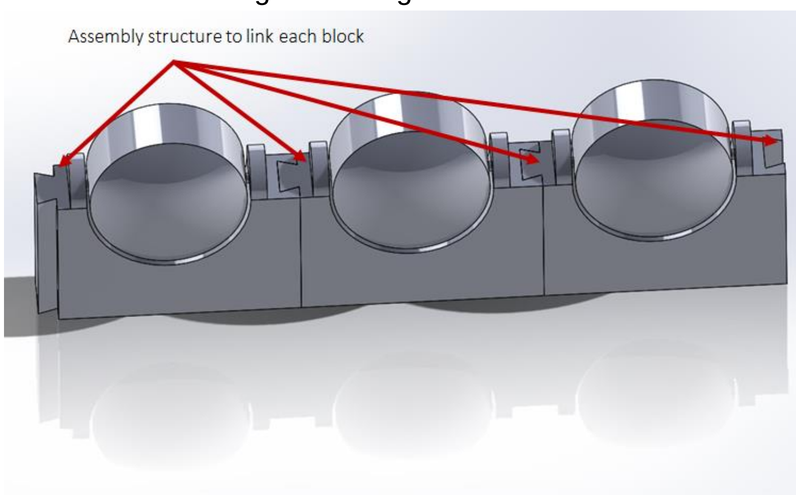

Figure 10: Connected LED blocks

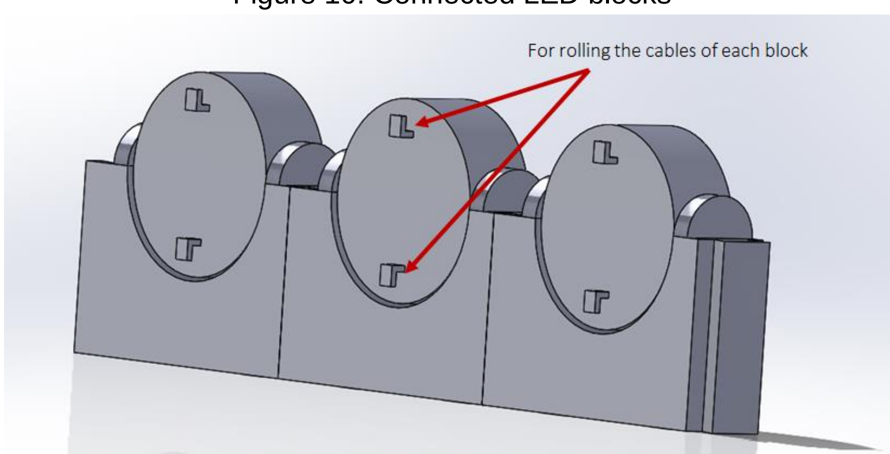

Figure 11: Cable Rolling Hooks